The goal of this assignment is to get your software tools ready and write some "helloworlds".

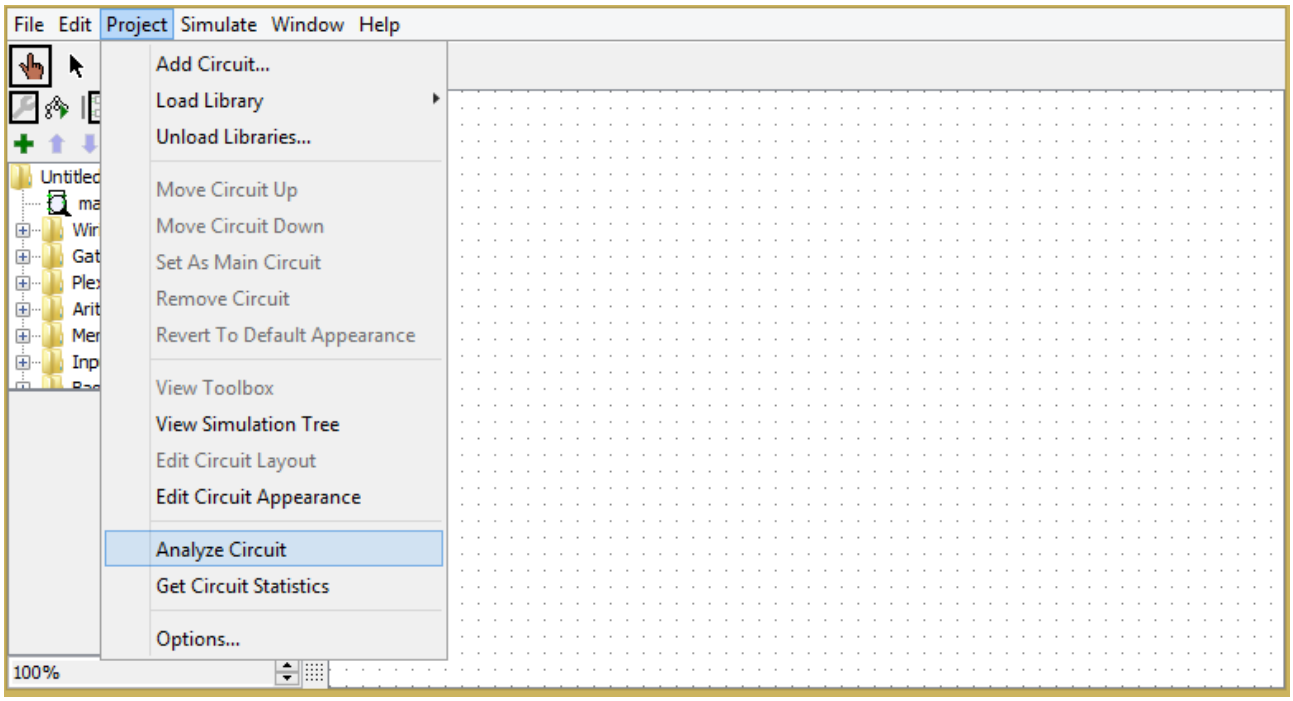
**Task 1**

Download Logisim from http://www.cburch.com/logisim/ If the *.exe file does not open, try *.jar

