7/3/18

8259 PIC:

End of Interrupt (EOI) Command:

1) Non specific EOI

2) Specific EOI

3) Automatic EOI

EOI command resets the set bit in ISR after the interrupt is served.

1) Non Specific Interrupts

When 8259 receives this command, it will reset the high priority IR level set bit of ISR.

2) Specific - EOI

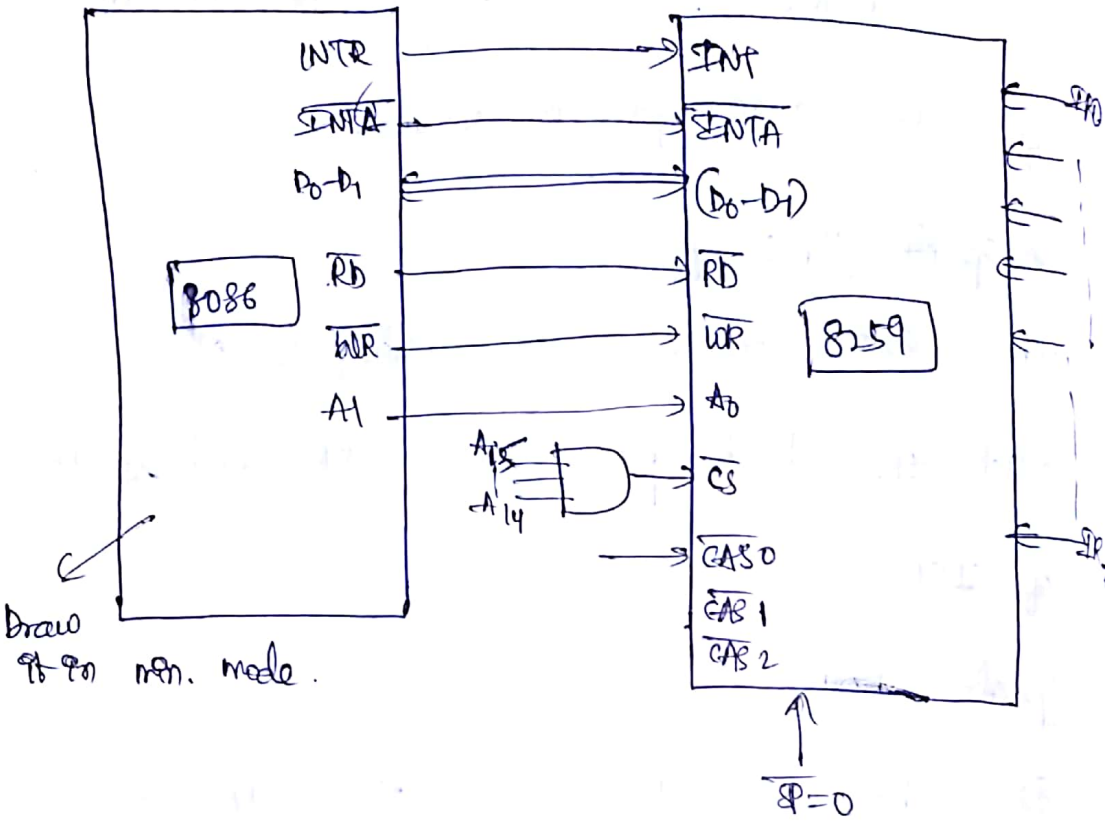
8259 has 8 IR levels. Each IR level can be specified by 3 bit address. When this command is given to 8259, it will reset the specified IR level of ISR.

3) Automatic EOI:

In response to the INTR pulse, 8259 will set the corresponding INTA IR level of ISR

and in response to 2nd INTA pulse, 8259 will reset the corresponding set bit of ISR by giving this command.

8259 Interfacing to 8086:



Draw on mem. mode.

