There are some non-obvious features of the 8255 PPI which you should be aware of.

8255 PPI USE PROBLEMS

When I/O Port A or B is assigned as an output port, individual port pins may output high or low signals; this is accomplished by writing data to the I/O port before assigning it as an output port. When you assign either or both halves of I/O Port C as an output port, all signals will initially be low.

When operating in Mode 0, beware of assigning upper and lower halves of I/O Port C to input and output. Some users have reported that when you read from I/O Port C you may alter output signal levels; when you write to I/O Port C you may write into the input signal's buffer, modifying any bit settings created by input signals. This is a problem that does not always appear.

In Modes 1 and 2 you must use the Control code to write to I/O Port C, one bit at a time. You cannot write an 8-bit pattern to I/O Port C. Also, in Modes 1 and 2, all control signals must be initialized by writing to the appropriate bit of I/O Port C using the appropriate Control code.

8255 PPI interrupt logic must be enabled. In Modes 1 and 2 observe that interrupt requests may be output via INTRA and INTRB (pins 3 and 0 of I/O Port C, respectively). These interrupt requests will only occur if you first output a high level to these bits of I/O Port C using the appropriate Control codes.

THE 8212 8-BIT INPUT/OUTPUT PORT

This is a simple, nonprogrammable, yet versatile and economic I/O device, capable of providing total I/O interface logic in small microcomputer systems; in larger microcomputer systems consider the 8212 I/O port for specific local I/O needs.

This device has a data latch, buffer and interrupt logic. The data latch consists of eight D-type flip-flops whose Q outputs are connected to tristate, non-inverting output buffers. In order to make effective use of the 8212 I/O port, it is important to understand the logic of this device; therefore it is illustrated in Figure 4-47. We will refer to this figure for clarification when describing individual signals.

In order to understand the latching and buffering of the 8212 I/O port, **note that the eight flip-flops are sensitive to the level of the C input and not to signal transitions.** When the clock input C is high, the Q outputs will track the D inputs. When the clock inputs C are low, the Q outputs hold their prior levels and are disconnected from the D inputs.

Interrupt logic provided by the 8212 I/O port is functionally similar to the handshaking logic described for Modes 1 and 2 of 8255 PPI operation.

Figure 4-45, which defines 8255 PPI logic bounds, also illustrates that portion of our general microcomputer systems which can be implemented using an 8212 I/O port.

8212 I/O PORT PINS AND SIGNALS

8212 I/O port pins and signals are illustrated in Figure 4-48. We will describe these pins and signals with reference to Figure 4-47.

DI0 - DI7 are data input pins. They are connected to the eight data inputs of the D-type flip-flops within the 8212 I/O port. The Q outputs of these flip-flops are connected to tristate buffers, which in turn output data via pins DO0 - DO7.

Data flow is controlled by the mode select (MD), strobe (STB), and device select logic.

Before this device can be selected, a low input must occur at $\overline{DS1}$ and a high input must occur at DS2.

