Arquitetura de Computadores
2006/2007 2º Semestre
Repescagem 1º Teste - 11/07/2007

Número: _____ Nome: _____

INSTRUÇÕES:

- A duração da prova é de 1,5 horas.
- Responda apenas nesta “Folha de Respostas”; nada mais será recebido.
- Identifique esta folha com o seu número e nome de forma bem legível.
- A cotação das perguntas está indicada entre parênteses à direita do texto.
- Nas perguntas Verdade/Falso ou de escolha múltipla as respostas erradas descontam.

Folha de Respostas

Grupo I

	R1	R2	R3	R4	R5	R6	R7	PC	SP	RE	
1. a)			2A7Eh					015Bh		0004h	(1)
1. b)		560Eh						015Ah		0004h	(1)
1. c)								0230h	FCAFh		(1)

1. d) (2 valores)

	Endereço	Dados	L / E
1	0159h	200Dh	L
2	FCB0h	0006h	E
3	FCAFh	015Ah	E
4	FE0Dh	?	L
5			

Grupo II

(A-F)

2.1 a)	A	(0,5)
2.1 b)	B	(2)

2.2 (1 valor)

1	ADD R1, const
2	MOV R2, M[R1]
3	INC R2
4	MOV M[R1], R2

	Linha	Instrução			Linha	Instrução	
2.3 a)	4	BR NEXT	(1)	2.3 b)	7	CMP R1, R0	(0,5)

(A-E)

2.3 c)	D	(1)
---------------	---	-----

Grupo III

3.1	Valor	
a)	0000 h	(0.5)
b)	3212 h	(0.75)
c)	001C h	(0.75)
d)	0018 h	(0.5)

(A-E)		
3.2	B	(1.5)

Grupo IV**4.1 (3 valores)**

Instrução (1)	
Endereço	Código Objecto
0333h	8270h
0334h	805Ah

Instrução (2)	
Endereço	Código Objecto
0335h	C520h
0336h	0323h

4.2 (2 valores)

Endereço	Instrução (Assembly)	Código Objecto
0335h	BR.N Again	E52Dh

Nota: Valor do desconto em caso de resposta errada:

- Perguntas de escolha múltipla: $C / (N-1)$ (C=Cotação; N=Número de alternativas)
- Perguntas Verdade/Falso: $C / 2$ (C=Cotação)

I

1. Considere os seguintes valores para os registos do processador P3:

R1	R2	R3	R4	R5	R6	R7	PC	SP	RE
0431h	AAC1h	BC29h	5784h	0230h	A43Bh	6812h	0159h	FCB0h	0006h

Para as alíneas a), b) e c), indique na folha de respostas qual o novo valor, em **hexadecimal**, para todos (**e apenas**) os registos que são alterados pelo ciclo completo de execução de cada instrução. Use o símbolo ? caso não tenha informação suficiente para determinar o novo valor de um registo.

As alíneas são independentes, isto é, assuma como valores iniciais para cada alínea os indicados na tabela acima.

1. a) XOR R3, 9657h (1)
1. b) ROLC R2, 3 (1)
1. c) CALL.NZ R5 (1)

1. d) Para o ciclo completo de execução da instrução

INT 13

indique na folha de respostas a sequência de acessos à memória, especificando o endereço, dados e tipo de acesso (**Leitura/Escrita**). Considere que a tabela de interrupções se encontra definida a partir do endereço **FE00h** e que a máscara correspondente à interrupção 13 está activa. (2)

Nota 1: A tabela tem 5 linhas, utilize apenas as que considerar necessárias.

Nota 2: Utilize os valores iniciais dos registos indicados na tabela no cimo desta página.

Nota 3: Use ? para indicar que não tem informação suficiente para determinar um dado valor.

II

2.1 Vamos avaliar as características de geração de código de 4 compiladores de C para o P3. Para tal compilou-se o programa seguinte tendo sido produzidos os resultados indicados na tabela.

