

UNIVERSIDADE ESTADUAL DE
CAMPINAS

FACULDADE DE ENGENHARIA ELÉTRICA E DE
COMPUTAÇÃO

PROPOSTA DE UM REGISTRADOR CÍCLICO PARA LÓGICA
MULTI-VALORES E APLICAÇÃO EM UM MULTIPLICADOR
QUATERNÁRIO

Aluno: Osvaldo Hugo Bertone

Orientador: Alberto Martins Jorge

Este exemplar corresponde à redação final da tese defendida por: <u>OSVALDO HUGO BERTONE</u>
e aprovada pela Comissão
Julgada em: <u>27/06/05</u> <u>ALM</u>
Orientador

UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA ELÉTRICA E DE
COMPUTAÇÃO

PROPOSTA DE UM REGISTRADOR CÍCLICO PARA LÓGICA
MULTI-VALORES E APLICAÇÃO EM UM MULTIPLICADOR
QUATERNÁRIO

Aluno: Osvaldo Hugo Bertone

Orientador: Alberto Martins Jorge

Dissertação de mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica

Banca Examinadora:

Alberto Martins Jorge
José Antônio Siqueira Dias
Nivaldo Vicençotto Serran
Elnatan Chagas Ferreira

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA – BAE – UNICAMP

Bertone, Osvaldo Hugo
B462p Proposta de um registrador cíclico para lógica multi-
valores e aplicação em um multiplicador quaternário
/Osvaldo Hugo Bertone. –Campinas, SP: [s.n.], 2005.

Orientador: Alberto Martins Jorge.
Dissertação (Mestrado) – Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Lógica a múltiplos valores. 2. Eletrônica digital. I.
Jorge, Alberto Martins. II. Universidade Estadual de
Campinas. Faculdade de Engenharia Elétrica e de
Computação. III. Título.

Título em Inglês: The cyclical register for MVL circuits (Multi-valued logic) and
quaternary multiplier

Palavras-chave em Inglês: MVL logic, Multi-valued logic, Cyclical register

Área de concentração: Microeletrônica

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: José Antonio Siqueira Dias, Nivaldo Vicençotto Serran e
Elnatan Chagas Ferreira

Data da defesa: 28/06/2005

RESUMO

Neste trabalho é proposto um Registrador Cíclico para circuitos MVL (*Multi-valued Logic*) utilizando transistores NMOS e PMOS para uma configuração de quatro níveis lógicos. Este circuito usa certas características secundárias (normalmente indesejadas) dos transistores MOS.

Uma particularidade deste registrador são os níveis lógicos auto-definidos com uma alta precisão. Isto permite incrementar a Lógica para mais valores, não estando limitada somente a Lógica Ternária ou Quaternária (as mais usadas em circuitos MVLs), seu uso pode ser estendido para Decimal, Hexadecimal ou mais.

O Registrador Cíclico proposto, além de armazenar um dado multi-valor com um nível de tensão preciso, ainda, fornece a saída com qualquer possível deslocamento lógico sem degradação da precisão.

Este registrador permitirá o desenvolvimento de circuitos lógicos como contadores, *toggle switches*, *shift registers*, *flip-flops* em vários níveis, deslocamentos de valores (negação de Post), conversores D/A e A/D, etc...

Algumas vantagens que este circuito oferece é sua alta resposta em frequência e sua pouca dependência dos parâmetros do transistor, alcançando uma robustez comparável com os circuitos binários.

Como uma aplicação deste registrador proposto é apresentado um Multiplicador Quaternário e comparado com um Multiplicador Binário utilizando a mesma tecnologia.

Neste texto serão desenvolvidos os circuitos e simulados no OrCad (PSPice [01]) utilizando um modelo de transistor NMOS e outro PMOS fornecidos pela *foundry AMS (Austria Micro Systems)* descritos no Apêndice I.

O Registrador Cíclico para circuitos MVL foi apresentado pelo autor no Congresso SUCESU 2005 no dia 31 de março de 2005 em Belo Horizonte, MG, Brasil.

ABSTRACT

The Cyclical Register for MVL circuits (Multi-valued Logic) proposed is composed by NMOS and PMOS Transistors. This circuit uses the advantage of certain secondary characteristics (normally undesirable) of the MOS transistors.

One peculiarity of this register is that the logical levels are defined by itself with a very high precision ; this, permits to increase the logic to many values. Since it is not limited to ternary or quaternary logic (more used MVLs), its use can be extended to decimal, hexadecimal and others.

The proposed cyclical register, besides storing the multi-value data with precise voltage level, still, supplies the output with any possible logical shift without the degradation of precision.

This register will allow the development of logical circuits as counter, toggle switch, shift register, flip-flop in several levels, shift of value, D/A and A/D converter, etc...

Some advantages that this circuit offers is its high frequency response and its minor dependency of the parameter of the transistors, providing a robustness comparable to the current binary circuits.

As an application of this proposed Register a Quaternary Multiplier is presented and compared with the Binary Multiplier with the same technology.

On this paper the circuits will be developed and simulated in the OrCad (Pspice [01]), using the transistors models NMOS and PMOS supplied by foundry AMS (Austria Micro Systems) detailed in the Appendix I.

The Cyclical Register for MVL circuits was presented by the author in the Congress SUCESU 2005 in March 31st, 2005 in Belo Horizonte, MG, Brazil.

Agradecimentos:

Ao meu orientador Prof. Alberto Martins Jorge e aos colegas do DEMIC, Luciana Prado do Nascimento, Marco Aurélio Seluque Fregonezi, Carlos Roberto Mingoto Jr. e José Carlos Silva.

Índice

1. Capítulo 1: Álgebra Multi-valores
 - 1.1 Introdução
 - 1.2 Características desejadas
2. Capítulo 2: Circuitos básicos
 - 2.1 Introdução
 - 2.2 Descrição do Registrador
 - 2.3 Funcionamento do Registrador
 - 2.4 Vantagens
 - 2.5 Considerações
 - 2.6 Módulo com detector de nível separado
 - 2.7 Implementação do Módulo
 - 2.8 Registrador Quaternário
3. Capítulo 3: Multiplicador Digital MVL
 - 3.1 Processo de Multiplicação
 - 3.2 Multiplicador
 - 3.3 Descrição de funcionamento
 - 3.4 Seqüência de *Clock*'s
4. Capítulo 4: Multiplicador e Somador
 - 4.1 Tabelas Verdade
 - 4.2 Modulo para o Registrador com saída DECODIFICADA
 - 4.3 Registrador com saída DECODIFICADA
 - 4.4 Tabela Verdade do Multiplicador e do Somador
 - 4.5 Considerações
 - 4.6 Circuito Multiplicador
 - 4.7 Circuito Somador

5. Capítulo 5: Multiplicador Quaternário Completo
 - 5.1 Módulo *Driver*
 - 5.2 Registrador *Driver*
 - 5.3 Circuito de finalização do Multiplicador
 - 5.4 Circuito do Multiplicador
 - 5.4.1 Descrição de funcionamento
 - 5.4.2 Ciclo de Multiplicação

6. Capítulo 6: Comparação com Sistemas Binários e Projeção
 - 6.1 Comparação com Sistemas Binários
 - 6.2 Projeção
 - 6.3 Conclusão

7. Apêndice I
 - I.1 Modelo *Pspice* NMOS
 - I.2 Modelo *Pspice* PMOS

8. Apêndice II
 - II.1 Alternativa de Circuito Multiplicador

9. Glossário

10. Bibliografia

Capítulo 1

Álgebra Multi-valores

1.1 Introdução

Quase tudo o que foi desenvolvido no mundo com relação a computação está baseado em sistemas binários, isto é, um sistema onde somente são reconhecidos dois valores (0 e 1), que são implementados em circuitos eletrônicos definidos normalmente por dois níveis de tensão, mas que podem ser implementados de outras formas. Estes circuitos foram escolhidos especialmente por sua robustez e na medida em que a tecnologia de circuitos integrados foi avançando, estes circuitos foram operando com frequências mais elevadas, dimensões cada vez menores e circuitos integrados em um *chip* cada vez mais complexo. Isto fez com que em um *chip* se tenha mais conexões e apesar dos transistores serem cada vez menores, o espaço que as conexões ocupam no *chip* são maiores em proporção, chegando a [02] [03] 70% da área utilizada.

Uma das soluções para este grande problema foi enviar em cada linha de conexão mais informação que simplesmente um “1 ou 0”; com o conceito de Álgebra de Post (trabalhos apresentados por Post [04] e Lukasiewicz [05] em 1920) de ordem “n” desenvolvida por Rosebloom em 1942 [06] nasce a Lógica Multi-valores, na qual será possível enviar mais níveis em cada linha de conexão, por exemplo, ternário “0,1 e 2” quaternário “0, 1, 2 e 3” e assim por diante, chegar a decimal ou hexadecimal. Dessa maneira se conseguirá reduzir a área ocupada pelas conexões.

Para um histórico sobre a evolução na área de Multi-valores se recomenda o trabalho de Serran [07].

Diversos circuitos foram implementados na última década usando, por exemplo, Josephson, junções semicondutoras [08-09] e Diodos Túnel [10], circuitos em Modo de Corrente [11-12] e CMOS Dinâmico [13-15].

Neste texto será apresentado um conceito diferente aproveitando certas características dos transistores CMOS, definição de níveis precisos pelo próprio circuito, conceito de decodificação dos níveis das linhas de interconexões e síntese de funções através de matrizes simples tiradas diretamente das Tabela Verdade. Não são limitadas somente a lógicas Ternárias ou Quaternárias, mas expansíveis para lógicas de “n” Valores.

Como exemplo será desenvolvido um Multiplicador Digital Quaternário de dois Dígitos Quaternários cada entrada e quatro Dígitos Quaternários de saída, utilizando modelo de transistores fornecidos pela *foundry AMS (Austria Micro Systems)*.

No Apêndice II se desenvolve alternativa de Multiplicador utilizando o conceito de matriz, e usando os modelos de transistores da *foundry AMS (Austria Micro Systems)*, mas mudando o valor da tensão de limiar (V_{TH0}) do transistor PMOS.

1.2 Características desejadas

As características desejadas dos circuitos que foram desenvolvidos neste trabalho são as seguintes:

- a) Robustez.
- b) Baixo consumo de potência.
- c) Níveis de tensão definidos e precisos.
- d) Dimensões mínimas dos componentes.
- e) Alta frequência de operação.
- f) Pouca dependência dos parâmetros dos componentes.
- g) Quantidade mínima de tipos de componentes.
- h) Facilidade de serem acoplados sem sofrer sobrecargas.
- i) Quantidade mínima de componentes por circuito.
- j) Quantidade mínima de conexões.
- k) Circuitos expansíveis para N-valores.
- l) Simplicidade.

No próximo capítulo será introduzido o Registrador Cíclico para circuitos MVL (*Multi-valued Logic*) que será aplicado nos capítulos 3, 4 e 5 ao desenvolvimento de um multiplicador quaternário.

Capítulo 2

Registrador Cíclico

Neste capítulo apresentamos o funcionamento do Registrador Cíclico que será base para a implementação do multiplicador descrito no capítulo 3, utilizando os modelos de transistores NMOS e PMOS descritos no Apêndice I.

2.1 Introdução

O Registrador Cíclico para circuitos MVL (*Multi Value Logic*) é um circuito que aproveita certas características dos transistores MOS. O registrador proposto serve para implementar circuitos em Lógica MVL (Lógica Multi-valor) podendo com este registrador ao mesmo tempo armazenar e decodificar os níveis de tensão.

O Registrador Cíclico é um circuito que armazena um dado, ou seja, um valor de um sistema lógico de numeração, que pode ser Sistema Binário, Ternário, Quaternário, ..., Decimal, Hexadecimal, etc. Este valor é armazenado como em uma memória do tipo RAM, ela permanece no circuito enquanto a tensão de alimentação é mantida e se apaga assim que a tensão é retirada do circuito.

Este Registrador é chamado de “Cíclico” por ter no seu circuito todos os níveis de tensão pertencentes à lógica utilizada no sistema, fechando um *loop* do ponto final do circuito com a entrada.

Para cada entrada de um valor lógico, é gerada uma seqüência crescente ou decrescente dos níveis lógicos até alcançar o valor de entrada que é realimentado e desta forma estabelecer um circuito estável com valores de níveis de tensão definidos pelo próprio circuito.

Um exemplo de um Registrador Cíclico Quaternário no Modo Decrescente está mostrado no diagrama de blocos da Figura 2.1.

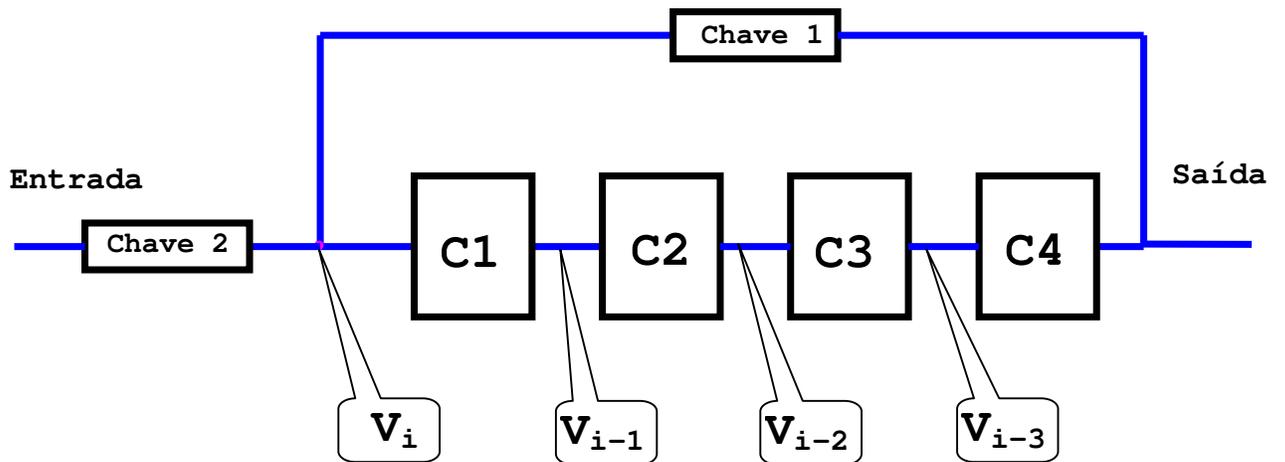


Figura 2.1: Diagrama em bloco de um Registrador Cíclico Quaternário no Modo Decrescente, também chamado de Inversor Cíclico Anti-horário..

Os blocos numerados com C_1 , C_2 , C_3 e C_4 são circuitos que decrescem “um nível lógico” o valor que tem na entrada, até chegar ao nível lógico zero onde o bloco com o nível lógico zero de entrada tem a saída com o nível lógico Máximo e assim continua a seqüência até chegar ao nível de entrada. Os níveis estão designados como V_i (o valor de entrada a ser armazenado) e V_{i-1} , V_{i-2} e V_{i-3} os níveis sucessivamente decrescentes até o mínimo que passa para o máximo e até chegar ao valor de entrada. A “chave 1” fecha a *loop* e a “chave 2” abre, e o circuito se estabiliza definindo seus próprios níveis.

2.2 Descrição do Registrador

Montam-se módulos formados por três transistores (mostrados na Figura 2.2 como NUP, NDN e PUP), estes módulos são conectados em cascata tantas vezes como valores tem a lógica a ser usada. Depois se conecta a saída do último módulo com a entrada do primeiro através de uma chave.

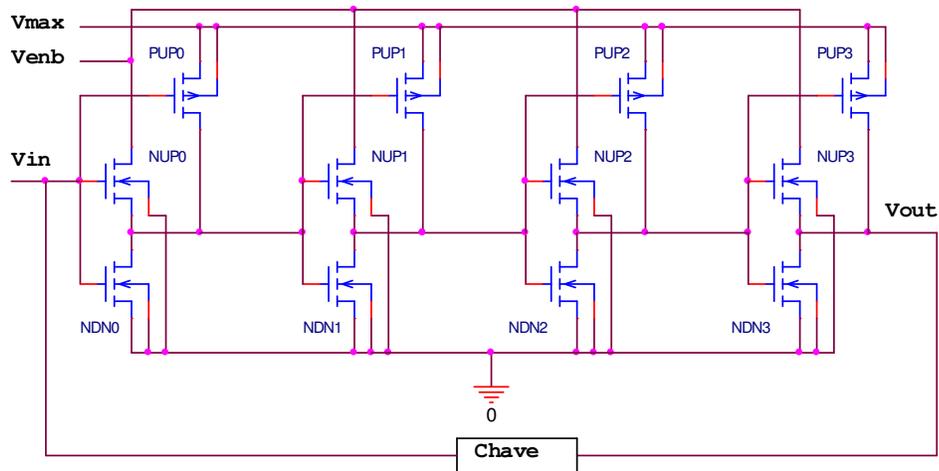


Figura 2.2: Registrador/Decodificador Quaternário, ou “Inversor Cíclico anti-horário”.

2.3 Funcionamento do Registrador

Para esta análise, partimos de um circuito básico formado por dois transistores NMOS (pode ser utilizado também transistores PMOS) como se pode ver na Figura 2.3.

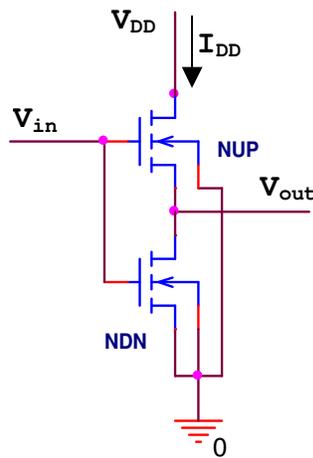


Figura 2.3:Circuito básico

Primeiramente faremos uma análise estática. Se aplicarmos uma tensão na “Porta” dos transistores (V_{IN}) terá uma tensão de saída (V_{OUT}) que será um valor menor que o valor de entrada.

A corrente da “Fonte” (I_{DD}) será igual para os dois transistores e a tensão de saída dependerá das dimensões dos mesmos, características e especialmente de sua tensão de limiar (V_{TH0}).

Fazendo a simulação no OrCad (PSpice) [01] obteremos uma resposta na saída como indicado na Figura 2.4, onde foi utilizada uma tensão de $V_{DD} = 3V$ e dimensões iguais para os dois transistores.

$L=0,3\mu m$ (Comprimento do canal) e $W=0,6\mu m$ (Largura do canal).

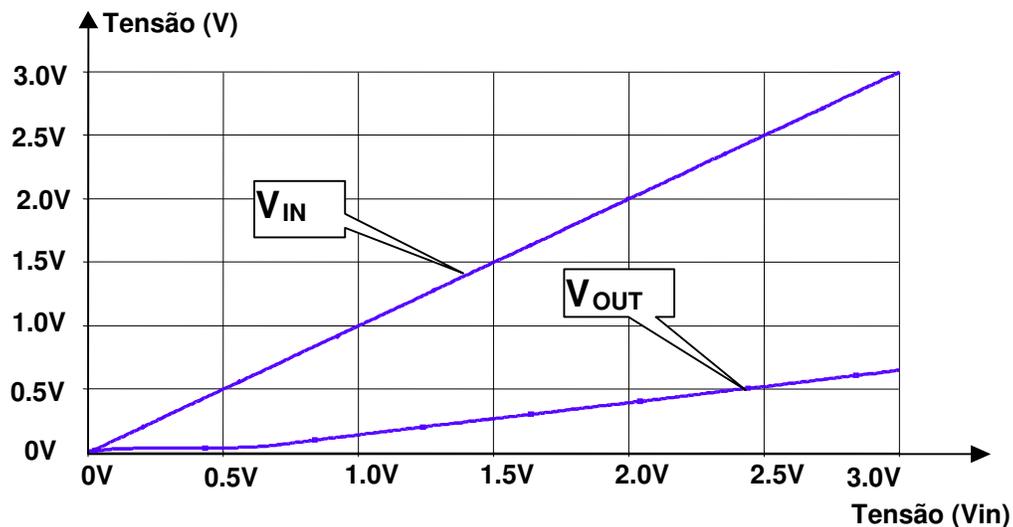


Figura 2.4: Curva de entrada e saída do Circuito básico.

A tensão de saída é muito menor que a tensão de entrada como se pode verificar na Figura 2.4. Isto não é interessante para o circuito proposto, o que necessitamos é uma tensão de saída *um pouco* menor que a tensão de entrada.

Para se obter uma tensão de saída *um pouco* menor que a tensão de entrada, devemos aumentar o comprimento do canal do transistor inferior e aumentar a largura do canal do transistor superior, por exemplo:

Transistor inferior: $L=3\mu m$ e $W=0,6\mu m$ e Transistor superior: $L=0,3\mu m$ e $W=3\mu m$

Com estas dimensões de transistores obtemos as curvas da Figura 2.5.

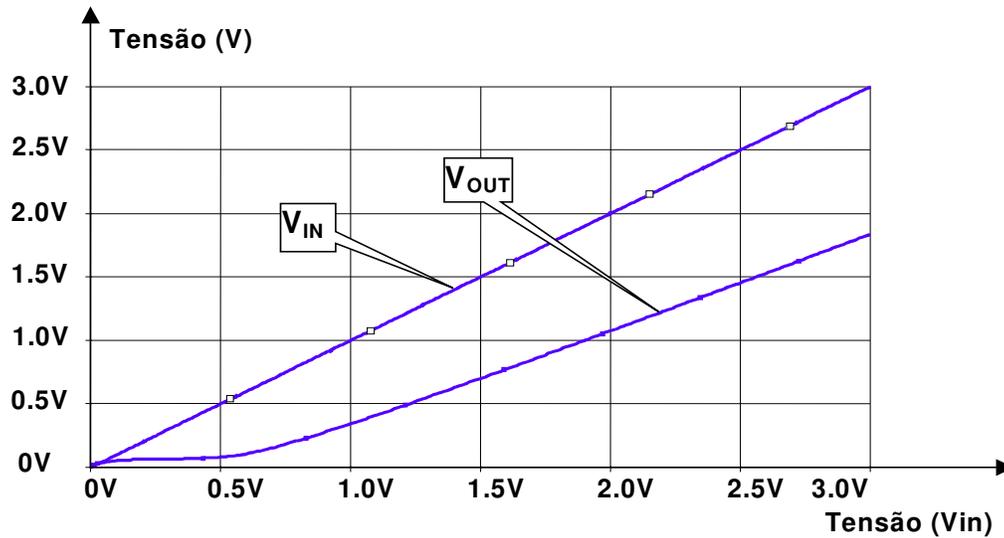


Figura 2.5: Curva de entrada e saída do Circuito básico com transistor inferior $L=3\mu\text{m}$; $W=0,6\mu\text{m}$ e transistor superior $L=0,3\mu\text{m}$; $W=3\mu\text{m}$..

Assim é possível fazer um registrador como o proposto na Figura 2.1, mas não é interessante utilizar transistores com dimensões grandes, pois isto aumentará o tamanho do *chip* e reduzirá a frequência máxima de operação devido ao aumento das capacitâncias e resistências dos componentes por serem maiores.

A continuação, podemos verificar o efeito do aumento da tensão de Dreno-Substrato nas curvas de saída, para isso utilizamos dois transistores iguais e com canal longo de $L=3\mu\text{m}$ e variamos parametricamente a tensão Dreno-Substrato (V_{DD}) de 3V ate 10V. Obtemos as curvas da Figura 2.6.

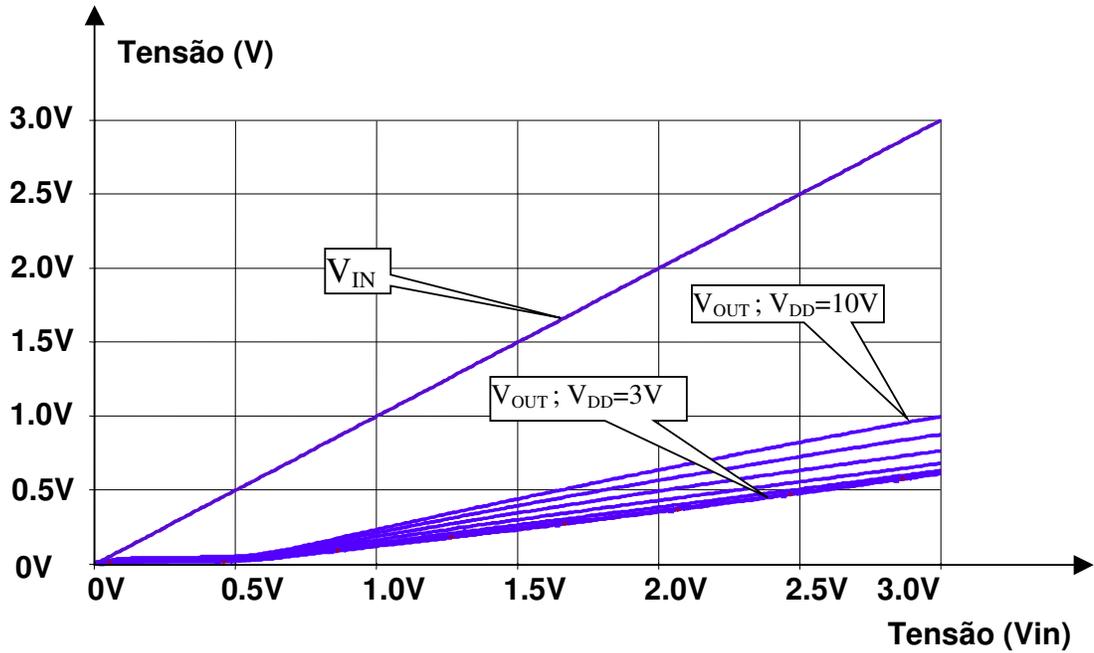


Figura 2.6: Curva de entrada e saída do Circuito básico com transistor inferior e superior iguais $L=3\mu\text{m}$; $W=0,6\mu\text{m}$ variando V_{DD} de 3V ate 10V.

Pode-se notar que na medida em que a tensão de Dreno-Substrato aumenta a curva de saída se aproxima à de entrada. Este efeito é explicado em detalhe no livro de “Yannis Tsividis, Chapter 6” [16] aqui daremos um resumo do efeito.

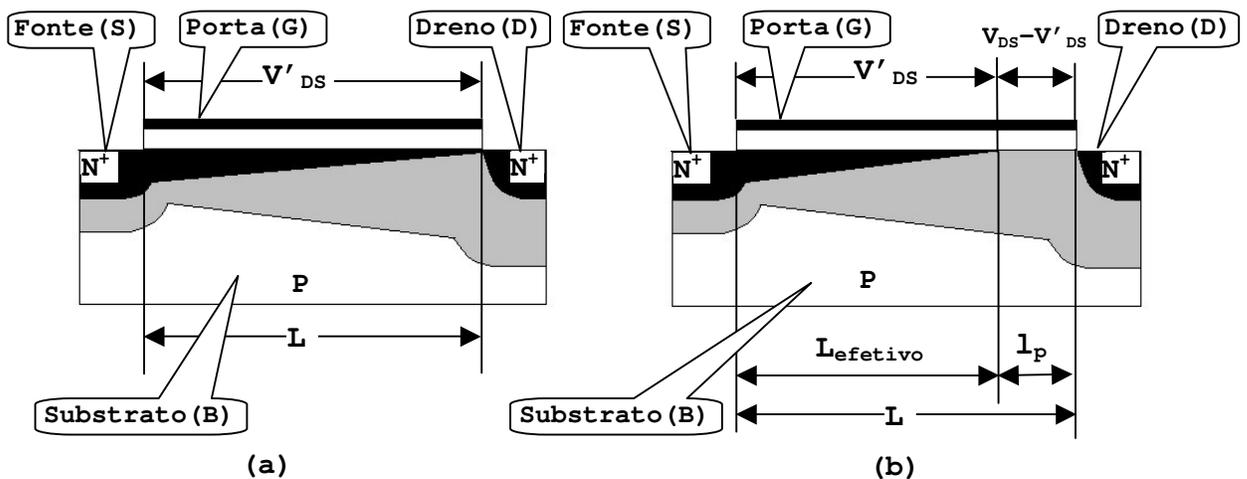


Figura 2.7: (a) Transistor com $V_{DS} = V'_{DS}$ (“pinchoff”), (b) Transistor com $V_{DS} > V'_{DS}$ (“pinchoff”).

A Figura 2.7(a) mostra o transistor tipo N onde a tensão V_{DS} é igual a V'_{DS} ($V_{DS}=V'_{DS}$) sendo V'_{DS} o valor no qual “*pinchoff*” acontece no extremo do Dreno no canal. Agora, se incrementamos V_{DS} acima de V'_{DS} o “*pinchoff*” acontece antes do fim do canal como mostra a Figura 2.7(b). A diferença de tensão entre o valor aplicado V_{DS} e o valor de “*pinchoff*” (V'_{DS}) cairá entre o ponto de “*pinchoff*” e o Dreno na região de Depleção de comprimento l_p , na medida em que a tensão V_{DS} aumenta, a região l_p também aumentará diminuindo desta forma o comprimento $L_{efetivo}$ do transistor. Este efeito é chamado de Modulação do comprimento do canal (*CLM- Channel Length Modulation*).

O efeito de Modulação do comprimento do canal ocorre somente no transistor superior do Circuito Básico, porque a tensão no Dreno do transistor inferior nunca supera o valor de entrada, portanto, nunca acontece o efeito de Modulação do comprimento de canal no transistor inferior. Desta forma o transistor superior terá um comprimento efetivo de canal menor na medida em que a tensão V_{DD} aumenta produzindo desta forma o resultado obtido pelo circuito da Figura 2.5, onde foram utilizados transistores com dimensões de canal diferentes. Isto explica porque obtemos curvas mais próximas do valor de entrada na medida em que aumentamos a tensão V_{DD} do circuito.

Utilizando transistores com canal longo o efeito não é ainda suficiente para nosso circuito, portanto agora experimentaremos transistores com canal curto, utilizando-os na suas dimensões mínimas, desta forma se obtém os resultados da Figura 2.8.

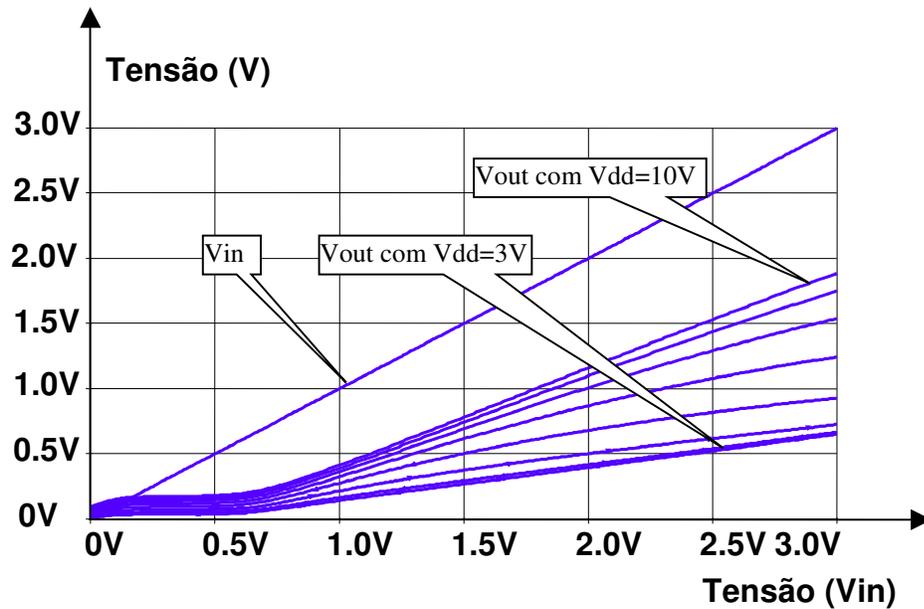


Figura 2.8: Curva de entrada e saída do Circuito básico com transistor inferior e superior iguais com canal curto $L=0,3\mu\text{m}$; $W=0,6\mu\text{m}$ variando V_{DD} de 3V ate 10V.

