MICROPROCESSOR SYSTEMS

Simple Microprocessor

Zuhaib A. Shaikh,
Asst. Prof., CSE Deptt.,QUEST
Web: zuhaib-shaikh.neocities.org

Introduction

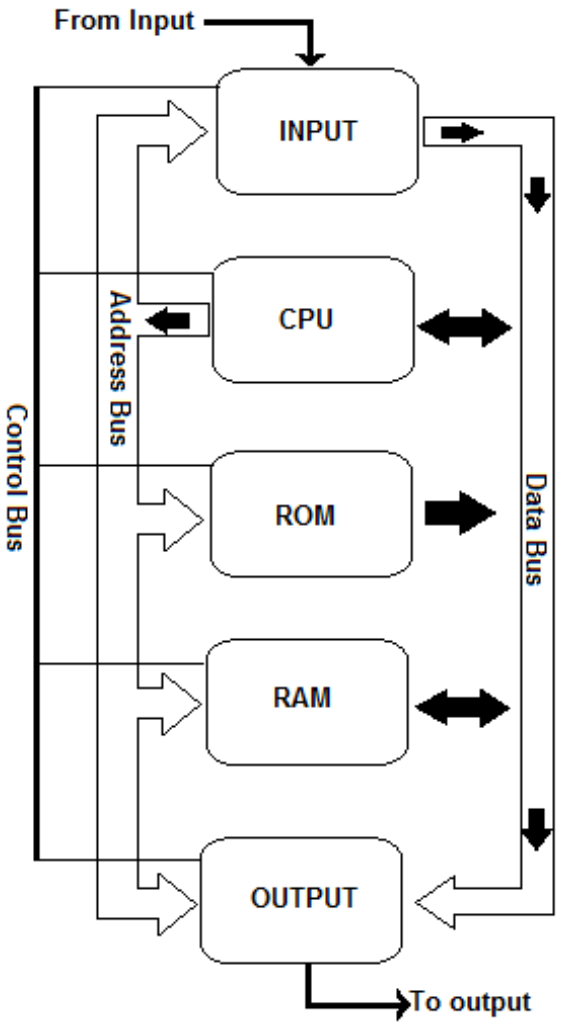
- After the invention of microprocessor, the size shape and price of digital computer have changed a lot
- The microprocessor is an IC (Integrated Circuit) that contains much of the processing capabilities of a computer
- It's a Large Scale Integration chip that is programmable
- A computer with a microprocessor as a CPU (Central Processing Unit) is called a microcomputer
- Microcomputer uses Stored Program concept
- It makes them reprogrammable

A simple Microcomputer

- A digital microcomputer contains five functional components
- Input, Output, CPU, ROM and RAM (Hardware parts)
- ROM mostly stores program or list of instructions
- These programs manipulate data present in RAM
- Programs stores in ROM are called *firmware*
- The microprocessor controls all units with control lines
- Besides control lines:
  - Address lines (16) selects a memory or port location
  - Data lines (8) are two-way paths to transfer data to and from memory / port
- In this simplified organization details have been omitted

A simple Microcomputer

- Two types of semiconductor memories (ROM and RAM)
- Input and output devices
- Bidirectional Data lines and Address lines



