



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE CIÊNCIAS EXATAS E NATURAIS
CURSO BACHARELADO INTERDISCIPLINAR EM CIÊNCIA E TECNOLOGIA

PABLO GERALDO LINHARES DO NASCIMENTO LEITE

INVESTIGAÇÃO DA IMPLEMENTAÇÃO DO SAP-1 EM FPGA BASYS 3
UTILIZANDO LOGISIM EVOLUTION COMO FERRAMENTA DIDÁTICA

PAU DOS FERROS

2024

PABLO GERALDO LINHARES DO NASCIMENTO LEITE

INVESTIGAÇÃO DA IMPLEMENTAÇÃO DO SAP-1 EM FPGA BASYS 3
UTILIZANDO LOGISIM EVOLUTION COMO FERRAMENTA DIDÁTICA

Monografia apresentada a Universidade Federal Rural do Semi-Árido *Campus* Pau dos Ferros como requisito parcial para obtenção do título de Bacharel em Interdisciplinar em Ciência e Tecnologia.

Orientador: Prof. Dr. Pedro Thiago Valério de Souza

PAU DOS FERROS

2024

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

L533i Leite, Pablo Geraldo Linhares do Nascimento.
Investigação da implementação do SAP-1 em FPGA
basys 3 utilizando logisim evolution como
ferramenta didática / Pablo Geraldo Linhares do
Nascimento Leite. - 2024.
38 f. : il.

Orientador: Pedro Thiago Valério de Souza.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Ciência e
Tecnologia, 2024.

1. SAP-1. 2. FPGA. 3. Logisim. 4. Arquitetura
e organização de computadores. 5. Circuitos
digitais. I. Souza, Pedro Thiago Valério de,
orient. II. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade
com AACR2 e os dados fornecidos pelo(a) autor(a).

Biblioteca Campus Pau dos Ferros
Bibliotecária: Erlanda Maria Lopes da Silva
CRB: 15/966

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

PABLO GERALDO LINHARES DO NASCIMENTO LEITE

INVESTIGAÇÃO DA IMPLEMENTAÇÃO DO SAP-1 EM FPGA BASYS 3
UTILIZANDO LOGISIM EVOLUTION COMO FERRAMENTA DIDÁTICA

Monografia apresentada a Universidade Federal Rural do Semi-Árido *Campus* Pau dos Ferros como requisito parcial para obtenção do título de Bacharel em Interdisciplinar em Ciência e Tecnologia.

Defendida em: 28/03/2025.

BANCA EXAMINADORA

Pedro Thiago Valério de Souza, Prof. Dr. (UFERSA)
Presidente

Francisco Carlos Gurgel da Silva Segundo, Prof. Dr. (UFERSA)
Membro Examinador

Cecílio Martins de Souza Neto, Prof. Dr. (UFERSA)
Membro Examinador

RESUMO

Este trabalho apresenta o desenvolvimento e a implementação do computador educacional SAP-1 utilizando a ferramenta *Logisim Evolution* para projeto e simulação, seguida da síntese em FPGA, com o objetivo de criar uma ferramenta didática moderna para o ensino de arquitetura de computadores. A metodologia incluiu a modelagem dos componentes do SAP-1 no *Logisim*, a simulação bem-sucedida das operações básicas, e a adaptação do projeto para síntese em FPGA. Os resultados demonstraram o pleno funcionamento lógico na simulação, porém a implementação física na FPGA Basys 3 não produziu os resultados esperados, sugerindo possíveis restrições do *Logisim* na geração de hardware complexo. Apesar dos desafios técnicos encontrados, o trabalho alcançou seu propósito educacional ao validar os conceitos teóricos e oferecer uma plataforma prática para o estudo de arquitetura computacional, enquanto aponta para a necessidade de futuras investigações sobre ferramentas alternativas de síntese.

Palavras-chave: SAP-1; logisim; FPGA; arquitetura e organização de computadores; circuitos digitais.

ABSTRACT

This paper presents the development and implementation of the SAP-1 educational computer using the Logisim Evolution tool for design and simulation, followed by synthesis in FPGA, with the aim of creating a modern didactic tool for teaching computer architecture. The methodology included the modeling of the SAP-1 components in Logisim, the well-completed simulation of the basic operations, and the adaptation of the design to FPGA specifications. The results demonstrated full logical functioning in the simulation, but the physical improvement in the Basys 3 FPGA did not produce the expected results, suggesting possible restrictions of Logisim in the generation of complex hardware. Despite the technical challenges encountered, the work achieved its educational objective by validating the theoretical concepts and offering a practical platform for the study of computer architecture, while pointing to the need for future investigations on alternative health tools.

Keywords: SAP-1; logisim; FPGA; computer architecture and organization; digital circuits.

LISTA DE FIGURAS

Figura 1 – Arquitetura do SAP-1	11
Figura 2 – Placa de Desenvolvimento FPGA Basys 3	18
Figura 3 – SAP-1 sem Controlador	19
Figura 4 – SAP-1 com Controlador	20
Figura 5 – RAM implementada com MUX	21
Figura 6 – Simulação do SAP-1 sem Controlador	22
Figura 7 – Simulação do SAP-1 com Controlador	23
Figura 8 – Teste com Circuito Simples	24
Figura 9 – MUX para o Barramento W	25
Figura 10 – Simulação do SAP-1 com MUX no barramento.....	26
Figura 11 – Contador de Programa	29
Figura 12 – Meio Somador	29
Figura 13 – Registrador de Endereço de Memória	30
Figura 14 – Memória RAM	31
Figura 15 – Registrador de Instrução	32
Figura 16 – Controlador/Sequenciador	33
Figura 17 – Contador em Anel	33
Figura 18 – Decodificador de Instrução	34
Figura 19 – Acumulador	35
Figura 20 – Somador-Subtrator	36
Figura 21 – Somador Completo	36
Figura 22 – Registrador B	37
Figura 23 – Registrador de Saída	38

LISTA DE TABELAS

Tabela 1 – Conjunto de Instruções do SAP-1	14
Tabela 2 – Programação do SAP-1	14
Tabela 3 – Resumo do ciclo de busca e execução do SAP-1	17
Tabela 4 – Instruções do Programa	21

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Contexto do Estudo	8
1.2	Descrição do Problema	8
1.3	Objetivos do Projeto	9
1.3.1	Objetivo Geral	9
1.3.2	Objetivos Específicos	9
2	FUNDAMENTAÇÃO TEÓRICA	11
2.1	Arquitetura do SAP-1	11
2.2	Conjunto de Instruções do SAP-1	13
2.3	Programação do SAP-1	14
2.4	Ciclo de Máquina e Ciclo de Instrução do SAP-1	15
2.4.1	Ciclo de Busca	15
2.4.2	Ciclo de Execução	15
2.4.3	Ciclo de Instrução	17
3	METODOLOGIA	18
3.1	Equipamentos e Ferramentas Utilizadas	18
3.2	Projeto de Circuito	18
3.3	Simulação do SAP-1	20
3.4	Síntese em FPGA	21
4	RESULTADOS E DISCUSSÕES	22
4.1	Resultados da Simulação	22
4.2	Resultados da Síntese em FPGA	23
5	CONCLUSÃO	27
	REFERÊNCIAS	28
	APÊNDICE A	29

1 INTRODUÇÃO

1.1 Contexto do Estudo

Albert Paul Malvino (1983) introduziu o sistema de computador *Simple-As-Possible* (SAP) como um modelo educacional em sua obra "*Digital Computer Electronics*". O SAP foi desenvolvido para apresentar os princípios essenciais do funcionamento de computadores, servindo como base para a compreensão de arquiteturas mais avançadas. O sistema é organizado em três versões, que evoluem em complexidade e funcionalidade progressivamente: SAP-1, SAP-2 e SAP-3. Inicialmente, o SAP era implementado em ambientes acadêmicos utilizando circuitos lógicos *Transistor-Transistor Logic* (TTL), conforme descrito no projeto original de Malvino.

Com o avanço tecnológico, os *Field Programmable Logic Devices* (FPGAs) surgiram como uma solução versátil para a implementação de sistemas digitais. FPGAs são dispositivos reconfiguráveis que possibilitam a criação de circuitos lógicos por meio de descrição de *hardware*. Eles são constituídos por blocos lógicos interconectáveis, que podem ser configurados para executar funções específicas. Essa tecnologia superou, em vários aspectos, os circuitos TTL, oferecendo maior adaptabilidade, redução de custos e rapidez no desenvolvimento de protótipos.

A síntese em FPGAs é tradicionalmente realizada através de Linguagens de Descrição de Hardware (HDLs), como Verilog ou VHDL, que exigem conhecimento avançado de descrição para sistemas digitais. Como alternativa educacional, o *Logisim Evolution* oferece uma abordagem gráfica intuitiva, convertendo automaticamente os projetos de circuitos em HDL equivalente para posterior síntese na FPGA. Essa funcionalidade torna a ferramenta particularmente valiosa para o ensino de arquitetura de computadores, pois permite que os estudantes foquem primeiro na compreensão conceitual dos circuitos através da interface visual, antes de lidar com a complexidade das HDLs.

1.2 Descrição do Problema

Este projeto tem como objetivo desenvolver o sistema SAP-1 através da ferramenta *Logisim Evolution* e realizar a síntese em uma placa FPGA. A implementação desse sistema pode ser empregada como ferramenta didática nas práticas de laboratório das disciplinas de Arquitetura e Organização de Computadores, e Circuitos Digitais.