As simplificações feitas na análise de transistores de canal longo (Figura 2.9), não são válidas para canal curto, portanto, os efeitos da distorção do campo elétrico no canal próximo às extremidades devem ser considerados, isto está bem explicado no livro de Yannis Tsividis [16] que mostra o efeito de canal curto (Figura 2.10) denominado como “*effective threshold voltage*” no qual a medida que o comprimento efetivo do canal diminui a tensão V_{TH0} de limiar do transistor também diminui, provocando o efeito mostrado na Figura 2.8.

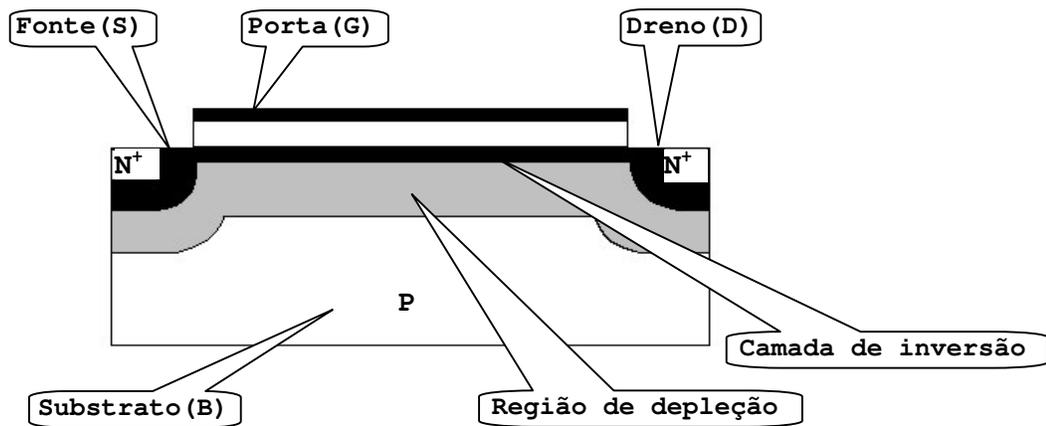


Figura 2.9: Transistor com canal longo.

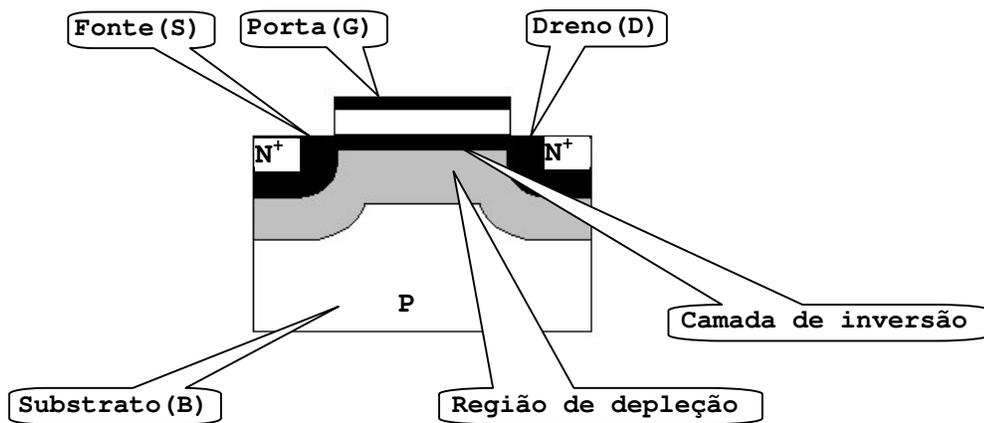


Figura 2.10: Transistor com canal curto.

O efeito da redução da tensão de limiar V_{TH0} pode-se ver na curva de $I_{DS} \times V_{GS}$ da Figura 2.11.

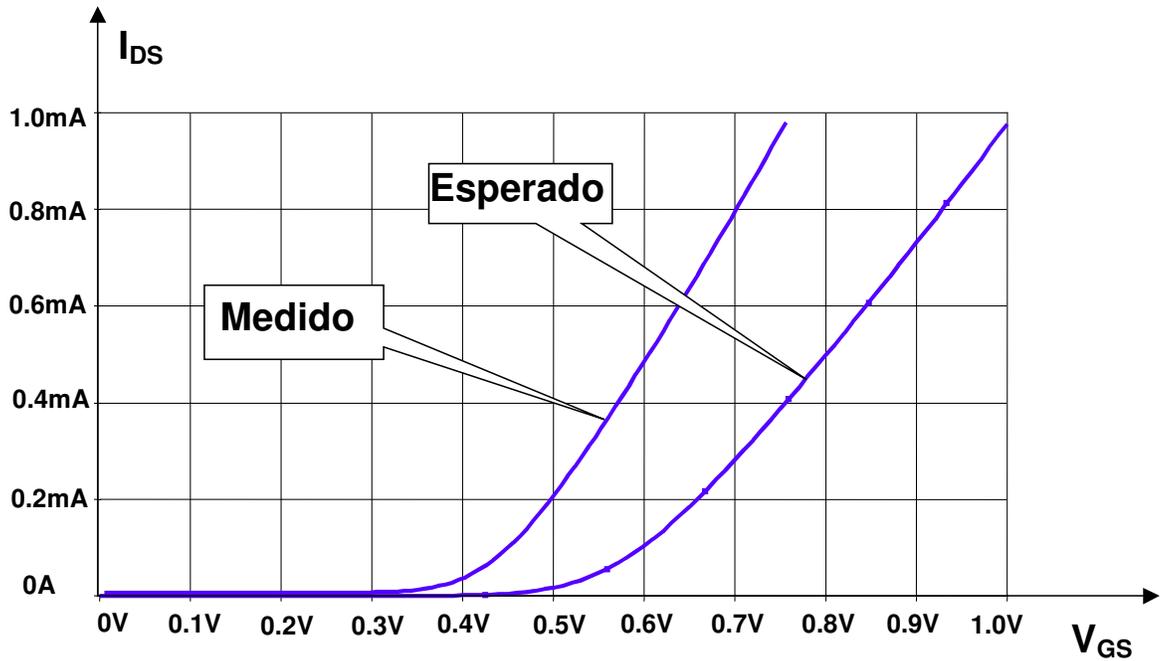


Figura 2.11: Curvas de I_{DS} x V_{GS} de um transistor com canal curto, curva prevista sem considerar os efeitos das extremidades (Esperado) e curva medida (Medido) [16]

Para completar o circuito acrescentamos um transistor PMOS (PUP), que terá a função de chavear a tensão mínima de entrada para o valor do nível máximo. Desta forma, completamos o módulo como pode ser visto na Figura 2.12.

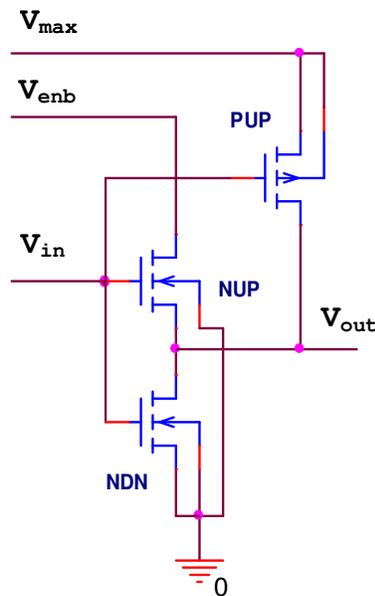


Figura 2.12: Módulo básico tipo NMOS

Na Figura 2.13 pode-se ver a curva de saída e entrada simulada no OrCad (PSpice) [01], os níveis indicados com 0,1,2 e 3 correspondem aos níveis lógicos e são definidos com precisão pelo próprio circuito.

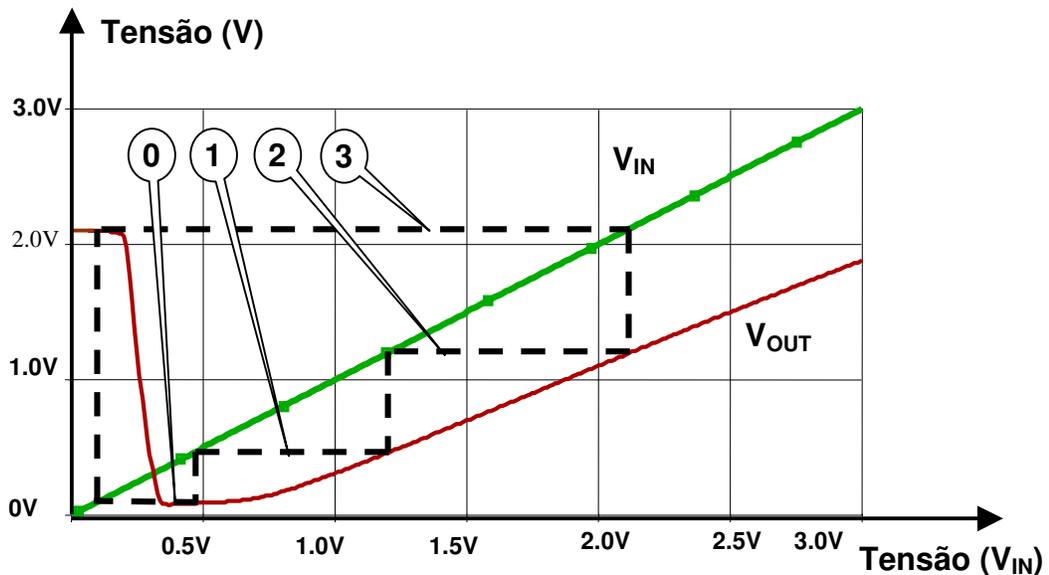


Figura 2.13: Curva de Entrada e Saída do Módulo

Analisando as curvas de entrada e saída, e definindo um valor de V_{max} que corresponde ao nível lógico 3, pode-se obter o traçado dos níveis. Para o valor máximo (3) de entrada se terá na saída um valor intermediário (2), se esse valor for entrada de outro módulo, a saída será outro valor menor, e assim sucessivamente até chegar ao valor inferior onde se terá na saída o valor máximo.

Ajustando os valores de V_{enb} e V_{max} é possível chegar a um funcionamento ótimo do circuito gerando níveis de tensão precisos, como pode-se ver na Figura 2.13.

Uma simplificação adicional poderá ser feita se juntamos os dois transistores NMOS (NUP e NDN) num só componente de quatro terminais, como se pode ver na Figura 2.14.

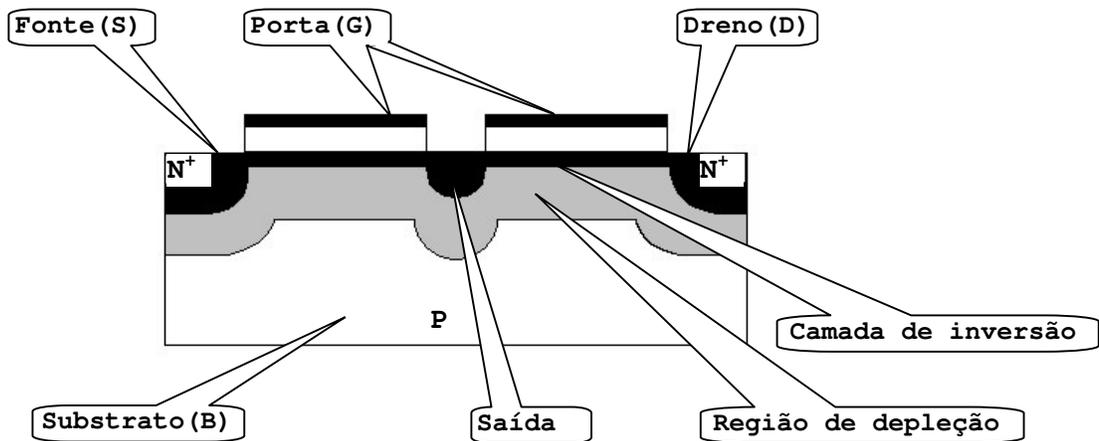


Figura 2.14: Componente Integrado de dois transistores NMOS.

Para concluir, podemos dizer que da mesma forma que foi descrito este Registrador Decrescente ou “Inversor Cíclico Anti-horário” poderá ser feito o “Inversor Cíclico horário” utilizando dois transistores PMOS e um NMOS.

Agora, podemos fazer uma análise dinâmica de funcionamento do Registrador Cíclico exposto neste texto.

Definindo estaticamente os níveis lógicos de tensão, alimentamos o Registrador (mostrado na Figura 2.2) com um sinal que forneça estes níveis lógicos.

Na Figura 2.15 temos uma simulação estática feita no OrCad (Pspice) [01] do circuito da Figura 2.2 utilizando transistores com tecnologia $0,35\mu\text{m}$ (ver modelos no apêndice I) e $V_{\text{TH0}}=0,4655\text{V}$ para o NMOS e $V_{\text{TH0}}=-1,6\text{V}$ para o PMOS, com dimensões de $L=0,3\mu\text{m}$ e $W=0,6\mu\text{m}$ que são as mínimas utilizadas pelo modelo.

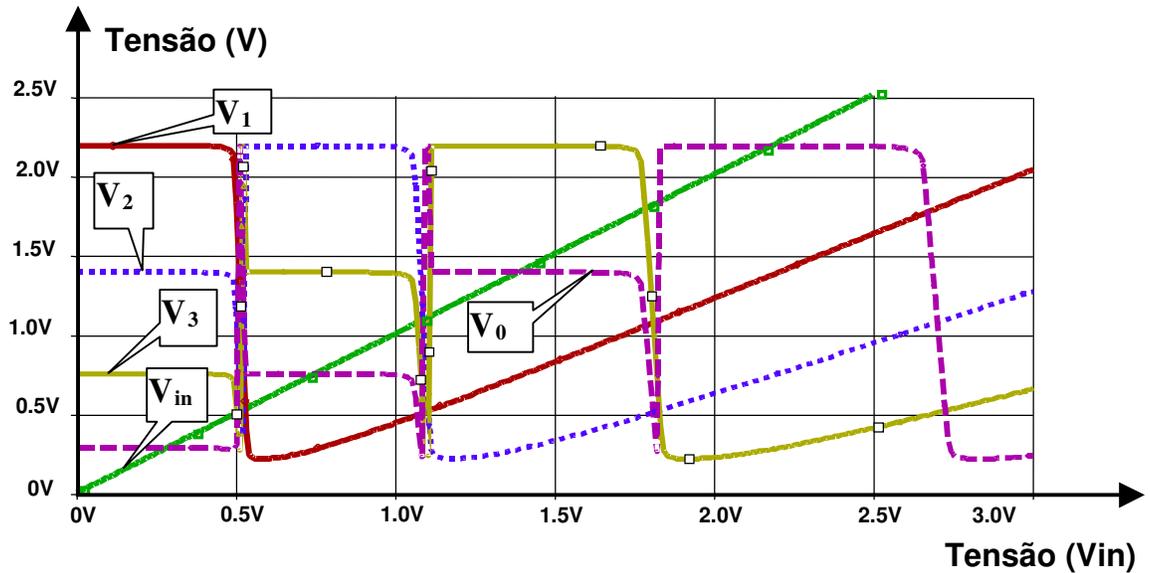


Figura 2.15: Curvas de resposta do Circuito da Figura 2.2 em forma estática, aplicando na entrada (V_{in}) um sinal de varredura de 0 a 3V.

Da Figura 2.15 podemos definir os níveis lógicos, que neste caso são:

$$V_0 = 0,293 \text{ V}$$

$$V_1 = 0,761 \text{ V}$$

$$V_2 = 1,403 \text{ V}$$

$$V_3 = 2,200 \text{ V}$$

Com estes níveis podemos aplicar um sinal discreto com os quatro níveis e verificar o comportamento dinâmico. Esta simulação é mostrada na Figura 2.16.

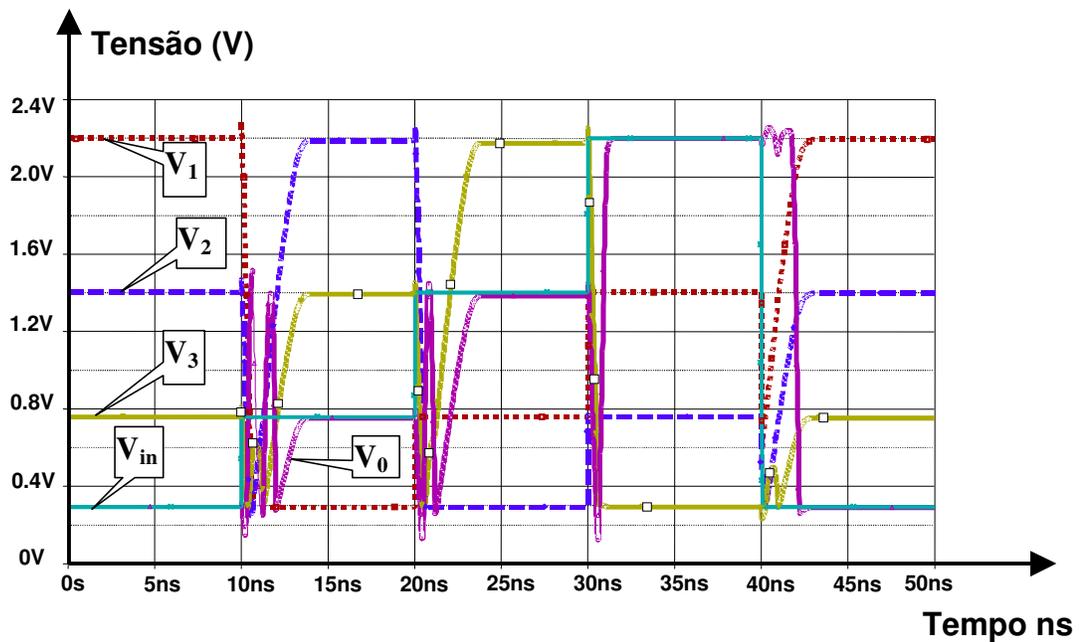


Figura 2.16: Curvas de resposta do Circuito da Figura 2.2 em forma dinâmica, aplicando na entrada (V_{in}) um sinal com os níveis lógicos.

Da análise das curvas é possível deduzir que os níveis são perfeitamente definidos pelo próprio circuito e que o tempo de *setup* do Registrador é, neste caso, de 1 a 3 nanossegundos. O tempo de *setup* pode ser otimizado com desenhos de transistores especialmente desenvolvidos para esta aplicação.

2.4 Vantagens

È possível destacar uma série de vantagens na utilização deste circuito, que são as seguintes:

- a) Níveis de tensões definidos pelo próprio circuito e precisos.

No momento de armazenar um dado em um Registrador Cíclico como o descrito neste texto na Figura 2.1, a “chave 2” estará fechada e a “chave 1” aberta, e o circuito terá as tensões de cada estágio dependendo da tensão de entrada (não necessariamente essa tensão de entrada terá um valor preciso, estará dentro de certa tolerância), depois, será conectada a “chave 1” e aberta a “chave 2”, então, o circuito buscará um ponto de estabilização definindo suas próprias tensões, e restabelecendo, ou restaurando, qualquer degradação que possa ter acontecido nas etapas anteriores.

- b) Dimensões mínimas dos componentes.

Sempre serão utilizados os componentes em suas dimensões mínimas, porque quanto menor for a dimensão da porta, menor será a capacitância de entrada do Registrador, obtendo-se uma constante de tempo de carga menor. E, quanto menor for o comprimento do canal “L” , maior será o efeito de canal curto, o que fará que utilizemos uma tensão V_{ENB} de alimentação da fonte menor para obter o ponto ótimo de funcionamento.

- c) Alta frequência de operação.

È possível demonstrar que estes circuitos são tão rápidos quanto os circuitos binários. Por ser utilizados sempre transistores em suas dimensões mínimas, eles possuem capacitâncias mínimas, fazendo que possam trabalhar a frequências da mesma ordem que os circuitos binários que utilizem os mesmos transistores. Esta comparação é feita no Capítulo 6.

d) Pouca dependência dos parâmetros dos componentes.

Como foi possível ver na explicação deste texto, não se fez nenhuma restrição dos parâmetros, a não ser o valor de V_{TH0} que para o transistor NMOS deve procurar-se um valor baixo e para o PMOS utilizar um valor alto, para obter a quantidade de níveis lógicos de tensão para a lógica MVL que queira ser desenvolvida.

e) Facilidade de serem acoplados sem sofrer sobrecargas.

Esta é uma das características importantes deste circuito, devido ao fato de que a impedância de entrada do circuito é muito alta por ser conectado somente nas Portas dos transistores NMOS e PMOS, e nunca nas Fontes ou Drenos, não existe corrente permanente de *setup* do registrador, portanto, o estágio anterior não será sobrecarregado e não sofrerá degradação do nível de tensão com o aumento do *Fan-out*. O *Fan-out* afetará o retardo do *setup* mas não o nível de tensão.

f) Registrador e Decodificador em um só circuito.

Esta propriedade é muito interessante, já que tendo uma saída decodificada, poderá ser utilizado qualquer dos valores de saída, podendo, por exemplo, ter uma saída com um deslocamento do valor de entrada para um, dois ou mais níveis.

g) Sintetizar funções através de matrizes utilizando as saídas decodificadas.

Esta é uma outra aplicação possível da saída decodificada, poderá ser utilizado um esquema da forma Entrada-Quaternária/Decodificador/Matriz/Codificador/Saída-Quaternária, de forma a implementar qualquer função definida por uma Tabela Verdade.

h) Quantidade mínima de tipos de componentes.

Utiliza somente dois tipos de transistores e com dimensões iguais para cada um, o que facilita grandemente o projeto e seu processo de fabricação.

i) Circuitos expansíveis para N-valores.

O Registrador Cíclico pode ser expandido para qualquer número de níveis na medida que se desenvolva componentes que suportem maior número de níveis, isto é utilizando cada vez um transistor NMOS com um V_{TH0} cada vez menor e um transistor PMOS com um V_{TH0} cada vez maior.

j) Facilidade de ajuste para um ponto ótimo de funcionamento.

Em circuitos com muitos componentes como Processadores e DSP's sempre poderá ser possível ter um circuito que defina dinamicamente (compensando variações de processos e temperatura) os níveis de tensão de V_{ENB} e V_{MAX} para um ponto de funcionamento ótimo do circuito, e estes níveis serem referência de fontes de alimentação do componente.

k) Robustez.

Neste item o circuito tem a seu favor os níveis de tensão precisos definidos por ele próprio e a possibilidade de ser ajustado no seu ponto ótimo de funcionamento. Deverá, por certo, verificar-se para cada caso o nível de Robustez do projeto tendo em conta os níveis de ruído máximos, variações nos processos de fabricação, etc.

l) Simplicidade.

O projeto de circuitos elaborados através destes Registradores poderá ser automatizado utilizando a característica do item g) ou seja, utilizando qualquer álgebra para definir uma determinada função e chegando a uma equação final que se transforma em uma ou mais Tabelas Verdade, estas tabelas poderão ser sintetizadas automaticamente e desenvolvido o circuito por complexo que seja em forma computadorizada, tendo uma redução no tempo de desenvolvimento do produto, e como consequência redução no tempo de lançamento no mercado.

2.5 Considerações

A grande desvantagem destes circuitos (comparado com os circuitos binários que utilizam um par complementar, um PMOS e um NMOS) é que eles têm um consumo de corrente permanente. Isto foi resolvido em parte com a substituição da linha V_{DD} pôr uma linha *Venb* (Tensão *Enable*) que alimenta o circuito no momento que este seja utilizado.

2.6 Módulo com detector de nível separado

Neste trabalho, utilizamos os transistores definidos no apêndice I, que não são adequados para um circuito lógico quaternário especialmente porque a tensão de limiar do transistor PMOS é muito pequena ($V_{TH0} = -0,617$ V). Nas simulações até agora, usamos $V_{TH0} = -1,6$ V para poder mostrar um registrador quaternário.

Com a finalidade de utilizar os transistores do modelo real (Apêndice I) em um circuito de lógica quaternária, faremos uma modificação no módulo, sendo que esta modificação não será necessária no caso de utilizar transistores com as características desejadas.

Utilizamos o módulo da Figura 2.12 e acrescentamos dois inversores formados por quatro transistores que fazem a detecção e chaveamento do nível inferior para o nível superior, o circuito e sua simulação são mostrados nas Figura 2.17, 2.18 e 2.19.

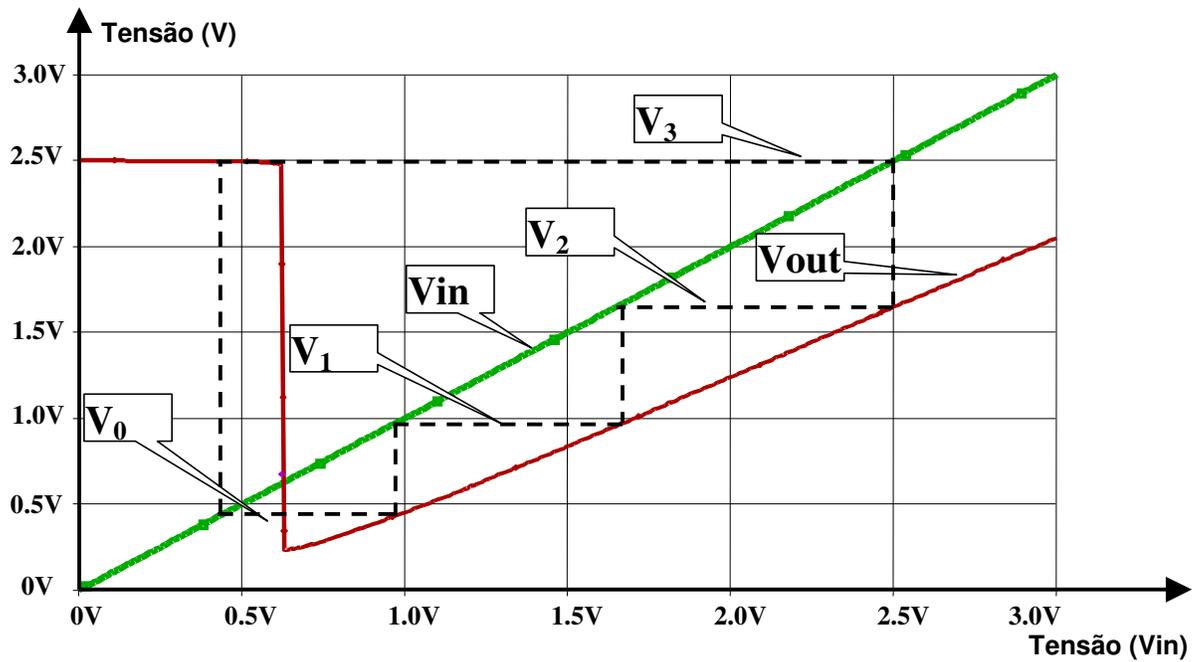


Figura 2.18 Curva de Entrada e Saída do circuito decrescente com transistores tipo NMOS com circuito separado para a detecção do nível de comutação.

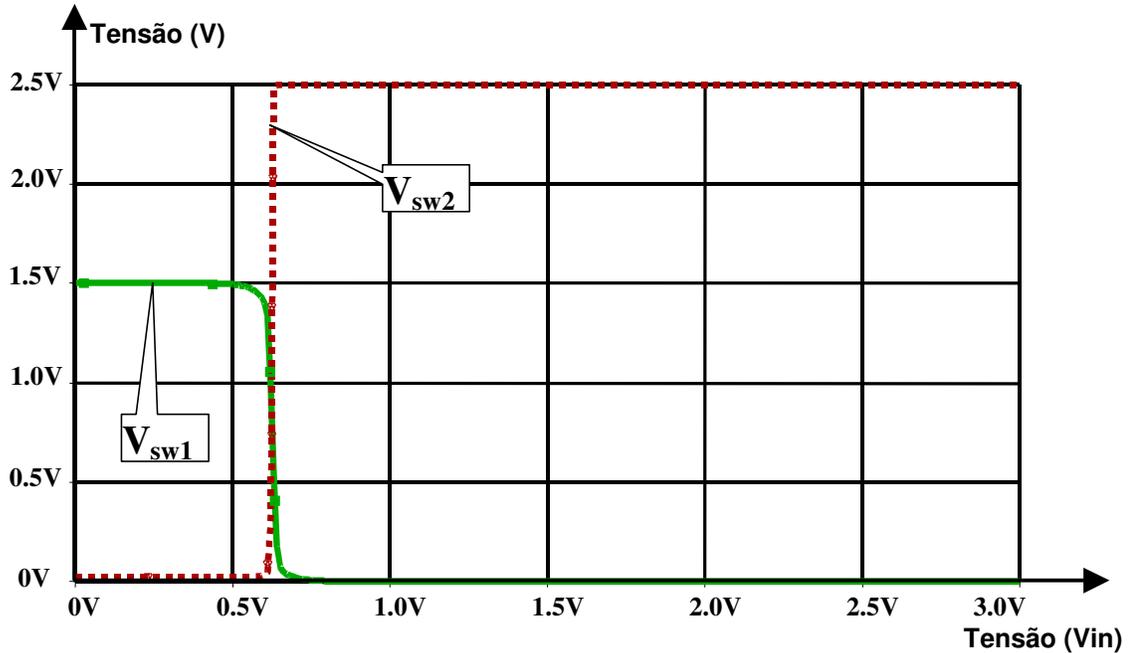


Figura 2.19 Curva do Circuito detector de nível do circuito decrescente com transistores tipo NMOS com circuito separado para a detecção do nível de comutação.

Este módulo é um circuito decrescente com transistores tipo NMOS chamado de “Inversor cíclico anti-horário” [16], onde é utilizado um circuito separado para a detecção do nível de comutação, neste caso do nível mínimo. Este circuito utiliza um número maior de transistores, mas permite a expansão para um número maior de níveis e tem uma robustez acentuada.