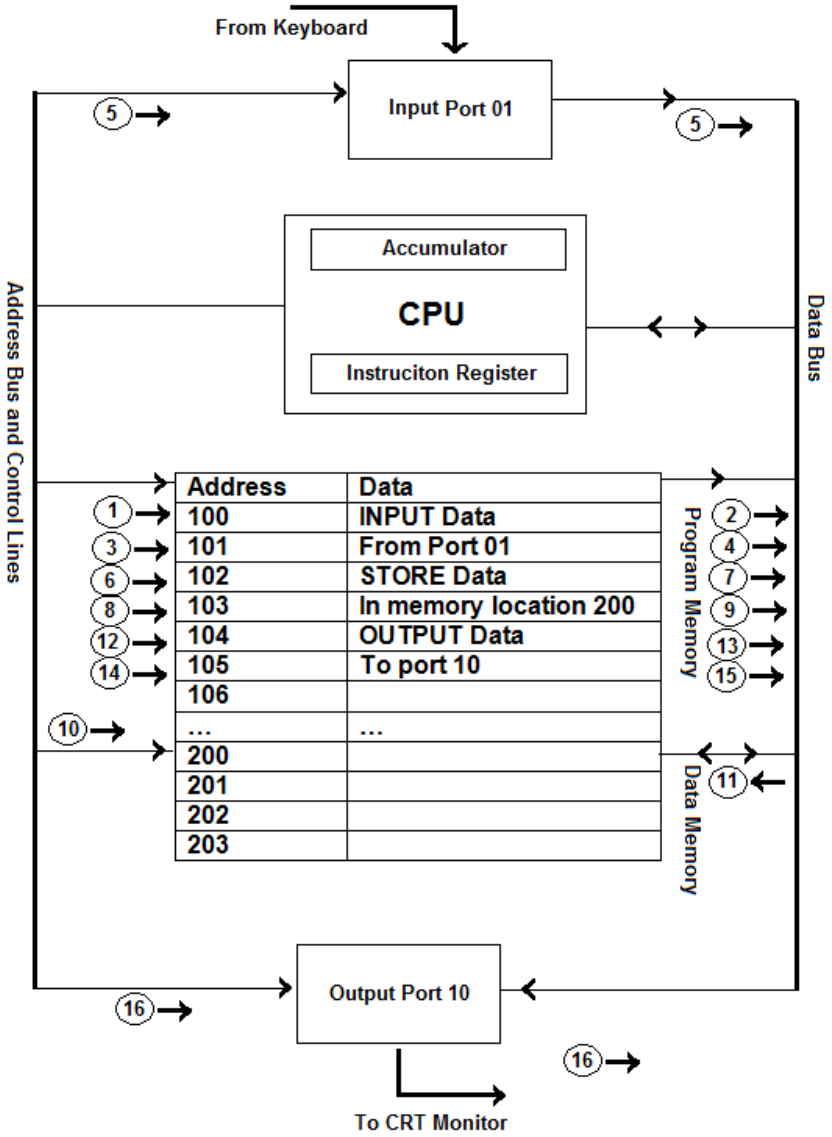
Simplified CPU Organization and Instruction Execution

- Due to the wide variety of microprocessor vendors and microcomputer assembling companies, a generic microcomputer is suitable to study
- While learning about a new microprocessor a programmer must know about its:
  - Architecture
  - Instruction set
  - Minimal System
  - Control Signals
  - Pin functions

Microcomputer Operations

- Consider the example of a simple microcomputer operation
  - Press a key on the keypad
  - Store the letter in memory
  - Display the character
- The instructions are stored in memory using instruction set
- Most of the instructions are stored in more than one memory locations
  - Like the first instruction will be composed of operation (input) and the port address where keypad is connected
  - Second instruction will be composed of operation and the memory address
  - Third instruction will be composed of operation and the display port number

Microcomputer Operations Diagram



Microcomputer Operations (Contd.)

- Step 1: CPU sends the address 100 on the address bus followed by enabling the *read* control line to memory
- Step 2: The memory puts the data byte on that address on the data bus. CPU reads it and put in a register (instruction register). CPU then decodes the instruction
- Step 3: CPU sends out the next address 101 to read the data for the first instruction followed by enabling the *read* control line to memory
- Step 4: The memory puts on the data bus the port number (01) from where to read the data. CPU can now interpret the whole instruction

Microcomputer Operations (Contd.)

