1 8085 Microprocessor Architecture

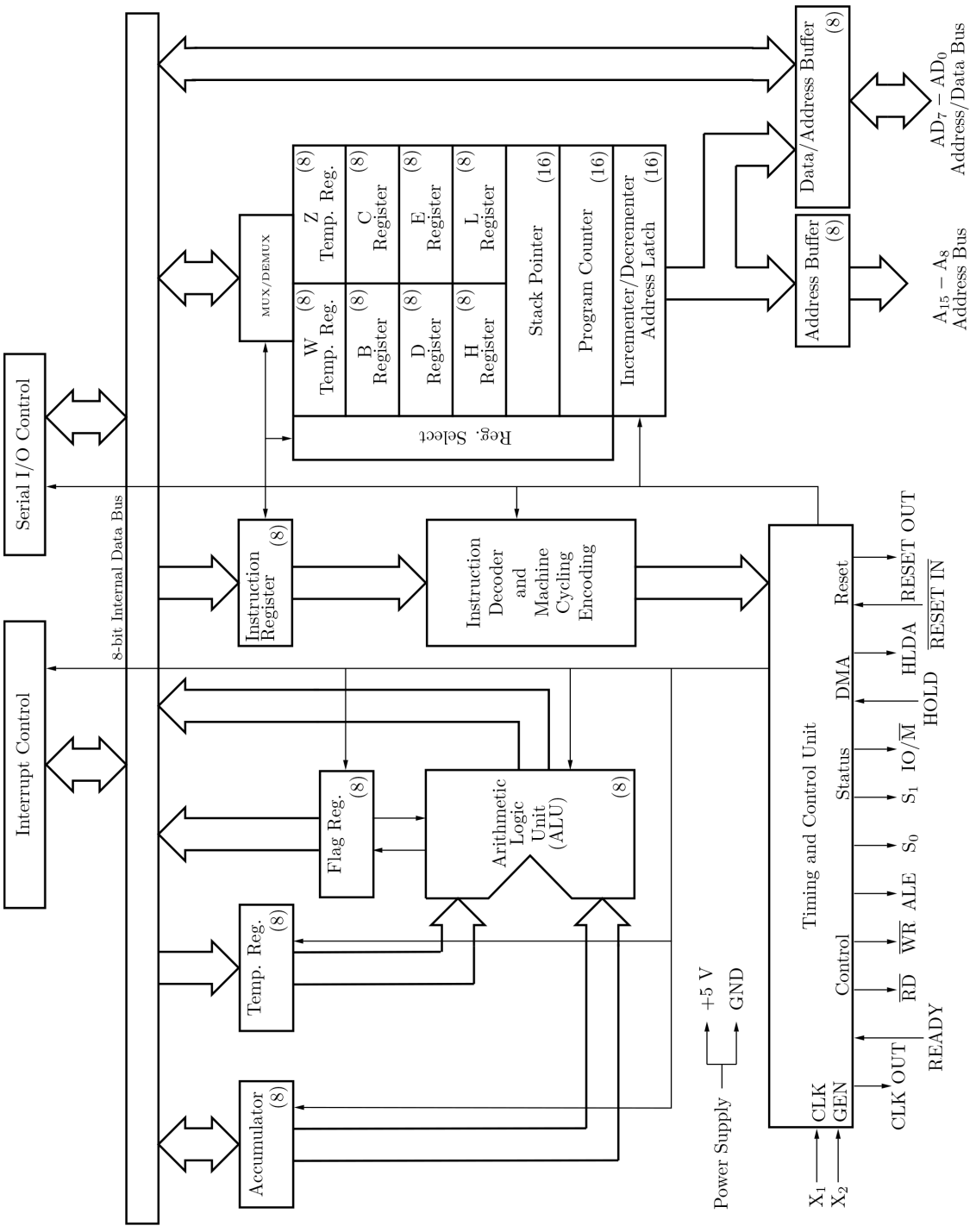


Figure 1: 8085 Architecture

The architecture of a microprocessor chip is a description of the physical layout of the various elements that form it. Two common architecture model used in computer organisation is **von Neumann architecture** and **Harvard architecture**.

The 8085 microprocessor uses the von Neumann architecture. The characteristics of a von Neumann architecture are as follows:

- Both *data* and *instructions* are stored in *R/W memory*.
- The contents of memory (data and instructions) are accessed by *location*.
- Instructions are accessed and executed *sequentially*.

