Programmable Interface Devices -8155

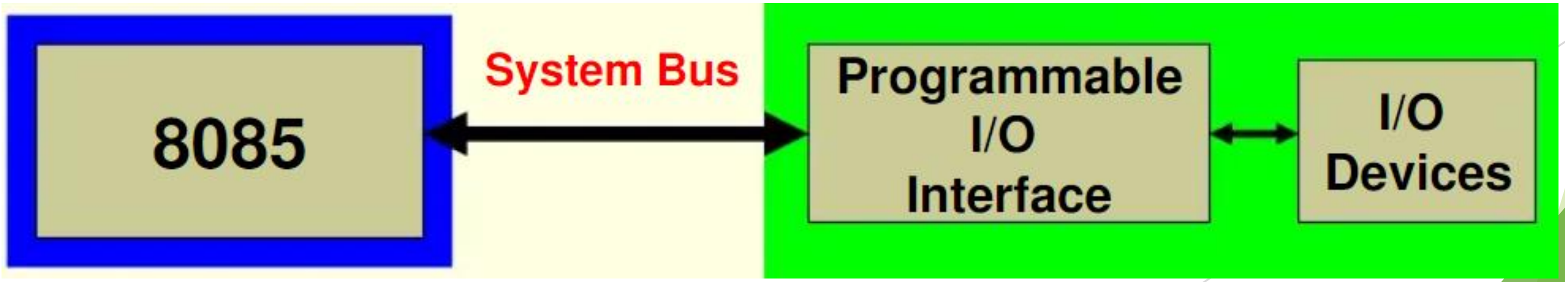
by Dr. Rashmi Panda
Dept. of Electronics and communication Engineering
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, RANCHI

Connecting Peripheral Devices with the MP

- I/O devices can not be connected directly with the MPU due to compatibility of data formats and speed between the peripheral devices and the MPU. Thus Intel designed several PPI chips (8155, 8156, 8255 etc.) to interface with i/O devices.

- Peripheral devices are connected to the MPU through interfacing circuits, which are known as I/O ports, where port is simply a buffer or latch.

- Interfacing devices contain one or more registers each of which is addressable and strobed by the MPU to latch the data on the data bus at the appropriate time during input or output operation.



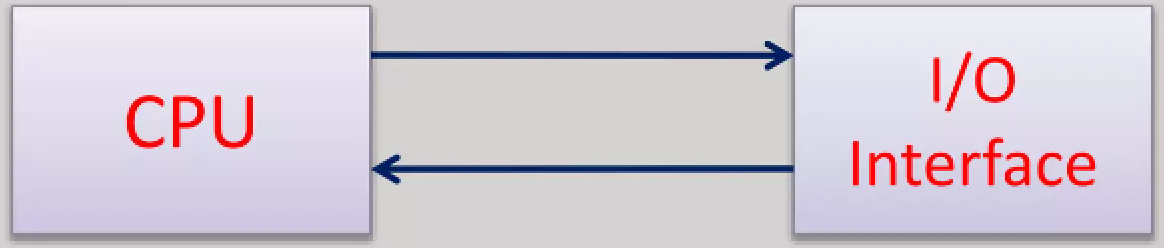
Communication of devices

Synchronous

Devices Share a Common Clock

Asynchronous

Devices communicate with the help of control signals.

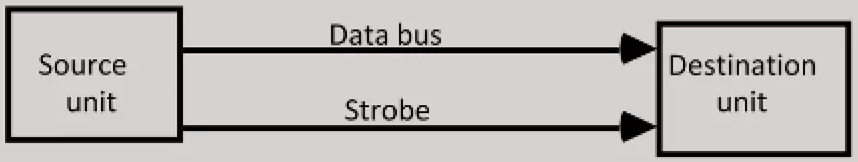


Strobe Control

- Single control line for transfer.
- Either the source or the destination can initiate the transfer.

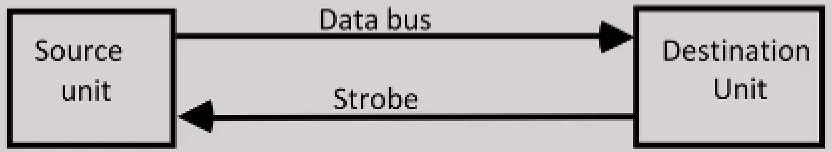
Source-Initiated Strobe
for Data Transfer

Block Diagram



Destination-Initiated Strobe
for Data Transfer

Block Diagram



Two Ways of Asynchronous Transfer

1. Strobe Signal

   It has only one control signal for initiating the data transfer

2. Handshaking

   It has two control signal for communication

Data Transfer by Source (Destination)

1. Source places the data on the data bus.
2. Then it activates the strobe pulse.
3. Both data and the strobe remains active for sufficient period of time so that the destination unit can receive it.
4. Destination uses the falling edge of the strobe pulse to transfer the data to its register.
5. Then the source removes the data from the data bus.
6. Then it disables the strobe signal.

Data Transfer by Destination

1. Destination activates the strobe pulse informing the source to provide data.
2. Source responds by placing the data.
3. The data will remain in valid state for sufficient clock pulse.
4. Falling edge may be used to transfer the data.
5. Destination then disable the strobe pulse.