- Step 5: The CPU issues the address of port (01) on the address bus followed by read control line. The port then transfers the data on the data bus
- Step 6: The MPU addresses the location 102 on the address bus followed by read control signal
- Step 7: The operation code for STORE instruction is output by the memory and is brought inside the MPU. MPU decodes it
- Step 8: MPU sends next address 103 on the address bus to look for the operand followed by enabling read control line
- Step 9: Memory transfers the byte on the data bus. MPU can now decode the whole instruction

Microcomputer Operations (Contd.)

- Step 10: (Execution) CPU now stores the data. It sends the address 200 on address bus and enable the write input to the memory
- Step 11: The MPU puts the data on the data bus.
- Step 12 to 16 are related to 3rd instruction

Simplified Memory Architecture

- Random access and Sequential access
- The generic microprocessor has 16 address lines
- 2^16 different combinations of 1's and 0's
- It is convenient to write them in hexadecimal
- So the first address is 0000h and last being FFFFh
- Generic microcomputer memory map
- First 256 bytes are ROM (1 page)
- In this organization memory is 256 X 4 bits
- There may be other organizations like 256 X 8bits, 256 X 1 bit
- The maximum memory supported by a microcomputer having 16 address lines is 64Kb

Address	Contents							
0000h	1	1	0	0	0	0	1	1
0001h	0	0	0	0	0	0	0	1
...								
00FEh	0	1	0	0	0	0	0	0
00FFh	1	0	0	0	1	1	1	0
0100h	1	0	0	0	0	0	0	1
...								
1FFFh	1	1	0	0	0	0	0	0
2000h	0	0	0	0	1	1	1	0
2001h	1	0	0	0	0	0	0	1
...								
20FEh	0	1	0	0	0	0	1	0
20FFh	1	0	0	0	0	0	0	1
2100h	0	1	0	0	0	0	1	1
...								
FFFFh	0	0	0	1	1	1	1	1

Instruction Set

- The group of instructions a microprocessor can execute
- Generic microprocessor instruction set
- Instruction sets are not standardized
- Categories of instructions
  - Arithmetic Instructions (Addition, Subtraction, Shift, Increment, Decrement)
  - Logical Instructions (AND, OR, NOT, EX-OR etc.)
  - Data Transfer instructions (Memory to CPU, CPU to memory)
  - Branch Instructions (JUMP)
  - Others

Arithmetic Instructions

- Following arithmetic operations
  - Addition between registers, registers and memory
  - Subtraction between registers, registers and memory
  - Increment register or memory contents
  - Decrement register or memory contents
  - Compare contents of two registers or memory – register
  - Negate the contents of register / memory contents

Logical Instruction

- Includes:
  - Logical AND between registers or memory – register
  - Logical OR between registers or memory – register
  - Logical EX-OR between registers or memory – register
  - Logical NOT of register or memory word
  - Shift Left contents of register / memory
  - Shift Right contents of register / memory

Data Transfer Instructions

- Includes:
  - Load a register with a value from memory (LOAD)
  - Store the register value to memory location (STORE)
  - Move data between registers (MOV)
  - Input from a port (INPUT)
  - Output to a port (OUTPUT)

Branch Instructions

- Includes:
  - Unconditional Branch
  - Conditional Branch
    - Zero
    - Not Zero
    - Equal
    - Not Equal
    - Positive
    - Negative

