

IC 82C55

The 82C55 **programmable peripheral interface (PPI)** is a very popular low-cost interfacing component that is used in many applications.

- It is a **40 pin IC/PPI**
- It has **3 ports - Port A, Port B, Port C - to connect I/O devices**
- Each **port has 8 pins** for data transmission from any I/O device
- 8 pins for control words/ data/ instruction transmission - MPU (Microprocessor Unit) to IC
- **V_{cc}**: Powers the IC
- **GND**: prevents overpower/ balances the power
- **CS = 0** : Activates the IC
- **RD = 0**: MPU Reads from the IC buffer
- **WR = 0**: MPU writes to the IC buffer
- **A1, A0** : Selects port

A1	A0	PORT
0	0	PORT A
0	1	PORT B
1	0	PORT C
1	1	Command Register

- **RESET**: Reconfigures the IC

Pin Diagram:

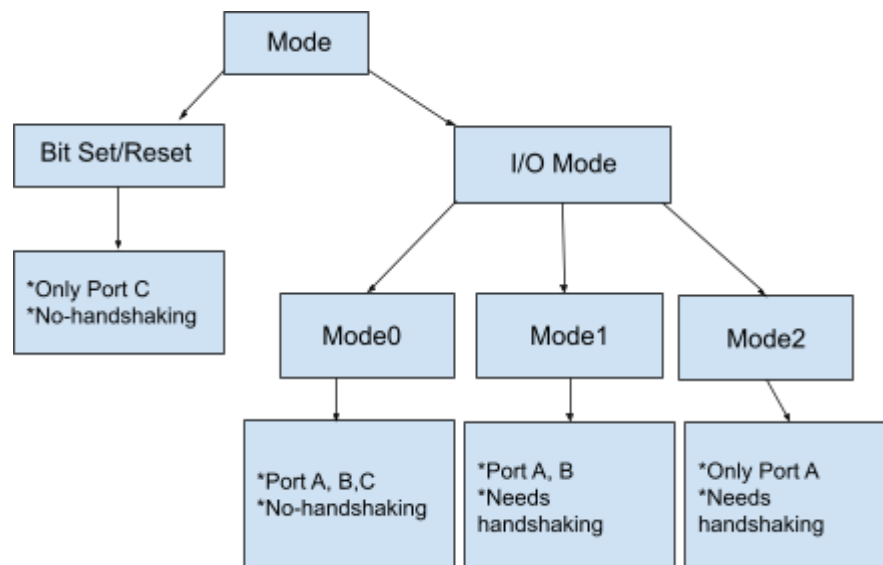
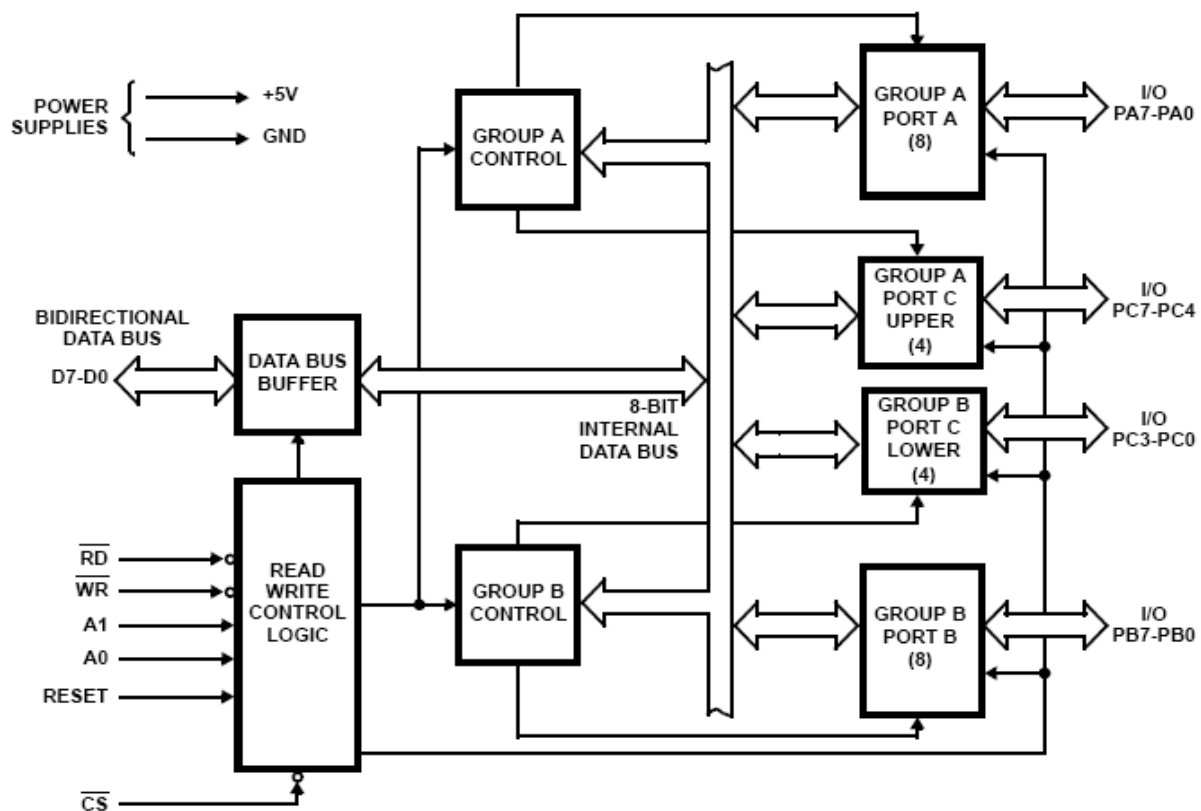
PA3	1		40	PA4
PA2	2		39	PA5
PA1	3		38	PA6
PA0	4		37	PA7
RD	5		36	WR
CS	6		35	RESET
gnd	7		34	D0
A1	8		33	D1
A0	9		32	D2
PC7	10	8255	31	D3
PC6	11	PPI	30	D4
PC5	12		29	D5
PC4	13		28	D6
PC0	14		27	D7
PC1	15		26	Vcc
PC2	16		25	PB7
PC3	17		24	PB6
PB0	18		23	PB5
PB1	19		22	PB4
PB2	20		21	PB3

