ELEC 4200 Lab#9 Interfacing External Devices to a Processor Core

AUBURN
UNIVERSITY

SAMUEL GINN
COLLEGE OF ENGINEERING

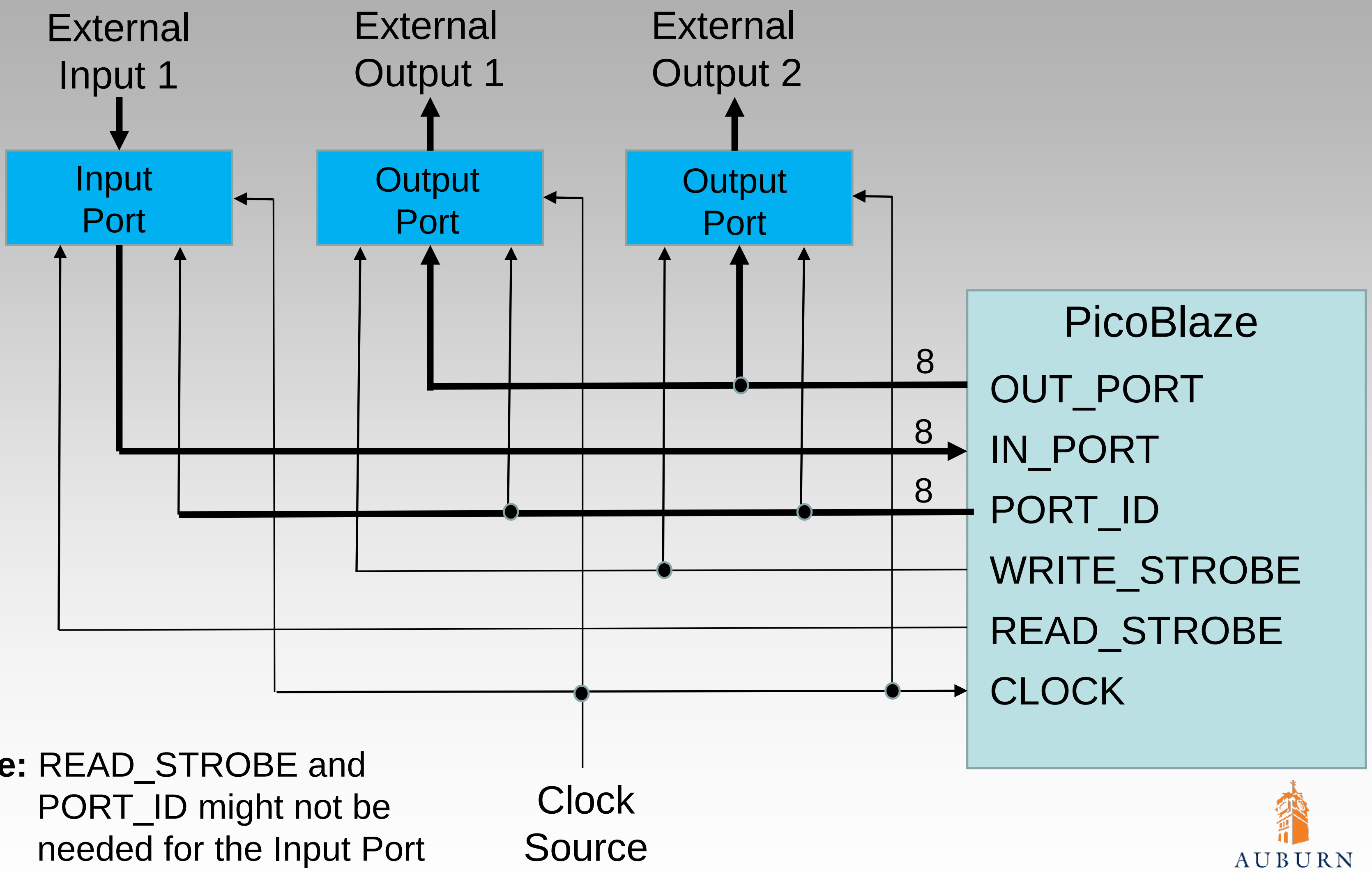
References (on lab web page) you may need:

- *PicoBlaze KCPSM6 User Manual*
- *PicoBlaze 8-bit Embedded Microcontroller User Guide*

Overview

- Design a simple output port and a simple input port, to be interfaced to *PicoBlaze*
  - Requires understanding of the interface (data, addressing and control/timing) between *PicoBlaze* and input/output ports
- Add input/output ports to top-level *PicoBlaze* system.
  - "System" = *PicoBlaze* CPU + instruction memory + two output ports + one input port
- Write a *PicoBlaze* assembly language program to use the input/output ports to read and test external inputs (button/switches), and write to external outputs (LEDs)

