



Universidade Federal de Pelotas
Bacharelado em Ciência da Computação
Técnicas Digitais

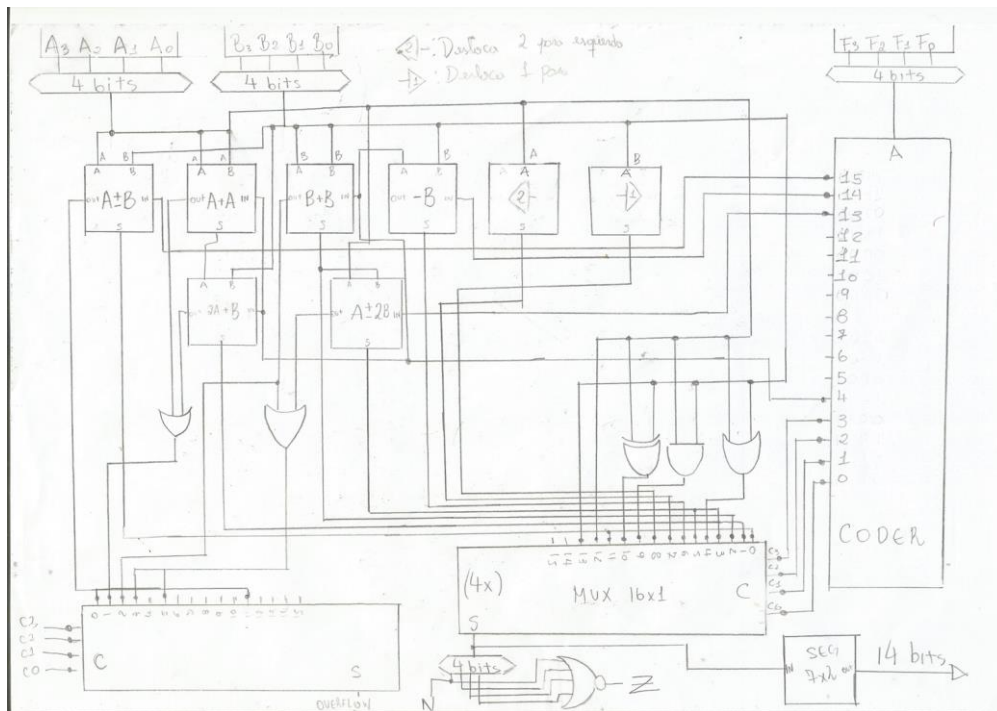
RELATÓRIO ULA

Juan Burtet
Thales Castro

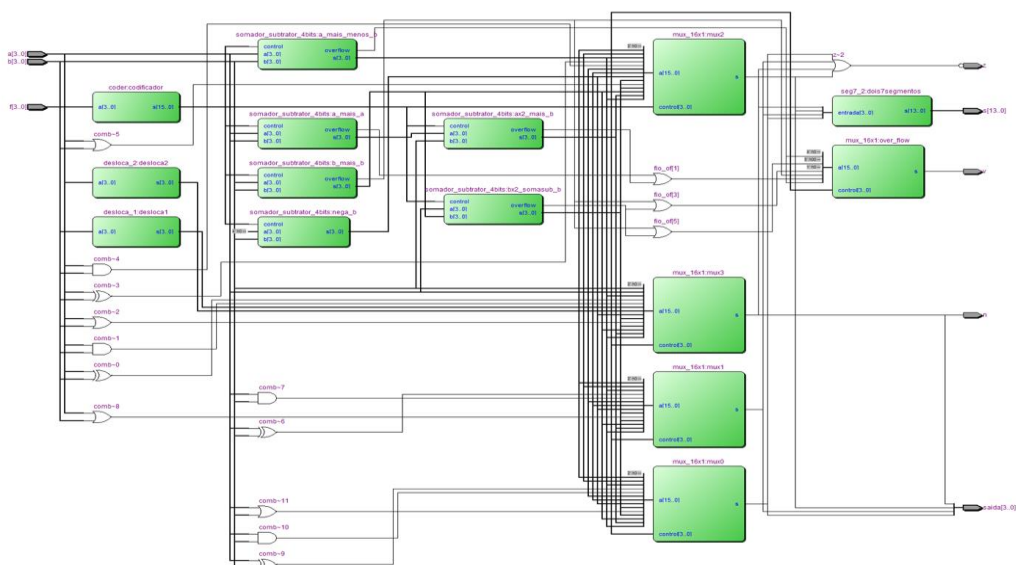
2016/2

1. Projeto Arquitetural

O desenho abaixo representa o projeto arquitetural de uma Unidade Lógica e Aritmética (ULA) de 4 bits. Com exceção do fio de overflow, que é um bit, todos os barramentos são em 4 bits. Há quatro multiplexadores (MUX) de 16x1 que geram as saídas, selecionadas pelo codificador. Os flags são acionados pelo caminho que leva as saídas aos dois displays de sete segmentos.



