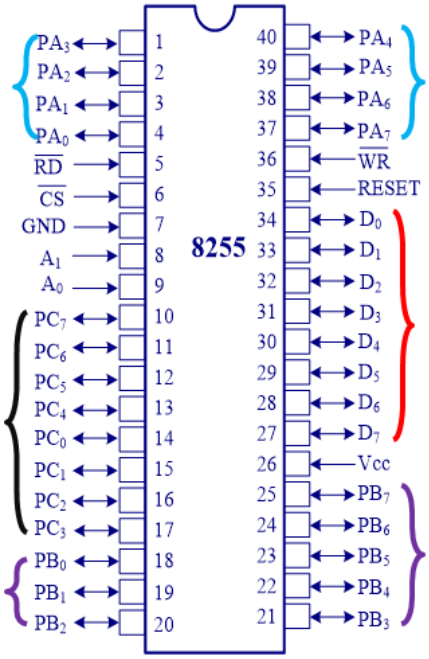
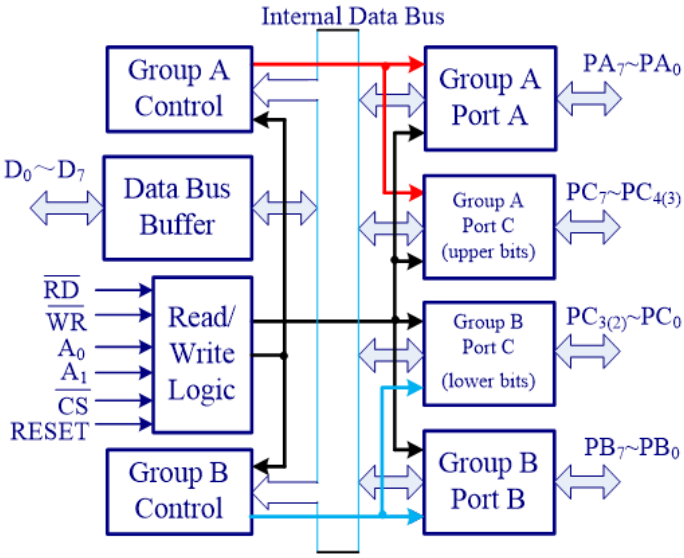
7 8255 PPI Chip

PPI: Programmable Parallel Interface (I/O module)





Data Ports

- Port A ($PA_0 \sim PA_7$): can be programmed all as input/output;
- Port B: can be programmed all as input/output;
- Port C: can be split into two separate parts *PCU* and *PCL*; any bit can be programmed individually.

Control Registers (CR): A 8-bit internal register, used to setup the chip, selected when $A_0 = 1, A_1 = 1$.

Groups of 8255 PPI Chip

- Group A: Port A and the upper bits of Port C.
- Group B: Port B and the lower bits of Port C.

Data Bus Buffer

- An interface between CPU and 8255;
- Bidirectional, tri-state, 8-bit.

Read/Write Control Logic

~CS	A_1	A_0	~RD	~WR	Function
0	0	0	0	1	PA->Data bus
0	0	1	0	1	PB->Data bus
0	1	0	0	1	PC->Data bus
0	0	0	1	0	Data bus->PA
0	0	1	1	0	Data bus->PB
0	1	0	1	0	Data bus->PC
0	1	1	1	0	Data bus->CR
1	×	×	1	1	D_0~D_7 in float

- Internal and external control signals.
- $RESET$: high-active, clear the control register, all ports are set as input port.

- $\overline{CS}$, $\overline{RD}$, $\overline{WR}$;
- A_1, A_0: port selection signals.

