



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica

Projeto de Estágio

FLUXO DE PROJETO DE CIRCUITOS INTEGRADOS
EM TECNOLOGIA FD-SOI 28NM BASEADO EM
ESTRUTURA DE CAMINHO DE DADOS

Débora Nunes Pinto de Oliveira

Campina Grande, Paraíba, Brasil

©Débora Nunes Pinto de Oliveira, 10 de maio de 2021

Débora Nunes Pinto de Oliveira

FLUXO DE PROJETO DE CIRCUITOS INTEGRADOS
EM TECNOLOGIA FD-SOI 28NM BASEADO EM
ESTRUTURA DE CAMINHO DE DADOS

Projeto de Estágio de Bacharelado submetida à Coordenadoria de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Orientador: Prof. Marcos Ricardo Alcântara Morais, Dr.

Campina Grande, Paraíba, Brasil

©Débora Nunes Pinto de Oliveira, 10 de maio de 2021

Débora Nunes Pinto de Oliveira

FLUXO DE PROJETO DE CIRCUITOS INTEGRADOS
EM TECNOLOGIA FD-SOI 28NM BASEADO EM
ESTRUTURA DE CAMINHO DE DADOS

Projeto de Estágio de Bacharelado submetida à Co-ordenadoria de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Aprovado em ___/___/___

Prof. Marcos Ricardo Alcântara Morais, Dr.

Orientador

Prof. Gutemberg Gonçalves dos Santos Júnior, Dr.

Avaliador

Campina Grande, Paraíba, Brasil

©Débora Nunes Pinto de Oliveira, 10 de maio de 2021

Agradecimentos

Agradeço ao Prof. Marcos Morais, pelo incentivo à pesquisa e incansável orientação e colaboração sem as quais não seria possível realizar este trabalho. Também agradeço aos colegas do laboratório XMEN, em especial Samuel, Fernando, Danilo e Agripino, pelo acompanhamento, estímulo e auxílio na execução dos experimentos. Por fim, gratifico a Prof. Gutemberg Gonçalves e a Hugo Gayoso o empenho e determinação em tornar esse estágio factível nas condições de pandemia.

Resumo

Esse projeto apresenta a documentação do fluxo de projeto de circuitos integrados para a tecnologia FD-SOI em 28nm. Neste trabalho, é proposto um fluxo de síntese lógica e física e concepção de leiaute para a transformação de uma programação lógica em um conjunto de poços de silício dopado. Com o intuito de otimizar a área, consumo e temporização do leiaute final, foi aplicado o método de estruturação do caminho de dados. Dessa forma, esse relatório de estágio reporta a consolidação do fluxo de implementação física utilizado no desenvolvimento de CIs de baixo consumo e alta densidade.

Palavras-chave: circuito integrado, microeletrônica, fluxo de projeto, síntese lógica, síntese física, estrutura de caminho de dados.

Sumário

| | |
|--|------------|
| Lista de Símbolos e Abreviaturas | iii |
| Lista de Tabelas | vi |
| Lista de Figuras | vii |
| 1 Introdução | 1 |
| 1.1 Justificativa | 1 |
| 1.2 Objetivo geral | 2 |
| 1.3 Objetivos específicos | 2 |
| 1.4 Local do estágio | 3 |
| 1.5 Organização do documento | 3 |
| 2 Tecnologias de transistores CMOS | 4 |
| 2.1 Redução de escala dos circuitos integrados | 4 |
| 2.2 Transistores tipo bulk | 5 |
| 2.3 Transistores SOI | 6 |
| 2.3.1 Transistores PD-SOI e FD-SOI | 7 |
| 2.4 Considerações finais | 9 |
| 3 Fluxo de projeto de circuitos integrados | 10 |
| 3.1 Etapas do fluxo de projeto | 10 |
| 3.2 Arquitetura do filtro FIR | 13 |
| 3.3 PDK FD-SOI 28nm | 14 |
| 3.4 Síntese lógica e física | 15 |

| | | |
|----------|---------------------------------------|-----------|
| 3.5 | <i>Place and Route</i> | 19 |
| 3.6 | Considerações finais | 22 |
| 4 | Otimização do caminho de dados | 24 |
| 4.1 | Estrutura de caminho de dados | 24 |
| 4.2 | SDP para o filtro FIR | 26 |
| 4.3 | Considerações finais | 32 |
| 5 | Conclusão | 33 |
| | Referências bibliográficas | 35 |

Lista de Símbolos e Abreviaturas

| | | |
|------------|--|----|
| V_t | Tensão de limiar (<i>threshold voltage</i>) | 6 |
| V_{DDs} | Potencial do substrato p ⁺ de um transistor SOI pMOS de poço invertido | 8 |
| V_{DD} | Tensão de alimentação do transistor [V] | 5 |
| V_{GNDs} | Potencial do substrato n ⁺ de um transistor SOI nMOS de poço invertido | 8 |
| V_{GS} | Tensão de porta-coletor (<i>gate-source voltage</i>) | 6 |
| BOX | Óxido enterrado (<i>buried oxide</i>) | 6 |
| CI | Circuito Integrado | 4 |
| CTS | <i>Clock Tree Synthesis</i> | 21 |
| D | Dreno (<i>drain</i>) | 5 |
| DRV | Violação de regra da tecnologia (<i>design rule violation</i>) | 18 |
| FBB | Polarização direta do corpo (<i>forward body bias</i>) | 8 |
| FD-SOI | Silício sobre isolante totalmente deplecionado (<i>fully depleted SOI</i>) | 7 |
| FIR | Resposta finita ao impulso (<i>finite impulse response</i>) | 13 |
| HDL | Linguagem de descrição de hardware (<i>hardware description language</i>) | 10 |
| HVT | Alta tensão de limiar (<i>high-voltage threshold</i>) | 8 |
| LVT | Baixa tensão de limiar (<i>low-voltage threshold</i>) | 8 |
| MMMC | Multimodal/multi-corner | 19 |
| P&R | <i>Place and Route</i> | 12 |
| PD-SOI | Silício sobre isolante parcialmente deplecionado (<i>partially depleted SOI</i>) | 7 |

| | | |
|-----|---|----|
| PDK | Kit de <i>design</i> de processo (<i>process design kit</i>) | 12 |
| RBB | Polarização reversa do corpo (<i>reverse body bias</i>) | 8 |
| RDF | Flutuação aleatória do dopante (<i>random dopant fluctuation</i>) | 5 |
| RTL | <i>Register transfer level</i> | 10 |
| S | Coletor (<i>source</i>) | 5 |
| SCE | Efeitos do canal curto (<i>short-channel effects</i>) | 5 |
| SDP | Estrutura de caminho de dados (<i>structured data path</i>) | 24 |
| Si | Silício | 4 |
| SOI | Silício sobre isolante (<i>silicon on insulator</i>) | 5 |
| WNS | <i>worst negative slack</i> | 16 |

Lista de Tabelas

| | | |
|-----|--|----|
| 3.1 | Dados de consumo, área e temporização para as simulações pós-síntese de cada fase da síntese lógica e física. | 16 |
| 3.2 | Condições de temperatura e tensão de alimentação V_{DD} para a análise MMMC. | 20 |
| 3.3 | Resultados das etapas de P&R para o filtro FIR. | 22 |
| 4.1 | Resultados das etapas de P&R para o filtro FIR com SDP. | 26 |
| 4.2 | Resultados das etapas de P&R para o filtro FIR com o SDP reestruturado. . | 29 |
| 4.3 | Resultados de consumo e otimização de tempo para todos os <i>corner</i> empregando os três fluxos de concepção de leiaute. | 31 |

Lista de Figuras

| | | |
|-----|---|----|
| 2.1 | Estrutura dos substratos para um transistor tipo bulk e SOI | 5 |
| 2.2 | Estrutura dos tiristores intrínsecos para o transistor tipo bulk e SOI. | 7 |
| 2.3 | Diagrama e corte lateral da estrutura CMOS para dispositivos SOI. | 8 |
| 3.1 | Fluxo de projeto de circuito integrado para a tecnologia FD-SOI. | 12 |
| 3.2 | Diagrama lógico da resposta $y[n]$ de um filtro FIR com entrada $x[n]$ e coeficientes $b_i, \forall i \in \mathbb{Z} \cap (0, N)$. | 13 |
| 3.3 | Diagrama de blocos da resposta de um filtro FIR com entradas <code>rst</code> , <code>clk</code> e <code>filter_in</code> e saída <code>filter_out</code> . | 14 |
| 3.4 | Comprimento das células disponíveis pelo PDK FD-SOI 28nm da <i>ST Microelectronics</i> para alta performance 12T (esquerda) e alta densidade 8T (direita). | 15 |
| 3.5 | Disposição das células de isolamento e das trilhas de alimentação do filtro FIR. | 18 |
| 3.6 | Leiaute do filtro FIR pós <i>placement</i> da <i>netlist</i> . | 20 |
| 3.7 | Leiaute do filtro FIR pós P&R. | 22 |
| 4.1 | Fluxo de desenvolvimento do SDP no fluxo de projeto de concepção e simulação física do leiaute. | 25 |
| 4.2 | Estruturas dos registradores na especificação do SDP. | 27 |
| 4.3 | Distribuição de densidade no leiaute com e sem SDP. | 28 |
| 4.4 | Distribuição de densidade no leiaute como o SDP reestruturado. | 30 |

Capítulo 1

Introdução

Neste capítulo serão apresentados a justificativa, os objetivos e o local do desenvolvimento desse projeto de estágio. Por fim, será descrita a organização dos capítulos e das seções deste documento.

1.1 Justificativa

Segundo afirma a lei de Moore [1], a tendência da evolução dos circuitos integrados (CIs) é o aumento do número de transistores por pastilha — em inglês, *die*. Por sua vez, observando a tendência de mercado, é evidente a redução de área dos CIs. Dessa forma, nas últimas décadas, a pesquisa em microeletrônica foca na miniaturização dos transistores.

Esse processo de redução de tamanho ao passo do aumento do número de transistores por *die* é denominado escalonamento da tecnologia CMOS. Quando empregados dispositivos tradicionais tipo bulk, o desafio da miniaturização é garantir o funcionamento correto do transistor com menor comprimento de porta. Uma vez que a redução da largura do canal implica intensificação do campo elétrico entre o dreno e o coletor, a potência e a temperatura do circuito superam a capacidade de alimentação e resfriamento dos equipamentos.

Com o intuito de solucionar esse problema foi proposta uma tecnologia FD-SOI com novas camadas de isolante. A inovação dessa técnica consiste nos poços de silício dopados, cujos potenciais podem ser configurados separadamente. Dessa forma, menores correntes de fuga percorrem o canal do transistor, reduzindo as perdas.

Entretanto, o processo de manufatura de CIs via litografia é sistemático e já consolidado para tecnologias com processos maiores. Portanto, torna-se necessário elaborar um fluxo de transformação de um programa lógico para o padrão de camadas da nova tecnologia. Essa nova metodologia de trabalho deve considerar as especificações exclusivas nas novas camadas dos poços dopados.

Com o intuito de usufruir a melhor otimização da escala da tecnologia, é também interessante compactar o leiaute final. Essa redução de área pode ser realizada caracterizando posições específicas de células do caminho de dados. Como consequência, essa metodologia facilita as correções de violação de temporização.

1.2 Objetivo geral

Tendo em vista as limitações da tecnologia típicas tipo *bulk* e a necessidade da configuração do fluxo de projeto otimizado para a menor largura de canal, o objetivo geral deste relatório de estágio é documentar o fluxo de projeto para uma tecnologia FD-SOI por meio de experimentos de síntese e concepção de leiaute. Esses ensaios constam na especificação, caracterização e otimização da arquitetura de um filtro FIR. Dessa forma, esse trabalho propõe dispor ao laboratório uma documentação do fluxo de implementação física utilizado no desenvolvimento de CIs de baixo consumo e alta densidade.

