

**GUSTAVO CAMPOS MARTINS**

**DESENVOLVIMENTO DE  
BIBLIOTECA DE CÉLULAS PARA  
CIRCUITOS DIGITAIS CMOS DE  
ALTA VELOCIDADE**

Trabalho de Conclusão de Curso  
apresentado à Escola de Engenharia de São  
Carlos, da Universidade de São Paulo

Curso de Engenharia de Computação

ORIENTADOR: Prof. Dr. João Navarro Soares Jr.

São Carlos

2010

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento  
da Informação do Serviço de Biblioteca – EESC/USP

M386d                    Martins, Gustavo Campos  
                            Desenvolvimento de biblioteca de células para  
                            circuitos digitais CMOS de alta velocidade / Gustavo  
                            Campos Martins ; orientador João Navarro Soares Jr. --  
                            São Carlos, 2010.

Trabalho de Conclusão de Curso (Graduação em  
Engenharia da Computação) -- Escola de Engenharia de São  
Carlos da Universidade de São Paulo, 2010.

1. Circuitos integrados MOS. 2. E-TSPC. 3. Biblioteca  
standard cell. 4. ASIC. 5. Circuitos digitais. 6. Dual-  
modulus prescaler. I. Título.

## **Agradecimentos**

Agradeço ao Prof. Dr. João Navarro Soares Jr. pela paciência e pelos ensinamentos em microeletrônica e projetos de circuitos integrados, que propiciaram a conclusão deste trabalho.

Agradeço aos meus pais, Maria e Marcos, por minha educação e pelo apoio moral e financeiro durante todos esses anos de graduação.



Resumo do Projeto de Formatura apresentado à EESC-USP como parte dos requisitos necessários para a obtenção da conclusão do curso de Engenharia de Computação

**DESENVOLVIMENTO DE BIBLIOTECA DE CÉLULAS PARA CIRCUITOS DIGITAIS  
CMOS DE ALTA VELOCIDADE**

Gustavo Campos Martins

11/2010

**Orientador:** Prof. Dr. João Navarro Soares Jr.

**Área de Concentração:** Microeletrônica

**Palavras-chave:** Circuitos Integrados MOS, E-TSPC, biblioteca *standard cell*, ASIC, circuitos digitais, *dual-modulus prescaler*

**RESUMO**

O desenvolvido de ASICs (*Application Specific Integrated Circuits*) é mais rápido e simples se for empregada uma biblioteca de células padrão (*standard cell*) que foi desenvolvida e testada previamente. No entanto, essa metodologia tende a produzir circuitos menos eficientes que consomem mais potência e trabalham em uma velocidade menor, em comparação com circuitos nos quais os transistores são dimensionados e posicionados individualmente no *layout* final (*full custom*). O objetivo deste trabalho é desenvolver uma biblioteca *standard cell*, utilizando a tecnologia AMS (AustriaMicroSystems) 0,35 µm, e empregá-la em alguns circuitos, e comparar o desempenho desses circuitos ao de circuitos desenvolvidos com metodologia *full custom*. A biblioteca *standard cell* desenvolvida neste trabalho contém algumas células utilizadas pela técnica E-TSPC, que é voltada para o projeto de circuitos digitais de alta velocidade. Como resultado, observou-se que os circuitos projetados com a biblioteca *standard cell* desenvolvida apresentam, de fato, menor desempenho em velocidade, cerca de 20% inferior, e ocupam maior área que os mesmos circuitos desenvolvidos em metodologia *full custom*, mas eles são muito mais rápidos, quase três vezes, que circuitos desenvolvidos apenas com células da biblioteca padrão da AMS.



Abstract of graduation project presented to EESC-USP as a partial fulfillment of the  
requirements to conclude the Computer Engineering course

## DEVELOPMENT OF A CELL LIBRARY FOR HIGH SPEED DIGITAL CMOS CIRCUITS

Gustavo Campos Martins

11/2010

**Advisor:** Prof. Dr. João Navarro Soares Jr.

**Concentration Area:** Microelectronics

**Keywords:** MOS Integrated Circuits, E-TSPC, standard cell library, ASIC, digital circuits,  
dual-modulus prescaler

### ABSTRACT

The development of ASICs (Application Specific Integrated Circuits) is faster and simpler if a standard cell library that was previously implemented and tested is employed. However, this methodology is more likely to produce less efficient circuits that consume more power and works at lower speeds, in comparison to circuits completely developed at transistor level (full custom). The aim of this work is to develop a standard cell library, in AMS (AustriaMicroSystems) 0.35  $\mu\text{m}$  technology, and employ it in some circuits, and to compare the performance of these circuits to the performance of circuits developed with the full custom methodology. The standard cell library presented here contains some cells used by the E-TSPC technique, which is directed to the design of high speed digital circuits. As result, it was observed that the circuits designed with the developed standard cell library shows, in fact, a lower speed performance, about 20% lower, and a larger area than the same circuits designed with the full custom methodology, but they are much faster circuits, almost three times, than the circuits designed with only AMS library's cells.



## Índice de Figuras

Figura 1: Representação de uma porta lógica CMOS estática .....	4
Figura 2: Portas lógicas dinâmicas: (a) porta n-dinâmica; (b) porta p-dinâmica .....	5
Figura 3: Porta lógica <i>data pre-charged PH</i> .....	7
Figura 4: <i>Latches</i> utilizados na técnica E-TSPC: (a) n-latch; (b) p-latch .....	8
Figura 5: Registradores dinâmicos TSPC: (a) registrador sensível à borda de subida de <i>clock</i> ; (b) registrador sensível à borda de descida de <i>clock</i> .....	9
Figura 6: Construção de blocos NMOS-like: (a) bloco n-dinâmico e (b) bloco n-dinâmico NMOS-like; (c) bloco p-dinâmico e (d) bloco p-dinâmico NMOS-like; (e) bloco n-latch e (f) bloco n-latch NMOS-like; (g) bloco p-latch e (h) bloco p-latch NMOS-like .....	10
Figura 7: Regras de composição para o TSPC. Os blocos são representados pelos quadrados e o fluxo de dados permitido é representado pelo sentido setas. O # indica ligações onde atrasos diferenciados do <i>clock</i> nos blocos envolvidos não causam problemas ( <i>skew safe communication</i> ); O * indica blocos <i>latch</i> depois dos quais portas CMOS estáticas podem ser inseridas; O $\diamond$ indica ligações onde há problemas potenciais de <i>holding time</i> (NAVARRO, 1998).....	11
Figura 8: Exemplo de n-data chains (NAVARRO, 2002) .....	14
Figura 9: Fusão da saída de dois data chains (NAVARRO, 2002).....	14
Figura 10: Fusão da entrada de dois data chains (NAVARRO, 2002) .....	15
Figura 11: Esquemático do <i>Dual-Modulus Prescaler 32/33</i> (versão com 3 D-FFs) (MIRANDA, 2006). A largura de canal dos transistores, W, em $\mu\text{m}$ está indicada na figura. Para a maior parte dos transistores, o comprimento de canal, L, é $0,35 \mu\text{m}$ ; para alguns transistores, aqueles em que aparecem duas dimensões, é dada a relação W/L.....	18
Figura 12: Esquemático do <i>Dual-Modulus Prescaler 32/33</i> (versão com 2 D-FFs) (MIRANDA, 2006). A largura de canal dos transistores, W, em $\mu\text{m}$ está indicada na figura. Para a maior parte dos transistores, o comprimento de canal, L, é $0,35 \mu\text{m}$ ; para alguns transistores, aqueles em que aparecem duas dimensões, é dada a relação W/L .....	18
Figura 13: Exemplo de <i>layout</i> da célula inversora da biblioteca padrão da AMS .....	21
Figura 14: Esquemático do D-FF convencional sensível à borda de subida. A largura de canal dos transistores, W, em $\mu\text{m}$ está indicada na figura; o comprimento de canal, L, é $0,35 \mu\text{m}$ .....	21
Figura 15: <i>Layout</i> do D-FF convencional sensível à borda de subida .....	22
Figura 16: Esquemático do n-latch com lógica NAND de 2 entradas. A largura de canal dos transistores, W, em $\mu\text{m}$ está indicada na figura; o comprimento de canal, L, é $0,35 \mu\text{m}$ para os transistores PMOS é $0,8 \mu\text{m}$ para o transistor NMOS .....	23
Figura 17: <i>Layout</i> do n-latch com lógica NAND de duas entradas .....	23

Figura 18: Esquemático do p-latch NMOS-like com lógica NOR de 2 entradas. A largura de canal dos transistores, W, em $\mu\text{m}$ está indicada na figura; o comprimento de canal, L, é 0,35 $\mu\text{m}$	24
Figura 19: Layout do p-latch NMOS-like com lógica NOR de 2 entradas .....	24
Figura 20: Esquemático do D-FF convencional sensível à borda de descida. A largura de canal dos transistores, W, em $\mu\text{m}$ está indicada na figura; o comprimento de canal, L, é 0,35 $\mu\text{m}$ .....	25
Figura 21: Layout do D-FF convencional sensível à borda de descida.....	25
Figura 22: Esquemático do D-FF NMOS-like sensível à borda de descida. A largura de canal dos transistores, W, em $\mu\text{m}$ está indicada na figura; o comprimento de canal, L, é 0,35 $\mu\text{m}$ para a maior parte dos transistores, exceção do transistor NMOS mais a esquerda onde é 0,8 $\mu\text{m}$ ....	26
Figura 23: Layout do D-FF NMOS-like sensível à borda de descida.....	26
Figura 24: Esquemático do latch NMOS-like sensível a nível lógico baixo. A largura de canal dos transistores, W, em $\mu\text{m}$ está indicada na figura; o comprimento de canal, L, é 0,35 $\mu\text{m}$ .....	27
Figura 25: Layout do latch NMOS-like sensível a nível lógico baixo .....	27
Figura 26: Esquemáticos dos Prescalers utilizando células da biblioteca proposta: (a) versão do Prescaler com 3 D-FFs; (b) versão do Prescaler com 2 D-FFs.....	28
Figura 27: Layout dos Prescalers desenvolvidos em (MIRANDA, 2006): (a) versão do Prescaler com 3 D-FFs, área de 2848 $\mu\text{m}^2$ ; (b) versão do Prescaler com 2 D-FFs, área de 1787 $\mu\text{m}^2$ .....	29
Figura 28: Layout da versão do Prescaler com 3 D-FFs desenvolvido com metodologia standard cell, 3080 $\mu\text{m}^2$ .....	30
Figura 29: Layout da versão do Prescaler com 2 D-FFs desenvolvido com metodologia standard cell, área de 2247 $\mu\text{m}^2$ .....	31
Figura 30: Forma de onda apresentada pelos circuitos desenvolvidos: (a) versão do Prescaler com 3 D-FFs; (b) versão do Prescaler com 2 D-FFs. O sinal <b>out</b> é a Saída32/33.....	32
Figura 31: Esquemático do Prescaler desenvolvido com células da biblioteca padrão disponibilizada pela AMS .....	35
Figura 32: Layout do Prescaler desenvolvido com células da biblioteca padrão da AMS, área de 3211 $\mu\text{m}^2$ .....	35
Figura 33: Forma de onda apresentada na saída do circuito Prescaler desenvolvido com células da biblioteca padrão da AMS. O sinal de entrada <b>clk</b> tem frequência de 0,85 GHz e tempos de subida e descida de 10% do período do sinal; <b>out</b> é o sinal Saída32/33 .....	36
Figura 34: Visualização dos resultados de simulação da versão do Prescaler com 3-DFFs .....	37
Figura 35: Diagrama de blocos da arquitetura integer N de sistetizadores de frequência .....	45

## Índice de Tabelas

Tabela 1: Regras para ligação de blocos adjacentes em um <i>data chain</i> .....	13
Tabela 2: Comparação entre as implementações <i>full custom</i> e <i>standard cell</i> da versão do <i>Prescaler</i> com 3 D-FFs .....	33
Tabela 3: Comparação entre as implementações <i>full custom</i> e <i>standard cell</i> da versão do <i>Prescaler</i> com 2 D-FFs .....	34
Tabela 4: Resultados da simulação do <i>Prescaler</i> desenvolvido com células da biblioteca padrão da AMS .....	36



## **Lista de Siglas**

AMS	=	AustriaMicroSystems
BSIM3v3	=	Berkeley <i>Short-channel IGFET Model 3, Version 3</i>
CMOS	=	<i>Complementary Metal-Oxide-Semiconductor</i>
D-FF	=	<i>Flip Flop</i> do tipo D
DRC	=	<i>Design Rule Check</i>
E-TSPC	=	<i>Extended – True Single Phase Clock</i>
L	=	Comprimento do canal do transistor
LVS	=	<i>Layout Versus Schematic</i>
NMOS	=	MOSFET tipo N
PH	=	<i>Pre-charged in High</i>
PL	=	<i>Pre-charged in Low</i>
PMOS	=	MOSFET tipo P
RTS	=	<i>Register Transfer System</i>
TSPC	=	<i>True Single Phase Clock</i>
VDD	=	Tensão de alimentação
W	=	Largura do transistor



## Sumário

1.	Introdução .....	1
1.1.	Objetivos .....	1
1.2.	Metodologia .....	2
1.3.	Estrutura da monografia .....	2
1.4.	Ferramentas .....	2
2.	Fundamentos Teóricos .....	3
2.1.	Lógica Combinatorial .....	3
2.1.1.	Lógica Estática CMOS.....	3
2.1.2.	Lógica Dinâmica CMOS .....	4
2.1.3.	Lógica <i>Data Pre-charged</i> .....	6
2.2.	<i>Latches</i> .....	7
2.3.	Registradores .....	8
2.4.	Blocos NMOS- <i>like</i> .....	9
2.5.	Políticas de <i>Clock</i> .....	10
2.6.	Técnica E-TSPC.....	12
2.6.1.	Regras de Composição .....	13
2.6.2.	Estruturas do E-TSPC .....	14
3.	Implementação dos Circuitos .....	16
3.1.	<i>Dual-Modulus Prescalers</i> Escolhidos .....	16
3.2.	Células Escolhidas para a Biblioteca.....	19
3.3.	Desenvolvimento da Biblioteca de Células .....	19
3.3.1.	D-FF Convencional Sensível à Borda de Subida .....	21
3.3.2.	N-Latch NMOS- <i>Like</i> com Lógica NAND de duas entradas .....	22
3.3.3.	P-Latch NMOS- <i>Like</i> com Lógica NOR de duas entradas.....	24
3.3.4.	D-FF Convencional Sensível à Borda de Descida .....	25
3.3.5.	D-FF NMOS- <i>Like</i> Sensível à Borda de Descida.....	26
3.3.6.	<i>Latch</i> NMOS- <i>Like</i> Sensível a Nível Lógico Baixo .....	27

3.4.	Implementação dos Circuitos <i>Prescalers</i> .....	28
4.	Resultados e Comparações.....	33
4.1.	<i>Prescaler</i> com Lógica Estática CMOS .....	34
5.	Conclusão .....	38
	REFERÊNCIAS .....	39
	Apêndice A – Modelos BSIM3v3 dos transistores para a ferramenta ELD0 .....	41
	Apêndice B – Sintetizador de Frequências .....	44

## **1. Introdução**

Os CIs (Circuitos Integrados) têm sido responsáveis pelo crescimento da indústria de eletrônica nas últimas décadas, possibilitando a elaboração de equipamentos cada vez mais complexos e eficientes. A tecnologia CMOS (*Complementary Metal Oxide Silicon*) é amplamente utilizada no desenvolvimento de CIs, aparecendo em cerca de 75% dos CIs fabricados (SEMICONDUCTOR INDUSTRY ASSOCIATION, 2009), em especial em sistemas digitais. O maior uso dessa tecnologia pode ser explicado pelo seu menor preço, por possibilitar maior nível de integração, pelo baixo consumo de potência dos circuitos e pela simplicidade no projeto.

Sempre houve demanda pelo aumento da velocidade de sistemas digitais e existem algumas técnicas no escopo de projeto de CIs que visam esse aumento. Uma dessas técnicas é a E-TSPC (Extended True Single Phase Clock), empregada em RTSSs (*Register Transfer Systems*) e que faz uso de portas lógicas CMOS estáticas, blocos *n-latches*, blocos *p-latches*, CMOS dinâmicas e *data precharged* (NAVARRO, 1998). Essa técnica foi empregada no desenvolvimento dos circuitos utilizados neste trabalho e será apresentada posteriormente.

A aplicação de blocos dinâmicos digitais apresenta diversas dificuldades. O desenvolvimento de uma biblioteca de células dinâmicas, bem como o desenvolvimento de softwares EDA (*Electronic Design Automation*) para checagem das regras e síntese de circuitos E-TSPC, facilitaria o uso de portas lógicas dinâmicas, possibilitando o projeto e aplicação de um número maior de blocos de alta velocidade.

O método de projeto *standard cell*, no qual o projetista utiliza células de uma biblioteca pré-desenvolvida, dá ao projetista maior facilidade no desenvolvimento de seus circuitos. No entanto, esse método tende a gerar circuitos com velocidade de operação menor que à dos circuitos gerados pelo método de projeto *full custom*, no qual o projetista dimensiona e posiciona individualmente os transistores no *layout* final. Isso se deve à maior flexibilidade no dimensionamento, posicionamento e roteamento dos transistores nos circuitos que o método *full custom* possibilita.

### **1.1. Objetivos**

Este trabalho consiste em estudar a viabilidade do projeto de circuitos E-TSPC utilizando uma biblioteca *standard cell* e comparar os circuitos gerados com essa biblioteca com circuitos gerados com a metodologia de projeto *full custom*. As características dos circuitos que serão comparadas são velocidade máxima, área do *layout* e consumo de potência. O trabalho foi desenvolvido com a tecnologia CMOS 0,35  $\mu\text{m}$  da AMS (AustriaMicroSystems), que, devido a

custos, é ideal para prototipagens. Nessa tecnologia são permitidos quatro níveis de metal e dois níveis de silício policristalino (AUSTRIAMICROSYSTEMS, 2003).

## 1.2. Metodologia

O trabalho foi dividido em quatro fases: (1) estudo da técnica E-TSPC e de outras técnicas de circuitos digitais de alta velocidade; (2) desenvolvimento e caracterização de uma biblioteca mínima de células; (3) desenvolvimento de circuitos E-TSPC utilizando a biblioteca de células e (4) comparação dos circuitos com circuitos *full custom* que também foram desenvolvidos com a tecnologia CMOS 0,35 µm da AMS.

A etapa de desenvolvimento da biblioteca de células consiste em selecionar as células que formarão a biblioteca, determinar dimensões para os transistores, simular os circuitos a fim de testar as dimensões escolhidas, desenhar o *layout* e caracterizar as células.

## 1.3. Estrutura da monografia

O próximo capítulo contém o embasamento teórico necessário para guiar o desenvolvimento do projeto. O terceiro capítulo apresenta a implementação da biblioteca de células e de circuitos que utilizam as células da biblioteca, os passos tomados durante a escolha das células e o desenho dos *layouts*, e a simulação dos circuitos extraídos dos *layouts* dos circuitos projetados com células e com a metodologia *full custom*. O quarto capítulo mostra os resultados da simulação dos *layouts* e as comparações realizadas. Finalmente, o último capítulo apresenta as conclusões do trabalho.

## 1.4. Ferramentas

Foram utilizadas no projeto as seguintes ferramentas EDA da Mentor Graphics (2010):

- **ELDO:** simulação de circuitos;
- **EZwave:** visualização dos resultados de simulação dos circuitos.
- **Design Architecture:** desenho de esquemáticos;
- **IC Station:** desenho de *layouts*;
- **Calibre:** verificação DRC (*Design Rule Check*) e extração de arquivos para simulação a partir do *layout*;

Maiores informações sobre os *softwares* utilizados podem ser encontradas em (MENTOR, GRAPHICS, 2010). Também foi utilizado o HIT-kit AMS, um *design kit* da AMS que é associado às ferramentas da Mentor Graphics e disponibiliza bibliotecas de células e modelos BSIM3v3 (*Berkeley Short-channel IGFET Mode 3, version 3*) para os transistores da tecnologia CMOS 0,35 µm que serão utilizados em simulações (Apêndice A) (AUSTRIAMICROSYSTEMS, 2010).

