

Trabalho Prático 1 - Introdução aos Sistemas Lógicos

1. Introdução

A proposta deste trabalho prático é desenvolver um circuito combinatório em Verilog que gere o resto da divisão por 7 para uma mensagem binária de 4 bits. É solicitado todas as etapas da implementação do circuito e o caso de teste para mensagens recebidas durante 5 dias.

Esta documentação apresenta as seguintes informações:

1. Introdução
2. Decisões de implementação (a)
3. Cálculo de saídas
 - (b) Forma canônica de cada saída
 - (c) Forma simplificada com PoS
 - (d) Diagrama do circuito resultante
 - (e) Mapa de Karnaugh de cada saída com simplificação SoP
 - (f) Tabela Verdade
4. Testes e visualização
 - (+) Saídas no display com 16 testes e dos 5 dias especificados
 - (g) Visualização das formas de onda do caso de teste
5. Conclusões
6. Bibliografia

2. Decisões de Implementação (a)

A partir das simplificações do mapa de Karnaugh para Produto de Somas e para Soma de Produtos foi escolhida a implementação com SoP pois a mesma apresenta menos operações em comparação com a mesma função na forma de PoS para a maior parte das saídas de interesse (Z0 a Z2).

Além disso, para implementação do circuito em verilog utilizando a ferramenta EDA Playground foi decidido testar todas as 16 possibilidades de entrada da tabela verdade (de 0000 a 1111) para confirmação do funcionamento do circuito e do código. O teste com todas as entradas foi apagado para entrega do trabalho, mantendo apenas dos 5 dias especificados no enunciado, mas sua saída tem a seguinte forma de onda:



Figura 1 - Forma de Onda de teste com as 16 entradas

Por fim, no código de testes foi adicionado o comando “#10” antes do display para que após as instruções iniciais a simulação aguarde 10 unidades de tempo antes de exibir as saídas, permitindo assim uma melhor visualização desde a primeira saída.

3. Cálculo das saídas

(b) A forma canônica de cada saída usando a notação $\sum m$ ou a notação $\prod M$

$$Z_0 = F(A, B, C, D) = \sum m(1, 3, 5, 8, 10, 12, 15)$$

$$Z_0 = F(A, B, C, D) = \prod M(0, 2, 4, 6, 7, 9, 11, 13, 14)$$

$$Z_1 = F(A, B, C, D) = \sum m(2, 3, 6, 9, 10, 13)$$

$$Z_1 = F(A, B, C, D) = \prod M(0, 1, 4, 5, 7, 8, 11, 12, 14, 15)$$

$$Z_2 = F(A, B, C, D) = \sum m(4, 5, 6, 11, 12, 13)$$

$$Z_2 = F(A, B, C, D) = \prod M(0, 1, 2, 3, 7, 8, 9, 10, 14, 15)$$

(c) A forma simplificada de cada saída como um Produto de Somas (PoS)