1.3 Objetivos específicos

Para a realização do objetivo geral, são definidos os seguintes objetivos específicos:

- Compreender o funcionamento e configurações de polarização dos transistores da tecnologia SOI;
- Estudar e documentar o fluxo de projeto da arquitetura do filtro FIR para síntese lógica e física;
- Executar ensaios para caracterização do leiaute e otimização de consumo, temporização e área para o filtro FIR sintetizado.

1.4 Local do estágio

O projeto de estágio descrito nesse documento foi realizado o Laboratório de Excelência em Microeletrônica do Nordeste (XMEN), cuja infraestrutura está localizada na Universidade Federal de Campina Grande (UFCG). O XMEN atua no desenvolvimento de atividades de ensino, pesquisa e inovação no contexto de microeletrônica. As atividades são realizadas por discentes nos níveis de graduação e pós-graduação, em um ambiente equipado com computadores de última geração.

1.5 Organização do documento

O capítulo 2 trata da modelagem dos transistores tipo *bulk* e SOI. São detalhados os limites físicos de escalonamento dos transistores *bulk* e as configurações de polarização dos transistores CMOS FD-SOI.

O capítulo 3 aborda o fluxo de projeto padrão adotado para a síntese lógica, física e concepção de leiaute do filtro FIR. São detalhadas as características do pacote de tecnologia disponível para o estudo.

O capítulo 4 trata do fluxo de projeto para aplicação do método de estruturação do caminho da dados sobre a arquitetura do filtro FIR. Em seguida, são expostos e discutidos os resultados para cada configuração proposta.

Por fim, o capítulo 5 trata das conclusões deste estudo e das contribuições desta documentação para futuros trabalhos.

Capítulo 2

Tecnologias de transistores CMOS

Esse capítulo trata da evolução dos circuitos integrados segundo a redução de escala dos transistores CMOS e da descrição da tecnologia SOI focada em aplicações de baixo consumo ou alta performance.

2.1 Redução de escala dos circuitos integrados

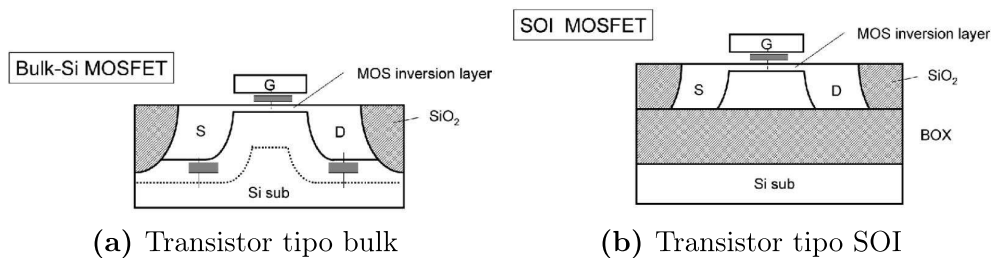
Os semicondutores marcam a evolução tecnológica da informação, uma vez que são indispensáveis em qualquer área com aplicação eletrônica. Esses semicondutores são geralmente compostos por silício (Si) dopado positiva ou negativamente e compõem o material-base para a construção de uma junção P-N. Associações específicas de junções P-N originam os transistores e diodos. Por sua vez, o conjunto desses dispositivos com capacitores ou indutores é nomeado circuito integrado (CI).

Apesar de comumente ocultos ao usuário, os CIs são amplamente difundidos na vida cotidiana do século XXI, visto que celulares, computadores, carros e geladeiras contém bilhões de transistores [2]. Conforme descreve a Lei de Moore [1], o mercado de circuitos eletrônicos demanda transistores cada vez mais rápidos, com menor área e consumo [3]. Entretanto, a minimização da escala da tecnologia CMOS desacelerou nos últimos 5 anos [4] em razão dos limites físicos dos transistores comuns tipo bulk. A estrutura e restrições do escalonamento desse tipo de dispositivo são tratadas na [Seção 2.2](#).

2.2 Transistores tipo bulk

Os transistores tipo bulk são construídos por poços p^+ ou n^+ em um substrato, conforme ilustrado na Figura 2.1a. Os poços representam o dreno (*drain*, D) e o coletor (*source*, S). Por sua vez, o canal é composto pelo substrato Si dopado inversamente ao poços D e S.

Figura 2.1: Estrutura dos substratos para um transistor tipo bulk e SOI



(a) Transistor tipo bulk

(b) Transistor tipo SOI

Fonte: SAKURAI, A.MATSUZAWA e T.DOUSEKI[5]

Com a miniaturização dos transistores bulk, a dispersão das impurezas no canal deve ser considerada no comportamento elétrico do dispositivo. Essa condição dificulta o deslocamento de portadores elétricos entre o coletor e o dreno. Por sua vez, a redução da área do isolante entre o canal e a porta aumenta a corrente de fuga e as perdas nas junções P-N.

Nessa situação, a tensão de alimentação V_{DD} deve ser elevada para a execução de operações mais rápidas [6] com os transistores tipo bulk. Como a potência consumida pelo CI é proporcional a V_{DD} , é evidente a redução de escala dos transistores bulk provoca um aumento do consumo e da temperatura de operação. Essa temperatura geralmente supera a capacidade de resfriamento dos aparelhos [5].

Segundo a teoria de escalonamento de Dennard [7], que descreve a miniaturização dos transistores, a concentração de portadores no canal deve aumentar para evitar essa queda da tensão de limiar das junções P-N e essa redução de mobilidade dos portadores. Contudo, o crescimento da dopagem do canal origina o aumento do campo elétrico na junção e a variabilidade do dispositivo, respectivamente nomeados efeitos do canal curto (*short-channel effects*, SCE) e de flutuação aleatória do dopante (*random dopant fluctuation*, RDF) dos transistores com largura do canal menor que 22nm.

Com o objetivo de solucionar esses desafios da miniaturização do tipo bulk, desenvolveu-se a tecnologia SOI (*silicon on insulator*). Esses dispositivos serão apresentados na Seção 2.3.

2.3 Transistores SOI

A inovação da tecnologia SOI em relação ao bulk consiste na construção de uma camada isolante de óxido de silício (SiO_2) nomeada BOX (buried oxide) abaixo do canal, conforme ilustrada na [Figura 2.1b](#). Como o óxido de silício é um bom isolante, a tecnologia SOI apresenta uma capacitância entre dreno e substrato cerca de 10 vezes menor que a tecnologia bulk [\[5\]](#). Sobre a camada BOX é depositado um substrato cristalino. Caso esse material seja um cristal singular de silício, o dispositivo é nomeado SOI. Por sua vez, filmes policristalinos definem os transistores TFT (*thin-film transistors*).

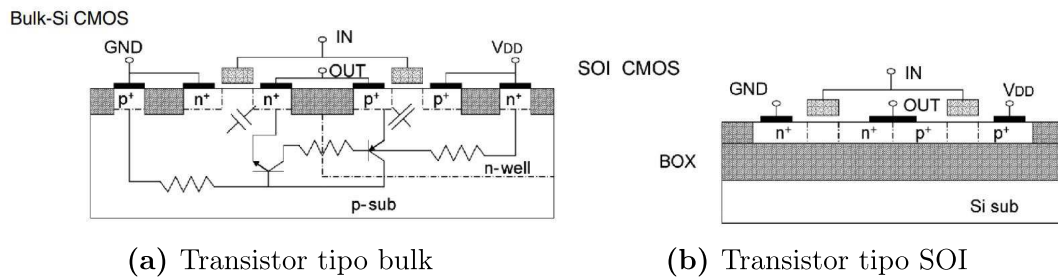
A baixa da capacitância entre o dreno e o substrato determinada pela BOX permite operações mais rápidas que o transistor tipo bulk para um mesmo potencial V_{DD} [\[5\]](#). Tendo em vista que as impurezas do tipo p^+ ou n^+ se difundem profundamente na camada SOI e se concentram nas paredes laterais do dispositivo, a tensão de limiar V_t é reduzida. Logo, a polarização do canal $V_{GS} - V_t$ é majorada [\[4\]](#), para V_{GS} a tensão entre a porta e o coletor. Como consequência, é reduzido o consumo de energia total do CI.

Além disso, a BOX impede a interferência da operação por radiação, habilita a operação do dispositivo em altas temperaturas e permite a construção de dispositivos flexíveis, uma vez que as propriedades do substrato e do canal podem ser definidas separadamente. Essa escolha não é possível no dispositivo tipo bulk, uma vez que o material do substrato compõe o canal de depleção entre o dreno e o coletor, conforme ilustrado na [Figura 2.1a](#).

A tecnologia SOI também evita o efeito de *latch-up*, o qual ocorre quando o tiristor intrínseco das junções P-N de um transistor tipo bulk — o qual está ilustrado na [Figura 2.2a](#) — é acionado. Conforme apresentado na [Figura 2.2b](#), a camada BOX nos transistores SOI provoca o isolamento vertical entre os dispositivos e evita a construção do tiristor entre as junções P-N em série. Logo, as células do tipo SOI podem ser ajustadas lateralmente com um menor espaçamento que a tecnologia bulk, resultando em uma menor área total do CI.

Por fim, os dispositivos SOI apresentam um maior custo benefício que a tecnologia bulk. Tendo em vista que a diferença entre os dispositivos tipo bulk e SOI consistem na camada BOX, cujo processo de oxidação do silício já é realizado na manufatura dos transistores bulk, é evidente que a tecnologia CMOS SOI não exige modificações na linha de produção. Além disso, a produção de um *wafers* com CMOS SOI para o nó de 22nm é mais barata que

Figura 2.2: Estrutura dos tiristores intrínsecos para o transistor tipo bulk e SOI.



Fonte: SAKURAI, A.MATSUZAWA e T.DOUSEKI[5].

a tecnologia bulk, uma vez que requer menos máscaras litográficas. A integração das duas tecnologias é facilmente atingida corroendo a BOX dos transistores SOI.

Esses dispositivos SOI são diferenciados em duas categorias: completamente ou parcialmente deplecionados. Ambas características serão apresentadas na Subseção 2.3.1.

2.3.1 Transistores PD-SOI e FD-SOI

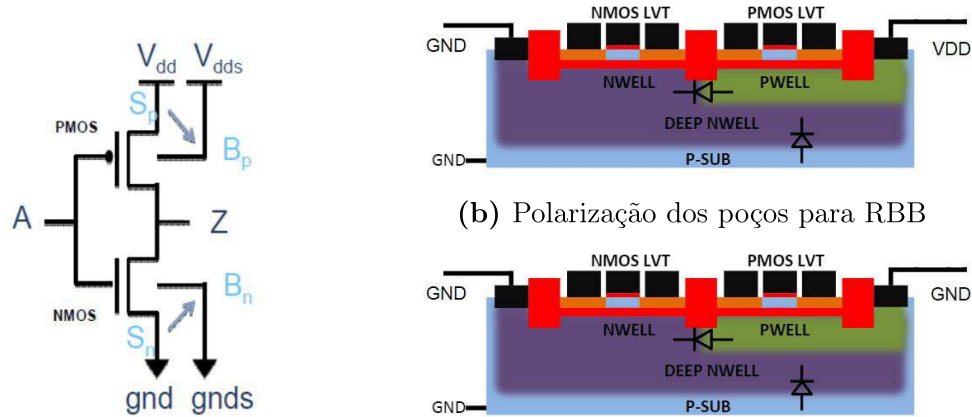
Em situações em que o canal é pequeno o suficiente para que a zona de depleção alcance toda a área entre a porta e a BOX, o transistor é nomeado FD (*fully depleted*). Caso contrário, o dispositivo é nomeado PD (*partially depleted*). Em dispositivos FD-SOI, a profundidade do canal é um terço do canal efetivo entre o dreno e o coletor com o intuito de evitar correntes de fuga. Dessa forma, a tecnologia FD-SOI não necessita dopagem do filme de silício sobre o óxido (SOI), reduzindo tensão de limiar e o efeito RDF em relação aos transistores PD-SOI.

A tensão de limiar V_t de um dispositivo bulk é alcançado com diferentes concentrações de dopagem do canal e largura de porta. Já na tecnologia FD-SOI, o potencial V_t é ajustado utilizando diferentes dopantes no substrato abaixo da BOX. Esse método não só evita o RDF, como também permite a escolha da polaridade do poço entre o dreno e o coletor. Esse poço é denominado corpo do transistor.

