# CSCI 355 Digital Logic and Computer Organization

# Laboratory Manual

Dr. Peter Walsh

Department of Computing Science

Malaspina University College

Nanaimo, B.C., V9R 5S5, Canada

Version 0.5 Fall 2005

# 1. Preamble

CSCI 355 is concerned with the analysis and design of digital circuits. Two methods of design verification are employed. First, a software model of the design is developed and tested using the hardware description language VHDL. Then, the design is physically realized and tested in the laboratory.

Design modeling using VHDL is discussed in the lecture portion of this course. Models are developed for simulation and for synthesis. Designs are realized either on a breadboard using standard ICs (integrated circuits) or on a XSA-50 FPGA based prototyping board.

Breadboarding and the XSA-50 board are overviewed in sections 2 and 3 respectively. Section 4 discusses experiment deliverables. The remainder of the manual contains a series of laboratory tasks (experiments) and data sheet appendices.

# 2. Breadboarding

During breadboarding, you will plug the necessary ICs into the breadboard, cut wires of proper length, and connect the ICs on the breadboard using the wires according to your design specification (schematic). At the end of this section, a tutorial from Hobby Electronics introduces basic breadboarding (`http://www.hobby-electronics.com/MiniTutorialIndex.htm`). The following descriptions assume you have read and understood the tutorial.

**IDL-800 Digital Circuit Evaluator**

Our laboratory is equipped with IDL-800 digital circuit evaluators. The IDL-800 has 19 different basic components. Initially, you will be concerned with the following 6 components:

1 power switch with indicator

2 pulse switches

3 level switches

4 removable solderless breadboard

5 fixed DC +5volts and ground

6 leds (lamps)


*IDL-800 Digital Circuit Evaluator*

## Wiring Technique

It is important to develop good wiring technique when realizing your designs on a breadboard. Good wiring technique will help both the student and the instructor test and debug your circuit realization. The following convention must be adhered to:

- Power to the IDL-800 should be off before you insert or remove wires or ICs from the unit.
- Insert your required ICs into the breadboard gap as described in the tutorial. Note that ICs have a notch that is used as an alignment reference. ICs have legs or pins that are the input/output interface to the circuits on the chip. The legs bend easily, you may have to align them (straighten them) before inserting the IC in the breadboard gap. Make sure adjacent IC legs are inserted into adjacent breadboard pin-holes. To achieve correct IC orientation, the two rows of IC legs must straddle the breadboard gap and the IC notch must be on the left hand side of the IC.
- To remove an IC from the breadboard, slide an IC chip extractor under the body of the chip and lift gently (a pencil will work if you dont have an extractor to hand).
- Cut wires to the appropriate length. Route wires around, not over ICs. Whenever possible, make manhatten (90 degree) wiring turns along unoccupied sections of the breadboard. Colour code your wires. Use red for +5volts, black for ground, green for data signals and blue for control signals. Use the rails to supply +5 and ground signals to the ICs.

The following wiring sequence is strongly recommended:

- wire the +5volts and ground rails
- wire the +5volts and ground signals (from the rails to the ICs)
- wire data signals
- wire control signals

One final note, on occasion, you may end up using an IC that is faulty. Consequently, it is strongly recommended that you verify the operation of each IC in isolation BEFORE you wire-up your circuit.

hobby ELECTRONICS

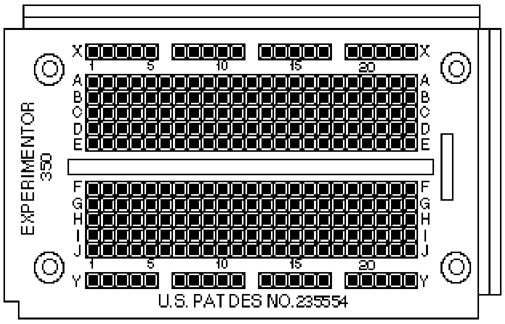
# *Breadboarding Basics*

**Home**

**Mini-Tutorial Index**

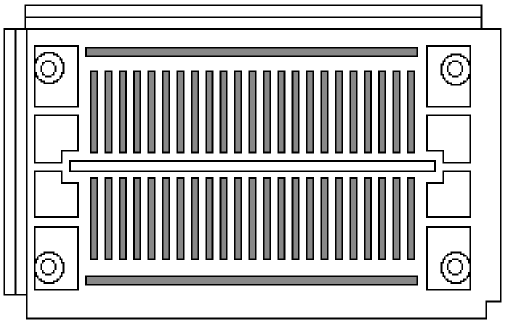
The best method of experimenting with ICs is to use a "breadboard" to build circuits. Breadboards, more formally known as *solderless modular sockets*, get their name from the early days of radio, when it was common to build vacuum tube circuit prototypes on a wooden breadboard.

Today's breadboards are a grid of insulating plastic atop a pattern of conducting metal strips. Here is a top view of a typical breadboard:




Component leads and wires are inserted into the holes and make contact with the conducting metal strips underneath, thus "connecting" them together.

The pattern of conducting strips underneath the insulating plastic is shown below. Notice there are two horizontal strips along both sides of the breadboard and a series of shorter vertical strips. The two horizontal strips are normally used for power supply connections, with one strip being the supply voltage and the other being the ground connection. (Breadboards with four horizontal strips are available, and are used for circuits requiring dual polarity or multiple voltage power supplies.) The horizontal strips are often known as *rails*.



Note the gap separating the vertical strips. The dual in-line package (DIP) IC is normally placed across this gap. One row of pins is one side of the gap, and the other row of pins is on the opposite side.

Breadboards come in a variety of sizes, and are usually measured in terms of the number of connection or "tie points" provided. Some breadboards come with binding posts for connecting a power supply; deluxe models have power supplies built in and with additional supports for potentiometers, LEDs, and meters.

# 3. XSA-50 Prototyping Board

The Xess XSA-50 prototyping board contains a Xilinx Spartin II FPGA with 50,000 gates (the board's manual is linked off the CSCI 355 course page). The FPGA can be programmed in-the-lab to execute a circuit synthesized from a VHDL model. The FPGA programming process is similar to downloading an S19 record file to a HC11 development board. The "object" code file is called a bit-file and has a `.bit` file-name extension. The XSTOOLS form Xess interface the development environment with the board through a connecting parallel port. The tools allow for board testing (`xstest`); bit-file downloading (`xsload`) and I/O (`xsport`). With regard to I/O, certain FPGA input-pins and 7-segment LED output-pins are mapped to pins in the parallel port. The I/O mappings are given below (see Appendix A for LED segment labels):

| port bit | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| pin | 58 | 51 | 65 | 47 | 42 | 48 | 50 |

| LED | a | b | c | d | e | f | g | dp |
|---|---|---|---|---|---|---|---|---|
| pin | 49 | 46 | 39 | 67 | 62 | 57 | 60 | 44 |

