MODULE 4

8255 PPI

Programmable Input-Output Port (PIO)Intel 8255

- Parallel data transfer between processor and slow devices
- Designed for used with 8-bit,16-bit and higher capacity MP's
- Consists of three 8-bit bidirectional I/O ports - Port A, Port B, and Port C
- It provides 3 modes of data transfer: Simple I/O, handshake I/O, and Bi-directional Handshake
- Additionally it has Bit Set Reset mode to alter individual pins of Port C

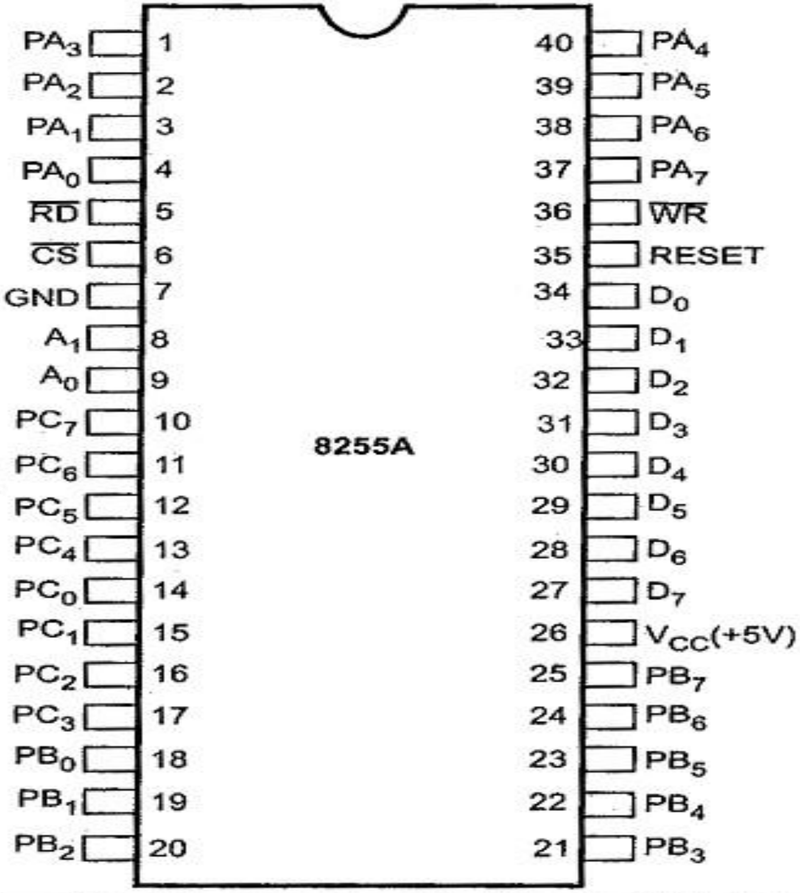
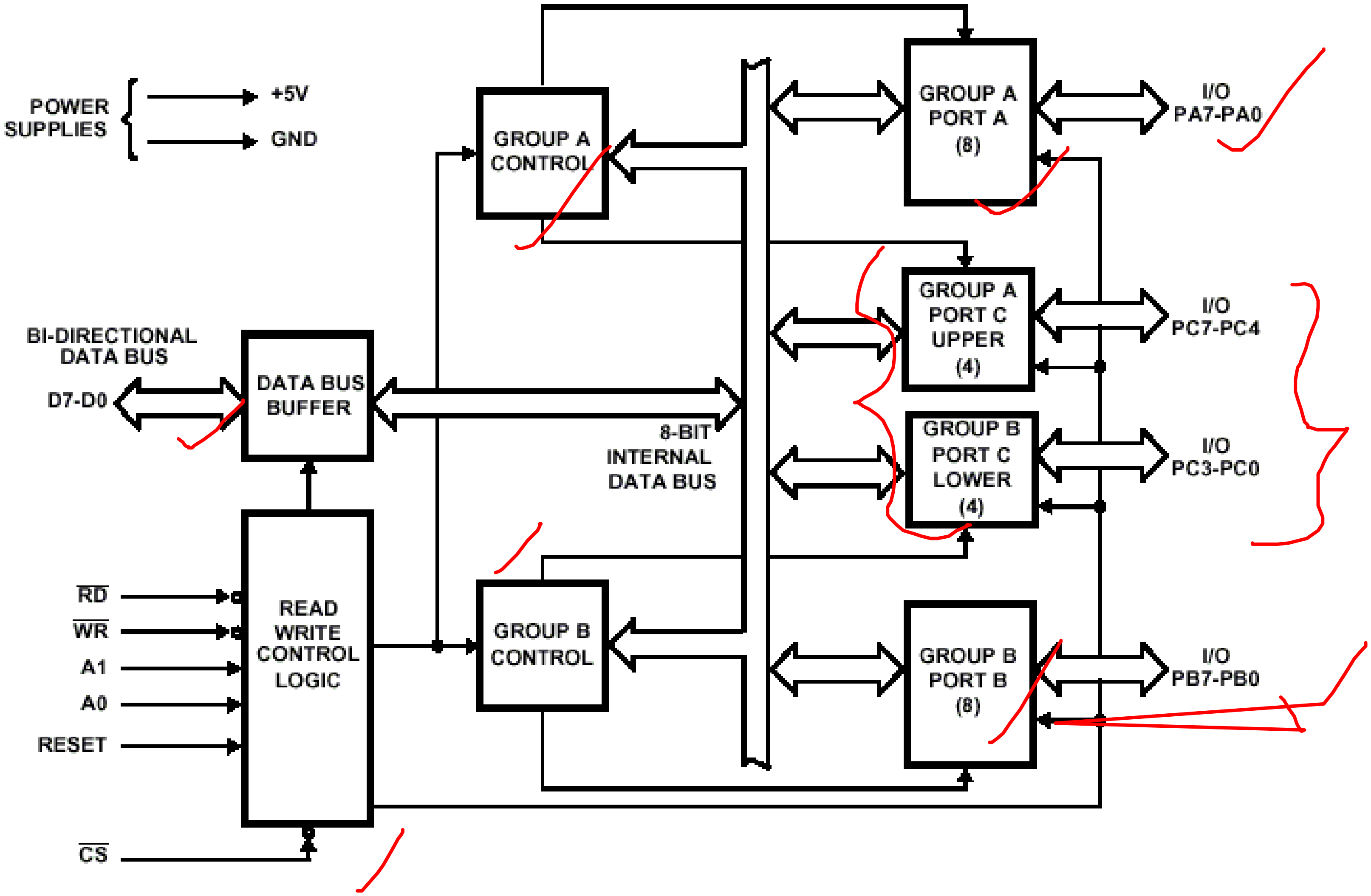


Fig. 14.1 Pin diagram of 8255A

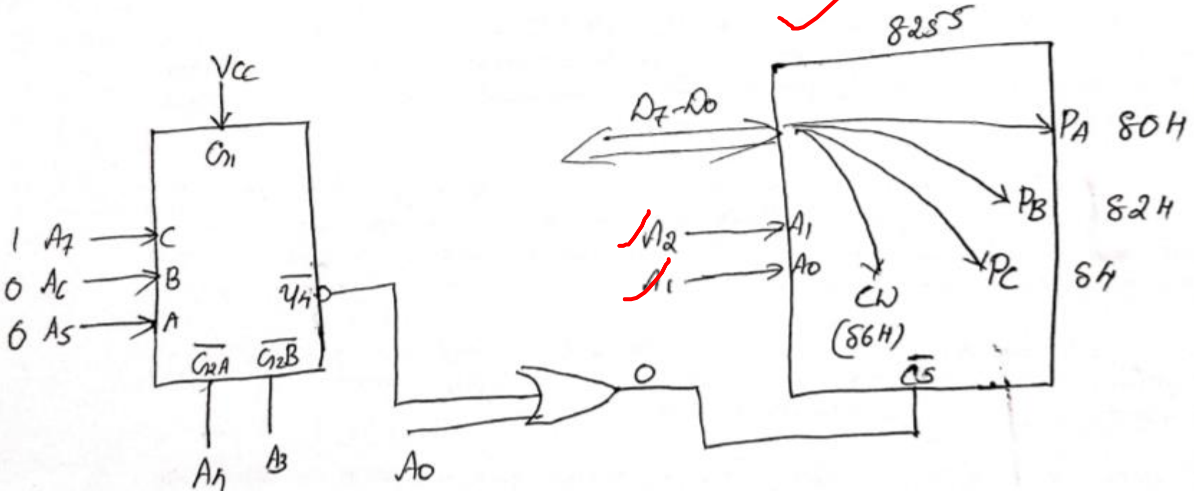
Internal Block Diagram of 8255



	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
P_A	1	0	0	0	0	0	0	0	80 H
P_B	1	0	0	0	0	0	1	0	82 H
P_C	1	0	0	0	0	1	0	0	84 H
CW	1	0	0	0	0	1	1	0	86 H

$\overline{CS}$ ← → Inter selection bank.

86



Blocks of 8255 architecture

Architecture can be divided in to

1)Data bus buffer

- 8-bit bidirectional buffer used to interface the internal data bus of 8255 with external (system)data bus
- MP transfer data to and from the 8255 through this buffer

2) Read/Write control logic

- It accepts address & control signals from the MP
- Control signals determine whether it is a read/write operation and also select or reset the 8255 chip
- Address bus (A1,A0) are used to select the Ports or Control word register as shown

For 8255	For 8086	Selection	Sample address
A_1 A_0	A_2 A_1		
0 0	0 0	Port A	80 H (i.e. 1000 0**00**0)
0 1	0 1	Port B	82 H (i.e. 1000 0**01**0)
1 0	1 0	Port C	84 H (i.e. 1000 0**10**0)
1 1	1 1	Control Word	86 H (i.e. 1000 0**11**0)

3) **Group A control**

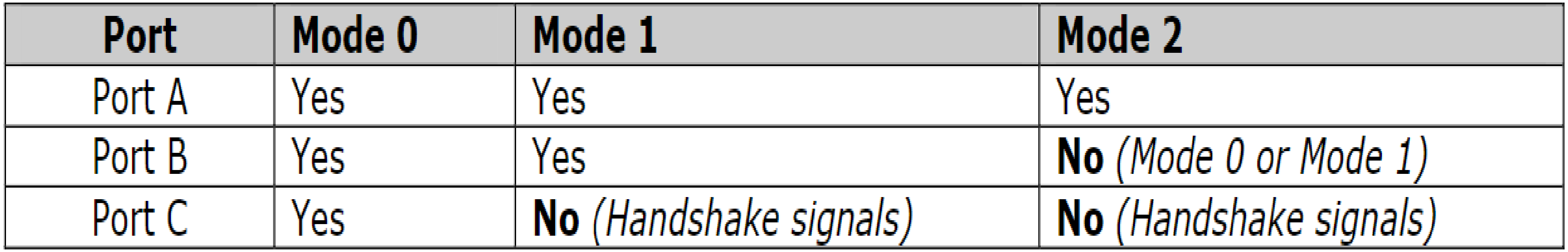
- This control block controls Port A and Port C_{upper} ie PC_7 –PC_4
- It accepts control signals from control word and forwards them to the respective ports

4)Group B control

- This control block controls Port B and Port C_{lower} ie PC_3 –PC_0
- It accepts control signals from control word and forwards them to the respective ports

5) Port A, Port B, Port C

- These are the 8-bit bi-directional ports
- They can be programmed to work in the various modes as follows



