



DIGILENT[®]

A National Instruments Company

Hands-On Experimentation using Digilent Analog Discovery 2

Complete analog & digital circuits in or out of the lab

Mircea Dabacan

Technical University of Cluj-Napoca, Romania

Mircea.Dabacan@ael.utcluj.ro

Digilent RO, Cluj-Napoca, Romania

Mircea.Dabacan@digilent.ro

www.digilentinc.com

Paris 2017

Analog Discovery 2

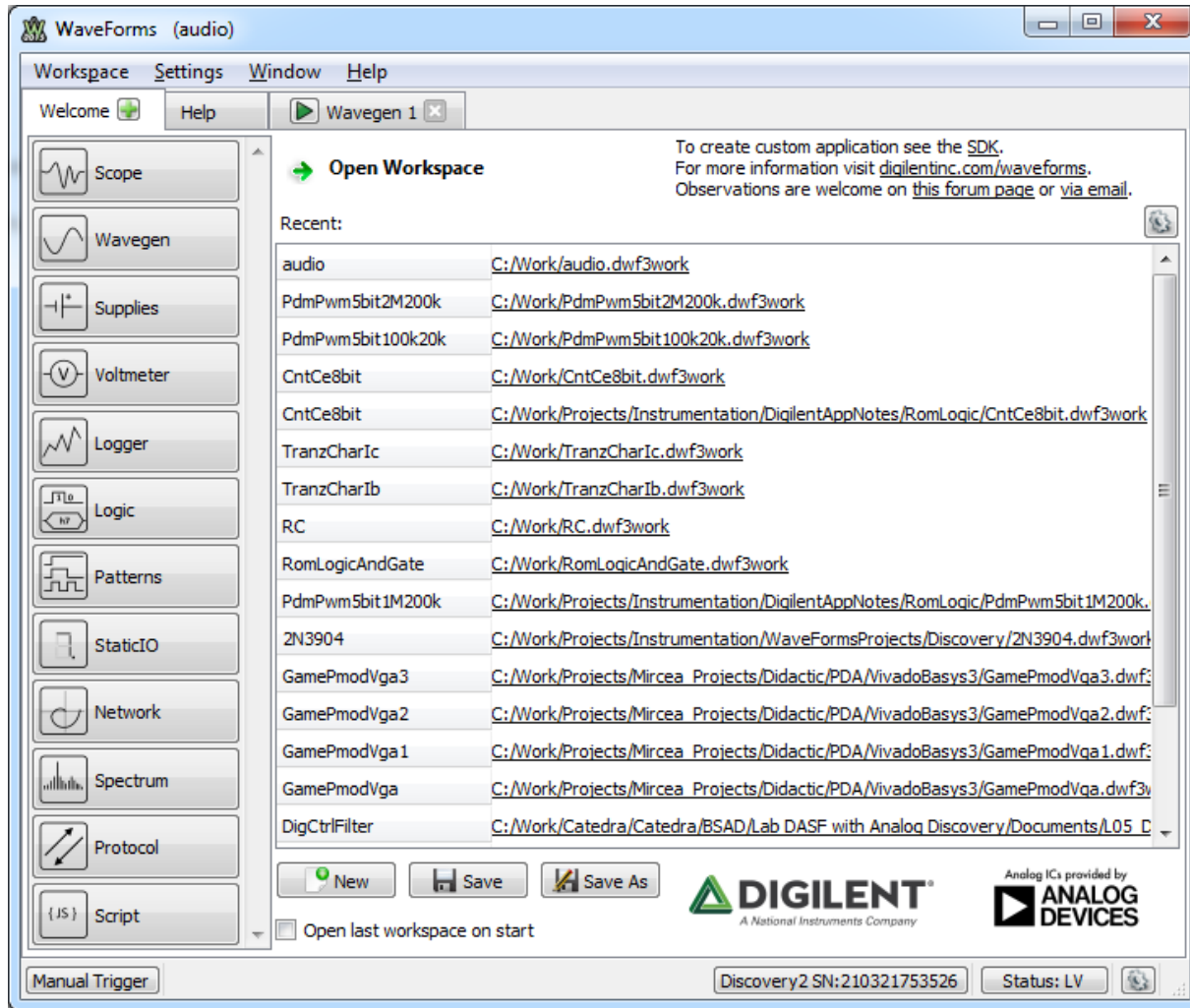
Primary HW resources:

- Two-channel scope (1M Ω , \pm 25V, differential, 14-bit, 100MS/s, 30MHz+ with the BNC Adapter)
- Two-channel AWG (\pm 5V, 14-bit, 100MS/s, 12MHz+ with the BNC Adapter)
- Stereo audio amplifier to drive external headphones with AWG signals
- 16-ch digital logic analyzer/pattern generator/static IO (3.3V CMOS, 100MS/s)
- Two input/output digital triggers for linking multiple instruments (3.3V CMOS)
- Two programmable power supplies (0...+5V , 0...-5V):
 - 250mW max for each supply or 500mW total when powered through USB
 - 2.1W max for each supply when powered by an auxiliary supply
 - 700mA maximum current for each supply

HW resources re-usage (by WaveForms):

- Single channel voltmeter (AC, DC, \pm 25V)
- Network analyzer – Bode, Nyquist, Nichols diagrams of a circuit; 1Hz to 10MHz
- Spectrum Analyzer –spectrum and measurements (noise, SFDR, SNR, THD, etc.)
- Digital Bus Analyzers (SPI, I²C, UART, Parallel)

WaveForms 2015



Cross platform

- Windows
- Linux
- OS

Supports Digilent instrumentation HW

- Electronics Explorer
- Analog Discovery
- Analog Discovery 2
- Digital Discovery

Programmer support

- SDK
- LabVIEW support

Optimal use of HW resources

Waveforms 2015

	AWG	Scope	Network Analyzer	Spectrum Analyzer	Pattern Generator	Logic Analyzer	Static I/O
LED I/V characteristics	simple	Time					
		Math					
		XY					
		Trigger					
Transistor characteristics	custom	Units					
	sync	offset =div					
RC Low Pass Filter		Measure	Bode				
		Export	Nyquist				
			Nichols				
Audio	Modulation						
	Sweep						
	Play						
3-bit And Gate					ROM logic - comb	Bus	SW
					Counter	Signal	LED
					Clock		
8-bit counter with CE					ROM logic - FSM		
PDM/PWM modulator				Spectrum	Custom		
				Measure			

LED I/V characteristics

Scope, Channel 2

- Offset: 0V
- Range: 1V/div
- Invisible

AWG, channel 1

- Mode: Simple
- Shape: Triangle
- Frequency: 1kHz
- Amplitude: 5V
- Offset: 0V
- Symmetry 50%

Scope, Math 1

- Mode: Custom
- C2/100
- Units: A
- Offset: 0A
- Range: 5mA/div

Scope, Trigger

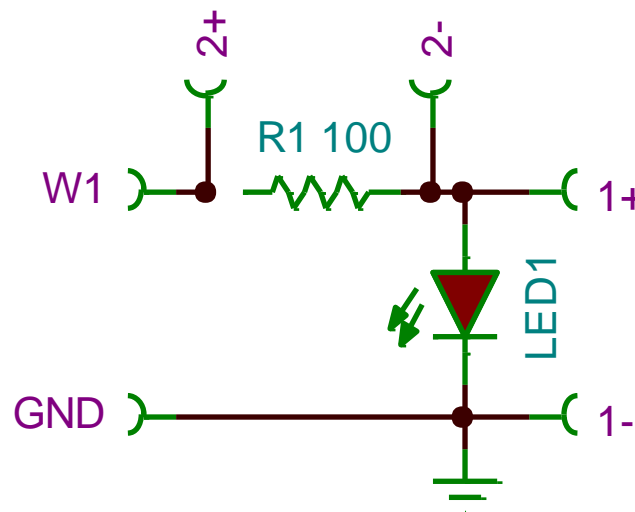
- Mode: Auto
- Source: Channel 1
- Cond: Rising
- Level: 0V

Scope, Time Base

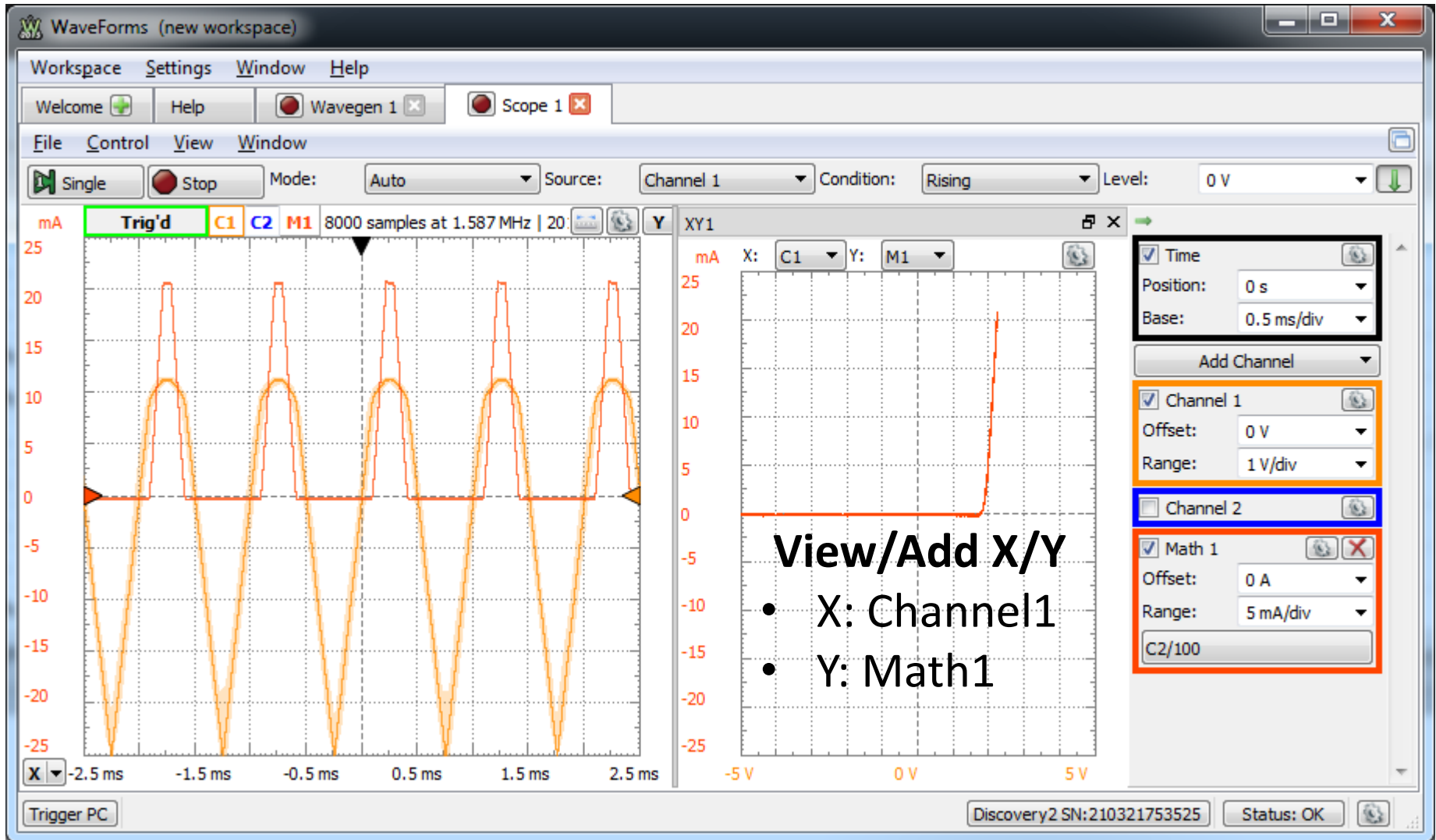
- Position: 0s
- Base: 0.5ms/div

Scope, Channel 1

- Offset: 0V
- Range: 1V/div



LED I/V characteristics



LED I/V characteristics

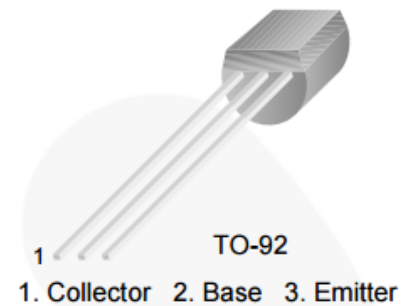
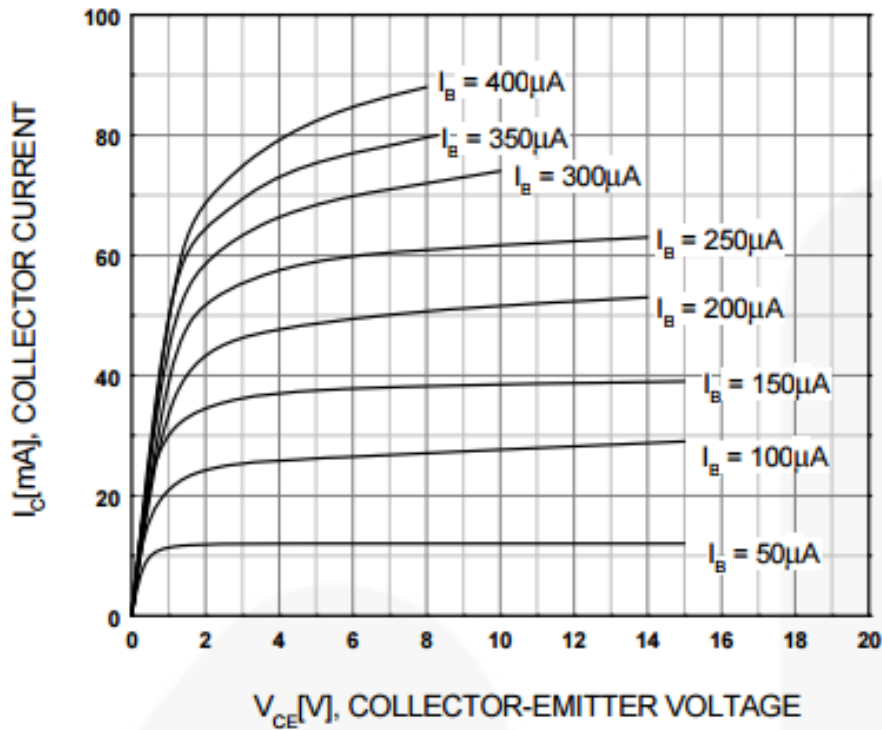
More easy experiments...

- Change stimulus shape to sinus
 - Does the X/Y view change? Why?
- Change stimulus shape to rectangle
- Play with frequency
 - you might first want to set the Min/Max limits for Frequency to 100mHz/1kHz.
 - Understand the human eye limitation in stimulus pulse frequency
 - Understand the basics of multiplexed LEDs display
- Play with symmetry (0 to 100%).
 - Understand PWM modulation of LEDs equivalent luminous intensity.