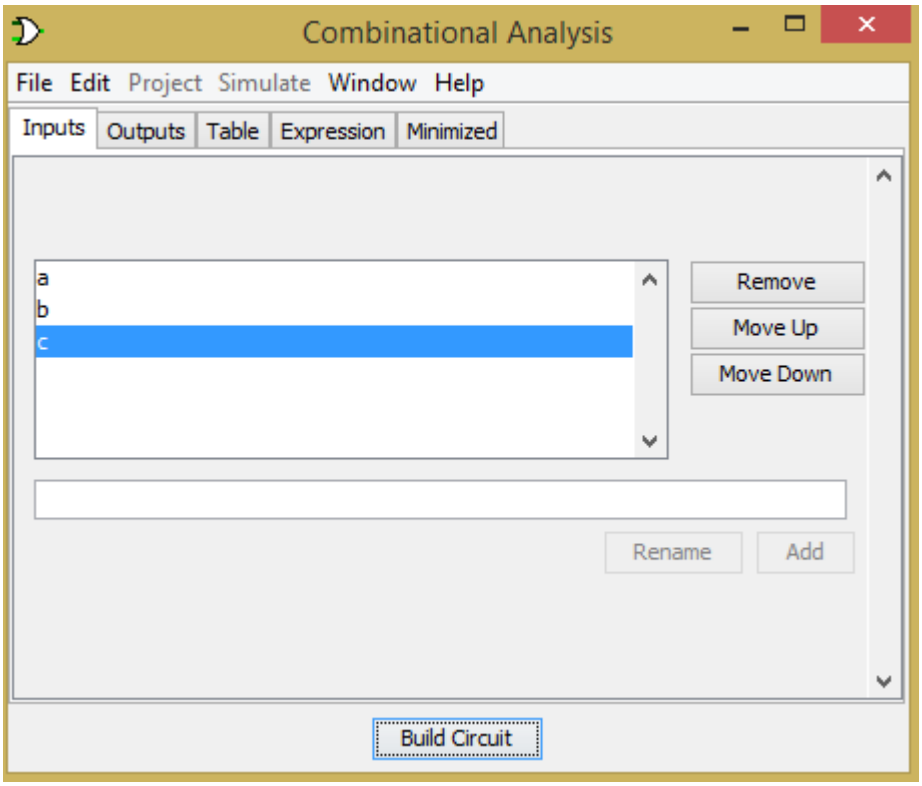
Start Logisim. Go to Help/Tutorial. Read throug the Beginner's tutorial.

Build a circuit that computes a Boolean function $y=\bar{a}\,\bar{b}\,\bar{c}+a\,\bar{b}\,\bar{c}+a\,\bar{b}\,c$ by completing the following steps.

