

PROPOSTA DE IMPLEMENTAÇÃO DE UMA LÓGICA TERNÁRIA EM TECNOLOGIA CMOS

Maria Nídia Ramos Daoud Yacoub

Tese de Doutorado

Orientador: Prof. Dr. José Antonio Siqueira Dias

Banca Examinadora:

Prof. Dr. Laércio Caldeira – EFEI/ITAJUBÁ
Prof. Dr. Nivaldo Vicençotto Serran – FASTEC/UNISAL
Prof. Dr. Elnatan Chagas Ferreira – FEEC/UNICAMP
Prof. Dr. Yuzo Iano – FEEC/UNICAMP
Prof. Dr. Alberto Martins Jorge – FEEC/UNICAMP

Este exemplar corresponde a redação final da tese
defendida por MARIA NÍDIA RAMOS D.
YACOUB e aprovada pela Comissão
Julgada em 15 / 09 / 2000
[Assinatura]
Orientador

Campinas, 15 de Setembro de 2000

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE



200.037127

UNIDADE BC
N.º CHAMADA:
T/UNICAMP
Y1p
V. EX.
TOMBO BC/ 44345
PROC. 16-392/01
C D
PREC. R\$ 11,00
DATA 10/05/01
N.º OPD

CM-00155154-B

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

Y1p

Yacoub, Maria Nídia Ramos Daoud

Proposta de implementação de uma lógica ternária em tecnologia CMOS.--Campinas, SP: [s.n.], 2000.

Orientador: José Antonio Siqueira Dias.

Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Lógica a múltiplos valores. 2. Sistema ternário. I. Dias, José Antônio Siqueira. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III Título.

RESUMO

Neste trabalho é apresentada uma metodologia de projeto de circuitos ternários para implementação em tecnologia CMOS.

O circuito é inicialmente descrito em termos de três variáveis lógicas, em forma de uma tabela contendo as entradas e saídas. As expressões lógicas para cada saída são obtidas através da simplificação da tabela de entrada por um método gráfico similar ao Mapa de Karnaugh usado em circuitos binários convencionais.

O projeto dos circuitos comparadores de níveis lógicos, assim como todas as possíveis funções lógicas para uma única entrada, são apresentados juntamente com o resultado de suas simulações SPICE. Projetos de circuitos com memória, incluindo registradores sensíveis à borda e registradores sensíveis ao nível do relógio, assim como um contador ternário assíncrono de três estágios são também descritos.

Por último, é apresentado um circuito integrado, projetado e confeccionado em tecnologia CMOS que implementa várias das funções ternárias descritas neste trabalho.

ABSTRACT

In this work we present a methodology for the design of ternary logic circuits to be implemented in a standard CMOS technology.

The circuit is initially described in a table form containing the inputs and outputs. The logic expressions for each output are obtained through a graphical simplification, similar to the Karnaugh Map, used for conventional binary circuits.

The design of all the logic level comparators circuits, as well as the design of all logic functions with one input, are presented together with the results of their Spice simulation.

Finally, we present the design of an integrated circuit manufactured in CMOS technology which includes several ternary functions described in this work.

AGRADECIMENTOS

Agradeço a Deus.

Agradeço a todos que de uma ou outra forma contribuíram para o desenvolvimento deste trabalho, e em particular:

- ao meu orientador Prof. Dr. José Antonio Siqueira Dias, pela sua constante solicitude, apoio e incentivo,*
- ao Prof. Dr. Aberto Martins Jorge pelas valiosas sugestões,*
- a todos os professores do Departamento de Microeletrônica, por sua constante disponibilidade,*
- ao Edson Ramalho e ao Alcino e a todos os demais colegas que muito contribuíram para a execução deste projeto,*
- ao CTI e, em particular, ao Dr. Saulo Finco pela prestimosa ajuda na confecção do circuito integrado,*
- ao CNPQ pela ajuda financeira no decorrer deste trabalho,*
- aos meus pais, Benedicto e Ilka, à Sra. Helena Sallum, à Zilma de Oliveira e ao meu esposo, Michel, e aos meus queridos filhos: Alexandre, Helena, Carolina, Ricardo, Vinicius e Elisa por todo o incentivo e inestimável apoio durante este inesquecível período de tempo.*

Ao meu esposo Michel

ÍNDICE

CAPÍTULO 1 – INTRODUÇÃO	1-1
CAPÍTULO 2 - DESCRIÇÃO E SIMPLIFICAÇÃO DE FUNÇÕES TERNÁRIAS	2-1
2.1 – Descrição das Funções Ternárias	2-1
2.2 – Simplificação de Funções Ternárias	2-2
2.2.1 – Funções Ternárias de 2 Entradas	2-3
2.2.2 – Funções Ternárias de 3 Entradas	2-8
2.2.3 – Funções Ternárias de 4 Entradas	2-11
2.2.4 – Funções Ternárias com 5 Entradas ou Mais	2-15
2.3 – Conclusões	2-17
CAPÍTULO 3 – PROJETO DAS FUNÇÕES TERNÁRIAS	3-1
3.1 – Considerações Gerais e Modelagem dos Transistores	3-1
3.2 – Projeto das Funções Ternárias de Uma Entrada	3-5
3.2.1 – Funções: F(100) e F(011)	3-7
3.2.2 – Funções: F(221) e F(112)	3-7
3.2.3 – Funções: F(220), F(002), F(001) e F(110)	3-8
3.2.4 – Funções: F(200), F(022), F(122) e F(211)	3-10
3.2.5 – Funções: F(010) e F(101)	3-12
3.2.6 – Funções: F(121) e F(212)	3-13
3.2.7 – Funções: F(020) e F (202)	3-15
3.2.8 – Função F(120)	3-16
3.2.9 – Função F(201)	3-18
3.2.10-Função F(021)	3-19
3.2.11-Função F(102)	3-19
3.2.12-Função F(210)	3-20
3.3.13-Função F(012)	3-21
3.3 – Resultados das Simulações Spice das Funções	3-22
3.4 – Funções Ternárias de Duas ou Mais Entradas	3-55
3.5 – Conclusões	3-59

CAPÍTULO 4 – PROJETO DE CIRCUITOS SEQUENCIAIS	4-1
4.1 – Registradores Não-Inversores	4-1
4.1.1 – Registrador Sensível a Nível (“Latch”)	4-1
4.1.2 – Registrador Sensível à Borda do Relógio (“Flip-flop” mestre-escravo)	4-3
4.2 – Registradores Inversores	4-5
4.2.1 – Registradores Sensíveis ao Nível, FF(120) e FF(201)	4-5
4.2.2 – Registradores Sensíveis à Borda do Relógio (“Flip-flop” mestre-escravo)	4-7
4.3 – Contadores Ternários	4-8
4.4 – Resultados das Simulações Spice	4-10
4.5 – Conclusões	4-21
CAPÍTULO 5 – PROJETO DE UM CIRCUITO INTEGRADO TERNÁRIO	5-1
5.1 – Descrição do Circuito Integrado	5-1
5.2 – Funções Implementadas e “Floor-Planning”	5-3
5.3 – “Buffer” Ternário do “Pad” de Saída	5-7
5.4 – “Layout” do CI	5-8
5.5 – Conclusões	5-13
CAPÍTULO 6 – CONCLUSÕES	6-1
REFERÊNCIAS	R-1

CAPÍTULO 1

INTRODUÇÃO

O uso da lógica binária em circuitos eletrônicos alcançou um alto grau de desenvolvimento no século XX. Ferramentas matemáticas e de “software” permitem a simulação dos projetos e, em muitos casos, até mesmo a sua implementação automática. Paralelamente, os processos tecnológicos também têm sido aperfeiçoados e transistores de dimensões cada vez menores podem ser usados. Todo este avanço tem propiciado a fabricação de circuitos integrados (CI) com alto grau de complexidade.

Um problema de difícil solução em projetos complexos é o grande número de conexões internas dos CIs e a área necessária para a confecção dos “pads” para as ligações externas. Dao [7] afirma que 70% da área do “chip” é utilizada em interconexões, 20% na isolação e apenas 10% nos blocos lógicos.

Uma das alternativas que tem sido apresentada para a solução deste problema é o uso das lógicas multi-valores. Estas lógicas permitem que mais informação seja transmitida em cada interconexão, diminuindo o número de linhas necessárias para os barramentos internos, assim como o número de “pads”. O uso destas lógicas pode resultar em ganhos substanciais em área de silício.

Desde 1920, com a apresentação dos trabalhos de Post [1] e Lukasiewicz [2] e com a formalização do conceito de *Álgebra de Post de ordem n*, feito em 1942 por Rosebloom [3], as lógicas multivalores (multi-valued logic) têm recebido muita atenção, não só dos matemáticos, como também dos engenheiros, que vislumbraram a possibilidade de aplicar os conceitos nelas envolvidos na Ciência da Computação.

Embora a literatura esteja repleta de publicações na área, e grande quantidade dos trabalhos é dirigida à implementação de Circuitos Integrados, somente nos últimos 20 anos é que as

realizações práticas foram mais bem sucedidas, devido ao avanço fenomenal dado pela microeletrônica. Como as pesquisas na área estão ainda numa fase muito embrionária, as mais variadas técnicas têm sido empregadas, sem ainda existir qualquer tendência de definição do que deve prevalecer como técnica comercial para o futuro [4-17]. Para um histórico sobre a evolução dos trabalhos na área sugerimos a leitura do excelente trabalho de Serran [26].

Dentre as várias técnicas desenvolvidas temos desde circuitos usando junções Josephson [18-19] e diodos túneis [20], até circuitos que empregam técnicas mais convencionais, como circuitos em modo de corrente [21-22], e CMOS dinâmico [23-25].

Uma proposta de lógica ternária, baseada na utilização da negação cíclica de Post, o operador AND e um conjunto de novos operadores lógicos, foi recentemente apresentada por Serran [26] e teve uma proposta de implementação baseada em circuitos em modo de corrente, de alta velocidade, onde atrasos da ordem de 300 ps são previstos nos resultados obtidos por simulações dos circuitos em computador [27].

Neste trabalho apresentamos uma alternativa inovadora para a implementação de uma lógica ternária usando um novo conceito para a minimização das funções lógicas, que se mostra prático e muito semelhante ao utilizado convencionalmente em circuitos digitais binários. Diferentemente do trabalho de Serran [27], não são definidos operadores lógicos. As funções ternárias são descritas através das operações entre conjuntos (*interseção* e *união*) já usadas em circuitos binários convencionais.

Todas as funções necessárias para que a lógica seja funcionalmente completa foram projetadas e simuladas em computador. Como pretendíamos obter uma técnica de projeto compatível com a grande maioria dos circuitos digitais atuais, utilizamos a tecnologia CMOS que, além de apresentar circuitos de baixo consumo, é a mais empregada em circuitos VLSI.

Finalmente, um circuito integrado (CI) que contém todos os blocos necessários a uma avaliação preliminar desta nova técnica foi projetado e implementado em uma tecnologia CMOS 0,8 μ m. O circuito integrado contém, além dos blocos básicos, circuitos seqüenciais,

como “latches”, “flip-flops” sensíveis à borda, contadores e “buffers”, visando dar um passo adicional ao futuro da implementação desta técnica.

O trabalho está descrito em cinco capítulos. Após o Capítulo 1 de introdução, é apresentada, no Capítulo 2, a lógica ternária proposta e o método gráfico de simplificação de funções lógicas com várias entradas. O projeto elétrico de cada função de uma única entrada e os resultados das simulações elétricas dos circuitos são apresentados no Capítulo 3, juntamente com o projeto de funções de duas ou mais entradas. Os circuitos com memória, “latches”, “flip-flops” e um contador ternário de três estágios são mostrados no Capítulo 4, assim como os resultados das simulações. No Capítulo 5 é descrito o projeto de um circuito integrado onde são implementadas várias funções ternárias, “flip-flops”, contador ternário e uma função ternária de duas entradas. Os resultados das simulações por computador, validando o projeto, são discutidas no Capítulo 6, onde apresentamos as conclusões finais do nosso trabalho.

CAPÍTULO 2

DESCRIÇÃO E SIMPLIFICAÇÃO DE FUNÇÕES TERNÁRIAS

Resumo: Neste Capítulo apresentamos a descrição e notação das funções ternárias, assim como a representação gráfica usada na simplificação das funções. Após a descrição das funções (seção 2.1), apresentamos o método de simplificação das funções. Usamos inicialmente um exemplo para apenas duas variáveis de entrada (Seção 2.2.1) e, a seguir, um exemplo para três variáveis de entrada (Seção 2.2.2) e para 4 variáveis de entrada (Seção 2.2.3). O método de simplificação é então expandido para um número qualquer de variáveis de entrada.

2.1 - DESCRIÇÃO DAS FUNÇÕES TERNÁRIAS:

Os três níveis lógicos são representados pelos algarismos “0”, “1” e “2”. Na Fig. 2.1(a) temos um exemplo de uma função ternária de uma única entrada, na Fig. 2.1(b), uma função ternária de duas entradas, e na Fig. 2.1(c), uma função ternária de três entradas.

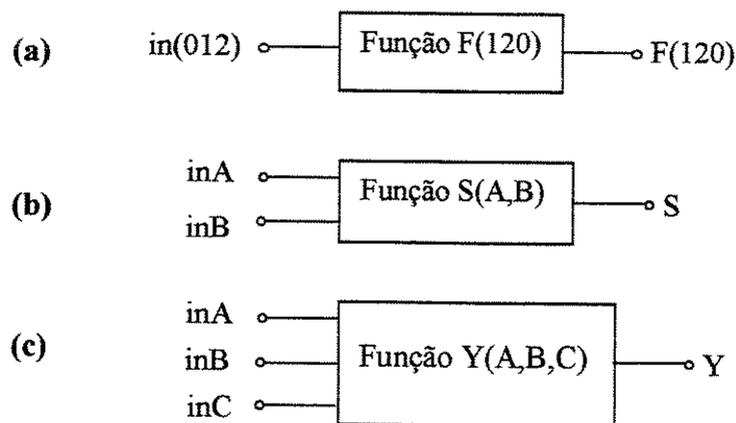


Figura 2.1: Notação de funções ternárias: (a) exemplo de função de uma única entrada; (b) exemplo de função de duas entradas; (c) exemplo de função de três entradas.

No caso de funções ternárias com uma única entrada, os níveis lógicos relativos às entradas “0”, “1” e “2”, respectivamente, são incluídos no sinal de entrada e de saída, facilitando a visualização da função. Como exemplo temos a função $F(120)$, apresentada na figura 2.1(a). Esta função é conhecida como inversor cíclico horário [27]. Dada uma entrada $in(012)$, a saída $out(120)$, será uma função da entrada tal que:

- para $in = “0” \Rightarrow F(120) = “1”$;
- para $in = “1” \Rightarrow F(120) = “2”$;
- para $in = “2” \Rightarrow F(120) = “0”$.

No caso de funções de duas ou mais entradas, que denominamos de funções compostas, não é possível uma descrição tão simplificada da função e torna-se necessário o uso de uma tabela da saída em função das entradas. Esta tabela é então representada graficamente através de um diagrama semelhante ao Mapa de Karnaugh. Este diagrama possibilita a descrição simplificada das expressões lógicas que definirão a função ternária representada.

A partir das expressões lógicas (basicamente operações “E” e “OU”), as funções podem ser projetadas através de chaves, devidamente interconectadas em série ou paralelo. Estas chaves serão implementadas com transistores tipo P ou tipo N, conforme será descrito no Capítulo 3. O controle das chaves é feito por funções simples, de uma única entrada, que detetam o nível lógico de uma determinada entrada, chamadas de comparadores lógicos. Os projetos de todas as possíveis funções de uma única entrada são também descritos no Capítulo 3.

2.2 - SIMPLIFICAÇÃO DE FUNÇÕES TERNÁRIAS:

As funções ternárias são definidas e simplificadas através de um diagrama ternário similar ao Mapa de Karnaugh usado no projeto de circuitos lógicos binários. As expressões são escritas usando comparadores de níveis lógicos, que são interligados adequadamente, de acordo com a lógica (E / OU).

Por simplicidade, começaremos pelas funções de duas entradas, e a seguir veremos as funções de três entradas, e, finalmente, funções de quatro entradas. O método é facilmente generalizado para qualquer número de entradas. Na prática, contudo, funções com mais de três entradas não são muito comuns pelo grande número de possibilidades de estados oferecidos pela lógica ternária.

2.2.1 – FUNÇÕES TERNÁRIAS DE DUAS ENTRADAS

Para facilitar a explicação, utilizaremos um exemplo. Suponha que uma função ternária “S” seja definida conforme a Tabela 2.1:

Estado	Entradas		Saída
	B	A	S
0	0	0	0
1	0	1	1
2	0	2	1
3	1	0	0
4	1	1	1
5	1	2	1
6	2	0	2
7	2	1	2
8	2	2	0

Tabela 2.1: Exemplo de uma função ternária de duas entradas, função S.

A representação gráfica desta função é feita em um diagrama (Fig. 2.2) similar ao Mapa de Karnaugh para circuitos binários. Neste diagrama são chamados de estados vizinhos os estados que diferem entre si em apenas uma das variáveis de entrada. Como exemplos, estados 0 (A=0 e B=0) e estado 6 (A=2 e B=0), assim como os estados 0 (A=0 e B=0) e 1 (A=0 e B=1). Os estados na diagonal diferem entre si em mais de uma variável de entrada e portanto não são considerados vizinhos. Quando os estados vizinhos possuem o mesmo valor lógico de

saída, eles podem ser agrupados em um único conjunto para a sua descrição em termos de comparadores. De modo análogo à simplificação de circuitos binários, estes conjuntos devem ser sempre retangulares (nunca em “L”) e devem englobar o maior número possível de elementos. Um mesmo elemento pode pertencer a mais de um conjunto [28, 29].

A\B	0	1	2
0	0	0	2
1	1	1	2
2	1	1	0

Figura 2.2: Representação gráfica da função S.

A função S pode ser escrita através dos seguintes comparadores de níveis lógicos:

$S=0$ para $(A=0 \text{ e } B \neq 2)$ ou $(A=2 \text{ e } B=2)$,

$S=1$ para $(A \neq 0 \text{ e } B \neq 2)$,

$S=2$ para $(A \neq 2 \text{ e } B=2)$

Inicialmente escolhemos os dois estados lógicos mais simples de serem implementados; no caso, os estados “ $S=1$ ” e “ $S=2$ ” (em negrito), que podem ser descritos com um único conjunto cada. O estado “ $S=0$ ” será obtido a partir da negação dos outros estados.

Os comparadores $(A=0)$ e $(B=2)$ são chamados de comparadores de igualdade enquanto $(A \neq 0)$ e $(B \neq 2)$ são chamados de comparadores de desigualdade. Os comparadores de desigualdade podem ser obtidos a partir dos comparadores de igualdade invertidos e vice-versa.

A função S pode ser descrita em termos de três comparadores de igualdade (devidamente invertidos, quando necessário): $(A=0)$, $(A=2)$ e $(B=2)$. No caso de circuitos com duas variáveis de entrada, os comparadores de igualdade representam uma única linha ou coluna no diagrama ternário, englobando três elementos. Já os comparadores de desigualdade representam duas linhas ou colunas. As operações efetuadas com as chaves representam a

interseção (chaves em série) ou união (chaves em paralelo) destes conjuntos (Fig.2.3). Todas as propriedades básicas da teoria de conjuntos (associatividade, distributividade) podem ser aplicadas na simplificação da lógica ternária.

$A \setminus B$	0	1	2
0	0	0	2
1	1	1	2
2	1	1	0

Figura 2.3: Representação da função S em termos de operações entre conjuntos.

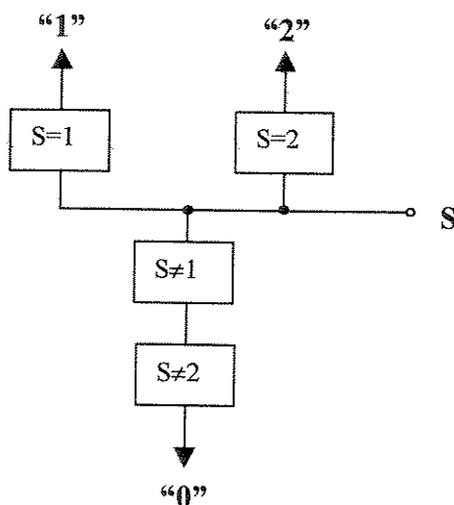


Figura 2.4: Diagrama em blocos da implementação da função S.

A implementação da função S em termos de diagrama de blocos é apresentado na Fig. 2.4 e o circuito equivalente usando-se chaves é apresentado na Fig. 2.5. O projeto interno dos comparadores usando-se transistores é feito no Capítulo 3.

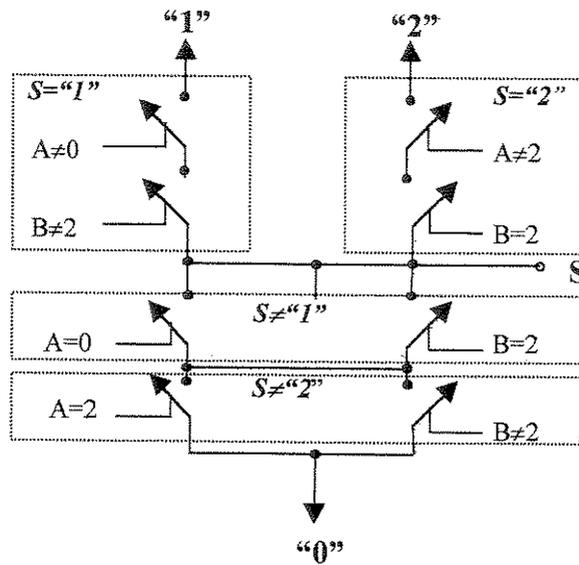


Figura 2.5: Representação interna da função S usando chaves.

Da Fig. 2.5 podemos observar que o circuito ($S \neq 1$) é obtido a partir do circuito ($S=1$) onde as chaves em série são substituídas por chaves em paralelo, pois basta que uma das condições ($A \neq 0$) ou ($B \neq 2$) ocorra para que a função S seja diferente de "1". O mesmo ocorre para ($S \neq 2$) em relação a ($S=2$).

A metodologia de simplificação de funções ternárias visa minimizar o número de comparadores necessários para a implementação final da função, assim como o número de chaves que interligará estes comparadores. Como no caso da simplificação de funções binárias pelo Mapa de Karnaugh, podem haver várias combinações de conjuntos de "0", "1" e "2" que levam a circuitos igualmente simples [28, 29].

A \ B	0	1	2
0	1	2	2
1	0	2	0
2	1	2	2

$A \neq 1 \text{ e } B = 0$ (linha 1, coluna 0)
 $A = 1 \text{ e } B \neq 1$ (linha 1, colunas 0 e 2)
 $B = 1$ (coluna 1)

Figura 2.6: Representação gráfica de uma função ternária, função Z.

Para ilustrarmos outros modos de agrupamentos dos conjuntos lógicos, usaremos outro exemplo. Considere a função ternária Z , que possui o diagrama de estados da Fig. 2.6. Ela pode ser descrita a partir dos seguintes comparadores:

$Z=0$ para $(A=1 \text{ e } B \neq 1)$,

$Z=1$ para $(A \neq 1 \text{ e } B=0)$,

$Z=2$ para $(B=1) \text{ ou } (A \neq 1 \text{ e } B \neq 0)$.

Podemos perceber que o conjunto será mais simples de se descrever se possuir número de elementos que sejam múltiplos de três, pois neste caso, bastará um único comparador de igualdade ou desigualdade. No caso da função “ Z ”, serão necessários três circuitos comparadores de igualdade, devidamente invertidos, quando necessário: $(A=1)$, $(B=1)$ e $(B=0)$. O nível lógico “2” pode ser escolhido como o padrão quando os demais níveis “0” ou “1” não ocorrerem e seriam necessárias quatro chaves para descrevê-lo (como no exemplo anterior). No entanto, como “ $Z=2$ ” utiliza os mesmos comparadores já usados para “ $Z=1$ ” e “ $Z=0$ ”, a sua própria expressão será utilizada para descrevê-lo, economizando uma chave. O circuito equivalente é apresentado na Fig. 2.7.

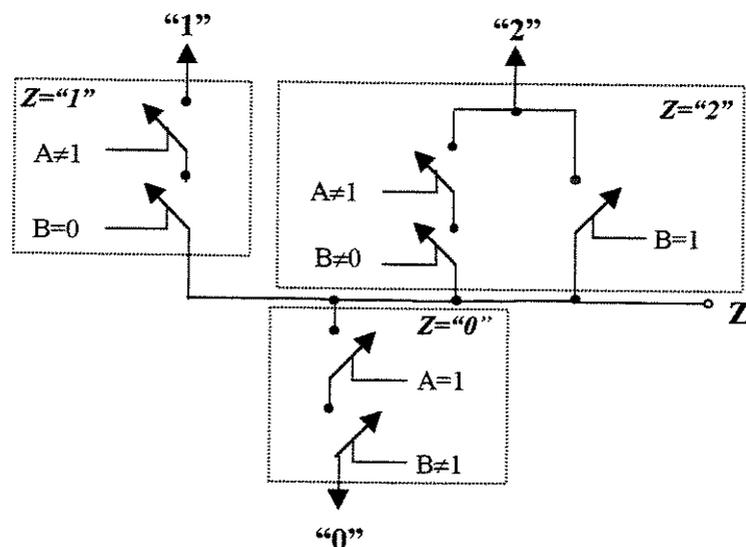


Figura 2.7: Representação interna da função “ Z ” usando chaves.

2.2.2 – FUNÇÕES TERNÁRIAS DE TRÊS ENTRADAS

Na Tabela 2.3 temos um exemplo de uma função Y com três variáveis de entrada (A, B, C), que está representada de forma gráfica na Fig. 2.8.

Entradas				Saída
Est.	C	B	A	Y
0	0	0	0	0
1	0	0	1	1
2	0	0	2	1
3	0	1	0	0
4	0	1	1	1
5	0	1	2	1
6	0	2	0	2
7	0	2	1	2
8	0	2	2	0

Entradas				Saída
Est.	C	B	A	Y
9	1	0	0	0
10	1	0	1	0
11	1	0	2	0
12	1	1	0	0
13	1	1	1	1
14	1	1	2	0
15	1	2	0	2
16	1	2	1	0
17	1	2	2	0

Entradas				Saída
Est.	C	B	A	Y
18	2	0	0	0
19	2	0	1	1
20	2	0	2	1
21	2	1	0	0
22	2	1	1	1
23	2	1	2	1
24	2	2	0	2
25	2	2	1	0
26	2	2	2	0

Tabela 2.3: Exemplo de uma função ternária de três entradas, função Y.

Podemos observar que quando acrescentamos uma variável em relação ao exemplo anterior de duas variáveis (Fig 2.2), o diagrama torna-se uma figura tridimensional com três planos, um para cada valor da variável “C”. Agora, além dos estados vizinhos dentre os elementos de um mesmo plano, os estados localizados na mesma coordenada A/B mas em planos diferentes, ainda são estados vizinhos, pois diferem entre si apenas na variável “C”. Os planos “C=0” e “C=2” também são planos vizinhos e, portanto, seus elementos podem ser agrupados juntos.

Neste exemplo, uma simples inspeção do gráfico nos mostra que as saídas “Y=1” e “Y=2” são mais fáceis de serem implementadas do que a saída “Y=0”, que será definida como o padrão, obtida a partir da negação de “Y=1” e “Y=2”.

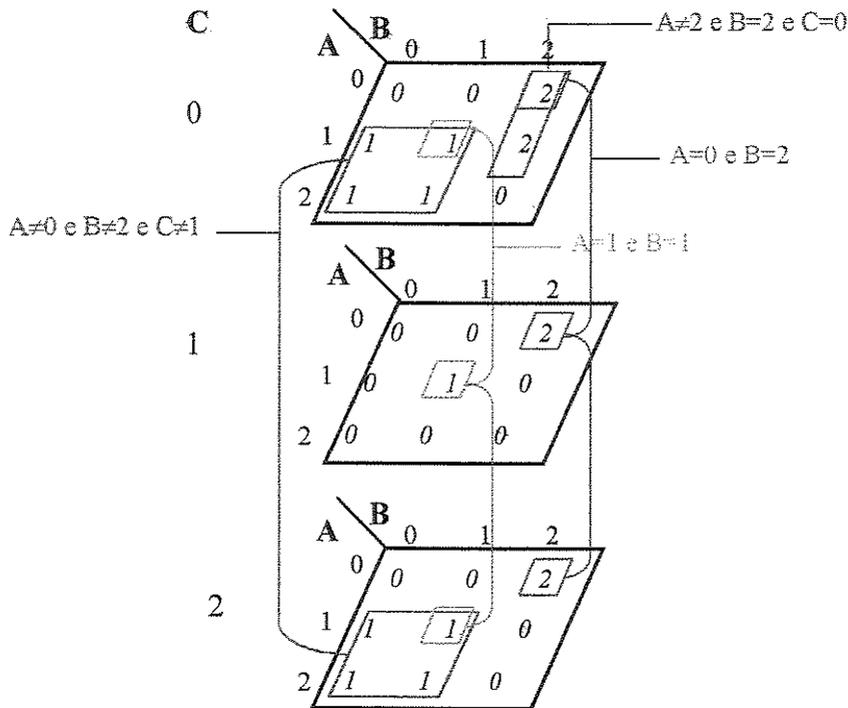


Figura 2.8: Representação gráfica de uma função ternária Y de três entradas.

Agrupando-se todos os elementos com um mesmo nível lógico, obtemos as seguintes expressões:

$Y=1$ se $(A \neq 0 \text{ e } B \neq 2 \text{ e } C \neq 1)$, ou se $(A=1 \text{ e } B=1)$.

$Y=2$ se $(A \neq 2 \text{ e } B=2 \text{ e } C=0)$, ou se $(A=0 \text{ e } B=2)$.

$Y=0$ se $(Y \neq 1 \text{ e } Y \neq 2)$ (estado padrão).

Da Fig. 2.9, podemos observar que “ $Y=0$ ” é obtido a partir do **OU** (conjunto união) das duas desigualdades: “ $Y \neq 1$ ” e “ $Y \neq 2$ ”. Estas desigualdades, por sua vez, podem ser obtidas a partir da inversão dos mesmos circuitos comparadores usados para gerar os respectivos comparadores de igualdade, e substituindo as chaves em série por chaves em paralelo, e vice-versa:

$Y=1$ se $(A \neq 0 \text{ e } B \neq 2 \text{ e } C \neq 1)$, ou \Rightarrow $Y \neq 1$ se $(A=0 \text{ ou } B=2 \text{ ou } C=1)$, e
se $(A=1 \text{ e } B=1)$. se $(A \neq 1 \text{ ou } B \neq 1)$.

$Y=2$ se $(A \neq 2 \text{ e } B=2 \text{ e } C=0)$, ou \Rightarrow $Y \neq 2$ se $(A=2 \text{ ou } B \neq 2 \text{ ou } C \neq 0)$, e
se $(A=0 \text{ e } B=2)$. se $(A \neq 0 \text{ ou } B \neq 2)$.

Os comparadores de níveis lógicos invertidos; por exemplo, $A=0$ e $A \neq 0$, podem ser obtidos pela inversão das funções detetoras de níveis lógicos, ou pelo uso de transistores de tipo diferente, N ou P, conforme o caso. Estes comparadores serão vistos em detalhe no Capítulo 3.

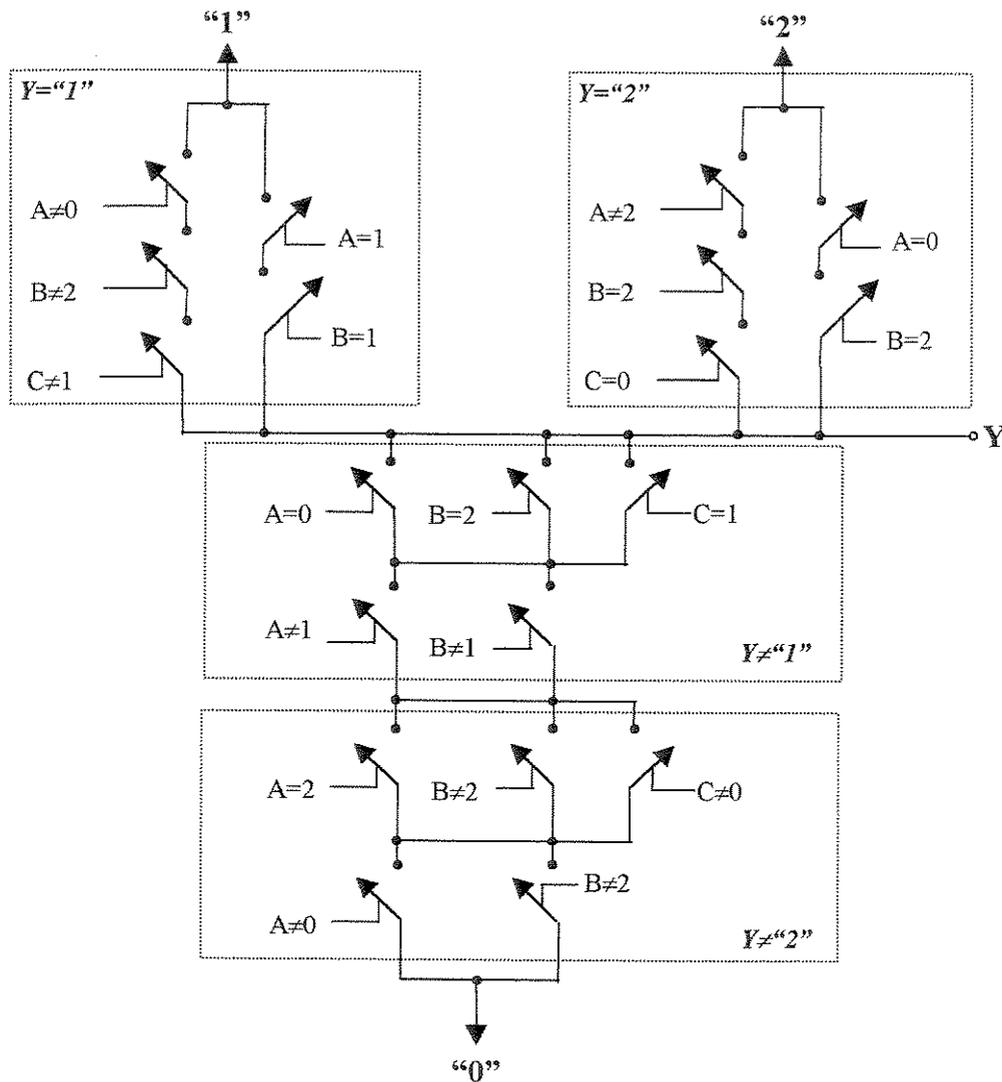


Figura 2.9: Representação com chaves da função ternária Y de três entradas.

2.2.3 – FUNÇÕES TERNÁRIAS DE QUATRO ENTRADAS

Suponhamos que uma função ternária W seja definida conforme a tabela 2.4. A representação gráfica desta função é apresentada na Fig. 2.10.

		Entradas				Saída
Est	D	C	B	A	W	
0	0	0	0	0	0	
1	0	0	0	1	1	
2	0	0	0	2	1	
3	0	0	1	0	0	
4	0	0	1	1	1	
5	0	0	1	2	1	
6	0	0	2	0	2	
7	0	0	2	1	2	
8	0	0	2	2	0	

		Entradas				Saída
Est	D	C	B	A	W	
9	0	1	0	0	0	
10	0	1	0	1	0	
11	0	1	0	2	0	
12	0	1	1	0	0	
13	0	1	1	1	1	
14	0	1	1	2	0	
15	0	1	2	0	2	
16	0	1	2	1	0	
17	0	1	2	2	0	

		Entradas				Saída
Est	D	C	B	A	W	
18	0	2	0	0	0	
19	0	2	0	1	1	
20	0	2	0	2	1	
21	0	2	1	0	0	
22	0	2	1	1	1	
23	0	2	1	2	1	
24	0	2	2	0	2	
25	0	2	2	1	0	
26	0	2	2	2	0	

		Entradas				Saída
Est	D	C	B	A	W	
27	1	0	0	0	0	
28	1	0	0	1	0	
29	1	0	0	2	0	
30	1	0	1	0	0	
31	1	0	1	1	0	
32	1	0	1	2	0	
33	1	0	2	0	2	
34	1	0	2	1	2	
35	1	0	2	2	0	

		Entradas				Saída
Est	D	C	B	A	W	
36	1	1	0	0	0	
37	1	1	0	1	0	
38	1	1	0	2	2	
39	1	1	1	0	0	
40	1	1	1	1	0	
41	1	1	1	2	2	
42	1	1	2	0	2	
43	1	1	2	1	0	
44	1	1	2	2	2	

		Entradas				Saída
Est	D	C	B	A	W	
45	1	2	0	0	0	
46	1	2	0	1	1	
47	1	2	0	2	2	
48	1	2	1	0	0	
49	1	2	1	1	0	
50	1	2	1	2	2	
51	1	2	2	0	2	
52	1	2	2	1	0	
53	1	2	2	2	2	

Tabela 2.4 (1ª parte): Exemplo de uma função ternária de quatro entradas, função W .

Entradas					Saída
Est	D	C	B	A	W
54	2	0	0	0	0
55	2	0	0	1	0
56	2	0	0	2	0
57	2	0	1	0	0
58	2	0	1	1	1
59	2	0	1	2	0
60	2	0	2	0	2
61	2	0	2	1	2
62	2	0	2	2	0

Entradas					Saída
Est	D	C	B	A	W
63	2	1	0	0	0
64	2	1	0	1	0
65	2	1	0	2	2
66	2	1	1	0	0
67	2	1	1	1	1
68	2	1	1	2	2
69	2	1	2	0	2
70	2	1	2	1	0
71	2	1	2	2	2

Entradas					Saída
Est	D	C	B	A	W
72	2	2	0	0	0
73	2	2	0	1	1
74	2	2	0	2	2
75	2	2	1	0	0
76	2	2	1	1	1
77	2	2	1	2	2
78	2	2	2	0	2
79	2	2	2	1	0
80	2	2	2	2	2

Tabela 2.4 (2ª parte): Exemplo de uma função ternária de quatro entradas, função W.

Na Fig. 2.10 temos a representação gráfica da função W de quatro entradas. Os nove planos estão dispostos de forma semelhante às células do gráfico de duas variáveis (Fig. 2.2), e tem a mesma relação de vizinhança entre si que as células da função S. Desta forma, os planos horizontais paralelos entre si são também planos vizinhos. O mesmo acontece para os planos verticais paralelos entre si. Planos em diagonais não são considerados vizinhos. Uma maneira simples de conhecermos os estados vizinhos é verificar se a diferença entre eles está em uma única variável de entrada.

Como foi dito anteriormente, podem existir várias maneiras diferentes de agruparmos os estados, que levam a expressões igualmente simples. A regra geral consiste em começarmos pelos elementos que possuem uma única opção de agrupamento, lembrando sempre que quanto maior o agrupamento, mais simples será a função.

No caso da função W é fácil percebermos que os estados “1” e “2” são os mais simples de serem implementados. No caso de funções em que é difícil diferenciar os estados mais simples, torna-se necessário descrever todos os estados e só então decidir pelo mais simples.

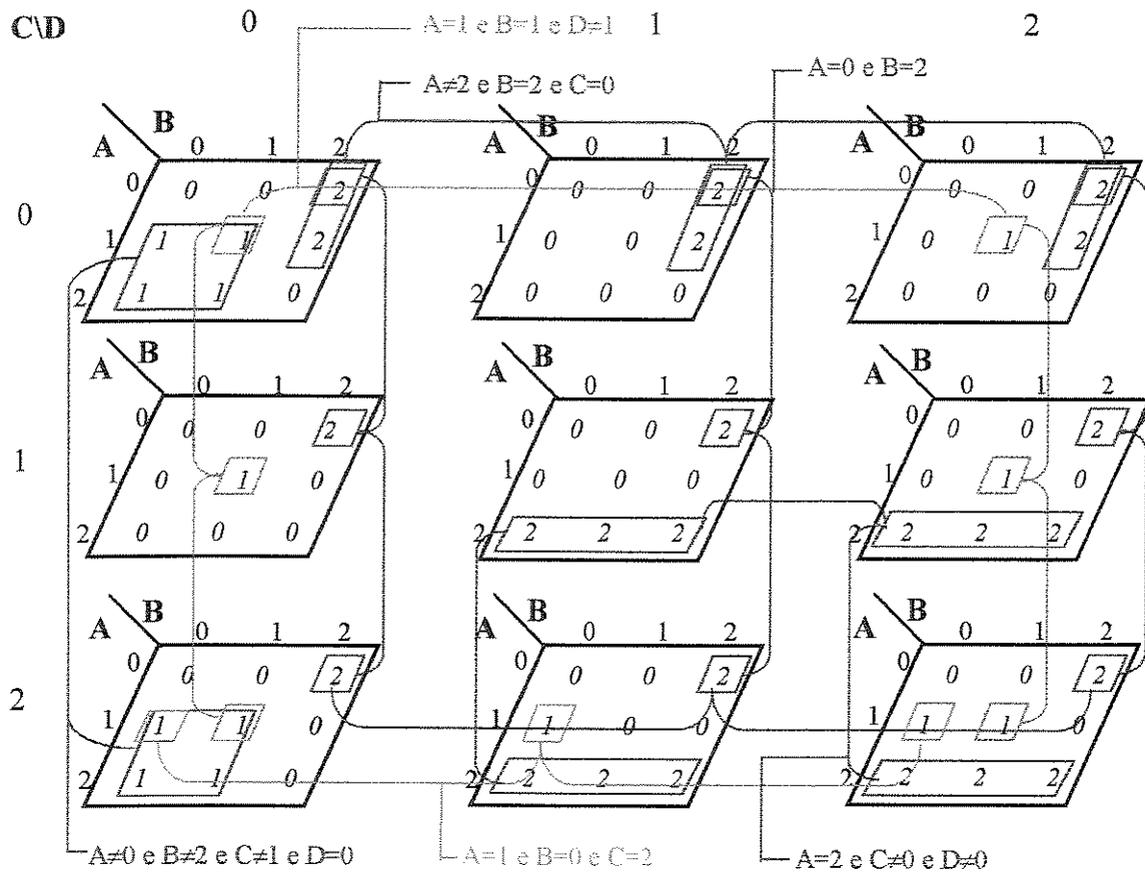


Figura 2.10: Representação gráfica de uma função de 4 variáveis, função W.

De acordo com a simplificação apresentada na Fig. 2.10, a função W pode ser descrita como:

$W = 1$ para $(A \neq 0)$ e $(B \neq 2)$ e $(C \neq 1)$ e $(D \neq 0)$, ou
para $(A=1)$ e $(B=1)$ e $(D \neq 1)$, ou
para $(A=1)$ e $(B=0)$ e $(C=2)$.

$W = 2$ para $(A=2)$ e $(C \neq 0)$ e $(D \neq 0)$, ou
para $(A \neq 2)$ e $(B=2)$ e $(C=0)$, ou
para $(A=0)$ e $(B=2)$.

$W = 0$ se $W \neq 1$ e $W \neq 2$ (estado padrão).

