UNIVERSIDADE ESTADUAL DE CAMPINAS - UNICAMP

FACULDADE DE ENGENHARIA ELÉTRICA - FEE

"DESENVOLVIMENTO DE UM CIRCUITO INTEGRADO PARA TESTABILIDADE DE PLACAS"

Autor- ARTHUR HENRIQUE CÉSAR DE OLIVEIRA
R.A.- 855066

Orientador - Prof.Dr. Carlos I. Z. Mammana

Tese apresentada à Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas - UNICAMP, como parte dos requisitos exigidos para obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA.

Março de 1990

Este exemplar corresponde à redação final da tese defendida por Arthur Henrique César de Oliveira e aprovada pela comissão julgadora em 19 de março de 1990

C.I.Z. Mammana

30/7/90

ÍNDICE DE ASSUNTOS:

AGRADECIMENTOS:

Agradeço ao grupo de projeto de circuitos integrados e suporte computacional do Instituto de Microeletrônica do Centro Tecnológico para Informática pelo apoio na implementação do CTIP - Circuito para Teste Integrado de Placas.

RESUMO:

Este trabalho de Mestrado em Engenharia Elétrica, trata do desenvolvimento de um circuito integrado modular para ser aplicado no projeto para testabilidade de placas eletrônicas digitais. É um CI programável que visa facilitar a implementação de Scan-Test e Self-Test nas placas.

Os capítulos 1, 2 e 3 servem de subsídio para o trabalho, conceituando o problema-teste de circuitos lógicos, geração de vetores de teste e projeto para testabilidade. No capítulo 4 é apresentado o projeto do Circuito para Teste Integrado de Placas (CTIP), partindo da especificação, simulação, lay-out, até os testes de validação. No capítulo 5 apresentam-se as conclusões e um exemplo de aplicação do CTIP.

ABSTRACT:

The subject of this Master in Electrical Engineering Thesis is the design of a modular integrated circuit to be used in board design for testability. This IC is programmable and aims to easy implementing PCBoard Scan and Self-Test.

Chapters 1, 2 and 3 are subsides for the others, defining the logic circuits testing-problem, test vector generation and design for testability. Chapter 4 presents the design of the Board Testing IC (CTIP), from specification, through simulation, lay-out and testing. Chapter 5 presents conclusions and an application example.

Observações- ao longo do texto:

1- o símbolo E representa o expoente do número anterior (por exemplo, 2 E8 significa 2 elevado à oitava potência);

2- o símbolo b representa o complemento lógico do sinal (por exemplo, Db, CKb).

DESENVOLVIMENTO DE UM CIRCUITO INTEGRADO

PARA TESTABILIDADE DE PLACAS

1. INTRODUÇÃO:

Para se verificar o correto funcionamento de um circuito lógico, seja um circuito integrado, uma placa eletrônica ou um sistema eletrônico, é necessária a realização de diversos testes e ensaios [Mil883]. Os Testes Elétricos podem ser divididos basicamente em:

a) testes funcionais (qualitativos), com os quais se verifica o funcionamento lógico do circuito, e que podem ser estáticos (frequência de execução bem menor que a de operação), ou dinâmicos (para se avaliar a correta temporização das operações);

b) testes paramétricos (quantitativos), os quais medem grandezas elétricas do circuito (parâmetros), como tensões, correntes, resistências (parâmetros DC), ou ainda tempos e frequências (parâmetros AC);

c) teste curto-aberto, para verificar a integridade das ligações e pinos de conexão (principalmente em memórias e placas nuas).

1.1. Problema-Teste:

A complexidade dos circuitos lógicos utilizados em sistemas digitais (como por exemplo computadores, sistemas de comunicação e de automação), tem crescido de maneira vertiginosa, atingindo a centenas de milhares de portas lógicas e flip-flops, principalmente pelo uso da integração de dispositivos em escala muito ampla (VLSI). Essa crescente complexidade dos circuitos (CI's, placas ou sistemas), implica numa grande dificuldade para se executar os testes elétricos exigindo um tempo proibitivo para se definir o conjunto de testes e aplicá-lo nos circuitos [Seg82]. Isto porque para um circuito lógico com "n" entradas e "m" elementos de armazenamento (flip-flops, latches ou memórias), existem 2 E(n+m) combinações de estados lógicos possíveis (fig.1). Por exemplo, para se testar um processador completamente (um chip LSI, com n=25 e m=50), fazendo-o executar todas as operações com todo tipo de operando, levaria cerca de 10 E9 anos utilizando os equipamentos de teste disponíveis [Wil183].

Portanto, torna-se impossível o teste pela operação exaustiva (modo normal de operação) de circuitos complexos, bem como tomaria muito tempo determinar o conjunto mínimo de operações que exercite todo o circuito.

Além do problema de geração e execução dos testes, o grande número de dados de teste (estímulos e respostas), bem como a complexa temporização dos circuitos aliada à crescente velocidade de operação, implicam na necessidade de Equipamentos Automáticos de Teste (ATE) complexos e caros, com grande poder de processamento, grande quantidade de memória para estímulos, e circuitos analógicos de alta precisão na interface com o circuito sob teste [Swan83] (ver fig.2). Por exemplo, os ATE's mais avançados possuem multiprocessadores, arquitetura "por-pino" [Swan83], 512 canais com 1Mbit de memória de alta velocidade, geração de estímulos até 100MHz, circuitos de compensação de atrasos (deskew), etc., com preços que ultrapassam US$2000000.00 [Els85,Hump84].

Para o caso de placas eletrônicas, os equipamentos de teste funcional acessam o circuito pelos conectores de borda, pois o acesso por "cama-de-pregos" é caro e complicado, exigindo maior número de canais. Com isso seu teste se assemelha ao teste de CI's, onde só se tem acesso aos pinos. O uso de testadores tipo "in-circuit" [Morr86], que isolam e testam os componentes na placa e são mais simples e baratos que os testadores funcionais, é prejudicado pelo emprego de componentes VLSI, que exigem testes funcionais complexos, e pelos dispositivos de montagem na superfície (SMD), que dificultam o acesso por "camas-de-pregos" [Turi85].

Além dos testes de produção, executados nas fábricas por esses ATE's complexos, ainda existe o problema do teste dos sistemas em campo. Esses testes são necessários no diagnóstico dos defeitos ocorridos durante o uso, para se proceder a manutenção.

1.2. Modo-Teste e Testabilidade:

Em função do "problema-teste" exposto, é necessário que durante o projeto dos circuitos sejam previstas facilidades para teste (Design for Testability - DFT [Will83,Abra83]), sendo que para facilitar o teste em campo podem-se empregar técnicas de auto-teste [Maun85], que reduzem o tempo de teste e dispensam ATE's complexos (ver cap.3).

Com o uso de técnicas de Projeto para Testabilidade, o circuito pode ser reconfigurado durante o teste pela atuação de sinais de controle, passando para o "Modo-Teste" (fig.3). Assim, circuitos complexos podem ser divididos, separando lógica sequencial da lógica combinacional (esta, mais fácil de testar) e permitindo o acesso a pontos internos ao circuito. Com isso podem ser gerados "testes estruturados", que são conjuntos mínimos de estímulos que testam cada bloco do circuito (ver cap.2).

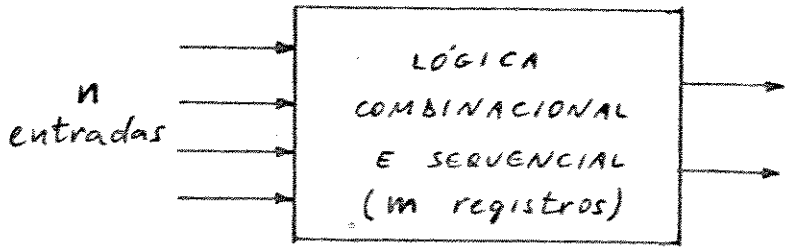


Fig.1 - Circuito lógico com blocos combinacionais e sequenciais (2 En+m combinações de estados possíveis)

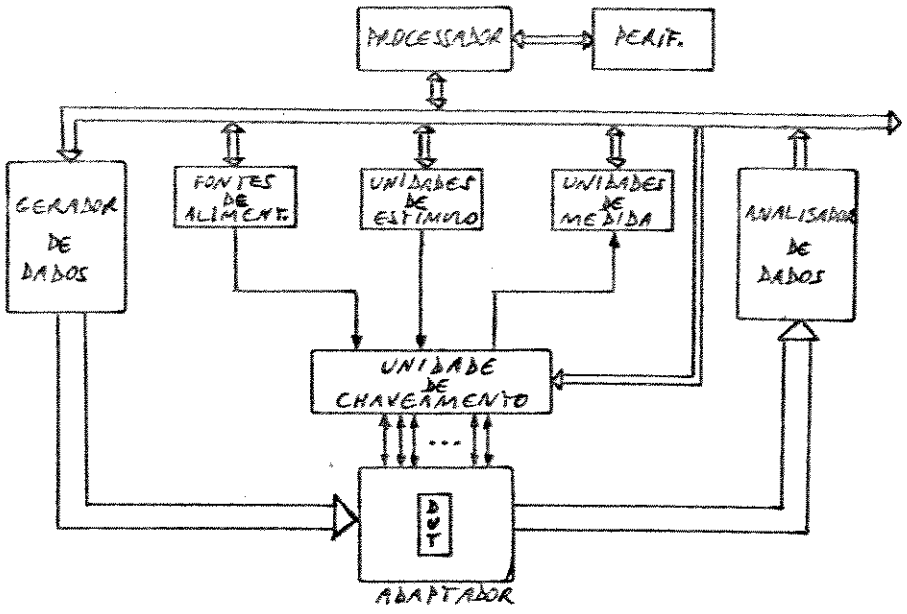


Fig.2 - Arquitetura típica de uma Estação de Teste (ATE)

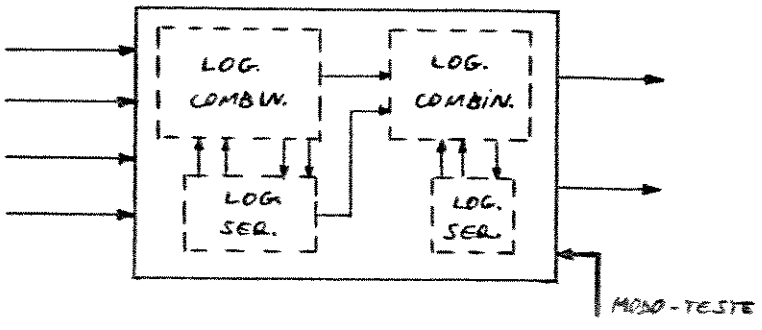


Fig.3 - Circuito projetado para testabilidade: é reconfigurado através de um sinal de controle durante o teste (Modo-Teste)

Neste trabalho serão discutidas algumas técnicas básicas de Projeto para Testabilidade, inclusive metodologias combinando Scan e Self-Test. Será então apresentado um chip (CTIP) que foi projetado para ser aplicado no projeto para testabilidade de placas digitais já prontas, baseado na metodologia combinada (cap. 3.4).

Assume-se neste trabalho, para fim de descrição dos tópicos, um circuito lógico como sendo formado por diversos blocos, com acesso apenas pelos pinos, seja um circuito integrado, uma placa ou um sistema.

2. GERAÇÃO DOS TESTES

Para a execução dos testes funcionais em circuitos lógicos, é necessário que se definam os estímulos de teste (vetores de teste) e as respostas esperadas, de modo que se possa verificar se um circuito sob teste responde corretamente. O número de estímulos de teste a serem aplicados, será limitado pelo tempo de teste e pela memória do ATE, de modo que os testes do circuito sejam realizados num tempo e a um custo aceitáveis [Goel80]. Além disso, esse número de estímulos de teste deve ser suficiente para exercitar a quase totalidade do circuito, sendo definida então a "cobertura de falhas", ou seja, qual a porcentagem que será detetada do total de falhas possíveis de ocorrer no circuito.

A seguir serão discutidos modelos de falhas e métodos de geração de testes para circuitos combinacionais e sequenciais.

Algumas classes especiais de circuitos, como memórias (RAM e ROM) e arranjos lógicos (PLA : Programmable Logic Array), serão tratadas separadamente, nos ítens 3.6 e 3.7, por exigirem testes e modelos de falhas específicos. Esses circuitos não permitem a aplicação direta dos métodos de geração de estímulos de teste descritos.

2.1. Modelos de Falhas:

Para se avaliar a cobertura de falhas, devem-se realizar simulações (lógicas ou elétricas) do circuito, utilizando os estímulos de teste definidos e adotando um modelo de falha que represente as falhas físicas que podem ocorrer no circuito eletrônico [Gali80,Breu72]. Os programas simuladores de falhas [Indr86] realizam sucessivas simulações lógicas, colocando uma falha em cada nó do circuito, e verificando se o resultado difere do obtido com o circuito sem falhas. Portanto o Simulador de Falhas indicará a cobertura de falhas de um conjunto de vetores de teste, podendo-se optar então por aceitar esse valor ou gerar maior número de vetores para atingir a cobertura de falhas desejada.

Em um circuito podem ocorrer diversas falhas físicas, causadas tanto pelo processo de fabricação como pelo uso. Essas falhas podem ser trilhas em curto ou interrompidas, dispositivos defeituosos (como componentes queimados ou com limiares alterados por contaminação), componentes trocados ou montados invertidos nas placas, etc. Esses defeitos causam erros de funcionamento, podendo-se adotar um modelo de falhas que represente as falhas físicas, causando os mesmos erros. Um dos modelos mais utilizados é o de "linhas-presas-em-0" e "linhas-presas-em-1" (stuck-at: s-a-0, s-a-1 [Chan85]), aplicado ao nível de descrição por portas lógicas (gate level, fig 4.a).

Além da adoção de um modelo de falhas, considera-se a probabilidade de ocorrer uma única falha no circuito (falha simples), ou a ocorrência de falhas múltiplas, caso em que pode ocorrer inclusive o mascaramento de uma falha por outra, exigindo maior trabalho na geração dos vetores de teste e algorítmos especiais [Breu76].

Existem vários simuladores de falhas e programas geradores de vetores de teste (ver 2.3) que utilizam o modelo "stuck-at" de falhas simples; entretanto, para circuitos de alta complexidade (como CI's VLSI) e dispositivos MOS, este modelo não é adequado [Ban84]. Isto porque, para circuitos complexos existe grande probabilidade de ocorrerem falhas múltiplas, e para dispositivos MOS a ocorrência de linhas interrompidas ou falta de contatos (fig 4.b) podem causar armazenamento de estados, implicando em comportamento sequencial [Ban84]. Assim, devem ser adotados modelos de falhas mais complexos, como "stuck-on" e "stuck-open" [Chan85], o que dificulta a geração dos vetores de teste e as simulações.

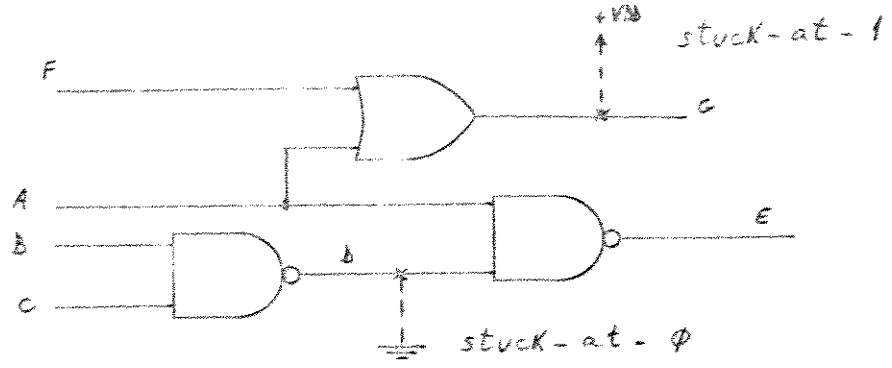


Fig.4a - Modelo "stuck-at" de falha (gate level)

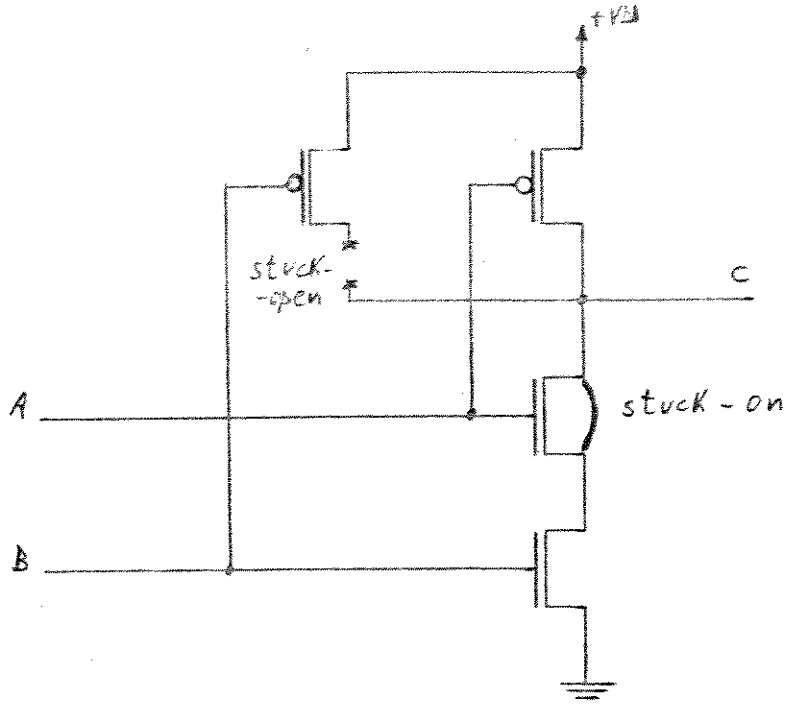


Fig.4b - Modelos "stuck-on" e "stuck-open" de falhas para MOS

2.2. Teste Exaustivo:

O teste exaustivo de um circuito lógico significa forçá-lo a passar por todos os estados lógicos possíveis, de modo que se verifique que em nenhum deles ocorrem erros de funcionamento. Como foi ressaltado na Introdução, o teste exaustivo de circuitos complexos é inviável pelo tempo e memória de dados necessários para sua execução, os quais aumentam exponencialmente com o tamanho do circuito [Goel80].

O teste funcional de circuitos combinacionais é relativamente simples, pois o estado nas saídas depende somente da combinação dos estados nas entradas, independentemente da ordem em que são aplicados. Já para o caso de circuitos sequenciais, o estado atual depende da sequência dos estados anteriores, o que torna seu teste bastante difícil [Breu76]. O problema é ainda maior pelo fato de se ter lógica mista (combinacional e sequencial) e poucos pontos de acesso, para circuitos complexos.

Assim, o teste funcional exaustivo só é viável para circuitos pequenos (por exemplo, com menos de 100 elementos lógicos), ou para partes do circuito. Portanto, devem ser adotadas alternativas para o teste do circuito por partes, utilizando técnicas de testabilidade (ver cap.3).

As vantagens do Teste Exaustivo são:
- pode ser facilmente definido pelo projetista, para circuitos simples, a partir das formas de onda de operação, não sendo necessário utilizar programas de geração de testes;
- pode ser realizado dinâmicamente, na velocidade de operação do circuito, para se avaliar a temporização.

2.3. Teste Estruturado:

Através da aplicação de técnicas de Projeto para Testabilidade (DFT) durante o projeto do circuito, pode-se dividir blocos complexos em blocos combinacionais e blocos sequenciais, ou ainda aumentar o número de pontos de acesso a nós internos, para se controlar ou observar seu estado (Modo-Teste). Assim, o problema de geração e aplicação dos testes será restrito a blocos simples separados, bastando o agrupamento final dos conjuntos mínimos de vetores de teste. Esses conjuntos mínimos de vetores de teste para blocos combinacionais, podem ser gerados por programas de computador, utilizando algorítmos já estabelecidos.

Assim, Teste Estruturado significa o teste de um circuito com o conjunto mínimo de vetores gerados especificamente para cada bloco (geralmente aplicados com o auxílio de técnicas de testabilidade, cap. 3).

Cabe ressaltar que como o circuito é reconfigurado durante o modo-teste (sendo introduzidas estruturas de teste e alterando a cadeia de elementos lógicos original), o teste estruturado só verifica a lógica (teste estático) e não a temporização, sendo por isso necessário acrescentar alguns estímulos para teste dinâmico no modo normal de operação do circuito [Oliv86].

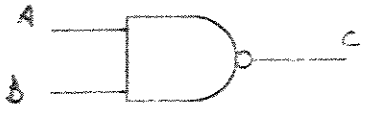
Para ilustrar a minimização do conjunto de estímulos de teste (teste estruturado), observe-se o caso de uma porta NAND de 2 entradas, como na fig.5 a seguir. Para seu teste exaustivo existem 4 padrões ou vetores de teste, como indicado na tabela verdade. Assumindo-se o modelo "stuck-at" de falhas simples, bastariam os vetores número 2, 3 e 4 para garantir seu correto funcionamento, pois o vetor 2 testa A-s-a-1 e X-s-a-0, o vetor 3 testa B-s-a-1 e X-s-a-0, e o vetor 4 testa simultaneamente A-s-a-0, B-s-a-0 e X-s-a-1, como ilustrado na fig.6. Pode-se observar que no caso de ocorrer uma das 6 falhas possíveis, o estado na saída será diferente do esperado (tabela verdade).

Para uma lógica tipo AND-OR-INVERT, ao invés dos 16 estímulos exaustivos, bastam 4 vetores (assinalados com * na tabela da fig.7) para o teste completo de circuito, como pode ser demontrado pelo método dos "caminhos críticos" (ver fig. 8).

Como foi mostrado, o número mínimo de vetores que testam completamente um circuito é menor que o número de estímulos exaustivos [Seg82]. Além disso, com a divisão do circuito em blocos, o número total de vetores será geralmente menor que a soma dos conjuntos de vetores parciais, porque vários deles podem ser aplicados em paralelo, como será visto depois na descrição do Scan-Test (cap.3.2).

As vantagens do Teste Estruturado são:
- economia de tempo de execução e menor memória de dados, pelo menor número de vetores;
- vetores podem ser gerados por algorítmos em computador.

As desvantagens do Teste Estruturado são:
- grande tempo de processamento para a geração dos vetores;
- o teste é estático;
- as estruturas de testabilidade alteram a temporização do circuito no modo normal de operação.



	A	B	C
	0	0	1
*	0	1	1
*	1	0	1
*	1	1	0

Fig.5 - Porta NAND com sua tabela verdade e vetores de teste (*)

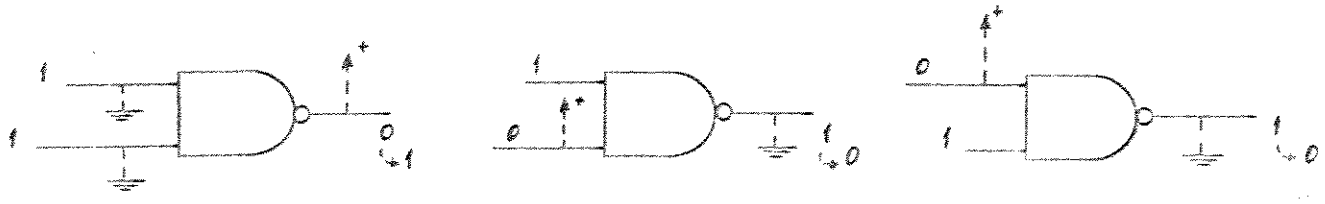
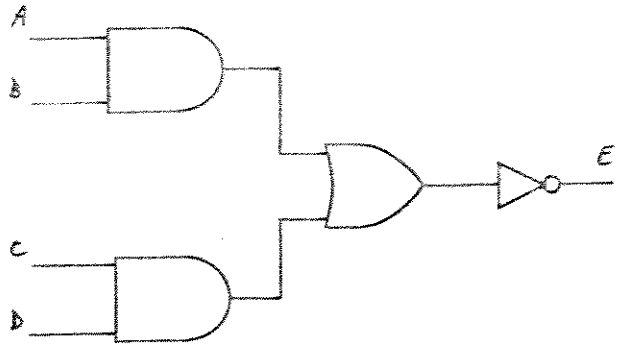


Fig.6 - Falhas "stuck-at-0" e "stuck-at-1" na NAND, com seus vetores de teste aplicados



	A	B	C	D	E
	0	0	0	0	1
	0	0	0	1	1
	0	0	1	0	1
*	0	0	1	1	0
	0	1	0	0	1
*	0	1	0	1	1
	0	1	1	0	1
	0	1	1	1	0
	1	0	0	0	1
	1	0	0	1	1
*	1	0	1	0	1
	1	0	1	1	0
*	1	1	0	0	0
	1	1	0	1	0
	1	1	1	0	0
	1	1	1	1	0

Fig.7 - Lógica AND-OR-INVERT com sua tabela verdade e vetores de teste, assinalados com *

A título de exemplo, para um circuito com mais de 1000 elementos lógicos, contendo 80 flip-flops, cerca de 4000 estímulos no modo-teste testaram 93% da lógica, enquanto no modo normal de operação seriam necessários alguns milhões de pulsos de clock (e sinais de entrada) para testar exaustivamente o circuito [Ori85].

2.3.1. Algorítmos de Geração para Circuitos Combinacionais:

Serão citados a seguir alguns métodos e algorítmos de computador, os quais geram o conjunto mínimo de vetores de teste, sendo eficazes para circuitos combinacionais complexos.

- Diferenças Booleanas, o qual trata com as expressões lógicas do circuito, calculando a função diferença (XOR) [Breu72, Mari71];

- Algorítmo D, faz tratamento matemático a partir dos cubos (notação ordenada das variáveis de entrada e saída) do circuito; são definidos os símbolos "D" e "Db", que representam os estados 1 e 0 das linhas que estão sendo testadas [Roth66, Breu72];

- Caminhos Críticos, utiliza os símbolos D e Db, realizando propagação reversa dos estados das saídas para as entradas; é simples de ser executado manualmente (ver fig.8) [Abra84];

- Algorítmo 9V, utiliza os estados 0, 1, X, D, Db, 0/D, 0/Db, 1/D e 1/Db, os quais permitem marcar opções em linhas ainda não definidas; trabalha com fronteiras de propagação, que avançam à medida em que novos estados são definidos a cada passo [Cha79].

- PODEM [Goel81]

- FAN [Fuji83]

2.3.2 Algorítmos de Geração para Circuitos Sequenciais:

Os métodos de geração de vetores de teste para circuitos sequenciais, não são genéricos nem eficientes, sendo necessário um grande esforço do projetista para determinar a sequência mínima de vetores de teste. Existem dois problemas que têm de ser tratados: a inicialização do circuito para um estado conhecido; e a quebra das linhas de realimentação.

O que se faz para solucionar estes problemas, é considerar o tempo como parâmetro [Breu76], modelando o circuito sequencial como se fosse combinacional entre dois pulsos de clock, e considerando o estado anterior Q' como uma entrada (ver fig.9). Assim podem-se empregar métodos similares aos combinacionais para gerar os vetores. Alguns métodos e algorítmos para geração de vetores para circuitos sequenciais são:

- Extended Backtracing, é uma extensão do princípio de sensibilização de caminhos [Breu76]; trabalha com estados lógicos em dois intervalos de tempo (PTF: previous time frame e CTF: current time frame) [Mari78];

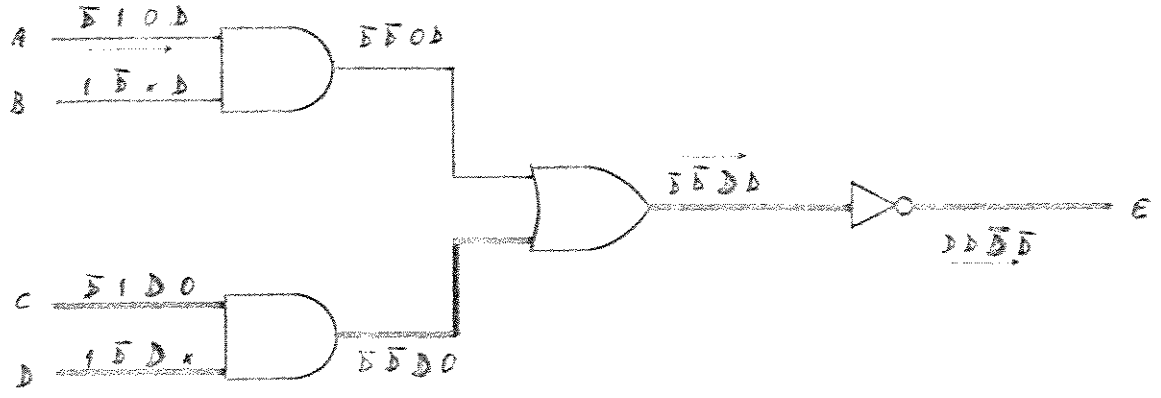
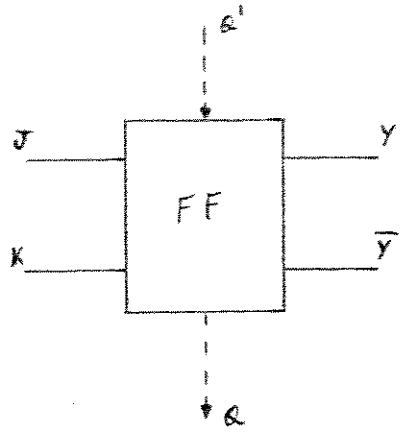


Fig.8 - Exemplo de aplicação do algorítmo "Critical Path Tracing"
a) assume-se o teste de um tipo de falha numa saída;
b) propagam-se os estados até as entradas, formando um caminho crítico;
c) propagam-se os estados das entradas para outras saídas eventualmente existentes, procurando testar falhas em outros caminhos;
d) cada combinação de estados nas entradas é um vetor de teste.



Q'	J	K	Q	Y	$\overline{Y}$
0	0	0	0	0	1
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	1	1	0
1	0	1	0	0	1
1	1	0	1	1	0
1	1	1	0	0	1

Fig.9 - Exemplo de modelamento de um circuito sequencial

- STG, baseado no algorítmo D, trabalha também com dois intervalos de tempo e propagação indexada [Mal185];

- PODEM-X [Goe81a];

- Seshu-Freeman, [Breu72];

- Auch-Cheng, [Breu72];

2.3.3. Programas de Geração Automática de Vetores de Teste (ATPG)

Existem alguns programas comerciais para geração de vetores de teste em computador, geralmente associados a simuladores lógicos, como por exemplo:

- TEGAS, trabalha com sensibilização dos caminhos (semelhante ao algorítmo D, [Breu72]) e geração heurística (pseudo-aleatória, ver 2.5); adota o modelo "stuck-at" de falhas e permite a definição de falhas lógicas representadas por uma função;

- HILO, trabalha com o método dos caminhos críticos e o modelo "stuck-at" de falhas; utiliza elementos de atraso nas realimentações, para tratar com circuitos sequenciais.

Os dois programas são eficientes para circuitos combinacionais, mas exigem a interferência do projetista para circuitos complexos, ou sua divisão em blocos.

2.4. Teste Pseudo-Exaustivo

Utilizando as técnicas de Projeto para Testabilidade, como exposto em 2.3, é possível particionar o circuito em blocos de um tamanho tal que seja viável o seu teste exaustivo. Para o caso de um bloco combinacional em que nenhuma das saídas dependa de todas as "n" entradas (circuitos parcialmente dependentes, "PD" [McC184]) é possível determinar um conjunto de estímulos que o exercite totalmente, sem ser necessário aplicar as 2 En combinações de estados possíveis das entradas. Esta técnica é chamada de Teste por Verificação (Verification Testing) e se baseia no tratamento das matrizes que descrevem o circuito [McC184]. É chamado de Teste Pseudo-Exaustivo porque, embora teste completamente cada bloco, o número total de estímulos é menor que 2 E(n+m). Suas principais vantagens são:
- o tempo de processamento para gerar os estímulos é relativamente pequeno;
- não é necessário realizar simulações de falhas.

Existem conjuntos de estímulos chamados de "Constant Weight Vectors" que são genéricos, e podem ser gerados facilmente por contadores [Tang83], sendo importantes para implementação de auto-teste, pois os estímulos podem ser gerados no próprio circuito.

2.5. Teste Aleatório

Existem vários trabalhos publicados [Sav84,Shed77], que demonstram que uma série aleatória de combinações das entradas, testa uma grande parte do circuito, podendo atingir 90% de cobertura de falhas. Essa série aleatória de estímulos pode ser gerada por blocos lógicos que façam parte do circuito sob teste [McCl81], de modo que um grande número de estímulos é gerado apenas aplicando-se pulsos de clock, não sendo necessário armazená-los no equipamento de teste. Assim, uma grande cobertura de falhas é atingida facilmente, sendo viável gerar milhões de estímulos de teste num tempo relativamente curto, na velocidade de operação do circuito, e sem a necessidade de um grande trabalho para particioná-lo [Sav84].

Os estímulos aplicados podem ser totalmente aleatórios (Random [Sav84]), ou seja, numa sequência não definida e podendo ocorrer repetição, ou podem ser pseudo-aleatórios (Pseudo-Random [Wag87]), quando nunca ocorre repetição de um vetor e a sequência embora pareça aleatória é determinística. Uma análise da eficiência dos testes aleatórios e pseudo-aleatórios (como "detetabilidade", "cobertura esperada" e "confiança" dos testes) é mostrada na bibliografia [Wag87].

O problema decorrente do uso de estímulos aleatórios é calcular a cobertura de falhas, o que só pode ser realizado por simuladores de falhas para blocos de um circuito particionado, pois consumiria um tempo muito grande para circuitos complexos pelo grande número de estímulos. A alternativa é realizar uma análise probabilística com auxílio de um algorítmo, como demonstrado em [Sav84].

Os testes pseudo-aleatórios são importantes por serem facilmente gerados por blocos lógicos, como por exemplo um contador ou um shift-register com realimentação linear (ver 3.3), e por poderem ser aplicados a lógica combinacional e sequencial. Estímulos pseudo-aleatórios são amplamente empregados em sistemas com Self-Test.

3. TESTABILIDADE

Como já foi exposto, é importante durante a fase de projeto de circuitos lógicos, que se verifique a necessidade de adotar técnicas para a execução dos testes funcionais. Assim, detetar pontos críticos para teste e incluir estruturas para facilitá-lo, durante a concepção de um circuito, é chamado de Projeto para Testabilidade, ou Projeto Visando a Testabilidade (Design For Testability - DFT [Will83]).

Testabilidade é a facilidade para se gerar e aplicar testes em um circuito, estando relacionada com a facilidade para se controlar ou observar os estados lógicos dos nós. Existem métodos para se avaliar o Grau de Testabilidade de um circuito, pela medida dos graus de Controlabilidade e Observabilidade (C/O) dos seus nós, como por exemplo os programas CAMELOT e SCOAP [Agra82].

Depois de analisada a testabilidade do circuito e detetados pontos críticos para o teste, devem ser adotadas técnicas para se melhorar seu grau de controle/observação, ou seja, facilitar a geração e aplicação dos testes funcionais. Entretanto, cabe ressaltar que o emprego dessas técnicas implica em maior tempo de projeto e aumento do circuito final com o consequente aumento da área ocupada.

3.1. Técnicas Básicas de Projeto para Testabilidade

Existem técnicas para orientar a concepção dos circuitos e técnicas para modificá-lo durante o teste. Várias técnicas podem ser combinadas durante o projeto de um circuito, sendo função do projetista determinar as soluções com base nos custos/benefícios.

As técnicas para orientar a concepção são geralmente soluções particulares (Ad-hoc techniques [Will83]), e visam evitar pontos críticos. As principais técnicas para orientar a concepção são [Gras81,Will83]:

- evitar circuitos assíncronos, onde os eventos ocorrem independentes de um clock principal;

- partição do circuito em blocos funcionais, que podem ser isolados para o teste;

- acréscimo de pontos (pinos) de teste, para controle/observação de nós internos;

- evitar elementos de atraso, que não são controláveis, e podem variar causando "spikes" ou falhas;

- colocar pinos de inicialização de blocos sequenciais (como Clear ou Preset), ligados a +V ou GND por resistores;

- uso de estruturas tipo barramento, para a isolação de blocos com saídas tipo 'tristate'.

