

## 1a Difference between Computer Architecture and Computer Organization:

SN	Parameters	Computer Architecture	Computer Organization
1	Define	It defines hardware design and functional behaviour of a computer system.	It defines physical implementation of a computer system.
2	Focus	It focuses that what a computer does.	It focuses how computer works.
3	Indicates	It indicates hardware part of a computer system.	It indicates the performance.
4	Language	It deals with high level language.	It deals with low level language
5	Sequence	It comes first.	Depend on architecture organization is set.
6	Type	There are less types of computer architecture.	One architecture has many organizations.
7	Cost	<del>It is more expensive.</del>	It is less expensive than architecture.

1a

8) Example:

Computer architecture: set of instructions,  
no. of bits used to represent a data.

Computer organizations: ~~INS, EDVAC, UNIVAC  
etc.~~ (8)

1b

RTL: RTL stands for Register Transfer Language used to transfer data from one register to another.

Microoperation is the operation executed on data stored in registers. These are 4 types of microoperations.

i) Register Transfer Microoperation

It is used to transfer data from one register to another.

RTL code: ~~R2  $\leftarrow$  R1~~

(Storing R1's data to R2.)

## ii) Arithmetic

This is used to perform basic arithmetic operations:

Operation:

Addition

Subtraction

Multiplication

Division

increment

decrement

## Microoperation

RTL code:

$$R_1 \leftarrow R_1 + R_2$$

$$R_1 \leftarrow R_1 - R_2$$

$$R_1 \leftarrow R_1 * R_2$$

$$R_1 \leftarrow R_1 / R_2$$

$$R_1 \leftarrow R_1 + 1$$

$$R_1 \leftarrow R_1 - 1$$

## iii) Logical Microoperation

This is used to perform logical operations.

Operation:

AND

OR

XOR

Complement

RTL code:

$$R_1 \leftarrow R_1 \text{ and } R_2$$

$$R_1 \leftarrow R_1 \text{ or } R_2$$

~~$$R_1 \leftarrow R_1 \text{ XOR } R_2$$~~

~~$$R_1 \leftarrow R_1'$$~~

## iv) Shift Microoperation

This is used for sequential data. There are 3 types of shift microoperations.

This logical shift goes through initial port.

a)  $R_1 = 1001$

$\text{shl}(R_1) = 0010$

$\text{shr}(R_1) = 0100$

where,  $\text{shl} = \text{shift logical left}$   
 $\text{shr} = " \rightarrow \text{shift right}$

b) Circular shift

This rotates data around two end bits.

There is no loss of bits.

$R_2 = 1000$   
~~Girl~~  $(R_2) = 0001$

~~$\text{arr}(R_2) = 0100$~~

c) Arithmetic shift

This shifts the sign bit.

$R_3 = 1100$   
 $\text{ASL}(R_3) = 1000$   
 $\text{AST}(R_3) = 1110$

$CABX -$   
A  
B)  
C

given,  $\frac{2b}{Y = \frac{AXB - C}{E + F}}$ , postfix AB-C-X FFH

22-8-5  
= 9

a) using 0 address instruction format:

comments:

PUSH A	$TOP \leftarrow A$
PUSH B	$TOP \leftarrow B$
MUL	$TOP \leftarrow AXB$
SUB	$TOP \leftarrow AXB - C$
PUSH E	$TOP \leftarrow E$
PUSH F	$TOP \leftarrow F$
SUM	$TOP \leftarrow E + F$
DIV	$TOP \leftarrow (AXB - C) / E + F$
POP Y	$Y \leftarrow TOP$

cache memo  
built using  
length  
address sp  
ag field

= 4

b) using 1 address instruction format

comments

LOAD E	$Acc \leftarrow E$
ADD F	$Acc \leftarrow E + F$
STORE T1	$T1 \leftarrow Acc$
LOAD A	$Acc \leftarrow A$
MUL B	$Acc \leftarrow AXB$
SUB C	$Acc \leftarrow AXB - C$
DIV T1	$Acc \leftarrow Acc / T1$
STORE Y	$Y \leftarrow Acc$

$$Q = \begin{smallmatrix} 0 & 1 & 0 & 0 & 1 \\ - & 1 & 0 & 1 & 1 \end{smallmatrix}$$

Ashr

38

$$\begin{array}{l} Q = 10110 \\ Q = 0101 \\ n = 4 \\ A = 00000 \end{array} \quad = M \quad \cdot \quad -M = 01001$$

$$, Q-1 = 0$$

$n$	$A$	$Q$	$Q-1$	Comments
4	00000	0101	0	initial
	01001	0101	0	since, $Q(Q-1) = 10$ , $A = A-M$ ,
3	00100	1010	1	Ashr( $AQ(Q-1)$ ), $n--$
	11011	1010	1	since, $Q(Q-1) = 01$ , $A = A+M$
2	11101	1101	0	Ashr( $AQ(Q-1)$ ), $n--$
	00110	1101	0	since, $Q(Q-1) = 10$ , $A = A-M$
1	00011	0110	1	Ashr( $AQ(Q-1)$ ), $n--$
	11010	0110	1	since, $Q(Q-1) = 01$ , $A = A+M$
0	11101	0011	0	Ashr( $AQ(Q-1)$ ), $n--$
	10110			
	011			
	11			
	10			

