Unit 6: Processor Organization · (13 m)

To understand the organization of the CPU let us consider the requirements placed on the CPU. The things that it must do,

1. Fetch Instruction

The CPU reads an Inst^{n} from memory

2. Interpret Instruction

The instruction is decoded to determine what action is required

3. Fetch data

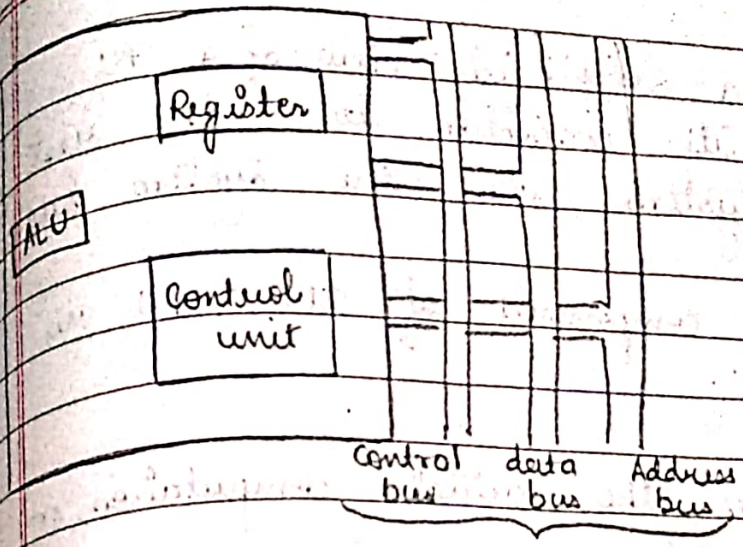
The execution of an Instruction may require reading data from memory or an I/o module.

4. Process data

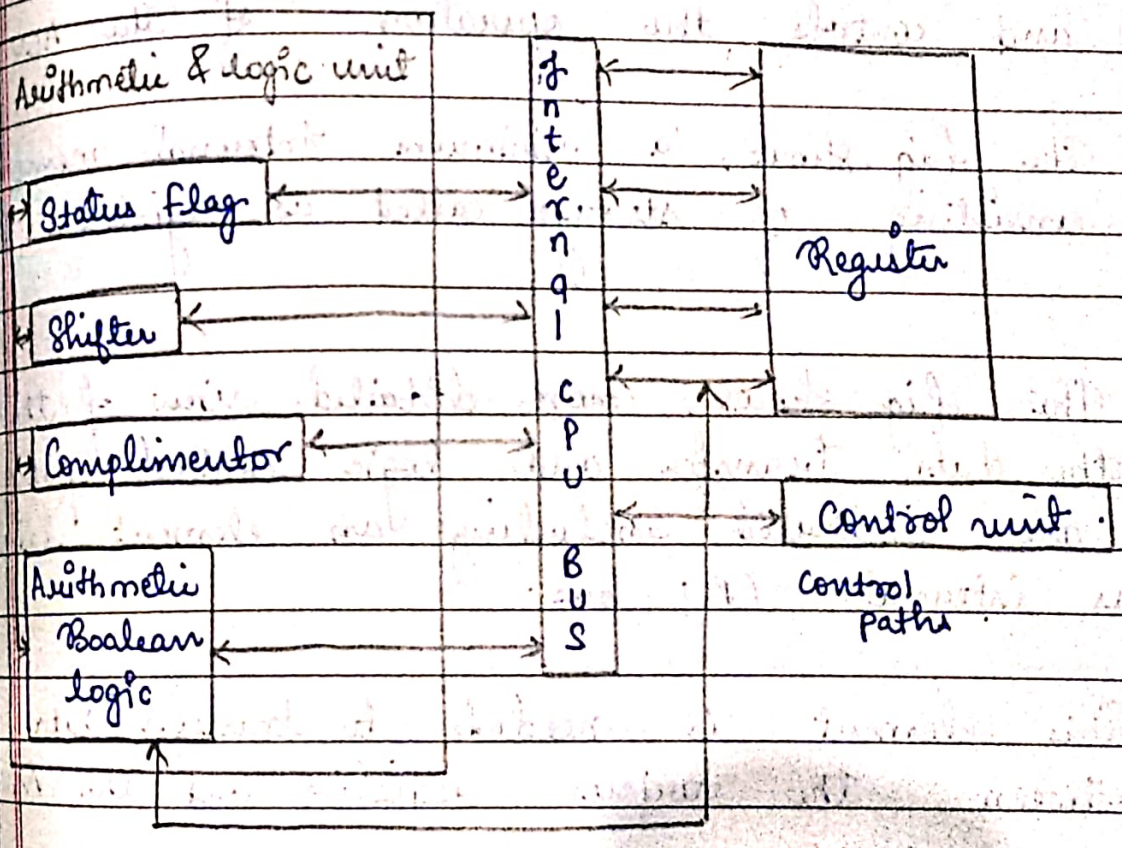
The Execution of an Instruction may require performing some arithmatic or logical operation on data.