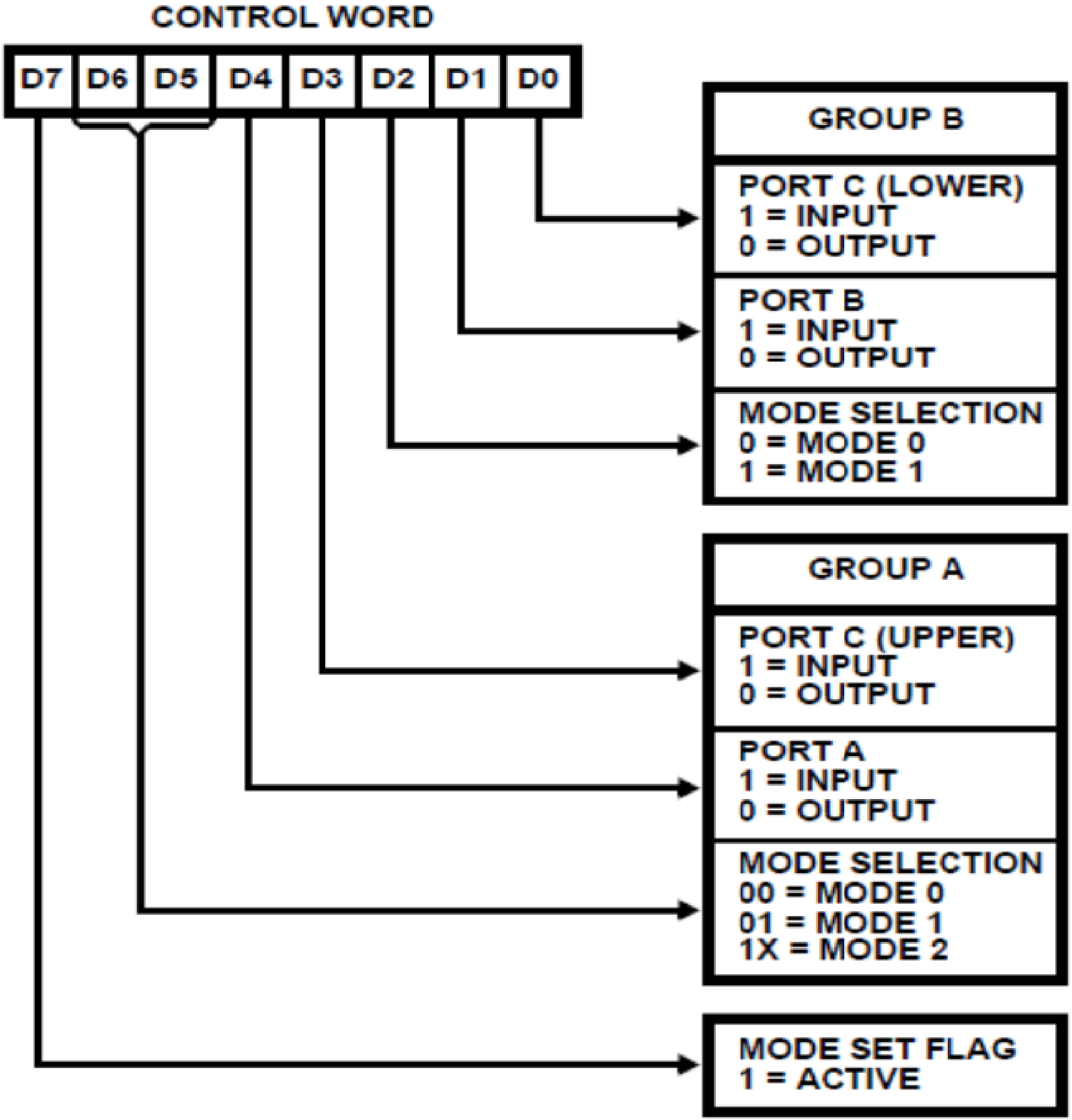
Port	Mode 0	Mode 1	Mode 2
Port A	Yes	Yes	Yes
Port B	Yes	Yes	**No** *(Mode 0 or Mode 1)*
Port C	Yes	**No** *(Handshake signals)*	**No** *(Handshake signals)*

- Only port C can be programmed to work in BSR mode to manipulate its individual bits.

1)Control word of 8255-I/O mode

- To do 8-bit data transfer using ports A,B or C, 8255 needs to be in the I/O mode
- The bit pattern for the control word in the I/0 mode is shown in the next slide

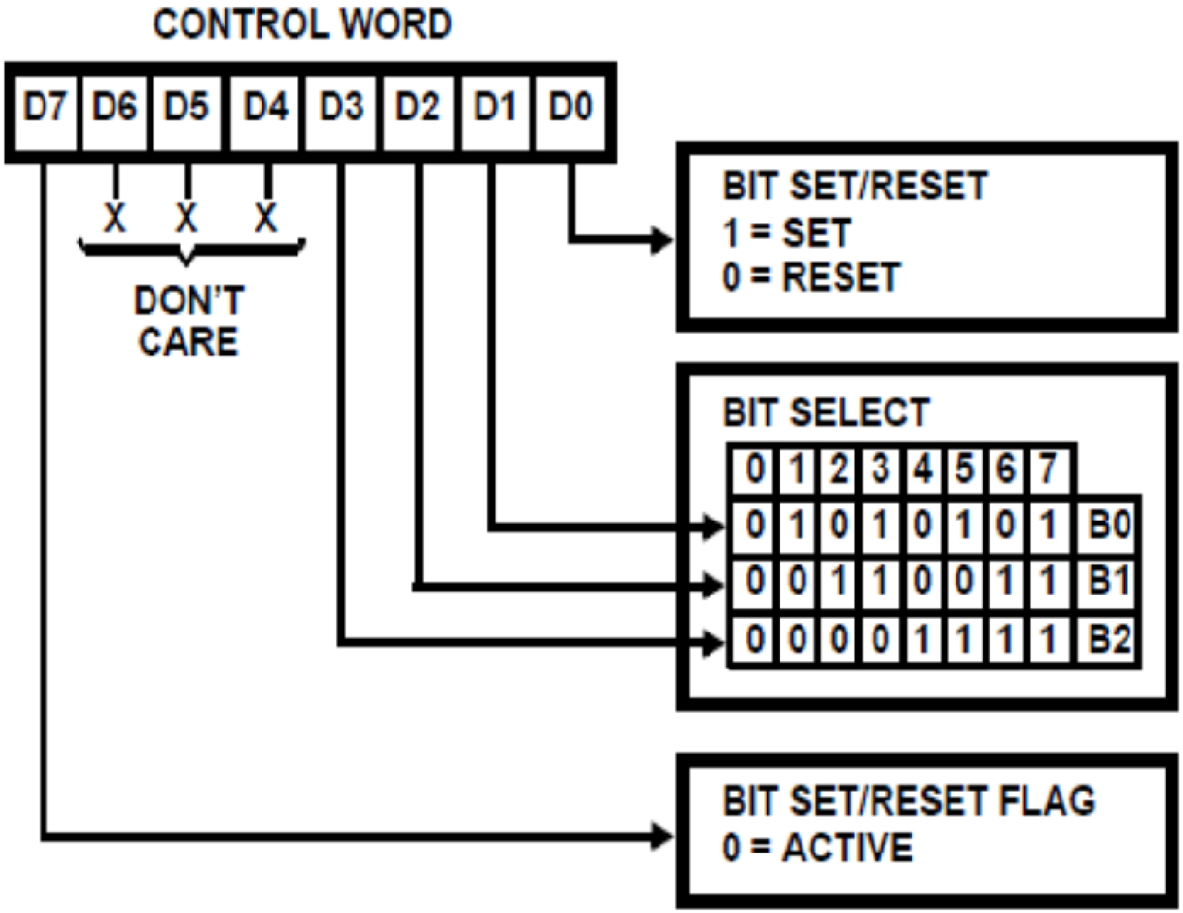
I/O mode control word register format



2)Control word of 8255-BSR mode

- This mode is only used for Port C
- Individual bits of Port C can be set/reset
- It is very useful as it provides 8 individually controllable lines
- BSR operation will not affect D_7 bit of BSR
- The bit pattern for the control word is shown in the next slide

BSR mode control word register format



0	1	2	3	4	5	6	7	
0	1	0	1	0	1	0	1	B0
0	0	1	1	0	0	1	1	B1
0	0	0	0	1	1	1	1	B2

Data Transfer modes of 8255

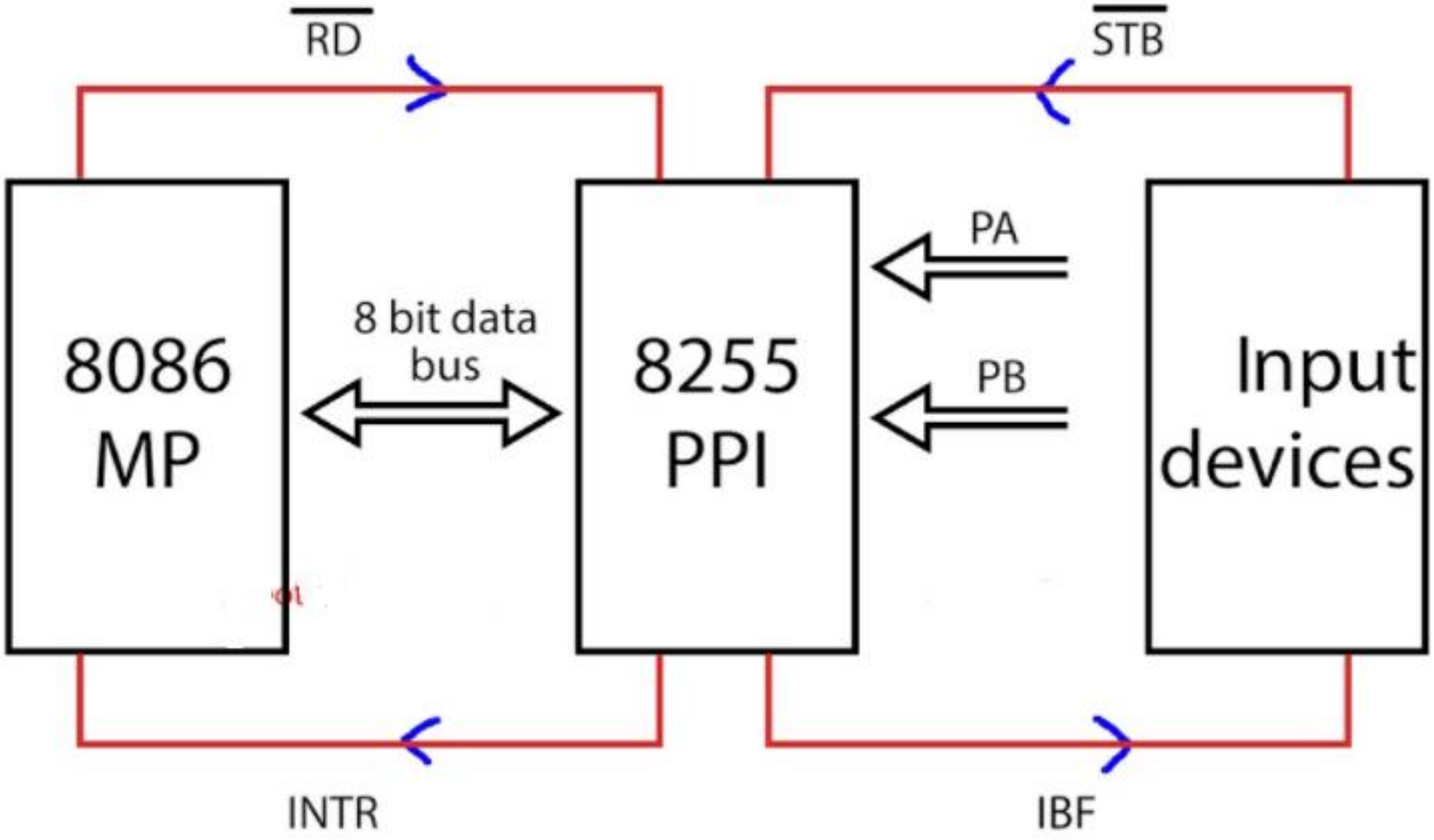
Mode 0(Simple I/O or Basic I/O)

- Port A, Port B used as 2 simple 8-bit I/O ports
- Port C is used as 2 simple 4-bit I/O ports
- Each port can be programmed individually as input or output port
- Ports do not have handshaking or interrupting capability
- Hence, slower devices cannot be interfaced

Mode 1 (Handshake I/O or Strobbed I/O)

- Handshake signals are exchanged between the devices before the data transfer takes place
- Port A and Port B are used as 2 8-bit ports that can be programmed in input or output mode
- Each port uses 3 lines from port C for handshake. The remaining lines are used for simple I/O
- Interrupt driven data transfer possible
- Hence, slower devices can be interfaced