$$Z_0 = (A+D) (A+B'+C') (B'+C'+D) (A'+B+D')(A'+C+D')$$

$$Z_1 = (C+D)(A'+B'+C')(A'+C'+D')(A+B'+D')(A+C)$$

$$Z_2 = (B+C)(A'+C'+D)(B'+C'+D')(A+B)$$

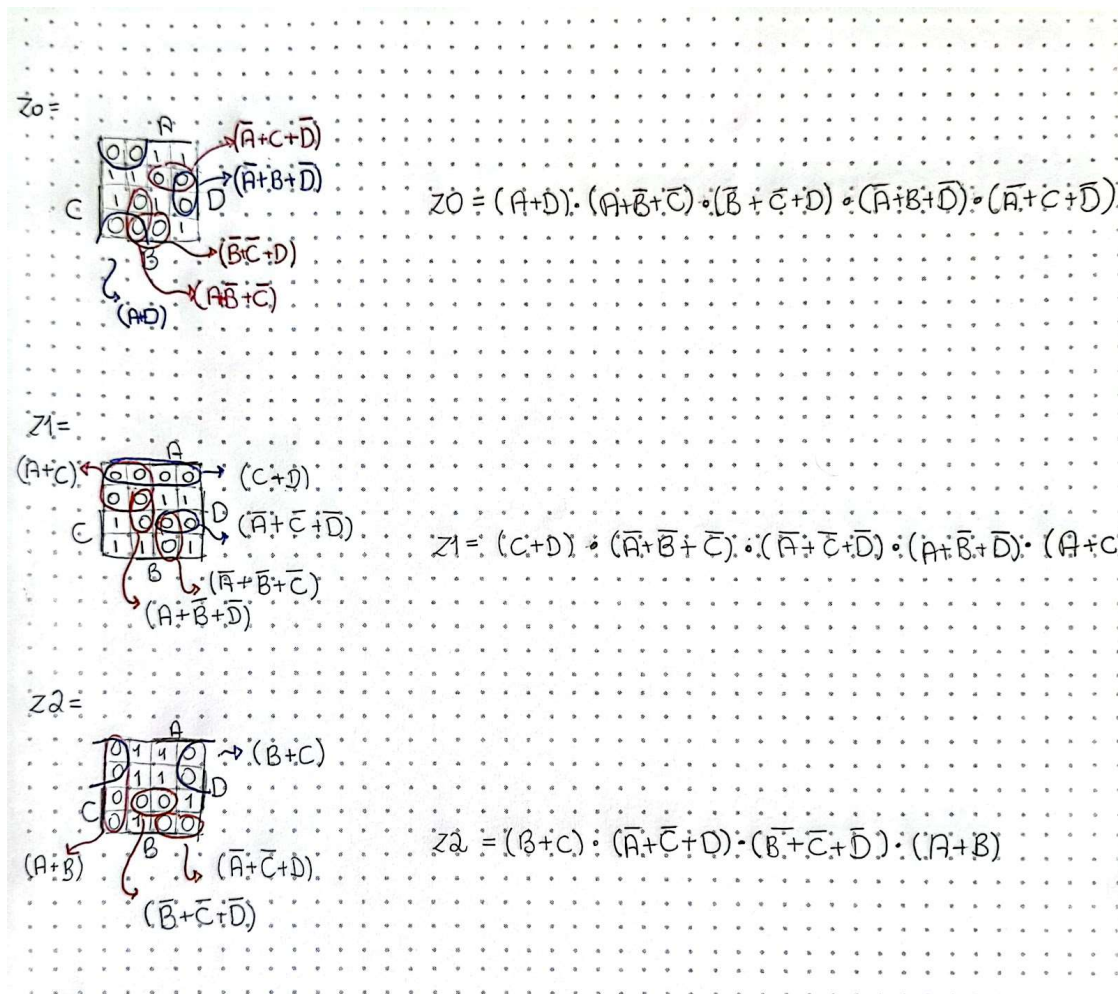


Figura 2 - Simplificação do Mapa de Karnaugh para PoS

(d) O diagrama do circuito resultante

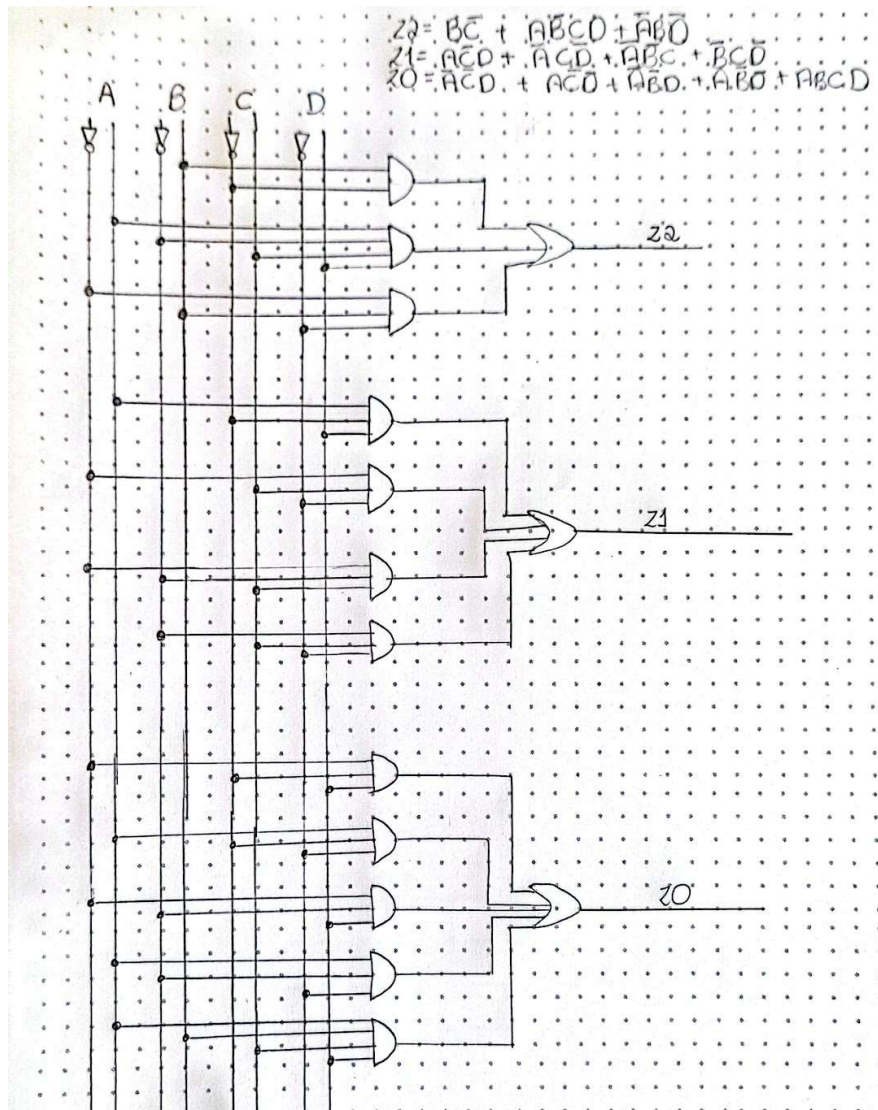


Figura 3 - Circuito resultante utilizando SoP

