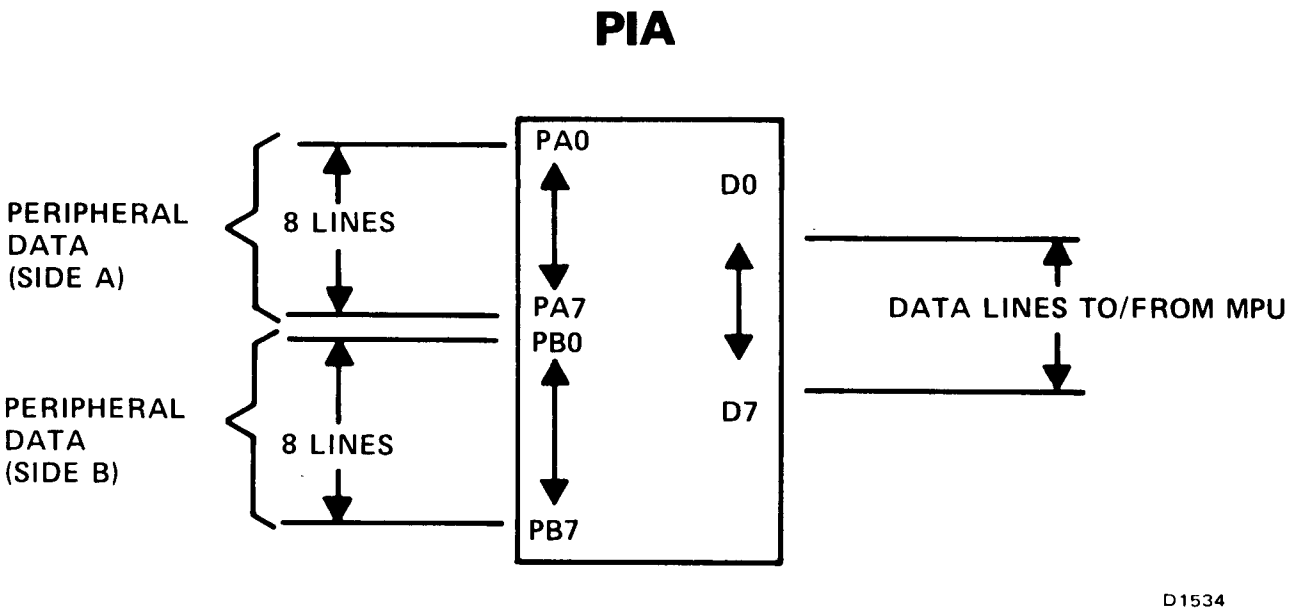
Peripheral Interface Adapter (PIA) - MC6820

