Date: / /

COMPUTER ORGANIZATION & ARCHITECTURE

Structure → components → CPU → Registers
→ ALU
→ CU → Sequencer → Control ROM → generates control word of 12 bits
→ I/O
→ Memory.
→ Interconnection

Function → how these components perform their functions.

Computer Architecture and organization
⇓
Structure.

⇓
Function.
→ Data movement
→ Computer Processing.
→ I/O.
→ Storage.

Example:-
⇒ For multiplication.
→ deciding either to use multiplication unit or addition unit is architecture problem
→ deciding how the unit will perform multiplicat is organization problem.

Storage
↓
write
↓
address
+
data.

read.
↓
address.

Control :- Controls the operations of the system.

Date: / /

Control Unit :- Controls the signal.
Generates the 12 bit code

Computer → CPU → CU → Sequence
↳ Control memory.
↳ Control unit registers and decoders

Multiplex Bus- Signal (Control signal) is generated to tell wether the bus will act as address, data or control bus.

It is cheap but complex.

Moores' law was valid till a limit but now not more b/c it causes the devices to heat up etc.
- Thus multicore idea was implemented.

In Multiprocessing / processing frequency of clock cycle determines how efficient it is.

Smaller the clock cycle, faster is the processing and best is the processor.

① Hardware : cannot be reprogrammed ⇒ fixed hardware
e.g. Hardwired

② Software + Software (CO-design) :-
e.g. general purpose computer.

op-code ← → operand

LDA 5H
ADD 6H.

Mnemonics → converted into binary by compiler
e.g. LDA = 0000
ADD = 0001

→ Size of computer depends on the memory.
If a memory has 8 compartments we need 3 bit bus. (i.e. $2^3 = 8$) to access each memory address

→ Interrupts.
normal execution of process is disturbed due to an error or call.

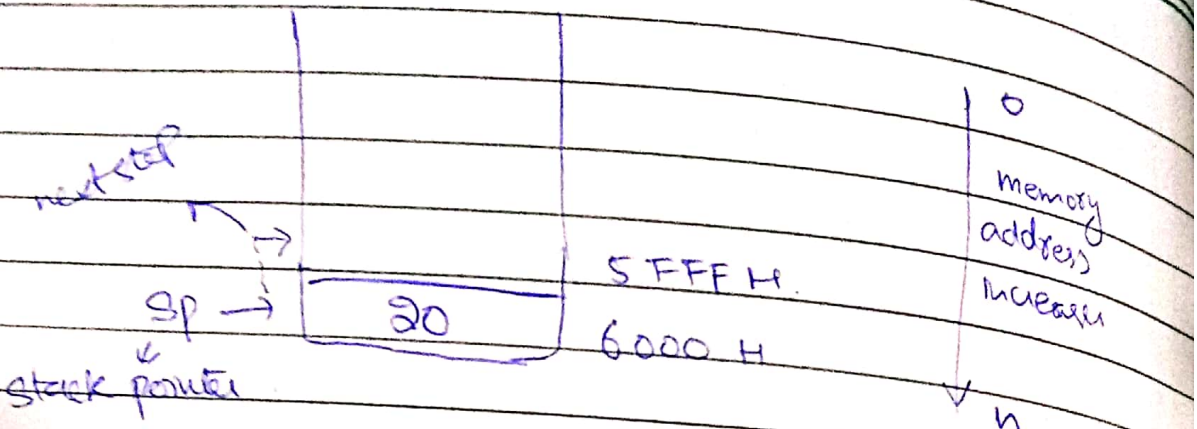
→ Interrupt handler → Interrupt Service Routine.

→ Due to interrupt a jump occurs to ISR or the call location.

→ When interrupt finishes its execution it will return to the saved context and resume normal execution.

