

Nome(legível)/Matrícula: <u><i>PROF. CLAYTON, Gabarito</i></u>	Nota: <u><i>AP1</i></u>
Disciplina: ARQUITETURA DE COMPUTADORES	
Professor: CLAYTON JONES ALVES DA SILVA	
Curso: ENGENHARIA DE COMPUTAÇÃO	Data:25/04/2022 2022.1

ATENÇÃO !

1. Não é permitido sair de sala nos primeiros 30 minutos do horário da prova. Após esse tempo, o aluno que sair de sala estará necessariamente concluindo a sua prova. Nenhum aluno poderá ingressar após a saída do primeiro aluno.
2. Após a correção, apenas o próprio aluno poderá examinar ou receber a prova.
3. Não é permitida a consulta a qualquer fonte.
4. A prova **poderá ser resolvida a lápis** e deverá ser devolvida integralmente ao professor (folhas de perguntas e respostas).
5. **Nos termos do Regimento Interno do Centro Universitário IBMEC RJ em caso de conduta inadequada o aluno estará sujeito a pena disciplinar.**

Secretaria Acadêmica

CONCEITOS BÁSICOS

1ª QUESTÃO (3,0 pontos)

- a. Um código de uma máquina computacional utiliza instruções que são convertidas e executadas uma a uma por uma máquina em um nível inferior, aplicando o modelo de arquitetura de máquinas de níveis. Como se designa a conversão de códigos utilizada? Qual é a outra forma consagrada para converter códigos de uma linguagem de determinado nível em códigos de outra linguagem, que serão executados em uma máquina em nível inferior, e como é implementada?
- b. Admitindo que o nível inferior a que se refere a questão anterior compreende a execução de instruções envolvendo registradores, como PC, Acc e MQ na máquina de Von Neumann. Segundo o modelo de 6 níveis, como se designa o nível imediatamente superior de que trata o enunciado? Explicar o seu significado.
- c. As portas lógicas executam instruções que operam dados na forma elétrica, abstraídos como 0s e 1s. Considerando o modelo da máquina de 6 níveis, a qual nível pertencem os circuitos combinacionais construídos com portas lógicas? Cite um exemplo desse tipo de circuito.
- d. Um programa armazenado em memória, conceito utilizado pela máquina de Von Neumann, ocupa os endereços compreendidos entre as posições de memória 0x0FF e 0x1FF. Considerando a arquitetura interna da máquina, qual é o elemento que armazenará especificamente esses endereços? Explicar o seu comportamento na execução das instruções.
- e. As instruções da máquina IAS possuem 20 bits, com *opcode* de 8 bits. O que significa o *opcode*? Quantos bits cada instrução deve possuir de operando? Admitindo instrução de modo imediato, com três campos de operando, quantos bits serão usados por campo? Admitindo instrução de modo direto, qual é o maior endereço de dado endereçado pela instrução (em hexadecimal)?
- f. Seja uma máquina computacional de arquitetura similar à máquina de Von Neumann. A máquina possui instruções de modo direto com operandos de 10 bits. Dado que a faixa de endereços que armazenam dados inicia-se em 0x150 e termina na última linha de endereços, quantas linhas de endereços poderão ser usadas? Se cada linha possuir 16 bits, quantos bytes de dados, no máximo, poderão ser utilizados?

NOME/MATRÍCULA:

- g. No modelo da arquitetura da máquina IAS, após as instruções serem buscadas precisam ser decodificadas. Explique o que significa *decodificação da instrução*. Qual é a relação da decodificação com o sinal de *clock*?
- h. Quais são os elementos da máquina IAS que podem armazenar o resultado da execução da instrução? Qual é a relação desses elementos com os circuitos lógicos e aritméticos?
- i. Na máquina de Von Neumann cada linha de memória armazena 40 bits. Considerando a questão f, admitindo que todos os dados sejam acessados em um intervalo de 2 nanosegundos, qual é a taxa de transmissão (velocidade) observada no barramento de dados?

PROGRAMA EM NÍVEL DE MÁQUINA E ASSEMBLY - IAS

2ª QUESTÃO (3,0 pontos)

Elaborar um programa para ler e somar os dados armazenados nos endereços 0x00B a 0x010 da memória de uma máquina IAS. A primeira instrução do programa deve desviar a execução para o endereço 0x100. O programa deve subtrair o resultado encontrado do dado armazenado no endereço 0x011 e multiplicar por 2. O resultado final deve ser armazenado no endereço 0x012. Apresentar o código em *assembly* e o código convertido para a linguagem de máquina (em hexadecimal) – admitir que os endereços das posições acima de 0x012 podem ser usados para armazenar dados temporários. O apêndice apresenta a tabela de códigos do processador IAS. Supondo que cada instrução seja executada em 0.01 milissegundos, qual será o tempo de execução do programa elaborado? Considerando instruções e dados, quantos *bytes* será utilizado pelo programa?