2.7 Implementação do Módulo

Foi criado um bloco no OrCad (Pspice) [01] do módulo denominado MODULO_N

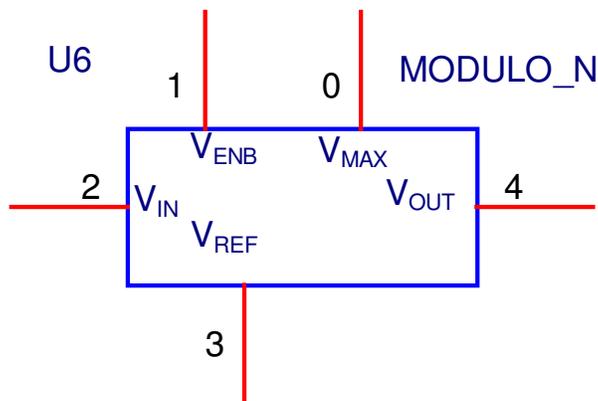


Figura 2.20 Bloco Módulo

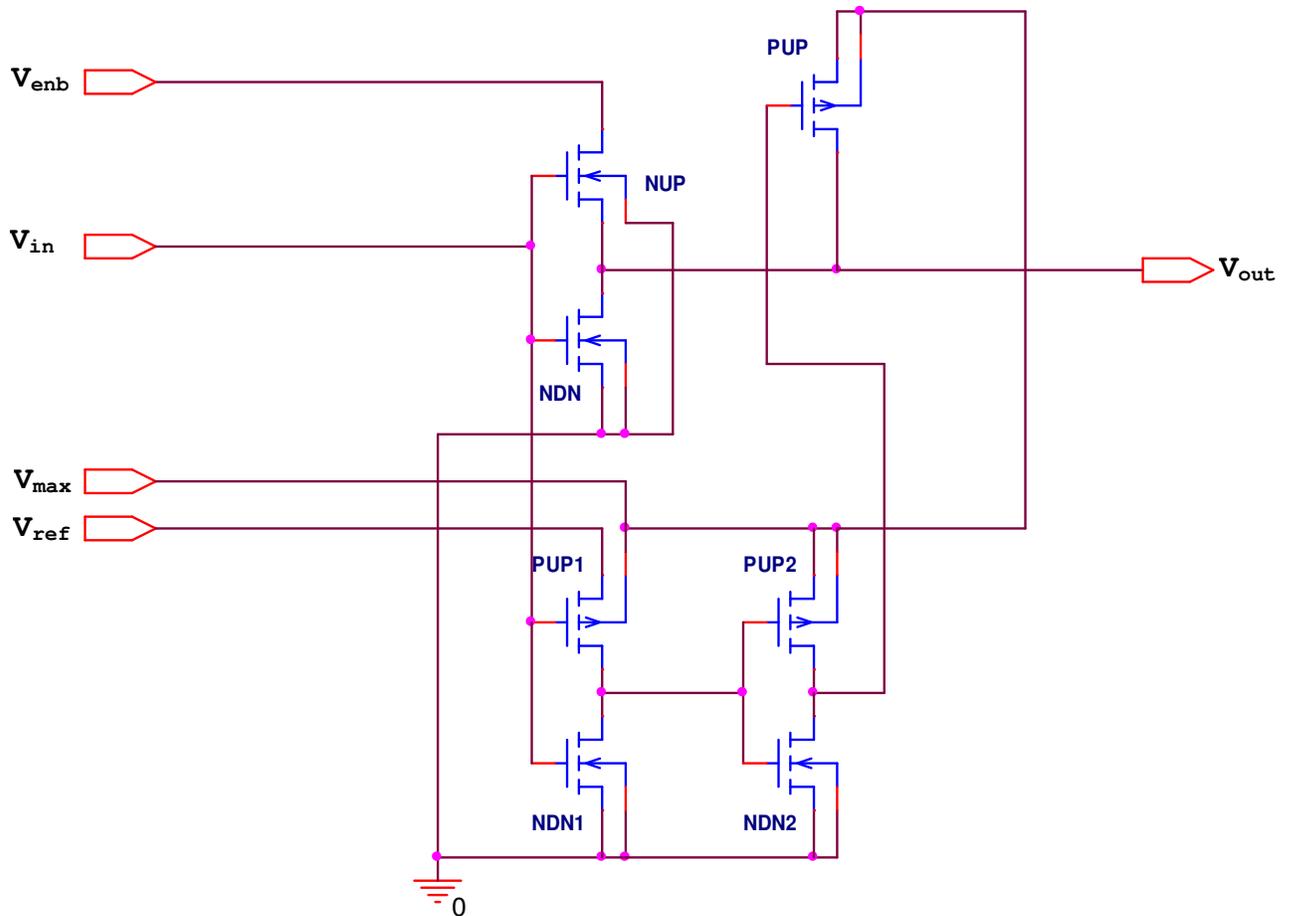


Figura 2.21 Circuito do Módulo

Os níveis de tensão utilizados nesta simulação são os seguintes:

$$V_{enb} = 12V$$

$$V_{ref} = 1.5V$$

$$V_{max} = 2.5V$$

O valor de $V_{enb} = 12V$ é utilizado para forçar o efeito de canal curto, a tensão $V_{ref} = 1.5V$ permite que o primeiro inversor (P_{UP1} e N_{DN1}) comute entre o nível lógico 0 e o nível lógico 1.

Com este circuito foram obtidos os seguintes níveis de tensão:

$$\text{nível "0"} = 0,421V$$

$$\text{nível "1"} = 0,956V$$

nível “2” = 1,647V

nível “3” = 2,500V

Estes níveis são precisos com uma variação de alguns milivolts, que deverão ser verificados na prática em medições de circuitos reais. Também é verdadeiro que para cada chip, lote de produção ou processo poderá ocorrer tensões diferentes e um ponto ótimo de funcionamento. Isto pode ser ajustado com um circuito que verifique o ponto ótimo do chip e ajuste em forma dinâmica as tensões de referência e máxima para os valores adequados, podendo também compensar as variações de temperatura.

2.8 Registrador Quaternário

Conectando os módulos em cascata e interconectando a saída do último módulo com a entrada do primeiro módulo através de uma chave poder-se-á criar um registrador cíclico. Quando se aplica uma tensão na entrada com a chave aberta, este adquirirá níveis de tensões em todos os módulos até chegar no nível mínimo onde comutará para o máximo, o qual definirá as tensões do circuito nos módulos seguintes. No meio ciclo seguinte, quando se desconecta a entrada e se fecha o *loop*, o circuito terá as tensões definidas com precisão, ficando este valor armazenado como uma memória.

Foi criado um bloco do registrador quaternário denominado `REGISTRADOR_QUAT_N`

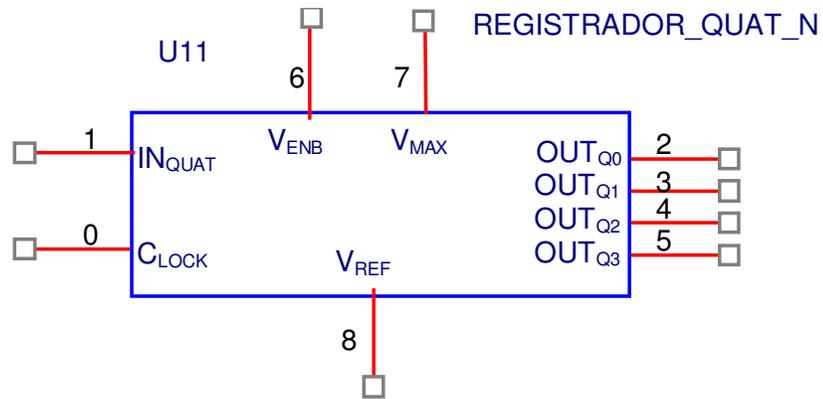


Figura 2.22 Bloco do Registrador Quaternário tipo N.

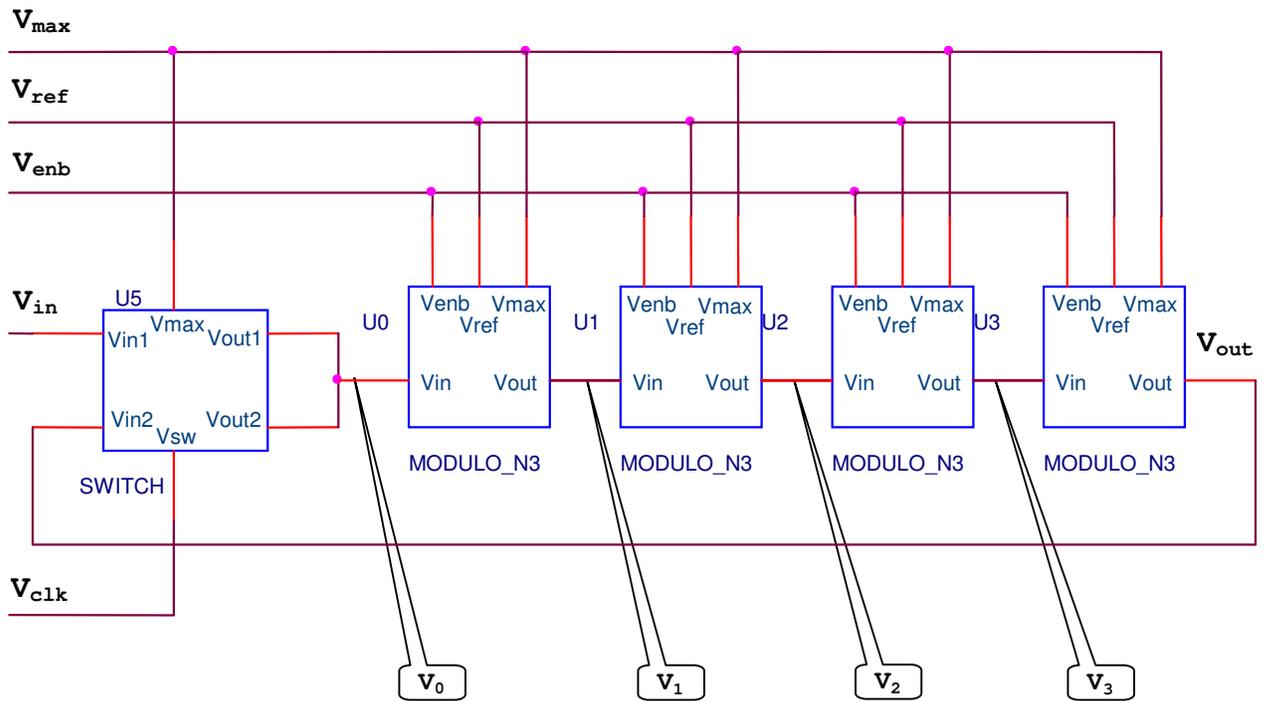


Figura 2.23 Circuito do Registrador quaternário tipo N.

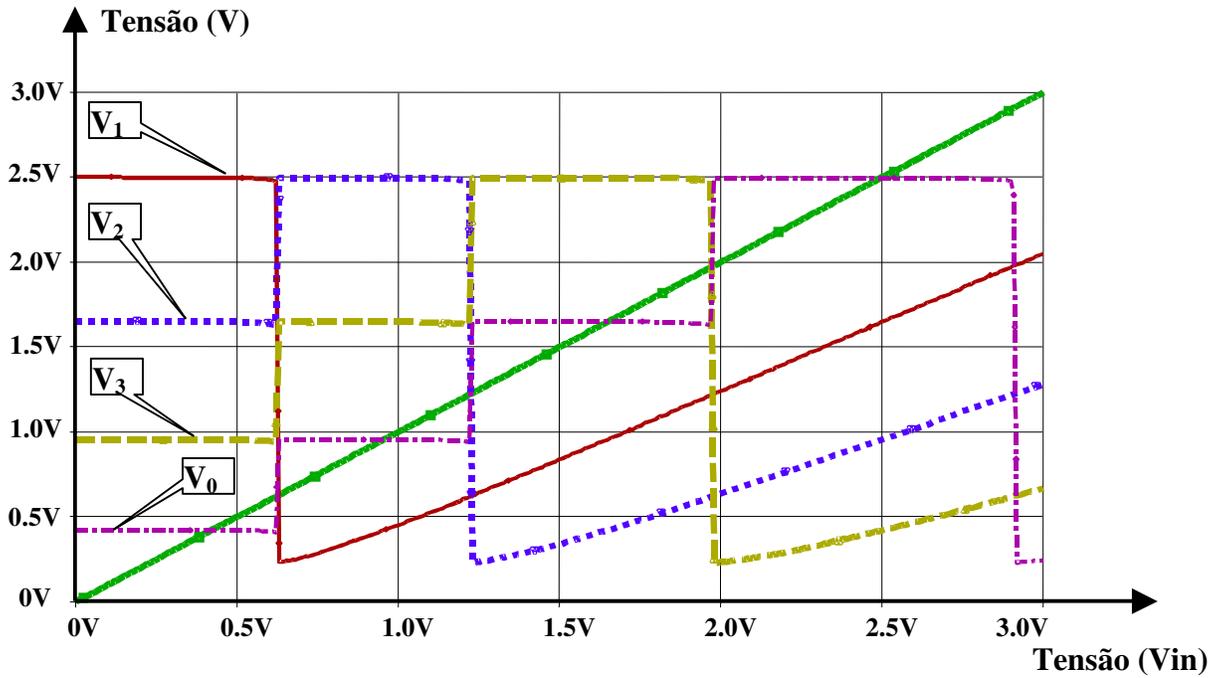


Figura 2.24 Curvas de simulação estática do Registrador Quaternário tipo N.

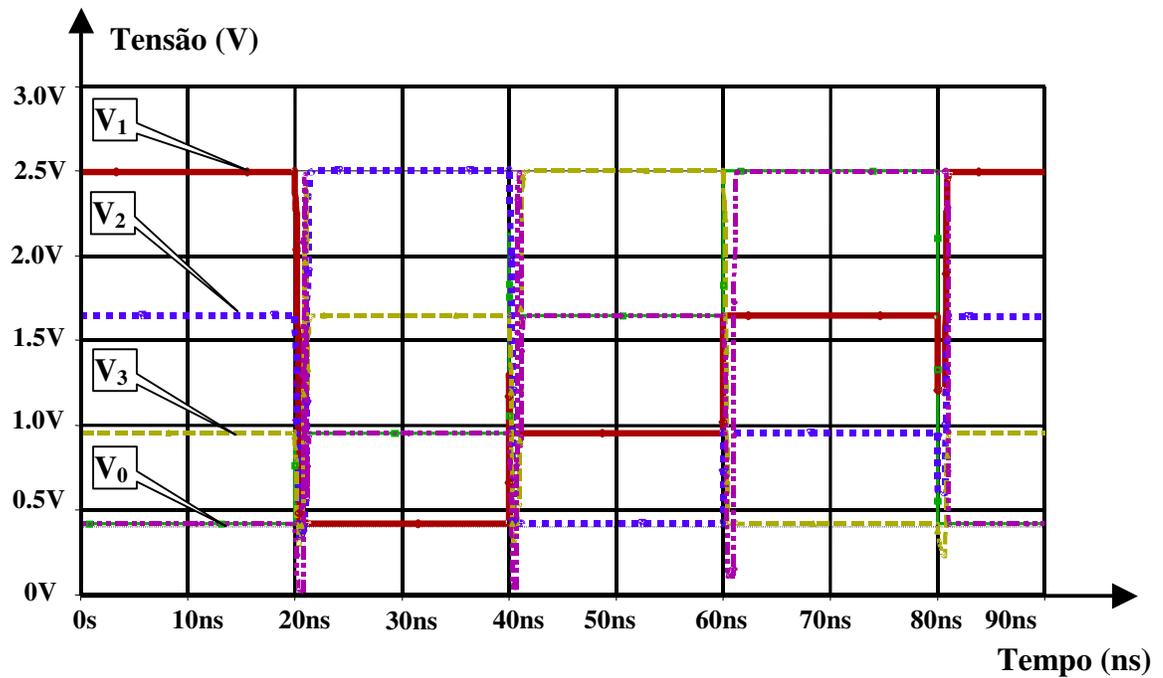


Figura 2.25 Curvas de simulação dinâmica do Registrador Quaternário tipo N.

Pode-se verificar pelo gráfico que os níveis de tensão para os valores lógicos obtidos dinamicamente são os mesmos que os determinados pelo gráfico da curva da Figura 2.24 obtido estaticamente pela varredura da tensão de entrada. Também pode-se ver neste gráfico, um tempo máximo de comutação “no pior caso” de menos de 1.5 ns, que é da mesma ordem que se teria em circuitos binários usando os mesmos transistores como pode ser-visto no Capítulo 6.

Com estes Registradores é possível montar um circuito multiplicador como uma das aplicações possíveis, que será tema do próximo capítulo.

Capítulo 3

Multiplicador Digital MVL

Neste capítulo serão introduzidos os conceitos básicos de sistemas de numeração e desenvolvido um processo lógico para realizar uma operação de multiplicação no sistema MVL Quaternário para um sistema numérico adotado.

No campo de aplicações aritméticas pode encontrar-se uma série de sistemas de numeração, que podem ser aplicados para codificação de sistema Multivalores, como por exemplo pode-se mencionar, sistemas ordinários ponderados (*ordinary weighted system*), sistemas simétricos ponderados (*symmetrical weighted system*), Dígitos com sinal (*Signed-digit*) e sistemas numéricos de resíduos simétricos (*Symmetrical residue number system*). Alguns trabalhos em lógica Multivalores utilizando estes sistemas podem ser encontrados nas referências [17], [18], [19], [20], [21], [22], [23] e [24].

No nosso trabalho adotaremos o sistema ordinário ponderado (*ordinary weighted system*) e um processo de multiplicação lógico que simplesmente é o utilizado na aritmética normal de números decimais que estamos acostumados desde nossa infância.

Para o sistema MVL quaternário o sistema ordinário ponderado será definido como mostra a equação 3.1 (geral) e 3.2 (quaternário):

$$A = a_{n-1} r^{n-1} + a_{n-2} r^{n-2} + \dots + a_1 r + a_0 \quad (3.1)$$

$$A = a_3 .4^3 + a_2 .4^2 + a_1 .4 + a_0 \quad (3.2)$$

sendo:

$$a_i \in \{0,1,2,3\}$$

$$r = 4$$

3.1 Processo de Multiplicação

Na continuação será descrito o processo de multiplicação de dois Dígitos vezes dois Dígitos com um resultado de quatro Dígitos, mostrada na Figura 3.1 com as seguintes definições:

Entradas A e B:

Dígito 0 do Numero A: A_0

Dígito 1 do Numero A: A_1

Dígito 0 do Numero B: B_0

Dígito 1 do Numero B: B_1

Resultado C:

Dígito 0 do Numero C: C_0

Dígito 1 do Numero C: C_1

Dígito 2 do Numero C: C_2

Dígito 3 do Numero C: C_3

B_1	B_0			
	x	A_1	A_0	
<hr/>				
		$A_0 \times B_1$	$A_0 \times B_0$	Primeira linha
	$A_1 \times B_1$	$A_1 \times B_0$		Segunda linha
<hr/>				
C_3	C_2	C_1	C_0	Resultado

Figura 3.1: Processo de Multiplicação

3.2 Multiplicador

O diagrama da Figura 3.2 mostra as interconexões em bloco dos circuitos: identificaremos com “SW” as chaves que interconectam os blocos, com “R” os registradores que armazenam os valores, com “X” os blocos que operam multiplicação e com “S” os que operam Soma.

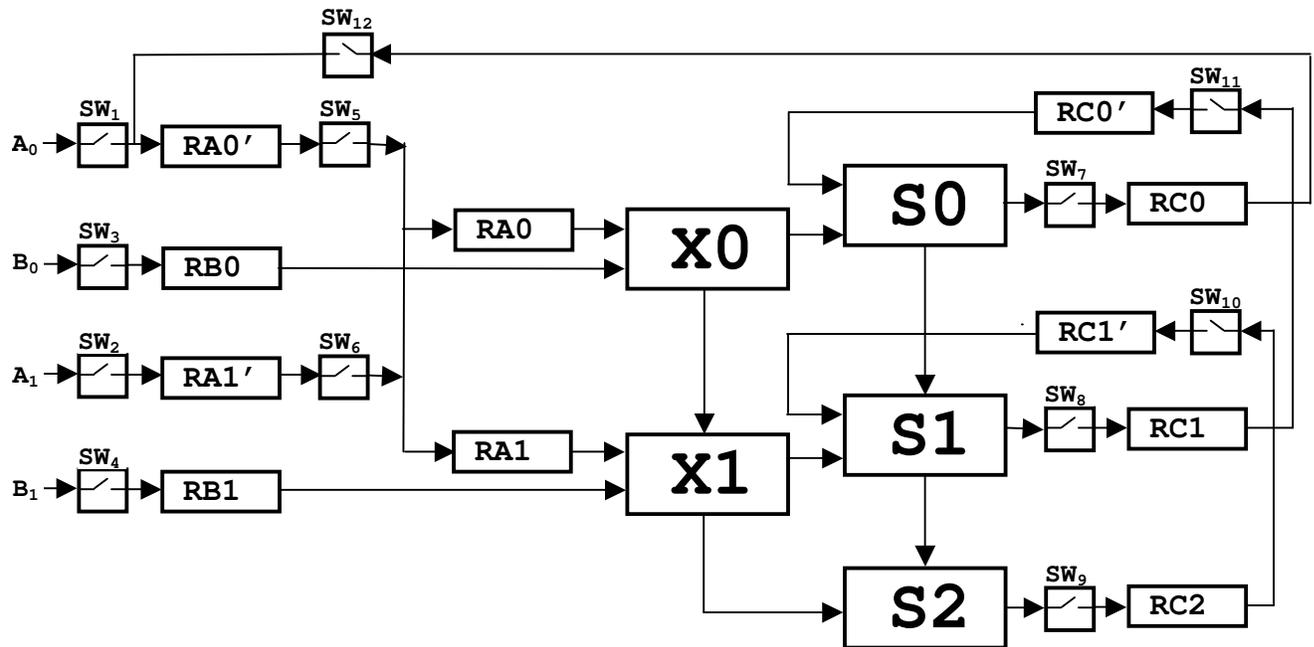


Figura 3.2 Diagrama em bloco do Circuito Multiplicador

Seguindo o diagrama em bloco da Figura 3.2 se obterá a seguinte seqüência (conforme a figura 3.1):

Primeira linha:

$$A_0 \times B_0 = C_0 \text{ gera Carry } C_{x_0}$$

$$A_0 \times B_1 + C_{x_0} = C_1' \text{ gera Carry } C_{x_1}$$

Onde:

C_{x_0} é o *carry* do produto $A_0 \times B_0$

C_{x_1} é o *carry* do produto $A_0 \times B_1$

C_1' é o transporte para segunda linha do Dígito 1

Segunda linha:

$$A_1 \times B_0 + C_1' = C_1 \text{ geram } C_{x_0}' \text{ e } C_{s_0}'$$

$$A_1 \times B_1 + C_{x_0}' + C_{s_0}' + C_{x_1} = C_2 \text{ geram } C_{x_1}' \text{ e } C_{s_1}'$$

$$C_{x_1}' + C_{s_1}' = C_3$$

Onde:

C_{x_0}' é o *carry* do produto $A_1 \times B_0$

C_{x_1}' é o *carry* do produto $A_1 \times B_1$

C_{s_0}' é o *carry* da soma com C_1'

C_{s_1}' é o *carry* da soma com C_{x_1}

3.3 Descrição de funcionamento

Foi considerado o tempo dividido em intervalos, que podem ser semiperíodos de *clock*, desta forma se teve a seguinte sequência de períodos.

1º Período:

O sinal do 1º Intervalo fecha as chaves SW_1 , SW_2 , SW_3 , e SW_4 sendo os dados armazenados nos registradores como segue:

$$RA_0' = A_0$$

$$RB_0 = B_0$$

$$RA_1' = A_1$$

$$RB_1 = B_1$$

2º Período:

O sinal do 2º Intervalo fecha a chave SW_5 e o dado passa do Registrador RA_0' para os Registradores RA_0 e RA_1 e o sinal de *clock* fecha as chaves SW_7 , SW_8 e SW_9 produzindo-se as seguintes operações:

$$A_0 \times B_0 = C_0 \text{ armazenado no registrador } RC_0$$

$$A_0 \times B_1 + Cx_0 = C_1' \text{ armazenado no registrador } RC_1$$

$$\text{Carry } Cx_1 \text{ armazenado no registrador } RC_2$$

3º Período:

O sinal do 3º Intervalo fecha as chaves SW_{10} , SW_{11} , e SW_{12} e os dados passam do Registrador RC_0 para o Registrador RA_0' , do Registrador RC_1 para o Registrador RC_0' e do Registrador RC_2 para o Registrador RC_1' .

4º Período:

O sinal do 4º Intervalo fecha a chave SW_6 e o dado A_1 do Registrador RA_1 passa para os Registradores RA_0 e RA_1 e o sinal de *clock* fecha as chaves SW_7 , SW_8 e SW_9 produzindo-se as seguintes operações:

$$A_1 \times B_0 + C_1' = C_1 \text{ armazenado no Registrador } RC_0$$

Gerando C_{x_0}' e C_{s_0}'

$$A_1 \times B_1 + C_{x_0}' + C_{s_0}' + C_{x_1} = C_2 \text{ armazenado no registrador } RC_1$$

Gerando C_{x_1}' e C_{s_1}'

$$C_{x_1}' + C_{s_1}' = C_3 \text{ armazenado no registrador } RC_2$$

O Resultado da operação fica armazenado da seguinte forma:

$$C_0 = RA_0'$$

$$C_1 = RC_0$$

$$C_2 = RC_1$$

$$C_3 = RC_2$$

3.4 Seqüência de pulsos de *clock*

Para que o processo de multiplicação se realize, serão gerados os seguintes sinais para fechamento das chaves, que farão que os dados passem dos registradores para os circuitos e dos circuitos para os registradores.

O primeiro sinal da Figura 3.3 é o sinal de *clock* que sincroniza toda a operação. O sinal seguinte chamado de 1º Período fechará as chaves que permitirão a primeira operação, e assim sucessivamente os demais sinais seguindo a explicação dada no item 3.3.

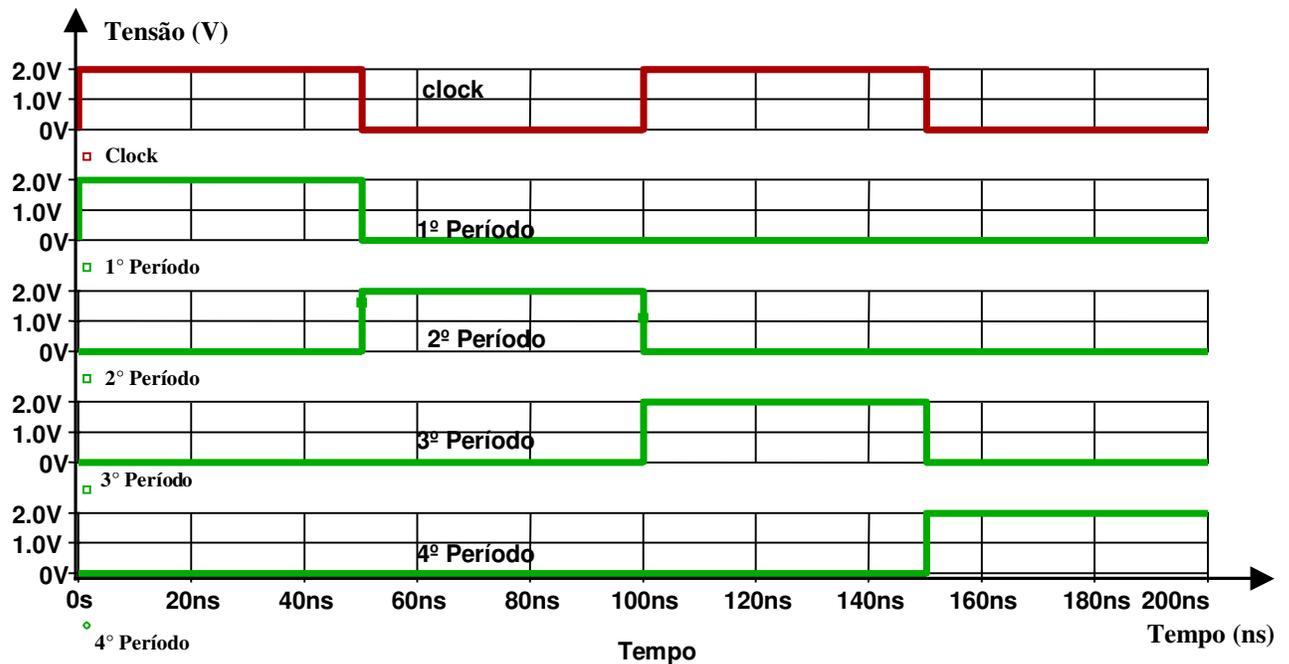


Figura 3.2 Sinais de *clock*, Primeiro é o sinal de *clock*, o segundo sinal fecha as chaves do primeiro período, o terceiro sinal fecha as chaves do segundo período, o quarto sinal fecha as chaves do terceiro período e o quinto sinal fecha as chaves do quarto período.

Como pode observar-se o processo de multiplicação descrito neste capítulo levará dois ciclos de *clock*, isto mostra que o tempo total de operação dependerá da frequência de *clock* que será função da resposta em frequência dos circuitos utilizados. Por esta razão nos circuitos que serão desenvolvidos nos próximos capítulos, será considerado o tempo de estabilização dos sinais, sendo este tempo um dos fatores importantes a ser estudado.

Com estas considerações em mente, nos próximos capítulos serão desenvolvidos os circuitos necessários para a implementação de todo o processo de multiplicação digital MVL quaternária.

Capítulo 4

Multiplicador e Somador

Neste capítulo apresentamos a implementação de um bloco Multiplicador e outro Somador utilizando como base as Tabelas Verdade [18] e registradores quaternários do Capítulo 2 .

4.1 Tabela Verdade

Para a implementação de um Multiplicador Digital Quaternário seguindo as seqüências de multiplicação do Capítulo 3 deveremos elaborar as Tabelas Verdade da multiplicação [18] com a soma do *Carry* do produto anterior, como pode ver-se no Capítulo 3 deste texto. Como no caso de Lógica Binária, as funções podem ser representadas por mapas de Karnaugh e utilizar técnicas de simplificação [04-05].

Os mapas são os seguintes:

X0		B			
Cy=0		0	1	2	3
A	0	0	0	0	0
	1	0	1	2	3
	2	0	2	0	2
	3	0	3	2	1

Figura 4.1 Tabela Verdade do Valor do Multiplicador com *Carry* na entrada = 0

CX0		B			
Cy=0		0	1	2	3
A	0	0	0	0	0
	1	0	0	0	0
	2	0	0	1	1
	3	0	0	1	2

Figura 4.2 Tabela Verdade do Carry do Multiplicador com *Carry* na entrada = 0

X1		B			
Cy=1		0	1	2	3
A	0	1	1	1	1
	1	1	2	3	0
	2	1	3	1	3
	3	1	0	3	2

Figura 4.3 Tabela Verdade do Valor do Multiplicador com *Carry* na entrada = 1

CX1		B			
Cy=1		0	1	2	3
A	0	0	0	0	0
	1	0	0	0	1
	2	0	0	1	1
	3	0	1	1	2

Figura 4.4 Tabela Verdade do *Carry* do Multiplicador com *Carry* na entrada = 1

X2		B			
Cy=2		0	1	2	3
A	0	2	2	2	2
	1	2	3	0	1
	2	2	0	2	0
	3	2	1	0	3

Figura 4.5 Tabela Verdade do Valor do Multiplicador com *Carry* na entrada = 2

CX2		B			
Cy=2		0	1	2	3
A	0	0	0	0	0
	1	0	0	1	1
	2	0	1	1	2
	3	0	1	2	2

Figura 4.6 Tabela Verdade do *Carry* do Multiplicador com *Carry* na entrada = 2

Também foram necessários os mapas da soma com *Carry* de “0” e “1”.

S0		B			
Cy=0		0	1	2	3
A	0	0	1	2	3
	1	1	2	3	0
	2	2	3	0	1
	3	3	0	1	2

Figura 4.7 Tabela Verdade do Valor do Somador com *Carry* na entrada = 0

CS0		B			
Cy=0		0	1	2	3
A	0	0	0	0	0
	1	0	0	0	1
	2	0	0	1	1
	3	0	1	1	1

Figura 4.8 Tabela Verdade do *Carry* do Somador com *Carry* na entrada = 0

S1		B			
Cy=1		0	1	2	3
A	0	1	2	3	0
	1	2	3	0	1
	2	3	0	1	2
	3	0	1	2	3

Figura 4.9 Tabela Verdade do Valor do Somador com *Carry* na entrada = 1

CS1		B			
Cy=1		0	1	2	3
A	0	0	0	0	1
	1	0	0	1	1
	2	0	1	1	1
	3	1	1	1	1

Figura 4.10 Tabela Verdade do *Carry* do Somador com *Carry* na entrada = 1

4.2 Módulo para o Registrador com saída DECODIFICADA

Agora, pode ser feita uma pequena modificação nas saídas do MÓDULO, e no lugar de sair com o valor quaternário, se utiliza a saída da chave da forma mostrada no circuito da Figura 4.11.

