Relatório do projeto da ULA

Organização e Arquitetura de Computadores – Turma C – 116394

Professor: Marcelo Grandi Mandelli

Responsáveis:

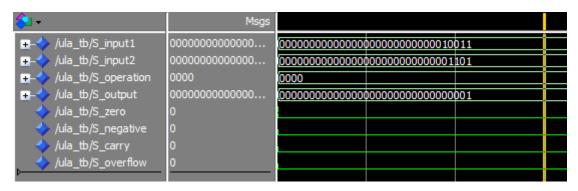
 Danillo Neves
 14/0135839

 Luiz Gustavo
 13/0123293

 Rodrigo Guimarães
 14/0170740

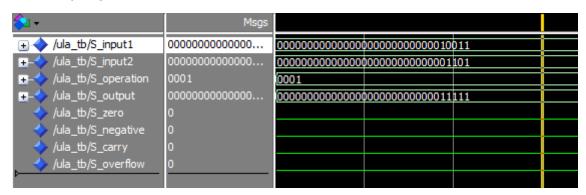
Abaixo estão representados os testes realizados nas simulações da ULA projetada:

• Operação AND (0000):



Acima é realizada a operação lógica AND gerando o output observado na imagem, e mantendo inativas as saídas zero, "negative", "carry" e "overflow".

• Operação OR (0001):



Acima é realizada a operação lógica OR gerando o output observado na imagem, e mantendo inativas as saídas zero, "negative", "carry" e "overflow".

Operação ADD (0010):

1° caso) ADD com output positivo:

4 •	Msgs				
→ /ula_tb/S_input1	0000000000000	0000000000000000	000000000000000000000000000000000000000	11	
+- /ula_tb/S_input2	0000000000000	000000000000000000000000000000000000000	0000000000000001	101	
	0010	0010			
 → /ula_tb/S_output	0000000000000	000000000000000000000000000000000000000	0000000000001000	000	
/ula_tb/S_zero	0				
/ula_tb/S_negative	0				
/ula_tb/S_carry	0				
/ula_tb/S_overflow	0				

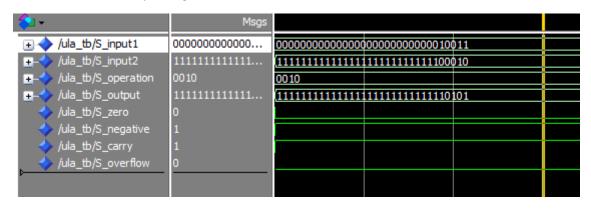
Input 1 (19 em decimal) + Input 2 (13 em decimal) = Output (32 em decimal). Observe que as saídas zero, "negative", "carry" e "overflow" estão inativas.

2° caso) ADD com output zero:



Input 1 (19 em decimal) + Input 2 (-19 em decimal) = Output (0 em decimal). Observe que a saída zero está ativa, indicando que output é nulo. Não ocorre número negativo, "carry" ou "overflow" nesse caso.

3° caso) ADD com output negativo:



Input 1 (19 em decimal) + Input 2 (-30 em decimal) = Output (-11 em decimal). Observe que a saída "negative" está ativa, indicando que output é negativo. Além disso, observamos que também ocorre "carry" nessa soma, ativando a saída "carry". Não ocorre output zero, ou "overflow" nesse caso.

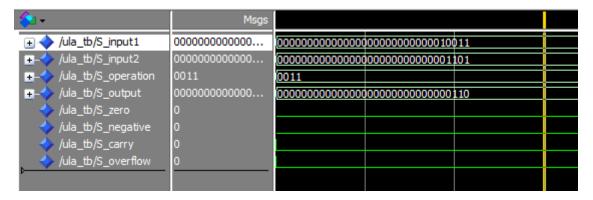
4° caso) ADD com "overflow":

4 •	Msgs				
→ /ula_tb/S_input1	11000000000000	11000000000000000	000000000000000000000000000000000000000	000	
+- /ula_tb/S_input2	1011000000000	10110000000000000	000000000000000000000000000000000000000	000	
	0010	0010			
	01110000000000	01110000000000000	000000000000000000000000000000000000000	000	
/ula_tb/S_zero	0				
/ula_tb/S_negative	0				
/ula_tb/S_carry	1				
/ula_tb/S_overflow	1				

Input 1 (-1073741824 em decimal) + Input 2 (-1342177280 em decimal) = Output (1879048192 em decimal). Observe que a saída "overflow" está ativa devido ao fato de somarmos dois números negativos e obtermos um número positivo. Além disso, observamos que também ocorre "carry" nessa soma, ativando a saída "carry". Não ocorre output zero, ou negativo nesse caso.

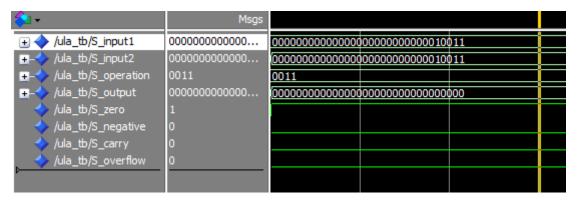
Operação SUB (0011):

1° caso) SUB com output positivo:



Input 1 (19 em decimal) - Input 2 (13 em decimal) = Output (6 em decimal). Observe que as saídas zero, "negative", "carry" e "overflow" estão inativas.

