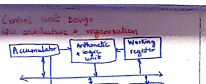
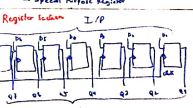
Control Unit Design
CPU architecture & organization



- The Accumulator is divided into different groups
  - → Registers
  - → ALU
  - → Interrupt Control
  - → Timing and Control Circuitary
- It consists of PIPO registers
- Also known as scratch pad memory.
- It stores data and address of memory
- Register organization affects the length of program and complexity of the program
- For better performance, no. of registers should be large
- The architecture of microcomputer depends on the number and type of the registers used in microprocessor
- It consist of 8 bit & 16 bit registers
- The register section varies from microprocessor to microprocessor
- The registers are used to store the data and address
- Registers are classified as
  - → Temporary Registers
  - → General Purpose Registers
  - → Special Purpose Register

Register Indian I/P



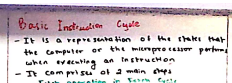
Arithmetic & Logic unit
It processes data to perform Arithmetic and logic operations
- It performs arithmetic operations like addition, subtraction and logical operations like AND, OR, Ex-OR, etc
- ALU is controlled by timing and control unit
- Not available to user
- Accepts operand from memory or register

Interrupt Control
It accepts different interrupt requests signals.
- When a valid interrupt request is present it informs control logic to take action in response to each signal

Timing & control circuitary
It is a control section of microprocessor made up of synchronous sequential logic circuit
It controls all internal and external circuits
It operates with reference to clock signal
The basic operation of a microprocessor is done by this unit
It synchronizes all the data transfer

Basic Instruction Cycle
- It is a representation of the states that the computer or the microprocessor performs when executing an instruction
- It comprises of 2 main steps
  - Fetch operation in Fetch Cycle
  - Execution operation in Execute Cycle

Also Interrupt cycle to check interrupts

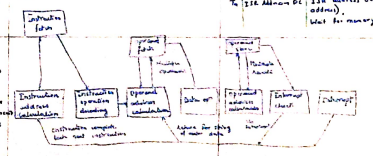


Fetch Cycle operations -
- Program Counter holds address of next instruction to fetch; hence the CPU (processor) fetches instruction from memory location pointed to by PC.
- It is done by providing the value of the PC to MAR and giving read control signal to the memory. On this the memory provides the value in the given address (which is the instruction) to MBR.
- The PC value has to be incremented to point to the next instruction
  (Sometimes the value of PC may have to be completely changed in case of some special instructions called as Branching Instructions)
- The instruction is loaded into IR from MBR
- Finally the processor interprets or decodes the instruction. The processor performs required operation in the execute cycle

Execute Cycle Operation
- Transfer of data between processor and memory or between processor and I/O modules
- Processing of data like some arithmetic or logical operations on data
- Change in the sequence of operation
  ie Branching Instructions

Interrupt Cycle operations
- Suspend the execution of current program
- Save the context of the current program under execution
- Set the PC value to start address of interrupt handler routine also called as interrupt service routine
- Interrupt service routine is a small program which when executed services the interrupting device
- Restore the context and continue execution of the interrupted program

Detailed instruction cycle



Instruction, micro-operation and Micro operations : Interpretation & Sequencing
- Programs are executed in a sequence of instructions.
- Each instruction consists of a series of steps that make up the instruction cycle ie. fetch, decode, etc. Each of these steps is, in turn, made up of a smaller series of steps called micro-operations and micro-instructions
- Control Signals are issued to perform these micro operations or micro instructions
- There are some registers like the register 'y' to provide one of the operand to ALU
- Another register 'Z' which is used to store the result given by ALU
- A temp register to store some temporary data
- There is also instruction decoder for decoding the instruction stored in IR and in turn provides the micro instructions or the control signals for the resources inside and outside of the CPU
- ALU also gets the control signals from decoder indicating op^n to be performed like Add, Sub, AND, etc.
- To execute any instruction it is divided into 4 cycles
  - → Fetch Cycle
  - → Execute Cycle
  - → Interrupt Cycle