(e) O mapa de Karnaugh de cada saída

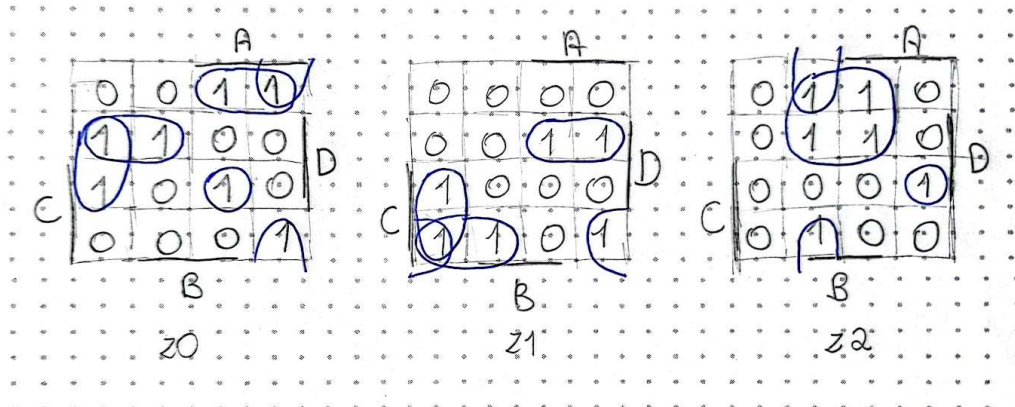


Figura 4 - Mapa de Karnaugh simplificado usando SoP

$$Z2 = BC' + AB'CD + A'BD'$$

$$Z1 = AC'D + A'CD' + A'B'C + B'CD'$$

$$Z0 = A'C'D + AC'D' + A'B'D + AB'D' + ABCD$$

Aluna: Carla Beatriz Ferreira (2022097470)
 Disciplina: Introdução aos Sistemas Lógicos (TW)
 Professor: Marcos Augusto Menezes Vieira
 Data: 06/10/2023