As práticas de laboratório convencionais exigem que os alunos montem circuitos experimentais utilizando componentes TTL. No entanto, essa metodologia tem desafios significativos quando os circuitos ganham proporções maiores e mais avançadas, como a complexidade de montagem, a dificuldade de depuração e a fragilidade dos circuitos. Com o uso do *Logisim Evolution* e FPGAs, o processo se torna mais eficiente e robusto. O projeto desenvolvido no *Logisim* pode ser facilmente simulado, anulando a necessidade de depuração manual de circuitos TTL. Isso permite que os alunos se dediquem mais ao entendimento dos conceitos de arquitetura de computadores, em vez de se preocuparem com problemas de *hardware*.

Além disso, a implementação do SAP-1 com circuitos TTL, conforme proposta por Malvino, requer a utilização de mais de 50 *chips* TTL e uma grande quantidade de conexões com cabos, o que aumenta a probabilidade de erros e falhas. Em contraste, a implementação em PFGA oferece maior robustez, integração e facilidade de depuração, uma vez que todo o sistema é sintetizado em uma única placa programável.

1.3 Objetivos do Projeto

1.3.1 Objetivo Geral

Desenvolver uma implementação didática do computador *Simple-As-Possible* (SAP-1) em FPGA, utilizando a ferramenta *Logisim Evolution* para projeto e simulação, com o propósito de enriquecer o ensino prático nas disciplinas de Arquitetura e Organização de Computadores, e Circuitos Digitais.

1.3.2 Objetivos Específicos

- Implementar o SAP-1 com sua arquitetura simplificada, destacando apenas os componentes essenciais, para que os alunos possam assimilar os conceitos básicos de arquitetura de computadores eliminando detalhes que poderiam dificultar o aprendizado.
- Utilizar o *Logisim Evolution* para permitir que os alunos visualizem e simulem o fluxo de dados e controle do SAP-1, além de realizar a síntese em FPGA, incentivando a aprendizagem ativa por meio da manipulação direta do sistema.

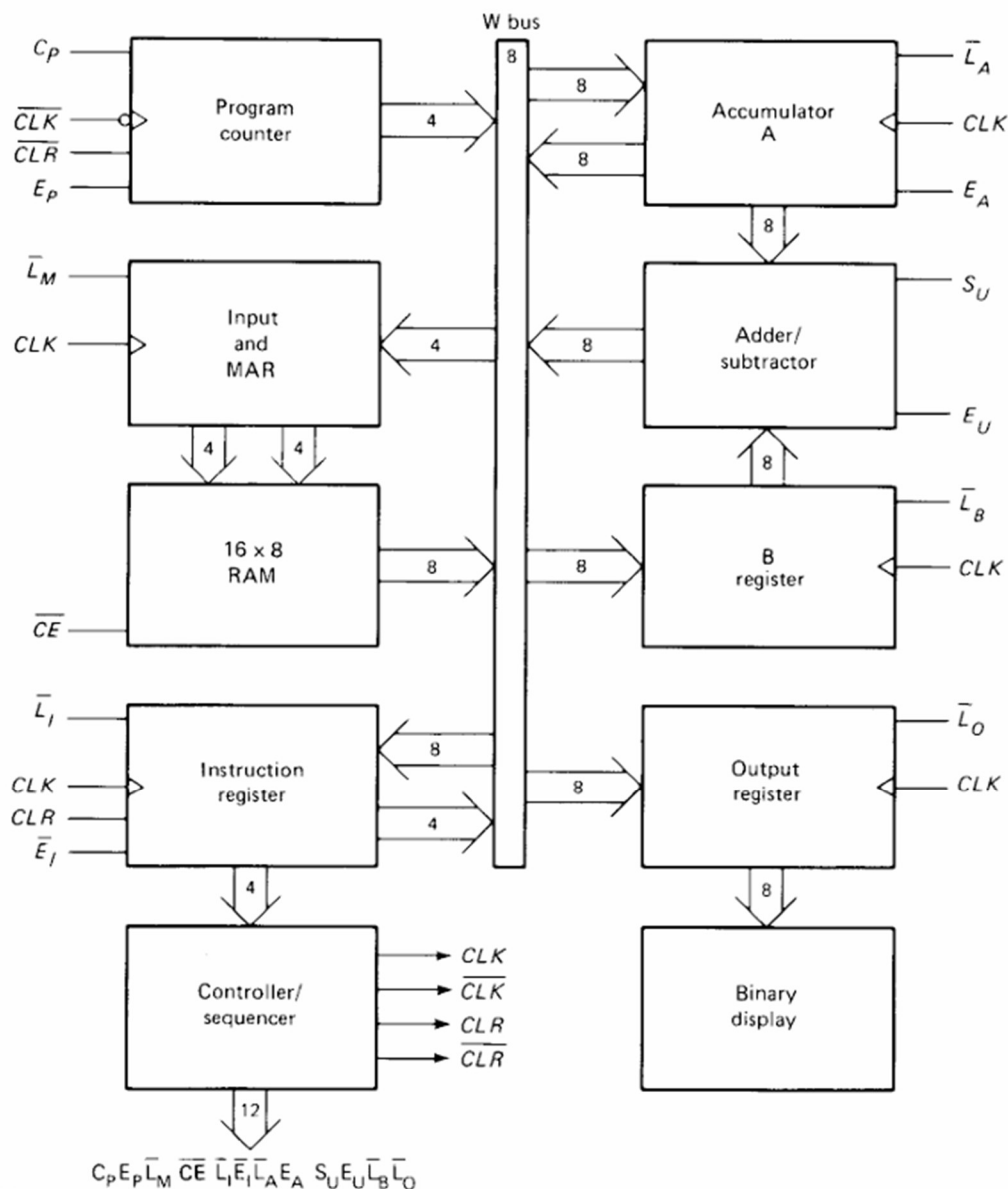
- Substituir a montagem tradicional com circuitos TTL por uma implementação em FPGA, reduzindo o tempo gasto com depuração de *hardware* e permitindo que os alunos foquem no entendimento arquitetural.
- Capacitar os alunos no uso de ferramentas como *Logisim Evolution* e fluxos de síntese em FPGA, preparando-os para desafios mais avançados em projetos de sistemas digitais.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Arquitetura do SAP-1

O SAP-1 é um computador de barramento compartilhado de 8 *bits*, e tem dez componentes principais, descritos nessa subseção. A Figura 1 ilustra sua arquitetura. Todas as entradas para o barramento W são de três estados, para possibilitar transferência de dados de forma ordenada.

Figura 1 – Arquitetura do SAP-1



Fonte: Malvino (1983).

Conforme a Figura 1, o SAP-1 é composto por dez componentes principais, que trabalham em conjunto para executar as operações básicas de um computador. Cada componente desempenha uma função específica no fluxo de dados e controle do sistema. A seguir, são descritos os componentes que formam a arquitetura do SAP-1.

1. Contador de Programa (PC - *Program Counter*): O Contador de Programa é um registrador de 4 *bits* responsável por armazenar o endereço da próxima instrução a ser executada. Durante a execução de um programa, o PC é incrementado automaticamente após cada instrução, garantindo que o computador acesse as instruções na sequência correta. O PC é essencial para o controle do fluxo de execução do programa.
2. Registrador de Endereço de Memória (MAR - *Memory Address Register*): O Registrador de Endereço de Memória é um registrador de 4 *bits* que armazena o endereço da memória RAM a ser acessado. O MAR recebe o valor do barramento de endereços e o utiliza para selecionar a localização correta na memória durante operações de leitura ou escrita.
3. Memória RAM (*Random Access Memory*): A Memória RAM do SAP-1 é uma memória de 16 endereços, cada um com 8 *bits* de largura. Ela armazena tanto as instruções do programa quanto os dados utilizados durante a execução. A RAM é acessada pelo MAR, que indica qual endereço deve ser lido ou escrito. Durante a leitura, o conteúdo do endereço selecionado é enviado ao barramento de dados.
4. Registrador de Instrução (IR - *Instruction Register*): O Registrador de Instrução é um registrador de 8 *bits* que armazena a instrução atual sendo executada. Ele recebe o valor da memória RAM e o divide em duas partes: o *opcode* (código da operação) e o operando (dado ou endereço). O *opcode* é enviado ao Controlador-Sequenciador, enquanto o operando pode ser utilizado como dado ou endereço, dependendo da instrução.
5. Controlador-Sequenciador (*Control Unit*): O Controlador-Sequenciador é o componente responsável por gerar os sinais de controle que coordenam as operações de todos os outros componentes. Ele interpreta o *opcode* recebido do IR e ativa os sinais

de controle necessários para executar a instrução. A palavra de controle tem o formato $CON = CpEpLm'Ce' Li'Ei'La'Ea SuEuLb'Lo'$.

6. Acumulador (ACC - *Accumulator*): O Acumulador é um registrador de 8 *bits* que armazena temporariamente os dados utilizados nas operações aritméticas e lógicas. Ele é o principal registrador de trabalho do SAP-1, recebendo dados da memória RAM ou da ULA (Unidade Lógica e Aritmética) e fornecendo dados para outras operações. O Acumulador também pode enviar seu conteúdo para o barramento de dados.
7. Somador-Subtrator (ULA - Unidade Lógica e Aritmética): O Somador-Subtrator é a unidade responsável por realizar operações aritméticas básicas, como soma e subtração. Ele opera com os valores armazenados no Acumulador e no Registrador-B, enviando o resultado de volta ao Acumulador. A ULA é controlada pelo Controlador-Sequenciador, que define qual operação será executada.
8. Registrador B (*B Register*): O Registrador-B é um registrador de 8 *bits* que armazena temporariamente um dos operandos utilizados nas operações da ULA. Ele recebe dados da memória RAM através do barramento de dados e os fornece para o Somador-Subtrator quando necessário.
9. Registrador de Saída (*OUT Register*): O Registrador de Saída é um registrador de 8 *bits* que armazena o resultado final de uma operação para exibição. Ele recebe dados do Acumulador e os envia para o *Display* Binário, permitindo que o usuário visualize o resultado das operações realizadas pelo SAP-1.
10. Display Binário (*Binary Display*): O *Display* Binário é o componente de saída do SAP-1, responsável por exibir o conteúdo do Registrador de Saída em formato binário. Ele permite que o usuário visualize os resultados das operações de forma clara e direta, facilitando a compreensão do funcionamento do computador.