5. Write data.

The results of an execution may require writing data to a memory or an I/o module.



CPU with system bus.



Internal Structure of CPU.

Fig. 1. Shows a simplified view of a CPU indicating its connection to the rest of the system via the system bus.

The measure components of the CPU are ALU and CU.

The ALU does the actual computation or processing of data.

The CU controls the movement of data and instructions into and out of the CPU and controls the operation of the ALU.

The fig. shows, a minimum internal memory consisting of storage called as registers.

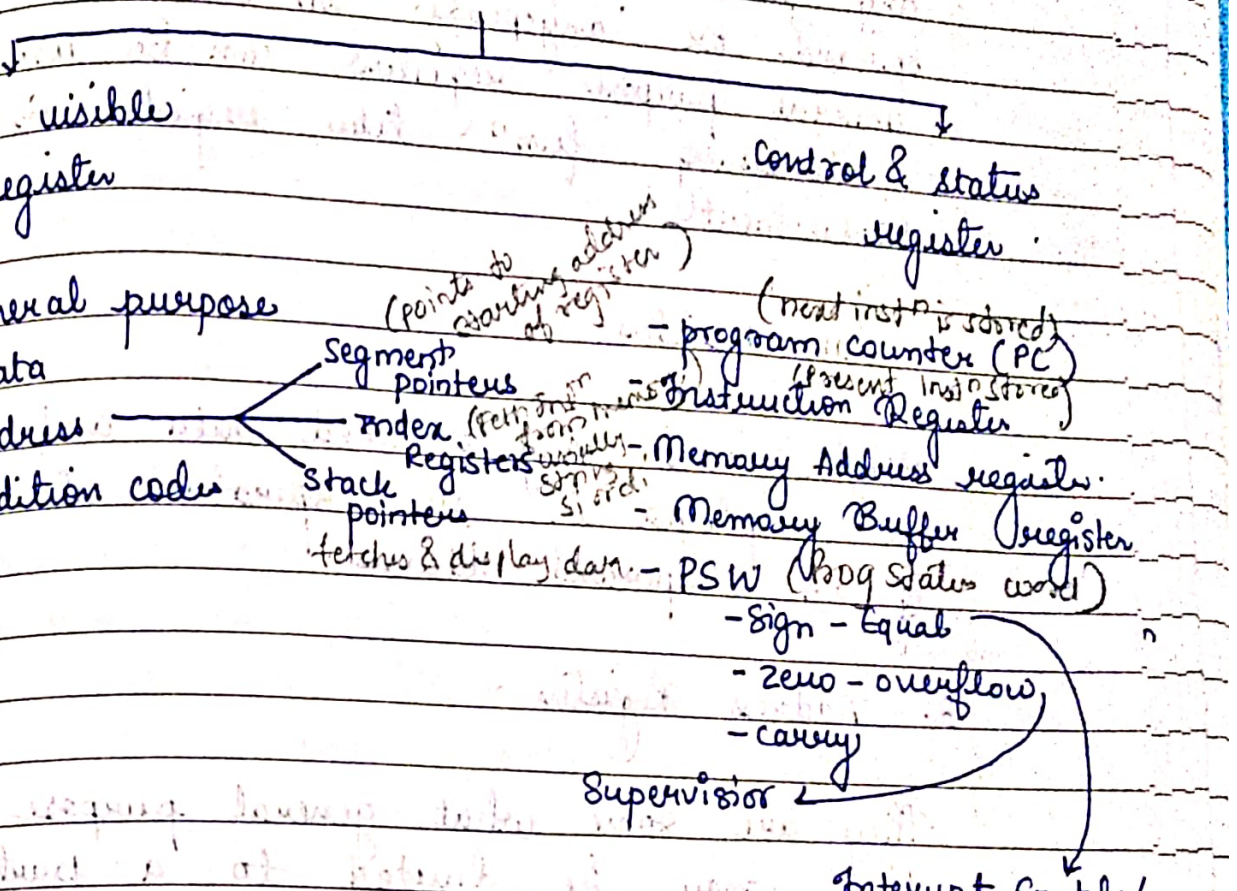
(2) The fig shows, more detailed view of the CPU the data transfer and logic control paths are indicated including an element labeled as internal CPU bus.