Since we are storing both data and program(instructions) in the memory, this architecture is also called Stored-Program architecture.

The various units of the 8085 microprocessor are discussed next.

2 Registers

The data and instructions which are stored in memory are fetched to the microprocessor and are stored temporarily in Registers for processing.

A register is nothing but a set of flip-flops. The number of flip-flops used depend on the size of the data that the microprocessor needs to process.

Registers can be broadly classified into two categories, namely

- General Purpose − These registers have not been explicitly defined for any purpose or to store any particular type of information.
- Special Purpose − These registers are used for some specific function.

2.1 General Purpose Registers

In 8085 microprocessor, we have six 8-bit general purpose registers identified as B, C, D, E, H and L. They can be combined as register pairs−BC, DE and HL−to perform some 16-bit operations. The HL register pair also works as a memory pointer (M), i.e., the 16-bit data is the address of a particular memory location.

2.2 Accumulator

The accumulator is an 8-bit register that is used to store one of the operands needed to perform arithmetic and logical operations. Also, the result of any operation is always stored in the accumulator.

2.3 Flag Register

Flag register is a set of flip-flops which gives the status of the result in the accumulator. After any arithmetic or logic operation, depending on the result, the flag register is changed. The bit positions in the flag register are shown below:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S	Z		AC		P		CY

Figure 2: Flag Register

The description and conditions of the flags are as follows:

S – Sign flag: After the execution of an arithmetic or logic operation, if bit D_7 of the accumulator (result) is 1, then the Sign flag is set (S=1).

Z – Zero flag: After the execution of an arithmetic or logic operation, if the result is zero (Acc = `00`H), then the Zero flag is set (Z=1).

AC – Auxiliary Carry flag: In an arithmetic operation, when a carry is generated from lower nibble to upper nibble, i.e., carry from D_3 bit to D_4 bit, then the Auxiliary Carry flag is set (AC=1). This flag is used only internally for BCD operations and is not available for the programmer to change the sequence of a program with a jump operation.

P – Parity flag: After the execution of an arithmetic or logic operation, if the result has an even number of 1s, then the Parity flag is set (P=1). For example, if the accumulator (after certain operation) has the data byte `0000 0011`, then the parity bit is set to 1.

CY – Carry flag: If an arithmetic operation results in a carry or borrow, then the Carry bit is set (CY=1).

All the flags are independent of each other. So depending on the result of the accumulator, some flags may be set(=1) and others may be reset(=0).
The flag register is also called Program Status Word (PSW).

2.4 Program Counter

It is a 16-bit register used to hold address of the memory location from which the next instruction is to be fetched. So we can say that this register is used to sequence the execution of the instructions.

2.5 Stack Pointer

It is a 16-bit register which points to a memory location where stack is implemented. The beginning of the stack is defined by loading a 16-bit address in the stack pointer.

2.6 Incrementer/Decrementer - Address Latch

In the schematic, these are shown in one block, but actually they are two separate circuits.

The Incrementer/Decrementer block increments the program counter when instructions are executed, increments and decrements the stack pointer when needed, and supports 16-bit increment and decrement instructions. For example, once the first instruction is fetched from a memory location, the program counter is incremented accordingly to point to the memory location of the next instruction.

The Address Latch holds the 16-bit value that is written to the address bus. This value is also the input to the 16-bit increment/decrement circuit.

2.7 Temporary Registers

It is an 8-bit register which is used to store one of the operand (other is stored in accumulator) for any arithmetic or logic operation.

There are two more 8-bit temporary registers identified as W and Z. These registers are used by microprocessor internally to hold any intermediate data, i.e., data which is needed during the execution of an instruction.

The temporary registers are not accessible to the programmer.

2.8 MUX/DEMUX

MUX/DEMUX unit is used to select a register out of all the available registers. This unit behaves as a MUX when data is going from the register to the internal data bus. It behaves as a DEMUX when data is coming to a register from the internal data bus of the microprocessor.

2.9 Register Select

The Register Select behaves as the select lines of MUX/DEMUX unit

2.10 Address/Data Buffer and Address Buffer

The contents of the stack pointer and program counter are loaded into the address buffer and address-data buffer. These buffers are then used to drive the external address bus and address-data bus. Any exchange of data between memory or I/O devices is done through these buffers.

2.11 Instruction Register

When an instruction (opcode) is fetched from the memory, it is loaded in the instruction register for further processing.

