

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE – UFCG
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA – CEEI
DEPARTAMENTO DE ENGENHARIA ELÉTRICA – DEE

Valmir Ferreira da Silva

Relatório de Estágio Supervisionado no Laboratório de Excelência em Microeletrônica do Nordeste - XMEN

Campina Grande - PB
26 de março de 2026

Valmir Ferreira da Silva

Relatório de Estágio Supervisionado no Laboratório de Excelência em Microeletrônica do Nordeste - XMEN

Relatório de Estágio Supervisionado submetido à Coordenadoria de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. Gutemberg Gonçalves dos Santos Júnior, D.Sc.

Campina Grande - PB

26 de março de 2026

Valmir Ferreira da Silva

Relatório de Estágio Supervisionado no Laboratório de Excelência em Microeletrônica do Nordeste - XMEN

Prof. Gutemberg Gonçalves dos Santos Júnior, D.Sc.
Orientador

Prof. Marcos Ricardo Alcantara Morais, D.Sc.
Convidado

Campina Grande - PB
26 de março de 2026

A meus pais e meus irmãos, que sempre me apoiaram.

Agradecimentos

Minha jornada foi marcada por pessoas especiais que estiveram presentes em cada etapa, sempre prontas a me apoiar. Tenho plena consciência de que, sem elas, eu não estaria onde estou hoje. A todas, dedico os meus mais sinceros agradecimentos.

Aos meus pais e meus irmãos, pilares e porto seguro da minha vida: vocês acreditaram em mim desde o primeiro instante e sei que estarão comigo até o fim. Seus ensinamentos, carinho, apoio e acolhimento foram fundamentais para que eu pudesse seguir em frente e superar todas as dificuldades. Vocês me inspiram diariamente a ser uma pessoa melhor.

Aos meus amigos de Campina Grande — Arthur Henrique, Juliana Pessoa, Lucas Ventura —, obrigado por compartilhar tantos momentos felizes dentro e fora da universidade. Vocês me ajudaram a perceber que os estudos não definem toda a minha vida, mas fazem parte dela. Com vocês, aprendi que é possível ir além dos portões da universidade, viver intensamente e desfrutar de todas as emoções que a vida proporciona.

Aos meus amigos do XMEN — Luiz, Joseilton, Ícaro, Davi, Maria, Cleisson, Túlio, Pedro Henrique, Pedro Arthur, Lucas, Agripino e José Ewerton —, minha gratidão por se tornarem uma família escolhida. Obrigado por estarem sempre ao meu lado, me apoiando, animando e fazendo com que eu me sentisse acolhido. Vocês terão sempre um lugar especial no meu coração, não apenas pelos bolos, risadas e encontros, mas pelo companheirismo, amizade e conquistas que vivemos juntos. Tenho certeza de que ainda alcançaremos muito mais.

Aos professores Gutemberg, Marcos e Georgina, agradeço pela orientação, pelo conhecimento compartilhado e pelo apoio constante. Vocês me proporcionaram conquistas valiosas dentro e fora da academia. Mais do que mestres, foram companheiros e amigos, dividindo risadas, histórias e vitórias. A inspiração que vocês me deram certamente também alcança muitas outras pessoas.

Resumo

Este relatório apresenta as atividades desenvolvidas durante o estágio supervisionado realizado no Laboratório de Excelência em Microeletrônica do Nordeste (XMEN), com foco no estudo e na aplicação do fluxo de implementação física de circuitos digitais. O trabalho teve como estudo de caso o núcleo de processamento RISC-X, um processador acadêmico compatível com a arquitetura RISC-V, desenvolvido no próprio laboratório. Inicialmente, foram analisados os fundamentos do fluxo de projeto de circuitos integrados, abrangendo a síntese lógica, a verificação de equivalência entre a descrição RTL e a netlist sintetizada, e as etapas de *Physical Design*, incluindo *floorplanning*, distribuição de alimentação, posicionamento de células, síntese da árvore de clock e roteamento. Também foram desenvolvidos *scripts* para automatizar etapas do fluxo, contribuindo para sua reprodutibilidade e organização. Os resultados de síntese mostraram que o aumento da frequência de operação implica crescimento no consumo de potência e na área ocupada, além da redução da margem temporal. Na implementação física, foi obtido o layout completo do circuito e realizadas análises de área, potência e temporização. Os resultados evidenciaram consumo de potência moderado, ocupação de área compatível com o porte do núcleo e a existência de violação de temporização no caminho crítico para a frequência de 100 MHz, indicando a necessidade de otimizações adicionais. De modo geral, o estágio proporcionou uma compreensão prática das principais etapas envolvidas no desenvolvimento físico de circuitos integrados digitais, bem como dos desafios associados à obtenção de um projeto funcional e otimizado para fabricação.

Palavras-chave: RISC-V; circuitos integrados; síntese lógica; implementação física; microeletrônica.

Abstract

This report presents the activities carried out during the supervised internship at the Laboratory of Excellence in Microelectronics of the Northeast (XMEN), focusing on the study and application of the physical implementation flow of digital circuits. The work adopted the RISC-X processing core as a case study, an academic processor compatible with the RISC-V architecture and developed within the laboratory itself. Initially, the fundamentals of the integrated circuit design flow were analyzed, covering logical synthesis, equivalence checking between the RTL description and the synthesized netlist, and the stages of *Physical Design*, including *floorplanning*, power distribution, cell placement, clock tree synthesis, and routing. Scripts were also developed to automate stages of the flow, contributing to its reproducibility and organization. The synthesis results showed that increasing the operating frequency leads to growth in power consumption and occupied area, as well as a reduction in timing margin. In the physical implementation stage, the complete circuit layout was obtained and analyses of area, power, and timing were performed. The results indicated moderate power consumption, area occupation compatible with the size of the core, and the existence of a timing violation in the critical path for the 100 MHz operating frequency, indicating the need for further optimizations. Overall, the internship provided a practical understanding of the main stages involved in the physical development of digital integrated circuits, as well as the challenges associated with achieving a functional and optimized design for fabrication.

Keywords: RISC-V; integrated circuits; logic synthesis; physical implementation; microelectronics.

Lista de ilustrações

Figura 1 – Fachada Embedded Lab 12
Figura 2 – Fluxo de síntese lógica e arquivos de entrada utilizados no processo de síntese digital 16
Figura 3 – Estimativa de consumo de potência obtido após a síntese 17
Figura 4 – Potência total em função da frequência de operação 18
Figura 5 – Resultados de temporização obtidos após a síntese lógica 19
Figura 6 – Resultados de temporização obtidos após a síntese lógica 20
Figura 7 – Fluxo de verificação de equivalência lógica. 22
Figura 8 – Conformal verificação de equivalência lógica 23
Figura 9 – Arquivos de entrada e saída do Innovus 24
Figura 10 – Estrutura inicial do floorplan do circuito. 25
Figura 11 – Rede de distribuição de alimentação com power rings e stripes. 26
Figura 12 – Posicionamento inicial das células padrão. 27
Figura 13 – Estrutura inicial da árvore de clock gerada durante o CTS. 28
Figura 14 – Árvore de clock após otimizações de CTS. 28
Figura 15 – Roteamento preliminar do circuito. 29
Figura 16 – Layout do circuito após o roteamento completo. 30
Figura 17 – Interface de análise de violações de regras de projeto (DRC). 31

Lista de tabelas

Tabela 1 – Métricas de área e potência obtidas após a síntese. 18
Tabela 2 – Métricas de temporização da síntese para diferentes frequências. 19
Tabela 3 – Resumo de área do circuito após implementação física 32
Tabela 4 – Composição do consumo de potência do circuito 32
Tabela 5 – Distribuição de potência por tipo de bloco 33
Tabela 6 – Resumo do pior caminho crítico identificado 33

Lista de abreviaturas e siglas

ASIC Application-Specific Integrated Circuit

CSR Control and Status Register

CTS Clock Tree Synthesis

DRC Design Rule Check

DSP Digital Signal Processor

EDA Electronic Design Automation

LEF Library Exchange Format

LUT Look-Up Table

LVS Layout Versus Schematic

PDK Process Design Kit

RISC-V Reduced Instruction Set Computing Five

RFVF RISC-V Formal Verification Framework

RISCV-DV RISC-V Random Instruction Generator (DV)

RTL Register Transfer Level

RVFI RISC-V Formal Interface

RVVI RISC-V Verification Interface

SoC System-on-Chip

STA Static Timing Analysis

TIE Tie High / Tie Low Cells

UART Universal Asynchronous Receiver/Transmitter

UVM Universal Verification Methodology

VHSIC Very High-Speed Integrated Circuit

Sumário

1 INTRODUÇÃO 11
1.1 Objetivo Geral 11
1.2 Objetivos Específicos 11

2 O LABORATÓRIO XMEN 12

3 O NÚCLEO DE PROCESSAMENTO 13

4 SÍNTESE LÓGICA 15
4.1 Fluxo de Síntese 15
4.2 Automação do Fluxo 17
4.3 Relatórios de Síntese 17

5 EQUIVALÊNCIA LÓGICA 22

6 FLUXO DE PHYSICAL DESIGN 24

7 RESULTADOS 32
7.1 Área do Circuito 32
7.2 Análise de Potência 32
7.3 Análise de Temporização 33

8 CONCLUSÃO 34

REFERÊNCIAS 35

1 Introdução

O estágio supervisionado constitui uma etapa fundamental na formação do engenheiro, permitindo a aplicação prática dos conhecimentos adquiridos ao longo da graduação. Durante essa etapa, o estudante tem a oportunidade de vivenciar o ambiente de desenvolvimento tecnológico e compreender o fluxo de projeto utilizado na indústria de circuitos integrados.

