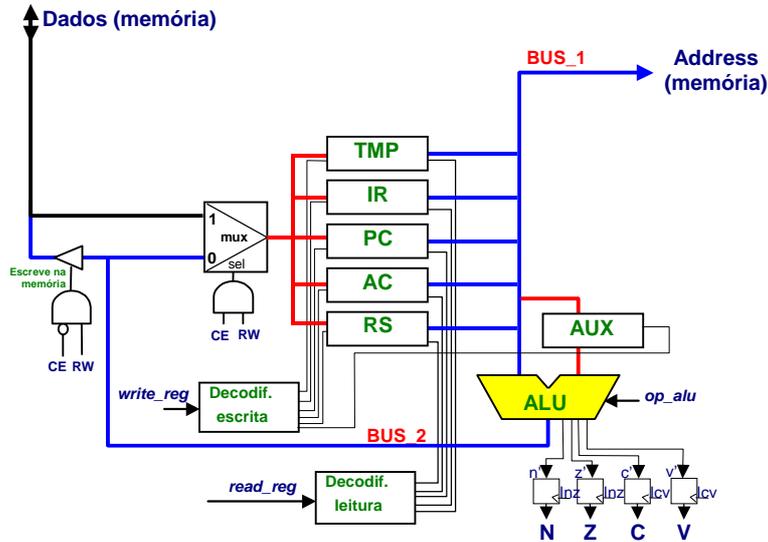


Exercícios com solução sobre a primeira parte da matéria (Unidades 1 e 2)

1. Na Figura abaixo é proposta uma organização alternativa para o Bloco de Dados (BD) da arquitetura Cleópatra. Considere que o Bloco de Controle alterou-se para o novo BD.



Pede-se, para esta nova organização:

- (a) Detalhar o ciclo de busca de instruções em microinstruções. Seu detalhamento deve gastar o **número mínimo absoluto** de ciclos de relógio permitido por esta nova organização; **Observação Importante:** não se esqueçam de gerar o código objeto **ANTES** de realizar a microssimulação e não deixem de prestar muita atenção aos **modos de endereçamento** usados.
- (b) Dado o trecho de programa abaixo para esta nova organização da arquitetura Cleópatra, realize a microssimulação, mostrando seus passos na tabela abaixo.

1	.code	10	endereço	código	objeto
2	prog: lda n	11	00h	44h	06h
3	add #0c2h	12	02h	50h	C2h
4	sta m	13	04h	24h	07h
5	.endcode	14	.endcode		
6	.data	15	.data		
7	n: db #0C8h	16	06h	C8h	
8	m: db #0h	17	07h	00h	
9	.enddata	18	.enddata		

Soluções:

CK	DATA	ADDRESS	AUX	TMP	IR	PC	AC	n/z/c/v	Microinstrução
0	44	00	00	00	44	00	00	0/0/0/0	IR ← PMEM(PC)
1	ZZ	00	00	00	44	01	00	0/0/0/0	PC++
2	06	01	00	06	44	01	00	0/0/0/0	TMP ← PMEM(PC)
3	ZZ	01	00	06	44	02	00	0/0/0/0	PC++
4	C8	06	00	06	44	02	C8	0/0/0/0	AC ← PMEM(TMP)
5	ZZ	C8	00	06	44	02	C8	1/0/0/0	AC ← AC; LNz
6	50	02	00	06	50	02	C8	0/0/0/0	IR ← PMEM(PC)
7	ZZ	02	00	06	50	03	C8	0/0/0/0	PC++
8	C2	03	00	C2	50	03	C8	0/0/0/0	TMP ← PMEM(PC)
9	ZZ	03	00	C2	50	04	C8	0/0/0/0	PC++
10	ZZ	C2	C2	C2	50	04	C8	0/0/0/0	AUX ← TMP
11	ZZ	C8	C2	C2	50	04	8A	1/0/1/0	AC ← AC+AUX; LNz; LCV
12	24	04	C2	C2	24	04	8A	1/0/1/0	IR ← PMEM(PC)
13	ZZ	04	C2	C2	24	05	8A	1/0/1/0	PC++
14	07	05	C2	07	24	05	8A	1/0/1/0	TMP ← PMEM(PC)
15	ZZ	05	C2	07	24	06	8A	1/0/1/0	PC++
16	ZZ	8A	8A	07	24	06	8A	1/0/1/0	AUX ← AC
17	8A	07	8A	07	24	06	8A	0/0/0/0	PMEM(TMP) ← AUX

2. O objetivo da disciplina de Organização de Computadores é a apreensão dos conceitos mais fundamentais de organização e arquitetura de computadores. Assim, demonstre seu atual conhecimento, definindo e diferenciando estes dois conceitos. Ou seja, o que é organização de computadores, o que é arquitetura de computadores e como estes conceitos são diferentes e complementares. No caso de organização de computadores, existem dois modelos fundamentais que se adaptam para descrever qualquer processador. Nomeie, defina, diferencie e descreva as principais características de cada um destes modelos de organização. Seja preciso e sucinto (2 pontos).