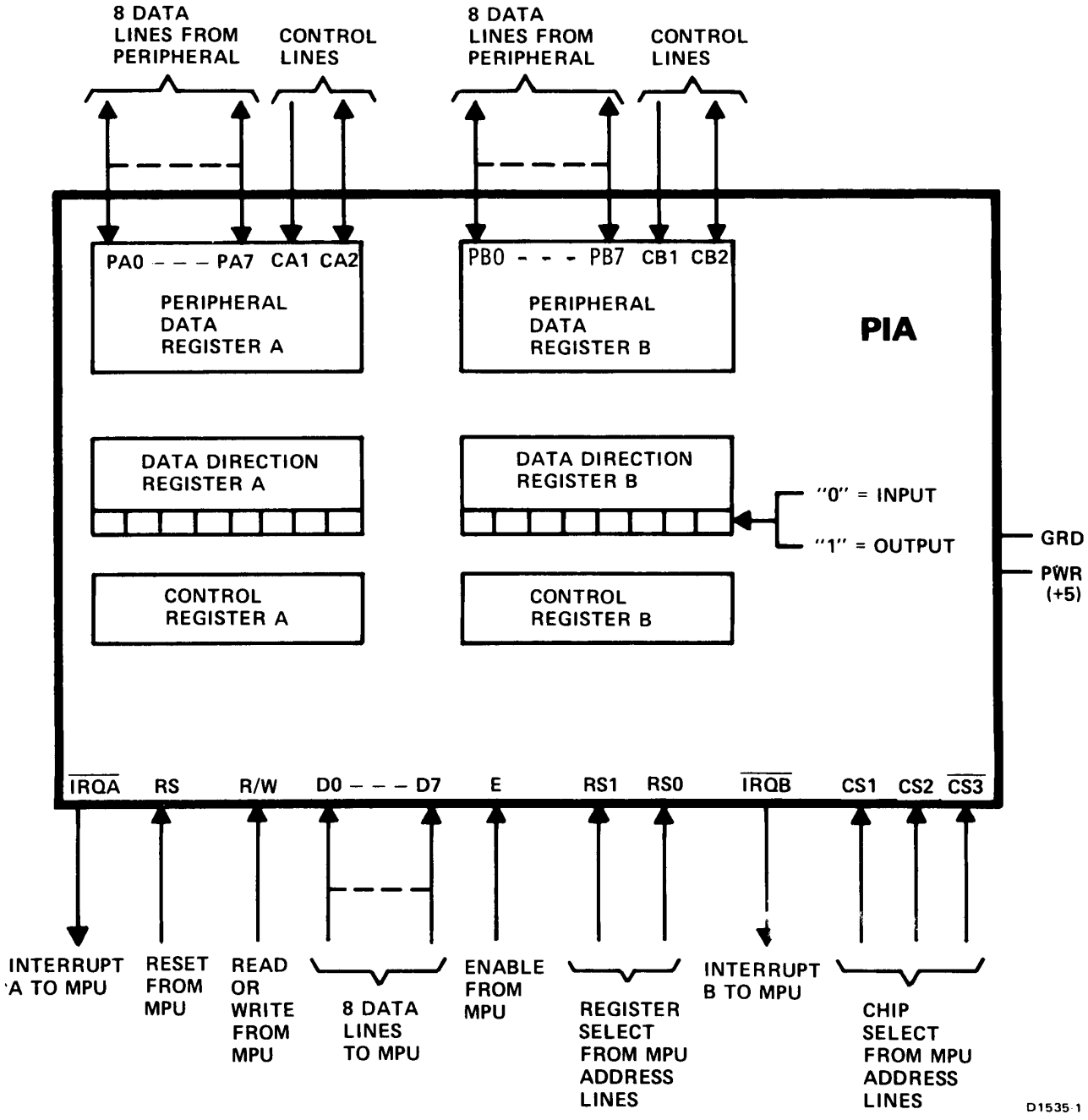
The Peripheral Interface Adapter (PIA) is a means used to interface peripheral equipment with the microprocessing unit (MPU). The PIA communicates with the MPU via an eight bit bi-directional data bus, three chip selects, two register selects, two interrupt request lines, one read/write line, an enable line, and a reset line. These will be discussed in detail later.

Each PIA has two eight bit bi-directional peripheral data buses for interfacing with peripheral equipment as shown in Figure 1.



Each Peripheral data line may be programmed to act as an input or an output. In addition to the two eight bit peripheral data buses, peripheral control lines CA2 and CB2 may be programmed to act as a peripheral data line as will be discussed later.

Each PIA consists of two control registers, two data direction registers, and two peripheral interface registers (peripheral data). The control registers and the data direction registers are used to control the data in and out of the PIA.


8 DATA LINES FROM PERIPHERAL
CONTROL LINES
8 DATA LINES FROM PERIPHERAL
CONTROL LINES
PA0 - - - PA7 CA1 CA2
PERIPHERAL DATA REGISTER A
PB0 - - - PB7 CB1 CB2
PERIPHERAL DATA REGISTER B
PIA
DATA DIRECTION REGISTER A
DATA DIRECTION REGISTER B
"0" = INPUT
"1" = OUTPUT
GRD
PWR (+5)
CONTROL REGISTER A
CONTROL REGISTER B
IRQA RS R/W D0 - - - D7 E RS1 RS0 IRQB CS1 CS2 CS3
INTERRUPT A TO MPU
RESET FROM MPU
READ OR WRITE FROM MPU
8 DATA LINES TO MPU
ENABLE FROM MPU
REGISTER SELECT FROM MPU ADDRESS LINES
INTERRUPT B TO MPU
CHIP SELECT FROM MPU ADDRESS LINES
D1535-1

A. Peripheral Data Lines PAO thru PA7

Each of these 8 data lines which interface with the outside world can be programmed to act as either an input or an output. This is accomplished by setting a "1" in the corresponding bit in the Data Direction Register (DDR) if the line is to be an output or a "0" in the DDR if the line is to be an input. When the data in the peripheral data lines are read into the MPU by a load instruction, those lines which have been designated as, input lines (0 in DDR) will be gated directly to the data bus and into the register selected in the MPU. In the input mode, each line represents a maximum of one standard TTL load.