Mode 1- Handshake I/O

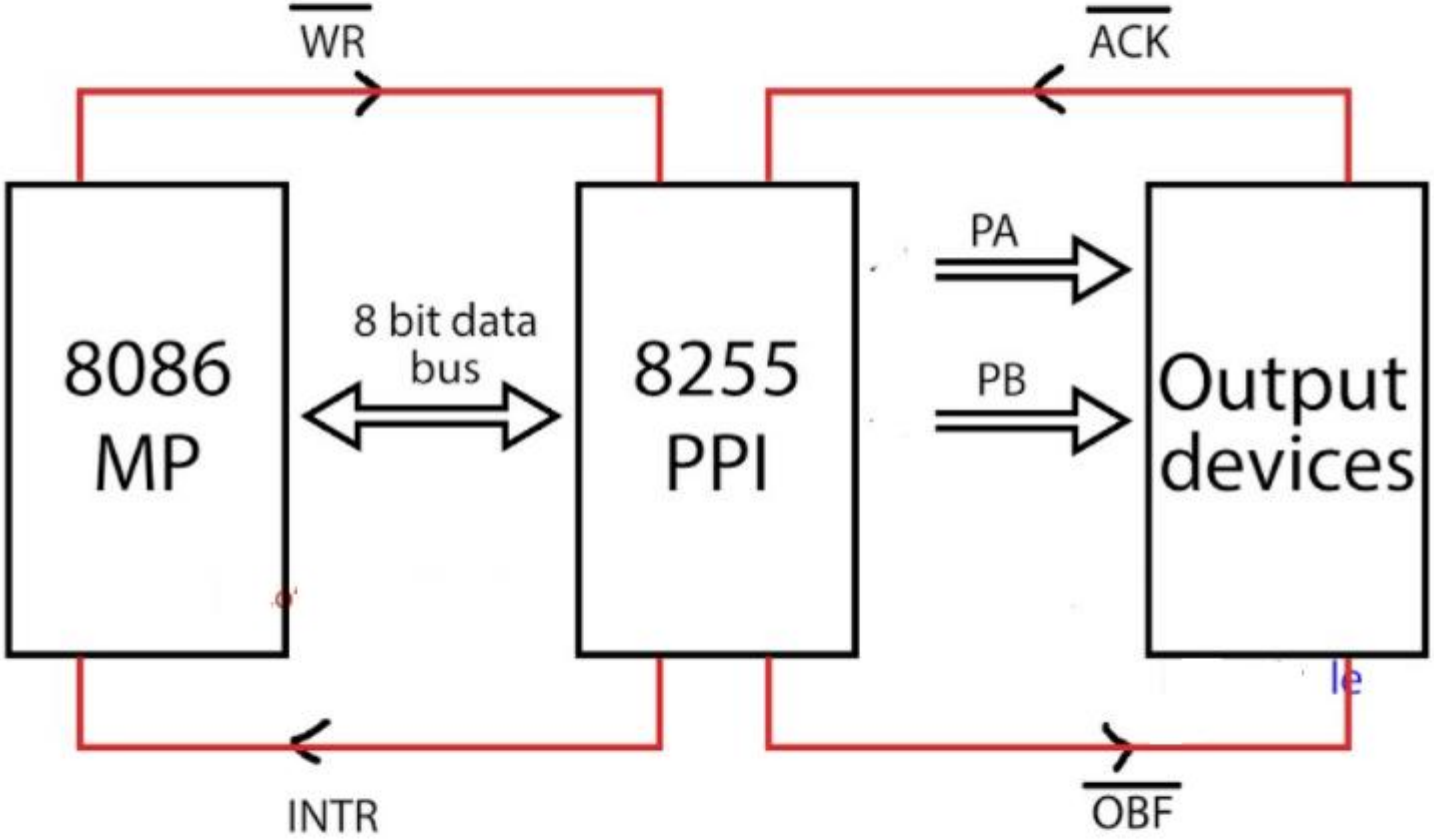


1. Input device gives STB signal (active low) to 8255 before sending data. If 8255 already has some data, then it doesn’t allow the Input device to send data.
2. 8255 accepts data and stores it in the input buffer register. As soon as it receives the data, it gives an IBF signal (input buffer is full) . If IBF = 1, it does not allow any other data to enter 8255 PPI.
3. Now 8255 has data that should be sent to 8086. It makes INTR high, which means it is interrupting the processor.
4. Whenever the processor is free, it services this INTR. Then the Processor sends an RD (active low) signal and makes data transfer via data bus.
5. As soon as the RD signal is given , INTR gets low.

Complete stepwise handshaking :

1. STB (active low) goes low.
2. IBF goes high.
3. INTR goes high.
4. RD (active low) goes low.
5. INTR goes low.
6. RD (active low) goes high.
7. IBF goes low.

Mode – 1 [Output Handshaking]



1. 8255 sends an interrupt whenever it is ready to accept data from the processor.
2. Whenever 8086 needs to send a signal (condition provided INTR is active), then 8086 sends WR (active low) signal. If 8255 has some data in it, then INTR will be low, and 8086 will not send data to 8255 PPI.
3. As 8255 receives data it resets OBF (output buffer is full (it is an active low signal).
4. Then the Output device acknowledges the received input.

Complete stepwise handshaking

1. WR signal is activated.
2. INTR is set to low. [now 8086 cannot send data again]
3. OBF goes low [Output buffer is full]
4. ACK goes low (it is an active low signal)
5. OBF goes back to high
6. ACK goes back to high
7. Data transfer is successful, so INTR goes back to high.

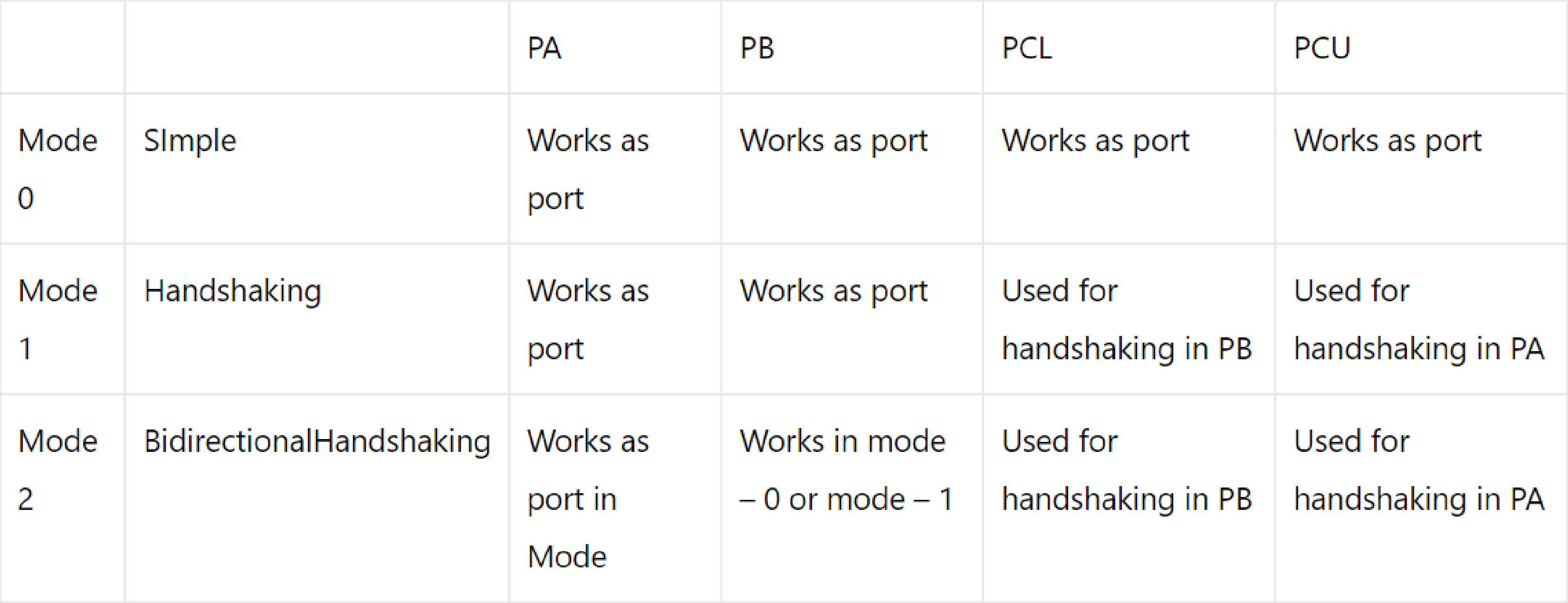
Mode 2 [Bidirectional Handshaking]

- In this mode, two-way handshaking can be done
- For normal input handshaking, we require 3 pins for handshaking
- Similarly 3 pins for output handshaking
- For bidirectional handshaking, we need 5 pins (because one INTR is enough)
- If PA is in mode 2, which means it is utilizing ⅝ pins of port C for handshaking.
- So there are 3 remaining pins of port C, so definitely port B cannot be operated in mode-2, but while port-A is in mode-2, port-B can be operated in Mode – 0 or mode – 1.

Mode 2 - Bi-directional Handshake

Input h/s (3)	Output h/s (3)
$\overline{STB}$	$\overline{OBF}$
IBF	$\overline{ACK}$
INTR	INTR

Total 5 lines needed (5 pin of PC)



		PA	PB	PCL	PCU
Mode 0	SImple	Works as port	Works as port	Works as port	Works as port
Mode 1	Handshaking	Works as port	Works as port	Used for handshaking in PB	Used for handshaking in PA
Mode 2	BidirectionalHandshaking	Works as port in Mode	Works in mode – 0 or mode – 1	Used for handshaking in PB	Used for handshaking in PA

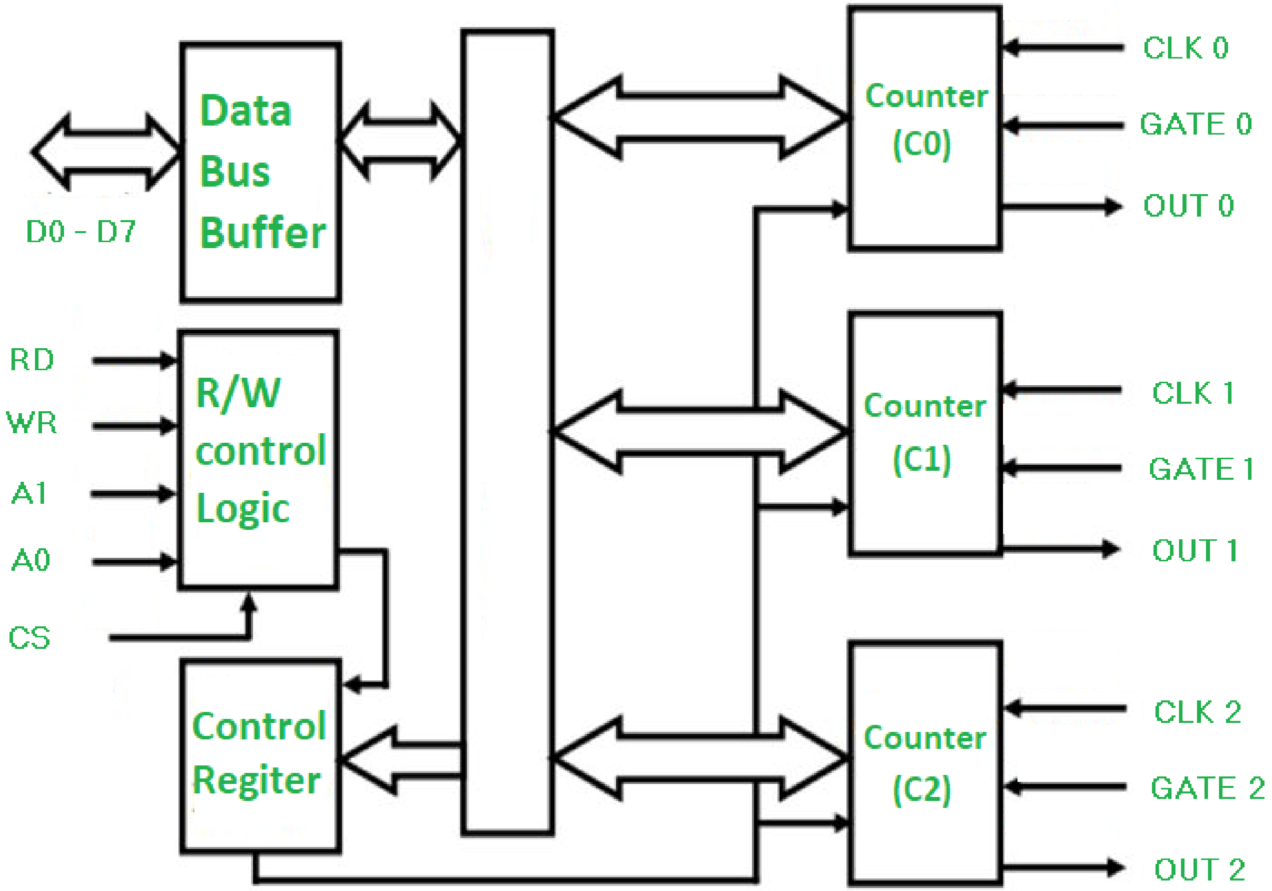
Module 4

Programmable interval timer 8254

8254 programmable interval timer

- 8254 is a device designed to solve the timing control problems in a microprocessor.
- It has 3 independent counters, each capable of handling clock inputs up to 10 MHz, and size of each counter is 16 bit.
- It operates in +5V regulated power supply and has 24 pin signals
- All modes are software programmable
- The 8254 is an advanced version of 8253 which did not offered the feature of read back command.

