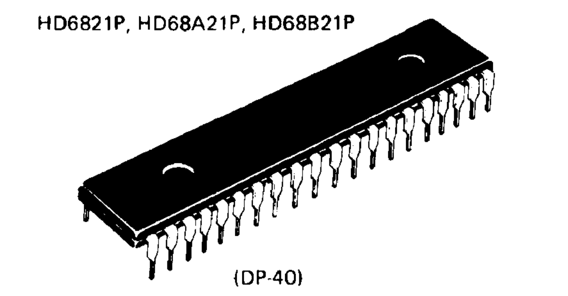
HD6821, HD68A21, HD68B21 — PIA (Peripheral Interface Adapter)

The HD6821 Peripheral Interface Adapter provides the universal means of interfacing peripheral equipment to the HD6800 Microprocessing Unit (MPU). This device is capable of interfacing the MPU to peripherals through two 8-bit bi-directional peripheral data buses and four control lines. No external logic is required for interfacing to most peripheral devices.

The functional configuration of the PIA is programmed by the MPU during system initialization. Each of the peripheral data lines can be programmed to act as an input or output, and each of the four control/interrupt lines may be programmed for one of several control modes. This allows a high degree of flexibility in the over-all operation of the interface.

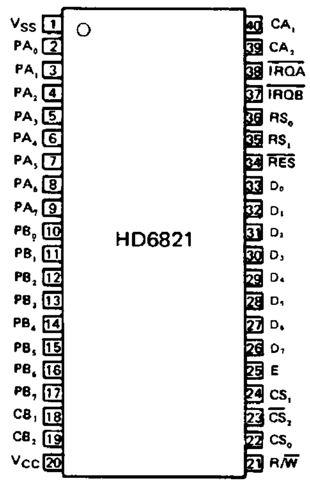


(DP-40)

■ FEATURES

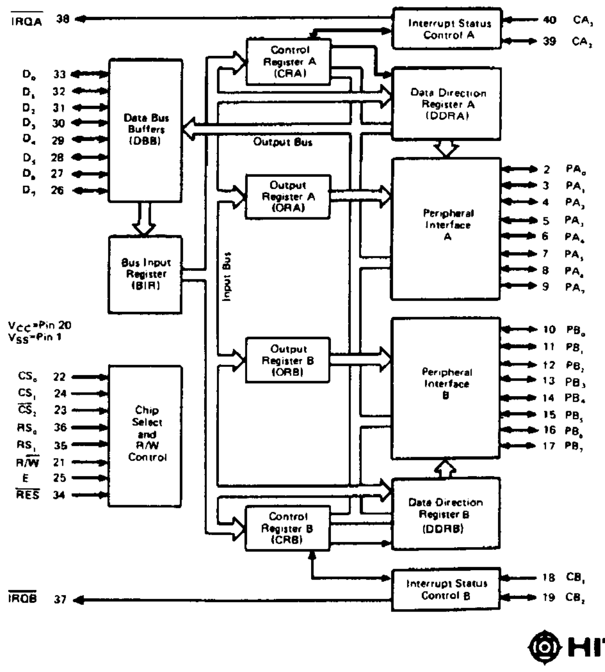
- Two Bi-directional 8-Bit Peripheral Data Bus for interface to Peripheral devices
- Two Programmable Control Registers
- Two Programmable Data Direction Registers
- Four Individually-Controlled Interrupt Input Lines: Two Usable as Peripheral Control Outputs
- Handshake Control Logic for Input and Output Peripheral Operation
- High-Impedance 3-State and Direct Transistor Drive Peripheral Lines
- Program Controlled Interrupt and Interrupt Disable Capability
- CMOS Drive Capability on Side A Peripheral Lines
- Two TTL Drive Capability on All A and B Side Buffers
- N Channel Silicon Gate MOS
- Compatible with MC6821, MC68A21 and MC68B21

■ PIN ARRANGEMENT



(Top View)

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}*	-0.3 ~ +7.0	V
Input Voltage	V_{in}*	-0.3 ~ +7.0	V
Operating Temperature	T_{opr}	-20 ~ +75	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ RECOMMENDED OPERATING CONDITIONS

Item	Symbol	min	typ	max	Unit
Supply Voltage	V_{CC}*	4.75	5.0	5.25	V
Input Voltage	V_{IL}*	-0.3	–	0.8	V
	V_{IH}*	2.0	–	V_{CC}	
Operating Temperature	T_{opr}	-20	25	75	°C

* With respect to V_{SS} (SYSTEM GND)

■ ELECTRICAL CHARACTERISTICS

● DC CHARACTERISTICS (V_{CC}=5.0V±5%, V_{SS}=0V, Ta=-20~+75°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ*	max	Unit
Input "High" Voltage	All Inputs	V_{IH}		2.0	–	V_{CC}	V
Input "Low" Voltage	All Inputs	V_{IL}		-0.3	–	0.8	V
Input Leakage Current	R/$\overline{W}$, $\overline{RES}$, RS_0, RS_1, CS_0, CS_1, $\overline{CS_2}$, CA_1, CB_1, E	I_{in}	V_{in} = 0~5.25V	-2.5	–	2.5	μA
Three-State (Off State) Input Current	D_0~D_7, PB_0~PB_7, CB_2	I_{TSI}	V_{in} = 0.4~2.4V	-10	–	10	μA
Input "High" Current	PA_0~PA_7, CA_2	I_{IH}	V_{IH} = 2.4V	-200	–	–	μA
Input "Low" Current	PA_0~PA_7, CA_2	I_{IL}	V_{IL} = 0.4V	–	–	-2.4	mA
Output "High" Voltage	D_0~D_7	V_{OH}	I_{OH} = -205μA	2.4	–	–	V
	PA_0~PA_7, CA_2		I_{OH} = -200μA	2.4**	–	–	
			I_{OH} = -10μA	V_{CC}-1.0	–	–	
	PB_0~PB_7, CB_2		I_{OH} = -200μA	2.4	–	–	
Output "Low" Voltage	D_0~D_7, $\overline{IRQA}$, $\overline{IRQB}$	V_{OL}	I_{OL} = 1.6mA	–	–	0.4	V
	Other Outputs		I_{OL} = 1.6mA	–	–	0.4	
			I_{OL} = 3.2mA	–	–	0.6	
Output "High" Current	D_0~D_7	I_{OH}	V_{OH} = 2.4V	-205	–	–	μA
	PA_0~PA_7, CA_2		V_{OH} = 2.4V**	-200	–	–	μA
	PB_0~PB_7, CB_2		V_{OH} = 1.5V	-1.0	–	-10	mA
Output Leakage Current (Off State)	$\overline{IRQA}$, $\overline{IRQB}$	I_{LOH}	V_{OH} = 2.4V	–	–	10	μA
Power Dissipation		P_D		–	260	550	mW
Input Capacitance	PA_0~PA_7, PB_0~PB_7, CA_2, CB_2, D_0~D_7	C_{in}	V_{in} = 0V, Ta = 25°C, f = 1.0MHz	–	–	12.5	pF
	R/$\overline{W}$, $\overline{RES}$, RS_0, RS_1, CS_0, CS_1, $\overline{CS_2}$, CA_1, CB_1, E			–	–	10	
Output Capacitance	$\overline{IRQA}$, $\overline{IRQB}$	C_{out}	V_{in} = 0V, Ta = 25°C, f = 1.0MHz	–	–	10	pF

* Ta = 25°C, V_{CC} = 5.0V ** HD68B21; V_{OH} = 2.2V min (PA_0~PA_7, CA_2)

• AC CHARACTERISTICS (V_{CC}=5.0V±5%, V_{SS}=0, Ta=-20~+75°C, unless otherwise noted.)

1. PERIPHERAL TIMING

Item		Symbol	Test Condition	HD6821		HD68A21		HD68B21		Unit
				min	max	min	max	min	max	
Peripheral Data Setup Time		t_{PDSU}	Fig. 1	200	–	135	–	100	–	ns
Peripheral Data Hold Time		t_{PDH}	Fig. 1	0	–	0	–	0	–	ns
Delay Time, Enable negative transition to CA_2 negative transition	Enable → CA_2 Negative	t_{CA2}	Fig. 2, Fig. 3	–	1.0	–	0.67	–	0.5	μs
Delay Time, Enable negative transition to CA_2 positive transition	Enable → CA_2 Positive	t_{RS1}	Fig. 2	–	1.0	–	0.67	–	0.5	μs
Rise and Fall Times for CA_1 and CA_2 input signals	CA_1, CA_2	t_r, t_f	Fig. 3	–	1.0	–	1.0	–	1.0	μs
Delay Time from CA_1 active transition to CA_2 positive transition	CA_1 → CA_2	t_{RS2}	Fig. 3	–	2.0	–	1.35	–	1.0	μs
Delay Time, Enable negative transition to Peripheral Data Valid	Enable→Peripheral Data	t_{PDW}	Fig. 4, Fig. 5	–	1.0	–	0.67	–	0.5	μs
Delay Time, Enable negative transition to Peripheral CMOS Data Valid	Enable → Peripheral Data PA_0~PA_7, CA_2	t_{CMOS}	V_{CC} – 30% V_{CC} Fig. 4	–	2.0	–	1.35	–	1.0	μs
Delay Time, Enable positive transition to CB_2 negative position	Enable → CB_2	t_{CB2}	Fig. 6, Fig. 7	–	1.0	–	0.67	–	0.5	μs
Delay Time, Peripheral Data Valid to CB_2 negative transition	Peripheral Data → CB_2	t_{DC}	Fig. 5	20	–	20	–	20	–	ns
Delay Time, Enable positive transition to CB_2 positive transition	Enable → CB_2	t_{RS1}	Fig. 6	–	1.0	–	0.67	–	0.5	μs
Peripheral Control Output Pulse Width, CA_2/CB_2	CA_2, CB_2	PW_{CT}	Fig. 2, Fig. 6	550	–	550	–	500	–	ns
Rise and Fall Time for CB_1 and CB_2 input signals	CB_1, CB_2	t_r, t_f	Fig. 7	–	1.0	–	1.0	–	1.0	μs
Delay Time, CB_1 active transition to CB_2 positive transition	CB_1 → CB_2	t_{RS2}	Fig. 7	–	2.0	–	1.35	–	1.0	μs
Interrupt Release Time, $\overline{IRQA}$ and $\overline{IRQB}$	$\overline{IRQA}$, $\overline{IRQB}$	t_{IR}	Fig. 9	–	1.6	–	1.1	–	0.85	μs
Interrupt Response Time	$\overline{IRQA}$, $\overline{IRQB}$	t_{RS3}	Fig. 8	–	1.0	–	1.0	–	1.0	μs
Interrupt Input Pulse Width	CA_1, CA_2, CB_1, CB_2	PWI	Fig. 8	500**	–	500**	–	500**	–	ns
Reset "Low" Time	$\overline{RES}$*	t_{RL}	Fig. 10	1.0	–	0.66	–	0.5	–	μs

* The Reset line must be "High" a minimum of 1.0μs before addressing the PIA.
** At least one Enable "High" pulse should be included in this period.

2. BUS TIMING

1) READ

Item		Symbol	Test Condition	HD6821		HD68A21		HD68B21		Unit
				min	max	min	max	min	max	
Enable Cycle Time		t_{cycE}	Fig. 11	1000	–	666	–	500	–	ns
Enable Pulse Width, "High"		PW_{EH}	Fig. 11	450	–	280	–	220	–	ns
Enable Pulse Width, "Low"		PW_{EL}	Fig. 11	430	–	280	–	210	–	ns
Enable Pulse Rise and Fall Times		t_{Er}, t_{Ef}	Fig. 11	–	25	–	25	–	25	ns
Setup Time	Address, R/$\overline{W}$–Enable	t_{AS}	Fig. 12	140	–	140	–	70	–	ns
Address Hold Time		t_{AH}	Fig. 12	10	–	10	–	10	–	ns
Data Delay Time		t_{DDR}	Fig. 12	–	320	–	220	–	180	ns
Data Hold Time		t_{DHR}	Fig. 12	10	–	10	–	10	–	ns

2) WRITE

Item		Symbol	Test Condition	HD6821		HD68A21		HD68B21		Unit
				min	max	min	max	min	max	
Enable Cycle Time		t_{cycE}	Fig. 11	1000	–	666	–	500	–	ns
Enable Pulse Width, "High"		PW_{EH}	Fig. 11	450	–	280	–	220	–	ns
Enable Pulse Width, "Low"		PW_{EL}	Fig. 11	430	–	280	–	210	–	ns
Enable Pulse Rise and Fall Times		t_{Er}, t_{Ef}	Fig. 11	–	25	–	25	–	25	ns
Setup Time		t_{AS}	Fig. 13	140	–	140	–	70	–	ns
Address Hold Time	Address, R/$\overline{W}$–Enable	t_{AH}	Fig. 13	10	–	10	–	10	–	ns
Data Setup Time		t_{DSW}	Fig. 13	195	–	80	–	60	–	ns
Data Hold Time		t_{DHW}	Fig. 13	10	–	10	–	10	–	ns

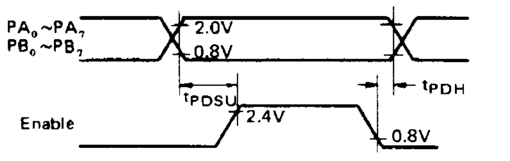
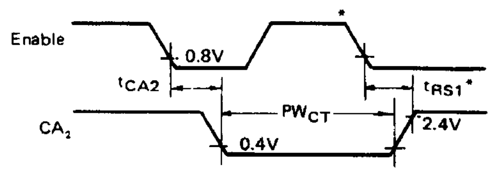


Figure 1 Peripheral Data Setup and Hold Times (Read Mode)



* Assumes part was deselected during the previous E pulse.

Figure 2 CA_2 Delay Time
(Read Mode; CRA5=CRA3=1, CRA4=0)

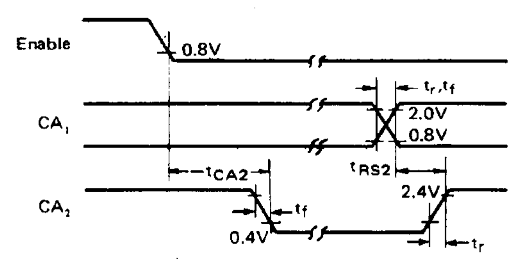


Figure 3 CA_2 Delay Time
(Read Mode; CRA5=1, CRA3=CRA4=0)

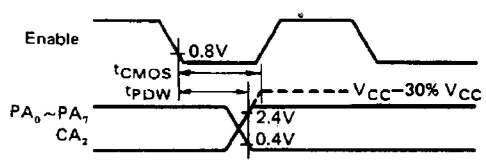
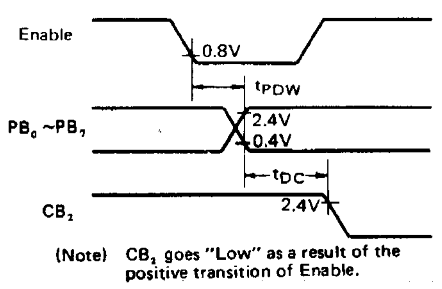
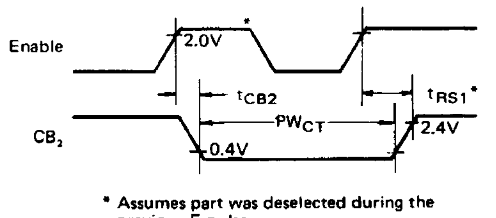


Figure 4 Peripheral CMOS Data Delay Times
(Write Mode; CRA5=CRA3=1, CRA4=0)



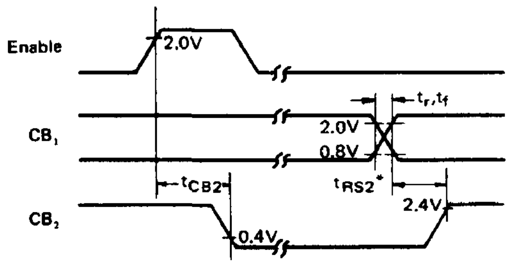
(Note) CB_2 goes "Low" as a result of the positive transition of Enable.

Figure 5 Peripheral Data and CB_2 Delay Times
(Write Mode; CRB5=CRB3=1, CRB4=0)



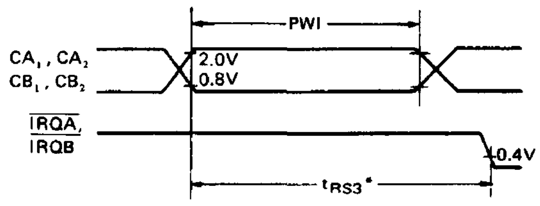
* Assumes part was deselected during the previous E pulse.

Figure 6 CB_2 Delay Time
(Write Mode; CRB5=CRB3=1, CRB4=0)



* Assumes part was deselected during any previous E pulse.

Figure 7 CB_2 Delay Time
(Write Mode; CRB5=1, CRB3=CRB4=0)



* Assumes Interrupt Enable Bits are set.

Figure 8 Interrupt Pulse Width and $\overline{IRQ}$ Response

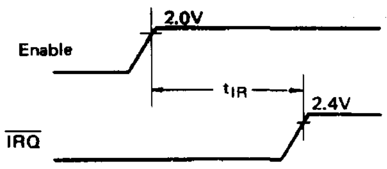
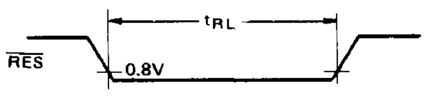


Figure 9 $\overline{IRQ}$ Release Time



* The $\overline{RES}$ line must be a V_{IH} for a minimum of 1.0 μs before addressing the PIA.

Figure 10 $\overline{RES}$ Low Time

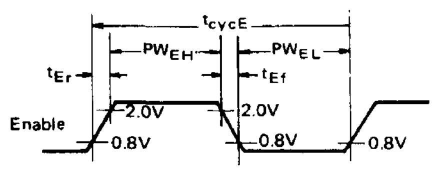


Figure 11 Enable Signal Characteristics

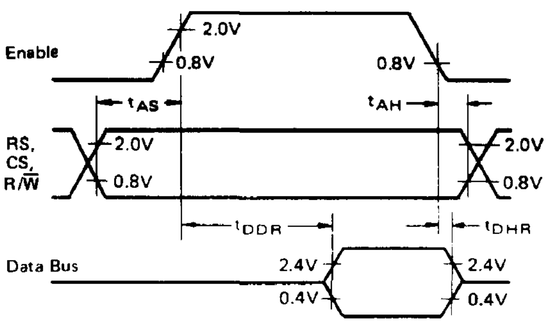


Figure 12 Bus Read Timing Characteristics
(Read Information from PIA)

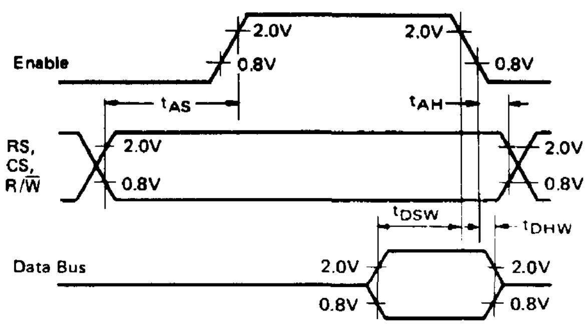
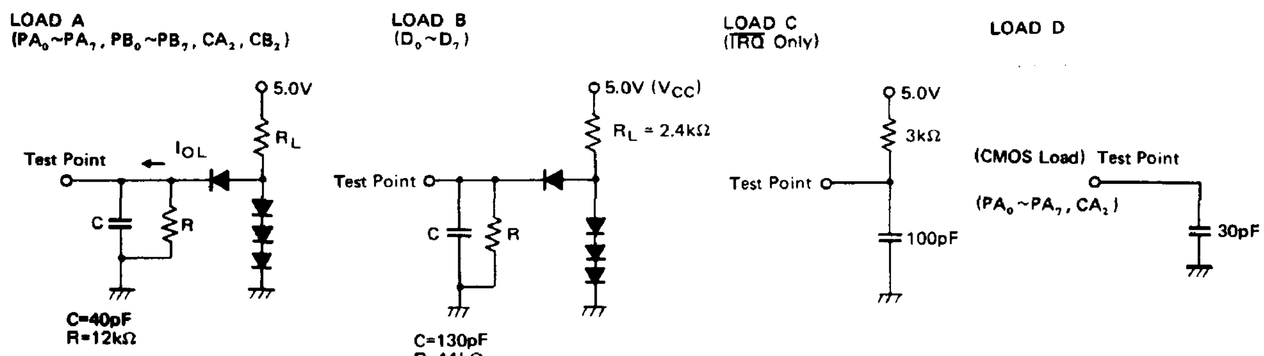


Figure 13 Bus Write Timing Characteristics
(Write Information into PIA)



All diodes are 1S2074Ⓗ or equivalent.

Adjust R_L so that I_{OL} = 1.6mA, then test V_{OL}
Adjust R_L so that I_{OL} = 3.2mA, then test V_{OL}

All diodes are 1S2074 Ⓗ or equivalent.

Figure 14 Bus Timing Test Loads

■ PIA INTERFACE SIGNALS FOR MPU

The PIA interfaces to the HD6800 MPU with an eight-bit bi-directional data bus, three chip select lines, two register select lines, two interrupt request lines, read/write line, enable line and reset line. These signals, in conjunction with the HD6800 VMA output, permit the MPU to have complete control over the PIA. VMA should be utilized in conjunction with an MPU address line into a chip select of the PIA.

- **PIA Bi-Directional Data (D_0~D_7)**

The bi-directional data lines (D_0 ~ D_7) allow the transfer of data between the MPU and the PIA. 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. The R/$\overline{W}$ line is in the Read ("High") state when the PIA is selected for a Read operation.

- **PIA Enable (E)**

The enable pulse, E, is the only timing signal that is supplied to the PIA. Timing of all other signals is referenced to the leading and trailing edges of the E pulse. This signal will normally be a derivative of the HMCS6800 System ϕ_2 Clock. This signal must be continuous clock pulse.

- **PIA Read/Write (R/$\overline{W}$)**

This signal is generated by the MPU to control the direction of data transfers on the Data Bus. A "Low" state on the PIA line enables the input buffers and data is transferred from the MPU to the PIA on the E signal if the device has been selected. A "High" on the R/$\overline{W}$ line sets up the PIA for a transfer of data to the bus. The PIA output buffers are enabled when the proper address and the enable pulse E are present.

- **Reset ($\overline{RES}$)**

The active "Low" $\overline{RES}$ line is used to reset all register bits in the PIA to a logical zero "Low". This line can be used as a power-on reset and as a master reset during system operation.

- **PIA Chip Select (CS_0, CS_1 and $\overline{CS_2}$)**

These three input signals are used to select the PIA. CS_0 and CS_1 must be "High" and $\overline{CS_2}$ must be "Low" for selection of the device. Data transfers are then performed under the control of the E and R/$\overline{W}$ signals. The chip select lines must be stable for the duration of the E pulse. The device is deselected when any of the chip selects are in the inactive state.

- **PIA Register Select (RS_0 and RS_1)**

The two register select lines are used to select the various registers inside the PIA. These two lines are used in conjunction with internal Control Registers to select a particular register that is to be written or read.

The register and chip select lines should be stable for the duration of the E pulse while in the read or write cycle.

- **Interrupt Request ($\overline{IRQA}$ and $\overline{IRQB}$)**

The active "Low" Interrupt Request lines ($\overline{IRQA}$ and $\overline{IRQB}$) act to interrupt the MPU either directly or through interrupt priority circuitry. These lines are "open drain" (no load device on the chip). This permits all interrupt request lines to be tied together in a wire-OR configuration.

Each $\overline{IRQ}$ line has two internal interrupt flag bits that can cause the $\overline{IRQ}$ line to go "Low ". Each flag bit is associated with a particular peripheral interrupt line. Also four interrupt enable bits are provided in the PIA which may be used to inhibit a particular interrupt from a peripheral device.

Servicing an interrupt by the MPU may be accomplished by a software routine that, on a prioritized basis, sequentially reads and tests the two control registers in each PIA for interrupt flag bits that are set.

The interrupt flags are cleared (zeroed) as a result of an MPU Read Peripheral Data Operation of the corresponding data register. After being cleared, the interrupt flag bit cannot be enabled to be set until the PIA is deselected during an E pulse. The E pulse is used to condition the interrupt control lines (CA_1, CA_2, CB_1, CB_2). When these lines are used as interrupt inputs at least one E pulse must occur from the inactive edge to the active edge of the interrupt input signal to condition the edge sense network. If the interrupt flag has been enabled and the edge sense circuit has been properly conditioned, the interrupt flag will be set on the next active transition of the interrupt input pin.

■ PIA PERIPHERAL INTERFACE LINES

The PIA provides two 8-bit bi-directional data buses and four interrupt/control lines for interfacing to peripheral devices.

- **Section A Peripheral Data (PA_0~PA_7)**

Each of the peripheral data lines can be programmed to act as an input or output. This is accomplished by setting a "1" in the corresponding Data Direction Register bit for those lines which are to be outputs. A "0" in a bit of the Data Direction Register causes the corresponding peripheral data line to act as an input. During an MPU Read Peripheral Data Operation, the data on peripheral lines programmed to act as inputs appears directly on the corresponding MPU Data Bus lines.

The data in Output Register A will appear on the data lines that are programmed to be outputs. A logical "1" written into the register will cause a "High" on the corresponding data line while a "0" results in a "Low". Data in Output Register A may be read by an MPU "Read Peripheral Data A" operation when the corresponding lines are programmed as outputs. This data will be read properly if the voltage on the peripheral data lines is greater than 2.0 volts for a logic "1" output and less than 0.8 volt for a logic "0" output. Loading the output lines such that the voltage on these lines does not reach full voltage causes the data transferred into the MPU on a Read operation to differ from that contained in the respective bit of Output Register A.

- ***Section B Peripheral Data (PB_0~PB_7)***

The peripheral data lines in the B Section of the PIA can be programmed to act as either inputs or outputs in a similar manner to PA_0~PA_7. However, the output buffers driving these lines differ from those driving lines PA_0~PA_7. They have three-state capability, allowing them to enter a high impedance state when the peripheral data line is used as a input. In addition, data on the peripheral data lines PB_0~PB_7 will be read properly from those lines programmed as outputs even if the voltages are below 2.0 volts for a "High". As outputs, these lines are compatible with standard TTL and may also be used as a source of up to 2.5 milliampere (typ.) at 1.5 volts to directly drive the base of a transistor switch.

- **Interrupt Input (CA_1 and CB_1)**

Peripheral Input lines CA_1 and CB_1 are input only lines that set the interrupt flags of the control registers. The active transition for these signals is also programmed by the two control registers.

- **Peripheral Control (CA_2)**

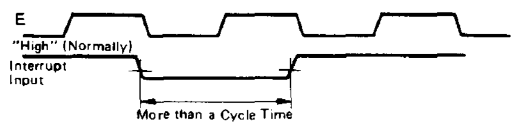
The peripheral control line CA_2 can be programmed to act as an interrupt input or as a peripheral control output. As an output, this line is compatible with standard TTL. The function of this signal line is programmed with Control Register A.

- **Peripheral Control (CB_2)**

Peripheral Control line CB_2 may also be programmed to act as an interrupt input or peripheral control output. As an input, this line has "High" input impedance and 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 2.5 milliampere (typ) at 1.5 volts to directly drive the base of a transistor switch. This line is programmed by Control Register B.

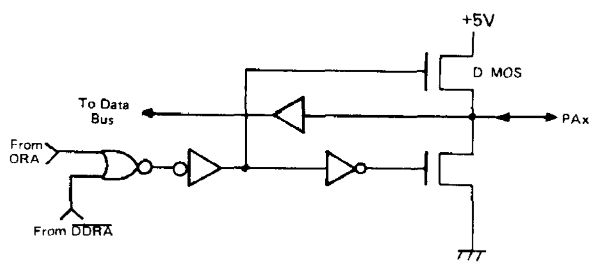
(NOTE) 1. Interrupt inputs CA_1, CA_2, CB_1 and CB_2 shall be used at normal "High" level. When interrupt inputs are "Low" at reset ($\overline{RES}$ = "Low"), interrupt flags CRA6, CRA7, CRB6 and CRB7 may be set.

2. Pulse width of interrupt inputs CA_1, CA_2, CB_1 and CB_2 shall be greater than a E cycle time. In the case that "High" time of E signal is not contained in Interrupt pulse, an interrupt flag may not be set.

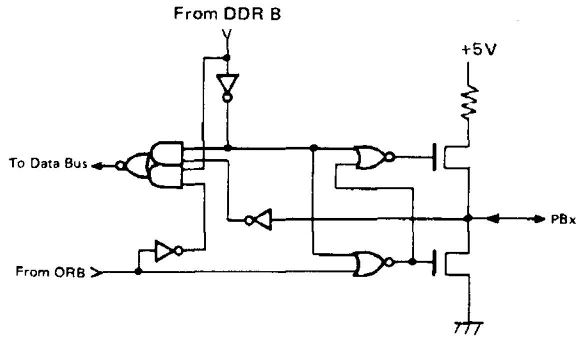


- **The equivalent Circuit of the Lines on Peripheral side**

The equivalent circuit of the lines on Peripheral side is shown in Fig. 15. The output circuits of A port is different from that of B port. When the port is used as input, the input is pullup to V_{CC} side through load MOS in A port and B port becomes "Off" (high impedance).



(a) Section A



(b) Section B

Figure 15 Peripheral Data Bus

■ **INTERNAL CONTROLS**

There are six locations within the PIA accessible to the MPU data bus: two Peripheral Registers, two Data Direction Registers, and two Control Registers. Selection of these locations is controlled by the RS_0 and RS_1 inputs together with bit 2 in the Control Register, as shown in Table 1.

Table 1 Internal Addressing

RS_1	RS_0	Control Register Bit CRA2	Control Register Bit CRB2	Location Selected
0	0	1	x	Peripheral Register A*
0	0	0	x	Data Direction Register A
0	1	x	x	Control Register A
1	0	x	1	Peripheral Register B*
1	0	x	0	Data Direction Register B
1	1	x	x	Control Register B

x = Don't Care
* Peripheral interface register is a generic term containing peripheral data bus and output register.

- **Initialization**

A "Low" reset line has the effect of zeroing all PIA registers. This will set PA_0~PA_7, PB_0~PB_7, CA_2 and CB_2 as inputs, and all interrupts disabled. The PIA must be configured during the restart program which follows the reset.

Details of possible configurations of the Data Direction and Control Register are as follows.

- **Data Direction Registers (DDRA and DDRB)**

The two Data Direction Registers allow the MPU to control the direction of data through each corresponding peripheral data line. A Data Direction Register bit set at "0" configures the corresponding peripheral data line as an input; a "1" results in an output.

- **Control Registers (CRA and CRB)**

The two Control Registers (CRA and CRB) allow the MPU to control the operation of the four peripheral control lines CA_1, CA_2, CB_1 and CB_2. In addition they allow the MPU to enable the interrupt lines and monitor the status of the interrupt flags. Bits 0 through 5 of the two registers may be written or read by the MPU when the proper chip select and register select signals are applied. Bits 6 and 7 of the two registers are read only and are modified by external interrupts occurring on control lines CA_1, CA_2, CB_1 or CB_2. The format of the control words is shown in Table 2.

Table 2 Control Word Format

	7	6	5	4	3	2	1	0
CRA	IRQA1	IRQA2	CA_2 Control			DDRA Access	CA_1 Control	

	7	6	5	4	3	2	1	0
CRB	IRQB1	IRQB2	CB_2 Control			DDRB Access	CB_1 Control	

Data Direction Access Control Bit (CRA2 and CRB2)

Bit 2 in each Control register (CRA and CRB) allows selection of either a Peripheral Interface Register or the Data Direction Register when the proper register select signals are applied to RS_0 and RS_1.

Interrupt Flags (CRA6, CRA7, CRB6, and CRB7)

The four interrupt flag bits are set by active transitions of signals on the four Interrupt and Peripheral Control lines when those lines are programmed to be inputs. These bits cannot be set directly from the MPU Data Bus and are reset indirectly by a Read Peripheral Data Operation on the appropriate section.

Control of CA_1 and CB_1 Interrupt Lines (CRA0, CRB0, CRA1, and CRB1)

The two lowest order bits of the control registers are used to control the interrupt input lines CA_1 and CB_1. Bits CRA0 and CRB0 are used to enable the MPU interrupt signals $\overline{IRQA}$ and $\overline{IRQB}$, respectively. Bits CRA1 and CRB1 determine the active transition of the interrupt input signals CA_1 and CB_1 (Table 3)

Control of CA_2 and CB_2 Peripheral Control Lines (CRA3, CRA4, CRA5, CRB3, CRB4, and CRB5)

Bits 3, 4 and 5 of the two control registers are used to control the CA_2 and CB_2 Peripheral Control lines. These bits determine if the control lines will be an interrupt input or an output control signal. If bit CRA5 (CRB5) is "0" CA_2 (CB_2) is an interrupt input line similar to CA1 (CB1) (Table 4). When CRA5 (CRB5) is "1", CA_2 (CB_2) becomes an output signal that may be used to control peripheral data transfers. When in the output mode, CA_2 and CB_2 have slightly different characteristics (Table 5 and 6).

Table 3 Control of Interrupt Inputs CA_1 and CB_1

CRA1 (CRB1)	CRA0 (CRB0)	Interrupt Input CA_1 (CB_1)	Interrupt Flag CRA7 (CRB7)	MPU Interrupt Request $\overline{IRQA}$ ($\overline{IRQB}$)
0	0	↓ Active	Set "1" on ↓ of CA_1 (CB_1)	Disabled – $\overline{IRQ}$ remains "High"
0	1	↓ Active	Set "1" on ↓ of CA_1 (CB_1)	Goes "Low" when the interrupt flag bit CRA7 (CRB7) goes "1"
1	0	↑ Active	Set "1" on ↑ of CA_1 (CB_1)	Disabled – $\overline{IRQ}$ remains "High"
1	1	↑ Active	Set "1" on ↑ of CA_1 (CB_1)	Goes "Low" when the interrupt flag bit CRA7 (CRB7) goes "1"

(Notes)
1. ↑ indicates positive transition ("Low" to "High")
2. ↓ indicates negative transition ("High" to "Low")
3. The Interrupt flag bit CRA7 is cleared by an MPU Read of the A Peripheral Register and CRB7 is cleared by an MPU Read of the B Peripheral Register.
4. If CRA0 (CRB0) is "0" when an interrupt occurs (Interrupt disabled) and is later brought "1", $\overline{IRQA}$ ($\overline{IRQB}$) occurs after CRA0 (CRB0) is written to a "1".

Table 4 Control of CA_2 and CB_2 as Interrupt Inputs – CRA5 (CRB5) is "0"

CRA5 (CRB5)	CRA4 (CRB4)	CRA3 (CRB3)	Interrupt Input CA_2 (CB_2)	Interrupt Flag CRA6 (CRB6)	MPU Interrupt Request $\overline{IRQA}$ ($\overline{IRQB}$)
0	0	0	↓ Active	Set "1" on ↓ of CA_2 (CB_2)	Disabled – $\overline{IRQ}$ remains "High"
0	0	1	↓ Active	Set "1" on ↓ of CA_2 (CB_2)	Goes "Low" when the interrupt flag bit CRA6 (CRB6) goes "1"
0	1	0	↑ Active	Set "1" on ↑ of CA_2 (CB_2)	Disabled – $\overline{IRQ}$ remains "High"
0	1	1	↑ Active	Set "1" on ↑ of CA_2 (CB_2)	Goes "Low" when the interrupt flag bit CRA6 (CRB6) goes "1"

(Notes)
1. ↑ indicates positive transition ("Low" to "High")
2. ↓ indicates negative transition ("High" to "Low")
3. The interrupt flag bit CRA6 is cleared by an MPU Read of the A Peripheral Register and CRB6 is cleared by an MPU Read of the B Peripheral Register.
4. If CRA3 (CRB3) is "0" when an interrupt occurs (Interrupt disabled) and is later brought "1", $\overline{IRQA}$ ($\overline{IRQB}$) occurs after CRA3 (CRB3) is written to a "1".

Table 5 Control of CB_2 as an Output – CRB5 is "1"

CRB5	CRB4	CRB3	CB_2 Cleared	CB_2 Set
1	0	0	"Low" on the positive transition of the first E pulse after MPU Write "B" Data Register operation.	"High" when the interrupt flag bit CRB7 is set by an active transition of the CB_1 signal. (See Figure 16)
1	0	1	"Low" on the positive transition of the first E pulse after an MPU Write "B" Data Register operation.	"High" on the positive edge of the first "E" pulse following an "E" pulse which occurred while the part was deselected. (See Figure 16)
1	1	0	"Low" (The content of CRB3 is output on CB_2)	
1	1	1	"High" (The content of CRB3 is output on CB_2)	

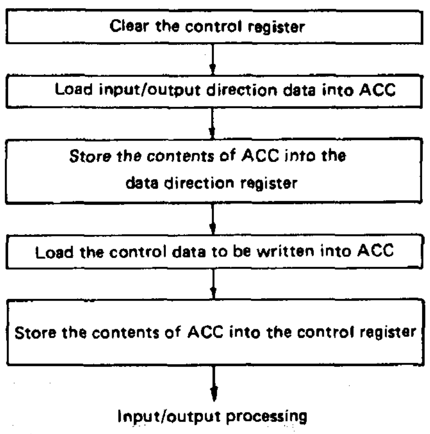
Table 6 Control of CA_2 as an Output – CRA5 is "1"

CRA5	CRA4	CRA3	CA_2 Cleared	CA_2 Set
1	0	0	"Low" on negative transition of E after an MPU Read "A" Data Operation.	"High" when the interrupt flag bit CRA7 is set by an active transition of the CA_1 signal. (See Figure 16)
1	0	1	"Low" on negative transition of E after an MPU Read "A" Data operation.	"High" on the negative edge of the first "E" pulse which occurs during a deselect. (See Figure 16)
1	1	0	"Low" (The content of CRA3 is output on CA_2)	
1	1	1	"High" (The content of CRA3 is output on CA_2)	

■ PIA OPERATION

● Initialization

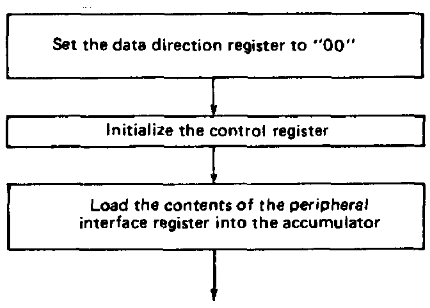
When the external reset input $\overline{RES}$ goes "Low", all internal registers are cleared to "0". Periperal data port (PA_0~PA7, PB_0~PB_7) is defined to be input and control lines (CA_1, CA_2, CB_1 and CB_2) are defined to be the interrupt input lines. PIA is also initialized by software sequence as follows.



- Program the data direction register access bit of the control register to "0" to allow to access the dada direction register.
- The data of the control line function is set into the accumulator, of which Data Direction Register Access Bit shall be programmed to "1".
- Transfer the control data from the accumulator into the control register.

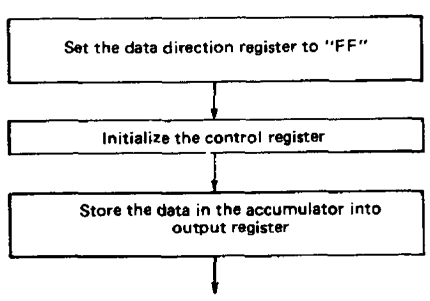
● Read/Write Operation Not Using Control Lines

<Read Operation>



CLR	CRA	● Clear the DDRA access bit of the control register to "0".
CLR	DDRA	● Clear all bits of the dada direction register.
LDAA	#$04	● Set DDRA access bit of the control register to "1" to allow to access the peripheral interface register.
STAA	CRA	
LDAA	PIRA	

<Write Operation>



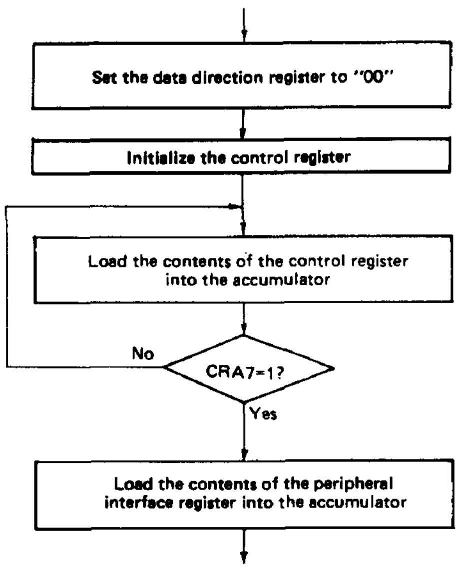
CLR	CRA	● Set DDRB access bit of the control register to "0".
LDAA	#$FF	● Set all bits of the data direction register to "FF".
STAA	DDRB	
LDAA	#$04	● Set DDRB access bit of the control register to "1" to allow to access the peripheral interface register.
STAA	CRB	
LDAA	DATA	● Write the data into the peripheral interface register.
STAA	PIRB	

- **Read/Write Operating Using Control Lines**

Read/write request from peripherals shall be put into the control lines as an interrupt signal, and then MPU reads or writes after detecting interrupt request.

< Read >

The following case is that Port A is used and that the rising edge of CA_1 indicates the request for read from peripherals.



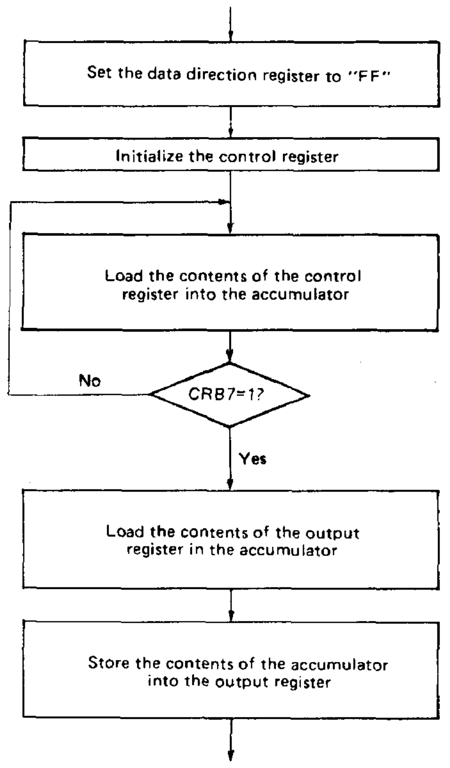
CLR CRA	• Set the DDRA access bit to "0".
CLR DDRA	• Set all bits of the data direction register to "0".
LDAA #$06 STAA CRA	• Program the rising edge of CA_1 to be active. $\overline{IRQA}$ is masked and DDRA access bit is set to 1.
LOOP LDAA CRA BPL LOOP	• Check whether the read request comes from peripherals or not.
LDAA PIRA	• Load the data from the peripheral interface register into the accumulator. CRA flag is reset after this read operation.

To read the peripheral data, the data is directly transfered to the data buses D_0~D_7 through PA_0~PA_7 or PB_0~PB_7 and they are not latched in the PIA. If necessary, the data should be held in the external latch until MPU completes reading it.

When initializing the control register, interrupt flag bit (CRA7, CRA6, CRB7, CRB6) cannot be written from MPU. If necessary the interrupt flag must be reset by dummy read of Peripheral Register A and B.

<Write>

Write operation using the interrupt signal is as follows. In this case, B port is used and interrupt request is input to CB_1. And the $\overline{IRQ}$ flag is set at the rising edge of CB_1.




```
     CLR   CRB    • Set DDRB access bit to "0"
     LDAA  #$FF } • Set all bits of DDRB to output "1".
     STAA  DDRB }
     LDAA  #$06 } • Program the rising edge of CB1 to be active. IRQB is masked
     STAA  CRB  }   and DDRB access bit is set to "1".
LOOP LDAA  CRB  } • Check whether the write request comes from peripherals
     BPL   LOOP }   or not.
     LDAA  PIRB   • Reset the CRB7 flag by the dummy read of the
                    peripheral interface register.
     STAB  PIRB   • Store the data of the accumulator B to the peripheral
                    interface register.
```

Interrupt request flag bits (CRA7, CRA6, CRB7 and CRB6) cannot be written and they cannot be also reset by write operation to the peripheral interface register. So dummy read of peripheral interface register is needed to reset the flags.

To accept the next interrupt, it is essential to reset indirectly the interrupt flag by dummy read of peripheral interface register.

Software poling method mentioned above requires MPU to continuously monitor the control register to detect the read/write request from peripherals. So other programs cannot run at the same time. To avoid this problem, hardware interrupt may be used. The MPU is interrupted by $\overline{IRQA}$ or $\overline{IRQB}$ when the read/write request is occurred from peripherals and then MPU analyzes cause of the interrupt request during interrupt processing.

- **Handshake Mode**

The functions of CRA and CRB are similar but not identical in the hand-shake modes. Port A is used for read hand-shake operation and Port B is used for write hand-shake mode.

CA_1 and CB_1 are used for interrupt input requests and CA_2 and CB_2 are control outputs (answer) in hand-shake mode.

Fig. 16, Fig. 17 and Fig. 18 show the timing of hand-shake mode.

< Read Hand-shake Mode>

CRA5="1", CRA4="0" and CRA3="0"

(1) A peripheral device puts the 8-bit data on the peripheral data lines after the control output CA_2 goes "Low".

(2) The peripheral requests MPU to read the data by using CA_1 input.

(3) CRA7 flag is set and CA_2 becomes "High" (CA_2 automatically becomes "High" by the interrupt CA_1). This indicates the peripheral to maintain the current data and not to transfer the next data.

(4) MPU accepts the read request by $\overline{IRQA}$ hardware interrupt or CRA read. Then MPU reads the peripheral register A.

(5) CA_2 goes "Low" on the following edge of read Enable pulse. This informs that the peripheral can set the next data to port A.

<Write Hand-shake>

CRB5 = "1", CRB4 = "0" and CRB3 = "0"

(1) A peripheral device requests MPU to write the data by using CB_1 input. CB_2 output remains "High" until MPU write data to the peripheral interface register.

(2) CRB7 flag is set and MPU accepts the write request.

(3) MPU reads the peripheral interface register to reset CRB7 (dummy read).

(4) Then MPU write data to the peripheral interface register. The data is output to port B through the output register.

(5) CB_2 automatically becomes "Low" to tell the peripheral that new data is on port B.

(6) The peripheral read the data on Port B peripheral data lines and set CB_1 to "Low" to tell MPU that the data on the peripheral data lines has been taken and that next data can be written to the peripheral interface register.

<Pulse mode>

CRA5 = "1", CRA4 = "0" and CRA3 = "1"

CRB5 = "1", CRB4 = "0" and CRB3 = "1"

This mode is shown in Figure 16, Figure 19 and Figure 20.

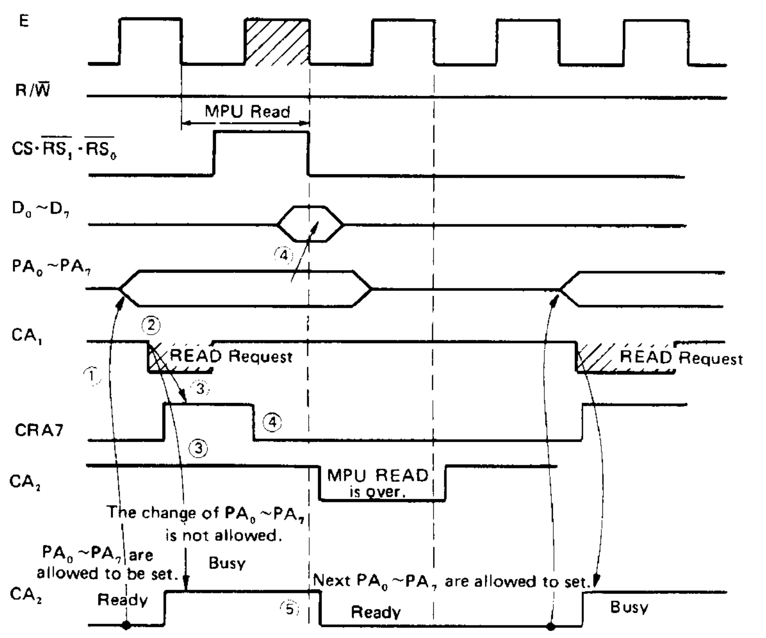
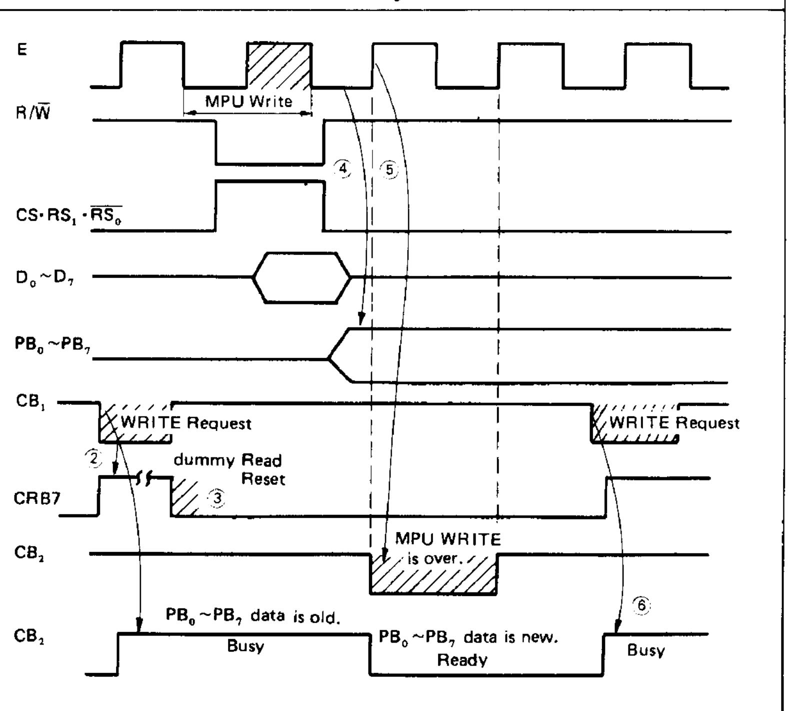
	CRA5	CRA4	CRA3	Timing
Port A (Read)	1	0	1	
	1	0	0	
	CRB5	CRB4	CRB3	Timing
Port B (Write)	1	0	1	
	1	0	0	

Figure 16 Timing of Hand-shake Mode and Pulse Mode

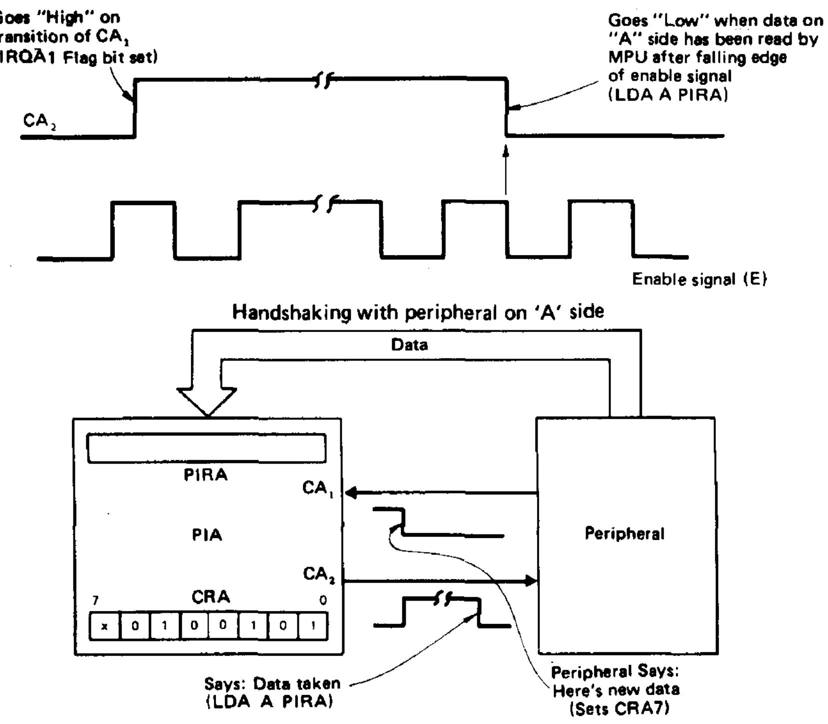


Figure 17 Bits 5, 4, 3 of CRA = 100 (Hand-shake Mode)

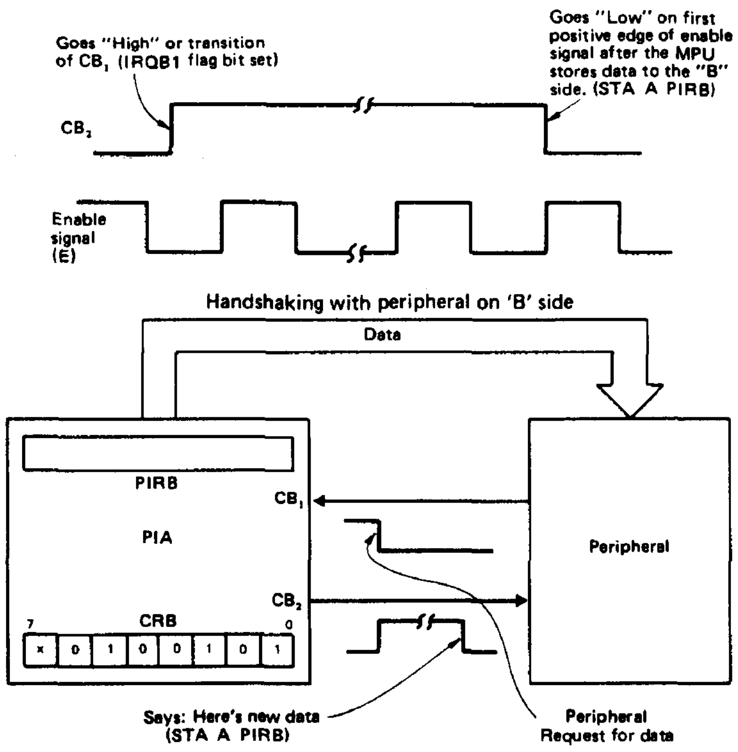


Figure 18 Bits 5, 4, 3 of CRB = 100 (Hand-shake Mode)

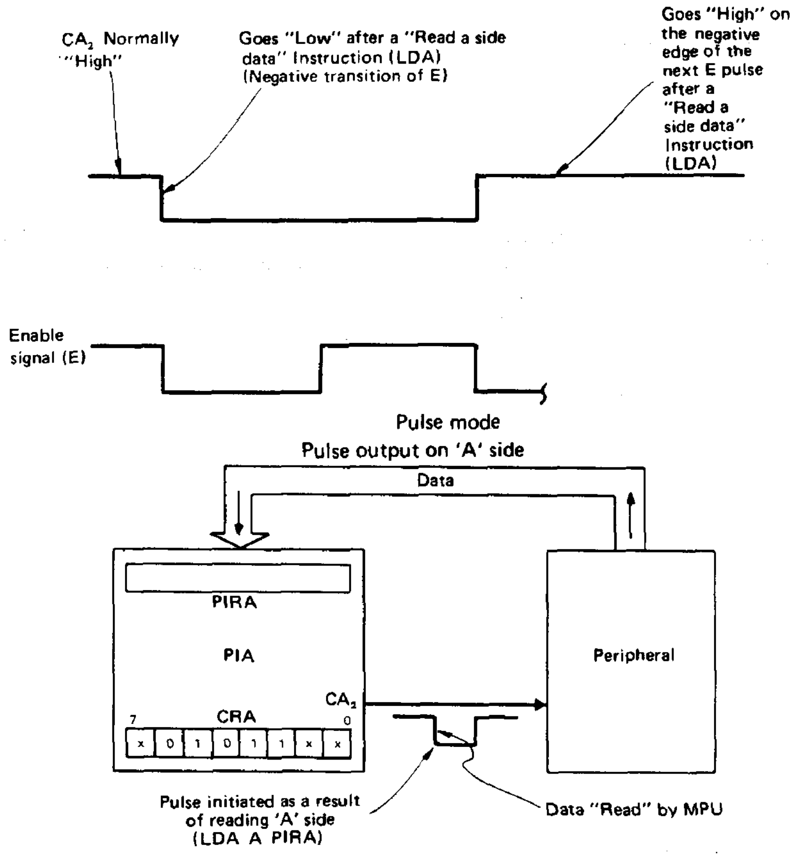


Figure 19 Bits 5, 4, 3 of CRA = 101 (Pulse Mode)

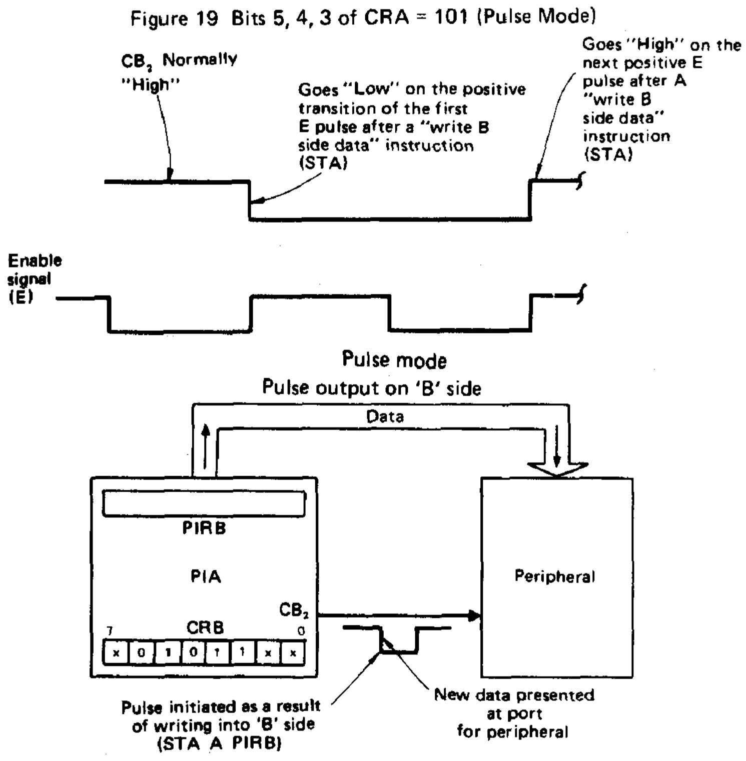


Figure 20 Bits 5, 4, 3 of CRB=101 (Pulse Mode)

■ SUMMARY OF CONTROL REGISTERS CRA AND CRB

Control registers CRA and CRB have total control of CA_1, CA_2, CB_1, and CB_2 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 through Bit 5 (6 bits), since Bit 6 and Bit 7 are set only by CA_1, CA_2, CB_1, or CB_2.

• Addressing PIAs

Before addressing PIAs, 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 interface resister have the same address, the control register bit 2 determines which register is being addressed. If Bit 2 in the control register is a logic "0", then the DDR is addressed. If Bit 2 in the control register is a logic "1", the peripheral interface 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, and 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, Bits 0, 1, 2, and 3 will be defined as inputs, while Bits 4, 5, 6, and 7 will be used as outputs. On the B side, all lines will be used as outputs.

```
     PIA1AD = 4004        (DDRA, PIRA)
     PIA1AC = 4005        (CRA)
     PIA1BD = 4006        (DDRB, PIRB)
     PIA1BC = 4007        (CRB)

1.   LDA A #%11110000     (4 outputs, 4 inputs)
2.   STA A     PIA1AD     (Loads A DDR)
3.   LDA A #%11111111     (All outputs)
4.   STA A     PIA1BD     (Loads B DDR)
5.   LDA A #%00000100     (Sets Bit 2)
6.   STA A     PIA1AC     (Bit 2 set in A control register)
7.   STA A     PIA1BC     (Bit 2 set in B control register)
```

Statement 2 addresses the DDR, since the control register (Bit 2) has not been loaded. Statements 6 and 7 load the control registers with Bit 2 set, so addressing PIA1AD or PIA1BD accesses the peripheral interface register.

• PIA Programming Via The Index Register

The program shown in the previous section can be accomplished using the Index Register.

```
1.  LDX   #$F004
2.  STX   PIA1AD    $F0→PIA1AD;$04→PIA1AC
3.  LDX   #$FF04
4.  STX   PIA1BD    $FF→PIA1BD;$04→PIA1BC
```

Using the index register in this example has saved six bytes of program memory as compared to the program shown in the previous section.

• Active Low Outputs

When all the outputs of given PIA port are to be active "Low" (True ≦ 0.4 volts), the following procedure should be used.

a) Set Bit 2 in the control register.
b) Store all 1s ($FF) in the peripheral interface register.
c) Clear Bit 2 in the control register.
d) Store all 1s ($FF) in the data direction register.
e) Store control word (Bit 2 = 1) in control register.

<Example>

The B side of PIA1 is set up to have all active low outputs. CB_1 and CB_2 are set up to allow interrupts in the HANDSHAKE MODE and CB_1 will respond to positive edges ("Low"-to-"High" transitions). Assume reset conditions. Addresses are set up and equated to the same labels as previous example.

```
1.  LDA A #4
2.  STA  A PIA1BC    Set Bit 2 in PIA1BC (control register)
3.  LDA B #$FF
4.  STA  B PIA1BD    All 1s in peripheral interface register
5.  CLR PIA1BC       Clear Bit 2
6.  STA  B PIA1BD    All 1s in data direction register
7.  LDA A #$27
8.  STA  A PIA1BC    00100111→→ control register
```

The above procedure is required in order to avoid outputs going "Low", to the active "Low" TRUE STATE, when all ls are stored to the data direction register as would be the case if the normal configuration procedure were followed.

• Interchanging RS_0 And RS_1

Some system applications may require movement of 16 bits of data to or from the "outside world" via two PIA ports (A side + B side). When this is the case it is an advantage to interconnect RS_1 and RS_0 as follows.

RS_0 to A1 (Address Line A1)
RS_1 to A0 (Address Line A0)

This will place the peripheral interface registers and control registers side by side in the memory map as follows.

Table	Example Address	
PIA1AD	$4004	(DDRA, PIRA)
PIA1BD	$4005	(DDRB, PIRB)
PIA1AC	$4006	(CRA)
PIA1BC	$4007	(CRB)

The index register or stackpointer may be used to move the 16-bit data in two 8-bit bytes with one instruction. As an example:

LDX PIA1AD   PIA1AD →→ IX_H: PIA1BD →→ IX_L

• PIA – After Reset

When the $\overline{RES}$ (Reset Line) has been held "Low" for a minimum of one microsecond, all registers in the PIA will be cleared.

Because of the reset conditions, the PIA has been defined as follows.

1. All I/O lines to the "outside world" have been defined as inputs.
2. CA_1, CA_2, CB_1, and CB_2 have been defined as interrupt input lines that are negative edge sensitive.
3. All the interrupts on the control lines are masked. Setting of interrupt flag bits will not cause $\overline{IRQA}$ or $\overline{IRQB}$ to go "Low".

■ SUMMARY OF CA_1-CB_1 PROGRAMMING

Bits 1 and 0 of the respective control registers are used to program the interrupt input control lines CA_1 and CB_1.

b1	b0	
0	0	b1 = Edge (0 = −, 1 = +)
0	1	b0 = Mask (0 = Mask, 1 = Allow)
1	0	
1	1	

Note that this is the same logic as Bits 4 and 3 for CA_2-CB_2 when CA_2-CB_2 are programmed as inputs.

■ SUMMARY OF CA_2-CB_2 PROGRAMMING

Bits 5, 4, and 3 of the control registers are used to program the operation of CA_2-CB_2.

	b5	b4	b3	
CA_2−CB_2 Input Mode →	0	0(−)	0	(Mask) CA_2−CB_2 Input Mode
	0	0(−)	1	(Allow) b4 = Edge (0 = −, 1 = +)
	0	1(+)	0	(Mask) b3 = Mask (0 = Mask,
	0	1(+)	1	(Allow) 1 = Allow)
CA_2−CB_2 Output Mode →	1	0	0	0 – Handshake Mode
	1	0	1	1 – Pulse Mode
	1	1	0	b3 Following Mode
	1	1	1	b3 Following Mode

I/O As Follow:
Control Lines:
CA_1 – Positive Edge, Allow Interrupt
CA_2 – Pulse Mode
CB_1 – Negative Edge, Mask Interrupt
CB_2 – Hand Shake Mode

Assume Reset Condition
PIA1AD
PIA1AC
PIA1BD
PIA1BC

PIA Configuration Solution

```
LDA A #$BC      10111100
STA A PIA1AD    I/O to DDRA
LDA A #$FF      1111 1111
STA A PIA1BD    I/O to DDRB
LDA A #$2F      0010 1111
STA A PIA1AC    To "A" Control
LDA A #$24      0010 0100
STA A PIA1BC    To "B" Control
```

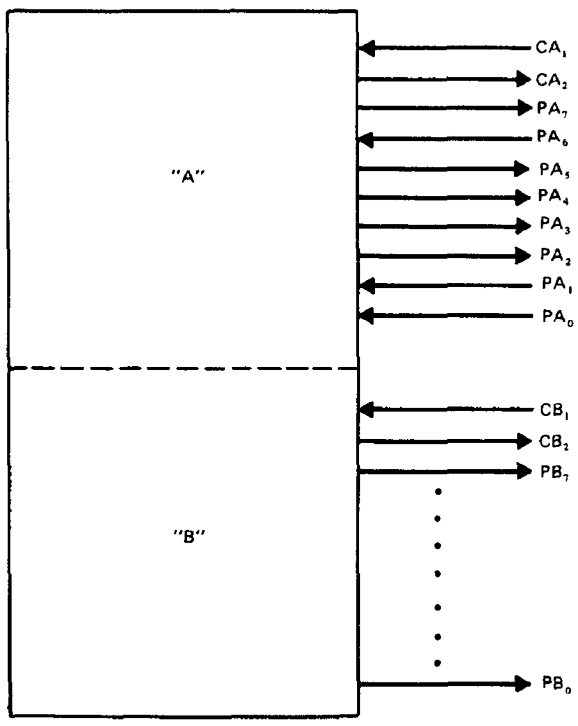


Figure 21 PIA Configuration Problem