CS 251 - Lecture 2

Graham Cooper

May 4th, 2015

Binary Refresh:

$$101 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

Jump:

28 × 4 = 112
PC ← offset × 4
Overwrite PC. Example: j 28 sets the PC to 28 × 4

jump and branch are I-format

beq \$1 , \$2 , 100

branch if equal
if \$1 == \$2 go to offset relative to PC, else continue to next instruction

We don't multiple by 4 so that we can get more space for the 16 bits we have for the value 100, ie going from 400, if we multiply by 4, it goes to 1600

PC ← 100 × 4 + (PC + 4)

Slide 26:

$$\begin{aligned} PC &\Leftarrow (-3) \times 4 + (PC + 4) \\ &= -12 + 120 \\ &= 108 \end{aligned}$$

Memory Access - I-Format

jump and branch are I-format

lw \$s1 , 100(\$s2)

100(\$s2) is computing an address
\$s1 ⇐ M[address]

sw \$1 , 100(\$2)

M[100 + \$s2] ⇐ \$s1 Memory is written to!!

lw/sw Eample:

```
0 add $4 , $7 , $0
4 addi $1 , $0 , 20
8 lw $3 0($4)
12 addi $1 , $1 , −1
16 sw $3 , O($5)
20 addi $4 , $4 , 4
24 addi $5 , $5 , 4
28 bne $1 , $0 , −6
32 add $8 , $3 , 0
```

0:4 - Setting $4 to beginning of an array
8:28 - Loop (goes 20 times)
8 - Load from memory into $3
16 - Store into Data memory, contents of $3
20:24 - Adding from one array to another array in memory

ARM vs MIPS

add command has to add when using the $0 register, while ARM has a move commmand which bypasses the add.