This element is needed to transfer data between the various registers and the ALU.

This is a small collection of elements compute CPU, I/O, memory

CPU: Control Unit, ALU and registers connected by data paths.

Register Organization :- (13M)



The registers in the CPU perform two roles.

1. user visible registers.

This enable the machine or assembly language programmer to minimize the main memory references by optimizing the use of registers.

a) General Purpose register

Can be assigned to a variety of functions by the programmer.

Any general purpose register can contain the operand or anyopcode in some cases, general purpose registers can be used for addressing fun^n like register indirect, displacement.

2. data Registers

Maybe used only to hold data and cannot be employed in the operation calculation of an operand address.

3. Address Registers

They are some what general purpose or they may be devoted to a particular addressing mode.

eg include the following:-

1. Segment Pointers

In a machine with segmented addressing a segment register holds the address of the base of the segment

2. Index Register

These are used for Index Addressing mode.

3. Stack Pointer

If their is a user visible stack addressing then typically the stack is in memory then their is a dedicated register that points to the top of the stack

4] Condition Code

also reffered as flags
Condition codes are the bits set by the CPU hardware as the result of the operations for eg. an arithmetic operation may perform a positive, negative, zero or overflow result.

In addition to the result itself being stored in a register or a memory a condition code is also set.

5. Control & Status Registers

Their are a variety of CPU registers that are employed to control the operation on the CPU. most of these, on most machines, are not visible to the user.

Some of theses may be visible to machine instructions executed in a control or OS mode.

Four registers are essential to instruction exec^{n}.

1. Program Counter

Contains the address of the next instruction to be fetched.

2. Instruction Register

Contains the instruction most recently fetched.

3. MAR

Contains the address of location in a memory.

4. MBR

Contains a word of data be written to memory or the word most recently read.

5. The CPU updates the PC after each instruction fetch so that the PC always points to the next instruction to be executed.

6. A Branch or Skip instruction will also modify the content of the PC.

7. The fetched instruction is loaded into an IR where the opcode and operand specifier are analyzed.

Date: __/__/____
Page: _____

8. Data are exchanged with the memory using the MAR and MBR.
MAR connects directly to the address bus & the MBR connects directly to the data bus.

PSW

1. All CPU design include a register or a set of registers often known as the program status word that contains status information

2. PSW contains conditional codes plus status information common fields or flags include:

① Sign
Contains the sign of the result of last arithmatic operation

② Zero
set when the result is zero.

③ Carry
set if an operation resulted into an carry (addition) or borrow (subtraction).
It is used for multi word arithmatic operation.

④ Equal
set if logical compare result is equal equivalent

③ overflow
used to indicate arithmetic overflow

④ Interrupt Enable & Disable

Used to enable & disable the Interrupt.

⑤ Supervisor

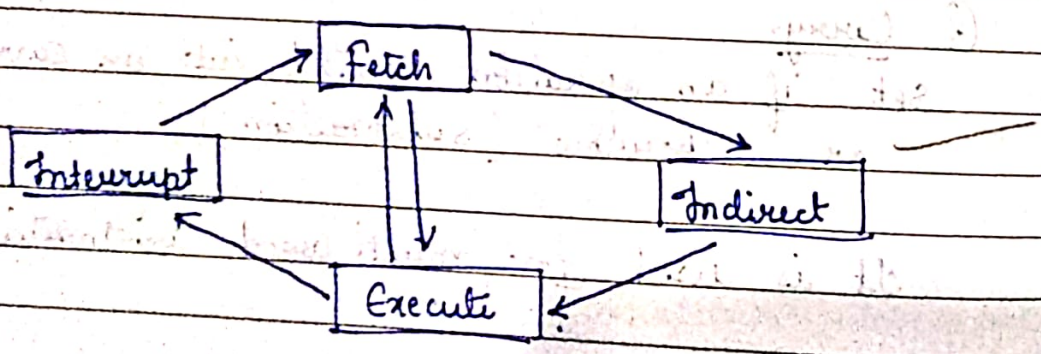
Indicates wheater the CPU is executing in supervisor or user mode.