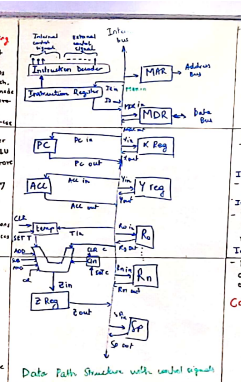
The execute cycle will differ based on the operations to be carried out in the instruction but the fetch and interrupt cycle will be common for all the cycles

Fetch Cycle
- It is concerned to fetch i.e. read from memory the instruction.
- A fetch is a time state and it equal to one clock pulse

Operation	Micro instructions
T_0 PC → MAR	Read, MAR in, Read, Clear y, Set Cin, Add, Zin
T_1 M → MBR, PC → PC+1	Zout, PCin, Wait for memory, Read cycle
T_2 MBR → IR	MBRout, IRin

Interrupt Cycle
- micro instructions for the interrupt cycle

Operation	micro-instructions
T_1 SP ← SP-1	Store (address), Decrement, Zin
T_2 SP → MAR	Zout, MARin, Write
T_3 PC → MBR	Read (return address), MBRin, Write
T_4 IRE Address → PC	IRE address out, PCin (new address), Wait for memory write cycle



Execute Cycle
- micro instructions for the execute cycle of direct addressed mode of operand source.

Operation	micro instruction
T_1 IR → MAR	IRout (address), MAR in, Read, Clear Cin
T_2 M → MBR	MBR out, add, Zin
T_3 MBR → R0	Zout, R0in

Micro instructions of the execute cycle of an indirect addressed operand instruction

Operation	Micro instruction
T_1 IR → MAR	IRout (address), MARin, Read, Clear Cin
T_2 M → MBR	MBRout, Tin, Wait for memory read cycle
T_3 MBR → MAR	MBRout (address) MARin, Read
T_4 M → MBR	Wait for memory read cycle
T_5 MBR → R0	MBRout, Add, Zin
T_6	Zout, R0in

Applications of microprogramming
In realization of control unit
- Microprogramming is used widely now for implementing the control unit of computers

In Operating System
- Microprogramme can be used to implement items of the primitives of OS
- This simplifies operation system implementation and also improves the performance of the OS

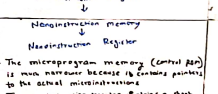
In high level language support
- In high level language users sub functions and and data types can be implemented using microprogramming

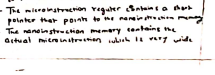
In micro diagnostics
- Microprogramming can be used for detection, isolation monitoring and repair of system errors

In User tailoring
- By using RAM for implementing control memory (CM), it is possible to tailor the machine to different applications

In Emulation
- Emulation refers to the use of a microprogram on one machine to execute programs originally written for another machine

Concepts of Nanoprogramming





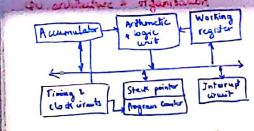
- The microprogram memory (control ROM) is even narrower because it contains pointers to the actual microinstructions
- The microinstruction register contains a short pointer that points to the nanoprogram memory
- The nanoinstruction memory contains the actual microinstructions which is very wide

Wilkes's Microprogrammed Control Unit
- Proposed by Wilkes in 1952
- In this, microinstruction has 2 major components
  - → Control Field
  - → Address Field

$C_0, C_1, C_2, \ldots, C_{n-1}$	$a_{n-1}, a_{n-2}, \ldots, a_1, a_0$

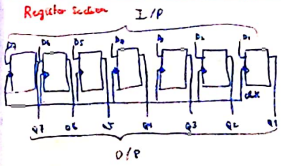
Attribute	Hardwired Control	micro-programmed Control
Speed	Fast	Slow
Cost of implementation	More	Cheaper
Implementation Approach	Sequential circuits	Programming
Flexibility	Not flexible Difficult to modify for new instruction	Flexible new machine instructions can easily handled
Ability to handle complex instruction	difficult	Easier
Design process	Complicated	Systematic
Decoding & Sequencing Logic	Complex	Easy
Applications	Risc MP	CSIC MP
Instruction set size	Small	Large
Control Memory size	Absent	Present
Chip area req. used	Less	More

