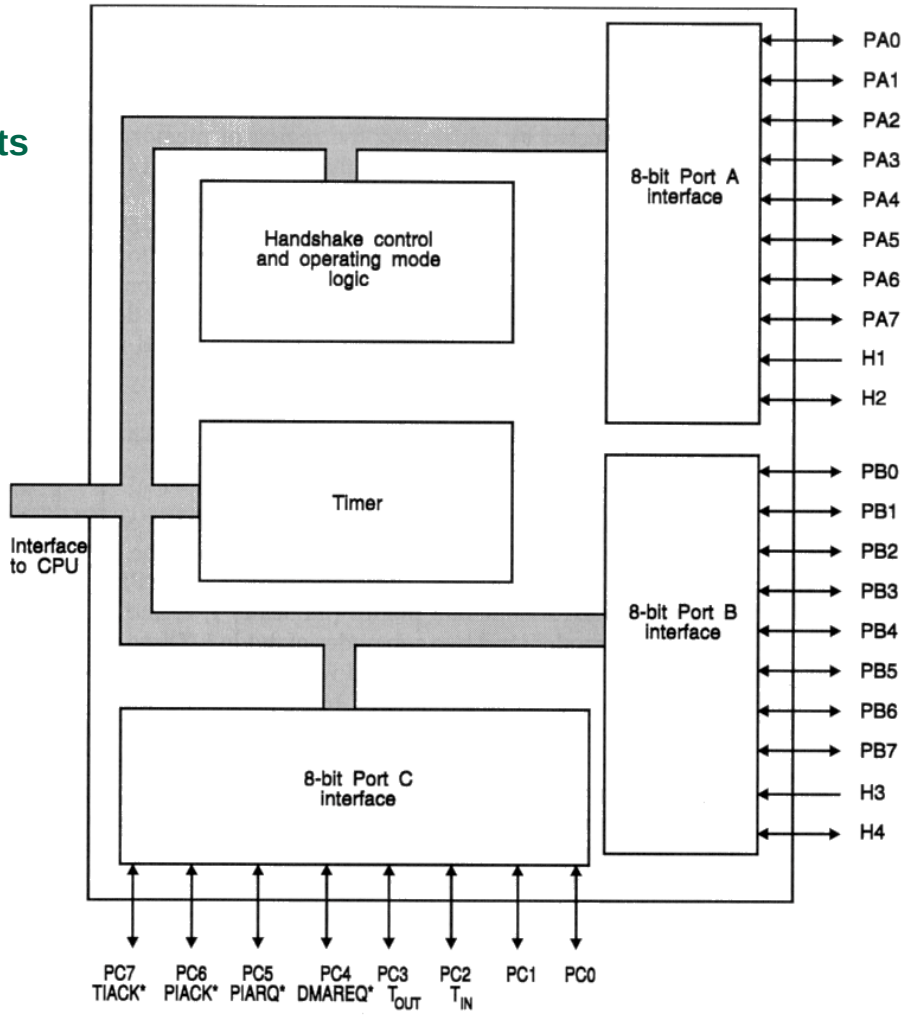
Lecture 11: PI/T parallel I/O, part I

- **General description of the parallel I/O function**
- **Buffering**
- **Handshaking**
  - Input and Output transfers
  - Timing Diagrams
- **Register model of the 68230**
  - Port General Control Register (PGCR)
  - Port Service Request Register (PSSR)
  - Port {A,B,C} Data Direction Register (PxDDR)
  - Port Interrupt Vector Register (PIVR)
  - Port {A,B,C} Data Register (PxDR)
  - Port {A,B} Alternate Data Register (PxADR)
  - Port Status Register (PSR)

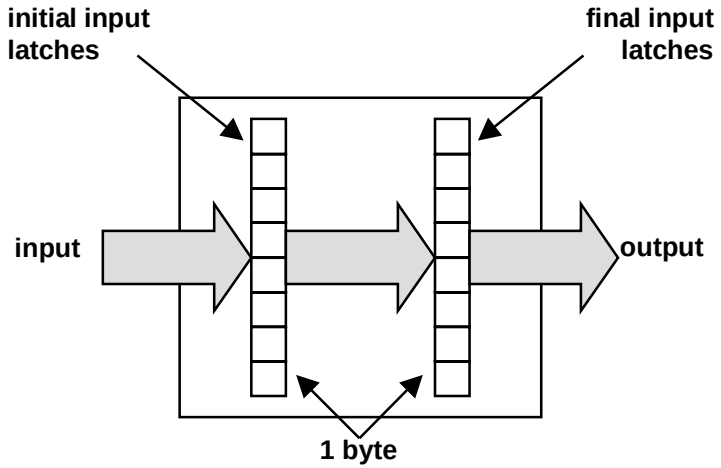
Parallel I/O general description

- **The parallel function has three ports**
  - Two independent 8-bit ports (and B)
  - A third dual-function port C
- **Ports A and B be can be used as I/O ports with various *handshaking* and *buffering* capabilities in four different modes**
  - Mode 0: Unidirectional 8-bit
  - Mode 1: Unidirectional 16-bit
  - Mode 2: Bidirectional 8-bit
  - Mode 3: Bidirectional 16-bit
- **Port C can be used as**
  - a simple 8-bit port without handshaking or double-buffering
  - an interrupt interface to the timer
  - an interrupt interface to parallel I/O
  - a support interface for DMA operation

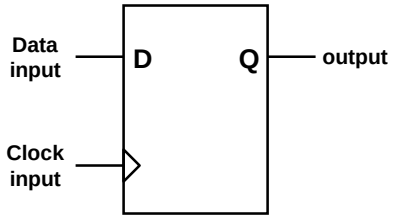


Double buffering and latches

- **The PI/T data is double-buffered**
  - The PI/T is able to receive a new input while storing the previous input
  - Data can be transferred at almost the maximum rate at which the CPU can read the PI/T, without information being lost



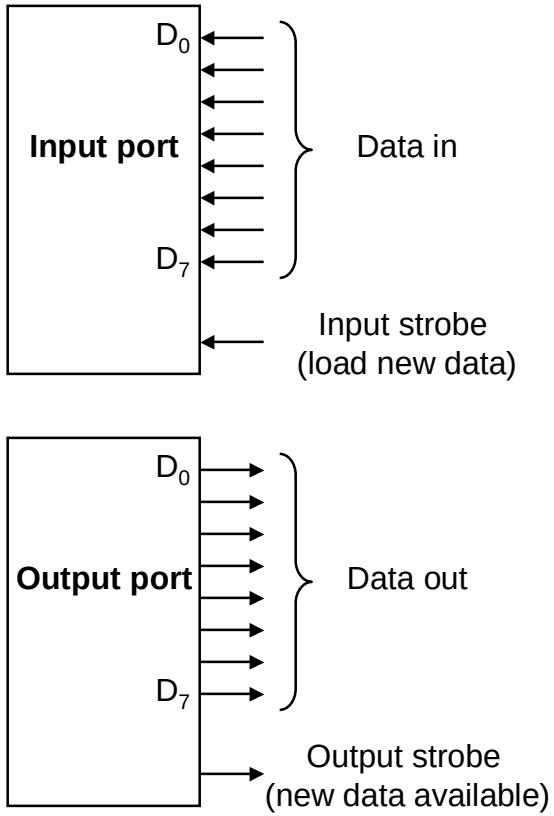
- **D-latch**
  - An up-arrow represents a 0-to-1, or positive-edge, transition
  - A down-arrow represents a 1-to-0, or negative-edge, transition
  - Data is latched on Q only during a positive-edge transition of the clock



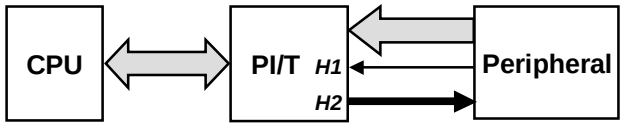
D	CLK	Q
0	↑	0
1	↑	1
x	0,1, ↓	No change

Handshaking

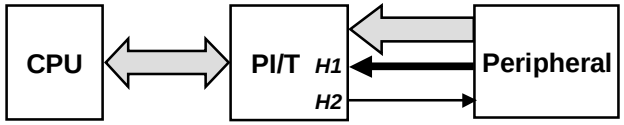
- **Handshaking**
  - Permits data transfers to be **interlocked** with an external activity, so that data is moved at a rate in keeping with the peripheral's capacity
  - **Interlocked** means that the next action cannot go ahead until the current action has been completed
  - An I/O transfer using handshaking is also called a **closed-loop** data transfer
- **An input port is informed of the arrival of new data through an input strobe**
- **An output port announces the availability of new data through an output strobe**
- **In the PI/T parallel interface, handshaking is achieved by means of four handshake signals**
  - H1 and H2 for port A
  - H3 and H4 for port B



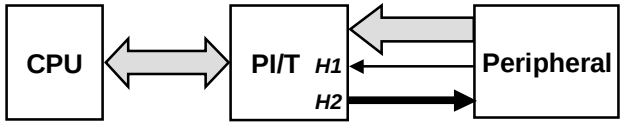
Interlocked handshake input mode



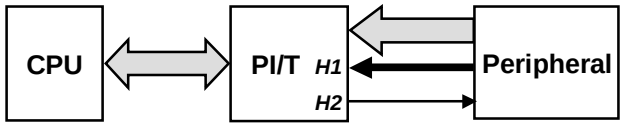
- Let's assume a data transfer through Port A. The PI/T has two data transfer control lines: an edge-sensitive input H1 and an output H2
- The PI/T indicates the peripheral that it is ready to accept new data by asserting its H2 output



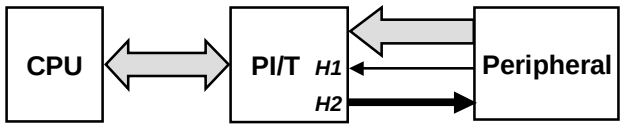
- The peripheral forces an active transition on the PI/T's H1 input, informing the PI/T that data is now available on its data input port
- Asserting H1 sets a status bit within the PI/T and generates an interrupt request to the 68000 if the PI/T is programmed to do so



- The PI/T negates its H2 output to inform the peripheral that the data has been received
- At the same time, the peripheral is indicating that it is no longer in a position to receive new data

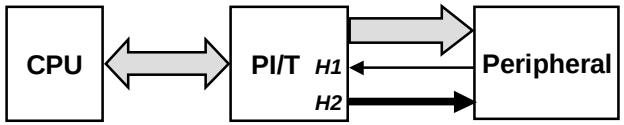


- The peripheral negates H1 to inform the PI/T that it has acknowledged the data transfer

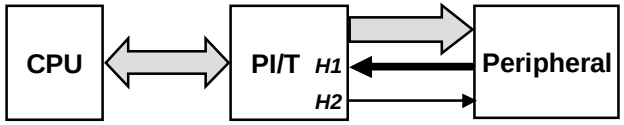


- The PI/T asserts H2 to indicate that it is once more ready to receive data from the peripheral
- At this stage, the system is in the same state as in the first step of this sequence, and a new cycle may commence

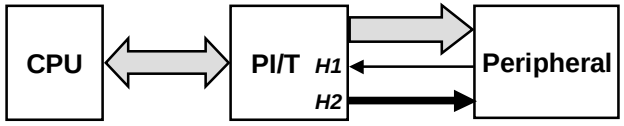
Interlocked handshake output mode



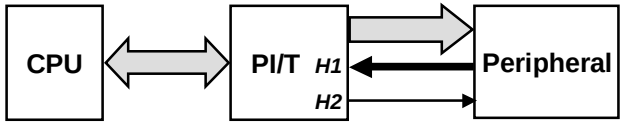
- The CPU loads data into the PI/T's output register, causing the H2 output to be asserted after a delay of two clock cycles
- The assertion of H2 indicates to the peripheral that the data is available



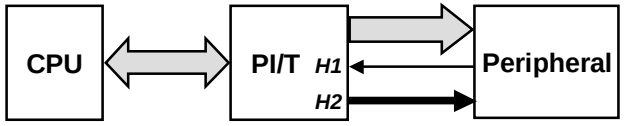
- The peripheral asserts H1 to indicate that it has read the data



- The assertion of H1 causes the PI/T to negate H2 indicating that it has acknowledged the peripheral's receipt of data



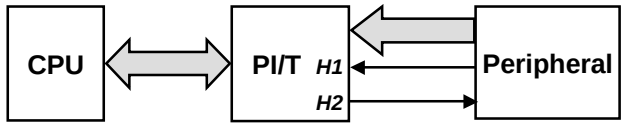
- The peripheral negates H1 to inform the PI/T that it is once more ready for data



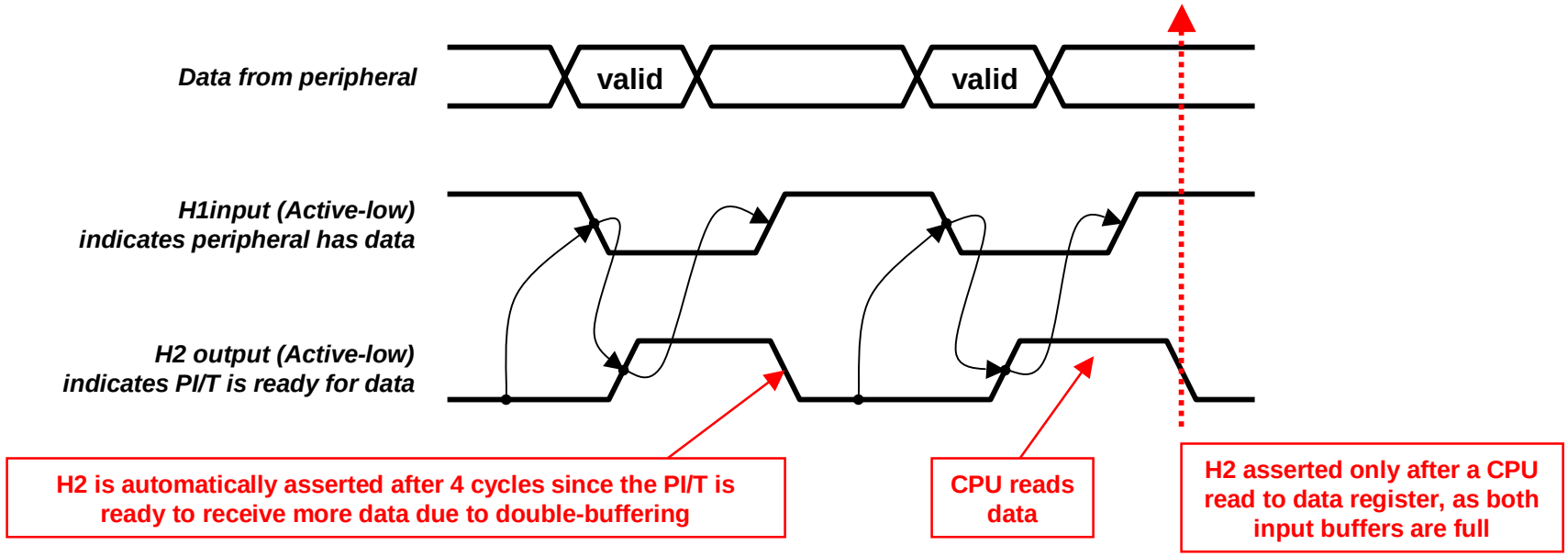
- The CPU loads new data into the PI/T output register, and H2 is asserted again to indicate a data-ready state

Coordination of double buffering and handshaking during input

- **The timing diagram depicts two consecutive inputs cycles using double-buffered input**
  - In the first cycle, H2 is negated after H1 has been asserted, and asserted automatically after approximately four clock cycles

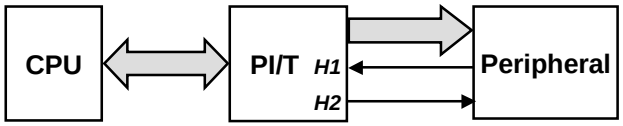


  - H2 is asserted this second time because the input has been transferred from the PI/T's ***initial input latches*** to its ***final input latches***, and the initial input latches are once more ready to accept data
  - However, on the second input cycle, H2 remains inactive high because both input buffers are full. The H2 output re-asserts itself only when the CPU reads from the PI/T and empties the final input latches

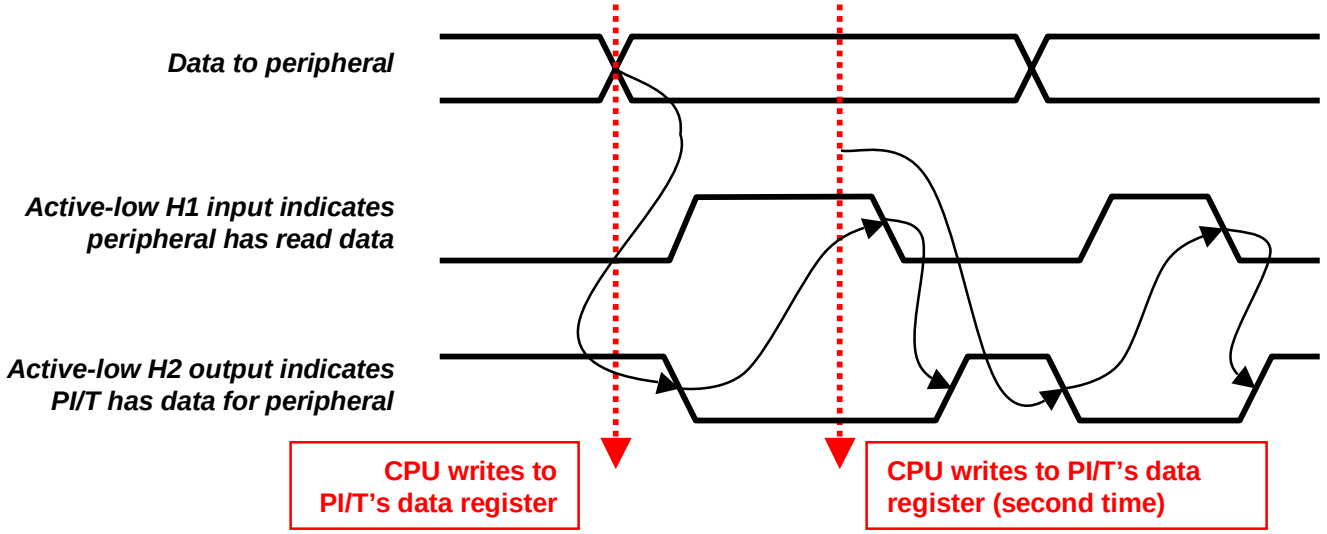


Coordination of double buffering and handshaking during output

- **The timing diagram depicts two consecutive output cycles using double-buffered output**
  - Initially both the PI/T's output buffers are empty



  - When data is first loaded into the PI/T by the CPU, the data is transferred to the IC's output terminals and H2 is asserted
  - When the next write to the PI/T's data register is made, the data is not immediately transferred to the output buffer, and therefore the PI/T is in a busy state and cannot accept new data. Only when H1 is asserted by the peripheral does the PI/T transfer its latest data to the output register. Now the PI/T may once more accept data from the CPU
  - As in the case of input transfers, the CPU knows when the PI/T is ready for data by examining the state of the H1 flag bit: H1S



Register model of the 68230

	Offset	7	6	5	4	3	2	1	0	
R0	$01	Port Mode Control		H34 Enable	H12 Enable	H4 Sense	H3 Sense	H2 Sense	H1 Sense	*Port General Control Register*
R1	$03	×	SVCRQ Select		Interrupt PFS		Port Interrupt Priority Control			*Port Service Request Register*
R2	$05	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	*Port A Data Direction Register*
R3	$07	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	*Port B Data Direction Register*
R4	$09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	*Port C Data Direction Register*
R5	$0B	Interrupt Vector Number						Interrupt Source		*Port Interrupt Vector Register*
R6	$0D	Port A Sub-mode		H2 Control			H2 Int. Enable	H1 SVRQ	H1 Status Ctrl	*Port A Control Register*
R7	$0F	Port B Sub-mode		H4 Control			H4 Int. Enable	H3 SVRQ	H3 Status Ctrl	*Port B Control Register*
R8	$11	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	*Port A Data Register*
R9	$13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	*Port B Data Register*
R10	$15	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	*Port A Alternate Register*
R11	$17	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	*Port B Alternate Register*
R12	$19	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	*Port C Data Register*
R13	$1B	H4 Level	H3 Level	H2 Level	H1 Level	H4S	H3S	H2S	H1S	*Port Status Register*
	$1D	×	×	×	×	×	×	×	×	*(null)*
	$1F	×	×	×	×	×	×	×	×	*(null)*
R14	$21	TOUT/TIACK* Control			ZD Control	×	Clock Control		Timer Enable	*Timer Control Register*
R15	$23	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	*Timer Interrupt Vector Register*
	$25	×	×	×	×	×	×	×	×	*(null)*
R16	$27	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	*Counter Preload Register High*
R17	$29	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	*Counter Preload Register Middle*
R18	$2B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	*Counter Preload Register Low*
	$2D	×	×	×	×	×	×	×	×	*(null)*
R19	$2F	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	*Count Register High*
R20	$31	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	*Count Register Middle*
R21	$33	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	*Count Register Low*
R22	$35	×	×	×	×	×	×	×	ZDS	*Timer Status Register*
	$37	×	×	×	×	×	×	×	×	*(null)*
	$39	×	×	×	×	×	×	×	×	*(null)*
	$3B	×	×	×	×	×	×	×	×	*(null)*
	$3D	×	×	×	×	×	×	×	×	*(null)*
	$3F	×	×	×	×	×	×	×	×	*(null)*

Brief overview of parallel I/O registers

- **Port General Control Register (PGCR)**
  - Selection of I/O modes (0, 1, 2 and 3) and handshaking signals (H1, H2, H3 and H4)
- **Port Service Request Register (PSSR)**
  - Selection of Port C functions: DMA requests, IRQ/IACK signals and handshaking signal priority
- **Port {A,B,C} Data Direction Register (PxDDR)**
  - Selection of individual port bits as inputs or outputs
- **Port Interrupt Vector Register (PIVR)**
  - Storage of vector number for vectored interrupts
- **Port {A,B} Control Register (PxCR)**
  - Selection of port sub-modes and handshake signals operation

Next lecture

- **Port {A,B,C} Data Register (PxDR)**
  - Contents of the I/O ports
- **Port {A,B} Alternate Data Register (PxADR)**
  - Instantaneous logic levels of the I/O pins of the port
- **Port Status Register (PSR)**
  - Status information of the handshake signals

Port General Control Register

- **PGCR7-PGCR6**
  - Select the operating mode of the PI/T
- **PGCR5-PGCR4**
  - Enables the handshake pairs H3-H4 and H1-H2. These bits have to be set before we can make use of the control inputs and outputs. Doing this avoids spurious operation of the handshake lines before the PI/T has been fully configured
- **PGCR3-PCGR0**
  - Determine the *sense* of the four handshake lines. These control lines can be programmed to be active-low or active-high

Bit	PGCR7	PGCR6	PGCR5	PGCR4	PGCR3	PGCR2	PGCR1	PGCR0
Function	Port mode control		H34 enable	H12 enable	H4 sense	H3 Sense	H2 Sense	H1 Sense

PGCR7	PGCR6	Port mode control
0	0	Mode 0
0	1	Mode 1
1	0	Mode 2
1	1	Mode 3

PGCR5	H3, H4 control
0	H34 disable
1	H34 enable

PGCR4	H1, H2 control
0	H12 disable
1	H12 enable

PGCR3-0	Sense (Assertion Level)
0	LOW
1	HIGH

Port Service Request Register

- **PSSR6-PSSR5 (service request)**
  - Determines whether the PI/T generates an interrupt or a DMA request
- **PSSR4-PSSR3 (operation select)**
  - Determines whether two of the dual-function pins belong to port C or perform special-purpose functions
- **PSSR2-PSSR0 (interrupt-priority control)**

Bit	PSRR7	PSRR6	PSRR5	PSRR4	PSRR3	PSRR2	PSRR1	PSRR0
Function	×	SRVRQ (DMA control)		Interrupt control		Port interrupt priority		

PSRR6	PSRR5	Interrupt pin function	
0	×	PC4/DMAREQ* = PC4	DMA not used
1	0	PC4/DMAREQ* = DMAREQ*	Associated with double-buffered transfers controlled by H1 H1 does not cause interrups in this mode
1	1	PC4/DMAREQ* = DMAREQ*	Associated with double-buffered transfers controlled by H3 H3 does not cause interrups in this mode

PSRR4	PSRR3	Interrupt pin function	
0	0	PC5/PIRQ* = PC5	No interrupt support
		PC6/PIACK* = PC6	No interrupt support
0	1	PC5/PIRQ* = PIRQ*	Autovectored interrupt supported
		PC6/PIACK* = PC6	Autovectored interrupt supported
1	0	PC5/PIRQ* = PC5	
		PC6/PIACK* = PIACK*	
1	1	PC5/PIRQ* = PIRQ*	Vectored interrupt supported
		PC6/PIACK* = PIACK*	Vectored interrupt supported

PSRR2	PSRR1	PSRR0	Order of priority interrupt Higuest ⟵⟶Lowest			
0	0	0	H1S	H2S	H3S	H4S
0	0	1	H2S	H1S	H3S	H4S
0	1	0	H1S	H2S	H4S	H3S
0	1	1	H2S	H1S	H4S	H3S
1	0	0	H3S	H4S	H1S	H2S
1	0	1	H3S	H4S	H2S	H1S
1	1	0	H4S	H3S	H1S	H2S
1	1	1	H4S	H3S	H2S	H1S

Port Data Direction Registers

- **Port Data Direction Registers: PADDR,PBDDR and PCDDR**
  - Select the direction and buffering characteristics of each of the appropriate port pins
    - A logical ONE makes the corresponding pin act as an OUTPUT
    - A logical ZERO makes the corresponding pin act as an INPUT
  - Port C behaves in the same fashion and determines whether each dual-function chosen for port C operation is an input or an output

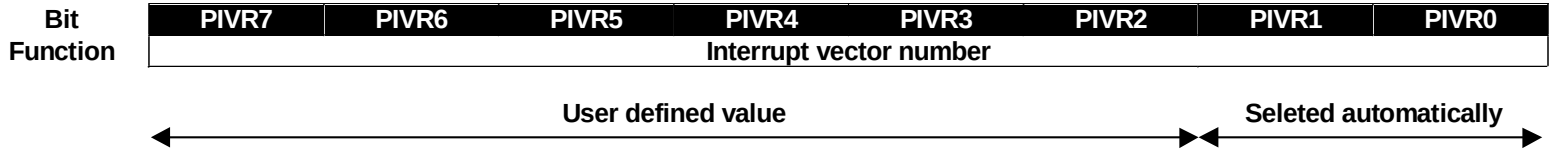
Bit	PADDR7	PADDR6	PADDR5	PADDR4	PADDR3	PADDR2	PADDR1	PADDR0
Function	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit	PBDDR7	PBDDR6	PBDDR5	PBDDR4	PBDDR3	PBDDR2	PBDDR1	PBDDR0
Function	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit	PCDDR7	PCDDR6	PCDDR5	PCDDR4	PCDDR3	PCDDR2	PCDDR1	PCDDR0
Function	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Port Interrupt Vector Register

- **When the parallel port section executes a vectored interrupt, it supplies the 68000 with one of four possible interrupt vector numbers**
  - Each of these numbers is associated with a specific interrupt cause
  - The upper-order six bits must be supplied by the programmer
  - The two least-significant bits are determined by the interrupt source
- **This arrangement has been implemented to avoid the need for four separate vector number registers**



PIVR1	PIVR0	Interrupt source
0	0	H1
0	1	H2
1	0	H3
1	1	H4

Port Alternate/Data Registers

- **Port Data Registers: PADR, PBDR and PCDR**
  - PADR and PBDR are holding registers between the CPU side of the PI/T and its I/O pins and internal buffer registers
  - PCDR is a holding register for moving data to and from port C or its alternate-function pins. The exact nature of an information transfer depends on the type of cycle being executed (read or write) and on the way in which port C is configured
- **Port Alternate Registers: PAAR and PBAR**
  - Provide another way of reading the state of ports A and B.
  - These registers are READ-ONLY and their contents reflect the *instantaneous* logic levels at the I/O pins.
  - These registers bypass the selected operating modes of ports A and B: if you want to know the state of port A and B you just read from PAAR or PBAR

Operation	Port C function		Alternate function	
	PCDDR = 0 (input)	PCDDR = 1 (output)	PCDDR = 0 (input)	PCDDR = 1 (output)
Read PCDR	Read pin	Read output register	Read pin	Read output register
Write PCDR	Output register buffer disabled	Output register buffer disabled	Write to output register	Write to output register

Port Status Register

- **PSR7-PSR4**
  - These bits show the *instantaneous* level at the respective handshake pins, and are independent of handshake pins sense bits in the PGCR
- **PSR3-PSR0**
  - These bits are the handshake status bits: H4S-H1S
  - They are set or cleared as specified by the appropriate operating mode
    - For example, if H1 is configured as active-low (by clearing bit 0 of the PGCR), an electrically low level at the H1 pin will set PSR0 to 1 to indicate that the status bit HS1 has been set

Bit	PSR7	PSR6	PSR5	PSR4	PSR3	PSR2	PSR1	PSR0
Function	H4 level	H3 level	H2 level	H1 level	H4S	H3S	H2S	H1S