Considere que não há *overflows* ou *underflows* nas operações aritméticas.

```
#define SIZE 64
register short int i;
short int A[SIZE],B[SIZE];

for (i=0;i<SIZE;i=i+1)
  if (A[i]==0)
    B[i]=B[i]/2;
  else
    B[i]=2*B[i];
```

A:	1	SIZE	EQU	64	B:	1	SIZE	EQU	64
	2	A	TAB	SIZE		2	A	TAB	SIZE
	3	B	TAB	SIZE		3	B	TAB	SIZE
	4	INIT:	MOV	R1,SIZE		4	INIT:	MOV	R1,SIZE
	5		DEC	R1		5		DEC	R1
	6		MOV	R2,1		6	FOR:	CMP	R0,M[R1+A]
	7	FOR:	CMP	R0,M[R1+A]		7		BR.NZ	ELSE
	8		BR.NZ	ELSE		8		DIV	M[R1+B],2
	9		SHRA	M[R1+B],R2		9		BR	NEXT
	10		BR	NEXT		10	ELSE:	MUL	2,M[R1+B]
	11	ELSE:	SHLA	M[R1+B],R2		11	NEXT:	DEC	R1
	12	NEXT:	DEC	R1		12		BR.NN	FOR
	13		BR.NN	FOR		13			
C:	1	SIZE	EQU	64	D:	1	SIZE	EQU	64
	2	A	TAB	SIZE		2	A	TAB	SIZE
	3	B	TAB	SIZE		3	B	TAB	SIZE
	4	INIT:	MOV	R1,SIZE		4	INIT:	MOV	R1,SIZE
	5		DEC	R1		5		DEC	R1
	6	FOR:	CMP	R0,M[R1+A]		6	FOR:	MOV	R2,2
	7		BR.NZ	ELSE		7		CMP	R0,M[R1+A]
	8		SHRA	M[R1+B],1		8		BR.NZ	ELSE
	9		BR	NEXT		9		DIV	M[R1+B],R2
	10	ELSE:	SHLA	M[R1+B],1		10		BR	NEXT
	11	NEXT:	DEC	R1		11	ELSE:	MUL	R2,M[R1+B]
	12		BR.NN	FOR		12	NEXT:	DEC	R1
	13					13		BR.NN	FOR

2.1 a) Indique, na folha de respostas, como é que se designa o modo de endereçamento dos vectores usados nos programas. **(0,5)**

A: Indexado	B: Por Registo.	C: Por Registo Indirecto.	D: Directo.	E: Baseado.	F: Nenhum dos anteriores.
--------------------	------------------------	----------------------------------	--------------------	--------------------	----------------------------------

2.1 b) Indique na folha de respostas que programas produzem um resultado correcto. (2)

- A:** Programas B e D.
B: Programas C e D.
C: Programas A e C.
D: Programas A e B.
E: Todos os programas são correctos.
F: Nenhuma das respostas anteriores está correcta.

2.2 Considere agora uma versão muito simples da arquitectura P3 – *pico-P3*. O *pico-P3* executa um subconjunto das instruções do P3: não tem instruções MUL nem DIV; as instruções aritméticas e lógicas só operam com registos; só os MOV acedem a variáveis na memória – MOV op1,op2 em que op = { Const. | Reg. | M [Reg.] }.

Complete o seguinte troço de código do *pico-P3* por forma a realizar a mesma função que a instrução INC M [R1+const] do P3. (1)

1	ADD R1, const
2	
3	INC R2
4	

← Rascunho; preencher na folha de respostas

2.3 Como o *pico-P3* não possui multiplicação pretende-se usar o programa seguinte para multiplicar R1 por R2, ficando o resultado em R2. Considera-se que os operandos são inteiros não negativos e o resultado é um inteiro de 16 bits (o valor dos operandos é tal que não provoca *overflow*).