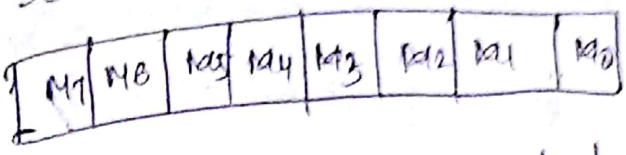
Command words: (CW)

- Initialisation CW's
  - ICW1
  - ICW2
  - ICW3
  - ICW4
- Operational CW's
  - OCW1
  - OCW2
  - OCW3

Atleast 2 ICW's are required

OCW:

It is programmed to mask the particular IR level



M7 – 1 → IR 7 is masked

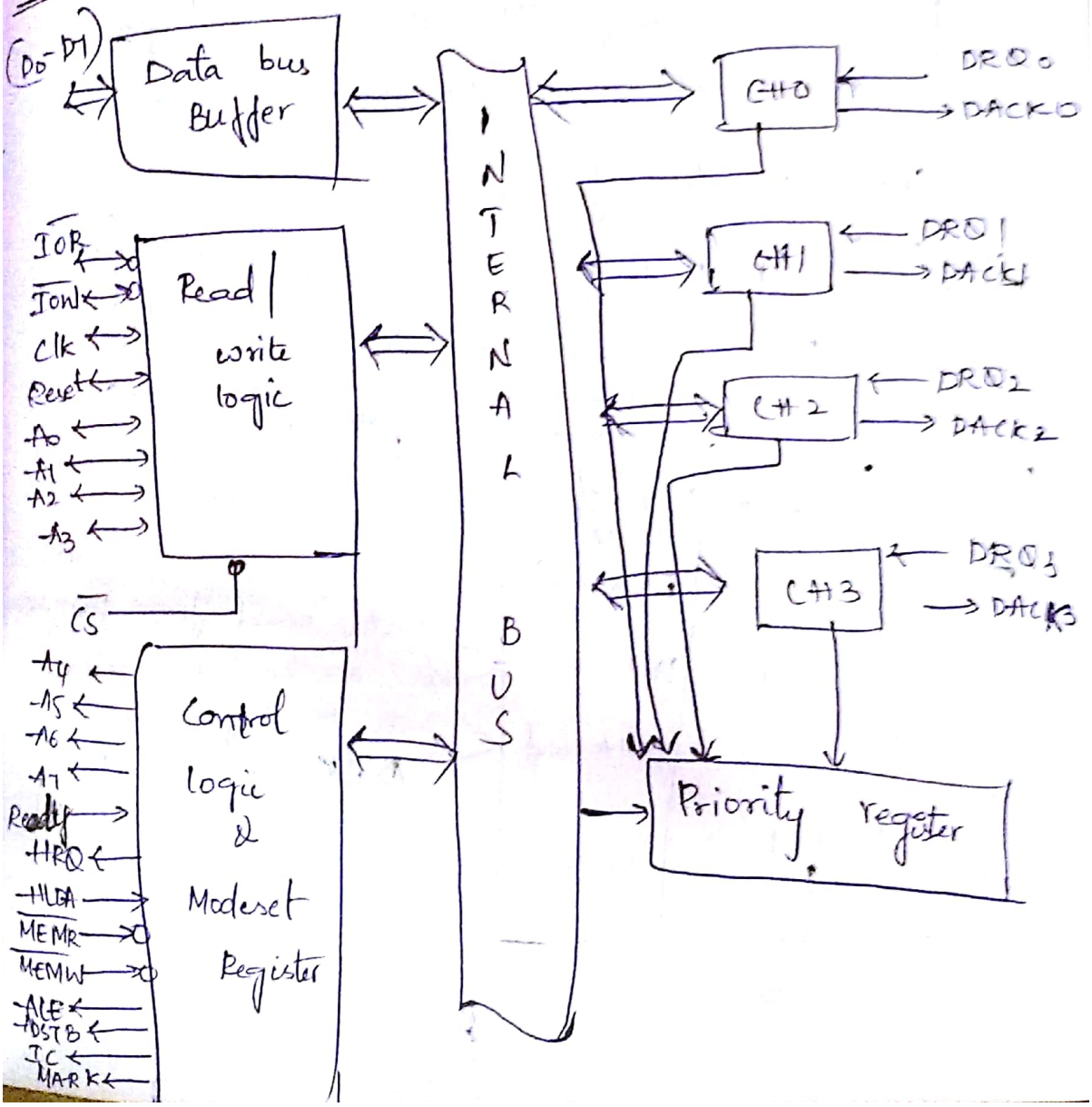
– 0 → IR7 is not masked

Mn – 1 → IR n is masked

– 0 → " " not masked

12/8/17

DMA Controller : (8257)