Operation Modes

- **Input/Output (IO) modes**: only can send data in the unit of byte.
  - Mode 0: simple I/O mode:
    - **PA, PB, PC**: PCU{PC4 ~ PC7}, PCL{PC0 ~ PC3}.
    - No *handshaking*, which is negotiation between two entitles before communication.
    - Each port can be programmed as input/output port.
  - Mode 1:
    - **PA, PB** can be used as input/output ports with *handshaking*.
    - **PC**: PCU{PC3 ~ PC7}, PCL{PC0 ~ PC2} are used as handshake lines for PA and PB, respectively.
  - Mode 2:
    - Only **PA** can be used for bidirectional handshake data transfer;
    - PCU{PC3 ~ PC7} are used as handshake lines for PA.
- **Bit Set/Reset (BSR) mode**
  - Only **PC** can be used as output port.
  - Each line of PC can be set/reset individually.

Control Register & Operation Modes: Mode selection word. Use the highest bit to distinguish two main modes.

- **IO modes**: $D_7D_6D_5D_4D_3D_2D_1D_0$ is `1*******`;
  - $D_6D_5D_4D_3$ are for Group A.
    - D_6D_5 is mode selection bits: `00` is mode 0, `01` is mode 1 and `1*` is mode 2.
    - D_4 is the selection bit for Port A (1 - input and 0 - output);
    - D_3 is the selection bit for PCU (1 - input and 0 - output);
  - $D_2D_1D_0$ are for Group B.
    - D_2 is mode selection bit: `0` is mode 0 and `1` is mode 1.
    - D_1 is the selection bit for Port B (1 - input and 0 - output);
    - D_0 is the selection bit for PCL (1 - input and 0 - output).
- **BSR mode**: $D_7D_6D_5D_4D_3D_2D_1D_0$ is `0*******`.
  - $D_6D_5D_4$ are not used;
  - $D_3D_2D_1$ indicate which line in PC: 000 - PC0, 001 - PC1, ..., 111 - PC7;
  - D_0 is used to set/reset (1/0).

IO Mode

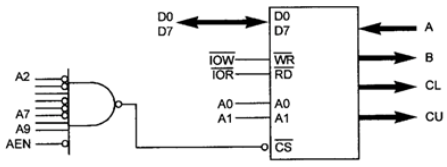
- **Mode 0**: (Simple I/O) for simple input/output scenario.
  - CPU directly read from or write to a port using `IN` and `OUT` instructions.
  - Input data are not latched, but output data are latched.

> [*Example*] Mode 0 Operations.

The 8255 shown in Figure 11-13 is configured as follows: port A as input, B as output, and all the bits of port C as output.
(a) Find the port addresses assigned to A, B, C, and the control register.
(b) Find the control byte (word) for this configuration.
(c) Program the ports to input data from port A and send it to both ports B and C.

Solution:
(a) The port addresses are as follows:

$\overline{CS}$	A1	A0	Address	Port
11 0001 00	0	0	310H	Port A
11 0001 00	0	1	311H	Port B
11 0001 00	1	0	312H	Port C
11 0001 00	1	1	313H	Control register



(b) The control word is 90H, or 1001 0000.

(c) One version of the program is as follows:

```
MOV   AL,90H      ;control byte PA=in, PB=out, PC=out
MOV   DX,313H     ;load control reg address
OUT   DX,AL       ;send it to control register
MOV   DX,310H     ;load PA address
IN    AL,DX       ;get the data from PA
MOV   DX,311H     ;load PB address
OUT   DX,AL       ;send it to PB
MOV   DX,312H     ;load PC address
OUT   DX,AL       ;and to PC
```

The control word is generated as follows.

$D_7 = 1$: I/O Mode;

$D_6 D_5 = 00$: mode 0 for port A and PCU;

$D_4 = 1$: Port A Input;

$D_3 = 0$: PCU Output;