2.12 Instruction Decoder and Machine Cycle Encoding

The instruction decoder decodes (understands) the instruction and establishes the sequence of events to follow, i.e., how many machine cycles are required to execute the instruction.

3 Arithmetic Logic Unit (ALU)

The Arithmetic Logic Unit performs the computing functions of the microprocessor. It takes two 8-bit inputs (from accumulator and temporary register) and performs one of the five basic operations: ADD, OR, XOR, AND and SHIFT-RIGHT.

If one of the operands (temporary register) is inverted, the ALU can perform SUB (subtraction) and CMA (complement) operations. SHIFT-LEFT operation is performed simply by adding the number to itself.

The detailed schematic of the 8085 ALU can be found at:
`http://www.righto.com/2013/01/inside-alu-of-8085-microprocessor.html`

4 Timing and Control Unit

This unit synchronizes all the microprocessor operations with the clock and generates the control signal necessary for communication between the microprocessor and peripherals.

This unit is responsible for proper operation of the microprocessor. After an instruction is decoded, the timing and control unit will generate various signals based on the instruction to make the microprocessor perform accordingly.

5 Other units

5.1 Interrupt Control

This unit is used to handle interrupt requests. It provides the mechanism for external hardware to interrupt the microprocessor and also have the microprocessor acknowledge the interrupts through various signals/pins.

5.2 Serial I/O Control

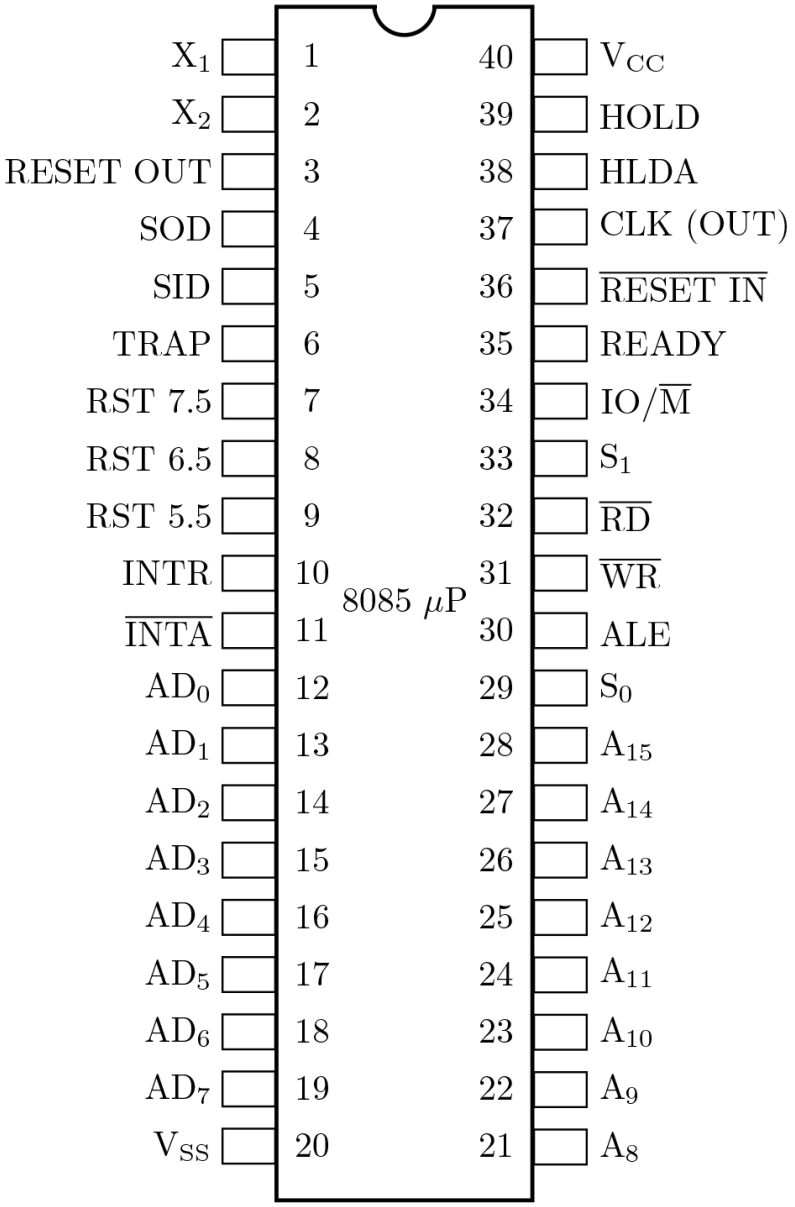
This unit is used to perform serial data communication. In serial transmission, data bits are sent over a single line, one bit at a time.