→ HRQ of DMA is connected to HOLD of 8086

→ If 8086 wants to allow DMA then 8086 send ackno- from HLDA (8086) to HLDA (8257)

→ Then DMA sends DACK to peripheral & DRQ is received by DMA from peripheral then data transmission takes place

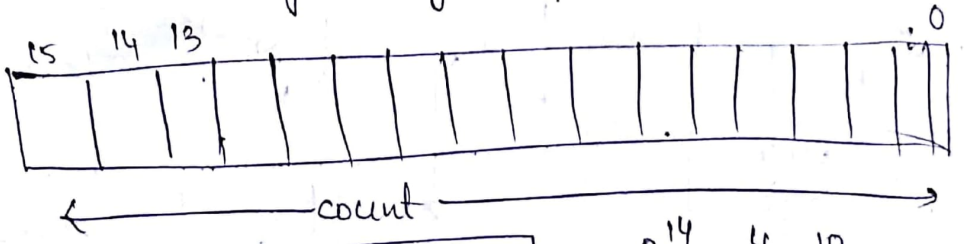
→ Each channel is capable of transmitting 16 kB

→ each channel has 2 16-bit registers. Add Reg & Count Register

→ A.R stores starting address of memory from where data transfer should begin

→ C.R. stores the count value

Count register format.



$2^{14} = 2^4 \cdot 2^{10}$

$= 10\,kB$

Bit 15	Bit 14	Type of DMA
0	0	DMA verify
0	1	DMA write
1	0	DMA read
1	1	None

Data read from peripheral
Data written into memory
Data read from memory
Data written in to peripheral

→ Channel 0 has high priority ch 3 low priority.

→ Data bus buffer used to to send data or control signals.

→ A_0-A_3 are used to select registers,

A_3	A_2	A_1	A_1	
0	0	0	0	– CH0 – AR
0	0	0	1	– CH0 – CR
0	0	1	0	– CH 1 – AR
0	0	1	1	– CH 1 – CR
0	1	0	0	– CH 2 – AR
0	1	0	1	– CH 2 – CR
0	1	1	0	– CH 3 – AR
0	1	1	1	– CH 3 – CR
1	0	0	0	– control/status word.

→ 8257 is operated in 3 states

1) Idle state – performs nothing

2) Slave u – µp will write CW to 8257/reads SW from 8257

3) Master u – 8257 will control the system bus.

→ $\overline{IOR}$, $\overline{IOW}$, A_0-A_3 used as i/p's during slave & o/p's during master.

→ $\overline{MEMR}$ – Memory Read. It is active during DMA read.

→ MEMWR - Memory Write. It is active during DMA write

→ AEN - Address Enable 2) ⇒ used to disable (D_0-D_7) of 8086

→ ADSTB - Address strobe - used to latch D_0-D_7 of 8086

→ TC - Terminal count - when data transfer is completed then TC = 1

→ MARK - for every 128 cycles it is going to be high. It indicates present DMA cycle is 128th from previous mark o/p. Mark always occurs at 128 cycles from the end of data block

Types of DMA transfer:

Byte - Byte by byte transfer.

Block - Block by block transfer.

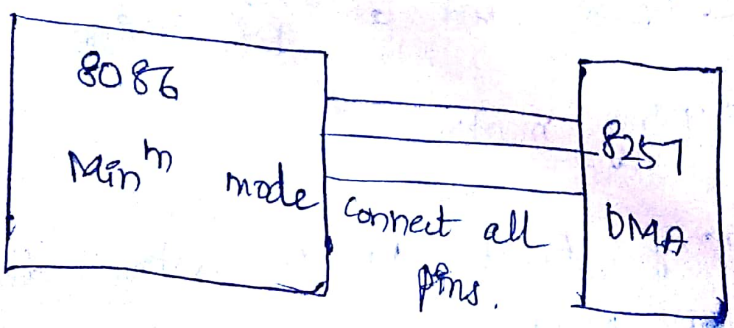
Ex: Total data 10kB

Block size 1kB

Demand - no. of cycles required - 10 until 10kB of data transfered.