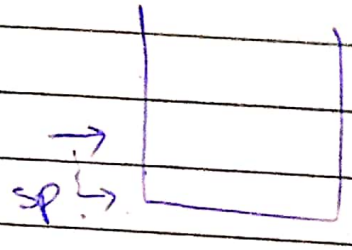
Date: / /

Stack:- Last IN First out (LIFO)



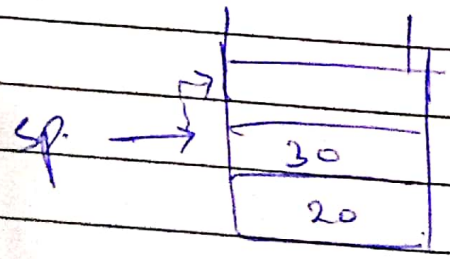
pop:-

SP = SP + 1.



Push:

SP = SP - 1.



Interrupts

- overflow, ÷ by 0
- Timer
- I/O
- hardware failure

→ 4, 5 = Interrupt execution.

Date: / /

1) No interrupt & Wait:
→ Long program waits till the whole interrupt, or call is executed.

1 - 4 - 5 - 2 - 4 - 5 - 3

2) Short Interrupt & Wait:
→ Program causes a jump or interrupt
→ Starts the interrupt execution.
→ Returns back to the program to start executing the program - Doesn't wait for it to finish its processing.
→ Interrupt ends. finishes processing → another interrupt generated and then again returns to the program.
→ Doesn't wait and utilize that time

1 - 4 - 2a - 5 - 2b - 4 - 3a - 5 - 3b.

3) Long Interrupt with:
→ Interrupt doesn't end till another interrupt is not generated.
→ Long wait till an interrupt ends.

1 - 4 - 2 - 5 - 4 - 3

→ Interrupt cycles
Checks if interrupt is present by interrupt signal. Checked by processor

Date: / /

→ PC pointing towards next instruction
⇒ interrupt occurs
→ address on PC stored in Stack.
→ ~~jump~~ PC now pointing towards interrupt handler.
⇒ → return comes.
⇒ retrieve return address from stack and load it in PC
⇒ PC starts executing next instruction of program.

Memory is a set of locations.
Buses connect different ~~buses~~ components together.

Bus → channel → Medium → white wire → broadcast.

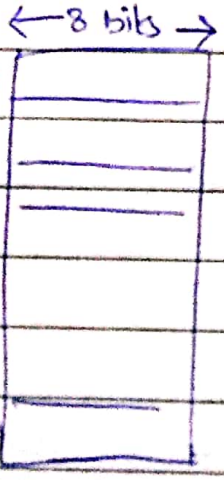
multi bus
↓
not cost effective
↓
easy handling

Uni bus
↓
cost effective.
↓
complex handling

⇒ buses are in the form of 2^n bits.

Memory.

16 x 8 RAM.



4 address bits are required to read address

Bad

Date: / /

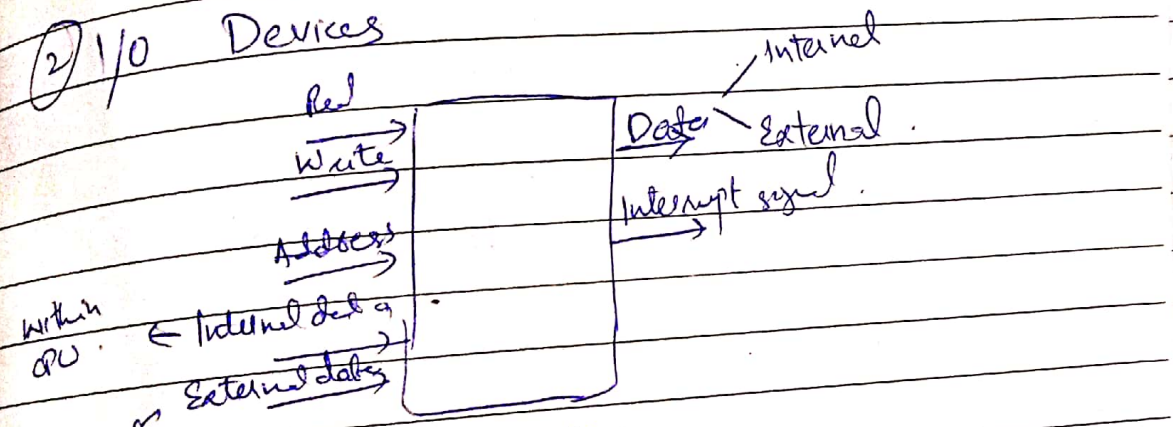
⇒ 8 bit bus for data writing and reading.

