

**Universidade Estadual de Campinas**  
**Faculdade de Engenharia Elétrica e Computação**  
**Departamento de Eletrônica e Microeletrônica**

**“PROJETO DE UM CIRCUITO INTEGRADO DIVISOR  
DE FREQUÊNCIAS / CONTADOR DE DÉCADA EM  
TECNOLOGIA GaAs – FAMÍLIA DCFL – PARA  
OPERAÇÃO COM CLOCK NA FAIXA DE 1 GHz”**

Daniel Cardoso de Souza

Orientador  
Prof. Dr. Luiz Carlos Kretly

Dissertação apresentada à Faculdade de Engenharia Elétrica e  
Computação como parte dos pré-requisitos necessários para a  
obtenção do título de Mestre em Engenharia Elétrica.

Este exemplar corresponde a redação final da tese  
defendida por Daniel Cardoso  
de Souza e aprovada pela Comissão  
Julgada em 6 / 8 / 98

*[Assinatura]*  
Orientador

Julho – 1998.



|              |                                     |
|--------------|-------------------------------------|
| UNIDADE      | BC                                  |
| N.º CHAMADA: |                                     |
| V.           | Es.                                 |
| T. MB3 BC/   | 35247                               |
| PROC.        | 395/98                              |
| C            | <input type="checkbox"/>            |
| D            | <input checked="" type="checkbox"/> |
| PREÇO        | R\$ 11,00                           |
| DATA         | 04/10/98                            |
| N.º CPD      |                                     |

CM-00117192-3

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

So89p

Souza, Daniel Cardoso de

Projeto de um circuito integrado divisor de frequências / contador de década em tecnologia GaAs – família DCFL – para operação com clock na faixa de 1 GHz. / Daniel Cardoso de Souza.--Campinas, SP: [s.n.], 1998.

Orientador: Luiz Carlos Kretly

Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Circuitos eletrônicos – Projetos. 2. Circuitos integrados digitais. 3. Arsenieto de gálio. 4. Circuitos integrados de altíssima velocidade. I. Kretly, Luiz Carlos. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

## AGRADECIMENTOS.

A Deus;

A meus queridos pais e irmão, Edilson e Maria Guiomar Cardoso de Souza, e Ricardo Cardoso de Souza;

A meu orientador, Prof. Luiz Carlos Kretly;

Aos professores que participaram da minha banca de defesa de Mestrado: Prof. Ioshiaki Dói, Prof. Luiz C. Barbosa e Prof. Edmundo S. Braga;

À CAPES, que concedeu a minha bolsa de Mestrado por meio do programa PICDT, e à FAPESP, que arcou com o custo de fabricação e encapsulamento do CI, por meio do seu programa de PMU.

Quero agradecer às seguintes pessoas que, direta ou indiretamente, colaboraram para o término deste trabalho: Ademilde Félix, Alexander G. Ghilardi, André M. Teixeira, Antônio H. de Sousa, Antônio Ricardo Z. Nascimento, Emilio C. Bortolucci, Evandro M. Martins, Fernando F. de Melo, Hayram Nicácio, Ivo Carvalho S. Jr., Jacobus W. Swart, Marcos A. V. da Silva, Paulo H. Machado, Walter José M. de M. Jr., Walter M. S. de Marco, Wilmar B. de Moraes e Zelinda da Providência D. Cardoso.

## RESUMO.

A crescente ênfase sobre a operação portátil de computadores e sistemas de telecomunicação prioriza circuitos de baixa potência, ainda que de alta velocidade. As opções tecnológicas existentes para aplicações digitais na faixa de 100 MHz até 1 GHz são as famílias ECL em silício, DCFL em arseneto de gálio (GaAs), bem como ASICs CMOS realizados em processos avançados de Si, e somente as duas últimas podem proporcionar baixos consumos de potência. Em GaAs, DCFL é a principal opção de família digital de baixa potência.

Neste trabalho, descreve-se o projeto *full-custom* de um CI divisor de frequências de módulo variável e contador de década, realizado na família DCFL de GaAs. A topologia deste CI é inteiramente baseada na arquitetura clássica do TTL 7490, que foi escolhida por causa de sua versatilidade, e toda a sua funcionalidade lógica é mantida: o CI proposto pode operar tanto como um contador BCD quanto como um divisor de frequências por N, com N na faixa de 2 até 10. A razão da divisão, N, pode ser configurada unicamente através de conexões **diretas** entre pinos do CI. Por isso, o CI projetado neste trabalho será referido como o *7490-like*. Suas aplicações são em síntese/divisão de frequências, contagem, instrumentação de alta frequência e na composição de circuitos digitais de alta velocidade, podendo-se usá-lo na entrada de outros blocos.

Todas as configurações de operação do circuito foram simuladas no *software* HSPICE, e os resultados mostram a operação do contador com taxas de *clock* de até 1 GHz, para capacitâncias de carga totais de 5 pF nas suas saídas e uma tensão de alimentação de 2V. O CI *7490-like* foi implementado na tecnologia de MESFETs **HGaAs-III**, com comprimento de porta de 0,6  $\mu\text{m}$ , pela *foundry* norte-americana Vitesse, por intermédio do Projeto Multi-Usuário brasileiro (PMU/FAPESP) em cooperação com o CMP francês. A área total do chip foi de  $2,616 \times 2,805 \text{ mm}^2$ . Entretanto, os protótipos do CI demoraram para ser entregues à Unicamp, e não foi possível testá-los dentro do prazo de conclusão desta Dissertação.

Este projeto foi concebido principalmente para fins educacionais, mas tem um conteúdo inovativo a nível internacional, tendo em vista que não há publicação na literatura técnica internacional, até onde foi possível pesquisar, de um CI com arquitetura clássica como a do 7490, mas operando próximo a 1GHz e com tecnologia de GaAs, mantendo sua equivalência funcional. Esta Dissertação pretende ser um guia para futuros projetistas de CIs, apresentando um roteiro bastante didático das etapas de um trabalho de projeto. O CI escolhido para esse fim foi um MSI com um nível de complexidade intermediário, não sendo nem muito simples nem complexo demais, mas adequado a um primeiro projeto. Outra contribuição é a formação de recursos humanos em projeto de CIs.

Este trabalho já foi parcialmente publicado na XII Conferência da Sociedade Brasileira de Microeletrônica, realizada em 1997, com o título "Design of a 7490-like Decade Counter Integrated Circuit, Using GaAs MESFET DCFL Family, for Frequencies Up to 1 GHz".



## ABSTRACT.

The increasing emphasis on the portable operation of computers and communication systems has placed a priority on low-power, yet high-speed, circuits. The existing viable technologies for digital applications in the range from 100 MHz up to 1 GHz are Si ECL and GaAs DCFL families, as well as high-speed CMOS ASICs implemented in advanced Si processes, and only the last two options offer low power consumption. In GaAs technology, DCFL is the main choice for a low-power digital family.

In this work, a variable modulus frequency divider and decade counter IC was designed in the GaAs DCFL family. This work describes the full-custom design procedures for this IC, starting from its logic design, until the completion of the final layout version. This DCFL counter circuit topology is entirely based upon the classical TTL 7490 architecture, which was chosen because of its versatility, and all its functionality is retained: this IC can operate either as a decade (BCD) counter, or as a frequency divider by  $N$ , being  $N$  any integer in the range from 2 to 10. The frequency division modulus  $N$  can be set solely by means of **direct** connections between certain IC pins. Therefore, the IC designed in this work will be referred to as the *7490-like*. This circuit's usual applications are: frequency synthesis or division, counting, high frequency instrumentation and as a block in the composition of high speed digital circuits; the IC can also be used before the input to other blocks.

All the circuit operating configurations were simulated in the HSPICE software, and results show counter operation with clock rates up to 1 GHz, with 5 pF total load capacitances in its outputs and the use of a 2V supply. The 7490-like IC was implemented in the HGaAs-III MESFET technology, featuring 0.6  $\mu\text{m}$  gate lengths, by the north-american foundry Vitesse Semiconductor, specialized in GaAs ASICs manufacturing, via Brazilian multi-user-projects (PMU/FAPESP) cooperation with the French CMP. The total chip area was  $2,616 \times 2,805 \text{ mm}^2$ . Unfortunately, the IC prototypes took too long to arrive at Unicamp, therefore it was not possible to test them in time to meet the deadline for the conclusion of this Dissertation.

This project was envisaged **mainly** for educational purposes, but it bears an innovative content at international level, given that there is no publication in the international technical literature (at least not as far as it was possible to investigate) of an IC with a classical architecture like that of the 7490, but operating close to 1 GHz in GaAs technology, keeping its functional equivalence. This Dissertation also aims to be a guide for future GaAs IC designers, presenting a very didactic outline of the steps of a design effort. To keep with this purpose, the chosen IC was an MSI with an intermediate complexity level, which is neither too simple nor too complex, being just well-suited for a first design.

This work has been published in part at the XII Conference of the Brazilian Microelectronics Society, under the title "Design of a 7490-like Decade Counter Integrated Circuit, Using GaAs MESFET DCFL Family, for Frequencies Up to 1 GHz".

# CONTEÚDO.

|                                                                                      |      |
|--------------------------------------------------------------------------------------|------|
| Glossário de Siglas e Abreviações                                                    | viii |
| Lista de Símbolos Deste Trabalho                                                     | xi   |
| <br>                                                                                 |      |
| 1. Materiais, Dispositivos e Tecnologia de GaAs                                      | 1    |
| 1.1. A Justificativa para o Projeto de CIs em GaAs                                   | 1    |
| 1.2. Propriedades do GaAs e Comparação com o Si ao Nível de Materiais Semicondutores | 2    |
| 1.2.1. Vantagens do GaAs em relação ao Si                                            | 2    |
| 1.2.2. Desvantagens do GaAs em relação ao Si                                         | 5    |
| 1.3. Princípios dos Dispositivos Planares em GaAs e Processos de Fabricação          | 6    |
| 1.3.1. Teoria da Operação do Diodo Schottky                                          | 6    |
| 1.3.2. Teoria da Operação do MESFET                                                  | 8    |
| 1.3.2.1. Transcondutância e Frequência de Ganho Unitário                             | 11   |
| 1.3.2.2. Não-idealidades do MESFET                                                   | 11   |
| 1.3.3. Processos de Fabricação de MESFETs de GaAs                                    | 14   |
| 1.3.3.1. Os Contatos Ôhmicos                                                         | 14   |
| 1.3.3.2. O MESFET                                                                    | 15   |
| 1.4. Aplicações de CIs de GaAs                                                       | 16   |
| 1.4.1. Aplicações de Circuitos Digitais Rápidos em GaAs                              | 16   |
| 1.4.2. Aplicações de Circuitos Analógicos de Alta Frequência em GaAs                 | 17   |
| 1.5. Os Problemas do Uso do GaAs na Implementação de CIs                             | 17   |
| 1.6. Justificativa e Objetivos Deste Trabalho                                        | 19   |
| 1.7. Roteiro do Texto                                                                | 20   |
| <br>                                                                                 |      |
| 2. CIs Digitais de Alta Velocidade em GaAs.                                          | 22   |
| 2.1. Histórico do Desenvolvimento da Tecnologia de CIs Digitais de MESFETs de GaAs.  | 22   |
| 2.2. Estado-da-arte dos CIs Digitais de GaAs.                                        | 24   |
| 2.3. Características das Famílias Lógicas E/D em GaAs.                               | 26   |
| 2.3.1. Família Lógica DCFL ( <i>Direct-Coupled FET Logic</i> ).                      | 26   |
| 2.3.2. Família Lógica SCFL ( <i>Source-Coupled FET Logic</i> ).                      | 30   |

|                                                                                        |     |
|----------------------------------------------------------------------------------------|-----|
| 2.3.3. Comparação Entre as Famílias Lógicas DCFL e SCFL.                               | 32  |
| 2.4. Conclusão.                                                                        | 32  |
| 3. Projeto de Portas Lógicas Básicas em DCFL.                                          | 34  |
| 3.1. Conceitos de Projeto de CIs Digitais em GaAs, com Ênfase em DCFL.                 | 34  |
| 3.1.1. Margens de Ruído (NMs).                                                         | 34  |
| 3.1.2. Dissipação de Potência.                                                         | 37  |
| 3.1.3. Tempos de Transição e Atrasos de Propagação.                                    | 38  |
| 3.1.4. Modelos de Dispositivos para a Simulação de CIs de GaAs no Processo da Vitesse. | 40  |
| 3.2. Projeto de Inversores DCFL.                                                       | 43  |
| 3.2.1. Filosofia de Projeto.                                                           | 49  |
| 3.2.2. Parâmetros Fixos do Projeto.                                                    | 49  |
| 3.2.3. A Influência do Parâmetro $\beta$ no Projeto de um Inversor.                    | 51  |
| 3.2.3.1. Simulações DC.                                                                | 51  |
| 3.2.3.2. Simulações Transientes.                                                       | 56  |
| 3.2.4. A Influência de $W_D$ .                                                         | 62  |
| 3.2.5. Variações da Temperatura, Processo, Fonte de Alimentação e <i>Fan-Out</i> .     | 66  |
| 3.3. Projeto de Portas NOR DCFL.                                                       | 74  |
| 3.4. Projeto de <i>Drivers</i> SDCFL e <i>Buffers</i> de Entrada.                      | 80  |
| 3.4.1. Projeto de Portas Lógicas SDCFL.                                                | 80  |
| 3.4.2. Projeto dos <i>Buffers</i> de Entrada.                                          | 91  |
| 4. Projeto do CI Contador e Divisor de Frequências 7490-like.                          | 94  |
| 4.1. Contador de Década TTL 7490.                                                      | 94  |
| 4.2. Projeto Lógico do Circuito Integrado 7490-like.                                   | 98  |
| 4.3. Projeto Elétrico do Circuito Integrado 7490-like.                                 | 101 |
| 4.3.1. Especificações.                                                                 | 101 |
| 4.3.2. Projeto do FF J'K' para Atender às Especificações do 7490-like.                 | 104 |
| 4.3.3. Descrição do 7490-like no Simulador.                                            | 109 |
| 4.4. Conclusão.                                                                        | 115 |

|                                                                                                                     |     |
|---------------------------------------------------------------------------------------------------------------------|-----|
| 5. <i>Layout</i> do CI 7490-like.                                                                                   | 116 |
| 5.1. Informações Gerais para o <i>Layout</i> de CIs.                                                                | 116 |
| 5.2. As Camadas do Processo da Vitesse e suas Características.                                                      | 117 |
| 5.3. Problemas de <i>Layout</i> de CIs.                                                                             | 123 |
| 5.3.1. <i>Layout</i> de CIs Digitais de Alta Velocidade                                                             | 126 |
| 5.4. Desenho do <i>Layout</i> do CI 7490-like no <i>Software Magic</i> .                                            | 129 |
| 5.4.1. A Configuração do Arquivo de Tecnologia para o <i>Magic</i> .                                                | 129 |
| 5.4.2. As Etapas para o Desenho do <i>Layout</i> .                                                                  | 130 |
| 5.4.2.1. <i>Layout</i> dos Flip-Flops.                                                                              | 131 |
| 5.4.2.2. Projeto do Resistor de Terminação de Entrada.                                                              | 135 |
| 5.4.2.3. Projeto da Distribuição da Alimentação e dos <i>Pads</i> .                                                 | 136 |
| 5.4.2.4. LVS.                                                                                                       | 138 |
| 5.4.3. Extração do Circuito e de Parasitas.                                                                         | 142 |
| 5.5. Encapsulamento e Diagrama de Pinagem.                                                                          | 144 |
| 5.6. Recomendações para o Teste do CI.                                                                              | 145 |
| 6. Conclusões e Sugestões para Futuros Trabalhos.                                                                   | 149 |
| Apêndice 1.                                                                                                         |     |
| Roteiro para a Inclusão de Gráficos de Simulações do HSPICE em um Documento do Word for Windows 7.0.                | 152 |
| Arquivos Netlist de Entrada para Simulação no HSPICE.                                                               | 153 |
| Apêndice 2.                                                                                                         |     |
| Arquivo de Tecnologia “cmp.tech26” Fornecido pelo CMP para o <i>Software Magic</i> , Processo HGaAs-III da Vitesse. | 220 |
| Apêndice 3.                                                                                                         |     |
| Simulações das Configurações de Divisão de Frequência do CI 7490-like.                                              | 233 |
| Apêndice 4.                                                                                                         |     |
| Células de Portas Lógicas Usadas no <i>Chip 7490-like</i> .                                                         | 239 |
| Apêndice 5.                                                                                                         |     |
| O Novo Processo HGaAs-IV.                                                                                           | 261 |
| Referências Bibliográficas.                                                                                         | 264 |

## GLOSSÁRIO DE SIGLAS E ABREVIACÕES.

AMS: *Austria Mikro-Systeme* (Micro-Sistemas da Áustria).

ASIC: *Application-Specific Integrated Circuit* (Circuito Integrado de Aplicação Específica).

BCD: *Binary-Coded Decimal* (Decimal Codificado em Binário).

BDCFL: *Buffered DCFL* (DCFL “Bufferizada”). O mesmo que SDCFL.

BFL: *Buffered FET Logic* (Lógica FET com Buffer).

BiCMOS: *Bipolar-CMOS*.

BJT: *Bipolar Junction Transistor* (Transistor Bipolar de Junção).

CAD: *Computer-Aided Design* (Projeto Auxiliado por Computador).

CCS: Centro de Componentes Semicondutores.

CDFL: *Capacitor-Diode FET Logic* (Lógica FET de Capacitor e Diodo).

CI: Circuito Integrado.

CIF: *Caltech Interchange Format* (Formato para Intercâmbio do Caltech).

CML: *Current-Mode Logic* (Lógica em Modo de Corrente).

CMOS: *Complementary MOS* (MOS Complementar).

CMP: *Circuits Multi-Projets* (Circuitos Multi-Projetos).

DBS: *Direct Broadcast Satellite* (Satélite de Radiodifusão Direta).

DCFL: *Direct-Coupled FET Logic* (Lógica FET de Acoplamento Direto).

DCTL: *Direct-Coupled Transistor Logic* (Lógica de Transistores de Acoplamento Direto).

DFET: *Depletion FET* (FET de Depleção).

D.P: *Delay x Power product* (produto atraso x potência).

ECL: *Emitter-Coupled Logic* (Lógica com Acoplamento de Emissor).

EDA: *Electronic Design Automation* (Automação de Projeto Eletrônico).

EFET: *Enhancement FET* (FET de Enriquecimento).

ES2: *European Silicon Structures* (Estruturas de Silício da Europa).

FET: *Field-Effect Transistor* (Transistor de Efeito de Campo).

FF: Flip-Flop.

FL: *FET Logic* (Lógica de FET).

GPS: *Global Positioning System* (Sistema de Posicionamento Global).

GSI: *Graphical Simulation Interface* (Interface de Simulação Gráfica).

HBT: *Heterojunction Bipolar Transistor* (Transistor Bipolar de Heterojunção).

HEMT: *High Electron Mobility Transistor* (Transistor de Alta Mobilidade Eletrônica).

IGFET: *Insulated-Gate FET* (FET de Porta Isolada).

I/O: *Input/Output* (Entrada/Saída).

JFET: *Junction FET* (Transistor de Efeito de Campo de Junção).

LBFL: *Low-power BFL* (BFL de Baixa Potência).

LDCC: *Lead Ceramic chip Carrier* (Porta-chip Cerâmico com Pinos).

LED: *Light-Emitting Diode* (Diodo Emissor de Luz).

LSCFL: *Low-power SCFL* (SCFL de Baixa Potência).

LSI: *Large-Scale Integration* (Integração em Alta Escala).

MCM: *Multi-Chip Module* (Módulo Multi-Chips).

MES: *MEtal-Semiconductor* (Metal-Semicondutor).

MESFET: *MEtal-Semiconductor FET* (Transistor de Efeito de Campo de Junção Metal-Semicondutor).

MMIC: *Monolithic Microwave Integrated Circuit* (Circuito Integrado Monolítico de Microondas).

MOS: *Metal-Oxide-Semiconductor* (Metal-Óxido-Semicondutor).

MS: *Master-Slave* (Mestre-Escravo).

MSI: *Medium-Scale Integration* (Integração em Média Escala).

NMOS: *N-channel MOS* (MOS de Canal N).

PCS: *Personal Communications Services* (Serviços de Comunicação Pessoal).

PLD: *Programmable Logic Device* (Dispositivo Lógico Programável).

PLL: *Phase-Locked Loop* (Malha de Fase Travada).

PML: *Philips Microwave Limeil*.

PMU: Projeto Multi-Usuário.

RAM: *Random-Access Memory* (Memória de Acesso Aleatório).

RISC: *Reduced Instruction Set Computer* (Computador com Conjunto de Instruções Reduzido).

SBD: *Schottky-Barrier Diode* (Diodo de Barreira Schottky).

SCFL: *Source-Coupled FET Logic* (Lógica FET com Acoplamento pela Fonte).

SDCFL: *Source-follower DCFL* (DCFL com Seguidor de fonte). O mesmo que BDCFL.

SDFL: *Schottky Diode-FET Logic* (Lógica FET com Diodo Schottky).

SDH: *Synchronous Digital Hierarchy* (Hierarquia Digital Síncrona).

SI: *Semi-Insulating* (Semi-Isolante).

SOI: *Silicon On Insulator* (Silício Sobre Isolante).

SONET: *Synchronous Optical NETwork* (Rede Óptica Síncrona).

SPEC: *Systems and Processes Engineering Corporation* (Corporação de Sistemas e Processos de Engenharia).

SPICE: *Simulation Program with Integrated Circuit Emphasis* (Programa de Simulação com Ênfase em Circuitos Integrados).

SSI: *Small-Scale Integration* (Integração em Pequena Escala).

TTL: *Transistor-Transistor Logic* (Lógica Transistor-Transistor).

UHF: *Ultra High Frequency* (Frequência Ultra Alta).

VHSIC: *Very High Speed Integrated Circuit* (Circuito Integrado de Velocidade Muito Alta).

VLSI: *Very Large-Scale Integration* (Integração em Escala Muito Alta).

## LISTA DE SÍMBOLOS DESTE TRABALHO.

$a$  = prefixo atto ( $10^{-18}$ ).

$A$  = área da seção transversal de um diodo; área de porta de um MESFET; área da seção transversal de uma linha de interconexão.

$A_v$  = ganho de tensão.

$b$  = altura da área ativa implantada, em um MESFET ou SBD.

$\beta$  = razão entre as razões  $W_g/L_{g,proj}$  dos EFETs e DFETs, na lógica DCFL.

$\beta_0$  = parâmetro de transcondutância de um MESFET.

$c$  = velocidade da luz no vácuo ( $3 \cdot 10^8$  m/s).

$C_D$  = capacitância de depleção de uma junção MES ou **pn**.

$C_{FO}$  = capacitância devida ao *fan-out* na saída de uma porta.

$C_{GS}$ ,  $C_{GD}$  = capacitâncias de depleção (não-lineares) das junções de porta-fonte e porta-dreno de um FET.

$C_{GSP}$ ,  $C_{GDP}$ ,  $C_{DSP}$  = capacitâncias parasitas entre os terminais de porta e fonte, porta e dreno e dreno e fonte, respectivamente.

$C_{in}$  = capacitância de entrada de uma porta lógica.

$C_{j0}$  = capacitância de depleção de uma junção não-polarizada.

$C_L$  = capacitância de carga na saída de uma porta ou circuito lógicos.

$C_{wire}$  = capacitância parasita total de uma linha de interconexão.

$E$  = campo elétrico no canal de um FET.

$E_c$  = valor do campo elétrico crítico para a máxima velocidade dos elétrons.

$\epsilon_r$  = constante dielétrica.

$f$  = prefixo femto ( $10^{-15}$ ); frequência de operação de um circuito.

FI: *fan-in*.

FO: *fan-out*.



$f_t$  = frequência de ganho de corrente unitário de um FET.

$g_{ds}$  = condutância entre dreno e fonte de um FET.

$g_m$  = transcondutância de um FET.

$g_{m,e}$  = transcondutância extrínseca de um FET.

$g_{m,i}$  = transcondutância intrínseca de um FET.

$I$  = corrente genérica.

$I_D$  = corrente direta de um diodo; corrente de dreno de um FET.

$I_{D,D}$  = corrente de dreno de um DFET.

$I_{D,E}$  = corrente de dreno de um EFET.

$I_{D,sat}$  = corrente de saturação de um FET.

$I_{GS}$  = corrente da junção de porta-fonte de um MESFET.

$I_{maxP2}$  = corrente máxima em um pad que seja ligado ao circuito por “metal 2”.

$I_{maxP3}$  = corrente máxima em um pad que seja ligado ao circuito por “metal 3”.

$I_{maxP4}$  = corrente máxima em um pad que seja ligado ao circuito por “metal 4”.

$I_s$  = corrente de saturação do diodo, tipicamente na faixa de  $10^{-15}$  a  $10^{-14}$  A.

$I_{SUB}$  = corrente de sub-limiar do MESFET.

$J_D$  = designação de um DFET em circuitos, e para simulação no HSPICE.

$J_E$  = designação de um EFET em circuitos, e para simulação no HSPICE.

$J_s$  = densidade de corrente na seção transversal de um diodo.

$k$  = constante de Boltzmann ( $1,381 \times 10^{-23}$  J/K); fator de escalonamento linear de um circuito.

$L$  = comprimento de uma interconexão do circuito; o comprimento de um resistor (da área ativa); a indutância de uma interconexão do circuito.

$\lambda$  = parâmetro de modulação do comprimento do canal.

$L_D$  = comprimento de porta de um DFET (valor de projeto).

$L_E$  = comprimento de porta de um EFET (valor de projeto).

$L_g$  = valor real do comprimento de porta de um FET; o comprimento de um diodo.

$L_{g,proj}$  = valor de projeto do comprimento de porta de um FET; valor de projeto do comprimento de um diodo.

$m$  = coeficiente de dopagem.

$\mu_n$  = mobilidade de elétrons livres.

$\mu_p$  = mobilidade de lacunas.

$N$  = razão da divisão de frequências do CI 7490-like, podendo estar na faixa de 2 até 10.

$n$  = coeficiente de emissão ou fator de idealidade do diodo ( $\cong 1$ ); ou o número de entradas que chaveiam simultaneamente em uma porta NOR.

$N_D$  = concentração de impurezas doadoras (dopantes).

$NM$  = margem de ruído.

$NM_H$  = margem de ruído alta ( $NM_H = V_{OH} - V_{IH}$ ), conforme a figura 3.1.

$NM_L$  = margem de ruído baixa ( $NM_L = V_{IL} - V_{OL}$ ), conforme a figura 3.1.

$P_{D, dyn}$  = dissipação de potência dinâmica.

$P_{D, st}$  = dissipação de potência estática.

$pw$  = largura do ciclo positivo de um sinal (*pulse width*).

$q$  = carga do elétron ( $1,6 \cdot 10^{-19}$  C).

$R$  = resistência de um resistor do *chip*.

$R_D$  = resistência parasita série de dreno de um FET.

$r_{ds}$  = resistência de pequenos sinais entre dreno e fonte de um FET.

$R_L$  = resistência de carga na saída de um circuito, para simulação.

$R_{out}$  = resistência de saída de uma porta lógica ou *driver*.

$R_s$  = resistência de folha de uma camada de interconexão.

$R_S$  = resistência parasita série de fonte de um FET; resistência série de um SBD.

$R_{wire}$  = resistência série de uma trilha de interconexão.

$t$  = espessura de uma linha de interconexão.

$T$  = temperatura absoluta.

$\tau$  = período de um sinal.

$\Delta t$  = o intervalo de tempo de carga ou descarga de um capacitor.

$t_{HL}$  = tempo de transição de um sinal do nível alto para o baixo. Ver figura 3.4.

$t_{LH}$  = tempo de transição de um sinal do nível baixo para o alto. Ver figura 3.4.

$t_{PD}$  = atraso de propagação de um sinal em um circuito lógico, definido como a média entre  $t_{PHL}$  e  $t_{PLH}$ .

$t_{PD,LT}$  = atraso de propagação de um sinal em uma linha de transmissão.

$t_{PHL}$  = atraso de propagação de um sinal, para a saída variando do nível alto para o baixo. Ver figura 3.4.

$t_{PLH}$  = atraso de propagação de um sinal, para a saída variando do nível baixo para o alto. Ver figura 3.4.

$v$  = velocidade de deriva de portadores de carga.

$v_{sat}$  = valor saturado da velocidade de deriva de portadores de carga.

$V_{BG}$  = tensão de *backgate*.

$V_{bi}$  = tensão da barreira de potencial de uma junção.

$V_D$  = tensão no terminal de dreno de um FET.

$V_{D,e}$  = tensão aplicada sobre um diodo real.

$V_{D,i}$  = tensão aplicada sobre um diodo ideal, ou tensão intrínseca sobre a junção do diodo.

$V_{DS}$  = tensão entre dreno e fonte de um FET.

$V_{DS,sat}$  = tensão entre dreno e fonte para o início da saturação de um FET.

$V_F$  = tensão direta aplicada sobre uma junção MES.

$V_G$  = tensão no terminal de porta de um FET.

$v_{gs}$  = tensão de pequeno sinal entre os terminais de porta e fonte de um FET.

$V_{GS}$  = tensão DC entre os terminais de porta e fonte de um FET.

$V_{GS,e}$  = tensão DC extrínseca entre os terminais de porta e fonte de um FET.

$V_{GS,i}$  = tensão DC intrínseca entre os terminais de porta e fonte de um FET.

$V_{IH}$  = mínima tensão na entrada de um circuito lógico, que este ainda interpretará como um nível lógico alto “1”.

$V_{IL}$  = máxima tensão na entrada de um circuito lógico, que este ainda interpretará como um nível lógico baixo “0”.

$V_{in}$  = tensão de entrada em um circuito qualquer.

$V_{MS}$  = margem de ruído segundo o “método do máximo quadrado”.

$V_{MS,H}$  = margem de ruído alta segundo o “método do máximo quadrado”.

$V_{MS,L}$  = margem de ruído baixa segundo o “método do máximo quadrado”.

$V_{OH}$  = tensão típica de nível lógico alto na saída de um circuito lógico.

$V_{OL}$  = tensão típica de nível lógico baixo na saída de um circuito lógico.

$V_{out}$  = tensão de saída de um circuito qualquer.

$v_p$  = velocidade de propagação de um sinal em uma linha de transmissão.

$V_{pp}$  = excursão pico-a-pico de uma tensão.

$V_R$  = tensão reversa aplicada sobre uma junção MES.

$V_T$  = tensão térmica, dada por  $kT/q$  e igual a 26 mV na temperatura ambiente.

$V_t$  = tensão de limiar de um FET.

$V_{t,D}$  = tensão de limiar de um FET de depleção.

$V_{t,E}$  = tensão de limiar de um FET de enriquecimento.

$W$  = largura de uma interconexão do circuito; largura de um resistor (da área ativa).

$W_D$  = largura de porta de um DFET.

$W_E$  = largura de porta de um EFET.

$W_g$  = largura de porta de um FET; largura de um diodo.

$x_n$  = altura da região de depleção, abaixo da barreira Schottky de um MESFET ou SBD.

# CAPÍTULO 1.

## Materiais, Dispositivos e Tecnologia de GaAs.

### 1.1. A Justificativa para o Projeto de CIs em GaAs.

A **principal** razão para o emprego de transistores de GaAs na implementação de circuitos é a velocidade de chaveamento muito alta desses dispositivos, no caso de circuitos digitais, ou a largura de banda muito alta, no caso de circuitos analógicos. Assim, as portas lógicas feitas com transistores de GaAs impõem muito menos atraso de propagação aos sinais, do que portas equivalentes feitas com transistores de silício, e assim os circuitos digitais de GaAs alcançam velocidades de operação muito mais altas, e com um consumo total de potência relativamente baixo. A grande motivação para o GaAs, portanto, é esta: **alcançar-se o melhor desempenho possível com um CI**. Existem ainda outras fortes razões para o emprego de dispositivos de GaAs, como por exemplo, para se obter circuitos tolerantes a radiação, circuitos para operação em temperaturas extremas, ou para se obter sistemas optoeletrônicos mais eficientes (isto será detalhado adiante).

Os atrasos de propagação mínimos para portas lógicas de GaAs são da ordem de dezenas de picosegundos, contra centenas de picosegundos para portas na tecnologia ECL F100K da National [1], que é a família digital comercial mais rápida em Si. Já no tocante ao consumo de potência, embora as portas lógicas de GaAs dissipem uma potência estática, ao contrário, por exemplo, das portas CMOS, elas consomem muito menos potência dinâmica [2], e dessa maneira, o produto atraso-potência (D.P), que é um índice de mérito comparativo, coloca algumas famílias lógicas de GaAs em vantagem de desempenho, se comparadas tanto com CMOS quanto com ECL. Alguns gráficos comparativos entre famílias de Si e GaAs são apresentados em [4], [5] e [6].

Atualmente, os CIs de GaAs já são um sucesso comercial, graças à acelerada expansão das telecomunicações nas faixas de UHF e microondas, motivada por um mercado de consumidores finais e de altíssima demanda, como é o caso da telefonia e computação móveis e da televisão via satélite. O futuro do emprego do GaAs na Eletrônica e Optoeletrônica é ainda mais promissor [7].

O processo de fabricação dos CIs de GaAs adapta um pouco da tecnologia dos CIs de Si, principalmente CMOS. Hoje, CIs VLSI de GaAs, com mais de um milhão de transistores, já se tornaram uma realidade, exibindo um desempenho que seria impossível, ou pelo menos difícil, de ser atingido com CIs equivalentes de Si. Portanto, a tecnologia de CIs VLSI de GaAs já tem sido empregada para realizar-se versões mais rápidas de outros CIs em Si. Em alguns casos, o ganho de velocidade oferecido pelos transistores de GaAs chega a reduzir ou eliminar a necessidade de um processamento paralelo de dados, resultando em um sistema de implementação lógica mais simples, compacta e de menor custo, porque usa menos transistores [2]. Em um futuro próximo, a mesma tecnologia será certamente usada para a construção de chips optoeletrônicos ou todo-ópticos, aproveitando a propriedade de *bandgap* direto do GaAs.

Os dois principais dispositivos eletrônicos construídos em tecnologia de GaAs são o diodo Schottky e o transistor MESFET. No campo dos dispositivos ópticos, LEDs, *lasers*, células solares e fotodetectores são as principais aplicações da tecnologia.

## 1.2. Propriedades do GaAs e Comparação com o Si ao Nível de Materiais Semicondutores.

A obtenção do arseneto de gálio, enquanto matéria-prima para a Microeletrônica, é bem mais complicada que a obtenção do silício, pois o GaAs é um cristal semicondutor composto (binário) que não existe na natureza.

O gálio é um elemento raro, tóxico, sendo obtido como um co-produto do refino de bauxita para a produção de alumínio, e também como um co-produto da obtenção de zinco. Processos padrão de purificação permitem obter-se gálio com altíssima pureza. O Ga é totalmente aplicado em processos de fabricação de semicondutores.

O arsênio é um co-produto (muito venenoso) da fusão de minérios de zinco e chumbo, podendo também ser obtido a partir de compostos sulfurosos como  $\text{As}_2\text{S}_4$  ou  $\text{As}_2\text{S}_3$ . A estrutura mais estável do arsênio é um metal cristalino. O As é produzido em quantidades cinco vezes maiores que o gálio, de onde resulta que é bem mais barato que este, porém é mais difícil de purificar do que o Ga. Somente 20% da produção total de arsênio são usados na indústria de semicondutores.

O Ga é um elemento químico da coluna IIIA da tabela periódica, e o As é um elemento da coluna VA; por isso, o GaAs é referido como um **composto III-V**. Além dele, existem vários outros semicondutores compostos de elementos das colunas III e V.

### 1.2.1. Vantagens do GaAs em relação ao Si.

As vantagens do GaAs sobre o Si, para a fabricação de dispositivos de alta velocidade e componentes optoeletrônicos, têm sua causa básica nos diagramas de bandas de energia para os dois materiais.

GaAs é um semicondutor de *bandgap* direto, isto é, o mínimo energético da sua banda de condução e o máximo energético da sua banda de valência estão situados no mesmo momento. Essa propriedade é muito favorável para a construção de dispositivos emissores de luz de GaAs porque, em um material de *bandgap* direto, um elétron desce da banda de condução para a de valência sem a emissão de um fonon, a qual ocorreria se houvesse também uma mudança de momento. Ou seja, a recombinação é mais eficiente no GaAs [5]. Isto torna o seu uso muito interessante para a implementação de dispositivos ópticos, e facilita a integração destes com dispositivos eletrônicos, formando funções optoeletrônicas. O Si, por outro lado, é um semicondutor de *bandgap* indireto porque a menor separação energética entre sua banda de condução e sua banda de valência não ocorre no mesmo momento.

As estruturas das bandas de energia do Si e do GaAs levam às suas curvas de velocidade de deriva dos elétrons,  $v$ , em função do campo elétrico aplicado,  $E$ , mostradas na figura 1.1. Seja a mobilidade de um portador de carga definida como a derivada da sua

velocidade em relação ao campo elétrico que o acelera, e simbolizada por  $\mu_n$ , para os elétrons, e  $\mu_p$ , para as lacunas. Analisando-se a curva de  $v \times E$  do GaAs na figura 1.1(a), observa-se que seus elétrons livres têm uma  $\mu_n$  muito alta para pequenas intensidades de  $E$  (cerca de seis a oito vezes maior do que no Si, na região de baixos campos  $E$ ), o que significa uma alta velocidade dos elétrons para baixas tensões aplicadas (baixos campos elétricos no material).

Esta característica do GaAs é decorrente da estrutura de sua banda de condução, que apresenta um “vale” estreito, onde a massa efetiva dos elétrons é muito pequena. Os elétrons no Si têm massa efetiva bem maior do que no GaAs, daí eles terem uma  $\mu_n$  bem menor (sua  $v$  aumenta lentamente com a aplicação de  $E$ ). Esta propriedade do GaAs leva a uma redução dos tempos de atraso de propagação de dispositivos, e é a principal responsável por torná-lo mais atrativo que o Si para transistores de alta frequência, que também precisam operar com baixas tensões para economizar potência. Tempos de trânsito tão pequenos quanto 15 a 10 ps, correspondendo a frequências de ganho de corrente unitário,  $f_t$ , na faixa de 15 a 25 GHz, podem ser obtidos para MESFETs com portas de comprimentos  $L_g$  na faixa de 0,5 a 1  $\mu\text{m}$ , representando uma vantagem de três a cinco vezes sobre FETs de Si [5].

Em seguida,  $v$  atinge um pico em um valor relativamente baixo de campo elétrico (o campo crítico  $E_c$ , tipicamente de 3,5 kV/cm), e a partir daí começa a decair lentamente para maiores valores de  $E$  ( $\mu_n$  pequena e negativa). A  $v$  no GaAs satura para campos  $E$  mais intensos, e seu valor na saturação, embora seja só um pouco maior do que o do Si, já é atingido com um campo  $E$  quatro vezes menor do que neste material [5].

Na figura 1.1, à medida que  $N_D$  aumenta,  $\mu_n$  diminui, isto é, diminui  $v$  para um mesmo valor de  $E$ , por efeito de maior espalhamento nas impurezas presentes na rede cristalina. Observa-se na figura 1.1(c) que, em temperaturas próximas de 0 K, as  $\mu_n$ 's são muito maiores do que na temperatura ambiente (300 K). Curvas de  $\mu_n$  em função da temperatura, parametrizadas para vários valores de  $N_D$  e  $L_g$ , são apresentadas em [6].

Além da vantagem intrínseca do GaAs sobre o Si no tocante a  $\mu_n$ , a tecnologia de fabricação dos substratos de GaAs onde serão processados os circuitos também colabora para melhorar ainda mais a velocidade desses circuitos. O fato é que o GaAs intrínseco pode ser crescido com uma alta resistividade, da ordem de  $10^7$  a  $10^9 \Omega\cdot\text{cm}$ , chamado de GaAs semi-isolante (SI). Um substrato SI de GaAs já proporciona, por si só, uma melhor e mais fácil isolamento entre os dispositivos em um CI, impedindo o fluxo de cargas que poderia causar alguma interação em altas frequências entre os FETs, portanto simplificando o processo de fabricação e viabilizando maior densidade de integração. Outra grande vantagem do substrato SI de GaAs é que ele proporciona baixas capacitâncias parasitas dos dispositivos para o substrato, menores do que para dispositivos de Si de mesma área (contudo, as tecnologias SOI já viabilizam capacitâncias ao substrato igualmente baixas para transistores de Si). Enfim, um substrato SI de GaAs também permite a construção de indutores e capacitores de melhor qualidade, e aumenta a imunidade dos transistores à radiação ionizante.

O GaAs possui alta tolerância natural à radiação ionizante, maior do que a do Si. Componentes de GaAs podem ser empregados para a construção de circuitos eletrônicos para ambientes com altos níveis de radiação. Algumas aplicações que exigem esses componentes *rad-hard* são: espaciais (satélites), nucleares (usinas), militares (sistemas de guiamento de mísseis), e em equipamentos médicos e industriais. Nestes campos de aplicação, os componentes ficam sujeitos a colisões de íons de alta energia que, apesar de não destruírem os

dispositivos, causam uma alteração de estados lógicos e um erro na informação que estiver armazenada ou sendo transmitida [8].

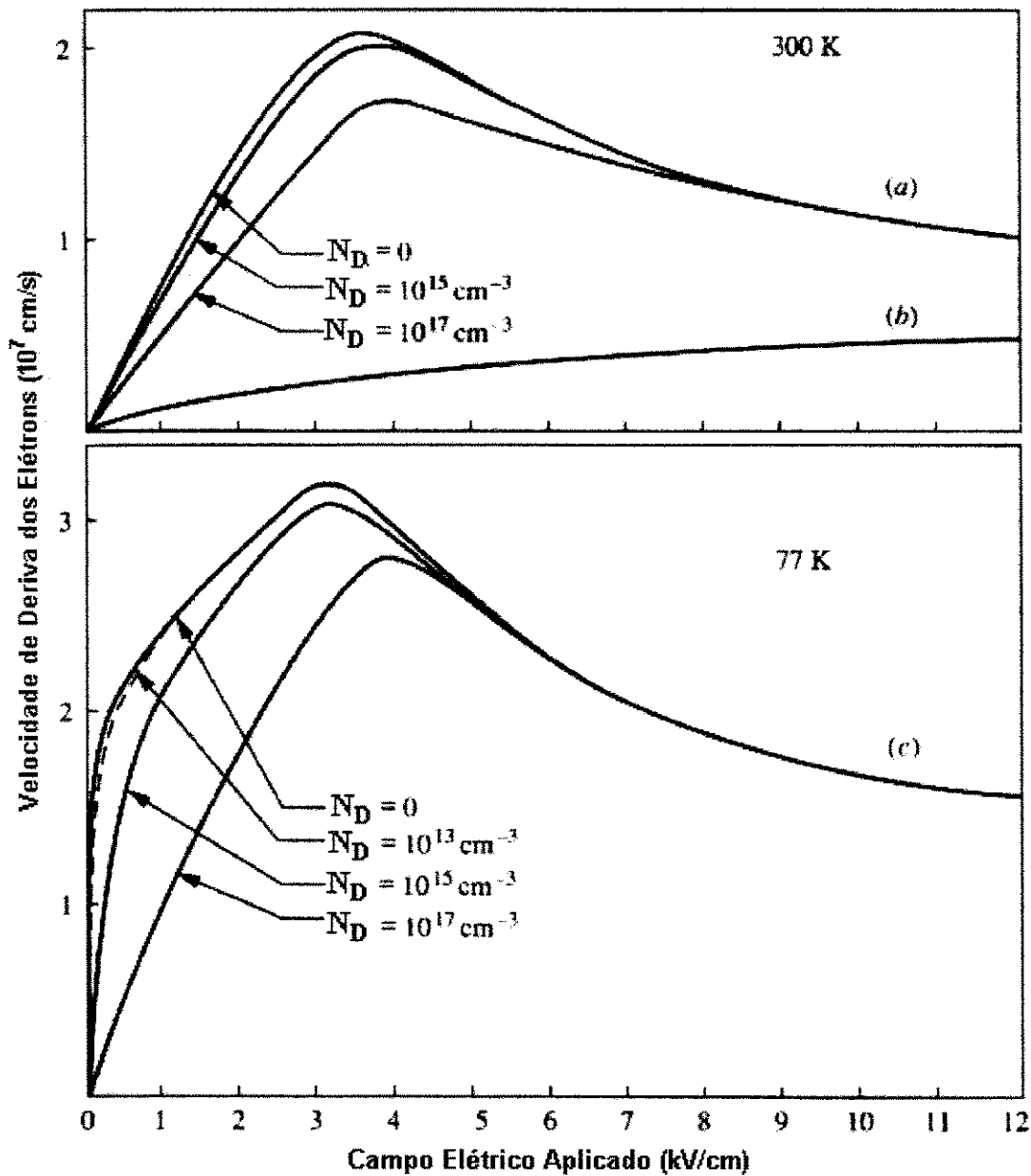


Figura 1.1 – Características de  $v \times E$  para o GaAs [(a) e (c)] e o Si (b), ambos com dopagem tipo n, tendo  $N_D$  como parâmetro [11].

O maior valor do *bandgap* do GaAs permite que ele possa operar em uma faixa mais ampla de temperaturas do que o Si, indo de  $-200$  a  $200$  °C [5]. Os elétrons no GaAs também apresentam menor potência de ruído do que no Si [9], viabilizando dispositivos de menor ruído em altas frequências.

Por causa de todas essas vantagens do GaAs, chegou-se a pensar que ele viria a substituir o Si para a construção de CIs. Mas o Si é superior em algumas características físicas, que são enumeradas na próxima seção, e os dispositivos de Si também foram se tornando mais rápidos e atingindo maiores frequências.



### 1.2.2. Desvantagens do GaAs em relação ao Si.

A tecnologia do GaAs é mais cara e complexa do que a do Si. A primeira dificuldade é o maior custo das *wafers* SI de GaAs, pois o processo de obtenção do cristal é mais difícil. O Si é cerca de vinte vezes mais barato do que o GaAs [10], devido à abundância de matéria-prima de boa pureza (quartzo). O Si também é mais robusto, ou seja, menos quebradiço, e de fácil purificação para a forma monocristalina. O GaAs, por outro lado, requer muitos cuidados extras na fabricação e manipulação das *wafers*, e a sua tecnologia de fabricação é recente em comparação com a do Si, que já vem sendo aprimorada há muito mais tempo. Portanto, mesmo que algumas topologias em GaAs garantam uma vantagem técnica quando comparadas com o Si, o fator custo muitas vezes é decisivo para a escolha.

Não se conseguiu realizar ainda um óxido estável, não-reativo e compatível com o GaAs, o que permitiria a fabricação de MOSFETs de GaAs. Assim, nos FETs de GaAs viáveis na prática, o eletrodo de controle do canal tem que ser ou uma junção Schottky metal/semicondutor (MESFET), ou uma junção **pn** (JFET) e, em ambos os casos, a tensão de porta é limitada pela condução direta dessas junções. Para o Si, ao contrário, existe um óxido compatível, estável e de ótima qualidade, o  $\text{SiO}_2$ , crescido sobre a superfície do Si para isolar o terminal de porta e o canal. Isso gera um MOSFET, que permite uma excursão muito maior da tensão de porta sem que haja a condução desta, levando a maiores margens de ruído.

Outra desvantagem do GaAs, em comparação com o Si, surge quando se tenta fabricar dispositivos complementares de GaAs, como MESFETs de canal **p**. A mobilidade das lacunas é menor do que a dos elétrons em ambos os materiais. O problema é que, no GaAs, a disparidade entre esses valores de mobilidade é muito grande:  $\mu_n = 5000 \text{ cm}^2/(\text{V.s})$  contra  $\mu_p = 250 \text{ cm}^2/(\text{V.s})$ . A  $\mu_p$  no GaAs é até mesmo um pouco menor do que no Si, para as mesmas concentrações de átomos aceitadores. Assim, FETs de GaAs de canal **p** só iriam introduzir um grande atraso de propagação se fossem usados em portas lógicas, ou uma limitação de frequência de operação, se usados em aplicações analógicas. Como a principal razão para se usar dispositivos de GaAs é atingir velocidades e frequências cada vez maiores, os dispositivos de canal **p** quase não são empregados na construção de circuitos.

O Si possui uma condutividade térmica cerca de três vezes maior que o GaAs [10], facilitando a dissipação de calor pelo *chip*, evitando a formação de “pontos quentes” e permitindo maiores níveis de integração.

As diversas vantagens físicas do Si o fazem ser a única escolha para a grande maioria das aplicações, que exigem baixo custo e produção em massa. A escala de integração em Si há muito tempo já excedeu a marca de um milhão de transistores por *chip*. Seu processo de fabricação nessas escalas já está bem caracterizado, e isso concorre para a redução de custo dos CIs fabricados e a sua maior confiabilidade, com um produto D.P razoável. Enquanto isso, a tecnologia de MESFETs de GaAs deve ser reservada para aplicações que requeiram o seu desempenho superior.

Uma tabela comparando as propriedades eletrônicas do GaAs e do Si, a nível de materiais semicondutores, é mostrada em [11]. Para uma discussão bastante aprofundada da física e das propriedades de transporte de dispositivos de GaAs, ver [6].

### 1.3. Princípios dos Dispositivos Planares em GaAs e Processos de Fabricação.

#### 1.3.1. Teoria da Operação do Diodo Schottky.

A junção metal-semicondutor (MES), ou contato Schottky, exibe uma barreira de potencial eletrostático na interface entre os dois materiais, chamada de barreira Schottky. A junção MES serve como eletrodo de porta para a construção de um MESFET, ou pode ser usada simplesmente como um diodo Schottky; mas também é empregada para fazer-se os contatos dos terminais de fonte e dreno do MESFET com o seu canal. As diferenças de fabricação e funcionamento entre a junção MES destes contatos ôhmicos e a junção retificadora do diodo Schottky, serão esclarecidas na seção 1.3.3.1.

Os diodos Schottky para aplicações em CIs de GaAs são planares, isto é, ambos os contatos (Schottky e ôhmico) ficam na superfície de uma camada condutiva de GaAs tipo **n**, a qual é definida sobre um substrato semi-isolante de GaAs. A figura 1.2 mostra um desenho em corte do SBD e o seu símbolo esquemático. A largura do diodo,  $W_g$ , que é definida como a largura do metal Schottky de ânodo, é medida perpendicularmente ao plano da página.

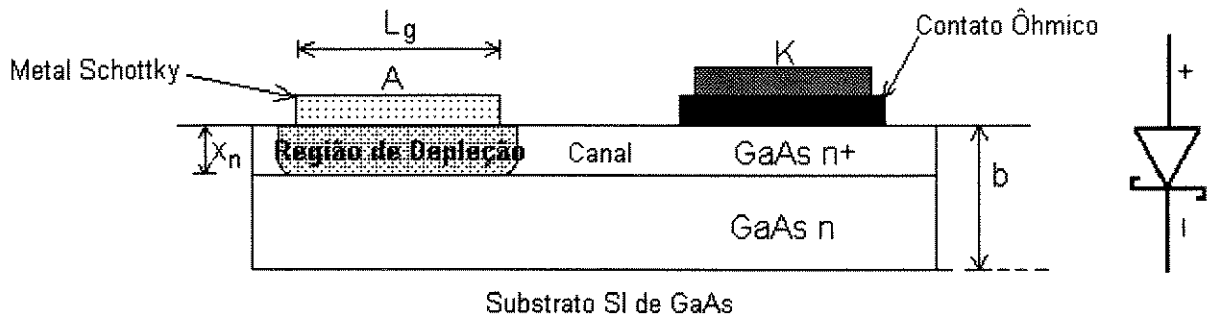


Figura 1.2 – Esquema em corte longitudinal do SBD e símbolo para circuitos.

Em toda a literatura sobre o assunto, é comum representar-se o SBD de GaAs pelo mesmo símbolo do diodo de junção **pn**, prática que também será adotada no restante deste trabalho. Portanto, o símbolo esquemático da figura 1.2 não será mais usado.

Ao contrário do diodo de junção **pn**, o SBD é um dispositivo exclusivamente de portadores majoritários. Apesar desta diferença fundamental, a corrente no SBD ainda depende exponencialmente da tensão intrínseca da junção,  $V_{D,i}$  segundo a mesma equação do diodo **pn** ideal:

$$I_D = I_S \left[ \exp \left( \frac{V_{D,i}}{nV_T} \right) - 1 \right] \quad (1.1)$$

A equação (1.1) descreve a característica I-V de um diodo ideal, isto é, apenas a junção em si. Diodos reais também incluem um efeito parasita na forma de uma resistência em série  $R_s$ , que provém da resistência dos contatos, da resistência de corpo do semiconductor, e de outros efeitos.  $R_s$  é inversamente proporcional a  $W_g$ , e a queda de tensão sobre ela reduz a tensão vista pelo diodo ideal interno,  $V_{D,i}$ , conforme a equação:

$$V_{D,i} = V_{D,e} - I_D R_s \quad (1.2)$$

onde  $V_{D,e}$  é a tensão externa aplicada nos terminais do diodo real. Correntes diretas altas no diodo são limitadas por  $R_S$ . A equação (1.2) pode ser visualizada no circuito equivalente do SBD, mostrado na figura 1.3:

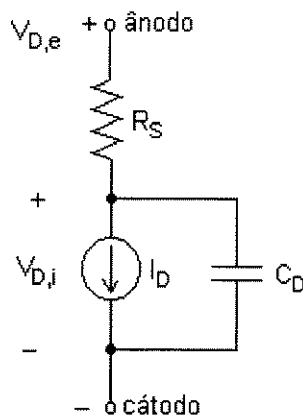


Figura 1.3 – Circuito equivalente do modelo do diodo Schottky usado no SPICE.

Neste circuito,  $I_D$  é uma fonte de corrente controlada por  $V_{D,i}$ , provendo a corrente do diodo segundo a equação (1.1).  $C_D$  é uma capacitância não-linear que modela a carga armazenada na região de depleção da junção MES do SBD, e a sua equação é igual a da capacitância de depleção de uma junção **pn**.

A constante de tempo que limita a resposta em frequência de um SBD é dada por  $R_S.C_D$ . Além de  $C_D$ , existe também a capacitância parasita geométrica do dispositivo, que aparece em paralelo com os contatos, mas na maioria dos casos  $C_D$  é muito maior do que essa capacitância [6]. No SBD não existe a capacitância de difusão (que é a principal causa da capacitância direta da junção **pn**), e assim o seu tempo de trânsito é muito menor do que o do diodo de junção **pn**, o que leva a uma resposta muito mais rápida sob polarização direta. Isso possibilita ao SBD ser usado como componente para circuitos na faixa de microondas.

Nas regras de projeto de *foundries*, normalmente o comprimento  $L_g$  do contato Schottky já vem pré-fixado em um valor adequado para reduzir a  $R_S$  dos diodos; no processo da Vitesse, usado neste trabalho, todos os diodos devem ter  $L_g = 0,6 \mu m$  (como estabelecido na página 4-9 de [12], que é o manual de projeto da Vitesse e descreve o seu processo HGaAs-III de  $0,6 \mu m$ ). Já a largura  $W_g$  dos diodos pode ser escalonada pelo projetista, conforme as exigências da aplicação no circuito. Mas se o projetista tiver controle também sobre  $L_g$ , é preciso lembrar que a corrente em um diodo diminui exponencialmente com o aumento do seu comprimento [11]. A razão para isso é que, quanto maior  $L_g$ , maior a queda  $IR$  no diodo e menor a tensão intrínseca  $V_{D,i}$ , diminuindo a corrente. Por isso, de qualquer maneira, os diodos devem ser projetados com  $L_g$  fixo, e a sua área deve ser ajustada por meio de  $W_g$ .

A dependência dos parâmetros do diodo com a área da sua seção transversal,  $A = W_g.L_g$ , é linearmente proporcional, conforme indicado nas relações (1.3). Os parâmetros do modelo de diodo do SPICE, usados para o SBD de GaAs, e métodos para a sua extração, são explicados em [11] e [13].

$$A \uparrow \Rightarrow I_S \uparrow, C_{j0} \uparrow, R_S \downarrow \quad (1.3)$$

### 1.3.2. Teoria da Operação do MESFET.

O MESFET é um dispositivo de estrutura planar, sendo o tipo de transistor mais conveniente para a fabricação em GaAs, pois, ao contrário do JFET, o MESFET não requer a utilização de regiões com dopagem tipo p.

A mesma discussão feita para o MESFET também é válida para o JFET, pois ambos são dispositivos unipolares de lei quadrática (não são exponenciais como o BJT). A diferença entre o JFET e o MESFET é que, neste, o contato da porta é uma junção MES retificadora (um diodo Schottky), e no JFET a junção da porta com o canal é uma junção pn, ambas desempenhando a mesma função. Porém, como os dois tipos de junção têm basicamente as mesmas características, o MESFET resulta muito parecido com o JFET a nível de terminais (mas é mais rápido). A estrutura idealizada em seção longitudinal de um MESFET é mostrada na figura 1.4.

A dopagem e a espessura da camada ativa tipo n são dimensionadas de modo que o canal possa ser totalmente depletado por meio da tensão  $V_{GS}$  aplicada entre porta e fonte, havendo, idealmente, o corte da corrente de dreno do transistor para  $V_{GS}$  menor ou igual a um certo valor, chamado de tensão de limiar (*threshold voltage*),  $V_t$ .

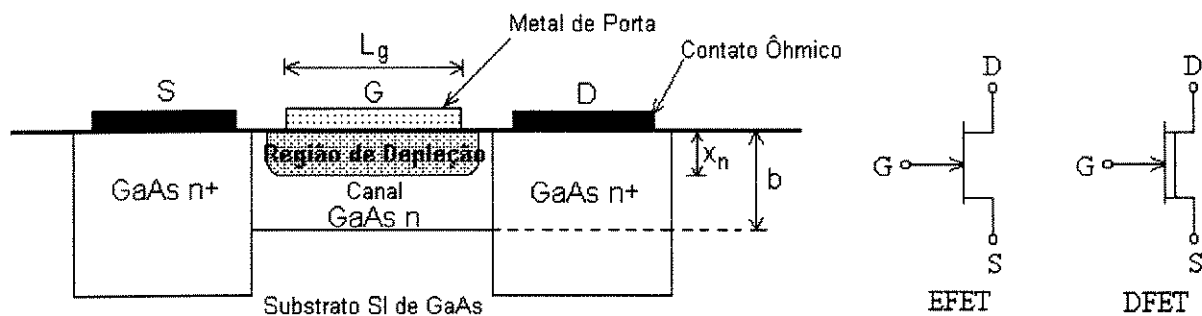


Figura 1.4 – Seção de um MESFET. A largura  $W_g$  do transistor é perpendicular ao plano da página, e  $x_n$  é a profundidade da região de depleção abaixo da porta.

Os desempenhos relativos de transistores de GaAs e Si têm sido muito discutidos na literatura [6]. As principais diferenças entre os MESFETs de GaAs e os MOSFETs de Si estão nas seguintes características:

1. **A forma de acoplamento entre o contato de porta e o canal:** a porta metálica de um MESFET faz contato direto com o canal, enquanto que, no MOSFET, o eletrodo de porta é separado do canal por uma fina camada isolante de  $\text{SiO}_2$ . Assim, a junção Schottky de porta do MESFET conduz uma corrente  $I_{GS}$  para a sua fonte, quando polarizada diretamente. Com o fluxo de corrente para o terminal de porta, o nível lógico alto fica grampeado em 0,7 V pela tensão da barreira de potencial.
2. **O mecanismo de formação do canal:** o canal de um MESFET, quer ele seja de enriquecimento ou de depleção, já é previamente implantado durante o seu processo de fabricação, e a sua espessura é controlada pela tensão aplicada na junção MES. O canal do transistor NMOS de enriquecimento é formado durante a operação do dispositivo em um circuito, quando a superfície do Si tipo p é invertida, passando a conter uma alta densidade de elétrons. E o MOSFET de depleção é formado com uma leve dopagem do canal, para deslocar o seu limiar de operação  $V_t$  para um valor negativo e torná-lo normalmente ligado.

3. **A viabilidade de transistores complementares:** não compensa utilizar FETs de GaAs de canal **p** para a construção de circuitos, porque suas frequências  $f_t$  são muito mais baixas que as dos FETs de canal **n**. A razão para isso é que  $\mu_p \ll \mu_n$ . As vantagens de uma lógica complementar (que usa transistores complementares na mesma porta lógica) podem ser vistas comparando-se NMOS com CMOS: maior excursão da tensão de saída e menor dissipação de potência. Circuitos complementares com JFETs de GaAs foram construídos e demonstraram uma velocidade razoável de chaveamento, porém limitada pela baixa  $\mu_p$ , que leva a uma transcondutância muito baixa em **p**-JFETs. O projeto complementar de CIs de GaAs só é útil quando uma redução de potência torna-se obrigatória [14].

4. **Menores capacitâncias:** um FET de GaAs não tem junções **pn** em torno de seus contatos de fonte e dreno, e portanto as capacitâncias inter-eletródicas em dispositivos de GaAs são muito menores. Essas diferenças entre os dispositivos influenciam os projetos de circuitos em GaAs e em MOS de modo decisivo, tornando-os bastante diferentes.

Os MESFETs atualmente usados em CIs podem ser de dois tipos: de enriquecimento (EFET) ou de depleção (DFET), dependendo do sinal de  $V_t$ : o EFET tem uma  $V_t$  positiva, enquanto o DFET tem uma  $V_t$  negativa. O MESFET de enriquecimento é dito “normalmente desligado” porque um  $V_{GS}$  positivo é necessário para abrir o canal para os elétrons fluírem da fonte para o dreno; isto é, ele não “funciona” com  $V_{GS} = 0$ . O MESFET de depleção é dito “normalmente ligado” porque ele requer a aplicação de uma tensão  $V_{GS}$  negativa para cortar a corrente de dreno,  $I_D$ ; isto é, ele “funciona” mesmo com  $V_{GS} = 0$  V.

Tanto EFETs como DFETs operam pelo mesmo princípio, a depleção de um canal já existente e dopado; a região de depleção é obtida através da junção MES da porta. As alturas dos canais de ambos os tipos de MESFET são controladas pela altura da região de depleção,  $x_n$ , através de  $V_{GS}$ . A única diferença é que o canal do DFET está aberto para  $V_{GS} = 0$ .

A figura 1.5 ilustra as regiões de operação de um FET genérico. Inicialmente, observa-se que as curvas características I-V de um FET são simétricas em torno da origem dos eixos, por causa da simetria de construção do dispositivo, resultando na existência das mesmas regiões de operação no primeiro e no terceiro quadrantes.

No **corte**, a região de depleção da porta penetra completamente através do canal, bloqueando o fluxo de elétrons e cortando  $I_D$ . Em uma primeira aproximação, o FET está no corte enquanto  $V_{GS} \leq V_t$ , mas se for desejada maior precisão de análise, pode-se considerar a existência da  $I_D$  residual para essa região de operação, que passa a se chamar **região de sub-limiar**. Mais detalhes sobre a região de sub-limiar serão dados adiante.

Quando  $V_{GS}$  é maior do que o valor de  $V_t$  do dispositivo, e uma pequena tensão  $V_{DS}$  é mantida aplicada ao FET, o seu canal passa a se comportar como um resistor de seção transversal retangular e área igual a  $W_g(b - x_n)$ , onde  $x_n$  é definida na figura 1.4. Se  $V_{GS}$  aumenta,  $x_n$  diminui proporcionalmente, aumentando a área do resistor de canal e reduzindo a sua resistência. Essa região de operação, válida para pequenos valores de  $V_{DS}$ , é chamada de **região linear ou ôhmica** da curva I-V do MESFET, porque nela o transistor se comporta como uma resistência linear  $r_{ds}$ , inversamente proporcional a  $V_{GS}$ . Da mesma forma que em um MOSFET, quanto maior a tensão  $V_{GS}$ , maior será a inclinação da “reta”  $I_D \times V_{DS}$  na região linear, que representa a condutância de pequenos sinais entre dreno e fonte:

$$g_{ds} = \left[ \frac{\partial I_D}{\partial V_{DS}} \right]_{V_{GS} \text{ constante}} \quad (1.4)$$

A linearidade da resistência de canal  $r_{ds}$  ocorre dentro de uma faixa limitada de  $V_{DS}$ . À medida em que  $V_{DS}$  aumenta, a inclinação da curva  $I_D \times V_{DS}$  diminui gradualmente, e  $r_{ds}$  aumenta. Depois que  $V_{DS}$  atinge e supera o seu valor de saturação  $V_{DS,sat}$ , com  $V_{GS} > V_t$ ,  $I_D$  torna-se quase constante e igual a um valor  $I_{D,sat}$ , e assim permanece para aumentos posteriores de  $V_{DS}$ , o que pode ser constatado na figura 1.5. Nessa nova região de operação, chamada de **região de saturação**, o FET opera por um novo mecanismo, havendo dois modelos para explicar o seu funcionamento: o modelo clássico de canal gradual, proposto por W. Shockley, e o modelo de saturação da velocidade dos elétrons. Uma análise matemática detalhada dos dois modelos é apresentada em [6].

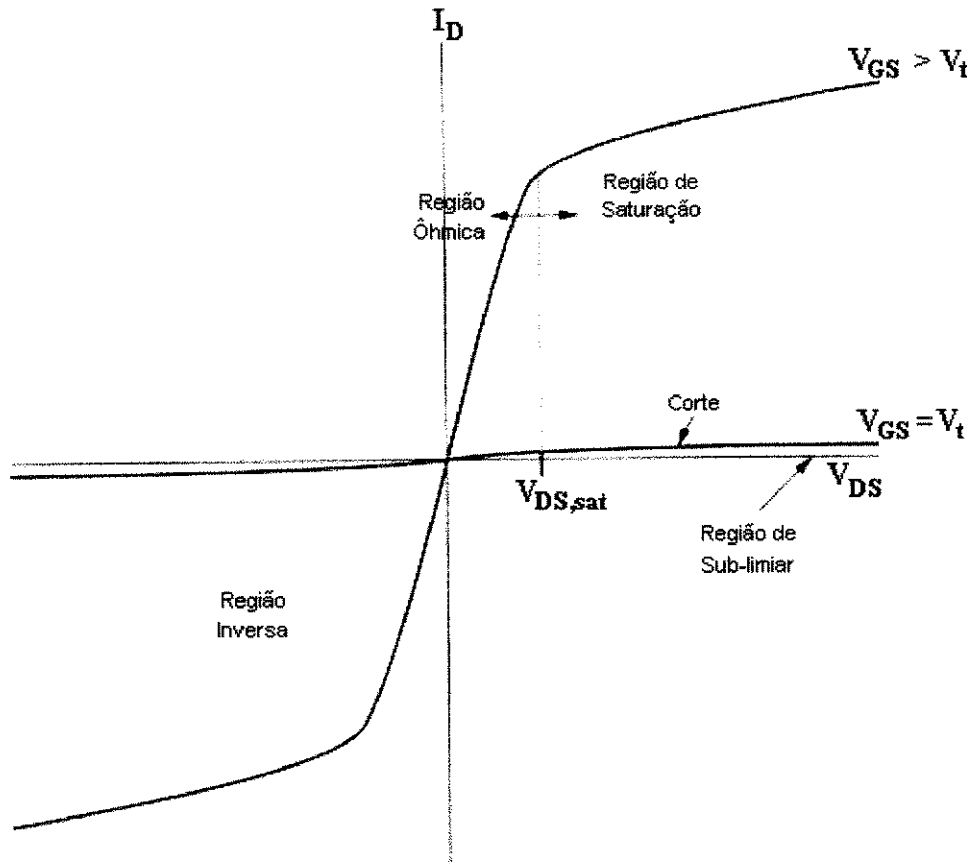


Figura 1.5 – Regiões de Operação de um FET qualquer.

O modelo de canal gradual aplica-se bem somente a FETs de porta longa ( $L_g$  muito maior do que a altura do canal,  $b - x_n$ ), e portanto não será usado neste trabalho. Já o modelo de saturação de velocidade atribui a saturação de  $I_D$  à saturação da velocidade dos elétrons no canal, que começa a ocorrer com o crescimento de  $E$  além de  $E_c$  ( $= 3,5 \text{ kV/cm}$ ), como visto na figura 1.1. Este modelo dá uma descrição melhor do funcionamento de MESFETs de porta curta ( $L_g < 2 \mu\text{m}$ ), porque  $E_c$  pode ser atingido, no canal, com uma menor  $V_{GS}$ , já que

$$E_c = \frac{V_{GS} - V_t}{L_g} \quad (1.5)$$

e assim o canal não chega a sofrer constrição para haver a saturação de  $I_D$ , porque a velocidade dos elétrons atinge um valor limite antes que isso aconteça, se  $L_g$  é curto. O ponto onde  $v$  satura inicialmente é a extremidade do canal próxima ao dreno. A tensão  $V_{DS,sat}$  e a corrente  $I_{D,sat}$ , para FETs de canal curto, são muito menores do que os valores previstos pelo modelo de canal gradual; a menor  $I_{D,sat}$  deve-se ao fato de que  $v$  agora tende a um limite.

Quando  $V_{DS}$  é aumentado já dentro da região de saturação, a corrente  $I_D$  continua a crescer, embora mais lentamente do que na região linear. O mecanismo que gera essa condutância finita  $g_{ds}$  na saturação é chamado de modulação do comprimento do canal; a interpretação do fenômeno, para FETs de canal curto, é dada em [11].

Menores comprimentos de porta  $L_g$  levam a maiores correntes de dreno por unidade de largura de porta,  $W_g$  [6].  $I_D$  só é inversamente proporcional a  $L_g$  para FETs de porta longa; para dispositivos de porta curta, se  $L_g$  é reduzido ainda mais,  $I_D$  também cresce, mas a uma taxa menor do que  $1/L_g$ . Isso se deve aos efeitos de saturação da velocidade.

A corrente  $I_D$  do MESFET, na região linear e na saturação, é diretamente proporcional a  $W_g$ , para  $W_g \geq 3 \mu m$ . Para larguras menores, os efeitos de canal estreito (frangeamentos laterais) invalidam essa relação, e além disso as tolerâncias do processo na definição de larguras causam flutuações indesejáveis no valor projetado de  $I_D$ . Por isso, MESFETs estreitos não devem ser utilizados.

#### 1.3.2.1. Transcondutância e Frequência de Ganho Unitário.

A transcondutância ( $g_m$ ) de um FET de GaAs é muito maior do que a de um FET de Si de mesmo  $L_g$ , tipicamente por um fator de três ou quatro, em níveis de tensão usuais. A alta  $g_m$  melhora muito a velocidade porque, com uma excursão  $v_{gs}$  menor obtém-se a mesma corrente de pico, para carregar ou descarregar uma capacitância de carga. Para fins de projeto,  $g_m$  é diretamente proporcional a  $W_g$ , e também é maior para MESFETs de menor  $L_g$ .

A **frequência de ganho de corrente unitário**,  $f_t$ , de um MESFET, que é o seu produto (ganho de corrente)  $\times$  (largura de banda), determina a velocidade máxima de chaveamento do transistor. O MESFET não sofre efeitos de armazenagem de portadores minoritários, o que lhe confere alta velocidade de chaveamento e altas  $f_t$ 's. As capacitâncias não-lineares e as parasitas fixas reduzem a  $f_t$ , e para combater o seu efeito é importante que o transistor possua uma alta  $g_m$ .

O valor de  $f_t$  é inversamente proporcional ao  $L_g$  do MESFET, contanto que as capacitâncias de porta não sejam dominadas pelas componentes de frangeamento, o que acontece para  $L_g$ 's muito curtos. Isso pode ser evitado com maiores níveis de dopagem do canal. O valor da frequência de corte  $f_t$ , para MESFETs fabricados em processos modernos, é da ordem de 35 GHz [15].

#### 1.3.2.2. Não-idealidades do MESFET.

A **resistência parasita da fonte**,  $R_s$ , é a resistência existente entre o contato de fonte e a borda da porta, proporcional à separação entre eles. O efeito de  $R_s$  é o de reduzir a tensão  $V_{GS}$  aplicada nos terminais (extrínseca),  $V_{GS,e}$ , pela queda de tensão através de  $R_s$ , como ilustrado na figura 1.6. Desse modo, a tensão  $V_{GS}$  realmente vista pela junção de porta (intrínseca),  $V_{GS,i}$ , é dada por

$$V_{GS,i} = V_{GS,e} - I_D \cdot R_S \quad (1.6)$$

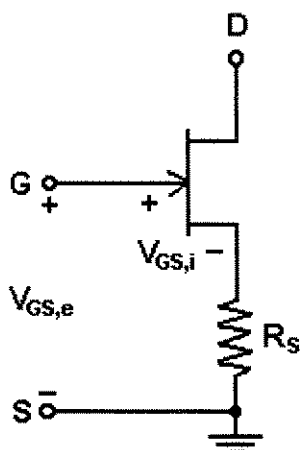


Figura 1.6 –  $R_S$  reduz a tensão  $V_{GS}$  intrínseca, diminuindo  $I_D$  e  $g_m$ .

$R_S$  também produz um efeito similar em  $g_m$ : a  $g_m$  disponível nos terminais de um MESFET é reduzida por  $R_S$ . Certamente, também existe uma resistência parasita de dreno,  $R_D$ , que deve ser igual a  $R_S$  para manter-se a simetria do dispositivo. Os valores típicos de  $R_S$  e  $R_D$  situam-se na faixa de dezenas de ohms.  $R_S$  e  $R_D$  dependem principalmente dos espaçamentos entre os terminais de porta e fonte e de porta e dreno, sendo proporcionais a esses espaçamentos.

A **corrente de porta**  $I_{GS}$  de um MESFET resulta da polarização direta da junção entre porta e fonte, uma vez que não existe isolamento entre a porta e o canal. Quando o terminal de porta é polarizado com 0,7 ~ 0,8 V acima do potencial de fonte, a corrente da junção torna-se considerável e é atingido o limite de tensão direta. Isso limita a excursão de tensão na saída de uma porta lógica que aciona a entrada de outra porta (porque limita o valor de  $V_{OH}$ ), limitando portanto as margens de ruído nos nós internos de um circuito, o que será detalhado no Capítulo 2. Este é um efeito indesejável da operação do MESFET, que não ocorre com circuitos MOS. A corrente  $I_{GS}$  faz com que o estágio acionado represente uma carga para o estágio acionador, e tem um efeito degradante sobre o desempenho de um circuito.

Se, por efeito do projeto das portas lógicas de um circuito, o  $V_{GS}$  aplicado em um FET ficar muito alto (digamos,  $V_{GS} = 0,8$  V), então  $I_{GS}$  torna-se alta demais e irá causar uma queda de tensão  $V_{DS} = I_{GS} \cdot R_S$  adicional sobre  $R_S$ . Essa tensão extra faz com que a curva característica do MESFET desloque-se para a direita por esse valor; isso pode ser visualizado nas figuras 1.5 e 3.11. Esse *offset* de  $V_{DS}$  é nocivo para uma porta lógica, porque causa um aumento da tensão de nível lógico baixo  $V_{OL}$ , com a correspondente redução da margem de ruído. Outros efeitos de uma tensão de entrada excessiva serão analisados no Capítulo 3.

O fenômeno de **backgating** (também chamado de *sidegating*) consiste na redução da corrente de dreno de um MESFET, causada pelo aumento do valor da sua tensão  $V_t$ , que por sua vez é provocado pela polarização negativa do substrato em relação ao seu canal. Este efeito é induzido em um MESFET por outros MESFETs próximos (daí o nome *sidegating*), cujas fontes estejam polarizadas negativamente em relação à fonte do primeiro MESFET. A tensão de fonte em um FET afeta a condução no canal de FETs adjacentes através de uma modulação da região de depleção existente na interface canal/substrato. Surge, portanto, uma interação parasita entre dispositivos vizinhos na superfície do substrato. Geralmente, a



intensidade dessa interação depende da distância e diferença de potencial entre os dispositivos, e é dependente do processo.

O *backgating* nos FETs de GaAs é um efeito relacionado com a junção entre o canal (área ativa) e o substrato semi-isolante que está por baixo; é equivalente ao efeito de corpo observado nos MOSFETs. A altura da região de depleção que se estende dentro do canal, junto à interface com o substrato, é quase diretamente proporcional ao valor da tensão de polarização negativa do substrato em relação ao canal [6], chamada de tensão de *backgate*,  $V_{BG}$ ; seus efeitos no formato dessa região de depleção são mostrados na figura 1.7. As resistências série de fonte e dreno  $R_S$  e  $R_D$ , também crescem com o aumento da tensão reversa  $V_{BG}$ , o que tem um efeito profundo no desempenho de circuitos de FETs de GaAs [16].

O *backgating* pode exercer uma grande influência no desempenho de circuitos digitais, mas pode ser desprezado para circuitos que operem com tensões de alimentação médias ou baixas, e para operação na temperatura ambiente ou acima dela. Os circuitos suscetíveis ao *backgating* são aqueles que operam com tensões de alimentação elevadas e/ou em baixas temperaturas (abaixo de 0°C). A influência do *backgating* em um circuito também depende do seu projeto.

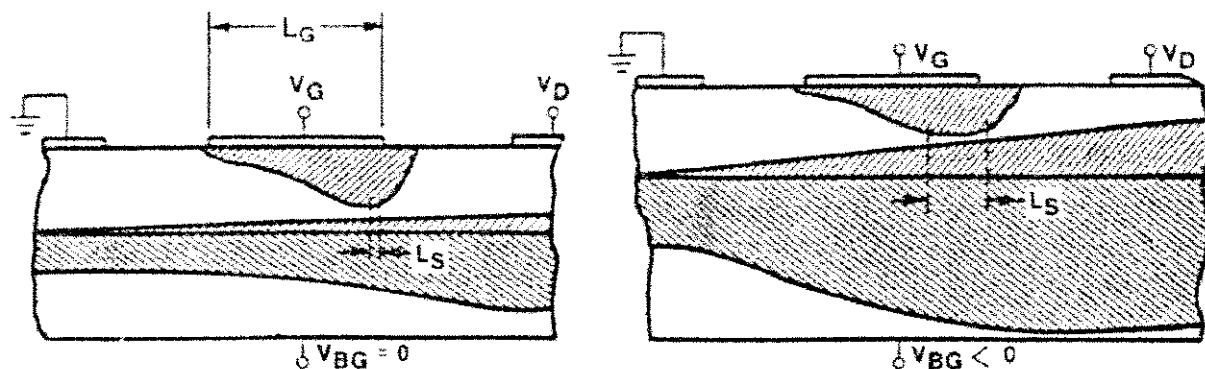


Figura 1.7 – Efeito de uma tensão  $V_{BG}$  negativa sobre a forma da região de depleção entre canal e substrato [6].

Os efeitos de *backgating* não são observados para substratos polarizados positivamente. Por isso, os CIs fabricados pela Vitesse devem receber uma polarização de substrato  $V_{BG} > 0$  V, aplicada sob os chips (por baixo da *die*), de valor igual a 0,6 V acima da menor tensão de alimentação aplicada ao CI. Os encapsulamentos LDCC oferecidos pela Vitesse possuem um pino próprio para a aplicação da tensão  $V_{BG}$ , e o modelo de MESFET da Vitesse para o SPICE inclui o nó de *backgate* para simular a influência dessa polarização. Para uma explicação mais detalhada e um equacionamento dos mecanismos desse fenômeno, consultar [11], [6] e [16].

A corrente de dreno de um FET não se anula quando  $V_{GS} \leq V_t$ , existindo uma pequena corrente  $I_D$  de fuga residual que é chamada **corrente de sub-limiar**,  $I_{SUB}$ . Essa corrente depende exponencialmente de  $V_{DS}$  e  $V_{GS}$ , e é inversamente proporcional a  $L_G$ .  $I_D$  continua a fluir mesmo com  $V_{GS}$  estando muitos décimos de volt abaixo de  $V_t$ , o que seria a região de corte em uma aproximação de primeira ordem.

A corrente  $I_{SUB}$  é irrelevante para o projeto de circuitos lógicos otimizados para alta velocidade, nos quais a dissipação de potência é de média para alta ( $> 1$  mW / porta lógica), mas é muito importante para o modelamento de circuitos de potência muito baixa ( $< 0,1$  mW /

porta lógica), ou para circuitos sensíveis a correntes de fuga, como circuitos de lógica dinâmica ou memórias.  $I_{SUB}$  aumenta exponencialmente com a temperatura, como todas as correntes de fuga, portanto circuitos de baixa potência operando em altas temperaturas têm mais chance de serem afetados.

As curvas características I-V do MESFET são muito **dependentes da temperatura**. O valor da tensão  $V_t$  e o parâmetro de transcondutância  $\beta_0$  do MESFET diminuem com o aumento da temperatura  $T$ , então  $I_D$  poderá tanto aumentar como diminuir com a temperatura, dependendo de qual parâmetro,  $V_t$  ou  $\beta_0$ , exercer o efeito mais forte. O resultado observado é que a variação de  $I_D$  com  $T$  depende de  $V_{GS}$ . Para valores pequenos de  $V_{GS}$ , o efeito de  $V_t$  é dominante e  $I_D$  aumenta com a temperatura; para valores grandes de  $V_{GS}$ , a influência de  $\beta_0$  é que domina, e  $I_D$  diminui com a temperatura. Portanto, existe um valor de  $V_{GS}$  no qual  $I_D$  é independente de  $T$ , o que pode ser aproveitado para circuitos de compensação térmica. E quando  $V_{GS}$  polariza diretamente a junção de porta, a temperatura do canal do dispositivo aumenta por causa da maior  $I_{GS}$ , reduzindo  $I_D$  pela redução da mobilidade.

Todo projeto deve ser capaz de funcionar com um desempenho satisfatório mesmo sob condições ambientais extremas. As equações do MESFET embutidas no SPICE consideram a variação da temperatura e seu efeito no comportamento do circuito.

### 1.3.3. Processos de Fabricação de MESFETs de GaAs.

O mesmo processo é usado para formar o SBD e o MESFET. O eletrodo de porta do MESFET e o eletrodo de ânodo do SBD são formados pela deposição de uma liga metálica referida como “metal de porta”, diretamente sobre um canal de GaAs tipo **n** (área ativa implantada). A dopagem tipo **n** do canal é produzida por implantação iônica de Si ou Se sobre o substrato SI de GaAs. O Si e o Se são dopantes tipo **n** para o GaAs.

#### 1.3.3.1. Os Contatos Ôhmicos.

É necessária uma etapa de metalização separada para os contatos ôhmicos, em processos de CIs de GaAs, a qual usa um metal diferente do exigido pelas portas ou diodos Schottky, e que também é diferente dos metais das camadas de interconexão.

Os contatos ôhmicos com o GaAs, com características I-V simétricas para as polarizações direta e reversa e baixa resistividade, são formados pela deposição de ligas de AuGe/Ni/Au sobre o GaAs com dopagem **n+**. Após a deposição, os metais formam um contato Schottky com o GaAs, e é feito um recozimento da *wafer*. A difusão do Ge dentro do GaAs forma um material **n+** fortemente dopado, e isso altera o mecanismo de fluxo de corrente através da barreira: a dopagem muito intensa na interface do metal com o GaAs estreita muito a barreira e também a região de depleção, e assim os elétrons podem fluir através dela por tunelamento. A característica I-V torna-se quase simétrica, porque passa a haver um mecanismo para um grande fluxo de corrente em ambas as direções; portanto, o tunelamento de elétrons através da barreira é o que leva ao comportamento ôhmico do contato. A figura 1.8 mostra a distorção sofrida pela curva característica I-V de uma junção MES, para GaAs tipo **n** moderadamente dopado e densamente dopado.

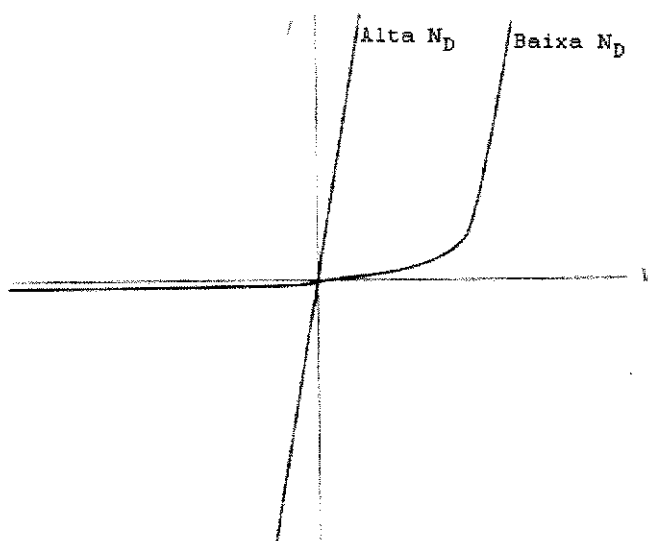


Figura 1.8 - Características I-V da barreira Schottky com baixa e alta dopagens. A corrente pode fluir em ambas as direções por tunelamento quando a dopagem é alta [11].

#### 1.3.3.2. O MESFET.

O desempenho de alta velocidade de um MESFET de GaAs depende de vários parâmetros do seu processo de fabricação, podendo ser melhorado com o emprego de menores  $L_g$ 's, maior dopagem do canal, e procurando-se minimizar  $R_s$ . À medida em que as dimensões dos dispositivos de GaAs são reduzidas, a velocidade dos elétrons aumenta, levando a um menor tempo de trânsito e, ao mesmo tempo, o consumo de potência decresce em consequência das dimensões menores [6].

Para os modernos circuitos lógicos compostos com EFETs e DFETs, é essencial que os valores de  $V_{LE}$  e  $V_{LD}$  sejam bastante uniformes em uma mesma *wafer* e de uma *wafer* para outra, viabilizando a interconexão de grandes quantidades de FETs, tanto em um mesmo *chip* quanto entre *chips* diferentes, formando circuitos digitais funcionais e confiáveis. Todas essas exigências refletem-se em exigências de qualidade dos processos de fabricação.

As etapas básicas do processamento para formar os dispositivos e interconectá-los, na fabricação de CIs de MESFETs de GaAs, são resumidas em [10] e [12].

O moderno processo auto-alinhado é uma técnica de fabricação que consiste em dopar-se as regiões de fonte e dreno de forma auto-alinhada com a porta, com o objetivo de reduzir a  $R_s$  e as capacitâncias parasitas de um FET. Antes de surgir essa técnica, as regiões de fonte e dreno eram dopadas primeiro e depois era depositado o metal de porta, que tinha que ter um sobrepasso grande sobre as áreas de fonte e dreno, gerando capacitâncias elevadas. A adoção de metais refratários (que suportam temperaturas muito altas) para a composição da porta do FET permitiu que se invertesse a sequência de fabricação, levando ao processo auto-alinhado. Neste, a porta é depositada primeiro, e depois as regiões de fonte e dreno são dopadas por implantação iônica, com a própria porta servindo de máscara para essa dopagem. A técnica de auto-alinhamento é usada tanto para a fabricação de FETs de Si quanto de GaAs.

Como o processo auto-alinhado exige o recozimento a alta temperatura do implante  $n^+$  após a implantação iônica, com os contatos Schottky de porta já depositados, faz-se

necessário empregar silicetos ou nitretos de um metal refratário (tungstênio), como  $TiW_xSi_y$  ou  $TiWSi(N)$ , para a composição das portas dos FETs. Isso visa a impedir a degradação da altura da barreira da junção Schottky, o que causaria a redução da queda de tensão direta do diodo de porta-fonte, e o aumento da corrente reversa de saturação. Os chamados “*gates refratários*”, portanto, suportam temperaturas de recozimento superiores a 800 °C. É um processo voltado para a fabricação CIs digitais, sendo empregado pela Vitesse. A sua sequência de fabricação é explicada em [10] e [11].

Para uma descrição das etapas dos processos de fabricação de CIs com MESFETs de GaAs por difusão e por implantação iônica, ver [13]. Para uma descrição aprofundada dos processos de fabricação de dispositivos, ver [6].

## 1.4. Aplicações de CIs de GaAs.

O futuro do mercado para componentes de GaAs é muito promissor, segundo várias firmas de consultoria tecnológica, graças ao estabelecimento de aplicações de alto volume de vendas e voltadas para consumidores finais. Os dispositivos de GaAs são sempre preferidos em aplicações cuja frequência de operação é muito alta, como circuitos de microondas ou sistemas ópticos. Para a mesma dissipação de potência, um circuito de GaAs costuma ser mais rápido (operar em maiores frequências) que um de Si, e, para a mesma velocidade de operação, o consumo de potência no circuito de GaAs é geralmente menor do que no de Si. Esta última qualidade é valiosa para sistemas operados com baterias: *laptops* e equipamentos para comunicações móveis pessoais.

Quer sejam analógicos ou digitais, os CIs de GaAs têm seu principal campo de aplicações na área de Telecomunicações, cujo crescimento explosivo é a força motriz por trás do crescimento do GaAs, sendo as comunicações digitais sem fio o setor de maior crescimento.

### 1.4.1. Aplicações de Circuitos Digitais Rápidos em GaAs.

CIs digitais de GaAs são usados onde há a necessidade de altas taxas de operação, principalmente em sistemas de telecomunicação por fibras ópticas, redes de dados, estações de trabalho, e em algumas aplicações de propósito especial, como supercomputação [7]. Alguns exemplos de aplicações são listados a seguir:

- Memórias de alta velocidade, com tempos de acesso da ordem de picosegundos [17];
- Microprocessadores em arquitetura RISC para computação de alto desempenho (ver [11] e [18]), como a linha Cray, por exemplo;
- Circuitos lógicos de uso geral, e também ASICs na forma de *gate arrays*, *standard cells*, PLDs e circuitos *full-custom*;
- Sistemas de alto desempenho para Telecomunicações, como circuitos multi/demultiplex, regeneradores de sinal, chaves, e circuitos de recuperação de *clock* para redes ópticas de dados SDH/SONET;
- Sistemas de alto desempenho para Teste & Instrumentação, como divisores de frequência, multiplicadores paralelos e conversores A/D e D/A.

CI's digitais de GaAs podem ser usados no *front-end* (porta de entrada) de um sistema de comunicação óptica para receber dados seriais em alta velocidade e depois interfacear diretamente com CI's CMOS de baixa velocidade, que constituem a lógica de processamento do protocolo. Nessa aplicação, os CI's de GaAs são usados para recuperação de *clock* e de dados e para realizar a conversão serial/paralela, e operam em taxas de multi-GHz. Os dados recuperados e paralelizados são então entregues aos blocos CMOS para processamento.

#### 1.4.2. Aplicações de Circuitos Analógicos de Alta Frequência em GaAs.

A principal aplicação analógica de dispositivos de GaAs é em CI's de microondas (MMICs) para sistemas de telecomunicações, especialmente de comunicações móveis (GPS, telefonia celular, PCS, *wireless phones*) e receptores de TV DBS. Exemplos de MMICs incluem amplificadores, mixadores, chaves, deslocadores de fase, osciladores, resonadores e moduladores. Os subcircuitos da parte de RF de telefones móveis (celulares ou PCS), que são implementados em GaAs, são todos analógicos: amplificadores de potência para transmissão, chaves transmite/recebe e amplificadores de baixo ruído [19]. Não há subcircuitos digitais neste tipo de aplicação [20].

A fabricação de MMICs exige a integração de MESFETs com elementos passivos (indutores e capacitores), para operação em dezenas de GHz, sendo um projeto mais crítico que o de circuitos digitais. Atualmente, a produção e venda de CI's analógicos de GaAs é muito maior do que a de CI's digitais (cerca de quatro vezes maior), porque o crescimento muito acelerado das tecnologias de comunicações sem fio criou um mercado multi-milionário e de altíssima demanda para os CI's analógicos de GaAs [20].

Nesse campo de aplicação, o GaAs enfrenta forte concorrência do Si pelo fato de este ser mais barato, e o custo é o fator mais importante na escolha da tecnologia de fabricação para circuitos de consumo. Felizmente para o GaAs, os mercados de telecomunicações (celulares, TV) estão se movendo cada vez mais para a região de microondas e ondas milimétricas, onde o somente o GaAs funciona (e muito bem). Implementar um CI ou MCM para essa mesma faixa de frequências e mantendo o mesmo desempenho, mas usando tecnologias CMOS ou BiCMOS, sairia muito mais caro. Considerando uma exigência de um sistema por alto desempenho, que precisa ser atendida, pode-se dizer que a tecnologia de GaAs oferece mesmo uma alternativa de baixo custo para o Si.

Não se pode esquecer a enorme importância do GaAs para a Optoeletrônica: desde o início da sua aplicação na fabricação de componentes, no final da década de 60, ele também vem sendo usado tanto para a emissão de luz, na forma de LEDs e, mais recentemente, de *lasers*, quanto para a sua detecção, na forma de fotodetetores (embora estes já sejam mais fabricados com outros semicondutores compostos, como InGaAsP e o InP, por uma questão de comprimento de onda da radiação emitida). Transistores de GaAs também são usados para realizar o acionador do *laser*.

#### 1.5. Os Problemas do Uso do GaAs na Implementação de CI's.

A principal desvantagem do GaAs, em relação ao Si, é o custo de fabricação de CI's. Durante o período de realização deste trabalho, o custo da fabricação de um CI de GaAs pela

Vitesse, por intermédio do CMP, esteve fixado em 2700 FF (francos franceses) por  $\text{mm}^2$  de área do *die*, aproximadamente US\$ 458/ $\text{mm}^2$  na época do envio do projeto (outubro de 1997); e este preço não incluía o encapsulamento de nenhum dos *chips* fabricados. No mesmo período, o custo da fabricação de um *chip* em CMOS 0,7 $\mu\text{m}$  pela ES2 esteve fixado em 1300 FF (~ US\$ 221) por  $\text{mm}^2$ , incluindo cinco *chips* encapsulados. A tabela a seguir compara os custos de fabricação de CIs em GaAs com os custos da fabricação de CIs em CMOS; a taxa de câmbio usada foi a de outubro de 1997, 1 US\$ = 5,895 FF.

| <i>Foundry</i> | Tecnologia                    | Preço / $\text{mm}^2$ |
|----------------|-------------------------------|-----------------------|
| ES2            | CMOS 0,7 $\mu\text{m}$        | 1300 FF = US\$ 221    |
| AMS            | CMOS 0,8 $\mu\text{m}$        | 1600 FF = US\$ 271,4  |
|                | BiCMOS 0,8 $\mu\text{m}$      | 2600 FF = US\$ 441    |
| Vitesse        | GaAs MESFET 0,6 $\mu\text{m}$ | 2700 FF = US\$ 458    |
| PML            | GaAs HEMT 0,2 $\mu\text{m}$   | 6000 FF = US\$ 1017,8 |

Tabela 1.1 – Comparação de preços entre tecnologias oferecidas pelo CMP (valores de 1997). Obs.: A lista de preços atualizada deve ser sempre obtida no *site* do CMP.

Por causa do alto custo, o potencial da tecnologia precisa ser muito bem explorado, da melhor maneira possível. Por isso é importante fazer-se uma comparação direta entre as características de GaAs e Si, tanto a nível de materiais semicondutores quanto de desempenho das portas lógicas básicas e circuitos; essa comparação é feita neste capítulo, e mais detalhadamente em [2], [5], [10] e [11].

Apesar das vantagens da tecnologia de GaAs e da mesma já ter atingido níveis VLSI, problemas ligados aos processos de fabricação ainda limitam o *yield* e o nível de integração para famílias de MESFETs E/D.

Os circuitos de GaAs descritos na literatura são caracterizados frequentemente por maiores velocidades e/ou menor consumo de potência do que seria observado em circuitos comparáveis de Si. Mas hoje já não é mais possível generalizar e afirmar que isso é sempre verdadeiro, pois os processos de Si têm se desenvolvido muito nos últimos anos. Além disso, o melhor desempenho de um *chip* de GaAs também pode custar um maior consumo de potência em comparação com o de um *chip* em CMOS, que é o preço a se pagar por um desempenho que poderia não ser viável de se atingir com outra tecnologia.

No emprego de CIs MSI e LSI de GaAs para a implementação de sistemas digitais complexos a nível de placa de circuito impresso (abordagem modular padrão), o principal problema de desempenho surge devido à alta razão entre o atraso de propagação,  $t_{PD}$ , fora dos CIs ( $t_{PD}$  *offchip*), e o  $t_{PD}$  dentro dos chips ( $t_{PD}$  *onchip*). Alguns fabricantes, tentando resolver esse problema, oferecem CIs em GaAs que são blocos LSI padrão, geralmente compatíveis com ECL. Estes componentes LSI ainda são muito mais caros que seus equivalentes ECL, mas permitem criar-se projetos de tecnologia mista. Assim, CIs de GaAs são usados ao longo de todos os caminhos críticos do projeto, enquanto as outras partes do sistema são implementadas com blocos ECL. Esta solução é economicamente interessante para a realização de sistemas digitais de alta velocidade, e daí vem a tendência de mercado da tecnologia de GaAs de oferecer componentes compatíveis com ECL [21]. Uma desvantagem dessa compatibilidade é um maior consumo de potência e menor velocidade dos componentes de GaAs. Por outro lado, isso facilita a atualização de projetos mais antigos, pela introdução de blocos de GaAs em substituição a blocos de ECL.

A referência [21] descreve a realização de um sistema processador em GaAs para emular o microprocessador 8085, analisando os problemas do projeto de sistemas processadores empregando CIs LSI de GaAs.

## 1.6. Justificativa e Objetivos Deste Trabalho.

A crescente ênfase sobre a operação portátil de computadores e sistemas de telecomunicação prioriza circuitos de baixa potência, ainda que de alta velocidade, e DCFL desponta como a principal opção de família digital de baixa potência em GaAs. Essa característica, combinada com a sua simplicidade topológica, torna DCFL a família mais promissora para CIs de GaAs em escala VLSI e ainda com grande potencial de utilização futura, considerando-se o constante aprimoramento dos processos de fabricação. Este fato justifica uma investigação das suas características e o treinamento em projetos de CIs nesta família lógica. As outras opções tecnológicas para aplicações digitais de baixa potência e alta complexidade, na faixa de 100 MHz até 1 GHz, limitam-se a ASICs CMOS realizados em processos avançados de Si, que não são disponíveis a partir de rodadas multi-projeto.

Neste trabalho, um CI divisor de frequências de módulo variável e contador de década foi projetado na família DCFL de GaAs, objetivando o estudo e domínio das características desta família lógica. Divisores de frequência de módulo variável, ou *prescalers*, são blocos importantes, por exemplo, em PLLs usadas para síntese de frequências. *Prescalers* feitos em tecnologias rápidas (ECL ou DCFL) tipicamente empregam cascatas de divisores por 2; entretanto, apenas umas poucas razões de divisão diferentes costumam estar disponíveis.

A topologia deste CI contador DCFL é inteiramente baseada na arquitetura clássica do TTL 7490, a qual foi escolhida por causa de sua versatilidade, e toda a sua funcionalidade lógica é mantida: o CI proposto pode operar tanto como um contador de década (contador BCD), ou como um divisor da frequência do sinal de entrada por N, sendo N um inteiro na faixa de 2 até 10. A razão da divisão, N, pode ser configurada unicamente através de conexões entre certos pinos do CI, sem a necessidade de qualquer componente externo. Por isso, o CI projetado neste trabalho será referido como o CI *7490-like*. Este circuito tem aplicações em síntese/divisão de frequências, contagem, instrumentação de alta frequência e na composição de circuitos digitais de alta velocidade, podendo-se usá-lo ainda na entrada de outros blocos.

Este trabalho objetiva descrever e sistematizar os procedimentos do projeto *full-custom* de um CI deste tipo, desde o seu projeto lógico (funcional) até a conclusão da versão final do *layout*, incluindo a etapa de dimensionamento das portas lógicas e a escolha de uma estratégia de *layout*.

Este projeto foi concebido principalmente para fins educacionais, mas possui um grau razoável de ineditismo a nível internacional, tendo em vista que este CI não tem similar na literatura técnica, até onde foi possível pesquisar. Ainda, é inovativa a sua escolha como veículo para treinamento de projetistas, pois ele é universalmente conhecido em cursos de graduação de Eng. Elétrica. Até a defesa desta Dissertação de Mestrado, ainda não havia sido realizado um CI similar que fosse do conhecimento do pesquisador responsável [22]. Outra contribuição é a formação de recursos humanos em projeto de CIs, tendo sido escrito um *paper* [22].

## 1.7. Roteiro do Texto.

O Capítulo 2 apresenta um resumo histórico do emprego de dispositivos de GaAs para a implementação de circuitos digitais rápidos. O estado-da-arte é comentado, e as famílias lógicas E/D mais populares em GaAs são brevemente analisadas. Maior atenção é dada à família DCFL, na qual será desenvolvido o projeto descrito neste trabalho.

O Capítulo 3 inicia com a definição e explicação de conceitos necessários para o projeto de CIs digitais, com ênfase em DCFL. São avaliados e comparados os compromissos de projeto (*trade-offs*) mais importantes em DCFL. Com o objetivo de verificar-se as relações de dependência entre o dimensionamento dos transistores de uma porta lógica e as características de desempenho DC e AC dessa porta, foram realizadas algumas séries de simulações de inversores, portas NOR e *buffers* de I/O. A partir dos resultados numéricos e gráficos dessas simulações, obtém-se o embasamento intuitivo e a experiência necessários para futuros dimensionamentos de MESFETs, em projetos de circuitos digitais DCFL.

O Capítulo 3 descreve essas séries de simulações, demonstrando técnicas de simulação em HSPICE, analisa todas as curvas geradas, e relata as conclusões derivadas desses resultados. São analisados os projetos DC e transiente de portas e *buffers*, e os compromissos entre larguras de dispositivos, dissipação de potência e atraso são determinados. Também são simuladas as influências do atraso com *fan-in*, *fan-out*, temperatura, e variações do processo.

Os Capítulos 4 e 5 descrevem os procedimentos de projeto *full-custom* para o CI 7490-like, desde suas especificações até a conclusão do *layout* final, incluindo a etapa de dimensionamento das portas lógicas e a escolha de uma estratégia de desenho de *layout*. O Capítulo 4 primeiramente enumera as especificações desejadas para o CI, e descreve o seu projeto lógico, a partir da interconexão de portas DCFL. Sua topologia 7490-like foi arquitetada de modo a preservar toda a funcionalidade original do 7490. Todas as configurações de operação do CI foram simuladas em computador, empregando o simulador de circuitos HSPICE, e os resultados comprovam a funcionalidade do CI 7490-like.

Após comprovada a funcionalidade através das simulações lógicas, foi feito o dimensionamento dos transistores das portas lógicas do circuito para a operação com taxas de clock de até 1 GHz. Primeiramente, foi dimensionado um flip-flop com o desempenho necessário para atender às especificações do 7490-like. Em seguida, foram criadas portas NOR e *buffers* de tamanhos padrão, para serem usados repetidamente como células. Com essas portas foram compostos os demais flip-flops, e foi então modelado o circuito contador completo. Os reguladores de corrente de proteção das entradas foram incluídos na simulação, assim como os *drivers* de saída com suas cargas capacitivas.

O Capítulo 5 trata principalmente do desenho do *layout* do CI: descrevem-se as características das camadas do processo da Vitesse, as etapas e procedimentos de desenho do *layout* e a extração de parasitas para o ambiente de CAD utilizado. Os *softwares* de EDA usados no projeto foram o HSPICE (para fazer-se as simulações do circuito, empregando modelos da Vitesse) e o Magic V6.3 (para o desenho do *layout* em um método *full-custom*). Os arquivos de descrição do circuito assim projetado foram submetidos para fabricação através do CMP, um consórcio de *foundries* de várias tecnologias, oferecendo serviços de prototipagem de ASICs a baixo custo para universidades, centros de pesquisa e indústrias.



Os protótipos do CI *7490-like* não foram recebidos no CCS da Unicamp em tempo para serem testados dentro do prazo para conclusão desta Dissertação. O Capítulo 5 também enumera algumas etapas que terão que ser cumpridas no processo de teste, como a confecção do *jig* de testes e a escolha do equipamento de medida, e apresenta sugestões de características e parâmetros que podem ser medidos e analisados. O Capítulo 6 sintetiza as conclusões do trabalho, as experiências adquiridas e as sugestões para futuros trabalhos.

O Apêndice 1 contém as listagens de todos os arquivos de simulação (*netlists*) para HSPICE. O Apêndice 2 fornece a listagem completa do arquivo de tecnologia “cmp.tech26” fornecido pelo CMP, após as correções que tiveram que ser realizadas no mesmo. O Apêndice 3 contém as formas de onda para todas as configurações de divisão de frequência do CI *7490-like*, desde a divisão por 2 até a divisão por 10 simétrica. O Apêndice 4 consiste das definições ao nível de *layout* de todas as células de portas e *buffers* criadas para o *chip 7490-like*, juntamente com seus esquemáticos no nível dos transistores. E o Apêndice 5 comenta sobre o novo processo HGaAs-IV que a Vitesse passa a oferecer a partir de 1998.

A bibliografia consultada foi bastante extensa, estando listada no final do trabalho, e inclui também alguns *sites* da Internet com informações sobre a área de concentração desta Dissertação.

## CAPÍTULO 2.

### CI's Digitais de Alta Velocidade em GaAs.

#### 2.1. Histórico do Desenvolvimento da Tecnologia de CI's Digitais de MESFETs de GaAs.

Nesta seção, é apresentado um resumo da evolução histórica dos circuitos integrados digitais de MESFETs de GaAs.

Em 1965, Carver A. Mead, trabalhando no Caltech, propôs e demonstrou, pela primeira vez, um transistor de efeito de campo cuja porta consistia de um diodo de barreira Schottky, que veio a ser o MESFET [23]. O autor construiu e caracterizou os primeiros dispositivos experimentais de GaAs, comprovando as previsões teóricas sobre seu funcionamento, e indicando suas vantagens. Desde o final da década de 60, o MESFET de GaAs passou a ser amplamente usado para amplificação de sinais em frequências de microondas, área que ainda domina até hoje. Em 1970, tornaram-se disponíveis os primeiros MESFETs comerciais [10]. O uso do MESFET para aplicações digitais começou em 1974, e hoje já está bem estabelecido na tecnologia LSI, inclusive com exemplos bem-sucedidos de realizações em níveis de complexidade VLSI.

As primeiras pesquisas para produzir-se CI's monolíticos de GaAs foram conduzidas pela HP, por divisões de Microeletrônica de grandes empresas do setor aeroespacial e militar, como a Rockwell, Hughes Aircraft, TRW e McDonnell Douglas, e por algumas empresas japonesas, destacando-se a Fujitsu Labs. e a NTT [24].

Em 1974, R. L. Van Tuyl e C. A. Liechti, dois engenheiros da HP, demonstraram pela primeira vez a viabilidade da aplicação de MESFETs de GaAs na implementação de CI's digitais de alta velocidade [25]. O processo usava somente DFETs, com  $L_g$  de  $1\mu\text{m}$  e uma  $f_t$  de 15 GHz. O CI era um circuito combinacional simples, utilizando a família lógica de FETs com *buffer* (BFL).

Até aquela época, os CI's digitais de alta velocidade eram baseados em lógica ECL de silício, empregando BJTs com  $f_t$  na faixa de 2 a 4,5 GHz. Como uma alternativa ao BJT, também havia o MESFET de Si, com atrasos da ordem de nano e picosegundos. Mas MESFETs fabricados em GaAs com  $L_g = 1\mu\text{m}$  exibiam  $f_t$ 's maiores que o dobro das  $f_t$ 's de MESFETs de Si com as mesmas dimensões. Os MESFETs de GaAs já eram usados em amplificadores de microondas, portanto suas características de pequenos sinais já tinham sido estudadas na literatura; o *paper* [25] foi o primeiro a demonstrar que o MESFET de GaAs poderia ser usado para um chaveamento muito rápido e que poderia ser integrado com diodos Schottky, para formar um circuito lógico com atrasos da ordem de picosegundos.

Em 1976, iniciaram-se as pesquisas em MESFETs "normalmente desligados" (de enriquecimento) de GaAs, verificando-se que seu uso viabilizaria portas lógicas de pequena área, baixa potência e alta velocidade (pequeno produto D.P). Os desenvolvimentos iniciais em EFETs de GaAs foram conduzidos por pesquisadores da Fujitsu Labs., utilizando pela primeira vez a família Lógica FET de Acoplamento Direto (DCFL), que na época era

chamada de DCTL [26]; o inversor usava um resistor como carga para as chaves a EFET. Em 1977, a Fujitsu apresentou circuitos com baixa dissipação de potência ( $264\mu\text{W}$  / FET) e um tempo de atraso de 280ps, com MESFETs de GaAs. Em 1978, um flip-flop mestre-escravo DCTL, consistindo de oito portas NOR e dois *buffers*, foi fabricado monoliticamente, apresentando máxima frequência de contagem de 610 MHz [27]. A combinação de alta velocidade e baixo produto D.P para portas de EFETs com acoplamento direto foi comprovada experimentalmente [28], tendo sido verificado que, apesar da dissipação de potência da porta ser diretamente proporcional à largura do transistor, a dependência da velocidade de operação com essa largura era fortemente sub-linear.

Com o surgimento do MESFET de enriquecimento, foi aberta a possibilidade de se combinar EFETs e DFETs em um mesmo CI monolítico [29], tendo sido explorada também uma topologia combinando EFETs e diodos túnel para realizar inversores [30], que seria depois abandonada.

Já no final da década de 70, começaram a surgir os primeiros CIs de GaAs em escala MSI, baseados somente em DFETs; exemplos incluem um contador/divisor por 8 e um multiplexador de 8 bits [31]. A partir dos resultados favoráveis observados nesses circuitos, foi reconhecido o potencial da tecnologia planar de CIs de GaAs para alcançar níveis de complexidade e velocidade muito mais altos, usando EFETs [32].

Em 1980, os CIs de GaAs já tinham seu potencial comprovado para CIs digitais da mais alta velocidade, em complexidades SSI e MSI, utilizando as topologias de depleção SDFL [33] e BFL [34]. A realização de CIs MSI com EFETs ainda não era possível por causa das exigências mais rígidas de processamento para esses dispositivos, mas os desempenhos medidos em osciladores em anel indicavam um baixíssimo produto D.P de 1,6 fJ, promissor para VLSIs [35]. Entretanto, a continuação do avanço para escalas LSI ainda estava dependendo da superação de algumas limitações em processos e tecnologia de dispositivos [36]. Discutiam-se os avanços necessários na tecnologia de CIs para tornar o GaAs viável em escalas LSI e VLSI; já era consenso que os melhores candidatos seriam MESFETs ou JFETs de enriquecimento, com sua significativa vantagem de velocidade sobre os concorrentes (o NMOS e o MESFET de Si, cujas tecnologias tinham potencial comprovado para VLSI [37], [38]). Em virtude da progressiva redução da excursão das tensões internas nos CIs, o IGFET de GaAs já não era mais tão necessário [39], apesar dos esforços para sua realização [40].

A partir de meados dos anos 80, os fabricantes norte-americanos de CIs digitais de GaAs convenceram-se da necessidade de compatibilizar seus produtos com os CIs digitais de Si, em termos de tensões de alimentação e níveis lógicos de entrada e saída, facilitando a inserção dos CIs de GaAs em diversos sistemas, cujos componentes são predominantemente de Si. A partir daí, tornou-se possível interfacear blocos feitos em ambas as tecnologias de maneira muito mais simples e direta, aproveitando as vantagens inerentes a cada tecnologia.

O grande passo para a produção em escala comercial de CIs digitais de GaAs foi o desenvolvimento de um método próprio para a fabricação dos mesmos, diferenciado do processo usado para MMICs [24] analógicos; até então, ambos usavam o mesmo processamento de CIs. O novo processo digital continuou a empregar os MESFETs de GaAs usuais, mas agora com portas feitas de um metal refratário (ao invés de ouro) e auto-alinhadas. Isso veio facilitar a fabricação e viabilizar a implementação de uma família lógica que sacrificasse um pouco da alta velocidade do GaAs em favor de um menor consumo de potência. Esta nova família foi a chamada DCFL [41], que será explicada adiante. Além disso,

este novo processo passou a adotar metalização de alumínio para interconexões, tal qual o Si. Os resultados foram um menor custo, maior rendimento de fabricação, e a possibilidade de integração em escala VLSI.

Na primeira metade dos anos 80, foram finalmente **fabricados** CIs de GaAs em LSI e VLSI, graças à maior adoção da tecnologia de implantação iônica e técnicas de estruturas auto-alinhadas. Em 1983, a NEC fabricou um CI somador LSI de 32 bits, com 420 portas lógicas [42]; a empresa optou por usar uma versão de baixa potência da topologia BFL, após tê-la comparado em desempenho com a porta básica DCFL. Porém, ainda em 1983 a Fujitsu foi pioneira em produzir um CI multiplicador paralelo de 16x16 bits em DCFL, com 3168 portas NOR auto-alinhadas [43], empregando EFETs e DFETs com  $L_g = 2\mu\text{m}$ . Foi o CI digital de GaAs mais complexo já realizado até então, e o primeiro em escala VLSI.

No campo das memórias de GaAs, a NTT fabricou a primeira RAM estática de 16 bits em 1981, e em 1984 projetou uma RAM estática de 16 kbits [13]. Nesse mesmo ano, iniciou-se a produção comercial de CIs de GaAs ([44], [45], [46]), na qual divisores de frequência na faixa de GHz foram bastante investigados ([47], [48]). Em 1985, a empresa Honeywell passou a comercializar *gate arrays* de GaAs, viabilizando o projeto de CIs por esta metodologia [10].

A partir daí, a tecnologia de GaAs disparou em complexidade, ou seja, no número de transistores por *chip*, atingindo circuitos em escala VLSI, voltados para o desenvolvimento de memórias e processadores em arquitetura RISC. Em 1987 surgiu a Lógica Diferencial de Transistores de Passagem (DPTL) [13], baseada em DFETs.

A partir da década de 90, transistores utilizando heteroestruturas como elemento de controle foram desenvolvidos. Esses transistores, que podem ser tanto FETs como bipolares, atingem maiores velocidades de chaveamento do que os MESFETs. Os dois principais representantes dessa nova geração de dispositivos são o HEMT e o HBT, usados para aplicações em muitas dezenas e até centenas de GHz (ondas milimétricas), desde MMICs para sistemas automotivos (prevenção de colisões) até sistemas de satélites. Os HEMTs recentemente emergiram como os dispositivos de estado sólido mais rápidos. Mesmo assim, dentre as várias tecnologias de CIs em GaAs existentes hoje, a tecnologia de MESFETs é a mais madura e mais comercializada.

Também em 1990 foi fabricado um microprocessador e computador totalmente em GaAs. A partir de então, o foco principal do mercado norte-americano para produtos de GaAs deslocou-se das aplicações militares para as comerciais, representadas pelo mercado das comunicações móveis (celulares e PCSs) [49].

## 2.2. Estado-da-arte dos CIs Digitais de GaAs.

O estado-da-arte da eletrônica baseada em circuitos de GaAs apresenta o uso cada vez maior de CIs digitais de alta velocidade e/ou para comunicações ópticas, e transistores de heteroestrutura.

Na área de pesquisas de ponta em Microeletrônica, que são os dispositivos de heteroestrutura em semicondutores compostos e/ou novos materiais, o projeto dos circuitos empregando esses dispositivos torna-se intimamente relacionado com o processo de

fabricação. Até porque esses novos dispositivos e circuitos visam aplicações em sistemas de comunicação de frequências muito altas: móveis pessoais (PCS) e ópticas.

As tecnologias emergentes de comunicações ópticas demandam sistemas comerciais em 10 Gbit/s, daí a necessidade de prover-se CIs **comerciais** capazes de operar nessas taxas de dados. Nos últimos anos, vários sistemas em 10 Gbit/s têm sido descritos, mas a próxima taxa de transmissão a ser implementada já deverá ser 4 vezes maior, isto é, de 40 Gbit/s. Já foram descritos na literatura, CIs para comunicações ópticas a esta nova taxa, empregando HBTs e HEMTs (estes com  $L_g$  de 0,2 a 0,1  $\mu\text{m}$  e empregando geralmente a família SCFL). Entre os CIs desenvolvidos destacam-se FFs tipo D e módulos Mux e Demux para 40 Gbit/s, empregando transistores de heteroestrutura [50].

O estado-da-arte em multiplex/demultiplexadores para comunicações ópticas mostra as seguintes realizações: um *chip* mux/demux de 16 bits operando a 4 Gbit/s, feito com MESFETs de  $L_g = 0,5\mu\text{m}$  [51]. Um demux de 4 bits em 20 Gbit/s [52], e um *chipset* mux/demux de 8 bits em 6,9 Gbit/s [53], foram demonstrados em 1991 com HBTs. No ano seguinte, um *chipset* mux/demux de 8 bits foi fabricado com MESFETs de  $L_g = 0,5\mu\text{m}$ , operando a 8 Gbit/s e usando um tipo especial de flip-flop de célula de memória [54]. Com a redução do comprimento dos MESFETs para 0,2  $\mu\text{m}$ , uma taxa de 25 Gbit/s foi alcançada para um seletor [55]. Enfim, um *chipset* LSI mux/demux com 16 canais e em 2,5 Gbit/s foi feito com MESFETs de 0,7  $\mu\text{m}$  [56].

Destacam-se também dois flip-flops: o primeiro em 20 Gbit/s usando HEMTs de 0,2  $\mu\text{m}$  em topologia DCFL [57], e um flip-flop dinâmico em 24 Gbit/s, fabricado em uma tecnologia de MESFETs com  $L_g = 0,2\mu\text{m}$ , na topologia LSCFL [58].

Atualmente, CIs de GaAs estão disponíveis principalmente na forma de matrizes de portas (*gate arrays*), com velocidades de operação superiores a 1 Gbit/s. A Vitesse comercializa uma família de *gate arrays* em GaAs, cujo maior membro é o VGFX350K, com 350 k *gates* [59].

A tecnologia de dispositivos e circuitos em Si também têm progredido muito em aumentar a frequência / velocidade de operação e baixar a potência consumida. Circuitos digitais bipolares de Si têm sido reportados na literatura, que empregam técnicas de processo antes só aplicadas em GaAs, e também processos especiais desenvolvidos em Si para a fabricação de BJTs de alto  $f_t$ , como a *Super Self-Align Technology*. Graças a isso, circuitos digitais de Si já começam a operar em taxas de bits antes exclusivas de CIs de GaAs, chegando até mesmo em 40 Gbit/s. Alguns exemplos notáveis incluem: um multiplex de 4 canais em ECL, desenvolvido em 1979 em um processo bipolar de alta frequência de Si, alcança taxas de 2,8 Gbits/s [60]; um multiplex para 50 Gbit/s e um divisor de frequência para 30 GHz foram implementados com BJTs de Si [61], em um processo de fabricação especial. Os processos especiais de Si que viabilizam essas taxas ainda não estão em caráter comercial. Os sistemas de comunicação por fibras ópticas são a principal aplicação a que se destinam esses circuitos.