## **2.Fundamentos Teóricos**

Este capítulo apresenta a fundamentação teórica necessária para entender a técnica E-TSPC, uma evolução da técnica TSPC (*True Single Phase Clock*), bem como sua apresentação. Ambas as técnicas, E-TSPC e TSPC, são utilizadas para o desenvolvimento de circuitos digitais RTS. A técnica E-TSPC acrescenta novas regras de projeto e mudanças nas topologias de algumas das células utilizadas a fim de aumentar a velocidade dos circuitos (NAVARRO, 1998).

A seguir são apresentados tipos de portas lógicas presentes na técnica E-TSPC, suas topologias e seus problemas. Os RTSs e a política TSPC também são discutidos. Por fim, as regras da E-TSPC e as novas estruturas por ela possibilitadas são apresentadas.

### **2.1. Lógica Combinatorial**

O bloco cuja saída é uma função lógica dos sinais aplicados nas entradas é chamado de bloco de lógica combinatorial (RABEY, 2006). Os tipos de portas lógicas utilizadas na técnica E-TSPC são: CMOS estática, CMOS n-dinâmica, CMOS p-dinâmica, blocos *p-latch*, blocos *n-latch* e *data pre-charged*. A seguir são apresentados cada um desses tipos de blocos e suas principais características.

#### **2.1.1. Lógica Estática CMOS**

Uma porta lógica estática CMOS é uma combinação de duas estruturas complementares, uma chamada PUN (*Pull-Up Network*) e outra chamada PDN (*Pull-Down Network*), dispostas como mostra a Figura 1. A estrutura PUN é ligada ao  $V_{DD}$ , estabelecendo condicionalmente um caminho de baixa impedância entre a saída e o  $V_{DD}$ , e a estrutura PDN é ligada  $V_{SS}$ , estabelecendo condicionalmente, de modo similar à PUN, um caminho de baixa impedância entre a saída e o  $V_{SS}$ . A estrutura PUN é composta apenas de transistores PMOS e a estrutura PDN apenas de transistores NMOS. A célula de lógica estática CMOS deve ser projetada de modo que as estruturas PUN e PDN trabalhem de modo complementar, não conduzindo simultaneamente, ou seja, nunca deve ser formado um caminho de baixa impedância entre o  $V_{DD}$  e  $V_{SS}$ . Deste modo, a saída estará ligada sempre a  $V_{DD}$ , nível lógico alto, ou  $V_{SS}$ , nível lógico baixo.

As vantagens das portas lógicas estáticas CMOS são:

- altas margens de ruído;

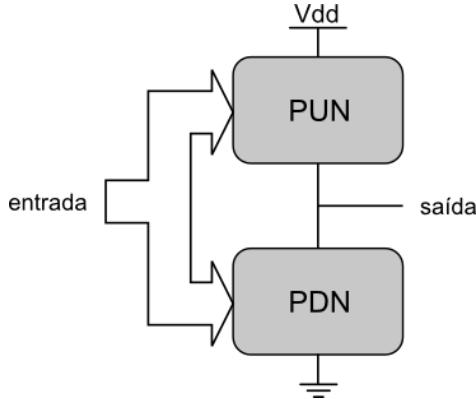


Figura 1: Representação de uma porta lógica CMOS estática

- baixa potência dissipada estaticamente, pois neste tipo de porta quando não há transições nos sinais de entrada a potência dissipada é devido a apenas correntes de fuga;
- tempos de subida e descida da saída podem ser próximos se a porta for projetada de forma correta;
- baixa sensibilidade a variações no processo de fabricação e facilidade de projeto (*ratioless*; o funcionamento não depende das dimensões dos transistores).

Por outro lado, suas deficiências são as seguintes:

- há gastos maiores com área, uma vez que o número de transistores empregados é  $2N$ , onde  $N$  é o número de sinais de entrada;
- funções lógicas complexas degradam a velocidade do circuito.

### 2.1.2. Lógica Dinâmica CMOS

A topologia das portas lógicas dinâmicas CMOS possui apenas uma estrutura PUN ou PDN e dois transistores, um PMOS e um NMOS, ligados ao sinal de *clock*, assim como apresentado na Figura 2 (RABEY, 2006). Do mesmo modo que nas portas estáticas CMOS, a estrutura PUN é formada apenas por transistores PMOS e a PDN é formada apenas por transistores NMOS. Existem dois tipos de portas dinâmicas, assim como mostra a Figura 2: porta n-dinâmica, que possui uma estrutura PDN, e porta p-dinâmica, que possui uma estrutura PUN.

As portas lógicas dinâmicas CMOS possuem duas fases de operação: a fase de pré-carga e a fase de avaliação. Durante a pré-carga, quando o sinal de *clock* está em nível lógico baixo em uma porta n-dinâmica ou em nível lógico alto em uma porta p-dinâmica, a saída é levada para  $V_{DD}$ , no caso de uma porta n-dinâmica, ou  $V_{SS}$ , no caso de uma porta p-dinâmica. Quando o sinal de *clock* é invertido inicia-se a fase de avaliação, na qual a saída assumirá um valor que depende dos sinais de entrada. Nessa fase a saída pode ter o seu valor invertido, caso a estrutura PUN ou PDN estiver conduzindo, ou apresentar alta impedância, mantendo o valor pré-carregado por um período de tempo.

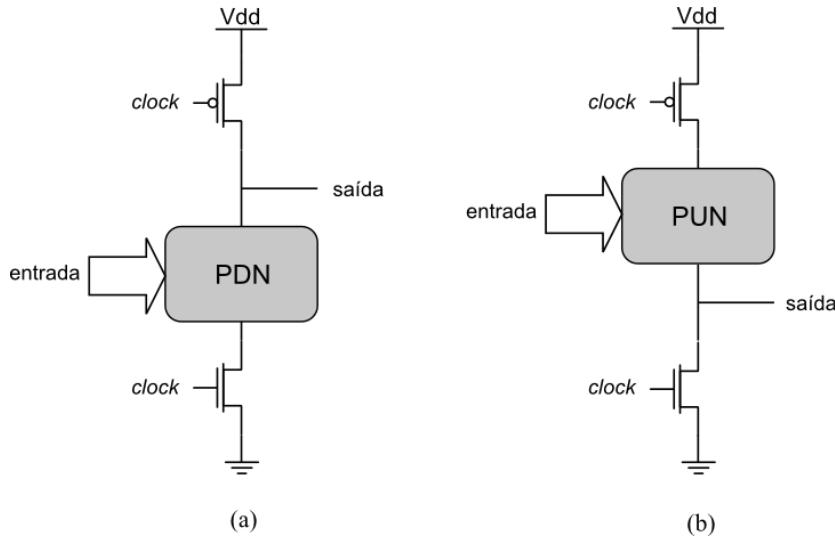


Figura 2: Portas lógicas dinâmicas: (a) porta n-dinâmica; (b) porta p-dinâmica

Observa-se que a saída não pode ser invertida duas vezes durante a fase de avaliação, pois, no caso da porta n-dinâmica, uma vez que o nó de saída é descarregado não há como carregá-lo novamente, pois o transistor que está ligado em  $V_{DD}$ , na n-dinâmica, ou em  $V_{SS}$ , na p-dinâmica estará cortado durante a avaliação. Com isso é possível afirmar que durante a fase de avaliação as portas n-dinâmicas não apresentarão a saída correta em transições “1” $\rightarrow$ “0” dos sinais de entrada e, de modo complementar, as portas p-dinâmicas não apresentarão a saída correta em transições “0” $\rightarrow$ “1” dos sinais de entrada.

As vantagens das portas dinâmicas são as seguintes:

- a carga capacitiva de entrada da porta é pequena e a área utilizada é menor. Isto ocorre porque o número de transistores utilizados é  $N + 2$ , onde  $N$  é o número de sinais de entrada, um número pequeno se comparado aos  $2N$  transistores das portas estáticas CMOS;
- a porta consome potência durante a pré-carga caso o valor pré-carregado anteriormente tenha sido modificado, mas não consome na avaliação. Desse modo, a porta dinâmica pode consumir maior potência que porta a estática. No entanto, se utilizada em circuitos nos quais a saída é modificada com alta frequência, as portas dinâmicas podem ser mais eficientes;
- nas portas n-dinâmicas a função lógica é executada apenas por transistores NMOS, tornando o circuito mais rápido, pois os transistores NMOS são duas ou mais vezes mais velozes que os PMOS.
- o funcionamento correto não depende das dimensões dos transistores utilizados (*ratioless*);

Em contrapartida, os problemas presentes nas portas lógicas dinâmicas são os seguintes:

- *charge sharing*: este efeito ocorre durante a fase de avaliação e consiste na redistribuição da carga armazenada entre os nós internos da estrutura PDN ou PUN;
- *charge leakage*: perda da carga armazenada devido às correntes de fuga quando a porta apresenta a saída em alta impedância, ocorre durante a avaliação;
- *acoplamento capacitivo entre entradas e saída*: quando um sinal de entrada tem o seu valor modificado, a saída sofre alteração devido à redistribuição da carga do canal dos transistores. O caso do acoplamento entre o *clock* e a saída é chamado de *clock-feedthrough*;
- *limite mínimo da frequência do clock*: devido ao *charge leakage* a porta não deve permanecer na fase de avaliação por muito tempo, impondo, assim, uma frequência mínima para o sinal de *clock*. Isso pode não ser considerado um problema, tendo em vista que essas portas são utilizadas principalmente em circuitos de alta velocidade;
- *maiores cuidados no projeto do circuito*: visto que certas transições dos sinais de entrada não são possíveis durante a fase de avaliação, devem-se seguir certas regras para compor circuitos com esse tipo de portas lógicas. Por exemplo, duas portas n-dinâmicas não podem estar ligadas diretamente; é necessário um número ímpar de inversões entre as duas portas. A técnica E-TSPC fornece regras para criar circuitos corretos com portas lógicas dinâmicas e que serão apresentadas posteriormente;
- *margens de ruído menores*: as margens de ruído da porta dinâmica são reduzidas devido aos problemas de *charge sharing* e acoplamento capacitivo.

Pode-se observar que a maior parte dos problemas citados ocorre devido ao fato que a saída pode permanecer em estado de alta impedância durante a fase de avaliação. Nesse estado, a saída da porta lógica não fica ligada diretamente nem ao  $V_{DD}$  e nem ao  $V_{SS}$ , uma condição que exige cuidados.

### **2.1.3. Lógica *Data Pre-charged***

As portas *data pre-charged*, ou simplesmente portas Dp, são pouco conhecidas e utilizadas. Essas portas são um misto entre as portas estáticas e as dinâmicas e nelas a saída é pré-carregada em função de alguns dos sinais entradas (YUAN et. al., 1993). As entradas que pré-carregam a saída serão chamadas, assim como em (NAVARRO, 1998), de pc-entradas (*pre-charging* entradas).

As portas Dp são divididas em duas classes: portas PH (*Pre-charged in High*), nas quais a saída é pré-carregada quando as pc-entradas estão em nível lógico alto, e portas PL (*Pre-charged in Low*), nas quais a saída é pré-carregada quando as pc-entradas estão em nível lógico baixo. A Figura 3 apresenta um exemplo de porta PH que executa a função lógica  $s = \overline{ab} + c$ , nessa porta os sinais de entrada  $a$  e  $c$  fazem a pré-carga da saída, ou seja, são as pc-entradas.

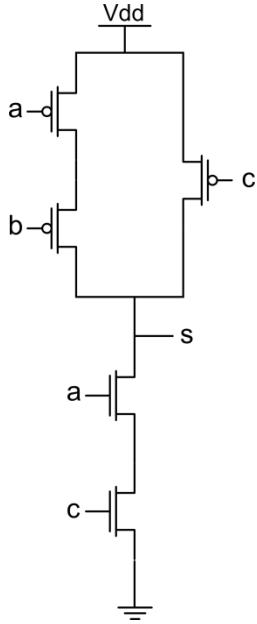


Figura 3: Porta lógica *data pre-charged* PH

Na porta apresentada na Figura 3, a estrutura formada pelos transistores PMOS é igual ao que é feito em uma porta estática. Já na estrutura formada pelos transistores NMOS, o transistor referente ao sinal de entrada *b* foi removido. Nesse circuito, durante a pré-carga as entradas *a* e *c* devem estar em nível alto e durante a saída pelo menos uma dessas entradas deve estar em nível baixo, para que o  $V_{DD}$  não fique ligado diretamente ao  $V_{SS}$ . Essas condições podem ser satisfeitas utilizando outros circuitos associados às portas Dps.

As portas lógicas Dps são mais rápidas que as portas estáticas CMOS, visto que o número de transistores do circuito é reduzido. Os problemas desse tipo de porta são: maior dificuldade em seu uso, já que é necessário garantir os valores das pc-entradas em dois momentos diferentes (pré-carga e avaliação); e pode consumir mais potência que as portas CMOS estáticas se a saída durante a avaliação for diferente da saída durante a pré-carga.

As portas Dps PH ou PL que fazem a pré-carga quando o *clock* está em nível lógico alto são chamadas de n-Dps e as que fazem a pré-carga quando o *clock* está em nível lógico baixo são chamados de p-Dps (NAVARRO, 1998). Elas não são muito utilizadas em geral, mas oferecem maior flexibilidade para a técnica E-TSPC. Neste trabalho, em função dos circuitos escolhidos, não serão aplicadas portas *data pre-charged*.

## 2.2. Latches

Os circuitos *latch* têm como função armazenar um *bit* durante certo período de tempo (RABEY, 2006). Os *latches* também podem executar funções lógicas operando sobre seus sinais de entrada. Na técnica E-TSPC são utilizados dois tipos de *latches*, chamados de n-latches e os p-latches. Ambos tipos são *ratioless* e apresentam estruturas PUN e PDN. A diferença entre deles,

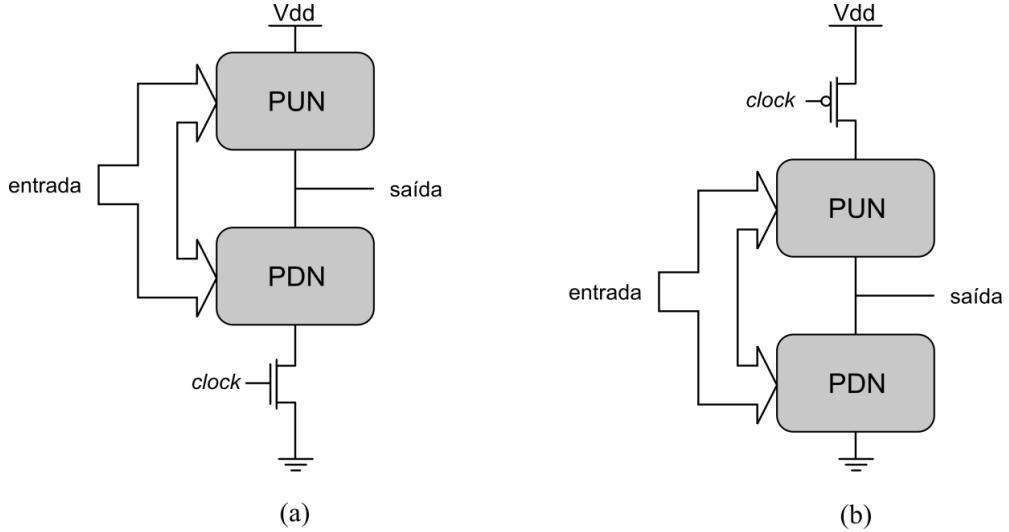


Figura 4: *Latches* utilizados na técnica E-TSPC: (a) n-latch; (b) p-latch

como pode ser visto na Figura 4, está no tipo e posicionamento do transistor no qual o sinal de *clock* está ligado.

É dito que o n-latch é um *latch* sensível a nível alto, pois quando o sinal de *clock* está em nível lógico alto o *latch* está em fase de avaliação, executando a função lógica definida pelas estruturas PUN e PDN. Quando o *clock* está em nível lógico baixo, o n-latch está “armazenando” o valor da saída, essa fase é chamada de fase de *holding*. Quando o transistor ligado ao *clock* está cortado, a transição ‘1’→’0’ da saída não ocorre. No entanto, o n-latch não armazena perfeitamente a saída, tendo em vista que os sinais de entrada não podem ser modificados de modo que a saída realize a transição ‘0’→’1’. O p-latch funciona de modo complementar ao n-latch.

### 2.3. Registradores

Assim como os *latches*, os registradores armazenam um *bit* de informação durante determinado período de tempo, mas eles se diferem dos *latches* no que diz respeito à atualização do valor armazenado, que é realizada apenas durante a subida ou descida do sinal de *clock*. Os registradores são fundamentais para a construção dos circuitos desenvolvidos com a técnica E-TSPC, os RTSs. Existem dois tipos de registradores: os estáticos e os dinâmicos (RABEY, 2006). Aqui são tratados apenas os registradores dinâmicos utilizados no desenvolvimento deste trabalho.

Os registradores dinâmicos mantém a informação armazenada na capacitância parasita vinculada à entrada do bloco ligado à saída do registrador. O tempo durante o qual a informação é armazenada é curto, devido às correntes de fuga. Sua principal vantagem sobre os registradores estáticos é o número reduzido de transistores, fazendo com que os registradores

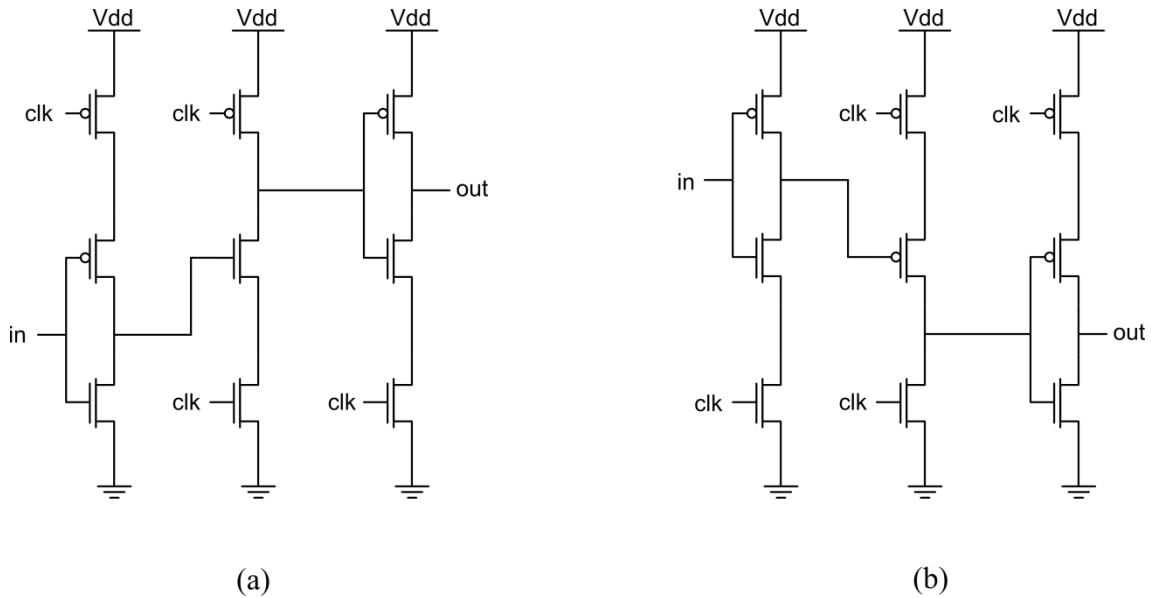


Figura 5: Registradores dinâmicos TSPC: (a) registrador sensível à borda de subida de *clock*; (b) registrador sensível à borda de descida de *clock*

dinâmicos apresentem menor área e maior velocidade. Os registradores utilizados neste trabalho são os registradores clássicos da técnica TSPC, apresentados na Figura 5.

