CALIFORNIA STATE UNIVERSITY FULLERTON

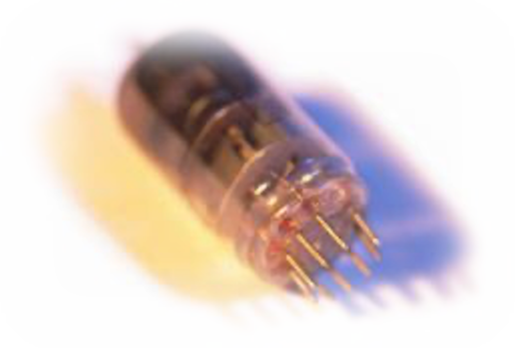
CPSC-440 Computer System Architecture

Lecture 3

Von Neumann Machines (IAS)

History of Computers

First Generation: Vacuum Tubes

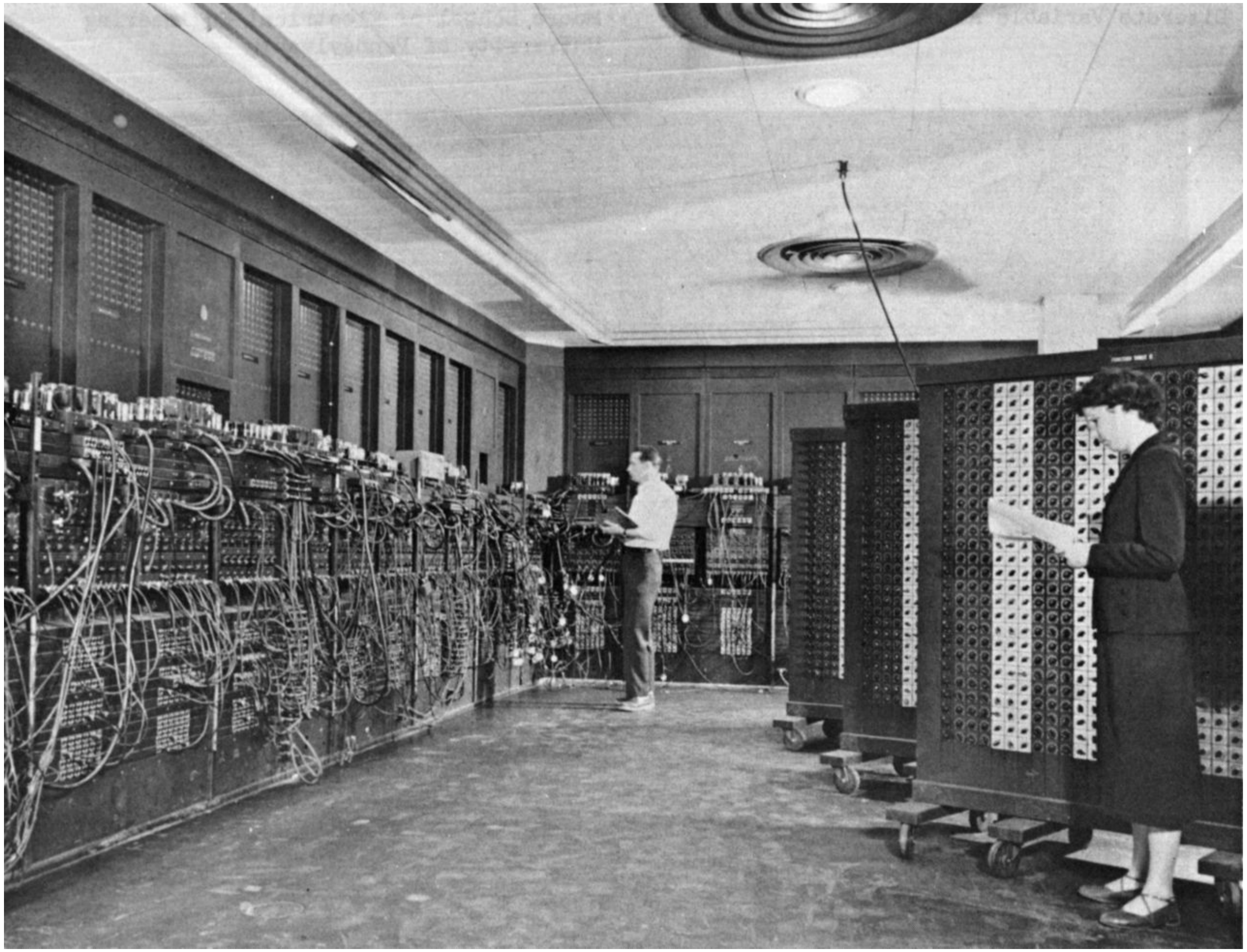


- ENIAC
  - Electronic Numerical Integrator And Computer
- Designed and constructed at the University of Pennsylvania
  - Started in 1943 – completed in 1946
  - By John Mauchly and John Eckert
- World's first general purpose electronic digital computer
  - Army's Ballistics Research Laboratory (BRL) needed a way to supply trajectory tables for new weapons accurately and within a reasonable time frame
  - Was not finished in time to be used in the war effort
- Its first task was to perform a series of calculations that were used to help determine the feasibility of the hydrogen bomb
- Continued to operate under BRL management until 1955 when it was disassembled

ENIAC

- Weighed 30 tons
- Occupied 1500 square feet of floor space
- Contained more than 18,000 vacuum tubes
- 140 kW Power consumption
- Capable of 5000 additions per second
- Decimal rather than binary machine
- Memory consisted of 20 accumulators, each capable of holding a 10 digit number
- Major drawback was the need for manual programming by setting switches and plugging/ unplugging cables

ENIAC



John von Neumann

EDVAC (Electronic Discrete Variable Computer)

- First publication of the idea was in 1945
- Stored program concept
  - Attributed to ENIAC designers, most notably the mathematician John von Neumann
  - Program represented in a form suitable for storing in memory alongside the data
- IAS computer
  - Princeton Institute for Advanced Studies
  - Prototype of all subsequent general-purpose computers
  - Completed in 1952

