Disciplina: CIC 116394 – Organização e Arquitetura de Computadores – Turma A

Prof. Marcus Vinicius Lamar

 $d_0$   $d_1$  /  $d_2$   $d_3$   $d_4$   $d_5$   $d_6$   $d_7$   $d_8$ 

\_ Matrícula: 1 2 / 0 1 2 3

2018/1

## Prova 1

- (4.0)1) Dado as funções em C ao lado.
- (2.0) a) Respeitando a convenção do uso de registradores, eficientemente os procedimentos para Assembly RISC-V RV32IMF.
- (2.0) b) Para o pior caso de um vetor de entrada de dimensão 100, qual a CPI média que as operações aritméticas/comparação em ponto flutuante devem ter para que seu programa sort (v, 100) seja executado em 184µs em um processador com frequência de clock de 1GHz, onde as operações da ISA RV32I e de leitura/escrita na memória (lw,sw,flw,fsw) são executadas em um ciclo de clock?

```
void swap(float v[], int k) {
   float temp;
   temp=v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
void sort(float v[], int n) {
   int i, t=1;
   while(t==1) {
      t=0;
      for(i=0;i<n-1;i++)
          if(v[i]>v[i+1]) {
               swap(v,i);
                t=1;
```

- (4.0) 2) Para o programa em linguagem de máquina do processador RISC-V dado no mapa de memória ao lado, onde os endereços e os conteúdos estão em hexadecimal:
- (2.0) a) Dessamble o programa escrevendo os mnemônicos e dados usados.
- (0.5) b) Quantos bytes de código (instruções) e quantos bytes de dados estão sendo utilizados na sua execução?
- (0.5) c) Qual é o valor impresso na tela?
- (1.0) d) Se transferirmos todo programa do endereço 0x00400000 para o endereço 0x00520050, reescreva o mapa de memória com o código em linguagem de máquina (em hexadecimal) de modo que ele seja executado corretamente neste novo endereço.

```
00400000 : 00000297
00400004 : 02428293
00400008 : 0002a007
0040000C : 0042a087
00400010 : 18107553
00400014 : 101075d3
00400018 : 00200893
0040001C : 00000073
00400020 : 00a00893
00400024 : 00000073
00400028 : c0b00000
0040002C : 40100000
```

(1.5) 3) O programa montador permite que o programador use diversas pseudoinstruções que tornam um programa em Assembly mais fácil de ser lido e escrito. Escreva o código da ISA RV32IMF que implementa as seguintes pseudoinstruções:

```
(0.5) a) jr t0
                                 \# PC = t0
(1.0) s) fisqrt.s ft0, ft1
                                 # ft0 = 1/sqrt(ft1)
```

## (2.0) 4) Responda:

- (1.0) a) A potência dissipada em circuitos digitais CMOS é devido a fatores estáticos e dinâmicos. Cite 2 fatores responsáveis pela potência estática e 2 fatores responsáveis pela potência dinâmica dissipada.
- (1.0) b) "Um processador capaz de executar 10.000.000 de instruções por segundo é mais rápido que outro capaz de executar 5.000.000 de instruções por segundo." A afirmação é verdadeira ou falsa? Por que?

Boa sorte!