Analizando os registradores dinâmicos da Figura 5 é possível observar que eles são formados por três blocos principais: um *p-latch*, um *n-latch* e um inversor dinâmico. A saída armazenada por esses registradores é igual ao sinal de entrada invertido, já que o sinal é invertido três vezes durante a operação do registrador (uma vez em cada bloco). No caso do registrador sensível à borda de descida, apresentado na Figura 5(a), o primeiro estágio, que é o bloco *p-latch*, se encontra em fase de avaliação quando o *clock* está em nível lógico alto e inverte o sinal de entrada passando-o para a entrada do inversor, que está em fase de pré-carga. O inversor dinâmico garante que a entrada do bloco *n-latch*, que o segue, irá ter apenas transições ‘1’ $\rightarrow$ ‘0’ durante a sua fase de *holding*, isto é, quando o *clock* estiver em nível lógico baixo, o que é necessário para o bloco *n-latch* manter o sinal de saída. O registrador sensível à borda de subida funciona de modo complementar ao registrador sensível à borda de descida. Neste trabalho também são utilizados registradores nos quais o *latch* do primeiro estágio realiza uma função lógica NOR ou NAND, tornando desnecessário o uso de uma porta lógica adicional.

#### 2.4. Blocos NMOS-like

Esses blocos são construídos através de modificação de portas lógicas dinâmicas e de blocos *latches* (NAVARRO, 1999). Nas portas n-dinâmicas e nos n-latches essa modificação é feita removendo a porção de transistores NMOS que trabalham de forma complementar à uma porção de transistores PMOS (isto é, quando uma conduz a outra não conduz e

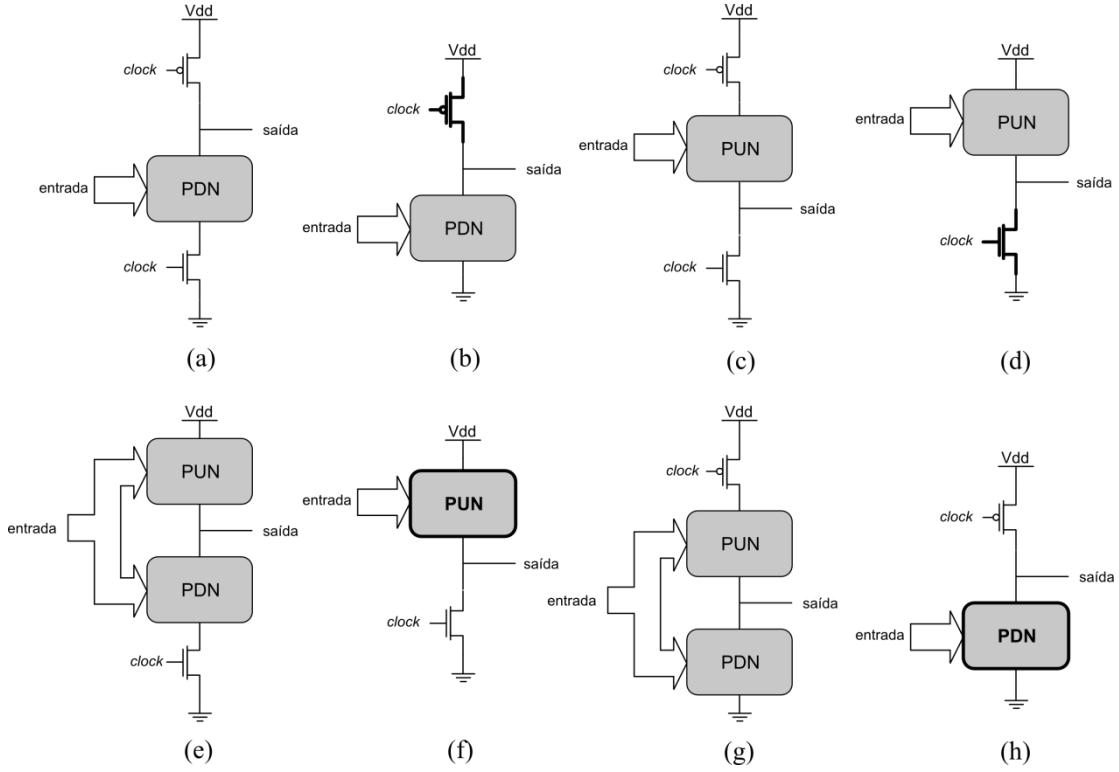


Figura 6: Construção de blocos NMOS-like: (a) bloco n-dinâmico e (b) bloco n-dinâmico NMOS-like; (c) bloco p-dinâmico e (d) bloco p-dinâmico NMOS-like; (e) bloco n-latch e (f) bloco n-latch NMOS-like; (g) bloco p-latch e (h) bloco p-latch NMOS-like

vice-versa). No caso das portas n-dinâmicas esses transistores complementares são os que estão ligados ao *clock* e no caso dos n-latches são as estruturas PDN. As portas p-dinâmicas e os p-latches também podem ser modificados, mas no seu caso são as estruturas formadas por transistores PMOS que são removidas. A Figura 6 ilustra a construção de blocos NMOS-like a partir de portas originais.

A remoção das estruturas complementares aumenta a velocidade dos blocos pois diminui o número de transistores em série. Maiores ganhos de velocidade são observados nos p-latches e portas p-dinâmicas, pois os transistores removidos são do tipo PMOS, que são duas ou mais vezes mais lentos que os NMOS.

É também importante notar que esses blocos não são mais *ratioless*, ou seja, a dimensão dos transistores irá influenciar no funcionamento do circuito. Na Figura 6 é apontada qual porção do bloco deve predominar (elementos destacados nos blocos NMOS-like), de modo que quando esses blocos estiverem conduzindo a saída será levada para a tensão do nó de alimentação ao qual estão ligados, independentemente se a outra porção do bloco está conduzindo ou não.

## 2.5. Políticas de Clock

As políticas de *clock* são regras para construção de RTSs, que definem quais são os blocos que podem ser utilizados, regras para a ligação desses blocos e regras que definem como o sinal de *clock* deve ser. Políticas com múltiplas fases de *clock* permitem uma utilização mais eficiente

do tempo, resultando em circuitos mais rápidos (RABEY, 2006). Porém, para distribuir essas múltiplas fases surgem diversos problemas que se intensificam com o aumento do tamanho dos circuitos. A tendência em circuitos em circuitos de alta velocidade CMOS VLSI (*Very Large Scale Integration*) é a utilização de um único *clock*, simplificando a sua distribuição (RABEY, 2006; WESTE, 1993). Dentre as políticas de *clock*, podem ser citadas a política NORA (*No-Race CMOS Technique*) (GONÇALVES, 1984) e a TSPC.

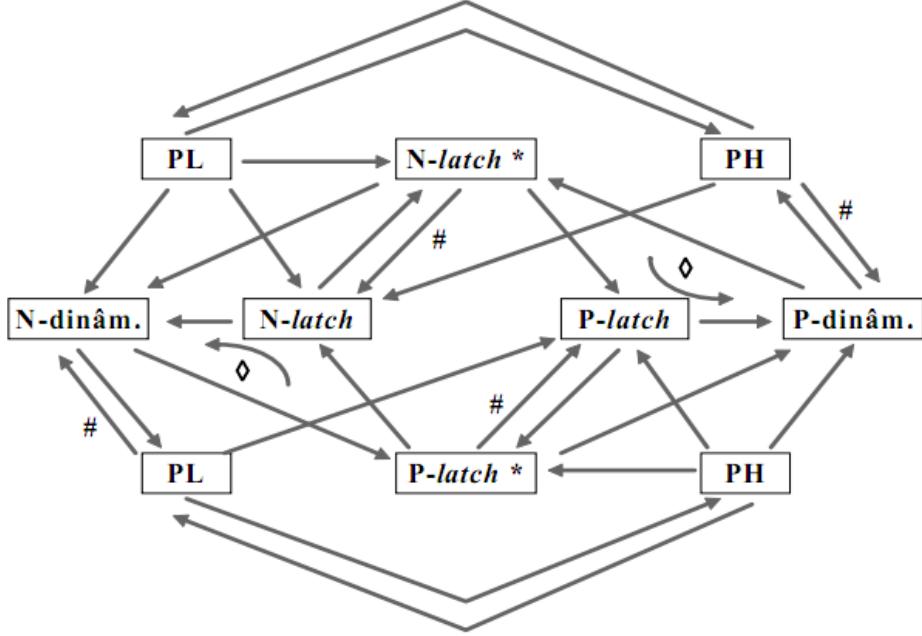


Figura 7: Regras de composição para o TSPC. Os blocos são representados pelos quadrados e o fluxo de dados permitido é representado pelo sentido setas. O # indica ligações onde atrasos diferenciados do *clock* nos blocos envolvidos não causam problemas (*skew safe communication*); O \* indica blocos *latch* depois dos quais portas CMOS estáticas podem ser inseridas; O  $\diamond$  indica ligações onde há problemas potenciais de *holding time* (NAVARRO, 1998)

A política TSPC utiliza uma única fase de *clock* no desenvolvimento dos circuitos (LARSSON, 1995). Para isso, algumas regras de composição de circuitos são utilizadas. A Figura 7 apresenta um diagrama das conexões permitidas pelo TSPC. O diagrama indica ligações onde pode ocorrer problema devido aos atrasos diferenciados do sinal de *clock* que chega a blocos interligados. Esse problema é chamado de *clock skew* e ocorre no caso em que um bloco está em fase de pré-carga e outro bloco ligado à saída do primeiro está em avaliação, gerando por sua vez uma saída incorreta. Para contornar esse problema pode-se controlar o valor dos atrasos dos *clocks* através do projeto de *buffers* (GRONOWSKI, 1998).

O diagrama da Figura 7 também indica ligações onde podem ocorrer problemas de *holding time* (NAVARRO, 1998). Esse problema aparece quando o tempo no qual um p ou n-*latch*, ligado a uma porta p ou n-dinâmica, permanece em *holding* é muito pequeno. Nesse caso a porta dinâmica não tem tempo para avaliar corretamente, retornando à fase de pré-carga antes de apresentar a saída correta.

## 2.6. Técnica E-TSPC

A técnica E-TSPC é uma extensão da técnica TSPC. Ela propõe novas regras de composição e mudanças na topologia de algumas portas lógicas e registradores, a fim de atingir maiores velocidades nos circuitos (NAVARRO, 1998). A técnica E-TSPC utiliza os seguintes blocos para compor os circuitos:

- portas lógicas estáticas CMOS;
- portas lógicas dinâmicas;
- *p-latches* e *n-latches*;
- blocos NMOS-like.

É utilizado também, nessa técnica, o conceito de *data chain* (NAVARRO, 1998). Para entender esse conceito é preciso entender o que é um *signal propagation path* (SPP; em português, caminho de propagação de sinal). Para isso, aqui é chamado de C um circuito digital qualquer. Um SPP de C é um subcircuito que possui as seguintes características:

- possui uma única entrada e uma única saída;
- contém um bloco  $B_{\text{entrada}}$  a partir do qual todos os blocos do SPP são acessíveis (quando se diz que um bloco  $B_1$  é acessível a partir de um bloco  $B_2$ , entende-se que a saída de  $B_1$  está ligada diretamente ou através de um ou mais blocos à entrada de  $B_2$ );
- contém um bloco  $B_{\text{saída}}$  que é acessível a partir de qualquer do SPP;
- a saída de todo bloco do SPP, com exceção do bloco  $B_{\text{saída}}$ , está ligada a apenas um bloco (a saída de  $B_{\text{saída}}$  pode estar ligada ou não a um bloco);
- uma das entradas de  $B_{\text{entrada}}$  é a única entrada do SPP;
- a saída de  $B_{\text{saída}}$  é a única saída do SPP.

Um *data chain* do circuito C é um SPP acíclico, isto é, um SPP que não possui um bloco acessível a partir de si mesmo. Há dois tipos de *data chains*: o *n-data chain* e o *p-data chain*. Em um *n-data chain*, quando o *clock* está em nível lógico alto todos os blocos estão em avaliação e quando o *clock* está em nível lógico baixo parte dos blocos está em *holding* (os *latches*) e parte está em pré-carga (portas Dps e dinâmicas). Além disso, um *n-data chain* possui as seguintes características:

- contém ao menos um *n-latch*, um *n-dinâmico* ou um bloco *n-Dp*;
- sua entrada deve estar ligada a uma entrada externa de C ou à saída de um bloco *p-dinâmico*, *p-Dp* ou *p-latch*;
- contém apenas blocos estáticos *n-latch*, *n-dinâmicos* e *n-Dps*;

- sua saída deve estar ligada à entrada de algum bloco p-dinâmico, p-Dp ou p-latch ou a nenhum outro bloco de C.

O p-*data chain* tem os blocos em avaliação na fase contrária ao n-*data chain*. Ele também possui características equivalentes às características do n-*data chain*, mas em seu caso troca-se as letras “p” por “n” e “n” por “p” em todas as denominações de blocos nas quais aparecem.

A seguir são apresentadas as regras de composição de circuitos E-TSPC. Essas regras definem como os *data chains* devem ser compostos e como ligá-los a outros *data chains*.

### 2.6.1. Regras de Composição

Para o correto funcionamento de um *data chain*, isto é, para que ele avalie as entradas em uma fase do *clock* e armazene a saída na outra, é necessário que ele siga algumas regras. As regras apresentadas aqui são como foram resumidas em (MIRANDA, 2006). Primeiramente, o *data chain* precisa ter uma das duas configurações a seguir (sintetizadas a partir de regras mais complexas do E-TSPC) (NAVARRO, 1998):

- deve possuir ao menos dois blocos, um bloco dinâmico e um *latch*, e um número par de blocos (inversores) entre eles, ou;
- deve possuir ao menos dois *latches* e um número par de blocos (inversores) entre eles;

Adicionalmente, blocos adjacentes no *data chain* necessitam ter um número de blocos inversores como apresentado na Tabela 1 (blocos adjacentes são blocos separados somente por blocos estáticos). Nessa tabela, n.r. significa que não há restrições para o número de blocos e n.p., que não é possível a ligação.

Tabela 1: Regras para ligação de blocos adjacentes em um *data chain*

	<b>Entrada do <i>latch</i></b>	<b>Entrada do n-dinâmico</b>	<b>Entrada do p-dinâmico</b>
<b>Entrada do <i>data chain</i></b>	n.r.	n.r.	n.r.
<b>Saída do <i>latch</i></b>	n.r.	n.p.	n.p.
<b>Saída do n-dinâmico</b>	n.r.	ímpar	n.p.
<b>Saída do p-dinâmico</b>	n.r.	n.p.	ímpar

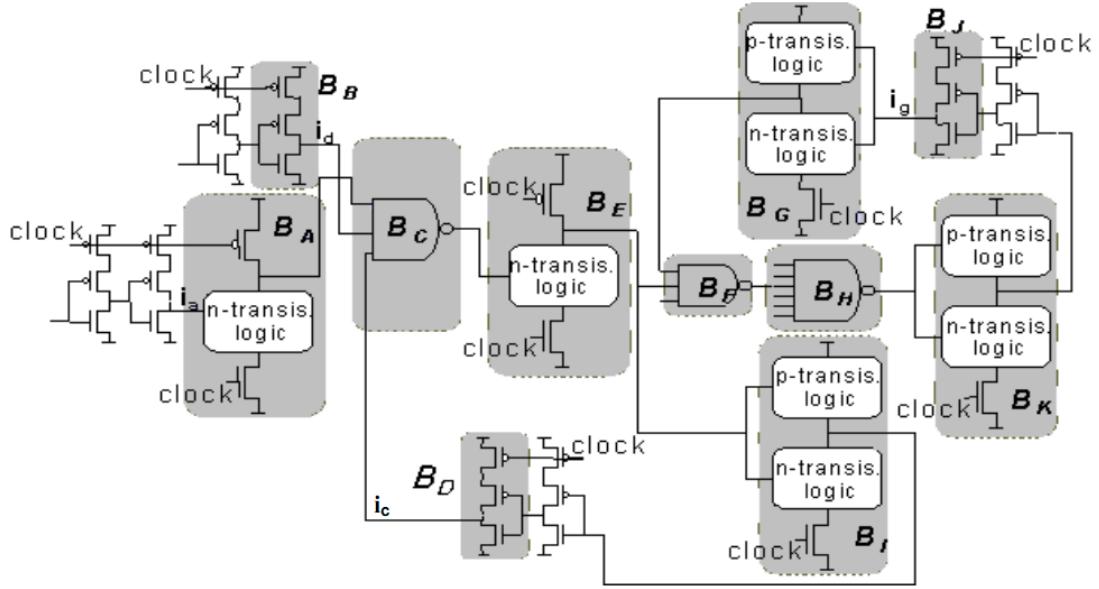


Figura 8: Exemplo de n-data chains (NAVARRO, 2002)

A Figura 8 contém exemplos de n-data chains. Os blocos  $B_A$ ,  $B_C$ ,  $B_E$  e  $B_I$  formam um n-data chain que tem como entrada  $i_a$ . Outro n-data chain é formado pelos blocos  $B_C$ ,  $B_E$ ,  $B_F$ ,  $B_H$  e  $B_K$  e tem como entrada  $i_c$ .

### 2.6.2. Estruturas do E-TSPC

Os data chains, n ou p, que possuem um único latch e que também é o ultimo bloco do data chain possuem uma característica explorada em algumas topologias de circuitos E-TSPC. Essa característica é que durante a fase de holding a saída destes data chains se encontram em alta impedância. A esses data chains é dado o nome de fo-data chains (fusible output data chains; ou data chains de saída fundível) (NAVARRO, 2002).

Como a avaliação de um n-data chain ocorre em uma fase de clock inversa ao da avaliação de um p-data chain, é possível unir as saída de um fo-n-data chain e um fo-p-data chain, assim como exemplificado na Figura 9. Isso é possível pois quando a saída do fo-n-data chain está em

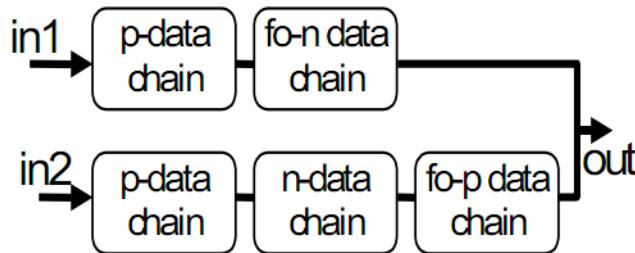


Figura 9: Fusão da saída de dois data chains (NAVARRO, 2002)

alta impedância, a saída do fo-p-*data chain* está ligada ao  $V_{DD}$  ou  $V_{SS}$  (fase de avaliação) e na seguinte fase de operação ocorre o oposto. Dessa forma, as saídas dos dois *data chains* nunca estarão simultaneamente em fase de avaliação.

Aos *data chains*, n ou p, que não possuem blocos dinâmicos ou blocos Dps é dado o nome de fi-*data chains* (*fusible inputs data chains*; ou *data chains* de entrada fundível). Nesses *data chains*, durante a fase de avaliação, variações na entrada não prejudicam o funcionamento do *data chain* (NAVARRO, 2002). É possível fundir a entrada de fi-n-*data chains* e fi-p-*data chains*, assim como mostrado na Figura 10.

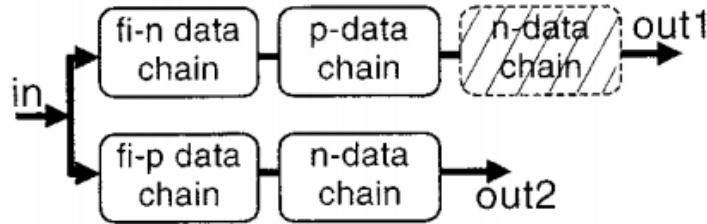


Figura 10: Fusão da entrada de dois *data chains* (NAVARRO, 2002)

O uso dessas estruturas pode aumentar consideravelmente a velocidade de operação dos circuitos, pois os dados da entrada são distribuídos por dois *data chains* e a saída vem de outros dois *data chains*.