Transistor characteristics



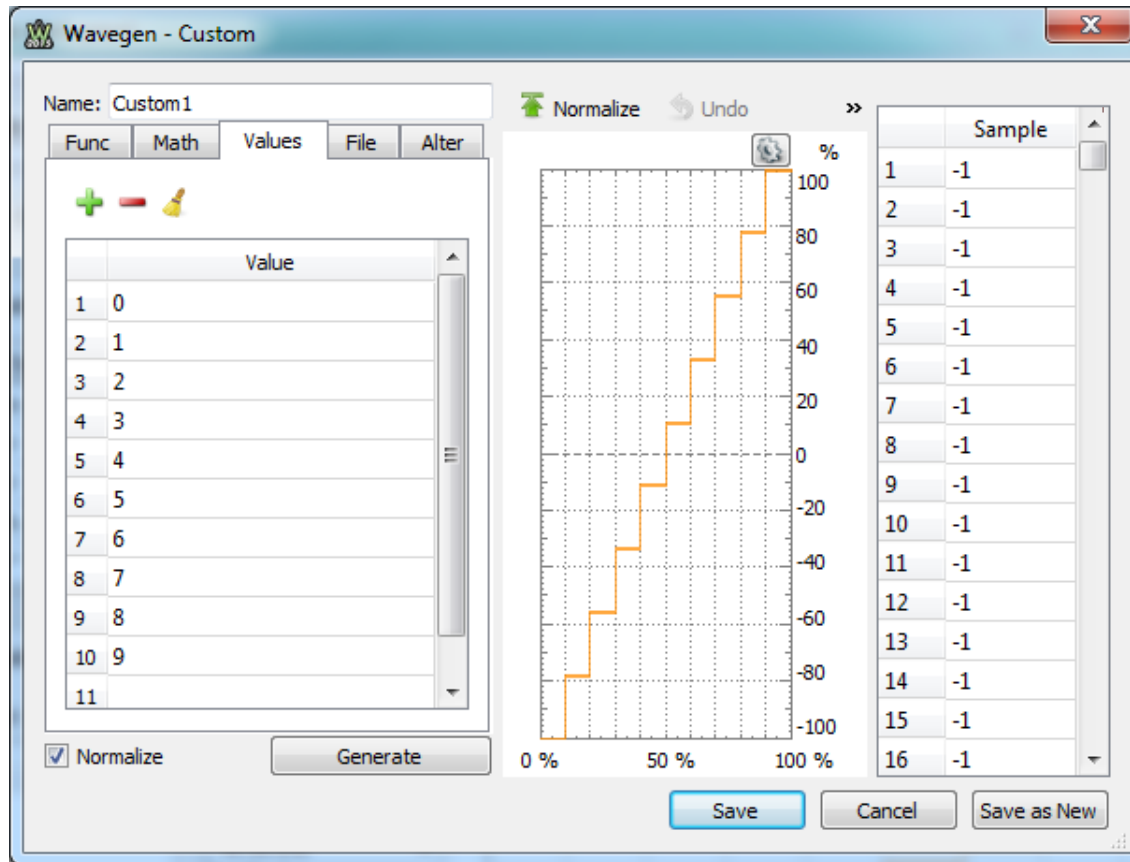
BC546 / BC547 / BC548 / BC549 / BC550 NPN Epitaxial Silicon Transistor



Transistor characteristics

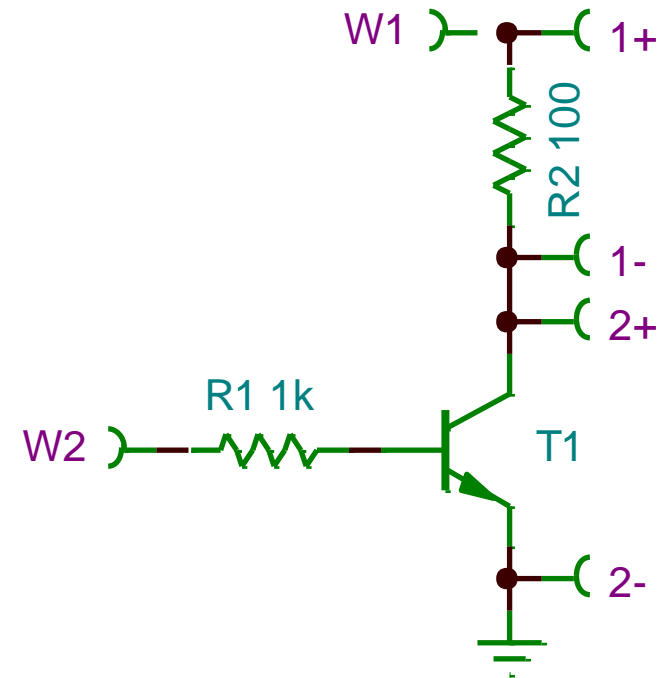
AWG, channel 2

- Mode: Custom
- Edit/Values:0,1,2,...9
- Normalize



AWG, channel 1

- Mode: Simple
- Shape: Triangle
- Frequency: 50Hz
- Amplitude: 2.5V
- Offset: 2.5V
- Phase: 270°



Transistor characteristics

AWG, channel 1

- Mode: Simple
- Shape: Triangle
- Frequency: 50Hz
- Amplitude: 2.5V
- Offset: 2.5V
- Phase: 270°

AWG, channel 2

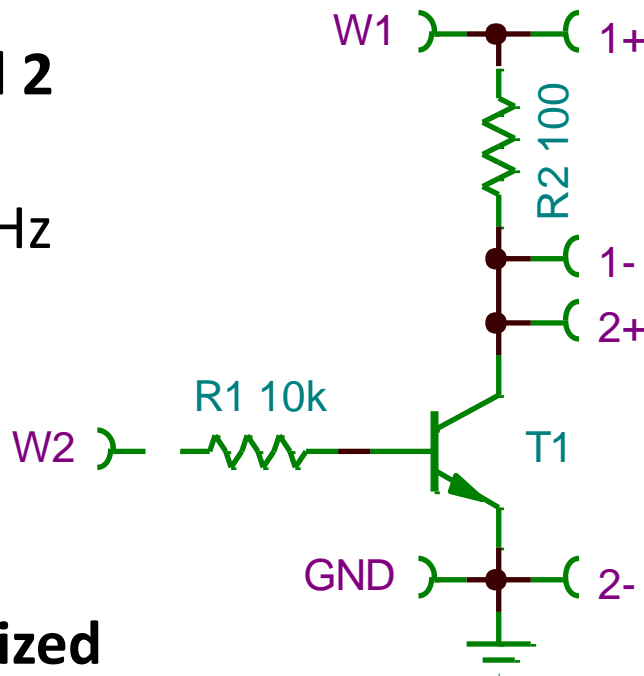
- Mode: Custom
- Frequency: 10Hz
- Amplitude: 1V
- Offset: 1.6V
- Phase: 0°

Scope, Channel 1

- Offset: -2.5V
- Range: 500mV/div

Scope, Channel 2

- Offset: -2.5V
- Range: 500mV/div



AWG, Synchronized

Scope, Trigger

- Mode: Auto
- Source: Channel 2
- Cond: Falling
- Level: 4.9V

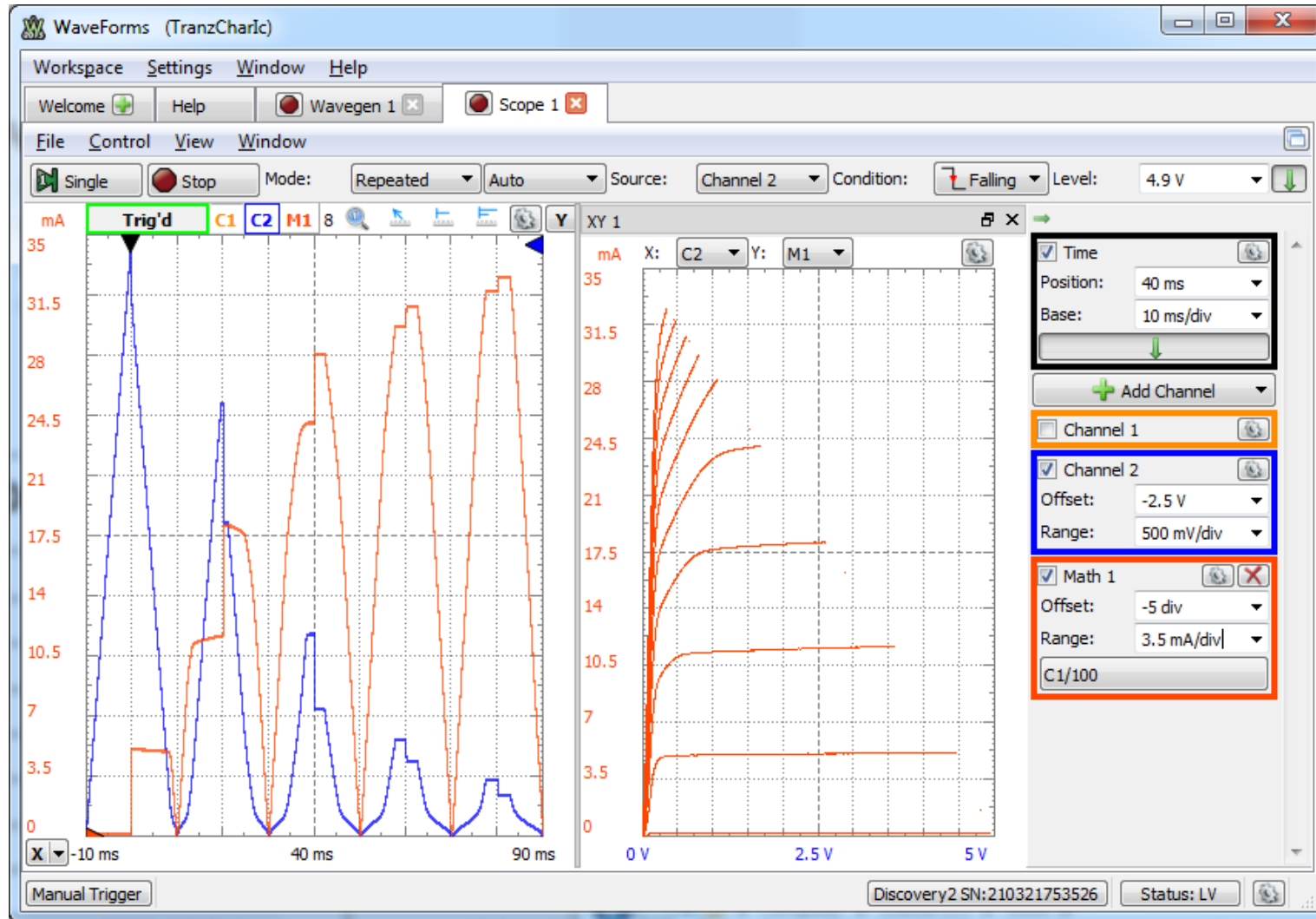
Scope, Time Base

- Position: 40ms
- Base: 10ms/div

Scope, Math 1

- Add Ch/Simple
- C1/100
- Settings: Offset as divisions
- Settings: Units = A
- Offset: -5div
- Range: 3mA/div

Transistor characteristics $I_C(V_{CE})$; I_B parameter



Scope

- View
- Add X/Y
- X: C2
- Y: M1

Transistor characteristics; I_B values

Scope, Channel1

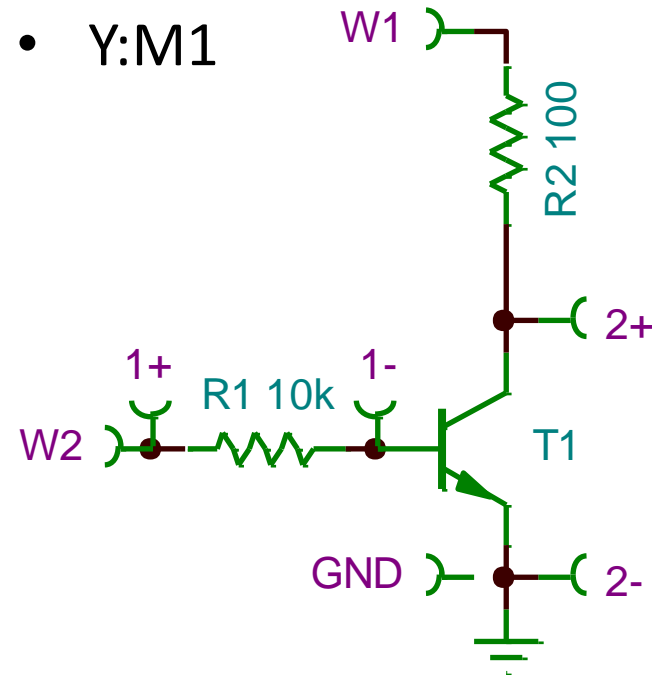
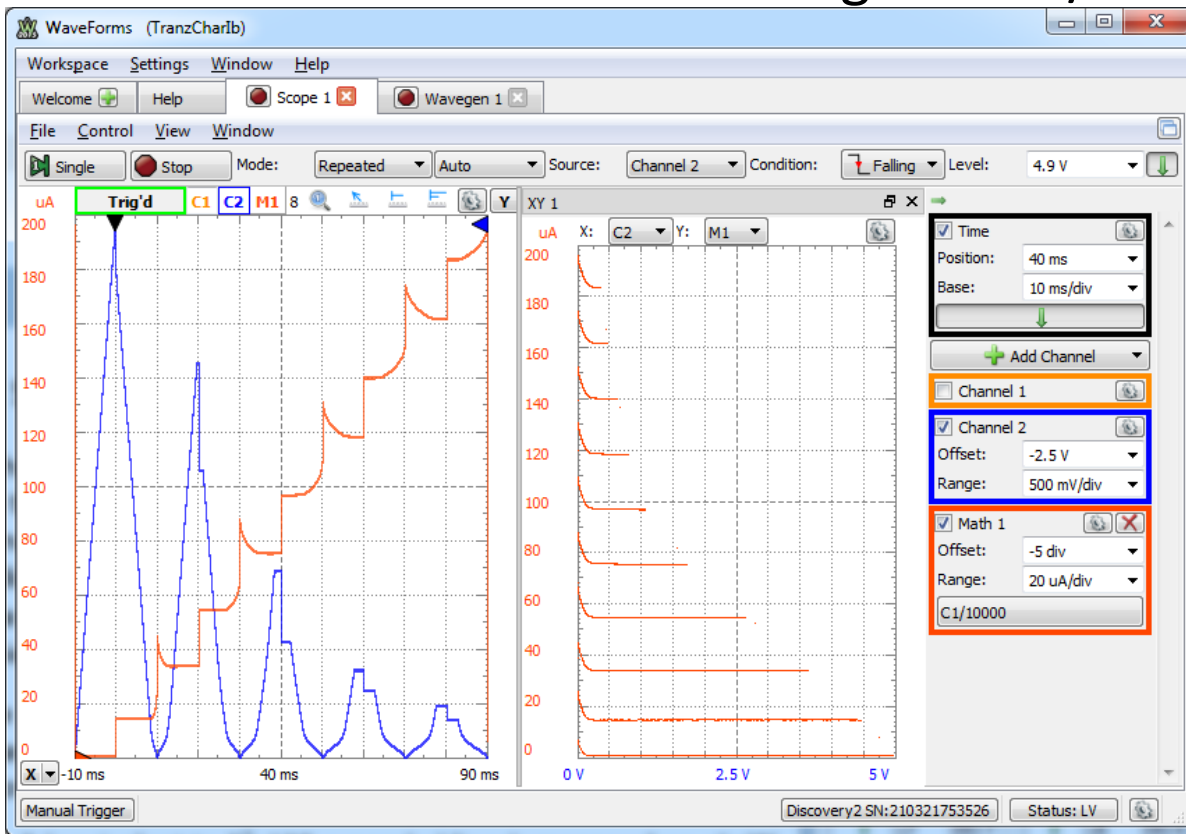
- Offset: -5div
- Range: 200mV/div