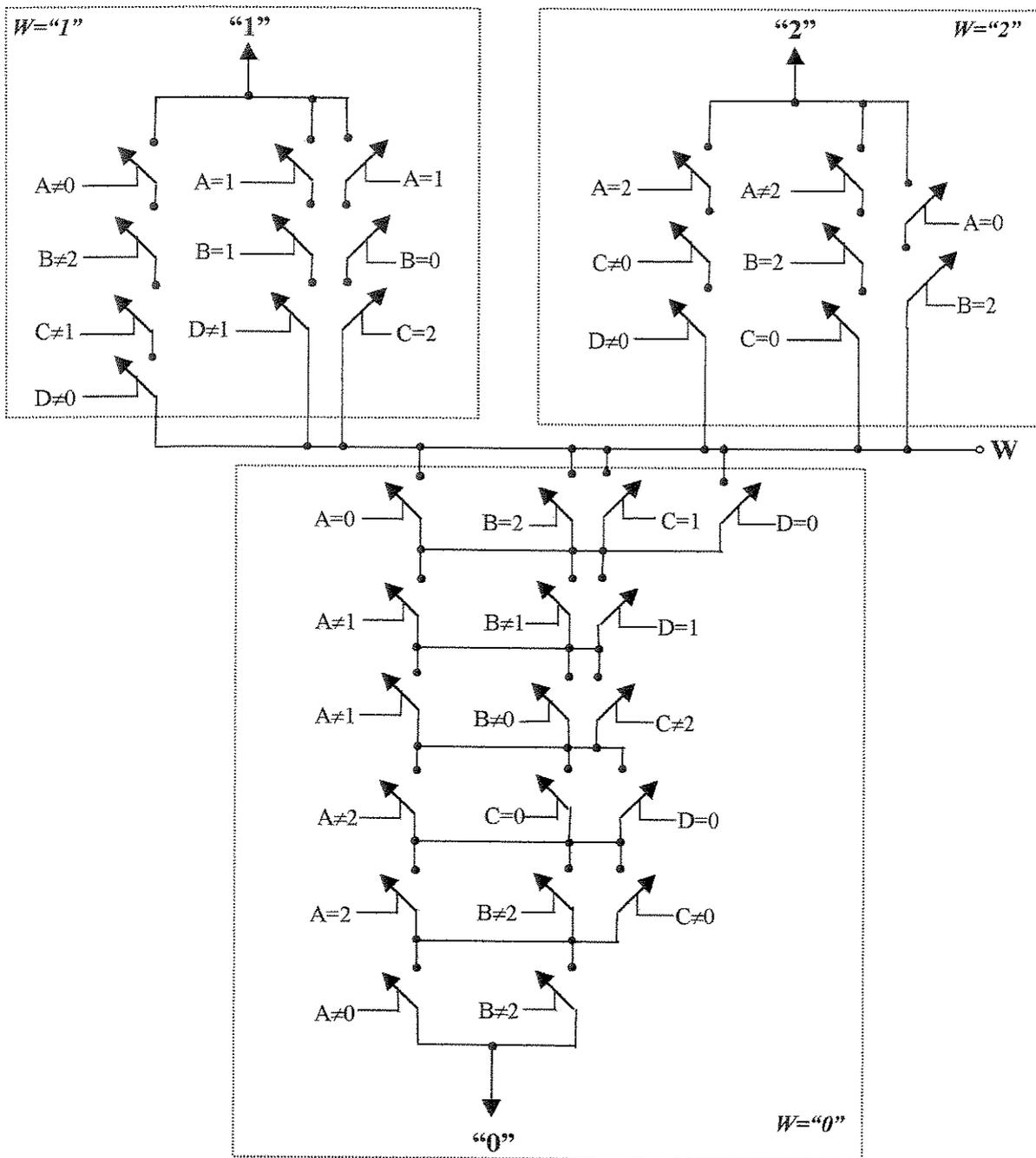


Figura 2.11: Representação em termos de chaves da função W .

Na Fig. 2.11 temos a implementação da função W usando chaves. De modo análogo aos exemplos anteriores de 2 e 3 variáveis de entrada, os estados " $W=0$ " são implementados através da negação dos estados " $W=1$ " e " $W=2$ ". Funções com 4 variáveis de entrada são mais complexas de serem implementadas e programas de CAD ("Computer Aided Design) podem ser desenvolvidos para a automatização da simplificação.

2.2.4 – FUNÇÕES TERNÁRIAS COM 5 ENTRADAS OU MAIS

Como foi dito no início do Capítulo, funções com mais de 3 variáveis de entrada não são comuns em circuitos convencionais, porém, o método de simplificação pode ser estendido para qualquer número de entradas. No caso de funções com 5 variáveis de entrada, o conjunto de 9 planos usados para o exemplo da função com 4 variáveis torna-se uma célula, e o gráfico deve conter 3 destas células (cada célula com 9 planos), como no exemplo da função para 3 variáveis. O esboço da representação gráfica para 5 variáveis encontra-se na Fig. 2.12.

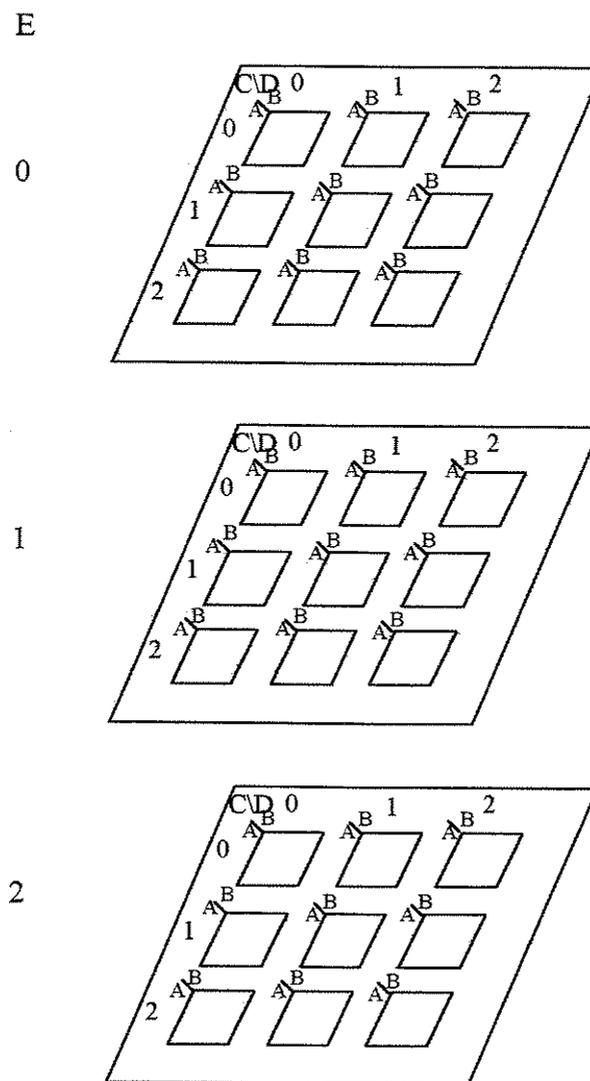


Figura 2.12: Esboço da representação gráfica para uma função ternária de 5 entradas.

No caso de funções ternária com 6 entradas, o gráfico da Fig. 2.12 é expandido horizontalmente 3 vezes para representar os 3 estados possíveis da variável F. O gráfico conterà 9 planos grandes, cada um com 9 planos pequenos (81 planos ao todo, representando 729 possibilidades de estado). A representação gráfica correspondente encontra-se esboçada na Fig. 2.13. Este exemplo é análogo ao da Fig. 2.11, onde para cada célula temos agora um plano.

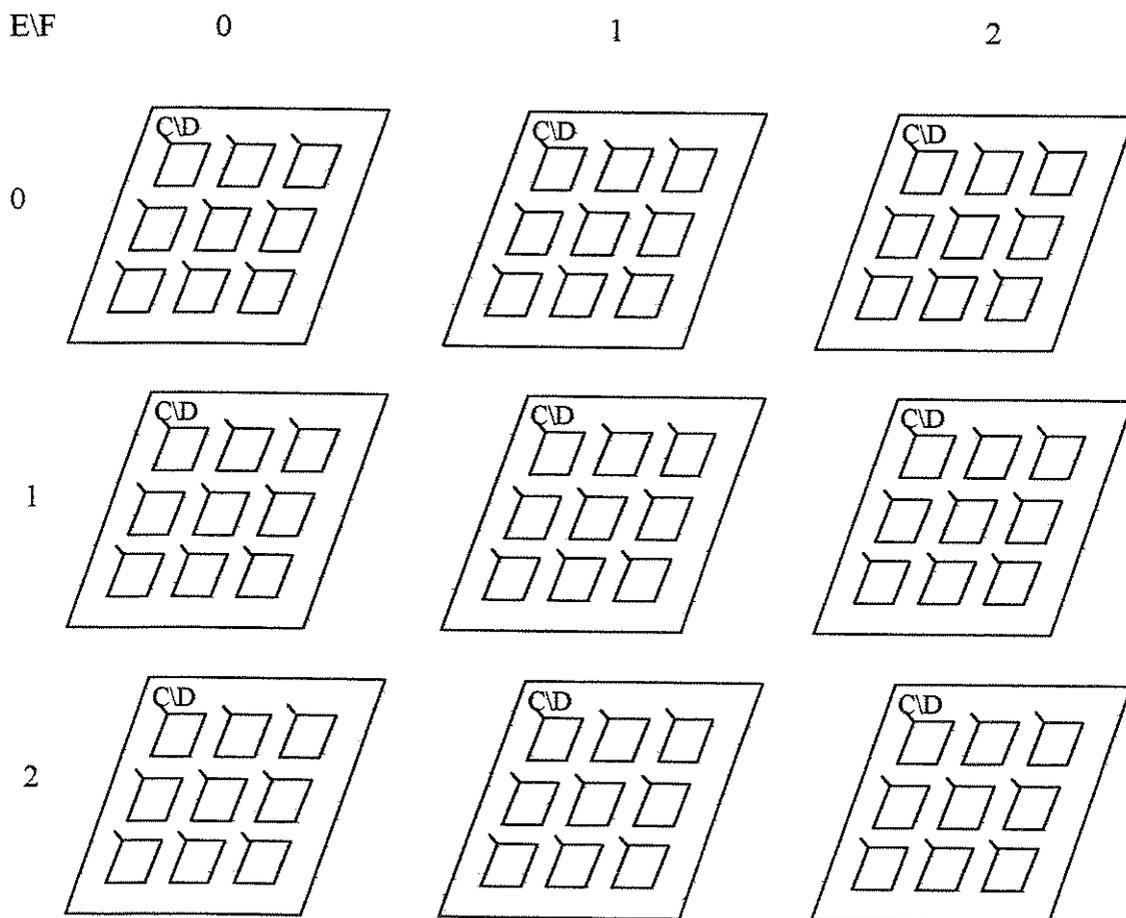


Figura 2.13: Esboço da representação gráfica para uma função ternária de 6 entradas.

Para cada variável de entrada que é acrescentada, a representação gráfica é expandida 3 vezes horizontalmente ou verticalmente a fim de representarmos todos os possíveis estados da nova variável. Quando o gráfico anterior já possui as 9 células (células que podem representar um único estado, um plano, ou um conjunto de planos), é necessário triplicarmos o gráfico

anterior verticalmente, e neste caso, o gráfico anterior é considerado como a célula básica do novo gráfico. Por exemplo, no caso de 7 variáveis de entrada o gráfico usado para as 6 variáveis de entrada (Fig. 2.13) deve ser triplicado verticalmente para acomodar os 3 possíveis estados da nova variável de entrada, "H". Em teoria não há limites para o número de variáveis de entrada de uma função.

2.3 -CONCLUSÕES

As funções ternárias são descritas usando 3 variáveis lógicas: "0", "1" e "2". Existem 27 diferentes funções ternárias de 1 única entrada, algumas delas são especialmente úteis na geração de funções de 2 ou mais entradas e são chamados de circuitos comparadores. Estas funções apresentam um nível lógico diferente apenas para um valor do nível lógico de entrada, e suas saídas variam entre os estados lógicos "0" e "2".

As funções ternárias com 2 ou mais variáveis de entrada são simplificadas antes de serem implementadas por um diagrama gráfico similar ao Mapa de Karnaugh usado para circuitos digitais binários. A representação gráfica para funções ternárias é feita de forma modular, onde cada célula possui sempre 9 sub-células e pode ser expandida para qualquer número de variáveis de entrada. O método é de fácil implementação, podendo ser facilmente automatizado por CAD ("Computer-Aided-Design").

CAPÍTULO 3

PROJETO DAS FUNÇÕES TERNÁRIAS

Resumo: No Capítulo 3 mostraremos o projeto interno de todas as funções ternárias de uma única entrada, assim como o projeto a nível de transistores de funções com múltiplas entradas. Na Seção 3.1 descreveremos o uso dos transistores como chaves e apresentaremos as principais equações a serem usadas no projeto. Na Seção 3.2 apresentaremos o projeto elétrico de cada função de uma única entrada. Os resultados da simulação SPICE das funções para as três condições: típica, máximo consumo e máximo atraso serão apresentados na Seção 3.3. O projeto de funções com duas ou mais entradas é descrito na Seção 3.4.

3.1 – CONSIDERAÇÕES GERAIS E MODELAGEM DOS TRANSISTORES

Os circuitos ternários são projetados com transistores NMOS e PMOS operando basicamente como chaves. Os transistores MOS estarão em condução quando a tensão porta-fonte V_{GS} for superior à tensão de limiar V_t [30], ou seja:

$$\text{para NMOS} \Rightarrow V_{GS} \geq V_t \quad \text{onde: } V_{GS} = V_G - V_S \text{ e } V_t > 0 \quad (\text{Equação 3.1a})$$

$$\text{para PMOS} \Rightarrow V_{GS} \leq V_t \quad \text{onde } V_{GS} = V_G - V_S \text{ e } V_t < 0 \quad (\text{Equação 3.1b})$$

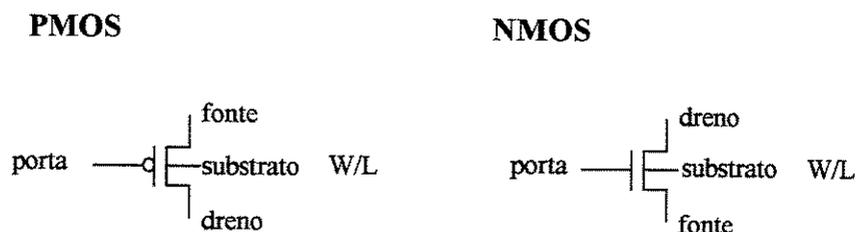


Figura 3.1: Representações gráficas dos transistores PMOS e NMOS.

Na Fig. 3.1 temos a representação gráfica dos transistores NMOS e PMOS, com suas respectivas tensões de porta, fonte e dreno. Os terminais de fonte e dreno dos transistores são, geralmente, simétricos. A diferença de posição dos terminais de fonte para os transistores PMOS em relação ao NMOS se deve à facilidade de interconexão das tensões de alimentação no esquema elétrico.

Os transistores PMOS são usados para chavear o nível lógico “2”, enquanto os transistores NMOS são usados para chavear o nível “0”. Para o nível lógico intermediário, nível “1”, podemos usar tanto o transistor PMOS quanto o NMOS. No caso do chaveamento de linhas de dados onde o nível lógico de entrada não é conhecido anteriormente, é necessário usarmos portas de transmissão. Evitamos, dessa forma, a queda de tensão V_t entre os terminais de fonte e dreno dos transistores (Equações 3.1a e 3.1b). Este uso é análogo para o caso de circuitos CMOS digitais binários convencionais [30].

Os valores de W/L indicam a largura e comprimento do canal, respectivamente. Estes valores estão diretamente relacionados com a corrente que flui através do transistor, dada pelas Equações 3.3 e 3.4. Portanto, quanto maior a relação W/L , maior a corrente no chaveamento e menor o atraso da função. Os valores de W/L indicados nos circuitos seguintes (Seção 3.2) foram determinados através da simulação SPICE para o pior caso de atraso (parâmetros obtidos da “foundry” AMS para o processo CMOS de 0.8 μ m).

Na operação de circuitos digitais CMOS, o consumo de corrente restringe-se ao momento de chaveamento dos transistores. O transistor MOS tipo enriquecimento que será usado no nosso projeto é definido como operando nas regiões “triodo” e “saturação” (Fig. 3.2) de acordo com as Equações 3.2a e 3.2b [30]:

$$\text{Região de saturação:} \quad V_{DS} \geq V_{GS} - V_t \quad (\text{Equação 3.2a})$$

$$\text{Região triodo:} \quad V_{DS} < V_{GS} - V_t \quad (\text{Equação 3.2b})$$

onde: V_{DS} é a tensão dreno-fonte e V_{GS} é a tensão porta-fonte.

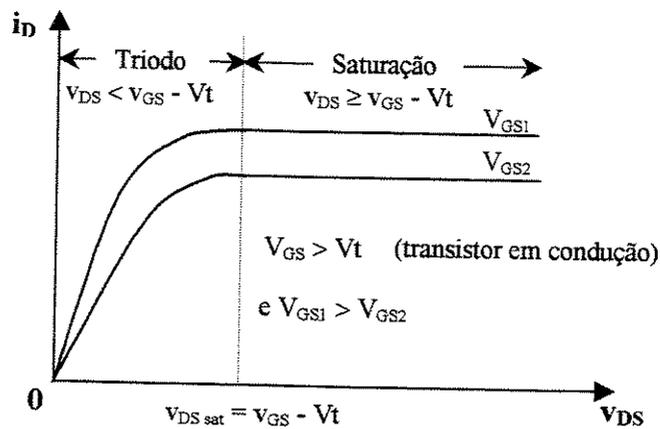


Figura 3.2: Gráfico da corrente de dreno i_D versus a tensão dreno-fonte v_{DS} .

As correntes através do transistor são dadas pelas seguintes equações:

região triodo: $i_D = K[2(v_{GS} - V_t)v_{DS} - v_{DS}^2]$ Equação 3.3a

região de saturação: $i_D = K(v_{GS} - V_t)^2 (1 + \lambda v_{DS})$ Equação 3.3b

onde $K = \frac{1}{2}\mu C_{ox}(W/L)$ Equação 3.4

e $V_t = V_{t0} + \gamma (\sqrt{v_{SB} + 2\phi_f} - \sqrt{2\phi_f})$ Equação 3.5

Na Equação 3.4 o parâmetro K é dependente dos parâmetros do processo utilizado como a mobilidade dos elétrons ou lacunas (μ_n ou μ_p , respectivamente) e a capacitância do óxido (C_{ox}). A dimensão do transistor (W/L) é o parâmetro que pode ser alterado no projeto de forma a atendermos as especificações dinâmicas do circuito.

O termo $(1 + \lambda v_{DS})$ da Equação 3.3b deve-se ao efeito da modulação de canal (diminuição do tamanho do canal com o aumento de v_{DS}) onde λ é uma constante positiva dependente do processo MOS.

A Equação 3.5 mostra a relação da tensão de limiar V_t com a tensão entre substrato e fonte, v_{SB} conhecida como efeito de corpo. A V_{t0} é a tensão de limiar para $v_{SB} = 0$, γ é uma constante para o dado processo de fabricação e $2\phi_f$ é o potencial eletrostático de equilíbrio para o

material tipo P, usado como substrato. Podemos perceber que quanto maior a diferença de tensão entre fonte e substrato, maior será V_t e conseqüentemente, menor a corrente i_D através do transistor. Este efeito é bastante significativo e, para compensá-lo, torna-se necessário o aumento da relação W/L .

Nos circuitos projetados tentamos, sempre que possível, cancelar o efeito de corpo conectando o substrato à fonte. No caso dos transistores PMOS usamos poços separados para o substrato quando necessário. No caso dos transistores NMOS isto já não é possível pois o substrato é comum para todos os transistores (processo CMOS - substrato tipo P e poço N).

Na maioria dos casos, no início do chaveamento o transistor que entrará em condução estará operando na região de saturação, pois sua tensão de dreno é igual ou maior à tensão de porta. Conforme a tensão v_{DS} for diminuindo, o transistor entrará na região linear, tendo sua corrente de dreno, i_D , diminuída. Nos casos em que a tensão de porta no momento do chaveamento é maior do que a tensão de dreno, o transistor operará sempre na região triodo.

Para um dado circuito, normalmente a relação W/L dos transistores PMOS é aproximadamente o dobro da relação (W/L) dos transistores NMOS. Esta diferença é necessária para compensar a diferença de mobilidade dos transistores PMOS (μ_p) que, normalmente é a metade da mobilidade dos transistores NMOS (μ_n). No caso de transistores em série, também torna-se necessário duplicar a relação (W/L), por causa da resultante soma das resistências dos transistores.

A alimentação dos circuitos é feita por três fontes externas: V_0 , V_1 e V_2 que definem os valores de referência a serem utilizados para os três níveis lógicos: “0”, “1” e “2”, respectivamente. O circuito foi projetado para operar com tensões de referência de “0V”, “1,5V” e “3,0V”; no entanto, valores mais altos para os níveis lógicos V_1 e V_2 levarão a circuitos mais rápidos. As medidas dos atrasos obtidos nas diversas funções em relação aos níveis de referência usados encontram-se no Capítulo 4.

3.2- PROJETO DAS FUNÇÕES TERNÁRIAS DE UMA ENTRADA

Para representarmos as funções ternárias de uma entrada usamos a notação $F(abc)$, onde “a”, “b” e “c” são os valores da saída relativos às entradas “0”, “1” e “2”, respectivamente.

A função $F(120)$, por exemplo, é conhecida como “inversor cíclico horário” [27]. Dada uma entrada $in(012)$, a saída $out(120)$, será uma função da entrada tal que:

- para $in=“0” \Rightarrow out=“1”$,
- para $in=“1” \Rightarrow out=“2”$,
- para $in=“2” \Rightarrow out=“0”$.

A função $F(201)$ é o inversor cíclico anti-horário. A importância destes inversores na implementação da lógica ficará clara no andamento deste trabalho onde, por exemplo, veremos que estas funções são usadas na formação dos “flip-flops” (circuitos registradores) ternários.

Para funções ternárias de uma única entrada, existem 27 possibilidades de variações da saída em função da entrada, conforme mostrado na Tabela 3.1.

As funções ternárias que apresentam como saída um nível lógico diferente para apenas um valor lógico de entrada são chamadas de funções detectoras de níveis lógicos; por exemplo, $F(100)$, $F(212)$ e $F(200)$. Dentre as funções detectoras, as que as saídas variam entre os níveis “0” e “2” são especialmente úteis na geração das demais funções e são chamadas de comparadores de níveis lógicos.

As funções $F(011)$, $F(100)$, $F(221)$ e $F(112)$ são as mais simples de serem implementadas, pois seus circuitos são semelhantes ao do inversor binário comum. No entanto, o uso destas funções é restrito, pois as suas saídas variam entre dois níveis próximos de tensão (“0” e “1”, ou entre “2” e “1”). Detalhes dos projetos de todas estas funções ternárias de uma entrada serão descritos a seguir.

	FUNÇÃO	COMENTÁRIOS	Nº de transistores
0	F(000)	V0	0
1	F(001)	Deteta o nível lógico "2"	6
2	F(002)	Comparador para o nível "2"	6
3	F(010)	Deteta o nível lógico "1"	12
4	F(011)	F(100) invertida	4
5	F(012)	Função "buffer" não inversora	8
6	F(020)	Comparador para o nível "1"	14
7	F(021)	Inverte os níveis "1" e "2"	14
8	F(022)	Comparador para o nível "0"	6
9	F(100)	Deteta o nível "0"	2
10	F(101)	Função F(010) invertida	12
11	F(102)	Inverte os níveis "0" e "1"	14
12	F(110)	Função F(001) invertida	8
13	F(111)	V1	0
14	F(112)	Função F(221) invertida	4
15	F(120)	Inversor cíclico anti-horário	16
16	F(121)	Deteta o nível lógico "1"	12
17	F(122)	Deteta o nível lógico "0"	6
18	F(200)	Comparador para o nível "0"	4
19	F(201)	Inversor cíclico horário	16
20	F(202)	Comparador para o nível "1"	14
21	F(210)	Inverte os níveis "0" e "2"	16
22	F(211)	Função F(122) invertida	8
23	F(212)	Deteta o nível lógico "1"	12
24	F(220)	Comparador para o nível "2"	4
25	F(221)	Deteta o nível "2"	2
26	F(222)	V2	0

Tabela 3.1: Relação das funções ternárias simples (funções de apenas uma entrada).

3.2.1 – FUNÇÕES: F(100) e F(011)

As funções F(100) e F(011) são as mais simples de serem implementadas pois o seu circuito correspondem ao de um inversor digital binário CMOS. A detecção do nível lógico de entrada é feita por dois transistores, um NMOS e outro PMOS (Fig. 3.3). Na primeira função, F(100), o transistor PMOS (M1) estará em condução apenas para o nível lógico “0” (Equação 3.1b). Para os níveis lógicos “1” e “2” o transistor NMOS (M2) estará em condução (Equação 3.1a). A função F(011) é obtida através da inversão da função F(100). Estas duas funções apresentam um nível lógico de saída diferente apenas para a entrada igual a “0” e, portanto, são consideradas funções detectoras do nível lógico “0”. Por outro lado, como a sua saída varia apenas entre os níveis “0” e “1”, o seu uso é limitado e elas serão chamadas de “detetoras simples”.

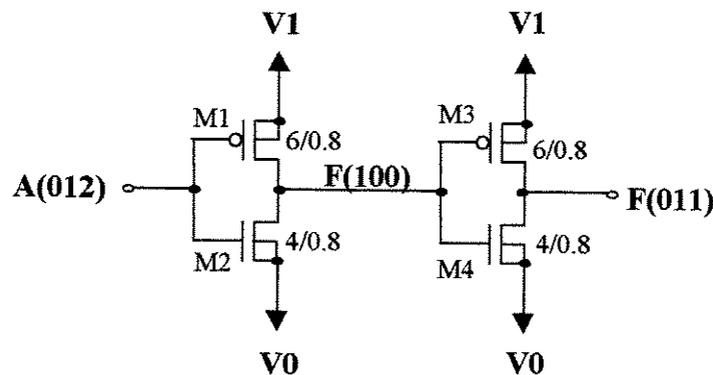


Figura 3.3: Esquema Elétrico das funções F(100) e F(011).

3.2.2 – FUNÇÕES: F(221) e F(112)

As funções F(221) e F(112) são chamadas de funções detectoras simples do nível lógico “2”. A obtenção de F(112) é feita a partir da inversão da função F(221). O circuito que implementa esta função (Fig. 3.4) é semelhante ao das funções F(100) e F(011). As diferenças são as tensões de alimentação (“V1” e “V2”) necessárias para a detecção do nível lógico “2”, e o tamanho dos transistores. O aumento nas dimensões dos transistores NMOS compensa o

aumento do V_t dos transistores tipo N devido ao efeito de corpo (substrato em “V0” enquanto fonte em “V1”).

Os transistores M1 e M3 estarão em condução para tensões abaixo de $(V_2 - V_t)$, ou seja, para os níveis lógicos “0” e “1”. Os transistores M2 e M4 só entrarão em condução para tensões de entrada superiores a $(V_1 + V_t)$, ou seja, para o nível lógico “2”.

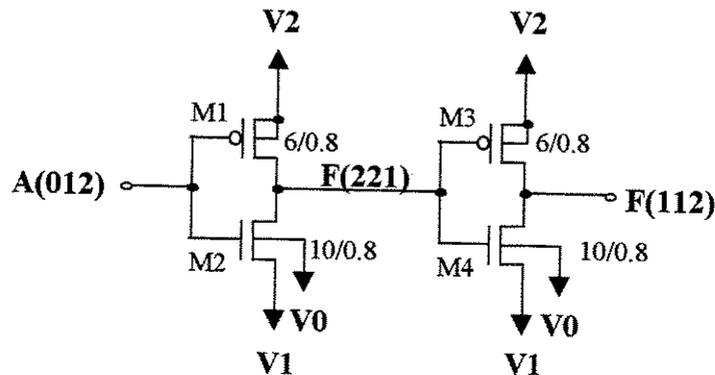


Figura 3.4: Esquema Elétrico das funções F(221) e F(112).

3.2.3 - FUNÇÕES: F(220), F(002), F(001) e F(110)

A função F(220) é chamada de comparador ou “detetor completo” para o nível lógico “2”. Como a tensão de saída varia entre os dois níveis extremos “V0” e “V2”, este circuito pode ser usado no projeto de várias funções: F(020), F(202), F(021), F(102), F(120) e outras.

A célula básica desta função (Fig. 3.5) continua sendo o inversor CMOS formado pelos transistores M1 e M2. Para o nível lógico “0” de entrada, o transistor M1 estará em condução, chaveando a saída para V2. Para o nível lógico “2”, o transistor M1 estará cortado enquanto M2 conectará a saída para V0. A dificuldade aparece para o nível lógico intermediário “1”. Neste caso, o transistor M1 estará conduzindo e para impedirmos a condução de M2 foi necessário acrescentarmos o transistor M3 que, quando em condução, fornecerá uma queda de tensão entre porta e fonte $v_{GS} = V_t$.

A função do diodo D1 é fornecer uma queda de tensão, V_{D1} , quando a porta do transistor M2 estiver sendo descarregada por M4, diminuindo a corrente de fuga. Para a confecção deste diodo usamos uma difusão tipo “p” dentro de um poço “n”.

O transistor M4 funciona como um resistor e foi inserido para possibilitar a descarga da carga capacitiva armazenada na porta de M2 quando este não estiver conduzindo. O valor do resistor pode ser controlado através da tensão de porta indicada como Vref (na simulação foi usado $V_{ref} = V1$). As dimensões de M4 foram definidas através das simulações e o valor encontrado é um compromisso entre velocidade e consumo estático (para os níveis lógicos de entrada: “1” e “2”, conforme Tabela 3.2). Quanto maior o comprimento do canal “L”, maior o valor da resistência, maior o atraso no chaveamento e, por outro lado, menor o consumo.

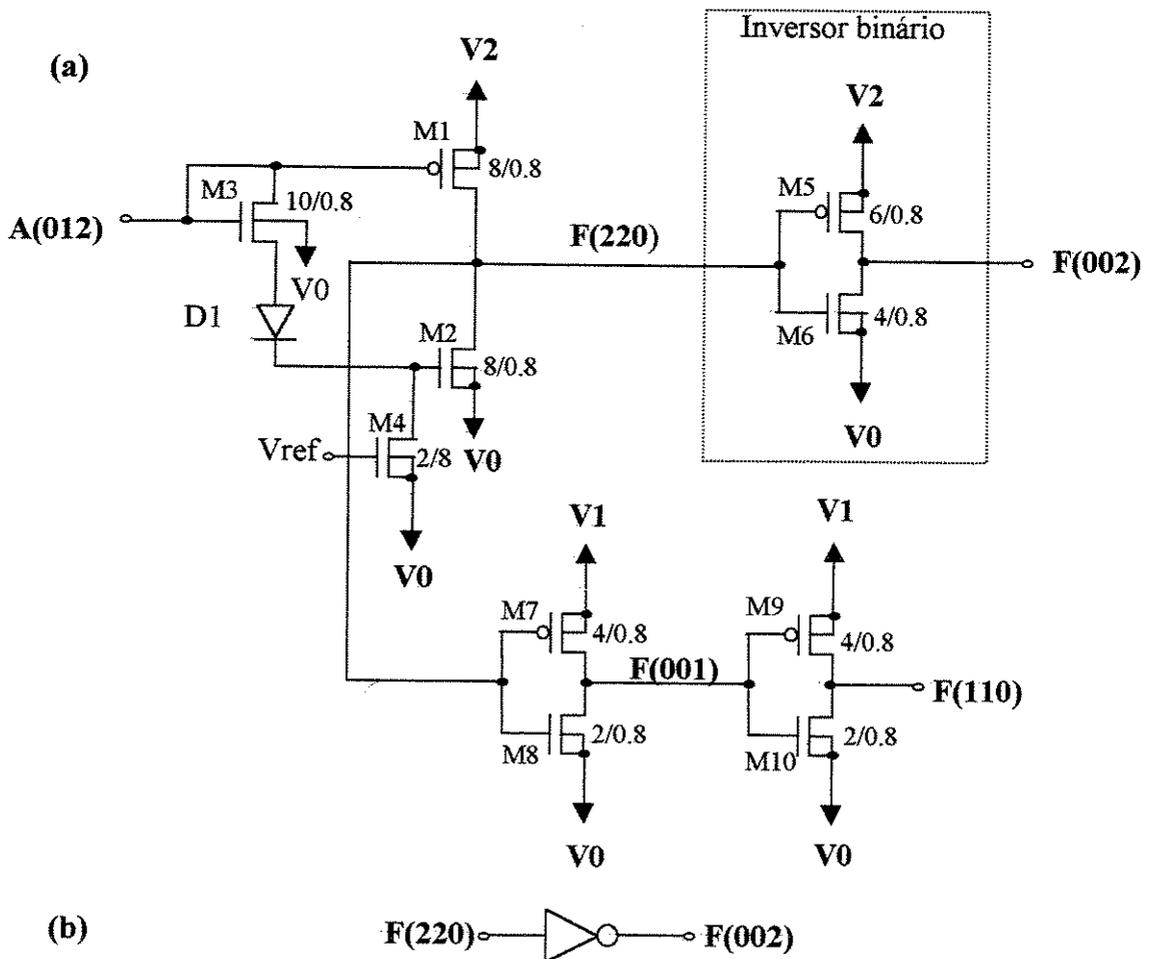


Figura 3.5: (a) Esquema Elétrico das funções F(220), F(002), F(001) e F(110);

(b) Símbolo utilizado para representar a obtenção de F(002) a partir de F(220).

Podemos então perceber que a tensão mínima de entrada, necessária para fazer M2 entrar em condução é $(V_{t_{M3}} + V_{d_{D1}} + V_{t_{M4}})$ que é maior que o nível de tensão usado para o estado lógico “1”.

Na Fig. 3.5(a) o circuito delimitado pela linha tracejada é chamado de inversor pois funciona exatamente como um inversor CMOS binário convencional. Por facilidade de representação, usaremos para este circuito o símbolo do inversor binário mostrado na Fig. 3.5(b). A função F(002) é obtida através da inversão binária de F(220).

A função F(001) também é obtida através da inversão binária de F(220), mas com os níveis lógicos alterados para V1 e V0. O transistor M7, por exemplo, só estará em condução para F(220) = “0”, ou seja, para o nível lógico “2” de entrada (A = “2”). A função F(110) é obtida a partir da simples inversão binária de F(001).

3.2.4 - FUNÇÕES: F(200), F(022), F(122) e F(211)

A função F(200) é chamada de comparador ou “detetor completo” para o nível lógico “0”. A sua saída só será “2” para o nível lógico “0” de entrada; caso contrário, assumirá o valor “0”. Da mesma forma que a função F(220), F(200) também é usada para formar várias outras funções: F(022), F(122), F(211), F(020), F(201), F(021), F(120), etc.

O projeto interno de F(200) (Fig. 3.6) é análogo ao de F(220), com a diferença que o nível lógico de entrada “1” deve agora levar a saída para “0” ao invés de “2”. O transistor M3 será do tipo P, e é usado para acrescentar uma queda de tensão (negativa) $V_{t_{M3}}$ à porta de M1, impedindo-o de entrar em condução para o nível de tensão usado para o valor lógico “1”. O diodo D1 é usado para acrescentar uma queda de tensão (negativa, novamente) quando a porta do transistor M1 estiver sendo descarregada através de M4. A tensão na porta de M4, V_{ref} , controlará o valor da corrente de descarga de M1, sendo que na simulação foi usado o mesmo valor $V_{ref} = V1$. O consumo estático ocorrerá para os níveis lógicos “0” e “1” de entrada.

O valor máximo da tensão de entrada necessária para fazer M1 entrar em condução é $(|V_{t_{M3}}| + |V_{d_{D1}}| + |V_{t_{M1}}|)$ que é menor que o nível de tensão usado para o valor lógico “1”. Os valores de V_t e V_d são usados em módulo pois eles são negativos.

A partir da inversão de $F(200)$, poderemos obter $F(022)$ (transistores M5 e M6). A função $F(122)$ também pode ser obtida a partir de $F(200)$ (transistores M7 e M8), onde o inversor utilizado possui como tensões de alimentação: V_1 e V_2 . O transistor M8 só estará em condução quando a sua porta estiver no nível lógico “2”, ou seja $F(200) = 2$, o que requer que a entrada A esteja no nível “0”. Para os demais valores de entrada ($A=“1”$ ou $A=“2”$), teremos $F(200)=“0”$ e o transistor M7 estará em condução. A função $F(211)$ é obtida através da inversão da função $F(122)$. Podemos observar que ambos os transistores M8 e M10 possuem as suas fontes V_1 enquanto o substrato está em V_0 , o que acarretará o aumento das suas respectivas tensões de threshold (V_t) devido ao efeito de corpo. Para compensar este aumento, a relação W/L dos transistores foi aumentada.

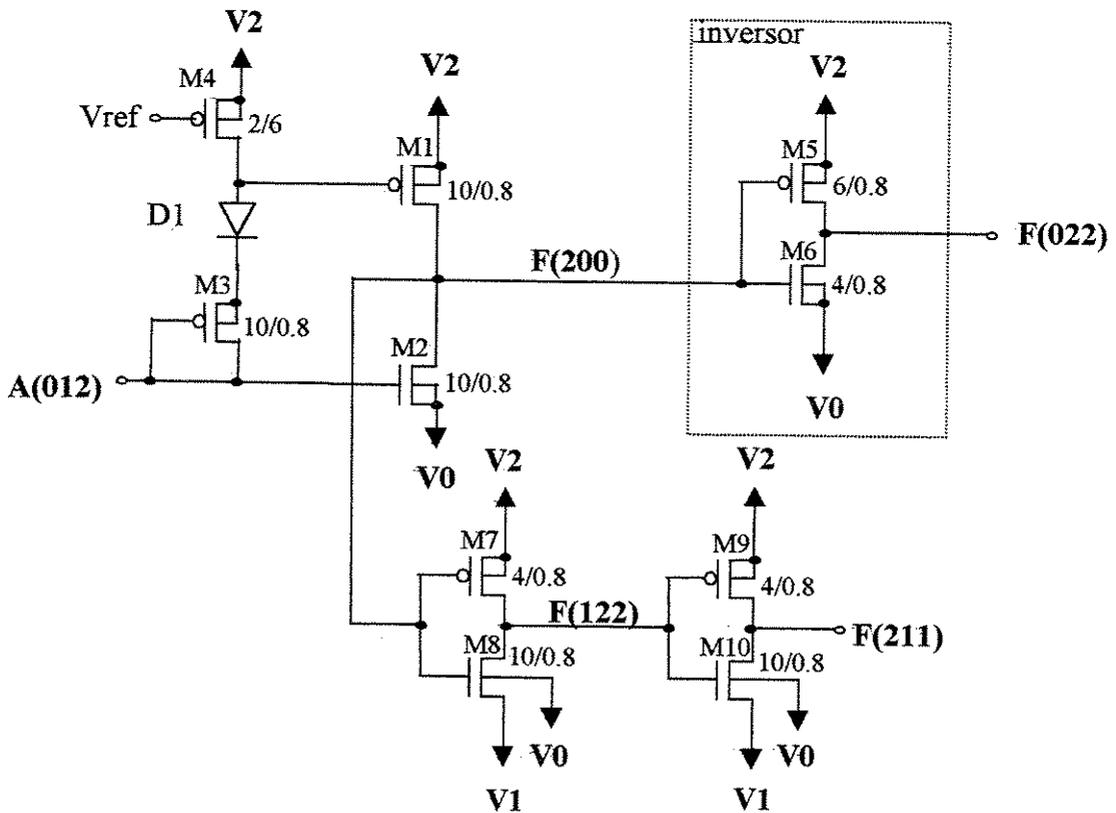


Figura 3.6: Esquema Elétrico das funções $F(200)$, $F(022)$, $F(122)$ e $F(211)$.

As funções $F(200)$ e $F(220)$, ou suas inversas, $F(022)$ e $F(002)$ são muito úteis pois detetam os níveis extremos “0” e “2”, respectivamente. A associação destas funções em série ou paralelo, indicando a interseção ou união dos elementos possibilita o projeto das demais funções que serão vistas a seguir.

3.2.5 - FUNÇÕES: $F(010)$ e $F(101)$

A dificuldade na detecção do nível de tensão intermediário relativo ao nível lógico “1” é que os transistores operam como chaves que detetam tensões acima ou abaixo de um determinado nível (equações 3.1a e 3.1b), ou seja, detetam os níveis extremos “0” ou “2”. A solução encontrada é combinar os circuitos detetores de “0” ou “2” de maneira a chavear a tensão de saída para o nível desejado (“1” para $F(010)$ ou “0” para $F(101)$) quando o nível lógico de entrada não for “0” nem “2”. Para o chaveamento dos transistores serão usadas sempre que possível (dependendo das tensões de alimentação) as funções detetoras simples ($F(100)$, $F(011)$, $F(221)$ ou $F(112)$), obtendo desta forma, um circuito com menor número de transistores.

Na Fig. 3.7 temos o esquema elétrico da função $F(010)$. Podemos observar que os transistores PMOS ($M1$ e $M2$) são usados para chavear a tensão para $V1$. Eles são ligados em série, pois basta que um dos detetores seja verdadeiro (a entrada seja “0” ou “2”) para que a saída não seja “1”. Os mesmos circuitos detetores são usados para chavear a tensão $V0$ para o caso da entrada ser “0” ou “2” (transistores $M3$ e $M4$). O motivo de não necessitarmos inverter os detetores usados para chavear a tensão para $V1$ ou para $V0$ (Capítulo 2 – Seção 2.3) deve-se ao fato de estarmos trabalhando com transistores diferentes (tipo “p” para $V1$ e tipo “n” para $V0$), o que já é uma inversão intrínseca.

A função $F(101)$ pode ser obtida através da simples inversão de função $F(010)$ (Fig. 3.7 - transistores $M5$ e $M6$) ou, independentemente de $F(101)$, como mostrado na Fig. 3.8. A escolha deve ser feita de modo a diminuir o número de transistores para o circuito como um

todo. Ou seja, se a função $F(010)$ já tiver sido gerada, é melhor usar o inversor; caso contrário, o circuito independente é o mais simples.

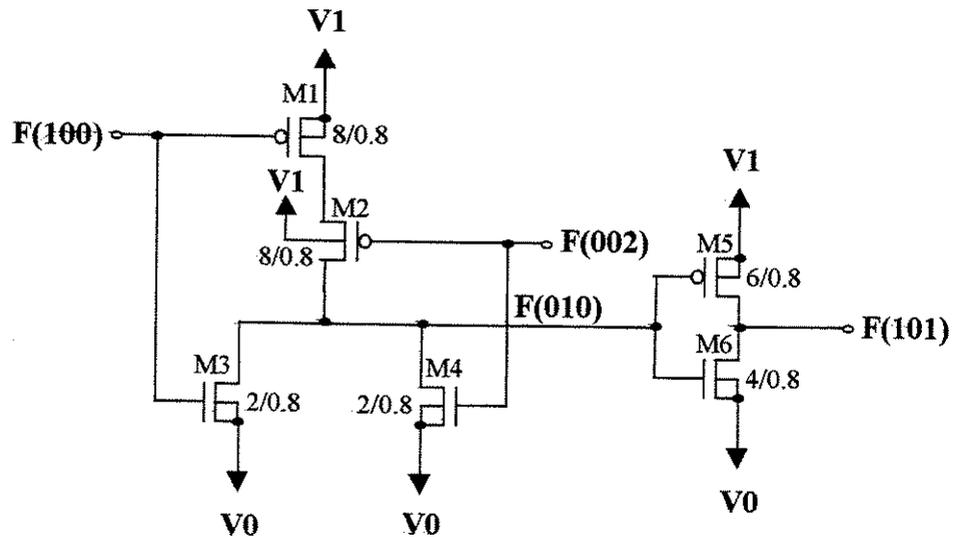


Figura 3.7: Esquema Elétrico das funções $F(010)$ e $F(101)$.

