LABORATÓRIO DE ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES I

AULA 5: Projeto de um Processador: planejamento do conjunto de instruções

Professor: Mateus Felipe Tymburibá Ferreira
Data: 08/07/2021

Aluno: Darmes Araújo Dias

1) Explique o que o programa embarcado deverá fazer.

O programa embarcado deverá somar os números pares de um vetor. Por exemplo, o vetor será 1,2,3,4,5,6,7,8 e a soma terá que ser 20.

2) Apresente a lista de instruções suportadas pelo seu processador.

Instrução	opcode	rd	rs	immediate	description
load	000	X	X	XXX	rd = MEM[rs*immediate]
store	001	X	X	XXX	MEM[rd*immediate]=rs
addi	010	X	-	XXXX	rd = rd + immediate
add	011	X	X	-	rd = rd + rs
beq	100	X	X	LABEL(XXX)	if (x == y) go to Label
and	101	X	X	-	rd = rd & rs
or	110	X	X	-	rd = rd | rs
halt	111	-	-	-	PC = PC

3) Explique a operação realizada por cada uma das instruções.

➤ load: carrega o registrador destiny com o conteúdo da posição do registrador source multiplicado pelo imediato.
➤ store: carrega no endereço de memória na posição do conteúdo do registrador destiny multiplicado pelo imediato, o conteúdo do registrador source.
➤addi: adiciona no conteúdo do registrador destiny o imediato.
➤add: faz a soma do conteúdo de dois registradores e armazena em um registrador destiny.

➤beq: faz a comparação entre os conteúdos de dois registradores e se caso forem iguais ocorre um desvio para o label indicado.
➤and: faz operação and com conteúdos de dois registradores e armazena o resultado em um registrador destiny.
➤or: faz operação or com conteúdos de dois registradores e armazena o resultado em um registrador destiny.
➤halt: Interrompe a execução.

4) Mostre a representação (sintaxe) em assembly de cada instrução.

Instrução	Representação em Assembly
Load Word	lw $s1, $t0(4)
Store Word	sw $s1, $t0(4)
Add immediate	addi $t0,$t1,4
Add	add $t0,$t1,$t2
Branch if equal	beq $t0,$t1, Label
And	and $t0,$t1,$t2
Or	or $t0,$t1,$t2
Halt	halt

5) Indique o formato binário de cada uma das instruções, apontando o tamanho (em número de bits) e a função de cada campo das instruções.

Instrução	opcode	rd	rs	immediate
load	000	X	X	XXX
store	001	X	X	XXX
addi	010	X	-	XXXX
add	011	X	X	-
beq	100	X	X	LABEL(XXX)
and	101	X	X	-
or	110	X	X	-
halt	111	-	-	-

Observações:
O número de x 's quer dizer o número de bits, por exemplo, como meu processador só possui dois registradores, em rs(reg. source) e rd(reg. destiny) só há um X, ou seja, será 0 ou 1.
Terei de prestar atenção nos labels, porque não poderia haver uma label a mais de 8 endereços de memória de outra label, porque o branch só pode ser feito de 8 em 8 endereços(devido a 3 bits).

6) Justifique todas as suas decisões de projeto.

A ideia sobre o que o programa embarcado deverá fazer eu tive porque queria algo bem simples, porém, tinha que envolver um loop, um comando de desvio e que houvesse halt, dessa maneira, acho que a soma de pares de um vetor atenderia bem os requisitos.
Meu processador terá apenas 2 registradores, e por ser um processador de 8 bits, vai operar apenas palavras de 8 bits. O processador terá apenas 8 instruções suportadas e acredito que elas suprem o que vai ser feito. E não serão usadas instruções com mais de 8 bits, evitando a complexidade de ter que dividir instruções.