Scope, Math 1

- Add Ch/Custom
- C1/10000
- Range: 20uA/div

Scope

- View
- Add X/Y
- X: C2
- Y:M1



RC Low Pass Filter

Time domain - step response

AWG, channel 1

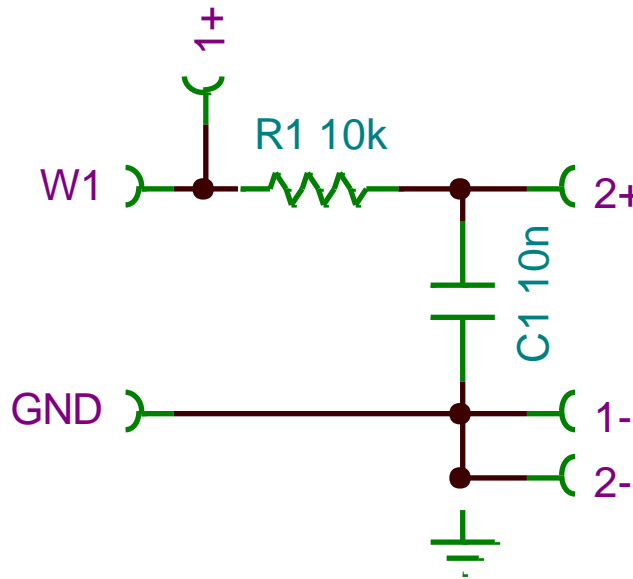
- Mode: Simple
- Shape: Square
- Frequency: 50Hz
- Amplitude: 2.5V
- Offset: 2.5V

Scope, Channel 1

- Offset: -2V
- Range: 1V/div

Scope, Channel 2

- Offset: -2V
- Range: 1V/div



Scope, Trigger

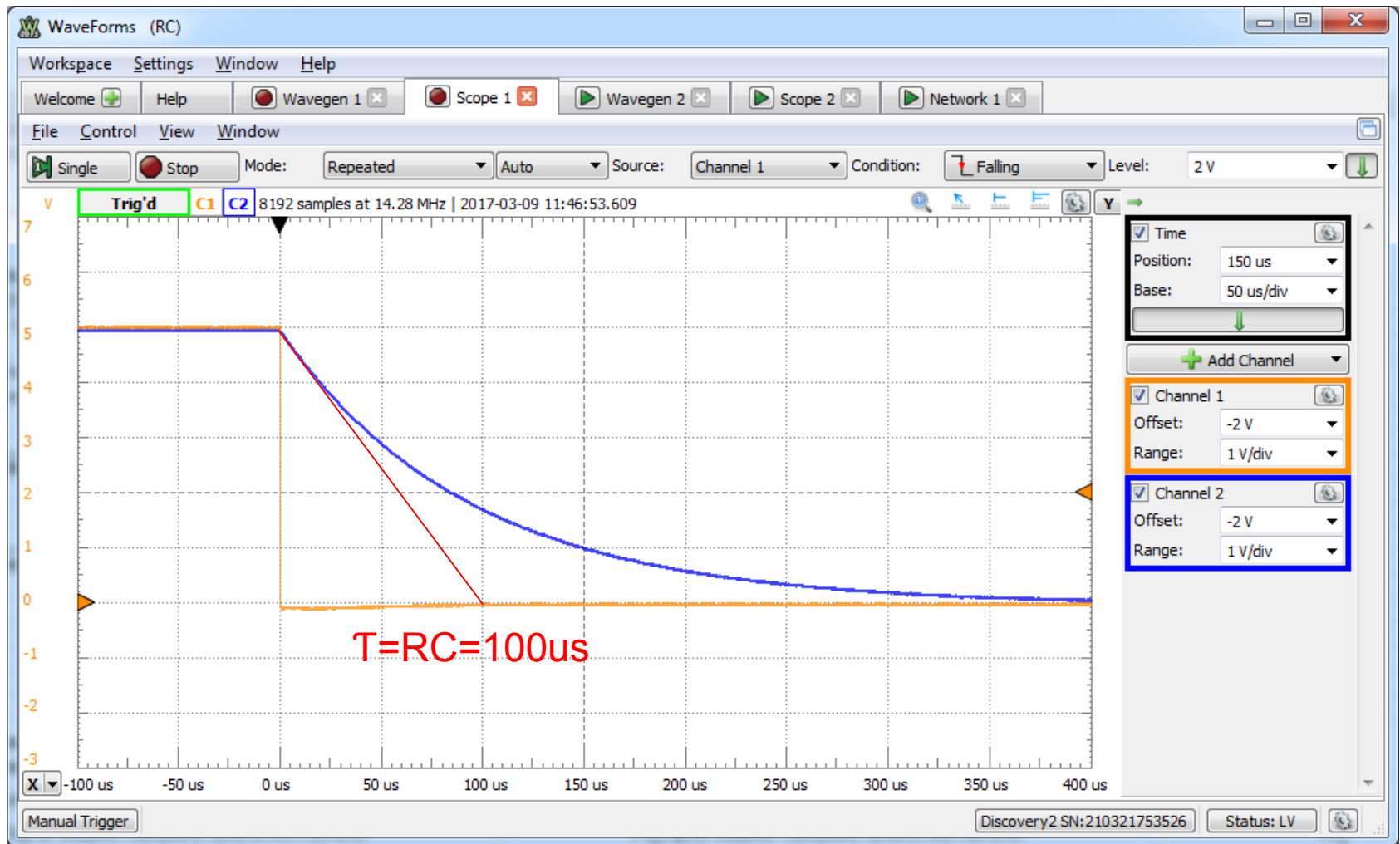
- Mode: Auto
- Source: Channel 1
- Cond: Falling
- Level: 2V

