08/2019

Monday	-	5	12	19	26
Tuesday	-	6	13	20	27
Wednesday	-	7	14	21	28
Thursday	1	8	15	22	29
Friday	2	9	16	23	30
Saturday	3	10	17	24	31
Sunday	4	11	18	25	-

CS 301

180010011

JULY' 19

Wk-28 Day 191-174

WEDNESDAY

10

Assignment - 2

1) The move instruction in SimpleRISC allows an immediate of x (where x < 32) bits to be moved to a register. But our register can store upto 32 bits. So how do we move a 32-bit immediate, say 0x12345678 to a register r1 ?

Sol) The mov instruction in SIMPLERISC treats a 16 bit immediate as a signed number. When it is transferred to a register which is 32 bit, an automatic sign extension happens.

i.e A register stores the content of the immediate in the later 16 bits and the sign in the first 16 bits.

To move a 32-bit immediate, say 0x12345678 to a register r1, we use u and h.

i.e movu: 16 bit immediate as unsigned number

movh: leftshift 16 bit immediately by 16 positions

i.e movh r0, 0x1234.

addu r0, 0x5678.

2) Role of the flag register is to store the result Of the last comparison.

Cmp instruction -

flag.E, flag.GT represents $r1 - r_2$.

0,1 represents $r_1 > r_2$.

1,0 represents $r_1 == r_2$.

0,0 represents $r_2 > r_1$.

No. Flags are unnecessary registers. Cmp is a very common operation. Flag registers gives a way to store the result of the last cmp without using a register visible to the programmer.

In an if-else statement or in a condition on loop we use values at the flag register. So, they are not necessary.

They can be removed because their functionality can be implemented by using other registers.