2.2 Conjunto de Instruções do SAP-1

O conjunto de Instruções de um computador são as operações básicas que podem ser executadas. O SAP-1 tem um conjunto de apenas cinco instruções, descritas na Tabela 1.

Tabela 1 – Conjunto de Instruções do SAP-1

Mnemônico	Op code	Operação
LDA	0000	Carregar dados da RAM no Acumulador
ADD	0001	Carregar dados da RAM no Registrador B e somar com os dados do Acumulador
SUB	0010	Carregar dados da RAM no Registrador B e subtrair dos dados do Acumulador
OUT	1110	Carregar dados do Acumulador no Registrador de Saída
HLT	1111	Parar processamento

Fonte: Adaptado de Malvino (1983).

2.3 Programação do SAP-1

A programação do SAP-1 é armazenada na memória RAM do sistema. É utilizado um formato de instrução de 8 *bits*, onde os 4 *bits* superiores representam o *opcode* (código da operação) e os 4 *bits* inferiores representam o operando (endereço de memória).

As instruções são armazenadas nas posições mais baixas da memória RAM (a partir do endereço 0x0), enquanto os dados são armazenados em posições mais altas. Na Tabela 2 é apresentado um exemplo de programação do SAP-1, mostrando como uma instrução é representada tanto em linguagem de montagem (*assembly*) quanto em linguagem de máquina (binário).

Tabela 2 – Programação do SAP-1

Endereço	Instrução/Dado	Instrução/Dado
0x0	LDA 0x9	0000 1001
0x9	0x11	0001 0001

Fonte: Adaptado de Yeek (2007).

Neste exemplo:

- A instrução LDA 0x9 (carregar o acumulador com o conteúdo do endereço 0x9) é armazenada no endereço 0x0 da memória RAM.

- O dado 0x11 (17 em decimal) está armazenado no endereço 0x9 da memória RAM.

Na representação em linguagem de máquina:

- O opcode da operação LDA é 0000 (4 *bits* superiores).
- O operando (endereço 0x9) é 1001 (4 *bits* inferiores).

2.4 Ciclo de Máquina e Ciclo de Instrução do SAP-1

O funcionamento do SAP-1 é baseado em um ciclo contínuo de busca e execução de instruções, conhecido como ciclo de máquina. Esse ciclo é dividido em seis estados de tempo, denominados T1 a T6, que são controlados por um contador em anel acionado pela borda de descida do *clock*.

2.4.1 Ciclo de Busca

O ciclo de busca é composto pelos três primeiros estados (T1, T2 e T3) e é comum a todas as instruções. Durante esse ciclo, o SAP-1 realiza as seguintes operações:

1. Estado T1 (Estado de Endereço):
 - O endereço armazenado no Contador de Programa (PC) é enviado ao Registrador de Endereço de Memória (MAR). Os comandos ativos são Ep e Lm'.
2. Estado T2 (Estado de Incremento):
 - O Contador de Programa (PC) é incrementado, apontando para o endereço da próxima instrução. O único comando ativo é Cp.
3. Estado T3 (Estado de Memória):
 - A instrução armazenada no endereço especificado pelo MAR é lida da memória RAM e armazenada no Registrador de Instrução (IR). Os comandos ativos são Ce' e Li'.

2.4.2 Ciclo de Execução

O ciclo de execução é composto pelos três estados restantes (T4, T5 e T6) e varia de acordo com a instrução sendo executada.

1. Instrução LDA:

- Estado T4: A instrução é decodificada e o endereço do operando é armazenado no MAR. Os comandos ativos são Lm' e Ei' ;
- Estado T5: O conteúdo da RAM no endereço especificado é transferido para o Acumulador. Os comandos ativos são Ce' e La' ;
- Estado T6: Sem operações.

2. Instrução ADD:

- Estado T4: A instrução é decodificada e o endereço do operando é armazenado no MAR. Os comandos ativos são Lm' e Ei' ;
- Estado T5: O conteúdo da RAM no endereço especificado é transferido para o Registrador B. Os comandos ativos são Ce' e Lb' ;
- Estado T6: A ULA realiza a operação de soma com o conteúdo do Acumulador e do Registrador B, e o resultado é armazenado no Acumulador. Os comandos ativos são Eu e La' .

3. Instrução SUB:

- Estado T4: A instrução é decodificada e o endereço do operando é armazenado no MAR. Os comandos ativos são Ei' e Lm' ;
- Estado T5: O conteúdo da RAM no endereço especificado é transferido para o Registrador B. Os comandos ativos são Ce' e Lb' ;
- Estado T6: A ULA realiza a operação de subtração com o conteúdo do Acumulador e do Registrador B, e o resultado é armazenado no Acumulador. Os comandos ativos são La' , Su e Eu .

4. Instrução OUT:

- Estado T4: A instrução é decodificada e o valor do Acumulador é transferido para o Registrador de Saída. Os comandos ativos são Ea e Lo' ;
- Estado T5: Sem operações;
- Estado T6: Sem operações.

5. Instrução HLT:

- Estado T4: A instrução é decodificada e nenhuma operação é realizada, pois o processamento é encerrado.

2.4.3 Ciclo de Instrução

O ciclo de instrução do SAP-1 é definido como o tempo necessário para buscar e executar uma instrução. No SAP-1, o ciclo de instrução é fixo e corresponde ao ciclo de máquina, ou seja, 6 estados de tempo (T1 a T6). Cada estado de tempo é caracterizado por um ciclo de *clock*, começando e terminando na borda de descida do *clock*. Como o SAP-1 é sensível à borda de subida do *clock*, as operações ocorrem no meio de cada estado de tempo, permitindo um tempo adequado para os sinais no barramento W. A Tabela 3 contém o resumo dos ciclos, bem como as palavras de controle necessárias para cada ciclo.

Tabela 3 – Resumo do ciclo de busca e execução do SAP-1

Operação	OpCode	Estado	CON 'h	CON 'b	Ativo
Todas	-	T1	5E3H	0101 1110 0011	Ep, Lm'
		T2	BE3H	1011 1110 0011	Cp
		T3	263H	0010 0110 0011	Ce', Li'
LDA	0000	T4	1A3H	0001 1010 0011	Lm', Ei'
		T5	2C3H	0010 1100 0011	CE', La'
		T6	3E3H	0011 1110 0011	Nenhum
ADD	0001	T4	1A3H	0001 1010 0011	Lm', Ei'
		T5	2E1H	0010 1110 0001	CE', Lb'
		T6	3C7H	0011 1100 0111	La', Eu
SUB	0010	T4	1A3H	0001 1010 0011	Lm', Ei'
		T5	2E1H	0010 1110 0001	CE', Lb'
		T6	3CFH	0011 1100 1111	La', Su, Eu
OUT	1110	T4	3F2H	0011 1111 0010	Ea, Lo'
		T5	3E3H	0011 1110 0011	Nenhum
		T6	3E3H	0011 1110 0011	Nenhum
HLT	1111	-	3E3H	0011 1110 0011	Nenhum

Fonte: Adaptado de Malvino (1983).

3 METODOLOGIA

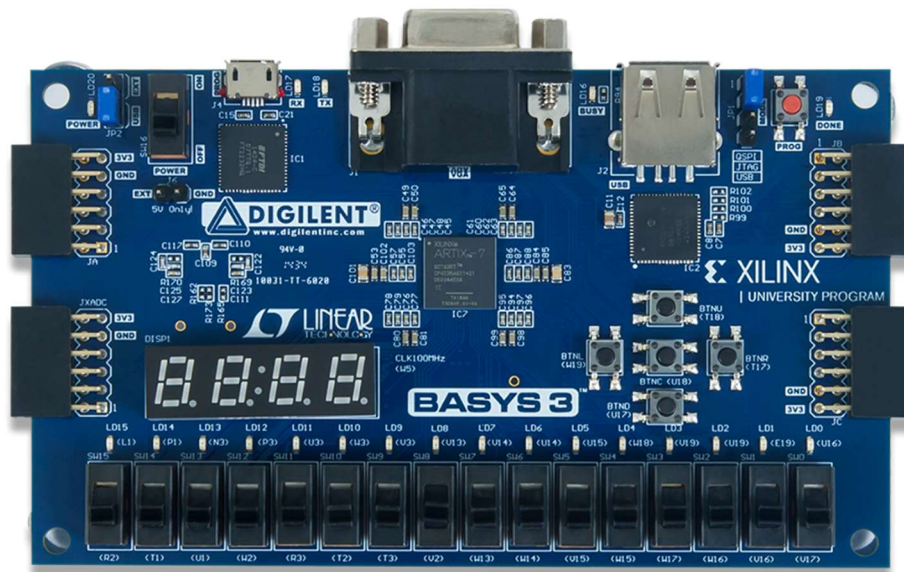
Este capítulo descreve a trajetória do desenvolvimento do projeto do SAP-1, desde o estudo teórico até a implementação prática.

3.1 Equipamentos e Ferramentas Utilizadas

Foram utilizadas as seguintes ferramentas e equipamentos:

- *Logisim Evolution*: Ferramenta de simulação de circuitos lógicos digitais, utilizada para projetar e simular os módulos do SAP-1.
- *FPGA Basys 3*: Placa de desenvolvimento FPGA da Xilinx, apresentada na Figura 2, utilizada para a síntese e teste do sistema.