Structure of von Neumann Machine

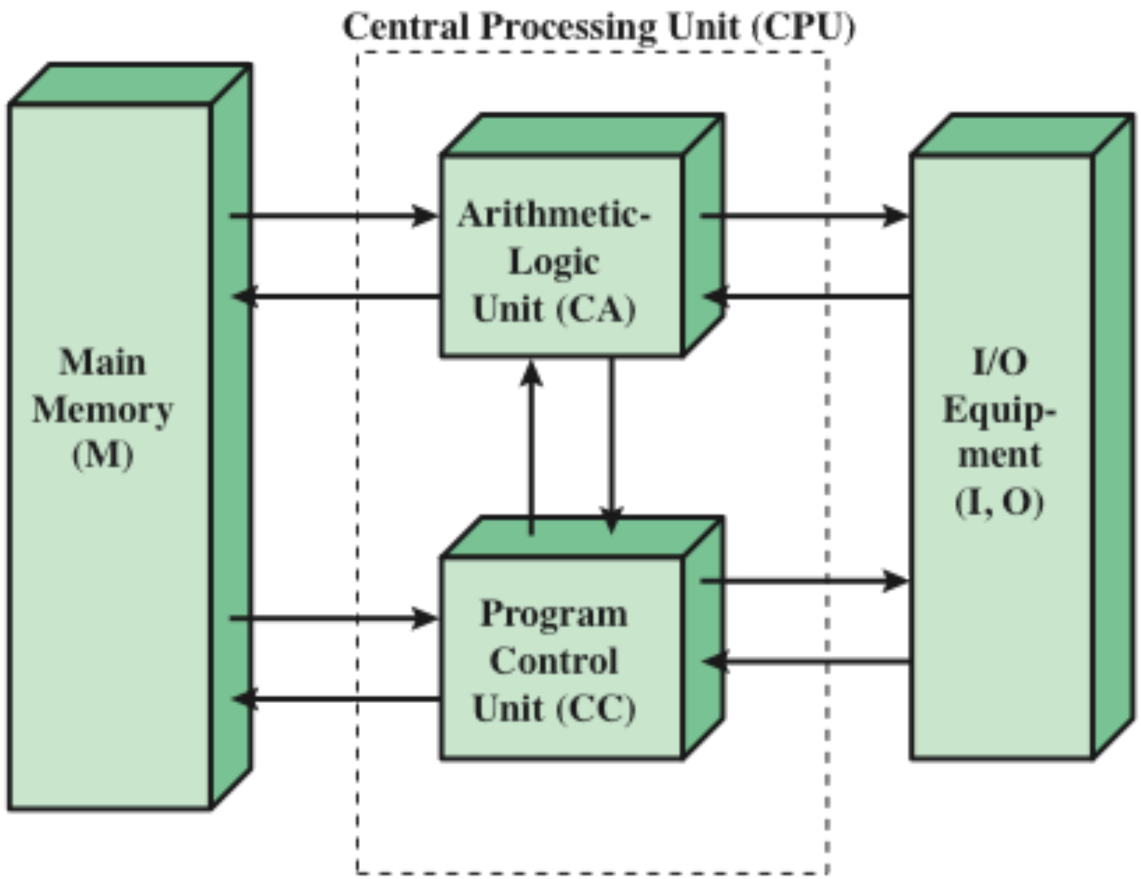


Figure 2.1 Structure of the IAS Computer

IAS Memory Formats

- The memory of the IAS consists of 1000 storage locations (called words) of 40 bits each