8254 Architecture

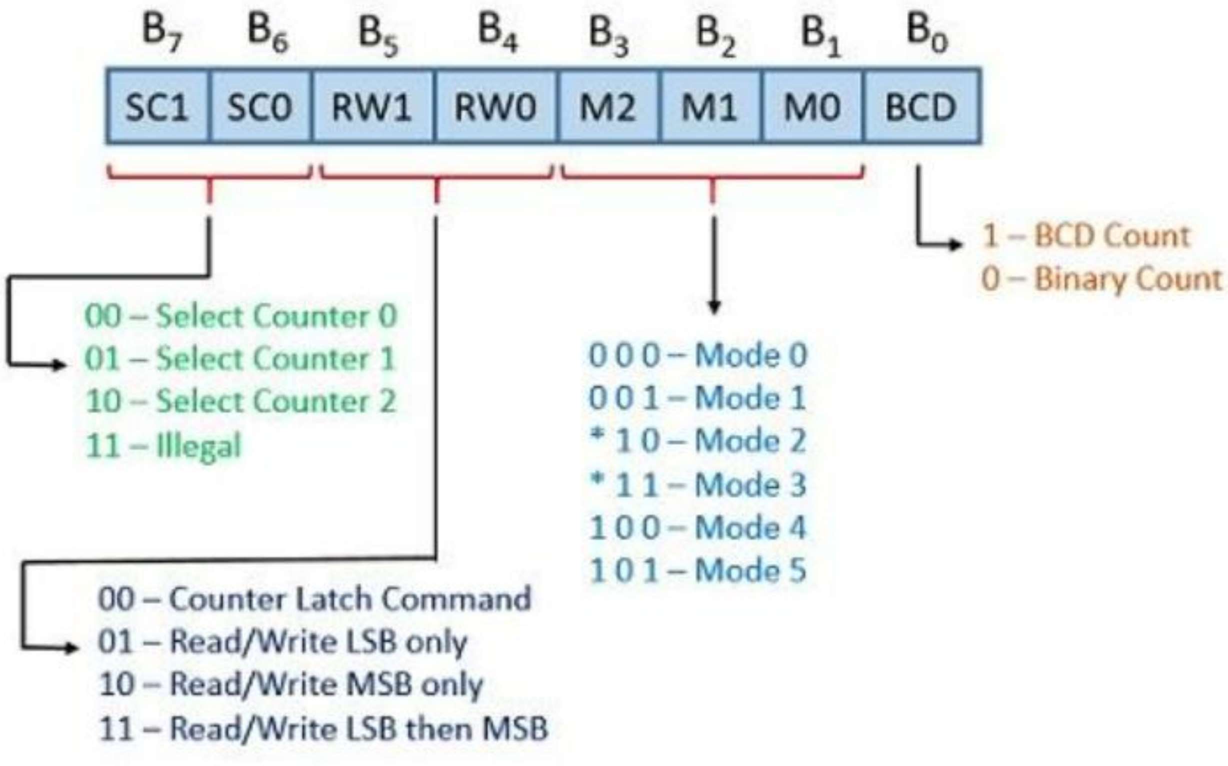


- **DATA BUS BUFFER**: This 3-state, bi-directional, 8-bit buffer is used to interface the 8254 to the system bus
- .**READ/WRITE LOGIC** : The Read/Write Logic accepts inputs from the system bus and generates control signals for the other functional blocks of the 8254.
- A1 and A 0 select one of the three counters or the Control Word Register to be read from/written into. •
- A “low” on the RD input tells the 8254 that the CPU is reading one of the counters

- A “low” on the WR input tells the 8254 that the CPU is writing either a Control Word or an initial count.
- Both RD and WR are qualified by CS; RD and WR are ignored unless the 8254 has been selected by holding CS low.
- **CONTROL WORD REGISTER** :The Control Word Register is selected by the Read/Write Logic when A1,A0 = 11.
- If the CPU then does a write operation to the 8254, the data is stored in the Control Word Register and is interpreted as a Control Word used to define the operation of the Counters.

- The Control Word Register can only be written to; status information is available with the Read-Back Command.
- COUNTER 0, COUNTER 1, COUNTER 2 :These three functional blocks are identical in operation, so only a single Counter will be described.
- The Counters are fully independent. Each Counter may operate in a different Mode. •
- The Control Word Register is not part of the Counter itself, but its contents determine how the Counter operates.

Programming of 8254



B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

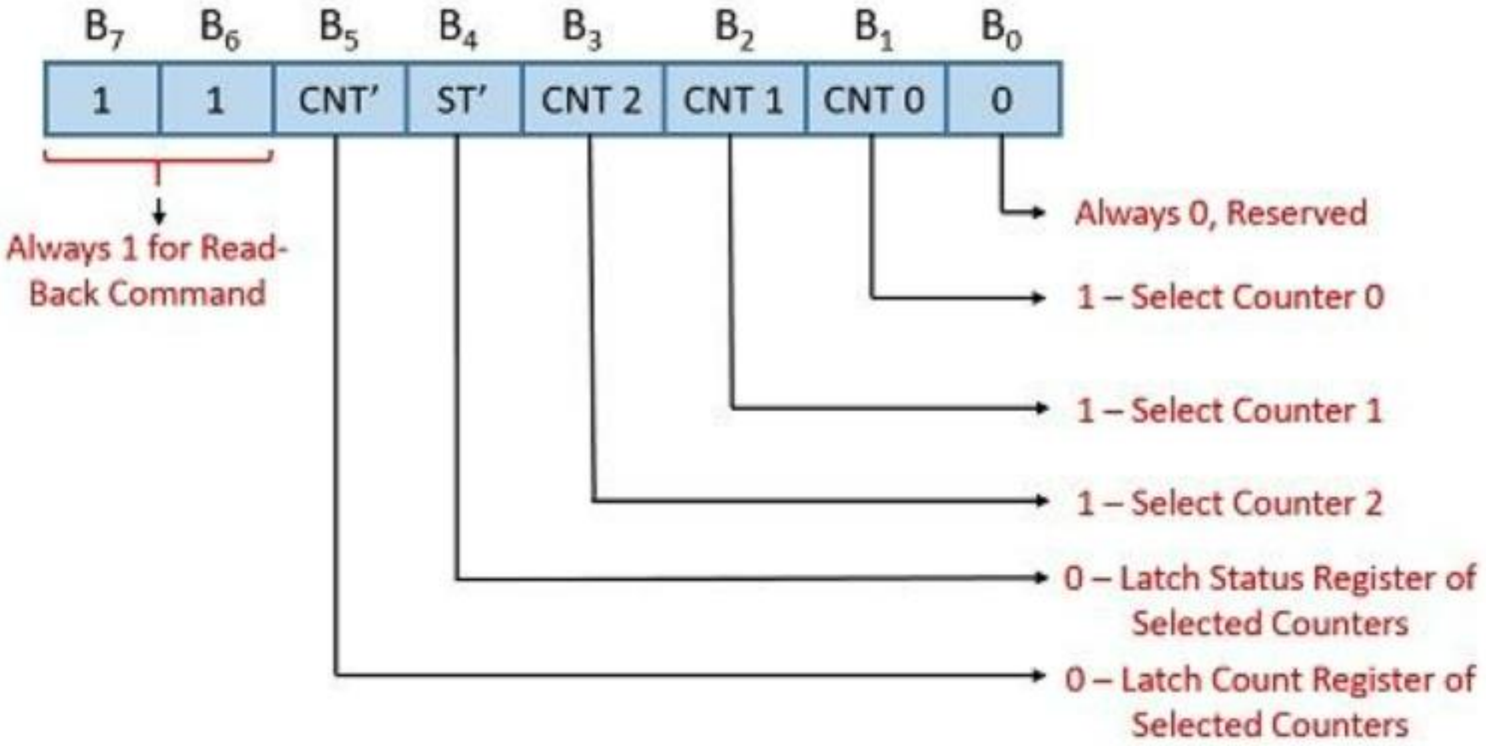
00 – Select Counter 0
01 – Select Counter 1
10 – Select Counter 2
11 – Illegal

00 – Counter Latch Command
01 – Read/Write LSB only
10 – Read/Write MSB only
11 – Read/Write LSB then MSB

0 0 0 – Mode 0
0 0 1 – Mode 1
* 1 0 – Mode 2
* 1 1 – Mode 3
1 0 0 – Mode 4
1 0 1 – Mode 5

1 – BCD Count
0 – Binary Count

Format of Control Word of 8254 Timer



B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
1	1	CNT’	ST’	CNT 2	CNT 1	CNT 0	0

Always 1 for Read-Back Command

Always 0, Reserved

1 – Select Counter 0

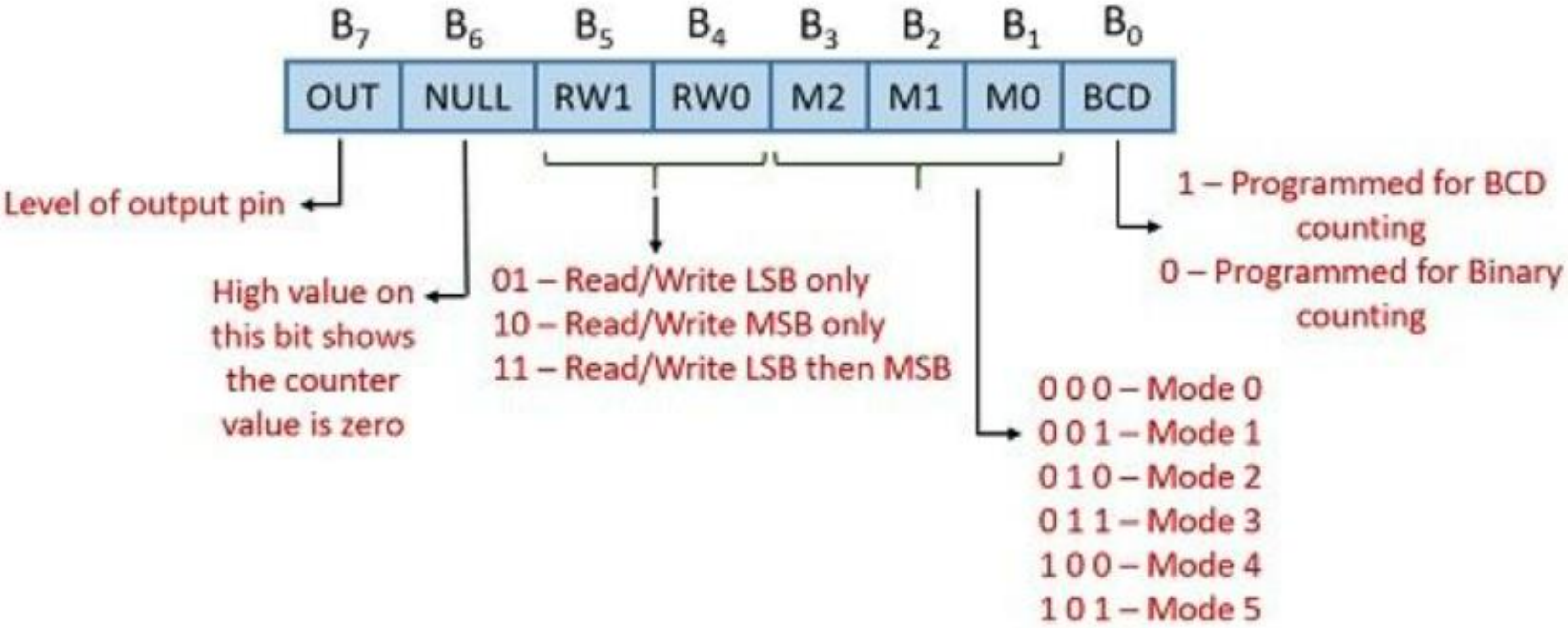
1 – Select Counter 1

1 – Select Counter 2

0 – Latch Status Register of Selected Counters

0 – Latch Count Register of Selected Counters

Format of Read-Back Control Word of 8254 Timer



B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
OUT	NULL	RW1	RW0	M2	M1	M0	BCD

Level of output pin

High value on this bit shows the counter value is zero

01 – Read/Write LSB only
10 – Read/Write MSB only
11 – Read/Write LSB then MSB

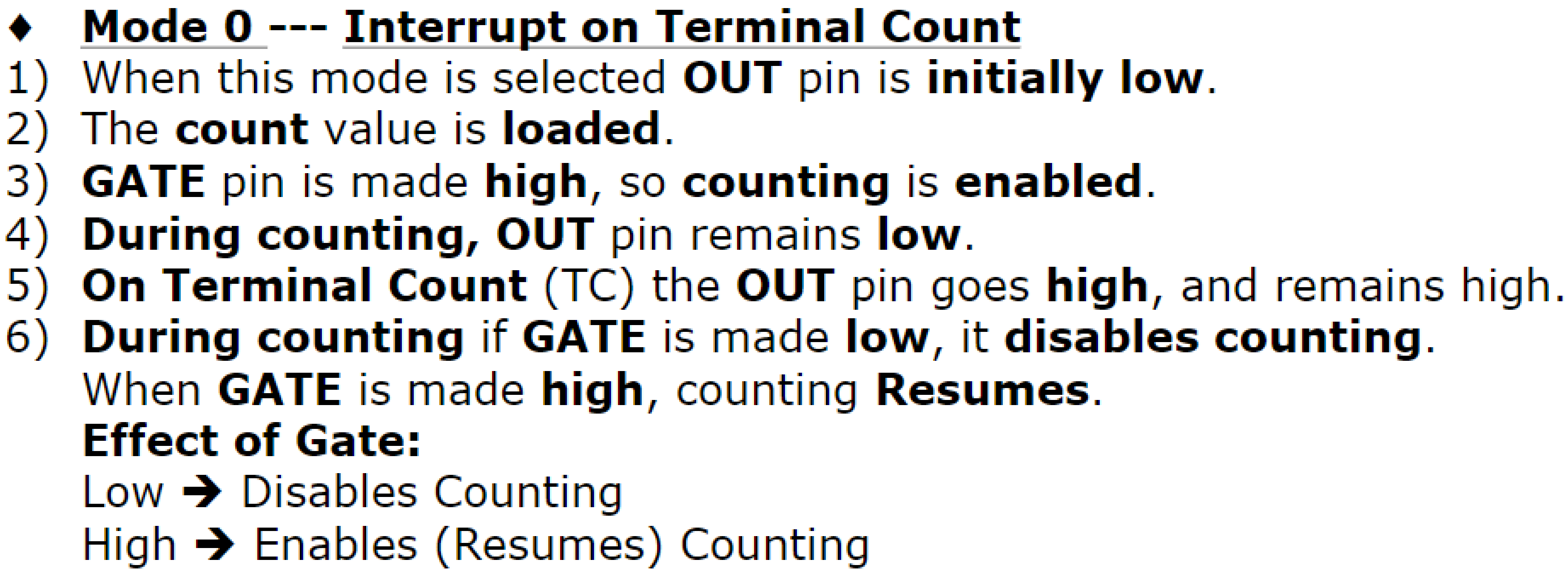
0 0 0 – Mode 0
0 0 1 – Mode 1
0 1 0 – Mode 2
0 1 1 – Mode 3
1 0 0 – Mode 4
1 0 1 – Mode 5

