Electronics-Robotics rev 8.7.21

Chapter Zero - Internal workings of a computer

1. Number systems

Numeral systems conversion table

Decimal Base-10	Binary Base-2	Octal Base-8	Hexadecimal Base-16
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17
24	11000	30	18
25	11001	31	19

26	11010	32	1A
27	11011	33	1B
28	11100	34	1C
29	11101	35	1D
30	11110	36	1E
31	11111	37	1F
32	100000	40	20

Example of use of all this:

Each of your computers which attaches to the Internet has a MAC (Media Access Control) address. MAC addresses have 48 bits.

(There are 281 trillion possible MACs) they are written as hex numbers letters may be uc or lc separated by: or – or not at all.

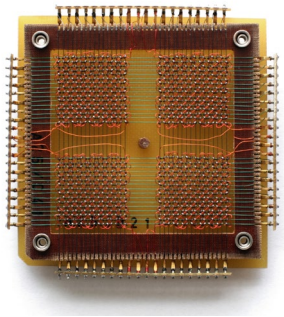
5c:f9:38:a5:46:2c OR 5c-f9-38-a5-46-2c OR 5C:F9:38:A5:46:2C OR 5CF938A5462C

2. We're going to look at a specific early minicomputer. We will look at its hardware and then how to program it.

DEC PDP 8i
Digital Equipment Corporation Programmable Data Processor
(Introduced: 1968 Cost: $18,500. In today's dollars: $150,000)
It was not called a computer, but a programmable data processor because computers were horrendously more expensive in those days.

Hardware:
Memory: 4096 words (each 12 bits) CORE memory

Each memory location has an address which is permanent and contents which can be changed
Many kinds of memory –
Internal: CORE -very fast, but costly
External: Paper Tape – cheap but slow
Later versions of this computer had magnetic tape and hard disk available at extra cost



The central processing unit (CPU) has some registers (a register is location in a store of data, used for a specific purpose and with quick access time.)
Program Counter contains the address of the NEXT command to be executed
Instruction register contains the current instruction
Memory Address register contains the address of memory location currently being accessed.
Memory Buffer contains the contents of currently addressed memory location

Accumulator - special register which is involved in most manipulations

This particular computer had a small, but extremely flexible instruction set.
Each instruction has an operation code (OPCODE) and many have an operand
(the quantity on which the opcode operates.)

3. Developing a program for the DEC PDP 8i
Here is an example:
We have two numbers in memory. We want to add them together and store the answer in memory.

Software:
We need to use 4 commands:
Clear Accumulator
Add to Accumulator
Store Accumulator contents in memory
Halt

Here are the specific commands we will need – there are a lot more:

Command	Binary	Octal	Mnemonic
Clear the accumulator	111 010 000 000	7200	CLA
Twos complement add	001 000 000 000	1000	TAD (address)
Deposit and clear Accumulator	011 000 000 000	3000	DCA (address)
Halt	111 100 000 010	7402	HLT

I will write the program as follows:
note that anything following the / is a comment and is ignored by the computer.

```
Page 1      /starting address – avoid page 0 for technical reasons.
CLA         /clear the accumulator
TAD A       /Add the first number
TAD B       /Add the second number
DCA C/store the answer
HLT         /stop working
*300        /this resets the address – keep data away from commands.
A, 2        /The first number
B, 3        /The second number
C,          /A place to hold the answer
```

This is what the assembler will generate:

```
Loc   Contents
0200  7200        CLA
0201  1300        TAD A
0202  1301        TAD B
0203  3302        DCA C
0204  7402        HLT

0300  0002        A,2
0301  0003        B,3
```

0302 0000 C,0

												XXXXXXX
Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit
2048	1024	512	256	128	64	32	16	8	4	2	1	Value

Incidentally, 12 bits can directly address 4096 locations (0-4095)
However, some instructions leave only 5 bits for the address of the operand, so addressing must be within the current page. A page is 128 locations or 200 octal addresses. If the operand is on another page you must resort to indirect addressing. How indirect addressing works is beyond the scope of this lesson.

FACTS:

1. The computer understands only binary numbers. They are written as hex or octal because those are far easier for humans to deal with, and much less error prone. However, they exist in memory as binary numbers.

2. Programs have been written which can translate the more human-friendly commands into the appropriate binary form. They are called assemblers, or compilers.

3. The binary numbers representing the instructions must be loaded into the computer's memory for the program to run.

4. More advanced programs will take what you write in a special language (such as BASIC or C), convert it into the appropriate binary representation, load it into the computer's memory, run it, and return the result. This makes using a computer not trivial, but WAY easier.