A redução das geometrias dos dispositivos ao nível submicrométrico, além de elevar a velocidade de operação, aumenta também a densidade de integração. O problema é que os projetos próximos do limite da resolução do processo atingem um nível de rendimento de fabricação (*yield*) inaceitável, exigindo processos complexos e de alto custo.

### 2.3. Características das Famílias Lógicas E/D em GaAs.

As várias famílias lógicas existentes na tecnologia de GaAs correspondem, cada uma, a uma configuração diferente das portas básicas. A escolha de uma família lógica para o desenvolvimento de um CI deve ser feita de acordo com a aplicação desejada para o circuito, sua complexidade, possíveis exigências de compatibilidade com outras tecnologias, escala de integração e valor do compromisso D.P. Os diferentes compromissos entre velocidade, potência e complexidade impostos pelos sistemas implicam que diferentes topologias (famílias) lógicas devam ser empregadas, de modo a atender aos requisitos conflitantes de alta velocidade bem como alta complexidade. As diferenças entre as famílias lógicas em GaAs e em Si não estão propriamente nas configurações dos circuitos básicos; as diferenças estão, principalmente, **na operação** de configurações semelhantes. Nesta seção, são descritas as duas principais famílias lógicas estáticas E/D em GaAs [9], DCFL e SCFL.

Uma porta lógica estática em GaAs é sempre constituída por um bloco lógico com FI entradas (*fan-in*), conectadas a uma fonte de alimentação através de uma carga. Esse bloco é basicamente um amplificador de tensão. Um acionador (*buffer*) pode ser conectado em seguida para fornecer maior corrente de saída, e acionar FO portas idênticas conectadas em paralelo na saída, ampliando a capacidade de *fan-out* da porta. Em contraste às portas estáticas, existem também as portas lógicas dinâmicas em GaAs, que empregam o FET como transistor de passagem [4]; as lógicas dinâmicas não serão abordadas neste trabalho.

Famílias lógicas empregando EFETs e DFETs no mesmo circuito tornaram-se realizáveis a partir de 1977 [26], graças ao avanço dos processos de fabricação. Atualmente são as famílias lógicas mais úteis [5]. Como geralmente os DFETs são usados como cargas ativas e os EFETs como chaves de entrada, resulta que tanto  $V_{GS}$  quanto  $V_{DS}$  têm sempre o mesmo sinal, não há necessidade de deslocadores de nível, e uma única fonte de alimentação ( $V_{DD}$  ou  $V_{SS}$ ) é geralmente suficiente. As tensões de limiar típicas para MESFETs de enriquecimento ( $V_{LE}$ ) e de depleção ( $V_{LD}$ ), estão em torno de 0,1 V a 0,3 V, e -0,4 V a -1 V, respectivamente.

#### 2.3.1. Família Lógica DCFL (*Direct-Coupled FET Logic*).

É uma família na qual a saída de uma porta lógica é acoplada diretamente à entrada da porta seguinte, de onde provém o seu nome. O inversor DCFL é composto de apenas dois transistores, um EFET como chave (comando) e um DFET como carga (figura 2.1(b)), em uma topologia similar a do inversor NMOS em Si, que também é uma lógica de acoplamento direto. Apesar disso, há diferenças significativas de funcionamento entre essas duas famílias lógicas. A mais marcante dessas diferenças é que o nível  $V_{OH}$  em DCFL não é determinado pela fonte  $V_{DD}$ , como nas portas em tecnologia MOS; ele fica “grampeado” em torno de 0,7 V no terminal de porta do transistor de entrada do estágio seguinte, por causa da polarização direta do diodo de porta-fonte deste transistor. Portanto, a excursão lógica em DCFL vale aproximadamente  $V_{pp} = 0,7 - V_{OL}$  V. Em NMOS, a excursão lógica vale  $V_{DD} - V_{OL}$ .

As primeiras portas lógicas em DCFL eram projetadas com transistores para chaveamento e resistores de carga, como na figura 2.1(a).

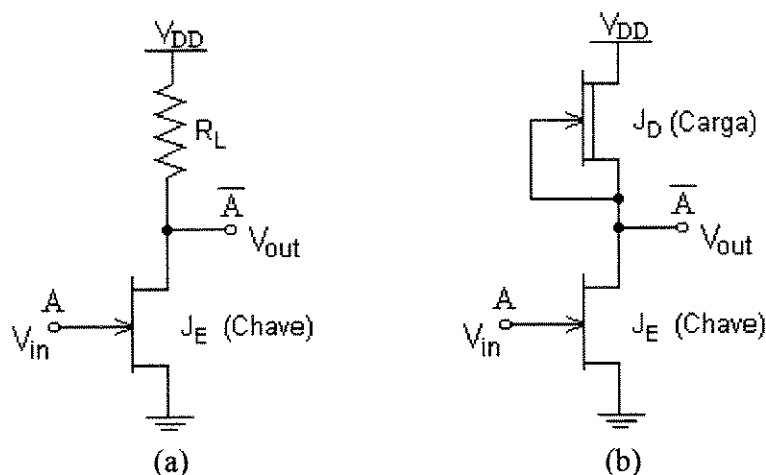


Figura 2.1 – Inversores DCFL com carga (a) resistiva e (b) ativa, realizada com um DFET.

O inversor DCFL com carga resistiva da figura 2.1(a) pode também ser tratado como um amplificador de fonte comum, com um ganho de tensão na região de transição dado por  $A_v = -g_m R_L$ . É importante que uma porta lógica possua um alto  $A_v$ , porque a sua margem de ruído (NM) é proporcional ao seu ganho de tensão na região de transição [11]. As cargas resistivas devem ter uma alta resistência para se obter um alto  $A_v$ , mas apenas aumentar  $R_L$  não é satisfatório porque a região de transição passará a ocorrer em um nível de  $I_D$  muito pequeno, onde  $g_m$  é baixa; para compensar esse efeito, seria preciso uma alta tensão de alimentação. Portanto, uma carga ativa é empregada para DCFL de um modo geral, consistindo de um DFET com  $V_{GS} = 0$ . Se  $V_{DD}$  for alta o bastante para que o DFET de carga permaneça na saturação durante a região de transição entre estados lógicos, o ganho de tensão realizado será bem mais alto. Um DFET como carga ativa leva a um  $A_v$  muito mais alto, para o mesmo valor de  $V_{DD}$  (maior excursão de saída,  $v_{ds}$ , para a mesma excursão de entrada,  $v_{gs}$ ) porque fornece uma corrente maior [9], carregando mais rapidamente a capacitância de entrada do estágio seguinte. Além disso, um transistor ocupa menos espaço que um resistor. O novo  $A_v$  é igual a

$$A_v = - \frac{g_m}{g_{ds,E} + g_{ds,D}} \quad (2.1)$$

que sugere que  $g_{ds}$  deveria ser feita pequena. Mas como os MESFETs devem ser feitos com pequeno  $L_g$  por uma série de outras vantagens, e  $L_g \downarrow \Rightarrow g_{ds} \uparrow$ , então os EFETs devem ser projetados para terem alta  $g_m$ , isto é, alta velocidade com uma pequena excursão de  $V_{GS}$ .

Conectar chaves EFET em paralelo resulta em uma porta NOR, onde o terminal de *gate* de cada EFET é uma entrada da porta, mostrada na figura 2.2. O uso de portas NAND em DCFL é desaconselhado por causa da sua topologia com o empilhamento de EFETs em série, cujo efeito principal é o aumento do valor de  $V_{OL}$ , reduzindo a NM da saída e levando a erros lógicos e mesmo a um menor rendimento do processo ([2], [11]). Isto pode ser constatado na porta NAND DCFL figura 2.3.

Frequentemente precisa-se de portas com um maior *fan-in*, e é então que surge a grande vantagem das portas NOR: o *fan-in* pode ser aumentado simplesmente acrescentando-

se mais EFETs em paralelo. Com as portas NAND, seria preciso acrescentar mais chaves EFET em série, o que reduziria drasticamente a  $V_{pp}$  na saída, reduzindo portanto as NMIs.

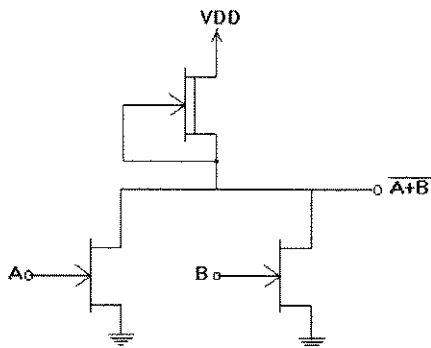


Figura 2.2 - Uma porta NOR de duas entradas na família DCFL.

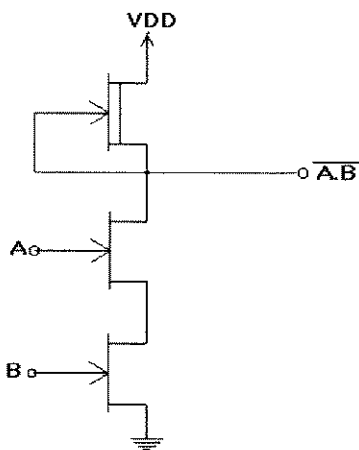


Figura 2.3 - Uma porta NAND de duas entradas na família DCFL.

Dentro de um CI, inversores e demais portas lógicas acionam as entradas de uma ou mais portas lógicas (*fan-out*) de carga. O capacitor e o diodo na figura 2.4 representam o circuito equivalente de entrada do EFET da porta de carga.

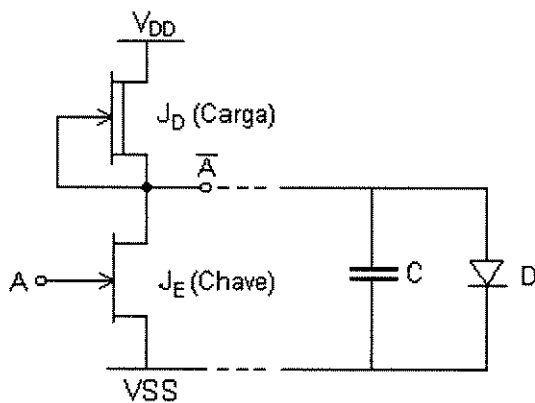


Figura 2.4 – Inversor DCFL acionando o EFET de entrada de outra porta lógica.

DCFL foi a primeira topologia desenvolvida com EFETs, e é considerada por muitos como sendo a família em tecnologia de GaAs mais apta para circuitos em escalas LSI e VLSI, porque é a mais simples dentre as famílias lógicas que utilizam MESFETs.



Para uma tecnologia ser realizável em escala VLSI, ela deve atender simultaneamente a uma série de requisitos indispensáveis, além, é claro, de um baixo atraso de propagação por porta lógica, viabilizando um sistema de alta velocidade. Uma família lógica apta para circuitos de complexidade VLSI deve possuir principalmente um baixo valor de D.P, da ordem de poucos fJ. Na busca por CIs VLSI e de alto desempenho em GaAs, a dissipação de potência das famílias lógicas somente de depleção é um obstáculo principal. As portas DCFL demonstram um potencial para baixa dissipação de potência e *layout* compacto, que são requisitos essenciais para todo candidato viável a aplicações em VLSI.

As principais características de DCFL são:

- \* Sacrifica um pouco da alta velocidade inerente aos circuitos em GaAs em favor de um baixo consumo de potência, mas ainda mantém um baixo tempo de atraso;
- \* Suas margens de ruído são muito pequenas, em torno de 150 a 200 mV, devido à excursão limitada do sinal de tensão nas saídas das portas ( $V_{pp}$  é pequeno); esse é o principal problema da lógica DCFL, e é o que a torna sensível ao ruído;
- \* Emprega MESFETs de enriquecimento e de depleção para a construção de todas as suas portas lógicas, de onde decorre sua simplicidade;
- \* Dentre todas as famílias digitais em GaAs, DCFL é a que utiliza a menor quantidade de componentes por porta lógica, resultando nas topologias de circuito mais simples de todas, além da área da porta ser reduzida; esta é talvez a sua principal vantagem;
- \* Sua simplicidade lhe confere potencial para pequena dissipação de potência, tipicamente de centenas de  $\mu W$  por porta lógica;
- \* O uso de um EFET como chave permite que  $V_{IL}$  e  $V_{OL}$  sejam positivos, porque a  $V_t$  de um EFET é positiva, e assim a lógica DCFL pode operar com uma única fonte de alimentação, a qual pode ser positiva ( $V_{DD}$  no dreno do DFET) ou negativa ( $V_{SS}$  ou  $V_{TT}$  na fonte do EFET), como na figura 2.4;
- \* Viabiliza circuitos em complexidade VLSI;
- \* Permite maiores níveis de integração ( $> 100k$  portas equivalentes), e maiores velocidades de operação que ECL ou BiCMOS;
- \* Baixa capacidade de acionamento de cargas: é muito sensível ao *fan-out*; o produto D.P de DCFL aumenta muito para um *fan-out* grande, ou na presença de carregamento capacitivo elevado;
- \* As portas DCFL são estruturalmente semelhantes às da lógica NMOS;
- \* As portas lógicas DCFL realizam um produto atraso-potência muito menor que o de qualquer outra família lógica em GaAs, tipicamente na faixa de 1 fJ a 50 fJ;
- \* Dados dois circuitos com a mesma função e complexidade, mas sendo um em CMOS e outro em DCFL, e operando em uma mesma frequência acima de um certo valor, pode-se afirmar que o CI de GaAs dissipa menor potência [2].

\* A fabricação de circuitos na lógica DCFL exige um controle rigoroso do processo e a realização precisa dos valores de  $V_t$  dos MESFETs, com alta uniformidade nos valores de  $V_{tE}$  e  $V_{tD}$ , em comparação com as demais lógicas; isso porque  $V_{tE}$  e  $V_{tD}$  são baixas, e variam inevitavelmente no processo de fabricação. Essa exigência torna-se ainda maior diante da excursão limitada de  $V_{GS}$ .

A tensão de alimentação  $V_{DD}$  tipicamente usada está em torno de 1 V a 2 V; quando se necessita trabalhar com baixo consumo, usa-se 1 V; porém, a máxima frequência de operação depende do valor de  $V_{DD}$ . Osciladores em anel em DCFL apresentaram um atraso de propagação muito baixo, em torno de 10 ps por transistor para uma carga mínima.

A corrente estática fluindo em uma porta DCFL pode geralmente ser considerada constante, porque a corrente no FET de carga (DFET) irá fluir ou pelo(s) FET(s) de chaveamento (EFETs) da mesma porta lógica (no caso de uma saída em  $V_{OL}$ ), ou através do diodo porta-fonte do FET de entrada da porta lógica seguinte (no caso de uma saída em  $V_{OH}$ ). Para  $V_{DD}$  maior do que aproximadamente 1,3 V, o DFET de carga vai estar saturado, e a corrente saindo da fonte  $V_{DD}$  será quase constante.

Para otimizar o circuito DCFL para altas velocidades, deve ser considerado o compromisso entre NM e velocidade. De modo a reduzir-se a NM mínima necessária, que é exigida para acomodar os espalhamentos do processo e as variações de temperatura, e portanto otimizar a velocidade, pode-se incluir um bloco de circuito de controle capaz de compensar por tais espalhamentos e variações.

A análise gráfica das portas DCFL é feita neste trabalho, e também em [11] e [66], enquanto que uma análise analítica bastante detalhada é desenvolvida em [6] e [41]. Procedimentos de projeto por computador são explicados em [9] e [67], e também serão desenvolvidos no Capítulo 3 deste trabalho. As vantagens e desvantagens de DCFL, em comparação com as lógicas BFL e SDFL, são discutidas em [3]; a referência [42] compara o desempenho de inversores DCFL com o de inversores LBFL. Um projeto VLSI em DCFL é descrito em [43].

### 2.3.2. Família Lógica SCFL (*Source-Coupled FET Logic*).

A família SCFL foi criada em 1984 por S. Katsu et al. [68]. Esta lógica segue uma filosofia de projeto de “Lógica em Modo de Corrente”, ou CML, na qual também se inclui a família ECL de BJTs de Si, daí a semelhança topológica entre as portas SCFL e ECL. O inversor SCFL tem como núcleo um par diferencial, com seguidores de fonte nas saídas, como mostrado na figura 2.5, podendo trabalhar com sinais em modo diferencial ou simples. A estrutura de par diferencial propicia a rejeição a sinais de modo comum (ruído). Os níveis de tensão das famílias ECL e SCFL são geralmente feitos compatíveis. SCFL é considerada a topologia de circuito de GaAs mais rápida existente hoje [4], alcançando frequências de operação de mais de 3 GHz e taxas de comutação (*toggle*) de mais de 5 GHz.

Na figura 2.5, T1 e T2 são os FETs do par diferencial, T3 e T4 são suas cargas ativas, e T5 é a fonte de corrente do par. Em algumas implementações, T3 e T4 são implementados com resistores ([4], [11]). Os sinais nas entradas  $V_{in1}$  e  $V_{in2}$  deslocam a corrente gerada por T5 entre os dois ramos do par, causando queda em uma ou outra carga. O deslocamento da

corrente (chaveamento) passa a ocorrer devido à diferença entre as duas entradas do par, e não depende mais da diferença entre uma entrada e a  $V_t$  do respectivo FET, como acontece em DCFL, aumentando muito a imunidade do circuito a variações de  $V_t$ .

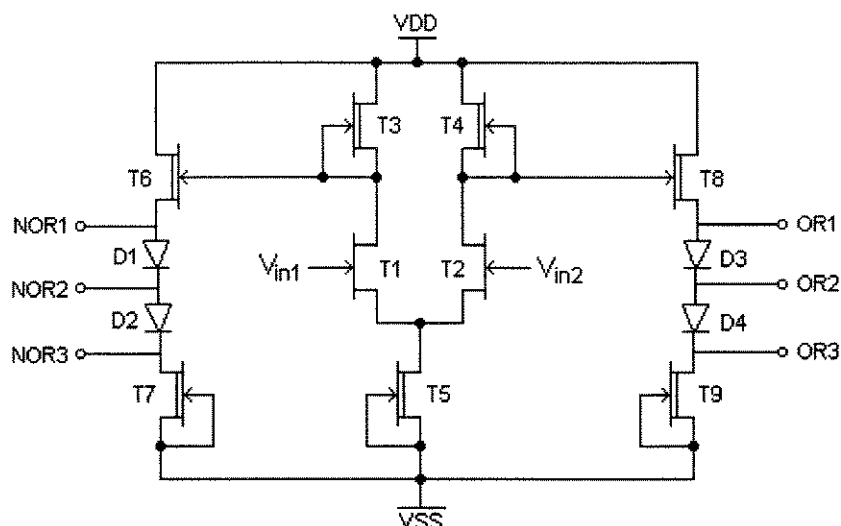


Figura 2.5 – Porta básica SCFL E/D.

Os diodos D1-4 podem ser necessários para ajustar níveis lógicos, de modo que uma porta acione outras. Na figura acima, também poderia não haver nenhum diodo, dependendo das características do projeto. As fontes T7 e T9 servem para polarizar os diodos e seguidores de fonte. Pode ser usada somente a fonte  $V_{DD}$ , tipicamente de 5 V, e  $V_{SS} = 0$  V [9]; no entanto, se for desejada a compatibilidade do circuito com ECL,  $V_{DD} = 0$  V e  $V_{SS} = -5,2$  V são adotados [11].

Os seguidores de fonte T6 e T8 aumentam a capacidade da porta de acionar outras portas (*fan-out*) ou capacitâncias de carga, por causa de sua baixa impedância de saída e alto ganho de corrente. Além disso, a existência de saídas complementares possibilita a implementação de funções digitais complexas, como flip-flops, com o emprego de poucas portas. Mas essas vantagens custam um aumento da complexidade, e consequentemente da potência consumida, por porta lógica. As portas SCFL trabalham com um número muito maior de transistores por porta lógica do que DCFL, assim o uso de SCFL só é recomendado para circuitos pouco complexos em escala MSI, e com carregamento pesado na saída (alto *fan-out* ou altas capacitâncias de carga).

Em SCFL, todos os FETs das portas lógicas chaveiam sempre entre corte e saturação, enquanto o chaveamento DCFL ocorre entre corte e região linear. Uma vez que na saturação a capacitância  $C_{GD}$  é menor e a  $g_m$  é maior, resulta uma diminuição do tempo de atraso na porta.

Como as portas SCFL têm uma topologia mais complexa, o seu *layout* também é bem mais elaborado. Além da porta lógica propriamente dita, ainda é preciso projetar-se os circuitos de polarização que vão gerar as tensões de controle das fontes de corrente no estágio diferencial e nos estágios seguidores de fonte de saída. Cada porta lógica demanda dois desses sinais de controle. E esses circuitos de polarização precisam ser estabilizados para variações de temperatura e de processo.

A topologia diferencial SCFL pode ser construída tanto em processos somente de depleção quanto em processos E/D. No primeiro caso, são usados DFETs, diodos

deslocadores de nível e resistores para compor a sua porta básica, cuja operação, similar a dos circuitos ECL, é descrita em [13].

SCFL pode ser combinada diretamente com DCFL no mesmo chip, pois as portas de ambas as famílias lógicas são implementadas com os mesmos tipos de FETs [69]. Um exemplo é a família de *gate arrays* SCFX, da Vitesse, que possui um núcleo DCFL em torno do qual são arranjos *sub-arrays* DCFL ou SCFL, e pode operar em taxas de até 3 GHz.

Os procedimentos de projeto para portas SCFL são explicados, de forma bastante pragmática, em [4], [9], [11] e [66]. Várias implementações de CIs SCFL são descritas na literatura: [4] descreve um chipset mux/demux de 4 bits, operando a 4 Gbit/s, e [70] descreve em detalhes um projeto de CI demultiplexador de 16 canais em SCFL, a 2,5 Gbit/s. Existe uma versão de baixa potência da lógica SCFL chamada de LSCFL, que utiliza somente EFETs, diodos deslocadores de nível e resistores. Seu modo de funcionamento é explicado em [13], e o projeto de um CI LSI em LSCFL é descrito em [71].

### 2.3.3. Comparação Entre as Famílias Lógicas DCFL e SCFL.

Os requisitos para CIs de velocidade muito alta (VHSICs), porém de média complexidade, são diferentes dos requisitos para CIs VLSI de alto desempenho, então diferentes topologias de circuitos lógicos devem ser empregadas em cada caso. Algumas somente são adequadas para CIs MSI, mas permitem uma velocidade muito alta de operação; é o caso da família SCFL, que realiza CIs otimizados para alta velocidade, em detrimento da economia de área ocupada e de potência consumida. Por outro lado, CIs digitais em alta escala de integração exigem, antes de tudo, pequena excursão de tensão e operação com baixa potência, mais típicos da família DCFL, que entretanto não é tão veloz.

Atualmente, DCFL é a família lógica mais popular para circuitos de alta complexidade e baixa potência (LSI/VLSI), enquanto SCFL é a melhor opção para aplicações de alta velocidade. A escolha de uma ou outra família lógica depende de qual seja a meta prioritária de projeto. Por exemplo, se o objetivo principal para um circuito for a operação em alta velocidade, emprega-se SCFL. Dependendo da aplicação, outras propriedades podem se tornar importantes (por exemplo, dissipação de potência). Há casos em que os circuitos dissipam tanta potência que o arrefecimento a ar não é suficiente, e um resfriamento a líquido precisa ser considerado. Mas a dissipação não é problema para CIs em uma escala relativamente pequena de integração, que dissipam apenas uns poucos watts de potência, como os descritos em [4] e [70], e assim eles são geralmente realizados em SCFL. Comparações entre diversas famílias lógicas de GaAs são feitas em [5], [13] e [70].

## 2.4. Conclusão.

As tecnologias de MESFETs em GaAs devem manter sua posição como alternativa adequada ao Si, para aplicações onde a velocidade é a preocupação principal. Para GaAs conseguir dominar em aplicações VLSI é mais difícil, pois esta área é dominada por tecnologias em Si bem estabelecidas. A principal vantagem do GaAs comparado ao Si é o seu potencial para maiores velocidades; mas, enquanto o produto D.P de DCFL é melhor que o de

ECL, ele não se compara ao de tecnologias de baixa potência em Si, como CMOS. Consequentemente, a dissipação de potência em GaAs ainda deve ser reduzida bastante para que ele se torne uma alternativa a CMOS em aplicações de potência muito baixa (*ultra low-power*). Porém, o uso de lógicas dinâmicas em GaAs é uma forma de reduzir a dissipação de potência dos CIs.

A família lógica em GaAs mais sensível ao ruído, DCFL, é também a mais usada atualmente, por causa de sua simplicidade, grau de compactação, e baixa dissipação de potência. Para uma investigação mais aprofundada de suas características, foi proposto o projeto de CI contador descrito neste trabalho.

## CAPÍTULO 3.

### Projeto de Portas Lógicas Básicas em DCFL.

#### 3.1. Conceitos de Projeto de CIs Digitais em GaAs, com Ênfase em DCFL.

Primeiramente, toda família lógica deve ser definida em termos de especificações consistentes de níveis de entrada e de saída, porque todos os CIs de uma mesma família precisam ser compatíveis entre si, e cada CI, individualmente, deve ter níveis lógicos de saída compatíveis com os níveis lógicos requeridos em suas entradas. A figura 3.1 apresenta um gráfico ao qual todo circuito digital precisa obedecer, para que as suas entradas sejam compatíveis com as suas próprias saídas. Esse gráfico tem que ser sempre observado em projetos de CIs digitais para qualquer família lógica.

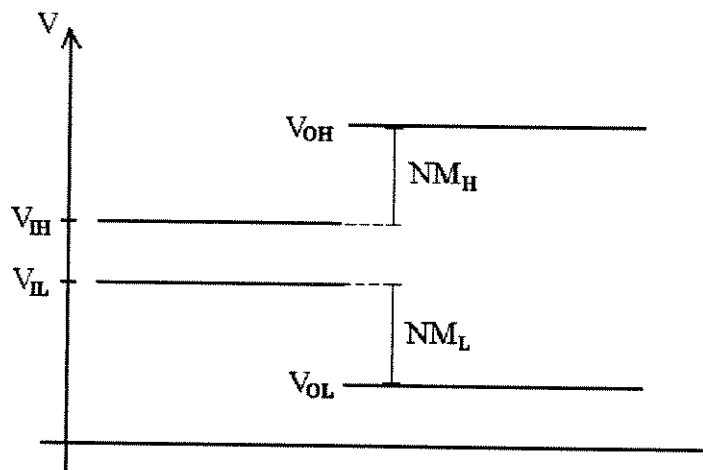


Figura 3.1 – Relações obrigatórias entre os níveis lógicos.  $NM_H$  e  $NM_L$  são as tensões máximas aceitáveis de ruído, para uma operação correta do circuito.

##### 3.1.1. Margens de Ruído (NMs).

As margens de ruído são parâmetros de um circuito digital que quantificam a sua capacidade de operar corretamente em um ambiente ruidoso; elas medem o quanto o circuito é tolerante ao ruído, e indicam a característica regenerativa dos circuitos digitais. As margens de ruído correspondem às máximas amplitudes de ruído que podem ser sobrepostas a um nível lógico de tensão ou corrente sem causar a mudança do circuito para o outro estado lógico. Dessa forma, existem as NMs alta e baixa, referentes às tolerâncias dos níveis lógicos alto e baixo ao ruído, respectivamente. Definem-se também as NMs estática e dinâmica, conforme o tipo de ruído presente seja DC ou pulsado, respectivamente, e que possuem valores diferentes. As NMs dinâmicas são maiores do que as estáticas porque, durante pulsos estreitos, maiores amplitudes de ruído podem ser toleradas [73]. Neste trabalho, serão consideradas somente as NMs para tensões de ruído DC.

As NMs de uma porta lógica são uma função dos valores de seus níveis lógicos estáveis, que por sua vez são função direta do projeto dessa porta lógica. Porém, existem

várias definições para as NMs e maneiras de estimá-las partindo da característica de transferência da porta, e que não são equivalentes entre si [11]. As margens  $NM_H = V_{OH} - V_{IH}$  e  $NM_L = V_{IL} - V_{OL}$  mostradas na figura 3.1 são superestimadas [11], mas são bastante intuitivas e úteis para fins de comparação entre alternativas de projeto. Os níveis  $V_{IL}$  e  $V_{IH}$  são determinados nos pontos onde a inclinação da curva de transferência de uma porta lógica é igual a  $-1$  [72]. Expressões analíticas para  $V_{IL}$  e  $V_{IH}$  são deduzidas em [6] e [66].

As NMs de um CI projetado em um método *full-custom* são fortemente influenciadas pelas razões entre as dimensões dos transistores que foram usadas nas portas lógicas.

Para as análises dos circuitos lógicos de MESFETs deve ser usada a função de transferência de tensão, porque estes circuitos possuem alta impedância de entrada e baixa impedância de saída ([11], [73]). Se essa condição é válida, a tensão total na entrada de uma porta é o resultado da soma da tensão de saída da porta anterior com a tensão do ruído, e este assume a forma de uma fonte de tensão em série com a interconexão entre as portas lógicas. Para essa configuração, a NM de tensão estática de pior caso pode ser encontrada graficamente com o “método do máximo quadrado” aplicado à curva de transferência de tensão de uma porta lógica. Esse método corresponde à definição mais pessimista de NM.

A determinação das NMs de uma porta lógica pelo “método do máximo quadrado” deve ser feita ligando-se essa porta a outra idêntica a ela, em série, e realimentando a saída da segunda porta para a entrada da primeira, formando assim um *latch*. Este arranjo é mostrado na figura 3.2 para o caso de dois inversores idênticos, Inv1 e Inv2, um acionando o outro. Tensões de ruído  $V_N$  iguais e opostas são inseridas na entrada de cada inversor, levando ao pior caso de NM.

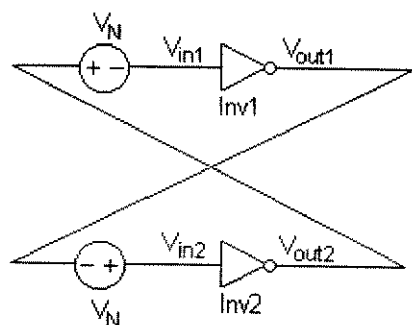


Figura 3.2 – *Latch* formado por um laço de inversores iguais, para a determinação das margens de ruído de pior caso [73].

Como a entrada de uma porta é igual à saída da outra, as curvas de transferência de ambas podem ser desenhadas sobrepostas no mesmo gráfico, da seguinte maneira: desenha-se a curva de transferência de Inv1 e, como  $Inv2 = Inv1$ , a curva de Inv2 pode ser obtida girando-se a curva de Inv1 e os eixos por  $180^\circ$ , em torno da reta  $V_{out} = V_{in}$ . A tensão de ruído  $V_N = V_{MS}$  necessária para inverter o estado do *latch* é numericamente igual à largura do maior quadrado que puder ser desenhado dentro dos “laços” formados pelas características de transferência. Isso pode ser compreendido pela sequência de desenhos da figura 3.3: como as fontes de ruído  $V_N$  são iguais e opostas, tanto  $V_{in1}$  quanto  $V_{in2}$  tendem ao ponto de equilíbrio instável B, a partir de qualquer dos pontos de equilíbrio estável A ou C. Nesta figura estão indicadas as curvas de Inv1 e Inv2.

O ponto A aproxima-se de B a partir de duas direções, formando os lados de um quadrado, de onde vem o nome “máximo quadrado” para esta NM. A figura 3.3(a) é a situação estável sem o efeito do ruído, (b) mostra os efeitos do ruído sobre as curvas de transferência e (c) é a situação em que a NM baixa foi excedida. Neste tipo de circuito biestável, uma vez que a sua NM for excedida, a saída não retornará mais ao seu estado original, mesmo que a tensão de ruído seja removida; então a única condição estável disponível passa a ser o estado oposto. As tensões de ruído  $V_N$  na figura 3.2 mudarão o estado do *latch* quando forem iguais a  $V_{MS}$ , que é a NM pelo método do máximo quadrado. Na situação de portas iguais,  $V_{MS,H} = V_{MS,L}$ . Mas as NMs alta e baixa não são necessariamente iguais se as características de transferência não forem simétricas. O mesmo método pode ser aplicado a casos em que as portas ou inversores não sejam idênticos ou não estejam igualmente carregados.

A maior  $V_{MS}$  será obtida na situação em que as larguras dos laços superior e inferior sejam iguais. Se  $V_{MS,H} = V_{MS,L}$ , resulta a melhor estabilidade do circuito, assim esta deve ser a primeira das metas de projeto.

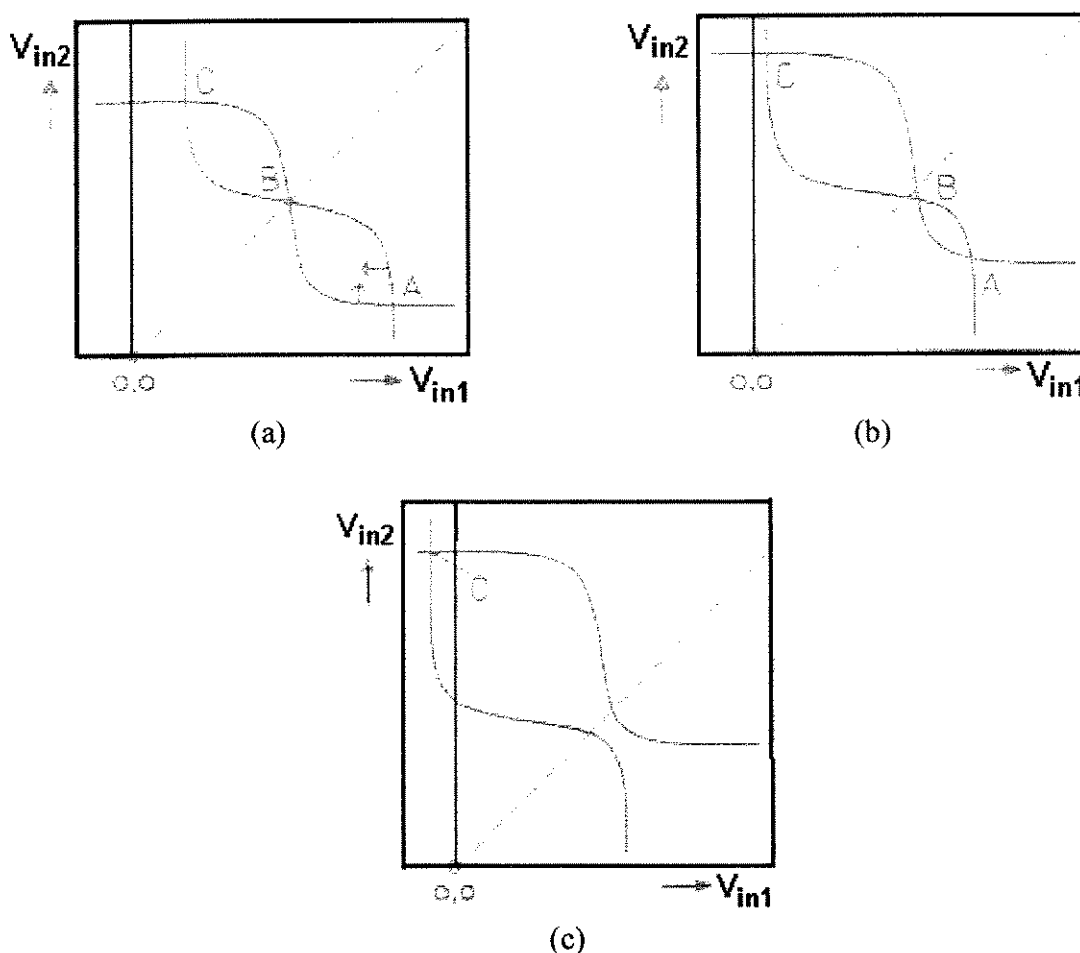


Figura 3.3 - As tensões de ruído iguais e opostas deslocam simultaneamente as curvas de transferência dos inversores. Um exemplo completo é dado na figura 3.18.

A velocidade e a robustez de um circuito (tolerância a variações do processo, da temperatura, da tensão de alimentação e ao ruído) são intercambiáveis. Aumentar a NM implica em maiores capacitâncias de entrada das portas lógicas, o que diminui a velocidade de



chaveamento, enquanto que um circuito com NM pequena imporá menor atraso aos sinais. Portanto deve ser feito um *trade-off* entre NM e atraso de propagação. Na grande maioria dos projetos, a NM deve receber maior prioridade do que a velocidade, de modo a garantir uma operação DC confiável e aumentar o rendimento da fabricação.

Qualquer que seja o método de cálculo das NMs de uma porta lógica, elas são sempre diretamente proporcionais à excursão lógica pico-a-pico na saída da porta,  $V_{pp} = V_{OH} - V_{OL}$ . E o nível  $V_{OL}$  deve ser tão pequeno quanto possível, para minimizar a corrente de sub-limiar do EFET da porta seguinte. Assim, enquanto um cálculo preciso das NMs é trabalhoso,  $V_{pp}$  serve como parâmetro para uma avaliação segura das mesmas, para fins de comparação, podendo ser obtido facilmente de uma simulação HSPICE da porta ou circuito lógicos. O HSPICE possui poderosos recursos de análise ([74] e [75]), que foram empregados para medir-se as características das formas de onda dos sinais, em todas as simulações. Os comandos **.measure** imprimem várias especificações elétricas de um circuito, permitindo medir-se os níveis lógicos,  $V_{pp}$ , os tempos de transição, os tempos de atraso de propagação e outras variáveis definidas pelo usuário.

Para a avaliação e comparação das NMs de diferentes alternativas de dimensionamento de portas lógicas, foram empregados os seguintes comandos de medida nas simulações:

1 - Para medir  $V_{OL}$ :

```
.measure tran Vol avg v(<output node>) from=<time1> to=<time2>
```

2 - Para medir  $V_{OH}$ :

```
.measure tran Voh avg v(<output node>) from=<time1> to=<time2>
```

3 - Para medir  $V_{pp}$ :

```
.measure tran vpp param='Voh-Vol'
```

Os comandos **.measure** são explicados resumidamente na página 2-13 de [12], e com muito mais detalhes em [75].

### 3.1.2. Dissipação de Potência.

A potência total consumida por um circuito digital é o resultado da soma de duas componentes: potência estática e potência dinâmica. A dissipação de potência estática,  $P_{D,st}$ , é a dissipação de potência DC do circuito digital, que ocorre enquanto o circuito não está chaveando entre estados lógicos; isto é, enquanto não ocorrem transições em nenhum nó interno do circuito. A família DCFL é uma lógica estática, suas portas consomem corrente da fonte de alimentação **durante** os estados lógicos alto e baixo.

Para portas DCFL, como o inversor da figura 2.1(b), a corrente que sai da fonte  $V_{DD}$  é solicitada por uma carga ativa  $J_D$ ; se  $V_{DD}$  é alta o bastante para manter  $J_D$  na saturação, então a corrente é aproximadamente constante e independente do estado lógico do inversor, pois  $J_D$  atuará como fonte de corrente. Entretanto, nas simulações de inversores e portas DCFL com  $V_{DD} = 2\text{ V}$ , observou-se que a  $P_{D,st}$  depende do estado lógico das portas: quando o nível lógico de saída é baixo,  $P_{D,st}$  é maior do que quando ele é alto.  $P_{D,st}$  pode ser obtida da simulação do circuito, pela plotagem do gráfico de  $P_{D,st}$  *versus*  $V_{in}$ , que é mostrado na seção 3.2.

A dissipação de potência dinâmica,  $P_{D,dyn}$ , é a potência dissipada durante as transições lógicas das portas do circuito. É devida principalmente às correntes de carga e descarga das capacitâncias presentes nas saídas de cada porta lógica, e também é consumida da fonte de alimentação do circuito, porque surge um percurso momentâneo para a corrente, da fonte para o terra, durante as transições lógicas. A potência  $P_{D,dyn}$  de uma porta só é significativa se a sua capacitância de carga e a frequência de operação forem altas, isto é, somente onde longas interconexões e/ou altos *fan-outs* são acionados. Para CIs de GaAs,  $P_{D,dyn}$  costuma ser uma pequena fração da  $P_{D,st}$  [11], podendo ser calculada pela expressão:

$$P_{D,dyn} = f.C_L.V_{pp}^2 \tag{3.1}$$

### 3.1.3. Tempos de Transição e Atrasos de Propagação.

Os tempos de subida e de descida de um sinal de entrada em um circuito digital são simbolizados por  $t_r$  e  $t_f$ , enquanto os tempos de subida e descida de um sinal de saída são simbolizados por  $t_{LH}$  e  $t_{HL}$ , respectivamente. As definições de  $t_r$  e  $t_{LH}$  são: os intervalos de tempo que vão de 10% até 90% (ou no sentido contrário para  $t_f$  e  $t_{HL}$ ) do valor pico-a-pico do sinal.

Os atrasos de propagação entre a saída e uma entrada especificada são designados por  $t_{PHL}$  e  $t_{PLH}$ , onde  $t_{PHL}$  refere-se à transição de descida da saída e  $t_{PLH}$  refere-se à transição de subida da saída. Os pontos de referência para suas medidas são os pontos de 50% nas bordas das tensões de entrada e de saída. A entrada pode ser, no caso de um flip-flop, o *Preset*, o *Clear*, ou o *Clock*, mas a definição é sempre a mesma. Os tempos de transição e atrasos de propagação de um sinal em relação a outro sinal que o origina são definidos na figura 3.4.

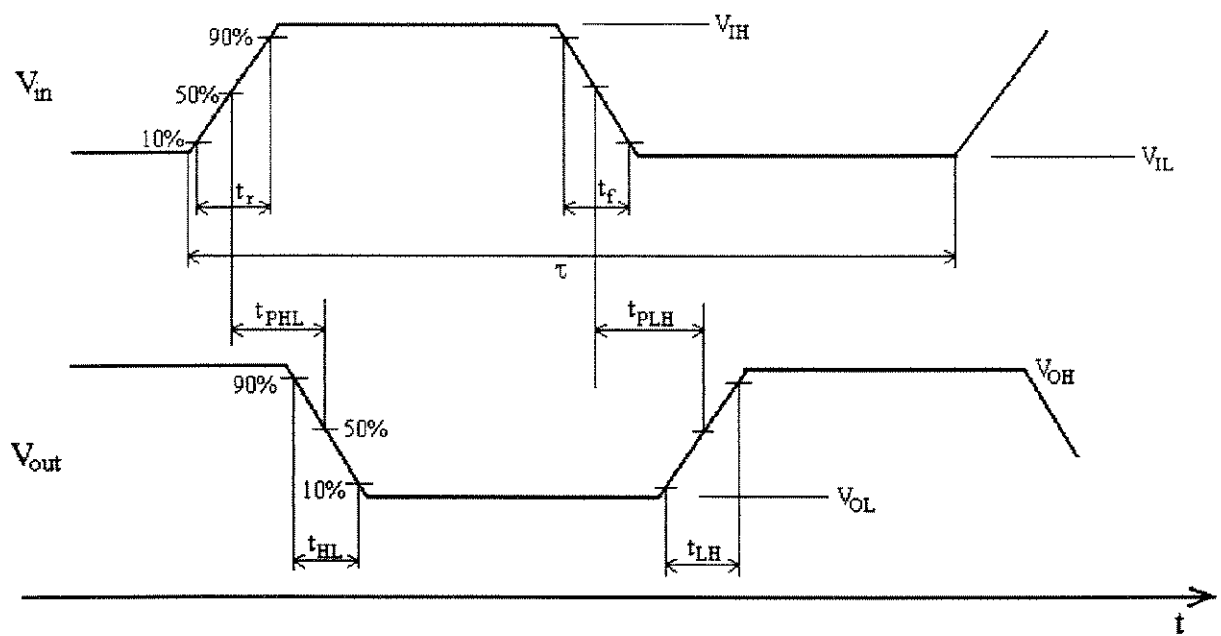


Figura 3.4 – Definições padrão de tempos de transição e atrasos de propagação.

De um modo geral, os atrasos  $t_{PLH}$  e  $t_{PHL}$  serão diferentes em um projeto; o que se pode fazer é procurar minimizar essa diferença através da escolha adequada das dimensões dos

FETs. O atraso de propagação global de uma porta ou circuito lógico é a média aritmética de  $t_{PLH}$  e  $t_{PHL}$ , simbolizado por  $t_{PD}$ .

Essas definições são muito convenientes para a caracterização de formas de onda, tanto simuladas quanto medidas em um osciloscópio. A partir das expressões para  $V_{OL}$ ,  $V_{OH}$  e  $V_{pp}$  da seção 3.1.1, os demais níveis de amplitude, tempos e atrasos da figura 3.4 são obtidos pelas seguintes expressões no HSPICE:

1- Para medir o ponto de 10% em uma onda:

```
.measure tran ten param='0.1*vpp+vol'
```

2 - Para medir o ponto de 90% em uma onda:

```
.measure tran ninety param='0.9*vpp+vol'
```

3 - Para medir o ponto de 50% em uma onda, para cálculo dos atrasos de propagação:

```
.measure tran fifty param='0.5*vpp+vol'
```

4 – Para medir  $t_{HL}$ :

```
.measure tran thl trig v(<output node>) val=ninety fall=<number>
+ targ v(<output node>) val=ten fall=<number>
```

5 – Para medir  $t_{LH}$ :

```
.measure tran tlh trig v(<output node>) val=ten rise=<number>
+ targ v(<output node>) val=ninety rise=<number>
```

6 – Para medir  $t_{PHL}$  quando a tensão no nó 1 desce causando a descida da tensão no nó 2:

```
.measure tran tphl trig v(1) val=fifty1 fall=<number1>
+ targ v(2) val=fifty fall=<number2>
```

7 – Para medir  $t_{PLH}$  quando a tensão no nó 1 desce causando a subida da tensão no nó 2:

```
.measure tran tplh trig v(1) val=fifty1 fall=<number1>
+ targ v(2) val=fifty rise=<number2>
```

8 – Para medir  $t_{PD}$  do nó 1 para o nó 2:

```
.measure tran tpd param='(tphl+tplh)*0.5'
```

9 – Para medir  $\tau$  do sinal no nó 1:

```
.measure tran T trig v(1) val=fifty1 rise=<number>
+ targ v(1) val=fifty1 rise=<number+1>
```

O tempo de atraso necessário para um sinal atravessar um circuito lógico depende muito da complexidade desse circuito, isto é, depende do número de portas que o sinal tem que atravessar. Importante é caracterizar o tempo de atraso das portas básicas (inversores e NOR), pois o tempo de atraso do circuito completo será função dos atrasos de cada porta.

Os circuitos devem ser analisados para as situações de pior caso, combinando-se extremos de temperatura, de variações de processo e de tensões de alimentação em algumas simulações. Isso visa a garantir que o projeto desses circuitos vai atender às suas especificações sob todas as condições ambientais, de processo e elétricas. Essa metodologia é essencial para o projeto de CIs confiáveis e previsíveis.

#### 3.1.4. Modelos de Dispositivos para a Simulação de CIs de GaAs no Processo da Vitesse.

O projeto de CIs requer o uso de modelos precisos, tanto para os dispositivos ativos como para os elementos passivos, que serão empregados para as simulações em computador do circuito. Esses modelos devem então receber os valores numéricos para as variáveis das suas equações; estas variáveis são chamadas de **parâmetros**. Para o projetista de CIs, que é o usuário do programa de simulação, um modelo é especificado mais por seus parâmetros, e nem tanto pelas equações.

Os valores dos parâmetros são determinados a partir de muitas séries de medições em dispositivos reais, com o tratamento estatístico adequado, para a caracterização do processo. No caso de fabricação contratada de um serviço de *foundry* externa, os valores dos parâmetros são fornecidos pela *foundry* onde o circuito projetado será fabricado. Esses valores são então passados como entrada para o modelo no simulador, para que haja uma boa correspondência entre os valores de tensão e corrente simulados em um circuito, e os valores medidos em circuitos reais. Valores precisos desses parâmetros são necessários para assegurar que a simulação seja confiável e representativa da realidade. O ajuste do modelo às medidas experimentais pode ser ainda melhorado se os parâmetros forem otimizados em computador antes de serem passados ao modelo. A *foundry* escolhida para a implementação do CI deste trabalho é a Vitesse Semiconductor Corp., sediada nos Estados Unidos.

As referências [11] e [13] apresentam os procedimentos e técnicas para a extração de parâmetros para o modelo de MESFET do SPICE, a partir de medidas realizadas em dispositivos reais. Esses procedimentos constituem métodos de caracterização: como determinar os valores dos parâmetros SPICE de vários dispositivos reais, fabricados em algum processo de semicondutores, para então utilizar esses valores na simulação e projeto de circuitos contendo esses dispositivos, que serão fabricados pelo mesmo processo. Os mesmos parâmetros podem também ser correlacionados com o processo de fabricação, servindo como controle de qualidade.

O simulador empregado neste trabalho é o HSPICE, uma versão do SPICE para estações de trabalho, com melhores modelos e recursos de análise. O modelo de MESFET de GaAs implementado no HSPICE e suportado pela Vitesse é baseado no modelo de Statz et al. [76], que foi desenvolvido inicialmente na Raytheon Co. em 1987, para o SPICE [11]. Atualmente, o modelo de MESFETs de Statz já é usado não somente pela Vitesse, mas também pelas versões comerciais mais recentes do SPICE3, como o IsSPICE4 da Intusoft Inc.. A versão do modelo de Statz usada pela Vitesse para a extração de parâmetros contém otimizações em relação à versão original publicada pelos autores [76].

O modelo de Statz inclui o efeito da saturação de velocidade de elétrons, que é próprio do GaAs, para obter uma concordância muito melhor com medidas experimentais de MESFETs, em comparação com outros modelos. Além disso, Statz também incorpora um modelo melhorado das capacitâncias internas do MESFET, e uma transição suave da operação do dispositivo para a região inversa (terceiro quadrante da curva I-V na figura 1.5), de modo que o MESFET seja completamente simétrico. Para famílias lógicas dinâmicas, que utilizam transistores de passagem [66], esta última característica é muito importante.

Uma descrição mais detalhada do modelo de Statz para MESFETs de GaAs, com uma lista dos seus parâmetros e a explicação do significado de cada parâmetro, é dada em [11], [12], [13] e [76]. Os valores *default* que o SPICE atribui aos parâmetros, não devem ser

usados, pois eles não representam uma *foundry*. Ao invés deles, devem ser usados valores determinados por caracterização, ou seja, extraídos de MESFETs fabricados, para que a simulação seja representativa de dispositivos reais de alguma *foundry*, isto é, para que a simulação tenha validade física. Os parâmetros de MESFETs fornecidos pela Vitesse descrevem os transistores fabricados pela empresa: auto-alinhados, com porta refratária e fabricados por implantação iônica. O circuito equivalente do MESFET correspondente ao modelo de Statz, para efeito de análise no HSPICE, é mostrado em [11], [12] e [66].

As capacitâncias internas dos MESFETs desempenham um papel importante na determinação do desempenho transiente de dispositivos e circuitos de GaAs. As equações das capacitâncias não-lineares  $C_{GS}$  e  $C_{GD}$  estão embutidas no modelo de Statz, e são calculadas pelo SPICE durante as simulações, em função das tensões presentes nos terminais do MESFET.  $C_{GS}$  e  $C_{GD}$  modelam as cargas armazenadas nas regiões de depleção entre as junções porta-fonte e porta-dreno, respectivamente, portanto elas variam com as tensões sobre essas junções. Além de  $C_{GS}$  e  $C_{GD}$ , existem também as capacitâncias parasitas fixas, resultantes dos efeitos de frangeamento entre os eletrodos na superfície do GaAs, e nas bordas laterais da camada de depleção da porta. Essas capacitâncias extras não são computadas no cálculo do modelo de MESFET, que é um cálculo de placas paralelas, porém elas devem ser consideradas nas simulações.

As capacitâncias totais entre os terminais do MESFET consistem da soma das capacitâncias variáveis com a tensão (de depleção) e das capacitâncias fixas (parasitas); estas dependem unicamente de características **geométricas** do MESFET, como os comprimentos dos eletrodos de dreno, porta e fonte e os espaçamentos entre eles, e são indicadas por  $C_{DSP}$ ,  $C_{GSP}$  e  $C_{GDP}$  na figura 3.5. As três capacitâncias parasitas podem ser incluídas no arquivo *netlist* de entrada do SPICE como elementos independentes, para cada MESFET, ou podem ser previamente embutidas no modelo de MESFET como elementos adicionais [11].

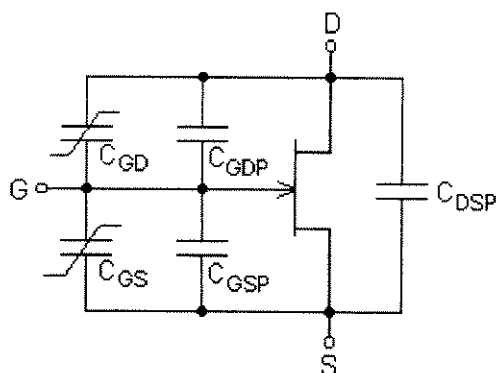


Figura 3.5 – Capacitâncias fixas e variáveis com a tensão, associadas ao MESFET.

As capacitâncias parasitas totais são escalonadas linearmente em função da área de porta  $A = W_g L_g$  do MESFET; seus valores por unidade de  $A$  são [11]:

$$C_{DSP} = 0,11 \text{ fF}/\mu\text{m}^2 \quad (3.2a)$$

$$C_{GSP} = C_{GDP} = 0,28 \text{ fF}/\mu\text{m}^2 \quad (3.2b)$$

Essas capacitâncias parasitas devem-se ao MESFET somente. Para a simulação de circuitos é indispensável considerar também as capacitâncias fixas de carga provenientes das interconexões metálicas entre os dispositivos e blocos no CI, como será visto no Capítulo 5.

O HSPICE não possui o dispositivo “MESFET” propriamente dito, apesar deste já existir nas demais versões comerciais do SPICE. Por isso, para as simulações de circuitos com MESFETs de GaAs no HSPICE, é usado o JFET como elemento dos circuitos, no lugar do MESFET. O JFET é simulado com parâmetros de MESFETs extraídos pela Vitesse, e distribuídos pelo CMP aos projetistas. Esses parâmetros correspondem ao modelo de Statz e ao processo HGaAs-III da Vitesse e são fornecidos nos arquivos de “biblioteca de parâmetros” **vsc3.11models**, **vsc3.11corners** e **vsc3.11skew**, conforme explicado em [12]. Os três arquivos de biblioteca são lidos pelo HSPICE e usados para a configuração do modelo de JFET implementado neste simulador; a partir daí, o JFET passa a representar o MESFET para o HSPICE. A forma de incluir esses arquivos de parâmetros em uma *netlist* do HSPICE é mostrada nas listagens do Apêndice 1.

O processo HGaAs-III permite realizar-se transistores com um  $L_g$  mínimo de  $0,6 \mu\text{m}$ . Entretanto, para todas as etapas do projeto de um CI (dimensionamento, simulação e desenho do *layout*), devem ser usadas as dimensões “gráficas” (*drawn dimensions*), o que, na tecnologia HGaAs-III, significa que o valor real  $L_g$  precisa ser acrescido de  $0,4 \mu\text{m}$ ; portanto, um MESFET com o  $L_g$  mínimo de  $0,6 \mu\text{m}$  (valor real) corresponderá, durante todo o projeto do circuito, a um FET com  $L_g$  igual a  $1 \mu\text{m}$ . Neste trabalho, ‘ $L_g$ ’ sempre denota os valores **reais** de comprimento de MESFETs, e ‘ $L_{g,\text{proj}}$ ’ denota os valores de **projeto**. Dessa forma, o projetista deve usar sempre o valor  $L_{g,\text{proj}}$ , definido como:

$$L_{g,\text{proj}} = L_g + 0,4 \mu\text{m} \quad (3.3)$$

A biblioteca fornecida pela Vitesse contém valores de parâmetros para MESFETs de comprimentos tipicamente usados em projetos de CIs. Há seis opções de JFETs contidas nos arquivos de biblioteca, e cada uma corresponde a um ajuste diferente do modelo de Statz a medidas experimentais, sendo referenciada por um nome de dispositivo particular. Esses são os seis MESFETs que a Vitesse caracterizou e determinou os parâmetros. É como se houvesse seis JFETs diferentes a escolher para usar nos circuitos, cada um definido por um conjunto próprio de valores dos seus parâmetros, que lhe são atribuídos por comandos **.model** do HSPICE, no arquivo **vsc3.11models**. Um projetista de CI só precisa especificar qual o nome do JFET, dentre os listados na primeira coluna da tabela 3.1, e o tamanho da instância desse transistor ( $W_g$  e  $L_{g,\text{proj}}$ ), e o simulador selecionará os valores de parâmetros adequados.

| Nome de transistor | Tipo de FET      | $W_g \times L_g$            | $L_{g,\text{proj}}$ |
|--------------------|------------------|-----------------------------|---------------------|
| jenh (jfet04)      | EFET             | $10 \times 0,6 \mu\text{m}$ | $1 \mu\text{m}$     |
| enh2 (jfet10)      | EFET             | $10 \times 1,1 \mu\text{m}$ | $1,5 \mu\text{m}$   |
| dp21 (jfet15)      | DFET (para RAMs) | $10 \times 1,6 \mu\text{m}$ | $2 \mu\text{m}$     |
| dp11 (jfet16)      | DFET             | $10 \times 0,6 \mu\text{m}$ | $1 \mu\text{m}$     |
| jdep (jfet19)      | DFET             | $10 \times 1,6 \mu\text{m}$ | $2 \mu\text{m}$     |
| dp12 (jfet20)      | DFET             | $10 \times 2,6 \mu\text{m}$ | $3 \mu\text{m}$     |

Tabela 3.1 – Nomes dos conjuntos de parâmetros para os JFETs padrão da Vitesse [12].

Na tabela 3.1, os nomes de transistor entre parênteses não devem ser usados, pois são nomes internos aos arquivos de biblioteca, devendo ser considerados somente os primeiros nomes. Em uma *netlist* de entrada para o HSPICE, cada diferente valor de  $L_{g,\text{proj}}$  de um JFET determina o uso do “nome de transistor” correspondente, segundo a tabela 3.1. Cada “nome de transistor” é válido para descrever somente os JFETs que tenham o mesmo  $L_{g,\text{proj}}$  especificado para esse nome. Não se deve utilizar, em uma *netlist*, qualquer instância de JFET

com  $L_{g,proj}$  diferente de um dos valores da última coluna da tabela 3.1, isto é, somente os valores de  $L_{g,proj}$  da tabela 3.1 são permitidos nas simulações. A razão para isso é que os valores dos parâmetros de um MESFET variam largamente com seu  $L_g$ , logo esses parâmetros só são válidos para simular MESFETs que tenham o mesmo  $L_g$  dos MESFETs de teste nos quais eles foram medidos. Quanto à largura  $W_g$ , ela pode ser escalonada livremente para qualquer um dos “nomes” de JFETs, porque o erro resultante da simulação é muito menor.

Para cada tipo padrão de MESFET na tabela 3.1, os conjuntos de parâmetros foram determinados para cinco cantos do processo HGaAs-III. Os cantos de processo considerados, em ordem crescente de velocidade do MESFET, foram: “*slowfunc*”, “*slow*”, “*typical*”, “*fast*”, “*fastfunc*”. Esses conjuntos de parâmetros refletem principalmente a variação de  $V_t$  dos MESFETs devida a variações do processo de fabricação, sendo que, se os valores algébricos de  $V_{tE}$  ou  $V_{tD}$  diminuem, os EFETs ou DFETs produzidos serão mais rápidos, porque aumentarão as suas correntes de dreno. Os conjuntos de parâmetros rotulados como *slowfunc* e *fastfunc* modelam um espalhamento adicional (mais largo) do processo, e são destinados para análises de Monte Carlo; esse tipo de análise não é necessária para este projeto [12], para o qual as análises em “cantos” padrão do processo já são suficientes.

A própria Vitesse recomenda o emprego da lógica DCFL nos projetos de CIs para HGaAs-III. Para DCFL, o EFET padrão utilizado é o “jenh” ( $L_{g,proj} = 1 \mu m$ ), e o DFET padrão para cargas ativas é o “jdep” ( $L_{g,proj} = 2 \mu m$ ) [12], listados na tabela 3.1.

Para simular-se diodos Schottky, existem em princípio duas alternativas: a primeira é usar-se o elemento “diodo” propriamente dito do HSPICE, representado pelo identificador ‘D’. A segunda maneira, que é o método adotado pela Vitesse, consiste em simular os diodos usando-se DFETs com o dreno e a fonte em curto ([12] e [13]), como na figura 3.6. Esse método elimina a necessidade de um modelo de simulação específico para diodos.

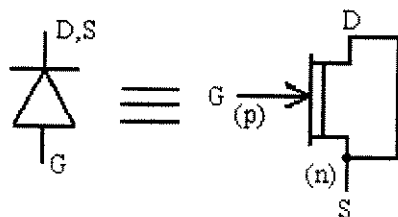


Figura 3.6 – Como um diodo deve ser realizado nas simulações de CIs para a Vitesse.

A dependência de alguns parâmetros do modelo de Statz com a  $W_g$  do MESFET é linearmente proporcional, conforme indicado nas relações (3.4), para um  $L_g$  mantido fixo.

$$W_g \uparrow \Rightarrow I_S \uparrow, C_{GS0} \uparrow, C_{GD0} \uparrow, R_S \downarrow, R_D \downarrow, \beta_0 \uparrow \quad (3.4)$$

### 3.2. Projeto de Inversores DCFL.

O projeto de circuitos em DCFL guarda alguma semelhança com o projeto em NMOS, porque ambas as famílias empregam FETs de enriquecimento e de depleção, e o desempenho

de uma porta lógica depende principalmente da razão entre as dimensões desses FETs (ambos são *ratioed designs*).

A porta lógica mais básica de qualquer família digital, seja em Si ou em GaAs, é o inversor. Por isso, todo estudo de uma família lógica deve iniciar pelo estudo do seu inversor, o que é feito nesta seção. A seção 3.3 discute o projeto da porta NOR em DCFL.

O inversor DCFL com carga ativa a DFET foi mostrado na figura 2.1(b), repetida aqui por conveniência.

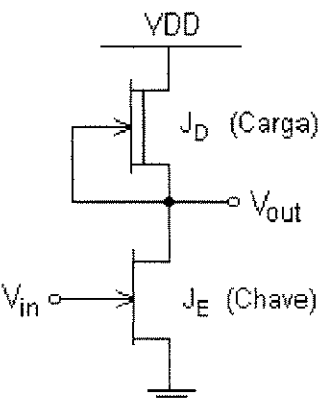


Figura 3.7 – Inversor DCFL.

Quando  $V_{in} > V_{IH}$ , a chave  $J_E$  fecha, passando uma corrente por ela e a sua carga  $J_D$ , e  $V_{out}$  será igual ao valor da queda de tensão na resistência  $r_{ds}$  de  $J_E$ ; se  $r_{ds}$  for pequena,  $V_{out}$  terá um valor baixo,  $V_{OL}$ . Quando  $V_{in} < V_{IL}$ , a chave permanece aberta; se este inversor não possuir uma carga externa que lhe solicite uma corrente (por exemplo, outra porta lógica ou uma carga passiva), então não passará corrente por  $J_D$  e  $V_{out} = V_{DD}$ . Mas se este inversor acionar uma carga que “puxe” uma corrente de entrada DC para si (como um estágio DCFL seguinte), então a  $V_{out}$  final será menor do que  $V_{DD}$ , devido à queda na resistência interna do DFET. Em ambos os casos, para uma  $V_{in}$  baixa,  $V_{out}$  terá um valor maior do que  $V_{OL}$ , que é referido como  $V_{OH}$ .

Dentro de um CI, um estágio DCFL sempre estará acionando outro, de forma que a sua  $V_{out}$  será limitada pela condução direta do diodo de porta do estágio seguinte. Como a junção Schottky em um MESFET limita a tensão direta de porta-fonte em aproximadamente 0,7 V, este fica sendo o valor típico de  $V_{OH}$  para circuitos de alimentação positiva. Os níveis  $V_{OL}$  e  $V_{OH}$  de portas DCFL dependem fortemente do dimensionamento dos transistores.

O DFET  $J_D$  deve ser mantido na saturação durante a região de transição da curva característica do inversor, para operar como uma fonte de corrente e assim exibir uma alta  $r_{ds}$ . Isso aumenta  $A_v$ , que por sua vez aumenta a NM do inversor, como explicado no Capítulo 2. No processo HGaAs-III, o DFET possui maior  $r_{ds}$  que o EFET, por ter um menor valor do parâmetro de modulação do comprimento do canal,  $\lambda$  ([9], [12]).  $J_D$  fornece a corrente de carga do capacitor quando o nível baixo  $V_{OL}$  é aplicado na entrada de  $J_E$ , levando então a saída para o nível alto  $V_{OH}$ ; por isso,  $J_D$  também é chamado “transistor de *pull-up*”. A razão de aspecto  $W_g/L_{g,proj}$  de  $J_D$ ,  $W_D/L_D$ , é dimensionada de acordo com a corrente necessária para carregar a capacitância de carga  $C_L$  num intervalo de tempo preestabelecido, dada pela relação:



$$I = C_L \frac{dV}{dt} \quad (3.5)$$

O transistor de enriquecimento  $J_E$  opera como uma chave, e quando um nível alto  $V_{OH}$  é aplicado em seu terminal de porta, deve puxar uma corrente maior que a fornecida por  $J_D$ , dessa forma descarregando  $C_L$ ; portanto,  $J_E$  é também chamado de “transistor de *pull-down*”. A razão  $W_g/L_{g,proj}$  de  $J_E$ ,  $W_E/L_E$ , determina principalmente o valor de  $V_{OL}$ ; quanto maior for  $W_E/L_E$ , menor será o  $V_{OL}$  porque maior será a capacidade de  $J_E$  de drenar a corrente de sua carga  $J_D$  (chamada de *pull-down strength* de  $J_E$ ), e ainda descarregar rapidamente  $C_L$ ; enfim, maior será a NM da porta lógica. Como  $J_E$  precisa conduzir mais corrente que  $J_D$ ,  $W_E/L_E$  precisa ser maior que  $W_D/L_D$  por um certo fator, que é definido como o parâmetro  $\beta$  de uma porta lógica, dado pela expressão:

$$\beta = \frac{(W_E/L_E)}{(W_D/L_D)} = \frac{W_E L_D}{W_D L_E} \quad (3.6)$$

O parâmetro  $\beta$  é a razão entre as razões de aspecto do EFET e do DFET, e tem muita influência sobre o desempenho de uma porta DCFL, pois determina os seus níveis lógicos.

A figura 3.8(a) indica as correntes em um inversor, sendo que  $i_d = i_e + i_{out}$  em todos os instantes; a figura 3.9 mostra essas três correntes, todas em função da tensão de saída do inversor,  $V_{out} (= V_{DS,E})$ , na forma da família de características  $I_D \times V_{DS}$  de um EFET, para vários valores de  $V_{in} (= V_{GS,E})$ , sobrepostas com a curva de carga do DFET e com a curva  $I \times V$  do diodo de carga D. A tensão de alimentação  $V_{DD}$  é igual a 2 V.

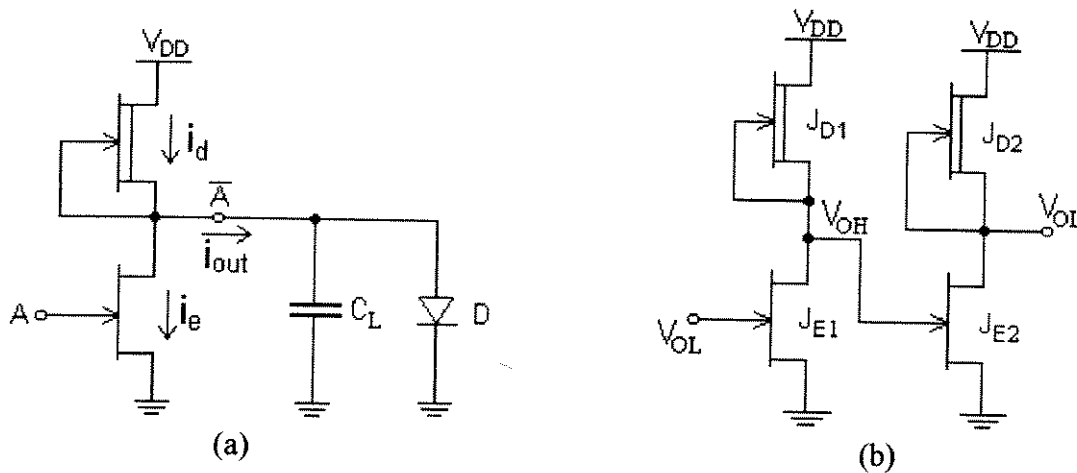


Figura 3.8 – a) Inversor DCFL com uma carga. O diodo e o capacitor  $C_L$  representam a junção de porta do EFET do estágio seguinte. b) Inversor DCFL acionando outro idêntico.

Para compreender a curva de  $i_d$ , basta lembrar que a tensão  $V_{DS}$  do DFET é igual a  $V_{DD} - V_{out}$ , portanto a sua curva  $I_{D,D} \times V_{DS,D}$  foi multiplicada por  $-1$  e depois deslocada para a direita por  $V_{DD}$ . Somando-se as curvas de corrente do EFET,  $i_e$ , com a corrente do diodo,  $i_{out}$ , resulta o conjunto de curvas da figura 3.10, tendo  $V_{GS,E}$  como parâmetro. Os pontos de intercessão entre essas curvas e a curva  $i_d$  do DFET determinam a característica de transferência  $V_{out} \times V_{in}$  do inversor, mostrada na figura 3.12; a justificativa para essas operações é o equacionamento  $i_d = i_e + i_{out}$  na figura 3.8(a).

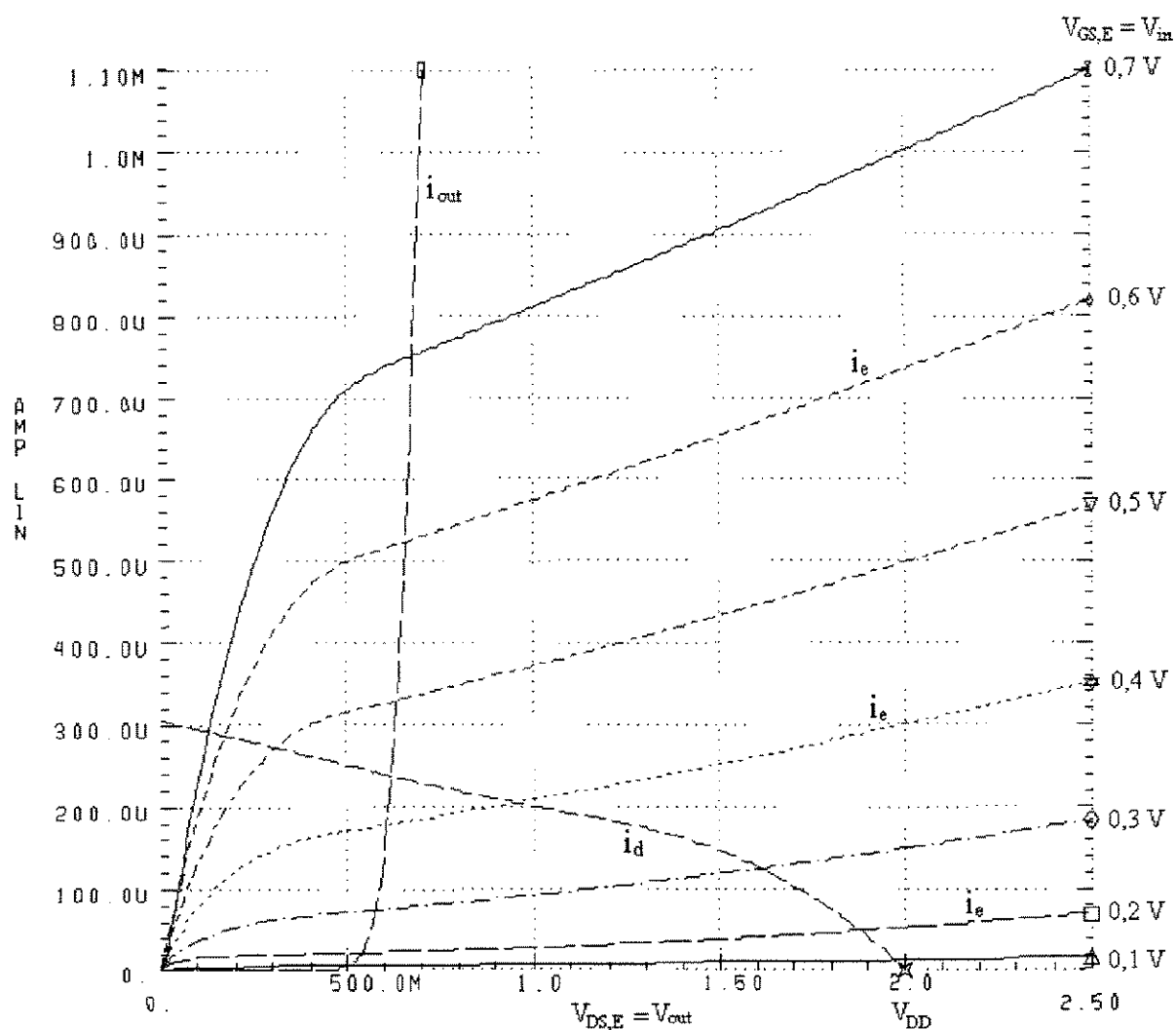


Figura 3.9 – Curvas  $I_D \times V_{out}$  do EFET ( $i_e$ ) e do DFET ( $i_d$ ) de um inversor DCFL, e curva da corrente de entrada  $I_{GS}$  de um inversor de carga ( $i_{out}$ ).

No Capítulo 1, foi explicado que a  $V_{in}$  de uma porta lógica nunca deve ser alta demais (maior do que 0,8 V), por causa do aumento de  $I_{GS}$  e da queda em  $R_s$ . Mas existe outro efeito nocivo de um valor excessivamente alto de  $V_{OH}$  aplicado na entrada de uma porta: como a tensão de dreno correspondente é baixa ( $V_{OL}$ ),  $V_{OH}$  provoca também a condução do diodo de porta-dreno do MESFET, além do diodo de porta-fonte. A corrente no diodo de porta-dreno flui em sentido oposto ao da corrente de dreno  $i_e$  do EFET, reduzindo-a. Com isso,  $i_d$  diminui, a queda de tensão sobre  $J_D$  também é reduzida, e  $V_{OL}$  aumenta.

O efeito de um  $V_{in}$  muito alto pode ser apreciado na curva do inversor da figura 3.12, e sua origem é mostrada na figura 3.11, onde valores em excesso de  $V_{GS,E}$  são usados. Nota-se que o valor de  $V_{out}$ , que ia ficando cada vez menor com o aumento de  $V_{in}$ , torna a crescer a partir de um certo  $V_{in}$ .

O valor correto de  $V_{OH}$  para acionar a entrada de um próximo estágio lógico é somente em torno de 0,7 V. Na figura 3.8(b), se o valor de  $V_{OH}$ , na saída do primeiro inversor, for menor do que 0,7 V (*underdrive*), as NM's da porta acionadora (o primeiro inversor) são

reduzidas. Mas se o valor de  $V_{OH}$  for muito alto, maior do que 0,8 V (*overdrive*), as NMs da porta acionada (o segundo inversor) é que serão reduzidas [77]. A curva do inversor mostra o efeito do *overdrive* na entrada: enquanto  $V_{in}$  é logicamente alta com um valor menor do que 0,7 V,  $V_{OL}$  será pequeno, mas qualquer aumento de  $V_{in}$  além de 0,7 V resulta em um aumento de  $V_{OL}$ . Isso diminui a NM do circuito e pode induzir a um erro lógico.

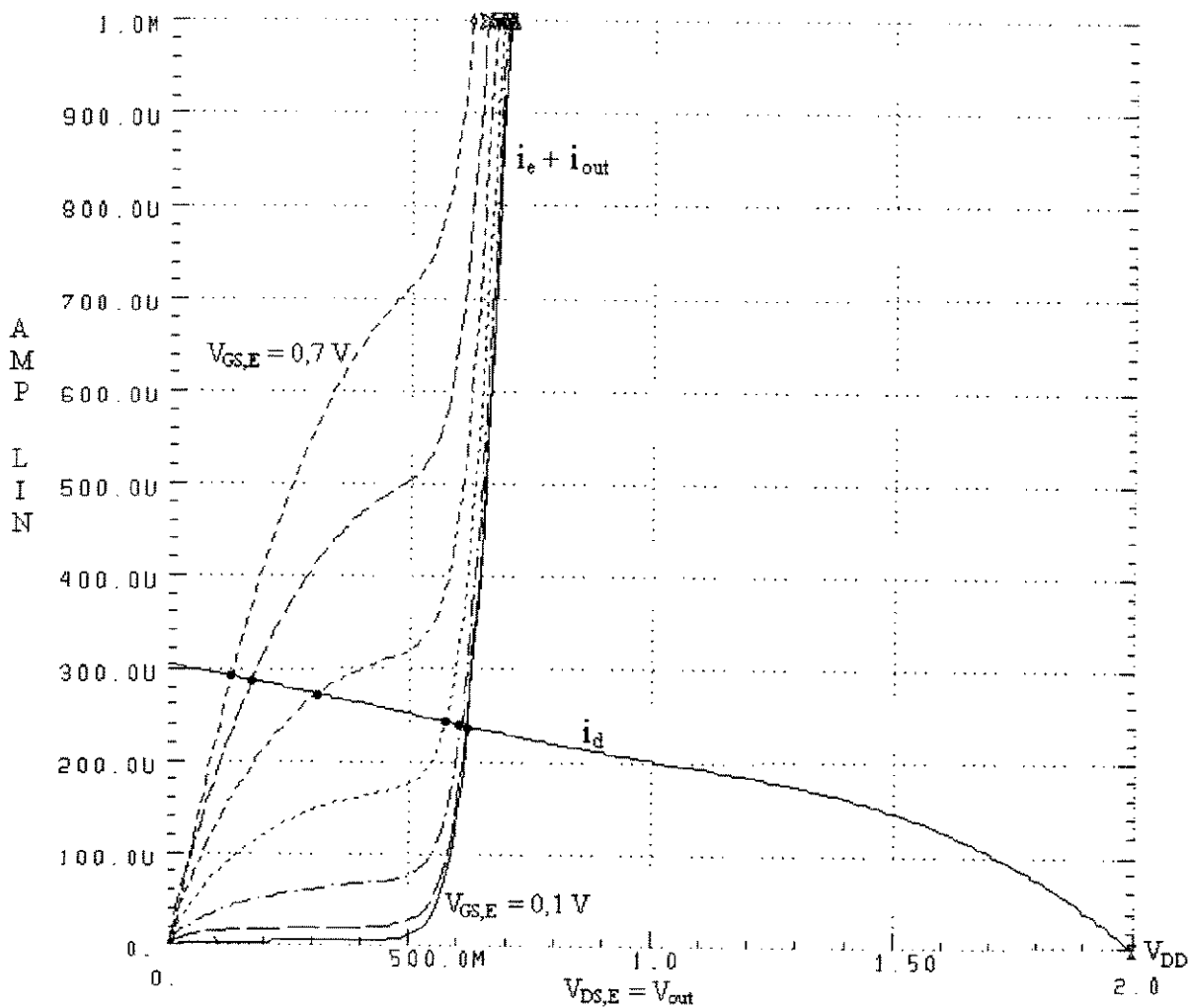


Figura 3.10 – Os pontos de interseção entre as curvas  $i_d$  e  $(i_e + i_{out})$  determinam as coordenadas  $V_{in}$  e  $V_{out}$  da curva de transferência do inversor.

As curvas das figuras 3.9 a 3.11 foram geradas pelo Programa 1 do Apêndice 1.

A origem de  $V_{OL}$  está na queda de tensão sobre a resistência  $r_{ds}$  do EFET quando percorrida por  $I_{D,E}$ ; para circuitos digitais, uma menor  $r_{ds}$  do EFET é uma vantagem.  $V_{OL}$  na saída de uma porta lógica deve estar o mais abaixo possível do valor de  $V_t$  para reduzir a corrente de sub-limiar no EFET do estágio lógico seguinte. O efeito de  $I_{SUB}$  na porta é descarregar  $C_L$  um pouco e deteriorar o  $V_{OH}$  e a NM da porta [77], como visto na figura 3.13. Para que  $V_{OL}$  seja a mais baixa possível, pode-se usar *buffers* na saída da porta.

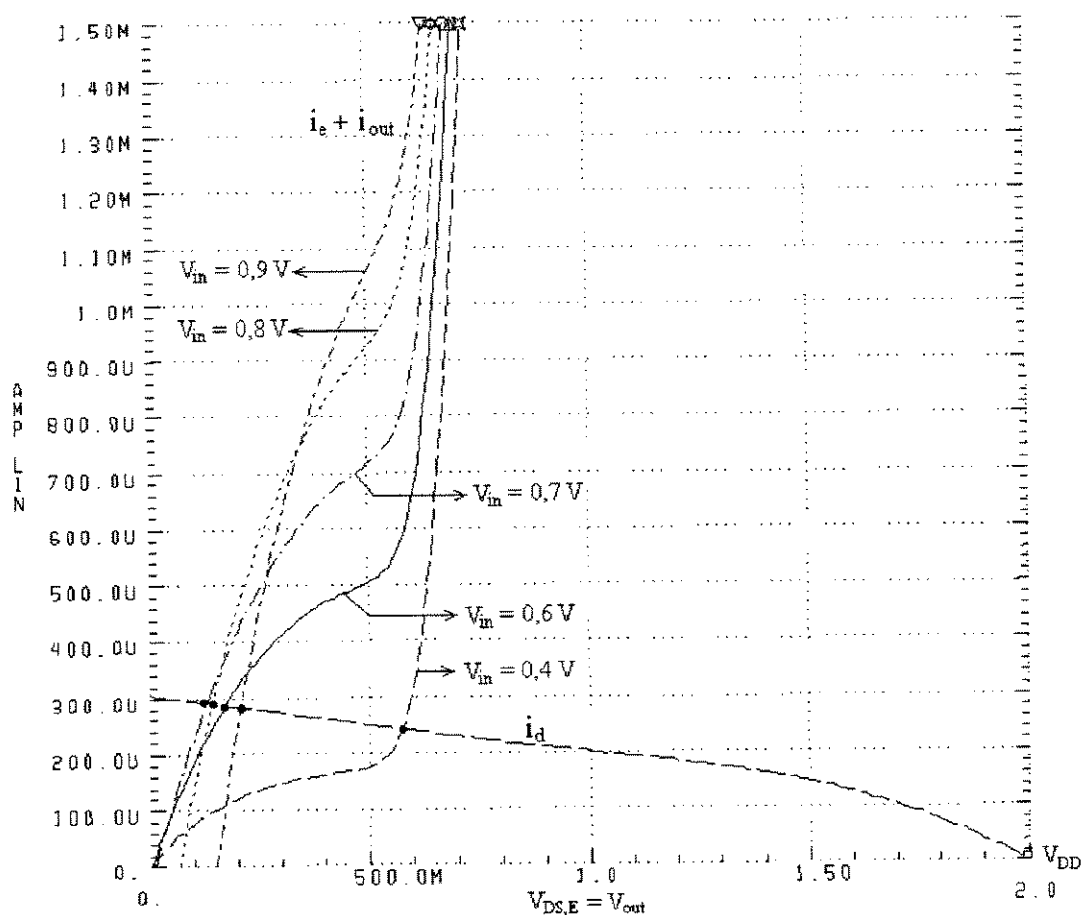


Figura 3.11 – Ilustração das curvas ( $i_e + i_{out}$ ) para  $V_{in} > 0,7$  V (*overdrive*).

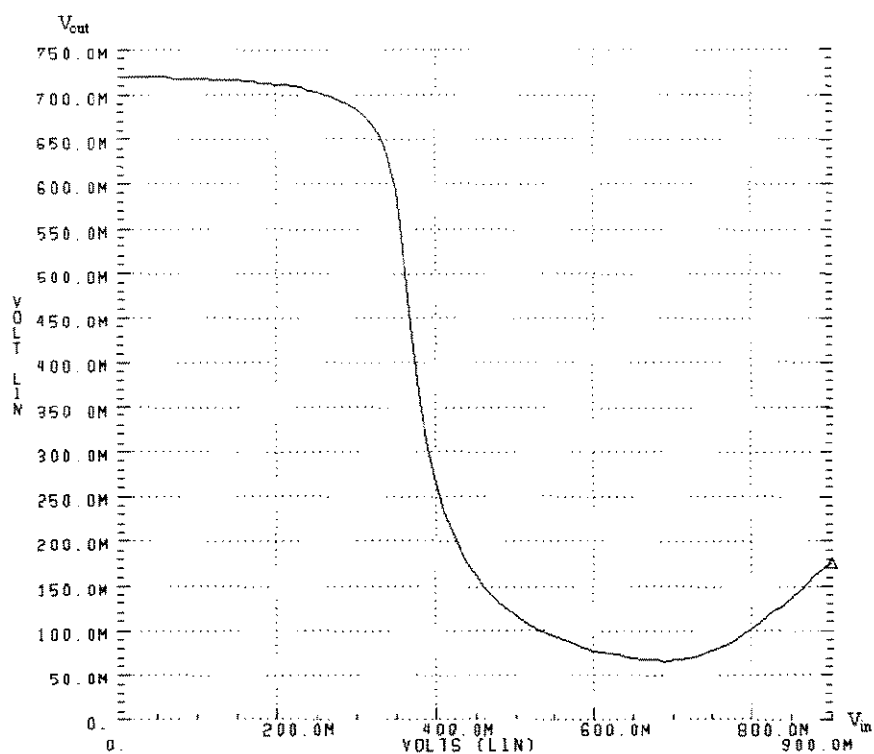


Figura 3.12 – Curva de transferência de um inversor DCFL, formada pela interseção das curvas de  $i_d$  e ( $i_e + i_{out}$ ) nas figuras 3.10 e 3.11.

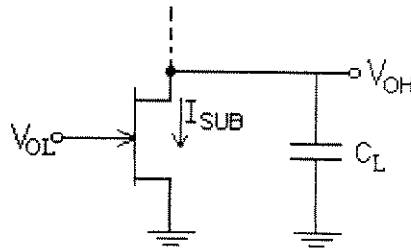


Figura 3.13 – Efeito da corrente  $I_{SUB}$ .

### 3.2.1. Filosofia de Projeto.

O dimensionamento dos transistores para a otimização do projeto é um processo iterativo, portanto deve-se procurar reduzir o número de iterações; por “otimização” entende-se explorar as diversas variações e possibilidades de projeto para poder escolher conscientemente a melhor alternativa. Com o objetivo de se adquirir subsídios para o dimensionamento das portas lógicas e *buffers* que irão compor os flip-flops do CI 7490-like, muitas simulações em computador foram realizadas previamente, para que fossem analisados os efeitos das variações das dimensões  $W_D$  e  $W_E$  e das relações  $\beta$ , dos inversores e das portas NOR, sobre seus desempenhos DC e transiente. Os compromissos de projeto (*trade-offs*) mais importantes foram confrontados entre si.

Como as equações precisas de  $I \times V$  e  $C \times V$  de um MESFET são muito complexas, neste trabalho não serão derivadas expressões analíticas dos parâmetros de desempenho dos circuitos, o que é feito com bastante rigor em [6]. As equações para inversores de lógica E/D deduzidas em [66] são usadas como ponto de partida para as análises deste capítulo, e não serão repetidas aqui. O dimensionamento final e mais preciso para todos os FETs do CI é determinado através de simulações em computador, usando o HSPICE, tal como nos projetos de CIs descritos em [4]. O método de projeto gráfico também propicia uma boa intuição e um conhecimento sobre a importância relativa de parâmetros do dispositivo, resultando em um bom “*feeling*” para o projetista. Esta deve ser a primeira etapa de um projeto.

Para avaliar-se os compromissos de projeto para uma porta lógica, é importante que ela seja simulada realisticamente, em condições semelhantes às do interior do CI. A forma de onda de entrada para a porta em análise e o carregamento na sua saída devem ser realistas.

Em um bom projeto, o circuito tem que garantir o ponto de operação, e não os parâmetros internos dos transistores. Deve-se evitar o emprego de estruturas críticas, a não ser que seja absolutamente necessário. As prioridades de projeto devem ser a estabilidade do circuito e sua funcionalidade DC, e não a sua velocidade, que deve ficar em segundo plano. Do contrário, a rodada de fabricação não terá um bom rendimento, e os resultados poderão ser CIs muito rápidos porém instáveis e muito críticos. Por isso, o mais importante é maximizar as NMs, e tal é a orientação deste projeto.

### 3.2.2. Parâmetros Fixos do Projeto.

O inversor a ser analisado por meio de simulações é igual ao da figura 3.7. Para facilitar a visualização e interpretação dos resultados, todas as simulações de circuitos neste capítulo são feitas com uma fonte de alimentação positiva  $V_{DD} = 2$  V.

Para circuitos digitais em GaAs, o valor do comprimento de porta do EFET,  $L_E$ , deve ser sempre o mínimo permitido pela tecnologia, que no caso do processo HGaAs-III é de  $0,6\ \mu\text{m}$ , desenhado e simulado como  $L_E = 1\ \mu\text{m}$ . As vantagens de usar o  $L_E$  mínimo em circuitos digitais são [11]:  $L_E \downarrow$ ,  $NM \uparrow$ ,  $g_m \uparrow$ ,  $r_{ds} \downarrow$ ,  $f_t \uparrow$ . Além disso, a  $C_{in}$  de uma porta lógica é diretamente proporcional à área de *gate* do EFET,  $W_g.L_g$ . Portanto, neste projeto será usado o EFET de nome “jenh”, recomendado pela Vitesse, que possui  $L_E = 1\ \mu\text{m}$ .

Quanto ao tipo de DFET, para o projeto do *chip 7490-like* foi usado o “jdep” da tabela 3.1, por ser o recomendado pela Vitesse para circuitos DCFL [12]. Portanto, as análises de portas deste Capítulo também empregarão somente os DFETs “jdep”, cujo  $L_D$  deve ser de  $2\ \mu\text{m}$ ; este é o  $L_{g,proj}$  sugerido pela Vitesse para os DFETs ([9], [12]).

Para efeito de comparação, usar o DFET “dp11” ( $L_D = 1\ \mu\text{m}$ ) em uma porta lógica, em vez do “jdep”, propicia uma maior velocidade da porta, porque menores valores de  $L_D$  viabilizam maior corrente de dreno por unidade de largura do MESFET. Entretanto,  $P_{D,st}$  aumenta e a  $V_{pp}$  na saída da porta diminui, reduzindo as NMs e tornando o circuito mais crítico, isto é, mais propenso a ter problemas de funcionalidade DC. Portanto, apesar da desvantagem no desempenho transiente, optou-se por seguir a recomendação da Vitesse e adotar  $L_D = 2\ \mu\text{m}$  em todo o projeto, pela segurança e estabilidade de funcionamento DC.

As três capacitâncias parasitas de cada FET, apresentadas nas equações 3.2a e 3.2b e na figura 3.14, não fazem parte do modelo do HSPICE, e devem ser incluídas nas *netlists* de entrada como elementos independentes para cada MESFET. Os DFETs não possuem as capacitâncias entre porta e fonte, pois estes terminais estão sempre em curto. As capacitâncias foram calculadas em função da área de cada MESFET da seguinte forma:

$$C_{GSPE} = C_{GDPE} = 0,28.W_E\ \text{fF, para todo EFET}$$

$$C_{out} = 2 \times C_{DSPE} + C_{DSPD} + C_{GDPD} = [0,22.W_E + (0,28 + 0,11).W_D.L_D]\ \text{fF, de } V_{out}\ \text{para o terra.}$$

sendo que os valores de  $W_E$ ,  $W_D$  e  $L_D$  já devem ser especificados em  $\mu\text{m}$ .

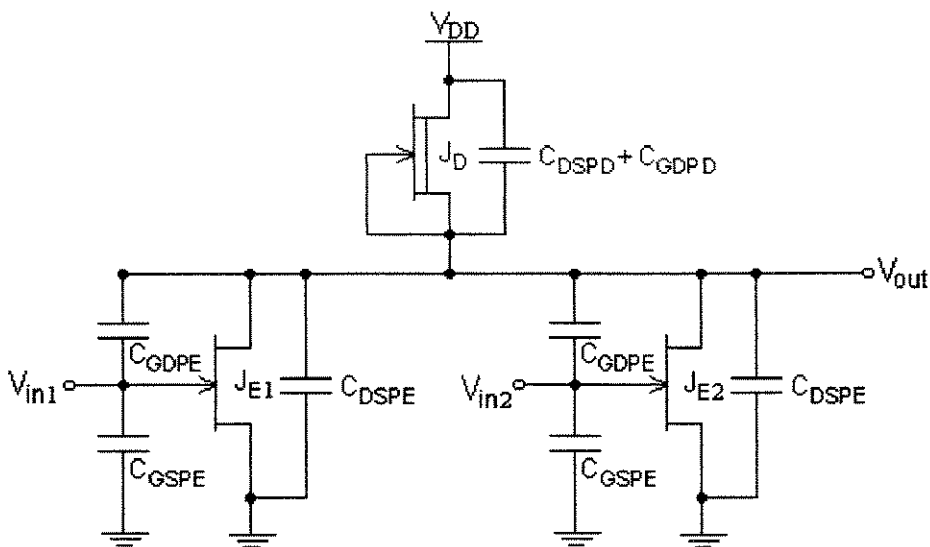


Figura 3.14 – Como as portas NOR foram descritas nas *netlists* para o HSPICE.

### 3.2.3. A Influência do Parâmetro $\beta$ no Projeto de um Inversor.

O objetivo desta seção é analisar a influência do parâmetro  $\beta$  nos desempenhos DC e transiente de inversores. É usado o inversor da figura 3.7 como modelo.

#### 3.2.3.1. Simulações DC.

As simulações DC analisam a influência dos parâmetros de projeto de uma porta lógica sobre os seus níveis lógicos, margens de ruído e  $P_{D,st}$ . Para as simulações DC, conecta-se a entrada da porta em análise diretamente a uma fonte de tensão DC, cujo valor é incrementado iterativamente em passos, por meio do comando de varredura `.dc` do HSPICE [75]. A saída dessa porta deve estar carregada realisticamente com capacitâncias de carga fixas,  $C_L$ , que modelam as capacitâncias das interconexões, e *fan-outs*, que são outras portas lógicas acionadas pela porta em análise.

O inversor da figura 3.7 foi simulado com uma razão  $W_D/L_D$  fixa em  $8\ \mu\text{m} / 2\ \mu\text{m}$ , fazendo-se o seu  $\beta$  ( $= W_E / 4$ ) variar na faixa de 7 a 13. As curvas de transferência DC, na figura 3.15, mostram que aumentar  $\beta$  diminui tanto  $V_{OH}$  quanto  $V_{OL}$ , e as figuras 3.16 e 3.17 mostram as dependências de  $V_{OH}$  e  $V_{OL}$  com  $\beta$ . Pela figura 3.15, percebe-se que um  $\beta < 7$  não serve, pois resulta em um  $V_{OL}$  muito alto e uma  $V_{MS,L}$  muito pequena; e valores de  $\beta$  maiores do que 13 já não reduzem mais  $V_{OL}$  significativamente, não sendo portanto recomendáveis tendo em vista o aumento de área ocupada pelos inversores e o aumento de seus atrasos de propagação (que será mostrado na seção seguinte).

De modo a aumentar  $V_{OH}$  para um inversor, deve-se aumentar a razão entre as correntes de depleção e de enriquecimento,  $W_D/W_E$ . Entretanto, para reduzir  $V_{OL}$  é preciso aumentar  $\beta$ , o que significa aumentar  $W_E$  [66], aumentando a  $I_D$  do EFET para o mesmo  $V_{in}$ ; logo, a curva de transferência do inversor é deslocada para a esquerda. Felizmente,  $V_{OH}$  tem um limite máximo aceitável bem definido (figura 3.12) e, do ponto de vista DC, deve ser escolhido um  $\beta$  que minimize  $V_{OL}$  enquanto ainda mantém  $V_{OH}$  próximo de 700 mV.

As NMs pelo método do máximo quadrado podem ser estimadas da figura 3.18.

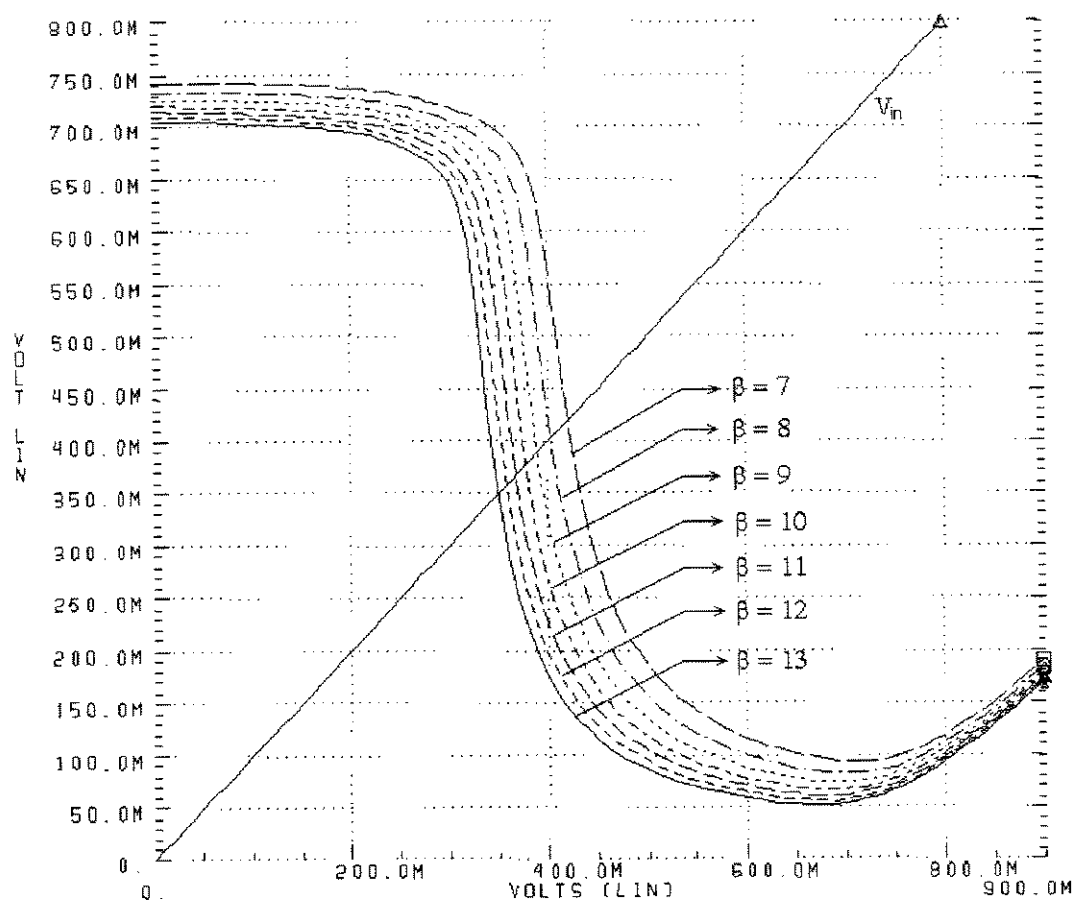


Figura 3.15 - Simulação DC do inversor da figura 3.7, tendo  $\beta$  como parâmetro e  $W_D$  constante e igual a  $8\ \mu\text{m}$ .

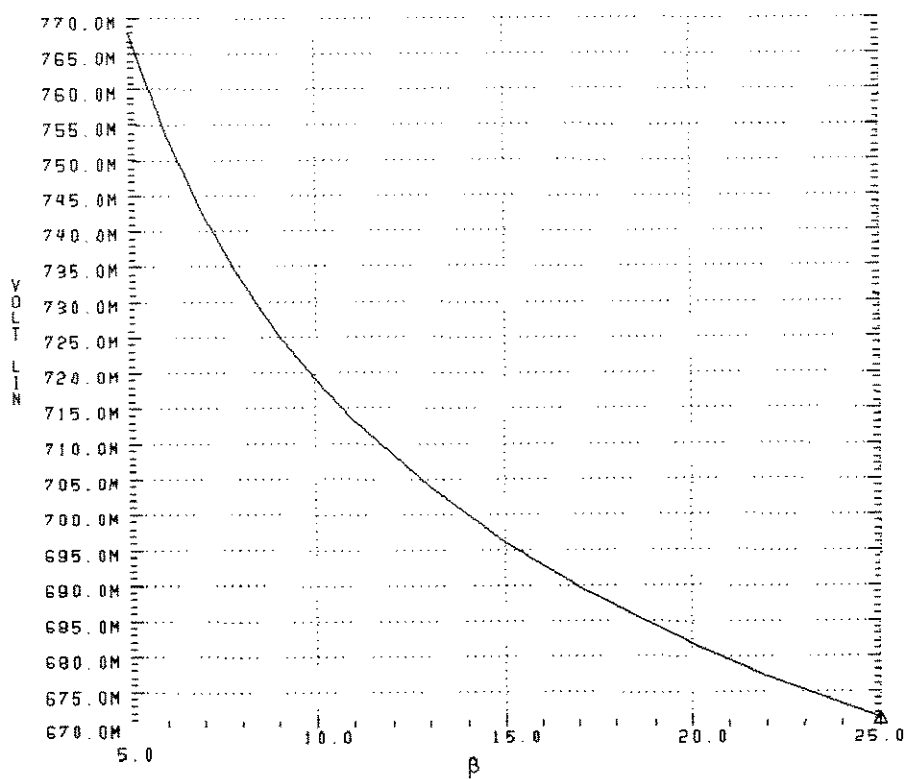


Figura 3.16 –  $V_{OH} \times \beta$  para um inversor DCFL.



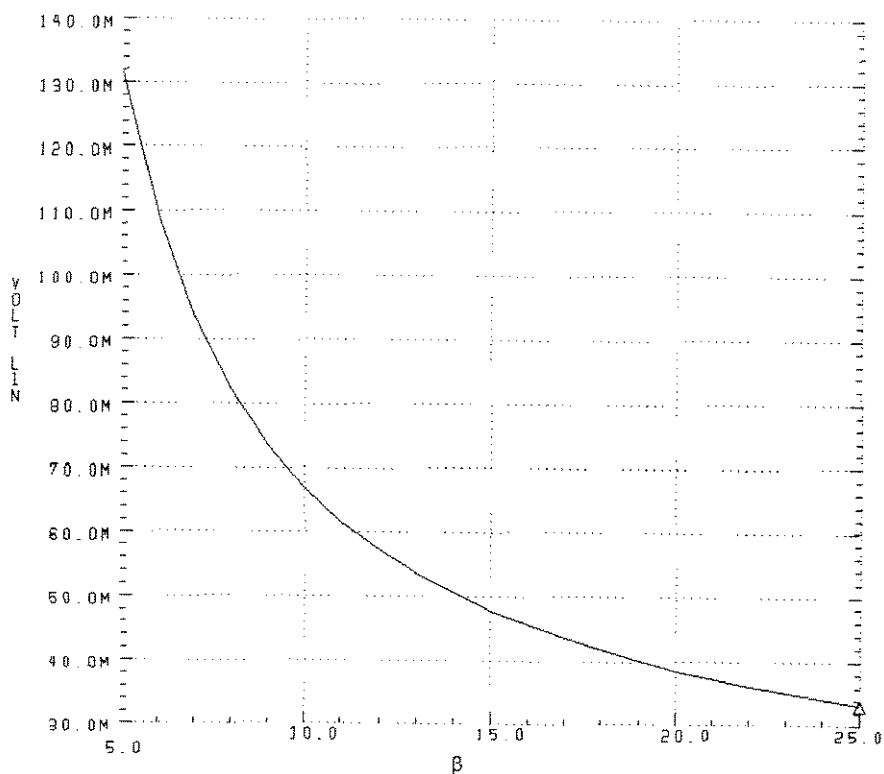


Figura 3.17 -  $V_{OL} \times \beta$  para um inversor DCFL.

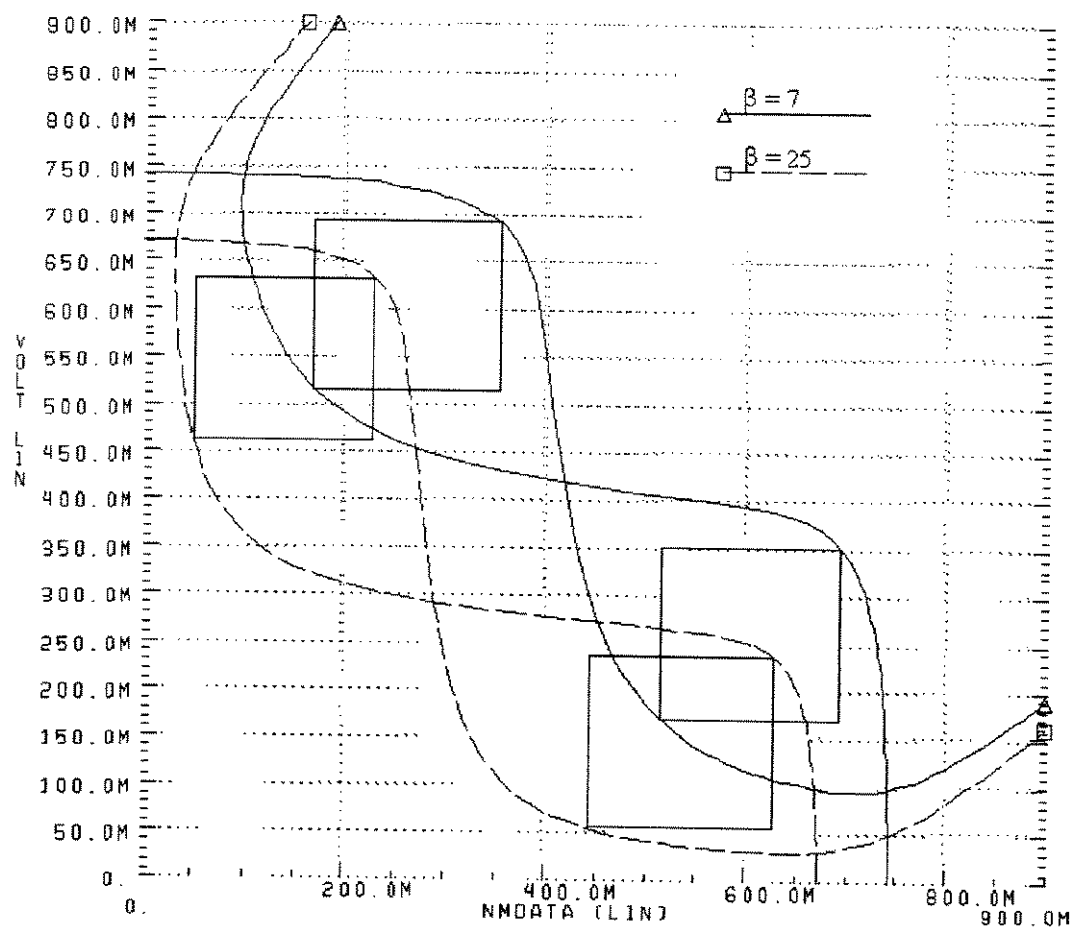


Figura 3.18 – Construção gráfica para avaliar-se as NMs pelo método do máximo quadrado.

A figura 3.19 mostra as curvas das correntes do inversor parametrizadas em  $\beta$ . As correntes em  $J_D$  e  $J_E$  aumentam com o aumento de  $V_{in}$ , mas a variação total de  $I_{D,E}$  é muito maior que a variação total de  $I_{D,D}$ , pois  $J_D$  está a maior parte do tempo em saturação, enquanto  $J_E$  vai do corte para a região linear. Pode-se observar  $I_{D,E}$  crescendo mesmo enquanto  $V_{GS,E} < V_{t,E}$  (220 mV), demonstrando o efeito da corrente  $I_{SUB}$  dos EFETs. A taxa de crescimento de  $I_{D,E}$  é inicialmente baixa, mas depois  $d(I_{D,E}) / d(V_{in})$  começa a aumentar cada vez mais, indicando que a curva de transferência entrou na sua região de transição. Quando os EFETs estão em plena condução,  $I_{D,E}$  fica em torno de 610  $\mu A$ , significando uma variação total de  $I_{D,E}$  de 608  $\mu A$ .

A excursão total de  $I_{D,D}$  é por volta de 150  $\mu A$ , indo de uma faixa de 460 ~ 470  $\mu A$ , enquanto  $V_{in} = V_{OL}$ , até um máximo de 610  $\mu A$  com  $V_{in} = 0,7$  V.  $I_{D,D}$  é aproximadamente independente de  $\beta$ , e como  $I_{D,D}$  é a própria corrente da fonte  $V_{DD}$ ,  **$P_{D,st}$  também varia muito pouco com  $\beta$** , em torno de apenas 20  $\mu W$ .

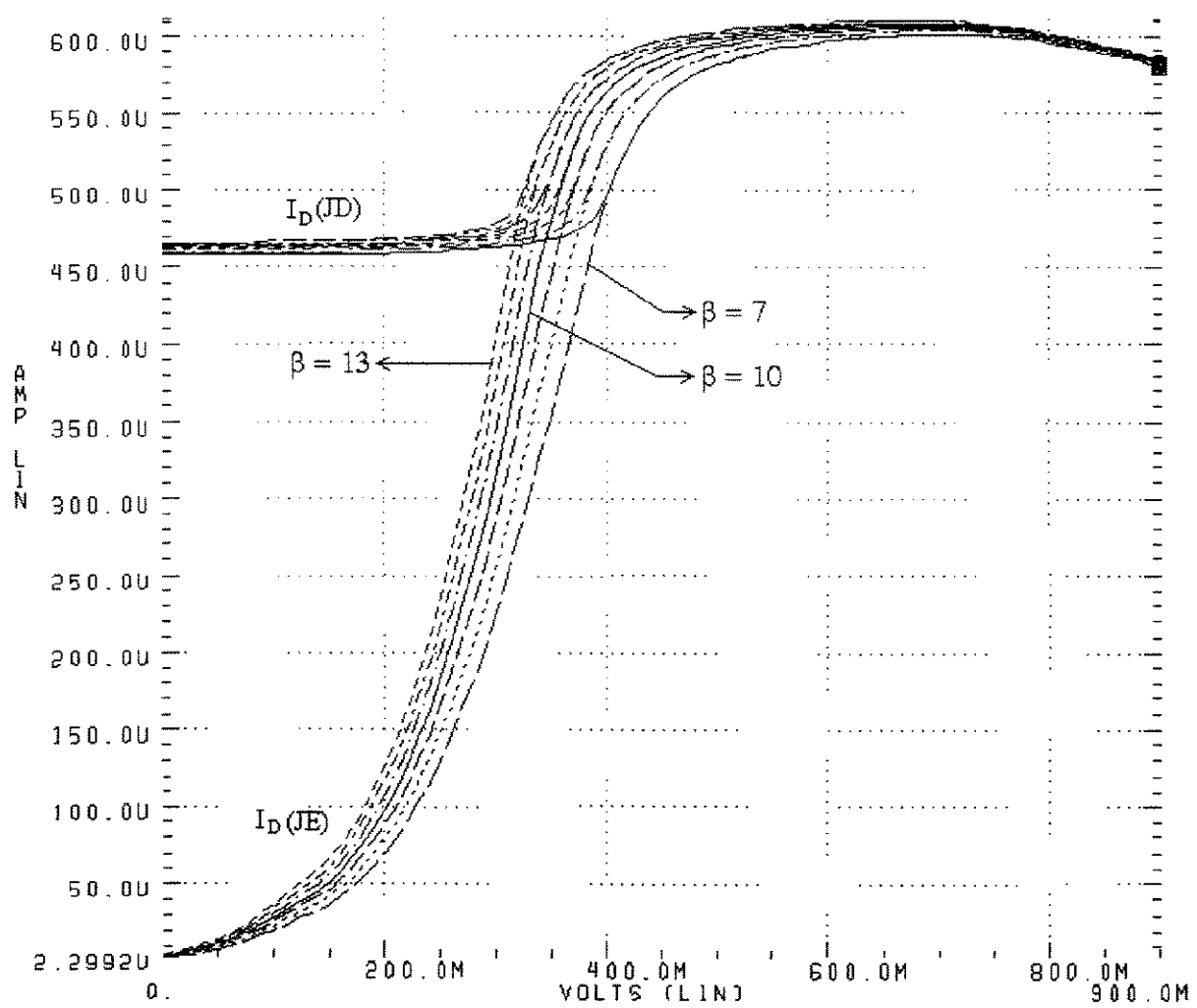


Figura 3.19 – Variação das correntes  $I_D$  dos FETs do inversor em função de  $V_{in}$ , tendo  $\beta$  como parâmetro. Com  $V_{in} = V_{OH}$ , as duas correntes sempre se igualam no estado estacionário.

A figura 3.20 mostra a variação da  $P_{D,st}$  do inversor em função de  $V_{in}$ , parametrizada em  $\beta$ . Como visto acima, enquanto  $V_{in} = V_{OL}$ , existe uma potência dissipada mesmo com  $J_E$  cortado, pois  $I_{D,D}$  não é zero. Um inversor consome mais potência enquanto  $V_{out} = V_{OL}$ , porque então  $V_{in} = V_{OH}$  e  $J_E$  está ligado, puxando maior corrente de  $J_D$ .  $P_{D,st}$ , para uma porta DCFL, deve ser especificada com a entrada em nível lógico alto, quando o consumo de potência atinge o seu valor máximo. Para uma mesma curva de transferência (um  $\beta$  fixo),  $P_{D,st}$  aumentou em 30%, para  $V_{in}$  variando de 0 a 0,7 V.

A  $P_{D,st}$  aumenta muito pouco com o incremento de  $\beta$ , pois  $W_D$  foi mantida constante ( $J_D$  comporta-se como uma fonte de corrente quase constante), e esta observação é visualizada na figura 3.21. Note-se que, para  $\beta$  variando de 5 a 25, o aumento correspondente de  $P_{D,st}$  foi de apenas 43  $\mu W$ , ou 3,6%. Os valores de potência da figura 3.21 foram tomados com  $V_{in} = V_{OH}$ , isto é, são os valores máximos do inversor.

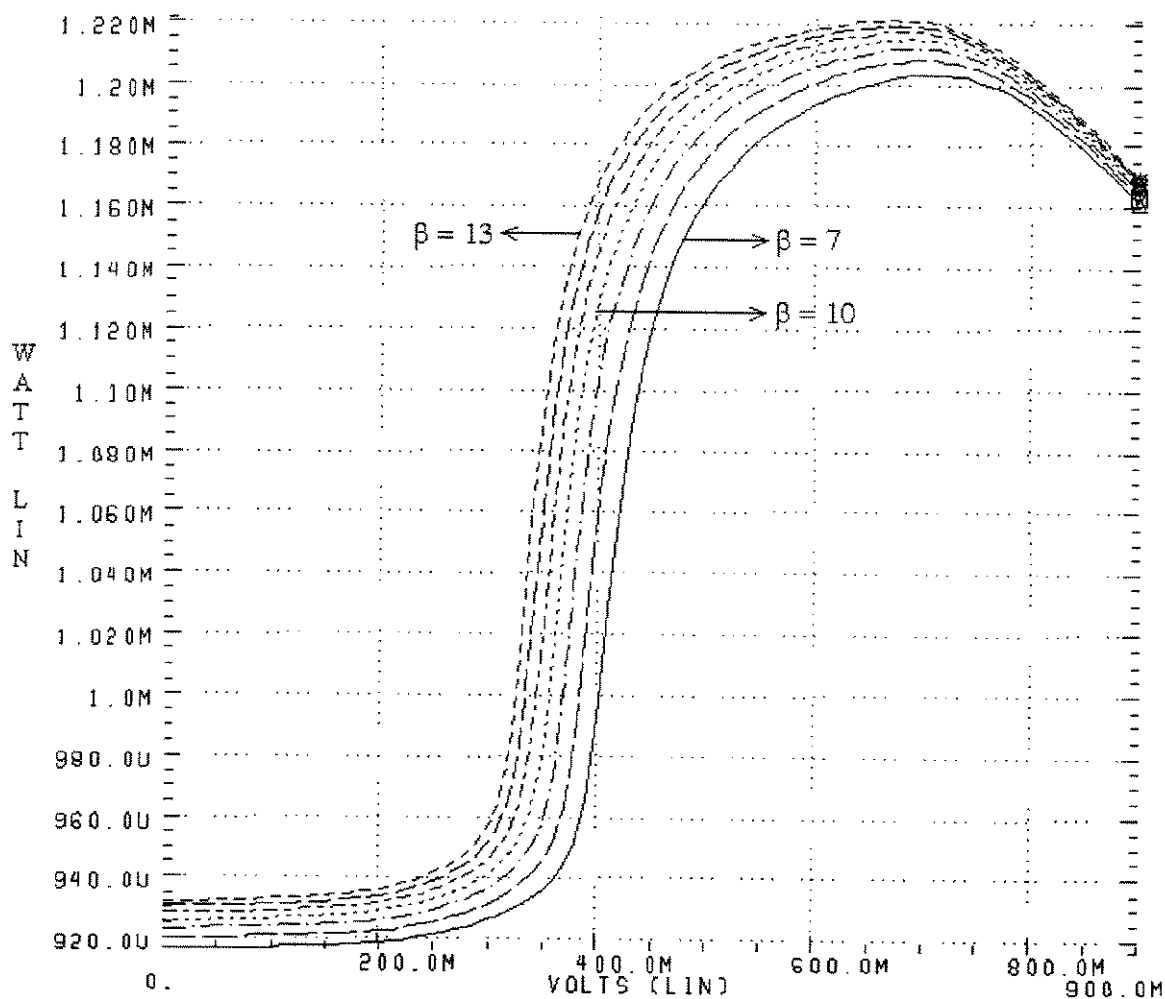


Figura 3.20 - Potência de um inversor DCFL em função de  $V_{in}$ , para vários valores de  $\beta$ .

As curvas das figuras 3.12 e da 3.15 até a 3.21 foram geradas pelos programas 2 e 3 do Apêndice 1, com pequenas adaptações. As condições de simulação para a obtenção desses resultados foram: parâmetros “típicos” do processo da Vitesse, temperatura de 25 °C, e o inversor em análise carregado com um *fan-out* de 2. Foi usado um  $FO = 2$  por ser característico da maioria dos circuitos digitais.