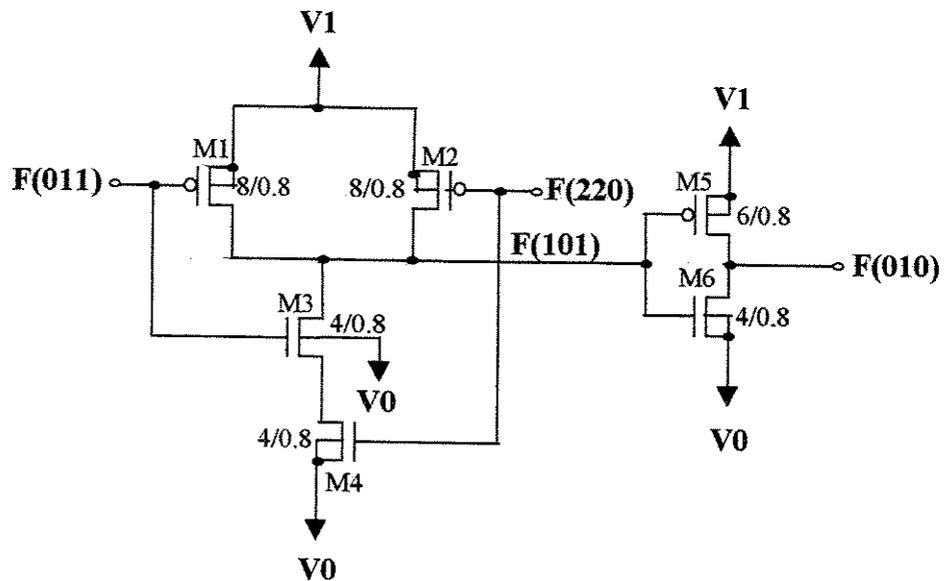


Figura 3.8: Esquema Elétrico alternativo para as funções $F(010)$ e $F(101)$.

3.2.6 - FUNÇÕES: $F(121)$ e $F(212)$

Estas funções são obtidas da mesma forma que as funções $F(010)$ e $F(101)$, ou seja, a detecção do nível "1" é obtida a partir da negação dos detetores dos níveis "0" e "2". A principal

diferença está nas tensões de alimentação que agora são V2 e V1. Os circuitos detetores foram alterados para poder chavear estas novas tensões.

Como no caso das funções anteriores, podemos gerar F(212) a partir de F(121) ou vice-versa. Estas duas opções estão mostradas nas Figuras 3.9 e 3.10, respectivamente.

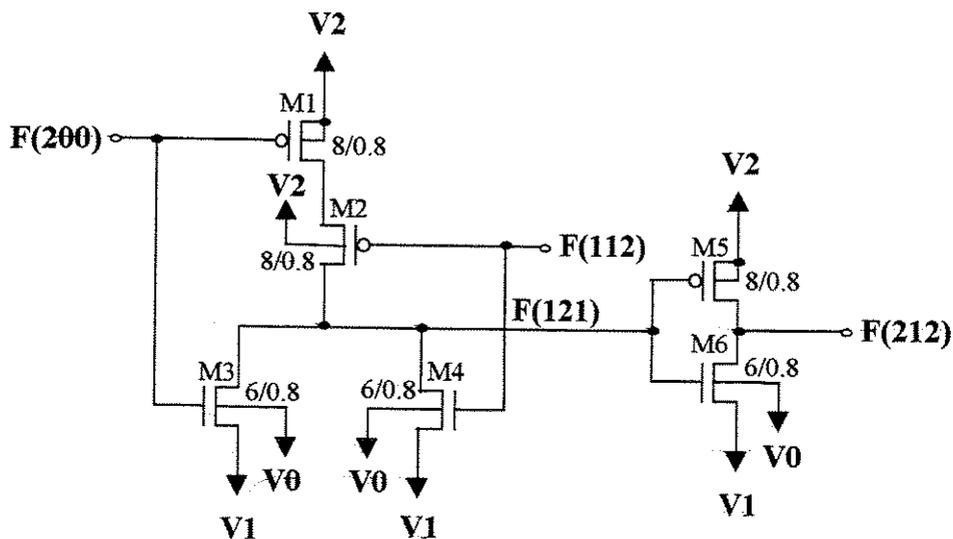


Figura 3.9: Esquema Elétrico das funções F(121) e F(212).

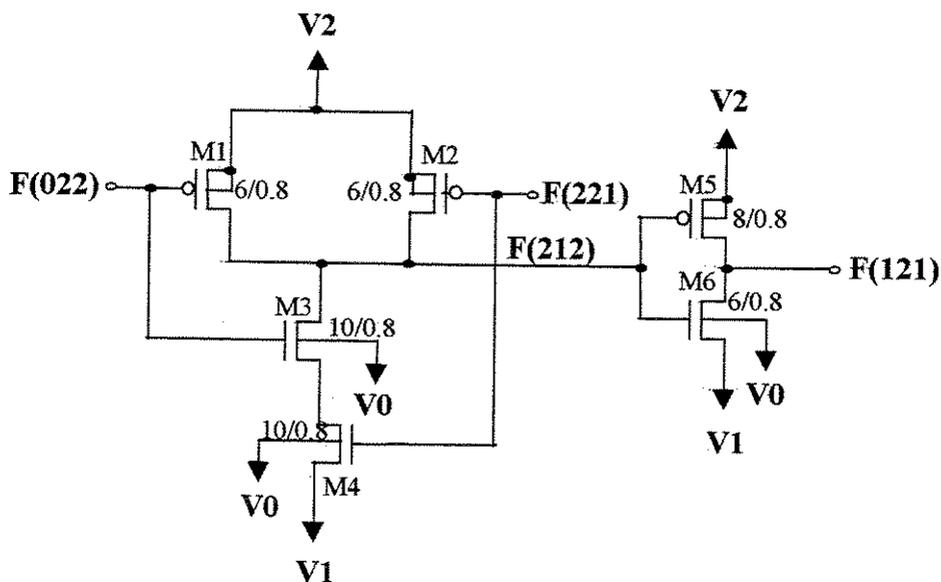


Figura 3.10: Esquema Elétrico alternativo para as funções F(212) e F(121).

Na Fig. 3.9, os transistores M1 e M2 chavearão a saída, F(121), para V2 apenas quando a entrada for “1”; caso contrário, um dos transistores estará cortado. Por outro lado, se a entrada for “0”, o transistor M3 forçará a saída para V1, ou, se a entrada for “2”, M4 fará a saída ir para V1. Os transistores M5 e M6 formam o circuito inversor que gerará a função F(212).

O funcionamento do circuito da Fig. 3.10 é análogo, mas a função F(212) é gerada primeiro e depois invertida (transistores M5 e M6) para formar F(121). Os transistores M1 e M2 chavearão F(212) para V2 quando a entrada for “0” ou “2”, respectivamente. Por outro lado, apenas quando os dois transistores M4 e M5 estiverem conduzindo, ou seja, quando a entrada for “1”, a saída será chaveada para V1.

3.2.7 - FUNÇÕES: F(020) e F(202)

Estas funções são chamadas de detetores completos ou comparadores para o nível lógico “1” pois suas saídas variam entre os valores extremos V0 e V2.

Na Fig. 3.11 temos a geração da função F(020), e sua inversa, F(202). Na Fig. 3.12 primeiro geramos F(202) e, através da sua inversão, F(020).

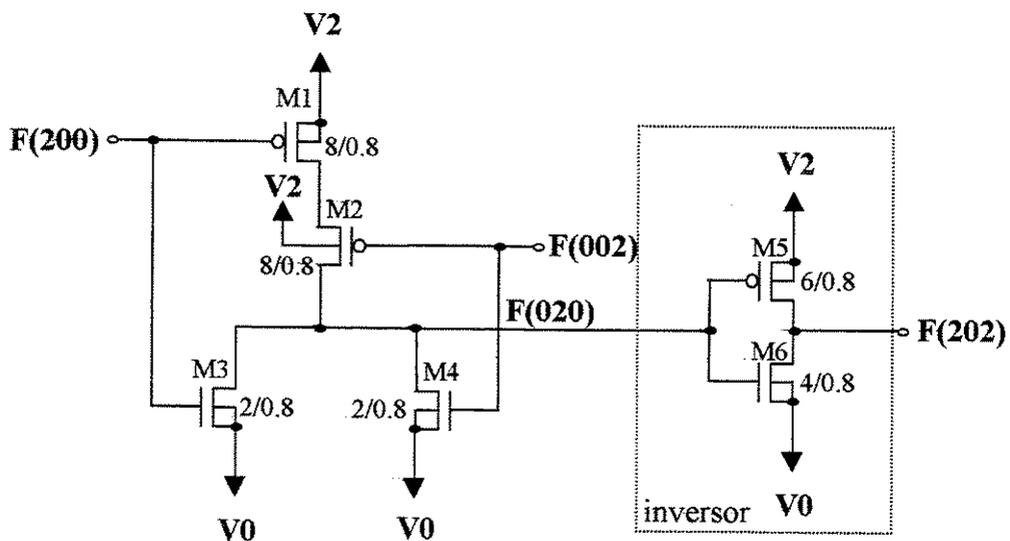


Figura 3.11: Esquema Elétrico das funções F(020) e F(202).

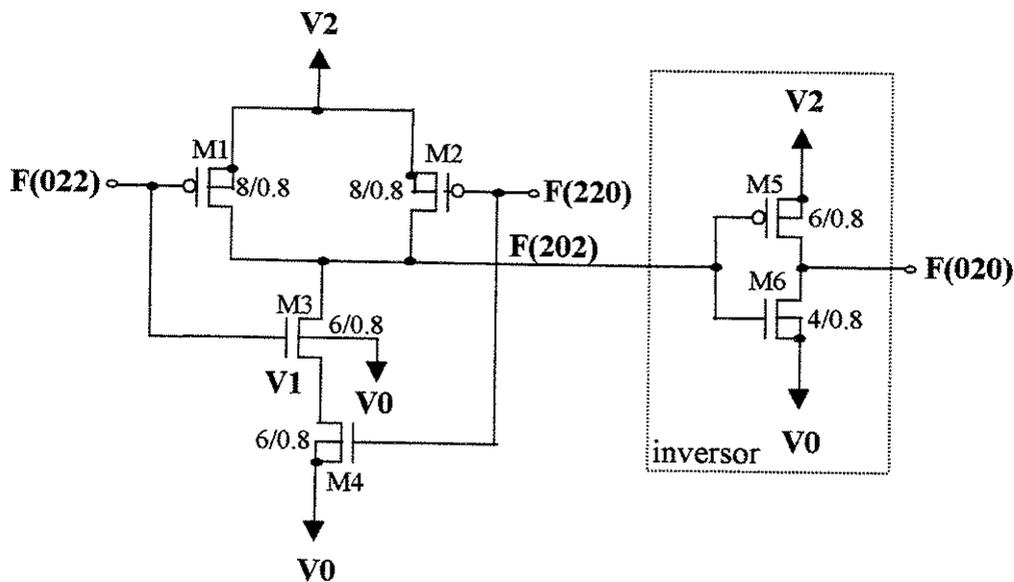


Figura 3.12: Esquema Elétrico alternativo para as funções F(202) e F(020).

Da comparação dos esquemas elétricos podemos perceber que as únicas diferenças entre as funções F(020) e F(202) em relação às funções F(010) e F(101), são as tensões de alimentação (V2 em vez de V1) e a utilização do comparador para o nível lógico “0”, F(200), em vez de F(100). A mudança para o detetor completo foi necessária para chavear V2 pois se usássemos F(100), o transistor M1 (Fig. 3.11) estaria sempre em condução.

Os circuitos usados para a deteção do nível lógico intermediário “1” são os mais complexos pois necessitam dos detetores dos outros dois níveis, “0” e “2”. Este aspecto deve ser levado em consideração na simplificação dos circuitos através do diagrama gráfico (Capítulo 2), de modo a evitar, quando possível, a comparação com o nível intermediário “1”.

3.2.8 - FUNÇÃO F(120)

O projeto de funções que possuem como saída os três níveis lógicos, como é o caso de F(120), necessita de circuitos comparadores para, pelo menos, dois níveis lógicos. Como vimos anteriormente, o comparador para o nível lógico “1” é o mais difícil de ser gerado e, portanto,

usaremos sempre os detetores para os níveis “0” e “2”. Esses comparadores possuem um pequeno consumo estático, que também estará presente nestas funções. A entrada será considerada “1” se os comparadores para os outros dois níveis for negativo. O chaveamento de V0 é feito com transistores tipo “n” enquanto o chaveamento de V2 é feito com transistores tipo “p”. Desta forma evitamos a queda de tensão V_t entre o dreno e a fonte dos transistores (Equações 3.1a e 3.1b). Para o nível lógico “1” podemos usar tanto transistores “P” quanto “N”.

A função F(120) é chamada de inversor cíclico horário, ou seja, dada uma entrada “0”, “1” ou “2”, a saída será sempre o próximo nível lógico à direita, como indicado na Fig. 3.13.

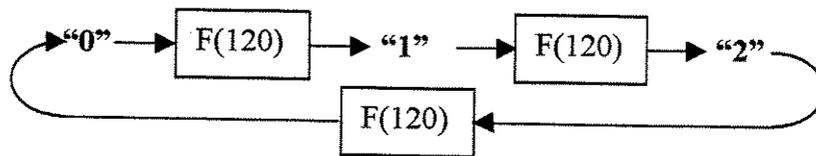


Figura 3.13: Representação gráfica da função F(120)

Na Fig. 3.14 temos o esquema elétrico da função F(120). A função F(022) deteta o nível “0” e chaveia M1 para V1. A função F(002) deteta o nível “2” e chaveia M4 para V0. Os dois transistores M2 e M3 são usados em série para o chaveamento de V2 quando a entrada não for “0” nem “2”.

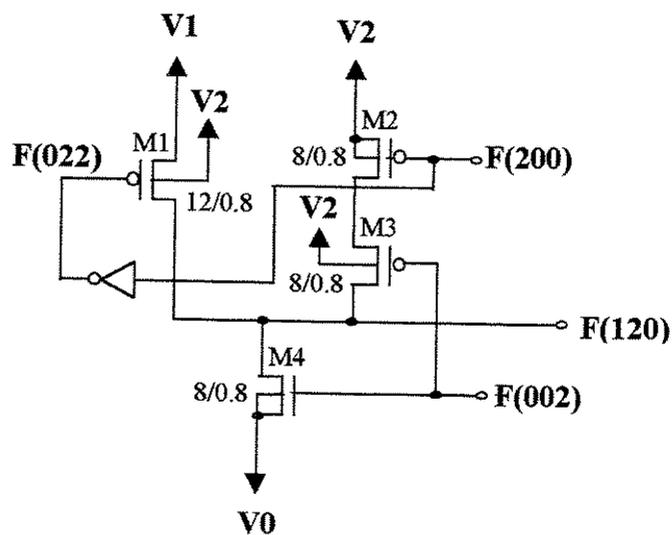


Figura 3.14: Esquema Elétrico da função F(120)

chaveará a função para V2. A saída será chaveada para V0 quando os comparadores para os níveis “0” e “2” forem negativos. Como no caso da função F(201), o chaveamento para V1 foi feito com transistor NMOS.

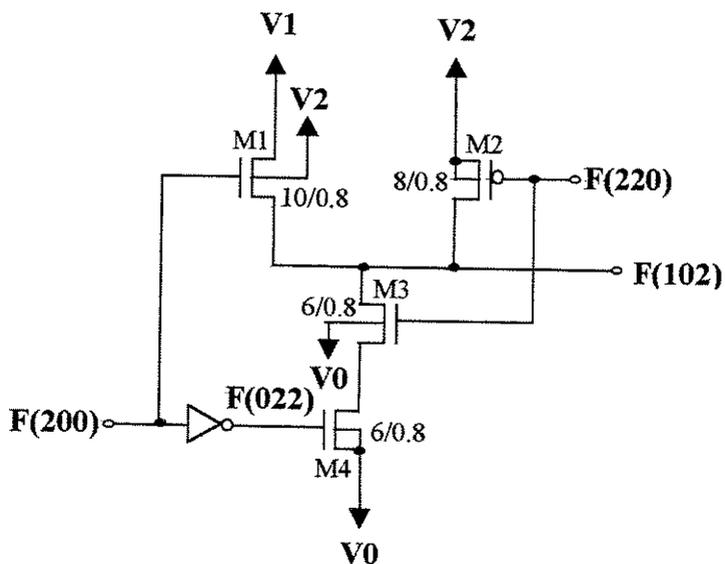


Figura 3.18: Esquema Elétrico da função F(102)

3.2.12 - FUNÇÃO F(210)

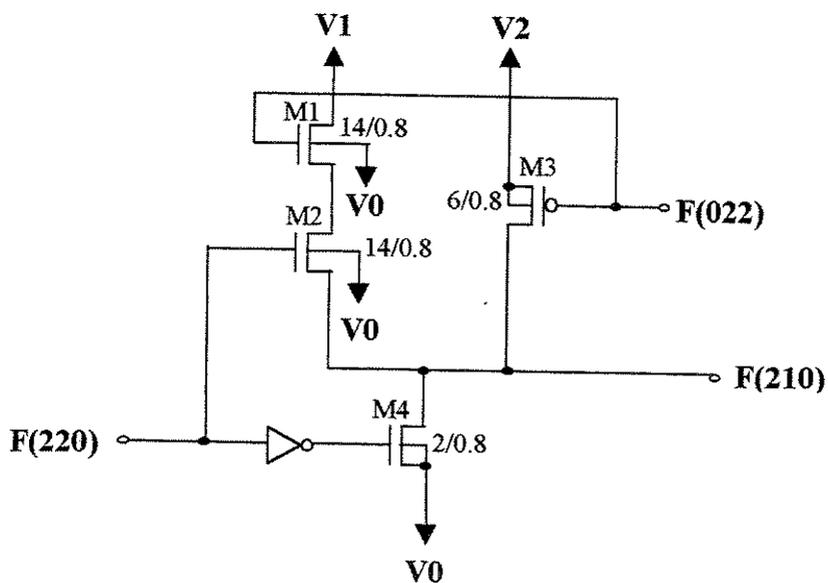


Figura 3.19: Esquema Elétrico da função F(210)

A função F(210) inverte os níveis lógicos “0” e “2”, mantendo o nível lógico “1” de entrada. O projeto da função é semelhante aos vistos anteriormente: A entrada é comparada aos níveis lógicos “0” e “2” através das funções F(022) e F(002), que chavearão a saída para as tensões V2 e V0, respectivamente. O chaveamento para V1 ocorrerá pela negação destes comparadores, que indicará o nível lógico “1” de entrada. Como são necessários dois transistores para chavearmos V1 e também temos aumento de V_t devido ao efeito de corpo, a relação W/L desses transistores teve de ser aumentada. O circuito correspondente é mostrado na Fig. 3.19.

3.2.13 - FUNÇÃO F(012)

F(012) é a função ternária equivalente ao “buffer” não inversor binário convencional (Fig. 3.20). A saída é exatamente o mesmo nível lógico da entrada, porém a sua alimentação é através das fontes externas de tensão. Este circuito será muito útil no projeto de circuitos registradores (“flip-flops”), pois uma vez realimentados, estes circuitos conseguem manter o nível lógico por tempo indeterminado. Esta função também é utilizada como “buffer ternário”, sendo utilizada no projeto dos “buffers” de saída (Capítulo 5).

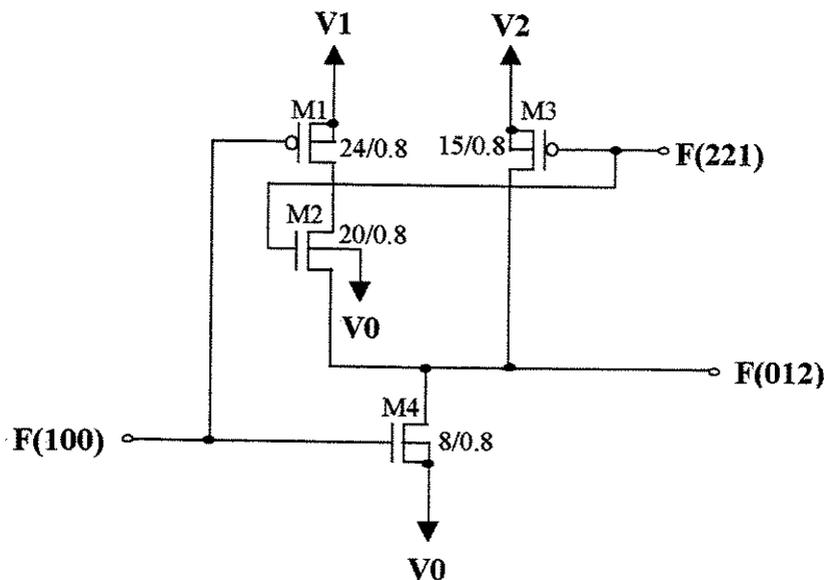


Figura 3.20: Esquema Elétrico da função F(012).

Entre as funções ternárias que apresentam três níveis lógicos de saída, F(012) é a que necessita de menor número de transistores pois os circuitos detectores dos níveis lógicos “0” e “2” são as funções mais simples de serem implementadas: F(100) e F(221), respectivamente. No entanto, os transistores usados para chavear o nível lógico “1” são maiores que nas demais funções. Isto ocorre porque a detecção do nível lógico “1” é feita pela negação dos outros dois níveis lógicos “0” e “2”, e portanto, temos dois transistores em série. Além disso, como a tensão do nível lógico “1” é mais baixa, o tempo necessário para carregar/descarregar o capacitor de porta do próximo estágio é maior, como ocorre com a função F(210).

Outro fator a ser ressaltado é a posição relativa dos transistores M1 e M2. Para evitarmos o efeito de corpo em M1, o seu substrato foi conectado à V1. Como consequência, o dreno de M1 não pode ser conectado ao dreno de M3 para evitarmos a polarização direta do substrato de M1 quando M3 estiver em condução (tensão de saída igual a V2). O transistor M2 provê esta isolação.

3.3- RESULTADOS DAS SIMULAÇÕES SPICE DAS FUNÇÕES

DESCRIÇÃO DOS PARÂMETROS MEDIDOS NA SIMULAÇÃO:

As funções ternárias descritas na Seção 3.2 foram simuladas para as três condições de processo fornecidas pelo fabricante: típica, maior atraso e maior consumo.

Os atrasos foram medidos da saída da função em relação à entrada. O instante de tempo escolhido foi aquele em que o sinal (de entrada ou saída) atinge o ponto médio de tensão entre os dois níveis lógicos próximos. No caso da variação ocorrer entre os níveis lógicos extremos (“0” para “2” ou “2” para “0”), o valor intermediário (nível lógico “1”) ocorrerá por um breve período de tempo e o atraso será medido quando a tensão estiver no nível intermediário entre os níveis lógicos “1” e “2”, ou “0” e “1”. Estes valores estão resumidos a seguir:

Varição de 0 \Rightarrow 1	tensão = 0,75V
Varição de 1 \Rightarrow 2	tensão = 2,25V
Varição de 2 \Rightarrow 0	tensão = 0,75V
Varição de 0 \Rightarrow 2	tensão = 2,25V
Varição de 2 \Rightarrow 1	tensão = 2,25V

Considere por exemplo os sinais indicados na Fig. 3.21, que representam a função F(120). Supondo que a entrada varie inicialmente do nível lógico “0” para o nível “1”, a saída variará do nível lógico “1” para o nível “2”. O instante considerado para a medida dos atrasos, conforme indicado na Fig. 3.21, será o ponto médio de tensão entre estes níveis lógicos; no caso, a diferença de tempo entre a entrada $V_{in} = 0,75V$ e $F(120) = 2,25V$.

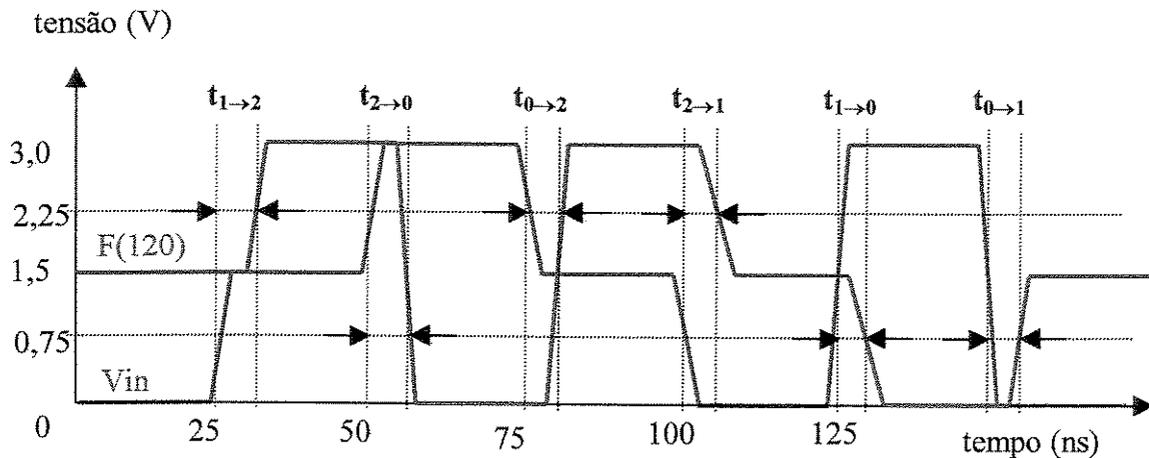


Figura 3.21: Representação dos atrasos de saída para a função F(120).

Note que na representação do tempo de atraso, o valor lógico considerado refere-se à saída. Por exemplo, o símbolo $t_{0 \rightarrow 1}$ indica o atraso da saída em relação à entrada para a variação da saída do nível lógico “0” para o nível “1,” e assim por diante.

Na Tabela 3.2 temos o resumo dos resultados das simulações feitas. A tabela mostra os valores de atraso da saída em relação à entrada para as diversas variações lógicas da saída. As funções foram simuladas para os três valores de parâmetros fornecidos pelo fabricante (AMS 0.8 μ m): típico, lento e rápido. O valor de consumo foi obtido pela simulação do processo rápido (pior

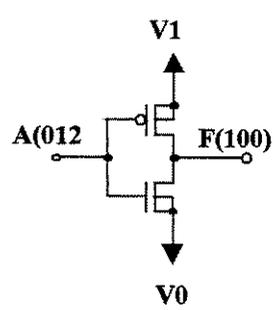
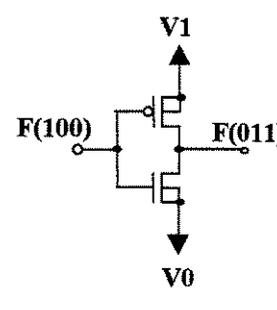
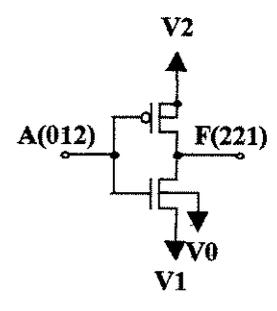
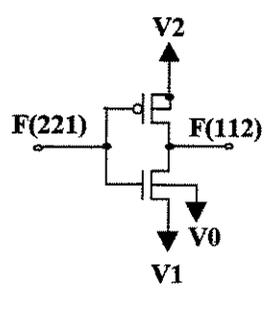
consumo) e indica o consumo estático para os três níveis lógicos de entrada: “0”, “1” e “2”. O consumo dinâmico pode ser observado através do gráfico de corrente através dos transistores. No caso de funções mais complexas, que utilizam outras funções (por exemplo, F(120) que utiliza F(002) e F(200)), traçamos apenas as correntes através do circuito principal. As correntes através dos outros circuitos (no caso do exemplo, F(002) e F(200)), podem ser vistas nos gráficos correspondentes a essas funções.

Os valores de área e perímetro de dreno e fonte dos transistores foram obtidos a partir do “layout” destas funções. Para a simulação de cada função foi usado um circuito inversor simples (como o da Fig. 3.5) com relação W/L de 4.0/0.8 para o transistor PMOS e de 2.0/0.8 para o transistor NMOS (a alimentação do inversor foi a apropriada para cada função).

Em algumas situações, duas variações diferentes nos valores lógicos de entrada levam à mesma variação lógica da saída. É o caso da função F(020). A saída variará de “0” para “2” tanto para a variação da entrada de “0” para “1”, como de “2” para “1”. Em casos como este, o atraso relacionado na tabela corresponde ao pior caso obtido.

Na lógica ternária, a mudança do valor lógico “0” para o “2” ou vice-versa, implicará em um valor transitório pelo nível lógico intermediário “1”. Esse fato, em alguns casos, acarreta “spikes” (picos de tensão) nas saídas. Por exemplo, a função F(120) apresentará “spikes” para a variação da entrada de “0” para “2” e vice-versa (vide gráfico correspondente). Esses “spikes” não apresentam problemas para circuitos combinacionais comuns, pois a saída é amostrada apenas quando a saída está estável. Cuidados especiais de amostragem devem ser tomados no caso de saída usadas como relógio ou sinais de “enable” (habilitadores).

No final da Seção apresentamos as várias formas de onda obtidas para cada função, para o caso típico dos parâmetros do fabricante. Além dos gráficos das tensões de saída em relação à entrada, mostramos as correntes através dos transistores. As formas de onda de entrada apresentadas seguem um mesmo padrão (“0”, “1”, “2”, “1”, “0”), que é o máximo de transições permitidas por simulação. No entanto todos os circuitos foram simulados para todas as possibilidades de transições e os atrasos medidos encontram-se na Tabela 3.1.

FUNÇÃO	ATRASOS/ CONSUMO	TIPOS DE PROCESSO			CIRCUITO
		TÍPICO	LENTO	RÁPIDO	
F(100)	$t_{0 \rightarrow 1}$	0,4ns	0,9ns	0,3ns	
	$t_{0 \rightarrow 2}$	-	-	-	
	$t_{1 \rightarrow 0}$	0,6ns	0,7ns	0,4ns	
	$t_{1 \rightarrow 2}$	-	-	-	
	$t_{2 \rightarrow 0}$	-	-	-	
	$t_{2 \rightarrow 1}$	-	-	-	
	consumo*	4,6pW / 4,5pW / 4,5pW			
F(011)	$t_{0 \rightarrow 1}$	0,9ns	2,0ns	0,5ns	
	$t_{0 \rightarrow 2}$	-	-	-	
	$t_{1 \rightarrow 0}$	1,1ns	2,2ns	0,5ns	
	$t_{1 \rightarrow 2}$	-	-	-	
	$t_{2 \rightarrow 0}$	-	-	-	
	$t_{2 \rightarrow 1}$	-	-	-	
	consumo*	6,9pW / 6,8pW / 6,8pW			
F(221)	$t_{0 \rightarrow 1}$	-	-	-	
	$t_{0 \rightarrow 2}$	-	-	-	
	$t_{1 \rightarrow 0}$	-	-	-	
	$t_{1 \rightarrow 2}$	0,7ns	1,2ns	0,4ns	
	$t_{2 \rightarrow 0}$	-	-	-	
	$t_{2 \rightarrow 1}$	0,4ns	1,6ns	0,2ns	
	consumo*	18,0pW / 18,0pW / 18,0pW			
F(112)	$t_{0 \rightarrow 1}$	-	-	-	
	$t_{0 \rightarrow 2}$	-	-	-	
	$t_{1 \rightarrow 0}$	-	-	-	
	$t_{1 \rightarrow 2}$	1,1ns	4,0ns	0,5ns	
	$t_{2 \rightarrow 0}$	-	-	-	
	$t_{2 \rightarrow 1}$	1,4ns	4,0ns	0,7ns	
	consumo*	29,3pW / 29,3pW / 24,8pW			

* Consumo estático para o processo rápido relativo aos três níveis lógicos de entrada: "0"/"1"/"2".

Tabela 3.2: Resultados das simulações SPICE para as diversas funções (1ª parte)

FUNÇÃO	ATRASOS/ CONSUMO	TIPOS DE PROCESSO			CIRCUITO
		TÍPICO	LENTO	RÁPIDO	
F(220)	$t_{0 \rightarrow 1}$	-	-	-	
	$t_{0 \rightarrow 2}$	6,0ns	4,3ns	4,1ns	
	$t_{1 \rightarrow 0}$	-	-	-	
	$t_{1 \rightarrow 2}$	-	-	-	
	$t_{2 \rightarrow 0}$	0,9ns	6,0ns	0,4ns	
	$t_{2 \rightarrow 1}$	-	-	-	
	consumo*	19,5pW / 4,3μW / 14,9μW			
F(002)	$t_{0 \rightarrow 1}$	-	-	-	
	$t_{0 \rightarrow 2}$	1,1ns	5,8ns	0,5ns	
	$t_{1 \rightarrow 0}$	-	-	-	
	$t_{1 \rightarrow 2}$	-	-	-	
	$t_{2 \rightarrow 0}$	4,4ns	3,3ns	3,8ns	
	$t_{2 \rightarrow 1}$	-	-	-	
	consumo*	28,6pW / 4,3μW / 14,9μW			
F(001)	$t_{0 \rightarrow 1}$	1,4ns	7,2ns	0,6ns	
	$t_{0 \rightarrow 2}$	-	-	-	
	$t_{1 \rightarrow 0}$	4,1ns	2,5ns	3,0ns	
	$t_{1 \rightarrow 2}$	-	-	-	
	$t_{2 \rightarrow 0}$	-	-	-	
	$t_{2 \rightarrow 1}$	-	-	-	
	consumo*	15,0pW / 4,3μW / 14,9μW			
F(110)	$t_{0 \rightarrow 1}$	4,5ns	3,9ns	3,1ns	
	$t_{0 \rightarrow 2}$	-	-	-	
	$t_{1 \rightarrow 0}$	2,1ns	9,5ns	0,9ns	
	$t_{1 \rightarrow 2}$	-	-	-	
	$t_{2 \rightarrow 0}$	-	-	-	
	$t_{2 \rightarrow 1}$	-	-	-	
	consumo*	17,2pW / 4,3μW / 14,9μW			

* Consumo estático para o processo rápido relativo aos três níveis lógicos de entrada: "0"/"1"/"2".

Tabela 3.2: Resultados das simulações SPICE para as diversas funções (continuação)

FUNÇÃO	ATRASOS/ CONSUMO	TIPOS DE PROCESSO			CIRCUITO
		TÍPICO	LENTO	RÁPIDO	
F(200)	$t_{0 \rightarrow 1}$	-	-	-	
	$t_{0 \rightarrow 2}$	2,6ns	6,8ns	1,2ns	
	$t_{1 \rightarrow 0}$	-	-	-	
	$t_{1 \rightarrow 2}$	-	-	-	
	$t_{2 \rightarrow 0}$	0,6ns	1,1ns	0,5ns	
	$t_{2 \rightarrow 1}$	-	-	-	
	consumo*	7,0 μ W / 1,1 μ W / 1,8pW			
F(022)	$t_{0 \rightarrow 1}$	-	-	-	
	$t_{0 \rightarrow 2}$	0,8ns	1,5ns	0,5ns	
	$t_{1 \rightarrow 0}$	-	-	-	
	$t_{1 \rightarrow 2}$	-	-	-	
	$t_{2 \rightarrow 0}$	2,5ns	6,3ns	1,2ns	
	$t_{2 \rightarrow 1}$	-	-	-	
	consumo*	7,0 μ W / 1,1 μ W / 27,5pW			
F(122)	$t_{0 \rightarrow 1}$	-	-	-	
	$t_{0 \rightarrow 2}$	-	-	-	
	$t_{1 \rightarrow 0}$	-	-	-	
	$t_{1 \rightarrow 2}$	0,8ns	1,5ns	0,5ns	
	$t_{2 \rightarrow 0}$	-	-	-	
	$t_{2 \rightarrow 1}$	3,4ns	11,3ns	1,4ns	
	consumo*	7,0 μ W / 1,1 μ W / 27,0pW			
F(211)	$t_{0 \rightarrow 1}$	-	-	-	
	$t_{0 \rightarrow 2}$	-	-	-	
	$t_{1 \rightarrow 0}$	-	-	-	
	$t_{1 \rightarrow 2}$	4,4ns	14,4ns	1,8ns	
	$t_{2 \rightarrow 0}$	-	-	-	
	$t_{2 \rightarrow 1}$	1,2ns	3,6ns	0,7ns	
	consumo	9,5 μ W / 1,4 μ W / 38,3pW			

* Consumo estático para o processo rápido relativo aos três níveis lógicos de entrada: "0"/"1"/"2".

Tabela 3.2: Resultados das simulações SPICE para as diversas funções (continuação)

FUNÇÃO	ATRASOS/ CONSUMO	TIPOS DE PROCESSO			CIRCUITO
		TÍPICO	LENTO	RÁPIDO	
F(010)	$t_{0 \rightarrow 1}$	5,7ns	4,8ns	4,0ns	
	$t_{0 \rightarrow 2}$	-	-	-	
	$t_{1 \rightarrow 0}$	1,9ns	5,8ns	0,9ns	
	$t_{1 \rightarrow 2}$	-	-	-	
	$t_{2 \rightarrow 0}$	-	-	-	
	$t_{2 \rightarrow 1}$	-	-	-	
	consumo*	29,5pW / 4,3μW / 14,9μW			
F(101)	$t_{0 \rightarrow 1}$	1,8ns	9,0ns	0,9ns	
	$t_{0 \rightarrow 2}$	-	-	-	
	$t_{1 \rightarrow 0}$	4,4ns	4,2ns	3,0ns	
	$t_{1 \rightarrow 2}$	-	-	-	
	$t_{2 \rightarrow 0}$	-	-	-	
	$t_{2 \rightarrow 1}$	-	-	-	
	consumo*	19,6pW / 4,3μW / 14,9μW			
F(121)	$t_{0 \rightarrow 1}$	-	-	-	
	$t_{0 \rightarrow 2}$	-	-	-	
	$t_{1 \rightarrow 0}$	-	-	-	
	$t_{1 \rightarrow 2}$	2,2ns	7,5ns	1,2ns	
	$t_{2 \rightarrow 0}$	-	-	-	
	$t_{2 \rightarrow 1}$	4,2ns	13,8ns	1,6ns	
	consumo*	9,5μW / 1,4μW / 49,5pW			
F(212)	$t_{0 \rightarrow 1}$	-	-	-	
	$t_{0 \rightarrow 2}$	-	-	-	
	$t_{1 \rightarrow 0}$	-	-	-	
	$t_{1 \rightarrow 2}$	2,7ns	6,7ns	1,2ns	
	$t_{2 \rightarrow 0}$	-	-	-	
	$t_{2 \rightarrow 1}$	1,8ns	7,3ns	0,8ns	
	consumo*	9,5μW / 1,4μW / 59,0pW			

* Consumo estático para o processo rápido relativo aos três níveis lógicos de entrada: "0"/"1"/"2".

Tabela 3.2: Resultados das simulações SPICE para as diversas funções (continuação)

FUNÇÃO	ATRASOS/ CONSUMO	TIPOS DE PROCESSO			CIRCUITO
		TÍPICO	LENTO	RÁPIDO	
F(020)	$t_{0 \rightarrow 1}$	-	-	-	
	$t_{0 \rightarrow 2}$	5,4ns	3,7ns	3,7ns	
	$t_{1 \rightarrow 0}$	-	-	-	
	$t_{1 \rightarrow 2}$	-	-	-	
	$t_{2 \rightarrow 0}$	3,4ns	8,8ns	1,5ns	
	$t_{2 \rightarrow 1}$	-	-	-	
	consumo*	9,5 μ W / 5,6 μ W / 14,9 μ W			
F(202)	$t_{0 \rightarrow 1}$	-	-	-	
	$t_{0 \rightarrow 2}$	2,8ns	7,0ns	1,3ns	
	$t_{1 \rightarrow 0}$	-	-	-	
	$t_{1 \rightarrow 2}$	-	-	-	
	$t_{2 \rightarrow 0}$	5,2ns	3,7ns	3,8ns	
	$t_{2 \rightarrow 1}$	-	-	-	
	consumo*	9,5 μ W / 5,6 μ W / 14,9 μ W			
F(120)	$t_{0 \rightarrow 1}$	3,1ns	8,4ns	1,3ns	
	$t_{0 \rightarrow 2}$	5,6ns	4,3ns	4,0ns	
	$t_{1 \rightarrow 0}$	1,1ns	6,3ns	0,5ns	
	$t_{1 \rightarrow 2}$	0,9ns	1,7ns	0,6ns	
	$t_{2 \rightarrow 0}$	1,3ns	5,7ns	0,5ns	
	$t_{2 \rightarrow 1}$	3,2ns	8,2ns	1,5ns	
	consumo*	9,5 μ W / 5,6 μ W / 14,9 μ W			
F(201)	$t_{0 \rightarrow 1}$	1,4ns	7,1ns	0,6ns	
	$t_{0 \rightarrow 2}$	2,8ns	7,1ns	1,3ns	
	$t_{1 \rightarrow 0}$	4,7ns	3,1ns	3,3ns	
	$t_{1 \rightarrow 2}$	2,7ns	7,1ns	1,2ns	
	$t_{2 \rightarrow 0}$	1,1ns	2,0ns	0,7ns	
	$t_{2 \rightarrow 1}$	1,2ns	7,5ns	0,5ns	
	consumo*	9,5 μ W / 5,6 μ W / 14,9 μ W			

* Consumo estático para o processo rápido relativo aos três níveis lógicos de entrada: "0"/"1"/"2".

Tabela 3.2: Resultados das simulações SPICE para as diversas funções (continuação)

FUNÇÃO	ATRASOS/ CONSUMO	TIPOS DE PROCESSO			CIRCUITO
		TÍPICO	LENTO	RÁPIDO	
F(021)	$t_{0 \rightarrow 1}$	2,2ns	11,0ns	0,4ns	
	$t_{0 \rightarrow 2}$	1,1ns	2,0ns	0,7ns	
	$t_{1 \rightarrow 0}$	2,2ns	6,0ns	0,9ns	
	$t_{1 \rightarrow 2}$	5,3ns	3,8ns	3,4ns	
	$t_{2 \rightarrow 0}$	2,7ns	6,9ns	1,2ns	
	$t_{2 \rightarrow 1}$	1,8ns	10,6ns	0,7ns	
	consumo*	9,5 μ W / 5,6 μ W / 14,9 μ W			
F(102)	$t_{0 \rightarrow 1}$	2,9ns	8,2ns	1,3ns	
	$t_{0 \rightarrow 2}$	1,3ns	6,9ns	0,6ns	
	$t_{1 \rightarrow 0}$	0,9ns	1,8ns	0,6ns	
	$t_{1 \rightarrow 2}$	1,2ns	6,8ns	0,5ns	
	$t_{2 \rightarrow 0}$	5,2ns	3,7ns	3,7ns	
	$t_{2 \rightarrow 1}$	3,0ns	8,7ns	1,7ns	
	consumo*	9,5 μ W / 5,6 μ W / 14,9 μ W			
F(210)	$t_{0 \rightarrow 1}$	5,6ns	4,7ns	3,7ns	
	$t_{0 \rightarrow 2}$	2,6ns	6,7ns	1,2ns	
	$t_{1 \rightarrow 0}$	1,6ns	7,7ns	0,7ns	
	$t_{1 \rightarrow 2}$	2,7ns	6,5ns	1,2ns	
	$t_{2 \rightarrow 0}$	1,8ns	8,5ns	0,7ns	
	$t_{2 \rightarrow 1}$	1,4ns	5,1ns	0,7ns	
	consumo*	9,5 μ W / 5,6 μ W / 14,9 μ W			
F(012)	$t_{0 \rightarrow 1}$	1,4ns	3,3ns	0,8ns	
	$t_{0 \rightarrow 2}$	1,6ns	8,6ns	0,7ns	
	$t_{1 \rightarrow 0}$	1,3ns	5,8ns	0,6ns	
	$t_{1 \rightarrow 2}$	2,4ns	5,7ns	1,0ns	
	$t_{2 \rightarrow 0}$	2,4ns	7,5ns	1,0ns	
	$t_{2 \rightarrow 1}$	2,4ns	7,5ns	1,0ns	
	consumo*	27,1pW / 25,0pW / 33,0pW			

* Consumo estático para o processo rápido relativo aos três níveis lógicos de entrada: "0"/"1"/"2".

