# MICROPROCESSOR SYSTEMS

## Supporting Hardware for 8086

Zuhaib A. Shaikh,
Asst. Prof., CSE Deptt.,QUEST
Web: zuhaib-shaikh.neocities.org

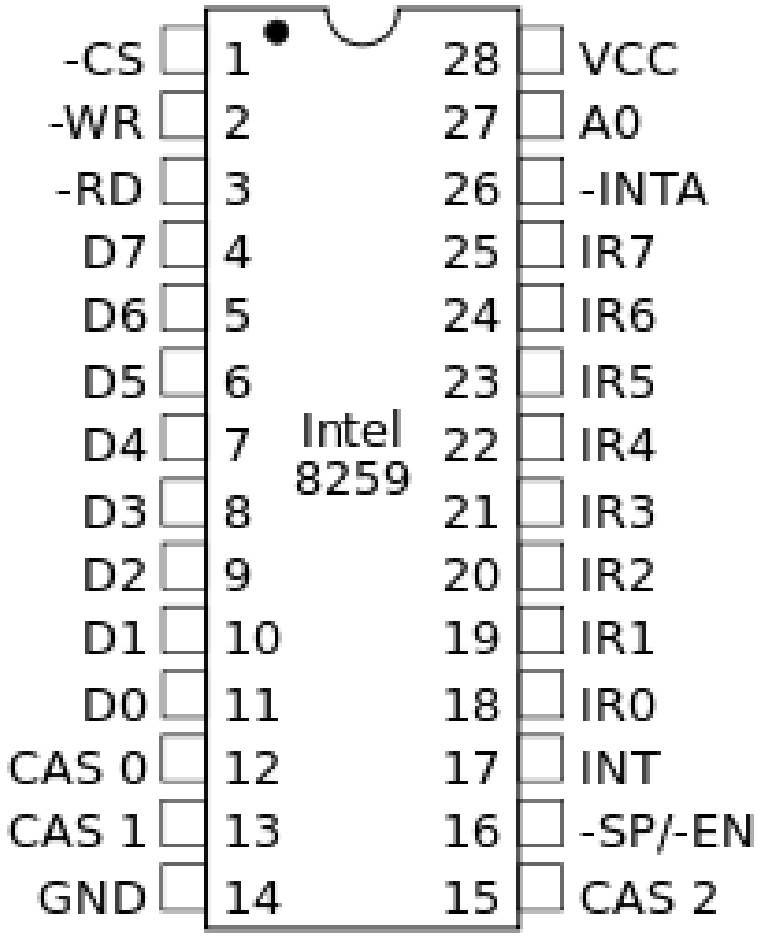
# Intel 8259A Programmable Interrupt Controller

(Pin Diagram)

- Programmable interrupt controller is used to handle interrupt signals from other components and communicate with µP.
- 8259A has 28 pins
  - CS: Chip Select to enable module
  - WR: Write control words
  - RD: Read 8259 status
  - A0: First bit to initiate w.r.t. address
  - D7 – D0: Data lines to send/receive status/control words
  - CAS0 – CAS2: Cascaded lines, used to distinguish multiple 8259s

# Intel 8259A Programmable Interrupt Controller

(Pin Diagram)

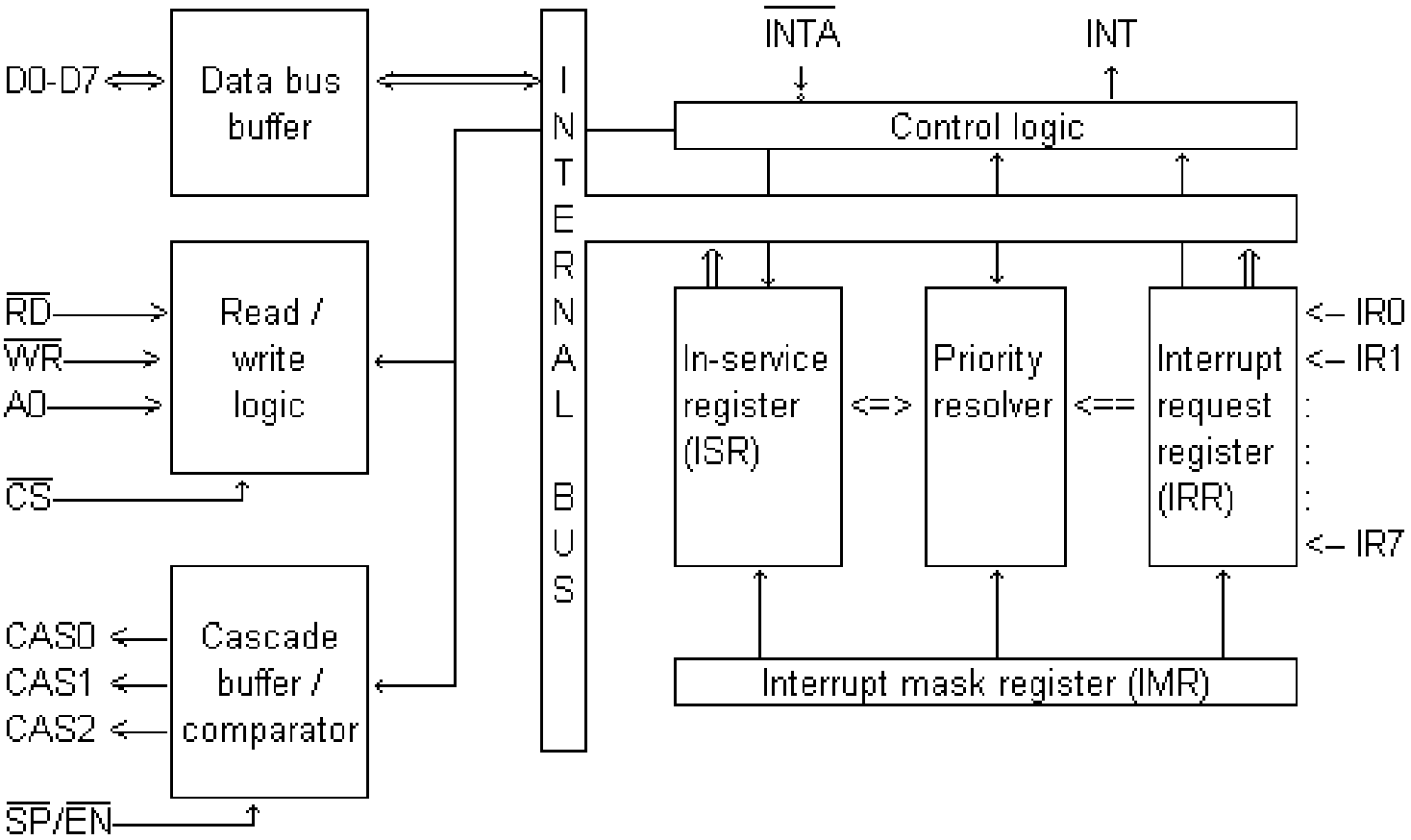


# Intel 8259A Programmable Interrupt Controller

(Pin Diagram)

- SP/EN: Slave Program/Enable Buffer to define Master/Slave mode and to enable transceiver buffer
- INT: Interrupt signal towards µP
- INTA: Interrupt acknowledgement from µP

# Intel 8259A Programmable Interrupt Controller



# Intel 8259A Programmable Interrupt Controller

(Block Diagram)

- IRR: receives interrupt requests and stores for priority resolver and inform control logic
- Priority resolver read IRR and selects highest priority to entertain
  - Priorities are set from highest to lowest as IRQ0 to IRQ7
  - Highest priority signal is informed to control logic
- ISR: sets a bit high for IRQ which currently is being entertained, maintained by control logic ckt
- IMR defines which interrupts are to be masked, according to Operational Command Word received from µP

# Intel 8259A Programmable Interrupt Controller

(Block Diagram)

- Data bus buffer is a transceiver that send/receive status/command words
- Read/Write logic is used to enable the component and define the mode
- Cascaded Buffer/Comparator is used to expand interrupts of 8259
  - By providing master/slave interfacing
  - Each 8259 is represented with unique address/ID

# Interrupt Handling

- Interrupt Generation
  - Device sends an interrupt signal.
  - Device signals the PIC by activating the IR line.
    - This changes its state from a 0 (No power) to a 1 (Power is going through the line.)
  - The PIC sets the bit representing the IRQ inside of the Interrupt Request Register (IRR) from bit 0 will be set to 1.
  - The PIC examine the Interrupt Mask Register (IMR) to see if the interrupt can be serviced.
    - If the interrupt can be serviced, the PIC determines if there are any higher priority interrupts waiting to be serviced. If there is, the interrupt request is waited until the higher priority interrupts are serviced.
    - If the interrupt can be serviced, and there are no higher priority interrupts, the PIC continues to the next step.