### **3.Implementação dos Circuitos**

Para implementar e testar uma biblioteca de células E-TSPC os seguintes passos foram executados:

1. escolher circuitos de testes E-TSPC desenvolvidos com metodologia *full custom*;
2. selecionar células para a biblioteca a partir dos circuitos escolhidos;
3. dimensionar as células;
4. verificar o funcionamento das células através de simulações;
5. implementar as células selecionadas (desenhar e simular os *layouts*);
6. implementar os circuitos escolhidos utilizando a biblioteca de células (desenhar e simular os *layouts*);
7. Simulação dos circuitos *full custom* e dos circuitos implementados com células e comparação do desempenho.

Os circuitos escolhidos para testes foram dois blocos *Dual-Modulus Prescaler* que dividem a frequência do sinal de entrada por 32 ou 33 (MIRANDA, 2006). Eles serão apresentados e detalhados na próxima seção.

Na segunda seção deste capítulo, é realizada a escolha das células que irão compor a biblioteca. Na terceira seção os esquemáticos e *layouts* das células implementadas são apresentados. Na seção final, os esquemáticos, *layouts* e simulações dos circuitos *Prescaler* são apresentados. A comparação entre os circuitos desenvolvidos neste trabalho e os circuitos desenvolvidos em (MIRANDA, 2006), que foram feitos com metodologia *full custom*, é realizada no capítulo seguinte. Também é feita a comparação com um circuito *prescaler* desenvolvido com a biblioteca de células estáticas CMOS disponibilizada pela AMS.

#### **3.1. *Dual-Modulus Prescalers* Escolhidos**

*Prescalers* são circuitos que dividem a freqüência de um sinal de entrada e apresentam em sua saída o sinal resultante dessa divisão. No caso dos circuitos *Prescaler* escolhidos, cujos esquemáticos são apresentados na Figura 11 e na Figura 12, um sinal de controle **SM** seleciona se a frequência do sinal de entrada **clk** será dividida por 32 ou 33, por esse motivo eles são chamados de *Dual-Modulus Prescaler*.

Os *Prescalers* escolhidos são formados por duas partes principais, uma de mais alta velocidade (parte síncrona do circuito), à qual o sinal de entrada **clk** está ligado, e uma que trabalha em velocidade menor (parte assíncrona do circuito), cuja entrada é ligada à saída da parte mais rápida.

No caso do *Prescaler* apresentado na Figura 11, a parte síncrona do circuito é um divisor que divide a frequência da entrada **clk** por 4 ou 5, escolhido de acordo com o valor do sinal **MC**. Essa parte é formada por 3 D-FFs (*Flip Flops* do tipo D) sensíveis à borda de descida, por esse motivo essa versão do *Prescaler* é chamada daqui em diante de versão com 3 D-FFs. A parte assíncrona do circuito é formada por 3 D-FFs sensíveis à borda de subida, que dividem a frequência do sinal **A** (saída da parte síncrona) por 8.

Na versão do *Prescaler* com 3 D-FFs, se o sinal de controle **SM** estiver no nível lógico baixo teremos que o sinal **MC** permanecerá no nível lógico alto, a parte síncrona do circuito irá contar sempre 4 e, consequentemente, a frequência da entrada **clk** do *Prescaler* será dividida por 32 ( $8 \cdot 4$ ). Por outro lado, caso o sinal de controle **SM** esteja no nível lógico alto, em certo momento (uma vez a cada oito períodos do sinal **A**) todas as entradas da porta NAND estarão em nível lógico alto, fazendo com que a saída dessa porta (sinal **MC**) vá para o nível lógico baixo. Isso faz com que a parte síncrona conte 5 ao invés de 4 e, consequentemente, com que a entrada **clk** do *Prescaler* seja dividida por 33 ( $7 \cdot 4 + 5$ ). Esse é o caso mais crítico do funcionamento do circuito, pois a parte síncrona precisa contar a entrada 5 vezes, exigindo mais velocidade de processamento dessa parte do circuito.

Na versão com 2 D-FFs do *Prescaler* (Figura 12), a parte síncrona do circuito é feita com 2 D-FFs sensíveis à borda de descida. Essa parte divide a freqüência de **clk** por 2 ou 3, dependendo do valor de **MC**. A parte assíncrona é feita com 4 D-FFs sensíveis à borda de descida, que dividem a saída da parte síncrona (sinal **B**) por 16. De modo semelhante ao que acontece na versão com 3 D-FFs, se o sinal de controle **SM** estiver no nível lógico baixo teremos que o sinal **MC** permanecerá no nível lógico alto, a parte síncrona do circuito irá contar sempre 2 e, consequentemente, a freqüência da entrada **clk** do *Prescaler* será dividida por 32 ( $16 \cdot 2$ ). Por outro lado, caso o sinal de controle **SM** esteja no nível lógico alto, em certo momento (uma vez a cada 16 períodos do sinal **B**) todas as entradas da porta NAND estarão em nível lógico alto, fazendo com que a saída dessa porta (sinal **MC**) vá para o nível lógico baixo. Isso faz com que a parte síncrona conte 3 ao invés de 2 e, consequentemente, com que a entrada **clk** do *Prescaler* seja dividida por 33 ( $15 \cdot 2 + 3$ ).

Em ambas as versões do *Prescaler*, a parte síncrona é constituída de blocos NMOS-*like*. Isso é feito porque essa parte do circuito precisa funcionar em mais alta velocidade, já que está ligada diretamente ao sinal de entrada **clk**. As dimensões dos transistores utilizados estão apresentadas na Figura 11 e na Figura 12.

Circuitos *Dual-Modulus Prescalers* são empregados em sintetizadores de freqüência. No Apêndice B é apresentado um sintetizador de arquitetura *integer N* que emprega *prescalers*.

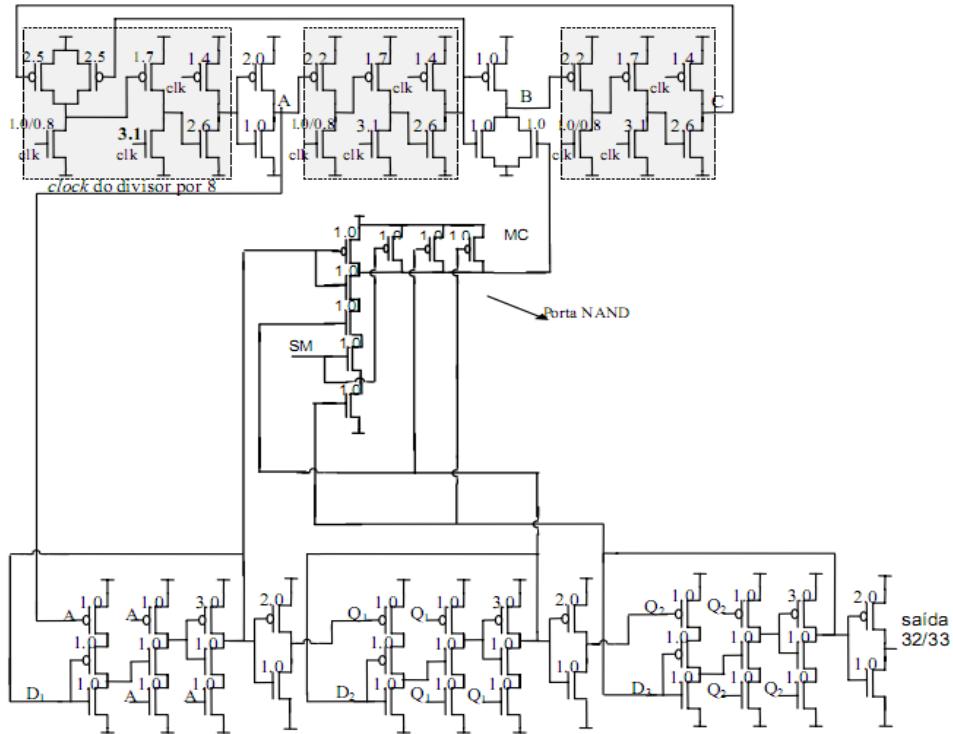


Figura 11: Esquemático do *Dual-Modulus Prescaler 32/33* (versão com 3 D-FFs) (MIRANDA, 2006). A largura de canal dos transistores, W, em  $\mu\text{m}$  está indicada na figura. Para a maior parte dos transistores, o comprimento de canal, L, é  $0,35\ \mu\text{m}$ ; para alguns transistores, aqueles em que aparecem duas dimensões, é dada a relação W/L

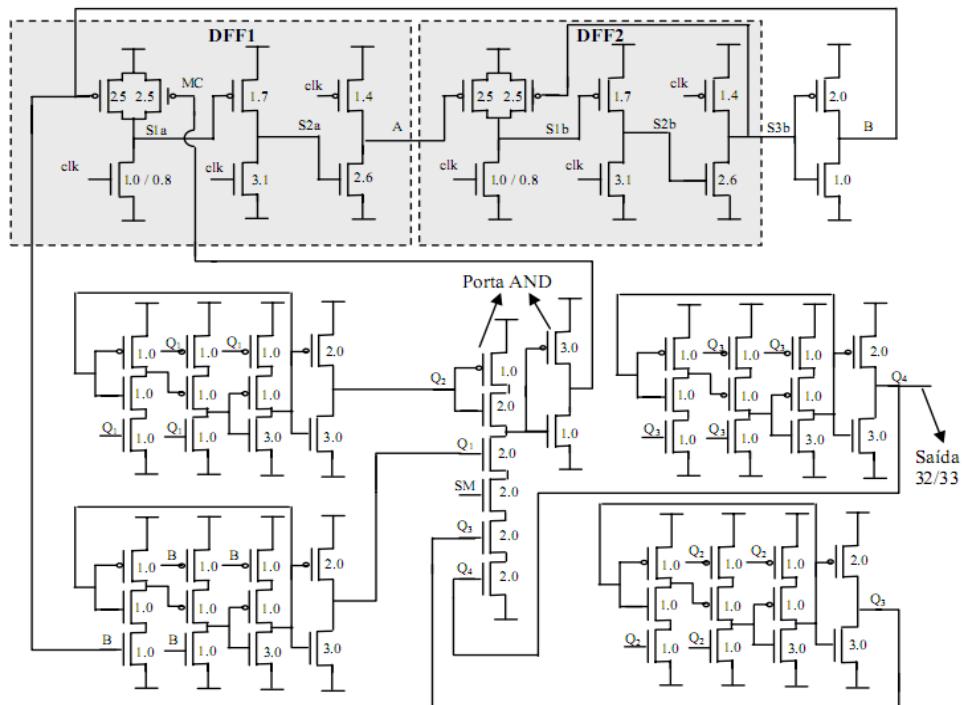


Figura 12: Esquemático do *Dual-Modulus Prescaler 32/33* (versão com 2 D-FFs) (MIRANDA, 2006). A largura de canal dos transistores, W, em  $\mu\text{m}$  está indicada na figura. Para a maior parte dos transistores, o comprimento de canal, L, é  $0,35\ \mu\text{m}$ ; para alguns transistores, aqueles em que aparecem duas dimensões, é dada a relação W/L

### 3.2. Células Escolhidas para a Biblioteca

A partir da analise dos circuitos *prescaler* da Figura 11 e Figura 12, escolheram-se inicialmente as seguintes células para compor a biblioteca

- C1. D-FF convencional sensível à borda de subida;
- C2. *latch* convencional sensível a nível lógico alto;
- C3. p-*latch* convencional;
- C4. n-*latch* NMOS-*like* com lógica NAND de duas entradas;
- C5. p-*latch* NMOS-*like* com lógica NOR de duas entradas;
- C6. D-FF convencional sensível à borda de descida;
- C7. *latch* convencional sensível a nível lógico baixo;
- C8. n-*latch* convencional;
- C9. D-FF NMOS-*like* sensível à borda de descida;
- C10. *latch* NMOS-*like* sensível a nível lógico baixo;
- C11. n-*latch* NMOS-*like*.

As células denominadas “convencionais” se referem às células como apresentadas na técnica TSPC, sem qualquer modificação. As células **C2** e **C3** são partes da célula **C1**, as células **C7** e **C8** são partes da célula **C6** e as células **C10** e **C11** são partes da célula **C9**. As células **C1**, **C6** e **C9** são implementadas por completo, isto é, não são feitas unindo sub-células, pois assim é possível obter uma implementação mais eficiente em termos de área e velocidade.

Nem todas as células listadas acima foram implementadas neste trabalho devido ao tempo disponível e porque nem todas são necessárias para o desenvolvimento dos *Prescalers*. As células necessárias nos *prescalers* são as células **C1**, **C4**, **C5**, **C6**, **C9** e **C10**.

As dimensões dos transistores foram obtidas a partir do esquemático dos *Prescaler* apresentados em (MIRANDA, 2006). Algumas dessas dimensões foram modificadas a fim de melhorar o desempenho das células, isto é, aumentar a freqüência de operação.

### 3.3. Desenvolvimento da Biblioteca de Células

Aqui são apresentados os esquemáticos das células da biblioteca, os *layouts* desenvolvidos e as simulações individuais das células. Antes de desenhar as células, cada uma delas foi simulada utilizando o *software* Eldo, a fim de testar o dimensionamento dos transistores. Em alguns casos as dimensões de transistores foram modificadas. Essas modificações, quando houve, são discutidas juntamente com a apresentação do funcionamento da célula.

Os seguintes passos foram executados para, a partir das dimensões, desenhar cada célula:

1. Desenho do esquemático da célula no *software* Design Architecture;
2. Desenho do *layout* com o *software* IC Station e auxilio do esquemático;

3. Verificação das regras de desenho no *layout* (DRC- *Design Rule Check*) com o *software* Calibre;
4. Comparação do *layout* com o esquemático (LVS - *Layout Versus Schematic*), para saber se o *layout* está correto do ponto de vista lógico, com IC Station;
5. Extração de um arquivo para simulação a partir do *layout* com o *software* Calibre;
6. Simulação do arquivo extraído do *layout* com o *software* Eldo;
7. Analise dos resultados das simulações com *software* EZwave.

Os *layouts* das células desenhadas devem possuir algumas características específicas para que as células desenvolvidas possam ser utilizadas junto às células da biblioteca padrão disponibilizadas pela AMS e para poder usar recursos de *placement* e roteamento automáticos (um exemplo de célula da biblioteca padrão pode ser visto na Figura 13). Essas características são as seguintes:

- trilhas de  $V_{DD}$  e  $V_{SS}$  com largura mínima de 1,8  $\mu\text{m}$ ;
- trilha de  $V_{DD}$  deve ser posicionada na parte superior do *layout* e trilha de  $V_{SS}$ , na parte inferior;
- altura efetiva da célula deve ser igual à 13  $\mu\text{m}$  (é chamada de altura efetiva a distância entre a borda superior da trilha de  $V_{DD}$  e a borda inferior da trilha de  $V_{SS}$ );
- ao criar o *layout* da célula deve-se selecionar na ferramenta IC Station, como tipo do *layout*, a opção *standard cell* (a opção padrão é *block*).
- área efetiva da célula deve ser sobreposta com as camadas de *layout* **CELBOX** e **fp1**.

As camadas **CELBOX** e **fp1** indicam a área efetiva da célula. Isto é importante para que as ferramentas de *placement* possam manipular as células corretamente, não as sobrepondo, não as mantendo muito distantes e posicionando-as corretamente sobre os *rows* (linhas utilizadas pelo IC Station para organizar a colocação de células).

Nos esquemáticos das células, apresentados nas subseções seguintes, a letra W representa o a largura do canal do transistor e a letra L o comprimento do canal. Nos esquemáticos, as dimensões são dadas em micrômetros.

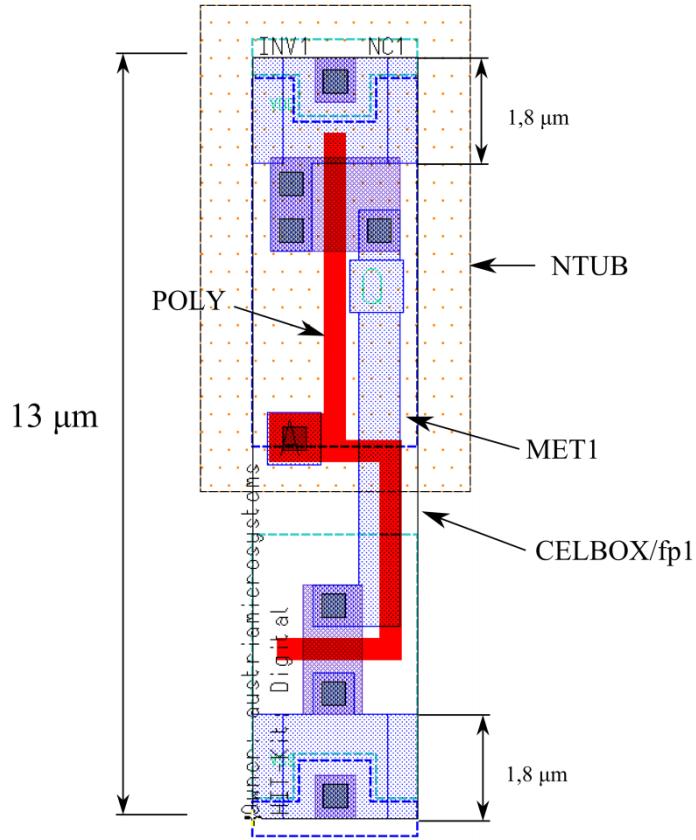


Figura 13: Exemplo de *layout* da célula inversora da biblioteca padrão da AMS

### 3.3.1. D-FF Convencional Sensível à Borda de Subida

A célula apresentada aqui é um D-FF convencional TSPC sensível à borda de subida. Ela é utilizada na parte assíncrona da versão do *Prescaler* com 3 D-FFs. A área total dessa célula é de  $186 \mu\text{m}^2$ . As Figuras 14 e 15 apresentam o esquemático e o *layout* dessa célula, respectivamente.