Neste contexto, o presente relatório descreve as atividades desenvolvidas no Laboratório XMEN, com foco no estudo e aplicação do fluxo de implementação física de circuitos digitais. O trabalho foi realizado utilizando como estudo de caso um núcleo de processamento desenvolvido no próprio laboratório, permitindo a análise das diferentes etapas do projeto, desde a síntese lógica, equivaleria lógica e por fim a etapa de design físico.

1.1 Objetivo Geral

O objetivo geral do estágio foi compreender e aplicar o fluxo de implementação física de circuitos digitais utilizando como estudo de caso um núcleo de processamento desenvolvido no Laboratório XMEN.

1.2 Objetivos Específicos

- Estudar o fluxo de projeto de circuitos integrados digitais.
- Criar um conjunto de *scripts* para futuras replicações e melhorias.
- Realizar a síntese lógica de um núcleo de processador.
- Executar verificação de equivalência lógica entre *RTL* e *netlist*.
- Realizar as etapas do fluxo de *Physical Design*.
- Avaliar métricas de potência, temporização e área.

2 O Laboratório XMEN

O Laboratório de Excelência em Microeletrônica do Nordeste conta com oito anos de experiência e está localizado no prédio do Laboratório de Sistemas Embarcados e Computação Pervasiva, situado na Rua Aprígio Veloso, 882, Campina Grande - PB, CEP 58429-900. Na Figura 1, apresenta-se a fachada da edificação onde se encontra o escritório da empresa.

Figura 1 – Fachada Embedded Lab



Fonte: Embedded Lab.

O laboratório é composto por uma equipe de alunos, um grupo de pesquisadores e três professores. Os alunos estão em processo de aprendizagem e aprimoramento de conhecimentos, participando de atividades técnicas e aulas teóricas expositivas. Os pesquisadores atuam no desenvolvimento de soluções inovadoras na área de microeletrônica, tanto analógica quanto digital. Já os professores desempenham o papel de orientar os pesquisadores e transmitir conhecimento aos alunos. Entre as principais atividades do laboratório destacam-se as parcerias com grandes empresas, como a RFIDO Design e a IDEA Sistemas Eletrônicos, além do trabalho contínuo em pesquisa e inovação em colaboração com a entidade Virtus-CC. É importante salientar que o laboratório oferece uma ampla gama de oportunidades, proporcionando aos seus integrantes contato com empresas nacionais e internacionais, bem como interação com universidades de diversas partes do mundo

3 O Núcleo de Processamento

O RISC-X é um núcleo de processamento desenvolvido no Laboratório XMEN da Universidade Federal de Campina Grande como parte de uma iniciativa acadêmica voltada ao ensino e à pesquisa em arquitetura de computadores e microeletrônica. O projeto foi fortemente inspirado no livro *Digital Design and Computer Architecture: RISC-V Edition* de Harris e Harris (HARRIS; HARRIS, 2021) e no processador CV32E40P desenvolvido pela OpenHW Group (OpenHW Group, 2022). Seu objetivo é implementar uma arquitetura compatível com o conjunto de instruções RISC-V, ao mesmo tempo em que serve como uma plataforma educacional para experimentação em projeto, verificação e integração de sistemas digitais.

O núcleo adota uma arquitetura de 32 bits baseada em um pipeline clássico de cinco estágios: *Instruction Fetch*, *Instruction Decode*, *Execute*, *Memory* e *Writeback*. A execução ocorre de forma *in-order*, seguindo o modelo tradicional de processadores RISC. Apesar da estrutura clássica, o projeto incorpora suporte a extensões modernas da especificação RISC-V. Entre os principais diferenciais do RISC-X está a flexibilidade de configuração do núcleo, permitindo habilitar ou desabilitar extensões como C (instruções comprimidas), F (operações de ponto flutuante de precisão simples) e M (multiplicação e divisão). Essa parametrização facilita a exploração de diferentes configurações arquiteturais e cenários de projeto (HARRIS; HARRIS, 2021).

Além do conjunto básico de instruções, o núcleo implementa a extensão Zicsr, responsável por operações sobre registradores de controle e status (CSRs). Entre os CSRs disponíveis estão registradores importantes como *misa*, *mtvec*, *mepc*, *mcause* e *mstatus*, que permitem configurar o comportamento do processador, lidar com exceções e controlar o fluxo de execução. Embora algumas funcionalidades ainda estejam em desenvolvimento, como o suporte completo ao registrador *mstatus* e ao mecanismo de interrupções, o núcleo já é capaz de tratar exceções básicas, como instruções ilegais ou acessos desalinhados, redirecionando a execução para o endereço configurado em *mtvec* (OpenHW Group, 2022).

O desenvolvimento do RISC-X também adotou boas práticas de projeto digital, visando facilitar sua integração em sistemas mais complexos, como futuros sistemas em chip (SoC). Nesse contexto, o núcleo foi projetado para possibilitar a conexão com barramentos padronizados e periféricos adicionais.

Um aspecto central do projeto foi a verificação do processador. O RISC-X passou por verificação formal utilizando o framework RISC-V Formal (WOLF; GLASER, 2017), que permite validar matematicamente a conformidade das instruções implementadas com a especificação da arquitetura. Além disso, foi realizada verificação funcional dinâmica

baseada em *Universal Verification Methodology* (UVM), utilizando o gerador de instruções RISCV-DV (Google, 2023). Esse gerador cria programas de teste em assembly que são executados tanto no núcleo quanto em simuladores de referência, como o Spike, permitindo comparar os resultados e detectar inconsistências no comportamento do processador.

O ambiente de verificação também utilizou a RVVI (*RISC-V Verification Interface*), uma evolução da interface RVFI voltada para ambientes dinâmicos de verificação. Essa interface gera registros detalhados das instruções executadas pelo núcleo, permitindo sua comparação com modelos de referência e facilitando a construção de *testbenches* escaláveis e reutilizáveis (WOLF; GLASER, 2017).

4 Síntese Lógica

A síntese lógica é uma das etapas mais importantes no fluxo de projeto de circuitos digitais, pois representa o momento em que uma descrição abstrata em linguagem de descrição de hardware (*Hardware Description Language* HDL), como VHDL, Verilog ou SystemVerilog, é mapeada em células lógicas básicas associadas a uma tecnologia de fabricação específica. Essa transformação é fundamental, pois permite a transição de um nível de projeto mais conceitual no qual o foco está na descrição funcional do sistema e no atendimento a um conjunto de especificações e restrições para uma implementação concreta que possa ser fabricada ou programada em um dispositivo físico, como um circuito integrado do tipo ASIC (*Application-Specific Integrated Circuit*) ou um FPGA (*Field-Programmable Gate Array*) (WOLF, 2008).

Durante o processo de síntese, a descrição em nível de registrador-transferência (*Register Transfer Level* – RTL) é analisada e transformada em uma representação estrutural composta por portas lógicas e elementos sequenciais disponíveis em uma biblioteca tecnológica específica. Esse processo envolve diversas etapas internas, como otimização lógica, mapeamento tecnológico e geração de uma *netlist* que descreve as interconexões entre os componentes do circuito (ROTH; JOHN, 2004; BROWN, 2013).

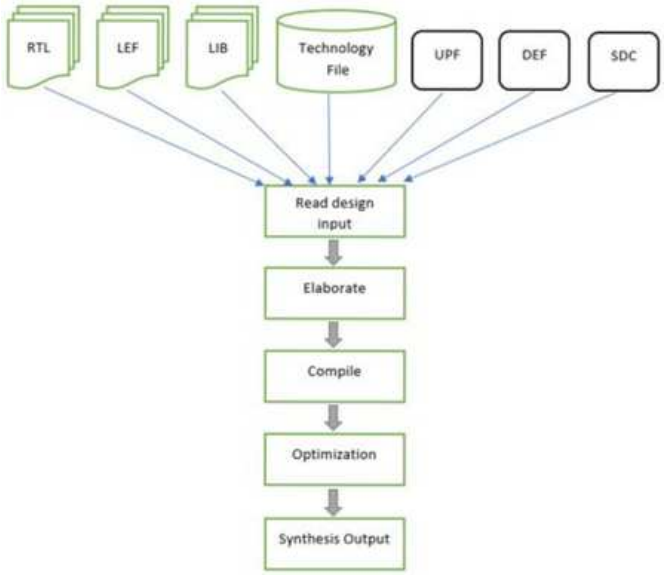
Além de preservar a funcionalidade descrita no modelo RTL, a síntese lógica deve garantir que o projeto atenda a importantes critérios de projeto, tais como desempenho temporal, consumo de potência e utilização de área. Para isso, são consideradas restrições fornecidas pelo projetista, incluindo requisitos de temporização, frequência de operação e limites de utilização de recursos. Dessa forma, a etapa de síntese desempenha um papel central no fluxo de desenvolvimento de sistemas digitais, pois estabelece a base estrutural sobre a qual as etapas posteriores de implementação física serão realizadas (WOLF, 2008; KAHNG et al., 2011).

