

Q1. 0,7/1 ✓

Q4. 0,1/2 ✓

Q2. 0,2/2 ✓

Q5. 0,7/1 ✓

Q3. 0,6/1 ✓

Q6. 1,1/3 ✓

34

EA 869 - Turma U - 1. Semestre 2006.

Prova 3 - 26/06/2005 - Prof. Léo Pini Magalhães

(com consulta a 1 folha A4 que não pode ser fotocópia - assine a sua folha)

Nome: Pedro Paulo Chaim CelsoNúmero: 035262

Q1. (1,0) Explique com suas palavras o processo de atendimento de uma interrupção originada em um alarme. Descreva desde um instante anterior à ocorrência do sinal do alarme até um instante após o atendimento da interrupção e retorno ao programa interrompido. A interrupção é habilitada por uma máscara de bits e o vetor de interrupção contém 2 posições por linha de interrupção.

O programa estará sendo executado normalmente até o momento em que se der a interrupção neste momento os valores de PC e PSW são armazenados na pilha, e quando do retorno, seja possível retornar ao mesmo ponto e com o mesmo estado de programa.

Enquanto a interrupção estiver sendo tratada esses valores ficam armazenados na pilha, e os registradores recebem valores novos de acordo com a rotina de serviço que está sendo tratada.

Por esse motivo que o vetor de interrupção é da forma abaixo:

O retorno ---  
-0,1



-0,1 - máscara de atendimento  
-0,1 - atendimento ao término da instr. corrente

Q2. (2,0) Você utiliza um processador com uma arquitetura para controle de interrupção que controla 3 linhas de interrupção (L1 a L3) sendo o controle de atendimento por nível (0 é o nível de menor prioridade). As linhas L1 a L3 tem a si associados os níveis 1 a 3 respectivamente. Os bits 0 e 1 da PSW armazenam a prioridade do programa corrente. Programe o processador de forma a atuar como a seguir:

- a rotina de serviço Rot3 poderá ser interrompida por qualquer linha;
- a rotina de serviço Rot2 poderá ser interrompida pelas linhas L3 e L2;
- a rotina de serviço Rot1 só poderá ser interrompida por L3;
- O programa principal só poderá ser interrompido por L3.

O programa principal inicia em 4000H, as rotinas em: Rot1 3000H, Rot2 1000H, Rot3 2000H. O esquema de interrupção é vetorizado associando duas posições a cada linha de interrupção, a primeira contendo Nova-PSW e a outra End-rotina. O vetor de interrupção inicia na posição 6000H.

a. (0,5) Forneça as instruções geradas pela CPU ao aceitar interrupção e iniciar o seu tratamento.

PUSH PC

PUSH PSW

JUMP

RS

MOVE 6000, PSW

JUMP (6000)

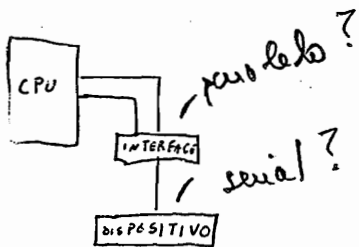
-0,15

-0,15

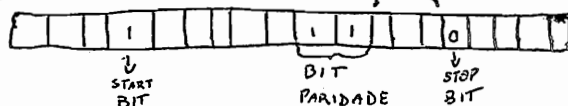
b. (1,5) Escreva um código (use ORG, DW, MOVE) para definir todos os elementos acima descritos com valores que determinem o comportamento descrito (vetor de interrupção, Rot1, Rot2, Rot3, programa principal).

Q3. (1,0)

(a) (0,5) Considere uma ligação através de uma interface serial assíncrona entre uma CPU e um dispositivo. Descreva sucintamente, representando esquematicamente, o relacionamento entre os 3 elementos citados. Descreva/esquematize a troca de dados, controle e endereçamento entre os 3 elementos.



A troca de dados é feita com caracteres no seguinte formato:



Os dados entram ao partir do dispositivo um à um e vão sendo enviados para a CPU, através da interface, esta por sua vez vai realizando o controle junto a CPU para saber se um novo dado pode ser enviado.

No caso oposto quando os dados partem da CPU a interface realiza o controle de disponibilidade junto ao dispositivo.

(b) (0,5) Explique como se dá a comunicação serial entre interface e dispositivo na forma síncrona.