Note, the most significant bit of the parallel port is also mapped but it is recommended that it always be set. The following annotated command sequence demonstrates how to download and exercise a two-input one-output logical AND gate. The inputs are mapped to port-pins 0 and 1 and, the output is mapped to LED segment *c*.

```
xstest -b xsa-50       -- test the xsa-50 board

xsload -fpga and2.bit  -- download and2.bit to the board

xsport 10000011        -- port-bit 0 and port-bit 1 are asserted (assume positive logic)
```

Products Support Purchasing XESS Misc. search

**XSA-50**

Price: $59.00
In Stock: Yes

Features:

- XC2S50 FPGA
- XC9572XL CPLD
- 8 MByte SDRAM
- 128 KByte Flash
- 100 MHz programmable oscillator
- Parallel port
- Keyboard/mouse PS/2 port
- 64-color VGA port
- 7-segment LED
- 1 pushbutton
- 4 DIP switches
- 84-pin prototyping interface (53 I/O pins)
- 5V-tolerant I/O
- 9V DC power jack
- 5V / 3.3V / 2.5V regulators
- Downloading cable
- XSTOOLs utilities CDROM
- Works with the XST-3.0 Board
- Works with XILINX ISE, WebPACK, iMPACT and ChipScope software

Documentation:

- XSA-50 Board Manual (PDF file)
- XSA-50 Board Pin List (Excel file)

Accessories:

- XStend Board 3.0
- 9 VDC power supply (North America)
- 9 VDC power supply (Europe)

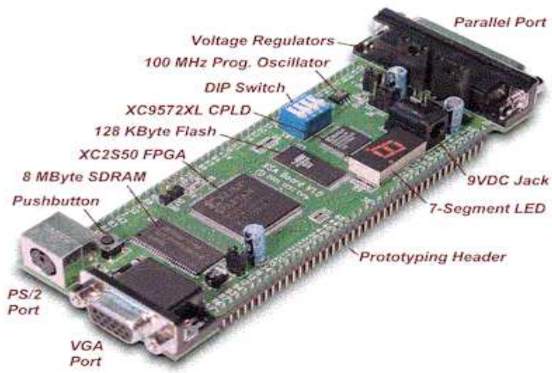
The XSA-50 Board keeps the same form-factor as our popular XS40 Boards while increasing the logic density to 50,000 gates with a Spartan-2 XC2S50 FPGA. The FPGA is combined with a 8 MByte synchronous DRAM and 128 KByte Flash to give you the resources for building a complete, soft-core RISC microcontroller system! Or anything else you might think of...

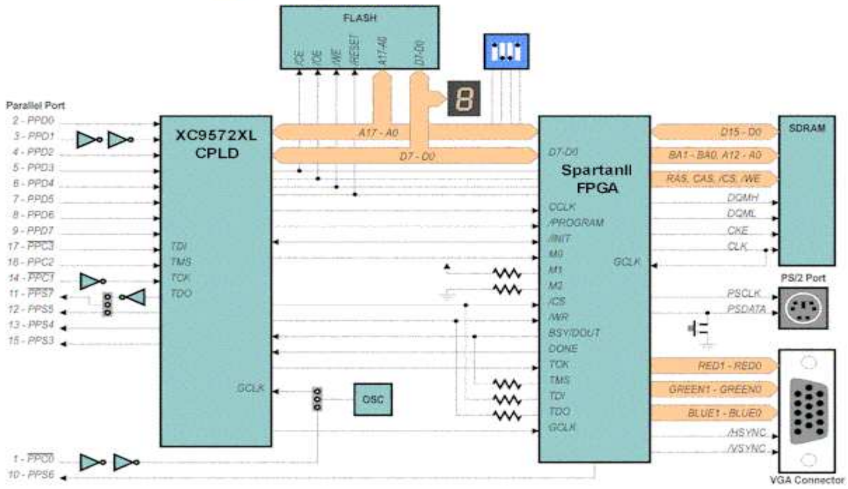
The bitstream for the XSA-50 can be stored in the on-board 128 KByte Flash so the FPGA loads its configuration as soon as power is applied. Or you can download directly to the board through the parallel port with the XSTOOLs utilities we provide. The interface CPLD on the XSA-50 also supports downloading with XILINX iMPACT and circuit test/debug with ChipScope software using our simple downloading cable. No more expensive XILINX cables!

In addition to the large FPGA, SDRAM and Flash chips, you also get a VGA port that produces vivid graphics in 64 colors. And the prototyping header gives you 53 general-purpose I/O pins for building interfaces to external devices.

And the XSA-50 helps you maintain your investment. Use your existing XStend Boards to add a stereo codec, LEDs, switches, and a dedicated prototyping area to the XSA-50. And keep your power supplies - the XSA-50 has the same power connections as an XS40 Board.

Think it will be a big switch to move from the XS40 Boards to the XSA-50? Well, we provide all the software utilities for programming the FPGA, setting the oscillator frequency, and downloading and uploading the RAM and Flash. And we make all the source code available for you to play with! How about design examples? We have parameterized modules for interfacing to the PS/2 keyboard port, displaying images through the VGA port, and reading/writing to the synchronous DRAM as if it were a simple static RAM. And we will be adapting all the chapters of our Pragmatic Logic Design online text to support the XSA-50.




webmaster@xess.com


Products Manuals Downloads How to Buy Online Quote Order Online
Help! FAQ Forum Tutorials Examples Links Home About Us Contact Us

# 4. Experiment Deliverables and Evaluation

The remainder of the manual contains a series of laboratory tasks (experiments). Each experiment has a pre-lab component and an in-lab component. In the pre-lab component, you develop solution(s) to the laboratory task(s) and verify your design(s) by building and testing VHDL model(s). All files relating to laboratory work can be found in `~csci355` on `cscicad`. For Lab$x$, the design file and the design testbench can be found in `~csci355/Lab`$x$. Models for `SN74??` TTL ICs can be found in directories of the same name.

In the in-lab component, your task is to realize your circuit design(s), verify them and record any relevant observations you made during the in-lab period.

Pre-lab VHDL code is due before 12:00 noon on the day BEFORE your scheduled in-lab time. If your model(s) fail any of the supplied test cases (in the testbench(s)) then you will not be allowed to proceed to the in-lab component. You will receive no credit for the experiment.

You must complete a experiment report for each laboratory task. The report is due at the beginning of your in-lab time; all report deliverables must be in place with the exception of deliverables that are due by the end of your in-lab time. Your report will be initialed by the instructor and returned to you for the duration of the in-lab time. Once you have constructed and verified the behaviour of your circuit, sign your report to indicate your circuit behaves as specified. Then, ask your instructor to check your work. If either your design or implementation if found to be flawed, you will receive no credit for the physical realization component of the experiment. Submit the complete report to your instructor at the end of the in-lab time.