4.1 Fluxo de Síntese

O fluxo de síntese envolve diversas etapas, incluindo a leitura da descrição RTL, bem como arquivos auxiliares necessários para a implementação do circuito. Entre esses arquivos encontram-se os arquivos LEF (Library Exchange Format), que descrevem aspectos físicos das células padrão; LIB, que contêm os modelos temporais e elétricos das bibliotecas; o technology file, que define parâmetros da tecnologia de fabricação; além de arquivos de restrição e descrição física como DEF (Design Exchange Format) e SDC (Synopsys Design Constraints). Esses arquivos são referenciados ao sintetizador na etapa inicial denominada

Read Design Input, conforme ilustrado na Figura 2.

Figura 2 – Fluxo de síntese lógica e arquivos de entrada utilizados no processo de síntese digital



Fonte: Adaptado de Shah (2022)

Após a leitura dos arquivos de entrada, o sintetizador realiza a etapa de Elaboration, na qual a descrição RTL é analisada e convertida em uma representação estrutural intermediária do projeto. Em seguida, na etapa de Compile, essa representação é mapeada para células da biblioteca tecnológica definida nos arquivos LIB. Posteriormente ocorre a fase de Optimization, onde são aplicadas técnicas para atender às restrições de temporização, área e potência especificadas no arquivo SDC. Na etapa de Scan Insertion, são adicionadas estruturas de teste, como cadeias de scan, para melhorar a testabilidade do circuito. Por fim, é gerado o Synthesis Output, correspondente ao netlist sintetizado que será utilizado nas etapas posteriores do fluxo de projeto digital.

Para o projeto desenvolvido no laboratório, foram aplicadas apenas restrições relacionadas à frequência máxima de operação do circuito. Nesse contexto, a síntese lógica foi realizada considerando o *PDK X-FAB* de 180 nm para os cenários de temporização, com períodos de clock correspondentes às frequências de 50 MHz, 80 MHz, 100 MHz e 120 MHz. Essa abordagem permite avaliar o comportamento do projeto sob diferentes requisitos de desempenho, possibilitando analisar como as métricas de temporização, área e potência variam em função das restrições impostas durante o processo de síntese.

4.2 Automação do Fluxo

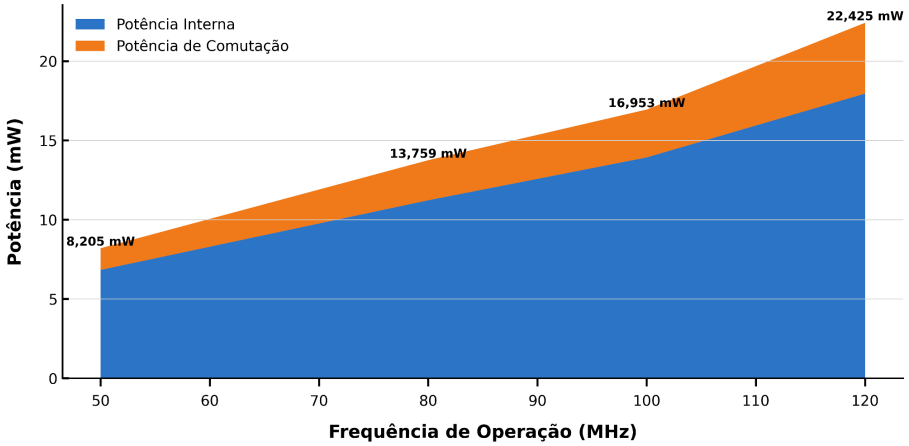
Com o objetivo de facilitar a replicação do fluxo de projeto, foram desenvolvidos *scripts* responsáveis por automatizar as etapas de síntese lógica. Esses scripts permitem executar o fluxo completo de forma reprodutível e organizada.

4.3 Relatórios de Síntese

Os relatórios de síntese apresentados permitem observar com clareza o impacto do aumento da frequência de operação sobre as principais métricas do projeto, em especial potência, área e temporização. De modo geral, os resultados indicam um comportamento esperado para fluxos de síntese orientados por restrições temporais: à medida que a frequência-alvo é elevada, o sintetizador tende a empregar células mais rápidas, realizar redimensionamento de portas e inserir recursos adicionais de otimização, o que melhora o atendimento temporal, porém com aumento de área e de consumo de potência.

A Figura 3 mostra que a potência total cresce progressivamente com a frequência de operação.

Figura 3 – Estimativa de consumo de potência obtido após a síntese



Fonte: Própria.

Observa-se também que a maior parcela do consumo corresponde à potência interna, enquanto a potência de comutação representa uma fração menor, embora também crescente. Esse comportamento pode ser justificado pela Equação 4.1, a qual demonstra que a potência de comutação é diretamente proporcional à frequência de operação. Além disso, o aumento da potência interna está relacionado ao fato de que a elevação da frequência intensifica a atividade dinâmica do circuito e, ao mesmo tempo, exigir o uso de células com maior capacidade de acionamento, contribuindo para o aumento do consumo interno. Assim, tanto a potência de comutação quanto a potência interna tendem a crescer com o aumento

da frequência, embora a segunda permaneça como a principal responsável pelo consumo total observado.

$$P_{dinâmica} \, \alpha \, C \cdot V^2 \cdot f \tag{4.1}$$

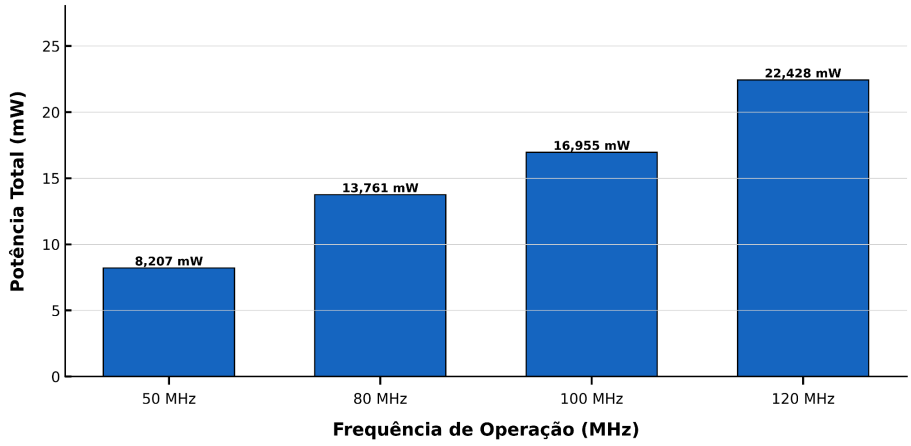
Pela Tabela 1, a potência total passa de 8,207 mW em 50 MHz para 22,428 mW em 120 MHz, o que representa um crescimento bastante expressivo ao longo da faixa analisada. Além disso, nota-se que a potência interna permanece como componente dominante em todas as frequências avaliadas, evidenciando que o custo energético das otimizações empregadas na síntese é significativo.

Tabela 1 – Métricas de área e potência obtidas após a síntese.

Frequência (MHz)	Área das Células (μm^2)	Área de Interconexão (μm^2)	Área Total (μm^2)	Potência Interna (mW)	Potência de Comutação (mW)	Potência Total (mW)
50	168927.955	70729.080	239657.035	6.839	1.366	8.207
80	178805.759	77458.951	256264.710	11.231	2.528	13.761
100	186980.544	83537.169	270517.712	13.933	3.020	16.955
120	204929.790	93890.694	298820.484	17.960	4.466	22.428

A Figura 4 reforça essa tendência ao apresentar a evolução monotônica da potência total em função da frequência. O crescimento não é apenas proporcional ao aumento da taxa de operação, mas também reflete o esforço adicional do processo de síntese ao incluir células maiores e que necessitam de mais energia para satisfazer restrições temporais mais severas . Em outras palavras, ao reduzir o período de clock, o sintetizador precisa tornar os caminhos críticos mais rápidos, o que geralmente implica aumento de capacitância comutada, maior complexidade da rede lógica e maior consumo energético.

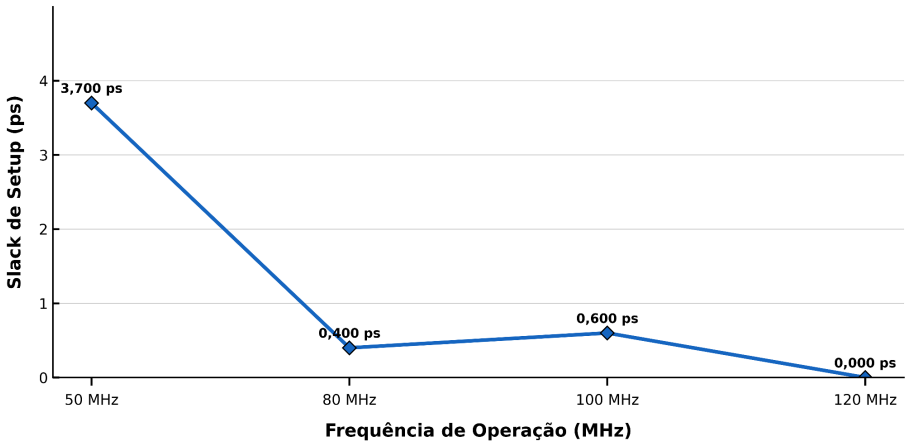
Figura 4 – Potência total em função da frequência de operação



Fonte: Própria.