REPRESENTAÇÃO DE DADOS NUMÉRICOS

3ª QUESTÃO (2,5 pontos)

Seja uma arquitetura que utiliza 12 bits para representar os dados numéricos.

- a. Qual seria a faixa de representação (em hexadecimal) para representar números inteiros em excesso 128?
- b. Dada a grandeza numérica, no sistema hexadecimal, 0x37A, na representação em complemento de 2, qual seria o correspondente valor no sistema decimal? Qual seria o seu simétrico, no sistema hexadecimal?
- c. Para representar números de ponto flutuante, a arquitetura usa 5 bits para o expoente. Admitindo que representa o expoente em bit sinal, qual seria a resolução de representação numérica no sistema? Admitindo representação da base em complemento de 1, qual seria a faixa de representação, em hexadecimal, definida pelo valor da base igual a -35, em decimal?

ARITMÉTICA BINÁRIA

4ª QUESTÃO (1,5 pontos)

Apresentar as operações abaixo solicitadas, apresentando o resultado em hexadecimal com 8 bits:

- a. 101101 + 01000, representação em binário puro
- b. 010111 – 000111, representação em complemento de 2
- c. 010111 – 000111, representação em complemento de 1

Tipo de Instrução	Código de operação	Representação simbólica	Descrição
Transferência de dados	00001010	LOAD MQ	Transfere o conteúdo do registrador MQ para o acumulador AC
	00001001	LOAD MQ,M(X)	Transfere o conteúdo da posição de memória X para MQ
	00100001	STOR M(X)	Transfere o conteúdo do acumulador para a posição de memória X
	00000001	LOAD M(X)	Transfere M(X) para o acumulador
	00000010	LOAD - M(X)	Transfere - M(X) para o acumulador
	00000011	LOAD M(X)	Transfere o valor absoluto de M(X) para o acumulador
	00000100	LOAD - M(X)	Transfere - M(X) para o acumulador

Tipo de Instrução	Código de operação	Representação simbólica	Descrição
Desvio incondicional	00001101	JUMP M(X,0:19)	A próxima instrução a ser executada é buscada na metade esquerda de M(X)
	00001110	JUMP M(X,20:39)	A próxima instrução a ser executada é buscada na metade direita de M(X)
Desvio condicional	00001111	JUMP+(X,0:19)	Se o número no acumulador é um valor não-negativo, a próxima instrução a ser executada é buscada na metade esquerda de M(X)
	00010000	JUMP+M(X,20:39)	Se o número no acumulador é um valor não-negativo, a próxima instrução a ser executada é buscada na metade direita de M(X)

Tipo de Instrução	Código de operação	Representação simbólica	Descrição
Aritmética	00000101	ADD M(X)	Soma M(X) a AC; armazena o resultado em AC
	00000111	ADD M(X)	Soma M(X) a AC; armazena o resultado em AC
	00000110	SUB M(X)	Subtrai M(X) de AC; armazena o resultado em AC
	00001000	SUB M(X)	Subtrai M(X) de AC; armazena o resto em AC
	00001011	MUL M(X)	Multiplica M(X) por MQ; armazena os bits mais significativos do resultado em AC, armazena os bits menos significativos em MQ.
	00001100	DIV M(X)	Divide AC por M(X); armazena o quociente em MQ e o resto em AC.
	00010100	LSH	Multiplica o acumulador por 2 (isto é, desloca os bits uma posição para a esquerda).
	00010101	RSH	Divide o acumulador por 2 (isto é, desloca os bits uma posição para a direita).

NOME/MATRÍCULA:

Tipo de Instrução	Código de operação	Representação simbólica	Descrição
Alteração de endereço	00010010	STOR M(X,8:19)	Substitui o campo de endereço à esquerda de M(X) pelos 12 bits mais à direita de AC.
	00010011	STOR M(X,28:39)	Substitui o campo de endereço à direita de M(X) pelos 12 bits mais à direita de AC.
	00001000	SUB M(X)	Subtrai M(X) de AC; armazena o resto em AC.
	00001011	MUL M(X)	Multiplica M(X) por MQ; armazena os bits mais significativos do resultado em AC, armazena os bits menos significativos em MQ.
	00001100	DIV M(X)	Divide AC por M(X); armazena o quociente em MQ e o resto em AC.
	00010100	LSH	Multiplica o acumulador por 2 (isto é, desloca os bits uma posição para a esquerda).
	00010101	RSH	Divide o acumulador por 2 (isto é, desloca os bits uma posição para a direita).