2° caso) SUB com output zero:



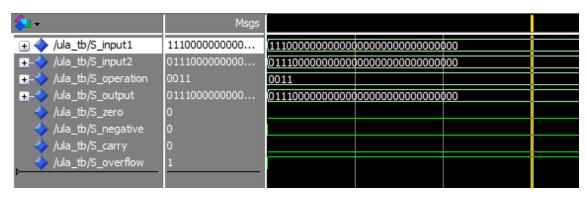
Input 1 (19 em decimal) - Input 2 (19 em decimal) = Output (0 em decimal). Observe que a saída zero está ativa, indicando que output é nulo. Não ocorre número negativo, "carry" ou "overflow" nesse caso.

3° caso) SUB com output negativo:

≨ 1 +	Msgs				
+> /ula_tb/S_input1	00000000000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000	11	
	0000000000000	00000000000000000	000000000000011	110	
	0011	0011			
	111111111111111	1111111111111111	111111111111111	101	
/ula_tb/S_zero	0				
/ula_tb/S_negative	1				
/ula_tb/S_carry	0				
/ula_tb/S_overflow	0				

Input 1 (19 em decimal) - Input 2 (30 em decimal) = Output (-11 em decimal). Observe que a saída "negative" está ativa, indicando que output é negativo. Não ocorre output zero, "carry" ou "overflow" nesse caso.

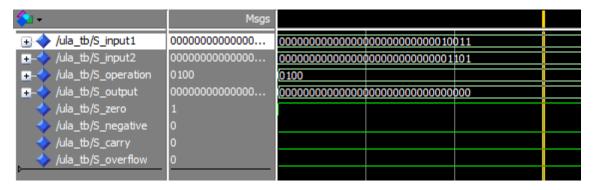
4° caso) SUB com "overflow":



Input 1 (-536870912 em decimal) - Input 2 (1879048192 em decimal) = Output (1879048192 em decimal). Observe que a saída "overflow" está ativa devido ao fato de subtrairmos um número positivo de um negativo e obtermos um número positivo. Não ocorre output zero ou negativo, nem "carry" nesse caso.

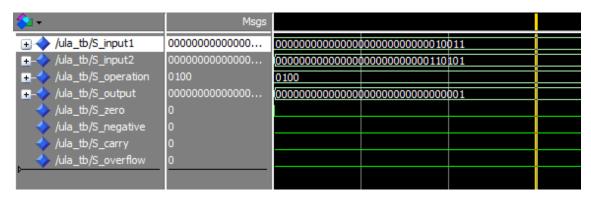
• Operação SLT (0100):

1° caso) SLT não válido setando Output com zero:



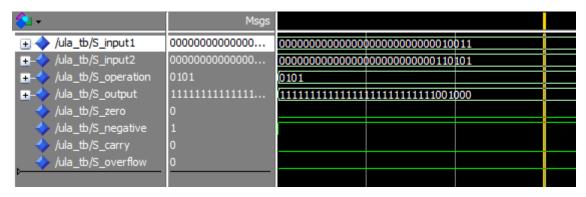
Acima verificamos se Input 1 (19 em decimal) < Input 2 (13 em decimal). Como a inequação não é válida, output é setado com zero.

2° caso) SLT válido setando Output com um:



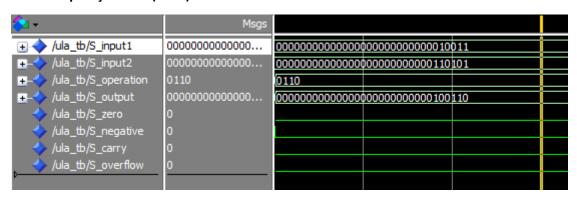
Acima verificamos se Input 1 (19 em decimal) < Input 2 (53 em decimal). Como a inequação é válida, output é setado com um.

Operação NOR (0101):



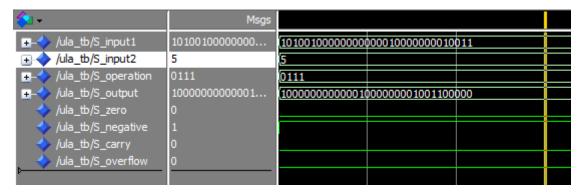
Acima é realizada a operação lógica NOR gerando o output observado na imagem. Observe que a saída "negative" está ativa pois o output gerado é negativo.

• Operação XOR (0110):



Acima é realizada a operação lógica XOR gerando o output observado na imagem. O output gerado é positivo mantendo assim as saídas zero e ""negative" inativas.

• Operação SLL (0111):



Acima executamos "shift left" lógico do Input 1 em 5 bits (valor de Input 2). Observe que o output gerado é negativo, ativando a saída zero.

Operação SRL (1000):



Acima executamos o "shift right" lógico do Input 1 em 5 bits (valor de Input 2). Não é gerado output zero ou negativo nesse caso da imagem.

Operação SRA (1001):



Acima executamos o "shift right" aritmético (com extensão de sinal) do Input 1 em 5 bits (valor de Input 2). Observe que o output gerado mantém o mesmo sinal negativo do Input 1, ativando a saída ""negative"".

Programas Utilizados para gerar o código e simulação:

- Quartus II 13.0sp1 (64-bit) Web Edition;
- ModelSim-Altera.