PicoBlaze Input/Output Ports

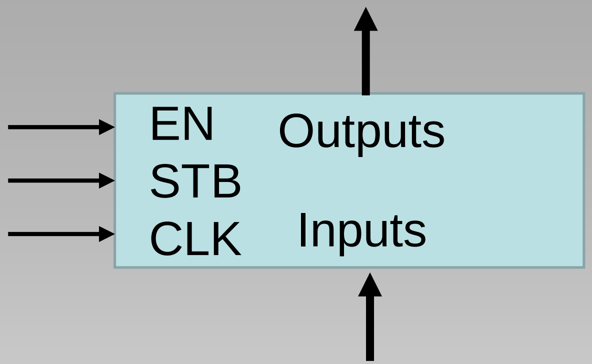


Note: READ_STROBE and PORT_ID might not be needed for the Input Port

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Output Port Design

VHDL Model Specifications



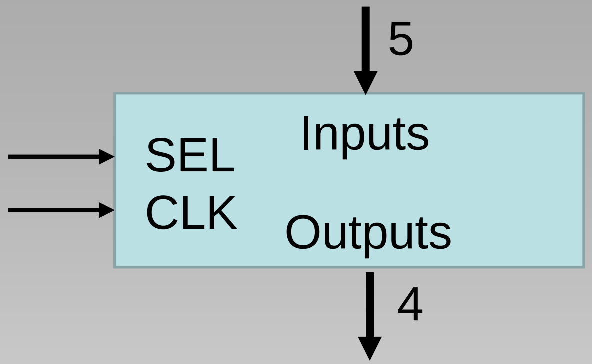
- Inputs: N bits (N between 1 and 8)
  - Will come from OUT_PORT from PicoBlaze
- Outputs: N bits (same as inputs)
  - Will drive LEDs (in this lab)
- Enable input: Capture input data only if Enable = ‘1’
  - Will come from PORT_ID
  - “One-hot” PORT_ID values allow up to eight output ports to each be selected by one bit of the PORT_ID.
- Strobe input: Capture input data only if Strobe = ‘1’
  - Will come from WRITE_STROBE
- Clock: Capture input data on ‘0’ -> ‘1’ transition of the clock
  - Will come from system clock

Instantiate two of these in the system, at port ID’s 02 and 04

- Each to drive four LEDs

References: Input/output instructions and port design information in the two documents listed on the title slide of this file.

Input Port Design



VHDL Model Specifications

- Inputs: 5 bits
  - In0 (bits 3-0) from 4 slide switches
  - In1 (bit 4) from a push button
- Outputs: 4 bits
  - Data to PicoBlaze via IN_PORT(3:0)
- Internal register: 4 bits
  - Capture selected input data
- SEL: select inputs for loading into the internal register
  - SEL=0 selects In0 (4 switches)
  - SEL=1 selects In1 (button on bit 0, i.e. output bits 3-0 = 0-0-0-button)
- CLK: capture data in the internal register on rising edge of clock

Instantiate in the system, at PORT_IDs 00 and 01

- In0 to be read for PORT_ID = 00
- In1 to be read for PORT_ID = 01

System Operation

- The top-level system is to contain five component instantiations:
  - PicoBlaze CPU
  - Instruction memory (with test program)
  - Two 4-bit output ports, each driving four LEDs
  - One input port (from 4 slide switches and a push button)
- The PicoBlaze assembly language program should execute the following set of operations continuously.
  1. Loop until the push button is pressed
  2. Send the binary number on the four slide switches to the LEDs.
  3. Execute a "delay loop" until "bouncing" finished (experiment with the delay)
  4. Loop until the push button is released
  5. Return to step 1.
  - First time through the loop: send the number on the slide switches to the first set of LEDs.
  - Second time through the loop, send the number on the slide switches to the second set of LEDs.
  - Alternate between these two patterns.

Pre-lab Assignment

- Review the PicoBlaze *KCPSM6 User Manual* pages on:
  - INPUT and OUTPUT instructions
  - READ and WRITE strobes
  - INPUT and OUTPUT port design
- Write VHDL models for an input port and an output port
  - Be prepared to verify these designs via simulation
- Write VHDL statements that declare the input and output port components and instantiate two output ports and one input port.
  - Modify the top-level model, connecting PicoBlaze, program memory, and the input and output ports
- Write a PicoBlaze assembly language program, *InputOutput.psm*, to implement the algorithm described on the previous slide
  - Be prepared to assemble this program and generate the PicoBlaze instruction memory
  - Verify input/output port operation via Active-HDL simulation before implementation

Lab Exercise

- Simulate your input and output port VHDL models to verify their correctness.
- Assemble your PicoBlaze program and include the generated instruction memory in your previous PicoBlaze system.
- Add the input and output ports to your PicoBlaze system.
  - The top-level entity must have outputs to 8 LEDs and inputs from 4 switches and a push button.
- Use *Active-HDL* to simulate and debug your PicoBlaze system and program
- Implement and download your Picoblaze system onto the NEXYS 4 DDR board
  - Demonstrate synthesized circuitry to the GTA

Tips and Tricks

- Consider how you will verify that input data from switches and buttons are being correctly accessed via your input port.
    - Can you properly select button and slide switches?
- Consider how you will verify that output data is being sent to the LEDs.
    - At two different output ports

Report Guidelines

Be sure to include all sections required by the lab manual guidelines. In addition be sure your report includes the following:

- Input and output port VHDL models
- Top-level VHDL model
- PicoBlaze assembly code
- Description of your assembly program and how it works
- Steps taken to simulate your VHDL model and program
- Steps taken to implement and test your design on the NEXYS 4 DDR board
- Synthesis results (LUTs, FFs, slices, etc)
- Experimental results - what went right and wrong in your design and program