Other Instructions

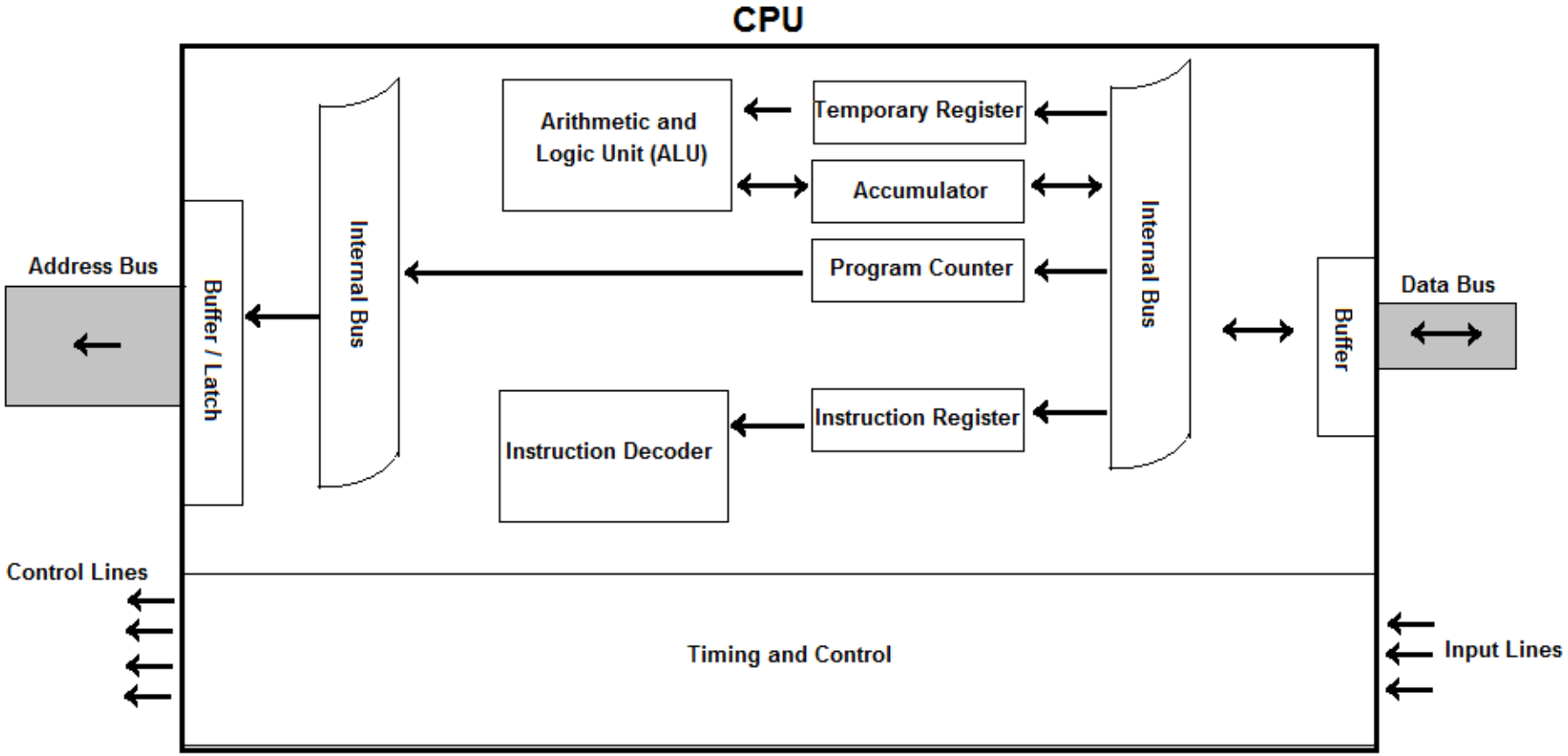
- Includes:
  - Push and Pop
  - NOP
  - HLT

Simplified CPU Architecture

- Microprocessor is the CPU
- CPU contains
  - Registers, computational circuitry, control circuitry
- Primary functions of CPU include:
  - Fetch, Decode and Execute instructions in proper order
  - Transfer data to and from memory and input/output
  - Respond to the interrupts
  - Provide Timing and control signals to other devices
- Various Registers
- The ALU
- Instruction Decoder
- Control and Timing Section
- Fetch Decode and Execute cycle

Simplified CPU Architecture

- Architecture of generic microprocessor



The Generic Microprocessor

- **Common characteristics of microprocessors**
  - **Power connections**
  - **Bit size**
  - **Data lines**
  - **Address lines**
  - **Control lines**
  - **Internal registers**
  - **Addressing modes**