6 8085 Microprocessor Pins and Signals

The 8085 is an 8-bit general-purpose microprocessor capable of addressing 64K of memory. The device has 40 pins, requires a +5V single power supply and can operate with a 3-MHz single-phase clock.

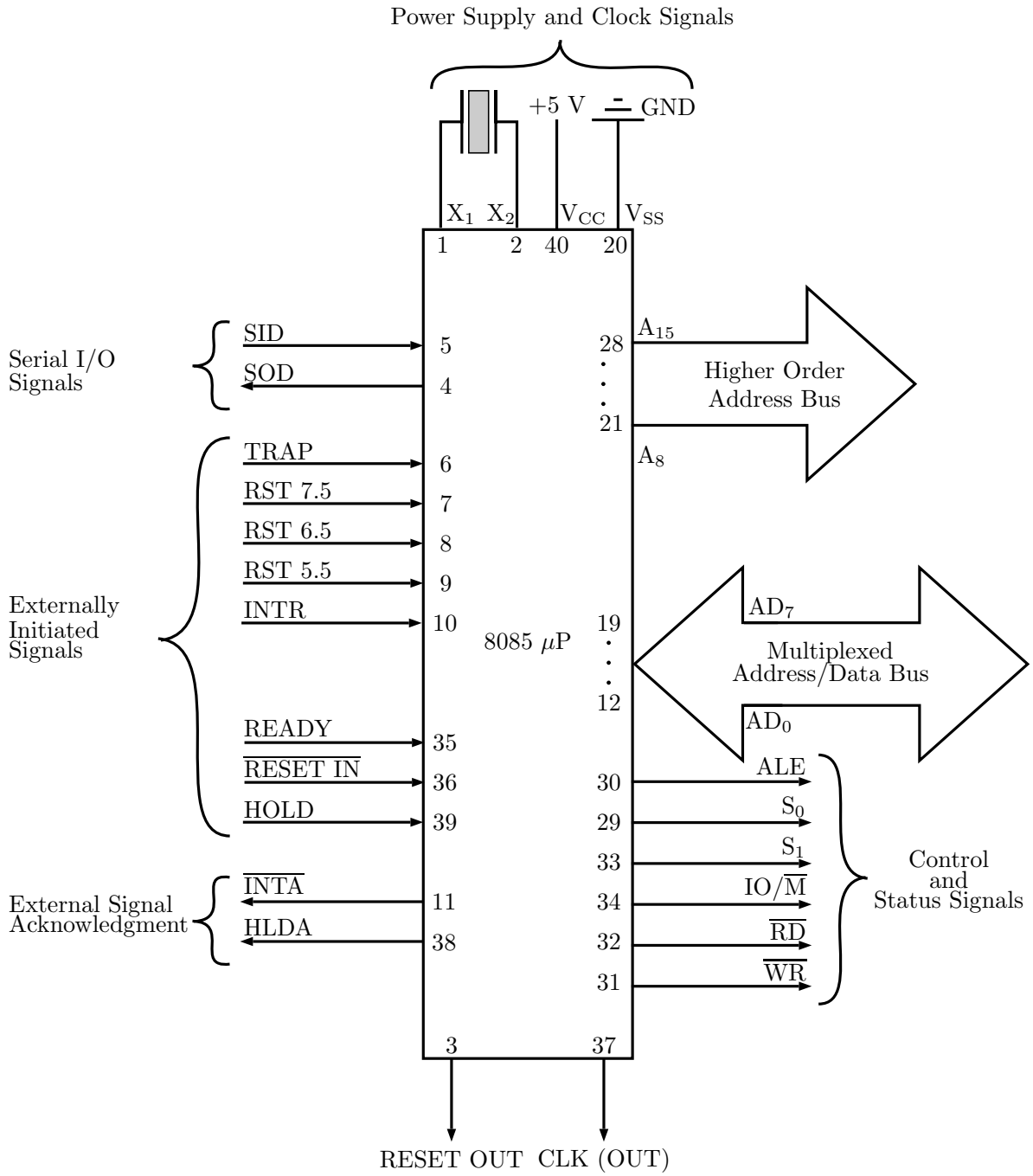
6.1 Pin Diagram

The 8085 microprocessor's pinout is shown below.