Tabela 3.2: Resultados das simulações SPICE para as diversas funções (continuação).

FUNÇÃO F(100)

Processo Típico

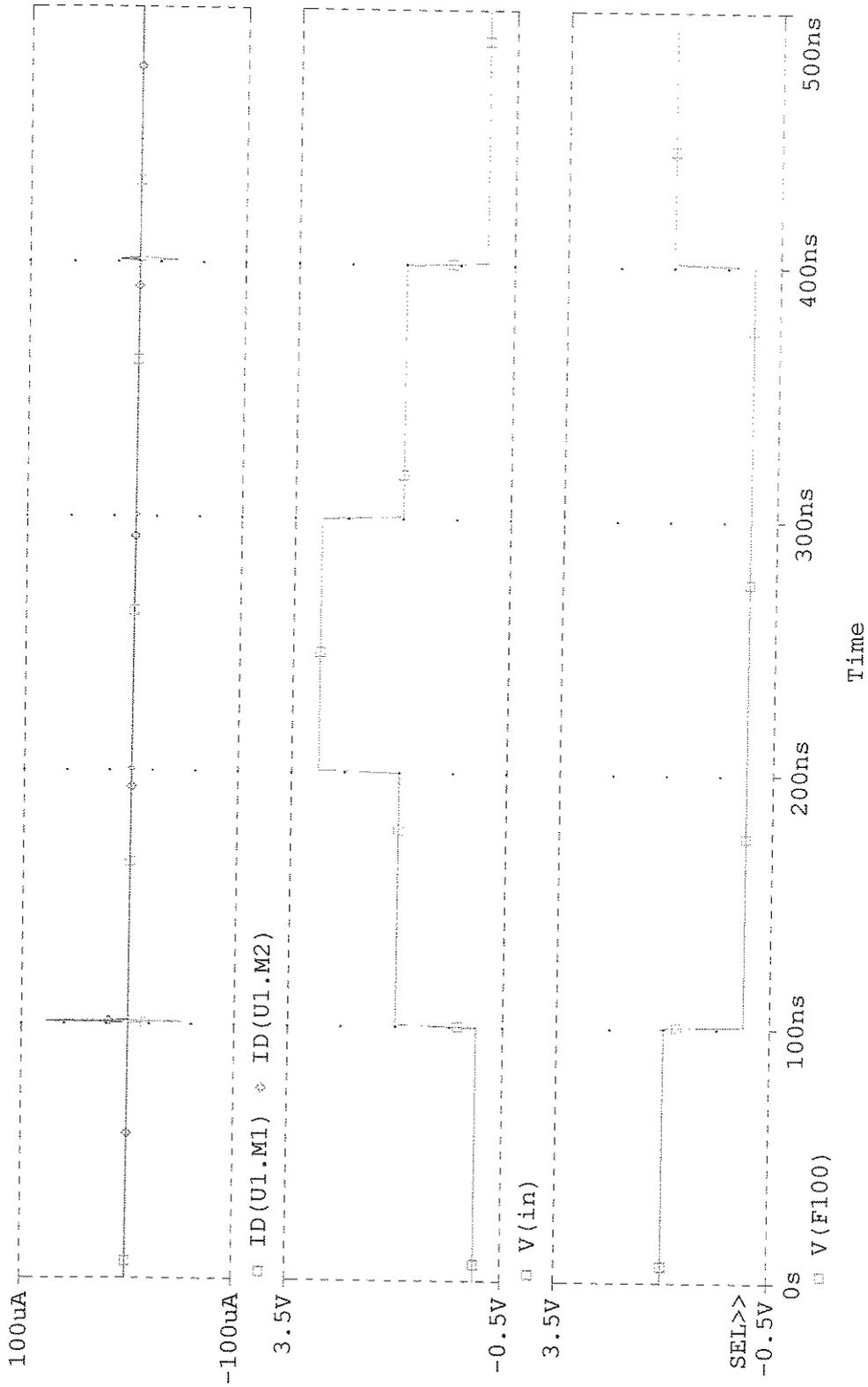


Figura 3.22: Simulação da função F(100).

FUNÇÃO F(011)

Processo Típico

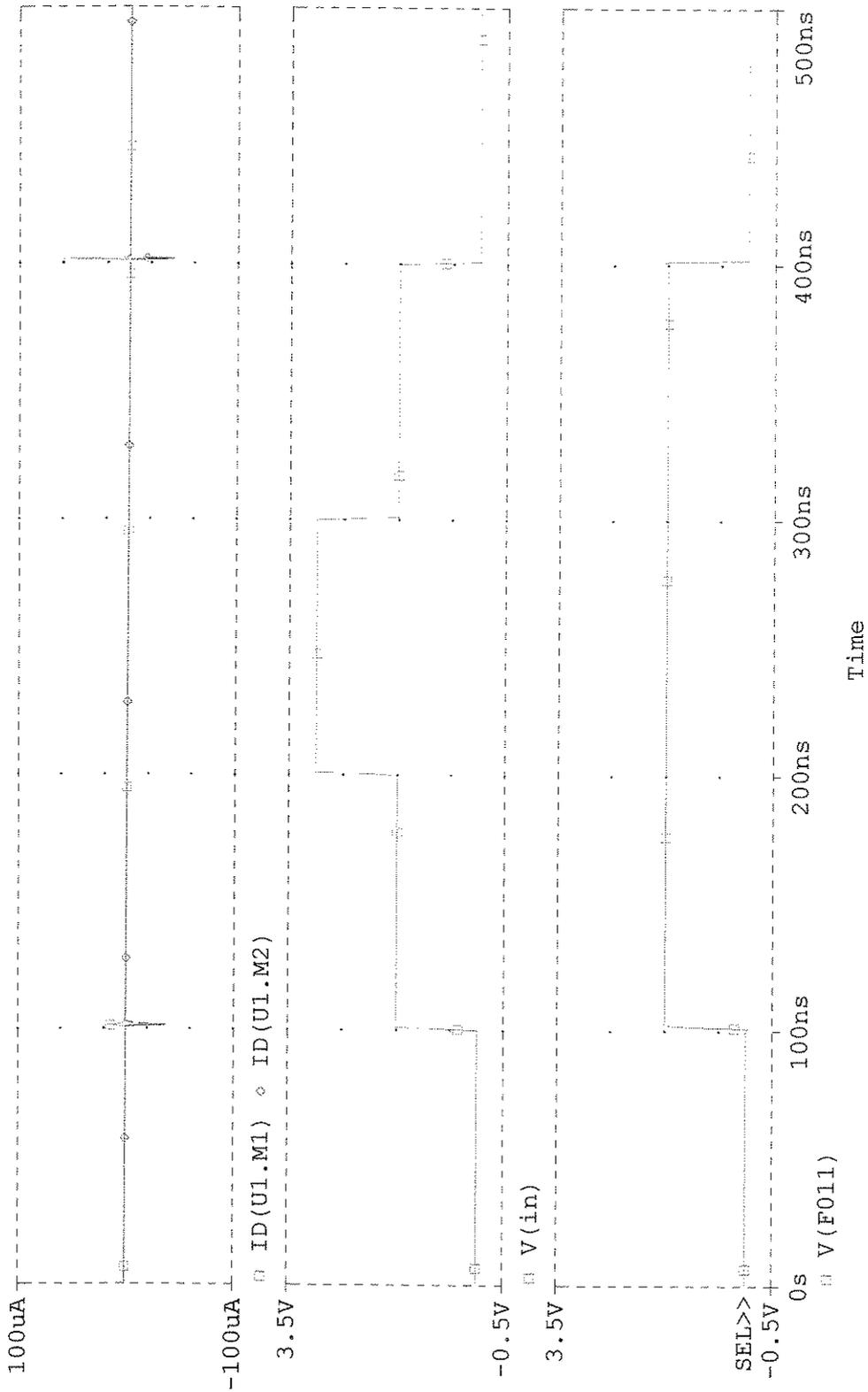


Figura 3.23: Simulação da função F(011).

FUNÇÃO F (221)

Processo Típico

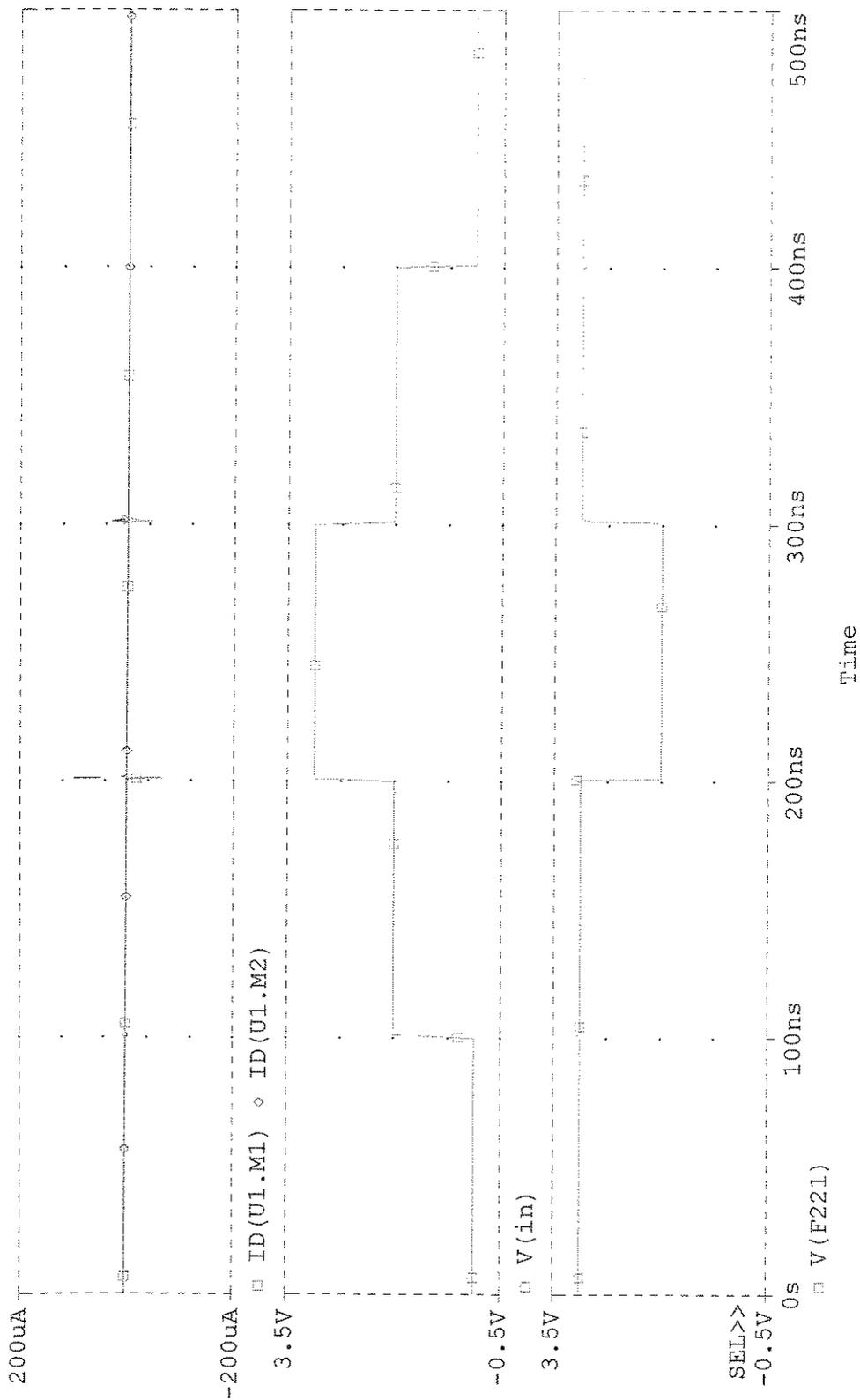


Figura 3.24: Simulação da função F(221).

FUNÇÃO F(112)

Processo Típico

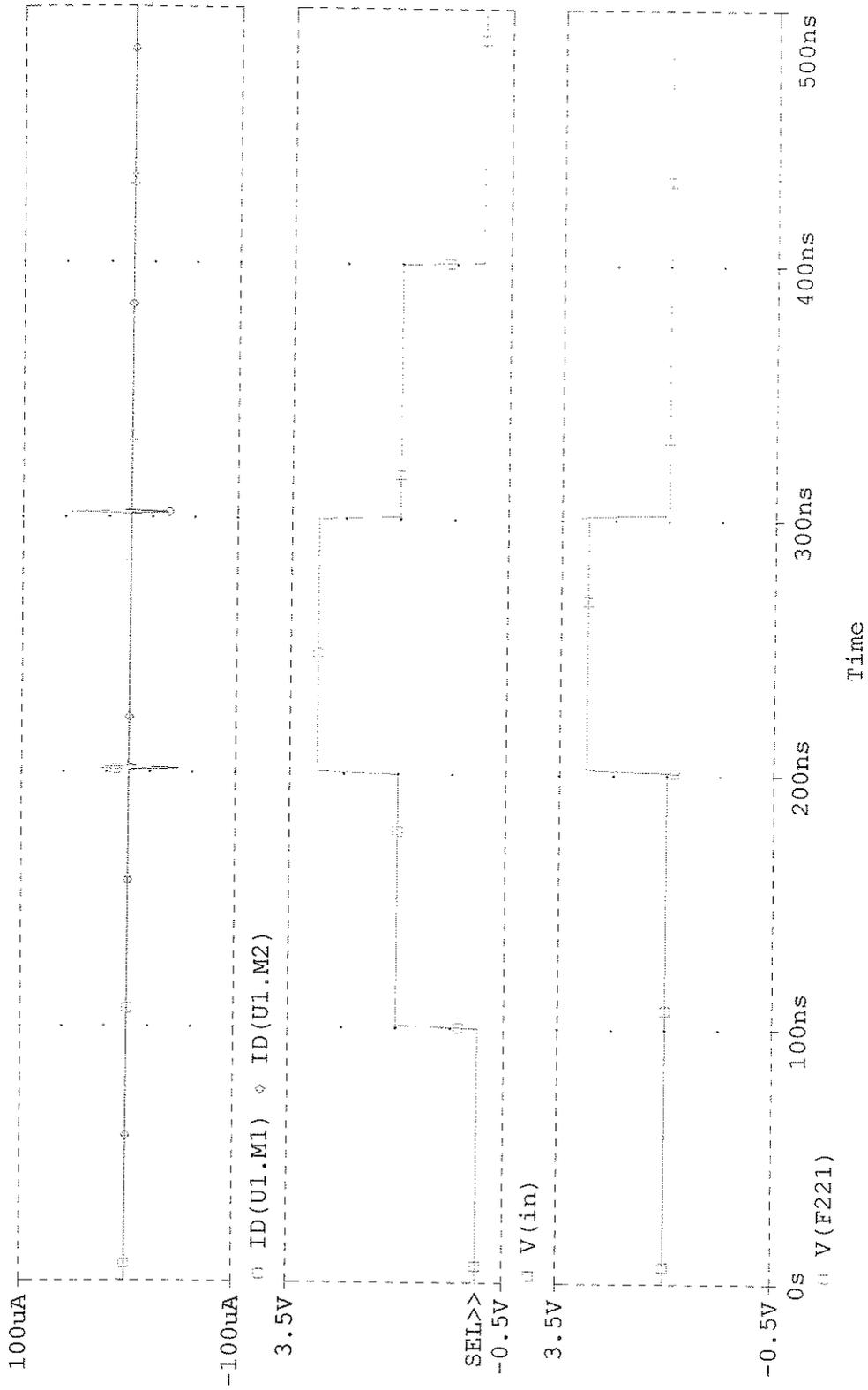


Figura 3.25: Simulação da função F(112).

FUNÇÃO F (220)

Processo Típico

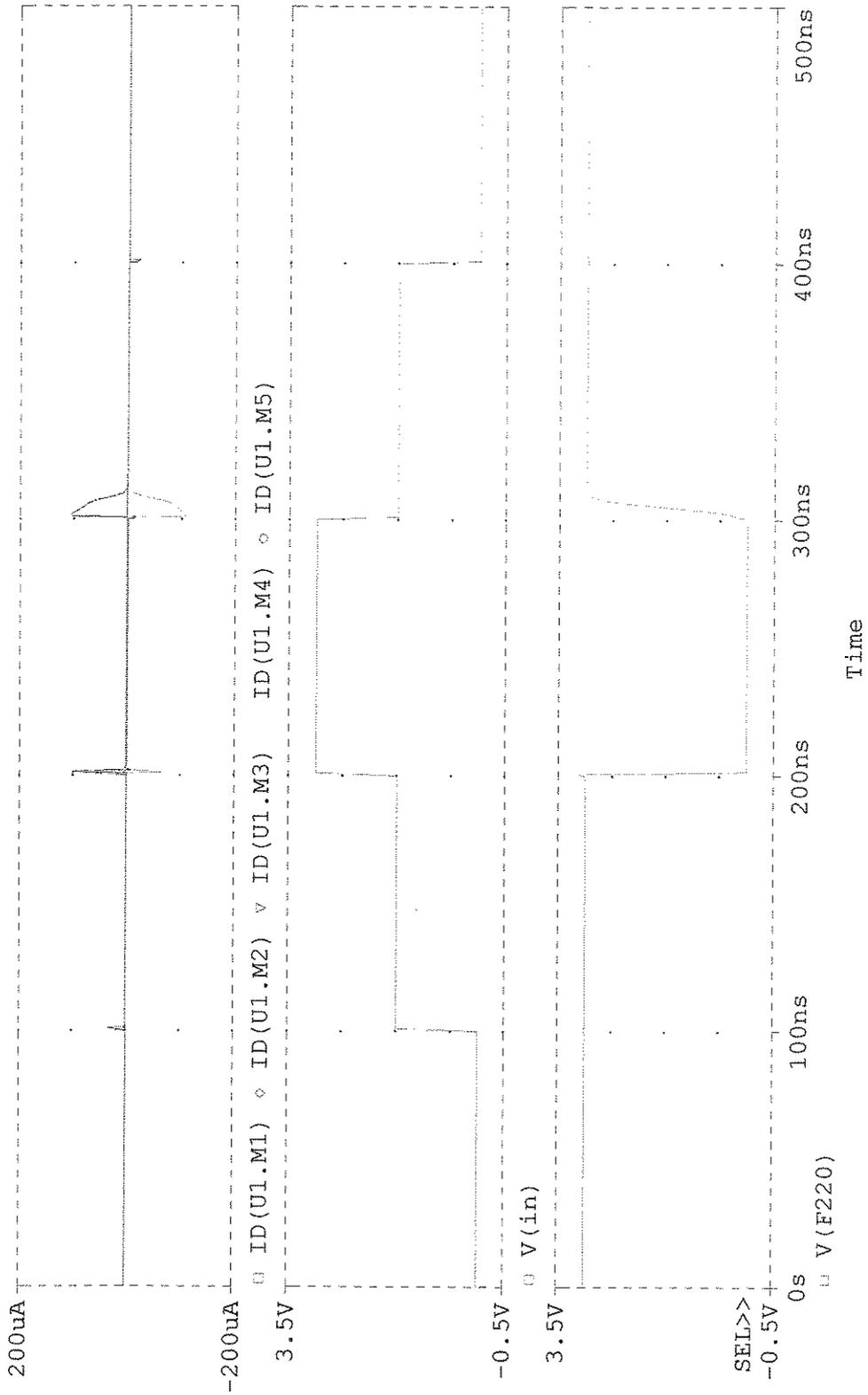


Figura 3.26: Simulação da função F (220) .

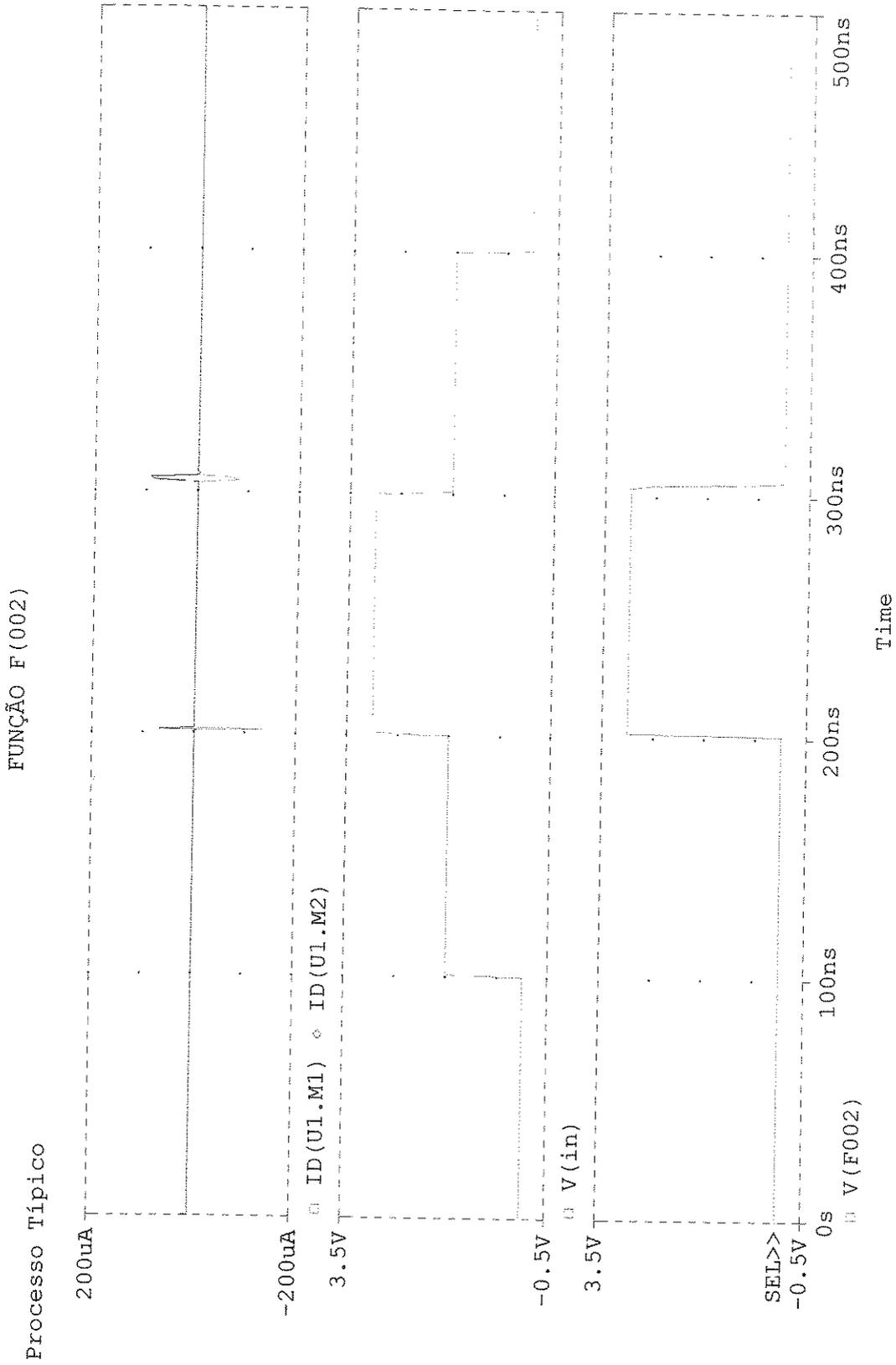


Figura 3.27: Simulação da função F (002).

FUNÇÃO F (001)

Processo Típico

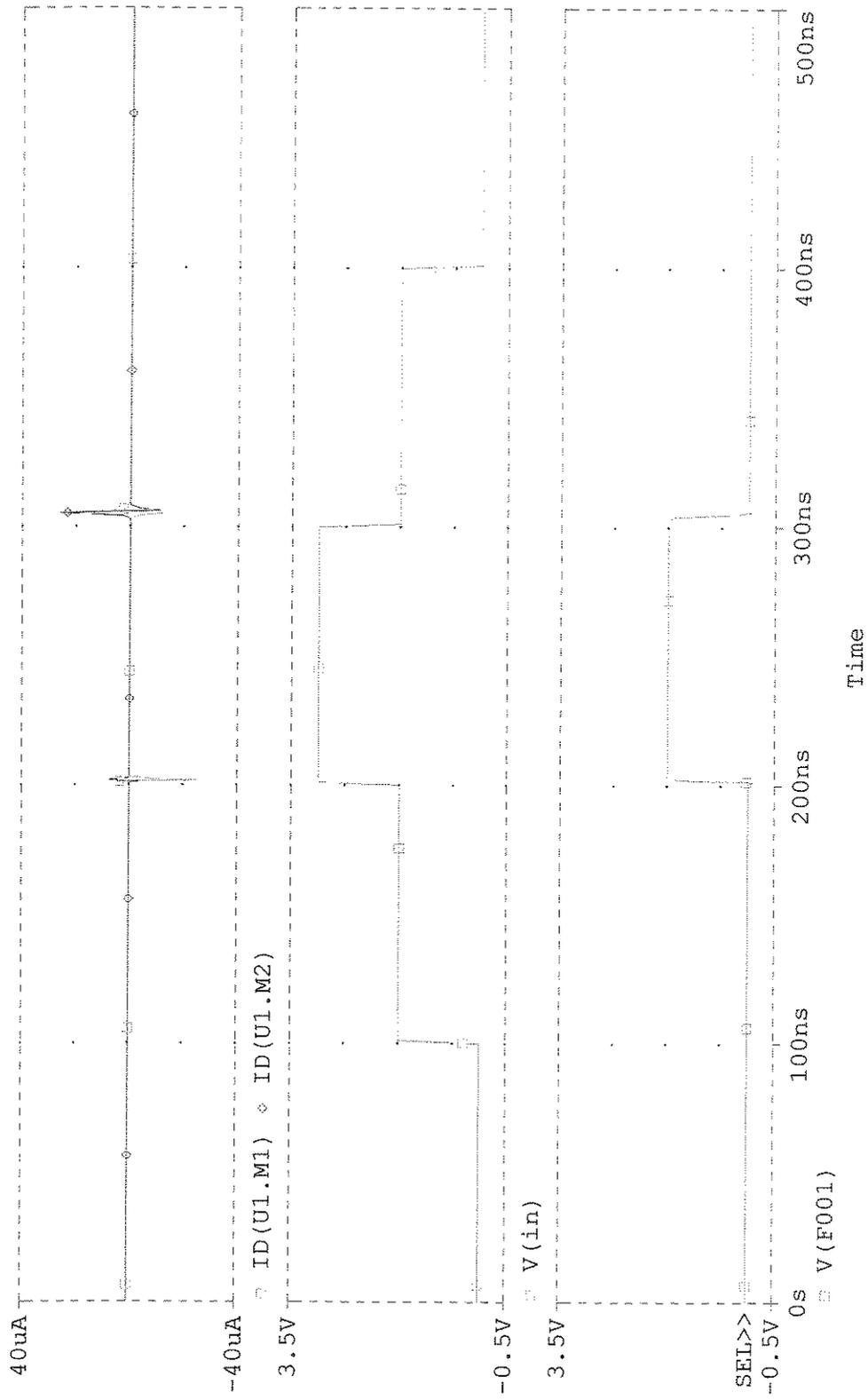


Figura 3.28: Simulação da função F (001).

FUNÇÃO F (110)

Processo Típico

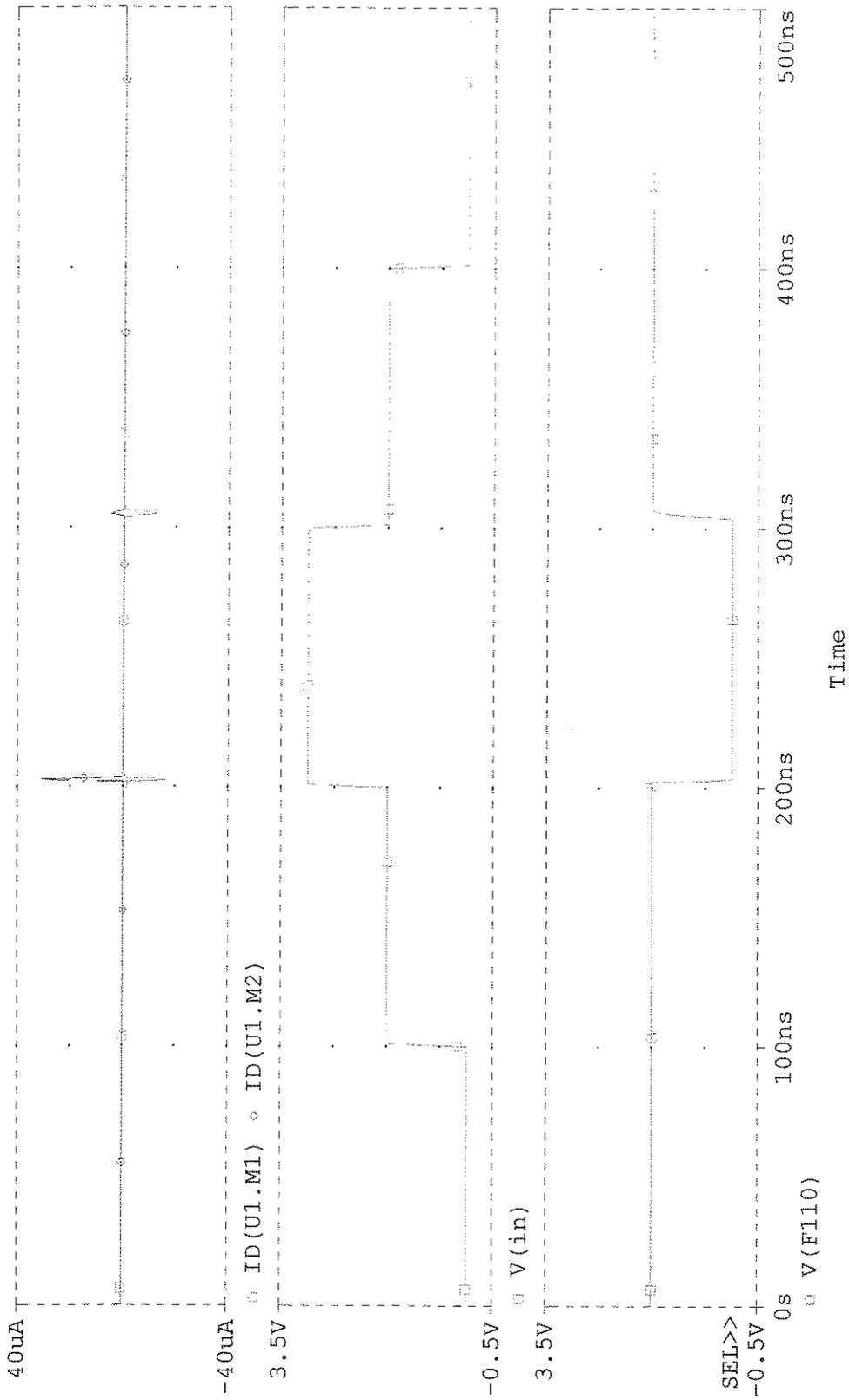


Figura 3.29: Simulação da função F(110).

FUNÇÃO F(200)

Processo Típico

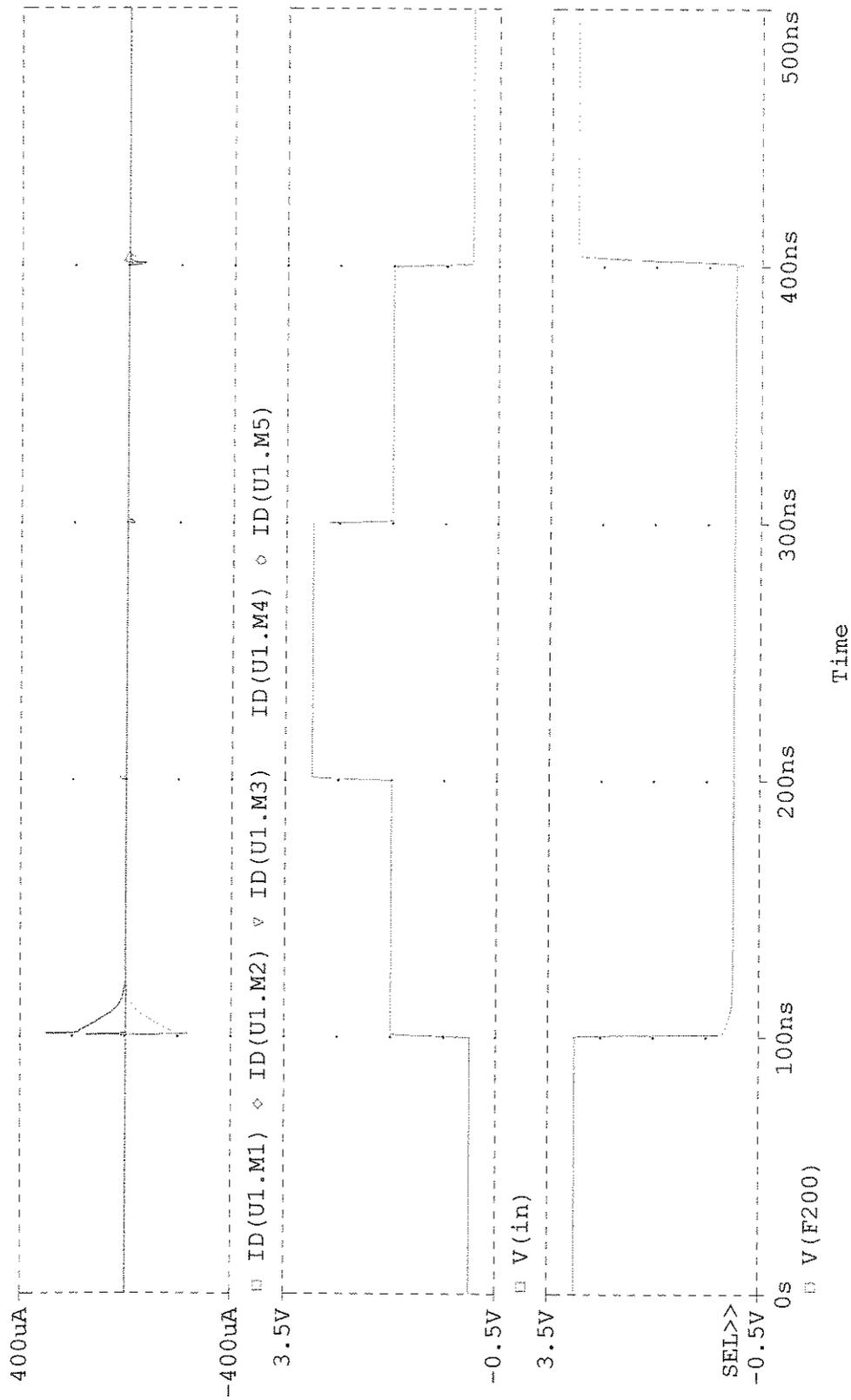


Figura 3.30: Simulação da função F(200).

FUNÇÃO F (022)

Processo Típico

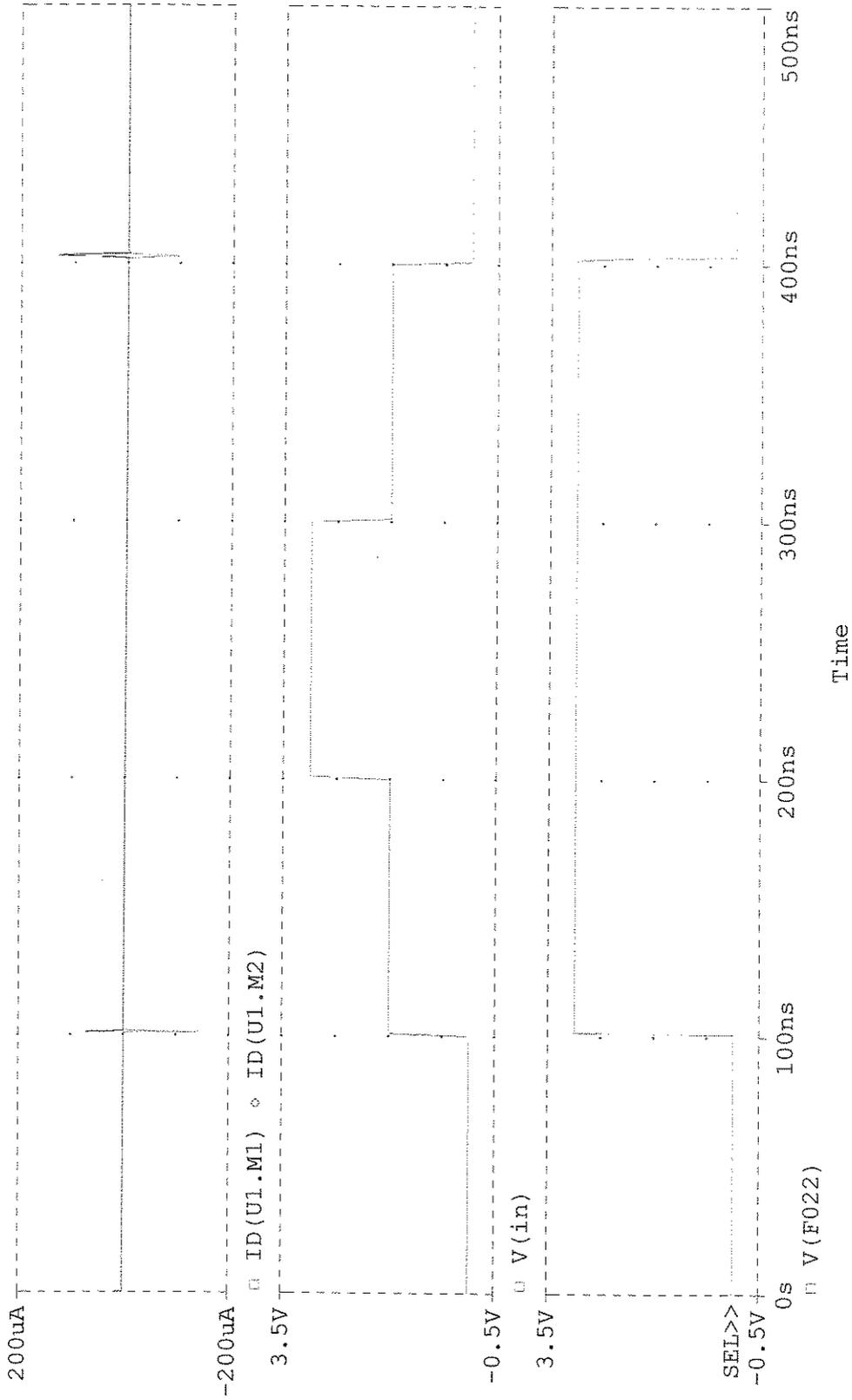


Figura 3.31: Simulação da função F (022).

FUNÇÃO F (122)

Processo Típico

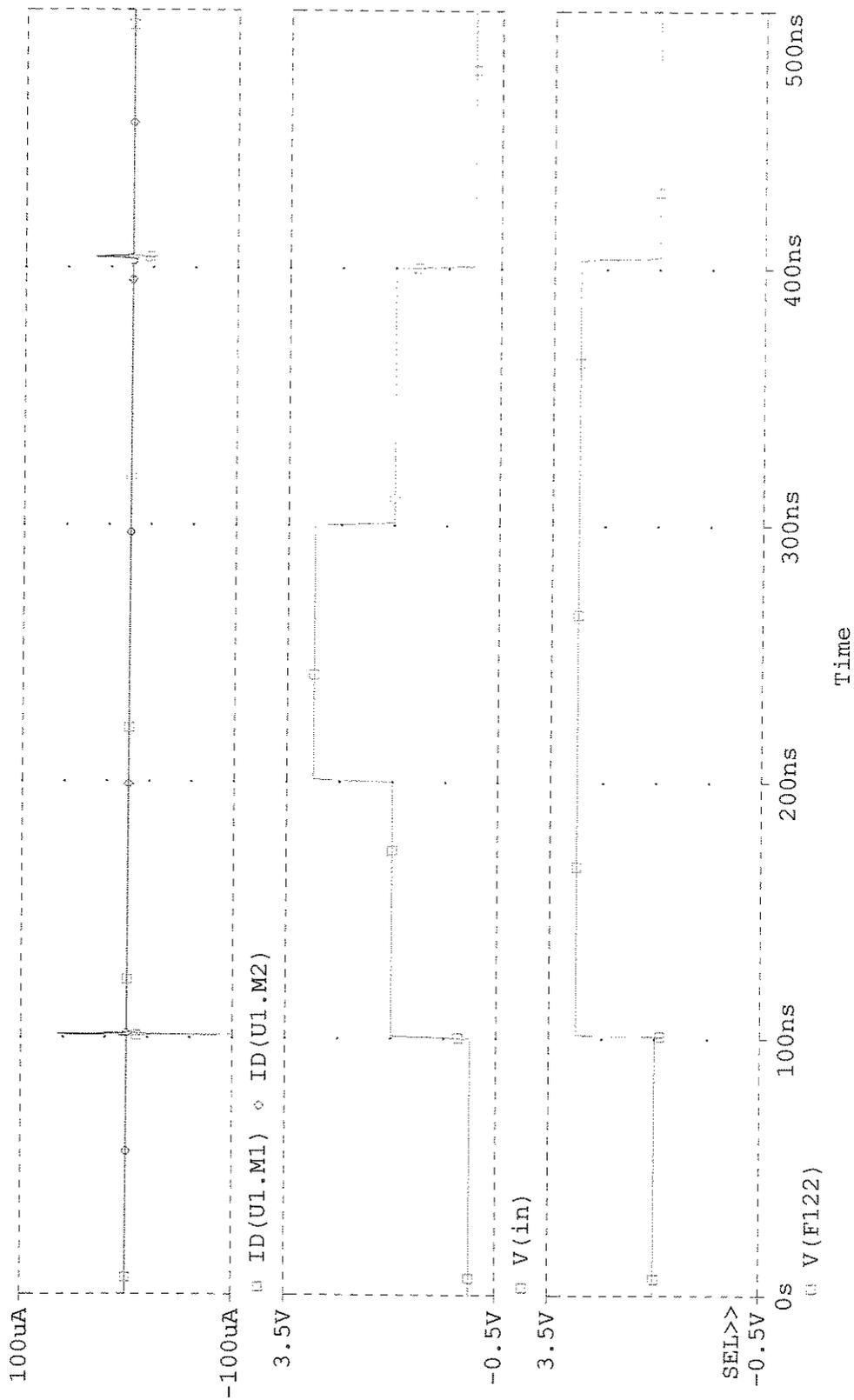


Figura 3.32: Simulação da função F(122).