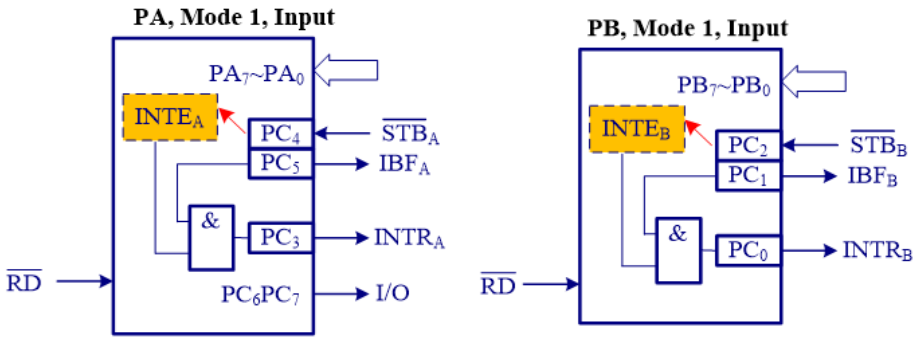
$D_2 = 0$: mode 0 for port B and PCL;

$D_1 = 0$: Port B Output;

$D_0 = 0$: PCL Output.

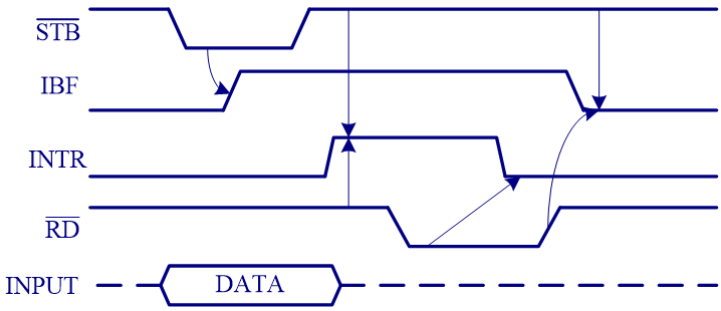
- **Mode 1**: (Strobe I/O) for handshake input/output scenario. Both input data and output data are latched.

  Mode 1 As Input Ports



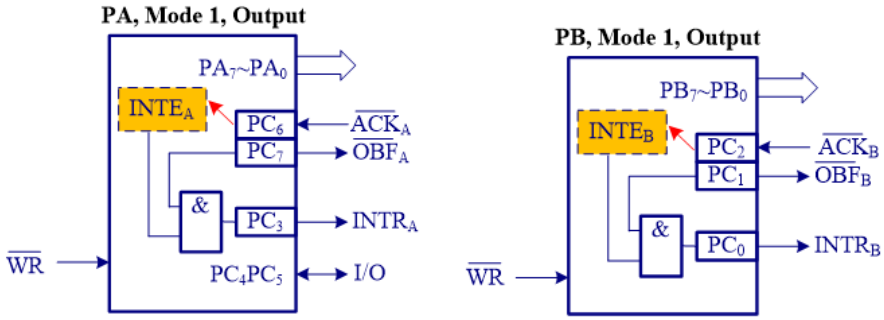
  - PC_4, PC_5, PC_3 and $PC_2 \sim PC_0$ are used as handshake lines for PA and PB respectively.
    - $\overline{STB}$: the strobe input signal from input device loads data into the port latch (high - not load, low - load);
    - IBF: the Input Buffer Full output signal to the device indicates that the input latch contains information (can also be used for programmed I/O) (high - full, low - not full).
    - $INTR$: the interrupt request is an output to CPU that requests an interrupts (for interrupted I/O).
  - PC_6 and PC_7 can be used as separate I/O lines for any purpose (controlled by D_3).

- $INTE$ is the interrupt enable signal. It is neither an input nor an output; it is an internal bit programmed via the PC_4 (port A) or PC_2 (port B) with BSR mode before configuration; (1 - allowed, 0 - forbidden).
- **Timing**