# Interrupt Handling

- The PIC signals the processor through the INT pin to inform the processor an interrupt has been fired.
- The processor now knows that an interrupt has been fired.

# Interrupt Handling

- Processor receives interrupt
  - The CPU completes execution of the current instruction.
  - The CPU examines the Interrupt Flag (IF) within FLAGS.
    - If IF is set, the CPU acknowledges the interrupt request through the INTR pin back to the PIC.
    - If IF is cleared, the interrupt request is ignored.
  - Push registers' contents into the stack
  - The PIC receives the acknowledgment signal through INTA.
  - The PIC places the interrupt vector number into the D0-D7 pins.
    - This interrupt vector number is obtained from the Initialization Control Word (ICW) during initialization of the PIC.
  - CPU load address into CS & IR from IVT and start executing interrupt-service procedure

# Programming 8259A

- Processor sends initialization command words and operational command words
- When started, ICWs are sent to PIC
  - ICW1, ICW2 AND ICW4 must be send
  - ICW3 is sent if PICs are in cascaded mode

**ICW1**

| A0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 0 | A7 | A6 | A5 | 1 | LTIM | ADI | SNGL | IC4 |

**IC4:** 0=no ICW4, 1=ICW4 required
**SGL:** 1=Single PIC, 0=Cascaded PIC
**ADI:** Address interval. Used only in 8085, not 8086. 1=ISR's are 4 bytes apart (0200, 0204, etc) 0=ISR's are 8 byte apart (0200, 0208, etc.)
**LTIM:** level triggered interrupt mode: 1=All IR lines level triggered. 0=edge triggered
**A5-A7:** 8085 only. ISR address lower byte segment. The lower byte is of which A7, A6, A5 are provided by D7-D5 of ICW1 (if ADI=1), or A7, A6 are provided if ADI=0.

# Programming 8259A

**ICW2**

| A0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 1 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

- Higher byte of ISR address (8085), or 8 bit vector address (8086).
- Address of IVT is represented by 8-bit combination, remaining bits are padded as zero from MSB
- W.r.t to IR, 1 or 0 is padded to generate specific IVT index

**ICW3**

| Mode | A0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|
| Master | 1 | S7 | S6 | S5 | S4 | S3 | S2 | S1 | S0 |
| Slave | 1 | 0 | 0 | 0 | 0 | 0 | ID3 | ID2 | ID1 |

- ICW3 is used to mention that which IR input is connected with a PIC slave for master PIC
- Slave PIC have ICW3 to have a unique 3-bit ID

# Programming 8259A

**ICW4**

| A0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | SFNM | BUF | M/S | AEOI | Mode |

- SFNM: 1=Special Fully Nested Mode, 0=Not Special Fully Nested Mode
- BUF: 1 =Buffered Mode
- M/S: 1=Master, 0=Slave
- AEOI: 1=Auto End of Interrupt, 0=Normal
- Mode: 0=8085, 1=8086

- Buffered Mode: In buffered mode, SP/EN work as output Enable
- Specially Fully Nested Mode: Master can receive high priority interrupt from slave
- EOI: Auto EOI will reset ISR bit after receiving INTA, else command word is used to end interrupt

# Programming 8259A

- OCWs are sent to PIC to direct its operations

**OCW1**

| A0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 1 | M7 | M6 | M5 | M4 | M3 | M2 | M1 | M0 |

- IRn is masked by setting Mn to 1; mask cleared by setting Mn to 0 (n=0..7)

**OCW2**

| A0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 1 | R | SL | EOI | 0 | 0 | L3 | L2 | L1 |

  - Used when AEOI is 0 in ICW4
  - Used to set/reset ISR register bits
  - Priorities can be modified/rotated from lowest to highest with help of OCW2

# Programming 8259A

| | R | SL | EOI | Action |
|---|---|---|---|---|
| EOI | 0 | 0 | 1 | Non specific EOI (L3L2L1=000) |
| | 0 | 1 | 1 | Specific EOI command (Interrupt to clear given by L3L2L1) |
| Auto rotation of priorities (L3L2L1=000) | 1 | 0 | 1 | Rotate priorities on non-specific EOI |
| | 1 | 0 | 0 | Rotate priorities in auto EOI mode set |
| | 0 | 0 | 0 | Rotate priorities in auto EOI mode clear |
| Specific rotation of priorities (Lowest priority ISR=L3L2L1) | 1 | 1 | 1 | Rotate priority on specific EOI command (resets current ISR bit) |
| | 1 | 1 | 0 | Set priority (does not reset current ISR bit) |
| | 0 | 1 | 0 | No operation |

**OCW3**

| A0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | ESMM | SMM | 0 | 1 | P | RR | ISR |

- RR : Request Register
- ISR: Interrupt service Register

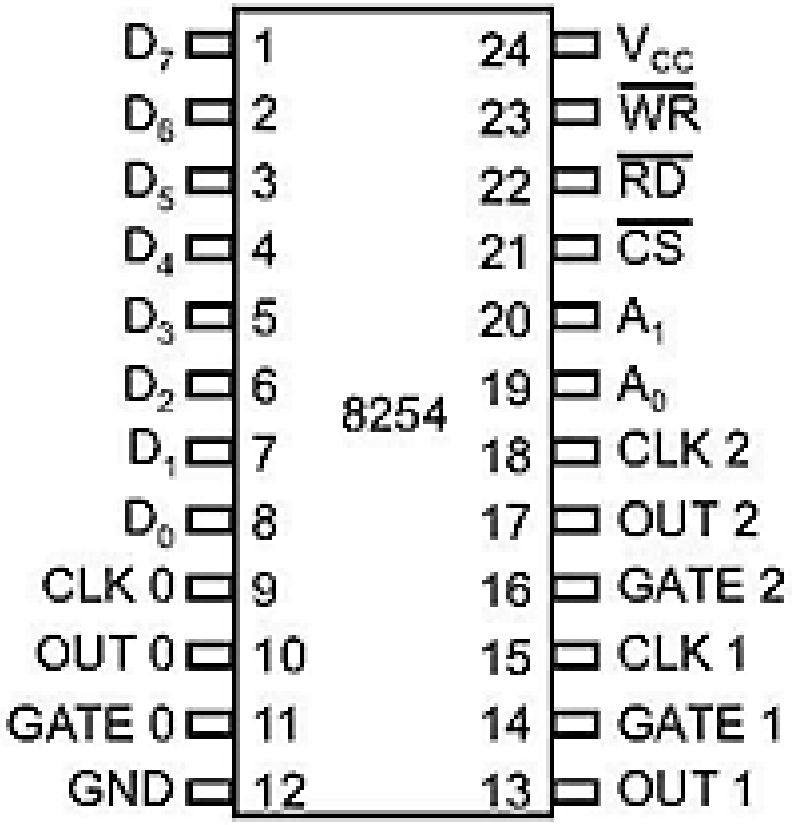
# Programming 8259A

- P: 1 = Poll command, 1= No poll command

| ESMM | SMM | Effect |
|---|---|---|
| 0 | X | No effect |
| 1 | 0 | Reset special mask |
| 1 | 1 | Set special mask |

