Microprocessor

Unit 2:
Basic Computer Architecture

Course Outline

Bo PG 95

- Microprocessor Architecture & Operations
  - 8085 Microprocessor Architecture
  - Address, Data And Control Buses
  - 8085 Pin Diagram and Functions
  - Operations in 8085
    - Microprocessor Initiated Operations
    - Internal Data Operation & Registers
    - Externally Initiated Operations
  - Memory and Memory Operations
  - Flag and Flag Register
  - Multiplexing & De-multiplexing of address/data bus
  - Generation Of Control Signals

8085 Microprocessor

- 8085 Microprocessor Architecture
- Address, Data And Control Buses
- 8085 Pin Diagram and Functions
  - See yourself, I have already uploaded uncompiled docx file

Operations Types in 8085

- All of the operations of the microprocessor can be classified into three types:
  - Microprocessor Initiated Operations
  - Internal Operations
  - Externally Initiated Operations

Microprocessor Initiated Operation

- These are the operations that the microprocessor itself starts.
- There are basically four different microprocessor initiated operations:
  - Memory Read
  - Memory Write
  - I/O Read (Get input from the input device)
  - I/O Write (Send data to the output device)

Internal Data Operations

- The 8085 can perform a number of internal operations such as:
  - storing data,
  - arithmetic and logic operations,
  - testing for condition,
  - Sequencing the execution of instructions
  - Store/retrieve data from the stack during execution.
- To perform these operations, the microprocessor needs an internal registers like B, C, D, E, H, L, Accumulator, temporary registers, Flags, and others.

Externally Initiated Operations

- 8085 microprocessor support some Externally initiated operations, which are also known as Peripheral operations.
- Different external input/output devices or signals can initiate these type operations.
- External devices can initiate one of the 4 following operations:
  - Reset
  - Interrupt
  - Ready
  - Hold

Externally Initiated Operations (contd…)

- Reset
  - All operations are stopped and the program counter is reset to 0000.
  - When this RESET pin is activated by any external key, then all the internal operations are suspended for that time.
  - After that the execution of the program can begin at the zero memory address.
- Interrupt
  - 8085 MP chip have some pins for interrupt like TRAP, RST 5.5, RST 6.5 and RST 7.5.
  - The microprocessor's operations are interrupted and the microprocessor executes other emergency operations, what is called a "service routine".
  - This routine handles the interrupt, (perform the necessary operations).
  - Then the microprocessor returns to its previous operations and continues.

Externally Initiated Operations (contd…)

- Ready
  - The 8085 has a pin called READY.
  - This pin is used by external devices to stop the 8085 until they catch up.
  - The Input/Output devices that are connected to microprocessor are of different speed,
    - which is need to be synchronized with the speed of microprocessor.
  - This signal is used mainly to synchronize slower external devices with the microprocessor.

Externally Initiated Operations (contd…)

- Hold
  - The 8085 has a pin called HOLD.
  - This pin is used by external devices to gain control of the buses.
  - When the HOLD signal is activated by an external device, the 8085 stops executing instructions and stops using the buses.
  - This would allow external devices to control the information on the buses.
  - Eg: DMA
  - In this DMA, the external Input/Output devices are directly communicate with the memory without interfering the processor every time.

Data flow between memory and MPU

- First of all, the 16-bit address is placed on the address bus from the PC.
- Lets say, the address is 1020H where the data is placed.
- The higher order address i.e. 10H is placed on the address bus A8-A15 while the lower order address i.e. 20H is placed on the multiplexed address and data bus AD0-AD7.
- The lower order address continues to remain on this address bus so long as ALE (Address Latch Enable) remains high. Once ALE goes low, it carries data.
- The control unit sends the signal to indicate what type of the operations is to be performed.
- Since, the data is to be read from the memory therefore it is sends to enable the memory chip.
- The byte, from the memory location is then placed in the data bus, is placed on the data bus and sent to the instruction decoder.
- The instruction is decoded and accordingly the task is performed by the ALU.

