UNIVERSIDADE FEDERAL DE RORAIMA

CENTRO DE CIÊNCIA E TECNOLOGIA

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

RELATÓRIO UNIFICADO: ANÁLISE E VERIFICAÇÃO DE CIRCUITOS LÓGICOS, FLUXO DE EXECUÇÃO NO MIPS E SOMADOR COMPLETO DE 8 BITS

Boa Vista – RR
2025

ACADÊMICOS:

LUCIANO DOS SANTOS NASCIMENTO

WESLEY SILVA ARAÚJO

RELATÓRIO UNIFICADO: ANÁLISE E VERIFICAÇÃO DE CIRCUITOS LÓGICOS, FLUXO DE EXECUÇÃO NO MIPS E SOMADOR COMPLETO DE 8 BITS

Trabalho Avaliativo para matéria de Arquitetura e Organização de Computadores em Ciências da Computação da Universidade de Roraima - UFRR.

Orientador: Herbert Oliveira Rocha

Boa Vista - RR
2025

Sumário

RESUMO....1
INTRODUÇÃO....2
2. ANÁLISE DE CIRCUITOS LÓGICOS COM FÓRMULAS BOOLEANAS E Z3....3
2.1 Conversão para Fórmulas Booleanas....3
Os circuitos lógicos foram convertidos em expressões booleanas para facilitar a análise. Por exemplo, a função F foi representada como:....3
2.2 Implementação e Verificação com Z3....3
2.3 Resultados....3
3. FLUXO DE EXECUÇÃO DE UMA INSTRUÇÃO DO TIPO R NO MIPS DE 32 BITS....3
3.1 Descrição do Fluxo de Execução....4
3.2 Implementação e Verificação com Z3....4
3.3 Resultados....4
4. ANÁLISE E VERIFICAÇÃO DO SOMADOR COMPLETO DE 8 BITS....4
4.1 Descrição do Somador de 8 Bits....4
4.2 Implementação e Verificação com Z3....5
4.3 Resultados....5
5.0 Unidade de Controle (UC) Codificada em uma Máquina de Estados Finitos Completa....5
Objetivo:....5
Circuito Original:....6
Simplificação:....6
Verificação com Z3:....6
6. Unidade Lógica e Aritmética (ULA) de 8 bits....6
Objetivo:....6
Fórmulas Booleanas Completas e Simplificadas:....6
Verificação com Z3:....7
3. Verificação da Fórmula Booleana f=((~y+z)·x)·(~x+y)·~(y+(~y·x))f=((~y+z)·x)·(~x+y)·~(y+(~y·x))....7
Objetivo:....7
Fórmula Original:....7
Simplificação:....7
Verificação com Z3:....8
REFERÊNCIAS....8

RESUMO

Este relatório unificado apresenta a análise e verificação de três sistemas: circuitos lógicos utilizando fórmulas booleanas, o fluxo de execução de instruções do tipo R no processador MIPS de 32 bits e o funcionamento de um somador completo de 8 bits. A validação foi realizada com o auxílio do solver Z3, garantindo a corretude e eficiência dos sistemas. Os resultados demonstraram que as implementações estão em conformidade com as especificações teóricas, sem inconsistências ou falhas.

INTRODUÇÃO

Este relatório tem como objetivo analisar e verificar três sistemas distintos: circuitos lógicos, o fluxo de execução de instruções no processador MIPS e o funcionamento de um somador binário de 8 bits. A abordagem utiliza o solver Z3 para modelagem e verificação formal, garantindo que os sistemas operem conforme o esperado. A análise inclui a identificação de redundâncias, a validação do pipeline de execução e a confirmação da propagação correta do carry no somador.