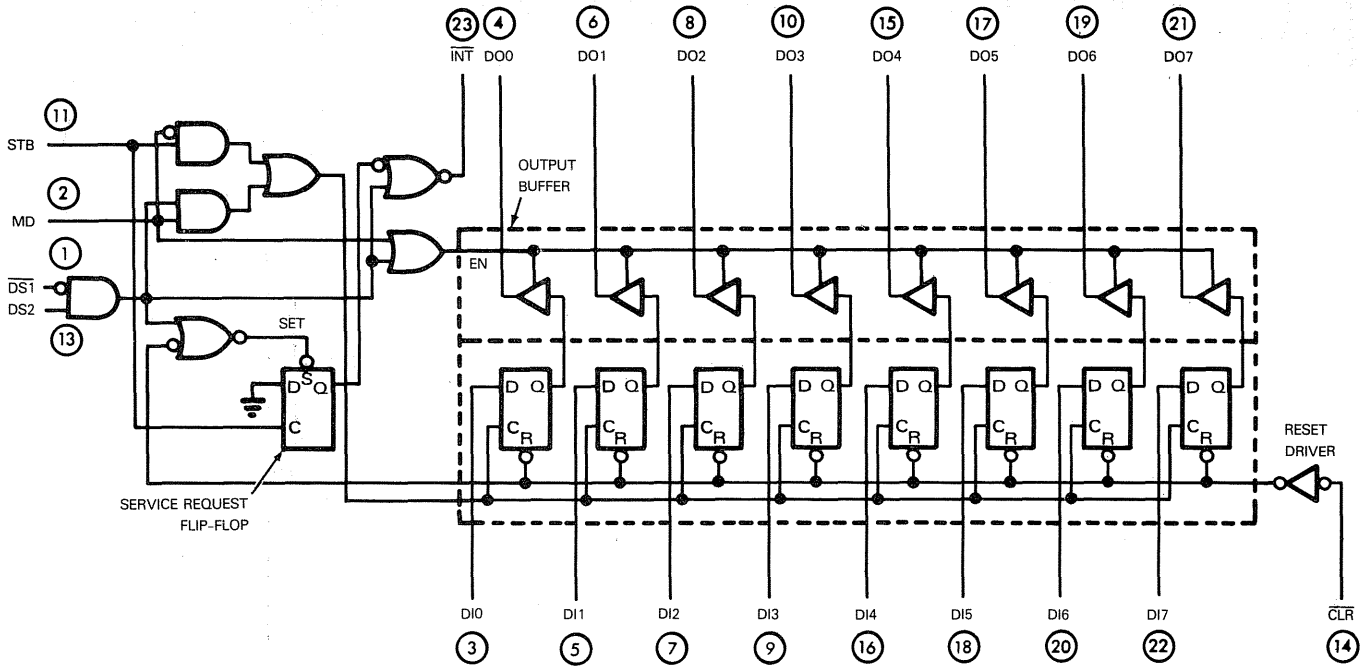
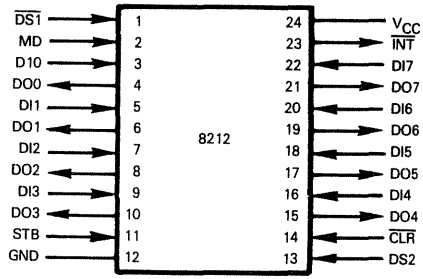


Figure 4-47. Internal Logic Of The 8212 I/O Port



PIN NAME	DESCRIPTION	TYPE
DI0 - DI7	Input Data Bus	Input
DO0 - DO7	Output Data Bus	Output
$\overline{DS1}$,DS2	Device select	Input
MD	Mode select	Input
STB	Data strobe	Input
$\overline{CLR}$	Device Clear	Input
$\overline{INT}$	Interrupt request	Output
V_{CC}	Power	
GND	Ground	

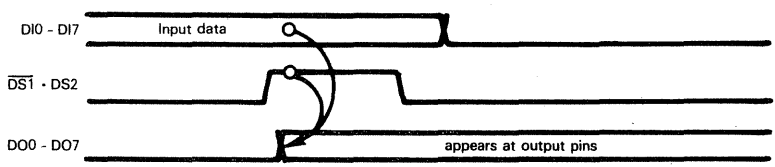
Figure 4-48. 8212 Input/Output Port Signals And Pin Assignments

The eight D-type flip-flops may be clocked either by device select logic, or by the strobe input (STB), depending on the condition of the mode input (MD).

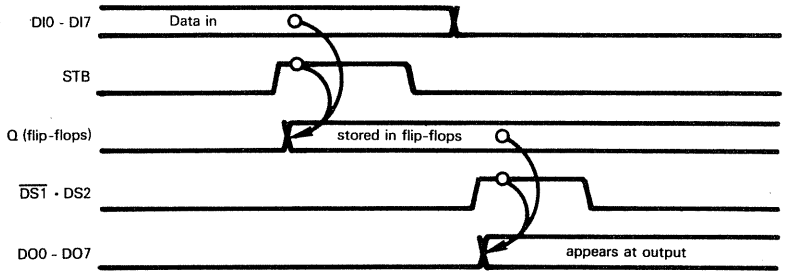
If MD is input high, then device select will clock the flip-flops, and the output buffers are permanently enabled by the high MD level; hence **this is defined as output mode.**