Each Lab is graded out of 100 points. The breakdown is as follows:

- **10 points:** report presentation

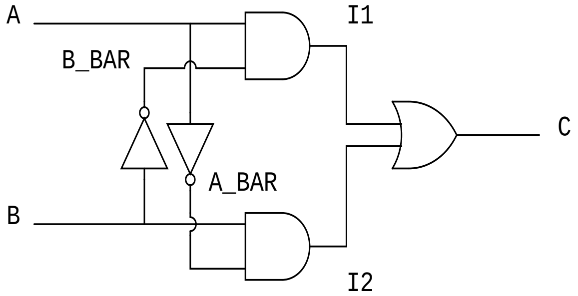
- **45 points:** report deliverables
- **45 points:** physical circuit behaviour

A sample report follows the statement of Lab0. All pages of your report must be stapled together. All pages of your report must have a page number on the top right hand corner in the form of `Page n of m`. You need not word-process your reports but hand drawn figures and writing must be clear and legible.

The exact wiring connections for your circuit must be detailed in pin-out sheets. You must complete at least one pin-out sheet for each IC employed in your physical circuit. For a given IC, each row in its pin-out sheet details a wire connecting one of its pins to another pin on the same IC or to a pin on a different IC. Consider wire `X`. Let the the source and destination of wire `X` be called `A` and `B`. When the pin-out is done correctly, you will find that `X`'s connection details are listed at least twice; first detailing the connection from `A` to `B` and then detailing the connection from `B` to `A`. At the end of a pin-out sheet, you may repeat *significant* connections such as connections to switches and leds. You must use the pin-out sheet template for all pin-outs.

# Experiment 0

With reference to the combinational system specified in the following logic schematic:




**Task:** Perform symbolic analysis.

**Deliverable (D1):** Network expression specifying `C` in terms of `A` and `B`.

**Task:** Perform literal analysis.

**Deliverable (D2):** Truth table with one output column for each of `A_BAR`, `B_BAR`, `I1`, `I2` and `C`.

**Task:** Develop a sinthesizable structural VHDL model of the combinational system using the VHDL models for the TTL ICs SN7404, SN7408 and SN7432.

**Deliverable (D3):** Electronic submission of source code (`make submit`).

**Task:** Map your VHDL model to a TTL-based physical design for the combinational system.

**Deliverable (D4):** IC logic schematic.

**Task:** Specify IC interconnections.

**Deliverable (D5):** One completed pin-out sheet (at least) for each IC employed in your physical design.

**Task:** In the laboratory, wire-up your physical design, verify its behaviour and sign-off on the design/implementation.

**Deliverable (D6):** A physical bread-board realization of the combinational system that behaves to specification. Details of the circuit-verification process. Student signature indicating that the circuit behaves as specified.

**Task:** In the laboratory, download your physical designs to the xsa-50 FPGA board, verify its behaviour and sign-off on the design/implementation.

**Deliverable (D7):** A physical FPGA realization of the combinational system that behaves to specification. A hardcopy of `lab0.ucf` (file detailing FPGA area contaraints and I/O pin assignments). Details of the circuit-verification process. Student signature indicating that the circuit behaves as specified.

**Task:** Document any relevant results, explanations or comments.

**Deliverable (D8):** A section in your report entitled Results/Explanations/Comments in which you have detailed any relevant results, explanations or comments.

# NOTES

*Experiment 0 Report: Page 1 of 6*

# CSCI 355

## Experiment 0 Report

### Peter Walsh (Lab Section F0001)

**Deliverable D1**

$$C = \mathtt{A}\overline{\mathtt{B}} + \overline{\mathtt{A}}\mathtt{B}$$

**Deliverable D2**

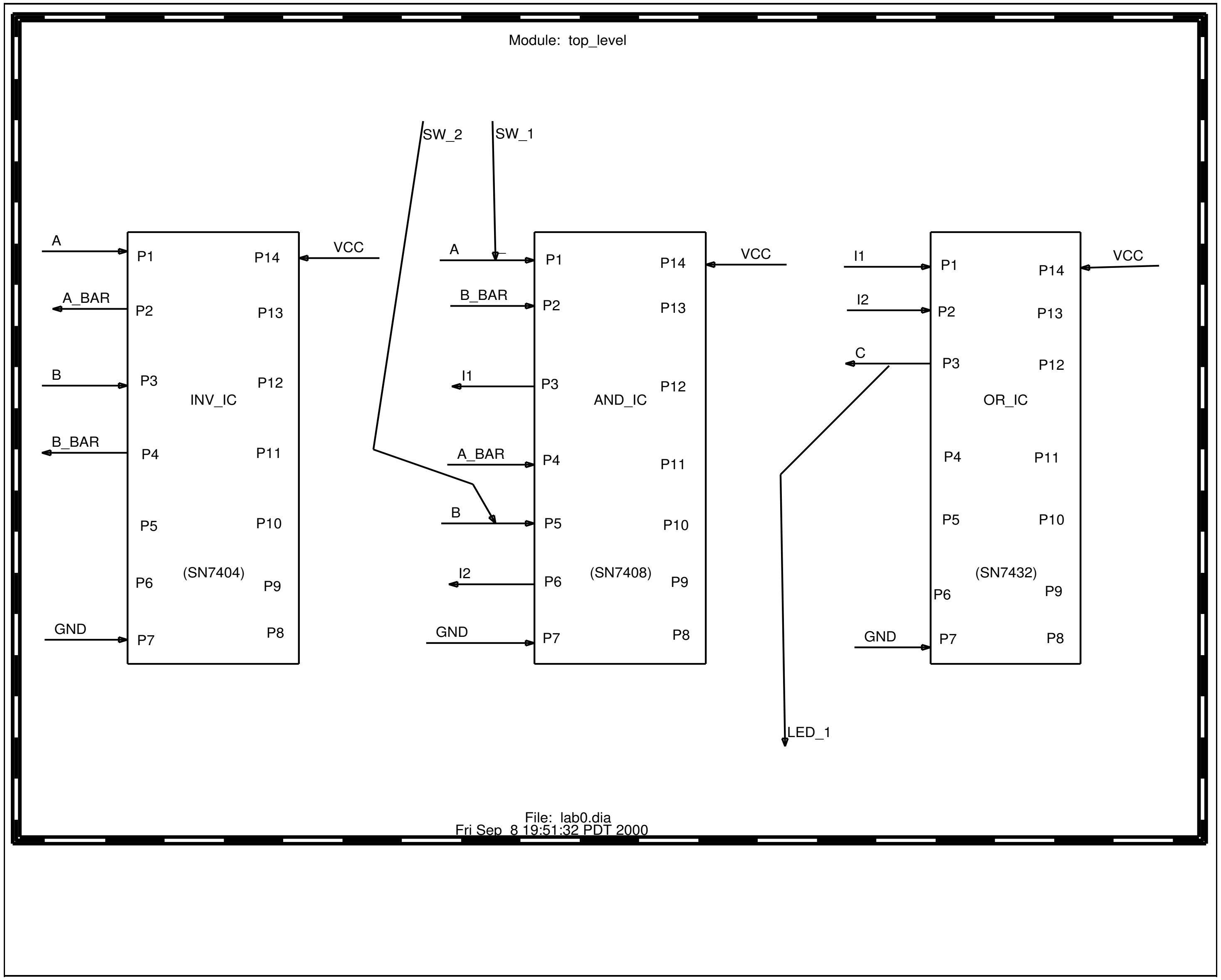
| A | B | A_BAR | B_BAR | I1 | I2 | C |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |

**Deliverable D3**

Submitted electronically through `make submit`.