As técnicas que modificam o circuito durante o teste, empregam estruturas já estabelecidas, que visam separar os blocos combinacionais da lógica sequencial no Modo-Teste, aumentando o controle/observação dos nós internos ('structured techniques' [Will83]). As principais técnicas estruturadas de projeto para testabilidade são descritas a seguir.

3.2. Scan-Test

O princípio de operação do Scan-Test é o acesso seriado para controlar ou observar os estados dos nós internos ao circuito. A sequência básica de operação (ver fig. 10) é:

1)acionando-se um sinal de controle (pino de teste) o circuito é reconfigurado, passando para o 'modo-teste';
2)os flip-flops originais do circuito e outros agregados para teste, são interconectados, formando um longo (ou vários) scan--shift-register (SSR);
3)os estímulos de teste (vetores estruturados ou exaustivos) são deslocados pelo SSR (scan-in), para atingir pontos internos;
4)desliga-se o sinal de controle e o circuito volta para o modo normal de operação, sendo que as saídas dos flip-flops contendo os dados introduzidos, excitam as entradas dos blocos combinacionais;
5)é aplicado um pulso de clock para que os flip-flops armazenem os resultados das saídas dos blocos combinacionais;
6)coloca-se o circuito novamente no modo-teste, recompondo o SSR;
7)os resultados armazenados, são deslocados ao longo do SSR para um pino de saída (scan-out), para posterior análise.

Existem variações na maneira de se implementar o Scan-Test, embora o princípio básico de operação seja o mesmo. As principais técnicas são [Mc84a]:

- Scan-Path, que utiliza um flip-flop com a entrada D multiplexada (Scan-flip-flop, fig. 11), que é uma adaptação dos já existentes no circuito, para formar o SSR. O scan-path pode ser implementado por flip-flops do circuito (total ou parcial) e por flip-flops agregados apenas para o teste.

- LSSD (Level-Sensitive Scan Design [Eich77]), que utiliza dois latches com clocks independentes para formar o SSR. É chamado de 'sensível a nível' por não depender dos tempos de transição do clock, sendo imune a 'hazards'. O latch L1 tem duas entradas e dois clocks, para operar no modo normal ou no modo 'scan', e o segundo latch (L2, ver fig. 12) tem um clock, trabalhando entrelaçado com L1.

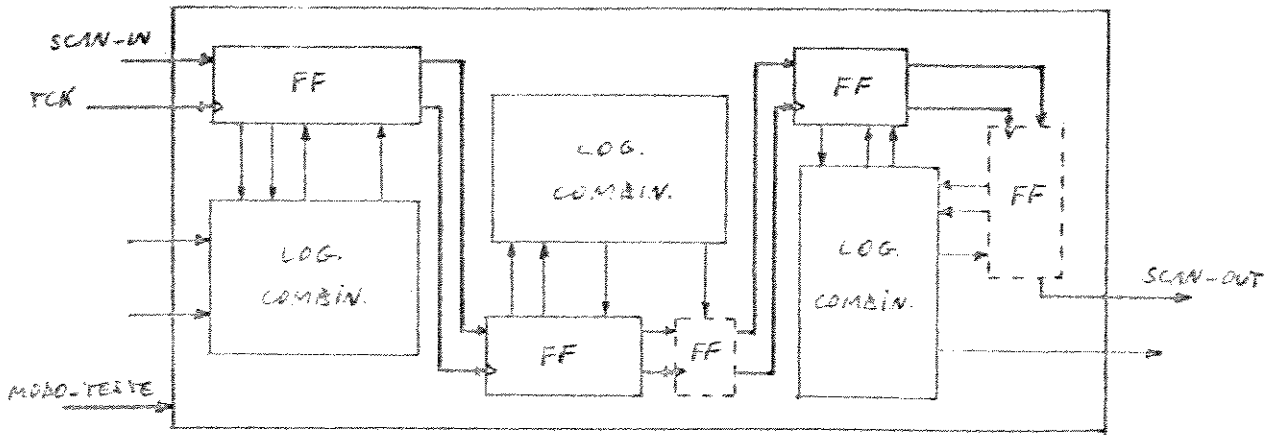


Fig.10 - Estrutura típica do Scan-Path, com o Scan-Shift-Register formado por flip-flops do circuito e outros adicionados

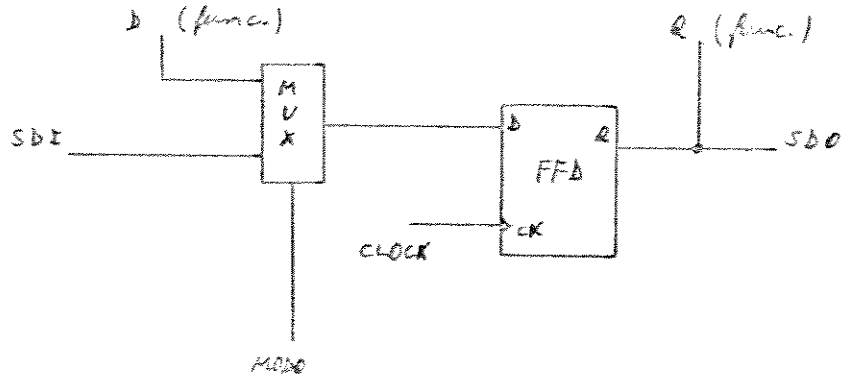


Fig.11 - Flip-Flop modificado para Scan-Test (MFFD)

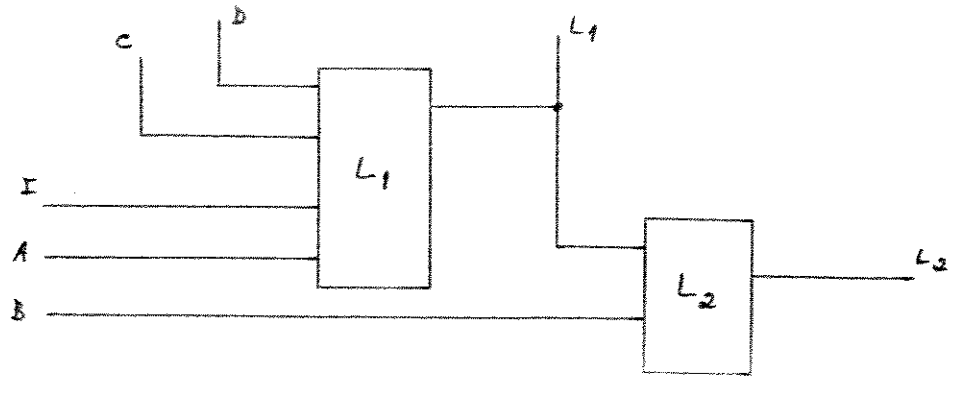


Fig.12 - Estrutura do duplo-latch para LSSD

- Scan-Set, utiliza um shift-register agregado apenas para o teste, o qual introduz estímulos e capta resultados dos flip-flops funcionais, que devem ter duas entradas (fig. 13). Sua vantagem é poder captar dados no modo normal de operação do circuito [Mc84a].

- Multiplexer-Scan, ao invés de utilizar um shift-register para acessar vários nós, utiliza um ou mais multiplexadores para observar os nós internos em sequência. Não atua na controlabilidade.

- Random-Access-Scan, trabalha com latches endereçáveis para controlar e observar os nós (fig. 14). Um decodificador gera os sinais de seleção de cada latch, a partir do endereço, de modo que pode-se acessar cada nó desejado, aleatóriamente. Tem a vantagem de se poder observar os estados internos no modo normal de operação do circuito.

As vantagens do emprego de técnicas tipo Scan-Test, são:

- acesso pelos pinos a muitos pontos internos, para controle e observação dos estados;

- aproveitam-se elementos internos para o teste, permitindo atingir uma alta cobertura de falhas sem ocupar muita área (de 5% a 25% de acréscimo);

- reduz o problema-teste à geração de vetores para blocos combinacionais;

- é uma metodologia de DFT bem estruturada, fácil de ser aplicada a qualquer tipo de circuito lógico.

Suas desvantagens são:

- requer geralmente 4 sinais de teste (MODO, SCAN-IN, SCAN-OUT e TCLOCK), o que pode implicar em mais pinos;

- o tempo de teste é grande pela entrada e saída de dados em série;

- o desempenho dinâmico do circuito piora, pela inclusão de elementos para teste e pela modificação dos flip-flops;

- o roteamento do scan-path é problemático, podendo ocorrer problemas de "race" entre clock e dado [Oliv86].

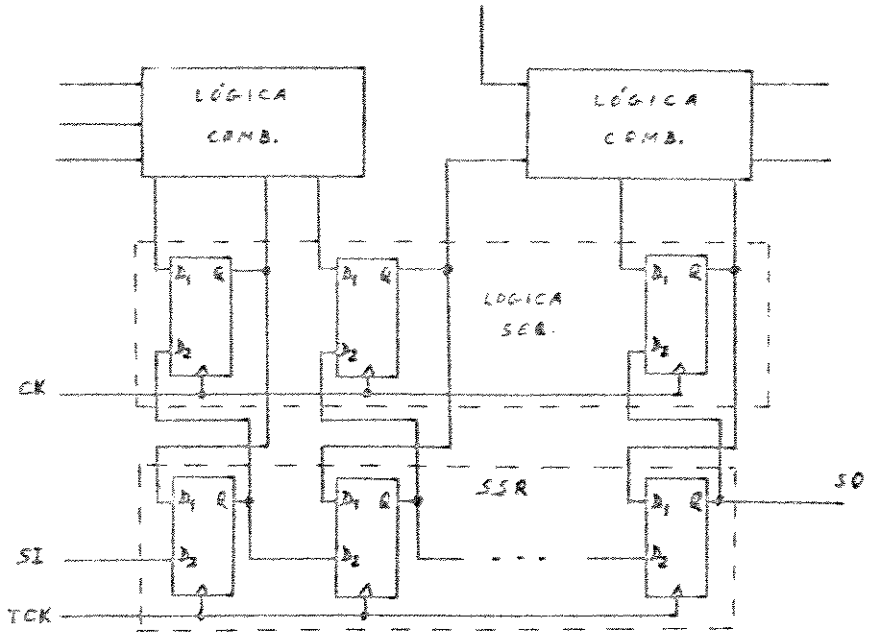


Fig.13 - Estrutura típica do Scan-Set

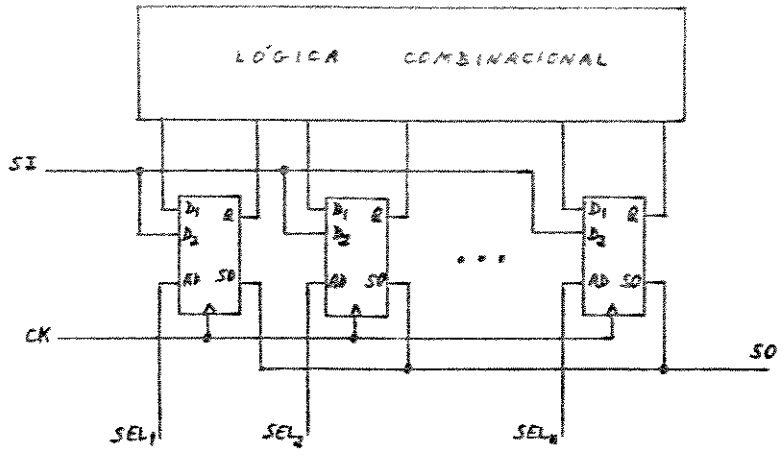


Fig.14 - Estrutura típica do Random-Access-Scan

3.3. Auto-Teste (Self-Test)

Uma solução para reduzir o tempo e o custo do teste de circuitos digitais, é agregar funções de teste no circuito, tornando-o auto-testável, de modo que não dependa de ATE's complexos [Maun85]. Assim foram desenvolvidas as técnicas de Self-Test, ou seja, técnicas de projeto para testabilidade visando a implementação de circuitos auto-testáveis.

A idéia básica das técnicas de self-test (ou BIST: Built-In Self-Test), é agregar ao circuito original blocos lógicos especiais, como contadores, shift-registers, memórias, analisadores de assinaturas e comparadores. Esses blocos de teste têm a função de gerar os dados para estimular o circuito (vetores armazenados ou pseudo-aleatórios), capturar os resultados da lógica funcional, compactá-los (por exemplo como uma assinatura) e comparar o código final com os valores corretos esperados (fig. 15), obtidos para um circuito bom ou por simulação. A lógica de teste é ativada por um sinal de controle (Modo-Teste) e deve operar quase na mesma velocidade de funcionamento do circuito original. Pela inclusão de elementos na lógica original, a velocidade do circuito será menor, mas deve-se minimizar essa variação através de um projeto cuidadoso.

Portanto, com a inclusão desses blocos, o teste é realizado apenas aplicando-se pulsos de clock e observando o resultado final da comparação, num tempo muito curto e a um baixo custo, embora haja um aumento da área final do circuito [Will84].

As principais técnicas para geração de estímulos no próprio circuito são descritas a seguir.

- Armazenamento em memória [McCl85], sendo que os vetores de teste, instruções ou microprogramas (para processadores) são armazenados em ROM, PROM ou EPROM, as quais são lidas durante o auto-teste. É uma boa solução para sistemas microprocessados, onde o teste pode ser realizado por partes (CPU, lógica de controle, memória, I/O, etc.) sob controle do processador. Para lógica aleatória, esta técnica só é viável para blocos pequenos, pela quantidade de memória necessária para armazenar os vetores.

- Geradores concorrentes [McCl85], formados por blocos lógicos como contadores cíclicos ou shift registers com realimentação linear (LFSR, ver fig. 16), os quais geram os estímulos ao mesmo tempo em que são aplicados no circuito sob teste. Esses estímulos podem ser "vetores de peso constante" (ver 2.4), vetores pseudo-aleatórios ou exaustivos, podendo-se aplicar um grande número de estímulos em alta velocidade e sem necessidade de armazená-los. O LFSR é muito utilizado como Gerador de Sequencias Binárias Pseudo-Aleatórias (PRBSG), como por exemplo o PRBSG de 4 bits da figura 16, por ser um bloco simples e pequeno [Komo83]. Os pontos de tomada da realimentação, são escolhidos para se obter a sequência de comprimento máximo no gerador (2 En -1 estímulos), com os polinomios étimos descritos na bibliografia [Froh77].

Utilizando-se geradores pseudo-aleatórios, uma grande quantidade de estímulos de teste pode ser gerada num tempo curto, tornando viável o teste de lógica combinacional e sequencial. Assim não é necessária a reconfiguração do circuito para separá-las, como ocorre quando são utilizados vetores estruturados.

Para se comparar os resultados dos testes internamente ao circuito, estes devem ser capturados e compactados de forma a se obter um número menor de bits. Assim, os bits resultantes da aplicação dos estímulos de teste em um circuito sem falha, após a compactação, podem ser armazenados como uma assinatura, para posterior comparação no próprio circuito. Entretanto, na compactação pode ocorrer o mascaramento de um erro por outro, fenômeno conhecido por 'aliasing' [Maun85], e a probabilidade de ocorrer esse mascaramento depende da técnica de compactação empregada. As principais técnicas de compactação dos resultados empregadas em self-test, são descritas a seguir.

- Contagem de transições do estado 1 para 0 e de 0 para 1 [Maun85], a qual embora simples de ser implementada, não é muito utilizada pela probabilidade de mascaramento de erros ser relativamente alta.

- Verificação da paridade [Cart82], sendo às vezes combinada com outra técnica para melhorar a detetabilidade.

- Contagem de estados 1 (ou 0), conhecida como 'Syndrome Analysis' [Barz81], apesar da pequena probabilidade de mascaramento de erros, só é aplicável com estímulos exaustivos.

- Análise de Assinaturas (Signature Analysis [Froh77,Rob87]), sendo a assinatura gerada pela compactação dos dados em um LFSR, chamado de Registrador de Assinaturas. Esse registrador pode ter uma ou várias entradas (MISR : Multiple Input Signature Register, ver fig. 17), e a assinatura é o código resultante que fica no registrador após um número estabelecido de pulsos de clock. Para uma mesma condição inicial, mesma sequência de dados de entrada e um mesmo número de pulsos, o código armazenado será sempre o mesmo, a não ser que ocorra o mascaramento de um erro por outro. A probabilidade de mascaramento de erros depende do número 'n' de estágios do LFSR, sendo dada por 1/2 En [Froh77]. Assim, para um registrador de assinaturas de 8 estágios, existe uma probabilidade de 0,4% de que um erro (troca de um bit) em uma das linhas de entrada seja mascarado por outro erro, de modo que a assinatura final seja igual à correta, ou em outras palavras, a probabilidade de detetar a ocorrência de mais de um erro em qualquer das linhas de entrada é de 99,6%.

A detetabilidade pode ser aumentada pelo uso de assinaturas múltiplas, pois assim a probabilidade final de mascaramento será o produto de valores fracionários, ficando muito reduzida [Rob87]. A análise de assinaturas é o método de compactação mais empregado em Self- Test, sendo simples de ser implementado e acrescentanoo poucos elementos ao circuito original.

A desvantagem do emprego dessas técnicas de geração e compactação, é que para se estimular e observar muitos pontos internos ao circuito, é necessário acrescentar muitos blocos lógicos, ou adaptar blocos funcionais, ou ainda realizar a multiplexagem das linhas de teste. O acréscimo de área de silício em circuitos integrados, pela inclusão dos blocos de teste, vai de 3% a 25% [Chan82], o que em certos casos pode ser proibitivo. Esta faixa vale também para acréscimo de área em placas de circuito impresso, pela introdução de CI's para auto-teste. Uma alternativa é combinar técnicas como Scan e Self-Test.

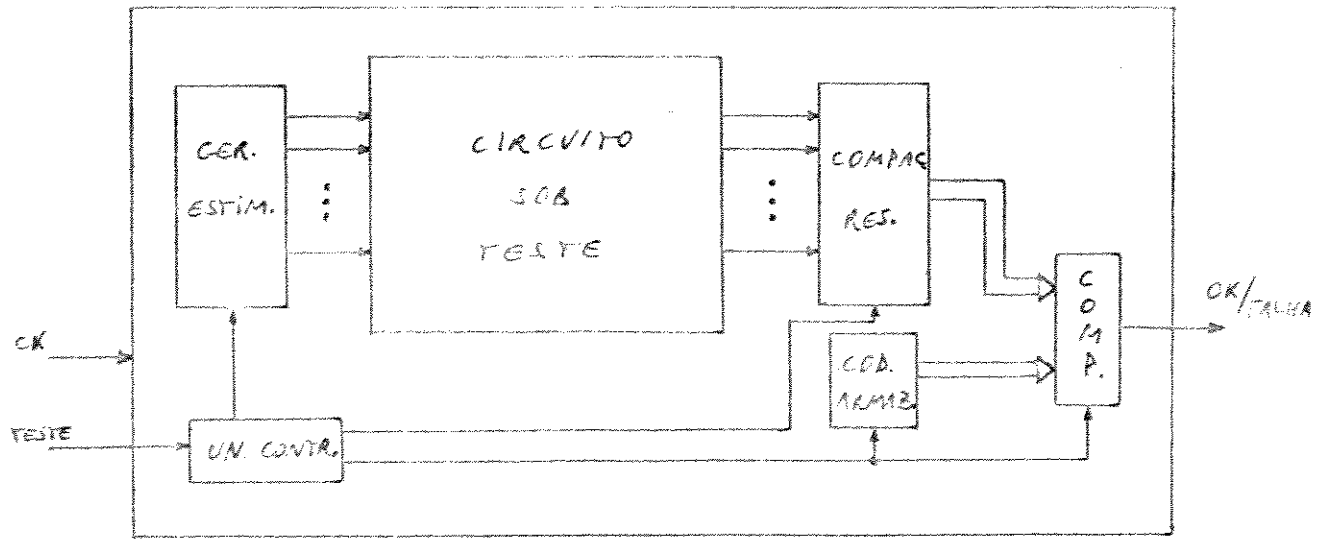


Fig.15 - Estrutura típica de Self-test

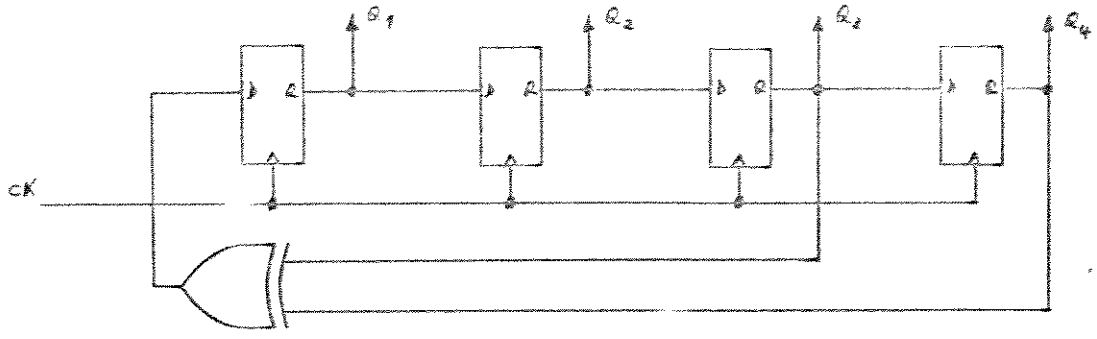


Fig.16 - Linear-Feedback-Shift-Register de 4 bits (LFSR)

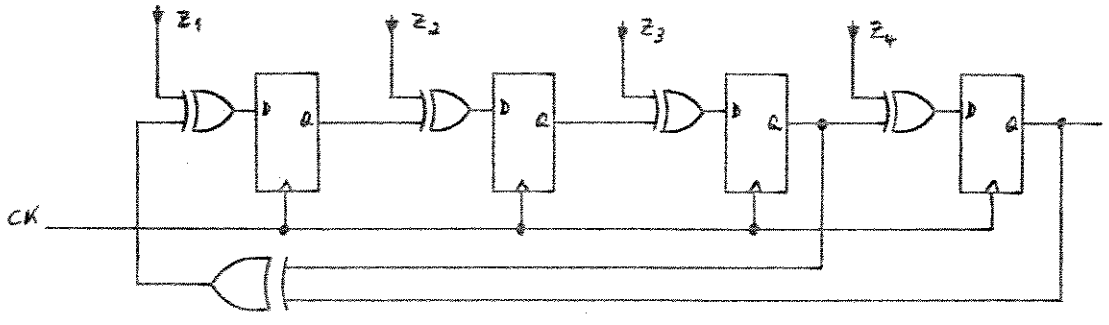


Fig.17 - Um LFSR como Multiple-Input-Signature-Register (MISR)

3.4. Metodologia Combinada: Scan e Self-Test

Combinando-se as técnicas de Self-Test e Scan-Test, pode-se aproveitar as vantagens de cada uma delas na implementação de circuitos com alto grau de testabilidade, ou seja (ver fig. 18), geração e compactação dos dados em alta velocidade e a um baixo custo, distribuição e captura dos dados em muitos pontos do circuito e utilização de parte dos elementos sequenciais do circuito para formar o SSR. Portanto, é muito interessante a combinação dessas duas técnicas, sendo descritas na bibliografia diversas maneiras de implementá-la, utilizando módulos centralizados ou distribuídos pelo circuito [El-Z83, Mc85a]. Alguns autores dão a essa metodologia o nome de S3 (Self-test using Signature analysis and Scan-path techniques [El-Z83a]).

Uma grande vantagem do emprego da metodologia combinada, é que a nível de placas ou sistemas pode-se aproveitar as facilidades de teste atualmente incorporadas a alguns CI's [Kub83, Gels87, Mend86]. O Scan-test tem sido bastante utilizado no projeto para testabilidade de circuitos comerciais, existindo inclusive um grupo de empresas na Europa e EUA tentando padronizá-lo a nível de circuitos integrados para facilitar sua utilização (JTAG - the Joint Test Action Group [Maun86]).

Assim, a metodologia combinada permite a compatibilização dos CI's, placas e sistemas do ponto de vista "teste", possibilitando implementar o auto-teste de todo o sistema e o aproveitamento dos recursos para testabilidade colocados individualmente nos circuitos integrados e placas. O auto-teste de sistemas é importante tanto na sua produção, como para a auto-diagnose em campo, sendo essencial em sistemas tolerantes a falhas [Oli87a].

3.5. BILBO

Uma técnica que simplifica a implementação da metodologia combinada de testabilidade, utiliza um bloco lógico chamado BILBO (Built-In Logic Block Observer [Maun85]) como elemento básico de teste.

O BILBO é um módulo reconfigurável (ver fig. 19), que dependendo dos sinais de controle B1 e B2 pode atuar como um shift-register com realimentação linear (LFSR), latch de aquisição paralela (P.L.), ou ainda um shift-register (SSR), segundo a tabela abaixo.

Tabela de operação do BILBO

B1	B2	modo de operação
0	0	SSR
1	0	LFSR
0	1	reset com CK
1	1	P.L.

Assim, vários BILBO podem ser colocados no circuito para melhorar sua testabilidade. Esses blocos teriam a função de gerar, deslocar, capturar e compactar os dados de teste, agregando versatilidade para execução de scan e self-test e facilidade de implementação. Sua desvantagem é ter um grande número de elementos lógicos (dos quais nem todos são utilizados numa dada configuração), podendo representar um acréscimo de área (e consequente custo) razoável quando empregado no projeto de um circuito integrado de média complexidade, ou quando formado por CI's comerciais SSI e MSI (família TTL e CMOS) para ser aplicado em placas digitais.

O Circuito para Teste Integrado de Placas - CTIP, a ser apresentado, é formado por um BILBO de 8 estágios, sendo uma solução para a aplicação da metodologia combinada em placas eletrônicas, pois reduz o custo de sua implementação.

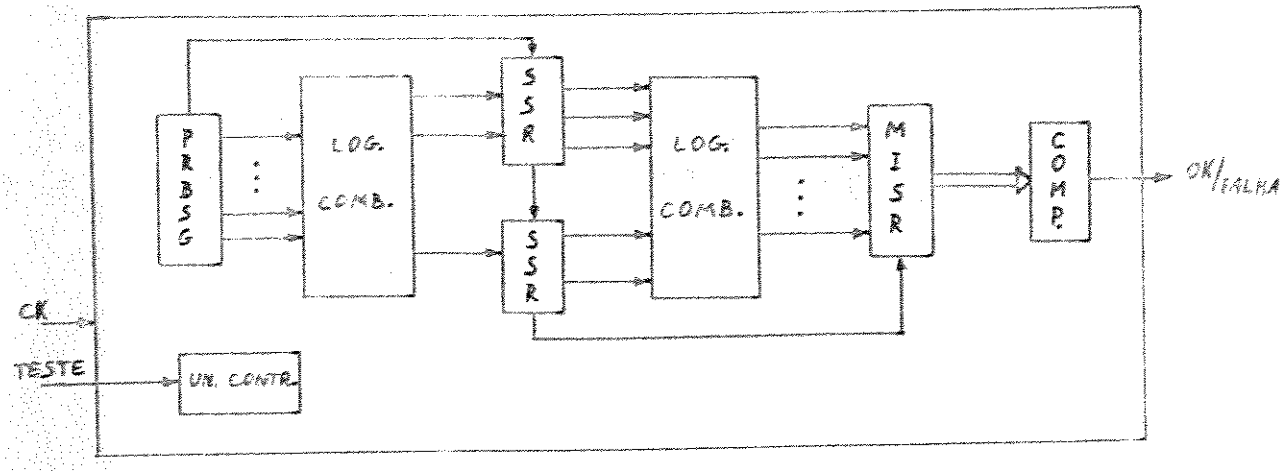


Fig.18 - Metodologia Combinada : Scan + Self-Test

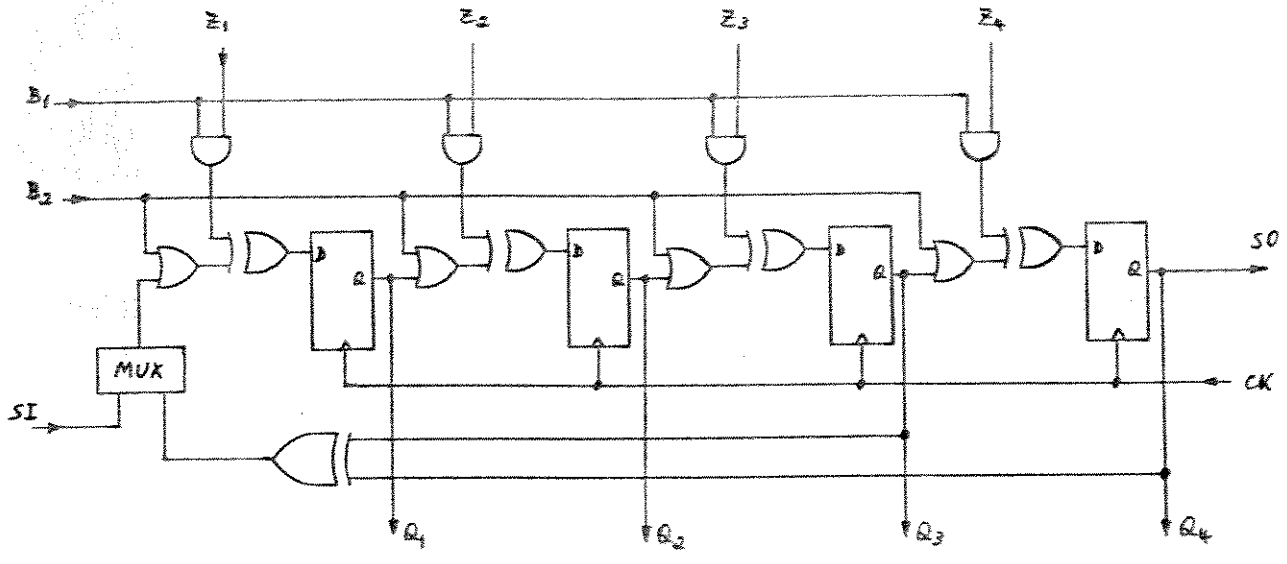


Fig.19 - Estrutura lógica do BILBO

3.6. Teste de Memórias

Por serem arranjos matriciais compactos e compostos de células sensíveis a nível elétrico (as quais podem se influir mutuamente) [Max83], as memórias têm de ser testadas exaustivamente e por algorítmos especiais, que verificam linhas em curto e a inter-relação entre células vizinhas. Além disso, as memórias geralmente têm um comportamento dinâmico crítico, pois se deseja que as mesmas operem na máxima velocidade, e assim os elementos agregados para testabilidade devem influir minimamente.

No caso de memórias só de leitura (ROM's), é necessário que se leia o conteúdo de cada posição durante o teste [Mc84a]. Para diminuir o número de dados a serem adquiridos, pode-se gerar internamente um código compactado, por exemplo "checksum", para comparação do resultado final.

Para as memórias de acesso aleatório (RAM's) existem vários algorítmos de teste consagrados [Glas79, Kino84], os quais garantem um alto grau de confiabilidade do seu correto funcionamento. Esses testes, por serem complexos e demorados, são geralmente executados em equipamentos com processadores dedicados. Alguns dos principais algorítmos são [Glas79]:

- March, em que, partindo-se de todas as células no estado 0, cada posição é escrita com 1 e lida, consecutivamente, e em seguida repete-se esse processo escrevendo e lendo o nível 0 em cada posição. Este teste verifica o funcionamento de cada célula e a decodificação de endereços.

- Galloping ou Ping-Pong, em que, partindo de todas as células em 0, escreve-se 1 na 1ª célula e lê-se todas as células, depois volta a célula 1 para 0, escreve 1 na 2ª célula e lê-se novamente todas as células, e assim por diante até a última. Depois repete-se esse procedimento para os estados complementares. Embora esse teste verifique todas as possibilidades de inter-relação, é muito demorado, podendo atingir algumas horas para uma memória de 256Kbits, por exemplo [Law83].

- Checkerboard, no qual se escreve e lê 0 e 1 em posições intercaladas, como um tabuleiro de xadrez, alternando depois as posições. Testa o funcionamento de todas as células e alguns tipos de interação.

Como os estímulos para teste de memórias são bem determinados, é possível implementar blocos de teste para gerá-los e depois compactar e comparar os resultados, tornando-as auto-testáveis (fig. 20). Entretanto, como já citado, existem restrições importantes para o projeto para testabilidade, como acréscimo de área, velocidade de operação, cobertura de falhas e tempo de teste [Salu87].

3.7. Teste de PLA's

Os Arranjos Lógicos Programáveis, ou PLA, por substituirem grandes blocos de lógica "ramdômica" com economia de área, têm sido amplamente utilizados como blocos funcionais em circuitos VLSI, bem como seus correspondentes a nível de componentes, os Dispositivos Lógicos Programáveis - PLD (ou PAL, ou ainda FPLA). Pela sua estrutura matricial, com a programação definida por elementos nos cruzamentos (ver fig. 21), as PLA estão sujeitas a um tipo de defeito que não pode ser modelado pelo "stuck-at": a existência ou falta indevida dos elementos nos cruzamentos das linhas ("cross-point defects" [Cha78a]), o que altera a função executada pela PLA.

Além do modelo "stuck-at" não ser adequado, o que impossibilita o uso de algorítmos de geração de vetores de teste, o grande número de linhas de entrada das PLA faz com que testes aleatórios não sejam eficientes, resultando numa baixa cobertura de falhas [Will84]. Para PLA's pequenas, com poucas linhas de entrada (por exemplo 8 linhas), podem ser aplicados estímulos de teste exaustivos, mas para PLA's grandes o número de estímulos exaustivos é proibitivo, como por exemplo 2 E20 combinações para uma PLA de 20 entradas.

Assim, têm sido propostas estruturas de PLA com linhas e colunas adicionais para teste por paridade [Fuji84], para as quais os estímulos de teste podem ser definidos facilmente. Uma alternativa muito interessante é adaptá-las para auto-teste [Mend86, Gels87], agregando um LFSR nas entradas para gerar todas as 2 E(n) -1 combinações de estado possíveis, e um MISR nas saídas para compactar os resultados, realizando-se assim um teste exaustivo a baixo custo.

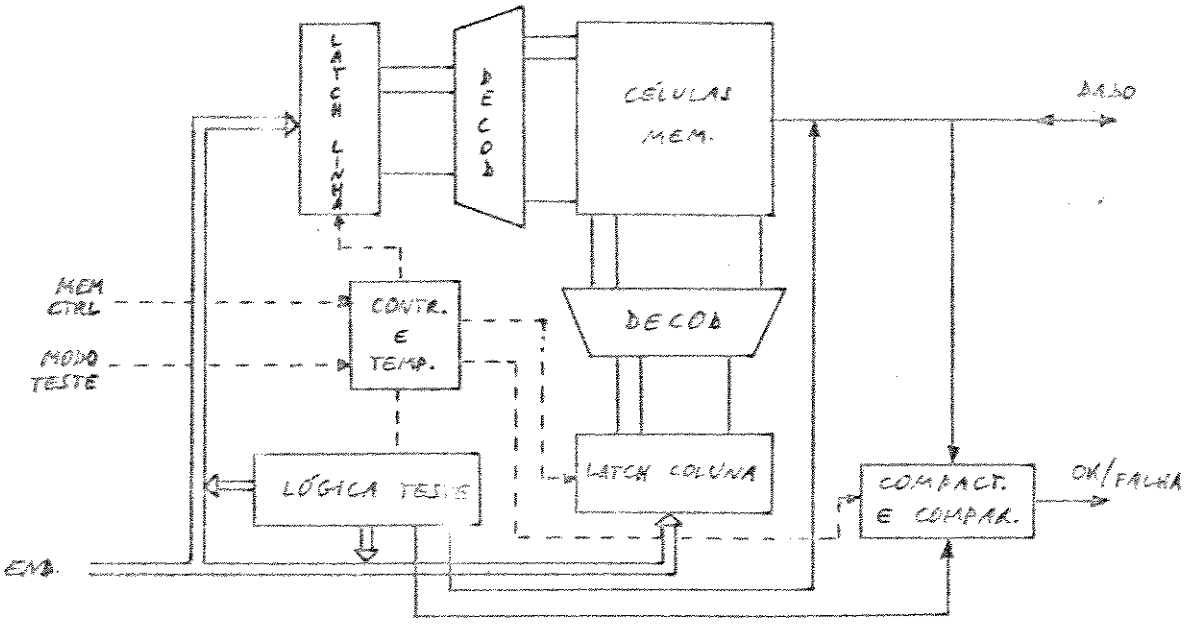


Fig.20 - Memória tipo RAM auto-testável

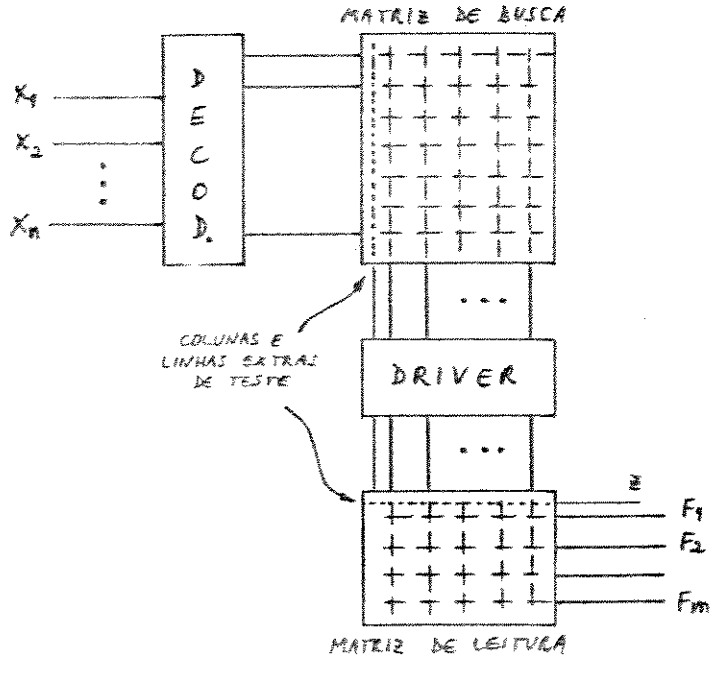


Fig.21 - Estrutura típica de uma PLA

4. CIRCUITO PARA TESTE INTEGRADO DE PLACAS - CTIP

A motivação para o desenvolvimento deste circuito integrado é facilitar o projeto para testabilidade de placas, através de um componente que contenha as estruturas para implementação de Scan-Test e Self-Test, as quais são as técnicas de DFT mais genéricas e eficientes.

