Unidade Lógico-Aritmética Simples

# Introdução

Neste trabalho tínhamos como objetivo realizar o projeto de uma calculadora lógico-aritmética capaz de executar um grupo de operações simples.

Não sei que por mais aqui

# Especificação Geral do Projeto

A calculadora é compostas por 3 unidades principais a Unidade de Controlo (UC) a Unidade de Dados (UD) e a Unidade de Interface.

A UC é uma máquina de estados (figura 1) que gere o funcionamento geral da calculadora. Tem apenas como entradas os últimos 3 botões de pressão mais à direita da FPGA, utilizados para interagir com o dispositivo, a partir dos mesmos a UC gera o sinal de saída que indica como a unidade de dados se deve comportar. Esta máquina de estados tem 5 estados:

* Idle (INI) – em espera
* Load R1 – guardar entrada no registo R1
* Load R2 – guardar entrada no registo 2
* Operação – executar uma das operações baseando-se nos três últimos interruptores da direita.
* Wait for Release (END) – esperar que o utilizador liberte os botões para evitar operações acidentais

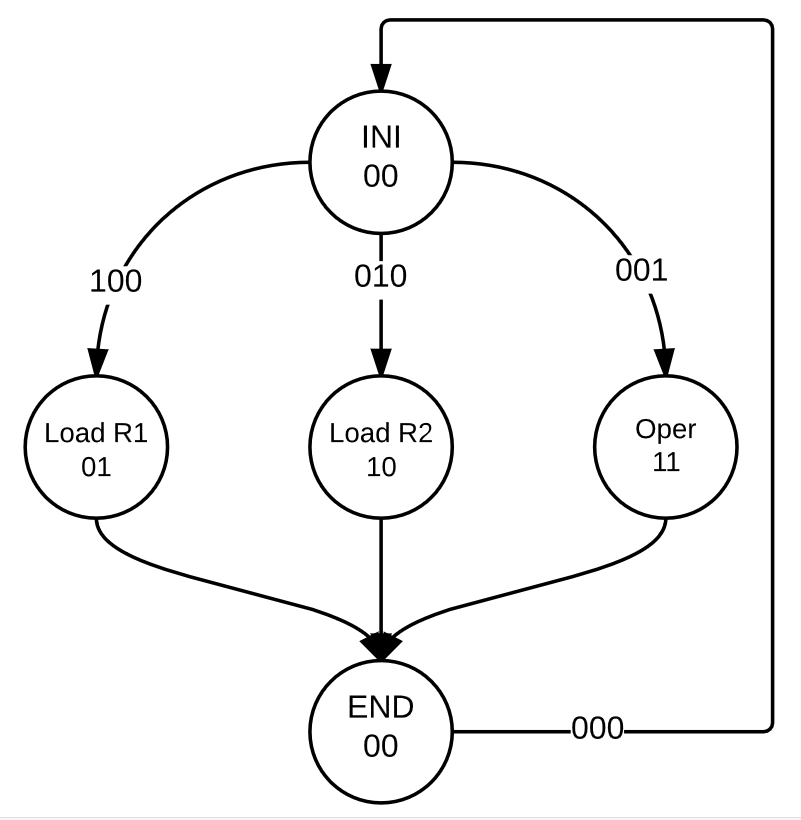


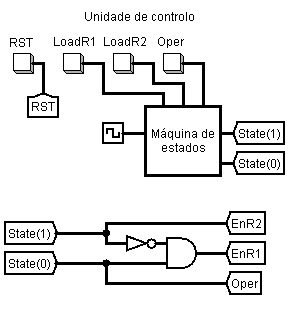
Figura 1: Máquina de estados

Abaixo encontra-se uma tabela com a relação entre os estados e as saídas da unidade de controlo:

|  |  |  |
| --- | --- | --- |
| Saída | | |
| Estado | State1 | State0 |
| Operação | 1 | 1 |
| *Load* R2 | 1 | 0 |
| *Load* R1 | 0 | 1 |
| *INI* | 0 | 0 |
| *END* | 0 | 0 |

Na figura 2 pode ver-se a unidade de controlo (desde os botões de pressão aos bits de controlo da datapath).

figura 2:



A UD é constituída por uma ALU e 2 registos, tem como entradas os 8 interruptores da FPGA, o sinal de reset e os 2 bits de estado fornecidos pela unidade de controlo e como saída o conteúdo do registo R1 ou R2. Esta seleção é feita pelo primeiro interruptor mais à esquerda, ou seja, pelo bit mais significativo do sinal de entrada.