2) 0x00400000 AVIPC 10,0 0x00400000 APP; to, to, ox24 0x00400008 Flw fto, o(to) 0x00400000 Flw fto, o(to) 0x00400000 Flw fto, o(to) 0x00400014 Fours Fal, Fto, Fto 0x0040018 addi a7, 2613,2 0x0040010 CCALL 0x0040010 FCALL 0x0040010 AX 0000000 0x00400014 PCALL 0x0040010 AX 0000000 0x00400014 PCALL 0x0040010 AX 00000000 0x00400014 PCALL 0x00400015 AX 4010 DOOO  PRoblema: Raps (olocal to = FC+4),  48 PTA fto = FX > 0x00  6 TTA fto = 0x000 00073 =  6 TTA fto = 0x0000 00073 =  6 TTA fto = 0x000 0000 0000 0000 0000 0000 0000		
0x00400000 AUIPC 10.0  0x00400004 Appi to to, ox24  0x00400008 FLW fts, o(\$0)  0x0040000C FLW ft1, y(\$1)  0x00400014 Fmul & Fal, FT0, FT1  0x00400014 Fmul & Fal, FT0, FT1  0x004001C CCGLL  0x00400014 PCALL  0x00400000 AUIPC ACCORD OCCOD  0x00400015 Appi COLOCCO PT & FC + 410  1x1 = 2,25 - 0x0  1x1 = 2,25 - 0x0  0x1 = 10x4 = 40 by tec  0x1 = 1x4 = 8bx TCS  1x1 = 5		
0x00400004 App; to, to, 0x24  0x00400008 FLW fts, 0 (10)  0x0040000C FLW ft1, 4 (11)  0x00400014 Fmuls Fal, FT0 Ft1  0x00400014 Fmuls Fal, FT0, Ft1  0x0040016 CC4LL  0x00400020 9ld; 07 Feno, 10  0x00400024 PCALL  0x00400024 PCALL  0x0040002C 0x40100000  PROBLEMA: RAIS 6060000 ft = FC+40  12 PTE. ft = -5,5 -0x0  ft1: 2,25 -0x4  C) -7.5/2,25 = -2,44444 - impresso,  10 PTES BADS = RX4 = 867 TCS  COTSIDE argo: to=FC  Cota ft = 0x000000073 = 4x4 = 5x4	Production of the Control of the Con	
0x00400008 FLW fts, & (to)  0x0040000C FLW ft1, 4(t1)  0x00400014 FMULS FAD, FTO FT1  0x00400014 FMULS FAD, FTO FT1  0x0040016 Oddi Q7, ZERD, 2  0x00400024 PCQLL  0x00400024 PCQLL  0x00400024 DXC0800000  0x00400024 DXC0800000  0x00400028 OXC0800000  0x00400028 OXC0800000  0x00400028 OXC0800000  0x00400028 OXC0800000  0x00400028 OXC0800000  0x00400028 OXC0800000  0x00400028 OXC08000000  0x00400028 OXC08000000  0x00400028 OXC08000000  0x00400029 PCQLL  0x00400028 OXC080000000  0x00400028 OXC0800000000  0x00400029 PCQLL  0x0040000000000000000000000000000000	02	
0x0040000C FLW ff1, 4(t1)  0x00400014 FmULS FAD, FT0 FF1  0x0040018 Oddi 97, 2413, 2  0x004001C CCGLL  0x0040020 9ddi 07, 2610, 10  0x00400024 P. CQLL  0x00400024 P. CQLL  0x0040002C Dx40100000  2 PROBLEMA: RAPS COLOCAL FT8 = PC+410  12 PTTTO ft8 = -5,5 > 0xC  ft1 = 2,25 > 0xY  C) -7,5/2,25 = -2,44444 > impressor  b) Bytes (5pigo = 10x4 = 40 bytes  0x7tes eards = 2x4 = 8bytes  1000 Caya Vrgly Flaw -> ftg = 3  6x1000000073 = 4x4 = 5x1000000073 = 4x1000000073 = 4x10000000073 = 5x10000000073 = 5x10000000073 = 5x100000000073 = 5x1000000000000000000000000000000000000	92	
0x00400019 Foirs Far F10 F11  0x00400018 addi at zero, 2  0x0040010 Ecall  0x0040020 addi at zero, 10  0x00400024 P. Call  0x00400024 P. Call  0x00400026 0x Cobo 0000  0x00400026 0x Cobo 0000  0x00400026 0x Cobo 0000  1x00400026 0x Cobo 0000  1x00400000000000000000000000000000	00	
0x00400014 Emuls Fal, F16, F11  0x0040018 addi at, 2612, 2  0x0040010 ECGLL  0x0040020 addi at 2010, 10  0x00400024 PCGLL  0x00400028 0x0000000  0x00400026 0x40100000  PRoblema: Raps colocal fta = PC+41,  42010 0000  (x1 = 2,25 - 0x4  C) -5,5/2,25 = -2,44444 + 1mfresso,  b) Bytes copias = 10x4 = 40 bytes  0x1000000003	0,2	
0x0040018 addi at tens, 2  0x004001C ECGUL  0x0040020 98di at tens, 10  0x00400024 PCQLL  0x00400028 0x0000000  0x0040002C 0x40100000  PROBlema: Rans Colocal tes=PC+4,  12 PATE: fts=PC+4,  12 PATE: fts=PC+4,  12 PATE: fts=PC+4,  13 PATE: fts=PC+4,  14 1: 2,25 - 0x4  C) -7.5/2,25 = -2,44444 > imfresso,  15 PATE: 6205 = 2x4 = 8647CS  10 COTSIBLATION TO FEE  CAUSA UTPERFIBE -> fts=5  1000	02	
0x0040020 9Ndj a7 jero, 10 0x00400024 P CALL 0x00400028 0x COBO 0000  0x0040002C 0x4010 0000  PROBLEMA: RAPS COLOCAPTO = PC+40  4 PTTTO fto = -5,5 > 0xC  1x1 = 2,25 > 0x4  C) -7,5/2,25 = -2,44444 > impresso,  b) Bytes (5pico = 10x4 = 40 bytes  0xtes papos = 2x4 = 8bxtes  Corsiblarios to = PC  Corsiblarios t	0,2	Military and the second
0x0040020 98dj a7 ženo,10 0x00400024 PCQLL 0x00400028 0x0000000 0x0040002C 0x40100000  PROBlema: Rans colocal \$40 = PC + 410  \$1 = 2,25 - 0x0  \$1 = 2,25 - 2,44444 + 10 bytec  \$1 + 10 = 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2	50	
0x00400024 P.CALL 0x00400028 0x0000000  0x0040002C 0x40100000  PROBLEMA: RANS COLOCATES=8C+40  1x1=2,25 -0x4  C) -5,5/2,25=-2,44444 - 11M100550  D) BYTES COPIGO = 10x4 = 40 bytes  0x14000000073 =  4 PROBLEMANDO: XO=PC  Corsibliarios: XO=PC  C	02	TO THE SECOND CONTRACT OF THE SECOND CONTRACT
0x00400028 0x00800000  0x00400028 0x40100000  PROBLEMA: RAPS (0local Ptb = PC+40  12 80700 ftb = -5,5 -0x0  ft1 = 2,25 -0x4  C) -7.5/2,25 = -2,44444 - 10 bytes  0x7tes papes = 2x4 = 8bytes  Corsiblarios to = PC  Corsibla	96	
0 x 0040 002C 0 x 4010 0000  PROBLEMA: RAPS COLOCAL # = PC+40  4 PTTTO fte = -5,5 > 0x C  ft1 = 2,25 > 0x 4  C) -5,5/2,25 = -2,44444 > impresso,  b) Bytes copies = 10 x 4 = 40 bytes  07 tes papos = 2 x 4 = 8 bytes  Corsiblarios to = PC  Corsi	0,6	NOCE TO A STATE OF THE PARTY OF
PROBLEMA: RAPS COLOCAL # 28 = 8C+40  1 8/150		
5 8750.   fto = -5,5 30xC    (t1 = 2,25 30x4)   () -5,5/2,25 = -2,44444 3impresso,   () -5,5/2,25 = -2,444444 3impresso,   () -5,5/2,25 = -2,44444 3impresso,   () -5,5/2,25 = -2,44444 3impresso,   () -5,5/2,25 = -2,44444 3impresso,   () -5,5/2,25 = -2,4444		
5 8750.   fto = -5,5 30xC    (t1 = 2,25 30x4)   () -5,5/2,25 = -2,44444 3impresso,   () -5,5/2,25 = -2,444444 3impresso,   () -5,5/2,25 = -2,44444 3impresso,   () -5,5/2,25 = -2,44444 3impresso,   () -5,5/2,25 = -2,44444 3impresso,   () -5,5/2,25 = -2,4444		HE SECRETARIAN CONTRACTOR OF THE SEC
(2) -5.5/2,25 = -2,44444 + impressor  b) Bytes copico = 10 x4 = 40 bytes  Bytes capos = 2x4 = 8bytes  corribbarios to = PC  Cra fto = 0x0000000073 =  Gausa Urgentian -> fto = 3  [1000]		
b) Bytes (5pigo = 10x4 = 40 bytes  Bytes papos = 2x4 = 8bytes  Corsiberargos to = PC  Corsi	080000	)
b) Bytes (5pigo = 10x4 = 40 bytes  Bytes papos = 2x4 = 8bytes  Corsiberarios to = PC  Corsi	010000	3
(0775 ible argoi to = PC  Corrsible argoi to =	Vo tela	05
Considerans to = Rx4 = 864 Tes  Considerans to + PC  Considerans to + PC  Considerans to + PC  Swither periods  Cousa Ungertien -> ftg = 3  ft1 = -5  1000		
Considerans to = Rx4 = 864 Tes  Considerans to + PC  Considerans to + PC  Considerans to + PC  Swither periods  Cousa Ungertien -> ftg = 3  ft1 = -5  1000		
Considerarios to = PC  Considerarios to = PC  Considerarios to = PC  Services person  Causa Urgention -> fte = 5  [1000]	0 25	
Causa Urgentise -> ftg = 5  1090	9 25	Nonemaka kutu maka kata anjuga da anjuga kata anjuga k
Causa Urgentise -> ftg = 5  1000		
Causa UrgerFish -> ft6 = 5		-75
Causa UrgenFlow -> ftg = 5	1,61 Klo	
LOSO		2
LOSO	0//	
	.5	
CONTRACTOR OF THE CONTRACTOR O	No 170 .	
7,77-10	7869	05
	Auragian Carlotte Communication Communicatio	