NOME/MATRÍCULA:

1ª QUESTÃO)

(a) INTERPRETAÇÃO; TRADUÇÃO. IMPLEMENTADA PELA CONVERSÃO DE TODAS AS INSTRUÇÕES DO CÓDIGO PARA INSTRUÇÕES DO NÍVEL 'INFERIOR'. APÓS O QUE, SE REALIZA A EXECUÇÃO NO NÍVEL DE DESTINO.

(b) NÍVEL DO 'SET' DE INSTRUÇÕES DA ARQUITETURA (ISA). COMPRENDE AS INSTRUÇÕES DO PROCESSADOR.

(c) NÍVEL DE LÓGICA DIGITAL. SOMADOR BINÁRIO, MULTIPLEXADOR, ...

(d) PROGRAM COUNTER (PC). ARMAZENA O ENDEREÇO DA PRÓXIMA INSTRUÇÃO E É INCREMENTADO DE 1 UNIDADE AUTOMATICAMENTE.

(e) OPCODE = CÓDIGO DE INSTRUÇÃO DO 'SET' DE INSTRUÇÕES DO PROCESSADOR.

$$\text{TAM}(\text{OPERAANDO}) = 20 - 8 \text{ bits} = 12 \text{ bits}$$

$$3 \text{ bits}/\text{campo} \Rightarrow 12/3 = 4 \text{ bits}$$

$$12 \text{ bits} \Rightarrow 0x\text{FFF}$$

(f) $\text{TAM}(\text{OPERAANDO}) = 10 \text{ bits} \Rightarrow E_{\text{MAX}} = 1023$

$$\text{Nº LINHAS} = 1023 - 336 + 1 = 688$$

$$\begin{array}{ccccccccc} 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ \downarrow & \downarrow \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{array} = \begin{array}{c} 256 \\ 64 \\ 16 \\ 336 \end{array}$$

$$\text{Nº max bytes} = 688 \times 2B = 1376B$$

(g) CONVERSÃO DO CÓDIGO DA INSTRUÇÃO NO CONJUNTO DE SÍMBOLOS DE CONTROLE PARA EXECUTÁ-LA NO NÚCLEO DE MICROARQUITETURA.

NOME/MATRÍCULA:

① MQ, Acc. Implementando os bits necessários das operações lógicas e aritméticas

$$\textcircled{2} \quad T_{\text{TF}} \text{ bytes} = 1374 \text{ B} \\ t = 2 \text{ mseg} \quad \Rightarrow \quad 687 \text{ GB/s}$$

2ª QUESTÃO)

JUMP M(x, 0:19)

LOAD M(x) 0x00B, ADD M(x) 0x00C
ADD M(x) 0x00D, ADD M(x) 0x00E
ADD M(x) 0x00F, ADD M(x) 0x010
SUS M(x) 0x011, LSH
STOR M(x) 0x012

EM HEXADECIMAL,

0x0DXXX
0x0A00B0500C
0x0500D0500E
0x0500F05010
0x0601114XXX
0x21012

(nº instruções) = 10 x 0.01 mseg = Texe

Texe = 0,1 mseg

TDM C5240 = 5 x 40 bits } 480 bits =
TDM D5205 = 7 x 40 } = 60 B

3ª QUESTÃO)

② 0x000 a 0xFFFF
-2048 + 2047

⑤ $0x37A = 001101111010 = +890$
 $b_1 = 110010000101$
 $b_2 = 110010000110 = -890$

c) 5 bits, bit signal $\Rightarrow -15 \text{ a} + 15$
 , 1060 resolution $\Rightarrow 10^{-15}$

$$T_{\Delta H}(\text{R2R}) = 12 - 5 \text{ bits} = 7 \text{ bits}, 1060 \xrightarrow{-63} \text{a} \xrightarrow{+63} 0x7F \quad 0x3F$$

$$-35 \begin{array}{c} \equiv \\ d \end{array} \begin{array}{c} 1100011 \\ \xrightarrow{\text{s}} \quad \xrightarrow{6 \text{ bits}} \\ (\rightarrow) \quad \xrightarrow{35} \end{array} \equiv 0x63$$

4ª Questão)

$$\begin{array}{r} 101101 \\ - 01000 \\ \hline 00110101 \\ \hline 0x35 \end{array}$$

$$\begin{array}{r} b) \\ \begin{array}{r} 00010111 \\ - 00000111 \\ \hline 00010111 \\ + 11111001 \\ \hline 00010000 \\ \hline 0x10 \end{array} \end{array}$$

$$\begin{array}{r} c) \\ \begin{array}{r} 00010111 \\ - 00000111 \\ \hline 00010111 \\ + 11111000 \\ \hline 00001111 \\ \hline 0x10 \end{array} \end{array}$$