Por exemplo, para polarização reversa do corpo (*reverse body bias*, RBB), deve-se aplicar um potencial positivo entre o corpo e o coletor. Seguindo o diagrama ilustrado na Figura 2.3a, RBB é atingido alimentando o poço abaixo da BOX com um potencial V_{DDs} maior que do coletor V_{DD} para o nMOS e um potencial V_{GNDs} menor que do coletor V_{GND}

para o pMOS [8]. Caso os poços abaixo da BOX sejam dopados negativa e positivamente para o nMOS e o pMOS, são criados diodos conforme ilustra a Figura 2.3b. Essa disposição é nomeada poços invertidos (*flipped-wells*), já que possuem dopantes opostos ao canal do transistor tipo bulk na mesma configuração.

Figura 2.3: Diagrama e corte lateral da estrutura CMOS para dispositivos SOI.



(a) Diagrama elétrico CMOS SOI multi- V_t

(c) Polarização dos poços para FBB

Fonte: Adaptado de FLATRESSE, CESANA e CAUCHY[8] e DE STREEL e BOL[9].

Para a RBB, o diodo entre os poços p^+ e n^+ estão diretamente polarizados. Logo, a tensão de limiar de ambos dispositivos nMOS e pMOS é majorada, ao custo da diminuição da corrente de fuga. Esse ajuste nomeado alta tensão de limiar (*high-voltage threshold*, HVT) permite a redução do consumo do CI. No caso inverso, isto é, para os diodo inversamente polarizados, a tensão de limiar de ambos dispositivos nMOS e pMOS é reduzida, ao custo do aumento da corrente de fuga. Esse ajuste nomeado alta tensão de limiar (*low-voltage threshold*, LVT) permite operações mais rápidas.

Essa última condição de operação é nomeada polarização direta do corpo (*forward body bias*, FBB), isto é, quando há um potencial positivo entre o corpo e o coletor. Seguindo o diagrama ilustrado na Figura 2.3a, FBB é atingido alimentando o poço abaixo da BOX com um potencial V_{DDs} menor que do coletor V_{DD} para o nMOS e um potencial V_{GNDs} maior que do coletor V_{GND} para o pMOS [8].

2.4 Considerações finais

A evolução das tecnologias CMOS foi descrita por meio das limitações físicas da miniaturização do transistor tipo bulk em relação a tecnologia SOI. Em seguida, foi apresentado as diferentes configurações de polarização da tecnologia SOI e os demais vantagens da camada isolante BOX — e.g. redução da corrente de fuga, canal não dopado, isolamento lateral das células. O próximo capítulo trata sobre como transformar uma arquitetura lógica para o as camadas de silício e óxido de silício representadas nos transistores FD-SOI de 28nm.

Capítulo 3

Fluxo de projeto de circuitos integrados

Esse capítulo trata da descrição das etapas do mapeamento da lógica de um filtro FIR para a implementação física por transistores FD-SOI de 28nm.

3.1 Etapas do fluxo de projeto

O aprendizado de programação lógica comumente inicia por linguagens a nível de sistemas operacionais, como C, Java ou Python. Contudo, essas linguagens de programação de alto nível não disponibilizam ao processador dados sobre a implementação do programa, tais como portas lógicas, flip-flops, buffers ou conexões entre esses dispositivos.

Com o intuito de descrever como o dado deve ser transmitido entre os registradores de um circuito, é adequado escrever um arquivo RTL (*register transfer level*) via programação em linguagem de descrição de hardware (*hardware description language*, HDL), como Verilog e VHDL. Essas linguagens constroem a abstração das operações lógicas implementadas a nível de bit. É importante diferenciar a HDL da linguagem de máquina: enquanto essa descreve a instrução binária a ser executada pelo processador, aquela representa as conexões físicas entre os diferentes elementos de um circuito lógico.

Entretanto, a compilação das HDLs não analisa se a integridade dos sinais entre os registradores ou a área total ocupada estão apropriadas aos pré-requisitos do projeto. Além disso, não são consideradas as variações ambientais no qual o CI está inserido, como temperatura. Para reportar uma conclusão sobre essas características, é necessária a análise dos

parâmetros físicos do circuito, os quais são nomeados parasitas.

Para inferir os parasitas de uma arquitetura lógica, é preciso medir o tempo e a magnitude entre os sinais de entrada e de saída de cada componente do circuito integrado. Essa análise é realizada por um fluxo de implementação física do circuito integrado, o qual transforma a lógica descrita no arquivo RTL nos poços dopados dos transistores, como ilustrado na [Figura 2.1b](#) e [Figura 2.3c](#). Esse diagrama dos polígonos pode ser exportado em forma de máscaras litográficas e empregado na manufatura do CI.

O passo do fluxo responsável pela transformação do arquivo RTL em portas lógicas é nomeado síntese lógica. Por sua vez, o mapeamento das portas lógicas para o conjunto ótimo de transistores é determinado pela síntese física. Finalmente, a representação dos transistores como polígonos dispostos em camadas de silício oxidadas ou dopadas negativa ou positivamente é nomeado concepção do leiaute do CI.

Tendo em vista que o número de componentes dos circuitos tendem a aumentar, o cronograma de produção de um circuito integrado é insuficiente para um trabalho sequencial das três fases. Dessa forma, as etapas de desenvolvimento de um circuito integrado são sobrepostas para garantir a finalização do CI em tempo hábil para consumo no mercado.

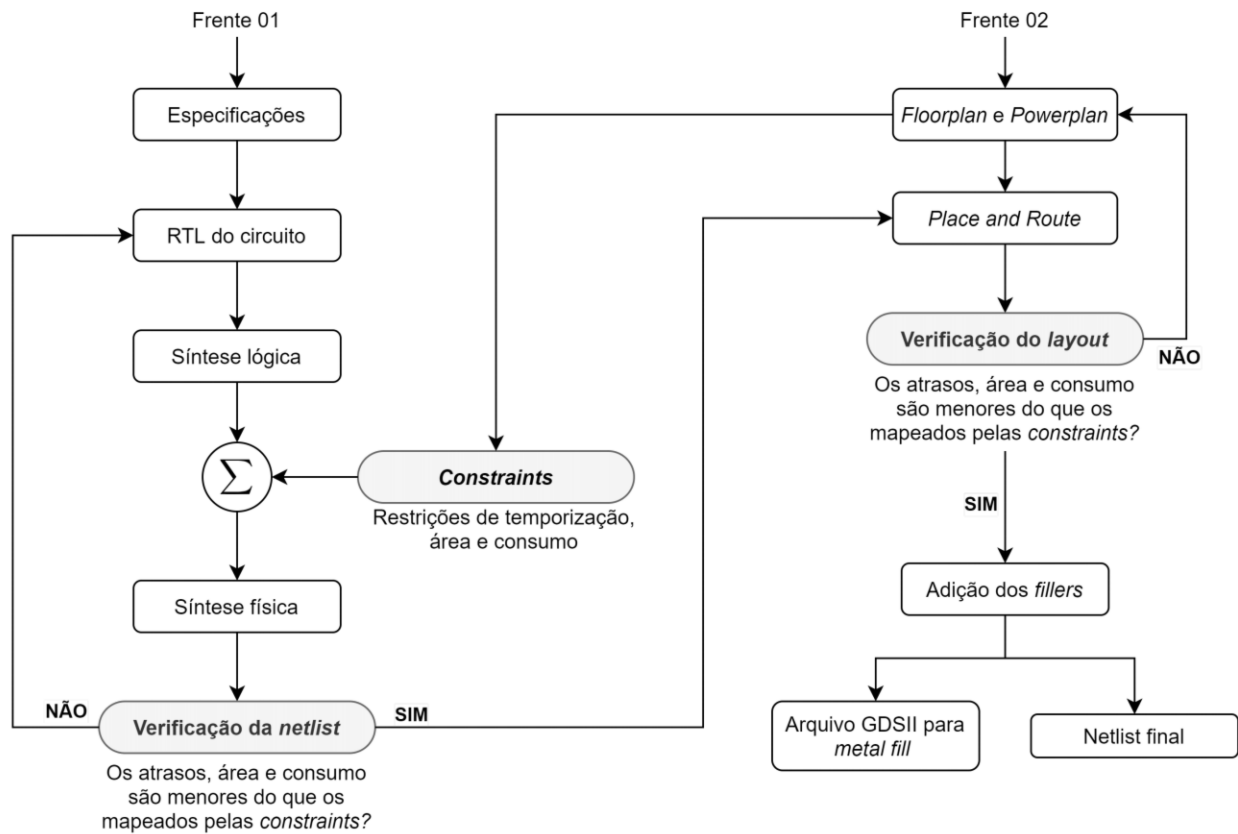
Para o trabalho coeso e paralelo das equipes, é evidente que cada fase possui formatos de entrada e saída bem determinados. Essa rigidez dos arquivos transferidos entre as etapas permitem o ajuste contínuo do fluxo, cujos objetivos principais são diminuir área e consumo e maximizar performance do CI conforme o processo de tecnologia disponível.

O fluxo proposto para este trabalho de estágio está ilustrado na [Figura 3.1](#). A sequência de projeto possui duas frentes: a etapa de síntese e a fase de concepção do tamanho, disposição das trilhas de alimentação do leiaute — em inglês, *floorplan* e *powerplan*, respectivamente. A partir das especificações do cliente, a arquitetura em nível RTL é descrita em HDL Verilog. Considerando que foi realizada uma verificação formal sobre o funcionamento da lógica implementada, segue-se para a síntese lógica, física e testes pós-síntese do circuito. O produto final dessa etapa é a *netlist*, isto é, uma lista de células e das conexões das portas que mapeiam a lógica do RTL na tecnologia CMOS FD-SOI.

Caso a arquitetura implementada opere adequadamente na tensão e frequência descritas na especificação, as células listadas na *netlist* são dispostas no leiaute e conectadas — em inglês, *Place and Route* (P&R). Em seguida, caso os requisitos de área e temporização desse

leiaute finalizado sejam menores ou equivalentes aos especificados pelo cliente, adicionam-se as células de preenchimento de densidade (*filler cells*) ao CI. Para finalizar o fluxo, exporta-se a *netlist* pós-layout e o arquivo GDSII referente às máscaras litográficas de manufatura.

Figura 3.1: Fluxo de projeto de circuito integrado para a tecnologia FD-SOI.



Fonte: Adaptado de [LEITE 10](#).

Neste trabalho, foram utilizadas as ferramentas EDA comerciais com licença estudantil da *Cadence Design Systems* para a construção do fluxo de projeto. Primeiramente, será apresentada a arquitetura do bloco empregado para teste do fluxo na [Seção 3.2](#). Em seguida, será descrito o kit de *design* de processo (*process design kit*, PDK) para FD-SOI de 28nm na [Seção 3.3](#), o qual será utilizado nos experimentos. Por fim, as duas frentes ilustradas na [Figura 3.1](#) referentes, respectivamente, a síntese lógica e física e a concepção do layout serão apresentadas nas [Seção 3.4](#) e [Seção 3.5](#).

3.2 Arquitetura do filtro FIR

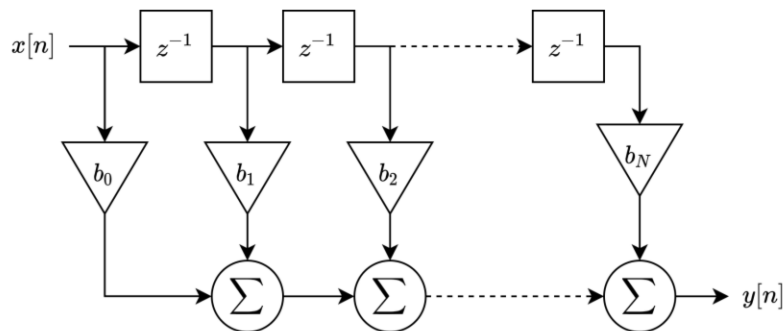
O circuito lógico empregado neste trabalho descreve a arquitetura de um filtro FIR (*finite impulse response*) de 16 bits. A resposta ao impulso de filtro FIR possui duração finita, ou seja, sua saída atinge zero após um tempo finito [10]. As aplicações dos filtros FIR são amplas na área de comunicações ópticas e processamento de sinais digitais.