O Circuito para Teste Integrado de Placas (CTIP) implementado no 1º Projeto Multi-Usuário CMOS, é composto de blocos lógicos configuráveis, que facilitam o teste de placas digitais. Esse CI é uma ferramenta para a aplicação da Metodologia Combinada em projeto para testabilidade de placas eletrônicas já prontas.

A concepção desse circuito foi baseada em uma família de CI's da Logical Solutions [Log], os quais são registradores de deslocamento para controle ou observação de nós (apenas para SCAN TEST). O CTIP engloba as funções de controle e observação em um único circuito configurável, contendo também estruturas para geração e compactação de dados, utilizadas em SELF-TEST. Cada chip contém um BILBO completo, além de outras funções (ver 4.1), permitindo a redução no tempo e no custo de implementação da metodologia combinada de projeto para testabilidade em placas digitais. Com isso se obtém um circuito modular bastante versátil e poderoso, e vários deles podem ser conectados a uma placa, de modo a torná-la testável sem modificações fundamentais do circuito original (Fig. 22), permitindo a execução de Scan-Test e Self-Test de toda a placa.

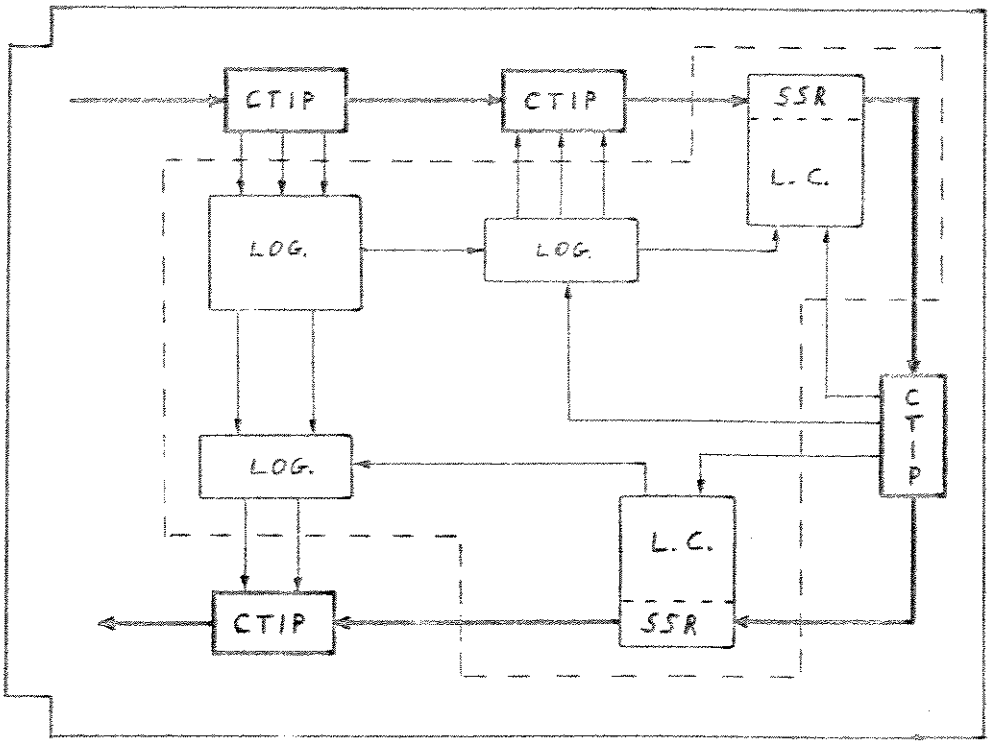


Fig.22 - Esquema para aplicação do CTIP em auto-teste de placas

4.1. Especificações Iniciais

Como especificações elétricas iniciais do CTIP, foram adotados valores de parâmetros compatíveis com TTL e com o processo CMOS do PMU. Assim será possível seu interfaceamento com as famílias lógicas mais utilizadas em placas eletrônicas. Esses parâmetros são tipicamente [Mil883]:

```
VIH   >   2,0V
VIL   <   0,8V
VOH   >   2,4V     (IOH = -400 uA)
VOL   <   0,8V     (IOL =   16 mA)
Ii    <   1 uA     (CMOS)
Fan-out   =  10    (TTL - STD)
fClk   :  até 20MHz
tr, tf  <  25ns    (Cl = 100pF)
```

Foi adotada frequência de clock máxima maior ou igual a 20MHz para que este CI não limite a velocidade de teste dos circuitos, os quais geralmente trabalham até 20MHz (TTL, CMOS).

Em função da área disponível na pastilha (aprox. 5 mm quad.) e da estrutura de aplicação, foram adotadas como especificações funcionais:

- registrador com 8 flip-flops;
- barramento bidirecional tristate (controle, observação ou inibição);
- menos de 40 pinos;
- modularidade, para se poder agregar vários chips formando cadeias de registradores maiores, com 16 ou mais flip-flops;
- operação como registrador de deslocamento (para SCAN-TEST), gerador de estímulos ou analisador de assinaturas (para SELF-TEST);
- operação paralela (ou endereçada) para controle ou observação de estados, visando trabalhos de depuração;
- pino CLEAR para limpar toda a cadeia de registradores.

4.2. Projeto Funcional

Visando as funções acima, o CTIP é composto basicamente do BILBO, para operar como SSR ou LFSR, de uma lógica de endereço, que permite o acesso paralelo aos flip-flops, e drivers bidirecionais tristate (Fig. 23).

O registrador de deslocamento (do BILBO) é formado por 8 flip-flops tipo D. As linhas I/Ø0 a I/Ø7 são bidirecionais, controladas pelo pino C/O, para estimular (controle) ou captar dados (observação) do circuito sob teste.

Os pinos B1, B2 e C/O controlam o modo de operação do CTIP, atuando no BILBO, nos drivers e na lógica de endereçamento (ver Fig. 26), segundo a seguinte tabela:

Tabela I - Modos de Operação do CTIP

B1	B2	C/O	FUNÇÃO (MODO)
Ø	Ø	x	SSR
1	Ø	Ø	MISR
1	Ø	1	PRBSG
Ø	1	x	reset c/ CLK
1	1	Ø	LATCH
1	1	1	reset c/ CLK

O pino CLK é a entrada de clock do circuito, que gatilha os flip-flops na subida do pulso.

Os pinos SI (scan-input) e SO (scan-output) são, respectivamente, a entrada e saída de dados em série do registrador de deslocamento.

O pino CLb é o reset de todo o circuito, independente dos outros sinais.

Na operação paralela, pode-se "setar" (Sb) ou "resetar" (Rb) cada flip-flop separadamente para o controle das linhas de saída, ou observar de forma contínua o estado de cada linha de entrada no pino OO (observation-output). Os pinos AØ, A1 e A2 definem o endereço da linha. Essa operação paralela é comandada apenas por C/O, AØ, A1, A2, Sb e Rb, não dependendo dos outros sinais.

Para evitar a ocorrência de RACE entre clock e dado no shift-register, o que pode causa o armazenamento de estados errados (ver [Oliv86]), deve-se tomar cuidado com o roteamento da linha de clock, de modo que o pulso atinja primeiro o último flip-flop da cadeia, alcançando por último o primeiro flip-flop (Fig. 24).

No projeto do CTIP, o roteamento do clock interno deve seguir essa orientação, e para evitar RACE entre módulos foi incluído um latch tipo D no pino SO (Fig. 25), o qual é habilitado com o sinal de clock em nível baixo. Com isso os dados em SI são armazenados na subida do pulso de clock e só mudam em SO na descida, como em uma operação MASTER-SLAVE.

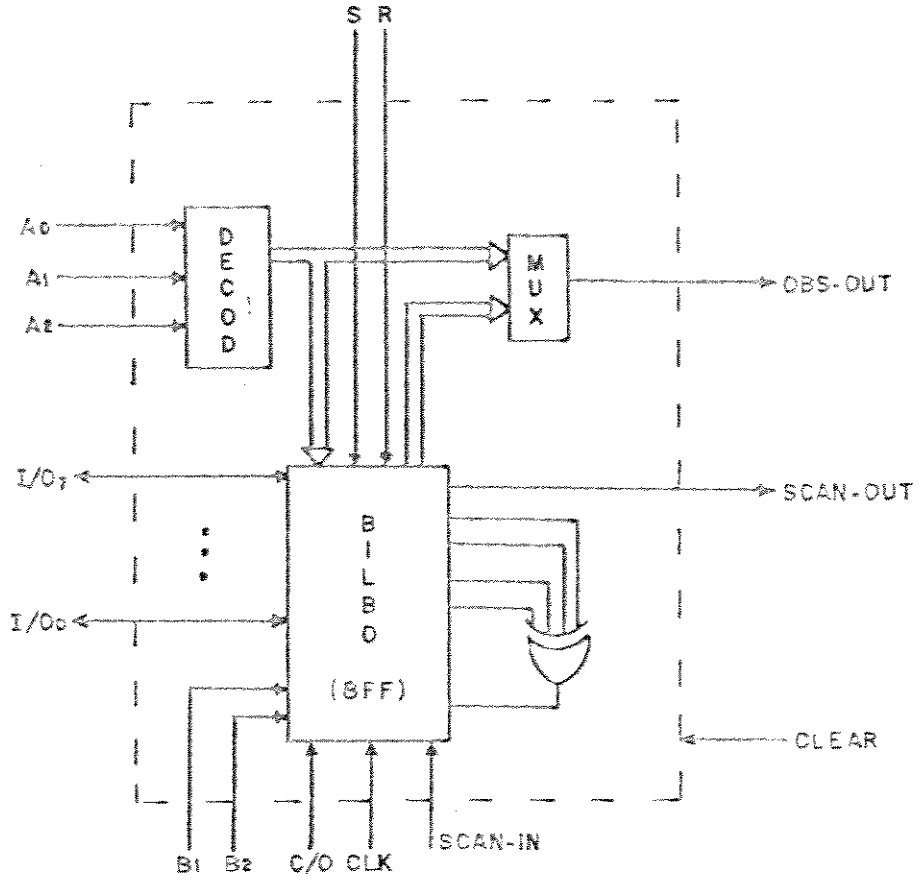


Fig.23 - Diagrama de blocos do CTIP

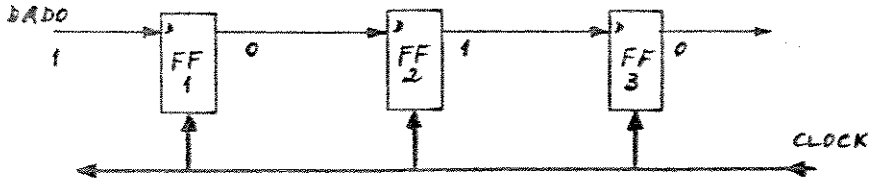


Fig.24 - Esquema de implementação do SSR para evitar "race"

4.3. Projeto Lógico

Para o projeto lógico, partiu-se do diagrama de blocos detalhado (Fig. 25) de modo a se definir macrocélulas que podem ser repetidas. Essas macrocélulas são:

- flip-flop tipo D com lógica de controle e lógica de endereçamento (MACROCIR, Fig. 26);

- multiplexador de entrada (INMUX, Fig. 27);

- lógica de realimentação formada por três portas XOR (FXOR, Fig. 28);

- latch tipo D com clear para a saída SO (LDCL);

- buffers de entrada;

- drivers de saída.

A célula MACROCIR engloba toda a lógica comum aos oito flip-flops da cadeia, de modo que pode ser repetida oito vezes para formar o BILBO. Contém a lógica de controle de entrada (que define o modo de operação segundo B1, B2 e C/O), o flip-flop tipo D com SETb e RESETb, a porta XOR de cada flip-flop do BILBO, a lógica de seleção (endereçamento) para atuar em Sb e Rb de cada flip- flop e a 'transmission-gate' que forma o multiplexador de OO (ver fig.26). Para cada flip-flop só varia a conexão das linhas de endereço (A ou Ab).

O multiplexador de entrada é controlado por B1 para selecionar SI ou o sinal de realimentação de FXOR como entrada do primeiro flip-flop (FF0), em função do modo de operação como SSR ou PRBSG/MISR respectivamente.

A lógica de realimentação FXOR é formada por três portas tipo Exclusive-OR, ligadas às saídas Q2, Q4, Q6 e Q7 dos flip-flops, gerando o sinal de realimentação para o primeiro flip-flop (passando pela célula INMUX), para formar o Shift-Register com Realimentação Linear.

Nos pinos de entrada existem buffers inversores para interfacear circuitos TTL externos (VL < 0,8V; VH > 2,0V) com lógica CMOS interna, com capacidade para excitar até 16 entradas de portas. No pino SI há um buffer não inversor que é ligado a apenas uma entrada de porta.

Os drivers de saída são: não-inversores tristate para as linhas de I/O; inversor tristate para OO; não-inversor para SO. Todos têm capacidade para excitar até 10 cargas TTL. A linha C/O controla os tristates.

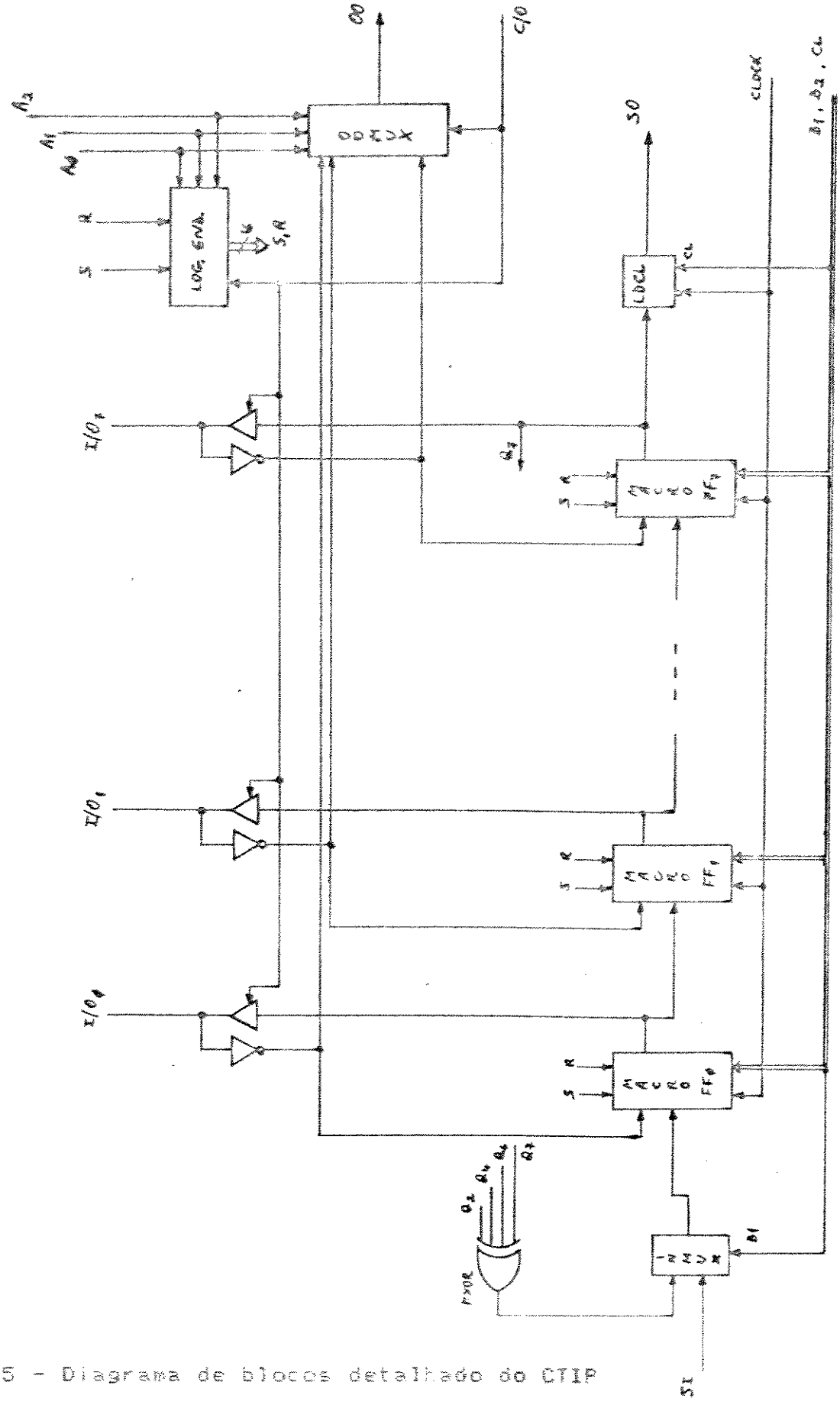


Fig.25 - Diagrama de blocos detalhado do CTIP

Simulação Lógica

Para a simulação lógica foi utilizado o programa SIMUL [CTI85], sendo que para simular atrasos não-unitários foram introduzidos em algumas linhas vários inversores (em número par).

Foi realizada a simulação da célula MACROCTIP, que é uma modificação da célula MACROCIR. A célula MACROCTIP possui um flip-flop ligado à saída Q de MACROCIR, e duas fases de clock (ver Fig. 29). Isto porque inicialmente se pensava em utilizar essa estrutura para evitar RACE. Como o SIMUL não tem elementos tristate, foram utilizados portas NAND em I/O e OO para verificação da lógica.

Para simular o circuito todo do BILBO, utilizou-se a forma hierárquica, sendo que a descrição de MACROCTIP foi utilizada como a macro "mffp", a qual era chamada oito vezes. Foi também conectada a porta XOR de 4 entradas de realimentação, sendo o multiplexador de entrada representado através de portas NAND (ver Fig. 30).

Os arquivos de descrição dos circuitos e as formas de onda resultantes das simulações lógicas da MACRO-CTIP e CIRC-CTIP são mostrados no Apêndice I.

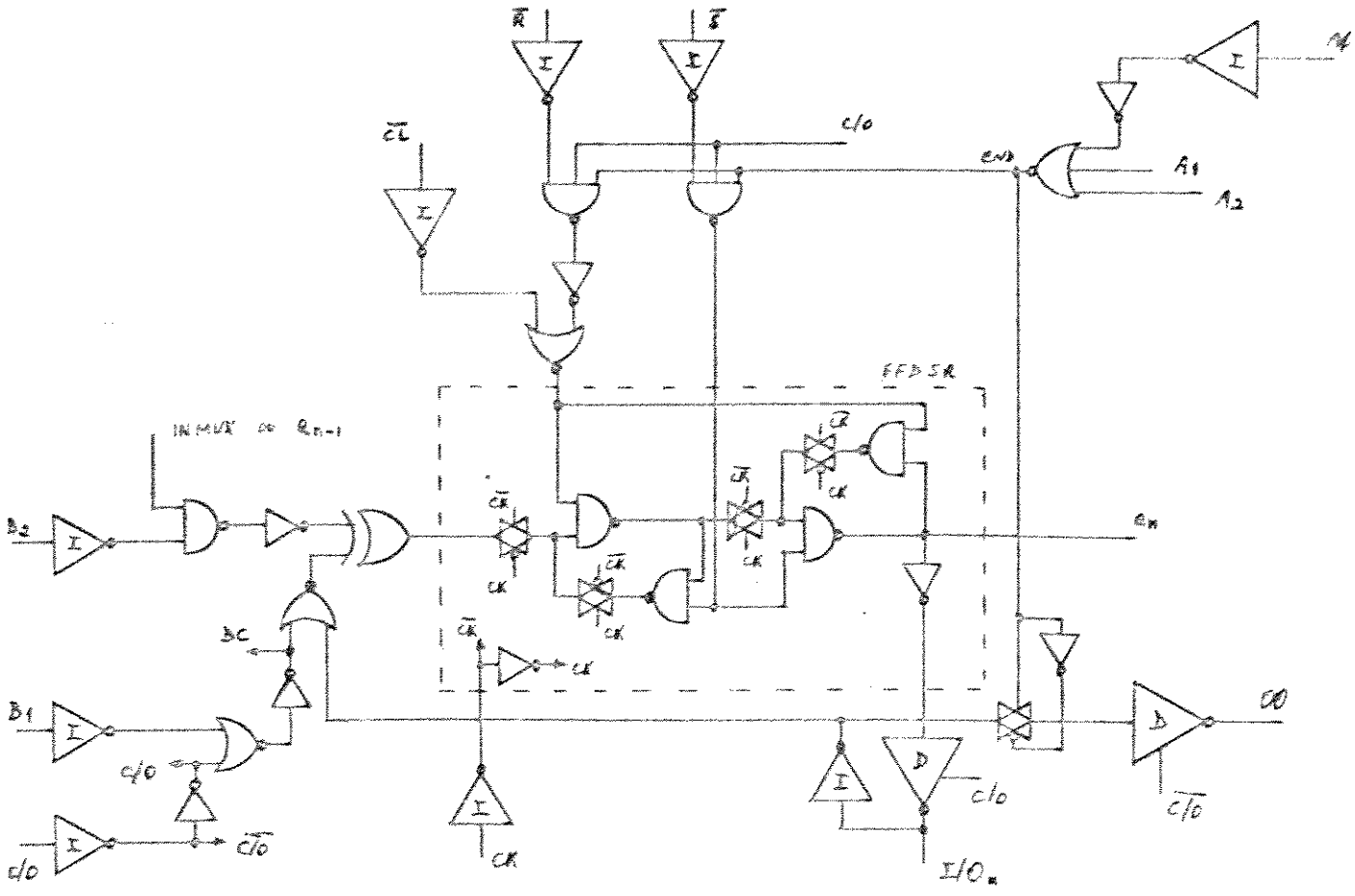


Fig.26 - Célula MACROCIR com buffers e drivers

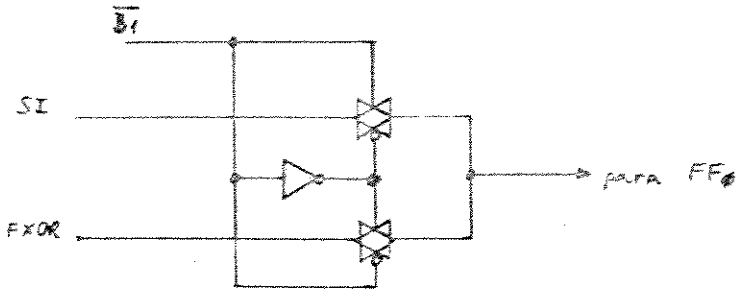


Fig.27 - Célula INMUX

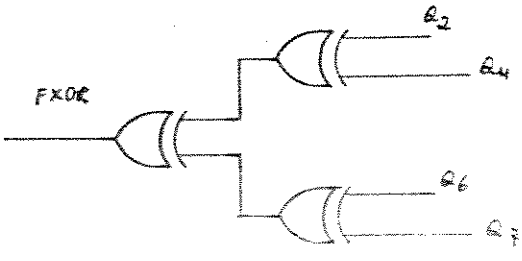


Fig.28 - Célula FXOR

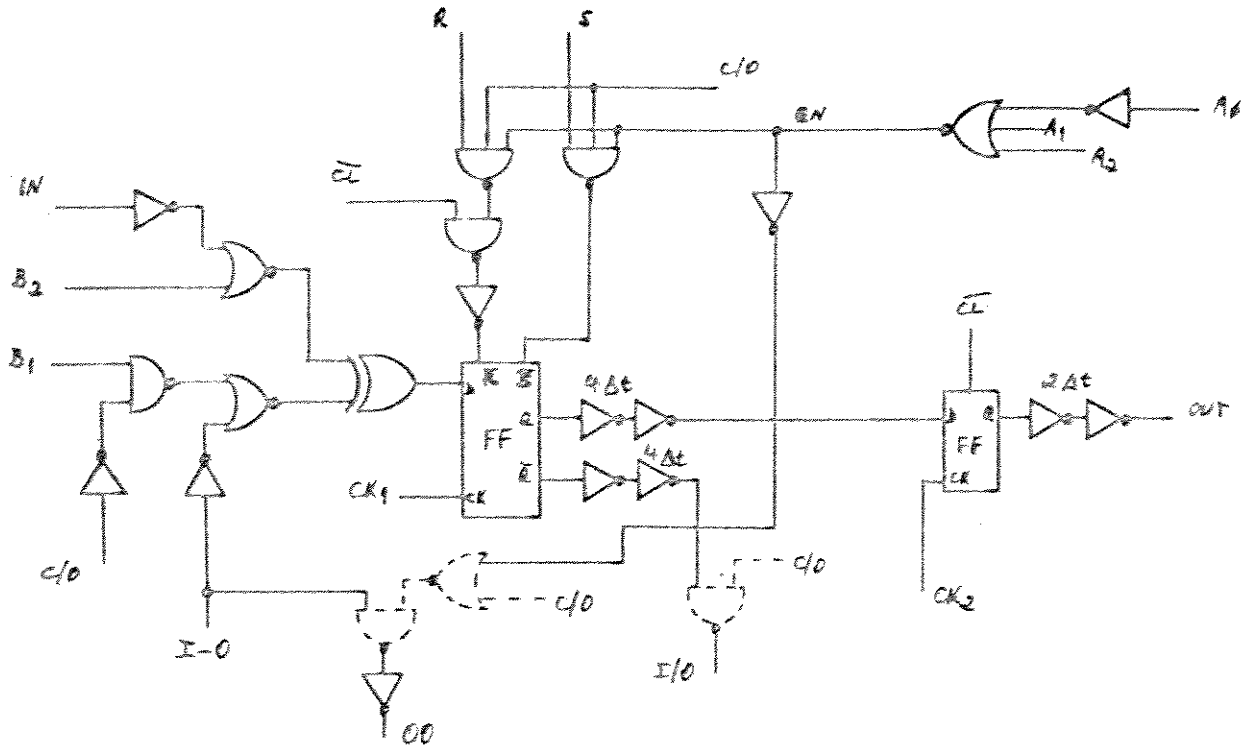


Fig.29 - Célula MACROCTIP utilizada na simulação lógica

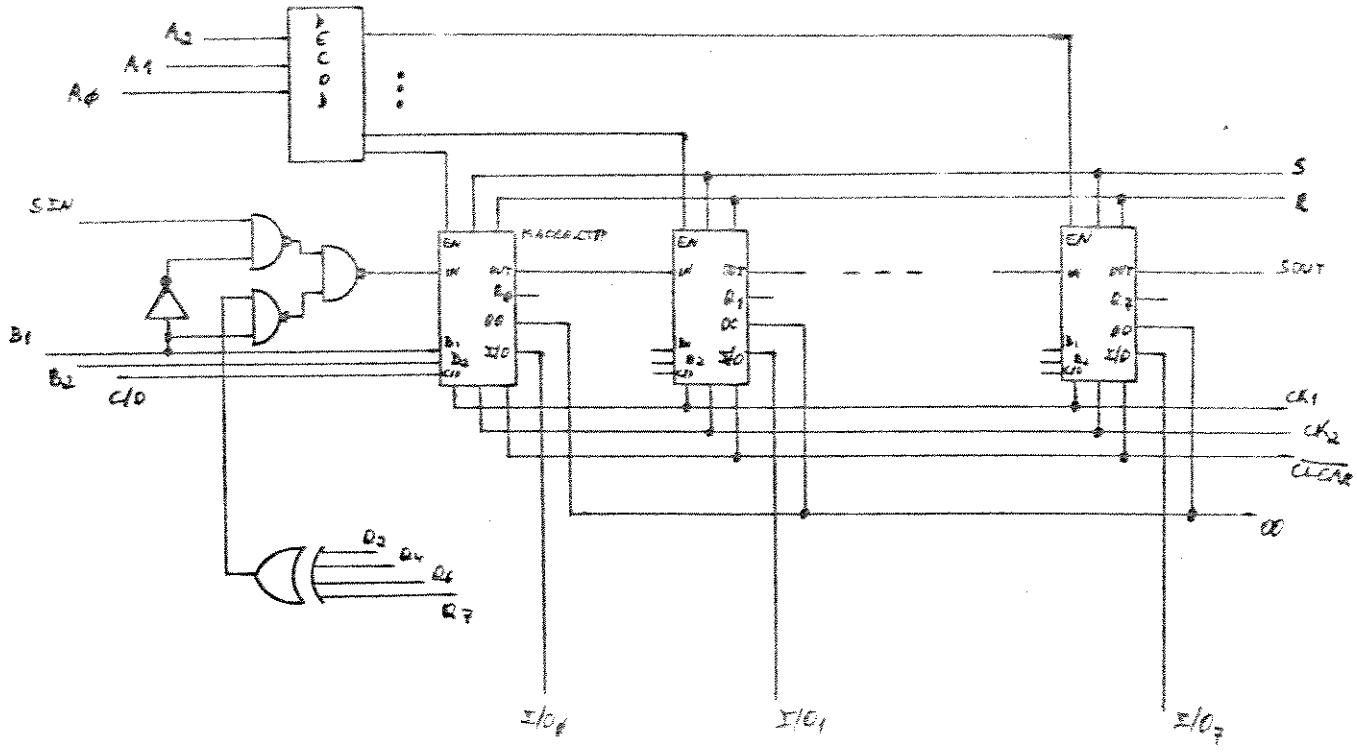


Fig.30 - Circuito para simulação lógica com MUX formado por portas NAND (CIRC CTIP)

4.4. Projeto Elétrico

Inicialmente foi utilizado o tamanho mínimo-ótimo dos transistores, dado por regras geométricas (L = 3 um; W = 6 um, para conter 1 contato), e realizada uma simulação do flip-flop D no SPICE 2G6 [Berk85]. Utilizando-se capacitores de carga de 0,1pF em Q e Qb, foram obtidos tempos de atraso e transição da ordem de 10ns, o que indica que os tamanhos mínimo-ótimos são suficientes para se atingir a frequência máxima de clock desejada (ver especificações).

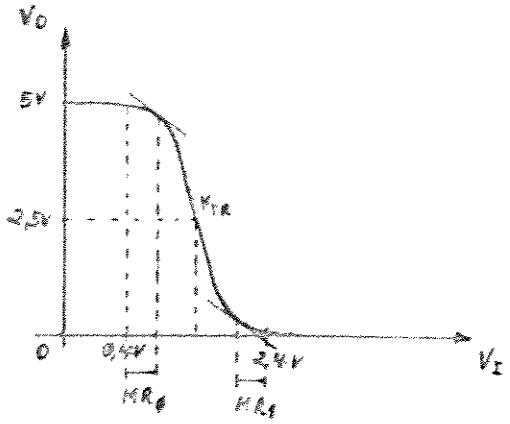
Para se equilibrar a resposta do par PN deve-se utilizar Wp aproximadamente duas vezes o valor de Wn (pela diferença de mobilidade dos portadores), quando então se obtém tr = tf e transição a 2,5V (ver gráficos com os resultados das simulações no Apêndice II).

Os parâmetros elétricos utilizados no SPICE foram os SLOW P - - SLOW N do manual do PMU [CTI86]. As cargas são relativas à capacitância de entrada típica de um inversor básico (Cin = 60fF).

Foram realizadas simulações de um inversor básico (Wn = 6 um e Wp= 12 um) e de um inversor triplo (INV 3X) para avaliar tempos de transição e pico de corrente, com capacitâncias de carga equivalentes a 8 e 30 entradas (CL = 0,5 pF e CL = 2,0 pF). Os tempos de transição obtidos na simulação foram entre 5 e 7ns. Por imposição da topologia adotada no lay-out (para permitir passagem de trilhas sobre os transistores), as dimensões mínimas passaram a ser Wn = 16 um e Wp = 34 um, e pela ressimulação o desempenho é melhor.

Para os buffers de entrada, foi projetada uma célula de interface com os níveis TTL (BUFTTL), através de simulações com uma rampa na entrada e plotando Vo x Vi. Como os níveis TTL são VH > 2,4V e VL < 0,4V, para igualar as margens de ruído para nível 0 e 1, a tensão de transição deve ser de 1,4V (Fig. 31) [Hodg83]. Pela simulação se obteve Wn = 80 um e Wp = 12 um.

Foram projetados dois tipos de buffers de entrada (com interface TTL): um buffer inversor com capacidade de excitar cargas de até 4 pF (BUFINV, Fig. 32) e um buffer não-inversor com baixa capacidade, para excitar duas entradas típicas (BUFNIN, Fig. 33), ver [Hodg83]. Pelas simulações, ambas aceitam níveis TTL na entrada e os tempos de transição são da ordem de 3 a 5ns.



$$MR_0 = MR_1$$

$$V_{TR} = \frac{2,4 + 0,4}{2} = 1,4\ V$$

Fig.31 - Curva Vo x Vi para a interface TTL

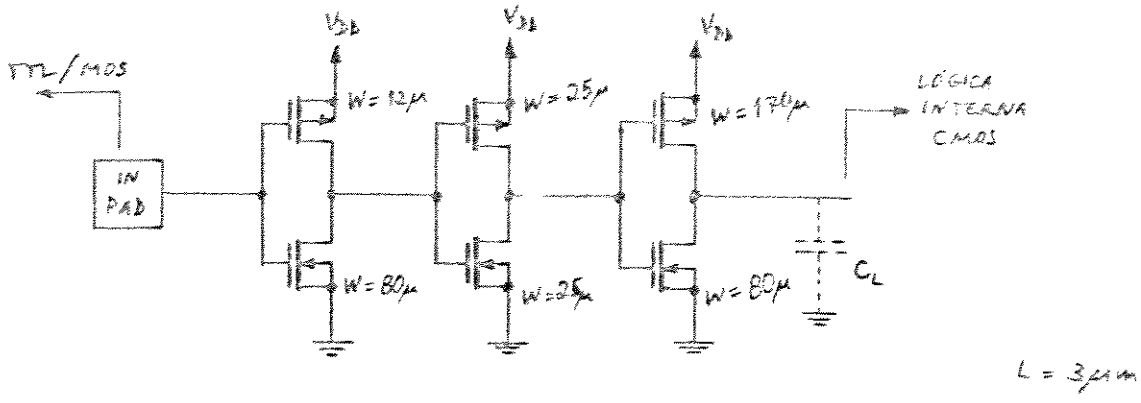


Fig.32a - Célula BUFINV - buffer de entrada inversor

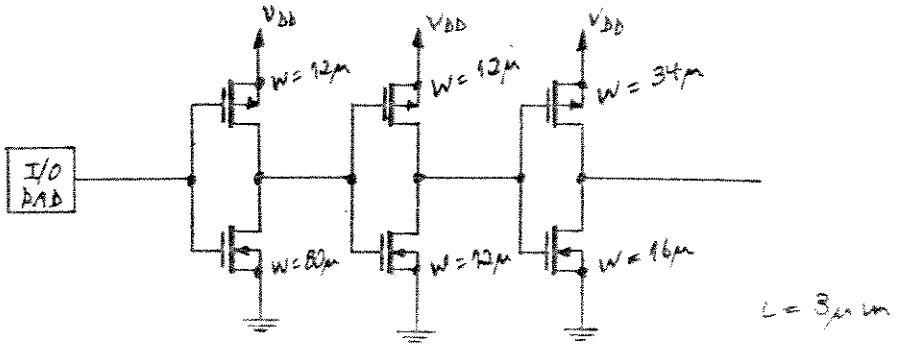


Fig.32b - Célula BUFTTL-IO - buffer inversor das linhas I/O

Para as saídas foi projetado um driver básico, composto de 3 estágios, que deve excitar cargas capacitivas de até 100 pF e cargas resistivas equivalentes a 10 cargas TTL (IOL = 16mA).

