Relatório da Lista 1 - Projeto de Sistemas em Chip

Jessé Barreto de Barros
Matrícula: 17/0067033
Sistemas Mecatrônicos
Universidade de Brasília

I. Exercício 1 - Portas Lógicas

a) Escreva o código VHDL para o circuito da figura 1. Observe que o circuito é puramente combinacional, portanto, não é necessário o uso de processos. Escreva uma expressão para d usando apenas operadores lógicos (não será aceito o uso de tabela verdade).

b) Realize uma simulação comportamental e verifique que o circuito funciona apropriadamente usando 8 possíveis valores para as entradas.

c) Sintetize o circuito. Abra o o reporte de síntese e verifique a expressão obtida pela ferramenta de síntese. Compare com a sua implementação. São expressões booleanas identicas?

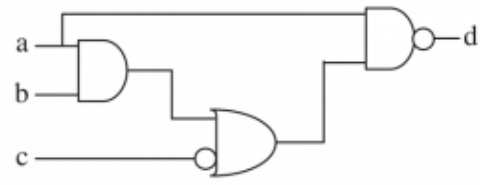

Figura 1. Circuito do Exercício 1.

A. Respostas

a) A expressão booleana para o circuito combinacional da Figura 1 é $d = \overline{a(ab + \bar{c})}$. Na sintaxe do VHDL essa expressão booleana é representada por

```
d <= not (a and ((a and b) or not c));
```

b) Os sinais de saída da simulação comportamental estão representados na Figura 2. Analisando as entradas e as saídas obtidas na simulação é possível visualizar que o circuito funciona corretamente para os 8 possíveis combinações de sinais de entrada.

c) O circuito sintetizado pelo Vivado esta presente na Figura 3 e a Figura 4 estão presentes, respectivamente, o circuito sintetizado em uma LUT e a tabela-verdade com a expressão booleana dessa LUT. A expressão booleana encontrada foi $d = \overline{a} + \overline{b}c$.

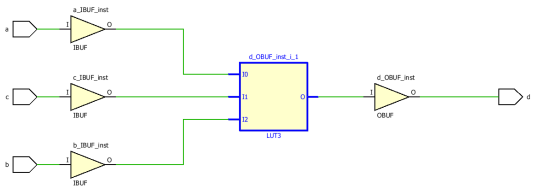

Figura 3. Circuito obtida através da síntese do Vivado.

I0	I2	I1	O=!I0 + I1 & !I2
0	0	0	1
1	0	0	0
0	0	1	1
1	0	1	1
0	1	0	1
1	1	0	0
0	1	1	1
1	1	1	0

Figura 4. Tabela verdade e expressão booleana da LUT utilizada pelo circuito sintetizado pelo Vivado. Nesse circuito os sinais foram renomedos então a equivalência para o circuito original é (I0, I2, I1) = (a, b, c).

Pela tabela verdade do circuito da Figura 1 é possível verificar que de fato a expressão booleana original e a expressão booleana obtida pela sintetização são correspondentes.

II. Exercício 2 - Tipos de Dados

a) Considere a implementação de uma ROM (read-only memory) baseada em um array 1Dx1D tipo CONSTANT. A ROM pode ser organizada como um array de oito palavras de 4 bits cada. Crie um array chamado rom e depois defina um CONSTANT do tipo rom.

b) Escolha os valores armazenados na ROM da seguinte forma. O primeiro valor é a representação binaria do último número da sua matrícula e para cada um dos seguintes valores incremente em ‘1’ o valor anterior. Declare esses valores usando a sua diretiva CONSTANT, ou seja, “CONSTANT my_rom: rom := (valores);”

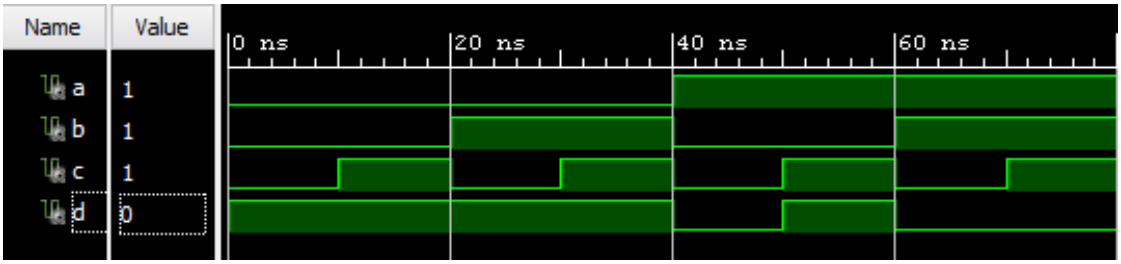

Figura 2. Resultado em formas de onda da simulação do circuito combinacional do exercício 1.