Disadvantage of Strobe

- Source does not know whether destination has received the data item that is placed in the bus.
- Destination that initiates the transfer do not know whether the source has places the data.
- Handshake solves this problem.

Handshake

Basic Principle – two control signal for data transfer

1. One control line as the same direction of data flow from source to destination.

   It is used to inform the destination whether there is valid data in data bus.

2. Another control signal from the destination to source.

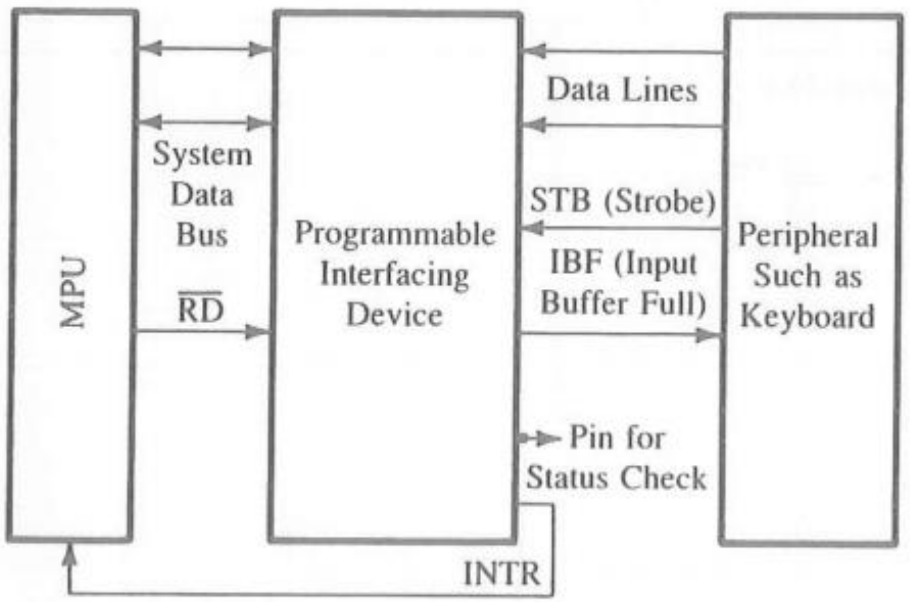
   It is used to inform the source whether it had accepted data.

* Allows arbitrary delays from one state to the next
* Permits each unit to respond at its own data transfer rate
* The rate of transfer is determined by the slower unit

Advantage of Handshake

- Handshaking provides high degree of flexibility and reliability because the successful completion of the data transfer relies on the active participation by both the units.
- If one unit is faulty data transfer will not be completed.
- Time out mechanism helps to detect the faulty system.
    - Time out is implemented by the internal clock that starts counting time when the unit enables one of its handshaking control signal. If the return handshake does not respond with in the time it assumes an error occurred.

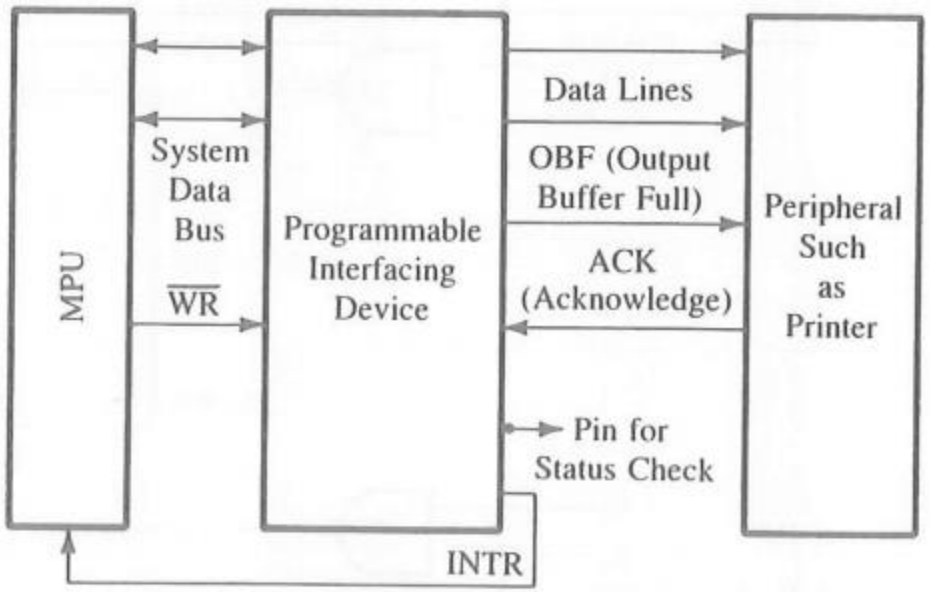
Handshake signals for data input



The steps in data input from a peripheral such as a keyboard are as follows:

1. A peripheral strobes or places a data byte in the input port and informs the interfacing device by sending handshake signal STB (Strobe).
2. The device informs the peripheral that its input port is full—do not send the next byte until this one has been read. This message is conveyed to the peripheral by sending handshake signal IBF (Input Buffer Full).
3. The MPU keeps checking the status until a byte is available. Or the interfacing device informs the MPU, by sending an interrupt, that it has a byte to be read.
4. The MPU reads the byte by sending control signal $\overline{RD}$.

