

O S T Q Q S S  
 O L M M J V S

## Exercício Teórico 2

Nathalia Henrique Vieira Pinto

1. ori \$0, \$zero, 10 #a=10  
 addi \$1, \$zero, -1 #b=-1  
 addi \$0, \$0, 1 #a+=1  
 add \$2, \$0, \$1 hc=a+b

3. ori \$0, \$zero, 3 #x=3  
 ori \$10, \$zero, 100 #T0=100S  
 mul \$1, \$0, \$10 #y=x.100S

5. ori \$0, \$zero, 0x1234  
 sll \$0, \$0, 16  
 or \$0, \$0, 0x5678

7. # \$0 = A[0], \$1 = h  
 lw \$T0, 32(\$0) #T0 = A[8]  
 add \$T1, \$1, \$T0 #T1 = h + A[8]  
 sw \$T1, 48(\$0) #A[0] = T1

9. # \$0 = A[0], \$1 = h, \$2 = i, \$3 = j  
 sll \$T0, \$c2, 2 #T0 = i.4  
 sll \$T1, \$S3, 2 #T1 = j.4  
 add \$T2, \$0, \$T0  
 add \$T3, \$0, \$T1  
 lw \$T4, 0(\$T2) #T4 = A[i]  
 add \$T5, \$S1, \$T4 #T5 = h + A[i]  
 sw \$T5, 0(\$T3) #A[i] = T5

2. ori \$0, \$zero, 3 #x=3  
 ori \$10, \$zero, 4 #T0 = 4  
 mul \$1, \$0, \$T0 #y = x.4

4. ori \$0, \$zero, 3 #x=3  
 sll \$S1, \$0, 2 #y = x/4

6. addi \$0, \$zero, -1 #x = -1  
 sra \$T1, \$0, 5 #y = x/32

8. # \$0 = A[0], \$1 = K, \$2 = i  
 sll \$T0, \$c2, 2 #T0 = i.4  
 add \$T1, \$0, \$T0 #T1 = A[i]  
 lw \$T2, 0(\$T1) #T2 = A[i]  
 add \$S3, \$0, \$T2 #h = K + A[i]

10. # \$0 = A[0], \$1 = i  
 sll \$T0, \$S1, 2  
 add \$T1, \$0, \$T0  
 lw \$T2, 0(\$T1) #h = A[i]  
 lw \$T2, 4(\$T1) #T2 = A[i+1]  
 sw \$T2, 0(\$T1) #A[i] = T2  
 sw \$T2, 4(\$T1) #A[i+1] = h

D S T Q Q S S  
D L M M J V S

11. ori \$S0, \$zero, 0 # j = 0  
 ori \$S1, \$zero, 10 # i = 10  
 loop:  
 addi \$S0, \$S0, 1 # j += 1  
 bne \$S1, \$S0, loop # while(j != i)  
 .data

12. addi \$T0, \$zero, 0x1001  
 sll \$T0, \$T0, 16 # T0 = 0x10010000  
 lw \$S0, 0(\$T0) # S0 = A  
 slt \$T1, \$zero, \$S0  
 bne \$T1, \$zero, \$T0 # if (A > 0) \$T0  
 sub \$S0, \$zero, \$S0 # !A  
 Fim:  
 sw \$S0, 0(\$T0) # A = S0  
 .data  
 Ap. word -1

13. addi \$T0, \$zero, 0x1001  
 sll \$T0, \$T0, 16  
 lw \$S0, 0(\$T0) # S0 = TEMP  
 ori \$T1, \$zero, 30 # T1 = 30  
 ori \$T2, \$zero, 50 # T2 = 50  
 slt \$T3, \$S0, \$T1  
 xor \$T3, \$T3, 1 # (T0 = 30) ? 1 : 0  
 slt \$T4, \$T2, \$S0  
 xor \$T4, \$T3, \$T4 # (T0 <= 50) ? 1 : 0  
 and \$TS, \$T3, \$T4 # and T3, T4  
 sw \$TS, 4(\$T0) # FLAG = TS  
 .data  
 TEMP: .WORD 20  
 FLAG: .WORD