# Intel 8254 Programmable Interval Timer

- Used to generate CLK signals

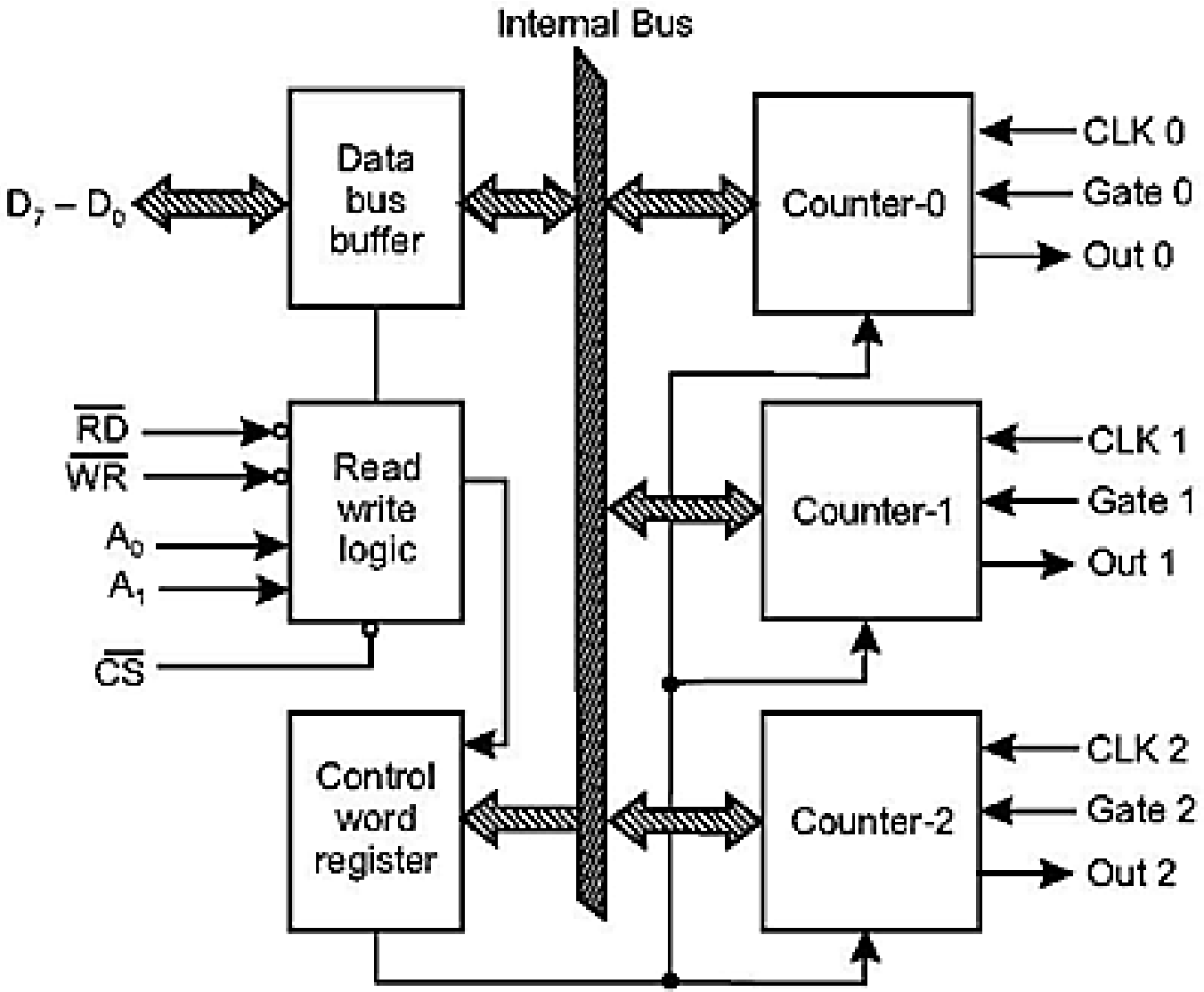


# Intel 8254 Programmable Interval Timer

- Processor receives interrupt
  - Address lines are used to select particular component

| A1 | A0 | Selection |
|---|---|---|
| 0 | 0 | Counter 0 |
| 0 | 1 | Counter 1 |
| 1 | 0 | Counter 2 |
| 1 | 1 | Control Word Register |

# Intel 8254 Programmable Interval Timer



# Intel 8254 Programmable Interval Timer

- 8254 can be programmed by sending command word

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| SC1 | SC0 | RW1 | RW0 | M2 | M1 | M0 | BCD |

  - BCD: 0=Binary counter, 1 = Binary Coded decimal

| SC1 | SC0 | |
|---|---|---|
| 0 | 0 | Counter 0 |
| 0 | 1 | Counter 1 |
| 1 | 0 | Counter 2 |
| 1 | 1 | Read-back command |

| RW1 | RW0 | |
|---|---|---|
| 0 | 0 | Counter Latch Command |
| 0 | 1 | R/W LSByte |
| 1 | 0 | R/W MSByte |
| 1 | 1 | R/W LSbyte then MSbyte |

# Intel 8254 Programmable Interval Timer

- Counter Read methods:
- A simple READ operation:
  - a. Select the Counter with the A1,A0 inputs.
    b. Inhibit the CLK of the selected counter by using either the GATE input or external logic. (The CLK must be inhibited or the count may be in the process of changing when it is read, giving an undefined result.)
    c. Note that stopping the CLK stops the count.
- Counter Latch Command (does not disturb the count in progress):
  - a. It is written to the Control Word Register like a Control Word, but two bits (D5,D4) distinguish this command from a Control Word.
    b. The selected Counter's latches count at the time the Counter Latch Command is received.
    c. The count is held in the latch until it is read by the CPU.
    d. The count is then unlatched automatically.
- Read-Back Command:
  - a. This command allows the user to check the count value, programmed Mode, and current states of the OUT pin and Null Count flag of the selected counter(s).
    b. This command is similar to several Counter Latch Commands, one for each counter latched.

# Intel 8254 Programmable Interval Timer

| M2 | M1 | M0 | Mode |
|---|---|---|---|
| 0 | 0 | 0 | Mode 0 |
| 0 | 0 | 1 | Mode 1 |
| X | 1 | 0 | Mode 2 |
| X | 1 | 1 | Mode 3 |
| 1 | 0 | 0 | Mode 4 |
| 1 | 0 | 1 | Mode 5 |

- **MODE 0: Interrupt on terminal count**
  1. Event counting.
  2. After the Control Word is written, OUT is initially low and remains low.
  3. When the counter reaches zero. OUT then goes high and remains high until a new count or a new Mode 0 Control Word is written into the Counter.

# Intel 8254 Programmable Interval Timer

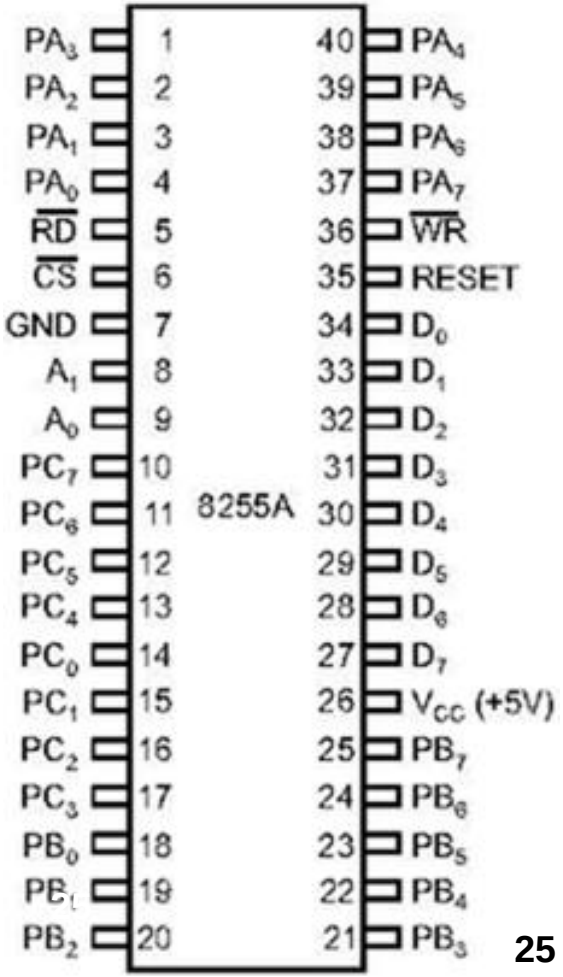
- **MODE 1: Hardware re-triggerable one-shot**
  1. OUT will be initially high. OUT will go low on the CLK pulse following a trigger to begin the one-shot pulse, and remain low until the Counter reaches zero.
  2. OUT will then go high and remain high until the CLK pulse after the next trigger.
- **MODE 2: Rate generator**
  1. Functions like a divide-by-N counter and used to generate a Real Time Clock interrupt.
  2. OUT will initially be high.
  3. When the initial count has decremented to one, OUT goes low for one CLK pulse.
  4. Out then goes high again, the Counter reloads the initial count and the process is repeated.
  5. MODE 2 is periodic. The same sequence is repeated indefinitely.
- **MODE 3: Square wave mode**
  1. Typically used for baud rate generation.
  2. Out will initially be high.
  3. When half the initial count is expired, OUT goes low for the remainder of the count.
  4. MODE 3 is periodic. The same sequence is repeated indefinitely.