Do ponto de vista temporal, os resultados da Figura 5 e da Tabela 2 mostram que o circuito atende às restrições em todas as frequências avaliadas, uma vez que o *Total Negative Slack* (TNS) permaneceu nulo e não houve caminhos violados. Entretanto, a margem temporal diminui à medida que a frequência aumenta. Em 50 MHz, o projeto apresenta folga confortável, com *slack* de 3,700 ps, indicando ampla margem para operação. Em 80 MHz e 100 MHz, o *slack* reduz-se para 0,400 ps e 0,600 ps, respectivamente, valores muito pequenos, mas ainda positivos. Já em 120 MHz, o *slack* atinge 0,000 ps, caracterizando uma condição limite, na qual o circuito ainda satisfaz a restrição imposta, porém sem margem adicional. Isso sugere que 120 MHz representa, dentro do cenário analisado, a região de operação máxima atingida pela síntese sem ocorrência de violações temporais.

Figura 5 – Resultados de temporização obtidos após a síntese lógica



Fonte: Própria.

Tabela 2 – Métricas de temporização da síntese para diferentes frequências.

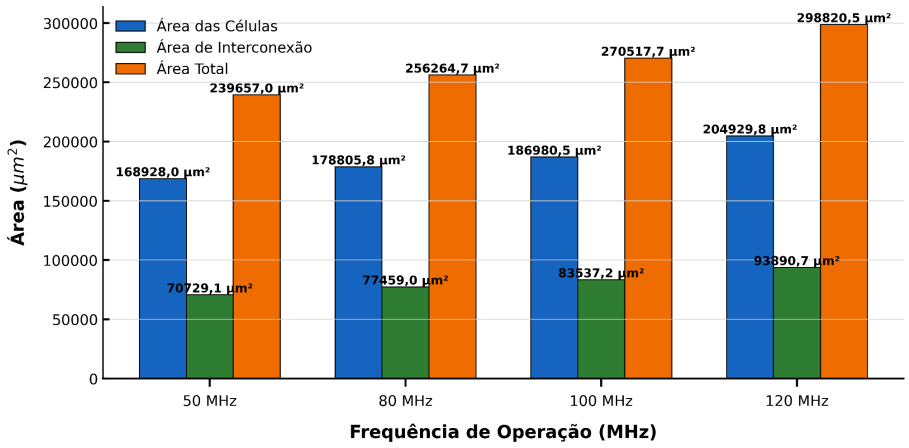
Frequência (MHz)	Período (ns)	Slack (ps)	TNS (ps)	Caminhos Violados
50	20.000	3.700	0.000	0
80	12.500	0.400	0.000	0
100	10.000	0.600	0.000	0
120	8.300	0.000	0.000	0

Esse resultado é importante porque evidencia um compromisso clássico de projeto entre desempenho e robustez temporal. Embora o circuito possa operar a 120 MHz segundo a síntese lógica, a ausência de folga indica que etapas posteriores do fluxo, como posicionamento, roteamento e extração parasítica, podem introduzir degradações suficientes para causar violações. Assim, do ponto de vista de projeto, frequências intermediárias, como

80 MHz ou 100 MHz, tendem a oferecer uma relação mais equilibrada entre desempenho, consumo e margem temporal.

Com relação à área, a Figura 6 correspondente e a Tabela 1 mostram que tanto a área das células quanto a área de interconexão aumentam com a frequência. A área total evolui de 239657,035 μm^2 em 50 MHz para 298820,484 μm^2 em 120 MHz. Esse crescimento confirma que o atendimento de restrições temporais mais agressivas exige maior uso de recursos físicos, seja por substituição de células por versões mais rápidas e maiores, seja por alterações estruturais realizadas durante a otimização lógica. Nota-se ainda que a área de interconexão também cresce de forma consistente, indicando que o impacto da síntese não se restringe apenas à lógica combinacional e sequencial, mas também afeta a complexidade estimada das conexões do circuito.

Figura 6 – Resultados de temporização obtidos após a síntese lógica



Fonte: Própria.

A análise conjunta dos relatórios permite concluir que o aumento da frequência de operação impõe penalidades simultâneas em potência e área. Em contrapartida, proporciona maior desempenho até o limite em que a restrição temporal pode ser satisfeita. Nesse contexto, a escolha da frequência de operação não deve considerar apenas o atendimento temporal isolado, mas também o custo associado em consumo de potência e utilização de área. Para este projeto, os resultados sugerem que frequências intermediárias podem representar um ponto de compromisso mais adequado, enquanto 120 MHz configura uma condição de máximo desempenho sintético, porém com margem temporal nula e maior custo físico.

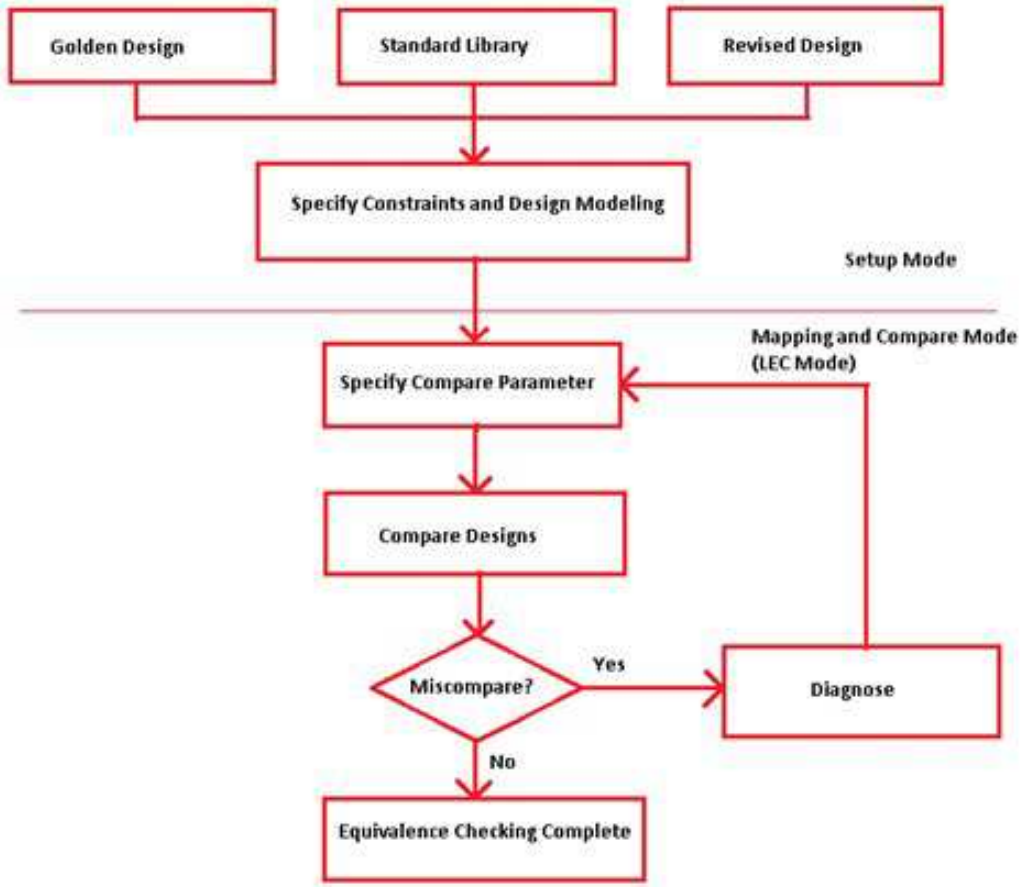
Por fim, os relatórios de síntese demonstram que o fluxo adotado foi capaz de gerar implementações temporalmente válidas em toda a faixa estudada, além de fornecer métricas consistentes para apoiar decisões de otimização. Essas informações são fundamentais para orientar as próximas etapas do fluxo de projeto, especialmente o *physical design*, no qual

será possível verificar se as tendências observadas na síntese lógica se mantêm após a consideração mais realista dos efeitos físicos do circuito.

5 Equivalência Lógica

A verificação de equivalência lógica tem como objetivo garantir que a implementação sintetizada mantém o mesmo comportamento funcional da descrição original em nível RTL. A Figura 7 apresenta o fluxo de verificação de equivalência lógica utilizando a ferramenta *Cadence Conformal*. Nesse processo, a descrição RTL do circuito é comparada com a *gate netlist* gerada após a síntese, garantindo que ambas sejam funcionalmente equivalentes. A figura também destaca os principais arquivos utilizados no fluxo, como bibliotecas tecnológicas (*Lib*), blocos de propriedade intelectual (*IP*) e a *SPICE Reference Netlist*.

Figura 7 – Fluxo de verificação de equivalência lógica.



Fonte: adaptado de (PATEL; OZA; PARMAR, 2019).

Neste trabalho, essa etapa foi realizada com a ferramenta *Cadence Conformal Logic Equivalence Checker*, em um script gerado automaticamente pelo fluxo de síntese, utilizando informações provenientes do ambiente do Genus e do Conformal. O procedimento foi aplicado ao módulo de topo `core`, que representa o núcleo principal do processador analisado. (Cadence Design Systems, 2026).