A escolha do circuito foi justificada pela composição do caminho de dados do filtro FIR. Conforme ilustrado na Figura 3.2, a arquitetura do filtro é formada por um pipeline de N atrasos de amostragem, $N + 1$ multiplicadores e N somadores. A saída $y[n]$ do bloco FIR cuja entrada é nomeada $x[n]$ é:

$$y[n] = \sum_{i=0}^N b_i x[n - i]$$

para n o número da amostra e b_i o coeficiente multiplicativo da entrada x no instante $(n - 1)h$, sendo h o período de amostragem.

Figura 3.2: Diagrama lógico da resposta $y[n]$ de um filtro FIR com entrada $x[n]$ e coeficientes $b_i, \forall i \in \mathbb{Z} \cap (0, N)$.

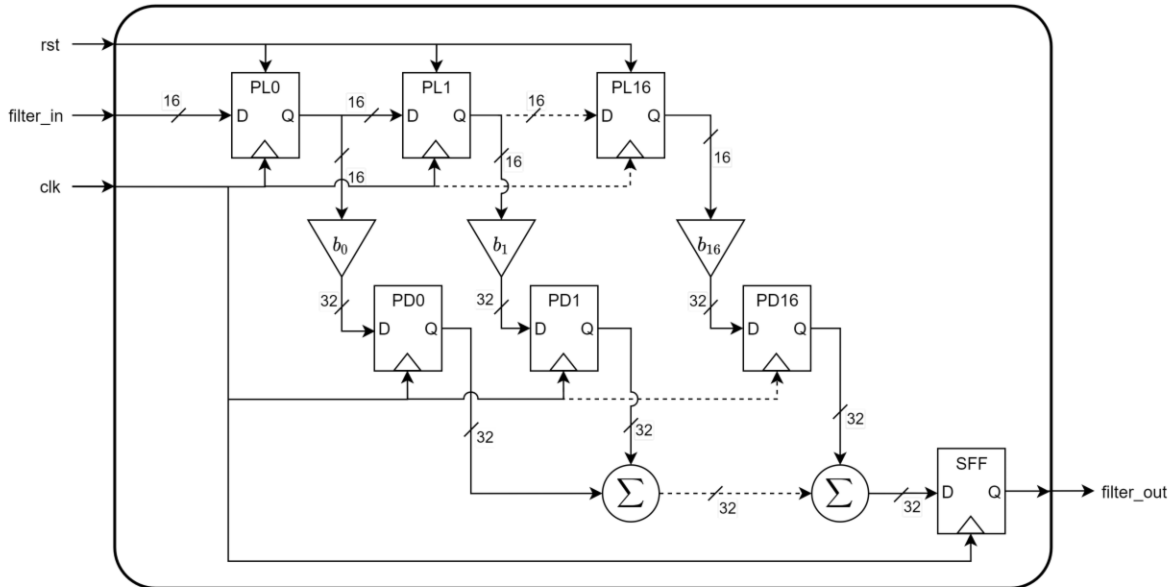


Fonte: Autoria própria.

É importante destacar que os coeficientes b_i e a entrada $x[n]$ possuem 16 bits cada. Dessa forma, a saída $y[n]$ produzida pela multiplicação de b_i por $x[n]$ possui 32 bits. Na Figura 3.3 está ilustrada o esquema do circuito descrito em Verilog.

Observando a Figura 3.3 e comparando-a com a Figura 3.2 fica claro que o sinal de entrada de 16 bits e a resposta do filtro FIR composta por 32 bits são `filter_in` e `filter_out`, respectivamente. O sinal `clk` representa o *clock* de acionamento dos flip-flops do *pipeline PLK* e do produto com os coeficientes *PDK*, para $K \in \mathbb{Z} \cap (0, 16)$. Por sua vez, o registra-

Figura 3.3: Diagrama de blocos da resposta de um filtro FIR com entradas `rst`, `clk` e `filter_in` e saída `filter_out`.



Fonte: Autoria própria.

o sinal *SFF* representa o flip-flop de saída do somatório (*sum flip-flop*). Finalmente, o sinal *rst* denomina o comando de anulação de todos os valores armazenados no *pipeline*, produto ou somador final, ou seja, $Q = 0$ para qualquer flip-flop do circuito.

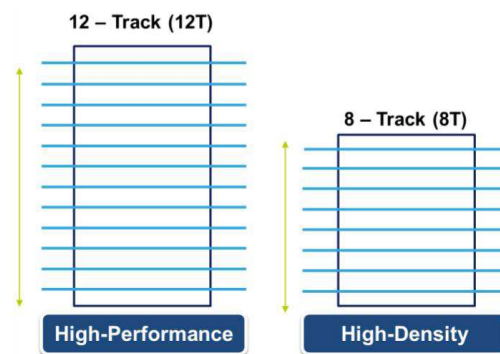
3.3 PDK FD-SOI 28nm

Os transistores da tecnologia FD-SOI 28nm do PDK da *ST Microelectronics* possuem profundidade de canal máxima de 7nm e espessura da BOX de 25nm [8]. Já a tensão de alimentação V_{DD} das células variam de 0.6V a 1.1V [8]. Esse kit da tecnologia FD-SOI também disponibiliza bibliotecas para LVT e HVT, ou seja, com poços regulares ou invertidos.

Apesar do pacote oferecer células com 12 ou 8 larguras de linha (*tracks*), para este projeto estavam disponíveis apenas as células de 12 larguras de linha. A Figura 3.4 ilustra a diferença entre esses dois conjuntos: as células 8T (8 *tracks*) são mais compactas e focadas em CIs de alta densidade, enquanto as células 12T são voltadas às aplicações de alta performance.

Com o intuito de minimizar as perdas, o kit da tecnologia FD-SOI 28nm também disponibiliza quatro diferentes larguras efetivas de canal: $L_{min} + 0nm$, $L_{min} + 4nm$, $L_{min} + 10nm$ e $L_{min} + 16nm$, para L_{min} a largura mínima do canal de 28nm. Ao aumentar o compri-

Figura 3.4: Comprimento das células disponíveis pelo PDK FD-SOI 28nm da *ST Microelectronics* para alta performance 12T (esquerda) e alta densidade 8T (direita).



Fonte: VENKATESH e CHOUDHURY[11].

mento do canal, o transistor se comporta de modo equivalente à processos maiores [10] e, portanto, com maior tensão de limiar e menores correntes de fuga. Essa modificação é exclusiva da tecnologia FD-SOI e consiste na mudança, em tempo de projeto, do comprimento do poli-silício da porta [10].

3.4 Síntese lógica e física

Conforme ilustrado na Figura 3.1, após a descrição do filtro FIR em Verilog, segue-se para a etapa de síntese lógica do circuito. Segundo a Seção 3.1, essa etapa converte a arquitetura descrita em HDL para o conjunto de portas lógicas padrão da tecnologia FD-SOI de 28nm. Neste trabalho, foi empregada a ferramenta *Genus* da *Cadence Design Systems*.

Previamente ao mapeamento das portas lógicas, é necessário indicar à ferramenta o caminho da biblioteca de células padrão, conforme as características descritas na Seção 3.3. Para este trabalho, foram escolhidas as células de 12 *tracks* com poços invertidos e largura de canal efetivo de 28nm. Dessa forma, a biblioteca utilizada não apresenta *polybiasing* e é adequada para polarização LVT-FBB.

Outras configurações devem ser informadas à ferramenta antes da síntese lógica, tais como o caminho para os relatórios pós-síntese, a lista das células que devem ser evitadas no mapeamento e os perfis de otimização de constantes e registradores equivalentes. É importante destacar que, para as simulações e testes pós-síntese de temporização do CI, é necessário indicar a frequência de operação e qual o *jitter* do *clock*. Para os experimentos

descritos neste documento, adotou-se a frequência de 1,5GHz e a incerteza de 6,67ns.

Conforme a documentação do *Genus*, a síntese lógica é separada em três etapas. A primeira fase foca na busca iterativa do melhor custo benefício entre temporização e área do circuito. O produto final dessa fase é uma *netlist* para portas genéricas.

Por sua vez, a segunda etapa da síntese física mapeia as portas genéricas da primeira etapa para células do kit de tecnologia disponível. Segundo a documentação da *Cadence Design Systems*, o *Genus* avalia múltiplas implementações e escolhe a alternativa que não viola a temporização dos sinais enquanto minimiza área e consumo. Nessa etapa, qualquer hierarquia construída no RTL é desagrupada com o intuito de facilitar as otimizações.

Por fim, a última etapa de síntese aprimora a arquitetura após o mapeamento na tecnologia CMOS FD-SOI. Nessa fase, busca-se diminuir o WNS (*worst negative slack*), isto é, o maior tempo de violação. O *slack* equivale à diferença entre o tempo de chegada esperado do dado e o real, de modo que quanto mais negativo este valor for, mais atrasado estará aquele caminho. Dessa forma, uma violação de temporização ocorre quando o valor armazenado no registrador é estável apenas após da subida do *clock* (violação de *setup*) ou modificado para outro valor antes da subida do *clock* (violação de *hold*).

A simulação pós-síntese relata informações sobre a área total ocupada, o consumo de energia estática e dinâmica e violações de temporização. Os dados reportados nas fases da síntese lógica para o filtro FIR estão apresentadas na [Tabela 3.1](#). Para finalizar a síntese lógica, foi exportado o arquivo SDC da *netlist* mapeada para CMOS FD-SOI de 28nm. Esse formato indica as restrições — em inglês, *constraints* — de área e temporização utilizadas para validar o *design*.

Tabela 3.1: Dados de consumo, área e temporização para as simulações pós-síntese de cada fase da síntese lógica e física.

| Fase | | <i>Slack</i> (ps) | Área de célula (μm^2) | Área de <i>net</i> (μm^2) | Número de instâncias | Consumo total (mW) | Perdas (mW) | Comprimento total de roteamento (μm) |
|----------------|---|-------------------|------------------------------------|--|----------------------|--------------------|-------------|---|
| Síntese lógica | A | 106 | 20021,281 | 10,369 | 13314 | - | 0,081 | - |
| | B | 46 | 7012,541 | 2277,081 | 4316 | - | 0,149 | - |
| | C | 0 | 6875,453 | 2211,932 | 4137 | 21,0371 | 0,143 | - |
| Síntese física | A | 83 | 12697,652 | 2937,205 | 6609 | - | 0,108 | 58744 |
| | B | 35 | 7014,336 | 3722,057 | 4263 | - | 0,150 | - |
| | C | 0 | 6884,102 | 3628,080 | 4117 | - | 0,145 | 44657 |
| | D | 3842 | 6899,280 | 2172,792 | 4137 | 19,7025 | 0,145 | 43657 |

Comparando os resultados de cada fase da síntese física, é evidente que a área ocupada

pelas células, o número de instâncias, as perdas e o *slack* diminuem. Essa observação é justificada pelo mapeamento das portas genéricas em transistores da tecnologia CMOS FD-SOI e pelas otimizações de área executadas na fase B e C. A área de *net* é pequena na fase A uma vez que a previsão do roteamento entre as células só é realizado a partir da fase B.

Em seguida, foi executada a síntese física. Para configurar essa etapa, é necessário indicar o arquivo DEF. Esse formato informa o tamanho do leiaute e as disposições das trilhas de alimentação e das células de isolamento no *die*. Essas células são nomeadas *end-caps* e *well-taps*. As *end-caps* estão localizadas ao redor do chip para limitar a resistência entre as trilhas de alimentação e substratos dos transistores FD-SOI. Por sua vez, as *well-taps* estão distribuídas regularmente em colunas e objetivam evitar *latch-up*.

