

DEPARTAMENTO DE COMPUTAÇÃO (DECOM)
LABORATÓRIO DE ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES I

Professor: Mateus Felipe Tymburibá Ferreira

Aluno: Pedro Henrique Cordeiro

PRÁTICA 11: interligação dos módulos em Verilog do nRisc

Correção de projeto :

A função do processador foi alterada. A função atual é listar o maior e do menor elemento de uma lista que se encontra na memória de dados.

Nesse documento se encontra presente as modificações e alterações realizadas para a nova função, junto das atividades propostas para esse trabalho.

Instruções:

Os bits que não estão definidos na tabela descrevem valores ou registradores a serem usados na instrução.

O processador contém 5 registradores de uso geral, no qual são descritos da seguinte maneira:

- Rw - Não é acessível.
- R0 - 00
- R1 - 01
- R2 - 10
- R3 - 11

RW é o registrador work, ele sempre recebe os resultados das operações, ele é um registrador especial.

Instrução	Sintaxe	Descrição	Bits de controle
Move from work	MOVFW R	Move um valor armazenado em WORK e salva em R.	0000 + 00 + 2bits
Jump one if work is greather	JOGW R	Se RW>R salta a próxima linha do programa.	0001 + 00 + 2 bits

Adds a work	ADDW value	$RW = RW + \text{value}$	0010 + 4bits
Clean the work	CLEARW	$RW = 0$	0011 + 0000
Move to Memory	MTM R	$\text{Memory}[RW] = R$	0100 + 00 + 2 bits
Move from Memory	MFM R	$R = \text{Memory}[RW]$	0101 + 00 + 2 bits
Jump if greater	JOG Ra,Rb	Se $Ra > Rb$ irá saltar a proxima linha de codigo.	0110 + 2 bits + 2 bits
Jump if equal	JOE Ra,Rb	Se $Ra == Rb$ irá saltar a proxima linha de codigo.	0111 + 2 bits + 2 bits
Jump if less	JOL Ra,Rb	Se $Ra < Rb$ irá saltar a proxima linha de codigo.	1000 + 2 bits + 2 bits
Move to	MOVE Ra,Rb	$Ra = Rb$	1001 + 2 bits + 2 bits
Go to	GT value	$PC = PC + \text{value}$	1010 + 4 bits
Fim de programa	END	Para o processador.	1011 + 1111
Go to Back	GTB value	$PC = PC - \text{value}$	1100 + 4 bits

Código:

1	clearw		00110000
2	mfm R1		01010001
3	move R2,R1	# de início o primeiro elemento é o menor	10011001
4	move R3,R1	# e o maior valor	10011110
5	addw 1111	#quantidade de valores a serem analisados	00101111
6	movfw R1		00000001
7	clearw		00110000

8	addw 0001		00100001
9	jogw R1	#inicio do loop	00010001
10	gt 0001		10100001
11	gt 1110	#fim do programa	10101110
12	mfm R0		01010000
13	addw 0001		00100001
14	jog R0,R3		01100011
15	gt 0001		10100001
16	gt 0101	#armazenamento de um valor maior	10100101
17	jol R0,R3		10000011
18	gt 0001		10100001
19	gt 0100	#armazenamento de um valor menor	10100100
20	gtb 0101	#volta inicio do loop	11000101
21	move R2,R0	#maior valor	10011000
22	gtb 0011	#volta inicio do loop	11000011
23	move R3,R0	#menor valor	10011100
24	gtb 0001	#volta inicio do loop	11000001
25	end		10110000

Nota: para o código acima, foi implícito a parte de carregamento de dados na memória de dados, portanto um pré carregamento de dados deve ser executado para então executar o algoritmo.

Datapath do Processador

O datapath do processador do trabalho se encontra em anexo com esse documento.

Descrição dos sinais de controle:

Controle_A	Seleciona a entrada de escrita no banco de registradores. 00:Escrita de algum valor vindo da instrução. Ex: CLEARW. 01:Escrita de algum valor vindo da memória. Ex:MFM. 10:Escrita de algum valor vindo da ULA. Ex:ADDW.
Controle_B	Seleciona para onde o resultado de operação da ULA irá. 0:Endereço de memória. 1:Dado a ser escrito em algum registrador.
Controle_ULA	Seleciona a operação que a ULA irá executar. 0:soma.

	1:subtração.
Controle_Escrita_Reg	Habilita escrita nos registradores.(0 desabilita, 1 habilita).
Controle_Escrita_MEMO	Habilita escrita na memória.(0 desabilita, 1 habilita).
Controle_Leitura_MEMO	Habilita a leitura de dados na memoria.(0 desabilita,1 habilita).
Controle_Jump	Habilita os Jumps. JOE,JOL e JOG. (0:desabilita, 1:habilita)
Controle_GoTo	Habilita GT. (0:desabilita, 1:habilita)
Controle_Input_ULA	Seleciona qual será a entrada para a ULA. 00:Valor de um registrador. Ex:JOG. 01:0 Ex:MTM. 10: Valor de um imediato. Ex:ADDW
Controle_ULA_GT	Seleciona a operação da mini ULA (ULA GT) 0 : soma. 1: subtração.
Halt	Habilita a execução do programa. Entrada em PC e no banco de registradores. 1:habilita 0:desabilita Nota: se Halt=0 tudo no processador para!

Caminho de cada instrução dentro do processador e sinais de controle para cada instrução:

MOVFW:

Caminho da instrução no processador:

Memoria de programa, Banco de registradores, ULA, Banco de registradores.

