

Digilent Analog Discovery (DAD) Tutorial

6-Aug-15

INTRODUCTION

The Diligent Analog Discovery (DAD) allows you to design and test both analog and digital circuits. It can produce, measure and record many types of signals. It allows circuit designers to both simulate circuits and observe their behaviors for testing, debugging and other kinds of signal analysis. Follow this link <http://tinyurl.com/DAD-vids> for short videos, and <http://tinyurl.com/DAD-page> for product page. Figure 1 shows the normal setup for a DAD. Figure 2 shows the functions of each of your DAD's pins.

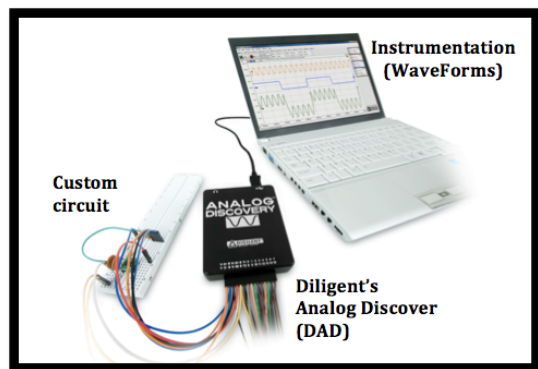


Figure 1: Connection of the DAD to a circuit and a USB port on a computer

Your DAD has the following basic features.

- 2-Channel Oscilloscope
- 2-Channel Waveform Generator
- 16-Channel Logic Analyzer
- 16-Channel Digital Pattern Generator
- $\pm 5\text{VDC}$ Power Supplies
- Spectrum Analyzer
- Network Analyzer
- Voltmeter
- Digital I/O
- Supported by MATLAB / MATLAB student edition

IMPORTANT NOTE

A ground connection is necessary between your circuit and the ground wire (black color) of the DAD. Generally all circuits in a design should have common ground connections.

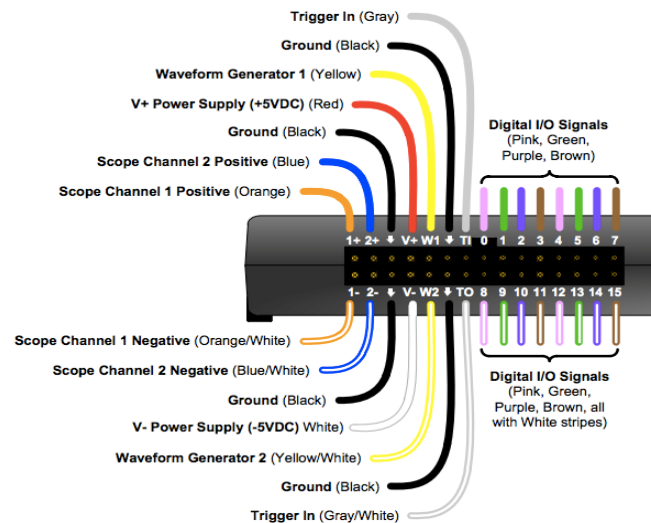


Figure 2: DAD pin configuration

DAD -- ANALOG FUNCTIONS

This section demonstrates some of the basic features of the DAD using Arbitrary Waveform Generator (AWG) and Scope instruments in the WaveForms software. AWG and Scope both make up the analog functions of the DAD – the former serves as analog output, while the latter serves as analog input. A breadboard board, augmented with a few other parts, can provide you a means to build and verify the proper operation of digital circuits. Some of the useful devices are described below.

Arbitrary Waveform Generator (AWG)

The AWG has two channels that can independently generate both standard waveforms (sine, triangular, sawtooth, etc.), as well as arbitrary waveforms that are defined using standard tools such as Excel. It supports waveform generation with maximum amplitude voltage of $\pm 5\text{V}$, and also features various customization controls for sweep, envelopes, FM and AM modulation.

Digilent Analog Discovery (DAD) Tutorial

6-Aug-15

On the DAD, the two AWG channels are W1 and W2, with yellow and yellow/white wires respectively. Within WaveForms, these two channels are represented as AWG1 and AWG2 (as shown in Figure 3). Note that, to correctly setup the connections, in addition to connecting W1 or W2 to your circuit, the ground terminal of your circuit must be connected to the ground wire (black color) of the DAD.

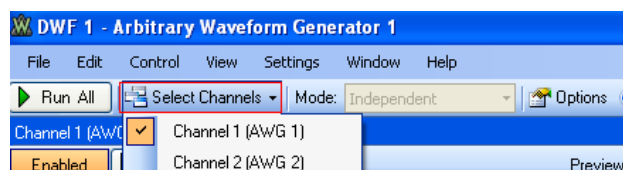


Figure 3: Channel selection in AWG

While the voltage instrument can be used to apply only fixed voltage to a circuit, AWG is more flexible and can apply varying voltages to a circuit. Within AWG in WaveForms, a channel can be selected by clicking on select channels (next to the *Run All* button) to select either AWG1 or AWG2 (as shown in Figure 4). Once a channel is selected, it can be started or stopped by clicking on *Run* button, and can be enabled or disabled by toggling the enable/disable button next to *Run AWG* button.

There are three main tabs in AWG program for controlling the generated waveform: Basic, Sweep and Advanced. The Basic simply plays

the waveform as it is specified without modification. The Sweep tab allows you to play the specified signal while varying the signals amplitude and frequency according to the pattern predefined in the Basic tab. The advanced tab allows you to vary the frequency and amplitude of the waveform according to some user-defined pattern. The advanced tab can also be used to perform frequency and amplitude modulation.