Hand shake signals for data output



1. The MPU writes a byte into the output port of the programmable device by sending control signal $\overline{WR}$.
2. The device informs the peripheral, by sending handshake signal OBF (Output Buffer Full), that a byte is on the way.
3. The peripheral acknowledges the byte by sending back the ACK (Acknowledge) signal to the device.
4. The device interrupts the MPU to ask for the next byte, or the MPU finds out that the byte has been acknowledged through the status check.

Hand shake signals for data input and output

1. Handshake signals ACK and STB are input signals to the device and perform similar functions, although they are called by different names.
2. Handshake signals OBF and IBF are output signals from the device and perform similar functions (Buffer Full).

8155
A multipurpose programmable interface

8155 PPI

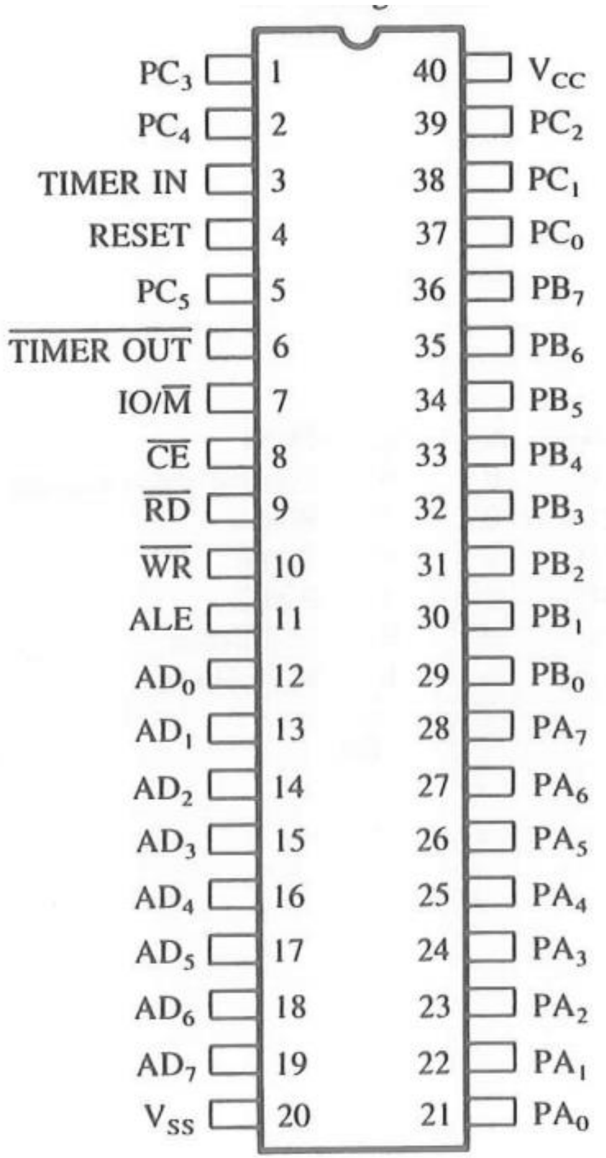
Designed to be compatible with 8085

- It includes
- 256 bytes of Read/Write memory
- Three I/O ports
  - Port A
  - Port B
  - Port C
- A 14-bit timer