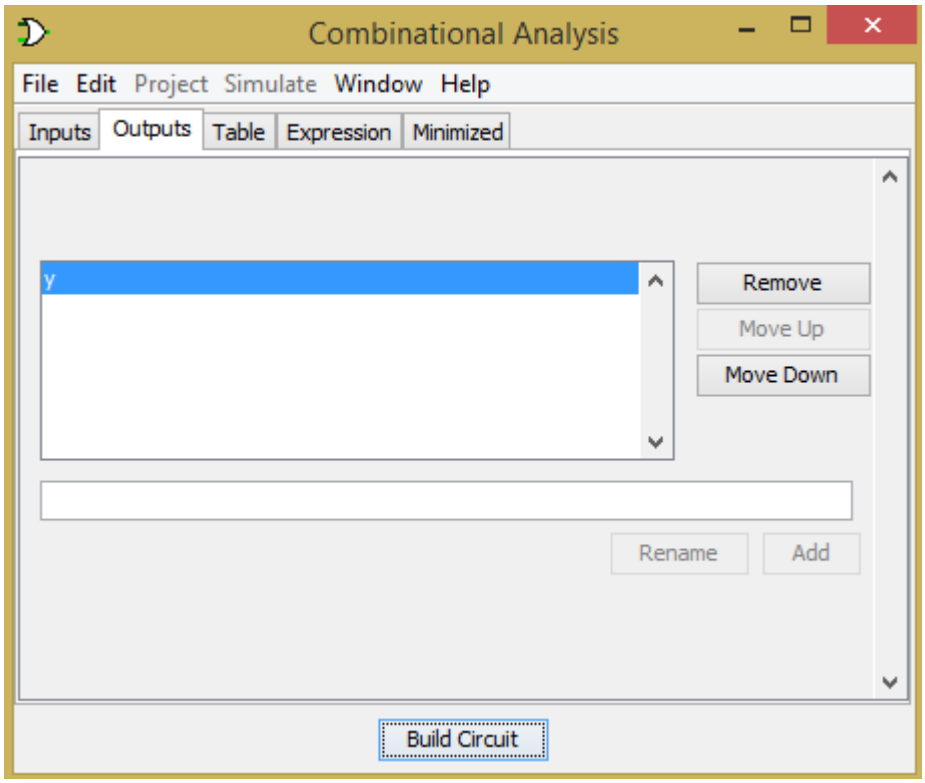
Go to Project -> Analyze circuit



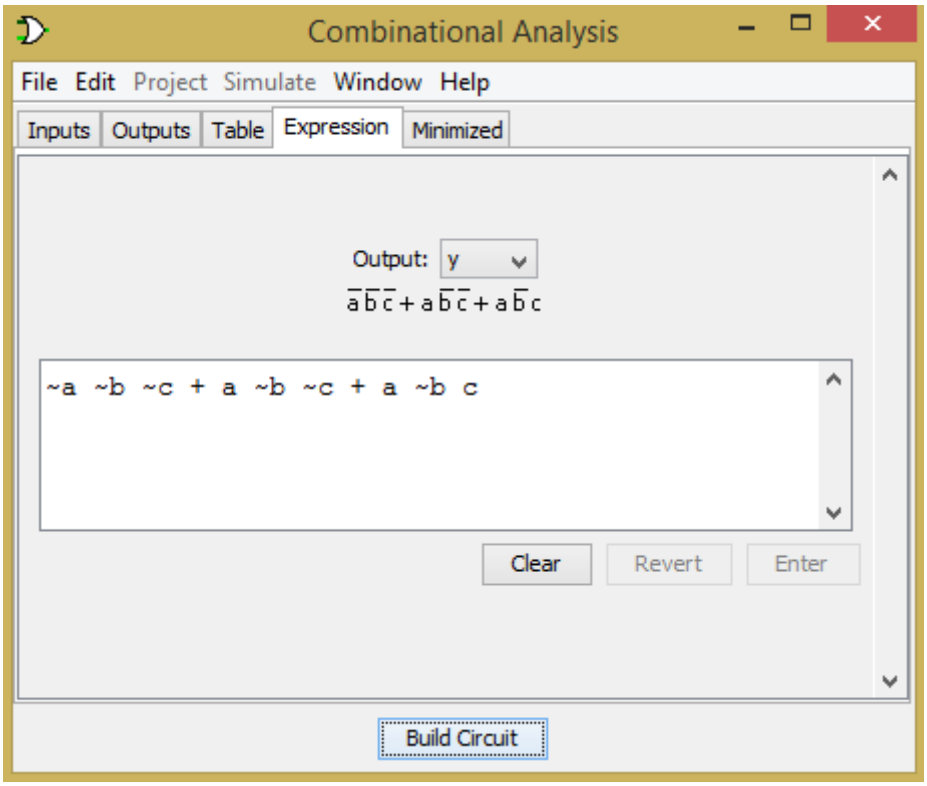

Add the input variables



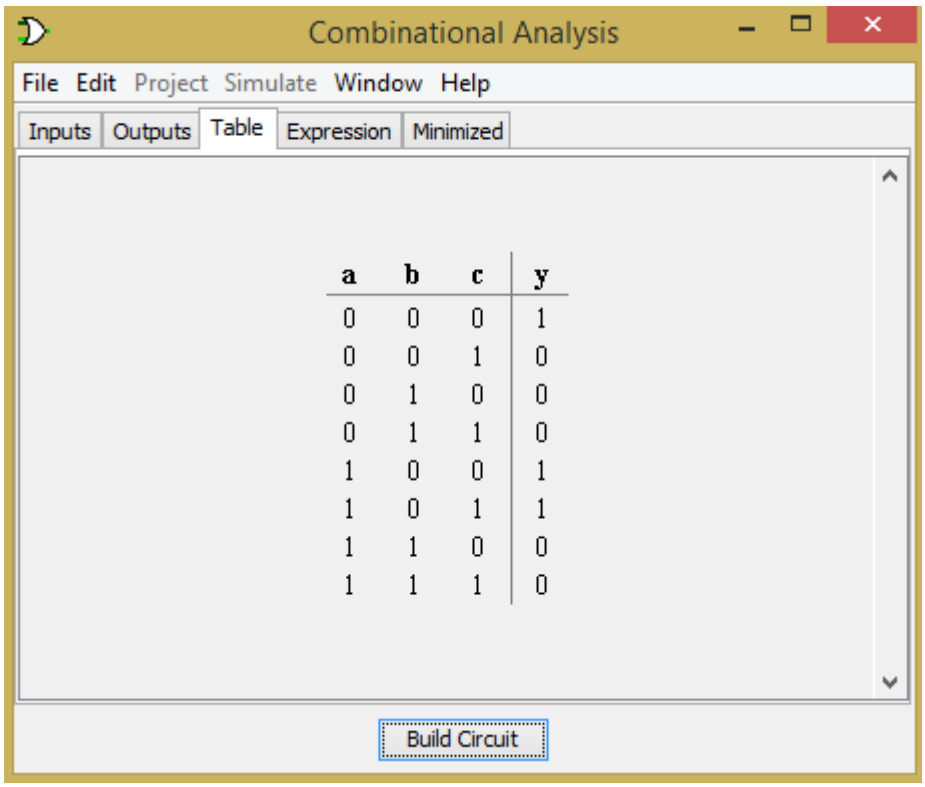
Add the output variable