(f) A tabela verdade

	A	B	C	D	Z2	Z1	Z0
0	0	0	0	0	0	0	0
1				1	0	0	1
2			1	0	0	1	0
3				1	0	1	1
4		1	0	0	1	0	0
5				1	1	0	1
6			1	0	1	1	0
7				1	0	0	0
8	1	0	0	0	0	0	1
9				1	0	1	0
10			1	0	0	1	1
11				1	1	0	0
12		1	0	0	1	0	1
13				1	1	1	0
14			1	0	0	0	0
15				1	0	0	1

4. Testes e visualização

Conforme discutido nas decisões de implementação, foram realizados testes com as 16 possibilidades de entrada (da tabela verdade) e posteriormente apenas com as 5 entradas do enunciado do trabalho prático. Com esses testes, obtivemos as saídas (display no log e formas de onda) exemplificadas abaixo nas imagens 5, 6 e 7.

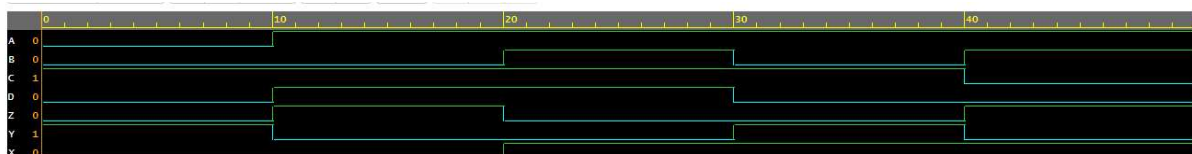
```
[2023-11-08 14:42:29 UTC] iverilog '-Wall' '-g2012' design.sv testbench.sv && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
Saídas 0: Z2=0, Z1=0, Z0=0
Saídas 1: Z2=0, Z1=0, Z0=1
Saídas 2: Z2=0, Z1=1, Z0=0
Saídas 3: Z2=0, Z1=1, Z0=1
Saídas 4: Z2=1, Z1=0, Z0=0
Saídas 5: Z2=1, Z1=0, Z0=1
Saídas 6: Z2=1, Z1=1, Z0=0
Saídas 7: Z2=0, Z1=0, Z0=0
Saídas 8: Z2=0, Z1=0, Z0=1
Saídas 9: Z2=0, Z1=1, Z0=0
Saídas 10: Z2=0, Z1=1, Z0=1
Saídas 11: Z2=1, Z1=0, Z0=0
Saídas 12: Z2=1, Z1=0, Z0=1
Saídas 13: Z2=1, Z1=1, Z0=0
Saídas 14: Z2=0, Z1=0, Z0=0
Saídas 15: Z2=0, Z1=0, Z0=1
Done
```

Figura 5 - Saídas do caso de teste com as 16 possibilidades de entrada

```
[2023-11-08 14:40:11 UTC] iverilog '-Wall' '-g2012' design.sv testbench.sv && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
DIA 1 (0010): 0 1 0
DIA 2 (1011): 1 0 0
DIA 3 (1111): 0 0 1
DIA 4 (1010): 0 1 1
DIA 5 (1100): 1 0 1
Done
```

Figura 6 - Saídas do caso de teste dos 5 dias da especificação

(g) A visualização das formas de onda do caso de teste



5. Conclusão

A partir desta atividade prática para a resolução do problema proposto pude implementar um circuito que retorna o resto da divisão por 7 de uma entrada de 4 bits e aprimorar meus conhecimentos em todas as etapas da implementação do circuito, aprendendo e praticando a linguagem Verilog.

Além disso, foi um ótimo exercício de revisão juntamente com o aprendizado do novo conteúdo apresentado pela disciplina de Introdução aos Sistemas Lógicos e durante sua modelagem e execução pude perceber minhas maiores dificuldades: simplificação do circuito em PoS e implementação em uma linguagem nova. O código implementado na plataforma também está disponível no link <https://edaplayground.com/x/adiQ>.

6. Referências

Slides virtuais da disciplina de Introdução aos Sistemas Lógicos. Disponibilizado via moodle. Departamento de Ciência da Computação. Universidade Federal de Minas Gerais. Belo Horizonte.

Examples. EDA Playground. Disponível em <<https://edaplayground.com/x/adiQ>>. Acesso em 30 de outubro de 2023.