# Intel 8254 Programmable Interval Timer

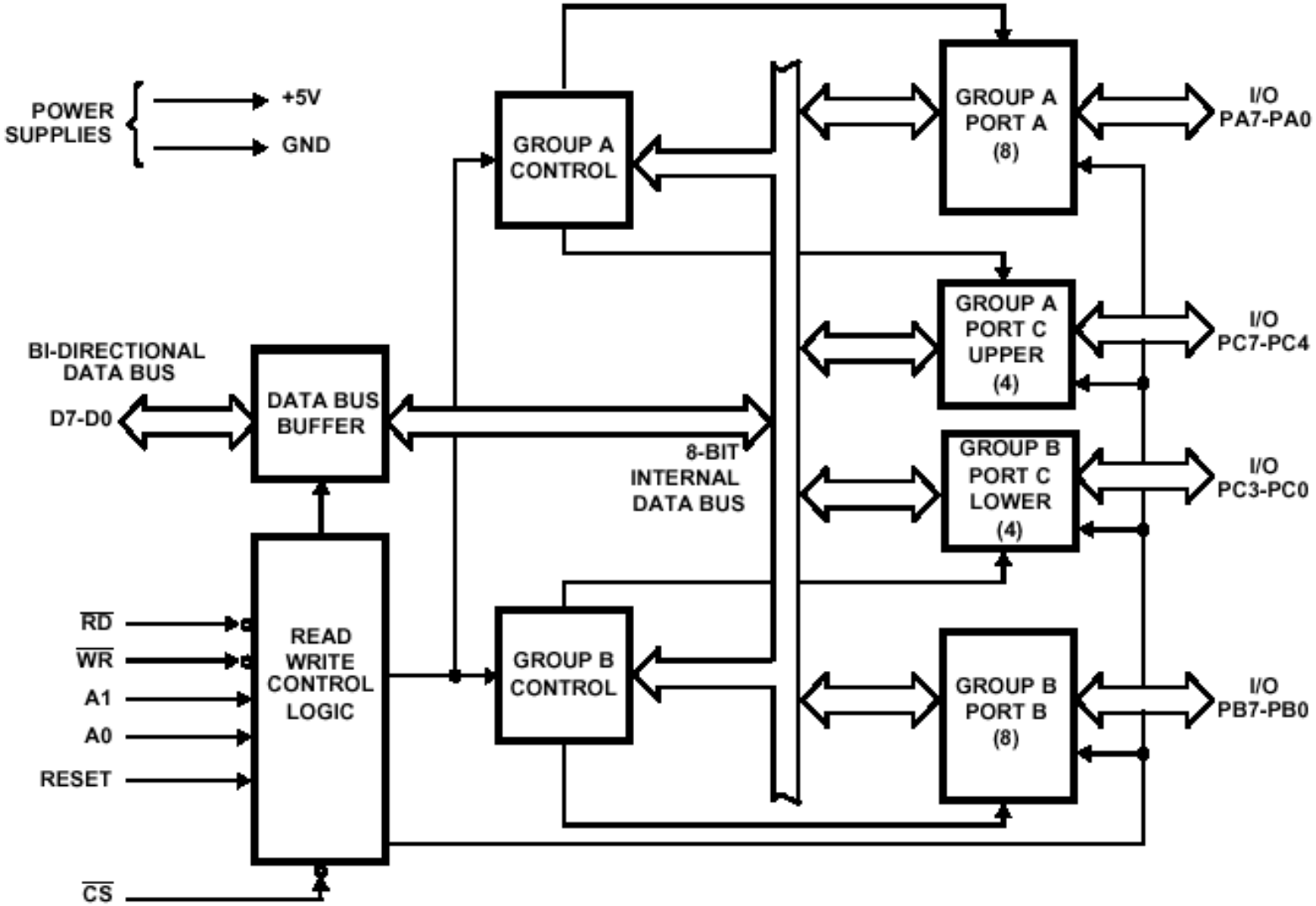
- **MODE 4: Software triggered strobe**
  1. OUT will initially be high.
  2. When the initial count expires, OUT will go low for one CLK pulse and then go high again.
  3. The counting sequence is "triggered" by writing the initial count.
  4. The Counter is loaded on the next CLK pulse following writing a Control Word and initial count.
- **MODE 5: Hardware triggered strobe (re-triggerable)**
  1. OUT will initially be high.
  2. Counting is triggered by a rising edge of GATE.
  3. When the initial count expires, OUT will go low for one CLK pulse and then go high again.
  4. The difference between MODE 4 and MODE 5 is that in MODE 5 the count will not be loaded until the CLK pulse after a trigger.

# Intel 8255 Programmable Peripheral Interface

- 8255 is used to provide connection and parallel communication
- Peripheral devices requires handshaking
  - Single Handshake
  - Double Handshake
- 8255 have three ports, which can be used individually or combinable