2. ANÁLISE DE CIRCUITOS LÓGICOS COM FÓRMULAS BOOLEANAS E Z3

2.1 Conversão para Fórmulas Booleanas

Os circuitos lógicos foram convertidos em expressões booleanas para facilitar a análise. Por exemplo, a função **F** foi representada como:

$$F=ABCD+ABC'D+ABC'D'+AB'CD+A'BCD+A'BCD'+A'BC'D+A'B'C'D$$

Essa expressão combina variáveis booleanas e operações lógicas (AND, OR) para descrever o comportamento do circuito.

2.2 Implementação e Verificação com Z3

O solver Z3 foi utilizado para:

- Validar a saída do circuito para diferentes entradas.
- Identificar e remover redundâncias na expressão booleana.
- Verificar a equivalência lógica entre diferentes implementações.

2.3 Resultados

Os testes confirmaram que:

- A função **F** atendeu às expectativas para todas as combinações de entrada.
- Redundâncias foram identificadas e removidas, otimizando a expressão.
- A equivalência lógica entre implementações foi validada com sucesso.

3. FLUXO DE EXECUÇÃO DE UMA INSTRUÇÃO DO TIPO R NO MIPS DE 32 BITS

3.1 Descrição do Fluxo de Execução

As instruções do tipo R no MIPS são executadas em cinco estágios:

1. **Busca da Instrução (IF):** A instrução é carregada da memória para o registrador de instrução (IR).
2. **Decodificação da Instrução (ID):** Os registradores-fonte e o campo funct são identificados.
3. **Execução (EX):** A ULA realiza a operação especificada pelo campo funct.
4. **Acesso à Memória (MEM):** Não utilizado para instruções do tipo R.
5. **Escrita no Registrador (WB):** O resultado é armazenado no registrador de destino.

3.2 Implementação e Verificação com Z3

O solver Z3 foi utilizado para modelar o pipeline e verificar a sequência de estágios (IF → ID → EX → WB). A validação incluiu:

- Garantir a ordem correta dos estágios.
- Verificar a ausência de conflitos ou hazards.

3.3 Resultados

Os testes confirmaram que:

- O fluxo de execução seguiu a sequência esperada.
- Nenhuma inconsistência foi detectada na execução das instruções.

4. ANÁLISE E VERIFICAÇÃO DO SOMADOR COMPLETO DE 8 BITS

4.1 Descrição do Somador de 8 Bits

O somador de 8 bits é composto por oito somadores completos de 1 bit, conectados em cadeia. Cada somador de 1 bit possui:

- Entradas: **A[i]**, **B[i]**, **Cin**.

- Saídas: **S[i]** (soma) e **Cout** (carry de saída).

As expressões booleanas para um somador de 1 bit são:

$$S=A\oplus B\oplus Cin$$

$$S=A\oplus B\oplus Cin$$

$$Cout=(A\cdot B)+(Cin\cdot(A\oplus B))$$

$$Cout=(A\cdot B)+(Cin\cdot(A\oplus B))$$

4.2 Implementação e Verificação com Z3

O solver Z3 foi utilizado para:

- Validar a saída da soma para diferentes combinações de entrada.
- Verificar a propagação correta do carry entre os estágios.
- Confirmar a equivalência do circuito com a operação de soma binária.

4.3 Resultados

Os testes demonstraram que:

- O somador de 8 bits produziu resultados corretos para todas as entradas testadas.
- A propagação do carry foi validada em todos os estágios.
- Nenhuma inconsistência foi encontrada na modelagem do circuito.

5.0 **Unidade de Controle (UC) Codificada em uma Máquina de Estados Finitos Completa**

Objetivo:

A Unidade de Controle (UC) é responsável por gerar os sinais de controle que coordenam as operações do processador multiciclo MIPS. O objetivo foi verificar a redundância no circuito da UC e simplificar a lógica.

Circuito Original:

A fórmula booleana original da UC era:

Foriginal = ((A∧B)∨C)∨C Foriginal=((A∧B)∨C)∨C

Simplificação:

Aplicando a propriedade de idempotência da porta OR (X∨X=XX∨X=X), a fórmula foi simplificada para:

Fsimplificada = (A∧B)∨C Fsimplificada =(A∧B)∨C

Verificação com Z3:

- A verificação com o Z3 confirmou que as duas fórmulas são equivalentes (UNSAT), ou seja, a redundância pode ser removida sem alterar o comportamento do circuito.