③

$$AQ = 111010011 = 45$$

product  $= 000101101 = 45$

anner



VHDL code for full adder using structural  
modelling:

// half adder

```
library ieee;
use ieee.std_logic_1164.all;
entity half_adder is
port (
    h-a, h-b : in std_logic;
    h-c, h-s : out std_logic);
end half-adder;
```

Architecture of half-adder is

begin

```
    h-s <= h-a xor h-b;
```

```
    h-c <= h-a and h-b;
```

```
end arch-half-adder;
```

library ieee std logic 11

use ieee.std\_logic\_1164.all

entity full-adder is

port

```
    f-a, f-b, fcin: in std_logic;
```

```
    f-out, fsum: out std_logic);
```

```
end full-adder;
```

architecture arch-full-adder is

~~component~~ signal component half-adder;

```
    h-a, h-b : in std_logic;
```

id = 22 - 8  
= 9

ive co  
is bl  
ord  
add  
to

AA

$h\text{-}c, h\text{-}s : \text{std-logic}^{\text{out}}$  ;  
end component ;

signal  $s_1, s_2, s_3 : \text{std-logic}$  ;  
begin

U1: half-adder port map ( $s_1, s_2, f\text{-}a, f\text{-}b$ ) ;  
U2: half-adder port map ( $s_1, f_{in}, s_3, f_{sum}$ ) ;  
 $f\text{-cout} \leftarrow s_1 \text{ or } s_3$  ;

end arch-full-adder ;

(8)

Q. b.

There are two category of registers.

a) User visible Registers

These are visible to user. User can use it to store data.

i) data register: used to store data

ii) address register: used to store address of instruction or data.

iii) general purpose register: used to store both data and instruction.

iv) condition register: used to store flags.

b) Status and Control Registers.

i) Program Counter (PC)

It holds the address of upcoming instruction CPU get information what is next instruction from here.

ii) Memory Address Register (MAR)

It holds the address of memory location of a data. We can set the address of data from CPU.

eld = 22 - 8 - 5  
9

live cache in  
is built u  
word length  
address  
tag fi

5)

4b

iii) Memory Buffer Register (MBR)

It is used to store data that is going to or from memory.



iv) Instruction Register

It is used to store the instruction that is currently fetched.

5b  
Difference between Microprogrammed and Hardwired control unit:

	Hardwired Control Unit	Microprogrammed control
1. Parameters define	The control signal are designed by hardware logic circuits.	The control signals are stored in Microprogram memory.
2. Complexity	It is complex to design.	It is easier to design.
3. Speed	It is faster.	It is slower.
4. Decoder	It has decoder in design.	There is no use of decoder.
5. Use	It is used in RISC.	It is used in CISC.
6. Update	It is difficult to update, we have to change the structure of hardware.	It is easier to update.
7. Flexibility	This is less flexible.  (8)	This is more flexible.
8. Signal generate	Control signal is generated by opcode.	Control signal is generated by program.

field = 22 - 3 - 5

itive cache is built word len address r tag

DS)

P  
Principle

5a

Modern design principles speeds up a computer.

The principles used in designing computer are:

a) Parallelism

Parallelism supports multiple task at once within the same speed and same time. So recent computer uses parallel CPU concept that increases efficiency.

b) Pipelining

Pipelining is used in Reduced Instruction Set Computers (RISC). It is used in segment registers and uses piping techniques that makes faster execution.

c) Multicores

Using multicores in microprocessor makes whole the computer system fast. Multiple task can be done using this. A computer which has multicores is more powerful than simple computer.

#### d) Multiprocessing

Multiprocessor means many processor in one computer system. It processes many instruction at once that supports simultaneous work. It increases overall capacity of a computer system.

Today's computers become more powerful, faster, efficient and accurate due to following these techniques.

(1)

= 512 blocks per sec

Q6

Normalization in floating point operation is the process to store the floating point into its approximation value.

Let's take an example.

$$x = 70.625 = 1000110101 \cdot 10^6$$

exponent  
significand

While storing floating point number computer has to store only significand and exponent value.

- Significand indicates sign and value and exponent indicates the point of decimal position
- 1 is always fixed at start, after then also fixed. 2 is also fixed. (6)
- Only significand and exponent occurs in registers that save the space.

for 32 bit data =  $x = 70.625$

Sign(1bit)	Exponent(8)	Significand (23) bit
1	$127 + 6 = 133$ 10000101	$1000110101 \cdot 10^6$ $= 000110101$

VHDL stands for Very High Speed Hardware Definition Language. It is one of the recent trend used in hardware definition. It is widely used.

example. VHDL code for and gate

library IEEE;  
use IEEE.STD\_LOGIC\_1164.all;

entity and\_gate is

port (

a,b : in STD\_LOGIC;

c : out STD\_LOGIC);

end and\_gate;

architecture arch\_and\_gate is

begin

c <= a and b;

end arch\_and\_gate;

VHDL is extended version of HDL

(Hardware Definition Language) that increases the efficiency of hardware definition. It is used to design circuits.