1 – Programmed for BCD counting
0 – Programmed for Binary counting

Format of Status Word of each Control Word of 8254 Timer

Timer modes of 8254

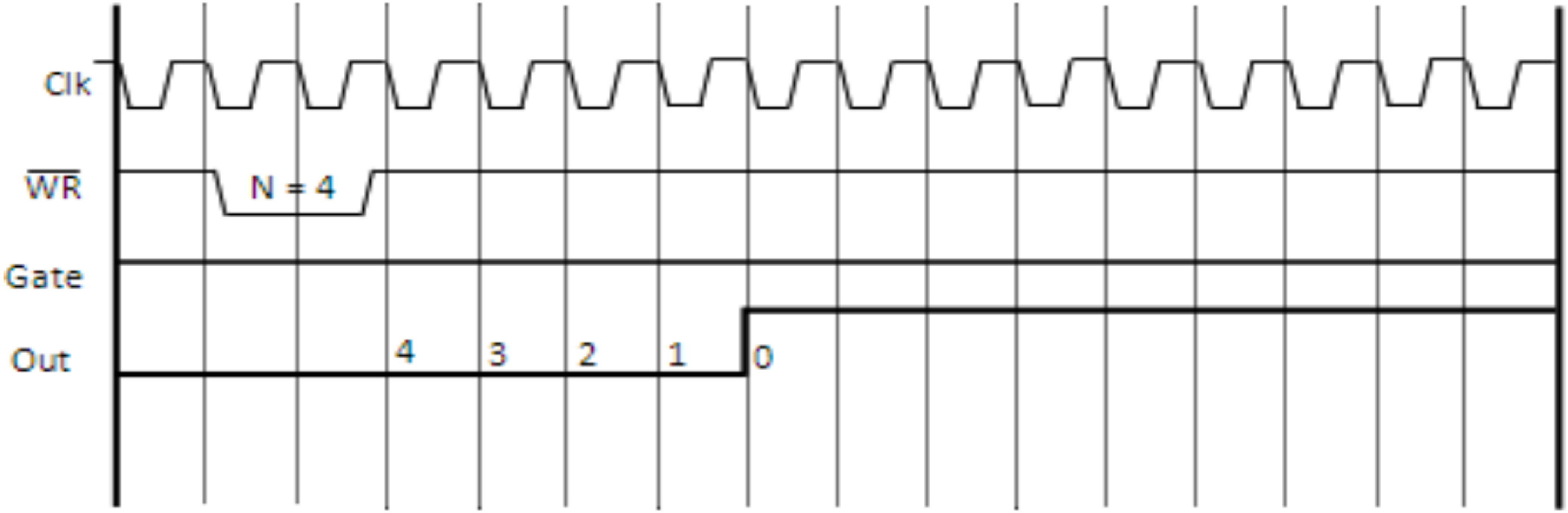
Mode 0:INTERRUPT ON TERMINAL COUNT :



- **Mode 0 --- Interrupt on Terminal Count**

1) When this mode is selected **OUT** pin is **initially low**.
2) The **count** value is **loaded**.
3) **GATE** pin is made **high**, so **counting** is **enabled**.
4) **During counting, OUT** pin remains **low**.
5) **On Terminal Count** (TC) the **OUT** pin goes **high**, and remains high.
6) **During counting** if **GATE** is made **low**, it **disables counting**.
   When **GATE** is made **high**, counting **Resumes**.
   Effect of Gate:
   Low ➔ Disables Counting
   High ➔ Enables (Resumes) Counting

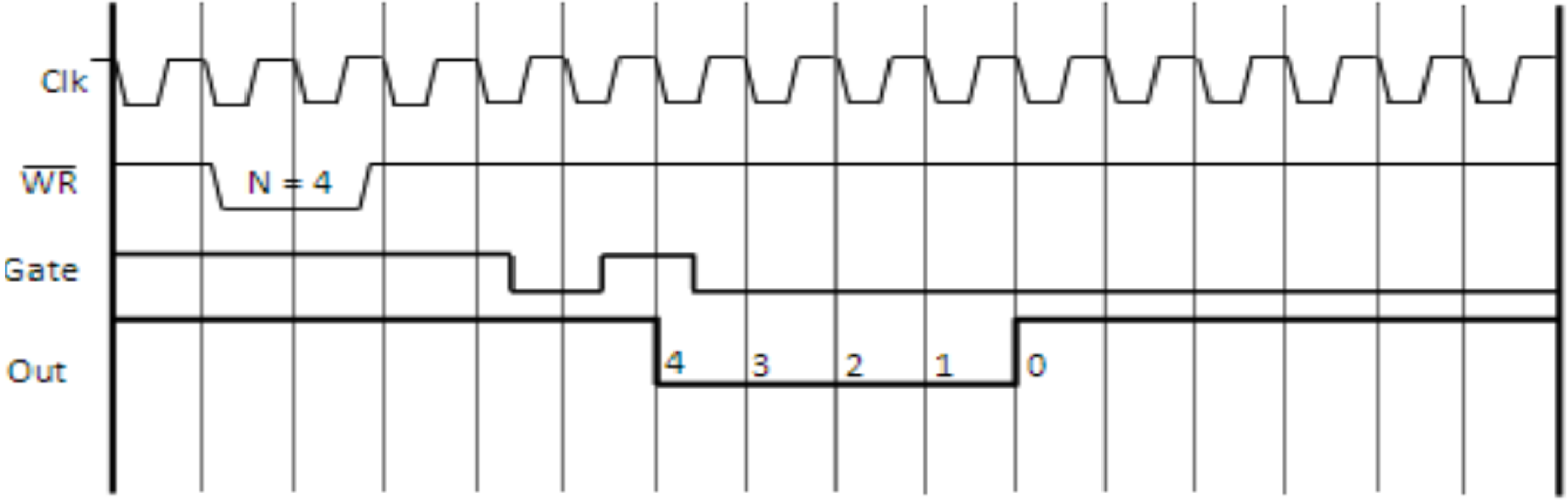
Mode 0: Interrupt On Terminal Count



- **Mode 1 --- Monostable Multivibrator**

1) When this mode is selected **OUT** pin is **initially high**.
2) The **count** value is **loaded**.
3) **Counting begins ONLY when** a **rising edge** is applied to the **GATE**.
4) **OUT** pin goes **low** and remains low **during counting**.
5) **On Terminal Count** (TC) the **OUT** pin goes **high**, and remains high.
6) **During counting** if **GATE** is made **low**, it **has no effect** on the Counting**.**
7) The **GATE pin** can be used as a **Trigger**.
   The **Counter** can be **re-triggered** by applying a **rising edge** on the **GATE**.
   This would **Restart** the **counting**, and hence re-trigger it.
   Effect of Gate:
   Low → No Effect
   High(Trigger) → Starts Counting, can also re-rtigger it.

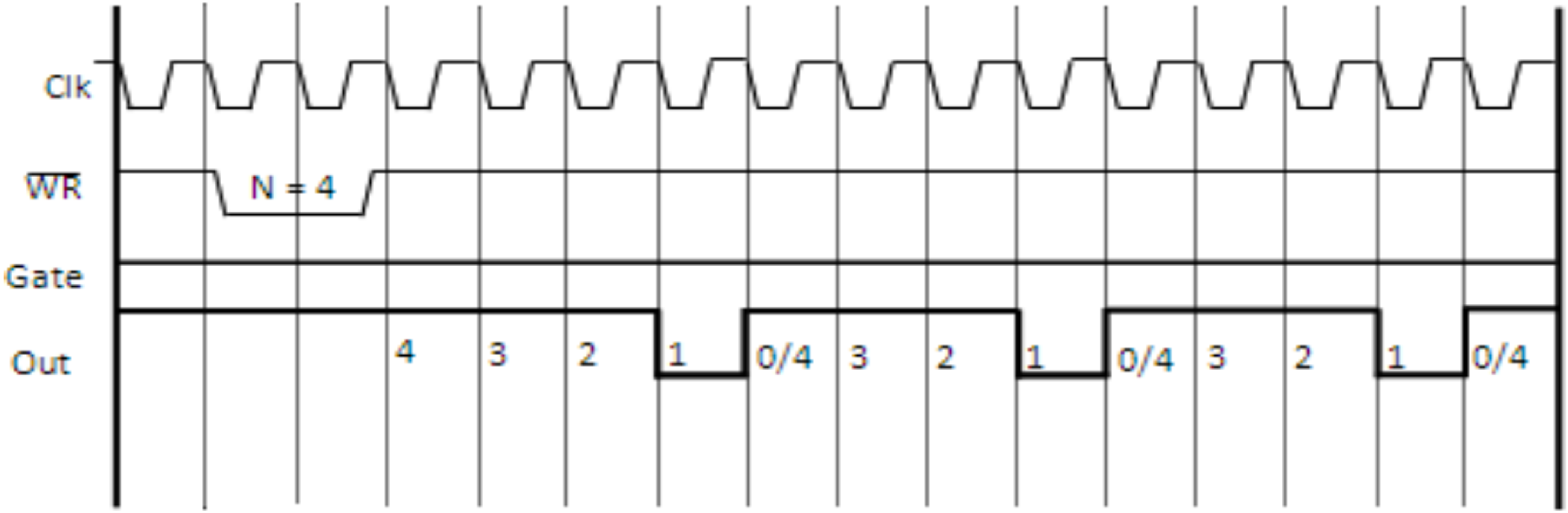
Mode 1: Monostable Multivibrator



♦ Mode 2 --- Rate Generator

1) When this mode is selected **OUT** pin is **initially high**.
2) The **count** value is **loaded**.
3) **GATE** pin is made **high**, so **counting** is **enabled**.
4) **During counting, OUT** pin remains **high**.
5) The OUT pin goes low for one clock cycle just before the TC.
6) The initial count is reloaded and the above process repeats.
   Thus, this mode produces a Continuous Pulse.
7) **During counting** if **GATE** is made **low**, it **disables counting**.
   When **GATE** is made **high**, counting **Restarts**.
   Effect of Gate:
   Low ➔ Disables Counting
   High ➔ Enables (Restarts) Counting
8) It is also called a divide by n counter, as for a count n, the input frequency is divided by n to produce the output frequency.