FUNÇÃO F (211)

Processo Típico

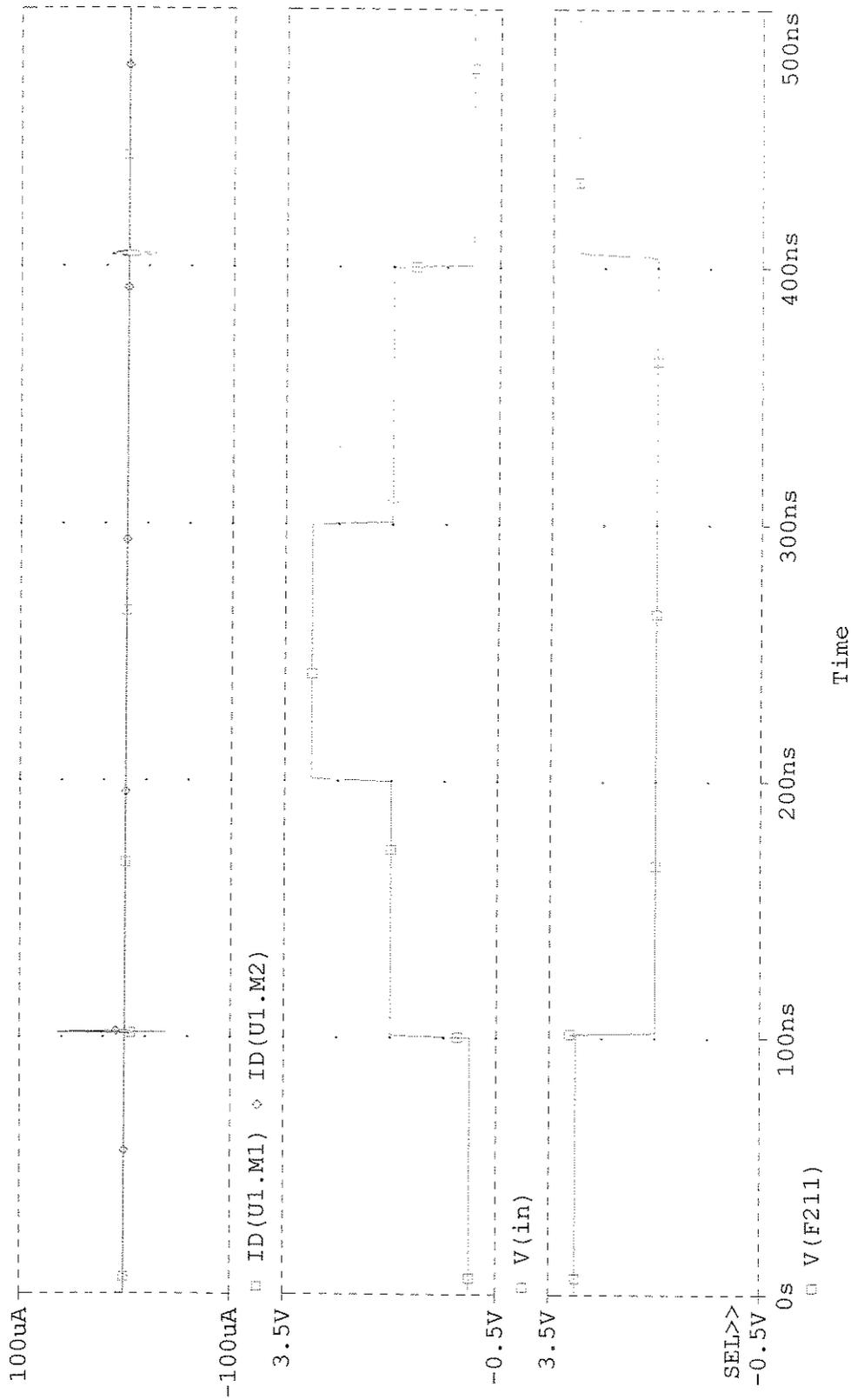


Figura 3.33: Simulação da função F (211).

FUNÇÃO F(211)

Processo Típico

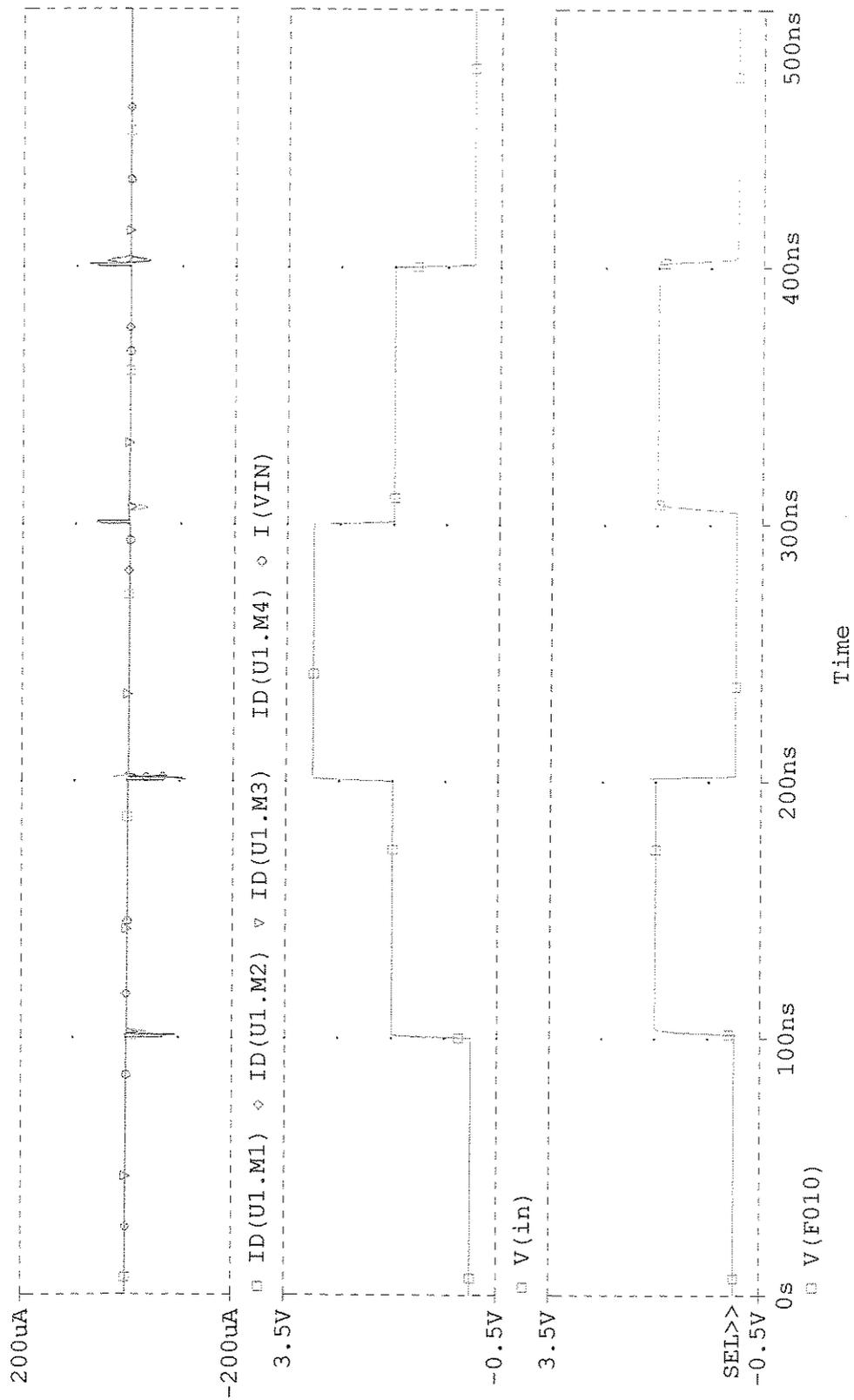


Figura 3.34: Simulação da função F(010).

FUNÇÃO F(101)

Processo Típico

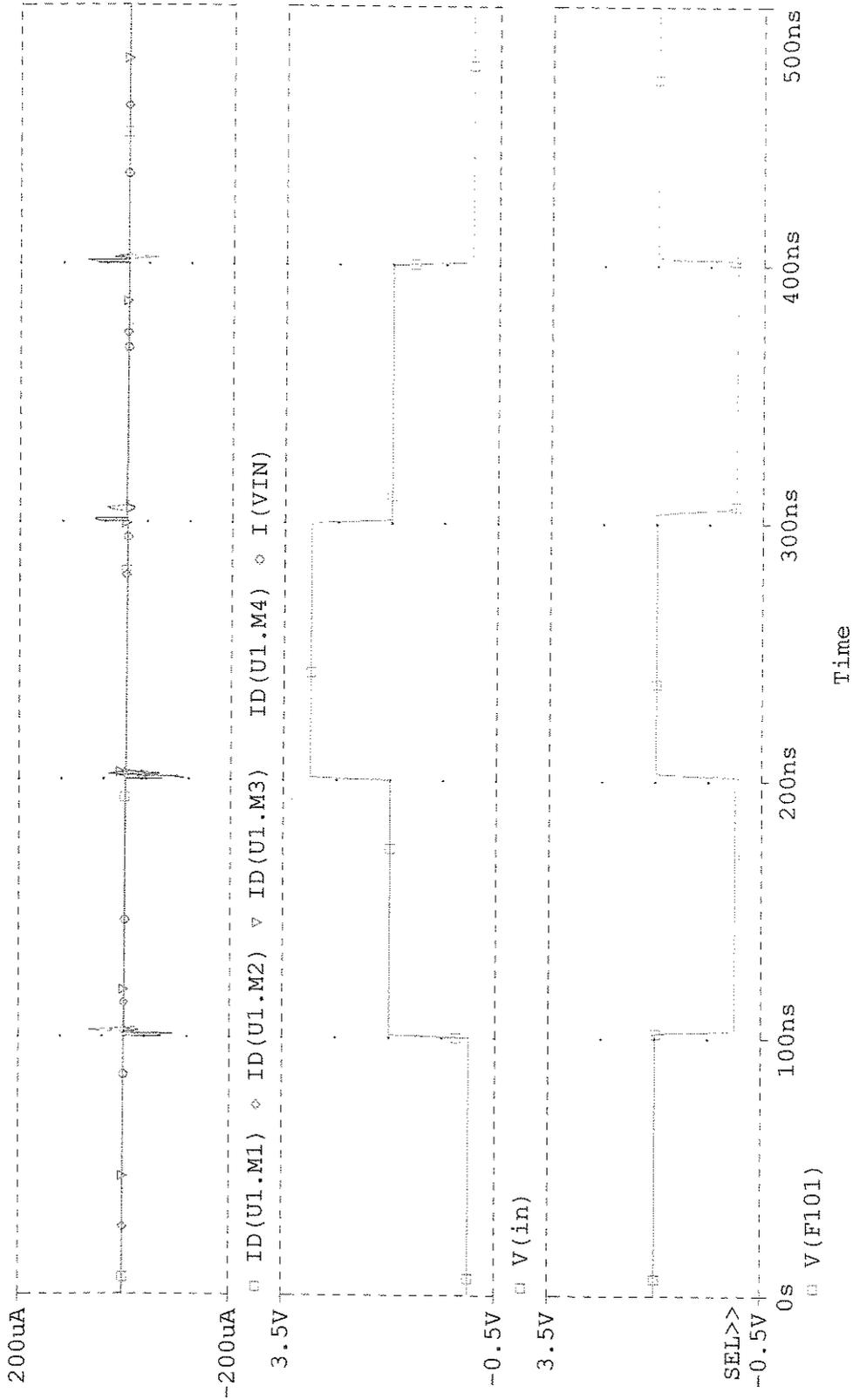


Figura 3.35: Simulação da função F(101).

FUNÇÃO F(121)

Processo Típico

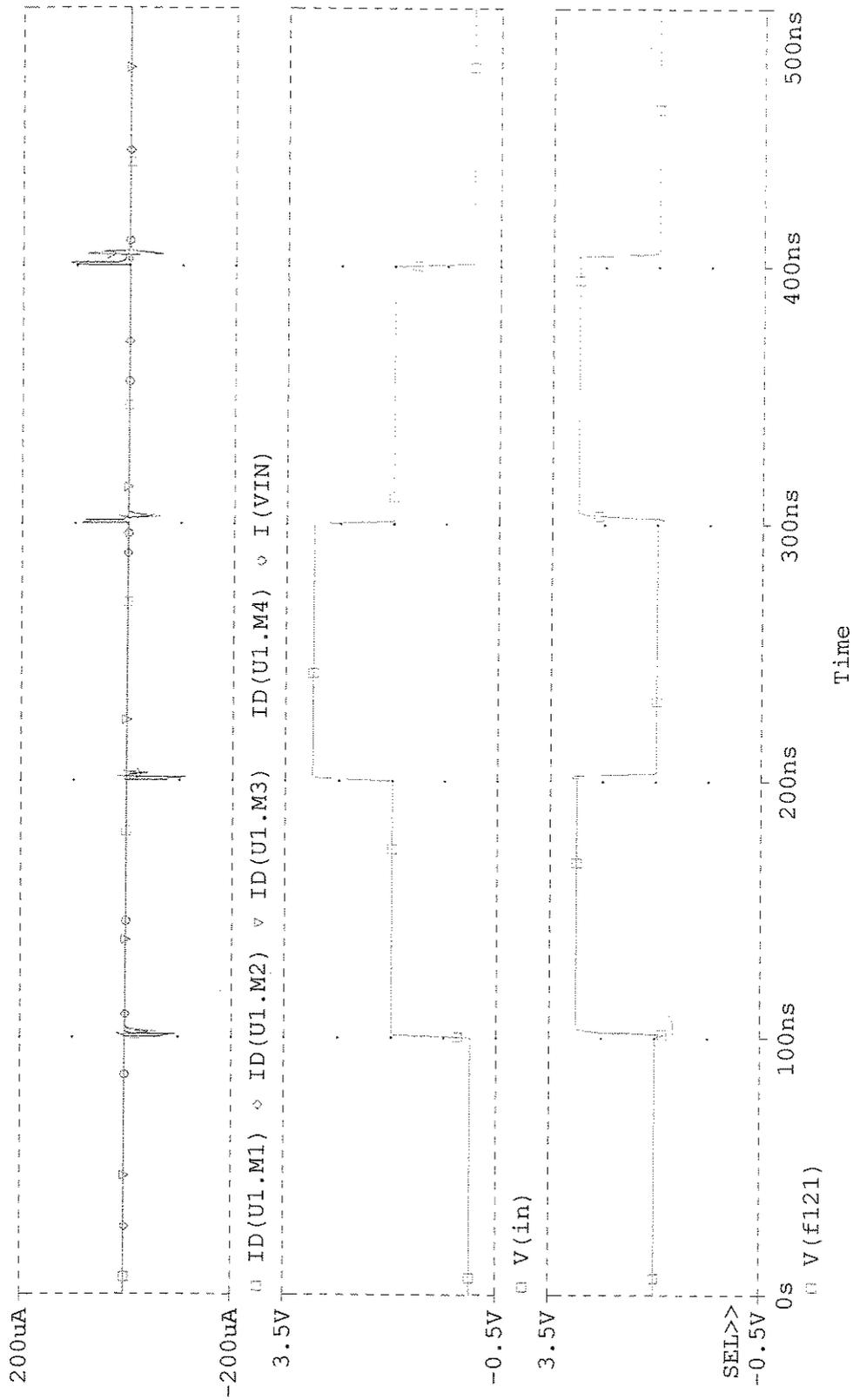
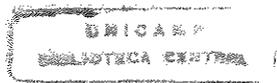


Figura 3.36: Simulação da função F(121).



FUNÇÃO F (212)

Processo Típico

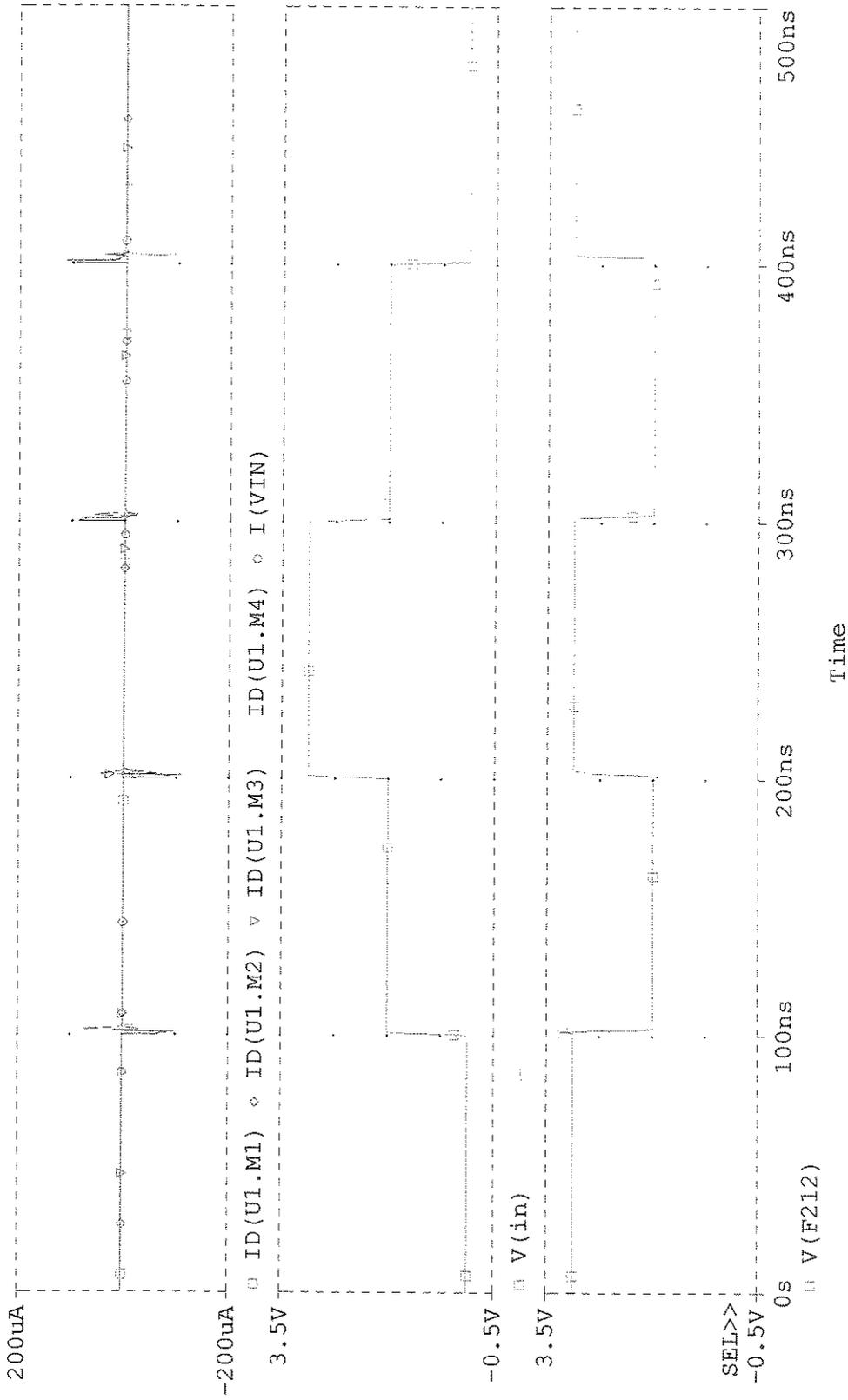


Figura 3.37: Simulação da função F(212).

FUNÇÃO F (020)

Processo Típico

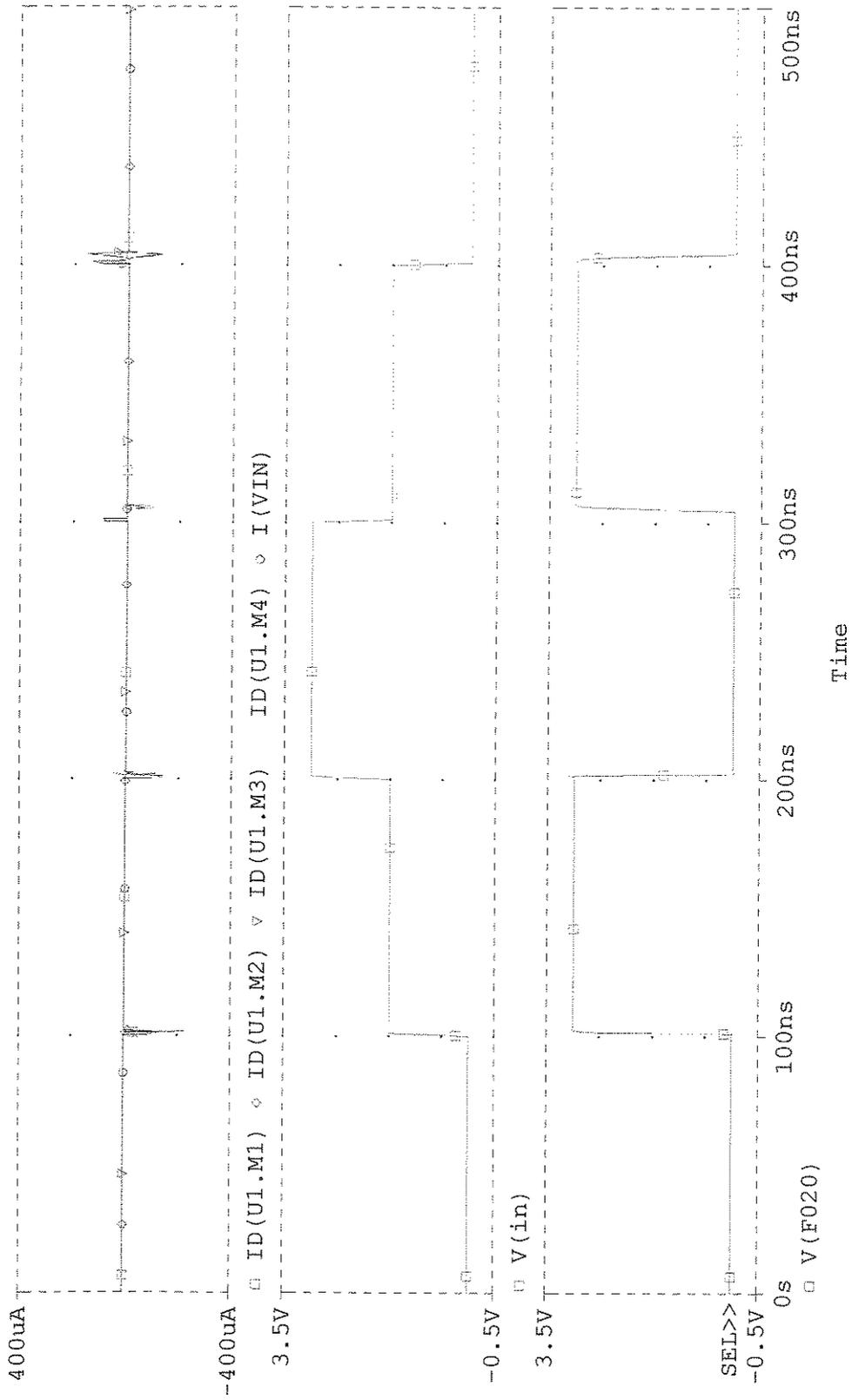


Figura 3.38: Simulação da função F(020).

FUNÇÃO F (202)

Processo Típico

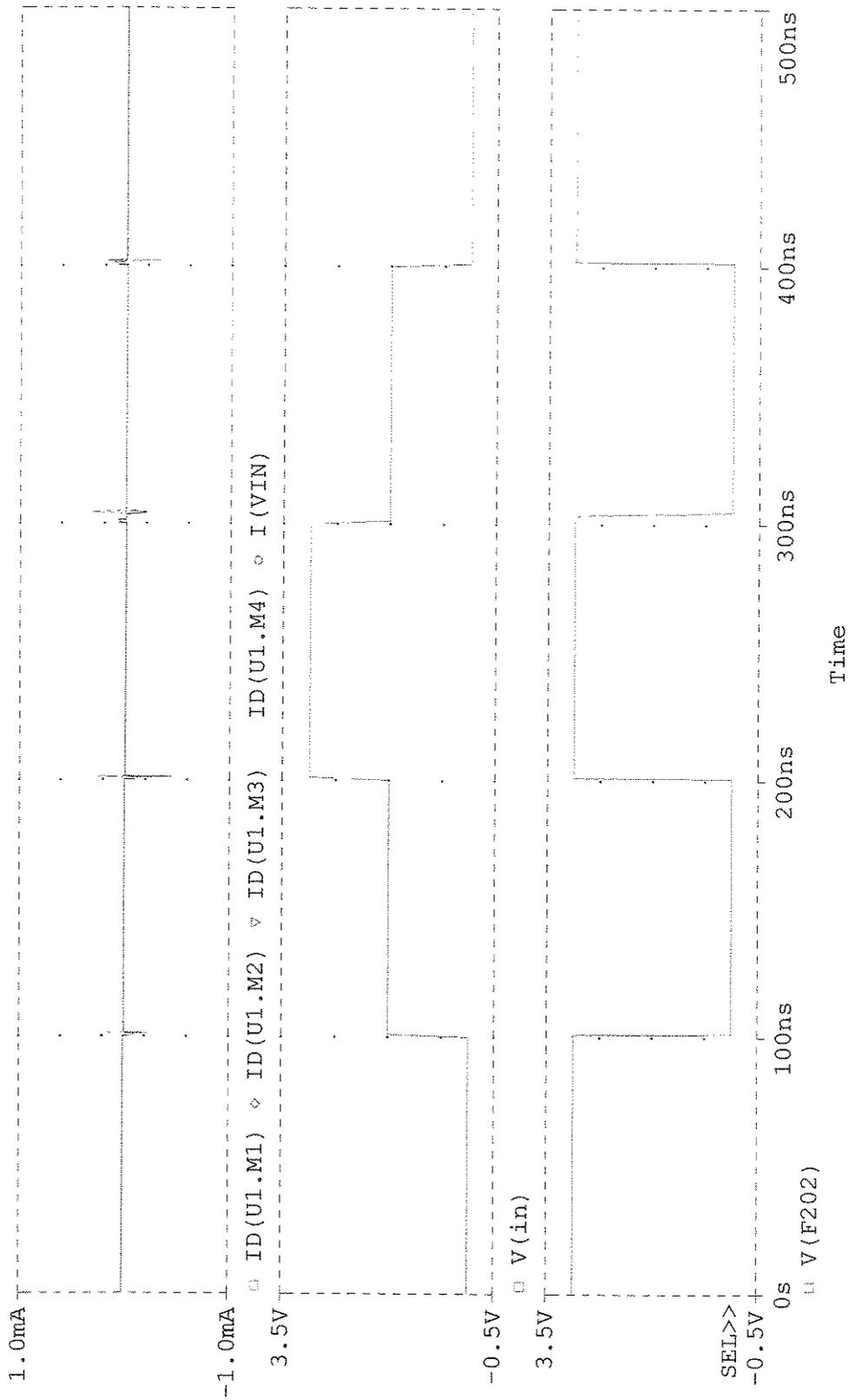


Figura 3.39: Simulação da função F (202).

FUNÇÃO F (120)

Processo Típico

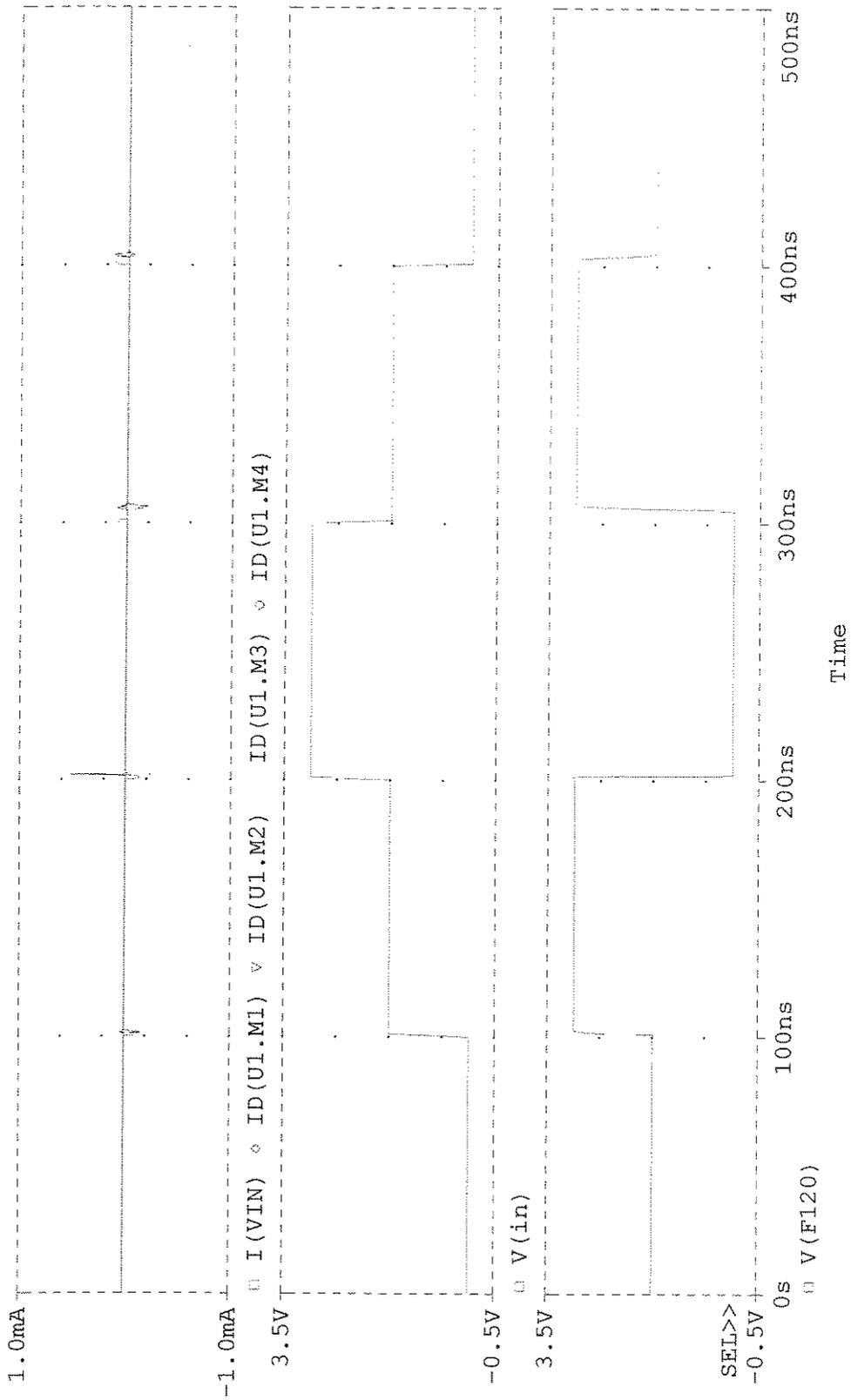


Figura 3.40: Simulação da função F(120).

FUNÇÃO F (201)

Processo Típico

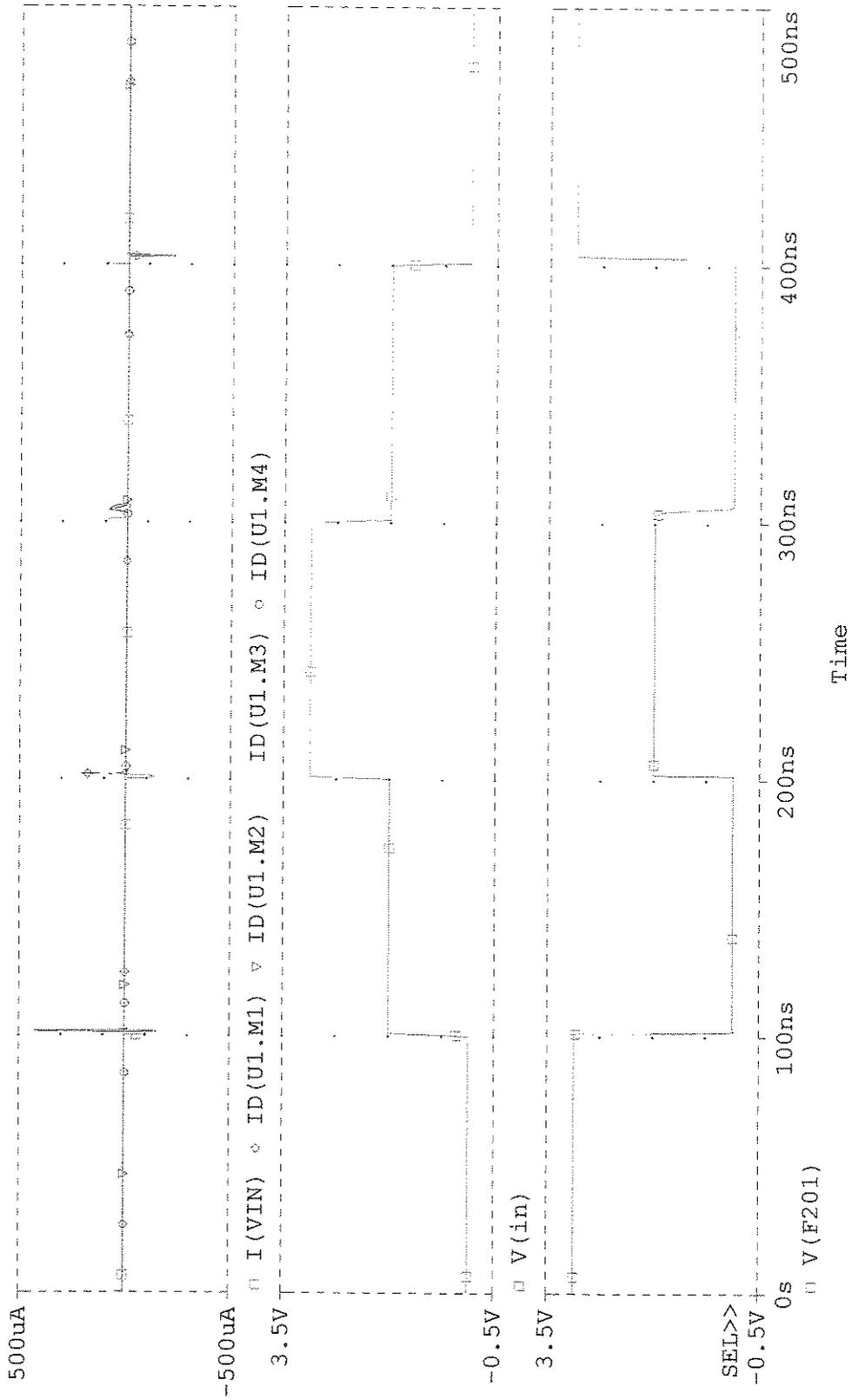


Figura 3.41: Simulação da função F(201).

FUNÇÃO F(021)

Processo Típico

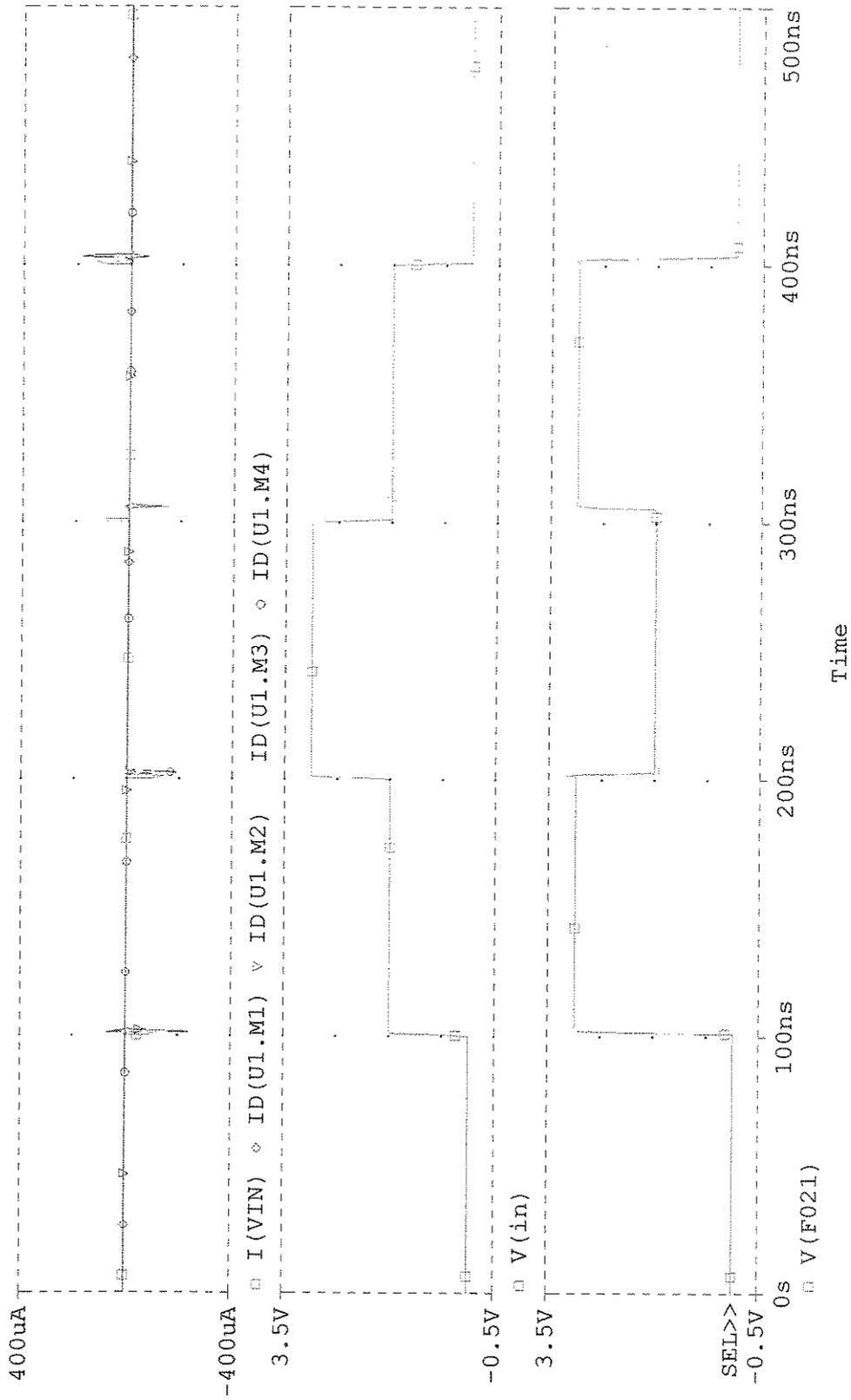


Figura 3.42: Simulação da função F(021).

FUNÇÃO F (102)

Processo Típico

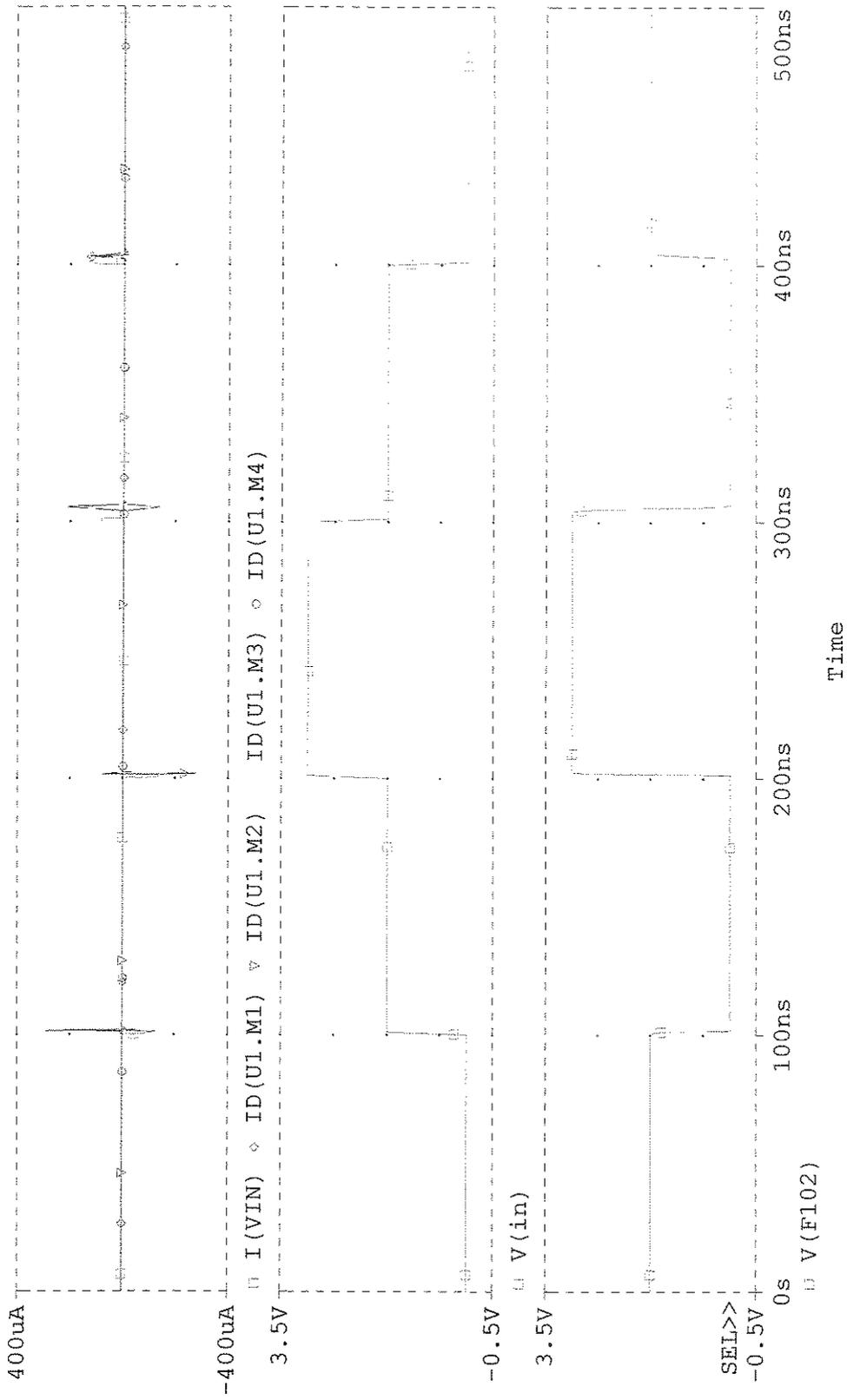


Figura 3.43: Simulação da função F(102).

FUNÇÃO F (210)

Processo Típico

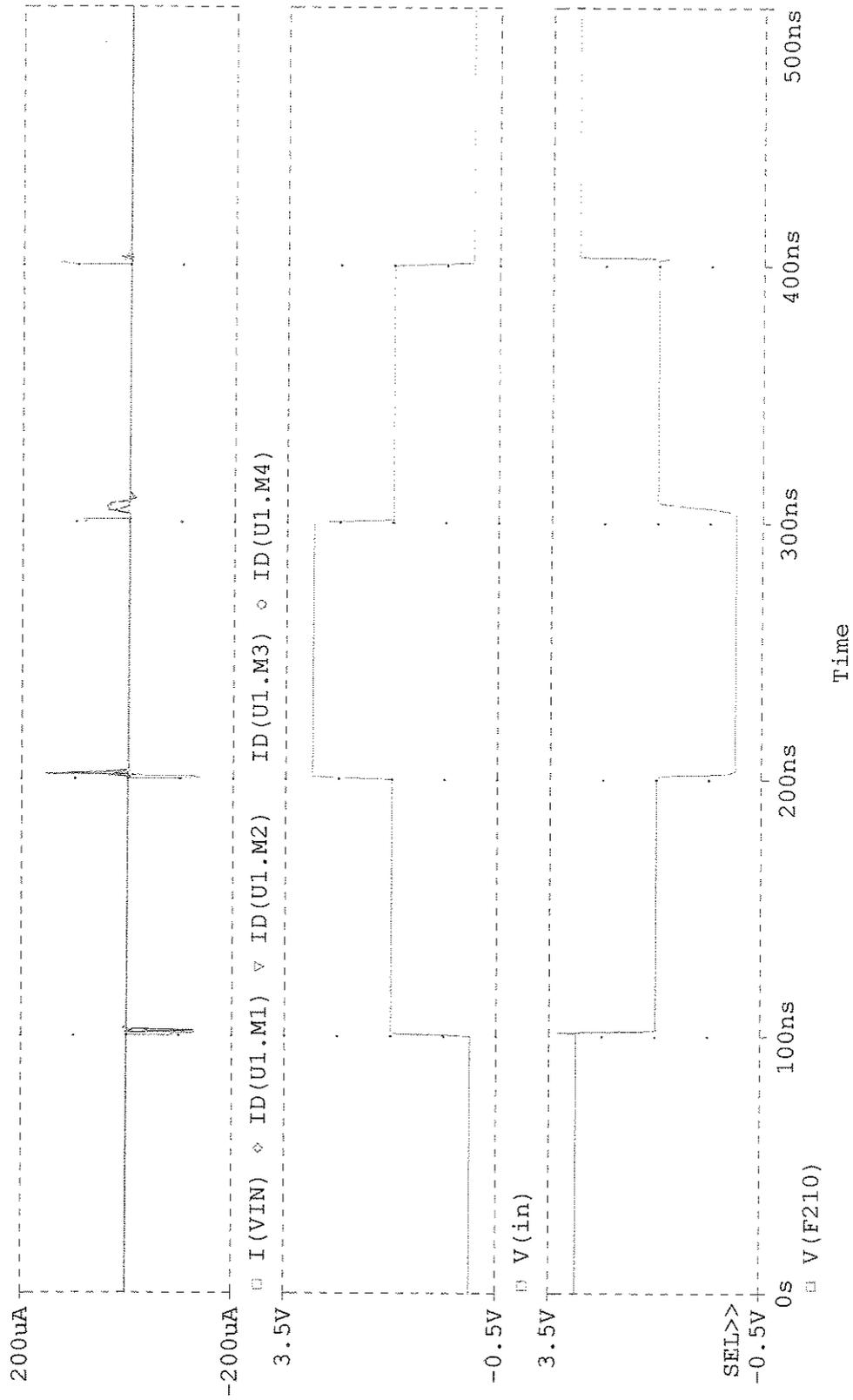


Figura 3.44: Simulação da função F(210).