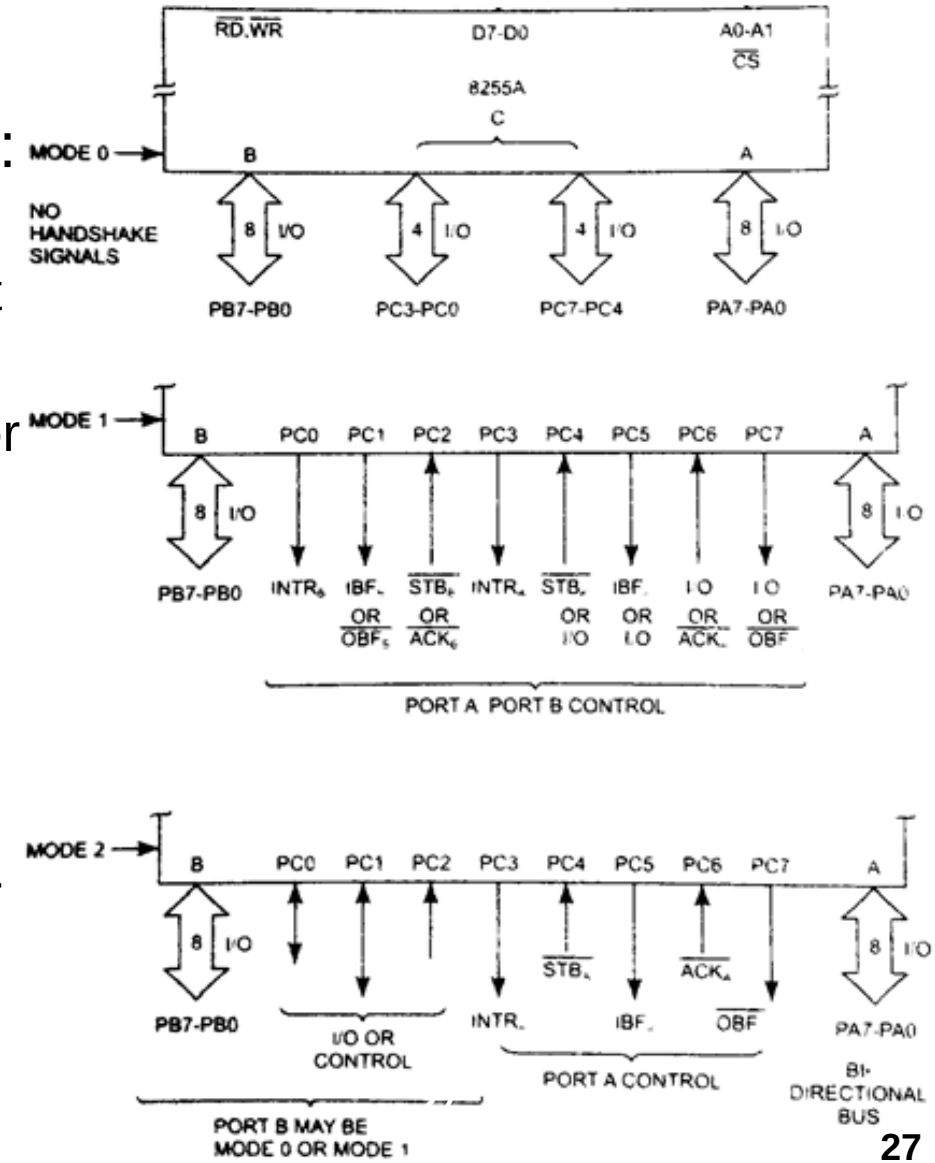


# Intel 8255 Programmable Peripheral Interface



# Intel 8255 Programmable Peripheral Interface

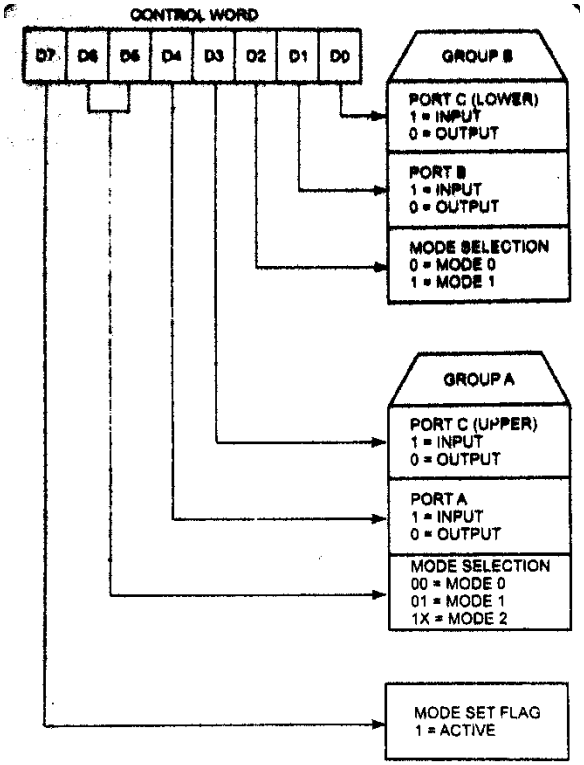
- 8255 provides three modes:
  - Mode 0: When all three ports are used for simple I/O without handshaking
  - Mode 1: When a port ( A and/or B) is used in single handshaking mode. Port C is used to carry handshaking data.
  - Mode 2: When port A is used for bidirectional handshaking. Port C is used to carry data for handshaking



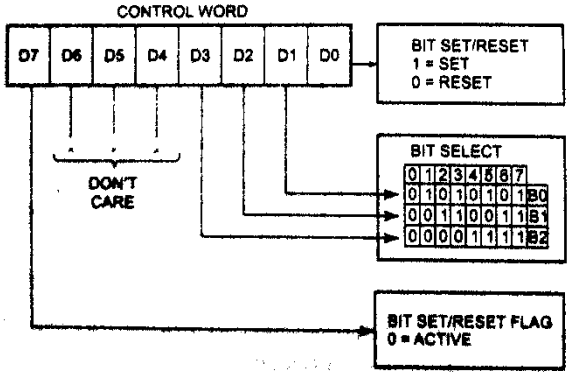
# Intel 8255 Programmable Peripheral Interface