⇒ Read & Write are control signals

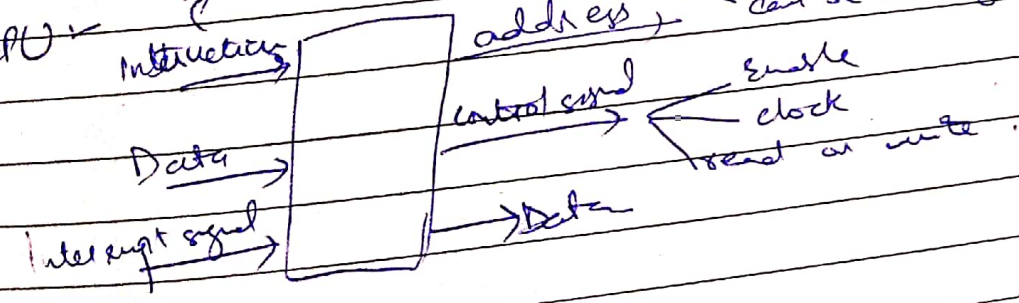
↓ ↓ ↓

address address data.

(2) I/O Devices



(3) CPU ← (can be add, sub etc.)



Synchronization System

↓

works on clock.

↓

on edges ← +ive.

-ive.

Asynchronous System

↓

works on levels.

↓

Date: / /

Status of CPU

Carry flag → carry occurs in ALU → 1 bit extra

Parity flg →

Sign flag → sign of ALU or comparator

Zero flag

Asynchronous needs to be handled carefully but is less expensive (energy).

⇒ Signals send at the same time are destroyed.

⇒ Arbitration is necessary
↓ controls trafic
no collition of data.
↓
divides the data
↓ ↓
central distributed.

representation of number.
↓ ↓
signed unsigned.

Date: / /

One's compliment.

① 3 → 000 000 11

-3 111. 111 00.

Magnitude :- $-2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 = -4$.

② -4

+1

-3 → result

Two's Compliment

① 3 = 0000 0011.

-3 = 1111 1100.

+ 1

1111 1101.

② Magnitude:

⁂ = -3

* Short cut. for 2's compliment.

→ Write the bits the same till the first 1 appear

→ After that flip the numbers.

eg

+2 0010

-2 1 1 1 0

same

flip flip same

Date: / /

Unsigned number.

$0 \rightarrow 2^n - 1$

b/c of 0.

Signed Number.

b/c of signed bit.

$-2^{(n-1)} \rightarrow 0 \rightarrow 2^{n-1} - 1$.

15 = 1111.

6 = 0110.

-6 = 1111 1010

1111	1111
0000	1111
1111	1010
10000	1001

16 = 0001 0000.

24 = 0001 1111.

-24 = 1110 0001

0001	0000.
1110	0001
1111	0001

-128 +64+32+16 +1

0	0000	0000
1	0000	0001
2	0000	0010
3	0000	0011
4	0000	0100
5	0000	0101
6	0000	0110
7	0000	0111
8	0000	1000
9	0000	1001
10	0000	1010
11	0000	1011
12	0000	1100
13	0000	1101
14	0000	1110
15	0000	1111
23	0000	0111
24	0001	1111

Date:

16

-24

0001

1110

1111

1634

-5 =

-9

Scanned by CamScanner

Date: / /

16 = 0001 0000

-24 = 1110 1000

0001 0000 16
0001 0001 17
0001 0010 18
0001 0011 19
0001 0100 20
0101 21
0110 22
0111 23
0001 1000 24

0001 0000
1110 1000
1111 1000

First Method

-5 = 1111 1011
-9 = 1111 0111

-14

1111 1011
1111 0111
1111 0010

Second Method

0000 0101 +5
0000 1001 -9

2's complement

1111 0010 → -14

Babar Register