6.2 Pin Configuration

The 8085 microprocessor signals can be classified into following groups as shown in the figure below.



Address Bus

The 8085 has 16 signal lines (pins) that are used as the address bus; however these lines are split into two segments: $A_{15} - A_8$ and $AD_7 - AD_0$. The eight signal lines, $A_{15} - A_8$, are unidirectional and used for the most significant bits, called the high-order address, of a 16-bit address. The signal lines $AD_7 - AD_0$ are used for a dual purpose, as explained in the next section.

Multiplexed Address/Data Bus

The signal lines $AD_7 - AD_0$ are bidirectional: they serve a dual purpose. They are used as the low-order address bus as well as the data bus. In executing an instruction, during the earlier part of the cycle, these lines are used as the low-order address bus. During the later part of the cycle, these lines are used as the data bus. (This is also known as multiplexing the bus.)

If we had 16 pins dedicated to address bus and 8 pins dedicated to data bus, then 24 pins are used only for addressing and data transfer. Since we do not require address and data at the same time (i.e., first we need to know the address and then in that address we perform the data transfer), we can share (multiplex) some of the pins for handling both address and data, and use the remaining pins for other functions. Thus, by multiplexing the address and data bus, we increase the functionality of the microprocessor and reduce pin count, thereby reducing the size of the microprocessor.

Control and Status Signals

This group of signals includes two control signals ($\overline{RD}$ and $\overline{WR}$), three status signals ($IO/\overline{M}$, S_1, S_0) to identify the nature of the operation, and one special signal (ALE) to indicate the beginning of the operation. These signals are as follows:

- ALE (output) — Address Latch Enable: This is a positive going pulse generated every time the 8085 begins an operation (machine cycle); it indicates that the bits on $AD_7 - AD_0$ are address bits. This signal is used primarily to latch the low-order address from the multiplexed bus and generate a separate set of eight address lines, $A_7 - A_0$.

- $\overline{RD}$ (output) — Read: This is a Read control signal (active low). This signal indicates that the data is to be read from the selected I/O or memory device, i.e., take the data from I/O or Memory and put it in the data bus.

- $\overline{WR}$ (output) — Write: This is a Write control signal (active low). This signal indicates that the data from the data bus is to be written into a selected I/O or memory device, i.e., give the data from μP to I/O or Memory.

- IO/$\overline{M}$ (output) — This is a status signal used to differentiate between I/O and Memory operations. When it is high, it indicates I/O operation; when it is low, it indicates a memory operation. This signal is combined with $\overline{RD}$ and $\overline{WR}$ to generate I/O and memory control signals.

- S_1 and S_0 (output) — These status signals, similar to IO/$\overline{M}$, can identify various operations, but they are rarely used in small systems.

All the operations and their associated status signals are listed in the table below.

	Status			
Machine Cycle	**IO/$\overline{M}$**	**S_1**	**S_0**	**Control Signals**
Opcode Fetch	0	1	1	$\overline{RD} = 0$
Memory Read	0	1	0	$\overline{RD} = 0$
Memory Write	0	0	1	$\overline{WR} = 0$
I/O Read	1	1	0	$\overline{RD} = 0$
I/O Write	1	0	1	$\overline{WR} = 0$
Interrupt Acknowledge	1	1	1	$\overline{INTA} = 0$
Halt	Z	0	0	
Hold	Z	X	X	$\overline{RD}$, $\overline{WR}$ = Z and $\overline{INTA} = 1$
Reset	Z	X	X	
Bus Idle	0	0	0	—

NOTE: Z = Tri-state (High Impedance), X = Unspecified (Don't Care)

Power Supply and Clock Signals

The power supply and frequency signals are as follow:

- V_{CC}: +5 V power supply.
- V_{SS}: Ground Reference.
- X_1, X_2 (input): A crystal is connected a these two pins. The frequency is internally divided by two; therefore to operate a system at 3 MHz, the crystal should have a frequency of 6 MHz.
- CLK (OUT) (output) — Clock Output: This signal can be used as the system clock for other devices.

Externally Initiated Signals, including Interrupts

The 8085 has five interrupt signals that can be used to interrupt a program execution. The microprocessor acknowledges an interrupt request by the $\overline{\text{INTA}}$ signal.