FUNÇÃO F(012)

Processo Típico

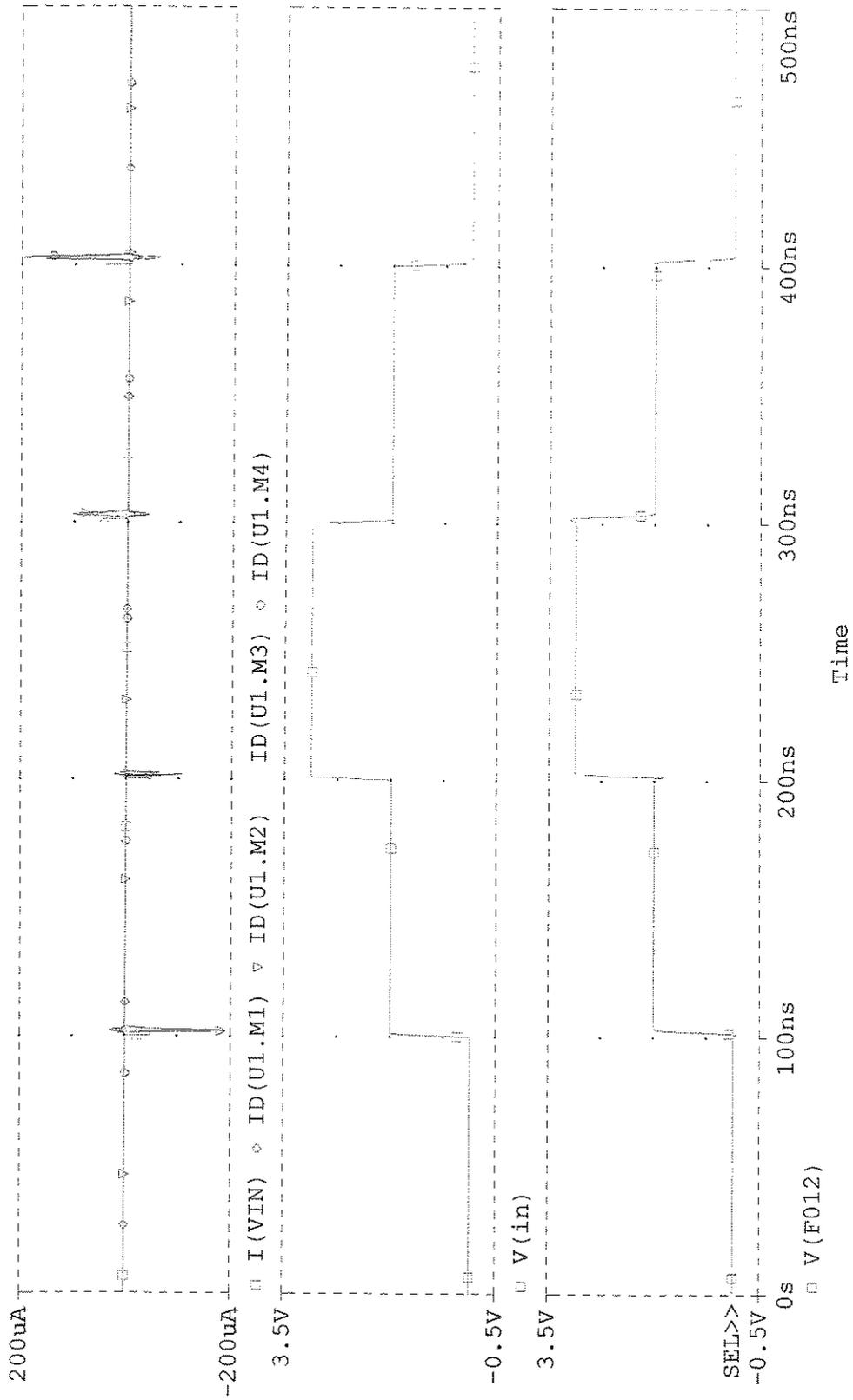


Figura 3.45: Simulação da função F(012).

3.4 – FUNÇÕES TERNÁRIAS DE DUAS OU MAIS ENTRADAS

Conforme visto no Capítulo 2, as funções ternárias com mais de uma entrada devem primeiro ser simplificadas através de um diagrama gráfico. A partir deste diagrama gráfico, obtemos as expressões lógicas para cada saída em termos de operações “E” e “OU”. Os circuitos são então implementados através de chaves devidamente interligadas em série ou em paralelo, conforme as operações sejam “E” ou “OU”, respectivamente. O controle das chaves é feito pelas funções comparadoras, cujos projetos internos foram descritos nas Seções 3.2 e 3.3.

Para a obtenção do circuito final, basta trocarmos as chaves por transistores MOS. A regra é a mesma usada para funções de uma única entrada (Seção 3.1): quando estivermos chaveando para V0, utilizaremos transistores NMOS e quando estivermos chaveando para V2 utilizaremos transistores PMOS. Desta forma, quando o transistor estiver em condução, garantimos a queda da tensão de limiar entre a “porta” e a “fonte” do transistor e evitaremos queda de tensão entre o “dreno” e a “fonte”. No caso do chaveamento para V1 ambos os transistores (tipo N ou tipo P) podem ser usados. Dependendo do transistor ser tipo N ou tipo P, escolheremos a função comparadora, que controlará o seu chaveamento, como ativa em nível alto ou baixo, respectivamente (Equações 3.1a e 3.1b). O projeto interno destas funções foi apresentado nas Seções 3.2 e 3.2.

Como exemplo usaremos a função S, simplificada no Capítulo 2 (Figuras 2.2-2.5), onde a função “S” é dada por:

$$S=1 \text{ para } (A \neq 0 \text{ e } B \neq 2);$$

$$S=2 \text{ para } (A \neq 2 \text{ e } B=2);$$

$$S=0 \text{ caso } S \neq 1 \text{ e } S \neq 2 \text{ (estado padrão)}$$

Por facilidade, reproduzimos na Figura 3.46 o circuito com chaves apresentado na Figura 2.5. Na figura 3.47 temos o projeto final, onde as chaves foram substituídas por transistores. Os resultados da simulação Spice são mostrados nas Figuras 3.48 e 3.49.

Na Figura 3.47 podemos ver que foram utilizados transistores tipo P para o chaveamento da tensão “V1”. As mesmas funções comparadoras usadas no chaveamento de “V1” e “V2” são usadas para o chaveamento de “V0”, pois como os transistores usados para “V0” são de tipo diferente em relação aos usados para “V1” e “V2”, a inversão é intrínseca.

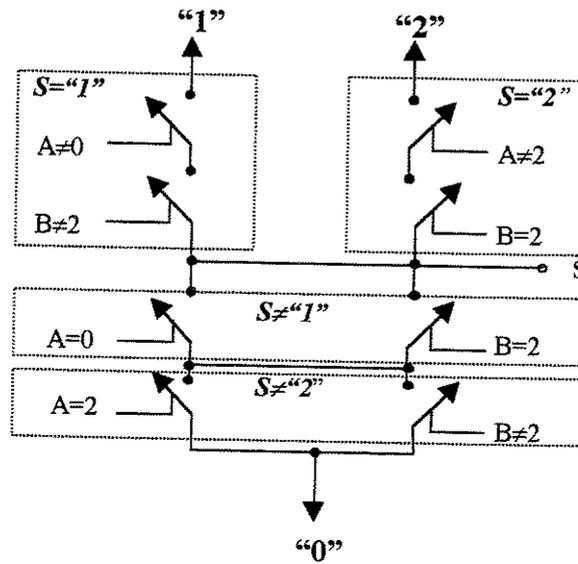


Figura 3.46: Representação interna da função S usando chaves.

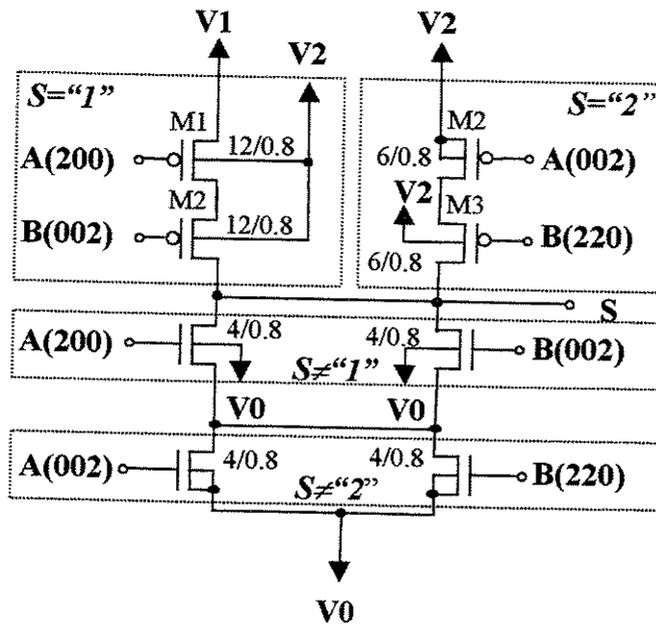


Figura 3.47: Implementação da função S com transistores.

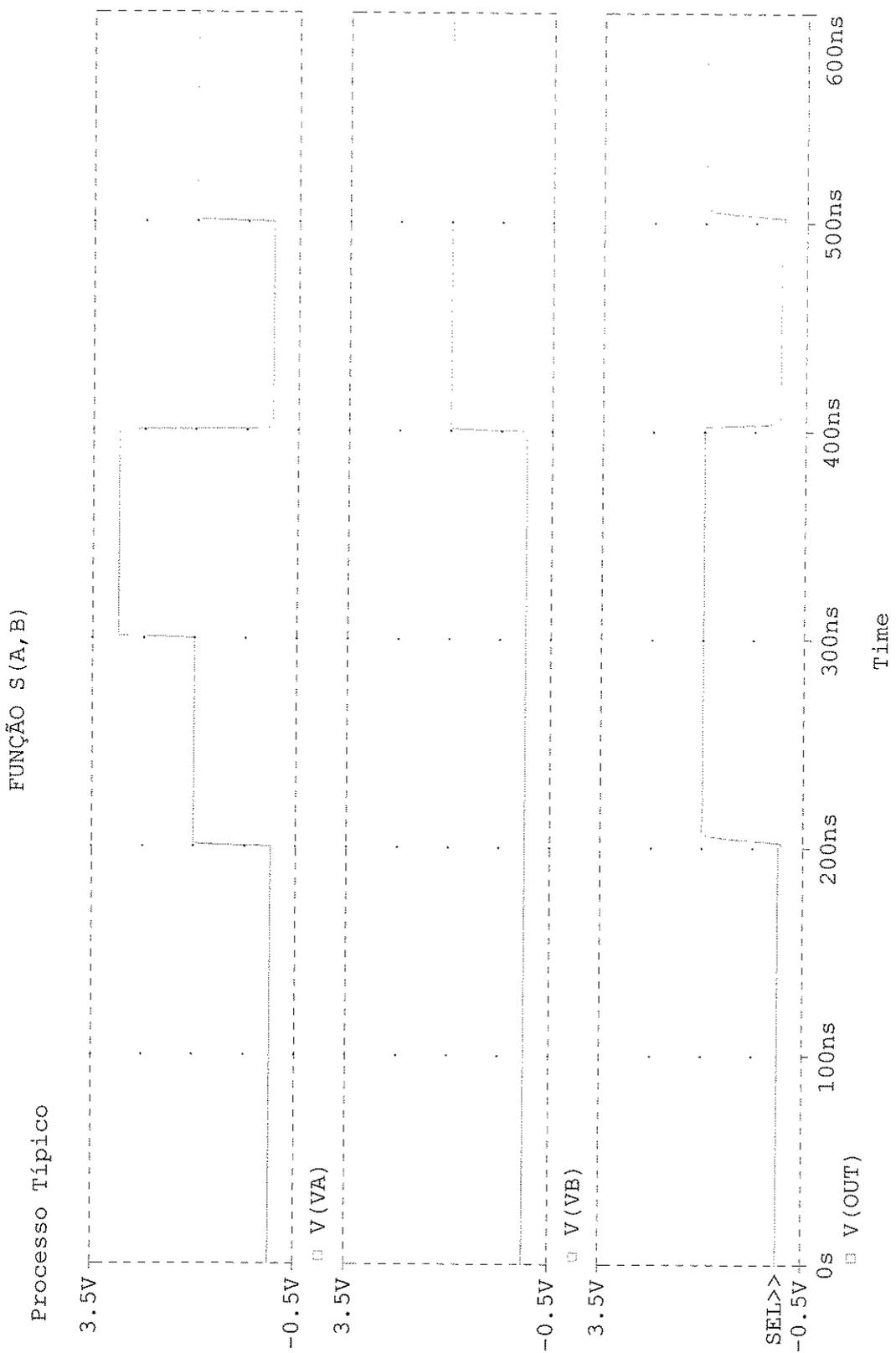


Figura 3.48: Simulação da função de 2 entradas, S(A,B).

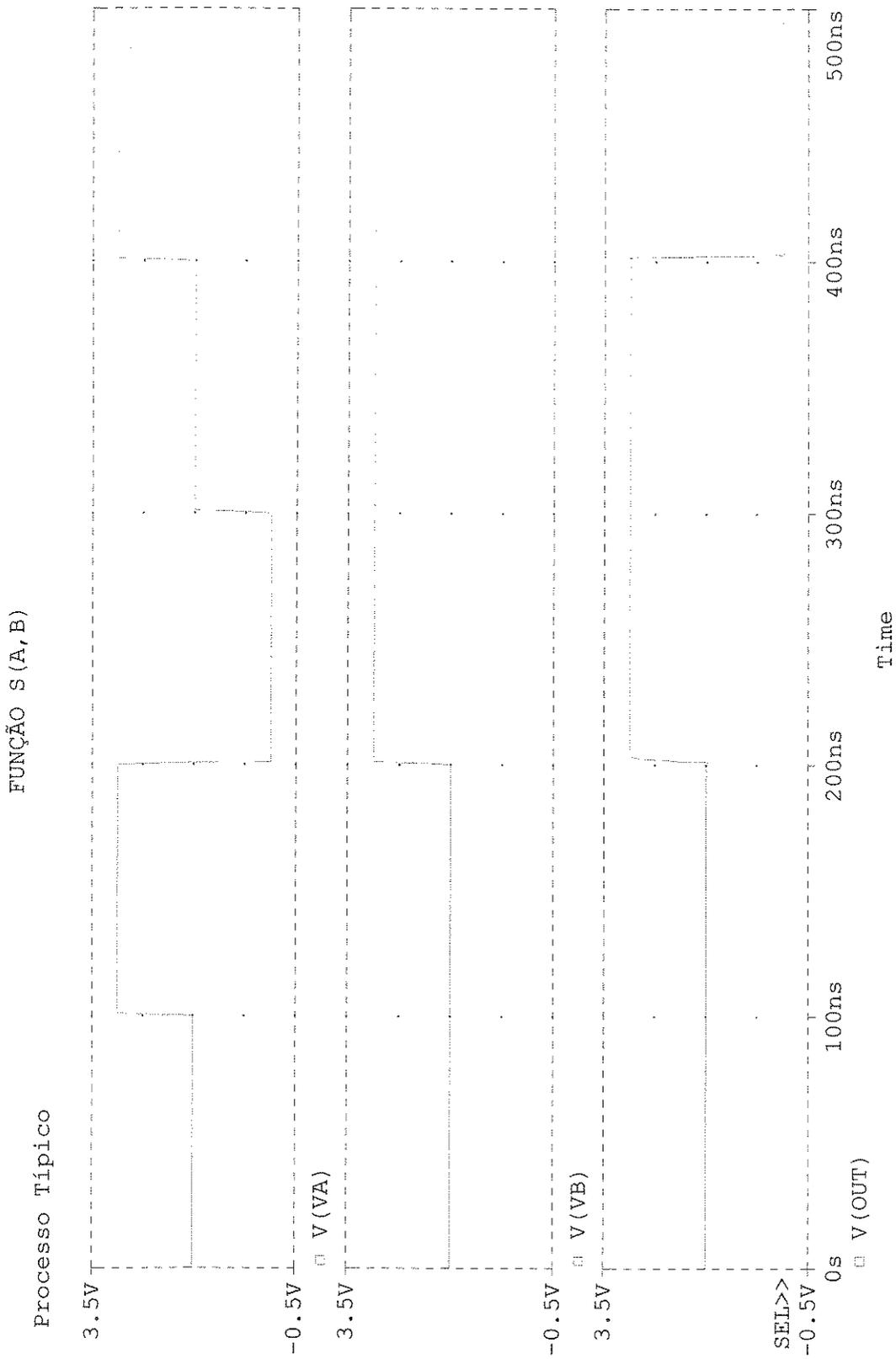


Figura 3.49: Simulação da função de 2 entradas, S(A,B) (continuação).

3.5- CONCLUSÕES

Neste Capítulo apresentamos o projeto interno das funções ternárias de uma entrada e os resultados das respectivas simulações SPICE. Projetos de funções ternárias com duas ou mais entradas também foram apresentados. Como no caso de circuitos digitais binários convencionais, os transistores operam basicamente como chaves. Os circuitos apresentam três fontes de alimentação (uma para cada nível lógico). Nas simulações foram usados os níveis “0V”, “1,5V” e “3,0V”; no entanto, os valores de tensão para os níveis lógicos “1” e “2” podem ser aumentados caso necessitemos de circuitos mais rápidos.

Os comparadores para os níveis lógicos “0” e “2”, usados para a geração de outras funções ternárias, são os únicos circuitos com consumo estático. Este consumo pode ser minimizado diminuindo-se o valor da tensão de referência “ V_{ref} ”.

A facilidade de implementação das funções torna possível o projeto dos circuitos por CAD.

CAPÍTULO 4

PROJETO DE CIRCUITOS SEQUENCIAIS

Resumo: Neste Capítulo descreveremos a implementação dos registradores (“flip-flops”), usados como elementos básicos no projeto de circuitos sequenciais. Os projetos dos registradores mais simples de serem implementados, o registrador sensível a nível (“latch”) FF(012) e o flip-flop mestre-escravo, FF(012_ms), serão descritos na Seção 4.1. Estes registradores podem ser usados tanto como elementos de memória como no projeto de contadores e máquinas de estado síncronas.

No caso de circuitos assíncronos, usaremos o “flip-flop”, FFINV, gerado através das funções inversoras, F(120) e F(201) (Seção 4.2), que possui as duas saídas: invertida e não invertida. Na Seção 4.3 apresentamos o projeto de um contador assíncrono de três algarismos que utiliza FFINV. Os resultados das simulações SPICE, juntamente com as formas de onda obtidas serão mostrados na Seção 4.4.

4.1 – REGISTRADORES NÃO INVERSORES

4.1.1 – REGISTRADOR SENSÍVEL A NÍVEL (“LATCH”)

Como foi visto no Capítulo 3, a função ternária que apresenta os três níveis lógicos de saída mais fácil de ser implementada é a função F(012). Ela é gerada a partir das funções F(100) e F(221), ambas com apenas dois transistores, não necessitando dos comparadores completos de níveis lógicos, F(220) e F(200). A função F(012) será usada como a célula básica para o projeto do “latch” não inversor FF(012).

O projeto do “latch ternário” (Fig. 4.1) é bastante semelhante ao do registrador binário sensível a nível. O circuito apresenta duas portas de transmissão que são complementares; ou

seja, quando uma está aberta, a outra está fechada (Fig. 4.2). Estas portas são controladas pelo sinal de relógio ou de habilitação, devidamente invertidos, para prover a complementariedade. Quando a porta TG1 está fechada (relógio no nível alto), a entrada está sendo amostrada através da função F(012) e a saída Q(012) será exatamente igual à entrada (após o tempo de atraso da função) e a porta TG2 está aberta. Na outra fase do relógio (nível baixo, equivalente ao “0”), a porta TG1 estará aberta e o sinal Q(012) será realimentado à entrada da função F(012) através de TG2.

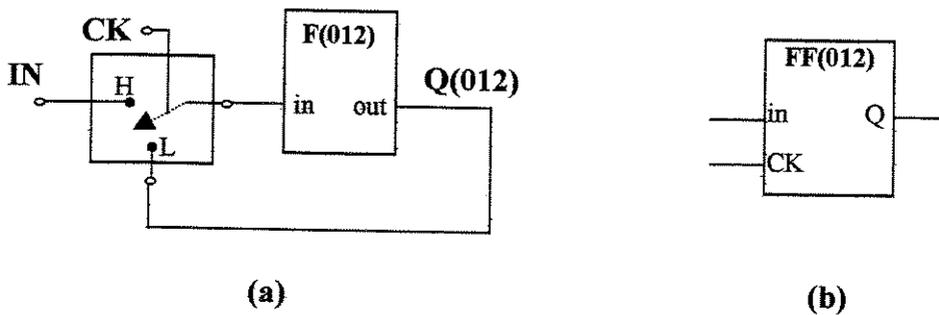


Figura 4.1: (a) Esquema do registrador sensível a nível (“latch”); (b) Símbolo.

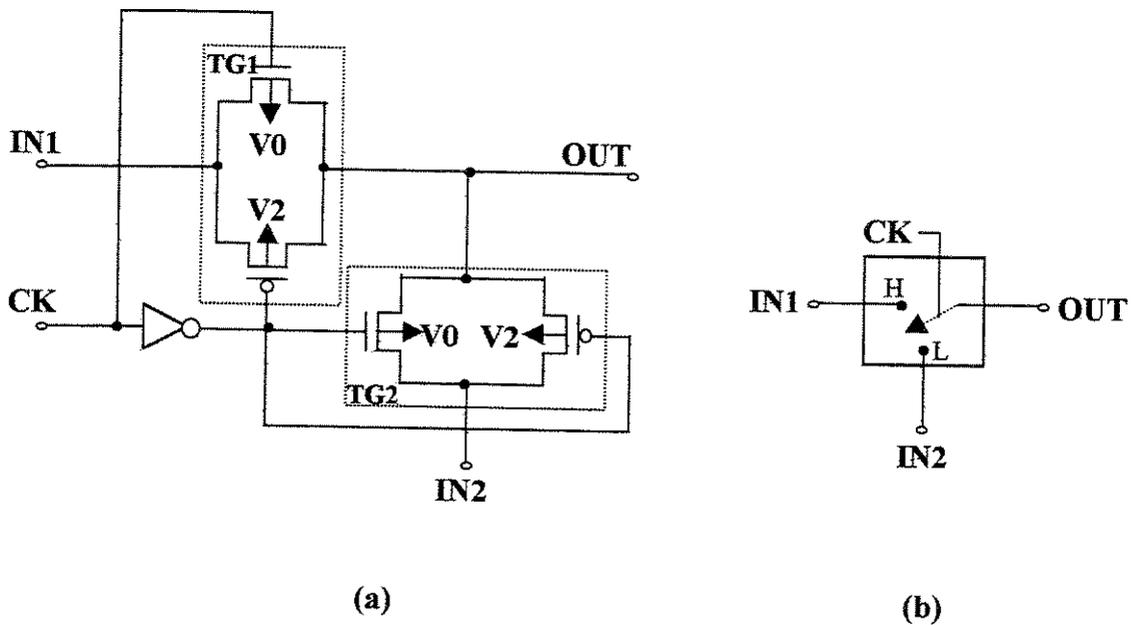


Figura 4.2:(a) Circuito interno das chaves complementares; (b) Símbolo usado nos projetos.

Na Fig. 4.3 apresentamos um esboço das formas de onda do relógio, assim como da saída Q(012), para uma dada forma de onda de entrada VIN. As formas de onda obtidas através da simulação SPICE podem ser vistas na Seção 4.4. Quando CK está alto (nível lógico “2”), a entrada VIN é amostrada em Q(012). Quando CK está baixo (nível lógico “0”), o último nível lógico amostrado através de U1 é realimentado para a entrada, e a saída é mantida constante, independentemente da entrada.

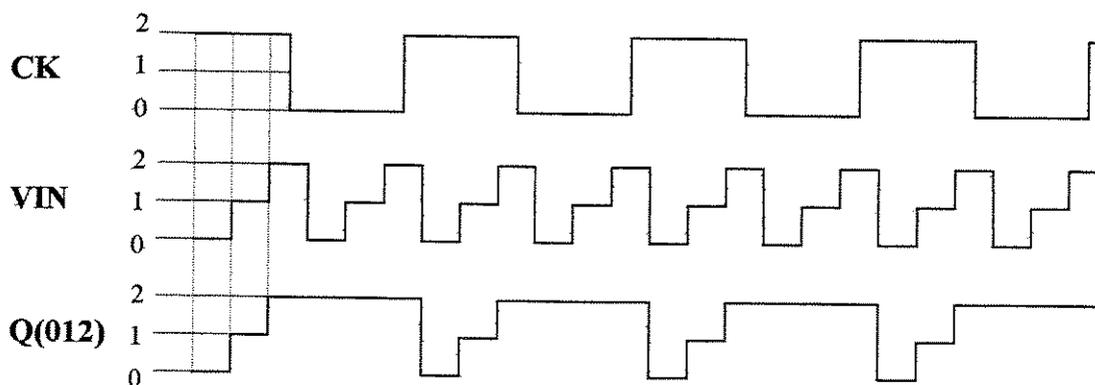


Figura 4.3: Formas de onda referentes ao circuito do “latch” (Fig 4.1), para uma dada entrada.

4.1.2- REGISTRADOR SENSÍVEL À BORDA DO RELÓGIO (“FLIP-FLOP” MESTRE-ESCRAVO)

O registrador sensível à borda do relógio (Fig. 4.4) é feito a partir de dois “latches” FF(012). O funcionamento é semelhante ao do “flip-flop” binário convencional. O primeiro “latch” é chamado de mestre e amostra o sinal de entrada (VIN) no nível baixo do relógio (CK), mantendo o sinal no nível alto do relógio. O segundo “latch” (escravo) possui a entrada de relógio invertida em relação ao primeiro. Ele amostrará o dado mantido no mestre durante o tempo que o relógio estiver alto (nível lógico “2”), e manterá a saída do escravo fixa durante a fase baixa do relógio. Deste modo, a combinação dos dois “latches” fará com que o sinal amostrado na saída do “flip-flop” seja o nível lógico presente na entrada, imediatamente antes da subida do relógio (último nível lógico amostrado). Externamente, o “flip-flop” mestre-escravo opera como sendo sensível à borda de subida do relógio.

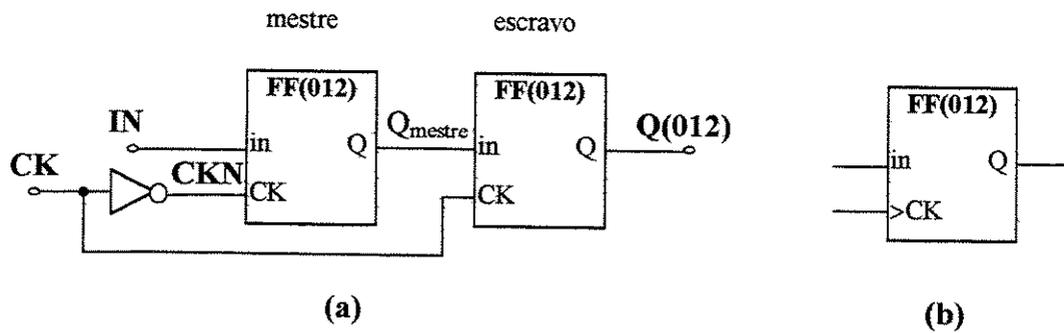


Figura 4.4: (a) Projeto do “flip-flop” mestre-escravo; (b) Símbolo

Na Fig. 4.5 apresentamos um esboço das formas de onda relativas ao “flip-flop” mestre-escravo, para uma dada forma de onda de entrada, VIN. Para auxiliar a compreensão, incluímos a forma de onda do ponto interno Q_{mestre} . O dado presente na entrada será amostrado na saída $Q(012)$ na borda de subida do sinal de relógio (o “hachurado” corresponde a um sinal desconhecido, presente antes da primeira amostragem). Para que a amostragem ocorra na borda de descida do relógio, basta invertermos as entradas CK e CKN dos “flip-flops” mestre e escravo. As formas de onda referentes à simulação SPICE do “flip-flop” encontram-se na Seção 4.4.

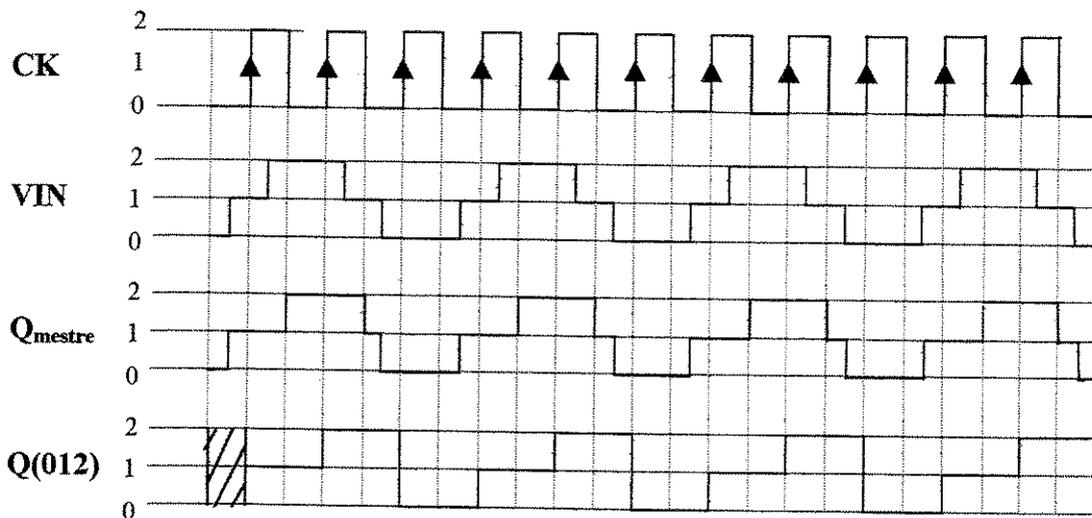


Figura 4.5: Formas de onda referentes ao “flip-flop” mestre-escravo para uma dada entrada.

4.2 –REGISTRADORES INVERSORES

4.2.1 – REGISTRADORES SENSÍVEIS AO NÍVEL, FF(120) E FF(201)

Os registradores gerados a partir das funções F(120) e F(201), apresentam duas saídas: uma “invertida” (inversão cíclica horária e anti-horária, respectivamente), e outra não-invertida (ou melhor, “desinvertida”). Esta saída invertida pode ser usada no projeto de contadores assíncronos, que serão descritos na próxima Seção.

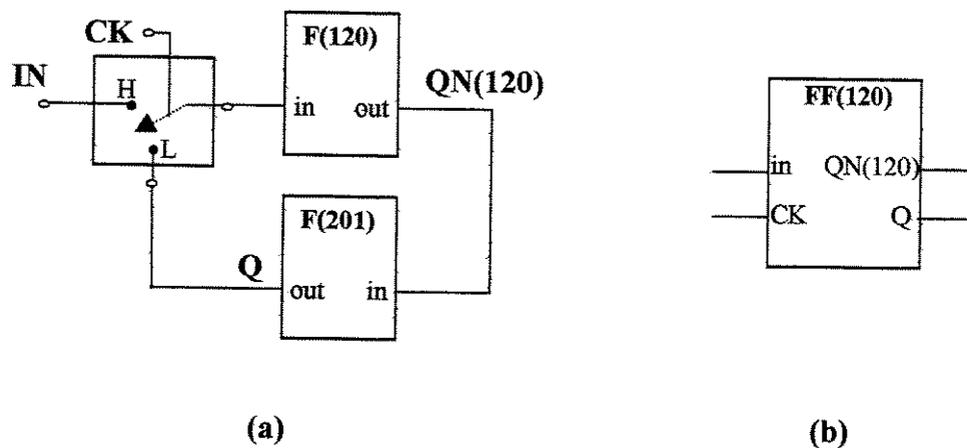


Figura 4.6: (a) Esquema do registrador sensível a nível, FF(120); (b) Símbolo.

O funcionamento dos registradores inversores é muito semelhante ao do “latch” não inversor descrito na Seção 4.1.1. A diferença é que a entrada invertida necessita ser “desinvertida” antes de ser realimentada, para o funcionamento correto do registrador. No caso do registrador FF(120), a “desinversão” é feita pela função F(201), enquanto que o registrador FF(201) utiliza a função F(120) na realimentação. O sinal de relógio controla as chaves complementares e determina a amostragem do sinal ou a manutenção da saída no último valor amostrado. Os projetos destes dois registradores são apresentados nas Figuras 4.6 e 4.7, respectivamente.

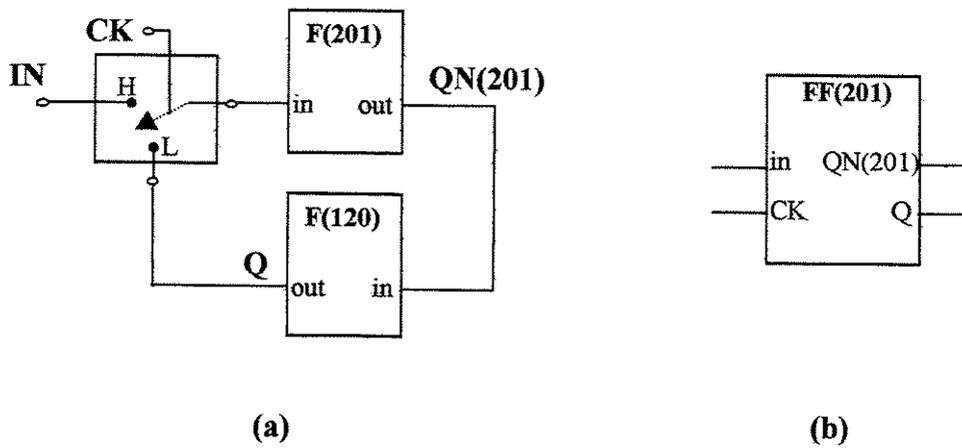


Figura 4.7: (a) Esquema do registrador sensível a nível, FF(201); (b) Símbolo.

Na Fig. 4.8 apresentamos um esboço da forma de onda nos “flip-flops” FF(120) e FF(201). A saída Q de ambos é igual. No caso, os “flip-flops” apresentados amostram a entrada durante o nível alto do relógio. Se a entrada de relógio das chaves complementares (Fig. 4.2) for invertida, os “flip-flops” amostrarão o sinal no nível baixo do relógio.

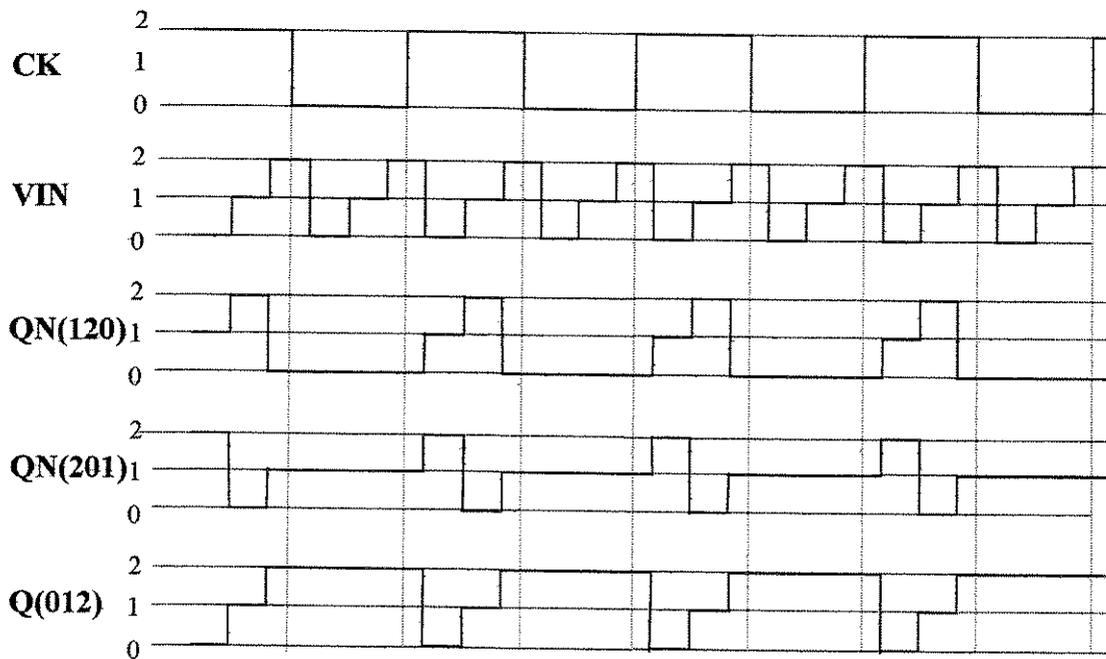


Figura 4.8: Formas de onda referentes aos “flip-flops” sensíveis a nível, FF(120) e FF(201).

4.2.2 – REGISTRADORES SENSÍVEIS À BORDA DO RELÓGIO (“FLIP-FLOP” MESTRE-ESCRAVO)

Os registradores sensíveis à borda do relógio com saída invertida, usados em contadores assíncronos, são gerados a partir de dois registradores sensíveis ao nível (FF(120) e FF(201)) em configuração mestre-escravo, semelhante à do flip-flop FF(012)_ms. O “flip-flop” FF(120)_ms utiliza o registrador FF(120) como mestre e o FF(201) como escravo, enquanto o FF(201)_ms utiliza o registrador FF(201) como mestre e o FF(120) como escravo. O funcionamento de ambos é análogo, diferenciando apenas na saída invertida (função de realimentação do escravo) disponível.

Durante o nível baixo do relógio, o registrador mestre amostra o sinal de entrada, enquanto o registrador escravo mantém a saída fixa. Na fase alta do relógio, o registrador mestre mantém fixa a sua saída no último valor amostrado antes da subida do relógio, e o registrador escravo habilita a amostragem do sinal armazenado no mestre. Os circuitos são apresentados nas Figuras 4.9 e 4.10.

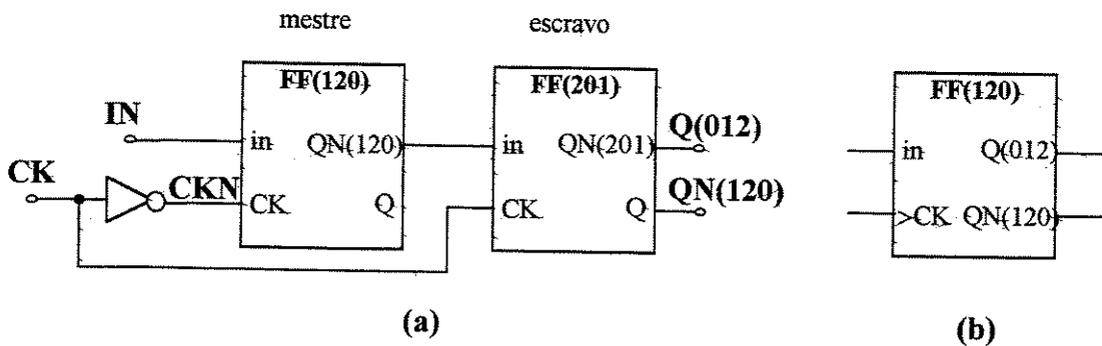


Figura 4.9: (a) Registrador sensível à borda do relógio, FF(120)_ms; (b) Símbolo.

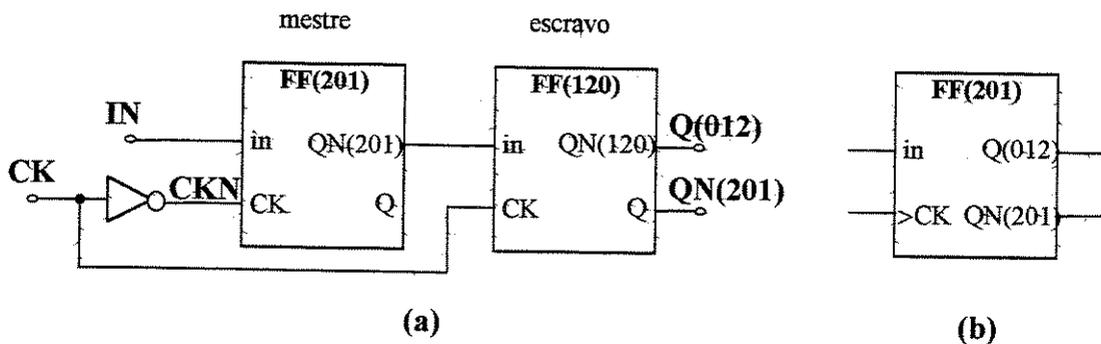


Figura 4.10: (a) Registrador sensível à borda do relógio, FF(201)_ms; (b) Símbolo.

Na Fig. 4.11 apresentamos um esboço das formas de onda relativas aos registradores sensíveis à borda de subida do relógio, para uma dada entrada VIN. As saídas Q(012) de ambos registradores são iguais, pois o sinal de entrada foi invertido e “desinvertido” (inversão cíclica) em ambos os casos. Os sinais QN(120) e QN(201) são relativos aos “flip-flops” FF(120)_ms e FF(201)_ms.

