

- 11.
- 1) 寄存器寻址模式
  - 2) 寄存器寻址模式
  - 3) 立即数寻址模式
  - 4) 寄存器寻址模式
  - 5) 寄存器间接寻址模式

- 3.
- (1) nop 为 `ADDI x0, x0, 0`
  - (2) ret. 为 `jalr x0, 0(x1)`
  - (3) call offset 为 `auipc x6, offset [31:12]`  
`jalr x1, x6, offset`
  - (4) mv. rd, rs 为 `addi rd, rs, 0`
  - (5) rdcycle 为 `csrrs rd, cycle[h], x0`
  - (6) sext.w rd, rs 为 `addiw rd, rs, 0`

- 7.
- (1) slt. t3, rs0, rs1  
`slt t4, rs0, rs2`

- (2) bgt t0, t1, overflow.

- (3) ARM 通过 CPSR 的状态寄存器反映当前指令的溢出状态  
 MIPS 通过指令触发中断的方式产生溢出信号.

8. (1)  $2^{xLEN} - 1$   $x$   $-1$   $x$

不会抛出异常, 使整个架构更为轻巧.

- (2) NV. Invalid operation.

DZ Divided by Zero

OF Overflow

UF Underflow

不会使得处理器陷入死循环

13) 除数为零产生“除以零”异常.

12. 11) M (2) M (3) S (4) S (5) U

13. ~~part1:~~ ~~add t3, a3, 1~~ a

~~part1:~~

add t3, x0, x0

addi t4, x0, 100

Loop: bge t3, t4, exit

mula t5, t2, 0(t1)

add t0, x0, t5

lw t0, 4(t0)

lw t1, 4(t1)

addi t3, t3, 1

j loop

14. bge a0, a1, more\_less

add a2, a0, a1

less: sub x a0, a1

~~lw~~ lw a2, 0(a0)



5. ~~add t0, x0, 0(t0).~~

~~addi t1, x0, 3~~

~~lw 1(t0), t1~~

~~lw 3(t0), t1~~

add x3, x0, t0

sw x3, 0(t0)

sw t1, 4(t0).

add t0, t1, t0

sw t1, 0(t0).

16. ~~lw x3, 0(t0)~~

~~lw x3, 0(t0)~~

~~sw t2, 0(t0)~~

5 lw x3, 0(t0)

lw x4, 0(t1)

~~lw t0, x3~~ sw x3, 0(t1)

~~lw t1, x3~~ sw x4, 0(t0)

17. 将  $a_i$  中的数向左移 1 位并重复 30 次 即令  $a_i$  乘以  $2^{30}$