Na forma síncrona as informações são passadas uma a uma de modo sincronizado e dado através de relógios que estão ajustados igualmente de modo que uma nova informação não parta da sua origem antes que a anterior tenha chegado a seu destino.

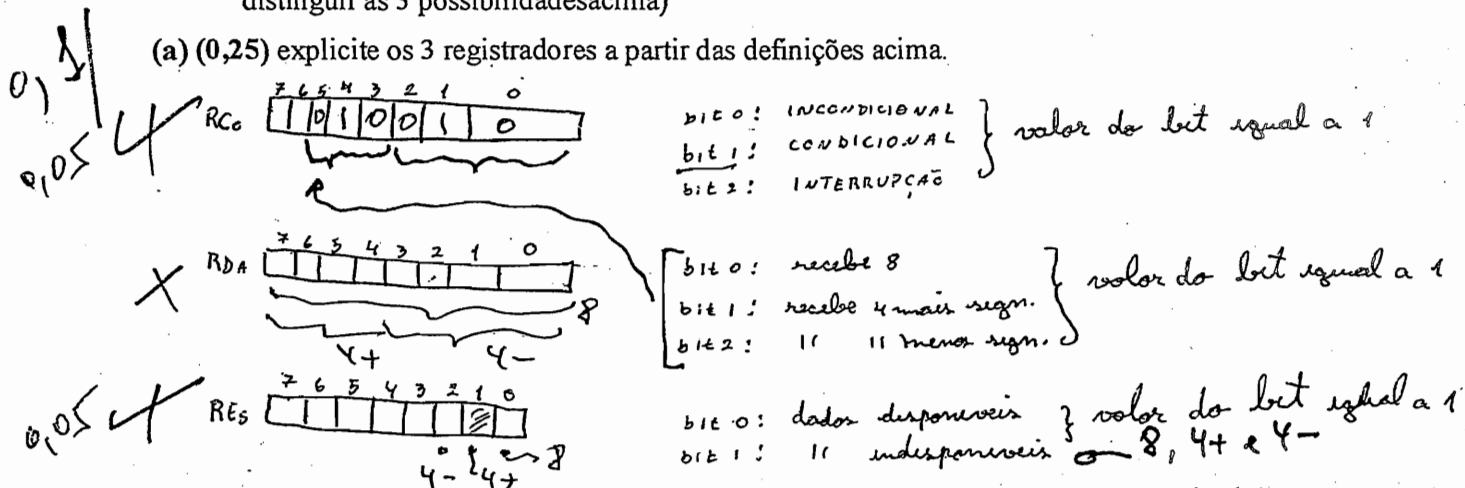
caracter de sincronismo!

Q4. (2,0) Você quer realizar entrada de dados paralela, 4 bits, através da interface XyF++. Esta interface possui 3 registradores de operação (1 byte cada): RCo ou registrador de Controle, REs ou registrador de estado e RDa ou registrador de dados.

**Definição da operação de XyF++:**

- 3 modos de operação: entrada em modo incondicional, condicional e por interrupção
- 3 configurações de entrada: RDa recebe 8 bits de entrada e RDa recebe 4 + 4 bits de entrada (2 entradas independentes de 4 bits)
- bits de REs com valor "1" sinalizam dado disponível em RDa (possibilidade de distinguir as 3 possibilidades acima)

(a) (0,25) explicita os 3 registradores a partir das definições acima.



(b) (0,75) programe a interface para operar com entrada em modo condicional de 4 bits (os 4 bits mais significativos de RDa). Considere arquitetura de E/S mapeada e máquina de 2 endereços. Chame os registradores da CPU por R1, R2, etc.

Handwritten assembly code: `MOVE #010010, RCo`

(c) (1,0) codifique a rotina para realizar a entrada de dados obedecendo (a) e (b).

Handwritten assembly code for the routine:

```

MOVE #010010, RCo
LOOP: MOVE REs, R1
      TSTBIT 1
      JZ LOOP
      MOVE RDa, R2
  
```

Q5. (1,0) Considere a representação em ponto flutuante normalizada com 1 bit para sinal, 6 bits para mantissa e 4 bits para sinal (em complemento de 1 e base 2).