- Both data and instructions are stored there
- Numbers are represented in binary form and each instruction is a binary code

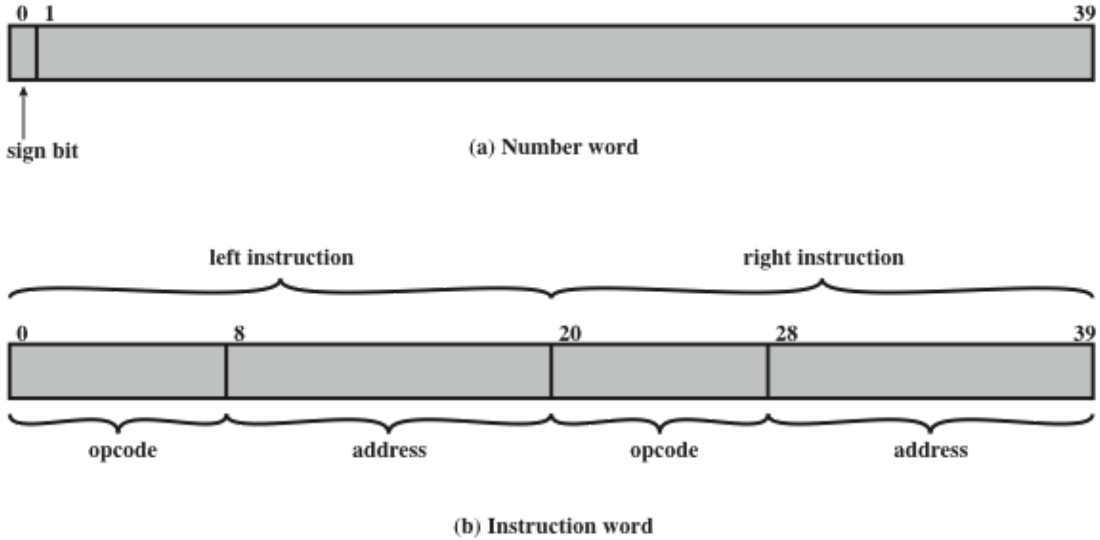
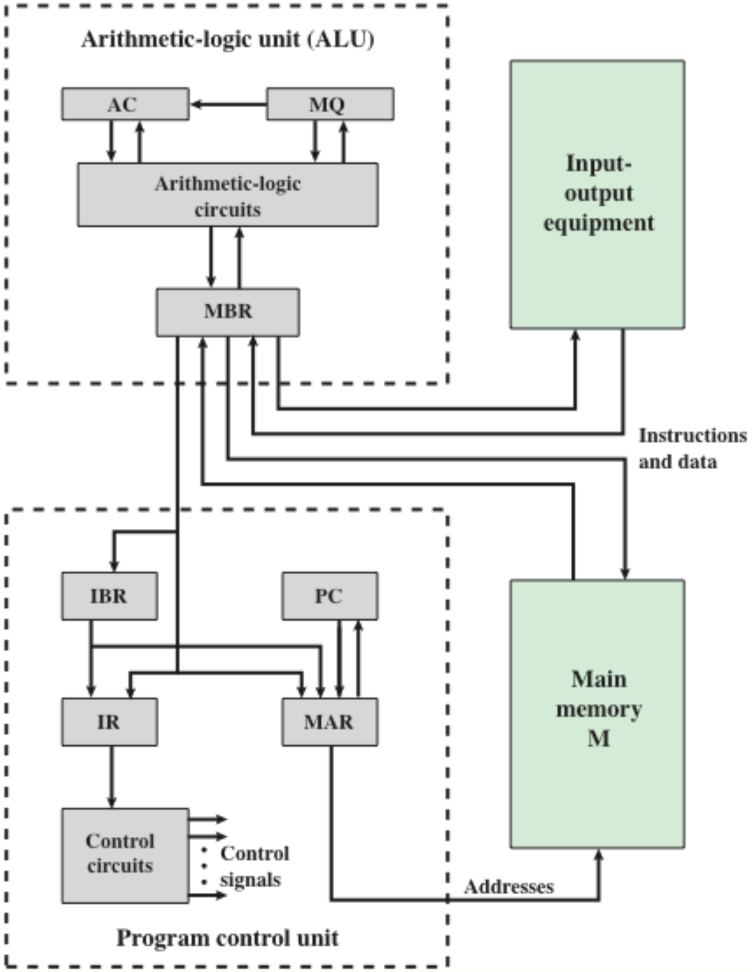