Experiment 0 Report: Page 2 of 6

## Deliverable D4



## Deliverable D5

# CSCI 355 Laboratory Pin-Out Sheet

**STUDENT NAME** Peter W  Page__3____Of__6____

**EXPERIMENT NUMBER** 0

**IC NUMBER** SN7432

**IC INSTANCE NAME** OR_IC

| *Source* | | *Destination* | |
|---|---|---|---|
| Pin Number | VHDL Alias Name (optional) | Pin Number | VHDL Alias Name (optional) |
| | | | |
| P1 | A1 | AND_IC.P3 | Y1 |
| P2 | B1 | AND_IC.P6 | Y2 |
| P3 | Y1 | LED_1 | |
| P7 | GND | +0V | |
| P14 | VCC | +5V | |
| | | | |
| LED_1 | | OR_IC.P3 | Y1 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# CSCI 355 Laboratory Pin-Out Sheet

**STUDENT NAME** Peter W Page__4____Of__6____

**EXPERIMENT NUMBER** 0

**IC NUMBER** SN7408

**IC INSTANCE NAME** AND_IC

| *Source* | | *Destination* | |
|---|---|---|---|
| Pin Number | VHDL Alias Name (optional) | Pin Number | VHDL Alias Name (optional) |
| | | | |
| P1 | A1 | INV_IC.P1 | A1 |
| | | SW_1 | |
| P2 | B1 | INV_IC.P4 | Y2 |
| P3 | Y1 | OR_IC.P1 | A1 |
| P4 | A2 | INV_IC.P2 | Y1 |
| P5 | B2 | INV_IC.P3 | A2 |
| | | SW_2 | |
| P6 | Y2 | OR_IC.P2 | B1 |
| P7 | GND | +0V | |
| P14 | VCC | +5V | |
| | | | |
| SW_1 | | AND_IC.P1 | A1 |
| SW_2 | | AND_IC.P5 | B2 |

# CSCI 355 Laboratory Pin-Out Sheet

**STUDENT NAME** Peter W  Page__5____Of__6____

**EXPERIMENT NUMBER** 0

**IC NUMBER** SN7404

**IC INSTANCE NAME** INV_IC

| *Source* | | *Destination* | |
|---|---|---|---|
| Pin Number | VHDL Alias Name (optional) | Pin Number | VHDL Alias Name (optional) |
| | | | |
| P1 | A1 | AND_IC.P1 | A1 |
| P2 | Y1 | AND_IC.P4 | A2 |
| P3 | A2 | AND_IC.P5 | B2 |
| P4 | Y2 | AND_IC.P2 | B1 |
| P7 | GND | +0V | |
| P14 | VCC | +5V | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

*Experiment 0 Report: Page 6 of 6*

**Deliverable D6**

Exhaustive testing (all 4 input test patterns) was employed. The physical system (bread-board) behaved as specified. `Signature:` *Peter W*

**Deliverable D7**

```
#PACE: Start of Constraints generated by PACE

#PACE: Start of PACE I/O Pin Assignments

NET "A"  LOC = "P50"  ;

NET "B"  LOC = "P48"  ;

NET "C"  LOC = "P44"  ;

#PACE: Start of PACE Area Constraints

#PACE: Start of PACE Prohibit Constraints

#PACE: End of Constraints generated by PACE
```

Exhaustive testing (all 4 input test patterns) was employed. The physical system (FPGA) behaved as specified. `Signature:` *Peter W*

**Deliverable D8**

Results/Explanations/Comments:

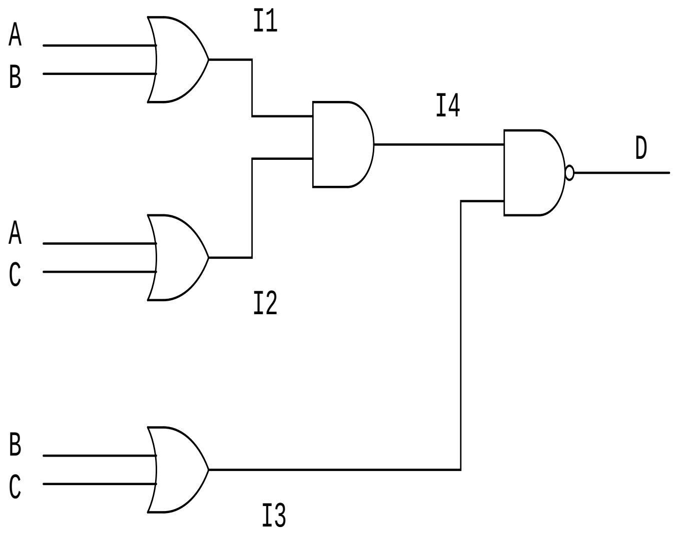
Nothing further to report.

<<<<<<<<<<<<< End of Report >>>>>>>>>>>>>

Photographs of physical realizations of the circuit from Experiment 0 can be accessed from the csci 355 course web page.

# Experiment 1

With reference to the combinational system specified in the following logic schematic:




**Task:** Perform symbolic analysis.

**Deliverable (D1):** Network expression specifying `D` in terms of `A`, `B` and `C`.

**Task:** Perform literal analysis.

**Deliverable (D2):** Truth table with one output column for each of `I1, I2, I3, I4` and `D`.

**Task:** Develop a structural VHDL model of the combinational system using the VHDL models for the TTL ICs SN7432, SN7408 and SN7400.

**Deliverable (D3):** Electronic submission of source code (`make submit`).

**Task:** Map your VHDL model to a TTL-based physical design for the combinational system.

**Deliverable (D4):** IC logic schematic.

**Task:** Specify IC interconnections.

**Deliverable (D5):** One completed pin-out sheet (at least) for each IC employed in your physical design.

**Task:** In the laboratory, wire-up your physical design, verify its behaviour and sign-off on the design/implementation.

**Deliverable (D6):** A physical realization of the combinational system that behaves to specification. Details of the circuit-verification process. Student signature indicating that the circuit behaves as specified.