A. *Respostas*

a) Para implementar a ROM baseada em um array 1Dx1D que possui 8 palavras de 4 bits cada foi implementado em VHDL um tipo de dado array que foi chamado de rom e depois definido uma constante desse novo tipo de dado através das linhas de código:

```
type rom_type is array (0 to 7) of
   std_logic_vector (3 downto 0);
constant rom : rom_type := <valores>;
```

A ROM implementada recebe um sinal de endereço de 3 bits para especificar qual das 8 (2^3) palavras deve ser escrita no sinal de saída de 4 bits.

b) Para declarar os valores presentes na ROM utilizou se a seguintes linhas de código que definem e declaram a constante utilizada como ROM:

```
constant rom : rom_type := (
               0 => "0011",
               1 => "0100",
               2 => "0101",
               3 => "0110",
               4 => "0111",
               5 => "1000",
               6 => "1001",
               7 => "1010");
```

III. Exercício 3 - Portas Lógicas

a) Criar um arquivo VHDL do CI 4-input AND-OR-INVERT (datasheet SN74LS55, Motorola).

b) Sintetizar o circuito e analisar o reporte de síntese. Quantas LUTs foram usadas? Quantos pinos de IO?

c) Criar um arquivo de testbench e realizar a simulação comportamental do circuito. Apresentar um printscreen da simulação.

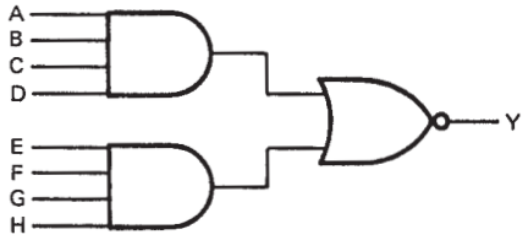

Figura 5. Circuito lógico implementado pelo CI SN74LS55 utilizado no exercício 3. (Diagrama obtido do datasheet)

A. *Respostas*

a) O circuito da Figura 5 foi implementado utilizando 8 entradas de 1 bit e uma saida de 1 bit. O comportamento desse circuito é expresso em VHDL por:

```
y <= not ((a and b and c and d) or (e and
    f and g and h));
```

b) Ao sintetizar o código em VHDL desenvolvido foi obtido o circuito da Figura 6 que possui 9 pinos de I/O (8 de entrada e 1 de saída), conforme o esperado, e utiliza duas LUTs, de 4 entradas e 5 entradas.

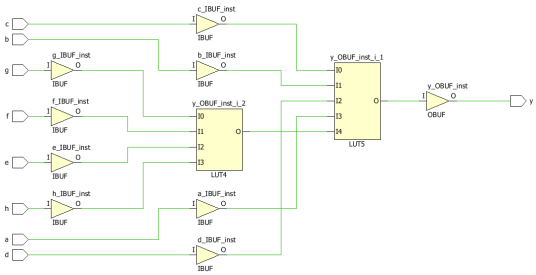

Figura 6. Diagrama do circuito sintetizado no Vivado utilizando o código do exercício 3.)

c) Os resultados obtidos pela simulação comportamental está presente na Figura 7.

IV. Exercício 4 - Portas Lógicas

a) Criar um arquivo VHDL do CI 3-input and 2-input AND-OR-INVERT gates (datasheet SN74LS51, Texas Instruments).

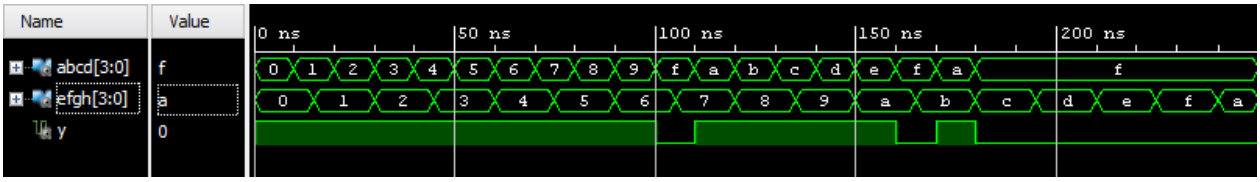

Figura 7. Resultados da simulação do circuito sintetizado no exercício 3.

b) Sintetizar o circuito e analisar o reporte de síntese. Quantas LUTs foram usadas? Quantos pinos de IO?

c) Criar um arquivo de testbench e realizar a simulação comportamental do circuito. Apresentar um printscreen da simulação.