No processo de verificação, a descrição RTL foi utilizada como modelo *golden*, enquanto a implementação sintetizada foi utilizada como modelo *revised*. O arquivo mostra que o modelo RTL é composto por módulos em SystemVerilog, incluindo módulos como `alu.sv`, `controller.sv`, `decoder.sv`, `register_file.sv`, `csr.sv` e os estágios do pipeline, todos reunidos e elaborados com raiz em `core`. Já a implementação sintetizada é lida a partir do arquivo compactado `fv/core/fv_map.v.gz`, correspondente à *netlist* gerada após a síntese lógica. Dessa forma, a comparação é estabelecida entre a especificação comportamental do projeto e sua representação estrutural já mapeada para a tecnologia de fabricação. (Cadence Design Systems, 2026)

O script também mostra que a verificação foi realizada considerando a biblioteca da tecnologia X-FAB XH018, por meio do arquivo de biblioteca `D_CELLS_HD_LPMOS_slow_1_62V_125C.lib`. Além disso, foram aplicadas configurações específicas para melhorar a robustez da comparação, como regras de nomenclatura para registradores e instâncias, tratamento de sinais não dirigidos no modelo *golden*, uso de modelagem balanceada e ativação de análise de *datapath*. Tais ajustes são importantes porque permitem que a ferramenta reconheça corretamente equivalências mesmo quando a síntese altera significativamente a estrutura interna do circuito, por exemplo, por compartilhamento de operadores ou reestruturação lógica. (Cadence Design Systems, 2026)

Após a preparação do ambiente, o Conformal executa a comparação hierárquica dinâmica entre os modelos e gera relatórios de verificação, pontos não mapeados e diagnóstico de não equivalência. O script ainda inclui comandos para reportar a quantidade total de pontos de comparação, bem como o número de pontos não equivalentes, abortados e desconhecidos. Assim, essa etapa constitui uma evidência essencial de que as transformações realizadas durante a síntese não alteraram a funcionalidade esperada do RISC-X conforme ilustrado na 8, não foram detectados pontos de não equivalência lógica durante o processo de checagem.

Figura 8 – Conformal verificação de equivalência lógica

```
================================================================================
Compared points      PO     DFF       Total
--------------------------------------------------------------------------------
Equivalent           101    1609      1710
================================================================================

SETUP> dofile ../sim/fv/core/rtl_to_fv_map.do
TCL_LEC> vpx add compared points -all
TCL_LEC> vpx compare
TCL_LEC>
```

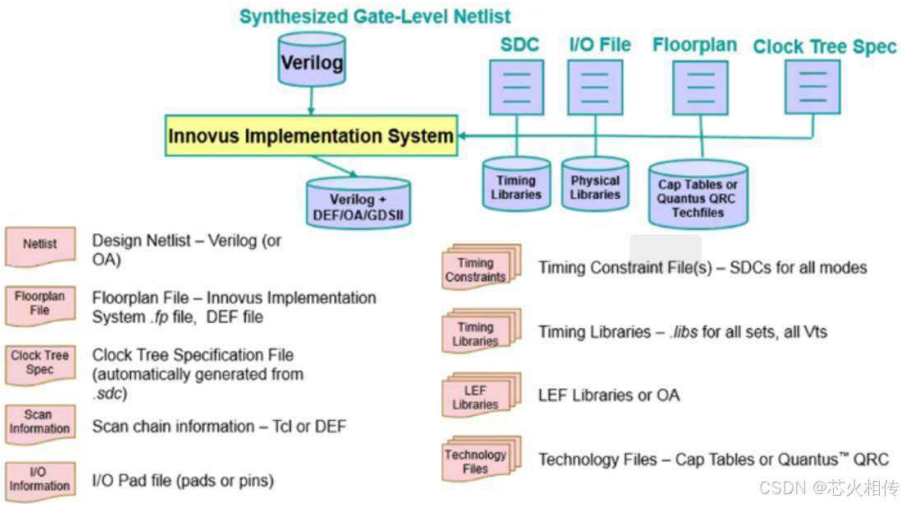
Fonte: Adaptado de (Cadence Design Systems, 2026).

6 Fluxo de Physical Design

Após a etapa de síntese lógica, inicia-se o processo de *Physical Design*, no qual a descrição lógica do circuito é convertida em uma representação física adequada para fabricação. Nessa etapa, a *netlist* gerada pela síntese é utilizada como ponto de partida para a criação do layout do circuito, envolvendo atividades como *floorplanning*, posicionamento das células (*placement*), síntese da árvore de clock (*clock tree synthesis*) e roteamento das interconexões.

Conforme ilustrado na Figura 9, o fluxo de implementação utiliza diferentes arquivos de entrada. O principal é a *netlist* do projeto, normalmente descrita em *Verilog*. Também são utilizados arquivos de restrições temporais no formato *SDC*, que definem parâmetros como frequência de operação e definição de clocks. As bibliotecas de temporização (*.lib*) fornecem os modelos de atraso e potência das células, enquanto as bibliotecas físicas (*LEF*) descrevem as características geométricas e regras de conexão da tecnologia. Outros arquivos, como *floorplan*, especificações da árvore de clock e arquivos de tecnologia, complementam as informações necessárias para a implementação.

Figura 9 – Arquivos de entrada e saída do Innovus



Fonte: Adaptado de (WEIXIN, 2023).

Ao final do fluxo, é gerada a base de dados física do projeto contendo a posição das células e o roteamento das interconexões. Entre os principais arquivos de saída destacam-se o *DEF* e o *GDSII*, que representam o layout final do circuito integrado. A implementação física foi realizada considerando uma frequência de operação de 100 MHz, em virtude da folga de temporização observada na figura 5, aproximadamente igual a 0.600 ns. Esse valor

indica que, após a etapa de síntese lógica, o projeto apresentava margem temporal suficiente para atender ao período de clock correspondente à frequência especificada, permitindo prosseguir com as etapas do fluxo de implementação física.

Após a síntese lógica e a verificação de equivalência entre a descrição RTL e a *netlist* sintetizada, inicia-se o processo de implementação física do circuito digital. Essa etapa tem como objetivo transformar a descrição lógica do circuito em uma representação geométrica completa, adequada para fabricação em tecnologia CMOS. O fluxo de *Physical Design* envolve diversas etapas sucessivas, incluindo planejamento do chip, distribuição de alimentação, posicionamento das células, síntese da árvore de clock e roteamento das interconexões.

A primeira etapa consiste no *floorplanning*, no qual são definidas as dimensões da área do circuito, a posição dos pinos de entrada e saída e a organização inicial da região onde as células padrão serão posicionadas. A Figura 10 apresenta a estrutura inicial do *floorplan*, destacando a região central destinada ao posicionamento das células lógicas.

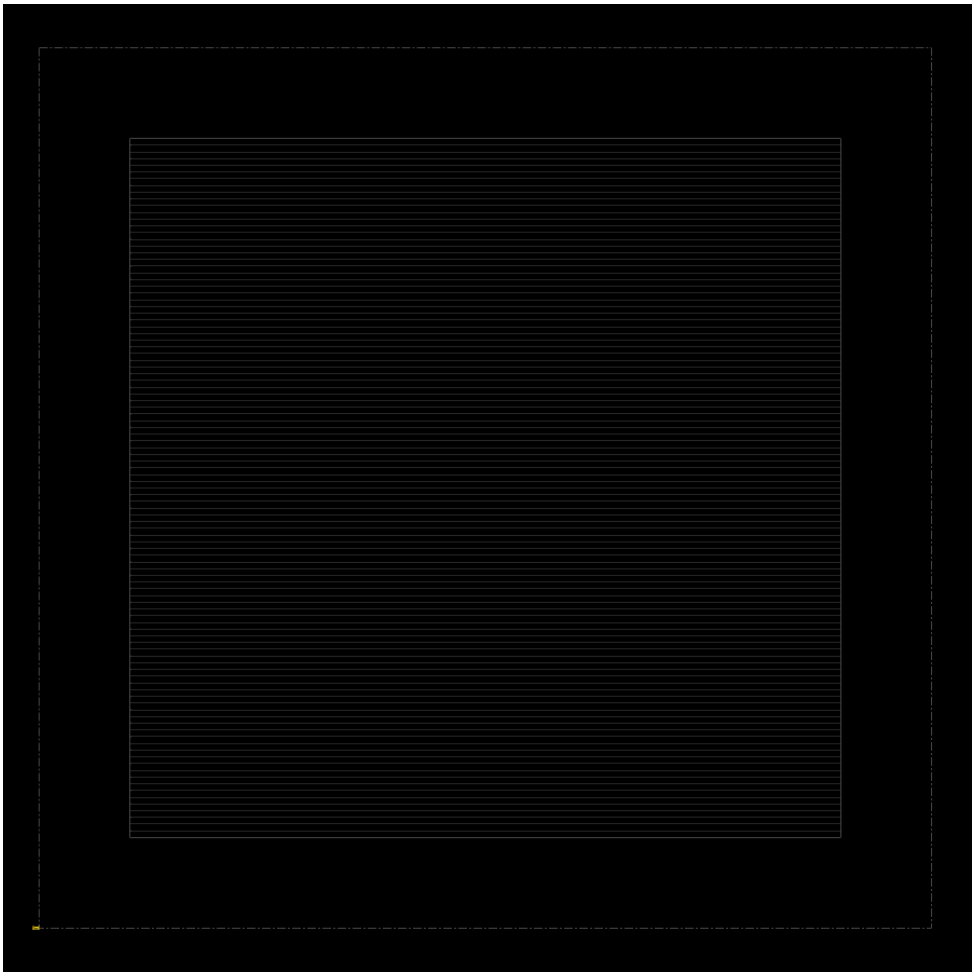


Figura 10 – Estrutura inicial do floorplan do circuito.

Após a definição do *floorplan*, é realizada a construção da rede de distribuição

de alimentação (*power distribution network*). Essa rede é composta por *power rings* e *power stripes*, responsáveis por distribuir uniformemente as tensões de alimentação (VDD) e referência (VSS) ao longo do circuito. A Figura 11 apresenta a estrutura da rede de alimentação implementada no projeto.