- 8255 can be programmed by sending command words
- 8255 has two command words

I/O Control Word

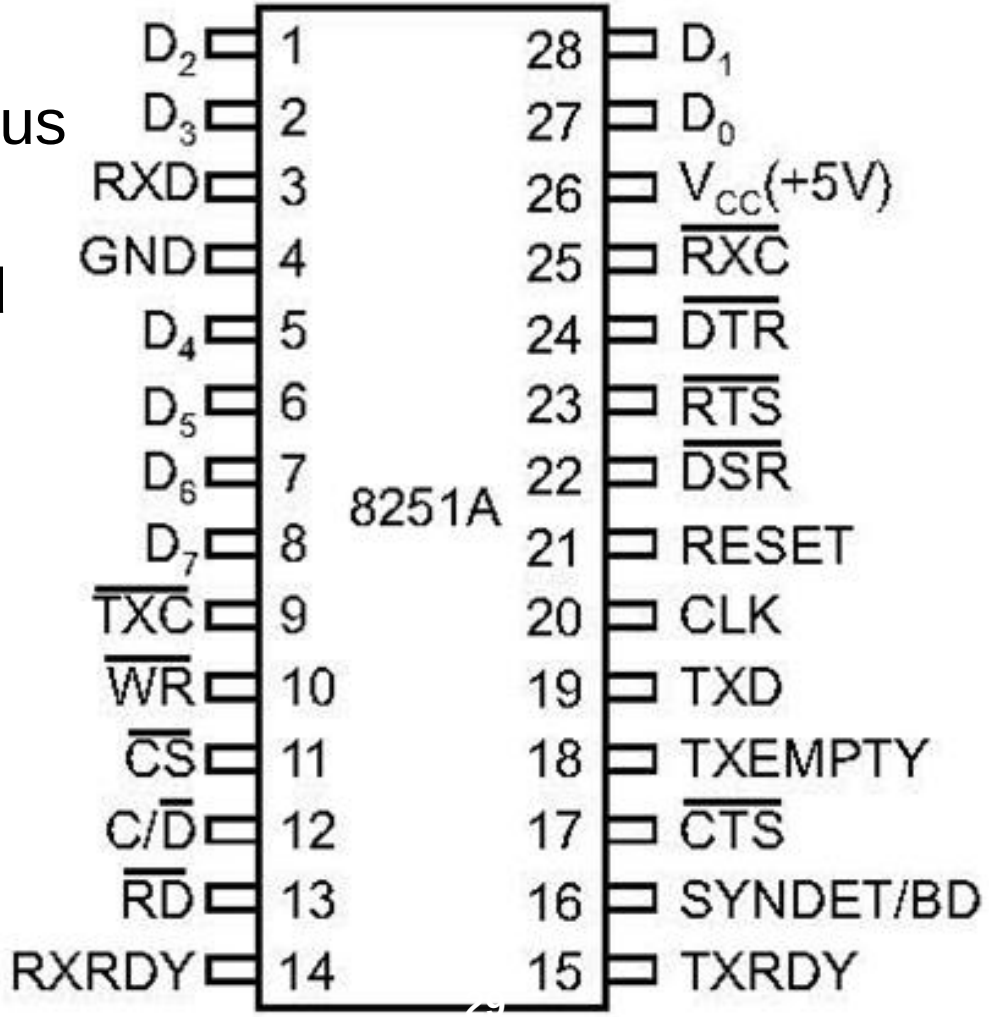



BSR Control Word



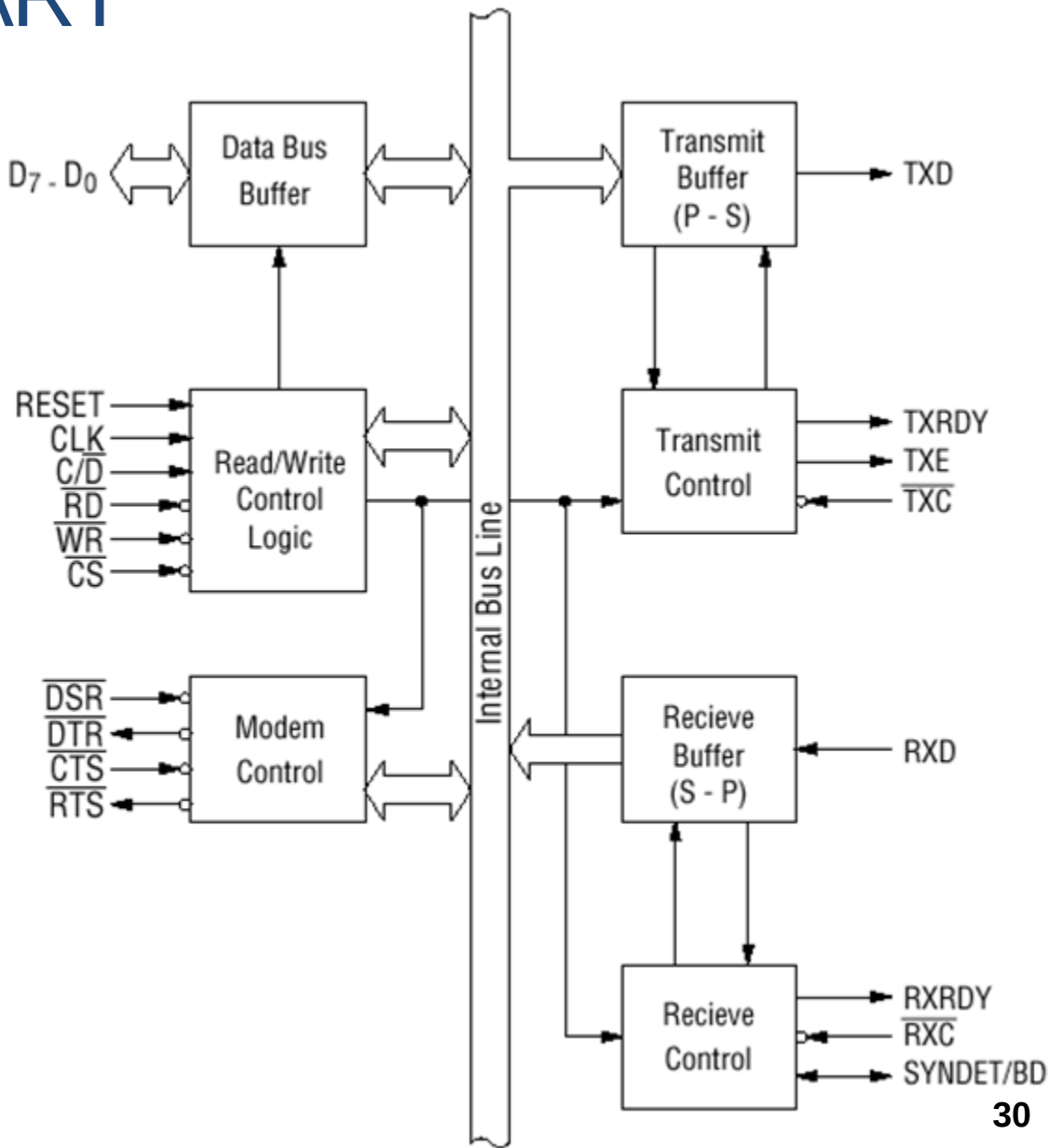
# Intel 8251A USART

- 8251A Universal Synchronous/Asynchronous Receiver/Transmitter
- 8251A component is used to peripherals with serial communication
- 28pin Module



# Intel 8251A USART

- 8251A have 5 main components:
  - Modem Control
  - Read/Write Logic
  - Data Bus Buffer
  - Transmitter
  - Receiver



# Intel 8251A USART

- 8251A can be programmed with help of command words

## Mode Instruction (Async)

| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|---|---|
| $S_1$ | $S_1$ | EP | PEN | $L_2$ | $L_1$ | $B_2$ | $B_1$ |

Baud Rate Factor

| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| SYNC | 1× | 16× | 64× |

Charactor Length

| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 5 bits | 6 bits | 7 bits | 8 bits |

Parity Check

| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| Disable | Odd Parity | Disable | Even Parity |

Stop bit Length

| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| Inhabit | 1 bit | 1.5 bits | 2 bits |

## Mode Instruction (Sync)

| $D_7$ | $D_6$ | $D_5$ | $D_4$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|---|---|
| SCS | ESD | EP | PEN | $L_2$ | $L_1$ | 0 | 0 |

Charactor Length

| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 5 bits | 6 bits | 7 bits | 8 bits |

Parity

| | | | |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| Disable | Odd Parity | Disable | Even Parity |

Synchronous Mode

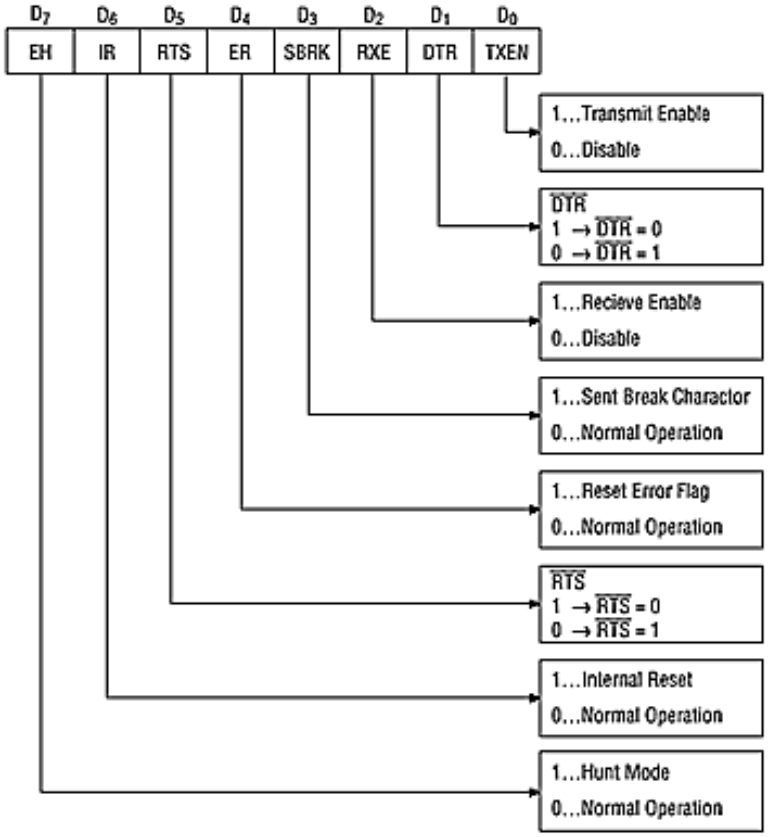
| 0 | 1 |
|---|---|
| Internal Synchronization | External Synchronization |

Number of Synchronous Charactors

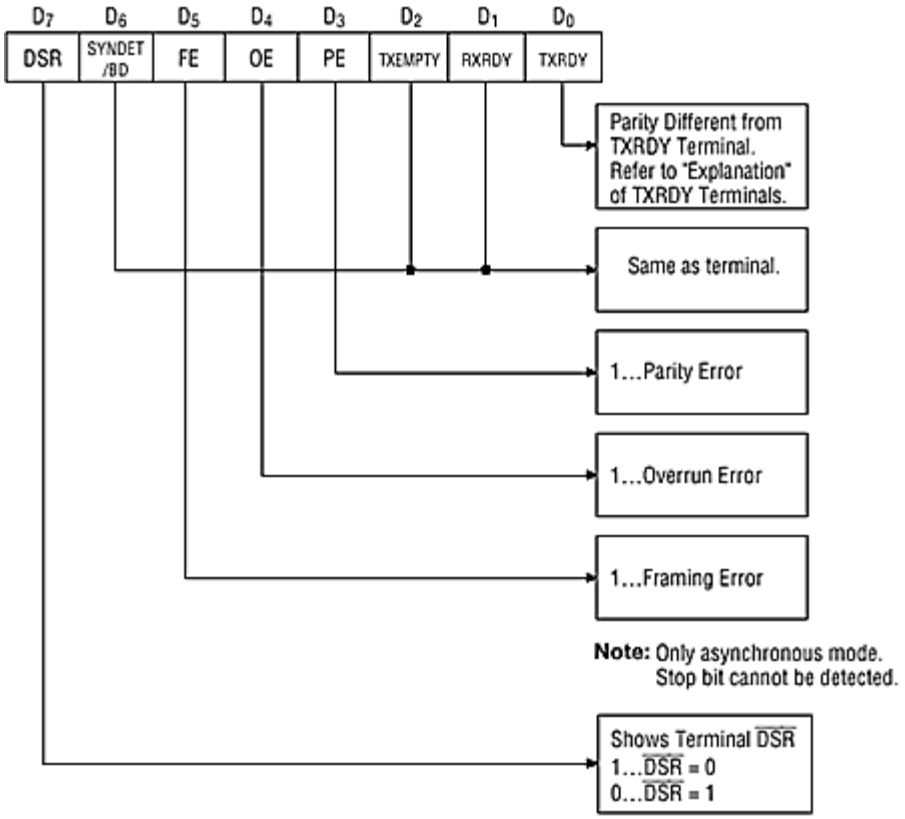
| 0 | 1 |
|---|---|
| 2 Charactors | 1 Charactor |