Flag and Flag Register

- The Flag register is a Special Purpose Register which shows the status of the task (microprocessor calculation).
- This is an 8-bit register but the only 5bit is used for the operation.
- The flag becomes set or reset after arithmetic and logical operation.
- The flag register has 5 flags which are
  1. Sign flag,
  2. Zero flag,
  3. Auxiliary carry flag,
  4. Parity flag, and
  5. the Carry flag.

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

Sign Flag

- Represented by the symbol S.
- After any type of arithmetic operation or logical operation,
  - if the value of D7 becomes 1 it basically shows that the number is negative and the sign flag is now set,
  - if the value of D7 becomes is 0, it basically shows that the number is positive and the sign flag is now reset.
- Eg:

```
MVI A 25H (load the value 25H in register A)
MVI B 05H (load the value 05H in register B)
SUB B (A = A – B)
```

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

- These set of instructions will basically reset the sign flag to 0 as 25 – 5 is always a positive number.

Zero Flag

- After any type of arithmetical or logical operation, if the output becomes 0 (00)H,
  - then zero flag is said to be set with value 1,
  - if not it becomes reset with the value 0.

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

- Eg: `MVI A 25H (load the value 25H in register A)`

  `SUB A (A = A – A)`

- These set of instructions will basically results 00H as 25 – 25 is always zero, so the zero flag is set to 1.

Carry Flag

- Carry is generated when performing n bit operations and the result is more than n bits,
  - then this flag becomes set i.e. 1,
  - otherwise it becomes reset i.e. 0.

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

- Eg:

```
MVI A 25H (load the value 25H in register A)
MVI B 85H (load the value 05H in register B)
ADD B (A = A + B)
```

- These set of instructions will basically results answer with borrow.
- So, carry flag is set to 1.

Auxiliary Carry Flag

- Carry is generated when performing n bit operations and the result is more than n bits,
  - then this flag becomes set i.e. 1,
  - otherwise it becomes reset i.e. 0.
- This is the only flag register which is not accessible by the programmer
- Eg:

```
MVI A 2CH (load the value 25H in register A)
MVI B 85H (load the value 05H in register B)
ADD B (A = A + B)
```

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

- These set of instructions will basically results auxiliary during calculation as addition of lower order nibbles C and 5 will generate a carry.
- So, auxiliary carry flag is set to 1.

Parity Flag

- If after any arithmetic or logical operation the result has even parity, an even number of 1 bits,
  - the parity register becomes set i.e. 1,
  - otherwise it becomes reset i.e. 0.

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

- Eg:

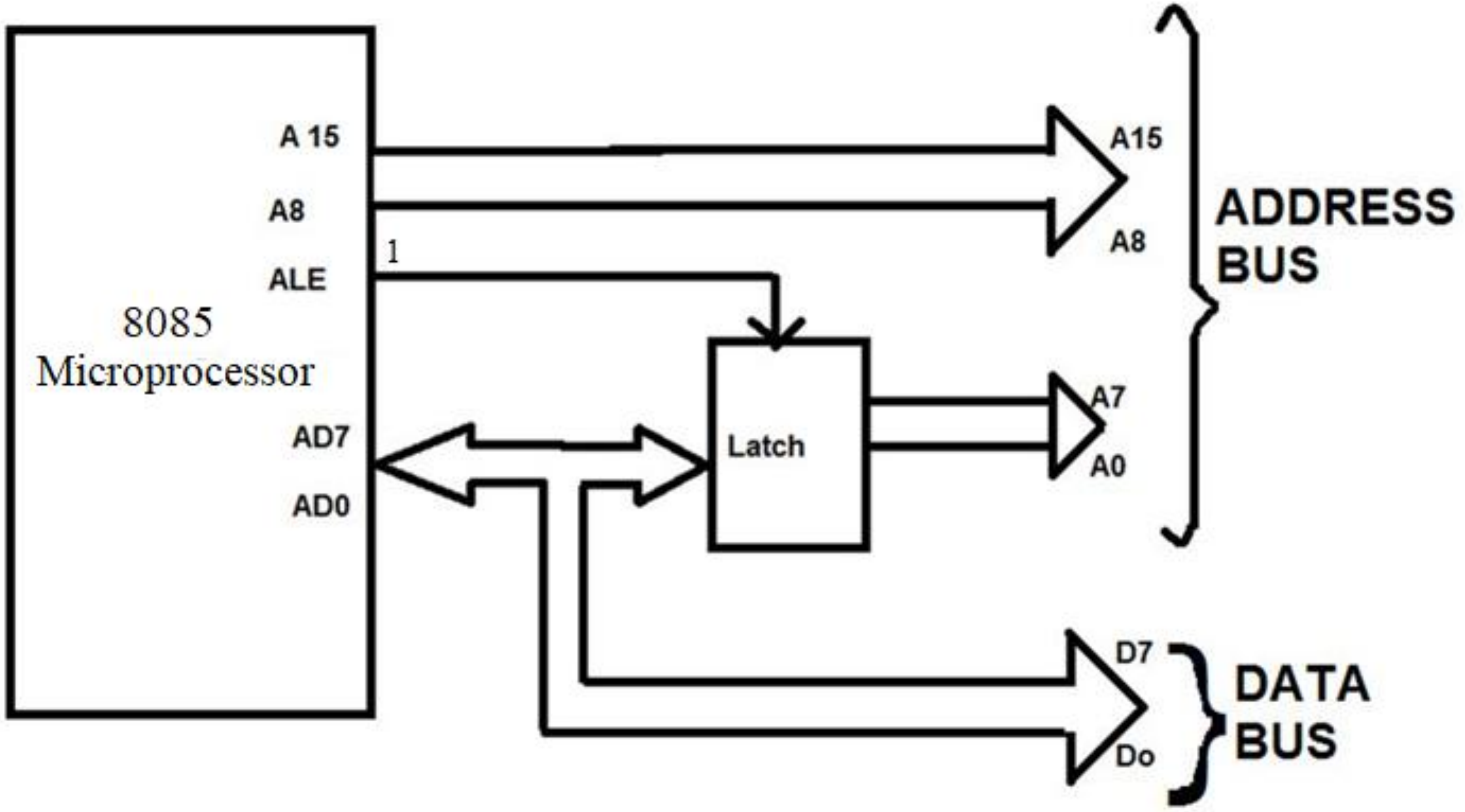
```
MVI A 01H (load the value 01H in register A)
MVI B 05H (load the value 05H in register B)
ADD B (A = A + B)
```

- This instruction will results 06H (i.e BCD -> 00000110), which contains even number of ones.
- So, parity flag is set to 1.

Multiplexing and Demultiplexing of Address-Data bus in 8085

Given, the ALE operates as a pulse during T1 we will be able to latch the address the when ALE goes low the address is saved and the AD7-AD0 lines can be used for their purpose as the bi-direction data lines.