If the MD signal is input low, then it negates chip select logic as a contributor to the flip-flop clock; and it does not enable the output buffers. Now the flip-flop clock inputs must be derived by the strobe input STB, and device select enables the output buffers. With this configuration STB can strobe data into the flip-flop at any time, whether or not the device has been selected. Device select enables the output buffers and occurrence of data at pins DO0 - DO7; hence **MD low is referred to as an input mode.**

To summarize, when MD is high, device select logic strobes data at DI0 - DI7 into the flip-flops, and data flows through immediately to DO0 - DO7:



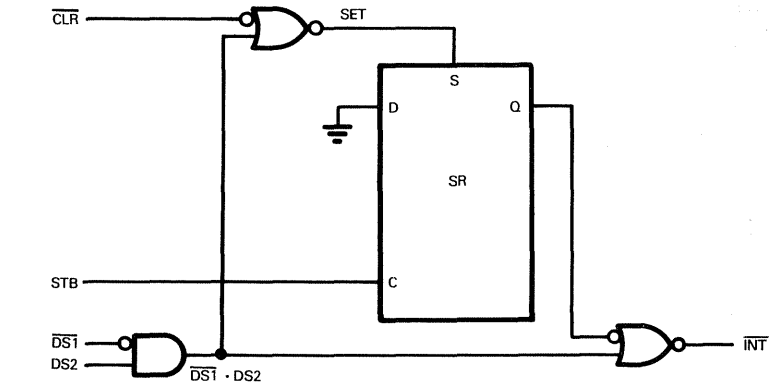
When MD is low, STB may at any time strobe data at DI0 - DI7 into the D-type flip-flops; the 8212 I/O port need not be selected for this to occur. Device select logic controls the instant at which new data is output via DO0 - DO7:

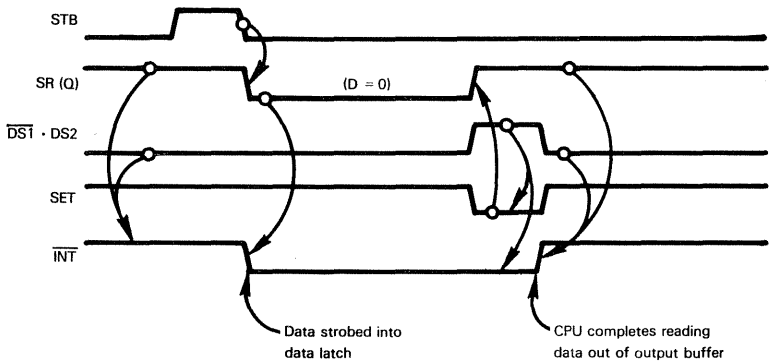


The D-type flip-flops may be reset at any time by setting the $\overline{CLR}$ input low.

The interrupt request signal $\overline{INT}$ functions much as it does in 8255 PPI Modes 1 and 2 operation. When the 8212 I/O port is used for data input, recall that STB high clocks data from DI0 - DI7 into the flip-flops. The $\overline{CPU}$ must be informed of this event. The high-to-low transition of STB clocks the service request flip-flop SR, causing SR (Q) to output low, since SR (D) is tied to ground. SR (Q) low is inverted at the $\overline{INT}$ NOR gate, therefore the NOR gate receives a high input and $\overline{INT}$ is output low. Thus **$\overline{INT}$ informs the CPU that data has been strobed into the data flip-flops.** Assuming that $\overline{CLR}$ is high, the device select logic outputs a second high to the NOR gate which drives the SR flip-flop SET input. SET is therefore input low, forcing SR (Q) high; SR (Q) high translates into a low input to the NOR gate generating $\overline{INT}$, which is forced high in consequence. Device select logic is output high when the contents of the 8212 I/O port is

read, therefore **the process of reading port contents sets $\overline{\text{INT}}$ high again.** This may be illustrated as follows:



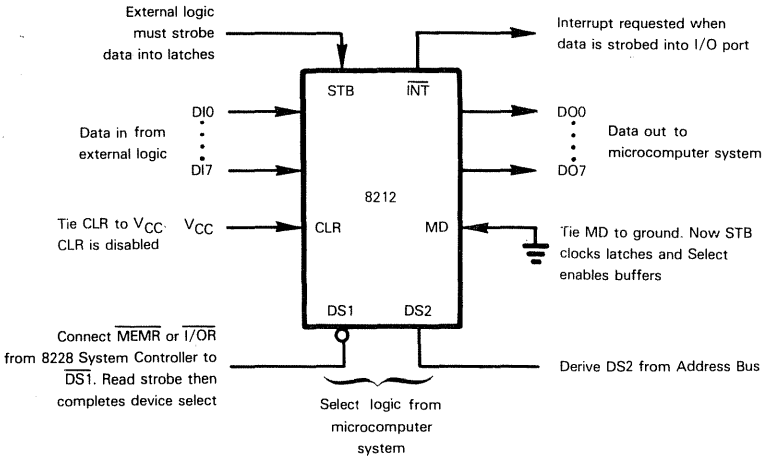


8212 I/O PORT UTILIZATION OPTIONS

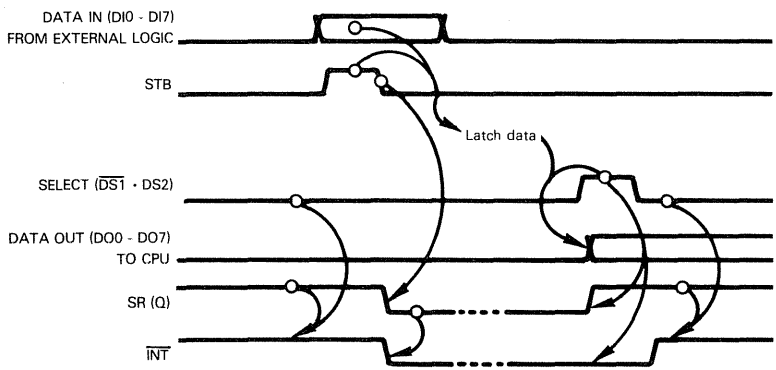
We will now look at some of the ways in which an 8212 I/O port can be used. Let us start by trying to emulate 8255 PPI modes — for a single 8255 I/O port.

Consider the 8212 I/O port being used for data input:

8212 I/O PORT USED FOR INPUT WITH HANDSHAKING



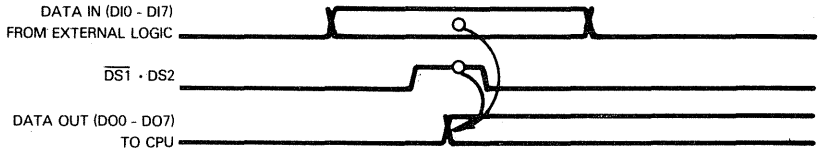
Input timing may be illustrated as follows:



As illustrated above, STB is a handshaking signal, equivalent to the 8255 $\overline{\text{STB}}$ control; $\overline{\text{INT}}$ can be used to signal the CPU that new data is in the latches and may be read. $\overline{\text{INT}}$ must serve the functions of 8255 IBF and INTR controls; **we have** thus **approximated 8255 Mode 1 input.**