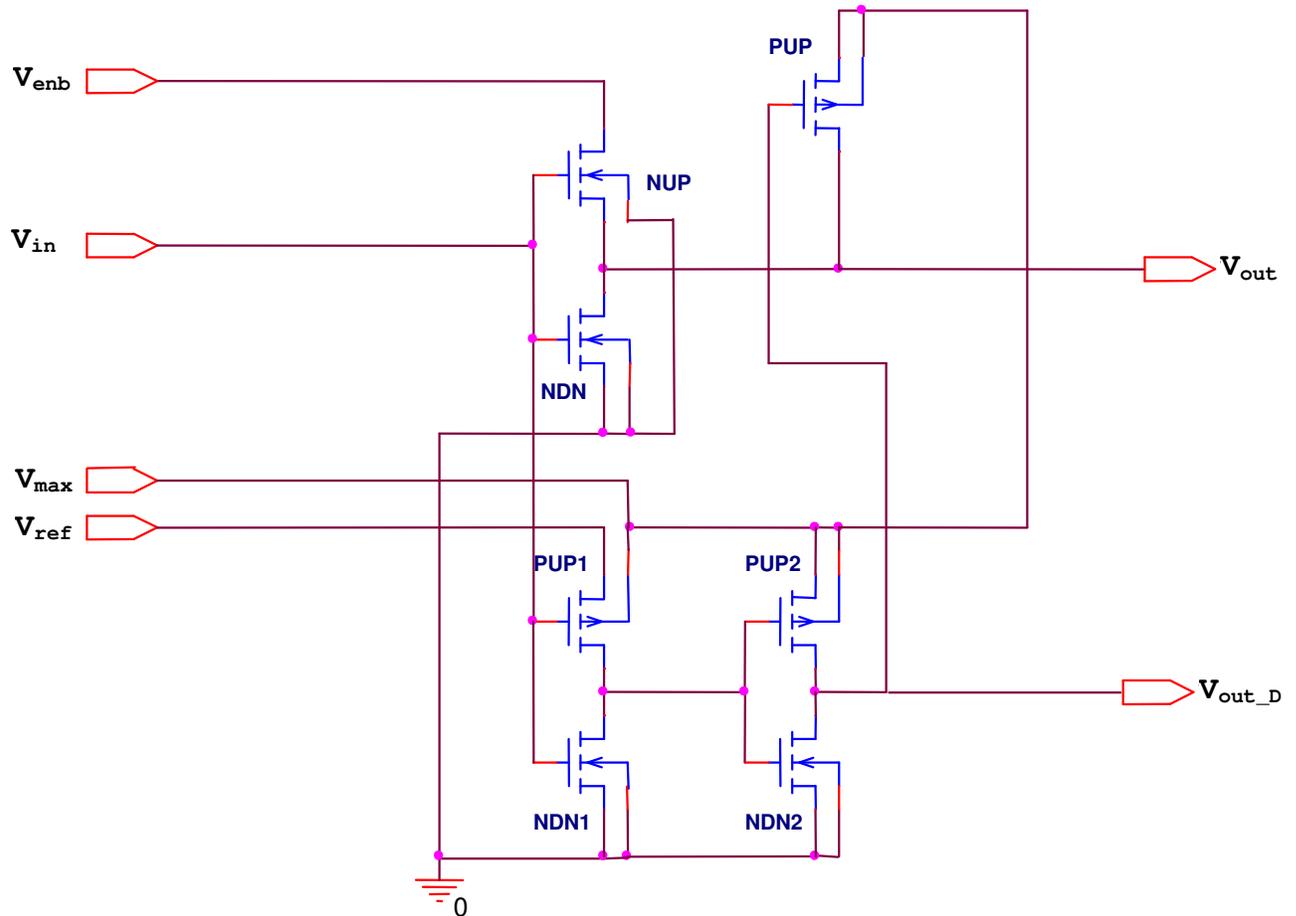


Figura 4.11 Circuito do Módulo para o Registrador com saída BINÁRIA.

Foi denominado como: MÓDULO_B e o diagrama em bloco esta mostrado na Figura 4.12.

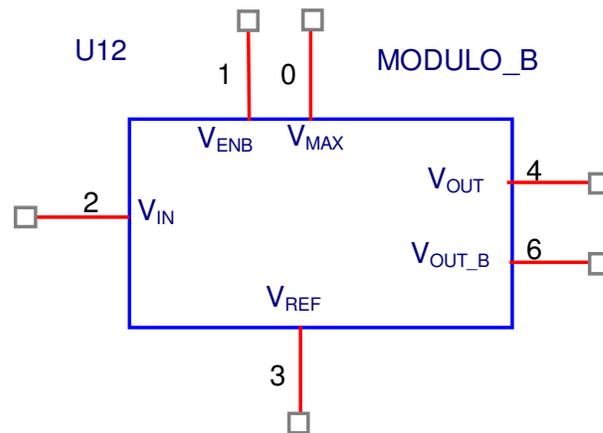


Figura 4.12 Bloco do Modulo para o Registrador com saída BINÁRIA.

A letra “B” significa “BINÁRIO”. A saída será “0” ou “1”, tendo uma interpretação um pouco diferente ao que estamos acostumados com sistemas Binários, pois a saída será “0” somente quando a entrada seja o nível “0” quaternário e será considerado como “Valor Verdadeiro”, e a saída será “1” para qualquer outro valor quaternário que não corresponda ao “0” quaternário e será considerado como “Valor Falso”, como mostra o mapa da Figura 4.13.

V _{IN}	V _{OUT}	V _{OUT_B}
0	3	0 (Verdadeiro)
1	0	1 (Falso)
2	1	1 (Falso)
3	2	1 (Falso)

Figura 4.13 Mapa do Bloco do Modulo para o Registrador com saída BINÁRIA.

4.3 Registrador com saída DECODIFICADA

Aplicando este novo bloco ao registrador quaternário, agora com saídas BINÁRIAS, denominado o registrador como “REGISTRADOR_QUAT_N_D onde a letra “D” significa saída com valores DECODIFICADO (ver significado no GLOSSÁRIO deste texto).

O bloco fica da forma mostrada na Figura 4.14:

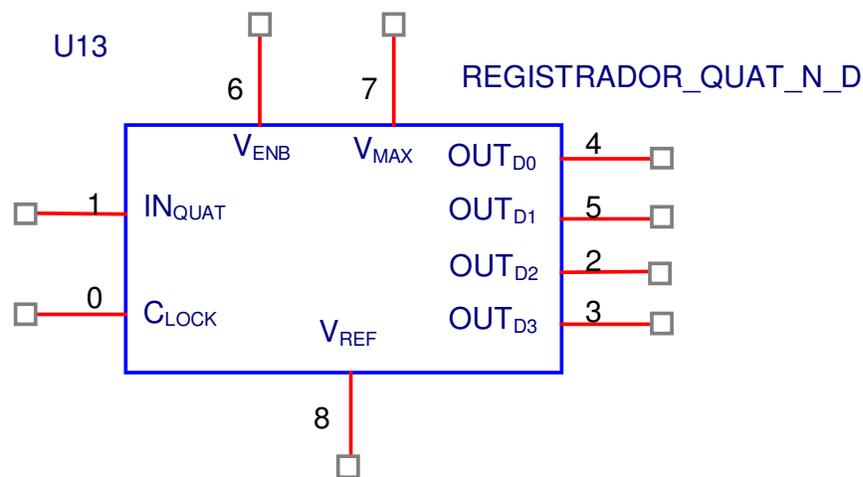


Figura 4.14 Bloco do Registrador com saída DECODIFICADA

Segue o mapa deste bloco na Figura 4.15:

QUATERNARIO	DECODIFICADO			
INQUAT	OUT_D_0	OUT_D_1	OUT_D_2	OUT_D_3
0	0	1	1	1
1	1	0	1	1
2	1	1	0	1
3	1	1	1	0

Figura 4.15 Mapa do Bloco do Registrador Quaternário com Saída DECODIFICADA.

Nas figuras 4.16 e 4.17 se mostra as simulações estática e dinâmica do registrador com saída Decodificada.

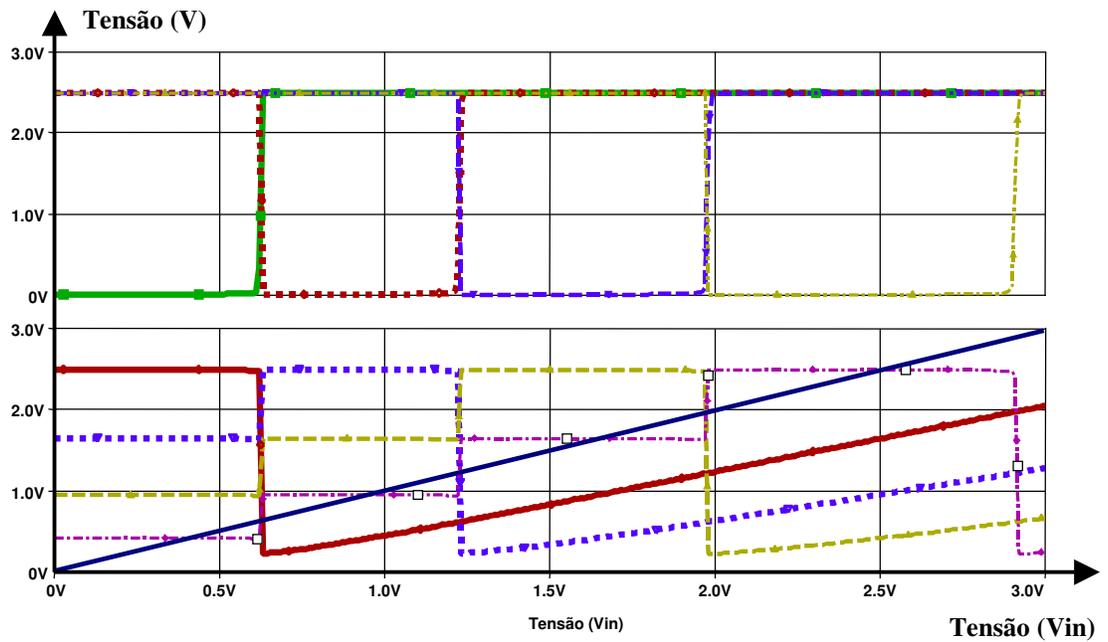


Figura 4.16 Simulação estática do Modulo para o Registrador com saída BINÁRIA

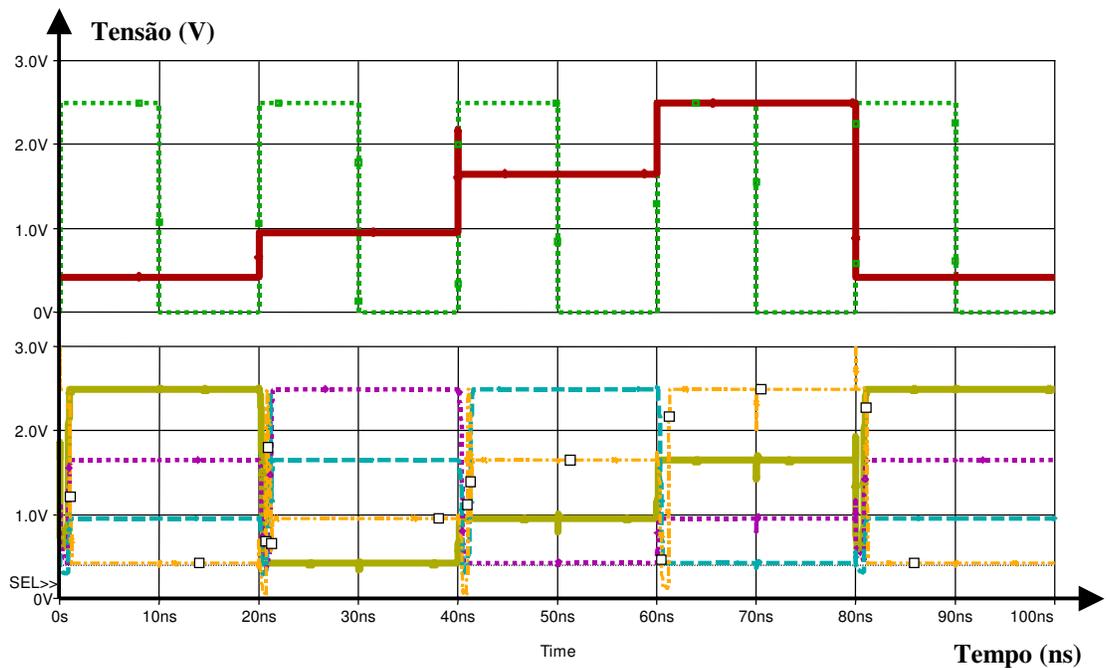


Figura 4.17 Simulação dinâmica do Modulo para o Registrador com saída BINÁRIA

4.4 Tabela Verdade do Multiplicador e do Somador

Utilizando agora as tabelas verdade de multiplicação, soma e a deste bloco, foi possível elaborar uma tabela da seguinte forma:

QUATERNARIO		DECODIFICADO				DECODIFICADO				VALOR QUATERNARIO									
A	B	A0	A1	A2	A3	B0	B1	B2	B3	X0	CX0	X1	CX1	X2	CX2	S0	CS0	S1	CS1
0	0	0	X	X	X	0	X	X	X	0	0	1	0	2	0	0	0	1	0
0	1	0	X	X	X	X	0	X	X	0	0	1	0	2	0	1	0	2	0
0	2	0	X	X	X	X	X	0	X	0	0	1	0	2	0	2	0	3	0
0	3	0	X	X	X	X	X	X	0	0	0	1	0	2	0	3	0	0	1
1	0	X	0	X	X	0	X	X	X	0	0	1	0	2	0	1	0	2	0
1	1	X	0	X	X	X	0	X	X	1	0	2	0	3	0	2	0	3	0
1	2	X	0	X	X	X	X	0	X	2	0	3	0	0	1	3	0	0	1
1	3	X	0	X	X	X	X	X	0	3	0	0	1	1	1	0	1	1	1
2	0	X	X	0	X	0	X	X	X	0	0	1	0	2	0	2	0	3	0
2	1	X	X	0	X	X	0	X	X	2	0	3	0	0	1	3	0	0	1
2	2	X	X	0	X	X	X	0	X	0	1	1	1	2	1	0	1	1	1
2	3	X	X	0	X	X	X	0	0	2	1	3	1	0	2	1	1	2	1
3	0	X	X	X	0	0	X	X	X	0	0	1	0	2	0	3	0	0	1
3	1	X	X	X	0	X	0	X	X	3	0	0	1	1	1	0	1	1	1
3	2	X	X	X	0	X	X	0	X	2	1	3	1	0	2	1	1	2	1
3	3	X	X	X	0	X	X	X	0	1	2	2	2	3	2	2	1	3	1

Figura 4.18 Tabela Verdade do Multiplicador Quaternário

4.5 Considerações

Para desenvolver o circuito multiplicador, se fizeram as seguintes considerações que levaram a uma simplificação:

- a) Utilizar a entrada e a saída DECODIFICADA.
- b) Observando as colunas X0, X1 e X2, pode-se ver que existe um deslocamento de uma unidade de uma coluna para a outra. Isto permite simplificar nosso circuito, utilizando um deslocador na saída DECODIFICADA para somar o *Carry*.

4.6 Circuito Multiplicador

Utilizando técnicas de simplificação e as considerações acima expostas se chegou ao seguinte circuito:

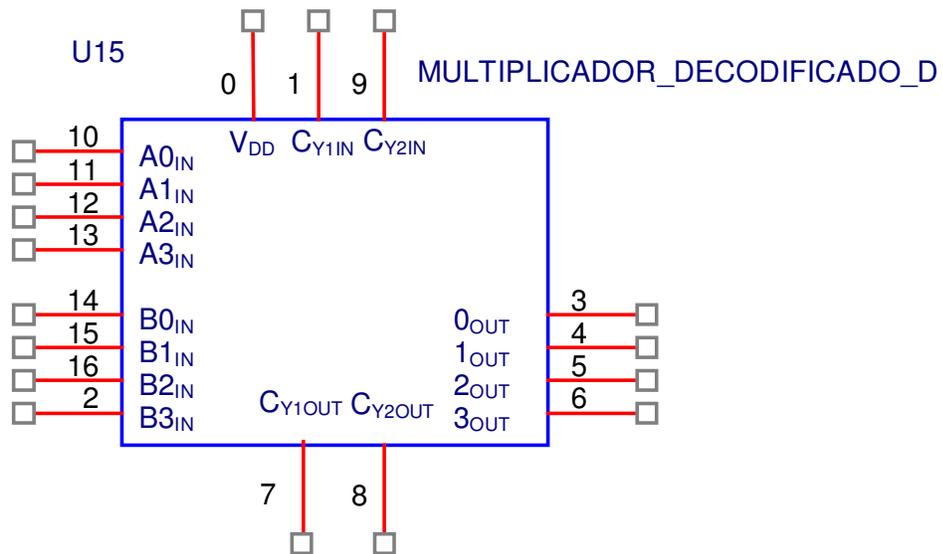


Figura 4.19 Bloco do Multiplicador DECODIFICADO _D

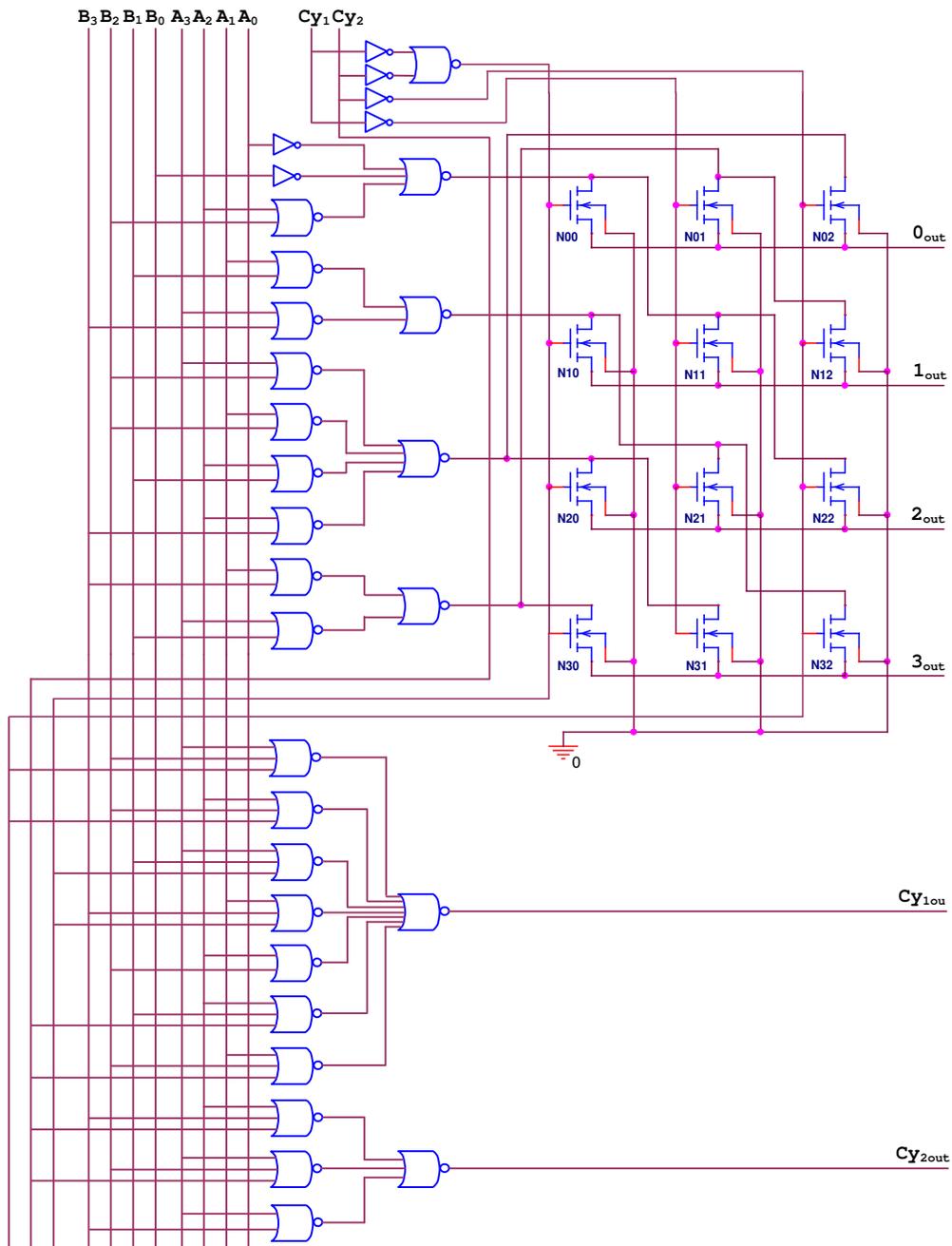


Figura 4.20 Circuito Multiplicador DECODIFICADO

Pode-se notar neste circuito que a primeira parte (lado esquerdo) esta composta por portas lógicas binárias INVERSORES e NAND`s, a parte superior sintetiza o valor do produto e a parte embaixo sintetiza o *Carry*. A segunda parte do circuito (lado direito) esta composta por chaves formando uma matriz que faz o deslocamento do *Carry* de entrada.

Desta forma se tem uma saída DECODIFICADA, que será a entrada do Somador.

4.7 Circuito Somador

O Somador também tem entrada e saída DECODIFICADA.

Este circuito é muito simples, já que utilizou um deslocador feito com transistores tipo N, da mesma forma que foi utilizado no multiplicador, tanto para o valor como para o *Carry* de entrada. Para o *Carry* de saída foram utilizadas portas lógicas.

O circuito fica da seguinte forma:

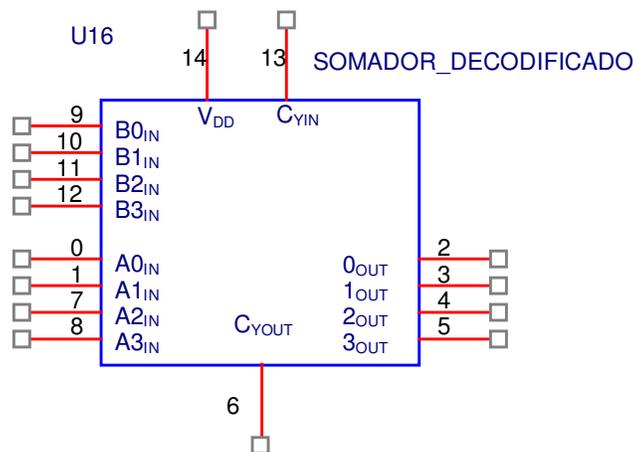


Figura 4.21 Bloco do Somador DECODIFICADO

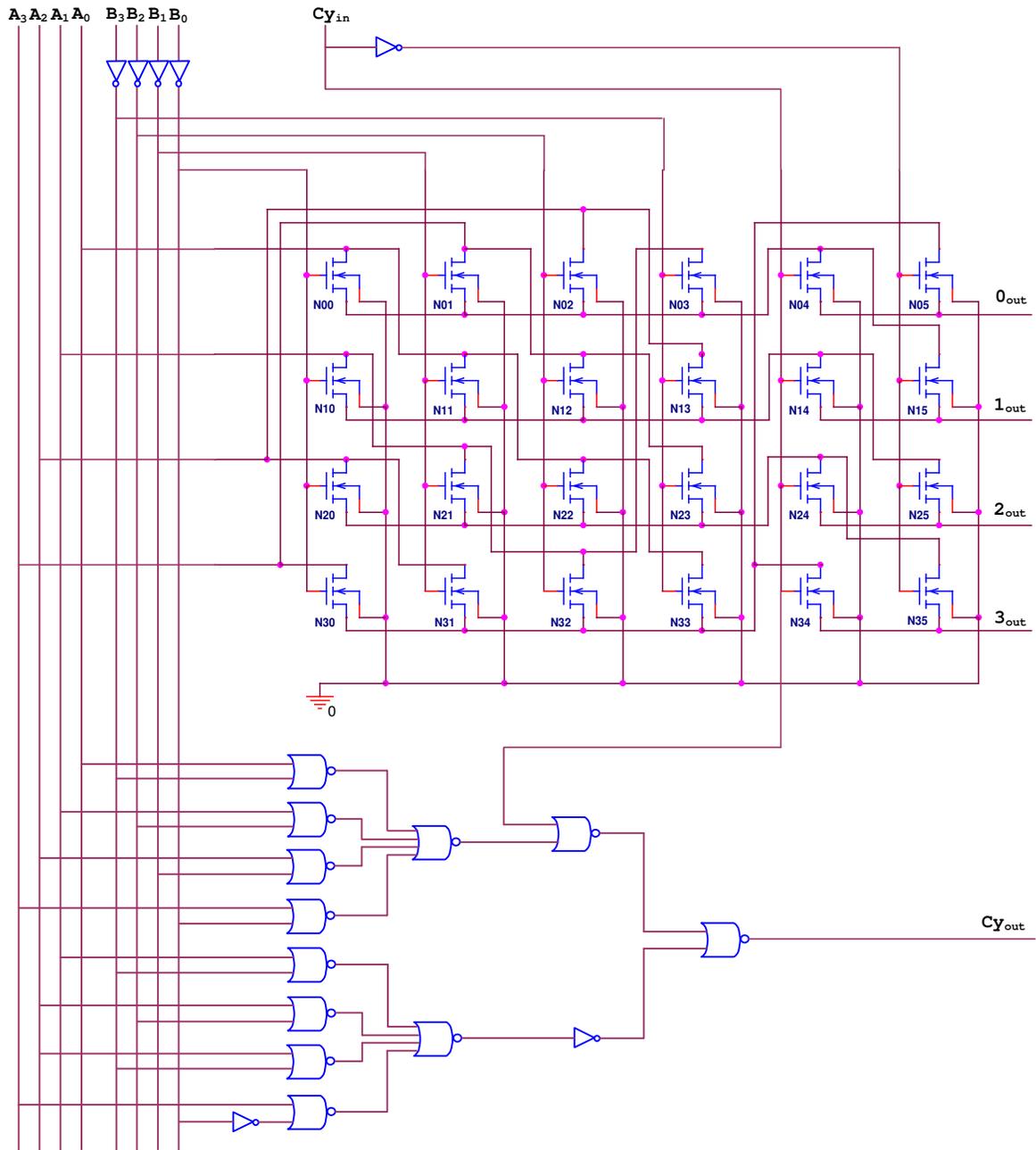


Figura 4.22 Circuito do Somador DECODIFICADO

Estes dois blocos, o Multiplicador e o Somador são circuitos muito rápidos compostos de portas lógicas e chaves convencionais; não foi feita a simulação no OrCad (Pspice) [01] porque será feita no circuito completo no próximo capítulo.

No próximo Capítulo serão desenvolvidos alguns circuitos adicionais necessários e o circuito Multiplicador completo com sua simulação no OrCad (Pspice) [01].

Capítulo 5

Multiplicador Quaternário Completo

Neste capítulo apresentamos a implementação de um Multiplicador Quaternário completo utilizando a lógica e método de multiplicação descritos no Capítulo 3 e o Registrador, Multiplicador e Somador desenvolvidos nos Capítulos 2 e 4.

5.1 Módulo *Driver*

Para completar o circuito, foi necessário fazer um registrador que tenha uma entrada DECODIFICADA e uma saída QUATERNÁRIA. Modificando o MODULO, foi acrescentado um transistor *Driver* como indica no circuito da Figura 5.2

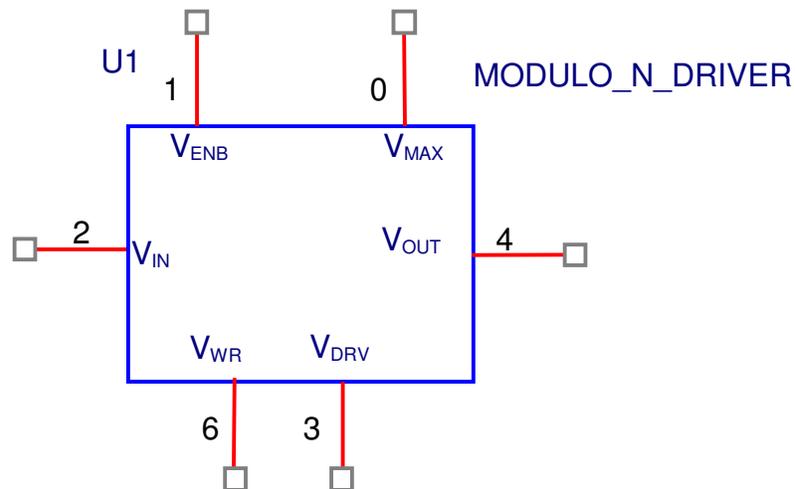


Figura 5.1 Bloco do Módulo_N Driver

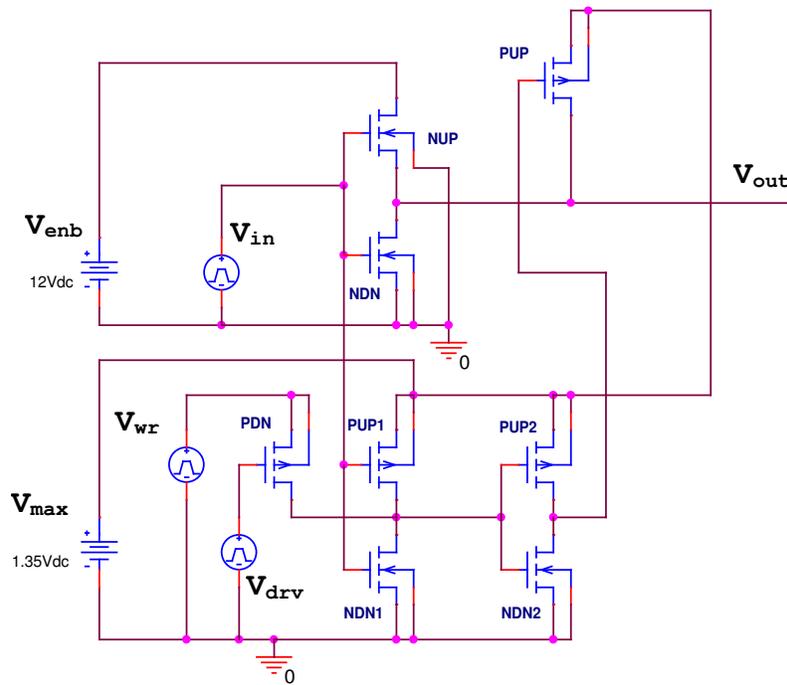


Figura 5.2 Circuito de teste do Módulo_N Driver

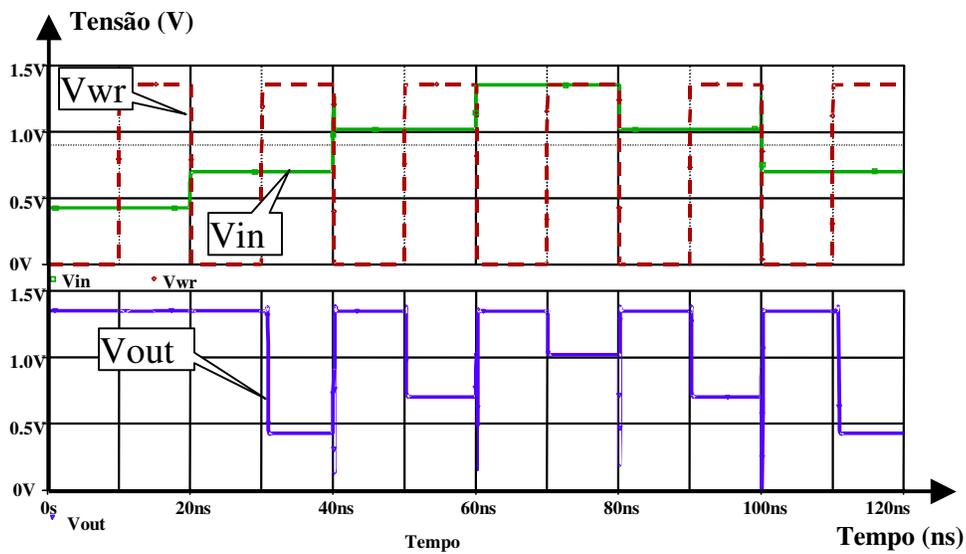


Figura 5.3 Curvas de Entrada e Saída do Módulo_N Driver

A colocação de um transistor *Driver* na chave inversora, pode requerer o ajuste do comprimento do canal do transistor N_{DN1} para que se tenha uma comutação segura com os níveis de entrada DECODIFICADAS, que no caso em estudo é de 0V e 1,5V. Quando a tensão de entrada V_{INDRV} no transistor P_{DN} é alta, o transistor estará cortado e o Módulo funciona como nos caso anterior dando o nível de saída de acordo com o nível de entrada, e quando a entrada V_{INDRV} seja 0V o transistor P_{DN} estará conduzindo e forçará a tensão de saída para o nível máximo.