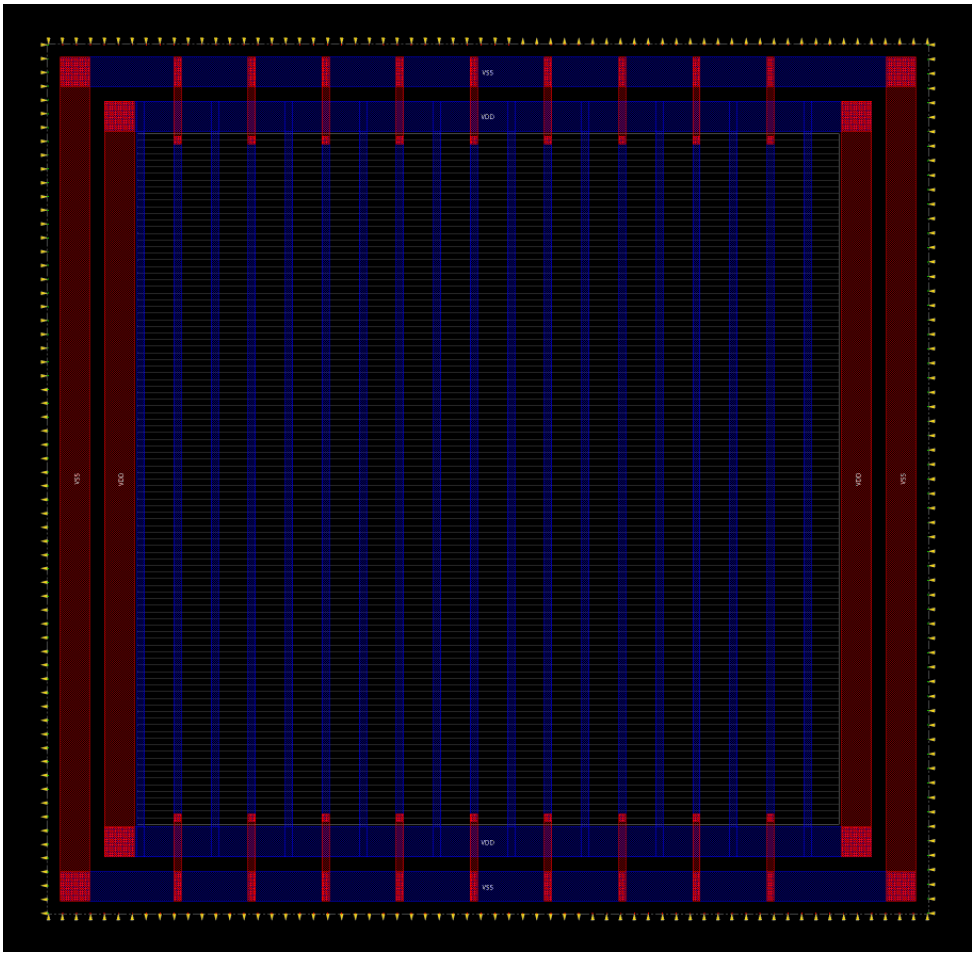


Figura 11 – Rede de distribuição de alimentação com power rings e stripes.

Em seguida, realiza-se o posicionamento das células padrão (*placement*), etapa na qual os elementos lógicos da *netlist* são distribuídos dentro da área definida no *floorplan*. Inicialmente é realizado um posicionamento preliminar, conforme ilustrado na Figura 12, seguido de otimizações para reduzir congestionamentos e melhorar a qualidade da temporização.

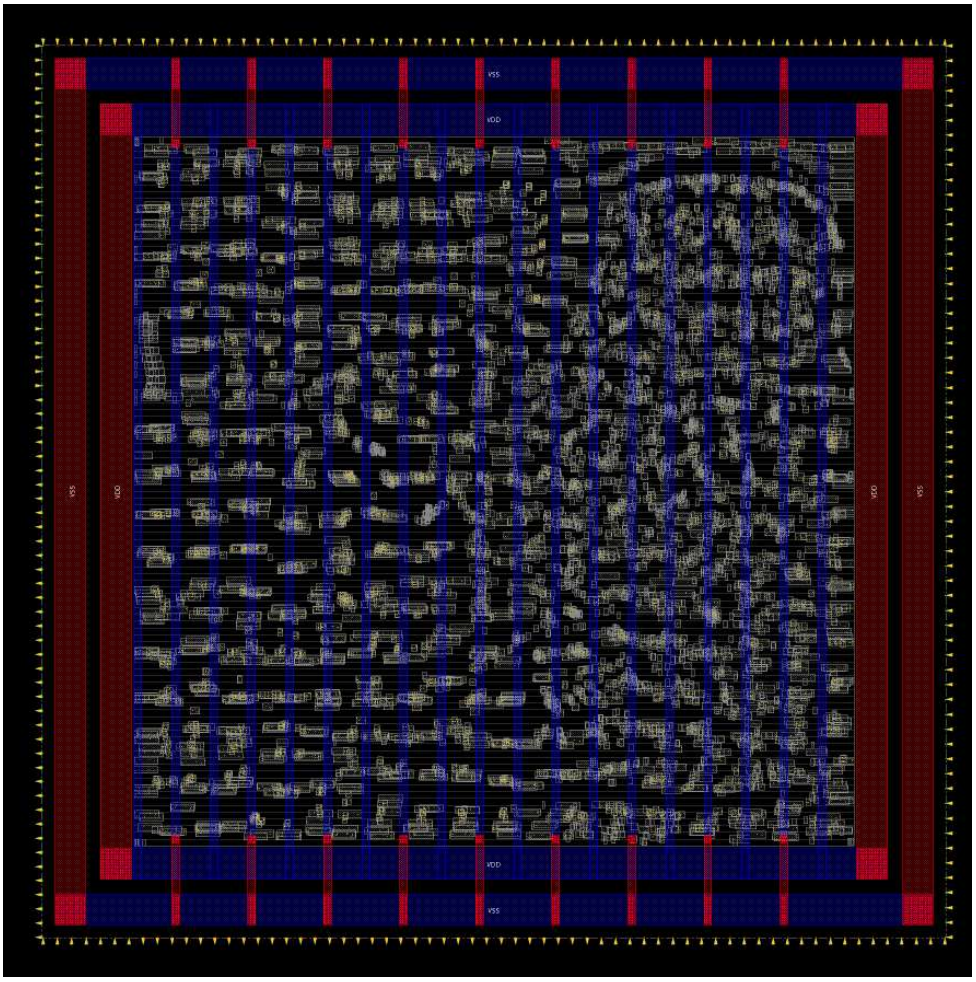


Figura 12 – Posicionamento inicial das células padrão.

Após o posicionamento das células, executa-se a síntese da árvore de clock (*Clock Tree Synthesis – CTS*). O objetivo dessa etapa é distribuir o sinal de clock para todos os elementos sequenciais do circuito mantendo o menor desbalanceamento possível (*clock skew*) e garantindo que os requisitos de temporização sejam atendidos.

A Figura 13 apresenta a estrutura da árvore de clock gerada para o circuito.

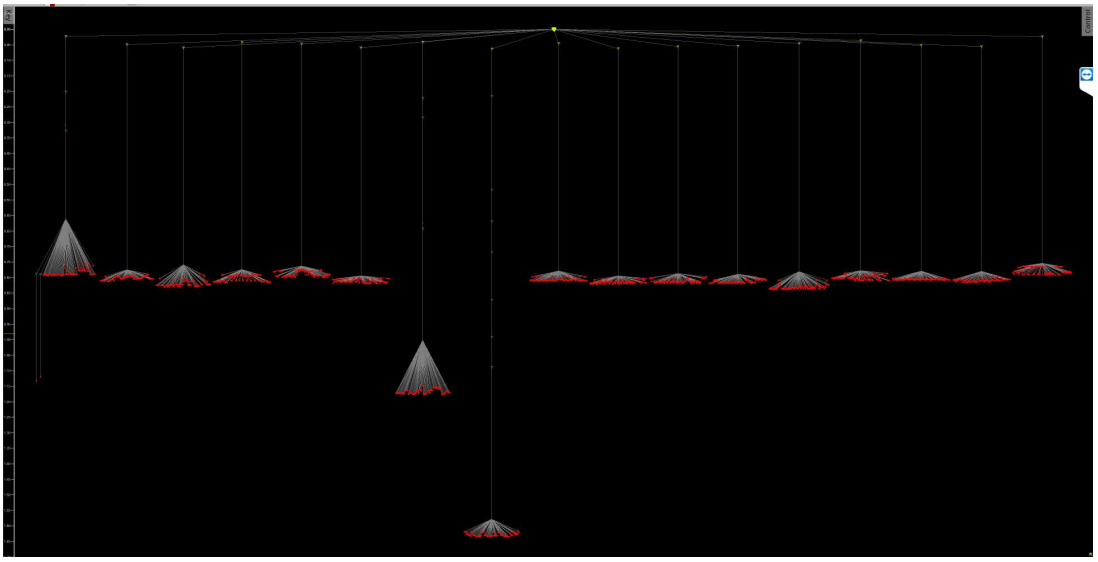


Figura 13 – Estrutura inicial da árvore de clock gerada durante o CTS.

Após as otimizações de CTS, a distribuição do clock torna-se mais equilibrada, conforme ilustrado na Figura 14.