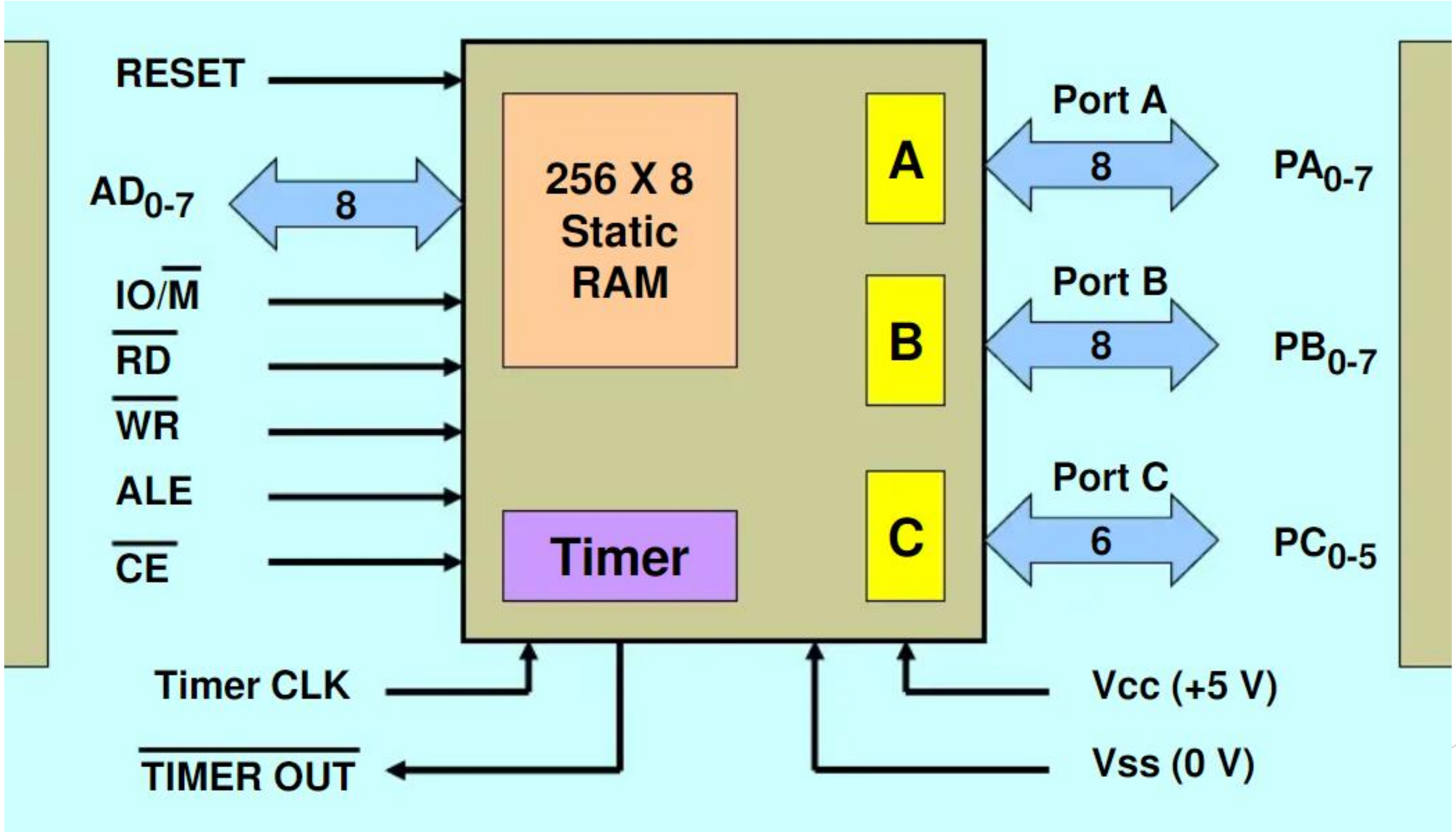
Features of 8155

- 1. Two programmable 8 bit i/o ports (port A and port B)
- 2. One programmable 6 bit i/o port (port C)
- 3. One programmable 14 bit binary counter and timer
- 4. **256 bytes** static RAM
- 5. Address bus AD0- AD7
- 6. Internal address latch to demultiplex AD0- AD7, using ALE line
- 7. It contains an internal select logic for memory and i/o using a command register and two I/O ports
- 8. It can be easily interfaced with 8085 microprocessor