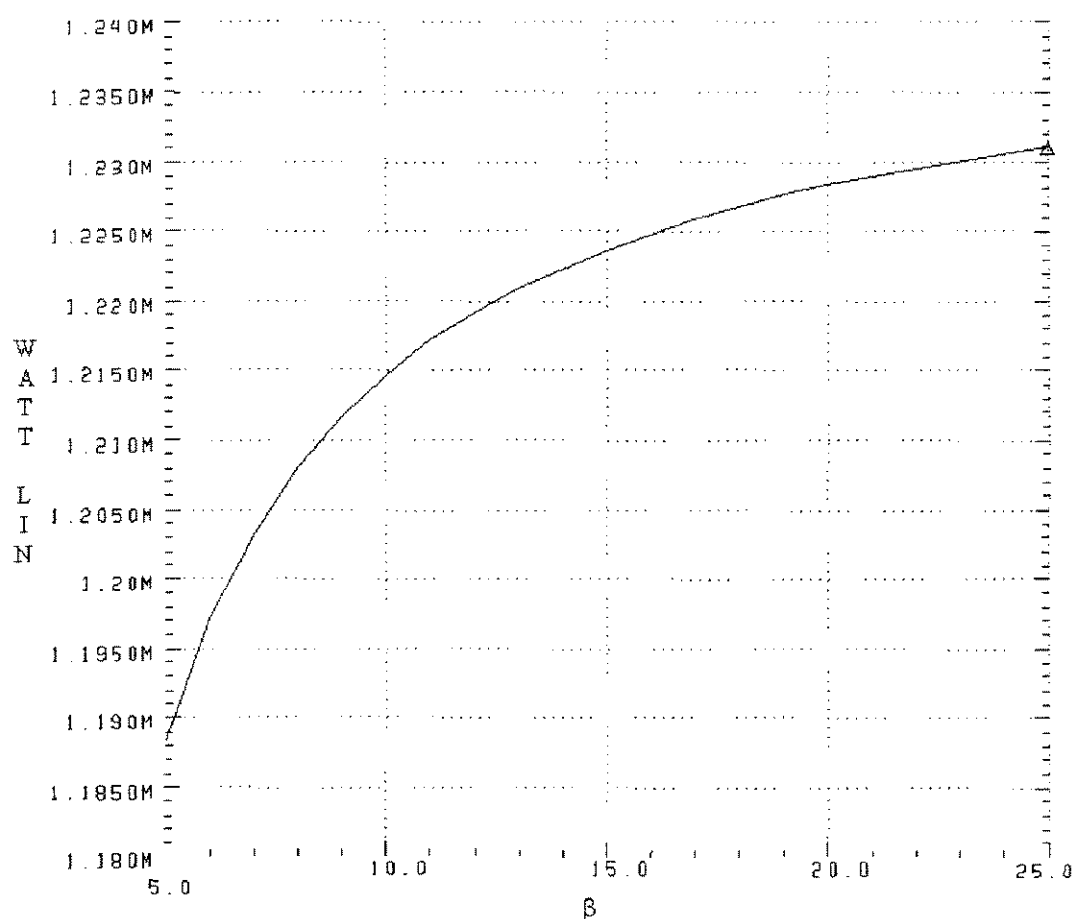


Figura 3.21 –Potência de um inversor DCFL em função de  $\beta$ , medida com  $V_{in} = V_{OH}$ .

### 3.2.3.2. Simulações Transientes.

Não se pode projetar um inversor com base em análises DC somente, pois são justamente as análises no domínio do tempo que mostram as respostas do circuito a sinais reais. Para simulações transientes, a porta lógica em análise não deve ser conectada diretamente à fonte de sinal de entrada (geralmente uma fonte de tensão do tipo **PULSE** no HSPICE), como é feito para simulações DC [11]. O sinal pulsado de entrada deve primeiro ser “conformado” por outra porta lógica, que seja típica da família em simulação, e a saída desta porta é que vai acionar a porta em análise; esse procedimento garante uma boa correlação entre os  $t_{PD}$ ’s simulados de uma porta lógica e os  $t_{PD}$ ’s reais no circuito-alvo. Esse é o método de se obter uma forma de onda de entrada realista em simulações transientes de portas digitais, e é adotado no circuito da figura 3.22, que será usado para determinar-se o  $t_{PD}$  do inversor X2.

A principal causa do atraso de propagação de um sinal por uma porta lógica são as diversas capacitâncias que devem carregadas e descarregadas nas transições de nível. Os fatores que contribuem para o  $t_{PD}$  são: o tempo de atraso intrínseco da porta em análise, devido a suas capacitâncias internas; a capacidade de acionamento (*drive*) da porta em análise; as capacitâncias parasitas de interconexão; e a capacitância total de entrada das portas acionadas (*fan-out*), que varia com a tensão.

Sendo assim, um carregamento realista na saída da porta em análise pode consistir de capacitores de carga fixos,  $C_L$ , e *fan-outs*. O carregamento por *fan-out* para X2 é proporcionado pelos inversores X4 e X5, e  $C_{L2}$  modela a carga parasita da interconexão; X3 modela um *fan-out* extra para X1. O circuito da figura 3.22 representa uma configuração completa de simulação, que pode ser alterada para simular diferentes intensidades de carregamento; como será demonstrado nesta seção, o efeito de  $\beta$  sobre o  $t_{PD}$  de uma porta lógica é dependente da capacitância de carga acionada por essa porta.

Os atrasos de propagação de um inversor,  $t_{PLH}$  e  $t_{PHL}$ , são linearmente proporcionais à  $C_L$  da saída, e à excursão  $V_{pp}$  e aos tempos de transição  $t_r$  e  $t_f$  da entrada [66], daí a importância de um pulso de entrada realista, com tempos de subida e de descida finitos, e de um carregamento realista na saída. Os sinais de entrada para as portas lógicas internas do CI não são pulsos perfeitos, e se o  $t_{PD}$  de **uma** porta for calculado com uma entrada idealizada deste tipo, seu valor será subestimado; e então, o  $t_{PD}$  de uma série de portas em cascata, calculado como a soma dos  $t_{PD}$ 's de cada porta, será super-otimista. Assim, no limite para uma entrada em degrau ( $t_r = t_f = 0$ ), os valores calculados de  $t_{PLH}$  e  $t_{PHL}$  são bem menores (mais otimistas) do que para uma entrada com  $t_r$  e  $t_f$  típicos de 100 ps. A dedução de expressões para  $t_{PLH}$  e  $t_{PHL}$  é feita em [66].

Outra maneira de realizar-se a análise transiente de portas lógicas é simulando um oscilador em anel, como em [9], mas para efeito de simulação também pode-se usar o arranjo da figura 3.22 [11]. O oscilador em anel tem a vantagem de ser uma estrutura de teste realizável na prática, que pode ser construído para medir-se o atraso de portas fabricadas fisicamente.

O sinal pulsado  $V_{in}$  para as simulações transientes deste Capítulo, tem as seguintes características:

$V_H = 0,7\text{ V}$ ,  $V_L = 0\text{ V}$ ,  $t_r = t_f = 100\text{ ps}$ ,  $\tau = 1\text{ ns}$  ( $f = 1\text{ GHz}$ ), e  $pw = 400\text{ ps}$ .

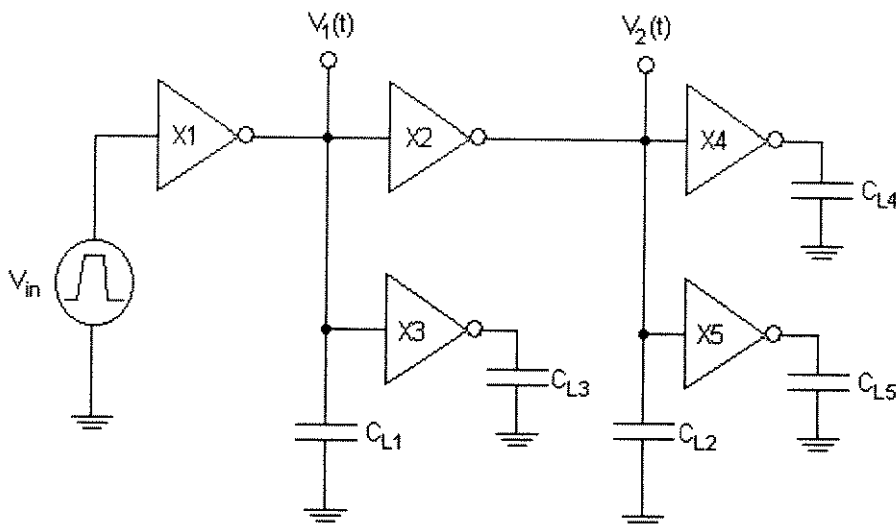


Figura 3.22 – Esquemático do circuito para simulação transiente do inversor X2.

O  $t_{PD}$  de X2 é medido pelo atraso entre seus sinais de entrada e de saída,  $V_1(t)$  e  $V_2(t)$ . É importante lembrar que, para a plotagem de funções de transferência DC, um arranjo como o da figura 3.22 não é adequado, devendo X2 ser ligado diretamente à fonte de tensão DC.

O programa 4 do Apêndice 1 faz várias simulações transientes do circuito da figura 3.22, variando o parâmetro  $\beta$  na faixa de 5 a 25, para que se possa avaliar os efeitos de  $\beta$  sobre o desempenho temporal de X2. Somente o  $\beta$  de X2 foi variado, os demais inversores foram mantidos sempre com  $\beta = 10$ . Experimentou-se três combinações de valores das  $C_L$ 's e do FO nas saídas de X1 e X2:  $C_L = 0$  fF e FO = 1;  $C_L = 50$  fF e FO = 1; e  $C_L = 100$  fF e FO = 2, modelando-se assim um caso otimista, um médio e um pessimista. O valor  $C_L = 0$  fF pode ser extrapolado para situações reais em que  $C_L$  é muito pequena.

A figura 3.23 mostra as formas de onda de  $V_2(t)$  das simulações com  $\beta$  indo de 7 a 13, para a primeira combinação de cargas acima. Para esta situação, observa-se que aumentar  $\beta$  diminui levemente o tempo de transição  $t_{HL}$  e aumenta  $t_{LH}$  e  $t_{PLH}$  do sinal de saída; além disso, os tempos de transição divergem, com  $t_{LH} > t_{HL}$ .

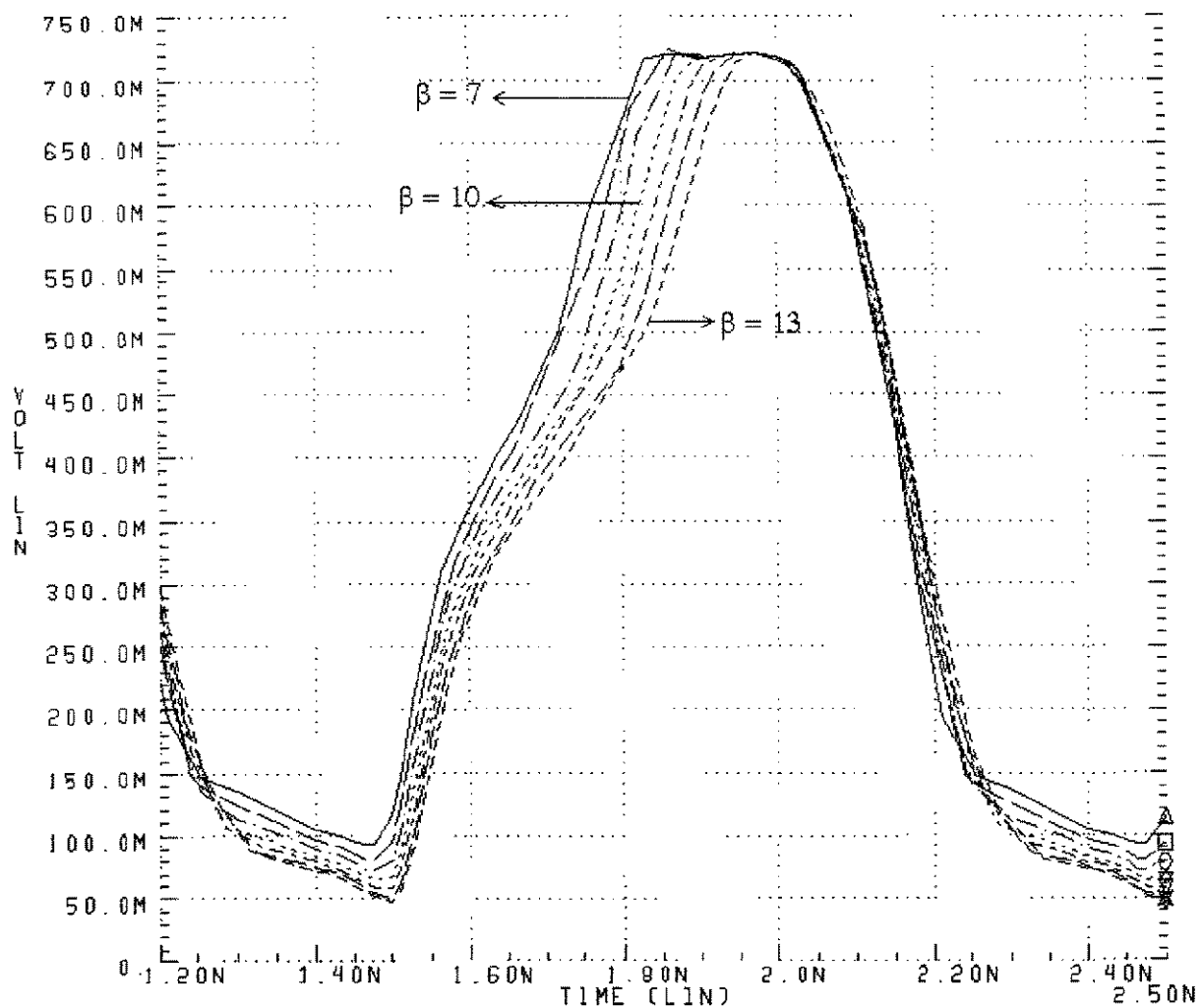


Figura 3.23 - Simulação transiente de um inversor com seu  $\beta$  variando de 7 a 13.

$t_{HL}$  é reduzido porque aumentar  $\beta$  significa aumentar  $W_E$ , aumentando a corrente do EFET para descarregar mais rapidamente  $C_L$ , o que leva a um menor  $\Delta t$  ( $= t_{HL}$ ), na equação (3.5). Porém, uma maior  $W_E$  também aumenta a  $C_{in}$  do próprio inversor, aumentando o  $t_r$  de  $V_1(t)$ , daí o aumento de  $t_{LH}$  e de  $t_{PLH}$ . O resultado líquido é um aumento de  $t_{PD}$  com o aumento de  $\beta$ .

A suposição de que o DFET seja uma fonte de corrente constante durante a região de transição não é rigorosamente verdadeira, porque à medida em que  $V_2(t) (=V_{DS,E})$  se aproxima de  $V_{OH}$ , o DFET sai da saturação, ao mesmo tempo em que o terminal de porta do EFET do estágio seguinte começa a solicitar uma corrente de entrada devido ao início da sua polarização direta. Nesta região, o carregamento das  $C_L$ 's é mais exponencial do que linear, o que se reflete na forma de onda de saída.

A figura 3.23 mostra  $V_2(t)$  para valores usuais de  $\beta$ ;  $\beta$ 's maiores degradariam completamente a saída temporal de X2. Os efeitos são mostrados na figura 3.24: o  $t_{PD}$  de X2 torna-se tão alto com  $\beta$ 's elevados que  $V_2(t)$  não consegue mais responder à entrada na frequência de 1 GHz.

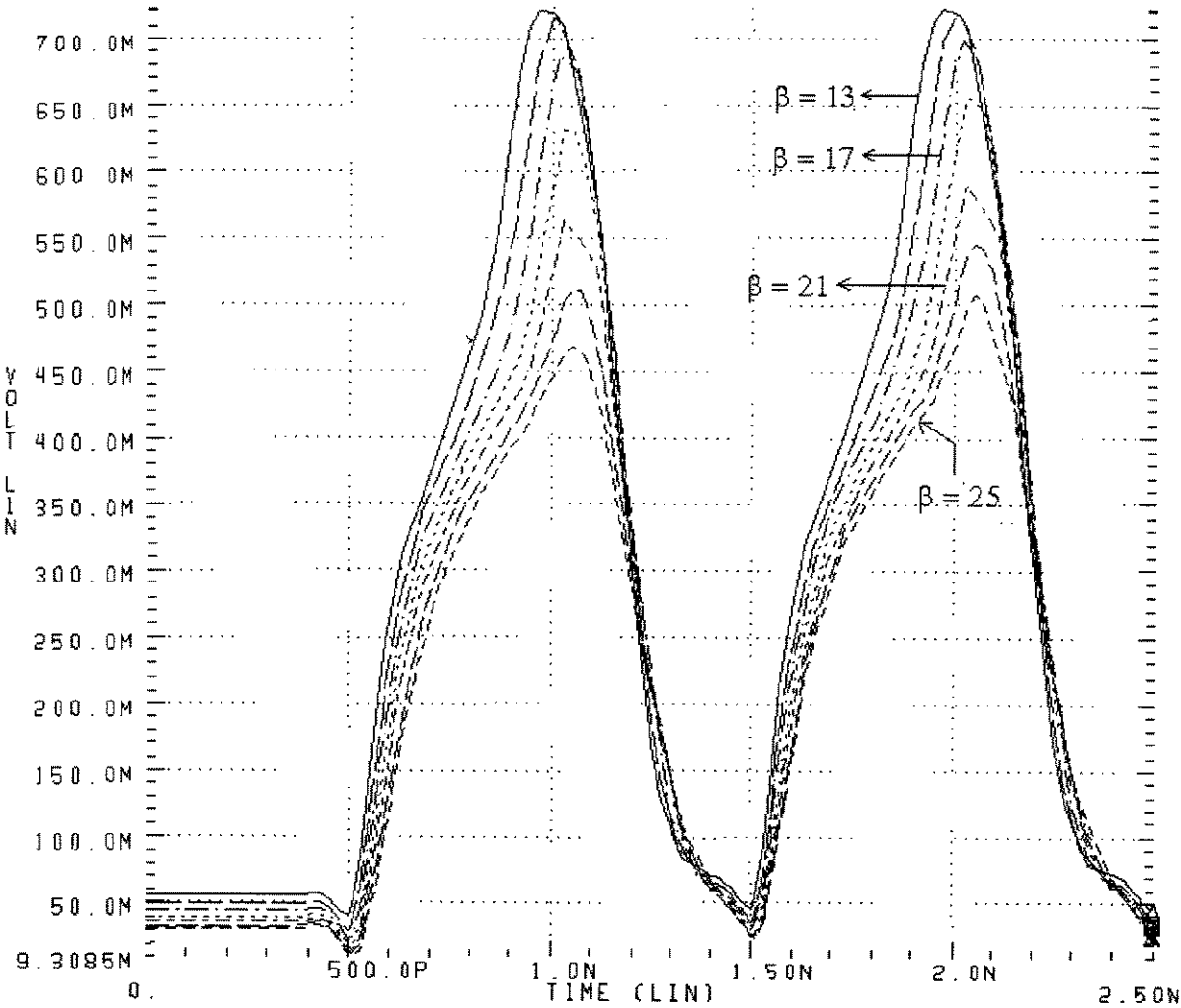


Figura 3.24 - Simulação transiente de um inversor com valores exagerados de  $\beta$ .

A figura 3.25 mostra três pares de sinais  $V_1(t)$  e  $V_2(t)$ , em três valores de  $\beta$ , para fins de comparação. As figuras 3.26, 3.27 e 3.28 apresentam os gráficos de  $t_{PD} \times \beta$  de X2, nas três condições de carregamento descritas acima. Elas foram geradas por adaptação do programa 2 (**volbeta.sp**), substituindo-se os valores de  $V_{OL}$  por valores de  $t_{PD}$ .

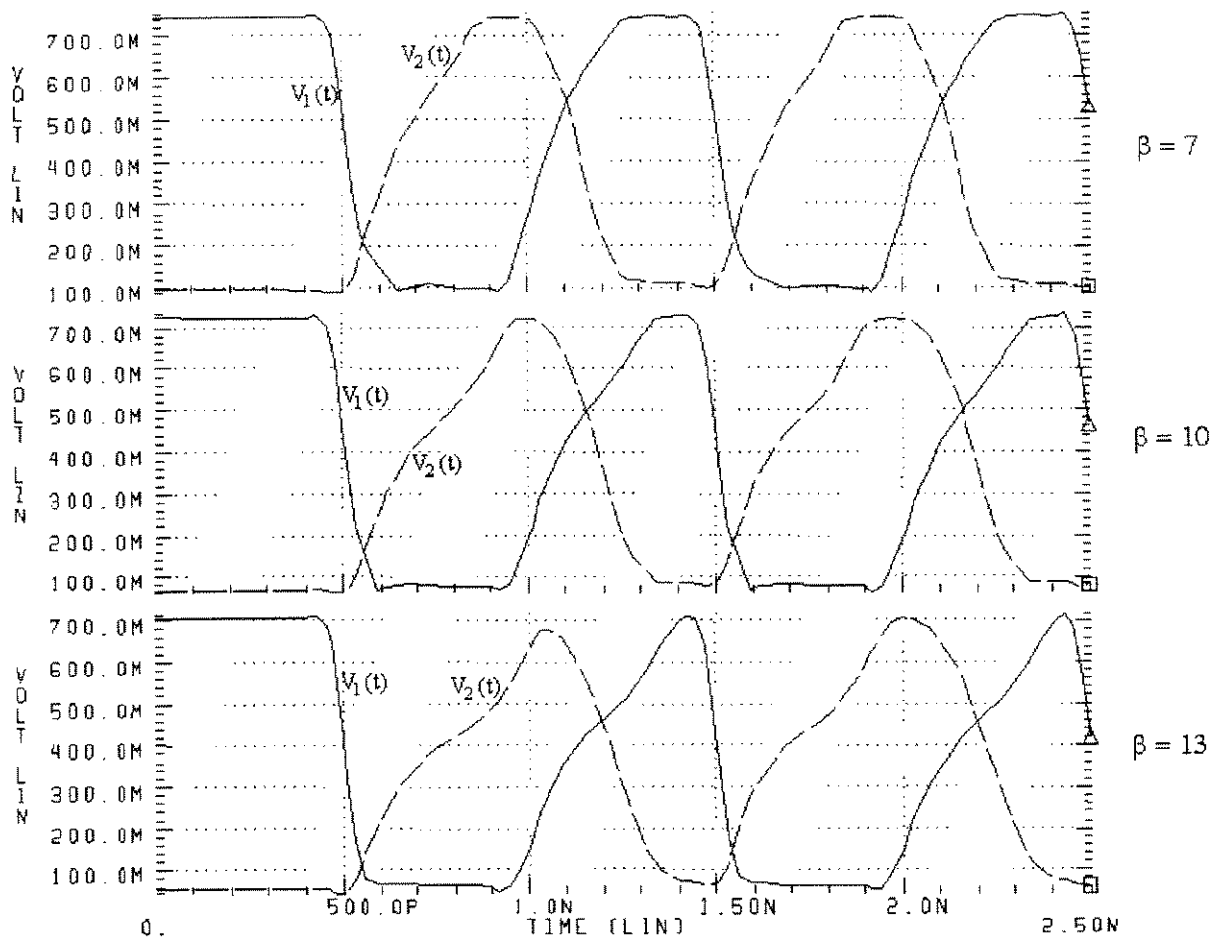


Figura 3.25 – Comparação das respostas transientes de um inversor, para três valores de  $\beta$ .

Para condições de carga média, como por exemplo,  $C_L = 50$  fF e  $FO = 1$ , verifica-se que o  $t_{PD}$  diminui inicialmente com o aumento de  $\beta$ , e depois torna a aumentar. Isso ocorre porque  $\beta \uparrow \Rightarrow W_E \uparrow, t_T \uparrow$ , mas um  $\beta$  alto também reduz  $t_{HL}$  e  $t_{PHL}$ . Até um certo valor de  $\beta$  (i.e., de  $W_E$ ), o efeito da redução de  $t_{PHL}$  se sobrepõe ao efeito do aumento de  $C_{in}$  e de  $t_{PLH}$ ; mas, a partir de um certo  $W_E$  e para uma dada  $C_L$ , a situação se inverte, e  $t_{PLH}$  passa a dominar, aumentando  $t_{PD}$ . Nesse momento, a porta do EFET do inversor já está tão larga que sua  $C_{in}$  já não permite mais uma compensação com a descarga mais rápida. O valor de  $\beta$  onde ocorre essa inversão de comportamento é próximo de 10 para a tecnologia empregada (Vitesse HGaAs-III).

A dependência de  $t_{PD}$  com  $\beta$  pode ser resumida da seguinte forma:

Para  $C_L$ 's pequenas:  $\beta \uparrow \Rightarrow t_{PD} \uparrow$

Para  $C_L$ 's médias, na faixa de 50 fF:  $\beta \uparrow \Rightarrow t_{PD} \downarrow$  até um certo valor de  $\beta$ , depois  $t_{PD} \uparrow$

Para  $C_L$ 's altas:  $\beta \uparrow \Rightarrow t_{PD} \downarrow$

Quanto maior  $C_L$ , maior o valor de  $\beta$  para o qual o  $t_{PD}$  se torna mínimo.



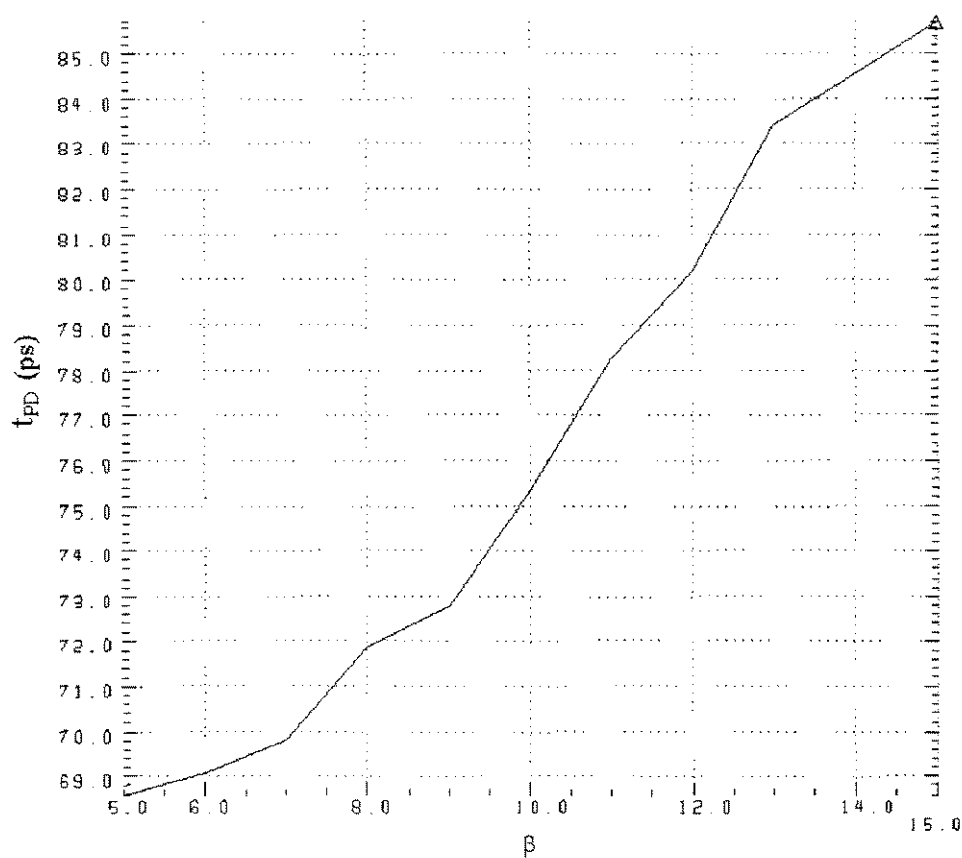


Figura 3.26 –  $t_{PD} \times \beta$  para uma carga capacitiva pequena na saída de um inversor.

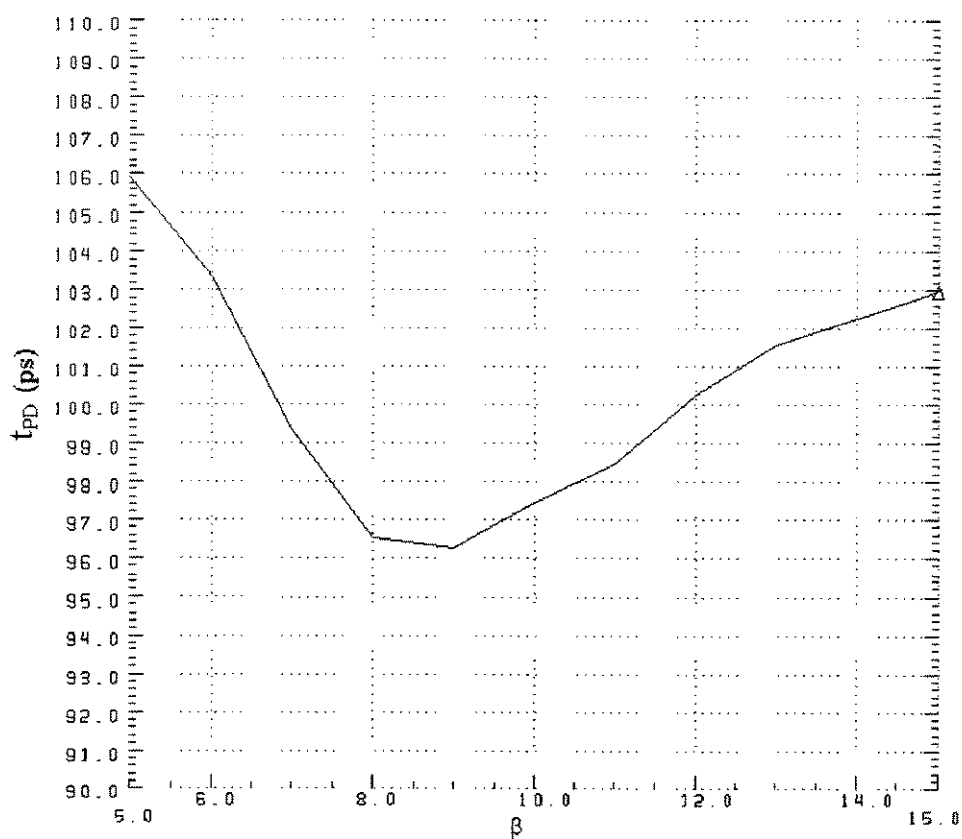


Figura 3.27 –  $t_{PD} \times \beta$  para uma carga capacitiva média na saída de um inversor.

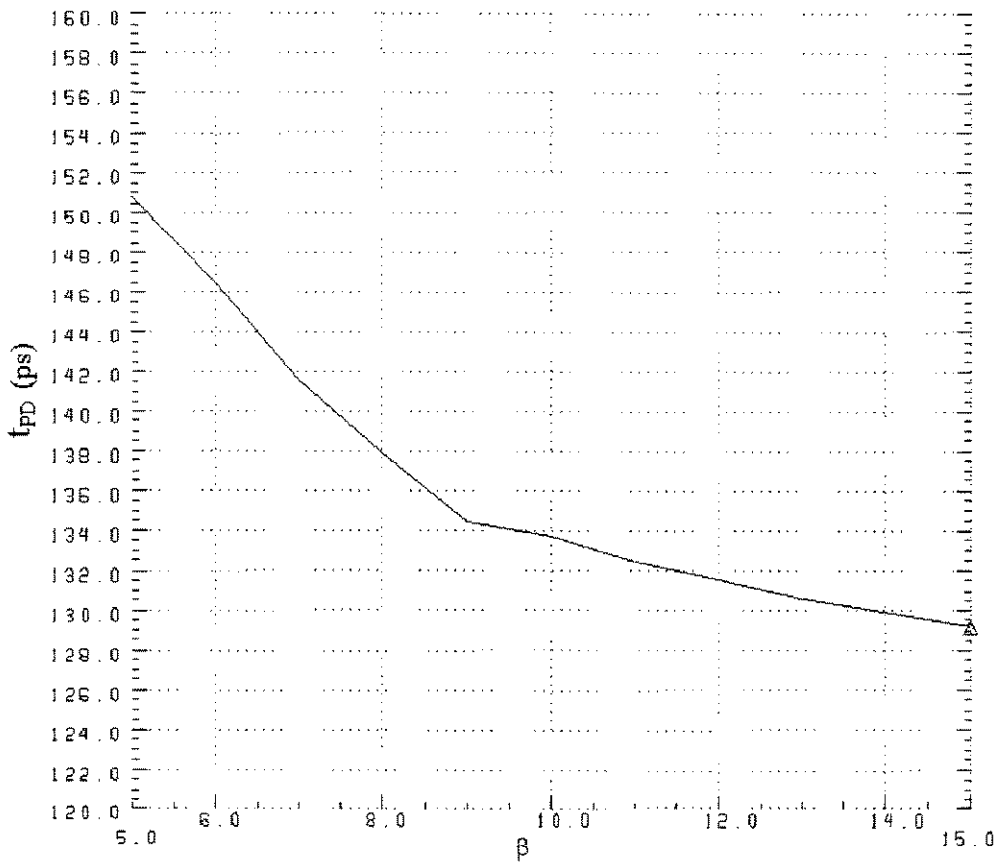


Figura 3.28 –  $t_{PD} \times \beta$  para uma carga capacitiva elevada na saída de um inversor.

Para o inversor DCFL, um valor de  $\beta = 10$  é a escolha ótima, provendo o melhor compromisso entre área, NM ([12], [78]), e desempenho transiente. Entretanto, é importante frisar que o valor ótimo de  $\beta$  para um inversor depende do valor de  $V_{LE}$ , e portanto deverá ser diferente para outra *foundry* ou outra tecnologia. Para  $V_{LE}$  menor, o  $\beta$  ótimo tende a ser menor [66]. Conclui-se que, em um projeto, os níveis lógicos desejados determinam a razão  $\beta$ , e um  $\beta$  até certo valor causa um bom aumento da NM do inversor e da sua confiabilidade, sem degradar demais a resposta transiente.

#### 3.2.4. A Influência de $W_D$ .

Até este ponto, o valor de  $W_D$  foi mantido fixo nas análises dos inversores; porém, ele é muito importante para o desempenho dinâmico de portas DCFL, pois  $W_D$  controla a corrente de dreno do DFET para a carga das diversas capacitâncias presentes na saída de uma porta.

A velocidade de uma porta lógica DCFL aumenta quando  $W_D$  é aumentada; e, para pequenos valores de  $W_D$ , o  $t_{PD}$  da porta é inversamente proporcional a  $W_D$ , linearmente [31]. Entretanto, a  $P_{D,st}$  de uma porta lógica aumenta sempre em proporção direta e linear com  $W_D$ , porque a  $I_D$  de um DFET fonte de corrente também é linearmente proporcional à sua largura  $W_D$ . Infelizmente, a redução correspondente de  $t_{PD}$  com  $W_D$  é sub-linear, ou seja, não é suficiente para compensar o aumento de  $P_{D,st}$ , resultando em aumento do produto D.P. Surge daí o compromisso atraso-potência:  $W_D \uparrow \Rightarrow t_{PD} \downarrow, P_{D,st} \uparrow$ . A figura 3.29 mostra a dependência linear de  $P_{D,st}$  com  $W_D$ , já validada na literatura, e a figura 3.30 mostra a variação de  $t_{PD}$  com  $W_D$ .

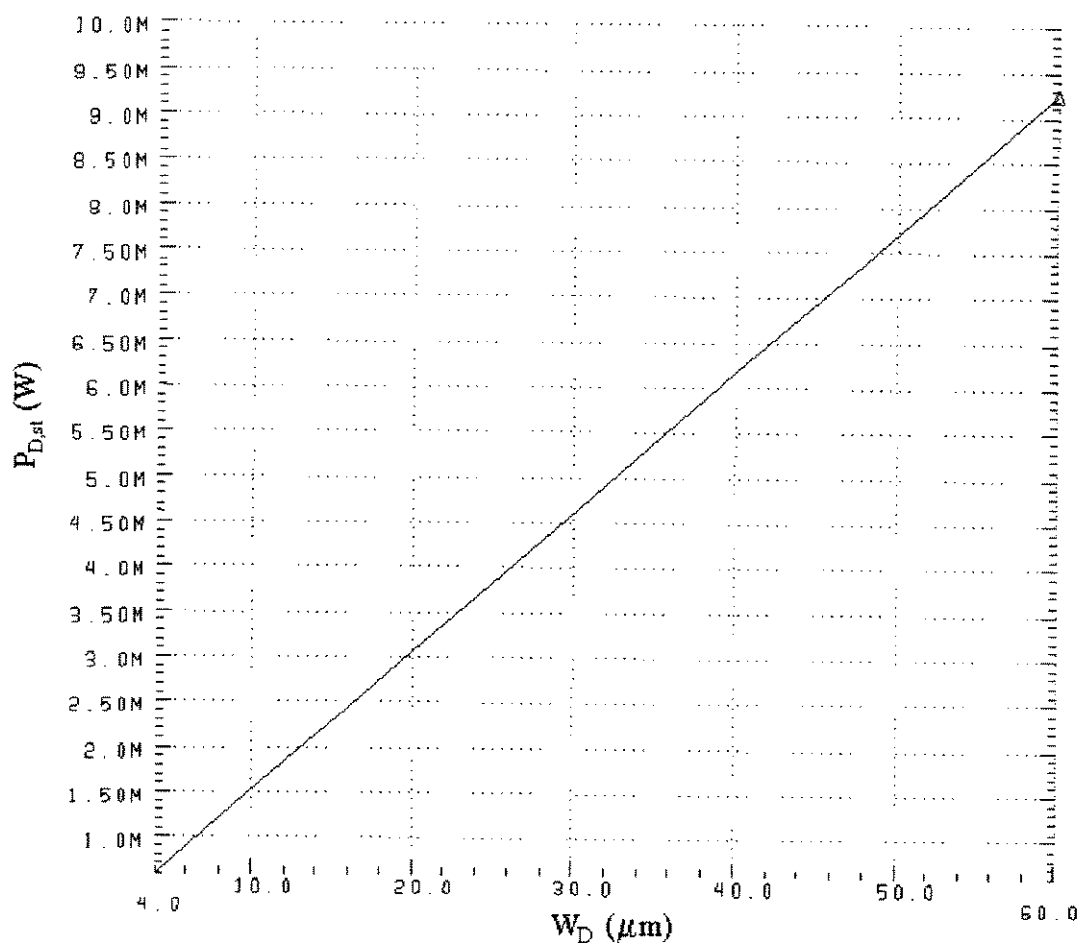


Figura 3.29 –  $P_{D,st}$  de um inversor *versus*  $W_D$ .

É certo que  $t_{PD}$  e  $P_{D,st}$  dependem muito mais fortemente do valor de  $W_D$  do que de  $\beta$  ou  $W_E$ . Por exemplo, sabe-se que reduzir o  $\beta$  de um inversor não diminui apreciavelmente o seu consumo de potência, isto somente pode ser conseguido com a redução de  $W_D$ .

A curva de  $t_{PD} \times W_D$  da figura 3.30 está em bom acordo com a literatura ([11], [31]). Se em um projeto, o  $t_{PD}$  das portas lógicas precisa ser menor do que um certo valor, as larguras  $W_D$  terão que ser aumentadas, inevitavelmente, dentro da região sub-linear de redução de  $t_{PD}$ . Portanto, a redução do produto D.P de uma porta lógica não pode ser uma tomada como meta de projeto, nem para portas lógicas e nem para *buffers*. Por exemplo, se for necessário um  $t_{PD} < 70$  ps para uma porta, será obrigatório usar-se um  $W_D > 20 \mu m$ ; para valores inferiores de  $W_D$ , esse  $t_{PD}$  não será realizável nem mesmo com um aumento de  $\beta$  [9].

O aumento de  $W_D$  com um  $\beta$  constante também reduz os tempos de transição  $t_{HL}$  e  $t_{LH}$ . O produto D.P cresce monotonicamente com  $W_D$  [9].

A figura 3.31 mostra as formas de onda de  $V_2(t)$  do circuito da figura 3.22, para  $W_D$  variando de  $4 \mu m$  a  $60 \mu m$ . Para esta análise, optou-se por variar  $W_D$  simultaneamente para todos os inversores, corrigindo também  $W_E$  de modo a manter  $\beta = 10$ . Verificou-se que esta alternativa é muito mais interessante do que varrer somente o  $W_D$  de X2, resultando em um melhor desempenho de X2 com baixos valores de  $W_D$ , o que é observado na figura 3.31. Varrer somente o  $W_D$  de X2 levaria a enormes descasamentos de carregamento por FO para

pequenos  $W_D$ 's, levando à falsa conclusão de que estes valores nunca deveriam ser empregados.

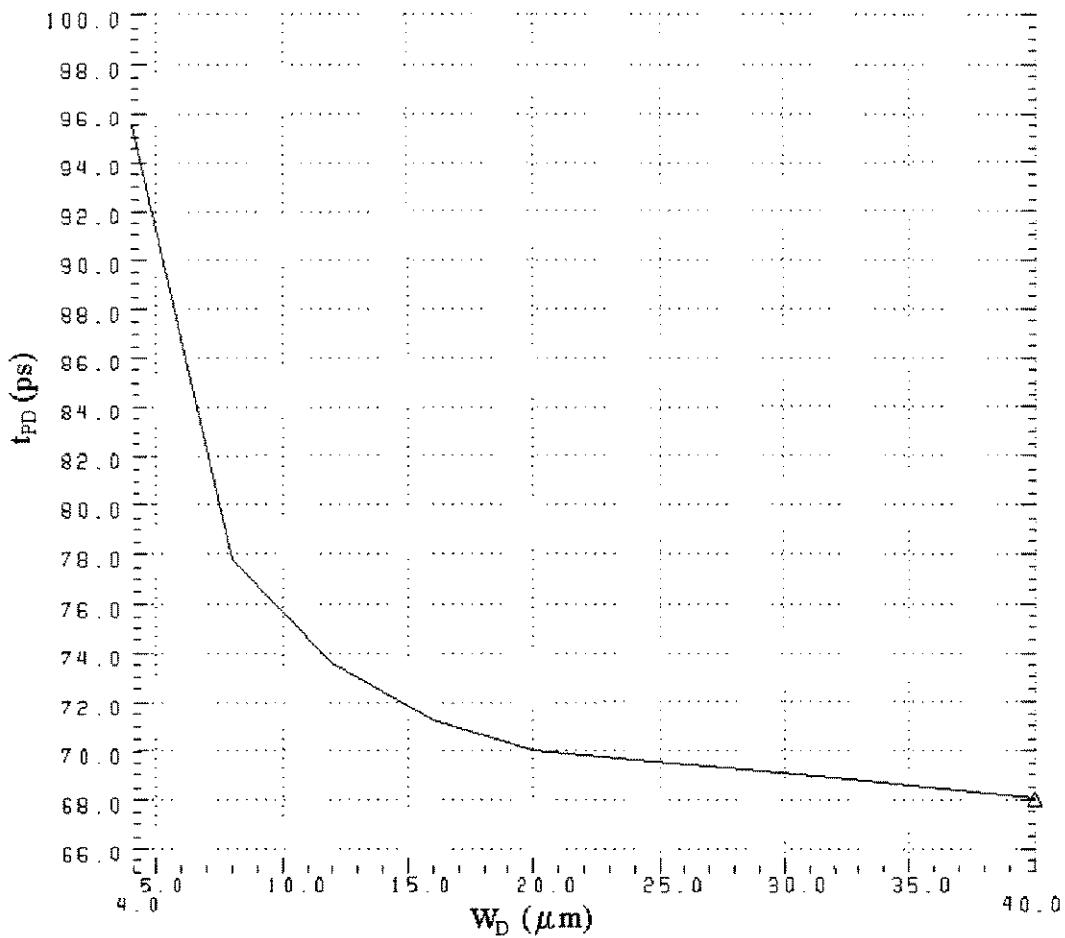


Figura 3.30 –  $t_{PD} \times W_D$ , para um inversor com  $\beta = 10$ ,  $C_L = 50$  fF e  $FO = 2$ .

A curva de transferência DC do inversor da figura 3.32, parametrizada em  $W_D$ , leva a crer que o valor de  $W_D$  em uma porta não tenha qualquer influência sobre os níveis lógicos dessa porta. Na operação do inversor com um sinal real de entrada, porém, observa-se que  $W_D$  influencia  $V_{OH}$ , porque ela estabelece a corrente de carga de  $C_L$ . Portanto, com  $\beta$  mantido constante,  $W_D \uparrow \Rightarrow V_{OH} \uparrow, V_{pp} \uparrow$ .

Um inversor INV1, com  $W_D = 4 \mu m$  e  $\beta = 10$ , tem  $V_{OH}$  menor do que o de um inversor INV2, que tenha  $W_D = 16 \mu m$  e  $\beta = 10$ . Aqui surge o compromisso de área  $\times$  potência  $\times$  velocidade: INV2 tem um melhor desempenho estático e dinâmico (menores atrasos e maior NM), mas ocupa uma área maior e consome quatro vezes mais potência.  $t_{PD1} \cong 2 \times t_{PD2}$ .

Concluindo, uma porta projetada com transistores mais largos chaveia mais rapidamente uma capacitância de carga, mas às custas de uma maior dissipação de potência. Quando um circuito contém portas operando em diferentes frequências, então  $W_g$ 's maiores são usados para as portas mais rápidas, enquanto  $W_g$ 's menores são usados para as portas que irão operar em frequências mais baixas.

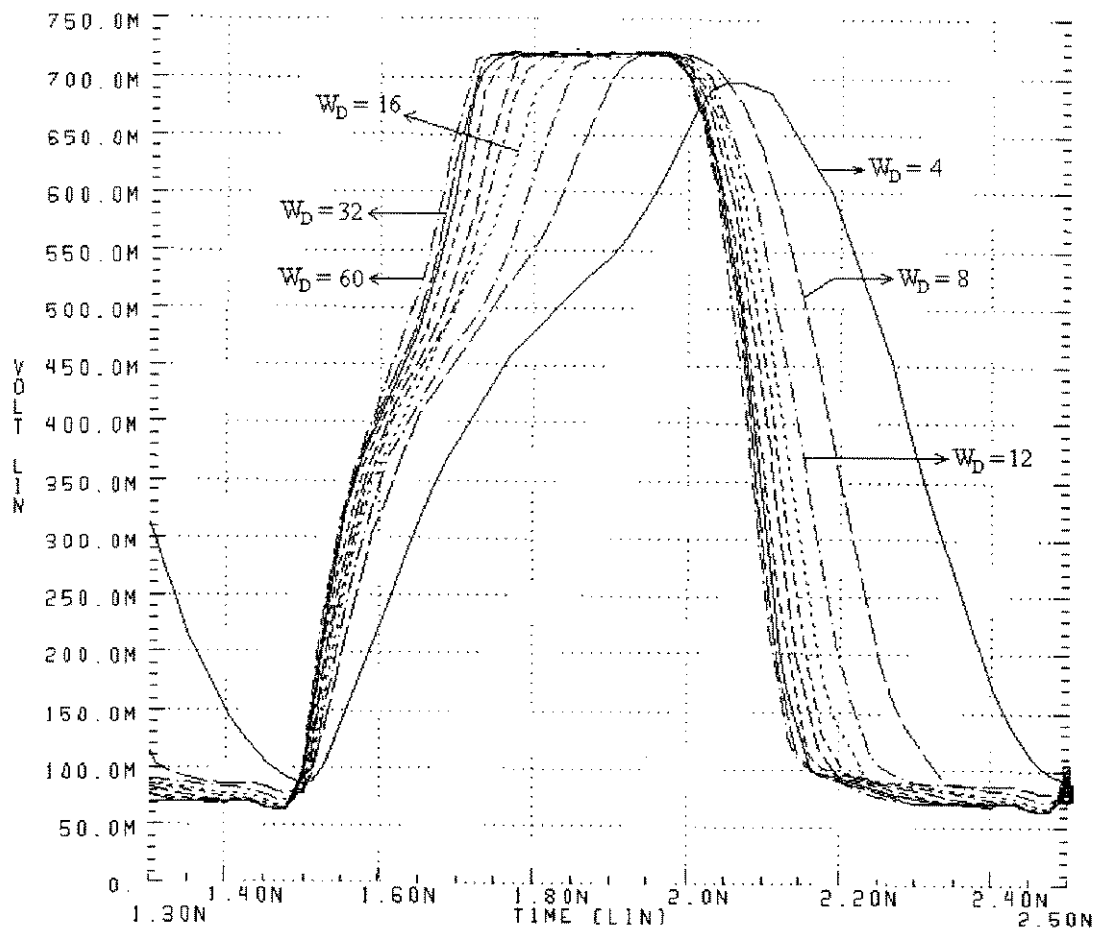


Figura 3.31 – Formas de onda de saída de X2 da figura 3.22, para  $W_D$  como parâmetro.

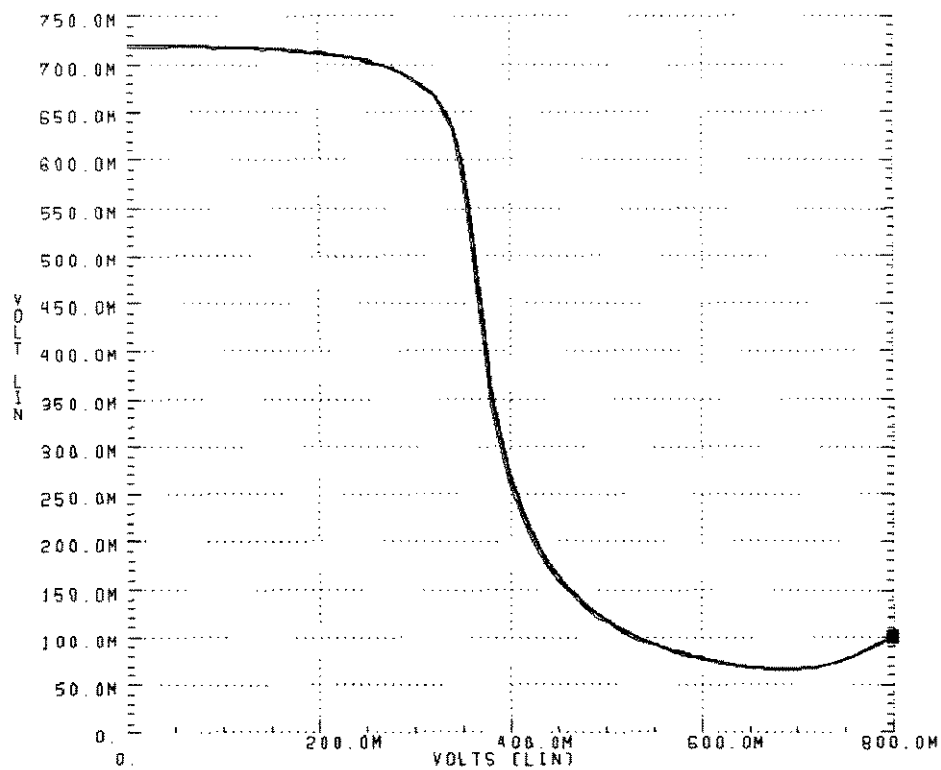


Figura 3.32 – Curvas de transferência de um inversor para  $\beta$  constante e igual a 10 e  $W_D$  variando de 4 a 60  $\mu\text{m}$ .

### 3.2.5. Variações da Temperatura, Processo, Fonte de Alimentação e *Fan-Out*.

Nesta seção, apresenta-se os resultados de simulações do inversor DCFL em temperaturas extremas de operação (0 e 85 °C), nos dois cantos do processo (*slow* e *fast*), com as tolerâncias máxima e mínima da fonte de alimentação  $V_{DD}$ , e para alguns valores de FO. Inicialmente considera-se somente o efeito da temperatura.

Quando a temperatura  $T$  aumenta,  $V_{OH}$  é reduzida, a NM de uma porta DCFL pode diminuir, e a dissipação de potência  $P_{D,st}$  aumenta. Em contrapartida, a velocidade dos circuitos DCFL aumenta proporcionalmente à temperatura. A figura 3.33 mostra curvas de transferência de um inversor, a figura 3.34 mostra curvas de potência e a figura 3.35 mostra uma resposta transiente, todas em quatro valores de  $T$ . O inversor foi simulado com  $FO = 2$  e  $C_L = 50$  fF.

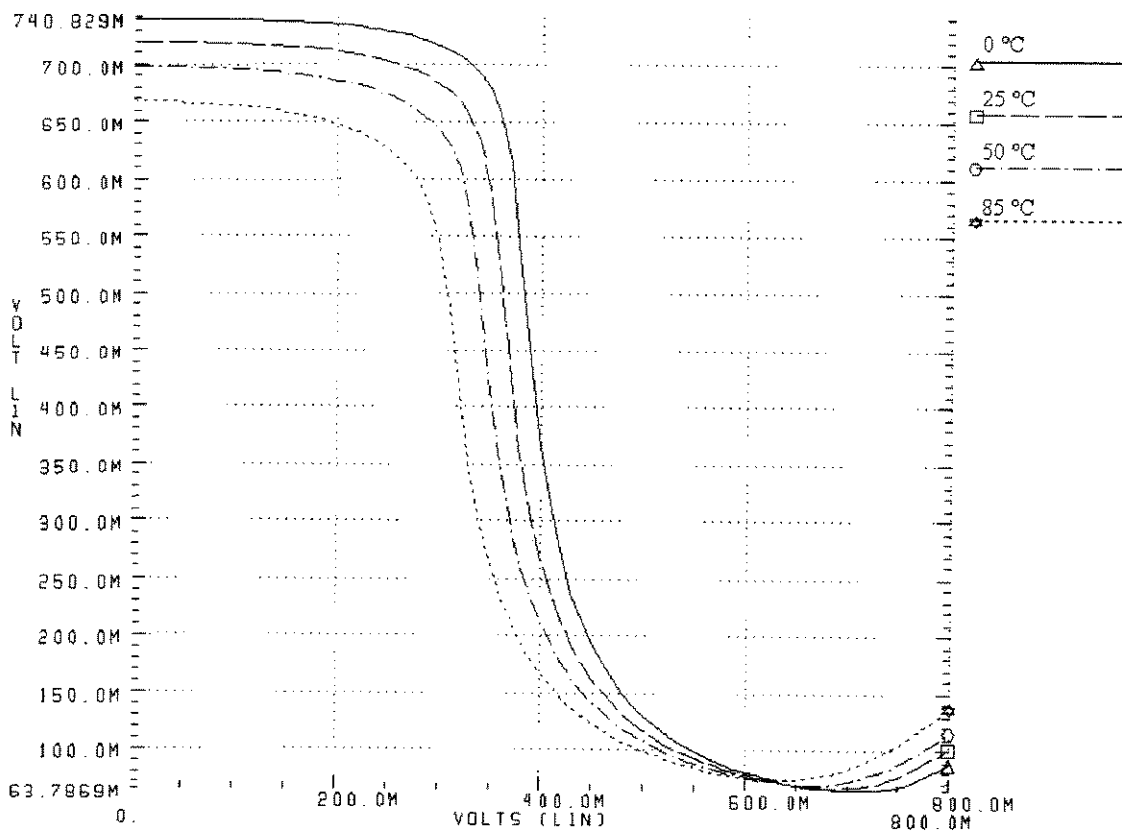


Figura 3.33 – Dependência da curva de transferência de um inversor com a temperatura.

$V_{OH}$  é reduzida significativamente (em 50 mV, para  $T$  indo de 25 °C a 85 °C, o que equivale a 7%), portanto qualquer circuito deve ser projetado de modo a ainda manter uma boa NM na sua máxima temperatura de operação. A causa do aumento de  $P_{D,st}$  e da velocidade com o aumento de  $T$  é que, para o DFET com  $V_{GS} = 0$  do inversor,  $I_{D,D}$  aumenta com  $T$  (conforme explicado no Capítulo 1), logo o aumento da temperatura tem o mesmo efeito de um aumento de largura do DFET, do ponto de vista de  $P_{D,st}$  e de  $t_{PD}$ . Mas na verdade  $W_D$  continua a mesma, por isso o aumento de  $I_{D,D}$  causa um aumento da queda de tensão na resistência  $r_{ds}$  do DFET, reduzindo  $V_{OH}$ .  $P_{D,st}$  é linearmente proporcional a  $T$ .

As figuras 3.33 e 3.34 são geradas pelo Programa 7, e a figura 3.35 é gerada pelo Programa 8, ambos listados no Apêndice 1.

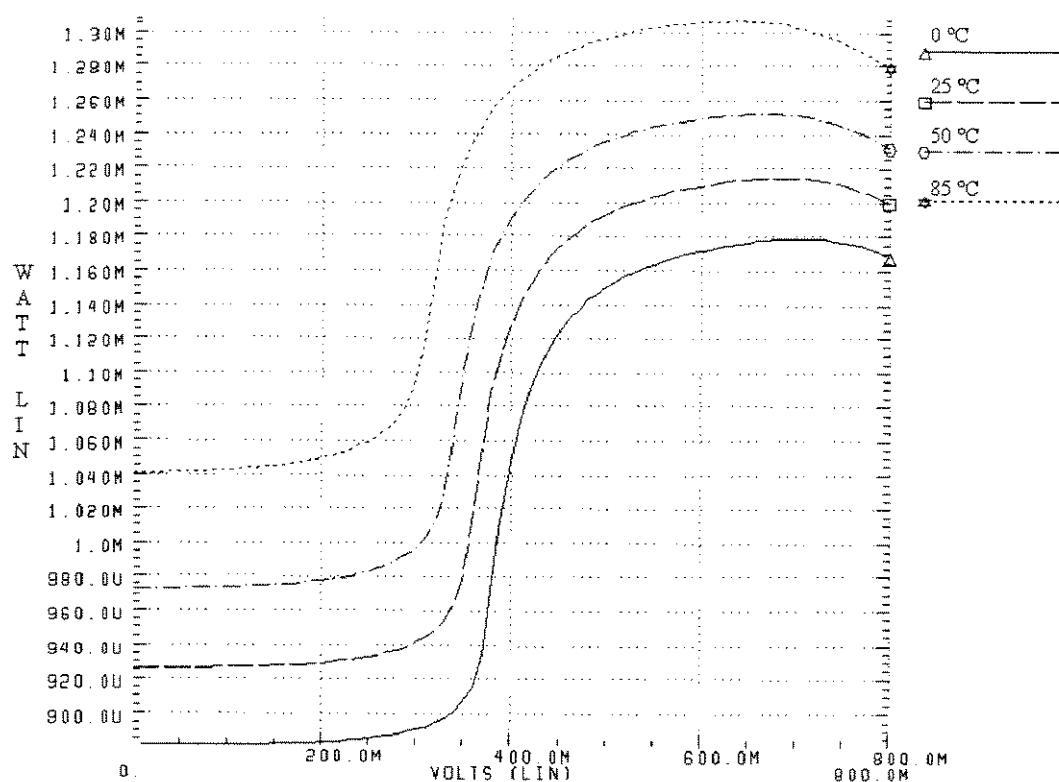


Figura 3.34 – Consumo de potência  $P_{D,st}$  do inversor para algumas temperaturas de operação.

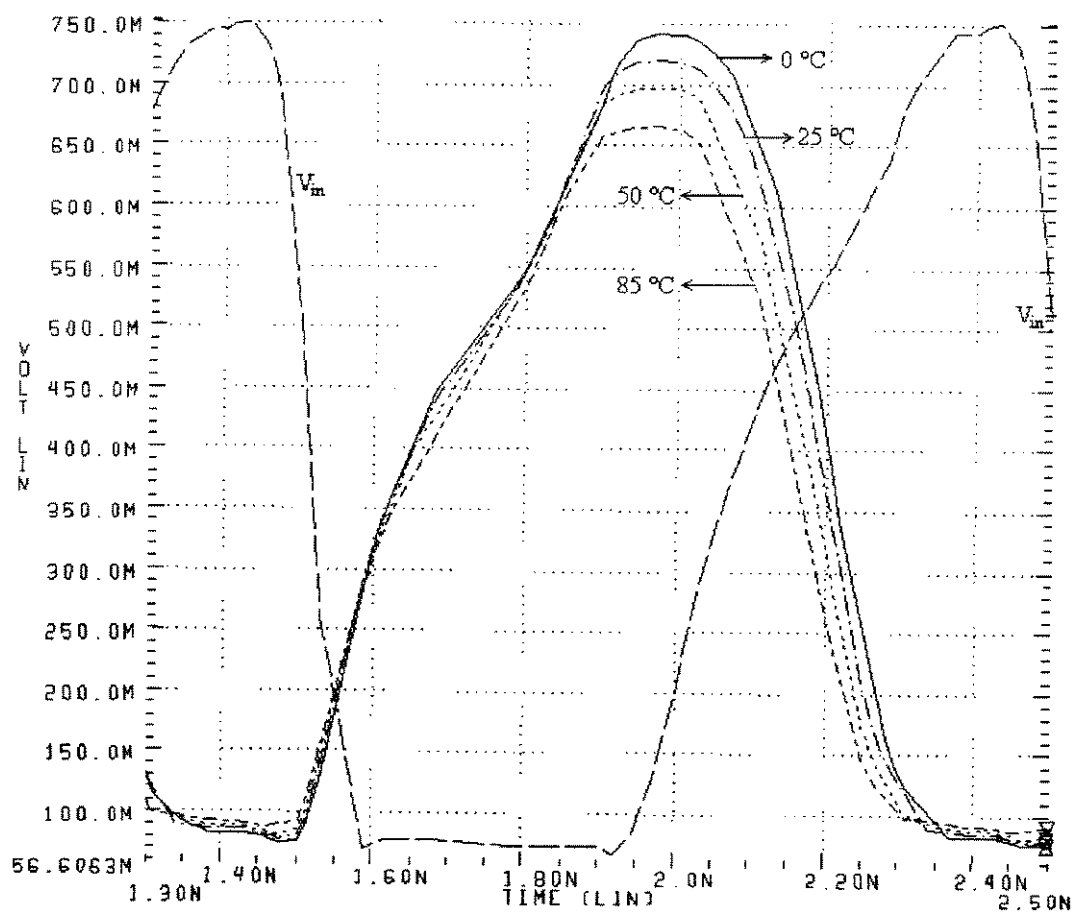


Figura 3.35 – Variação da resposta transiente de um inversor com a temperatura.

A Vitesse não permite ao projetista controlar os valores de  $V_t$  dos MESFETs; mas  $V_{t,E}$  e  $V_{t,D}$  apresentam uma variância relativamente ampla em torno de seus valores nominais (de 60 mV no máximo [12]), devida a tolerâncias inerentes ao processo de fabricação, sendo que ambas as  $V_t$ 's variam no mesmo sentido. A variação de  $V_t$  dos FETs afeta a tensão de limiar do **circuito** inversor DCFL. Por exemplo, quando  $V_{t,E}$  aumenta,  $I_{D,E}$  diminui para um mesmo valor de  $V_{in}$ , e assim a curva de transferência da figura 3.12 desloca-se para a direita, porque uma maior  $V_{in}$  é necessária para produzir a mesma  $V_{out}$ . O efeito é o mesmo de uma redução de  $\beta$ , com um aumento do limiar lógico do inversor DCFL e redução de NM.

Essa sensibilidade da NM e do limiar do circuito para com as  $V_t$ 's dos FETs é o grande problema da lógica DCFL. Por exemplo, para um circuito fabricado no canto lento (*slow corner*) do processo, a  $V_t$  dos FETs é maior, já que o atraso aumenta e o consumo de potência diminui para  $V_t$ 's maiores. Já para simular os efeitos de um decréscimo de  $V_t$  no processo, são usados os modelos *fast* dos FETs. A influência de  $V_t$  sobre  $P_{D,st}$  é mais forte do que a da temperatura. Resumindo, pode-se dizer que:

$$\text{slow} \rightarrow \text{typical} \rightarrow \text{fast} \Rightarrow V_t \downarrow \Rightarrow V_{pp} \downarrow, NM \downarrow \Rightarrow t_{PD} \downarrow, P_{D,st} \uparrow \quad (3.7)$$

As variações de  $V_t$  são consideradas no projeto por meio do arquivo de extremos do processo da Vitesse, com três bibliotecas com valores de parâmetros dos MESFETs. A biblioteca típica (*typical*), que vem sendo usada nas simulações até agora, contém os transistores nominais. É usada para confirmar a funcionalidade do circuito, e para um primeiro dimensionamento. O uso da biblioteca *fast* a 85°C modela a maior velocidade de chaveamento e o pior caso de dissipação de potência. E a simulação do circuito com o uso da biblioteca *slow* a 0°C, modela o pior caso de velocidade. A tolerância da fonte de alimentação  $V_{DD}$ , neste projeto, é de 10%.

A variação dos parâmetros dos MESFETs do caso típico para o rápido é considerável e maior do que as variações do caso típico para o lento, indicando que deve-se deixar uma margem de segurança na escolha das dimensões dos FETs. As figuras 3.36 e 3.37 mostram a variação das curvas de transferência de tensão e de potência, nos dois extremos do processo; essas figuras foram geradas pelo Programa 9 do Apêndice 1. Quando a curva de transferência vai para a direita, o circuito fica mais lento; quando vai para a esquerda, o circuito fica mais rápido.

As figuras 3.38 e 3.39 mostram curvas de transferência e suas imagens especulares, para a estimativa das NMs pelo método do máximo quadrado; segundo este método, o ganho de NM do processo típico para o lento é muito pequeno, ficando em torno de 10 mV. Mas a perda de NM do processo rápido em comparação com o típico é considerável, aproximadamente 45 mV. As figuras 3.38 e 3.39 foram geradas no HSPICE, inserindo-se os valores numéricos de  $V_{in}$  versus  $V_{out}$  no Programa 2 do Apêndice 1. O melhor caso de NM, no processo lento a 0 °C, é de 200 mV. A NM mínima ocorre para o processo *fast* a 85 °C, e vale 140 ~ 150 mV.

As figuras 3.40 e 3.41 apresentam respostas transientes de um inversor em 1 GHz, confrontando inversores do processo típico com o lento, e do típico com o rápido; o FO usado foi de 2 e  $C_L = 50$  fF. Do ponto de vista do desempenho transiente, quanto maior  $V_{DD}$ , maior a diferença de potencial aplicada entre os terminais do DFET, aumentando sua  $I_D$  e diminuindo os tempos de atraso da porta lógica. Para  $V_{DD}$  com um valor abaixo do nominal, o inversor fica mais lento. As figuras 3.40 e 3.41 foram geradas pelo Programa 10 do Apêndice 1.



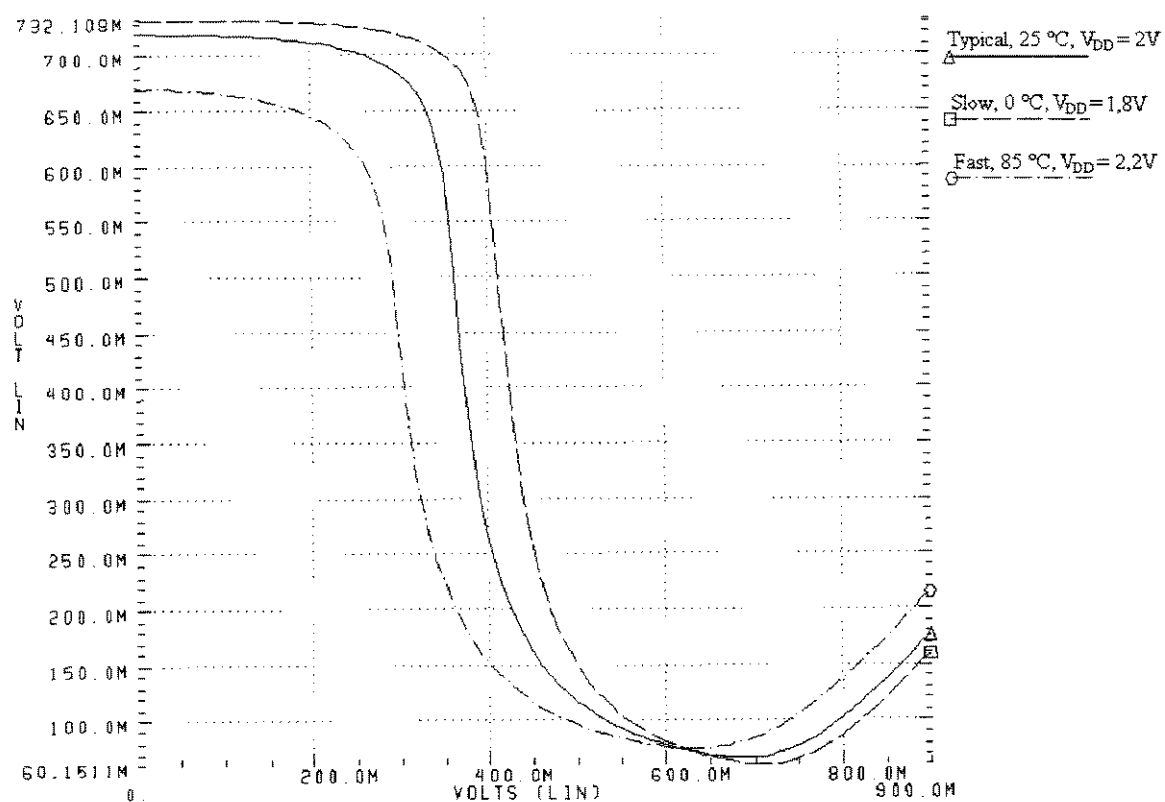


Figura 3.36 – Curva de transferência de um inversor, em condições típicas e nos dois extremos de operação.

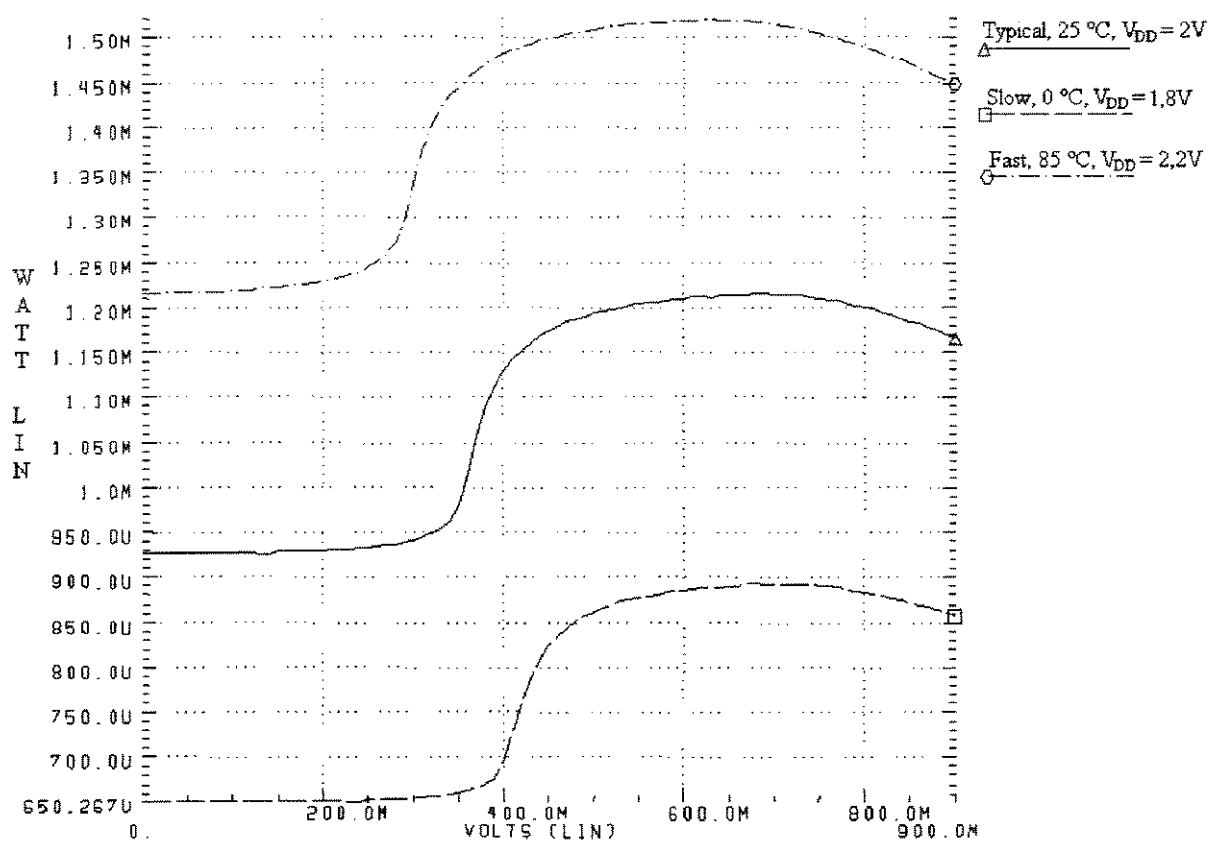


Figura 3.37 – Curvas de dissipação de potência de um inversor, em condições típicas e em dois extremos de operação.

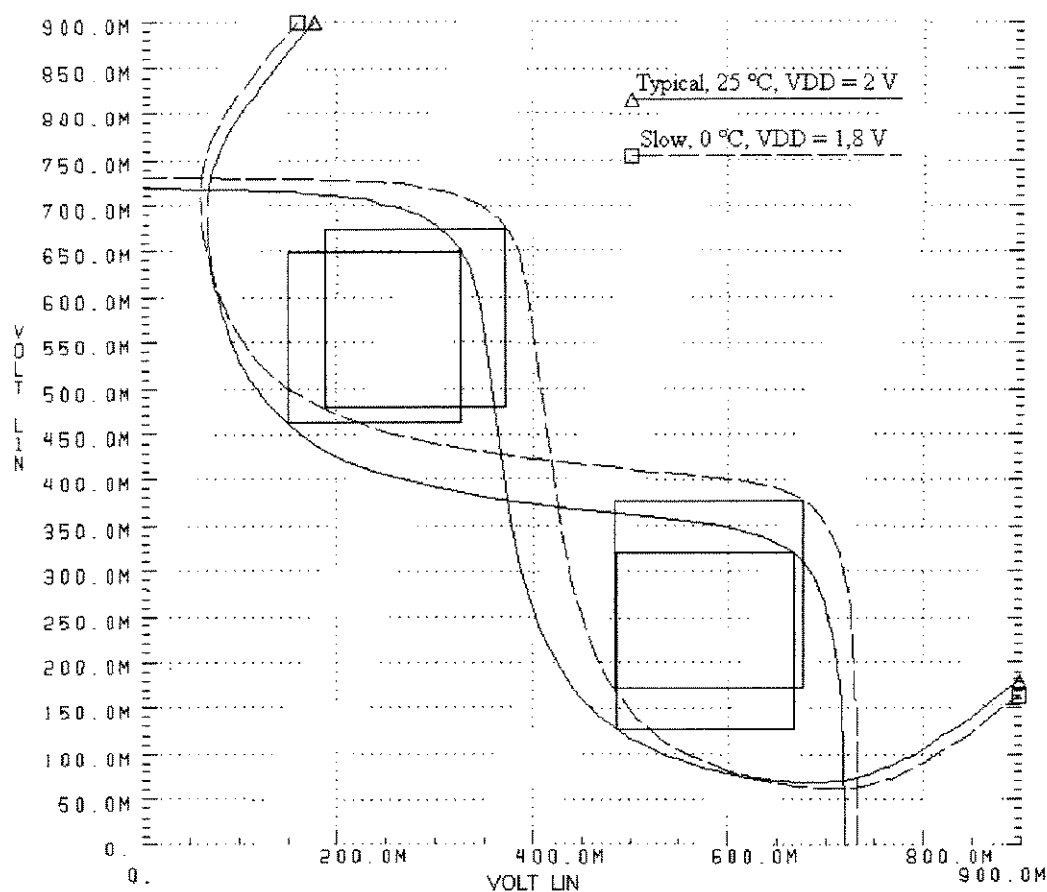


Figura 3.38 – Gráfico para a comparação das NMs nos cantos típico e lento do processo.

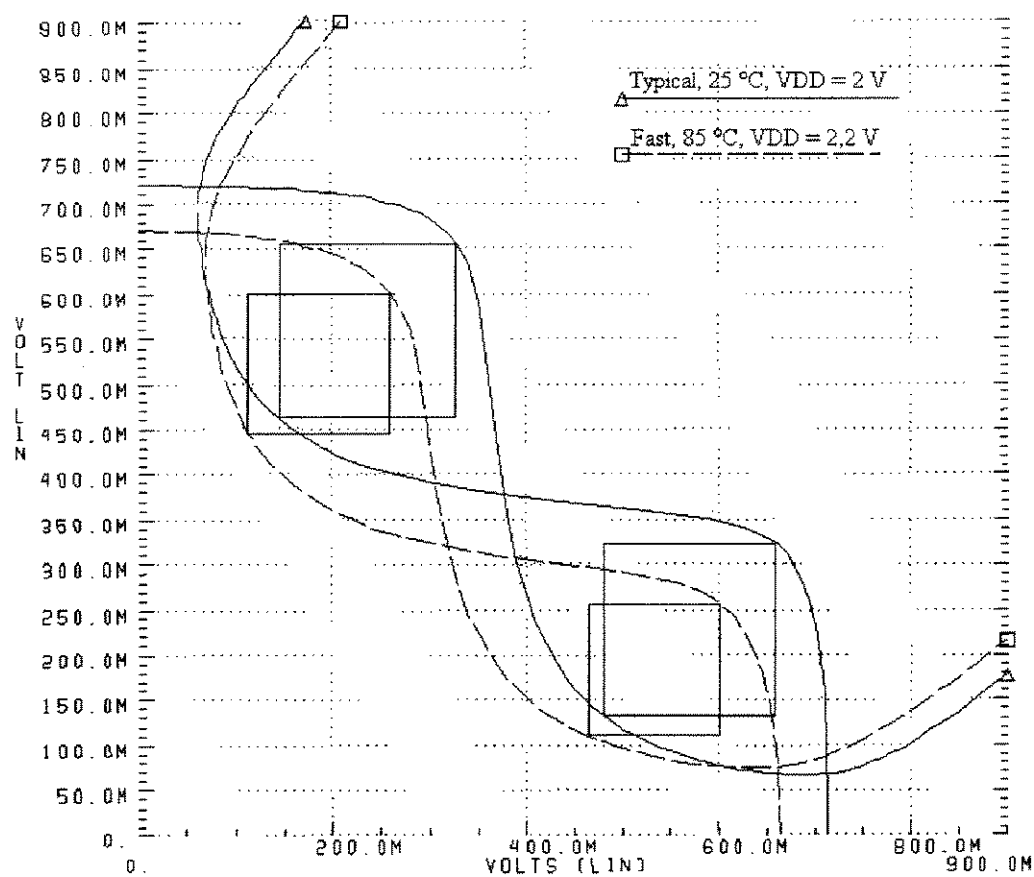


Figura 3.39 - Gráfico para a comparação das NMs nos cantos típico e rápido do processo.

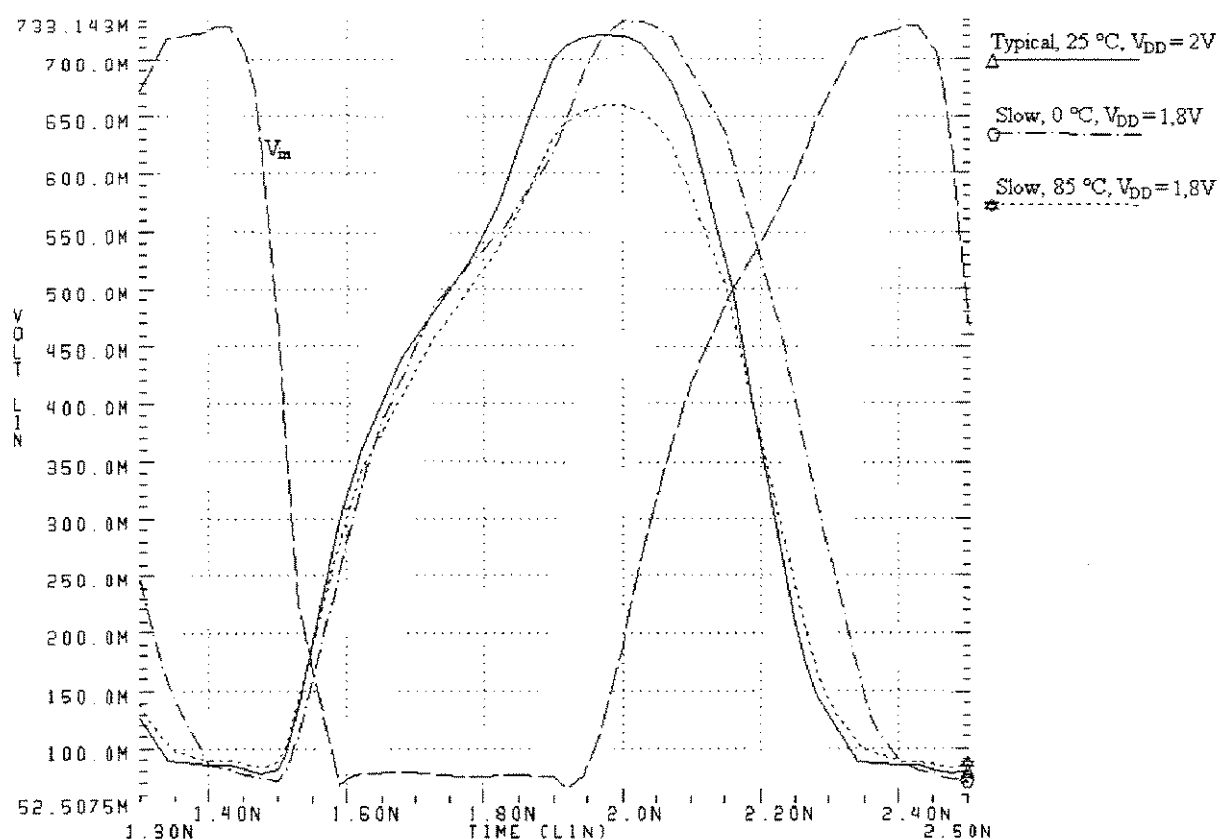


Figura 3.40 – Comparação das respostas de um inversor do canto *slow*, em duas temperaturas, com a resposta de um inversor típico.

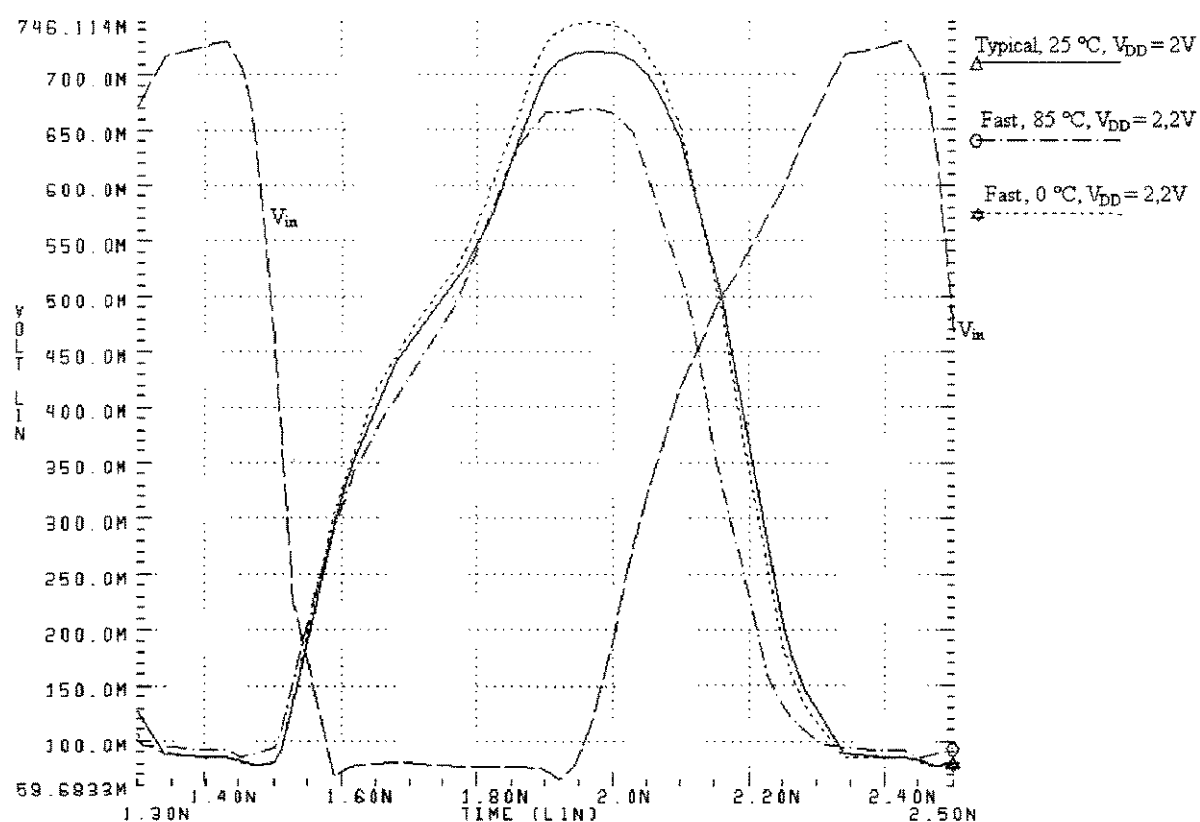


Figura 3.41 - Comparação das respostas de um inversor do canto *fast*, em duas temperaturas, com a resposta de um inversor típico.

Finalmente, analisou-se o efeito do *fan-out* sobre os desempenhos DC e transiente de um inversor. Como foi dito no Capítulo 2, as portas DCFL não conseguem acionar *fan-outs* elevados sem uma degradação considerável no nível do sinal de saída e em  $t_{PD}$ . Para um *fan-out* igual a FO, a carga capacitiva  $C_L$  na saída da porta passa a valer  $C_L \cdot FO$ .

O aumento do FO reduz a excursão  $V_{pp}$  do sinal de saída, porque reduz o nível de  $V_{OH}$ . Assim, se  $FO \uparrow \Rightarrow NM \downarrow$ . Isto pode ser melhor visualizado na curva de transferência DC da figura 3.42. Entretanto, a  $P_{D,st}$  de uma porta praticamente não depende de  $C_L$  ou FO [9]; ocorre um pequeno aumento de  $P_{D,st}$  para a saída em nível lógico alto, o que é mostrado na figura 3.43. O aumento de  $C_L$  tende a aumentar diretamente a  $P_{D,dyn}$ , porém esse aumento é compensado pela redução de  $f$ , à qual  $P_{D,dyn}$  também é proporcional (ver equação (3.1)). As curvas das figuras 3.42 e 3.43 foram obtidas para um inversor fabricado no canto *fast* do processo, com  $W_D = 8 \mu m$  e  $\beta = 10$ , operando a  $85^\circ C$  e com  $V_{DD} = 1,8 V$ . Ou seja, a simulação DC do inversor foi feita no pior caso de  $V_{OH}$ , NM e  $P_{D,st}$ . A *netlist* para o HSPICE é o Programa 11.

A carga que uma porta lógica representa depende das dimensões de seus transistores. A capacitância de entrada de um inversor é proporcional à área  $W_E \cdot L_E$  de seu EFET; quanto menor  $W_E \cdot L_E$ , menor o FO a que a porta equivale, e maior o  $V_{OH}$  do sinal de entrada. Por isso, a área  $W_E \cdot L_E$  do primeiro inversor, ou da primeira porta, de um circuito, deve ser pequena. Essa constatação é relevante para o projeto do FF JK DCFL, que será feito no Capítulo 4.

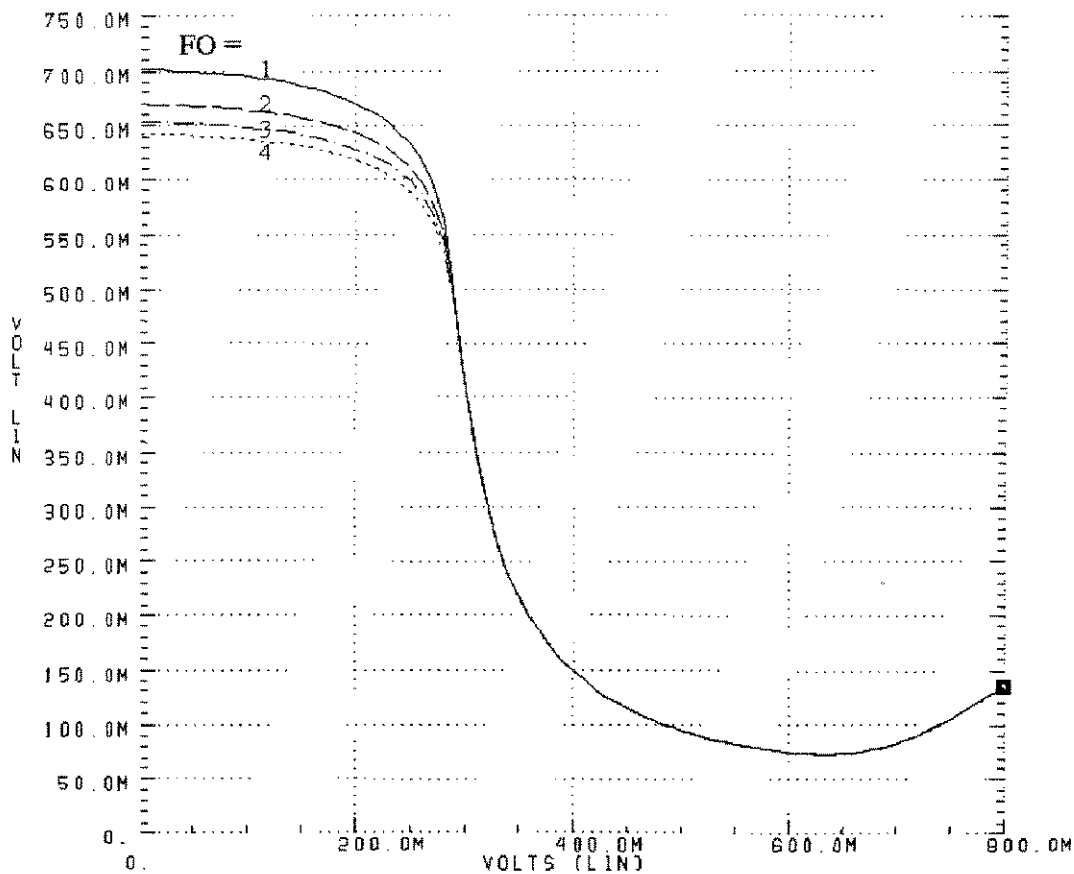


Figura 3.42 – Variação da curva de transferência de um inversor com o FO.

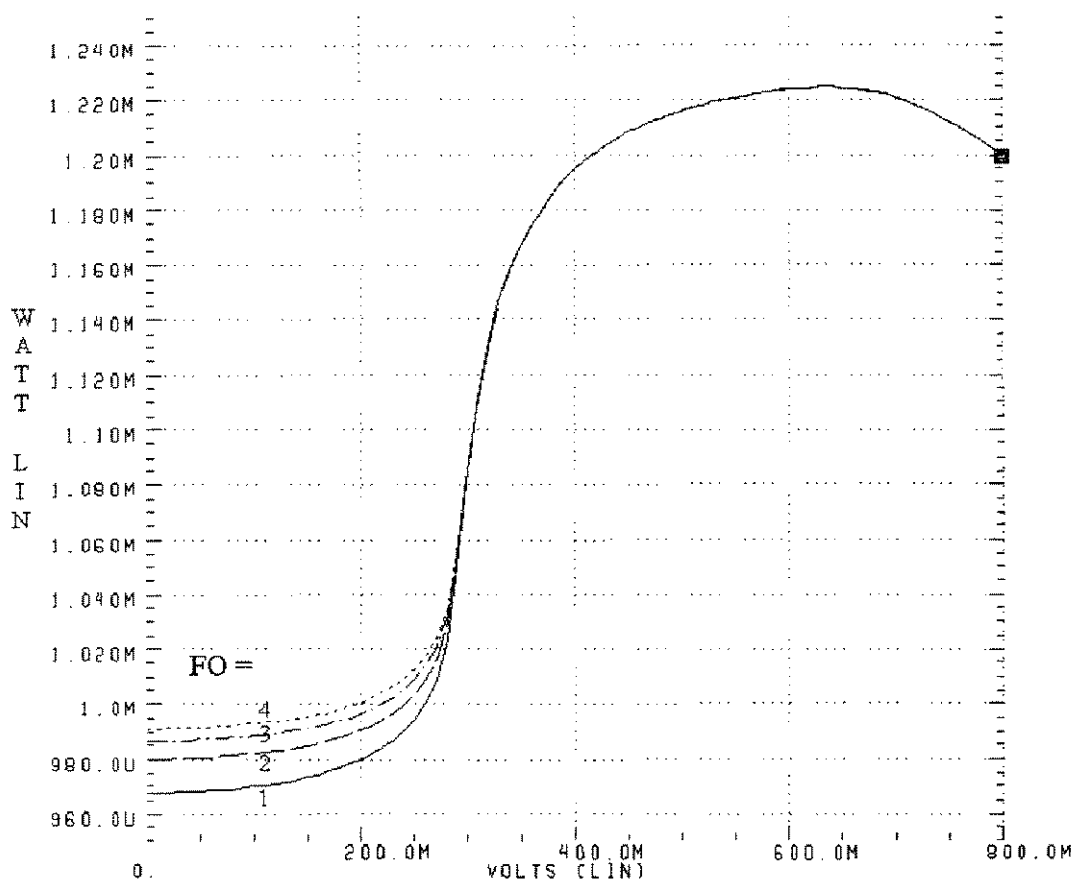


Figura 3.43 – Aumento da potência no estado lógico alto da saída de uma porta DCFL.

A influência do FO sobre o desempenho transiente do inversor pode ser avaliada pelas formas de onda de saída da figura 3.44. O atraso de propagação  $t_{PD}$  aumenta linearmente com o *fan-out* [9]. O inversor, com  $W_D = 8 \mu m$  e  $\beta = 10$ , foi simulado em uma frequência de entrada de 1 GHz, para FO variando na faixa de 1 a 4, no canto *slow* do processo, a  $0^\circ C$  e com  $V_{DD} = 1,8 V$ . Essas condições foram usadas para se obter sinais de saída do inversor e atrasos de propagação de pior caso. O Programa 12 é a *netlist* de entrada para essa simulação.

Percebe-se que o inversor simulado não consegue acionar *fan-outs* maiores do que 2 na frequência de 1 GHz. Para tanto, seria preciso aumentar as dimensões  $W_D$  e  $W_E$  desse inversor, sem que houvesse um aumento correspondente na carga. Mas em frequências mais baixas, o mesmo inversor certamente conseguiria acionar FO's maiores.

Para o acionamento de *fan-outs* de uma porta DCFL quando  $FO > 4$ , ou para o acionamento de cargas  $C_L$  acima de aproximadamente 200 fF, devem ser acoplados *buffers* de saída nessas portas. Portas lógicas bufferizadas (seção 3.4) mantém um alto  $V_{OH}$ , pequenos tempos de subida e descida do sinal de saída, e realizam um  $V_{OL}$  na faixa de 20 mV, enquanto que o de DCFL situa-se na faixa de 70 a 80 mV.

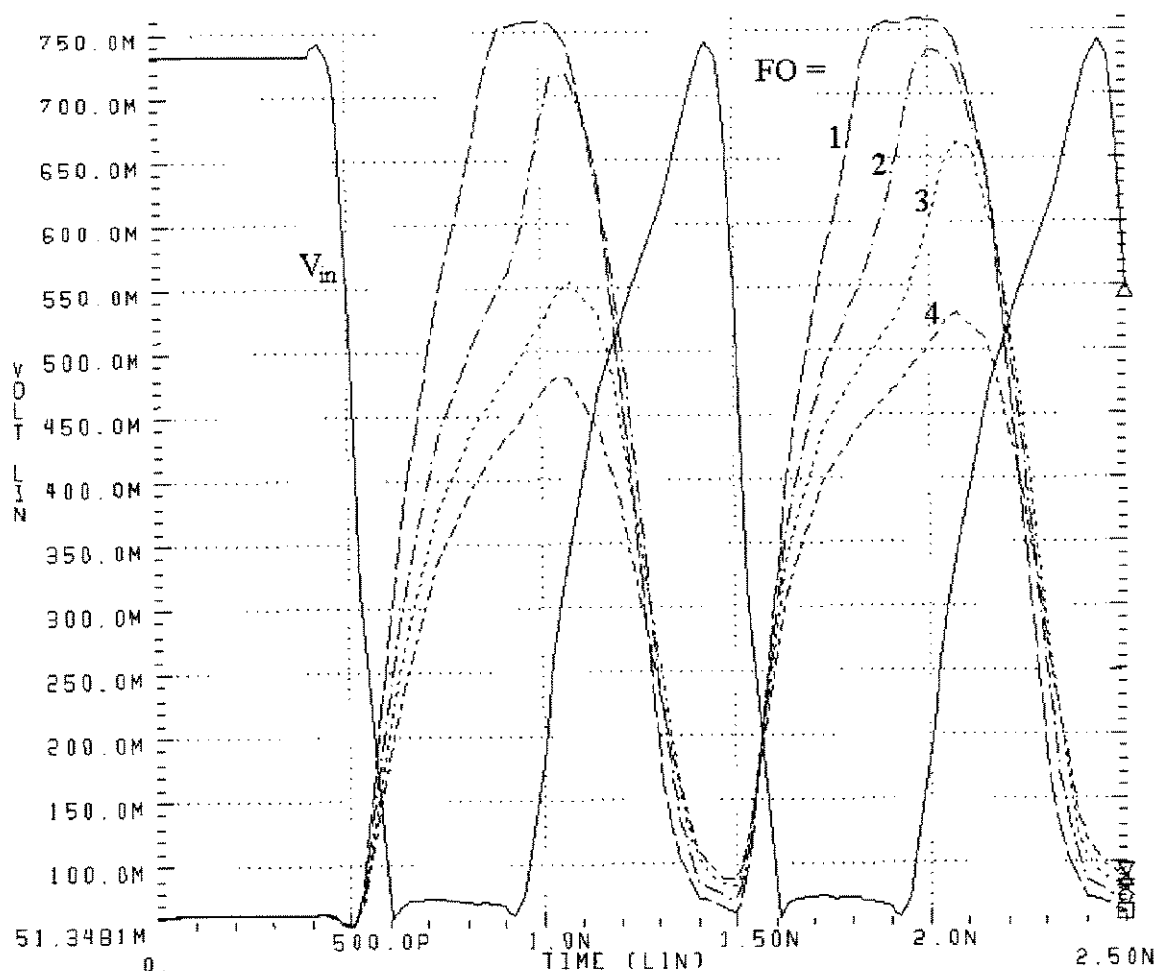


Figura 3.44 – Simulação transiente do inversor para FO variável de 1 a 4.

### 3.3. Projeto de Portas NOR DCFL.

O número de entradas de uma porta NOR é o seu *fan-in* (FI); o inversor pode ser visto como uma porta NOR de FI = 1. Portanto, as conclusões obtidas para inversores também são válidas para portas NOR, que podem ser tratadas simplesmente como inversores de  $\beta$  variável [78]. Na porta NOR de FI = 3 da figura 3.45, os EFETs em paralelo têm que ser todos iguais por simetria; logo, a expressão de  $\beta$  para o dimensionamento dos EFETs é a mesma do inversor, equação (3.6). Porém, o  $\beta$  efetivo da porta NOR é proporcional ao número de entradas que chaveiam simultaneamente,  $n$ :

$$\beta_{\text{NOR}} = n \cdot \beta_{\text{inv}} \quad (3.8)$$

onde  $\beta_{\text{inv}}$  é o  $\beta$  de um inversor, conforme a equação (3.6).  $\beta_{\text{NOR}}$  é, no máximo, igual a FI. $\beta_{\text{inv}}$ . A variação de  $\beta_{\text{NOR}}$  com  $n$  é ilustrada na figura 3.46, que compara as curvas de transferência possíveis para portas NOR de FI  $\leq 4$  e  $\beta = 10$ . Percebe-se que, se o *fan-in* de uma NOR for alto demais (FI > 4), isto poderá reduzir muito o limiar lógico da porta, comprometendo a sua NM alta.

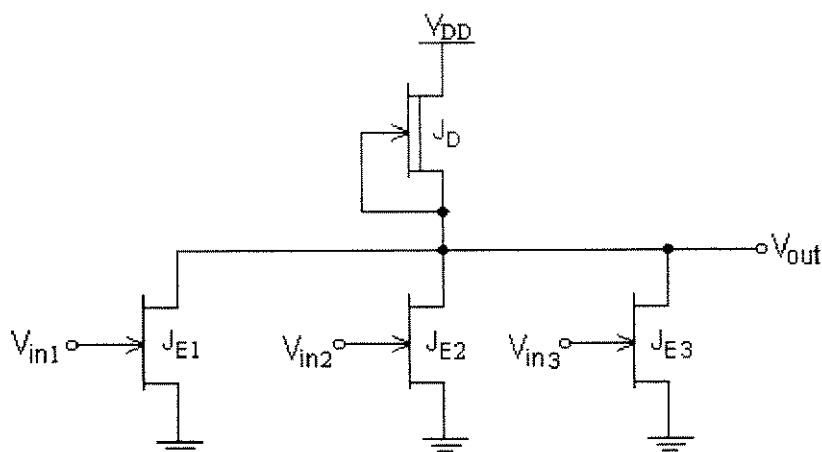


Figura 3.45 - Uma porta NOR de FI = 3 na família DCFL.

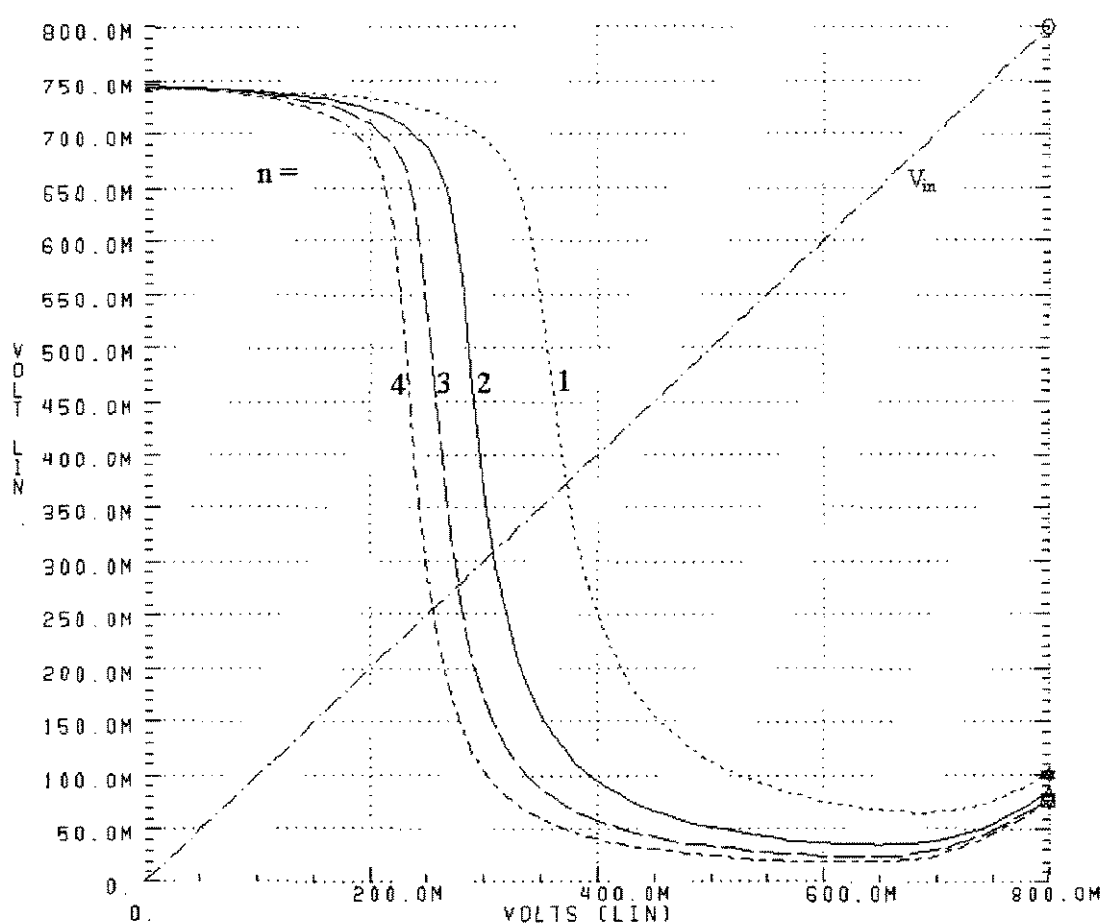


Figura 3.46 – Curvas de transferência de uma porta NOR de  $FI \leq 4$ , parametrizadas em  $n$ .

Além da possibilidade de que dois ou mais sinais de entrada de uma NOR chaveiem simultaneamente, tanto para nível '1' quanto para nível '0', o que acontece geralmente é que, quando uma das entradas já estiver em '1', outra vai para '1'; o efeito de se aumentar o número de EFETs ligados em uma NOR é apenas uma redução do valor de  $V_{OL}$  na saída da porta, mas não ocorre transição na saída. Isto é mostrado nas curvas da figura 3.47, onde o eixo vertical corresponde ao sinal D1 e o eixo horizontal, ao sinal A1 do circuito da figura 3.48. A curva de número 1 corresponde ao caso em que há uma transição lógica em A1, e as demais entradas das duas portas mantém-se em  $V_{OL}$ . Na curva 2,  $B1 = V_{OH}$  durante a transição em A1 e, na curva 3,  $B1 = C1 = V_{OH}$  durante a transição.

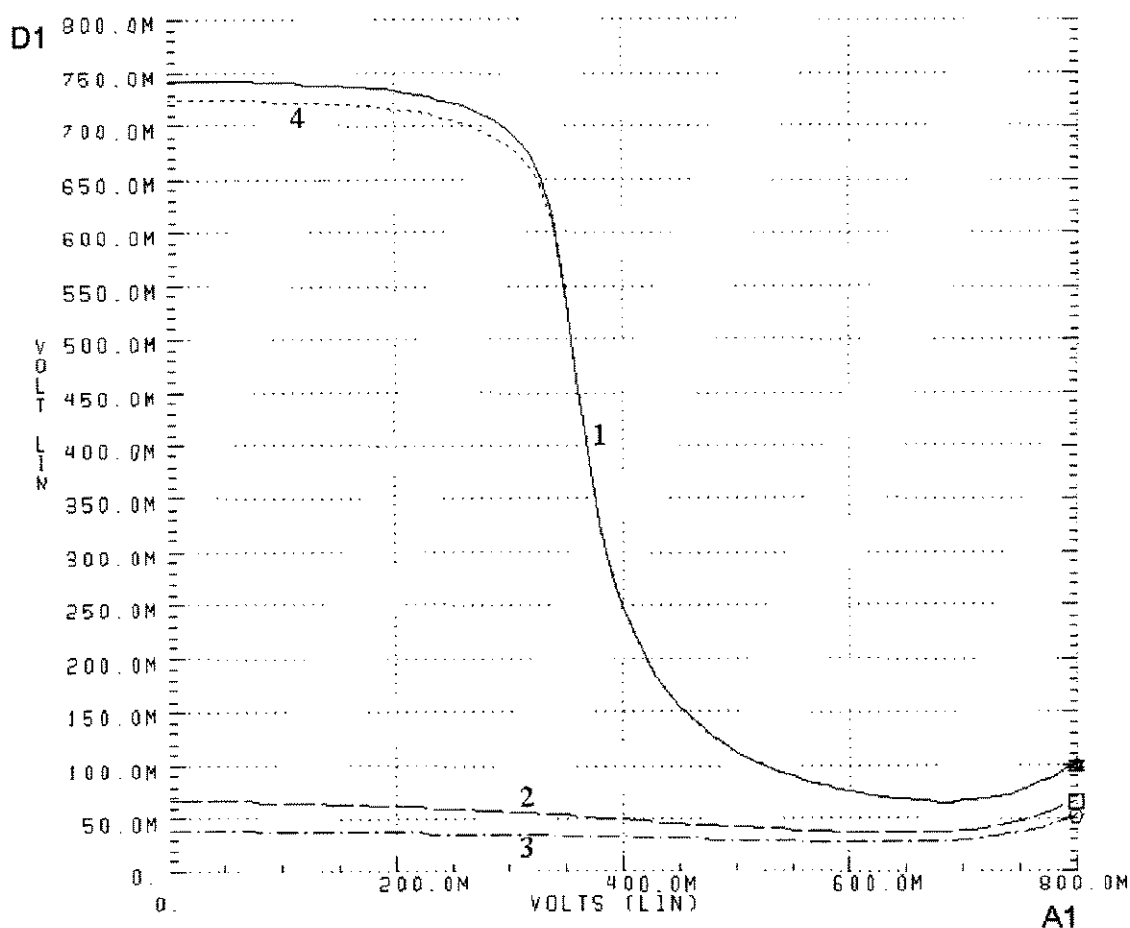


Figura 3.47 – Quatro curvas de transferência D1 x A1 possíveis para o circuito da figura 3.48.

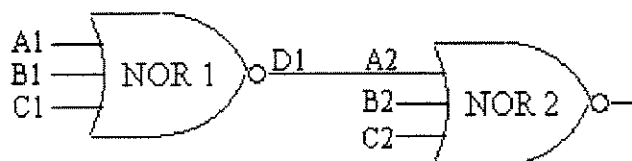


Figura 3.48 – Uma porta NOR de FI = 3 aciona outra NOR idêntica.

A curva 4 da figura 3.47 é uma transição de D1 quase idêntica à curva 1, com  $B1 = C1 = V_{OL}$ . A diferença entre as curvas 1 e 4 é que, nesta,  $B2 = C2 = V_{OH}$ , e isto aumenta a  $C_{in}$  da porta NOR 2, pois mais capacitâncias tornam-se parte do seu circuito equivalente. Por isso, NOR 2 aparece como uma  $C_L$  maior para NOR 1.

Analisando a figura 3.47, conclui-se que as portas NOR têm que ser projetadas de modo a apresentarem um  $V_{OL}$  aceitável mesmo no pior caso de operação, que é a situação de apenas uma entrada estar em nível alto (somente um EFET ligado). Neste caso, o  $V_{OL}$  já precisa ser suficientemente baixo para garantir boas NM's. Em função do exposto acima, as portas NOR deste trabalho foram todas projetadas com  $\beta = 10$ , tal como os inversores.

Na porta NOR da figura 3.45 foram ligados EFETs idênticos em paralelo, portanto a corrente  $I_{D,E}$  em cada um que esteja com a sua entrada em  $V_{OH}$  ( $\cong 0,7$  V), será aproximadamente igual à corrente do DFET da NOR dividida pelo número de entradas simultaneamente altas. Se a porta NOR da figura 3.45 for comparada com um inversor que



tenha os mesmos valores de  $W_D$  e  $W_E$ , observa-se que a corrente no DFET da NOR tem o mesmo valor da corrente no DFET do inversor. Se a largura do DFET da porta NOR for aumentada, o  $t_{PD}$  da porta será reduzido da mesma forma que acontece para o inversor.

As figuras 3.49 (a), (b) e (c) mostram as correntes nos FETs da porta NOR da figura 3.45, em função da tensão de entrada A1. Na figura 3.49(a), durante a transição de A1, as tensões em B1 e C1 mantêm-se em  $V_{OL}$ ; em 3.49(b),  $V_{B1} = V_{OH}$  e  $V_{C1} = V_{OL}$ ; e em 3.49(c),  $V_{B1} = V_{C1} = V_{OH}$ . Em 3.49(b) e (c), a soma das correntes instantâneas em cada EFET é aproximadamente igual à corrente do DFET, que pouco varia em função do número de EFETs ativos. Então não há um aumento real da capacidade de corrente de descarga com o aumento do FI da porta NOR. Outra conclusão é que a potência consumida quase não varia com o FI.

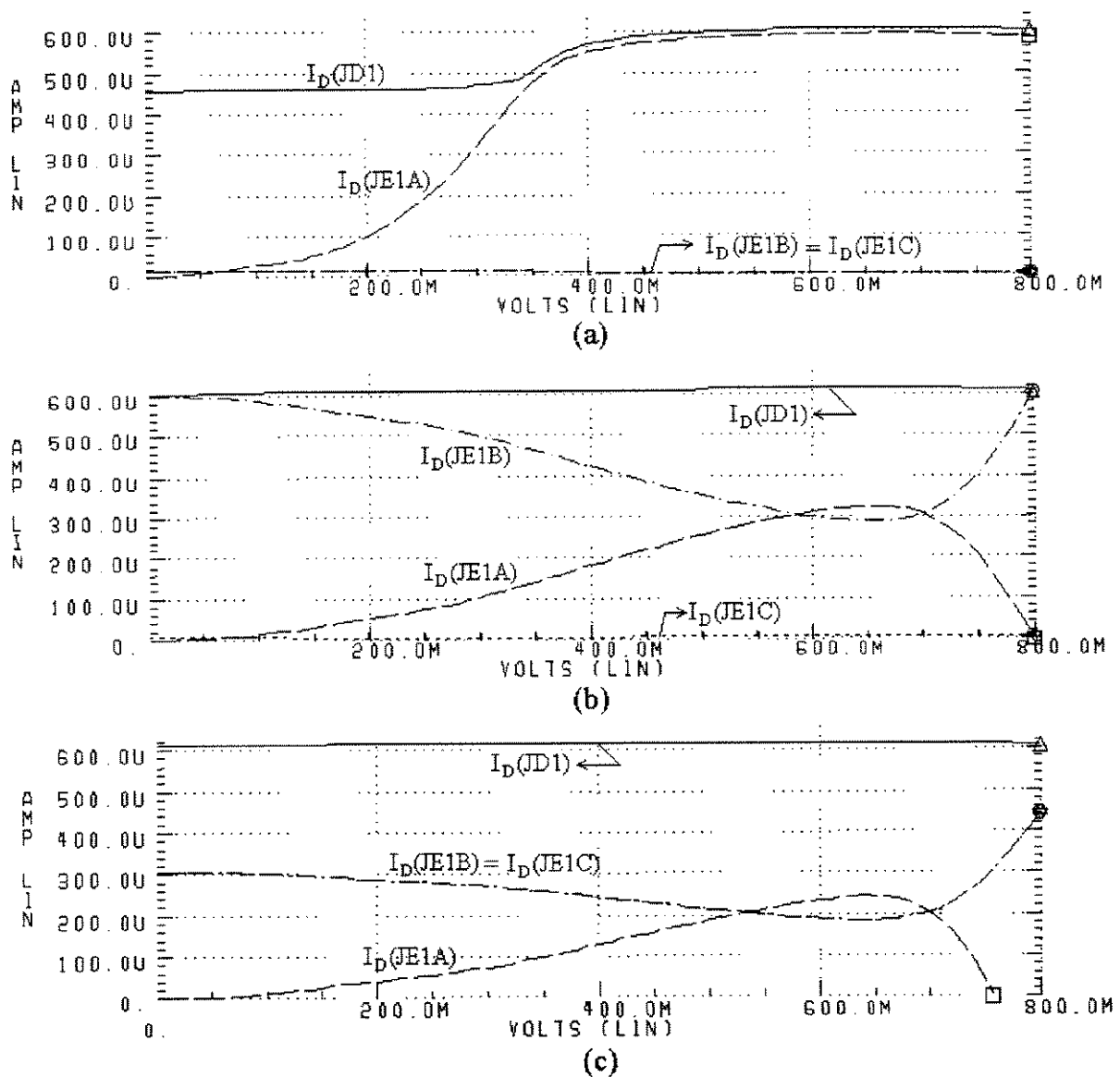


Figura 3.49 – Correntes nos FETs da porta NOR da figura 3.45, em função da tensão na entrada A1. Em (a),  $B1 = C1 = V_{OL}$ ; em (b),  $B1 = V_{OH}$ ,  $C1 = V_{OL}$ ; em (c),  $B1 = C1 = V_{OH}$ .

As curvas das figuras 3.46, 3.47 e 3.49 foram geradas com os Programas 13 a 17 do Apêndice 1. Os Programas de 13 a 16 simulam o chaveamento **simultâneo** de 1 a 4 entradas de uma porta NOR, e o Programa 17 simula o chaveamento de uma entrada, mas para diferentes combinações de estados fixos nas outras entradas.

Para a análise transiente, foi simulado um circuito semelhante ao da figura 3.22, com os inversores substituídos por portas NOR. Variou-se o FI da porta NOR em análise (X2) de 1 até 5 (FI = 1 é um inversor para comparação), enquanto as demais portas foram mantidas sempre com  $W_D = 8 \mu\text{m}$ ,  $\beta = 10$  e FI = 3; assim, a  $C_L$  total para a porta X2 foi mantida fixa. O  $W_D$  de X2 também foi mantido em  $8 \mu\text{m}$ . A variação da resposta transiente de X2 com o FI, na frequência de 1 GHz, é mostrada na figura 3.50. O sinal de excitação foi aplicado sempre na mesma entrada da porta X2, com as novas entradas sendo acrescentadas à porta e excitadas com  $V_{OL}$  (70 mV). A *netlist* do circuito para a simulação é o Programa 18.

Como os EFETs da NOR não ficam completamente cortados quando suas entradas estão em  $V_{OL}$ , ocorre a soma das suas correntes  $I_{SUB}$ , e eles desviam uma fração FI vezes maior da corrente de carga fornecida pelo DFET ( $FI \cdot I_{SUB}$ ), do que no caso de um inversor. Por causa disso, na resposta transiente da figura 3.50, o  $t_{LH}$  e o  $t_{PLH}$  são maiores para uma porta NOR do que para um inversor com DFET de mesma largura, e são proporcionais ao FI. Além disso, aumentar o FI causa o aumento das capacitâncias internas e de entrada da porta NOR, o que aumenta o tempo de atraso de um sinal que tenha que passar por essa porta.

O FI degrada principalmente o tempo  $t_{LH}$  e o atraso  $t_{PLH}$ , por isso não devem ser feitas portas NOR com FI > 3 [66]. Além disso, para uma porta NOR de FI muito alto, as NMs **nas suas entradas** são reduzidas. Se for realmente necessária uma porta NOR de quatro ou mais entradas, cada uma das entradas deverá ser acionada por um pequeno buffer SDCFL (discutido na próxima seção), para que o  $V_{OL}$  nessas entradas seja próximo de 0 V [77]. Portas NOR de entradas muito largas (alto  $W_E$ ) também são mais lentas.

A figura 3.51 mostra  $t_{PLH}$ ,  $t_{PHL}$  e  $t_{PD}$  em função do FI, confirmando o que se observa nas formas de onda da figura 3.50: o  $t_{PLH}$  é bem maior do que o  $t_{PHL}$  e também é mais sensível ao aumento do FI (aumenta mais depressa com FI). Para o FI sendo incrementado de 1 a 5, o aumento correspondente de  $t_{PLH}$  foi de 56%, enquanto o aumento correspondente de  $t_{PHL}$  foi de 28%, de seus valores em FI = 1.

Para um circuito DCFL real não apresentar “gargalos” limitando sua taxa de clock, suas portas NOR devem ser balanceadas de acordo com suas cargas e *fan-out*. Por exemplo, se a porta NOR 1 tem FO = 1 enquanto a porta NOR 2 tem FO = 3, esta porta deve ter FETs mais largos para compensar pela sua maior carga.