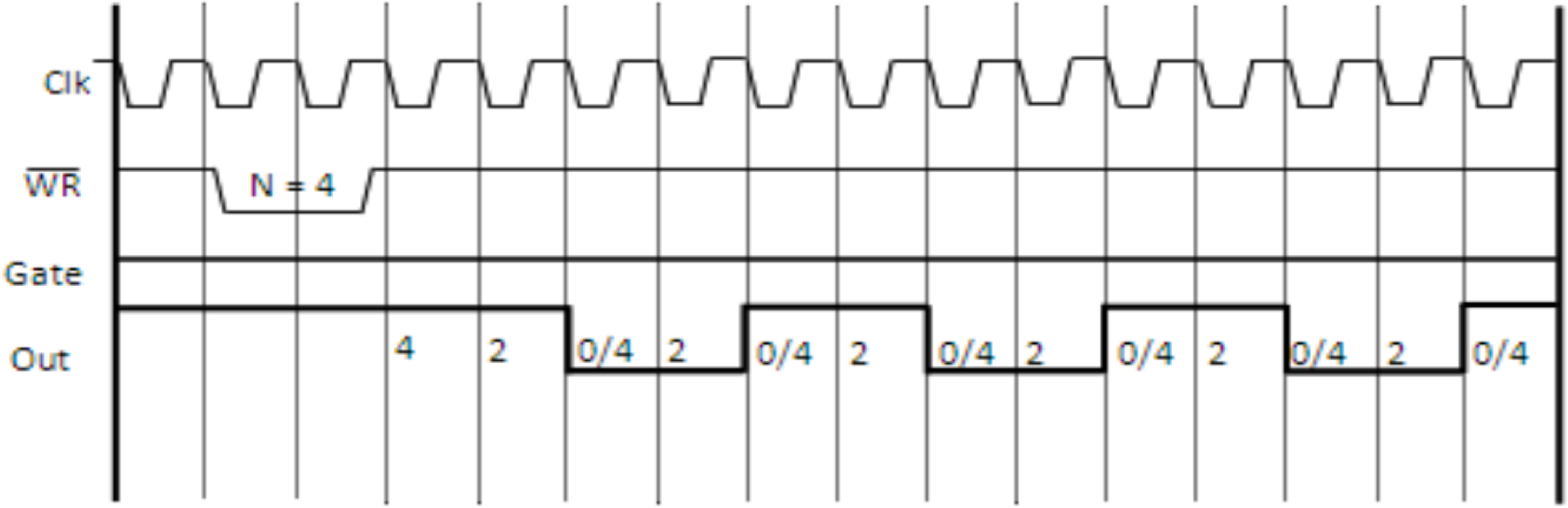
Mode 2: Rate Generator



- **Mode 3 --- Square Wave Generator**

1) When this mode is selected **OUT** pin is **initially high**.
2) The **count** value is **loaded**.
3) **GATE** pin is made **high**, so **counting** is **enabled**.
4) **OUT** pin remains **high** for **half of the count** (n/2) and remains **low** for the **remaining half**.
5) **On TC,** the **Count** is **reloaded** and the **process repeats** itself producing a **continuous square wave**.
6) **During counting** if **GATE** is made **low**, it **disables counting**.
   When **GATE** is made **high**, counting **Restarts**.
   Effect of Gate:
   Low ➔ Disables Counting
   High ➔ Enables (Restarts) Counting
7) If the **count** is **ODD**, the **OUT** pin remains **high** for **(n+1)/2** and **low** for **(n-1)/2.**

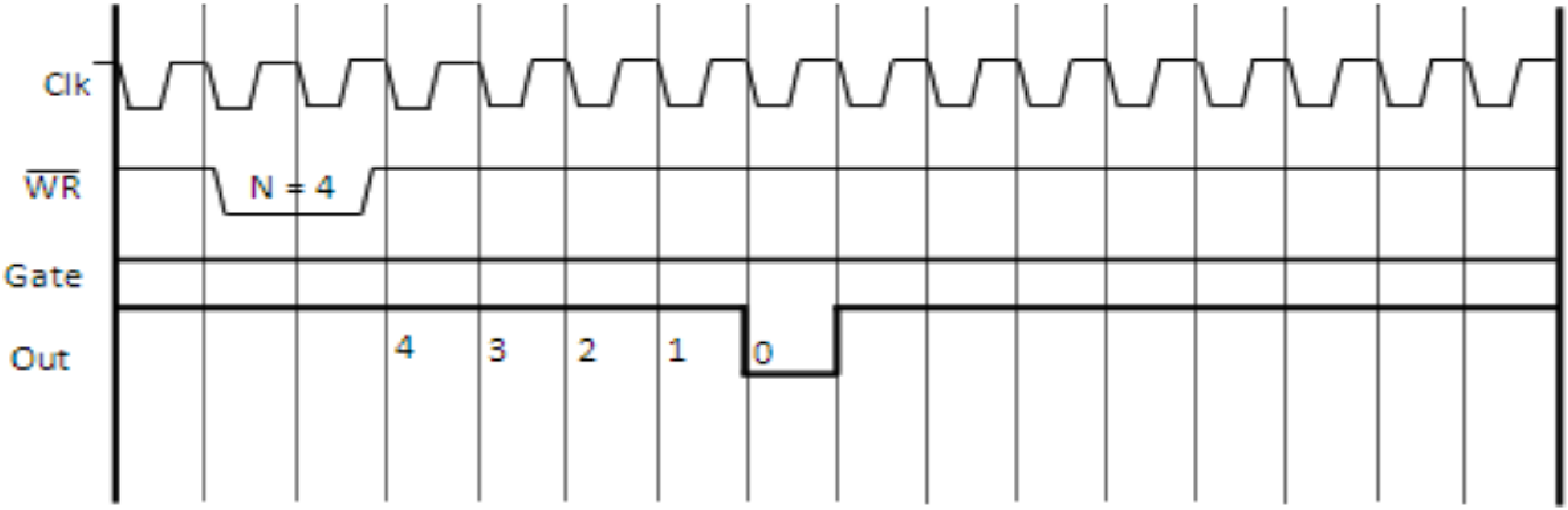
Mode 3: Square Wave Generator



- **Mode 4 --- Software Triggered Strobe**

1) When this mode is selected **OUT** pin is **initially high**.
2) The **count** value is **loaded**.
3) **GATE** pin is made **high**, so **counting** is **enabled**.
4) **During counting, OUT** pin remains **high**.
5) The **OUT** pin goes **low** for **one** clock **cycle, just after TC**.
6) **After that OUT** pin goes **high** and remains high.
7) **During counting** if **GATE** is made **low**, it **disables counting**.
   When **GATE** is made **high**, counting **Restarts**.
   Effect of Gate:
   Low ➔ Disables Counting
   High ➔ Enables (Restarts) Counting

Mode 4: Software Triggered Strobe



- **Mode 5 --- Hardware Triggered Strobe**

1) When this mode is selected **OUT** pin is **initially high**.
2) The **count** value is **loaded**.
3) Counting starts ONLY after a trigger is applied to the GATE pin.
4) Also, the GATE pin need not remain high for the counting to continue.
5) **During counting, OUT** pin remains **high**.
8) The **OUT** pin goes **low** for **one** clock **cycle, just after TC**.
9) **After that OUT** pin goes **high** and remains high.
6) **Thus GATE is used as a Trigger.**

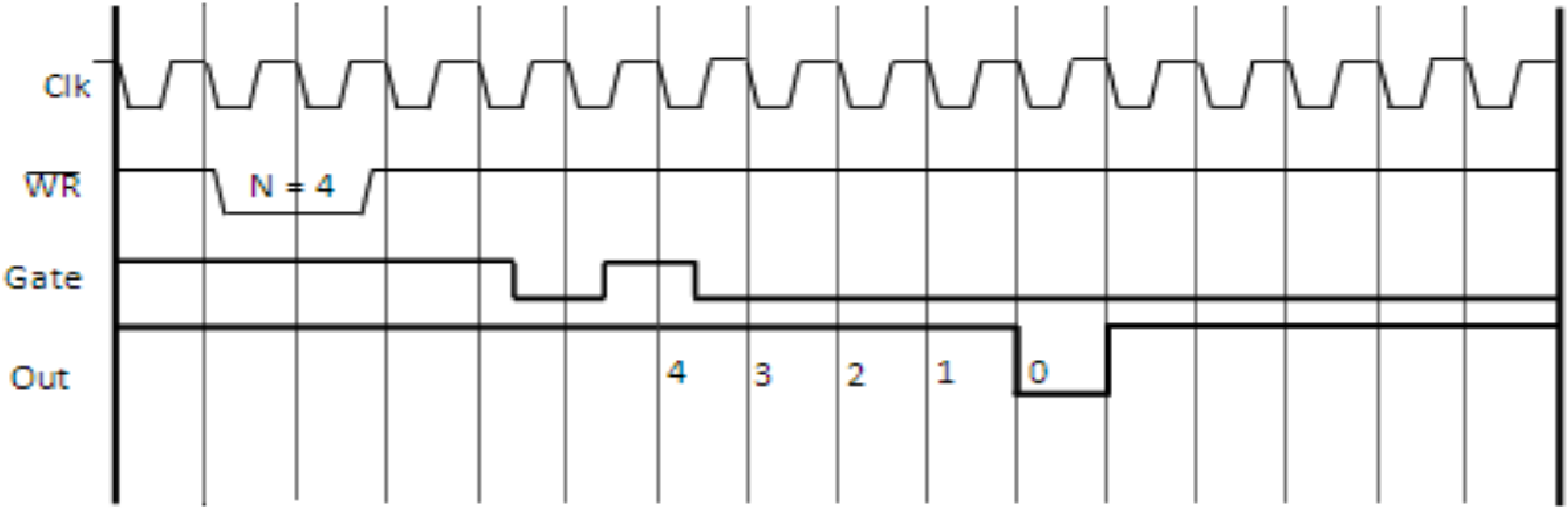
   I.e. It has to be triggered to start counting.

   Effect of Gate:

   Low ➔ No Effect on counting

   High (Trigger) ➔ Starts Counting, can also re-rtigger it.

Mode 5: Hardware Triggered Strobe



MODULE 4

8257 DMAC

DMA Controller 8257

- DMA stands for Direct Memory Access.
- Transfer data at the fastest rate.
- It allows the device to transfer the data directly to/from memory without any interference of the CPU.
- Using a DMA controller, the device requests the CPU to hold its data, address and control bus, so the device is free to transfer data directly to/from the memory. The DMA data transfer is initiated only after receiving HLDA signal from the CPU.

Features of 8257

- It has four channels which can be used over four I/O devices.
- Each channel has 16-bit address and 16-bit counter.
- Each channel can transfer data up to 64kb.
- Each channel can be programmed independently.
- Each channel can perform read transfer, write transfer and verify transfer operations.
- It generates MARK signal to the peripheral device that 128 bytes have been transferred.
- Its frequency ranges from 250Hz to 3MHz.
- It operates in 2 modes, i.e., **Master mode** and **Slave mode**.

DRQ_0–DRQ**3**

- These are the four individual channel DMA request inputs, which are used by the peripheral devices for using DMA services. When the fixed priority mode is selected, then DRQ_0 has the highest priority and DRQ_3 has the lowest priority among them.

$DACK_o$ – $DACK_3$

- These are the active-low DMA acknowledge lines, which updates the requesting peripheral about the status of their request by the CPU. These lines can also act as strobe lines for the requesting devices.

$D_o – D_7$

- These are bidirectional, data lines which are used to interface the system bus with the internal data bus of DMA controller. In the Slave mode, it carries command words to 8257 and status word from 8257. In the master mode, these lines are used to send higher byte of the generated address to the latch. This address is further latched using ADSTB signal.

IOR

- It is an active-low bidirectional tri-state input line, which is used by the CPU to read internal registers of 8257 in the Slave mode. In the master mode, it is used to read data from the peripheral devices during a memory write cycle.

IOW

- It is an active low bi-direction tri-state line, which is used to load the contents of the data bus to the 8-bit mode register or upper/lower byte of a 16-bit DMA address register or terminal count register. In the master mode, it is used to load the data to the peripheral devices during DMA memory read cycle.

CLK

- It is a clock frequency signal which is required for the internal operation of 8257.

RESET

- This signal is used to RESET the DMA controller by disabling all the DMA channels

A_o - A_3

- These are the four least significant address lines. In the slave mode, they act as an input, which selects one of the registers to be read or written. In the master mode, they are the four least significant memory address output lines generated by 8257.

CS

- It is an active-low chip select line. In the Slave mode, it enables the read/write operations to/from 8257. In the master mode, it disables the read/write operations to/from 8257.

A_4 - A_7

- These are the higher nibble of the lower byte address generated by DMA in the master mode.

READY

- It is an active-high asynchronous input signal, which makes DMA ready by inserting wait states.

HRQ

- This signal is used to receive the hold request signal from the output device. In the slave mode, it is connected with a DRQ input line 8257. In Master mode, it is connected with HOLD input of the CPU.

HLDA

- It is the hold acknowledgement signal which indicates the DMA controller that the bus has been granted to the requesting peripheral by the CPU when it is set to 1.

MEMR

- It is the low memory read signal, which is used to read the data from the addressed memory locations during DMA read cycles.

MEMW

- It is the active-low three state signal which is used to write the data to the addressed memory location during DMA write operation.

ADST

- This signal is used to convert the higher byte of the memory address generated by the DMA controller into the latches.

AEN

- This signal is used to disable the address bus/data bus.

TC

- It stands for ‘Terminal Count’, which indicates the present DMA cycle to the present peripheral devices.