- INTR (input) — Interrupt Request: This is used as a general purpose interrupt.
- $\overline{\text{INTA}}$ (output) — Interrupt Acknowledge: This is used to acknowledge an interrupt.
- RST 7.5, RST 6.5, RST 5.5 (inputs) — Restart Interrupts: These are vectored interrupts that transfer the program control to specific memory locations. They have higher priorities than the INTR interrupt. Among these the priority order is 7.5, 6.5, and 5.5.
- TRAP (input): This is a non-maskable interrupt and has the highest priority.

In addition to the interrupts, three pins — $\overline{\text{RESET IN}}$, HOLD, and READY — accept the externally initiated signals as inputs. To respond to the HOLD request, the 8085 has one signal called HLDA. The functions of these signals are discussed below.

- $\overline{\text{RESET IN}}$ (input): When the signal on this pin goes low, the program counter is set to zero, the buses are tri-stated, and the microprocessor is reset.

- RESET OUT (output): This signal indicates that the microprocessor is being reset. The signal can be used to reset other devices.
- HOLD (input): This signal indicates that a peripheral is requesting the use of the address and data buses. When HOLD pin is activated by an external signal, the microprocessor relinquishes control of buses and allow the external device to directly use them. For example, the HOLD signal is used in Direct Memory Access (DMA) data transfer.
- HLDA (output) — Hold Acknowledge: This signal acknowledges the HOLD request.
- READY (input): This signal is used to synchronize slower peripherals with the microprocessor. When this pin is low, the microprocessor enters into a Wait State. When the peripheral is ready to send or accept data, the READY pin gets high.

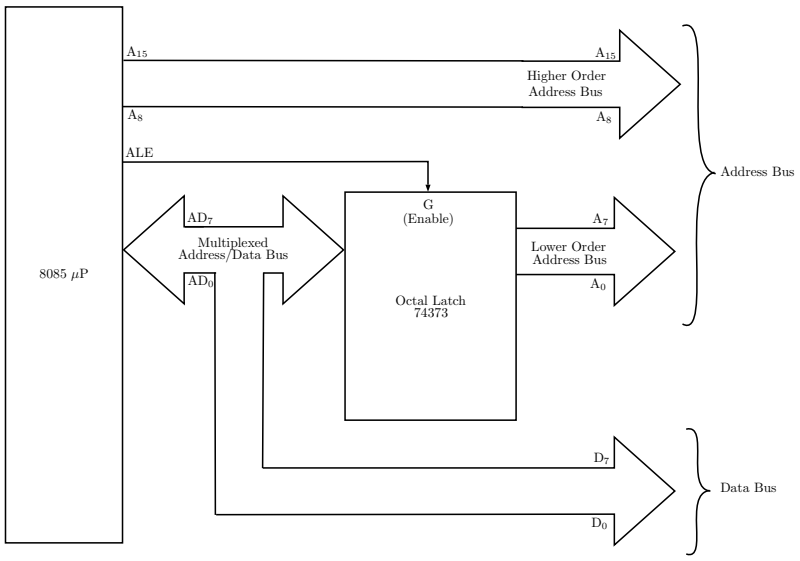
Serial I/O

The 8085 has two signals to implement serial transmission:

- SID (input) — Serial Input Data: This signal is used to input serial data.
- SOD (output) — Serial Output Data: This signal is used to output serial data

6.3 Demultiplexing the bus $AD_7 - AD_0$

In order to save the pin count and increase the functionality of the microprocessor, we went for multiplexed address and data bus. In 8085, the higher order address lines i.e., $A_{15} - A_8$ are directly available, but the lower order address lines i.e., $A_7 - A_0$ are multiplexed with the data lines i.e., $D_7 - D_0$ to get the multiplexed address/data bus i.e., $AD_7 - AD_0$. To properly identify a peripheral, we first need to know it's address. After that, data transfer can take place. In 8085, address is of 16 bits given by $A_{15} - A_0$. Thus we need to demultiplex the $AD_7 - AD_0$ bus. This is done by using a latch and the ALE signal as shown in the figure below.



When ALE is high (during T_1), the latch is enabled; this means that whatever data is there in the multiplexed $AD_7 - AD_0$ bus gets latched and is treated as the lower order address. When ALE is low (after T_1), the latch is disabled; so whatever data is available in the multiplexed bus is treated as data. As the latch is disabled after T_1, the lower order address bus does not change its value.