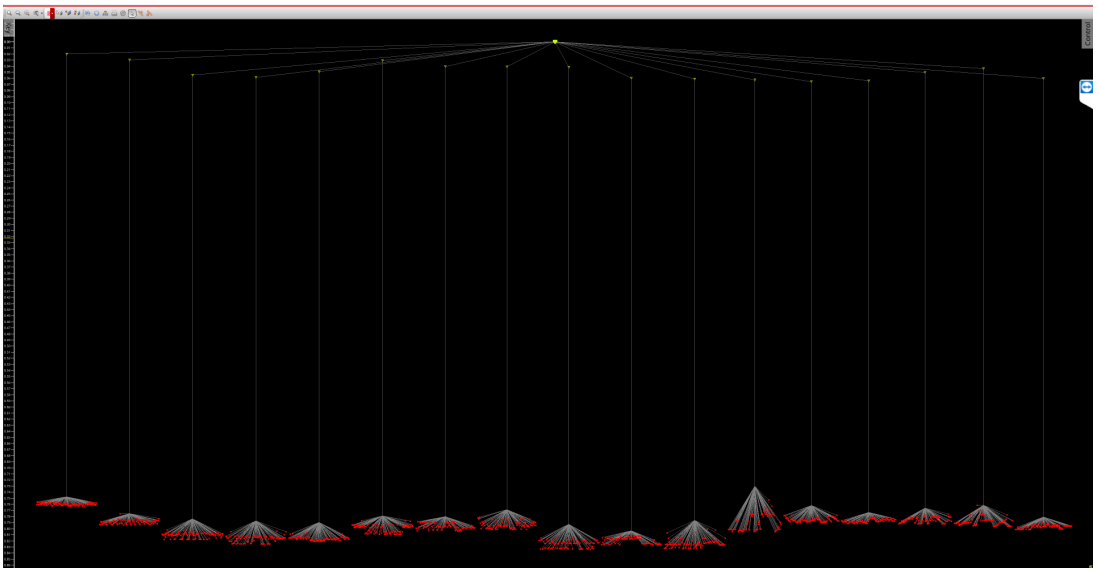


Figura 14 – Árvore de clock após otimizações de CTS.

A etapa seguinte consiste no roteamento (*routing*), responsável por conectar eletricamente todas as células do circuito por meio das camadas metálicas disponíveis na tecnologia. Inicialmente é realizado um roteamento preliminar, mostrado na Figura 15.

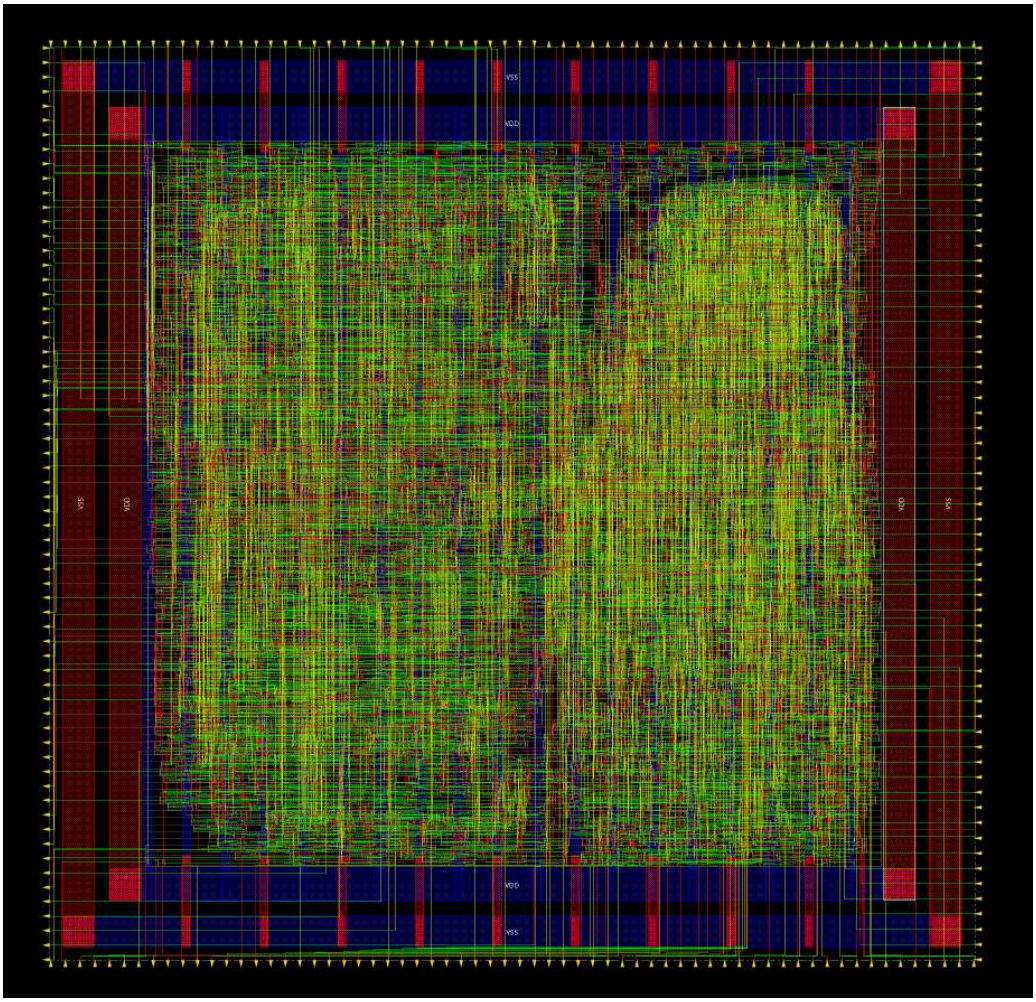


Figura 15 – Roteamento preliminar do circuito.

Após a conclusão do roteamento detalhado, obtém-se o layout final do circuito integrado, conforme apresentado na Figura 16. Esse layout representa a implementação física completa do projeto.

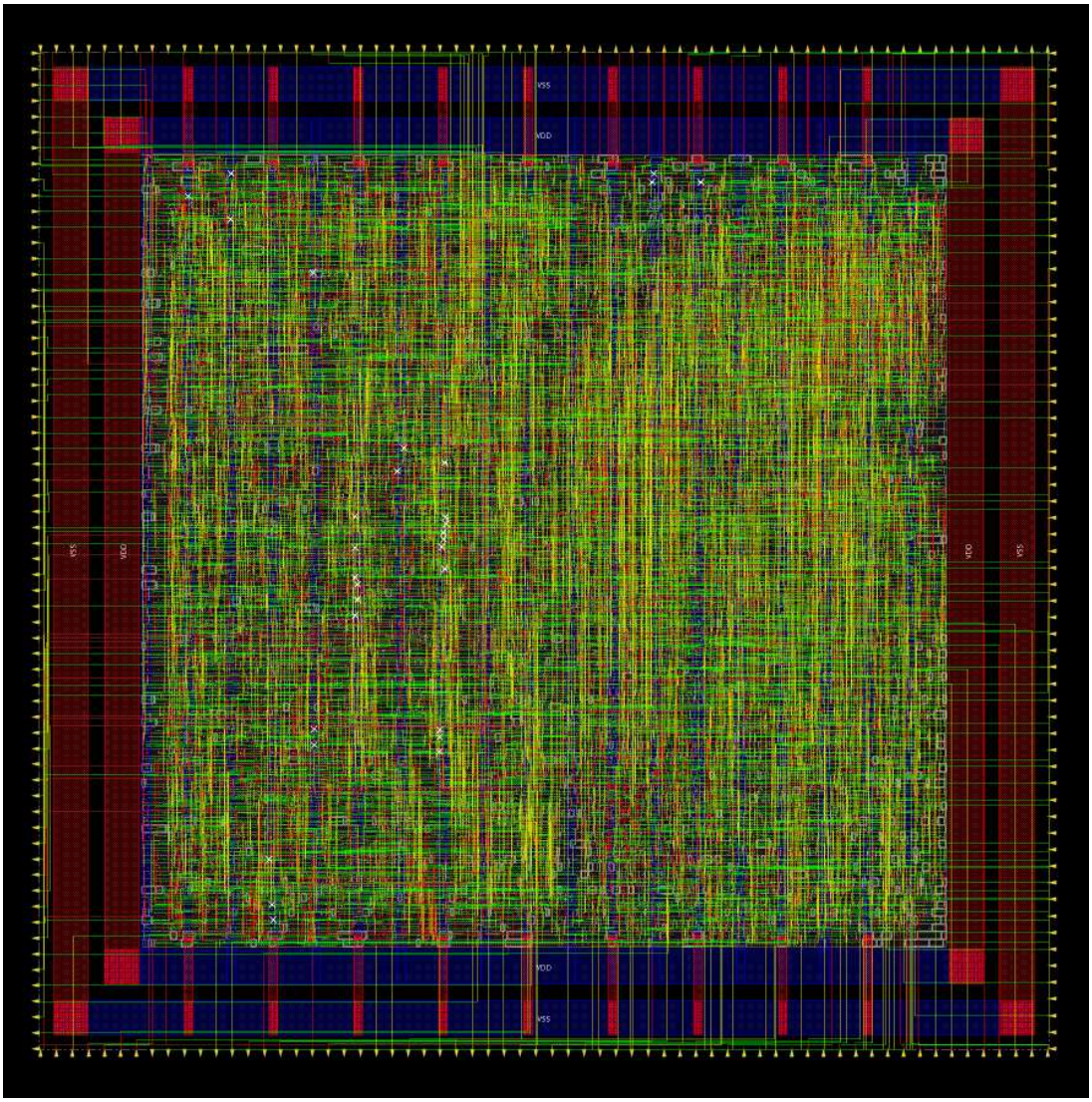


Figura 16 – Layout do circuito após o roteamento completo.