5.2 Registrador *Driver*

Com este novo módulo se monta o REGISTRADOR com entrada DECODIFICADA e saída QUATERNÁRIA, que fica da seguinte forma:

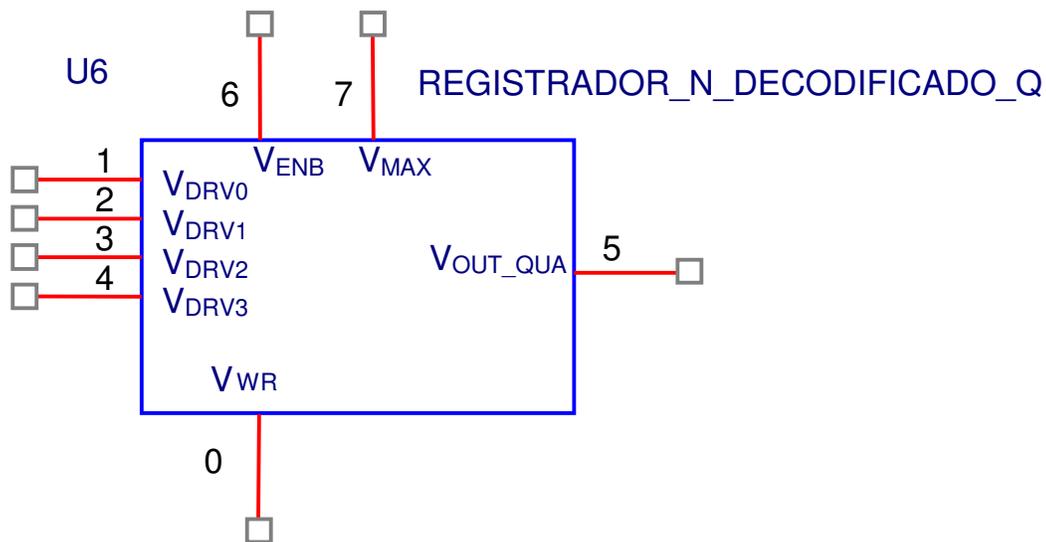


Figura 5.4 Bloco do Registrador com entrada DECODIFICADA e saída QUATERNÁRIA

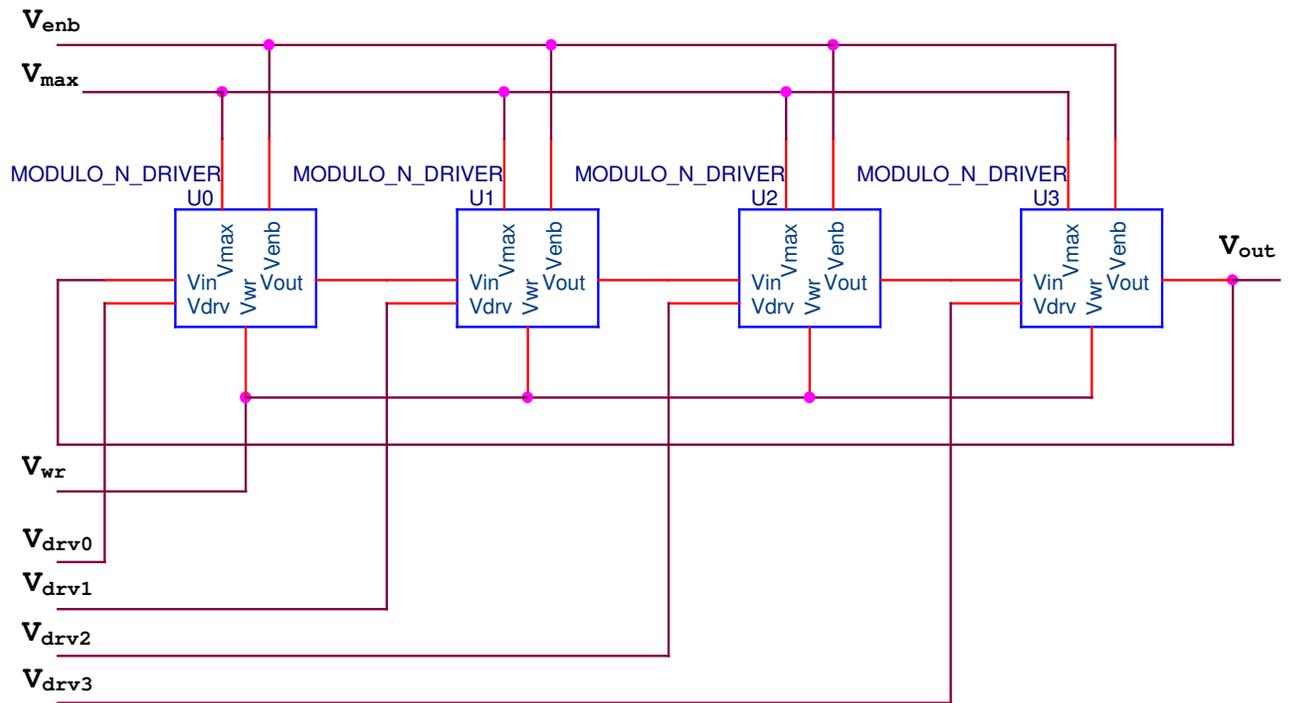


Figura 5.5 Circuito do Registrador com entrada DECODIFICADA e saída QUATERNÁRIA

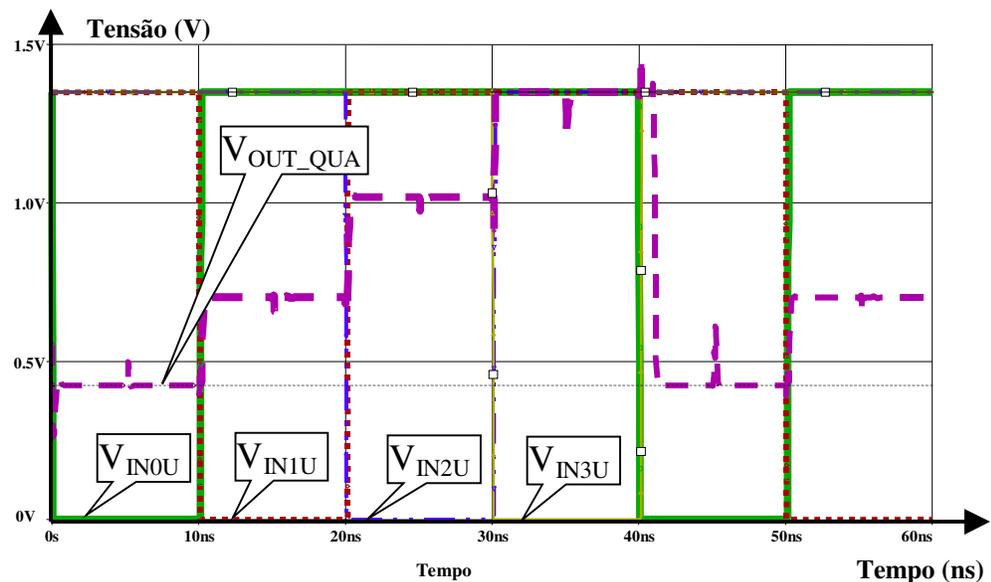


Figura 5.6 Curvas de Entrada e Saída do Registrador com entrada DECODIFICADA e saída QUATERNÁRIA

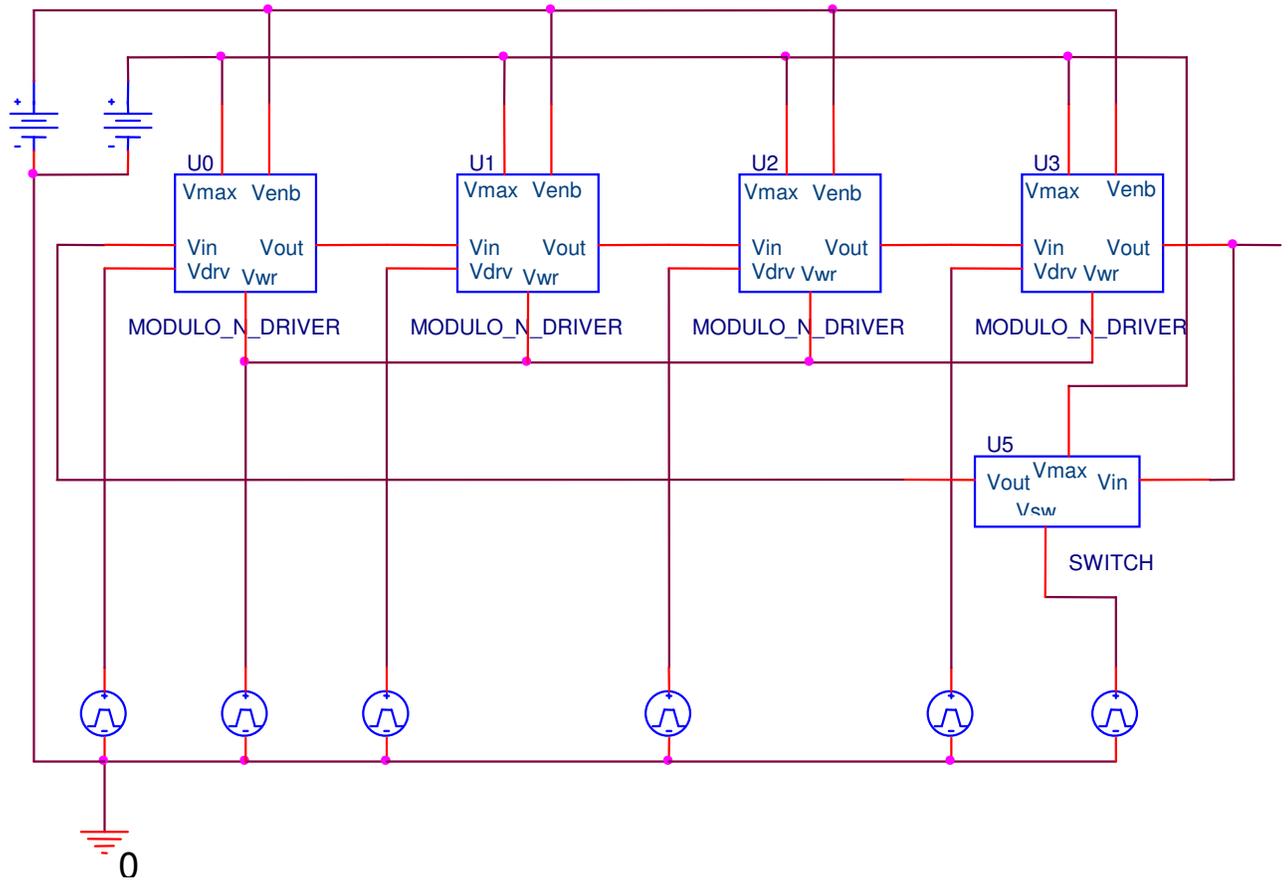


Figura 5.7: Circuito de teste do Registrador com entrada DECODIFICADA e saída QUATERNÁRIA

As entradas V_{IN0D} , V_{IN1D} , V_{IN2D} e V_{IN3D} terão sempre um único valor zero e as demais terão o valor máximo, desta forma a entrada zero forçará a saída do seu módulo para o valor máximo definindo os demais níveis e estabilizando as tensões QUATERNÁRIAS quando a chave feche o *loop*, desta forma se terá a saída quaternária correspondente à codificação da entrada.

5.3 Circuito de finalização do Multiplicador

Foi necessário fazer mais um bloco para a soma dos *carry*'s do último Dígito de cada linha com saída DECODIFICADA, o diagrama em bloco e o circuito são mostrados nas Figura 5.8 e 5.9:

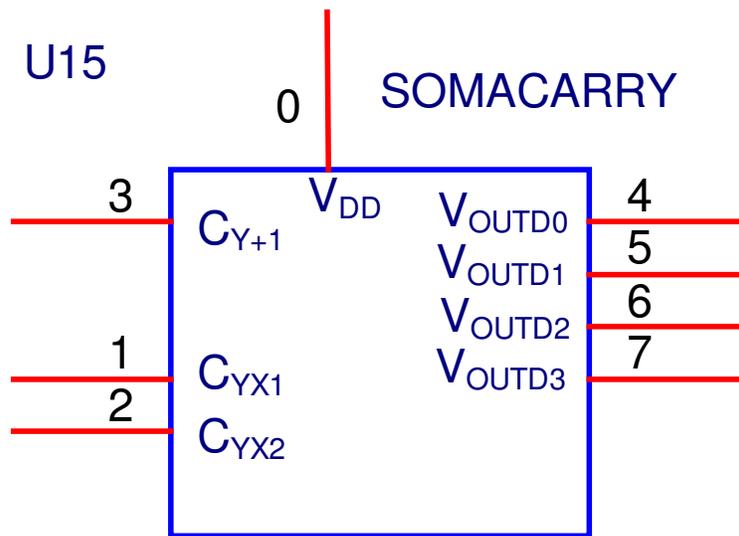


Figura 5.8 Bloco do circuito de finalização do Multiplicador

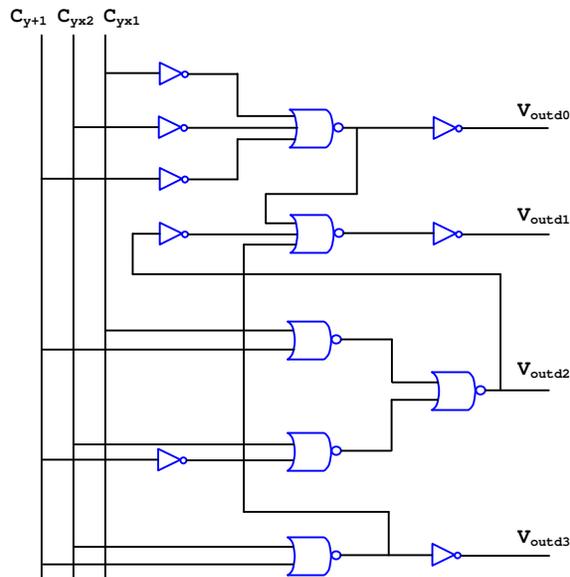


Figura 5.9 Circuito de Finalização do Multiplicador

Este circuito tem entrada do *Carry* da Multiplicação e do *Carry* da Soma definidos por duas linhas para a multiplicação e uma linha para a soma com lógicas dadas pela tabelas da Figura 5.16. Sendo uma lógica binária, foi implementado com portas lógicas convencionais. Na saída se tem quatro linhas DECODIFICADAS.

5.4 Circuito do Multiplicador

Com todos estes blocos é possível montar o circuito multiplicador, que tem duas entradas quaternárias de um Dígito quaternário em cada entrada e com saída de dois Dígitos quaternários. Para completar o módulo foi aplicada às entradas do *carry* do estagio anterior tanto da multiplicação como da soma e uma entrada quaternária do transporte da multiplicação.

O circuito e a simulação de um estagio de Multiplicação de 1 x 1 Dígito Quaternário é mostrado na continuação.

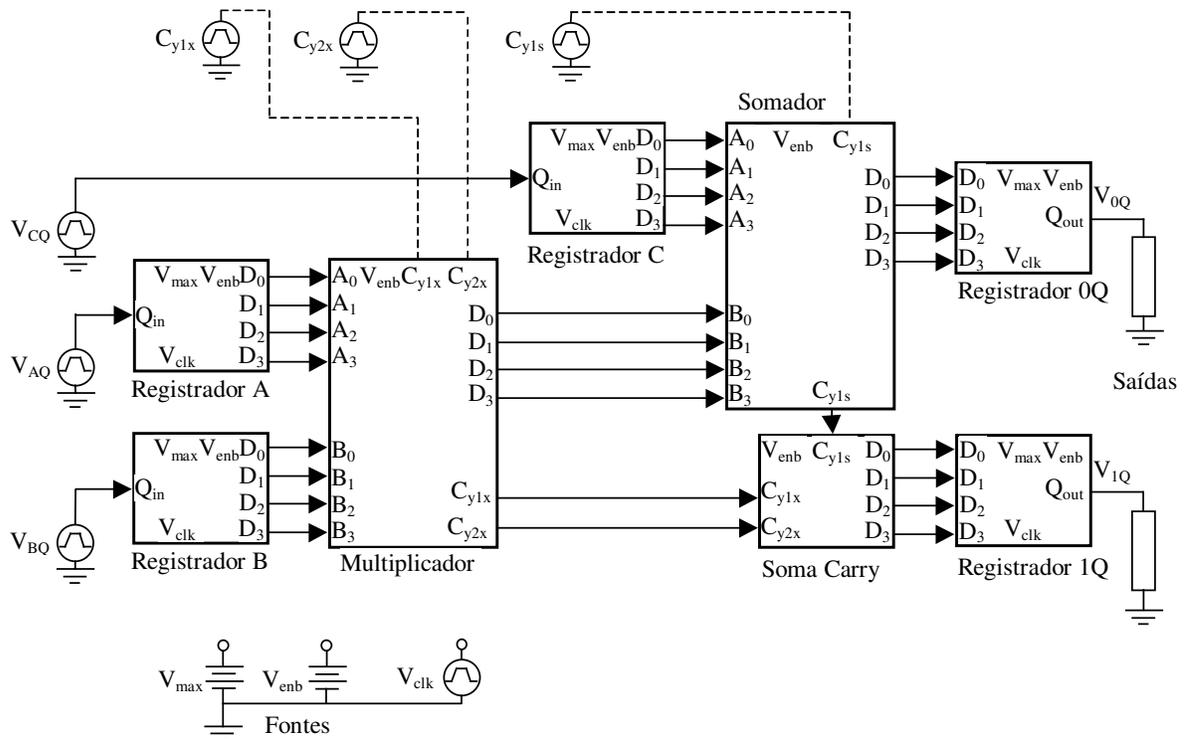


Figura 5.10 Circuito de Simulação de um estagio de Multiplicação

Na Figura 5.11 as primeiras três curvas mostram as entradas A e B, e a entrada do transporte da linha anterior C, e as outras duas curvas mostram o resultado da multiplicação e o *Carry* para o próximo Dígito.

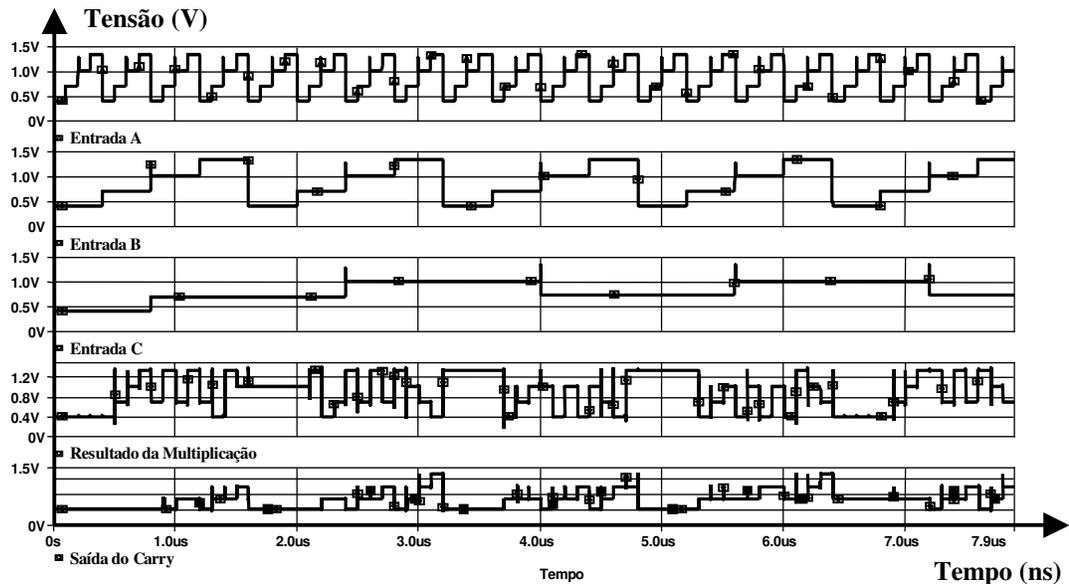


Figura 5.11 Curvas de Entrada e Saída do Multiplicador Completo

No gráfico da Figura 5.12 pode-se notar na primeira curva o consumo de potência total, e na segunda o consumo no caso do V_{ENB} desligado.

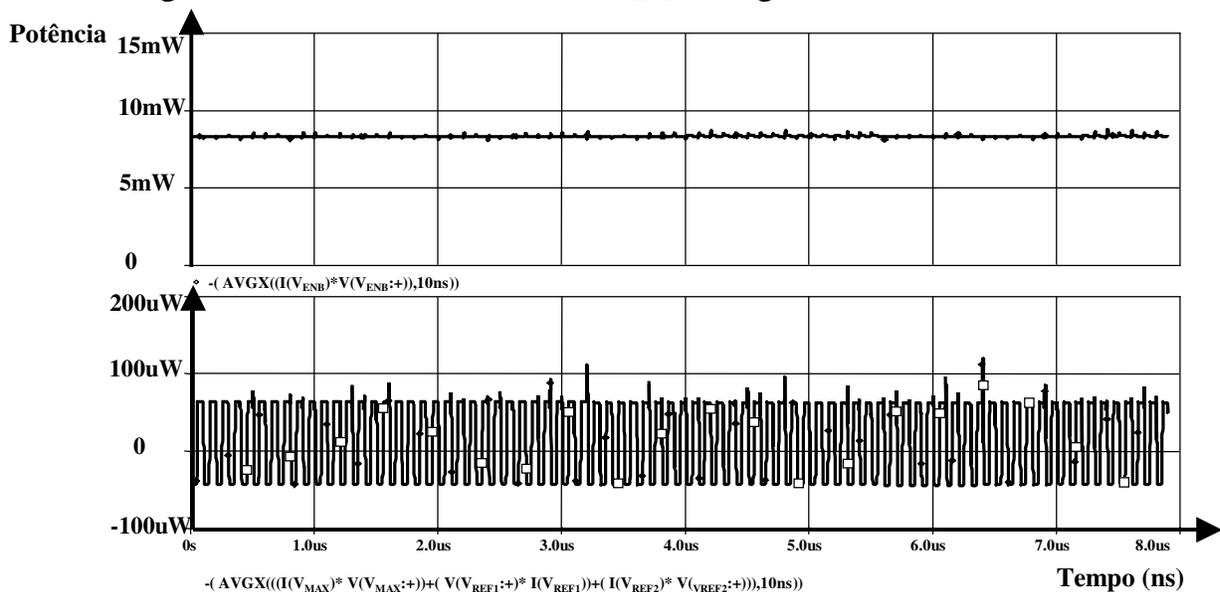


Figura 5.12 Curvas de consumo de Potência do Multiplicador

No gráfico da Figura 5.13 pode ver-se o consumo de potência total nos primeiros 120ns e nos seguintes 120ns (de 120ns ate 240ns) o consumo de potência em *stand-by* com o V_{ENB} desligado. Na curva inferior se mostra a MEDIA de potência com uma janela de 20ns.

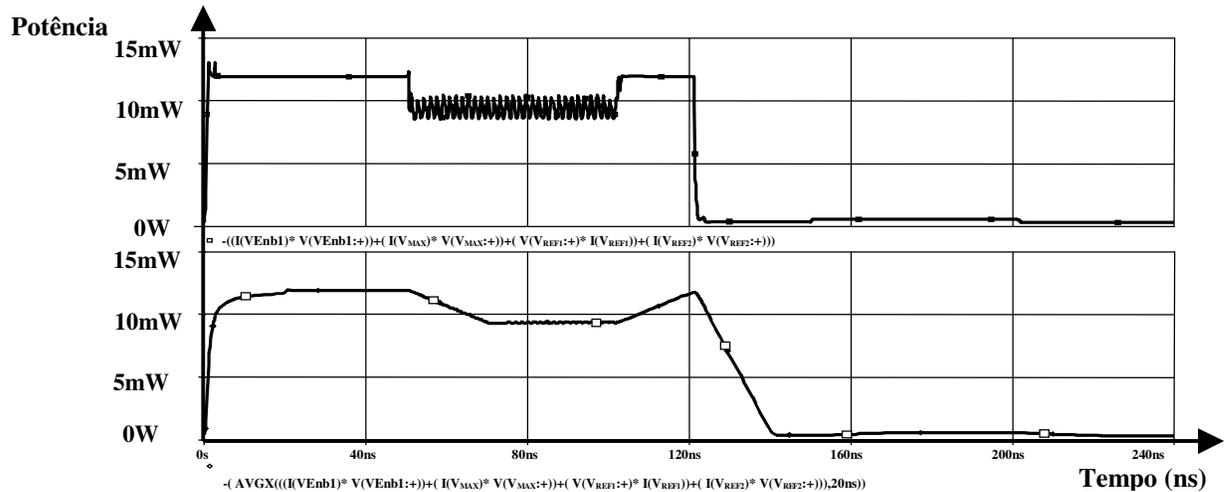


Figura 5.13 Curvas de consumo de Potência instantânea e média com V_{ENB} ligado e desligado

5.4.1 Descrição de funcionamento

Com este módulo é possível montar um multiplicador de ‘n Dígitos’ quaternários; para exemplificar esta implementação mostraremos a seguir um circuito de duas entradas com dois Dígitos quaternários cada uma e com saída de quatro Dígitos quaternários, e com a sequência de *clock's* para realizar a multiplicação.

O circuito é da forma mostrada na Figura 5.14:

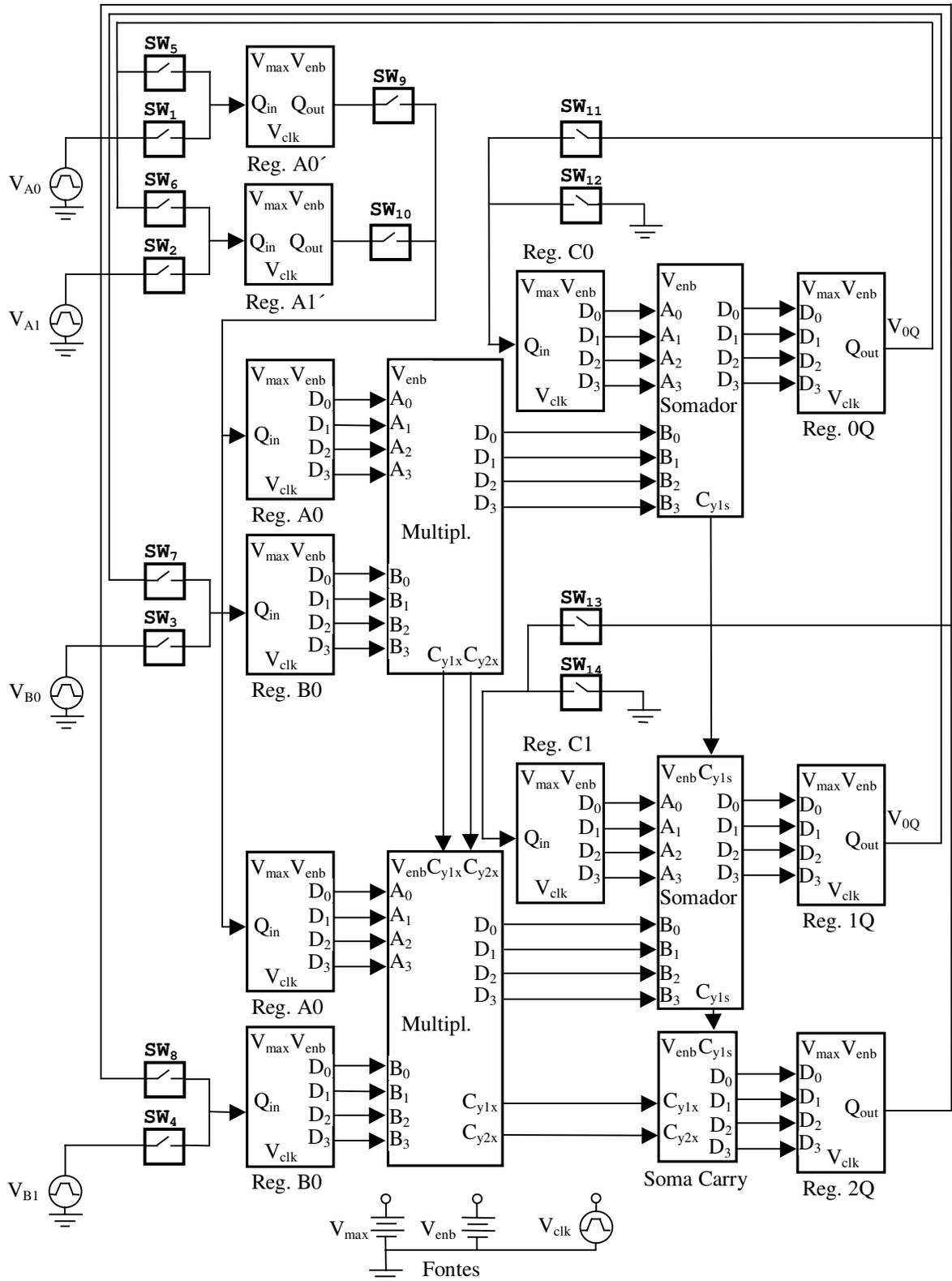


Figura 5.14 Circuito completo do Multiplicador com duas entrada de dois Dígitos Quaternários e saída de quatro Dígitos quaternários

5.4.2 Ciclo de Multiplicação

Foi considerado um período de *clock* de 100ns, ficando a sequência de operações da seguinte forma:

1º semi período (50ns): O sinal ‘Ciclo1A’ fecha as chaves Sw_1 , Sw_2 , Sw_3 , Sw_4 , Sw_{12} e Sw_{14} . Desta forma os dados são guardados nos registradores Reg_1 , Reg_2 , Reg_3 , Reg_4 , Reg_9 e Reg_{10} .

2º semi período: O sinal ‘Ciclo1B’ fecha a chave Sw_9 e passa o valor do registrador Reg_1 para o registrador Reg_5 e Reg_7 , durante esse semiperíodo de *clock* os registradores Reg_{11} , Reg_{12} e Reg_{13} recebem o valor da multiplicação do primeiro Dígito da entrada A pelos dois Dígitos da entrada B.

3º semi período: O sinal ‘Ciclo2A’ fecha as chaves Sw_5 , Sw_{11} e Sw_{13} passando os valores dos registradores Reg_{11} , Reg_{12} e Reg_{13} para os registradores Reg_1 , Reg_9 e Reg_{10} .

4º semi período: O sinal ‘Ciclo2B’ fecha a chave Sw_{10} e passa o valor do registrador Reg_2 para o registrador Reg_5 e Reg_7 , durante esse semiperíodo de *clock* os registradores Reg_{11} , Reg_{12} e Reg_{13} recebem o valor da multiplicação do segundo Dígito da entrada A pelos dois Dígitos da entrada B.

