EA 869 A – Introdução a Sistemas de Computação Digital
Prova 3 – 23/06/2010 – Prof. Marco A. A. Henriques

Obs:

- escolha e resolva apenas 4 das 5 questões; se forem resolvidas 5 só serão corrigidas as 4 primeiras;
- prova sem consulta;
- se precisar, use o verso das folhas, indicando o número da questão e do item;
- não é permitido retirar o grampo das folhas;
- a interpretação das questões faz parte da avaliação.

RA: _____ Nome: _____ Assin.:_____

Questão	Q 1	Q 2	Q3	Q4	Q5	Total
Nota						

Questão 1 (2,5) Seja uma interface serial de dados.

(a) (0,5) Defina um registrador de 1 byte na interface de forma que ela possa ser configurada para operar como entrada ou saída, com 1 ou 2 StopBits, com 1 ou 2 StartBits, transmitindo ASCII de 7 ou 8 bits, usando ou não interrupção e operando em 2 baud-rates (taxas de transmissão) diferentes (19200 bps ou 115000 bps). O registrador deverá desempenhar papel duplo: registrador de controle (usando seus bits menos significativos) e de estado (usando os outros bits).

(b) (1,0) Mostre um trecho de programa em Assembly que configure a interface acima definida e realize uma saída condicional de um conjunto de muitos bytes a 115000 bps, com 2 StopBits, 1 StartBit, ASCII de 8 bits, sem interrupção, considerando que a CPU oferece uma arquitetura de E/S mapeada em memória e é uma máquina de 2 endereços.

(c) (1,0) Mostre um trecho de programa em Assembly que configure a interface acima definida e realize uma entrada condicional de um conjunto de muitos bytes a 19200 bps, com 1 StopBit, 2 StartBits, ASCII de 7 bits, sem interrupção, considerando que a CPU oferece uma arquitetura de E/S isolada e é uma máquina de 2 endereços.

Questão 2 (2,5)

(a) (0,5) Explique quais são as diferenças entre entrada/saída isolada e entrada/saída mapeada em memória, destacando pelo menos duas características que as diferenciam.

(b) (1,0) Em entrada e saída serial de dados, a transferência pode ser síncrona ou assíncrona. Mostre, em um gráfico tempo X bit, como seria a transmissão dos bits correspondentes aos valores hexadecimais E, A, 8, 6, 9, A (4 bits cada), nesta sequência, tanto na transferência síncrona como na assíncrona. Considere que para cada valor de 4 bits são enviados os bits de controle necessários. Considere ainda que as transmissões se baseiam em 1 start bit, 2 stop bits, sem bit de paridade e que o caractere de sincronismo é formado pelos valores A e 5, nesta ordem.

(c) (0,5) Apresente e justifique duas razões para se separar as etapas de ligação e de carregamento de um programa?

(d) (0,5) Explique o que são e aponte as diferenças entre ligação dinâmica em tempo de carregamento e ligação dinâmica em tempo de execução.

Questão 3. (2,5) Considere o seguinte programa em linguagem Assembly e as tabelas TIM e TPI (parciais).

```
        ORG     $809C       ; $809C – valor em hexadecimal
cont:   DW      2           ; inicia a variável "cont"
inicio: MOVE    cont, D1    ; D1 ← (cont)
        ADDI    #3, D1      ; D1 ← (D1) + 3
        MOVE    D1, saida   ; saida ← (D1)
        JUMP    fim         ; ir para fim
saida:  DS      3           ; define espaço para a variável saida
fim:    STOP                ; parar a execução
```

TIM

Mnemônico	C.O. (hexadec.)	Compr.(bytes)
ADDI , D1	C0	2
MOVE , D1	B3	3
MOVE D1,	A3	3
JUMP	5D	3
STOP	D0	1

TPI

Mnemônico	Compr.(bytes)
ORG	- -
DW	2
DS x	2 * x

(0,5) É necessário montar este programa com um montador de dois passos? Por que?

(1,0) Mostre a tabela de símbolos gerada por um montador de dois passos, apresentando os cálculos usados para obtê-la.

(1,0) Produza o programa em linguagem de máquina do programa fonte acima e mostre como o mesmo ficaria carregado na memória (endereços e seus conteúdos), destacando os cálculos usados.

Questão 4 (2,5) Para trabalhar com um Carregador Absoluto, o programa Montador faz a montagem do código assembly e gera um arquivo (módulo objeto) com várias linhas, tendo cada uma os seguintes campos separados por espaços:

- campo 1: tipo (0 para texto; 1 para fim)
- campo 2: endereço de carregamento (se tipo=0) ou de execução (se tipo =1)
- campo 3: comprimento em bytes
- campo 4: conteúdo binário (em hexadecimal)

Faça um fluxograma mostrando como deve funcionar um Carregador Absoluto capaz de realizar as funções de carregamento e execução do código após carregado. O fluxograma deve ser detalhado o suficiente para permitir que se acompanhe cada etapa de seu funcionamento.

Questão 5 (2,5) Um programa em linguagem Assembly é formado por dois segmentos distintos descritos abaixo. Após a etapa de montagem, o montador de ligação direta produziu os respectivos arquivos objeto também descritos. Para executar o programa, é chamado um carregador de ligação direta, que recebe do sistema operacional um endereço inicial de carga (IPLA) igual a $00 00 09 F0 e os dois arquivos objeto na ordem apresentada abaixo. Os fluxogramas relativos ao carregador de ligação direta estão disponibilizados na próxima página.

Segmento PROG_A

```
PROG_A:     MOVE.B   VAR_1, D0
            MOVE.B   D0, VAR_2
            RTS
VAR_2:      DS.W     1
            END
```

Arquivo PROG_A.obj

```
ESD.PROG_A.SD.0000.0010
TXT.00.6.103900000000
TXT.06.6.13C00000000E
TXT.0C.2.4E75
TXT.0E.2.0000
RLD.02.4.VAR_1
RLD.08.4.PROG_A
END
```

Segmento PROG_B

```
            GLOB     VAR_1
PROG_B:     MOVE.W   #1, VAR_1
            JSR      PROG_A
            RTS
VAR_1:      DS.W     1
            END      PROG_B
```

Arquivo PROG_B.obj

```
ESD.VAR_1.LD.0010.0002
ESD.PROG_B.SD.0000.0012
TXT.00.8.31FC000100000010
TXT.08.6.4EB900000000
TXT.0E.2.4E75
TXT.10.2.0000
RLD.04.4.PROG_B
RLD.0A.4.PROG_A
END.00
```

obs.: "." é usado para separar campos em *.obj.

(a) (0,5) Apresente a tabela GEST gerada pelo carregador, justificando o cálculo realizado para obter cada elemento da tabela.

(b) (1,5) Mostre na tabela abaixo os endereços e os códigos efetivamente carregados e já ajustados.

endereço	0 - 1	2 - 3	4 - 5	6 - 7	8 - 9	A - B	C - D	E - F
0000 09F								

(c) (0,5) Quais endereços sofreram ajustes de relocação e quais sofreram ajustes de ligação? Justifique suas respostas.