Projecto de Sistemas Digitais

Laboratório 1

Unidade Lógico-Aritmética Simples

Rafael Gonçalves, 73786 Gonçalo Ribeiro, 73294

17 de Outubro de 2014

Conteúdo

1 Arquitectura **2**
1.1 Descrição 2
1.2 Esquemas 3

2 Datapath **4**
2.1 Descrição 4
2.2 Registos 4
2.3 Unidade Lógico-Aritmética 4
2.3.1 Somador 4
2.3.2 Multiplicador 4
2.3.3 Shift Right Aritmético 4
2.3.4 Função Lógica NOR 4
2.3.5 Simulação 5

3 Unidade de Controlo **6**
3.1 Descrição 6
3.2 Máquina de Estados 6
3.3 Sinais de Controlo 6
3.4 Simulação 7

4 Simulação do Circuito **8**

1 Arquitectura

1.1 Descrição

A arquitectura que decidimos implementar divide-se em duas secções distintas, como na maioria das arquitecturas dedicadas:

- Uma Unidade de Controlo, mais especificamente uma Máquina de Estados
- Uma Datapath, cuja especificação corresponde à disponibilizada no enunciado (ver Figura 1)

De modo a melhor fazer a interface com os elementos da placa de desenvolvimento, as várias componentes do sistema foram projectados com vista a fazer a interface com os botões, interruptores e displays da Basys.

Uma vez que da Datapath saem os valores dos registos, que posteriormente terão de ser seleccionados para ir para os displays de sete segmentos, é necessário um multiplexer no exterior da mesma, cujo sinal de selecção provém dos interruptores da placa de desenvolvimento.

1.2 Esquemas

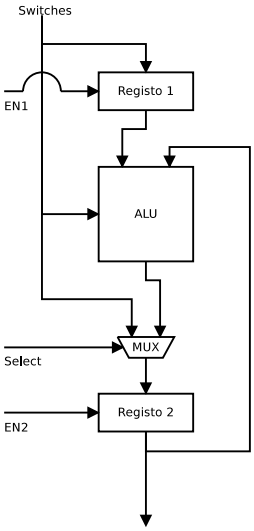

Figura 1: Datapath

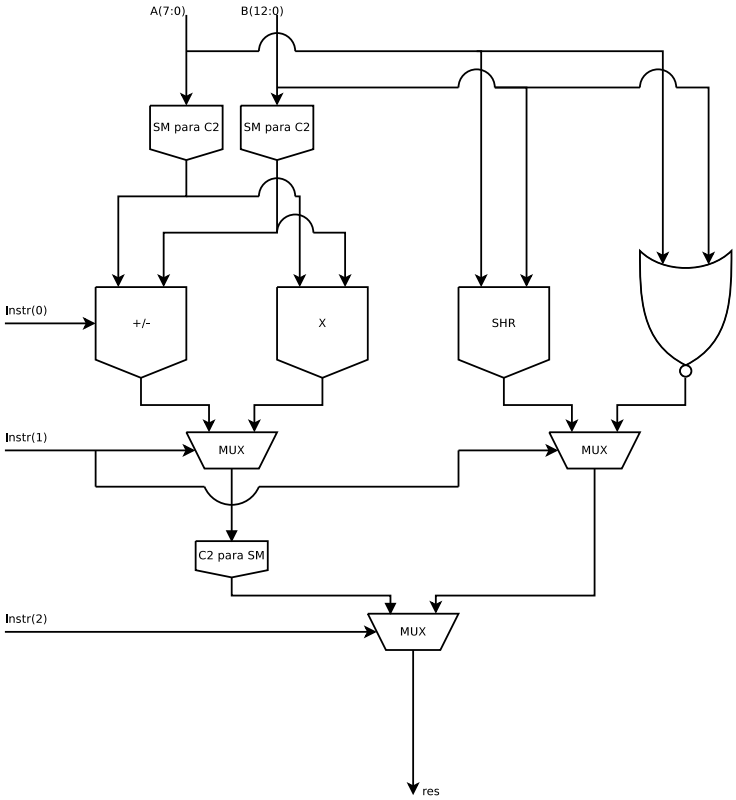

Figura 2: Unidade Lógico-Aritmética

2 Datapath

2.1 Descrição

A decisão mais relevante na datapath foi a codificação de sinal utilizada nos registos e na Unidade Lógico-Aritmética.

