

## LISTA 8 - Resolução

### Questão 1)

a) *Instrução*: padrão de código binário que comanda o processador na execução de determinada tarefa.

*Palavra binária*: número de bits que a ULA reconhece e processa de uma só vez.

*Mnemônicos*: palavras curtas que representam a instrução binária na máquina (add, mov, if, for).

b) *Linguagem de máquina*: conjunto de instruções binárias para a execução de determinadas tarefas.

*Linguagem de programação*: método padronizado para comunicar instruções para um computador. Pode ser de alto nível, mais próxima da linguagem humana, ou baixo nível, mais próxima da linguagem de máquina.

Assembly é uma linguagem de programação de baixo nível.

### Questão 2)

*Opcode*: Código binário da operação a ser executada.

*Operando*: Dados ou endereços a serem manipulados, atrelados à um opcode.

Com um Opcode de 8 bits, podem existir até  $2^8 = 256$  instruções.

### Questão 3)

*Ciclo de Máquina*: É uma forma de contar o tempo no processamento, definida como uma quantidade fixa de períodos do clock, que varia conforme o microprocessador. Essa quantidade é a quantia necessária de períodos (ou pulsos) para que o microprocessador busque uma palavra (opcode ou operando) na memória ROM, armazene ela no IR (Instruction Register), decodifique a palavra e a execute.

*Ciclo de Instrução*: É o tempo de uma instrução ser completamente executada e pode ser variável para cada instrução. Como algumas instruções incluem mais de um operando, podem ser necessários vários ciclos de máquina para compor um único ciclo de instrução.

### Questão 4)

a)	<i>LDA A:</i>	Ro = A
	<i>ADD C:</i>	Acc = A+C; Ro = A+C
	<i>STA X:</i>	X = A+C
	<i>LDA B:</i>	Ro = B
	<i>MPY D:</i>	Acc = B*D; Ro = B*D
	<i>SUB E:</i>	Acc = B*D-E; Ro = B*D-E
	<i>STA Y:</i>	Y = B*D-E
	<i>LDA X:</i>	Ro = X

*ADD Y:*         $Acc = X+Y; Ro = X+Y$   
*DIV F:*         $Acc = (X+Y)/F; Ro = (X+Y)/F$   
*STA X:*         $X = (X+Y)/F$

Ou seja,  $X = (A+C+B*D-E)/F$

**b)** Com um opcode de 4 bits, podem haver  $2^4 = 16$  instruções.

#### **Questão 5)**

No 8051 são necessários 12 períodos de Clock (12T) para cada ciclo de máquina.

$f = 12\text{MHz}$

$T = 1/f = 1/12 \mu\text{s}$

Ciclo de máquina =  $12T = 1\mu\text{s}$

#### **Questão 5)**

**a)** A diferença principal é na quantidade de instruções no set. O conjunto RISC possui uma quantidade reduzida de instruções de tamanho fixo, quase todas com mesmo tempo de execução. O conjunto CISC possui uma quantidade maior de instruções de de tamanho variado entre elas, logo, programas em RISC tendem a ocupar mais linhas de código.

**b)** A arquitetura Harvard, diferente da Von Neumann, não possui barramento compartilhado pelas memórias RAM e ROM. Possui um duto separado para dados (RAM) e instruções (ROM). É mais rápido pois consegue acessar as duas memórias ao mesmo tempo, permite pipelining e, em geral, usado com conjunto RISC.

**c)** Pipelining (linha de montagem) é uma técnica que permite mais de uma instrução ser executada ao mesmo tempo. O microprocessador lê uma instrução ao mesmo tempo que executa a instrução lida anteriormente. Requer RISC, pois as instruções devem ter o mesmo tamanho (com exceção das de salto).

#### **Questão 6)**

Von Neumann é geralmente CISC. Harvard é geralmente RISC e, se o for, permite pipelining.

#### **Questão 7)**

Total:  $2 + 8 + 32 = 42$

**1.** Um programa em RISC, por ter instruções mais simples, ocupa mais linhas que um CISC para a execução da mesma tarefa.

**4.** Não necessariamente o Opcode é maior. As instruções apenas são mais específicas, podendo ter vários operandos.

**16.** A diferença entre Harvard e VonNeumann está no duto de dados, que na VonNeumann é dividido entre duto de dados e de endereços.