commodore
semiconductor group
NMOS
950 Rittenhouse Rd., Norristown, PA 19403 • Tel.: 215/666-7950 • TWX: 510/660-4168

6525 TRI-PORT INTERFACE

CONCEPT . . .

The 6525 TRI-PORT Interface (TPI) is designed to simplify the implementation of complex I/O operations in microcomputer systems. It combines two dedicated 8-bit I/O ports with a third 8-bit port programmable for either normal I/O operation or priority interrupt/handshaking control. Depending on the mode selected, the 6525 can provide 24 individually programmable I/O lines or 16 I/O lines, 2 handshake lines and 5 priority interrupt inputs.

FEATURES:

- 24 individually programmable I/O lines or 16 I/O lines, 2 handshake lines and 5 interrupt inputs.
- Priority or non-priority interrupts
- Automatic handshaking
- Completely static operation
- Two TTL Drive Capability
- 8 directly addressable registers
- 1 MHz, 2MHz and 3MHz operation

6525 Addressing

6525 REGISTERS (Direct Addressing)

*000	R0	PRA—Port Register A
001	R1	PRB—Port Register B
010	R2	PRC—Port Register C
011	R3	DDRA—Data Direction Register A
100	R4	DDRB—Data Direction Register B
101	R5	DDRC—Data Direction Register C/Interrupt Mask Register
110	R6	CR—Control Register
111	R7	AIR—Active Interrupt Register

*NOTE: RS2, RS1, RS0 respectively

ORDER NUMBER:

MXS 6525 ___

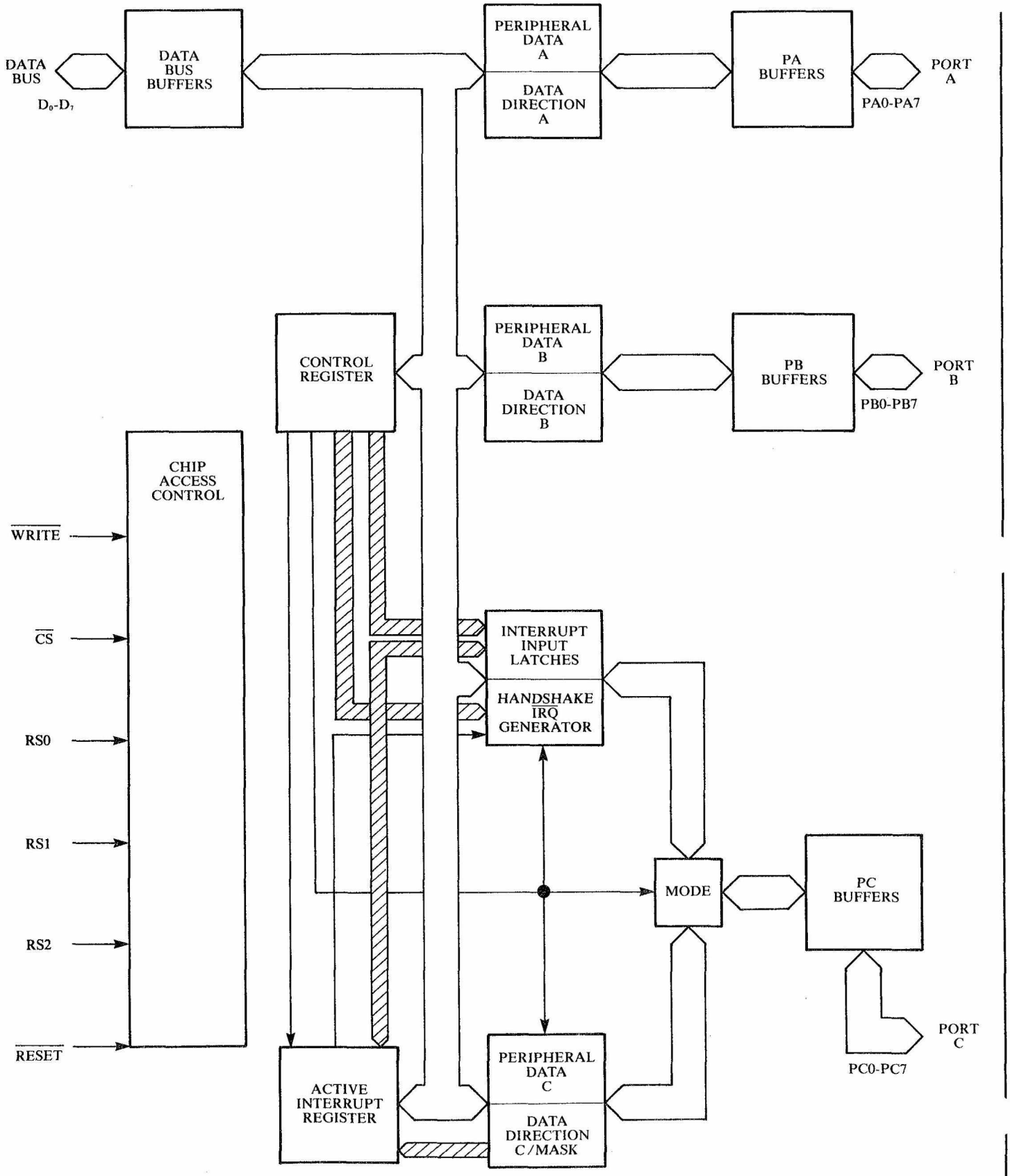
FREQUENCY RANGE
NO SUFFIX = 1 MHz
A = 2 MHz
B = 3 MHz

PACKAGE DESIGNATOR
C = CERAMIC
P = PLASTIC

6525 PIN CONFIGURATION

V_{SS}	1	40	DB7
PA0	2	39	DB6
PA1	3	38	DB5
PA2	4	37	DB4
PA3	5	36	DB3
PA4	6	35	DB2
PA5	7	34	DB1
PA6	8	33	DB0
PA7	9	32	PC7
PB0	10	31	PC6
PB1	11	30	PC5
PB2	12	29	PC4
PB3	13	28	PC3
PB4	14	27	PC2
PB5	15	26	PC1
PB6	16	25	PC0
PB7	17	24	RS0
$\overline{CS}$	18	23	RS1
$\overline{WRITE}$	19	22	RS2
V_{DD}	20	21	$\overline{RST}$

6525 INTERNAL ARCHITECTURE



MAXIMUM RATINGS

RATING	SYMBOL	VALUE	UNIT
SUPPLY VOLTAGE	V_{CC}	−0.3 to +7.0	V_{dc}
INPUT VOLTAGE	V_{in}	−0.3 to +7.0	V_{dc}
OPERATING TEMPERATURE RANGE	T_A	0 to +70	°C
STORAGE TEMPERATURE RANGE	T_{stg}	−55 to +150	°C

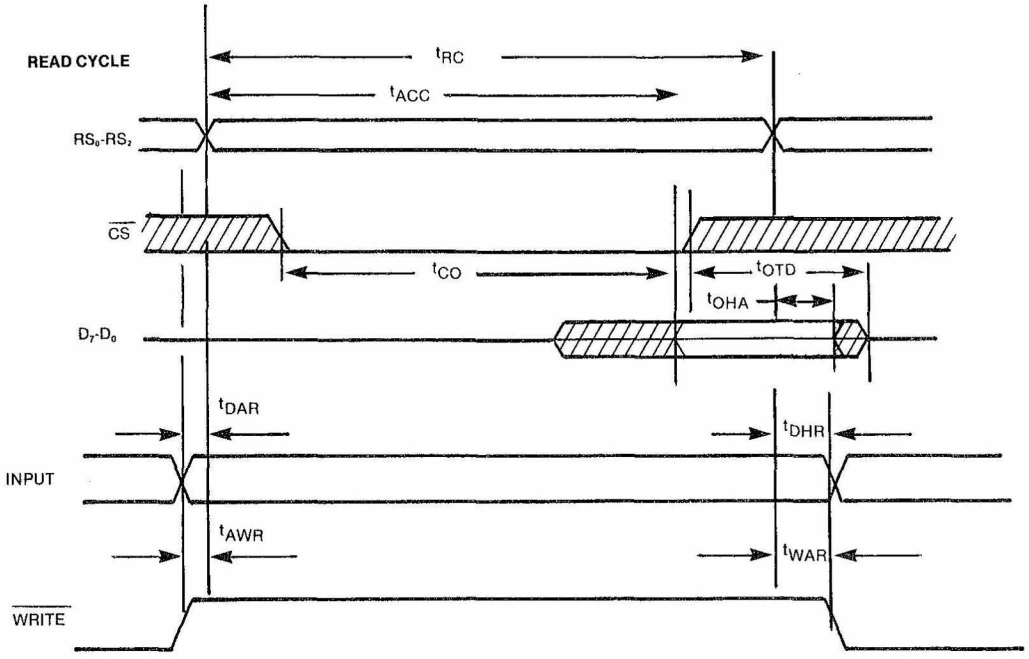
This device contains circuitry to protect the inputs against damage due to high static voltages, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this circuit.

CHARACTERISTICS (V_{CC} = 5.0 V ± 5%, V_{SS} = 0V, T_A = 0° to 70°C)

CHARACTERISTIC	SYMBOL	MIN	TYP	MAX	UNIT
Input High Voltage (Normal Operating Levels)	V_{IH}	+2.0	1.5	V_{CC}	Vdc
Input Low Voltage (Normal Operating Levels)	V_{IL}	−0.3	1.2	+.8	Vdc
Input Leakage Current V_{in} = 0 to 5.0 Vdc $\overline{WRITE}$ $\overline{RST}$, $\overline{CS}$, RS_0-RS_2	I_{IN}	0	±1.0	±2.5	μAdc
Three-State (Off State) Input Current (V_{in} = 0.4 to 2.4 Vdc, V_{CC} = max) D0-D7, PA0-PA7, PB0-PB7, PC0-PC7	I_{TSI}	0	±2.0	±10	μAdc
Output High Voltage (V_{CC} = min, Load = 200 μAdc)	V_{OH}	2.4	3.5	V_{CC}	Vdc
Output Low Voltage (V_{CC} = min, Load = 3.2 mAdc)	V_{OL}	V_{SS}	0.2	0.4	Vdc
Output High Current (Sourcing) (V_{OH} = 2.4 Vdc)	I_{OH}	−200	−1000	—	μAdc
Output Low Current (Sinking) (V_{OL} = 0.4 Vdc)	I_{OL}	3.2	—	—	mAdc
Supply Current	I_{CC}	—	50	100	mA
Input Capacitance (V_{in} = 0V, T_A = 25°C, f = 1.0 MHz) D0-D7, PA0-PA7, PB0-PB7, PC0-PC7 PC7, $\overline{WRITE}$ $\overline{RST}$, RS_0-RS_2, $\overline{CS}$	C_{in}	—	7	10	pF
Output Capacitance (V_{in} = 0V, T_A = 25°C, f = 1.0 MHz)	C_{out}	—	7	10	pF

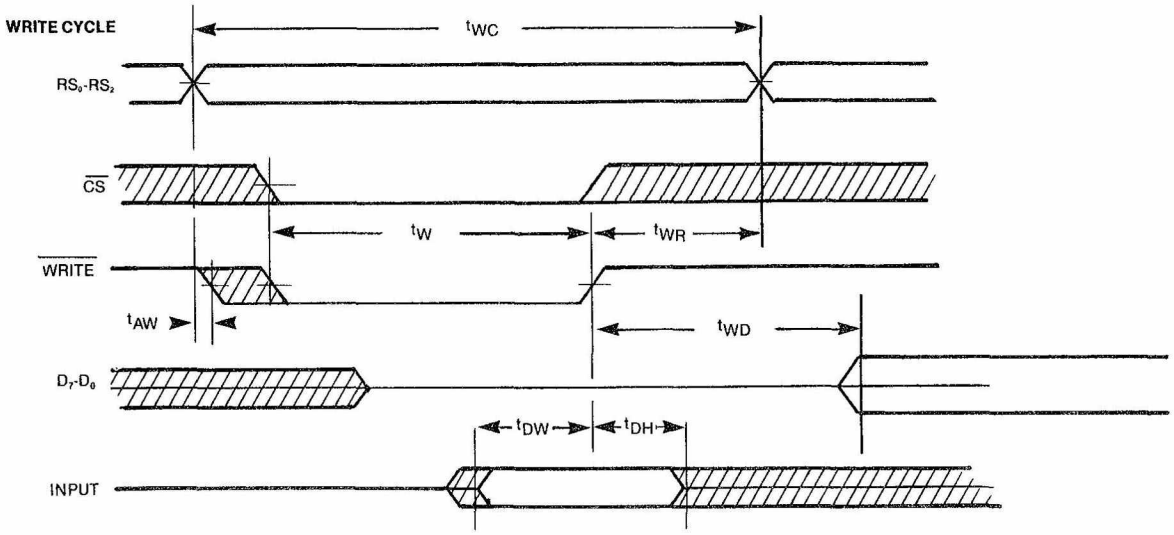
Note: Negative sign indicates outward current flow, positive indicates inward flow.

READ CYCLE



TIMING DIAGRAM

WRITE CYCLE



READ CYCLE

Symbol	Parameter	1MHz MIN	1MHz MAX	2MHz MIN	2MHz MAX	3MHz MIN	3MHz MAX	UNITS
t_{RC}	Read Cycle Time	700		350		220		nS
t_{ACC}	Access time	450		225		160		nS
t_{CO}	Chip Select to Output Valid	450		225		160		nS
t_{OTD}	Chip Deselected to Output Off	0	100	0	100	0	100	nS
t_{OHA}	Output Hold From Address Change	50		50		50		nS
t_{DAR}	Peripheral Data Set-Up	120		60		40		nS
t_{DHR}	Peripheral Data Hold	0		0		0		nS
t_{AWR}	Write to Address Setup	0		0		0		nS
t_{WAR}	Write to Address Hold	0		0		0		nS

WRITE CYCLE

Symbol	Parameter	1MHz MIN	1MHz MAX	2MHz MIN	2MHz MAX	3MHz MIN	3MHz MAX	UNITS
t_{WC}	Write Cycle Time	700		350		220		nS
t_{AW}	Address to write set-up time	0		0		0		nS
t_{W}	Write Pulse Width	450		225		160		nS
t_{WR}	Write Release Time	250		150		90		nS
t_{DW}	Data to Write Overlap	150		75		75		nS
t_{DH}	Data Hold	50		40		40		nS
t_{WD}	Write to Peripheral Output	1000		500		330		nS

6525 Internal Registers

	D7	D6	D5	D4	D3	D2	D1	D0
CR	CB_1	CB_0	CA_1	CA_0	IE_4	IE_3	IP	MC
AIR				A_4	A_3	A_2	A_1	A_0
DDRC When MC = 1				M_4	M_3	M_2	M_1	M_0
PRC When MC = 1	CB	CA	$\overline{IRQ}$	I_4	I_3	I_2	I_1	I_0

CA, CB Functional Description

CA OUTPUT MODES

CA_1	CA_0	MODE	DESCRIPTION
0	0	"Handshake" on Read	CA is set high on an active transition of the I_3 interrupt input signal and set low by a microprocessor "Read A Data" operation. This allows positive control of data transfers from the peripheral device to the microprocessor.
0	1	Pulse Output	CA goes low for I μS after a "Read A Data" operation. This pulse can be used to signal the peripheral device that data was taken.
1	0	Manual Output	CA set low.
1	1	Manual Output	CA set high.

CB OUTPUT MODES

CB_1	CB_0	MODE	DESCRIPTION
0	0	"Handshake" on Write	CB is set low on microprocessor "Write B Data" operation and is set high by an active transition of the I_4 interrupt input signal. This allows positive control of data transfers from the microprocessor to the peripheral device.
0	1	Pulse Output	CB goes low for I μS after a microprocessor "Write B Data" operation. This can be used to signal the peripheral device that data is available.
1	0	Manual Output	CB set low.
1	1	Manual Output	CB set high.

INTERRUPT MASK REGISTER DESCRIPTION

When the Interrupt Mode is selected (MC = 1), the Data Direction Register for Port C (DDRC) is used to enable or disable a corresponding interrupt input. For example: If $M_0 = 0$ then I_0 is disabled and any I_0 interrupt latched in the interrupt latch register will not be transferred to the AIR and will not cause IRQ to go low. The interrupt latch can be cleared by writing a zero to the appropriate bit in PRC.

PORT REGISTER C DESCRIPTION

Port Register C (PRC) can operate in two modes. The mode is controlled by bit MC in register CR. When MC = 0, PRC is a standard I/O port, operating identically to PRA & PRB. If MC = 1, then port register C is used for handshaking and priority interrupt input and output.

PRC When MC = 0:

PC_7	PC_6	PC_5	PC_4	PC_3	PC_2	PC_1	PC_0

PRC When MC = 1:

CB	CA	$\overline{IRQ}$	I_4	I_3	I_2	I_1	I_0

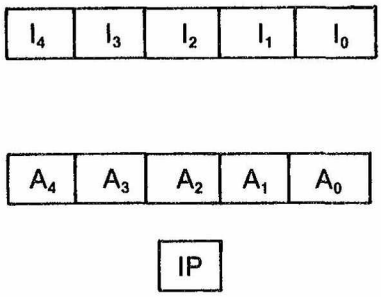
INTERRUPT EDGE CONTROL

Bits IE_4 and IE_3 in the control register (CR) are used to determine the active edge which will be recognized by the interrupt latch.

If IE_4 (IE_3) = 0 then I_4 (I_3) latch will be set on a negative transition of I_4 (I_3) input.

If IE_4 (IE_3) = 1 then I_4 (I_3) latch will be set on a positive transition of the I_4 (I_3) input.

All other interrupt latches (I_2, I_1, I_0) are set on a negative transition of the corresponding interrupt input.



Interrupt Latch Register
Clears on Read of AIR Using Following Equation:

ILR ← ILR ⊕ AIR

Active Interrupt Register
Clears following read of AIR

Interrupt Priority Select
IP = 0 No Priority
IP = 1 Interrupts Prioritized

FUNCTIONAL DESCRIPTION

1. IP=0:No Priority

In this mode, the first interrupt latched into the Interrupt Latch Register (ILR) is transferred immediately into the Active Interrupt Register (AIR) and IRQ is pulled low (assuming the interrupt isn't masked). Upon reading the AIR, IRQ is reset high, the interrupt latch in the ILR is cleared as described above and the chip is ready to receive new interrupts. In non-priority mode, if multiple interrupts occur simultaneously, all corresponding bits in the AIR will be set. When AIR is read, all associated bits in the ILR will be cleared, therefore it is a software effort to recognize that multiple interrupts have occurred and to service them appropriately.

2. IP=1:Priority Interrupt

In this mode, the interrupts are prioritized in the following order: 14>13>12>11>10.

With priority selected, only one bit in the AIR can be set at any time. When an interrupt occurs, it is latched in the ILR, then priority is compared. If the interrupt is of highest priority, it will be transferred to the AIR and an Interrupt Request will be generated. To fully understand the operation of the priority interrupts, consider the following examples:

A. The least complicated case involves a single interrupt which the processor services completely before another interrupt occurs:
   1. An interrupt is received on I1.
   2. Bit I1 is set in the ILR.
   3. IRQ goes low.
   4. Bit A1 is set in the AIR.
   5. Processor responds to IRQ by reading AIR to determine which interrupt has occurred.
   6. A1 is pushed onto interrupt stack and I1 is cleared.
   7. A1 is cleared and IRQ goes high.
   8. Processor services interrupt and signals completion by writing to AIR.
   9. Interrupt stack automatically pops on write to AIR, completing interrupt sequence.

B. The next case occurs when an interrupt is in the process of being serviced and a lower priority interrupt occurs:

   1. Interrupt I1 is received and latched.
   2. IRQ goes low and A1 is set.
   3. Processor reads AIR and determines I1 has occurred.
   4. A1 is pushed onto interrupt stack and I1 is cleared.
   5. A1 is cleared and IRQ goes high.
   6. Processor services I1, during which an I0 interrupt occurs and is latched.
   7. The interrupt stack prevents I0 from interrupting I1 service.
   8. Upon completion of I1 service, processor writes AIR, popping interrupt stack.
   9. IRQ goes low and A0 is set, beginning a new interrupt sequence for I0.

C. The final case occurs when an interrupt is in the process of being serviced and a higher priority interrupt occurs:

   1. Interrupt I1 is received and latched.
   2. IRQ goes low and A1 is set.
   3. Processor reads AIR and determines I1 has occurred.
   4. A1is pushed onto interrupt stack and I1 is cleared.
   5. A1 is cleared and IRQ goes high.
   6. Processor services I1, during which an I2 interrupt occurs and is latched.
   7. IRQ goes low and A2 is set.
   8. I1 service is interrupted and processor automatically stacks program counter.
   9. Processor reads AIR and determines A2 has occurred.
   10. A2 is pushed onto interrupt stack and I2 is cleared.
   11. A2 is cleared and IRQ goes high.
   12. Processor services I2.
   13. Upon completion of I2 service, processor writes AIR, popping the interrupt stack, which restores A1 on top of stack.
   14. Return From Interrupt causes processor to resume service of I1 which had been interrupted.
   15. Interrupt stack prevents lower priority interrupt of resumed I1 service.
   16. Upon completion of I1 service, processor writes AIR, popping interrupt stack and completing interrupt sequence.

NOTE: A five-level interrupt stack maintains priority information for all interrupts under service. This stack is pushed on any READ of AIR and popped on any WRITE to AIR. No extraneous reading or writing of AIR should be performed as this will cause unwanted stack operations.
The only time a READ of AIR should occur is in response to an interrupt request.
The only time a WRITE to AIR should occur is to signal the 6525 that an interrupt service is complete.