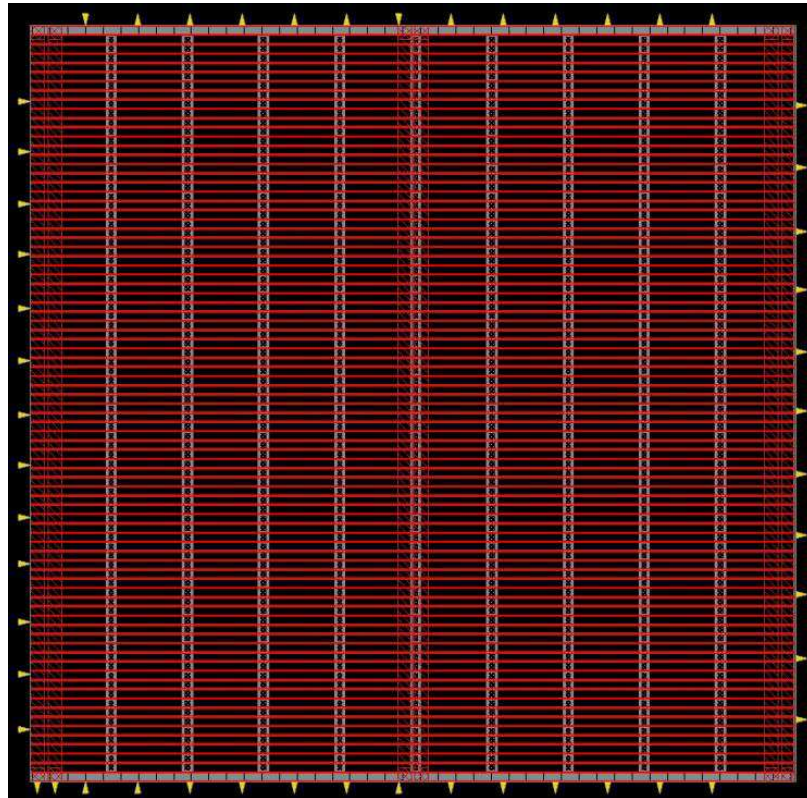
Para o filtro FIR, foi definido um *die* de tamanho $96\mu\text{m}\times 96\mu\text{m}$. É importante destacar que o leiaute foi construído a nível de bloco. Caso fosse considerado a nível de topo, seria necessário espaçar as portas I/O do *die*, ou seja, aplicar uma configuração *core*. Na [Figura 3.5](#) está ilustrado o *floorplan* e *powerplan* construído para o filtro FIR. A concepção desse leiaute foi realizado utilizando a ferramenta *Innovus* da *Cadence Design Systems*.

Há duas tensões de alimentação do circuito, cujas trilhas de alimentação foram dispostas em três colunas igualmente espaçadas. Observando a [Figura 3.5](#), é evidente a localização das células de *end-caps* ao redor do *die* e 9 colunas regulares de *well-taps*. O espaçamento entre as portas de dado do bloco, isto é, dos sinais *filter_in*, *filter_out*, *rst* e *clock* foram considerados equivalentes para cada lado do *die*. Por sua vez, as portas de tensão de alimentação VDD e GND foram designadas abaixo das trilhas vertical à esquerda.

O processo de síntese física é composto por quatro etapas. De modo semelhante à síntese lógica, a primeira fase se refere ao mapeamento da arquitetura à um conjunto de portas lógicas genéricas. Entretanto, na síntese física, a ferramenta *Genus* computa um mapa de congestão conforme um roteamento simples das células da *netlist*. Essa estimativa relata a qualidade do *floorplan* desenhado e indica os a localização dos gargalos de temporização no *die*. Dessa forma, na síntese física, o roteamento do caminho de dados pode ser otimizado antes de mapear as portas genéricas para a tecnologia do PDK.

Analogamente à síntese lógica, a segunda etapa da síntese física mapeia as portas genéricas para portas da tecnologia FD-SOI 28nm. Porém, na síntese física, busca-se minimizar o comprimento dos fios de roteamento entre as células, o qual foi estimado na primeira fase.

Figura 3.5: Disposição das células de isolamento e das trilhas de alimentação do filtro FIR.



Fonte: Autoria própria.

Nessa etapa, a localização dos registradores é mantida constante. Entretanto, é investigada a união de flip-flops que representam a mesma constante ou variável no circuito. A terceira etapa é equivalente na síntese física e lógica.

Por fim, a última etapa da síntese física executa, em plano de fundo do *Genus*, a ferramenta *Innovus*. A disposição das células sobre o *die* — em inglês, *placement* — é realizada e as violações de regra da tecnologia (*design rule violation*, DRV), tais como espaçamento entre trilhas de dados e pinos ou sobreposição de células, são solucionadas. Uma nova *netlist* é retornada para o *Genus* e refinada conforme otimização de temporização e área.

Os resultados da síntese física estão apresentados na Tabela 3.1. Analogamente à síntese lógica, é evidente que a área ocupada pelas células, o número de instâncias e o *slack* diminuem da fase A à fase C. Entretanto, esse comportamento de decréscimo do *slack*, da área ocupada e número de instâncias não ocorre entre as fases C e D. Tendo em vista que a ferramenta *Innovus* é mais pessimista na análise temporal em relação ao *Genus*, as

otimizações de área aplicadas na fase D provocam maiores violações de temporização que na fase C. Para compensar as violações relatadas, células de atraso — em inglês, *buffer* — são adicionadas aos caminhos de *clock*. Dessa forma, aumenta-se o número de instâncias, a área ocupada por células e o *slack* do caminho mais lento.

Observando [Tabela 3.1](#), a partir da fase A até a fase D é minimizado o comprimento total de fios de roteamento. Essa característica não é estimada na síntese lógica. As perdas da fase B são maiores que a fase A em razão da prioridade às correções de DRV.

Comparando os resultados da síntese física e lógica expostos na [Tabela 3.1](#), fica claro que a consideração dos parâmetros físicos do CI — e.g. tamanho das células, congestionamento, camadas de roteamento — permite uma melhor otimização de consumo total e temporização para a mesma quantidade de instâncias finais. Considerando que a análise temporal da síntese física é mais pessimista que da síntese lógica, por decisão de projeto, o aumento das perdas é desconsiderado.

3.5 *Place and Route*

Após o mapeamento das porta lógicas para células-padrão da tecnologia FD-SOI 28nm, inicia-se a disposição das instâncias listadas na *netlist* pós-síntese física no leiaute ilustrado na [Figura 3.5](#). Essas etapa é nomeada *placement* e já foi estimada pela quarta fase da síntese física. Em seguida, é construída a árvore de propagação de *clock*. Por fim, é realizado o roteamento entre as células do leiaute e a correção das violações de temporização.

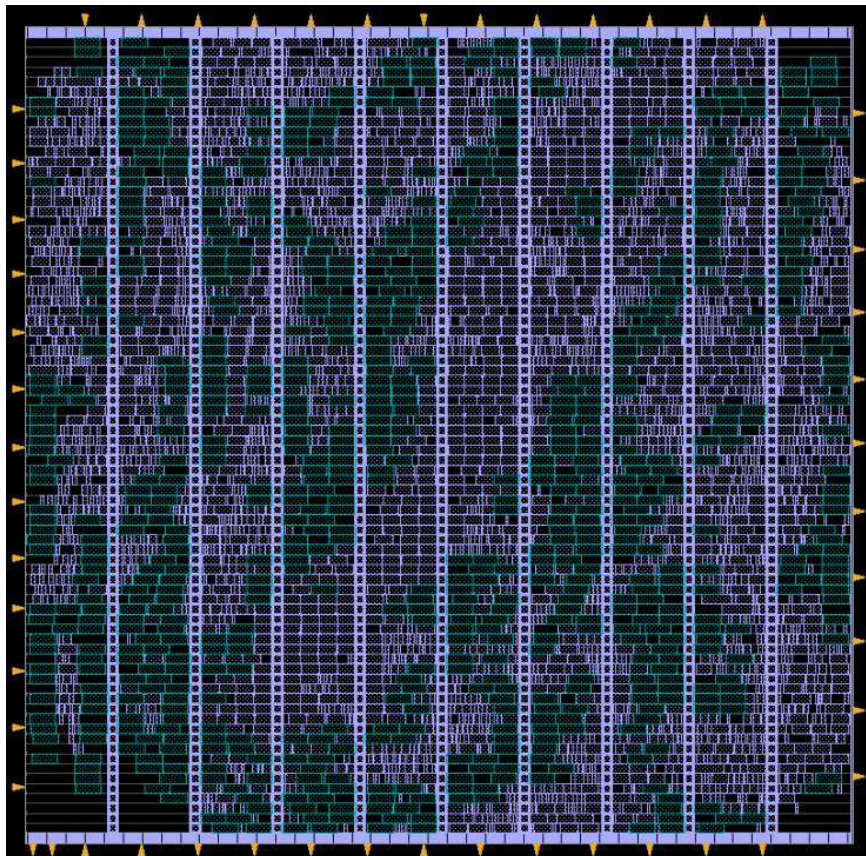
Primeiramente, é necessário configurar a ferramenta *Innovus* para a análise temporal e espacial durante as otimizações do leiaute. Foram consideradas três condições de tensão e temperatura — no inglês, *corners* — conforme apresentado na [Tabela 3.2](#). Esse tipo de ambiente é denominado multimodal/multi-*corner* (MMMC). Para análise mais pessimista, as violações de *setup* e *hold* são consideradas no *corner* lento e rápido, respectivamente. Também devem ser informados ao *Innovus* o caminho para os arquivos da tecnologia, para os relatórios das simulações e para a *netlist* pós-síntese física.

A etapa de *placement* dispõe as células da *netlist* no leiaute considerando as restrições do *floorplan*, tais como localização das *well-taps* e *end-caps*. Nessa fase, busca-se evitar os erros de sobreposição de instâncias, espaçamento pequeno entre trilhas de roteamento e

Tabela 3.2: Condições de temperatura e tensão de alimentação V_{DD} para a análise MMMC.

| Corner | Temperatura (°C) | V_{DD} (V) | Comportamento |
|---------|------------------|--------------|---------------|
| Worst | 125 | 0,90 | Lento |
| Typical | 25 | 1,0 | Típico |
| Best | -25 | 1,1 | Rápido |

curto circuito entre os poços. Dessa forma, estima-se um roteamento inicial entre as células com o intuito de analisar o congestionamento e extrair dos parasitas. O leiaute do filtro FIR pós-*placement* está ilustrado na Figura 3.6.

Figura 3.6: Leiaute do filtro FIR pós *placement* da *netlist*.

Fonte: Autoria própria.

Caso a simulação indique que todas as células estão legalmente posicionadas, é realizada uma otimização de temporização. É importante citar que a localização legal das células no *placement* não garante ausência de DRVs na etapa final do projeto, uma vez que o roteamento será modificado nas iterações de otimização. Nessas rodadas de aprimoramento, a ferramenta do *Innovus* investiga corrigir as DRVs, reduzir a maior violação de temporização,

reduzir a área e minimizar o consumo, nessa ordem.

Na análise temporal, primeiramente são otimizados todos os caminhos em *asetup*. Em seguida, se o menor *slack*, isto é, a maior violação de *setup*, não pertencer a um caminho de dados entre dois registradores, a ferramenta itera uma segunda rodada sobre os caminhos registrador-para-registrador. As técnicas usadas para correção das violações são adição ou remoção de *buffers*, redimensionamento das portas dos transistores, reestruturação da *netlist*, remapeamento lógico das portas, troca de pinos das células-padrão, movimentação de instâncias e fusão ou replicação de registradores.

Em seguida, o sinal do *clock* é roteado na etapa nomeada *Clock Tree Synthesis* (CTS). Esse roteamento é geralmente restrito às camadas mais altas em razão da importância do sinal de *clock* para o correto funcionamento do circuito. As regras de espaçamento para esse roteamento são menores que para roteamento de dados, portanto, devem ser especificadas à ferramenta. Outras informações devem ser indicadas ao *Innovus* para construção da árvore de *clock* ótima, tais como a lista de células de *buffer* e inversores disponíveis no PDK.