Scope, Time Base

- Position: 2ms
- Base: 500us/div

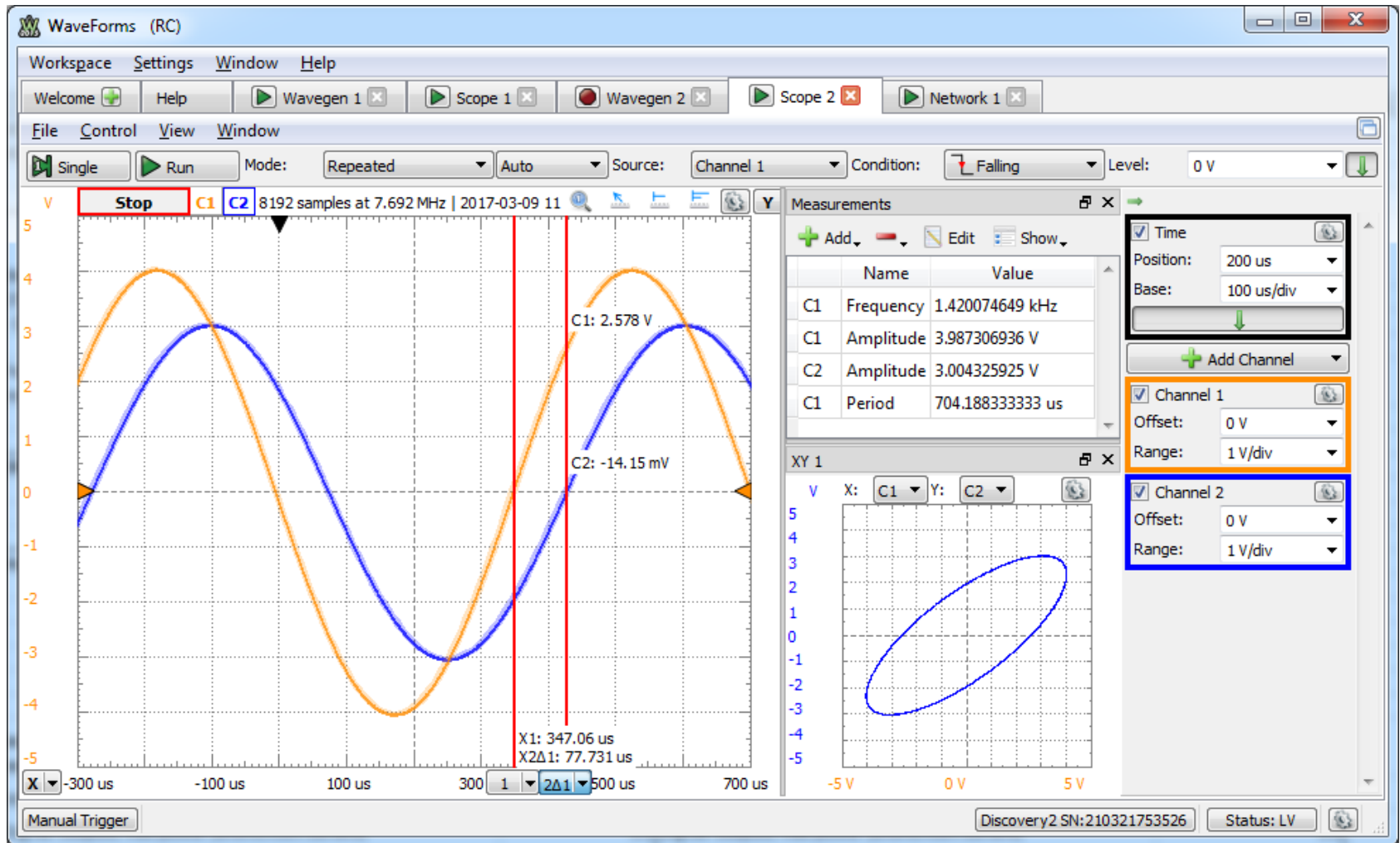
RC Low Pass Filter

Time domain - step response



RC Low Pass Filter

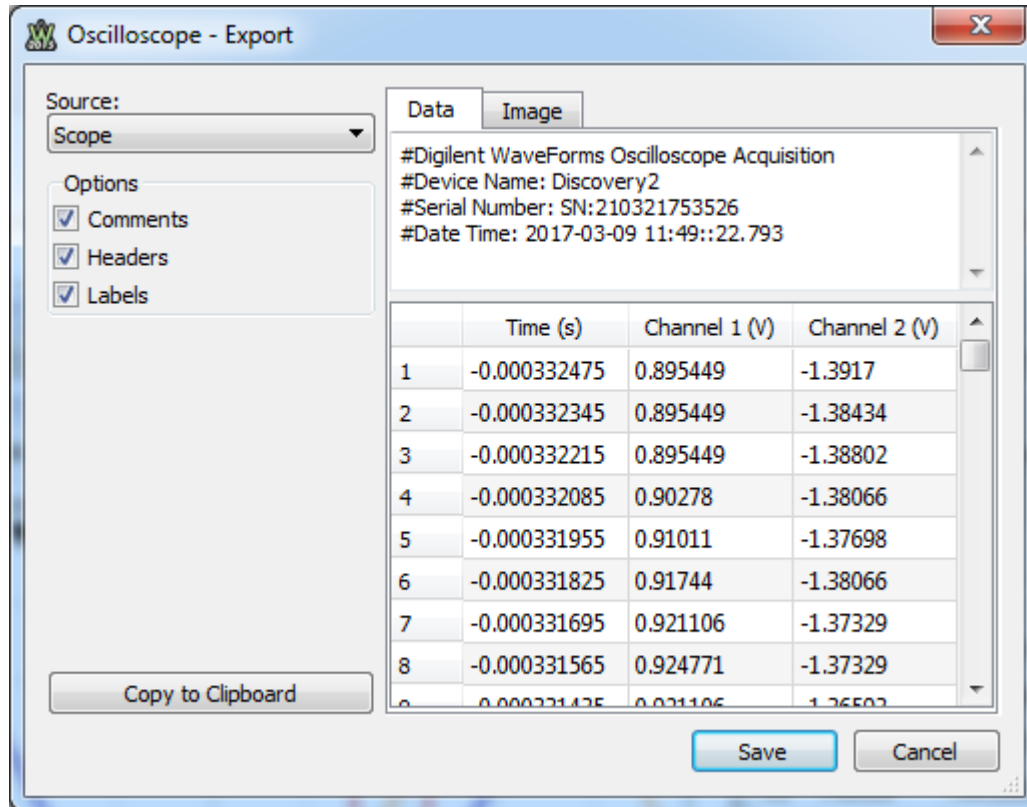
Time domain - sinus response



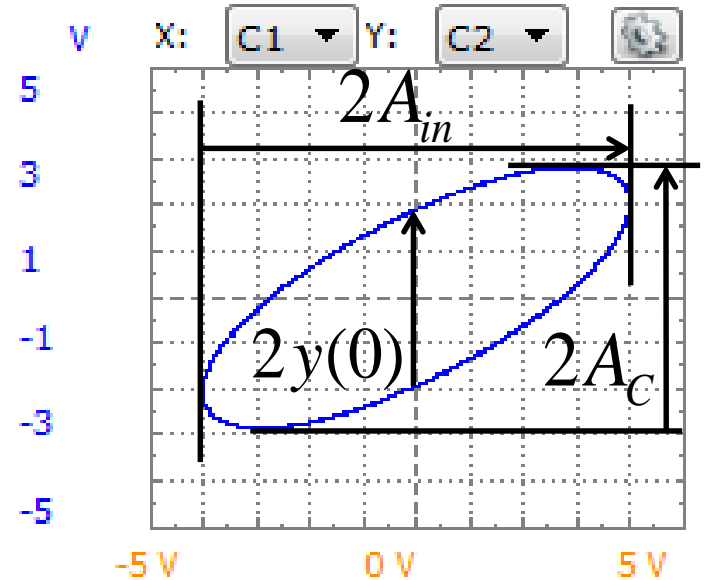
RC Low Pass Filter

Time domain - sinus response

Export – Data – Scope



Export – Image – XY1



$$x(t) = V_{in}(t) = A_{in} \sin(2\pi ft)$$

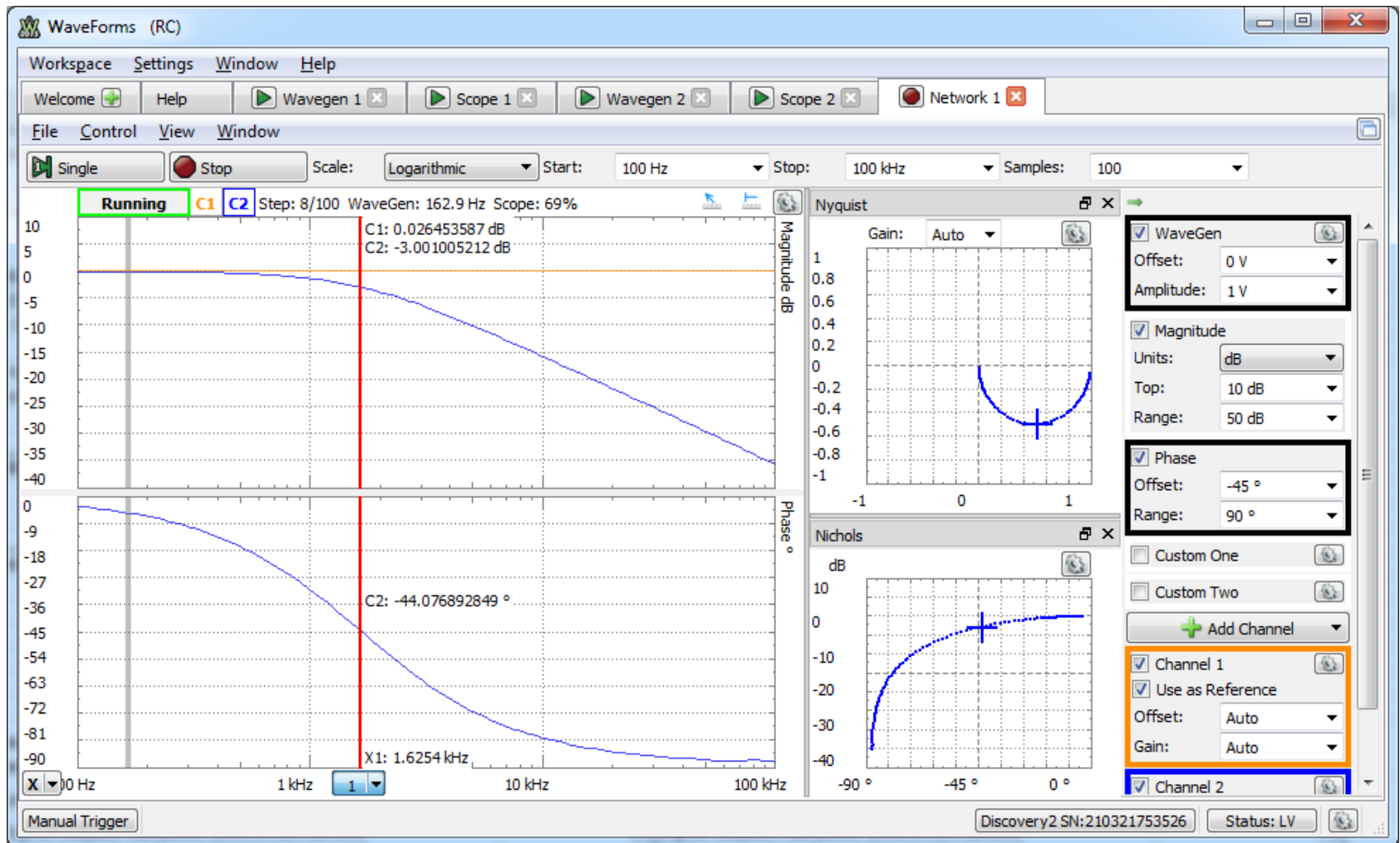
$$y(t) = V_C(t) = A_C \sin(2\pi ft + \varphi)$$

$$y(0) = V_C(0) = A_C \sin(\varphi)$$

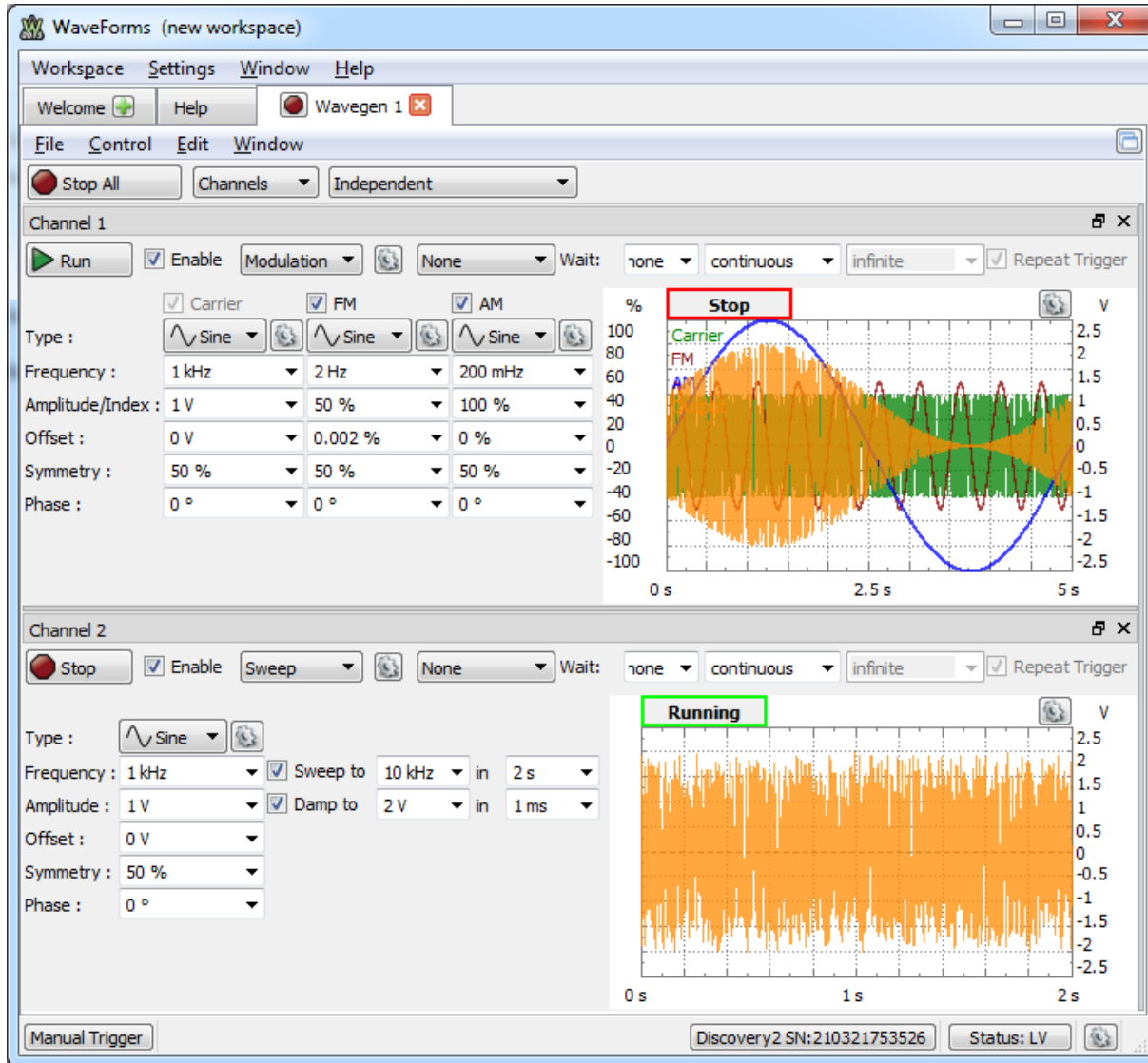
$$\varphi = \arcsin\left(\frac{y(0)}{A_C}\right) = \arcsin\left(\frac{2y(0)}{2A_C}\right)$$

RC Low Pass Filter

Frequency domain – Bode, Nyquist, Nichols diagrams

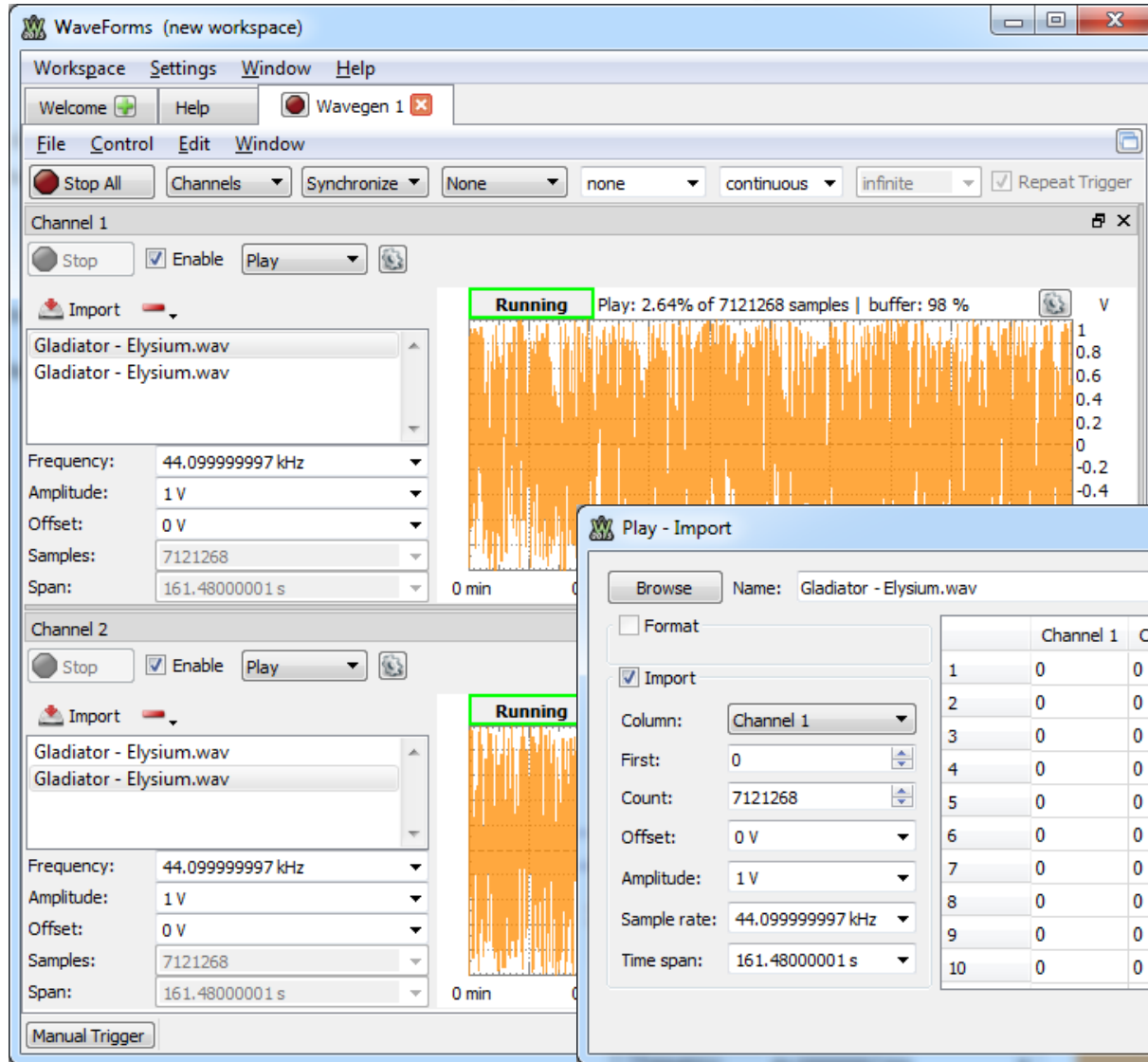


Audio - Wavegen - modulation, sweep

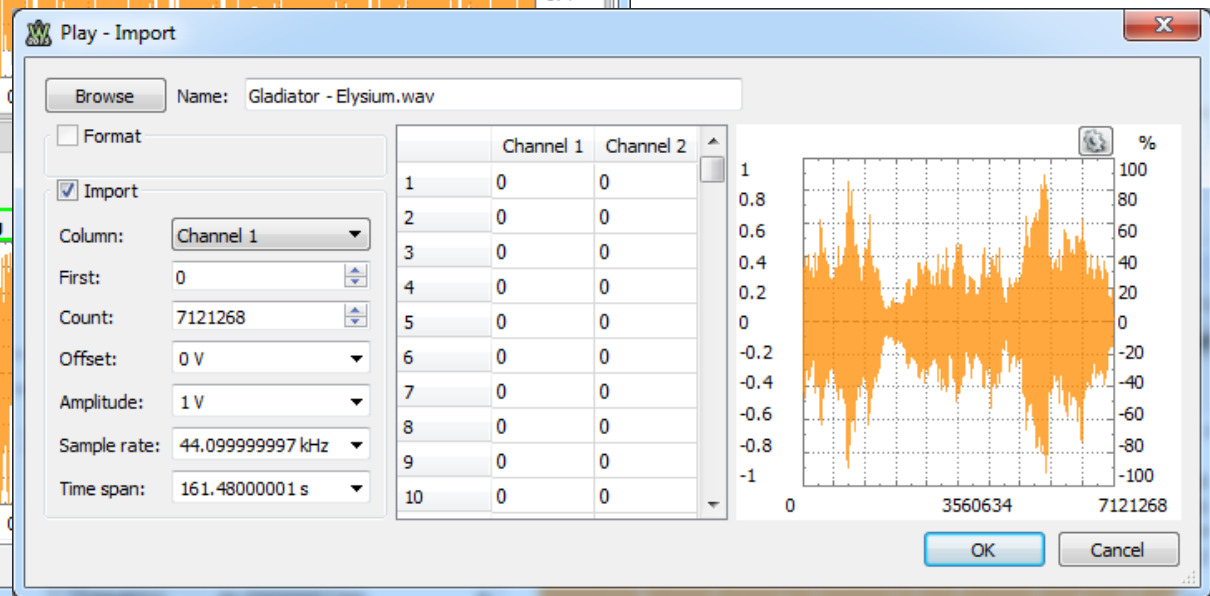


- Attach an audio headset to the Analog Discovery 2.
- AWG channel 1 in the right speaker, channel 2 in the left speaker.
- Control the amplitude for convenient audio volume.

Audio - Wavegen - play



When importing a stereo .wav file, select a different file channel for each AWG channel. Keep the file sampling rate for correct audition, or change it for effects.