- PA0-PA7: It is the 8-bit bi-directional I/O pins used to send the data to the peripheral or to receive the data from the peripheral.
- PB0-PB7: Similar to PA
- PC0-PC7: This is also 8-bit bidirectional I/O pins. These lines are divided into two groups.
PC0 to PC3 (Lower Groups)
PC4 to PC7 (Higher groups)

**** DATA BUS:** It is a 8-bit bidirectional Data bus. Used to interface between 82C55 data bus and system bus. The internal data bus and Outer pins D0-D7 pins are connected internally. The direction of the data buffer is decided by Read/Control Logic.

Following Table gives the basic operation,

A ₁	A ₀	<u>RD</u>	<u>WR</u>	<u>CS</u>	Input operation
0	0	0	1	0	PORT A → Data bus
0	1	0	1	0	PORT B → Data bus
1	0	0	1	0	PORT C → Data bus
<u>Output operation</u>					
0	0	1	0	0	Data bus → PORT A
0	1	1	0	0	Data bus → PORT B
1	0	1	0	0	Data bus → PORT C
1	1	1	0	0	Data bus → control



Mode0 - Only low level devices - LED, Buzzer, Switch

Mode1 - High Level Unidirectional Devices - Mouse, Keyboard, Printer

Mode2 - High Level Bi-directional Device - Hard drive, Touchscreen

****Mode is decided by the 8 bit "control bits" sent from the MPU to the IC****

Control bits:

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

D7: 0 -> Bit Set/Reset Mode

D7: 1 -> I/O Mode

Bit Set/Reset Mode

We use bit set/reset mode when the MPU needs to set/reset or configure/reconfigure any device which is only connected to port C.

Control bits:

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

1. D7 : 0 - Defines how the IC will configure itself in bit set reset mode
2. D6, D5, D4: X, X, X (DON'T CARE)
3. D3, D2, D1: Selects from which pin of port C the actual data will be transmitted

D3	D2		D1	Port C pin
0	0		0	0
0	0		1	1
0	1		0	2
0	1		1	3
1	0		0	4
1	0		1	5
1	1		0	6
1	1		1	7

4. D0: Sends the actual data to the IC

D0 -> 1 - Means Set the device

D0 -> 0 - Means Reset the device

I/O MODE

We use I/O mode when different types of I/O devices are connected to the IC and want to communicate with the microprocessor.