O primeiro estágio é um inversor básico interno (INV: Wn= 16 um e Wp = 34 um), o segundo é um inversor triplo (INV 3X) e o terceiro é um par PN com grandes dimensões para excitar as cargas (L = 3 um, Wn = 350 um, Wp = 800 um). Isto nos dá os fatores 3X e 22X em relação ao inversor básico respectivamente. Segundo a literatura (ver [Lin75]), para uma relação de CL/CI = 200 (sendo a capacidade de carga do inversor básico CI aprox. 0,5 pF) seriam necessários os fatores 6X e 36X para um compromisso ótimo entre tempo de atraso e área. Entretanto pela limitação da área disponível na pastilha, foram utilizados fatores menores (3X e 22X), o que pela simulação elétrica é satisfatório, fornecendo tempos de atraso da ordem de 16ns, tempos de transição da ordem de 20ns e VOL = 0,8V para IOL = 16mA (ver simulação DR3ST).

Utilizando-se esse par PN de saída podem ser implementados drivers não-inversores e drivers tristate (Fig. 34), a serem utilizados no CTIP.

Para o projeto da lógica funcional interna, as dimensões do inversor básico foram utilizadas para todos os elementos lógicos. Inicialmente foi utilizado Wn = 6 um e Wp = 12 um e foram realizadas várias simulações, partindo do flip-flop D e acrescentando circuitos até se chegar à célula MACROCIR (ver Fig. 26), inclusive com os buffers e drivers conectados. Foram estimadas capacitâncias de carga para os sinais em função do número de entradas básicas a serem estimuladas. Foram verificados o funcionamento do circuito e os tempos de resposta (atrasos e transição) de modo a se poder atingir clock máximo até 20MHz. Posteriormente foram realizadas ressimulações com Wn = 16 um e Wp = 34 um, que são as dimensões impostas pela topologia adotada, como exposto anteriormente.

Os arquivos para o Spice e os resultados das simulações elétricas são apresentados no Apêndice II.

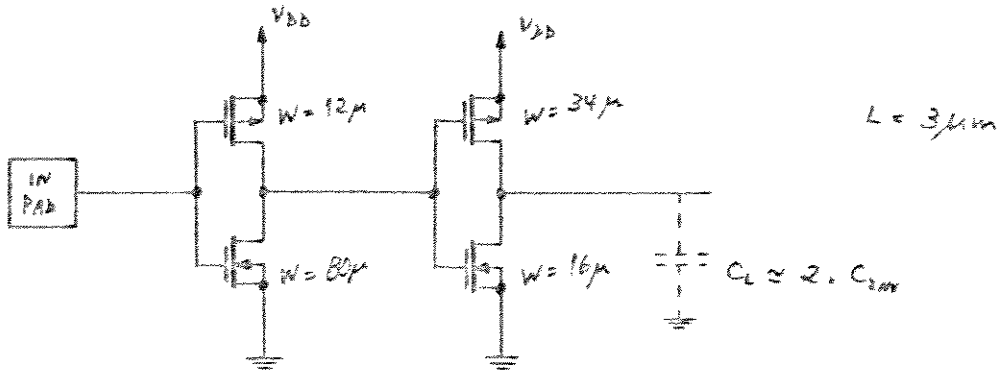


Fig.33 - Célula BUFNIN - buffer de entrada não-inversor

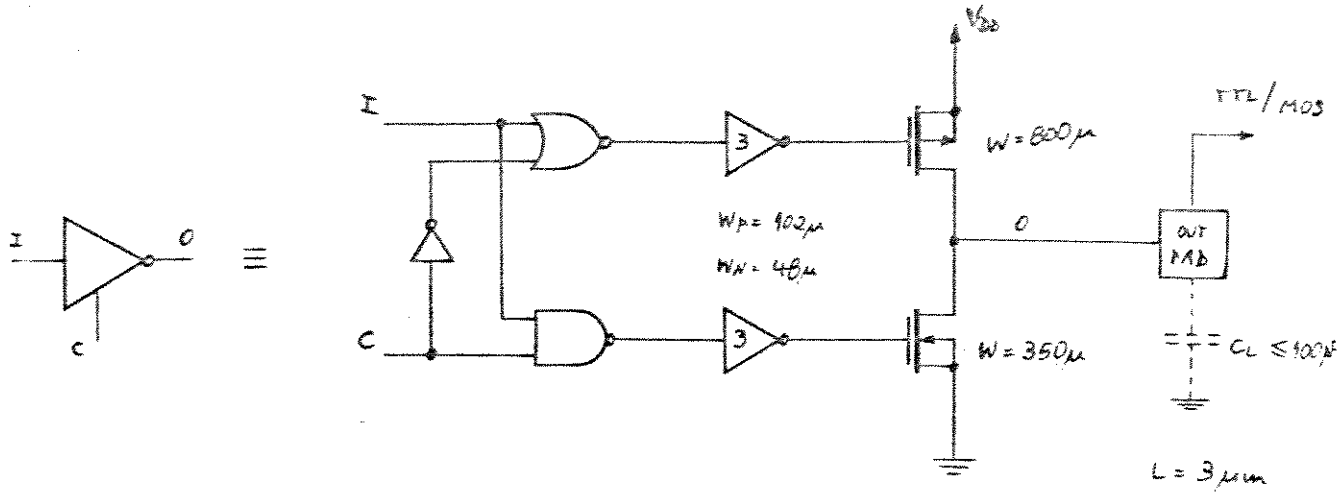


Fig.34 - Diagramas da célula DR3ST - driver tristate de saída

4.4. Projeto Gráfico

Para o projeto do 'layout' do circuito foi utilizado o editor gráfico KIC [Berk85a], com base nas regras geométricas mínimas do manual do PMU [CTI86].

Foi utilizada a estrutura de macros, com o desenho de várias células básicas que foram interligadas para formar células mais complexas.

Para o projeto da parte lógica interna foi levada em conta a distribuição das linhas de controle, que acionam várias células. Assim foi adotada uma estrutura tipo 'célula-canal' (Fig. 35), com os transistores P e N dispostos no sentido vertical da figura e as linhas de controle e alimentação caminhando na horizontal.

Os desenhos das células principais e do circuito completo estão no Apêndice III.

A célula MACROCIR contém o flip-flop tipo D e toda a lógica de controle e endereçamento. As dimensões dos transistores são basicamente L = 3 um, Wn = 16 um e Wp = 34 um. Os contatos têm dimensões mínima (3 x 3 ou 5 x 5 um). Trilhas longas de poli-silício têm largura de 5 um e trilhas longas de metal têm largura de 6 um. O espaçamento entre as trilhas paralelas de sinal é de 4 um (10 um para clock). As trilhas de alimentação internas são de 10 um, sendo de 40 um na periferia.

Para definir o endereçamento de cada célula, as linhas A0, A1 e A2 podem ser conectadas a um inversor ou direto à NOR. Do mesmo modo as saídas Q podem ser conectadas a linhas de metal para realizar a realimentação através de FXOR.

Para o projeto dos diversos buffers e drivers foram desenhados: um transistor PMOS de saída, duas configurações do NMOS de saída, pads sobrepostos a poço P, duas configurações de dispositivo de proteção, dois circuitos de acionamento dos drivers (controle tristate; estágio intermediário mais flip-flop), e dois circuitos de entrada com interface TTL (buffer inversor = BUFINV, Fig. 32; buffer não-inversor = BUFNIN, Fig. 33).

Os dispositivos de proteção de entrada contra descargas eletrostáticas (ESD), foram desenhados com base nas dimensões dos dispositivos da Biblioteca de Células-Padrão desenvolvida no Instituto de Microeletrônica do CTI [Gio86], sendo modificados em função do exíguo espaço disponível. São compostos de um resistor de 1K8 ohm de difusão P, que forma um diodo distribuído para VDD, e de um diodo quadrado para VSS, ambos cercados por anéis de guarda (Fig. 36). O diodo de VSS forma na realidade um transistor NPN, com o coletor em VDD, de modo que a maior parte da corrente (para VI < VSS) vem de VDD. Como não é possível simular seu funcionamento, eles serão caracterizados depois da fabricação.

O transistor PMOS possui L = 3 um e W = 840 um, foi dividido em três transistores (W = 280 um) em paralelo e dobrado em forma de "L" para aproveitar espaço. Existe um anel de guarda ao seu redor para evitar "latch-up" [Boyd85, Herm85], e foram feitos vários contatos de dreno e fonte para reduzir a resistência de contato final, em função das correntes de saída envolvidas.

As duas versões dos transistores NMOS têm dimensões totais de W = 370 um. Em um caso são 3 transistores (W = 123 um) em paralelo (células de I/O) e no outro caso são quatro transistores (W = 93 um) em paralelo, para economizar área. Ambas possuem anel de guarda e vários contatos de dreno e fonte (ver Apêndice III).

Pela simulação elétrica, o pico de corrente nos transistores foi de 50mA. Segundo informações obtidas com a "foundry" (ver [Gio86]), para o cálculo da largura das trilhas de metal e do número mínimo de contatos, considera-se a relação entre a corrente de pico e a corrente estática igual a 6, e assim tem-se:

$$Ipt = \frac{2}{3} Ipico = 34mA \quad ,$$

Ipt: corrente na trilha comum a 2 dos 3 transistores
Ipico = 50mA

$$l = \frac{Ipt}{n\ d} = 6\ um \quad ,$$

l: largura mínima da trilha
d = 1mA/um , densidade de corrente no metal
n = 6, razão Ipico/Iestática

$$Nc = \frac{Ipt}{n\ dpc\ pc} = 3 \quad ,$$

Nc: número de contatos necessários
dpc = 0,2mA/um, densidade de corrente no contato
pc = 12 um, perímetro de 1 contato de 3 um

Para cargas TTL, a corrente estática pode atingir 16mA, o que implica:

$$Ipt = \frac{2}{3} (6 \times 16mA) = 64mA \quad , \quad \text{e então } l = 11 \text{ um e } Nc = 5$$

Além disso, se a saída for chaveada em frequências altas (próximas de 20MHz), Iestática pode ser próxima de Ipico/3, o que exigiria trilhas de 12 um.

Portanto, o número de contatos utilizados (Nc) 18) é mais que suficiente, mas pela restrição de área foram utilizadas trilhas de metal de 7 um, o que pode ser crítico para 10 cargas TTL, ou frequências altas.

Quanto aos circuitos de acionamento dos drivers, foi projetado um controle para os drivers tipo tristate (CTRIO) composto de duas portas lógicas e inversores triplos (ver Fig. 34), para acionar os gates do PMOS e NMOS de saída, e foi projetada uma célula básica composta de um latch tipo D e um inversor triplo (LDCL, Fig. 37), para a saída SCAN-OUT.

Utilizando essas células básicas, foram montadas todas as configurações para entrada e saída, como IO3ST, OODRIV, SCANOUT, INPBUFFER e SINBUF (ver Apêndice III), que são as células gráficas principais.

As outras células principais são: as portas XOR de realimentaçào (FXOR), o multiplexador de entrada (INMUX) e a célula MACROCIR, com as quais foi montado todo o circuito (CHIPCTIP).

As oito células MACROCIR interligadas, que compõem a lógica funcional (BILBO), formam a célula MACROEXEMPLO. Foram colocados dois inversores em série na linha de clock, de modo que após excitar as quatro últimas células, o clock é "bufferizado" para excitar as outras quatro (FF3 a FF0).

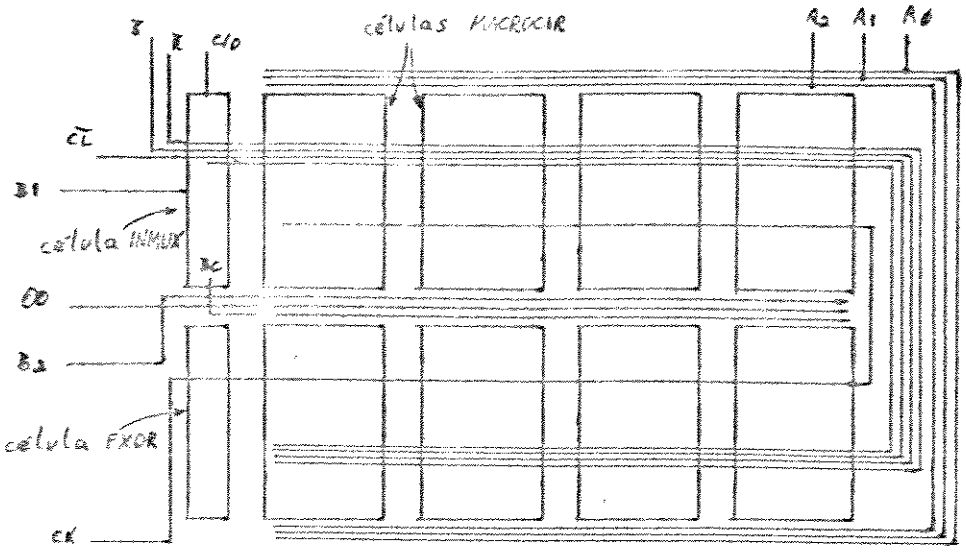


Fig.35 - Estrutura Célula - Canal utilizada no lay-out

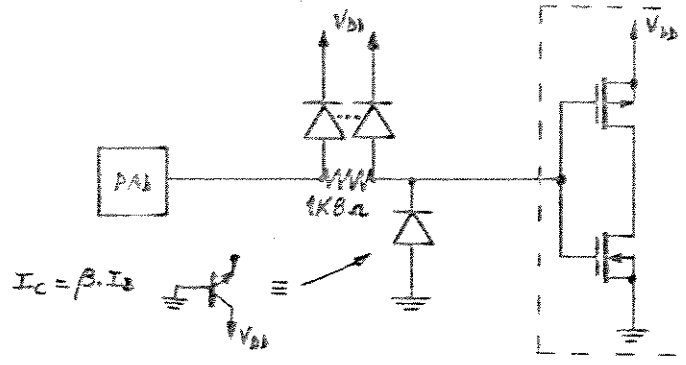


Fig.36 - Diagrama do Dispositivo de Proteção contra ESD

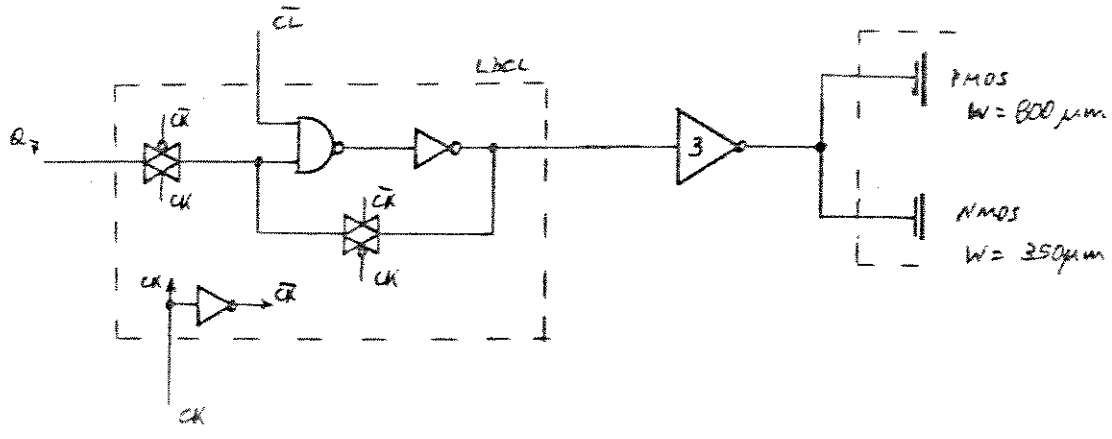


Fig.37 - Célula LDCL do pino SCAN-OUT

4.6. Testes de Validação

Para caracterizar a pastilha CTIP foram realizados testes elétricos e ensaios de descarga eletrostática.

Os testes elétricos compreendem testes funcionais, de modo a verificar os modos de operação e integridade da lógica, e testes paramétricos, para avaliar os parâmetros mais importantes como freqüência máxima de operação, capacidade dos drivers e reconhecimento de níveis TTL nas entradas. Esses testes visam realimentar o projeto e as simulações, e foram verificados os valores obtidos com o SPICE.

4.6.1. Teste Funcional

Para o teste funcional não foram gerados vetores de teste específicos, porque como o circuito contém um BILBO, ele é facilmente testável. Basicamente o circuito é colocado para operar nos modos de geração (PRBSG), deslocamento (SSR) e análise de assinatura (MISR), segundo os estímulos utilizados na simulação lógica, sendo obtidas duas assinaturas (n' e N'' indicadas abaixo) que indicam se o componente está bom. Foi utilizada a montagem da figura 38.

Procedimento

A seguir são descritas as operações básicas de teste, com a respectiva tabela de estados das entradas e saídas, as quais cobrem as possibilidades funcionais:

Inicialização - Modo RESET c/ CK

a) Modo SSR - inicialização através de "scan-in" de 8 estados "1", com observação endereçada simultânea de cada linha de entrada (I/O = 0) na saída OD (operação paralela);

b) Modo PRBSG - partindo de todos 0 em "1", operação como gerador de seqüências pseudo-aleatórias, sendo colhida uma assinatura n' nas linhas I/O após n pulsos de clock (n=99, n' = 11000011);

c) Zeramento de toda a cadeia de flip-flops através de CLb, com observação nas saídas (I/O);

d) SET endereçado de cada flip-flop, através de Sb, com observação nas linhas de saída I/O (operação paralela);

e) RESET endereçado de cada flip-flop, através de Rb, com observação nas linhas I/O (operação paralela);

f) Modo PARALLEL LATCH - aquisição paralela de 8 estados '1' das linhas de entrada (I/O);

g) Modo MISR - partindo de todos 0 em '1' e mantendo as entradas (I/O) em '1', operação como registrador de assinaturas de entradas múltiplas, sendo que após N pulsos de clock estará armazenada uma assinatura conhecida N° (N =99, N° = 01010001);

h) Modo SSR - a assinatura N° de g) será retirada no pino SO através de 'scan-out', com observação endereçada simultânea de cada linha de entrada (I/O = 1) na saída OO (operação paralela);

i) Modo RESET c/ CK - reset de toda a cadeia com a aplicação de um pulso de clock e observação nas linhas de saída I/O.

OBS:
- como foram detetados erros de projeto (descritos a seguir, em Resultados) que afetam SO e OO, utilizou-se C/O=1 nos passos a) e h) para ativar OO, que então passou a observar as linhas de saída;

- para observar N° no passo h) (pelo erro em SO), não foram aplicados os 8 pulsos de clock, retirando-se a assinatura nas linhas I/O, com C/O = 1.

TABELA II - DADOS PARA TESTE FUNCIONAL

MODO	CLb	B1	B2	C/O	SI	I/O	A0-2	Sb	Rb	CK	I/O	SO	OO
RESET(CK)	1	0	1	1	1	0	0	1	1	1	0	x	x
SSR+OO(0)	1	0	0	0	1	0	0-7	1	1	8	z	x	0(8)
PRBSG	1	1	0	1	x	x	x	1	1	n	x →n'	x	z
CL		0	0	1	x	x	0	1	1	-	0	x	z
SET end.	1	0	0	1	x	x	0-7	0	1	-	1	x	z
RESET end.	1	0	0	1	x	x	0-7	1	0	-	0	x	z
P.Latch(1)	1	1	1	0	x	1	x	1	1	1	z	x	x
MISR	1	1	0	0	x	1	x	1	1	N	z	x	x
SSR+OO(1)	1	0	0	0	1	1	0-7	1	1	8	z	N°	1(8)
RESET(CK)	1	1	1	1	1	x	x	1	1	1	0	x	z

Condições

Os testes funcionais foram realizados à temperatura ambiente (25 +- 2°C), com alimentação de 5V (+- 2%), com níveis de entrada de 0V e 5V e sem carga nas saídas *. O equipamento de análise das saídas (Analisador Lógico) teve seu limiar de decisão entre níveis alto e baixo ajustado para 1,4V.

Inicialmente foi utilizada uma frequência de clock baixa (1 MHz), sendo depois aumentada para o teste dinâmico (até 13 MHz).

* OBS: Foram colocados resistores de 10K nas saídas (p/ GND e VDD) para evitar que flutuem quando em tristate, e para garantir que as saídas conduzem para "1" (res. em GND) e para "0" (res. em VDD).

Resultados

Foram detetadas falhas funcionais em DO e SO nos passos a) e h), concluindo-se após análise do circuito que foram cometidos dois erros no projeto das máscaras:

- a célula DODRIV foi conectada à linha C/O e não C/Ob como deveria ser, passando DO a operar no modo de controle (C/O=1), observando as linhas de saídas I/O;

- a célula SCANOUT foi conectada à linha CL, e não CLb como deveria, portanto o pino SO está sempre em 0 durante a operação do circuito (CLb=1).

Assim, foram adotadas modificações no procedimento para teste funcional, sendo que o resto do circuito opera corretamente.

4.6.2. Teste Paramétrico

As tabelas III e IV contém os parâmetros, condições de medida, valores medidos e valores esperados com as unidades, para as 10 peças testadas.

Para levar as saídas a nível "0" ou "1", para efetuar as medidas, foram utilizadas partes do teste funcional. Como cargas, foram utilizadas fontes de corrente ajustáveis. Do mesmo modo, para verificar a resposta do circuito a níveis extremos de entrada (VIHmin e VILmax), foi rodado o teste funcional com níveis do gerador de dados alterados.

Para determinar o consumo, foi medida a corrente quiescente (Iest) e a corrente dinâmica (Idin) em função da frequência de clock (parte do teste funcional), com as saídas em aberto.

Como parâmetros AC foram medidos o tempo de subida e descida nas saídas em função da carga capacitiva, o tempo de atraso entre clock e saídas e tempos de atrasos para SET, RESET, CLEAR, e C/D (nas saídas). A faixa de alimentação operacional (VDD min, max) foi avaliada utilizando parte do teste funcional.

Resultados

Os parâmetros medidos para as 10 amostras testadas estão dentro da faixa esperada segundo as simulações. Apenas a corrente de consumo estática e a frequência máxima de clock não corresponderam ao esperado.

O valor médio de Iest é de 5,2 mA, quando deveria ser menor que 1 uA, pois não há componentes que drenem corrente estaticamente, mas como não havia um equipamento tipo microprovador para análise interna das estruturas do chip, isto não pode ser explicado.

A frequência máxima de clock obtida foi de 13 MHz, o que está de acordo com os tempos de atraso medidos para o chip, como por exemplo tCK = 25 ns (cabe ressaltar que os testes foram realizados em uma placa experimental tipo "Protoboard", que tem capacitâncias parasitas altas). O valor de VIHmin para as linhas I/O está alto porque para executar o teste foi necessário colocar resistores em série com os canais do gerador de dados para as linhas bidirecionais (ver fig.38), o que altera a medida.

Embora alguns parâmetros tenham variado, o resultado geral é bom, e o CTIP está apto para ser usado em placas.

Tabela III - RESULTADOS DOS TESTES PARAMÉTRICOS DC

Proj - CT1P-A Data - Condições - V_{DD} = 5,0 V T_{amb} = 24 ± 3°C

Param.	Condição.	1	2	3	4	5	6	7	8	9	10	Un.
V_{DD} min/max	funcional	3,0/6,5			3,0/6,5							V
I_{EST}		4,1	5,9	5,5	5,8	5,5	5,2	5,8	4,5	5,3	4,5	mA
I_{DIN}	1MHz C_L ≈ 40pF	6,0			7,5							mA
V_{IL} max	funcional	←	←	←	←	←	←	←	←	←	←	0,8 V
V_{IH} min	funcional entradas	←	←	←	←	←	←	←	←	←	←	4,0 V
	entr. I/O 6-7	3,6	3,6	3,4	3,2	3,4	3,6	3,2	3,6	3,6	3,6	V
I_{IL}	entradas	←	←	←	←	←	←	←	←	←	←	< 1mA
	entr. I/O	←	←	←	80	20	←	20	←	←	←	< 1mA
I_{IH}	todas entr.	←	←	←	←	←	←	←	←	←	←	< 10nA
V_{OL} max	I_{OL} = -16 mA	0,74	0,72	0,60	0,58	0,59	0,72	0,57	0,71	0,73	0,71	V
V_{OH} min	I_{OH} = 16 mA	4,0	4,0	4,1	4,2	4,1	4,0	4,2	4,0	3,9	4,0	V
I_{CURTO} max	curto - circuito p/ V_{DD} ou GND	38	38				38	45	38	38	39	mA
f_{CK} max	funcional (PRBSG)	93										MHz

Tabela IV - RESULTADOS DOS TESTES PARAMÉTRICOS AC

Proj - CT/P-A Data - Condições - V_{DD} = 5,0V T_{AMB} = 24 ± 3°C

Param.	Condição.	1	2	3	4	5	6	7	8	9	10	Un.
t_r MAX	$C_L \simeq 20pF$	6			6							ns
	$C_L \simeq 48pF$	10			9							ns
	$C_L \simeq 96pF$	16			14							ns
t_f MAX	$C_L \simeq 20pF$	7			6							ns
	$C_L \simeq 48pF$	10			10							ns
	$C_L \simeq 96pF$	16			15							ns
t_{dCK} MAX	$C_L \simeq 20pF$											
	I/O_{0-3} L→H	28			26							ns
	I/O_{4-7} L→H	24			22							ns
	I/O_{0-3} H→L	24			22							ns
	I/O_{4-7} H→L	20			17							ns
t_{dS} MAX	$C_L \simeq 20pF$	25			22							ns
t_{dR} MAX	$C_L \simeq 20pF$	25			22							ns
t_{dCL} MAX	$C_L \simeq 20pF$	20			20							ns
t_{dCIO}	$C_L \simeq 20pF$	10			10							ns
	I/O z→L											
$t_{dA_{0-2}}$ MAX	$C_L = 20pF$	30			27							ns
	I/O (5mA)											

4.6.3. Ensaios de ESD

Para validar os dispositivos de proteção de entrada, foram aplicados pulsos de descarga eletrostática segundo o método 3015.2 da MIL STD 883C (fig. 39). Foram determinados alguns pinos para se verificar a sensibilidade a ESD, medindo-se a integridade dos pinos após os pulsos através de testes de continuidade e fuga (testes paramétricos).

O procedimento e as condições de teste são especificados na MIL STD. Entretanto, foram aplicados pulsos de apenas 500V (e não 2000V segundo a norma), por limitação do equipamento, e foram exercitados os quatro pares de pinos da tabela V abaixo, para as mesmas 2 peças, pelo pequeno número de protótipos disponíveis.

TABELA V - PARES DE PINOS PARA ENSAIO DE ESD

1	entrada SI (pino 7) e GND (pino 12)
2	saída SO (pino 11) e GND (pino 12)
3	entrada B1 (pino 6) e saída O0 (pino 8)
4	VDD (pino 24) e GND (pino 12)
5	I/O7 (pino 13) e GND (pino 12)

Resultados

Foram ensaiadas e medidas 2 peças, não se detetando falhas por ESD. Portanto, o circuito é protegido contra ESD até 500V, pelo menos.

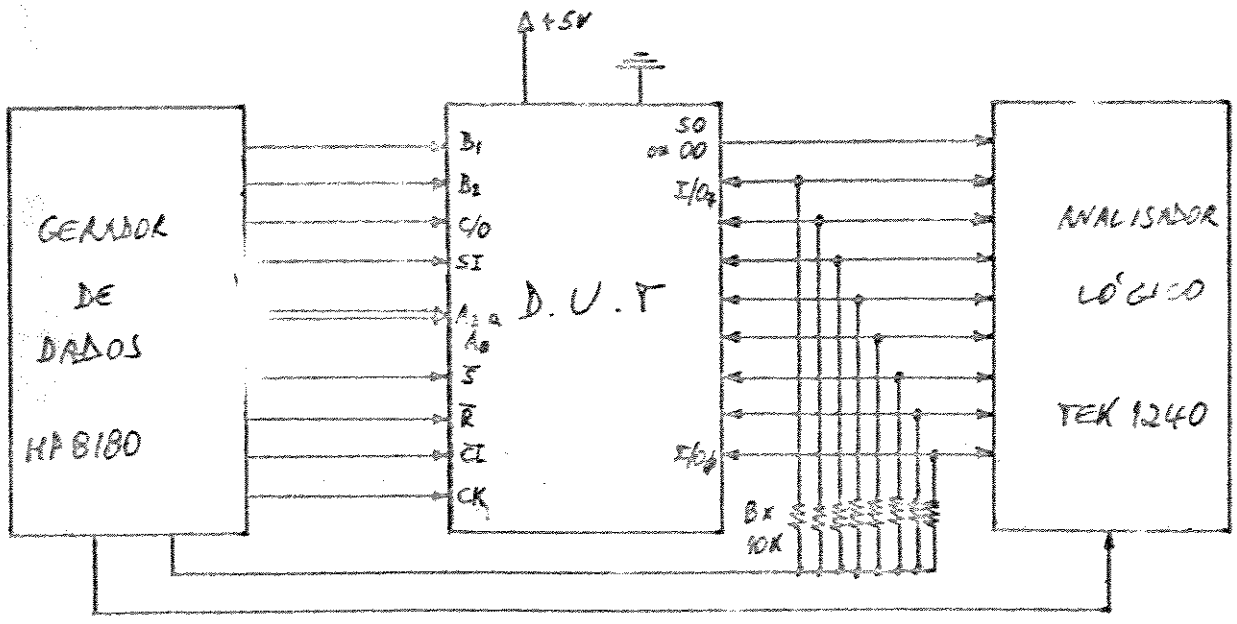


Fig.38 - Esquema para o teste funcional do CTIP

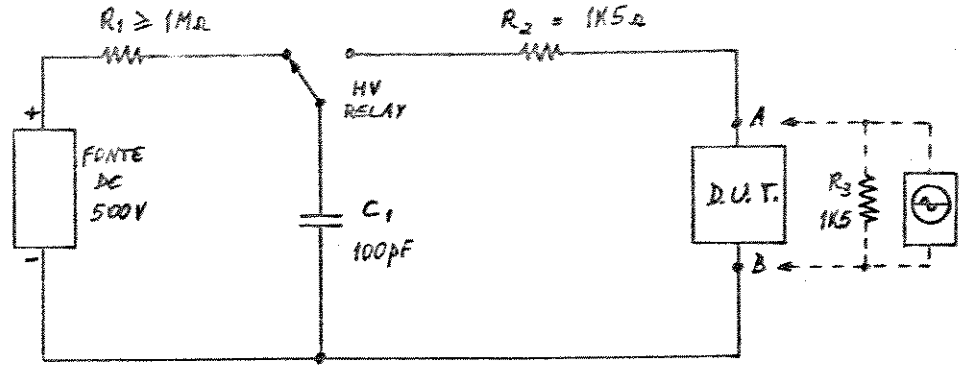


Fig.39 - Esquema para ensaio de ESD

5. CONCLUSÃO

5.1. Exemplo de Aplicação

O CTIP foi aplicado em uma placa eletrônica desenvolvida pelo Instituto de Automação do CTI, que é uma interface para os codificadores angulares de robôs. A placa contém uma lógica de seleção de endereços e três circuitos codificadores (ver fig. 42), com um total de 25 chips TTL, SSI e MSI.

Para implementar seu auto-teste foram acrescentados (fig. 43): um chip CTIP como gerador de sequências pseudo-aleatórias; um segundo chip CTIP como registrador de assinaturas; um CI buffer-tristate TTL (74LS365); um CI inversor TTL (74LS04); uma nand TTL de 13 entradas (74LS133) para comparar a assinatura final e acionar um LED de diagnóstico; resistores de pull--up/down.

Os sinais de controle do teste são gerados por um oscilador e um divisor de frequência externos à placa, montados junto com uma fonte de alimentação +5V. São aplicados na placa um clock rápido (CLK) para o circuito funcional, um clock lento (TCK) para os módulos de teste e um sinal C/O que controla o ciclo de teste (ver fig. 44).

No semi-ciclo em que C/O está alto, o PRBSG gera dados para estimular o circuito da interface (endereços, IORQ e fases de entrada) e o MISR compacta os dados colocados no barramento pelos drivers tri-state da placa, gerando uma assinatura. No início desse semi-ciclo, é gerado um pulso estreito CLRb, por diferenciação de C/O, o qual inicializa a interface gerando os sinais de PRESET para os contadores UP/DOWN, zera o MISR através de CLb e inicializa o PRBSG com 1 no primeiro bit (o PRBSG é zerado no semi-ciclo em que C/O é baixo). Assim, em todo ciclo de teste o circuito começa do mesmo estado, devendo gerar a mesma assinatura na ausência de falhas.

No semi-ciclo em que C/O está baixo, os buffers-tristate conectados ao PRBSG são desligados, o PRBSG é zerado através de CLb e o MISR é ativado para colocar a assinatura (por ele registrada no semi-ciclo anterior) no barramento. Essa assinatura, que é 01111111 para uma placa boa, é devidamente invertida e aplicada na porta NAND, que é ativada por C/Ob para verificar se a assinatura está correta, acendendo-se então o LED verde (para indicar "OK").

Foram simuladas fisicamente diversas falhas na placa, como linhas em curto, pinos em aberto e atrasos por cargas altamente capacitivas, verificando-se que o teste é capaz de detetá-las (LED apagado). Não foi feito um levantamento da cobertura de falhas (ou uma simulação de falhas em computador), porque a idéia é apenas demonstrar a aplicação do CTIP em uma placa.

Assim, fica demonstrada a simplicidade de implementação de um circuito auto-testável em uma placa eletrônica, com o CTIP.

5.2. Conclusão

O projeto do CTIP, resultou em um CI com possibilidade de aplicação comercial, pois facilita o teste de placas digitais. A inclusão de vários módulos do CTIP em uma placa, aumenta sua testabilidade, sem alterações fundamentais de todo o circuito lógico para reconfigurá-lo durante o modo teste.

O CTIP ocupou uma área de 2,0 mm x 2,5 mm, contendo 8 flip-flops. Possui 24 pinos (Fig. 41) e opera a 5V, sendo compatível com TTL e CMOS. Foi implementado na área do CTI no PMU (CHIPCTI) junto com mais quatro circuitos, sendo a pastilha final, de 4,2 mm x 5,2 mm, colocada em um encapsulamento cerâmico tipo CERDIP-24 (ver fig. 40).

Apesar dos dois erros cometidos no projeto gráfico, que impossibilitam seu funcionamento como Scan-Shift-Register e a observação endereçada das linhas de entrada, o CTIP pode ser empregado em auto-teste de placas, atuando como gerador de estímulos pseudo-aleatórios e registrador de assinaturas. Para eliminar esses erros, o circuito poderia ser reprocessado na "foundry" com duas pequenas correções no lay-out final. Os valores paramétricos obtidos estão numa faixa aceitável, viabilizando seu uso em placas com CI's TTL e CMOS.