Audio - Wavegen - Custom

Wavegen - Custom (Custom2)

Name: Custom2

Func Math Values File Alter

Samples: 4096 device: 4096

X from: 0.00 to: 1.00

Script Options

```
sin(2*PI*X) + 0.1*random()
```

Normalize Generate

Sample	Value
1	0.0384711
2	-0.0170583
3	-0.0138004
4	-0.0414627
5	-0.00368721
6	-0.0211587
7	0.0466554
8	-0.0178903
9	-0.0233753
10	-0.00331133
11	0.00530308
12	0.0229463

Wavegen - Custom (Custom1)

Name: Custom1

Func Math Values File Alter

Start: 0 %

Length: 100 %

Type: Sine

Cycles: 1

Amplitude: 100 %

Offset: 0 %

Symmetry: 50 %

Phase: 0 °

Normalize Generate Save Cancel Save as New

Sample	Value
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0

WaveForms (audio)

Workspace Settings Window Help

Welcome Help Wavegen 1

File Control Edit Window

Run All Channels Synchronize None none continuous infinite Repeat Trigger

Channel 1

Run Enable Custom

New Import Edit

Custom1

Frequency: 1 kHz

Sample Rate: 4.096 MHz

Amplitude: 1 V

Offset: 0 V

Phase: 0 °

Manual Trigger

Discovery2 SN:210321753526 Status: LV

Ready

Graph showing a complex waveform over 1 ms.

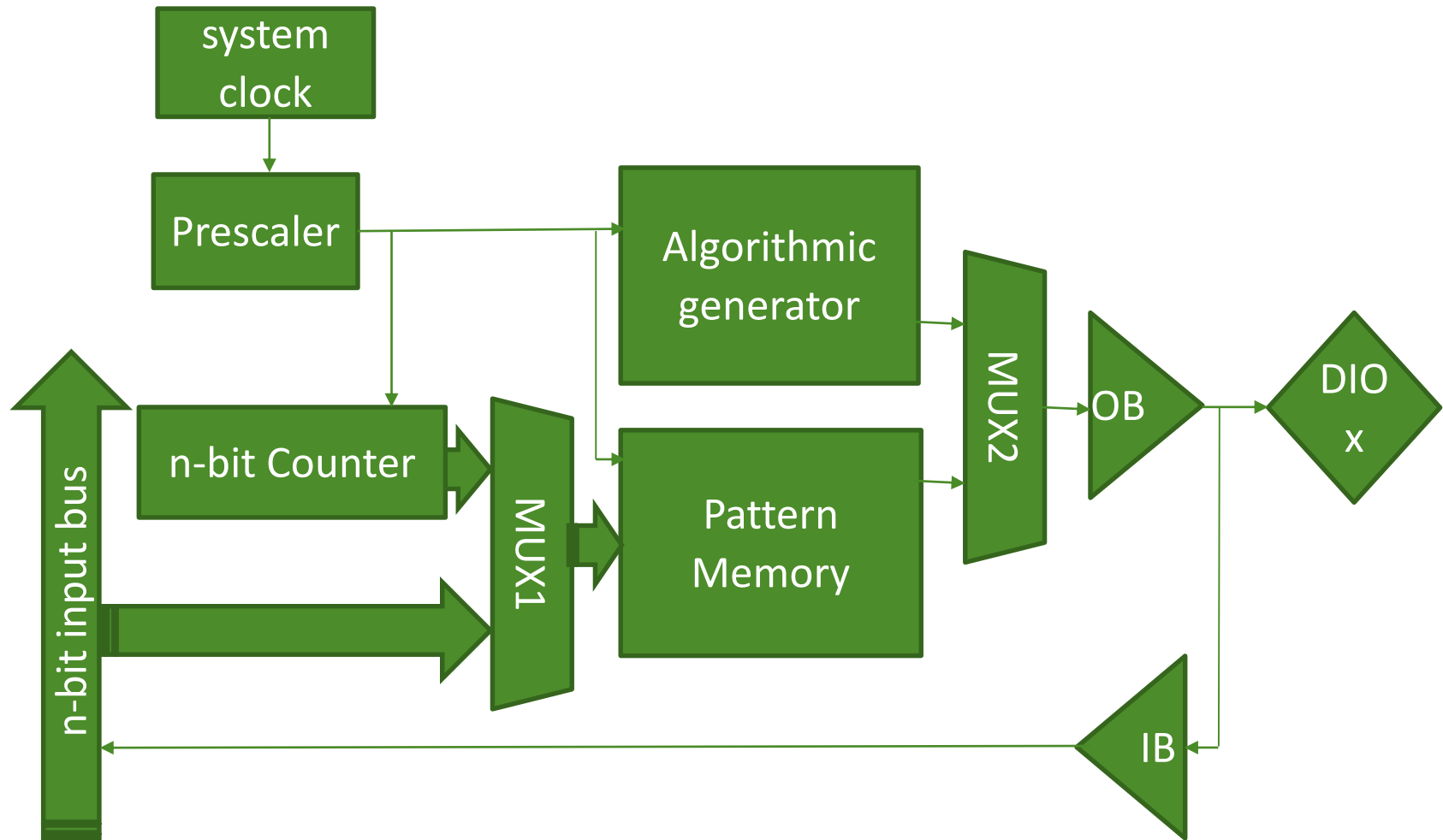
Generate AWG

buffer data:

- Equations
- Functions
- Sample values
- Files
- Modifiers
- Mouse draw

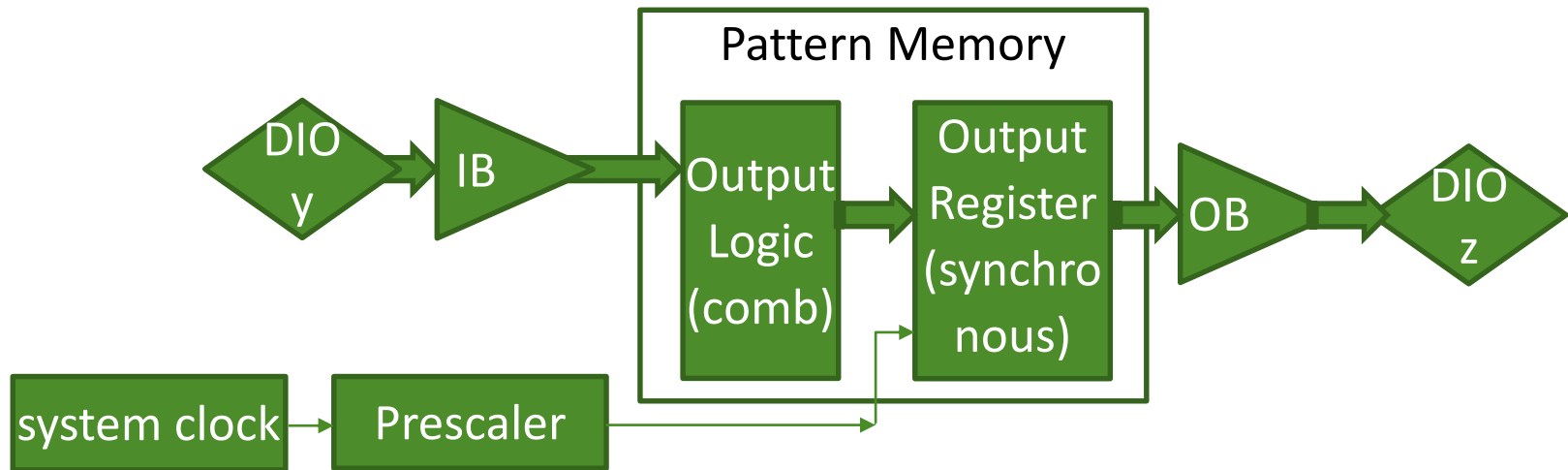
ROM Logic

DIO pin block diagram



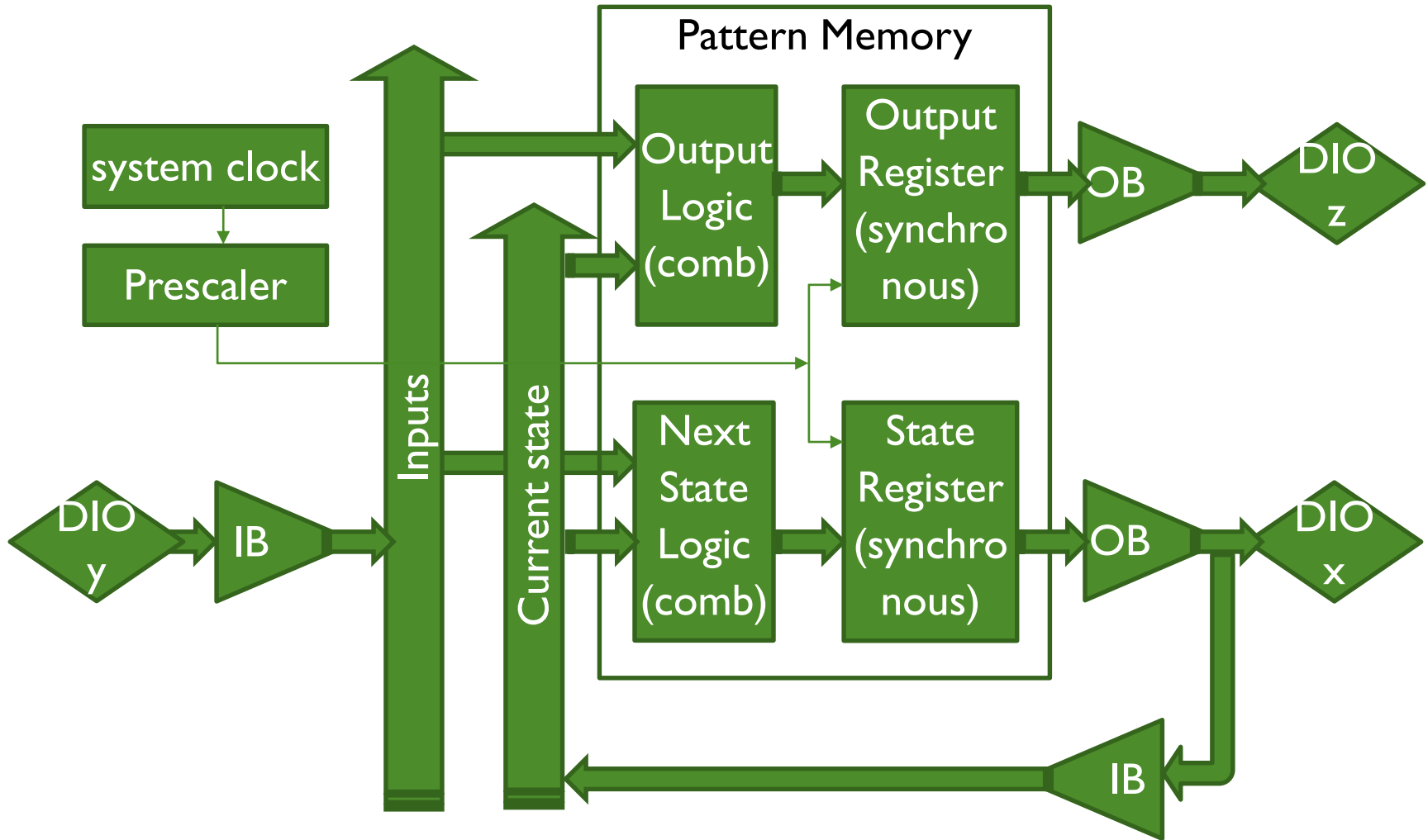
ROM Logic

Synchronized combinatorial circuit



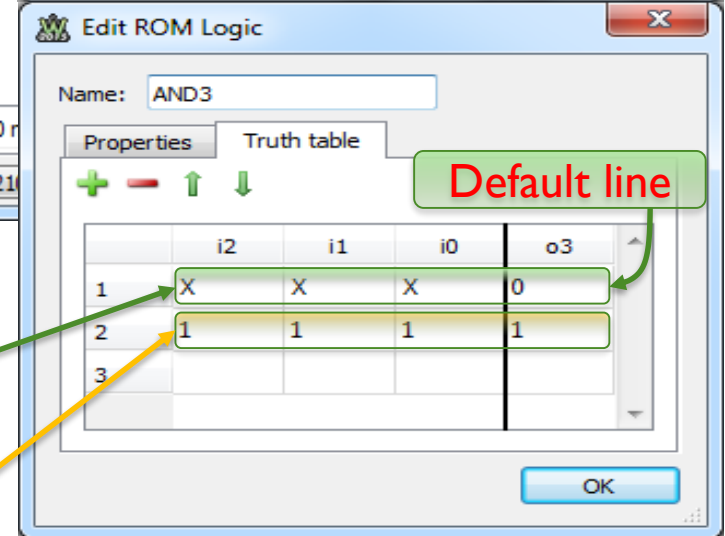
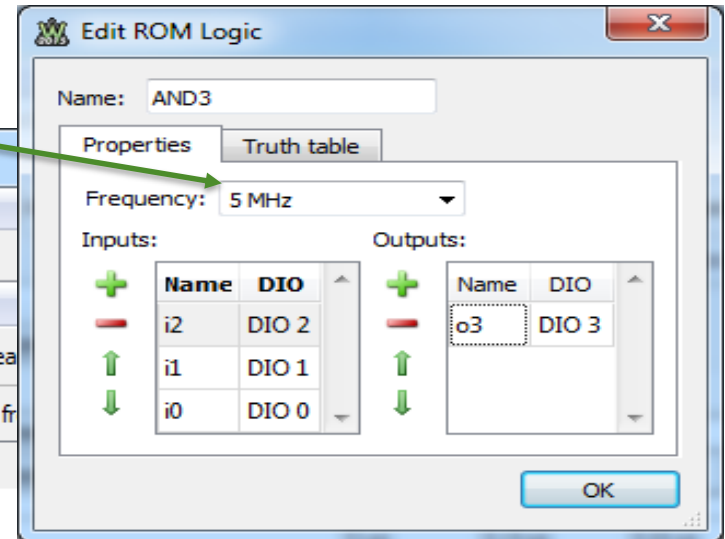
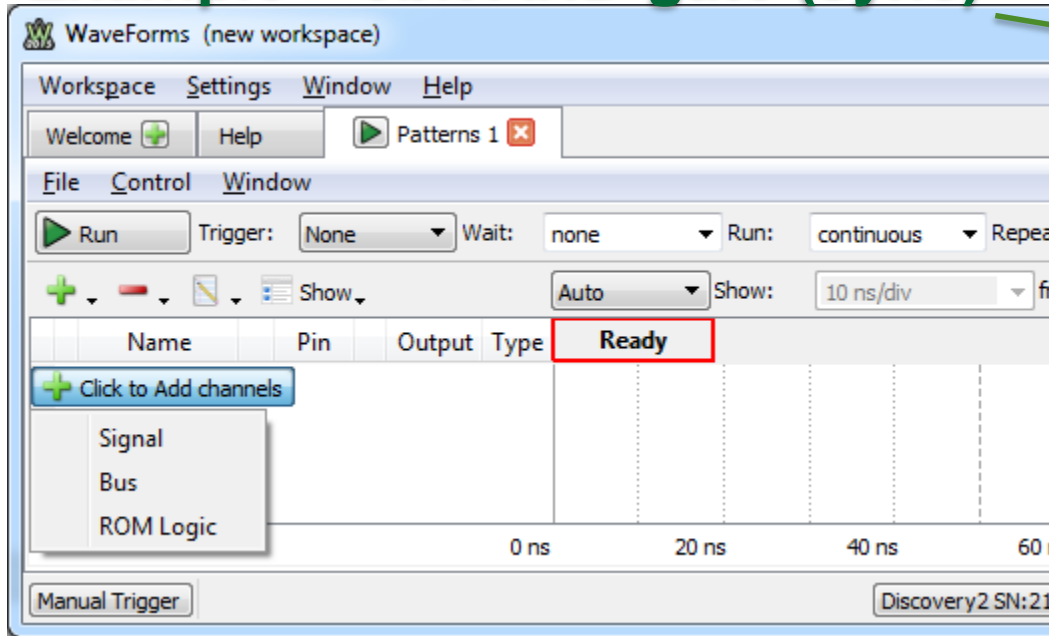
ROM Logic