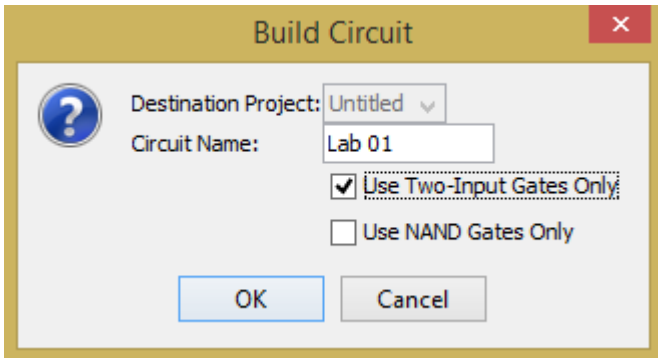

Insert the equation



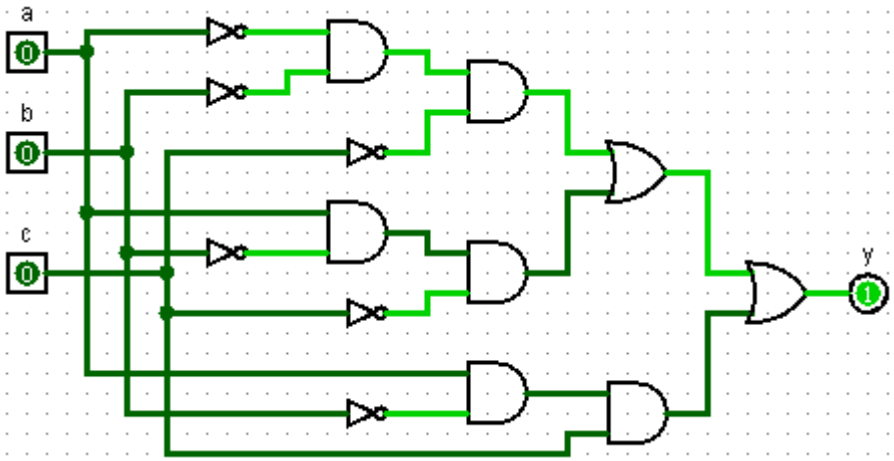

Now you can also look at the truth table:




Press "Build circuit" and choose "Use Two-Input Gates Only"




This is what you should get:



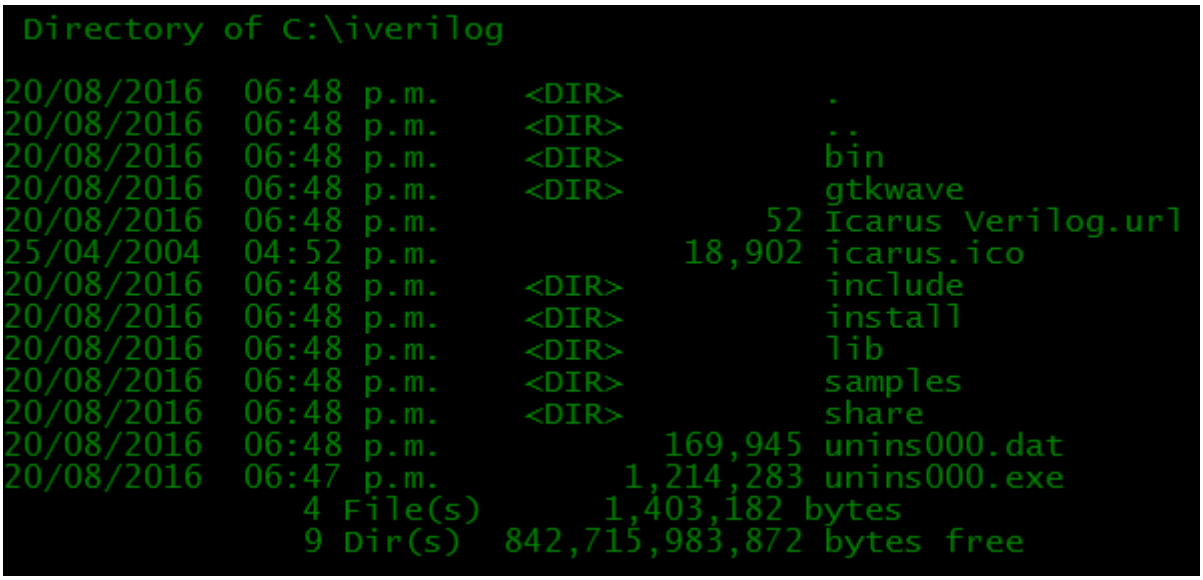

Now you can change inputs and observe the output.

**Task 2**

Install Icarus Verilog from http://iverilog.icarus.com/

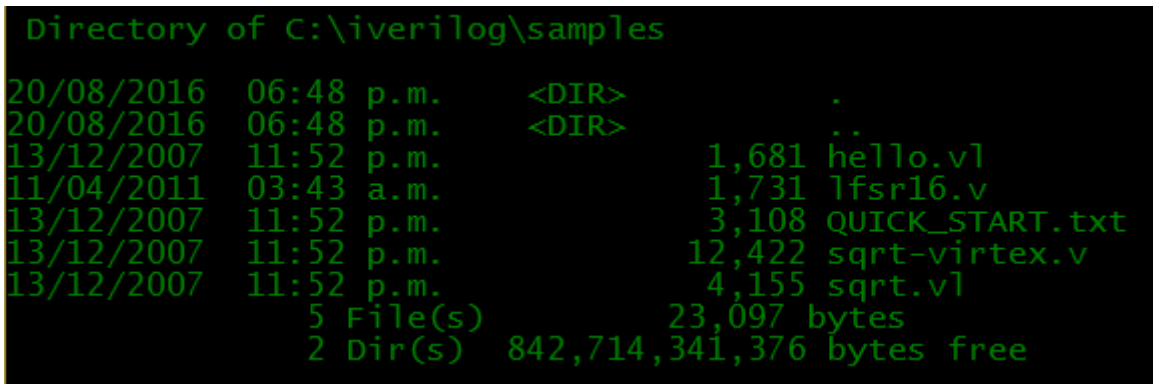
Icarus Verilog for Windows: http://bleyer.org/icarus/

