

Computer Architecture - CS 301

Rishit Saiya - 180010027, Assignment - 2

September 11, 2020

1

There is a limit of load registers and so we use Modifiers here. So, using Modifiers gives us facility to store 32 bit number/immediate into the register. The whole idea of using the modifiers is to redistribute the whole chunk of instructions and calculations and break them in order to store them in the registers.

Some characters are added to the instructions to modify and then make new instructor to work as a Modifier.

Example: *u* is added to an instruction which makes it unsigned, which further implies that all the signed bits have to be 0. *h* is added to an instruction which makes a 16-bit shift to the left.

Goal: Set $r1 \leftarrow 0x12345678$

The following Assembly code with the help of modifiers helps us to achieve the required:

```
movh r1, 0x 12 34      @   (12 34 00 00)
addu r1, 0x 56 78      @ + (00 00 56 78)
```

movh instruction stores 12 34 in first 16 bits & assigns the last 16 bits as 0. **addu** instruction makes all the unsigned bits at 0 and adds to the 12 34 00 00 immediate.

2

The **cmp** instruction is used, this calls the flag registers. The flag registers contains the result of the last comparison. So, they become useful to create loops in the assembly code and help in conditional branched instructions.

Consider an example where $r1 > r2$, then save 2 in r3, else 3 in r3.

```

    cmp r1, r2          @ Comparing r1 & r2 using bgt flag(branch greater than)
    bgt .foo            @ If r1 > r2, then go to foo branch
    mov r3, 3           @ Else r3 <- 3
    ...
    ...
foo:
    mov r3, 2           @ Set r3 <- 2

```

By the definition, if $r1 > r2$, $\text{flag.GT} = 1$, $\text{flag.E} = 0$. If $r1 < r2$, $\text{flag.GT} = 0$, $\text{flag.E} = 0$. If $r1 == r2$, $\text{flag.GT} = 0$, $\text{flag.E} = 1$. These are saved until next call and change according to further calls.

In case of branching, **beq** & **bgt** instructions use flags registers. **beq**: branch if $\text{flag.E} = 1$ & **bgt**: branch if $\text{flag.GT} = 1$. We can't remove them because without them, the branching & looping sequence generation will not be possible.