

Arquitetura de Computadores I
1ª série de problemas – Soluções
13.10.2014

1. O que é um circuito integrado? Que impacto teve a tecnologia integrada no desenvolvimento dos computadores? Que geração de computadores é caracterizada pela utilização de circuitos integrados?

Um circuito integrado (CI) é um circuito realizado num substrato monocristalino de material semiconductor. O material semiconductor mais usado na realização de CIs é o silício.

A tecnologia integrada permitiu uma redução muito significativa das dimensões e do preço dos computadores, estando associada ao aparecimento dos “minicomputadores”. A utilização de circuitos integrados caracteriza a 3ª geração de computadores digitais.

2. Qual a tecnologia usada pela 1ª geração de memórias de semicondutores de grande escala de integração (LSI)? Qual a empresa que as introduziu no mercado?

Tecnologia NMOS. Usada nas primeiras memórias de 1kbits, introduzidas pela Intel em 1971.

3. Por que é que John von Neumann é considerado o “pai” dos computadores digitais?

John von Neumann é creditado como sendo o criador do modelo de computador com programa armazenado em memória, modelo comum a todos os computadores digitais.

4. Descreva em assembly do MIPS o cálculo da seguinte expressão: $x = x + y + z - q$;
Assuma que **x, y, z, q** estão armazenados nos registos **\$s1-\$s4**.

```
add $t0,$s1,$s2
add $t0,$t0,$s3
sub $s1,$t0,$s4
```

5. Escreva em assembly do MIPS, duas versões, uma sem utilizar ponteiros e a outra recorrendo a ponteiros, do seguinte segmento de código C:

```
int A[100], B[100];
for (i=1; i < 100; i++)
{
    A[i] = A[i-1] + B[i];
}
```

No início os únicos valores em registos são os endereços base dos arrays A e B nos registos **\$a0** e **\$a1**. Não utilize instruções de multiplicação.

Sem recurso a ponteiros:

	move \$t0, 1	# i = 1
loop1:	slti \$t7,\$t0, 100	# \$t7 = (i < 100)
	bge \$t7, \$zero, endloop	# if (i >= 100) terminar ciclo
	sll \$t1, \$t0, 2	# \$t1 = i * 4
	addu \$t2, \$a0, \$t1	# \$t2 = &A[i]
	addu \$t3, \$a1, \$t1	# \$t3 = &B[i]
	lw \$t4, -4(\$t2)	# \$t4 = A[i-1]
	lw \$t5, 0(\$t3)	# \$t5 = B[i]
	add \$t6, \$t4, \$t5	
	sw \$t6, 0(\$t2)	# A[i] = A[i-1] + B[i]
	addiu \$t0, \$t0, 1	# i = i + 1

```

        j loop1
endloop:

```

Utilizando ponteiros:

```

        addiu $t2, $a0, 4      # $t2 = &A[1]
        addiu $t3, $a1, 4      # $t3 = &B[1]
        addiu $t0, $a0, 400    # $t0 = &A[100] (endereço contíguo ao do ultimo elemento de A)
loop2:  slt $t7, $t2, $t0      # $t2 = (p < &A[100])
        bge $t7, $zero, endloop # if (p >= &A[100]) terminar ciclo
        lw $t4, -4($t2)        # $t4 = A[i-1]
        lw $t5, 0($t3)         # $t5 = B[i]
        add $t6, $t4, $t5
        sw $t6, 0($t2)         # A[i] = A[i-1] + B[i]
        addiu $t2, $t2, 4      # incremento do ponteiro para A
        addiu $t3, $t3, 4      # incremento do ponteiro para B
        j loop2
endloop:

```

6. No código MIPS assembly seguinte, quantas vezes é acedida a memória de instruções? E a memória de dados?

```

lw      $v1, 0($a0)
addi    $v0, $v0, 1
sw      $v1, 0($a1)
addi    $a0, $a0, 1

```

É acedida 2 vezes a memória de dados e 4 vezes a memória de instruções.

7. Use os valores dos registos e da memória indicados na Tabela para responder às questões seguintes. As questões são independentes – em cada uma delas assuma que os valores iniciais são os indicados na Tabela.

Registo	Valor	Endereço de memória	Valor
R1	12	12	16
R2	16	16	20
R3	20	20	24
R4	24	24	28

- a) Quais os valores de R1, R2, e R3 depois de executada a instrução: add R3, R2, R1
R2 = 16, R1 = 12, R3 = 28
- b) Quais os valores de R1 e R3 depois de executada a instrução: load R3, 12(R1)
R1 = 16, R3 = 28
- c) Quais os valores nos registos depois de executada a instrução: addi R2, R3, 16
R3 = 20, R2 = 36

8. A instrução **la** (Load Address) é uma instrução virtual. Qual a tradução em instruções nativas de la \$t0, 0x10010020

```

lui $at, 0x1001      ou      lui $at, 0x1001
ori $t0, $at, 0x0020  addiu $t0, $at, 0x0020

```

9. No MIPS as únicas instruções nativas de **branch** são branch on equal e branch on not equal. Indique como são traduzidas para instruções nativas as seguintes instruções:

- a) bgt \$t0, \$t1, Label
- ```

 slt $at, $t1, $t0
 bne $at, $0, Label

```

- b) ble \$t0, \$t1, Label

```
slt $at,$t1,$t0
beq $at,$0, Label
```