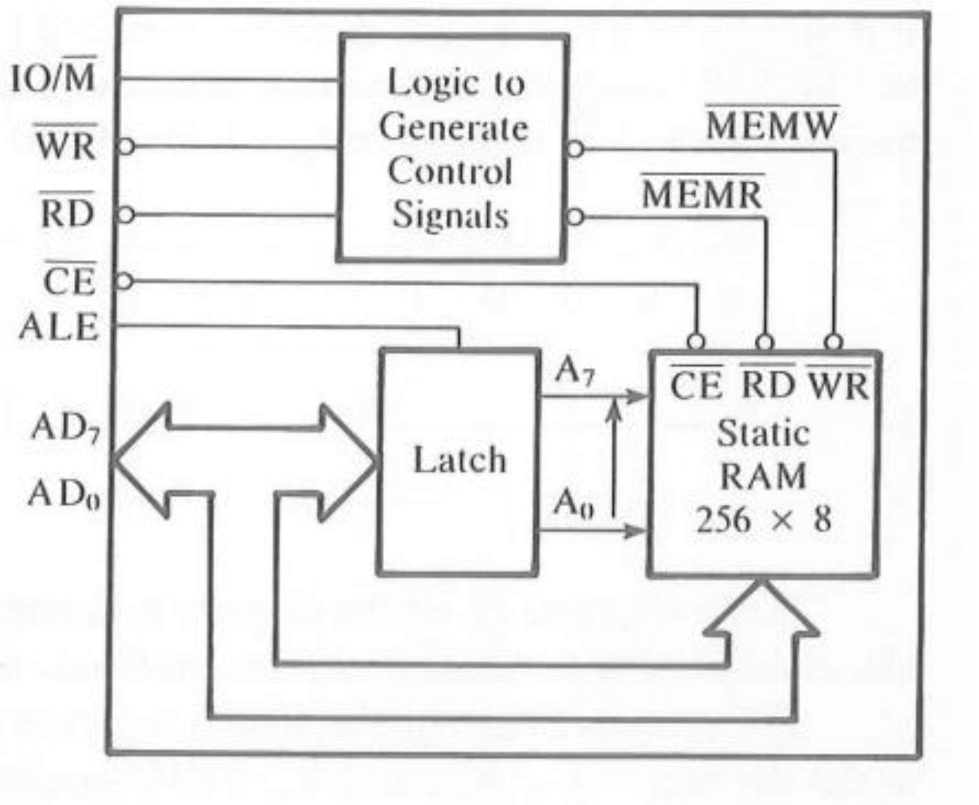
PIN Diagram - 8155



Block Diagram - 8155



Internal structure of 8155



Application design with 8155

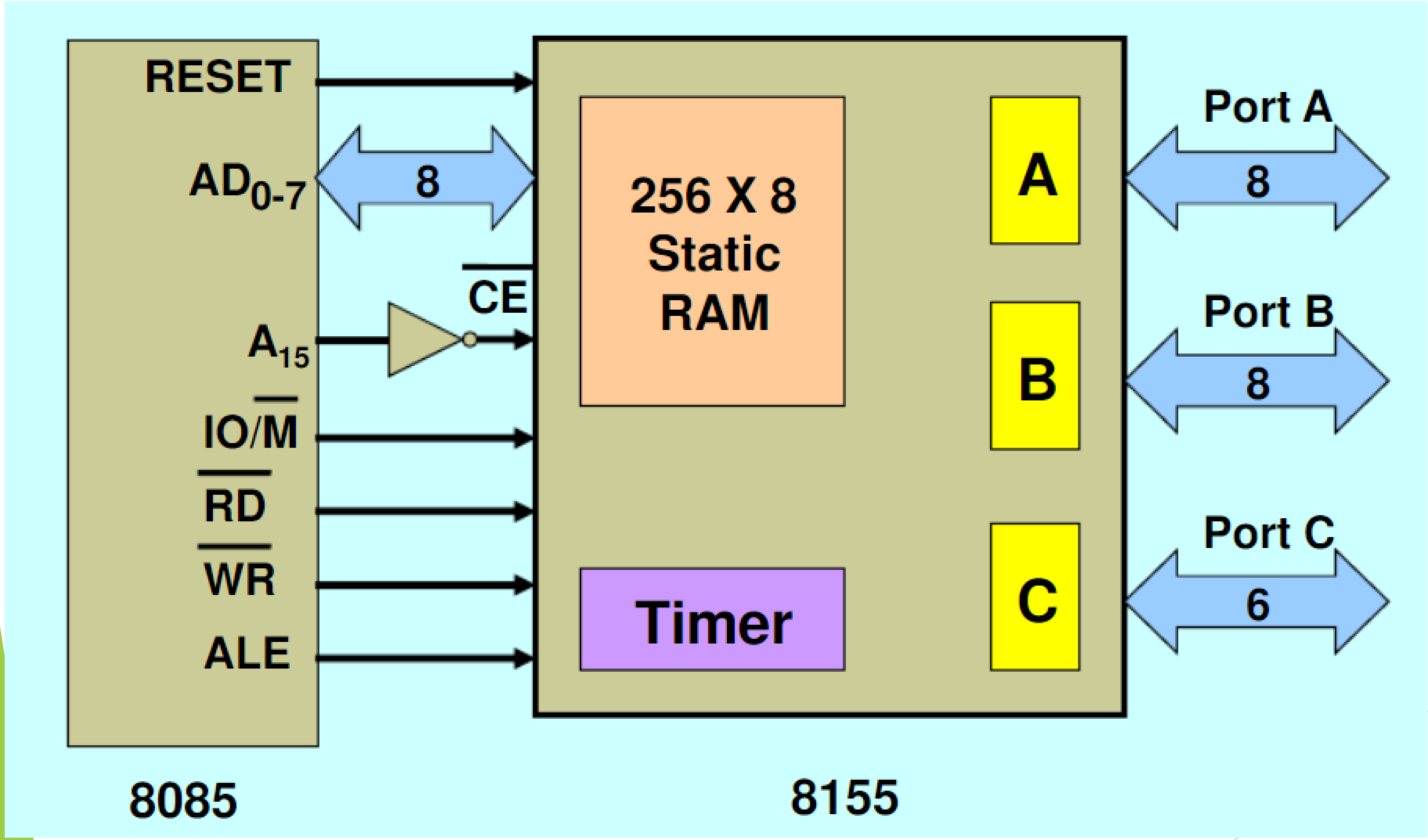
Objectives

- Interfacing 8155 with 8085
- Programming 8155

Interfacing 8155 with 8085

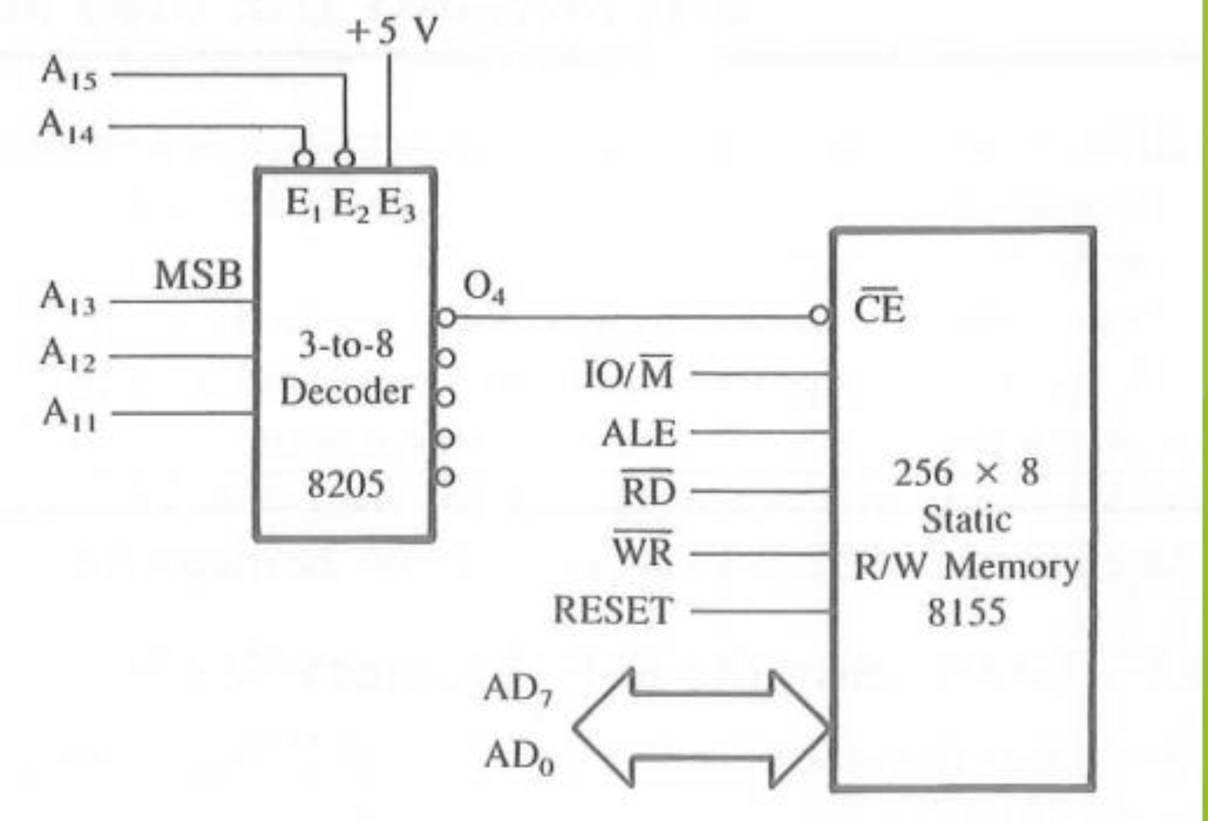
- The interfacing of 8155 with 8085 is done using I/O mapped I/O.
- 8155 has on chip de-multiplexing circuit, therefore AD 0 -AD 7 pins of 8085 are directly connected to AD 0 -AD 7 pins of 8155.
- 8155 also contains separate (internal) Control signal generator circuit, therefore the IO/M(bar), RD(bar) and WR(bar) control signals are directly connected to 8155.