Control Unit design
the architecture & organization



- The ALU is divided into different groups
  - → Registers
  - → ALU
  - → Interrupt Control
  - → Timing and Control Circuitary
- It consists of PIPO registers
- Also known as scratch pad memory. It stores data and address of memory
- Register organization affects the length of program and simplification of the program
- For better performance, no. of registers should be large
- The archit of microcomputer depends on the number and type of the register used in microprocessor
- It consist of 8 bit & 16 bit registers
- The register section varies from microprocessor to microprocessor
- The registers are used to store the data and address
- Registers are classified as
  - → Temporary Registers
  - → General Purpose Register
  - → Special Purpose Register

Register section



Arithmetic & Logic unit
It processes data i.e. Perform Arithmetic and logic operations

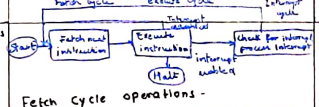
- It performs arithmetic operations like addition, Subtraction and logical operations like ANDing, ORing, Ex-ORing, etc.
- ALU is controlled by timing and control unit
- Not available to User
- Accepts operand from memory or register

Interrupt Control
- It accepts different interrupt request signals.
- When a valid interrupt request is present it informs control logic to take action in response to each signal

Timing & control circuitary
- It is a control section of microprocessor made up of synchronous sequential logic circuit
- It controls all internal and external circuits
- It operates with reference to clock signal
- The basic operation of a microprocessor is done by this unit
- It synchronizes all the data transfer

Basic Instruction Cycle
- It is a representation of the states that the computer or the microprocessor performs when executing an instruction
- It comprises of 2 main steps
  - Fetch operation in Fetch Cycle
  - Execution operation in Execute Cycle

Also Interrupt Cycle to check interrupts

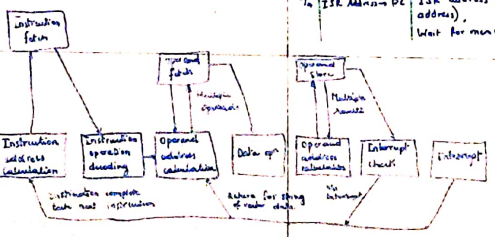


Fetch Cycle operations -
- Program Counter holds address of next instruction to fetch; hence the CPU (processor) fetches instruction from memory location pointed to by PC.
- It is done by providing the value of the PC to MAR and giving read control signal to the memory. On this the memory provides the value in the given address (which is the instruction) to MBR
- The PC value has to be incremented to point to the next instruction (Sometimes the value of PC may have to be completely changed in case of some special instructions called as Branching Instruction)
- The instruction is loaded into IR from MBR
- Finally the processor interprets or decodes the instruction. The processor performs required operation in the execute cycle

Execute Cycle Operation
- Transfer of data between processor and memory or between processor and I/O module
- Processing of data like some arithmetic or logical operations on data
- Change of the sequence of Operation i.e. Branching Instruction

Interrupt Cycle operations
- Suspend the execution of current program
- Save the context of the current program under execution
- Set the PC value to start address of interrupt handler routine, also called as Interrupt service routine
- Interrupt service routine is a small program which when executed, services the interrupting source
- Process the Interrupt Service Routine (ISR) and then
- Restore the context and continue execution of the interrupted program

Detailed instruction cycle



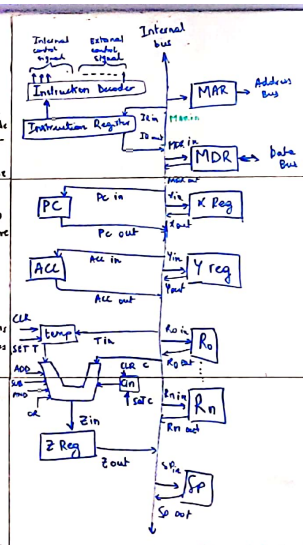
Instruction, micro instruction and Micro operations - Interpretation & Sequencing
- Programs are executed in a sequence of instructions.
- Each instruction consists of a series of steps that make up the instruction cycle i.e. Fetch, decode, etc. Each of these steps is, in turn, made up of a smaller series of steps called micro-operations and micro-instructions
- Control Signals are issued to perform these micro operations or micro instruction
- There are some registers like the register Y in 8085 one of the operand to ALU
- Another register 'Z' which is used to store the result given by ALU
- A temp register to store some temporary data
- There is also instruction decoder for decoding the instruction stored in IR and in turn provides the micro instructions or the control signals for the resources inside and outside of the CPU
- ALU also gets the control signals from decoder indicating op" to be performed like Add, Sub, AND, etc
- To execute any instruction it is divided into 3 cycles
  - → Fetch cycle
  - → Execute Cycle
  - → Interrupt Cycle
- The execute cycle will differ based on the operations to be carried out in the instruction but the fetch and interrupt cycle will be common for all the cycles