Por fim, as células são roteadas em duas etapas. A primeira fase divide o leiaute em retângulos e cria as conexões entre as células de cada área delimitada. O objetivo dessa etapa é minimizar a congestão. Em seguida, a segunda etapa roteia as *nets* entre os retângulos, de modo a evitar a criação de curtos ou violações de espaço.

Para o kit FD-SOI 28nm, foram inseridos diodos nas portas de alguns transistores com o objetivo de corrigir os efeitos de antena. Por fim, foram realizadas otimizações de *hold* e *setup* para garantir *slack* nulo ou positivo para qualquer *corner*.

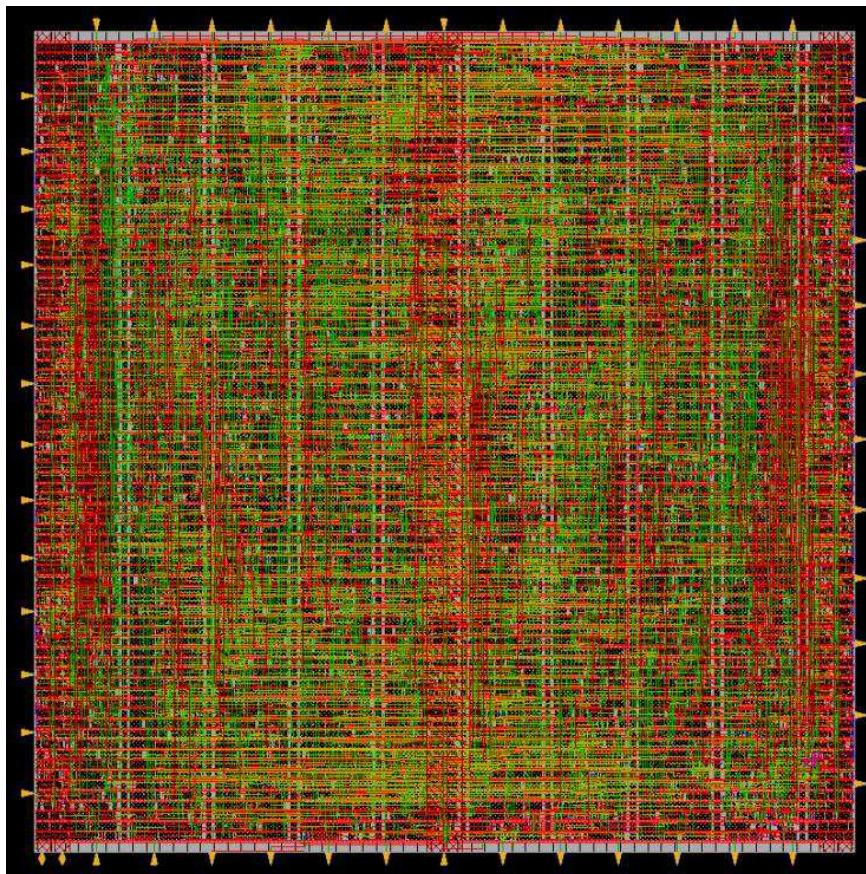
Os resultados para cada etapa do P&R estão listados na [Tabela 3.3](#). A análise de *hold* não é reportada na fase de *placement*, uma vez que ainda não há células de atraso na árvore de *clock*. Comparando a densidade entre as fases, fica evidente o aumento da ocupação do *die* no decorrer do P&R, uma vez que são acrescentadas células para correção de DRVs e efeito de antena. Não foram identificados curtos ou erros de espaçamento no leiaute.

Para o envio do leiaute à manufatura, é necessário preencher os espaços vazios nas camadas com células-padrão denominadas *fillers*. Em seguida, sobrepõe-se uma fina camada de metal com o intuito de satisfazer a densidade mínima de metal para o *wafer*. Por fim, a *netlist* pós-roteamento é exportada juntamente com os parasitas antes do *metal fill*. Esses parasitas são armazenados em arquivos do formato SPEF e podem ser carregados para

Tabela 3.3: Resultados das etapas de P&R para o filtro FIR.

| <i>Etapa P&R</i> | WNS de setup (ns) | Violações de setup | WNS de hold (ns) | Violações de hold | Densidade (%) |
|----------------------|-------------------|--------------------|------------------|-------------------|---------------|
| <i>Placement</i> | -0,002 | 4 | - | - | 86,975 |
| <i>CTS</i> | 0,005 | 0 | -0,034 | 102 | 88,356 |
| <i>Route</i> | 0 | 0 | 0,001 | 0 | 90,402 |

análise em ferramentas especializadas para consumo e temporização, tais como *Voltus* e *Tempus*, respectivamente. A Figura 3.7 ilustra o leiaute pós adição das *fillers*.

Figura 3.7: Leiaute do filtro FIR pós P&R.

Fonte: Autoria própria.

3.6 Considerações finais

O fluxo de projeto de circuitos integrados foi descrito conforme a funcionalidade, entradas e saídas de cada etapa. As fases de síntese lógica e física transformam lógica a nível de RTL e

em HDL para a arquitetura de portas lógicas. Essas portas lógicas são mapeadas em poços da tecnologia FD-SOI com largura de porta 28nm na fase de *placement*, CTS e roteamento. Essas duas frentes do fluxo de projeto se relacionam a partir das *constraints*, isto é, das especificações de atraso do *clock*, tamanho do *die* e localização das trilhas de alimentação do CI. O fluxo foi aplicado sobre a arquitetura de um filtro FIR e os resultados obtidos identificaram um circuito sem violação de temporização ou DRVs. O próximo capítulo busca otimizar o fluxo proposto a partir da especificação do posicionamento do caminho de dados do filtro FIR.

Capítulo 4

Otimização do caminho de dados

Esse capítulo trata da otimização da estrutura de caminho de dados no fluxo de projeto de circuitos integrados para minimização de consumo e aprimoramento da temporização.

4.1 Estrutura de caminho de dados

Com o objetivo de melhorar a performance, o consumo e a ocupação de um leiaute, é possível especificar o posicionamento das células do caminho de dados. Essa técnica de otimização é nomeada estrutura do caminho de dados (*structured data path*, SDP). O caminho de dados — em inglês, *data path* — se refere ao conjunto de registradores que conectam a entrada à saída de uma arquitetura.

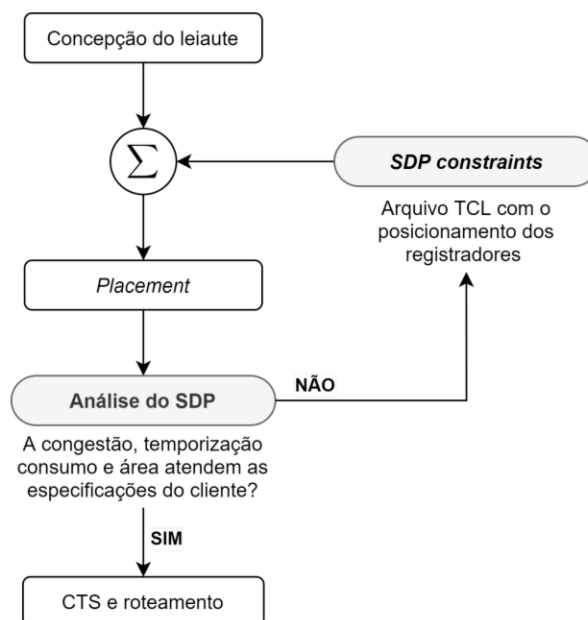
A especificação correta do posicionamento dos registradores garante o roteamento uniforme na área disponível do leiaute. Logo, o método SDP deve ser aplicado quando o caminho de dados é denso, a performance não atingiu a especificação do usuário ou o tempo para lançamento da arquitetura no mercado não permite um fluxo personalizado. [11].

No caso do filtro FIR, o leiaute final é denso, uma vez que a taxa final de ocupação do *die* é aproximadamente 90%, conforme a Tabela 3.3. Tendo em vista que a densidade máxima para a fabricação do CI é 95%, aplicou-se o SDP para minimizar a taxa de ocupação. No filtro FIR, o *data path* é composto pelo *pipeline* de células de atraso, a matriz de multiplicação e o somador de saída. Esses blocos estão representados na Figura 3.3 como *PL*, *PD* e *SFF*, respectivamente.

Considerando que a metodologia do SDP é realizada por intervenção manual, é necessário conhecimento prévio da relação entre velocidade, consumo e área. Nos experimentos realizados, foi definido que a melhor distribuição das células do *pipeline* é por contiguidade, isto é, alinhamento lateral. Analogamente, o melhor posicionamento dos registradores *PD* é paralelamente às linhas do *pipeline*. Dessa forma, o roteamento dos sinais de dados é restrito às camadas mais baixas e o roteamento de *clock*, que pertence às camadas superiores, possui maior espaço para otimização.

O fluxo de SDP está ilustrado na [Figura 4.1](#). A etapa de especificação do posicionamento dos registradores é informada ao *Innovus* antes da fase de *placement* por meio de um arquivo TCL. A ferramenta, por sua vez, interpreta as posições especificadas pelo usuário nesse arquivo e alinha as células-padrão no leiaute. Caso o leiaute final pré-CTS não obedeça os limites máximos de congestão, temporização, consumo e área, deve-se refazer o fluxo para um novo posicionamento do caminho de dados.

Figura 4.1: Fluxo de desenvolvimento do SDP no fluxo de projeto de concepção e simulação física do leiaute.



Fonte: Autoria própria.

É importante destacar que a regularidade do nome dos sinais na *netlist* minimiza o número de iterações de busca da ferramenta e o tempo de execução. Na [Seção 4.2](#) está descrita a aplicação do método SDP para a arquitetura de um filtro FIR.

4.2 SDP para o filtro FIR

Na [Figura 4.2](#) está ilustrado o caminho de dados desenvolvido para o filtro FIR pós-*placement*. Primeiramente, na [Figura 4.2a](#), é possível observar a contiguidade horizontal dos 17 níveis do *pipeline*. Para o *pipeline*, cada coluna representa o conjunto de 16 registradores referentes à uma célula de atraso *PL* apresentada na [Figura 3.3](#).

De modo análogo, a [Figura 4.2b](#) ilustra a disposição da matriz de multiplicação. Cada coluna da matriz representa os registradores *PD* do filtro FIR. Considerando que a *netlist* foi otimizada por área e congestão na síntese física, as constantes b_i foram ampliadas para 20 bits por replicação de registradores. De modo semelhante, os primeiros 5 bits de cada constante, isto é, as 5 primeiras linhas da matriz possuem valores equivalente e foram, portanto, unificados.

Por decisão de projeto, optou-se por centralizar os registradores das colunas restantes com o intuito de facilitar o *placement* das células *tie-hie* ou *tie-low* no leiaute. Essas células são mantidas em 0 ou 1 e simbolizam os valores das constantes multiplicativas b_i .

Por fim, na [Figura 4.2c](#) estão apresentados os bis do registrador *SFF*, o qual é responsável por atualizar o sinal de saída de 32 bits `filter_out`. Cada registrador está localizado próximo da sua respectiva porta com o objetivo de minimizar o roteamento.

