❖ RTL (Register Transfer Language)

- A register transfer language is a symbolic notation used to describe the microperation transfers between registers.

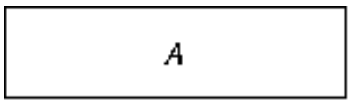
- Registers are denoted by capital letters and are sometimes followed by numerals, e.g.

    1. MAR – Memory Address Register (holds addresses for the memory unit)
    2. PC – Program Counter (holds the next instruction's address)
    3. IR – Instruction Register (holds the instruction being executed)
    4. R1 – Register 1 (a CPU register)

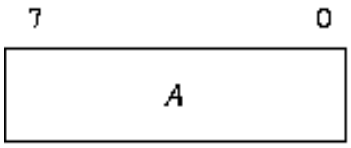
- We can indicate individual bits by placing them in parentheses. e.g., PC (8-15), R2 (5), etc.

- Block Diagrams of Registers

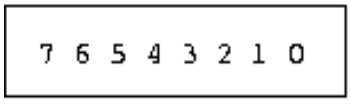
-



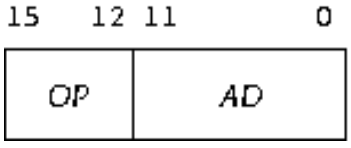
A simple register



Register showing bit positions



A *Register showing individual bits*



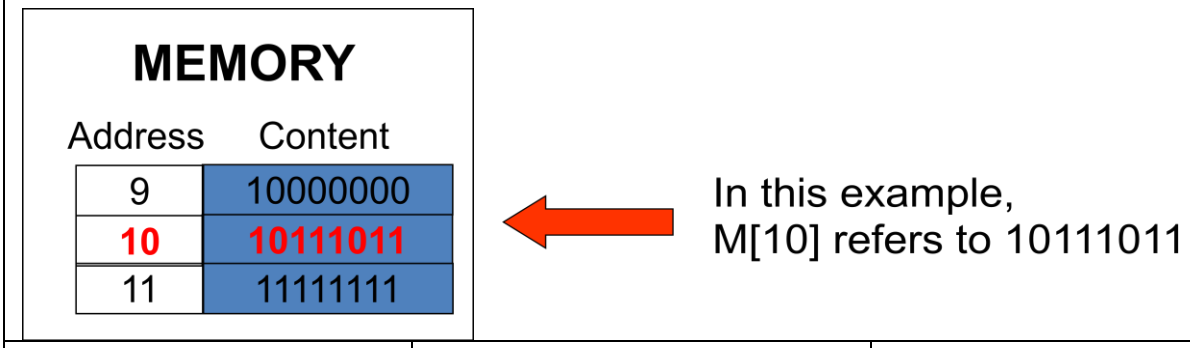
IR *Register showing named portions*

- Basic Symbols for Register Transfer

Symbol	**Description**	**Examples**
Letters (and numerals)	Denotes a register	MAR, R2
R followed by a number is referring to a register: **R2** = second register/register no 2 **R2**		
Parentheses ( )	Denotes a part of a register	R2(0-7), R2(L)
Arrow →	Denotes Transfer of information	R2 ←R1
R2 ←R1 = Stores the value of R1 into R2		
Comma ,	Separates 2 microoperations	R1 ←R2, R6 ←R7
R1← R2, R6← R7 ➔ Stores R2 into R1 and at the same time stores R7 into R6.		
Square brackets	Specifies an address for memory	M[R3]

M[R3] = content of the memory address in R3

M[10] = contents of memory address 10



Colon	Denotes termination of control function	K:

K: R1←R2

➔If K=1, then stores R2 into R1.

K1K2': R3←R2

- If K=1 and K2=0, then stores R2 into R3.

K: a control signal generated in the control unit, 0 or 1

MATHEMATICAL AND LOGICAL SYMBOLS

- Addition is indicated by the + sign:

  R1← R2+R3

  Add R2 and R3, stores in R1

  R2← R4+R1

  Add R4 and R1, stores in R2

- Subtraction is handled not with the minus sign but with complementing.

1's complement :

$R5 \leftarrow R3+\overline{R4}$ — R3 minus R4 in 1's complement

2's complement :

$R5 \leftarrow R3+\overline{R4}+1$ — R3 minus R4 in 2's complement

ARITHMETIC MICROOPERATIONS

Symbolic Designation	Description
$R0 \leftarrow R1 + R2$	Addition
$R0 \leftarrow \overline{R1}$	Ones Complement
$R0 \leftarrow \overline{R1} + 1$	Two's Complement
$R0 \leftarrow R2 + \overline{R1} + 1$	R2 minus R1 (2's Comp)
$R1 \leftarrow R1 + 1$	Increment (count up)
$R1 \leftarrow R1 - 1$	Decrement (count down)

COMPLEX LOGICAL SYMBOLS

- Content of R5 will be stored in R4 only **IF** both condition K1 **and** condition K2 are true:

If K1=1 **and** K2=1, then stores R5 into R4

Content of R5 will be stored in R4 **IF** either condition K1 **or** condition K2 were true, a **+** sign would be used:

In (K1 + K2),"+" means "**OR**"
In R1 ← R1 + R3, "+" means "**plus**"

- If **– then – else** is implemented with **commas** and **multiple colons**.
- If K1 true, then stores R4 into R6, else if K2 true, then stores R5 into R6, else store R7 into R6.