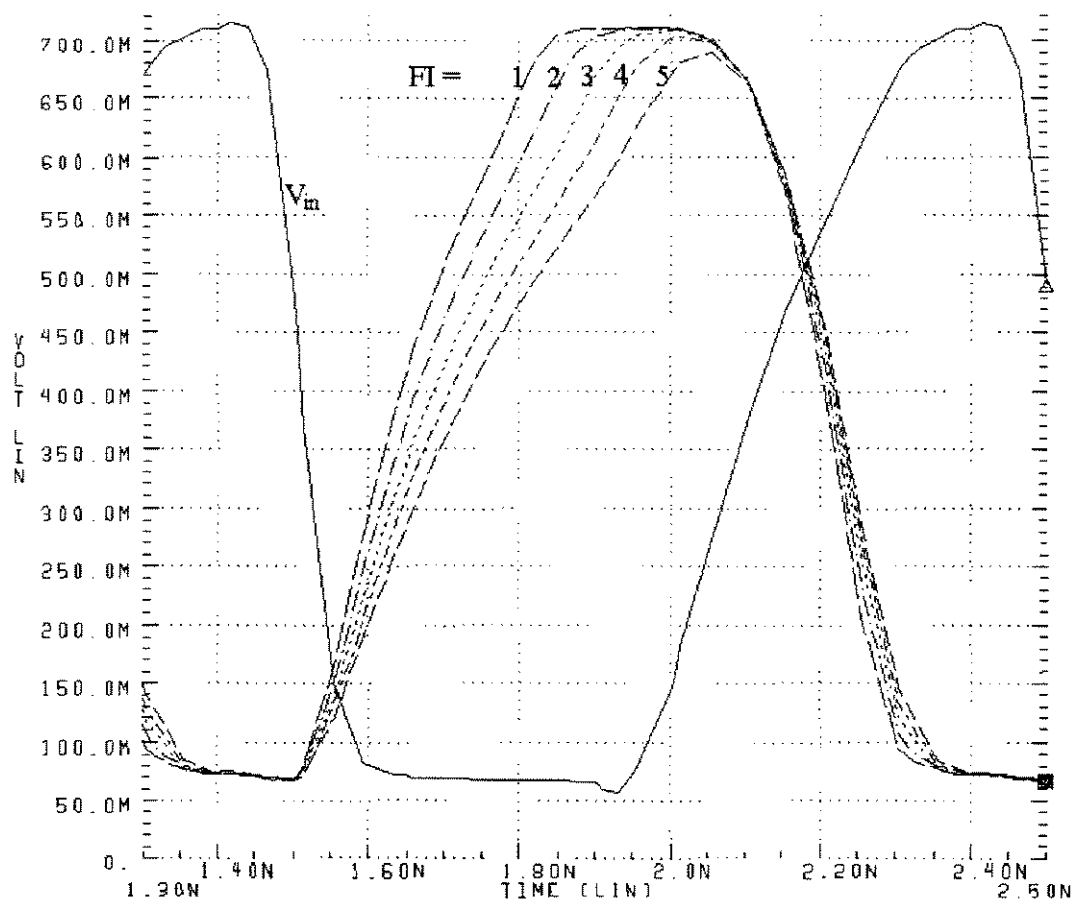


Figura 3.50 – Resposta transiente de uma porta NOR em função do seu FI.

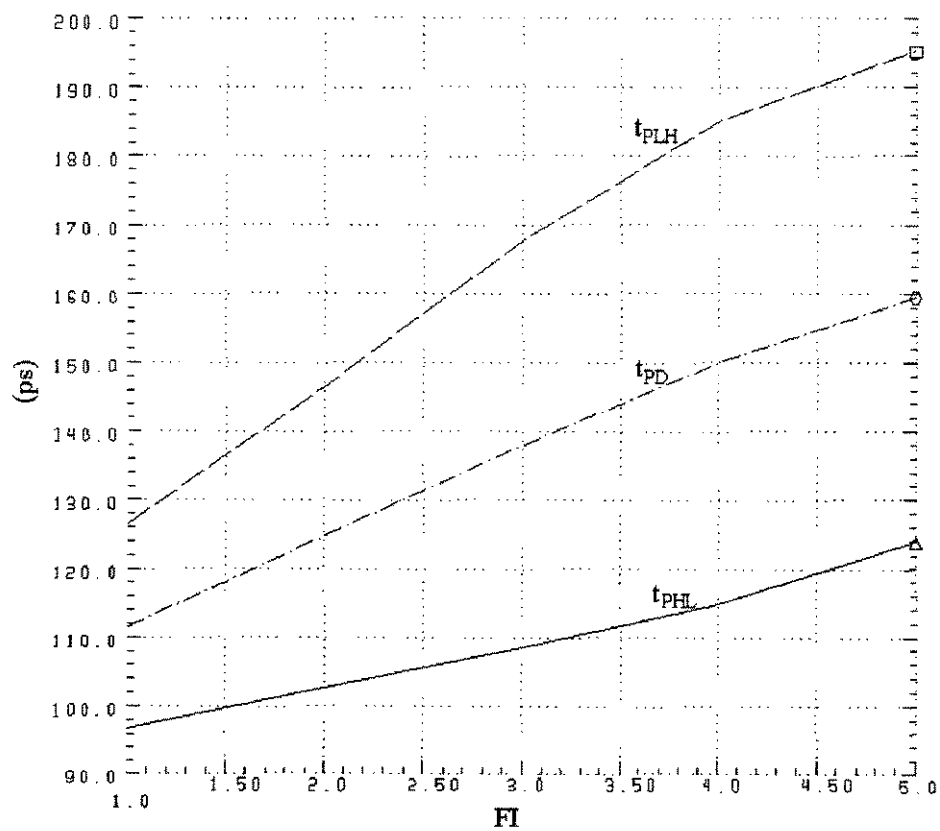


Figura 3.51 – Os atrasos de propagação de uma porta NOR em função do seu FI.

### 3.4. Projeto de *Drivers* SDCFL e *Buffers* de Entrada.

#### 3.4.1. Projeto de Portas Lógicas SDCFL.

*Buffers* são blocos de circuito que têm a função de isolar dois outros circuitos, ou um circuito e a sua carga; eles são usados na entrada e na saída de *chips*. Os *buffers* de entrada, discutidos na seção 3.4.2, devem fazer o casamento de impedância com a linha que aciona o *chip* e proteger seus circuitos internos contra tensões excessivas nos pinos de entrada. Os *buffers* de saída (ou *drivers*) isolam a parte lógica do *chip* da sua carga, e são necessários para o acionamento eficiente de grandes capacitâncias de carga, como por exemplo as interconexões internas mais longas de um *chip*, ou um alto número de portas lógicas de fan-out, ou os *pads* do *chip*.

Os *drivers* não têm função do ponto de vista lógico (não alteram o valor lógico de um sinal), eles só visam garantir baixos tempos  $t_{HL}$  e  $t_{LH}$  para um sinal. Para carregar e descarregar capacitâncias elevadas rapidamente, estes *buffers* precisam fornecer altas correntes de saída (ver equação (3.5)). Simplesmente aumentar as larguras dos FETs de portas DCFL não traria o mesmo nível de desempenho, com a mesma ocupação de área de *chip*. Neste projeto, serão utilizados os *drivers* SDCFL (*Source-follower DCFL*), também chamados de BDCFL (*Buffered DCFL*) por alguns autores.

SDCFL é uma família de portas bufferizadas totalmente compatível com DCFL, cuja estrutura consiste de uma porta DCFL (inversor ou NOR) acionando um seguidor de fonte, e que está mostrada na figura 3.52(a). O emprego de SDCFL para  $C_L$ 's altas propicia: maior  $V_{OH}$  e menor  $V_{OL}$  na saída de blocos lógicos, e portanto NMs muito maiores em comparação com as de portas DCFL de mesma área, acionando uma mesma carga  $C_L$ ; e/ou aumento da capacidade de acionamento de fan-out de um circuito DCFL, pois SDCFL tem um alto ganho de corrente. As capacitâncias parasitas do seguidor de fonte que forma a família SDCFL são mostradas na figura 3.52 (b).

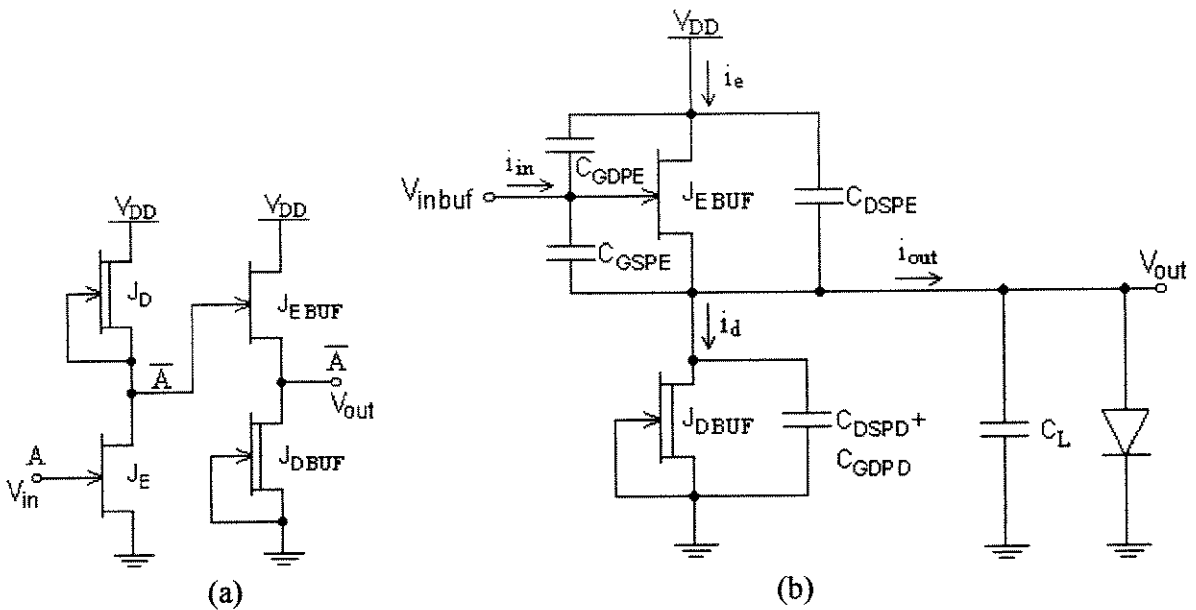


Figura 3.52 – a) Inversor SDCFL; b) Seguidor de fonte com capacitâncias parasitas e uma  $C_L$ .

O seguidor de fonte é um circuito em que a saída segue a entrada com a diferença de uma queda de tensão. Na figura 3.52(b), se  $V_{inbuf}$  aumenta,  $i_e$  e  $i_d$  também aumentam, forçando  $V_{out}$  a aumentar porque  $i_e > i_d$ . A excursão de  $V_{inbuf}$  pode ser muito maior do que em um inversor DCFL, chegando tipicamente a 1,5 V (para  $V_{DD} = 2$  V). Já quando  $V_{inbuf}$  é baixa, a saída cai a um valor próximo de  $V_{inbuf} - V_t$ , portanto  $V_{OL}$  alcança um valor muito baixo. A curva de transferência do seguidor de fonte, na figura 3.53, mostra que, em DC,  $V_{out}$  segue  $V_{inbuf}$  até um alto valor desta e, para uma  $V_{inbuf}$  baixa,  $V_{out}$  é próxima de 0 V. Essa curva de transferência pode ser plotada por um método gráfico [66] similar ao da obtenção da curva de transferência do inversor DCFL (figuras 3.9 a 3.12).

$J_{E_{BUF}}$  solicita uma corrente de entrada  $i_{in}$  muito pequena, mesmo quando  $V_{inbuf}$  está no valor máximo de 1,5 V. A  $C_{GS}$  do EFET auxilia a passagem da corrente de carga vinda da porta que aciona o seguidor de fonte. A figura 3.54 mostra a curva de  $i_{in} \times V_{inbuf}$ , parametrizada em função do FO do seguidor de fonte. Como consequência, a parte lógica da porta SDCFL deve ser menor (larguras menores).

As NMs de circuitos DCFL podem ser melhoradas mantendo-se  $V_{OH}$  próximo dos 700 mV e reduzindo-se  $V_{OL}$ . Pois o  $V_{OL}$  de uma porta SDCFL é de aproximadamente 20 mV (contra 70 mV de DCFL), que é um valor excelente para cortar quase que completamente o EFET de entrada da porta lógica que vier a seguir ([77], [78]). O *buffer* SDCFL também pode constituir um meio **eficiente** de se acelerar os elementos lógicos porque, quando a  $C_L$  ou o FO são elevados, o seguidor de fonte os aciona mais rapidamente do que um inversor DCFL de mesma área. Finalmente, o emprego de uma porta SDCFL como driver de saída de um *chip* acarreta uma ótima excursão do sinal de saída do CI, da ordem de 1,2 V para  $V_{DD} = 2$  V.

O compromisso pelo uso de SDCFL vem na forma de um consumo de potência bem maior e um aumento de complexidade da porta bufferizada, que ocupa maior área pois tem dois FETs a mais. **Não é recomendável** empregar-se portas SDCFL para compor a parte lógica de um circuito, em substituição às portas DCFL, pois a área do circuito (o seu custo de fabricação) e as correntes de todo o CI aumentam drasticamente, e esses efeitos não serão compensados pelo benefício das maiores NMs nos nós internos do CI.

O dimensionamento de um *buffer* SDCFL é feito para o acionamento de uma  $C_L$  especificada. Quando o dimensionamento do seguidor de fonte é muito fraco para a  $C_L$  que ele tem que acionar, a onda que resulta na saída é uma dente-de-serra, evidenciando a carga e descarga do capacitor, mas nunca atingindo algum dos níveis lógicos (alto ou baixo), porque não há corrente suficiente. À medida em que o dimensionamento do *buffer* vai sendo aumentado, o sinal de saída vai adquirindo a forma quadrada, com rampas de subida e descida cada vez mais verticais, e a duração dos níveis lógicos (ciclo de trabalho) vai aumentando.

Os *buffers* SDCFL somente são eficientes para o acionamento de  $C_L$ 's grandes, acima de 100 fF [78]. O projetista precisa ter cuidado para não sobredimensionar o seguidor de fonte, porque usá-lo para acionar uma  $C_L$  pequena ( $C_L < 100$  fF), ou usar um *buffer* forte demais para uma dada  $C_L$ , apenas causará um  $t_{PD}$  ainda maior para o sinal, ao invés de um ganho de velocidade. Portanto, o projetista precisa estar ciente de que o uso da lógica SDCFL não é necessariamente benéfico, pois pode tornar a resposta do circuito mais lenta do que antes. Este efeito nocivo será demonstrado adiante. As equações dos atrasos de propagação do seguidor de fonte são deduzidas, e o seu funcionamento é explicado com detalhes, em [66].

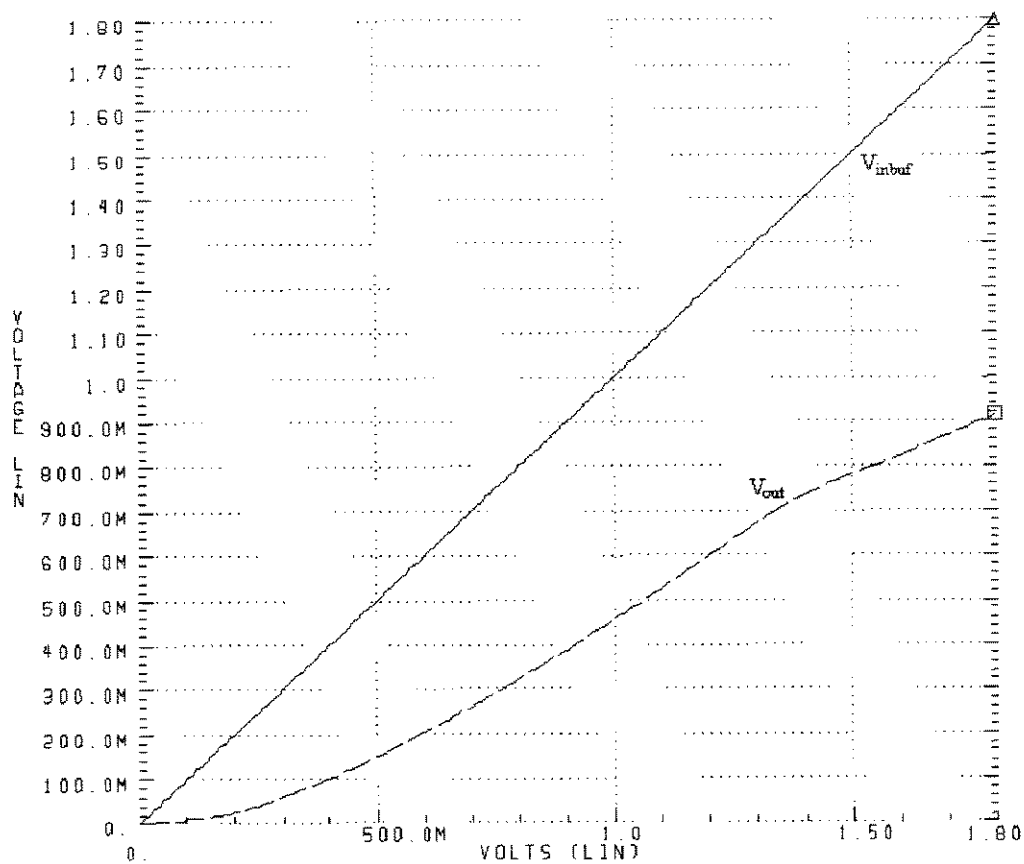


Figura 3.53 – Curva de transferência do seguidor de fonte.

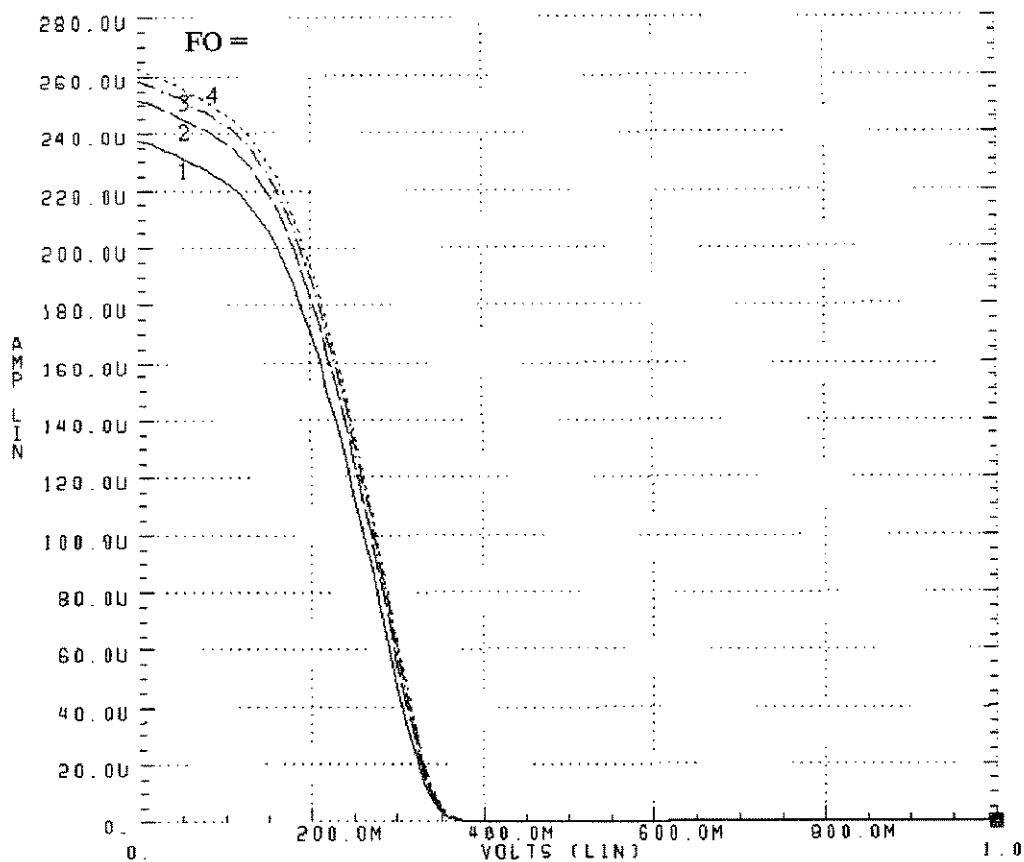


Figura 3.54 – Corrente de entrada do seguidor de fonte, parametrizada em FO.

O *buffer* SDCFL deve ser sempre acionado por uma porta DCFL, como na figura 3.52(a), e o  $V_{OH}$  do *buffer* será compatível com os níveis de entrada DCFL, perfazendo o inversor SDCFL. A figura 3.55 mostra a curva característica do inversor SDCFL, parametrizada para a quantidade de fan-out na sua saída. A comparação destas curvas com as da figura 3.42 revela as vantagens do inversor SDCFL sobre o DCFL:  $V_{OL}$  é menor por 50 mV e, apesar da redução do nível  $V_{OH}$  com o aumento do FO ser igual àquela do inversor DCFL, o valor de  $V_{OH}$  permanece acima de 700 mV até um  $FO \cong 7$ .

A figura 3.55 esclarece o significado da maior capacidade de acionamento de FO que o uso do *buffer* SDCFL proporciona, e também mostra porque SDCFL somente deve ser usado para acionar FO's e/ou  $C_L$ 's elevados: o valor de  $V_{OH}$  é alto demais para cargas leves ( $FO < 5$ ), e isso prejudicaria as portas lógicas DCFL que fossem acionadas pela porta SDCFL. Daí a importância de saber quando usar SDCFL. As curvas das figuras 3.54 e 3.55 foram geradas pela simulação do Programa 20.

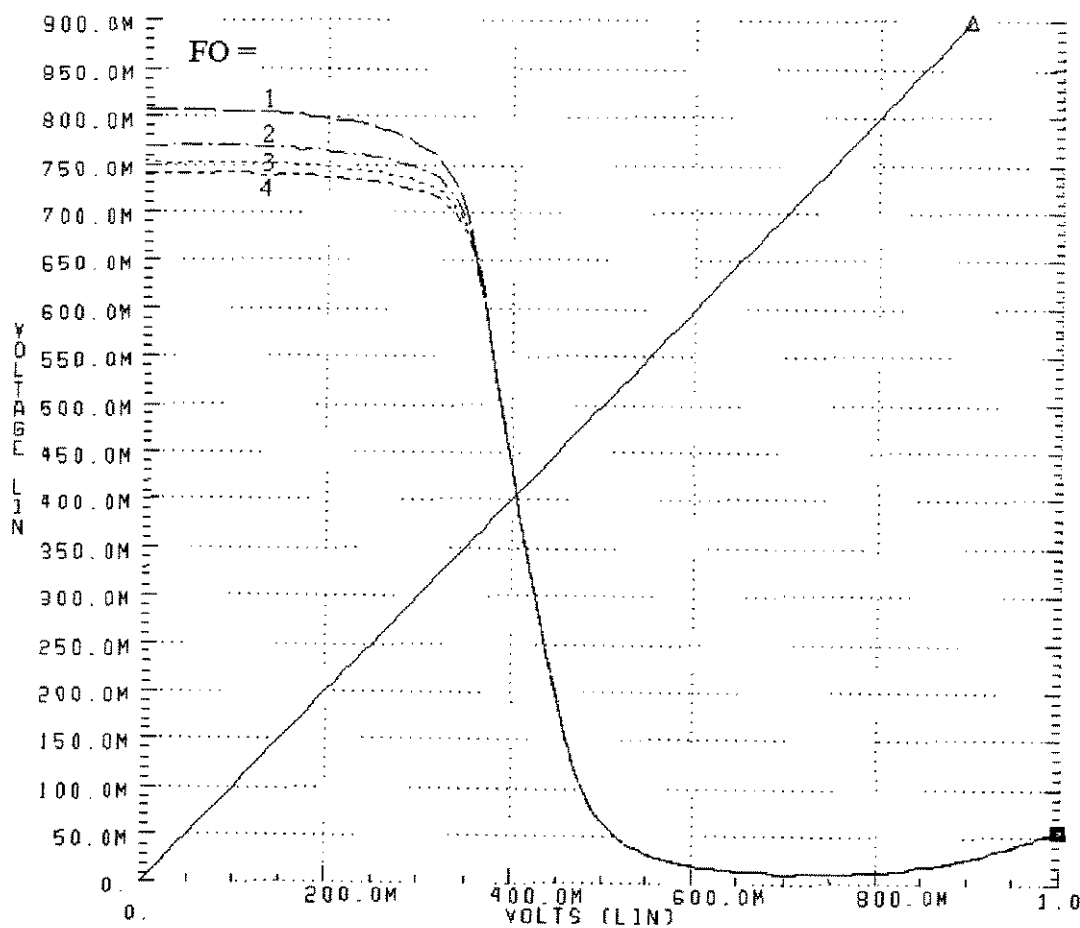


Figura 3.55 – Efeito do FO sobre a curva de transferência DC de um inversor SDCFL.

A parte lógica da porta SDCFL (inversor ou NOR) deve ter um valor de  $\beta$  em torno de 5, de modo a otimizar o desempenho da porta SDCFL como um todo; isto é ilustrado na figura 3.56. Uma porta com  $\beta_{inv} = 10$  resulta em maior atraso, devido à característica de alta impedância de entrada do seguidor de fonte. O Programa 19 compara curvas de transferência do inversor SDCFL, variando o  $\beta$  da parte DCFL na faixa de 3 até 12, e o resultado é mostrado na figura 3.57; para  $\beta_{inv} = 5$ , a curva do inversor SDCFL intersecciona a reta da tensão  $V_{in}$  bem perto de 400 mV, o que significa um limiar lógico da porta quase no meio da sua excursão total de 800 mV.

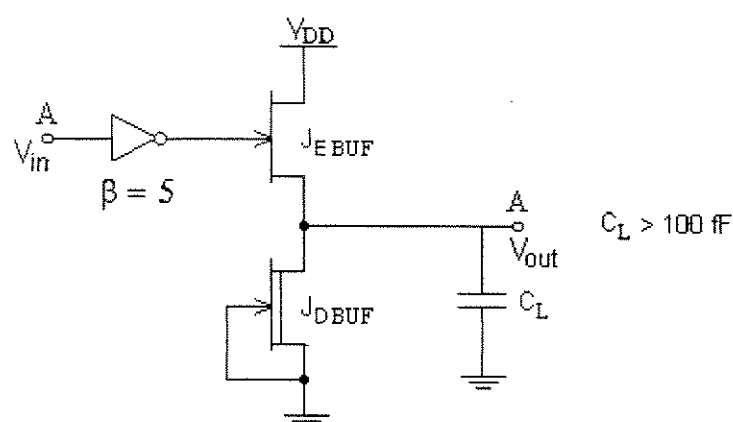


Figura 3.56 – O  $\beta$  da porta DCFL que aciona o seguidor de fonte deve ser igual a 5.

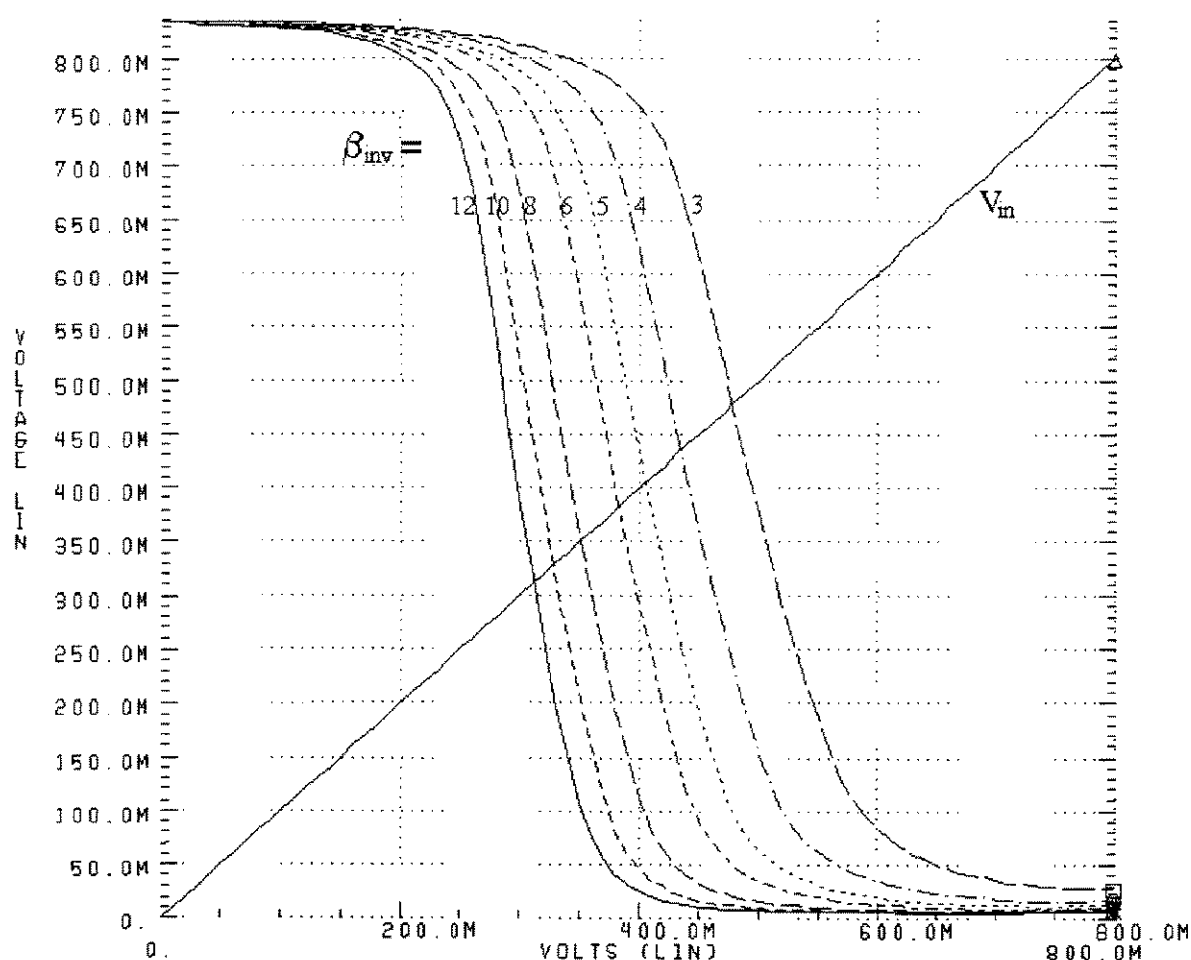


Figura 3.57 – Determinação de um  $\beta$  ótimo para a parte inversora da porta SDCFL.

Para o projeto do seguidor de fonte do inversor SDCFL, analisa-se inicialmente a influência da largura do DFET do *buffer*,  $W_D$ . Os compromissos envolvendo  $W_D$ , potência e velocidade para portas SDCFL são basicamente os mesmos que no caso das portas DCFL. A relação  $W_D \uparrow \Rightarrow P_{D,st} \uparrow$  continua linear, conforme ilustra a figura 3.58. A figura 3.59 mostra várias curvas da  $P_{D,st}$  do inversor SDCFL, parametrizadas em  $W_D$ . Com  $\beta_{buf}$  mantido constante,  $W_D \uparrow \Rightarrow V_{OH} \uparrow$ ,  $NM \uparrow$ , o que é comprovado pela figura 3.60. Porém, aumentar  $W_D$  continuamente não garante a redução contínua de  $t_{PD}$ : o valor ótimo de  $W_D$  depende do valor de  $C_L$ .



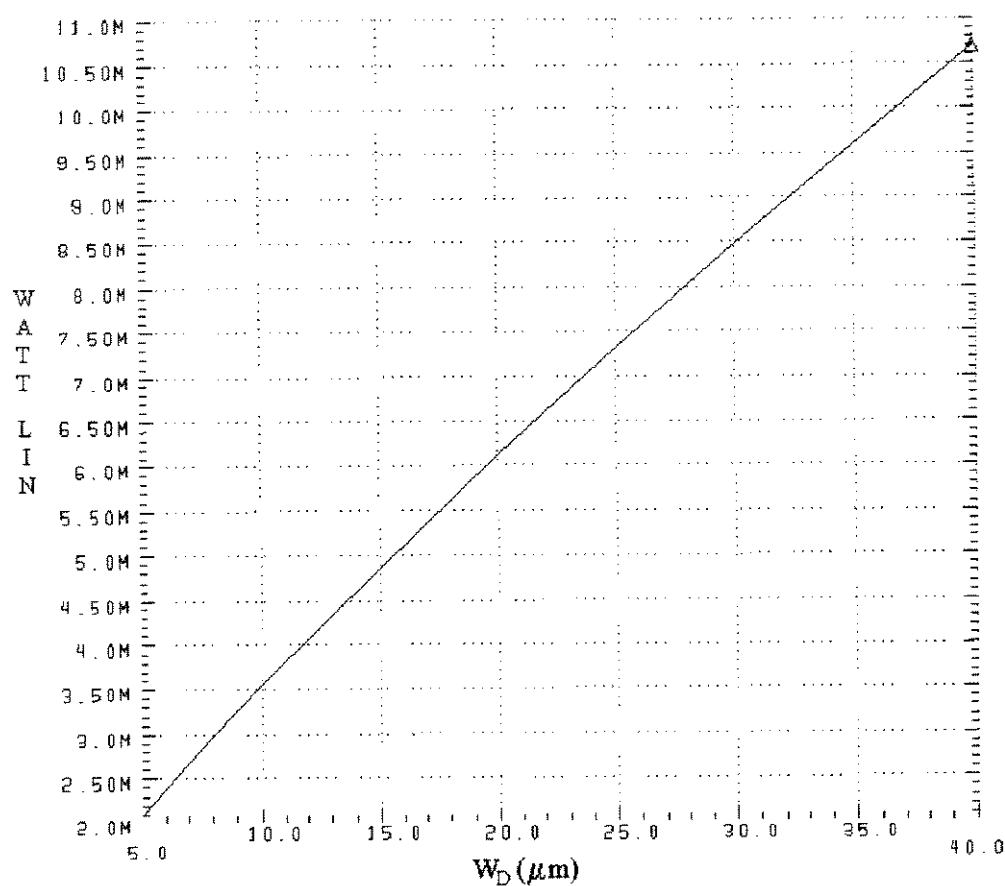


Figura 3.58 – Potência do inversor SDCFL *versus*  $W_D$  do seguidor de fonte.

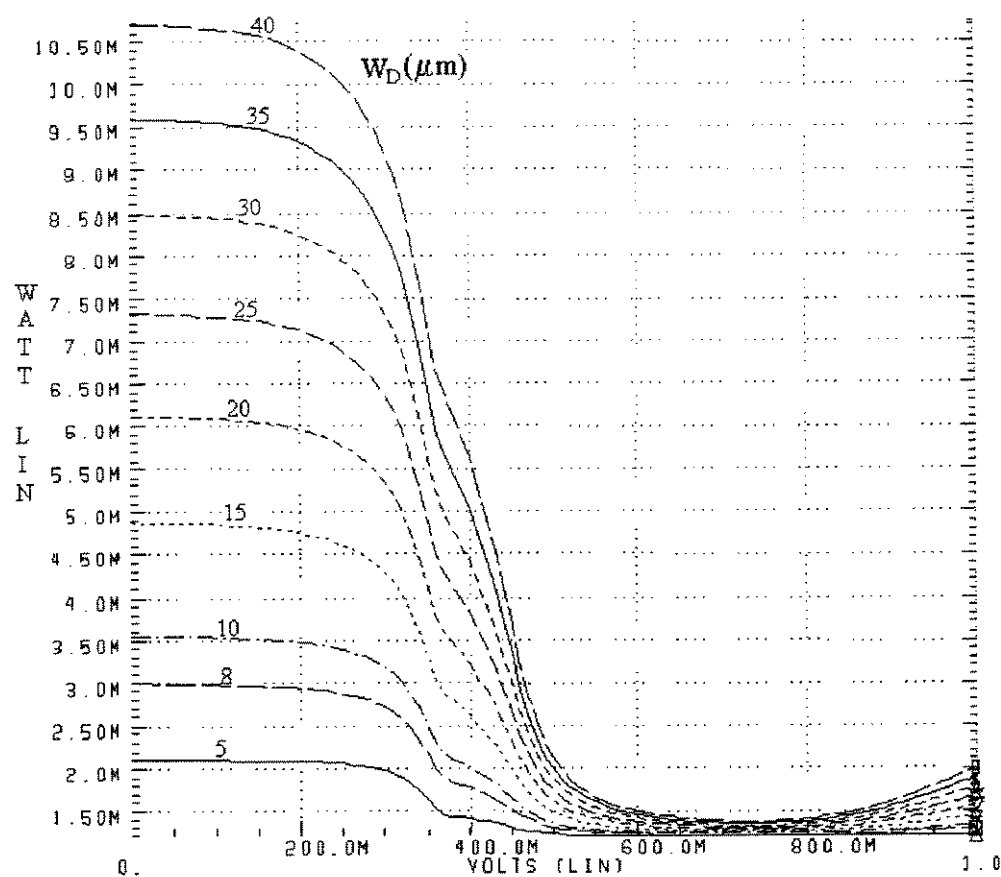


Figura 3.59 – Curvas de potência *versus*  $V_{in}$ , para vários valores de  $W_D$ .

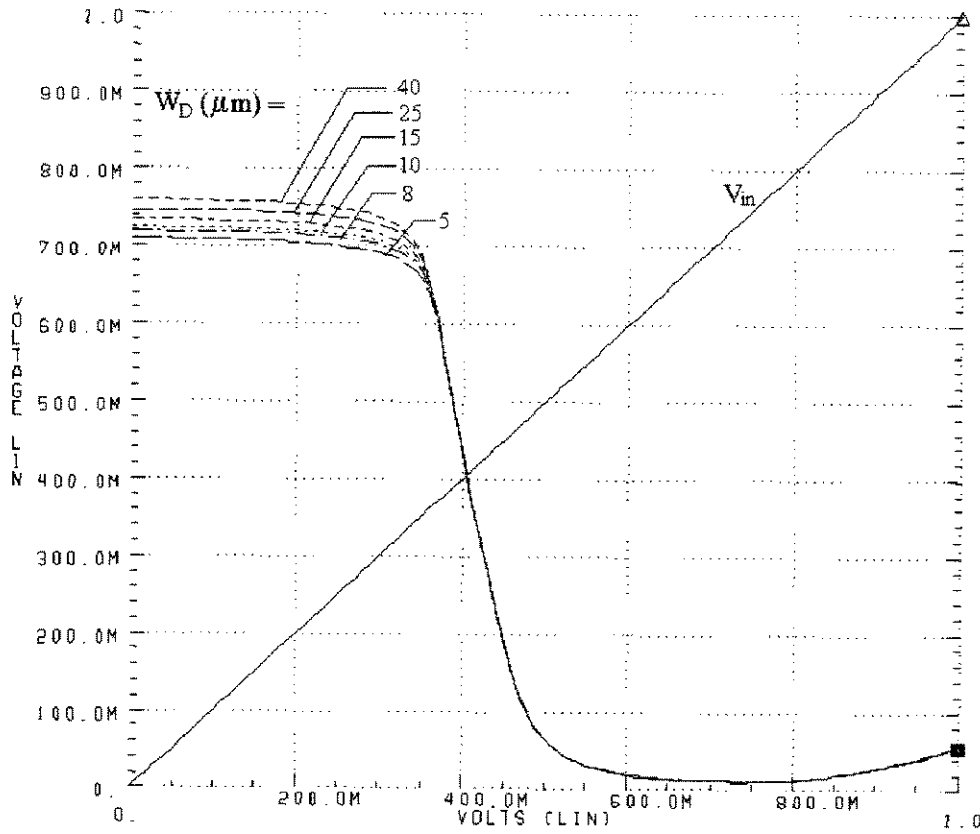


Figura 3.60 – Influência de  $W_D$  na curva de transferência do inversor SDCFL com  $\beta_{\text{buf}} = 3$ .

As figuras de 3.58 a 3.60 correspondem a um inversor SDCFL no qual o  $\beta$  do *buffer* é igual a 3 e a porta DCFL é mantida sempre fixa com  $W_D = 8 \mu\text{m}$  e  $\beta = 5$ . O FO para o *buffer* consiste de 4 inversores DCFL, com  $W_D = 8 \mu\text{m}$  e  $\beta = 10$ , e uma  $C_L$  de 100 fF. Este circuito é simulado por meio do Programa 21.

Para diminuir-se  $t_{\text{LH}}$  e  $t_{\text{PLH}}$ , a razão  $W_D/W_E$  deve ser feita pequena, pois  $C_L$  é carregado através do EFET, e qualquer corrente extra puxada pelo DFET reduz a corrente disponível para a carga de  $C_L$ . Contudo, para reduzir  $t_{\text{HL}}$  e  $t_{\text{PHL}}$ ,  $W_D$  deve ser aumentado, pois é o DFET que descarrega  $C_L$ . Mais uma vez, surgem requisitos conflitantes para um projeto ótimo, portanto deve-se dimensionar a razão de larguras  $W_E/W_D$  para que produza um sinal de saída do *buffer* SDCFL com  $t_{\text{HL}} \cong t_{\text{LH}}$ . Para tanto, a corrente de carga de  $C_L$  deve ser igual à corrente de descarga. Essa condição, aplicada à figura 3.52(b), implica em [66]:

$$i_e(V_{\text{GS}} = 0,7 \text{ V}) + i_{\text{in}}(V_{\text{GS}} = 0,7 \text{ V}) - i_d(V_{\text{out}} = V_{\text{OL}}) = i_d(V_{\text{out}} = V_{\text{OH}}) \quad (3.9)$$

onde o primeiro membro da equação representa a corrente de carga fornecida por  $J_E$ , subtraída a corrente de  $J_D$ , e o segundo membro representa a corrente de descarga de  $J_D$ , quando  $J_E$  está cortado. A equação (3.9) é resolvida substituindo-se as correntes por suas expressões segundo o modelo de Statz, com  $L_E = 1 \mu\text{m}$  e  $L_D = 2 \mu\text{m}$ , e encontrando um valor de  $W_E/W_D$  que satisfaça a igualdade resultante.

Um valor de  $\beta = 3$  para o seguidor de fonte resulta em um compromisso ótimo de velocidade *versus*  $V_{\text{pp}}$  [66] para as portas SDCFL, mantendo a intersecção entre as curvas de  $V_{\text{in}}$  e  $V_{\text{out}}$  no ponto médio da excursão (400 mV), como pode ser visto na figura 3.60. Portanto, um valor de 3 será **inicialmente** adotado para o  $\beta$  dos *buffers* SDCFL neste trabalho.

A figura 3.61 mostra a resposta transiente de um inversor SDCFL em 1 GHz, parametrizada para vários valores de  $W_D$ . Para essa mesma simulação, mediu-se os valores dos tempos de atraso, que estão plotados em função de  $W_D$  na figura 3.62. As dimensões do inversor SDCFL e a carga acionada foram os mesmos do Programa 21, mas esta simulação transiente é feita pelo Programa 22.

As formas de onda da figura 3.61 revelam que, para a carga total usada na simulação, o buffer precisaria ter  $W_D \geq 30 \mu\text{m}$  para fornecer um sinal de saída válido. A figura 3.62 mostra que o  $t_{PD}$  do inversor SDCFL possui um valor mínimo para um certo  $W_D$  do *buffer*, e  $W_D$ 's maiores tornam a aumentar o  $t_{PD}$ . Isto confirma o que foi dito acima:  $t_{PD}$  não é monotonicamente decrescente com  $W_D$ , e o valor ótimo de  $W_D$  depende de  $C_L$ .

A equação (3.5) mostra que, para ser mantida uma mesma taxa  $dv/dt$ , a corrente  $I$ , e portanto  $W_D$ , devem ser aumentadas pelo mesmo fator que a carga  $C_L$ , como resumido na equação (3.10). Aplicando esse escalonamento simples e linear, o projetista já pode chegar no dimensionamento final do *buffer*, desde que não sejam ultrapassadas as restrições de área e/ou potência a que o projeto esteja submetido. Se for necessário, após o escalonamento linear, podem ser feitos os “ajustes finos”, apenas para se otimizar algum parâmetro de desempenho que ainda não esteja satisfatório.

$$\frac{I_2}{I_1} = \frac{C_{L2}}{C_{L1}} \Rightarrow \frac{W_{D2}}{W_{D1}} \quad (3.10)$$

Os atrasos de propagação  $t_{PLH}$  e  $t_{PHL}$  também são proporcionais aos tempos  $t_r$  e  $t_f$  do sinal de entrada, e ao FO acionado pelo seguidor de fonte, daí a importância de simulações realistas [66]. A figura 3.63 mostra o efeito do fan-out na resposta transiente do inversor SDCFL, onde se confirma que os atrasos  $t_{PHL}$  e  $t_{PLH}$  aumentam quase linearmente com o aumento de FO (e de  $C_L$ ). No *chip 7490-like*, há situações de FO's e  $C_L$ 's elevados, em que um sinal tem que ser aplicado a um grande número de portas de EFETs, ou a uma longa linha de metal até outro bloco do *chip*. Para acionar o diodo de porta de cada EFET, a fonte desse sinal deve ser um seguidor de fonte com  $W_D$  e  $W_E$  dimensionados para manter um  $V_{OH}$  seguro. Da mesma forma, os pads podem ser vistos como grandes capacitâncias, sendo acionados por seguidores de fonte bastante largos.

O inversor SDCFL simulado teve uma porta inversora com  $W_D = 8 \mu\text{m}$  e  $\beta = 5$ , e um *buffer* com  $W_D = 20 \mu\text{m}$ ,  $\beta = 3$ . Os inversores de fan-out tinham  $W_D = 8 \mu\text{m}$  e  $\beta = 10$ , e o FO foi variado de 1 a 4. A simulação é especificada pelo Programa 23.

A figura 3.64 [5] compara NMs e  $t_{PD}$ 's, de um inversor DCFL e de um SDCFL, em função do FO. Percebe-se as NMs muito maiores do inversor SDCFL, que no entanto possuem a mesma taxa de redução em função do FO que as NMs do inversor DCFL. A figura 3.65 [5] também compara NMs e  $t_{PD}$ 's de DCFL e SDCFL, mas em função das razões entre as larguras de seus transistores. Estas duas figuras servem apenas para efeito de comparação e verificação de taxas de variação desses dois parâmetros.

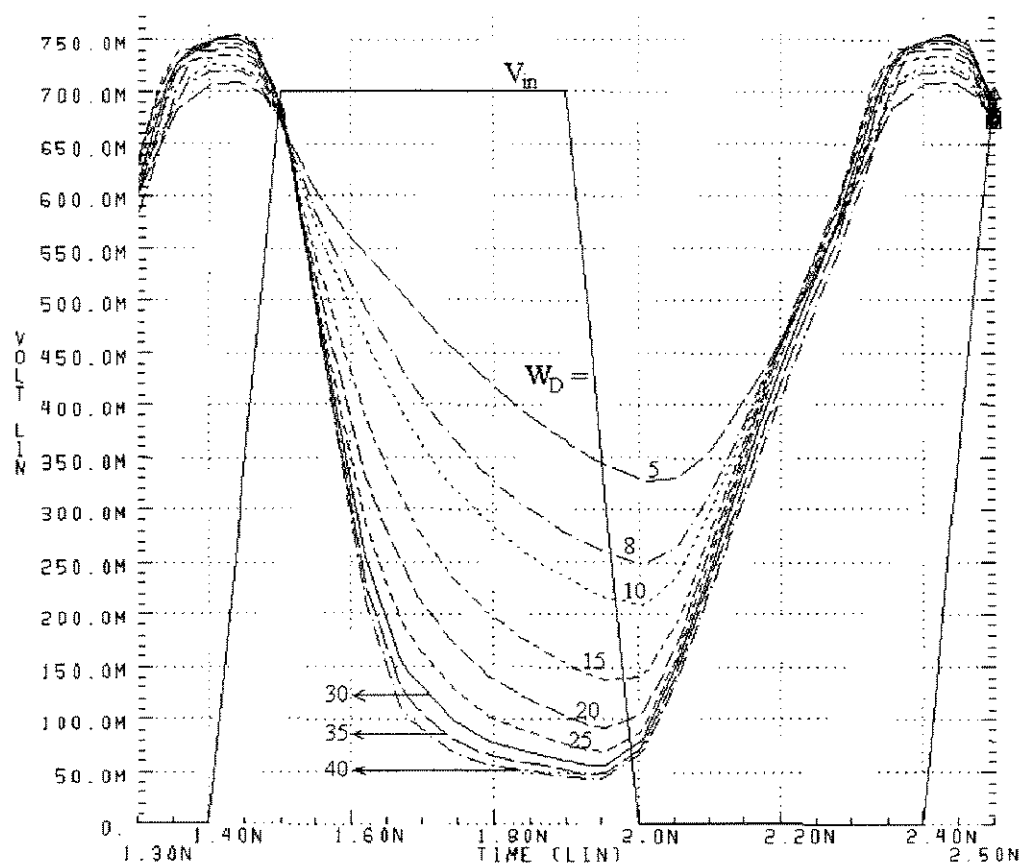


Figura 3.61 – Resposta transiente de um inversor SDCFL parametrizada em  $W_D$ , com  $\beta = 3$ .

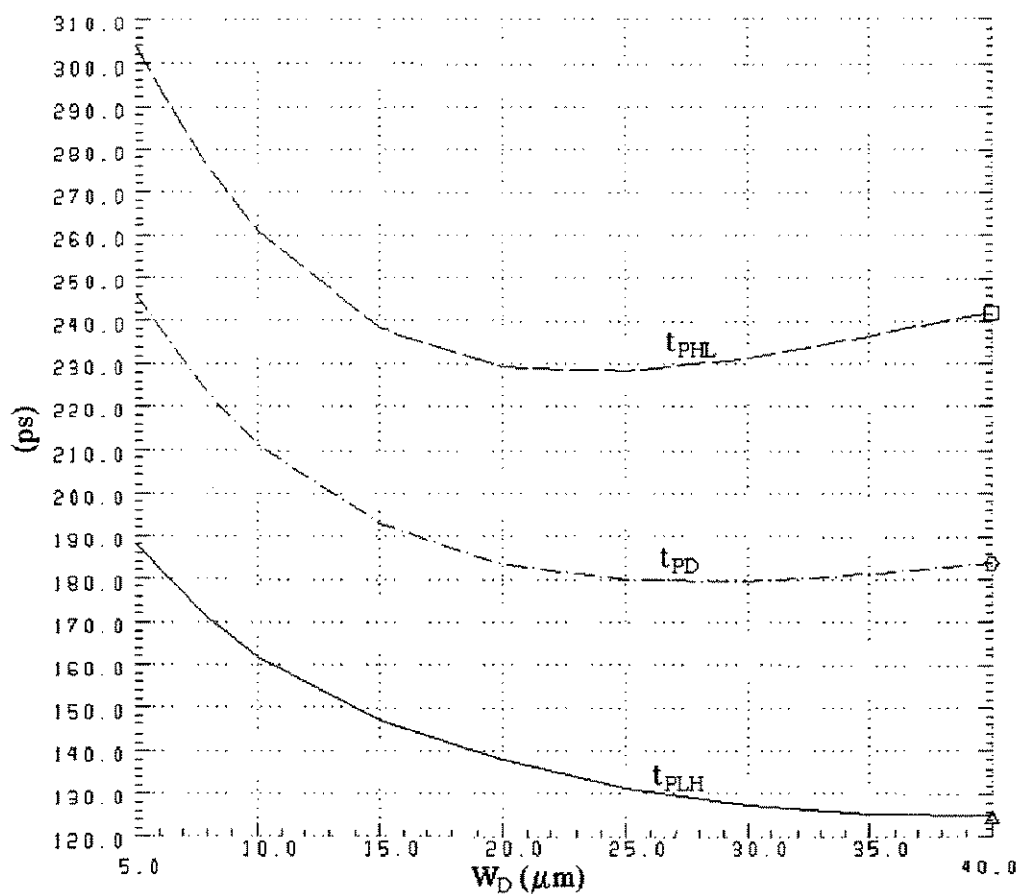


Figura 3.62 – Tempos de atraso do inversor SDCFL em função de  $W_D$  do *buffer*.

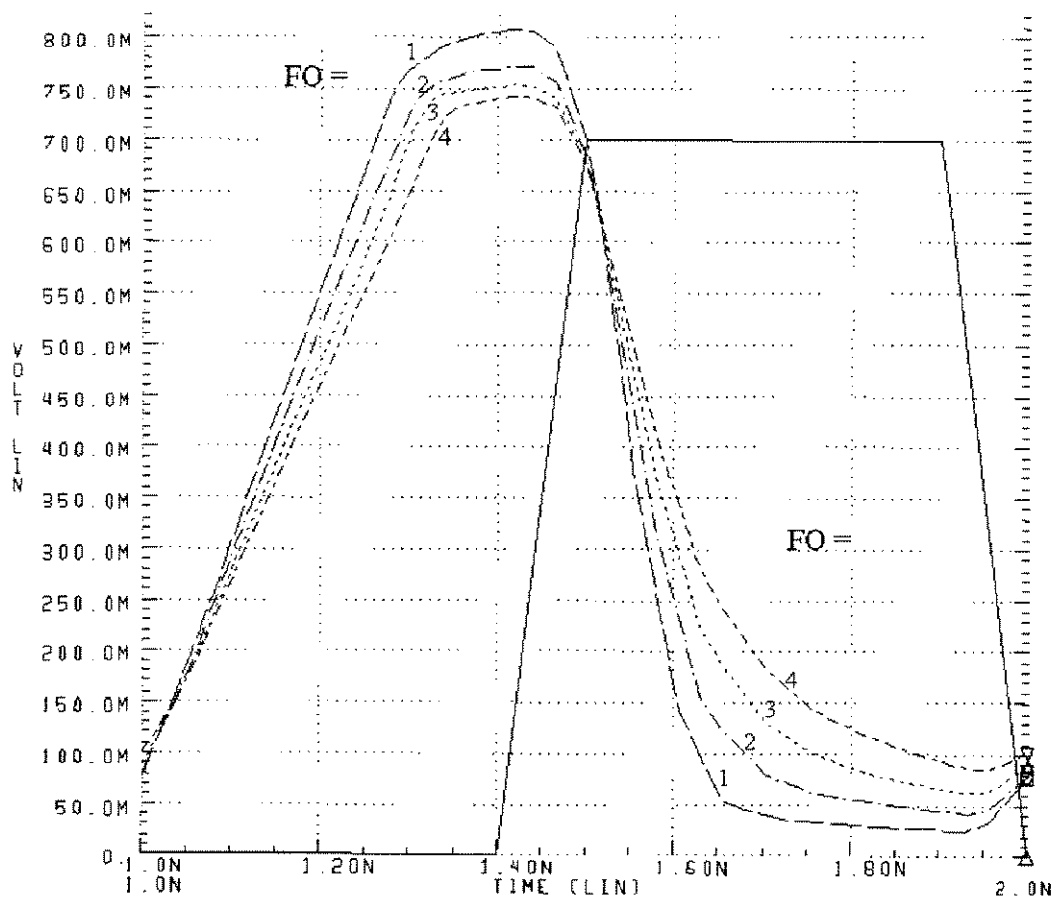


Figura 3.63 – Efeito do FO sobre os atrasos do sinal de saída do inversor SDCFL.

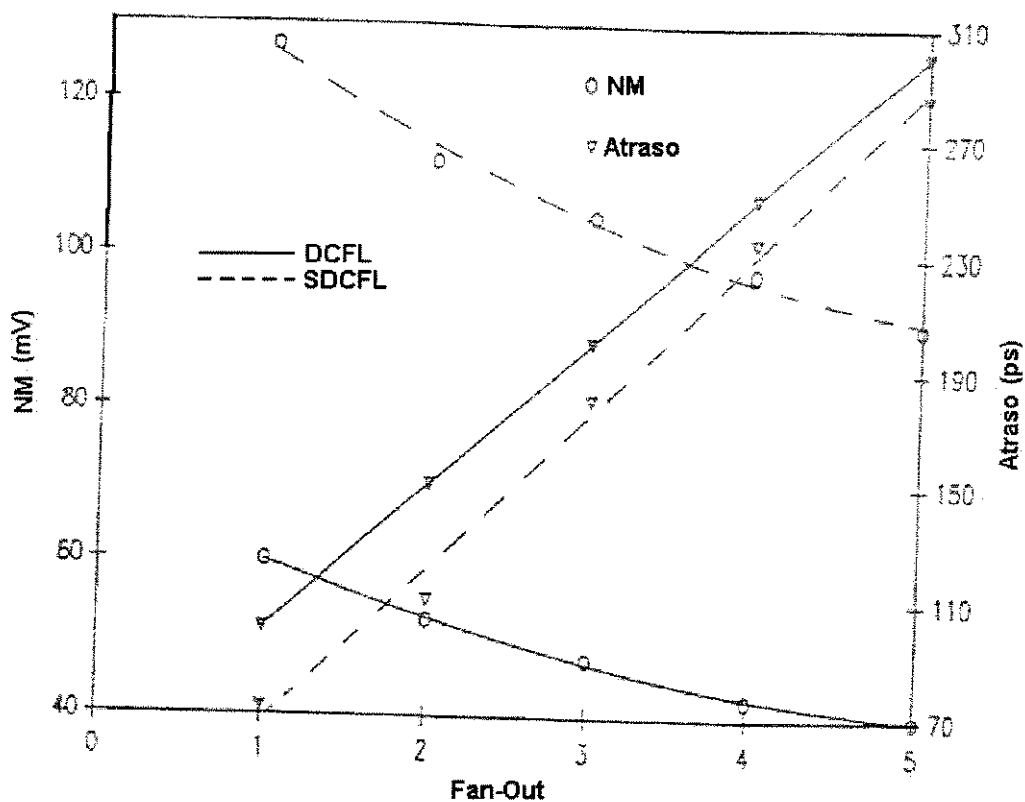


Figura 3.64 – NM's e  $t_{PD}$ 's de inversores DCFL e SDCFL, em função do fan-out [5].

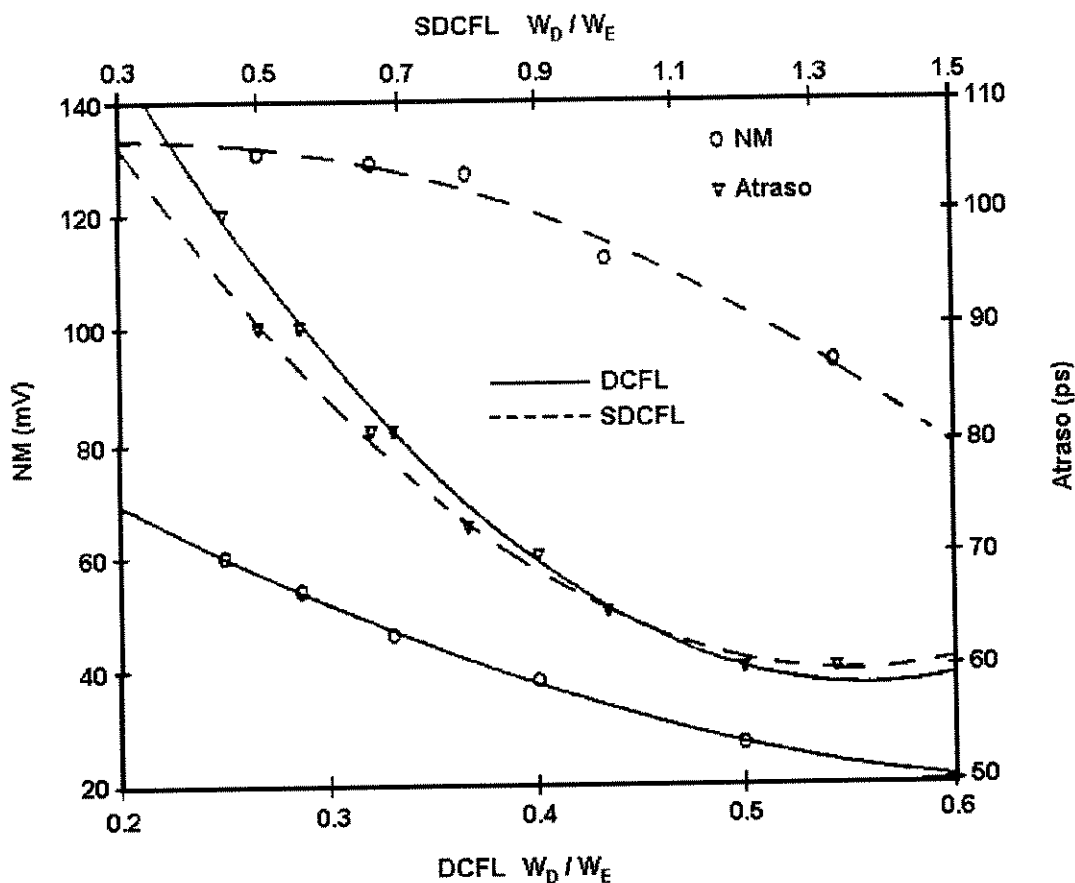


Figura 3.65 – NMs e  $t_{PD}$ 's para um inversor DCFL e um seguidor de fonte SDCFL, em função das suas razões entre larguras.

Existem também outros tipos de *drivers*. O chamado “superbuffer”, mostrado na figura 3.66, é muito adequado para acionar cargas capacitivas ainda mais elevadas, porque produz um surto de corrente muito grande em sua saída; o superbuffer é mais rápido que o *buffer* SDCFL para uma mesma  $C_L$ . No entanto, ele também gera muito ruído nas linhas de alimentação do *chip*, porque provoca picos de corrente bastante altos nessas linhas, durante suas transições lógicas. Por exemplo, durante a transição de descida da saída, os dois FETs do *buffer* ficam em condução durante um tempo igual ao  $t_{PD}$  do inversor que o aciona, gerando um pulso de corrente entre o  $V_{DD}$  e o terra. Este pulso de corrente gera pulsos de tensão nessas linhas, através de suas resistências ôhmicas e suas indutâncias, e esses pulsos de tensão degradam a NM de famílias lógicas com entrada em fonte-comum, como é o caso de DCFL. Assim, o ruído de tensão de terra gerado pelos superbuffers é significativo [11] e pode até levar a transições indevidas em outras portas lógicas do CI ligadas à mesma linha de terra, que recebem o mesmo pulso de tensão. Além disso, os superbuffers apresentam consumo excessivo de potência [78].

Os superbuffers precisam ter seus próprios pads e barramentos de alimentação, separados da alimentação do resto do circuito, para não contaminá-la com o seu ruído de tensão. Esses pads e pinos independentes são chamados de alimentações “suja”, designados por  $V_{DDO}$ . Nesse caso, os pads e barramentos das tensões e terras lógicas ( $V_{DDL}$ ) devem ser independentes de  $V_{DDO}$ . Por causa desses inconvenientes, e considerando-se que as NMs de DCFL já são baixas, o superbuffer não foi usado neste circuito.

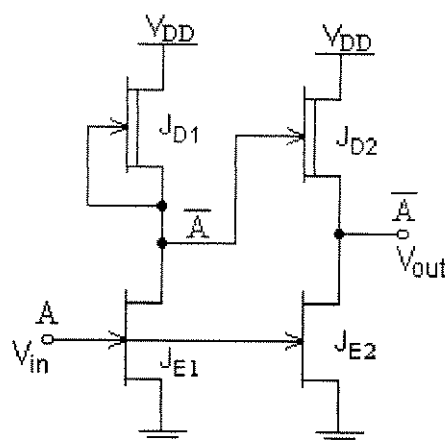


Figura 3.66 – Superbuffer.

Para concluir esta seção, destaca-se que os estágios de saída seguidores de fonte restauram os mesmos níveis lógicos da entrada do CI, e têm a grande vantagem de proporcionar uma resistência de saída bem baixa,  $R_{out} = 1/g_m$  aproximadamente, para o acionamento eficiente da sua carga. Para realizar-se *drivers* de baixa impedância de saída, larguras muito altas dos FETs são necessárias.

É interessante observar que, para aumentos de largura dos transistores do *buffer* em uma boa faixa, não é necessário que o DFET do parte inversora DCFL seja aumentado conjuntamente, pois, como visto na figura 3.54, o *buffer* puxa uma corrente de entrada muito baixa mesmo quando tem dimensões grandes. Ele representa uma carga muito leve para o inversor DCFL.

### 3.4.2. Projeto dos *Buffers* de Entrada.

Um *buffer* de entrada deve proteger os circuitos lógicos internos do *chip* contra tensões na entrada que estejam fora da faixa normal de operação do CI, limitando-as. E mesmo com uma  $V_{in}$  excessiva, a corrente de entrada nunca deve ser excessiva, o que poderia danificar trilhas internas ou os contatos de *gate* dos FETs, inutilizando-os. Sua faixa de operação recomendada é de  $GND \leq V_{in} \leq V_{DD}$  ou  $V_{TT} \leq V_{in} \leq GND$ , conforme a alimentação seja positiva ou negativa respectivamente; quando forem especificados os limites máximo e mínimo da  $V_{in}$ , o *buffer* de entrada deve ser dimensionado adequadamente. Para este projeto, sabe-se que os *buffers* de entrada devem ser projetados para apresentarem impedâncias de entrada de  $50 \Omega$ .

O mais importante é que o CI deve ser compatível consigo mesmo, isto é, os *buffers* de entrada têm que ser compatíveis com os *buffers* de saída. Isto é função do projeto de ambos. Foram consideradas e comparadas duas opções de *buffers* de entrada. Em primeiro lugar, foi simulado um *buffer* que consistia de diodos de proteção e um resistor limitador de corrente, mostrado na figura 3.67.

A função de  $R_{Lim}$  é limitar a corrente nos diodos a valores seguros. Porém, a presença desse resistor em série com a entrada, causa um atraso proporcional ao valor da sua resistência, e limita a frequência máxima de entrada no circuito (introduz uma frequência de corte que limita a resposta em frequência). A partir de uma certa frequência, a tensão no nó 2

não consegue mais acompanhar a tensão aplicada no pino do CI (1), por causa do atraso RC. Por isso, o *buffer* da figura 3.67 foi abandonado para este projeto.

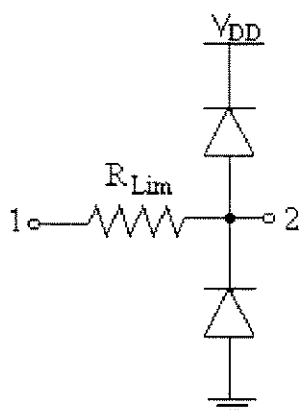


Figura 3.67 – *Buffer* de entrada simples.

Concluiu-se que o melhor circuito *buffer* de entrada para este projeto consiste de um DFET conectado como fonte de corrente (com  $V_{GS} = 0$ ), atuando portanto como um regulador de corrente de entrada. Essa é a topologia de circuito de proteção adotada nos CIs da GigaBit Logic [64], e é mostrada na figura 3.68. Quanto maior a  $W_D$  do DFET, maior a velocidade de resposta do *buffer* (menor o seu atraso); porém, se  $W_D$  for muito grande, a corrente do regulador para a entrada do bloco seguinte fica muito alta. Portanto,  $W_D$  precisa ser dimensionada com cuidado para evitar sobrecarga (*overdrive*) do E-FET do estágio seguinte; se  $W_D$  for pequena, o DFET pode limitar a resposta em alta frequência de todo o circuito: a *slew rate* de saída torna-se insuficiente, com severas limitações aos tempos de transição, e não será mais possível entrar com o sinal no *chip*.

O regulador de corrente é conectado em série com as entradas de sinal do *chip*. Ele mantém a corrente de entrada confinada a uma faixa, mesmo com  $V_{in}$  fora da faixa  $V_{DD}$ –GND ou GND– $V_{TT}$ , portanto dispensa diodos de proteção, pois já garante uma proteção eficiente contra sobre- ou sub- tensões.

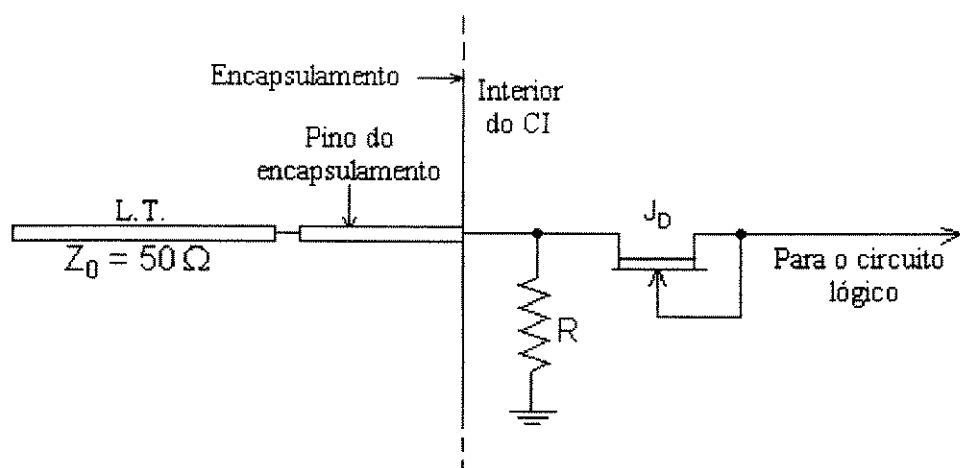


Figura 3.68 – Bloco de proteção de entrada do *chip* 7490-like.

Para finalizar, é instrutivo mencionar que o *buffer* da figura 3.69 também foi empregado como bloco de proteção de entrada em versões iniciais do projeto, mas percebeu-se, nas simulações, que ele não serviria para o circuito 7490-like. O problema é que seu  $V_{IH}$  é



maior do que 1,4 V, mas o  $V_{OH}$  dos *drivers* de saída SDCFL do chip é igual a 1,2 V. Portanto, com o *buffer* da figura 3.69, o CI 7490-like deixava de satisfazer ao gráfico fundamental da figura 3.1 e perdia a característica regenerativa dos circuitos digitais, pela qual  $V_{OH}$  precisa ser maior que  $V_{IH}$ .

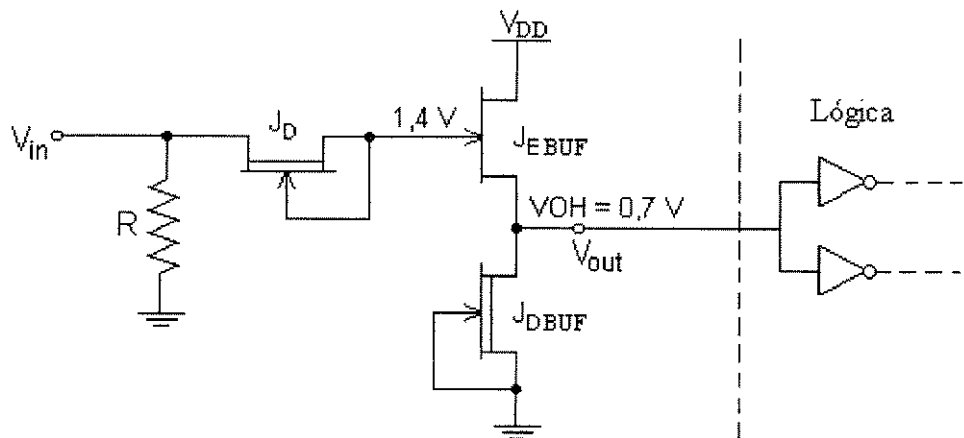


Figura 3.69 – *Buffer* inadequado para o CI 7490-like, por ter níveis lógicos incompatíveis.

## CAPÍTULO 4.

### Projeto do CI Contador e Divisor de Frequências 7490-like.

A seção 4.1 descreve a arquitetura interna e as várias configurações de operação do CI TTL 7490, no qual se baseia o projeto deste trabalho. A seção 4.2 descreve o projeto lógico (isto é, funcional) do CI 7490-like, para a tecnologia de MESFETs de GaAs, na família DCFL. O projeto lógico é feito de tal forma que o CI proposto mantém toda a funcionalidade original do 7490, podendo ser usado tanto como um divisor da frequência do sinal de entrada por  $N$ , sendo  $N$  um inteiro na faixa de 2 até 10, quanto como um contador de década (BCD). A razão da divisão  $N$ , pode ser configurada unicamente através de conexões **diretas** entre pinos do CI, sem a necessidade de qualquer componente externo. As simulações lógicas em computador, empregando o *software* HSPICE, comprovaram a funcionalidade.

A seção 4.3 descreve o dimensionamento dos MESFETs das portas lógicas e *buffers* do circuito, para a operação em frequências de *clock* de até 1 GHz. O projeto de um flip-flop (FF) capaz de atender aos requisitos de projeto do 7490-like é descrito, e em seguida explica-se como o circuito foi dimensionado nas simulações finais, antes do desenho do *layout*.

Todo o projeto do CI é descrito neste capítulo; trata-se de um projeto *full-custom*, com três níveis de hierarquia e realizado em uma metodologia *bottom-up*, a qual é eficiente para projetos de baixa complexidade como este (menos de 10k portas) [79].

#### 4.1. Contador de Década TTL 7490.

O 7490, da família bipolar TTL em Si, é um circuito contador de década / divisor por  $N$  assíncrono, com  $N$  podendo ser ajustado na faixa de 2 até 10. O fato de ser um contador assíncrono lhe confere uma ótima flexibilidade, a ponto de ele nunca necessitar de qualquer componente externo para as suas diversas configurações como divisor de frequências por  $N$ . Além disso, um contador síncrono possui um maior consumo de potência do que um contador assíncrono de mesmo módulo, porque no contador síncrono todos os FFs têm que operar na mesma taxa de clock, e existe uma relação direta entre o consumo de potência e a taxa de operação das células.

A pinagem e um esquema lógico interno do 7490 são mostrados nas figuras 4.1a e 4.1b. Trata-se de um circuito integrado em escala MSI, composto de quatro flip-flops Mestre-Escravo (MS) gatilhados na descida do sinal de *clock*, configurados de modo a constituírem um divisor 2 e um divisor por 5 independentes entre si, e mais uma lógica combinacional extra para permitir as diferentes configurações de divisão de frequências. Os quatro FFs são *resetados* para 0000 se ambas as entradas  $R0$  são colocadas em nível lógico '1', e são *presetados* para 1001 ( $9_{10}$ ) se ambas as entradas  $R9$  são colocadas em nível lógico '1'. Assim, as duas entradas  $R0$  constituem um *clear* ativo em nível alto, e as duas entradas  $R9$  constituem um *preset* para 9, também ativo em nível alto.

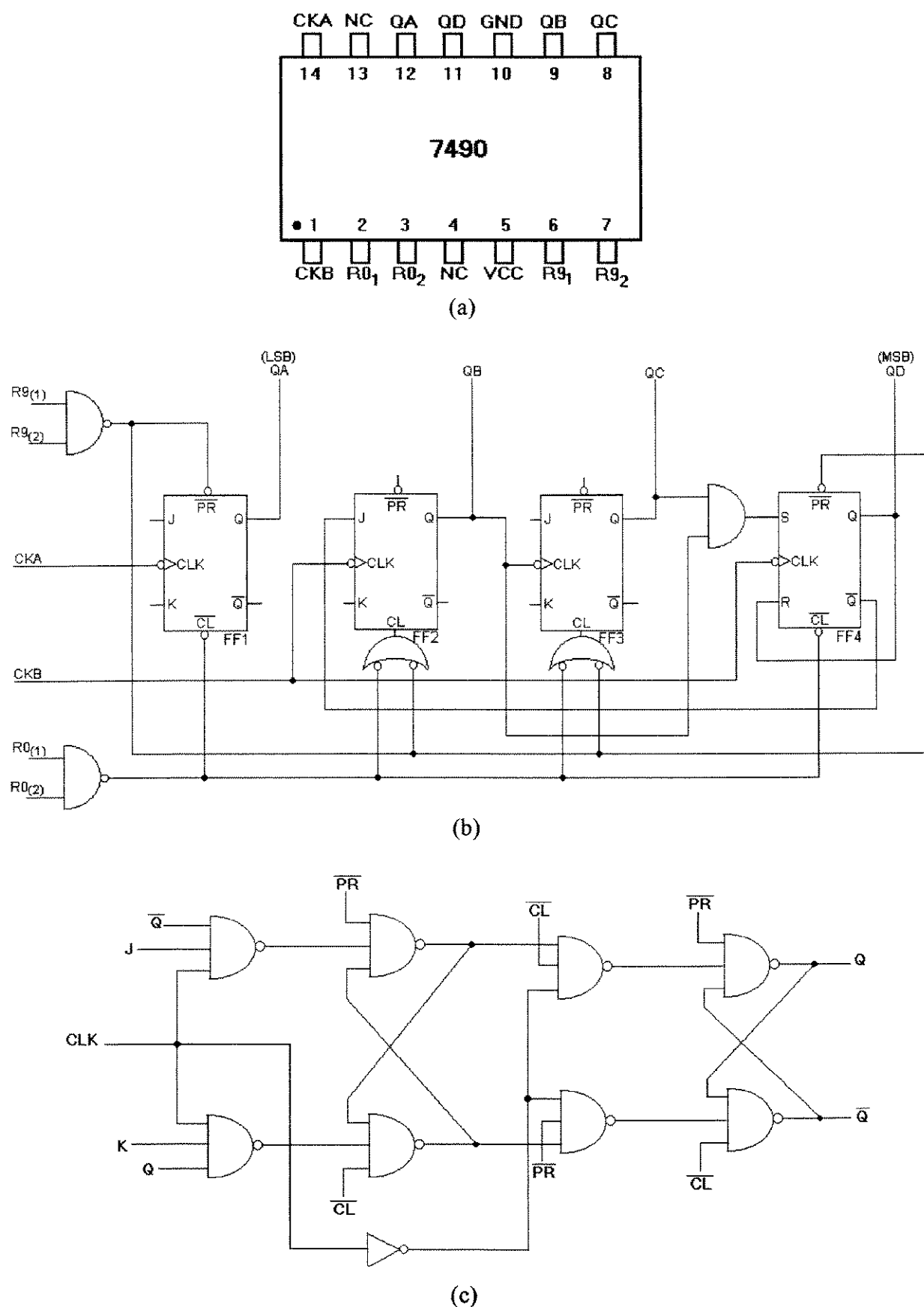


Figura 4.1 - (a) Pinagem e (b) esquema lógico do TTL 7490, e (c) esquema lógico de um dos Flip-flops JK Mestre-Escravo com *preset* e *clear*.

Na figura 4.1b, as entradas J, K e  $\overline{PR}$  mostradas sem conexão estão eletricamente em nível lógico ‘1’, e são apenas para referência, já que no CI real elas não existem. O 2º e o 3º FFs tem *clear* ativo em nível alto. O esquema lógico de um dos FFs JK MS, com *preset* e *clear*, é mostrado na figura 4.1c. O FF SR é semelhante ao JK, com a diferença de que no SR não há a realimentação das saídas Q e  $\overline{Q}$  para as portas das entradas K e J. Por isso, o FF SR tem um estado proibido: as entradas S e R não podem ser ambas iguais a 0.

As folhas de dados da Texas Instruments para o 7490 ([80], [81]) descrevem apenas três modos de operação do mesmo; nas tabelas 4.1 e 4.2, cada estado surge após uma transição negativa do *clock*:

1) Para ser utilizado como um contador BCD (de 0 a 9 em decimal), isto é, como um contador de década, a saída QA deve ser externamente conectada à entrada CKB, e os pulsos de contagem (sinal de *clock*) devem ser aplicados à entrada CKA. As saídas apresentarão uma contagem de 0 a 9, como mostra a tabela 4.1.

| Valor Decimal | QD | QC | QB | QA |
|---------------|----|----|----|----|
| 0             | 0  | 0  | 0  | 0  |
| 1             | 0  | 0  | 0  | 1  |
| 2             | 0  | 0  | 1  | 0  |
| 3             | 0  | 0  | 1  | 1  |
| 4             | 0  | 1  | 0  | 0  |
| 5             | 0  | 1  | 0  | 1  |
| 6             | 0  | 1  | 1  | 0  |
| 7             | 0  | 1  | 1  | 1  |
| 8             | 1  | 0  | 0  | 0  |
| 9             | 1  | 0  | 0  | 1  |

Tabela 4.1 – Contagem BCD do TTL 7490.

2) Para ser utilizado como um divisor por 10 simétrico, a saída QD deve ser conectada externamente à entrada CKA, e o sinal de *clock* deve ser aplicado à entrada CKB. Então a saída QA fornecerá uma onda quadrada com 1/10 da frequência do sinal de *clock* e um ciclo de trabalho de 50% (daí ser um divisor simétrico). Observando-se as palavras de saída na tabela 4.2, nota-se que o 7490 conta alternadamente em números pares e ímpares.

| Valor Decimal | QD | QC | QB | QA |
|---------------|----|----|----|----|
| 0             | 0  | 0  | 0  | 0  |
| 2             | 0  | 0  | 1  | 0  |
| 4             | 0  | 1  | 0  | 0  |
| 6             | 0  | 1  | 1  | 0  |
| 8             | 1  | 0  | 0  | 0  |
| 1             | 0  | 0  | 0  | 1  |
| 3             | 0  | 0  | 1  | 1  |
| 5             | 0  | 1  | 0  | 1  |
| 7             | 0  | 1  | 1  | 1  |
| 9             | 1  | 0  | 0  | 1  |

Tabela 4.2 – Divisão por 10 simétrica do TTL 7490.

3) O FF1 pode ser usado como um divisor por 2 do sinal aplicado em CKA, e os FFs 2, 3 e 4 podem ser usados como um divisor por 5 (contador de 0 até 4), com entrada em CKB. Não há conexões entre pinos do CI.

Também são possíveis divisões de frequência por outros valores na faixa de 2 a 10, sem o acréscimo de componentes externos, embora as folhas de dados ([80], [81]) não mostrem as ligações para a realização dessas divisões e nem sequer mencionem essas possibilidades. Para suprir essa deficiência, a tabela 4.3 apresenta todas as configurações de divisão por N do 7490, de N = 2 até N = 10, e indica quais as ligações que devem ser feitas entre os pinos do CI para a sua operação em cada uma dessas configurações de divisão.

| N                       | Entrada do sinal | Saída do sinal                        | Ligações necessárias entre terminais                                                                              |
|-------------------------|------------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| 2                       | 14 (CKA)         | 12 (QA)                               | Terminal 2 (ou 3) ao GND;<br>Terminal 6 (ou 7) ao GND.                                                            |
| 3                       | 1 (CKB)          | 8 (QC)                                | Terminal 6 (ou 7) ao GND;<br>Terminal 2 ao terminal 8;<br>Terminal 3 ao terminal 9.                               |
| 3<br>(outra)            | 1 (CKB)          | 9 (QB)                                | Terminal 2 (ou 3) ao GND;<br>Terminal 1 ao terminal 7;<br>Terminal 8 ao terminal 6.                               |
| 4                       | 1 (CKB)          | 8 (QC)                                | Terminal 6 (ou 7) ao GND;<br>Terminais 2 e 3 ao terminal 11.                                                      |
| 5                       | 1 (CKB)          | 8 (QC) ou<br>11 (QD)                  | Terminal 2 (ou 3) ao GND;<br>Terminal 6 (ou 7) ao GND.                                                            |
| 6                       | 14 (CKA)         | 8 (QC)                                | Terminal 6 (ou 7) ao GND;<br>Terminal 1 ao terminal 12;<br>Terminal 9 ao terminal 2;<br>Terminal 8 ao terminal 3. |
| 7                       | 14 (CKA)         | 8 (QC) ou<br>11 (QD)                  | Terminal 2 (ou 3) ao GND;<br>Terminal 1 ao terminal 12;<br>Terminal 7 ao terminal 8;<br>Terminal 6 ao terminal 9. |
| 8                       | 14 (CKA)         | 8 (QC)                                | Terminal 6 (ou 7) ao GND;<br>Terminal 1 ao terminal 12;<br>Terminais 2 e 3 ao terminal 11.                        |
| 9                       | 14 (CKA)         | 8 (QC) ou<br>11 (QD)                  | Terminal 6 (ou 7) ao GND;<br>Terminais 1 e 2 ao terminal 12;<br>Terminal 3 ao terminal 11.                        |
| 10<br>(contagem<br>BCD) | 14 (CKA)         | 12, 9, 8 e 11<br>(QA, QB, QC e<br>QD) | Terminal 2 (ou 3) ao GND;<br>Terminal 6 (ou 7) ao GND;<br>Terminal 1 ao terminal 12.                              |
| 10 simétrico            | 1 (CKB)          | 12 (QA)                               | Terminal 2 (ou 3) ao GND;<br>Terminal 6 (ou 7) ao GND;<br>Terminal 11 ao terminal 14.                             |

Tabela 4.3 – Como realizar as configurações de divisão de frequências do CI TTL 7490. As entradas não utilizadas devem ser ligadas em GND.

Ao dividir uma frequência por um número N, obtém-se um pulso na saída do contador para cada N pulsos na entrada. Para cada valor de N, as entradas do CI são conectadas de

maneira que o ciclo nas saídas se reinicie depois de N pulsos. Para N = 3 existe mais de uma configuração correta [82].

### 4.2. Projeto Lógico do Circuito Integrado 7490-like.

O flip-flop JK é uma definição lógica, portanto a sua tabela-verdade, mostrada na tabela 4.4, é independente do tipo de portas lógicas usadas para a sua implementação. Todo FF que possuir uma tabela-verdade igual à da tabela 4.4 é um FF JK, independentemente da sua topologia lógica ou da tecnologia em que estiver implementado.

| J | K | Q <sub>n+1</sub> |
|---|---|------------------|
| 0 | 0 | Q <sub>n</sub>   |
| 0 | 1 | 0                |
| 1 | 0 | 1                |
| 1 | 1 | $\overline{Q_n}$ |

Tabela 4.4 – Tabela-verdade do FF JK.

Os circuitos sequenciais nas famílias TTL e CMOS são baseados em lógica NAND, isto é, todos os seus tipos de flip-flops são implementados somente com portas NAND e inversores (NOT) ([2], [79], [83]), como o FF JK da figura 4.1c. Isto representa um problema para a adaptação dessas topologias para a família lógica DCFL, porque nesta somente são implementadas com facilidade as portas NOR e inversores. Portas NAND raramente são usadas em circuitos DCFL, pelas razões explicadas no Capítulo 2.

Uma primeira tentativa de se contornar esse problema, para preservar a mesma funcionalidade padrão dos FFs TTL e CMOS, consistiu em se construir portas NAND a partir de portas NOR e inversores, de acordo com o teorema de DeMorgan. Dispõe-se de  $\overline{A + B}$ ; esse teorema mostra como se expressa uma operação NAND em termos de NOR e NOT:

$$\overline{AB} = \overline{A + B} = \overline{\overline{\overline{A + B}}} \tag{4.1}$$

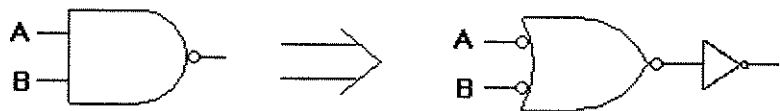


Figura 4.2 - Representação gráfica de um dos teoremas de DeMorgan.

Entretanto, essa adaptação traz um sério inconveniente, pois aumenta muito o número de FETs por porta lógica, aumentando demais a área, o consumo de potência e o atraso de cada porta. A figura 4.3 mostra uma porta NAND DCFL construída a partir da equivalência lógica da figura 4.2, onde pode-se perceber que, para aumentar o seu fan-in (FI), um inversor a mais é necessário para cada entrada adicional. Enfim, essas portas NAND adaptadas empregam (2.FI + 2) FETs a mais do que uma porta NOR DCFL de mesmo fan-in. Se o FF JK da figura 4.1c fosse construído com as portas NAND da figura 4.3, ele teria 64 FETs a mais do que se fosse feito somente com portas NOR.

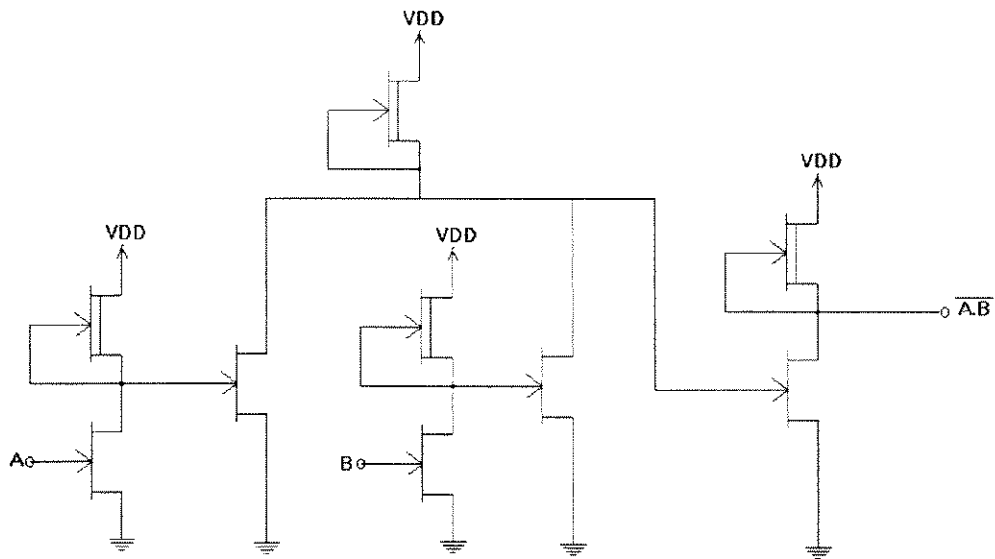


Figura 4.3 - Uma porta NAND de duas entradas adequada para a lógica DCFL, realizada a partir de uma porta NOR e inversores.

Tantos MESFETs a mais são um enorme problema para se realizar um circuito completo, pois significam uma área de *die* muito grande, excessivo consumo de potência e um atraso muito maior do circuito. Como a principal razão para se empregar dispositivos de GaAs é alcançar o melhor desempenho possível para um circuito, o projeto deve ser criterioso de modo a aproveitar essa velocidade, e não limitá-la. Como o objetivo deste projeto é fazer-se um contador em DCFL que apresente um certo comportamento, bem definido, a nível de terminais externos (isto é, a mesma funcionalidade do TTL 7490), pode-se pensar em projetá-lo com flip-flops feitos unicamente de portas NOR e inversores. Depois procura-se compensar internamente as diferenças de lógica entre as versões TTL e DCFL dos dois contadores, fazendo-se adaptações para que, externamente, a operação de ambos seja a mesma.

E assim foi feito: mantendo a mesma topologia do FF da figura 4.1c, mas trocando todas as portas NAND por portas NOR DCFL, obtém-se o FF MS J'K' da figura 4.4.

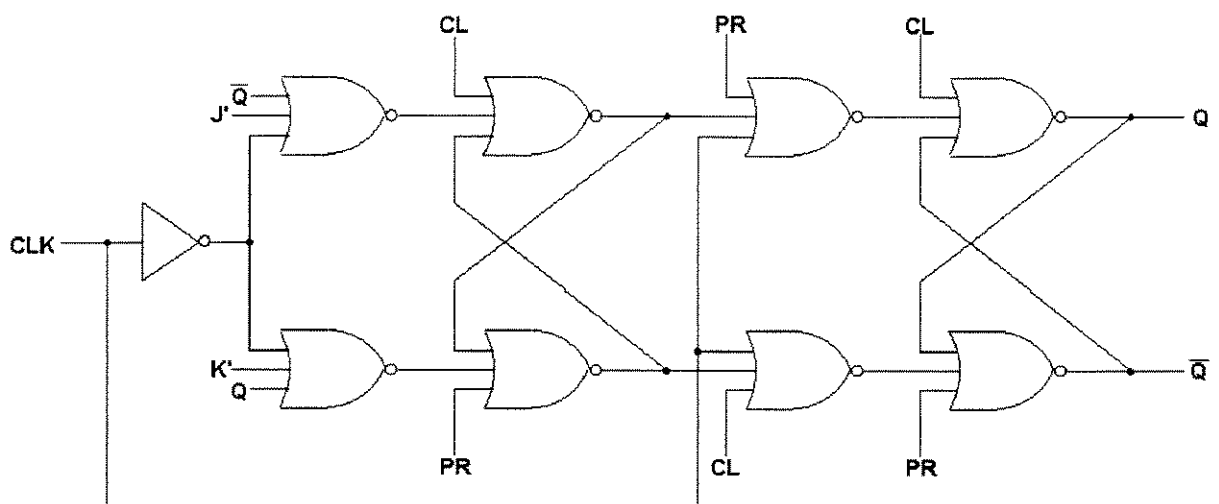


Figura 4.4 - Flip-flop Mestre-Escravo J'K' com *Preset* e *Clear*, feito com portas NOR.

As diferenças de operação entre este flip-flop e o FF JK da figura 4.1c são:

1) A tabela-verdade do flip-flop J'K' é:

| J' | K' | $Q_{n+1}$        |
|----|----|------------------|
| 0  | 0  | $\overline{Q_n}$ |
| 0  | 1  | 0                |
| 1  | 0  | 1                |
| 1  | 1  | $Q_n$            |

Tabela 4.5 – Tabela-verdade do FF J'K' da figura 4.4.

2) *Preset* e *Clear* são ativos em nível alto, e não devem estar ambos em nível lógico '1' simultaneamente. Quando  $CL=1$ ,  $Q=0$  e  $\overline{Q}=1$ ; quando  $PR=1$ ,  $Q=1$  e  $\overline{Q}=0$ .

3) O inversor que, na versão NAND do FF JK, fica na entrada do segundo *latch*, deve ser colocado na entrada do primeiro *latch*, para tornar o FF J'K' gatilhado na borda de descida do *clock*; sem essa mudança da posição do inversor, este FF seria gatilhado na subida do *clock*. Em comparação, os flip-flops JK NAND da figura 4.1c são naturalmente gatilhados na descida do *clock*. As entradas de *Preset* e *Clear* são ativas em nível baixo, e não podem ser ambas iguais a 0.

Se for necessário, o FF J'K' pode ser convertido em um FF JK, por meio do arranjo mostrado na figura 4.5. Mas, neste trabalho, esta adaptação não foi necessária, e o circuito 7490-like foi projetado mesmo com os FFs J'K'.

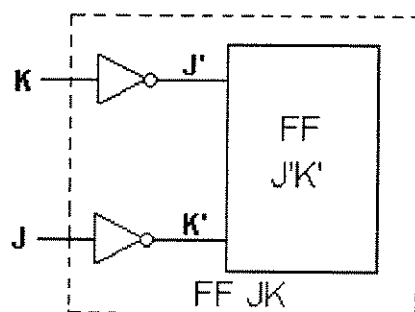


Figura 4.5 – Como obter um FF JK a partir de um FF J'K'.

Realizou-se simulações do flip-flop J'K' para verificar sua operação. Agora pode-se usá-lo para montar um contador de década 7490-like, e depois analisar as diferenças de operação entre os contadores TTL e DCFL.

Empregando-se o recurso de subcircuitos do HSPICE, que permite tratar-se um conjunto de componentes interconectados como um bloco tipo “caixa preta”, com apenas as entradas e saídas visíveis externamente, descreveu-se no *software* o circuito interno do 7490-like, mostrado na figura 4.6. As simulações em HSPICE desse modelo foram efetuadas, e a visualização gráfica dos resultados confirmou que a funcionalidade manteve-se a mesma do 7490 TTL. Por exemplo, para o 7490-like DCFL ser utilizado como um contador BCD, a saída QA também deve ser conectada à entrada CKB, e o sinal de *clock* aplicado à entrada CKA, da mesma forma que o 7490 TTL.

Pode-se observar as mudanças em algumas conexões nesta versão *full-NOR* do 7490, em comparação com o esquema do 7490 TTL da figura 4.1(b). Essas mudanças foram feitas



para compensar as diferenças de tabela-verdade entre os flip-flops JK (NAND) e J’K’ (NOR), de modo que o comportamento a nível de terminais, dos dois contadores, seja idêntico. Ou seja, para tornar as diferenças internas “transparentes ao usuário”. Isso significa que, da perspectiva de um usuário, o CI 7490-like DCFL da figura 4.6 tem funcionalidade idêntica ao CI 7490 TTL.

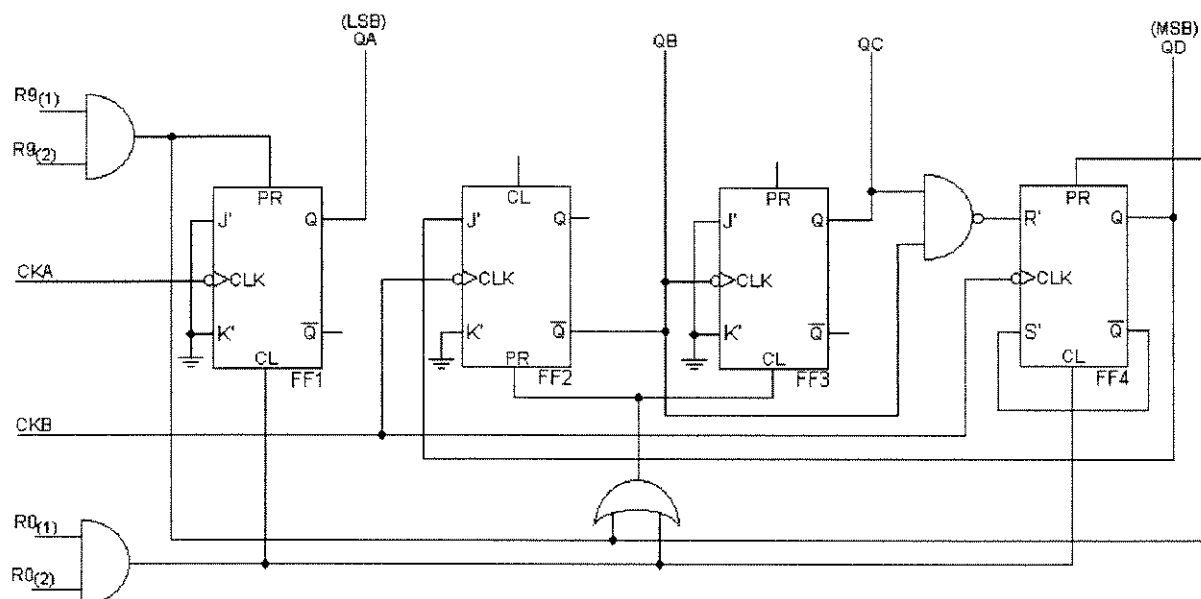


Figura 4.6 – Esquema lógico simplificado do CI contador/divisor DCFL “7490-like” projetado com flip-flops NOR J’K’.

As portas NAND e AND, no circuito do 7490-like, são construídas a partir de portas NOR e NOT usando-se os teoremas de DeMorgan, e todas as portas são implementadas na topologia DCFL. Os FFs são do tipo J’K’ mestre-escravo. A arquitetura do contador permite que o flip-flop FF1 possa ser acessado isoladamente do restante do circuito, o que será importante na etapa de teste do CI.

Foram simuladas, no HSPICE, todas as configurações de divisão de frequência por N (as formas de onda são mostradas no Apêndice 3), bem como a contagem BCD (mostrada na figura 4.7), e os sinais de saída obtidos foram idênticos aos observados nos ensaios de bancada com o CI 7490 TTL. As ligações entre pinos, para cada configuração de divisão, são as mesmas da tabela 4.3. A funcionalidade lógica está comprovada, passa-se agora ao dimensionamento dos MESFETs para alcançar-se um desempenho elétrico especificado.

### 4.3. Projeto Elétrico do Circuito Integrado 7490-like.

#### 4.3.1. Especificações.

A especificação principal para o CI 7490-like é que ele seja capaz de operar em frequências de clock de até 1 GHz, quando no modo de contagem BCD, e com suas saídas acionando uma carga resistiva  $R_L = 50 \Omega$  em série com uma capacitância total  $C_L = 5 \text{ pF}$ . Esta será a carga padrão nas saídas do contador 7490-like, para as simulações de todas as suas configurações de operação, e é uma das condições de projeto para os seus FFs. A capacitância

de um pino de sinal do encapsulamento LDCC-28, escolhido para alojar o chip *7490-like*, é de 4 pF, de acordo com a tabela 6-3 de [12]. O valor de 5 pF usado para  $C_L$  engloba também uma margem de segurança correspondente à capacitância do *pad* e à tolerância do encapsulamento.

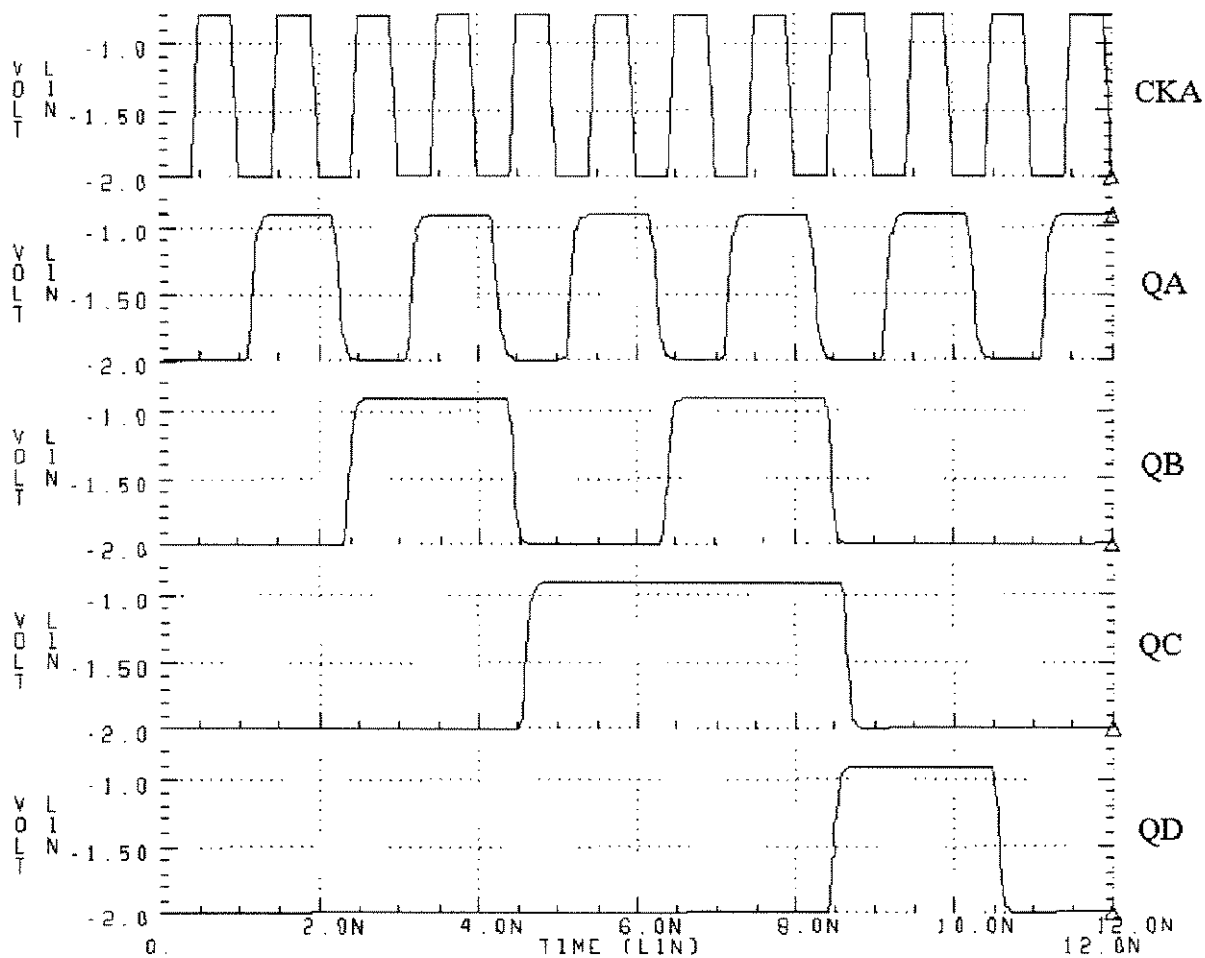


Figura 4.7 – Operação do CI 7490-like como contador BCD, para um clock de 1GHz, com alimentação negativa  $V_{TT} = -2$  V.

O CI 7490-like deverá alcançar esse desempenho temporal especificado empregando uma única fonte de alimentação, que pode ser positiva ( $V_{DD} = 2\text{ V}$ ) ou negativa ( $V_{TT} = -2\text{ V}$ ), com tolerância de 10%. A excursão lógica normal nas entradas deve estar restrita a  $GND \leq V_{in} \leq V_{DD}$ , quando com  $V_{DD} = 2\text{ V}$ , ou a  $V_{TT} \leq V_{in} \leq GND$ , quando com  $V_{TT} = -2\text{ V}$ , e a  $V_{pp}$  nas saídas deve ser no mínimo igual a 1 V. O CI deve operar na faixa comercial de temperaturas de junção, de  $0^\circ\text{C}$  a  $85^\circ\text{C}$ .

É fundamental que todo CI digital a ser projetado obedeça ao gráfico da figura 3.1, o que é determinado exclusivamente pelas características dos seus *buffers* de entrada e de saída. Estes blocos definem as características do CI ao nível de terminais, portanto eles têm que ser padronizados para proporcionarem especificações consistentes de níveis lógicos de tensão de operação, como em qualquer família lógica já estabelecida. Por exemplo, TTL e CMOS têm suas especificações bem definidas para níveis lógicos. E neste projeto, é fundamental que o CI seja compatível consigo mesmo, isto é, que ele tenha níveis de saída compatíveis com os níveis de entrada, porque todas as suas configurações de operação dependem de ligações entre os pinos do próprio CI. Essa exigência orienta o projeto dos circuitos de entrada e saída.

Em um projeto de CI, também é muito importante considerar as questões de compatibilidade com outros CIs, mesmo de outras famílias lógicas. Em tecnologia de silício, a família lógica ECL é mais rápida comercialmente disponível, provendo níveis de desempenho na faixa das centenas de Mbits/s com CIs MSI. Por isso é comum projetar-se CIs de GaAs para terem níveis lógicos compatíveis com os da família ECL, viabilizando o interfaceamento de GaAs com sistemas de Si ([70], [11]), o que é um requisito na maioria dos projetos.

Essas questões de compatibilidade dizem respeito unicamente aos blocos de entrada e saída e à tensão de alimentação. Para obter-se, com este CI DCFL, níveis lógicos de entrada e saída compatíveis com os da família ECL, basta que a alimentação positiva  $V_{DD}$  seja substituída por uma alimentação negativa  $V_{TT} = -2$  V, ligada com polaridade contrária à da fonte  $V_{DD}$ . Os níveis de saída compatíveis com ECL são  $V_{OH} = -0,8$  V e  $V_{OL} = -1,8$  V. Felizmente, os blocos de proteção de entrada e os drivers de saída SDCFL usados neste CI já são intrinsicamente compatíveis com os níveis lógicos ECL, quando é usada a alimentação negativa. As características de desempenho DC e transiente do circuito *7490-like* não são alteradas com a mudança de polaridade da alimentação, da positiva  $V_{DD} = 2$  V para a negativa  $V_{TT} = -2$  V.

Os pinos de entrada de sinal do CI *7490-like* precisam ter uma impedância de entrada de  $50\ \Omega$ . Como será usado o circuito de proteção da figura 3.68, os DFETs reguladores de corrente têm que ser dimensionados para permitirem o ingresso de sinais de 1 GHz com o mínimo de limitação dos tempos de transição (bordas dos sinais), portanto esses DFETs precisam ser bem largos para fornecerem uma alta corrente. Os *buffers* de entrada devem ser dimensionados para que o CI possa tolerar uma faixa de 1,5 V em torno da excursão nominal da entrada  $V_{in}$ , isto é, sinais de entrada devem estar na faixa  $GND - 1,5\text{ V} \leq V_{in} \leq V_{DD} + 1,5\text{ V}$  (de 3,5 V a -1,5 V), ou de  $V_{TT} - 1,5\text{ V} \leq V_{in} \leq GND + 1,5\text{ V}$  (de -3,5 V a 1,5 V). Pinos de entrada não utilizados devem ser conectados à menor tensão de alimentação do CI (GND, no caso de alimentação positiva, ou  $V_{TT}$ , no caso de alimentação negativa).

A funcionalidade DC do circuito *7490-like* teve a prioridade mais alta de projeto, procurando-se garantir, em primeiro lugar, ótimas NMs, dentro dos extremos de variação dos parâmetros com o processo e a temperatura. O circuito deve ser, acima de tudo, funcional; caso contrário, para nada servirá a sua máxima velocidade de operação; esta vem a seguir na ordem de prioridades deste projeto. A minimização da área do chip ficou em terceiro lugar de prioridade, tendo relação direta com o custo de prototipagem do CI. A dissipação de potência recebeu menor prioridade, dado que a escala de integração do circuito *7490-like* é pequena.

O ciclo de projeto é iterativo, por isso deve-se procurar reduzir o número de iterações e nunca redimensionar portas lógicas cegamente, sem ter um objetivo bem definido *a priori*. Na etapa das simulações em computador, um projetista de CIs não pode perder tempo: deve ir direto ao problema e simular o CI, desde o início do projeto, já considerando as condições externas aproximadas em que a versão final do CI vai precisar operar. Por exemplo, desde o início das simulações, logo após a validação do conceito, devem ser incluídas as cargas  $C_L$  e  $R_L$  finais nas saídas do CI, bem como as variações de temperatura e de processo. Enfim, as condições externas de operação do CI devem estar sempre presentes, e o circuito é que terá que ser adaptado para elas.

Em uma linha de projeto *bottom-up*, o projetista deve primeiro simular e depurar, separadamente, cada bloco componente (célula) do circuito, começando pelos mais simples, e

somente depois passar à simulação e depuração dos blocos de nível mais alto (que são compostos por aqueles blocos mais simples), até chegar ao projeto do circuito final, que é composto por todos esses blocos operando juntos [79]. Seguindo-se essa metodologia, o número de simulações do CI 7490-like pôde ser minimizado, economizando-se bastante tempo de simulações em computador.

As alternativas de encapsulamento oferecidas pela Vitesse estão especificadas no manual [12], e a própria foundry recomenda o uso dos encapsulamentos LCC e LDCC, por terem menores indutâncias de pinos de alimentação. Considerando-se o reduzido número de pinos de I/O do chip 7490-like (figura 4.6), o LDCC de 28 pinos é mais do que suficiente para ele. As folhas de dados para cada tipo de encapsulamento estão no Capítulo 6 de [12], onde observa-se que o LDCC de 28 pinos possui uma largura de banda de 3,9 GHz, o que é suficiente para permitir que os sinais tenham tempos de subida e descida de 100 ps [11]. Portanto, esses serão os tempos de transição  $t_r$  e  $t_f$  do sinal de *clock* das simulações.

4.3.2. Projeto do FF J’K’ para Atender às Especificações do 7490-like.

Em princípio, as saídas dos FFs do 7490-like deverão acionar as capacitâncias de 5 pF externas ao chip, portanto elas devem ser dotadas de *buffers* de saída SDCFL bastante largos. Porém, as saídas Q e  $\bar{Q}$  também precisam acionar outras portas lógicas dentro do chip (FO’s internos), e esses *buffers* de saída, pelo fato de serem muito largos, são fortes demais para acionarem entradas de portas DCFL. Assim, as portas NOR das saídas Q e  $\bar{Q}$  devem ser conectadas a pré-*buffers* com dimensões apropriadas para acionarem cargas médias, e um desses pré-*buffers* é que vai realmente acionar o inversor SDCFL de saída do chip. Essa topologia está esquematizada na figura 4.8.

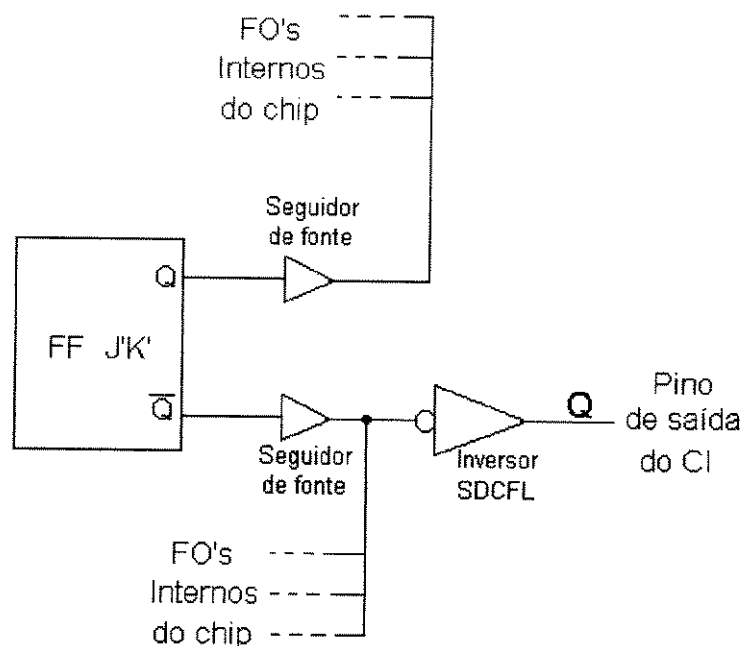


Figura 4.8 – Circuito de saída de cada flip-flop do CI 7490-like.

Um inversor SDCFL deve ser dimensionado para acionar a carga externa de 5 pF; primeiro determina-se a largura do DFET do seguidor de fonte, e então o EFET é dimensionado para  $\beta = 3$ . Nas saídas do CI, deseja-se uma excursão do sinal melhor do que 1 V (1,1 V dá uma boa margem de segurança, como será visto adiante), com  $C_L = 5$  pF sendo descarregada em um intervalo de tempo de 100 ps. Usando-se a equação (3.5), calcula-se a corrente  $I$  necessária para descarregar essa  $C_L$ , em um dado  $\Delta t$  ( $t_{HL}$ ) e com um certo  $\Delta V$ :

$$I = 5 \text{ pF} \frac{1,1 \text{ V}}{100 \text{ ps}} = 55 \text{ mA} \quad (4.2)$$

A Vitesse fornece uma tabela para o dimensionamento dos DFETs do processo HGaAs-III [12] na qual, para um valor de corrente  $I_{D,D}$  desejado, é fornecida a razão  $W_D/L_D$  necessária. Essa tabela abrange até correntes de 1,2 mA a 70 °C, mas pode-se extrapolar as razões  $W_D/L_D$  para correntes maiores, porque a relação entre elas é quase linear (Capítulo 3). A tabela indica  $W_D/L_D = 20/2,4$  para  $I = 1,2$  mA; para  $I = 55$  mA e  $L_D = 2 \text{ }\mu\text{m}$ :

$$\begin{array}{l} 20/2,4 \quad \text{_____} \quad 1,2 \text{ mA} \\ W_D/2 \quad \text{_____} \quad 55 \text{ mA} \end{array} \quad \rightarrow \quad W_D = 763 \text{ }\mu\text{m}, \text{ que aproxima-se para } 800 \text{ }\mu\text{m}.$$

Como  $\beta = 3$  para o buffer SDCFL,  $W_E = 1200 \text{ }\mu\text{m}$ . A parte inversora DCFL foi dimensionada com larguras compatíveis, através de séries de simulações, de modo a minimizar o atraso global:  $W_{Dinv} = 500 \text{ }\mu\text{m}$  e  $W_{Einv} = 1250 \text{ }\mu\text{m}$ . Dimensiona-se assim os inversores SDCFL para as saídas do CI (subcircuito **buffer2** no Programa 24 do Apêndice 1). As dimensões finais, em  $\mu\text{m}$ , são as seguintes:

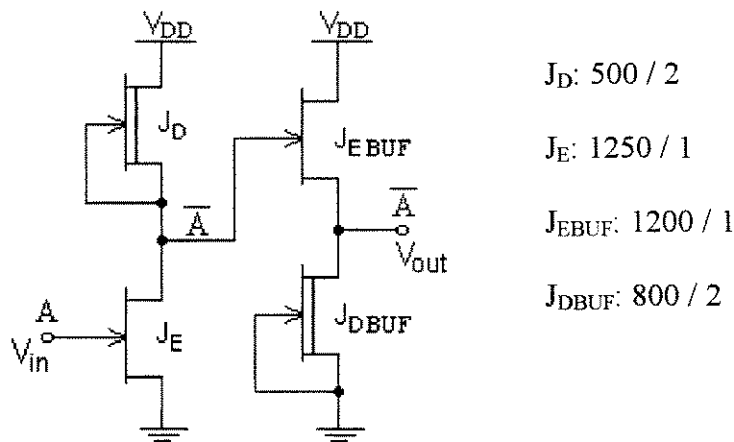


Figura 4.9 – Inversor SDCFL de saída do CI. As dimensões estão em  $\mu\text{m}$ .

A tensão numa carga  $C_L$  de 5 pF apresentou uma excelente excursão; no processo típico a 25 °C e com  $V_{DD} = 2 \text{ V}$ , o  $V_{OL}$  é muito baixo (12 mV),  $V_{OH} = 1,21 \text{ V}$ , e o  $t_{PD}$  atingido foi de 100 ps. O Programa 24 realiza a simulação transiente do inversor SDCFL da figura 4.9 para uma frequência de 500 MHz (a máxima frequência de saída do 7490-like), e as formas de onda de saída na carga  $C_L$  são mostradas na figura 4.10. O  $V_{OL}$  de pior caso, para o processo *fast*, é igual a 34 mV. No processo *slow*, o  $t_{PD}$  de pior caso foi igual a 122 ps, excelente para este projeto.