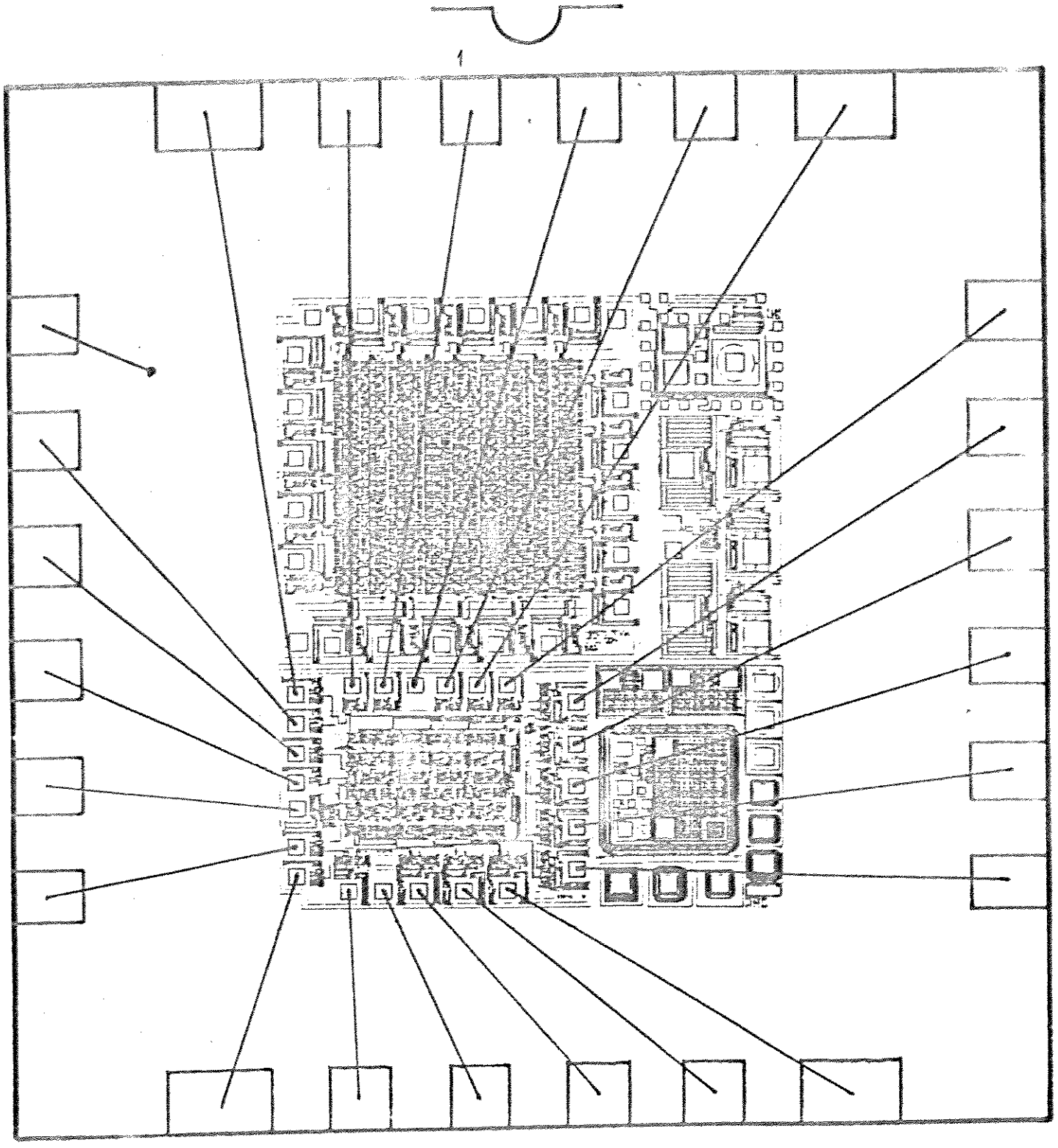


Fig.40 - Diagrama de solda do CTIP

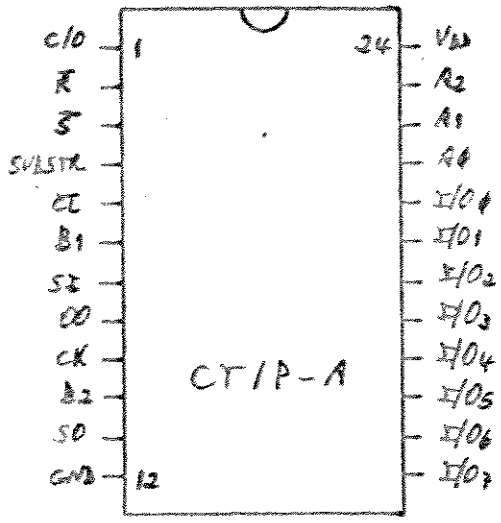


Fig.41 - Diagrama de pinos do CTIP

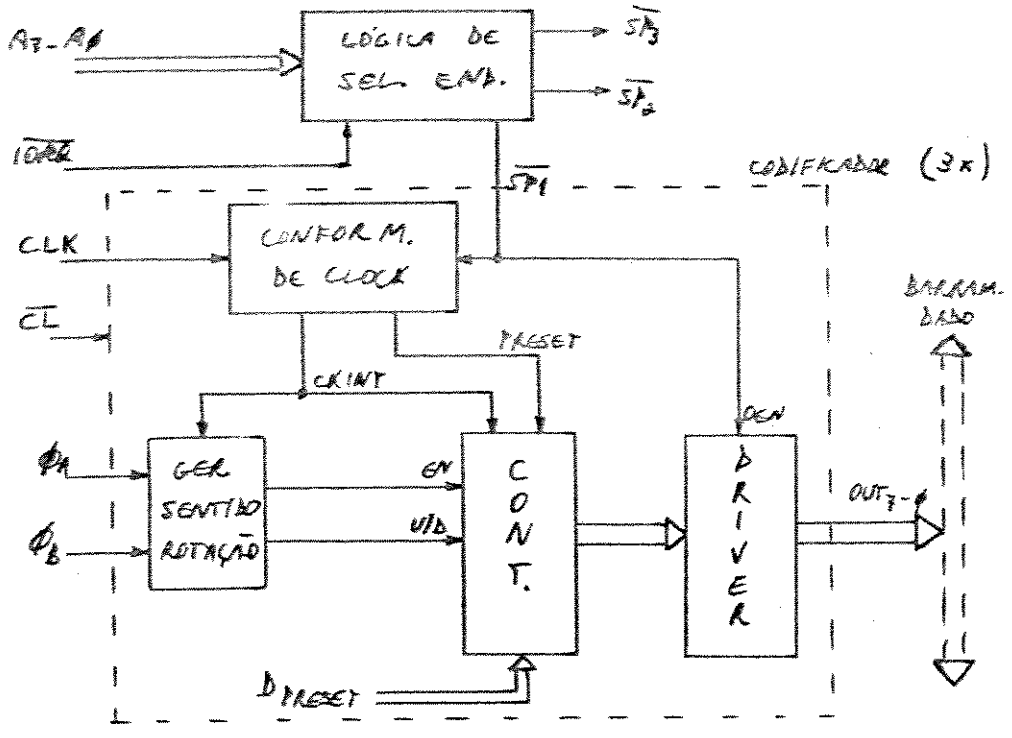


Fig.42 - Diagrama de blocos da placa de Interface para Codificador Angular

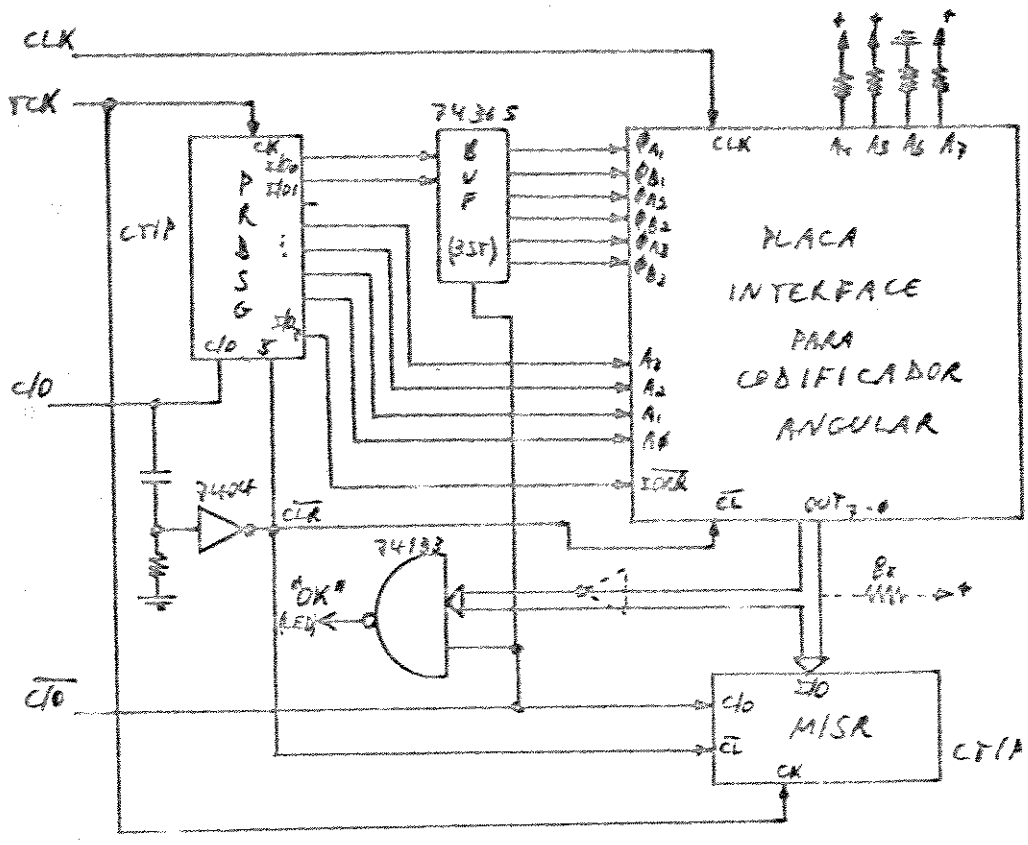


Fig.43 - Esquema de Auto-teste implementado na placa

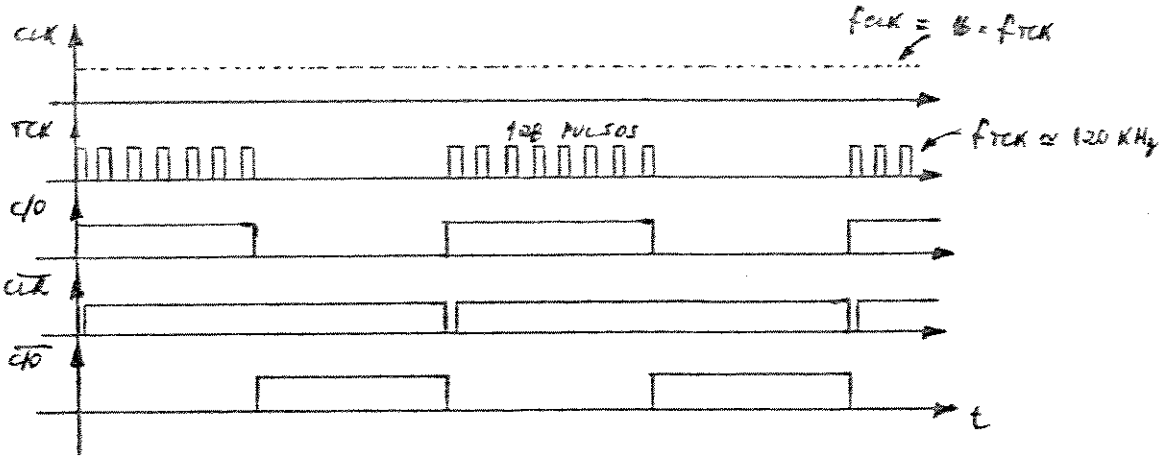


Fig.44 - Forma dos sinais de teste aplicados

6. BIBLIOGRAFIA

[Abra83] Abraham, Jacob A. : "Design for Testability", IEEE Custom Integrated Circuits Conference, Rochester , N.Y., 1983.

[Abra84] Abramovici, M., Menon, P.R. and Miller, D.T.: "Critical Path Tracing: An Alternative to Fault Simulation", IEEE Design & Test, febr. 1984.

[Agra82] Agrawal, V.D. and Mercer, M.R.: "Testability Measures - What do they tell us?", Proc. IEEE Test Conference, 1982.

[Ban84] Banerjee, P. and Abraham, J.A.: "Characterization and Testing of Physical failures in MOS Logic Circuits", IEEE Design & Test, aug. 1984.

[Barz81] Barzilai, Z. et al: "The Weighted Syndrome Sums Approach to VLSI Testing", IEEE Trans. on Comp., vol. C-30, dec. 1981.

[Been87] Beenker, F.P.M.: "Systematic and Structured Methods for Digital Board Testing", VLSI Systems Design, January 1987.

[Berk85] SPICE 2G Users' Guide, University of California at Bekerley, 1985.

[Ber85a] K1CGUIDE, University of California at Berkerley, 1985.

[Boyd85] Boyd, Stuart and Faucher, Marc: "Latch-up Characterization Using Specially Designed Test Structures", Proc. IEEE CICC, 1985.

[Breu72] Breuer, Melvin A. : Design Automation of Digital Systems, chap. 7, Prentice-Hall Inc., New Jersey, 1972.

[Breu76] Breuer, M.A. and Friedman, A.: Diagnosis & Reliable Design of Digital Systems, Comp. Science Press, 1976.

[Cart82] Carter, W.C.:" The Ubiquitous Parity Bit", in Proc. 12th Annual Intl. Symposium on Fault-Tolerant Computing, Santa Monica, CA, june 1982.

[Cha78] Cha, C.W., Donath, W.E. and Ozguner, F.: "9V Algorithm for Test Pattern Generation of Combinational Digital circuits", IEEE Trans. on Comp., vol. C-27, nº 3, march 1978.

[Cha78a] Cha, Charles W.: "A Testing Strategy for PLAs", Proc. 15th Design Autom. Conf., Las Vegas, 1978.

[Chan82] Chandramouli, R.: "Designing VLSI Chips for Testability", Electronics Test, november, 1982.

[Chan85] Chandramouli, R. and Sucar, H.: "Defect analysis and Fault Modeling in MOS Technology", Proc. Intl. Test Conference, 1985.

[Clar85] Clarke, R.: "Confusion Surrounds Simulator Specifications", Electronics Test, aug. 1985.

[CTI85] SIMUL : Manual do Usuário, CTI, 1985.

[CTI86] Manual do Projeto Multiusuário, CTI, Março 1986.

[Eich77] Eichelberger, E. B. and Williams, T. W.: "A Logic Design Structure for LSI Testability", Proc. 14th Design Aut. Conf., june, 1977.

[Els85] Elster, J.: "Directory of Automatic Test Equipment", VLSI Systems Design, oct. 1985.

[El-Z83] El-Ziq, Y.M. and Butt, H.H.: "A Mixed-Mode Built-In Self-Test Technique Using Scan-Path and Signature Analysis", Proc. Intl. Test Conference, 1983.

[El-83a] El-Ziq, Y.M.: "S-Cubed: VLSI Self-Test Using Signature Analysis and Scan-Path Techniques", IEEE, 1983.

[Froh77] Frohwerk, Robert A. : "Signature Analysis : A New Digital Field Service Method", HP Journal, May 1977.

[Fuji83] Fujiwara, H. and Shimono, T.: "On the Acceleration of Test Generation Algorithms", IEEE Trans. on Comp., vol. C-22, nº 12, dec. 1983.

[Fuji84] Fujiwara, Hideo: "A New PLA Design for Universal Testability", IEEE Trans. on Comp., vol. C-33, nº 8, august, 1984.

[Gali80] Galiay, J., Crouzet, Y. and Verginault, M.: "Physical versus Logical Fault Models for MOS LSI Circuits: Impact on Their Testability", IEEE Trans. on Comp., vol.C-29, june 1980.

[Gels87] Gelsinger, P.P.: "Design and Test of the 80386", IEEE Design & Test, june 1987.

[Gio86] Giorgio, Rosana C.: Relatórios Técnicos da Biblioteca de Células-Padrão, IM/CTI, docs. nos. 022/86, 025/86.

[Glas79] Glaser, A. and Subak-Sharpe, G.: Integrated Circuits Engineering, chap. 11, Addison-Wesley Publ. Co., 1979.

[Goel80] Goel, P.: "Test Generation Costs, Analysis and Projections", Proc. 17th Design Automation Conference, June, 1980.

[Goel81] Goel, P.: "An implicit enumeration algorithm to generate tests for combinational logic circuits", IEEE Trans. on Comp., vol. C-30, march 1981.

[Goel81a] Goel, P. and Rosales, B.C.: "PODEM-X : An Automatic Test Generation System for VLSI Logic Structures", Proc. 18th Design Automation Conference, 1981.

[Gras81] Grason, J. and Nagle, A.W.: "Digital Test Generation and Design for Testability", Journal of Digital Systems, vol. V, nº 4, 1981.

[Herm85] Herman, Linda: "Controlling CMOS Latch-up", VLSI Design, april 1985.

[Hodg83] Hodges, D. A. and Jackson, H.G.: Analysis and Design of Digital Integrated Circuits, chap. 3, McGraw-Hill Inc., 1983.

[Hump84] Humphry, S.E.: "VLSI Systems Claims to Test 512-pin Devices at Up to 100MHz", Electronics Test, july 1984.

[Indr86] Indrajo, G. and Wang, L.T.: "Fault Simulation Rates Effectiveness of Test Patterns", EDN, sept. 1986.

[Jain85] Jain, S.K. and Agrawal, V.D.: "Statistical Fault Analysis", IEEE Design & Test, feb. 1985.

[Kino84] Kinoshita, K. and Saluja, K. K.: "Built-In Testing of Memory Using On-Chip Compact Testing Scheme", Proc. Intl. Test Conference, 1984.

[Komo83] Komonytsky, Donald : "Synthesis of Techniques Creates Complete System Self-Test", Electronics, March 10, 1983.

[Kone80] Konemann, Bernd et al : "Built-In Test for Complex Digital Integrated Circuits", IEEE JSSC, vol SC-15, nº 3, June 1980.

[Kone83] Konemann, B. and Zwiehoff, G.: "Built-In Testing: State-of-the-Art", IEEE, 1983.

[Kub83] Kuban, J. and Bruce, B.: "The MC6804P2 Built-In Self-Test", Proc. Intl. Test Conference, 1983.

[Law83] Lawrence Jr., J. D.: "Parallel Testing of Memory Devices", Test & Measurement World, october 1983.

[Lin75] Lin, Hung C. and Linholm, L.W. : "An Optimized Output Stage for MOS Integrated Circuits", IEEE JSSC, vol. SC-10, nº 2, April 1975.

[Log] SMT Testability Chip Set Description and Specifications, data sheet 8305, Logical Solutions Technology Inc.

[Mall85] Mallella, S. and Wu, S.: "A Sequential Circuit Test Generation System", Proc. Intl. Test Conference, 1985.

[Marl78] Marlett, R.A.: "EBT: A Comprehensive Test Generation Technique for Highly Sequential Circuits", Proc. 15th Annual Design Automation Conference, june 1978.

[Mari71] Marinos, P.N.: "Derivation of Minimal Complete Sets of Test-Input Sequences Using Boolean Differences", IEEE Trans. on Comp., vol. C-20, nº 1, jan. 1971.

[Maun85] Maunder, Colin : "Built-In Test - A Review", Electronics & Power, March 1985.

[Maun86] Maunder, Colin: "The Joint Test Action Group", Computer - Aided Engineering Journal, August 1986.

[Max83] Maxwell, P. C.: "Testing Integrated Circuits", Journal of Electrical and Electronics Engineering, Australia, vol. 3, nº 4, december 1983.

[McCl81] McCluskey, E.J. and Bozorgui-Nesbat, S.: "Design for Autonomous Test", IEEE Trans. on Comp., vol. C-30, nº 11, nov. 1981.

[McCl82] McCluskey, E.J.: "Built-In Verification Test", in Proc. 1982 IEEE Test Conf., Philadelphia, PA, nov. 1982.

[McCl84] McCluskey, E.J.: "Verification Testing - A Pseudo-Exhaustive Test Technique", IEEE Transactions on Computers, vol. C-33, nº 6, june 1984.

[Mc84a] McClusKey, E. J. : "A Survey of Design for Testability Scan Techniques", VLSI Design, December 1984.

[McCl85] McClusKey, Edward J. : "Built-In Self-Test Techniques", IEEE Design & Test, April 1985.

[Mc85a] McClusKey, Edward J. : "Built-In Self-Test Structures", IEEE Design & Test, April 1985.

[Mend86] Mendelsohn, A.: "Self-Testing IC's Begin to Emerge - Tentatively", Electronics, feb.24, 1986.

[Mil883] Military Standard Test Methods and Procedures for Microelectronics - Mil-Std-883C, FSC 5962, Depart. of Defense, USA, august 25, 1983.

[Morr86] Morris, D.S.: "In-Circuit, Functional or Emulation - Choosing the right test solution", CAD Journal, june 1986.

[Oliv86] Oliveira, Arthur H., Taveira, J.G.M. e Orives, R.A. : "Técnicas Avançadas para Testabilidade de CIs", I SSBM, Campinas, SP, 1986.

[Oliv87] Oliveira, Arthur H.C. e Akira, L.A.: "CTIP - Circuito para Teste Integrado de Placas", Anais II CSBM, São Paulo, 1987.

[Oli87a] Oliveira, Arthur H. C. e Orives, R. A.: "Metodologia para Testabilidade de Sistemas", Anais do II Simpósio de Sistemas de Computadores Tolerantes a Falhas, Campinas, 1987.

[Ori85] Orives, Ramon A. et al: "Projeto para Testabilidade de CI's - Uma Aplicação Prática", II SBCCI, Porto Alegre, RS, 1985.

[Powe83] Powell, T.: "Software gauges the testability of computer designed IC's", Electronic Design, nov. 24, 1983.

[Rob87] Robinson, J. P. and Saxena, N. R.: " A Unified View of Test Compression Methods", IEEE Trans. on Comp., vol. C-36, nº 1, january, 1987.

[Roth66] Roth, J.P.: "Diagnosis of Automata Failures: A Calculus and a Method", IBM Journal of R.& D., vol. 10, july 1966.

[Roth67] Roth, J.P., Bouricius, W.G. and Schneider, P.R.: "Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits", IEEE Trans. on Electr. Comp., vol EC-16, nº 5, oct. 1967.

[Salu87] Saluja, K. K., Sng, H. S. and Kinoshita, K.: "Built-In Self-Testing RAM: A Practical Alternative", IEEE Design & Test, february 1987.

[Sav84] Savir, J., Ditlow, G.S. and Bardell, P.H.: "Random Pattern Testability", IEEE Trans. on Comp., vol C-33, nº 1, jan. 1984.

[Seg82] Segers, M.T.M. René : "The Impact of Testing on VLSI Design Methods", IEEE, vol. SC-17, nº 3, june 1982.

[Shed77] Shedletsky, J.J.: "Random testing : Practicality versus verified Effectiveness", Proc. 7th Intl. Conf. on Fault-Tolerant Computers, june 1977.

[Swan83] Swan, R. and McMinn, C.: "General-purpose Testers puts a separate set of resources behind each VLSI-device Pin", Electronics, sept. 8, 1983.

[Tang83] Tang, D.T. and Woo, L.S.: "Exhaustive Test Pattern Generation with Constant-Weight Vectors", IEEE Trans. on Comp., vol. c-32, december 1983.

[Tang84] Tang, D.T. and Chen, C.L.: "Iterative Exhaustive Pattern Generation for Logic Testing", IBM Journal of Res. and Dev., vol. 28, nº 2, march 1984.

[Tsui85] Tsui, Frank F. : "The Cost and Speed Barriers in LSI/VLSI Testing - Can they be overcome by Testability Design?", Proc. IEEE Intl. Test Conference, 1985.

[Tul83] Tuloss, Rodham E.: "Automated board testing: coping with complex circuits", IEEE Spectrum, july 1983.

[Turi85] Turino, J.: "Coming Major Changes in Test", Semiconductor International, jan. 1985.

[Wag87] Wagner, K.D., Chin, C.K. and McCluskey, E.J.: "Pseudorandom Testing", IEEE Trans. on Comp., vol C-36, nº 3, march 1987.

[Will83] Williams, Thomas W. and Parker, Keneth P. : "Design for Testability - A Survey", Proceedings of the IEEE, vol. 71, nº 1, January 1983.

[Will84] Williams, Thomas W.: "VLSI Testing", Computer, october, 1984.

APÊNDICE I

Arquivos de descrição dos circuitos e resultados da simulação lógica do CTIP no SIMUL:

- Célula MACRO.CTIP

- Célula CIRC.CTIP

Os arquivos contém, pela ordem, a descrição dos circuitos em portas lógicas básicas (elemento, entradas, saídas), estímulos de entrada (nome do sinal, estado inicial, instantes de transição) e sinais a serem plotados no tempo. O SIMUL é um simulador lógico de atraso unitário [CTI85].

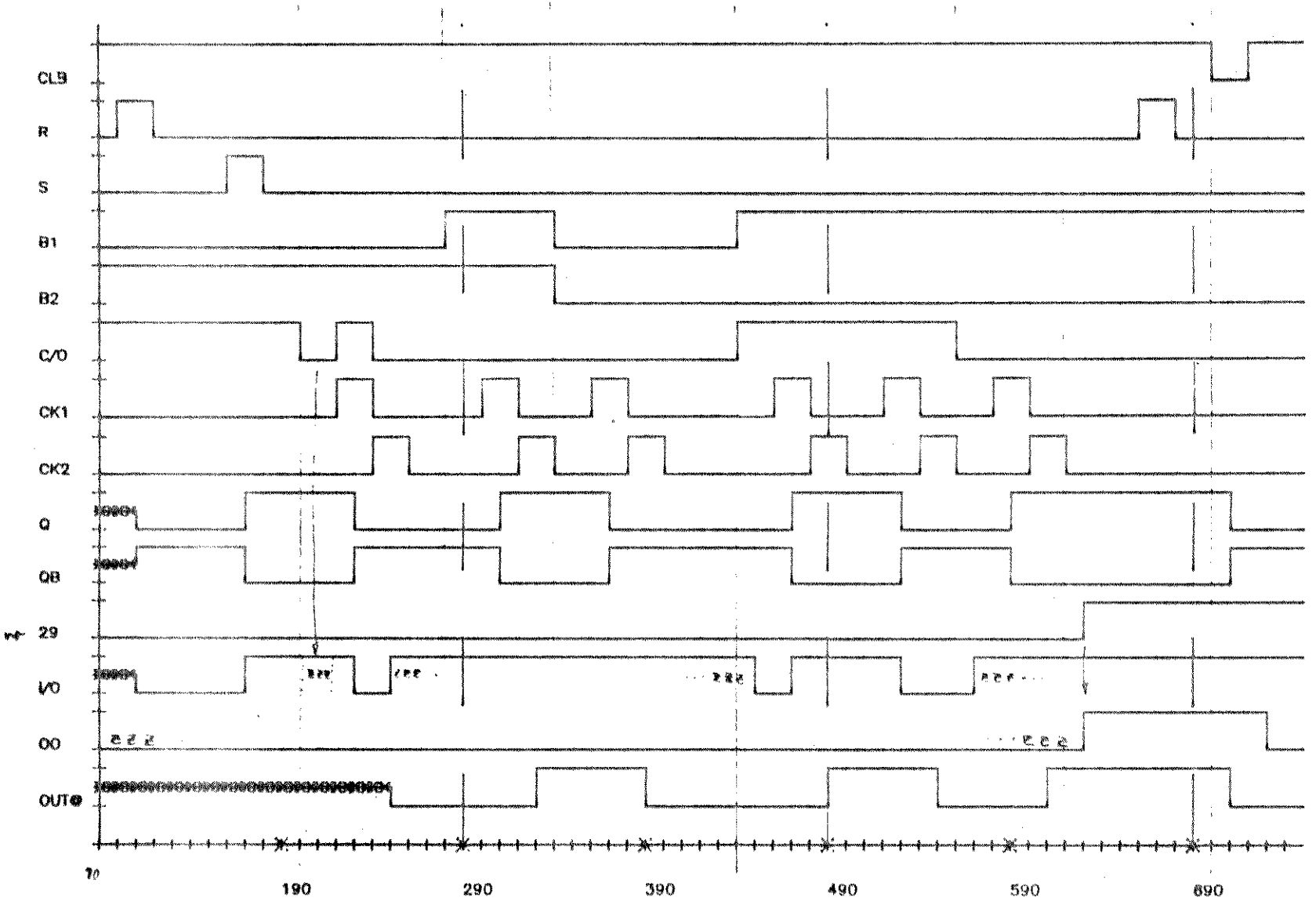
CÉLULA MACRO.CIR (ver fig. 29)

```
* e l e
nand IN 0 0 0 0 16
nor 16 B2 0 0 0 17
nand C/O 0 0 0 0 31
nand B1 31 0 0 0 18
nand I-O 0 0 0 0 19
nor 18 19 0 0 0 20
xor 17 20 0 0 0 21
nand A0 0 0 0 0 24
nor 24 A1 A2 0 0 EN
nand S C/O EN 0 0 26
nand R C/O EN 0 0 27
dff 21 0 CK1 32 26 12 28
nor C/O 33 0 0 0 29
nand I-O 29 0 0 0 34
nand OB C/O 0 0 0 I/O
dff 0 0 CK2 CLB 0 14
nand CLB 27 0 0 0 23
nand 23 0 0 0 0 32
nand 12 0 0 0 0 39
nand 39 0 0 0 0 40
nand 40 0 0 0 0 41
nand 41 0 0 0 0 0
nand 28 0 0 0 0 43
nand 43 0 0 0 0 44
nand 44 0 0 0 0 45
nand 45 0 0 0 0 OB
nand 14 0 0 0 0 46
nand 46 0 0 0 0 OUT
nand EN 0 0 0 0 33
nand 34 0 0 0 0 OO
end
ext
a S 0 0 160 180
a A2 0 0
a A1 0 0
a A0 0 0 80 200 620
a R 0 0 100 120 660 680
a B1 0 0 280 340 440
a B2 1 0 340
a C/O 1 0 200 220 240 440 560
a IN 0 0 440 500 560 620
a CK1 0 0 220 240 300 320 360 380
a CK1 1 460 480 520 540 580 600
a CK2 0 0 240 260 320 340 380 400
```

```
a CK2 1 480 500 540 560 600 620
a I-O 0 0 280 340 620 720
a CLB 1 0 700 720
end
out
OUT OO 1 0 29 OB 0
CK2 CK1 C/O B2 B1 S R CLB
end
```

MACRO - CTIP (OK)

Tue Nov 25 08:01:07 1986



Incremento de tempo : 10

tempo (ns)

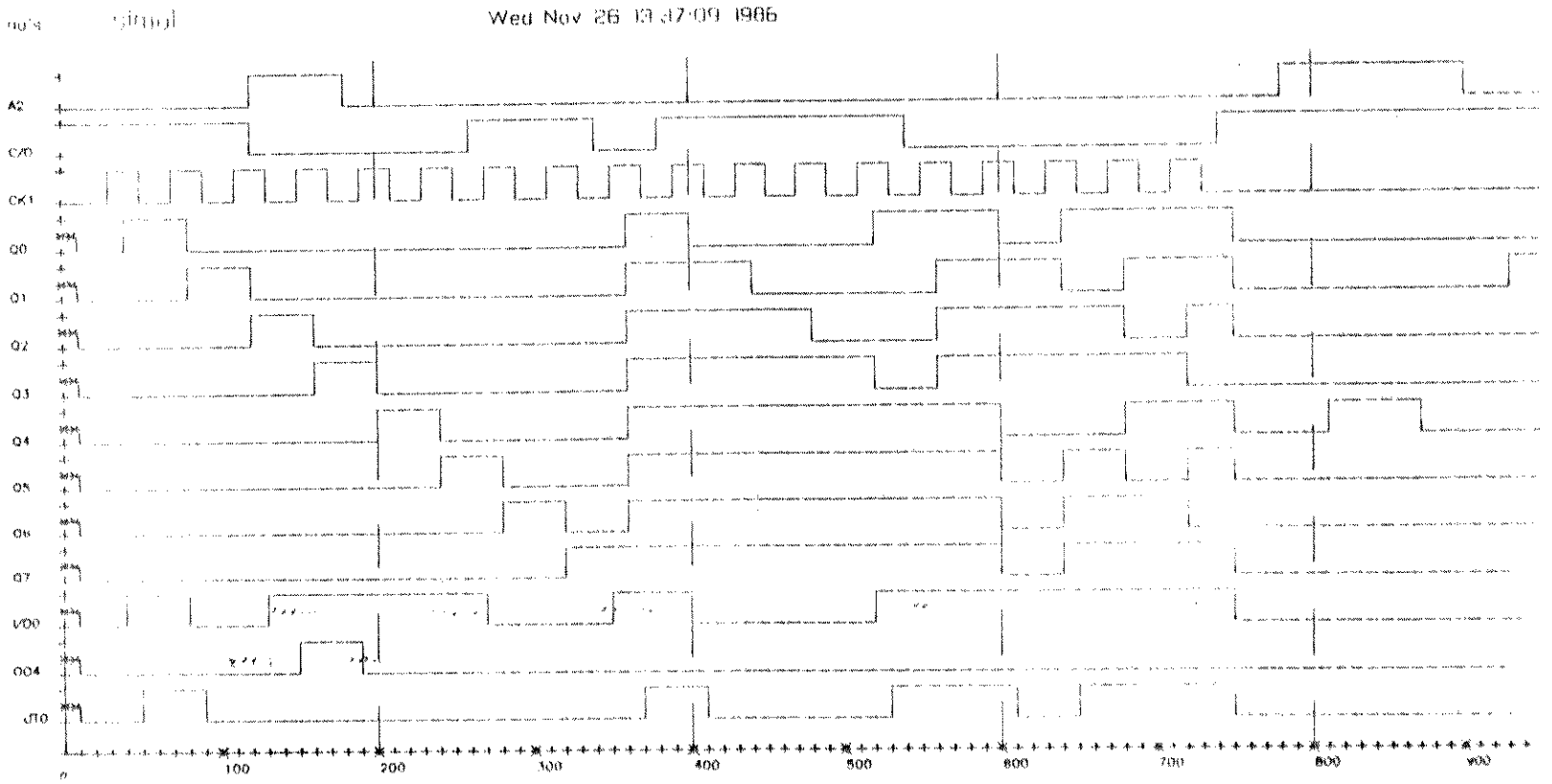
célula cirg.cnt (ver fig. 30)

```
macr mffp
mffp IN C/O B1 B2 CLB Q
* S R I-0 CK1 CK2 I/O
* A0 A1 A2 0 0 00
* 0 0 0 0 0 OUT
nand IN 0 0 0 0 16
nor 16 B2 0 0 0 17
nand C/O 0 0 0 0 31
nand B1 31 0 0 0 18
nand I-0 0 0 0 0 19
nor 18 19 0 0 0 20
xor 17 20 0 0 0 21
nor A0 A1 A2 0 0 EN
nand S C/O EN 0 0 26
nand R C/O EN 0 0 27
dff 21 0 CK1 32 26 12 28
nor C/O 33 0 0 0 29
nand I-0 29 0 0 0 34
nand QB C/O 0 0 0 I/O
dff Q 0 CK2 CLB 0 14
nand CLB 27 0 0 0 23
nand 23 0 0 0 0 32
nand 12 0 0 0 0 39
nand 39 0 0 0 0 40
nand 40 0 0 0 0 41
nand 41 0 0 0 0 0
nand 28 0 0 0 0 43
nand 43 0 0 0 0 44
nand 44 0 0 0 0 45
nand 45 0 0 0 0 QB
nand 14 0 0 0 0 46
nand 46 0 0 0 0 OUT
nand EN 0 0 0 0 33
nand 34 0 0 0 0 00
end


ele
nand SIN 100 0 0 0 101
nand B1 0 0 0 0 100
nand B1 102 0 0 0 103
nand 101 103 0 0 0 IN0
xor Q2 Q4 Q6 Q7 0 102
mffp IN0 C/O B1 B2 CLB Q0
* S R I-00 CK1 CK2 I/O0
* A0 A1 A2 0 0 000
* 0 0 0 0 0 OUT0
mffp OUT0 C/O B1 B2 CLB Q1
* S R I-01 CK1 CK2 I/O1
* A0B A1 A2 0 0 001
* 0 0 0 0 0 OUT1
mffp OUT1 C/O B1 B2 CLB Q2
* S R I-02 CK1 CK2 I/O2
* A0 A1B A2 0 0 002
* 0 0 0 0 0 OUT2
mffp OUT2 C/O B1 B2 CLB Q3
* S R I-03 CK1 CK2 I/O3
* A0B A1B A2 0 0 003
* 0 0 0 0 0 OUT3
```

cont.

```
mffp OUT3 C/O B1 B2 CLB Q4
* S R I-04 CK1 CK2 I/Q4
* A0 A1 A2B 0 0 004
* 0 0 0 0 0 OUT4
mffp OUT4 C/O B1 B2 CLB Q5
* S R I-05 CK1 CK2 I/Q5
* A0B A1 A2B 0 0 005
* 0 0 0 0 0 OUT5
mffp OUT5 C/O B1 B2 CLB Q6
* S R I-06 CK1 CK2 I/Q6
* A0 A1B A2B 0 0 006
* 0 0 0 0 0 OUT6
mffp OUT6 C/O B1 B2 CLB Q7
* S R I-07 CK1 CK2 I/Q7
* A0B A1B A2B 0 0 007
* 0 0 0 0 0 OUT7
nand A0 0 0 0 0 A0B
nand A1 0 0 0 0 A1B
nand A2 0 0 0 0 A2B
end

ext
a R 0 0 860 880
a S 0 0 800 820 920
a A2 0 0 120 180 780 900
a A1 0 0
a A0 0 0 900
a CLB 0 0 20 740 760
a B1 0 0 340 740
a B2 0 0 340 380 540 580
a C/O 1 0 120 260 340 380 540 740
a SIN 0 0 20 60 420 500
a CK1 0 0 30 50 70 90 110 130
a CK1 1 150 170 190 210 230 250 270
a CK1 0 290 310 330 350 370 390 410
a CK1 1 430 450 470 490 510 530 550
a CK1 0 570 590 610 630 650 670 690
a CK1 1 710 730
a CK2 0 0 40 60 80 100 120 140
a CK2 1 160 180 200 220 240 260 280
a CK2 0 300 320 340 360 380 400 420
a CK2 1 440 460 480 500 520 540 560
a CK2 0 580 600 620 640 660 680 700
a CK2 1 720 740
a I-00 0 0 140 240 340 380 540 590
a I-01 0 0 140 240 340 380 540 580
a I-02 0 0 140 240 340 380 540 580
a I-03 0 0 140 240 340 380 540 580
a I-04 0 0 140 240 340 380 540 660
a I-05 0 0 140 240 340 380 540 660
a I-06 0 0 140 240 340 380 540 660
a I-07 0 0 140 240 340 380 540 660
end
out
OUT7 OUT6 OUT5 OUT4 I/Q7 I/Q4 I/Q0 Q7 Q6 Q5
Q4 Q3 Q2 Q1 Q0 I-04 CK1 C/O CLB A2
end
```