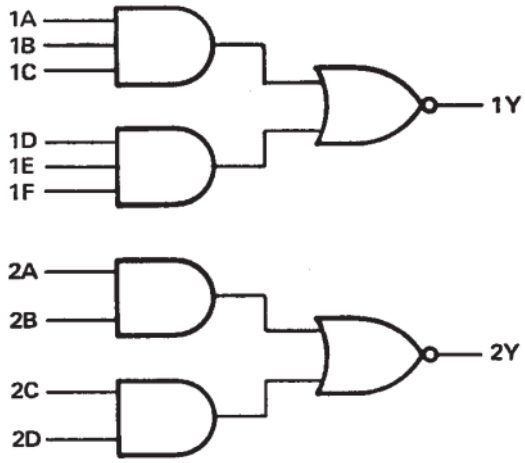

Figura 8. Circuito lógico implementado pelo CI SN74LS51 utilizado no exercício 4 (Diagrama obtido do datasheet).

A. *Respostas*

a) O circuito da Figura 8 foi implementado utilizando 10 entradas de 1 bit e duas saida de 1 bit. O comportamento desse circuito é expresso em VHDL por:

```
y1 <= not ((a1 and b1 and c1) or (d1 and
    e1 and f1));
y2 <= not ((a2 and b2) or (c2 and d2));
```

b) Ao sintetizar o código em VHDL desenvolvido foi obtido o circuito da Figura 9 que possui 12 pinos de I/O (10 de entrada e 2 de saída), conforme o esperado, e utiliza duas LUTs, de 6 entradas e 4 entradas.

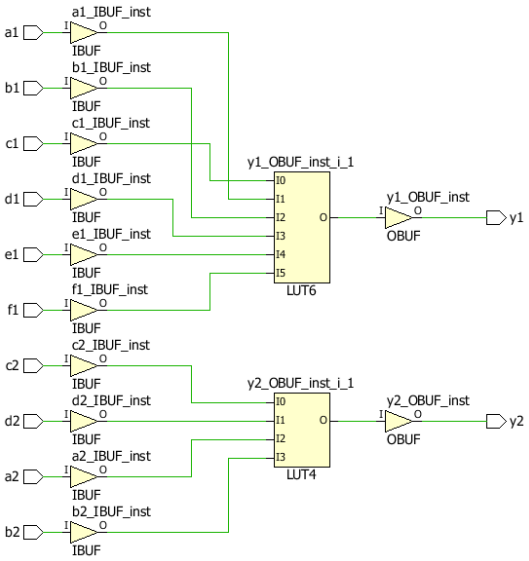

Figura 9. Diagrama do circuito sintetizado no Vivado utilizando o código do exercício 4.)

c) O resultado obtido pela simulação comportamental está presente na Figura 10.

V. Exercício 5 - Projeto circuito greater-than

O circuito greater-than (maior que) compara duas entradas (a,b) e envia a saída para '1' lógico quando a é maior que b. Deseja-se criar um circuito greater-than de 4-bits usando uam metodologia bottom-up usando unicamente operadores lógicos. Projete o circuito como segue:

a) Apresente a tabela verdade de um circuito greater-than de 2-bits e a respectiva expressão no formato de soma de produtos.

b) Baseado nessa expressão, obtenha e apresente o código VHDL.

c) Realize uma simulação e verifique o correto funcionamento do circuito greater-than de 2-bits. Apresente o arquivo testbench e um print de simulação.

d) Sintetize o circuito e apresente uma tabela com consumo de recursos (LUTs, Flip-flops, DSPs, BRAMs)

e) Use circuitos greater-than de 2-bits e comparadores de igualdade de 2-bits e construa um circuito greater-than de 4 bits. Apresente um diagrama de blocos da estrutura.

f) Realize uma simulação e verifique o correto funcionamento do circuito greater-than de 4-bits. Apresente o arquivo testbench e um print de simulação.