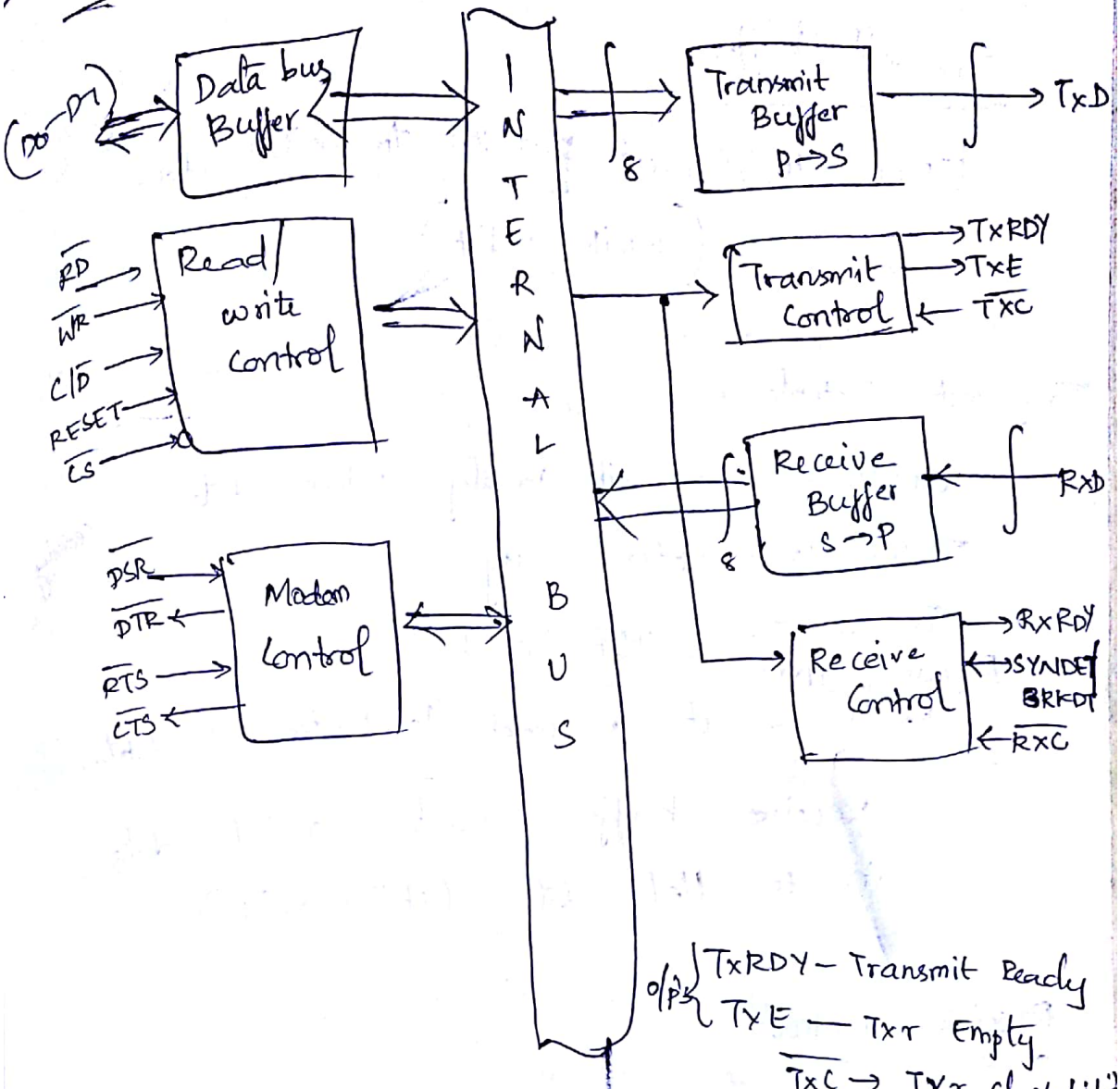
8257 does nothing.

Interfacing 8257 with 8086



14/2/18

8251 USART



o/p's { TxRDY – Transmit Ready
TxE – Txr Empty.
TxC → TXr clock i/p

TxD – Transmit Data

RxD – Receive Data

RxRDY – Receive Ready.