Control bits:

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

1. **D7: 1** - Defines the IC will configure itself in bit I/O mode
2. **D6,D5:** Selects the **mode** in which **Port A** will work
00 - MODE0
01 - MODE1
10 - MODE2
3. **D4:** defines what **type of device** (Input device / Output device) is connected to **port A**
D4 -> **1** - INPUT DEVICE
D4 -> **0** - OUTPUT DEVICE
4. **D3:** Defines what type of device (**Input device / Output device**) is connected with the upper **4 pins (PC7-PC4)** of port C
D3 -> 1 - INPUT DEVICE
D3 -> 0 - OUTPUT DEVICE
D3 -> **X** - All 4 pins are **busy** in handshaking
5. **D2:** Selects the **mode** in which **Port B** will work
0 - MODE0
1 - MODE1
6. **D1:** defines what **type of device** (Input device / Output device) is connected to port B
D1 -> 1 - INPUT DEVICE
D1 -> 0 - OUTPUT DEVICE
7. **D0:** Defines what **type of device** (Input device / Output device) is connected with the **lower 4 pins (PC0-PC3)** of port C
D0 -> 1 - INPUT DEVICE
D0 -> 0 - OUTPUT DEVICE
D0 -> X - All 4 pins are busy is handshaking

Example 1 : In port A, 1 printer is connected and in port B and C 4 LEDs are connected. Write the control bits.

1	0	1	0	x	0	0	0
---	---	---	---	---	---	---	---

Example : In port A, 1 printer is connected and in port B 1 keyboard is connected. Write the control bits.

1	0	1	0	x	1	1	x
---	---	---	---	---	---	---	---

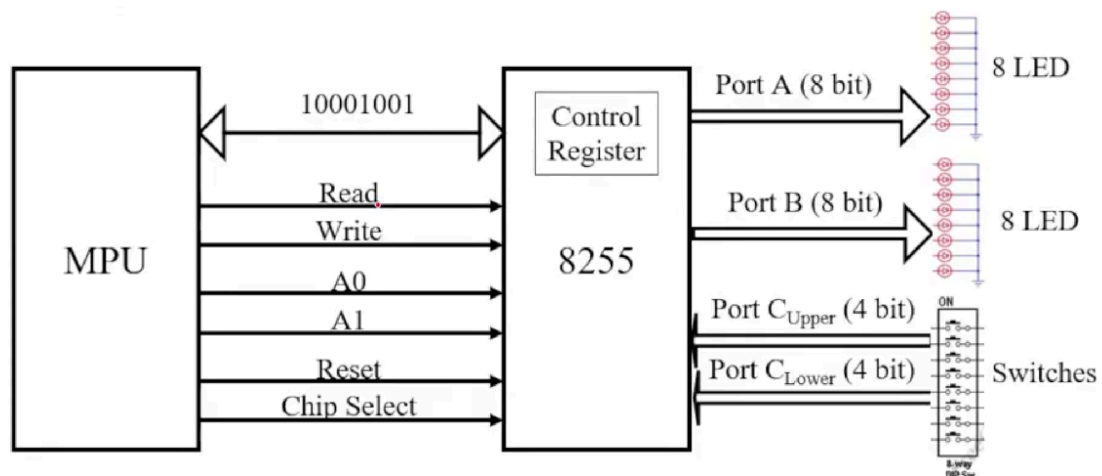
Example 3: In port B, 1 keyboard is connected and in port A 5 switches are connected and in port C 3 LEDs are connected. Write the control bits.

1	0	0	1	0	1	1	x
---	---	---	---	---	---	---	---

Example 4: In port A, 1 pendrive is connected and in port B 4 switches are connected. Write the control bits.

1	1	x	x	x	0	1	x
---	---	---	---	---	---	---	---

Mode0 - Communication (No Handshaking)



Mode0 works for only low level devices

outputs are - Latched [1 bit], inputs are - Buffered [8 bit]

Steps:

1. MPU selects the IC/ activates the IC with **CS = 0**
2. **A1, A0 = 1,1** and activates the **command registers**/ prepares the IC to configure
3. **WR = 0**
4. Sends **8 bit control** bits through the databus [Here, 10001001]
5. Once IC is configured further data is transmitted through the data bus line

Handshaking

Handshaking is a **I/O control** method to synchronize I/O devices with the microprocessor