14. addi \$T0, \$zero, 0x1001  
 sll \$T0, \$T0, 16  
 ori \$S0, \$zero, 99 # i = N-1  
 ori \$S1, \$zero, 0 # j = 0  
 ori \$T6, \$zero, 100 # N = 100  
 ori \$T6, \$zero, 1 # T6 = 1  
 sub \$T1, \$S2, \$T6 # T1 = N-1  
 loop:  
 loop:  
 sll \$T2, \$S1, 2  
 add \$T2, \$T2, \$T0  
 lw \$T3, 0(\$T2) # T3 = array[i]  
 lw \$T4, 4(\$T2) # T4 = array[j+1]  
 sht \$TS, \$T3, \$T4  
 bne \$TS, \$zero, else # (array[i] > array[j+1])?  
 sw \$T4, 0(\$T2) # SWAP  
 sw \$T3, 4(\$T2) # SWAP  
 else: addi \$S1, \$S1, 1 # j++  
 bne \$S1, \$T1, loop # for(j = 0; j < N-1; j++)  
 sub \$S0, \$zero, \$T6 # i--  
 and \$S1, \$zero, 0 # j = 0  
 bne \$S0, \$zero, loop # for(i = N-1; i > 0; i--)  
 Fim:

D	S	T	O	Q	S	S
0	L	M	M	J	Y	S

15. addi \$t0, \$zero, 0x100

addi \$t0, \$t0, 16

lw \$s0, 6(\$t0) #x=2

addi \$t3,\$zero,1 #T3=1

addi \$t4, \$zero, 1 # T4=1

andi \$t2, \$s0, 1

bne \$t2, \$zero, \$a7 #if(\$a2==0)

impon:

addi \$t1, \$zero, 0 #i=0

addi \$t5, \$zero, 5 #TS=5

loop\_i1:

smul \$t3, \$t3, \$t0 #T3=x^5

addi \$t1, \$t1, 1 #i++

bne \$t1, \$t5, loop\_i1 #for(i=0; i<ts; i++)

addi \$t5, \$zero, 3 # TS=3

addi \$t1, \$zero, 0 #i=0

loop\_i2:

smul \$t4, \$t4, \$s0 #T4=x^3

addi \$t1, \$t1, 1 #i++

bne \$t1, \$t5, loop\_i2 #for(i=0; i<ts; i++)

sub \$s1, \$t3, \$t4 #y=x^5-x^3

addi \$s1, \$s1, 1 #y+=1

JFim # Fim

par:

addi \$t1, \$zero, 0 # i=0

addi \$t5, \$zero, 4 # TS=4

loop\_p1:

mul \$t3, \$t3, \$s0 #T3=x^4

addi \$t1, \$t1, 1 #i++

bne \$t1, \$t5, loop\_p1 #for(i=0; i<ts)

addi \$t5, \$zero, 3 # TS=3

addi \$t1, \$zero, 0 #i=0

loop\_p2:

mul \$t4, \$t4, \$s0 # T4=x^3

addi \$t1, \$t1, 1 #i++

bne \$t1, \$t5, loop\_p2 #for(i=0, i<ts)

add \$s1, \$t3, \$t4 #y=x^4+x^3

smult \$t3, \$s0, \$s0 #T3=x^2

add \$t3, \$t3, \$t3 #T3=2x^2

sub \$s1, \$s1, \$t3 #y=y-T3

JFim

Fim:

sw \$s1 4(\$t0) # memoria

•data

X: .Word 2

D S T O O S S  
O L M M J V S

16. addi \$T0, \$2000, 0x1001

sll \$T0, \$T0, 16

lw \$S0, 0(\$T0) # x=2

ori \$T1, \$2000, 0 # i=0

ori \$T2, \$2000, 1 # T2=1

ori \$S1, \$2000, 1 # S1=1

slt \$T4, \$2000, 1c0

bne \$T4, \$2000, main # if(x>0)

main:

addi \$T3, \$2000, 4 # T4=4

loop1:

mul \$T2, \$T2, \$S0 # T2=x^4

addi \$T1, \$T1, 1 # i++

bne \$T1, \$T3, loop1 # For(i=0, i<T3)