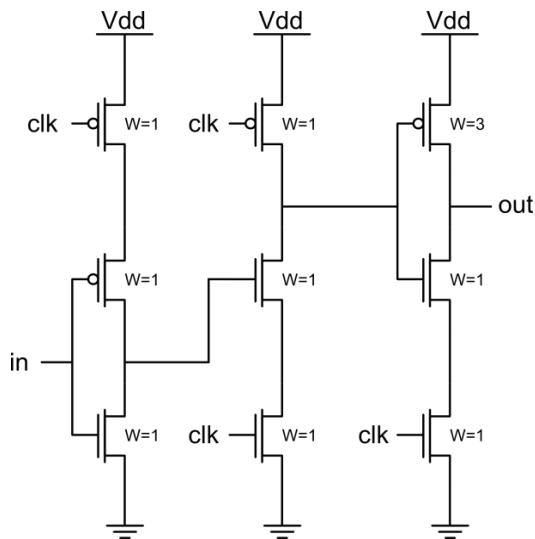


Figura 14: Esquemático do D-FF convencional sensível à borda de subida. A largura de canal dos transistores, W, em  $\mu\text{m}$  está indicada na figura; o comprimento de canal, L, é  $0,35 \mu\text{m}$

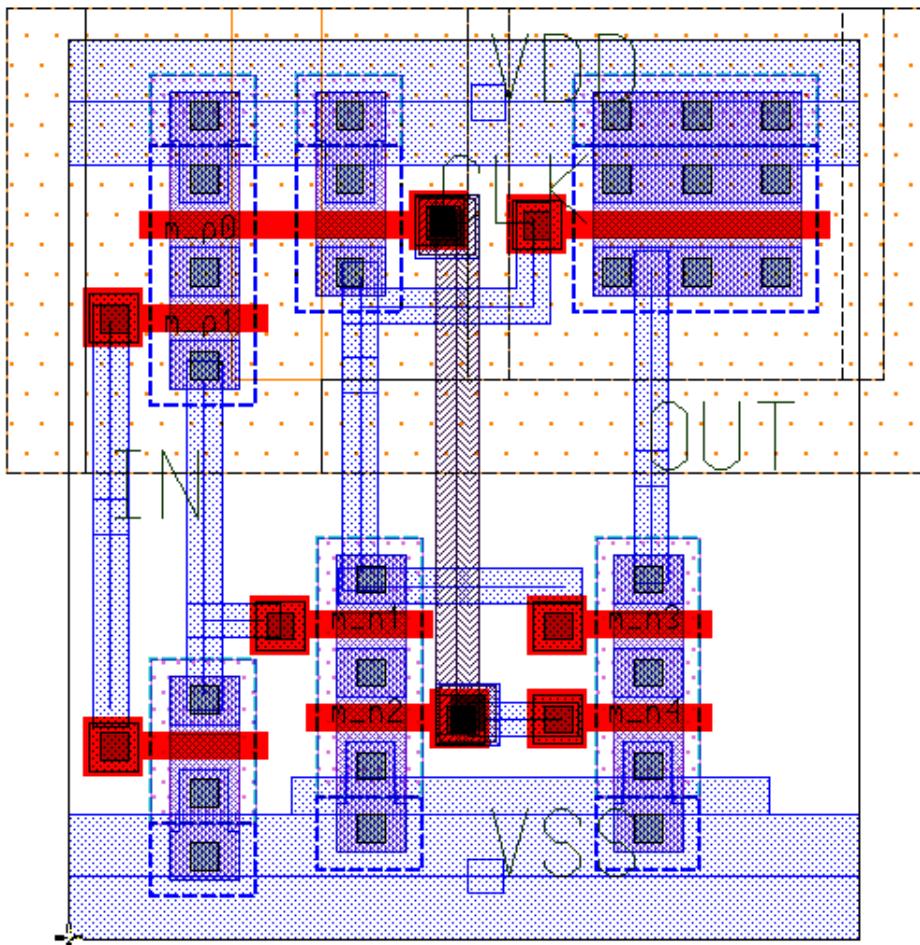


Figura 15: *Layout* do D-FF convencional sensível à borda de subida

### 3.3.2. N-Latch NMOS-*Like* com Lógica NAND de duas entradas

A célula apresentada aqui é um n-Latch NMOS-*Like* com lógica NAND de duas entradas. Ela é utilizada na parte síncrona de ambos os *Prescalers* implementados. A largura de canal dos transistores PMOS foi modificada de 2,5  $\mu\text{m}$  para 3  $\mu\text{m}$ . Isso diminui o tempo de subida da saída, no entanto diminui também a tensão na saída quando ela estiver em nível lógico alto. A área total dessa célula é de 105  $\mu\text{m}^2$ . As Figuras 16 e 17 apresentam o esquemático e o *layout* dessa célula, respectivamente.

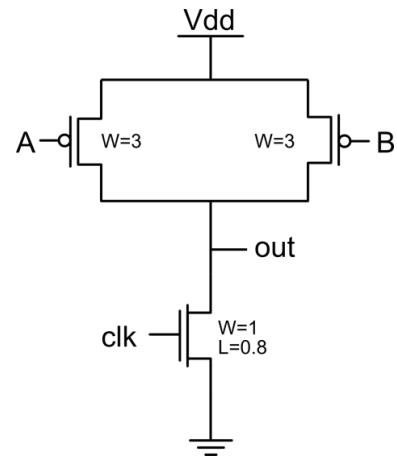


Figura 16: Esquemático do  $n$ -latch com lógica NAND de 2 entradas. A largura de canal dos transistores,  $W$ , em  $\mu\text{m}$  está indicada na figura; o comprimento de canal,  $L$ , é  $0,35 \mu\text{m}$  para os transistores PMOS é  $0,8 \mu\text{m}$  para o transistor NMOS

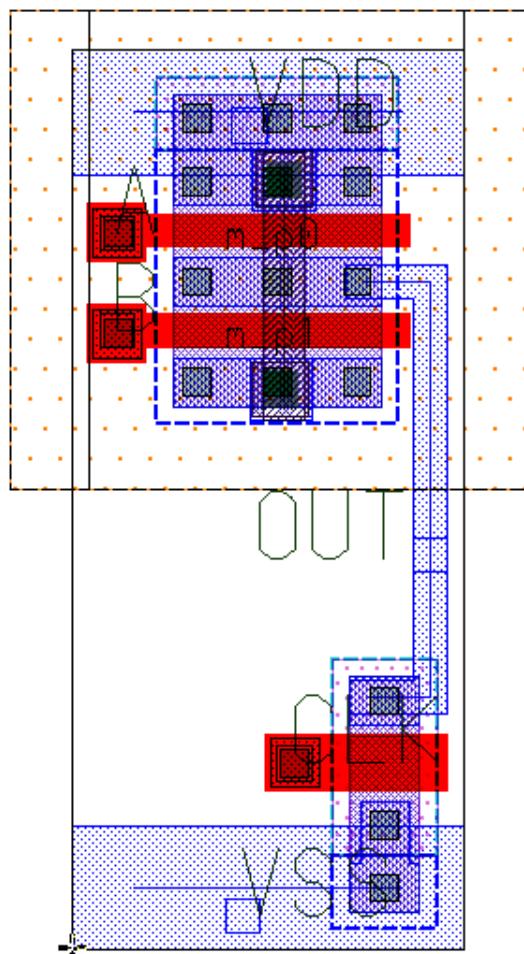


Figura 17: Layout do  $n$ -latch com lógica NAND de duas entradas

### 3.3.3. P-Latch NMOS-Like com Lógica NOR de duas entradas

A célula apresentada aqui é um *p-Latch NMOS-Like* com lógica NOR de duas entradas. Ela é utilizada na parte síncrona na versão com 3 D-FFs do *Prescaler*. A saída desta célula é conectada à entrada de um *latch NMOS-like* sensível a nível lógico baixo (célula C10) para formar um *flip flop* sensível à borda de descida. A área total dessa célula é de  $90 \mu\text{m}^2$ . As Figuras 18 e 19 apresentam o esquemático e o *layout* dessa célula, respectivamente.

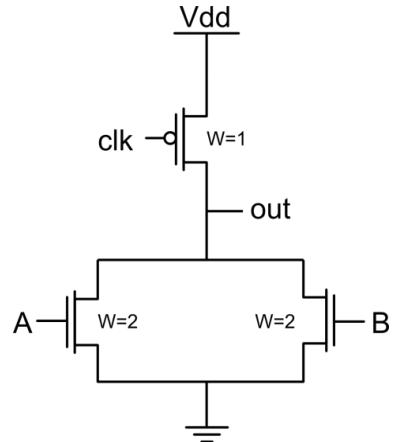


Figura 18: Esquemático do p-latch NMOS-like com lógica NOR de 2 entradas. A largura de canal dos transistores, W, em  $\mu\text{m}$  esta indicada na figura; o comprimento de canal, L, é  $0,35\ \mu\text{m}$

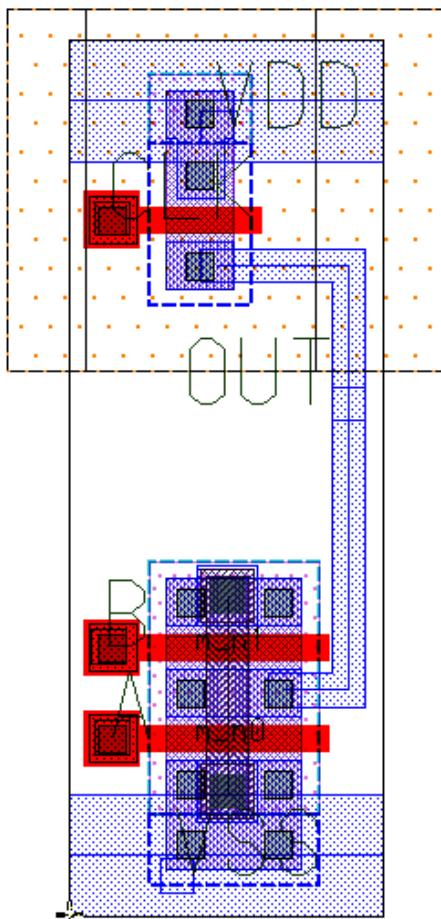


Figura 19: Layout do p-latch NMOS-like com lógica NOR de 2 entradas

### 3.3.4. D-FF Convencional Sensível à Borda de Descida

A célula apresentada aqui é um D-FF convencional TSPC sensível à borda de descida. Ela é utilizado na parte assíncrona na versão do *Prescaler* com 2 D-FFs. A área total dessa célula é de  $186 \mu\text{m}^2$ . As Figuras 20 e 21 apresentam o esquemático e o *layout* dessa célula, respectivamente.

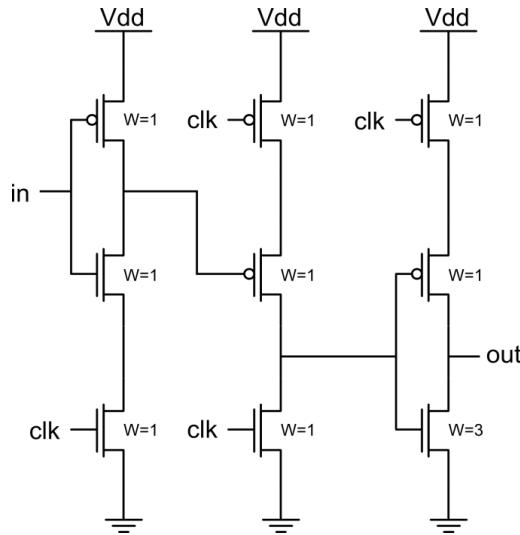


Figura 20: Esquemático do D-FF convencional sensível à borda de descida. A largura de canal dos transistores, W, em  $\mu\text{m}$  está indicada na figura; o comprimento de canal, L, é  $0,35 \mu\text{m}$

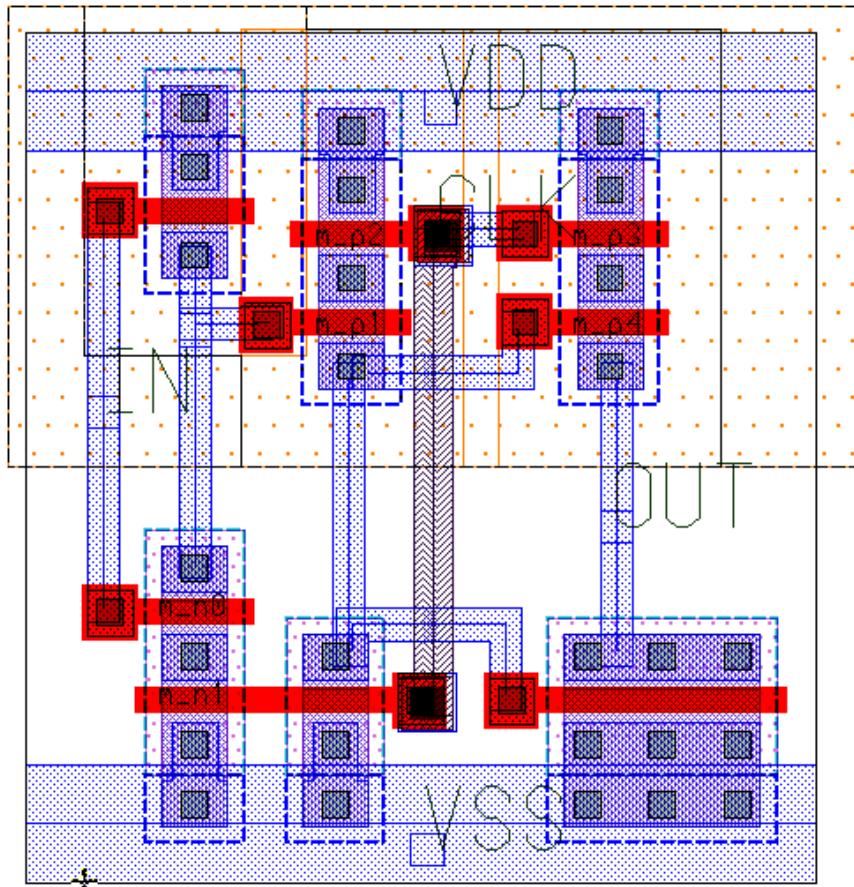


Figura 21: *Layout* do D-FF convencional sensível à borda de descida

### 3.3.5. D-FF NMOS-Like Sensível à Borda de Descida

A célula apresentada aqui é um D-FF modificado sensível à borda de descida. Ela é utilizada na parte assíncrona na versão do *Prescaler* com 3 D-FFs. A área total dessa célula é de  $199 \mu\text{m}^2$ . As Figuras 22 e 23 apresentam o esquemático e o *layout* dessa célula, respectivamente.

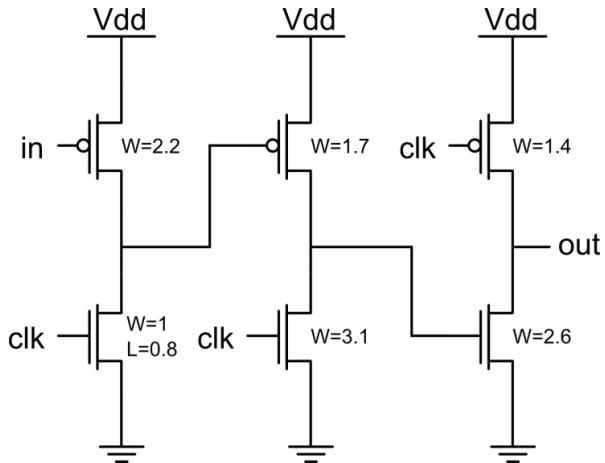


Figura 22: Esquemático do D-FF NMOS-*like* sensível à borda de descida. A largura de canal dos transistores,  $W$ , em  $\mu\text{m}$  está indicada na figura; o comprimento de canal,  $L$ , é  $0,35 \mu\text{m}$  para a maior parte dos transistores, exceção do transistor NMOS mais a esquerda onde é  $0,8 \mu\text{m}$

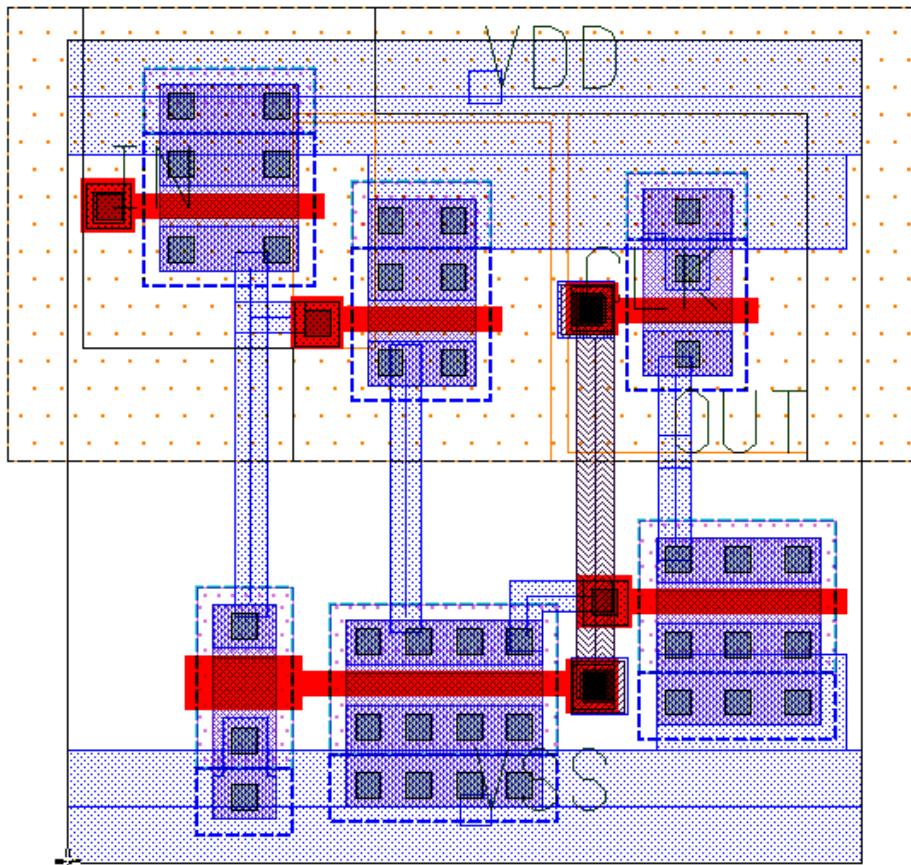


Figura 23: *Layout* do D-FF NMOS-*like* sensível à borda de descida

### 3.3.6. Latch NMOS-Like Sensível a Nível Lógico Baixo

A célula apresentada aqui é um *Latch NMOS-Like* sensível a nível lógico baixo. Ela é utilizada na parte síncrona dos dois *Prescalers* implementados. Seu *layout* foi desenhado a partir de alterações no *layout* do D-FF NMOS-like sensível à borda de descida (circuito C9). A área total dessa célula é de  $150 \mu\text{m}^2$ . As Figuras 24 e 25 apresentam o esquemático e o *layout* dessa célula, respectivamente.

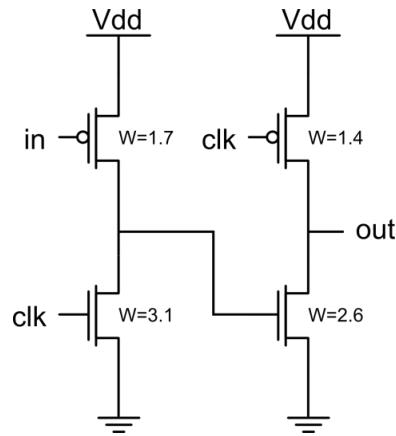


Figura 24: Esquemático do *latch NMOS-like* sensível a nível lógico baixo. A largura de canal dos transistores,  $W$ , em  $\mu\text{m}$  está indicada na figura; o comprimento de canal,  $L$ , é  $0,35 \mu\text{m}$

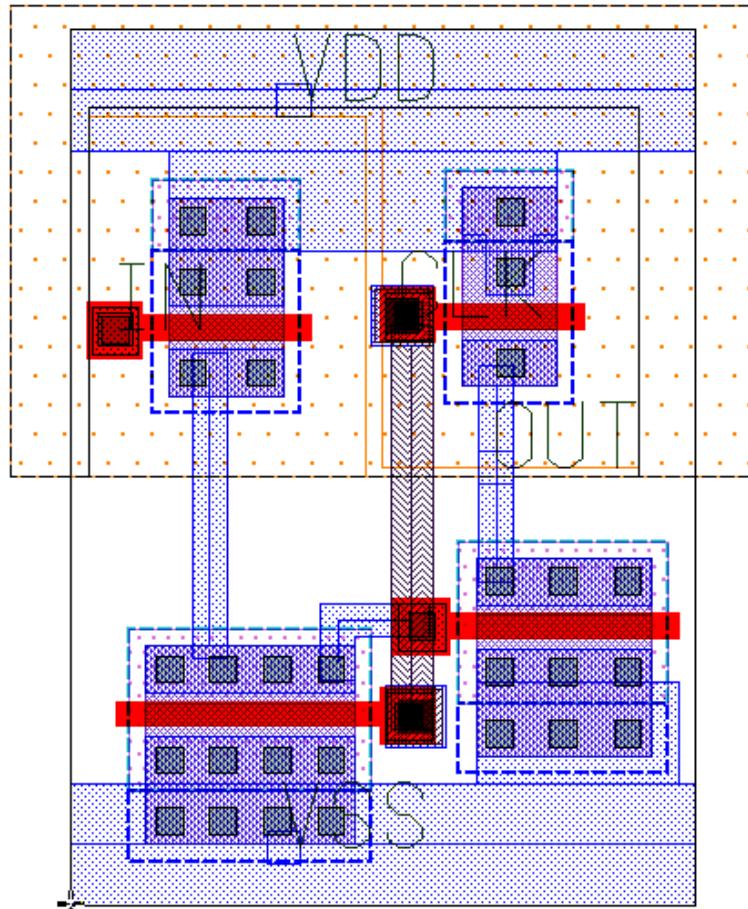


Figura 25: *Layout* do *latch NMOS-like* sensível a nível lógico baixo

### 3.4. Implementação dos Circuitos *Prescalers*

Com base nas células da biblioteca proposta, os esquemáticos dos dois *Prescalers* foram desenhados (Figura 26) e posteriormente desenhados os *layouts* completos. Em ambos os *Prescalers*, o *layout* da parte assíncrona e o *layout* da parte síncrona foram desenhados separadamente e, em seguida, unidos em um único *layout*. Os *layouts* finais dos *Prescalers* podem ser vistos na Figura 28 e na Figura 29.