On the other hand, when an output data instruction (STA A PIA) is executed, data will be transferred via the data bus to the peripheral data register. A "1" output will cause a "high" on the corresponding data line and a "0" output will cause a "Low" on the corresponding data line. Data in Peripheral Register A that have been programmed as outputs may be read by an MPU "LDA A from PIA" instruction. If the voltage is above 2 volts for a logic "1" or below .8 volts for a logic "0", the data will agree with that data outputted. However, if these output lines have been loaded such that they do not meet the levels for logic "1", the data read back into the MPU may differ from the data stored in the PIA Peripheral Register A.

B. Peripheral Data Lines PB0 thru PB7

The 8 data lines which interface with the outside world on the B side may also be programmed to act either as an input or as an output. This is also accomplished by setting a "1" in the corresponding bit in the Data Direction Register (DDR) if the line is to be an output or a "0" in the DDR if the line is to be an input. The output buffers driving these lines have three state capability, allowing them to enter a high impedance state when the peripheral data line is used as an input. Data in Peripheral Register B that have been programmed as outputs may be read by an MPU "LDA A from PIA"

instruction even though the lines have been programmed as outputs. If the line has been programmed as an output ("1"), reading the line will indicate a logic "1" due to buffering between the register and the output pin.

C. Data Lines (DO-D7)

The 8 bi-directional data lines permit transfer of data to/from the PIA and the MPU. The MPU receives data from the outside world from the PIA via these 8 data lines or sends data to the outside world through the PIA's via the 8 data lines. The data bus output drivers are three state devices that remain in the high impedance (off) state except when the MPU performs a PIA read operation.

D. Chip Select Lines (CS1, CS2, CS3)

These are the lines which are tied to the address lines of the MPU. It is through these lines that a particular PIA is selected (addressed). For selection of a PIA, the CS1 and CS2 lines must be high and the CS3 must be low. After the chip selects have been addressed, they must be held in that state for the duration of the E (enable) pulse, which is the only timing signal supplied by the MPU to the PIA. This enable pulse (E) is normally the Ø2 clock. One of the address lines should be ANDed with the VMA line with this output tied to a chip select.

E. Enable Line (E)

The enable pulse (E) is the only timing signal that is supplied to the PIA by the MPU. Timing on all other signals is referenced to the leading or trailing edges of the E pulse.

F. Reset Line (RS)

This line is used to reset all registers in the PIA to a logical zero. This would be used primarily during a reset or power on operation. This line is normally in the high state. The transition of high to low to high resets all registers in the PIA.

G. Read/Write Line (R/W)

This signal is generated by the MPU to control the direction of the data transfers on the Data Bus. A low state on the PIA Read/Write line enables the input buffers and data is transferred from the MPU to the PIA (MPU write) on the E signal if the device has been selected. A high on the Read/Write line sets up the PIA for a transfer of data to the data bus (MPU read). The PIA output buffers are enabled when the proper address & the enable pulse are present thus transferring data to the MPU.

H. Interrupt Request Lines (IRQA & IROB)

These lines are used to interrupt the MPU either directly or indirectly through interrupt priority circuitry. These lines are "open source" (no load device on the chip) and are capable of sinking a current of 1.6 ma from an external source. This permits all interrupt request lines to be tied together in a "wired OR" configuration. Interrupts are serviced by a software routine that sequentially reads & tests, on a prioritized basis, the two control registers in each PIA for interrupt flag bits (Bit 6 & 7) that are set. Discussion on the control registers & how the flag bits get set will follow. When the MPU reads the Peripheral Data Register, the Interrupt Flag (Bit 6 or Bit 7) is cleared & the Interrupt Request is cleared.

These request lines (IRQA & IRQB) are active low.

I. Interrupt Input Lines (CA1 & CB1)

These lines are input only to the PIA and set the interrupt flag (Bit 7) of the control registers in the PIA. Discussion of these lines in conjunction with the control register will follow.