6. Unidade Lógica e Aritmética (ULA) de 8 bits

Objetivo:

A ULA de 8 bits foi projetada para realizar as seguintes operações:

1. Subtração (A−BA−B)
2. XOR (A⊕BA⊕B)
3. NAND (¬(A∧B)¬(A∧B))
4. NOR (¬(A∨B)¬(A∨B))
5. Shift de 2 bits à esquerda (A<<2A<<2)

Fórmulas Booleanas Completas e Simplificadas:

Operação	Fórmula Booleana Completa	Fórmula Booleana Simplificada
Subtração	A+(∼B+1)A+(∼B+1)	A−BA−B (implementação direta em hardware)
XOR	(Ai∧¬Bi)∨(¬Ai∧Bi)(Ai∧¬Bi)∨(¬Ai∧Bi)	Ai⊕BiAi⊕Bi (já simplificado)
NAND	¬(Ai∧Bi)¬(Ai∧Bi)	¬(Ai∧Bi)¬(Ai∧Bi) (já simplificado)
NOR	¬(Ai∨Bi)¬(Ai∨Bi)	¬(Ai∨Bi)¬(Ai∨Bi) (já simplificado)
Shift Left	Ai−2Ai−2 (para i≥2i≥2), 00 (para i<2i<2)	A<<2A<<2 (implementação direta)

Verificação com Z3:

- Cada operação foi verificada com o Z3 para garantir que o comportamento está correto.
- Foram testados valores específicos para as entradas AA e BB, e as saídas foram comparadas com os resultados esperados.
- Todos os testes retornaram SAT, confirmando que as operações estão corretas.

3. *Verificação da Fórmula Booleana f=((∼y+z)⋅x)⋅(∼x+y)⋅∼(y+(∼y⋅x))f=((∼y+z)⋅x)⋅(∼x+y)⋅∼(y+(∼y⋅x))*

Objetivo:

Verificar a fórmula booleana fornecida, simplificá-la e confirmar se a versão simplificada é equivalente à original.

Fórmula Original:

f=((∼y+z)⋅x)⋅(∼x+y)⋅∼(y+(∼y⋅x))f=((∼y+z)⋅x)⋅(∼x+y)⋅∼(y+(∼y⋅x))

Simplificação:

1. ∼y+z∼y+z é equivalente a y→zy→z.
2. ∼x+y∼x+y é equivalente a x→yx→y.
3. y+(∼y⋅x)y+(∼y⋅x) simplifica para y+xy+x.
4. ∼(y+x)∼(y+x) é equivalente a ∼y⋅∼x∼y⋅∼x.
5. Substituindo na fórmula original:

   f=x⋅(y→z)⋅(x→y)⋅∼y⋅∼xf=x⋅(y→z)⋅(x→y)⋅∼y⋅∼x
6. Como x⋅∼xx⋅∼x é sempre falso, a fórmula simplificada é:

   fsimplificada=Falsefsimplificada=False

Verificação com Z3:

- A fórmula original e a fórmula simplificada foram comparadas usando o Z3.
- O Z3 retornou UNSAT, confirmando que as duas fórmulas são equivalentes.

REFERÊNCIAS

BROWN, S.; VRANESIC, Z. *Fundamentals of Digital Logic with VHDL Design*. 3. ed. McGraw-Hill, 2013.

PATTERSON, D. A.; HENNESSY, J. L. *Computer Organization and Design: The Hardware/Software Interface*. 5. ed. Morgan Kaufmann, 2017.

Documentação do solver Z3: https://ericpony.github.io/z3py-tutorial/.

Material didático sobre circuitos digitais, arquitetura de computadores e somadores binários.