Instruction Cycle. (BM)

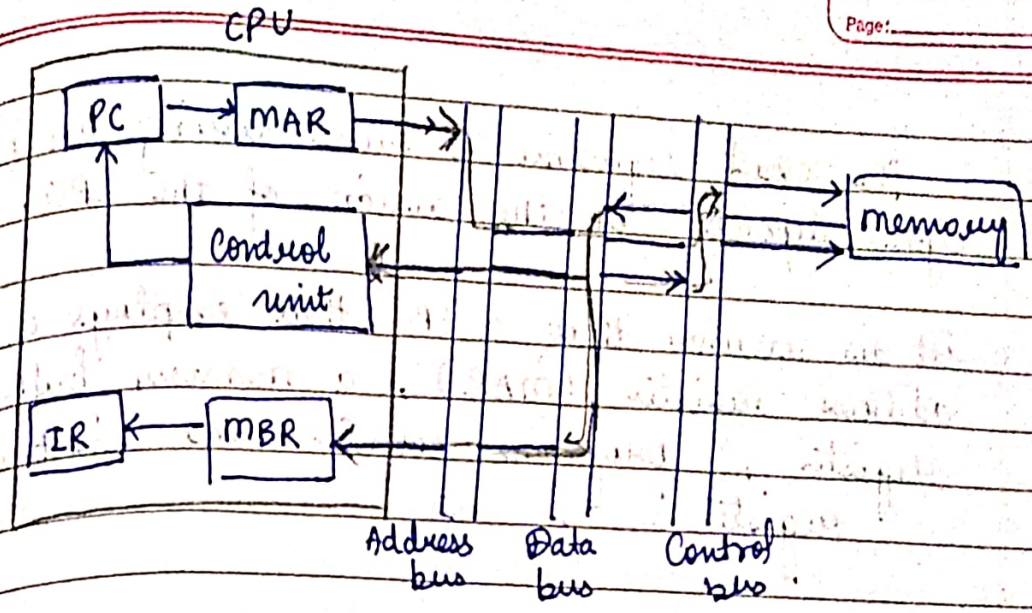
- Fetch
- Excpt
- ~~Interpret~~ Interrupt

mov rax, rbx
mov rax, [2008H]
mem loc
Indirect mode.

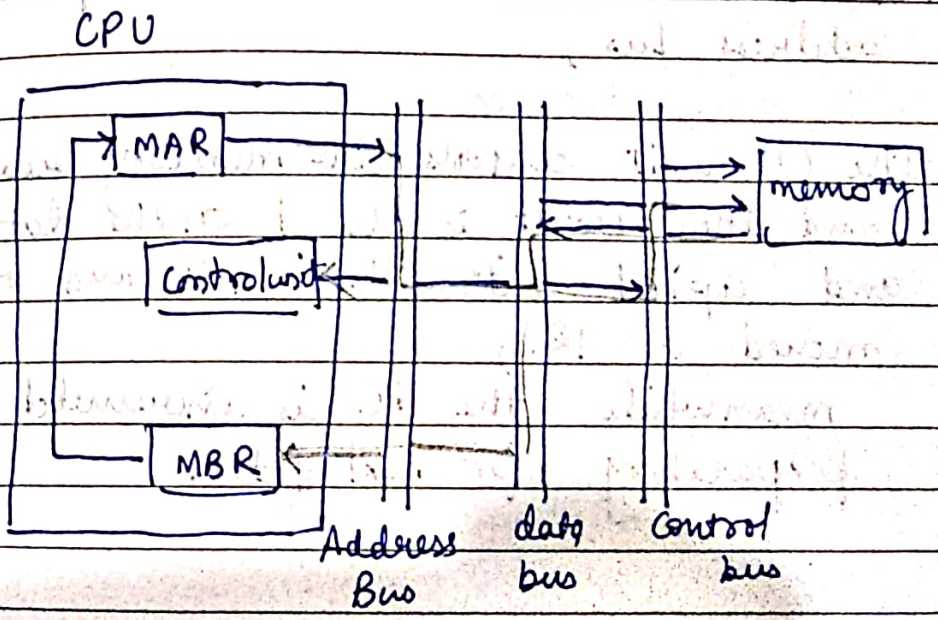
* Indirect Cycle.