2.3 a) Durante o teste do programa verificou-se que, para certos valores dos operandos, o programa não produz resultados correctos. Admita que o programa pode ser corrigido alterando uma só instrução. Indique na folha de respostas a linha do programa e a instrução correcta. (1)

```

1  MUL:  CMP    R1, R0
2         BR.NZ NZ
3  ZERO: MOV    R2, R0
4         BR    NZ
5
6  NZ:   DEC    R1
7         CMP    R1, R0
8         BR.Z  NEXT
9         MOV    R3, R2
10 SUM:  ADD    R2, R3 ; R2 acumulador
11         DEC    R1
12         BR.P  SUM
13 NEXT:
```

2.3 b) No programa há uma instrução que pode ser eliminada sem alterar a funcionalidade do programa. Identifique a linha e a instrução na folha de respostas. (0,5)

No programa de multiplicação inseriram-se as seguintes instruções no início do programa (antes da linha1).

```
CMP    R1, R2
BR.N   MUL
XCH    R1, R2
```

2.3 c) O que se pretendeu com esta modificação ao programa? (1)
(Assuma que a incorrecção referida em 2.3 a) foi corrigida.)

- A: Evitar o ocorrência de *overflow* no cálculo do produto.
- B: Testar se algum dos operandos é negativo, caso em que o algoritmo original não produz resultado correcto.
- C: Testar se os operandos são iguais caso em que o cálculo pode ser optimizado.
- D: Reduzir o tempo médio de cálculo do produto.
- E: Nenhuma das respostas anteriores está correcta.

III

3.1 Considere a seguinte rotina de tratamento de uma interrupção:

```
1  Intr:  MOV R1, 4
2         OR R1, M[SP+2]
3         RTI
```

Assuma que a interrupção ocorreu imediatamente após executar a instrução “CMP R0, 0” localizada no endereço 3210 h.

Responda na folha de respostas às questões que se seguem. Se não for conhecido um dado valor use o símbolo ‘?’.

- a) Qual o valor do RE após executar a instrução da linha 1. (0,5)
- b) Qual o conteúdo da posição de memória M[SP+1]. (0,75)
- c) Qual o conteúdo de R1 após executar a instrução da linha 2. (0,75)
- d) Após executar RTI qual o valor do RE. (0,5)

3.2 No sistema do P3 para permitir a leitura da tecla 5 (I5) por interrupção é necessário efectuar as seguintes acções: (1,5)

- A - Inicializar o SP, colocar em FE20h o endereço da rotina de tratamento da interrupção, usar 0005h como máscara de interrupções.
- B - Inicializar o SP, colocar em FE05h o endereço da rotina de tratamento da interrupção, usar 0020h como máscara de interrupções e evocar ENI.
- C - Inicializar o SP, colocar em FE05h o endereço da rotina de tratamento da interrupção, usar 0005h como máscara de interrupções e evocar ENI.
- D - Colocar em FE05h o endereço da rotina de tratamento da interrupção, usar 0020h como máscara de interrupções e evocar ENI.
- E - Nenhuma das alternativas anteriores é correcta.

IV

4.1 Considere o seguinte troço de código em Assembly do P3:

```

ASCII_A    EQU    'A'
ASCII_0    EQU    '0'

                ORIG    8059h
MASK        WORD    000Fh
DATA_SIZE   WORD    4
DATA        STR    1A3Ch, 2468h, 357Ah, 1592h

                ORIG    0320h
                MOV     R1, R0
                MOV     R2, M[MASK]
Again:      MOV     R3, M[R1+DATA]
                AND     R3, R2
                CMP     R3, 9
                BR.P    Letter
Number:     ADD     R3, ASCII_0
                BR     Store
Letter:     SUB     R3, 10
                ADD     R3, ASCII_A
Store:     MOV     M[R1+DATA], R3
                INC     R1
                CMP     R1, M[DATA_SIZE]    ; (1)
                JMP.N  Again                ; (2)
End:       BR     End

```

Traduza as instruções Assembly assinaladas com os símbolos (1) e (2) para o código objecto correspondente, indicando o endereço de memória e o respectivo conteúdo em hexadecimal. **(3)**

Nota 1: Tenha em atenção a base de representação das constantes.

Nota 2: Na folha de respostas preencher apenas as linhas necessárias.

4.2 A instrução (2) do código apresentado acima pode ser facilmente alterada de forma a melhorar o desempenho do programa. Preencha a tabela da folha de respostas, indicando a instrução alternativa que conduz ao mesmo resultado e qual a sua codificação. **(2)**