	d) como o programa não usa Eranocus ABS	OCUTOS
	O Coolgo en Cirquagen de nagvina pontara que	700
	O enpeles exos 20050 Permanece exogan	reve 10
	O MEGMO,	
		1
	3) 9) Ir to -> 19LR 200, to,0	9>
	V	A
	b) NET PAGE GLECKAR NEWHUM REST, 59 MODOR ALEN	Lefto,
	addi 48,98,-8 fisgatis 180,	891
	Sw to, 0 (59)	
	F60 ft, 4(58)	
	oidd'i to, ZERS, 1	10
	fevs.5.w fto, to	1,0
	JEGIT. 5 Ste Ste	
	LW to, OCSP)	
	Alm Ax1, 4(59)	
	addi SP. SP. 8	
	0000 91,71,0	
Chart Neverthald Streeth uson will call a Charge and electric hadden	4)	
	a) Potêicia Estática:	
	- CORMETE de Euga dos Transistores	9,25
-	- CORRETTE NEVERGA POS PIDEDS	0 25
	Potacia DiNAMICa:	0,25
	- CONCETTE Le consa do gate pos TVa	ngisjanas
	- CUNTO CINCUITO de Chaveamento	0,25
	G) VERDADSIRA &	
	Não está se talando em desempenho e sim e,	
	Afenas velocinage na execução de instruções	> 1/