Data Sheet Description

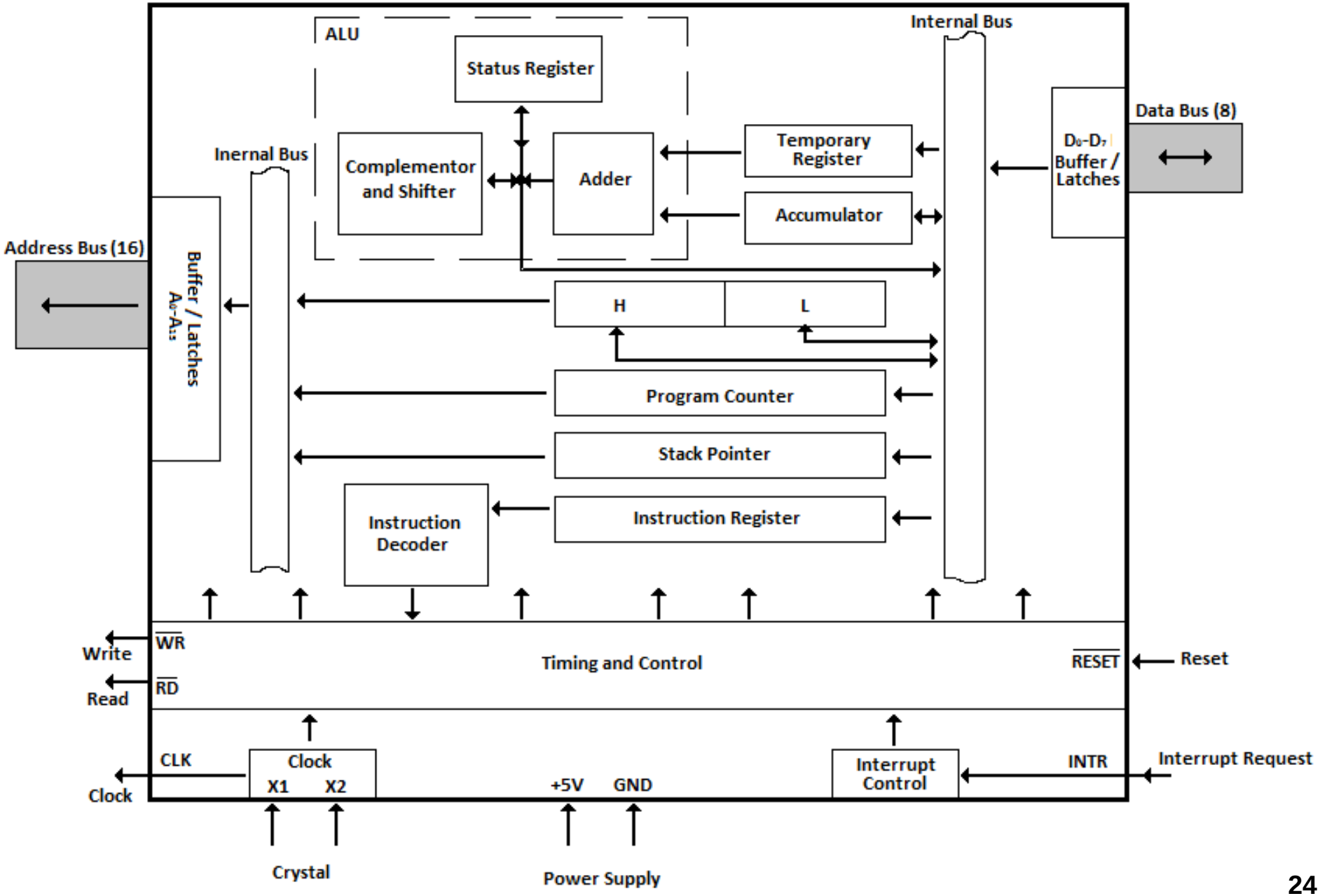
- **Contains information like:**
  - IC packaging
  - Pin Diagram
  - Structure of microprocessor
  - Timing Diagram
  - Schematics of minimal system
  - Instruction set