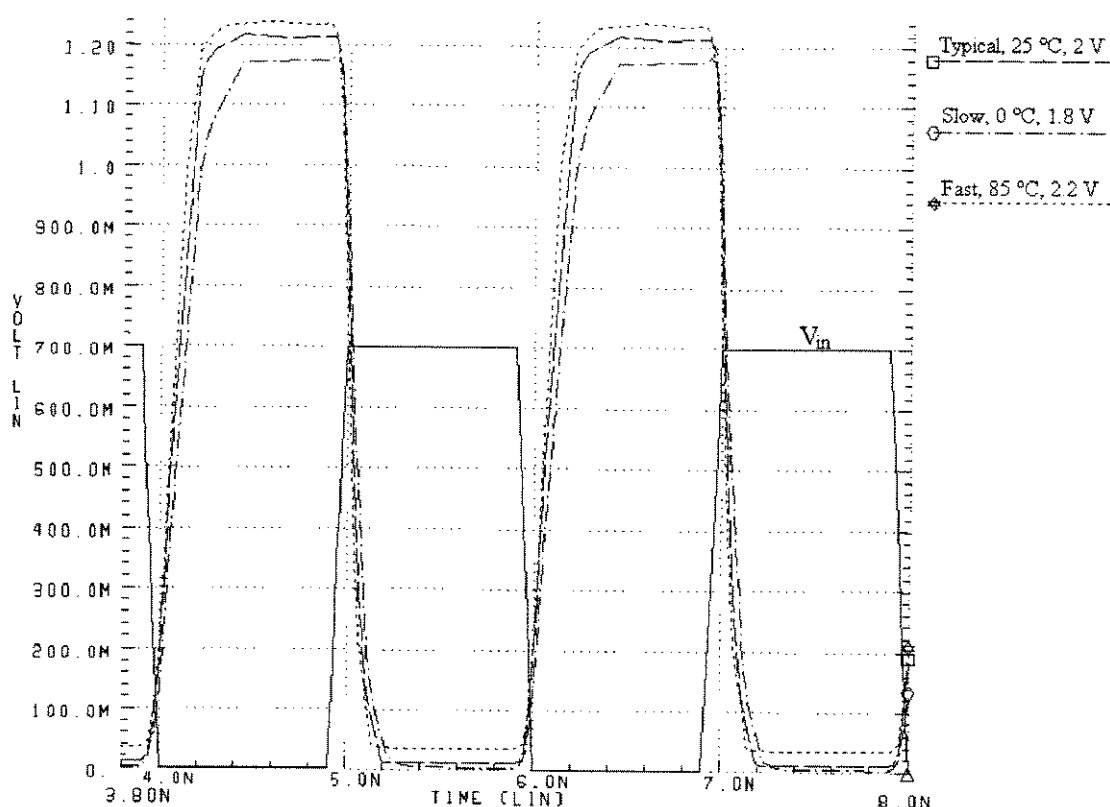


Figura 4.10 – Simulação do buffer de saída do CI 7490-like para  $C_L = 5$  pF.

Passa-se agora ao projeto de um FF J'K' geral, adequado para ser usado como célula para este CI. O dimensionamento das portas lógicas desse FF será executado de acordo com as recomendações do manual de projeto da Vitesse [12]. Por razões já explicadas, um  $\beta = 10$  será adotado em todas as portas, e as capacitâncias parasitas serão incluídas em todas elas. Essas capacitâncias são  **muito** importantes para a validade dos resultados das simulações e para estimativas realistas dos atrasos de propagação. As capacitâncias não-lineares dos MESFETs são calculadas automaticamente pelo HSPICE [11].

A equação (3.5) pode ser reescrita na forma:

$$\Delta t = \frac{C_L \cdot \Delta V}{I} \quad (4.3)$$

Em um circuito com uma série de portas em cascata, como é o caso de um FF,  $C_L$  da equação (4.3) é diretamente proporcional à largura  $W_E$  do transistor de entrada da porta seguinte. Mas a corrente  $I$  é diretamente proporcional à largura do transistor através do qual  $C_L$  é carregada ou descarregada.

Para acionar uma grande  $C_L$ , o FET tem que ser muito largo, e é possível que a largura necessária seja tão grande que sua fabricação seja impraticável. A  $C_L$  das saídas do CI é a maior capacitância de carga no projeto do chip (5 pF neste projeto), e para acioná-la é necessário um inversor bufferizado SDCFL, com transistores bem largos. Esse inversor, por sua vez, pelo fato de ser muito largo, representa também uma  $C_L$  alta para a porta que vai vir atrás dele (embora seja uma capacitância já menor do que 5 pF). Mas essa porta ainda poderá ter que ser bufferizada. E assim sucessivamente, vai ocorrendo um *downscaling* de capacitâncias e de larguras de FETs, das saídas em direção às entradas. Daí resulta que as

larguras dos FETs sejam dimensionadas progressivamente menores ao longo da cascata de portas, formando uma série geométrica na direção da saída para a entrada ([12], [66], [84]). A Vitesse [12] recomenda uma razão de escalonamento igual a 2 para as portas lógicas de um FF DCFL; isso significa que cada DFET deve ter a metade da largura do DFET da porta seguinte, como no projeto de exemplo em [12]. A referência [66] apresenta uma dedução geral para o projeto de circuitos por esta regra de dimensionamento, chamados tecnicamente de “*tapered designs*” (projetos com escalonamento progressivo) [84].

O dimensionamento dos DFETs de um circuito com séries de portas em cascata, como um FF, deve sempre começar pelas portas das saídas  $Q$  e  $\bar{Q}$  e prosseguir em direção às portas das suas entradas, porque, no início, a única  $C_L$  conhecida é a das saídas do CI (5 pF), e o dimensionamento ótimo de uma porta lógica ou *buffer* é função da  $C_L$  presente na sua saída. Uma vez determinadas as larguras dos DFETs, as  $W_E$ 's da porta em questão são calculadas impondo-se  $\beta = 10$ . Um FF MS contém dois *latches*, sendo que o *latch* escravo deve ser feito maior do que o *latch* mestre (para maior amplificação do sinal de entrada). Nunca se deve dimensionar o *latch* mestre igual ao escravo, isto é, nunca projetar um FF MS com dois *latches* iguais em cascata. O desempenho piora em tudo ( $V_{pp}$ ,  $t_{PD}$ ,  $t_{LH}$ ,  $t_{HL}$ ,  $P_{D,st}$ ), e a área do FF fica consideravelmente maior, em comparação com um FF MS projetado com escalonamento progressivo das portas na direção das entradas.

O manual [12] descreve o projeto de um FF para operação em 100 MHz. Como desejamos um FF para operação em 1 GHz, então de acordo com [4] pode-se esperar que, decuplicando as larguras dos DFETs e mantendo o mesmo  $\beta$ , seja possível alcançar uma frequência 10 vezes maior de operação.

Os projetos de CIs descritos na referência [4] são um exemplo de escalamento de portas lógicas proporcional à velocidade desejada de operação: se um FF deve operar a  $k$  vezes a velocidade de outro FF, ele é feito com  $k$  vezes as dimensões deste; multiplica-se as razões  $W_D/L_D$  e  $W_E/L_E$  das suas portas lógicas por um mesmo número inteiro  $k$ . Por exemplo, a partir de um FF “unitário”, para fazer-se outro que opere com o dobro da velocidade, este é feito com transistores com o dobro do tamanho (mas dissipando o dobro da potência). Resumindo, o fator de escalamento dos dispositivos é igual ao fator de escalamento das frequências (velocidades) de operação. Este escalonamento linear das portas do circuito é o método mais simples de *resizing* para um circuito em cadeia escalonada, como um FF. O resultado das simulações é que todos os tempos de atraso ( $t_{PHL}$ ,  $t_{PLH}$  e  $t_{PD}$ ) e de transição ( $t_{HL}$ ,  $t_{LH}$ ) diminuem, e diminuem tanto mais quanto maior o valor desse escalonamento.

Entretanto, esse método só é compensador até um certo valor de  $k$ , que depende da topologia do circuito e da frequência de operação que se pretende atingir. Chega um valor de  $k$  a partir do qual esse escalonamento uniforme deixa de ser interessante, porque passa a ocasionar o aumento do atraso  $t_{PD}$  do circuito em relação ao atraso obtido para o valor anterior de  $k$ , e já não compensa mais diante do aumento de área e consumo de potência do circuito. Nessa situação, se for necessária uma melhoria de desempenho (um aumento de velocidade), ela deve ser conseguida por métodos mais elaborados: pode-se redimensionar o *buffer* ou somente certas portas estratégicas (o caminho crítico). O re-dimensionamento de um FF para a otimização da velocidade deve novamente começar nas suas saídas e progredir em direção às entradas.

Portanto, o FF J’K’ final para integrar o contador 7490-like mantém as proporções do FF da Vitesse, mas com as dimensões decuplicadas ( $k = 10$ ); este escalonamento não chega a atingir

ainda a saturação e a inversão do comportamento de  $t_{PD}$ . Do FF inicial para este FF<sub>x10</sub> houve um ganho apreciável de velocidade da resposta. As respostas transientes dos FFs inicial e final, operando no modo J’K’, são comparadas na figura 4.11, onde Q1 é a saída do FF do início do projeto, e Q2 é a saída do FF final. As figuras 4.11 e 4.12 mostram simulações transientes do FF J’K’ dimensionado para este projeto, com *clock* de 1 GHz; a figura 4.12 apresenta a sua operação no modo *toggle*. Os valores dos parâmetros de desempenho relevantes são mostrados na tabela 4.7.

As dimensões finais das portas, em  $\mu\text{m}$ , são as seguintes:

|       | DFET     | EFET    |
|-------|----------|---------|
| Mast1 | 10,4 / 2 | 52 / 1  |
| Mast2 | 20,8 / 2 | 104 / 1 |
| Sla1  | 41,6 / 2 | 208 / 1 |
| Sla2* | 83,2 / 2 | 208 / 1 |

\* $\beta = 5$  porque vai acionar buffer

Tabela 4.6 – Dimensões finais das portas lógicas do FF J’K’ para compor o 7490-like.

A simulação deste FF e as medidas dos seus parâmetros de desempenho são feitas por meio do Programa 25. Este FF ainda não é “bufferizado” nas saídas; depois que os pré-*buffers* forem incluídos em suas saídas, o seu desempenho ainda deverá melhorar.

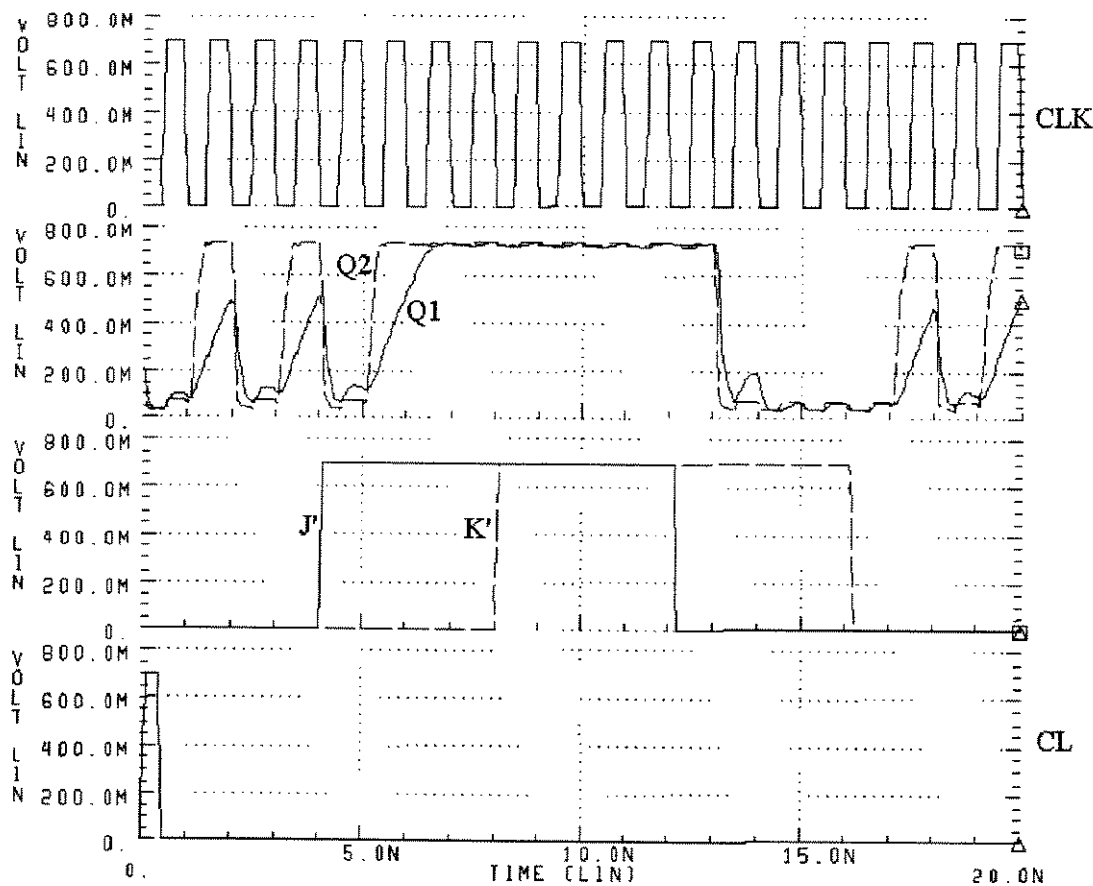


Figura 4.11 – Simulação do FF J’K’ final (Q2), comparado com o FF J’K’ do início do projeto (Q1).



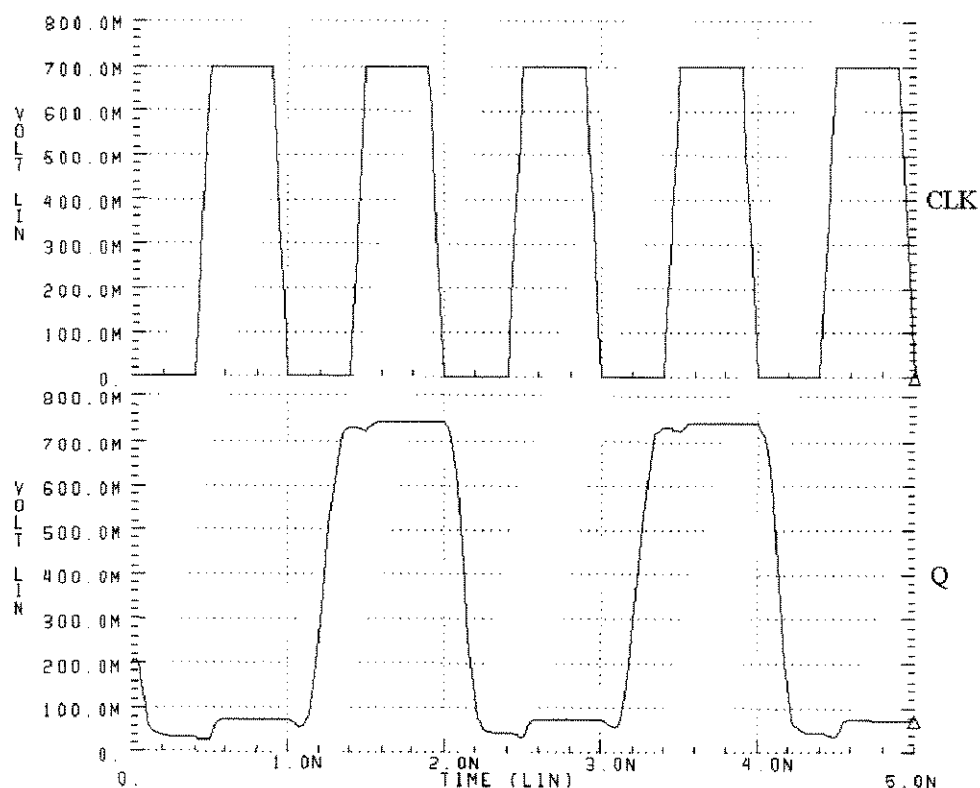


Figura 4.12 – Detalhe da simulação do FF J’K’ no modo *toggle*.

|                 | Mínimo | Típico | Máximo |
|-----------------|--------|--------|--------|
| $V_{OL}$ (mV)   | 55,8   | 61,6   | 62,6   |
| $V_{OH}$ (mV)   | 676,3  | 734,2  | 747    |
| $V_{pp}$ (mV)   | 613,6  | 672,6  | 691,2  |
| $P_{D,st}$ (mW) | 39,8   | 55,5   | 75,5   |
| $t_{PD}$ (ps)   | 219    | 229    | 281,8  |

Tabela 4.7 – Parâmetros de desempenho mais relevantes para o FF J’K’.

O atraso  $t_{PD}$  é medido na saída Q. Devido à simetria de construção do FF, é certo considerar que o atraso para a saída  $\bar{Q}$  tenha o mesmo valor.  $P_{D,st}$  é medida com as entradas J, K e CLK, e a saída Q, em nível alto. Os valores para a tabela 4.7 são escritos no programa **FF.lis**, gerado pelo HSPICE na simulação de **FF.sp**.

#### 4.3.3. Descrição do 7490-like no Simulador.

As mesmas células de portas lógicas utilizadas no primeiro FF foram utilizadas também para a descrição dos demais FFs. Levou-se em consideração as particularidades de cada um dos quatro FFs J’K’ do CI 7490-like, fazendo-se todas as simplificações permitidas pela lógica para minimizar a quantidade de EFETs no subcircuito de cada FF, e portanto minimizar a área e o consumo de potência do circuito total. A partir do esquema lógico do 7490-like da figura 4.6, pôde-se efetuar as seguintes simplificações em cada FF J’K’:

- O FF1 não tem as entradas J’ e K’, logo ele foi feito sem os EFETs para essas duas entradas, o que, logicamente, equivale a deixá-las sempre conectadas ao nível lógico baixo;

- O FF2 não tem a entrada  $K'$  nem as entradas de *Clear* (CL), o que resulta na eliminação de quatro transistores;
- O FF3 não possui as entradas  $J'$  e  $K'$ , e nem as entradas de *Preset* (PR), eliminando-se mais cinco transistores do seu subcircuito;
- O FF4 é um  $S'R'$ , portanto ele não tem a realimentação da saída Q para a porta de entrada, eliminando-se um FET do mesmo; a conexão da saída  $\bar{Q}$  com a entrada  $S'$  (figura 4.6) foi feita internamente ao seu subcircuito.

Os esquemáticos lógicos detalhados de cada FF do *7490-like* são mostrados no Capítulo 5, figuras 5.15 a 5.18.

Como os drivers de saída são, na verdade, inversores SDCFL, a topologia da figura 4.6 teve que ser readaptada para preservar-se a sua função lógica, resultando o circuito da figura 4.13. Comparando-se as figuras 4.13 e 4.6, percebe-se uma outra simplificação na topologia do CI: a porta NAND que acionava a entrada  $R'$  do FF4 foi substituída por uma OR, eliminando-se dois inversores que seriam necessários para a conversão NOR-NAND.

A *netlist* do circuito *7490-like* foi escrita a partir da interligação dos quatro subcircuitos de flip-flops, mais as portas lógicas auxiliares, de acordo com a figura 4.13. No nível do circuito principal, os inversores/drivers SDCFL foram conectados às saídas corretas de cada FF, para acionarem as cargas capacitivas externas ao chip.

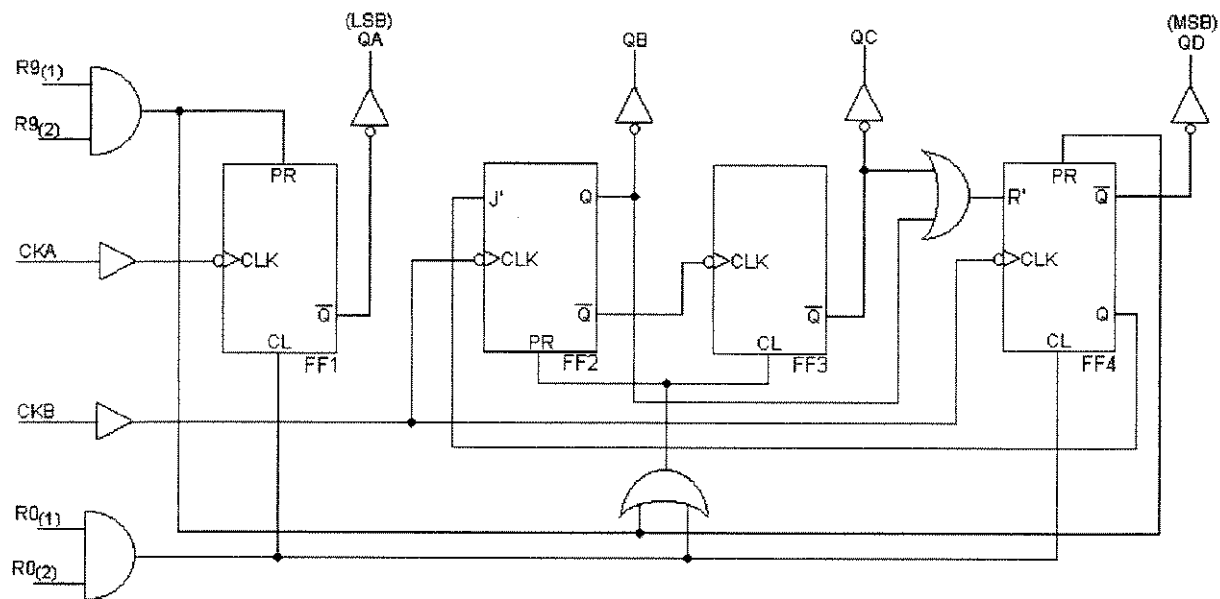


Figura 4.13 – Esquema lógico final do divisor de frequências *7490-like*.

Os seguidores de fonte (pré-buffers) nas saídas dos FFs, para o acionamento dos seus *fan-outs* internos ao CI, foram dimensionados em função das cargas específicas em cada saída de FF, para obter-se as melhores excursões de sinal e atrasos em cada caso. Assim, surgiram três tamanhos diferentes de pré-buffers, cujos subcircuitos foram denominados de **buffer**, **buffer3** e **buffer4** (**buffer2** é o inversor SDCFL de saída). Para a simetria das saídas Q e  $\bar{Q}$  dos FFs (uma delas é somente interna ao CI e a outra aciona o **buffer2**), os pré-buffers foram colocados em ambas, nos quatro FFs.  $V_{OH}$  e as correntes de saída não podem ser muito altos para os pré-buffers, daí a impossibilidade do uso de inversores SDCFL iguais aos das saídas. Se os pré-buffers forem fortes demais, trarão problemas de velocidade aos FFs.

Para efeito das simulações do circuito *7490-like*, os pré-buffers foram incorporados à estrutura interna dos flip-flops (dentro dos subcircuitos). Os pré-buffers usados em cada FF podem ser vistos nas figuras 5.15 a 5.18. Para o dimensionamento desses seguidores de fonte, leva-se em consideração o fato de que uma  $C_L$  é carregada ou descarregada através das  $R_{out}$ 's dos FETs, daí a necessidade das  $R_{out}$ 's serem as menores possíveis, o que é conseguido ao mesmo tempo em que se aumenta a capacidade de corrente dos FETs, ou seja, com um aumento das suas larguras. O dimensionamento final para os pré-buffers foi alcançado através de numerosas simulações em computador.

Seguidores de fonte também foram incluídos nas saídas das portas AND e OR de *preset* e *clear* (figura 4.13), que foram dimensionadas com  $\beta = 5$ , para acionarem os altos fan-outs DCFL destas portas. As portas AND de *Preset* e *Clear*, que são idênticas, foram implementadas como mostrado na figura 4.14. Ambas precisam acionar um FO = 7 e longas linhas de interconexão, por isso elas são formadas por portas NOR com FETs mais largos (subcircuito **slaPRCL**), e dotadas de seguidores de fonte (subcircuito **buffer**) em suas saídas. Já a porta OR que aciona as entradas PR do FF2 e CL do FF3, tem como carga um FO = 6 mais a capacitância total das interconexões, e foi usado um inversor SDCFL na sua saída (subcircuito **clkbuffer2**). As definições de todos os subcircuitos utilizados neste projeto, nos níveis de esquemático e de layout, são dadas no Apêndice 4.

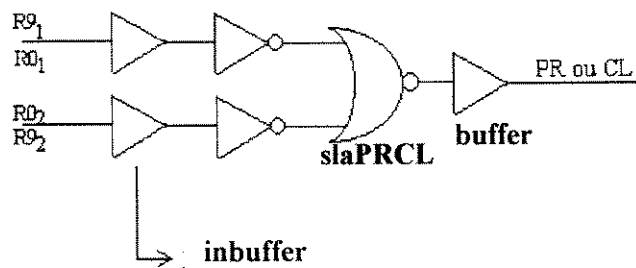


Figura 4.14 - Portas AND usadas no *7490-like*.

Os pinos de entrada do CI têm que ser terminados internamente (*on-chip*) por resistores, para evitar-se reflexões do sinal nos mesmos e para obter-se formas de onda de entrada limpas. Portanto, o subcircuito de regulação de corrente da figura 3.68 (subcircuito **inbuffer**) foi ligado nas entradas de sinal (clock, preset e clear) do CI. A célula **inbuffer** possui um resistor de terminação (*pull-down*) para GND (ou VTT) que, em paralelo com a resistência equivalente do DFET e do EFET que vêm a seguir, deve prover uma resistência total de entrada de aproximadamente 50  $\Omega$ , para o casamento da resistência de entrada do CI. O valor apropriado do resistor de terminação foi determinado em 250  $\Omega$ , por meio de simulações no HSPICE.

O DFET regulador de corrente precisa ter uma largura total muito grande (Capítulo 3), sendo descrito em **inbuffer**, tanto na netlist como no layout, como vários DFETs menores ("dedos de gate") em paralelo (ver a figura 5.7 e a célula **inbuffer** no Apêndice 4). A corrente gerada é alta, e é toda entregue ao terminal de porta do EFET do estágio que vem a seguir (o subcircuito **clkbuffer**). Se fosse um terminal de porta único, a corrente nele seria excessiva e iria rompê-lo; por isso, o EFET de entrada de **clkbuffer** também teve que ser desenhado com o paralelismo de vários transistores (*gates*), para a corrente ser dividida entre cada um. Este EFET foi desenhado com um número de gates suficiente para operar em uma faixa de 1,5 V

em torno da excursão nominal da entrada, isto é, aproximadamente de 3,5 V a -1,5 V. Isso significa que, se  $V_{in}$  estiver dentro dessa faixa, a corrente de entrada não lhe causará dano.

Entradas não utilizadas devem ser conectadas à menor tensão de alimentação do CI (GND, no caso de alimentação positiva, ou  $V_{TT}$ , no caso de alimentação negativa).

A figura 4.15 mostra o esquemático lógico detalhado do circuito *7490-like*, em sua versão final que foi enviada ao CMP para fabricação; no total, a versão definitiva utiliza **191** MESFETs e 64 portas lógicas equivalentes. Neste esquemático são indicados os números dos subcircuitos (na forma X<n>) e dos nós, usados na netlist de simulação. Este esquemático deve estar sempre acessível para a edição da netlist do HSPICE, durante sessões de simulação, e durante o desenho do layout, pois a figura 4.15 é a perspectiva mais conveniente do circuito para todas essas finalidades.

A *netlist* para a simulação do CI *7490-like*, na configuração de contador de década, é o Programa 26 do Apêndice 1. Para simular o CI nas demais configurações de divisão de frequências, basta mudar a conectividade dos subcircuitos na seção rotulada “Circuito Principal” do Programa 26. Os resultados das simulações das configurações de divisão por N são mostrados no Apêndice 3. Os Programas 27 e 28 contêm as linhas que devem ser substituídas no Programa 26 para a simulação do contador de década nos processos *slow* e *fast*.

As tabelas 4.8 e 4.9 fornecem os parâmetros de desempenho DC e AC simulados para o circuito *7490-like*. Seus valores de excursão do sinal de saída mantiveram-se sempre superiores a 1 V.

| Parâmetros DC | Mínimo | Típico | Máximo |
|---------------|--------|--------|--------|
| Potência (W)  | 0,763  | 1,18   | 1,4    |
| $V_{IH}$ (V)  | -1,2   | -1     | 0      |
| $V_{IL}$ (V)  | -1,99  | -1,8   | -1,8   |
| $V_{OH}$ (V)  | -0,93  | -0,8   | -0,76  |
| $V_{OL}$ (V)  | -1,99  | -1,98  | -1,91  |
| $I_{IH}$ (mA) | 32     | 35     | 46     |
| $I_{IL}$ (mA) | 2,8    | 4      | 4,6    |
| $I_{OH}$ (mA) | 23,7   | 24     | 24     |
| $I_{OL}$ (mA) | 0,2    | 0,2    | 0,7    |

Tabela 4.8 – Especificações de desempenho DC do CI *7490-like*.

As condições de simulação foram as seguintes:  $V_{TT} = -2V \pm 10\%$ , sinal de *clock* com  $V_H = -0,8V$  e  $V_L = -2V$ , e uma carga de  $R_L = 50\Omega$  e  $C_L = 5$  pF. A potência total do CI foi avaliada com o comando **.op** do HSPICE; seu valor foi medido em um instante em que o clock e três saídas estavam altas. O comando **.op** imprime as correntes nos terminais de cada MESFET do circuito, no arquivo de saída com extensão **.lis**. Para o desenho do *layout* do chip, é necessário utilizar todos os valores de correntes dos FETs contidos no arquivo “**deccnt.lis**”, para o correto dimensionamento das trilhas de interconexão e de alimentação do chip.

As simulações do *7490-like* como contador de década e como divisor por 10 resultaram nas formas de onda corretas, mostradas nas figuras 4.16 e 4.17.

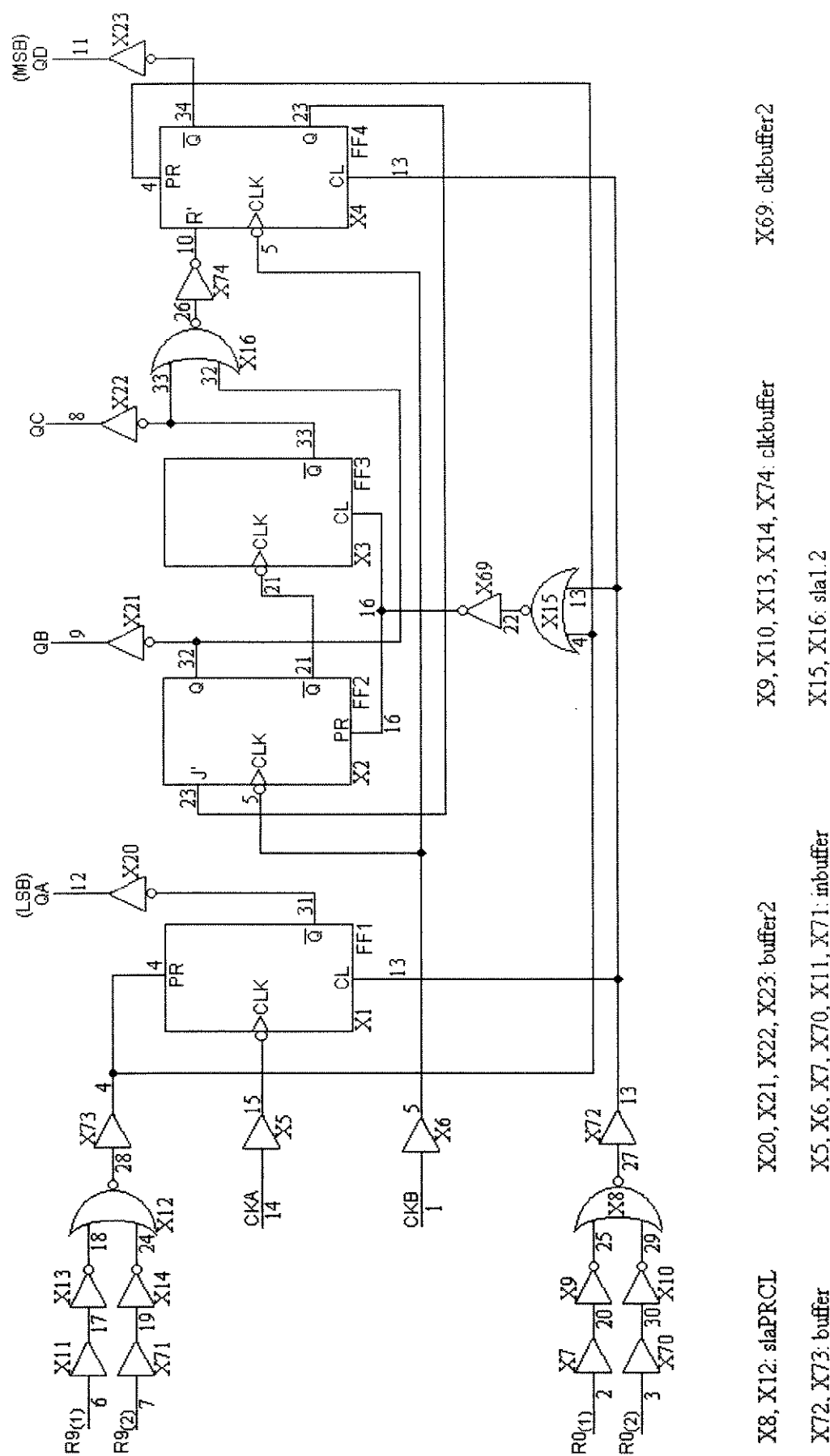


Figura 4.15 - Esquemático lógico final, detalhado, do CI 7490-like.

| Parâmetros AC                         | Mínimo | Típico | Máximo |
|---------------------------------------|--------|--------|--------|
| Atraso de CKA para QA (ps)            | 406,7  | 423    | 493,4  |
| Atraso de QA para QB (ps)             | 430,3  | 434,6  | 480,1  |
| Atraso de CKA para QC (ps)            | 748,6  | 1049,6 | 1078   |
| Atraso de CKA para QD (ps)            | 899,3  | 921,7  | 1070,4 |
| Atraso de CKB para QB (ps)            | 225,7  | 259,8  | 293    |
| Atraso nas portas AND de PR e CL (ps) | 95,3   | 113,7  | 146,5  |

Tabela 4.9 - Especificações de desempenho AC do CI 7490-like, operando como contador de década.

Os dois últimos atrasos da tabela 4.9 foram medidos pelo Programa 29. Os atrasos de QA para QB são muito maiores do que os atrasos de CKB para QB, porque QA tem uma carga muito maior para acionar ( $C_L = 5\text{ pF}$ ). O atraso médio das portas lógicas dos FFs é de 85 ps.



Figura 4.16 - Simulação final do 7490-like como contador de década.

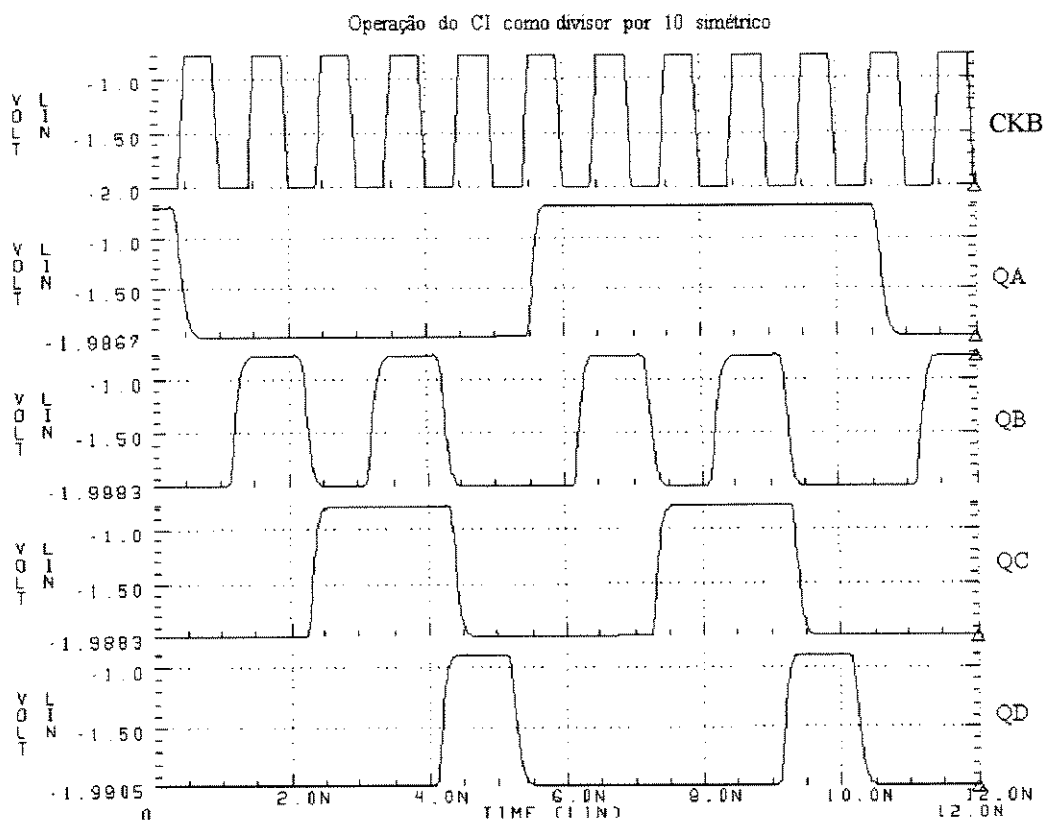


Figura 4.17 - Formas de onda para a operação do CI como um divisor por 10 simétrico.

#### 4.4. Conclusão.

Na última rodada de simulações antes de iniciar-se o desenho do *layout*, o circuito típico e nos piores casos de temperatura e de processo já tinha sido simulado com sucesso. A partir deste momento, a necessidade de otimizações adicionais somente poderá ser avaliada após estar disponível um melhor conhecimento dos atrasos de interconexão e componentes parasitas do *layout*. Além disso, espera-se receber o CI fabricado pela *foundry*, para que possam ser feitas medidas experimentais, e comparar-se os desempenhos simulado e medido.

## CAPÍTULO 5.

### *Layout do CI 7490-like.*

#### 5.1. Informações Gerais para o *Layout* de CIs.

O processo de confecção de um *layout* de CI consiste em transformar um esquemático de circuito em um arquivo de máscaras, que sejam compatíveis com o processo de fabricação de alguma *foundry*. No caso deste trabalho, os esquemáticos do circuito, feitos à mão, foram usados juntamente com uma *netlist* do mesmo circuito, para a tarefa de desenhar-se essas máscaras em um ambiente de software de CAD. O processo de fabricação para o qual as máscaras do CI 7490-like foram desenhadas foi o HGaAs-III (o 'H' significa "*High integration*"), o processo de arseneto de gálio da *foundry* Vitesse Semiconductor Corporation, com comprimento mínimo de porta igual a 0,6  $\mu\text{m}$ . A Vitesse é uma *foundry* norte-americana especializada na produção de CIs de GaAs.

Todo projeto de CI é feito visando-se a sua implementação física através de um processo de fabricação. Para que a implementação desse CI seja possível no processo escolhido, o seu *layout* deve ser feito em total concordância com um conjunto de regras de projeto, elaboradas pelas pessoas responsáveis pelo processo de fabricação. Há dois conjuntos de regras de projeto que o projetista do CI precisa obedecer durante o desenho do *layout*: as regras de projeto **geométricas** e as regras de projeto **elétricas**.

As regras de projeto geométricas referem-se às dimensões mínimas dos elementos (*features* ou *critical dimensions*, *CD*) que compõem o circuito. Consistem em uma lista de regras que especificam as larguras mínimas permitidas para as trilhas de interconexão, os espaçamentos mínimos permitidos entre elementos dos diversos tipos de camadas, e os *overlaps* necessários entre algumas camadas para a formação dos FETs e das vias.

As regras de projeto elétricas consistem dos valores nominais para os parâmetros elétricos de cada tipo de camada do processo, como as resistências de folha das camadas, as resistências dos contatos, as densidades de corrente máximas para cada tipo de camada e as capacitâncias entre as camadas e destas para o substrato.

Existem ainda as regras de projeto litográficas, que incluem o posicionamento das *dies* na *wafer* e a configuração dos retículos, mas o projetista não precisa se preocupar com estas regras porque elas se referem a etapas do processo de fabricação que são de responsabilidade exclusiva da *foundry*.

A *foundry* precisa fornecer as regras de projeto para cada um de seus processos de fabricação de CIs, e os projetistas devem observar todas essas regras durante a etapa de desenho do *layout*. Violações propositalmente às vezes são permitidas, mas somente por meio de autorização especial da *foundry*.

As regras de projeto geométricas, elétricas e litográficas para o processo HGaAs-III estão listadas no capítulo 4 da referência [12], juntamente com as ilustrações explicativas. Recomenda-se que as dimensões e espaçamentos mínimos somente sejam usados quando resultarem em melhorias significativas de desempenho e/ou densidade (economia de área)



[12]. Quanto mais “agressivo” o *layout*, maior o potencial de resultar um circuito de alto desempenho. Porém, com um *layout* “conservativo”, os circuitos resultantes têm maior probabilidade de funcionar, embora com menor nível de desempenho. Portanto, melhor desempenho geralmente é conseguido às custas do rendimento da fabricação. Neste projeto, as trilhas e vias do *chip* foram desenhadas com uma margem de segurança, acrescentada aos valores mínimos das regras de projeto, como recomendado em [11].

A *foundry* também fornece um *design kit* contendo as regras de projeto, para a verificação automática de *layouts*. O *design kit* geralmente consiste de vários arquivos de definições, modelos e esquemáticos de componentes, células padrão e regras de projeto, todos escritos para uma dada plataforma de CAD existente no mercado (por exemplo, Mentor Graphics, Cadence, HSPICE, Synopsys, etc.). No caso da Vitesse, um *design kit* completo para projetos *full-custom* está disponível para a plataforma Cadence, mas infelizmente não dispúnhamos da mesma durante este projeto. O *layout* do CI 7490-like foi todo desenhado com a utilização do software de CAD “*Magic*”, versão 6.3 para estações Sun, configurado para o processo HGaAs-III por meio do arquivo de tecnologia “*cmp.tech26*”, que foi fornecido pelo CMP. O *Magic* é a base sobre a qual já foram realizadas muitas pesquisas em GaAs, industriais e acadêmicas [11].

Os procedimentos de verificação de projeto visam gerar arquivos de *layout* livres de erros, a partir dos quais as máscaras possam ser preparadas. A verificação das regras de projeto geométricas e de formação dos componentes é feita por uma ferramenta de DRC. A verificação das regras de projeto elétricas é feita por uma outra rotina, o ERC. E a verificação da correspondência entre o *layout* e o esquemático inicial é feita por uma ferramenta chamada LVS. Para fazer o LVS, os softwares geram uma *netlist* em formato SPICE, a partir do esquemático do circuito, e comparam essa *netlist* com o circuito extraído do *layout*. Para este projeto, só estava disponível a ferramenta de DRC.

## 5.2. As Camadas do Processo da Vitesse e suas Características.

Os usuários do processo HGaAs-III (projetistas) deverão utilizar treze tipos de camadas para seus *layouts*, que são listadas a seguir com seus nomes em português e em inglês, e as abreviações adotadas:

- Camada 1 – Área Ativa, *active area*, AC;
- Camada 2 – Implante de Depleção, *depletion implant*, DI;
- Camada 3 – Metal de Porta, *gate metal*, GM;
- Camada 4 – Metal Ôhmico, *ohmic metal*, OM;
- Camada 5 – Via 1, que pode ser de dois subtipos: *via1sd* ou VSD, e *via1g* ou VG;
- Camada 6 – Metal 1, *metal 1*, M1;
- Camada 7 – Via 2, *via 2*, V2;
- Camada 8 – Metal 2, *metal 2*, M2;
- Camada 9 – Via 3, *via 3*, V3;
- Camada 10 – Metal 3, *metal 3*, M3;
- Camada 11 – Passivação (negativa), *glass*, OG;
- Camada 14 – Via 4, *via 4*, V4;
- Camada 15 – Metal 4, *metal 4*, M4.

A sequência de camadas é mostrada na figura 5.1 a seguir:

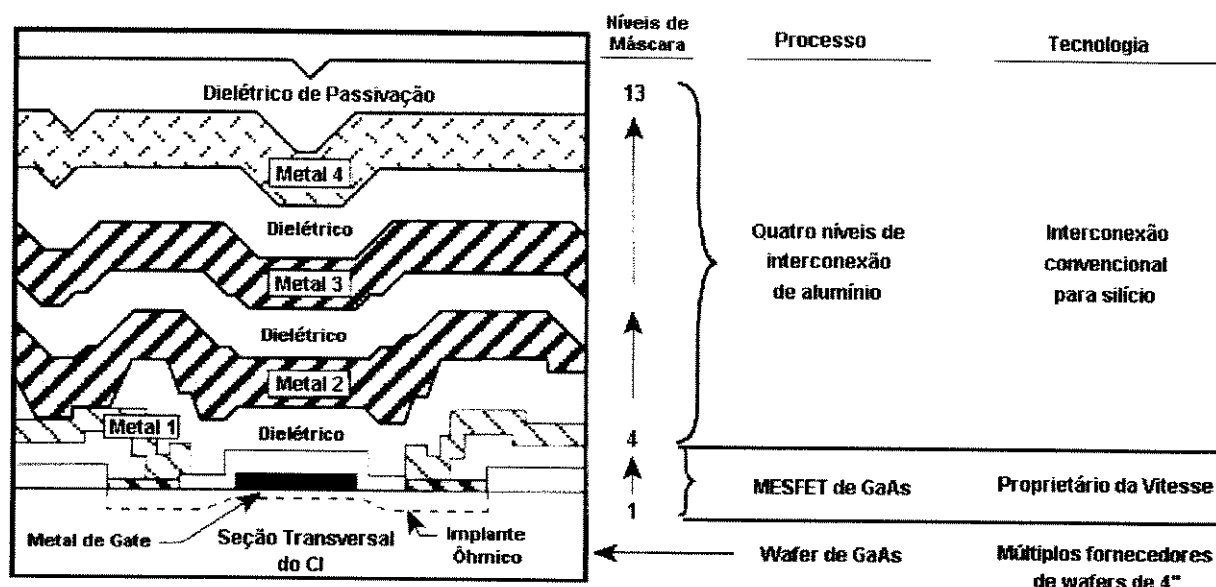


Figura 5.1 – Sequência de camadas do processo HGaAs-III da Vitesse.

O GaAs é um cristal anisotrópico, portanto as propriedades dos FETs de GaAs, assim como o desempenho de um CI, dependem fortemente das orientações dos canais dos FETs em relação ao substrato. Curvas características I-V de quatro FETs idênticos, porém fabricados ao longo de quatro direções diferentes na superfície do GaAs, exibem notáveis diferenças em transcondutância e  $V_t$  [6]. A natureza anisotrópica do cristal do GaAs tem um efeito direto sobre o desenho do *layout* dos CIs: os terminais de porta de todos os MESFETs devem ser orientados na direção horizontal. Portanto, no *layout* de um CI de GaAs não devem ser desenhados transistores na vertical. Contudo, essa regra não se aplica a diodos, que não são afetados pela anisotropia [11].

A camada de área ativa é usada para formar as regiões ativas dos dispositivos (fontes, drenos e canais). A figura 5.2 mostra o *layout* de um EFET na tecnologia HGaAs-III, com as dimensões mínimas indicadas. O *overlap* da porta sobre a área ativa deve ser o mínimo permitido ( $0,7 \mu\text{m}$ ) para minimizar a capacitância parasita associada.

Já um DFET é composto de área ativa, implante de depleção e metal de porta sobrepostos ( $\text{dfet} = \text{ac} + \text{di} + \text{gm}$ ). O implante de depleção determina quais MESFETs se tornarão DFETs. As etapas para compor um DFET são as seguintes:

- 1 – Criar uma área do tipo dfet sobre a área ativa do futuro MESFET (comando **paint dfet** do *Magic*);
- 2 – O dfet já contém área ativa; basta remover a área ativa em excesso das bordas.
- 3 – O que sobra é somente o metal de porta (entretanto, deveria sobrar metal de porta mais implante de depleção; isso foi um erro do arquivo de tecnologia usado);
- 4 – Pintar tudo com o implante de depleção.

A figura 5.3 mostra o *layout* de um DFET (fora de escala) na forma como ele é empregado em portas lógicas DCFL e seguidores de fonte, isto é, com a porta ligada na fonte. O próprio metal de porta é usado para essa conexão.



A largura da área ativa de um MESFET precisa ser maior ou igual a  $2\text{ }\mu\text{m}$  em todo o projeto ( $W_g \geq 2\text{ }\mu\text{m}$ ).

Na figura 5.2, a separação entre o terminal de porta e os terminais de fonte e dreno do MESFET deve ser **sempre a mínima**, igual a  $0,8\text{ }\mu\text{m}$  (pág. 4-7 do manual [12]). Essa distância é muito importante para o melhor desempenho dos MESFETs, porque ela determina diretamente os valores das resistências parasitas série  $R_S$  e  $R_D$  do FET; para minimizar-se  $R_S$  e  $R_D$ , os espaçamentos entre os três terminais devem ter o valor mínimo permitido pela tecnologia. O desenho dos MESFETs com essas dimensões mínimas torna-se ainda mais crucial pelo fato de que as resistividades do metal ôhmico e da área ativa são muito altas ([11], [12]). O DRC para *Magic* somente verifica se essa distância é menor do que a mínima; se for, ele fornece uma indicação de erro. O projetista, entretanto, também precisa verificar se essa separação não é grande demais (maior do que  $1\text{ }\mu\text{m}$ ); se for, ele deve corrigi-la para  $0,8\text{ }\mu\text{m}$  nos *layouts* dos MESFETs. Isso é válido para EFETs e DFETs; nestes, não precisa haver um espaçamento maior entre porta e dreno ou entre porta e fonte, por causa da presença do implante de depleção.

Na figura 5.3, a borda horizontal superior do implante de depleção pode até mesmo tangenciar a borda do metal ôhmico de dreno, sem problemas. O metal ôhmico serve para fazer os contatos ôhmicos dos terminais dos dispositivos e somente pode ser usado sobre área ativa, não servindo para interconexões.

Os contatos ôhmicos dos terminais de fonte e dreno não devem ser usados como metalização de interconexão. Eles são feitos para proverem uma conexão de baixa perda, estável e linear em  $I/V$  para o GaAs do MESFET que está por baixo. Os contatos ôhmicos são referidos no *Magic* como “via1sd”. O chamado “contato 0” é formado pela sobreposição entre metal de porta e metal ôhmico sobre área ativa.

As interconexões em um *chip* são realizadas por alguns (dois ou mais) níveis de metal de baixa resistividade e independentes entre si, sendo que o metal mais usado para esse fim é o alumínio. Materiais de maior resistividade, como o metal de porta (ou metal Schottky) e o metal ôhmico, obviamente causam uma maior queda de tensão do que o alumínio, quando percorridos pela mesma corrente, portanto somente são adequados para interconexões bem curtas e de baixa corrente.

O conceito de resistência de folha  $R_s$  (*sheet resistance*) para uma trilha metálica é explicado em [11], e os valores de  $R_s$  para as camadas do processo da Vitesse são dados na tabela 4-1 do manual [12], em  $\Omega/\text{quadrado}$ , para duas temperaturas. O *software* de desenho do *layout* recebe os valores de  $R_s$  no arquivo de tecnologia e os utiliza para “extrair” (estimar) as resistências parasitas das diversas camadas de interconexão, portanto o projetista geralmente não precisa calcular essas resistências para o seu CI. Mesmo assim, os valores de  $R_s$  da tabela 4-1 são úteis para a comparação entre camadas (e decisão de qual a melhor camada para usar em uma certa função) e para o projeto de resistores, conforme explicado adiante. A resistência total de uma trilha de alimentação pode ser usada para estimar-se a máxima queda de tensão ôhmica na mesma, sabendo-se a sua corrente máxima.

No *layout* de um *chip*, existem basicamente dois tipos de interconexões, quanto à função que desempenham: as de distribuição da tensão de alimentação e as de transmissão de sinal. Metal 1 e metal 2 devem ser usados somente para as linhas de sinal, enquanto o metal 3 é reservado preferencialmente para os barramentos de tensão de alimentação por ser uma

camada mais espessa, e portanto tem menor  $R_s$  e suporta maior corrente por unidade de largura. O metal 4 deve ser usado como um plano único de metal sobre todo o *chip*, para uma fonte de alimentação ou o terra. Neste trabalho, o plano de metal 4 foi usado como plano de terra (ou de  $V_{TT}$ , conforme a polaridade da alimentação). E interconexões com o metal de porta devem ser limitadas a distâncias muito pequenas e a baixas correntes, devido à alta  $R_s$  ( $\Omega/\text{quadrado}$ ) desse material.

As vias são estruturas de contato entre as camadas de metalização. Na prática, há cinco tipos de vias, como segue:

Via 1sd: conecta “metal 1” com “metal ôhmico” ( $sd = \text{source-drain}$ ).

Via 1g: conecta “metal 1” com “metal de porta” ( $g = \text{gate}$ ).

Via 2: conecta “metal 2” com “metal 1”.

Via 3: conecta “metal 3” com “metal 2”.

Via 4: conecta “metal 4” com “metal 3”.

Todas as vias são geradas automaticamente pelo *Magic*, porque são tratadas como macros (*design logs*); isto é, o projetista não precisa compor uma via manualmente, pintando cada uma das camadas que a compõem. O *design log* é tratado como uma camada única: basta pintar a via desejada, e o *Magic* pinta automaticamente as suas camadas componentes, nos tamanhos certos de acordo com as regras de projeto geométricas. Para o *Magic*, uma via já contém as suas camadas superior e inferior por definição. Não é permitido o empilhamento de vias, somente o seu tangenciamento.

Isso às vezes pode gerar alguma confusão na realização do DRC. Por exemplo, a via 2 deve ser um quadrado de pelo menos  $1,8\ \mu\text{m}$  de lado, envolvida por um quadrado de metal 1 cujos lados devem ficar a uma distância de  $0,5\ \mu\text{m}$  de cada lado da via. Esse arranjo é mostrado na figura 5.4.

Para o *Magic*, entretanto, a via 2 deve ter uma largura mínima de  $2,8\ \mu\text{m}$  ( $1,8 + 2 \times 0,5$ )  $\mu\text{m}$ , e a separação mínima entre “vias 2” é de  $1\ \mu\text{m}$ , ao invés dos dois  $2\ \mu\text{m}$  estipulados na regra de projeto 77SP1. Isso porque a via 2 compreende também o metal em sua volta.

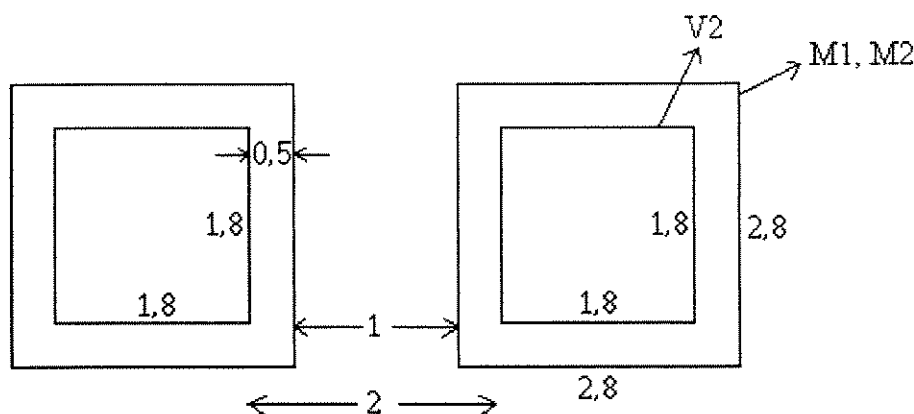


Figura 5.4 – Definição de via 2. Todas as dimensões estão em  $\mu\text{m}$ .

Pode ser desenhada uma via de qualquer tamanho maior que o mínimo, mas mesmo assim os *overlaps* entre as camadas superior e inferior e a via serão automaticamente desenhados com os valores mínimos, por conta do arquivo de tecnologia. A figura 5.5 mostra uma via 1 com dimensões de  $5 \times 10\ \mu\text{m}$ .

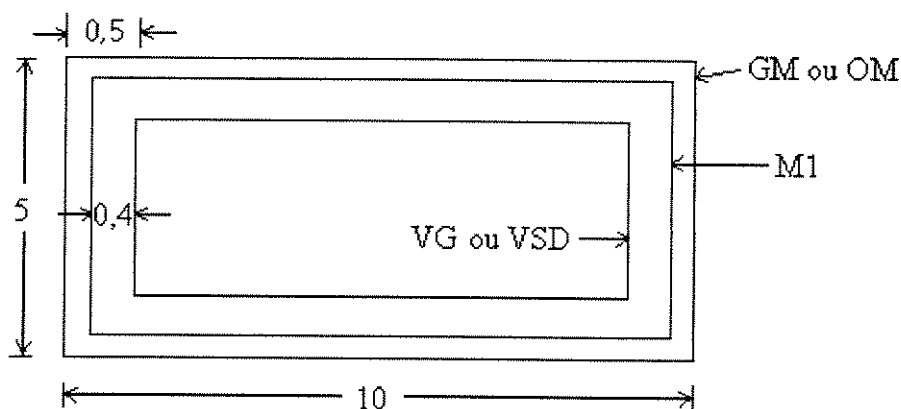


Figura 5.5 – Exemplo de uma via 1. Todas as dimensões estão em  $\mu\text{m}$ .

O mesmo raciocínio vale para as vias 3 e 4. Assim, qualquer via 3 desenhada pelo *Magic* estará interna a um retângulo de metal 2 e a outro de metal 3, com uma separação de  $0,6 \mu\text{m}$  nos quatro lados. E o tamanho mínimo para uma via 3 é de  $3,8 \mu\text{m}$  ( $2,6 + 2 \times 0,6$ )  $\mu\text{m}$ . Finalmente, o tamanho mínimo para se desenharmos uma via 4 é de  $6 \mu\text{m}$  ( $4 + 2 \times 1$ )  $\mu\text{m}$ , conforme a figura 5.6 abaixo, já que o metal 4 tem que ser usado como uma folha única cobrindo todo o circuito. Um detalhe importante é que a regra de separação entre a via 4 e a borda do metal 4 (regra 1415OL1 da página 4-21 de [12]) não é verificada pela seção de DRC do arquivo de tecnologia, portanto a obediência a essa regra deve ser assegurada pelo usuário.

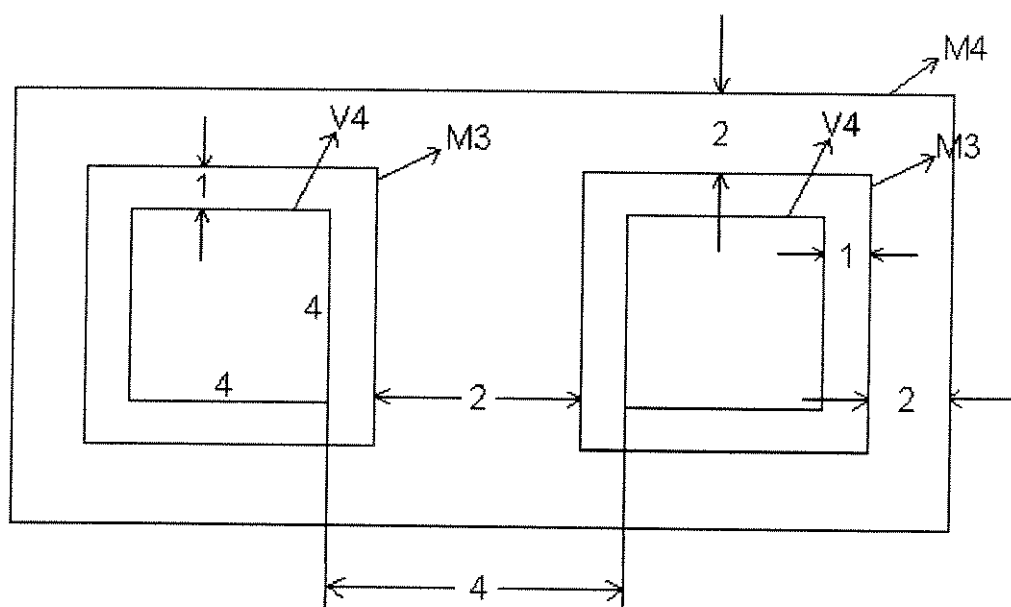


Figura 5.6 – Regras de tamanho e separação para a Via 4.

Existem duas variantes do processo HGaAs-III, diferenciadas pelo tipo de dielétrico usado entre as camadas de metal de interconexão. Na versão HGaAs-III “P” (comercial), o dielétrico usado é o *polyimide* ( $\epsilon_r = 3 \sim 3,5$ ), e na versão HGaAs-III “O” é o  $\text{SiO}_2$  ( $\epsilon_r = 3,9$ ); a maior  $\epsilon_r$  do  $\text{SiO}_2$  resulta em circuitos mais lentos, mas capazes de operar em uma faixa de temperatura militar ( $-55^\circ\text{C}$  a  $125^\circ\text{C}$ ), enquanto o *polyimide* reduz as capacitâncias e leva a circuitos mais rápidos, mas opera em uma faixa de temperatura menor ( $0^\circ\text{C}$  a  $100^\circ\text{C}$ ). Isso

acontece porque uma menor  $\epsilon_r$  do dielétrico intercamadas acarreta menores capacitâncias das interconexões. O projeto do *chip 7490-like* foi feito para a variante comercial HGaAs-III “P”.

Finalmente, existe a camada de *pad*, que especifica a localização e o tamanho dos *bonding pads* do *chip*. A camada *pad* também é um *design log* para o *Magic*, contendo as camadas de metal 2, 3 e 4, as vias entre elas e a camada da janela na passivação (a passivação é uma camada isolante de nitreto de silício depositada sobre todo o *chip*, para sua proteção; nos centros dos *pads* são abertas janelas na passivação para a soldagem dos fios de ligação do *chip* com a sua cápsula). Os *pads* são dimensionados pelo *Magic* de fora para dentro, isto é, da camada mais larga em direção à mais estreita, vistas de cima, conforme as regras da página 4-29 de [12]. Para que a janela na passivação tenha o tamanho recomendado, o tamanho total do *pad* tem que estar de acordo com essas regras, sabendo-se que o *Magic* emprega os valores mínimos de todas as distâncias entre bordas de camadas. As dimensões mínimas das camadas que compõem os *pads* são as seguintes:

Via 3: 105 x 68  $\mu\text{m}$ ;

Janela na passivação: 112 x 75  $\mu\text{m}$ ;

Via 4: 119 x 82  $\mu\text{m}$ ;

Metal 4: 126 x 89  $\mu\text{m}$ ;

Metal 3: 133 x 96  $\mu\text{m}$ ;

Metal 2: 137 x 100  $\mu\text{m}$  (que é a dimensão do *pad* completo).

Portanto, se um *pad* é desenhado com o tamanho total de 137 x 100  $\mu\text{m}$ , as suas camadas componentes são geradas com as dimensões acima; porém, o próprio CMP informou que as janelas na passivação sobre um *bonding pad* não precisam ter 112 x 75  $\mu\text{m}$  **obrigatoriamente**. Estas dimensões são apenas recomendadas, porque são grandes o suficiente para a soldagem dos fios de conexão, e menos área é desperdiçada. Assim, áreas de *pad* maiores são permitidas, desde que o *chip* resultante ainda caiba no seu encapsulamento. Importante é que os *pads* do *chip* devem ser todos iguais (todos do mesmo tamanho).

As regras de *layout* das pistas (*scribe lines* ou *saw lanes*), dadas na página 4-30 do manual da Vitesse, não precisam ser consideradas pelo projetista em uma rodada MPW, pois as *scribe lines* são feitas automaticamente pela *foundry*. O projetista só precisa saber que, dentro da área da cavidade do encapsulamento que for escolhido para alojar o *chip*, é preciso reservar uma borda de 25  $\mu\text{m}$  em cada lado para as *saw lanes*, conforme ilustrado na página 4-42 de [12]. Isto é, a cavidade do encapsulamento também inclui as larguras laterais para as *saw lanes*, que são medidas a partir da borda externa dos *pads*.

### 5.3. Problemas de *Layout* de CIs.

Para desenhar-se MESFETs de alta capacidade de corrente (portanto muito largos), a largura total da porta tem que ser dividida em vários “dedos” conectados em paralelo, que são como unidades de porta, ao invés de se desenhar uma única fita de metal de porta com a largura total do transistor. Isso equivale a colocar múltiplos MESFETs em paralelo de uma forma bastante compacta, mostrada na figura 5.7, chamada de “MESFET com portas interdigitadas”. Isso minimiza a resistência série do terminal de porta, que se fosse muito alta acarretaria uma considerável degradação do ganho. Além disso, contorna-se a dificuldade de

fabricação de uma fita muito comprida e fina de metal, que não teria uma boa precisão nas suas dimensões.

A largura total do MESFET da figura 5.7 é igual à largura de um dedo multiplicada por oito, onde oito é o número de dedos. O número de dedos é escolhido de modo a resultar em uma largura razoável de cada dedo; neste trabalho, não foram utilizados dedos com  $W_g$  maior do que  $50\text{ }\mu\text{m}$ . Um procedimento de otimização do *layout* de MESFETs de potência é apresentado em [85].

Logo no início do desenho de um *layout*, o projetista deve primeiro dimensionar os conjuntos de transistores em paralelo, para a capacidade de corrente dos mesmos, e somente depois interligar esses blocos de portas. Isso resultará em uma boa economia de tempo nas fases seguintes do trabalho. Quando MESFETs são conectados em paralelo, como na figura 5.7, é preciso que se saiba em detalhes quanta corrente flui e onde ela flui, para que cada tipo de camada que compõe o MESFET seja dimensionada respeitando-se seus limites de máxima densidade de corrente. Onde forem passar correntes altas, as linhas ou barramentos têm que ser dimensionados adequadamente.

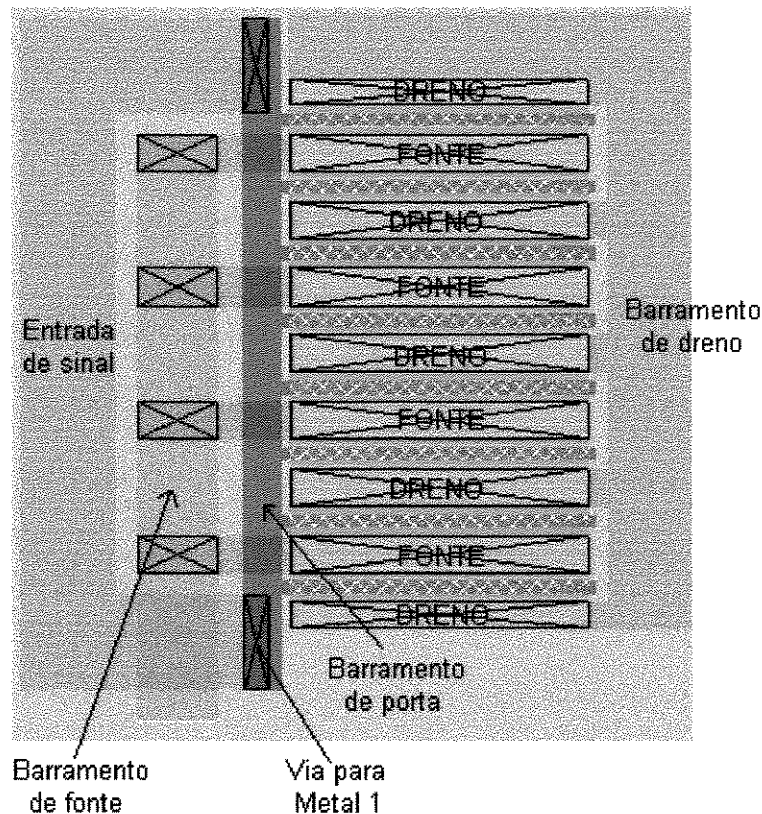


Figura 5.7 – Transistor de porta muito larga, realizado com oito FETs em paralelo.

Quanto maior a resistência total  $R_{\text{wire}}$  de uma trilha de metal em um circuito, maior será a variação de tensão ao longo dessa trilha quando percorrida por uma corrente  $I$ , porque maior será a queda  $I.R$  na resistência. Esse efeito é nocivo principalmente para os barramentos de alimentação, porque os potenciais de  $V_{DD}$  e terra vistos pelos FETs dentro do *chip*, já não serão iguais aos valores aplicados nos terminais do CI, e poderão apresentar variações de um FET para outro. Entretanto, os barramentos de alimentação devem assegurar tensões de  $V_{DD}$  e terra constantes e uniformes para todos os dispositivos no *chip*, o que evidencia a necessidade



de se procurar reduzir as quedas ôhmicas (I.R) de tensão nessas linhas, bem como suas indutâncias.

A resistividade de um metal aumenta proporcionalmente à temperatura, portanto a queda de tensão nas interconexões também aumenta na mesma proporção. Como o arquivo de tecnologia do *Magic* não considera variações de temperatura automaticamente para o cálculo dos parasitas, os coeficientes térmicos da tabela 4-1 de [12] não têm muita utilidade. Porém, a tabela 4-1 lista os valores de  $R_s$  em 85°C, e estes podem ser embutidos no arquivo de tecnologia para a extração de resistências de pior caso.

Se a corrente passando em uma linha de distribuição de alimentação sofre alguma variação transiente muito rápida  $dI/dt$ , como um pico de corrente, a indutância  $L$  dessa linha acarreta um transiente de tensão na mesma, segundo a equação (5.1). O valor do pico de tensão indutivo (ruído) nos barramentos de alimentação, limita a taxa de variação da corrente (*slew rate*) nos mesmos.

$$V = L \frac{dI}{dt} \quad (5.1)$$

Como a corrente de alimentação de um circuito DCFL é relativamente constante, a maior contribuição para os transientes de corrente vem do chaveamento dos *drivers* de saída (os *buffers* que acionam os *pads*). O ruído na alimentação devido a esses transientes de chaveamento é um dos principais problemas a serem evitados desde o projeto de um *chip*. Foi por isso que não se utilizou *drivers* do tipo “superbuffer” neste circuito, como explicado no Capítulo 4; estes *drivers* geram uma grande  $dI/dt$ . E os fios de conexão do *chip* com os pinos do seu encapsulamento acrescentam muita indutância no caminho das tensões de alimentação e sinais. A perturbação de tensão causada pela indutância pode ser até maior do que aquela causada pela resistência da linha, quando nela ocorre um pico de corrente [11].

O efeito de eletromigração [11] determina que a largura  $W$  de uma trilha deve ser dimensionada proporcionalmente à corrente máxima que passará na trilha durante a operação do circuito, para limitar a densidade de corrente na trilha a valores aceitáveis. A densidade de corrente é dada por  $I/A$ , onde  $A = W.t$ ,  $t$  é a espessura da trilha, e a largura  $W$  é sempre medida perpendicularmente ao sentido da corrente na trilha, como mostrado na figura 5.8.

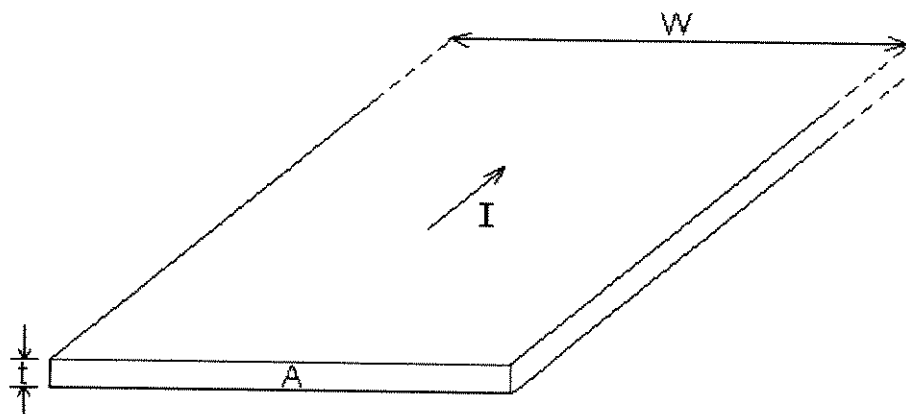


Figura 5.8 – Esquema de uma trilha de interconexão no *chip*.

Altas densidades de corrente aceleram o processo da eletromigração, diminuem o tempo médio para a falha da interconexão e reduzem a confiabilidade do circuito. Os limites

máximos de corrente para cada tipo de camada do processo HGaAs-III são listados na tabela 4-2 de [12], em mA/ $\mu\text{m}$  de W, em uma trilha da camada considerada. Basta que esses limites de corrente sejam observados no dimensionamento das larguras de interconexões, vias e contatos de porta dos MESFETs, e o projetista não precisará mais se preocupar com as densidades de corrente. A eletromigração é mais uma razão, além da redução da resistência ôhmica das linhas, para que as trilhas de alimentação sejam feitas tão largas quanto possível em VHSICs MSI.

É por tudo isso que as camadas de metal usadas para os barramentos de  $V_{DD}$  e terra devem ser as mais externas disponíveis, haja vista que estas são mais espessas (maior  $t$ ) e por isso possuem menor  $R_s$  e suportam maiores correntes por unidade de largura do que as outras, e com menor queda de tensão. Neste trabalho, como já foi dito, as camadas de metal 3 e metal 4 são usadas para a distribuição global da alimentação por todo o *chip*, aquela na forma de barramentos e esta na forma de um plano de terra sobre o circuito; enquanto linhas de metal 1 e metal 2 distribuem as tensões de alimentação localmente para cada porta lógica. Sempre que for possível, é interessante aumentar a largura W das trilhas **de alimentação**, porque isso aumenta a corrente máxima aceitável nessas trilhas (e diminui as suas densidades de corrente), e também reduz as variações de tensão.

As topologias em que as entradas das portas lógicas são realizadas por FETs em fonte-comum, como a família DCFL, são especialmente sensíveis às variações do potencial de terra, porque um potencial maior de terra na fonte dos EFETs reduz a NM baixa das portas. Foi por isso que optou-se por usar o plano de metal 4 como um plano de terra. Felizmente, DCFL não é tão sensível a variações de  $V_{DD}$  porque esta tensão é aplicada no dreno de um FET que está ligado como fonte de corrente, a qual é aproximadamente constante contanto que ele esteja na saturação. Assim, os níveis lógicos são menos sensíveis à variação da tensão  $V_{DD}$ .

Em circuitos com longas trilhas de interconexão, as capacitâncias parasitas do circuito são muito maiores do que as capacitâncias internas dos transistores, servindo de cargas para as portas lógicas. Portanto, as capacitâncias do circuito precisam ser consideradas para que o simulador forneça previsões confiáveis do  $t_{PD}$  dos sinais. A tabela 4.5b (pg. 4-40) de [12] fornece os valores das capacitâncias de placas paralelas e de frangeamento, devidas aos *overlaps* entre cada par de tipos de camada e entre cada camada e o substrato. A tabela 4.5b também está embutida na seção *extract* do arquivo de tecnologia “cmp.tech26”; portanto, as capacitâncias parasitas das interconexões são calculadas e extraídas automaticamente pelo software *Magic*, e o projetista não precisa fazer cálculos manuais de capacitância para o *chip*.

### 5.3.1. Layout de CIs Digitais de Alta Velocidade.

As interconexões metálicas entre os elementos de um CI de alta velocidade têm uma influência muito grande sobre o seu desempenho, por efeito do atraso de propagação do sinal ao longo das mesmas. O tempo  $t_{PD,LT}$  que o sinal demora para percorrer o comprimento L de uma linha de transmissão é dado por:

$$t_{PD,LT} = \frac{L}{v_p} \quad (5.2)$$

onde  $v_p (= c/\sqrt{\epsilon_r})$  é a velocidade de propagação do sinal na linha.

Nas frequências de interesse para este trabalho, as interconexões de sinal dentro do *chip* podem ser modeladas por elementos concentrados R e C ([11], [12], [70]), desde que os tempos de subida e descida dos sinais envolvidos sejam pelo menos dez vezes maiores que seus atrasos de propagação nessas interconexões, ou seja, se  $t_{PD,LT} \ll t_r$  e  $t_f$ , as linhas nesse caso são ditas “curtas”. Esta simplificação do modelamento aplica-se bem às interconexões em CIs MSI de GaAs [70], como o *chip* deste trabalho, que possui até 3 mm de lado. Por exemplo, o  $t_{PD,LT}$  para uma linha de sinal longa, com metade do comprimento do *chip* ( $L = 1,5$  mm) e fabricada pelo processo HGaAs-III ( $\epsilon_r = 4,5$ ), é de aproximadamente 10,6 ps pela equação (5.2), ou seja, muito pequeno comparado aos tempos de transição internos  $t_r$  e  $t_f$ , que são da ordem de 100 ps a 300 ps. Como o *chip* 7490-like é pequeno, suas interconexões são “curtas”, e portanto seus atrasos  $t_{PD,LT}$  podem ser simulados por R e C concentrados.

Quanto maior a largura  $W$  de uma linha, o valor da sua resistência série  $R_{wire}$  diminui exponencialmente [70] e, em princípio, deve-se procurar realizar uma pequena  $R_{wire}$ . Porém, utilizando-se os metais 2 e 3 (que têm as menores  $R_s$ ) para as interconexões mais longas de sinal e alimentação do CI, pode-se desprezar a  $R_{wire}$  das linhas [70]. Assim, toda interconexão do *chip* pode ser modelada apenas por uma capacitância parasita total  $C_{wire}$  ([12], [70]), que reúne as contribuições de placas paralelas e de frangeamento, da linha ao substrato e aos cruzamentos com outras linhas em planos diferentes. O atraso de um sinal é determinado principalmente pela  $C_{wire}$  e pela resistência de saída da porta lógica que a aciona,  $R_{out}$ ; esta resistência é calculada pelo HSPICE, e depende da  $W_g$  dos FETs, da tensão de alimentação e do nível lógico.

A minimização de  $C_{wire}$  de uma linha é mais crucial, para a operação em alta velocidade do circuito, do que a minimização de  $R_{wire}$ , porque  $R_{wire}$  é quase sempre mascarada por  $R_{out}$ . Então as linhas **de sinal** não devem ser desenhadas muito mais largas do que o necessário, pois quanto maior a largura  $W$  de uma interconexão, maior será a sua  $C_{wire}$ . Um gráfico de  $C_{wire} \times W$  para GaAs, é mostrado em [70].

Quanto mais rápida for uma família lógica, mais sensível ela será ao ruído proveniente das transições [73], que surge nas interconexões de sinal e de alimentação, em consequência principalmente de: *crosstalk* entre linhas, variações de tensão nas resistências  $R_{wire}$  das linhas de alimentação, e picos de tensão resultantes de transientes de corrente nas indutâncias das linhas. Essas observações são aplicáveis tanto interna quanto externamente ao CI.

O *crosstalk* é o acoplamento, de natureza predominantemente capacitiva, entre linhas de sinal próximas dentro de um *chip*, resultando em interferência entre os dois sinais. Problemas de *crosstalk* surgem principalmente entre linhas de interconexão longas e paralelas onde trafeguem sinais distintos de alta velocidade, sendo a interferência entre eles tanto mais intensa quanto mais próximas estiverem essas linhas. Para reduzir os efeitos de *crosstalk*, o *layout* deve alternar linhas de sinal com linhas de terra, pois linhas de sinal coplanares (de uma mesma camada de metal) podem sofrer um forte acoplamento se estiverem distantes de linhas de terra, ou se houver só uma linha de terra para muitas linhas de sinal. O encapsulamento também contribui para o *crosstalk*, portanto o posicionamento dos *pads* também deve alternar *pads* de sinal com *pads* de terra onde for possível. Neste projeto, procurou-se seguir essa recomendação dentro das restrições impostas pelo *layout* e pela pinagem do *chip*.

O *crosstalk* pode ser bem controlado apenas com o posicionamento relativo das linhas de sinal de entrada e de saída. A maior parte do ruído induzido se propaga no sentido oposto ao do sinal que o induziu, o que significa que entradas de blocos lógicos não devem estar ativas durante as transições de *drivers* de saída próximos. Este requisito afeta diretamente o posicionamento dos *pads* de entrada e saída do *chip*, portanto deve ser considerado no início do *layout*. Outro cuidado no *layout* para evitar o *crosstalk* é não desenhar linhas de sinal paralelas muito próximas uma da outra.

Em circuitos projetados para altas velocidades de operação, a topologia adotada para o desenho do *layout* é crucial para o seu desempenho, porque ela estabelece os valores das capacitâncias parasitas do circuito, que constituem a fonte dominante de atraso e *crosstalk*. No *layout* do circuito 7490-like, procurou-se empregar a notação em anel (*ring notation*), criada para reduzir o acoplamento entre as linhas de sinal de alta frequência e os barramentos de alimentação de  $V_{DD}$  [86]. Na notação em anel, o barramento de terra (GND) deve ser roteado entre o barramento de  $V_{DD}$  e os blocos da lógica do circuito, a fim de desacoplar o ruído de alta frequência, devido aos chaveamentos, das linhas de  $V_{DD}$ ; a mesma estratégia é adotada para o posicionamento dos *pads*. A notação em anel também permite realizar-se *layouts* mais rápidos e compactos [86]. O *layout* de um inversor, por exemplo, adquire a aparência mostrada na figura 5.9.

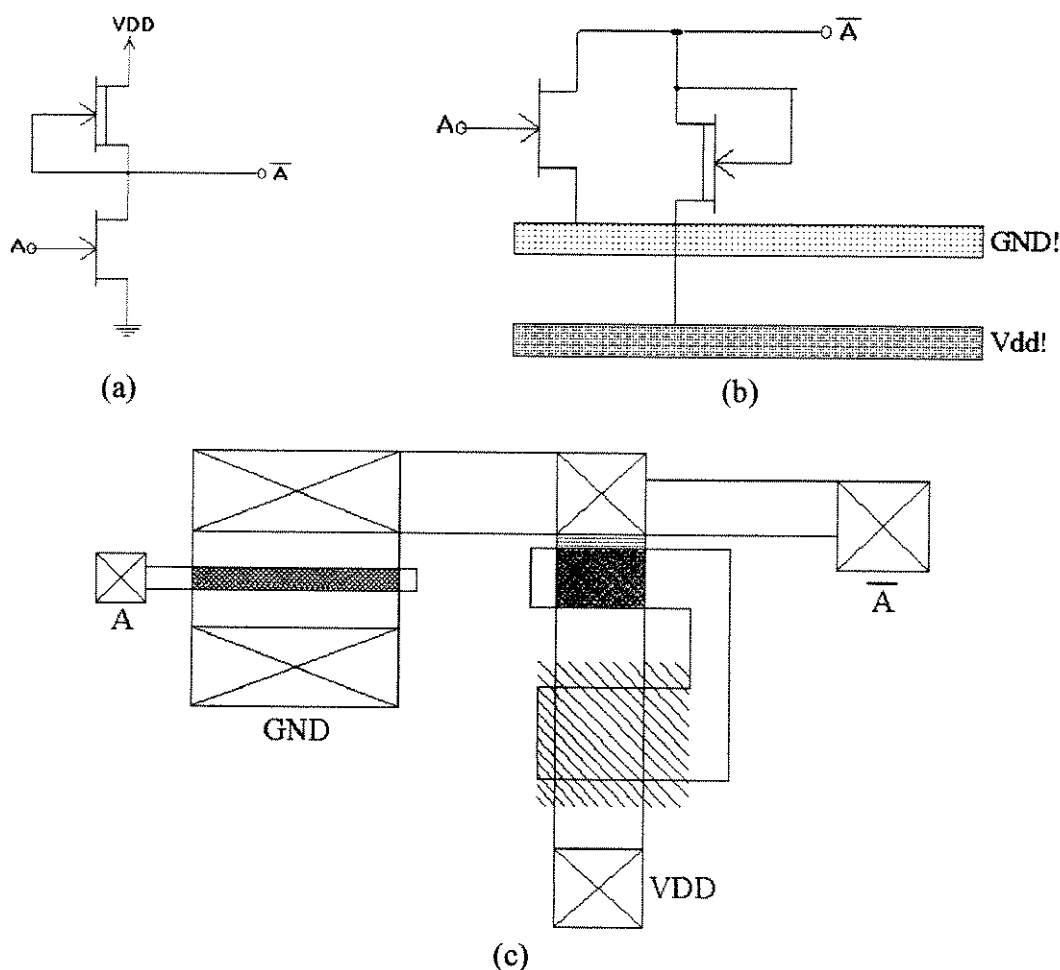


Figura 5.9 – *Layout* de um inversor: (a) estilo convencional, (b) estilo da notação em anel, e (c) *layout* real na notação em anel.

## 5.4. Desenho do *Layout* do CI 7490-like no *Software Magic*.

### 5.4.1. A Configuração do Arquivo de Tecnologia para o *Magic*.

O *layout* do CI 7490-like foi desenhado com a utilização do CAD *Magic*, configurado para o processo de fabricação HGaAs-III pelo arquivo de tecnologia “cmp.tech26”. O *layout* foi desenhado no estilo Manhattan [11], em que todas as geometrias têm que ser ou horizontais ou verticais. A referência [11] contém um breve tutorial para usuários do *Magic*, e explica a estrutura dos seus arquivos de tecnologia. Essas mesmas informações podem ser encontradas com muito mais detalhes nas documentações oficiais sobre o programa [87] e [88], as quais são tutoriais que precisam ser bem estudados pelo projetista de CIs.

O programa *Magic* é, em princípio, independente de qualquer processo, sendo configurado para o processo de uma *foundry* específica por meio de arquivos de tecnologia, que contém o conjunto das regras de projeto suportadas pela *foundry*. A medida em que os processos de fabricação são aprimorados, as regras vão sendo atualizadas, e a *foundry* deve fornecer novas versões do arquivo de tecnologia.

A regra de projeto G2 (pág. 4-3 de [12]) exige que o *layout* seja todo desenhado em um grid de  $0,1\ \mu\text{m}$  ( $\lambda = 0,1\ \mu\text{m}$ , onde  $\lambda$  é a medida do lado do quadrado). Porém, o arquivo de tecnologia hgaas3.tech26, de que se dispunha até então, implementava um *grid* de  $0,4\ \mu\text{m}$  no *Magic*, portanto era impróprio para uso neste projeto, já que não se poderia desenhar, por exemplo, um EFET de  $L_g = 1\ \mu\text{m}$  nesse grid. Foi solicitado um novo arquivo de tecnologia ao CMP, que enviou o arquivo “cmp.tech26”; este implementa um *grid* de  $0,1\ \mu\text{m}$ , e passou a ser usado neste projeto em substituição ao arquivo de tecnologia anterior. Quanto maior a resolução do grid, isto é, quanto menor o valor de  $\lambda$ , maior a flexibilidade de desenho do *layout*.

Entretanto, durante o desenho do *layout* do circuito 7490-like, quando se tentou pela primeira vez desenhar um *pad* do *chip*, percebeu-se que não existia a camada de metal 4 para a interconexão desse *pad* com o circuito. Acontece que o metal 4 e a via 4 não estavam implementados no arquivo “cmp.tech26”, o qual tinha sido escrito somente para projetos com três camadas de metal.

O problema é que o contrato do CMP com a *foundry* Vitesse para a realização de “fornadas” multi-projeto estipula que é obrigatório o uso de *pads* de quatro metais em todos os circuitos, mesmo nos de baixa complexidade. Isto é, mesmo que um circuito seja tão simples que nem precise usar os quatro níveis de metal, os seus *pads* terão que ser de quatro metais, conforme a página 4-29 de [12].

Nesse momento, contactou-se o CMP para que este fornecesse um arquivo de tecnologia atualizado, com os quatro níveis de metal e grid de  $0,1\ \mu\text{m}$ . Obteve-se a inesperada resposta de que não havia um tal arquivo disponível, mas ainda assim era mantida a exigência do uso de quatro metais nos *pads*, para os projetos.

Para resolver esse problema, recorreu-se ao antigo arquivo “hgaas3.tech26”, que possuía a definição dos quatro metais, apesar de seu grid ser de  $0,4\ \mu\text{m}$ . Foi preciso parar o desenho do *layout* do 7490-like para estudar a estrutura interna dos arquivos de tecnologia do *Magic*, explicada na referência [88], e assim poder modificar cada uma das diversas seções do arquivo “cmp.tech26” para incluir os quatro metais, tendo o arquivo “hgaas3.tech26” como

modelo para essas alterações. Assim, foi escrita uma nova versão do arquivo “cmp.tech26”, com os quatro níveis de metal e as regras de DRC correspondentes, e que está listado no Apêndice 2. Depois de pronto o arquivo, o trabalho de *layout* pôde ser retomado.

As regras de projeto para HGaAs-III estão listadas no Capítulo 4 de [12], mas a sua verificação automática pelo computador depende de elas também estarem incluídas na seção “drc” do arquivo de tecnologia para o *Magic*. No arquivo distribuído pelo CMP, “cmp.tech26”, algumas regras de projeto do manual não têm comandos de verificação automática, então o *Magic* não pode acusar a ocorrência de violações a essas regras. Dessa forma, o *layout* pode conter erros de DRC que, mesmo estando expressos no manual de projeto da Vitesse, não sejam acusados pelo DRC do *Magic*. É fundamental que o projetista verifique visualmente essas regras, ou edite o arquivo de tecnologia para incluir a verificação automática para elas.

Todas as dimensões e separações listadas nos arquivos CIF gerados pelo *Magic* são, na verdade, iguais ao dobro do valor da dimensão ou espaçamento correspondente no *layout* do *chip*, e estão expressas em centimicrons. Essa relação de 2:1 do CIF para o *layout* deve-se à configuração da seção *cifoutput* do arquivo “cmp.tech26” para o *Magic*, que contém as regras para a geração de arquivos de *layout* no formato CIF. Preferiu-se não alterar essa configuração porque implicaria em também ter que mudar a seção *cifinput* do mesmo arquivo de tecnologia. Portanto, para se saber o valor de uma medida qualquer no *layout*, em  $\mu\text{m}$ , é preciso dividir o seu valor correspondente, listado no arquivo CIF, por 200.

É preciso atentar também para o fato de que todas as dimensões medidas com o comando *box* do *Magic* são dadas em decimicrons para esta tecnologia, porque  $\lambda = 0,1 \mu\text{m} = 1 \text{ d}\mu\text{m}$  ( $1 \mu\text{m} = 10 \lambda$ ), e o *Magic* somente mede múltiplos inteiros de  $\lambda$ .

#### 5.4.2. As Etapas para o Desenho do Layout.

O *layout* do *chip 7490-like* foi desenhado com o uso de hierarquia, isto é, primeiro criou-se células de *layout* para todos os diferentes tipos e tamanhos de portas lógicas e *buffers* usados no circuito, para essas células serem depois empregadas na composição do *layout* do *7490-like*. Todas as células foram criadas diretamente a partir dos subcircuitos de portas e *buffers* usados nas simulações em HSPICE. Portanto, para cada subcircuito do HSPICE, em forma de *netlist*, foi desenhada uma célula de *layout*, no *Magic*. As células foram todas desenhadas na notação em anel.

As simulações no HSPICE forneceram os valores das correntes DC e transientes em cada FET, para vários instantes de tempo, valores esses que são necessários para o dimensionamento dos FETs. Para o desenho destas células no nível de máscaras, primeiro desenhou-se, em folhas de papel, os esquemáticos de cada um dos subcircuitos de portas lógicas e *buffers* criados no HSPICE, anotando-se nesses esquemáticos as dimensões dos FETs e todas as correntes envolvidas. Assim, em um mesmo lugar tinha-se toda a informação necessária para desenhar os FETs e suas interconexões no *Magic*, facilitando a consulta. Esses esquemáticos, feitos no nível de transistores, têm a vantagem de permitir uma correspondência de um para um com os transistores do *layout*. Todas as células criadas para o *layout* do *chip 7490-like* estão mostradas no Apêndice 4, juntamente com seus esquemáticos ao nível de transistores.

O uso de hierarquia é recomendado pelo próprio tutorial do *Magic* [11], pois a extração de um circuito hierárquico é muito mais rápida do que a extração de uma versão planificada do mesmo circuito. O ponto mais importante a considerar quando se faz um *layout* hierárquico é a colocação de *labels* em todos os terminais de entrada e saída de cada célula, que nada mais são do que os nomes dos nós. O uso de *labels* descritivas para os nós de cada célula facilitará enormemente o entendimento da *netlist* do circuito extraído, depois de pronto o *layout*. Se não forem usadas *labels*, o *Magic* atribui nomes default para os elementos e nós, na *netlist* extraída do circuito, e será muito difícil fazer a correlação entre os elementos e nós dessa *netlist* com os seus equivalentes no *layout*. Além disso, as *labels* facilitam o reuso de células por outros projetistas, em um ambiente de projeto cooperativo.

É recomendado que cada nó do *layout* receba o mesmo nome que já possui na *netlist* do circuito para o HSPICE; ou seja, os nomes dos nós são copiados da *netlist* do HSPICE para o *layout*. Para detalhes sobre o uso de *labels*, devem ser consultados os tutoriais [11] e [87].

Para iniciar-se o desenho do *layout* do CI 7490-like no método *full-custom*, foi necessário antes reunir:

- os esquemáticos das diversas células de portas lógicas e *buffers* usadas no circuito, ao nível de transistores;
- os esquemáticos **lógicos** detalhados dos blocos do circuito (os flip-flops usados no CI) e do circuito global do 7490-like (figuras 5.15 a 5.18 e figura 4.15);
- e os resultados das simulações finais do circuito no pior caso de dissipação de potência (biblioteca *fast* a 85°C), para dimensionar-se as trilhas de metal de interconexão em função das correntes máximas que passarão por elas.

#### 5.4.2.1. *Layout* dos Flip-Flops.

Dispondo das células das portas lógicas, iniciou-se o *layout* do circuito 7490-like, começando pelo flip-flop FF1. Este foi composto pela conexão das células adequadas em um arranjo linear na notação em anel, reproduzindo a simetria natural das estruturas de FFs. Neste FF, e em todo o circuito, as ligações entre a saída de uma porta lógica e a entrada da porta seguinte foram feitas com metal 1, o qual depois ligava-se ao gate do EFET dessa porta por meio de uma via 1g. Para minimizar a resistência dessa conexão, a via 1g foi sempre colocada o mais perto possível do terminal de gate do transistor.

Uma recomendação forte para o *layout*, que no entanto não chega a ser uma regra de projeto, é **evitar passar trilhas de metal 2 diretamente sobre MESFETs e diodos**. Tal proximidade de dois sinais de alta frequência, no metal e no dispositivo, traria maiores problemas de *crosstalk*. Também deve-se evitar desenhar trilhas metálicas com bordas coincidentes ou muito próximas, especialmente bordas de metal 2 muito próximas de bordas de metal de porta [11]. Os efeitos indesejáveis resultantes consistem de maiores capacitâncias por frangeamento, que aumentam os atrasos dos sinais e o *crosstalk* entre as trilhas. Essas recomendações foram seguidas em todo o *layout* do CI 7490-like, para não ocasionar problemas de desempenho.

Um esquemático do FF1 a nível de transistores foi desenhado na notação em anel, para se assemelhar ao *layout* correspondente. A figura 5.10 mostra esse “esquemático orientado a *layout*” (*layout-oriented schematics*) para o FF1, realçando os barramentos de GND e  $V_{DD}$  que devem “correr” entre as duas metades do FF, isolando-as entre si. E, ao mesmo tempo, os



barramentos de GND isolam o barramento de  $V_{DD}$  da lógica do circuito. A figura 5.11 mostra o *layout* real do FF1 feito no *Magic*.

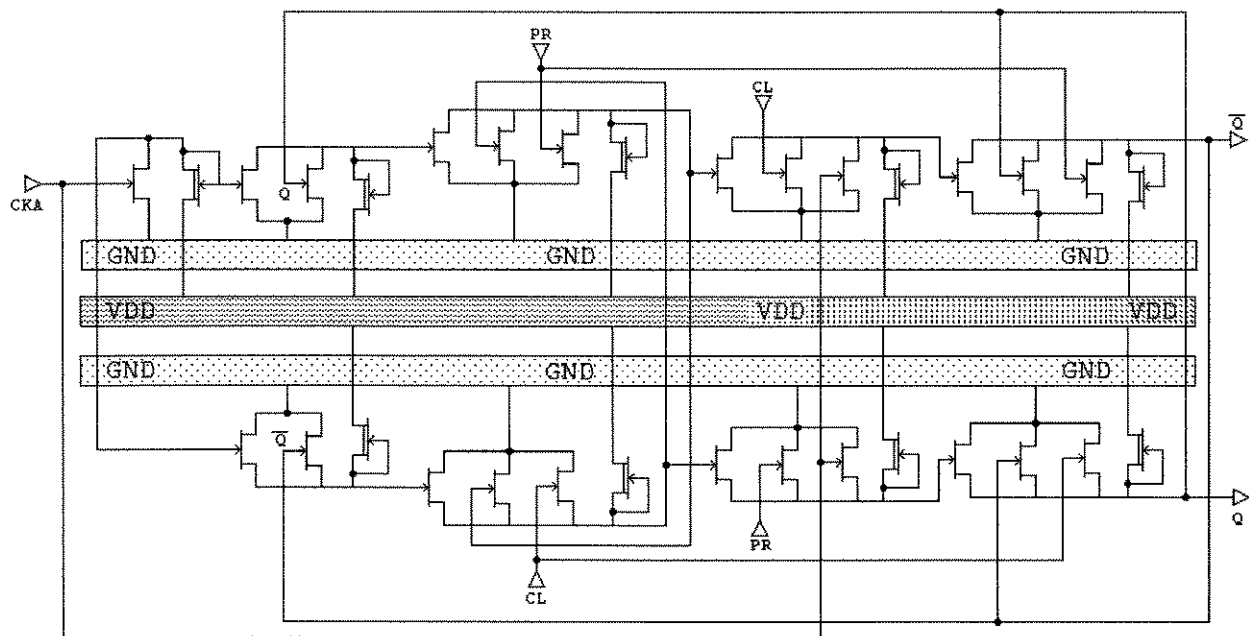


Figura 5.10 – Esquemático do FF1 na notação em anel, destacando os barramentos de tensões de alimentação.

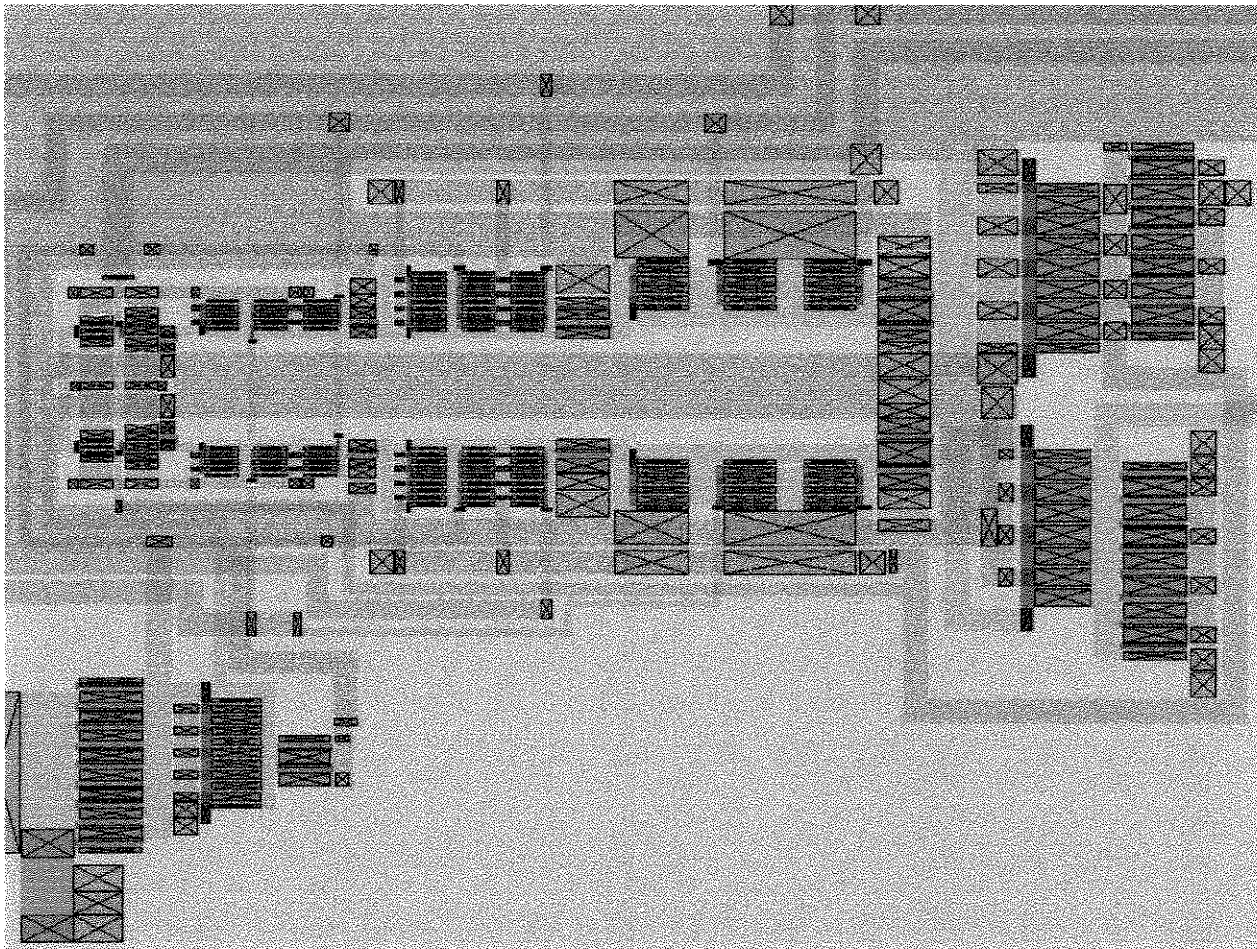


Figura 5.11 – *Layout* do FF1 do CI 7490-like.



Os *layouts* de todos os flip-flops foram desenhados procurando-se minimizar os atrasos internos de cada FF; por isso, suas portas foram posicionadas tão próximas umas das outras quanto possível. Percebeu-se logo a necessidade de alguns FFs ficarem na vertical e outros na horizontal, para caberem na área máxima do *chip*. Assim, os quatro FFs resultaram blocos bastante compactos. A figura 5.12 mostra o *layout* final do FF3, exceto pelo seu *buffer* de saída, que fica no alto à direita; esse FF teve que ser desenhado numa disposição vertical pelo problema da limitação de área. Os *layouts* dos demais FFs e de outros blocos do circuito são mostrados no Apêndice 4.

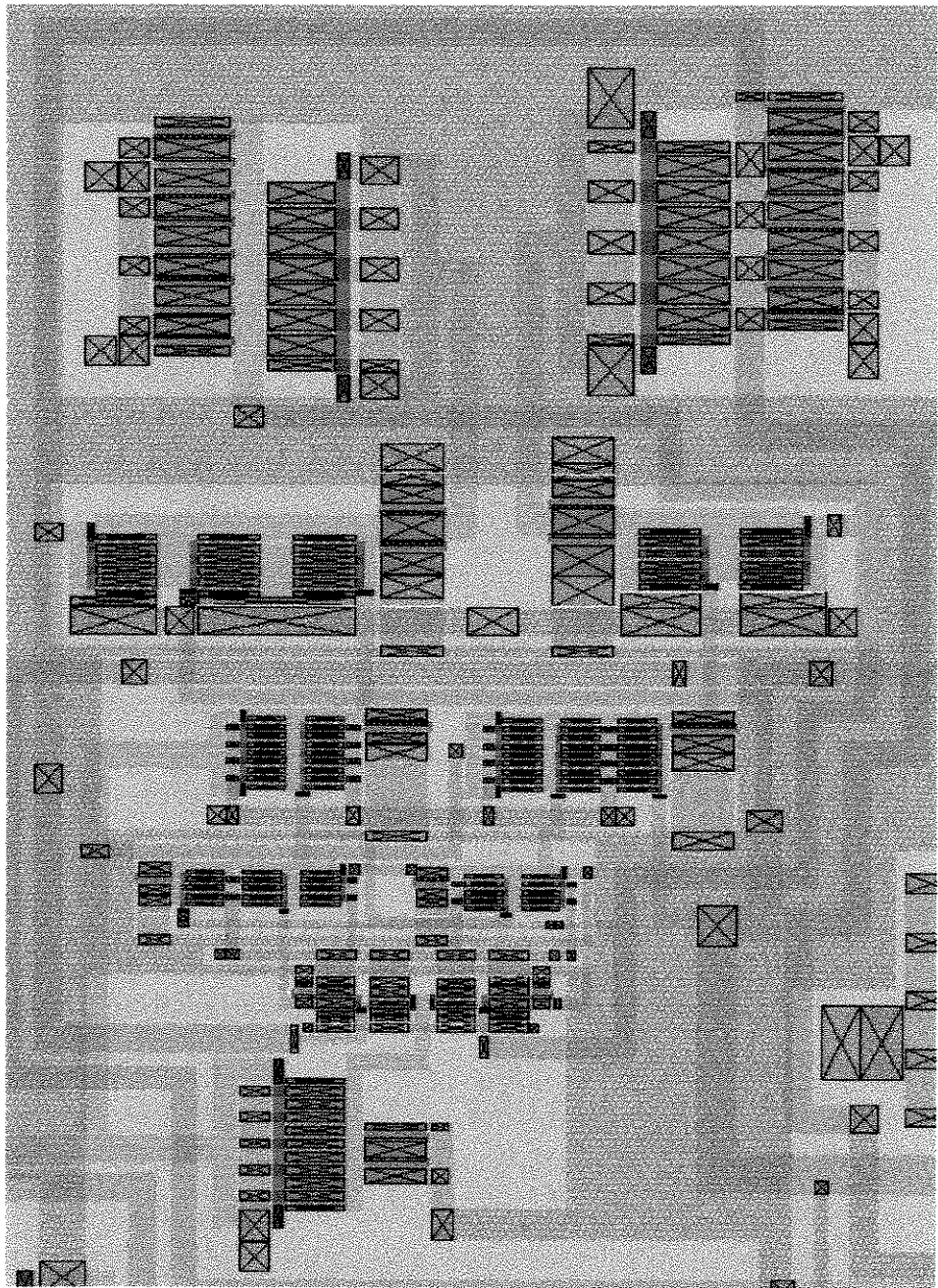


Figura 5.12 – *Layout* final do FF3.

A figura 5.13 mostra o posicionamento (*floorplanning*), feito manualmente, dos FFs e demais blocos do *layout* do CI; esse posicionamento final só foi alcançado após alguns rearranjos de células e blocos funcionais, para se obter a redução dos comprimentos das

interconexões. O roteamento de todo o *chip* também foi feito manualmente, procurando-se atender ao objetivo geral de minimizar todos os comprimentos de linhas de sinal.

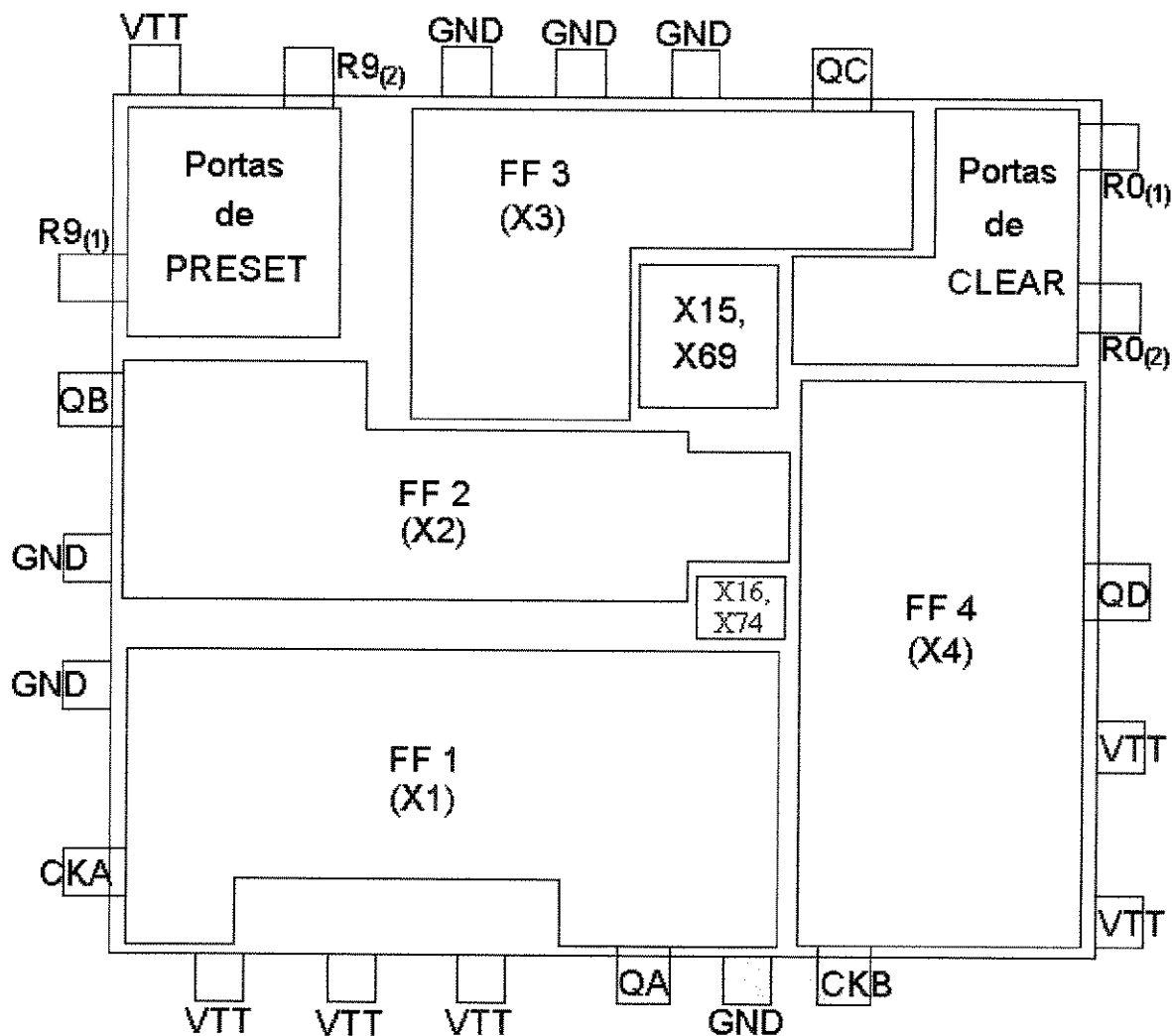


Figura 5.13 - *Floorplan* do CI 7490-like.

O posicionamento dos blocos do circuito determina diretamente a designação das funções dos *pads* e a pinagem do futuro CI encapsulado. Neste momento, o posicionamento das células padrão dos FFs foi realizado com todo o cuidado para que as saídas do *chip* (QA, QB, QC e QD) ficassem bem separadas entre si, minimizando-se interferências mútuas; disso resultou uma saída em cada lado do *chip*, como pode-se ver na figura 5.13. A saída QA e a entrada CKB foram colocadas bem próximas porque, em todas as configurações, ou QA liga-se a CKB ou somente uma das duas é usada, ficando a outra sem sinal. Por exemplo, a divisão por  $N = 2$  só usa o FF1 (CKA e QA), e nas divisões por 3 e por 5, o clock é aplicado em CKB e QA é inoperante.

Infelizmente não se conseguiu situar os *pads* de *Preset* e *Clear* nos pontos médios dos lados do *chip*, por isso eles estão mais distantes de alguns *pads* de saída do que de outros.

Uma margem de tolerância foi acrescentada a todas as larguras de trilhas de interconexão, onde foi possível. O dimensionamento das larguras das interconexões e das vias inter-camadas levou em conta principalmente os limites máximos de densidade de corrente,

em mA/ $\mu\text{m}$  de largura, para cada tipo de camada. As simulações no HSPICE forneceram os valores das correntes DC em cada interconexão, para vários instantes de tempo, valores esses que são necessários para o dimensionamento das mesmas.

Para garantir-se a operação do CI em alta velocidade, as interconexões que representam os nós críticos, isto é, que transportam os sinais com as maiores frequências (de clock), devem ter suas capacitâncias parasitas minimizadas. Essas conexões críticas tiveram seus comprimentos tão reduzidos quanto foi possível, além de que foram feitas em metal 2 ou metal 3, que apresentam menores  $C_{\text{wire}}$ . Os nós críticos são determinados por análise lógica de um circuito digital, e para este circuito, são as linhas de CKA, CKB, *Preset* e *Clear* dos FFs.

#### 5.4.2.2. Projeto do Resistor de Terminação de Entrada.

A célula **inbuffer** (figura 3.68) é o circuito de proteção e regulação de corrente de entrada, e é ligada diretamente aos *pads* de entrada de sinal do *chip 7490-like*. Essa célula possui um resistor de *pull-down* de  $250\ \Omega$ , que propicia uma resistência total de aproximadamente  $50\ \Omega$ . Resistores são criados como uma mesa de área ativa apenas, que será convertida a implante  $n^+$  durante a implantação, da mesma forma que as regiões de fonte e dreno dos MESFETs. A partir da  $R_s$  do implante  $n^+$  [12], o valor da resistência de um resistor é calculado pela expressão:

$$R = R_s \frac{L}{W} \quad (5.3)$$

onde

$R_s = 210\ \Omega \pm 10\%$ , para o processo HGaAs-III;

$L$  é o comprimento do resistor;

$W$  é a largura do resistor.

$W$  deve ser dimensionado primeiro, em função da corrente máxima que vai passar no resistor, de acordo com o limite de corrente máxima no implante  $n^+$ , dado na tabela 4-2 de [12]. Escolhendo  $W$ , determina-se  $L$  a partir da equação (5.3). Entretanto, é bom lembrar que a resistência final desse resistor é o resultado da soma das resistências da área ativa e dos contatos dessa área ativa com o metal de interconexão.

Os DFETs reguladores de corrente têm que ser muito largos, sendo feitos como vários dedos de porta em paralelo. Em todo o circuito, as células **clkbuffer** são as que vêm logo depois das células **inbuffer**, recebendo toda a corrente fornecida por estas. Por isso o EFET de entrada da porta **clkbuffer** também teve que ser realizado com o paralelismo de vários *gates*. Quanto maior o número de dedos de *gate* em paralelo do EFET, maior a corrente total que poderá ser suportada na entrada dessa porta lógica; portanto, este EFET deve ser bem particionado em muitos EFETs em paralelo. Dessa forma, a corrente total de entrada em uma **clkbuffer**, proveniente do DFET regulador de corrente de **inbuffer**, se divide por um número grande de *gates* de transistores, e a corrente  $I_{GS}$  não será excessiva em nenhum deles.

Os *layouts* das células **inbuffer** e **clkbuffer** são mostrados no Apêndice 4. Os EFETs de **clkbuffer** foram desenhados com um número de *gates* suficiente para que a corrente de **inbuffer** não lhes cause dano, se  $V_{in}$  estiver dentro de uma faixa de 1,5 V em torno da excursão nominal da entrada.

#### 5.4.2.3. Projeto da Distribuição da Alimentação e dos *Pads*.

A maior corrente do circuito é a sua corrente de alimentação (de  $V_{DD}$  e terra), cujos valores de pior caso, em vários instantes, são obtidos a partir das simulações. A distribuição das correntes de alimentação no interior do CI 7490-like é feita por meio de barramentos de  $V_{DD}$  (feitos de metal 3), e do plano de terra interno de metal 4, o qual distribui eficientemente o potencial de terra para todos os pontos do *chip*, reduzindo tanto a queda I.R quanto a indutância de terra e o *crosstalk* entre sinais. Onde foi possível, as conexões de terra das portas lógicas a esse plano foram feitas por uma sequência de vias intermediárias, até chegar-se à via 4. Nos FFs, foram feitas pequenas linhas locais de terra, ligadas em suas duas extremidades ao metal 4, para conectar apenas algumas portas lógicas vizinhas ao terra.

Múltiplos *pads* de  $V_{DD}$  e de terra devem ser usados no *chip*, o que implica em múltiplos pinos de  $V_{DD}$  e terra no CI encapsulado. Há várias razões para **não** se tentar usar um único *pad* para cada tensão de alimentação do *chip*, isto é, só um *pad* para  $V_{DD}$  e um outro para terra. Em primeiro lugar, dessa forma ambos os *pads* terão que ser dimensionados para suportar toda a corrente de alimentação do *chip* (incluindo uma margem de segurança), e ficarão grandes demais. Assim, a área total do *chip* também resulta muito grande, e quanto maior a área do *chip*, maior o seu custo de fabricação. Além do mais, como já foi dito, a área da cavidade do encapsulamento do CI inclui também espaços laterais para as *sawing lanes*, portanto a área máxima aceitável do *chip* é um pouco menor, descontando-se as larguras das *sawing lanes*. E projetando-se os *pads* dessa maneira, o *chip* pode ficar tão grande que não caiba no encapsulamento pretendido para ele.

Em vez de se fazer poucos *pads*, cada um sendo grande o suficiente para suportar toda a corrente prevista para sua função no *chip*, o método correto é fazer-se muitos *pads* de tamanho pequeno, cada um conduzindo uma fração da corrente total. A quantidade de *bonding pads* em volta do circuito aumenta, mas o tamanho de cada um diminui bastante, o que diminui a área total do *chip* e, o que é mais importante, diminui o seu custo. O maior número de *pads* também permite ao projetista reduzir tanto o ruído indutivo quanto as quedas I.R nos barramentos de alimentação, como será explicado adiante.

Todos os *pads* do *chip* 7490-like foram feitos com o tamanho recomendado em [12], de  $137 \times 100 \mu\text{m}$ , onde o lado de  $100 \mu\text{m}$  é o que está voltado para o interior do circuito, e é onde se ligam as linhas de sinal e alimentação do/para o *chip*. Essas linhas podem ser de metal 2, metal 3 ou metal 4, e suas larguras máximas, para cada tipo de metal, são limitadas à largura do metal do mesmo tipo nos *pads* aos quais elas se ligam. Portanto, a partir dos *pads* do *chip*, pode ser desenhada uma trilha de metal 2 de até  $100 \mu\text{m}$  de largura, ou uma trilha de metal 3 de até  $96 \mu\text{m}$ , ou uma trilha de metal 4 de até  $89 \mu\text{m}$ , o que é esquematizado na página 4-29 de [12] e na seção 5.2 deste Capítulo. Dependendo de qual das três camadas de metal vai ligar o *pad* ao circuito, a largura da via correspondente estabelece a corrente máxima por *pad*, pois a máxima corrente que pode passar através de uma via é determinada pela sua largura perpendicularmente à direção do fluxo da corrente [12].

Para *pads* ligados por metal 2, a via 3, de  $68 \mu\text{m}$  de largura (ver seção 5.2 acima), suporta uma corrente máxima de  $I_{\text{maxP2}} = 135 \text{ mA}$ . Para *pads* ligados por metal 3, a largura de  $82 \mu\text{m}$  da via 4 suporta  $I_{\text{maxP3}} = 229 \text{ mA}$ , e nos *pads* ligados por metal 4, a corrente não passa mais pela via 4, assim a largura de  $89 \mu\text{m}$  do metal 4 suporta  $I_{\text{maxP4}} = 249 \text{ mA}$ .

O número mínimo de *pads* para uma mesma função do CI tem que satisfazer, em primeiro lugar, aos limites de máxima corrente através da conexão de cada *pad* com o *chip*. A potência total máxima do circuito é determinada pelas simulações, logo sabe-se imediatamente qual a corrente máxima de alimentação consumida pelo CI. Dividindo-se esta corrente pela corrente máxima em um *pad* de metal 3 (*pad* de  $V_{DD}$ ),  $I_{\max M3}$ , determina-se quantos *pads* de  $V_{DD}$  serão necessários no *chip*. Como  $I_{\max M4} \cong I_{\max M3}$ , o número de *pads* de terra é igual ao número de *pads* de  $V_{DD}$ . Ou seja, ao invés de se dimensionar o tamanho, dimensiona-se o número de *pads*; o tamanho dos *pads* é fixo. A corrente em cada *pad* de alimentação é de 100 mA aproximadamente.