sub \$S1, \$T2, \$S2 # y=x^4-1

J main

main:

addi \$T3, \$2000, 3 # T3=3

loop2:

mul \$T2, \$T2, \$S0 # T2=x^3

addi \$T1, \$T1, 1 # i++

bne \$T1, \$T3, loop2 # For(i=0, i<T3)

addi \$S1, \$T2, 1

J main

main:

sw \$S1, 4(\$T0) # memoria

data

X: .word -2

17. addi \$T0, \$2000, 0x1001

sll \$T0, \$T0, 16

ori \$T1, \$2000, 0 # i=0

ori \$S0, \$2000, 0 # S0=0

ori \$S1, \$2000, 1 # S1=1

ori \$T4, \$2000, 100 # TH=100

loop:

sll \$T3, \$T1, 2 # T5 = i \* 4

add \$T4, \$T3, \$T0 # \*T4 = array[i]

sw \$S1, 0(\$T4)

add \$T2, \$S0, \$S1 # T2 = S1 + S0

add \$S0, \$2000, \$S1 # S0 = S1

add \$S1, \$2000, \$T2 # S1 = T2

addi \$T1, \$T1, 1 # i++

bne \$T1, \$T4, loop # For(i=0, i<100)

# main main da simulador 45

D S T O O S S  
D L M M J V S

18. addi \$T0,\$Zero,0x100  
 sll \$T0,\$T0,16  
 lw \$S0,0(\$T0) #x=60  
 ori \$T1,\$Zero,S0 #A=S0  
 ori \$T2,\$Zero,100 #T2=100  
 slt \$T3,\$S0,\$T1 #(T0>S0)?1:0  
 xor \$T3,\$T3,1 #(T0=S0)?1:0  
 slt \$T4,\$T2,\$S0 #(T0<100)?1:0  
 xor \$T4,\$T4,1 #(T0<100)?1:0  
 and \$T5,\$T3,\$T4  
 or \$T1,\$Zero,150 #T1=150  
 or \$T2,\$Zero,200 #T2=200  
 slt \$T3,\$S0,\$T1 #(T0>150)?1:0  
 xor \$T3,\$T3,1 #(T0>150)?1:0  
 slt \$T4,\$T3,\$S0 #(T0>200)?1:0  
 xor \$T4,\$T4,1 #(T0<200)?1:0  
 and \$T6,\$T3,\$T4  
 or \$T7,\$T6,\$T5  
 sw \$T7,4(\$T0) #FLAG=T5

data

numero: .word 60

flag: .word

19. addi \$T0,\$Zero,0x100+0  
 sll \$T0,\$T0,16  
 lw \$T1,0(\$T0) #A=23  
 lw \$T2,4(\$T0) #B=98  
 lw \$T3,8(\$T0) #C=17  
 slt \$T6,\$T1,\$T2  
 bne \$T6,\$Zero,pxr1 #(A<B)?1:0  
 sw \$T2,0(\$T0) #A=B  
 sw \$T1,4(\$T0) #B=A  
 lw \$T1,0(\$T0)  
 lw \$T2,4(\$T0)  
 pxr1:  
 slt \$T6,\$T2,\$T3  
 bne \$T6,\$Zero,pxr2 #(B<C)?1:0  
 sw \$T3,4(\$T0) #B=C  
 sw \$T2,8(\$T0) #C=B  
 lw \$T2,4(\$T0)  
 lw \$T3,8(\$T0)  
 pxr2:  
 slt \$T6,\$T1,\$T2

bne \$T6,\$Zero,pxr3 #(A<B)?1:0

sw \$T2,0(\$T0)

sw \$T1,4(\$T0)

lw \$T1,0(\$T0)

lw \$T2,4(\$T0)

pxr3:

sw \$T2,12(\$T0)

data

A: .word 23

B: .word 98

C: .word 17

D S T Q Q S S  
D L M M J V S

20- addi \$10, \$zero, 0x1001

\$11\$10, \$10, 16

ori \$T1, \$zero, 100 #T1 = 100

ori \$T2, \$zero, 0 #T2 = 0

SOMA:

lw \$T2, 0(\$10) #T2 = vetor[1]