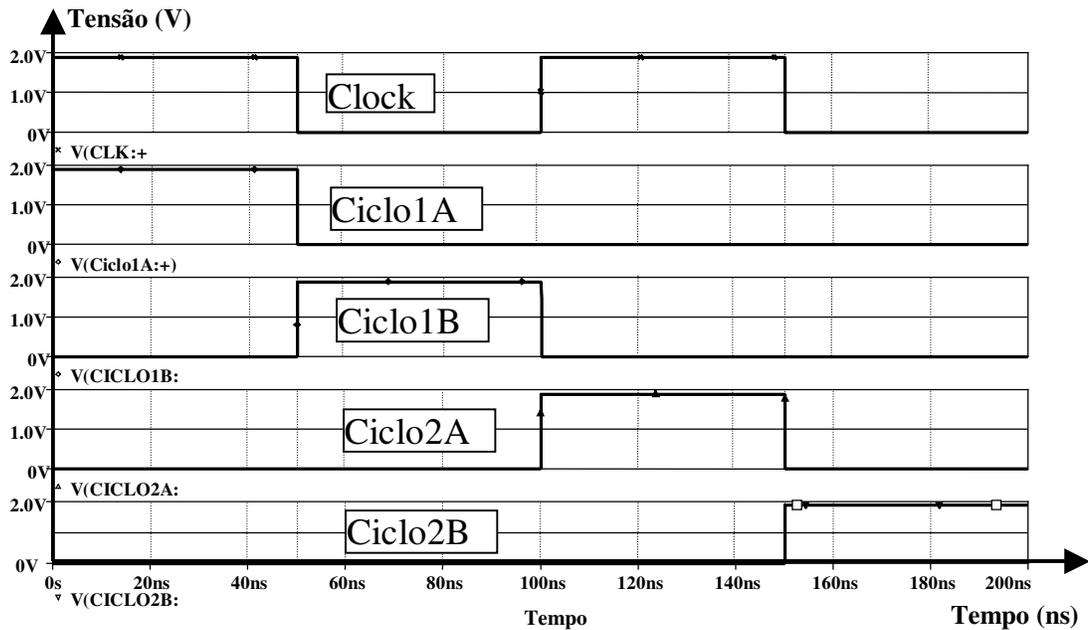


Figura 5.15 Sinais de *Clock*: Primeiro é o sinal de *Clock*, o segundo sinal fecha as chaves do primeiro período, o terceiro sinal fecha as chaves do segundo período, o quarto sinal fecha as chaves do terceiro período e o quinto sinal fecha as chaves do quarto período.

Para passar o *carry* entre blocos se utilizou a seguinte lógica:

Carry Mult	CyX1	CyX2
0	1	1
1	0	1
2	1	0

Carry Soma	CtS1
0	1
1	0

Figura 5.16 Tabela de Lógica do *Carry* para a Multiplicação e a Soma

Este circuito multiplicador de duas entradas quaternárias de dois Dígitos cada, com uma saída quaternária de quatro Dígitos e as chaves necessárias para fazer a multiplicação utilizou 1068 transistores com um consumo de potência médio de 16mW e com as seguintes vantagens:

- a) Utilização de transistores em suas dimensões mínimas de L e W
- b) Parâmetros dos transistores não críticos.
- c) Níveis de tensão precisos.
- d) Tempo de comutação relativamente rápido menor que 4ns.
- e) Dois ciclos de *Clock* para realizar a operação.
- f) Sempre será possível ajustar o nível das fontes para valores ‘ótimos’ de funcionamento.

Com a escolha de transistores mais adequados poder-se-ia utilizar o Módulo de três transistores no lugar de sete, e também pode-se eliminar os registradores Reg₃ e Reg₄ que estão duplicados pelos registradores Reg₆ e Reg₈, pode-se chegar a uma quantidade total de transistores para nosso multiplicador de aproximadamente 750 transistores.

No próximo capítulo será feita uma comparação do multiplicador desenvolvido neste texto com um multiplicador Binários utilizando os mesmos transistores.

Capítulo 6

Comparação com Sistemas Binários e Projeção

Neste capítulo será feita a comparação entre, o multiplicador proposto no capítulo 5, o multiplicador do Apêndice II e um multiplicador Binário.

6.1 Comparação com Sistema Binário.

A comparação será feita com um Multiplicador Digital de 4x4 bits binário, seu equivalente Multi-Valor Quaternário de 2x2 Dígitos Quaternário. Esta comparação somente foi feita em número de transistores e tempo de execução.

6.1.1 Multiplicador Binário.

A figura 6.1 mostra um circuito de um multiplicador binário de 4x4.

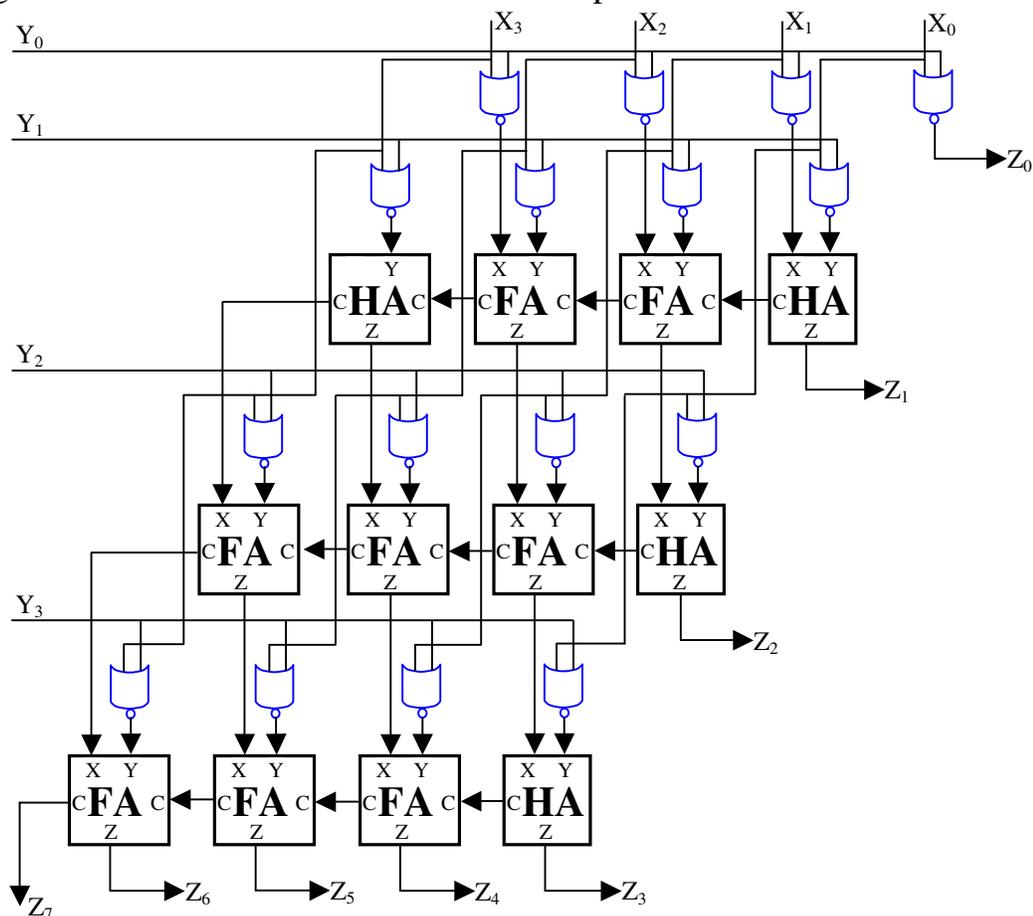


Figura 6.1 : Multiplicador binário de 4x4.

Para este circuito são necessária portas AND que executam a multiplicação dos bit's e somadores denominados como FA (*Full Adder*) e HA (*Half Adder*).

A figura 6.2 mostra um somador FA.

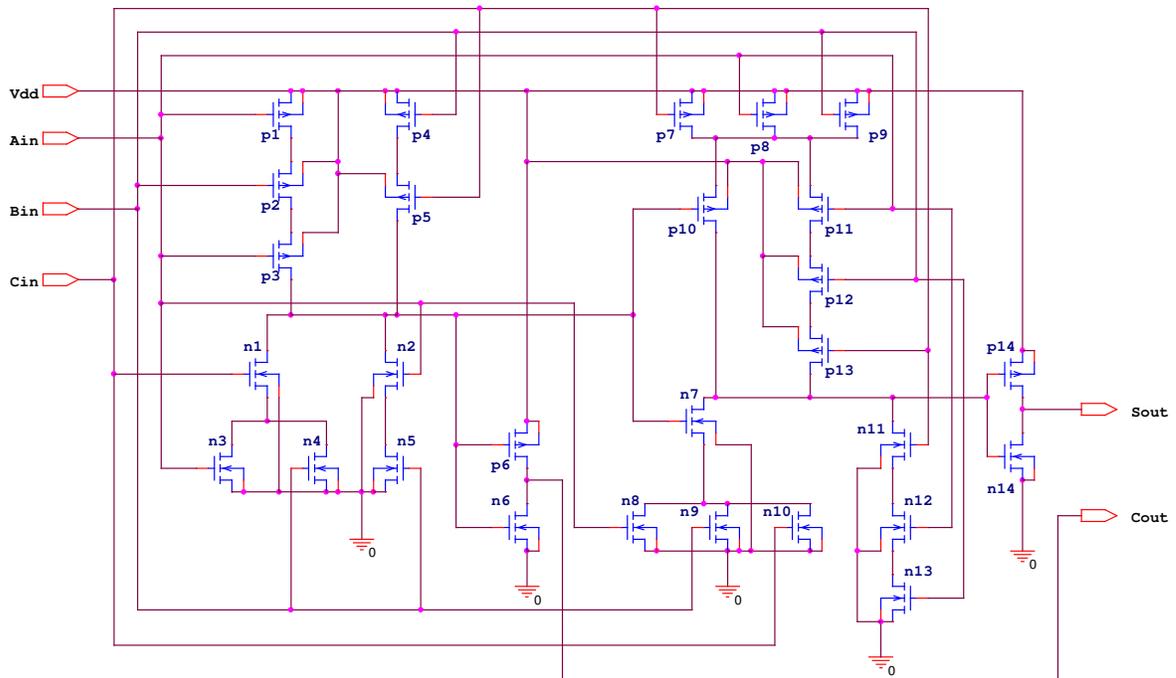


Figura 6.2 : Somador binário *Full Adder* de um bit.

Fazendo agora a simulação do circuito, teremos na Figura 6.3 o circuito utilizado para a simulação e nas figuras 6.4 e 6.5 as curvas que mostram os resultados da multiplicação e uma das comutações onde pode-se ver o atraso.

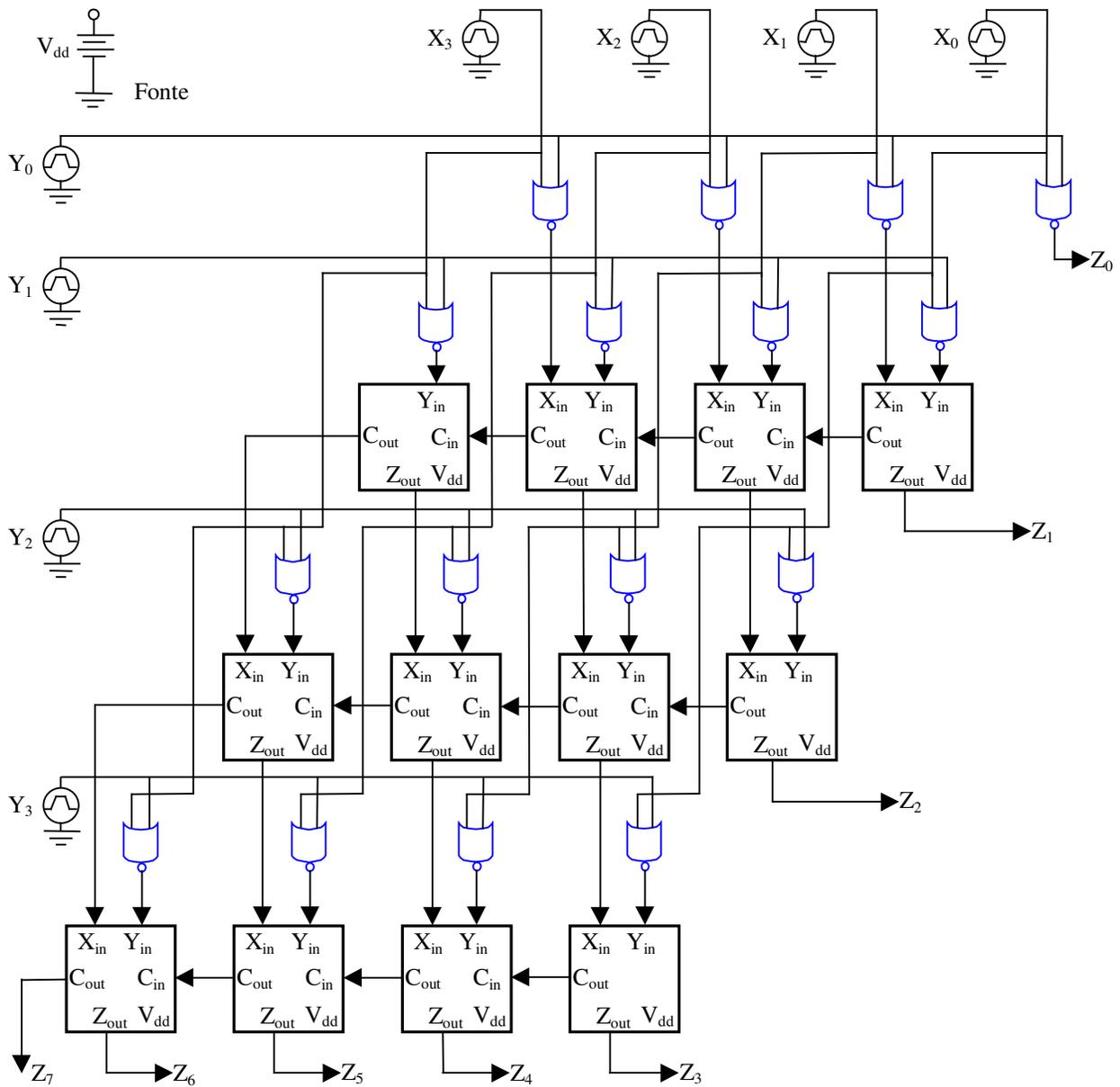


Figura 6.3 : Circuito Multiplicador utilizado na simulação.

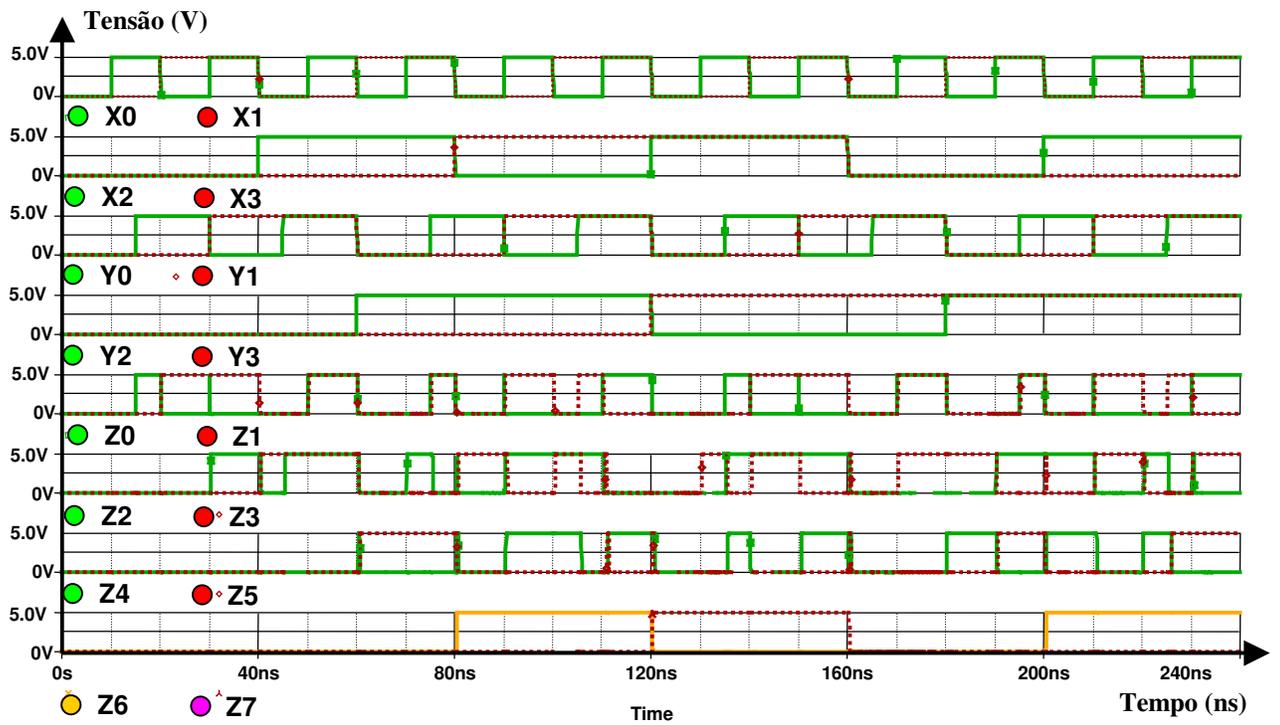


Figura 6.4 : Resultados da multiplicação.

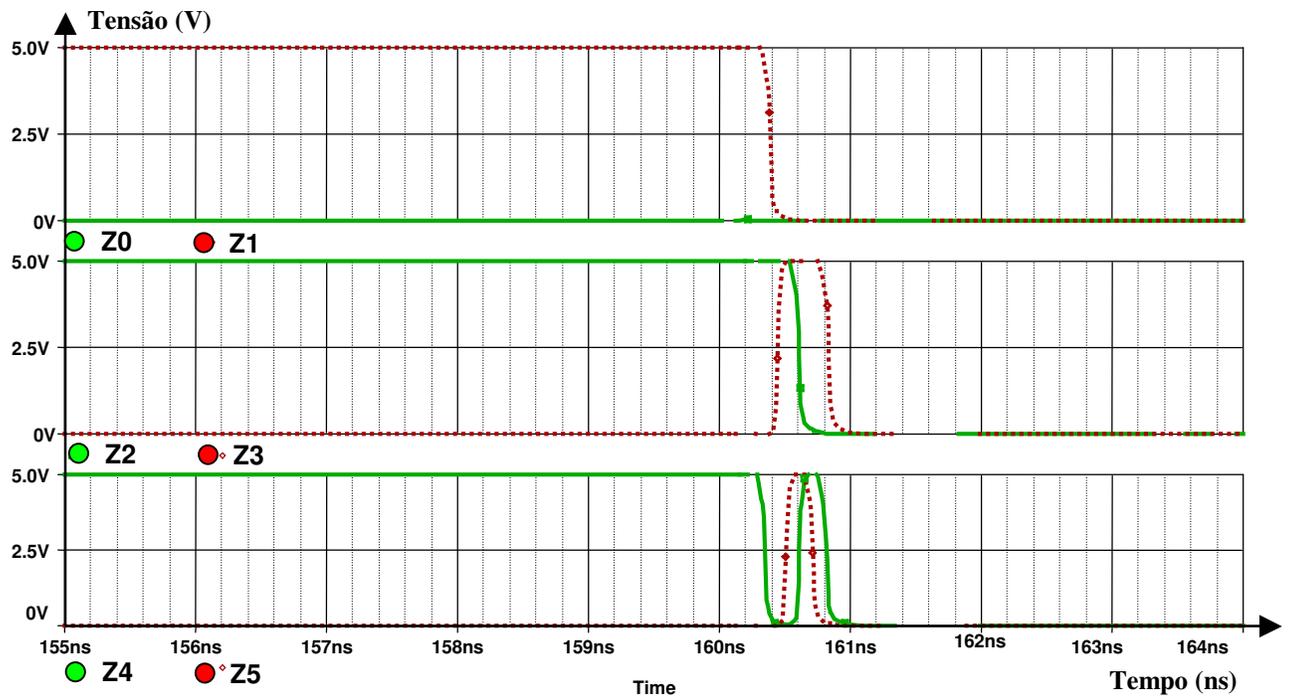


Figura 6.5 : Zoom mostrando o atraso na comutação.

Acrescentando no multiplicador binário os registradores de entrada, teremos uma quantidade de transistores de 608 e um atraso de 2ns.

A tabela de comparação ficará como mostrada na figura 6.6.

	Multiplicador Binário.	Multiplicador Quaternário Proposto.	Multiplicador Quaternário proposto no Apêndice II
Tempo de Multiplicação (ns)	2	4	2
Quantidade de Transistores	608	694	558

Figura 6.6 Comparação de Multiplicador Binário e Multiplicador Quaternário Proposto e o do Apêndice II.

Algumas considerações devem ser feitas para que os circuitos sejam totalmente equivalentes. Deveremos incluir no multiplicador binário um decodificador de quaternário para binário na entrada e um decodificador de binário para quaternário na saída. Isto leva aproximadamente 296 transistores, descontando os registradores de entrada binários teremos que aumentar do número de transistores em 104 o que nos dará um total de 712 transistores para o circuito binário equivalente e um atraso um pouco maior devido aos decodificadores. O tempo do multiplicador quaternário se mostra o dobro do binário por utilizar no nosso exemplo dois ciclos de *clock* para a multiplicação e não por ser menos rápidos que o binário. No caso do multiplicador do apêndice II pode verificarse que com um ciclo de *clock* os atrasos ficam comparáveis com o binário.

Em conclusão podemos dizer que os tempos e a quantidade de transistores são da mesma ordem, mas o circuito quaternário ocupará a metade da área em barramentos, e

sabemos que a área ocupada em barramentos é consideravelmente maior que a ocupada por circuitos ativos dentro de um *chip* e também em placas de circuitos impressos.

6.2 Projeção

A ideia exposta neste trabalho abre um caminho em procura de transistores com certas características que permitam aumentar o número de níveis, que poderá chegar a Decimal, Hexadecimal ou de 32, 64, 128, etc... o objetivo será agora não somente de velocidade aumentando a frequência de *clock*, mas também de quantidade de níveis. Como exemplo segue abaixo um circuito com nossos transístores com 10 níveis (decimal):

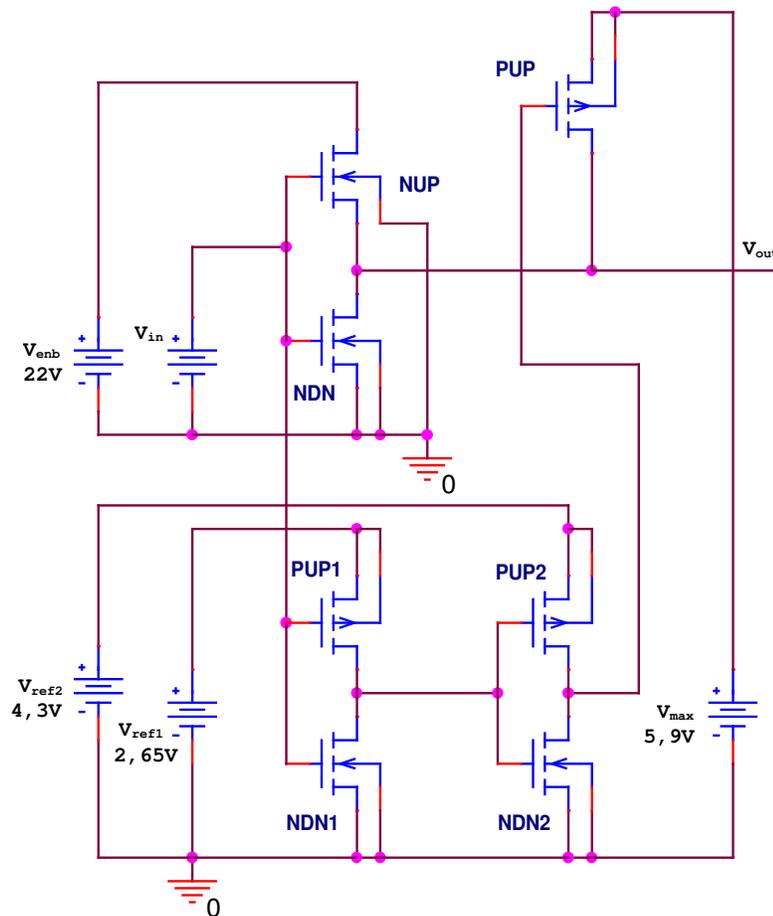


Figura 6.7 Circuito de um Módulo com 10 níveis lógicos.

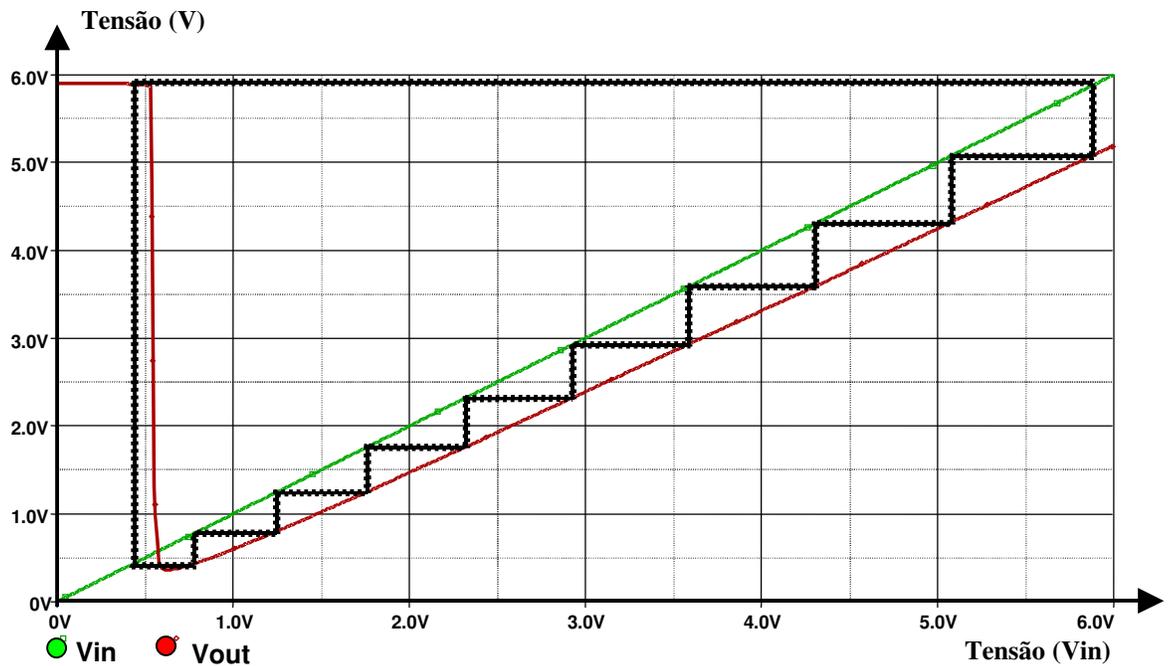


Figura 6.8 Curva de Entrada e Saída de um circuito com 10 níveis lógicos

Este circuito utiliza quatro transistores para chavear o nível zero para o máximo, isto pode ser eliminado utilizando um par de transistores nos quais o V_{TH0} do PMOS seja o suficientemente alto e o V_{TH0} do NMOS seja o suficientemente baixo para que seja possível a utilização do Módulo N com três transistores.

Para um circuito integrado com muitos transistores (como um microprocessador) pode-se ter um circuito dentro do próprio *chip* que defina as tensões de referência e alimentação que levem ao *chip* a um ponto de funcionamento ótimo, e essas tensões serão referência para fontes externas que alimentam o *chip*, compensando desta forma as variações de processo de fabricação e temperatura de trabalho. Desta forma os níveis lógicos de tensões internos serão diferentes para cada *chip*, mas poderá haver uma padronização das tensões de barramentos e os CI's trabalhar com tensões padronizadas nas saídas e entrada de barramentos.

6.2 Conclusão

Pode-se verificar que das características desejadas definidas no início deste trabalho, foram alcançadas com um grau excelente nos seguintes pontos:

- 1- Robustez.
- 2- Níveis de tensões definidos.
- 3- Dimensões mínimas dos componentes.
- 4- Alta frequência de operação.
- 5- Pouca dependência dos parâmetros dos componentes.
- 6- Quantidade mínima de tipos de componentes.
- 7- Facilidade de serem acoplados sem sofrer sobrecargas.
- 8- Quantidade mínima de conexões.
- 9- Circuitos expansíveis para N-valores.
- 10-Simplicidade.

E com um grau aceitável:

- 1- Quantidade mínima de componentes por circuito.
- 2- Baixo consumo de potência

Para finalizar este trabalho pode-se destacar que o conceito de se ter um registrador cíclico, que além de armazenar um dado em níveis de Multi-valor com precisão, fornece todos os outros níveis, decodificando o valor armazenado e permitindo que se obtenha a saída desejada, seja qualquer um dos valores lógicos sem perda de precisão. O registrador permitirá a criação de circuitos lógicos tais como Contadores, Toggle Switch, Registrador de Deslocamento, Flip Flop em “n” níveis, Deslocador Topo e Deslocador Base, Conversores D/A e Conversores A/D, etc...