Figura 2 – Placa de Desenvolvimento: Basys 3 Artix-7 FPGA



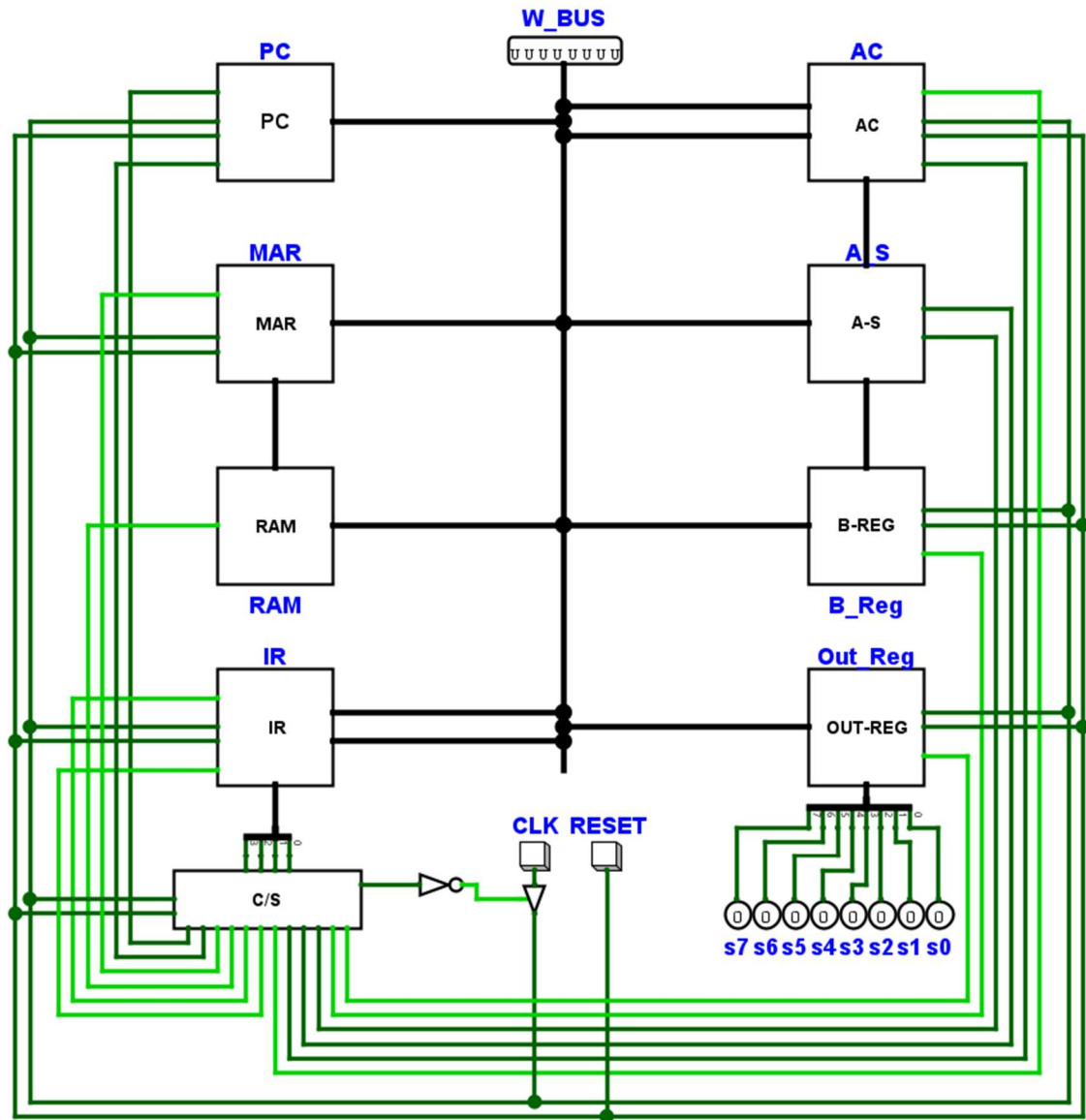
Fonte: Digilent (2024).

- *Vivado Design Suite*: Software da Xilinx para síntese e implementação de projetos em FPGA.

3.2 Projeto de Circuito

O projeto do SAP-1 foi desenvolvido em etapas, começando pela implementação individual de cada componente no *Logisim Evolution*. Após a validação individual dos

Figura 4 – SAP-1 com Controlador



Fonte: Adaptado de Logisim Evolution (2024).

3.3 Simulação do SAP-1

As simulações do sistema sem Controlador e completo foi realizada no *Logisim Evolution*. Um programa de exemplo, contendo as instruções e dados apresentados na Tabela 4 para a operação $7 + 5 - 2$, foi carregado na memória RAM. As simulações permitiram verificar o funcionamento correto de cada módulo e do sistema como um todo, garantindo que todas as operações fossem executadas conforme esperado. Os resultados da simulação são apresentados no capítulo 4.

Tabela 4 – Instruções do Programa

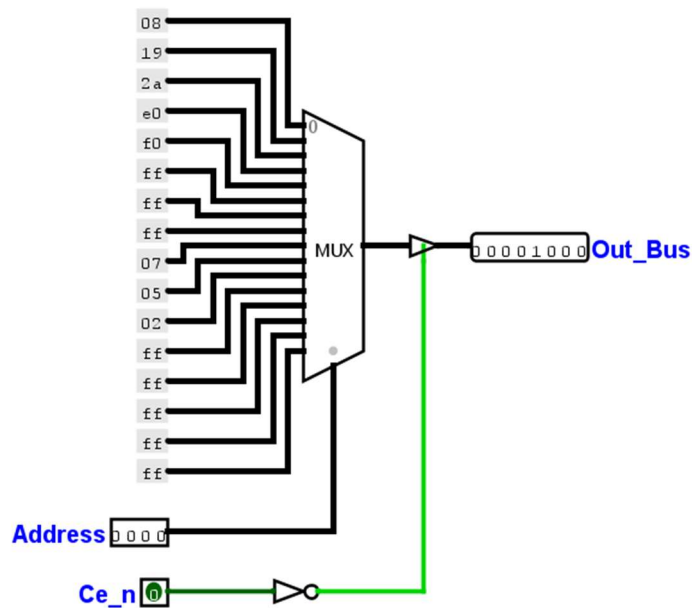
Endereço	Dado (assembly)	Dado
0x0	LDA 0x8	0x08
0x1	ADD 0x9	0x19
0x2	SUB 0xa	0x2a
0x3	OUT	0xe0
0x4	HLT	0xf0
0x8	0x07	0x07
0x9	0x05	0x05
0xa	0x02	0x02

Fonte: Autoria própria (2024).

3.4 Síntese em FPGA

Para a síntese em FPGA, a memória RAM foi substituída por um Multiplexador (MUX) para simplificar o processo de síntese, o módulo desse circuito é apresentado na Figura 5. O projeto foi sintetizado pelo *Logisim Evolution*, que precisa apenas que o *Vivado Design Suite* esteja instalado para implementar na FPGA *Basys 3*.

Figura 5 – RAM implementada com MUX



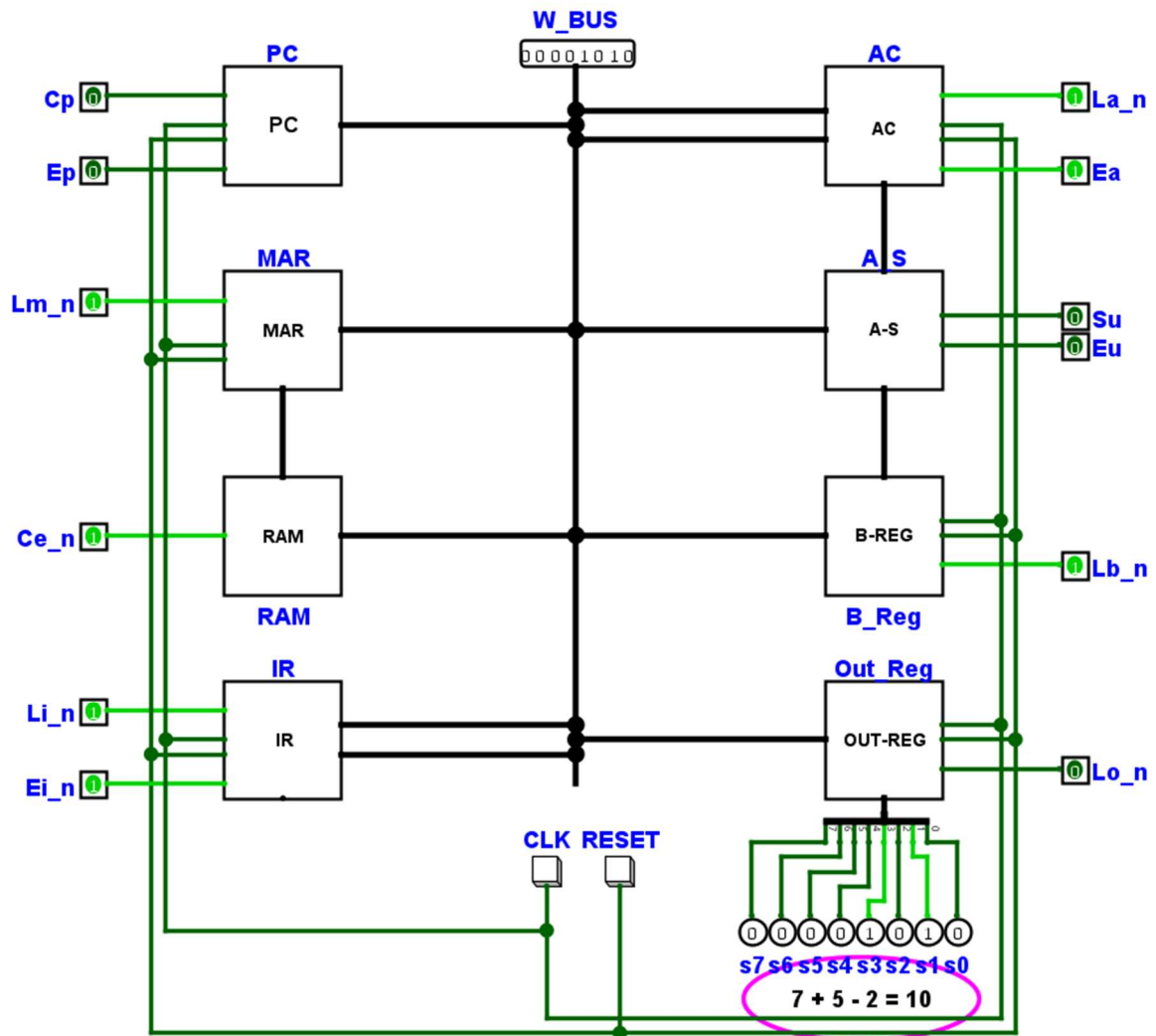
Fonte: Adaptado de Logisim Evolution (2024).

4 RESULTADOS E DISCUSSÕES

4.1 Resultados da Simulação

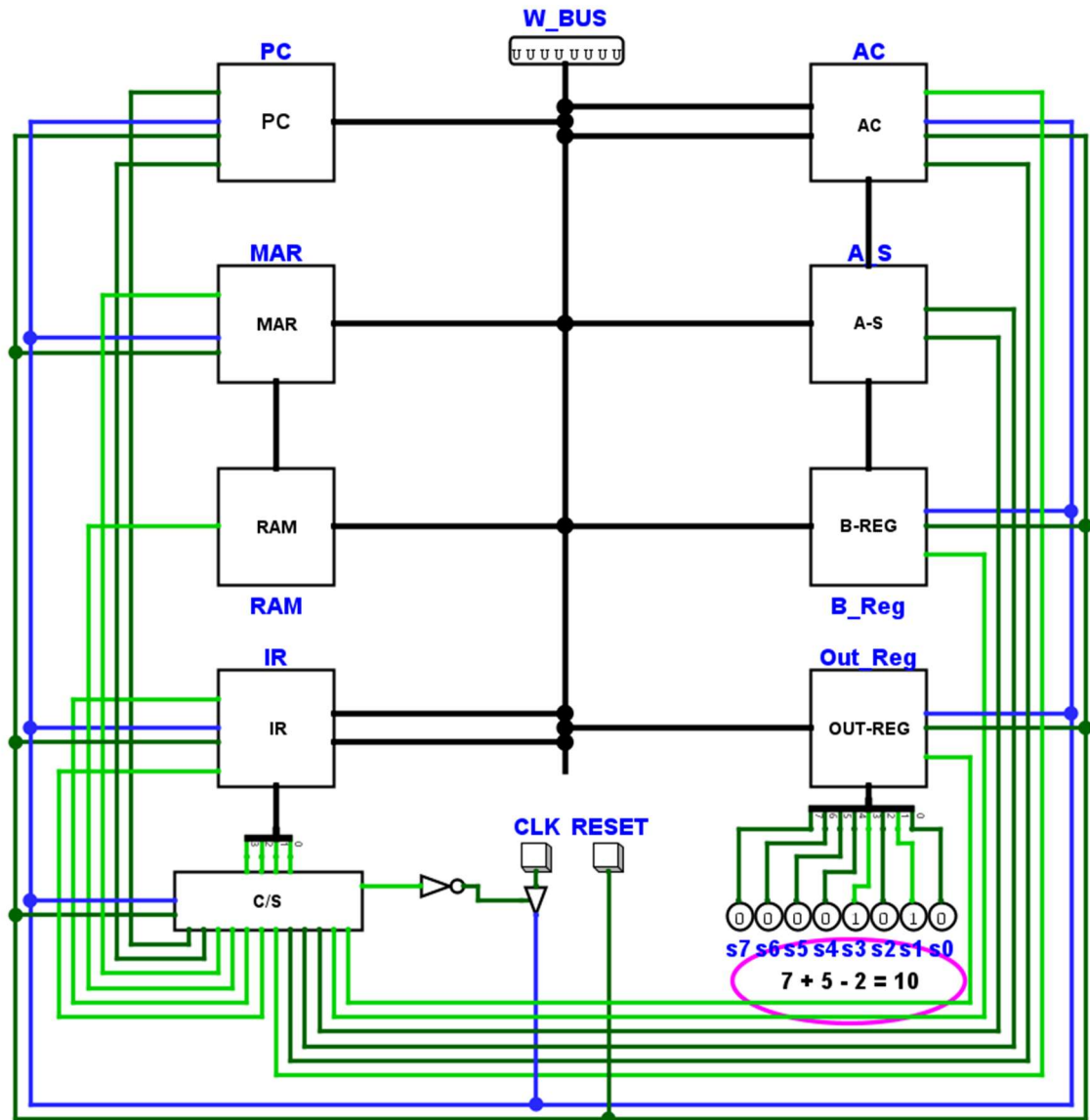
As simulações no *Logisim Evolution* foram bem-sucedidas, com o sistema executando corretamente as instruções para a operação $7 + 5 - 2$. O resultado final foi exibido no *Display* Binário, confirmando o funcionamento adequado do SAP-1. As Figuras 6 e 7 apresentam os resultados das Simulações do sistema com Controlador e sem, respectivamente.

Figura 6 – Simulação do SAP-1 sem Controlador



Fonte: Adaptado de Logisim Evolution (2024).

Figura 7 – Simulação do SAP-1 com Controlador

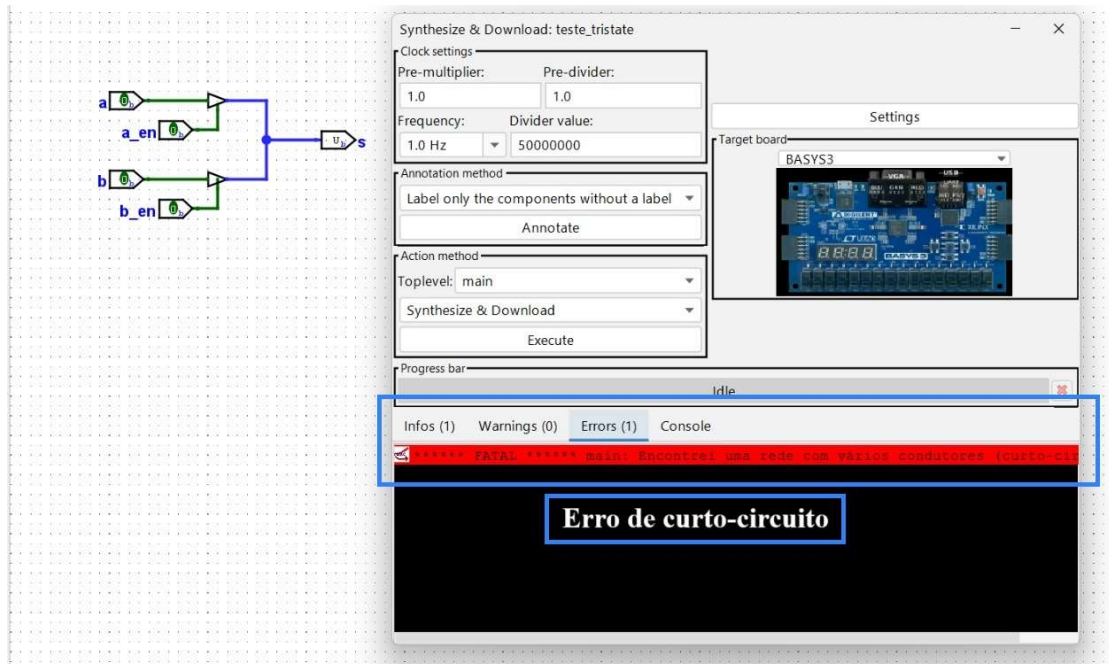


Fonte: Adaptado de Logisim Evolution (2024).

4.2 Resultados da Síntese em FPGA

A síntese inicial na FPGA *Basys 3* apresentou um erro de curto-circuito, impedindo a conclusão do processo. Para identificar a causa do problema, foi realizado um teste com um circuito simples, apresentado na Figura 8, que também não foi sintetizado corretamente. Através do teste, concluiu-se que o erro era causado pelos *buffers tri-state*, que o *Logisim Evolution* não consegue sintetizar adequadamente.

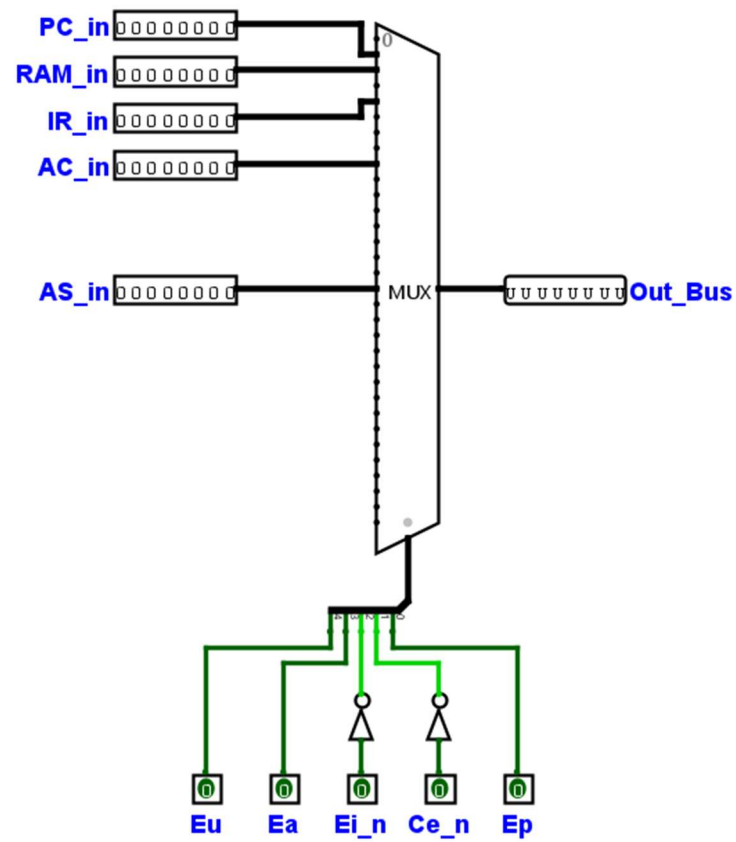
Figura 8 – Teste com Circuito Simples



Fonte: Adaptado de Logisim Evolution (2024).

Para resolver o problema, todos os *buffers tri-state* foram removidos e todas as saídas que apontam para o Barramento W foram direcionadas para um multiplexador (MUX). Uma nova simulação foi realizada, confirmando o funcionamento correto do sistema. A Figura 9 contém o módulo do MUX e a Figura 10 apresenta essa modificação simulada. A síntese foi então concluída, mas o teste na FPGA não produziu os resultados esperados, com os LEDs que representavam o *Display* Binário permanecendo apagados após a execução de todos os ciclos de instruções. Com isso, é possível que o *Logisim Evolution*, atualmente, não seja capaz de gerar arquivos de síntese adequados para circuitos complexos. Porém, são necessários mais estudos para verificar outras possibilidades.

Figura 9 – MUX para o Barramento W



Fonte: Adaptado de Logisim Evolution (2024).

5 CONCLUSÃO

Este projeto teve como objetivo implementar o computador SAP-1 utilizando o *Logisim Evolution* para projeto e simulação, seguido da síntese em uma placa FPGA *Basys 3*. A simulação no *Logisim* foi bem-sucedida, com o sistema executando corretamente as operações propostas. No entanto, a síntese em FPGA apresentou desafios técnicos, principalmente devido às limitações do *Logisim* em relação à síntese de *buffers tri-state*.

A substituição dos *buffers tri-state* por uma lógica com um multiplexador resolveu o problema de síntese, mas o teste na FPGA não produziu os resultados esperados. Isso sugere que o *Logisim Evolution* pode não ser a ferramenta mais adequada para a síntese de circuitos complexos em FPGA. Futuros trabalhos podem explorar o uso de outras ferramentas de projeto, como o próprio *Vivado*, para a implementação completa do SAP-1.

Apesar dos desafios, o projeto alcançou seu objetivo principal de proporcionar uma compreensão prática da arquitetura de computadores e do processo de projeto e síntese de sistemas digitais. Ele serviu como uma valiosa experiência de aprendizado, preparando os estudantes para desafios mais complexos na área de engenharia de computação.

REFERÊNCIAS

BASYS 3 ARTIX-7 FPGA TRAINER BOARD: RECOMMENDED FOR INTRODUCTORY USERS. Digilent. Disponível em: <https://digilent.com/shop/basys-3-artix-7-fpga-trainer-board-recommended-for-introductory-users/?srsltid=AfmBOopN70sIlw0rCA9vvahCOsrDnouyJA8SmHpV42YUY4ATwE2WrNiQ>. Acesso em: 15 fev. 2025.

CHUKWULOB, Chukwuemeka Akugbe; OKONKWO, Karen Chioma; SAHEED, Ibrahim Ayodeji; OKORIE, Joseph Ifeanyichukwu; ADEYEMO, Michael Temitayo. *Design and Implementation of a Simple-As-Possible 1 (SAP-1) Computer*. Projeto acadêmico apresentado ao curso "Computer Architecture (CPE 503)", supervisionado por ENGR. H. A. MOHAMMED, 2021.

DESIGNING AND IMPLEMENTING A SAP-1 COMPUTER. SAP-1-COMPUTER. Disponível em: <https://karenok.github.io/SAP-1-Computer/>. Acesso em: 17 jan. 2025.

GARCIA, F. D. (2016a, outubro 28). Operação do microprocessador SAP-1. Embarcados - Sua fonte de informações sobre Sistemas Embarcados; Embarcados. Disponível em: <https://embarcados.com.br/operacao-do-microprocessador-sap-1/>. Acesso em: 17 jan. 2025.

GARCIA, F. D. (2016b, novembro 3). *Projeto do SAP-1 construído no simulador*. Embarcados - Sua fonte de informações sobre Sistemas Embarcados; Embarcados. Disponível em: <https://embarcados.com.br/projeto-do-sap-1-construido-no-simulador/>. Acesso em: 17 jan. 2025.

LOGISIM [Português]. Cburch.com. Disponível em: <https://www.cburch.com/logisim/pt/index.html>. Acesso em: 17 jan. 2025.

MALVINO, A. P. *Digital Computer Electronics*, 2nd ed. McGraw-Hill, 1983.

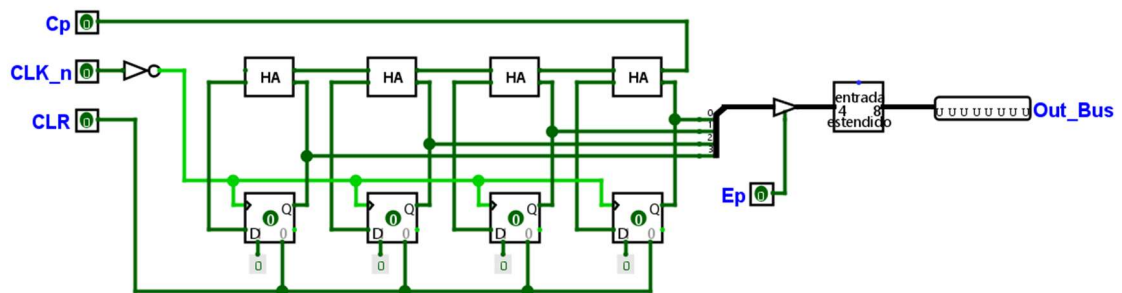
YECK, Lee Siang. *Simple-As-Possible Computer System Development on Field Programmable Gate Array*. 2007. Dissertação (Bacharel em Engenharia (Hons) em Engenharia Elétrica e Eletrônica) – *Universiti Teknologi PETRONAS*, Bandar Seri Iskandar, Tronoh, Perak Darul Ridzuan.

APÊNDICE A

A-1 Contador de Programa (PC)

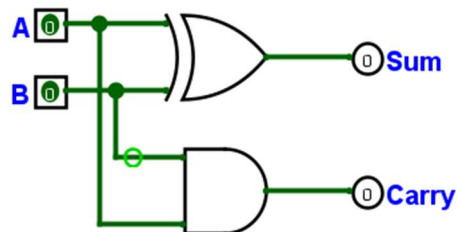
O contador de Programa (Figura 11) é um registrador constituído por 4 *Flip-Flops* Tipo D conectados a 4 Meio Somadores (Figura 12). O comando Cp ativa o incremento, e o comando Ep libera a saída por um *Buffer-Tristate*, que passa por um extensor de 4 para 8 bits onde a saída é direccionada para o Barramento W.

Figura 11 – Contador de Programa



Fonte: Adaptado de Logisim Evolution (2024).

Figura 12 – Meio Somador

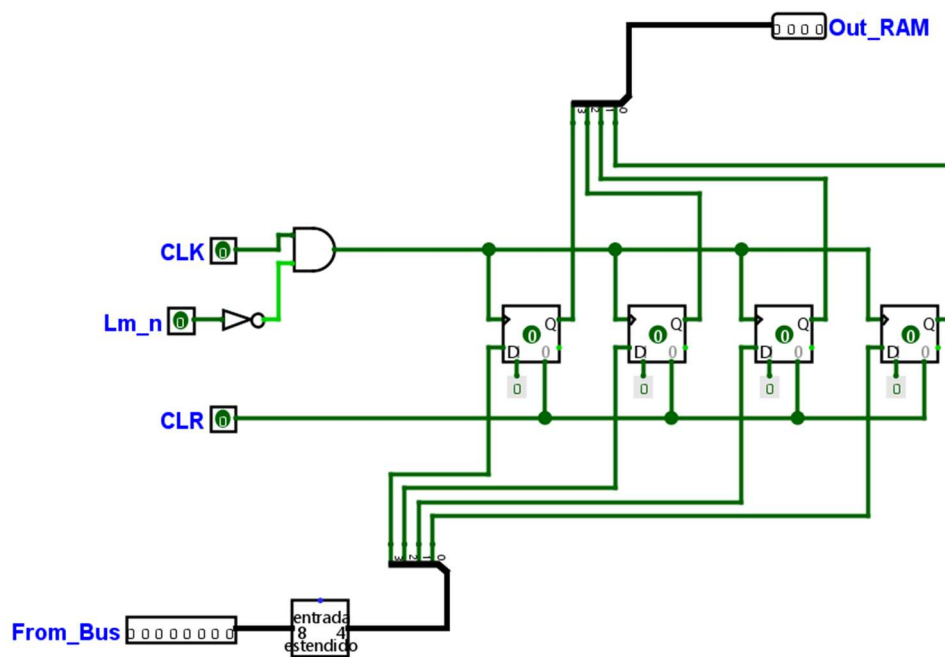


Fonte: Adaptado de Logisim Evolution (2024).

A-2 Registrador de Endereço de Memória (MAR)

O Registrador de Endereço de Memória (Figura 13) é um composto por 4 *Flip-Flops* Tipo D. A entrada de dados vem do Barramento W e passa por um extensor de 8 para 4 *bits*. O comando Lm' passa por uma porta lógica *AND* junto com o *clock* para habilitar os *Flip-Flops* que apontam a saída dos dados para a RAM.