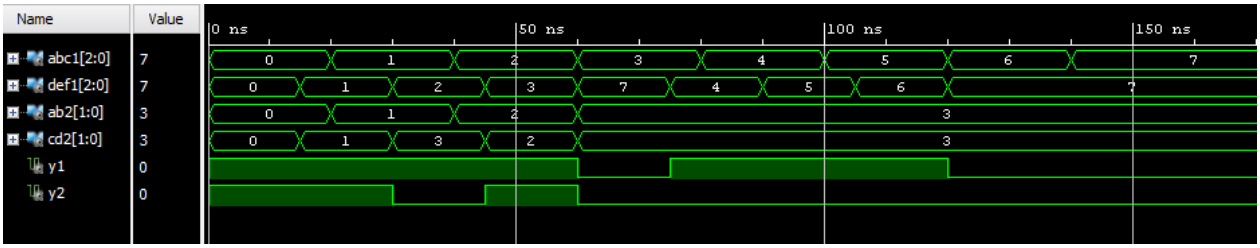

Figura 10. Resultados da simulação do circuito sintetizado no exercício 4.

g) Sintetize o circuito integrado e apresente uma tabela com consumo de recursos (LUTs, Flip-flops, DSPs, BRAMs)

A. *Respostas*

a) Para um circuito *greater-than* de 2 entradas de 2 bits, (a_1a_0) e (b_1b_0), temos a tabela presente na Tabela I. A expressão booleana que representa essa tabela verdade por soma de produtos é $z = \overline{a_1}a_0\overline{b_1}\overline{b_0} + a_1\overline{a_0}\overline{b_1}\overline{b_0} + a_1\overline{a_0}\overline{b_1}b_0 + a_1a_0\overline{b_1}\overline{b_0} + a_1a_0\overline{b_1}b_0 + a_1a_0b_1\overline{b_0}$. Para tornar a expressão mais legível utiliza-se o Mapa de Karnaught para minimizar a expressão lógica para a equação equivalente $z = a_1\overline{b_1} + a_1a_0\overline{b_0} + a_0\overline{b_1}\overline{b_0}$.

a_1	a_0	b_1	b_0	z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Tabela I. TABELA VERDADE PARA UM CIRCUITO *greater-than* DE 2 BITS.

b) Com a expressão minimizada implementou-se em VHDL um ENTITY de 2 entradas de 2 bits e 1 saída de 1 bit cujo comportamento é expresso pelo seguinte código em VHDL:

```
z <= (a(1) and (not b(1))) or ((a(1)) and
    (a(0)) and (not b(0))) or ((a(0)) and
    (not b(1)) and (not b(0)));
```

c) O resultado obtido pela simulação comportamental do *greater-than* de 2 bits está presente na Figura 11.

d) Após sintetizar e obter o relatório de recursos utilizados é possível observar que essa implementação de um circuito de *greater-than* de 2 bits utiliza apenas 1 LUT, ou seja, ¡¡ 1% dos recursos da FPGA utilizada na disciplina. É possível visualizar os recursos utilizados na Figura 12.

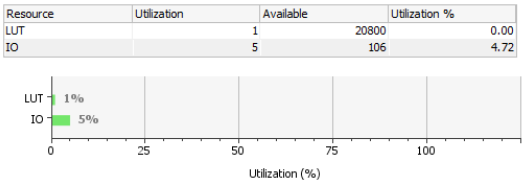

Figura 12. Relatório de uso de recursos após a sintetização no Vivado).

d) O diagrama do diagrama de blocos do comparador *greater-than* de 4 bits pode ser visualizada na Figura 13.