Figure 2.2 IAS Memory Formats

Structure of IAS Computer

Registers

- Memory Buffer Register (MBR)
  - Word to be stored/received in/from memory or I/O unit
- Memory Address Register (MAR)
  - Memory Address of the word to be (written from)/(read into) the MBR
- Instruction Register (IR)
  - Contains 8-bit opcode
- Instruction Buffer Register (IBR)
  - Temporarily holds the right-hand instruction
- Program Counter (PC)
  - Contains address of the next instruction pair to be fetched from memory
- Accumulator (AC) and Multiplier Quotient (MQ)
  - Employed to temporarily hold operands and results of ALU operations



Instruction Type	Opcode	Symbolic Representation	Description
Data transfer	00001010	LOAD MQ	Transfer contents of register MQ to the accumulator AC
	00001001	LOAD MQ,M(X)	Transfer contents of memory location X to MQ
	00100001	STOR M(X)	Transfer contents of accumulator to memory location X
	00000001	LOAD M(X)	Transfer M(X) to the accumulator
	00000010	LOAD –M(X)	Transfer –M(X) to the accumulator
	00000011	LOAD |M(X)|	Transfer absolute value of M(X) to the accumulator
	00000100	LOAD –|M(X)|	Transfer –|M(X)| to the accumulator
Unconditional branch	00001101	JUMP M(X,0:19)	Take next instruction from left half of M(X)
	00001110	JUMP M(X,20:39)	Take next instruction from right half of M(X)
Conditional branch	00001111	JUMP+ M(X,0:19)	If number in the accumulator is nonnegative, take next instruction from left half of M(X)
	00010000	JUMP+ M(X,20:39)	If number in the accumulator is nonnegative, take next instruction from right half of M(X)
Arithmetic	00000101	ADD M(X)	Add M(X) to AC; put the result in AC
	00000111	ADD |M(X)|	Add |M(X)| to AC; put the result in AC
	00000110	SUB M(X)	Subtract M(X) from AC; put the result in AC
	00001000	SUB |M(X)|	Subtract |M(X)| from AC; put the remainder in AC
	00001011	MUL M(X)	Multiply M(X) by MQ; put most significant bits of result in AC, put least significant bits in MQ
	00001100	DIV M(X)	Divide AC by M(X); put the quotient in MQ and the remainder in AC
	00010100	LSH	Multiply accumulator by 2; i.e., shift left one bit position
	00010101	RSH	Divide accumulator by 2; i.e., shift right one position
Address modify	00010010	STOR M(X,8:19)	Replace left address field at M(X) by 12 rightmost bits of AC
	00010011	STOR M(X,28:39)	Replace right address field at M(X) by 12 rightmost bits of AC