Figura 13 – Registrador de Endereço de Memória

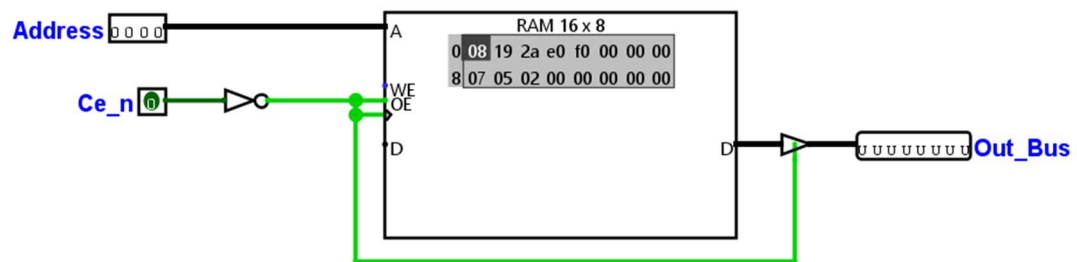


Fonte: Adaptado de Logisim Evolution (2024).

A-3 Memória RAM

A Memória RAM (Figura 14) tem 16 palavras de 8 *bits*. A entrada de endereço de 4 *bits*, que vem do MAR, seleciona os endereços que contêm os dados ou instruções. O comando *Ce'* é usado como *clock* e também habilita a saída dos dados através de um Buffer-Tristate que aponta para o Barramento W.

Figura 14 – Memória RAM

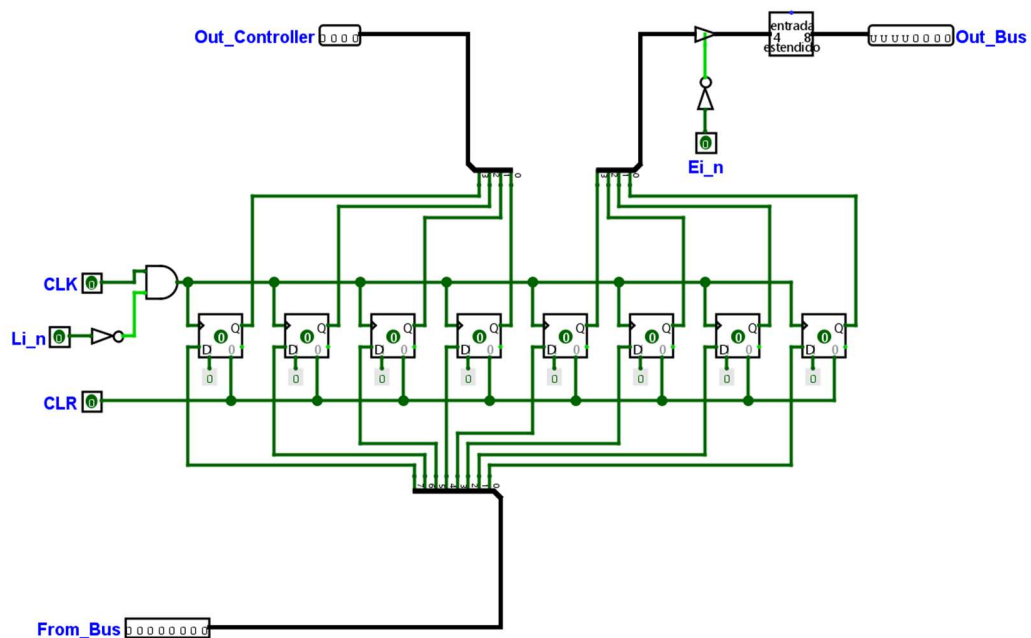


Fonte: Adaptado de Logisim Evolution (2024).

A-4 Registrador de Instrução (IR)

O Registrador de Instrução (Figura 15) é composto por 8 *Flip-Flops* Tipo D. A entrada de dados vem do Barramento W, e o comando Li' passa por uma porta lógica *AND* junto com o *clock* para habilitar os *Flip-Flops*, que direcionam os *bits* mais significativos para o Controlador e os *bits* menos significativos para um *Buffer-Tristate* ativado pelo comando Ei' , que direciona os dados para um extensor de 4 para 8 *bits* que tem a saída apontada para o Barramento W.

Figura 15 – Registrador de Instrução

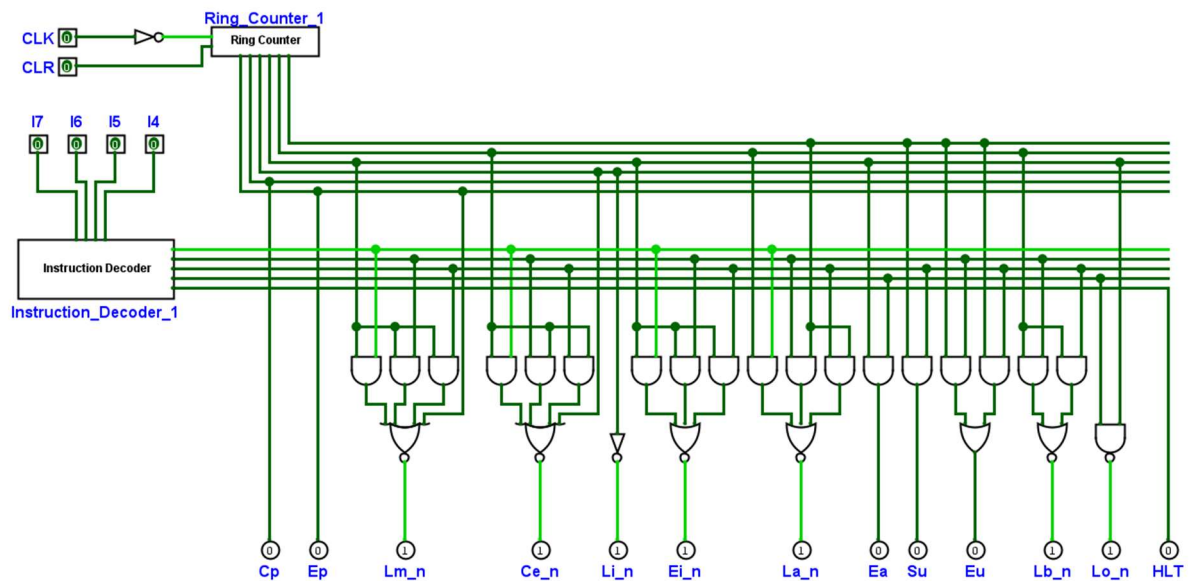


Fonte: Adaptado de Logisim Evolution (2024).

A-5 Controlador/Sequenciador

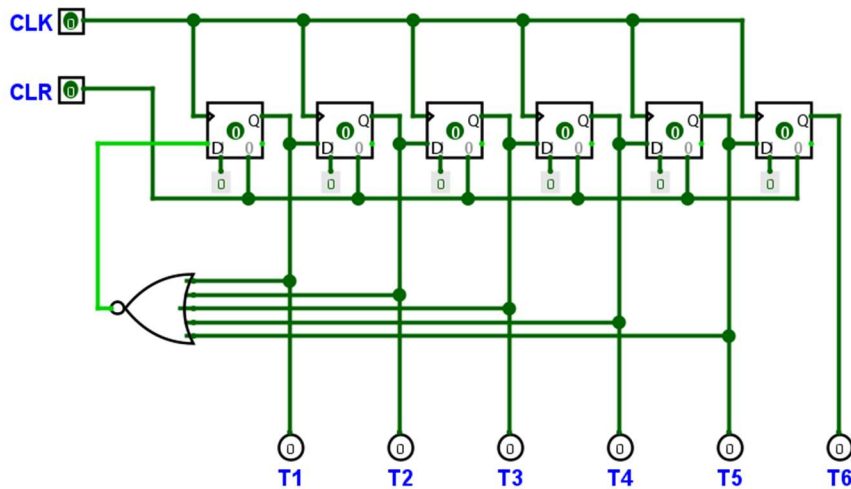
O Controlador/Sequenciador (Figura 16) é um circuito combinacional composto pelo Decodificador de Instrução (Figura 17), Contador em Anel (Figura 18) e diversas portas lógicas que, juntos, originam os sinais de comandos.

Figura 16 – Controlador/Sequenciador



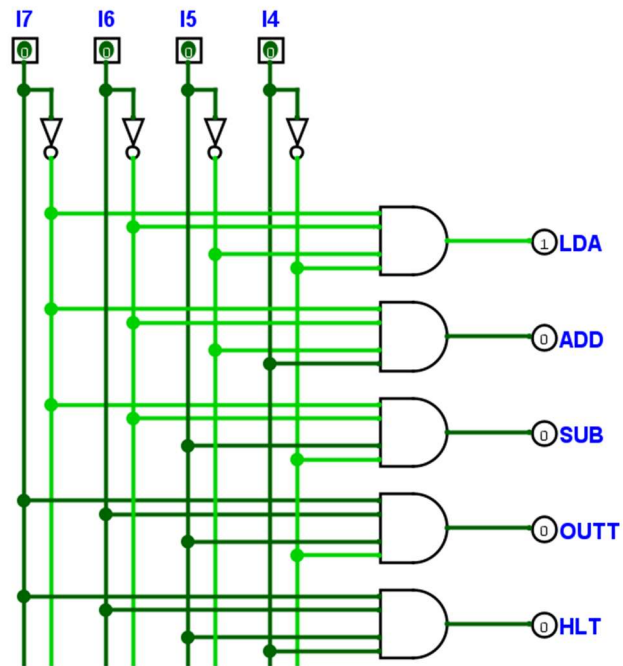
Fonte: Adaptado de Logisim Evolution (2024).