There are three methods for creating a waveform in AWG: Standard, Custom and Player.

The standard tab allows you the option of selecting from a predefined set of waveforms such as sinusoidal, square and triangular waveform. When a waveform is selected, its shape appears on the right window. Notice that *time* is on the horizontal axis while *voltage* is on the vertical axis.

Within the Basic window, modify the frequency, amplitude and/or offset of the waveform and notice how its shape changes on the right. The values of the parameters can be changed by moving the slider up or down, typing directly into the dropdown control above the slider, or by choosing a value directly in the dropdown control. The minimum and maximum

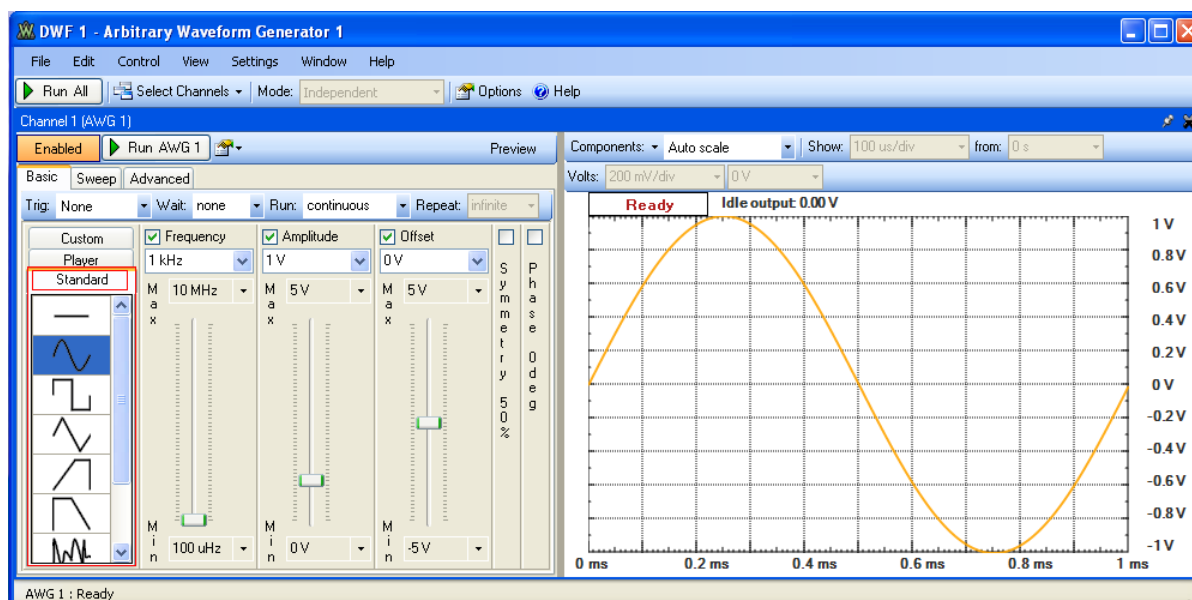


Figure 4: Standard tab – standard waveforms

Digilent Analog Discovery (DAD) Tutorial

values above and below a respective slider indicate the minimum and maximum values that are selectable for that parameter. To select a value outside of these limits, you must set the limit to a value higher than the desired value. For example, to enable a frequency in the range of 500 KHZ, set max frequency to 1 MHZ.

The Custom tab allows you to create signals according to your own specifications, as shown in Figure 5. It contains four standard controls: *Edit* for creating a waveform, *File* for opening an existing waveform, *Remove* to remove a waveform, and *Export* to export a waveform to a file.

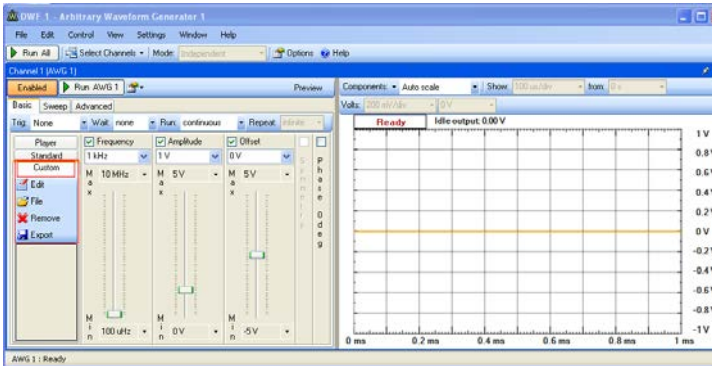


Figure 5: Custom tab – custom waveforms

Clicking on the editor opens the Editor window that allows you to design the shape of a waveform based on various parameters such as symmetry, phase, amplitude etc. (see Figure 6). Change the values of the parameters within the Function tab of the Editor and observe how the shape of the waveform changes. You can also draw the waveform directly on the shape window, or type discrete values into the sequence-value pairs on the right hand side of the window.

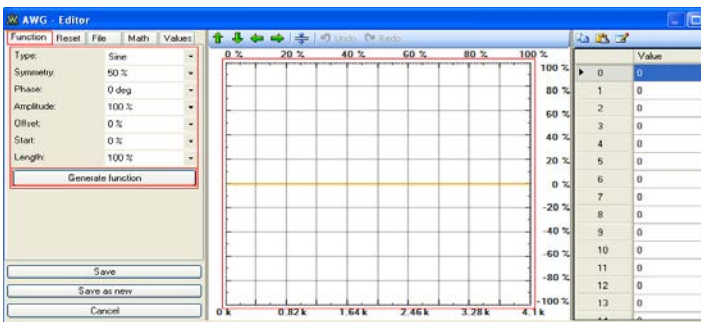


Figure 6: Editor window of custom waveform generation

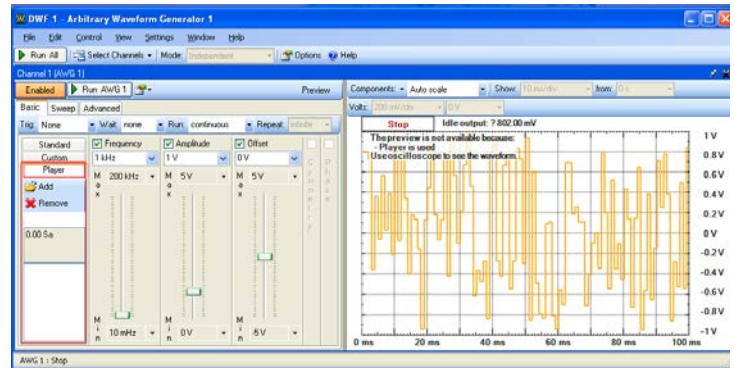


Figure 7: Player tab – playback a signal

The Player allows you to import and play a signal from a file such as a wave audio file. Once a file is added using the add button within the Player tab, its waveform is automatically displayed within the waveform window on the right.

It is important to note that, to play an audio file at the same rate at which it was recorded, the sampling rate selected for the playback must be the same sampling rate used when the audio was recorded. Observe that when a file is loaded (as in Figure 7), the filename and the sampling rate are displayed within the Player tab.

Warning: to avoid getting shocked, you should make sure that the Enable/Disable button is disabled when working on your circuit. This will cut off all voltage supply to the AWG.

Oscilloscope (Scope)

The Scope has two separate channels for reading analog inputs of up to ± 25 V on each input. It supports various sampling mode such as average, decimate, minimum and maximum. It has data measurement tools for each channel, and allows captured data to be exported to standard file formats (for example, .csv). There are also more advanced features such as FFT, XY plots and Histograms.

The Scope channels on the DAD are the terminals 1+ and 1- for channel 1, and 2+ and 2- for channel two. Note that for a given channel, the positive terminal of the scope must connect to the positive terminal of your circuit, and negative terminal of the same channel connected

Digilent Analog Discovery (DAD) Tutorial

6-Aug-15

to the negative terminal of the circuit (often the ground signal).

Once the terminals of the Scope are properly connected, clicking on *Run* within the Oscilloscope program will begin reading the voltage signals on the channels of the channels.

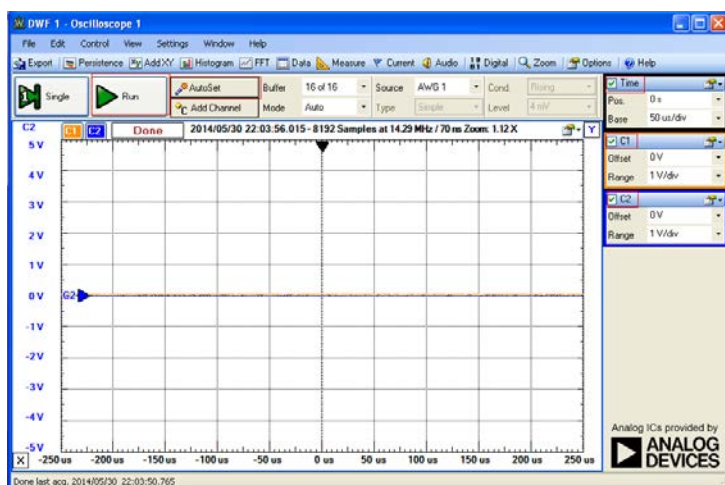


Figure 8: Scope

Clicking on the checkbox C1 and C2 will enable or disable channels 1 and 2 respectively, as shown in Figure 8. The captured data is displayed in the main window. The horizontal axis is time, while the vertical axis is the measure voltage. To adjust the display of the waveform, adjust the time control at the top right corner of the screen until a suitable display of the waveform is obtained on the screen. This functionality can also be achieved by using the *AutoSet* function next to the *Run* button.

Notice that the waveform scrolls past the display window when the time base high, or if the window shows a long time frame. To modify the display of the waveform, select view from the main menu bar and click on advanced trigger; a new menu bar will appear. By default, *Run* is set to Screen. Changing run to Shift or Repeat and observe how the display of the waveform updates to stay on the screen.

The trigger of Scope is controlled by the *condition* and *level* parameters close to the *Zoom* button. *Level* sets the voltage at which Scope is triggered, and *condition* specifies if the triggering should occur at rising or falling edge

of the specified voltage. The default settings are *0V* and *rising*, for *level* and *condition* parameters, respectively. By moving up and down the horizontal triangles around the right edge of the waveform window, the trigger level can be adjusted. Clicking and dragging the black triangle at the top adjusts the location of zero time.

Each channel is controlled by a respective channel toolbox (as shown in Figure 8). *Range* sets the voltage per division, whereas the *offset* sets the offset of the voltage display for the channel in volts. Clicking and dragging (up or down) the blue triangle on the left corner of the waveform window, or clicking and dragging the vertical axis up or down will also adjust the offset of the channel. Note that the vertical axis label is color-coded to indicate which channel is active. To make a channel active, click on its waveform, or click C1 or C2 at the top left corner of the waveform window (close to the top of the vertical axis).

Scope also allows the acquisition of a single-window worth of data. This can be achieved by clicking on *Single* next to the *Run* button. This mode is more efficient for recording a burst of voltage when a trigger condition is met, in that it retains the measured voltage burst in buffer and display window until another *Single* command is issued.

Scope provides some easy to use measurement controls. Click *View*, *Measurements*. The measurement window will open and display a number of measurement functions such as max/min, average, peak-to-peak, etc., as shown in Figure 9. From this window you can add the types of measurements you want to make and those measurements will automatically appear in the measurement window.

Exercise 1

Use one or two of the standard waveforms in the Basic tab to blink an LED on your circuit.

Connect channel 1 or channel 2 of AWG to the positive terminal of the LED, while ensuring that the negative terminal of the LED is connected to ground. The LED should light up

Digilent Analog Discovery (DAD) Tutorial

6-Aug-15

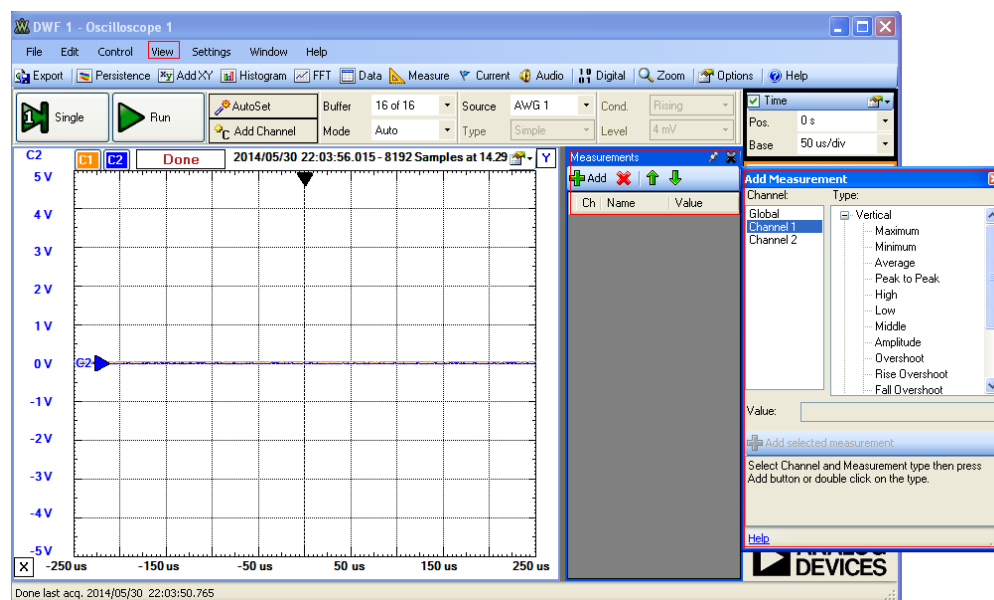


Figure 9: Measurement window of Scope (right)

only during the time when the applied voltage is high enough to supply 2V to the LED. Ensure that the corresponding channel is selected in WaveForms. Click *Run AWG*. The LED should start to blink based on the selected waveform. Change the frequency and amplitude of the waveform and observe its impact on the blinking LED.

Exercise 2

Measuring the voltage across an LED using Scope.

Connect the two terminals of either channel 1 or channel 2 of the Scope to the appropriate terminals of the LED. If the blinking of the LED is not being controlled via the 1 kHz output generated by your CPLD, you can also use the AWG (in a way similar to the previous setup) to power and control the blinking of the LED. Observe the waveform recorded in the waveform window. Can you perform any measurements?

DAD -- DIGITAL FUNCTIONS

This section describes the digital functions of the DAD, which consist of the Digital Pattern Generator for outputting digital waveforms and Logic Analyzer for displaying digital signals. Additionally, a third unit called Static I/O is also within WaveForms for simulating bit I/Os, 7-

segment slider and progress bar functionalities. The Digital Pattern Generator, Logic Analyzer and Static I/O, when used in concert with a breadboard, will help you quickly build and verify the proper operation of your design. Some of the most commonly used features are described below.

Digital Pattern Generator

The Pattern Generator has 16 signals for outputting various digital patterns such as three types of counters (binary, gray and Johnson), as well as clock and other custom defined digital signal patterns. A maximum voltage of 3.3V can be sourced from each signal line of the pattern generator output. This feature also supports import and export of data using standard file formats.

The 16 signals (labeled 0 through 15) are also shared with the logic analyzer and static I/O units. In most cases, these general-purpose signal lines are used to interface with the bus, reset, clock and other general I/O pins of your circuit. Once the connections are wired up correctly, you could then run the appropriate WaveForms program and configure the necessary parameters for reading and outputting data, or for performing simulations.

Digilent Analog Discovery (DAD) Tutorial

6-Aug-15

In this section, we will demonstrate some commonly used application scenarios of the DAD for digital output.

- One (or more) of the 16 I/O signal lines can be used to generate an output clock signal or an arbitrary output pattern. For our example, we will generate a frequency of 1 kHz (1000 Hz), and also generate constant and random outputs.
- A group of I/O signals can be utilized as a bus. For our example, we will generate a simple 8-bit wide binary counter pattern on that bus line.

We will also utilize a few of the virtual instruments available in WaveForms to visualize the patterns generated by the application scenarios described here.

As previously stated, the digital pattern generator allows you to generate both standard and custom patterns. To add signal lines in WaveForms, open Digital Pattern Generator an

d add a signal as shown in Figure 10. In this example, we have added 3 signals (DIO 0 - DIO 2) with a clock signal with frequency of 1 kHz. To do this, select *Add* (next to the green plus sign), then *Bus*. Now select DIO 0; hold down the shift key while selecting DIO 1 and DIO 2. Now select the right green arrow. For format, select *Hexadecimal* and for Endianness select *LSB*. Notice that when a signal is added initially, it is represented as *Constant* in the *Type* column. Click on the *Type* control to select *Binary Counter*. (If you were generating a single clock signal, you would select *Clock*.) Now select *Edit* and then *Edit Parameters of "Bus 0."* Select the desired frequency of changing values. Note that a 1 kHz frequency will make a 0.5 kHz (1 kHz / 2) signal on DIO 0. Change the *Value* of *Frequency* to 1 kHz as shown in Figure 10a. Close the *Edit "Bus 0"* window by selecting the × on the top right of this window. Figure 10a shows what this should. Figure 10b shows the result pattern

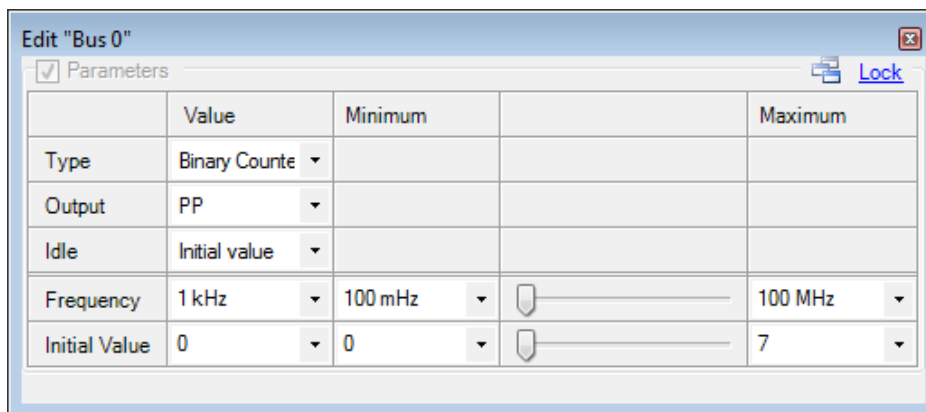


Figure 10a: Selecting the frequency for the Digital Pattern Generator

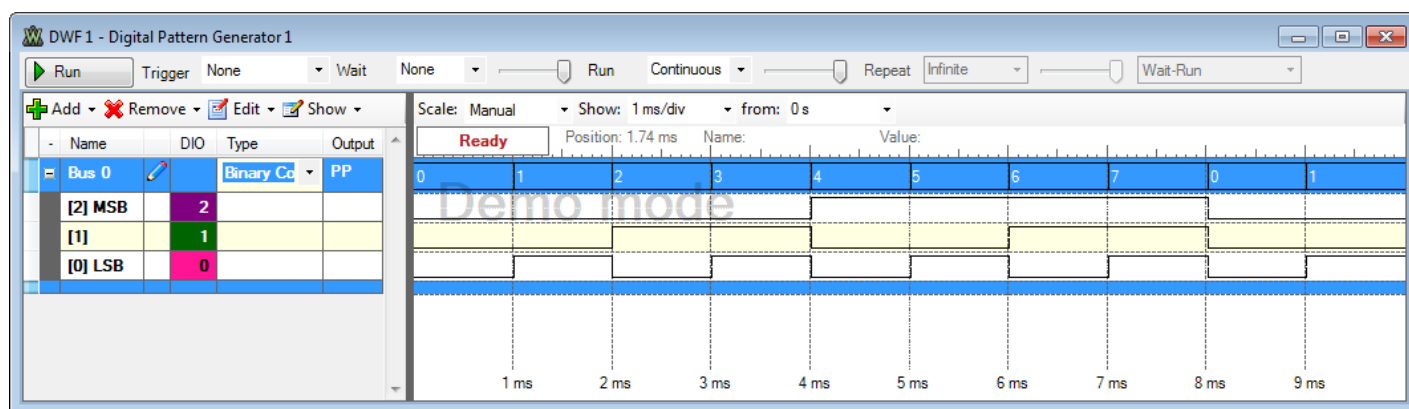


Figure 10b: Digital pattern changing at 1 kHz, generated using the Digital Pattern Generator.

Digilent Analog Discovery (DAD) Tutorial

6-Aug-15

generated (after I changed the *Scale* to *1 ms/div*).

If you need to change the parameters of the signal (such as frequency, phase, duty cycle, etc.), click on the pencil icon and the *Pin Edit* window shown above will be displayed with the name of the signal being edited shown in the title windows title bar. You can also do this by right clicking the signal and selecting *Edit Parameters of "DIO x"*.

To begin generating signal, click on the *Run* button. The pattern of the just described signal will show in window on the right. Note that, depending on the signal parameters chosen, you may have to adjust the scale of the signal display in order to visualize the full pattern. In this example, the display has been changed to *manual*, and a *1ms per division* scale as shown in Figure 10c. A *division* is the spacing between the vertical lines on the display.

To run with a circuit, connect the corresponding I/O pins of the DAD (in this case pins 0, 1 and 2) to LED circuits on your design. Edit the parameters of the signal (i.e., change the frequency to something that is slow enough to see, like 1 Hz) by changing its frequency. Observe the effect on the LED.

You can also generate clock signals of a desired frequency. Figure 10c shows a *Clock* with a 2 kHz frequency on DIO 3.

You can visualize the output of the generated clock signal on the Static I/O virtual instrument as shown in Figure 11. The corresponding LED on the instrument (in the top figure, select the *Config* value of *BIT I/O*) should blink at the frequency set in Digital Pattern Generator; to generate the bottom, select the *Config* value of figure has a *Progress Bar*.

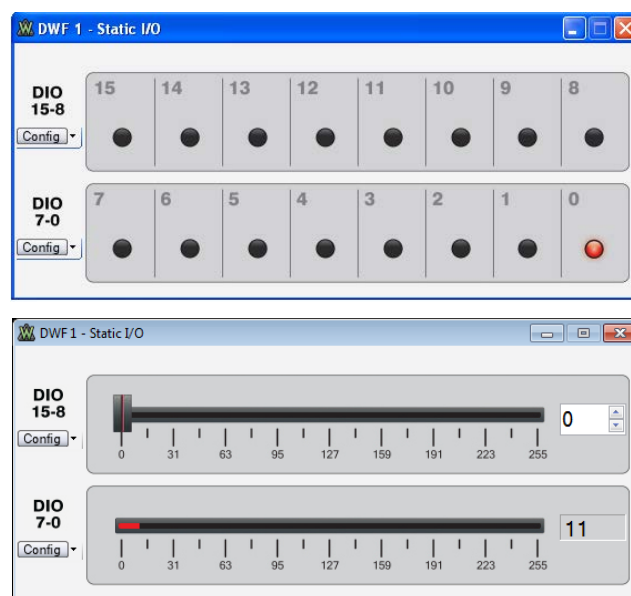


Figure 11: Signal visualized with virtual instrument

In addition to generating pattern and clock signals with fixed frequencies, the digital pattern generator can also generate constant, random and custom logic patterns.

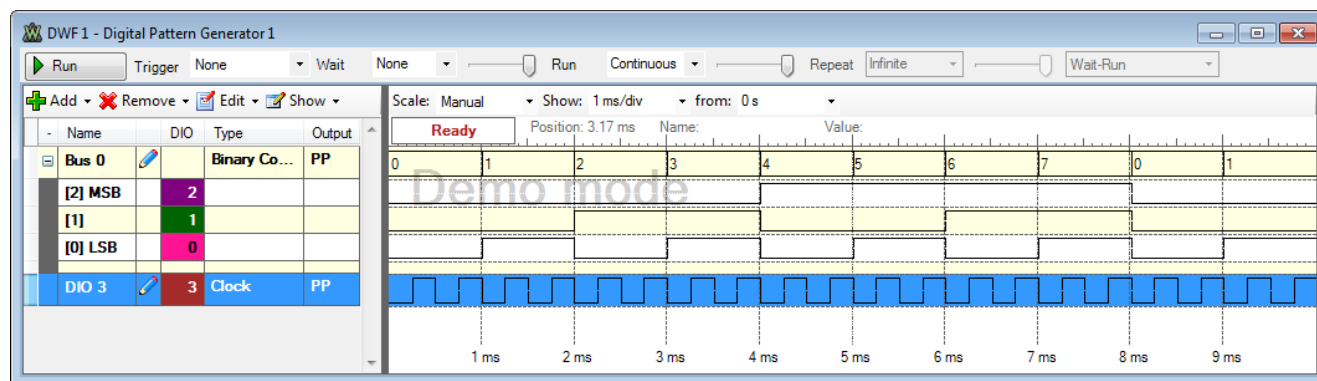


Figure 10c: 2 kHz clock signal on DIO 3.

Digilent Analog Discovery (DAD) Tutorial

6-Aug-15

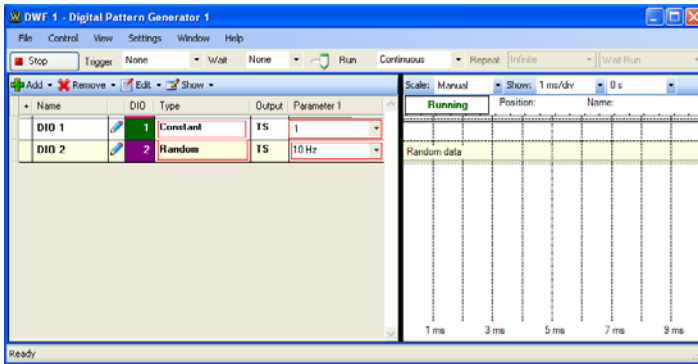


Figure 12: Single-bit constant and random data generation

Figure 12 shows two signals: one defined as *Constant*, and the other as *Random*. The constant type can be used to generate instantaneous single bit output. The random type randomly generates a single 1-bit value.

The Parameter field displays the parameter(s) of each signal. To display this field, click on the *Show Icon*, then *Parameter 1*. In this example, *Parameter 1* of signal *DIO 1* can be changed from 1 to 0, and vice versa. This will flip the generated signal on that line. For *DIO 2*, *Parameter 1* represents the frequency of the randomly generated data.

The pattern generator's I/O lines can be used as a *bus*, i.e., a collection of I/O. Click on the add new signal button and select *bus*. This will allow you to add a group of signals to be used for pattern generation. In this case, Figure 13 shows that we have added an 8-bit bus (signal

lines 8 to 15). The pattern on this bus is a Binary Counter. You can choose from other types of pattern in the *Type* dropdown menu.

When creating a bus, you can also choose the format of its data representation. In this example, the bus data is displayed as binary. You can always modify and select other formats, such as decimal or hexadecimal from the *Edit Properties of "Bus x"* when you right-click on the bus signal.

You can visualize the output binary counter pattern on the Static I/O virtual instrument, as shown in Figure 14. In the Static I/O instrument, configure the corresponding signal lines (in this example, signal lines 8 to 15) as *Progress Bar*. To do this, simply click on *Config*. If the pattern generator is running, the progress bar should continuously count from 0 to 255.

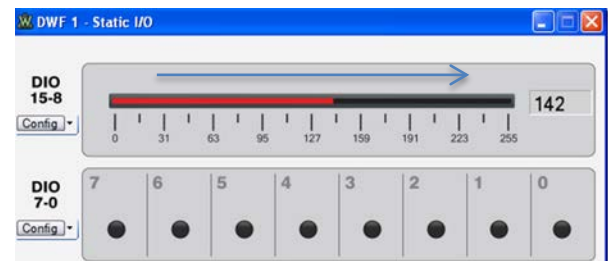


Figure 14: Using a virtual instrument to visualize an 8-bit binary counter pattern

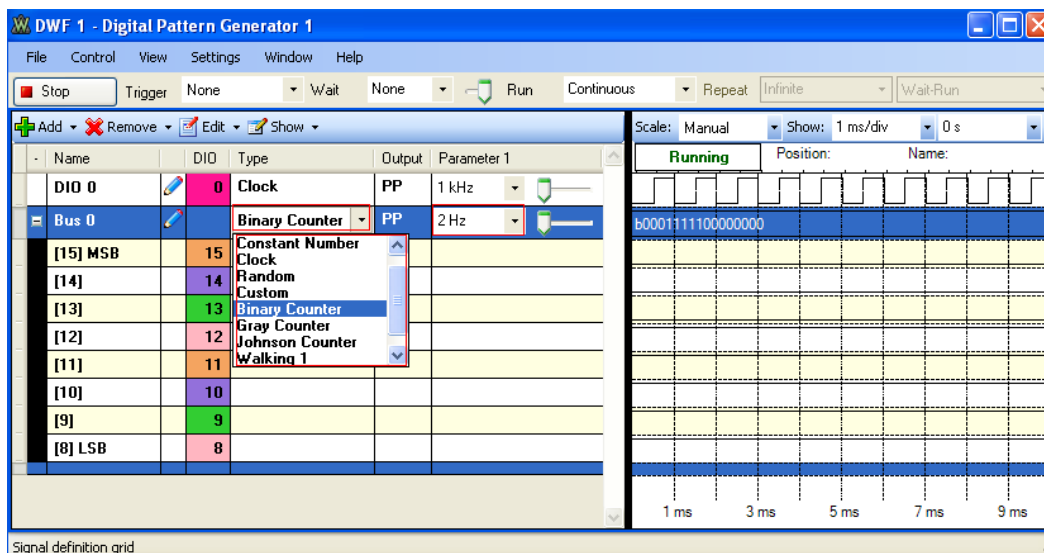


Figure 13: Generating a digital pattern on a bus signal

Digilent Analog Discovery (DAD) Tutorial

6-Aug-15

Logic Analyzer

The Logic Analyzer has 16 signals for recording and analyzing digital patterns. The signal lines are shared with the Pattern Generator and Static I/O units. The physical interface to the logic analyzer unit has LVCMOS (Low-Level Complementary Metal-Oxide Semiconductor) logic levels. LVCMOS is a low-level voltage class of CMOS which provides support for logic levels that operate at low-voltages.

Besides being able to analyze distinct signal logic states, the Logic Analyzer can also serve as an interpreter for commonly used communication protocols (such as SPI, I2C, UART, parallel bus), allowing a circuit designer to quickly troubleshoot communication protocols at the physical layer.

Recall that, in the Digital Pattern Generator subsection above, we demonstrated the use of Pattern Generator in a number of use cases: clock generation, pattern generation on an 8-bit bus, and constant and random pattern generation on a single signal line. In this section, we will make use of those setups, and additionally configure a few parameters in the Logic Analyzer. The Logic Analyzer should display any pattern generated in Pattern Generator when both programs are running. EEL 3701 students will use this to get a waveform that should match Quartus' voltage simulation.

Click *Add, Signals* and look at Figure 15. Select the signal used for the clock in Pattern Generator (in this case *DIO 0*). Also, add a bus signal – the same bus signal used in Pattern Generator (in this case, *DIO15 – DIO8*). Make sure that Pattern Generator is running. Click *Run*; you should see the pattern displayed in the window.

Note that you may need to adjust the scale of the signal display in order to visualize the full pattern. In this example, the display has been changed to *Base: 1s/div* scale, and *Run: Shift* as shown in the Figure 15. A *division* is the unit spacing between the vertical running lines on the display.