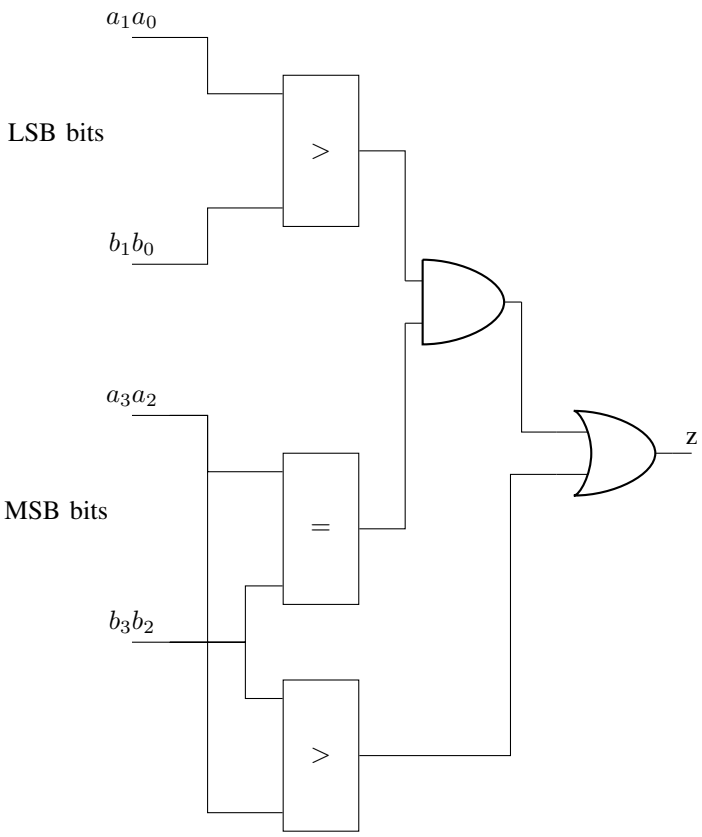

Figura 13. Diagrama de blocos do *greater-than* de 4 bits implementado utilizando 2 *greater-than* de 2 bits e 1 *comparador* de 2 bits.

f) O resultado obtido pela simulação comportamental do

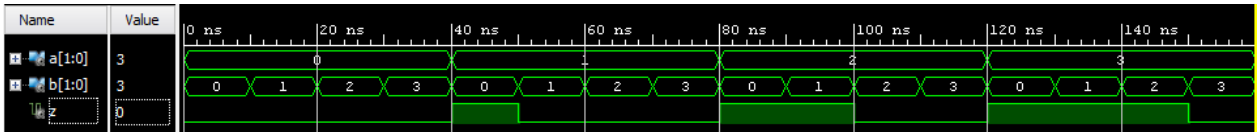

Figura 11. Resultados da simulação do circuito sintetizado no exercício 5 item c.

greater-than de 4 bits está presente na Figura 14.

g) Após sintetizar o código em VHDL da implementação do circuito *greater-than* o Vivado apresenta um relatório dos recursos utilizados na placa de desenvolvimento, Figura 15. Nesse relatório é possível ver que o circuito implementado utiliza duas LUTs, de 5 e 4 entradas, respectivamente, e 9 pinos de I/O. Isso equivale a <%1 do número de LUTs disponíveis e 8% do número de pinos de entradas e saídas.

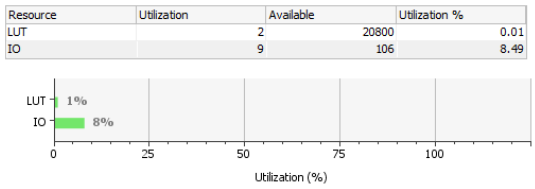

Figura 15. Relatório de uso de recursos após a sintetização no Vivado da implementação do circuito *greater-than* de 4 bits.

VI. Exercício 6 - Barrel-shifter multifuncional

Considere um circuito barrel-shifter de 8-bits que pode realizar rotações à direita e à esquerda. O circuito possui uma entrada de controle de 1-bit, chamada lr, especifique a seguinte solução.

a) Projete uma arquietura de hardware usando um barrel-shifter à direita, um barrel-shifter à esquerda e um multiplexador de 2-to-1 para selecionar o resultado desejado. Apresente o código VHDL.

b) Realize uma simulação e verifique o correto funcionamento do circuito. Apresente o arquivo testbench e um print de simulação.

c) Sintetize o circuito e apresente uma tabela com consumo de recursos (LUTs, Flip-flops, DSPs, BRAMs).

A. Respostas

a) A implementação do *barrel-shifter* utiliza a arquitetura presente na Figura 16. O projeto em VHDL utiliza um *barrel-shifter* para a direita cujo comportamento e descrito pelas linhas de código:

```
output(7) <= data(0);
output(6 downto 0) <= data(7 downto 1);
```

E um *barrel-shifter* para a esquerda com o comportamento descrito pelo codigo-fonte:

```
output(0) <= data(7);
output(7 downto 1) <= data(6 downto 0);
```

Ambos os *barrel-shifter* são conectados a um multiplexador de 2-1 de 8 bits e a saída do circuito é controlado pelo sinal lr que controla se o circuito efetua *shift* para a esquerda ou direita.