CIRC. CTIP



Incremento de tempo : 10

APÊNDICE II

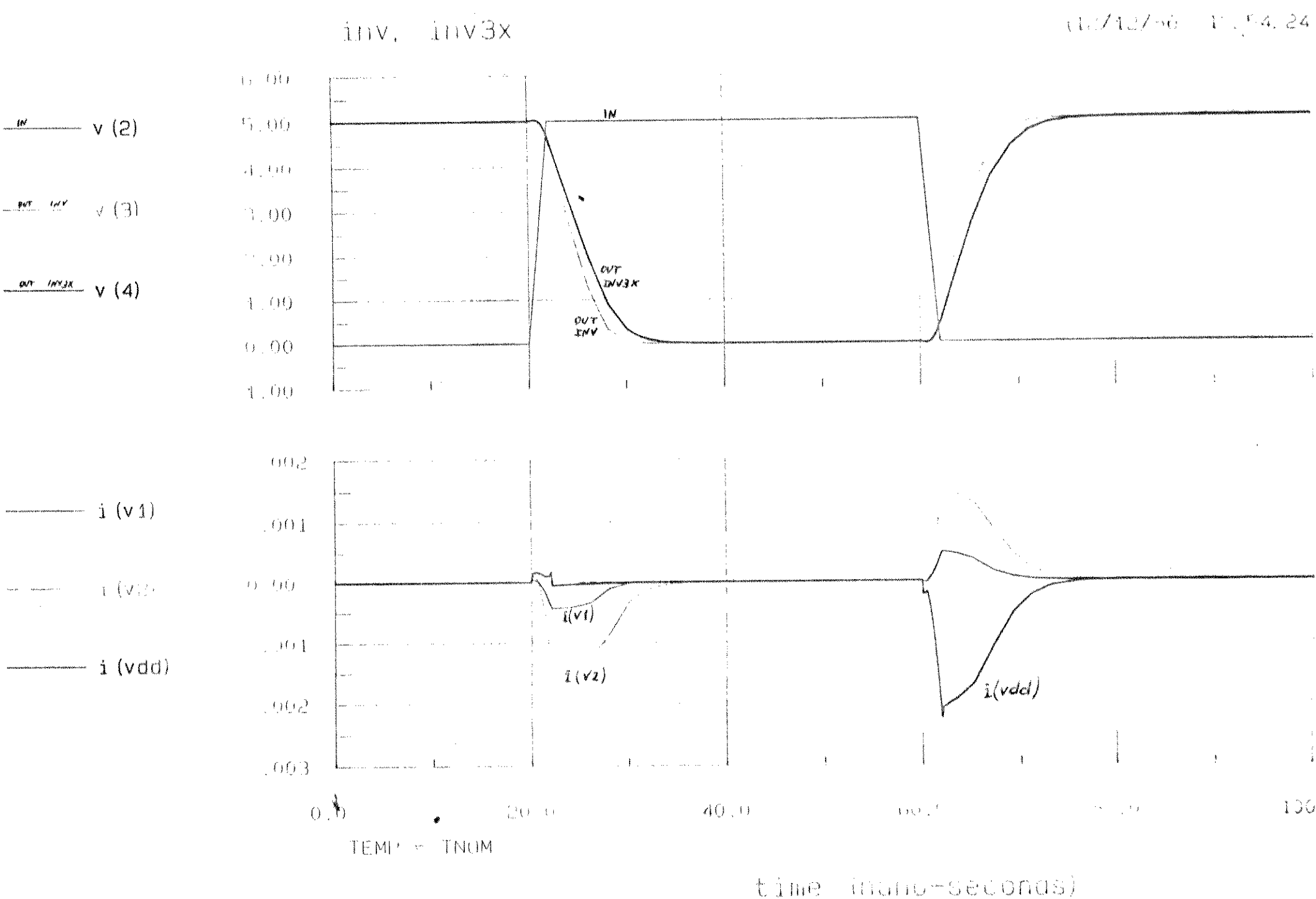
Arquivos de descrição das células e resultados das simulações elétricas do CTIP no SPICE 2G6:

- INV e INV3X para Wp = 12 um, Wn = 6 um e
  para Wp = 34 um e Wn = 16 um
- Interface TTL
- BUFINV
- BUF-TTL-I/O
- BUFNIN
- DR3ST
- MACROCIR

Os arquivos listados, contém pela ordem, os sinais de entrada e alimentação, a descrição dos circuitos com seus elementos, nós, dimensões e valores, os parâmetros dos modelos dos dispositivos e linhas de controle da simulação.

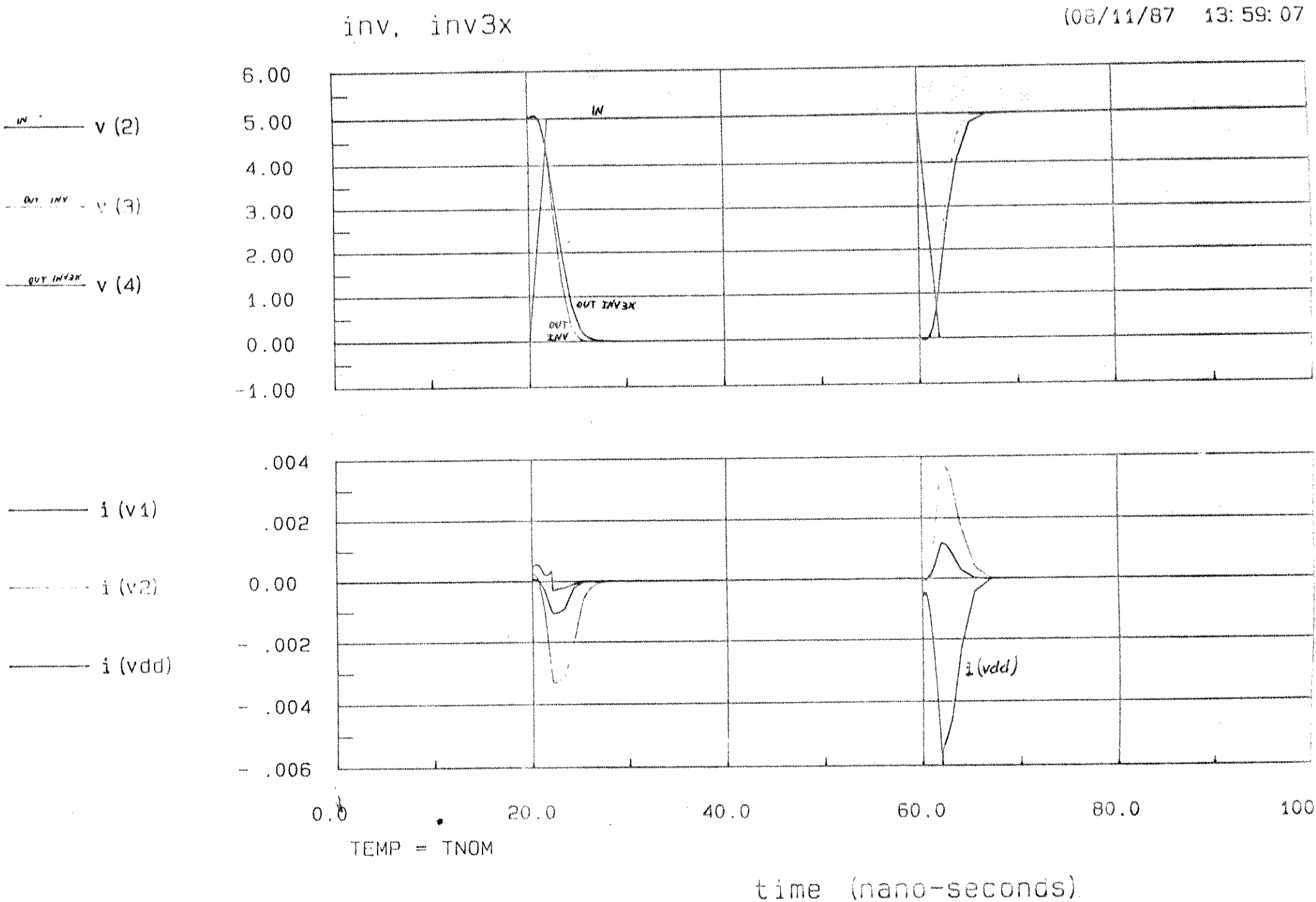
CÉLULAS INV e INV3X (com W, L mínimos)

```
inv, inv3x
Vdd 1 0 DC 5
V1 2 0 PULSE(0 5 20N 2N 2N 38N 100N)
.SUBCKT INV 1 2 3
MP1 3 2 1 1 MODP L=3U W=12U AD=90P AS=90P PD=39U PS=39U
+NRD=0.625 NRS=0.625
MN1 3 2 0 0 MODN L=3U W=6U AD=45P AS=45P PD=27U PS=27U
+NRD=1.25 NRS=1.25
.ENDS INV
.SUBCKT INV3X 1 2 3
MP1 3 2 1 1 MODP L=3U W=36U AD=270P AS=270P PD=87U PS=87U
+NRD=0.21 NRS=0.21
MN1 3 2 0 0 MODN L=3U W=18U AD=135P AS=135P PD=51U PS=51U
+NRD=0.42 NRS=0.42
.ENDS INV3X
*descricao do circuito
X10 1 2 3 INV
X11 1 2 4 INV3X
C1 5 0 0.5P
C2 6 0 2P
V1 3 5 DC 0
V2 4 6 DC 0
.NODESET V(3)=5 V(4)=5
.MODEL MODP PMOS LEVEL=2 VTO=-0.9 TOX=525E-10 NSUB=2.6E15
+XJ=0.10U LD=0.40U UO=250 UCRIT=0.85E5 UEXP=0.16 VMAX=2.75E4
+NEFF=5.0 DELTA=1.2 PSH=110 CGSO=4.4E-10 CGDO=4.4E-10
+CJ=160U CJSW=260P MJ=0.46 MJSW=0.23 PI=0.65
.MODEL MODN NMOS LEVEL=2 VTO=0.90 TOX=525E-10 NSUB=1.1E16
+XJ=0.02U LD=0.35U UO=665 UCRIT=0.93E5 UEXP=0.16 VMAX=4.74E4
+NEFF=3.3 DELTA=1.7 PSH=40 CGSO=3.7E-10 CGDO=3.7E-10
+CJ=315U CJSW=345P MJ=0.59 MJSW=0.275 PI=0.85
.TRAN 0.1N 100N
.GRAPH TRAN V(2) V(3) V(4) I(V1) I(V2) I(Vdd)
.PLOT TRAN V(2) V(3) V(4) I(V1) V(2) I(Vdd)
.OPTIONS LIMPTS=10000 PIVTOL=1.0E-30
+ABSTOL=1N VNTOL=1M
.END
```


inv, inv3x
IN
v (2)
OUT INV
v (3)
OUT INV3X
v (4)
5.00
4.00
3.00
2.00
1.00
0.00
-1.00
IN
OUT INV3X
OUT INV
i (v1)
i (v2)
i (vdd)
.002
.001
0.00
-.001
-.002
-.003
i(v1)
i(v2)
i(vdd)
0.0
20.0
40.0
60.0
80.0
100
TEMP = TNOM
time (nano-seconds)

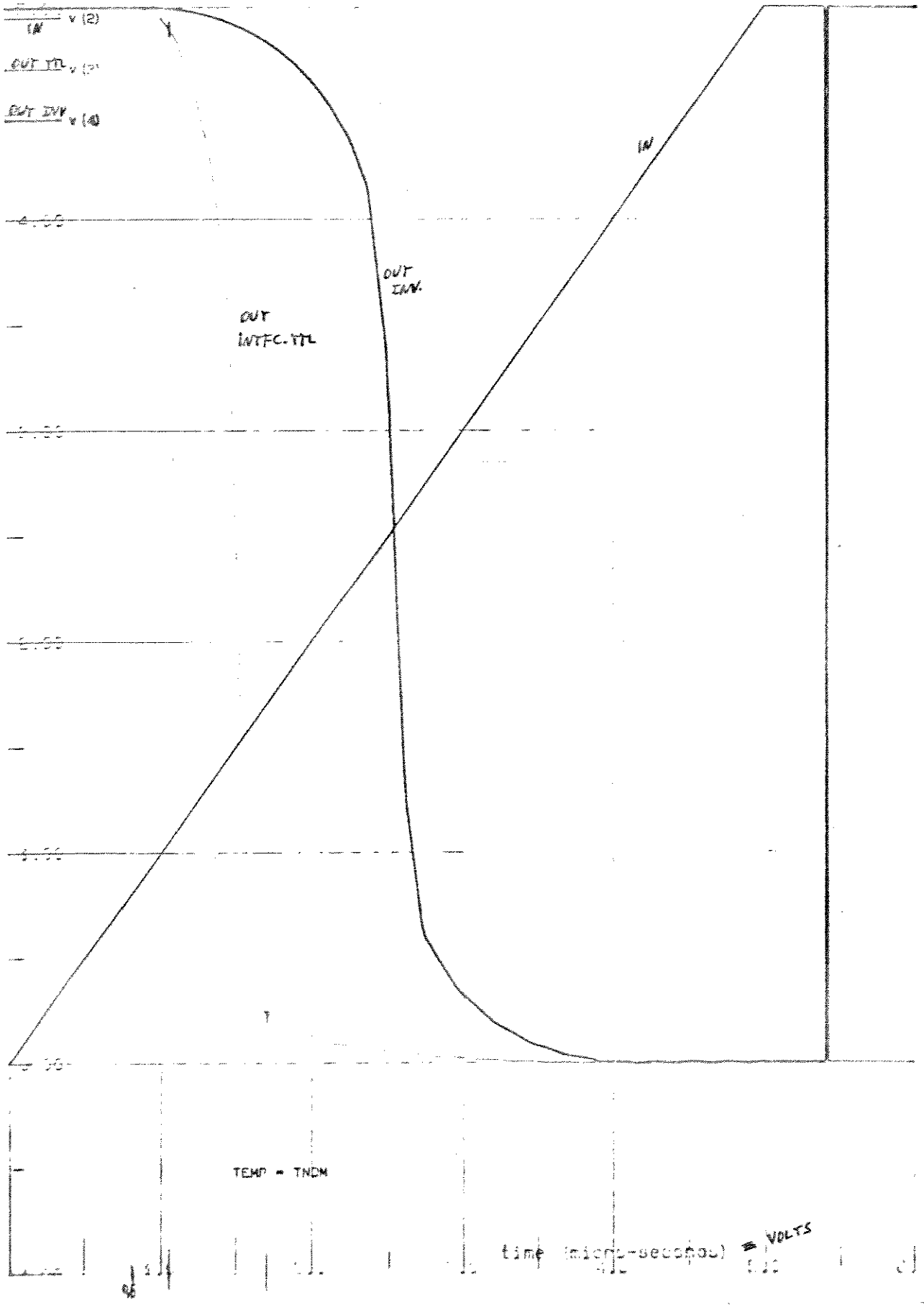
células INV e INV3X (com W, L finais)

```
inv, inv3x
Vdd 1 0 DC 5
VI 2 0 PULSE(0 5 20N 2N 2N 38N 100N)
.SUBCKT INV 1 2 3
MP1 3 2 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 3 2 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS INV
.SUBCKT INV3X 1 2 3
MP1 3 2 1 1 MODP L=3U W=102U AD=816P AS=816P PD=220U PS=220U
+NRD=0.078 NRS=0.078
MN1 3 2 0 0 MODN L=3U W=48U AD=384P AS=384P PD=112U PS=112U
+NRD=0.167 NRS=0.167
.ENDS INV3X
*descricao do circuito

X10 1 2 3 INV
X11 1 2 4 INV3X
C1 5 0 0.5P
C2 6 0 2P
V1 3 5 DC 0
V2 4 6 DC 0
.NODESET V(3)=5 V(4)=5
.MODEL MODP PMOS LEVEL=2 VTO=-0.9 TOX=525E-10 NSUB=2.6E15
+XJ=0.10U LD=0.40U UO=250 UCRIT=0.85E5 UEXP=0.16 VMAX=2.75E4
+NEFF=5.0 DELTA=1.2 RSH=110 CGSO=4.4E-10 CGDO=4.4E-10
+CJ=160U CJSW=260P MJ=0.46 MJSW=0.23 PB=0.65
.MODEL MODN NMOS LEVEL=2 VTO=0.90 TOX=525E-10 NSUB=1.1E16
+XJ=0.02U LD=0.35U UO=665 UCRIT=0.93E5 UEXP=0.16 VMAX=4.74E4
+NEFF=3.3 DELTA=1.7 RSH=40 CGSO=3.7E-10 CGDO=3.7E-10
+CJ=315U CJSW=345P MJ=0.53 MJSW=0.275 PB=0.85
.TRAN 0.1N 100N
.GRAPH TRAN V(2) V(3) V(4) I(V1) I(V2) I(Vdd)
.PLOT TRAN V(2) V(3) V(4) I(V1) V(2) I(Vdd)
.OPTIONS LIMPTS=10000 PIVTOL=1.0E-30
+ABSTOL=1N VNTOL=1M
.END
```


inv, inv3x
08/11/87 13:59:07
IN
v (2)
OUT INV
v (3)
OUT INV3X
v (4)
6.00
5.00
4.00
3.00
2.00
1.00
0.00
-1.00
IN
OUT INV3X
OUT INV
i (v1)
i (v2)
i (vdd)
.004
.002
0.00
-.002
-.004
-.006
i (vdd)
0.0
20.0
40.0
60.0
80.0
100
TEMP = TNOM
time (nano-seconds)

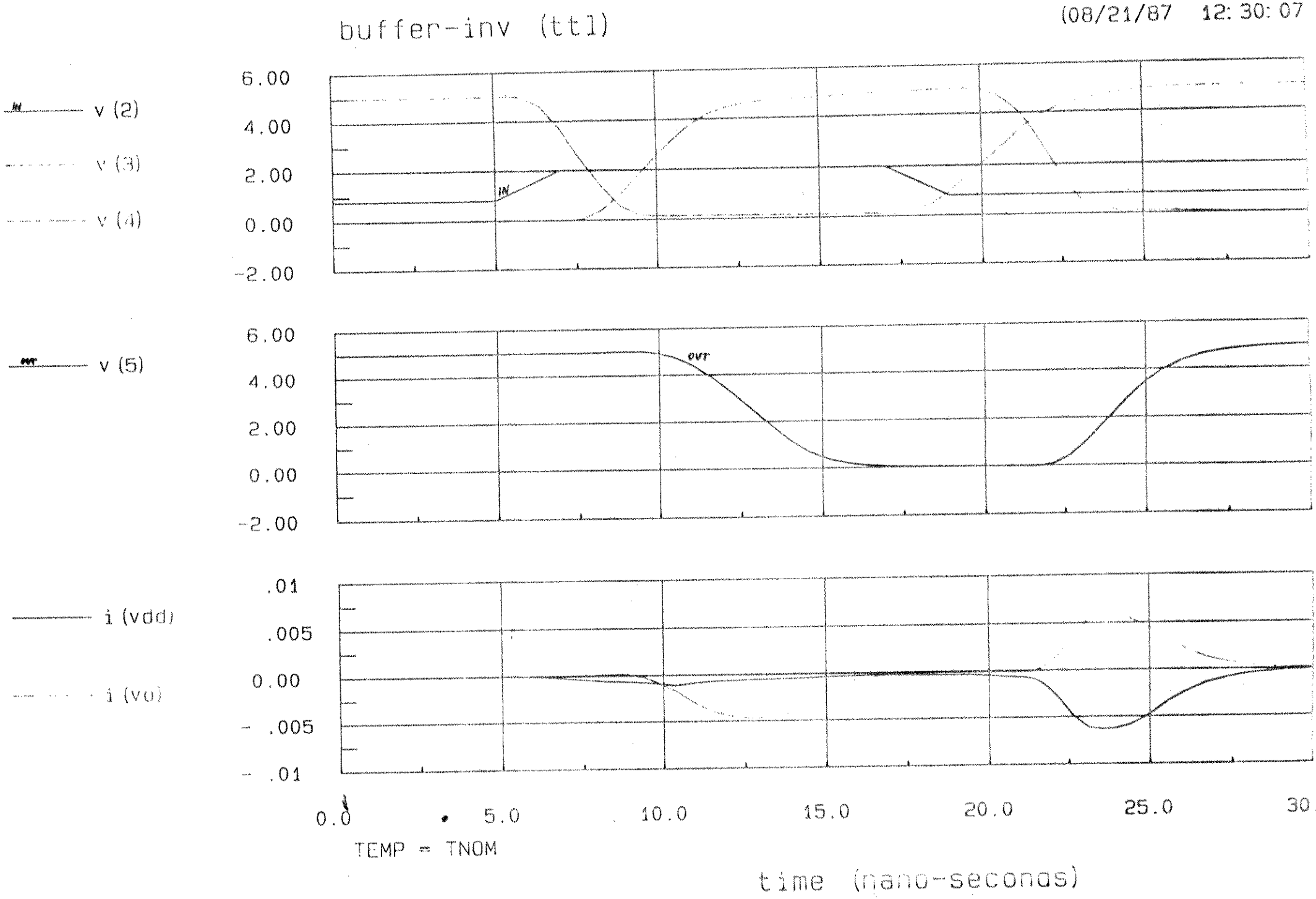
CÉLULA INTERF. TTL (CURVA TRANSFER.)

```
buffer-TTL e inv
Vdd 1 0 DC 5
VI 2 0 PULSE(0 5 2N 5U 2N 400N 6U)
.SUBCKT INV 1 2 3
MP1 3 2 1 1 MODP L=3U W=12U AD=90P AS=90P PD=39U PS=39U
+NRD=0.625 NRS=0.625
MN1 3 2 0 0 MODN L=3U W=6U AD=45P AS=45P PD=27U PS=27U
+NRD=1.25 NRS=1.25
.ENDS INV
.SUBCKT BUFTTL 1 2 3
MP1 3 2 1 1 MODP L=3U W=12U AD=90P AS=90P PD=39U PS=39U
+NRD=0.625 NRS=0.625
MN1 3 2 0 0 MODN L=3U W=80U AD=600P AS=600P PD=175U PS=175U
+NRD=0.094 NRS=0.094
.ENDS BUFTTL
.SUBCKT INV3X 1 2 3
MP1 3 2 1 1 MODP L=3U W=36U AD=270P AS=270P PD=87U PS=87U
+NRD=0.21 NRS=0.21
MN1 3 2 0 0 MODN L=3U W=18U AD=135P AS=135P PD=51U PS=51U
+NRD=0.42 NRS=0.42
.ENDS INV3X
.SUBCKT BUFFER 1 2 3
MP1 3 2 1 1 MODP L=3U W=220U AD=1650P AS=1650P PD=455U PS=455U
+NRD=0.034 NRS=0.034
MN1 3 2 0 0 MODN L=3U W=100U AD=750P AS=750P PD=215U PS=215U
+NRD=0.075 NRS=0.075
.ENDS BUFFER
*descricao do circuito
X10 1 2 3 BUFTTL
X11 1 2 4 INV
C1 3 0 0.1P
C2 4 0 0.05P
.NODESET V(3)=5 V(4)=5
.MODEL MODP PMOS LEVEL=2 VTO=-0.9 TOX=525E-10 NSUB=2.6E15
+XJ=0.10U LD=0.40U UO=250 UCRIT=0.85E5 UEXP=0.16 VMAX=2.75E4
+NEFF=5.0 DELTA=1.2 RSH=110 CGSO=4.4E-10 CGDO=4.4E-10
+CJ=160U CJSW=260P MJ=0.46 MJSW=0.23 PB=0.65
.MODEL MODN NMOS LEVEL=2 VTO=0.90 TOX=525E-10 NSUB=1.1E16
+XJ=0.02U LD=0.35U UO=665 UCRIT=0.93E5 UEXP=0.16 VMAX=4.74E4
+NEFF=3.3 DELTA=1.7 RSH=40 CGSO=3.7E-10 CGDO=3.7E-10
+CJ=315U CJSW=345P MJ=0.53 MJSW=0.275 PB=0.85
.TRAN 20N 6U
.GRAPH TRAN V(2) V(3) V(4)
.PLOT TRAN V(2) V(3) V(4)
.OPTIONS LIMPTS=10000 PIVTOL=1.0E-30
+ABSTOL=1N VNTOL=1M
.END
```


IN V (2)
OUT TTL V (3)
OUT INV V (4)
IN
OUT INV.
OUT INTFC. TTL
TEMP = TNOM
time (micro-seconds)
VOLTS

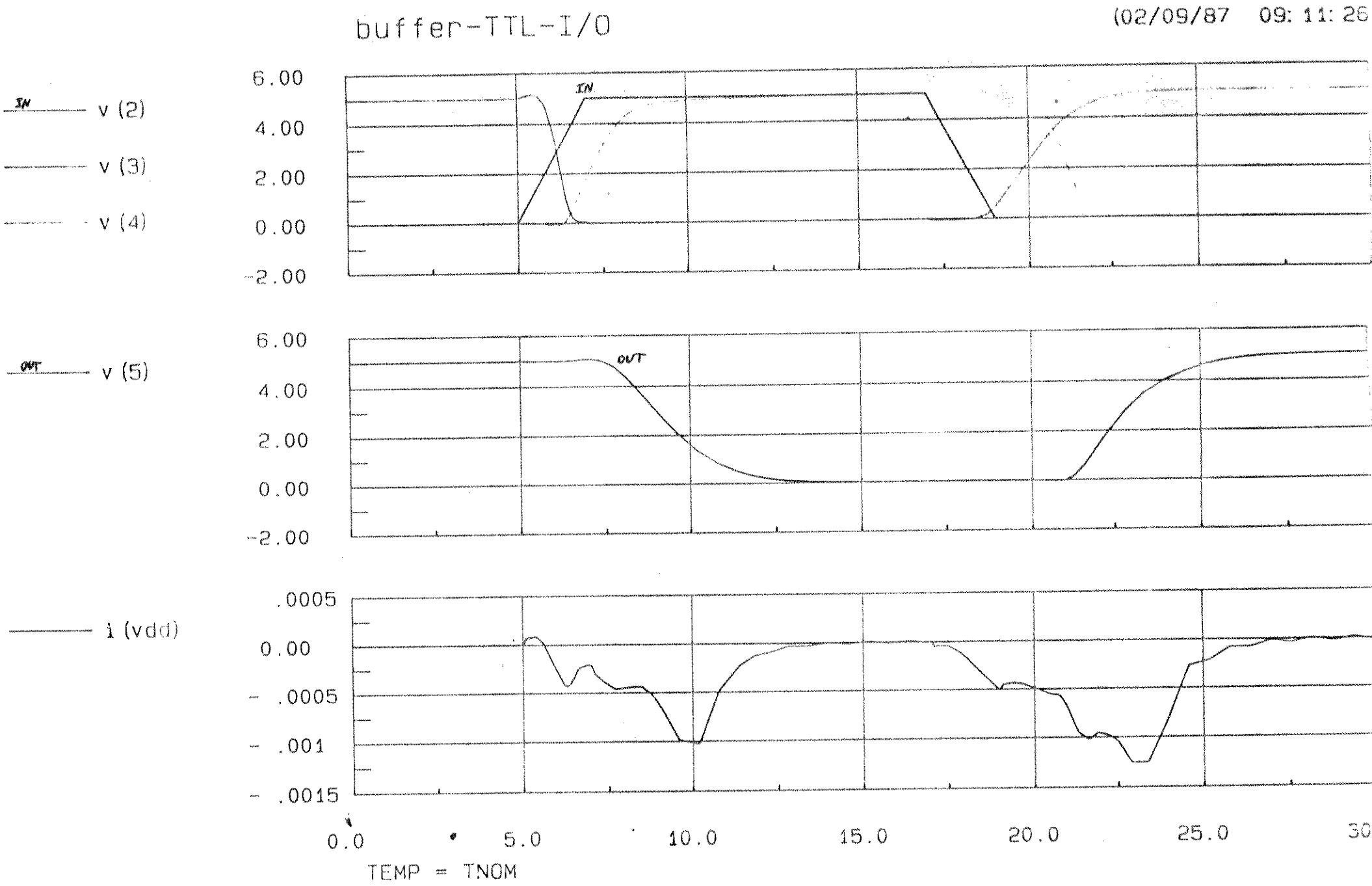
CÉLULA DE ENTRADA (INTERF. TTL, INVERSORA) ver fig. 32a

```
buffer-inv (ttl)
Vdd 1 0 DC 5
V1 2 0 PULSE(0.8 2.0 5N 2N 2N 10N 30N)
.SUBCKT INV 1 2 3
MP1 3 2 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 3 2 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS INV
.SUBCKT BUFTTL 1 2 3
MP1 3 2 1 1 MODP L=3U W=12U AD=96P AS=96P PD=40U PS=40U
+NRD=0.667 NRS=0.667
MN1 3 2 0 0 MODN L=3U W=80U AD=640P AS=640P PD=176U PS=176U
+NRD=0.1 NRS=0.1
.ENDS BUFTTL
.SUBCKT BUFFER 1 2 3
MP1 3 2 1 1 MODP L=3U W=170U AD=1360P AS=1360P PD=356U PS=356U
+NRD=0.047 NRS=0.047
MN1 3 2 0 0 MODN L=3U W=80U AD=640P AS=640P PD=176U PS=176U
+NRD=0.1 NRS=0.1
.ENDS BUFFER
.SUBCKT IAJ 1 2 3
MP1 3 2 1 1 MODP L=3U W=12U AD=96P AS=96P PD=40U PS=40U
+NRD=0.667 NRS=0.667
MN1 3 2 0 0 MODN L=3U W=12U AD=96P AS=96P PD=40U PS=40U
+NRD=0.667 NRS=0.667
.ENDS IAJ
.SUBCKT IAJ2 1 2 3
MP1 3 2 1 1 MODP L=3U W=25U AD=200P AS=200P PD=66U PS=66U
+NRD=0.32 NRS=0.32
MN1 3 2 0 0 MODN L=3U W=25U AD=200P AS=200P PD=66U PS=66U
+NRD=0.32 NRS=0.32
.ENDS IAJ2
*descricao do circuito
X10 1 2 3 BUFTTL
X11 1 3 4 IAJ2
X12 1 4 5 BUFFER
VO 5 6 DC 0
C1 6 0 4P
.NODESET V(5)=5
.MODEL MODP PMOS LEVEL=2 VTO=-0.9 TOX=525E-10 NSUB=2.6E15
+XJ=0.10U LD=0.40U UO=250 UCRIT=0.85E5 UEXP=0.16 VMAX=2.75E4
+NEFF=5.0 DELTA=1.2 RSH=110 CGSO=4.4E-10 CGDO=4.4E-10
+CJ=160U CJSW=260P MJ=0.46 MJSW=0.23 PB=0.65
.MODEL MODN NMOS LEVEL=2 VTO=0.90 TOX=525E-10 NSUB=1.1E16
+XJ=0.02U LD=0.35U UO=665 UCRIT=0.93E5 UEXP=0.16 VMAX=4.74E4
+NEFF=3.3 DELTA=1.7 RSH=40 CGSO=3.7E-10 CGDO=3.7E-10
+CJ=315U CJSW=345P MJ=0.53 MJSW=0.275 PB=0.85
.TRAN 0.1N 30N
.GRAPH TRAN V(2) V(3) V(4) V(5) I(Vdd) I(VO)
.PLOT TRAN V(2) V(3) V(5)
.OPTIONS LIMPTS=10000 PIVTOL=1.0E-30
+ABSTOL=1N VNTOL=1M
.END
```


buffer-inv (ttl)
(08/21/87 12: 30: 07
v (2)
v (3)
v (4)
v (5)
i (vdd)
i (vo)
6.00
4.00
2.00
0.00
-2.00
.01
.005
0.00
- .005
- .01
IN
OUT
0.0
5.0
10.0
15.0
20.0
25.0
30
TEMP = TNOM
time (nano-seconds)