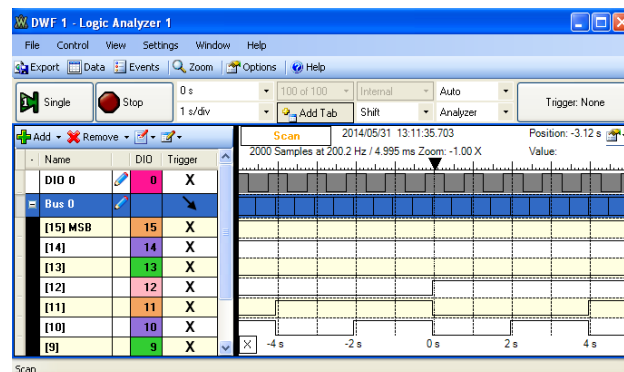


Figure 15: Logic Analyzer – displaying clock and 8-bit bus signals

As with the clock and bus signal recording, we can similarly use Logic Analyzer to record and analyze the constant and random data generated by Pattern Generator. Add the corresponding signals for the *constant* and *random* signal lines (in this case, *DIO1* and *DIO2* respectively) as shown in Figure 16. Click *Run* and observe the behavior.

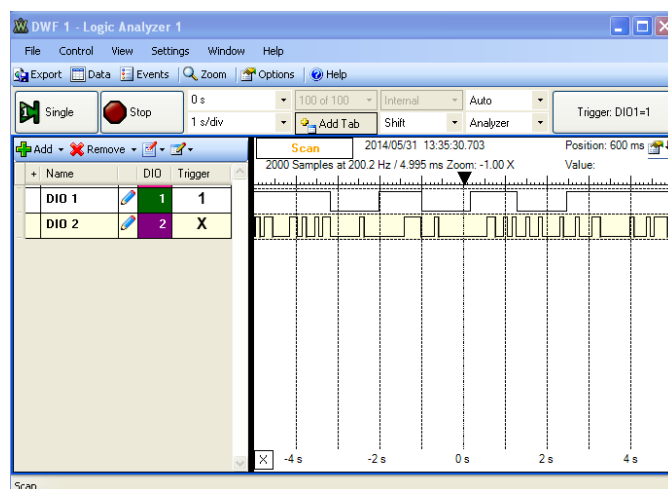


Figure 16: Logic Analyzer – displaying toggled and random data input

While the random signal is randomly generated, the constant signal can be changed. You can toggle the value of the constant signal in Pattern Generator and watch its corresponding effect in Logic Analyzer.

Just like in the Scope instrument for analog signals, Logic Analyzer also has a measurement tool for measuring points of interest on signals (see Figure 17). To measure the difference in time between two points on a signal, drag the X

Digilent Analog Discovery (DAD) Tutorial

6-Aug-15

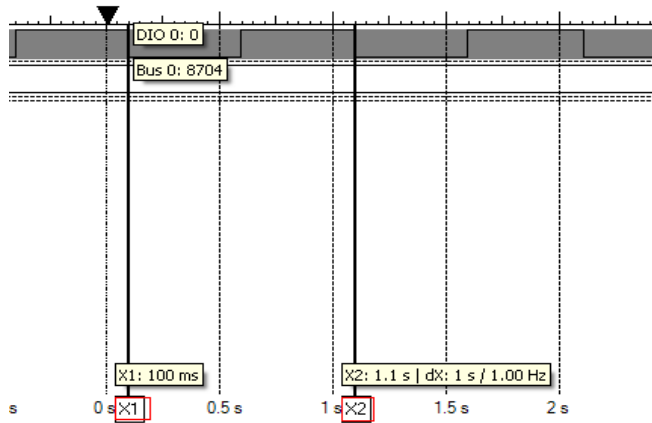


Figure 17: Simple measurement in Logic Analyzer

icon at the bottom left corner in the display to a point of interest. That point becomes *X1*. Drag another *X* to a second point of interest. That point becomes *X2*. Place the cursor on either *X1* or *X2* markers and the difference of the two points will be displayed.

If you need to save your recorded data to a file, Logic Analyzer allows you to export captured data to a standard file format (.csv) from whence the data can be further analyzed or processed in applications such as MATLAB.

Exercise 3

Use the pattern generator to blink an LED on your circuit

Connect an I/O signal to the positive terminal of the LED, while ensuring that the negative terminal of the LED is connected to ground. Note the I/O signal and configure Pattern Generator to output a *clock* pattern on that signal line. Run the generator and observe its effect on the LED. For the same frequency and duty cycle as with AWG experiment in Exercise 1, what differences do you observe with the blinking of the LED in this case? Can you explain the phenomenon behind the difference in the blinking of the LED in both experiments?

Exercise 4

Analyzing your data bus using the logic analyzer

Connect the 8 LSBs of your data bus to 8 consecutive I/Os on the DAD. Note their numbers and configure them appropriately in Logic Analyzer WaveForms program. Optionally, connect the clock and reset of your circuit to any other two free signal lines on the DAD. Have your CPLD program, write an 8-bit value on the data bus. Now, run the analyzer and observe the output on the bus line. (Remember that the bus in Logic Analyzer can be made to display values in several formats including binary, decimal and hexadecimal). Is the data on the bus line in Logic Analyzer same as the output data written in your CPLD program? What measurements can you make on the data bus signal (with respect to the clock signal)?