J. Peripheral Control Line (CA2)

This line can be programmed to act either as an interrupt input or as a peripheral output. As an output, this line is compatible with standard TTL and as an input represents one standard TTL load. The function of this line is programmed with Control Register A (Bits 3,4,& 5).

K. Peripheral Control Line (CB2)

This line may also be programmed to act as an interrupt input or as a peripheral output. As an input, this line has greater than 1 megohm input impedance & is compatible with standard TTL. As an output, it is compatible with standard TTL and may also be used as a source of up to 1 millamp at 1.5 volts to directly drive the base of a transistor switch. The function of this line is programmed with Control Register B (Bits 3,4, & 5).

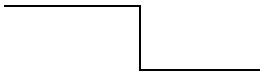
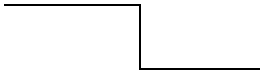
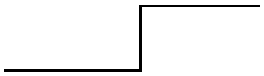
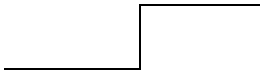
CONTROL REGISTER A (CRA)

7	6	5	4	3	2	1	0
IRQA1	IRQA2	CA2 CONTROL			DDRA	CA1 CONTROL	

CA1 Control (Bit 0 & 1)

Peripheral control line CA1 is an input only line which may be used to cause an interrupt by setting the interrupt flag IRQA1 (Bit 7) of control register A(CRA). Bits 0 and 1 of CRA are used to determine how the interrupt is to be handled.

After the MPU reads Peripheral Data Register A, the IRQA1 (Bit 7) will be cleared.

Transition of interrupt input line CA1	Status of Bit 1 in CRA	Status of Bit 0 in CRA	IRQA1 (Interrupt flag) Bit 7 of CRA	Status of IRQA Line (MPU Interrupt Request)
	0	0	1	MASKED (Remains High)
	0	1	1	GOES LOW (Processor Interrupted)
	1	0	1	MASKED (Remains High)
	1	1	1	GOES LOW (Processor Interrupted)

(All other combinations of CA1 transition and status of bit 0 and bit 1 will be ignored)

Data Direction Access Control (DDRA)-(Bit 2)

This bit, in conjunction with the register select lines (RSO & RS1), is used to select either the Peripheral Data Register or the Data Direction Register. To address the A side control register, RS1 is set to a logic "0" and RSO is set to a logic "1".

RS1	RS0	CRA(Bit 2)	Register Selected
0	0	1	Peripheral Data Register A
0	0	0	Data Direction Register A
0	1	-	Control Register A

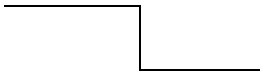
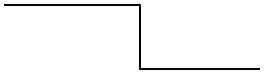
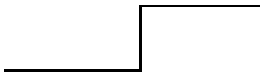
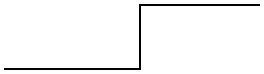
CONTROL REGISTER B (CRB)

7	6	5	4	3	2	1	0
IRQB1	IRQB2	CB2 CONTROL			DDRB	CB1 CONTROL	

CB1 Control (Bit 0 & 1)

Peripheral control line CB1 is an input only line which may be used to cause an interrupt by setting the interrupt flag IRQB1 (Bit 7) of control register B (CRB). Bits 0 and 1 of CRB are used to determine how the interrupt is to be handled.

After the MPU reads Peripheral Data Register B, the IRQB1 (Bit 7) will be cleared.

Transition of interrupt input line CB1	Status of Bit 1 in CRB	Status of Bit 0 in CRB	IRQB1 (Interrupt flag) Bit 7 of CRB	Status of IRQB Line (MPU Interrupt Request)
	0	0	1	MASKED (Remains High)
	0	1	1	GOES LOW (Processor Interrupted)
	1	0	1	MASKED (Remains High)
	1	1	1	GOES LOW (Processor Interrupted)

(All other combinations of CB1 transition and status of bit 0 and bit 1 will be ignored)

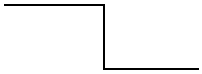
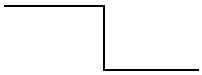
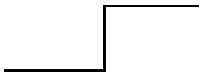
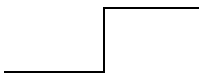
Data Direction Access Control (DDRB)-(Bit 2)

This bit, in conjunction with the register select lines (RSO & RS1), is used to select either the Peripheral Data Register or the Data Direction Register. To address the B side control register, RS1 is set to a logic "1" and RSO is set to a logic "1".