**Task:** In the laboratory, download your physical designs to the xsa-50 FPGA board, verify its behaviour and sign-off on the design/implementation.

**Deliverable (D7):** A physical FPGA realization of the combinational system that behaves to specification. A hardcopy of `lab1.ucf` (file detailing FPGA area contaraints and I/O pin assignments). Details of the circuit-verification process. Student signature indicating that the circuit behaves as specified.

**Task:** Document any relevant results, explanations or comments.

**Deliverable (D8):** A section in your report entitled Results/Explanations/Comments in which you have detailed any relevant results, explanations or comments.

# NOTES

# Experiment 2

With reference to the VHDL description in `~csci355/Lab2/lab2.vhd`:

**Task:** Extract the IC logic schematic from the VHDL description.

**Deliverable (D1):** IC logic schematic.

**Task:** Extract the combinational logic schematic.

**Deliverable (D2):** Logic schematic.

**Task:** Perform symbolic analysis.

**Deliverable (D3):** Network expression specifying the output in terms of its inputs.

**Task:** Perform literal analysis.

**Deliverable (D4):** Truth table with one output column for each of `I1, I2, I3, I4` and `E`.

**Task:** Derive a minimal SOP expression for the modeled circuit.

**Deliverable (D5):** A listing of all the prime-implicants, the essential prime-implicants and the secondary prime-implicants. A minimal SOP expression and a K-Map showing the corresponding implicant cover.

**Task:** Develop a testbench for the modeled circuit You must use a gold function as exemplified in the testing of SN7400. You must use your minimal SOP expression as the gold function.

**Deliverable (D6):** Electronic submission of source code (`make submit`).

**Task:** Specify IC interconnections.

**Deliverable (D7):** One completed pin-out sheet (at least) for each IC employed in the physical design.

**Task:** In the laboratory, wire-up your physical design, verify its behaviour and sign-off on the design/implementation.

**Deliverable (D8):** A physical realization of the combinational system that behaves to specification. Details of the circuit-verification process. Student signature indicating that the circuit behaves as specified.

**Task:** Document any relevant results, explanations or comments.

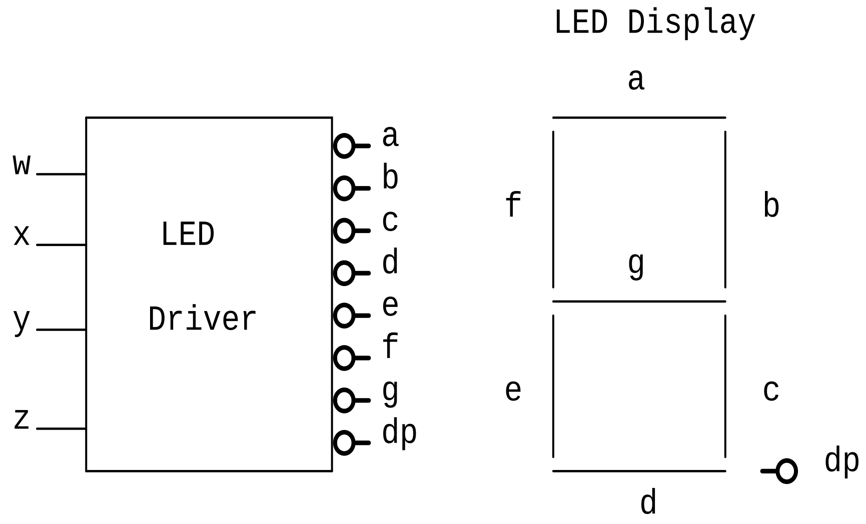
**Deliverable (D9):** A section in your report entitled Results/Explanations/Comments in which you have detailed any relevant results, explanations or comments.

# NOTES

# Experiment 3

A seven segment display is used to display a decimal digit (or a hex digit). Such displays are often found in electronic watches, clocks and appliances. The seven segment display used in this lab consists of seven LED segments and a decimal point (see Appendix A, TIL 729).

Consider the combinational system (LED Driver) and the seven segment display (LED Display) depicted in the following logic schematic:




The LED Driver takes as input, a four bit binary number ( `w,x,y,z` with `z` the LSB) and produces active low output to drive the LED Display. If the input is between `0000` and `1001` inclusive then the LED Display's decimal point `dp` is illuminated.[1] If the input is between `1010` and `1111` inclusive then the LED Display's decimal point `dp` is not illuminated and the LED Display's segments `a` through `g` are illuminated so as to display the hex character (in upper case) that represents the input. For example, if the input is `1111` then the segments `a`, `e`, `f` and `g` are illuminated. All other segments (including the decimal point `dp`) are not illuminated.

**Task:** Perform problem analysis.

[1] illuminated means the associated input signal is asserted (active low)

**Deliverable (D1):** Truth table for the LED Driver with one output column for each of `a, b, c, d, e, f, g` and `dp`.

**Task:** Use espresso to derive a minimized SOP expression for the LED Driver.

**Deliverable (D2):** Hardcopy of espresso output.

**Task:** Perform symbolic analysis.

**Deliverable (D3):** Logic schematic derived from espresso output and a network expression specifying LED driver output in terms of its inputs.

**Task:** Develop a structural VHDL model of the LED Driver using the VHDL models for the TTL ICs SN7432, SN7408, SN7411 and SN7404.

**Deliverable (D4):** Electronic submission of source code (`make submit`).

**Task:** Map your VHDL model to an IC-based physical design using TTL components and one TIL 729 and a resistor pack (see Appendix A).

**Deliverable (D5):** IC logic schematic.

**Task:** Specify IC interconnections.

**Deliverable (D6):** One completed pin-out sheet (at least) for each IC employed in your physical design.

**Task:** In the laboratory, wire-up your physical design, verify its behaviour and sign-off on the design/implementation.

**Deliverable (D7):** A physical realization of the combinational system that behaves to specification. Details of the circuit-verification process. Student signature indicating that the circuit behaves as specified.

**Task:** In the laboratory, download your physical designs to the xsa-50 FPGA board, verify its behaviour and sign-off on the design/implementation.

**Deliverable (D8):** A physical FPGA realization of the combinational system that behaves to specification. A hardcopy of `lab3.ucf` (file detailing FPGA area contaraints and I/O pin assignments). Details of the circuit-verification process. Student signature indicating that the circuit behaves as specified.

**Task:** Document any relevant results, explanations or comments.

**Deliverable (D9):** A section in your report entitled Results/Explanations/Comments in which you have detailed any relevant results, explanations or comments.

# NOTES

# Experiment 4

Consider again the combinational system (LED Driver) and the seven segment display (LED Display) from Experiment 3. The LED Driver takes as input, a four bit binary number (`w,x,y,z` with `z` the LSB) and produces active low output to drive the LED Display. For this experiment, if the input is between `0000` and `1001` inclusive then the LED Display's segments `a` through `g` are illuminated so as to display the decimal character that represents the input.[2] The LED Display's decimal point `dp` is NOT illuminated. See the testbench in `~csci355/Lab4` to determine the exact segment pattern for each decimal digit.