For example, let's say data `CD` is to be written in the memory address `F003`. The higher order address bus, $A_{15} - A_8$ will have the value `F0`. Now during T_1, ALE goes high; latch is enabled. So the value `03` available in the multiplexed bus will be latched to the output of 74373 IC and we get the lower order address, $A_7 - A_0$ as `03`. After T_1, ALE goes low; latch is disabled. The value `CD` available in the multiplexed bus will be treated as the data.

Previous Questions

2018

Addn. Endsem

1. What is the purpose of READY pin in 8085? [1]
2. Explain about program counter(PC), stack pointer(SP), and flag register in 8085. [1]

Endsem

1. What is the purpose of HOLD and READY pins in 8085? [1+1]

Midsem

1. Draw the architecture of 8085. [2]
2. What is the purpose of Program counter(PC). Draw a suitable circuit to generate $\overline{\text{MEMR}}$, $\overline{MEMW}$, $\overline{\text{IOR}}$ and $\overline{\text{IOW}}$ signals in an 8085 microprocessor system. [1+2]
3. Write the 8-bit format of Flag register in 8085 microprocessor. [1]

2017

Endsem

1. Define tristate. Why are they essential in a bus oriented system? [1+1]
2. How is the lower address bus and data bus demultiplexed externally in an 8085 microprocessor system. [2]

Midsem

1. What are tristate devices. Why are they essential in bus oriented system? [1]
2. Draw internal architecture diagram of 8085 microprocessor. [2]
3. Explain the role of program counter. [1]
4. What is the function of READY pin in an 8085 microprocessor? [1]
5. What are the different Control and Status signals available in 8085 microprocessor? [2]

6. Draw a suitable circuit to generate $\overline{\text{MEMR}}$ signal out of IO/$\overline{\text{M}}$ and $\overline{\text{RD}}$ control signal for a 8085 microprocessor system. [1]

2016

Endsem

1. What is the purpose of ALE pin in 8085? [1]
2. Explain about program counter(PC) and flag register of an 8085 microprocessor. [3]
3. Draw a suitable circuit to generate $\overline{\text{MEMR}}$ and $\overline{\text{IOW}}$ signals out of IO/$\overline{\text{M}}$, $\overline{\text{RD}}$ and $\overline{\text{WR}}$ control signals in an 8085 microprocessor system. [2]

Midsem

1. What is the use of ALE pin in 8085 microprocessor. [1]
2. Draw the internal architecture of 8085 microprocessor. [2]

2015

Endsem

1. Write the 8-bit format of flag register in 8085 microprocessor. With suitable examples discuss how PF and ZF flags are set. [1+2]

Midsem

1. Discuss about accumulator and program counter of 8085 microprocessor. [2]
2. After addition of two signed numbers in an 8085 system, the status of the flag register is `00010101`. What information does it convey about the flags and the sum? [2]
3. What is the function of $\overline{\text{RESET IN}}$ and READY pins of 8085 microprocessor? [2]

2014

Endsem

1. What are the two 16-bit special purpose registers available in 8085 and specify the function of the registers. [1]

2. Draw the internal architecture of 8085 microprocessor and explain about accumulator and flag register. [2]

Midsem

1. Explain the role of accumulator and program counter. [2]
2. After addition of two signed numbers in an 8085 system, the contents of flag register is 94_H. What information does it convey about the data conditions of the sum? [2]
3. What is the function of S_1 and S_0, $\overline{\text{RESET IN}}$ pins of 8085 microprocessor? [1+1]
4. What are tri-state logic devices? Why are they essential in a bus-oriented system? [2]

2013

Endsem

1. Specify the content of Program counter presently, if the 8085 has just fetched a byte from a memory location $F42C_H$. [1]
2. After the addition of two signed numbers in an 8085 system, the status of the Flag register is 00010101. What information does it convey about the data conditions of the sum? [2]
3. What are tri-state devices? Why are they essential in a bus-oriented system? [2]
4. What are the control signals necessary in the memory-mapped I/O scheme? How are they generated from $\overline{\text{RD}}$, $\overline{\text{WR}}$ and IO/$\overline{\text{M}}$ signal of 8085 microprocessor? [1+1]

Midsem

1. How is the Low-order address bus and data bus demultiplexed externally in an 8085 microprocessor system. [2]
2. What is the function of READY and X_1, X_2 pins in an 8085 microprocessor. [2]
3. Explain the role of flag register in 8085 microprocessor. [2]
4. What are tri-state logic devices? Why are they essential in a bus-oriented system? [2]