RS1	RS0	CRB(Bit 2)	Register Selected
0	0	1	Peripheral Data Register B
0	0	0	Data Direction Register B
0	1	-	Control Register B

CA2 Control (Bit 3,4, & 5 of CRA)

This line, in addition to generating an interrupt signal, may also be used as an additional output signal. Bits 3, 4, & 5 of the control register determine the function of this line.

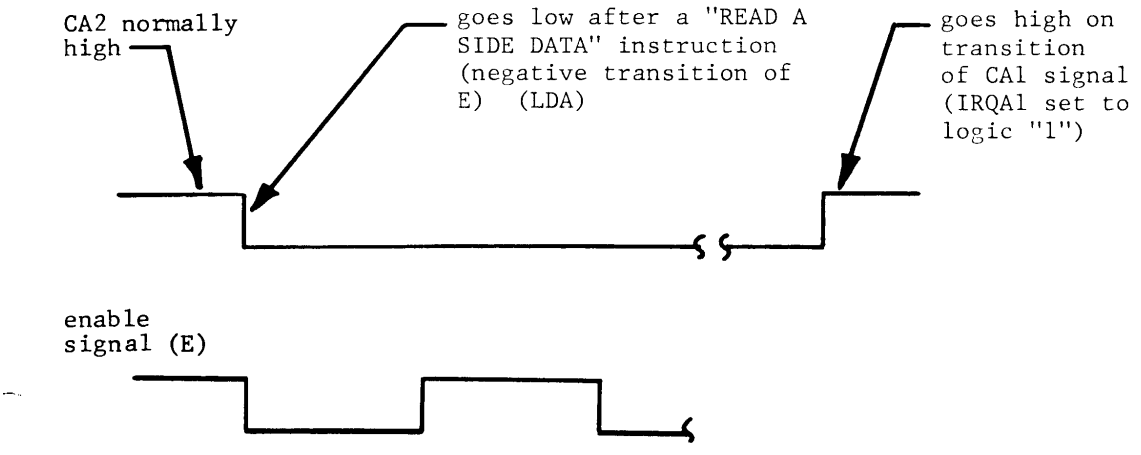
Transition of interrupt input line CA2	Status of Bit 5 in CRA	Status of Bit 4 in CRA	Status of Bit 3 in CRA	IRQA2 (Interrupt flag) Bit 6 of CRA	Status of IRQA Line (MPU Interrupt Request)
	0	0	0	1	MASKED (Remains High)
	0	0	1	1	GOES LOW (Processor Interrupted)
	0	1	0	1	MASKED (Remains High)
	0	1	1	1	GOES LOW (Processor Interrupted)

(All other combinations of CA2 transition and status of bit 3 and bit 4 will be ignored)

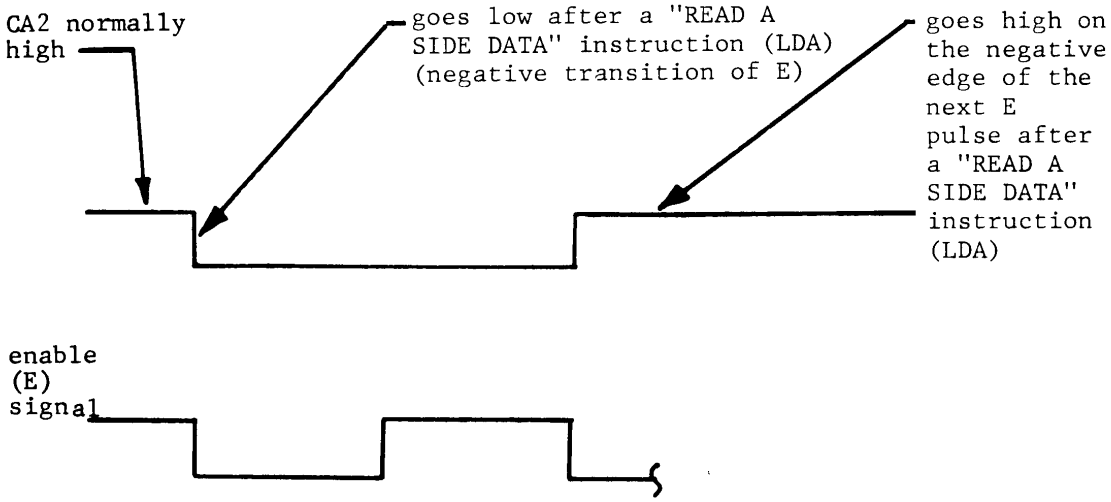
CA2 Used As An Output

If bit 5 of CRA is set to a logic "1", CA2 is designated as an output. The four options utilizing CA2 as an output are shown below. In all four options the IRQA2 flag (bit 6 of CRA remains clear and-the IRQA interrupt request line remains high.

Bit 5, 4, 3 of CRA = 100 (Handshake Mode)



BIT 5,4,3 of CRA = 101



BIT 5,4,3 of CRA = 110

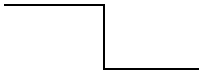
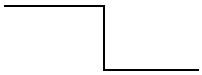
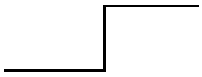
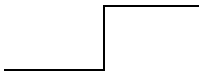
CA2 will always be low with Bits 5, 4, & 3 equal to 110

Bit 5,4,3 of CRA = 111

CA2 will always be high with Bits 5, 4, and 3 equal to 111.

CB2 Control (Bit 3,4, & 5 of CRB)

This line, in addition to generating an interrupt signal, may also be used as an additional output signal. Bits 3, 4, & 5 of the control register determine the function of this line.

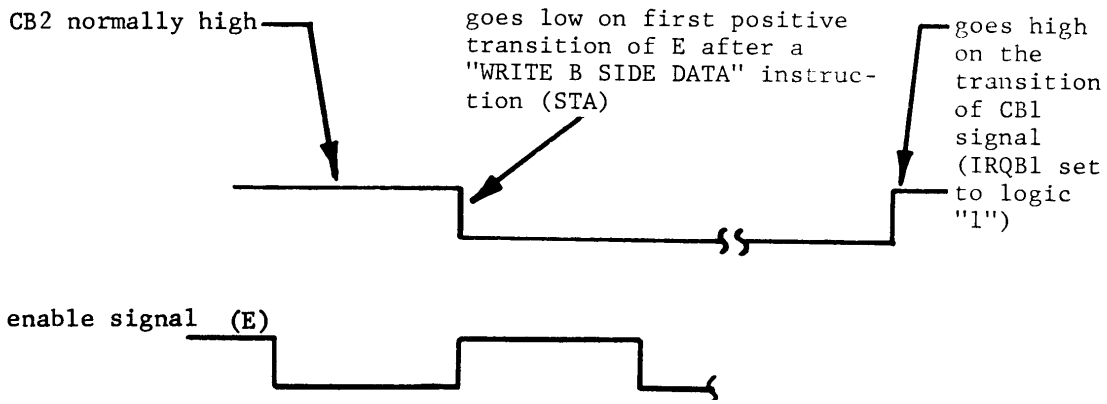
Transition of interrupt input line CB2	Status of Bit 5 in CRB	Status of Bit 4 in CRB	Status of Bit 3 in CRB	IRQB2 (Interrupt flag) Bit 6 of CRA	Status of IRQB Line (MPU Interrupt Request)
	0	0	0	1	MASKED (Remains High)
	0	0	1	1	GOES LOW (Processor Interrupted)
	0	1	0	1	MASKED (Remains High)
	0	1	1	1	GOES LOW (Processor Interrupted)

(All other combinations of CB2 transition and status of bit 3 and bit 4 will be ignored)

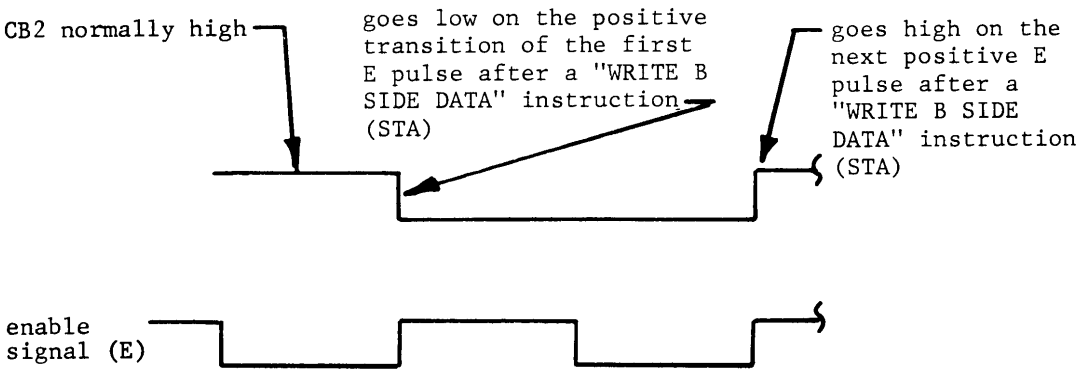
CB2 Used as an Output

If bit 5 of CRB is set to a logic "1", CB2 is designated an an output. The four options utilizing CB2 as an output are shown below. In all four options, the IRQB2 flag (bit 6 of CRB) remains clear and the IRQ interrupt request line remains high

Bit 5, 4, 3 of CRB = 100 (Handshake Mode)



Bit 5,4,3 of CRB = 101



Bit 5,4,3 of CRB = 110

CB2 will always be low with Bits 5,4, & 3 of CRB is equal to 110.

Bit 5,4,3 of CRB = 111

CB2 will always be high when bits 5,4, & 3 of CRB is equal to 111.

SUMARY OF PIA CONTROL REGISTERS:

a) Register selects RSO & RS1

If RS1 is set to a logic "0", then "A" side is selected

If RS1 is set to a logic "1", then the "B" side is selected.

If RSO is set to a logic "0", and CRA (or CRB) Bit 2 is set to a logic "1", the peripheral data register is selected.

If RSO is set to a logic "0", and CRA (or CRB) Bit 2 is set to a logic "0", then the data direction register is selected.

If RSO is set to a logic "1", the control register is selected.

b) CA1 or CB1 Interrupt Line

If bit 0 of CRA (or CRB) is set to a logic "0", all interrupts caused by CA1 (or CB1) are disallowed by the PIA.

c) CA2 or CB2 Interrupt Line

If bit 3 of CRA (or CRB) is set to a logic "0", all interrupts caused by CA2 (or CB2) are disallowed by the PIA. If bit 5 of CRA (or CRB) is set to a logic "1", then the CA2 (or CB2) line is used as an output line per previous table.

Summary of Control Registers CRA & CRB

Control Registers CRA & CRB have total control of CA1, CA2, CB1, and CB2 lines. The status of eight bits of the control registers may be read into the MPU. However, the MPU can only write into bit 0 thru bit 5 (6 bits), since bit 6 & bit 7 are set only by CA1, CA2, CB1, or CB2.

Addressin& PIA's

Before addressing PIA's, the Data Direction (DDR) must first be loaded with the bit pattern that defines how each line is to function i.e. as an input or an output. A logic "1" in the Data Direction Register defines the corresponding line as an output while a logic "0" defines the corresponding line as an input Since the DDR and the Peripheral Data Lines have the same address, the control register bit 2 determines which register is being addressed. It bit 2 in the control register is a logic "0", then the DD R is addressed. If bit 2 in the control register is a logic "1", the Peripheral Data Register is addressed. Therefore, it is essential that the DDR be loaded first before setting bit 2 of the control register.

Example: Given a PIA with an address of 4004, 4005, 4006, & 4007. 4004 is the address of the A side Peripheral Interface Register. 4005 is the address of the A side control register. 4006 is the address of the B side Peripheral Interface Register. 4007 is the address of the B side control register. On the A side, bit 0, 1, 2, & 3 will be defined as inputs while bit 4,5,6 & 7 will be used as outputs. On the B side, all lines will be used as outputs.

The program to accomplish the above is as follows

```
            PIA1AD = 4004
            PIA1AC = 4005
            PIA1BD = 4006
            PIA1BC = 4007

1.   IDA  A #%  11110000          (4 inputs, 4 outputs)
2.   STA  A     PIA1AD            (loads A DDR)
3.   LDA  A #%  11111111          (All outputs)
4.   STA  A     PIAIBD            (Loads B DDR)
5.   LDA  A #%  00000100          (sets bit 2)
6.   STA  A     PIA1AC            (Bit 2 set in A contr. reg)
7.   STA  A     PIA1BC            (Bit 2 set in B contr. reg)
```

Statement 2 addresses the DDR since the Control Register (Bit 2) has not been loaded. Statement 6 & 7 loads the control registers with bit 2 set, so addressing PIAIAD or PIAIBD accesses the Data Register.