aaram - Ahmad Aram
hBenali - Adam Benali

1

#1.

a. not included in mips

abs: ALU would it implemented first before its implemented in the ISA main reason being you can technically do it but requires XOR and add/sub which the ALU can do, hence why its a pseudo instruction in mips.

Sgt: set greater. Then is a pseudo instruction in mips

B. Sequence for ABS: sra $T0, $T0, 31 # sets sign bit of T0 to T1
xor $T0, $T0, $T1 # Flip all bits
sub $T0, $T0, $T1 # subtract to set abs value if it was neg you set pos, if pos it stays positive.

SGT:

slt $T1, $T2, $T3

- would check if Y is greater than X

C. ABS: R type because 2 reg registers

Sgt: R type as well, since its the opposite of SIT.

#2

a. 0x 14 = 20 in decimal

6. C code

T1, T2, S2
y z x

```
.....
int X = 10
int z = 0
int y = 0
while (X > Y) {
  X = X-1;
  z = z+2
}
```

C Instructions for h

- counting the 1i instructions for $T1 and $T2

0 = 6 77

1 = 13 77

2 = 20 77

3 = 27

N·7 t6

3. a

PC = 0x 0000 0000 Branches needed = 1

Address 0X 0002 0000

pc+1 address 2^{05}

0000 0002 000000

- $2^{15}-1$ is the maximum address for one Branch instructions. So you can reach 0X 0002000 in just 1 Branch.

0 0 1 0

- as a side note it also goes $-2^{27}-1$ is to go backwards.

3 B. PC = 0X 0000 0000 Jumps needed = 1

Address : 0X0002 0000

- $2^{28}-1$ and $-2^{27}-1$ are the max values an address can have within one jump.

3 C. 26 bits 2^{26} = total size Cannot Jump to that address Because PC will not reach That.

PC 0X 0000 0000

address 0X FFFF FF00

30.

PC : 0x 0000 0000

address 0x0002 00 0000 = in decimal

8 Bit i TYPE instruction: $\sqrt{2^8}$

its exponential as $2^8 - 1$ is significantly smaller Then 0x 20000 is 131072

0x __ __

3e. PC : 0x 0000 0000

address : 0x 0002 0000

2^8 Jumps

3 F

- Resister addressing - add operations add rd, rs, rt
- Immediate addressing - addi instruction addi rt, rs, I
- Base addressing - lw $T2, 5 Load word instruction.
- PC - relative addressing - beqz $To, StEnd ← Labie
- Pseudo - Direct Addressing - Jump instructions J End ← Labie.