Notes
06/ 06/ 2021

Branch Instruction (can be considered as conditional jump)

opcode	rs	rt	offset

31: 26	25:21	20:16	15:0
6 bits	5 bits	5 bits	16 bits

Branch target address is obtained by using PC-relative addressing.
Branch target address = (PC+4) + (sign extended offset << 2)
32 bits 32 bits
sign extended offset << 2 = sign extended offset x 4

Jump (can be considered as unconditional jump)

Jump target address is obtained by using Pseudo-direct addressing.
Jump target address = first 4 bit of (PC+4) concatenated with 28 bit of (Target << 2)

opcode	Target

31: 26	25:0
6 bits	26 bits

Example:
Check the following example on page 131 (pdf page 158).

```
Loop:sll    $t1,$s3,2   # Temp reg $t1 = 4 * i
     add $t1,$t1,$s6    # $t1 = address of save[i]
     lw  $t0,0($t1)     # Temp reg $t0 = save[i]
     bne $t0,$s5, Exit  # go to Exit if save[i] ≠ k
     addi $s3,$s3,1     # i = i + 1
     j   Loop           # go to Loop
Exit:
```

If we assume we place the loop starting at location 80000 in memory, what is the MIPS machine code for this loop?

The assembled instructions and their addresses are:

80000	0	0	19	9	2	0
80004	0	9	22	9	0	32
80008	35	9	8	0		
80012	5	8	21	2		
80016	8	19	19	1		
80020	2	20000				
80024	...					

Highlights:
In this class we learned the operation of datapath for implementing Branch and J type instruction as shown in Figure 4.20, and 4.24
Also we review PC-relative and Pseudo-direct addressing mechanisms.
Reference: Section 4.1 - 4.4