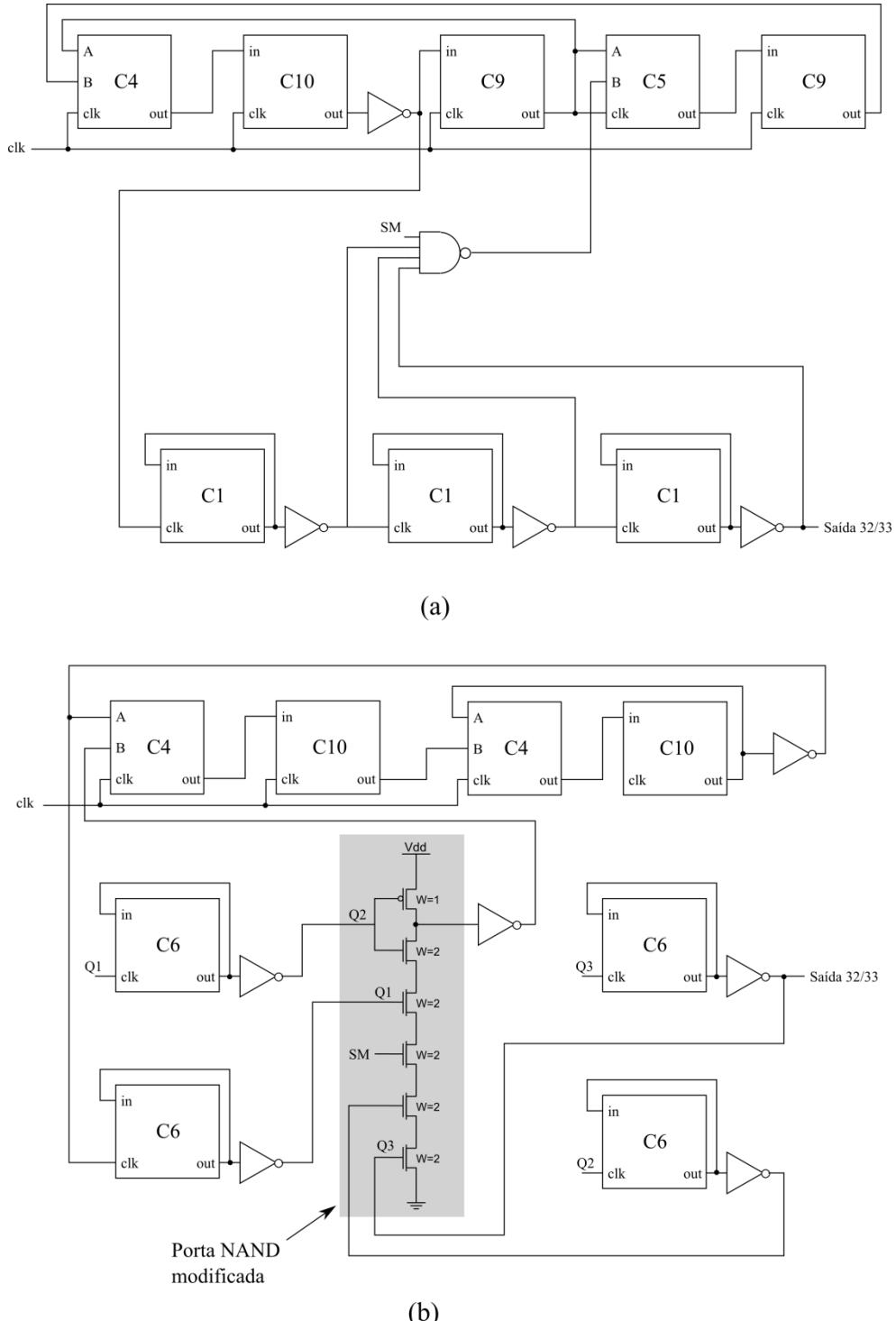


Figura 26: Esquemáticos dos *Prescalers* utilizando células da biblioteca proposta: (a) versão do *Prescaler* com 3 D-FFs; (b) versão do *Prescaler* com 2 D-FFs

Os inversores utilizados nesses *layouts* são inversores da biblioteca padrão disponibilizada pela AMS, assim como a porta NAND de 4 entradas utilizada na versão com 3 D-FFs do *Prescaler*. Os transistores dessas portas possuem dimensões próximas às dimensões mostradas nos esquemáticos da Figura 11 e da Figura 12. A porta lógica NAND modificada utilizada na versão com 2 D-FFs do *Prescaler* foi desenhada de modo *full custom*.

Para comparação, os *layouts* desenvolvidos para os *Prescalers* com metodologia *full custom* em (MIRANDA, 2006) são apresentados na Figura 27. Na Figura 27a um *guard ring* (anel de guarda) é colocado ao redor da parte síncrona do circuito, de modo que o ruído gerado por essa porção rápida do circuito não passe para outros circuitos.

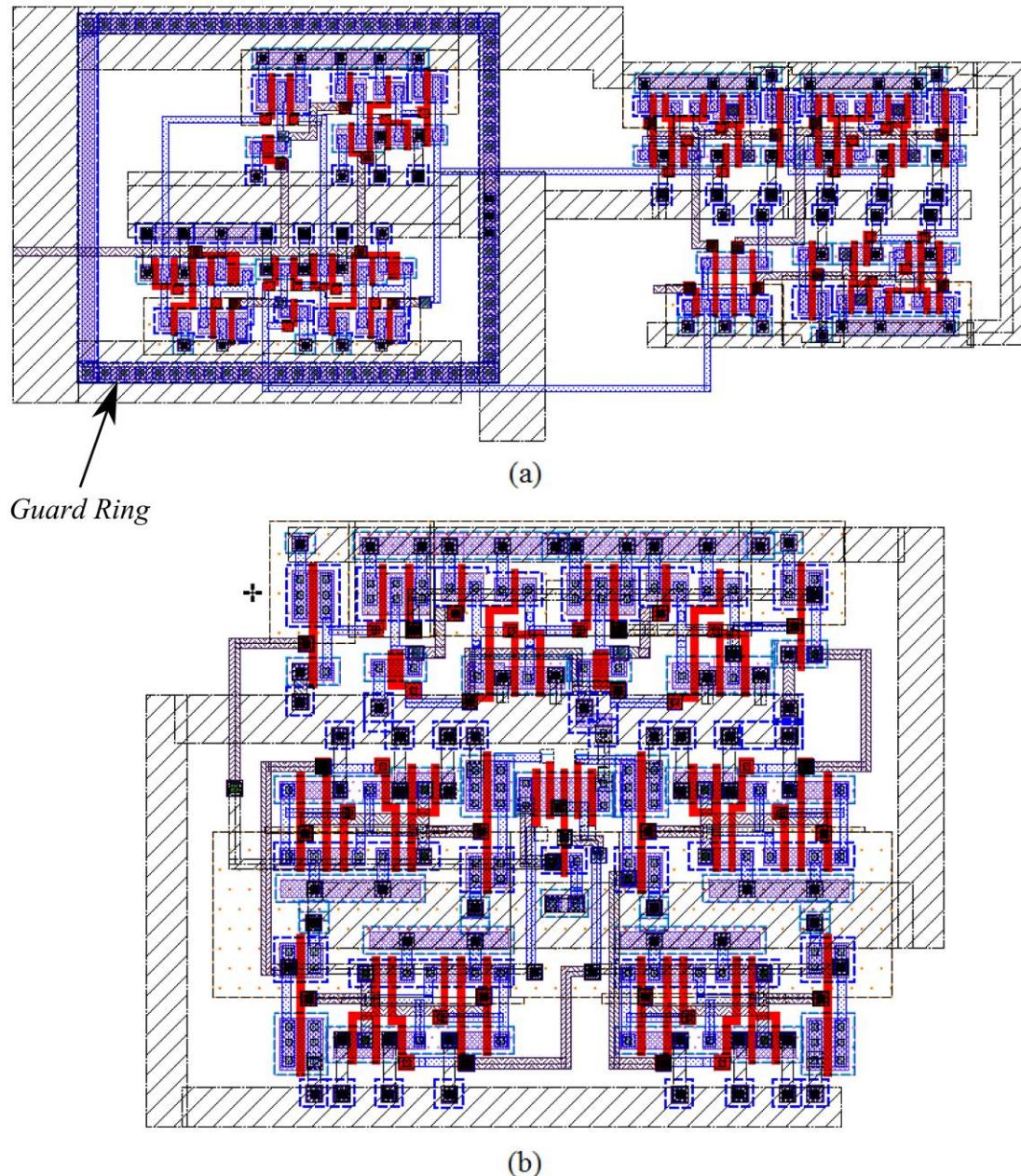


Figura 27: *Layout* dos *Prescalers* desenvolvidos em (MIRANDA, 2006): (a) versão do *Prescaler* com 3 D-FFs, área de  $2848 \mu\text{m}^2$ ; (b) versão do *Prescaler* com 2 D-FFs, área de  $1787 \mu\text{m}^2$

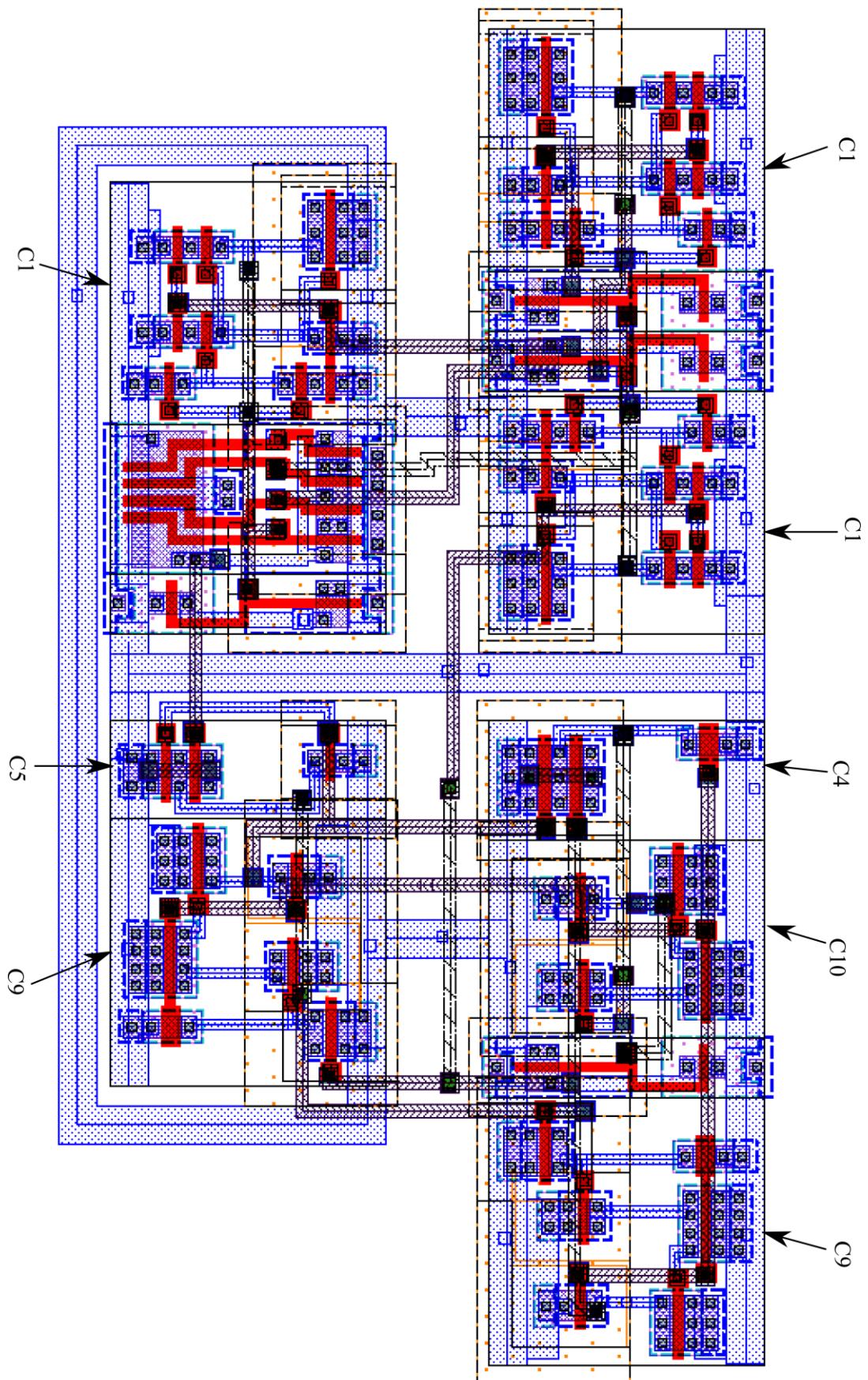


Figura 28: *Layout* da versão do *Prescaler* com 3 D-FFs desenvolvido com metodologia *standard cell*,  $3080 \mu\text{m}^2$ .

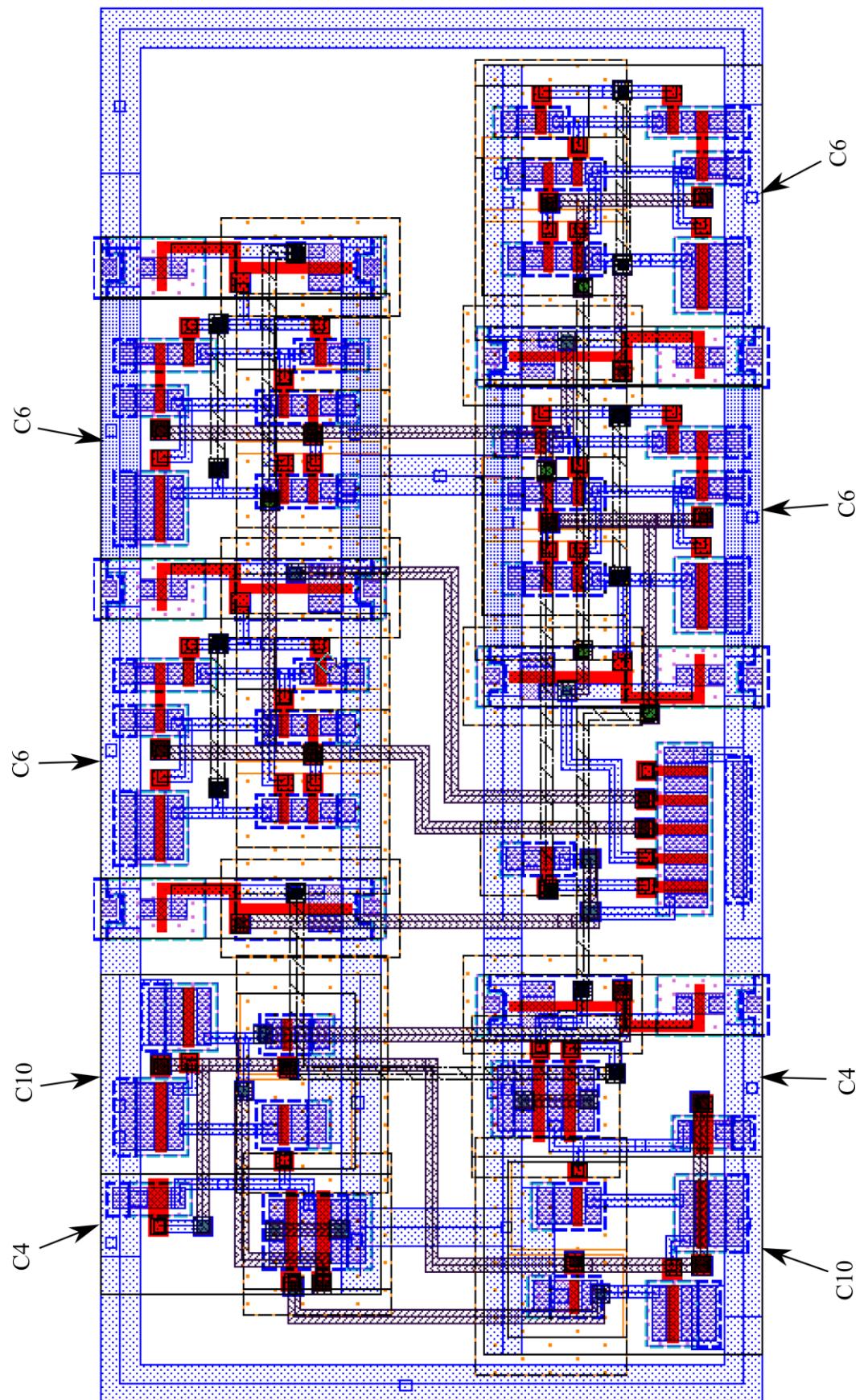


Figura 29: *Layout* da versão do *Prescaler* com 2 D-FFs desenvolvido com metodologia *standard cell*, área de  $2247 \mu\text{m}^2$

Os *layouts* foram extraídos para simulação utilizando a opção R+C+CC do *software* Calibre, que considera as resistências e capacitâncias das linhas utilizadas no roteamento do circuito. As simulações realizadas com o ELDO utilizaram o modelo BSIM3v3, parâmetros típicos (Apêndice A), tensão de alimentação,  $V_{DD}$ , igual a 3V. A Figura 30 apresenta a forma de onda resultante que aparece na saída dos circuitos desenvolvidos. A linha pontilhada é o sinal de entrada **clk**, uma onda quadrada com tempo de subida e descida igual a 10% do período da onda, e a linha contínua é o sinal de saída. O sinal **clk** utilizado para gerar a saída da Figura 30a tem frequência de 2 GHz e o utilizado para gerar a saída da Figura 30b tem frequência de 1,5 GHz. Os gráficos da Figura 30 foram extraídos do *software* EZwave. As frequências de **clk** utilizadas aqui estão dentro dos limites de operação dos circuitos, porém não são as frequências máximas de operação, que serão discutidas no próximo capítulo.