Handshaking

Mode0 - No Handshaking

Mode1 -

- Input handshaking
- Output handshaking

Mode2 -

- Input handshaking
- Output handshaking

Mode1 - Handshaking

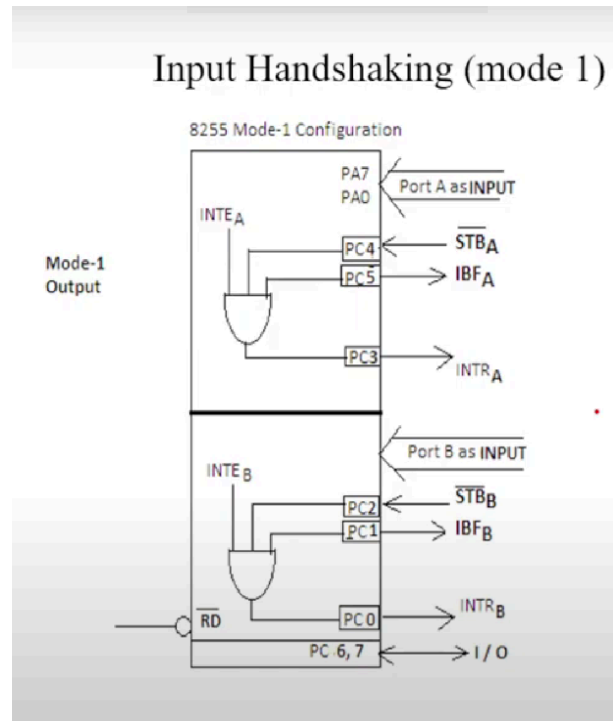
Input Handshaking:

Steps (PORT A):

1. Device sends 8 bit data to data buffer of the IC through PA0-PA7
2. Immediately the strobe signal ($\overline{STB_A} = 0$) is sent from device to IC through PC4 to let the IC know that data is being sent
3. Once the data bus buffer is filled with data the input buffer full ($IBF_A = 1$) signal is sent from IC to device through PC5
4. $\overline{STB_A} = 1$
5. $INTR_A = 1$ is sent from PC3 from IC to MPU for receiving the data
6. If MPU wants, $\overline{RD} = 0$ and reads the data
7. $INTR_A = 0$, $IBF_A = 0$

Steps (PORT B):

1. Device sends 8 bit data to data buffer of the IC through PB0-PB7
2. Immediately the strobe signal ($\overline{STB_B} = 0$) is sent from device to IC through PC2 to let the IC know that data is being sent
3. Once the data bus buffer is filled with data the input buffer full ($IBF_B = 1$) signal is sent from IC to device through PC1
4. $\overline{STB_B} = 1$
5. $INTR_B = 1$ is sent from



<p>PC0 from IC to MPU for receiving the data</p> <p>6. If MPU wants, $\overline{RD} = 0$ and reads the data</p> <p>7. $INTR_B = 0$, $IBF_B = 0$</p>	
--	--

Timing diagram of input handshaking (Mode 1):

<p>STB The strobe input loads data into the port latch on a 0-to-1 transition</p> <p>IBF Input buffer full is an output indicating that the input latch contain information</p> <p>INTR Interrupt request is an output that requests an interrupt</p> <p>INTE The interrupt enable signal is neither an input nor an output; it is an internal bit programmed via the PC4(port A) or PC2(port B) bits.</p> <p>PC7,PC6 The port C pins 7 and 6 are general-purpose I/O pins that are available for any purpose.</p>	<p>The diagram illustrates the input handshaking for Mode 1. On the left, two logic blocks are shown: 'Mode 1 Port A' and 'Mode 1 Port B'. Port A has inputs INTE A, PC4, PC5, PC3, and PC6+7. Port B has inputs INTE B, PC2, PC1, and PC0. On the right, the 'Timing Diagram' shows the sequence of events: 1. STB (red) transitions from 0 to 1, which is labeled 'Data strobed into port'. 2. IBF (blue) transitions from 0 to 1, labeled '(Buffer full)'. 3. INTR (blue) transitions from 0 to 1, labeled '(Interrupt request)'. 4. RD (red) transitions from 1 to 0, labeled 'Data read by microprocessor'. 5. The Port data (red) is shown as a bus that changes when STB goes high and is read when RD goes low.</p>
--	---

Output Handshaking:

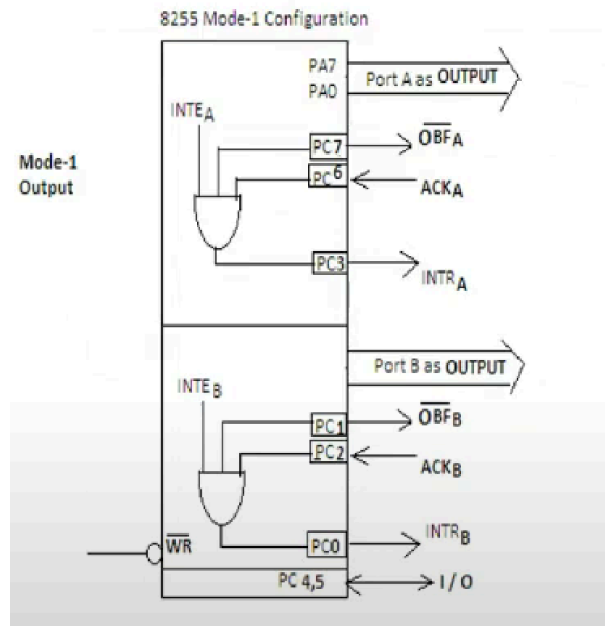
Steps (PORT A):

1. $\text{INTR}_A = 1$ is sent from PC3 from the device via IC to the MPU to check if there is any data to output.
2. If there is, MPU $\rightarrow \text{WR} = 0$ and sends data to the data bus buffer of the IC
3. IC sends the MPU output buffer full signal ($\text{OBF}_A = 0$) through PC7 to tell the device to take data
4. $\text{INTR}_A = 0$
5. Once the device is ready it sends $\text{ACK}_A = 0$ to the IC through PC6
6. The device then takes data from the data bus buffer through PA0-PA7 for providing output
7. Once all data is taken, $\text{OBF}_A = 1$, $\text{INTR}_A = 1$

Steps (PORT B):

1. $\text{INTR}_B = 1$ is sent from PC0 from the device via IC to the MPU to check if there is any data to output
2. If there is, MPU $\rightarrow \text{WR} = 0$ and sends data to the data bus buffer of the IC
3. IC sends the MPU output buffer full signal ($\text{OBF}_B = 0$) through PC1 to tell the device to take data
4. $\text{INTR}_B = 0$
5. Once the device is ready it sends $\text{ACK}_B = 0$ to the IC through PC2

Output Handshaking (mode 1)



*
-

ACK will be an active low pin in the image

6. The device then takes data from the data bus buffer through PB0-PB7 for providing output 7. Once all data is taken, <u>OBF_B</u> = 1, <u>INTR_B</u> = 1	
---	--

Mode 1 - input handshaking - free pins/ available to connect low level devices to Port C - Pin 6,7

Mode 1 - output handshaking - free pins/ available to connect low level devices to Port C - Pin 4,5

Example 1: Port A/ Port B is connected to an Input device. Write the handshaking steps.

Ans. same as described above

Example 2: Port A/ Port B is connected to an Output device. Write the handshaking steps.

Ans. same as described above

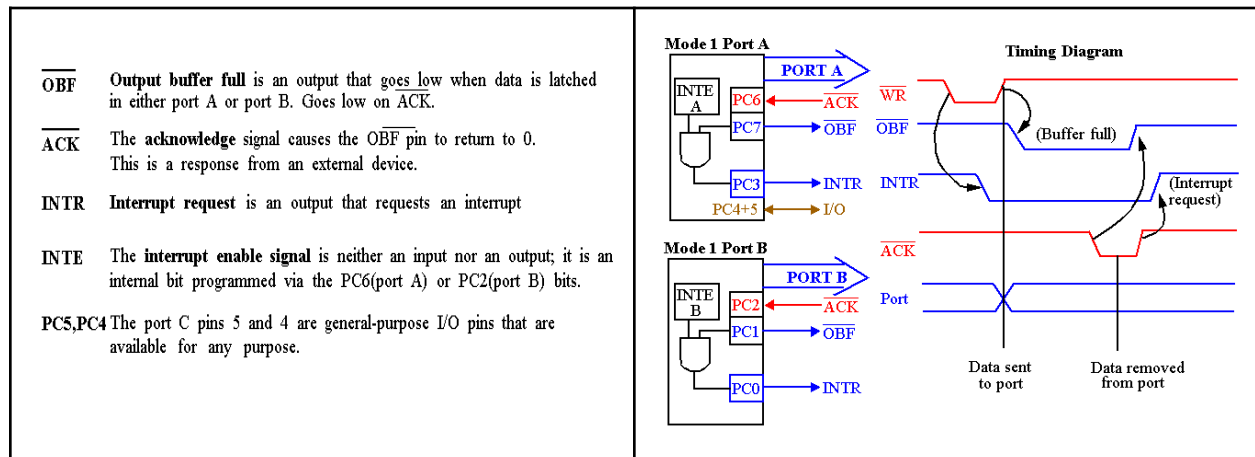
Example 3: Port A is connected to an input device and Port B is connected to an output device. Write the handshaking steps.

