Required reading

• P. Chu, *FPGA Prototyping by VHDL Examples*

  *Chapter 16, PicoBlaze I/O Interface*

  *Chapter 17, PicoBlaze Interrupt Interface*
<table>
<thead>
<tr>
<th>Syntax</th>
<th>Example</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>sX</td>
<td>s7</td>
<td>Value at register 7</td>
</tr>
<tr>
<td>KK</td>
<td>ab</td>
<td>Value ab (in hex)</td>
</tr>
<tr>
<td>PORT(KK)</td>
<td>PORT(2)</td>
<td>Input value from port 2</td>
</tr>
<tr>
<td>PORT((sX))</td>
<td>PORT((s1))</td>
<td>Input value from the port specified by register s1</td>
</tr>
<tr>
<td>RAM(KK)</td>
<td>RAM(4)</td>
<td>Value from the RAM location 4</td>
</tr>
</tbody>
</table>
Addressing modes

Immediate mode

SUB s7, [07]
ADD CY s2, [08]

Direct mode

ADD sa, sf
INPUT s5, [2a]

Indirect mode

STORE s3, (sa)
INPUT s9, (s2)
Output Decoding of Four Output Registers
Output Instructions

OUTPUT sX, KK
PORT(KK) <= sX

OUTPUT sX, (sY)
PORT((sY)) <= sX
Timing Diagram of an Output Instruction

- clk
- instruction
- port_id
- out_port
- write_strobe
- en_d(0)
- en_d(1)
- en_d(2)
- en_d(3)
- out_data2

from PicoBlaze

decoded signals

content of s0 sampled and stored

content of s0
## Truth Table of a Decoding Circuit

<table>
<thead>
<tr>
<th>write_strobe</th>
<th>input</th>
<th>output</th>
<th>en_d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>–</td>
<td>–</td>
<td>0000</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0001</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0010</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0100</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1000</td>
</tr>
</tbody>
</table>
## Input Instructions

<table>
<thead>
<tr>
<th>INPUT</th>
<th>sX, KK</th>
<th>sX &lt;= PORT(KK)</th>
<th>DIR</th>
<th>C Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>sX, (sY)</td>
<td>sX &lt;= PORT((sY))</td>
<td>IND</td>
<td>C Z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INPUT</th>
<th>sX, KK</th>
<th>sX &lt;= PORT(KK)</th>
<th>DIR</th>
<th>C Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>sX, (sY)</td>
<td>sX &lt;= PORT((sY))</td>
<td>IND</td>
<td>C Z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INPUT</th>
<th>sX, KK</th>
<th>sX &lt;= PORT(KK)</th>
<th>DIR</th>
<th>C Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>sX, (sY)</td>
<td>sX &lt;= PORT((sY))</td>
<td>IND</td>
<td>C Z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INPUT</th>
<th>sX, KK</th>
<th>sX &lt;= PORT(KK)</th>
<th>DIR</th>
<th>C Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>sX, (sY)</td>
<td>sX &lt;= PORT((sY))</td>
<td>IND</td>
<td>C Z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INPUT</th>
<th>sX, KK</th>
<th>sX &lt;= PORT(KK)</th>
<th>DIR</th>
<th>C Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>sX, (sY)</td>
<td>sX &lt;= PORT((sY))</td>
<td>IND</td>
<td>C Z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INPUT</th>
<th>sX, KK</th>
<th>sX &lt;= PORT(KK)</th>
<th>DIR</th>
<th>C Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>sX, (sY)</td>
<td>sX &lt;= PORT((sY))</td>
<td>IND</td>
<td>C Z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INPUT</th>
<th>sX, KK</th>
<th>sX &lt;= PORT(KK)</th>
<th>DIR</th>
<th>C Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>sX, (sY)</td>
<td>sX &lt;= PORT((sY))</td>
<td>IND</td>
<td>C Z</td>
</tr>
</tbody>
</table>
Block Diagram of Four Continuous-Access Ports
Timing Diagram of an Input Instruction

- clk
- instruction
- port_id
- in_port
- read_strobe
- register s0

input s0, 02
02
sampled data

data is sampled
Block Diagram of Four Single-Access Ports
FIFO Interface

clk  rst
clk  rst

clk  rst

FIFO

8
din  dout

8
full  empty

write  read
Operation of the First-Word Fall-Through FIFO

<table>
<thead>
<tr>
<th>clk</th>
<th>write</th>
<th>read</th>
<th>din</th>
<th>dout</th>
<th>empty</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D</td>
<td>D</td>
<td></td>
</tr>
</tbody>
</table>

FIFO operation: write, write, read, write, read, write, read

FIFO data: A, B, A, B, C, D

1 2 3 4 5 6 7 8
Operation of the “Standard” FIFO

 clk
write
read
din
A B C D
C D
empty
FIFO operation write write read write read write read read
FIFO data A B A B C D A B C D

ECE 448 – FPGA and ASIC Design with VHDL
Interrupt Flow

```vhdl
; ====== main loop ======
forever:
    ...
    enable interrupt
    ...
    add s0, s3
    sub s5, 01
    ...
    call critical_timing
    ...
    jump forever

;=== time critical segment ===
critical_timing:
    disable interrupt
    ...
    enable interrupt
    return

;=== interrupt service routine ===
isr:
    test s2, 01
    ...
    return enable

;=== interrupt vector ===
address 3FF
jump isr
```
Timing Diagram of an Interrupt Event

The instruction is preempted and "call 3FF" is implicitly executed
Interrupt Related Instructions

RETURNI ENABLE
  PC <= STACK[TOS] ; TOS <= TOS – 1;
  I <= 1;  C<= PRESERVED C;  Z<= PRESERVED Z

RETURNI DISABLE
  PC <= STACK[TOS] ; TOS <= TOS – 1;
  I <= 0;  C<= PRESERVED C;  Z<= PRESERVED Z

ENABLE INTERRUPT
  I <=1;

DISABLE INTERRUPT
  I <=0;
Interrupt Interface with a Single Event

Diagram:

- int request
- set flag
- clr
- flag FF
- KCPSM3
- in_port
- reset
- instruction
- interrupt
- out_port
- port_id
- read_strobe
- write_strobe
- interrupt_ack
- address
Interrupt Interface with Two Requests

![Diagram showing the interrupt interface with two requests.]
Time-Multiplexed Seven Segment Display
Block Diagram of the Hexadecimal Time-Multiplexing Circuit
Hexadecimal Multiplexing Circuit Based on PicoBlaze and mod-500 Counter