Table 2.1: The IAS Instruction Set

Table 2.1

The IAS Instruction Set

Example 1

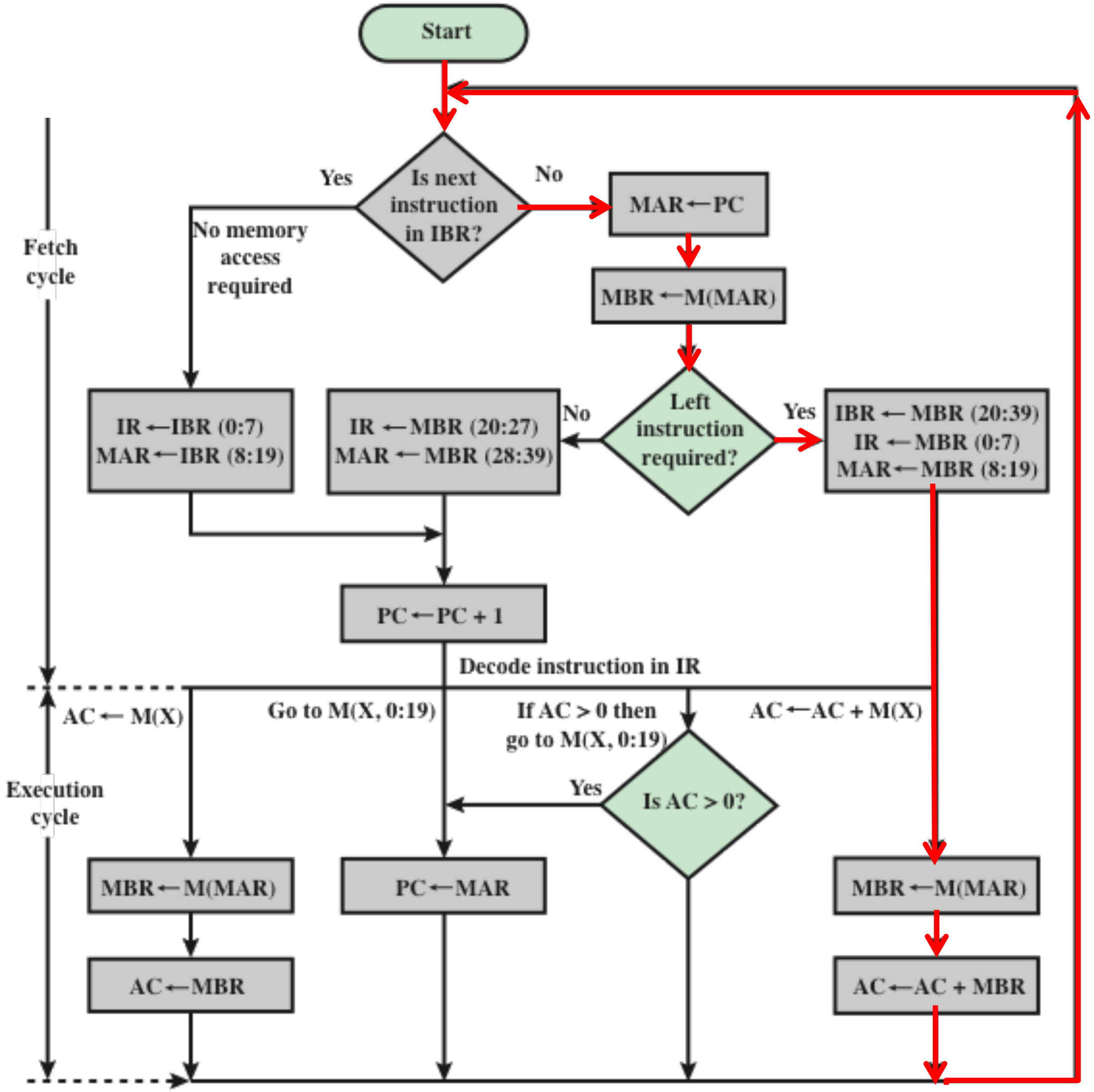
- What would the machine code instruction look like to add the contents of memory address 5CD (HEX) with the accumulator and stores the result back into the accumulator?