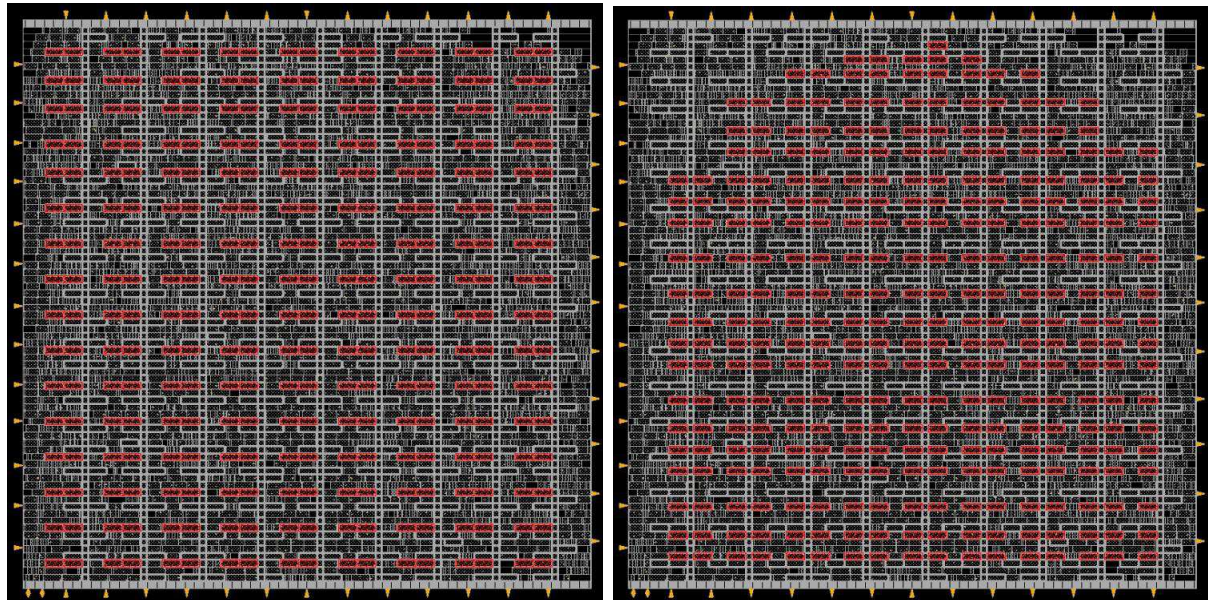
Para finalização do leiaute, foi aplicado o restante do processo de P&R descrito na [Seção 3.5](#). De modo semelhante à [Tabela 3.3](#), os resultados coletados para o filtro FIR com o método SDP estão apresentados na [Tabela 4.1](#). Além disso, na [Figura 4.3](#) estão apresentados os mapas de distribuição de densidade para o leiaute com e sem a especificação do SDP.

Tabela 4.1: Resultados das etapas de P&R para o filtro FIR com SDP.

| <i>Etapa P&R</i> | WNS de setup (ns) | Violações de setup | WNS de hold (ns) | Violações de hold | Densidade (%) |
|----------------------|-------------------|--------------------|------------------|-------------------|---------------|
| <i>Placement</i> | -0,001 | 1 | - | - | 83,906 |
| <i>CTS</i> | 0 | 0 | -0,034 | 75 | 89,08 |
| <i>Route</i> | 0 | 0 | 0,001 | 0 | 91,012 |

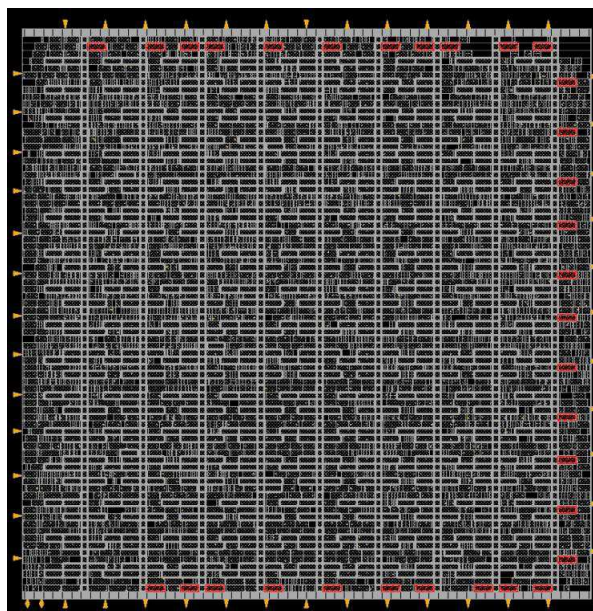
Comparando os resultados ilustrados na [Figura 4.3a](#) e [Figura 4.3b](#) com e sem a aplicação do SDP, é evidente que a distribuição da densidade é mais homogênea quando a posição dos

Figura 4.2: Estruturas dos registradores na especificação do SDP.



(a) Pipeline

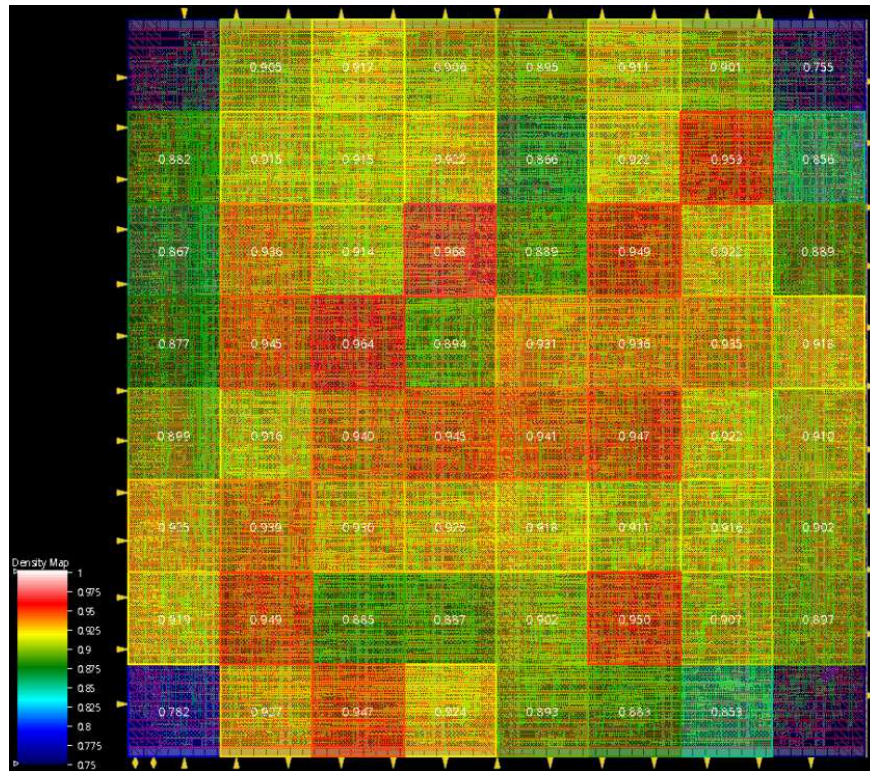
(b) Matriz de multiplicação



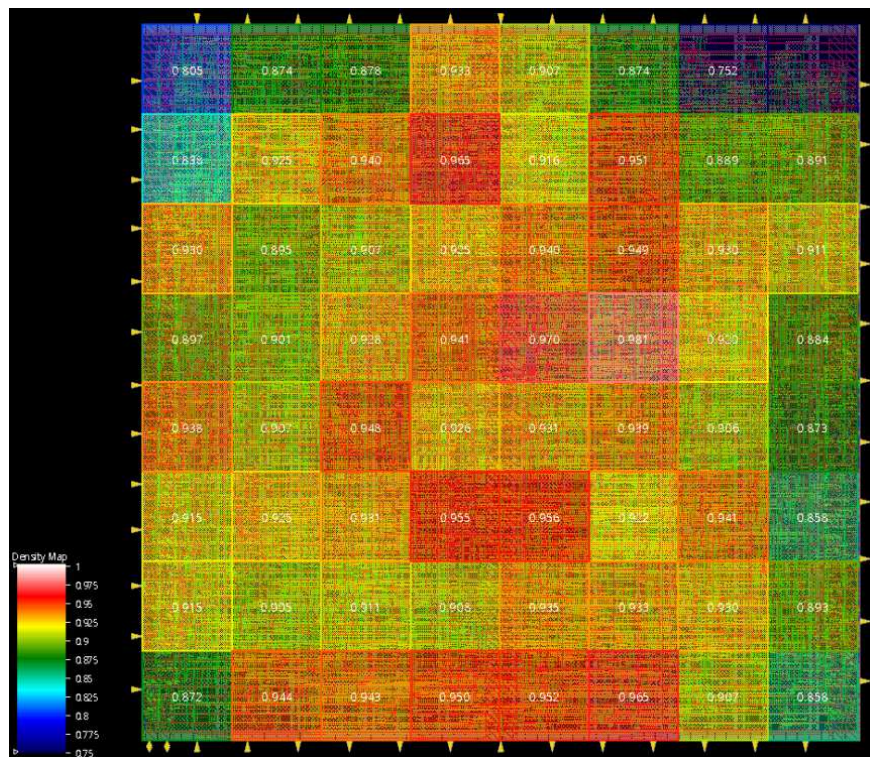
(c) Somador final

Fonte: Autoria própria.

Figura 4.3: Distribuição de densidade no leiaute com e sem SDP.



(a) Sem SDP



(b) Com SDP

Fonte: Autoria própria.

registradores é especificada. Entretanto, comparando os dados apresentados na [Tabela 3.3](#) e [Tabela 4.1](#), é claro que a taxa média de ocupação do leiaute aumentou.

Analisando os relatórios da síntese do filtro FIR, foi verificada que as iterações de otimização do *placement* foram bloqueadas por erros de sobreposição com os registradores da matriz de multiplicação. Logo o SDP foi reestruturado considerando apenas o grupo do *pipeline* e do somador de saída ilustrados na [Figura 4.2a](#) e [Figura 4.2c](#) respectivamente.

Os resultados coletados para o filtro FIR com novo SDP estão apresentados na [Tabela 4.2](#). Por sua vez, o mapa de densidade para o novo leiaute está ilustrado na [Figura 4.4](#).

Tabela 4.2: Resultados das etapas de P&R para o filtro FIR com o SDP reestruturado.

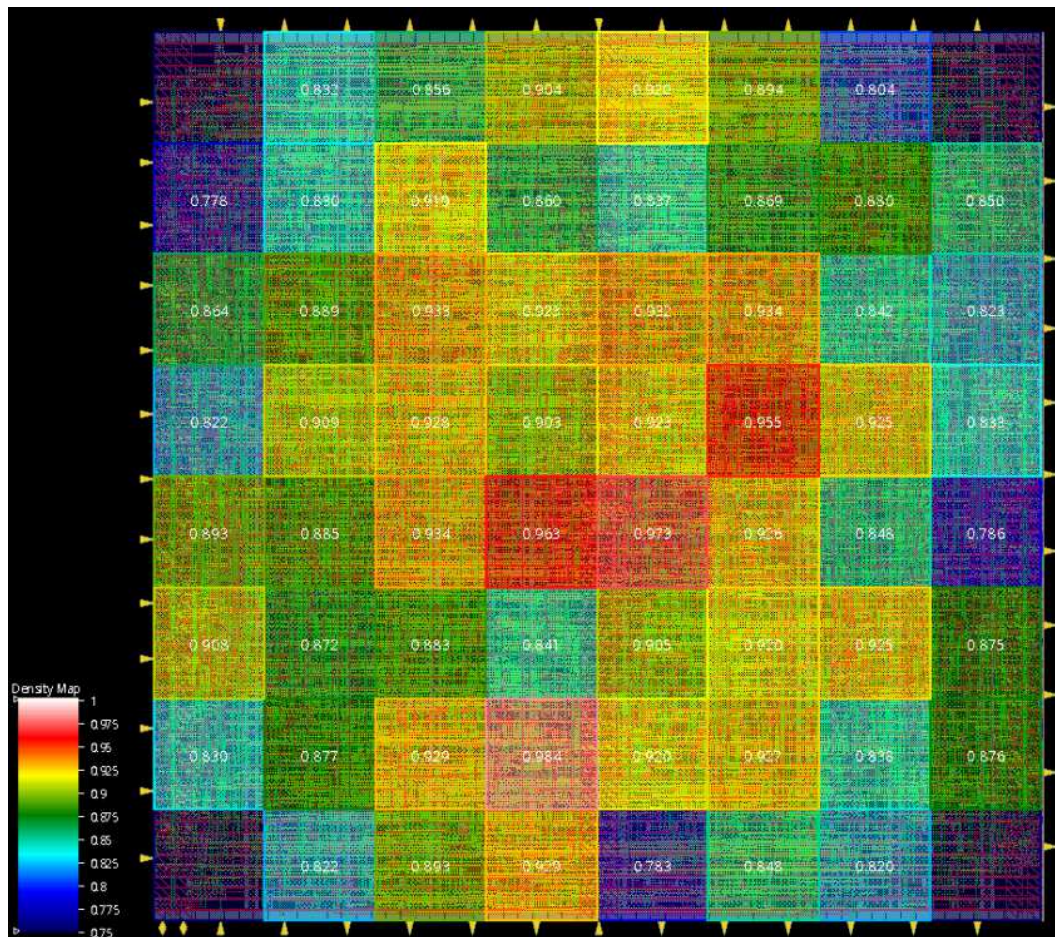
| <i>Etapa P&R</i> | WNS de setup (ns) | Violações de setup | WNS de hold (ns) | Violações de hold | Densidade (%) |
|----------------------|-------------------|--------------------|------------------|-------------------|---------------|
| <i>Placement</i> | -0,003 | 7 | - | - | 86,25 |
| <i>CTS</i> | 0 | 0 | -0,034 | 82 | 88,63 |
| <i>Route</i> | 0 | 0 | 0 | 0 | 90,394 |

Comparando a [Figura 4.3a](#) com a [Figura 4.4](#), é possível observar a uniformização da densidade para o novo SDP em relação ao leiaute original. Além disso, contrapondo as [Figura 4.3b](#) e [Figura 4.4](#), é evidente que a homogeneização do roteamento é maior quando a matriz de multiplicação não é congelada no algoritmo de *placement*.

