EELE 367 - Introduction to Logic Circuits

Lab 9(c) – Counters using Single Process and a 2^n Clock Divider

Objective

The objective of this lab is to gain experience designing counters with a single process in VHDL. This lab will also give experience building a selectable, 2^n clock divider based on a ripple counter. The clock divider will allow the counter to be run at a speed that can observed on the LEDs of the DE0-CV board. This lab will also give experience using a logic analyzer to measure the frequency of a counter.

Outcomes

After completing this lab you should be able to:

- Create a counter using a single process in VHDL.
- Create a selectable, 2^n clock divider subsystem.
- Measure the frequency of the counter using a logic analyzer.

Deliverables

The deliverable(s) for this lab are as follows:

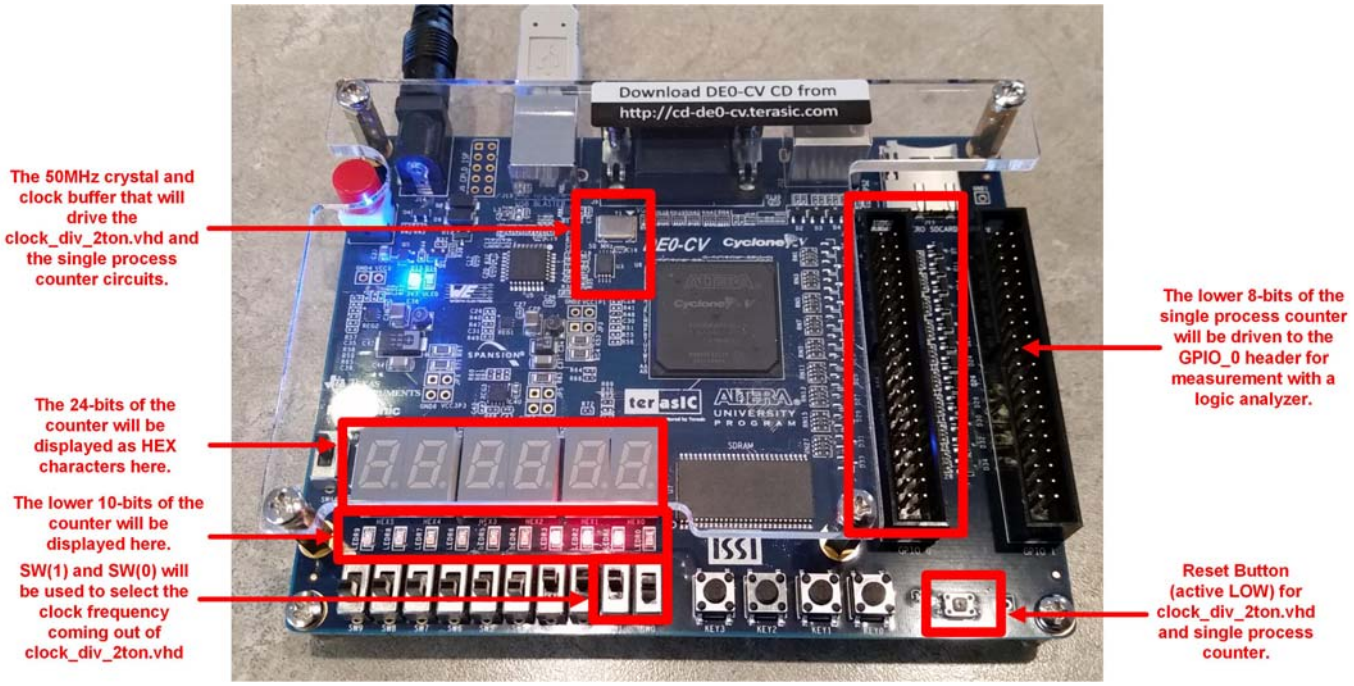
- Demonstration of your single process counter showing its output on the DE0-CV LEDs and character displays (60%).
- Demonstration of a logic analyzer measurement of the counter (30%).
- Upload your top.vhd file and measurement screenshot to the lab DropBox (10%).

Lab Work & Demonstration

Part 1 – System Design

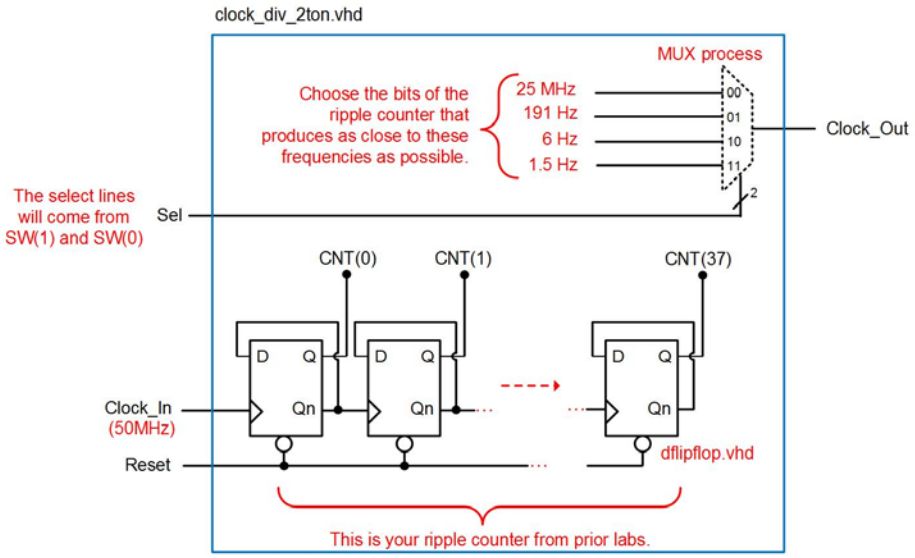
You are going to design a 24-bit counter using a single process within VHDL. The 24-bits will be used to drive the six HEX character displays on the DE0-CV board through your char_decoder.vhd subsystem. The lower 10-bits of the counter will drive the 10x red LEDs on the DE0-CV board. The lower 8-bits of the counter will drive the GPIO_0 header on the DE0-CV board to enable a logic analyzer measurement of the counter. Note that you will need to add the GPIO_0 port to your entity and find the pin assignments for this I/O from the DE0-CV User's Guide.

You will need to first create a programmable clock divider that can drive the counter at a rate that is visible on the DE0-CV FPGA board. The clock divider will be based on your 38-bit ripple counter from prior labs. Since each bit of the ripple counter produces a frequency that is ½ of the prior bit, the clock frequencies available are multiples of 2^n. As such, we typically call this a "2^n Clock Divider". You are going to put this ripple counter into its own VHDL file called clock_div_2ton.vhd. The clock divider will take in the 50MHz clock from the DE0-CV crystal and produce one of four selectable clock frequencies (25 MHz, 191 Hz, 6 Hz, and 1.5Hz). The output clock will be selected by a multiplexer process whose select lines come from the slider switches on the FPGA board. The following figure shows DE0-CV I/O that will be used in this lab.



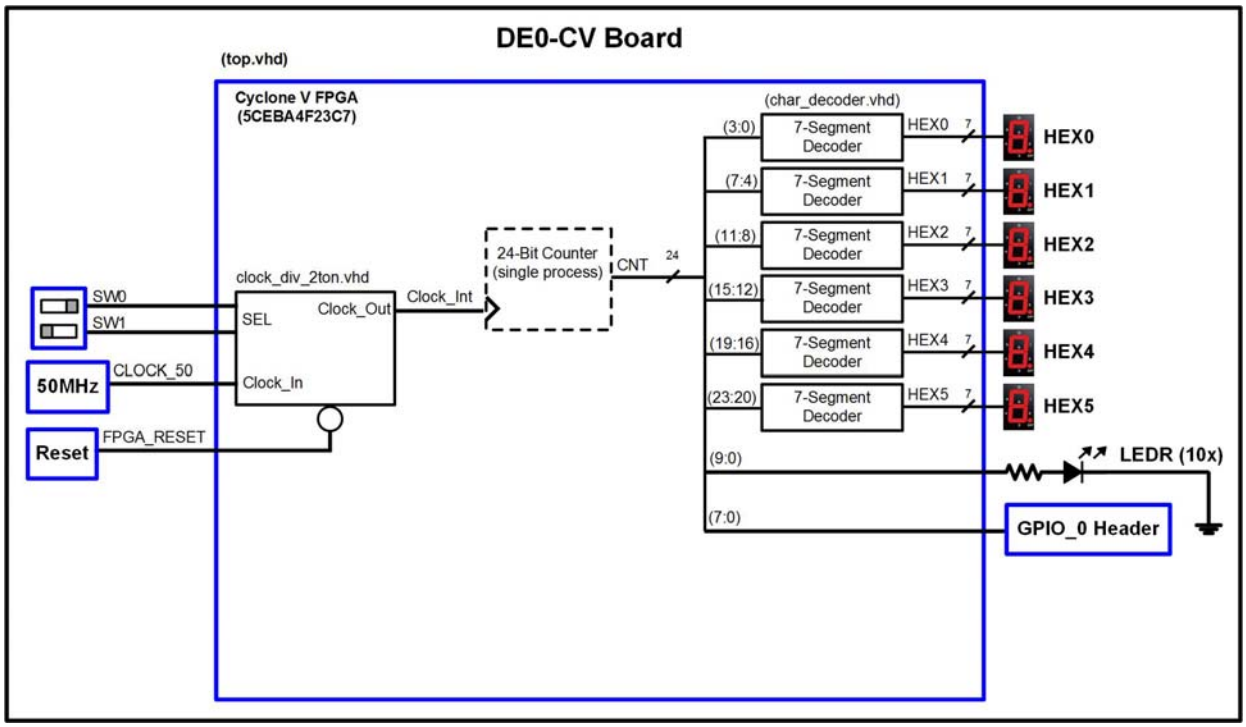
A) Create a new Quartus project by copying your one of your prior labs. Name it "Lab9c_Counters_using_1_Process_n_2ton_Clock_Div".

B) Create the 2ⁿ clock divider subsystem. This will require creating a new VHDL file in Quartus called clock_div_2ton.vhd. Within this new file you will insert your 38-bit ripple counter from prior labs. You will then route four of the bits of the counter to a multiplexer process. Choose the four bits so that they give clock frequencies 25 MHz, 191 Hz, 6 Hz, and 1.5 Hz. The multiplexer process will select one of these four signals to be driven out as the "Clock_Out" signal. The multiplexer process will have two select lines. The following figure gives the block diagram of the clock_div_2ton.vhd subsystem.




```
entity clock_div_2ton is
  port (Clock_In  : in   std_logic;
        Reset     : in   std_logic;
        Sel       : in   std_logic_vector(1 downto 0);
        Clock_Out : out  std_logic);
end entity;
```

C) Create a 24-bit counter using a single process in your top.vhd. The counter should be clocked with the output of the clock_div_2ton.vhd subsystem and receive its reset from the FPGA_RESET push button on the DE0-CV board. You should use SW(1) and SW(0) as the select lines for the clock_div_2ton.vhd subsystem. You should display the HEX values of the 24-bit counter on the 6x HEX displays. You should route the lower 10-bits of the counter to the red LEDs. You should route the lower 8-bits of the counter to the GPIO_0 header. The following block diagram shows the overview of the top level of your system.



D) Compile your design. Fix any Errors.

E) Download and test your design. Fix any Errors.

F) Demonstrate your design.

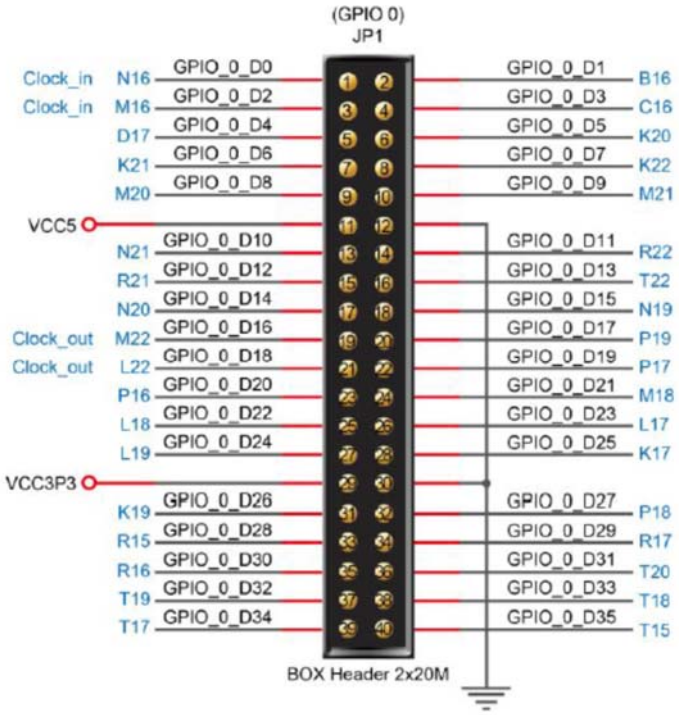
DEMO – Show the lab instructor the proper operation of your system. You should be able to change the frequency of the counter values being displayed on the LEDs and HEX displays of the DE0-CV board.

Part 2 – Taking a Logic Analyzer Measurement of your Counter

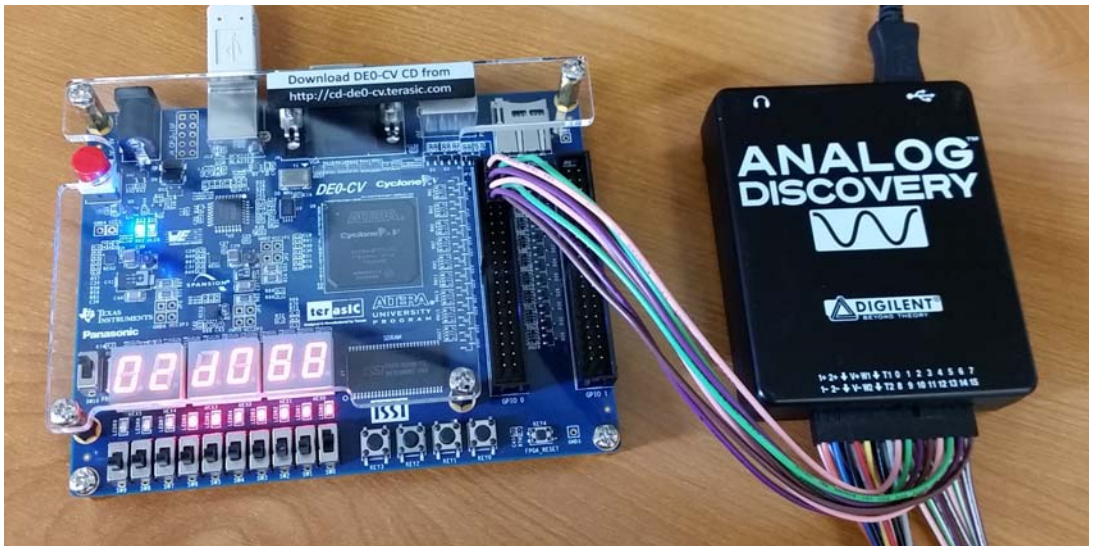
Now you are going to take a logic analyzer measurement of the lower 8-bits of your counter, which are being driven to the pins on the GPIO_0 header. A logic analyzer is an instrument that measures the digital values on a set of signals (aka, a "bus"). Since only 1's and 0's are stored, the circuitry to implement a logic analyzer is much simpler than an oscilloscope. This allows the logic analyzer to have more channels. Logic Analyzers can have anywhere between 16 to 128 channels. This is useful when debugging digital systems because entire vectors can be observed in real-time. The busses can be displayed in different radices and in both waveform and listing format.

You are going to use a portable instrument called the "Analog Discovery". This is a low-cost instrument that contains two channels of oscilloscope, two AWG outputs, two DMM channels, two power supplies, and 16 channels of logic analysis. The Analog Discovery is connected to a PC through a USB cable. The USB cable provides power to the Analog Discovery and allows the data to be viewed in an application called *Waveforms*.

A) Connect the Analog Discovery to the GPIO_0 port on the DE0-CV. The logic analyzer channels on the Analog Discovery are labeled as numbers (i.e., 0, 1, 2…). You should connect the lower 8 channels of the logic analyzer to the lower 8-bits of the GPIO_0 header. The following figure shows an excerpt from the DE0-CV User's Manual. Note that the logic analyzer channels 0-7 are the wires that are NOT striped. Also connect one of the grounds of the Analog Discovery to a ground on the GPIO_0 header.

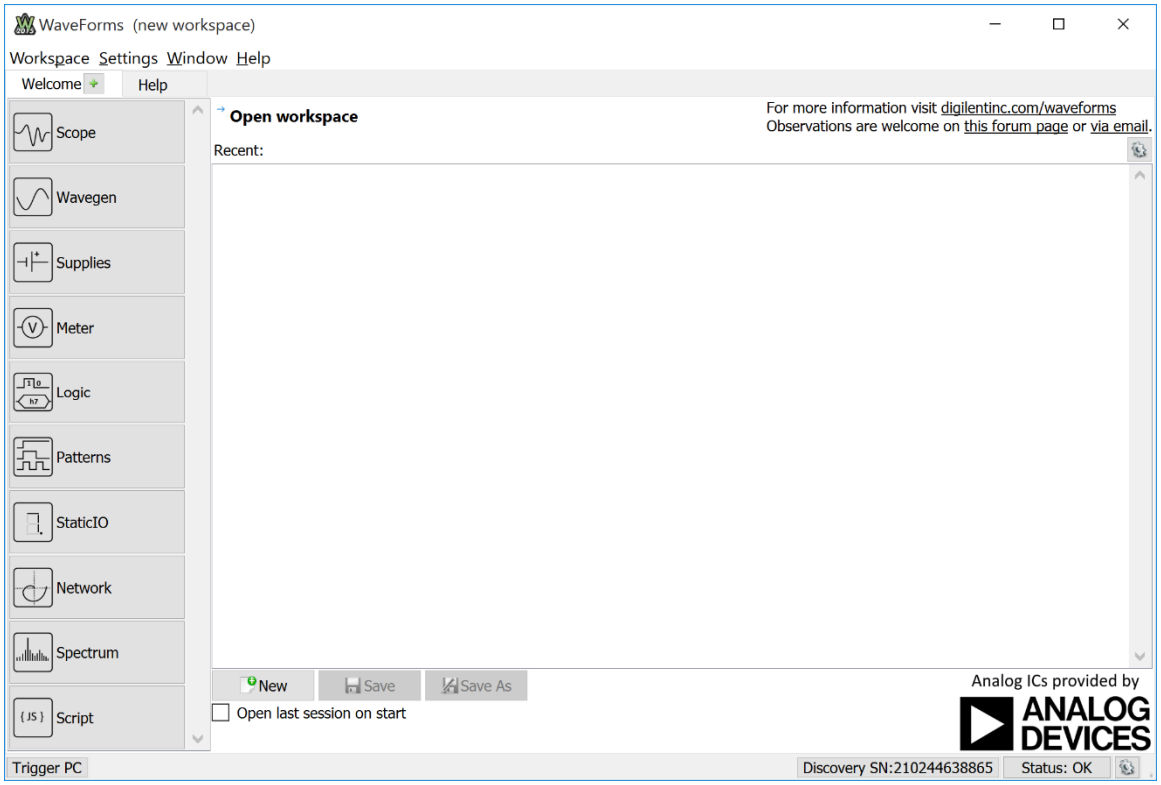


Your connection will look like this:



B) Plug in the Analog Discovery to the PC (if not already plugged in).

C) Launch the Analog Discovery software *Waveforms* (Start – All Programs – Digilent – Waveforms – Waveforms). If the software is not installed, it can be downloaded from the Digilent website. The software will automatically recognize the Analog Discovery and connect. The following window will appear:



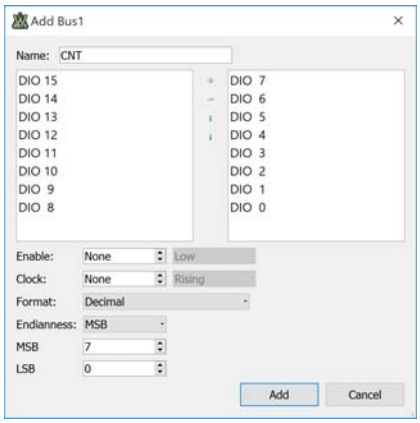
From this window, you can launch all of the tools associated with the Analog Discovery. Each tool brings up a new tab within the workspace window. Multiple tools can be launched and ran at the same time.

D) Launch the logic analyzer and configure the measurement.

- Click on the “Logic” icon. This will bring up a tab with a blank waveform on it.

- Click on the “Click to Add channels” button. A drop-down will appear. Select “bus”. This window allows you to setup a new bus (or vector) and assign which channels of the logic analyzer are going to be used.

  - Name the bus “CNT”
  - Highlight channels “DIO 0” to DIO 7” and then click the “+” button to add them as the active channels for this bus.
  - The “enable” and “clock” selections should be left at “None”.
  - Leave the format as “Decimal”. This is where you can select different radices.
  - Leave the “Endianness” at “MSB” and the MSB=7, LSB=0.
  - Click “Add”.

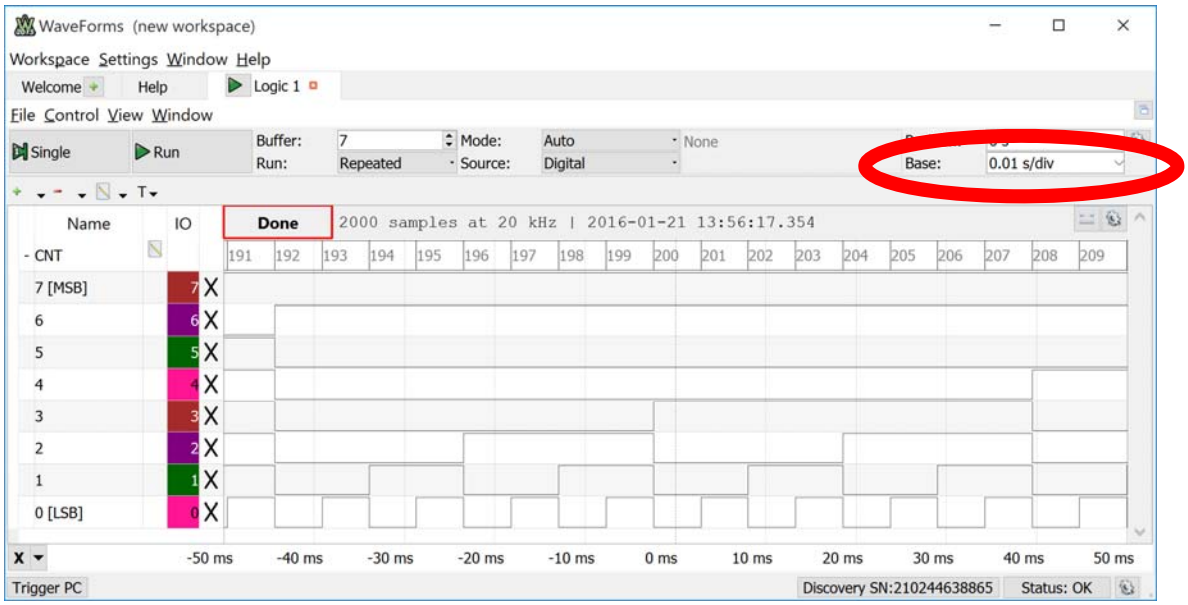


The new bus will show up in the waveform window.

E) Set the frequency of your counter to 191 Hz using the SW(1) and SW(0) switches. The logic analyzer will not actually run at 25MHz so we can't take measurements at that counter speed. The 6 Hz and 1.5 Hz speeds are too slow to build up any interesting data in a reasonable amount of time.

F) Run the measurement. While there are a variety of settings that you can still configure, go ahead and click on the "Single" button to take a measurement. After the measurement, you will see a decimal value for CNT and also some edges on the signals (7:0). You will note that you are probably zoomed out too far to see anything meaningful.

Play with zooming out by selecting different values in the *Base* setting drop-down. Each time you zoom out, you'll need to click the "Single" button to re-run the analyzer and fill the screen. At 191 Hz, a setting of 0.01 s/div is a good zoom factor.

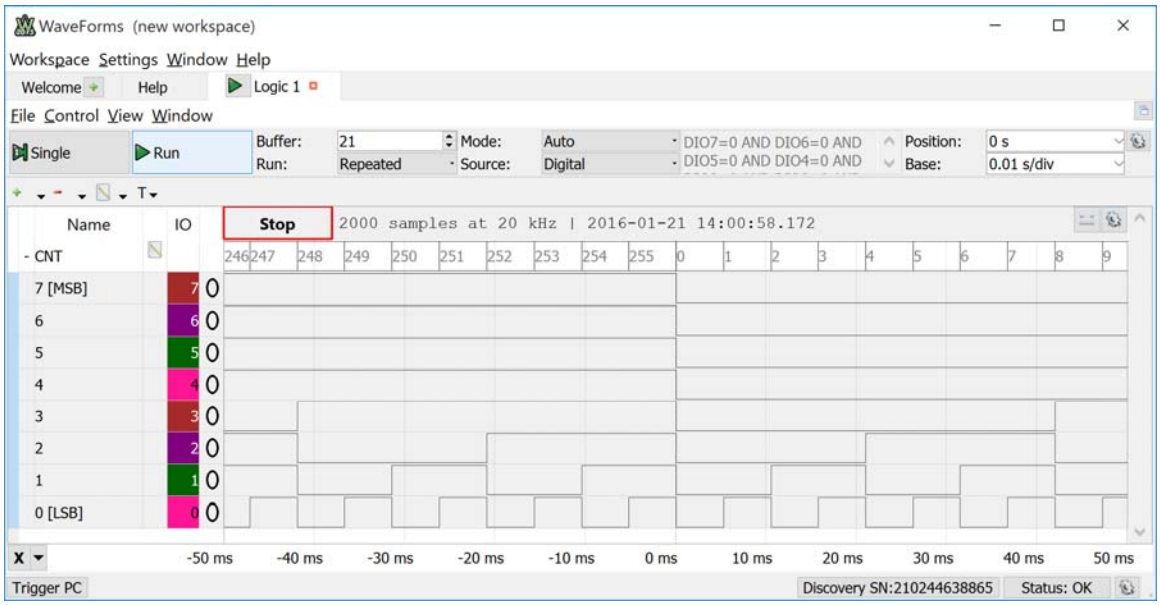


G) Setup the Trigger. A "trigger" is a pattern that the logic analyzer will search for and then place it in the center of the measurement. Let's setup the trigger to look for CNT=x"00".

- Highlight all of the bits within CNT.

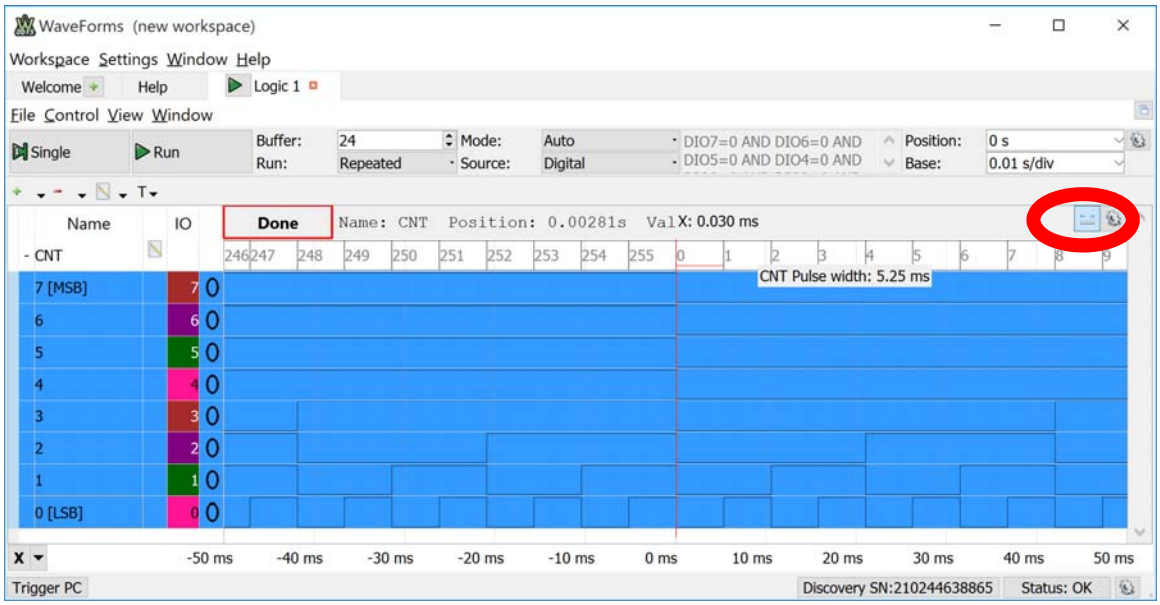
- Click the trigger button (it is the "T" right below the run button). Select "0 LOW".

Now when you click the "Single" button, you'll notice that the center of the waveform screen is the transition between 255 to 0.



H) Run analyzer in real time. If you click the "Run" button (instead of the "Single" button), the analyzer will continually take measurements and replace the data on the screen. Click "Run" and allow the analyzer to free run. Since we have the trigger set to CNT=x"00", the data will always look the same. To prove to yourself that the data is actually live, unplug one of the channels from the DE0-CV board and observe that the signal goes flat in the analyzer.

I) Measure the frequency of the counter. You can take a variety of measurements using the "HotTrack" feature within the waveform window. This button is in the upper right corner of the waveform pane of the window. It is to the left of a gear icon. Click on "HotTrack". Now as you move your cursor around on the waveform, you'll see measurements pop up. Place your cursor over the bus itself and measure the period.



Does this period make sense? Take its inverse to find the frequency. How close are you to what you expect?

G) Save your workspace. You can save your Analog Discovery measurement setup. This is useful for logic analyzer measurements in which there are numerous steps required to to setup the vectors and triggers.

- Click “Workspace – Save”.

- Give a descriptive name and save within your Quartus project directory.

H) Take a screenshot of your measurement. Use either a snipping tool or Alt-PrtScr to take a screenshot of your logic analyzer measurement. Save as a JPG. You will need to upload this to the lab DropBox.

I) Demonstrate your design.

DEMO – Show the lab instructor your logic analyzer measurement. Once checked off, upload your top.vhd file and measurement screenshot to the lab DropBox..

Lab Grading

Demo 1 – Proper operation of counter displayed on LEDS	**_____ / 60**
Demo 2 – Logic analyzer measurement of your counter	**_____ / 30**
Review of your top.vhd file and measurement screenshot	**_____ / 10** (uploaded to DropBox)
Total	**_____ / 100**

Post Lab Survey (Just for your own knowledge. This is not graded)

1) Can you create a counter using a single process in VHDL?
2) Can you create a selectable 2^n clock divider?
3) Can you use a logic analyzer to measure the frequency of a counter?