Durante o processo de implementação física também são realizadas verificações automáticas de regras de projeto (*Design Rule Check – DRC*). Essas verificações garantem que o layout esteja em conformidade com as regras de fabricação da tecnologia utilizada.

A Figura 17 apresenta a interface de análise das violações encontradas durante o processo de verificação.

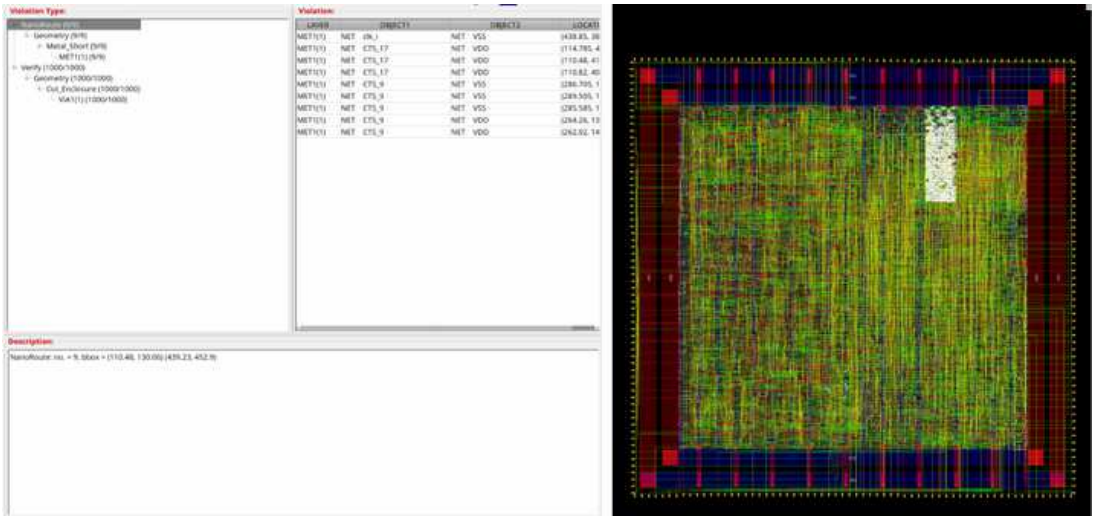


Figura 17 – Interface de análise de violações de regras de projeto (DRC).

7 Resultados

Após a execução completa do fluxo de implementação física utilizando a ferramenta Cadence Innovus, foram gerados diversos relatórios contendo métricas de área, potência e temporização do circuito. Esses resultados permitem avaliar a qualidade da implementação física e identificar possíveis oportunidades de otimização no projeto.

7.1 Área do Circuito

O relatório de área apresenta a quantidade total de instâncias e a área ocupada pelo circuito após o processo de implementação física. Conforme apresentado na Tabela 3, o design denominado *core* possui aproximadamente 7992 instâncias de células padrão, ocupando uma área total de aproximadamente 208671.95 μm^2.

Tabela 3 – Resumo de área do circuito após implementação física

Módulo	**Número de Instâncias**	**Área Total (μm^2)**
RISC-X	7992	208671.95

Esses valores indicam que o circuito apresenta uma densidade moderada de células padrão, sendo compatível com implementações típicas de núcleos de processamento em tecnologias CMOS.

7.2 Análise de Potência

A análise de potência permite estimar o consumo energético do circuito considerando a atividade de comutação dos sinais e os parâmetros elétricos da biblioteca tecnológica. A Tabela 4 apresenta a decomposição do consumo de potência total do circuito.

Tabela 4 – Composição do consumo de potência do circuito

Tipo de Potência	**Valor (mW)**	**Percentual (%)**
Potência Interna	13.885	64.85
Potência de Comutação	7.523	35.14
Potência de Fuga	0.0028	0.01
Potência Total	**21.411**	**100**

Observa-se que a maior parcela do consumo de potência está associada à potência interna das células, seguida pela potência de comutação, enquanto a potência de fuga representa uma fração muito pequena do consumo total.

A Tabela 5 apresenta a distribuição do consumo de potência entre diferentes grupos de células do circuito.

Tabela 5 – Distribuição de potência por tipo de bloco

Grupo	**Interna (mW)**	**Switching (mW)**	**Total (mW)**	**(%)**
Sequencial	11.18	1.109	12.29	57.38
Combinacional	2.399	3.842	6.242	29.15
Rede de Clock	0.311	2.572	2.883	13.47
Total	13.89	7.523	21.41	100

Nota-se que os elementos sequenciais são responsáveis pela maior parcela do consumo energético do circuito, seguidos pela lógica combinacional e pela rede de distribuição de clock.

7.3 Análise de Temporização

A análise de temporização estática (STA) permite identificar os caminhos críticos do circuito e verificar se os requisitos de temporização são atendidos para a frequência de operação especificada. A Tabela 6 apresenta um resumo do pior caminho encontrado durante a análise.

Tabela 6 – Resumo do pior caminho crítico identificado

Parâmetro	**Valor**
Clock	clk_100MHz
Período de Clock	10 ns
Startpoint	ex_stage_inst_alu_operand_2_ex_reg[0]
Endpoint	ex_stage_inst_alu_operand_1_ex_reg[28]
Atraso do Caminho de Dados	11.045 ns
Tempo Requerido	9.760 ns
Slack	**-1.282 ns**

O valor negativo de *slack* indica que o caminho crítico não atende completamente às restrições de temporização estabelecidas para a frequência de operação de 100 MHz. Esse comportamento sugere a necessidade de otimizações adicionais, como redimensionamento de células, inserção de buffers ou melhorias no balanceamento da árvore de clock.

Apesar da presença das violações de temporização, pode se obter uma visão geral do fluxo de síntese e implementação física, permitindo a análise das principais métricas de desempenho do sistema.

8 Conclusão

Este trabalho apresentou a aplicação prática do fluxo de implementação física de circuitos digitais utilizando como estudo de caso um núcleo de processamento baseado na arquitetura RISC-V. O desenvolvimento foi realizado no contexto das atividades do Laboratório XMEN, permitindo a exploração de diversas etapas fundamentais do projeto de circuitos integrados.

Inicialmente foram estudados os conceitos relacionados à síntese lógica e à verificação de equivalência entre a descrição RTL e a netlist sintetizada. Em seguida, foi executado o fluxo completo de *Physical Design*, incluindo as etapas de floorplanning, distribuição de alimentação, posicionamento das células, síntese da árvore de clock e roteamento das interconexões.

Os resultados obtidos demonstraram que o circuito apresenta consumo de potência moderado e uma utilização eficiente da área de silício. A análise de temporização revelou a existência de um caminho crítico que ainda necessita de otimizações adicionais para garantir o atendimento completo das restrições de temporização.

Apesar dessas limitações, o trabalho permitiu compreender de forma prática os desafios envolvidos no processo de síntese e implementação física de sistemas digitais complexos. Além disso, a criação de *scripts* e fluxos automatizados contribui para futuras reproduções do projeto e para o aprimoramento do processo de desenvolvimento.

Como trabalhos futuros, pretende-se realizar otimizações adicionais de temporização e análises de potência considerando perfis de trabalho distintos, bem como a execução das etapas finais de *signoff*, incluindo verificações completas de DRC, LVS e análise de integridade de energia. Essas atividades são essenciais para garantir que o circuito esteja totalmente preparado para fabricação em tecnologia CMOS.

Referências

BROWN, P. P. C. *FPGA Prototyping by Verilog Examples*. [S.l.]: Wiley, 2013.

Cadence Design Systems. *Script de verificação de equivalência lógica RTL versus netlist do módulo core*. 2026. Arquivo gerado automaticamente pelo fluxo Genus/Conformal em 08 mar. 2026.

Google. *RISCV-DV: Random Instruction Generator for RISC-V Processors*. 2023. <https://github.com/google/riscv-dv>. Acessado em: 9 mar. 2026.

HARRIS, S. L.; HARRIS, D. *Digital Design and Computer Architecture: RISC-V Edition*. Burlington, MA: Morgan Kaufmann, 2021. ISBN 978-0128200643.

KAHNG, A. B. et al. *VLSI Physical Design: From Graph Partitioning to Timing Closure*. [S.l.]: Springer, 2011.

OpenHW Group. *CV32E40P: 32-bit RISC-V Core*. 2022. <https://github.com/openhwgroup/cv32e40p>. Acessado em: 9 mar. 2026.

PATEL, P.; OZA, P.; PARMAR, R. *A Guide on Logical Equivalence Checking - Flow, Challenges, and Benefits*. 2019. Acesso em: 23 mar. 2026. Disponível em: <https://www.design-reuse.com/article/61105-a-guide-on-logical-equivalence-checking-flow-challenges-and-benefits/>.

ROTH, C. H.; JOHN, L. K. *Digital Systems Design Using VHDL*. [S.l.]: Cengage Learning, 2004.

WEIXIN. *CSDN* . 2023. Acesso em: 23 mar. 2026. Disponível em: <https://blog.csdn.net/weixin_37584728/article/details/131977067>.

WOLF, C.; GLASER, J. *The RISC-V Formal Verification Framework*. 2017. <https://github.com/YosysHQ/riscv-formal>. Framework de verificação formal para processadores RISC-V.

WOLF, W. *Modern VLSI Design: IP-Based Design*. [S.l.]: Prentice Hall, 2008.