Pin Diagram and Functions

- 40-pin DIP
- Regulated Power supply (1,2)
- Clock (38-40)
- 16-bit address bus (5-20)
- 8 bi-directional data lines (21-28)
- RD*, WR* control outputs (30,31)
- RESET (33)
- INTR (35)
- Steps in processing interrupts

CPU Architecture

- Arithmetic and Logic Unit
- Several Registers
- Program Counter
- Instruction Decoder
- Bus Buffers and latches
- Control Lines
- T-States and Machine Cycles
- Microcontrollers


ALU
Status Register
Internal Bus
Data Bus (8)
Inernal Bus
Complementor and Shifter
Adder
Temporary Register
Accumulator
D0-D7 Buffer / Latches
Address Bus (16)
Buffer / Latches A0-A15
H
L
Program Counter
Stack Pointer
Instruction Decoder
Instruction Register
Write
WR
Read
RD
Timing and Control
RESET
Reset
CLK
Clock
X1
X2
Clock
+5V
GND
Interrupt Control
INTR
Interrupt Request
Crystal
Power Supply

Using Data / Address Registers

- HL Register Pair
- It is a pointer to memory
- Simple program for adding three consecutive numbers in memory and storing in next location

Using the Stack Pointer

- 16-bit special purpose counter-register that contains an address
- The address refer to the area of memory called stack
- Stack is used when there is a procedure call or interrupt
- Stack is also used to store temporarily the registers
- SP initialized at one location higher than the last word of stack
- Stack is read/write memory
- Data is written using PUSH or CALL instruction
- Data is read using POP or RETURN instruction
- It is a Last-In-First-Out Data Structure

Using the Stack Pointer (Contd.)

- When reading the data from the stack, the data comes in reverse order
- When to POP data, first data is read and then SP is incremented
- PUSH and POP are always used in pair
- Every microprocessor has its own operand type to PUSH and POP

Programming Generic Microprocessor

- A machine program is what a microprocessor can understand directly (composed of 1's and 0's)
- Program are difficult for human beings to understand
- The machine language program can be made easier to understand by replacing binary codes with Hexadecimal equivalents
- Still difficult to understand / program
- The program:

  (1) Load the binary number (1011 0100) into accumulator

  (2) Take 1's complement of the number

  (3) Store the complemented number in memory location 2100h

Programming Generic Microprocessor (Contd.)

- The last representation was easy and understandable by the humans
- One step up from the machine language is the "*assembly language*"
- Generally, the assembly language combines 1 or more bytes of machines code into a phrase or instruction
- Assembly language represents binary op-codes with *mnemonics*
- Like the mnemonic for the second instruction in former program is CMA (Complement Acc.) while its op-code is 0010 1111
- A program called *assembler* can translate mnemonics into binary codes (machine code)

Programming Generic Microprocessor (Contd.)

- A program listing containing assembly language instructions, their machine codes and descriptions
- Such a listing contains several fields like address, machine code, label, mnemonic and operands
- The assembler only requires mnemonic and operands to translate
- The listing containing mnemonics and operands is called *source code* and is passed to assembler
- Assembler translates and produces an object code containing machine codes to be stored in consecutive memory locations

Programming Generic Microprocessor (Contd.)

Address (hex)	Contents (hex)	Label	Op code	Operand	Comments
2000 2001 2002	21 20 20		LXI	H,2020H	Initialize HL register pair at 2020H
2003	06		MVI	B, 01H	Initialize register B with 01H
2004	01				
2005	7E		MOV	A, M	Move binary number from memory Location 2020H to accumulator
2006	2F		CMA		Complement accumulator
2007	80		ADD		Add B to A to form 2s complement
2008	77		MOV	M, A	Move 2s complement number from accumulator to memory location 2020H
2009	76		HLT		Stop MPU

Questions