



Universidade Federal de Minas Gerais  
Departamento de Computação  
Disciplina: Organização de Computadores II  
Professor: Omar Paranaíba Vilela Neto  
Monitores: Luis Pedraza - Laysson Oliveira  
Data: 10/04/2018



### Trabalho Prático I

O trabalho prático consiste em implementar, na linguagem de descrição de hardware Verilog, ao longo de 3 etapas sucessivas, versões de um processador superscalar I2O2 prototipáveis em uma placa FPGA.

A primeira etapa (Trabalho Prático I) de desenvolvimento do trabalho será realizar a versão multiciclo do processador com suporte para as instruções da unidade lógico-aritmética (ULA).

Para isso, deve ser desenvolvido um banco de registradores que contém 16 registradores de tamanho de 16 bits cada um. O módulo deve receber dois endereços de registradores, de entrada, um dado de entrada a ser escrito em algum registrador, um endereço de registrador para determinar onde esse dado deve ser salvo, um sinal de leitura, um sinal de escrita, e dois registradores de saída.

Após isso, deverá ser construída uma unidade lógico-aritmética (ULA), que deve ser conectada ao banco de registradores já construído. Dessa forma, a unidade a ser desenvolvida tem tamanho de “palavra” de 16 bits. Isto é, deve receber como entrada dois operandos de 16 bits e gerar um valor resultante de 16 bits, estando assim de acordo com o banco de registradores.

O conjunto de instruções da ULA a ser desenvolvida é descrito e explicado na tabela abaixo.

CODOP	Instrução	Mnemônico	Operação
0	Add	Add \$s4, \$s3, \$s2	$\$s4 = \$s3 + \$s2$
1	Sub	Sub \$s4, \$s3, \$s2	$\$s4 = \$s3 - \$s2$
2	Slti	Slti \$s4, Imm, \$s2	if( $\$s2 > Imm$ ) ? $\$s4 = 1 : \$s4 = 0$
3	And	And \$s4, \$s3, \$s2	$\$s4 = \$s3 \& \$s2$ ( And Binário)
4	Or	Or \$s4, \$s3, \$s2	$\$s4 = \$s3   \$s2$ ( Or Binário)
5	Xor	Xor \$s4, \$s3, \$s2	$\$s4 = \$s3 \wedge \$s2$ ( Xor Binário)
6	Andi	Andi \$s4, Imm, \$s2	$\$s4 = \$s2 \& Imm$ ( And Binário)

7	Ori	Ori \$s4, Imm, \$s2	$\$s4 = \$s2 \mid Imm$ ( Or Binário)
8	Xori	Xori \$s4, Imm, \$s2	$\$s4 = \$s2 \wedge Imm$ ( Xor Binário)
9	Addi	Addi \$s4, Imm, \$s2	$\$s4 = \$s2 + Imm$
10	Subi	Subi \$s4, Imm, \$s2	$\$s4 = \$s2 - Imm$

Na tabela acima deve-se observar os seguinte detalhes:

- I. O **CODOP** ( código de operação ) de cada instrução é um número inteiro de **4 bits**.
- II. Os endereços **\$s4**, **\$s3** e **\$s2** são endereços de registradores de **4 bits** cada um.
- III. **Imm** significa uma valor imediato de **4 bits**.

De acordo com o enunciado anterior, o formato da instrução deve seguir o padrão a seguir:

CODOP	R1 ou Imm	R2	R3
-------	-----------	----	----

Onde:

R1 é o primeiro operando, R2 é o segundo operando e R3 é o endereço do registrador de destino. É importante resaltar que o tamanho da instrução resulta em um binário de **16 bits**.

O trabalho deverá ser desenvolvido com as ferramentas de descrição e simulação da Intel Altera, Quartus e ModelSim, respectivamente, e prototipado em FPGAs ( DE2-115 ). Tais plataformas serão disponibilizadas pelo laboratório NanoComp. Em regime de aluguel controlado pelo monitor da disciplina.

Para a prototipação na FPGA algumas regras devem ser obedecidas, são estas:

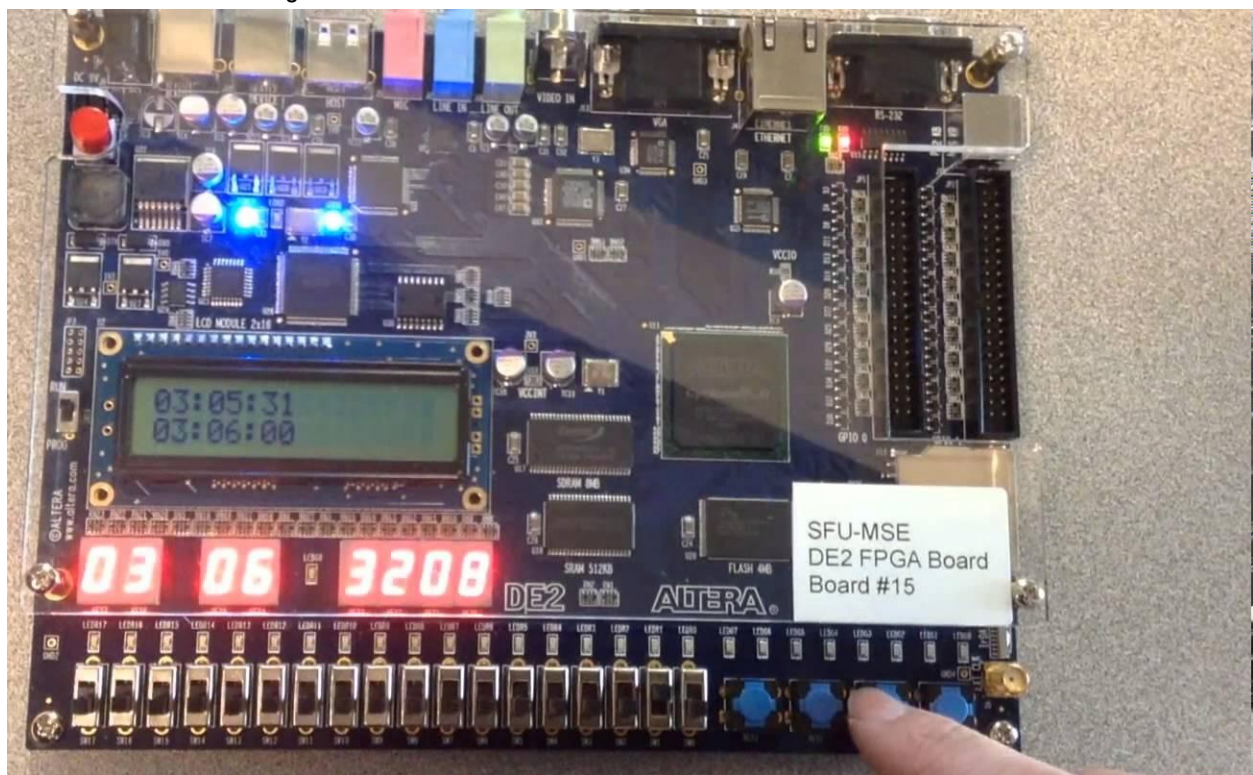
## I. ENTRADA:

- a) Os botões KEY0 e KEY3 devem ser utilizados para alterar o valor mostrado no display de 7 segmentos, da seguinte forma:
- b) Se a chave KEY0 for apertada o conteúdo do registrador cujo endereço é passado nos switches de SW8 a SW11 será impresso nos displays HEX7 e HEX6, e o conteúdo do registrador cujo endereço é passado nos switches de SW4 a SW7 será impresso nos displays HEX5 e HEX4.
- c) Se a chave KEY3 for apertada os *switches* serão interpretados como uma instrução, onde o SW0 é o bit menos significativo e SW15 o bit mais significativo. Nesse caso:
- d) **SW0 a SW3** são os bits que endereçam um dos registradores operando da instrução.( **\$s2**, como mostrado na tabela de instruções)

- e) **SW4 a SW7** endereçam outro dos registradores operandos ou um valor imediato, dependendo da instrução.( **\$s3 ou Imm**, como mostrado na tabela de instruções)
- f) **SW8 a SW11** endereçam o registrador destino da instrução.( **\$s4**, na tabela de instruções)
- g) **SW12 a SW15** determinam que instrução deve ser realizada, isto é, o CODOP.

## II. SAIDA:

- a) Os resultados das operações deverão ser impressos no display de 7 segmentos. Conforme mostrado na figura:



Tomando a imagem acima como exemplo:

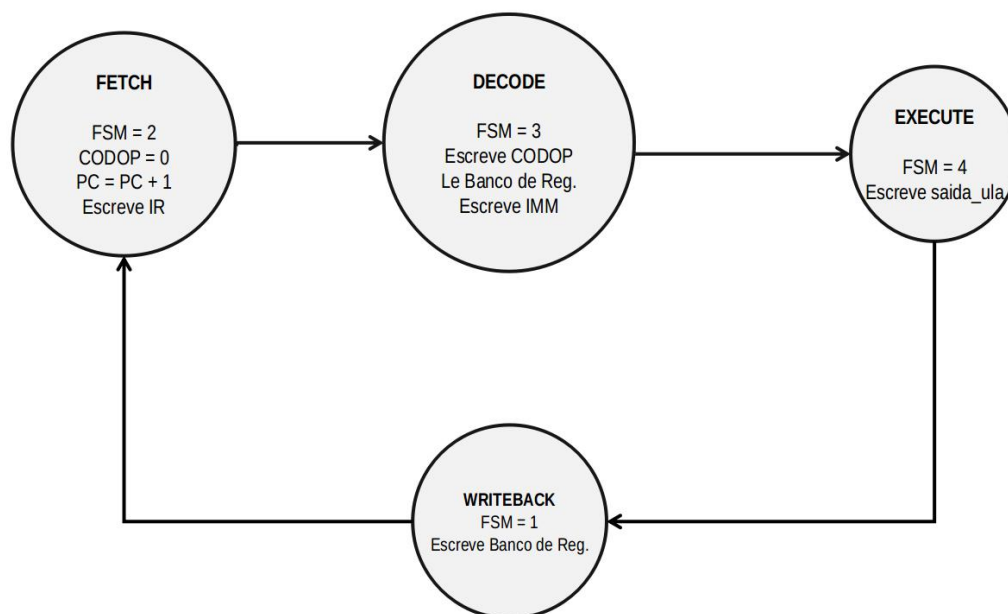
### I. Se apertada a chave KEY0:

- a) 03 é o conteúdo do registrador cujo endereço é passado nos switches de **SW8 a SW11**.
- b) 06 é o conteúdo do registrador cujo endereço é passado nos switches de **SW4 a SW7**.
- c) \*Nesse caso o valor 3208 da figura não importa.

- I. Se apertada a chave KEY3, vamos supor que a operação realizada seja **add \$s4, \$s3, \$s2**:
- a) O conteúdo do registrador **\$s3** é 03, como mostrado nos displays HEX7 e HEX6, os mais à esquerda na figura.
  - b) O conteúdo do registrador **\$s2** é 06, como mostrado nos displays HEX5 e HEX4 na figura.
  - c) 3208 é o resultado da soma.

*\*Vale ressaltar que a imagem é meramente ilustrativa, portanto o valor 3208 não equivale à soma 3 + 6 e foi utilizado apenas para exemplificar em quais displays o resultado deve ser mostrado.*

Finalmente para a implementação da máquina multiciclo, os sinais de controle podem ser representados através de uma máquina de estados finitos. Abaixo está representada a máquina de estados finitos que representa os sinais de controle para o processador multiciclo a ser desenvolvido.



Máquina de estados finitos para a máquina multiciclo.

Onde:

FSM: indica o estado da máquina (FETCH = 1, DECODE = 2, EXECUTE = 3, WRITEBACK = 4).

CODOP: indica a operação a ser executada.

PC: contador de programa.

IR: uma variável que lê a instrução (16 bits).

IMM: valor do imediato (4 bits).

saída\_ula: guarda o valor da saída da ULA.

No estado de FETCH é lida a instrução que será executada na máquina. O valor de FSM muda para 2, indicando que a máquina deve passar para o estado de DECODE. O valor do PC é incrementado para ler a próxima instrução após a execução completa da instrução atual. A instrução atual é salva em IR, para ser decodificada no próximo estágio (DECODE).

No estado de DECODE, o valor de FSM muda para 3, indicando que a máquina deve passar para o estado de EXECUTE. O valor da instrução a ser executada é salvo em CODOP e os valores dos operandos são lidos do banco de registradores. Em caso que a operação seja do tipo imediato, o valor é salvo em IMM.

No estado de EXECUTE, o valor de FSM muda para 4, indicando que a máquina deve passar para o estado de WRITEBACK. Neste estágio, é importante ressaltar que `saida_ula` recebe o valor dependendo da operação indicada em CODOP.

Finalmente no WRITEBACK, o valor de FSM muda para 1, indicando que a máquina deve voltar para o estado de FETCH e a `saida_ula` é salva no banco de registradores.

### **AValiação:**

Observe que o trabalho é composto por três critérios:

- I. Descrição
- II. Simulação
- III. Prototipação

Portanto, a nota final do trabalho é composta pelas notas individuais de cada critério citado acima. Onde, os arquivos de descrição e simulação devem ser submetidos por meio da meta-turma no moodle, enquanto que a prototipação será avaliada por meio de apresentação ao monitor na sala 2310 em horários a serem marcados com o mesmo.

Cronograma		
Etapa	Descrição	Data Entrega
1	Versão multiciclo do processador com suporte para as instruções da ULA	Maio 10 de 2018
2	Versão com pipeline do processador	Junho 12 de 2018
3	Versão processador superscalar I2O2	Julho 5 de 2018

### **Monitorias:**

As aulas de monitoria sobre o trabalho serão realizadas na sala 2310, nos horários a seguir:

Segunda feira, das 10:00 até 12:00 e das 14:00 até 16:00

Quarta feira, das 14:00 até 16:00