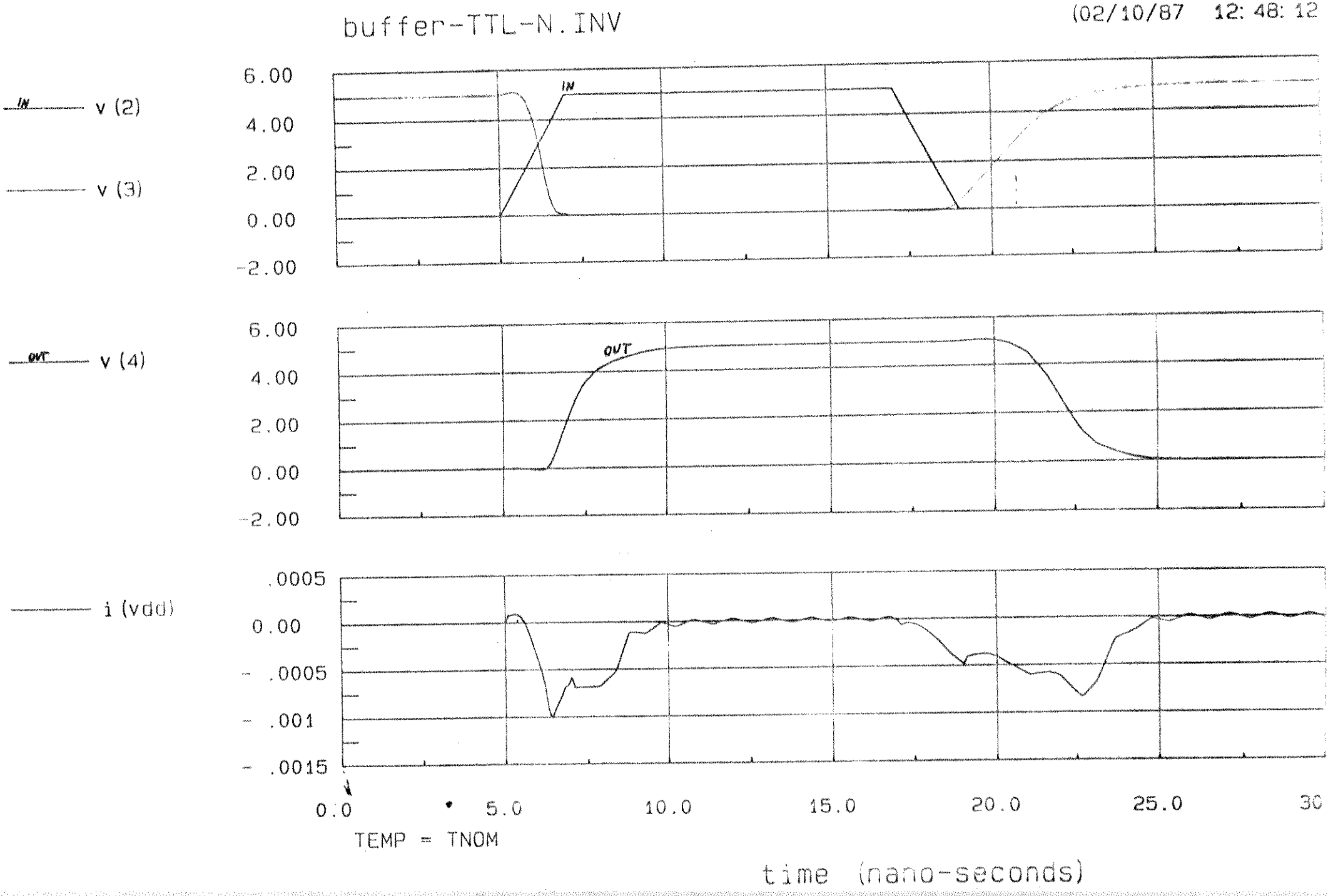
CIRCUITO DE ENTRADA DA CÉLULA I/O (INVERSOR) (ver fig. 32b)

```
buffer-TTL-I/O
Vdd 1 0 DC 5
VI 2 0 PULSE(0 5 5N 2N 2N 10N 30N)
.SUBCKT INV 1 2 3
MP1 3 2 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 3 2 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS INV
.SUBCKT BUFTTL 1 2 3
MP1 3 2 1 1 MODP L=3U W=12U AD=96P AS=96P PD=40U PS=40U
+NRD=0.667 NRS=0.667
MN1 3 2 0 0 MODN L=3U W=80U AD=640P AS=640P PD=176U PS=176U
+NRD=0.1 NRS=0.1
.ENDS BUFTTL
.SUBCKT INV3X 1 2 3
MP1 3 2 1 1 MODP L=3U W=36U AD=270P AS=270P PD=87U PS=87U
+NRD=0.21 NRS=0.21
MN1 3 2 0 0 MODN L=3U W=18U AD=135P AS=135P PD=51U PS=51U
+NRD=0.42 NRS=0.42
.ENDS INV3X
.SUBCKT BUFFER 1 2 3
MP1 3 2 1 1 MODP L=3U W=220U AD=1650P AS=1650P PD=455U PS=455U
+NRD=0.034 NRS=0.034
MN1 3 2 0 0 MODN L=3U W=100U AD=750P AS=750P PD=215U PS=215U
+NRD=0.075 NRS=0.075
.ENDS BUFFER
.SUBCKT IAJ 1 2 3
MP1 3 2 1 1 MODP L=3U W=12U AD=96P AS=96P PD=40U PS=40U
+NRD=0.667 NRS=0.667
MN1 3 2 0 0 MODN L=3U W=12U AD=96P AS=96P PD=40U PS=40U
+NRD=0.667 NRS=0.667
.ENDS IAJ
.SUBCKT TG 1 2 3 4 5
MP1 5 2 4 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 4 3 5 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS TG
*descricao do circuito
X10 1 2 3 BUFTTL
X11 1 3 4 IAJ
X12 1 4 5 INV
X13 1 0 1 5 6 TG
X14 1 6 7 INV
X15 1 5 8 INV
C1 5 0 0.2P
.NODESET V(5)=5 V(6)=5
.MODEL MODP PMOS LEVEL=2 VTO=-0.9 TOX=525E-10 NSUB=2.6E15
+XJ=0.10U LD=0.40U U0=250 UCRIT=0.85E5 UEXP=0.16 VMAX=2.75E4
+NEFF=5.0 DELTA=1.2 RSH=110 CGSO=4.4E-10 CGDO=4.4E-10
+CJ=160U CJSW=260P MJ=0.46 MJSW=0.23 PB=0.65
.MODEL MODN NMOS LEVEL=2 VTO=0.90 TOX=525E-10 NSUB=1.1E16
+XJ=0.02U LD=0.35U U0=665 UCRIT=0.93E5 UEXP=0.16 VMAX=4.74E4
+NEFF=3.3 DELTA=1.7 RSH=40 CGSO=3.7E-10 CGDO=3.7E-10
+CJ=315U CJSW=345P MJ=0.53 MJSW=0.275 PB=0.85
.TRAN 0.1N 30N
.GRAPH TRAN V(2) V(3) V(4) V(5) V(6) V(7) V(8) I(Vdd)
.PLOT TRAN V(2) V(3) V(5)
.OPTIONS LIMPTS=10000 PIVTOL=1.0E-30
+ABSTOL=1N VNTOL=1M
.END
```


buffer-TTL-I/O
02/09/87 09:11:26
IN
v (2)
v (3)
v (4)
6.00
4.00
2.00
0.00
-2.00
IN
OUT
v (5)
6.00
4.00
2.00
0.00
-2.00
OUT
i (vdd)
.0005
0.00
- .0005
- .001
- .0015
0.0
5.0
10.0
15.0
20.0
25.0
30
TEMP = TNOM
time (nano-seconds)

CÉLULA INTERF. TTL DE ENTRADA - NÃO INVERSORA (ver fig. 33)

```
buffer-TTL-N.INV
Vdd 1 0 DC 5
VI 2 0 PULSE(0 5 5N 2N 2N 10N 30N)
.SUBCKT INV 1 2 3
MP1 3 2 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 3 2 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS INV
.SUBCKT BUFTTL 1 2 3
MP1 3 2 1 1 MODP L=3U W=12U AD=96P AS=96P PD=40U PS=40U
+NRD=0.667 NRS=0.667
MN1 3 2 0 0 MODN L=3U W=80U AD=640P AS=640P PD=176U PS=176U
+NRD=0.1 NRS=0.1
.ENDS BUFTTL
.SUBCKT INV3X 1 2 3
MP1 3 2 1 1 MODP L=3U W=36U AD=270P AS=270P PD=87U PS=87U
+NRD=0.21 NRS=0.21
MN1 3 2 0 0 MODN L=3U W=18U AD=135P AS=135P PD=51U PS=51U
+NRD=0.42 NRS=0.42
.ENDS INV3X
.SUBCKT BUFFER 1 2 3
MP1 3 2 1 1 MODP L=3U W=220U AD=1650P AS=1650P PD=455U PS=455U
+NRD=0.034 NRS=0.034
MN1 3 2 0 0 MODN L=3U W=100U AD=750P AS=750P PD=215U PS=215U
+NRD=0.075 NRS=0.075
.ENDS BUFFER
.SUBCKT IAJ 1 2 3
MP1 3 2 1 1 MODP L=3U W=12U AD=96P AS=96P PD=40U PS=40U
+NRD=0.667 NRS=0.667
MN1 3 2 0 0 MODN L=3U W=12U AD=96P AS=96P PD=40U PS=40U
+NRD=0.667 NRS=0.667
.ENDS IAJ
.SUBCKT TG 1 2 3 4 5
MP1 5 2 4 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 4 3 5 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS TG
*descricao do circuito
X10 1 2 3 BUFTTL
X11 1 3 4 INV
X12 1 0 1 4 5 TG
X14 1 5 6 INV
.NODESET V(4)=0 V(5)=0
.MODEL MODP PMOS LEVEL=2 VTO=-0.9 TOX=525E-10 NSUB=2.6E15
+XJ=0.10U LD=0.40U UO=250 UCRIT=0.85E5 UEXP=0.16 VMAX=2.75E4
+NEFF=5.0 DELTA=1.2 RSH=110 CGSO=4.4E-10 CGDO=4.4E-10
+CJ=160U CJSW=260P MJ=0.46 MJSW=0.23 PB=0.65
.MODEL MODN NMOS LEVEL=2 VTO=0.90 TOX=525E-10 NSUB=1.1E16
+XJ=0.02U LD=0.35U UO=665 UCRIT=0.93E5 UEXP=0.16 VMAX=4.74E4
+NEFF=3.3 DELTA=1.7 RSH=40 CGSO=3.7E-10 CGDO=3.7E-10
+CJ=315U CJSW=345P MJ=0.53 MJSW=0.275 PB=0.85
.TRAN 0.1N 30N
.GRAPH TRAN V(2) V(3) V(4) V(5) V(6) I(Vdd)
.PLOT TRAN V(2) V(3) V(5)
.OPTIONS LIMPTS=10000 PIVTOL=1.0E-30
.OPTIOL-1N VNTOL 1M
.END
```

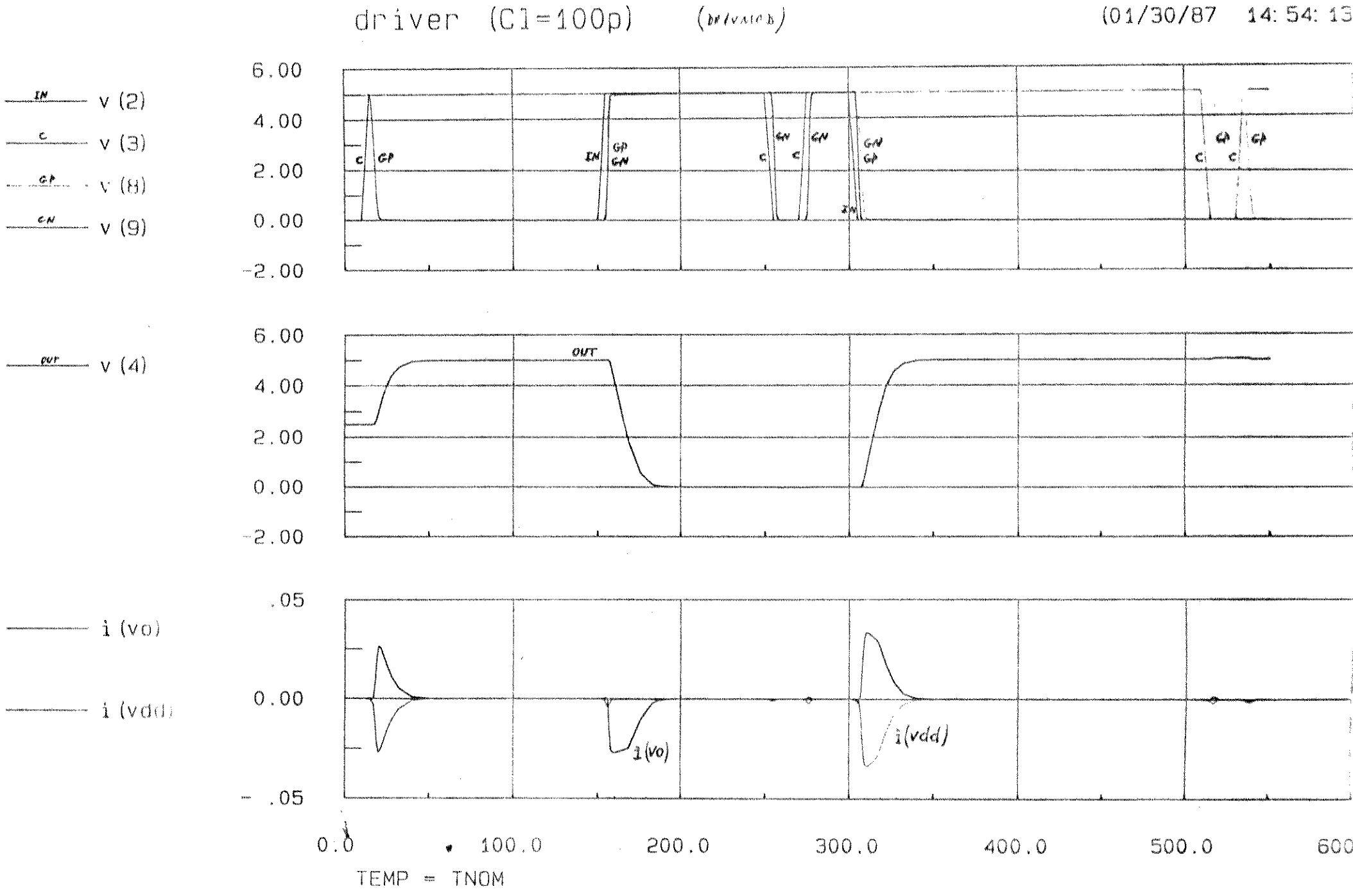

buffer-TTL-N.INV
02/10/87 12: 48: 12
IN v (2)
v (3)
OUT v (4)
i (vdd)
6.00
4.00
2.00
0.00
-2.00
IN
OUT
.0005
0.00
- .0005
- .001
- .0015
0.0
5.0
10.0
15.0
20.0
25.0
30
TEMP = TNOM
time (nano-seconds)

CÉLULA DRIVER - TRISTATE DE SAÍDA (INVERSOR) (ver fig. 34)

```
driver (C1=100p)



Vdd 1 0 DC 5
VI 2 0 PULSE(0 5 150N 5N 5N 145N 400N)
VC 3 0 PULSE(0 5 10N 5N 5N 235N 260N)
.SUBCKT NAND 1 2 3 4
MP1 4 2 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MP2 4 3 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 4 2 5 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
MN2 5 3 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS NAND
.SUBCKT INV 1 2 3
MP1 3 2 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 3 2 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS INV
.SUBCKT TG 1 2 3 4 5
MP1 5 2 4 1 MODP L=3U W=12U AD=90P AS=90P PD=39U PS=39U
+NRD=0.625 NRS=0.625
MN1 4 3 5 0 MODN L=3U W=6U AD=45P AS=45P PD=27U PS=27U
+NRD=1.25 NRS=1.25
.ENDS TG
.SUBCKT NOR 1 2 3 4
MP1 5 2 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MP2 4 3 5 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 4 2 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
MN2 4 3 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS NOR
.SUBCKT INV3X 1 2 3
MP1 3 2 1 1 MODP L=3U W=102U AD=816P AS=816P PD=220U PS=220U
+NRD=0.078 NRS=0.078
MN1 3 2 0 0 MODN L=3U W=48U AD=384P AS=384P PD=112U PS=112U
+NRD=0.17 NRS=0.17
.ENDS INV3X
```

cont

```
X1  1  2  5  6 NOR
X2  1  2  3  7 NAND
X3  1  3  5 INV
X4  1  6  8 INV3X
X5  1  7  9 INV3X
MP1  4  8  1  1 MODP L=3U W=800U AD=6400P AS=6400P PD=1616U PS=1616U
+NRD=0.01 NRS=0.01
MN1  4  9  0  0 MODN L=3U W=350U AD=2800P AS=2800P PD=716U PS=716U
+NRD=0.023 NRS=0.023
C1  10  0 100P
VO  4  10 DC 0
.NODESET V(4)=0
.MODEL MODP PMOS LEVEL=2 VTO=-0.9 TOX=525E-10 NSUB=2.6E15
+XJ=0.10U LD=0.40U UO=250 UCRIT=0.85E5 UEXP=0.16 VMAX=2.75E4
+NEFF=5.0 DELTA=1.2 RSH=110 CGSO=4.4E-10 CGDO=4.4E-10
+CJ=160U CJSW=260P MJ=0.46 MJSW=0.23 PB=0.65
.MODEL MODN NMOS LEVEL=2 VTO=0.90 TOX=525E-10 NSUB=1.1E16
+XJ=0.02U LD=0.35U UO=665 UCRIT=0.93E5 UEXP=0.16 VMAX=4.74E4
+NEFF=3.3 DELTA=1.7 RSH=40 CGSO=3.7E-10 CGDO=3.7E-10
+CJ=315U CJSW=345P MJ=0.53 MJSW=0.275 PB=0.85
.TRAN 1N 550N
.GRAPH TRAN V(2) V(3) V(8) V(9) V(4) I(VO) I(Vdd)
.PLOT TRAN V(2) V(3) V(8) V(9) V(4) I(VO) I(Vdd)
.OPTIONS LIMPTS=700 PIVTOL=1.0E-30
+ABSTOL=1N VNTOL=1M
.END
```


driver (C1=100p)
(01/30/87 14:54:13
IN v (2)
C v (3)
GP v (8)
GN v (9)
OUT v (4)
i (vo)
i (vdd)
6.00
4.00
2.00
0.00
-2.00
.05
0.00
-.05
OUT
i(vo)
i(vdd)
0.0
100.0
200.0
300.0
400.0
500.0
600
TEMP = TNOM
time (nano-seconds)

CIRCUITO COMPLETO PARA SIMULAÇÃO DA CÉLULA MACROX/R

(CIRCMAX)

```
Vdd 1 0 DC 5
VCK 3 0 PULSE(0 5 110N 5N 5N 35N 70N)
VRB 14 0 PULSE(5 0 35N 5N 5N 15N 525N)
VSB 16 0 PULSE(5 0 60N 5N 5N 15N 600N)
VCLR 12 0 PULSE(5 0 10N 5N 5N 10N 575N)
VC/O 4 0 PULSE(5 0 130N 5N 5N 165N 310N)
VI/O 31 0 PULSE(0 5 155N 2N 2N 80N 350N)
VA0 24 0 PULSE(5 0 20N 2N 2N 100N 520N)
VA1 22 0 DC 0
VA2 23 0 DC 0
VIN 32 0 PULSE(5 0 170N 5N 5N 60N 130N)
VIX 33 0 PULSE(5 0 450N 2N 2N 200N 300N)
.SUBCKT NAND 1 2 3 4
MP1 4 2 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MP2 4 3 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 4 2 5 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
MN2 5 3 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS NAND
.SUBCKT INV 1 2 3
MP1 3 2 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 3 2 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS INV
.SUBCKT INV3X 1 2 3
MP1 3 2 1 1 MODP L=3U W=36U AD=270P AS=270P PD=87U PS=87U
+NRD=0.21 NRS=0.21
MN1 3 2 0 0 MODN L=3U W=18U AD=135P AS=135P PD=51U PS=51U
+NRD=0.42 NRS=0.42
.ENDS INV3X
.SUBCKT TG 1 2 3 4 5
MP1 5 2 4 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 4 3 5 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS TG
.SUBCKT NOR 1 2 3 4
MP1 5 2 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MP2 4 3 5 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 4 2 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
MN2 4 3 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS NOR
```

CONT

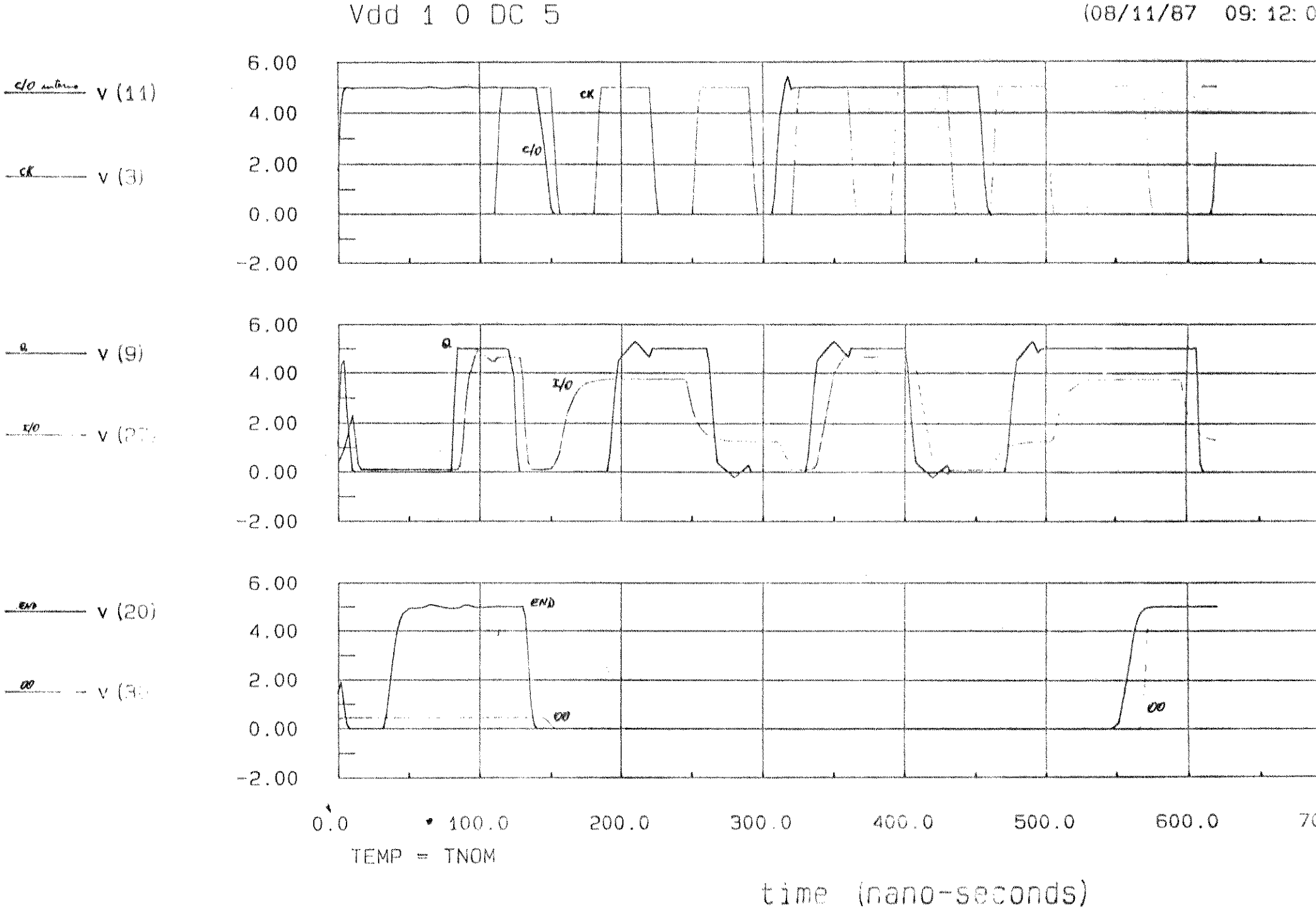
```
.SUBCKT NOR3 1 2 3 4 5
MP1 7 2 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MP2 6 3 7 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MP3 5 4 6 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 5 2 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
MN2 5 3 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
MN3 5 4 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS NOR3
.SUBCKT NAND3 1 2 3 4 5
MP1 5 2 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MP2 5 3 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MP3 5 4 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 7 2 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
MN2 6 3 7 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
MN3 5 4 6 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS NAND3
.SUBCKT XOR 1 2 3 4
XS1 1 2 3 5 NOR
MP1 6 2 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MP2 6 3 1 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MP3 4 5 6 1 MODP L=3U W=34U AD=272P AS=272P PD=84U PS=84U
+NRD=0.235 NRS=0.235
MN1 4 2 7 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
MN2 7 3 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
MN3 4 5 0 0 MODN L=3U W=16U AD=128P AS=128P PD=48U PS=48U
+NRD=0.5 NRS=0.5
.ENDS XOR
.SUBCKT FFDSR 1 5 3 7 8 14 13
XF1 1 2 3 5 4 TG
XF2 1 3 2 9 4 TG
XF3 1 3 2 6 10 TG
XF4 1 2 3 12 10 TG
XF5 1 4 7 6 NAND
XF6 1 6 8 9 NAND
XF7 1 10 8 11 NAND
XF8 1 11 7 12 NAND
XF9 1 3 2 INV
XF10 1 11 13 INV3X
XF10B 1 12 14 INV3X
.ENDS FFDSR
.SUBCKT LDCL 1 2 3 6 8
XL1 1 3 4 INV
XL2 1 7 8 INV3X
XL3 1 3 4 2 5 TG
XL4 1 4 3 8 5 TG
XL5 1 5 6 7 NOR
.ENDS LDCL
```

Cont

```
.SUBCKT BUFTTL 1 2 3
MP1 3 2 1 1 MODP L=3U W=12U AD=90P AS=90P PD=39U PS=39U
+NRD=0.625 NRS=0.625
MN1 3 2 0 0 MODN L=3U W=80U AD=600P AS=600P PD=175U PS=175U
+NRD=0.094 NRS=0.094
.ENDS BUFTTL
.SUBCKT BUFFER 1 2 3
MP1 3 2 1 1 MODP L=3U W=220U AD=1650P AS=1650P PD=455U PS=455U
+NRD=0.034 NRS=0.034
MN1 3 2 0 0 MODN L=3U W=100U AD=750P AS=750P PD=215U PS=215U
+NRD=0.075 NRS=0.075
.ENDS BUFFER
.SUBCKT BUFINV 1 2 3
XB1 1 2 4 BUFTTL
XB2 1 4 5 INV3X
XB3 1 5 3 BUFFER
.ENDS BUFINV
.SUBCKT BUFNIN 1 2 3
XB5 1 2 4 BUFTTL
XB6 1 4 3 INV3X
.ENDS BUFNIN
.SUBCKT DRIVNP 1 2 3 4
MP1 4 3 1 1 MODP L=3U W=800U AD=6000P AS=6000P PD=1615U PS=1615U
+NRD=0.00938 NRS=0.00938
MN1 4 2 0 0 MODN L=3U W=350U AD=2625P AS=2625P PD=715U PS=715U
+NRD=0.0214 NRS=0.0214
.ENDS DRIVNP
.SUBCKT DR3ST 1 2 3 4
XD1 1 2 5 6 NOR
XD2 1 2 3 7 NAND
XD3 1 3 5 INV
XD4 1 6 8 INV3X
XD5 1 7 9 INV3X
XD6 1 9 8 4 DRIVNP
.ENDS DR3ST
*descricao do circuito
X1 1 4 6 BUFINV
X2 1 6 11 INV3X
X3 1 3 2 BUFINV
X4 1 12 13 BUFINV
X5 1 14 15 BUFINV
X6 1 16 17 BUFINV
X7 1 24 25 BUFINV
X8 1 25 21 INV3X
X9 1 27 28 BUFINV
X10 1 15 11 20 18 NAND3
X11 1 17 11 20 8 NAND3
X12 1 18 19 INV
X13 1 19 13 7 NOR
X14 1 21 22 23 20 NOR3
X20 1 32 33 5 NAND
X15 1 5 2 7 8 9 16 TP20R
X16 1 10 11 27 DR3ST
X17 1 20 26 INV
X18 1 26 20 28 29 TG
X19 1 29 6 30 DR3ST
```

CONT

```
R1  31  27  1K
R2  27  0  2K
R3  27  1  2K
RPU  29  1  10K
C6  6  0  2P
C11  11  0  0.2P
C2  2  0  2P
C13  13  0  1.4P
C15  15  0  0.9P
C17  17  0  0.9P
C25  25  0  0.7P
C21  21  0  0.5P
C10  27  0  10P
C00  30  0  10P
.NODESET V(6)=0 V(2)=5 V(13)=0 V(18)=5 V(25)=0 V(9)=0
+V(10)=5 V(27)=0 V(29)=5 V(30)=0 V(5)=0
.MODEL MODP PMOS LEVEL=2 VTO=-0.9 TOX=525E-10 NSUB=2.6E15
+XJ=0.10U LD=0.40U UO=250 UCRIT=0.85E5 UEXP=0.16 VMAX=2.75E4
+NEFF=5.0 DELTA=1.2 RSH=110 CGSO=4.4E-10 CGDO=4.4E-10
+CJ=160U CJSW=260P MJ=0.46 MJSW=0.23 PB=0.65
.MODEL MODN NMOS LEVEL=2 VTO=0.90 TOX=525E-10 NSUB=1.1E16
+XJ=0.02U LD=0.35U UO=665 UCRIT=0.93E5 UEXP=0.16 VMAX=4.74E4
+NEFF=3.3 DELTA=1.7 RSH=40 CGSO=3.7E-10 CGDO=3.7E-10
+CJ=315U CJSW=3128P MJ=0.53 MJSW=0.275 PB=0.85
* def analise
.TRAN 2N 620N UIC
.PLOT TRAN V(11) V(3) V(7) V(8) V(9) V(27) V(20) V(30)
.GRAPH TRAN V(11) V(3) V(7) V(8) V(9) V(27) V(20) V(30)
.GRAPH TRAN V(2) V(10) V(5)
.OPTIONS LIMPTS=1000 PIVTOL=1.0E-30 TRTOL=40 RELTOL=0.1
+VNTOL=1E-3 ABSTOL=1E-6 ITL4=30 CHGTOL=1.0E-9
.END
```


Vdd 1 0 DC 5
(08/11/87 09: 12: 0
c/o antirace
v (11)
ck
v (3)
a
v (9)
x/o
v (2
end
v (20)
oo
v (3
6.00
4.00
2.00
0.00
-2.00
c/o
ck
a
x/o
end
oo
0.0
100.0
200.0
300.0
400.0
500.0
600.0
TEMP = TNOM
time (nano-seconds)

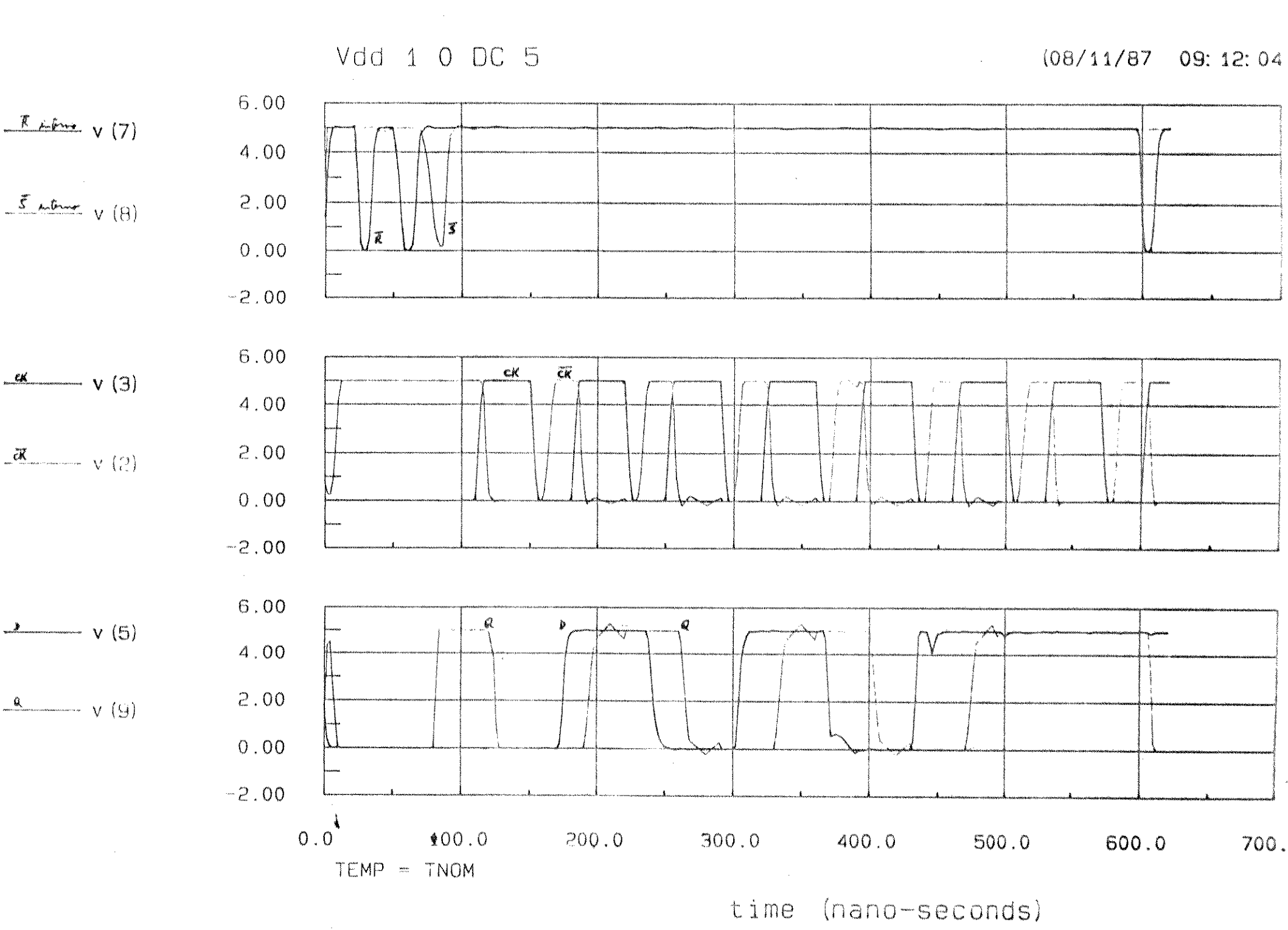
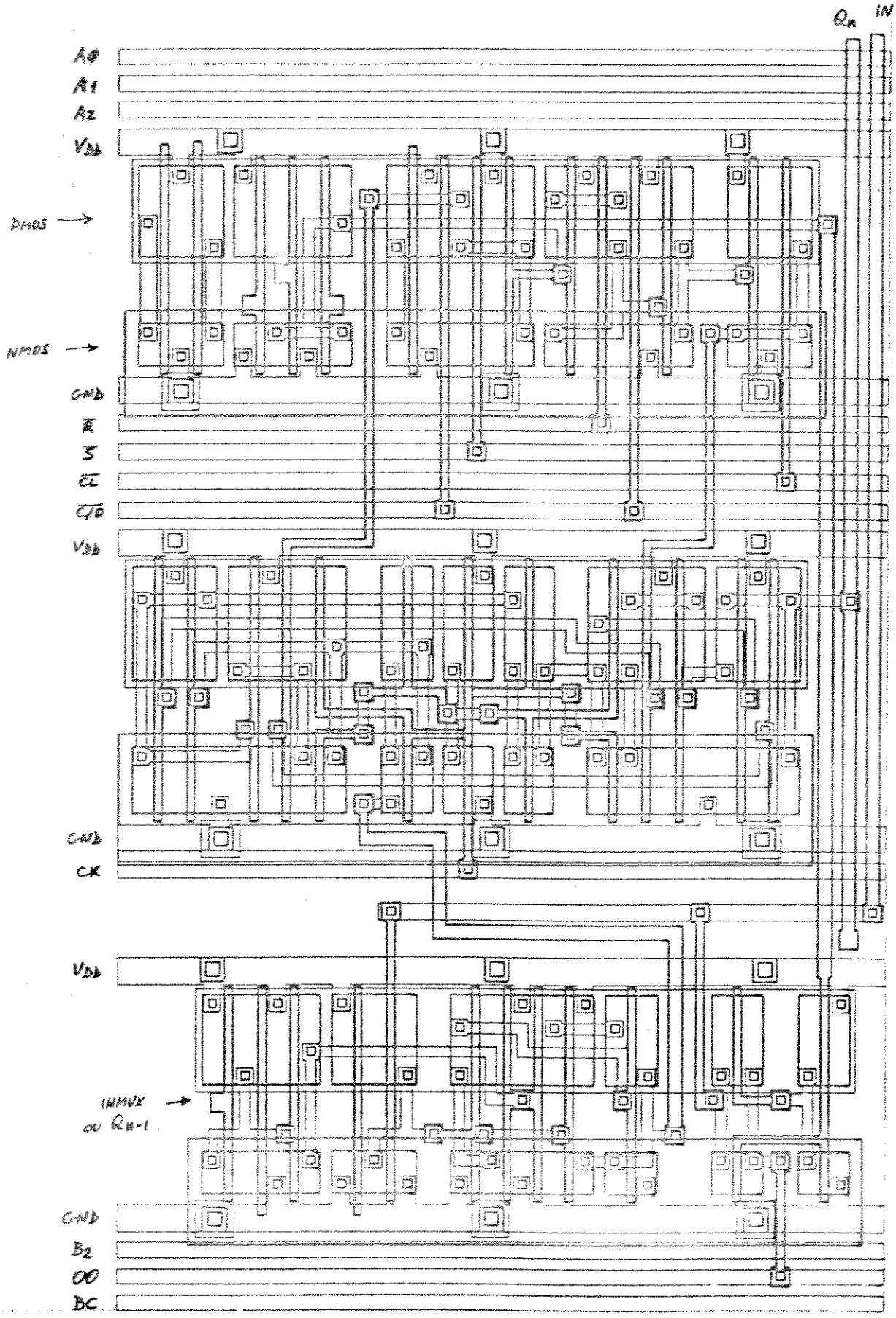