# Intel 8251A USART

## Command Word

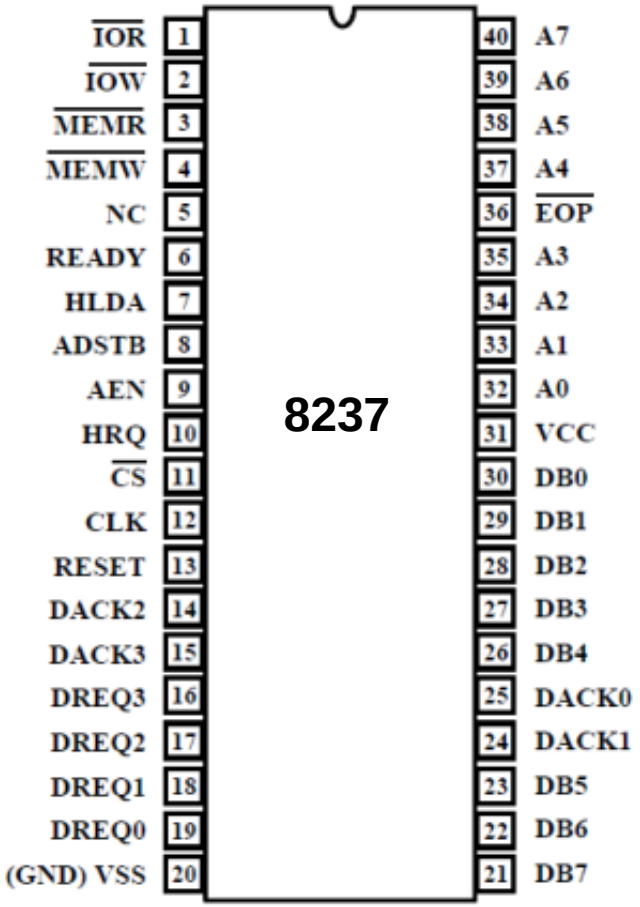



## Status Word

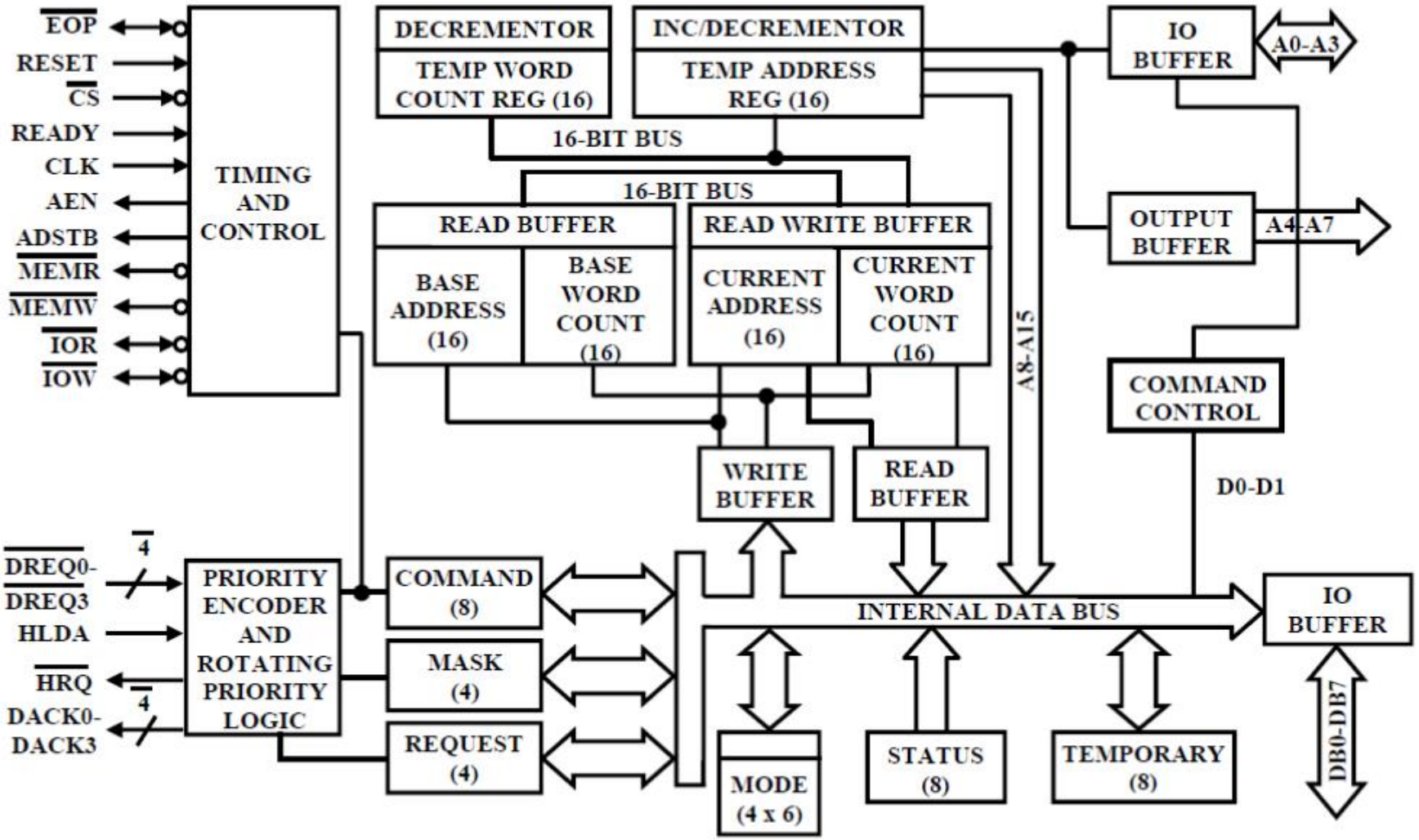


# Intel 8237A Programmable DMA Controller

- Direct Memory Access technique is used to transfer memory contents from source to destination
- 40Pin DIP
- Four channel DMA controller
- Multiple DMAs can be used in cascading mode
- 1.6M byte per second



# Intel 8237A Programmable DMA Controller



# Intel 8237A Programmable DMA Controller

- A0 to A3 pins along with other control pins are used to distinguish the operation

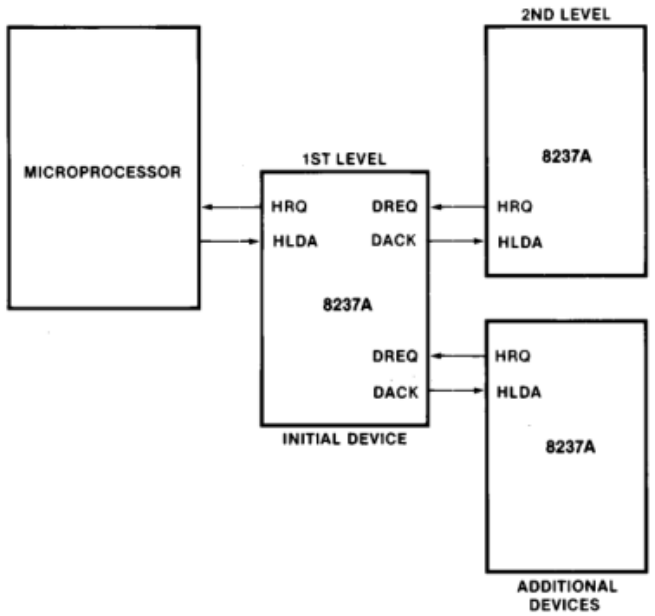
| Register | Operation | Signals | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | $\overline{CS}$ | $\overline{IOR}$ | $\overline{IOW}$ | A3 | A2 | A1 | A0 |
| Command | Write | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| Mode | Write | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| Request | Write | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| Mask | Set/Reset | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| Mask | Write | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| Temporary | Read | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| Status | Read | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

- Same pins with CS can be used to access particular channel
- Differentiated with A1 and A2 pins

| Channel | Register | Operation | Signals | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | $\overline{CS}$ | $\overline{IOR}$ | $\overline{IOW}$ | A3 | A2 | A1 | A0 |
| 0 | Base and Current Address | Write | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | Current Address | Read | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | Base and Current Word Count | Write | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| | Current Word Count | Read | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

# Intel 8237A Programmable DMA Controller

- Modes
  - Idle Mode: When 8237A is waiting for DREQ signal and CS signal
  - Active Mode
    - Single transfer mode: One byte transfer held until word count is resettled (Terminal Count)
    - Block transfer mode: Data transfer occurs until TC or EOP and can be deactivated by DACK. Auto initialization can be programmed
    - Demand transfer mode: Data transfer occurs until TC or EOP and can be deactivated when DREQ goes low. Auto initialization can be programmed
  - Cascaded Mode
    - Multiple 8237As can be connected
    - 1st Level is considered as master and 2nd level as slave
    - In cascading mode, all active modes can be used