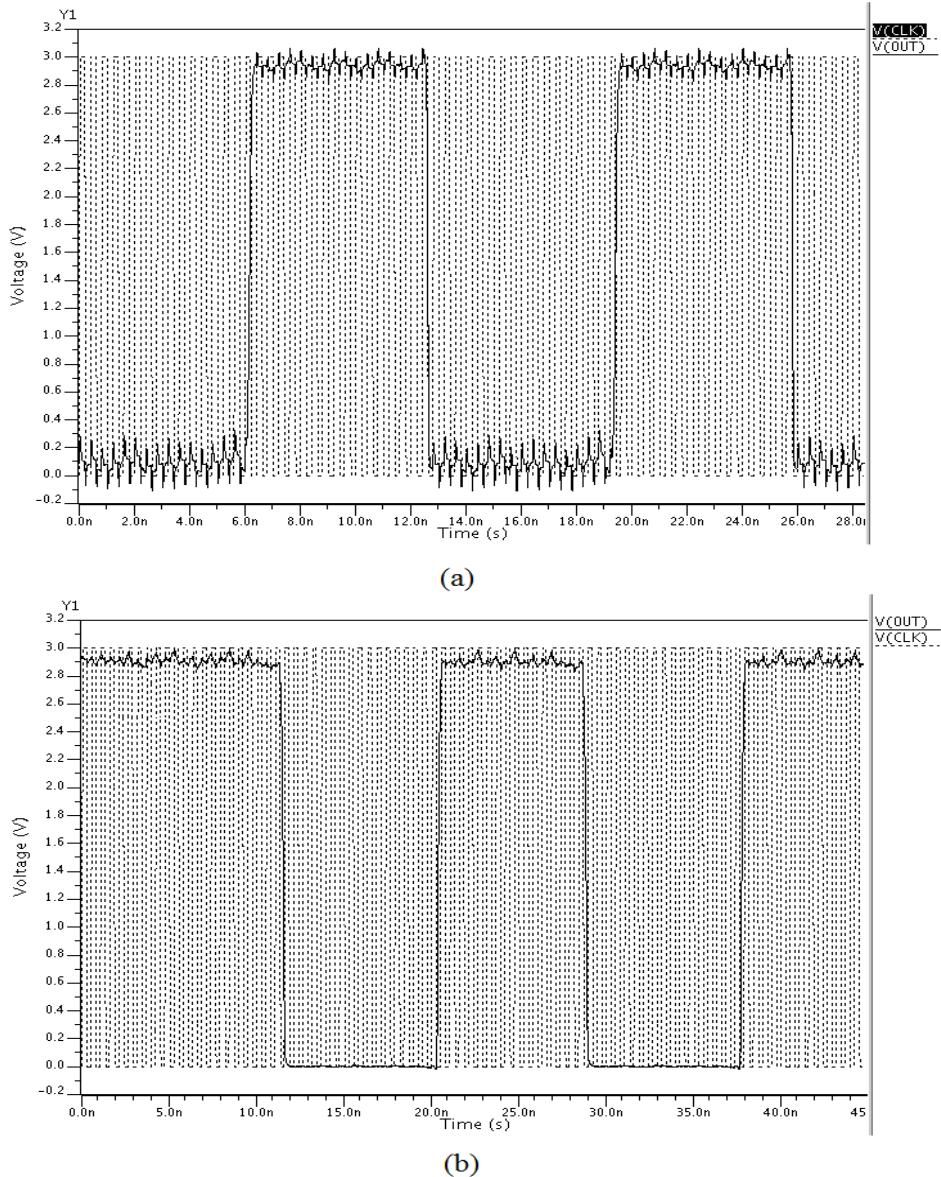


Figura 30: Forma de onda apresentada pelos circuitos desenvolvidos: (a) versão do *Prescaler* com 3 D-FFs; (b) versão do *Prescaler* com 2 D-FFs. O sinal **out** é a Saída32/33

## 4.Resultados e Comparações

Para cada um dos *layouts* dos *Prescalers* apresentados no capítulo anterior, realizaram-se simulações a partir de arquivos extraídos do *layout*, opção R+C+CC, utilizando-se o modelo BSIM3v3 típico e com tensões de alimentação,  $V_{DD}$ , iguais a 3,0V, 2,5V e 2,0V. Para cada simulação encontrou-se a máxima frequência de operação (maior frequência do sinal quadrado na entrada **clk**, com tempo de subida e descida de 10% do período do sinal, para o qual o circuito funciona corretamente),  $f_{MAX}$ , e a média da corrente consumida (corrente que passa por  $V_{DD}$ ) em dois ciclos do sinal de saída,  $I_{AVG}$ . A partir da média da corrente obteve-se a potência consumida por Giga-hertz utilizando a Equação 1.

$$\frac{P}{f} = \frac{I_{AVG} \cdot V_{DD}}{f_{MAX}} \quad (1)$$

A Tabela 2 apresenta os resultados obtidos na simulação dos *layouts* da versão do *Prescaler* com 3 D-FFs. Nessa tabela, o valor  $D_F$  indica a degradação que há com a implementação *standard cell* em relação à implementação *full custom* no que diz respeito à frequência máxima de operação. O valor  $D_A$  é relativo à degradação no consumo de potência.

Como o esperado, a implementação que utilizou a metodologia *full custom* apresentou melhores resultados. É possível dizer que essa implementação é cerca de 20% melhor que a implementação *standard cell*, em relação à frequência máxima de operação e consumo de potência.

Tabela 2: Comparação entre as implementações *full custom* e *standard cell* da versão do *Prescaler* com 3 D-FFs

$V_{DD}$ (V)	$f_{MAX}$ (GHz)		$D_F$ (%)	$\frac{P}{f}$ (mW/GHz)		$D_A$ (%)
	<i>standard cell</i>	<i>full custom</i>		<i>standard cell</i>	<i>full custom</i>	
3,0	2,52	3,05	17	0,953	0,746	28
2,5	2,10	2,57	18	0,667	0,511	30
2,0	1,61	1,87	14	0,430	0,340	26

O *layout* da versão do *Prescaler* com 3 D-FFs desenvolvido com metodologia *standard cell* ocupa uma área de  $3080 \mu\text{m}^2$ . O *layout* desse circuito feito com metodologia *full custom* ocupa  $2848 \mu\text{m}^2$ . A metodologia *full custom* tende a gerar *layouts* menores, porém neste caso não houve grande redução. Isso ocorre porque o *layout full custom* usa um *guard ring* em torno da parte síncrona do circuito, aumentando a área ocupada pelo *layout*.

A Tabela 3 apresenta os resultados obtidos na simulação dos *layouts* da versão do *Prescaler* com 2 D-FFs. A implementação *full custom* nesse caso também é melhor em termos de frequência máxima de operação e consumo de potência. Quando a tensão de V<sub>DD</sub> abaixa para 2V, observa-se que a diferença entre  $f_{MAX}$  dos dois *layouts* não é grande. Isso ocorre pois o pior roteamento da metodologia *standard cell* deixa de ser um problema e  $f_{MAX}$  passa a ser limitada pelo projeto do circuito (topologia, tipo de lógica e dimensionamento dos transistores utilizados).

Tabela 3: Comparação entre as implementações *full custom* e *standard cell* da versão do *Prescaler* com 2 D-FFs

V <sub>DD</sub> (V)	$f_{MAX}$ (GHz)		D <sub>F</sub> (%)	$\frac{P}{f}$ (mW/GHz)		D <sub>A</sub> (%)
	<i>standard cell</i>	<i>full custom</i>		<i>standard cell</i>	<i>full custom</i>	
3,0	1,90	2,31	18	1,014	0,836	21
2,5	1,42	1,80	21	0,731	0,590	24
2,0	1,02	1,01	-1	0,472	0,453	4

O *layout standard cell* da versão do *Prescaler* com 2 D-FFs ocupa uma área de 2247  $\mu\text{m}^2$ . O *layout full custom* ocupa uma área de 1787  $\mu\text{m}^2$ . Aqui é possível observar uma redução maior na área ocupada pelo circuito *full custom*.

#### 4.1. *Prescaler* com Lógica Estática CMOS

Para comparação, foi também desenvolvida uma terceira versão do *Prescaler* com 3 D-FFs utilizando células da biblioteca padrão da AMS, que é constituída somente de células estáticas CMOS. A Figura 31 apresenta o esquemático deste *Prescaler*. A Figura 32 apresenta o *layout* final do novo *Prescaler* e a Figura 33 mostra a forma de onda apresentada pelo circuito com o sinal de entrada **clk clock** de frequência 0,85 GHz, **SM** igual a “1” (divisão por 33) e V<sub>DD</sub> igual a 3 V.

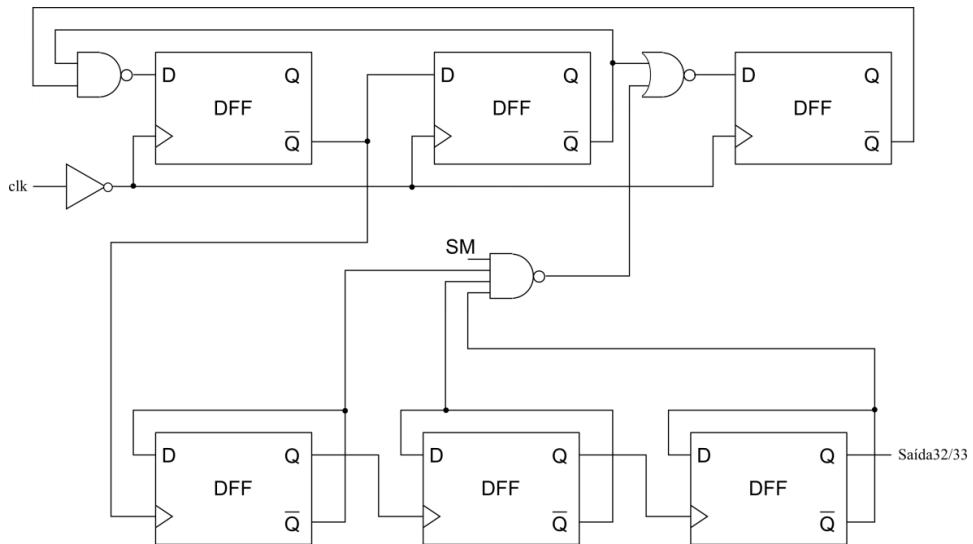


Figura 31: Esquemático do *Prescaler* desenvolvido com células da biblioteca padrão disponibilizada pela AMS

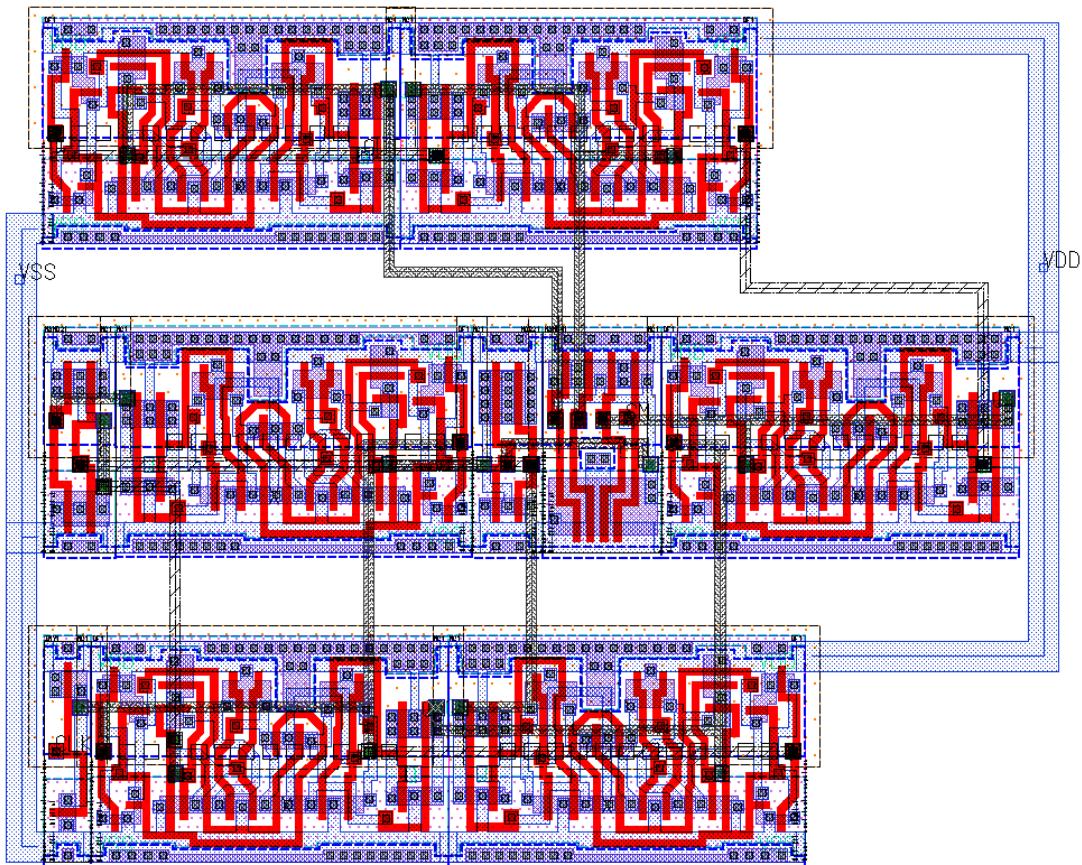


Figura 32: Layout do *Prescaler* desenvolvido com células da biblioteca padrão da AMS, área de  $3211 \mu\text{m}^2$

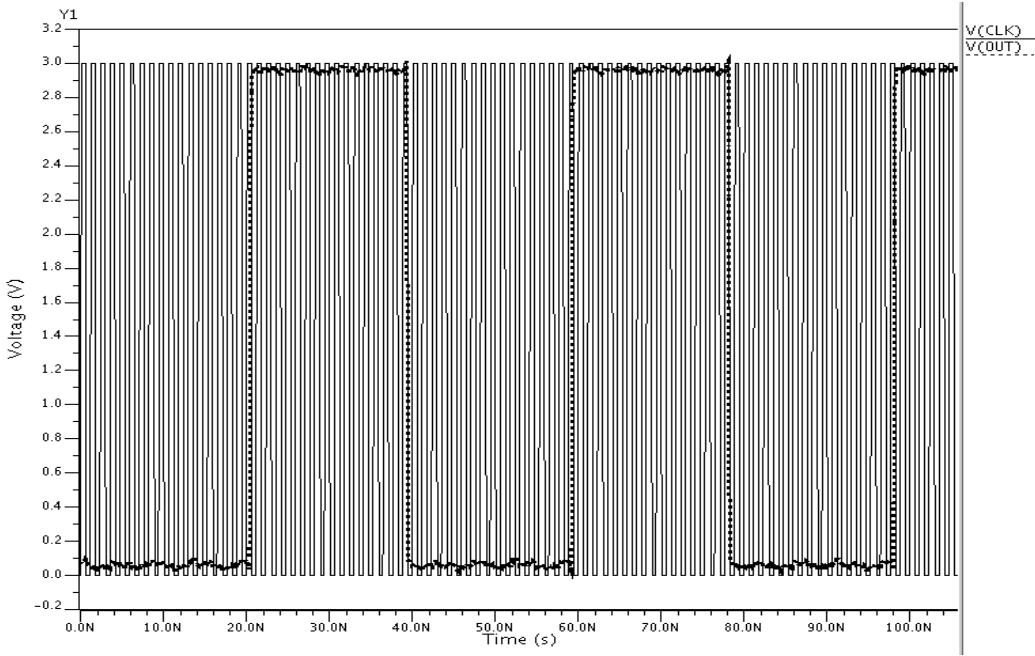


Figura 33: Forma de onda apresentada na saída do circuito *Prescaler* desenvolvido com células da biblioteca padrão da AMS. O sinal de entrada **clk** tem frequência de 0,85 GHz e tempos de subida e descida de 10% do período do sinal; **out** é o sinal Saída32/33

O *layout* ocupa uma área de  $3211 \mu\text{m}^2$ . Essa área é maior que a ocupada pelos circuitos que utilizam a técnica E-TSPC. A Tabela 4 mostra os resultados obtidos a partir da simulação do *layout* desses *Prescaler*. A frequência de operação é bastante inferior e o consumo de potência é maior que o dos circuitos E-TSPC que utilizam a biblioteca desenvolvida neste trabalho.

Tabela 4: Resultados da simulação do *Prescaler* desenvolvido com células da biblioteca padrão da AMS

$V_{DD}$ (V)	$f_{MAX}$ (GHz)	$\frac{P}{f}$ (mW/GHz)
3,0	0,86	2,099
2,5	0,70	1,442
2,0	0,51	0,912

A Figura 34 permite uma melhor visualização dos resultados de simulação das diferentes implementações da versão do *Prescaler* com 3 D-FFs. O circuito implementado com metodologia *standard cell* com células E-TSPC é cerca de 3 vezes mais rápido que o circuito desenvolvido apenas com células da biblioteca padrão AMS, e os circuitos *full custom* são apenas 20% mais rápidos que os circuitos *standard cell*.

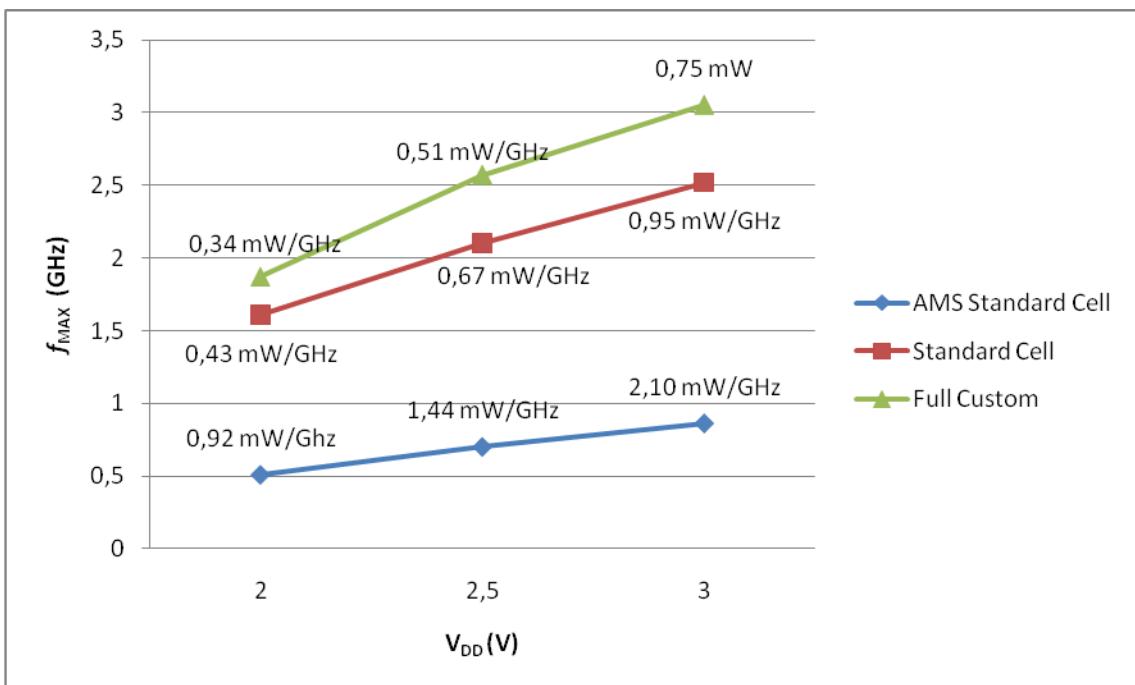


Figura 34: Visualização dos resultados de simulação da versão do *Prescaler* com 3-DFFs

## 5. Conclusão

Esta monografia apresenta o desenvolvimento de uma biblioteca de células E-TSPC com blocos suficientes para desenvolver dois diferentes circuitos *Dual-Modulus Prescaler 32/33*. A partir da comparação do desempenho dos circuitos com similares implementados com a metodologia *full custom*, é feita a verificação da eficiência do uso da biblioteca. Para completar as comparações, um dos *Prescalers* foi ainda desenvolvido com células da biblioteca padrão da AMS.

Neste trabalho foi visto que os circuitos desenvolvidos com a metodologia *standard cell* sofrem uma degradação na velocidade e um aumento no consumo de potência com relação aos circuitos desenvolvidos com a metodologia *full custom*. Porém, a degradação de velocidade e potência é de apenas 20% e a metodologia *standard cell* aumenta consideravelmente a produtividade no desenvolvimento de circuitos integrados. Assim, o projetista deve escolher qual metodologia se ajusta melhor ao seu problema, vindo a utilizar a metodologia *full custom*, se necessitar, apenas nas regiões mais críticas do circuito.