Apêndice I

Modelos PSpice utilizados neste trabalho

I.1 Modelo do NMOS

```
.model nmos035H NMOS LEVEL=7
* -----
***** SIMULATION PARAMETERS
* -----
* format   : PSPICE
* model    : MOS BSIM3v3
* process  : CS[ADFI]
* revision : 2;
* extracted : CSA C61417; 2000-12; ese(5487)
* doc#     : 9933016 REV_2
* -----
*          TYPICAL MEAN CONDITION
* -----
*   *** Flags ***
+MOBMOD =1.000e+00 CAPMOD =2.000e+00
+NLEV  =0      NOIMOD =1.000e+00
*   *** Threshold voltage related model parameters ***
+K1    =6.044e-01
+K2    =2.945e-03 K3    =-1.72e+00 K3B   =6.325e-01
+NCH   =2.310e+17 VTH0  =4.655e-01
+VOFF  =-5.72e-02 DVT0  =2.227e+01 DVT1  =1.051e+00
+DVT2  =3.393e-03 KETA  =-6.21e-04
+PSCBE1 =2.756e+08 PSCBE2 =9.645e-06
+DVT0W =0.000e+00 DVT1W =0.000e+00 DVT2W =0.000e+00
*   *** Mobility related model parameters ***
+UA    =1.000e-12 UB    =1.723e-18 UC    =5.756e-11
+U0    =4.035e+02
*   *** Subthreshold related parameters ***
+DSUB  =5.000e-01 ETA0  =3.085e-02 ETAB  =-3.95e-02
+NFACTOR=1.119e-01
*   *** Saturation related parameters ***
+EM    =4.100e+07 PCLM  =6.831e-01
+PDIBLC1=1.076e-01 PDIBLC2=1.453e-03 DROUT =5.000e-01
+A0    =2.208e+00 A1    =0.000e+00 A2    =1.000e+00
```

+PVAG =0.000e+00 VSAT =1.178e+05 AGS =2.490e-01
 +B0 =-1.76e-08 B1 =0.000e+00 DELTA =1.000e-02
 +PDIBLCB=2.583e-01
 * *** Geometry modulation related parameters ***
 +W0 =1.184e-07 DLC =4.000e-09
 +DWB =0.000e+00 DWG =0.000e+00
 +LL =0.000e+00 LW =0.000e+00 LWL =0.000e+00
 +LLN =1.000e+00 LWN =1.000e+00 WL =0.000e+00
 +WW =0.000e+00 WWL =0.000e+00 WLN =1.000e+00
 +WWN =1.000e+00
 * *** Temperature effect parameters ***
 +AT =3.300e+04 UTE =-1.80e+00
 +KT1 =-3.30e-01 KT2 =2.200e-02 KT1L =0.000e+00
 +UA1 =0.000e+00 UB1 =0.000e+00 UC1 =0.000e+00
 +PRT =0.000e+00
 * *** Overlap capacitance related and dynamic model parameters ***
 +CGDO =1.120e-10 CGSO =1.120e-10 CGBO =1.100e-10
 +CGDL =1.350e-10 CGSL =1.350e-10 CKAPPA =6.000e-01
 +CF =0.000e+00 ELM =5.000e+00
 +XPART =1.000e+00 CLC =1.000e-15 CLE =6.000e-01
 * *** Parasitic resistance and capacitance related model parameters ***
 +RDSW =6.043e+02
 +CDSC =0.000e+00 CDSCB =0.000e+00 CDSCD =8.448e-05
 +PRWB =0.000e+00 PRWG =0.000e+00 CIT =1.000e-03
 * *** Process and parameters extraction related model parameters ***
 +TOX =7.700e-09 NGATE =0.000e+00
 +NLX =1.918e-07
 * *** Substrate current related model parameters ***
 +ALPHA0 =0.000e+00 BETA0 =3.000e+01
 * *** Noise effect related model parameters ***
 +AF =1.400e+00 KF =2.810e-27 EF =1.000e+00
 +NOIA =1.000e+20 NOIB =5.000e+04 NOIC =-1.40e-12
 * *** Common extrinsic model parameters ***
 +LINT =-1.67e-08 WINT =2.676e-08 XJ =3.000e-07
 +RSH =8.200e+01 JS =2.000e-05
 +CJ =9.300e-04 CJSW =2.800e-10
 +CBD =0.000e+00 CBS =0.000e+00 IS =0.000e+00
 +MJ =3.100e-01 N =1.000e+00 MJSW =1.900e-01
 +PB =6.900e-01 TT =0.000e+00
 +PBSW =6.900e-01

I.2 Modelo do PMOS

```
.model pmos035 PMOS LEVEL=7
* -----
***** SIMULATION PARAMETERS *****
* -----
* format : PSPICE
* model : MOS BSIM3v3
* process : CS[ADFI]
* revision : 2;
* extracted : CSA C61417 A64770; 2000-12; ese(5487)
* doc# : 9933016 REV_2
* -----
* TYPICAL MEAN CONDITION
* -----
* *** Flags ***
+MOBMOD =1.000e+00 CAPMOD =2.000e+00
+NLEV =0 NOIMOD =1.000e+00
* *** Threshold voltage related model parameters ***
+K1 =5.675e-01
+K2 =-4.39e-02 K3 =4.540e+00 K3B =-8.52e-01
+NCH =1.032e+17 VTH0 =-6.17e-01
+VOFF =-1.13e-01 DVT0 =1.482e+00 DVT1 =3.884e-01
+DVT2 =-1.15e-02 KETA =-2.56e-02
+PSCBE1 =1.000e+09 PSCBE2 =1.000e-08
+DVT0W =0.000e+00 DVT1W =0.000e+00 DVT2W =0.000e+00
* *** Mobility related model parameters ***
+UA =2.120e-10 UB =8.290e-19 UC =-5.28e-11
+U0 =1.296e+02
* *** Subthreshold related parameters ***
+DSUB =5.000e-01 ETA0 =2.293e-01 ETAB =-3.92e-03
+NFACTOR=8.237e-01
* *** Saturation related parameters ***
+EM =4.100e+07 PCLM =2.979e+00
+PDIBLC1=3.310e-02 PDIBLC2=1.000e-09 DROUT =5.000e-01
+A0 =1.423e+00 A1 =0.000e+00 A2 =1.000e+00
+PVAG =0.000e+00 VSAT =2.000e+05 AGS =3.482e-01
+B0 =2.719e-07 B1 =0.000e+00 DELTA =1.000e-02
+PDIBLCB=-1.78e-02
```

```

*      *** Geometry modulation related parameters ***
+W0   =4.894e-08 DLC   =4.000e-09
+DWB  =0.000e+00 DWG  =0.000e+00
+LL   =0.000e+00 LW   =0.000e+00 LWL  =0.000e+00
+LLN  =1.000e+00 LWN  =1.000e+00 WL   =0.000e+00
+WW   =0.000e+00 WWL  =0.000e+00 WLN  =1.000e+00
+WWN  =1.000e+00
*      *** Temperature effect parameters ***
+AT   =3.300e+04 UTE  =-1.35e+00
+KT1  =-5.70e-01 KT2  =2.200e-02 KT1L  =0.000e+00
+UA1  =0.000e+00 UB1  =0.000e+00 UC1  =0.000e+00
+PRT  =0.000e+00
*      *** Overlap capacitance related and dynamic model parameters ***
+CGDO =7.420e-11 CGSO =7.420e-11 CGBO  =1.100e-10
+CGDL =1.290e-10 CGSL =1.290e-10 CKAPPA =6.000e-01
+CF   =0.000e+00 ELM  =5.000e+00
+XPART =1.000e+00 CLC  =1.000e-15 CLE  =6.000e-01
*      *** Parasitic resistance and capacitance related model parameters ***
+RDSW =1.853e+03
+CDSC =6.994e-04 CDSCB =2.943e-04 CDSCD =1.970e-04
+PRWB =0.000e+00 PRWG =0.000e+00 CIT  =1.173e-04
*      *** Process and parameters extraction related model parameters ***
+TOX  =7.700e-09 NGATE =0.000e+00
+NLX  =1.770e-07
*      *** Substrate current related model parameters ***
+ALPHA0 =0.000e+00 BETA0 =3.000e+01
*      *** Noise effect related model parameters ***
+AF   =1.290e+00 KF    =1.090e-27 EF    =1.000e+00
+NOIA =1.000e+20 NOIB  =5.000e+04 NOIC  =-1.40e-12
*      *** Common extrinsic model parameters ***
+LINT =-8.14e-08 WINT  =3.845e-08 XJ    =3.000e-07
+RSH  =1.560e+02 JS    =2.000e-05
+CJ   =1.420e-03 CJSW  =3.800e-10
+CBD  =0.000e+00 CBS   =0.000e+00 IS    =0.000e+00
+MJ   =5.500e-01 N     =1.000e+00 MJSW  =3.900e-01
+PB   =1.020e+00 TT    =0.000e+00
+PBSW =1.020e+00

```

Apêndice II

II.1 Alternativa de Circuito Multiplicador

Será descrito a seguir uma alternativa para um multiplicador utilizando o conceito de matriz e o registrador descrito neste texto no Capítulo 2 que decodifica a entrada Multivalor em todos o seus níveis.

Se se considera uma Tabela Verdade de duas entradas como por exemplo o caso da Tabela Verdade da Multiplicação da Figura 4.1 do Capítulo 4 .

X0			B		
Cy=0	0	1	2	3	
	0	0	0	0	0
A	1	0	1	2	3
	2	0	2	0	2
	3	0	3	2	1

Figura II.1 Tabela Verdade do Valor do Multiplicador com $Carry = 0$

Considerando também as quatro saídas de cada registrador, poderá ser montada uma matriz onde para cada entrada somente um termo da matriz terá os dois valores em zero, assim poderá definir-se para essa condição o valor de saída da matriz, qualquer que seja a matriz. Desta forma será possível implementar qualquer função dada por uma matriz que será expandida para N entradas, sempre que a coincidência dos zeros encontre como resultado um único ponto de intersecção na matriz..

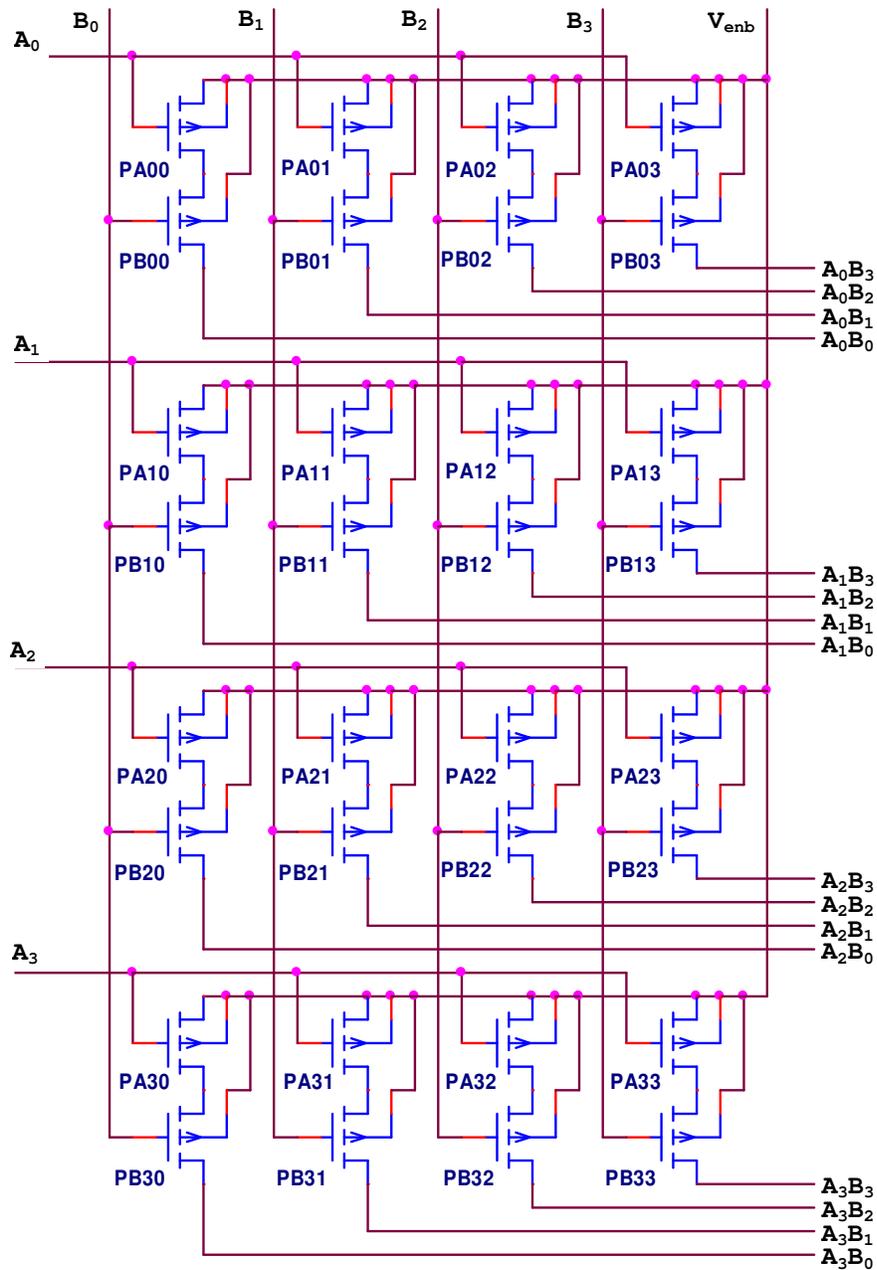


Figura II.2 Matriz quaternária com transistores tipo PMOS.

Os transistores Tipo P da Matriz são os especificados no Apêndice I mas com o $V_{TH0} = -1,6V$, e somente conduzem quando no *Gate* tem entrada zero, nos outros valores o transistor estará cortado. Desta forma a Tensão V_{ENB} somente estará presente na saída em que os dois transistores estejam conduzindo.

É de notar que poderão ser feitas as simplificações que são usadas nos sistemas binários, agrupando neste caso os valores iguais em linhas ou colunas. Para o caso da Multiplicação a Matriz ficará da forma mostrada na Figura II.3.

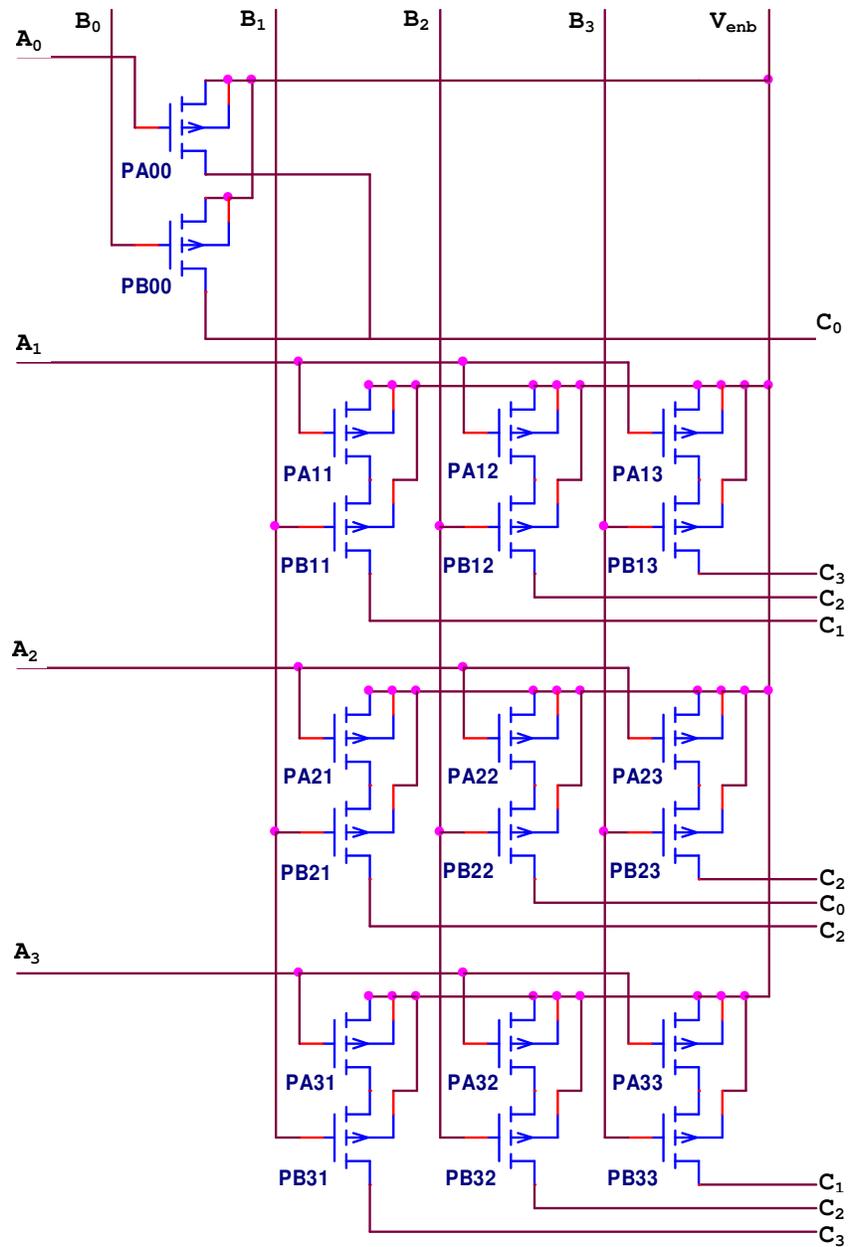


Figura II.3 Matriz correspondente à Tabela Verdade da Multiplicação.

Como pode notar-se houve uma redução de 12 transistores na simplificação da 1ª coluna e da 1ª linha.

A continuação será acrescentada à Matriz na saída uns transistores de *Pull-down* que levarão os valores da saída que não fora o selecionado para o valor de zero, deixando somente o valor de saída selecionado com a tensão V_{enb} .

O bloco que mostra esta função com entradas e saída decodificada é mostrado na Figura II.4, e o circuito esta mostrado na Figura II.5.

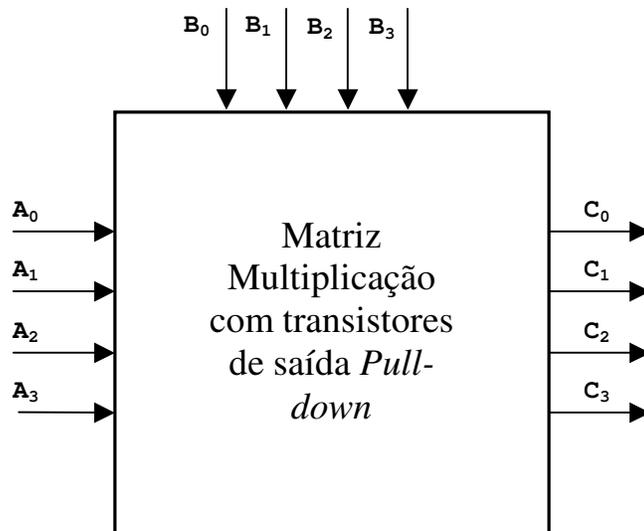


Figura II.4 Bloco da Matriz de Multiplicação com entradas e saída decodificada e com transistores de saída *Pull-down*

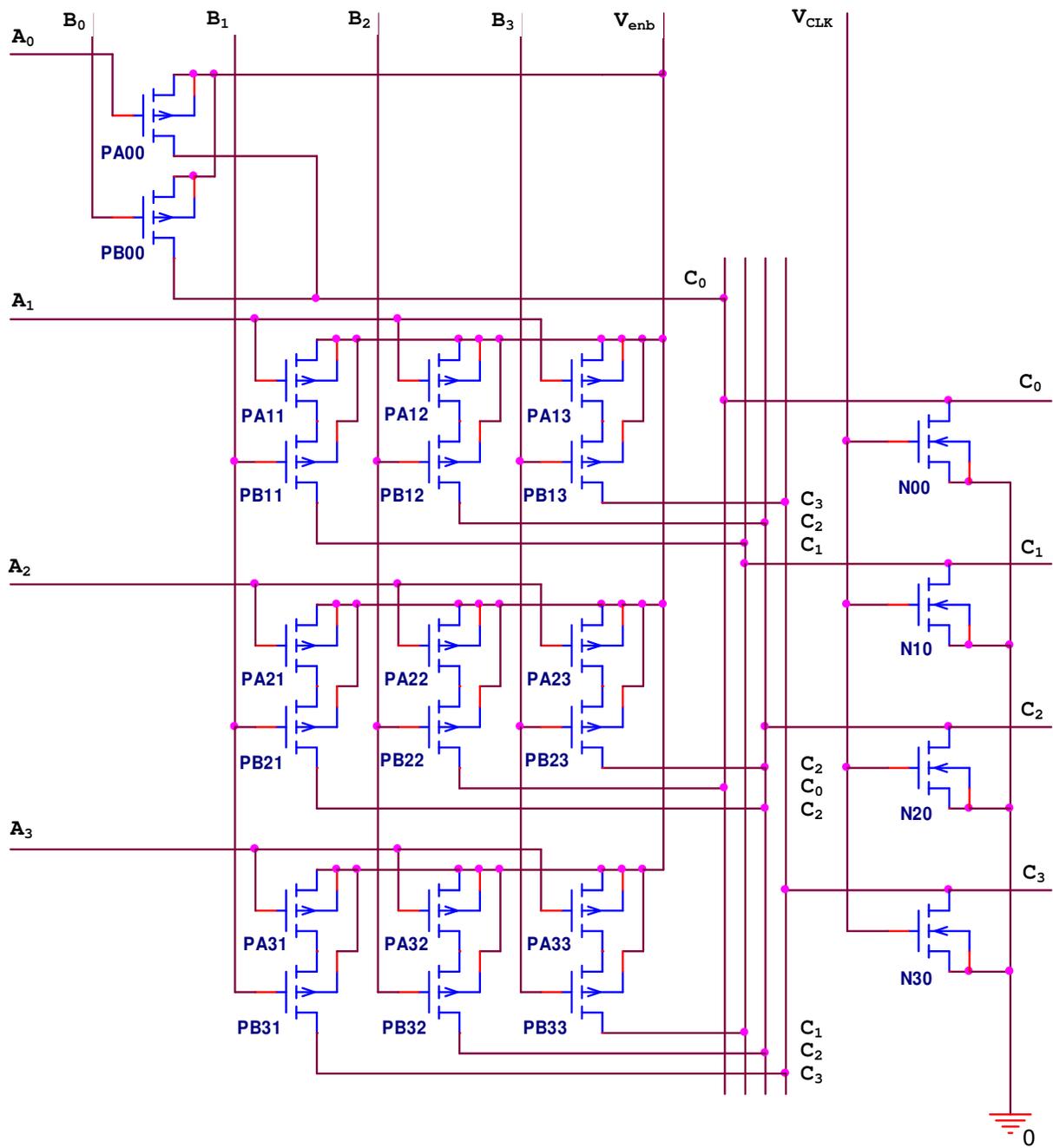


Figura II.5 Matriz Multiplicação com transistores de saída *Pull-down*

Com a finalidade de a Matriz não consumir potência as tensões de V_{ENB} e V_{CLK} serão alternadas com o sinal de *Clock*, desta forma quando a tensão V_{CLK} estiver alta e os transistores NMOS conduzindo, a tensão V_{ENB} estará baixa (zero volt) não permitindo que circule corrente pelo circuito e vice-versa quando V_{ENB} estiver alta os transistores NMOS estarão cortados.

Como pode ver-se a saída está invertida, ou seja quem define o valor será a linha que tenha a tensão alta, desta forma a próxima matriz deverá ser invertida (com transistores tipo NMOS e terá as saídas com *pull-up* com transistores tipo PMOS).

Na Figura II.6 e II.7 pode-se ver o caso de uma matriz Somadora com transistores tipo NMOS e transistores de *Pull-up* na saída.

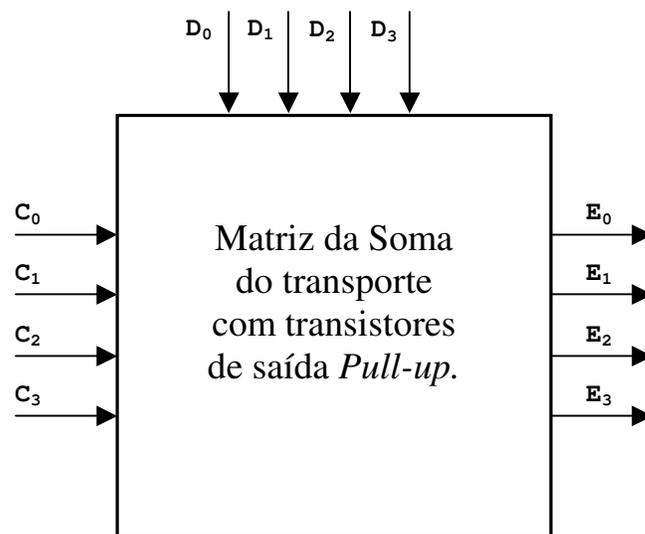


Figura II.6 Bloco de uma Matriz Somadora com transistores tipo NMOS e com transistores de saída *Pull-up* .

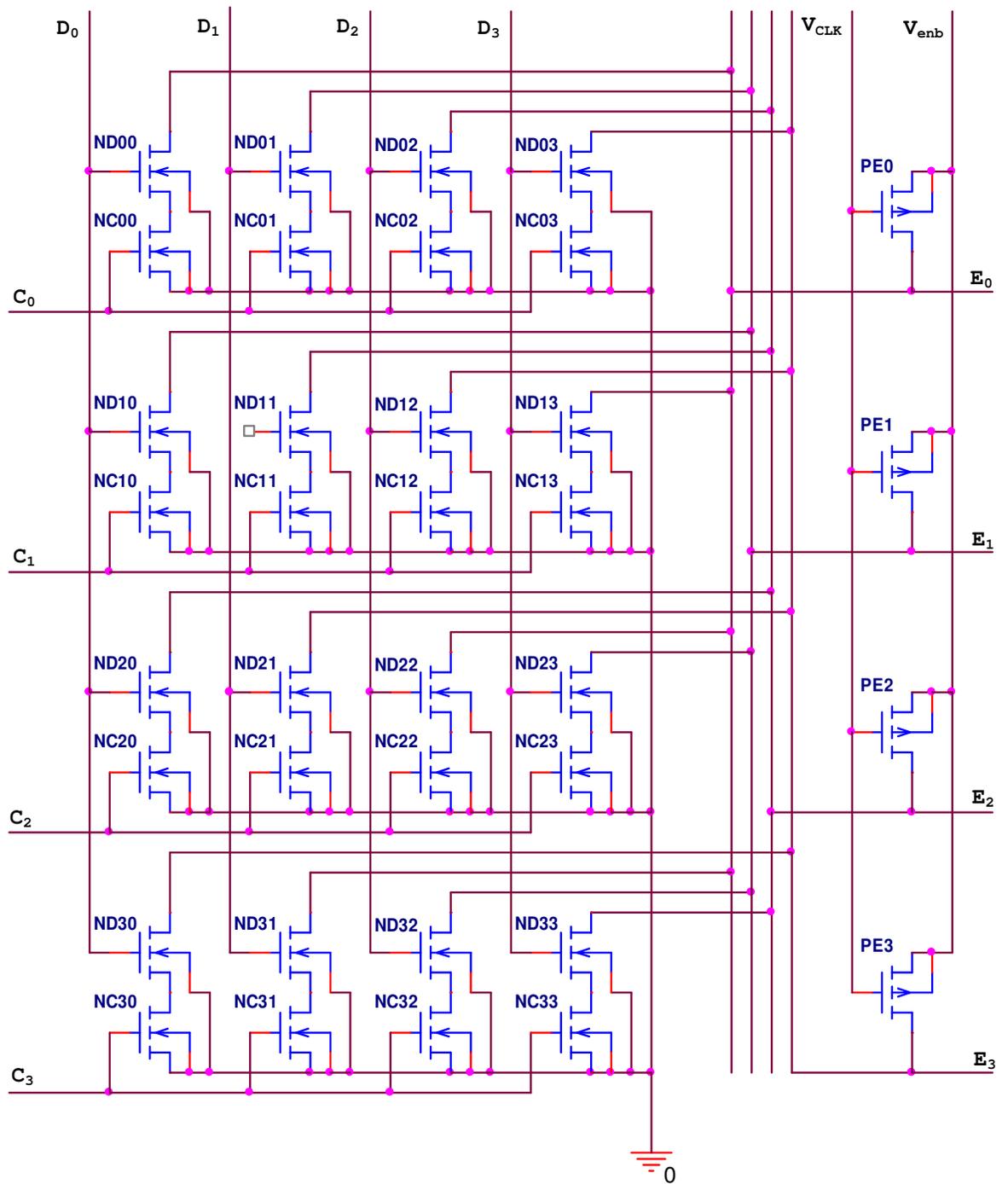


Figura II.7 Circuito de uma Matriz Somadora com transistores tipo NMOS e com transistores de saída *Pull-up*

No final da operação deverá ser feita a Codificação para os níveis quaternário de saída.

Uma das alternativas será usando chaves duplas e níveis de tensões que poderão ser os níveis do circuito (para conexões internas do *chip*) ou níveis padronizados de barramento para conexões com outros componentes.

O circuito de Saída poderá ser como mostrado na Figura II.8.

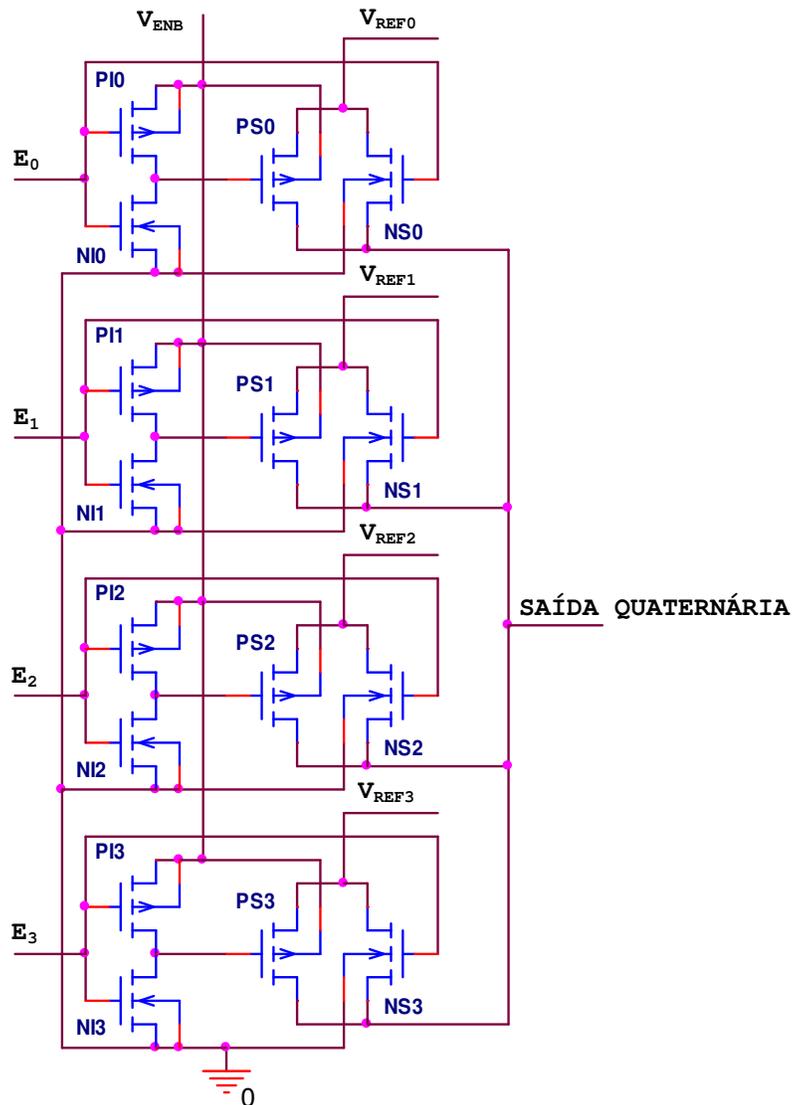


Figura II.8 Circuito de Saída, codificação para os níveis QUATERNÁRIOS.

Com estes circuitos poderá ser elaborado um multiplicador com operação em um só ciclo de *Clock*, como mostra na Figura II.9.

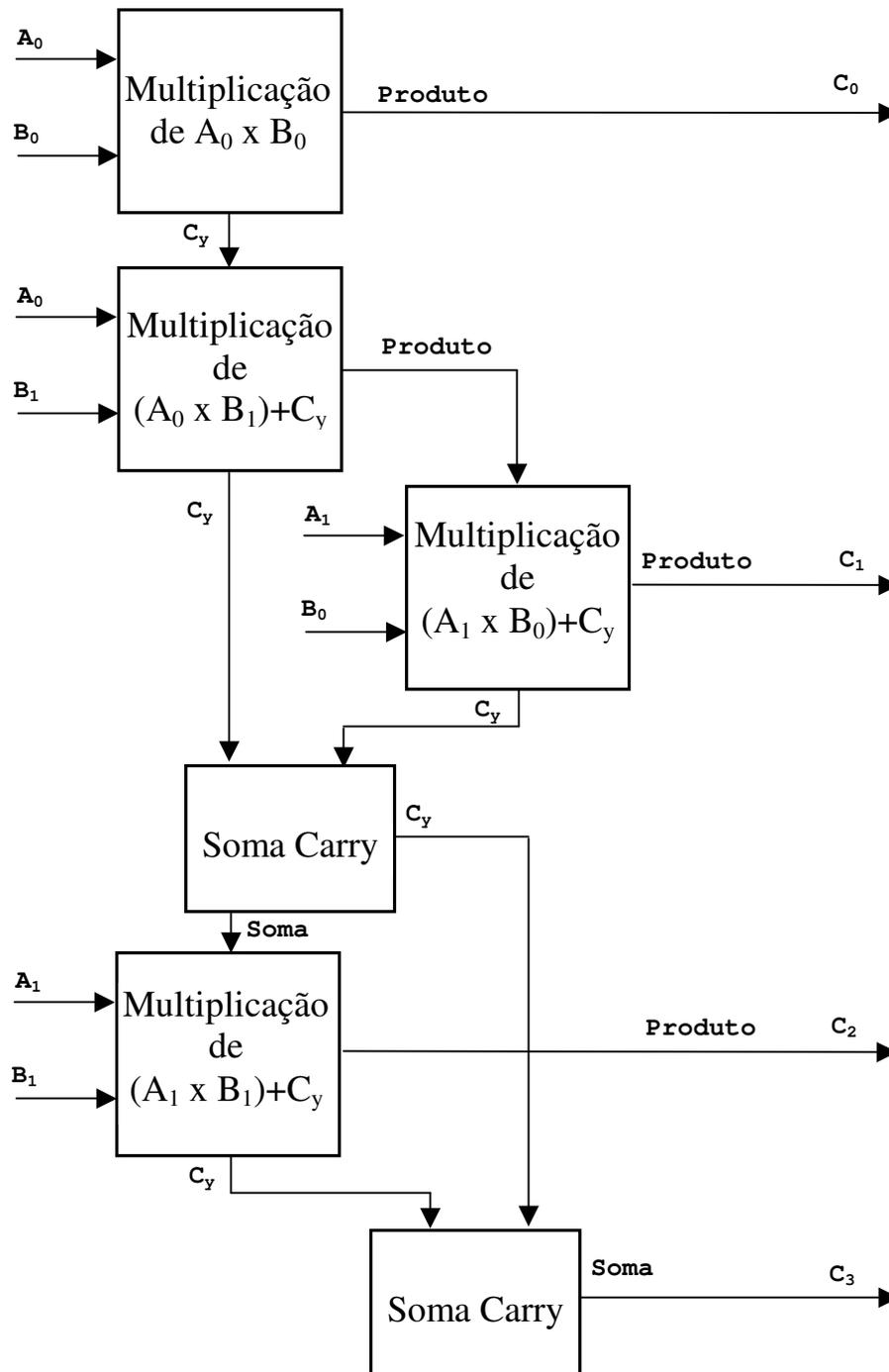


Figura II.9 Diagrama em bloco de um Circuito Multiplicador Digital QUATERNÁRIO de dois Dígitos de entrada (A_0, A_1 e B_0, B_1) e Saída de quatro Dígitos (C_0, C_1, C_2 e C_3).