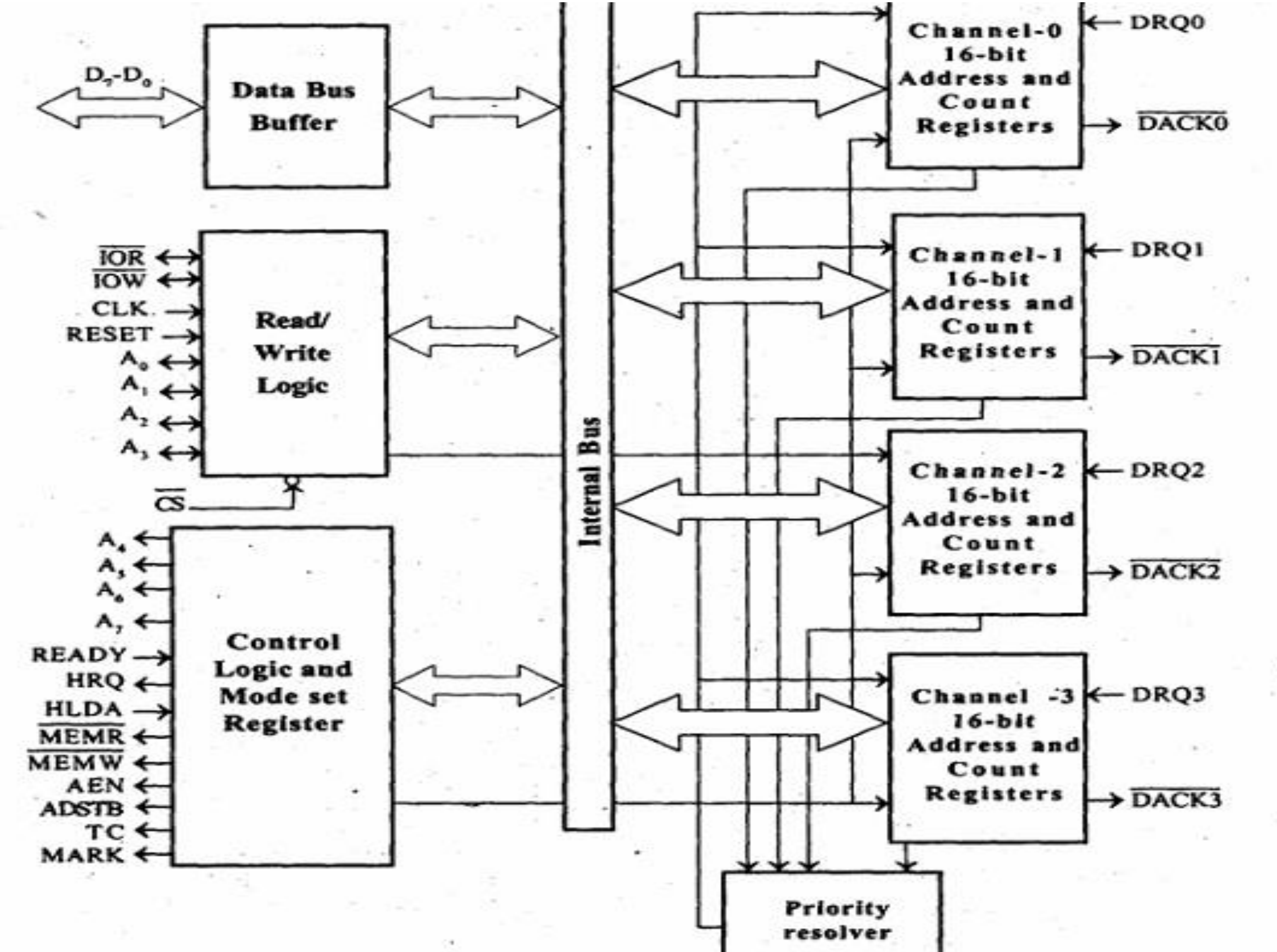
MARK

- The mark will be activated after each 128 cycles or integral multiples of it from the beginning. It indicates the current DMA cycle is the 128th cycle since the previous MARK output to the selected peripheral device.

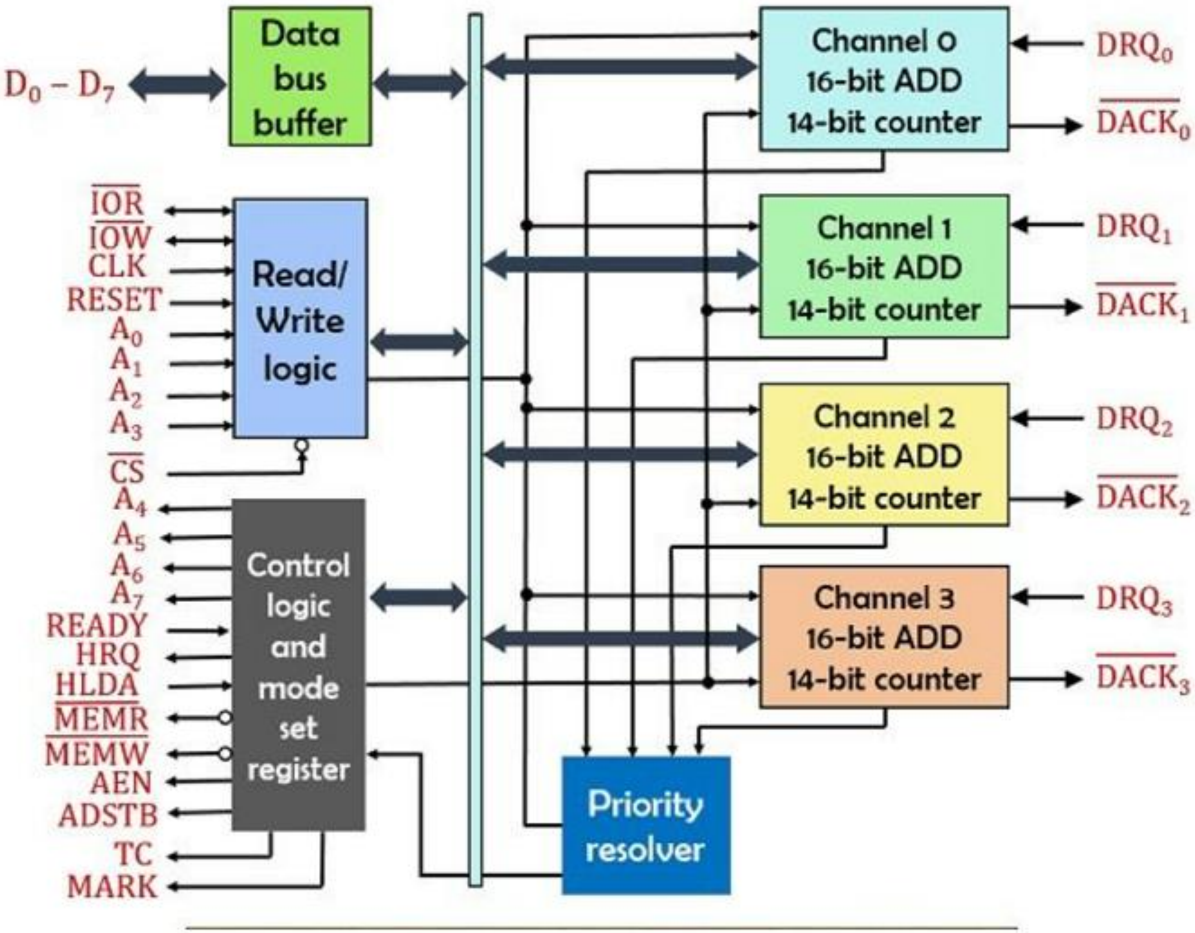
V_{cc}

- It is the power signal which is required for the operation of the circuit.

DMA Controller 8257


D7-D0
Data Bus Buffer
IOR
IOW
CLK
RESET
A0
A1
A2
A3
Read/ Write Logic
CS
A4
A5
A6
A7
READY
HRQ
HLDA
MEMR
MEMW
AEN
ADSTB
TC
MARK
Control Logic and Mode set Register
Internal Bus
Channel-0 16-bit Address and Count Registers
DRQ0
DACK0
Channel-1 16-bit Address and Count Registers
DRQ1
DACK1
Channel-2 16-bit Address and Count Registers
DRQ2
DACK2
Channel -3 16-bit Address and Count Registers
DRQ3
DACK3
Priority resolver

DMA Controller 8257



It contains Five main Blocks.

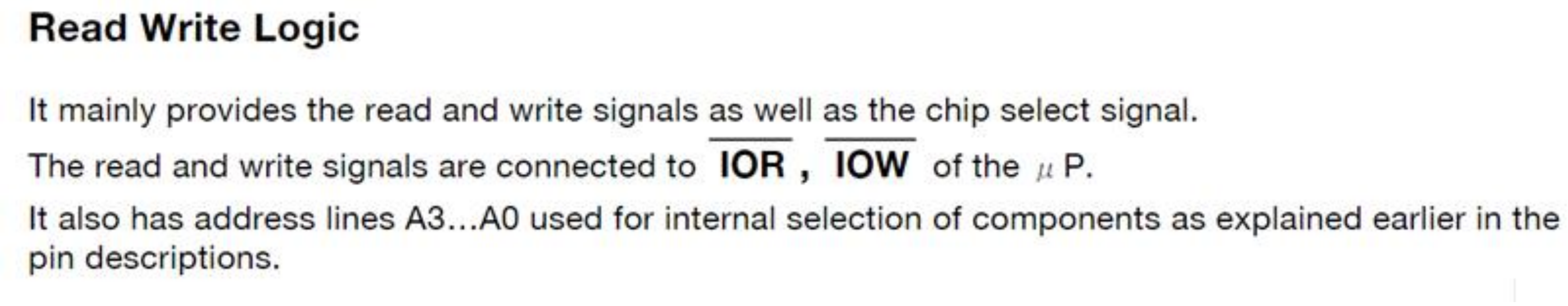
- Data bus buffer
- Read/Control logic
- Control logic block
- Priority resolver
- DMA channels.

DMA Channels

- 8257 has 4 independent DMA channels (CH0 to CH3), hence four I/O devices can request for DMA simultaneously
- Each channel consists of two 16-bit registers (i) Address register (ii) Count Register.
- Address register holds the starting address of the memory block to be accessed by I/O device
- Count register holds the number of bytes to be transferred during DMA action.
- The low order 14-bits of count register specify the number of bytes; therefore each channel can support data transfer of 2^{14}=16KB during DMA process.
- The high order 2-bits of count register specify the mode of operation (read, write or verify)

Data Bus Buffer: This 8-bit Bidirectional buffer is used to interface the 8257 to the system Data Bus

- It allows the transfer of data and information between 8257 and microprocessors.



Read Write Logic

It mainly provides the read and write signals as well as the chip select signal.

The read and write signals are connected to $\overline{\textbf{IOR}}$ **,** $\overline{\textbf{IOW}}$ of the μP.

It also has address lines A3…A0 used for internal selection of components as explained earlier in the pin descriptions.

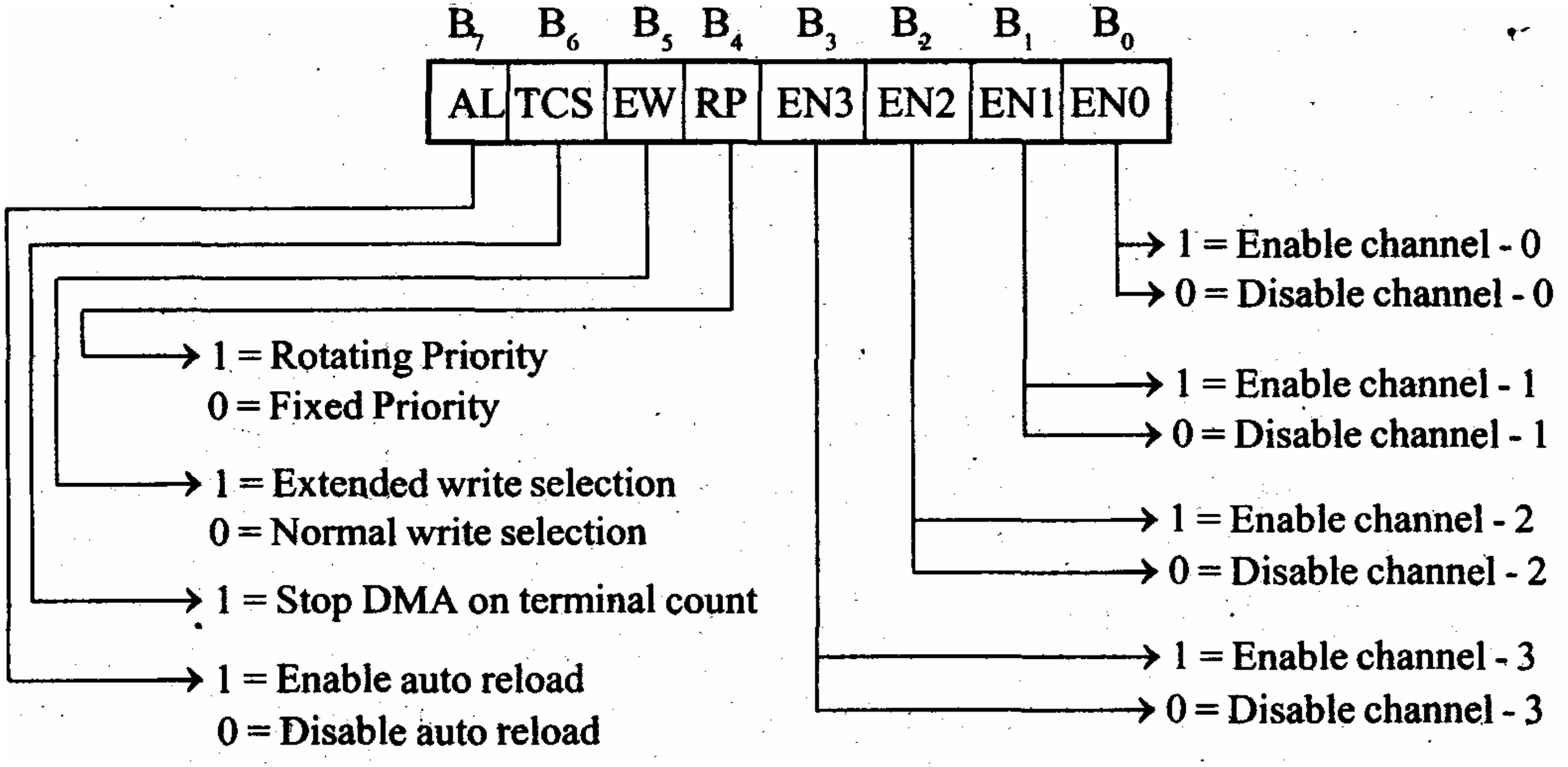
Priority Resolver

- This logic block determines the priorities of the channels when more than one I/O device request for DMA.
- By default the priority resolver work in fixed priority mode. In this mode the CH0 has the highest priority while the CH3 has lowest.
- In rotating priority, the priority of the channels has a circular sequence. The channel which has being just serviced move to the lowest priority and channel next to it move to the highest priority, hence each channel achieves the highest priority in rotation.