O número e a disposição dos *pads* ao redor do *chip* devem ainda ser otimizados para minimizar as quedas I.R de tensão nas resistências dos barramentos de  $V_{DD}$  globais. Para a mínima queda I.R em um barramento de alimentação, o ideal é que ele atravessasse o *chip* de um lado a outro, conectado a *pads* em ambas as extremidades. Se o barramento for ligado ao *pad* somente em uma extremidade, toda a corrente  $I$  entrará pelo mesmo *pad*. Com o barramento ligado a *pads* em ambas as suas extremidades, em cada *pad* entra uma corrente de  $I/2$ ; assim, a densidade de corrente na linha e a queda I.R de um *pad* até o centro do barramento são ambas reduzidas à metade. Isto pode ser visualizado na figura 5.14.

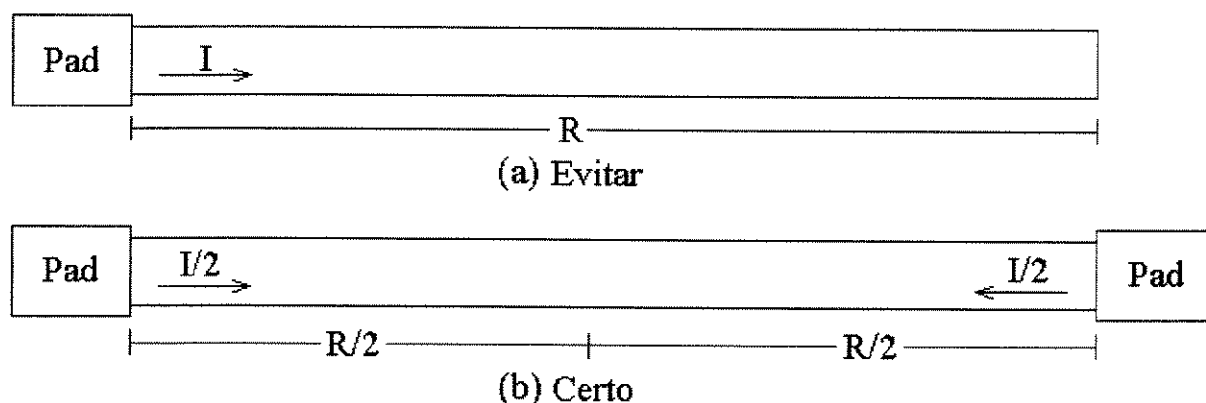


Figura 5.14 – Duas alternativas de barramento de alimentação. Mas as quedas I.R nos barramentos dependem da distribuição de cargas ao longo de seus comprimentos.

As portas lógicas alimentadas por um barramento devem ser distribuídas, preferencialmente, ao longo do seu comprimento, porque assim a corrente total através desse barramento vai diminuindo progressivamente, e a queda de tensão no barramento é distribuída mais uniformemente ao longo do seu comprimento, variando gradualmente com a posição. No arranjo da figura 5.14a, se todas as portas lógicas ficassem concentradas na extremidade direita do barramento, todas sentiriam a mesma queda I.R na tensão  $V_{DD}$ , sendo  $I$  a corrente total. Com as portas distribuídas ao longo do barramento, e este conectado a *pads* em ambas as extremidades, tem-se a melhor topologia de alimentação. No desenho dos barramentos de  $V_{DD}$  globais, o projetista deve dimensionar suas larguras primeiro em função da corrente máxima de alimentação (larguras mínimas) e depois ainda procurar minimizar as suas quedas I.R e indutâncias.

O arranjo dos *pads* e barramentos de alimentação no *chip* afeta diretamente a sua indutância, porque cada disposição corresponde a um modelo diferente de linha de transmissão. O grande desafio é distribuir um  $V_{DD}$  e um terra limpos para todos os blocos do *chip*, pois fontes de alimentação contaminadas ou “sujas” acarretam falhas intermitentes do

circuito. De um modo geral, quanto mais distribuídas e entrelaçadas forem as conexões de alimentação, menor será sua indutância total. Neste projeto, na medida do possível, procurou-se agrupar as linhas de  $V_{DD}$  e terra próximas e paralelas no *layout* dos FFs [11], procurando-se seguir a notação em anel, que diminui as indutâncias de ambas as linhas [78]. Deve-se desenhar os barramentos de  $V_{DD}$  tão largos quanto possível, formando mesmo grandes áreas (“ilhas”) de metal 3 cobrindo o substrato do *chip*, mas com o mínimo de sobreposição com camadas de FETs e sinais. As linhas de  $V_{DD}$  devem tender a formar um plano de metal 3 cobrindo todo o *chip* [89]; assim, tanto quedas I.R quanto indutâncias são reduzidas para  $V_{DD}$ .

#### 5.4.2.4. LVS.

Um sistema de CAD mínimo compreende uma interface de entrada (especificação) do circuito, um simulador lógico ou elétrico, e um editor de *layouts* a nível de máscaras.

O sistema de CAD usado para este projeto não dispunha do recurso de LVS, então esta forma de verificação do *layout* teve que ser feita manualmente, isto é, exibindo-se cada um dos blocos do *layout* na tela do computador, e comparando-se esse bloco do circuito com o seu esquema lógico, ou mesmo com um esquema a nível de transistores. Por exemplo, ajustava-se a janela gráfica do *Magic* para exibir a parte do *layout* correspondente a um dos flip-flops, e comparava-se esse *layout* com o esquema lógico do mesmo flip-flop. Uma vez que já se estava familiarizado com a aparência do *layout* das várias portas NOR criadas, preferiu-se empregar esquemas a nível lógico para cada flip-flop e depois um esquema lógico detalhado para o contador inteiro (figura 4.15). Como os quatro flip-flops do contador são todos diferentes entre si, foi preciso fazer essa verificação para cada um deles, com um esquema lógico próprio para cada um. Essa abordagem para realizar o LVS foi a mais prática e eficiente possível, diante das limitações do sistema de CAD empregado.

As figuras 5.15 a 5.18 mostram os esquemas lógicos dos quatro FFs que compõem o contador 7490-like. Para facilitar grandemente a comparação desses esquemas com seus *layouts* correspondentes, cada um deles teve que ser desenhado com a mesma orientação horizontal e vertical do seu *layout*. Se eles não fossem desenhados assim, o trabalho iria ficar muito cansativo, demorado e sujeito a erros, porque seria preciso “girar” (reverter) mentalmente uma porta que, no esquema, aparece na fileira de baixo, mas no *layout* aparece na fileira de cima. Ou seria preciso “virar” cada porta em um eixo vertical se, no esquema, a propagação do sinal fosse da esquerda para a direita (como é usual) mas, no *layout*, essa propagação fosse no sentido contrário, por estar o *layout* do flip-flop orientado da direita para a esquerda. Nas figuras 5.15 a 5.18, cada flip-flop já está orientado adequadamente de acordo como está desenhado no *layout*.

Ao lado de cada porta lógica, está indicado o nome da célula, conforme definido no Capítulo 4, e o nome da instância dessa célula no FF em que ela se encontra. O nome da instância é hierárquico e segue o padrão SPICE. Por exemplo, o FF3 é instanciado na netlist com o nome X3; a porta NOR “mast2.2” do FF3, instanciada dentro do subcircuito dele como X4, é referenciada globalmente como X3.X4.

O esquemático do FF1 da figura 5.10, a nível de transistores e na notação em anel, não se mostrou tão adequado para a realização do LVS quanto o esquema lógico da figura 5.15.

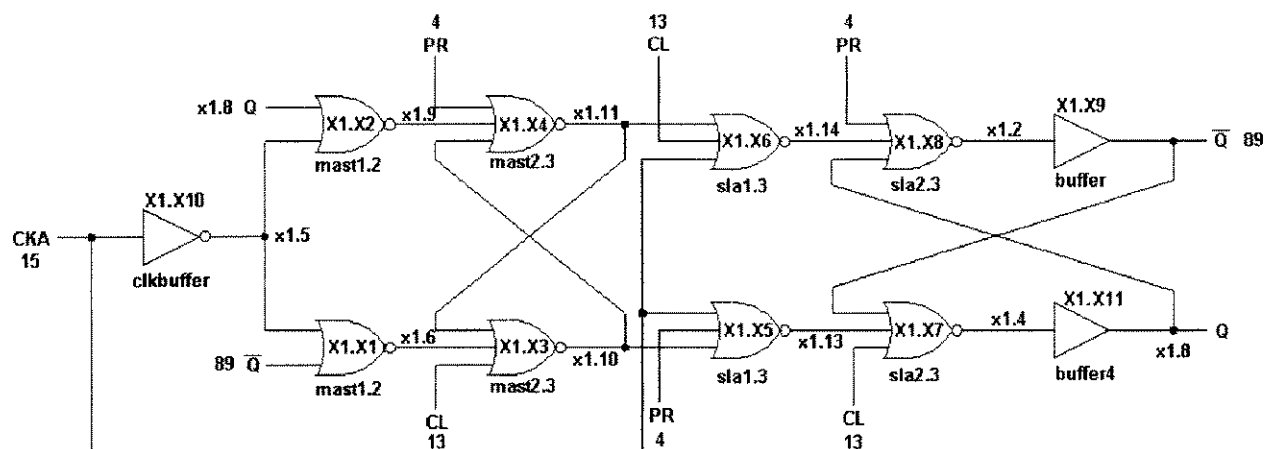


Figura 5.15 – Esquema lógico do FF1.

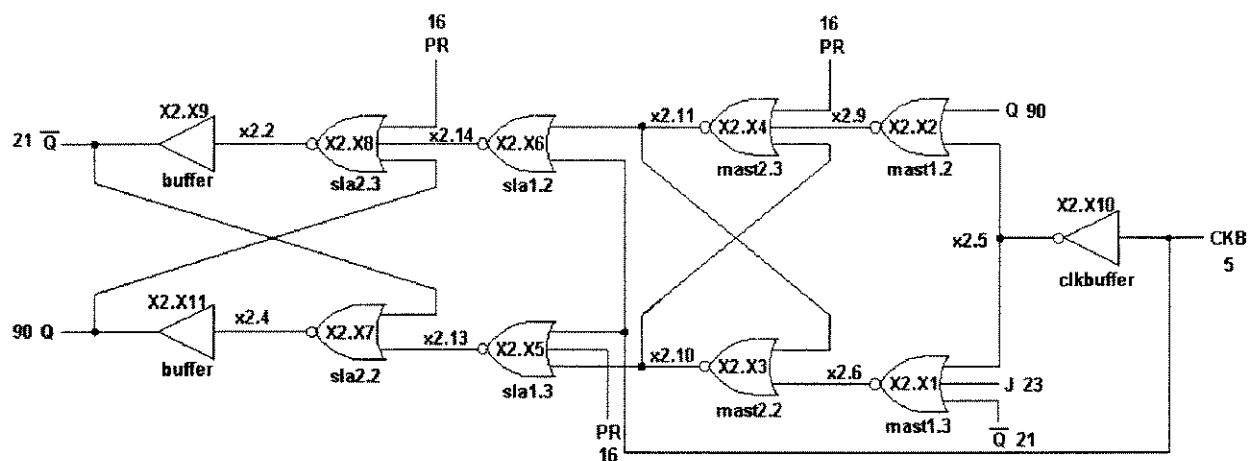


Figura 5.16 – Esquema lógico do FF2.

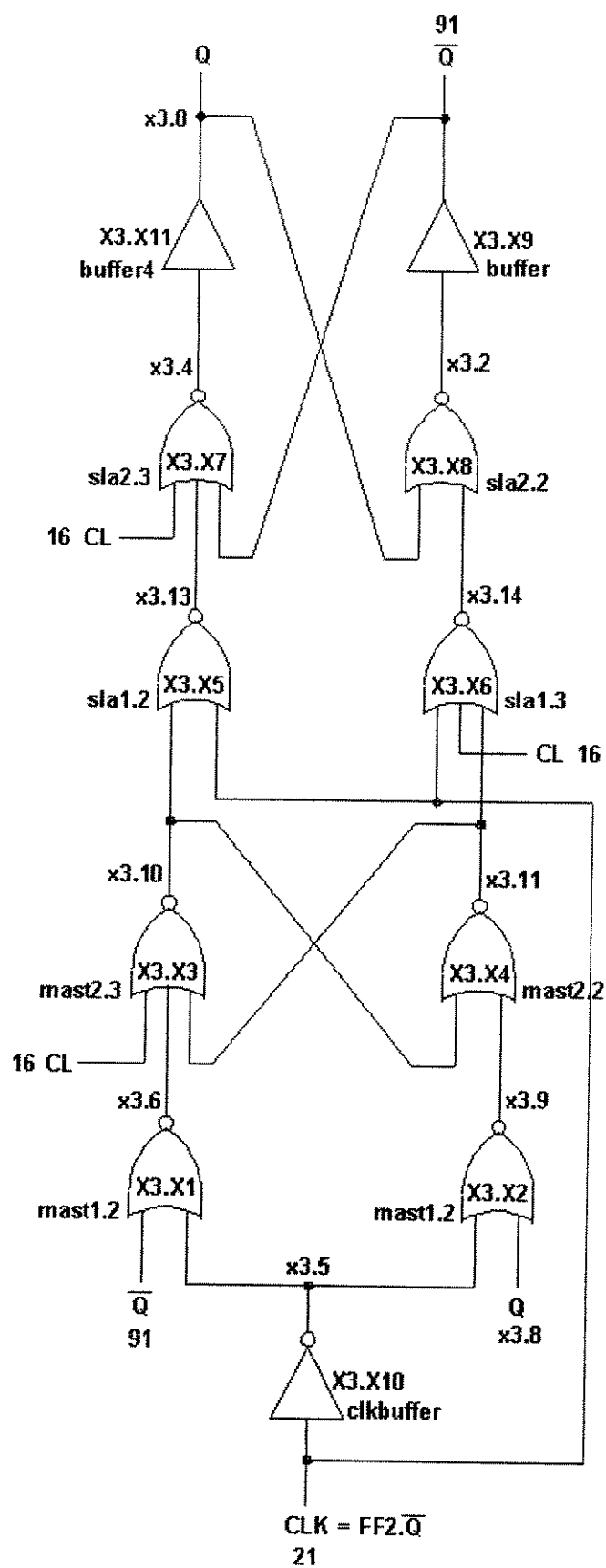


Figura 5.17 – Esquema lógico do FF3.



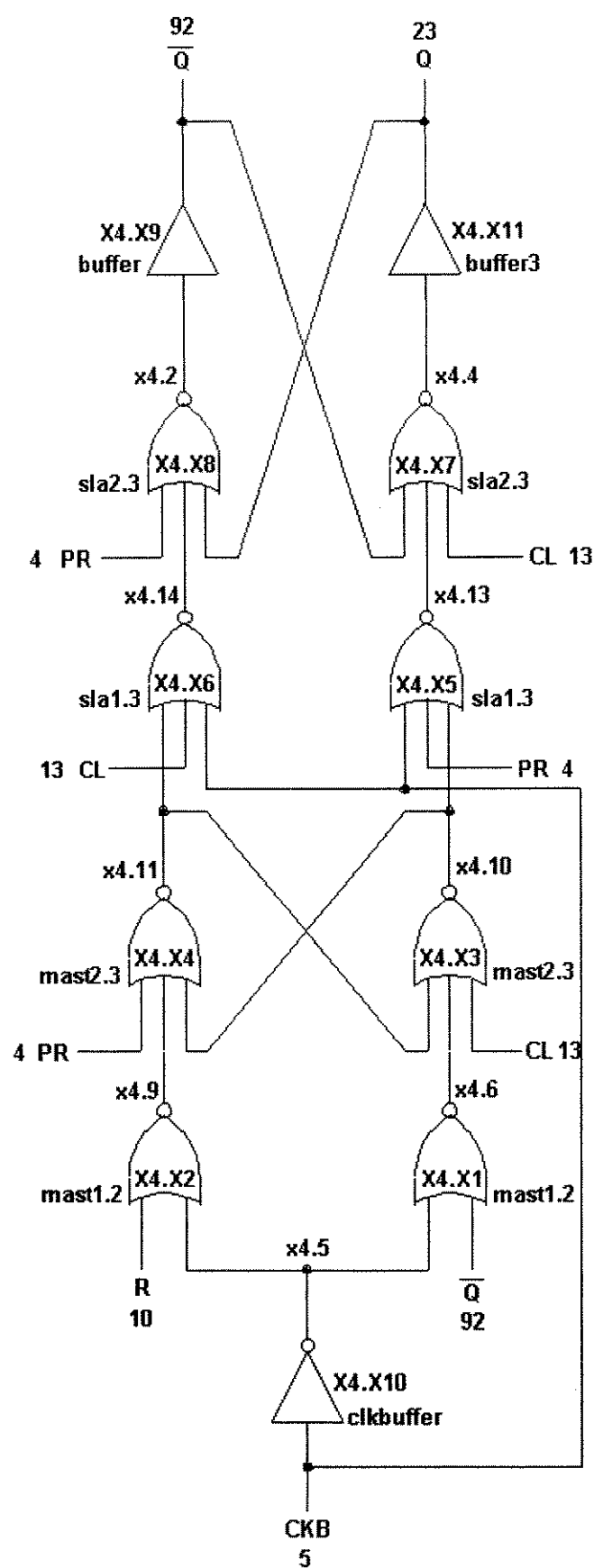


Figura 5.18 – Esquema lógico do FF4.

#### 5.4.3. Extração do Circuito e de Parasitas.

A extração do circuito é a obtenção, a partir das geometrias presentes no *layout*, de uma *netlist* dos dispositivos desenhados e de suas interconexões; as dimensões dos FETs e diodos são estimadas, e os efeitos parasitas são considerados pela inclusão das resistências e capacitâncias associadas às geometrias. Para possibilitar isso, as características elétricas de cada tipo de camada do HGaAs-III estão especificadas na seção *extract* do arquivo “cmp.tech26”. O processo de extração culmina com a obtenção de um circuito sem hierarquia (planificado), isto é, com todas as suas células instanciadas.

A extração de parâmetros do *layout* é uma etapa importantíssima do projeto de um *chip*, indispensável para o sucesso de um projeto. Pela extração de parâmetros, o projetista vai determinar o efeito das resistências e capacitâncias parasitas do *layout* sobre o desempenho do circuito, e portanto vai poder simular um circuito muito mais próximo da realidade, muito mais “físico”. Além disso, a extração de parâmetros pode indicar erros de *layout* (na falta de um LVS), como uma ligação entre nós do circuito que não deveria existir, ou a falta de uma conexão entre dois nós que deveriam ser um mesmo nó (deveriam estar conectados).

O *Magic* gera arquivos de *layout* com extensão **.mag**. Quando o circuito é extraído, é gerado um arquivo de extração para cada célula de todos os níveis da hierarquia, com a extensão **.ext**. Isso significa que, para cada arquivo **.mag** de *layout*, é gerado um arquivo **.ext** correspondente. Finalmente, o circuito hierárquico descrito nos vários arquivos **.ext** deve ser totalmente planificado para a simulação; essa tarefa é feita fora do ambiente do *Magic*, por um programa de conversão entre os dois formatos, chamado *ext2sim*. Depois de terminada a extração hierárquica dentro do *Magic*, o projetista deve executar o programa *ext2sim* sobre o arquivo **.ext** da célula-raiz da hierarquia, e todos os elementos do circuito serão instanciados em um arquivo **.sim**, para a simulação. As opções de linha de comando do programa *ext2sim* são detalhadas na página *man* do Unix sobre esse programa, e também em [11] e [87]. Os três tipos de arquivo (**.mag**, **.ext** e **.sim**) estão no formato ASCII, sendo editáveis pelo usuário.

Os arquivos **.ext** e **.sim** gerados têm um formato próprio de organizar as informações sobre o circuito e têm que ser interpretados pelo usuário. Todas as informações sobre os formatos dos arquivos **.ext** e **.sim** e sobre o programa *ext2sim* podem ser obtidas a partir das suas páginas *man*, no sistema operacional Unix. É fortemente recomendado imprimir estas páginas, para poder consultá-las durante a análise do circuito extraído.

Existe um arquivo **.ext** para cada célula e subcélula de um *layout* hierárquico, mas só existirá um único arquivo planificado **.sim** para o circuito inteiro. Se os nomes dos nós do circuito forem os mesmos tanto no *layout* quanto na *netlist* do HSPICE, então os valores das resistências e capacitâncias parasitas extraídos do circuito, e listados no arquivo **.sim**, já podem ser copiados diretamente para dentro da *netlist* do HSPICE, e usados em novas simulações. Neste projeto, a retro-anotação foi feita dessa maneira. Outra alternativa seria usar o programa *sim2spice*, que converte um arquivo no formato **.sim** para um novo arquivo, no formato de uma *netlist* do SPICE. A esta *netlist* de FETs, diodos e capacitores, o projetista tem que acrescentar os comandos de simulação para o SPICE.

Os valores de capacitância da seção *extract* do arquivo “cmp.tech26” estão todos expressos em décimos de attofarad por  $\mu\text{m}^2$ , no caso das capacitâncias de placas paralelas de *overlap*, e em centésimos de attofarad por  $\mu\text{m}$ , no caso das capacitâncias de frangeamento por

unidade de comprimento da interconexão. Portanto, é preciso cuidado para que as capacitâncias extraídas pelo *Magic* sejam corretamente interpretadas.

Após a conclusão do desenho do *layout*, com as dimensões dos FETs usadas na simulação final, os valores das capacitâncias parasitas foram extraídos do *layout* e retro-annotados para a *netlist* do HSPICE, permitindo executar-se simulações pós-*layout*. Essas simulações, mais realistas, fazem uma verificação mais confiável do desempenho real do CI. Assim fechou-se o ciclo de projeto.

Para este CI, as simulações pós-*layout* ainda indicaram um desempenho satisfatório do circuito, sendo que a operação em 1 GHz foi mantida. O Apêndice 3 mostra as formas de onda das configurações de divisão de frequência do circuito final, que estão satisfatórias. Porém, o consumo de potência e a área poderiam ser reduzidos por meio de alterações no *layout* das células de portas lógicas. Mais especificamente, percebeu-se que as portas internas dos FFs poderiam ser muito menores (menor  $W_g$  dos FETs). Infelizmente, devido a contratempos como o erro no arquivo de tecnologia, não houve mais tempo para realimentar-se os resultados das simulações pós-*layout* para o *layout* do 7490-like e fazer-se as devidas modificações, porque o prazo-limite para a rodada da *foundry* já estava quase expirando, e essa seria a última rodada na tecnologia HGaAs-III. Portanto, o circuito final não ficou otimizado.

A figura 5.19 mostra o *layout* final do *chip* 7490-like projetado; sua área total compreende a área do circuito propriamente dito, mais o acréscimo de área devido aos *pads*. Concluída a versão final do *layout*, mediu-se essa área total pelo *Magic*, e calculou-se o seu custo total associado (fabricação + encapsulamento), que tiveram os seguintes valores:

**Área aproximada da *die*:**  $2,616 \times 2,805 \text{ mm} = 7,338 \text{ mm}^2$

**Custo total:**

Fabricação:  $(2700 \text{ FF}) \times (7,338 \text{ mm}^2) = 19812,6 \text{ FF}$

Encapsulamento: LDCC 28 pinos = 270 FF

**Total: 20082,6 FF ~ US\$ 3462,5**

O projeto foi então avaliado por uma comissão da FAPESP, que o aprovou para ser fabricado no âmbito do CMP, com o nome oficial de FAPESP48\_CCS. No *Magic*, gerou-se o arquivo de descrição do *layout* do circuito, no formato CIF, o qual foi anexado a um *e-mail* e enviado ao CMP, para fabricação na última rodada do processo HGaAs-III da Vitesse, marcada para 20 de outubro de 1997. Depois do envio do arquivo de *layout*, caso este ainda possuía algum erro, isto é informado via *e-mail* pelo coordenador de projetos no CMP, que então solicita ao projetista que corrija o erro e envie um novo *layout*.

A fabricação do chip FAPESP48\_CCS pelo CMP está documentada no Relatório Anual dessa entidade referente a 1997, o *French MPC Activity Report – 1997*. No Apêndice 5, estão anexadas fotocópias da capa do relatório e da página em que está documentada a fabricação do CI, pela rodada G97-3, incluindo um resumo da função do circuito. Existe uma discrepância entre a área avaliada pelo *Magic* (acima) e a área fornecida pelo CMP, sendo que esta é maior. Essa diferença deve-se provavelmente às linhas de corte (*sawing lanes*) entre as *dies* na *wafer*; com a área oficial, o novo custo total do CI sobe para US\$ 4026 aproximadamente.

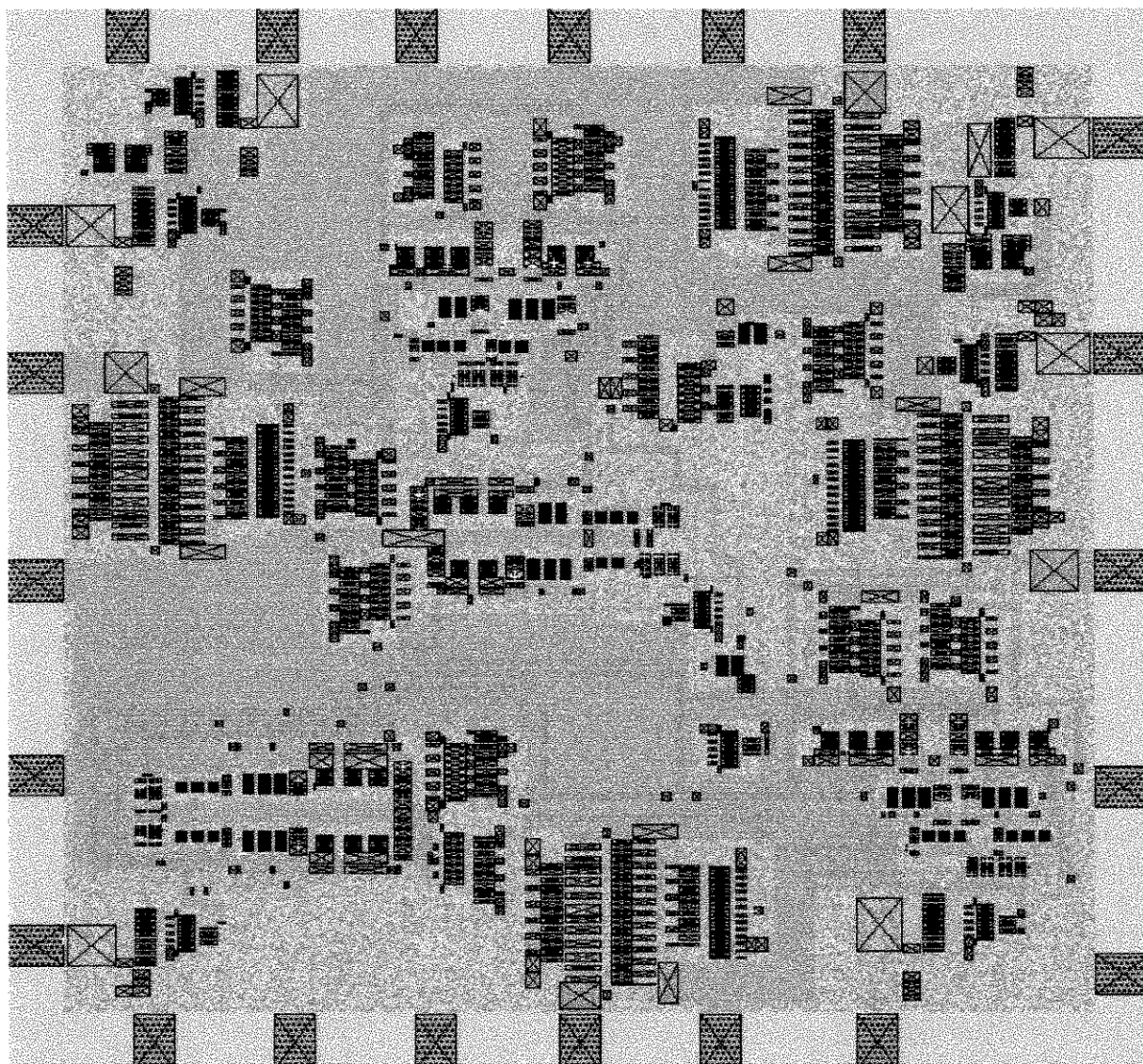


Figura 5.19 – *Layout* final do *chip* 7490-like enviado ao CMP para fabricação.

## 5.5. Encapsulamento e Diagrama de Pinagem.

O tipo de encapsulamento escolhido para alojar a *die* foi o LDCC (*Leaded Ceramic Chip Carrier*) de 28 pinos, por causa de sua boa dissipação de potência combinada com sua capacidade de alta frequência (3,9 GHz). Este encapsulamento aceita múltiplas conexões de  $V_{DD}$  e terra. O esquemático e a pinagem do LDCC-28 são dados na página 6-4 de [12]; a área da cavidade do *chip* (área máxima de *die*) é de  $3,3 \times 3,3$  mm.

Enviou-se por fax o diagrama de pinagem (*bonding diagram*) do CI para o tipo de encapsulamento escolhido, mostrado na figura 5.20. Esse diagrama é requisitado pelo CMP. As ligações dos *pads* de  $V_{TT}$  com a *die* são feitas por metal 4. As ligações dos *pads* de GND e dos *pads* de saída (QA, QB, QC e QD) com a *die* são feitas por metal 3. Finalmente, as ligações dos *pads* dos sinais de entrada (CKA, CKB, R9 e R0) são feitas por metal 2.

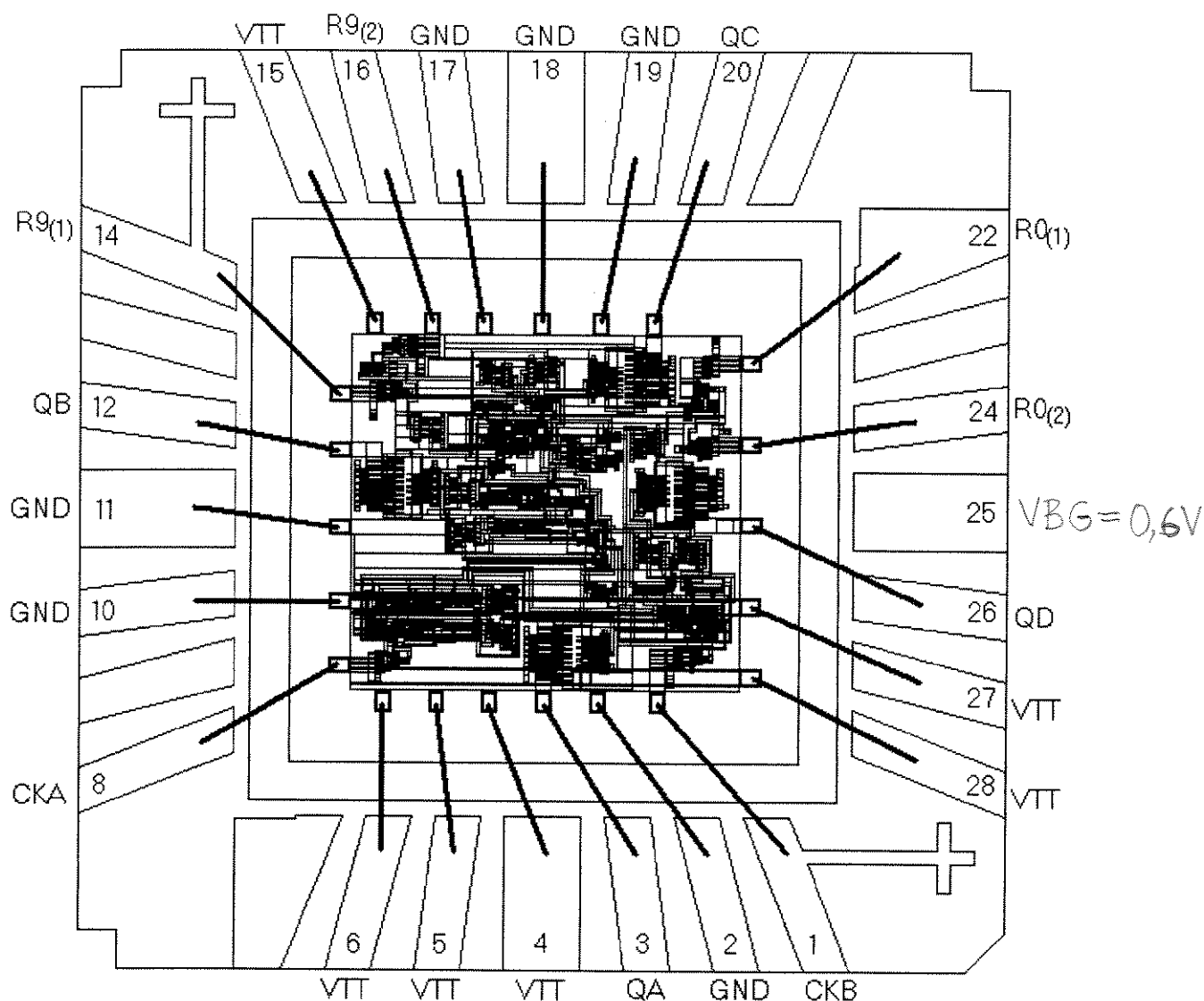


Figura 5.20 - Chip FAPESP48\_CCS.

## 5.6. Recomendações para o Teste do CI.

Até a data da conclusão desta Dissertação, os protótipos do CI 7490-like não haviam sido recebidos no CCS para serem testados. O teste e as medidas deverão ser feitos no segundo semestre deste ano.

Deve ser projetado e construído o *jig* de testes, o que deverá ser feito pelo CCS. O substrato é tipicamente uma capa de teflon parafusada sobre base de alumínio; esta é um cilindro de pequena altura cuja seção transversal deve ser um polígono com um número de lados igual ao número de conexões para o CI. Em cada lado é parafusado um conector BNC, ligado até um pino do CI por uma linha impressa no substrato. O CI deve ficar preso com epoxy a uma cavidade no centro da base de alumínio, que também serve de dissipador de calor. Capacitores de desacoplamento, conectados aos pinos de alimentação do CI, filtram (suavizam) variações da tensão da fonte causadas por variações da corrente do CI.

Deve-se planejar e depois descrever, com detalhes, todo o *setup* de medidas usado para a caracterização do CI: os dados dos equipamentos empregados nas medidas (marca, modelo, e características principais), as conexões feitas e como foram feitas, e as características dos sinais aplicados (amplitude, frequência, etc.). Devem ficar registrados os procedimentos das medidas e a sequência dos testes, assim como os resultados e formas de onda para as diversas configurações de uso do CI.

As medidas elétricas no *chip* poderão ser realizadas no DSIF / FEEC, consistindo de testes preliminares DC e funcional a baixa frequência, mais os testes de RF e operação até a frequência de 1 GHz. No DSIF também podem ser medidos chips não-encapsulados (*die probing*), por meio de uma estação de prova (*probe station*).

Precisam ser registradas as formas de onda de saída do CI para a frequência máxima de clock na entrada. A frequência máxima de clock é tal, que para uma frequência acima dela resultam formas de onda de saída muito deterioradas ou saídas errôneas; para este CI, ela é definida como a frequência de entrada em que a saída correta ainda é obtida com uma  $V_{pp} > 800\text{ mV}$ . Para diferentes razões de divisão pode haver diferentes frequências máximas de operação, como no 7490 TTL (a frequência máxima da divisão por 3 é menor; algumas divisões podem chegar a frequências de clock maiores do que outras).

A tabela 5.1 indica quais as ligações que devem ser feitas entre os pinos do CI para a sua operação em cada uma das configurações de divisão por N.

| N            | Entrada do sinal | Saída do sinal        | Ligações necessárias entre terminais                                                                                         |
|--------------|------------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------|
| 2            | 8 (CKA)          | 3 (QA)                | Terminal 22 (ou 24) ao $V_{TT}$ ;<br>Terminal 14 (ou 16) ao $V_{TT}$ .                                                       |
| 3            | 1 (CKB)          | 20 (QC)               | Terminal 14 (ou 16) ao $V_{TT}$ ;<br>Terminal 20 ao terminal 22;<br>Terminal 12 ao terminal 24.                              |
| 3<br>(outra) | 1 (CKB)          | 12 (QB)               | Terminal 22 (ou 24) ao $V_{TT}$ ;<br>Terminal 1 ao terminal 14;<br>Terminal 20 ao terminal 16.                               |
| 4            | 1 (CKB)          | 20 (QC)               | Terminal 14 (ou 16) ao $V_{TT}$ ;<br>Terminais 22 e 24 ao terminal 26.                                                       |
| 5            | 1 (CKB)          | 20 (QC) ou<br>26 (QD) | Terminal 22 (ou 24) ao $V_{TT}$ ;<br>Terminal 14 (ou 16) ao $V_{TT}$ .                                                       |
| 6            | 8 (CKA)          | 20 (QC)               | Terminal 14 (ou 16) ao $V_{TT}$ ;<br>Terminal 1 ao terminal 3;<br>Terminal 20 ao terminal 24;<br>Terminal 12 ao terminal 22. |
| 7            | 8 (CKA)          | 20 (QC) ou<br>26 (QD) | Terminal 22 (ou 24) ao $V_{TT}$ ;<br>Terminal 1 ao terminal 3;<br>Terminal 12 ao terminal 14;<br>Terminal 20 ao terminal 16. |
| 8            | 8 (CKA)          | 20 (QC)               | Terminal 14 (ou 16) ao $V_{TT}$ ;<br>Terminal 26 aos terminais 22 e 24;<br>Terminal 1 ao terminal 3.                         |
| 9            | 8 (CKA)          | 20 (QC) ou<br>26 (QD) | Terminal 14 (ou 16) ao $V_{TT}$ ;<br>Terminal 3 aos terminais 24 e 1;<br>Terminal 26 ao terminal 22.                         |

|                         |         |                                        |                                                                                                      |
|-------------------------|---------|----------------------------------------|------------------------------------------------------------------------------------------------------|
| 10<br>(contagem<br>BCD) | 8 (CKA) | 3, 12, 20 e 26<br>(QA, QB, QC e<br>QD) | Terminal 22 (ou 24) ao $V_{TT}$ ;<br>Terminal 14 (ou 16) ao $V_{TT}$ ;<br>Terminal 1 ao terminal 3   |
| 10<br>simétrico         | 1 (CKB) | 3 (QA)                                 | Terminal 22 (ou 24) ao $V_{TT}$ ;<br>Terminal 14 (ou 16) ao $V_{TT}$ ;<br>Terminal 8 ao terminal 26. |

Tabela 5.1 – Configurações de operação do CI 7490-like. Presume-se o uso de alimentação negativa  $V_{TT}$ .

Os pinos de entrada não utilizados devem ser ligados no terra (para alimentação com  $V_{DD} = 2\text{ V}$ ) ou em  $V_{TT}$  (para alimentação com  $V_{TT} = -2\text{ V}$ ), para reduzir-se o ruído. A figura 5.21 mostra um exemplo de *setup* de medidas para o 7490-like, utilizando equipamentos disponíveis no DSIF.

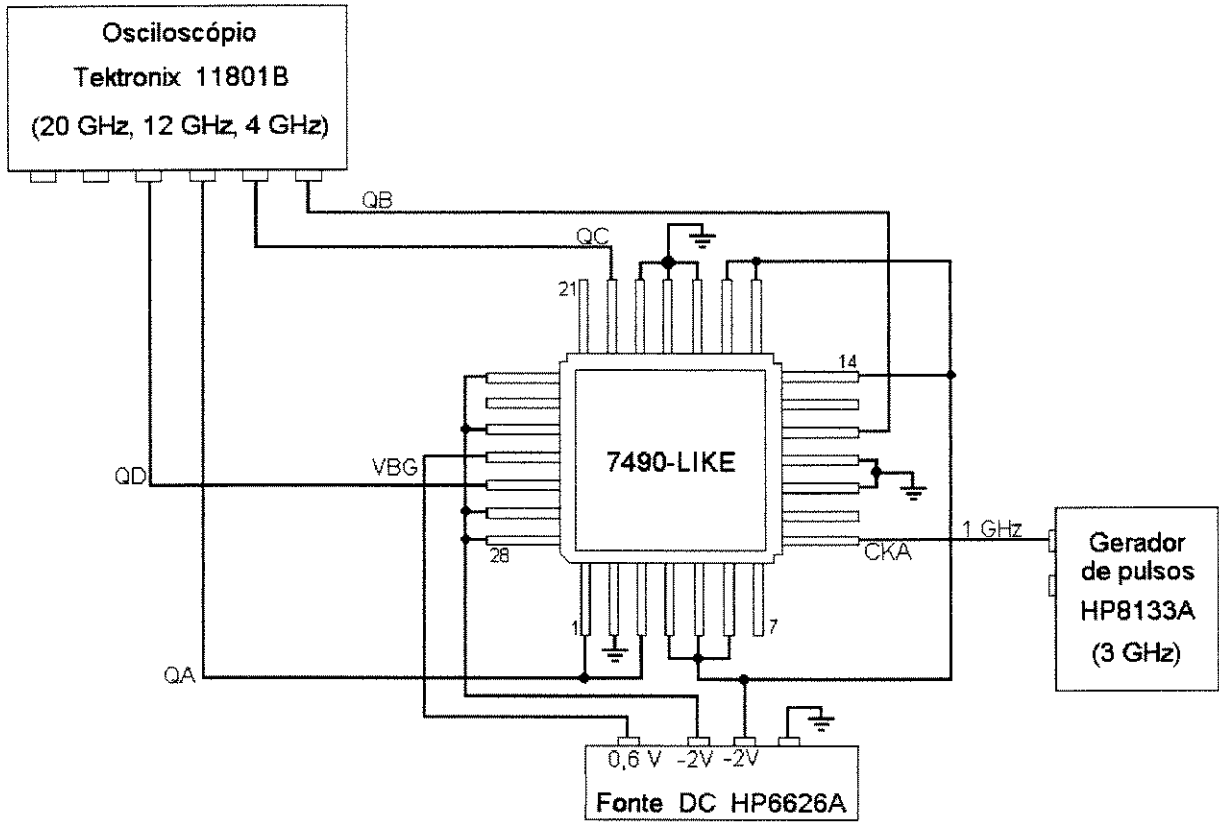


Figura 5.21 – *Setup* de teste para a configuração do CI como contador de década.

Finalmente, deve ser feita a comparação entre o desempenho medido e o simulado, com a justificativa das diferenças entre os valores medidos dos parâmetros elétricos. Devem ser montadas duas tabelas: uma mostrando o desempenho **simulado** do circuito 7490-like, isto é, as características de desempenho previstas nas simulações em HSPICE, e outra tabela com o desempenho **medido** do CI, e explicadas as diferenças entre os valores nas duas tabelas. A mesma comparação pode ser feita somente para o flip-flop FF1, que pode ser acessado isoladamente do restante do circuito.

Diversas informações sobre o teste de CIs digitais de GaAs são dadas no Capítulo 6 de [11]. A referência [70] deverá ser um importante guia para o teste do CI 7490-like, pois ela explica detalhadamente todos os procedimentos para o teste de um CI digital de GaAs compatível com ECL. Conceitos de teste de CIs digitais de alta frequência, de um modo geral,

são dados em [90]. A referência [91] descreve os procedimentos de medida para cada um dos parâmetros DC e AC de um CI, como os níveis lógicos, correntes de entrada e saída, consumo de potência, tempos de *setup* e *hold* e atrasos de propagação.

As mesmas regras de *layout* e projeto de sistemas que são aplicáveis para ECL, a nível de placa de circuito impresso, também são aplicáveis a CIs de GaAs, com um pouco mais de rigor. Portanto, para o projeto, construção e teste de uma circuito com CI de GaAs, é de grande ajuda ter alguma experiência em projetos com ECL. Por isso, os guias de projeto dos principais fabricantes de tecnologia ECL, [1] e [92], devem ser estudados para o teste do CI de GaAs. Finalmente, o *databook* da GigaBit Logic [64] contém um excelente guia de projetos, com várias *application notes* que cobrem todos os aspectos do teste e operação de CIs de GaAs em sistemas.



## CAPÍTULO 6.

### Conclusões e Sugestões para Futuros Trabalhos.

A lógica DCFL representa o melhor compromisso entre velocidade, área e consumo de potência em GaAs [77], no sentido de que uma função lógica pode ser realizada com bem poucos FETs, e de que, como  $V_{pp}$  é baixa (0,6 ~ 0,7 V), a tensão de alimentação usada pode ser baixa também, reduzindo o consumo de potência. A simplicidade topológica e de projeto das portas DCFL fazem desta família uma candidata atraente para VLSI. Entretanto, suas NMs são baixas em comparação com outros tipos de lógica, e ela é muito sensível a variações de  $V_t$  no processo de fabricação.

Neste trabalho, uma versão do CI TTL 7490 foi projetada e construída na família DCFL, com um desempenho de velocidade muito superior, mas mantendo a mesma arquitetura e funcionalidade lógica do 7490, sendo por isso chamada de *7490-like*. Trata-se de um CI contador de década e divisor de frequências com módulo N inteiro e configurável, na faixa de 2 a 10, unicamente através de conexões **diretas** entre pinos do CI. Foi descrito o projeto *full-custom* desse CI, desde a sua funcionalidade lógica até a conclusão da versão final do seu *layout*.

As razões para a escolha da arquitetura do 7490 para este projeto foram:

- Sua versatilidade de configuração;
- Sua universalidade e uso muito difundido, pois trata-se de uma *commodity* comercial bem conhecida;
- O nível de complexidade dessa arquitetura é mediano, não sendo nem muito simples nem excessivamente complexo, oferecendo uma complexidade adequada para um primeiro projeto em GaAs digital.

As aplicações do CI *7490-like* são em síntese/divisão de frequências, contagem, instrumentação de alta frequência e na composição de circuitos digitais de alta velocidade, podendo-se usá-lo como um prescaler na entrada de outros blocos. Em muitas aplicações lógicas, uma frequência precisa ser dividida por um número inteiro. Um dos casos mais típicos é a sincronização de um relógio digital de parede com a frequência de 60 Hz da rede elétrica, a qual deve ser reduzida a 1 Hz mediante divisão. Outro caso típico em que se necessita da divisão de frequências é no circuito da base de tempo usada em contadores digitais. Geralmente parte-se de um oscilador de quartzo, muito estável, que gera uma frequência da ordem de alguns MHz, a qual deve ser reduzida, mediante divisões sucessivas, a 1000 Hz, 100 Hz, 10 Hz ou mesmo 1 Hz.

Todas as configurações funcionais do circuito *7490-like* foram simuladas com sucesso no HSPICE, e os resultados mostram a sua operação como um contador de década em taxas de *clock* de até 1 GHz, acionando linhas de 50  $\Omega$  com capacitâncias de carga de 5 pF nas suas saídas. A tensão de alimentação pode ser de 2 V ou -2 V; com esta alimentação negativa, os níveis de entrada e saída do CI são compatíveis com os níveis da lógica ECL. O circuito foi otimizado para a operação em altas velocidades, em detrimento de um baixo consumo de potência.

O *layout* do *chip* 7490-like foi desenhado no *software* de CAD Magic, para ser implementado no processo de MESFETs E/D HGaAs-III da *foundry* norte-americana Vitesse, com  $L_g = 0,6 \mu\text{m}$ . A topologia de *layout* foi escolhida para minimizar o acoplamento entre as linhas de sinal de alta frequência e os barramentos de  $V_{DD}$  ou  $V_{TT}$ .

O projeto foi submetido à apreciação do Projeto Multi-Usuário brasileiro (PMU/FAPESP), tendo sido aprovado para fabricação na Vitesse, por intermédio do PMU francês (CMP). O arquivo de *layout* foi então enviado ao CMP para a implementação do CI 7490-like na última rodada da tecnologia HGaAs-III da Vitesse. A área total do *chip* foi de  $8,69 \text{ mm}^2$ . O CI 7490-like foi implementado, porém os seus protótipos demoraram demais para ser entregues à Unicamp, e por isso não foi possível testá-los dentro do prazo de conclusão desta Dissertação.

Este projeto foi concebido principalmente para fins educacionais, de formação de recursos humanos em projeto de CIs, e esta Dissertação pretende ser também um guia para futuros projetistas, apresentando um roteiro bastante **didático** das etapas de um trabalho de projeto; por isso o CI escolhido foi um MSI com um nível de complexidade intermediário. Este projeto também possui um grau razoável de ineditismo. Finalmente, este trabalho também demonstrou muitos aspectos úteis e instrutivos do projeto de CIs digitais de um modo geral, e mais especificamente, do projeto de um CI digital em DCFL. Alguns desses aspectos são listados a seguir:

1 - A conversão, exigida pela tecnologia, de um FF totalmente baseado em funções NAND para uma lógica baseada em funções NOR, e a topologia do FF JK com portas NOR.

2 - Diversos compromissos (*trade-offs*) de dimensionamento de portas básicas DCFL.

3 - Determinação da razão  $\beta$  ótima ( $\beta = 5$ ) para a parte inversora de uma porta SDCFL, que não é sugerida na literatura consultada.

4 - Formação de uma “biblioteca” de *netlists* para simulação de portas e FFs no HSPICE, que está disponibilizada nesta Dissertação. Esta biblioteca contribui principalmente com a apresentação de técnicas de medição de parâmetros em HSPICE ( $t_{PD}$ ,  $t_{HL}$ ,  $t_{LH}$ ,  $V_{pp}$ , ...).

5 - Estratégias de simulação e de projeto:

5.1 - A metodologia de projeto *bottom-up*, com três níveis de hierarquia, foi explorada.

5.2 - A abordagem heurística: quando houver dúvida para escolher entre duas ou mais alternativas de projeto, deve-se simular e comparar, para determinar qual dos projetos melhor atende às especificações.

5.3 - O método de escalonamento progressivo (*tapered design*) para o dimensionamento de circuitos com cascatas de portas, com os FFs. Um FF MS nunca deve ser projetado com as dimensões das portas do *latch* mestre iguais às dimensões das portas correspondentes do *latch* escravo. Isso não aumenta a velocidade do FF, e somente desperdiça área.

5.4 - Não devem ser empregadas portas DCFL “bufferizadas” (SDCFL) para toda a lógica de um circuito, é uma medida exagerada e resulta em uma área e um consumo de potência muito grandes. O uso de SDCFL deve ser restrito aos casos absolutamente necessários.

6 - A configuração do 7490 para a divisão de frequências por 3, que não foi encontrada na literatura técnica consultada.

Este trabalho já foi parcialmente publicado na XII Conferência da Sociedade Brasileira de Microeletrônica, realizada em 1997, com o título “Design of a 7490-like Decade Counter Integrated Circuit, Using GaAs MESFET DCFL Family, for Frequencies Up to 1 GHz” [22]. Os apoios financeiros foram concedidos pelas agências PICDT/CAPES (bolsa de Mestrado), PMU/FAPESP (custo de fabricação e encapsulamento do CI), e FINEP.

Como uma sugestão para próximos trabalhos na área de CIs digitais de GaAs, recomenda-se a realização de blocos mais complexos que possam aproveitar os conhecimentos e experiência deste trabalho. Do leque de aplicações para CIs digitais de GaAs listado na seção 1.4.1., aquelas voltadas para redes de Telecomunicações optoeletrônicas são as mais promissoras, como circuitos de recuperação de clock e *laser drivers*. Pode-se considerar também blocos de processamento aritmético, como divisores, ULAs e circuitos de ponto flutuante. Uma aplicação mais desafiadora seria um conversor A/D ou D/A, o que envolveria o projeto de blocos analógicos e digitais.

Outra sugestão, dando continuidade a este trabalho, é a elaboração do *data-sheet* completo do CI 7490-like, nos moldes padrão JEDEC de *data-sheets* de circuitos integrados, a partir das medidas em bancada do *chip*.

Uma sugestão promissora para o aperfeiçoamento de circuitos sequenciais é abandonar a topologia tradicional de flip-flop com portas NOR, e passar a utilizar flip-flops de célula de memória, que foram introduzidos e explicados nas referências [54] e [57]. A topologia tradicional de FFs como a da figura 4.4, não é, inerentemente, um circuito de alta velocidade, pois necessita de pelo menos quatro portas NOR no caminho do sinal, o que acarreta um maior tempo de atraso aos sinais.

Como última sugestão para um trabalho na mesma área, propõe-se a criação de um ambiente real **de projeto** de CIs digitais de GaAs. O HSPICE é uma ferramenta poderosa de análise, mas não auxilia diretamente no projeto. Seria certamente de grande valia que as equações que determinam os parâmetros de desempenho em função dos parâmetros de tecnologia e de projeto fossem inseridas em um software de processamento numérico e simbólico, como o Mathematica por exemplo. Assim, a partir das especificações iniciais, esse sistema de expressões analíticas (que são precisas mas complicadas para resolução manual) seria resolvido fornecendo os parâmetros de projeto necessários. As referências [6] e [66] apresentam as equações e análises mais rigorosas dentre a bibliografia consultada neste trabalho, e servem muito bem para a finalidade desta proposta de trabalho.

## APÊNDICE 1.

### Roteiro para a Inclusão de Gráficos de Simulações do HSPICE em um Documento do Word for Windows 7.0.

As simulações foram feitas por meio do simulador HSPICE, e seus resultados gráficos (curvas de transferência DC, formas de onda no tempo ou correntes nos FETs) foram visualizados no *software* pós-processador GSI (*Graphical Simulation Interface*). Este programa lê certos arquivos gerados por simulações do HSPICE e os exibe graficamente, permitindo que depois sejam impressos em um outro formato de arquivo, para que sejam finalmente anexados a um documento do Word. Esse recurso foi extensivamente usado neste trabalho, e a seguir é apresentado um roteiro para a inclusão de gráficos de simulações, gerados pelo GSI, em um arquivo **.doc** do Word.

A simulação de circuitos no HSPICE pode gerar arquivos de saída com os resultados em forma gráfica, dependendo das diretivas usadas no arquivo de entrada **arq.sp**. Assim, simulações transientes geram arquivos **arq.trn**, e simulações de varredura DC geram arquivos **arq.swn**, onde n é um índice que inicia em 0 (zero) e é incrementado de uma unidade para cada subprograma **.ALTER** presente em **arq.sp**.

Os arquivos **arq.trn** e **arq.swn** contém informações gráficas e somente podem ser visualizados no GSI. Dentro do ambiente do GSI, pode-se abrir múltiplos arquivos ao mesmo tempo, exibindo várias curvas no mesmo gráfico e vários gráficos na mesma tela, e configurar a tela com o título, as cores dessas curvas, o número de gráficos, etc. Em seguida toda a tela pode ser impressa para um arquivo gráfico no formato *postscript*, **arq.ps**, para que os gráficos possam ser vistos por outros programas.

O arquivo no formato **.ps** pode ser **importado** para o *software* CorelDraw. Neste *software*, pode-se configurar a página em *portrait* ou *landscape*, e a figura pode ser girada e/ou ampliada como for mais adequado. Em seguida a figura deve ser **exportada** (e não salva) pelo CorelDraw no formato **arq.bmp**, que já pode ser aberto no Word. Para tanto, basta executar-se, já dentro do Word, o comando Inserir → Figura para abrir o arquivo gráfico no formato **.bmp** dentro de um documento **.doc**.

Caso seja desejado, a figura pode ser aberta primeiro no Paint do Windows 95 para se fazer o seu acabamento final neste aplicativo, e depois ela pode ser copiada e colada do Paint para o Word. É bem mais fácil e rápido editar a figura no Paint do que no CorelDraw, por ser este programa muito complicado e de utilização mais difícil.

O HSPICE também imprime os resultados numéricos de simulações, incluindo todos os resultados de comandos **.measure**, no arquivo **arq.lis**, o que é muito útil para análises mais quantitativas. As tensões e correntes em vários instantes de simulação podem ser impressas com o comando **.op**, ou as tensões e correntes somente em certos elementos podem ser incluídas no arquivo **arq.lis** com o comando **.print**.

# Arquivos Netlist de Entrada para Simulação no HSPICE.

## Programa 1: “curvas.sp”.

Este programa gera as curvas características para a plotagem da curva de transferência do inversor DCFL.

```
MESFET Characteristic Curves
* Para gerar as curvas características, VGS eh um parametro
* variavel com .ALTER. Para cada VGS, eh feito um DC sweep de VDS,
* indo de 0 a (VDD+0.5).

.protect
.include '~/vsc3.1lmodels'
.lib '~/vsc3.1lcorners' typical
.unprotect

.options nomod post probe
.temp 25

* curvas.sw0
.param VGS=0.1

VDD vdd 0 DC 2V
Vback 99 0 DC 0.6V
VGS 1 0 DC VGS
VDS 2 0 DC 0

JD vdd 2 2 99 jdep W=4u L=2u
JE 2 1 0 99 jenh W=10u L=1u
R 2 3 10      * Esta resistencia deve ser bem baixa
jd2 0 3 0 99 jdep w=10u l=2u

.dc VDS 0 2.5 0.01
.probe DC i1(jd) i1(je) i2(jd2) par('i1(je)+i2(jd2)')

* curvas.sw1
.ALTER VGS=0.2V
.param VGS=0.2

* curvas.sw2
.ALTER VGS=0.3V
.param VGS=0.3

* curvas.sw3
.ALTER VGS=0.4V
.param VGS=0.4

* curvas.sw4
.ALTER VGS=0.5V
.param VGS=0.5

* curvas.sw5
.ALTER VGS=0.6V
```

```
.param VGS=0.6

* curvas.sw6
.ALTER VGS=0.7V
.param VGS=0.7

* curvas.sw7
.ALTER VGS=0.8V
.param VGS=0.8

* curvas.sw8
.ALTER VGS=0.9V
.param VGS=0.9

.end
```

## Programa 2: “betadc.sp”.

```
Inverter Beta Analysis
* Esta suite simula uma familia de inversores com WD/LD = 8/2
* e beta variavel de 7 a 13.
* O objetivo eh estudar a influencia de beta no desempenho DC de um
* inversor.

.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

.options nomod post probe
.temp 25

* A parametrizacao e o uso de .ALTER permitem resimular o circuito com
* varios betas

.param beta=7 cl=50fF vdc=2V
.param WE='beta*4e-6'

VDD vdd 0 DC vdc
Vback 99 0 DC 0.6V
Vin 1 0 DC 0

* Foi simulado um fanout de 2

JD1 vdd 2 2 99 jdep W=8u L=2u
JE1 2 1 0 99 jenh W=WE L=1u

JD2 vdd 4 4 99 jdep W=8u L=2u
JE2 4 2 0 99 jenh W=WE L=1u

JD3 vdd 5 5 99 jdep W=8u L=2u
JE3 5 2 0 99 jenh W=WE L=1u
```

```

CL1 2 0 cl
CL2 4 0 cl
CL3 5 0 cl

.dc Vin 0 0.9 0.01
* Imprime pares de valores para a geracao da curva espelhada.
* Os valores numericos sao impressos no arquivo "betadc.lis".
.print DC v(2) v(1)
.probe DC v(1) v(2) i1(jd1) i1(je1) i2(je1) par('i1(jd1)*vdc')

.ALTER
.param beta=8
.ALTER
.param beta=9
.ALTER
.param beta=10
.ALTER
.param beta=11
.ALTER
.param beta=12
.ALTER
.param beta=13

.end

```

### Programa 3: "volbeta.sp".

Este programa gera curvas de grandezas elétricas em função de parâmetros de projeto do inversor, como  $\beta$  ou  $W_D$ , através de interpolação dos pares de valores fornecidos. Esta rotina pode ser adaptada para gerar-se as curvas de  $V_{OH}$  *versus*  $\beta$ ,  $P_{D,st}$  *versus*  $\beta$ ,  $P_{D,st}$  *versus*  $W_D$ ,  $t_{PD}$  *versus*  $W_D$ , etc.. Os valores de tensão, potência ou atraso são copiados de um arquivo **.lis** que os gere, como **betadc.lis**, ou **WDac.lis**.

```

VOL x beta

.options post nomod probe

v 1 0 pw1(beta, vol)      * Plota vol em funcao de beta
r 1 0 1

.data voldata
    beta      vol
    5         131.4980m
    6         108.8115m
    7         93.3249m
    8         82.0959m
    9         73.5894m
    10        66.9269m
    11        61.5699m
    12        57.1704m
    13        53.4936m
    15        47.6979m
    17        43.3380m
    19        39.9401m
    20        38.5077m
    22        36.0499m
    25        33.1281m

```

```
.enddata

.dc data=voldata
.probe v(1)
.end
```

#### Programa 4: “betaac.sp”.

Inverter Beta Analysis

\* Esta suite simula uma familia de inversores com  $WD/LD = 8/2$  e beta variavel de 5 a 25.

\* O objetivo eh estudar a influencia de beta no desempenho transiente de um inversor

```
.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect
```

```
.options nomod post probe
.temp 25
```

\* A parametrizacao e o uso de .ALTER permitem resimular o circuito com varios betas

```
.param beta=5 cl=50f vdc=2V
.param WE='beta*4*1e-6'
.param CGSP='.28*beta*4e-15'
.param COUT='.11*beta*4e-15 + .39*8*2*1e-15'
.param CGSPfix='.28*40e-15'
.param COUTfix='.11*40e-15 + .39*8*2*1e-15'
```

```
VDD vdd 0 DC vdc
Vback 99 0 DC 0.6V
.global 99 vdd
```

```
*           in      out
.subckt inversor 1      2
'
JD vdd 2 2 99 jdep W=8u L=2u
JE 2 1 0 99 jenh W=40u L=1u
```

```
CGSPE 1 0 CGSPfix
CGDPE 2 1 CGSPfix
COUT 2 0 COUTfix
```

```
.ends
```

\* Inversor em analise

```
JD vdd 3 3 99 jdep W=8u L=2u
JE 3 2 0 99 jenh W=WE L=1u
```

```
CGSPE 2 0 CGSP
CGDPE 3 2 CGSP
```



```
COUT 3 0 COUT
```

```
*   in   out
X1  1    2   inversor
X3  2    4   inversor
X4  3    5   inversor
X5  3    6   inversor
```

```
CL1 2 0 cl
CL2 3 0 cl
CL3 4 0 cl
CL4 5 0 cl
CL5 6 0 cl
```

```
VCLK 1 0 PULSE(Vil, Vih, td, tr, tf, pw, per)
.param Vil=0V Vih=0.7V td=400ps tr=100ps tf=100ps pw=400ps per=1ns
```

```
.tran 20ps 2.5ns
.probe v(2) v(3)
```

```
* calcula o valor minimo da entrada
.measure tran Vol1 avg v(2) from=1.7ns to=1.9ns
```

```
* calcula o valor maximo da entrada
.measure tran Voh1 avg v(2) from=2.35ns to=2.45ns
```

```
* calcula o swing logico da entrada
.measure tran Vpp1 param='Voh1-Vol1'
```

```
* calcula o ponto de 50% para o calculo do tpd
.measure tran fifty1 param='0.5*Vpp1+Vol1'
```

```
* calcula o valor minimo da saida
.measure tran Vol2 avg v(3) from=1.3ns to=1.5ns
```

```
* calcula o valor maximo da saida
.measure tran Voh2 avg v(3) from=1.95ns to=2ns
```

```
* calcula o swing logico da saida
.measure tran Vpp2 param='Voh2-Vol2'
```

```
* calcula os pontos de 10% e 90% para calculo de tr e tf
.measure tran ten2 param='0.1*Vpp2+Vol2'
.measure tran ninety2 param='0.9*Vpp2+Vol2'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty2 param='0.5*Vpp2+Vol2'
```

```
* calcula o thl da saida do inversor
.measure tran thl trig v(3) val=ninety2 fall=2
+      targ v(3) val=ten2      fall=2
```

```
* calcula o tlh da saida do inversor
.measure tran tlh trig v(3) val=ten2      rise=2
+      targ v(3) val=ninety2 rise=2
```

```

* calcula o tphl
.measure tran tphl trig v(2) val=fifty1 rise=2
+          targ v(3) val=fifty2 fall=2

* calcula o tplh
.measure tran tplh trig v(2) val=fifty1 fall=2
+          targ v(3) val=fifty2 rise=2

* calcula o atraso de propagacao tpd
.measure tran tpd param='(tphl+tplh)*0.5'

.ALTER
.param beta=6
.ALTER
.param beta=7
.ALTER
.param beta=8
.ALTER
.param beta=9
.ALTER
.param beta=10
.ALTER
.param beta=11
.ALTER
.param beta=12
.ALTER
.param beta=13
.ALTER
.param beta=15
.ALTER
.param beta=17
.ALTER
.param beta=19
.ALTER
.param beta=21
.ALTER
.param beta=23
.ALTER
.param beta=25

.end

```

### Programa 5: “WDdc.sp”.

Inverter WD/LD Analysis

- \* Esta suite simula uma familia de inversores com WD/LD variavel de 2 a 30 e beta = 10,
- \* para estudar a influencia de WD/LD no desempenho DC de um inversor

```

.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

```

```
.options nomod post probe
.temp 25
```

\* A parametrizacao e o uso de .ALTER permitem resimular o circuito com varios WD's

```
.param WD=4u cl=50fF vdc=2V
.param WE='5*WD'
```

```
VDD vdd 0 DC vdc
Vback 99 0 DC 0.6V
Vin 1 0 DC 0
```

\* Foi simulado um fanout de 2

```
JD1 vdd 2 2 99 jdep W=WD L=2u
JE1 2 1 0 99 jenh W=WE L=1u
```

```
JD2 vdd 4 4 99 jdep W=WD L=2u
JE2 4 2 0 99 jenh W=WE L=1u
```

```
JD3 vdd 5 5 99 jdep W=WD L=2u
JE3 5 2 0 99 jenh W=WE L=1u
```

```
CL1 2 0 cl
CL2 4 0 cl
CL3 5 0 cl
```

```
.dc Vin 0 0.8 0.01
.print DC v(2) il(jd1) il(je1) par('il(jd1)*vdc')
.probe DC v(2) il(jd1) il(je1) par('il(jd1)*vdc')
```

```
* WDdc.sw1
.ALTER
.param WD=8u
```

```
* WDdc.sw2
.ALTER
.param WD=12u
```

```
* WDdc.sw3
.ALTER
.param WD=16u
```

```
* WDdc.sw4
.ALTER
.param WD=20u
```

```
* WDdc.sw5
.ALTER
.param WD=24u
```

```
* WDdc.sw6
.ALTER
.param WD=32u
```

```
* WDdc.sw7
.ALTER
```

```
.param WD=40u

* WDdc.sw8
.ALTER
.param WD=48u

* WDdc.sw9
.ALTER
.param WD=60u

.end
```

## Programa 6: "WDac.sp".

Inverter WD/LD Analysis

\* Esta suite simula uma familia de inversores com WD/LD variavel de 2 a 30 e beta = 10,  
\* para estudar a influencia de WD no desempenho AC de um inversor

```
.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect
```

```
.options nomod post probe
.temp 25
```

\* A parametrizacao e o uso de .ALTER permitem resimular o circuito com varios WD's

```
.param WD=4u cl=50fF vdc=2V
.param WE='5*WD' * Para manter beta = 10
.param CGSP='.28*WE*1e-15'
.param COUT='.11*WE*1e-15 + .39*WD*2*1e-15'
```

```
VDD vdd 0 DC vdc
Vback 99 0 DC 0.6V
.global 99 vdd
```

```
*          in    out
.subckt inversor 1    2
```

```
JD vdd 2 2 99 jdep W=WD L=2u
JE 2 1 0 99 jenh W=WE L=1u
```

```
CGSPE 1 0 CGSP
CGDPE 2 1 CGSP
COUT 2 0 COUT
```

```
.ends
```

```
*   in    out
X1  1     2   inversor
X2  2     3   inversor   *inversor em analise
X3  2     4   inversor
```

```
X4 3 5 inversor
X5 3 6 inversor
```

```
CL1 2 0 cl
CL2 3 0 cl
CL3 4 0 cl
CL4 5 0 cl
CL5 6 0 cl
```

```
VCLK 1 0 PULSE(Vil, Vih, td, tr, tf, pw, per)
.param Vil=0V Vih=0.7V td=400ps tr=100ps tf=100ps pw=400ps per=1ns
```

```
.tran 20ps 2.5ns
.probe v(2) v(3)
```

```
* calcula o valor minimo da entrada
.measure tran Vol1 avg v(2) from=1.6ns to=1.9ns
```

```
* calcula o valor maximo da entrada
.measure tran Voh1 avg v(2) from=2.3ns to=2.4ns
```

```
* calcula o swing logico da entrada
.measure tran Vpp1 param='Voh1-Vol1'
```

```
* calcula o ponto de 50% para o calculo do tpd
.measure tran fifty1 param='0.5*Vpp1+Vol1'
```

```
* calcula o valor minimo da saida
.measure tran Vol2 avg v(3) from=2.3ns to=2.5ns
```

```
* calcula o valor maximo da saida
.measure tran Voh2 avg v(3) from=1.9ns to=1.97ns
```

```
* calcula o swing logico da saida
.measure tran Vpp2 param='Voh2-Vol2'
```

```
* calcula os pontos de 10% e 90% para calculo de tr e tf
.measure tran ten2 param='0.1*Vpp2+Vol2'
.measure tran ninety2 param='0.9*Vpp2+Vol2'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty2 param='0.5*Vpp2+Vol2'
```

```
* calcula o thl da saida do inversor
.measure tran thl trig v(3) val=ninety2 fall=2
+      targ v(3) val=ten2 fall=2
```

```
* calcula o tlh da saida do inversor
.measure tran tlh trig v(3) val=ten2 rise=2
+      targ v(3) val=ninety2 rise=2
```

```
* calcula o tphl
.measure tran tphl trig v(2) val=fifty1 rise=2
+      targ v(3) val=fifty2 fall=2
```

```
* calcula o tplh
```

```

.measure tran tplh trig v(2) val=fifty1 fall=2
+          targ v(3) val=fifty2 rise=2

* calcula o atraso de propagacao tpd
.measure tran tpd param='(tphl+tplh)*0.5'

* WDac.tr1
.ALTER
.param WD=8u

* WDac.tr2
.ALTER
.param WD=12u

* WDac.tr3
.ALTER
.param WD=16u

* WDac.tr4
.ALTER
.param WD=20u

* WDac.tr5
.ALTER
.param WD=24u

* WDac.tr6
.ALTER
.param WD=32u

* WDac.tr7
.ALTER
.param WD=40u

* WDac.tr8
.ALTER
.param WD=48u

* WDac.tr9
.ALTER
.param WD=50u

* WDac.tra
.ALTER
.param WD=54u

* WDac.trb
.ALTER
.param WD=60u

.end

```

### **Programa 7: “temp.sp”.**

Inverter Sensitivity Analysis

- \* Esta suite simula inversores com  $WD/LD = 8/2$  e  $\beta = 10$ .
- \* O objetivo eh estudar a influencia da temperatura no desempenho

```

* DC de um inversor.

.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

.options nomod post probe
.temp 0

.param cl=50fF vdc=2V

VDD vdd 0 DC vdc
Vback 99 0 DC 0.6V
Vin 1 0 DC 0

JD1 vdd 2 2 99 jdep W=8u L=2u
JE1 2 1 0 99 jenh W=40u L=1u

JD2 vdd 3 3 99 jdep W=8u L=2u
JE2 3 2 0 99 jenh W=40u L=1u

JD3 vdd 4 4 99 jdep W=8u L=2u
JE3 4 2 0 99 jenh W=40u L=1u

CL1 2 0 cl
CL2 3 0 cl
CL3 4 0 cl

.dc Vin 0 0.9 0.01
.print DC v(2) par('il(jd1)*vdc')
.probe DC v(2) par('il(jd1)*vdc')

.ALTER
.temp 25

.ALTER
.temp 50

.ALTER
.temp 85

.end

```

## Programa 8: “tempac.sp”.

### Inverter Sensitivity Analysis

\* Esta suite simula inversores com  $WD/LD = 8/2$  e  $\beta = 10$ .  
 \* O objetivo eh estudar a influencia da temperatura no desempenho  
 \* DC de um inversor.

```

.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

.options nomod post probe

```

```

.temp 0

.param    cl=50fF    vdc=2V
.param    CGSP='.28*40e-15'
.param    COUT='.11*40e-15 + .39*8*2*1e-15'

VDD vdd 0 DC vdc
Vback 99 0 DC 0.6V
.global 99 vdd

*          in      out
.subckt inversor      1      2

JD vdd 2 2 99 jdep W=8u L=2u
JE 2 1 0 99 jenh W=40u L=1u

CGSPE 1 0 CGSP
CGDPE 2 1 CGSP
COUT 2 0 COUT

.ends

*    in    out
X1 1 2    inversor
X2 2 3    inversor    *inversor em analise
X3 2 4    inversor
X4 3 5    inversor
X5 3 6    inversor

CL1 2 0 cl
CL2 3 0 cl
CL3 4 0 cl
CL4 5 0 cl
CL5 6 0 cl

VCLK 1 0 PULSE(Vil, Vih, td, tr, tf, pw, per)
.param Vil=0V Vih=0.7V td=400ps tr=100ps tf=100ps pw=400ps per=1ns

.tran 20ps 2.5ns
.probe v(2) v(3)

* calcula o valor minimo da entrada
.measure tran Vol1 avg v(2) from=1.6ns to=1.9ns

* calcula o valor maximo da entrada
.measure tran Voh1 avg v(2) from=2.35ns to=2.45ns

* calcula o swing logico da entrada
.measure tran Vpp1 param='Voh1-Vol1'

* calcula o ponto de 50% para o calculo do tpd
.measure tran fifty1 param='0.5*Vpp1+Vol1'

* calcula o valor minimo da saida
.measure tran Vol2 avg v(3) from=1.35ns to=1.5ns

```



```

* calcula o valor maximo da saida
.measure tran Voh2 avg v(3) from=1.9ns to=2.05ns

* calcula o swing logico da saida
.measure tran Vpp2 param='Voh2-Vol2'

* calcula os pontos de 10% e 90% para calculo de tr e tf
.measure tran ten2 param='0.1*Vpp2+Vol2'
.measure tran ninety2 param='0.9*Vpp2+Vol2'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty2 param='0.5*Vpp2+Vol2'

* calcula o thl da saida do inversor
.measure tran thl trig v(3) val=ninety2 fall=2
+      targ v(3) val=ten2      fall=2

* calcula o tlh da saida do inversor
.measure tran tlh trig v(3) val=ten2      rise=2
+      targ v(3) val=ninety2 rise=2

* calcula o tphl
.measure tran tphl trig v(2) val=fifty1 rise=2
+      targ v(3) val=fifty2 fall=2

* calcula o tplh
.measure tran tplh trig v(2) val=fifty1 fall=2
+      targ v(3) val=fifty2 rise=2

* calcula o atraso de propagacao tpd
.measure tran tpd param='(tphl+tplh)*0.5'

.ALTER
.temp 25

.ALTER
.temp 50

.ALTER
.temp 85

.end

```

## Programa 9: “protem.sp”.

### Inverter Sensitivity Analysis

- \* Esta suite simula uma familia de inversores com  $W_D/L_D = 8/2$  e  $\beta = 10$ .
- \* O objetivo eh estudar a influencia do processo e da temperatura no desempenho
- \* DC de um inversor. A tolerancia da fonte VDD eh de 10%

```

.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

```

```

.options nomod post probe
.temp 25

.param cl=50fF vdc=2V

VDD vdd 0 DC vdc
Vback 99 0 DC 0.6V
Vin 1 0 DC 0

JD1 vdd 2 2 99 jdep W=8u L=2u
JE1 2 1 0 99 jenh W=40u L=1u

JD2 vdd 3 3 99 jdep W=8u L=2u
JE2 3 2 0 99 jenh W=40u L=1u

JD3 vdd 4 4 99 jdep W=8u L=2u
JE3 4 2 0 99 jenh W=40u L=1u

CL1 2 0 cl
CL2 3 0 cl
CL3 4 0 cl

.dc Vin 0 0.9 0.01
.print DC v(2) v(1) *par('il(jd1)*vdc')
.probe DC v(2) par('il(jd1)*vdc')

.ALTER Slowest Case
.del lib '~/vsc3.11corners' typical
.lib '~/vsc3.11corners' slow
.temp 0
.param vdc=1.8V

.ALTER Highest Power Case
.del lib '~/vsc3.11corners' slow
.lib '~/vsc3.11corners' fast
.temp 85
.param vdc=2.2V

.end

```

## Programa 10: “prteac.sp”.

### Inverter Sensitivity Analysis

- \* Esta suite simula uma familia de inversores com  $WD/LD = 8/2$  e  $\beta = 10$ .
- \* O objetivo eh estudar a influencia do processo e da temperatura no
- \* desempenho DC de um inversor. A tolerancia da fonte VDD eh de 10%

```

.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

.options nomod post probe
.temp 25

```

```

.param    cl=50fF    vdc=2V
.param    CGSP='.28*40e-15'
.param    COUT='.11*40e-15 + .39*8*2*1e-15'

VDD vdd 0 DC vdc
Vback 99 0 DC 0.6V
.global 99 vdd

*          in    out
.subckt inversor    1    2

JD vdd 2 2 99 jdep W=8u L=2u
JE  2  1 0 99 jenh W=40u L=1u

CGSPE 1 0 CGSP
CGDPE 2 1 CGSP
COUT  2 0 COUT

.ends

*    in    out
X1   1    2    inversor
X2   2    3    inversor    *inversor em analise
X3   2    4    inversor
X4   3    5    inversor
X5   3    6    inversor

CL1 2 0 cl
CL2 3 0 cl
CL3 4 0 cl
CL4 5 0 cl
CL5 6 0 cl

VCLK 1 0 PULSE(Vil, Vih, td, tr, tf, pw, per)
.param Vil=0V Vih=0.7V td=400ps tr=100ps tf=100ps pw=400ps per=1ns

.tran 20ps 2.5ns
.probe v(2) v(3)

* calcula o valor minimo da entrada
.measure tran Vol1 avg v(2) from=1.7ns to=1.9ns

* calcula o valor maximo da entrada
.measure tran Voh1 avg v(2) from=2.35ns to=2.45ns

* calcula o swing logico da entrada
.measure tran Vpp1 param='Voh1-Vol1'

* calcula o ponto de 50% para o calculo do tpd da entrada
.measure tran fifty1 param='0.5*Vpp1+Vol1'

* calcula o valor minimo da saida
.measure tran Vol2 avg v(3) from=1.4ns to=1.5ns

```

```

* calcula o valor maximo da saida
.measure tran Voh2 avg v(3) from=1.95ns to=2ns

* calcula o swing logico da saida
.measure tran Vpp2 param='Voh2-Vol2'

* calcula os pontos de 10% e 90% para calculo de tr e tf
.measure tran ten2 param='0.1*Vpp2+Vol2'
.measure tran ninety2 param='0.9*Vpp2+Vol2'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty2 param='0.5*Vpp2+Vol2'

* calcula o thl da saida do inversor
.measure tran thl trig v(3) val=ninety2 fall=2
+      targ v(3) val=ten2      fall=2

* calcula o tlh da saida do inversor
.measure tran tlh trig v(3) val=ten2      rise=2
+      targ v(3) val=ninety2 rise=2

* calcula o tphl
.measure tran tphl trig v(2) val=fifty1 rise=2
+      targ v(3) val=fifty2 fall=2

* calcula o tplh
.measure tran tplh trig v(2) val=fifty1 fall=2
+      targ v(3) val=fifty2 rise=2

* calcula o atraso de propagacao tpd
.measure tran tpd param='(tphl+tplh)*0.5'

.ALTER Slowest Case
.protect
.del lib '~/vsc3.11corners' typical
.lib '~/vsc3.11corners' slow
.unprotect
.temp 0
.param vdc=1.8V

.ALTER 2nd slow
.temp 85

.ALTER Highest Power Case
.protect
.del lib '~/vsc3.11corners' slow
.lib '~/vsc3.11corners' fast
.unprotect
.temp 85
.param vdc=2.2V

.ALTER 2nd fast
.temp 0

.end

```

## Programa 11: "FOdc.sp".

Inverter Fan-Out Analysis

- \* Esta suite simula um inversor com  $WD/LD = 8/2$  e  $\beta = 10$ .
- \* O objetivo é estudar a influência do fan-out no seu desempenho DC.

```
.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' fast
.unprotect

.options nomod post probe
.temp 85

.param cl=50fF vdc=1.8V

VDD vdd 0 DC vdc
Vback 99 0 DC 0.6V
Vin 1 0 DC 0

JD1 vdd 2 2 99 jdep W=8u L=2u
JE1 2 1 0 99 jenh W=40u L=1u

JD2 vdd 3 3 99 jdep W=8u L=2u
JE2 3 2 0 99 jenh W=40u L=1u

CL1 2 0 cl
CL2 3 0 cl

.dc Vin 0 0.8 0.01
.print DC v(2) par('il(jd1)*vdc')
.probe DC v(2) par('il(jd1)*vdc')

.ALTER FO=2
JD3 vdd 4 4 99 jdep W=8u L=2u
JE3 4 2 0 99 jenh W=40u L=1u
CL3 4 0 cl

.ALTER FO=3
JD4 vdd 5 5 99 jdep W=8u L=2u
JE4 5 2 0 99 jenh W=40u L=1u
CL4 5 0 cl

.ALTER FO=4
JD5 vdd 6 6 99 jdep W=8u L=2u
JE5 6 2 0 99 jenh W=40u L=1u
CL5 6 0 cl

.end
```

## Programa 12: "FOac.sp".

Inverter Fan-Out Analysis

- \* Esta suite simula um inversor com  $WD/LD = 8/2$  e  $\beta = 10$ ,

\* para estudar a influencia do fan-out no seu desempenho transiente.

```
.protect
.include '~/vsc3.1lmodels'
.lib '~/vsc3.1lcorners' slow
.unprotect
```

```
.options nomod post probe
.temp 0
```

```
.param cl=50fF vdc=1.8V
.param CGSP='.28*40e-15'
.param COUT='.11*40e-15 + .39*16*1e-15'
```

```
VDD vdd 0 DC vdc
Vback 99 0 DC 0.6V
.global 99 vdd
```

```
*          in    out
.subckt inversor 1 2
```

```
JD vdd 2 2 99 jdep W=8u L=2u
JE 2 1 0 99 jenh W=40u L=1u
```

```
CGSPE 1 0 CGSP
CGDPE 2 1 CGSP
COUT 2 0 COUT
```

```
.ends
```

```
* in out
X1 1 2 inversor
X2 2 3 inversor *inversor em analise
X3 2 4 inversor
X4 3 5 inversor
```

```
CL1 2 0 cl
CL2 3 0 cl
CL3 4 0 cl
CL4 5 0 cl
```

```
VCLK 1 0 PULSE(Vil, Vih, td, tr, tf, pw, per)
.param Vil=0V Vih=0.7V td=400ps tr=100ps tf=100ps pw=400ps per=1ns
```

```
.tran 20ps 2.5ns
.probe v(2) v(3)
```

```
.ALTER FO=2
* in out
X5 3 6 inversor
CL5 6 0 cl
```

```
.ALTER FO=3
* in out
X6 3 7 inversor
```

```
CL6 7 0 cl
```

```
.ALTER FO=4
*   in   out
X7   3   8   inversor
CL7 8 0 cl

.end
```

### Programa 13: “norDC.sp”.

SIMULACAO DC DE PORTA NOR

\* Esta suite analisa o chaveamento de uma entrada de uma  
\* porta NOR de FI=3 e beta=10, no aspecto DC

```
.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect
```

```
.options post nomod probe
.temp 25
```

\* Adotou-se os valores de 70mV para VOL, e de 700mV para VOH

```
.param cl=50f
.param v4=70m v5=70m v7=70m v8=70m
VDD vdd 0 DC 2V
Vback 99 0 DC 0.6V
```

```
JD1 vdd 2 2 99 jdep W=8u L=2u
JE1A 2 3 0 99 jenh W=40u L=1u
JE1B 2 4 0 99 jenh W=40u L=1u
JE1C 2 5 0 99 jenh W=40u L=1u
```

```
JD2 vdd 6 6 99 jdep W=8u L=2u
JE2A 6 2 0 99 jenh W=40u L=1u
JE2B 6 7 0 99 jenh W=40u L=1u
JE2C 6 8 0 99 jenh W=40u L=1u
```

```
CL1 2 0 cl
CL2 6 0 cl
```

```
V3 3 0 dc 0
V4 4 0 dc v4
V5 5 0 dc v5
V7 7 0 dc v7
V8 8 0 dc v8
```

```
.dc v3 0 0.8 10m
.print dc v(3) v(2) par('il(jd1)*2') *o no. '2' eh o valor de VDD
.probe dc v(3) v(2) il(jd1) il(jela) il(jelb) il(jelc) par('il(jd1)*2')
```

```
.alter
.param v4=700m

.alter
```

```
.param v5=700m

.alter
.param v4=70m v5=70m
.param v7=700m v8=700m

.end
```

## Programa 14: “norDC2.sp”.

SIMULACAO DC DE PORTA NOR

\* Esta suite analisa o chaveamento de duas entradas de uma  
 \* porta NOR de FI=3 e beta=10, no aspecto DC

```
.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

.options post nomod probe
.temp 25

.param cl=50f
.param v5=70m v7=70m v8=70m
VDD vdd 0 DC 2V
Vback 99 0 DC 0.6V

JD1 vdd 2 2 99 jdep W=8u L=2u
JE1A 2 3 0 99 jenh W=40u L=1u
JE1B 2 3 0 99 jenh W=40u L=1u
JE1C 2 5 0 99 jenh W=40u L=1u

JD2 vdd 6 6 99 jdep W=8u L=2u
JE2A 6 2 0 99 jenh W=40u L=1u
JE2B 6 7 0 99 jenh W=40u L=1u
JE2C 6 8 0 99 jenh W=40u L=1u

CL1 2 0 cl
CL2 6 0 cl

V3 3 0 dc 0
V5 5 0 dc v5
V7 7 0 dc v7
V8 8 0 dc v8

.dc v3 0 0.8 10m
.print dc v(3) v(2) par('i1(jd1)*2')
.probe dc v(3) v(2) i1(jd1) i1(jela) i1(jelb) i1(jelc) par('i1(jd1)*2')

.end
```



### Programa 15: “norDC3.sp”.

```
SIMULACAO DC DE PORTA NOR
* Esta suite analisa o chaveamento de tres entradas de uma
* porta NOR de FI=3 e beta=10, no aspecto DC

.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

.options post nomod probe
.temp 25

.param cl=50f
.param v7=70m v8=70m
VDD vdd 0 DC 2V
Vback 99 0 DC 0.6V

JD1 vdd 2 2 99 jdep W=8u L=2u
JE1A 2 3 0 99 jenh W=40u L=1u
JE1B 2 3 0 99 jenh W=40u L=1u
JE1C 2 3 0 99 jenh W=40u L=1u

JD2 vdd 6 6 99 jdep W=8u L=2u
JE2A 6 2 0 99 jenh W=40u L=1u
JE2B 6 7 0 99 jenh W=40u L=1u
JE2C 6 8 0 99 jenh W=40u L=1u

CL1 2 0 cl
CL2 6 0 cl

V3 3 0 dc 0
V7 7 0 dc v7
V8 8 0 dc v8

.dc v3 0 0.8 10m
.print dc v(3) v(2) par('il(jd1)*2')
.probe dc v(3) v(2) il(jd1) il(jela) il(jelb) il(jelc) par('il(jd1)*2')

.end
```

### Programa 16: “norDC4.sp”.

```
SIMULACAO DC DE PORTA NOR
* Esta suite analisa o chaveamento de quatro entradas de uma
* porta NOR de FI=4 e beta=10, no aspecto DC

.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

.options post nomod probe
.temp 25

.param cl=50f
```

```

.param v7=70m v8=70m v9=70m
VDD vdd 0 DC 2V
Vback 99 0 DC 0.6V

JD1 vdd 2 2 99 jdep W=8u L=2u
JE1A 2 3 0 99 jenh W=40u L=1u
JE1B 2 3 0 99 jenh W=40u L=1u
JE1C 2 3 0 99 jenh W=40u L=1u
JE1D 2 3 0 99 jenh W=40u L=1u

JD2 vdd 6 6 99 jdep W=8u L=2u
JE2A 6 2 0 99 jenh W=40u L=1u
JE2B 6 7 0 99 jenh W=40u L=1u
JE2C 6 8 0 99 jenh W=40u L=1u
JE2D 6 9 0 99 jenh W=40u L=1u

CL1 2 0 cl
CL2 6 0 cl

V3 3 0 dc 0
V7 7 0 dc v7
V8 8 0 dc v8
V9 9 0 dc v9

.dc v3 0 0.8 10m
.print dc v(3) v(2) par('i1(jd1)*2')
.probe dc v(3) v(2) i1(jd1) i1(je1a) i1(je1b) i1(je1c) par('i1(jd1)*2')

.end

```

### Programa 17: “norDCorth.sp”.

SIMULACAO DC DE PORTA NOR  
 \* Esta suite analisa o chaveamento de uma  
 \* entrada de uma porta NOR de FI=1, 2, 3 e 4  
 \* e beta=10, no aspecto DC

```

.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

.options post nomod probe
.temp 25

.param cl=50f
VDD vdd 0 DC 2V
Vback 99 0 DC 0.6V

JD1 vdd 2 2 99 jdep W=8u L=2u
JE1A 2 3 0 99 jenh W=40u L=1u

JD2 vdd 10 10 99 jdep W=8u L=2u
JE2A 10 2 0 99 jenh W=40u L=1u
JE2B 10 7 0 99 jenh W=40u L=1u
JE2C 10 8 0 99 jenh W=40u L=1u

```

```

CL1 2 0 cl
CL2 10 0 cl

V3 3 0 dc 0
V7 7 0 dc 70m
V8 8 0 dc 70m

.dc v3 0 0.8 10m
.print dc v(3) v(2)
.probe dc v(3) v(2)

.alter FI=2
JE1B 2 4 0 99 jenh W=40u L=1u
V4 4 0 dc 70m

.alter FI=3
JE1C 2 5 0 99 jenh W=40u L=1u
V5 5 0 dc 70m

.alter FI=4
JE1D 2 6 0 99 jenh W=40u L=1u
V6 6 0 dc 70m

.end

```

### Programa 18: "norAC.sp".

```

SIMULACAO TRANSIENTE DE PORTA NOR
* Esta suite analisa o chaveamento de uma
* entrada de uma porta NOR de FI=1, 2, 3, 4 e 5
* e beta=10, no aspecto transiente

.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

.options post nomod probe
.temp 25

.param cl=50f
.param CGSP='.28*40e-15'
.param COUT1='.11*40e-15 + .39*16*1e-15' *para NOR de FI=1
.param COUT3='.33*40e-15 + .39*16*1e-15' *para NOR de FI=3

VDD vdd 0 DC 2V
Vback 99 0 DC 0.6V
.global 99 vdd

*          in1      in2      in3      out
.subckt nor3      3      4      5      2

JD vdd 2 2 99 jdep W=8u L=2u

```

```
JEA 2 3 0 99 jenh W=40u L=1u
JEB 2 4 0 99 jenh W=40u L=1u
JEC 2 5 0 99 jenh W=40u L=1u
```

```
CGSPEA 3 0 CGSP
CGDPEA 2 3 CGSP
CGSPEB 4 0 CGSP
CGDPEB 2 4 CGSP
CGSPEC 5 0 CGSP
CGDPEC 2 5 CGSP
COUT 2 0 COUT3
```

```
.ends
```

```
*      in1  in2  in3  out
X1 1 2 2 4 nor3
X3 4 6 6 8 nor3
X4 9 10 10 14 nor3
X5 9 12 12 15 nor3
```

```
* Porta NOR a ser medida
```

```
* Saida: 9 Entradas: 4, 5, 7, 11, 13
```

```
JD vdd 9 9 99 jdep W=8u L=2u
```

```
JEA 9 4 0 99 jenh W=40u L=1u
```

```
CGSPEA 4 0 CGSP
```

```
CGDPEA 9 4 CGSP
```

```
COUT1 9 0 COUT1
```

```
CL1 4 0 cl
```

```
CL2 9 0 cl
```

```
CL3 8 0 cl
```

```
CL4 14 0 cl
```

```
CL5 15 0 cl
```

```
VCLK 1 0 PULSE(Vil, Vih, td, tr, tf, pw, per)
```

```
.param Vil=0V Vih=0.7V td=400ps tr=100ps tf=100ps pw=400ps per=1ns
```

```
V2 2 0 dc 70m
```

```
V6 6 0 dc 700m
```

```
V10 10 0 dc 70m
```

```
V12 12 0 dc 700m
```

```
.tran 10ps 2.5ns
```

```
.probe TRAN v(4) v(9) il(jea)
```

```
* calcula o valor minimo da entrada
```

```
.measure tran Voll avg v(4) from=1.7ns to=1.9ns
```

```
* calcula o valor maximo da entrada
```

```
.measure tran Voh1 avg v(4) from=2.35ns to=2.45ns
```

```
* calcula o swing logico da entrada
```

```
.measure tran Vpp1 param='Voh1-Voll'
```

```
* calcula o ponto de 50% para o calculo do tpd
```

```
.measure tran fifty1 param='0.5*Vpp1+Voll'
```

```

* calcula o valor minimo da saida
.measure tran Vol2 avg v(9) from=1.35ns to=1.5ns

* calcula o valor maximo da saida
.measure tran Voh2 avg v(9) from=1.95ns to=2.05ns

* calcula o swing logico da saida
.measure tran Vpp2 param='Voh2-Vol2'

* calcula os pontos de 10% e 90% para calculo de tr e tf
.measure tran ten2 param='0.1*Vpp2+Vol2'
.measure tran ninety2 param='0.9*Vpp2+Vol2'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty2 param='0.5*Vpp2+Vol2'

* calcula o thl da saida do inversor
.measure tran thl trig v(9) val=ninety2 fall=2
+      targ v(9) val=ten2      fall=2

* calcula o tlh da saida do inversor
.measure tran tlh trig v(9) val=ten2      rise=2
+      targ v(9) val=ninety2 rise=2

* calcula o tphl
.measure tran tphl trig v(4) val=fifty1 rise=2
+      targ v(9) val=fifty2 fall=2

* calcula o tplh
.measure tran tplh trig v(4) val=fifty1 fall=2
+      targ v(9) val=fifty2 rise=2

* calcula o atraso de propagacao tpd
.measure tran tpd param='(tphl+tplh)*0.5'

.alter FI=2
JEB  9  5 0 99 jenh W=40u L=1u
CGSP  5 0 CGSP
CGDPEB  9 5 CGSP
COUT2 9 0 COUT1
V5 5 0 dc 70m

.alter FI=3
JEC  9  7 0 99 jenh W=40u L=1u
CGSPEC  7 0 CGSP
CGDPEC  9 7 CGSP
COUT3 9 0 COUT1
V7 7 0 dc 70m

.alter FI=4
JED  9  11 0 99 jenh W=40u L=1u
CGSPED  11 0 CGSP
CGDPED  9 11 CGSP

```

```
COUT4 9 0 COUT1
V11 11 0 dc 70m
```

```
.alter FI=5
JEE 9 13 0 99 jenh W=40u L=1u
CGSPPEE 13 0 CGSP
CGDPPEE 9 13 CGSP
COUT5 9 0 COUT1
V13 13 0 dc 70m
```

```
.end
```

### Programa 19: “bufdc.sp”.

SDCFL driver Analysis

\* O objetivo eh estudar a influencia do beta do inversor  
 \* que aciona o buffer e escolher o beta otimo

```
.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect
```

```
.options nomod post probe
.temp 25
```

```
.param cl=50f vdc=2V cl2=200f beta=3 WE='beta*4e-6'
```

```
VDD vdd 0 DC vdc
Vback 99 0 DC 0.6V
Vin 1 0 DC 0
```

```
JD1 vdd 2 2 99 jdep W=8u L=2u
JE1 2 1 0 99 jenh W=WE L=1u
CL1 2 0 cl
```

```
JEbuf vdd 2 3 99 jenh W=60u L=1u
JDbuf 3 0 0 99 jdep W=40u L=2u
CL2 3 0 cl2
* O noh 3 eh a saida do buffer
```

```
JD2 vdd 4 4 99 jdep W=8u L=2u
JE2 4 3 0 99 jenh W=40u L=1u
CL3 4 0 cl
```

```
.dc Vin 0 0.8 0.01
.probe DC v(3)
```

```
.ALTER
.param beta=4
```

```
.ALTER
```

```
.param beta=5

.ALTER
.param beta=6

.ALTER
.param beta=8

.ALTER
.param beta=10

.ALTER
.param beta=12

.end
```

## Programa 20: “bufdc2.sp”.

SDCFL Fan-Out Analysis

\* O objetivo eh estudar a influencia do fan-out no seu desempenho DC.

```
.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

.options nomod post probe
.temp 25

.param cl=50f vdc=2V
.param cl2=500f

VDD vdd 0 DC vdc
Vback 99 0 DC 0.6V
Vin 1 0 DC 0

* O beta do inversor que aciona o buffer deve ser 5
JD1 vdd 2 2 99 jdep W=8u L=2u
JE1 2 1 0 99 jenh W=20u L=1u
CL1 2 0 cl

JEbuf vdd 2 3 99 jenh W=30u L=1u
JDbuf 3 0 0 99 jdep W=20u L=2u
CL2 3 0 cl2
* O noh 3 eh a saida do buffer

JD2 vdd 4 4 99 jdep W=8u L=2u
JE2 4 3 0 99 jenh W=40u L=1u
CL3 4 0 cl

.dc Vin 0 1 0.01
.print DC v(3) i2(jebuf) par('(i1(jd1)+i1(jebuf))*vdc')
.probe DC v(3) i2(jebuf) par('(i1(jd1)+i1(jebuf))*vdc')

.ALTER FO=2
JD3 vdd 5 5 99 jdep W=8u L=2u
JE3 5 3 0 99 jenh W=40u L=1u
```

```
CL4 5 0 cl
```

```
.ALTER FO=3
JD4 vdd 6 6 99 jdep W=8u L=2u
JE4 6 3 0 99 jenh W=40u L=1u
CL5 6 0 cl
```

```
.ALTER FO=4
JD5 vdd 7 7 99 jdep W=8u L=2u
JE5 7 3 0 99 jenh W=40u L=1u
CL6 7 0 cl
```

```
.end
```

### Programa 21: "bufdc3.sp".

SDCFL WD Analysis

\* O objetivo eh estudar a influencia de WD no seu desempenho DC.

```
.protect
.include '~/vsc3.1lmodels'
.lib '~/vsc3.1lcorners' typical
.unprotect
```

```
.options nomod post probe
.temp 25
```

```
.param WD=5u cl=50f vdc=2V cl2=100f WE='3*WD/2'
```

```
VDD vdd 0 DC vdc
Vback 99 0 DC 0.6V
Vin 1 0 DC 0
.global 99 vdd
```

\* O beta do inversor que aciona o buffer deve ser 5

```
JD1 vdd 2 2 99 jdep W=8u L=2u
JE1 2 1 0 99 jenh W=20u L=1u
CL1 2 0 cl
```

```
JEbuf vdd 2 3 99 jenh W=WE L=1u
JDbuf 3 0 0 99 jdep W=WD L=2u
CL2 3 0 cl2
```

\* O noh 3 eh a saida do buffer

```
*          in      out
.subckt inversor 1      2      *beta = 10
```

```
JD vdd 2 2 99 jdep W=8u L=2u
JE 2 1 0 99 jenh W=40u L=1u
```

```
.ends
```



```

*      in      out
X1      3      4      inversor
CL3 4 0 cl
X2      3      5      inversor
CL4 5 0 cl
X3      3      6      inversor
CL5 6 0 cl
X4      3      7      inversor
CL6 7 0 cl

.dc Vin 0 1 0.01
.print DC v(3) i2(jebuf) par('(i1(jd1)+i1(jebuf))*vdc')
.probe DC v(3) i2(jebuf) par('(i1(jd1)+i1(jebuf))*vdc')

.alter
.param WD=8u

.alter
.param WD=10u

.alter
.param WD=15u

.alter
.param WD=20u

.alter
.param WD=25u

.alter
.param WD=30u

.alter
.param WD=35u

.alter
.param WD=40u

.end

```

## Programa 22: “bufac2.sp”.

SDCFL WD Analysis

\* O objetivo eh estudar a influencia do WD no  
 \* desempenho transiente.

```

.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

.options nomod post probe

```

```

.temp 25

.param WD=5u

.param cl=50f    vdc=2V  WE='3*WD/2'
.param CGSP='.28*40e-15'
.param COUT='.11*40e-15 + .39*16*1e-15'
.param cl2=100f

VDD vdd 0 DC vdc
Vback 99 0 DC 0.6V
.global 99 vdd

VCLK 1 0 PULSE(Vil, Vih, td, tr, tf, pw, per)
.param Vil=0V Vih=0.7V td=400ps tr=100ps tf=100ps pw=400ps per=1ns

* O beta do inversor que aciona o buffer deve ser de 5
JD1 vdd 2 2 99 jdep W=8u L=2u
JE1 2 1 0 99 jenh W=20u L=1u
CL1 2 0 cl

.param CGSP1='.28*20e-15'
.param COUT1='.11*20e-15 + .39*16*1e-15'
CGSP1 1 0 CGSP1
CGDP1 2 1 CGSP1
COUT1 2 0 COUT1

* SDCFL com beta=3
JEbuf vdd 2 3 99 jenh W=WE L=1u
JDbuf 3 0 0 99 jdep W=WD L=2u
CL2 3 0 cl2

.param CGSPbuf='.28*WE*1e6*1e-15'
.param COUTbuf='.11*WE*1e6*1e-15 + .39*WD*2*1e6*1e-15'
CGSPEbuf 2 3 CGSPbuf
CGDPEbuf 2 vdd CGSPbuf
COUTbuf 3 0 COUTbuf
* O noh 3 eh a saida do buffer

*
      in      out
.subckt inversor      1      2      *beta = 10

JD vdd 2 2 99 jdep W=8u L=2u
JE 2 1 0 99 jenh W=40u L=1u
CGSPE 1 0 CGSP
CGDPE 2 1 CGSP
COUT 2 0 COUT

.ends

*   in   out
X1  3    4   inversor
CL3 4 0 cl
X2  3    5   inversor
CL4 5 0 cl
X3  3    6   inversor
CL5 6 0 cl

```

```
X4 3 7 inversor
CL6 7 0 cl
```

```
.tran 10ps 3ns
.probe TRAN v(1) v(3)
```

```
* calcula o swing logico da entrada
.measure tran Vpp1 param='Vih-Vil'
```

```
* calcula o ponto de 50% para o calculo do tpd
.measure tran fifty1 param='0.5*Vpp1+Vil'
```

```
* calcula o valor minimo da saida
.measure tran Vol2 avg v(3) from=1.85ns to=1.95ns
```

```
* calcula o valor maximo da saida
.measure tran Voh2 avg v(3) from=2.35ns to=2.5ns
```

```
* calcula o swing logico da saida
.measure tran Vpp2 param='Voh2-Vol2'
```

```
* calcula os pontos de 10% e 90% para calculo de tr e tf
.measure tran ten2 param='0.1*Vpp2+Vol2'
.measure tran ninety2 param='0.9*Vpp2+Vol2'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty2 param='0.5*Vpp2+Vol2'
```

```
* calcula o thl da saida do inversor
.measure tran thl trig v(3) val=ninety2 fall=2
+      targ v(3) val=ten2 fall=2
```

```
* calcula o tih da saida do inversor
.measure tran tih trig v(3) val=ten2 rise=2
+      targ v(3) val=ninety2 rise=2
```

```
* calcula o tphl
.measure tran tphl trig v(1) val=fifty1 rise=2
+      targ v(3) val=fifty2 fall=2
```

```
* calcula o tphi
.measure tran tphi trig v(1) val=fifty1 fall=2
+      targ v(3) val=fifty2 rise=2
```

```
* calcula o atraso de propagacao tpd
.measure tran tpd param='(tphl+tphi)*0.5'
```

```
.alter
.param WD=8u
```

```
.alter
.param WD=10u
```

```

.alter
.param WD=15u

.alter
.param WD=20u

.alter
.param WD=25u

.alter
.param WD=30u

.alter
.param WD=35u

.alter
.param WD=40u

.end

```

### Programa 23: “bufac.sp”.

SDCFL Fan-Out Analysis

\* O objetivo eh estudar a influencia do fan-out no  
 \* desempenho transiente.

```

.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

```

```

.options nomod post probe
.temp 25

```

```

.param cl=50fF vdc=2V
.param CGSP='.28*40e-15'
.param COUT='.11*40e-15 + .39*16*1e-15'
.param CGSP1='.28*20e-15'
.param COUT1='.11*20e-15 + .39*16*1e-15'
.param CGSPbuf='.28*30e-15'
.param COUTbuf='.11*30e-15 + .39*40*1e-15'
.param cl2=50f

```

```

VDD vdd 0 DC vdc
Vback 99 0 DC 0.6V
.global 99 vdd

```

```

VCLK 1 0 PULSE(Vil, Vih, td, tr, tf, pw, per)
.param Vil=0V Vih=0.7V td=400ps tr=100ps tf=100ps pw=400ps per=1ns

```

\* O beta do inversor que aciona o buffer deve ser de 5

```

JD1 vdd 2 2 99 jdep W=8u L=2u
JE1 2 1 0 99 jenh W=20u L=1u

```

```

CL1 2 0 cl
CGSP1 1 0 CGSP1
CGDP1 2 1 CGSP1
COUT1 2 0 COUT1

* SDCFL com beta=3
JEbuf vdd 2 3 99 jenh W=30u L=1u
JDbuf 3 0 0 99 jdep W=20u L=2u
CL2 3 0 cl2
CGSPEbuf 2 3 CGSPbuf
CGDPEbuf 2 vdd CGSPbuf
COUTbuf 3 0 COUTbuf
* 0 noh 3 eh a saida do buffer

*
      in      out
.subckt inversor 1      2      *beta = 10

JD vdd 2 2 99 jdep W=8u L=2u
JE 2 1 0 99 jenh W=40u L=1u
CGSPE 1 0 CGSP
CGDPE 2 1 CGSP
COUT 2 0 COUT

.ends

*   in      out
X1 3 4      inversor
CL3 4 0 cl

.tran 10ps 3ns
.probe TRAN v(1) v(3)

* calcula o swing logico da entrada
.measure tran Vpp1 param='Vih-Vil'

* calcula o ponto de 50% para o calculo do tpd
.measure tran fifty1 param='0.5*Vpp1+Vil'

* calcula o valor minimo da saida
.measure tran Vol2 avg v(3) from=1.7ns to=2ns

* calcula o valor maximo da saida
.measure tran Voh2 avg v(3) from=2.35ns to=2.45ns

* calcula o swing logico da saida
.measure tran Vpp2 param='Voh2-Vol2'

* calcula os pontos de 10% e 90% para calculo de tr e tf
.measure tran ten2 param='0.1*Vpp2+Vol2'
.measure tran ninety2 param='0.9*Vpp2+Vol2'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty2 param='0.5*Vpp2+Vol2'

* calcula o thl da saida do inversor
.measure tran thl trig v(3) val=ninety2 fall=2
+      targ v(3) val=ten2      fall=2

```

```

* calcula o tlh da saida do inversor
.measure tran tlh trig v(3) val=ten2      rise=2
+      targ v(3) val=ninety2 rise=2

* calcula o tphl
.measure tran tphl trig v(1) val=fifty1 rise=2
+      targ v(3) val=fifty2 fall=2

* calcula o tplh
.measure tran tplh trig v(1) val=fifty1 fall=2
+      targ v(3) val=fifty2 rise=2

* calcula o atraso de propagacao tpd
.measure tran tpd param='(tphl+tplh)*0.5'

.ALTER FO=2
*   in   out
X2   3   5   inversor
CL4 5 0 cl

.ALTER FO=3
*   in   out
X3   3   6   inversor
CL5 6 0 cl

.ALTER FO=4
*   in   out
X4   3   7   inversor
CL6 7 0 cl

.end

```

## Programa 24: “buffer2.sp”.

```

output buffer design
* Esta netlist simula o buffer de saida
* projetado para o 7490-like

.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

.temp 25
.options nopage nomod post probe

*           in   out

```

```

.subckt  buffer2  1      4
JD1 vdd 2 2 99 jdep W=50u L=2u
JD2 vdd 2 2 99 jdep W=50u L=2u
JD3 vdd 2 2 99 jdep W=50u L=2u
JD4 vdd 2 2 99 jdep W=50u L=2u
JD5 vdd 2 2 99 jdep W=50u L=2u
JD6 vdd 2 2 99 jdep W=50u L=2u
JD7 vdd 2 2 99 jdep W=50u L=2u
JD8 vdd 2 2 99 jdep W=50u L=2u
JD9 vdd 2 2 99 jdep W=50u L=2u
JD10 vdd 2 2 99 jdep W=50u L=2u

JE1  2 1 0 99 jenh W=50u L=1u
JE2  2 1 0 99 jenh W=50u L=1u
JE3  2 1 0 99 jenh W=50u L=1u
JE4  2 1 0 99 jenh W=50u L=1u
JE5  2 1 0 99 jenh W=50u L=1u
JE6  2 1 0 99 jenh W=50u L=1u
JE7  2 1 0 99 jenh W=50u L=1u
JE8  2 1 0 99 jenh W=50u L=1u
JE9  2 1 0 99 jenh W=50u L=1u
JE10 2 1 0 99 jenh W=50u L=1u
JE11 2 1 0 99 jenh W=50u L=1u
JE12 2 1 0 99 jenh W=50u L=1u
JE13 2 1 0 99 jenh W=50u L=1u
JE14 2 1 0 99 jenh W=50u L=1u
JE15 2 1 0 99 jenh W=50u L=1u
JE16 2 1 0 99 jenh W=50u L=1u
JE17 2 1 0 99 jenh W=50u L=1u
JE18 2 1 0 99 jenh W=50u L=1u
JE19  2 1 0 99 jenh W=50u L=1u
JE20  2 1 0 99 jenh W=50u L=1u
JE21  2 1 0 99 jenh W=50u L=1u
JE22  2 1 0 99 jenh W=50u L=1u
JE23  2 1 0 99 jenh W=50u L=1u
JE24  2 1 0 99 jenh W=50u L=1u
JE25  2 1 0 99 jenh W=50u L=1u

CGSPE1 1 0 350fF
CGDPE1 1 2 350fF
Cout1 2 0 527.5fF

JEbuf1 vdd 2 4 99 jenh W=50u L=1u
JEbuf2 vdd 2 4 99 jenh W=50u L=1u
JEbuf3 vdd 2 4 99 jenh W=50u L=1u
JEbuf4 vdd 2 4 99 jenh W=50u L=1u

JEbuf5 vdd 2 4 99 jenh W=50u L=1u
JEbuf6 vdd 2 4 99 jenh W=50u L=1u
JEbuf7 vdd 2 4 99 jenh W=50u L=1u
JEbuf8 vdd 2 4 99 jenh W=50u L=1u

JEbuf9  vdd 2 4 99 jenh W=50u L=1u
JEbuf10 vdd 2 4 99 jenh W=50u L=1u
JEbuf11 vdd 2 4 99 jenh W=50u L=1u
JEbuf12 vdd 2 4 99 jenh W=50u L=1u

JEbuf13 vdd 2 4 99 jenh W=50u L=1u
JEbuf14 vdd 2 4 99 jenh W=50u L=1u
JEbuf15 vdd 2 4 99 jenh W=50u L=1u
JEbuf16 vdd 2 4 99 jenh W=50u L=1u

```

```

JEbuf17 vdd 2 4 99 jenh W=50u L=1u
JEbuf18 vdd 2 4 99 jenh W=50u L=1u
JEbuf19 vdd 2 4 99 jenh W=50u L=1u
JEbuf20 vdd 2 4 99 jenh W=50u L=1u

```

```

JEbuf21 vdd 2 4 99 jenh W=50u L=1u
JEbuf22 vdd 2 4 99 jenh W=50u L=1u
JEbuf23 vdd 2 4 99 jenh W=50u L=1u
JEbuf24 vdd 2 4 99 jenh W=50u L=1u

```

```

JDbuf1 4 0 0 99 jdep W=50u L=2u
JDbuf2 4 0 0 99 jdep W=50u L=2u
JDbuf3 4 0 0 99 jdep W=50u L=2u
JDbuf4 4 0 0 99 jdep W=50u L=2u
JDbuf5 4 0 0 99 jdep W=50u L=2u
JDbuf6 4 0 0 99 jdep W=50u L=2u
JDbuf7 4 0 0 99 jdep W=50u L=2u
JDbuf8 4 0 0 99 jdep W=50u L=2u
JDbuf9 4 0 0 99 jdep W=50u L=2u
JDbuf10 4 0 0 99 jdep W=50u L=2u
JDbuf11 4 0 0 99 jdep W=50u L=2u
JDbuf12 4 0 0 99 jdep W=50u L=2u
JDbuf13 4 0 0 99 jdep W=50u L=2u
JDbuf14 4 0 0 99 jdep W=50u L=2u
JDbuf15 4 0 0 99 jdep W=50u L=2u
JDbuf16 4 0 0 99 jdep W=50u L=2u

```

```

CGSPE2 2 4 336fF
CGDPE2 2 vdd 336fF
Cout2 4 0 756fF

```

```

.ends buffer2

```

```

X1 1 2 buffer2
CL 2 0 5pF

```

```

* Frequencia de 500 MHz
.param Vil=0V Vih=0.7V td=900ps tr=100ps tf=100ps pw=900ps per=2ns
VCKA 1 0 PULSE(Vil, Vih, td, tr, tf, pw, per)

```

```

* Setup de simulacao
.global 99 vdd
.param vdc=2V
VBACKGATE 99 0 DC 0.6
VDD vdd 0 DC 2

```

```

.tran 50ps 8ns
.probe v(1) v(2) i(CL)

```

```

* calcula o swing logico da entrada
.measure tran Vpp1 param='Vih-Vil'

```

```

* calcula o ponto de 50% para o calculo do tpd
.measure tran fifty1 param='0.5*Vpp1+Vil'

```

```

* calcula o valor minimo da saida

```



```

.measure tran Vol2 avg v(2) from=5.3ns to=5.9ns

* calcula o valor maximo da saida
.measure tran Voh2 avg v(2) from=6.4ns to=6.9ns

* calcula o swing logico da saida
.measure tran Vpp2 param='Voh2-Vol2'

* calcula os pontos de 10% e 90% para calculo de tr e tf
.measure tran ten2 param='0.1*Vpp2+Vol2'
.measure tran ninety2 param='0.9*Vpp2+Vol2'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty2 param='0.5*Vpp2+Vol2'

* calcula o thl da saida do inversor
.measure tran thl trig v(2) val=ninety2 fall=4
+      targ v(2) val=ten2      fall=4

* calcula o tlh da saida do inversor
.measure tran tlh trig v(2) val=ten2      rise=3
+      targ v(2) val=ninety2 rise=3

* calcula o tphl
.measure tran tphl trig v(1) val=fifty1 rise=3
+      targ v(2) val=fifty2 fall=3

* calcula o tplh
.measure tran tplh trig v(1) val=fifty1 fall=3
+      targ v(2) val=fifty2 rise=3

* calcula o atraso de propagacao tpd
.measure tran tpd param='(tphl+tplh)*0.5'

.ALTER Slowest Case
.protect
.del lib '~/vsc3.11corners' typical
.lib '~/vsc3.11corners' slow
.unprotect
.temp 0
.param vdc=1.8V

.ALTER Highest Power Case
.protect
.del lib '~/vsc3.11corners' slow
.lib '~/vsc3.11corners' fast
.unprotect
.temp 85
.param vdc=2.2V

.end