Solução:

Organização de Computadores estuda a visão que o engenheiro que constrói a máquina possui do processador. Este estuda e domina a composição de elementos de hardware tais como ULAs, registradores de dados e de controle, definição da estrutura de controle global do sistema (por exemplo usando FSMs ou implementações microprogramadas), etc. Arquitetura de Computadores consiste na visão que aparece do processador ao acrescentar uma primeira camada de abstração sobre o hardware do processador. O principal elemento é a abstração de um processador como um sistema capaz de executar instruções de um conjunto de instruções finito e bem definido (em inglês, isto é denominado Instruction Set Architecture do processador, ou ISA). Além deste elemento, os programas são vistos como manipulando conteúdos de memória e de alguns registradores do hardware (aqueles ditos “visíveis” ao programador do processador), tipicamente os registradores que armazenam dados úteis de um programa e algumas informações de controle. Esta é a visão do programador em linguagem de montagem ou de máquina. Os dois modelos de organização mais difundidos para classificar organizações são o modelo de Von Neumann e o modelo Harvard. O primeiro aplica-se a processadores onde dados e instruções compartilham um meio de armazenamento único (memória unificada de dados e instruções), enquanto o modelo Harvard pressupõe que dados e instruções residem em memórias distintas. Em tese, organizações Harvard facilitam a implementação de paralelização do processamento em relação ao modelo seqüencial de programação, habilitando mais facilmente a construção de processadores de alto desempenho.

3. Gere o código objeto relativo ao trecho de programa e área de dados abaixo. A seguir, realize a microsimulação completa do trecho, usando a tabela de microsimulação dada. O número de ciclos da tabela é meramente indicativo. Ou seja, sua microsimulação pode ter mais, menos o exatamente o mesmo número de linhas que as disponíveis.

1	.code	1	endereço	código	objeto
2	prog: lda n,I	2	00h	48h	06h
3	add #0c2h	3	02h	50h	C2h
4	sta #m	4	04h	20h	07h
5	.endcode	5	.endcode		
6	.data	6	.data		
7	n: db #xuxu	7	06h	08h	
8	m: db #0h	8	07h	00h	
9	xuxu: db #C8h	9	08h	C8h	
10	.enddata	10	.enddata		

Soluções:

CK	DATA	ADDRESS/MAR	MDR	IR	PC	AC	n/z/c/v	Microinstrução
1	ZZ	00	00	00	00	00	0/0/0/0	MAR ← PC
2	48	00	48	00	01	00	0/0/0/0	MDR ← PMEM(MAR); PC++
3	ZZ	00	48	48	01	00	0/0/0/0	IR ← MDR
4	ZZ	01	48	48	01	00	0/0/0/0	MAR ← PC
5	06	01	06	48	02	00	0/0/0/0	MDR ← PMEM(MAR); PC++
6	ZZ	06	06	48	02	00	0/0/0/0	MAR ← MDR
7	08	06	08	48	02	00	0/0/0/0	MDR ← PMEM(MAR)
8	ZZ	08	08	48	02	00	0/0/0/0	MAR ← MDR
9	C8	08	C8	48	02	00	0/0/0/0	MDR ← PMEM(MAR)
10	ZZ	08	C8	48	02	C8	1/0/0/0	AC ← MDR; LNz=1

7	ZZ	02	C8	48	02	C8	1/0/0/0	MAR ← PC
8	50	02	50	48	03	C8	1/0/0/0	MDR ← PMEM(MAR); PC++
9	ZZ	02	50	50	03	C8	1/0/0/0	IR ← MDR
10	ZZ	03	50	50	03	C8	1/0/0/0	MAR ← PC
11	C2	03	C2	50	04	C8	1/0/0/0	MDR ← PMEM(MAR); PC++
12	ZZ	03	C2	50	04	8A	1/0/1/0	AC ← AC+AUX; LNz=1; LCV=1
13	ZZ	04	C2	50	04	8A	1/0/1/0	MAR ← PC
14	20	04	20	50	05	8A	1/0/1/0	MDR ← PMEM(MAR); PC++
15	ZZ	04	20	20	05	8A	1/0/1/0	IR ← MDR
16	ZZ	05	20	20	05	8A	1/0/1/0	MAR ← PC
17	07	05	07	20	06	8A	1/0/1/0	MDR ← PMEM(MAR); PC++
18	ZZ	07	07	20	06	8A	1/0/1/0	MAR ← MDR
19	8A	07	07	20	06	8A	1/0/1/0	PMEM(MAR) ← AC