- Reset out pin of 8085 is connected to reset pin of 8155.
- Chip enable is active low signal which is obtained by decoding high order address lines. (Decoder Circuit)

Interfacing 8085 with 8155



Interfacing the 8155 memory section Example

- Explain the decoding logic and the memory address range of 8155 shown in this figure.



Programming 8155

8155 is a Programmable Peripheral Interface

- 8085 can send data to 8155 using data bus
- This data can be
  - For I/O devices connected to 8155
  - Timer registers of 8155
  - Instruction/ Command word for 8155
- Commands for 8155 are stored in a 8-bitControl Register inside 8155

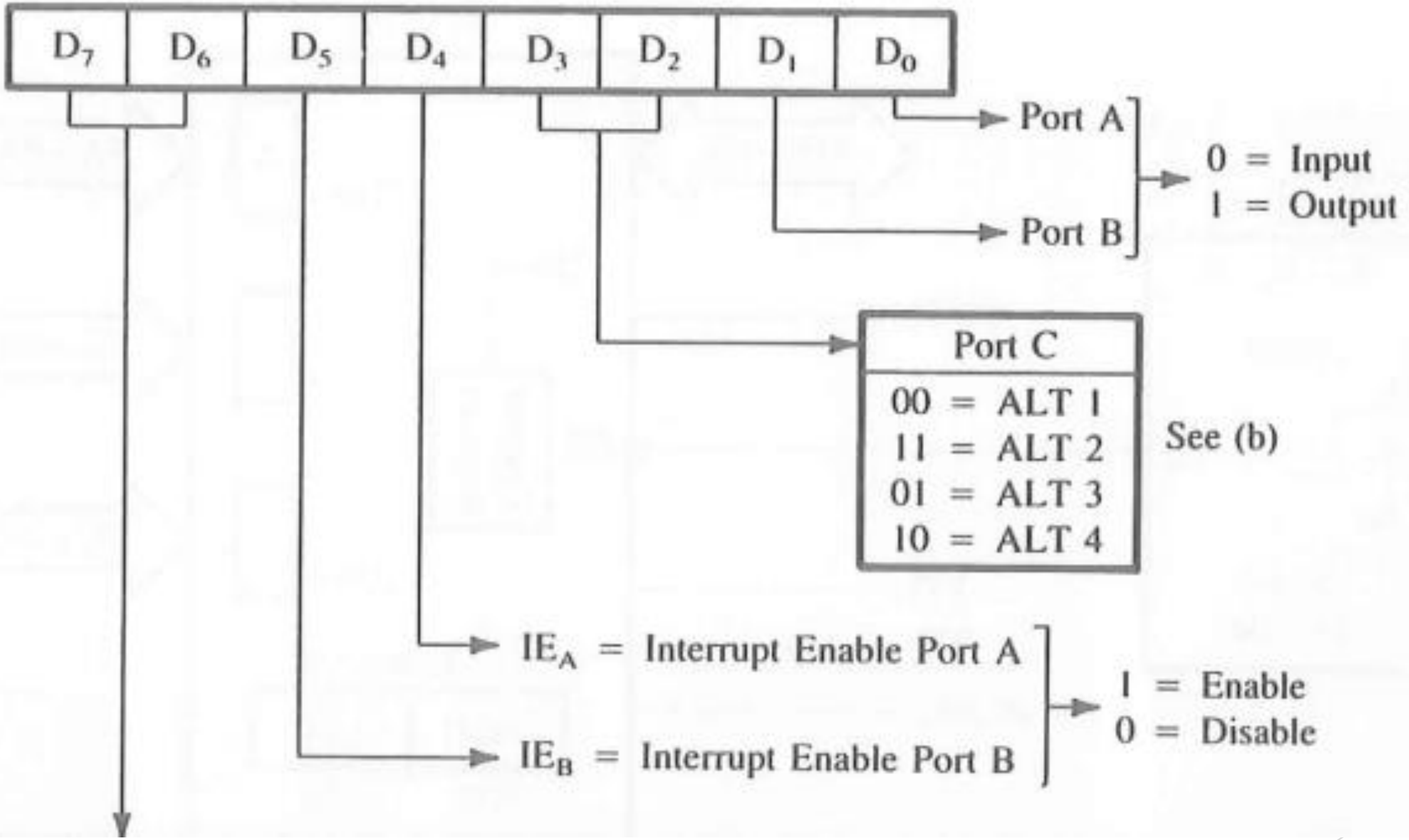
What type of Commands can be givento 8155?