**Task:** Perform problem analysis.

**Deliverable (D1):** Truth table for the LED Driver with one output column for each of `a, b, c, d, e, f, g` and `dp`.

**Task:** Design a ROM to implement the LED Driver. Use a decoder and `nand/or` gate(s) to implement the ROM.

**Deliverable (D2):** Logic schematic for the LED Driver.

**Task:** Develop a structural VHDL model of the LED Driver using the VHDL models for the TTL ICs SN7432, SN7410, SN7413 and SN74154. Note, you can use two instances of SN7410.

**Deliverable (D3):** Electronic submission of source code (`make submit`).

**Task:** Map your VHDL model to an IC-based physical design using TTL components and one TIL 729 and a resistor pack (see Appendix A).

**Deliverable (D4):** IC logic schematic.

**Task:** Specify IC interconnections.

---

[2] illuminated means the associated input signal is asserted (active low)

**Deliverable (D5):** One completed pin-out sheet (at least) for each IC employed in your physical design.

**Task:** In the laboratory, wire-up your physical design, verify its behaviour and sign-off on the design/implementation.

**Deliverable (D6):** A physical realization of the combinational system that behaves to specification. Details of the circuit-verification process. Student signature indicating that the circuit behaves as specified.

**Task:** In the laboratory, download your physical designs to the xsa-50 FPGA board, verify its behaviour and sign-off on the design/implementation.

**Deliverable (D7):** A physical FPGA realization of the combinational system that behaves to specification. A hardcopy of `lab4.ucf` (file detailing FPGA area contaraints and I/O pin assignments). Details of the circuit-verification process. Student signature indicating that the circuit behaves as specified.

**Task:** Document any relevant results, explanations or comments.

**Deliverable (D8):** A section in your report entitled Results/Explanations/Comments in which you have detailed any relevant results, explanations or comments.

# NOTES

# Experiment 5

Develop a 4 bit adder/subtractor by augmenting a 4 bit binary adder and then, augmenting the adder/subtractor to yield a 4 bit magnitude comparator. The comparator has two 4-bit inputs (`A` = `A1` through `A4` with `A1` the MSB and `B` = `B1` through `B4` with `B1` the MSB) and and three 1-bit outputs `x, y, z`. The comparator outputs are derived from the outputs of the subtraction `A - B` as follows. If `(A-B)` is equal to zero then `A=B`; assert `x` and deassert `y` and `z`. If `(A-B)` yields no carry-out (borrow) then `A<B`; assert `y` and deassert `x` and `z`. If `(A-B)` is not equal to zero and the subtraction yields a carry-out (borrow) then `A>B`; assert `z` and deassert `x` and `y`. Note, the comparator outputs are active high.

**Task:** Design a 4-bit comparator.

**Deliverable (D1):** Annotated logic schematic for the comparator. Note, use adder's logic symbol in your schematic.

**Task:** Develop a structural VHDL model of the comparator using the VHDL models for the TTL ICs SN7486, SN7404, SN7408 and SN7483. VHDL models can be found in `~csci355/Lab5`.

**Deliverable (D2):** Electronic submission of source code (`make submit`).

**Task:** Map your VHDL model to a TTL-based physical design for the combinational system.

**Deliverable (D3):** IC logic schematic.

**Task:** Specify IC interconnections.

**Deliverable (D4):** One completed pin-out sheet (at least) for each IC employed in your physical design.

**Task:** In the laboratory, wire-up your physical design, verify its behaviour and sign-off on the design/implementation.

**Deliverable (D5):** A physical realization of the combinational system that behaves to specification. Details of the circuit-verification process. Student signature indicating that the circuit behaves as specified.

**Task:** Document any relevant results, explanations or comments.

**Deliverable (D6):** A section in your report entitled Results/Explanations/Comments in which you have detailed any relevant results, explanations or comments.

# NOTES

# Experiment 6

Consider the component declaration for a memory device called `lab6` given in `~csci355/Lab6/tb.cew`. The modeled device is a zero-delay SR single-bit memory comprising an SR latch, an SR gated latch with an active-high enable `C`, a negative master-slave SR flip flop with a clock `C` and a positive edge-triggered SR flip flop with a clock `C`. There are four outputs, one for each of the modeled SR latches/flip flops.

**Task:** Identify which of `w, x, y` and `z` corresponds to the `Q` (state) output of the four modeled SR latches/flip flops. To this end, develop a testbench that can discriminate between the four SR latches/flip flops.

For the master-slave device, design a test case that will demonstrate the phenomenon of ones catching (you may assume a `nor` gate realization).

**Deliverable (D1):** Electronic submission of testbench source code (`make submit`) and an explanation of your testing strategy in the form of annotated timing diagrams.

**Task:** Design a negative master-slave SR flip flop using `nor` gates and `and` gates.

**Deliverable (D2):** Logic schematic.

**Task:** Design a TTL-based physical design for the negative master-slave SR flip flop.

**Deliverable (D3):** IC logic schematic.

**Task:** Specify IC interconnections.

**Deliverable (D4):** One completed pin-out sheet (at least) for each IC employed in your physical design.

**Task:** In the laboratory, wire-up your physical design, verify its behaviour and sign-off on the design/implementation.

**Deliverable (D5):** A physical realization of a negative master-slave SR flip flop that behaves to specification. Details of the circuit-verification process. Student signature indicating that the circuit behaves as specified.

**Note** In addition to demonstrating the *normal* behaviour of your circuit to the instructor, you must also demonstrate the phenomenon of ones catching using the test-case developed in conjunction with deliverable D1.

**Task:** Document any relevant results, explanations or comments.

**Deliverable (D6):** A section in your report entitled Results/Explanations/Comments in which you have detailed any relevant results, explanations or comments.

# NOTES

# Experiment 7

Design and implement a synchronous digital system (with an active low asynchronous clear) to detect the non-resetting binary input sequence 1100 or 1000. Use the state table given below as the starting point for your design/implementation.

Design your circuit as a Mealy machine using T flip flops (to be constructed using JK flip flops with asynchronous preset and clear).

| Present State | Next State | | Output (`Z_BAR`) | | Encoding |
|---|---|---|---|---|---|
| | $X=0$ | $X=1$ | $X=0$ | $X=1$ | $Q_AQ_BQ_C$ |
| $S_0$ | $S_0$ | $S_1$ | 1 | 1 | 001 |
| $S_1$ | $S_4$ | $S_2$ | 1 | 1 | 101 |
| $S_2$ | $S_3$ | $S_2$ | 1 | 1 | 111 |
| $S_3$ | $S_5$ | $S_1$ | 0 | 1 | 010 |
| $S_4$ | $S_5$ | $S_1$ | 1 | 1 | 011 |
| $S_5$ | $S_0$ | $S_1$ | 0 | 1 | 000 |

**Task:** Design/develop the sequence detector's combinational parts.

**Deliverable (D1):** Next state K-maps, T FF input K-maps and the output K-map. Minimized equations for T FF inputs and for the output. Annotated logic schematic for the sequential circuit (both combinational parts and memory).