Mealy State Machine



ROM Logic

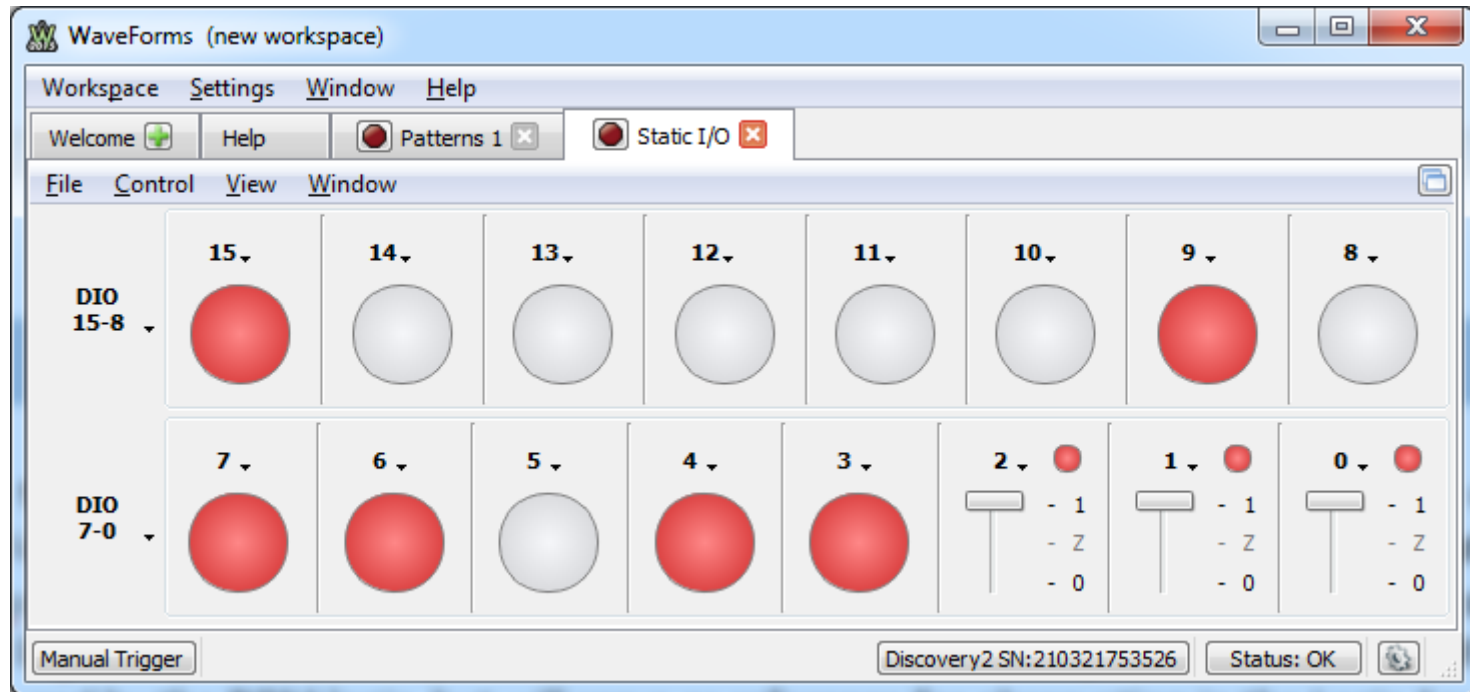
Example – 3bit AND gate (sync)



- Rename signals in the Properties tab, (optional).
- Use 0, 1 or X (don't care) in the input pane.
- An n-bit expanded Truth Table has 2^n lines. A line with k X values is the collapsed version of 2^k expanded lines.
- All possible input cases need to be covered.
- The table lines are "sequential", i.e. a line can override lines above it in the table.

ROM Logic

Example – 3bit AND gate (sync); static verification



ROM Logic

Example – 3bit AND gate (sync); dynamic verification Adding stimuli and clock reference in the Pattern Generator

The screenshot shows the WaveForms software interface with the following configuration:

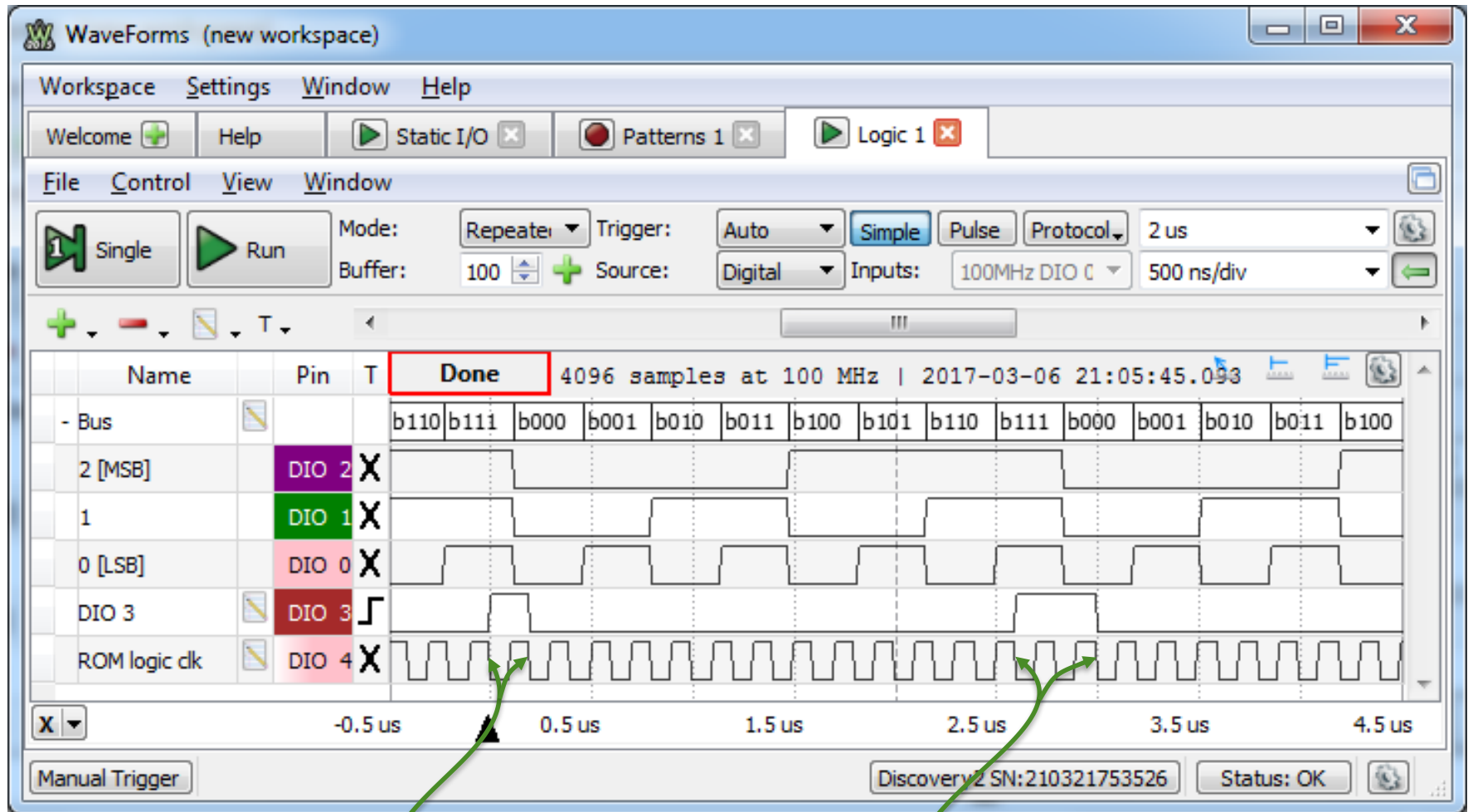
Name	Pin	Output	Type	Parameter1	Running
- ROM		PP	ROM	5 MHz	No preview available
o3	DIO 3				No preview available
- Bus		PP	Binary Counter	2.941 MHz	
2 [MSB]	DIO 2				
1	DIO 1				
0 [LSB]	DIO 0				
ROM Logic clk	DIO 4	PP	Clock	5 MHz	

Two red callouts highlight the following configurations:

- Stimulus for the AND gate:** Binary Counter, 2.941 MHz, connected to DIO 2, 1, and 0.
- ROM Logic reference Clock:** Clock, 5 MHz, connected to DIO 4.

ROM Logic

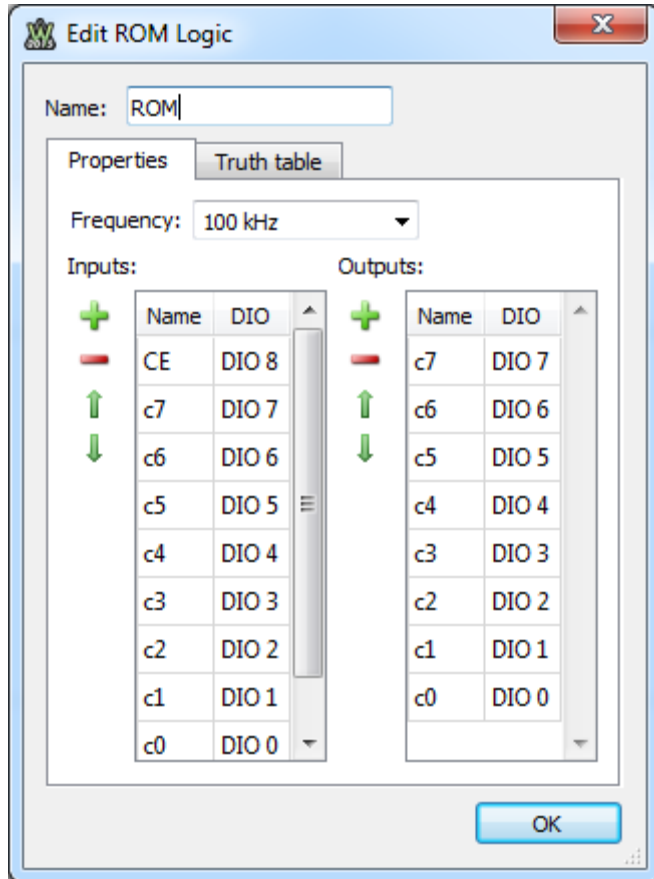
Example – 3bit AND gate (sync); dynamic verification
Observing output (DIO3) on the Logic Analyzer



DIO3 output synchronized on the 5MHz ROM logic clk
(falling edge)

ROM Logic

Example – 8bit binary counter with CE



- f. If a signal is used in both input and output panes, it is a State Variable; the input pane shows its value in the Current State and the output pane indicates its value in the Next State.
- g. The same functional block might be split in multiple ROM logic objects, to optimize the Truth Table(s).

ROM Logic

Example – 8bit binary counter with CE

Edit ROM Logic

Name: ROM

Properties Truth table

	CE	c7	c6	c5	c4	c3	c2	c1	c0	c7	c6	c5	c4	c3	c2	c1	c0
1	X	X	X	X	X	X	X	X	0	c7	c6	c5	c4	c3	c2	c1	CE
2	X	X	X	X	X	X	X	0	1	c7	c6	c5	c4	c3	c2	CE	/CE
3	X	X	X	X	X	X	0	1	1	c7	c6	c5	c4	c3	CE	/CE	/CE
4	X	X	X	X	X	0	1	1	1	c7	c6	c5	c4	CE	/CE	/CE	/CE
5	X	X	X	X	0	1	1	1	1	c7	c6	c5	CE	/CE	/CE	/CE	/CE
6	X	X	X	0	1	1	1	1	1	c7	c6	CE	/CE	/CE	/CE	/CE	/CE
7	X	X	0	1	1	1	1	1	1	c7	CE	/CE	/CE	/CE	/CE	/CE	/CE
8	X	0	1	1	1	1	1	1	1	CE	/CE	/CE	/CE	/CE	/CE	/CE	/CE
9	X	1	1	1	1	1	1	1	1	/CE	/CE	/CE	/CE	/CE	/CE	/CE	/CE
10																	

OK

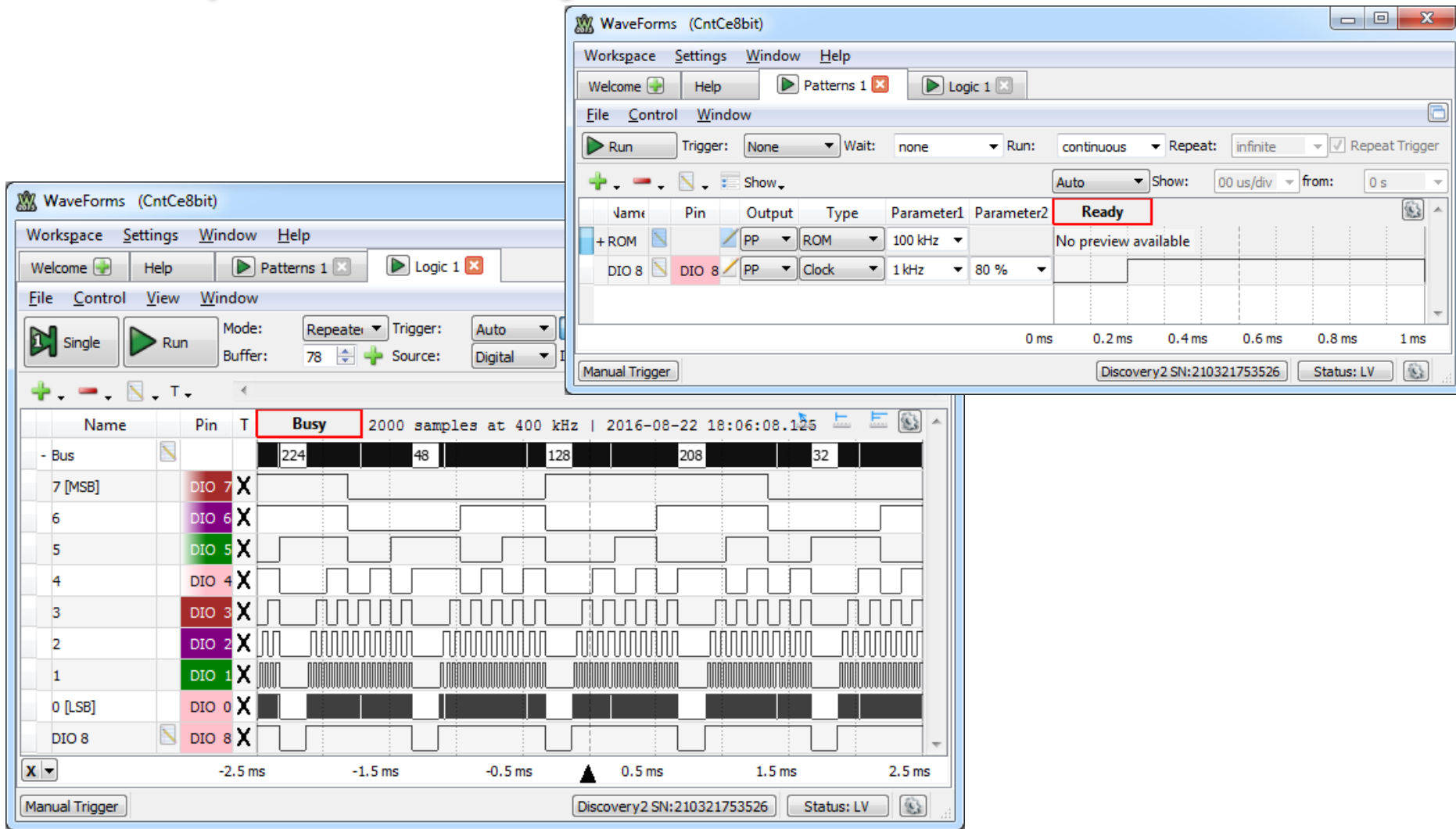
- g. 1, 0, "SignalName" or "/SignalName" in the output pane ("/" for logical NOT).
- h. A yellow background of a cell indicates a syntax error.