Abaixo está o desenho do circuito no RTL Viewer do próprio Quartus II.



2. Resultados da Simulação

Para esta ULA, decidiu-se pela tabela de operação a seguir.

Código de Entrada	Palavra de Controle	Saída
"0000"	"0000000000000000"	$A + B$
"0001"	"0000000000000001"	$2A + B$
"0010"	"0000000000000010"	$B * 2$
"0011"	"0010000000000011"	$A - 2B$
"0100"	"0000000000000100"	$A \text{ OR } B$
"0101"	"0000000000000101"	$A + B + B$
"0110"	"0100000000000110"	$!B + 1$
"0111"	"0000000000000111"	2 Bits à Esquerda
"1000"	"0000000000001000"	1 Bit à Direita
"1001"	"0000000000001001"	$A \text{ AND } B$
"1010"	"0000000000001010"	$A \text{ XOR } B$
"1011"	"1000000000001011"	$A - B$
"1100"	"0000000000001100"	A
"1101"	"0000000000001101"	B
"1110"	"0000000000001110"	0
"1111"	"0000000000001111"	0

Os quatro bits menos significativos foram usados para comandar os MUXs que escolhem a saída e o overflow. Os três bits mais significativos receberam a utilidade de ativar a subtração em um somador/subtrator de quatro bits:

- S(15): mudança entre as operações "0000" ($A + B$) e "1011" ($A - B$);
- S(14): ativação da operação "0110" com o complemento de dois da entrada B;
- S(13): mudança entre as operações "0011" ($A - 2B$) e "0101" ($A + B + B$).

Outros componentes que fazem parte do sistema:

- Codificador 4x16;
- Deslocador de 1 bit para direita;
- Deslocador de 2 bits para esquerda;
- Somador/Subtrator de quatro bits;
- MUX 16x1;
- Componente que leva a saída de quatro bits para ser apresentado em 2 displays de 7 segmentos.

Cabe frisar que o sistema faz todas as operações. Contudo, os quatro bits finais do codificador selecionaram qual delas será apresentada na saída.

3. Verificação com forma de onda

É mostrado a seguir o comportamento do sistema em todas as operações usando valores fixos na entrada de A (0101) e B (0010). Foram adicionadas as saídas de quatro bits para melhor visualização dos resultados.

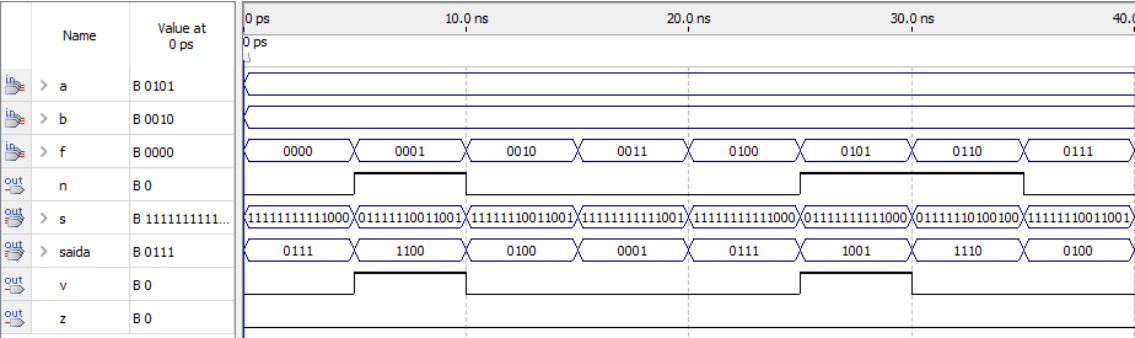


Tabela com a representação dos valores das formas de onda:

Controle	Saída	Negativo	Zero	Overflow
“0000”	“0111”	0	0	0
“0001”	“1100”	1	0	1
“0010”	“0100”	0	0	0
“0011”	“0001”	0	0	0
“0100”	“0111”	0	0	0
“0101”	“1001”	1	0	1
“0110”	“1110”	1	0	0
“0111”	“0100”	0	0	0

Restante das formas de onda:

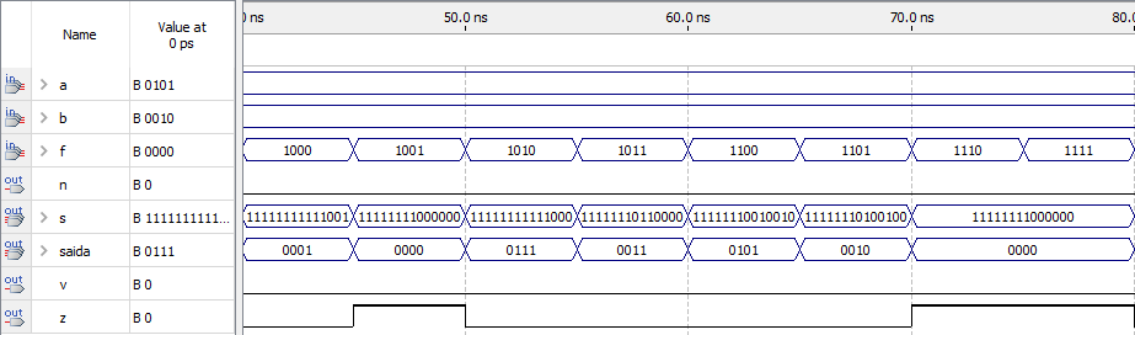


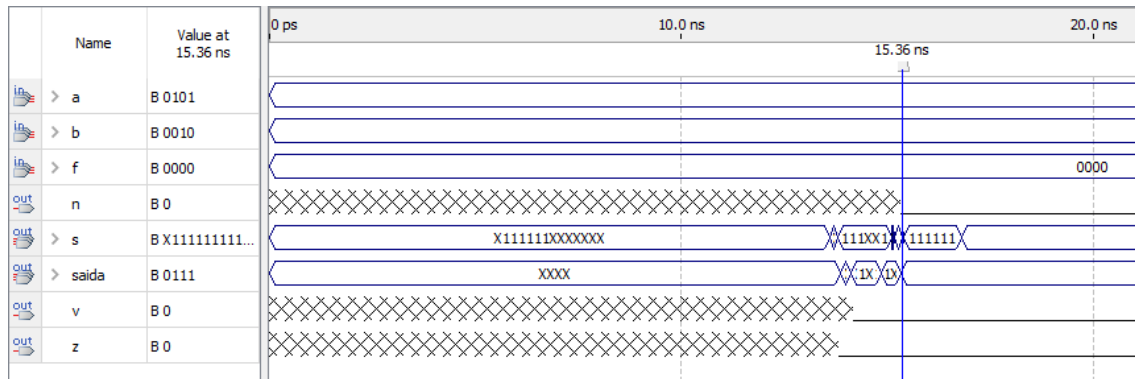
Tabela com o restante das formas de onda:

Controle	Saída	Negativo	Zero	Overflow
“1000”	“0001”	0	0	0
“1001”	“0000”	0	1	0
“1010”	“0111”	0	0	0
“1011”	“0011”	0	0	0
“1100”	“0101”	0	0	0
“1101”	“0010”	0	0	0
“1110”	“0000”	0	1	0
“1111”		0	1	0

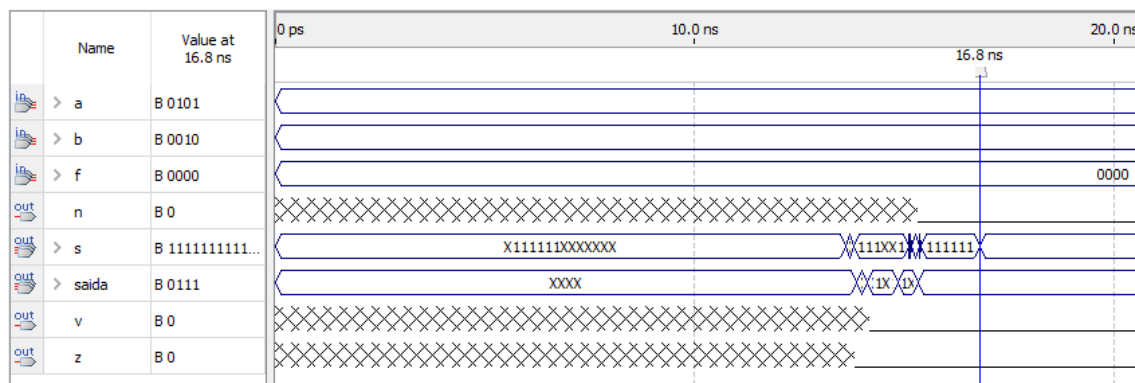
4. Resultados de Síntese

Frequência Máxima de Operação: feito um teste com atrasos das mesmas formas de onda anteriores, notou-se que o maior atraso encontrado foi na operação “0000” (A + B), tanto nas saídas de quatro bits, quanto nas saídas para os displays.

Saída de quatro bits: atraso de 15.36 ns.



Saída nos displays: atraso de 16.8 ns.



Cálculo da frequência ($f = 1/T$):

- $T = 15.36\text{ns} \implies f = 65,1\text{ MHz}$
- $T = 16.8\text{ns} \implies f = 59,5\text{ MHz}$