Fetch Cycle
- It is concerned to fetch i.e. Read from memory for instruction.
- t-state is a time state and it equal to one clock pulse

	Operation	m-instructions
T_1	PC → MAR	Pout, MARin, Read, Clear, y, Set Cin, Add, Zin
T_2	m → MBR PC ← PC+1	Zout, PCin, Wait for memory fetch cycle
T_3	MBR → IR	MBRout, IRin

Interrupt Cycle
- Microinstruction for the interrupt cycle

	Operation	micro-instructions
T_1 T_2	SP ← SP - 1 SP → MAR	SPout(address), Decrement, Zin Zout, MARin, SPin
T_3	PC → MBR	PCout (return address), MBRin, write
T_4	ISR Address → PC	ISR address out, PCin (new address), Wait for memory write cycle



Execute Cycle
- Micro instructions for the execute cycle of direct addressed mode of operand instruction.

	Operation	micro instruction
T_1	IR → MAR	IRout (address), MAR in, Read, Clear Cin
T_2	m → MBR	Rout, Yin, Wait for memory, Read cycle
T_3 T_4	MBR + R_i → R_i	MBRout, add, Zin Zout, Rin

- Micro instructions of the execute cycle of an indirect addressed operand instruction

	Operation	micro instruction
T_1	IR → MAR	IRout (address), MAR in, Read, Clear Cin
T_2	m → MBR	Rout, Yin, Wait for memory read cycle
T_3	MBR → MAR	MBRout (address) MAR in, Read
T_4	m → MBR	Wait for memory read cycle
T_5 T_6	MBR + R_i → R_i	MBRout, Add, Zin Zout, Rin

Levels of microprogramming

In realization of control unit
- Microprogramming is used widely now for implementing the control unit of computers

In Operating System
- Microprograms can be used to implement some of the primitives of OS
- This simplifies operation system implementation and also improves the performance of the OS

In high level language support
- In high level language varies sub functions and and data types can be implemented using microprogramming

In micro diagnostics
- Microprogramming can be used for detection isolation monitoring and repair of system errors

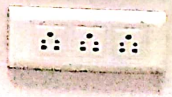
In user tailoring
- By using RAM for implementing control memory (CM), it is possible to tailor the machine to different applications

In Emulation
- Emulation refers to the use of a microprogram on one machine to execute programs originally written for another machine

Concepts of Nanoprogramming

Microprogram Counter
↓
Microprogram Memory
↓
Microinstruction Register
↓
Nanoinstruction Memory
↓
Nanoinstruction Register

- The microprogram memory (control ROM) is much narrower because it contains pointers to the actual microinstructions
- The microinstruction register contains a short pointer that points to the nanoinstruction memory
- The nanoinstruction memory contains the actual microinstruction which is very wide



Wilkie's Microprogrammed Control Unit

- Proposed by Wilkie's in 1952
- In this, microinstruction has 2 major components
  - → Control field
  - → Address field

$C_0, C_1, C_2, \ldots, C_{n-1}$	$a_{n-1}, a_{n-2}, \ldots, a_1, a_0$

Attribute	Hardwired Control	micro-programmed Control
Speed	Fast	Slow
Cost of implementation	More	Cheaper
Implementation Approach	Sequential circuits	Programming
Flexibility	-Not flexible -Difficult to modify for new instruction	-Flexible -New machine instructions can easily handled
Ability to handle complex instruction	Difficult	Easier
Design process	Complicated	Systematic
Decoding & Sequencing Logic	Complex	Easy
Applications	Risc µp	CSIC µp
Instruction set size	Small	Large
Control Memory	Absent	Present
Chip area required	Less	More