add \$T3, \$T3, \$T2 #T3 += T2

addi \$T0, \$T0, 1 #T0 += 1

addi \$T0, \$T0, 4 #T0 += 4

bne \$0, \$T1, SOMA #for(1=0; 1<100; 1++)

sw \$T3, 0(\$10) #vetor[100] = T2

	ALU	3	$4N+1$	401
	MEM	5	$2N$	100
	Desvio	4	$N$	100
	$N$	$49-100$	$7N+1$	701

$$CPI_m = 3 \cdot \frac{401}{701} + 5 \cdot \frac{200}{701} + 4 \cdot \frac{100}{701} = 3,7132$$

$$\text{Tempo de execução: } \frac{200 \cdot 3,7132}{100 \cdot 10^{-9}} = 2,60295 \cdot 10^{-5}$$

ALU	3	$3N+4$	304
MEM	5	$N+1$	101
Desvio	4	$N$	100
$N$	100	$4N+5$	405

$$3 \cdot \frac{304}{405} + 5 \cdot \frac{101}{405} + 4 \cdot \frac{100}{405} = CPI_m = 4,4864$$

$$\text{Tempo de execução: } \frac{405 \cdot 4,4864}{100 \cdot 10^{-9}}$$

$$\text{Tempo de execução: } 1,81699 \cdot 10^{-5}$$

ALU	3	$5N+4$	504
MEM	5	$N+1$	101
Desvio	4	$N$	100
$N$	100	$7N+5$	705

$$3 \cdot \frac{504}{705} + 5 \cdot \frac{101}{705} + 4 \cdot \frac{100}{705} = CPI_m = 2,7829$$

$$\text{Speedup} = \frac{705 \cdot 4,4864 - 1,81699 \cdot 10^{-5}}{705 \cdot 2,7829 \cdot 10^{-5}} = 1,0121$$

Jandaia

22. add \$10, \$zero, 0x1001

sw \$10, \$10, 16

jal main\_recta

calculator:

addi \$sp, \$sp, -8

sw \$ta, 4(\$sp) # adresa ta

sw \$ad, 0(\$sp) # adresa ad

ori \$rs, \$zero, 0 # rs=0

ori \$vo, \$zero, 0 # vo=0

loop:

slt \$at, \$ts, \$ad

beq \$at, \$zero, end-loop # if (a <= b)

andi \$t1, \$ts, 1

beq \$t1, \$zero, even # if ( $t1 = i \% 2$ )

add:

add \$t2, \$zero, \$ts #  $y[i] = i$

store and sum:

even:

sw \$t2, \$ts, 1

addi \$t2, \$t2, -1 #  $y[i] = 2 \cdot i - 1$

store and sum:

ori \$t6, \$zero, 4 # t6=4

mul \$t3, \$ts, \$t6 # desplazamiento = 4 \* i

add \$t4, \$a0, \$t3

sw \$t2, 0(\$t4)

add \$vo, \$vo, \$t2

addi \$t5, \$ts, 1

j loop

End loop:

lw \$ra, 4(\$sp)

lw \$ad, 0(\$sp)

addi \$sp, \$sp, 8

jr \$ra

23. add \$10, \$zero, 0x1001

sw \$10, \$10, 16

lw \$so, 0(\$10) #  $x = 2$

lw \$s1, 4(\$10) #  $y = 5$

or \$a0, \$zero, \$s0

or \$a1, \$zero, \$s1

jal pow

sw \$vo, 8(\$10)

j kim

pow:

ori \$vo, \$zero, 1 # K < 1

or \$t1, \$zero, \$ad #  $T1 = y$

F:

beq \$t1, \$zero, end # if ( $y = 0$ )

ori \$t2, \$zero, 0 #  $x = 0$

ori \$t3, \$zero, \$vo #  $T3 = K$

multi:

add \$t2, \$t2, \$ad

addi \$t3, \$t3, -1

bne \$t3, \$zero, multi # if ( $t3 \neq 0$ )

or \$vo, \$t2, \$zero

addi \$t1, \$t1, -1

j F

Fim:

• data

x: .Word 2

y: .Word 5

K: .Word -1

D S T O O S S  
D L M M J V S

24. add \$t0, \$zero, 6, \$100

slt \$t0, \$r0, 16

lw \$s0, 0(\$r0) # \$s0 = r

lw \$s1, 4(\$r0) # \$s1 = array[0]

ori \$a0, \$t0, \$zero # a0 = \$1

or \$a1, \$s0, \$zero # a1 = N

jal create

or \$s2, \$r0, \$zero # \$s2 = array[0]

j main

create:

addi \$sp, \$sp, -12 # new frame

sw \$ra, 4(\$sp) # return ra

sw \$s0, 0(\$sp) # return s0

ori \$t2, \$zero, 30 # t2 = 30

slt \$t3, \$t2, \$a1

bne \$t3, \$zero, \$1x # if (30 <= N)

j start

start:

ori \$r0, \$zero, 0 # soma = 0

ori \$t4, \$zero, 6 # i = 0

loop:

beq \$t4, \$a1, end # if (i == n)

andi \$t5, \$t4, 1  
bne \$t5, \$zero, 1 # if (i >= n)

brw \$t5, \$zero, 1, main # if (i >= n)

for:

sw \$r0, 12(\$sp)

ori \$a3, \$t4, \$zero # a3 = i

jal square

lw \$t6, 12(\$sp)

Jandaia

slt \$t7, \$r0, 1 # t7 = 2 \* i

slt \$t8, \$t4, 1 # t8 = 2 \* i

add \$t9, \$t7, \$t8

addi \$t7, \$t7, 1 # t7 = 2 \* i + 1

sw \$t7, 0(\$a0) # array[i] = t7

add \$t6, \$t6, \$t7 # soma += array[i]

ori \$r0, \$t6, \$zero

j main

main:

sw \$r0, 12(\$sp)

ori \$a3, \$t4, \$zero # a3 = i

jal square

sw \$r0, 0(\$a0) # array[i] = i^2

lw \$t6, 12(\$sp)

add \$r0, \$r0, \$t6 # soma += array[i]

j inc.

inc:

addi \$t4, \$t4, 1 # i +

addi \$a0, \$a0, 4

j loop

end:

lw \$ra, 4(\$sp) # Restaura ra

lw \$r0, 0(\$sp) # Restaura r0

addi \$sp, \$sp, 12 # Restaura frame

jr \$ra

square:

mult \$a3, \$a3 # i \* i

mflo \$r0 # r0 = i^2

jr \$r0

main:

data

x: .word 10

adr: .word 0x10010008

25. 1. Lw \$S1, mem(\$S2) - Is o resultado da memória no endereço memt S2 é armazenado em S1 - Ativa Mem Read e Alusrc e RegWrite. A ULA soma mem + S2  
 2. SW \$S1, mem(\$S2) - Armazena o resultado de S1 na memória no endereço mem + S2  
 Ativa MemWrite e Alusrc. A ULA soma mem + S2.

3. Bge \$S1, \$S2, pulos - salta para pulos se S1 > S2 (sentido oposto) - Ativa Branch. A ULA compara S1 - S2.  
 4. ADD \$S1, \$S2, \$S3 - Soma S2 e S3 e armazena o resultado em S1  
 Ativa RegRd e RegWrite. A ULA soma S2 + S3

26. a) Lw  $\rightarrow 4hs + 2hs + 4hs + 1hs = 11hs$

$$Sw \rightarrow 4hs + 2hs + 4hs = 10hs$$

$$Bge \rightarrow 1hs + 2hs + 1hs = 4hs$$

$$Add \rightarrow 1hs + 2hs + 1hs = 4hs$$

b) GPC: Mono: Te = 11 · 100% = 11hs

$$Multi: Te = 11 \cdot 22\% + 10 \cdot 14\% + 7 \cdot 2\% + 4 \cdot 49\% = 8,07hs$$

$$Sup = \frac{11}{8,07} = 1,3630$$

ABC: Mono: Te = 11 · 100% = 11hs

$$Multi: Te = 11 \cdot 14\% + 10 \cdot 49\% + 7 \cdot 22\% + 3 \cdot 2\% = 7,57hs$$

$$Sup = \frac{11}{7,57} = 1,44120$$