De modo a compreender intuitivamente os valores em uso, os displays devem mostrar os valores com uma codificação sinal-módulo. Assim, e uma vez que os valores dos displays originam nos registos da Datapath, decidimos que todo o armazenamento de valores deve ser feito em formato sinal-módulo.

2.2 Registos

Os registos foram especificados através das funcionalidades de abstracção da linguagem VHDL, isto é, definimos uma arquitectura de registo com número de bits arbitrário. O tamanho de cada instância é depois determinado no código de instanciação.

Neste caso, foram instanciados dois registos, um de 7 bits e outro de 13 bits, consoante a arquitectura especificada no enunciado.

2.3 Unidade Lógico-Aritmética

Uma vez que os dois operadores das diversas unidades funcionais que compõem a ULA estão codificados no formato sinal-módulo, é necessário fazer a conversão de sinal-módulo para complemento para dois.

O Somador (que efectua as operações soma e diferença) e o Multiplicador foram incluídos da biblioteca *signed*, pelo que os seus operandos devem ser convertidos em complemento para dois à entrada, e o resultado deve ser convertido para sinal-módulo à saída (ver Figura 2).

2.3.1 Somador

O Somador efectua uma soma ou diferença, consoante os sinais de controlo, convencional, em complemento para dois.

2.3.2 Multiplicador

O multiplicador efectua um produto convencional em complemento para dois. Uma vez que o produto ocupa mais bits que aqueles disponíveis no registo que o deve armazenar, são desprezados vários dos bits mais significativos do resultado.

2.3.3 Shift Right Aritmético

A função shift afecta apenas os bits de módulo do valor armazenado no registo 2. O bit de sinal permanece inalterado.

2.3.4 Função Lógica NOR

A função lógica NOR aplica-se aos bits de sinal dos dois operadores e aos 6 bits menos significativos dos dois operadores. Os restantes bits do segundo operador (de dimensão 13) são negados (isto é, NOR 0).

2.3.5 Simulação

As operações realizadas na simulação da Figura 3 são as indicadas na Tabela 1. É de notar que as operações não comutativas são efectuadas na ordem $Op_2 \cdots Op_1$.

Op_1	Op_2	Operação	Código	Resultado
-63	4	MUL	01X	-252
5	5	ADD	000	10
-5	6	SUB	001	11
5	-3			-8
0bXXX X001	0b0 0000 0000 0111	SHR	10X	0b0 0000 0000 0011
0bXXX X001	0b0 0000 0000 0011			0b0 0000 0000 0001
0bXXX X010	0b0 0000 0000 1111			0b0 0000 0000 0011
0b000 0001	0b0 0000 0000 0010	NOR	11X	0b1 1111 1111 1100
5	127	MUL	01X	635
63	4095			4033
-63	4095			-4033

Tabela 1: Operações da Simulação da Figura 3

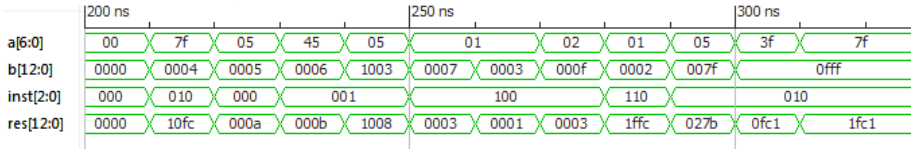

Figura 3: Simulação da Unidade Lógico-Aritmética

3 Unidade de Controlo

3.1 Descrição

A Unidade de Controlo implementa a Máquina de Estados e é o componente responsável por controlar os multiplexers e os enables e sinais de reset dos registos da Datapath. Como input tem os sinais da FPGA que permitem escolher a operação a realizar.