Remember:

- c. An n-bit expanded Truth Table has 2^n lines. A line with k X values is the collapsed version of 2^k expanded lines.

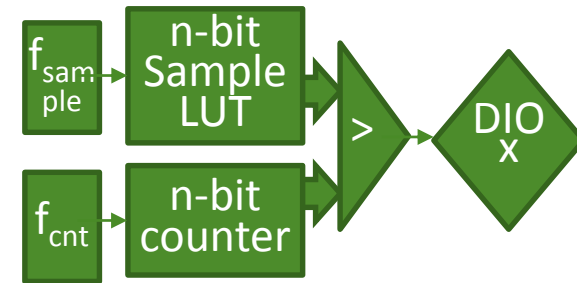
ROM Logic

Example – 8bit binary counter with CE; verification

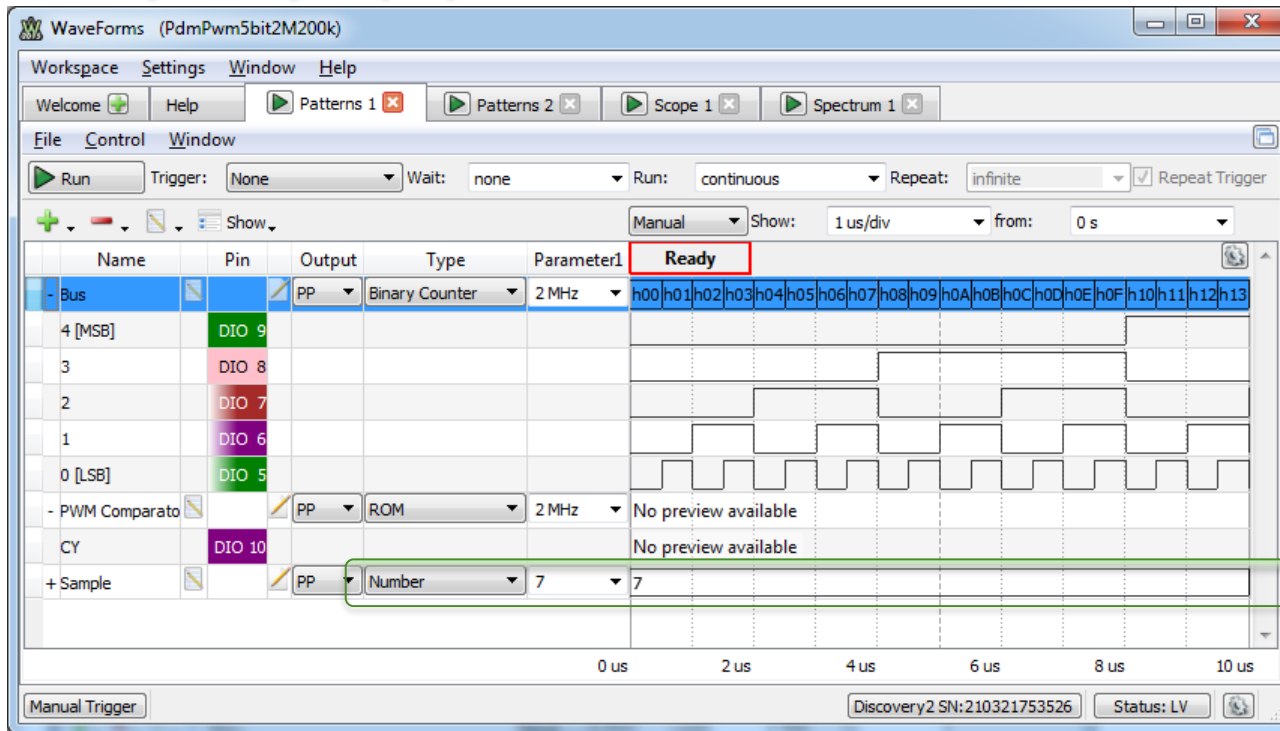


ROM Logic

Example – PWM modulator – constant sample



Workspace/Open/(No)/PdmPwm5bit2M200k.dwf3work



$$f_{carrier} = \frac{f_{cnt}}{2^n}$$

$$DF = \frac{sample}{2^n}$$

Number

ROM Logic

Example – PWM modulator

Edit ROM Logic

Name: PWM Comparator

Properties Truth table

Frequency: 2 MHz

Inputs:

Name	DIO
c4	DIO 9
c3	DIO 8
c2	DIO 7
c1	DIO 6
c0	DIO 5
s4	DIO 4
s3	DIO 3
s2	DIO 2
s1	DIO 1
s0	DIO 0

Outputs:

Name	DIO
CY	DIO 10

OK

Edit ROM Logic

Name: PWM Comparator

Properties Truth table

+ - ↑ ↓

	c4	c3	c2	c1	c0	s4	s3	s2	s1	s0	CY
1	X	X	X	X	X	X	X	X	X	X	0
2	X	X	X	X	0	X	X	X	X	1	1
3	X	X	X	X	1	X	X	X	X	0	0
4	X	X	X	0	X	X	X	X	1	X	1
5	X	X	X	1	X	X	X	X	0	X	0
6	X	X	0	X	X	X	X	1	X	X	1
7	X	X	1	X	X	X	X	0	X	X	0
8	X	0	X	X	X	X	1	X	X	X	1
9	X	1	X	X	X	X	0	X	X	X	0
10	0	X	X	X	X	1	X	X	X	X	1
11	1	X	X	X	X	0	X	X	X	X	0

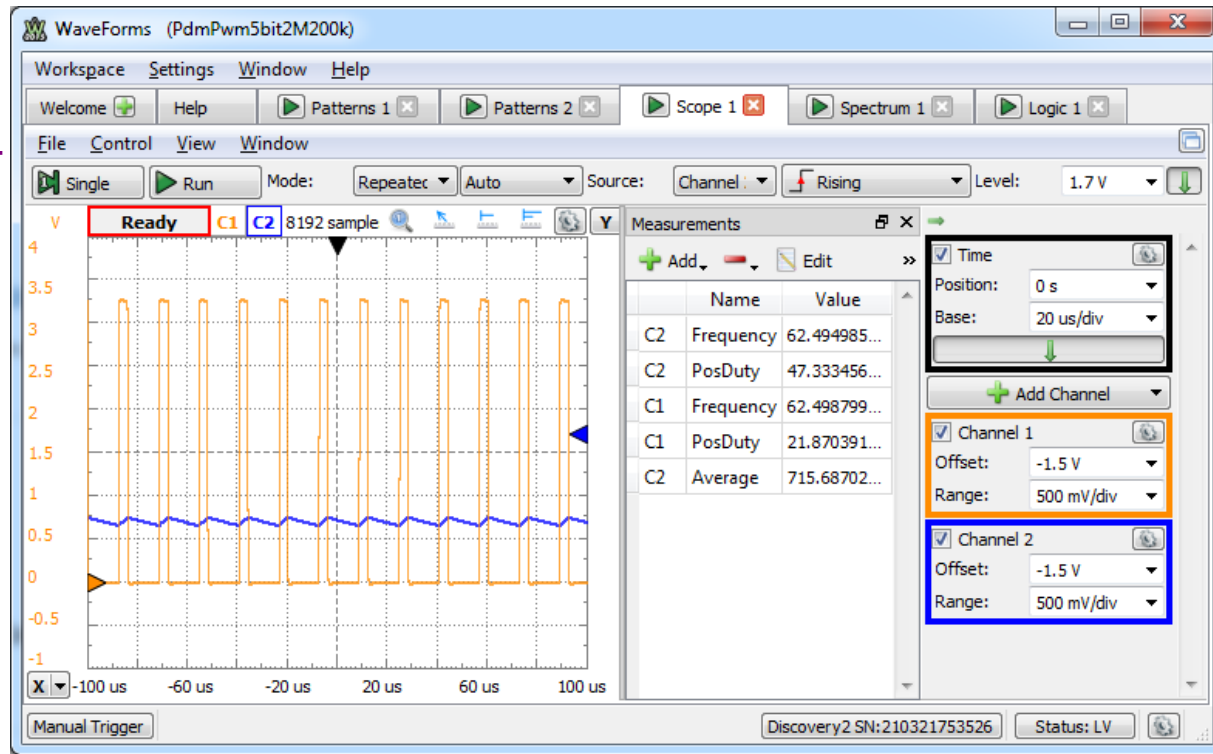
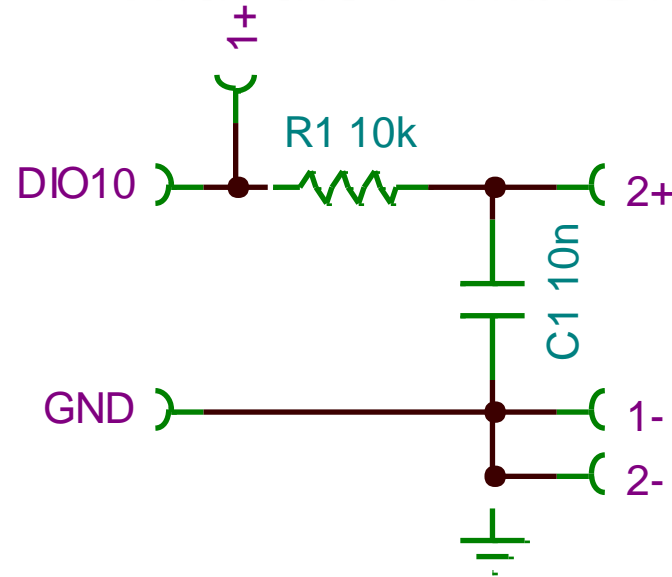
Default line

LSB compare

MSB compare

OK

RC Low Pass Filter With the PWM/PDM Modulator



Scope, Trigger

- Mode: Auto
- Source: Channel 2
- Cond: Rising
- Level: 1.7V

Scope, Time Base

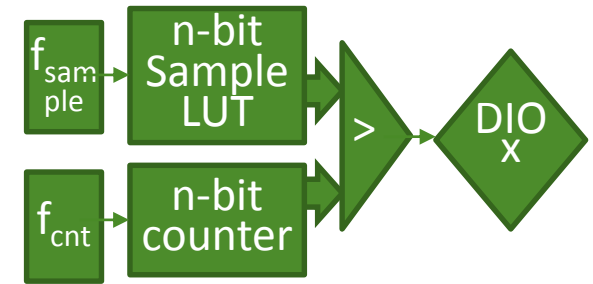
- Position: 0
- Base: 100us/div

Scope, Ch 1 and Ch 2

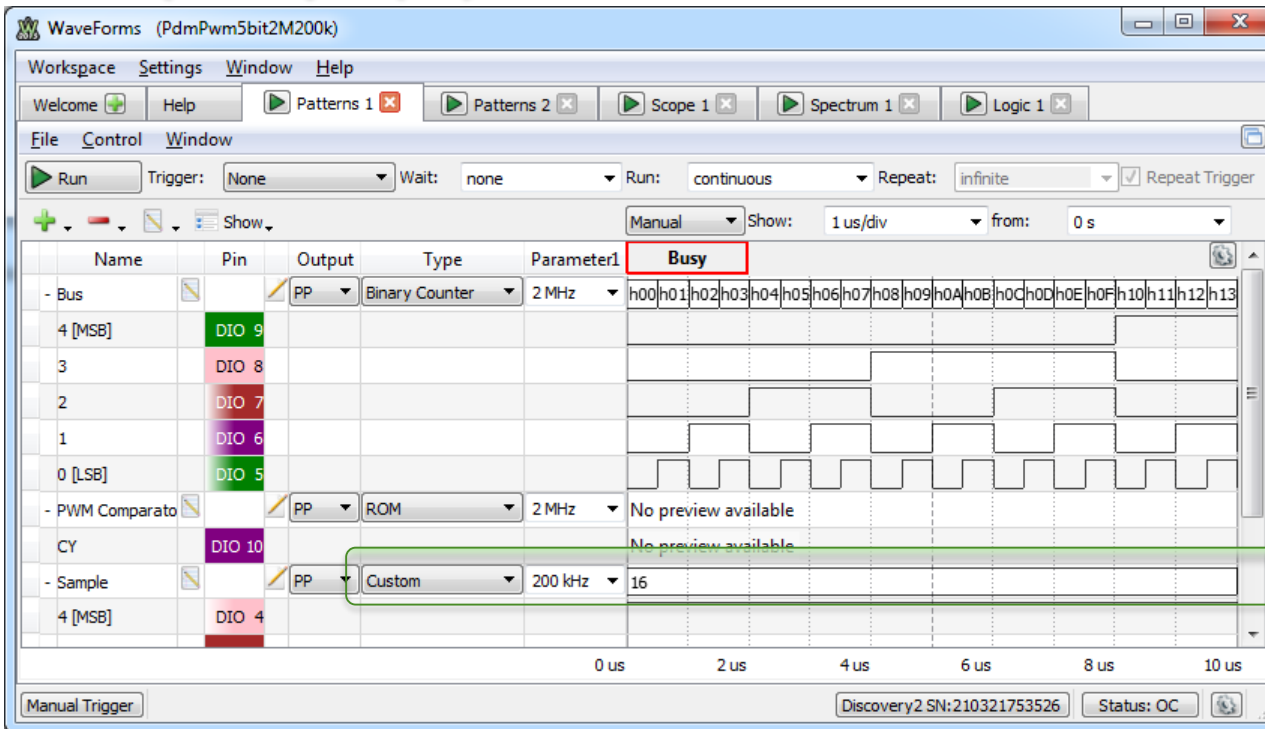
- Offset: -1.5V
- Range: 500mV/div

ROM Logic

Example – PWM modulator – variable sample



Workspace/Open/(No)/PdmPwm5bit2M200k.dwf3work



$$f_{carrier} = \frac{f_{cnt}}{2^n}$$

$$DF = \frac{sample}{2^n}$$

Custom

ROM Logic

Example – PWM modulator – variable sample (sinus)