Left Instruction		Right Instruction	
0	8	20	28 … 39
0000 0101	0101 1100 1101	XXXX XXXX	XXXX XXXX XXXX
Opcode	Address	Opcode	Address

Symbolic	Description
ADD M(X)	Add M(X) to AC; put the result in AC

Example 1

Left instruction first



M(X) = contents of memory location whose address is X
(i:j) = bits i through j

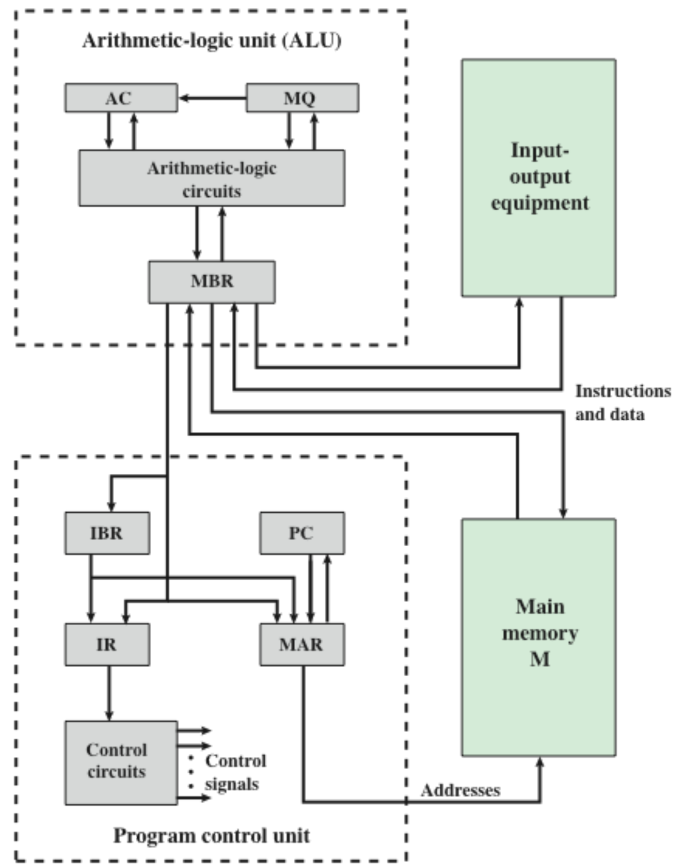


Figure 2.4 Partial Flowchart of IAS Operation

Example 1

The execution path would depend on the right instruction

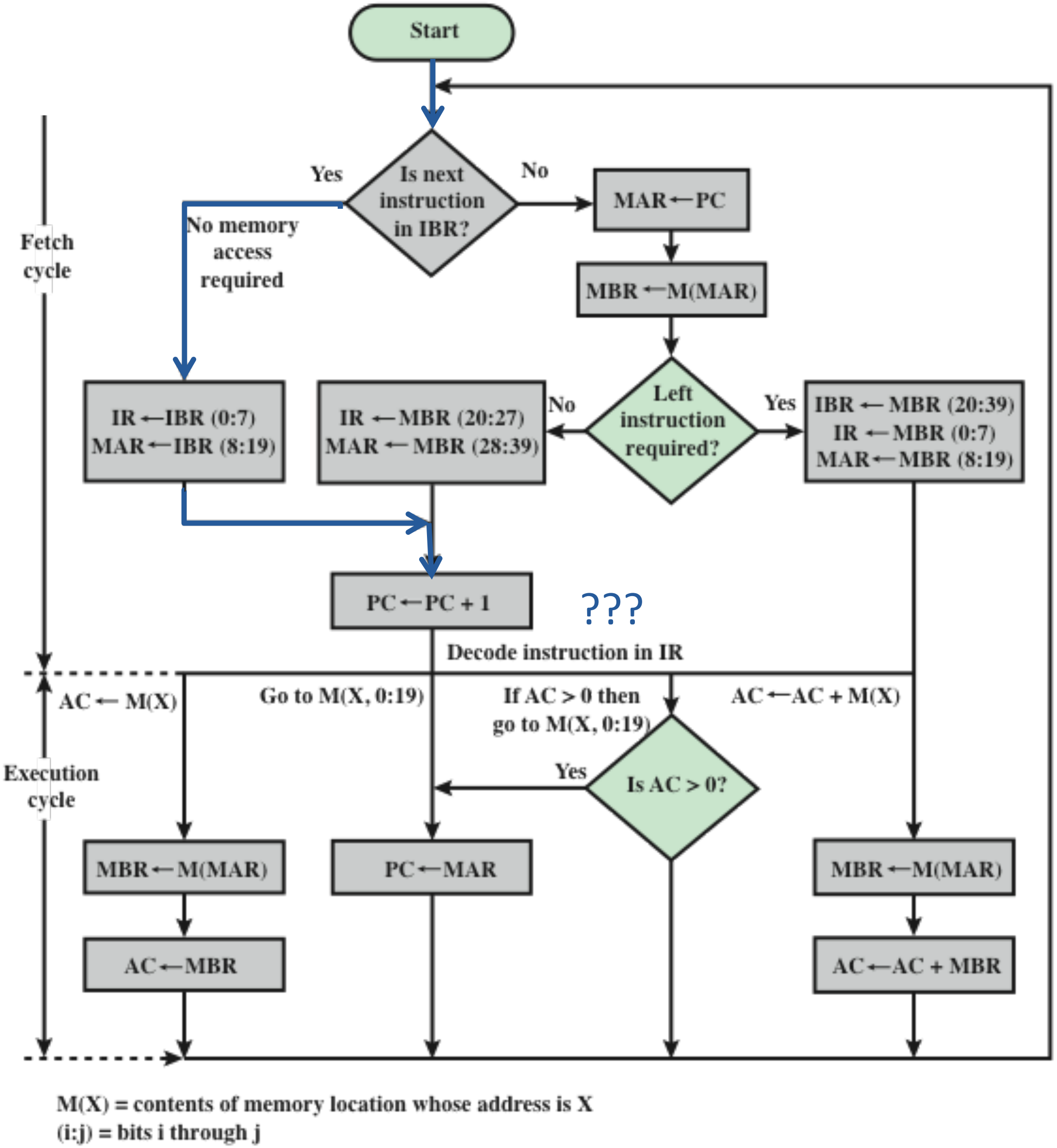
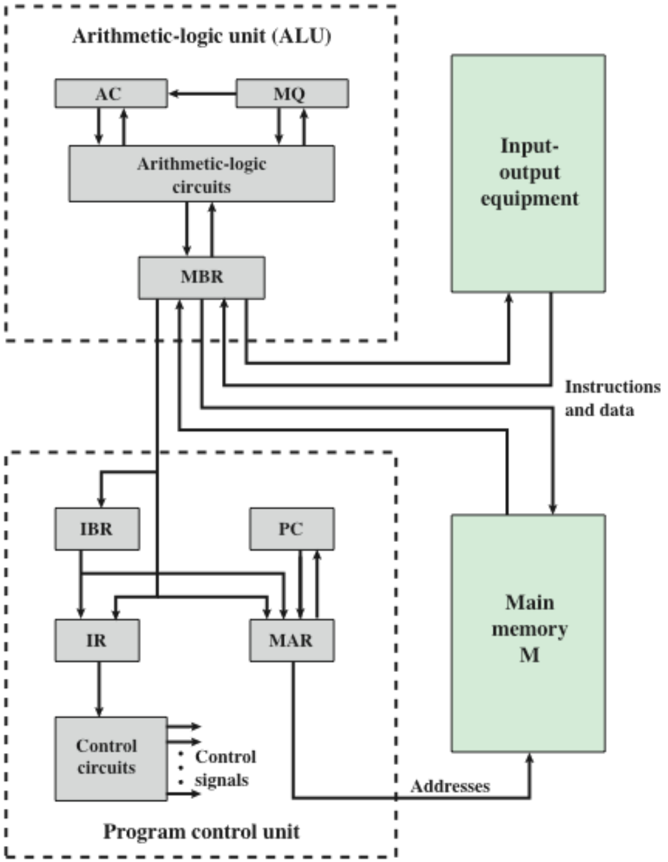