Entretanto, ao comparar as [Tabela 3.3](#), [Tabela 4.1](#) e [Tabela 4.2](#), é claro que a densidade obtida é menor para o leiaute sem SDP. Para obter uma conclusão sobre o impacto do SDP sobre a performance do circuito, é preciso analisar o consumo de cada leiaute. A potência estática, dinâmica e perdida para cada fluxo de projeto e para cada *corner* está apresentada na [Tabela 4.3](#). O número de iterações de correções de temporização para finalização do leiaute na etapa de roteamento também estão contabilizadas na [Tabela 4.3](#).

Observando a [Tabela 4.3](#), é possível concluir que o consumo de potência total do leiaute cresce para os projetos sem SDP, com SDP parcial e para o SDP englobando a matriz de multiplicação, nessa ordem. Uma vez que a regularidade do SDP não minimizou o roteamento, houve o aumento da capacitância dos fios e consequente aumento de potência dinâmica. As perdas e a potência estática se mantêm aproximadamente constantes para os três fluxos em cada *corner*. Logo, é possível desconsiderar o impacto dessas mudanças sobre a potência total em relação às contribuições da variação da potência dinâmica.

Figura 4.4: Distribuição de densidade no leiaute como o SDP reestruturado.



Fonte: Autoria própria.

Tabela 4.3: Resultados de consumo e otimização de tempo para todos os *corner* empregando os três fluxos de concepção de leiaute.

| <i>Corner</i> /fluxo | | Potência estática (mW) | Potência dinâmica (mW) | Perdas (mW) | Potência total (mW) | Correções de <i>setup</i> | Correções de <i>hold</i> |
|----------------------|-----------------|---------------------------|---------------------------|----------------|------------------------|------------------------------|-----------------------------|
| <i>Worst</i> | Sem SDP | 10,63 | 6,83 | 0,90 | 18,35 | 7 | - |
| | Com SDP | 10,81 | 7,64 | 0,90 | 19,35 | 3 | |
| | Com SDP parcial | 10,82 | 7,30 | 0,91 | 19,02 | 1 | |
| <i>Typical</i> | Sem SDP | 17,02 | 11,32 | 0,08 | 28,42 | - | - |
| | Com SDP | 17,32 | 12,59 | 0,08 | 29,98 | | |
| | Com SDP parcial | 17,32 | 12,05 | 0,08 | 29,44 | | |
| <i>Best</i> | Sem SDP | 13,43 | 8,88 | 0,16 | 22,47 | - | 8 |
| | Com SDP | 13,66 | 9,90 | 0,16 | 23,71 | | 4 |
| | Com SDP parcial | 13,67 | 9,46 | 0,16 | 23,29 | | 1 |

Por sua vez, o número de iterações para correção de violações temporais diminuiu com a inclusão do SDP. Tendo em vista que o roteamento para células alinhadas ocorre nas camadas mais baixas, a densidade das camadas superiores diminuiu. Essas camadas são empregadas para o roteamento dos sinais de *clock* na fase do CTS. Dessa forma, o método SDP facilita o ajuste da árvore de *clock* para correção das violações de temporização.

Tendo em vista a discussão dos resultados, concluiu-se que os experimentos com SDP não produziram impacto sobre a performance do filtro FIR. Nos experimentos escolhidos, as vantagens pelo método são apenas visíveis no fluxo de desenvolvimento, visto a redução no número de iterações de otimização na etapa de roteamento.

Uma alternativa para ampliar a análise do caminho de dados seria construir um terceiro arquivo de SDP incluindo apenas a matriz de multiplicação e o registrador de saída. Esses resultados podem informar se o melhor posicionamento das células do pipeline são paralelamente ou ortogonalmente à matriz do produto. Em seguida, pode-se implementar uma arquitetura com caminho de dados tradicional sem otimização de constantes, como um filtro FIR dinâmico, tendo em vista que a arquitetura e SPDs implementado são demasiadamente pequenos para a observação dos benefícios do método na performance do CI.

Por fim, é interessante expandir a análise de consumo para o bloco implementado. Para tal, pode-se propor dois experimentos. Primeiramente, o bloco FIR pode ser implementado sobre um processo maior. O resultado esperado é menor potência dinâmica e perda por correntes de fuga.

Os experimentos propostos não foram executados visto finalização da carga horária.

4.3 Considerações finais

Nesse capítulo, foi apresentado o método de otimização de área e performance SDP. Essa técnica consiste no congelamento das posições dos registradores do caminho de dados conforme especificação do usuário. O SDP foi aplicado em duas configurações para o filtro FIR de 16 bits. Os resultados apresentados não indicaram otimização da densidade para o fluxo com SDP em relação ao fluxo sem SDP. Entretanto, as violações de *hold* ou *setup* do leiaute com SDP foram corrigidas em um menor número de iterações, uma vez que o roteamento das células alinhadas por contiguidade ocorre nas camadas mais baixas da tecnologia. Para observar os efeitos do SDP na ocupação de um *die*, poderia-se refazer o experimento para a arquitetura de um filtro FIR dinâmico. Analogamente, para investigar o impacto do método SDP no consumo do leiaute, pode-se comparar os resultados obtidos para o filtro FIR em um processo com maior largura de porta, ou seja, cuja tensão de limiar é maior que para as células-padrão LVT FD-SOI.

Capítulo 5

Conclusão

Neste relatório de projeto, foram discutidas as dificuldades para miniaturização do transistor tipo bulk. Com base nas limitações físicas desse dispositivo, foi apresentado o comportamento e configurações de polarização da tecnologia FD-SOI. Com essa fundamentação teórica, foi possível definir os parâmetros para construção do fluxo de projeto de um circuito integrado para transistores com largura de canal de 28nm.

Para tal, foi primeiramente construída a arquitetura de um filtro FIR com coeficientes constantes. O processo de transformação da lógica programada em portas lógicas foi realizado empregando linguagem de descrição de *hardware* para representar o transporte das variáveis em nível de registrador. Em seguida, essas portas lógicas foram mapeadas para células-padrão da tecnologia FD-SOI com base no pacote de *design* da *ST Microelectronics*.

Para a execução correta do fluxo nas ferramentas da *Cadence Design Systems*, foram investigados e especificados os parâmetros da tecnologia às ferramentas de síntese lógica, física e concepção de leiaute. As decisões de projeto, tal como dimensão do *die*, disposição e potencial das trilhas de alimentação, foram realizadas conforme experiência da equipe implementação física do laboratório XMEN.

A otimização do leiaute final para aplicação em circuitos de alta densidade e baixo consumo foi efetuada via estruturação do caminho de dados do filtro FIR. Uma vez que a união e replicação das constantes da arquitetura foi adotada durante a síntese lógica, a regularidade do caminho de dados foi prejudicada. Dessa forma, resultados esperados sobre o consumo ou área do CI não puderam ser observados nos experimentos executados.

Tendo em vista a finalização da carga horária do estágio, experimentos futuros para ampliação da análise da arquitetura com estruturação do caminho de dados foram exclusivamente propostos. Entretanto, os ensaios realizados consolidaram o fluxo de projeto para a tecnologia FD-SOI 28nm. Como contribuição desse estágio, as variáveis da tecnologia necessárias para o correto funcionamento das ferramentas e o exemplo de aplicação do fluxo de projeto proposto estão documentadas para uso posterior em projetos do laboratório XMEN.

Referências bibliográficas

- 1 MOORE, G. E. **Cramming more components onto integrated circuits**, Reprinted from *Electronics*, volume 38, number 8, April 19, 1965, pp.114 ff. *IEEE Solid-State Circuits Society Newsletter*, v. 11, n. 3, p. 33–35, 2015.
- 2 ARDEN, W. et al. **More than Moore**. *International Technology Roadmap for Semiconductors*, p. 3–11, 2015. Disponível em: http://www.itrs2.net/uploads/4/9/7/7/49775221/irc-itrs-mtm-v2_3.pdf. Acesso em: 12 de abril, 2021.
- 3 PATTERSON, D.; HENNESSY, J. *Organização e Projeto de Computadores*. 5. ed. Rio de Janeiro: Elsevier, 2017. 1-44 p. ISBN 978-8535287936.
- 4 CHENG, K.; KHAKIFIROOZ, A. **Fully depleted SOI (FDSOI) technology**. *Sci China Inf Sci*, v. 59, n. 6, Abril 2016. Disponível em: <https://doi.org/10.1007/s11432-016-5561-5>. Acesso em: 12 de abril, 2021.
- 5 SAKURAI, T.; A.MATSUZAWA; T.DOUSEKI. *Fully-Depleted SOI CMOS Circuits and Technology for Ultralow-Power Applications*. 1. ed. Holanda: Springer, 2006. 23-42 p. ISBN 978-0-387-29217-5.
- 6 VITALE, S. A.; W.WYATT, P.; CHECKA, N.; KEDZIERSKI, J.; .KEAST, C. L. **FDSOI Process Technology for Subthreshold Operation Ultralow-Power Electronics**. *Proceedings of the IEEE*, v. 98, n. 2, p. 333–342, 2010. Disponível em: <http://dx.doi.org/10.1109/jproc.2009.2034476>. Acesso em: 12 de abril, 2021.
- 7 DENNARD, R. H.; GAENSSKEN, F. H.; YU, H.; RIDEOUT, V. L.; BASSOUS, E.; LEBLANC, A. **Design of Ion-Implanted MOSFET's with Very Small Physical Dimensions**. *Proceedings of the IEEE*, v. 87, n. 4, p. 668–678, 1999. ISSN 0018–9219/99.
- 8 FLATRESSE, P.; CESANA, G.; CAUCHY, X. *Planar fully depleted silicon technology to design competitive SOC at 28nm and beyond*. Genova, 2012. Disponível em: https://www.soitec.com/pdf/planar_fd_silicon_technology_competitive_soc_28nm.pdf. Acesso em: 12 de abril, 2021.
- 9 DE STREEL, G.; BOL, D. **Study of Back Biasing Schemes for ULV Logic from the Gate Level to the IP Level**. *Journal of Low Power Electronics and Applications*, v. 4, p. 168–187, 2014. Disponível em: <https://www.mdpi.com/2079-9268/4/3/168>. Acesso em: 12 de abril, 2021.

- 10 LEITE, T. ***Fluxo de Projeto de Circuitos Integrados em Tecnologia FD-SOI 28nm***. Trabalho de conclusão de curso em Engenharia Elétrica — Universidade Federal de Campina Grande, Campina Grande, 2015.
- 11 VENKATESH, N. S.; CHOUDHURY, B. **Best-in-Class Standard-Cell Libraries for HighPerformance, Low-Power and High-Density SoC Design in 28nm FD-SOI Technology**. *Technology Marketing, STMicroelectronics Pvt.*, 2015. Disponível em: <https://bit.ly/3v4L6ve>. Acesso em: 12 de abril, 2021.