3.2 Máquina de Estados

A Máquina de Estados implementada é a da Figura 4. Tem um estado inicial e um final, dois estados de load e um de operação.

Nos estados *load1* e *load2* o valor à entrada da datapath é armazenado nos registo correspondente. No estado *op* o resultado da ULA é armazendo no registo R2. A operação que a ULA deve realizar é controlada pelo utilizador sob a forma do sinal INST à entrada da datapath.

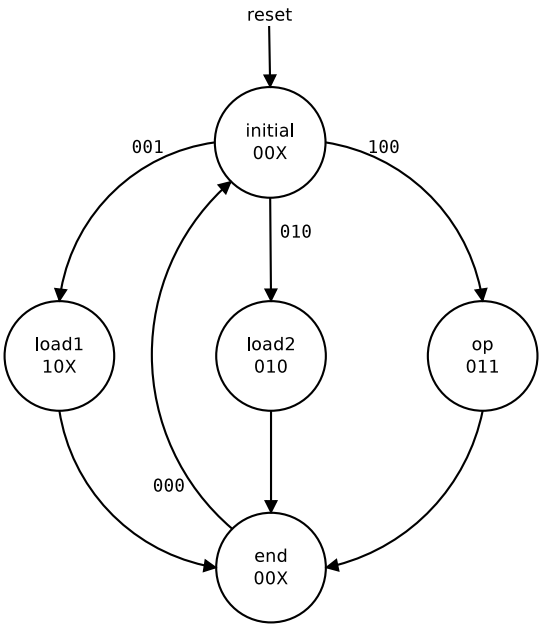

Figura 4: Máquina de Estados

3.3 Sinais de Controlo

Os sinais que entram na Datapath vindos da Unidade de Controlo são três: dois sinais controlam a escrita nos registos R1 e R2; um outro sinal controla o multiplexer que existe antes de R2.

No estado *load1* o enable de R1 é posto a high. No estado *load2* o enable de R2 fica a high e o multiplexer deixa passar o valor à entrada da datapath.

Por fim no estado *op* o enable de R2 é posto a high e o multiplexer deixa passar o sinal à saída da ULA.

3.4 Simulação

A simulação da Unidade de Controlo pode ser vista na Figura 5. Nesta simulação pode verificar-se que a sequência de estados corresponde à descrita na Figura 4 e que os sinais de controlo estão correctos.

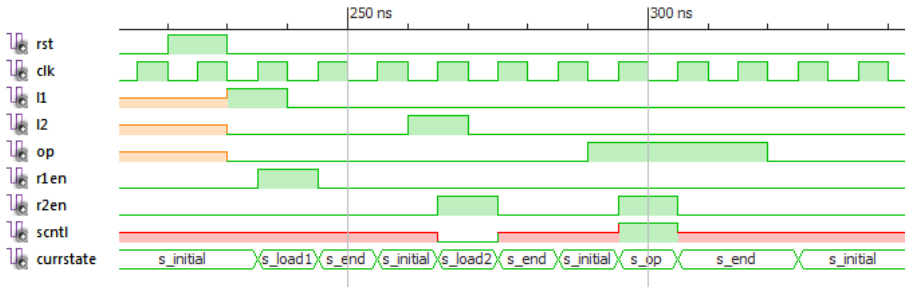

Figura 5: Simulação da Unidade de Controlo

4 Simulação do Circuito

A simulação que verifica o correcto funcionamento dos diversos elementos em conjunção pode ser vista na Figura 6. São realizadas algumas operações de significado trivial, com o único propósito de demonstrar a interligação dos diversos blocos.

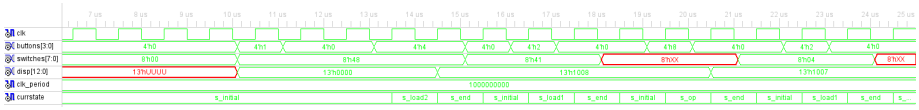

Figura 6: Simulação do Circuito Completo