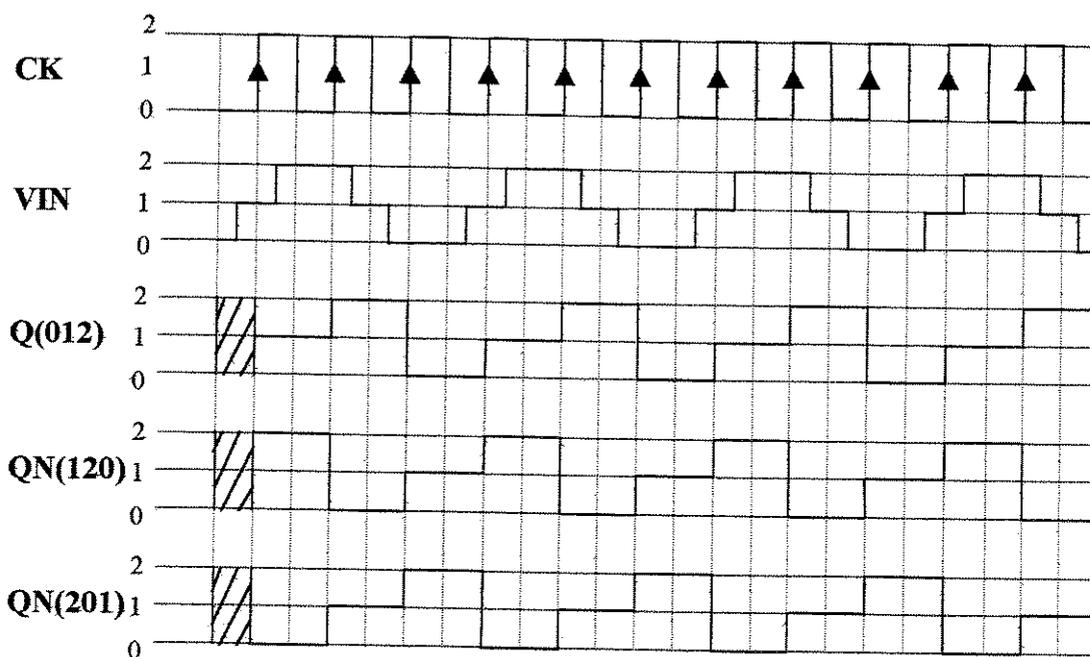


Figura 4.11: Formas de onda referentes aos “flip-flops” mestre-escravo, FF(120)_ms e FF(201)_ms.

4.3 – CONTADORES TERNÁRIOS

Contadores assíncronos crescentes e decrescentes podem ser facilmente projetados através da conexão em série dos “flip-flops” FF(120)_ms ou FF(201)_ms. A saída invertida do registrador é realimentada para a sua entrada, fazendo que a cada pulso do relógio (borda de descida ou de subida) a saída inverta (inversão cíclica) o seu nível lógico. A saída não invertida, Q(012), é usada para gerar o relógio do próximo registrador. Para termos uma contagem ternária, necessitamos de um circuito detetor de nível lógico (no caso, F220, detetor

do nível lógico “2”), que gerará um pulso de relógio a cada três variações da saída do registrador anterior (relativo ao algarismo menos significativo).

Na Fig. 4.12 apresentamos um contador ternário assíncrono de três estágios, gerado a partir de “flip-flops” FF(120)_ms. O número de algarismos pode ser aumentado indefinidamente através da conexão de mais registradores em série. Contudo, devemos levar em consideração o atraso para obtenção do algarismo mais significativo, que também aumentará de acordo com o número de registradores em série.

Na Fig. 4.13 temos o esboço das formas de onda relativas ao contador ternário apresentado na Fig. 4.12. O contador foi simulado a partir da condição inicial: $Q_0 = Q_1 = Q_2 = “0”$. Qualquer outra condição inicial levará à mesma seqüência cíclica de estados.

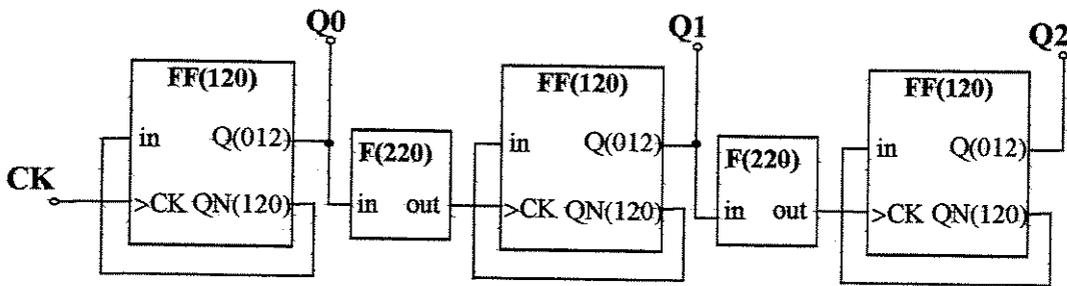


Figura 4.12: Contador ternário assíncrono de 3 estágios.

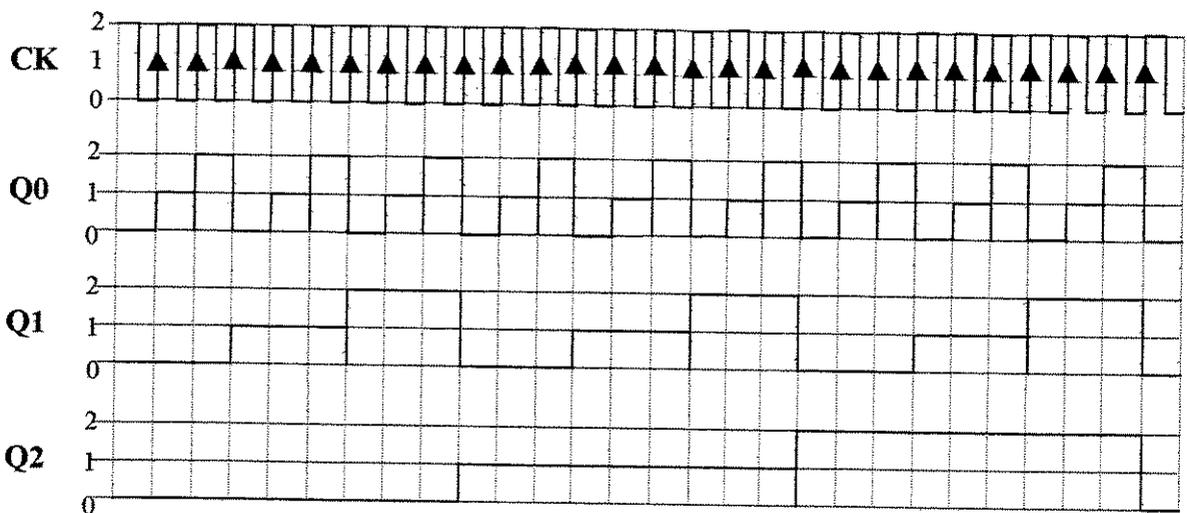


Figura 4.13: Esboço das formas de onda de saída do contador ternário assíncrono.

Na Fig. 4.13, podemos perceber um pequeno ruído lógico (“spike”) na mudança da saída de “1” para “2”. Este “spike” é inerente ao sistema ternário, pois a saída é obtida a partir da inversão cíclica horária, F(201), da função F(120). Na mudança do nível lógico “1” para o “2”, F(120) mudará de “2” para “0”, passando necessariamente pelo estado “1”. Este nível “1” temporário causará o “spike” na saída da função F(201). Se para um dado projeto, este “spike” for inconveniente, a saída deverá ser amostrada quando estável. O resultado da simulação SPICE do contador encontra-se no final da Seção 4.4.

4.4 – RESULTADOS DAS SIMULAÇÕES SPICE

Os registradores apresentados nas Seções 4.1 e 4.2, assim como o contador assíncrono ternário mostrado na Seção 4.3, foram simulados para as três condições de processo fornecidas pelo fabricante (típico, lento e rápido). Os resultados foram resumidos na Tabela 4.1. No final da Seção encontram-se as formas de onda obtidas na condição “típica” para cada circuito.

Como no caso das funções ternárias, os valores de área e perímetro de dreno e fonte dos transistores foram obtidos a partir do “layout” do circuito. Como cargas de saída dos registradores foram usados inversores simples (como da função F(100)) com relação W/L de 4.0/0.8 para o transistor PMOS e de 2.0/0.8 para o transistor NMOS.

Para os registradores sensíveis a nível, FF(012), FF(120) e FF(201), a Tabela 4.1 indica o valor de dois tipos de atraso. O atraso da saída em relação à entrada de dados, quando o relógio está habilitado, é denominado t_D , enquanto o atraso da saída em relação à borda do relógio é chamado de t_{CK} . Os “latches” FF(120) e FF(201) possuem a saída Q normal e a saída invertida, QN120 e QN201, respectivamente. Para os “latches”, FF(120) e FF(201), o atraso para a obtenção da saída QN é aproximadamente a metade do atraso para a obtenção da saída Q, pois esta é gerada a partir da “desinversão” da saída invertida.

No caso do contador, como os “flip-flops” são ligados em série, o maior atraso corresponde à obtenção do algarismo mais significativo (Q2). Os atrasos referentes a esta saída também são

apresentados no final da Tabela 4.1. De acordo com a simulação Spice, a frequência máxima do relógio é de 26MHz para o processo lento e de 100MHz para o processo típico.

O valor do consumo não foi indicado na Tabela 4.1 pois depende dos dados amostrados e armazenados. O consumo estático deve-se exclusivamente às funções comparadores completos (F220) e F(200), como descrito no Capítulo 3. As funções FF(012) e FF(012)_ms não utilizam os comparadores completos e possuem consumo estático praticamente nulo (como no caso dos circuitos CMOS binários convencionais).

FUNÇÃO/ SAÍDA	ATRASOS	TIPOS DE PROCESSO		
		TÍPICO	LENTO	RÁPIDO
FF(012)/ SAÍDA Q	tD _{0→1}	1,7ns	4,0ns	0,9ns
	tD _{0→2}	2,0ns	9,9ns	0,9ns
	tD _{1→0}	2,3ns	5,0ns	1,2ns
	tD _{1→2}	1,8ns	6,8ns	0,8ns
	tD _{2→0}	2,6ns	6,3ns	1,2ns
	tD _{2→1}	2,8ns	9,1ns	1,2ns
	tCK _{0→1}	1,5ns	3,9ns	0,7ns
	tCK _{0→2}	2,2ns	10,2ns	0,9ns
	tCK _{1→0}	2,0ns	4,8ns	0,8ns
	tCK _{1→2}	1,6ns	6,8ns	0,6ns
	tCK _{2→0}	2,6ns	6,4ns	1,0ns
	tCK _{2→1}	2,7ns	9,2ns	1,0ns
FF(012)_ms/ SAÍDA Q	tCK _{0→1}	1,8ns	4,6ns	0,7ns
	tCK _{0→2}	3,1ns	12,9ns	1,2ns
	tCK _{1→0}	2,3ns	5,5ns	0,9ns
	tCK _{1→2}	2,1ns	9,1ns	0,8ns
	tCK _{2→0}	3,0ns	7,7ns	1,3ns
	tCK _{2→1}	3,2ns	11,0ns	1,2ns

Tabela 4.1: Resultados das simulações SPICE para os diversos registradores (1ª parte)

FUNÇÃO/ SAÍDA	ATRASOS/ CONSUMO	TIPOS DE PROCESSO		
		TÍPICO	LENTO	RÁPIDO
FF(120)/ SAÍDA Q	$t_{D0 \rightarrow 1}$	3,2ns	11,9ns	1,4ns
	$t_{D0 \rightarrow 2}$	4,8ns	16,0ns	2,1ns
	$t_{D1 \rightarrow 0}$	7,8ns	12,3ns	5,2ns
	$t_{D1 \rightarrow 2}$	4,9ns	15,5ns	2,2ns
	$t_{D2 \rightarrow 0}$	2,0ns	3,4ns	1,3ns
	$t_{D2 \rightarrow 1}$	6,5ns	15,6ns	5,1ns
	$t_{CK0 \rightarrow 1}$	3,1ns	11,6ns	1,2ns
	$t_{CK0 \rightarrow 2}$	5,0ns	16,0ns	2,1ns
	$t_{CK1 \rightarrow 0}$	7,6ns	12,2ns	4,9ns
	$t_{CK1 \rightarrow 2}$	4,8ns	15,7ns	2,0ns
	$t_{CK2 \rightarrow 0}$	2,0ns	3,5ns	1,3ns
	$t_{CK2 \rightarrow 1}$	8,7ns	15,9ns	5,3ns
FF(120)/ QN120	$t_{D0 \rightarrow 1}$	3,2ns	8,9ns	0,7ns
	$t_{D0 \rightarrow 2}$	6,5ns	6,5ns	4,4ns
	$t_{D1 \rightarrow 0}$	2,0ns	8,5ns	0,8ns
	$t_{D1 \rightarrow 2}$	1,4ns	2,9ns	0,8ns
	$t_{D2 \rightarrow 0}$	2,0ns	7,9ns	0,9ns
	$t_{D2 \rightarrow 1}$	3,3ns	8,4ns	1,6ns
	$t_{CK0 \rightarrow 1}$	3,2ns	8,9ns	0,7ns
	$t_{CK0 \rightarrow 2}$	6,9ns	6,8ns	4,7ns
	$t_{CK1 \rightarrow 0}$	2,2ns	8,7ns	0,9ns
	$t_{CK1 \rightarrow 2}$	1,3ns	2,9ns	0,7ns
	$t_{CK2 \rightarrow 0}$	1,9ns	8,1ns	0,7ns
	$t_{CK2 \rightarrow 1}$	3,0ns	8,3ns	1,2ns

Tabela 4.1: Resultados das simulações SPICE para os diversos registradores (continuação)

FUNÇÃO/ SAÍDA	ATRASOS/ CONSUMO	TIPOS DE PROCESSO		
		TÍPICO	LENTO	RÁPIDO
FF(201)/ SAÍDA Q	td ₀ →1	4,9ns	12,2ns	2,3ns
	td ₀ →2	6,5ns	3,7ns	4,4ns
	td ₁ →0	4,6ns	15,3ns	2,2ns
	td ₁ →2	2,9ns	10,5ns	1,3ns
	td ₂ →0	4,7ns	15,4ns	2,1ns
	td ₂ →1	8,7ns	13,7ns	5,1ns
	tck ₀ →1	4,7ns	12,1ns	1,4ns
	tck ₀ →2	6,8ns	4,7ns	4,0ns
	tck ₁ →0	4,5ns	15,0ns	1,9ns
	tck ₁ →2	2,8ns	10,8ns	1,1ns
	tck ₂ →0	4,8ns	15,4ns	2,0ns
	tck ₂ →1	8,9ns	13,9ns	5,5ns
	FF(201)/ QN201	td ₀ →1	1,9ns	8,0ns
td ₀ →2		3,3ns	8,0ns	1,6ns
td ₁ →0		5,0ns	4,6ns	3,6ns
td ₁ →2		3,3ns	8,3ns	1,6ns
td ₂ →0		1,6ns	3,3ns	0,9ns
td ₂ →1		1,9ns	9,0ns	0,7ns
tck ₀ →1		1,8ns	8,3ns	0,6ns
tck ₀ →2		3,0ns	7,7ns	1,3ns
tck ₁ →0		5,2ns	4,7ns	3,9ns
tck ₁ →2		3,3ns	8,4ns	1,4ns
tck ₂ →0		1,4ns	3,2ns	0,7ns
tck ₂ →1		2,1ns	9,3ns	0,8ns

Tabela 4.1: Resultados das simulações SPICE para os diversos registradores (continuação)

FUNÇÃO/ SAÍDA	ATRASOS/ CONSUMO	TIPOS DE PROCESSO		
		TÍPICO	LENTO	RÁPIDO
FF(120)_ms SAÍDA Q	tCK ₀ →1	2,8ns	11,7ns	0,9ns
	tCK ₀ →2	3,2ns	8,4ns	1,4ns
	tCK ₁ →0	5,4ns	6,2ns	4,0ns
	tCK ₁ →2	3,6ns	8,8ns	1,6ns
	tCK ₂ →0	2,4ns	6,5ns	1,0ns
	tCK ₂ →1	3,7ns	13,9ns	1,4ns
FF(120)_ms QN(120)	tCK ₀ →1	5,6ns	15,2ns	2,3ns
	tCK ₀ →2	2,4ns	15,9ns	4,7ns
	tCK ₁ →0	4,7ns	15,3ns	2,0ns
	tCK ₁ →2	3,8ns	14,2ns	1,4ns
	tCK ₂ →0	5,0ns	15,8ns	2,2ns
	tCK ₂ →1	9,0ns	15,1ns	5,6ns
CONTADOR SAÍDA Q0	tCK ₀ →1	2,8ns	11,7ns	0,9ns
	tCK ₁ →2	3,6ns	9,0ns	1,6ns
	tCK ₂ →0	2,5ns	6,3ns	1,1ns
CONTADOR SAÍDA Q1	tCK ₀ →1	5,9ns	19,1ns	2,7ns
	tCK ₁ →2	7,0ns	16,1ns	3,6ns
	tCK ₂ →0	5,7ns	13,6ns	3,0ns
CONTADOR SAÍDA Q2	tCK ₀ →1	9,1ns	26,4	4,7ns
	tCK ₁ →2	10,1ns	23,2ns	5,4ns
	tCK ₂ →0	8,9ns	20,6ns	5,0ns

Tabela 4.1: Resultados das simulações SPICE para os diversos registradores (continuação)

FF(012)

Processo Típico

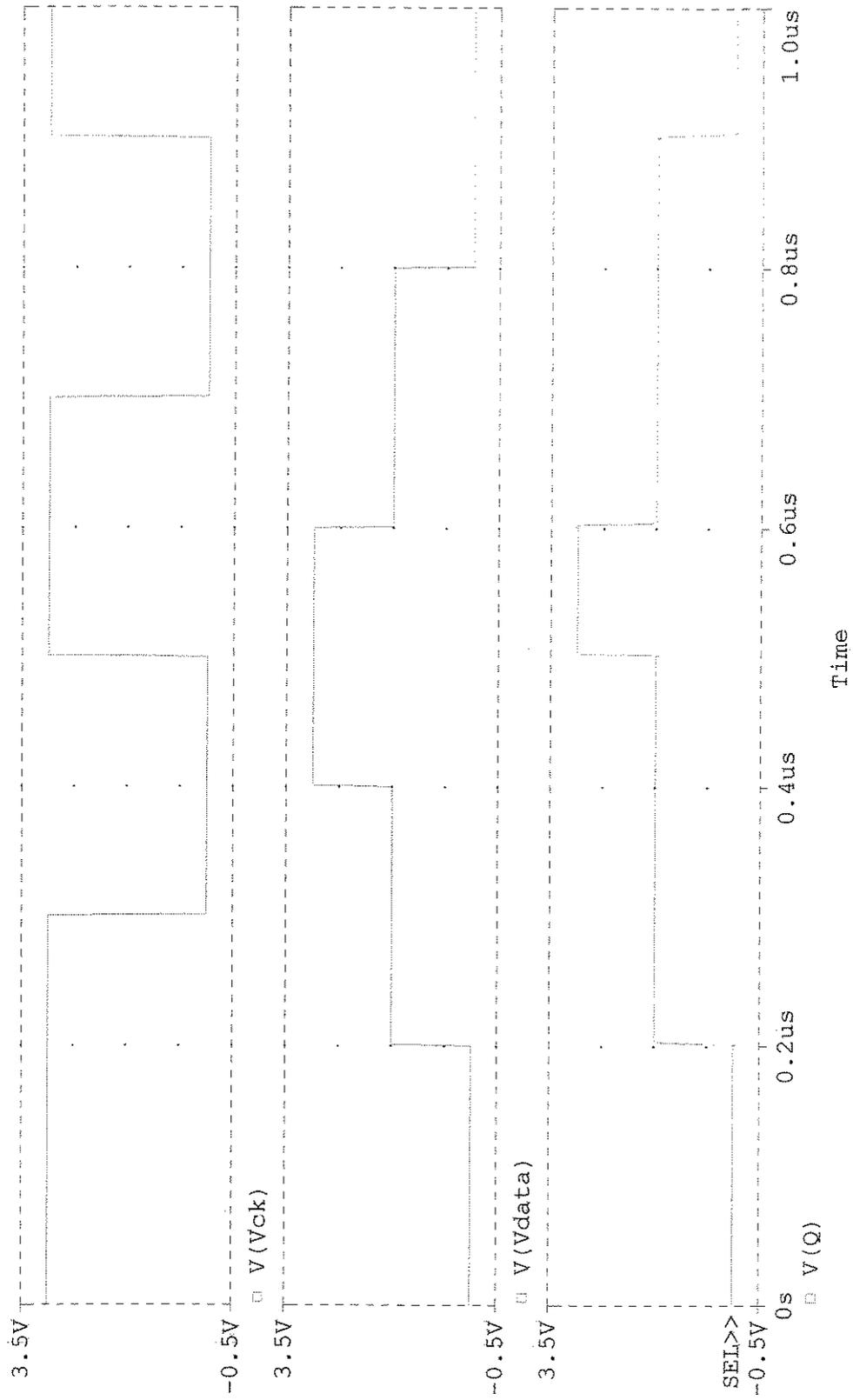


Figura 4.14: Simulação do "latch" FF(012).

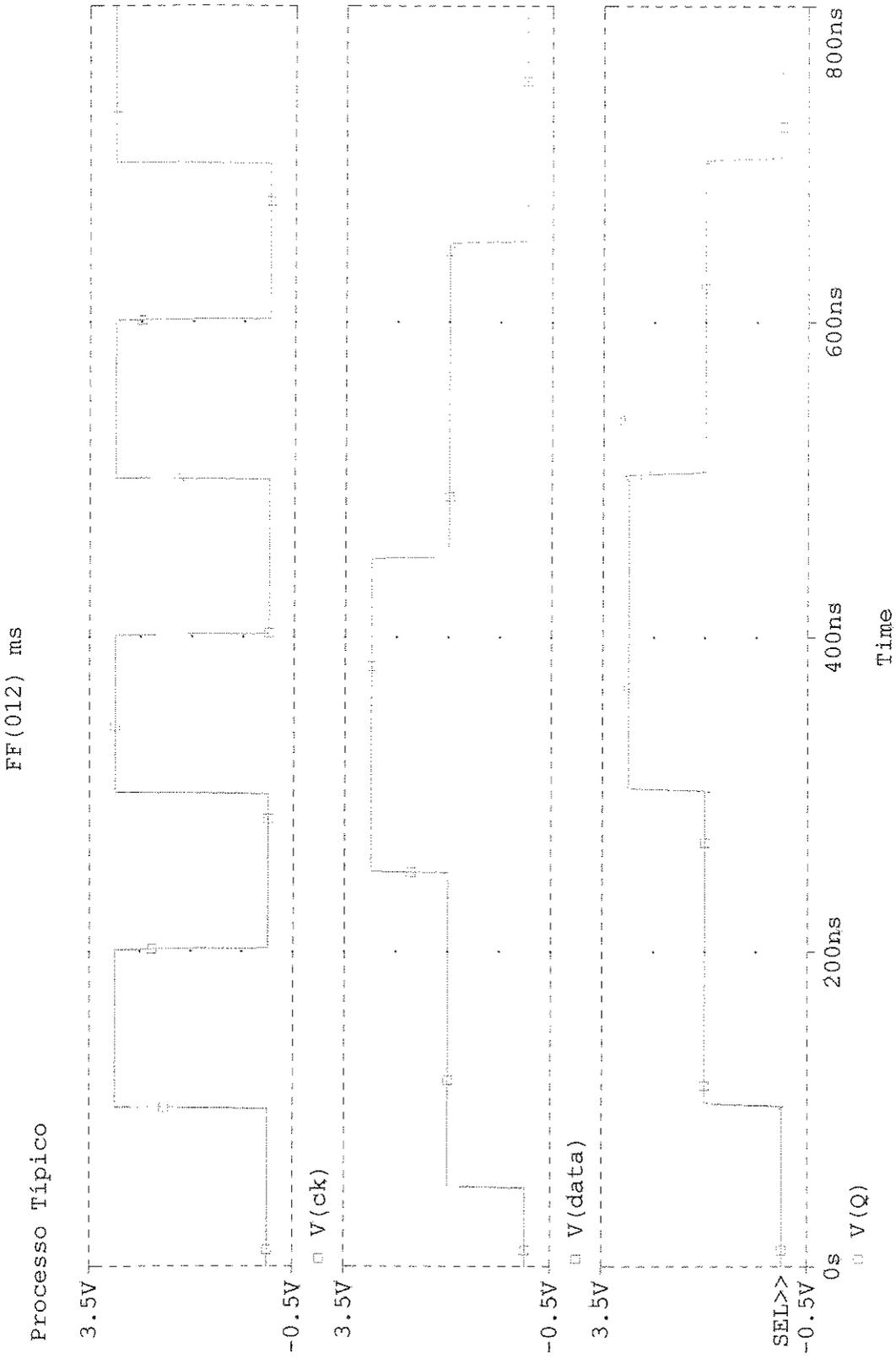


Figura 4.15: Simulação do "flip-flop" FF (012) ms.

FF (120)

Processo Típico

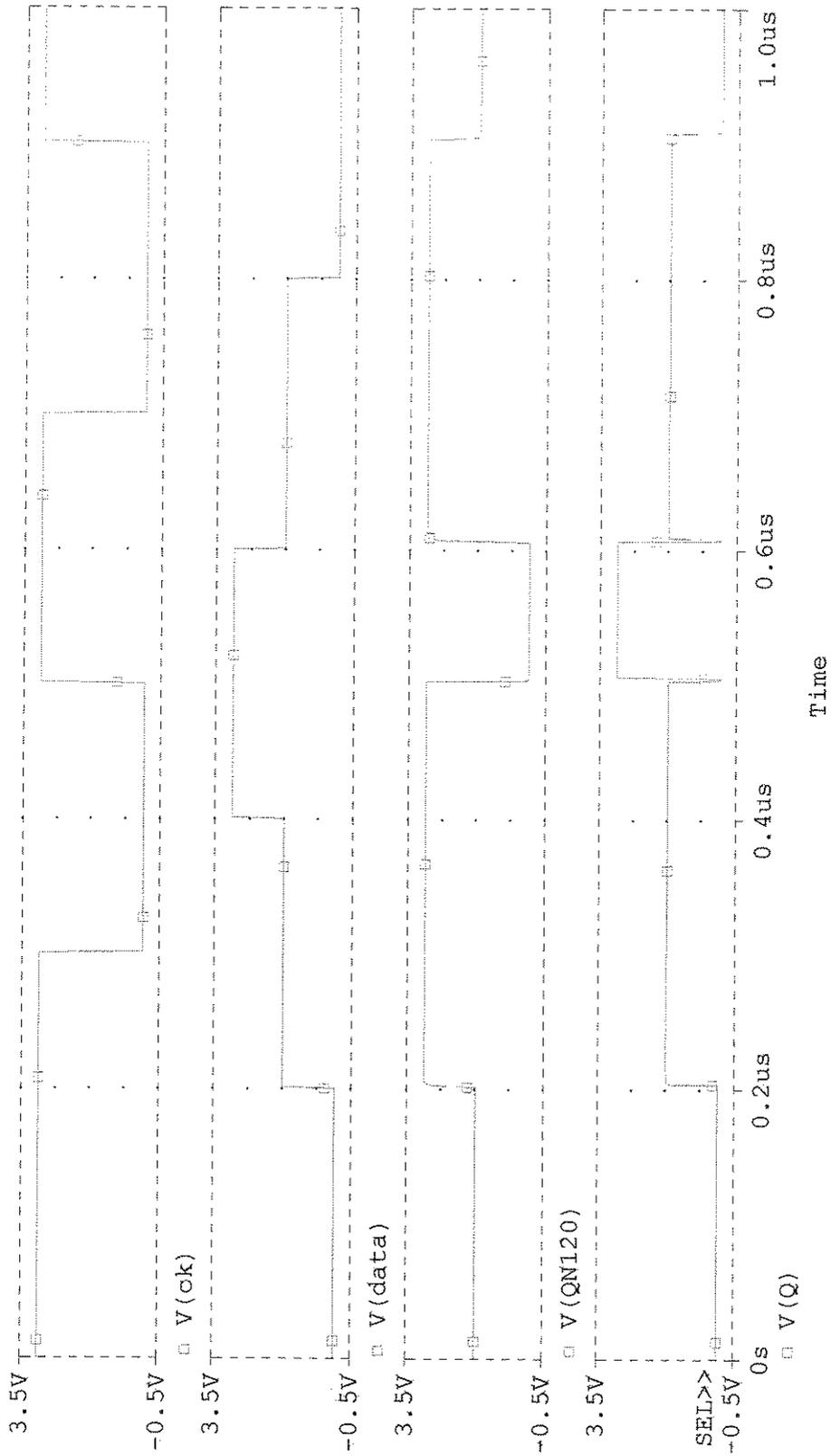


Figura 4.16: Simulação do "latch" FF(120).

FF(201)

Processo Típico

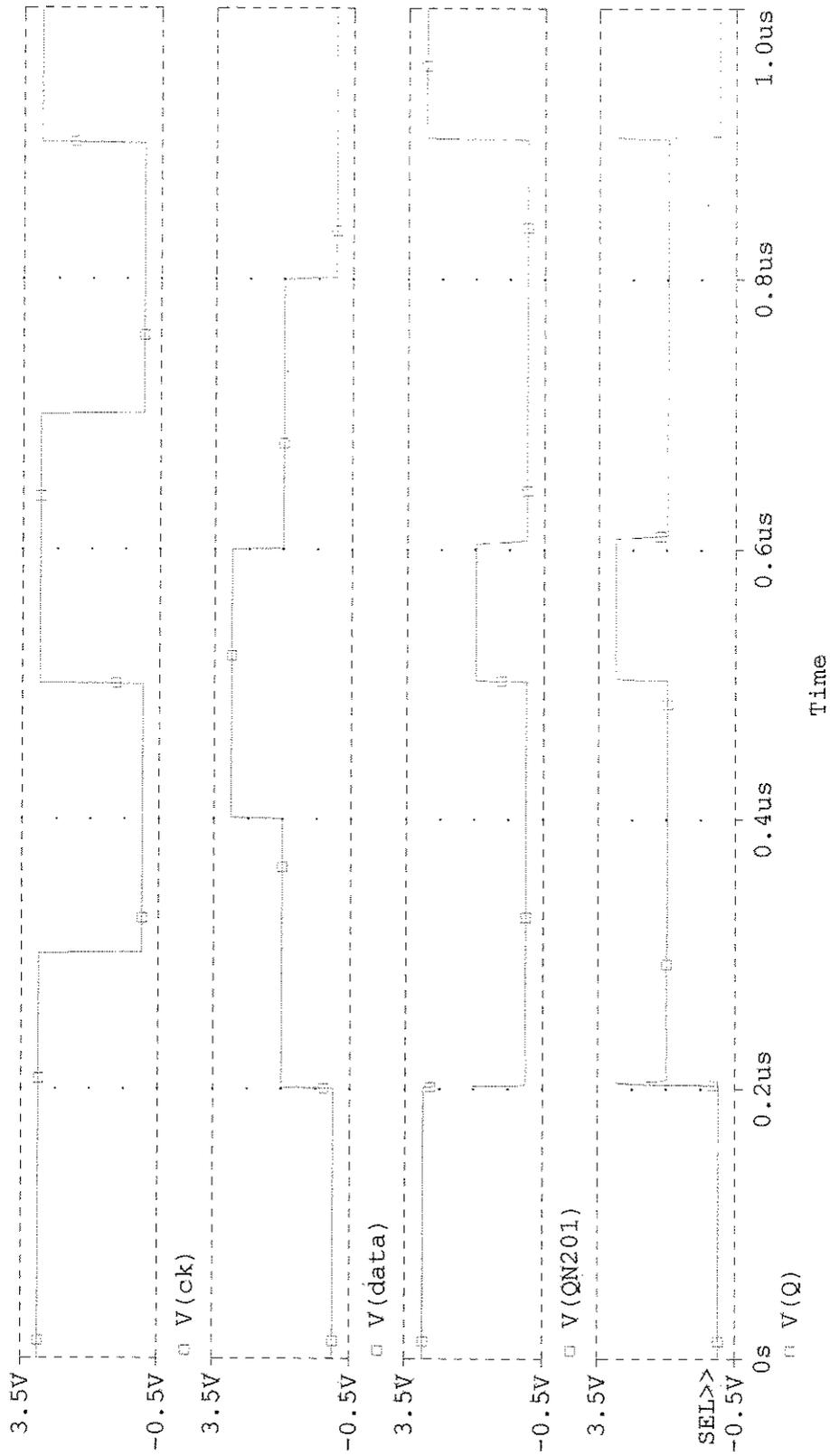


Figura 4.17: Simulação do "latch" FF(201).

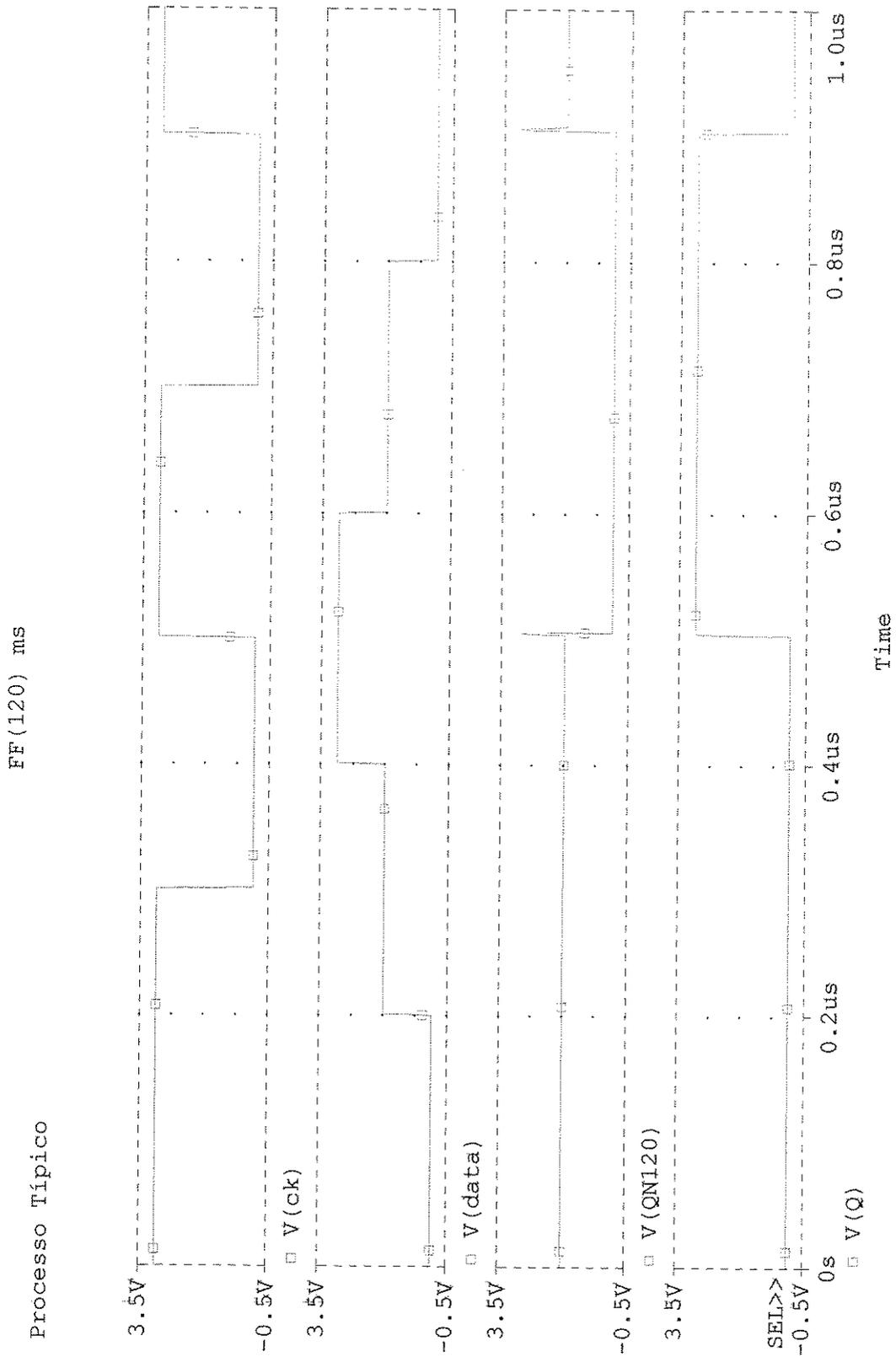


Figura 4.18: Simulação do "flip-flop" FF(120) ms.

CONTADOR

Processo Típico

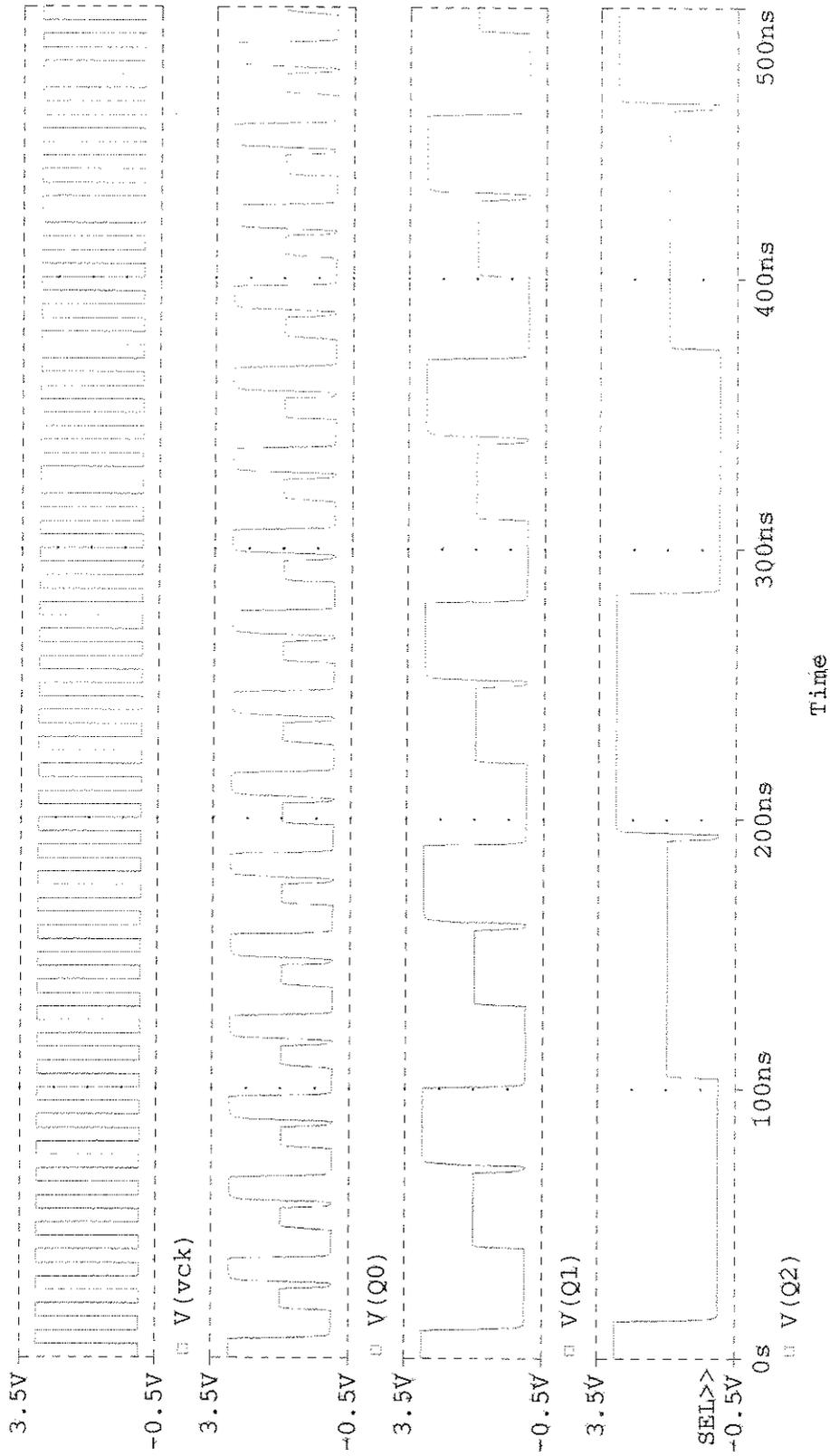


Figura 4.19: Simulação do contador ternário assíncrono de 3 estágios (frequência do relógio de 100MHz).

4.5- CONCLUSÕES

As funções ternárias descritas no Capítulo 3 podem ser usadas para a geração de circuitos com memória. Neste Capítulo descrevemos o uso da função F(012) para a geração do “latch” FF(012) e a seguir, a interligação de dois desses “latches” para a formação do “flip-flop” sensível à borda do relógio, FF(012)_ms. Estes registradores possuem consumo estático praticamente nulo por não necessitarem dos comparadores completos F(220) ou F(200) na sua implementação. No entanto, eles possuem apenas a saída não invertida, Q e, portanto, devem ser usados apenas em contadores/máquinas de estados síncronas.

As funções inversoras cíclicas, F(120) e F(201) são usadas para a geração dos “latches” FF(120) e FF(201), respectivamente. Estes dois “latches” podem ser interligados em série para a formação do registrador sensível à borda do relógio, FF(120)_ms. Este “flip-flop” possui ambas as saídas, Q e QN(120). A saída invertida pode ser realimentada na entrada do “flip-flop” para a implementação de osciladores ternários ou de contadores assíncronos. O projeto de um contador assíncrono de três estágios foi apresentado através da interligação de três “flip-flops” FF(120)_ms, e na simulação SPICE apresentou um atraso máximo de 20,6 ns para a obtenção do algarismo mais significativo (Q2), no caso do processo lento.

De acordo com o resultado das simulações SPICE, os registradores e o contador apresentados podem ser usados até a frequência máxima de 26MHz para o processo lento e até 100MHz para o processo típico.

CAPÍTULO 5

PROJETO DE UM CIRCUITO INTEGRADO TERNÁRIO

Resumo: Neste Capítulo descreveremos o projeto de um circuito integrado (CI) em tecnologia CMOS 0,8 μ m. Após uma descrição geral (Seção 5.1), apresentaremos as funções ternárias implementadas e o “floor-planning” do CI (Seção 5.2). Na Seção 5.3, mostraremos o projeto e “layout” do “buffer” ternário usado nos “pads” de saída. O “layout” final de CI e de alguns circuitos serão apresentados na Seção 5.4.

5.1 – DESCRIÇÃO DO CIRCUITO INTEGRADO

O objetivo deste circuito integrado é testar a metodologia de projeto de circuitos ternários a partir de funções comparadoras de níveis lógicos. Algumas das principais funções descritas no Capítulo 3 foram implementadas, assim como uma função de duas entradas apresentada no Capítulo 2. Os registradores e o contador ternário apresentados no Capítulo 4 também foram incluídos.

O CI possui 28 pads, sendo 5 de entrada, 20 de saída e 3 de alimentação (V0, V1 e V2). A sua dimensão total é de 2270,0 μ m por 2191,9 μ m e a dimensão do “core” (área do circuito sem considerar os “pads”) é de 1901,5 μ m por 1823,4 μ m. O número de transistores, incluindo os dos “pads” é de aproximadamente 1270.

O circuito integrado foi fabricado em tecnologia CMOS de 0,8 μ m da AMS (Espanha), através do projeto PMU do IBERCHIP oferecido pelo CTI (Centro Tecnológico de Informática). O encapsulamento, disponível pelo CTI, foi em um DIL de 40 pinos segundo o esquema apresentado na Fig. 5.1.