Mesmo com a utilização da metodologia *standard cell*, a técnica E-TSPC permite gerar circuitos que apresentam velocidades de operação muito superiores, da ordem de 200%, e ocupam áreas menores do que circuitos que utilizam apenas células da biblioteca padrão AMS. No entanto, deve-se ter cautela ao utilizar a técnica devido às dificuldades de implementação de circuitos E-TSPC. O projetista pode aplicar a técnica aos sub-circuitos mais críticos do sistema, a fim de aumentar sua velocidade. A evolução de *softwares EDA* pode facilitar o projeto de circuitos E-TSPC, automatizando a verificação das regras E-TSPC e a geração de circuitos a partir de uma linguagem de descrição de *hardware*.

A falta de documentação da biblioteca padrão da AMS, no que diz respeito ao modo como as células foram desenvolvidas, dificultou o uso de células da biblioteca desenvolvida neste trabalho com células da biblioteca padrão em um mesmo projeto. Isso ocorreu devido às diferentes distâncias que as camadas interiores das células apresentam em relação às bordas da célula, causando falhas no DRC e exigindo um redesenho do *layout* de algumas células.

Como sugestão para trabalhos futuros fica: entender o padrão utilizado nas células da biblioteca da AMS e desenvolver uma biblioteca com mais células; desenvolver células com transistores de largura 3 vezes maior, a cujas saídas possam ser ligadas cargas maiores; desenvolver *softwares* para verificação das regras E-TSPC e síntese de circuitos E-TSPC.

## **REFERÊNCIAS**

- AUSTRIAMICROSYSTEMS (2003). **0.35 µm CMOS process technology**. Disponível em: <<http://www.austriamicrosystems.com/05foundry/indexc35.htm>>. Acesso em: 10 Oct. 2010.
- AUSTRIAMICROSYSTEMS (2010). **HIT-Kit – High performance interface tool kit (PDK) for Cadence, Mentor, Agilent ADS tool, circuit simulation, device models**. Disponível em: <<http://asic.austriamicrosystems.com/hitkit/index.html>>. Acesso em: 20 Oct. 2010.
- GONÇALVES, N. F. (1984). **NORA**: a racefree CMOS technique for register transfer systems. Tese (Doutorado) – Katholieke Universiteit Leuven, Leuven, Bélgica, 1984.
- GRONOWSKI, P.E. et. al. (1998). High-performance digital microprocessor design. **IEEE J. Solid-State Circuits**, New York, v.33, n.5, p.676-686.
- LARSSON, P. (1995). Skew safety and logic flexibility in a true single phase clocked system. In: CIRCUITS AND SYSTEMS – ISCAS '95. IEEE INTERNATIONAL SYMPOSIUM ON, 1995. Seattle, WA. **Proceedings...** New York: IEEE. p.941-944.
- MENTOR GRAPHICS (2010). **Website: the EDA technology leader – Mentor Graphics**. Disponível em: <<http://www.mentor.com/>>. Acesso em: 20 Oct. 2010.
- MIRANDA, F. P. H. (2006). **Estudo e projeto de circuitos dual-modulus prescalers em tecnologia CMOS**. Dissertação (Mestrado) – Departamento de Engenharia Eletrônica, Escola Politécnica da USP, São Paulo, 2006.
- NAVARRO, J. (1998). **Técnicas para projeto de ASICs CMOS de alta velocidade**. Tese (Doutorado) – Departamento de Engenharia Eletrônica, Escola Politécnica da USP, São Paulo, 1998.
- NAVARRO, J.; VAN NOIJE, W. (1997). A 1.6-GHz dual modulus prescaler using the extended true single phase clock CMOS circuit technique for high speed applications. **Journal of Solid-State Devices and Circuits**, São Paulo, v.5, p.21-26.
- NAVARRO, J.; VAN NOIJE, W. (2002). Extended TSPC structures with double input/output throughput for gigahertz CMOS circuit design. **IEEE Transactions on Very Large Scale Integration Systems**. Princeton, v.10, n.3, Jun.

RABEY, J.M.; CHANDRAKASAN, A.; NIKOLIC, B. (2006). **Digital integrated circuits.** 2.ed. Berkeley: Prentice Hall. 761 p.

SEMICONDUCTOR INDUSTRY ASSOCIATION (SIA). (2009). **The international technology roadmap for semiconductors.** San Jose, 2009.

WESTE, N.H.E.; ESHRAGHIAN, K. (1993). **Principles of CMOS VLSI design.** 2 ed. Reading, Mass.: Addison-Wesley.

YUAN, J.-R. et al. (1993). New domino logic precharged by clock and data. **Electronics Letters**, New York, v.29, n.25, p.2188-2189.

## Apêndice A

Modelos BSIM3v3 dos transistores para a ferramenta ELDO. Estes são os parâmetros típicos.

```
.MODEL MODN NMOS LEVEL=53 MODTYPE=ELDO

* -----
***** SIMULATION PARAMETERS *****
* -----
* format   : ELDO, AccusimII, Continuum
* model    : MOS BSIM3v3
* process   : C35
* revision  : 4.0;
* extracted : B10866 ; 2002-12; ese(5487)
* doc#     : ENG-182 REV_4
* -----
*                               TYPICAL MEAN CONDITION
* -----
*
+THMLEV =0
*      *** Flags ***
+NOIMOD =3 FLKLEV =0
+MOBMOD =1.000e+00 CAPMOD =2.000e+00 VERSION=3.240e+00 NQSMOD=0.00e+00
+DERIV =1
*      *** Threshold voltage related model parameters ***
+K1     =5.0296e-01
+K2     =3.3985e-02 K3      =-1.136e+00 K3B      =-4.399e-01
+NPEAK  =2.611e+17 VTH0    =4.979e-01
+VOFF   =-8.925e-02 DVT0    =5.000e+01 DVT1    =1.039e+00
+DVT2   =-8.375e-03 KETA    =2.032e-02
+PSCBE1 =1.000e+30 PSCBE2 =1.000e-06
+DVT0W  =1.089e-01 DVT1W   =6.671e+04 DVT2W   =-1.352e-02
*      *** Mobility related model parameters ***
+UA     =4.705e-12 UB      =2.137e-18 UC      =1.000e-20
+U0     =4.758e+02
*      *** Subthreshold related parameters ***
+DSUB   =5.000e-01 ETA0    =1.415e-02 ETAB    =-1.221e-01
+NFACTOR=4.136e-01
*      *** Saturation related parameters ***
+EM     =4.100e+07 PCLM    =6.948e-01
+PDIBLC1=3.571e-01 PDIBLC2=2.065e-03 DROUT   =5.000e-01
+A0     =2.541e+00 A1      =0.000e+00 A2      =1.000e+00
+PVAG   =0.000e+00 VSAT    =1.338e+05 AGS     =2.408e-01
+B0     =4.301e-09 B1      =0.000e+00 DELTA   =1.442e-02
+PDIBLCB=3.222e-01
*      *** Geometry modulation related parameters ***
+W0     =2.673e-07 DLC     =3.0000e-08
+DWC    =9.403e-08 DWB     =0.000e+00 DWG     =0.000e+00
+LL     =0.000e+00 LW      =0.000e+00 LWL     =0.000e+00
+LLN    =1.000e+00 LWN    =1.000e+00 WL      =0.000e+00
+WW     =-1.297e-14 WWL    =-9.411e-21 WLN    =1.000e+00
+WWN    =1.000e+00
*      *** Temperature effect parameters ***
+AT     =3.300e+04 UTE    =-1.800e+00
+KT1    =-3.302e-01 KT2    =2.200e-02 KT1L    =0.000e+00
+UA1    =0.000e+00 UB1    =0.000e+00 UC1    =0.000e+00
+PRT   =0.000e+00
*      *** Overlap capacitance related and dynamic model parameters ***
+CGSO   =1.200e-10
```

```

+CGDO    =1.200e-10 CGBO    =1.100e-10
+CGDL    =1.310e-10 CGSL    =1.310e-10 CKAPPA =6.000e-01
+CF      =0.000e+00 ELM     =5.000e+00
+XPART   =1.000e+00 CLC     =1.000e-15 CLE     =6.000e-01
***** Parasitic resistance and capacitance related model parameters ***
+RDSW    =3.449e+02
+CDSC    =0.000e+00 CDSCB   =1.500e-03 CDSCD   =1.000e-03
+PRWB    =-2.416e-01 PRWG    =0.000e+00 CIT     =4.441e-04
*      *** Process and parameters extraction related model
parameters ***
+TOX     =7.575e-09 NGATE   =0.000e+00
+Nlx     =1.888e-07
+XL      =0.000e+00 XW      =0.000e+00
*      *** Substrate current related model parameters ***
+ALPHA0  =2.600e-06 ALPHA1  =5.000e+00 BETA0   =2.100e+01
*      *** Noise effect related model parameters ***
+AF      =1.507e+00 KF      =2.170e-26 EF     =1.000e+00
+NOIA    =1.121e+19 NOIB    =5.336e+04 NOIC    =-5.892e-13
*      *** Common extrinsic model parameters ***
+ALEV    =2          RLEV    =2
+RD      =0.000e+00 RS      =0.000e+00 RSH    =7.000e+01
+RDC     =0.000e+00 RSC     =0.000e+00 LD     =-5.005e-08
+WD      =9.403e-08
+LDIF    =0.000e+00 HDIF    =8.000e-07 WMLT   =1.000e+00
+LMLT    =1.000e+00 DEL     =0.000e+00 XJ     =3.000e-07
+DIOLEV  =4          JS      =5.100e-07 JSW    =0.600e-12
+IS      =0.000e+00 N       =1.000e+00
+DCAPLEV=2          CBD     =0.000e+00 CBS    =0.000e+00
+CJ      =8.400e-04 CJSW    =2.500e-10 FC     =0.000e+00
+MJ      =3.400e-01 MJSW    =2.300e-01 TT     =0.000e+00
+XTI     =2.026e+00 PB     =6.900e-01 PBSW   =6.900e-01
* -----

```

```

.MODEL MODP PMOS LEVEL=53 MODTYPE=ELDO
* -----
***** SIMULATION PARAMETERS *****
* -----
* format   : ELDO, AccusimII, Continuum
* model    : MOS BSIM3v3
* process   : C35
* revision : 4.0;
* extracted : C64685 ; 2002-12; ese(5487)
* doc#     : ENG-182 REV_4
* -----
*           TYPICAL MEAN CONDITION
* -----
*
+THMLEV =0
*      *** Flags ***
+NOIMOD =3 FLKLEV =0
+MOBMOD =1.000e+00 CAPMOD =2.000e+00 VERSION=3.24e+00 NQSMOD=0.000e+00
+DERIV  =1
*      *** Threshold voltage related model parameters ***
+K1     =5.9959e-01
+K2     =-6.038e-02 K3     =1.103e+01 K3B    =-7.580e-01
+NPEAK  =9.240e+16 VTH0   =-6.915e-01
+VOFF   =-1.170e-01 DVT0   =1.650e+00 DVT1   =3.868e-01
+DVT2   =1.659e-02 KETA   =-1.440e-02
+PSCBE1 =1.000e+30 PSCBE2 =1.000e-06

```

```

+DVT0W =1.879e-01 DVT1W =7.335e+04 DVT2W =-6.312e-03
*      *** Mobility related model parameters ***
+UA =5.394e-10 UB =1.053e-18 UC =1.000e-20
+U0 =1.482e+02
*      *** Subthreshold related parameters ***
+DSUB =5.000e-01 ETA0 =2.480e-01 ETAB =-3.917e-03
+NFACTOR=1.214e+00
*      *** Saturation related parameters ***
+EM =4.100e+07 PCLM =3.184e+00
+PDIBLC1=1.000e-04 PDIBLC2=1.000e-20 DROUT =5.000e-01
+A0 =5.850e-01 A1 =0.000e+00 A2 =1.000e+00
+PVAG =0.000e+00 VSAT =1.158e+05 AGS =2.468e-01
+B0 =8.832e-08 B1 =0.000e+00 DELTA =1.000e-02
+PDIBLCB=1.000e+00
*      *** Geometry modulation related parameters ***
+W0 =1.000e-10 DLC =2.4500e-08
+DWC =3.449e-08 DWB =0.000e+00 DWG =0.000e+00
+LL =0.000e+00 LW =0.000e+00 LWL =0.000e+00
+LLN =1.000e+00 LWN =1.000e+00 WL =0.000e+00
+WW =1.894e-16 WWL =-1.981e-21 WLN =1.000e+00
+WWN =1.040e+00
*      *** Temperature effect parameters ***
+AT =3.300e+04 UTE =-1.300e+00
+KT1 =-5.403e-01 KT2 =2.200e-02 KT1L =0.000e+00
+UA1 =0.000e+00 UB1 =0.000e+00 UC1 =0.000e+00
+PRT =0.000e+00
*      *** Overlap capacitance related and dynamic model parameters ***
+CGSO =8.600e-11
+CGDO =8.600e-11 CGBO =1.100e-10
+CGDL =1.080e-10 CGSL =1.080e-10 CKAPPA =6.000e-01
+CF =0.000e+00 ELM =5.000e+00
+XPART =1.000e+00 CLC =1.000e-15 CLE =6.000e-01
*      *** Parasitic resistance and capacitance related model
parameters ***
+RDSW =1.033e+03
+CDSC =2.589e-03 CDSCB =2.943e-04 CDSCD =4.370e-04
+PRWB =-9.731e-02 PRWG =1.477e-01 CIT =0.000e+00
*      *** Process and parameters extraction related model parameters ***
+TOX =7.754e-09 NGATE =0.000e+00
+NLX =1.770e-07
+XL =0.000e+00 XW =0.000e+00
*      *** Substrate current related model parameters ***
+ALPHA0 =1.000e-09 ALPHA1 =1.500e+00 BETA0 =3.250e+01
*      *** Noise effect related model parameters ***
+AF =1.461e+00 KF =1.191e-26 EF =1.000e+00
+NOIA =5.245e+17 NOIB =4.816e+03 NOIC =8.036e-13
*      *** Common extrinsic model parameters ***
+ALEV =2 RLEV =2
+RD =0.000e+00 RS =0.000e+00 RSH =1.290e+02
+RDC =0.000e+00 RSC =0.000e+00 LD =-7.130e-08
+WD =3.449e-08
+LDIF =0.000e+00 HDIF =8.000e-07 WMLT =1.000e+00
+IMLT =1.000e+00 DEL =0.000e+00 XJ =3.000e-07
+DIOLEV =4 JS =2.800e-07 JSW =3.700e-13
+IS =0.000e+00 N =1.000e+00
+DCAPLEV=2 CBD =0.000e+00 CBS =0.000e+00
+CJ =1.360e-03 CJSW =3.500e-10 FC =0.000e+00
+MJ =5.400e-01 MJSW =4.600e-01 TT =0.000e+00
+XTI =1.973e+00 PB =1.020e+00 PBSW =1.020e+00
* -----

```

## Apêndice B

### Sintetizador de Frequências

Sintetizadores de Frequências são circuitos que fornecem sinais em diferentes frequências e são empregados na sintonia de canais dentro de uma banda de transmissão RF (Radio Frequência). Um Sintetizador pode ser implementado como um PLL (*phase looked loop*) onde é colocado um divisor programável no seu caminho de realimentação. Pela programação deste divisor pode-se controlar a freqüência do sinal gerado. Esta é a chamada arquitetura *integer N* que é mostrada na Figura 35. Os blocos que compõe tal arquitetura são: o **PD** (*phase detector*), o **LPF** (*low pass filter*), o **VCO** (*voltage controlled oscillator*), e o **divisor M** (divisor programável chamado *Pulse Swallow*).

Para o funcionamento do circuito deve-se aplicar um sinal de referências externo, com uma freqüência  $f_{Ref}$ , ao **PD**. O **VCO** e o Sintetizador não estabilizam enquanto o sinal de saída do **divisor M** não atingir freqüência  $f_{Ref}$ . Por outro lado, quando isto ocorrer teremos o chamado estado *locked* e o sinal de saída terá freqüência  $f_{osc} = Mf_{Ref}$ , que dependerá do valor de  $M$ . Dessa forma, desde que possamos modificar o valor de  $M$ , teremos controle sobre a freqüência de saída.

O circuito **divisor M** ou *Pulse Swallow* é implementado com um **Prescaler**, um contador programável chamado *Swallow Counter* e um **contador principal**. O funcionamento do divisor inicia com a carga dos dois contadores, o **principal**, com um valor fixo  $P$ , e o *Swallow Counter*, com um valor programável  $S$ . O valor  $S$  deve ser menor ou igual a  $P$ ,  $S \leq P$ . No estado inicial o valor do sinal **SM** fica no nível lógico ALTO, o que faz com que o circuito **Prescaler** divida a freqüência do sinal de saída do **VCO** por  $N+1$ . Dessa forma, a cada  $N+1$  ciclos, os valores do **contador principal** e do *Swallow Counter* serão reduzidos em um, até que o *Swallow Counter* chegue à zero. Neste momento o valor do sinal **SM** é alterado e o **Prescaler** passa a dividir por  $N$  o sinal de saída do **VCO**. Agora, a cada  $N$  ciclos, o **contador principal** será reduzido em um, até que também chegue à zero. Os contadores nesse instante são reinicializados com  $P$  e  $S$  e o processo é repetido. Com isso teremos que em um ciclo completo de operação do circuito Sintetizador o **Prescaler** divide às vezes por  $N+1$  e divide  $(P-S)$  vezes por  $N$ . O resultado final na saída do *Pulse Swallow* será a divisão do seu sinal de entrada, que é à saída do **VCO**, por  $M$ , onde

$$M = (N + 1)S + N(P - S) = NP + S \quad (2)$$

Quando o Sintetizador de Freqüências estiver no estado *locked*, ele apresentará uma freqüência de saída igual a

$$f_{osc} = Mf_{Ref} = (NP + S)f_{Ref} \quad (3)$$

Controlamos, portanto, a freqüência de saída do **VCO** com o valor  $S$  programado no **Swallow Counter**.

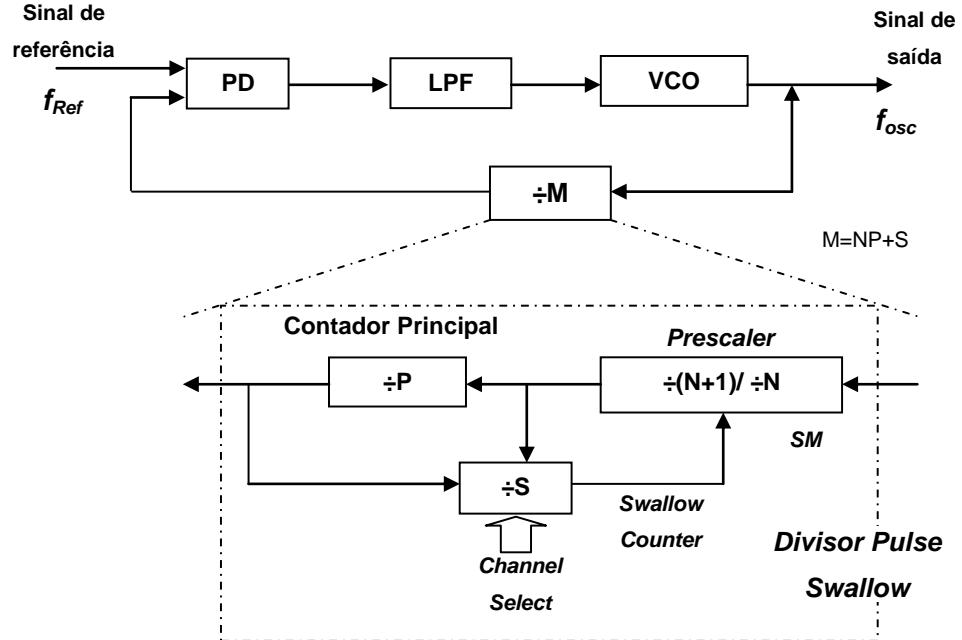


Figura 35: Diagrama de blocos da arquitetura *integer N* de sintetizadores de freqüência