LAB 3: A Debounced Switch and Counters

OBJECTIVES

To understand the design, function and operation of a debounced switch and simple counter circuits.

MATERIALS

- Your entire lab kit (including your DAD/NAD)
- UF's *DAD/NAD Waveforms 2015 Tutorial*
- Suggested Quartus Components
  - In "others | maxplus2" library
    - 7474: Dual D-flip flops
  - In "primitives | storage" library
    - dff
  - In "primitives | logic" library
    - not, and2, or2, bor2, etc.
  - In "primitives | pin" library
    - input, output
  - In "primitives | other" library
    - vcc, gnd

You must adhere to the *Lab Rules and Policies* document for **every** lab. Re-read, if necessary. All equations and K-Maps must be included in your pre-lab document.

INTRODUCTION

NOTES ON UF-3701 BOARD AND QUARTUS

The ground and Vcc connections are the most important one for the 3701 PLD board. Connect the two ground pin (labeled GND on your 3701 PLD PCB) to the ground on your protoboard. Similarly, connect the two power pins (labeled 3.3V on your 3701 PLD PCB) to the power on your breadboard. Be very careful that you do **NOT** reverse these pins; if you do, the chip will be destroyed and you will need to spend a lot of time and money to correct the problem.

One of the pins labeled C2P (i.e., CLK2P, MAX 10 pin G9) or C3P (i.e., CLK3P, MAX10 pin F13) should be used for any clock input. These are the available global clock pins on our PLD PCB.

***Warning**: Tri-state all unused PLD pins, as described in the Quartus Tutorial.*

Don't forget to also ground your DAD; the DAD/NAD's ground wire is black.

DEBOUNCED SWITCHES

The switches that you have been using this semester are known as single-pole-single-throw (SPST) switches. When you move the SPST switch in a switch circuit from ON to OFF or from OFF to ON, the resulting output bounces around between low and high voltages for a short time. If the switch circuit output is used as a synchronizing signal (such as a clock) in a digital machine, weird things will happen. If the machine is a counter, the count may seem to jump wildly. This is obviously undesirable, so a debouncing circuit must be built. We discussed several debouncing circuits in class. You will design and build one of these debounced switch circuits using a single-pole-double-throw (SPDT) switch (see Figure 0). You will use this debounced switch circuit in this lab and rest of the labs this semester. The SPDT switch has three aligned pins. The center pin is connected to one or the other of the outside pins, depending on the position of the switch. You can use your multimeter on the resistance (Ω) setting to verify the operation of this switch. There are two pins on the switch in addition to the three aligned pins. These two pins (shown on the middle of the top image in Figure 0) have no useful electrical purpose and are used for mounting only. In lab 0, you mounted the SPDT switch onto a small PCB (printed circuit board) shown in the bottom image of Figure 0.

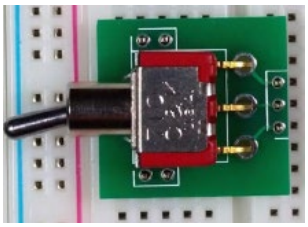

Figure 0: SPDT switch

Your debounced switch circuit should use the two axial resistors (sometimes incorrectly called radial resistors) in your lab kit. A 1 kΩ ¼ W (which you have in your lab kit) axial resistor is shown in Figure 1.

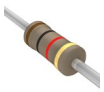

Figure 1: Axial resistor.

When flipping the SPDT switch, hold the switch down with another finger. If this switch circuit appears to be bouncing, you probably have poor soldering; re-solder the SPDT pins on the small SPDT PCB.

COUNTERS

A synchronous counter is a device that progresses through a known sequence with every clock input signal. The counter advances to the next state/number at a rising (or falling) edge of each clock pulse. The counter sequence is arbitrary, i.e., it may count up, down, or in some strange sequence. The counter you will design in this lab will have a custom count sequence with some special additional inputs.

PRE-LAB REQUIREMENTS

1. Make a debounced switch circuit for your clock input (as discussed in class) using your SPDT switch and other circuitry. You may use a NAND chip or a NOR chip in your design. (Although a 74'74 chip could be used, you are **not** allowed to use this in 3701.) Another alternative is to use the PLD itself to provide the two NAND or two NOR gates. If you build the debouncer with NAND or NOR chips, you should place it in a corner of your breadboard. You are expected to know how to use the PLD for the two NAND or two NOR gates, i.e., to use only the PLD for your circuit designs (in addition to resistors, an SPDT switch, [regular] SPST switches, and LED switches). **Hint: This means that you should use the PLD for the two NANDs or NORs of the debounced switch circuit in this lab, but also know how to do it with NAND or NOR chips.**

   There is no easy way to test your debounced circuit without an oscilloscope. Luckily, the DAD/NAD has an

LAB 3: A Debounced Switch and Counters

oscilloscope function (called Scope). See the Oscilloscope (Scope) section of the DAD/NAD Tutorial for help in determining how to measure the switch bouncing. Quartus' simulation can **NOT** be used to simulate the debounce circuit that was taught in class because there are no resistor components available in Quartus. A voltmeter will not help, since the bounce rate is in the order of milliseconds. Instead, you will test your debounced circuit with a counter designed in the next part of the pre-lab. This will also serve as a your first counter design, to assure that you understand the proper design technique before attempting a more complex counter later in this lab.

Use your DAD/NAD to measure the bouncing of a normal (SPST) switch circuit. See the Appendix for information on settings that may be helpful to observe the switch bouncing. Move the switch from one position to another and get a screen shot of the bouncing with an appropriate time base. Move the switch back to the original position and get a second screenshot. Move the switch one more time and get a third screen shot. It may be necessary to try each of these a few times to see the bouncing. Put at least three of these screenshots into your lab document and interpret these images by explaining the number of clocks that would occur if this switch was connected to a 5-bit counter that counts from 0 to 31.

Now use your DAD/NAD to measure the possible bouncing of your debounced circuit (with the SPDT switch). Move the switch from one position to another and get a screen shot of the bouncing with an appropriate time base. Move the switch back to the original position and get a second screenshot. Put both of these screenshots into your lab document and interpret these images explaining the number of clocks that would occur if this switch was connected to a 5-bit counter that counts from 0 to 31.

2. Design a counter (shown in Figure 2) to count through the sequence 00, 10, 11, 01, 00, … Note that there is only a single input (CLK) and two active-high outputs (Q1 and Q0). (To start the counter at a known value, use the pre-sets and pre-clears of the two flip-flops.)

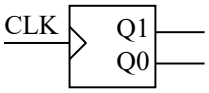

Figure 2: Simple counter block diagram.

   a. Make a next-state truth table. The "inputs" for this table are Q1 and Q0; the "outputs" are Q1+ and Q0+.
   b. Using D-flip-flops, determine the next state equations for $D_i = Q_i^+ = f(Q1,Q0)$. Use K-maps, if necessary, for each D_i to get MSOP or MPOS equations. Note: There will be two **2-input** K-Maps.
   c. Design the required counter circuit in Quartus (called `Lab3_2bit_Cnt`). (I suggest that you do it first on paper, but this is not required and will not be submitted.) I suggest that you use one of the below two possible D- flip-flops available in Quartus.
      i. Use "others | maxplus2 | 7474" for the left item in Figure 3.
      ii. Use "primitives | storage | dff" for the right item in Figure 3.
   d. Simulate the circuit and, as always, annotate this simulation. Verify that your design counts as required with each rising CLK edge.

3. Create a **component** in Quartus for 7-segment Decoder! Test your 2-bit counter in part 2 with this component**.**

4. The above items, including circuit schematic (with PLD pin numbers) and annotated Quartus simulation results should be part of the submitted lab document. (As usual, all pre-lab material must be submitted through Canvas prior to the start of your lab. Be sure to also submit your archive files.)

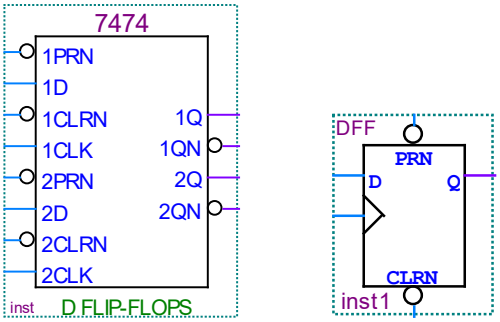

Figure 3: Two D-FF available in Quartus.

5. Test your 2-bit counter design, using your debounced switch for the clock input. Use your DAD/NAD and appropriate LED circuits for the count (Q) outputs.

   a. Download your counter design to your PLD board with **nothing else connected to the board** except power and ground. Remove power from your breadboard.
   b. Move your PLD board onto a pre-wired area with your designed debounced CLK input and the two count outputs. Reconnect power to your breadboard.
   c. Toggle (flip) the debounced CLK input switch to verify that your counter counts as expected. If your counter output does not exactly match the required count sequence as you toggle the switch input (but it worked in simulation), then your debounced switch circuit is **not designed and/or built correctly**. If necessary, verify your debounced switch circuit design and construction. (The only way to easily test your debounced switch circuit is with a counter.)

6. Replace the debounced CLK input circuit to your counter and replace it with a normal (un-debounced) SPST switch input circuit for the CLK. Write down the outputs with 10 successive clocks. Compare each successive count to what you **should** get. How does counting with an un-debounced clock input compare to counting with a debounced clock input? Put this info in your submitted lab document.

LAB 3: A Debounced Switch and Counters

7. Design a counter (called `Lab3_3bit_Cnt`) that will count forward with the following sequence:

   000, 100, 010, 011, 111, 000, …

   This counter will also count backward in the reverse order:

   000, 111, 011, 010, 100, 000, …

   A block diagram for the counter is shown if Figure 4.

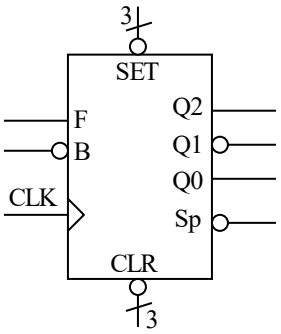

Figure 4: Forward/Back counter block diagram.

Your counter can also pause the counting. These three modes (forward [F], backward [B], pause) will be controlled with 2-inputs, F and B. When neither forward nor backward is true, the counter will ignore the CLK input and hold its count value, i.e., pause.

F and B should **never** be simultaneously true, so your counter should deal with this case in the most cost-effective fashion, i.e., if you assume that a user will never make both inputs true, design the most inexpensive circuit that you can that accomplishes the required goals, i.e., use "don't cares." Note that the counter does not include $Q_2Q_1Q_0$=%001, %101 and %110, where % is a prefix for binary. These three counts should contribute "don't cares" in your next-state truth tables and K-maps.

Your counter should have a means to **asynchronously** set and clear **each** bit. SET(L) and CLR(L) are the inputs to asynchronously set and clear a particular counter bit. These SET and CLR inputs will allow you to start the counter at any desired count. (If you initialize your counter at the count $Q_2Q_1Q_0$=%001, %101 or %110, the next count is not specified in the problem description. The next count will be determined by the values selected for the "don't cares" associated with these counts.)

As you may recall, when we first started discussing circuits with feedback, I stated that with these types of circuits it is often easier to deal with voltages rather than with logic. Let me suggest that you design this (and all) counter(s) with **active-high state-bits** (to generate the next-state circuits) and then generate the appropriate output circuits with the required activation levels. In this case, use active-high Q_2, Q_1, and Q_0 in your design of the counter next state circuits. But when creating the final circuit, the outputs will be as shown in the block diagram of Figure 4.

Finally, the counter should have an additional output indicating the count is at a "special value," `Sp`. Special should be true only when the count is "**111**" and `F` is true or when the count is "**011"** when `B` is true.

   a. Make a next-state truth table with the inputs: F, B, Q2, Q1, Q0 and outputs $Q2^+$ $Q1^+$, $Q0^+$, and Sp. (Ignore the SET and CLR for the design. These are controlled directly with the FF set and clear inputs.)
   b. Using D-flip-flops, determine the next state equations for $D_i = Q_i^+ = f(F,B,Q2,Q1,Q0)$. Use a K-map for each output to determine MSOP or MPOS equations. Note: There will be three **5-input** K-Maps.
   c. Determine the equation for the output Sp. Use a K-map to get an MSOP or an MPOS equation.
   d. Design the required counter circuit (called `Lab3_3bit_Cnt`) in Quartus. (I suggest that you do it first on paper, but this is not required and will not be submitted.) Don't forget to include the SET and CLR inputs in your circuit. (In Quartus, choose D-FF's with asynchronous Set and Clear inputs. Both of the D-FF's shown in Figure 3 have asynchronous Set and Clear inputs.)
   e. Simulate the circuit and add annotations.
      i. Verify that your design counts forward, counts backwards and holds the count with the appropriate input combinations.
      ii. Verify that each bit can be set and cleared by using the SET_i and CLR_i (i=0,1,2) inputs.
      iii. What is the next count for each value of F and B for $Q_2Q_1Q_0$=%101 and %110?
      iv. As always, include the circuit schematic (with PLD pin numbers) and annotated Quartus simulation results in your lab document. (Also as usual, submit your archived Quartus file.)
   f. Build this circuit on your breadboard. (You can undo the 2-bit counter if you would like to, since you will not demo this in lab.) Reprogram your PLD board, and verify that this counter operates properly. Use a debounced-switch circuit for the CLK input. Use appropriate switch circuits for the other inputs and your DAD/NAD for the count (Q_2, Q_1, Q_0) and Sp outputs. (Note that the DAD/NAD does not deal directly with active-low outputs, so you will either have to interpret the active-low outputs in reverse or provide the active-high versions as well as the active-low outputs for the DAD/NAD.) In addition to using the DAD/NAD for the outputs, you must also use the 7-segment display (with the Hex to-7-segment Decoder designed in Lab 2) for the Q outputs. Use the active-high versions of Q to directly view decimal version of the count values.

8. As always, put your design and simulation in your lab document and submit this file along with your design archives.

9. Now re-design the counter using a T-FF for bit 1 (the most significant bit) and a JK-FF for bit 0 (the least significant

bit). This new design will require that you add to your next-state truth table from part 2a, determine equations for the T1, J0, and K0 inputs, draw and simulate the new circuit diagram in Quartus, and verify with the simulation that it counts properly. You do **NOT** need to build/demo this circuit on your breadboard.

Note: Since you may find that you need to make a change with one or both of your counters during lab, leave a spot to program the PLD on a corner of your bread board.

IN-LAB

1. As always, bring a printout of your Summary document to lab.
2. Demonstrate your 3-bit forward/back counter design.
   a. Verify that your design counts forward, counts backward, and holds the count with the appropriate input combinations.
   b. Verify that each bit can be set and cleared by using the SET_i and CLR_i (i=0,1,2) inputs.
3. Now replace the debounced CLK input with a normal (non-debounced) switch circuit. Demonstrate your 3-bit counter design with the non-debounced CLK.

LAB 3: A Debounced Switch and Counters

APPENDIX

In order to see the switch bouncing, use the following DAD/NAD settings.

- Time Base: 50 us/div (or 20 us/div)
- Offset: 0
- Level: 1.5 V
- Condition: Either
- Mode: Repeated, Normal
- Range: 1 V/div

Figure A.1 shows the time base at 1 ms/div. Note that the bouncing is apparent, but the amount of bouncing cannot be determined because of the too large time base.

Figures A.2 and A.3 show bouncing with the time base at 50 us/div and 20 us/div, respectively.

A recording of switch bouncing with different time base settings is available at the following location on our class website: https://mil.ufl.edu/3701/docs/Bouncing.mov.

Figure A.1: Bouncing with time base of 1 ms/div.

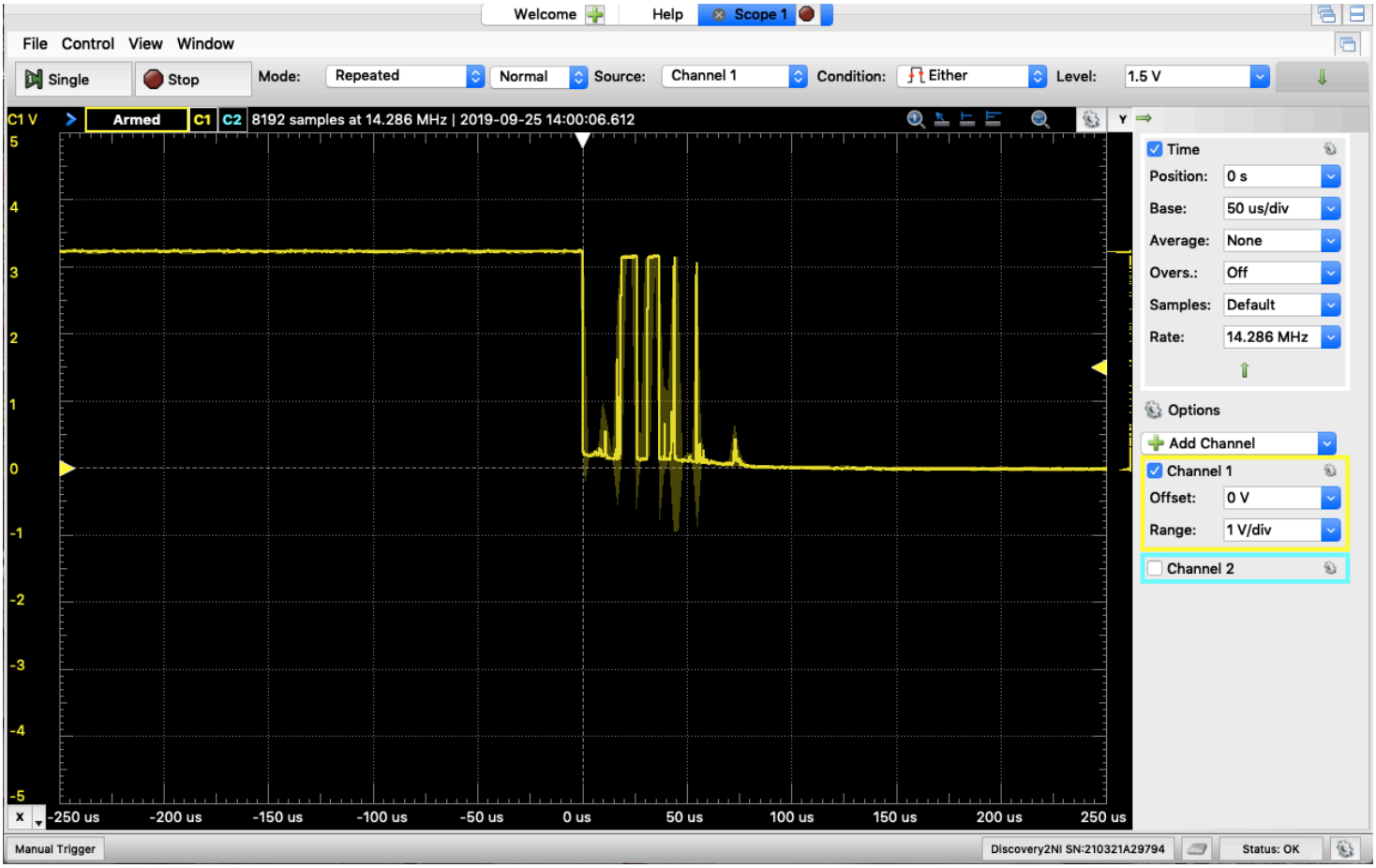

Figure A.2: Bouncing with time base of 50 us/div.

LAB 3: A Debounced Switch and Counters

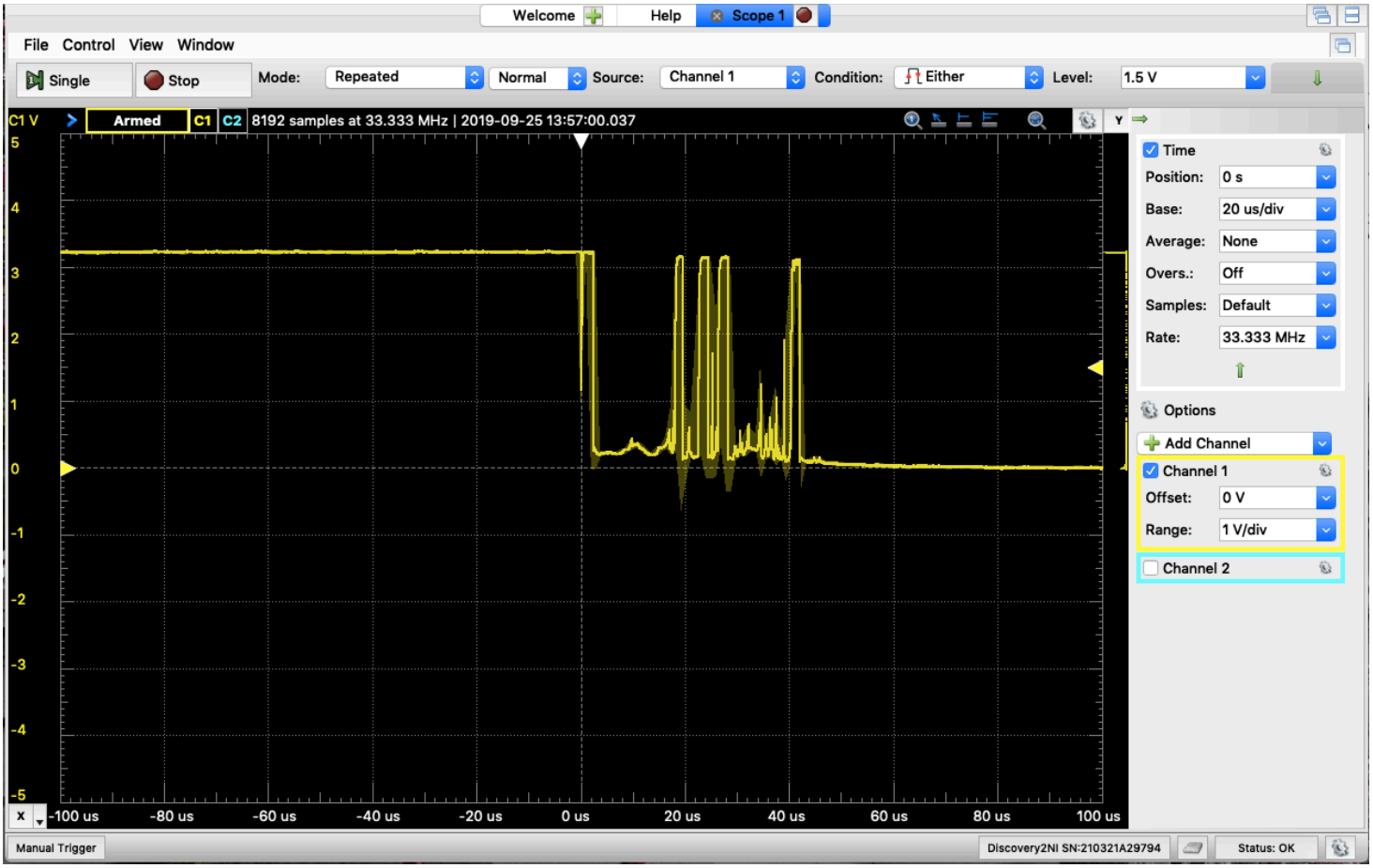

Figure A.3: Bouncing with time base of 20 us/div.