2.19, 2.21, 2.22, 2.24, 2.25, 2.26

AEX 4

2.19) $t0 = 0xAAAA AAAA $t1 = 0x1234 5678

2.19 1) xall $t2, $t0, 44 → $t2 = $0
ou $t2, $t2, $t1
$t2 = $t1

2.19 2) xall $t2, $t0, 4 (I)
xandi $t2, $t2, -1 (II)
↳ 0xFFFF
(I) $t2 = 0xAAAA AAA0
(II) 0xAAAA AAA0
AND 0x0000 FFFF
0x0000 AAA0
↳ não estende o bit de sinal

2.19. 3) srl $t2, $t0, 3
xandi $t2, $t2, 0xFFEF

(000 x x x x | x x x x
0001 0101 0101 0101 | 0101 0101 0101 0101
AND 0x1555 5555
0x0000 FFEF
0x0000 5545
AND 0101
1110
0100

2.21) not $t1, $t2 $t1 = $\overline{t2}$
tinha nor como matriz: NOR $t1, $t2, $0
$\overline{(A+B)}$: $\overline{A}.\overline{B} \rightarrow \overline{A}$
↳ NOR ↓ se $\overline{B}=1$
logo B = 0

2.22) $t1 = A $s1 = &C[0]
$t2 = B
A = C[0] << 4 (xsll)
xlw $t0, 0($s1)
xsll $t1, $t0, 4

2.24) PC = 0x2000 0000 → 0x4000 0000
jump? → não porque não altera os quatro bits mais significativos do PC
beq? → não pois:
⊕ 0111 1111 1111 1111
<<2
⊕ 0111 1111 1111 1111 00
+ 100
(00 1000 0000 0000 0000 0001 0000 0100)
⊕ 0x0002 0000 → 0x2002 0000

2.25)

2.25. 1) I

2.25. 2) rpt $t2, Loop # use (R[rs] > 0)

?

R[rs]--;

senão PC = PC + 4 + BranchAddr

2.26) LOOP: slt $t2, $0, $t1 } (N+1)
beq $t2, $0, DONE }
subi $t1, $t1, 1 } (N)
addi $s2, $s2, 2 }
j LOOP

DONE

2.26.1) $t1 = 10
$s2 = 20

2.26.3) 2(N+1) + N.3
5N + 2

for(i = N; i ≥ 0; i--){
B+= 2 (x10)
}