Multiplexing and Demultiplexing of Address-Data bus in 8085



Control Bus and Control Signal

Control Bus in 8085

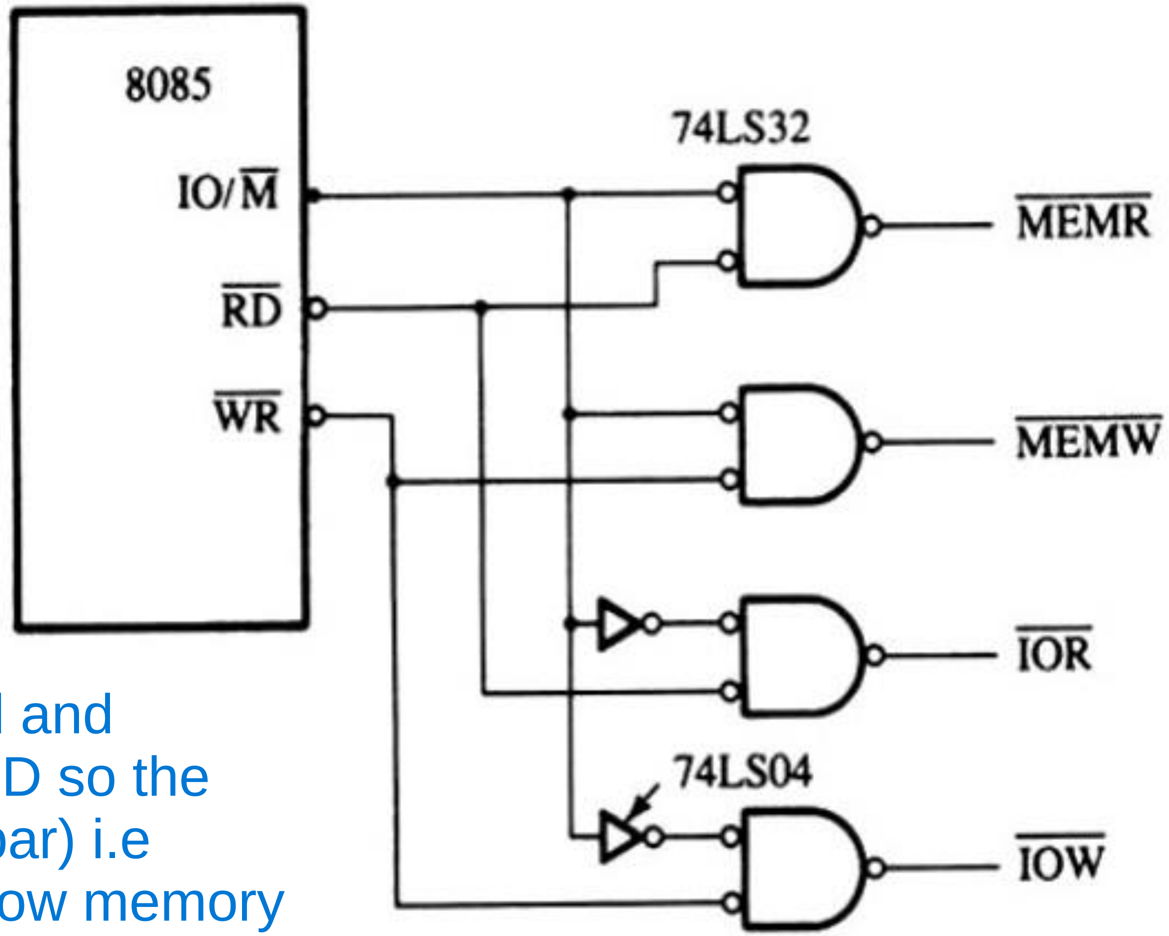
- This is a group of parallel lines used by the microprocessor to issue control signals such as IO/M, RD, WR.
- As you have known from the pin diagram of 8085, there are three different pins at which the microprocessor issues these control signals.
    1. Pin 30 for IO/M
    2. Pin 31 for WR
    3. Pin 32 for RD
- Control signals are the signals used to control the operations related to memory and other associated peripherals.

- How do we control *which operation* (read/write) is supposed to be done at *which location* (Memory/IO)?
    - The same data bus is used for *read* as well as *write* operations.
    - These two types of operations also need to be done at two possible locations (memory or I/O).
- The control signals issued by the microprocessor distinguish between these different types of operations.

Control Signals in 8085

- The three control signals are explained here in brief.
  - **IO/$\overline{\mathbf{M}}$ (Control signal)**
    - This signal specifies whether the operation (read or write) is being performed on memory or an i/o device.
  - **$\overline{\boldsymbol{RD}}$ (Control signal)**
    - Goes low for read operation and becomes high otherwise.
  - **$\overline{\boldsymbol{WR}}$ (Control signal)**
    - Goes low for a write operation and becomes high otherwise.

