2ª Prova - Gabarito 2006/2

TURMA A/B

1) a) abs rs, rt ⇒

```
          SLT $at, rt, $ZERO          # at = 1 rt < 0
          beq $at, $ZERO, Positivo
          SUB rt, $ZERO, rt
Positivo: OR rs, rt, $ZERO            # mov rs, rt
```

b) addi.s Ft, Fs, imm ⇒

```
ORi $at, $ZERO, imm    # Li $at, imm
mtC1 $at, Ft
cvt.s.w Ft, Ft
add.s Ft, Fs, Ft
```

a) neg rt, rs ⇒

```
SUB rt, $ZERO, rs    |  NOR $at, rs, rs
                     |  addi rt, $at, 1
```

b) multi Ft, Fs, imm ⇒

```
ORi $at, $ZERO, imm    # Li $at, imm
mtC1 $at, Ft
cvt.s.w Ft, Ft
mul.s Ft, Fs, Ft
```

2) overflow: - situações

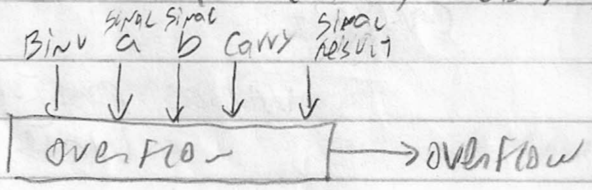
$(A \geq 0) + (B \geq 0) = R < 0$ ①

$(A < 0) + (B < 0) = R \geq 0$ ②

$(A \geq 0) - (B < 0) = R < 0$ ③

$(A < 0) - (B \geq 0) = R \geq 0$ ④

⑤ Soma ou sub > 32 bits ↳ carry out = 1

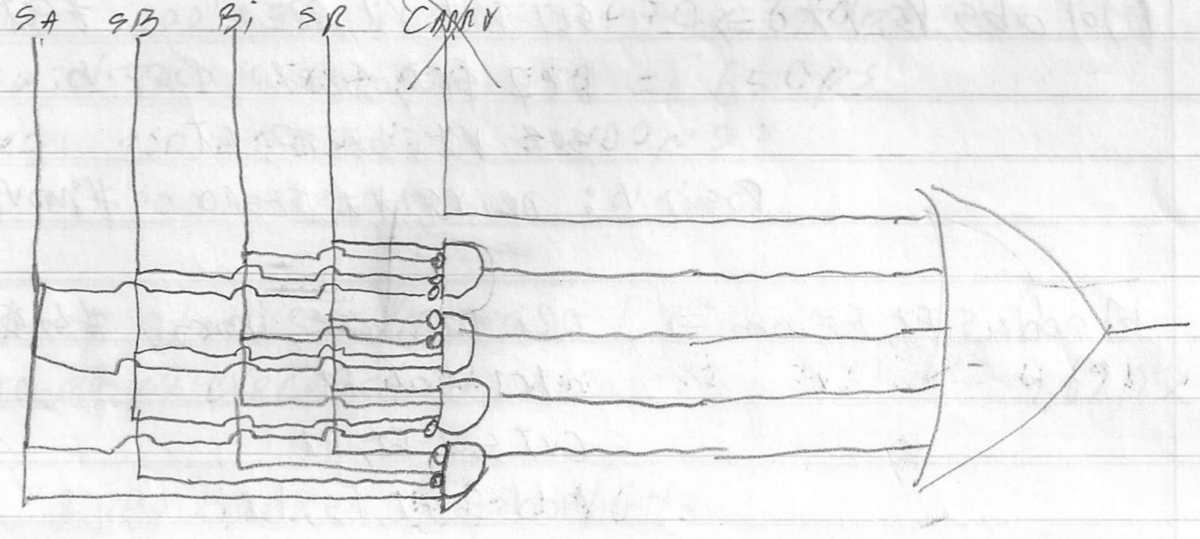


5 casos:

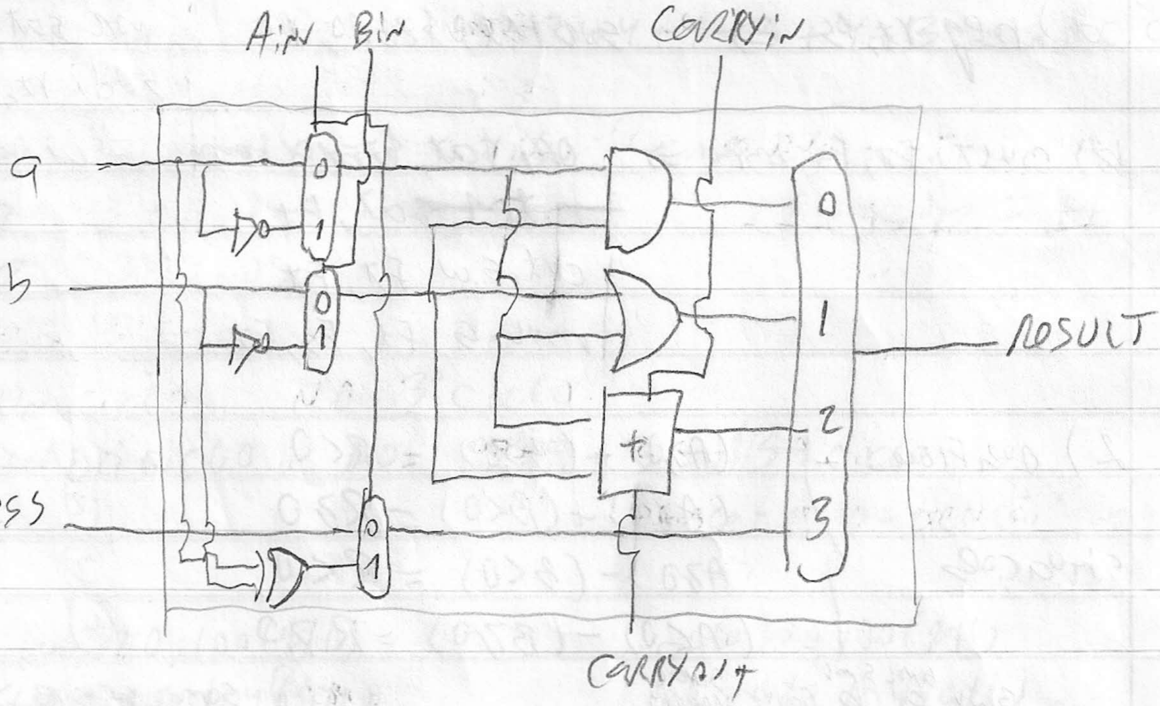
SA Sinal a	SB Sinal b	Bi Binv	Carry	SR Sinal result	Overflow
0	0	0	X	1	1
1	1	0	X	0	1
0	1	1	X	1	1
1	0	1	X	0	1
[X	X	X	1	X	1]

Projeto

OVERFLOW = ~~CARRY~~ + $\overline{SA}\cdot\overline{SB}\cdot\overline{Bi}\cdot SR + SA\cdot SB\cdot\overline{Bi}\cdot\overline{SR} +$
$+\ \overline{SA}\cdot SB\cdot Bi\cdot SR + SA\cdot\overline{SB}\cdot Bi\cdot\overline{SR}$ +



3)



Operações implementadas	
AND	0000
OR	0001
SOMA	0010
SUB	0110
SLT	0111
NOR	1100

Várias possíveis soluções

→ Bloqueio do CARRYin

→ escolho 1111 → XOR

logo bit Ainv → seleciona saída 3

4) OVERFLOW e instrução não definida
ver livro : Fig. 5.39 e 5.40

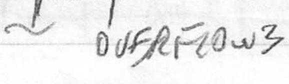
Mudança na μ instrução
→ incluir sinais OVERFLOW na entrada
→ incluir sinais $\underbrace{\text{causaInt, escreveCausa, escreveEPC}}$

ALU CTRL	SRC1	SRC2	Reg ctrl	memo	PCwrite	Exceção	seq

Exceção	OVERFLOW	CausaInt = 1 EscreveCausa EscreveEPC
	instrução não recog.	CausaInt = 0 EscreveCausa EscreveEPC
PCwrite ctrl	Exceção	OriPC = 11 EscrevePC

LABEL	ALU CTRL	SRC1	SRC2	Reg CTRL	Memo	PCwrite	Exceção	seq
FETCH	Add	PC	4		read PC	ALU		seq
	Add	PC	EXTSHFT	Read				DISPATCH 1
MEM1	Add	A	Extend					dispatch 2
LW2					read ALU			seq
				write MDR				Fetch
SW2					write ALU			Fetch
RFORM1	FUNCODE	A	B					seq
				WRITE ALU				Fetch
OVERFLOW3	SUB	PC	4			EXEÇÃO	OVERFLOW	FETCH
NOT recog1	SUB	PC	4			Exeção	instrução não recog.	Fetch
BEQ1	SUB	A	B				ALU COND.	FETCH
JUMP1							JUMP ADD	FETCH

DISPATCH 1 → qualquer outro endereço deve apontar para "NOT Recog1"
Alterar o controle de próximo estado



se houver Overflow
vai p/ OVERFLOW 3
Caso contrário segue
o seq.

5)
a) Uniciclo:
Tipo-R: 200 + 50 + 150 + 70 = 470ps
LW : 200 + 50 + 150 + 200 + 70 = 670ps
SW: 200 + 50 + 150 + 250 = 650ps
Beq: 200 + 50 + 150 = 400ps
J : 200 = 200ps

Freq. clock = 1/670ps = 1,492 GHz
Tempo p/ executar 7 instruções: 7x670p = 4,69nS

b) Freq. clock = 1/250p = 4GHz
Tempo execução:
Tipo-R : 4 ciclos
LW : 5 ciclos
SW : 4 ciclos
Beq : 3 ciclos
JP : 3 ciclos

total:
5+5+4+4+3+4+3 = 28 ciclos

logo: Tempo execução:
28 x 250p = 7nS

c) Uniciclo: No 3º ciclo
Multiciclo: No 5+5+3 = 13º ciclo
↳ busca + decod + execução

d)
```
LW $t0, 100($t1)
LW $t2, 104($t1)
ADD $t3, $t0, $t2
SW $t3, 100($t1)
MOV $t4, $t3   #OPCIONAL
J LABEL2
```

```
LW $t0, 100($t1)
LW $t2, 104($t1)
beq $t2, $ZERO, LABEL2
add $t3, $t0, $t2
SW $t3, 100($t1)
J LABEL2
```