```

## Programa 25: "FF.sp".

NOR-based, negative-edge triggered JK-MS Flip-flop

```
.protect
.include '~/vsc3.1lmodels'
.lib '~/vsc3.1lcorners' typical
.unprotect

*          in1    in2    in3    out
.subckt mast1.3    3      4      5      2    *Beta=10

J1 vdd 2 2 99 jdep w=10.4u l=2u
J21 2 3 0 99 jenh W=26u L=1u
J22 2 3 0 99 jenh W=26u L=1u
J31 2 4 0 99 jenh W=26u L=1u
J32 2 4 0 99 jenh W=26u L=1u
J41 2 5 0 99 jenh W=26u L=1u
J42 2 5 0 99 jenh W=26u L=1u

CGSPEA 3 0 14.56f
CGDPEA 2 3 14.56f
CGSPEB 4 0 14.56f
CGDPEB 2 4 14.56f
CGSPEC 5 0 14.56f
CGDPEC 2 5 14.56f
COUT   2 0 25.27f

.ends mast1.3

*          in1    in2    in3    out
.subckt mast2.3    3      4      5      2    *Beta=10

J1 vdd 2 2 99 jdep W=20.8u L=2u

J21 2 3 0 99 jenh W=26u L=1u
J22 2 3 0 99 jenh W=26u L=1u
J23 2 3 0 99 jenh W=26u L=1u
J24 2 3 0 99 jenh W=26u L=1u

J31 2 4 0 99 jenh W=26u L=1u
J32 2 4 0 99 jenh W=26u L=1u
J33 2 4 0 99 jenh W=26u L=1u
J34 2 4 0 99 jenh W=26u L=1u

J41 2 5 0 99 jenh W=26u L=1u
J42 2 5 0 99 jenh W=26u L=1u
J43 2 5 0 99 jenh W=26u L=1u
J44 2 5 0 99 jenh W=26u L=1u

CGSPEA 3 0 29.12f
CGDPEA 2 3 29.12f
CGSPEB 4 0 29.12f
CGDPEB 2 4 29.12f
CGSPEC 5 0 29.12f
```

```
CGDPEC 2 5 29.12f
COUT 2 0 50.54f
```

```
.ends mast2.3
```

```
*          in1    in2    in3    out
.subckt sla1.3 3      4      5      2    *Beta=10
```

```
J1 vdd 2 2 99 jdep W=41.6u L=2u
```

```
J21 2 3 0 99 jenh W=26u L=1u
J22 2 3 0 99 jenh W=26u L=1u
J23 2 3 0 99 jenh W=26u L=1u
J24 2 3 0 99 jenh W=26u L=1u
J25 2 3 0 99 jenh W=26u L=1u
J26 2 3 0 99 jenh W=26u L=1u
J27 2 3 0 99 jenh W=26u L=1u
J28 2 3 0 99 jenh W=26u L=1u
```

```
J31 2 4 0 99 jenh W=26u L=1u
J32 2 4 0 99 jenh W=26u L=1u
J33 2 4 0 99 jenh W=26u L=1u
J34 2 4 0 99 jenh W=26u L=1u
J35 2 4 0 99 jenh W=26u L=1u
J36 2 4 0 99 jenh W=26u L=1u
J37 2 4 0 99 jenh W=26u L=1u
J38 2 4 0 99 jenh W=26u L=1u
```

```
J41 2 5 0 99 jenh W=26u L=1u
J42 2 5 0 99 jenh W=26u L=1u
J43 2 5 0 99 jenh W=26u L=1u
J44 2 5 0 99 jenh W=26u L=1u
J45 2 5 0 99 jenh W=26u L=1u
J46 2 5 0 99 jenh W=26u L=1u
J47 2 5 0 99 jenh W=26u L=1u
J48 2 5 0 99 jenh W=26u L=1u
```

```
CGSPEA 3 0 58.24f
CGDPEA 2 3 58.24f
CGSPEB 4 0 58.24f
CGDPEB 2 4 58.24f
CGSPEC 5 0 58.24f
CGDPEC 2 5 58.24f
COUT 2 0 101f
```

```
.ends sla1.3
```

```
*          in1    in2    in3    out
.subckt sla2.3 3      4      5      2    *Beta=10
```

```
J11 vdd 2 2 99 jdep W=41.6u L=2u
J12 vdd 2 2 99 jdep W=41.6u L=2u
```

```
J21 2 3 0 99 jenh W=41.6u L=1u
J22 2 3 0 99 jenh W=41.6u L=1u
J23 2 3 0 99 jenh W=41.6u L=1u
J24 2 3 0 99 jenh W=41.6u L=1u
```

```
J25 2 3 0 99 jenh W=41.6u L=1u
J26 2 3 0 99 jenh W=41.6u L=1u
J27 2 3 0 99 jenh W=41.6u L=1u
J28 2 3 0 99 jenh W=41.6u L=1u
J29 2 3 0 99 jenh W=41.6u L=1u
J210 2 3 0 99 jenh W=41.6u L=1u
```

```
J31 2 4 0 99 jenh W=41.6u L=1u
J32 2 4 0 99 jenh W=41.6u L=1u
J33 2 4 0 99 jenh W=41.6u L=1u
J34 2 4 0 99 jenh W=41.6u L=1u
J35 2 4 0 99 jenh W=41.6u L=1u
J36 2 4 0 99 jenh W=41.6u L=1u
J37 2 4 0 99 jenh W=41.6u L=1u
J38 2 4 0 99 jenh W=41.6u L=1u
J39 2 4 0 99 jenh W=41.6u L=1u
J310 2 4 0 99 jenh W=41.6u L=1u
```

```
J41 2 5 0 99 jenh W=41.6u L=1u
J42 2 5 0 99 jenh W=41.6u L=1u
J43 2 5 0 99 jenh W=41.6u L=1u
J44 2 5 0 99 jenh W=41.6u L=1u
J45 2 5 0 99 jenh W=41.6u L=1u
J46 2 5 0 99 jenh W=41.6u L=1u
J47 2 5 0 99 jenh W=41.6u L=1u
J48 2 5 0 99 jenh W=41.6u L=1u
J49 2 5 0 99 jenh W=41.6u L=1u
J410 2 5 0 99 jenh W=41.6u L=1u
```

```
CGSPEA 3 0 58.24f
CGDPEA 2 3 58.24f
CGSPEB 4 0 58.24f
CGDPEB 2 4 58.24f
CGSPEC 5 0 58.24f
CGDPEC 2 5 58.24f
COUT 2 0 133.5f
```

```
.ends sla2.3
```

```
*          in  out
.subckt clkbuffer 1 2
Jck11 vdd 2 2 99 jdep W=30u L=2u
Jck12 vdd 2 2 99 jdep W=30u L=2u
```

```
Jck21 2 1 0 99 jenh W=30u L=1u
Jck22 2 1 0 99 jenh W=30u L=1u
Jck23 2 1 0 99 jenh W=30u L=1u
Jck24 2 1 0 99 jenh W=30u L=1u
Jck25 2 1 0 99 jenh W=30u L=1u
Jck26 2 1 0 99 jenh W=30u L=1u
Jck27 2 1 0 99 jenh W=30u L=1u
Jck28 2 1 0 99 jenh W=30u L=1u
Jck29 2 1 0 99 jenh W=30u L=1u
Jck210 2 1 0 99 jenh W=30u L=1u
```

```
CGSPE 1 0 84fF
CGDPE 1 2 84fF
Cout 2 0 79.8fF
```

```
.ends clkbuffer
```

```
***** Flip-flop *****
```

```
*          CLK    J    K    Q    QN    PR    CL
.subckt FF1      1    7    12   4     2    17    18
```

```
* CLK inverter
* in out
X10 1 5 clkbuffer
```

```
* MASTER SECTION
```

```
* in1 in2 in3 out
X1 5 2 7 6 mast1.3
X2 5 4 12 9 mast1.3
```

```
* in1 in2 in3 out
X3 6 11 18 10 mast2.3
X4 9 10 17 11 mast2.3
```

```
* SLAVE SECTION
```

```
* in1 in2 in3 out
X5 1 10 17 13 sla1.3
X6 1 11 18 14 sla1.3
```

```
* in1 in2 in3 out
X7 13 2 18 4 sla2.3
X8 14 4 17 2 sla2.3
```

```
.ends FF1
```

```
*****
* ***** CIRCUITO PRINCIPAL *****
*****
```

```
* CLK    J    K    Q    QN    PR    CL
X1 1 2 3 5 6 7 4 FF1
```

```
* Setup de simulacao
```

```
.global 99 vdd
.param vdc=2V
VBACKGATE 99 0 DC 0.6
VDD vdd 0 DC vdc
```

```
* Frequencia de 1 GHz
```

```
.param Vil=0V Vih=0.7 td=400ps tr=100ps tf=100ps pw=400ps per=1ns
VCLK 1 0 PULSE(Vil, Vih, td, tr, tf, pw, per)
```

```
*Entradas J e K
```

```
VJ 2 0 PULSE(Vil, Vih, 4ns, tr, tf, 8ns, 16ns)
VK 3 0 PULSE(Vil, Vih, 8ns, tr, tf, 8ns, 16ns)
```

```
*Entrada de clear
```

```

VCL 4 0 PULSE(Vil, Vih, 0ps, tr, tf, 300ps, 1s)
VPR 7 0 DC 0

.temp 25
.options nopage nomod ingold=0 post probe

.param cl=100f
C1 5 0 cl
C2 6 0 cl

.tran 50ps 20ns
.probe v(1) v(2) v(3) v(5)
.op 9.8ns

* calcula o swing logico da entrada
.measure tran Vpp1 param='Vih-Vil'

* calcula o ponto de 50% para o calculo do tpd
.measure tran fifty1 param='0.5*Vpp1+Vil'

* calcula o valor minimo da saida
.measure tran Vol2 avg v(5) from=13.5ns to=17ns

* calcula o valor maximo da saida
.measure tran Voh2 avg v(5) from=5.5ns to=13ns

* calcula o swing logico da saida
.measure tran Vpp2 param='Voh2-Vol2'

* calcula os pontos de 10% e 90% para calculo de tr e tf
.measure tran ten2 param='0.1*Vpp2+Vol2'
.measure tran ninety2 param='0.9*Vpp2+Vol2'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty2 param='0.5*Vpp2+Vol2'

* calcula o tphl
.measure tran tphl trig v(1) val=fifty1 fall=13
+           targ v(5) val=fifty2 fall=3

* calcula o tplh
.measure tran tplh trig v(1) val=fifty1 fall=5
+           targ v(5) val=fifty2 rise=3

* calcula o atraso de propagacao tpd
.measure tran tpd param='(tphl+tplh)*0.5'

.ALTER Slowest Case
.protect
.del lib '~/vsc3.11corners' typical
.lib '~/vsc3.11corners' slow
.unprotect
.temp 0
.param vdc=1.8V

```

```
.ALTER Highest Power Case
.protect
.del lib '~/vsc3.11corners' slow
.lib '~/vsc3.11corners' fast
.unprotect
.temp 85
.param vdc=2.2V

.end
```

## Programa 26: “decnt.sp”.

Decade Counter 7490-like

```
.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' typical
.unprotect

*           in1    in2    out    * beta=10
.subckt mast1.2    3      4      2

J1 0 2 2 99 jdep w=10.4u l=2u
J21 2 3 vtt 99 jenh w=26u l=1u
J22 2 3 vtt 99 jenh w=26u l=1u
J31 2 4 vtt 99 jenh w=26u l=1u
J32 2 4 vtt 99 jenh w=26u l=1u

CGSPEA 3 vtt 14.56f
CGDPEA 2 3 14.56f
CGSPEB 4 vtt 14.56f
CGDPEB 2 4 14.56f
COUT 2 vtt 19.55f

.ends mast1.2

*           in1    in2    in3    out    *Beta=10
.subckt mast1.3    3      4      5      2

J1 0 2 2 99 jdep w=10.4u l=2u
J21 2 3 vtt 99 jenh W=26u L=1u
J22 2 3 vtt 99 jenh W=26u L=1u
J31 2 4 vtt 99 jenh W=26u L=1u
J32 2 4 vtt 99 jenh W=26u L=1u
J41 2 5 vtt 99 jenh W=26u L=1u
J42 2 5 vtt 99 jenh W=26u L=1u

CGSPEA 3 vtt 14.56f
CGDPEA 2 3 14.56f
CGSPEB 4 vtt 14.56f
CGDPEB 2 4 14.56f
```

```

CGSPEC 5 vtt 14.56f
CGDPEC 2 5 14.56f
COUT 2 vtt 25.27f

```

```

.ends mast1.3

```

```

*           in1    in2    out
.subckt mast2.2    3      4      2    *Beta=10

```

```

J1 0 2 2 99 jdep W=20.8u L=2u

```

```

J21 2 3 vtt 99 jenh W=26u L=1u
J22 2 3 vtt 99 jenh W=26u L=1u
J23 2 3 vtt 99 jenh W=26u L=1u
J24 2 3 vtt 99 jenh W=26u L=1u

```

```

J31 2 4 vtt 99 jenh W=26u L=1u
J32 2 4 vtt 99 jenh W=26u L=1u
J33 2 4 vtt 99 jenh W=26u L=1u
J34 2 4 vtt 99 jenh W=26u L=1u

```

```

CGSPEA 3 vtt 29.12f
CGDPEA 2 3 29.12f
CGSPEB 4 vtt 29.12f
CGDPEB 2 4 29.12f
COUT 2 vtt 39.10f

```

```

.ends mast2.2

```

```

*           in1    in2    in3    out
.subckt mast2.3    3      4      5      2    *Beta=10

```

```

J1 0 2 2 99 jdep W=20.8u L=2u

```

```

J21 2 3 vtt 99 jenh W=26u L=1u
J22 2 3 vtt 99 jenh W=26u L=1u
J23 2 3 vtt 99 jenh W=26u L=1u
J24 2 3 vtt 99 jenh W=26u L=1u

```

```

J31 2 4 vtt 99 jenh W=26u L=1u
J32 2 4 vtt 99 jenh W=26u L=1u
J33 2 4 vtt 99 jenh W=26u L=1u
J34 2 4 vtt 99 jenh W=26u L=1u

```

```

J41 2 5 vtt 99 jenh W=26u L=1u
J42 2 5 vtt 99 jenh W=26u L=1u
J43 2 5 vtt 99 jenh W=26u L=1u
J44 2 5 vtt 99 jenh W=26u L=1u

```

```

CGSPEA 3 vtt 29.12f
CGDPEA 2 3 29.12f
CGSPEB 4 vtt 29.12f
CGDPEB 2 4 29.12f
CGSPEC 5 vtt 29.12f
CGDPEC 2 5 29.12f
COUT 2 vtt 50.54f

```



.ends mast2.3

```
*          in1    in2    out
.subckt slal.2    3      4      2    *Beta=10
```

J1 0 2 2 99 jdep W=41.6u L=2u

J21 2 3 vtt 99 jenh W=26u L=1u  
J22 2 3 vtt 99 jenh W=26u L=1u  
J23 2 3 vtt 99 jenh W=26u L=1u  
J24 2 3 vtt 99 jenh W=26u L=1u  
J25 2 3 vtt 99 jenh W=26u L=1u  
J26 2 3 vtt 99 jenh W=26u L=1u  
J27 2 3 vtt 99 jenh W=26u L=1u  
J28 2 3 vtt 99 jenh W=26u L=1u

J31 2 4 vtt 99 jenh W=26u L=1u  
J32 2 4 vtt 99 jenh W=26u L=1u  
J33 2 4 vtt 99 jenh W=26u L=1u  
J34 2 4 vtt 99 jenh W=26u L=1u  
J35 2 4 vtt 99 jenh W=26u L=1u  
J36 2 4 vtt 99 jenh W=26u L=1u  
J37 2 4 vtt 99 jenh W=26u L=1u  
J38 2 4 vtt 99 jenh W=26u L=1u

CGSPEA 3 vtt 58.24f  
CGDPEA 2 3 58.24f  
CGSPEB 4 vtt 58.24f  
CGDPEB 2 4 58.24f  
COUT 2 vtt 78.21f

.ends slal.2

```
*          in1    in2    in3    out
.subckt slal.3    3      4      5      2    *Beta=10
```

J1 0 2 2 99 jdep W=41.6u L=2u

J21 2 3 vtt 99 jenh W=26u L=1u  
J22 2 3 vtt 99 jenh W=26u L=1u  
J23 2 3 vtt 99 jenh W=26u L=1u  
J24 2 3 vtt 99 jenh W=26u L=1u  
J25 2 3 vtt 99 jenh W=26u L=1u  
J26 2 3 vtt 99 jenh W=26u L=1u  
J27 2 3 vtt 99 jenh W=26u L=1u  
J28 2 3 vtt 99 jenh W=26u L=1u

J31 2 4 vtt 99 jenh W=26u L=1u  
J32 2 4 vtt 99 jenh W=26u L=1u  
J33 2 4 vtt 99 jenh W=26u L=1u  
J34 2 4 vtt 99 jenh W=26u L=1u  
J35 2 4 vtt 99 jenh W=26u L=1u  
J36 2 4 vtt 99 jenh W=26u L=1u  
J37 2 4 vtt 99 jenh W=26u L=1u  
J38 2 4 vtt 99 jenh W=26u L=1u

```
J41 2 5 vtt 99 jenh W=26u L=1u
J42 2 5 vtt 99 jenh W=26u L=1u
J43 2 5 vtt 99 jenh W=26u L=1u
J44 2 5 vtt 99 jenh W=26u L=1u
J45 2 5 vtt 99 jenh W=26u L=1u
J46 2 5 vtt 99 jenh W=26u L=1u
J47 2 5 vtt 99 jenh W=26u L=1u
J48 2 5 vtt 99 jenh W=26u L=1u
```

```
CGSPEA 3 vtt 58.24f
CGDPEA 2 3 58.24f
CGSPEB 4 vtt 58.24f
CGDPEB 2 4 58.24f
CGSPEC 5 vtt 58.24f
CGDPEC 2 5 58.24f
COUT 2 vtt 101f
```

```
.ends sla1.3
```

```
*          in1    in2    out
.subckt sla2.2 3      4      2
J11 0 2 2 99 jdep W=41.6u L=2u
J12 0 2 2 99 jdep W=41.6u L=2u
```

```
*Beta=5 porque vai acionar buffer
```

```
J21 2 3 vtt 99 jenh W=41.6u L=1u
J22 2 3 vtt 99 jenh W=41.6u L=1u
J23 2 3 vtt 99 jenh W=41.6u L=1u
J24 2 3 vtt 99 jenh W=41.6u L=1u
J25 2 3 vtt 99 jenh W=41.6u L=1u
```

```
J31 2 4 vtt 99 jenh W=41.6u L=1u
J32 2 4 vtt 99 jenh W=41.6u L=1u
J33 2 4 vtt 99 jenh W=41.6u L=1u
J34 2 4 vtt 99 jenh W=41.6u L=1u
J35 2 4 vtt 99 jenh W=41.6u L=1u
```

```
CGSPEA 3 vtt 58.24f
CGDPEA 2 3 58.24f
CGSPEB 4 vtt 58.24f
CGDPEB 2 4 58.24f
COUT 2 vtt 110.7f
```

```
.ends sla2.2
```

```
*          in1    in2    in3    out
.subckt sla2.3 3      4      5      2
J11 0 2 2 99 jdep W=41.6u L=2u
J12 0 2 2 99 jdep W=41.6u L=2u
```

```
*Beta=5 porque vai acionar
```

```
J21 2 3 vtt 99 jenh W=41.6u L=1u
J22 2 3 vtt 99 jenh W=41.6u L=1u
J23 2 3 vtt 99 jenh W=41.6u L=1u
J24 2 3 vtt 99 jenh W=41.6u L=1u
J25 2 3 vtt 99 jenh W=41.6u L=1u
```

```
J31 2 4 vtt 99 jenh W=41.6u L=1u
J32 2 4 vtt 99 jenh W=41.6u L=1u
J33 2 4 vtt 99 jenh W=41.6u L=1u
J34 2 4 vtt 99 jenh W=41.6u L=1u
J35 2 4 vtt 99 jenh W=41.6u L=1u
```

```
J41 2 5 vtt 99 jenh W=41.6u L=1u
J42 2 5 vtt 99 jenh W=41.6u L=1u
J43 2 5 vtt 99 jenh W=41.6u L=1u
J44 2 5 vtt 99 jenh W=41.6u L=1u
J45 2 5 vtt 99 jenh W=41.6u L=1u
```

```
CGSPEA 3 vtt 58.24f
CGDPEA 2 3 58.24f
CGSPEB 4 vtt 58.24f
CGDPEB 2 4 58.24f
CGSPEC 5 vtt 58.24f
CGDPEC 2 5 58.24f
COUT 2 vtt 133.5f
```

```
.ends sla2.3
```

```
*          in1    in2    out
.subckt slaPRCL 3      4      2      *Beta=5
J11 0 2 2 99 jdep W=50u L=2u
J12 0 2 2 99 jdep W=50u L=2u
```

```
J21 2 3 vtt 99 jenh W=50u L=1u
J22 2 3 vtt 99 jenh W=50u L=1u
J23 2 3 vtt 99 jenh W=50u L=1u
J24 2 3 vtt 99 jenh W=50u L=1u
J25 2 3 vtt 99 jenh W=50u L=1u
```

```
J31 2 4 vtt 99 jenh W=50u L=1u
J32 2 4 vtt 99 jenh W=50u L=1u
J33 2 4 vtt 99 jenh W=50u L=1u
J34 2 4 vtt 99 jenh W=50u L=1u
J35 2 4 vtt 99 jenh W=50u L=1u
```

```
CGSPEA 3 vtt 70f
CGDPEA 2 3 70f
CGSPEB 4 vtt 70f
CGDPEB 2 4 70f
COUT 2 vtt 133f
```

```
.ends slaPRCL
```

```
*          in    out
.subckt buffer 2      4
JEbuf1 0 2 4 99 jenh W=50u L=1u
JEbuf2 0 2 4 99 jenh W=50u L=1u
JEbuf3 0 2 4 99 jenh W=50u L=1u
JEbuf4 0 2 4 99 jenh W=50u L=1u
JEbuf5 0 2 4 99 jenh W=50u L=1u
JEbuf6 0 2 4 99 jenh W=50u L=1u
JEbuf7 0 2 4 99 jenh W=50u L=1u
JEbuf8 0 2 4 99 jenh W=50u L=1u
```

```

JDbuf1  4  vtt vtt 99 jdep W=50u L=2u
JDbuf2  4  vtt vtt 99 jdep W=50u L=2u
JDbuf3  4  vtt vtt 99 jdep W=50u L=2u
JDbuf4  4  vtt vtt 99 jdep W=50u L=2u
JDbuf5  4  vtt vtt 99 jdep W=50u L=2u
JDbuf6  4  vtt vtt 99 jdep W=50u L=2u
JDbuf7  4  vtt vtt 99 jdep W=50u L=2u
JDbuf8  4  vtt vtt 99 jdep W=50u L=2u

```

```

CGSPE 2 4 112f
CGDPE 2 0 112f
Cout 4 vtt 356f
.ends buffer

```

```

*
      in  out
.subckt buffer3 2 4
JEbuf1 0 2 4 99 jenh W=45u L=1u
JEbuf2 0 2 4 99 jenh W=45u L=1u
JEbuf3 0 2 4 99 jenh W=45u L=1u
JEbuf4 0 2 4 99 jenh W=45u L=1u
JEbuf5 0 2 4 99 jenh W=45u L=1u
JEbuf6 0 2 4 99 jenh W=45u L=1u
JEbuf7 0 2 4 99 jenh W=45u L=1u
JEbuf8 0 2 4 99 jenh W=45u L=1u

```

```

JDbuf1  4  vtt vtt 99 jdep W=50u L=2u
JDbuf2  4  vtt vtt 99 jdep W=50u L=2u
JDbuf3  4  vtt vtt 99 jdep W=50u L=2u
JDbuf4  4  vtt vtt 99 jdep W=50u L=2u
JDbuf5  4  vtt vtt 99 jdep W=50u L=2u
JDbuf6  4  vtt vtt 99 jdep W=50u L=2u
JDbuf7  4  vtt vtt 99 jdep W=50u L=2u
JDbuf8  4  vtt vtt 99 jdep W=50u L=2u

```

```

CGSPE 2 4 101f
CGDPE 2 0 101f
Cout 4 vtt 351.6f
.ends buffer3

```

```

*
      in  out
.subckt buffer4 2 4      *Mais leve que buffer3
JEbuf1 0 2 4 99 jenh W=45u L=1u
JEbuf2 0 2 4 99 jenh W=45u L=1u
JEbuf3 0 2 4 99 jenh W=45u L=1u
JEbuf4 0 2 4 99 jenh W=45u L=1u
JEbuf5 0 2 4 99 jenh W=45u L=1u
JEbuf6 0 2 4 99 jenh W=45u L=1u
JEbuf7 0 2 4 99 jenh W=45u L=1u

```

```

JDbuf1  4  vtt vtt 99 jdep W=50u L=2u
JDbuf2  4  vtt vtt 99 jdep W=50u L=2u
JDbuf3  4  vtt vtt 99 jdep W=50u L=2u
JDbuf4  4  vtt vtt 99 jdep W=50u L=2u
JDbuf5  4  vtt vtt 99 jdep W=50u L=2u
JDbuf6  4  vtt vtt 99 jdep W=50u L=2u

```

```

JDbuf7  4 vtt vtt 99 jdep W=50u L=2u
JDbuf8  4 vtt vtt 99 jdep W=50u L=2u

```

```

CGSPE 2 4 88,2f
CGDPE 2 0 88,2f
Cout 4 vtt 346,7f
.ends buffer4

```

```

*           in   out
.subckt  buffer2  1   4
JD1 0 2 2 99 jdep W=50u L=2u
JD2 0 2 2 99 jdep W=50u L=2u
JD3 0 2 2 99 jdep W=50u L=2u
JD4 0 2 2 99 jdep W=50u L=2u
JD5 0 2 2 99 jdep W=50u L=2u
JD6 0 2 2 99 jdep W=50u L=2u
JD7 0 2 2 99 jdep W=50u L=2u
JD8 0 2 2 99 jdep W=50u L=2u
JD9 0 2 2 99 jdep W=50u L=2u
JD10 0 2 2 99 jdep W=50u L=2u

```

```

JE1  2 1 vtt 99 jenh W=50u L=1u
JE2  2 1 vtt 99 jenh W=50u L=1u
JE3  2 1 vtt 99 jenh W=50u L=1u
JE4  2 1 vtt 99 jenh W=50u L=1u
JE5  2 1 vtt 99 jenh W=50u L=1u
JE6  2 1 vtt 99 jenh W=50u L=1u
JE7  2 1 vtt 99 jenh W=50u L=1u
JE8  2 1 vtt 99 jenh W=50u L=1u
JE9  2 1 vtt 99 jenh W=50u L=1u
JE10 2 1 vtt 99 jenh W=50u L=1u
JE11 2 1 vtt 99 jenh W=50u L=1u
JE12 2 1 vtt 99 jenh W=50u L=1u
JE13 2 1 vtt 99 jenh W=50u L=1u
JE14 2 1 vtt 99 jenh W=50u L=1u
JE15 2 1 vtt 99 jenh W=50u L=1u
JE16 2 1 vtt 99 jenh W=50u L=1u
JE17 2 1 vtt 99 jenh W=50u L=1u
JE18 2 1 vtt 99 jenh W=50u L=1u
JE19 2 1 vtt 99 jenh W=50u L=1u
JE20 2 1 vtt 99 jenh W=50u L=1u
JE21 2 1 vtt 99 jenh W=50u L=1u
JE22 2 1 vtt 99 jenh W=50u L=1u
JE23 2 1 vtt 99 jenh W=50u L=1u
JE24 2 1 vtt 99 jenh W=50u L=1u
JE25 2 1 vtt 99 jenh W=50u L=1u

```

```

CGSPE1 1 vtt 350fF
CGDPE1 1 2 350fF
Cout1 2 vtt 527.5fF

```

```

JEbuf1 0 2 4 99 jenh W=50u L=1u
JEbuf2 0 2 4 99 jenh W=50u L=1u
JEbuf3 0 2 4 99 jenh W=50u L=1u
JEbuf4 0 2 4 99 jenh W=50u L=1u

```

```

JEbuf5 0 2 4 99 jenh W=50u L=1u
JEbuf6 0 2 4 99 jenh W=50u L=1u

```

```

JEbuf7 0 2 4 99 jenh W=50u L=1u
JEbuf8 0 2 4 99 jenh W=50u L=1u

JEbuf9 0 2 4 99 jenh W=50u L=1u
JEbuf10 0 2 4 99 jenh W=50u L=1u
JEbuf11 0 2 4 99 jenh W=50u L=1u
JEbuf12 0 2 4 99 jenh W=50u L=1u

JEbuf13 0 2 4 99 jenh W=50u L=1u
JEbuf14 0 2 4 99 jenh W=50u L=1u
JEbuf15 0 2 4 99 jenh W=50u L=1u
JEbuf16 0 2 4 99 jenh W=50u L=1u

JEbuf17 0 2 4 99 jenh W=50u L=1u
JEbuf18 0 2 4 99 jenh W=50u L=1u
JEbuf19 0 2 4 99 jenh W=50u L=1u
JEbuf20 0 2 4 99 jenh W=50u L=1u

JEbuf21 0 2 4 99 jenh W=50u L=1u
JEbuf22 0 2 4 99 jenh W=50u L=1u
JEbuf23 0 2 4 99 jenh W=50u L=1u
JEbuf24 0 2 4 99 jenh W=50u L=1u

```

```

JDbuf1 4 vtt vtt 99 jdep W=50u L=2u
JDbuf2 4 vtt vtt 99 jdep W=50u L=2u
JDbuf3 4 vtt vtt 99 jdep W=50u L=2u
JDbuf4 4 vtt vtt 99 jdep W=50u L=2u
JDbuf5 4 vtt vtt 99 jdep W=50u L=2u
JDbuf6 4 vtt vtt 99 jdep W=50u L=2u
JDbuf7 4 vtt vtt 99 jdep W=50u L=2u
JDbuf8 4 vtt vtt 99 jdep W=50u L=2u
JDbuf9 4 vtt vtt 99 jdep W=50u L=2u
JDbuf10 4 vtt vtt 99 jdep W=50u L=2u
JDbuf11 4 vtt vtt 99 jdep W=50u L=2u
JDbuf12 4 vtt vtt 99 jdep W=50u L=2u
JDbuf13 4 vtt vtt 99 jdep W=50u L=2u
JDbuf14 4 vtt vtt 99 jdep W=50u L=2u
JDbuf15 4 vtt vtt 99 jdep W=50u L=2u
JDbuf16 4 vtt vtt 99 jdep W=50u L=2u

```

```

CGSPE2 2 4 336fF
CGDPE2 2 0 336fF
Cout2 4 vtt 756fF

```

```

.ends buffer2

```

```

*
.subckt inbuffer 1 2

```

```

Rin 1 vtt 250

```

```

jd11 1 2 2 99 jdep w=50u l=2u
jd12 1 2 2 99 jdep w=50u l=2u
jd13 1 2 2 99 jdep w=50u l=2u
jd14 1 2 2 99 jdep w=50u l=2u
jd15 1 2 2 99 jdep w=50u l=2u
jd16 1 2 2 99 jdep w=50u l=2u

```

```
jd17 1 2 2 99 jdep w=50u l=2u
jd18 1 2 2 99 jdep w=50u l=2u
jd19 1 2 2 99 jdep w=50u l=2u
```

```
Cdspd 1 2 351fF
.ends inbuffer
```

```
*
      in out
.subckt clkbuffer 1 2
J11 0 2 2 99 jdep W=30u L=2u
J12 0 2 2 99 jdep W=30u L=2u

J21 2 1 vtt 99 jenh W=30u L=1u
J22 2 1 vtt 99 jenh W=30u L=1u
J23 2 1 vtt 99 jenh W=30u L=1u
J24 2 1 vtt 99 jenh W=30u L=1u
J25 2 1 vtt 99 jenh W=30u L=1u
J26 2 1 vtt 99 jenh W=30u L=1u
J27 2 1 vtt 99 jenh W=30u L=1u
J28 2 1 vtt 99 jenh W=30u L=1u
J29 2 1 vtt 99 jenh W=30u L=1u
J210 2 1 vtt 99 jenh W=30u L=1u
```

```
CGSPE 1 vtt 84f
CGDPE 2 1 84f
COUT 2 vtt 79.8f
.ends clkbuffer
```

```
*
      in out
.subckt clkbuffer2 1 3 *Mais forte que clkbuffer
JD1 0 2 2 99 jdep W=40u L=2u
JD2 0 2 2 99 jdep W=40u L=2u
JD3 0 2 2 99 jdep W=40u L=2u
JD4 0 2 2 99 jdep W=40u L=2u
JD5 0 2 2 99 jdep W=40u L=2u

JE1 2 1 vtt 99 jenh W=50u L=1u
JE2 2 1 vtt 99 jenh W=50u L=1u
JE3 2 1 vtt 99 jenh W=50u L=1u
JE4 2 1 vtt 99 jenh W=50u L=1u
JE5 2 1 vtt 99 jenh W=50u L=1u
JE6 2 1 vtt 99 jenh W=50u L=1u
```

```
CGSPE 1 vtt 84f
CGDPE 2 1 84f
COUT 2 vtt 189f
```

```
JEbuf1 0 2 3 99 jenh W=55u L=1u
JEbuf2 0 2 3 99 jenh W=55u L=1u
JEbuf3 0 2 3 99 jenh W=55u L=1u
JEbuf4 0 2 3 99 jenh W=55u L=1u
JEbuf5 0 2 3 99 jenh W=55u L=1u
JEbuf6 0 2 3 99 jenh W=55u L=1u
JEbuf7 0 2 3 99 jenh W=55u L=1u

JDbuf1 3 vtt vtt 99 jdep W=50u L=2u
```

```

JDbuf2 3 vtt vtt 99 jdep W=50u L=2u
JDbuf3 3 vtt vtt 99 jdep W=50u L=2u
JDbuf4 3 vtt vtt 99 jdep W=50u L=2u
JDbuf5 3 vtt vtt 99 jdep W=50u L=2u
JDbuf6 3 vtt vtt 99 jdep W=50u L=2u
JDbuf7 3 vtt vtt 99 jdep W=50u L=2u
JDbuf8 3 vtt vtt 99 jdep W=50u L=2u
JDbuf9 3 vtt vtt 99 jdep W=50u L=2u
JDbuf10 3 vtt vtt 99 jdep W=50u L=2u

```

```

CGSPE2 2 3 108fF
CGDPE2 2 0 108fF
Cout2 4 vtt 432.3fF
.ends clkbuffer2

```

\*\*\*\*\* Flip-flops \*\*\*\*\*

```

*          CLK      QN      PR      CL
.subckt FF1      1      3      17      18

```

```

* CLK inverter
* in out
X10 1 5 clkbuffer

```

\* MASTER SECTION

```

* in1 in2 out
X1 5 3 6 mast1.2
X2 5 8 9 mast1.2

```

```

* in1 in2 in3 out
X3 6 11 18 10 mast2.3
X4 9 10 17 11 mast2.3

```

\* SLAVE SECTION

```

* in1 in2 in3 out
X5 1 10 17 13 sla1.3
X6 1 11 18 14 sla1.3

```

```

* in1 in2 in3 out
X7 13 3 18 4 sla2.3
X8 14 8 17 2 sla2.3

```

```

* in out
X9 2 3 buffer
X11 4 8 buffer4
.ends FF1

```

```

*          CLK      J      Q      QN      PR
.subckt FF2      1      7      8      3      17

```

```

* CLK inverter
* in out
X10 1 5 clkbuffer

```



```

* MASTER SECTION
*   in1   in2   in3   out
X1    5     3     7     6     mast1.3
*   in1   in2   out
X2    5     8     9           mast1.2

*   in1   in2   out
X3    6    11    10           mast2.2
*   in1   in2   in3   out
X4   10     9    17    11     mast2.3

```

```

* SLAVE SECTION
*   in1   in2   in3   out
X5     1    10    17    13     sla1.3
*   in1   in2   out
X6     1    11    14           sla1.2

*   in1   in2   out
X7    13     3     4           sla2.2
*   in1   in2   in3   out
X8     8    14    17     2     sla2.3

```

```

*   in out
X9     2  3   buffer   *Saida QN
X11    4  8   buffer   *Saida Q

```

.ends FF2

```

*           CLK   QN   CL
.subckt FF3  1     3   18

```

```

* CLK inverter
*   in out
X10 1  5 clkbuffer

```

```

* MASTER SECTION
*   in1   in2   out
X1     5     3     6     mast1.2
X2     5     8     9     mast1.2

*   in1   in2   in3   out
X3     6    11    18    10     mast2.3
*   in1   in2   out
X4     9    10    11           mast2.2

```

```

* Slave section
*   in1   in2   out
X5     1    10    13     sla1.2
*   in1   in2   in3   out
X6     1    11    18    14     sla1.3

*   in1   in2   in3   out
X7    18    13     3     4     sla2.3
*   in1   in2   out
X8     8    14     2           sla2.2

```

```

*   in  out
X9   2   3   buffer
X11  4   8   buffer4

```

```

.ends FF3

```

```

*           CLK      R      Q      QN      PR      CL
.subckt FF4   1       7      8      3      17      18

```

```

* CLK inverter
*   in  out
X10 1   5 clkbuffer

```

```

* MASTER SECTION

```

```

*   in1   in2   out
X1   5     3     6   mast1.2
X2   5     7     9   mast1.2

```

```

*   in1   in2   in3   out
X3   6    11    18    10   mast2.3
X4  10     9    17    11   mast2.3

```

```

* Slave section

```

```

*   in1   in2   in3   out
X5   1    10    17    13   sla1.3
X6   1    11    18    14   sla1.3

```

```

*   in1   in2   in3   out
X7   18    13     3     4   sla2.3
X8   8     14    17     2   sla2.3

```

```

*   in  out
X9   2   3   buffer
X11  4   8   buffer3

```

```

.ends FF4

```

```

*****
***** CIRCUITO PRINCIPAL *****
*****

```

```

*   CLK      QN      PR      CL
X1   15      31      4      13   FF1

```

```

*   CLK      J      Q      QN      PR
X2   5      23     32     21     16   FF2

```

```

*   CLK      QN      CL
X3   21     33     16   FF3

```

```

*   CLK      R      Q      QN      PR      CL

```

```
X4    5      10      23      34      4      13      FF4
```

```
* Setup de simulacao
```

```
.param vdc=-2V
```

```
.global 99 vtt
```

```
VBACKGATE 99 vtt DC 0.6
```

```
VTT vtt 0 DC vdc
```

```
* Frequencia de 1 GHz
```

```
.param Vil=0V Vih=1.2V td=400ps tr=100ps tf=100ps pw=400ps per=1ns
```

```
.param fifty='vdc+0.5*(Vih-Vil)'
```

```
VCKA 14 vtt PULSE(Vil, Vih, td, tr, tf, pw, per)
```

```
X5 14 15 inbuffer
```

```
*VCKB 1 vtt PULSE(Vil, Vih, td, tr, tf, pw, per)
```

```
X6 12 5 inbuffer
```

```
*A ENTRADA DE X6 EH A SAIDA QA PARA A SIMULACAO DA CONTAGEM BCD.
```

```
* Porta de Clear
```

```
VCL 2 vtt PULSE(Vil, Vih, 0ps, tr, tf, 300ps, 1s)
```

```
X7 2 20 inbuffer
```

```
X70 2 30 inbuffer
```

```
* in out
```

```
X9 20 25 clkbuffer
```

```
X10 30 29 clkbuffer
```

```
* in1 in2 out
```

```
X8 25 29 27 slaPRCL
```

```
X72 27 13 buffer
```

```
* Porta de Preset (desabilitada)
```

```
X11 vtt 17 inbuffer
```

```
X71 vtt 19 inbuffer
```

```
* in out
```

```
X13 17 18 clkbuffer
```

```
X14 19 24 clkbuffer
```

```
* in1 in2 out
```

```
X12 18 24 28 slaPRCL
```

```
X73 28 4 buffer
```

```
* Porta de PR de X2 e CL de X3
```

```
* in1 in2 out
```

```
X15 13 4 22 sla1.2
```

```
X69 22 16 clkbuffer2
```

```
* Porta NAND da entrada de X4
```

```
* in1 in2 out
```

```
X16 32 33 26 sla1.2
```

```
X74 26 10 clkbuffer
```

```
*Saida
```

```
X20 31 12 buffer2
X21 32 9 buffer2
X22 33 8 buffer2
X23 34 11 buffer2
```

```
.param CL=5pF
CL1 12 vtt CL
*R1 12 vtt 50
```

```
CL2 9 vtt CL
R2 9 vtt 50
```

```
CL3 8 vtt CL
R3 8 vtt 50
```

```
CL4 11 vtt CL
R4 11 vtt 50
```

```
.temp 25
.options nopage nomod post probe
```

```
.tran 50ps 24ns
.probe v(14) v(12) v(9) v(8) v(11)
.op 5.8ns
```

```
*O comando '.op' imprime os valores das correntes em todos os FETs, nos
*instantes especificados. Serve para quando voce vai dimensionar as
*larguras das trilhas do CI e para avaliar o maximo consumo de potencia
*do CI.
```

```
***** ATRASOS DE PROPAGACAO DE CKA PARA QA *****
```

```
* calcula os valores minimo e maximo da saida do FF1
.measure tran Vol1 avg v(12) from=6.4ns to=7.1ns
.measure tran Voh1 avg v(12) from=5.4ns to=6.15ns
```

```
* calcula o swing logico da saida
.measure tran Vppl param='Voh1-Vol1'
```

```
* calcula os pontos de 10% e 90% para calculo dos tempos de subida e
descida
```

```
.measure tran ten1 param='0.1*Vppl+Vol1'
.measure tran ninety1 param='0.9*Vppl+Vol1'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty1 param='0.5*Vppl+Vol1'
```

```
* calcula o t1h da saida
.measure tran t1h1 trig v(12) val=ten1 rise=2
+      targ v(12) val=ninety1 rise=2
```

```
* calcula o t1l da saida
.measure tran t1l1 trig v(12) val=ninety1 fall=3
```

```

+          targ v(12) val=ten1    fall=3

* calcula o tphl - quando o clock desce e Q tambem
.measure tran tphl1 trig v(14) val=fifty    fall=6
+          targ v(12) val=fifty1 fall=4

* calcula o tplh - quando o clock desce e Q sobe
.measure tran tplh1 trig v(14) val=fifty    fall=5
+          targ v(12) val=fifty1 rise=3

* calcula o atraso de propagacao tpd de CK para Q
.measure tran tpd1 param='(tphl1+tplh1)*0.5'

***** ATRASOS DE PROPAGACAO DE QA PARA QB *****

* calcula os valores minimo e maximo da saida do FF2
.measure tran Vol2 avg v(9) from=3.2ns to=4.6ns
.measure tran Voh2 avg v(9) from=5.2ns to=6.6ns

* calcula o swing logico da saida
.measure tran Vpp2 param='Voh2-Vol2'

* calcula os pontos de 10% e 90% para calculo dos tempos de subida e
descida
.measure tran ten2 param='0.1*Vpp2+Vol2'
.measure tran ninety2 param='0.9*Vpp2+Vol2'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty2 param='0.5*Vpp2+Vol2'

* calcula o tlh da saida
.measure tran tlh2 trig v(9) val=ten2    rise=2
+          targ v(9) val=ninety2 rise=2

* calcula o thl da saida
.measure tran thl2 trig v(9) val=ninety2 fall=2
+          targ v(9) val=ten2    fall=2

* calcula o tphl - quando o clock desce e Q tambem
.measure tran tphl2 trig v(12) val=fifty1    fall=4
+          targ v(9) val=fifty2 fall=2

* calcula o tplh - quando o clock desce e Q sobe
.measure tran tplh2 trig v(12) val=fifty1    fall=3
+          targ v(9) val=fifty2 rise=2

* calcula o atraso de propagacao tpd de CK para Q
.measure tran tpd2 param='(tphl2+tplh2)*0.5'

***** ATRASOS DE PROPAGACAO DE CKA PARA QC *****

* calcula os valores minimo e maximo da saida do FF3
.measure tran Vol3 avg v(8) from=7.5ns to=12.5ns
.measure tran Voh3 avg v(8) from=13.5ns to=16.5ns

* calcula o swing logico da saida
.measure tran vpp3 param='Voh3-Vol3'

```

```

* calcula os pontos de 10% e 90% para calculo dos tempos de subida e
descida
.measure tran ten3 param='0.1*vpp3+vol3'
.measure tran ninety3 param='0.9*vpp3+vol3'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty3 param='0.5*vpp3+vol3'

* calcula o tih da saida
.measure tran tlh3 trig v(8) val=ten3      rise=2
+          targ v(8) val=ninety3 rise=2

* calcula o thl da saida
.measure tran thl3 trig v(8) val=ninety3 fall=2
+          targ v(8) val=ten3      fall=2

* calcula o tphl - quando o clock desce e Q tambem
.measure tran tphl3 trig v(14) val=fifty      fall=16
+          targ v(8) val=fifty3 fall=2

* calcula o tplh - quando o clock desce e Q sobe
.measure tran tplh3 trig v(14) val=fifty      fall=12
+          targ v(8) val=fifty3 rise=2

* calcula o atraso de propagacao tpd de CK para Q
.measure tran tpd3 param='(tphl3+tplh3)*0.5'

***** ATRASOS DE PROPAGACAO DE CK PARA QD *****

* calcula os valores minimo e maximo da saida do FF4
.measure tran Vol4 avg v(11) from=10ns to=16ns
.measure tran Voh4 avg v(11) from=17.5ns to=18.5ns

* calcula o swing logico da saida
.measure tran Vpp4 param='Voh4-Vol4'

* calcula os pontos de 10% e 90% para calculo dos tempos de subida e
descida
.measure tran ten4 param='0.1*Vpp4+Vol4'
.measure tran ninety4 param='0.9*Vpp4+Vol4'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty4 param='0.5*Vpp4+Vol4'

* calcula o tih da saida
.measure tran tlh4 trig v(11) val=ten4      rise=1
+          targ v(11) val=ninety4 rise=1

* calcula o thl da saida
.measure tran thl4 trig v(11) val=ninety4 fall=1
+          targ v(11) val=ten4      fall=1

* calcula o tphl - quando o clock desce e Q tambem
.measure tran tphl4 trig v(14) val=fifty1      fall=8
+          targ v(11) val=fifty4 fall=1

* calcula o tplh - quando o clock desce e Q sobe
.measure tran tplh4 trig v(14) val=fifty1      fall=6
+          targ v(11) val=fifty4 rise=1

```

```
* calcula o atraso de propagacao tpd de CK para Q
.measure tran tpd4 param='(tphl4+tplh4)*0.5'

.end
```

### Programa 27: “deccntslow.sp”.

Os subcircuitos são os mesmos, e as linhas seguintes devem ser substituídas no Programa 26.

```
.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' slow
.unprotect

(.....)

*****
***** CIRCUITO PRINCIPAL *****
*****

(.....)
.param vdc=-1.8V

(.....)

.temp 0

(.....)

***** ATRASOS DE PROPAGACAO DE CK PARA QA *****

* calcula os valores minimo e maximo da saida do FF1
.measure tran Vol1 avg v(12) from=6.7ns to=7.2ns
.measure tran Voh1 avg v(12) from=7.6ns to=8.3ns

* calcula o swing logico da saida
.measure tran Vppl param='Voh1-Vol1'

* calcula os pontos de 10% e 90% para calculo dos tempos de subida e
descida
.measure tran ten1 param='0.1*Vppl+Vol1'
.measure tran ninety1 param='0.9*Vppl+Vol1'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty1 param='0.5*Vppl+Vol1'

* calcula o t1h da saida
.measure tran t1h1 trig v(12) val=ten1 rise=2
+      targ v(12) val=ninety1 rise=2

* calcula o t1l da saida
.measure tran t1l1 trig v(12) val=ninety1 fall=3
+      targ v(12) val=ten1 fall=3
```

```

* calcula o tphl - quando o clock desce e Q tambem
.measure tran tphl1 trig v(14) val=fifty fall=6
+      targ v(12) val=fifty1 fall=3

* calcula o tplh - quando o clock desce e Q sobe
.measure tran tplh1 trig v(14) val=fifty fall=5
+      targ v(12) val=fifty1 rise=3

* calcula o atraso de propagacao tpd de CK para Q
.measure tran tpd1 param='(tphl1+tplh1)*0.5'

***** ATRASOS DE PROPAGACAO DE QA PARA QB *****

* calcula os valores minimo e maximo da saida do FF2
.measure tran Vol2 avg v(9) from=5.3ns to=6.7ns
.measure tran Voh2 avg v(9) from=7.2ns to=8.8ns

* calcula o swing logico da saida
.measure tran Vpp2 param='Voh2-Vol2'

* calcula os pontos de 10% e 90% para calculo dos tempos de subida e
descida
.measure tran ten2 param='0.1*Vpp2+Vol2'
.measure tran ninety2 param='0.9*Vpp2+Vol2'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty2 param='0.5*Vpp2+Vol2'

* calcula o tlh da saida
.measure tran tlh2 trig v(9) val=ten2 rise=2
+      targ v(9) val=ninety2 rise=2

* calcula o thl da saida
.measure tran thl2 trig v(9) val=ninety2 fall=2
+      targ v(9) val=ten2 fall=2

* calcula o tphl - quando o clock desce e Q tambem
.measure tran tphl2 trig v(12) val=fifty1 fall=4
+      targ v(9) val=fifty2 fall=2

* calcula o tplh - quando o clock desce e Q sobe
.measure tran tplh2 trig v(12) val=fifty1 fall=3
+      targ v(9) val=fifty2 rise=2

* calcula o atraso de propagacao tpd de CK para Q
.measure tran tpd2 param='(tphl2+tplh2)*0.5'

***** ATRASOS DE PROPAGACAO DE CKA PARA QC *****

* calcula os valores minimo e maximo da saida do FF3
.measure tran Vol3 avg v(8) from=9.5ns to=12ns
.measure tran Voh3 avg v(8) from=5.5ns to=9ns

* calcula o swing logico da saida
.measure tran vpp3 param='Voh3-Vol3'

* calcula os pontos de 10% e 90% para calculo dos tempos de subida e
descida
.measure tran ten3 param='0.1*vpp3+vol3'

```



```

.measure tran ninety3 param='0.9*vpp3+vol3'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty3 param='0.5*vpp3+vol3'

* calcula o tlh da saida
.measure tran tlh3 trig v(8) val=ten3 rise=1
+      targ v(8) val=ninety3 rise=1

* calcula o thl da saida
.measure tran thl3 trig v(8) val=ninety3 fall=2
+      targ v(8) val=ten3 fall=2

* calcula o tphl - quando o clock desce e Q tambem
.measure tran tphl3 trig v(14) val=fifty fall=9
+      targ v(8) val=fifty3 fall=2

* calcula o tplh - quando o clock desce e Q sobe
.measure tran tplh3 trig v(14) val=fifty fall=14
+      targ v(8) val=fifty3 rise=2

* calcula o atraso de propagacao tpd de CK para Q
.measure tran tpd3 param='(tphl3+tplh3)*0.5'

***** ATRASOS DE PROPAGACAO DE CKA PARA QD *****

* calcula os valores minimo e maximo da saida do FF4
.measure tran Vol4 avg v(11) from=12ns to=18ns
.measure tran Voh4 avg v(11) from=9.3ns to=10.9ns

* calcula o swing logico da saida
.measure tran Vpp4 param='Voh4-Vol4'

* calcula os pontos de 10% e 90% para calculo dos tempos de subida e
descida
.measure tran ten4 param='0.1*Vpp4+Vol4'
.measure tran ninety4 param='0.9*Vpp4+Vol4'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty4 param='0.5*Vpp4+Vol4'

* calcula o tlh da saida
.measure tran tlh4 trig v(11) val=ten4 rise=1
+      targ v(11) val=ninety4 rise=1

* calcula o thl da saida
.measure tran thl4 trig v(11) val=ninety4 fall=1
+      targ v(11) val=ten4 fall=1

* calcula o tphl - quando o clock desce e Q tambem
.measure tran tphl4 trig v(14) val=fifty1 fall=10
+      targ v(11) val=fifty4 fall=1

* calcula o tplh - quando o clock desce e Q sobe
.measure tran tplh4 trig v(14) val=fifty1 fall=8
+      targ v(11) val=fifty4 rise=1

* calcula o atraso de propagacao tpd de CK para Q
.measure tran tpd4 param='(tphl4+tplh4)*0.5'

```

.end

### Programa 28: "deccntfast.sp".

Os subcircuitos são os mesmos, e as linhas seguintes devem ser substituídas no Programa 26.

```
.protect
.include '~/vsc3.11models'
.lib '~/vsc3.11corners' fast
.unprotect
```

```
(....)
```

```
*****
***** CIRCUITO PRINCIPAL *****
*****
```

```
(....)
```

```
.param vdc=-2.2V
```

```
(....)
```

```
.temp 85
```

```
(....)
```

```
***** ATRASOS DE PROPAGACAO DE CKA PARA QA *****
```

```
* calcula os valores minimo e maximo da saida do FF1
```

```
.measure tran V011 avg v(12) from=6.6ns to=7.1ns
```

```
.measure tran V0h1 avg v(12) from=5.6ns to=6.2ns
```

```
* calcula o swing logico da saida
```

```
.measure tran Vppl param='V0h1-V011'
```

```
* calcula os pontos de 10% e 90% para calculo dos tempos de subida e descida
```

```
.measure tran ten1 param='0.1*Vppl+V011'
```

```
.measure tran ninety1 param='0.9*Vppl+V011'
```

```
* calcula o ponto de 50% para o calculo do atraso de propagacao
```

```
.measure tran fifty1 param='0.5*Vppl+V011'
```

```
* calcula o t1h da saida
```

```
.measure tran t1h1 trig v(12) val=ten1 rise=2
```

```
+ targ v(12) val=ninety1 rise=2
```

```
* calcula o t1l da saida
```

```
.measure tran t1l1 trig v(12) val=ninety1 fall=3
```

```
+ targ v(12) val=ten1 fall=3
```

```
* calcula o tph1 - quando o clock desce e Q tambem
```

```
.measure tran tph11 trig v(14) val=fifty fall=6
```

```
+ targ v(12) val=fifty1 fall=4
```

```

* calcula o tplh - quando o clock desce e Q sobe
.measure tran tplh1 trig v(14) val=fifty fall=5
+      targ v(12) val=fifty1 rise=3

* calcula o atraso de propagacao tpd de CK para Q
.measure tran tpd1 param='(tphl1+tplh1)*0.5'

***** ATRASOS DE PROPAGACAO DE QA PARA QB *****

* calcula os valores minimo e maximo da saida do FF2
.measure tran Vol2 avg v(9) from=5ns to=6.6ns
.measure tran Voh2 avg v(9) from=7.2ns to=8.6ns

* calcula o swing logico da saida
.measure tran Vpp2 param='Voh2-Vol2'

* calcula os pontos de 10% e 90% para calculo dos tempos de subida e
descida
.measure tran ten2 param='0.1*Vpp2+Vol2'
.measure tran ninety2 param='0.9*Vpp2+Vol2'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty2 param='0.5*Vpp2+Vol2'

* calcula o tlh da saida
.measure tran tlh2 trig v(9) val=ten2 rise=2
+      targ v(9) val=ninety2 rise=2

* calcula o thl da saida
.measure tran thl2 trig v(9) val=ninety2 fall=2
+      targ v(9) val=ten2 fall=2

* calcula o tphl - quando o clock desce e Q tambem
.measure tran tphl2 trig v(12) val=fifty1 fall=5
+      targ v(9) val=fifty2 fall=2

* calcula o tplh - quando o clock desce e Q sobe
.measure tran tplh2 trig v(12) val=fifty1 fall=4
+      targ v(9) val=fifty2 rise=2

* calcula o atraso de propagacao tpd de CK para Q
.measure tran tpd2 param='(tphl2+tplh2)*0.5'

***** ATRASOS DE PROPAGACAO DE CKA PARA QC *****

* calcula os valores minimo e maximo da saida do FF3
.measure tran Vol3 avg v(8) from=9.5ns to=13ns
.measure tran Voh3 avg v(8) from=5.5ns to=8.5ns

* calcula o swing logico da saida
.measure tran vpp3 param='Voh3-Vol3'

* calcula os pontos de 10% e 90% para calculo dos tempos de subida e
descida
.measure tran ten3 param='0.1*vpp3+vol3'
.measure tran ninety3 param='0.9*vpp3+vol3'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty3 param='0.5*vpp3+vol3'

```

```

* calcula o tlh da saida
.measure tran tlh3 trig v(8) val=ten3 rise=1
+      targ v(8) val=ninety3 rise=1

* calcula o thl da saida
.measure tran thl3 trig v(8) val=ninety3 fall=1
+      targ v(8) val=ten3 fall=1

* calcula o tphl - quando o clock desce e Q tambem
.measure tran tphl3 trig v(14) val=fifty fall=8
+      targ v(8) val=fifty3 fall=1

* calcula o tplh - quando o clock desce e Q sobe
.measure tran tplh3 trig v(14) val=fifty fall=4
+      targ v(8) val=fifty3 rise=1

* calcula o atraso de propagacao tpd de CK para Q
.measure tran tpd3 param='(tphl3+tplh3)*0.5'

***** ATRASOS DE PROPAGACAO DE CK PARA QD *****

* calcula os valores minimo e maximo da saida do FF4
.measure tran Vol4 avg v(11) from=0ns to=8.5ns
.measure tran Voh4 avg v(11) from=9.2ns to=10.6ns

* calcula o swing logico da saida
.measure tran Vpp4 param='Voh4-Vol4'

* calcula os pontos de 10% e 90% para calculo dos tempos de subida e
descida
.measure tran ten4 param='0.1*Vpp4+Vol4'
.measure tran ninety4 param='0.9*Vpp4+Vol4'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty4 param='0.5*Vpp4+Vol4'

* calcula o tlh da saida
.measure tran tlh4 trig v(11) val=ten4 rise=1
+      targ v(11) val=ninety4 rise=1

* calcula o thl da saida
.measure tran thl4 trig v(11) val=ninety4 fall=1
+      targ v(11) val=ten4 fall=1

* calcula o tphl - quando o clock desce e Q tambem
.measure tran tphl4 trig v(14) val=fifty1 fall=10
+      targ v(11) val=fifty4 fall=1

* calcula o tplh - quando o clock desce e Q sobe
.measure tran tplh4 trig v(14) val=fifty1 fall=8
+      targ v(11) val=fifty4 rise=1

* calcula o atraso de propagacao tpd de CK para Q
.measure tran tpd4 param='(tphl4+tplh4)*0.5'

.end

```

### Programa 29: “div3.sp”.

Os subcircuitos são os mesmos, só muda o Circuito Principal. Para obter os atrasos mínimo e máximo, usar os processos *fast* e *slow*, respectivamente.

```
*****
***** CIRCUITO PRINCIPAL *****
*****

*   CLK   QN   PR   CL
X1  15    31   4    13                FF1

*   CLK   J    Q    QN   PR
X2  5     23   32   21   16                FF2

*   CLK   QN   CL
X3  21    33   16                FF3

*   CLK   R    Q    QN   PR   CL
X4  5     10   23   34   4    13    FF4

* Setup de simulacao
.param vdc=-2V
.global 99 vtt
VBACKGATE 99 vtt DC 0.6
VTT vtt 0 DC vdc

* Frequencia de 1 GHz
.param Vil=0V Vih=1.2V td=400ps tr=100ps tf=100ps pw=400ps per=1ns
.param fifty='vdc+0.5*(Vih-Vil)'

X5 vtt 15 inbuffer

VCKB 1 vtt PULSE(Vil, Vih, td, tr, tf, pw, per)
X6 1 5 inbuffer

* Porta de Clear (desabilitada)
X7 vtt 20 inbuffer
X70 vtt 30 inbuffer
*   in out
X9 20 25 clkbuffer
X10 30 29 clkbuffer
*   in1   in2   out
X8 25 29 27 slaPRCL
X72 27 13 buffer

* Porta de Preset
X11 8 17 inbuffer
X71 1 19 inbuffer
*   in out
X13 17 18 clkbuffer
X14 19 24 clkbuffer
```

```

*      in1    in2      out
X12    18     24       28    slaPRCL
X73    28     4        buffer

```

```

* Porta de PR de X2 e CL de X3
*      in1    in2      out
X15    13     4        22    sla1.2
X69    22     16       clkbuffer2

```

```

* Porta NAND da entrada de X4
*      in1    in2      out
X16    32     33       26    sla1.2
X74    26    10       clkbuffer

```

```

*Saida
X20 31 12 buffer2
X21 32 9  buffer2
X22 33 8  buffer2
X23 34 11 buffer2

```

```

.param CL=4pF
CL1 12 vtt CL
R1 12 vtt 50

```

```

CL2 9 vtt CL
R2 9 vtt 50

```

```

CL3 8 vtt CL

```

```

CL4 11 vtt CL
R4 11 vtt 50

```

```

.temp 25
.options nopage nomod post probe

```

```

.tran 50ps 12ns
.probe v(1) v(9) v(8) v(11) v(4)

```

```

***** ATRASOS DE PROPAGACAO DE CKB PARA QB *****

```

```

* calcula os valores minimo e maximo da saida do FF1
.measure tran Vol1 avg v(9) from=6.4ns to=8ns
.measure tran Voh1 avg v(9) from=5.4ns to=6.1ns

```

```

* calcula o swing logico da saida
.measure tran Vppl param='Voh1-Vol1'

```

```

* calcula os pontos de 10% e 90% para calculo dos tempos de subida e
descida
.measure tran ten1 param='0.1*Vppl+Vol1'
.measure tran ninety1 param='0.9*Vppl+Vol1'

```

```

* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty1 param='0.5*Vpp1+Vol1'

* calcula o tlh da saida
.measure tran tlh1 trig v(9) val=ten1 rise=2
+      targ v(9) val=ninety1 rise=2

* calcula o thl da saida
.measure tran thl1 trig v(9) val=ninety1 fall=2
+      targ v(9) val=ten1 fall=2

* calcula o tphl - quando o clock desce e Q tambem
.measure tran tphl1 trig v(1) val=fifty fall=6
+      targ v(9) val=fifty1 fall=2

* calcula o tplh - quando o clock desce e Q sobe
.measure tran tplh1 trig v(1) val=fifty fall=5
+      targ v(9) val=fifty1 rise=2

* calcula o atraso de propagacao tpd de CK para Q
.measure tran tpd1 param='(tphl1+tplh1)*0.5'

***** ATRASOS DE PROPAGACAO DA PORTA DE PRESET *****

* calcula os valores minimo e maximo da saida
.measure tran Vol2 avg v(4) from=4.1ns to=6.4ns
.measure tran Voh2 avg v(4) from=3.7ns to=3.9ns

* calcula o swing logico da saida
.measure tran Vpp2 param='Voh2-Vol2'

* calcula os pontos de 10% e 90% para calculo dos tempos de subida e
descida
.measure tran ten2 param='0.1*Vpp2+Vol2'
.measure tran ninety2 param='0.9*Vpp2+Vol2'
* calcula o ponto de 50% para o calculo do atraso de propagacao
.measure tran fifty2 param='0.5*Vpp2+Vol2'

* calcula o tlh da saida
.measure tran tlh2 trig v(4) val=ten2 rise=2
+      targ v(4) val=ninety2 rise=2

* calcula o thl da saida
.measure tran thl2 trig v(4) val=ninety2 fall=2
+      targ v(4) val=ten2 fall=2

* calcula o tphl - quando o clock desce e Q tambem
.measure tran tphl2 trig v(8) val=fifty1 fall=2
+      targ v(4) val=fifty2 fall=2

* calcula o tplh - quando o clock desce e Q sobe
.measure tran tplh2 trig v(1) val=fifty1 RISE=4
+      targ v(4) val=fifty2 rise=2

* calcula o atraso de propagacao tpd de CK para Q
.measure tran tpd2 param='(tphl2+tplh2)*0.5'

.end

```

## APÊNDICE 2.

### Arquivo de Tecnologia “cmp.tech26” Fornecido pelo CMP para o *Software Magic*, Processo HGaAs-III da Vitesse.

O arquivo de tecnologia para o CAD Magic, listado a seguir, é baseado na versão originalmente desenvolvida na Universidade de Middlesex, e fornecida gratuitamente pelo CMP. Entretanto, sua versão original não implementava as quatro camadas de metal que são exigidas pelo próprio CMP para os *pads* de todos os CIs projetados e, dessa forma, ela não poderia ser usada para este circuito. A versão fornecida a seguir foi complementada com o **metal 4** e a **via 4** pelo autor desta Dissertação.

Para uma compreensão das diversas seções que compõem um arquivo de tecnologia para o Magic, devem ser consultadas as referências [11] e [88].

```
tech
    cmp
end

planes
    implant

    substrate
    metal1
    metal2
    metal3
    metal4
end

types
    substrate      active_area,act
    substrate      gate_metal,gm
    substrate      ohmic_metal,om,source_drain_metal,sdm
    metal1         metal_1,m1
    metal2         metal_2,m2
    metal3         metal_3,m3
    metal4         metal_4,m4

    substrate      contact_0,ohmic_contact,oc
    substrate      ohmic_contact_dfet,ocd
    metal1         vialg,vg
    metal1         vialsd,vsd
    metal2         via2,v2
    metal3         via3,v3
    metal3         pad
    metal3         probe_pad,ppad
    metal4         via4,v4

    metal4         glass
    metal4         glass2

    substrate      efet,ef
```



```

substrate      dfet,df
implant        depletion_implant,di
end

```

```

styles
  styletype      mos

  gate_metal      1

  source_drain_metal  2
  source_drain_metal  33

  contact_0       1
  contact_0       11
  contact_0       33

  ocd             1
  ocd             7
  ocd             33

  active_area     2

  metal_1         20
  metal_2         21
  metal_3         5
  metal_4         4

  vialg           1
  vialg           20
  vialg           32

  vsd             2
  vsd             20
  vsd             32

  via2            20
  via2            21
  via2            32

  via3            5
  via3            21
  via3            32

  via4            5
  via4            4
  via4            32

  efet            6
  efet            11

  dfet            6
  dfet            7

  depletion_implant 7

```

```

    pad      5
    pad      4
    pad      32
    pad      33

    ppad     5
    ppad     4
    ppad     32
    ppad     33

    glass    32
    glass2   32

    error_s   42
    error_p   42
    error_ps  42
end

contact
    vialg    gate_metal    metal_1
    vsd      active_area   metal_1
    via2     metal_1       metal_2
    via3     metal_2       metal_3
    via4     metal_3       metal_4
    pad      metal_3       glass
    ppad     metal_3       glass2
end

compose
    paint    via2          vialg      via2
    paint    via2          vsd        via2

    compose  efet          gate_metal  act
    compose  contact_0     gate_metal  om
    decompose dfet         gate_metal  act

    paint    ocd           gate_metal  ocd
    paint    ocd           active       ocd
    paint    ocd           om           ocd

    paint    dfet         act          dfet
    paint    dfet         gate_metal   dfet

    paint    pad          via3         pad
    paint    pad          metal_2      pad
    paint    pad          metal_3      pad
    erase    pad          metal_2      space
    erase    pad          metal_3      space
    paint    pad          metal_4      pad
end

connect
    active_area    ohmic_metal,ohmic_contact,ocd,vsd
    gate_metal     ohmic_contact,ocd,vialg,efet,dfet
    ohmic_metal    ohmic_contact,ocd,vsd
    metal_1        vialg,vsd,via2
    metal_2        via2,pad,ppad,via3
    metal_3        via3,pad,ppad,via4

```

```

metal_4          via4,pad,ppad
ohmic_contact    vialg,dfet,efet
ohmic_contact_dfet vialg,dfet
vialg            via2,vsd
vsd              via2
via2             via3
via3             via4
pad              via2,m2,via3,m3,via4,m4
ppad             via2,m2,via3,m3,via4,m4
end

```

```

cifoutput
  style          cmp
  scalefactor 10

  layer GAC active_area,vsd,efet,dfet,oc,ocd,om
    labels active_area
    calma 1 0

  layer GDI dfet,ocd
    grow 70
    or depletion_implant
    labels depletion_implant
    calma 2 0

  layer GGM gm,vialg,efet,dfet,oc,ocd
    labels gm,efet,dfet,oc,ocd
    calma 3 0

  layer GOM vsd/substrate,om
    or oc,ocd
    labels om
    calma 4 0

  layer GV1 vsd
    or vialg
    shrink 50
    calma 5 0

  layer GM1 m1,vg,vsd,v2
    labels m1,vg,vsd,v2
    calma 6 0

  layer GV2 via2
    shrink 50
    calma 7 0

  layer GM2 m2,v2,v3,pad,ppad
    labels m2,v2,v3
    calma 8 0

  layer GV3 pad,ppad
    shrink 1540
    or via3
    shrink 60
    calma 9 0

  layer GM3 pad,ppad
    shrink 200

```

```

    or m3
    or v3
    bloat-or m3 pad 200
    bloat-or m3 ppad 200
    bloat-or m3,v3/metal3 * 0 pad,ppad 200
    or via3,via4
    labels m3
    calma 10 0

    layer GV4 pad,ppad
        shrink 800
        or via4
        shrink 100
        calma 14 0

    layer GM4 pad,ppad
        shrink 550
        bloat-or m4,v4 * 0 pad/metal4,ppad/metal4 550
        or m4,via4
        calma 15 0

    layer GOG pad,ppad
        shrink 1250
        labels pad,ppad
        calma 11 0

    layer XP pad
end

```

```

cifinput
    style          cmp
    scalefactor    10

    layer depletion_implant GDI
        labels GDI

    layer active_area      GAC
        labels GAC

    layer glass            GOG
        grow 1250

    layer metal_2          GM2
        and-not XP
        labels GM2

    layer metal_1          GM1
        labels GM1

    layer gate_metal       GGM
        labels GGM

    layer om               GOM
        and GAC
        labels GOM

```

|       |         |       |     |
|-------|---------|-------|-----|
| layer | dfet    |       | GDI |
|       | shrink  | 70    |     |
|       | and     | GAC   |     |
|       | and     | GGM   |     |
|       |         |       |     |
| layer | ocd     |       | GDI |
|       | shrink  | 70    |     |
|       | and     | GAC   |     |
|       | and     | GOM   |     |
|       | and     | GGM   |     |
|       |         |       |     |
| layer | oc      |       | GAC |
|       | and     | GGM   |     |
|       | and     | GOM   |     |
|       | and-not | GDI   |     |
|       |         |       |     |
| layer | vsd     |       | GV1 |
|       | grow    | 50    |     |
|       | and     | GOM   |     |
|       | and     | GAC   |     |
|       | and     | GM1   |     |
|       |         |       |     |
| layer | vialg   |       | GV1 |
|       | grow    | 50    |     |
|       | and     | GGM   |     |
|       | and     | GM1   |     |
|       |         |       |     |
| layer | via2    |       | GV2 |
|       | grow    | 60    |     |
|       | and     | GM1   |     |
|       | and     | GM2   |     |
|       |         |       |     |
| layer | via3    |       | GV3 |
|       | grow    | 100   |     |
|       | and     | GM2   |     |
|       | and     | GM3   |     |
|       | and-not | glass |     |
|       |         |       |     |
| layer | metal_3 |       | GM3 |
|       | labels  | GM3   |     |
|       | and-not | glass |     |
|       |         |       |     |
| layer | metal_4 |       | GM4 |
|       |         |       |     |
| layer | via4    |       | GV4 |
|       | grow    | 100   |     |
|       |         |       |     |
| layer | pad     |       | GOG |
|       | grow    | 1250  |     |
|       |         |       |     |
| calma | GAC     | 1 *   |     |
| calma | GDI     | 2 *   |     |
| calma | GGM     | 3 *   |     |
| calma | GOM     | 4 *   |     |
| calma | GV1     | 5 *   |     |
| calma | GM1     | 6 *   |     |
| calma | GV2     | 7 *   |     |
| calma | GM2     | 8 *   |     |
| calma | GV3     | 9 *   |     |
| calma | GM3     | 10 *  |     |

```

calma    GOG    11 *
calma    GV4    14 *
calma    GM4    15 *

end

mzrouter
style irouter
layer    m2      32      64      256      1
layer    m1      64      32      256      1
layer    gm      128     128     512      1
contact  v2      m1      m2      1024
contact  vg      m1      gm      2056
notactive gm      vg
style garouter
layer    m2      32      64      256      1
layer    m1      64      32      256      1
contact  v2      m1      m2      1024
end

drc

edge4way gm dfet,ocd 20 dfet,ocd 0 0 \
"11W1(T): active area of transistor must be at least 2.0 microns (20
lambda)"

edge4way gm efet 20 efet 0 0 \
"11W1(T): active area of transistor must be at least 2.0 microns (20
lambda)"

edge4way efet,dfet,oc,ocd om,vsd,act 20 act,efet,dfet,vsd,oc,ocd,om 0 0
\
"11W1(T): active area of transistor must be at least 2.0 microns (20
lambda)"

width act,efet,dfet,vsd,oc,ocd,om 15 \
"11LW1(R): active as resistor must be at least 1.5 microns (15 lambda)"

spacing act,efet,dfet,vsd,oc,ocd,om act,efet,dfet,vsd,oc,ocd,om 20
touching_ok \
"11SP1: active area separation must be at least 2.0 microns (20
lambda)"

width depletion_implant 25 \
"22LW1: The depletion implant must be >= 2.5 um in each direction (25
lambda)"

spacing depletion_implant depletion_implant 20 touching_ok \
"22SP1: Unconnected depletion implants must be >= 2.0 um apart (20
lambda)"

edge4way dfet,ocd space,gm,act,om 7 depletion_implant space,gm,act,om
7\
"23OL1: The depletion implant must overlap the depletion channel by >=
0.7 um" implant

spacing efet di 7 touching_illegal \

```

"21SW1: The depletion implant must be  $\geq 0.7$  um from an enhancement-mode channel (7 lambda)"

spacing efet dfet 20 touching\_illegal \

"21SW1: efet/dfet separation must be at least 2.0 microns (20 lambda)"

width gm,vg,efet,dfet,oc,ocd 10 "33LW1: gate metal width must be at least 1.0 microns (10 lambda)"

spacing gm,vg,efet,dfet,oc,ocd gm,vg,efet,dfet,oc,ocd 15 touching\_ok \

"33SP1: gate metal/gate metal separation must be at least 1.5 microns (15 lambda)"

edge4way efet,dfet,oc,ocd gm 7 gm 0 0 \

"31OL1: gate metal must overhang transistor by at least 0.7 microns (7 lambda)"

edge4way efet,dfet,oc,ocd space 7 0 0 0 \

"31OL1: gate metal must overhang transistor by at least 0.7 microns (7 lambda)"

edge4way space efet,dfet,oc,ocd 7 0 0 0 \

"31OL1: gate metal must overhang transistor by at least 0.7 microns (7 lambda)"

spacing gm,vg,efet,dfet,oc,ocd om,vsd 8 touching\_ok \

"34SP1: gate metal/ohmic metal separation must be at least 0.8 microns (8 lambda)"

spacing gm,vg,efet,dfet,oc,ocd oc,ocd 8 touching\_ok \

"34SP1: gate metal/ohmic metal separation must be at least 0.8 microns (8 lambda)"

spacing act,efet,dfet,vsd,oc,ocd,om gm,vg,efet,dfet,oc,ocd 5 touching\_ok \

"31SP2: active area/gate metal separation must be at least 0.5 microns (5 lambda)"

edge4way act,vsd,om gm,vg 5 space 0 0 \

"31SP2: active area/gate metal separation must be at least 0.5 microns (5 lambda)"

edge4way gm efet,dfet,oc,ocd 8 efet,dfet,oc,ocd 0 0 \

"31OL2: gate metal edge must be at least 0.8 microns (8 lambda) from parallel active area edge"

edge4way act,vsd,om efet,dfet,oc,ocd 8 efet,dfet,oc,ocd 0 0 \

"31OL2: gate metal edge must be at least 0.8 micron (8 lambda) from parallel active area edge"

edge4way efet,dfet,oc,ocd act,vsd,om 8 act,vsd,om 0 0 \

"31OL2: gate metal edge must be at least 0.8 micron (8 lambda) from parallel active area edge"

edge4way act,gm,efet,dfet,space om,oc,ocd 20 om,oc,ocd,vsd/substrate 0 0 \

"44LW1: ohmic metal width must be at least 2.0 microns (20 lambda)"

width om,vsd 20 "44LW1+: (non-via) ohmic metal width must be at least 2.0 microns (20 lambda)"

```

width ohmic_contact 15\
"43OL1: ohmic contact width must be 1.5 microns (15 lambda)"

edge4way om oc,ocd 15 oc,ocd 0 0 \
"43OL2: ohmic contact must extend at least 1.5 microns (15 lambda)
beyond edge"

edge4way oc,ocd om 20 om,vsd/substrate 0 0 \
"43OL2: ohmic metal must extend at least 2.0 microns (20 lambda) beyond
edge"

edge4way oc,ocd gm,vg,efet,dfet 9 gm,vg,efet,dfet,oc,ocd 0 0 \
"43OL3: gate metal must extend at least 0.9 micron (9 lambda) beyond
ohmic contact"

width vialg 22 "55LW1: vialg width must be at least 2.2 microns (22
lambda)"

width vsd 22 "55LW1: vsd width must be at least 2.2 microns (22
lambda)"

spacing vg vsd 5 touching_illegal \
"55SP1: vialg/vsd separation must be at least 0.5 microns (5 lambda)"

spacing vg vg 5 touching_ok \
"55SP1: vialg/vialg separation must be at least 0.5 microns (5 lambda)"

spacing gm,vg,efet,dfet,oc,ocd vialsd 4 touching_illegal \
"54OL1: gate metal/vsd separation must be at least 0.4 microns (4
lambda)"

width m1,vsd,vg,v2 12 "66LW1: metal 1 width must be at least 1.2
microns (12 lambda)"

spacing m1,vsd,vg,v2 m1,vsd,vg,v2 18 touching_ok \
"66SP1: metal 1/metal 1 separation must be at least 1.8 microns (18
lambda)"

width via2 28 "77LW1: via 2 width must be at least 2.8 microns (28
lambda)"

spacing via2 via2 10 touching_ok \
"77SP1: via2/via2 separation must be at least 1.0 microns (10 lambda)"

width m2,v2,v3/metal2,pad/metal2,ppad/metal2 18 "88LW1: metal 2 width
must be at least 1.8 microns (18 lambda)"

spacing m2,v2,v3/metal2,pad/metal2,ppad/metal2
m2,v2,v3/metal2,pad/metal2,ppad/metal2 18 touching_ok \
"88SP1: metal-2/metal-2 separation must be at least 1.8 microns (18
lambda)"

width via3 38 "99LW1: via 3 width must be at least 3.8 microns (38
lambda)"

```



```

spacing via3 via3 18 touching_ok \
"99SP1: Via3 separation must be >= 1.8 um (18 lambda)"

width m3,v3,pad,ppad 30 "1010LW1: metal 3 width must be at least 3.0
microns (30 lambda)"

spacing m3,v3,pad,ppad m3,v3,pad,ppad 40 touching_ok\
"1010SP1: metal 3/metal 3 separation must be at least 4.0 microns (40
lambda)"

width via4 60 "1414LW1: via4 width must be at least 6.0 microns (60
lambda)"

spacing via4 via4 20 touching_ok \
"1414SP1: Via4 separation must be >= 2.0 um (20 lambda)"

spacing m4,v4/metal4,pad/metal4,ppad/metal4
m4,v4/metal4,pad/metal4,ppad/metal4 50 touching_ok \
"31: metal4 plane separation must be >= 5.0um (50 lambda)"

edge ~pad/metal3 pad 1000 pad 0 0 \
"BP4: pads must be at least 100 microns (1000 lambda)"

spacing pad pad 50 touching_ok \
"BP4: pad/pad separation must be at least 5 microns (50 lambda)"

spacing pad/metal4 m4,v4/metal4,pad/metal4,ppad/metal4 50 touching_ok \
"BP-4: pad/metal4 separation must be at least 5.0 microns (50 lambda)"

spacing pad m3,v3,pad,ppad 90 touching_ok \
"1010SP1: pad/metal 3 separation must be at least 9.0 microns (90
lambda)"

spacing pad/metal2 m2,v2,v3/metal2,pad/metal2,ppad/metal2 18
touching_ok \
"88SP1: pad/metal 2 separation must be at least 1.8 microns (18
lambda)"

spacing pad m1,vsd,vg,v2 90 touching_illegal \
"BP-4: pad/metal1 separation must be at least 9.0 microns (90 lambda)"
end

extract

style default
lambda 10

resist act 210000
resist gm 1500
resist om 10000
resist m1 70
resist m2 35
resist m3 25

```

```

resist      m4  25

overlap     m4  m3  215
overlap     m4  m2  108
overlap     m4  m1  80
overlap     m4  gm  72
overlap     m4  om  74
areacap     m4  67

perimc      m4  m3  5700
perimc      m4  m2  1500
perimc      m4  m1  300
perimc      m4  gm  0
perimc      m4  om  0
perimc      m4  space  0

overlap     m3  m2  215
overlap     m3  m1  127
overlap     m3  gm  108
overlap     m3  om  113
areacap     m3  97

perimc      m3  m2  3300
perimc      m3  m1  1300
perimc      m3  gm  1000
perimc      m3  om  1000
perimc      m3  space  1200

overlap     m2  m1  311
overlap     m2  gm  215
overlap     m2  om  237
areacap     m2  176

perimc      m2  m1  3700
perimc      m2  gm  1700
perimc      m2  om  1800
perimc      m2  space  2100

overlap     m1  gm  696
overlap     m1  om  993
areacap     m1  404

perimc      m1  gm  4800
perimc      m1  om  4600
perimc      m1  space  3600

areacap     gm  1024
perimc      gm  space  6100

fetresist  efet,dfet linear 210

fet efet    active          2  ENH B!  0  0
fet dfet    active,om,ocd,oc 1  DP1 B!  0  0
fet ocd     active,om,oc    1  X  B!  0  0

end

```

wiring

```

contact    vsd      22  act 0    m1  0
contact    vg       22  gm  0    m1  0
contact    via2     28  m1  0    m2  0
contact    via3     38  m2  0    m3  0
contact    via4     60  m3  0    m4  0
end

```

```

router
  layer1 metal_1 4 m1,vsd,vg,v2 5
  layer2 metal_2 5 m2,v2,v3/metal2,ppad/metal2 5
  contacts v2 8
  gridspacing 13
end

```

```

plowing
end

```

```

plot
  style versatec

```

```

efet \
  07c0 0f80 1f00 3e00 \
  7c00 f800 f001 e003 \
  c007 800f 001f 003e \
  007c 00f8 01f0 03e0

```

```

dfet,ocd \
  1f00 0f80 07c0 03e0 \
  01f0 00f8 007c 003e \
  001f 800f c007 e003 \
  f001 f800 7c00 3e00

```

```

vsd \
  c3c3 c3c3 0000 0000 \
  0000 0000 c3c3 c3c3 \
  c3c3 c3c3 0000 0000 \
  0000 0000 c3c3 c3c3

```

```

pad \
  0040 0080 0100 0200 \
  0400 0800 1000 2000 \
  4000 8000 0001 0002 \
  0004 0008 0010 0020

```

```

act,vsd/substrate,efet,dfet \
  0000 4242 6666 0000 \
  0000 2424 6666 0000 \
  0000 4242 6666 0000 \
  0000 2424 6666 0000

```

```
gm,vg/substrate,efet,dfet,oc,ocd \
  0808 0400 0202 0101 \
  8080 4000 2020 1010 \
  0808 0004 0202 0101 \
  8080 0040 2020 1010
```

```
oc,ocd,om \
  1414 2222 0000 2222 \
  4141 2222 0000 2222 \
  1414 2222 0000 2222 \
  4141 2222 0000 2222
```

```
m1,vialg/metall,vsd/metall,pad,probe \
  8080 0000 0000 0000 \
  0808 0000 0000 0000 \
  8080 0000 0000 0000 \
  0808 0000 0000 0000
```

```
pad,probe \
  0000 0000 1c1c 3e3e \
  3636 3e3e 1c1c 0000 \
  0000 0000 1c1c 3e3e \
  3636 3e3e 1c1c 0000
```

```
m2,v2,pad,probe \
  0000 1111 0000 0000 \
  0000 1111 0000 0000 \
  0000 1111 0000 0000 \
  0000 1111 0000 0000
```

```
m3,v3,pad,probe \
  2020 2020 2020 2020 \
  2020 2020 2020 2020 \
  0000 0000 0000 0000 \
  0000 0000 0000 0000
```

```
vg,vsd,v2,v3      X
```

```
style gremlin
efet                      9
dfet,ocd                  10
pad,probe                  12
act,vsd/substrate,efet,dfet,ocd  17
gm,vg/substrate,efet,dfet,ocd  19
m1,vsd,vg,v2/metall,pad,ppad  22
oc,ocd,om                  30
pad,ppad                    23
m2,v2,pad,probe            28
m3,v3,pad,probe            15
vg,vsd,v2,v3                X
```

```
end
```

### APÊNDICE 3.

## Simulações das Configurações de Divisão de Frequência do CI 7490-like.

Este Apêndice contém as formas de onda para todas as configurações de divisão de frequência do CI 7490-like, desde a divisão por 2 até a divisão por 10 simétrica, para a versão final do CI. Estas simulações foram realizadas para o processo típico a 25 °C, com  $V_{TT} = -2V$ . Deve-se observar que a excursão das saídas é sempre quase igual a 1,1 V, indo de  $V_{OL} = -1,99 V$  a  $V_{OH} = -0,9 V$ . A frequência de entrada foi de 1 GHz em todos os casos.

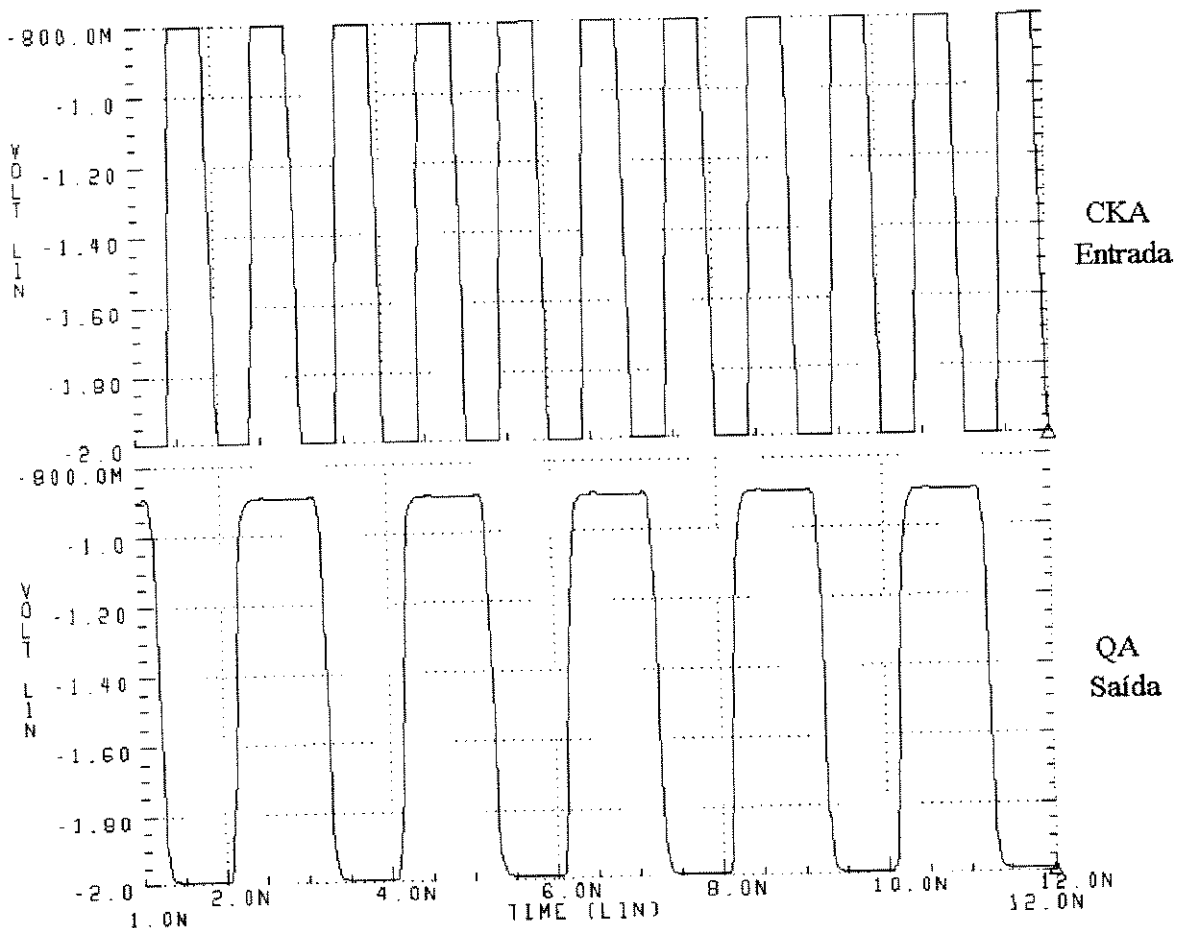


Figura A.1 – Divisão da frequência de clock por 2.

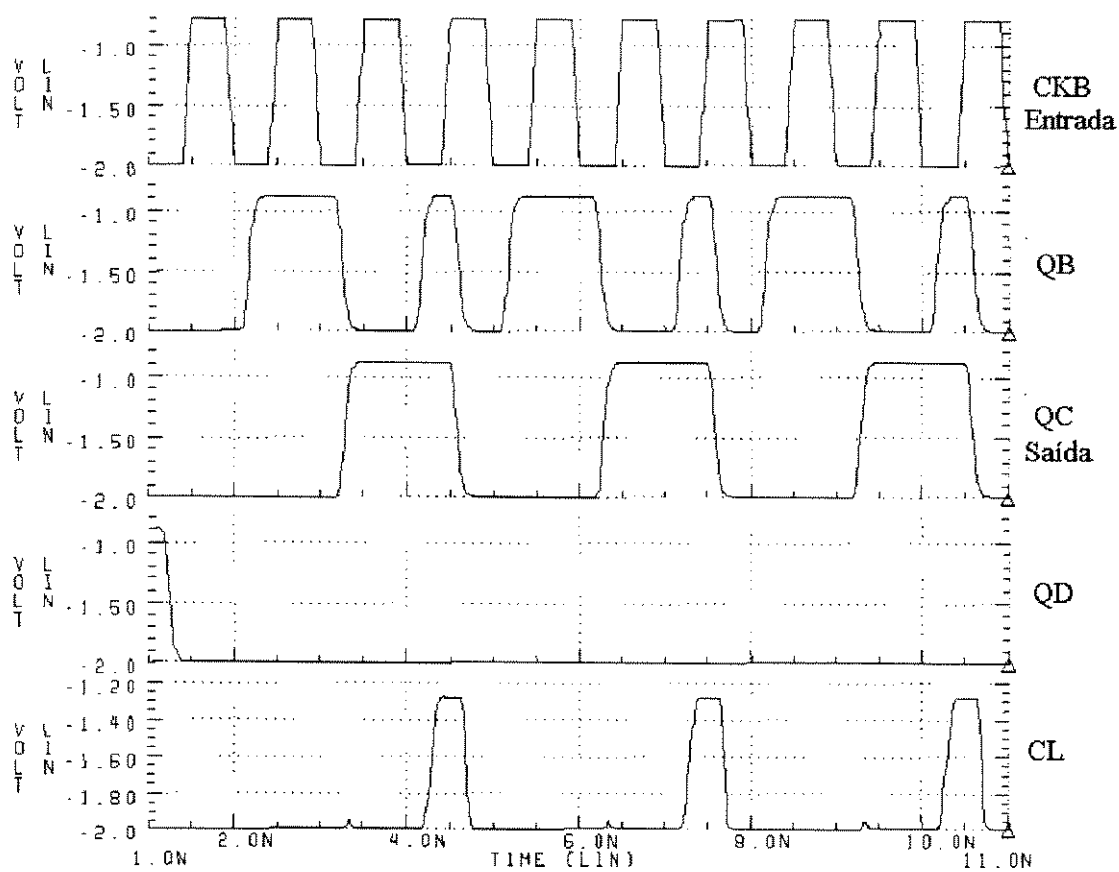


Figura A.2 – Divisão da frequência de clock por 3.

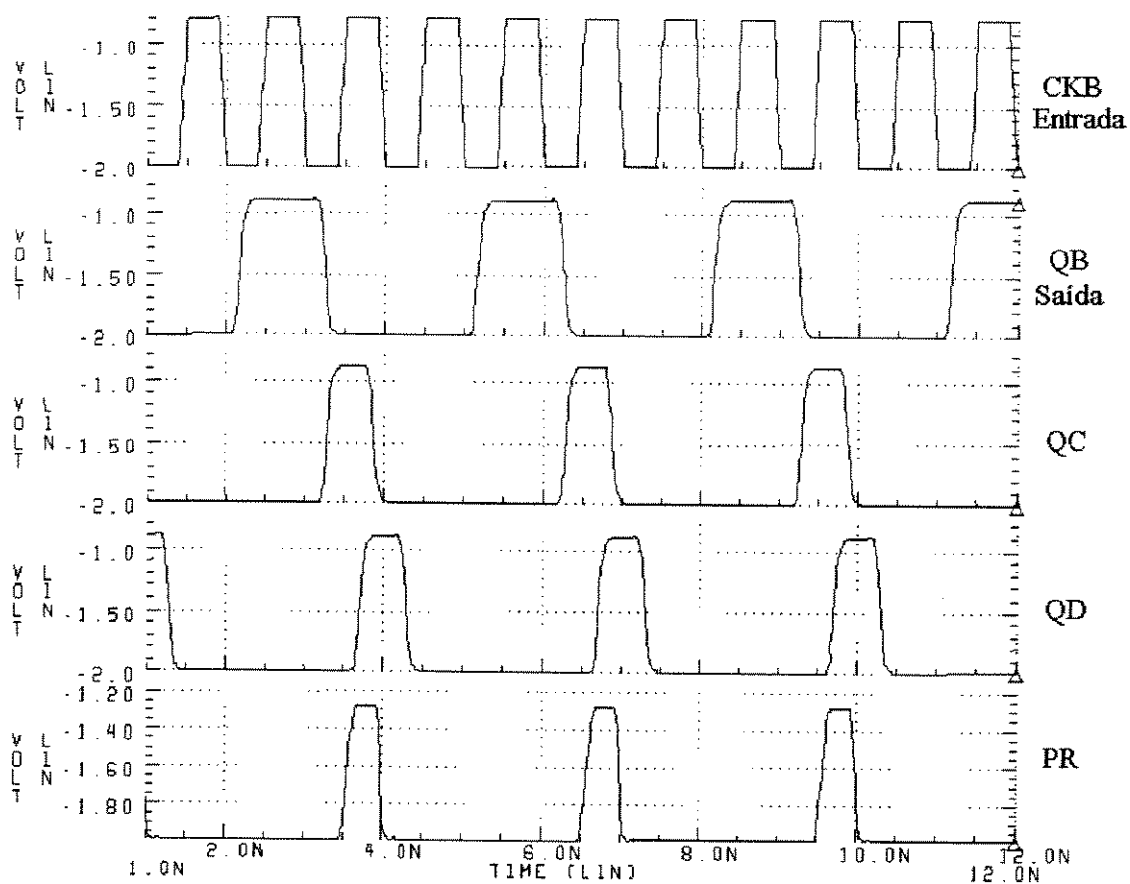


Figura A.3 – Outra divisão da frequência de clock por 3.

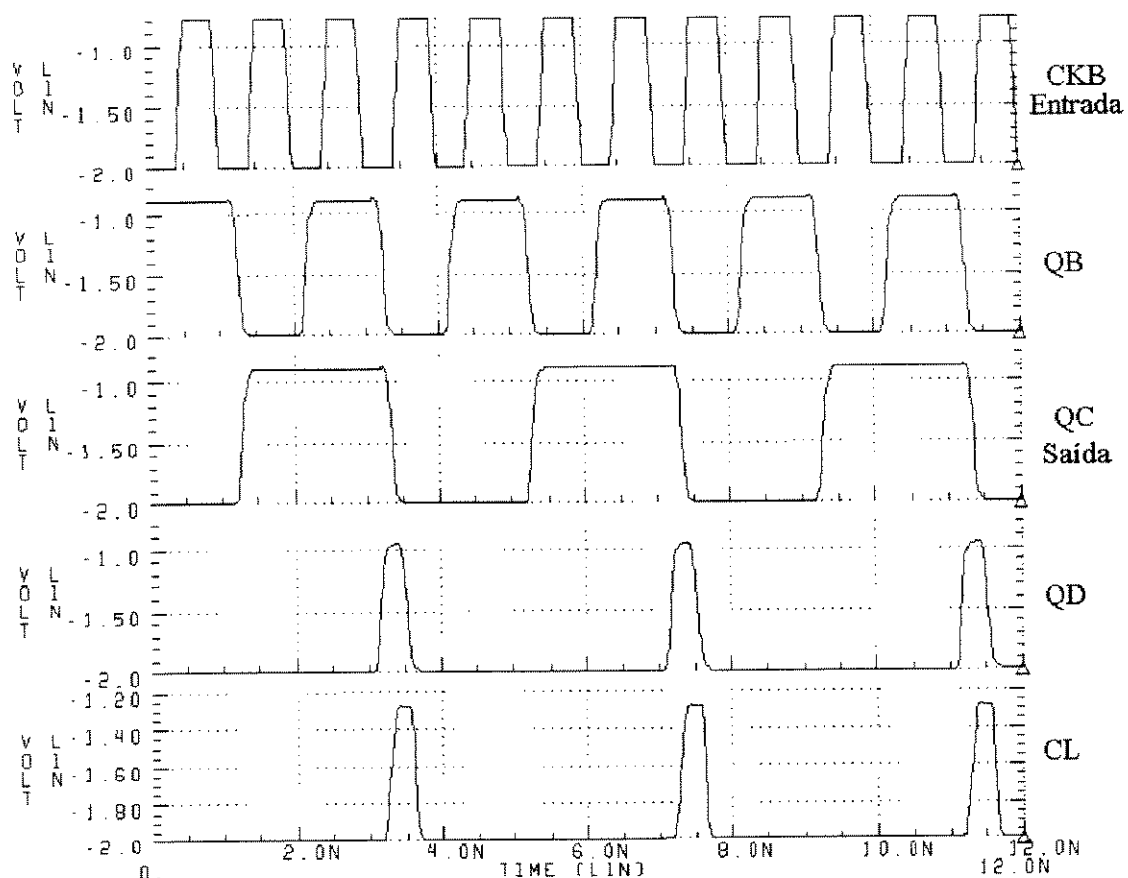


Figura A.4 – Divisão da frequência de clock por 4.

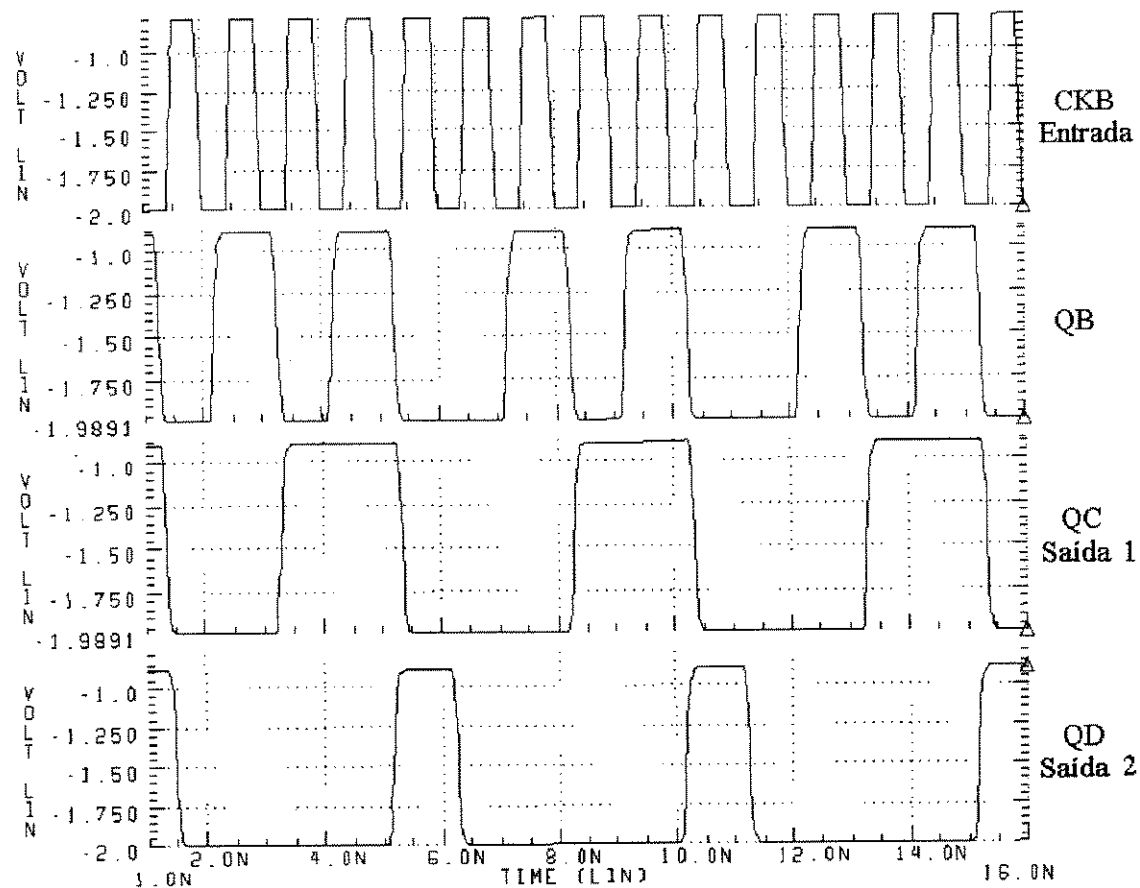


Figura A.5 – Divisão da frequência de clock por 5.

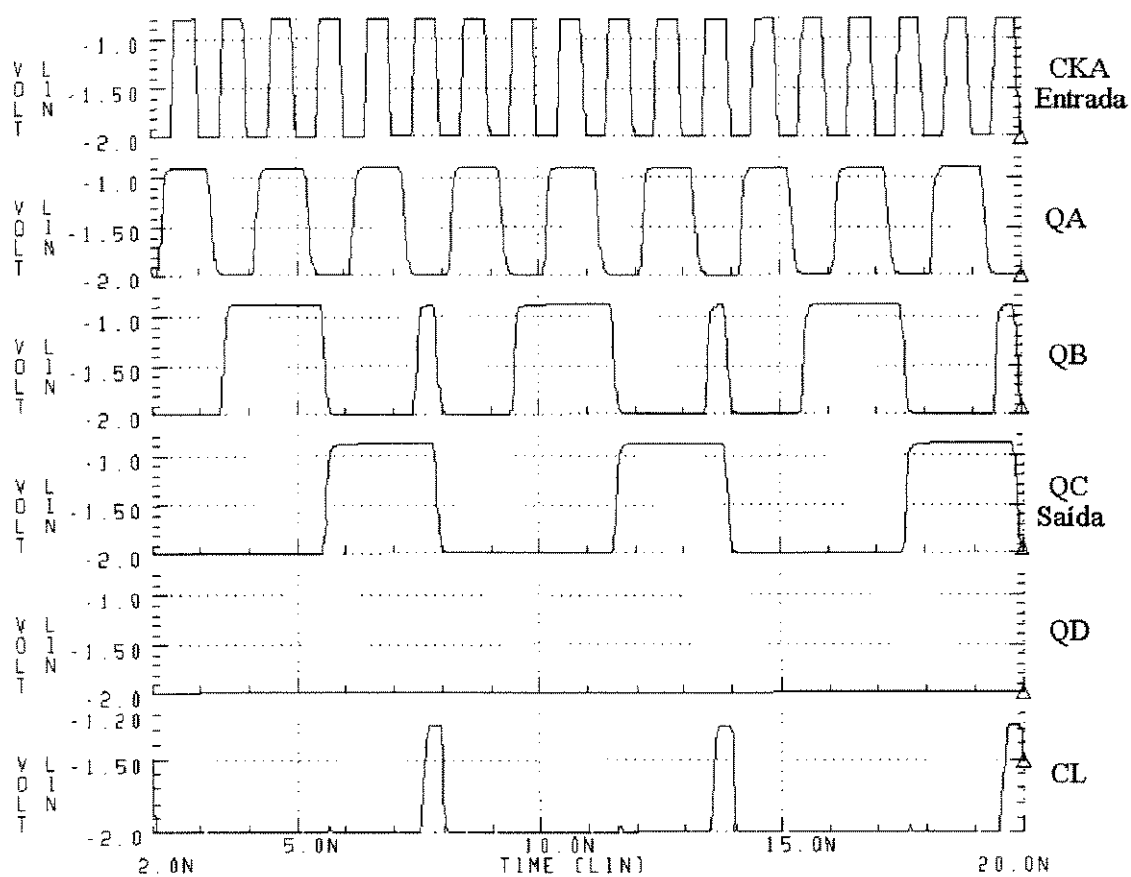


Figura A.6 – Divisão da frequência de clock por 6.

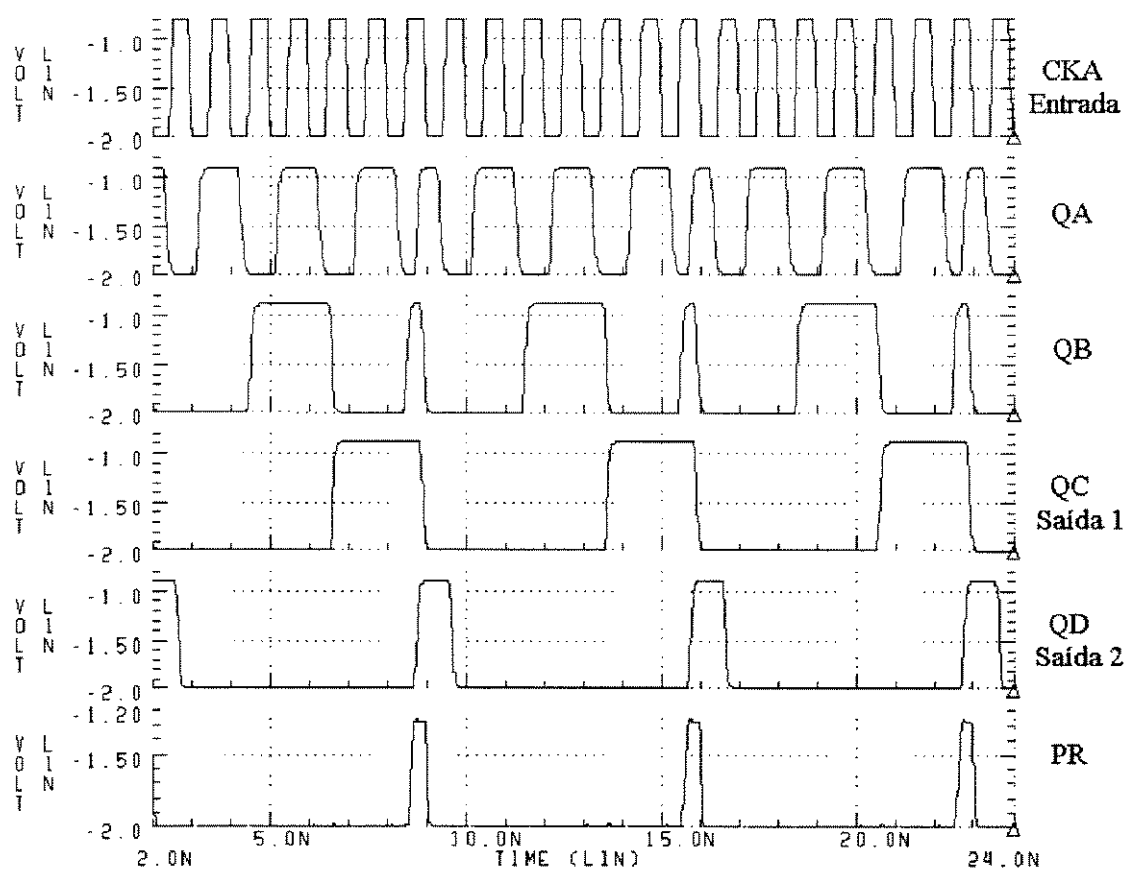


Figura A.7 – Divisão da frequência de clock por 7.





Figura A.8 – Divisão da frequência de clock por 8.

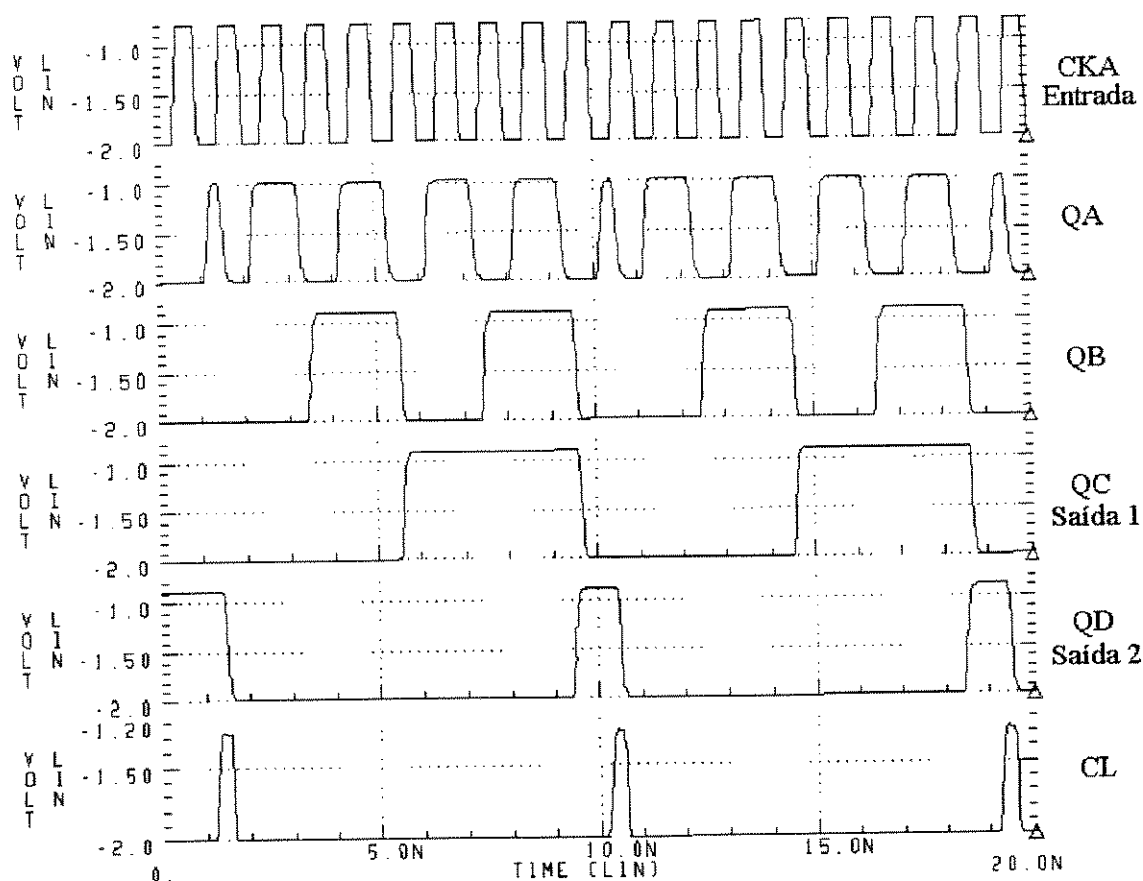


Figura A.9 – Divisão da frequência de clock por 9.

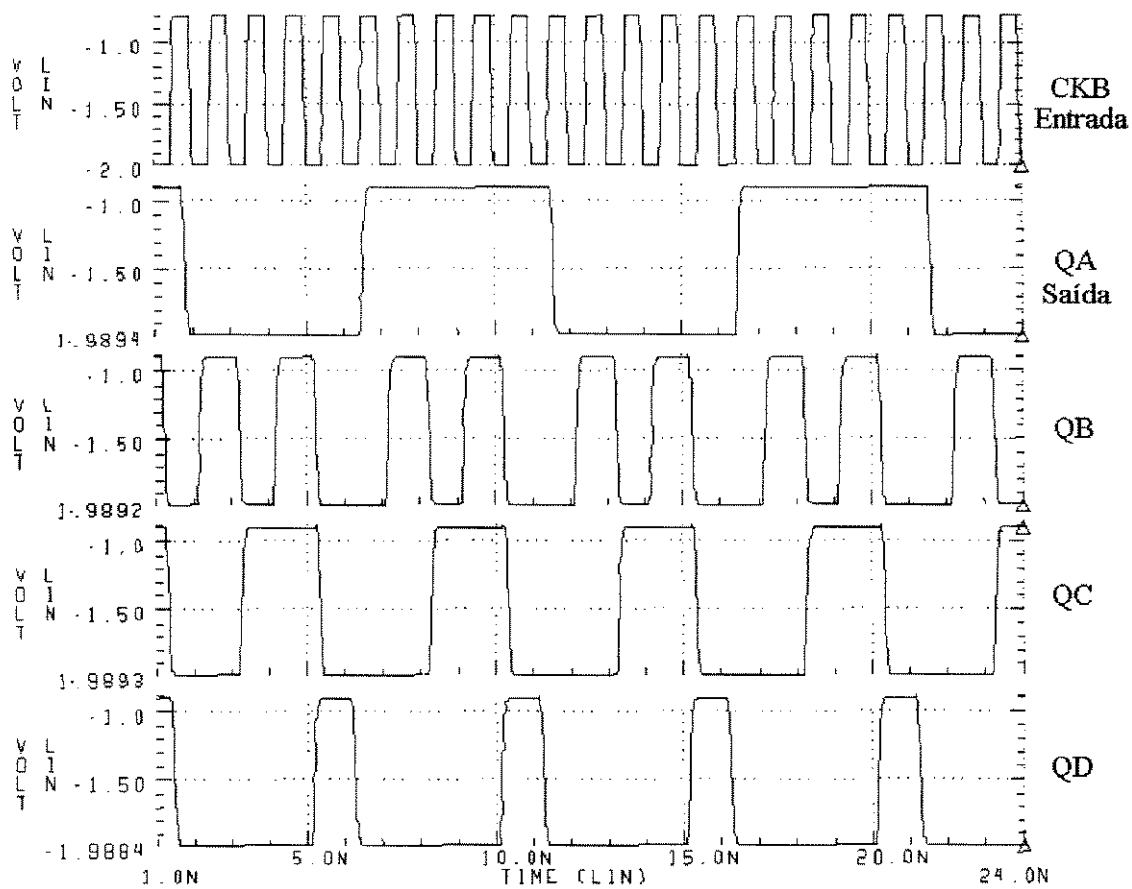


Figura A.10 – Divisão simétrica da frequência de clock por 10.

## APÊNDICE 4.

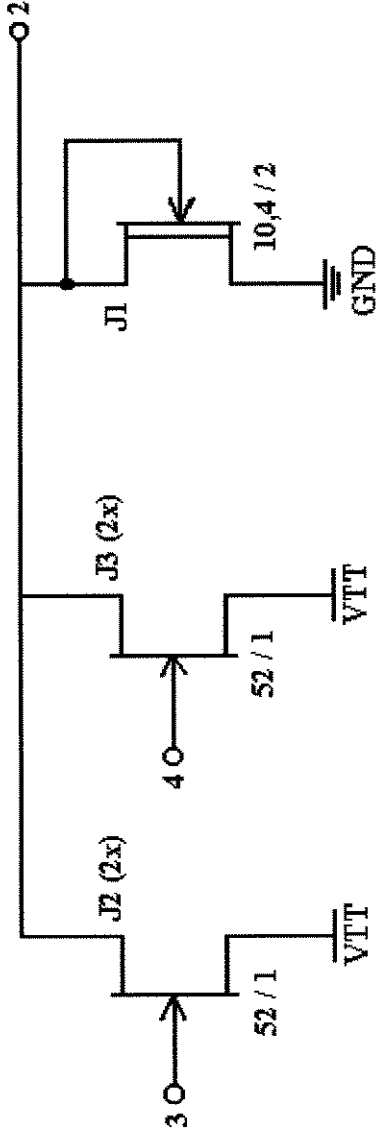
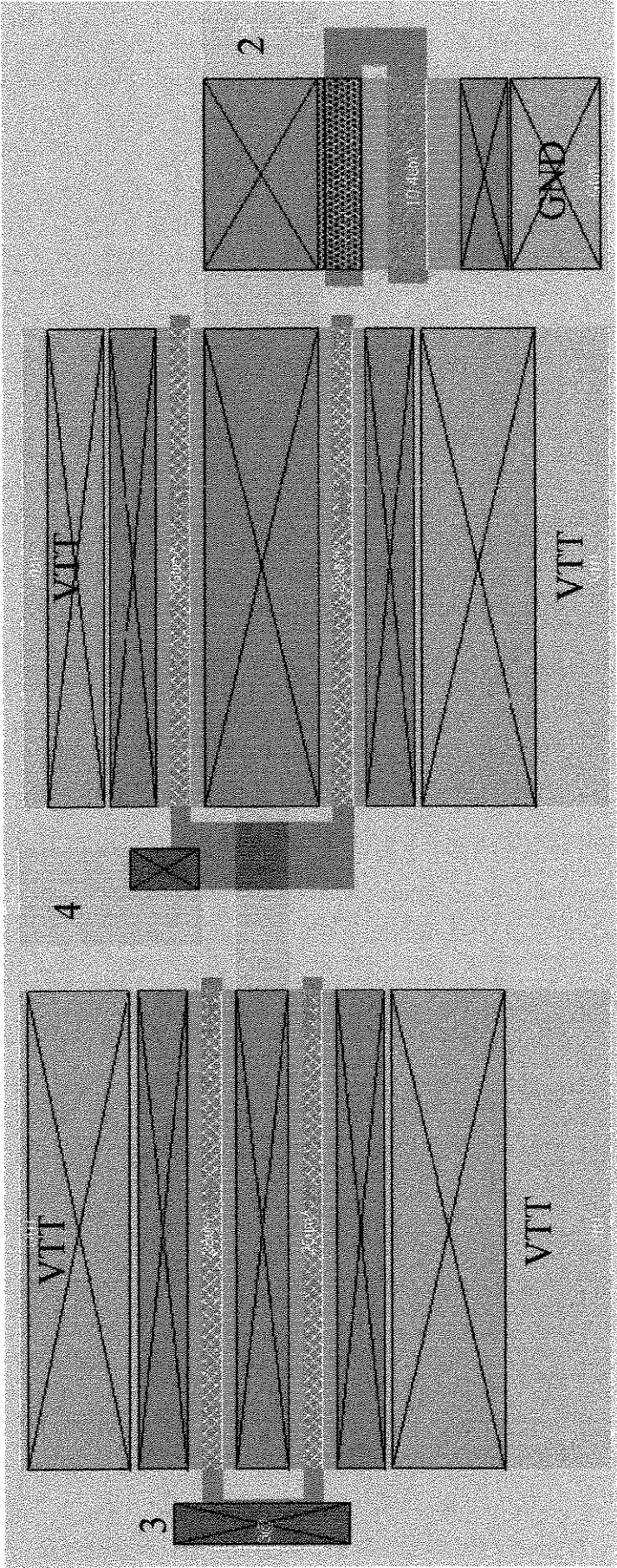
### CÉLULAS DE PORTAS LÓGICAS USADAS NO *CHIP* 7490-LIKE.

Este Apêndice contém as definições das portas lógicas e *buffers* usados para a composição dos FFs e do *chip* 7490-like. Os esquemáticos são feitos presumindo-se uma alimentação negativa. São indicadas as dimensões totais dos *gates* dos MESFETs e o número de dedos em paralelo que compõem cada *gate*. Por exemplo, “J2 (4x)” significa que o MESFET J2 é formado por quatro *gates* em paralelo. As dimensões estão na forma  $W_g/L_{g,proj}$ .

O nome de uma porta indica o seu FI e a sua posição nos FFs. Por exemplo, a NOR “mast1\_2” é adequada para o primeiro nível de portas do *latch* mestre (*master*), e tem duas entradas. A NOR “sla2\_3” foi projetada para o segundo nível de portas do *latch* escravo (*slave*), e tem FI = 3. As figuras 5.15 a 5.18 mostram essas portas em uso nos quatro FFs do CI.