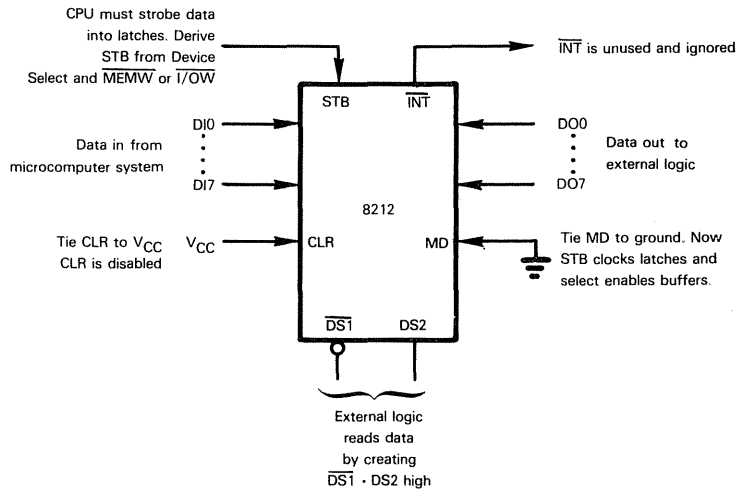
8212 I/O PORT USED FOR SIMPLE INPUT

We can simplify data input protocol by tying STB high (to V_{CC}). DATA IN passes through the latches continuously and SELECT enables DATA OUT. You can ignore $\overline{INT}$. **The 8212 is now operating as a simple, gated buffer,** where SELECT enables the output of whatever data occurs at DI0 - DI7; **this is equivalent to 8255 Mode 0 input.** Timing may be illustrated as follows:

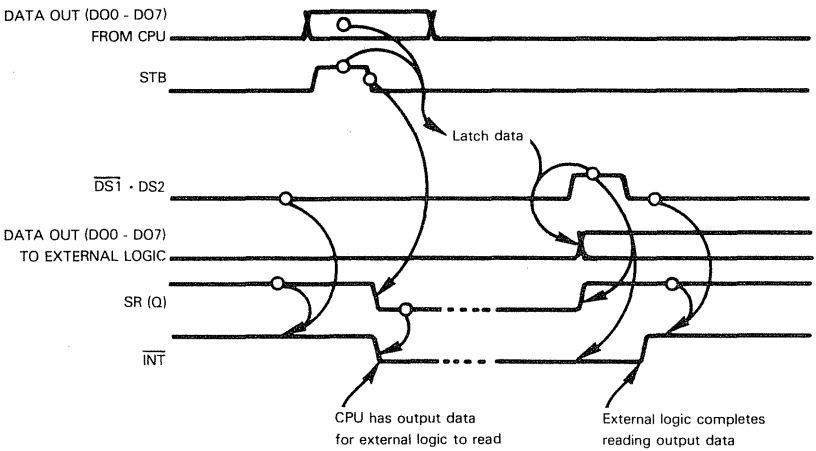


8212 I/O PORT USED FOR OUTPUT WITH HANDSHAKING

The 8212 I/O Port being used for output may be illustrated as follows:

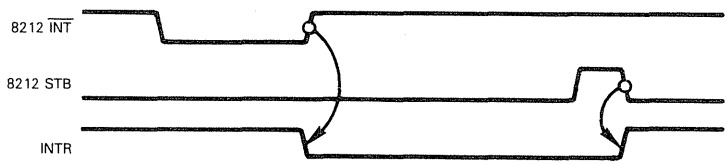


This is a simple reversal of the input logic just described. The CPU creates STB and external logic creates DS1•DS2. DI0 - DI7 is connected to the microcomputer system Data Bus and DO0 - DO7 is connected to external logic. INT goes low when the CPU strobes data into the I/O port, INT therefore informs external logic that data is ready to be output. External logic resets INT upon reading the data. This timing may be illustrated as follows:



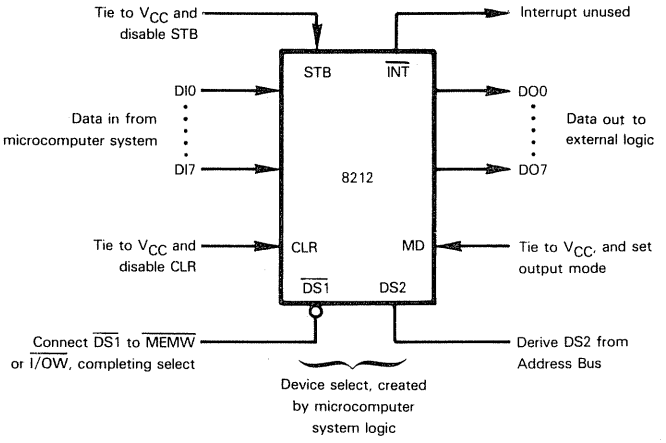
This use of the 8212 I/O port for data output is directly comparable to 8255 Mode 1 output.