- To configure the I/O ports as Input or Output
- To start/stop timer etc.
- To use handshake mode or not

Control word for 8155

- A command/instruction for 8155 is also called control word
- This control word is written to control register of 8155
- Control word of 8155 is of8-bits

Control Word Definition for 8155



Timer commands

Timer Commands

00 = NOP—No effect on timer

01 = Stop—Stop counting if timer is running; otherwise, no effect on timer

10 = Stop After TC (terminal count)—Stop after at end of the count if timer is running; otherwise, no effect on timer

11 = Start—Start timer if it is not running

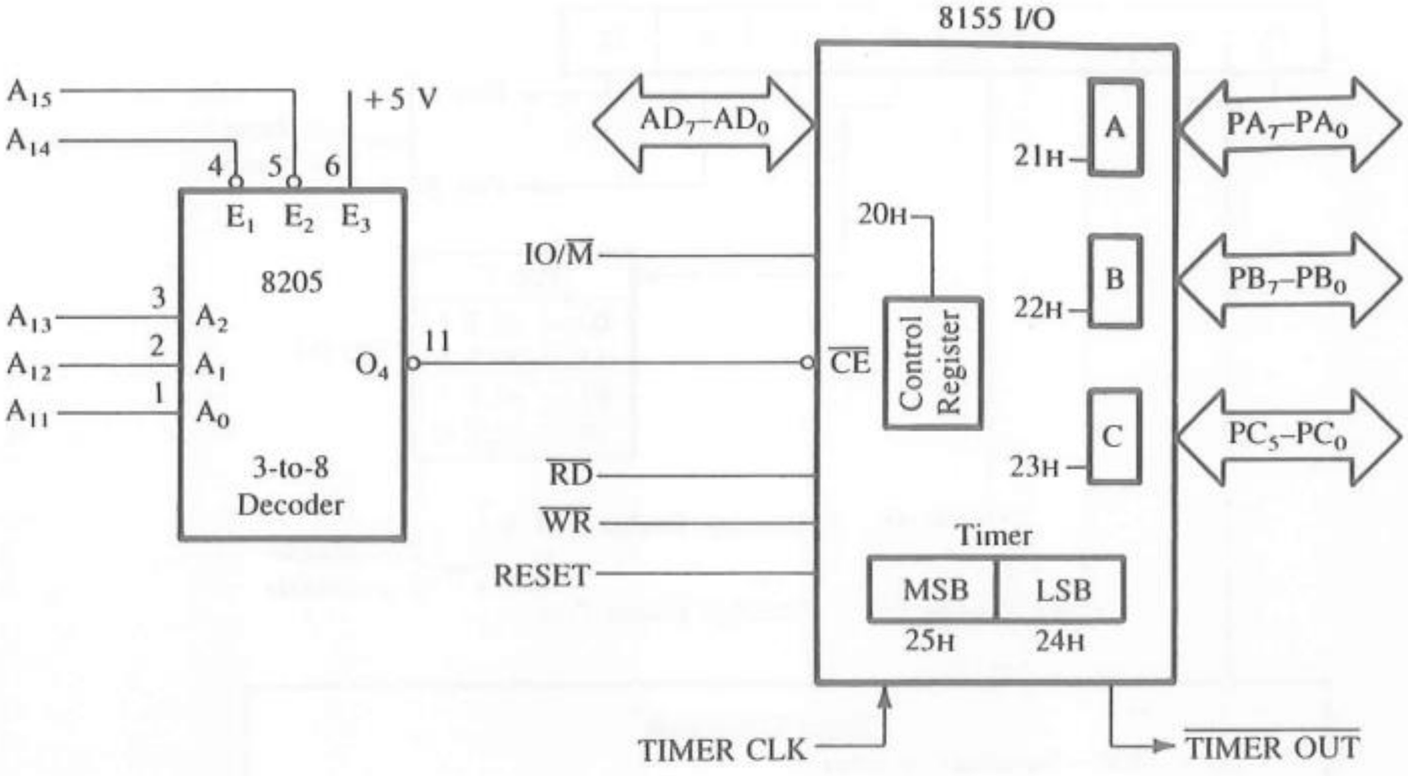
—If timer is running, stop at end of the count. Reload new mode and count, and start again