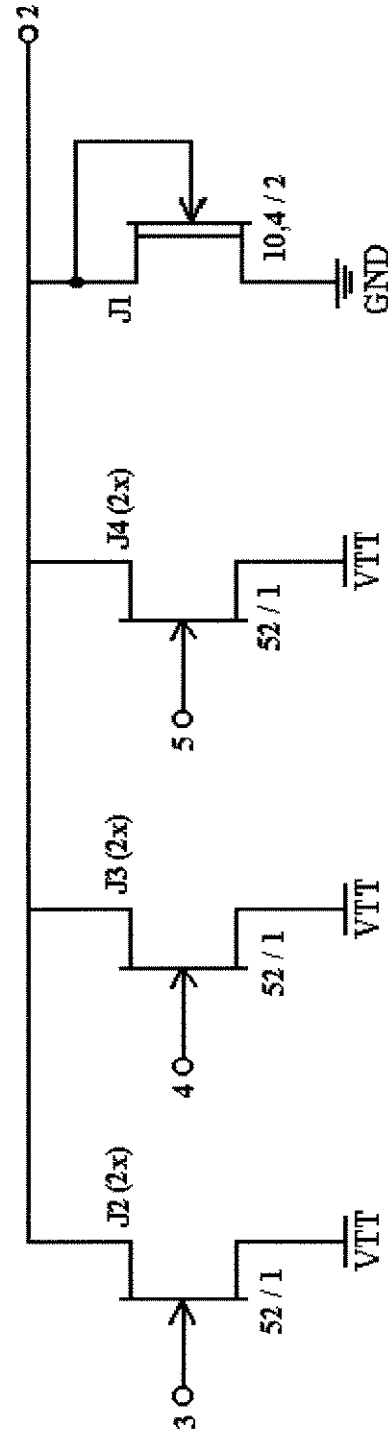
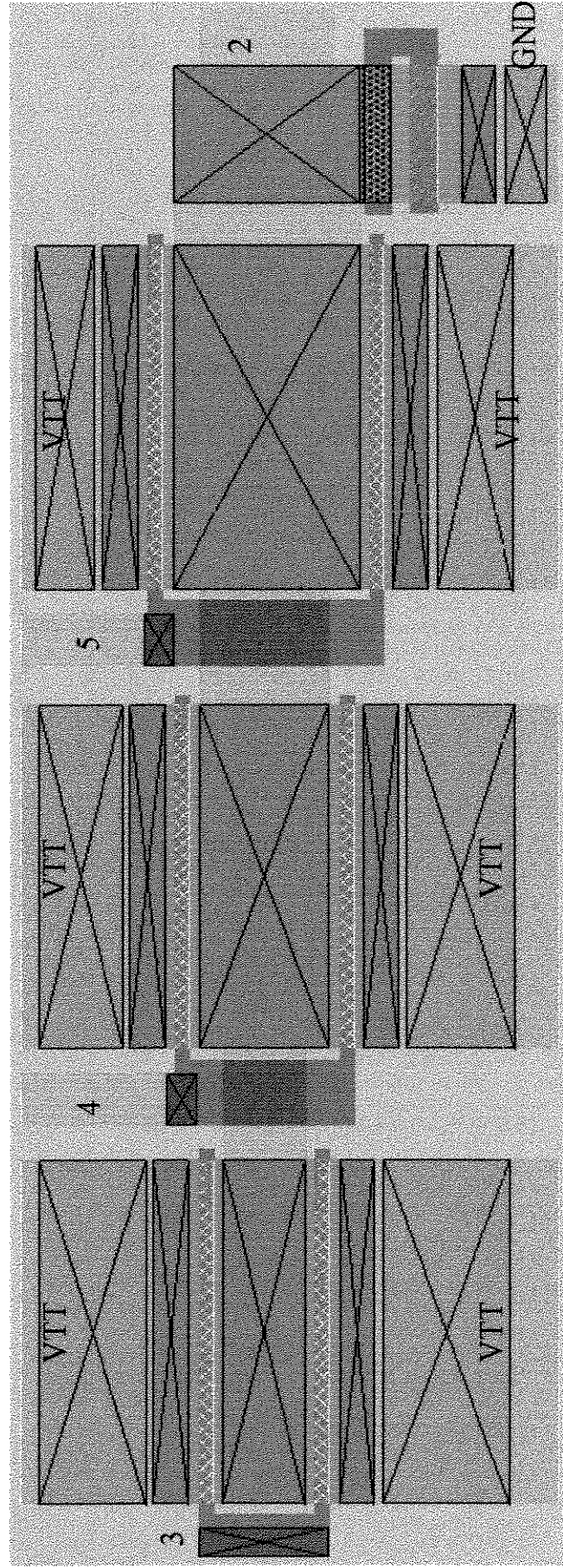
Estas portas e *buffers* correspondem diretamente aos subcircuitos usados para as simulações do FF e do contador de década no HSPICE. Estas células foram criadas para aproveitar-se os recursos de hierarquia do editor de *layouts* Magic, facilitando assim o trabalho de composição do *layout* do *chip* completo.

Além dos *layouts* das portas e *buffers*, também são mostrados os desenhos de alguns blocos funcionais do *chip*, como o circuito de proteção de entrada e o de saída, a porta de preset ( $R_{9(1)}$  e  $R_{9(2)}$ ), e dois dos flip-flops.



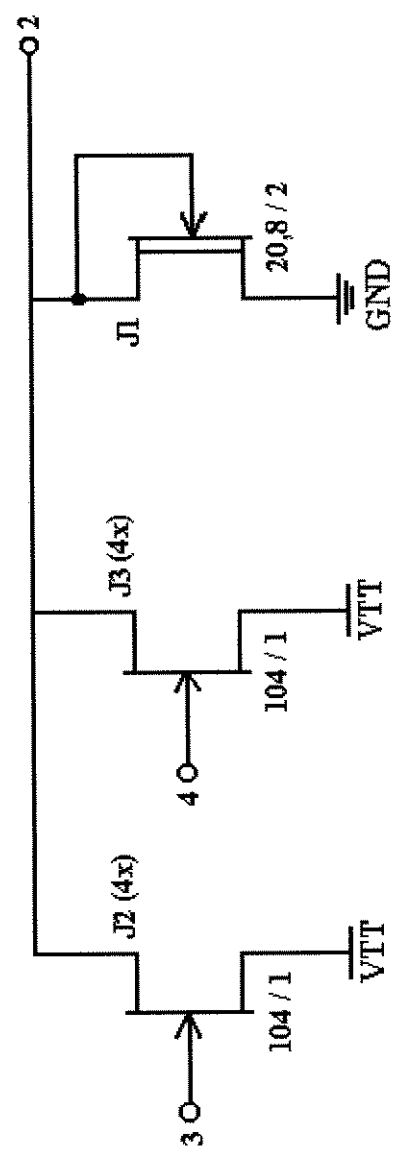
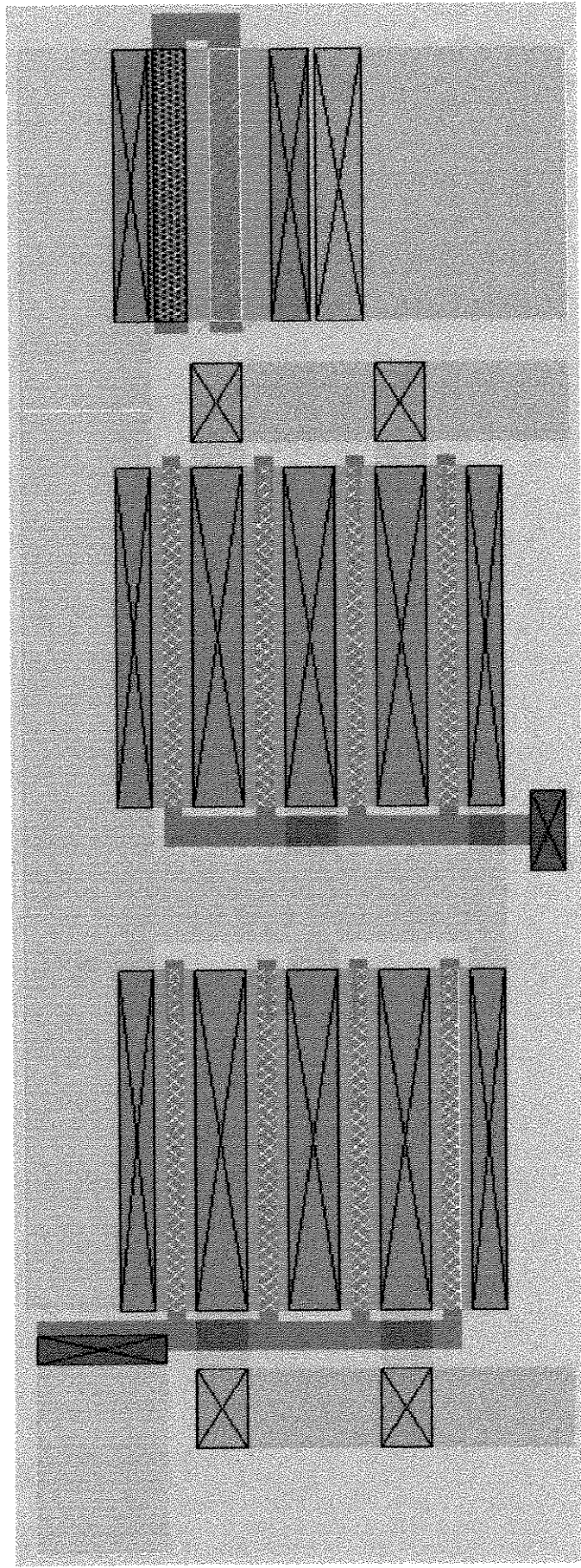
Dimensões em  $\mu\text{m}$ . A largura indicada nos transistores é a largura total.

### CÉLULA MAST1\_3.



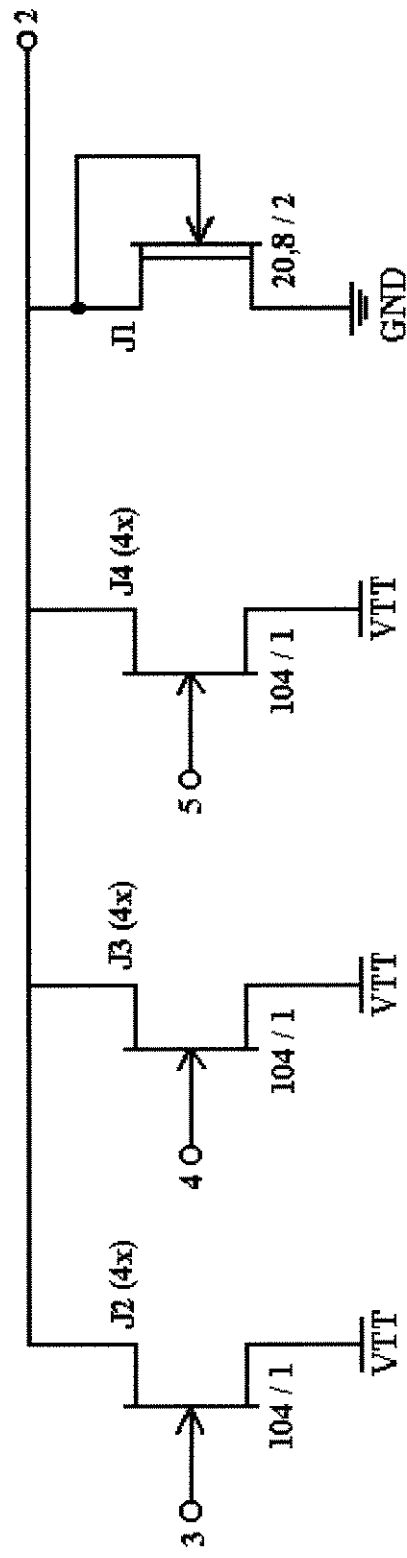
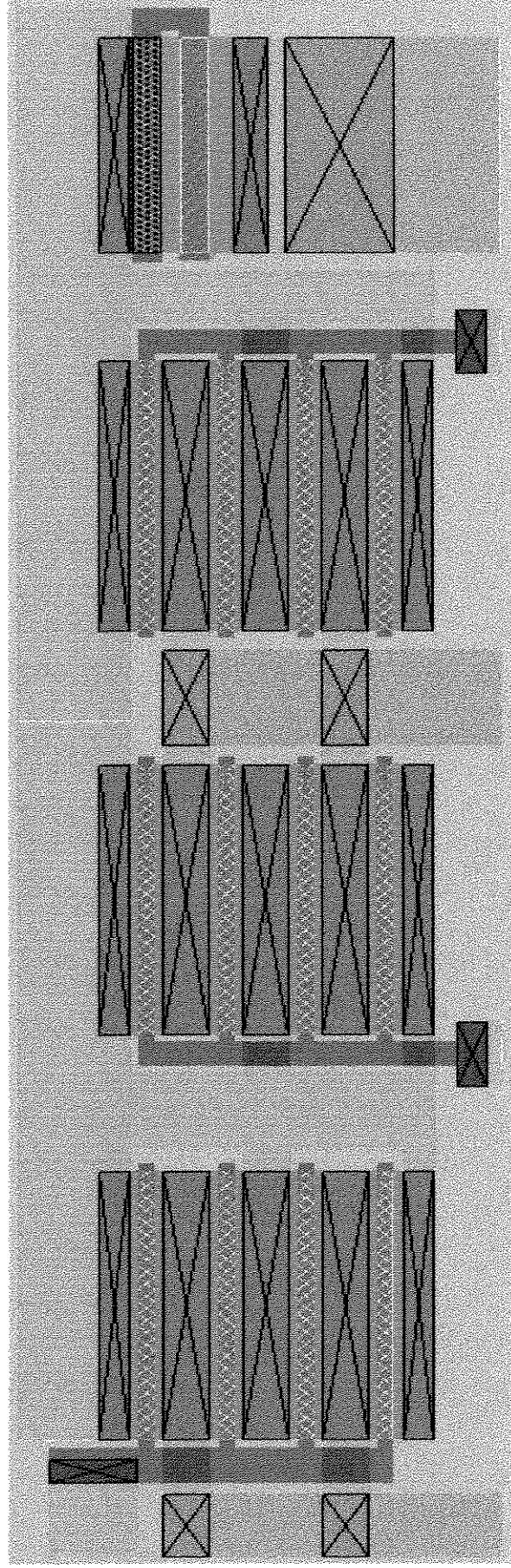
Dimensões em  $\mu\text{m}$ . A largura indicada nos transistores é a largura total.





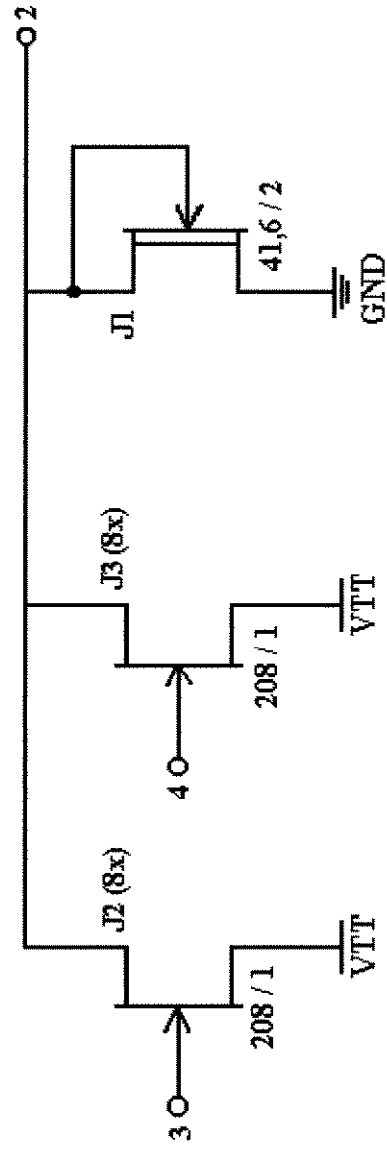
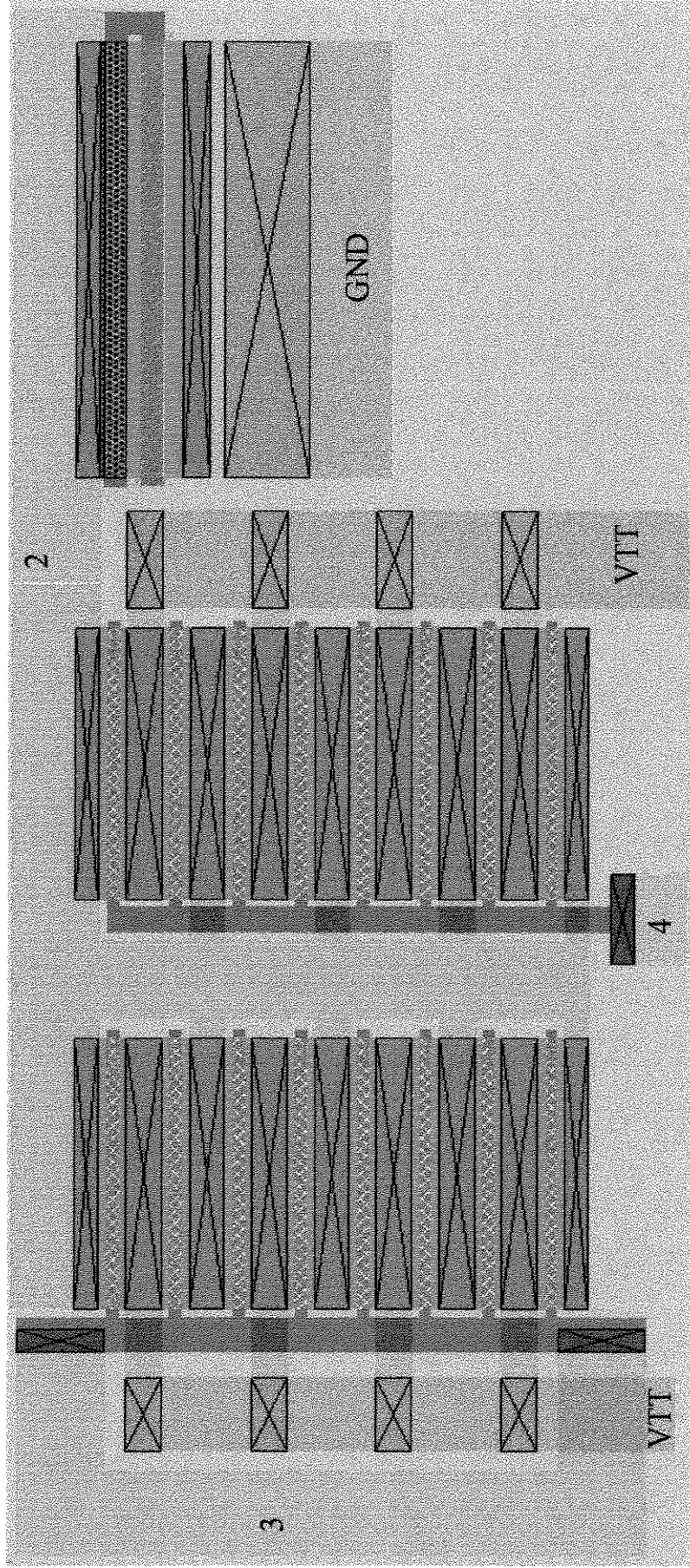
Dimensões em  $\mu\text{m}$ . A largura indicada nos transistores é a largura total.

CÉLULA MAST2\_3.



Dimensões em  $\mu\text{m}$  . A largura indicada nos transistores é a largura total.

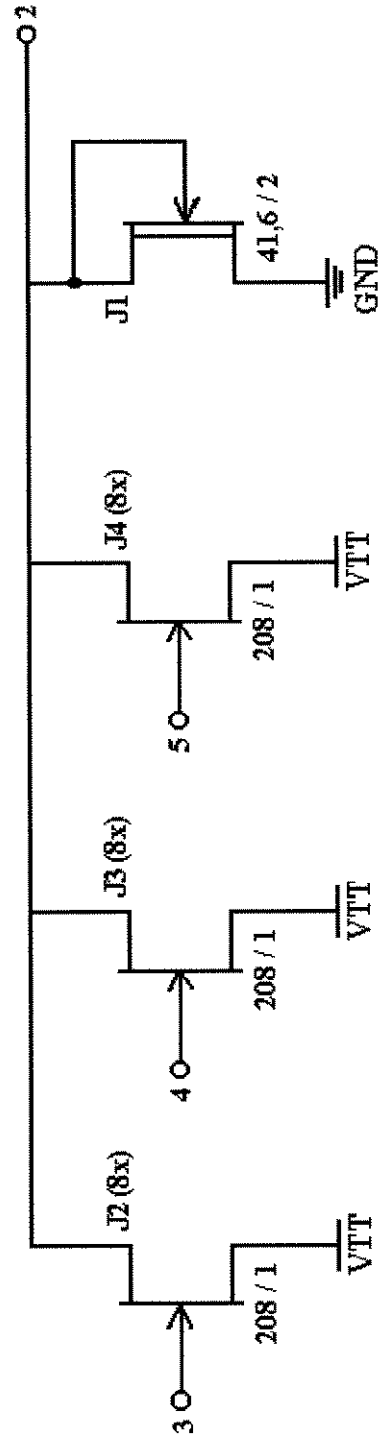
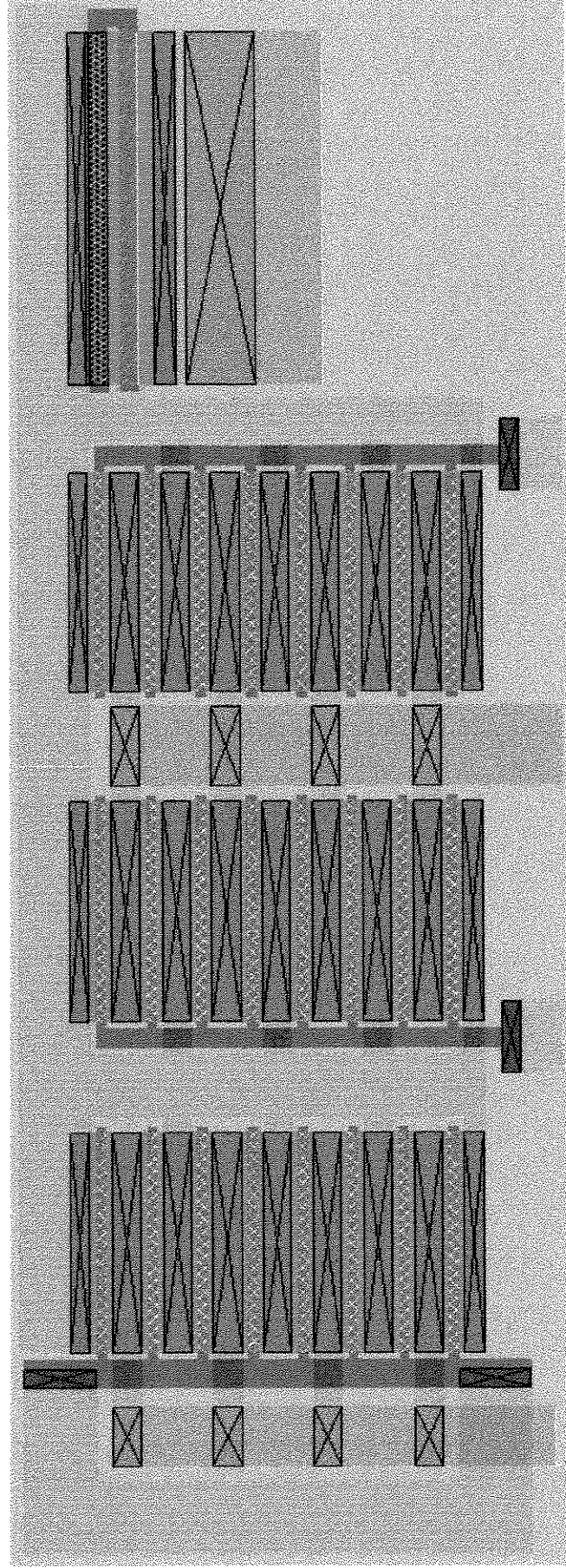
CÉLULA SLA1\_2.



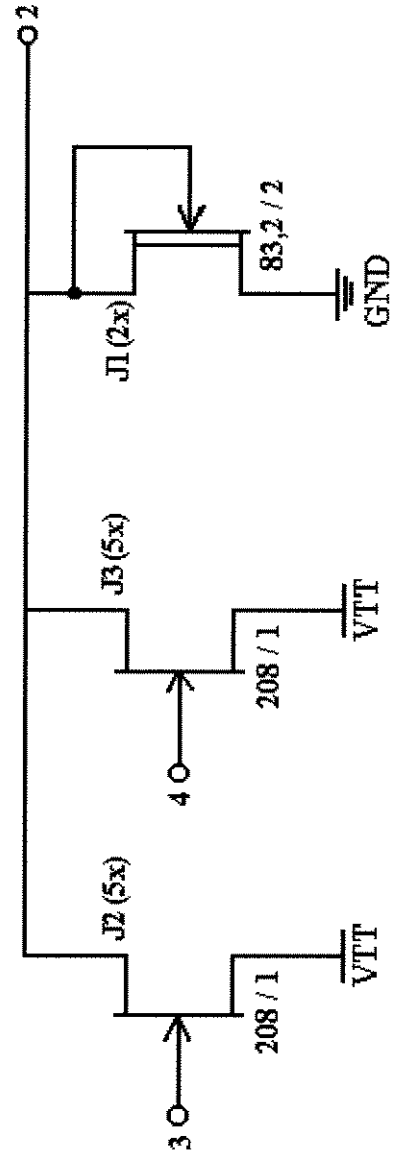
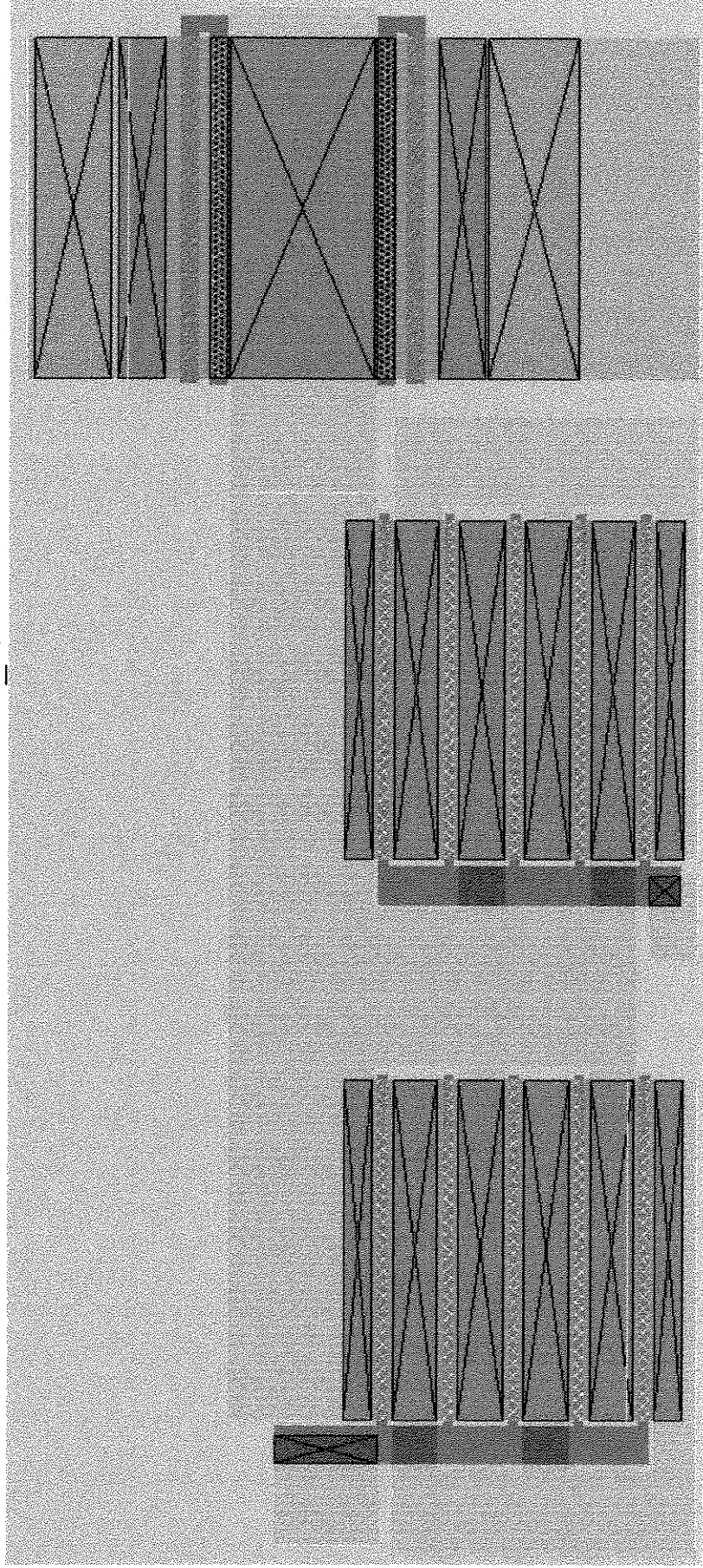
Dimensões em  $\mu\text{m}$ . A largura indicada nos transistores é a largura total.



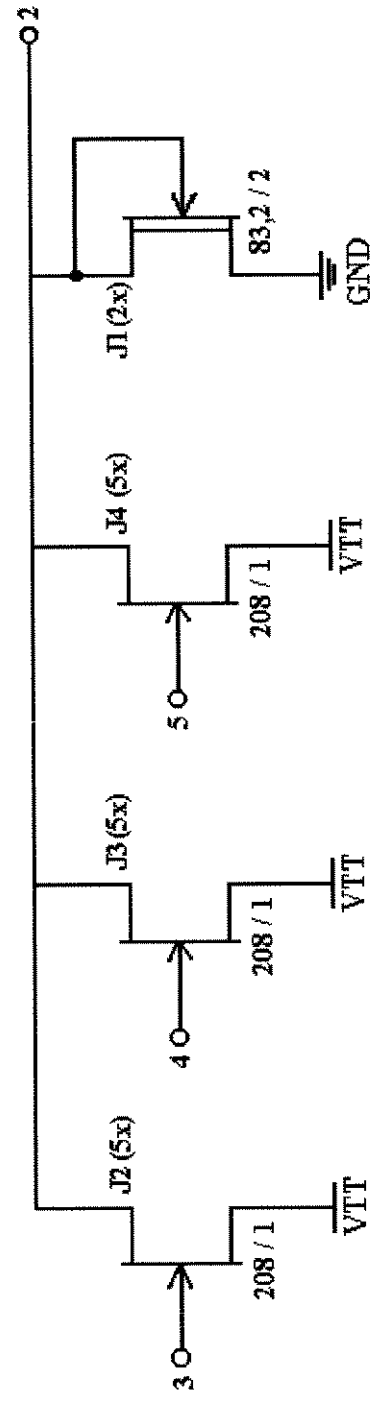
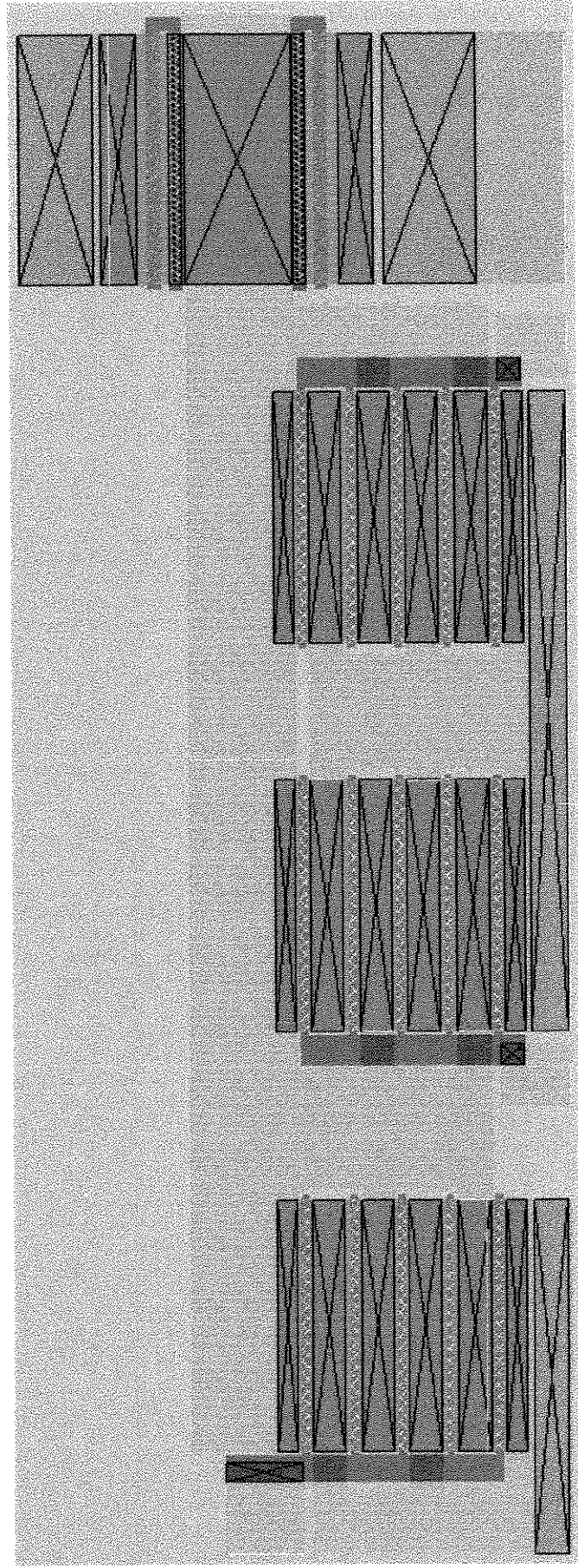
CÉLULA SLA1\_3.



Dimensões em  $\mu\text{m}$ . A largura indicada nos transistores é a largura total.



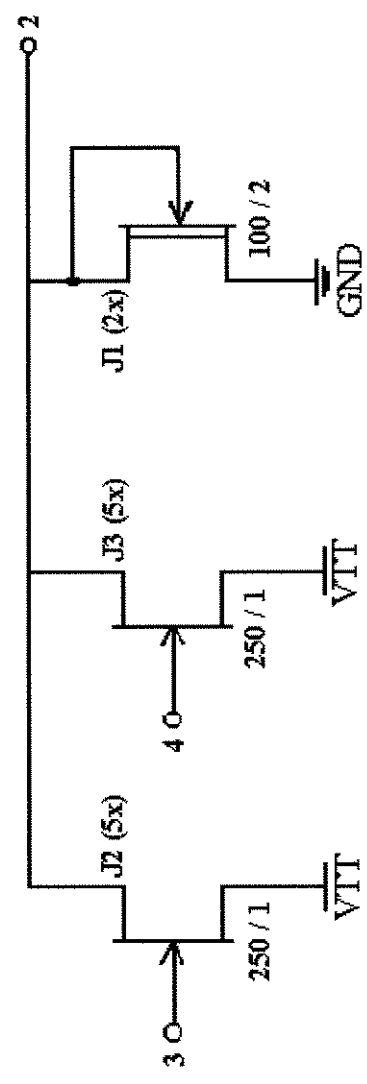
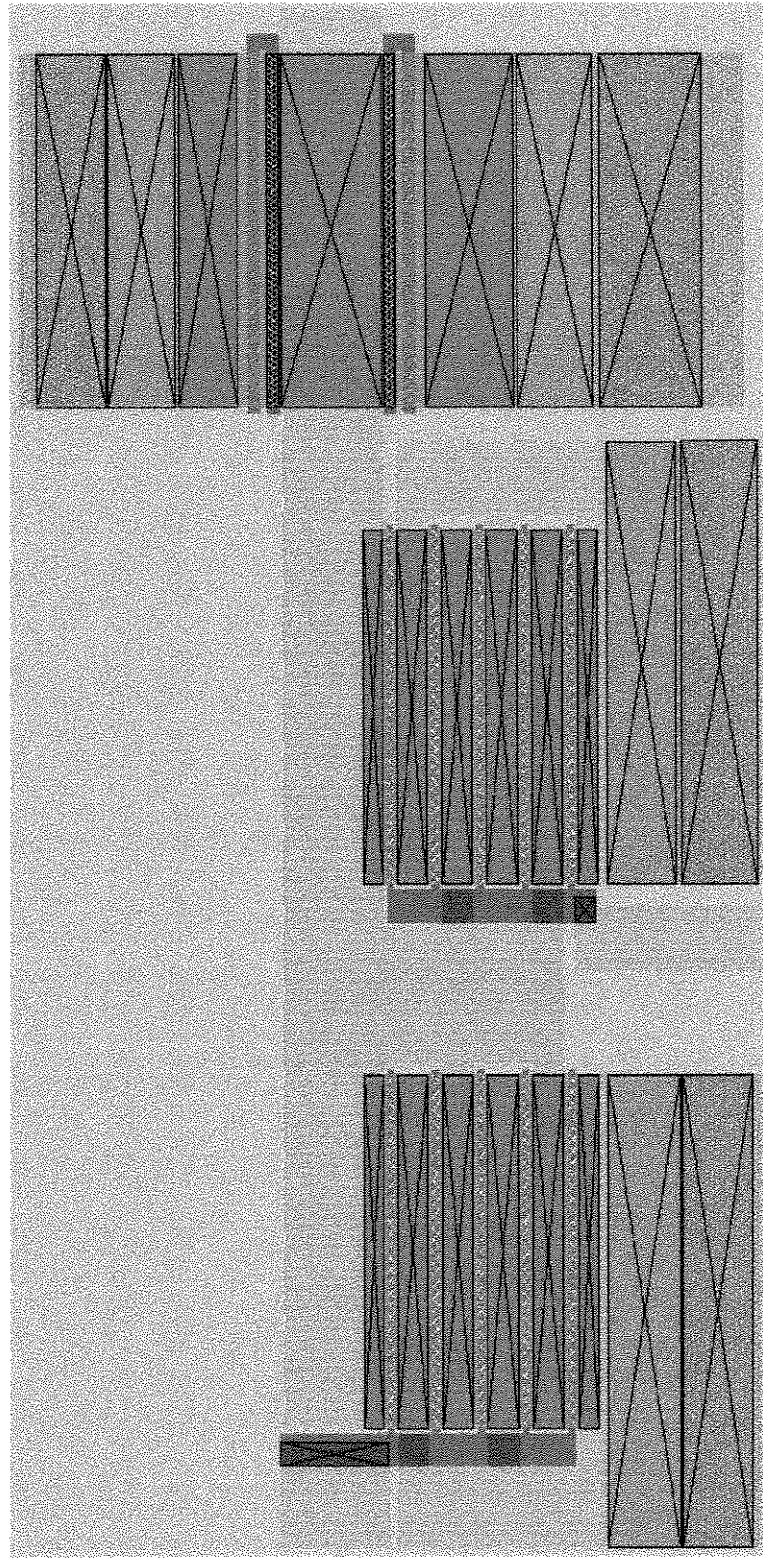
Dimensões em  $\mu\text{m}$  . A largura indicada nos transistores é a largura total.



Dimensões em  $\mu\text{m}$ . A largura indicada nos transistores é a largura total.

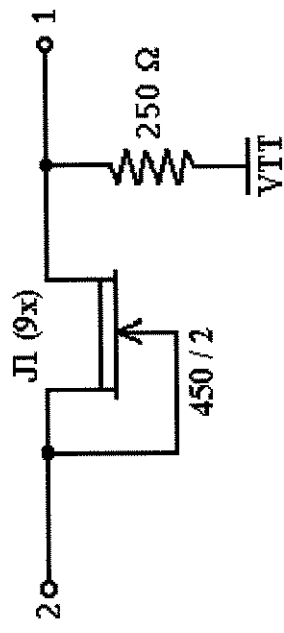
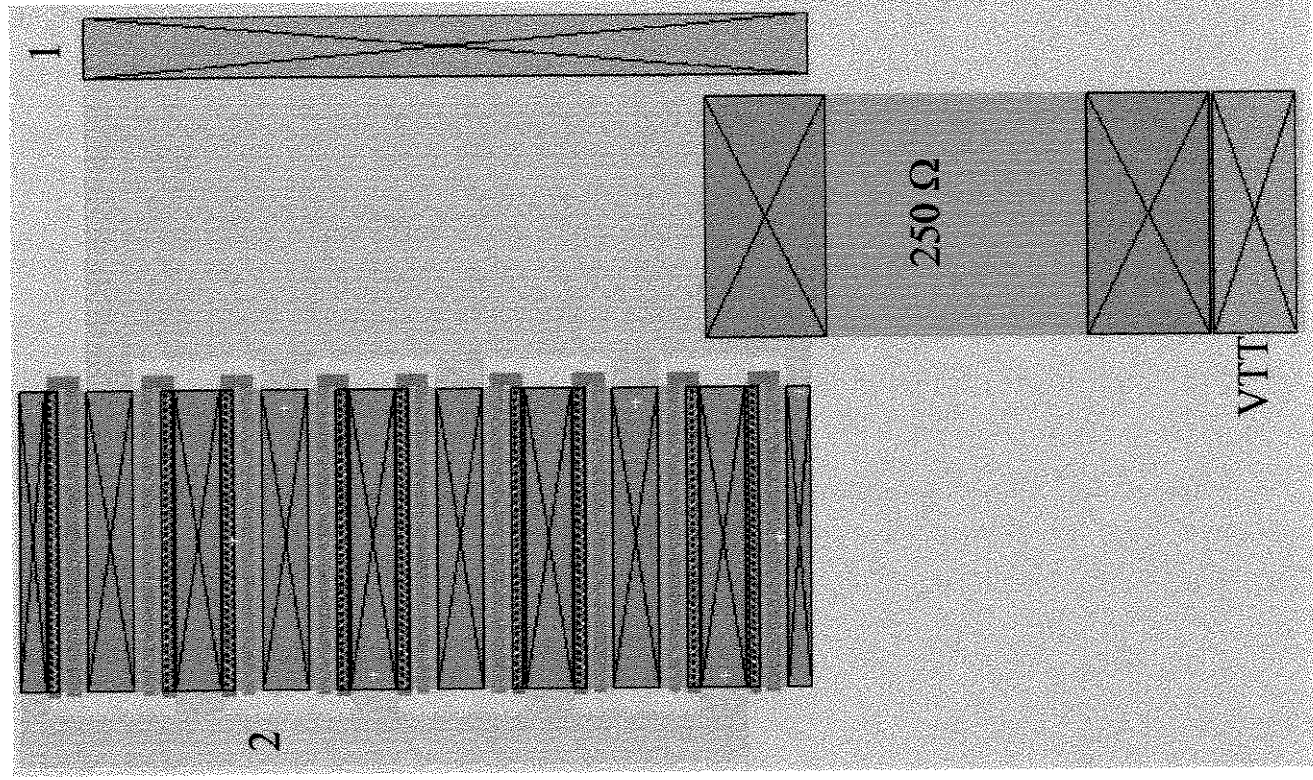


# CÉLULA SLAPRCL.



Dimensões em  $\mu\text{m}$ . A largura indicada nos transistores é a largura total.

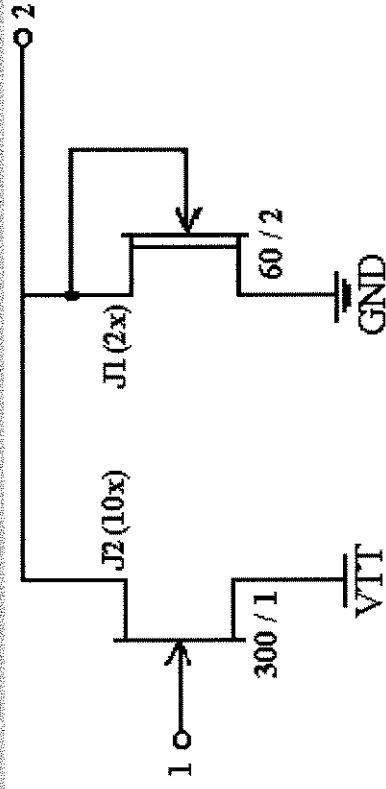
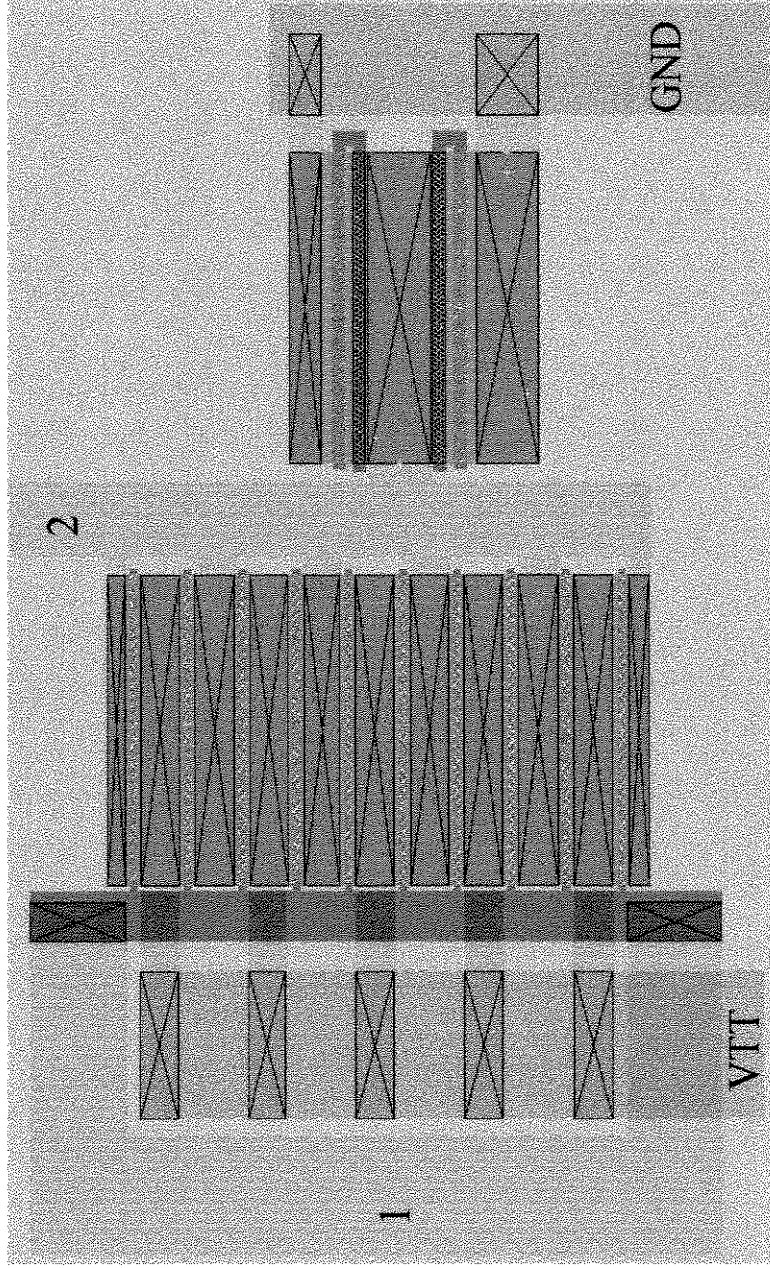
# CÉLULA INBUFFER.



Dimensões em  $\mu\text{m}$ .

A largura indicada no transistor é a largura total.

# CÉLULA CLKBUFFER.

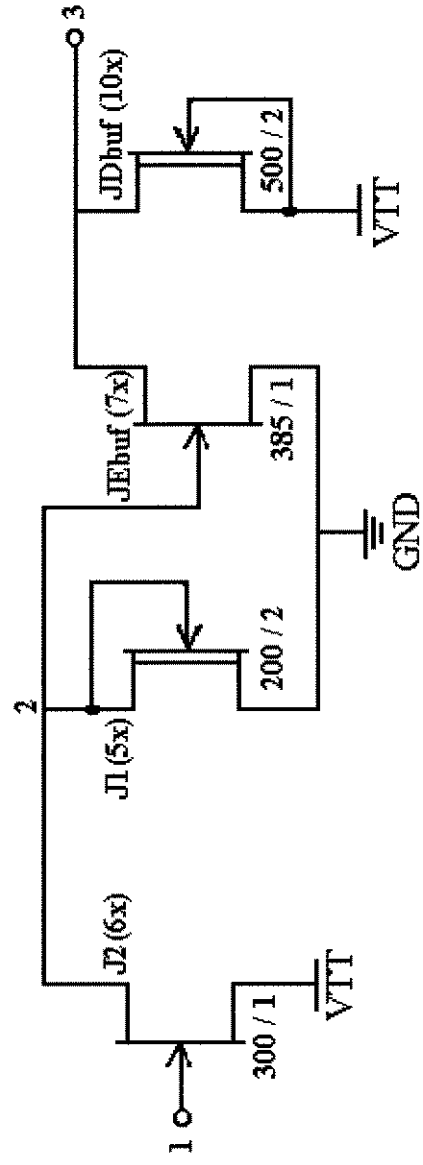
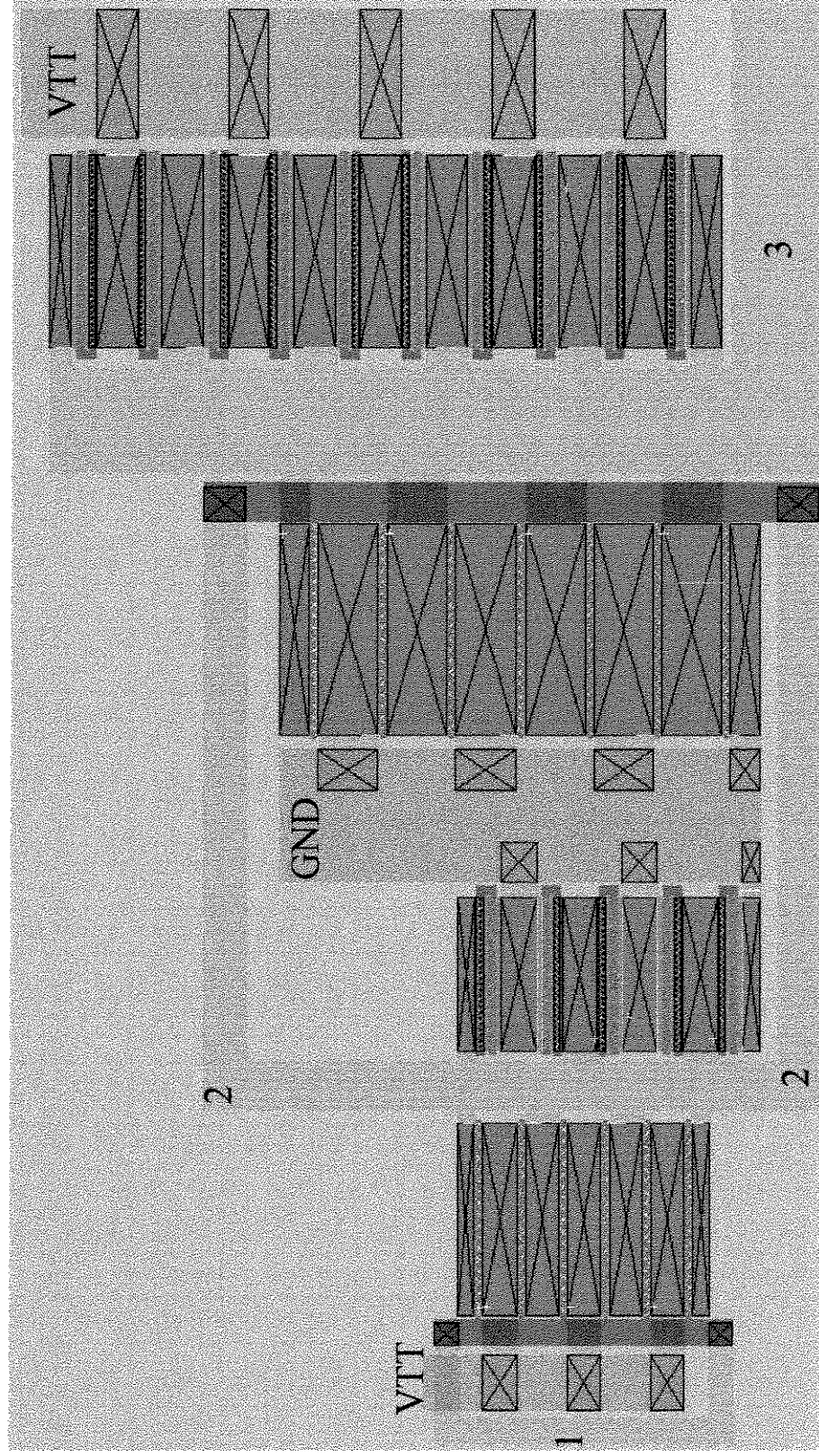


Dimensões em  $\mu\text{m}$ .

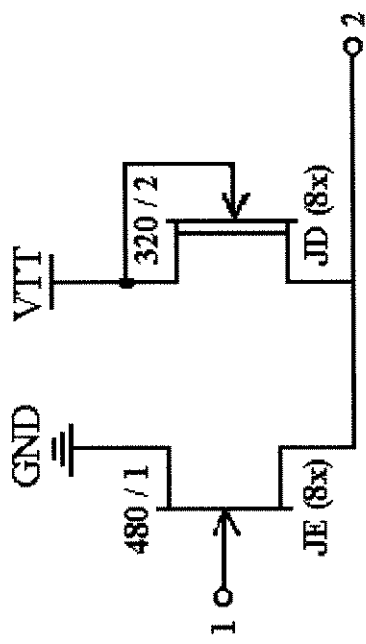
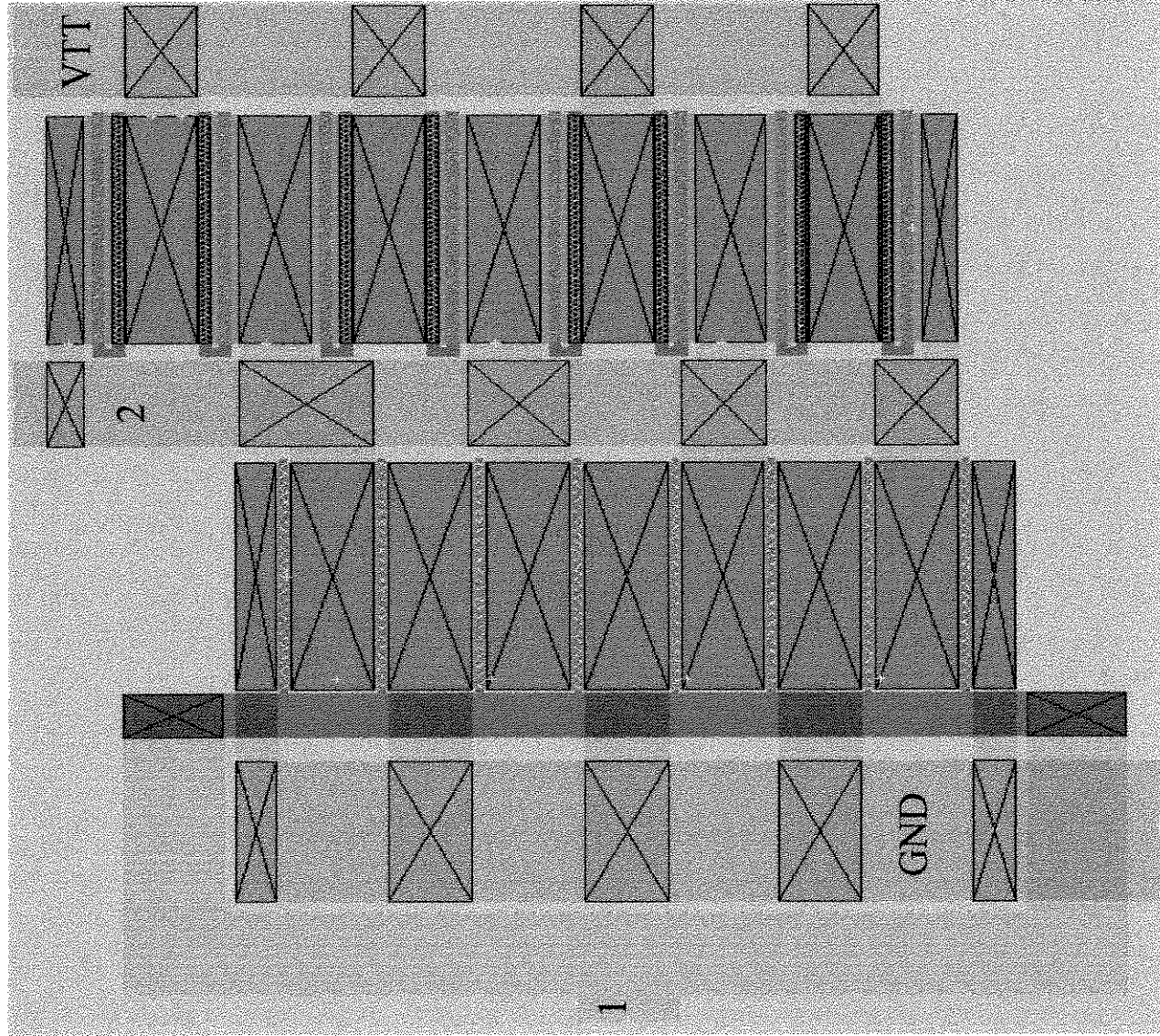
A largura indicada nos transistores é a largura total.



CÉLULA CLKBUFFER2.

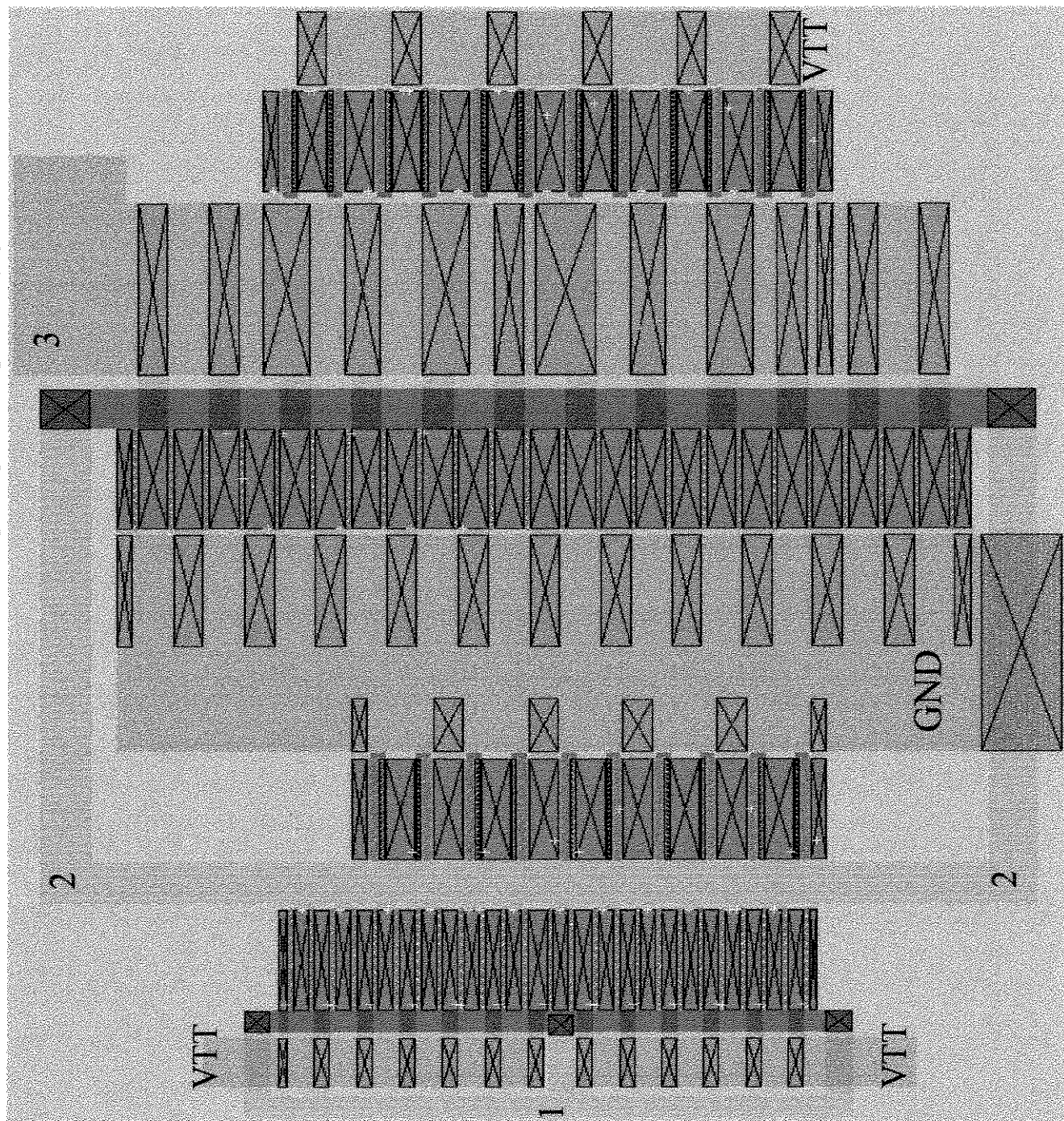
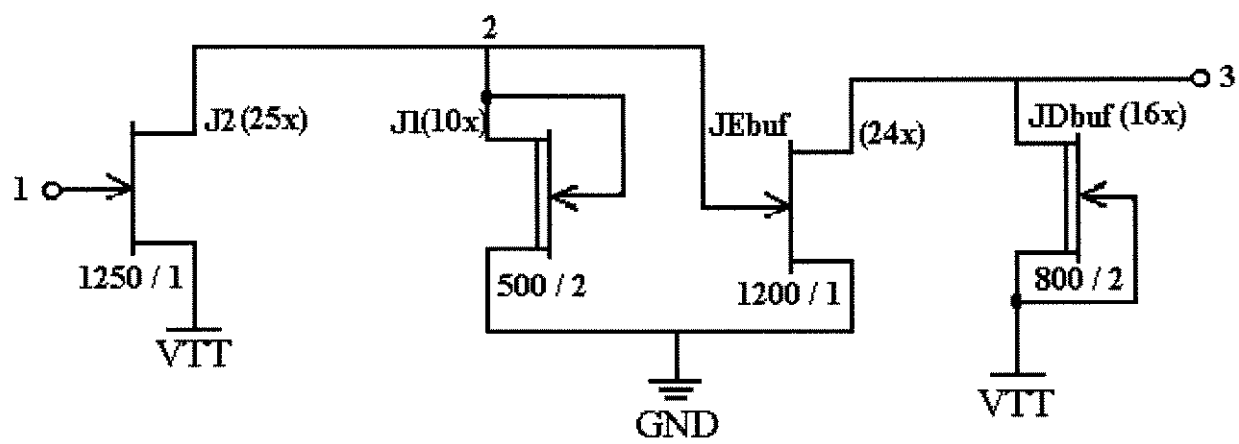


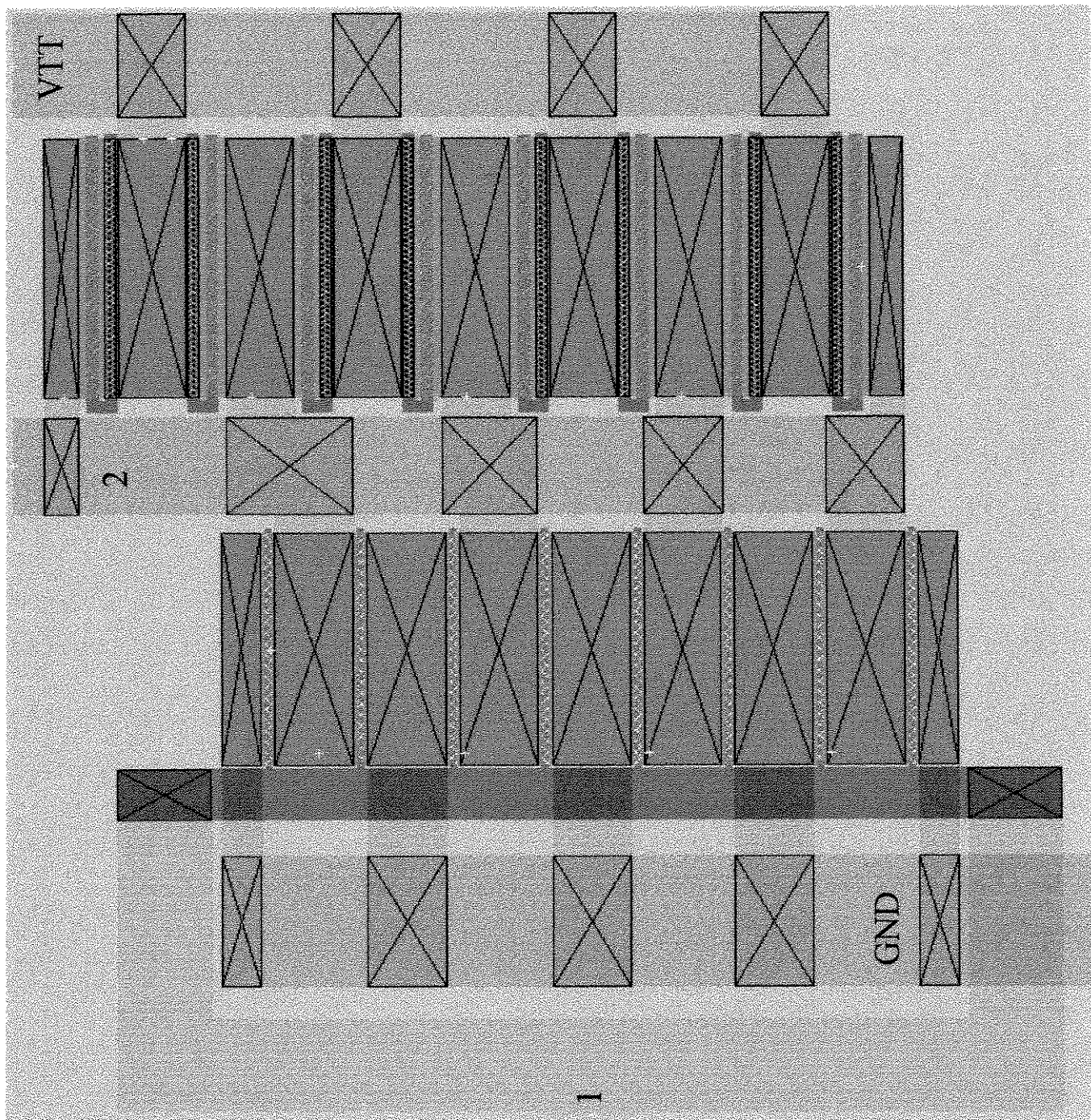
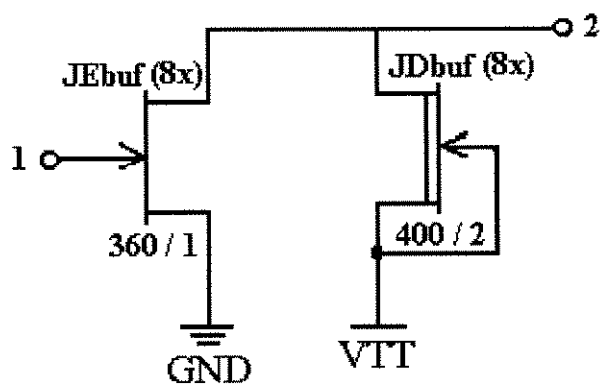
CÉLULA BUFFER.





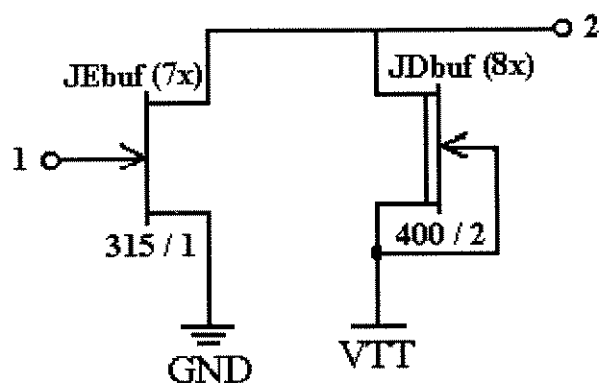
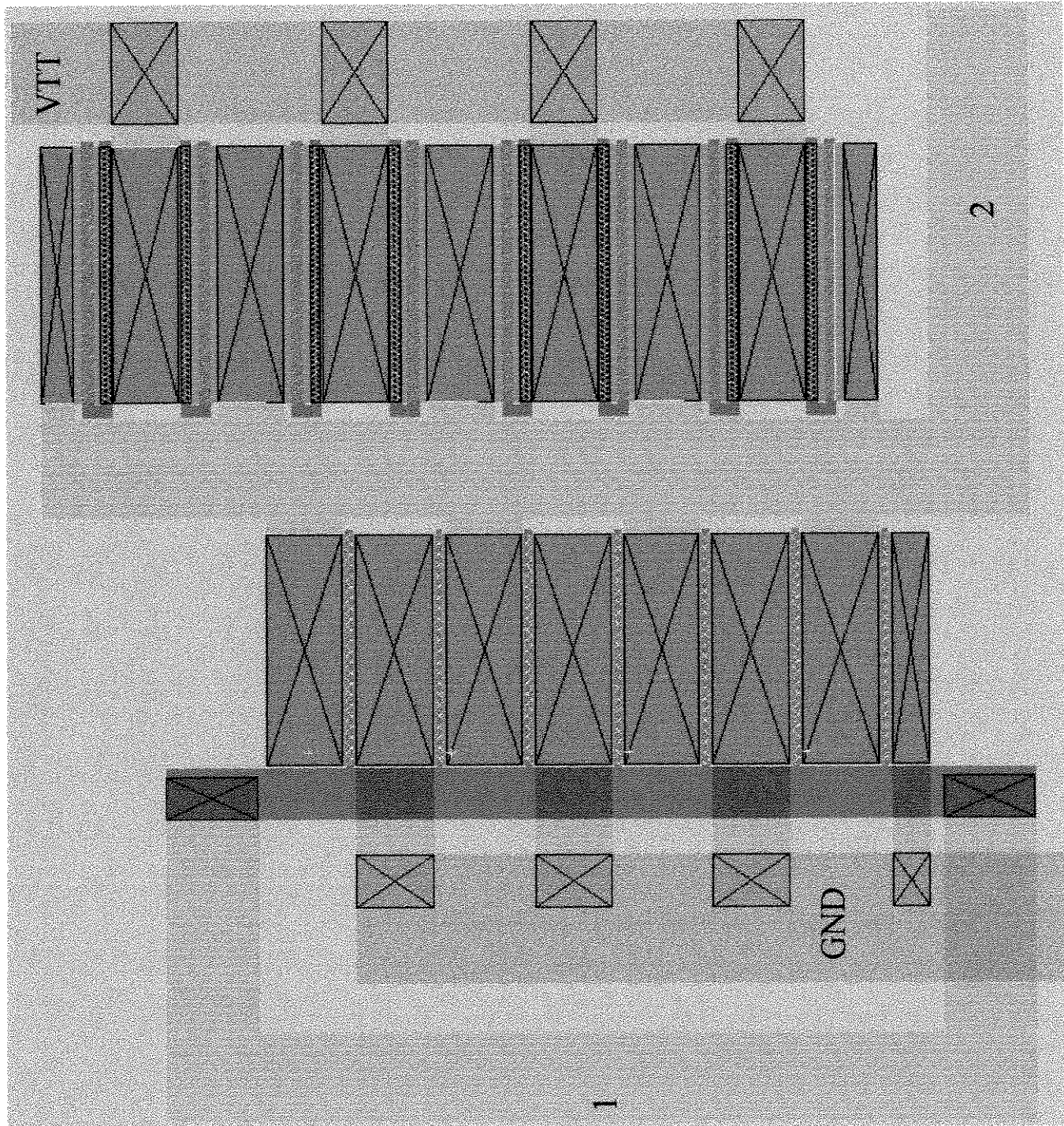
CÉLULA BUFFER2.



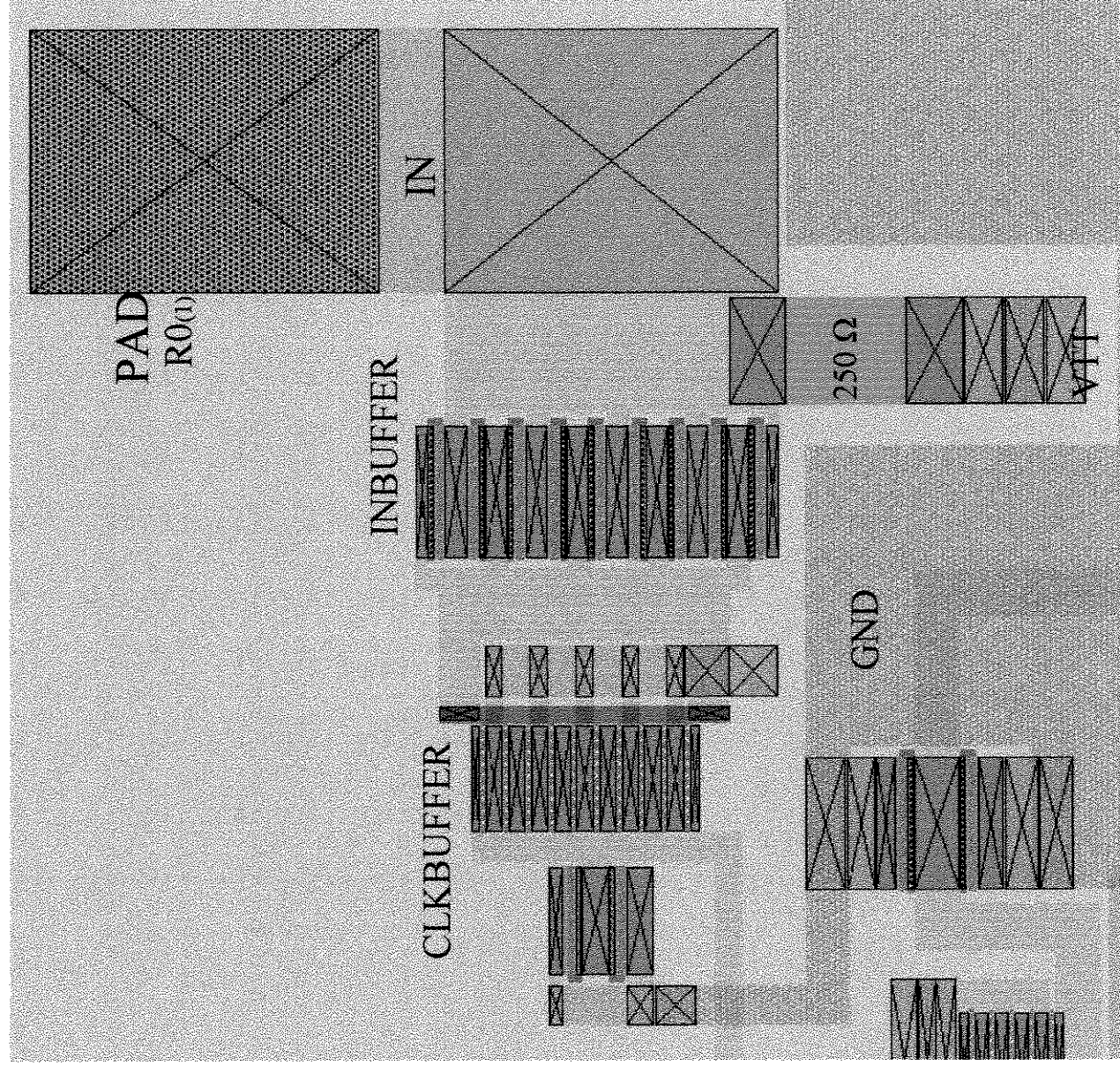




CÉLULA BUFFER4.



CIRCUITO DE ENTRADA DA PORTA DE *CLEAR* (detalhe).





CÉLULA BUFFER2 DA SAÍDA QC (detalhe).

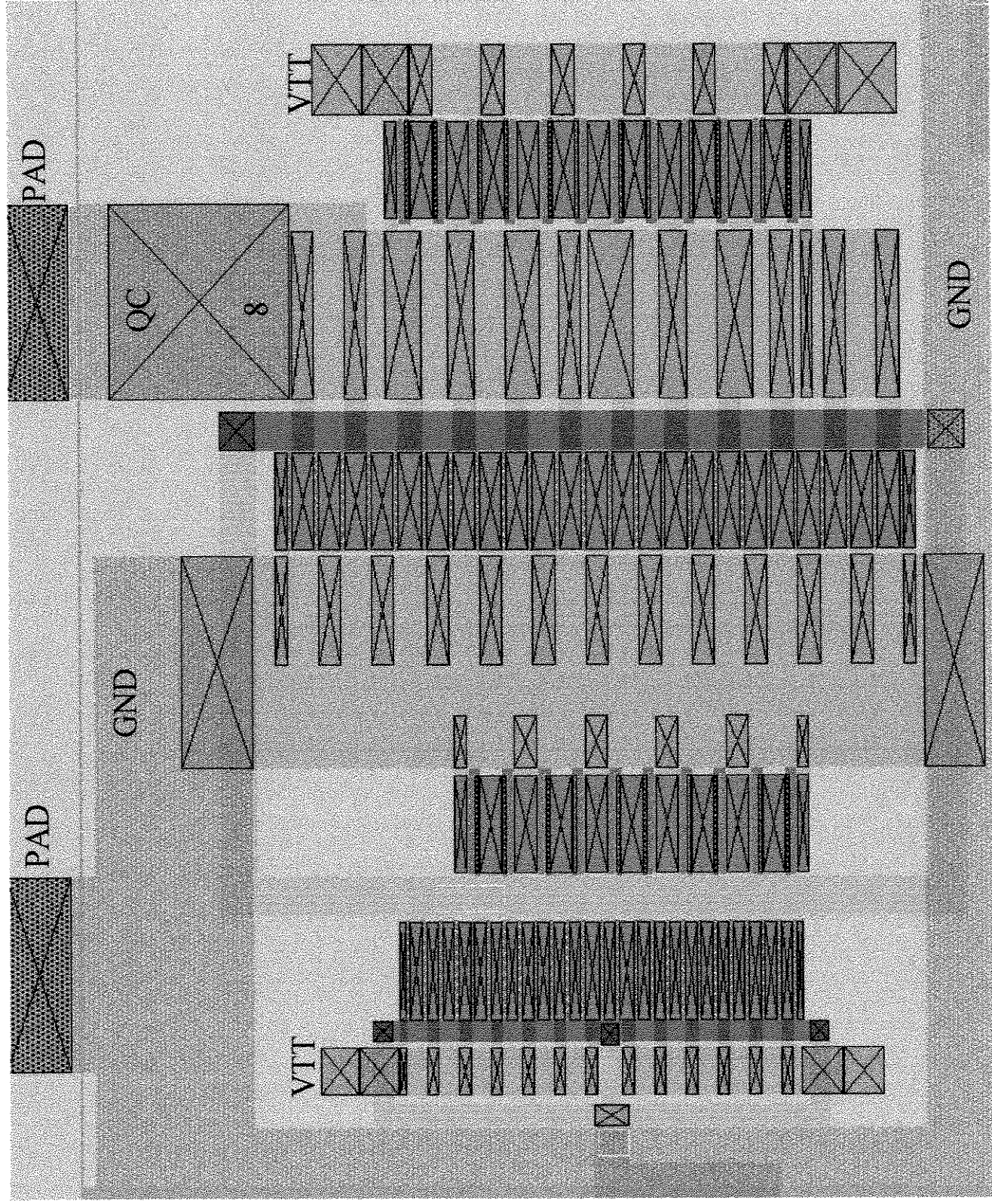
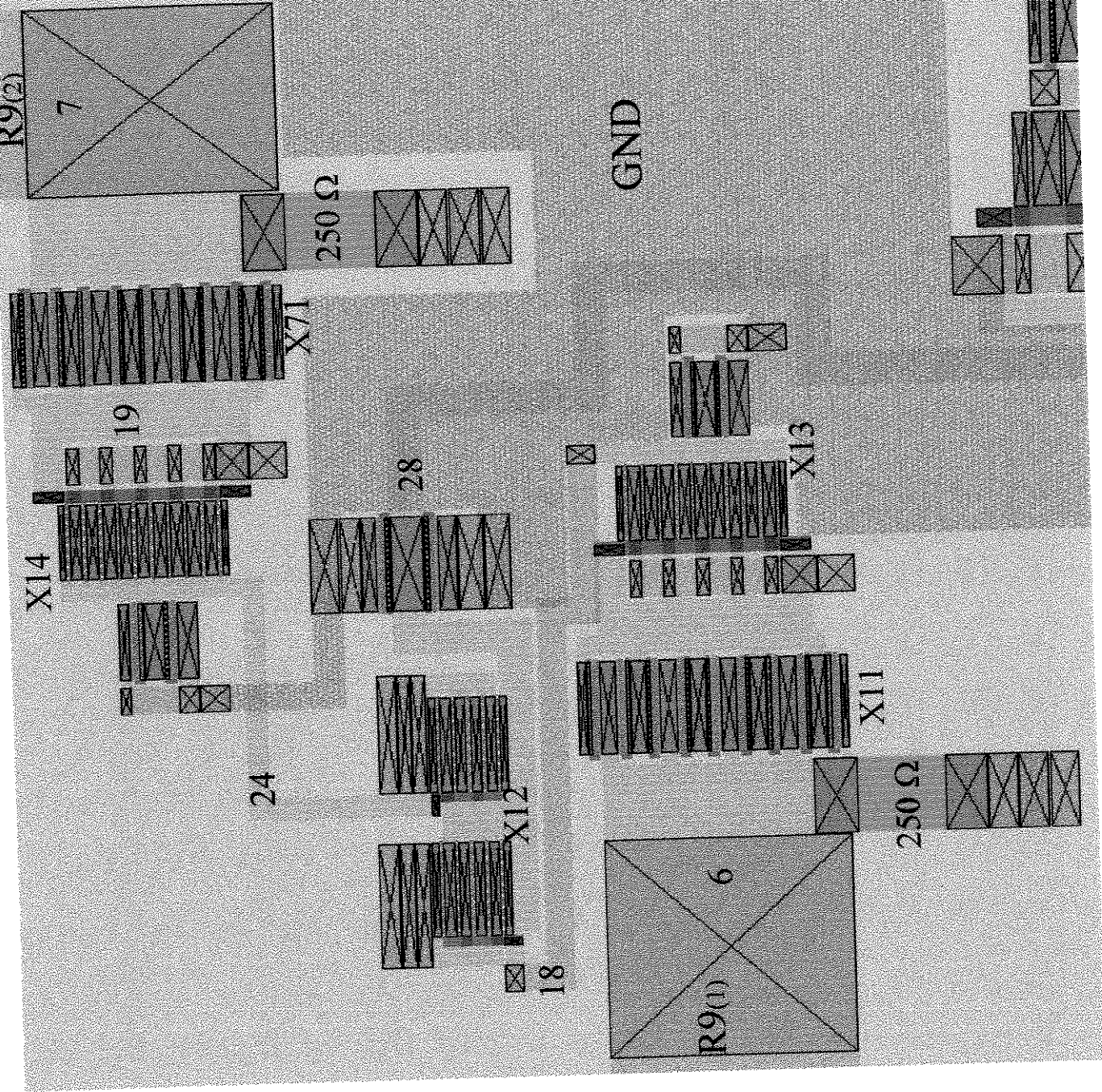
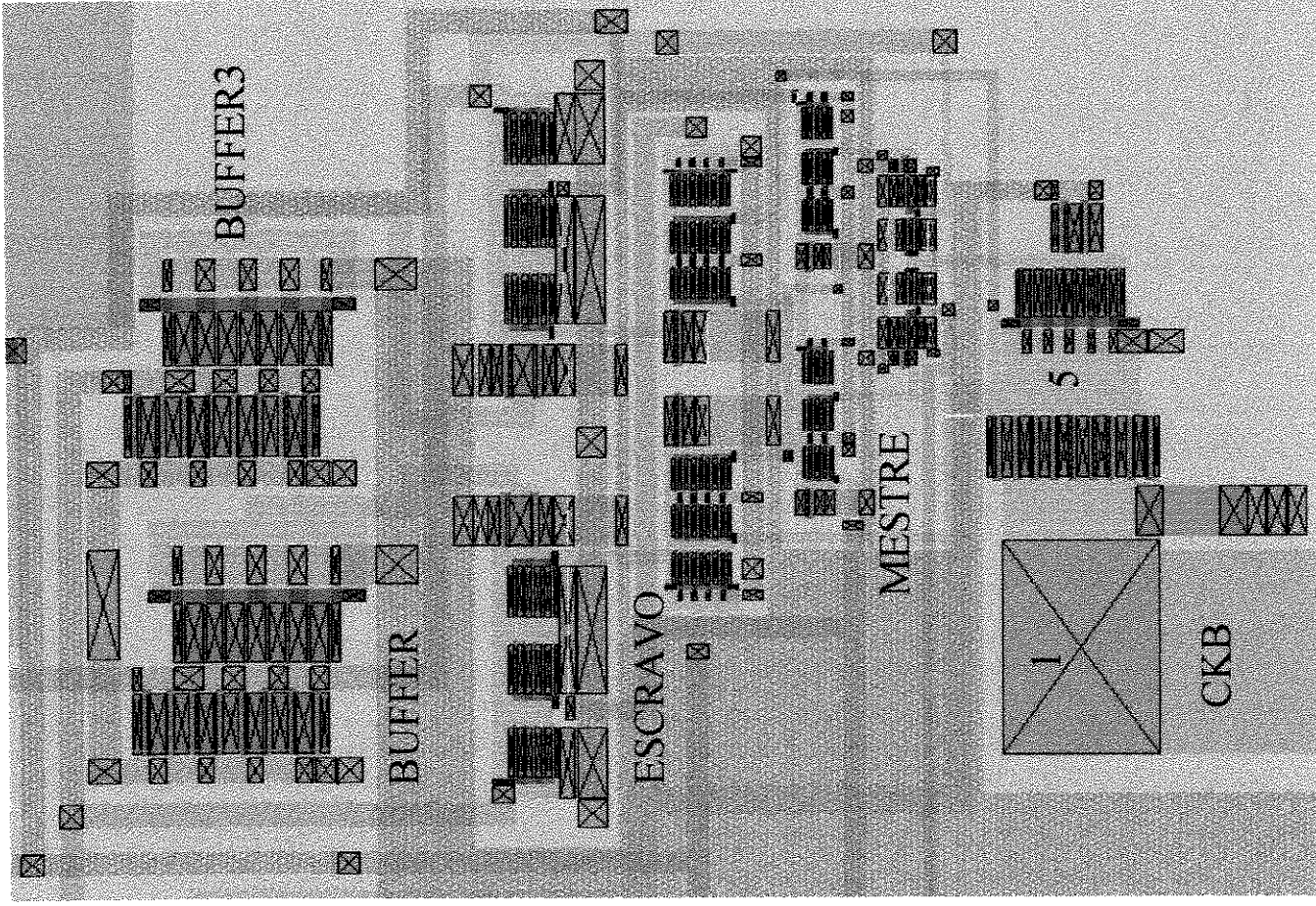


Diagram for Problem 10: A circuit with a  $250\ \Omega$  resistor in series with a parallel combination of a  $7\ \Omega$  resistor and a dependent current source labeled  $R9(2)$ .

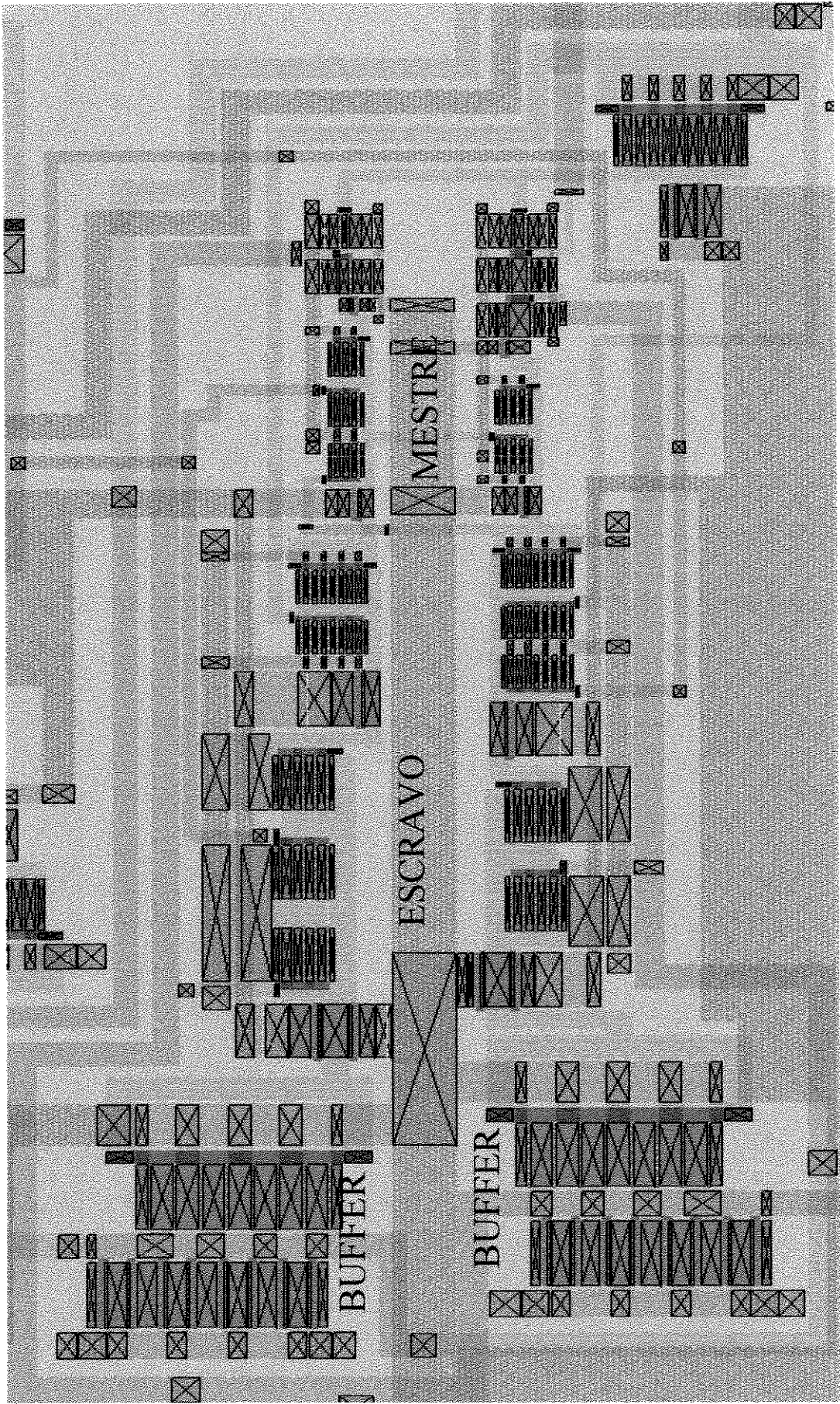




FLIP-FLOP 4.



FLIP-FLOP 2.





## APÊNDICE 5.

### O Novo Processo HGaAs-IV.

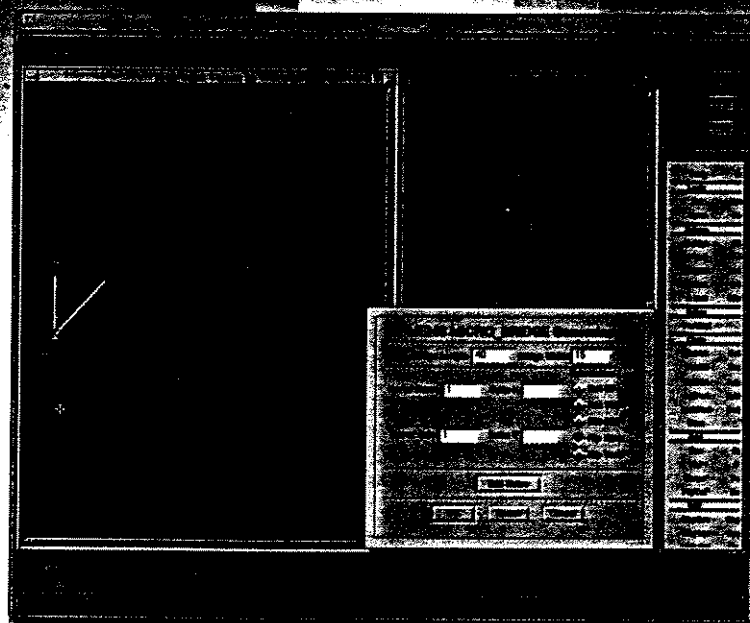
A partir de 1998, o CMP passou a oferecer o novo processo HGaAs-IV ( $L_g = 0,4 \mu\text{m}$ ) da Vitesse, para CIs digitais e mistos (analógico-digitais) de alta velocidade em GaAs, em escala VLSI. Este novo processo veio substituir o processo digital HGaAs-III ( $L_g = 0,6 \mu\text{m}$ ), cuja última rodada foi realizada em outubro de 1997, tendo sido então descontinuado. HGaAs-IV trouxe melhorias nos MESFETs e interconexões, suportando DCFL para projetos VLSI (frequências de até 800 MHz), e SCFL para componentes MSI (até 10 GHz) [15].

Os *design kits* para o processo HGaAs-III eram distribuídos pelo CMP (alguns gratuitamente), para uso nas seguintes plataformas de CAD: Cadence, Compass e HSPICE/Magic. O *design kit* para ferramentas Compass incluía bibliotecas de células-padrão em HGaAs-III, desenvolvidas pela empresa americana SPEC. Os *design kits* para HGaAs-IV estão disponíveis exclusivamente para plataforma Cadence.

low volume production in a MPW run

CIRCUITS  
MULTI PROJETS

*CMP*



Engineering kit for Micro Electro Mechanical Systems (MEMS)  
developed at CMP

## FRENCH MPC ACTIVITY REPORT 1997

46, Avenue Félix Viallet  
38031 GRENOBLE  
FRANCE

HS/SPI - MEMS - MEMS/SERIES

**G97-3 run**

Technology: **VITESSE GaAs 0.6 (H-GaAs3)**  
 Starting date: 20 October 1997  
 Delivery of chips: 17 March 1998

| Institution           | Town     | Count. | Circuit's name    | Area mm2         | ERI* | Circuit's function                                                                                                                                                                                                                                                                                                                       |
|-----------------------|----------|--------|-------------------|------------------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unicamp-FEE           | Campinas | BR     | fapesp48          | 8.69             | E    | An all digital-frequency divider and decade counter circuit. Frequency division is possible by any integer in the range from 2 to 10. The topology is based upon the TTL 7490 topology. The circuit is expected to operate for input frequencies up to 1 GHz. It employs 191 MESFETs but many of these are made up of gates in parallel. |
| EPFL-DE-LEG           | Lausanne | CH     | circuit6          | 3.91             | R    | Test of comparator.                                                                                                                                                                                                                                                                                                                      |
| EPFL-DE-LEG           | Lausanne | CH     | circuit_rk        | 7.38             | R    | Test of a prescaler, a Wallace adder.                                                                                                                                                                                                                                                                                                    |
| U. Middlesex          | London   | UK     | mumecg4_2         | 6.83             | R    | A complete transmitter & 1.6 GHz clock generation circuit. This is a complete Serialiser chip based on the test structures manufactured previously.                                                                                                                                                                                      |
| <b>3 Institutions</b> |          |        | <b>4 Circuits</b> | <b>26.81 mm2</b> |      |                                                                                                                                                                                                                                                                                                                                          |

**P97-1 run**

Technology: **PML GaAs 0.2 (ED02AH)**  
 Starting date: 11 February 1997  
 Delivery of chips: 2 September 1997

| Institution    | Town               | Count. | Circuit's name | Area mm2 | ERI* | Circuit's function                                                                                                                                                         |
|----------------|--------------------|--------|----------------|----------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ENSEA          | Cergy              | FR     | fond_deph      | 1.50     | E    | Monolithic active delay line, for use about 4 GHz (1.5 ns).                                                                                                                |
| ENST           | Paris              | FR     | morin1         | 6.00     | E    | The circuit's function is the measurement of the reflection coefficient of waveguide loads at 94 GHz. The circuit is made with fin line, diodes, resistors and capacitors. |
| IEMN           | Villeuneuve-d'Ascq | FR     | pmlself        | 1.50     | R    | This circuit is a high Q factor active inductor at 22 GHz using a 2 x 30 µm wide Hemt.                                                                                     |
| IRCOM          | Limoges            | FR     | merge4         | 3.00     | R    | Narrow tunable microwave filter using active inductors.                                                                                                                    |
| IRCOM          | Limoges            | FR     | dephaseur      | 3.00     | R    | Broadband analog tunable microwave amplifier/phase-shifter.                                                                                                                |
| IRCOM          | Limoges            | FR     | fixe           | 1.50     | R    | First-order recursive filter non-tunable at microwaves.                                                                                                                    |
| IRCOM          | Limoges            | FR     | tunable        | 3.00     | R    | Active first-order tunable microwave recursive filter.                                                                                                                     |
| TIMA           | Grenoble           | FR     | cimcs3_pml     | 6.00     | R    | Micromachining test structures for characterization.                                                                                                                       |
| Univ. of Leeds | Leeds              | UK     | tmnvco         | 6.00     | E    | 38 GHz VCO with tunable matching network.                                                                                                                                  |
| Univ. Firenze  | Firenze            | IT     | lay_mg_01      | 1.50     | R    | Test of coupling between spiral inductor. Test of diode. Test of 3 transistors of different size (HEMT).                                                                   |
| Nanyang T. U.  | Singapore          | SG     | ntu            | 3.00     | R    | Feedback amplifier.                                                                                                                                                        |
| Nanyang T. U.  | Singapore          | SG     | mixer1         | 3.00     | R    | Mixer.                                                                                                                                                                     |
| IEF            | Warsaw             | PL     | ipetest        | 1.50     | E    | Circuit for testing transistor parameters and for the on-wafer prober calibration.                                                                                         |
| IEF            | Warsaw             | PL     | bamp06         | 1.50     | E    | Broadband low noise amplifier. Frequency range approx. 0.5 - 6 GHz.                                                                                                        |
| Nat. Taiwan U. | Tapei              | TW     | 2lnamp         | 1.50     | R    | Two staged low noise amplifier for frequency 2.4 GHz.                                                                                                                      |
| Nat. Taiwan U. | Tapei              | TW     | localamp       | 1.50     | R    | Local signal amplifier for frequency 2.2 GHz.                                                                                                                              |
| Nat. Taiwan U. | Tapei              | TW     | mixer2         | 6.00     | R    | A coplanar waveguide mixer operate at 60 GHz which use a twin-dipole antenna on RF signal-in.                                                                              |
| Nat. Taiwan U. | Tapei              | TW     | mixer180       | 1.50     | R    | A mixer circuit with hybrid 180° for up or down frequency.                                                                                                                 |
| Nat. Taiwan U. | Tapei              | TW     | poweramp       | 1.50     | R    | Two parallel power amplifier for frequency at 2.4 GHz.                                                                                                                     |
| Nat. Taiwan U. | Tapei              | TW     | vco            | 1.50     | R    | Voltage Controlled Oscillator (VCO) works at 2.2 GHz.                                                                                                                      |
| NCCU-WC-DEE    | Chiayi             | TW     | inputmatch     | 1.50     | R    | A 12 GHz two-stage low noise amplifier.                                                                                                                                    |

## Referências Bibliográficas.

- 1 - National Semiconductor, "F100K ECL Logic Databook and Design Guide", 1990.
- 2 - I. Deyhimy (Vitesse), "Gallium Arsenide Joins the Giants", IEEE Spectrum, February 1995, pgs. 33-40.
- 3 - K. Lehovec & R. Zuleeg (McDonnell Douglas), "Analysis of GaAs FET's for Integrated Logic", IEEE Trans. on Electron Devices, vol. ED-27, n° 6, June 1980, pgs. 1074-1091.
- 4 - P. S. Lassen, "High-Speed GaAs Digital Integrated Circuits for Optical Communication Systems", PhD Thesis, Technical University of Denmark, Electromagnetics Institute, May 1993.
- 5 - C. Garnier, "Curso de Projeto de CIs em GaAs", ministrado na Unicamp em dezembro de 1994.
- 6 - M. Shur, "GaAs Devices and Circuits", Plenum Press, New York and London, 1987.
- 7 - "The Forecast for GaAs Integrated Circuits", Compound Semiconductor, vol. 1, n° 2, September/October 1995, pgs. 21-24.
- 8 - "GaAs Transistor With Improved Radiation Hardened Characteristics", Compound Semiconductor, Nov./Dec. 1995, vol. 1, n° 3, pgs. 10-11.
- 9 - R. M. F. Custódio, "Análise e Projeto de Portas Lógicas em Tecnologia MESFET de GaAs", Relatório de Estágio de Iniciação Científica Orientado por Jacobus W. Swart, FEEC/UNICAMP, Abril de 1996.
- 10 - J. W. Swart & A. S. Lujan (DSIF/UNICAMP), "Integração de Processos: MESFET em GaAs", Anais da V Oficina de Microeletrônica, Campinas, SP, Julho de 1996.
- 11 - S. I. Long & S. E. Butner, "Gallium Arsenide Digital Integrated Circuit Design", McGraw-Hill, 1990.
- 12 - Vitesse Semiconductor Corp., "Foundry Design Manual", Version 6.0, 1993.
- 13 - I. J. Chueiri (UFPR), "Uma Contribuição ao Projeto de CIs com MESFET em GaAs", Dissertação de Mestrado Orientada por Jacobus W. Swart, Fac. de Eng. Elétrica da UNICAMP, Maio de 1992.
- 14 - R. Zuleeg et al. (McDonnell Douglas), "Double-Implanted GaAs Complementary JFET's", IEEE Electron Device Letters, vol. EDL-5, n° 1, January 1984, pgs. 21-23.
- 15 - "Vitesse Announces New GaAs Process", Compound Semiconductor, vol. 1, n° 2, September/October 1995, pg. 11.
- 16 - L. C. Kretly (UNICAMP), "Dispositivos Semicondutores de Alta Velocidade: Contribuição ao Modelamento e à Implantação de Tecnologia de MESFETs de GaAs com

Geometria Micron e Submicron”, Tese de Doutorado Orientada por Atílio J. Giarola, Fac. de Eng. Elétrica da UNICAMP, Junho de 1992.

17 – M. Hirayama et al. (NTT), “A GaAs 16-kbit Static RAM Using Dislocation-Free Crystal”, IEEE Trans. on Electron Devices, vol. ED-33, n° 1, January 1986, pgs. 104-110.

18 – E. R. Fox et al. (Control Data Corp.), “Reduced Instruction Set Architecture for a GaAs Microprocessor System”, Computer, vol. 19, n° 10, October 1986, pgs. 71-80.

19 – M. Muraguchi & M. Aikawa (NTT), “Microwave Components for Cellular Portable Radiotelephone”, Solid-State Electronics, vol. 38, n° 9, 1995, pgs. 1551-1557.

20 – “Cutting the Cord”, Compound Semiconductor, vol. 1, n° 2, September/October 1995, pgs. 33-40.

21 - A. Núñez et al., “Some Experiences in Designing a LSI GaAs Processor Using Karl-III”, Proceedings of the 2<sup>nd</sup> Abakus Workshop, Innsbruck, August 25-28, 1988, pgs. 73-79.

22 – L. C. Kretly & D. C. de Souza (DEMIC/UNICAMP), “Design of a 7490-Like Decade-Counter Integrated Circuit, Using GaAs MESFET DCFL Family, for Frequencies Up to 1 GHz”, Anais do XII Congresso da Sociedade Brasileira de Microeletrônica, Caxambu (MG), 28 de julho a 1 de agosto de 1997.

23 - C. A. Mead (Caltech), “Schottky Barrier Gate Field Effect Transistor”, Proceedings of the IEEE, vol. 54, n° 2, February 1966, pgs. 307-308.

24 - R. Cates (Vitesse), “Gallium Arsenide Finds a New Niche”, IEEE Spectrum, April 1990, pgs. 25-28.

25 – R. L. Van Tuyl & C. A. Liechti (Hewlett-Packard), “High-Speed Integrated Logic with GaAs MESFET’s”, IEEE Journal of Solid-State Circuits, vol. SC-9, n° 5, October 1974, pgs. 269-276.

26 – M. Fukuta et al. (Fujitsu Labs. Ltd.), “Normally-Off GaAs MESFET for Femto-Joule Pico-Second Logic Circuits”, IEEE Trans. on Electron Devices, vol. ED-24, n° 9, September 1977, pg. 1209.

27 – M. Fukuta et al. (Fujitsu Labs. Ltd.), “Low-Power GaAs Digital Integrated Circuits With Normally-Off MESFET’s”, IEEE Trans. on Electron Devices, vol. ED-25, n° 11, November 1978, pg. 1340.

28 – R. E. Lundgren et al. (Hughes Res. Labs.), “Fast Enhancement-Mode GaAs MESFET Logic”, IEEE Trans. on Electron Devices, vol. ED-26, n° 11, November 1979, pg. 1827.

29 – E. Khon, “GaAs-MeSFET for Digital Application”, Solid-State Electronics, vol. 20, n° 1, January 1977, pgs. 29-33.

30 – K. Lehovec (Univ. of Southern California), “GaAs Enhancement Mode FET-Tunnel Diode Ultra-Fast Low Power Inverter and Memory Cell”, IEEE Journal of Solid-State Circuits, vol. SC-14, n° 5, October 1979, pgs. 797-800.

- 31 – R. L. Van Tuyl et al. (Hewlett-Packard), “GaAs MESFET Logic with 4-GHz Clock Rate”, IEEE Journal of Solid-State Circuits, vol. SC-12, n° 5, October 1977, pgs. 485-496.
- 32 – R. C. Eden & B. M. Welch (Rockwell Res. Center), “GaAs VLSI Technology: Status and Issues”, Journal of Electrochemical Society, vol. 126, n° 8, August 1979, pg. 340C.
- 33 – S. I. Long et al. (Rockwell Res. Center), “MSI High-Speed Low-Power GaAs Integrated Circuits Using Schottky Diode FET Logic”, IEEE Trans. on Microwave Theory and Techniques, vol. MTT-28, n° 5, May 1980, pgs. 466-472.
- 34 – M. Gloanec et al. (Thomson-CSF), “An E-Beam Fabricated GaAs D-Type Flip-Flop IC”, IEEE Trans. on Microwave Theory and Techniques, vol. MTT-28, n° 5, May 1980, pgs. 472-478.
- 35 – T. Mizutani et al. (NTT), “High-Speed Enhancement-Mode GaAs MESFET Logic”, IEEE Trans. on Microwave Theory and Techniques, vol. MTT-28, n° 5, May 1980, pgs. 479-483.
- 36 – C. A. Liechti (Hewlett-Packard), “Pathways and Pitfalls for GaAs ICs”, Journal of Electrochemical Society, vol. 127, n° 8, August 1980, pg. 389C.
- 37 – H. M. Darley et al., “Silicon MESFET Integrated Circuit Technology”, Journal of Electrochemical Society, vol. 126, n° 8, August 1979, pg. 340C.
- 38 – G. Nuzillat et al. (Thomson-CSF), “A Subnanosecond Integrated Switching Circuit with MESFET’s for LSI”, IEEE Journal of Solid-State Circuits, vol. SC-11, n° 3, June 1976, pgs. 385-394.
- 39 – R. L. Van Tuyl & E. B. Stoneham (Hewlett-Packard), “GaAs VLSI – A Technology Challenge”, Journal of Electrochemical Society, vol. 127, n° 8, August 1980, pg. 392C.
- 40 – T. Mimura et al. (Fujitsu), “GaAs MOSFET for Low-Power High-Speed Logic Applications”, IEEE Trans. on Electron Devices, vol. ED-26, n° 11, November 1979, pg. 1828.
- 41 – K. Suyama et al. (Fujitsu), “Design and Performance of GaAs Normally-Off MESFET Integrated Circuits”, IEEE Trans. on Electron Devices, vol. ED-27, n° 6, June 1980, pgs. 1092-1097.
- 42 – R. Yamamoto et al. (NEC), “Design and Fabrication of Depletion GaAs LSI High-Speed 32-Bit Adder”, IEEE Journal of Solid-State Circuits, vol. SC-18, n° 5, October 1983, pgs. 592-598.
- 43 – Y. Nakayama et al. (Fujitsu), “A GaAs 16x16 bit Parallel Multiplier”, IEEE Journal of Solid-State Circuits, vol. SC-18, n° 5, October 1983, pgs. 599-603.
- 44 – D. MacMillan & T. Gheewala (GigaBit Logic), “Learn Gallium-Arsenide Basics Before Applying High-Speed ICs”, EDN Magazine, March 22, 1984, pgs. 239-246.

- 45 - W. Twaddell (Western Editor), "GaAs No Longer Next Year's Technology as Digital Circuits Come to Market", EDN Magazine, May 17, 1984, pgs. 67-74.
- 46 - D. Bursky, "Digital GaAs ICs Emerge From the Lab With Improved Materials, Structures", Electronic Design, December 12, 1985, pgs. 79-88.
- 47 - T. Andrade & J. R. Anderson (Avantek), "High Frequency Divider Circuits Using Ion-Implanted GaAs MESFET's", IEEE Electron Device Letters, vol. EDL-6, n° 2, February 1985, pgs. 83-85.
- 48 - M. F. Chang et al. (Rockwell Intl. Corp.), "High-Speed GaAs Frequency Dividers Using a Self-Aligned Dual-Level Double Lift-Off Substitution Gate MESFET Process", IEEE Electron Device Letters, vol. EDL-6, n° 6, June 1985, pgs. 279-281.
- 49 - W. Andrews, "US Takes Step to Support Fragile GaAs Wafer Base", Electronic News, May 9, 1994, pg. 28.
- 50 - T. Otsuji et al. (NTT), "40 Gb/s ICs for Future Lightwave Communications Systems", IEEE Journal of Solid-State Circuits, vol. SC-32, n° 9, September 1997, pgs. 1363-1369.
- 51 - M. Ida et al., "A 4 Gbit/s GaAs 16:1 Multiplexer / 1:16 Demultiplexer LSI Chip", IEEE Journal of Solid-State Circuits, vol. SC-24, n° 8, August 1989, pgs. 928-931.
- 52 - K. Runge (Bellcore), "1:4 Demultiplexer Architecture for Gbit/s Lightwave Systems", Electronics Letters, vol. 27, n° 9, 25<sup>th</sup> April 1991, pgs. 753-755.
- 53 - R. B. Nubling et al., "High-Speed 8:1 Multiplexer and 1:8 Demultiplexer Implemented with AlGaAs/GaAs HBTs", IEEE Journal of Solid-State Circuits, vol. SC-26, n° 10, October 1991, pgs. 1354-1361.
- 54 - K. Tanaka et al. (Oki Electric Ind. Co.), "8-Gb/s 8:1 Multiplexer and 1:8 Demultiplexer IC's Using GaAs DCFL Circuit", IEEE Journal of Solid-State Circuits, Vol. SC-27, n° 10, October 1992, pgs. 1359-1362.
- 55 - M. Ohhata et al., "25 Gbit/s Selector Module Using 0.2 $\mu$ m GaAs MESFET Technology", Electronics Letters, vol. 29, n° 11, 27<sup>th</sup> May 1993, pgs. 950-951.
- 56 - N. Higashisaka et al. (Mitsubishi Electric Corp.), "GaAs DCFL 2.5 Gbps 16-bit Multiplexer/Demultiplexer LSI's", IEEE Journal of Solid-State Circuits, vol. SC-29, n° 7, July 1994, pgs. 808-814.
- 57 - M. Shikata et al. (Oki Electric Ind. Co.), "A 20-Gb/s Flip-Flop Circuit Using Direct-Coupled FET Logic", IEEE Journal of Solid-State Circuits, vol. SC-28, n° 10, October 1993, pgs. 1046-1051.
- 58 - T. Otsuji et al. (NTT), "A Super-Dynamic Flip-Flop Circuit for Broad-Band Applications up to 24 Gb/s Utilizing Production-Level 0.2- $\mu$ m GaAs MESFET's", IEEE Journal of Solid-State Circuits, vol. SC-32, n° 9, September 1997, pgs. 1357-1362.

- 59 – “Vitesse Expands GaAs IC Product Line”, Compound Semiconductor, Nov./Dec. 1995, vol. 1, n° 3, pg. 4.
- 60 – J. B. Hughes et al., “A Versatile ECL Multiplexer IC for the Gbit/s Range”, IEEE Journal of Solid-State Circuits, October 1979, pgs. 812-817.
- 61 - A. Felder et al., “46 Gb/s Demux, 50 Gb/s Mux, and 30 GHz Static Frequency Divider in Silicon Bipolar Technology”, IEEE Journal of Solid-State Circuits, vol. SC-31, n° 4, April 1996, pgs. 481-486.
- 62 - “M/A-COM Acquires Cray’s GaAs Facility to Meet Increasing Demand for GaAs ICs”, Compound Semiconductor, Nov./Dec. 1995, vol. 1, n° 3, pg. 13.
- 63 – A. Barna & C. A. Liechti, “Optimization of GaAs MESFET Logic Gates with Subnanosecond Propagation Delays”, IEEE Journal of Solid-State Circuits, vol. SC-14, n° 4, August 1979, pgs. 708-715.
- 64 – GigaBit Logic Inc., “1989 GaAs IC Databook and Designer's Guide” (disponível no CPqD da Telebrás).
- 65 – M. J. Helix et al., “Fan-Out and Speed of GaAs SDFL Logic”, IEEE Journal of Solid-State Circuits, vol. SC-17, n° 6, December 1982, pgs. 1226-1231.
- 66 – O. Wing, “Gallium Arsenide Digital Circuits”, Kluwer Academic Publishers, 1990.
- 67 – C. H. Hyun et al. (Univ. of Minnesota), “Analysis of Noise Margin and Speed of GaAs MESFET DCFL Using UM-SPICE”, IEEE Trans. on Electron Devices, vol. ED-33, n° 10, October 1986, pgs. 1421-1425.
- 68 – S. Katsu et al., “A Source Coupled FET Logic – A New Current-Mode Approach to GaAs Logics”, IEEE Trans. on Electron Devices, vol. ED-32, n° 6, June 1985, pgs. 1114-1118.
- 69 – R. Wilson, “Vitesse Pumps GaAs for Communications”, Electronic Engineering Times, September 5, 1994, pg. 4.
- 70 - T. Abrão, “Circuitos Integrados Digitais de Alta Velocidade em GaAs: Demultiplexador de 16 Canais em 2,5 Gb/s”, Dissertação de Mestrado Orientada por Fátima S. Correra, Escola Politécnica da USP, São Paulo, 1996.
- 71 - T. Takada et al. (NTT), “A 2 Gb/s Throughput GaAs Digital Time Switch LSI Using LSCFL”, IEEE Trans. on Electron Devices, vol. ED-32, n° 12, December 1985, pgs. 2748-2753.
- 72 - A. S. Sedra & K. C. Smith, “Microelectronic Circuits”, Saunders College Publishing, 3<sup>rd</sup> edition, 1990.
- 73 - J. Lohstroh (Philips Res. Labs.), “Static and Dynamic Noise Margins of Logic Circuits”, IEEE Journal of Solid-State Circuits, Vol. SC-14, n° 3, June 1979, pgs. 591-598.



- 74 – Meta-Software Inc., “HSPICE User’s Manual Volume 1: Getting Started”, Version H92, 1995.
- 75 – Meta-Software Inc., “HSPICE User’s Manual Volume 3: Analysis and Methods”, Version H92, 1995.
- 76 – H. Statz et al. (Raytheon), “GaAs FET Device and Circuit Simulation in SPICE”, IEEE Trans. on Electron Devices”, vol. ED-34, n° 2, February 1987, pgs. 160-169.
- 77 – P. Remy et al., “A GaAs Electrical Rule Checker”, Proceedings of the X SBCCI (Brazilian Symposium on Integrated Circuit Design), Gramado (RS), August 25-27, 1997, pgs. 1-10.
- 78 – I. Moussa, “Applications des Circuits Numériques en Arseniure de Gallium dans les Systèmes à Haut Débit de Communication et dans les Calculateurs Performants”, Thèse de Docteur, TIMA/INPG, 1996.
- 79 – J. Huber & M. Rosneck, “Successful ASIC Design the First Time Through”, Van Nostrand Reinhold, 1991.
- 80 – Texas Instruments Inc., “The TTL Databook”, 1980.
- 81 – Texas Instruments Inc., “Logic Selection Guide and Databook” CD-ROM, January 1997.
- 82 – “Divisor de Frequência por um Número ‘N’ Compreendido Entre 2 e 10”, 103 Dicas e Macetes em Eletrônica, Editora Seleções Eletrônicas Ltda., Rio de Janeiro, 1990, pg. 15.
- 83 – Texas Instruments Inc., “Designing with TTL Integrated Circuits”, McGraw-Hill, 1971.
- 84 – G. A. S. Machado, “Low-Power HF Microelectronics: a Unified Approach”, The Institution of Electrical Engineers, 1996.
- 85 – A. Higashisaka et al. (NEC), “A High-Power GaAs MESFET with an Experimentally Optimized Pattern”, IEEE Trans. on Electron Devices, vol. ED-27, n° 6, June 1980, pgs. 1025-1029.
- 86 – K. Eshraghian et al., “Speed-Area-Power Optimization for DCFL and SDCFL Class of Logic Using Ring Notation”, Microprocessing and Microprogramming, vol. 32, 1991, pgs. 75-82.
- 87 – J. Ousterhout et al., “Magic Version 6 Tutorials”, Univ. of California at Berkeley, September 19, 1990.
- 88 – J. Ousterhout et al., “Magic Version 6 Maintainer’s Manuals”, Univ. of California at Berkeley, September 19, 1990.
- 89 – R. Kozlowski (National Semiconductor), “Follow PC-Board Design Guidelines for Lowest CMOS EMI Radiation”, EDN Magazine, May 17, 1984, pgs. 149-154.

- 90 – J. Novellino, “High-Speed Circuits Require Special Test Techniques”, *Electronic Design*, July 24, 1995, pgs. 71-80.
- 91 – A. K. Stevens, “Introduction to Component Testing – Application Electronics”, Addison-Wesley Publishing Co., 1986.
- 92 – W. R. Blood Jr. , “MECL System Design Handbook”, Motorola Semiconductor Products Inc., 2<sup>nd</sup> Edition, 1972.
- 93 – T. Gheewala & D. MacMillan (GigaBit Logic), “High-Speed GaAs Logic Systems Require Special Packaging”, *EDN Magazine*, May 17, 1984, pgs. 135-142.
- 94 – R. G. Pereira, “Transistores de Efeito de Campo Metal-Semicondutor (MESFET)”, *Anais da V Oficina de Microeletrônica*, Campinas, SP, Julho de 1996.
- 95 – R. P. Ribas & A. Guyot, “Can Asynchronous Design Reduce Power Dissipation in GaAs ICs?”, <http://tima-cmp.imag.fr/Homepages/ribas/publication.html>.
- 96 – M. Strum, “Considerações de Projeto: Os Caminhos para o Leiaute”, Curso de Metodologia de Projeto de ASICs ministrado no XII Congresso da Sociedade Brasileira de Microeletrônica, Caxambu (MG), 28 de julho a 1 de agosto de 1997.
- 97 – S. F. C. Neto, “Projeto de Máscaras de Estruturas de Testes para Circuitos Integrados em GaAs”, Relatório de Iniciação Científica Orientado por Luiz C. Kretly, FEE/UNICAMP, 1<sup>o</sup> Semestre de 1986.

## Fontes de Pesquisa na Internet.

- 1 – **CMP:** <http://tima-cmp.imag.fr/CMP/CMP.html>

A *homepage* do CMP contém todas as informações para a realização de qualquer projeto de CI para uma das suas *foundries* associadas, e a sua posterior fabricação. Essa *homepage* fornece a programação das rodadas de cada *foundry* para o ano corrente, com os prazos finais de envio de projetos, a relação de *design kits* disponíveis para cada *foundry* e plataforma de CAD, que podem ser adquiridos por intermédio do CMP, e a lista dos preços de fabricação dos protótipos, em francos franceses por mm<sup>2</sup> (FF/mm<sup>2</sup>).

- 2 – **Vitesse Semiconductor Corporation:** <http://www.vitesse.com>
- 3 – **GaAsNET Industry Resource on GaAs Electronics:** <http://www.gaasnet.com>
- 4 – **Magic - A VLSI Layout System:** <http://www.research.digital.com/wrl/projects/magic>
- 5 – **TriQuint Semiconductor, Inc.:** <http://www.triquint.com>
- 6 – **Intusoft Inc.:** <http://www.intusoft.com>
- 7 – **ISI Citation Databases:** <http://webofscience.com>