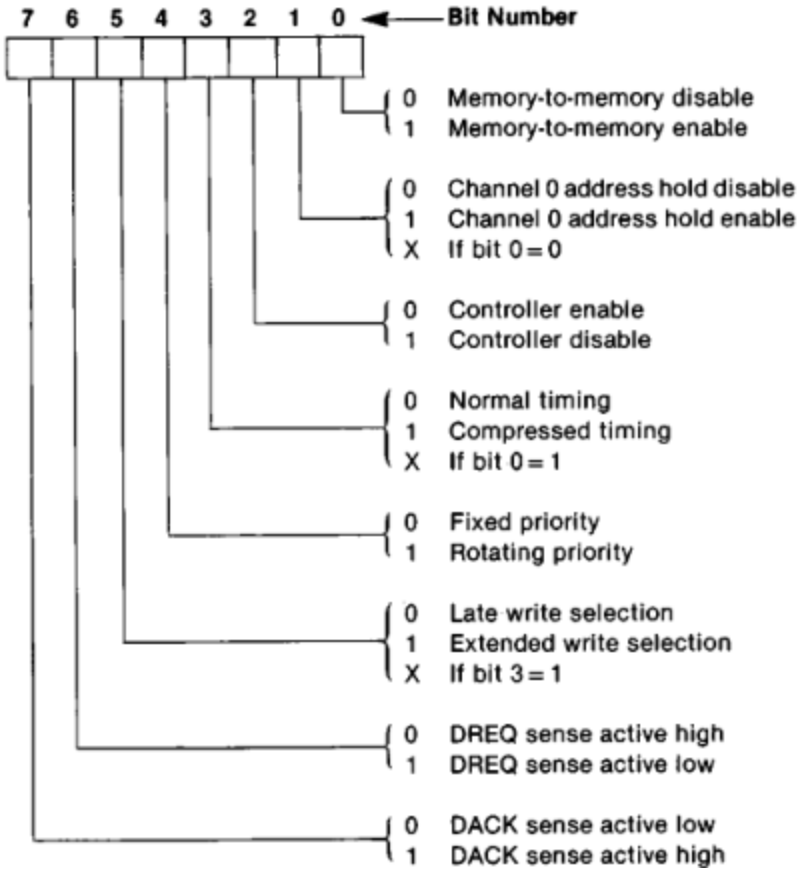


# Intel 8237A Programmable DMA Controller

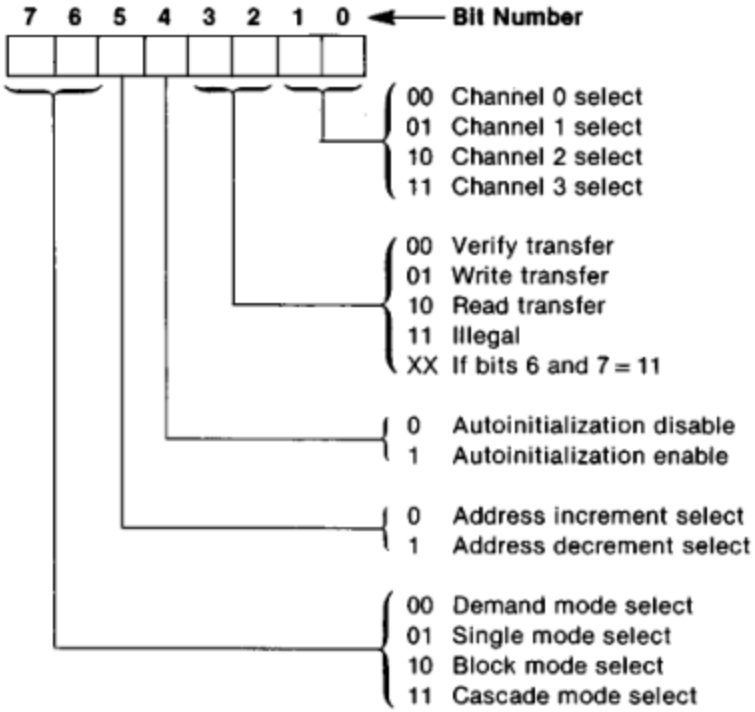
- Transfer Types
  - Memory to Memory: 8237A uses Channel 0 for source and 1 for destination. The 8237A works in block transfer mode.
  - Auto initialization: Contents of CAR & CWCR are automatically restored from BAR & BWCR after EOP and when DREQ is received without CPU involvement. The mask bit is not altered. Contents of BAR & BWCR should be modified accordingly
  - Priority: Two priorities are available in 8237A:
    - Fixed: Priorities from DREQ0 to DREQ3
    - Rotating: Priorities are rotated bit wise
  - Compressed Timing: Read/Write time cycle is made double to increase throughput
  - Address generation: 8237A generates address from A0 to A7 with help of AEN connected to latch. The upper part of address A8-A15 are sent from data lines using ADSTB

# Intel 8237A Programmable DMA Controller

## Command Word

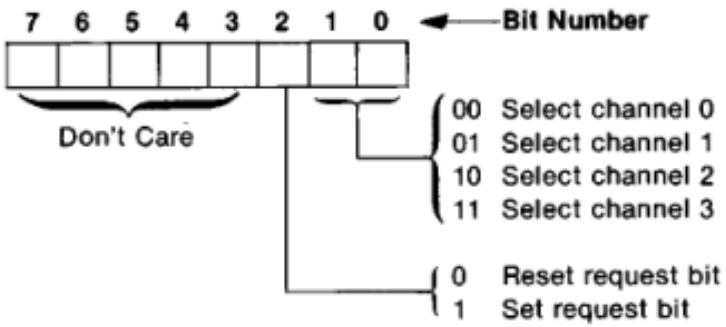



## Mode Word

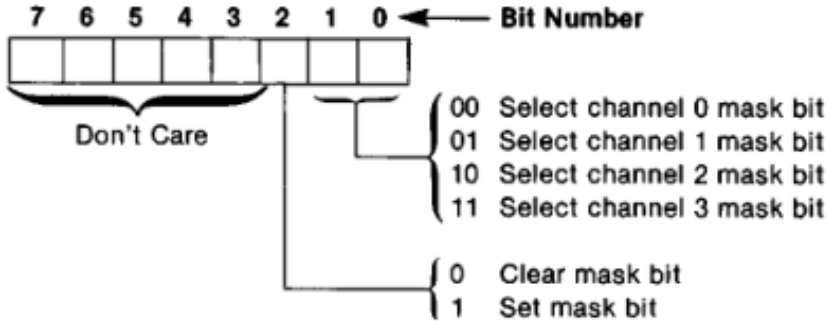


# Intel 8237A Programmable DMA Controller

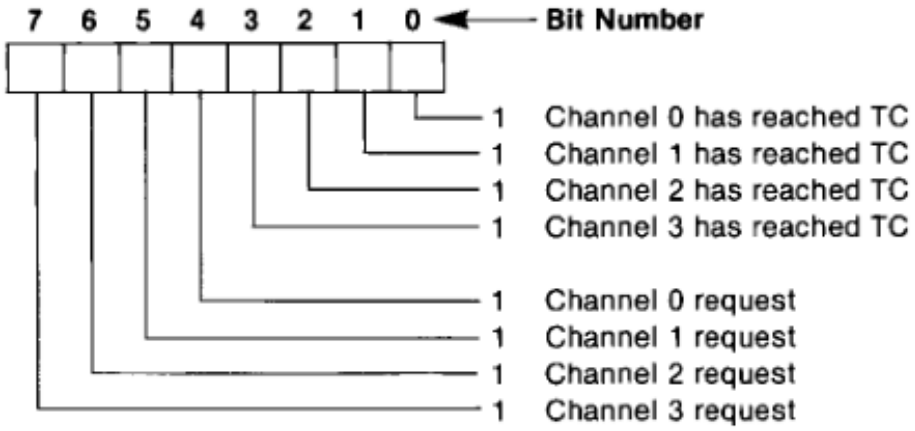
## Request Word



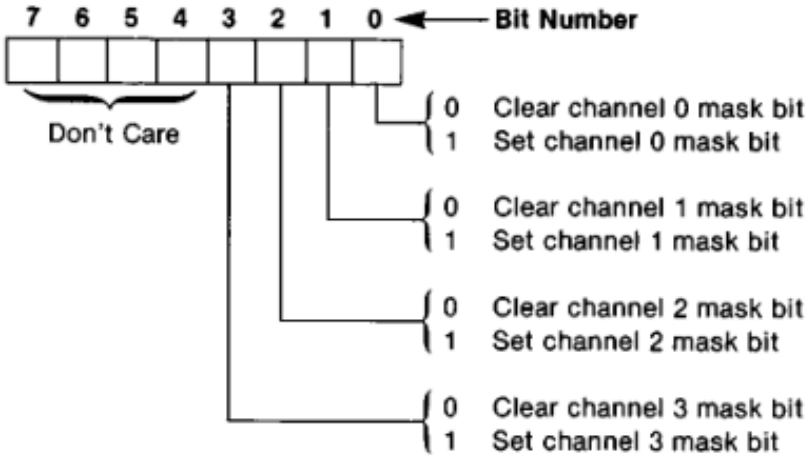

## Mask Set/Reset Word




## Status Word




## Mask Set Word



# Questions