

Programmable Circuits (III)

A Loop for the Count (cont.)

- We can then leverage the mnemonics to write an assembly program to get the sum of 5, 4, 3, 2, 1, and store the sum in RA.
 - Initialize RB = 0 and RA = 5. Have a loop, in each iteration, the result of the sum of the values stored at RA and RB is assigned to RB, and then decrement RA's value by 1. As long as the value in RA is not zero, the loop continues. Put the result to RA after the loop terminates.
 - If using pseudocode, we can have something like

```
0    RB <= 0
1    RA <= 5
2    while (RA != 0) {
3        RB <= RA + RB
4        RA <= RA - 1
5    }
6    RA <= RB
```

- If using mnemonics to translate the above pseudocode into assembly program, we can have

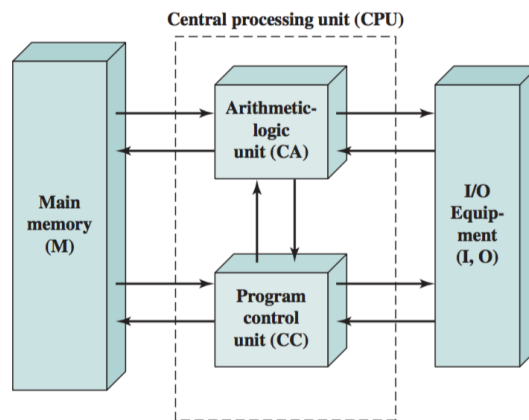
```
0    LOAD RB 0 # initialize RB = 0
1    LOAD RA 5 # initialize RA = 5
2    ADD  RA RB RB # Add RA and RB and save the result to RB
3    ADD  RA -1 RA # RA <= RA - 1
4    BRZ  6 # If RA = 1, result of above line is 0. (ALU result is 0,
CR=1) jump to line 6
5    BRA  2 # branch to line 2
6    LOAD RA RB
```

- We then can translate the above assembly program to machine instructions, which will be the program stored in Program Memory to calculate the sum of 5, 4, 3, 2, 1.

#	J1	J0	S1	S0	C1	C0	00	D3	D2	D1	D0	
0	0	d	1	0	0	1	1	d	d	d	d	(way2: S1S0=01 D3-D0=0000)
1	0	d	0	1	0	1	0	0	1	0	1	
2	0	d	0	0	0	0	1	d	d	d	d	
3	0	d	0	1	0	0	0	1	1	1	1	(way2: S1S0=11 D3-D0=dddd)
4	1	0	d	d	d	d	d	0	1	1	0	
5	1	1	d	d	d	d	d	0	0	1	0	
6	0	d	0	0	0	1	0	d	d	d	d	

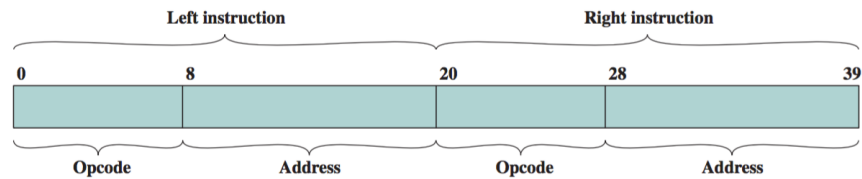
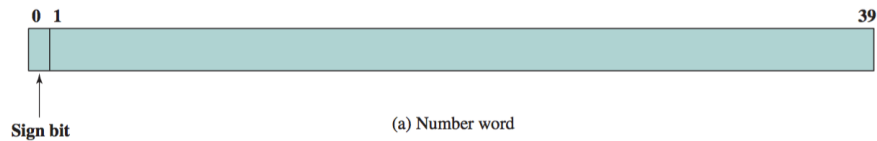
What is the IAS computer?

- We've talked about the programmable circuits. Let's take a look at the first electronic computer that could store programs in memory.
- This first electronic computer is called IAS computer, as it's the first electronic computer built at the Institute for Advanced Study (IAS) in Princeton.
- IAS computer is also call Von Neumann machine, as the paper describing its design was edited by John Von Neumann.
- IAS computer is the father of all modern computers, and its primary concept was storing the program in memory.
- Here is the structure of the IAS computer.



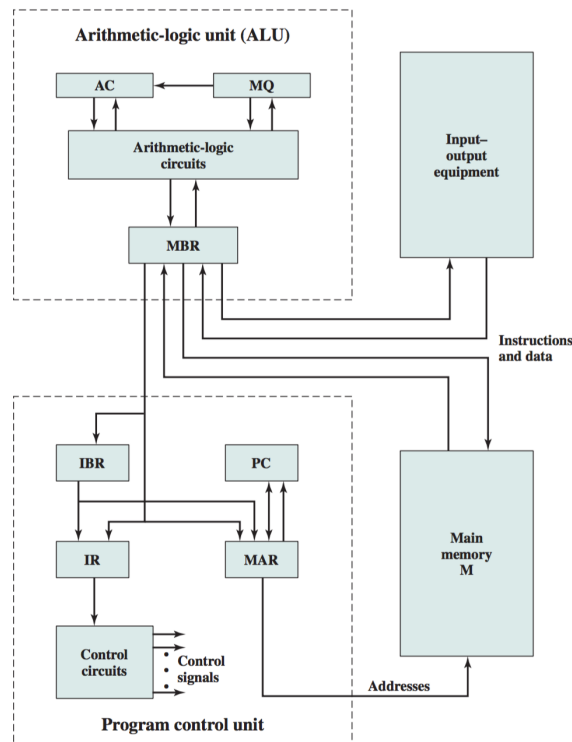
- It contains
 - A **main memory**, which stores both data and instructions
 - An **ALU** capable of operating on binary data (**CA: central arithmetical**)
 - A **control unit**, which interprets the instructions in the memory and causes them to be executed (**CC: central control**)
 - **I/O** equipment operated by the control unit
- IAS computer contains 4096 (2^{12}) **memory locations**. The memory locations are called, **words**.

- Each word is **40-bit long**, and could hold **one piece of data or two instructions**. (binary format)
 - Each number is represented by a sign bit and a 39-bit value.
 - Each instruction consisting of an 8-bit operation code (opcode) specifying the operation to be performed and a 12-bit address designating one of the words in memory (numbered from 0 to 4095)



Data Path

- Let's look into the CA and CC and see how different parts are connected.
- In addition to PC and IR, a few more registers are used. Although they have different names, they are used to store various information.



- **AC** (accumulator) and **MQ** (multiplier quotient): hold temporarily operands and results of ALU operations.
- **MBR** (memory buffer register): contains a word to be stored in memory or sent to the I/O unit, or is used to receive a word from memory or from the I/O unit.
- **IBR** (instruction buffer register): hold temporarily the right-hand instruction from a word in memory.
- **MAR** (memory address register): specifies the address in memory of the word to be written from or read into the MBR.