E/S

$$3,49 = 11,0111$$

$$10,0101$$

0,2 a) (0,5) Dê a representação dos números +3,49 e +2,36 e apresente o resultado da soma dos mesmos em ponto flutuante e sua transformação para decimal. Mostre seu raciocínio e comente a questão da precisão.

$$3,49_{(10)} \rightarrow \begin{array}{|c|c|c|c|c|c|c|c|} \hline & 1 & 1 & 0 & 1 & 1 & & \\ \hline 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline \end{array} \times$$

$$2,36_{(10)} \rightarrow \begin{array}{|c|c|c|c|c|c|c|c|} \hline & 1 & 0 & 0 & 1 & 0 & 1 & \\ \hline 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ \hline \end{array} \times$$

$$3 \rightarrow 010,100$$

$$\begin{array}{r} 3,49 \\ + 2,36 \\ \hline 5,85 \end{array}$$

$$\begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ \hline \hline & 0 & 1 & 1 & 1 & 0 & 0 & & & & & \\ \hline \end{array}$$

$$\begin{array}{r} 0,36 \\ \times 2 \\ \hline 0,72 \\ \times 2 \\ \hline 1,44 \\ \times 2 \\ \hline 2,88 \\ \times 2 \\ \hline 5,76 \end{array}$$

Os valores se mostram coerentes dada a sua proximidade; a discrepância se deve ao tamanho da mantissa ser reduzido.

(b) (0,5) Qual a precisão da mantissa e qual a precisão da representação para expoente igual a  $2^{-2}$ ?

$$\text{PRECISÃO DA MANTISSA} = 2^{-6} = \boxed{2^{-6}}$$

$$\text{PRECISÃO DA REPRESENTAÇÃO} = 2^{-6} \cdot 2^{-2} = \boxed{2^{-8}}$$

Q6. (3,0) Considere o seguinte programa em linguagem Assembly:

```

ORG      1050H      ; H define um número em hexadecimal
1050 start: MOVE    x1, R0
1053      INCR      R0
1054 aqui: JZ       start
1057      STOP
1058 x4:   DW       6H
1058 x1:   EQU      x4
105A x5:   ADR      x1
      END
  
```

Considere as Tabelas TIM (Tab. de Instr. de Máquina) e TPI (Tab. de Pseudo-instruções):

TIM			TPI	
Mnemônico	CO - hexad.	Compr.(bytes)	Mnemônico	Compr. (bytes)
MOVE ,R0	45	3	ORG	--
INCR R0	52	1	EQU	--
JZ	6A	3	ADR	-- 2
STOP	B2	1	DW	2
SUB R1	65	1	END	--

0,6

(a) (1,0) Realize o passo1 do montador apresentado em aula e apresente a Tabela de Símbolos. Mostre seus passos.

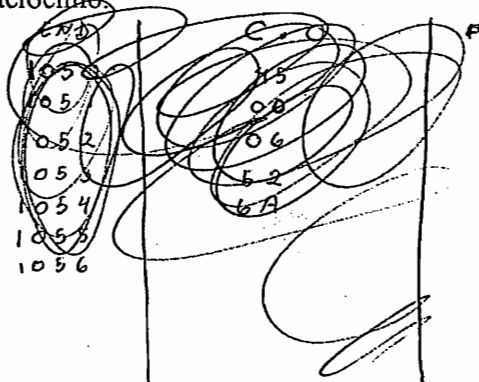
TABELA DE SÍMBOLOS	
RÓTULO	END.
START	1050
AQUI	1054
X4	1058
X1	105A X
X5	105E X

Esta tabela foi construída através da observação do programa e consulta às tabelas TIM e TPI para saber o ~~valor~~ tamanho em bytes que cada instrução ocupa.

ou

0,5

(b) (1,0) Realize o passo2 do montador apresentado em aula e apresente o código de máquina com os endereços finais e nas posições adequadas de memória. Mostre o seu raciocínio.



END.	C.O.	CONTEÚDO
1050	45	40
1051		5A
1052		
1053	52	1058
1054	6A	
1055		40
1056		50
1057	02	
1058		00
1059		06
105A		00 10
105B		06 X 58
105C		10 X X
105D		5A X X

? X - 0,166

- 0,166

- 0,166

0

(c) (1,0) Modifique o código fonte apresentado de forma a ser possível o seu processamento por um montador de 1 passo? Justifique brevemente.

```

ORG 1050 H
START: MOVE 6H, R0
        INCR R0
AQUI: JZ START
        STOP
X4: DW 6H
X1: EQU X4
X5: ADR X1
END
  
```

H define um número em hexadecimal

X esta alteração deve ser feita pois no montador de um passo as variáveis devem ser declaradas antes do seu uso no programa.

7