Table: ALT 1–ALT 4: Port C Bit Assignments, Defined by Bits D_3 and D_2 in the Control Register

ALT	D_3	D_2	PC_5	PC_4	PC_3	PC_2	PC_1	PC_0
ALT 1	0	0	I	I	I	I	I	I
ALT 2	1	1	O	O	O	O	O	O
ALT 3	0	1	O	O	O	$\overline{STB}_A$	BF_A	$INTR_A$
ALT 4	1	0	$\overline{STB}_B$	BF_B	$INTR_B$	$\overline{STB}_A$	BF_A	$INTR_A$

I = Input, STB = Strobe, INTR = Interrupt Request
O = Output, BF = Buffer Full, Subscript A = Port A
B = Port B

Determine the addresses of the control/status register, I/O ports, and timer registers



8155 timer and square wave generator

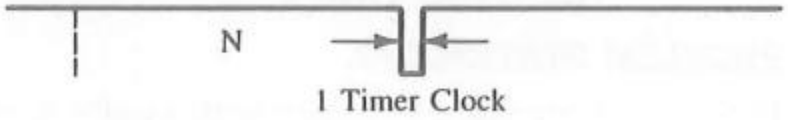
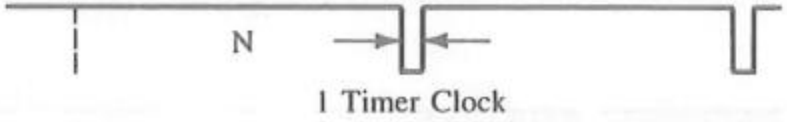
Loading Format

M_2	M_1	T_{13}	T_{12}	T_{11}	T_{10}	T_9	T_8

MSB

T_7	T_6	T_5	T_4	T_3	T_2	T_1	T_0

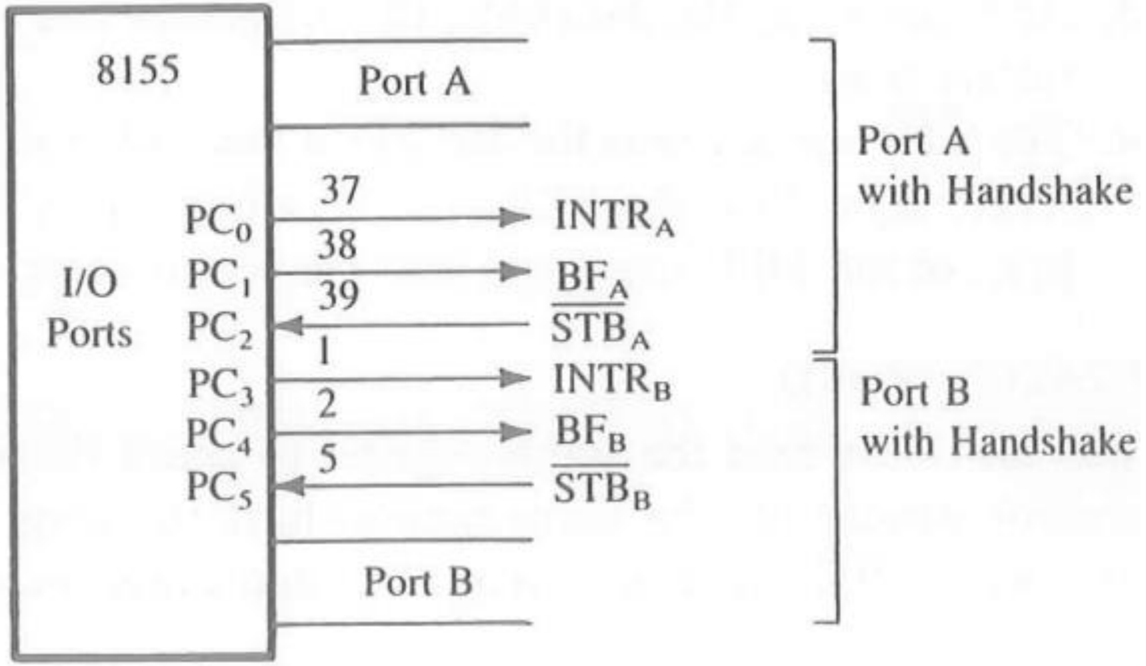
LSB

Modes	M_2 M_1	Timer Output
Mode 0: In this mode, the timer output remains high for half the count and goes low for the remaining count, thus providing a single square wave. The pulse width is determined by the count and the clock frequency.	0 0	Single Square Wave Cycle 
Mode 1: In this mode, the initial timer count is automatically reloaded at the end of each count, thus providing a continuous square wave.	0 1	Square Wave 
Mode 2: In this mode, a single clock pulse is provided at the end of the count.	1 0	Single Pulse Upon Terminal Count 
Mode 3: This is similar to Mode 2, except the initial count is reloaded to provide a continuous wave form.	1 1	Pulse Every Terminal Count 

N = Count in Timer

- Page440(Gaonkar)

8155 ports in Handshake mode



- Page 441 (Gaonkar)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY RANCHI

भारतीय सूचना प्रौद्योगिकी संस्थान राँची

SINCE 2016

|| उद्यमेन हि सिध्यन्ति कार्याणि ||