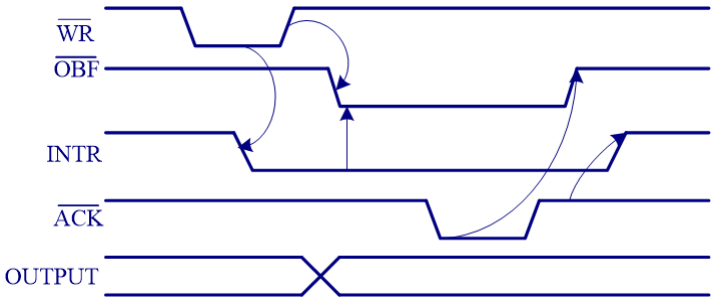


  - Input device first put data on $PA_0 \sim PA_7$, then activate $\overline{STB_A}$ (set to 0),after that data is latched in Port A and $\overline{STB_A}$ is high;
  - 8255 activates IBF_A which indicates that device that the input latch contains information but CPU has not taken it yet. So device cannot send new data until IBF_A is cleared;
  - When $IBF_A, \overline{STB_A}, INTE_A$ are all high, 8255 activates $INTR_A$ to inform CPU to take data in PA by interruption;
  - CPU responds to the interruption and read in data from PA; the $\overline{RD}$ signal will clear $INTR_A$ signal;
  - After CPU finishes reading data from PA (i.e., $\overline{RD}$ signal goes high), the IBF_A signal is cleared.

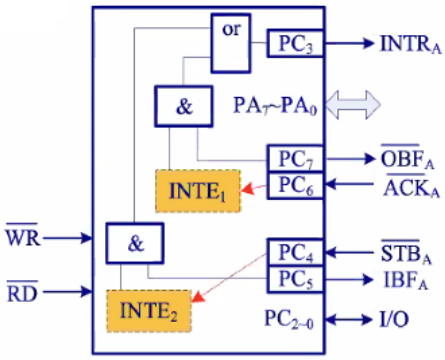
Mode 1 As Output Ports



- PC_6, PC_7, PC_3 and $PC_2 \sim PC_0$ are used as handshake lines for PA and PB respectively.
  - $\overline{ACK}$: the acknowledge input signal indicates that the external device has taken the data (high - not taken, low - taken);
  - $\overline{OBF}$: the Output Buffer Full is an output signal that indicates the data has been latched in the port (high - not full, low - full);
  - $INTR$: Interrupt request is an output to CPU that requests an interrupts with BSR mode before configuration; (1 - allowed, 0 - forbidden).
- PC_4 and PC_5 can be used as separate I/O lines for any purpose (controlled by D_3)
- $INTE$ is the interrupt enable signal. It is neither an input nor an output; it is an internal bit programmed via the PC_6 (port A) or PC_2 (port B) with BSR mode before configuration; (1 - allowed, 0 - forbidden).
- **Timing**



    - If $INTR_A$ active, CPU responds to the interruption and writes data to the PA and clears the $INTR_A$ signal ($INTR_A$ high means ready to receive data).
    - When data has been latched in PA, 8255 activates $\overline{OBF_A}$ which informs the output device to pick up data;
    - After the output device has taken the data, it sends $\overline{ACK_A}$ signal to 8255 which indicates that the device has received the data, and also makes $\overline{OBF_A}$ go high, indicating CPU can write new data to 8255;
    - When $\overline{OBF_A}$, $\overline{ACK_A}$ and $INTE_A$ are all high, 8255 sends an $INTR_A$ to inform CPU to write new data to PA by interruption.
- **Mode 2**: (Bidirectional Bus) for bidirectional handshake input/output scenario. Both input data and output data are latched.
    - Only PA can be used as *both input and output port*;
    - $PCU = PC_3 \sim PC_7$, used as handshake lines for PA;
        - $PC_3 : INTR_A$;
        - $PC_4 : \overline{STB}_A$;
        - $PC_5 : IBF_A$;
        - $PC_6 : \overline{ACK}_A$;
        - $PC_7 : \overline{OBF}_A$.
    - Both input and output data are latched.



    Control Code: $D_7D_6D_5D_4D_3D_2D_1D_0$ is `011xxxxxx`. The last three bits is used for Port B control.

- The fundamental difference among mode 0, 1, 2 is polling v.s. interruption.