Ans. In this case MPU removes the redundant INTR pin usage. And for both cases the INTR signal goes from PC3 of Port C. Rest of the process is the same. In this case, Pin PC0, PC6, PC7 are free to connect low level devices.

Example 4: Port B is connected to an input device and Port A is connected to an output device. Write the handshaking steps.

Ans. In this case MPU removes the redundant INTR pin usage. And for both cases the INTR signal goes from PC3 of Port C. Rest of the process is the same. In this case, Pin PC0, PC4, PC5 are free to connect low level devices.

Timing diagram of output handshaking (Mode 1):



Mode 2 - Handshaking

In mode 2, only bi-directional I/O devices can be connected. In this case the MPU hands over the control to the 82C55 IC right after only sending the control bits to the IC. With each clock pulse the IC configures itself either in input mode or output mode. When the device inputs any data the IC processes it during the input clock pulse and when the device needs to output any data the IC processes it during output clock pulse.

Input & Output handshaking:

Case 1: when port A is connected to a High-level bidirectional device and port B and port C is connected to low level device

Control Bits:

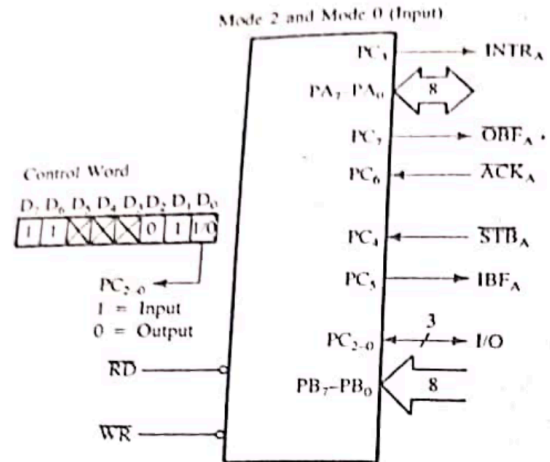
1	1	x	x	x	0	1/0	1/0
D7	D6	D5	D4	D3	D2	D1	D0

D1 - 1 - for input device
0 - for output device

D0 - 1 - for input device
0 - for output device

D3 - Fully dedicated to port A for handshaking

Steps: The Signals work just as mode 1 during the Input/ Output clock pulse of the 82C55 IC with different pin configuration.
Explain by yourself



Case 2: when port A is connected to a High-level bidirectional device and port B is connected to high-level unidirectional device

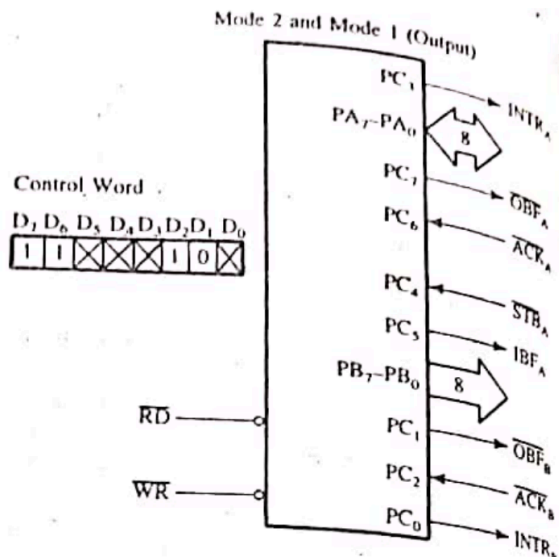
Control Bits:

1	1	x	x	x	1	1/0	x
D7	D6	D5	D4	D3	D2	D1	D0

D1 - 1 - for input device
0 - for output device

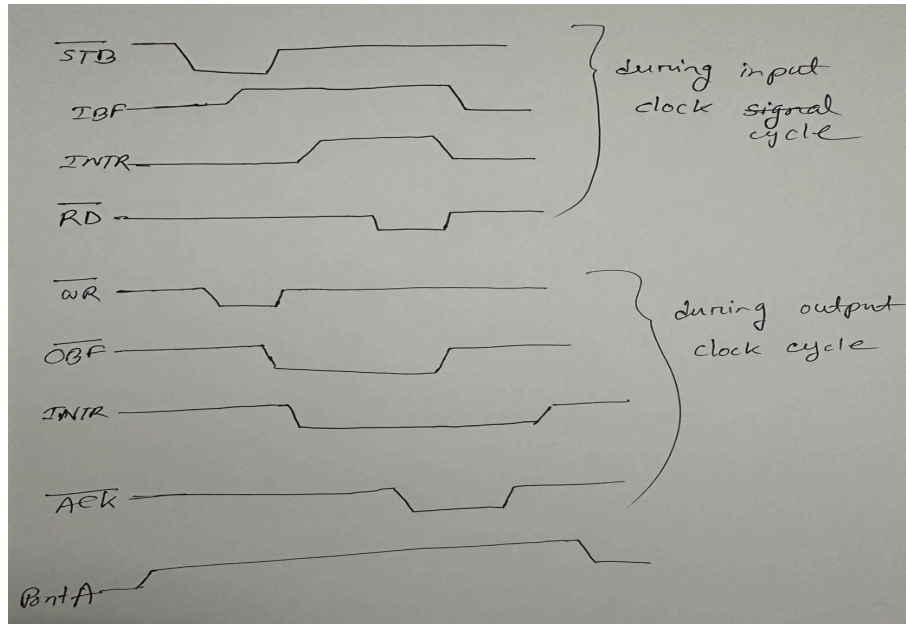
D3 & D0 - Fully dedicated to port A and Port B for handshaking

Steps: The Signals work just as mode 1 during the Input/ Output clock pulse of the 82C55 IC with different pin configuration.

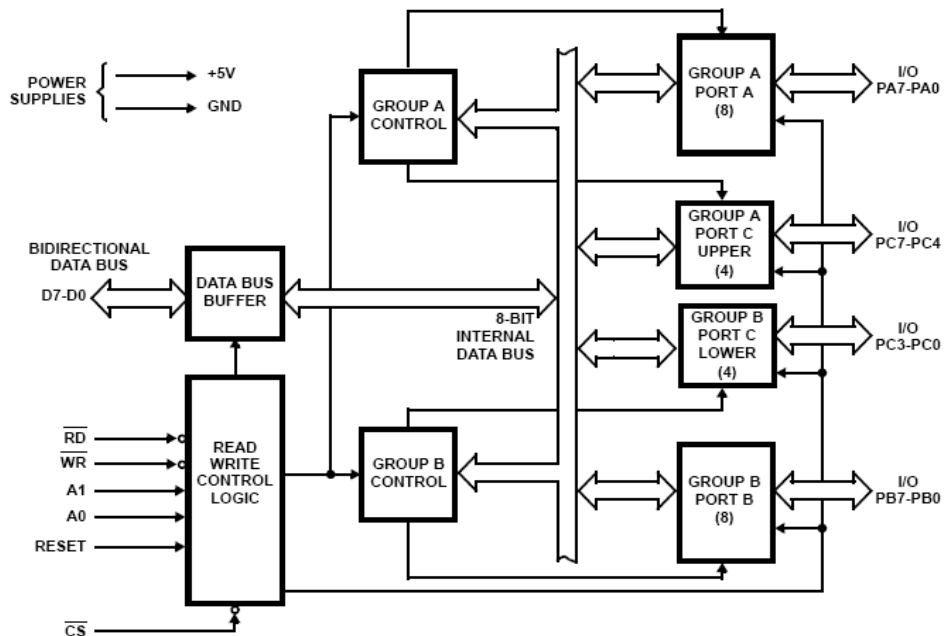


Explain by yourself

Timing diagram of handshaking (Mode 2):



How MPU and 82C55 communicates



Prepared by - Umme Jannat Taposhi (UJT)