Chapter 2 - Computer Evolution and Performance

A Brief History of Computers (Section 2.1) on pp. 16-38

Let's take a quick look at the history of computers so that we know where it all began. Further, we will find that most things have dramatically changed while some have not.

ENIAC (Electronic Numerical Integrator and Computer)

Designed by:	John Mauchly and John Presper Eckert
Reason:	Response to wartime needs
Completed:	1946
Specifics:	30 tons, 15,000 sq ft, 18000 vacuum tubes, 5000 additions/sec decimal machine with 20 accumulators each accumulator could hold a 10-digit decimal number

Programming the ENIAC involved plugging and unplugging cables. If however, the program could be stored somehow and held in memory along with the data, this process would not be necessary.

This is known as the "stored-program concept." Much of the credit was given to John von Neumann who helped engineer a computer (IAS) that utilized this concept and was completed in 1952.

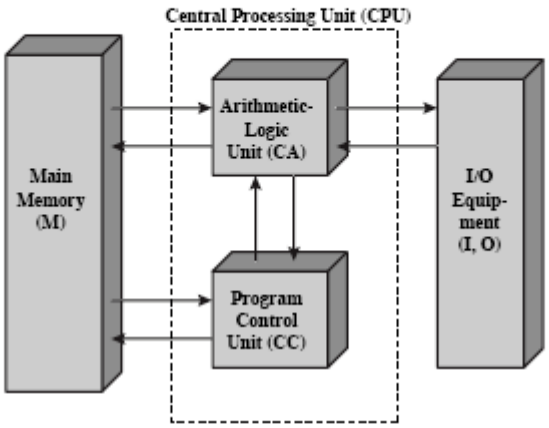

Central Processing Unit (CPU)
Main Memory (M)
Arithmetic-Logic Unit (CA)
Program Control Unit (CC)
I/O Equip-ment (I, O)

Figure 2.1 Structure of the IAS Computer

What is amazing is that today, most computers have the same general structure and function. Wow!!

IAS Computer

Designed by:	John von Neumann
Reason:	Incorporate the stored-program concept
Completed:	1952
Specifics:	1000 words (40-bit words) of storage, both data and instructions are stored, binary representations

A word of storage can represent either an instruction or a number. The format for each is as follows:

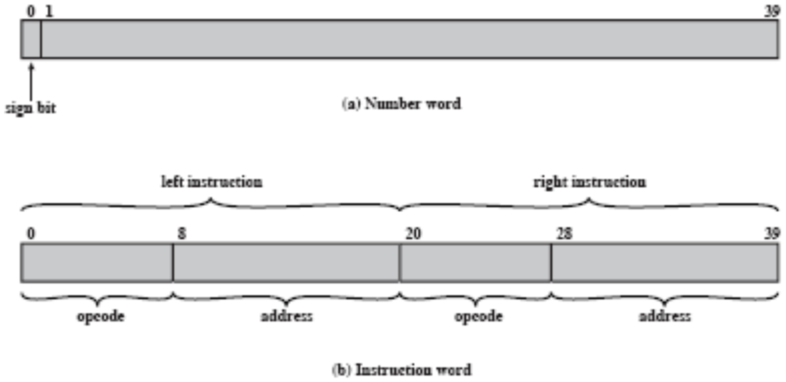


Figure 2.2 IAS Memory Formats

Number: bit 0 is sign bit, bits 1-39 are the number

Instruction: bits 0-7 opcode of left instruction
bits 8-19 address of one of the words of memory
bits 20-27 opcode of right instruction
bits 28-39 address of one of the words of memory

Note: An instruction word really contains two instructions (left and right instructions).

Before looking at Figure 2.3 on p.19, let's define the following terms because they are still applicable today:

- Memory Buffer Register (MBR) - used to store/retrieve a word to/from memory
- Memory Address Register (MAR) - specifies the address used to store/retrieve a word to/from MBR
- Instruction Register (IR) - contains opcode (8-bit) of the current instruction being executed
- Instruction Buffer Register (IBR) - used to hold right instruction
- Program Counter (PC) - contains address of next instruction to be executed
- Accumulator (AC) and Multiplier Quotient (MQ) - holds results of ALU operations. For a multiply of 2 words, the AC holds the most significant 40-bits and the MQ holds the least significant 40-bits.