The Instruction Cycle.



* MAR - Memory Address Register.
* MBR - Memory Buffer Register.
* PC - Program counter.
* IR - ~~Interrupt~~ Instruction Byte.
* data flow fetch cycle.



* data flow, Indirect Cycle.

The exact sequence of event during an inst cycle depends on the design of the CPU.

Let us assume that a CPU that employs a memory address register (MAR), a memory buffer register, program counter and instruction register.

② During the fetch cycle an instruction is read from memory, fig. shows the flow of data during the cycle.

This address is moved to MAR and placed on the address bus,

The CU unit controls the memory read and the result is placed on the data bus and copied into the MBR and then moved to IR,
meanwhile the PC is incremented by 1 preparatory for next fetch.

The control unit examines the contents of the IR to determine if it contains an operand specifier using indirect addressing

Date: / /
Page:

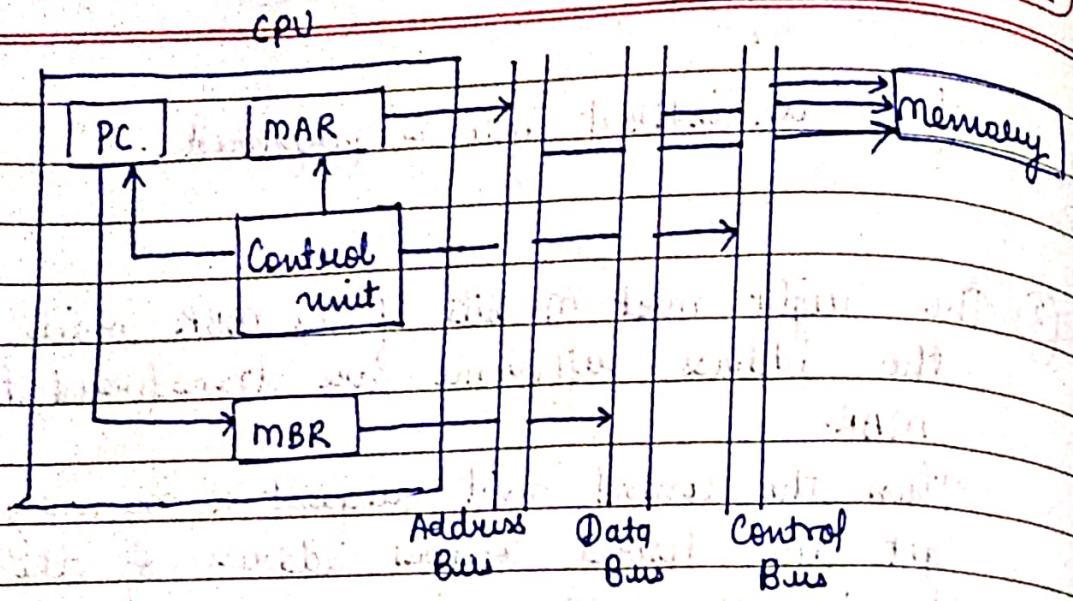
if so an indirect cycle is performed.

(3) The right most n bits of the MBR which contains the address reference are transferred to the MAR.
Then the control unit request a memory read to get the desired ~~output~~ address of the operand into the MBR.

The fetch and indirect cycles are simple and predictable.

The execute cycle takes many forms; the form depends on which of the various machine instructions is in the IR.

This cycle may involve transfering data among registers, read or write from memory or I/O.



* Data flow Interrupt Cycle.

① The current contents of the PC must be saved so that the CPU can receive the normal activities of the interrupt

Thus, the content of PC are transferred to MBR to be written into memory.

The Special memory location reserved for this purpose is loaded into the MAR from the control unit.

It might be a stack pointer,
the PC is loaded with the address of the interrupt routine.

As a result the next instruction cycle will begin by fetching the appropriate Inst^{n}.