Figure 2.4 Partial Flowchart of IAS Operation



Example 2

- What is the assembly language code for the program:

	Left Instruction		Right Instruction
0	8	20	28 … 39
0010 0001	1100 0110 1111	0001 0100	XXXX XXXX XXXX
Opcode	Address	Opcode	Address

Address	Machine Code
06B	21C6F14XXX

NOTE: IAS doesn’t actually have an assembly language

Example 2

- What is the assembly language code for the program:

Left Instruction		Right Instruction	
0	8	20	28 … 39
0010 0001	1100 0110 1111	0001 0100	XXXX XXXX XXXX
Opcode	Address	Opcode	Address

Address	Machine Code
06B	21C6F14XXX

Address	Symbolic
06B	STOR M(C6F)
	LSH

Example 3

- Write an IAS program to compute the results of the following equation:

$$Y = \frac{N(N+1)}{2}$$

- Assume that the result of the computation doesn't overflow and N is a positive integer

Example 3

$$Y = \frac{N(N+1)}{2}$$

Location	Instruction/Value	Comments
0	<>	Constant (N) [initialized to some value]
1	1	Constant; Integer value = 1
2	2	Constant; Integer value = 2
3	0	Variable Y (initialized to integer zero)
4	0	Variable X (initialized to integer zero)
5L	LOAD M(0)	N → AC
5R	ADD M(1)	AC + 1 → AC; (N+1)
6L	STOR M(4)	AC → X; X=N+1
6R	LOAD MQ,M(4)	X→ MQ; MQ=N+1
7L	MUL M(0)	MQ*M(0) → N(N+1) → AC
7R	DIV M(2)	AC/2 → AC; AC=N(N+1)/2
8L	STOR M(3)	AC → Y; saving the Sum in variable Y
8R	JUMP M(8,20:39)	Jump to 8R; Done

Example 4

- Write an IAS program to compute the results of the following equation:

$$Y = \sum_{X=1}^{N} X$$

- Assume that the result of the computation doesn't overflow, and that X, Y, and N are positive integers

Example 4

$$Y = \sum_{X=1}^{N} X$$

Location	Instruction/Value	Comments
0	<>	Constant (N) [initialized to some value]
1	1	Constant (loop counter increment)
2	1	Variable i (loop index value; current)
3	1	Variable Y = Sum of X values (Initialized to One)
4	LOAD M(0)	N → AC (the max limit)
5L	SUB M(2)	Compute N–i → AC
5R	JUMP + M(6,20:39)	If AC > 0 [i < N] then jump to 6R
6L	JUMP M(6,0:19)	Loop here (HALT)
6R	LOAD M(2)	i<N so continue; Get loop counter i
7L	ADD M(1)	i+1 in AC
7R	STOR M(2)	AC → i
8L	ADD M(3)	i + Y in AC
8R	STOR M(3)	AC → Y
9L	JUMP M(5,0:19)	Jump to 5L

Homework Problems

- Problems are available on Canvas