**Task:** Develop a structural VHDL model of the sequential machine using the VHDL models for the TTL ICs SN7476, SN7486 and SN7432. You are allowed two instances of SN7476 and, one instance of each of the other two ICs. VHDL models can be found in `~csci355/Lab7`.

**Deliverable (D2):** Electronic submission of source code (`make submit`).

**Task:** Map your VHDL model to a TTL-based physical design for the sequential system.

**Deliverable (D3):** IC logic schematic.

**Task:** Specify IC interconnections.

**Deliverable (D4):** One completed pin-out sheet (at least) for each IC employed in your physical design.

**Task:** In the laboratory, wire-up your physical design, verify its behaviour and sign-off on the design/implementation.

**Deliverable (D5):** A physical realization of the sequential system that behaves to specification. Details of the circuit-verification process. Student signature indicating that the circuit behaves as specified.

**Task:** Document any relevant results, explanations or comments.

**Deliverable (D6):** A section in your report entitled Results/Explanations/Comments in which you have detailed any relevant results, explanations or comments.

# NOTES

# Experiment 8

Design and implement a 2-bit counter that counts in the cyclic Grey codes

$$00_2\ 01_2\ 11_2\ 10_2\ 00_2\ 01_2\ \ldots$$

The counter accepts two synchronous control signals to set and clear the count, as described below.

| SET | CLR | $Z^+$ | Action |
|---|---|---|---|
| 0 | 0 | $Z$+1 | Up count |
| 0 | 1 | 00 | Clear count |
| 1 | 0 | 11 | Set count |
| 1 | 1 | - | Illegal input |

Design your synchronous circuit as a Moore machine using JK flip flops with asynchronous preset and clear.

**Task:** Develop the counter's state diagram/table and perform state assignment using Armstrong Humphrey Rules.

**Deliverable (D1):** State diagram and state table with state assignment.

**Task:** Design/develop the counter's combinational parts.

**Deliverable (D2):** Next state K-maps, JK FF input K-maps and the output K-maps. Minimized equations for JK FF inputs and for the outputs. Annotated logic schematic for the sequential circuit (both combinational parts and memory).

**Task:** Develop a structural VHDL model of the sequential machine using the VHDL models for the TTL ICs SN7474, SN7408 and SN7404. You are allowed two instances of SN7474 and, one instance of each of the other two ICs. VHDL models can be found in `~csci355/Lab8`.

**Deliverable (D3):** Electronic submission of source code (`make submit`).

**Task:** Map your VHDL model to a TTL-based physical design for the sequential system.

**Deliverable (D4):** IC logic schematic.

**Task:** Specify IC interconnections.

**Deliverable (D5):** One completed pin-out sheet (at least) for each IC employed in your physical design.

**Task:** In the laboratory, wire-up your physical design, verify its behaviour and sign-off on the design/implementation.

**Deliverable (D6):** A physical realization of the sequential system that behaves to specification. Details of the circuit-verification process. Student signature indicating that the circuit behaves as specified.

**Task:** Document any relevant results, explanations or comments.

**Deliverable (D7):** A section in your report entitled Results/Explanations/Comments in which you have detailed any relevant results, explanations or comments.

# NOTES

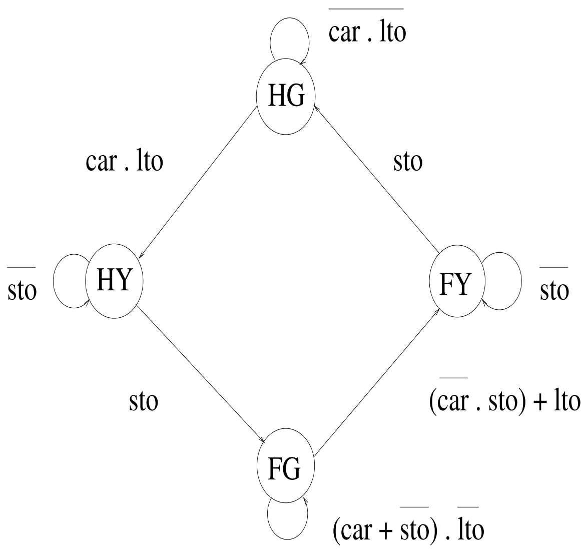
# Experiment 9

Design and implement a controller for a traffic light at the intersection of a highway and a farm-road. The behaviour of the controller is as follows. The highway lights remain green as long as there are no cars on the farm-road. When the highway lights have been green for a minimum of ten seconds, a car on the farm-road will cause the highway lights to go yellow for three seconds and then red. Then the farm-road lights will turn green and remain green for a minimum of three seconds. The presence of other cars on the farm-road beyond the three seconds will enable the farm-road lights to remain green for a maximum of ten seconds. The farm-road lights will then go yellow for three seconds and then red.

The controller is to have four states, highway green (`HG`), highway yellow (`HY`), farm-road green (`FG`) and farm-road yellow (`FY`). In state `HG`, the highway lights are green and the farm-road lights are red. In state `HY`, the highway lights are yellow and the farm-road lights are red. In state `FG`, the farm-road lights are green and the highway lights are red. In state `FY`, the farm-road lights are yellow and the highway lights are red. The controller is to have five control signals: `clear_BAR`, `car` , `sto`, `lto` and `clk`. The control signal `clear_BAR` is asynchronous. All other control signals are synchronous. When `clear_BAR` is asserted, the controller resets to state `HG`. When `car` is asserted, a car is present on the farm-road. When `lto` is asserted, a long-delay timer has reached the ten second mark. When `sto` is asserted, a short-delay timer has reached the three second mark.

There are five single-bit inputs and six single-bit outputs. The five inputs correspond to the control signals described earlier. Three of the outputs correspond to the red, green and yellow lights visible from the highway. They are `hw_R, hw_G, hw_Y`. The other three correspond to the red, green and yellow lights visible from the farm-road. They are `fr_R, fr_G, fr_Y`.

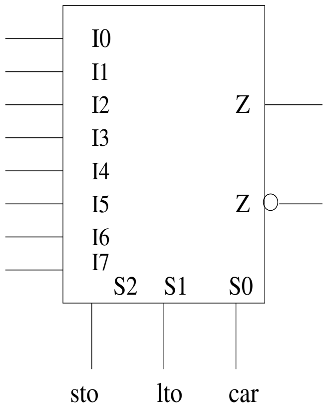
State transitions are shown in the following state diagram.