.data

```
# Pior caso, vetor reversamente ordenado
V: .float 100, 99, 98, 97, 96, 95, 94, 93, 92, 91, 90, 89, 88, 87, 86,
85, 84, 83, 82, 81, 80, 79, 78, 77, 76, 75, 74, 73, 72, 71,
70, 69, 68, 67, 66, 65, 64, 63, 62, 61, 60, 59, 58, 57, 56,
55, 54, 53, 52, 51, 50, 49, 48, 47, 46, 45, 44, 43, 42, 41,
40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26,
25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11,
10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0
newl:
       .string "\n"
tab:
       .string "\t"
.text
main:
       la
              a0, V # argumento0 V
              a1, 100 # argumento1 100
       li
       jal
             show
              a0, V # argumento0 V
       la
       li
              a1, 100 # argumento1 100
       jal
              sort
                   # chama sort
       la
              a0, V # argumento0 V
       li
              al, 100 # argumento1 100
       jal
              show
       li
              a7, 10 # serviço exit
0,2
              t0, a0
sort:
       mv
                            # salva ponteiro v em t0
              t1, a1, -1
       addi
                            # salva n-1 em t1
       mv
              t5, ra
                             # salva ra em t5 (procedimento não folha)
       li
              t2, 1
                             \# t=1
while:
       beg
              t2, zero, endwhile
                                    # while(t==1)
                                                   0,2
              t2, zero
       mv
                                     # t=0
       mv
              t3, zero
                                     \# i = 0
              t3, t1, while
for:
       bge
                                   # i>=n-1 ? fim do for, volta para o While
       slli
              t4, t3, 2
                                     # i*4
       add
              t4, t0, t4
                                     # v+4*i
       flw
              fal, 0(t4)
                                     # le v[i]
       flw
              f#3, 4(t4)
                                     # le v[i+1]
       #addi s0,s0,1 ######## Apenas para confirmar o número de vezes que passa pelo fle.s
              t4, f. Q, f. B
       fle.s
                                  # v[i]<=v[i+1] ?
       bne
              t4, zero, nextfor
                                    # sim -> nextfor
              a0, t0
       mv
                                     # a0=v
       mv
              a1, t3
                                     # a1=i
       jal
              swap
                                     # call swap
       li
              t2, 1
                                     \# t=1
nextfor: addi
              t3, t3, 1
                                     \# i = i + 1
                                                             0,2
              for
       j
                                     # j for
endwhile: jr
              t5
                                     # retorna para o ra salvo
```

```
swap:
       slli
               a1, a1, 2
                            # multiplica k*4
       add
               a0, a0, a1
                                                         0,5
                               # soma endereço base
       flw
               ft0, 0(a0)
                               # le v[k]
       flw
               ft1, 4(a0)
                               # le v[k+1]
       fsw
               ft0, 4(a0)
                               # grava v[k]
       fsw
               ft1, 0(a0)
                              # grava v[k+1]
       ret
```

show: mv t0,a0 mv t1,a1

mv t2,zero

loop1: beq t2,t1,fim1

li a7,2

flw fa0,0(t0)

ecall
li a7,4
la a0,tab
ecall

addi t0,t0,4 addi t2,t2,1

j loop1

fim1: li a7,4

la a0, newl

ecall ret