Go to the iverilog directory



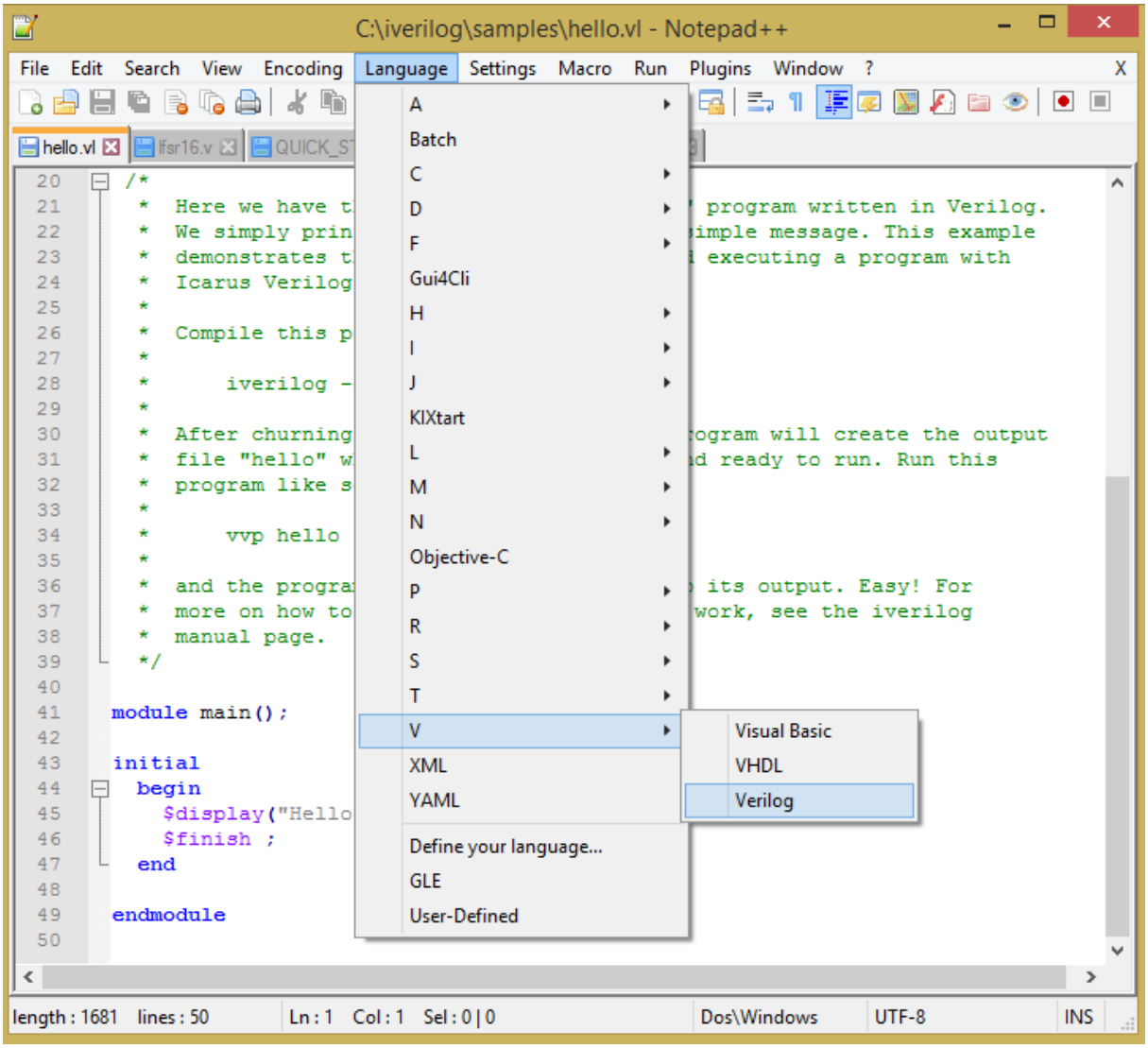

There's a subdirectory "samples" that contains a few sample files and a quick start guide.

cd samples




Open and read the file hello.vl

The Notepad++ editor provides a proper syntax highlighting for Verilog files




Compile the file:

iverilog -o hello hello.vl

The program will create the output file "hello". Run this program:

vvp hello

**Task 3**

Write a Verilog code that computes a Boolean function $y=\bar{a}\bar{b}\bar{c}+a\bar{b}\bar{c}+a\bar{b}c$

```
`timescale 1ns/1ns

module sillyfunction (input a, b, c,
                      output y);

     assign y = ~a & ~b & ~c |
                a & ~b & ~c |
                a & ~b & c;
endmodule
```

Save it in a file silly.v

Then write a testbench (in the same file) to test the sillyfunction module

```
module sillyfunction_tb ();

     reg a, b, c;
     wire y;

     // instantiate device under test
     sillyfunction dut (a, b, c, y);

     initial begin
          $dumpfile("silly.vcd");
          $dumpvars(0, dut);
     end

     // apply inputs one at a time
     initial begin
          a = 0; b = 0; c = 0; #10;
          c = 1;                #10;
          b = 1; c = 0;         #10;
          c = 1;                #10;
```

```
            a = 1; b = 0; c = 0; #10;
            c = 1;               #10;
            b = 1; c = 0;        #10;
            c = 1;               #10;
      end
endmodule
```

Compile the source file:

iverilog -o silly silly.v

This should create the output file "silly". Run this program:

vvp silly

The program will create a dumpfile silly.vcd which you can analyse in gtkwave:

gtkwave silly.vcd

You should get the following waveform