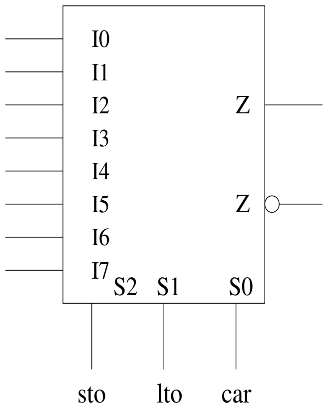


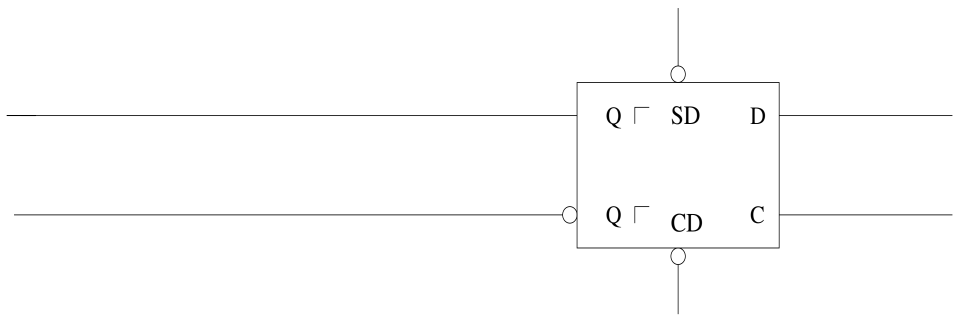

State assignment is shown in the following state assignment table.

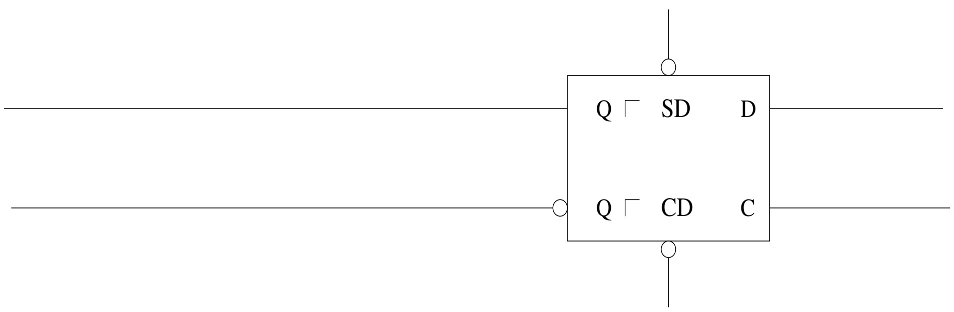
| State | Encoding |
|---|---|
| `HG` | 00 |
| `HY` | 01 |
| `FG` | 11 |
| `FY` | 10 |

Design your synchronous circuit as a Moore machine using two D flip flops (with asynchronous preset and clear). You must implement the next state combinational function using two 8 to 1 multiplexors. You must implement the output function using a maximum of four 2-input `nor` gates. The following incomplete logic schematic overviews the design you must employ.


I0
I1
I2
I3
I4
I5
I6
I7
Z
Z
S2
S1
S0
sto
lto
car


I0
I1
I2
I3
I4
I5
I6
I7
Z
Z
S2
S1
S0
sto
lto
car


Q
SD
D
Q
CD
C


Q
SD
D
Q
CD
C

**Task:** Develop the controller's state table.

**Deliverable (D1):** State table (using state names or state encoding).

**Task:** Design/develop the controller's combinational parts.

**Deliverable (D2):** Next state K-maps, D FF input K-maps and the output K-maps.

**Task:** Implement the next state and output functions using `nor` gates and multiplexors (as described previously).

**Deliverable (D3):** A completed logic schematic. All inputs and outputs must be shown.

**Task:** Develop a structural VHDL model of the sequential machine using the VHDL models for the TTL ICs SN74151, SN7474 and SN7402. You are allowed two instances of SN74151 and, one instance of each of the other two ICs. VHDL models can be found in `~csci355/Lab9`.

**Deliverable (D3):** Electronic submission of source code (`make submit`).

**Task:** Map your VHDL model to a TTL-based physical design for the sequential system.

**Deliverable (D4):** IC logic schematic.

**Task:** Specify IC interconnections.

**Deliverable (D5):** One completed pin-out sheet (at least) for each IC employed in your physical design.

**Task:** In the laboratory, wire-up your physical design, verify its behaviour and sign-off on the design/implementation.

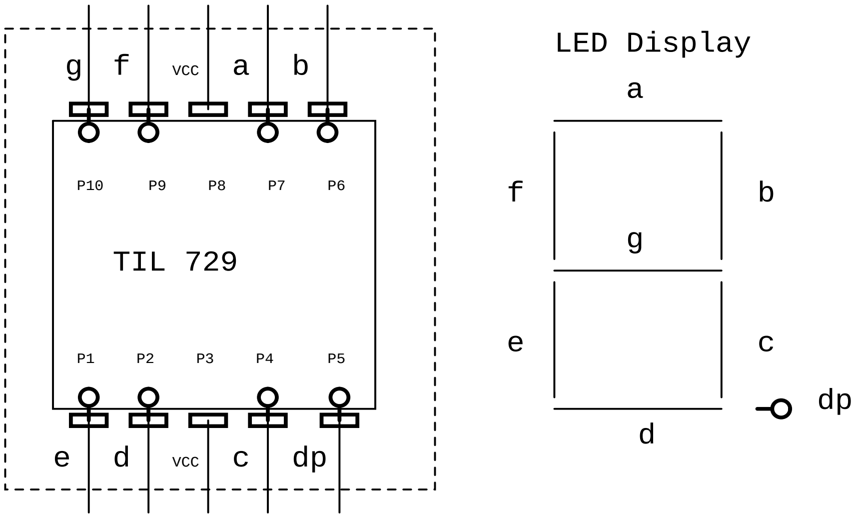
**Deliverable (D6):** A physical realization of the sequential system that behaves to specification. Details of the circuit-verification process. Student signature indicating that the circuit behaves as specified.

**Task:** Document any relevant results, explanations or comments.

**Deliverable (D7):** A section in your report entitled Results/Explanations/Comments in which you have detailed any relevant results, explanations or comments.

# NOTES

# Appendix A (Data Sheets)




Note, `a, b, c, d, e, f, g` and `dp` are active low and MUST be grounded through a resistor. For this task, use an R470 resistor IC. It is a 16 pin IC containing 8 resistors. Assume the normal IC orientation and pin labeling. The following pin pairs specify the the I/O pins for the 8 resistors: (`P1, P16`), (`P2, P15`) (`P3, P14`), (`P4, P13`), (`P5, P12`), (`P6, P11`), (`P7, P10`), (`P8, P9`).