Sinais de controle da instrução:

Controle_A: 01 Controle_B: 1 Controle_ULA: 0 Controle_Escrita_Reg: 1 Controle_Escrita_Memo: 0 Controle_ULA_GT : 0	Controle_Leitura_Memo: 0 Controle_Jump: 0 Controle_GoTo: 0 Controle_Input_ULA: 01 Halt : 1
--	--

JOG,JOE,JOL e JOGW:

Caminho da instrução no processador:

Memoria de programa, Banco de registradores, ULA, portas logicas, Somador, PC.

Sinais de controle da instrução:

Controle_A: 10 Controle_B: 0 Controle_ULA: 1 Controle_Escrita_Reg: 0 Controle_Escrita_Memo: 0 Controle_ULA_GT : 0	Controle_Leitura_Memo: 0 Controle_Jump: 1 Controle_GoTo: 0 Controle_Input_ULA: 00 Halt : 1
--	--

GT:

Caminho da instrução no processador:

Memoria de programa, Somador, Mux ,PC.

Sinais de controle da instrução:

Controle_A: 10 Controle_B: 1 Controle_ULA: 1 Controle_Escrita_Reg: 0	Controle_Leitura_Memo: 0 Controle_Jump: X Controle_GoTo: 1 Controle_Input_ULA: 10
---	--

Controle_Escrita_Memo: 0 Controle_ULA_GT : 0	Halt : 1
---	----------

GTB:

Caminho da instrução no processador:

Memoria de programa, Somador, Mux ,PC.

Sinais de controle da instrução:

Controle_A: 10 Controle_B: 1 Controle_ULA: 1 Controle_Escrita_Reg: 0 Controle_Escrita_Memo: 0 Controle_ULA_GT : 1	Controle_Leitura_Memo: 0 Controle_Jump: X Controle_GoTo: 1 Controle_Input_ULA: 10 Halt : 1
--	--

MTM:

Caminho da instrução no processador:

Memoria de programa,Banco de registradores,ULA,Memoria de Dados.

Sinais de controle da instrução:

Controle_A: 10 Controle_B: 0 Controle_ULA: 1 Controle_Escrita_Reg: 0 Controle_Escrita_Memo: 1 Controle_ULA_GT : 0	Controle_Leitura_Memo: 0 Controle_Jump: 0 Controle_GoTo: 0 Controle_Input_ULA: 10 Halt : 1
--	--

MFM:

Caminho da instrução no processador:

Memoria de programa,Banco de Registradores,ULA,Memoria de Dados,Banco de Registradores.

Sinais de controle da instrução:

Controle_A: 10 Controle_B: 0 Controle_ULA: 0 Controle_Escrita_Reg: 1 Controle_Escrita_Memo: 0 Controle_ULA_GT : 0	Controle_Leitura_Memo: 1 Controle_Jump: 0 Controle_GoTo: 0 Controle_Input_ULA: 10 Halt : 1
--	--

MOVE:

Caminho da instrução no processador:

Memoria de programa,Banco de registradores,ULA,Banco de registradores.

Sinais de controle da instrução:

Controle_A: 01 Controle_B: 1 Controle_ULA: 0 Controle_Escrita_Reg: 1 Controle_Escrita_Memo: 0 Controle_ULA_GT : 0	Controle_Leitura_Memo: 0 Controle_Jump: 0 Controle_GoTo: 0 Controle_Input_ULA: 10 Halt : 1
--	--

ADDW:

Caminho da instrução no processador:

Memoria de programa,Banco de registradores,ULA,Banco de registradores.

Sinais de controle da instrução:

Controle_A: 01 Controle_B: 1 Controle_ULA: 0 Controle_Escrita_Reg: 1 Controle_Escrita_Memo: 0	Controle_Leitura_Memo: 0 Controle_Jump: 0 Controle_GoTo: 0 Controle_Input_ULA: 01 Halt : 1
---	--

Controle_ULA_GT : 0	
---------------------	--

CLEARW:

Caminho da instrução no processador:

Memoria de programa,Banco de registradores.

Sinais de controle da instrução:

Controle_A: 00 Controle_B: 0 Controle_ULA: 0 Controle_Escrita_Reg: 1 Controle_Escrita_Memo: 0 Controle_ULA_GT : 0	Controle_Leitura_Memo: 0 Controle_Jump: 0 Controle_GoTo: 0 Controle_Input_ULA: 00 Halt : 1
--	--

END:

Caminho da instrução no processador:

Memoria de programa,Banco de registradores,ULA,Banco de registradores.

Sinais de controle da instrução:

Controle_A: XX Controle_B: X Controle_ULA: X Controle_Escrita_Reg: 0 Controle_Escrita_Memo: 0 Controle_ULA_GT : 0	Controle_Leitura_Memo: 0 Controle_Jump: 0 Controle_GoTo: 0 Controle_Input_ULA: XX Halt : 0
--	--

Explicação do Bloco de Controle: esse bloco tem o objetivo de selecionar a saída do mux jump, que tem as possibilidades de saída entre o valor 1 e 2. Esse valor posteriormente seria somado ao valor de PC atual. É usado nas instruções de desvio condicional de avanço duplo, em que o salto é de apenas uma linha seguinte de código. As entradas são os sinais vindo da ULA e o sinal de controle de Jump para habilitar o salto duplo.

Zero	Negativo	Controle_Jump	Saída	Instrução
0	0	0	0	
0	0	1	1	JOG
0	1	0	0	JOL
0	1	1	1	
1	0	0	0	
1	0	1	1	JOE
1	1	0	0	
1	1	1	0	

Os módulos dos componentes e as questões a serem respondidas estão em anexo no formato PDF.

Questão 3 (Explicação do Módulo de Simulação):

A cada ciclo de execução do processador, era gerado um print dos registradores e das primeiras posições da memória de dados. Junto a esse conjunto de dados o código em binário e o valor de PC eram printados também.

O módulo de simulação foi construído de forma que a cada ciclo de execução, o clock externo variasse.