Os restantes bits do sinal de entrada correspondem ao valor que se pretende armazenar num dos registos ou caso o estado de funcionamento indique que se pretende fazer uma operação estes indicam qual a operação a executar segundo a seguinte tabela:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Interruptores | | | | | | | Operação | |
| I6 | I5 | I4 | I3 | I2 | I1 | I0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | Soma | R2 <- R2 + R1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | Subtração | R2 <- R2 - R1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | Multiplicação | R2 <- R2 \* R1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | XOR | R2 <-R2 xor R1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | *Shift-Right* Aritmético | R2 <- R2 sra (R1%8) |
| Outros | | | | | | | Nenhuma |  |

O valor de entrada é convertido de sinal-módulo para complemento para 2 antes de ser armazenado em qualquer dos registos.

O registo R1 armazena apenas o valor da entrada que pode conter um valor dentro do intervalo de [-63; +63], este intervalo de valores pode ser representado com recurso a 7 bits logo essa será a dimensão do registo R1. No caso do R2, este pode armazenar o valor de entrada mas também armazena o resultado das operações, este resultado tem uma resolução máxima dentro do intervalo [-4095; +4095] sendo necessário que este registo seja capaz de armazenar 13 bits.

Na figura 3 é apresentado um esquema geral do circuito da unidade de dados.

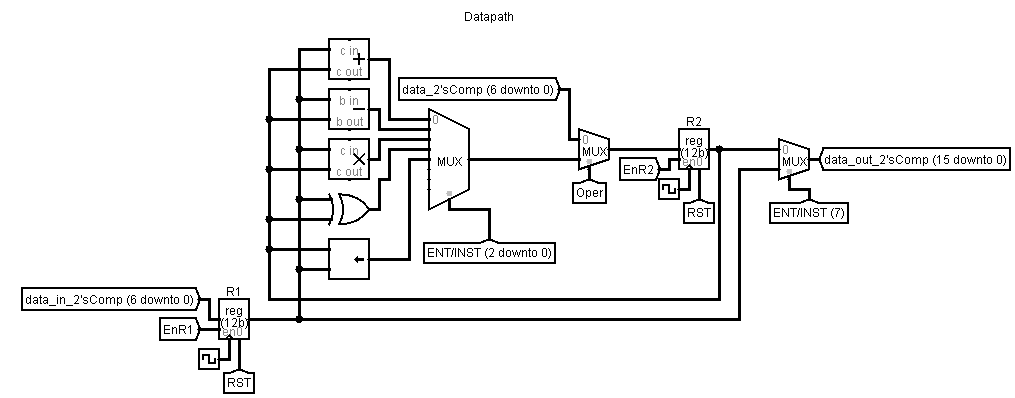


figura 3:

Os dois bits de estado fornecidos pela UC configuram os *enables* dos registos e o bit de seleção do multiplexer M1, de forma a obter a o funcionamento desejado. Quando se pretende guardar a entrada num dos registos o *enable* do respetivo registo deve ser o único ativado e no caso do registo R2 o seletor do multiplexer M1 deve estar colocado a ‘0’ de forma a selecionar o sinal de entrada como o sinal a armazenar no registo. Quando se pretende fazer uma operação o seletor do multiplexer M1 de estar a ‘1’ para selecionar a saída da ALU como entrada do registo R2 e o respetivo *enable* ativado. Esta configuração encontra-se na tabela abaixo:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Descrição | State1 | State0 | *Enable* R1 | *Enable* R2 | Selector M1 |
| Operação | 1 | 1 | 0 | 1 | 1 |
| *Load* R2 | 1 | 0 | 0 | 1 | 0 |
| *Load* R1 | 0 | 1 | 1 | 0 | X |
| *Idle* | 0 | 0 | 0 | 0 | X |

A partir desta tabela da verdade é possível retirar as expressões lógicas para os *enables* e o seletor que requerem menos *hardware*:

A unidade aritmética e lógica (ALU) é a parte central da unidade de dados pois trata-se do bloco responsável pela realização de todas as operações.

Após cada operação é feito um teste ao resultado para verificar se ocorreu *overflow*, caso esta situação se verifique o registo R2 é reinicializado a zero. Este teste é feito no caso da soma e da subtração com recurso a um bit adicional, sendo que o resultado dessas operações um sinal de 14 bits, caso os 2 bits mais significativos sejam diferentes isto significa que ocorreu *overflow*. No caso da multiplicação como é feita entre um sinal de 7 bits e outro de 13 bits o resultado é um sinal de 20 bits. Para testar a ocorrência de overflow verifica-se se os 9 bits mais significativos são todos iguais, isto significa que não ocorreu *overflow* e por isso o resultado deve ser armazenado em R2. O resultado da multiplicação tem sempre de ser truncado a 13 bits antes de ser armazenado.