Figura 17 – Contador em Anel



Fonte: Adaptado de Logisim Evolution (2024).

Figura 18 – Decodificador de Instrução

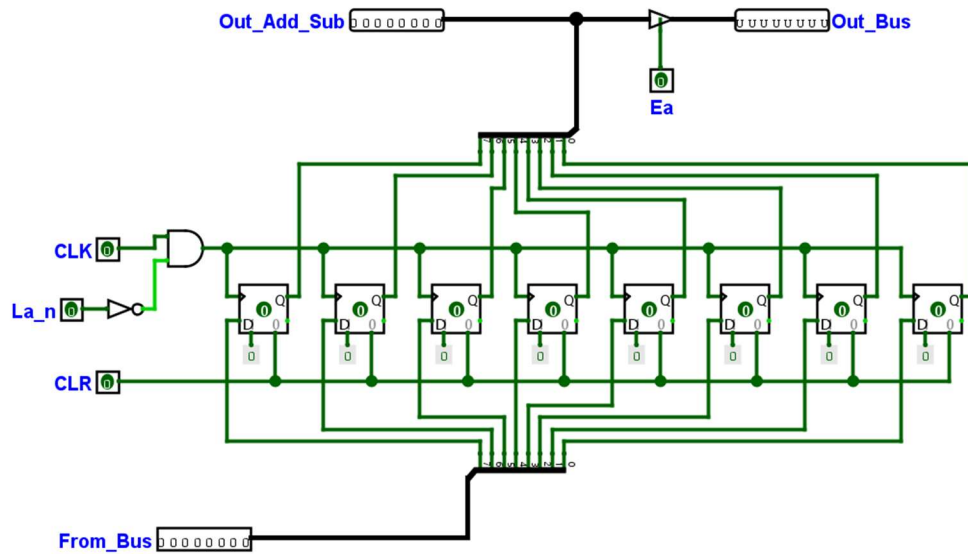


Fonte: Adaptado de Logisim Evolution (2024).

A-6 Acumulador

O Acumulador (Figura 19) é um registrador composto por 8 *Flip-Flops* Tipo D que recebem os dados do Barramento W e é habilitado pelo comando La', que passa por uma porta lógica *AND* junto com o *clock*. Tem uma saída para a ULA e outra saída para o Barramento W, que passa por um Buffer-Tristate ativado pelo comando Ea.

Figura 19 – Acumulador

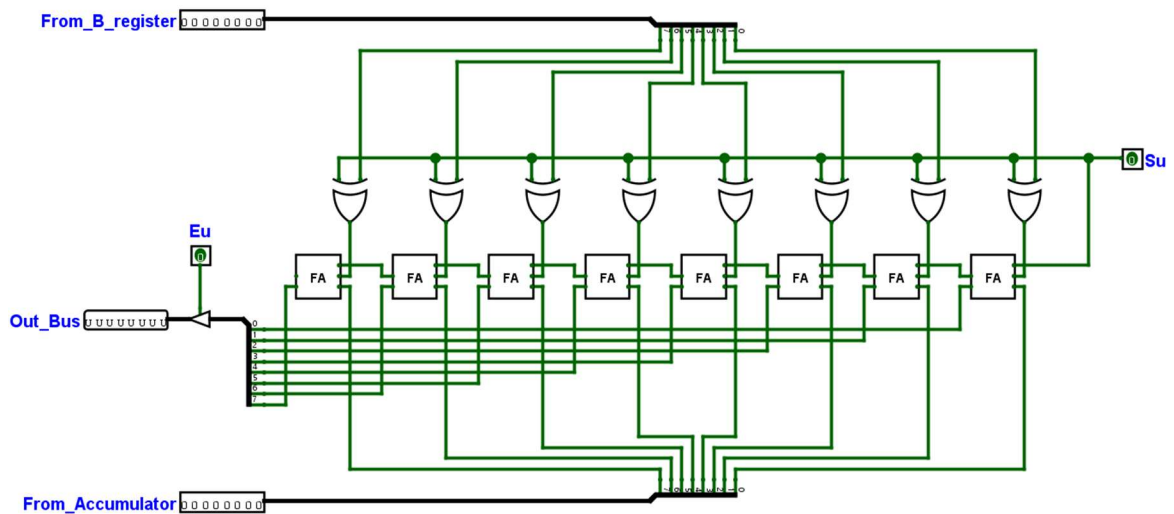


Fonte: Adaptado de Logisim Evolution (2024).

A-7 Somador-Subtrator (ULA)

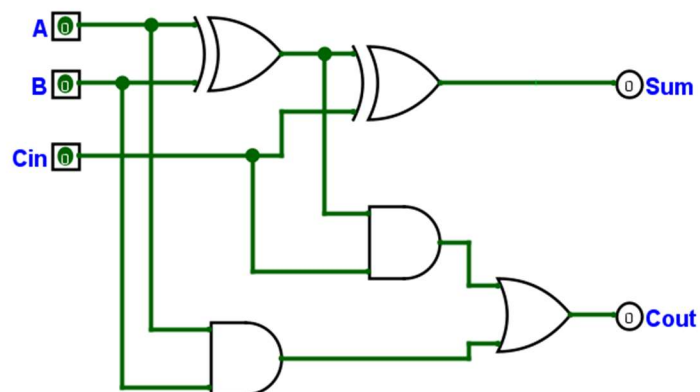
O Somador-Subtrator (Figura 20) é composto por 8 Somadores Completos (Figura 21). Suas entradas vêm do Acumulador e Registrador B. O comando Su passa por portas lógicas *XOR* e habilita a operação de subtração, e o comando Eu habilita a saída para o Barramento W.

Figura 20 – Somador-Subtrator



Fonte: Adaptado de Logisim Evolution (2024).

Figura 21 – Somador Completo

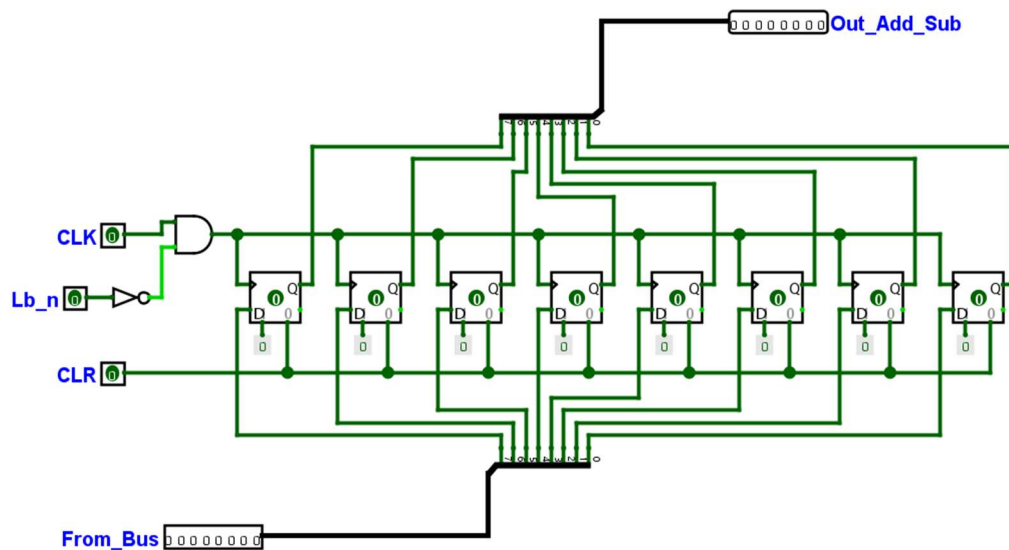


Fonte: Adaptado de Logisim Evolution (2024).

A-8 Registrador B

O Registrador B (Figura 22) é composto por 8 *Flip-Flops* Tipo D que recebem os dados do Barramento W e é habilitado pelo comando Lb', que passa por uma porta lógica *AND* junto com o *clock*. Tem sua saída apontada para a ULA.

Figura 22 – Registrador B

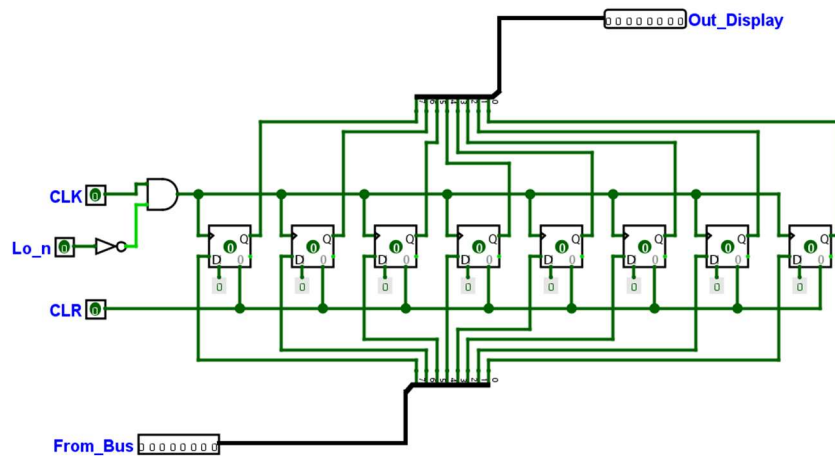


Fonte: Adaptado de Logisim Evolution (2024).

A-9 Registrador de Saída

O Registrador de Saída (Figura 23) é composto por 8 *Flip-Flops* Tipo D que recebem os dados do Barramento W e é habilitado pelo comando Lo', que passa por uma porta lógica *AND* junto com o *clock*. Tem sua saída apontada para o *Display* Binário.

Figura 23 – Registrador de Saída



Fonte: Adaptado de Logisim Evolution (2024).