Vdd 1 0 DC 5
(08/11/87 09: 12: 04
R interne v (7)
S interne v (8)
CK v (3)
CK v (2)
D v (5)
Q v (9)
6.00
4.00
2.00
0.00
-2.00
0.0
100.0
200.0
300.0
400.0
500.0
600.0
700.
TEMP = TNOM
time (nano-seconds)

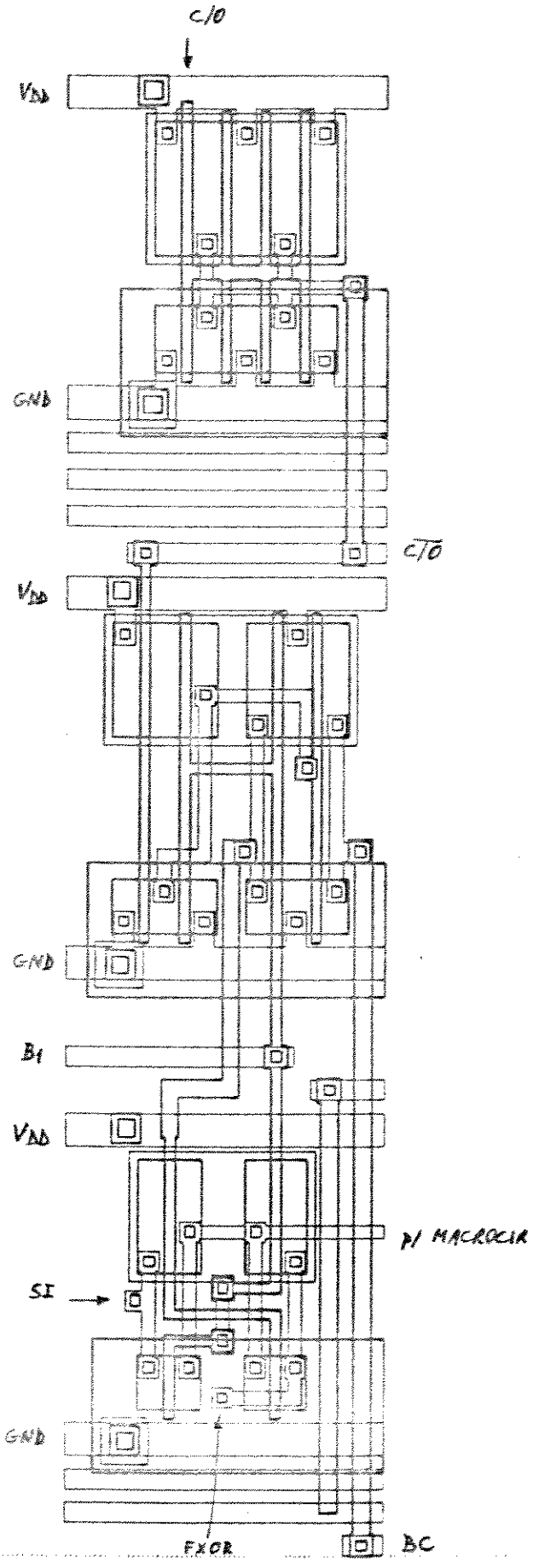
Fig 2

APÊNDICE III

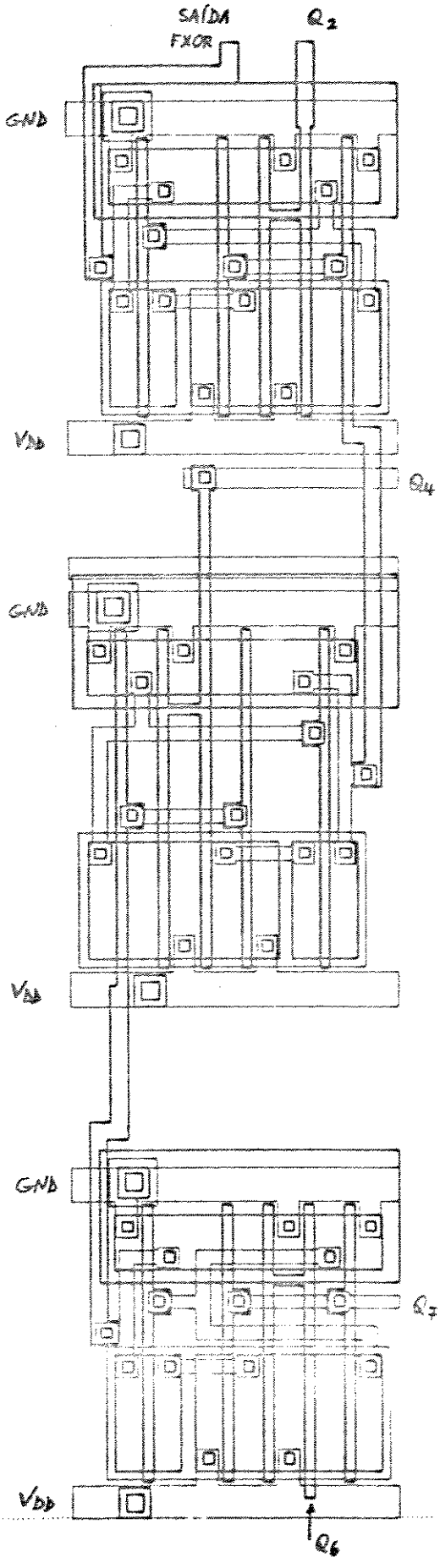
Lay-out das células principais e do circuito completo - CTIP. As células, desenhadas com o editor gráfico KIC, foram codificadas no formato CIF (ver [CTI86]) e plotadas.



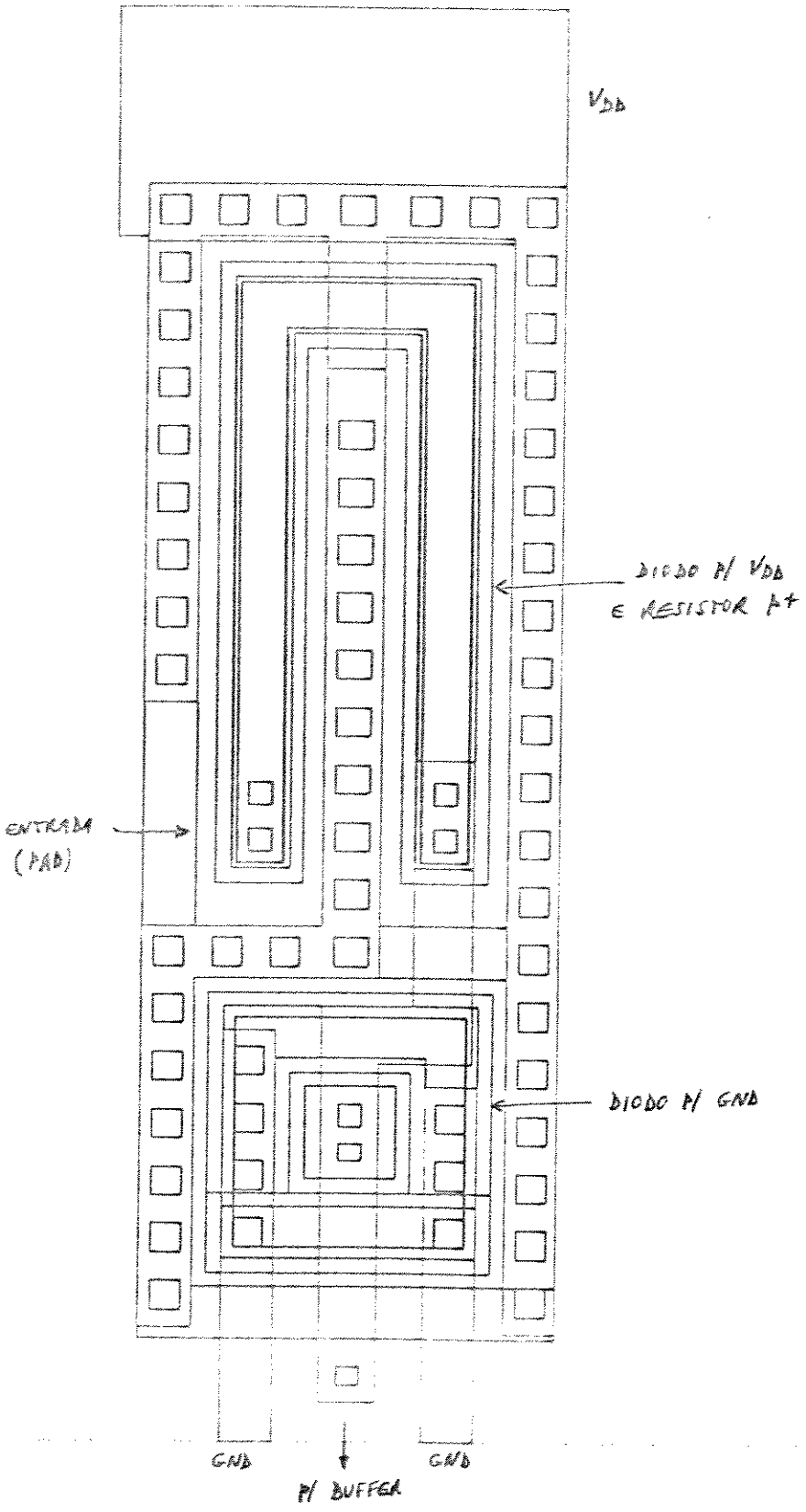
INTER> MACROCIR.CIF



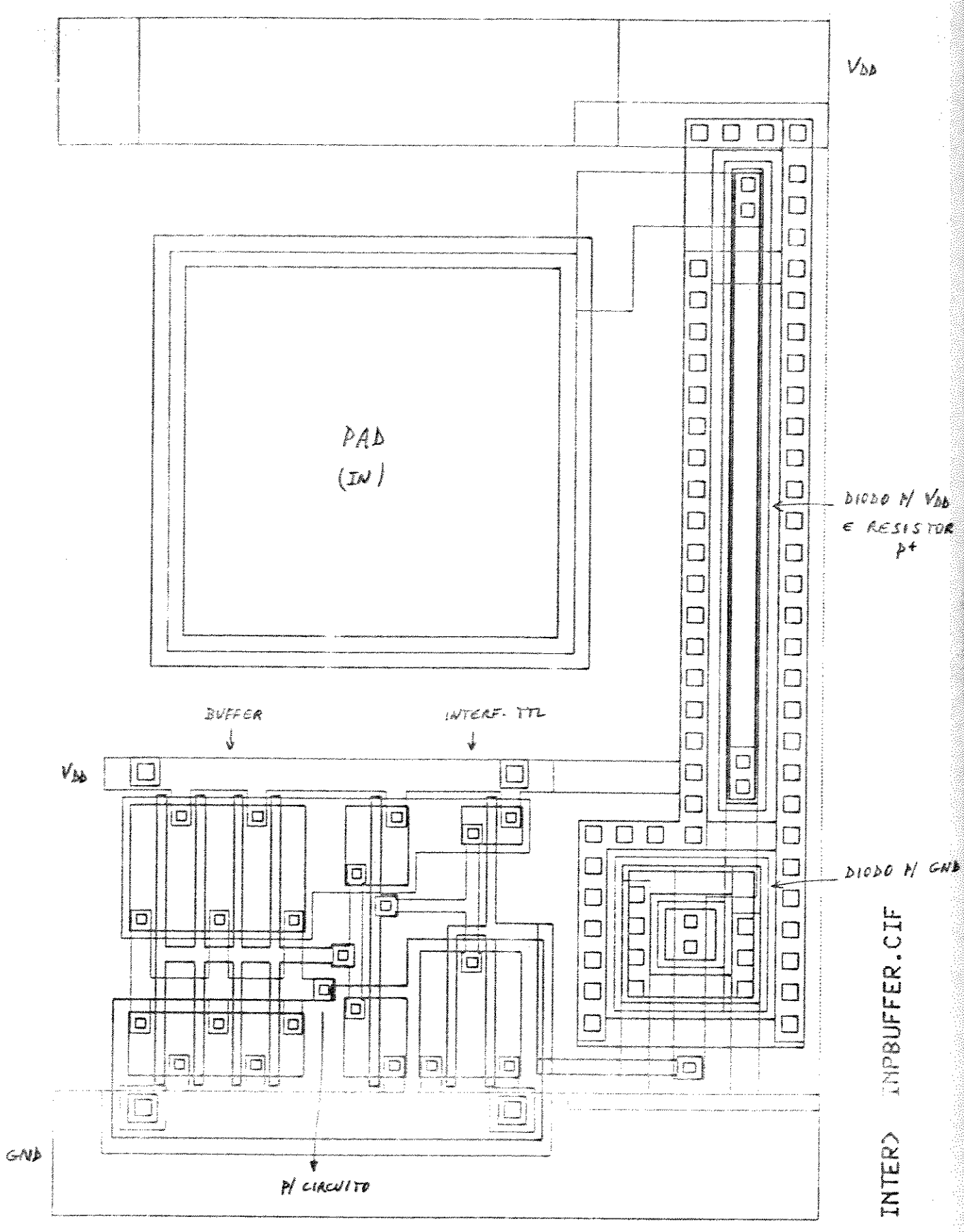
INTER> INMUX.CIF



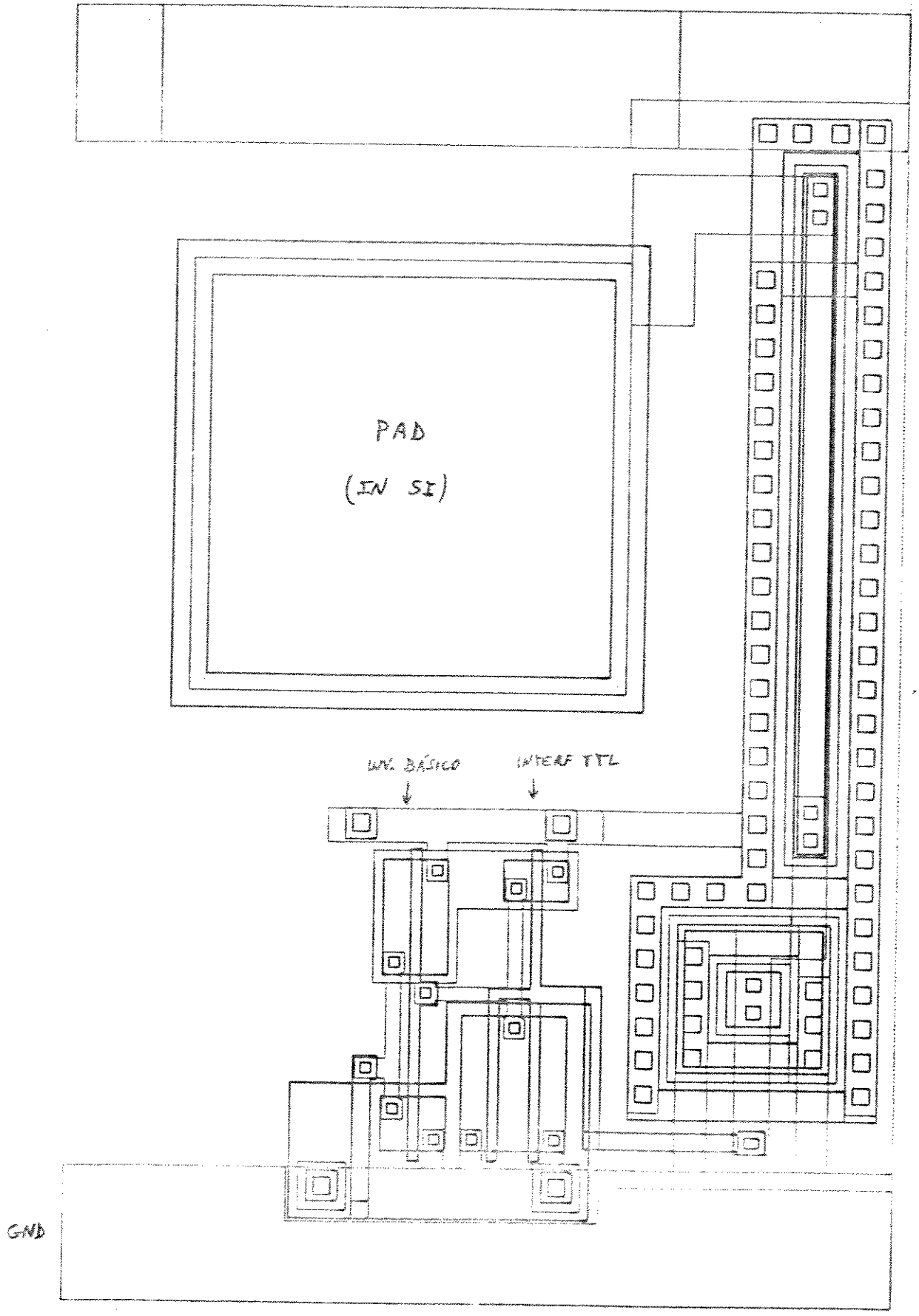
INTERF FXOR.CIF



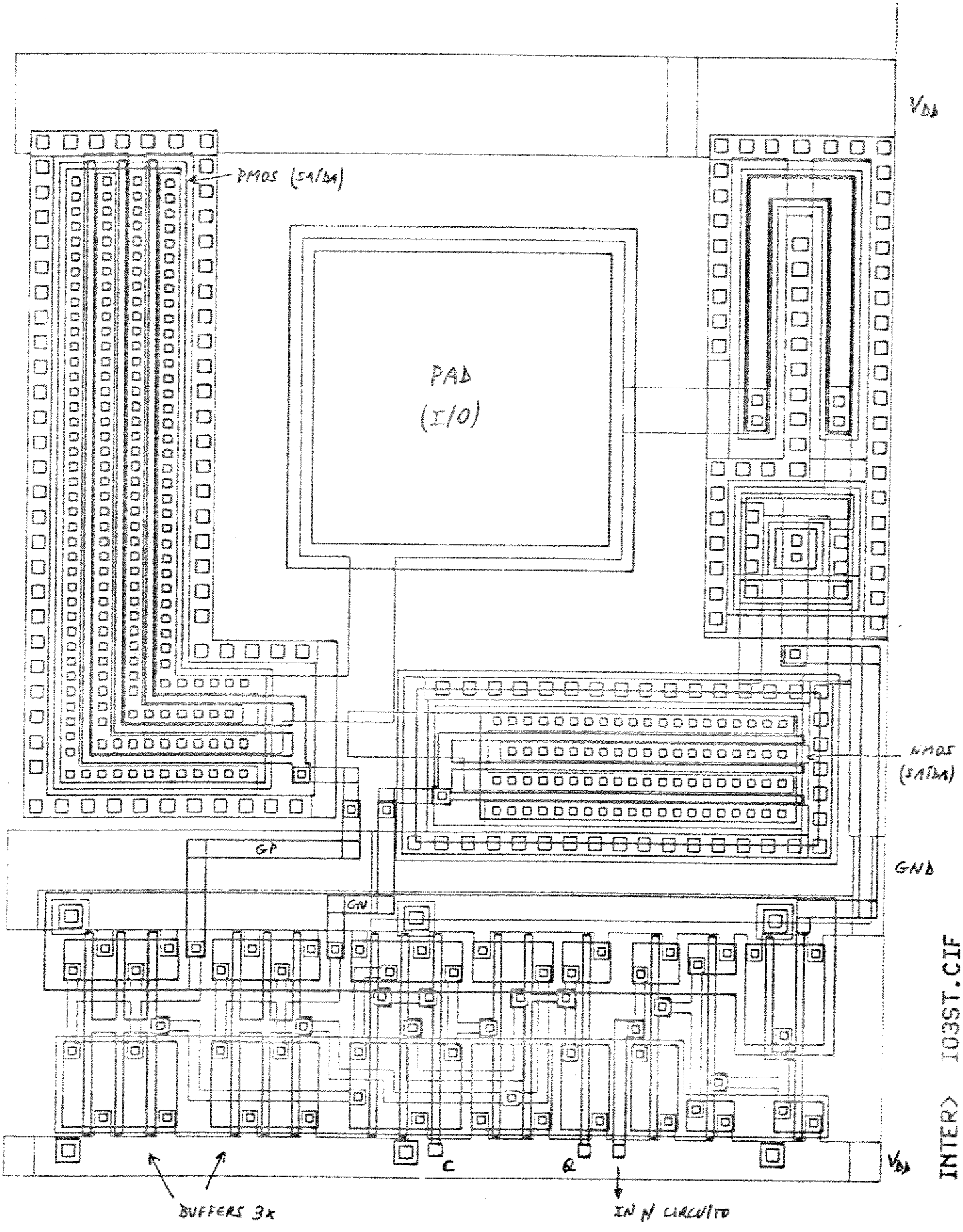
INTER> DISPROTC.CIF



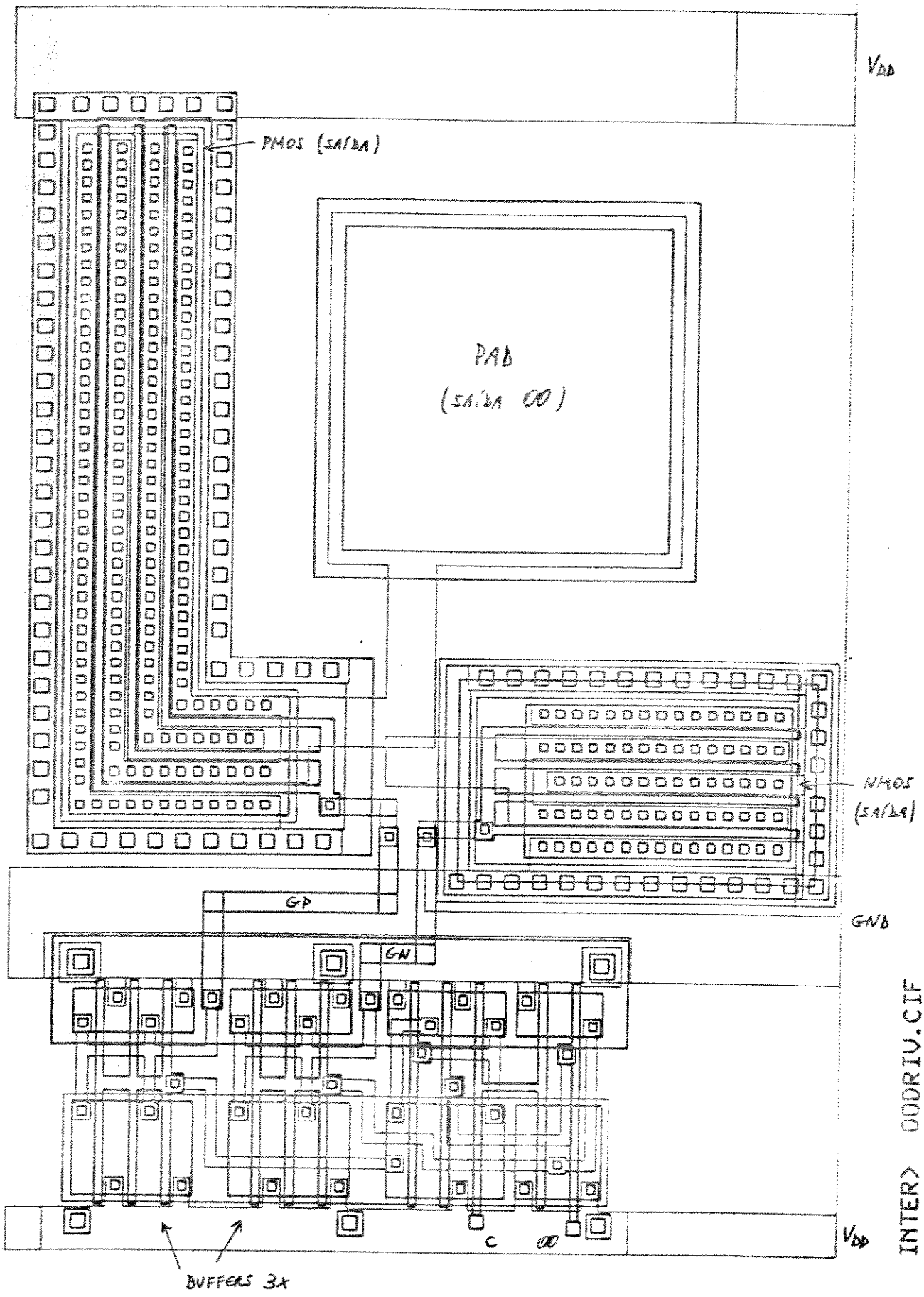
INTER> INPBUFFER.CIF



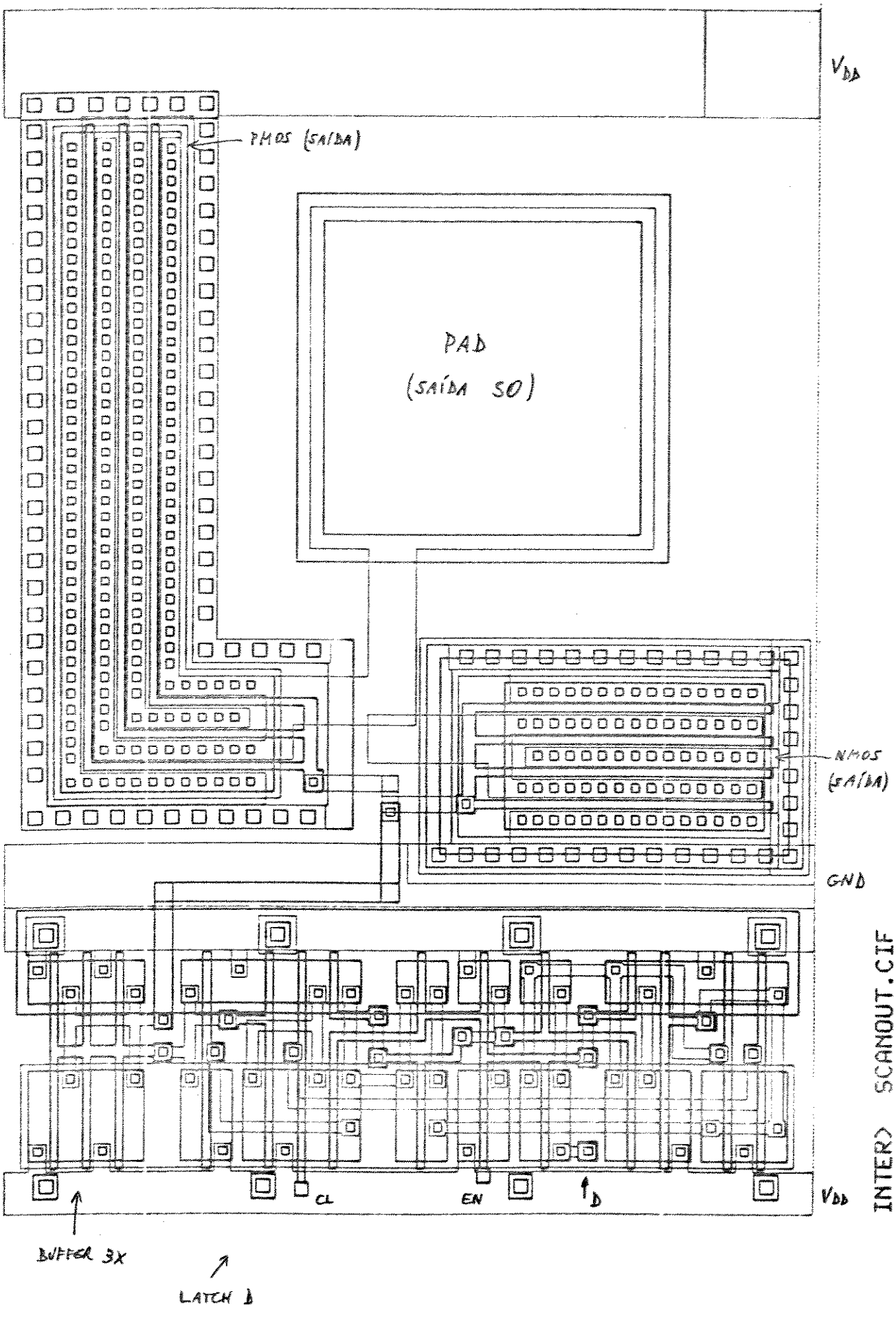
INTER> SINBUF.CIF



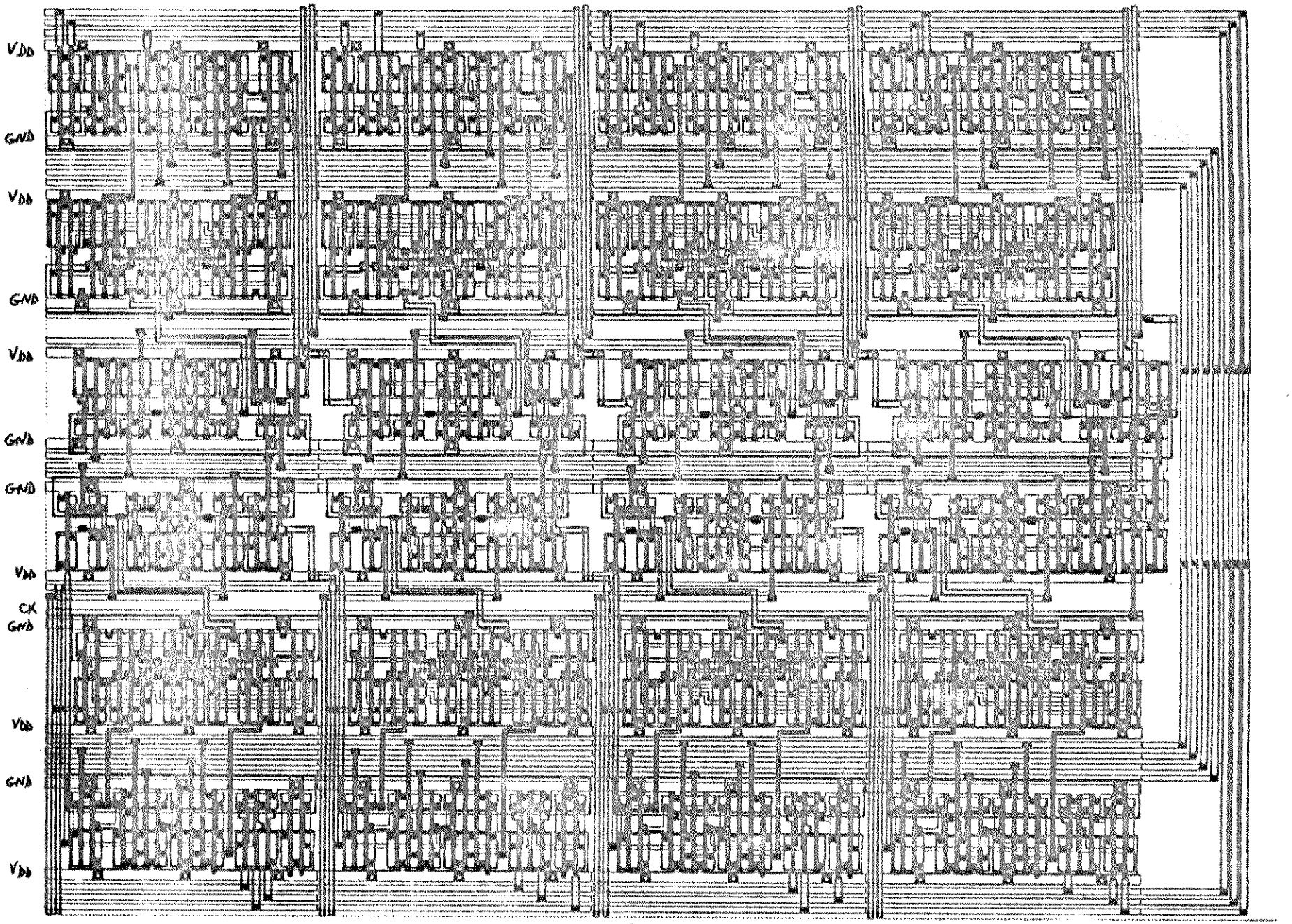
INTER> I03ST.CIF



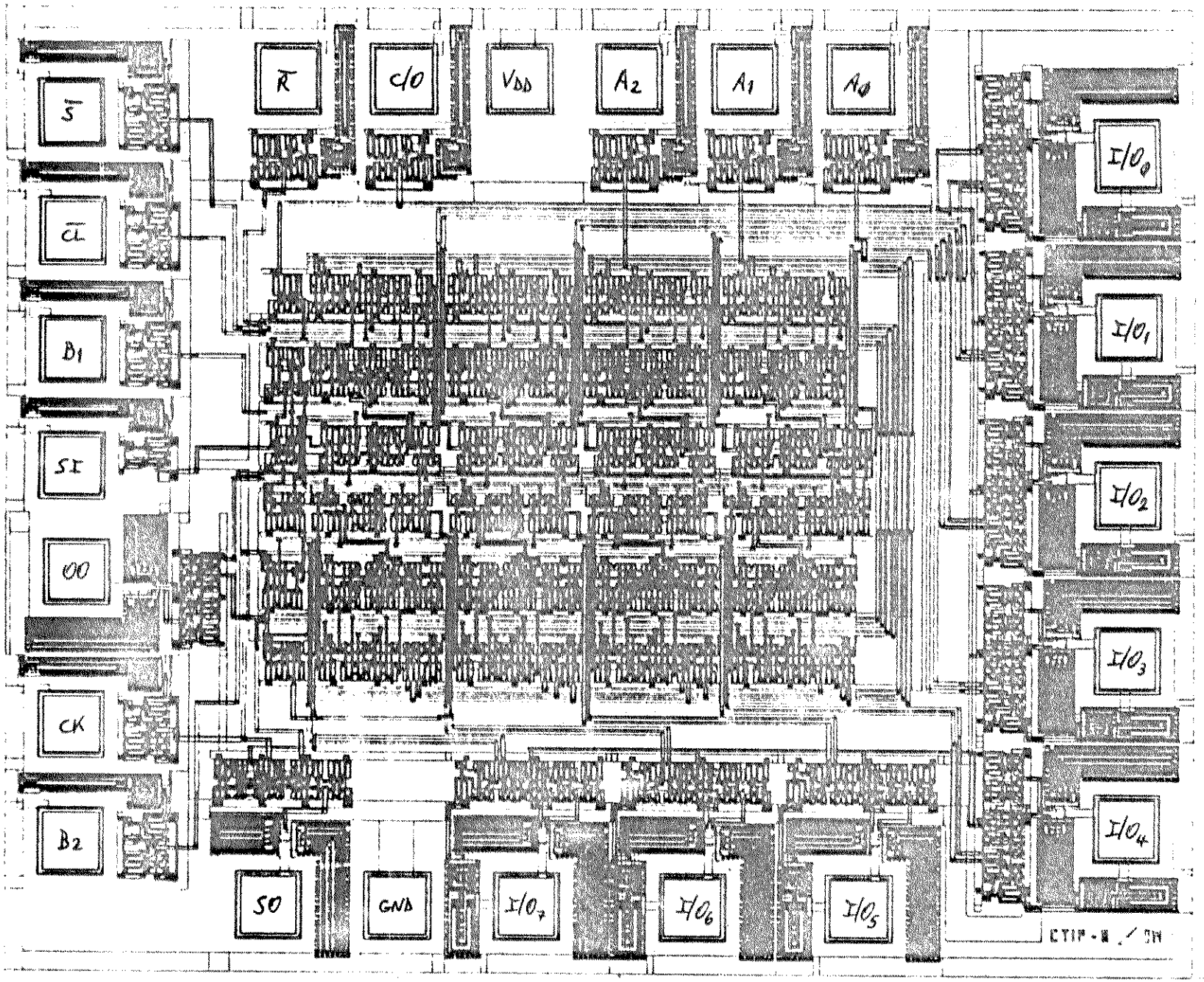
INTER> 00DRIV.CIF



INTER> SCANOUT.CIF



INTER> MACROEXEMPLO.CIF



INTER> CHIPCTIP3E.CIF