P#1: Using the figure below, explain the instruction fetch cycle using the above terminology.

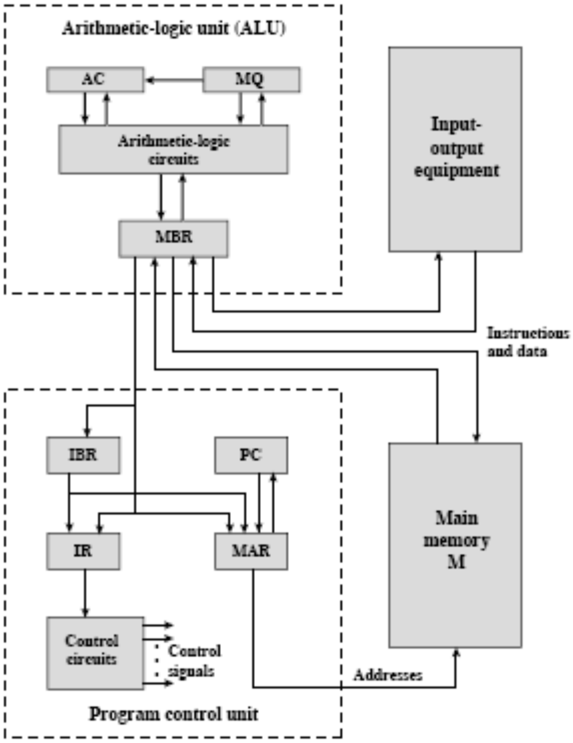
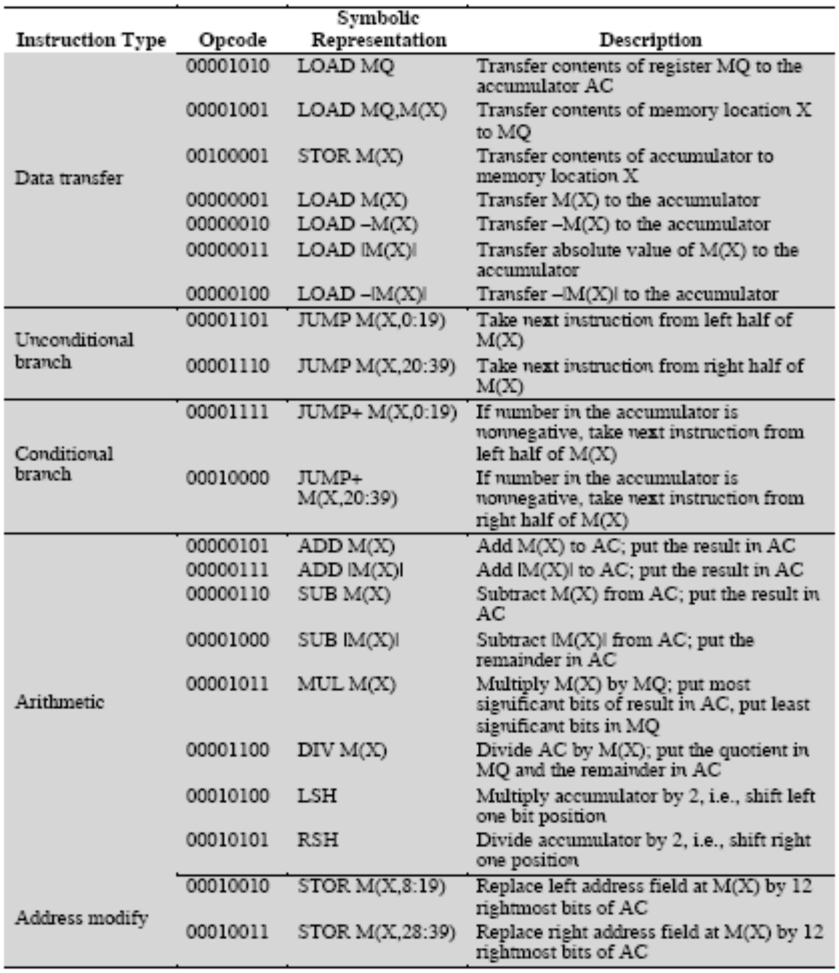