No diagrama em bloco da Figura II.9 o bloco mais complexo é o que multiplica $(A_1 \times B_0)$ e soma o resultado da multiplicação de $(A_0 \times B_1)$ e tem 105 transistores, este bloco é mostrado na Figura II.15 e simulado mostrando as curvas nas Figuras II.16 e II.17.

Será necessário ter Registradores na entrada que Decodifiquem os valores QUATERNÁRIOS e Codificadores na saída que transformem os resultados em valores QUATERNÁRIOS. Como Registradores podem ser utilizados os propostos no Capítulo 2, considerando três componentes por Módulo.

Se consideramos um bloco de um multiplicador como o descrito na Figura II.10 que tenha quatro linhas de entrada de valores QUATERNÁRIOS (A_0, A_1, B_0 e B_1) e quatro linhas de saída de Valores QUATERNÁRIOS (C_0, C_1, C_2 e C_3) podemos calcular a quantidade de transistores necessárias para implementá-lo.

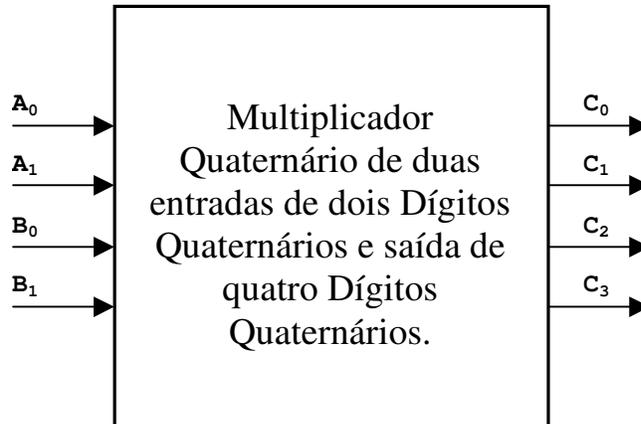


Figura II.10 Multiplicador Digital QUATERNÁRIO de duas entradas de dois Dígitos QUATERNÁRIOS (A_0, A_1 e B_0, B_1) e saída de quatro Dígitos QUATERNÁRIOS (C_0, C_1, C_2 e C_3).

O Multiplicador da Figura II.10 com registradores de entrada e codificadores de saída utiliza 520 transistores e executa a operação em um ciclo de *clock*.

A continuação nas Figura II.11, II.12, II.13, II.14 e II.15 são mostrados os circuitos do bloco multiplicador de um Dígito com a soma do transporte ou do *carry* e a saída do produto mais o *carry*, e nas Figuras II.16 e II.17 as curvas de simulação onde pode verificar-se o atraso na resposta.

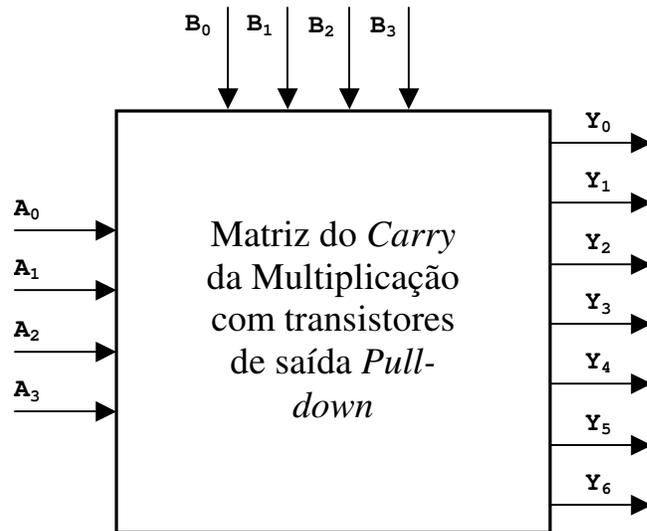


Figura II.11 Bloco da Matriz do *Carry* da Multiplicação com transistores de saída *Pull-down*.

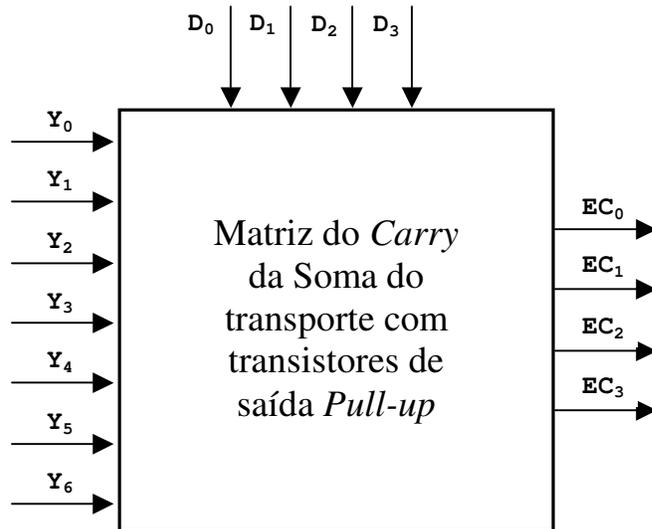


Figura II.12 Bloco da Matriz do *Carry* da Soma do transporte com transistores de saída *Pull-up*.

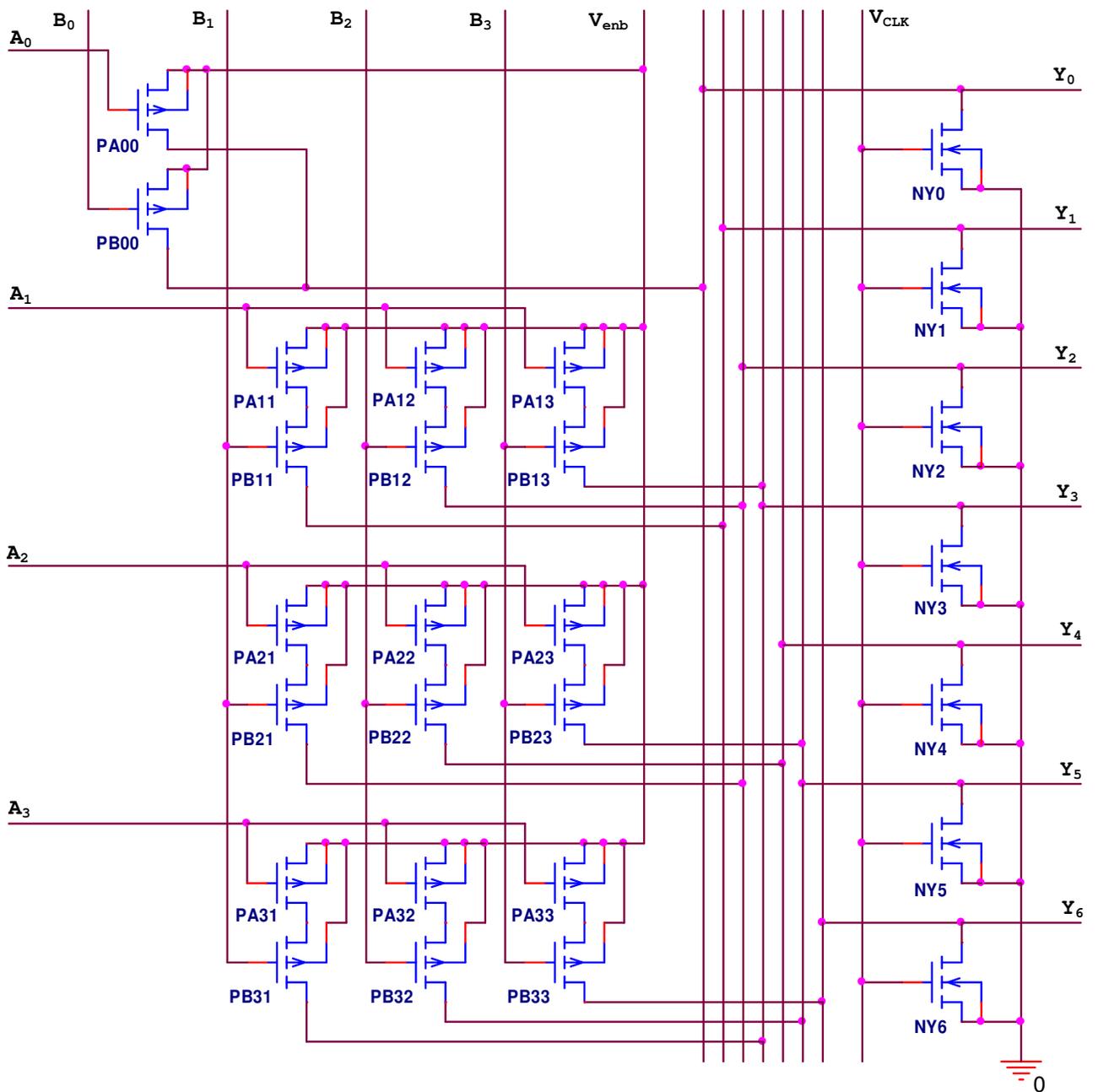


Figura II.13 Matriz do Carry da Multiplicação com transistores de saída *Pull-down*.

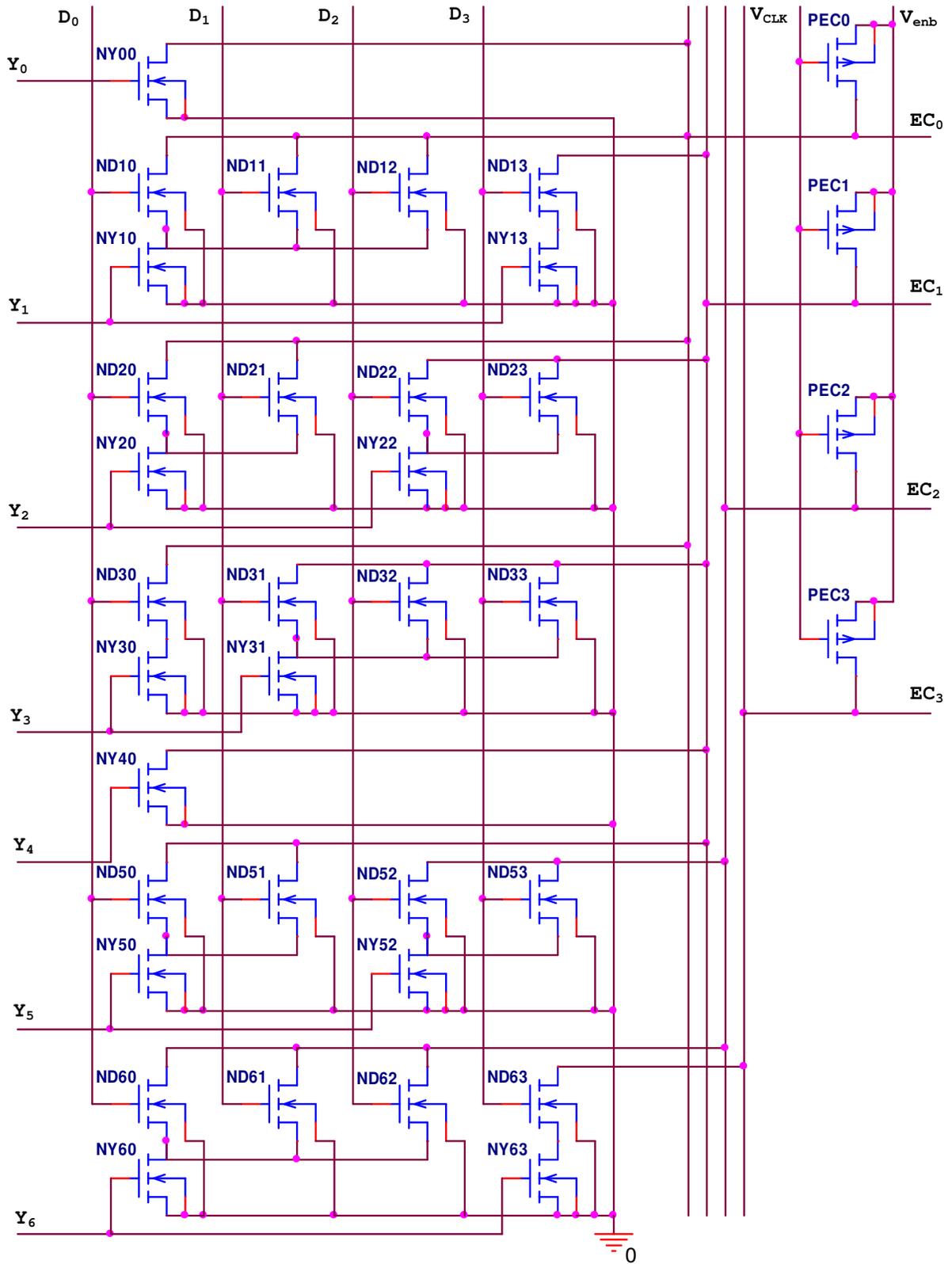


Figura II.14 Matriz do *Carry* da Soma do transporte com transistores de saída *Pull-up*.

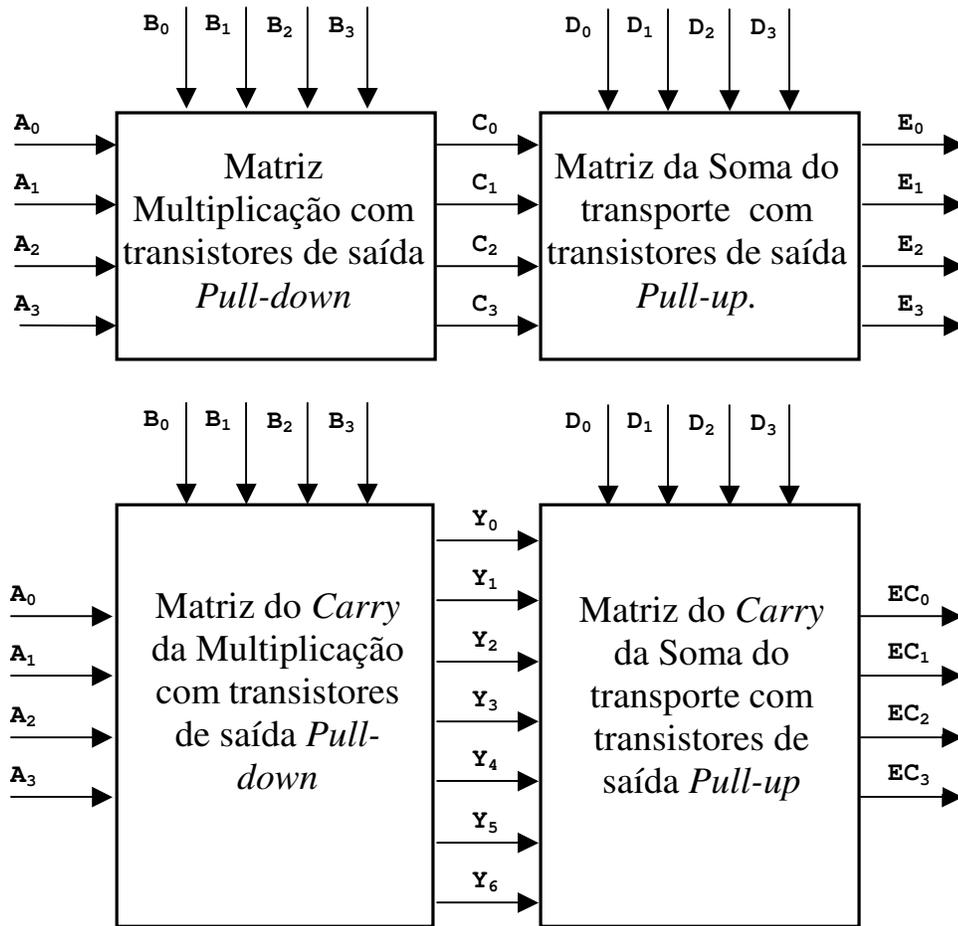


Figura II.15 Diagrama em bloco do circuito de teste do estagio de multiplicação DECODIFICADA de um DÍGITO.

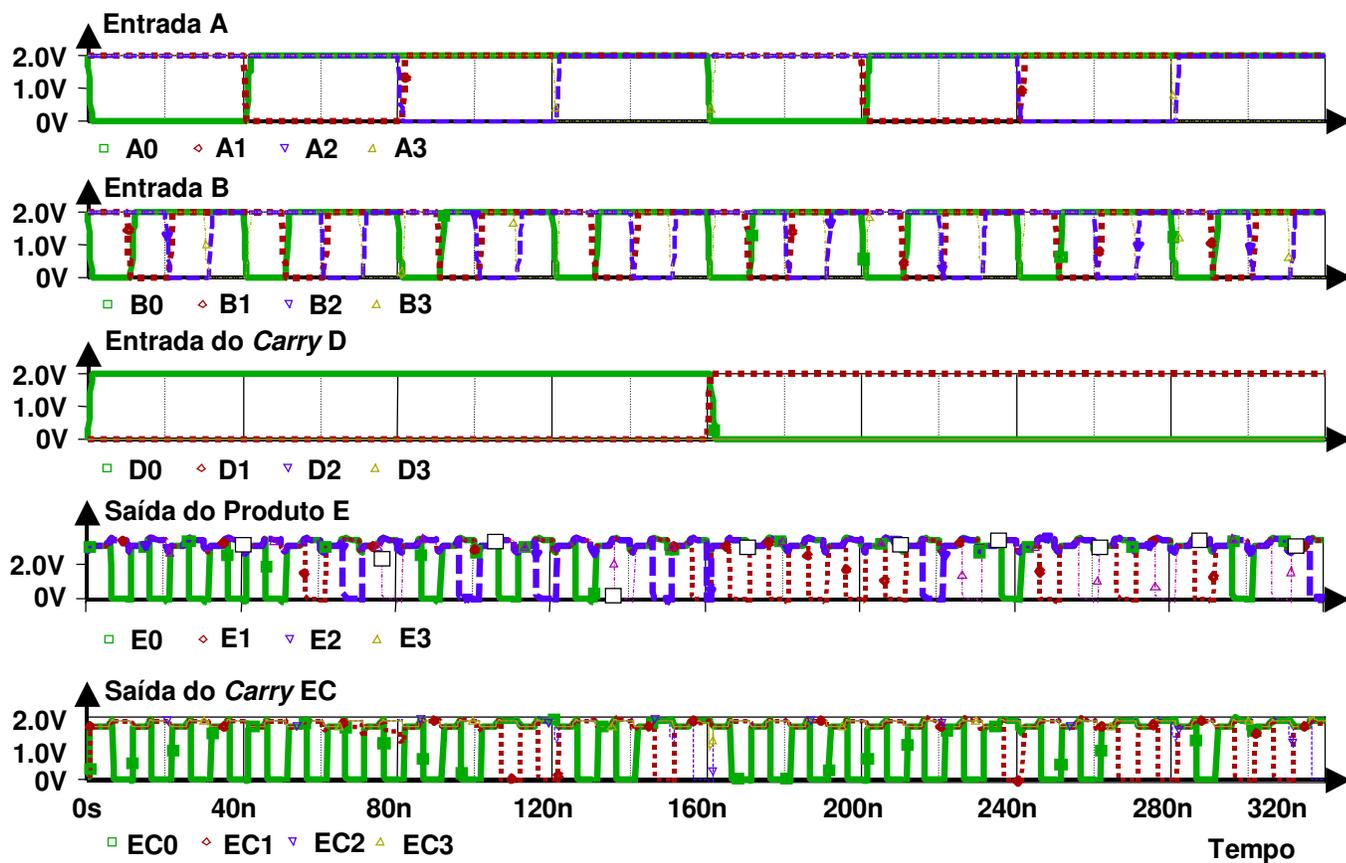


Figura II.16 Curvas do Circuito da Figura II.15. A 1º e 2º curvas são entradas A e B, a 3º curva é o Carry D de entrada, a 4º o Resultado do produto E e a 5º o Carry de saída

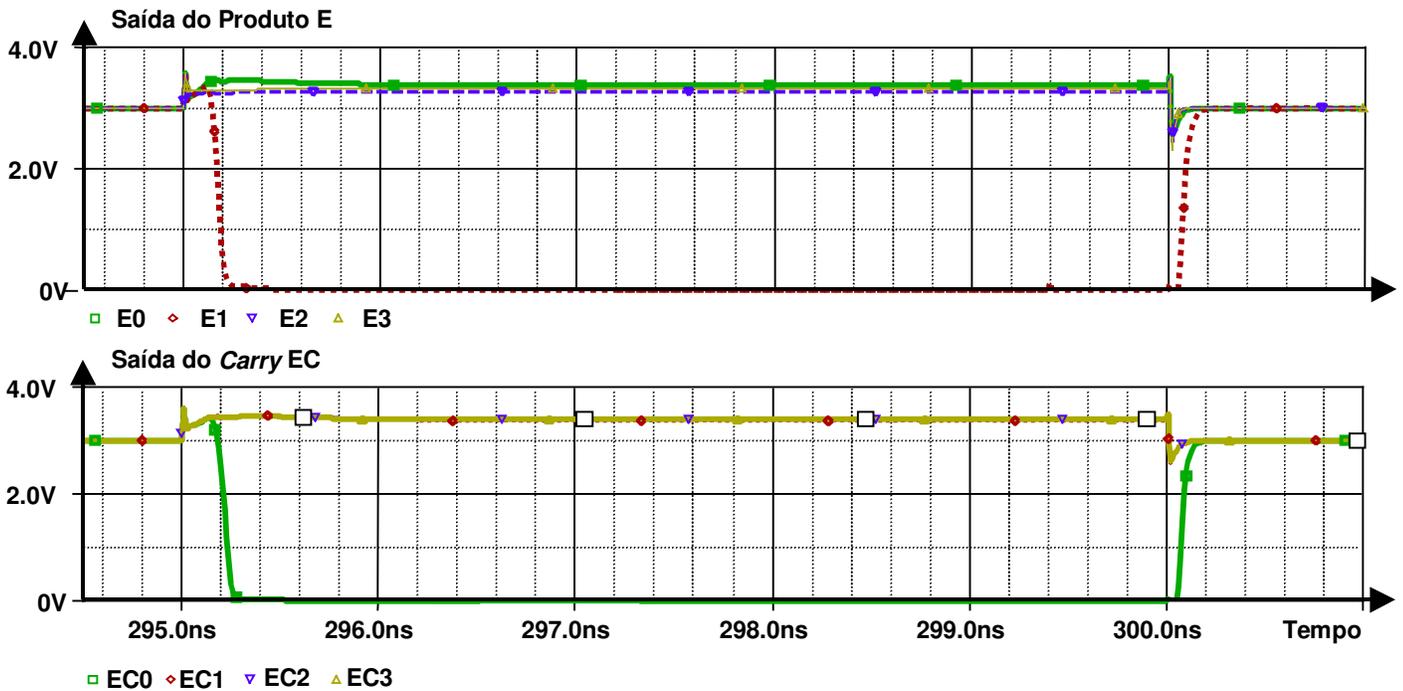


Figura II.17 Curvas do Circuito da Figura II.15. A 1ª curva mostra o resultado da multiplicação e a 2ª o *Carry* de saída ampliados onde se verifica o retardo de comutação de $< 0,3\text{ns}$.

Da observação da curva da Figura II.17 se verifica um atraso na resposta de menos de 0,3 nanosegundos, do que podemos deduzir que o atraso do multiplicador completo como mostrado no diagrama em bloco da Figura II.9 e considerando os registradores de entrada e de saída, poderá ser de aproximadamente 2 nanosegundos, para os transistores utilizados neste texto.

Como conclusão podemos dizer que com este método (Registrador/Matriz/Codificador) partindo de uma Tabela Verdade sempre será fácil implementar qualquer função.

Glossário

QUATERNÁRIO: Refere-se à Lógica Matemática onde existem QUATRO caracteres numéricos ou seja somente se tem os números '0', '1', '2' e '3'.

TERNÁRIO: Refere-se à Lógica Matemática onde existem TRÊS caracteres numéricos ou seja somente se tem os números '0', '1' e '2'.

BINÁRIO: Refere-se à Lógica Matemática onde existem DOIS caracteres numéricos ou seja somente se tem os números '0' e '1'.

DECODIFICADO: Refere-se a uma Lógica onde existe somente UM caracter numéricos considerado VERDADEIRO ou seja somente se tem o número '0'. Isto pode ser considerado como BINÁRIO onde alguns valores são proibitivos ou não existem.

BIT: Dígitos binário, um único 0 ou 1, ativado ou desativado, armazenado no seu computador. Quatro bits formam um nibble (termo raramente usado), e 8 bits formam um byte, o equivalente a um único caractere. As CPUs possuem 8, 16 ou 32 bits. Isso se refere à quantidade de informações que podem processar de cada vez.

BYTE: *Binary digit Eight*, Oito dígitos binários (em uma unidade) É um grupo de oito dígitos binários que são trabalhados e armazenados como uma unidade. Normalmente são representados por um único caractere (uma letra, número ou mesmo um símbolo)

CHIP: Pastilha de silício que pode ter a função de 7.000.000 de transistores Os chips são usados na composição das memórias de computadores.

CMOS: Tecnologia empregada na construção de alguns tipos de circuitos integrados, o termo CMOS também é utilizado para designar um tipo de memória que armazena informações de configuração que perduram, através de bateria, mesmo se o computador for desligado, como relógio e calendário, dados do disco rígido ou senha de acesso ao computador.

ALGORITMO : Uma seqüência de passos ou cálculos que orientam um dispositivo na realização de uma determinada tarefa.

IEEE: *Institute of Electrical and Electronic Engineers*. Instituto de Engenheiros Elétricos e Eletrônicos, Entidade sediada nos EUA e que normaliza especificações na área de elétrica e eletrônica. Fundada em 1884, possui hoje mais de 300.000 membros espalhados em 147 países.

MATRIZ: Sistema ou método eletrônico que faz a disposição de pontos organizados em linhas e colunas equidistantes umas das outras. Método usado em software de planilhas eletrônicas para expor os dados organizados em linhas e colunas.

Bibliografia

Referências

- [01]- Paolo Antognetti, Giuseppe Massobrio, Semiconductor Device Modeling with PSpice, McGraw-Hill Book Company 1st printing 1988.
- [02]- Yacoub, M. N. R. D. Proposta de implementação de uma lógica ternária em tecnologia CMOS. 2000.
- [03]- Dao, T. T., J. K. Russel, D. R. Preedy, E. J. McCluskev, “*Multilevel $I^2 L$ Threshold Gates*” Dig.IEEE Int. Solid-State Circuits Conf., Feb. 1977, pp. 110-111.
- [04]- Post E. L., “*Introduction to a General Theory of Elementary Propositions*” – Amer. J. of Mathematics, Vol. 43, 1921 p. 163-185.
- [05]- Lukasiewicz, J., -“*O Logice Trójwartościowej*”, Ruch Filozoficzny, Vol. 5, 1920, pp. 169-171 English Translation – “*On three valued logic*” in L. Borkowski (ed) , Selected Works, North-Holland, Amsterdam.
- [06]- Rosebloom, P. C., “*Post Algebra’s I. Postulates and General Theory*” – Amer. J. of Math. Vol.64, 1942, pp.167-188.
- [07]- Serran, N.V. Circuitos digitais ternários baseados na álgebra de Post, tese de doutorado. Unicamp, 1996.
- [08]- Haidar, A. M., F. Q. Li, M. Morisue, “*Design of Josephson Ternary Delta-Gate*” – IEICE Transactions of Information and Systems, E76D: (8) Aug. 1993, pp. 853-862.
- [09]- Morisue, M., Endo, T. Morooka, Y. Kogure, A. Kanasugi, “*A novel Josephson Ternary Multiplier*” – IEEE Translation on Applied Superconductivity, 7: (2) Jun. 1997 (Part 3), pp. 2979-2982.

- [10]- Huber, J. L., J. Chen, J. A. McCormack, C. W. Zhou, M. A. Reed, “*An RTD Transistor Switching block and its possible application in binary and ternary adders*” – IEEE Transaction on Electron Devices, Dec. 1997, 44: (12), pp. 2149-2153.]
- [11]- Current, K. W., “*Current-mode CMOS multiple-valued logic circuits*” – IEEE Journal of Solid-State Circuits, 1994,SSC-29, (2), pp. 95-107.
- [12]- Shousha, A., “*Switched-Current CMOS Ternary Logic Circuits*” Int. Journal of Electronics 79: (5) Nov.1995, pp.617-625.
- [13]- Wu, C., H. Huang, “*Design and Application of pipelined dynamic CMOS Ternary differential Logic*” – IEEE Journal of Solid-State Circuits, 1993, SSC-28, (8), pp. 895-906.
- [14]- Herrfeld, A., S. Hentschke, “*CMOS ternary dynamic differential logics*” Electronic Letter, 1994, 30: (10), pp. 762-763.
- [15]- Toto, F., R. Saletti, “*CMOS dynamic ternary circuit with full logic swing and zero- static power consumption*” Electronic Letter, 34: (11) May. 28th 1998. pp. 1083-1084.
- [16]- Yannis Tsividis, Operation and Modeling of the MOS Transistor, WCB/McGraw-Hill 1999.
- [17]- Kameyama, M. ; Kawahito, S. ;Higuchi, T. ; Computer, Volume:23, Issue: 4 April 1988, pg.43-56.
- [18]- Chu, W. S.; Current, W. “*Current-Mode CMOS quaternary multiplier circuit*” Electronics Letters 16th February 1995 Vol. 31 No. 4.
- [19]- Clarke, C. T.; Nudd, G. R.; Summerfield, S., “*Current Mode Techniques for Multiple Valued Arithmetic and Logic*” Department of Computer Science, Department of Engineering, University of Warwick, Coventry, UK, pp.279-282.
- [20]- Kawahito, S.; Kameyama, M.; Higushi, T.; Yamada, H., “*A 32x32-bit Multiplier Using Multiple-Value MOS Current-Mode Circuit*” Solid-State Circuit, IEEE Journal of, Vol. 23, issue 1st. February 1988, pp. 124-132.

- [21]- Ho, H. L.; Smith, K. C., “*Integrator-Chain Multiplier*”, Multiple-Valued Logic, 1990, Proceeding of the Twentieth International Symposium on, 23-25 May 1990, pp. 363-369.
- [22]- Kawahito, S.; Kameyama, M.; Higushi, T.; Yamada, H., “*Multiple-Valued Radix-2 Signed-digit Arithmetic Circuits for High-Performance VLSI systems.*” Solid-State Circuit, IEEE Journal of, Vol. 25, issue 1st. February 1990, pp. 125-131.
- [23]- Ishizuka, O.; Ohta, A.; Tanno, K.; Tang, Z.; Handoko, D., “*VLSI design of a quaternary multiplier with direct generation of partial products*”, Multiple-Valued Logic, 1997, Proceeding, 1997 27th International Symposium on, 28-30 May 1997, pp. 169-174.
- [24]- Shimabukuro, K.; Zukeran, C., “*Reconfigurable Current-Mode Multiple-Valued Residue Arithmetic Circuits*” IEEE Journal of Solid-State Circuits, 1998, pp. 282-287.
- [25]- Gamal A. E., “*A CMOS 32b Wallace Tree Multiplier-Accumulator*” Dig. IEEE Int'l Solid-State Circuit Conf., THPM 15.5, 1986.