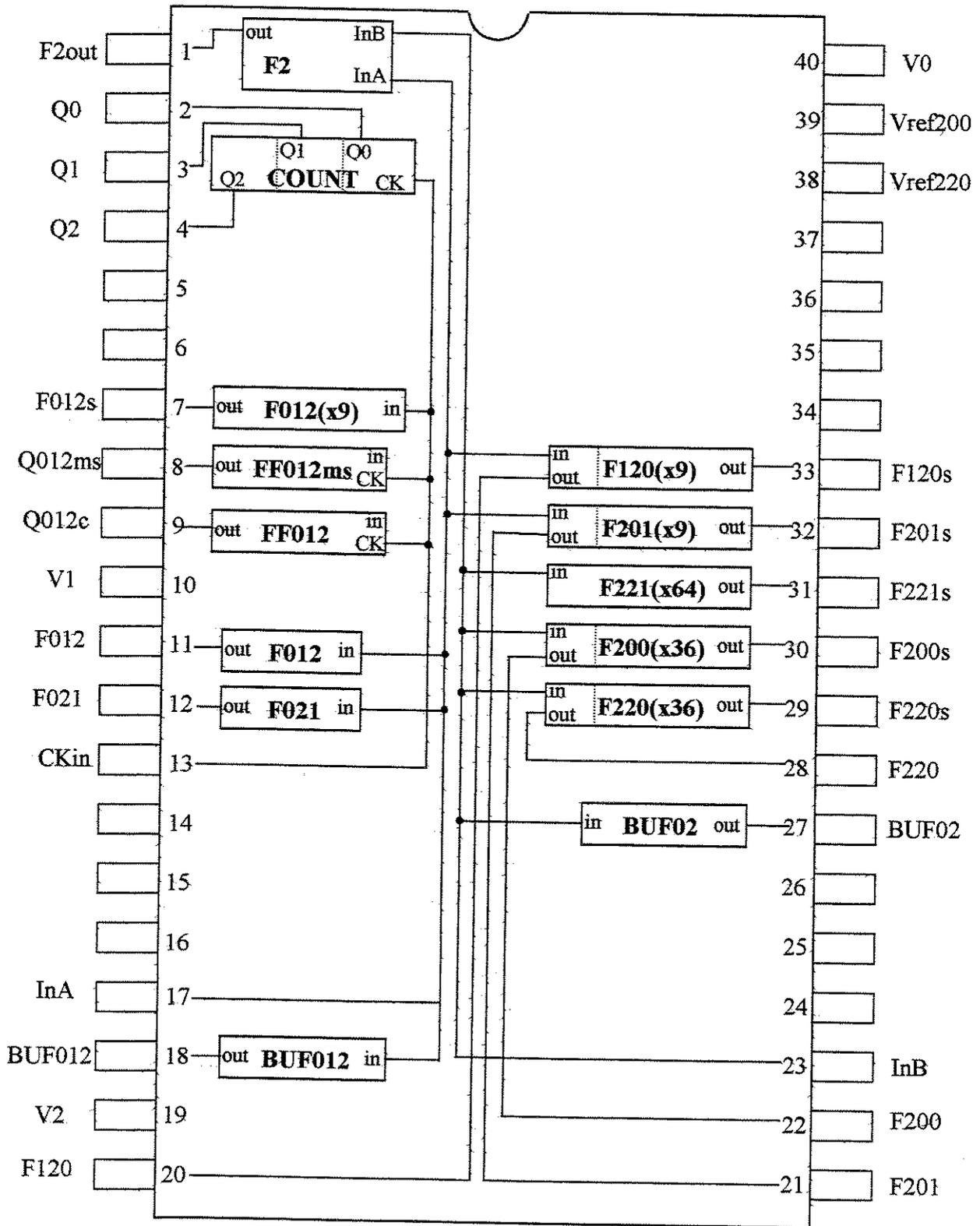


Figura 5.1: Esquema de pinagem do CI ternário

5.2 – FUNÇÕES IMPLEMENTADAS E “FLOOR-PLANNING”

Como pode ser visto pelo esquema de Fig. 5.1, o circuito integrado consiste de várias funções ternárias independentes. Algumas funções, além de terem suas saídas visualizadas através de pads, foram também interligadas em cadeias seqüenciais, para facilitar e dar maior precisão às medidas de atraso. Este é o caso das funções F220, F200, F012, F120 e F201. A função F221 é a mais simples de ser implementada e foi interligada em uma seqüência de 64 portas para avaliação do processo (típico, rápido ou lento).

Circuitos mais complexos como os registradores FF012 e FF012_ms e o contador ternário apresentados no Capítulo 4, também foram incluídos. As saídas do contador podem ser usadas como osciladores ternários no próprio teste de outras funções dentro do CI. A função F2 de duas entradas apresentada no Capítulo 2 (Tabela 2.2), também foi implementada. Na Tabela 5.1 encontra-se a descrição de todas as funções do circuito com os respectivos “pads” de entrada e saída.

FUNÇÃO	DESCRIÇÃO	PINAGEM
ALIMENTAÇÃO	V0 = 0,0V V1 = 1,5V V2 = 3,0V	Entradas: P10 (V1) P19 (V2) P40 (V0)
BUF012	“Buffer” ternário usado em todas as saídas ternárias. Um “buffer” individual foi incluído para teste.	Entrada: P17 (InA) Saída: P18 (BUF012)
BUF02	“Buffer” binário usado nas saídas binárias F(220) e F(200). “Buffer” individual foi incluído para teste.	Entrada: P23 (InB) Saída: P27 (BUF02)
F221s	Seqüência de 64 funções F(221), ligadas em série. Através da medida do atraso da seqüência de portas, o atraso de uma única porta pode ser obtido, e assim, caracterizarmos o processo como típico, rápido ou lento.	Entrada: P23 (InB) Saída: P31 (F221s)

Tabela 5.1: Descrição das funções internas e pinagem do CI ternário(1ª parte).

FUNÇÃO	DESCRIÇÃO	PINAGEM																														
F220	Função F(220) (comparador completo para o nível lógico "2"), usada em vários circuitos internos. Para facilitar o teste foi incluída uma função individual, e a tensão de referência pode ser controlada através da entrada Vref220.	Entradas: P23 (InB) P38 (Vref220) Saída: P28 (F220)																														
F220s	Seqüência de 36 funções F(220), interligadas em série, usada para maior precisão nas medidas de atraso da função.	Entradas: P23 (InB) P38(Vref220) Saída: P29 (F220s)																														
F200	Função F(200) (comparador completo para o nível lógico "0"), usada em vários circuitos internos. Para facilitar o teste foi incluída uma função individual, e a tensão de referência pode ser controlada através da entrada Vref200.	Entradas: P23 (InB) P38 (Vref220) Saída: P28 (F220)																														
F200s	Seqüência de 36 funções F(200), interligadas em série, usada para maior precisão nas medidas de atraso da função.	Entradas: P23 (InB) P39(Vref200) Saída: P30 (F200s)																														
F021	Função ternária (021).	Entrada: P17 (InA) Saída: P12 (F021)																														
F2	<p>Função ternária de duas entradas, conforme tabela abaixo:</p> <table border="1" data-bbox="566 1332 879 1707"> <thead> <tr> <th>InA</th> <th>InB</th> <th>F2(out)</th> </tr> </thead> <tbody> <tr><td>"0"</td><td>"0"</td><td>"0"</td></tr> <tr><td>"0"</td><td>"1"</td><td>"0"</td></tr> <tr><td>"0"</td><td>"2"</td><td>"2"</td></tr> <tr><td>"1"</td><td>"0"</td><td>"1"</td></tr> <tr><td>"1"</td><td>"1"</td><td>"1"</td></tr> <tr><td>"1"</td><td>"2"</td><td>"2"</td></tr> <tr><td>"2"</td><td>"0"</td><td>"1"</td></tr> <tr><td>"2"</td><td>"1"</td><td>"1"</td></tr> <tr><td>"2"</td><td>"2"</td><td>"0"</td></tr> </tbody> </table>	InA	InB	F2(out)	"0"	"0"	"0"	"0"	"1"	"0"	"0"	"2"	"2"	"1"	"0"	"1"	"1"	"1"	"1"	"1"	"2"	"2"	"2"	"0"	"1"	"2"	"1"	"1"	"2"	"2"	"0"	Entradas: P17 (InA) P23 (InB) Saída: P1 (F2out)
InA	InB	F2(out)																														
"0"	"0"	"0"																														
"0"	"1"	"0"																														
"0"	"2"	"2"																														
"1"	"0"	"1"																														
"1"	"1"	"1"																														
"1"	"2"	"2"																														
"2"	"0"	"1"																														
"2"	"1"	"1"																														
"2"	"2"	"0"																														

Tabela 5.1: Descrição das funções internas e pinagem do CI ternário(continuação).

FUNÇÃO	DESCRIÇÃO	PINAGEM
F012	“Driver” ternário, não inversor	Entrada: P17 (InA) Saída: P11 (F012)
F012s	Seqüência de nove “drivers” F012 usada para medidas de atraso da função.	Entrada: P17 (InA) Saída: P7 (F012s)
FF012	“Latch” não inversor, sensível ao nível alto do relógio.	Entradas: P17 (InA) P13(CKin) Saída: P9 (Q012c)
FF012ms	“Flip-flop” mestre-escravo, não inversor, sensível à borda de subida do relógio.	Entradas: P17 (InA) P13(CKin) Saída: P8 (Q012ms)
F120	Inversor cíclico horário, F120.	Entrada: P17 (InA) Saída: P20 (F120)
F120s	Seqüência de nove inversores cíclicos horários, F120, usada para medidas mais precisas de atrasos da função	Entrada: P17 (InA) Saída: P33 (F120s)
F201	Inversor cíclico anti-horário, F201.	Entrada: P17 (InA) Saída: P21 (F201)
F201s	Seqüência de nove inversores cíclicos anti-horários, F201, usada para medidas mais precisas de atrasos da função	Entrada: P17 (InA) Saída: P32 (F201s)
CONTADOR TERNÁRIO	Contador ternário de três estágios, gerado a partir das funções F120 e F201. As mudanças de estado ocorrem sempre nas bordas de subida do relógio.	Entrada: P13(CKin) Saídas: P2 (Q0) P3 (Q1) P4 (Q2)

Tabela 5.1: Descrição das funções internas e pinagem do CI ternário(continuação).

Na Fig. 5.2, apresentamos o planejamento do “layout” do CI (“floor, planning”). Os “pads” de alimentação foram separados para facilitar o roteamento. Os “pads” com a indicação “IB” são entradas com proteção contra tensões estáticas. A maioria dos “pads” de saída possuem “buffers” ternários, BUF012. As saídas dos comparadores F(220) e F(200) são binárias e

foram conectadas a “buffers” binários convencionais, BUF02. O processo CMOS utilizado possui dupla camada de metal e uma única camada de polissilício.

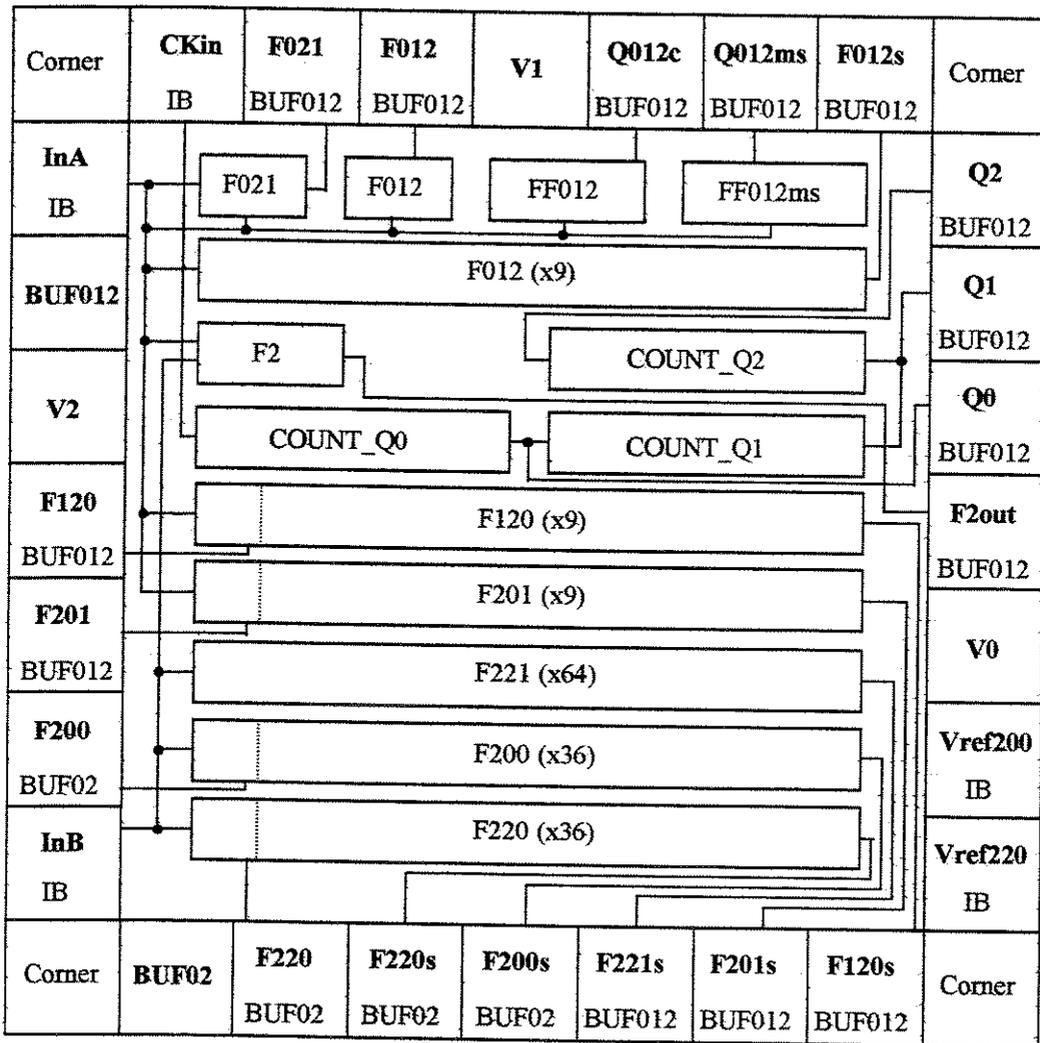


Figura 5.2: “Floor- planning” do CI.

5.3 – “BUFFER” TERNÁRIO DO “PAD” DE SAÍDA

O “driver” de saída usado como “buffer” ternário foi a função F012 (Fig. 3.20), com a relação W/L dos transistores devidamente aumentada para fornecer maior corrente para a carga. O mesmo princípio de escalonamento dos transistores de saída usado em “buffers” binários convencionais, foi usado para o “buffer” ternário. Para as funções F(100) e F(221) foram usadas cadeias de três comparadores simples ao invés de um único comparador, onde as relações “W/L” dos transistores foram aumentadas progressivamente (Fig. 5.3).

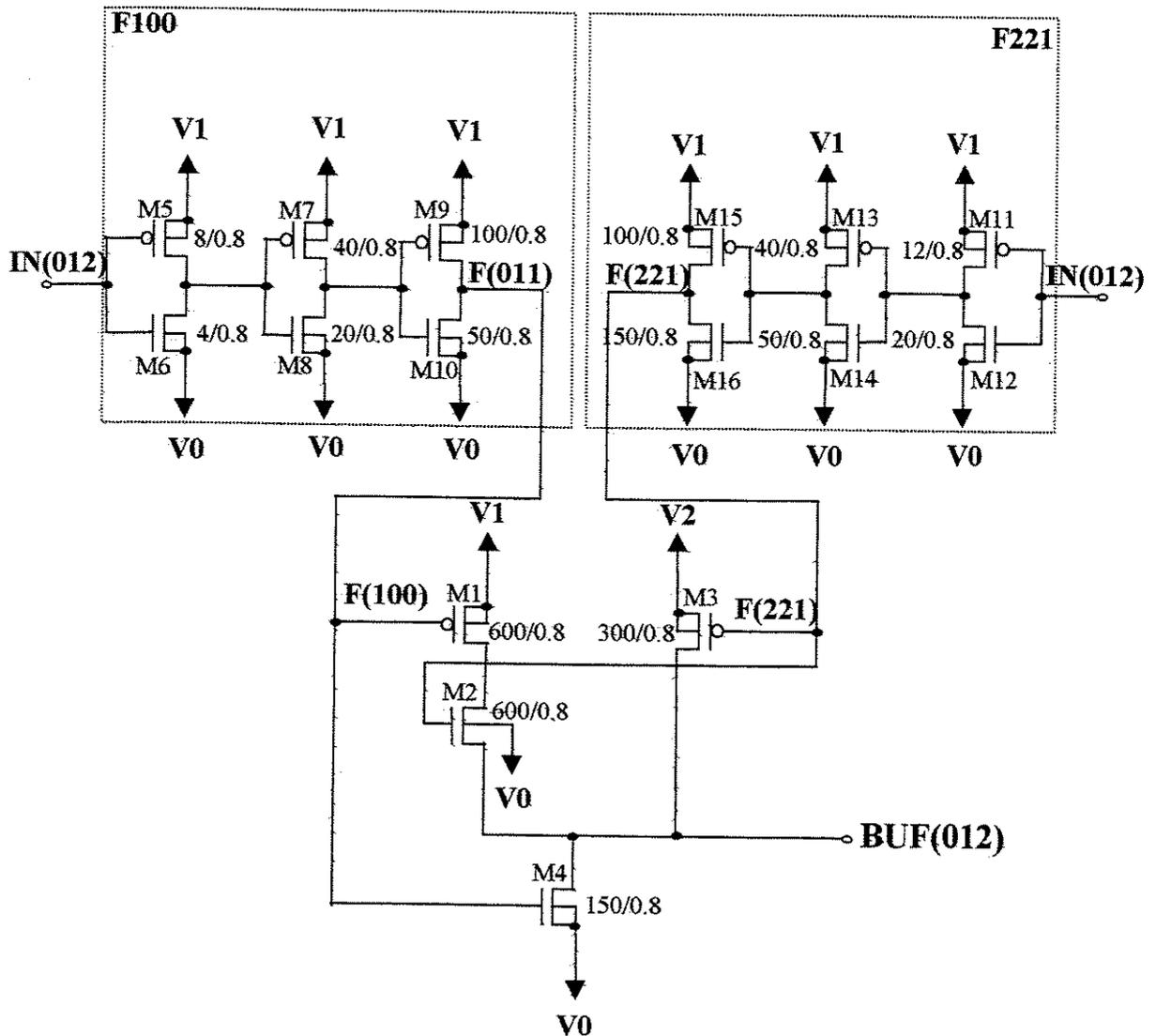


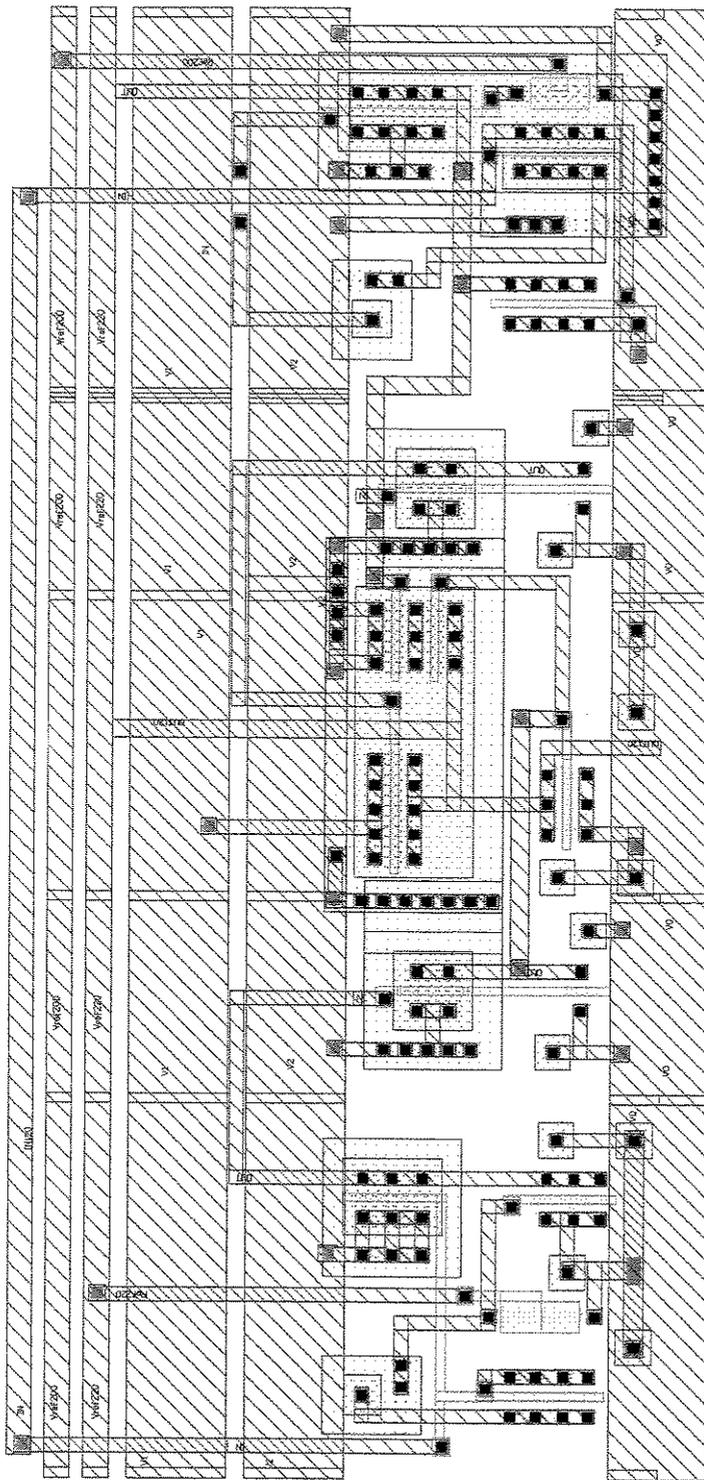
Figura 5.3: Esquema elétrico do “buffer” ternário de saída, BUF012.

O princípio de funcionamento do circuito do “buffer” é o mesmo da função F(012): a função F(100) detecta o nível lógico “0”, chaveando o transistor M4 para V0. A função F(221) detecta o nível lógico “2”, chaveando a saída para V2. A saída será chaveada para V1 quando a entrada não for “0” nem “2”, através de M1 e M2. Os tamanhos dos transistores foram determinados através da simulação SPICE, considerando-se uma carga de saída de 20pF.

5.4 – “LAYOUT” DO CI

O circuito integrado desenvolvido consiste de várias funções independentes e, portanto necessita de muitos “pads” para fornecer acesso às várias saídas. Como consequência, o fator mais determinante no tamanho do CI foi o número de “pads” (circuito “pad limited”). A área de “core” disponível foi utilizada para formarmos as “seqüências” de funções (Tabela 5.1), que facilitariam as medidas mais precisas de atrasos.

A preocupação principal do “layout” foi a modularidade. As funções utilizam a mesma disposição de trilhas para a alimentação (V0, V1 e V2), assim como para as tensões de referência usadas nos comparadores completos (F220 e F200). Com o padrão utilizado, a interligação das funções entre si fica bastante facilitada. Exemplos de “layout” das funções internas podem ser vistos nas Figuras 5.4 e 5.5. O layout do “pad” de saída com o “buffer” ternário (BUF012) é apresentado na Fig. 5.6. Na Fig. 5.7 apresentamos o “layout” do circuito completo.



LEGENDA

	NTUB
	DIFF
	POL1
	NPLU
	CONT
	MET1
	VIA
	MET2

Figura 5.4: “Layout” da função F(120).

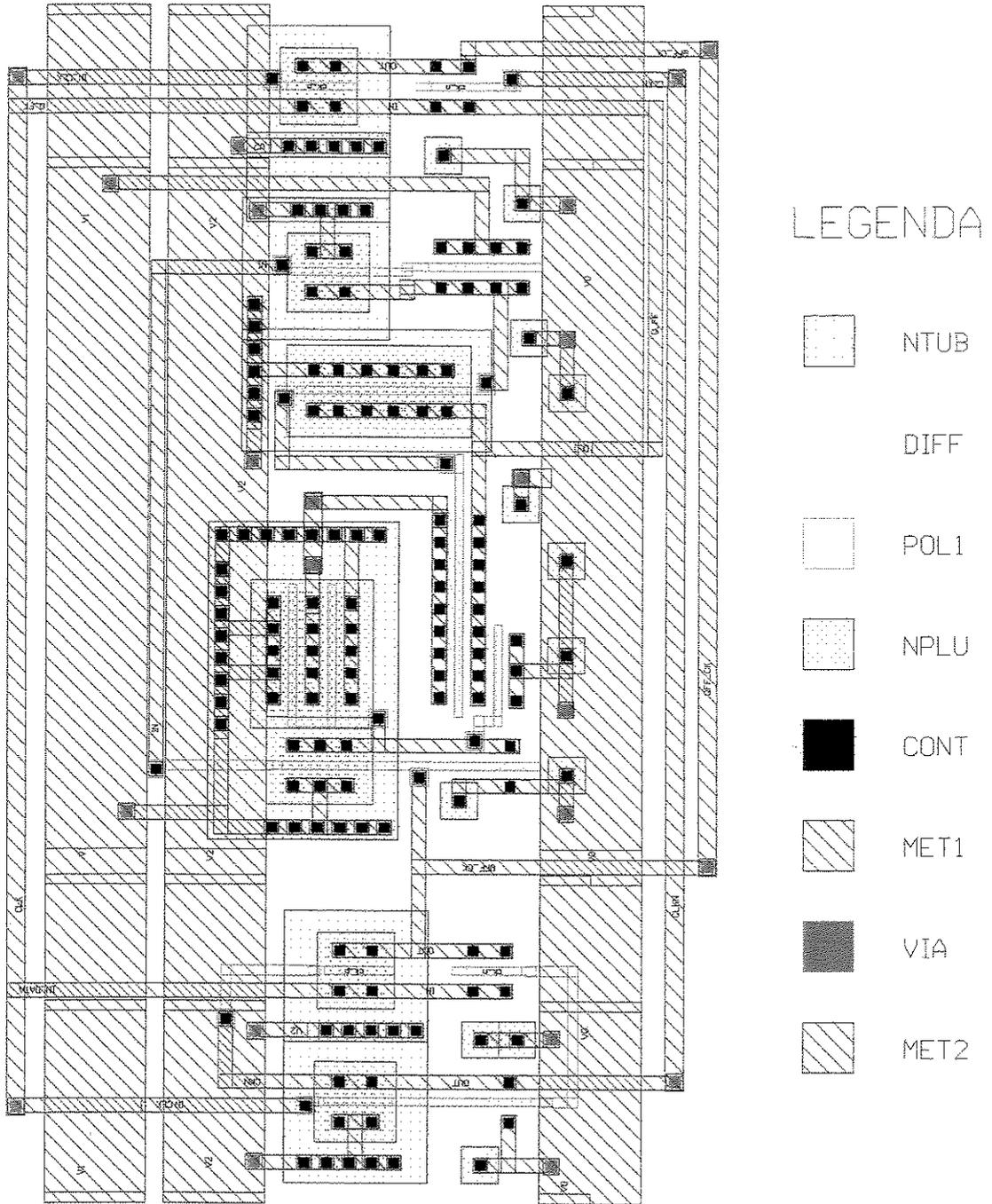


Figura 5.5: "Layout" do "latch" FF(012).

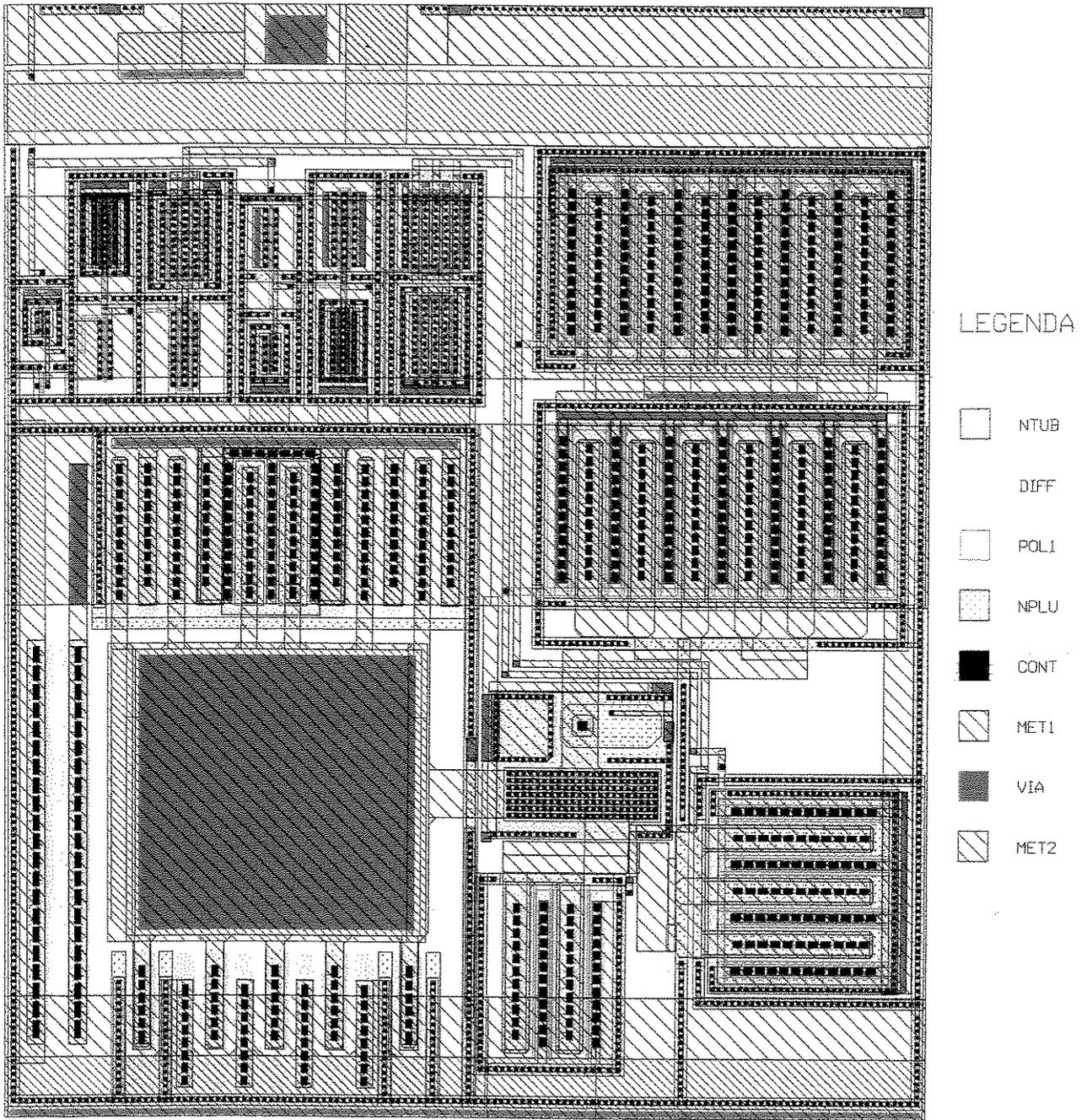


Figura 5.6: "Layout" do "pad" de saída com o "buffer" ternário.

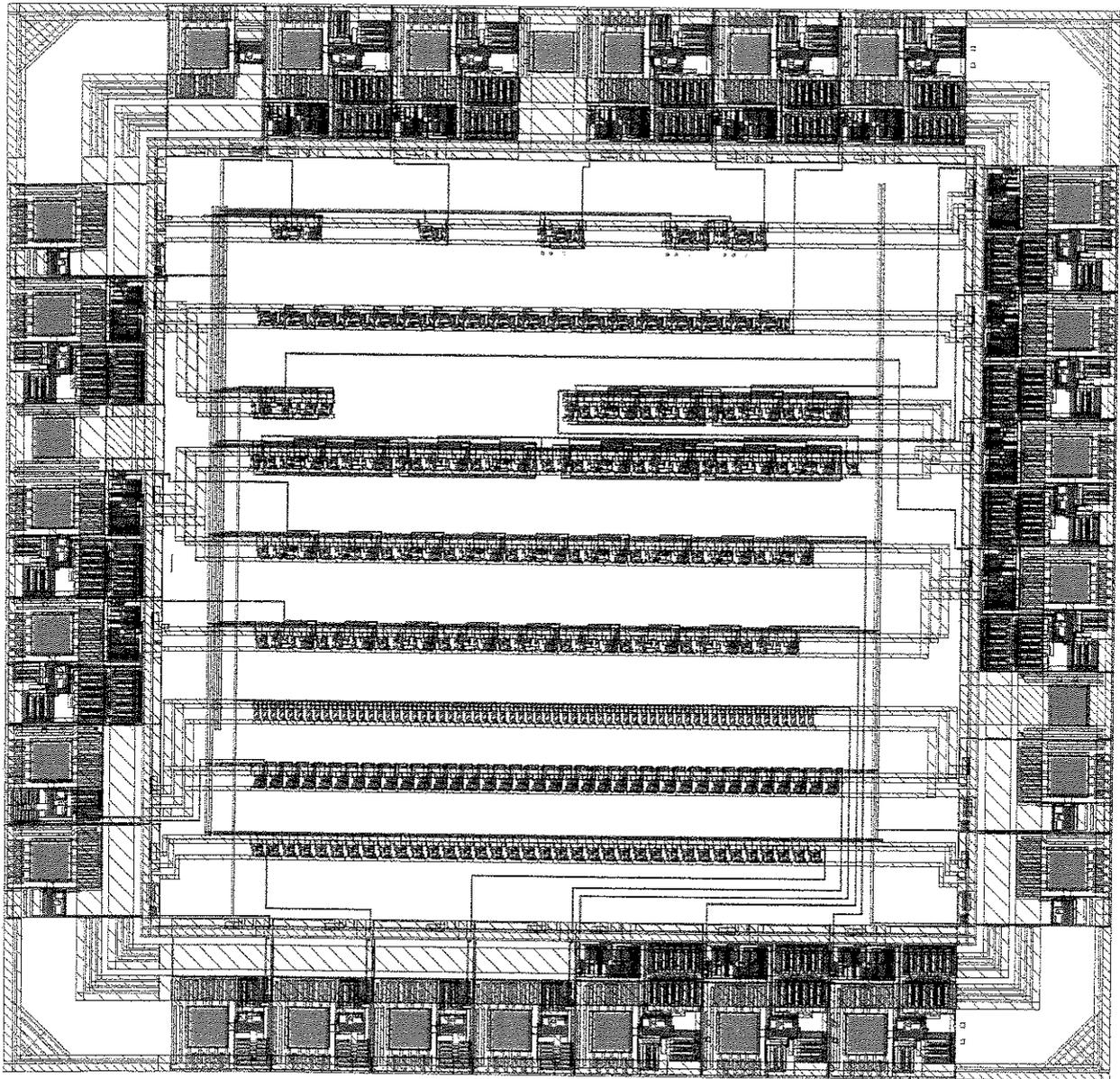


Figura 5.7: “Layout” do circuito integrado.

5.5 - CONCLUSÕES

O circuito integrado ternário projetado foi confeccionado em um processo CMOS convencional (CMOS 0,8 μ m /AMS). A facilidade de implementação do CI é semelhante à dos circuitos binários convencionais, sendo possível o projeto de blocos modulares, facilmente interligáveis entre si.

Embora o CI tenha sido projetado para operar com tensões de “0V”, “1,5V” e “3,0V”, as tensões do nível lógico “1” e “2” podem ser aumentadas em 20% do seu valor para a obtenção de circuitos mais rápidos.

Não foi possível obtermos a caracterização do CI devido a um erro no “layout” dos “pads” de saída ternários (o transistor NMOS do último estágio de amplificação da função F(221) teve o seu dreno indevidamente curto-circuitado com a fonte por uma pequena trilha de metal). O projeto será corrigido e mandado para fabricação novamente. Esperamos desta forma, poder caracterizá-lo devidamente e publicar os resultados em um futuro breve.

CAPÍTULO 6

CONCLUSÕES

O projeto de circuitos em lógica multi-nível tem sido visto como uma alternativa para a crescente complexidade das interconexões dos circuitos integrados, possibilitando a diminuição do número de entradas e saídas e de linhas dos barramentos. Além disso, a capacidade das memórias é aumentada, de acordo com o número de níveis utilizado na lógica.

Neste trabalho desenvolvemos um método de projeto de circuitos ternários em tecnologia CMOS, baseado em circuitos comparadores. A parte combinacional do circuito a ser implementado é descrita em forma de uma tabela ternária, a qual é simplificada através de um método gráfico similar ao Mapa de Karnaugh usado para circuitos binários convencionais. Funções de várias entradas são descritas através de expressões lógicas (interseção ou união) de funções comparadoras de níveis lógicos para cada entrada. A estrutura utilizada é de fácil implementação por programação (CAD).

O projeto elétrico e as simulações elétricas SPICE das diversas funções de uma única entrada foram apresentadas para os diversos tipos de processo. O atraso máximo obtido nas simulações foi de 13,8ns, referente à função F(121), nas condições de processo lento. Os circuitos comparadores de níveis lógicos (F(220), F(002), F(200), F(022), F(020) e F(202)) utilizados no projeto de funções de várias entradas apresentaram atrasos menores do que 9,0ns.

Os circuitos seqüenciais foram desenvolvidos através da realimentação de funções ternárias, em um modo similar ao utilizado para circuitos binários convencionais. Foram apresentados dois tipos básicos de circuitos registradores, um não inversor e outro com saída “invertida” e não “invertida”. Projetos de registradores sensíveis ao nível do relógio (“latches”) e de registradores sensíveis à borda do relógio (“flip-flops” mestre-escravo) foram mostrados,

juntamente com os resultados das simulações Spice. Os maiores atrasos obtidos, referentes ao processo lento, foram de 16,0ns.

Um contador assíncrono ternário de três estágios, foi implementado a partir da ligação em série de três “flip-flops” inversores. Os resultados da simulação Spice indicaram uma frequência máxima de operação de 20 MHz para o processo lento de de 50MHz para o processo típico.

Finalmente, apresentamos o projeto de um circuito integrado, fabricado em tecnologia CMOS 0,8 μm . As principais funções ternárias, os registradores e o contador foram incluídos, além de um “buffer” ternário nos “pads” de saída.

A principal contribuição do trabalho é apresentar um método de projeto de circuitos ternários, bastante simples de ser implementado em tecnologia CMOS convencional. A facilidade de simplificação das funções pelo método gráfico e a modularidade dos circuitos possibilita o projeto automatizado.

Como sugestões para trabalhos futuros temos o desenvolvimento de um programa para simplificação de funções ternárias e o projeto de memórias ternárias. A técnica de projeto apresentada poderá também ser expandida para lógica quaternária, e eventualmente, dependendo da tecnologia e da tensão de alimentação, para cinco ou mais níveis.

REFERÊNCIAS

- [1] Post E. L., "*Introduction to a General Theory of Elementary Propositions*" - Amer. J. of Mathematics, Vol. 43, 1921 p. 163-185.
- [2] Lucasiewicz, J.,- "*O Logice Tr"ojwart"osciowej*", Ruch Filozoficzny, Vol. 5, 1920, pp. 169-171 English Translation – "*On three valued logic*" in L. Borkowski (ed.), Selected Works, North-Holland, Amsterdam.
- [3] Rosembloom, P. C., "*Post Algebra's I. Postulates and General Theory*"- Amer. J. of Math., Vol. 64, 1942, pp. 167-188.
- [4] Frieder, G., A. Fong, C. Y. Chao, "*A Balanced Ternary Computer*", Conference Record of the 1973 Intern. Symp. On Multiple-Valued Logic, Toronto, May 1973, pp. 68-88.
- [5] Epstein, G., G. Frieder, D. C. Rine, "*The Development of Multiple-Valued Logic as Related to Computer Science*" – Computer, Sep. 1974, pp. 20-32.
- [6] Vranesic, S. G., V. C. Hamacher, "*Threshold Logic in Fast Ternary Multipliers*", Proc. 1975 Int. Symp. on Multiple-Valued Logic, Bloomington, May 1975, pp. 373-387.
- [7] Dao, F. T., J. K Russel, D. R Preedy, E. J. McCluskev, "*Multilevel F^2L Threshold Gates*", Dig. IEEE Int. Solid-State Circuits Conf., Feb. 1977, pp.110-111.
- [8] Vranesic, Z.G., "*Multiple-Valued Logic, An Introduction and Overview*", IEEE Tran. Comp. Vol. C-26, Dec. 1977, pp. 1181-1182.
- [9] Kabat, W. C., A. S. Woojcik, "*On the Design of 4-valued digital systems*", IEEE Trans. Comput., Vol. C-30, Sep. 1981, pp. 667-671.

- [10] Kerkhoff, H. G., M. L. Tervoert, "*Multiple-Valued Logic Charge-Coupled Devices*", IEEE Trans. Comput., Vol. C-30, Sep. 1981, pp. 644-652.
- [11] Davio, M., J. P. Deschamps, "*Synthesis of Discrete Functions Using F²L Technology*", IEEE Trans. Comp., Vol.C-30, 1981, pp. 653-661.
- [12] Rich, D. A., K. C. Naiff, K. G. Smalley, "*A Four-State ROM using multilevel Process Technology*" – IEEE Journal of Solid-State Circuits, Vol. SC-19, 1984, pp. 174-179.
- [13] Hurst, S. L. "*Multiple-Valued Logic – Its Status and Its Future*" – IEEE Trans. on Computers, Vol. C-33, 12 Dec. 1984, pp.1160-1179.
- [14] Rich, D. A., "*A Survey of Multi-Valued Memories*" – IEEE Trans. Computers, Feb. 1986, pp. 99-106.
- [15] Hurst, S. L., "*A Survey: Development in Optoelectronics and its Applicability to Multiple Valued Logic*" – Proc. 16th Int. Symp. Multiple-Valued Logic, May 1986, pp.179-187.
- [16] Butler, J. T., "*Multiple-Valued Logic*", Computer, April 1988, pp. 13-15.
- [17] Shamir, J., H. J. Caulsfield, W. Micelli, R. J. Seymour, "*Optical Computing and the Fredkin Gates*", Applied Optics, Vol.25, N. 10, May 1986, pp. 1604-1607.
- [18] Haidar, A. M., F. Q. Li, M. Morisue, "*Design of Josephson Ternary Delta-Gate*" – IEICE Transactions on Information and Systems, E76D: (8) Aug. 1993, pp. 853-862.
- [19] Morisue, M., J. Endo, T. Morooka, Y. Kogure, A. Kanasugi, "*A novel Josephson ternary multiplier*" – IEEE Transactions on Applied Superconductivity, 7: (2) Jun 1997 (Part 3)., pp. 2979-2082

- [20] 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 Transactions on Electron Devices, Dec. 1997, 44: (12), pp. 2149-2153.
- [21] Current, K. W., “*Current-mode CMOS multiple-valued logic circuits*” – IEEE Journal of Solid-State Circuits, 1994, SSC-29, (2), pp. 95-107.
- [22] Shousha, A., “*Switched-Current CMOS Ternary Logic Circuits*”- Int. Journal of Electronics 79: (5) Nov. 1995, pp. 617-625.
- [23] 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.
- [24] Herrfeld, A., S. Hentschke, “*CMOS ternary dynamic differential logic*” – Electronics Letter, 1994, 30: (10), pp. 762-763.
- [25] Toto, F., R. Saletti, “*CMOS dynamic ternary circuit with full logic swing and zero-static power consumption*” – Electronics Letters, 34: (11) May, 28th 1998, pp. 1083-1084.
- [26] Serran, N. V., “*Circuitos Digitais Ternários Baseados na Álgebra de Post – estudo, definição de operadores e implementação*”- Tese de Doutorado, Universidade de Campinas (UNICAMP), Brasil, 1996.
- [27] Serran, N. V., A. M. Jorge, J. A. S. Dias, “*A proposal for the implementation of ternary digital circuits*” – Microelectronics Journal – Vol. 28, 1997, pp. 533-541.
- [28] Taub, H., D. Schilling, “*Eletrônica Digital*” – McGraw Hill, 1982.
- [29] Camilo, D., J. B. T. Yabu-Uti, Y. Yano, “*Circuitos Lógicos – Teoria e Laboratório*” – Livraria Ciência e Tecnologia Editora Ltda (LCTE), 1984.
- [30] Sedra, A. S., K. C. Smith, “*Microelectronic Circuits*” – Oxford University Press, 1991.