RxC – Receive clock

8251 is Connected to 8086.

It supports Universal synchronous Asynchronous receiver Transmitter.

~~8251 is Conne~~

B/w serial I/o devices & 8086, 8251 is connected. There are 7 blocks

Transmit buffer:

Used to convert parallel to serial data (8 bit → 1 bit)

Transmit Controls:

TXRDY – Transmit ready- when data is rea

TXE – Transmitter empty

TxC – Transmitter clock

Receive Buffer:

RxD – it is used to receive serial data.

Receive buffer converts serial data in to llel data (1bit → 8bits)

Receive control:

RxRDY – Receive data ready =1 ⇒ ready to receive data

SYNDET/BRKDT – Sync Detect / Break Detect.

C/$\overline{D}$=0 ⇒ Data is transferred b/w 8251 & 8086

C/$\overline{D}$=1 ⇒ Control lines are transferred b/w 8251 & 8086

$\overline{DSR}$

$\overline{DTR}$

$\overline{RTS}$

$\overline{CTS}$

$\overline{DSR}$ – Data set Ready

It is i/p. It is checked by CPU using status read operation. It is used to check if the data set is ready when communicating with a modem

$\overline{DTR}$ – Data Terminal Ready

→ o/p. Used to indicate that the device is ready to accept data when 8251 is communicating with a modem.

$\overline{RTS}$ – Request to send

→ i/p. when it is zero, it indicates modem that the receiver is ready to receive a data byte from the modem.

$\overline{CTS}$ – clear to send

→ o/p. when $\overline{CTS}$ = 0, 8251 is enabled to transmit serial data provided the enable bit in the command byte is set to '1'

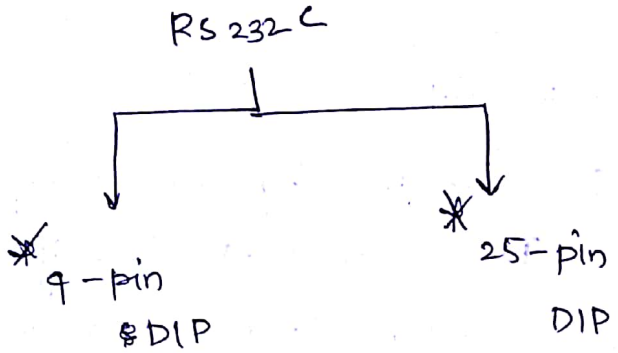
Command words:

RS 232C:

→ Serial Comm^n standard or protocol

which is connected b/w 8251 & I/o devices

→ EIA (Electronics Industrial Association) in 1960 introduced this connector.



It is developed before TTL logic level

(5V – logic 1
0V – " 0).

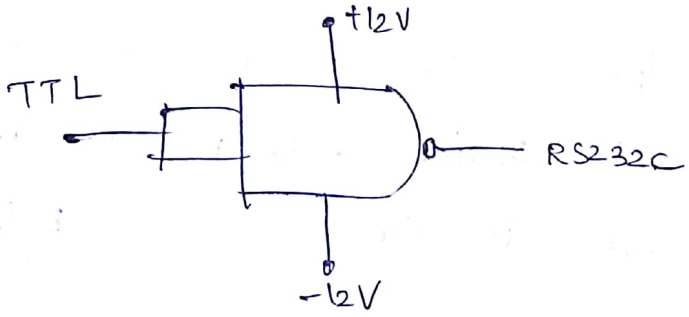
RS232C logic levels:

+3V to +15V → logic '0'

−3V to −15V → logic '1'

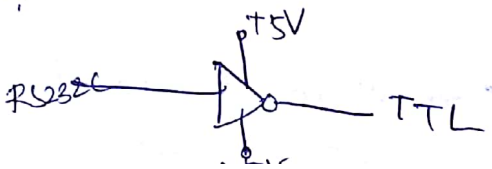
Line Driver: Used to Convert

TTL to RS232C



IC 1488.

Line Receiver:



IC 1489

devices

ciation) in 1960

logic level
- logic 1)
- " 0)

Interfacing serial o/p device to 8251: