The Motorola 68230 Parallel Interface Timer (PI/T)
• A general purpose Parallel Interface and Timer, PI/T chip that offer several very complex modes of operation.

Port A
- Port A Data Register, PADDR: $0FF010
- Data Direction Register, PADDR: $0FF004
- Port A Control Register, PACR: $0FF00C

Port B
- Port B Data Register, PBDR: $0FF012
- Data Direction Register: PBDDR: $0FF006
- Port B Control Register, PBCR: $0FF00E

Port General
- Control Register, PGCR: $0FF000
- Status Register, PSR: $0FF1A
68230 Block Diagram
## 68230 Registers Address Equates

<table>
<thead>
<tr>
<th>Register</th>
<th>EQU</th>
<th>Base Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIT</td>
<td>EQU</td>
<td>$0FF000</td>
<td>Base Address of PI/T</td>
</tr>
<tr>
<td>PGCR</td>
<td>EQU</td>
<td>PIT</td>
<td>Address of Port General Control Register</td>
</tr>
<tr>
<td>PSRR</td>
<td>EQU</td>
<td>PIT+2</td>
<td>Port service request register</td>
</tr>
<tr>
<td>PADDR</td>
<td>EQU</td>
<td>PIT+4</td>
<td>Data direction register A</td>
</tr>
<tr>
<td>PBDDR</td>
<td>EQU</td>
<td>PIT+6</td>
<td>Data direction register B</td>
</tr>
<tr>
<td>PACR</td>
<td>EQU</td>
<td>PIT+$0C</td>
<td>Port A control register</td>
</tr>
<tr>
<td>PBCR</td>
<td>EQU</td>
<td>PIT+$0E</td>
<td>Port B control register</td>
</tr>
<tr>
<td>PADR</td>
<td>EQU</td>
<td>PIT+$10</td>
<td>Port A data register</td>
</tr>
<tr>
<td>PBDR</td>
<td>EQU</td>
<td>PIT+$12</td>
<td>Port B data register</td>
</tr>
<tr>
<td>PSR</td>
<td>EQU</td>
<td>PIT+$1A</td>
<td>Port status register</td>
</tr>
<tr>
<td>TCR</td>
<td>EQU</td>
<td>PIT+$20</td>
<td>Timer control register</td>
</tr>
<tr>
<td>TSR</td>
<td>EQU</td>
<td>PIT+$34</td>
<td>Timer status register</td>
</tr>
</tbody>
</table>
68230 Parallel I/O

Input Handshake Timing

Handshaking Lines

(PIT ready)

Parallel Port Data

H3 or H1

H4 or H2

(PB or PA)

valid

Data Latched in PIT

Data to CPU (or Buffer)
68230 Parallel I/O

Output Handshake Timing

CPU Write to port (or buffer)

valid

Parallel Port Data

H3 or H1

H4 or H2

(PB or PA)

Handshaking Lines

(Peripheral ready)

(PIT ready)
68230 Parallel I/O Data
Latching/Buffering

CPU

Single Buffered

Output: single-buffered
or double-buffered

Double Buffered

OUT BUFFER

PORT A
or
PORT B
Data

Input: Non-latched
or double-buffered

Non-Latched

DOUBLE BUFFERED

IN BUFFER

Double Buffered

Port A

CPU

Double Buffered

Output: single-buffered
or double-buffered

Non-Latched

IN PORT

Double Buffered

Port A

CPU

Double Buffered

Output: single-buffered
or double-buffered

Non-Latched

IN PORT

Double Buffered

Port A
• **Mode 0**
  - **Unidirectional 8-bit, separate PA & PB**
    - Submode 00 - Double Buffered in, Single Buffered out
    - Submode 01 - Non-Latched in, Double Buffered out
    - Submode 1X - Non-Latched in, Single Buffered out
  - **Mode 1**
    - **Unidirectional 16-bit, combined PA**
    - Submode X0 - Double Buffered (DB) in, Single Buffered out
    - Submode X1 - Non-latched (NL) in, Double Buffered (DB) out
  - **Mode 2**
    - **Bidirectional 8-bit, separate PA & PB**
    - Port A - NL in, SB out (No handshake, unidirectional per bit)
    - Port B - DB bidirectional (H1, H2 for output & H3, H4 for input)
  - **Mode 3**
    - **Bidirectional 16-bit, combined PA & PB**
    - PA & PB - DB bidirectional (H1, H2 for output & H3, H4 for input)
# Format of Port General Control Register

**PGCR**

<table>
<thead>
<tr>
<th>PGCR7</th>
<th>PGCR6</th>
<th>PGCR5</th>
<th>PGCR4</th>
<th>PGCR3</th>
<th>PGCR2</th>
<th>PGCR1</th>
<th>PGCR0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port Mode Control</td>
<td>H34</td>
<td>H12</td>
<td>H4</td>
<td>H3</td>
<td>H2</td>
<td>H1</td>
<td></td>
</tr>
<tr>
<td>00 Mode 0</td>
<td>Enable</td>
<td>Enable</td>
<td>sense</td>
<td>sense</td>
<td>sense</td>
<td>sense</td>
<td></td>
</tr>
<tr>
<td>01 Mode 1</td>
<td>0 Disable</td>
<td>0 Active low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 Mode 2</td>
<td>1 Enable</td>
<td>1 Active high</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

\[
\text{PGCR} = \%00110000
\]

**Means:**

- Mode 0, Unidirectional 8-bit, separate PA & PB
- Both H34 and H12 handshaking enabled
- H4-H4 active low
### Format of Port A Control Register in Mode 0

<table>
<thead>
<tr>
<th>PACR7</th>
<th>PACR6</th>
<th>PACR5</th>
<th>PACR4</th>
<th>PACR3</th>
<th>PACR2</th>
<th>PACR1</th>
<th>PACR0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Submode:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>H2 Control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>submode 0</td>
<td>0xx</td>
<td>Edge-sensitive input</td>
<td></td>
<td>H2 Interrupt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>submode 1</td>
<td>100</td>
<td>output- negated</td>
<td>0</td>
<td>0 Disabled</td>
<td>0X</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>submode 1x</td>
<td>101</td>
<td>output - asserted</td>
<td>1</td>
<td>1 Enabled</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>110</td>
<td>output - interlocked</td>
<td></td>
<td></td>
<td>XX</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>111</td>
<td>Output - pulsed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Handshake</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

PACR = %00000000  
PADDR = %00000000

**Means:**

Port A is used as an input port  
Submode 0 (Double Buffered input)  
H2 Edge-sensitive  
H2 interrupt disabled  
H1 interrupt disabled
# Format of Port B Control Register in Mode 0

<table>
<thead>
<tr>
<th>PBCR7</th>
<th>PBCR6</th>
<th>PBCR5</th>
<th>PBCR4</th>
<th>PBCR3</th>
<th>PBCR2</th>
<th>PBCR1</th>
<th>PBCR0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submode:</td>
<td>H4 Control</td>
<td>H4 Interrupt</td>
<td>H3 Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00 submode 0</td>
<td>0xx Edge-sensitive input</td>
<td>0 Disabled</td>
<td>0X H3 interrupt disabled</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01 submode 1</td>
<td>100 output- negated</td>
<td>1 Enabled</td>
<td>10 H3 interrupt enabled</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 submode 1x</td>
<td>101 output - asserted</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>110 output - interlocked</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Handshake</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>111 Output - pulsed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>handshake</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

```
PBCR = %00000000
PBDDDR = %11111111
```

**Means:**

- Port B is used as an output port
- Submode 1 (Double Buffered output)
- H4 Edge-sensitive
- H4 interrupt disabled
- H3 interrupt disabled
Port Status Register, PSR

- Reflects activity of the handshake lines

<table>
<thead>
<tr>
<th>PSR7</th>
<th>PSR6</th>
<th>PSR5</th>
<th>PSR4</th>
<th>PSR3</th>
<th>PSR2</th>
<th>PSR1</th>
<th>PSR0</th>
</tr>
</thead>
<tbody>
<tr>
<td>H4</td>
<td>H3</td>
<td>H2</td>
<td>H1</td>
<td>H4S</td>
<td>H3S</td>
<td>H2S</td>
<td>H1S</td>
</tr>
</tbody>
</table>

Reflects activity of the handshake lines. The PSR register reflects the activity of the handshake lines. The levels of the lines (H4, H3, H2, H1, H4S, H3S, H2S, H1S) are set by the line depending on the mode. PSR0-PSR3 must be cleared by the program by writing a 1 onto them.

Example:
- MOVE.B #$0F,PSR clears bits PSR0-PSR3
- BTST.B #0,PSR checks if status of H1
PI/T Handshaking Input Example

- Continuously check the input handshaking line H1 of port A, if active a new data byte is read from port A and stored in a buffer in memory.

```assembly
ORG $1000
MOVE.B #$30,PGCR  ; Initialize PGCR to enable handshaking
MOVE.B #$80,PACR  ; Initialize port A, submode 1x
MOVE.B #$00,PADDR ; Set Port A as input
MOVE.B #$0F,PSR   ; Clear PSR’s four low status bits.
LEA BUFFER,A0     ; Load DATA address in A0
WAIT
MOVE.B PSR,D0     ; Copy PSR into D0
BTST.B #0,D0      ; Check if bit 0 of PSR = 1
BEQ WAIT
MOVE.B PADR,D1    ; Get a byte from port A
MOVE.B D1,(A0)+   ; Store byte in memory buffer
MOVE.B #$0F,PSR   ; Clear PSR’s four low status bits.
BRA WAIT          ; Busy-wait on H1 for more values
STOP #$2700
ORG $2000
BUFFER DS.B 1000  ; Reserve 1000 bytes for buffer
END $1000
```