## IC Harvard Architecture.

Harvard Architecture has designed by Harvard. It has two different memory for storing data and instruction.

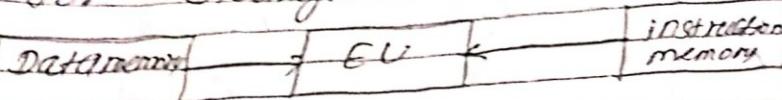


Figure: Harvard Architecture.

- It is mostly used in today's computer.
- It is used in RISC that supports pipelining.
- It is faster as there is no bottleneck between data and instructions that occur in Von Neumann architecture.

### Disadvantages:

- It uses more memory space even waste of memory.
- It is more costlier than Von-Neumann architecture.

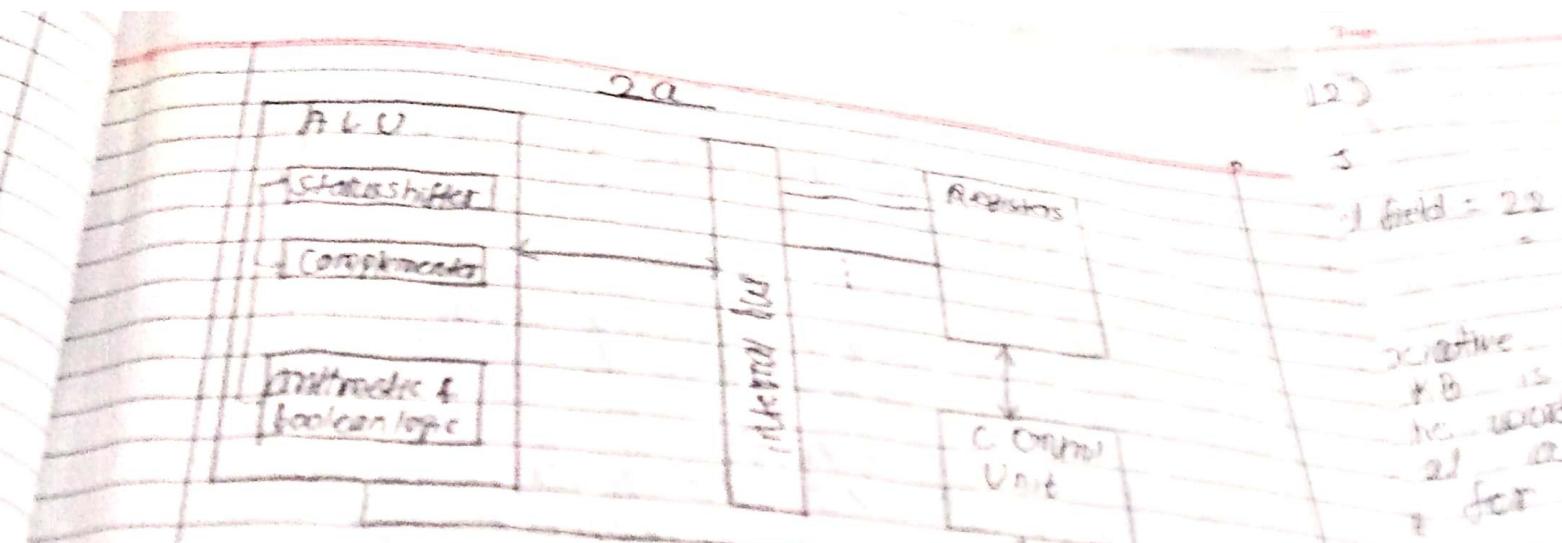


Fig: block diagram of CPU

CPU stands for Central Processing Unit that is the core of Computer Systems.

as Arithmetic Logic Unit

It is full form of ALU that performs arithmetic and logical operations. It has shifter that performs shifting operations, complementer complements the first data or oprands. Arithmetic and Boolean logic perform addition, subtraction, Boolean operation, AND, OR and other operations.

12)

13

$$\text{3) field} = 22$$

associative  
GKB is  
The word  
-al for  
for

3a

$$\begin{array}{l} M = 5 \\ Q = 12 \\ n = 4 \end{array} \quad \begin{array}{l} = 0101 \\ = 1100 \\ A = 0000 \end{array} \quad - M = 1011$$

n	Q A	Q	Comment
4	0000	1100	initial condition
	0001	100-	shl AQ
	1100	100-	A = A - M
3	0001	1000	since, A < 0, Q0 = 0, A = AM, n -
	0011	000-	shl AQ
	1110	000-	A = A - M
2	0011	0000	since, A > 0, Q0 = 0, rest of A, n -
	0110	000-	shl AQ
	0001	000-	A = A - M
1	10001	0001	since, A > 0, Q0 = 1, n -
	0010	001-	shl AQ
	1001	001-	A = A - M
0	0010	0010	since, A < 0, Q0 = 0, n - rest of A

2

13

(Q) : Quotient (Q) = 0010 = 2  
 remainder (A) = 0010 = 2