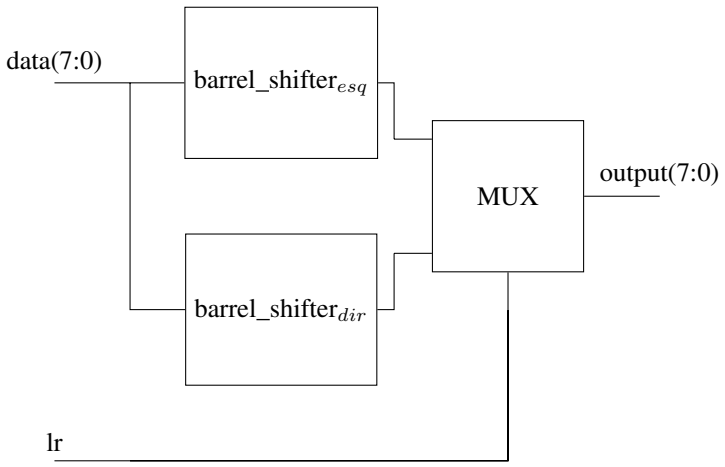

Figura 16. Diagrama do *barrel-shifter* multifuncional implementado no exercício 6 item a.

b) O resultado obtido pela simulação comportamental do *barrel-shifter* multifuncional de 8 bits está presente na Figura 17.

c) Após sintetizar o código em VHDL da implementação do circuito do *barrel-shifter* multifuncinal é possível obter através do Vivado o relatório dos recursos utilizados do FPGA, Figura 18. Nesse relatório é possível ver que o circuito implementado utiliza duas LUTs, de 8 e 3 entradas (4 LUTs do FPGA), respectivamente, e 17 pinos de I/O. Isso equivale a <%1 do número de LUTs disponíveis e 16% do número de pinos de entradas e saídas.

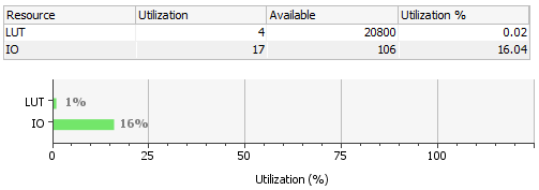

Figura 18. Relatório de uso de recursos após a sintetização no Vivado da implementação do circuito do *barrel-shifter* multifuncinal.

VII. Exercício 7 - Comparação de arquiteturas

a) Implemente em VHDL os circuitos somadores de 4 bits mostrados na seguitne figura. Use entradas de 4-bits. Use as

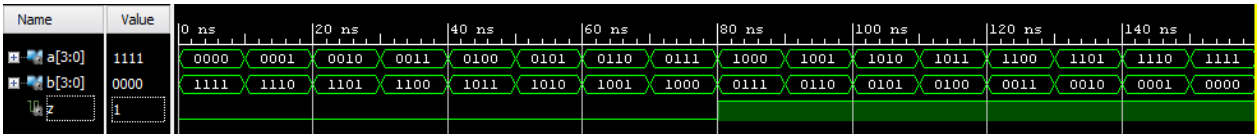

Figura 14. Resultados da simulação do circuito sintetizado no exercício 5 item f.

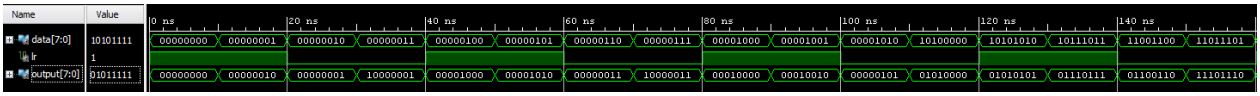

Figura 17. Resultados da simulação do circuito sintetizado no exercício 6 item b.

bibliotecas std_logic_unsigned.all e std_logic_arith.all.

b) Realize uma simulação e verifique o correto funcionamento do circuito. Apresente o arquivo testbench e um print de simulação

c) Sintetize os circuitos e apresente uma tabela de consumo de recursos entre das. Qual arquitetura é mais ótima?

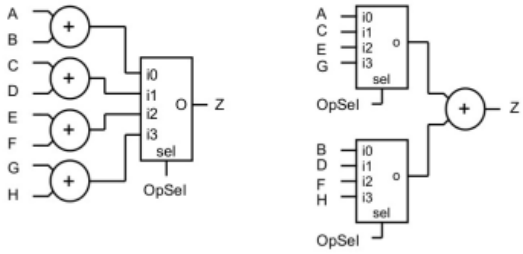

Figura 19. Diagramas das arquiteturas de somadores de 4 bits do Exercício 7. Na esquerda há uma representação da arquitetura 1 e na direita está uma representação da arquitetura 2.

A. *Respostas*

a) A implementação dos dois somadores de 4 bits estão presentes no arquivo VHDL *ex7_somador_1.vhd* e *ex7_somador_2.vhd*.

b) Os resultados das duas simulações comportamentais estão presentes nas Figuras 20 e 21, respectivamente, para as arquiteturas do somador 1 e do somador 2. Pela simulação é possível conferir que ambas as metodologias apresentam os mesmos resultados e que ambos resultados estão corretos.

c) Ao sintetizar ambos as arquiteturas percebemos que apesar da diferença da implementação os circuitos sintetizados utilizam os mesmos recursos da placa de desenvolvimento os circuitos sintetizados para as arquiteturas 1 e 2 estão presentes na Figura 22 e na Figura 23, respectivamente.

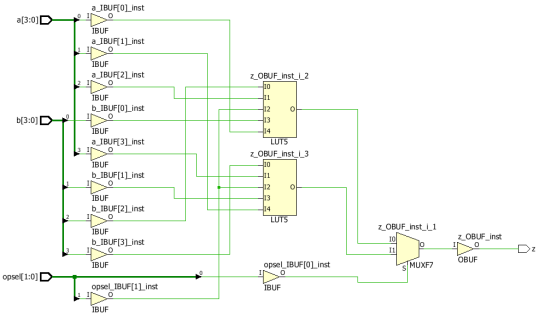

Figura 22. Circuito sintetizado pelo vivado utilizando a arquitetura 1 para o somador do exercicio 7 item c.

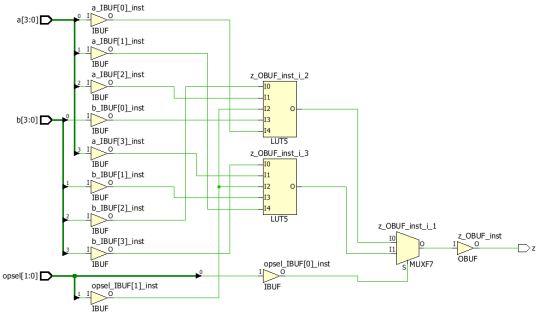

Figura 23. Circuito sintetizado pelo vivado utilizando a arquitetura 2 para o somador do exercicio 7 item c.

REFERÊNCIAS

[1] V. A. Pedroni, *Circuit design with VHDL*. MIT press, 2004.

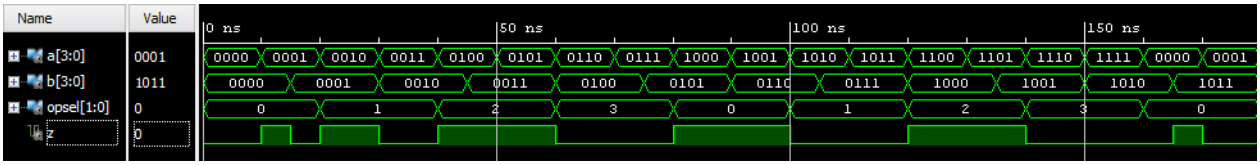

Figura 20. Resultados da simulação do circuito sintetizado no exercício 7 item b utilizando a arquitetura 1 do somador.

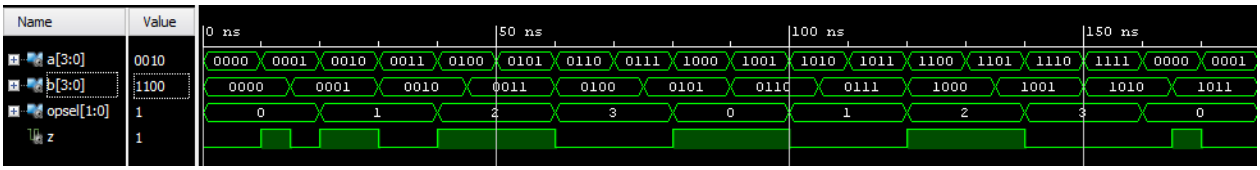

Figura 21. Resultados da simulação do circuito sintetizado no exercício 7 item b utilizando a arquitetura 2 do somador.