8212 $\overline{INT}$ serves the same function as 8255 OBF. 8212 $\overline{DS1}$•DS2 is equivalent to 8255 ACK. The 8255 interrupt request INTR, however, must be derived from external logic as follows:

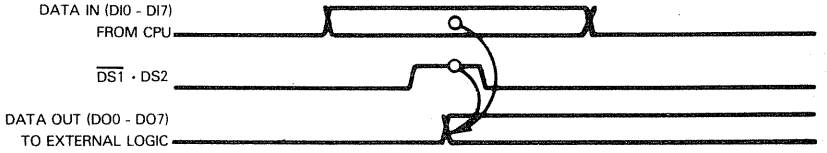


The 8212 I/O port can be configured for simple, Mode 0 output as follows:

8212 I/O PORT USED FOR SIMPLE OUTPUT

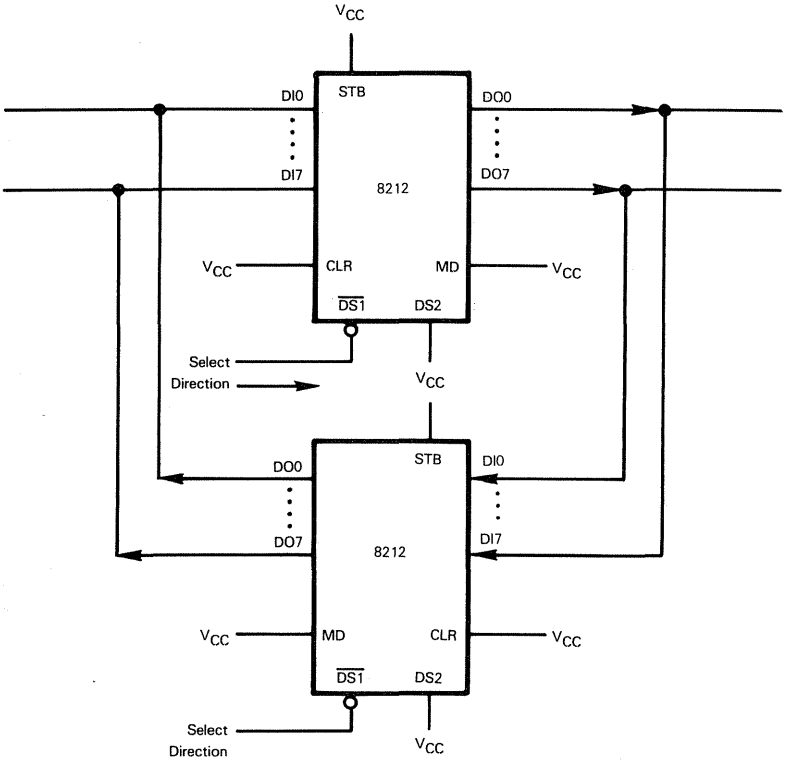


Simple output timing may be illustrated as follows.

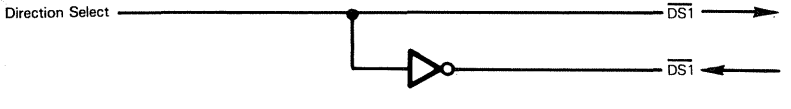


In many small microcomputer configurations two 8212 I/O ports may be wired with one for simple input, the other for simple output, then combined back-to-back to form a bidirectional bus driver:

8212 I/O PORTS AS BIDIRECTIONAL BUS DRIVERS



Device select logic is the key to this use of the 8212 I/O ports. Tie DS2 of both devices to V_{CC}, then use $\overline{DS1}$ to select one device or the other:



Whatever signal is controlling the direction of data flow on the bus becomes the "Direction select". In its simple form it is connected to $\overline{DS1}$ of one 8212; inverted it is connected to $\overline{DS1}$ of the other 8212.