Building a sinus sample string

$$\text{sample}_i = 16 + \text{int} \left(15 * \sin \left(\frac{2 \cdot \pi \cdot i}{100} \right) \right) \in [1 \dots 31]$$

$$f_{\text{modulator}} = \frac{f_{\text{sample}}}{100} = \frac{200\text{KHz}}{100} = 2\text{kHz}$$

The screenshot shows the 'Edit Sample' window with the following settings:

- Type: Custom
- Output: PP
- Idle: Initial
- Frequency: 200 kHz
- Minimum: 100 mHz
- Maximum: 100 MHz
- Show: 100 from: >>

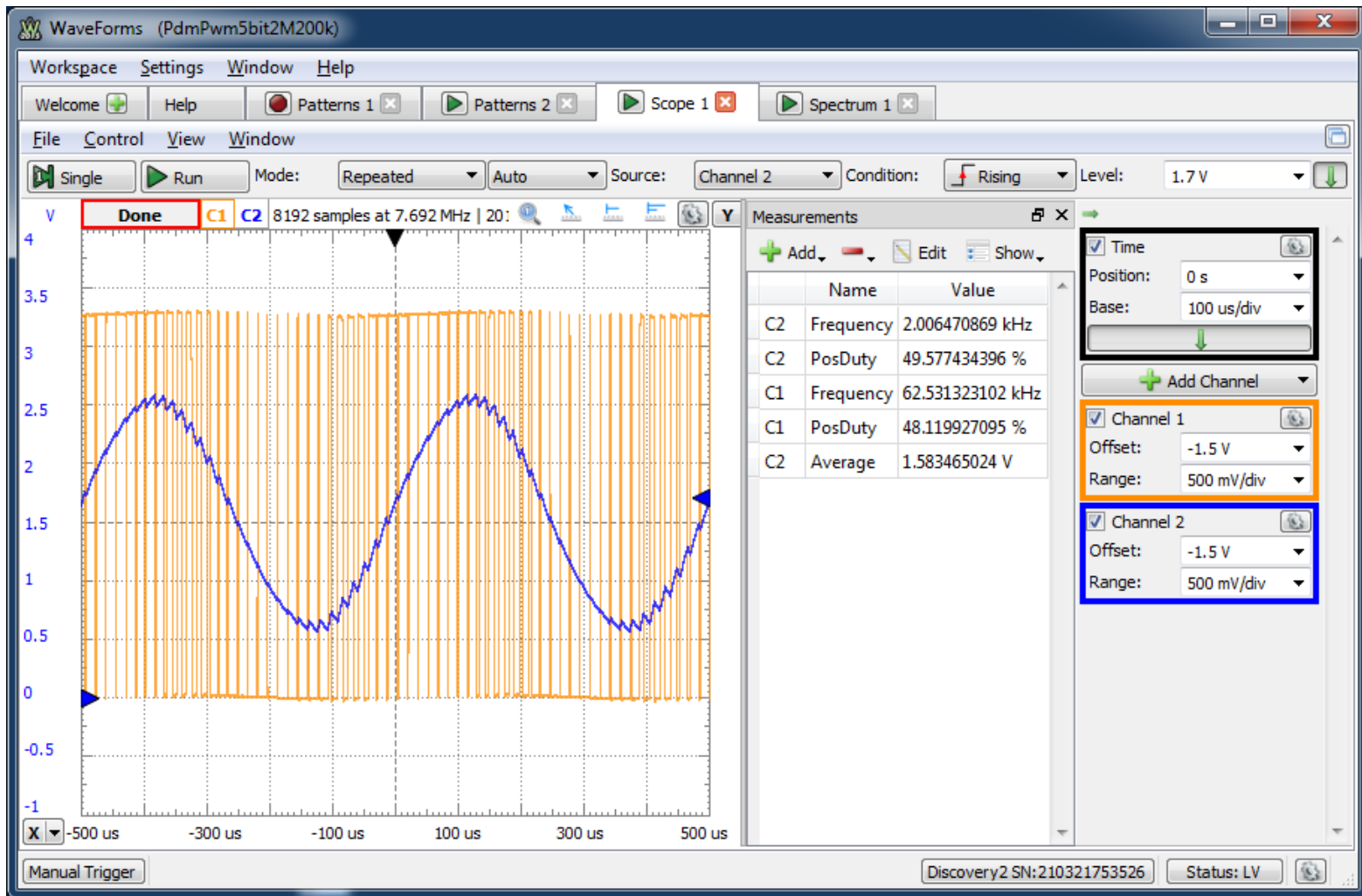
The waveform view shows a sinusoidal signal. The data table below shows the binary representation of the samples:

Sample	4	3	2	1	0
1	16	1	0	0	0
2	16	1	0	0	0
3	17	1	0	0	1
4	18	1	0	0	1
5	19	1	0	0	1
6	20	1	0	1	0
7	21	1	0	1	0
8	22	1	0	1	1
9	23	1	0	1	1
10	24	1	1	0	0
11	24	1	1	0	0
12	25	1	1	0	1
13	26	1	1	0	1
14	26	1	1	0	1
15	27	1	1	0	1
16	28	1	1	1	0
17	28	1	1	1	0

Copy 100 cells of "samples" into Sample column

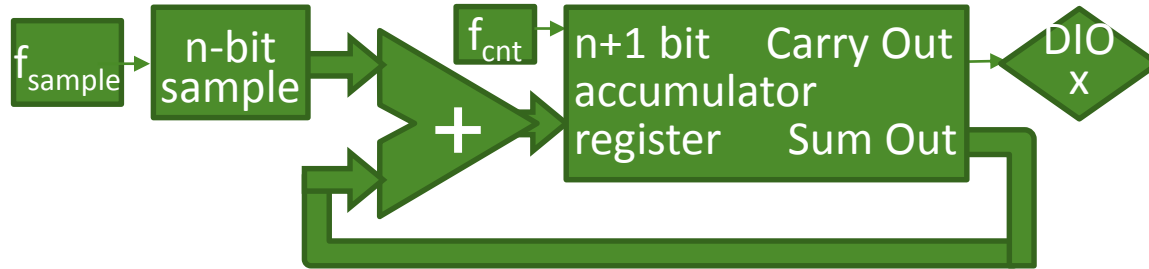
ROM Logic

Example – PWM modulator – variable sample (sinus)



ROM Logic

PDM modulator – constant sample



$$sample \in [0 \dots 2^n - 1)$$

$$DF_{avg} = \frac{\text{int} \left(\frac{M \cdot sample}{2^n} \right)}{M}$$

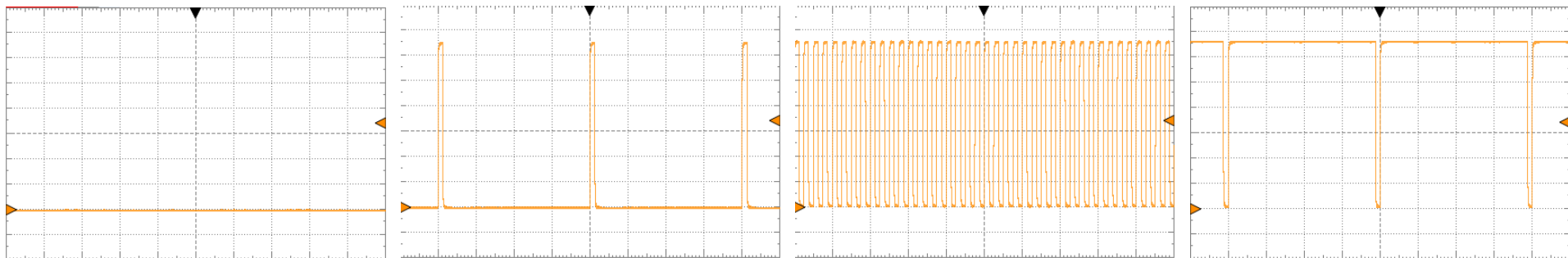
$$f_{carrier} \in \left(0, \frac{f_{cnt}}{2^n} \dots \frac{f_{cnt}}{2} \right)$$

$$\begin{cases} sample = 0 \\ DF_{avg} = 0 \\ f_{carrier} = 0 \end{cases}$$

$$\begin{cases} sample = 1 \\ DF_{avg} = \frac{1}{2^n} \\ f_{carrier} = \frac{f_{cnt}}{2^n} \end{cases}$$

$$\begin{cases} sample = \frac{2^n}{2} \\ DF_{avg} = \frac{1}{2} \\ f_{carrier} = \frac{f_{cnt}}{2} \end{cases}$$

$$\begin{cases} sample = 2^n - 1 \\ DF_{avg} = \frac{2^n - 1}{2^n} \\ f_{carrier} = \frac{f_{cnt}}{2^n} \end{cases}$$



ROM Logic - PDM modulator

Workspace/Open/(No)/PdmPwm5bit2M200k.dwf3work

WaveForms (PdmPwm5bit2M200k)

Workspace Settings Window Help

Welcome Help

File Control Window

Run Trigger: None Wait: none Run: continuous Repeat: infinite Repeat Trigger

Manual Show: 1 us/div from: 0 s

Name	Pin	Output	Type	Parameter1	Ready
Bus		PP	Binary Counter	2 MHz	h00h01h02h03h04h05h06h07h08h09h0Ah0Bh0Ch0Eh0Fh10h11h12h13
4 [MSB]	DIO 9				
3	DIO 8				
2	DIO 7				
1	DIO 6				
0 [LSB]	DIO 5				
- PWM Comparato		PP	ROM	2 MHz	No preview available
CY	DIO 10				No preview available
+ Sample		PP	Number	7	7

Edit ROM Logic

Name: PDM Accumulator

Properties Truth table

Frequency: 2 MHz

Inputs:

Name	DIO
a4	DIO 9
a3	DIO 8
a2	DIO 7
a1	DIO 6
a0	DIO 5
s4	DIO 4
s3	DIO 3
s2	DIO 2
s1	DIO 1
s0	DIO 0

Outputs:

Name	DIO
cy	DIO 10
a4	DIO 9
a3	DIO 8
a2	DIO 7
a1	DIO 6
a0	DIO 5

OK

Edit ROM Logic

Name: PDM Accumulator

Properties Truth table

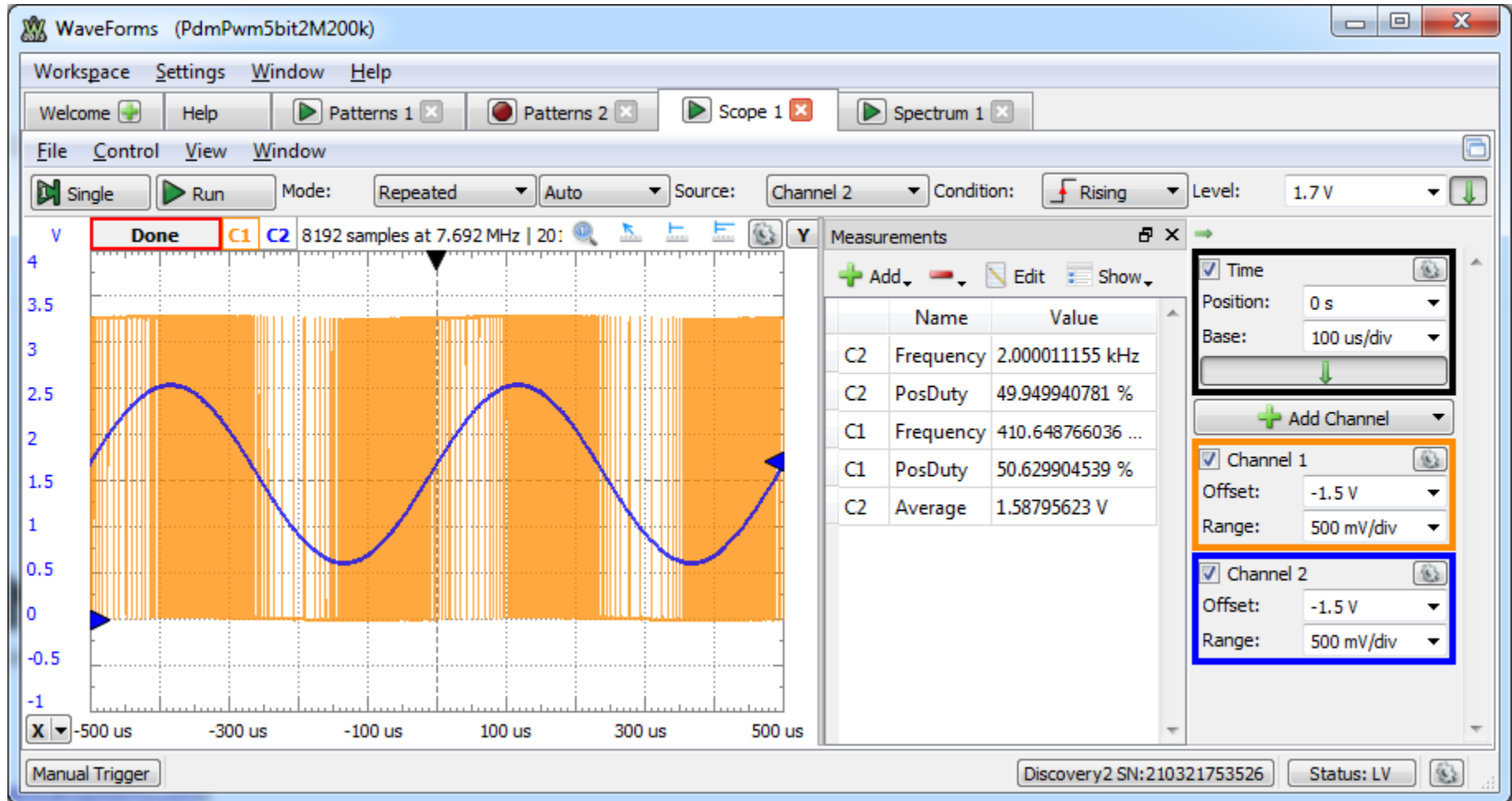
Addition truth table: 5bit+5bit=Carry&5bit
1024 lines copied from xcel file
(RomLogicTruthTables - PDM)

	a4	a3	a2	a1	a0	s4	s3	s2	s1	s0	cy	a4	a3	a2	a1	a0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1
5	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1
7	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0

OK

ROM Logic

Example – PDM modulator – variable sample (sinus)



ROM Logic

Example – PDM modulator – variable sample (sinus)

