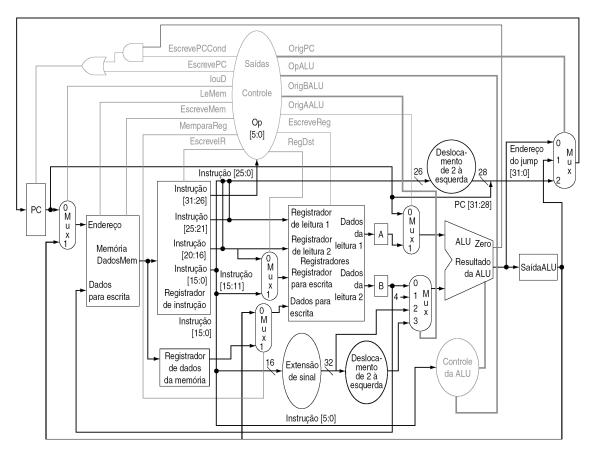
Arquitetura e Organização de Computadores Turma C - 2016/02

Projeto MIPS Multiciclo

Objetivo: montar e simular euma versão do processador MIPS multiciclo.

Descrição:

Neste trabalho deve-se desenvolver uma arquitetura MIPS multiciclo em VHDL. Tomando por base os trabalhos da disciplina, deve-se interligar todos os módulos relativos à parte operativa e à parte de controle realizando a arquitetura ilustrada na figura 1.



O diagrama acima não suporta a execução de todas as instruções do MIPS. As instruções básicas a serem implementadas são as seguintes:

• LW, SW, ADD, ADDi, SUB, AND, NAND, OR, NOR, XOR, SLT, J, BEQ, BNE

Não são necessárias as instruções ADDu e SUBu. Para o projeto da disciplina, os diferentes grupos deverão implementar algumas das instruções adicionais descritas a seguir.

0								
•	ORi: o	ori rs	, rt,	imedi	ato b	oreg[rt]	= breg[rs] 0x0000iiii;	
	0xd	rs	rt	imediato				
•	ANDi: andi rs, rt, imediato breg[rt] = breg[rs] & 0x0000iiii;							
	0xc	rs	rt	imediato				
•	<pre>SLL: deslocamento lógico à esquerda sll rd, rt, shamt breg[rd] = breg[rt] << shamt;</pre>							
	0x0	rs	rt	rd	shamt	0		
•	<pre>SRL: deslocamento lógico à direita srl rd, rt, shamt breg[rd] = breg[rt] >> shamt;</pre>							
	0×0	rs	rt	rd	shamt	2		
•	BGEZ: desvio se maior que ou igual a zero bgez rs, label pc = pc + deslocamento*4;							
	0x1	rs	1	deslocamento				
•	BLTZ: desvio se menor que zero bgez rs, label pc = pc + deslocamento*4;							
	0×1	rs	Θ	deslocamento				
•	SLTI: setar se menor que constante imediata slti rd, rs, imediato breg[rd] = breg[rs] < sgn_ext(imediato);							
	0xa	rs	rd	imediato				
•	JAL: JUMP and LINK jal LABEL breg[31] = PC + 4; PC = PC(31,28) & imm26 & "00"							
	0×3	imediato 26 bits						
•	<pre>JR: jump register. jr \$rs</pre>							
	0×0	rs	0	0	0	0×8		

O processador deve ser simulado no ModelSim e implementado em FPGA. Neste caso, para verificação de seu funcionamento, deve-se observar o PC, o RI, o RDM e a saída da ULA através dos mostradores de 7 segmentos.

Para entrada e saída de dados deve-se utilizar:

- botão para acionar o relógio
- mostrador de 7 segmentos para exibir conteúdo de RI e SaidaALU

O PC deve ter 32 bits. A memória tem apenas 256 palavras de 32 bits, de forma que apenas 8 bits do PC devem ser utilizados no seu endereçamento. Para executar um programa gerado pelo MARS, devem ser carregados o código e os dados do programa. Os endereços da área de dados devem ser mapeados para a região de memória que começa no endereço 128 no FPGA. Assim, o endereço de dado deve ser gerado concatenando os bits [8 dowto 2] do registrador de saída da ULA com o bit '1' na posição mais significativa: '1' & alu_out(8 downto 2).

Um arquivo comprimido com todos os módulos VHDL do MIPS multiciclo é disponibilizado no Moodle. O código MIPS a ser carregado na memória está contido no arquivo **mem.mif**.

Para exibição dos dados nos mostradores, utilizar os acionadores de display de 7 segmentos feito na primeira aula de laboratório.

A verificação do processador consistirá na execução de programas gerados a partir do MARS.

Entrega: até 15 de dezembro. Apresentar o código VHDL simulando no ModelSim e executando em FPGA.