Control Signal Generation: Using NAND Gate



Here Nand gate is used and first input is IO/M and RD so the output is memoryread(bar) i.e output is also in active low memory but then we put not gate the output is IOR(bar) i.e I/O is active low

Control Signals Generation: Using Decoder

IO/M(bar)	RD(bar)	WR(b)	output
0	0	0	0
0	0	1	MEMR
0	1	0	MEMW
0	1	1	0
1	0	0	0
1	0	1	IOR
1	1	0	IOW
1	1	1	0

Here, In IO/M 0 means memory and 1 means I/M and RD(Bar) 0 means read and WR(b) 0 means write. and both low means no operation

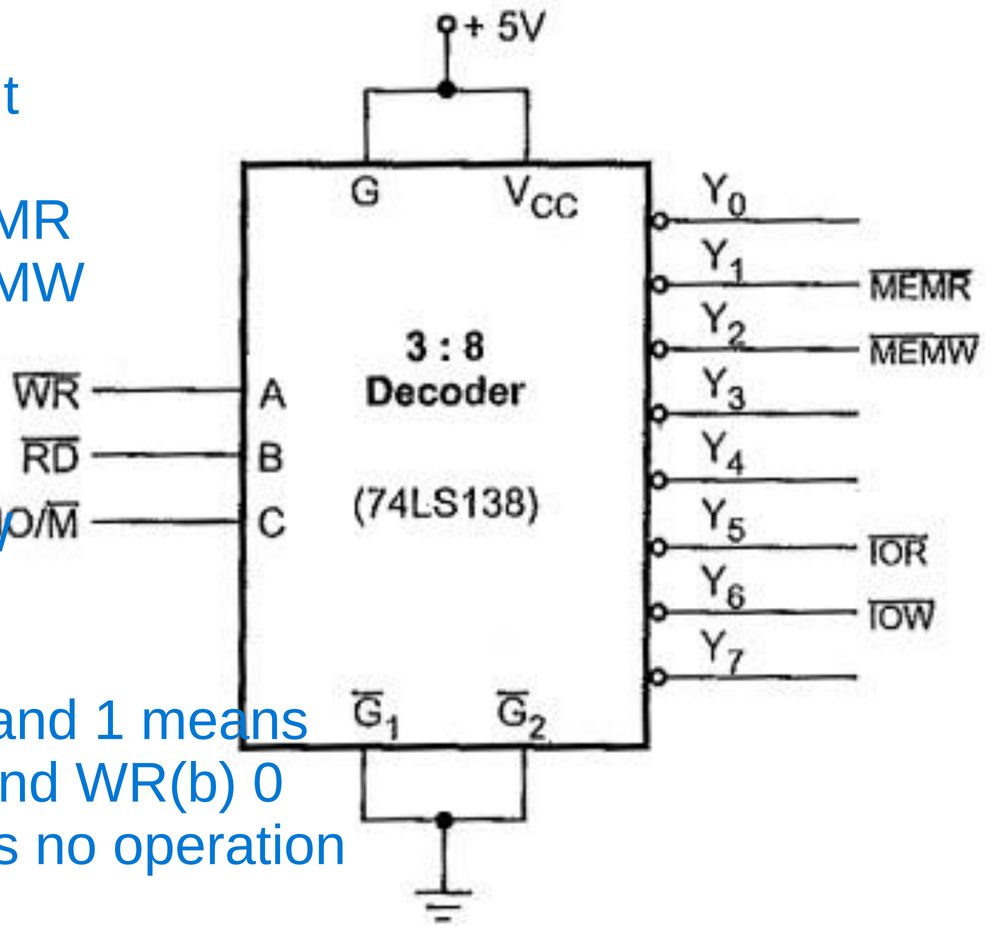


Fig. 4.9 Generation of control signals using 3 : 8 decoder

Truth Table of Control Signal

- The table explaining the control signals and status signals during various processes.

Operation	Status			Control signals
	IO/M	S1	S0	
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
Bus Idle	0	0	0	$\overline{INTA}, \overline{WR}, \overline{RD}$= 1