Control Logic and Mode Set Register

It generates the internal control signals for the DMAC.
It also provides external signals such as HRQ (Hold), HLDA, AEN, ADSTB, TC, MARK etc. as explained earlier.
Additionally, it has the Mode Set register.

Mode Set Register of 8257



B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
AL	TCS	EW	RP	EN3	EN2	EN1	EN0

1 = Enable channel - 0
0 = Disable channel - 0

1 = Rotating Priority
0 = Fixed Priority

1 = Enable channel - 1
0 = Disable channel - 1

1 = Extended write selection
0 = Normal write selection

1 = Enable channel - 2
0 = Disable channel - 2

1 = Stop DMA on terminal count

1 = Enable channel - 3
0 = Disable channel - 3

1 = Enable auto reload
0 = Disable auto reload

Extended Write Mode

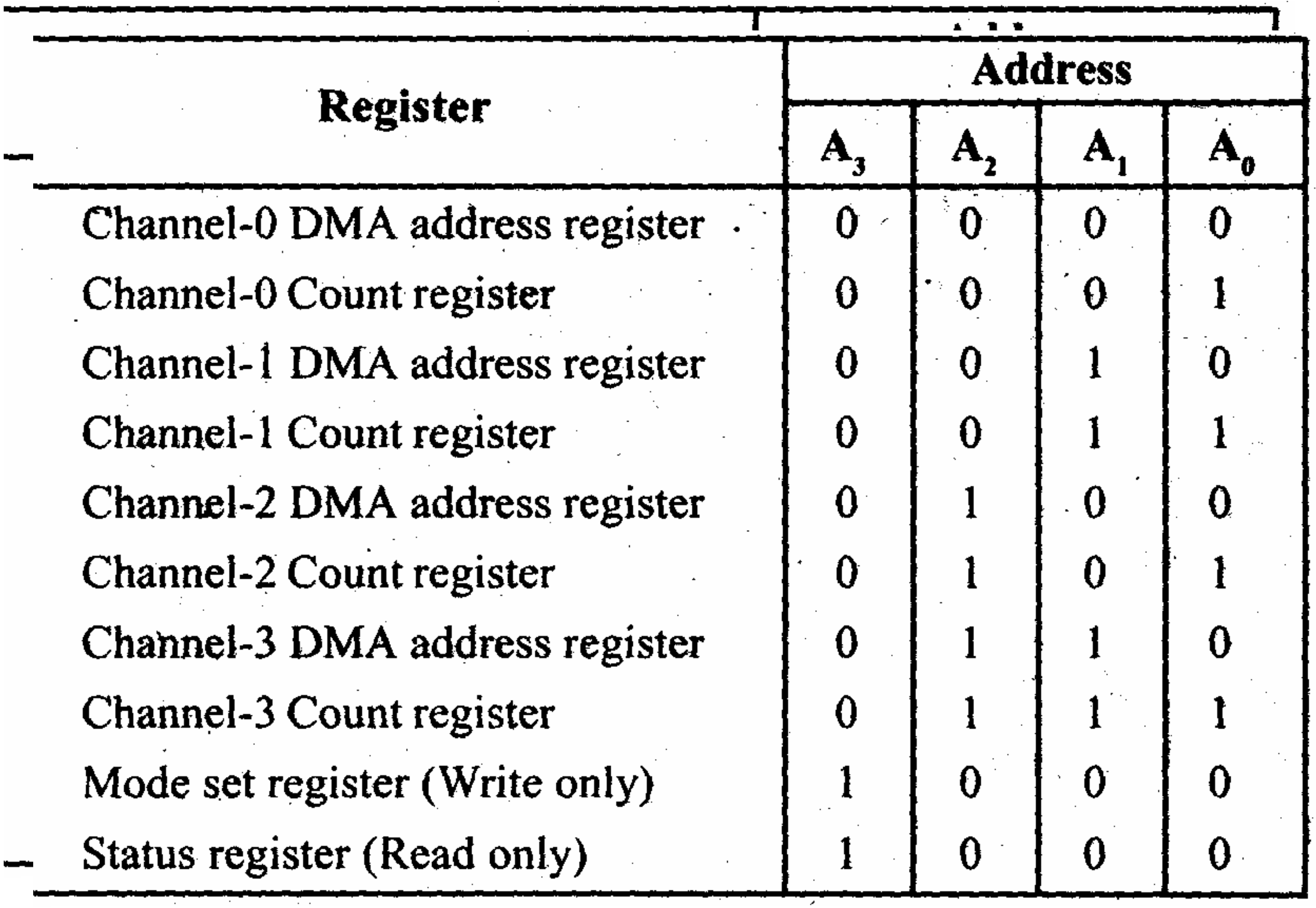
Here the Write control signal gets activated one T-State in advance.
This is similar to the Advanced write signals of 8086 Maximum Mode.
Once the Write signal gets activated (goes low), the I/O device has to respond by Making Ready Signal=1 to indicate that it is ready for the transfer. A slow device may require additional time to get ready and hence this will force the DMAC to insert Wait states in between every cycle, thus making the whole operation slow.
To avoid this, we use the extended write mode.
Here, the Write signal goes low one T-state in advance hence the I/O gets a little extra time to become ready. This reduces the chances of inserting Wait states and hence prevents a slightly slow I/O device from making the whole DMA operation slow.

Auto Load Mode

This is a continuous self-reloading mode.
It is applicable only for Channel 2.
When this mode is selected the original count and address register values of Channel 2 are stored as a back up in Channel 3 registers.
After every byte is transferred, Channel 2 registers keep changing but Channel 3 registers maintain the original values.
When Channel 2 reaches TC, There is an Automatic reload of address and count information from Channel 3 registers to Channel 2 registers and the DMA transfer restarts. This mode is basically used to perform repetitive DMA transfers.

- *mode set register* and *status* registers are *c*ommon for all the channels
- Thus there are a total of ten registers. The CPU selects one

Register Selection



Register	Address			
	A_3	A_2	A_1	A_0
Channel-0 DMA address register	0	0	0	0
Channel-0 Count register	0	0	0	1
Channel-1 DMA address register	0	0	1	0
Channel-1 Count register	0	0	1	1
Channel-2 DMA address register	0	1	0	0
Channel-2 Count register	0	1	0	1
Channel-3 DMA address register	0	1	1	0
Channel-3 Count register	0	1	1	1
Mode set register (Write only)	1	0	0	0
Status register (Read only)	1	0	0	0

DMA Address Register

- Each DMA channel has one DMA address register. The function of this register is to store the address of the starting memory location, which will be accessed by the DMA channel. Thus the starting address of the memory block which will be accessed by the device is first loaded in the DMA address register of the channel.

Terminal Count Register

- Each of the four DMA channels of 8257 has one terminal count register (TC).
- This 16-bit register issued for ascertaining that the data transfer through a DMA channel ceases or stops after the required number of DMA cycles.
- The low order 14-bits of the terminal count register are initialized with the binary equivalent of the number of required DMA cycles minus one.

After each DMA cycle, the terminal count register content will be decremented by one and finally it becomes zero after the required number of DMA cycles are over.

The bits 14 and 15 of this register indicate the type of the DMA operation (transfer).

Count Register: 16 bit (D15...D0)

D7	D6	D5	D4	D3	D2	D1	D0

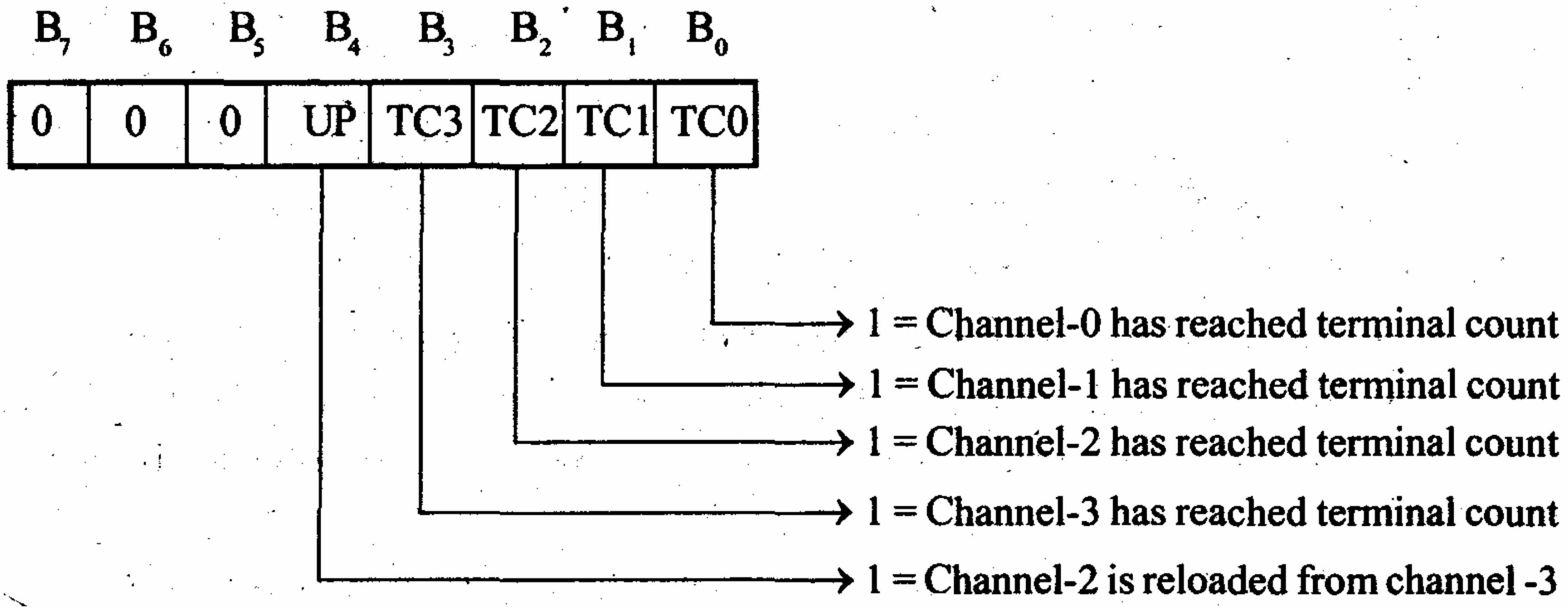
🡐 Lower 8 bits of the 14 bit count 🡒

D15	D14	D13	D12	D11	D10	D9	D8

🡐 Mode bits (2) 🡒 🡐 Higher 6 bits of the 14 bit count 🡒

Mode Bits		**Mode Selected**
0	0	DMA Verify Cycle
0	1	DMA Write Cycle (I/O to Memory)
1	0	DMA Read Cycle (Memory to I/O)
1	1	No action

Status Register of 8257



B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
0	0	0	UP	TC3	TC2	TC1	TC0

1 = Channel-0 has reached terminal count

1 = Channel-1 has reached terminal count

1 = Channel-2 has reached terminal count

1 = Channel-3 has reached terminal count

1 = Channel-2 is reloaded from channel -3

- If any of the lower 4 bit is set,it indicates that the specific channel has reached the terminal count condition
- The update flag is not affected by the read operation
- If UP=1,content of channel 3 are reloaded to channel 2 register
- Whenever the channel 2 reaches a TC condition , after transferring one block and the next block is to be transferred using the autoload feature of 8257
- UP flag is set every time channel 2 registers are loaded with contents of channel 3 register

Update flag

- Update flag is not affected by the read operation.
- Flag can only be cleared by resetting 8257 or by resetting the auto load bit of the mode set register
- If UP=1,content of channel 3 are reloaded to channel 2 register
- UP flag is set every time channel 2 registers are loaded with contents of channel 3 register