Figure 2.3 Expanded Structure of IAS Computer

The IAS had 21 instructions as follows:

Table 2.1 The IAS Instruction Set



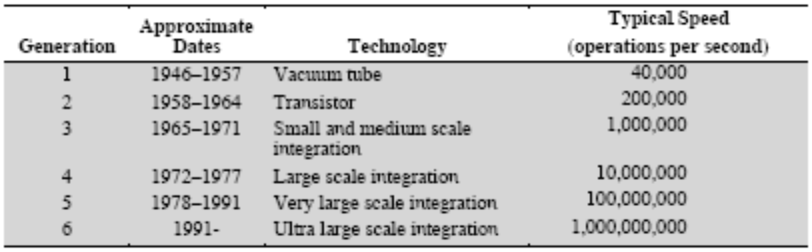
Instruction Type	Opcode	Symbolic Representation	Description
Data transfer	00001010	LOAD MQ	Transfer contents of register MQ to the accumulator AC
	00001001	LOAD MQ,M(X)	Transfer contents of memory location X to MQ
	00100001	STOR M(X)	Transfer contents of accumulator to memory location X
	00000001	LOAD 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

How does the IAS computer add two numbers if we suppose the numbers are stored in memory locations 100 and 101, and the sum is to be saved in memory location 102?

What does the instruction cycle for adding the two numbers above look like?

Computer Generations are as follows:

Table 2.2 Computer Generations



Generation	Approximate Dates	Technology	Typical Speed (operations per second)
1	1946–1957	Vacuum tube	40,000
2	1958–1964	Transistor	200,000
3	1965–1971	Small and medium scale integration	1,000,000
4	1972–1977	Large scale integration	10,000,000
5	1978–1991	Very large scale integration	100,000,000
6	1991-	Ultra large scale integration	1,000,000,000

We usually refer to a single transistor as a "discrete component." The computers of the 50s and 60s were made up of these discrete components such as transistors, resistors, and capacitors which were soldered to circuit boards. SSI produced the invention of the integrated circuit.

Integrated circuits are made from a wafer of silicon. Wafers are broken up into chips and each chip can contain gates, memory cells, and input/output attachment points. When this process first began, only a few gates or memory cells could be put on a chip and therefore, the name SSI. Figure 2.8 shows the growth of components per chip over the years.

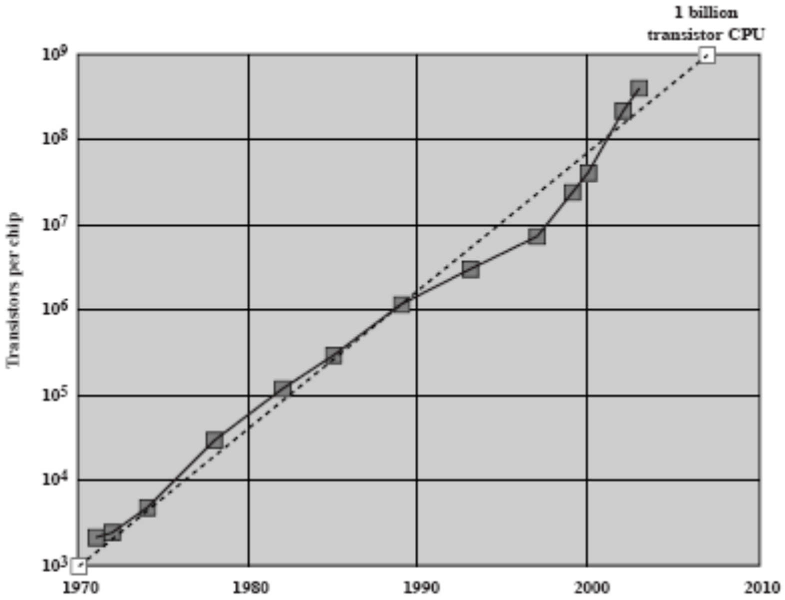


Figure 2.8 Growth in CPU Transistor Count [BOHR03]

Let's skip to Appendix B – Digital Logic.