

Remote Triggered FPGA based Automation System

User Guide

Usage Scenario

- User will access the website of “Remote triggered FPGA based automation system” (<http://59.181.142.81/fpga/automation.php>)



Fig. 2.1 – Webpage of “Remote triggered FPGA based automation system”

- The user will upload his design by clicking on “Upload Verilog File”

Step 1: Upload Design Under Test Verilog File

Upload Verilog File

Fig. 2.2 – Upload Design file (.v)

- After uploading the design file, the system checks for 1-bit input ports and lists options for the user to select the clock pin.

Step 1: Upload Design Under Test Verilog File

Upload Verilog File

Uploaded Verilog File: *t/c.v*

Found following ports in the top module:

1. `clk` [1-bit] in
2. `reset` [1-bit] in
3. `vehicle_sensor` [1-bit] in
4. `highway_signal` [1:0] out
5. `farm_signal` [1:0] out

Check 2: Success!! [Top module has no inout type ports]

Check 3: Success!! [Top module has one or more 1-bit ports to function as the clock pin]

Check 4: Success!! [Top module has valid number of total input and output pins]

*** All checks passed ***

Step
2:

Select the clock pin

☒ `clk`
☐ `reset`
☐ `vehicle_sensor`

Set clock pin

Fig. 2.3 – Select the appropriate clock pin and click on “Set clock pin” button

- The user has to give appropriate test vector input for the corresponding clock.
- User can provide the input for maximum 256 clock cycles.

Set clock pin

Selected Clock Pin: *clk*

Step 3: Provide input vectors & Execute Design

Clock Cycle	reset	vehicle_sensor
0	<input type="text"/>	<input type="text"/>
1	<input type="text"/>	<input type="text"/>
2	<input type="text"/>	<input type="text"/>
3	<input type="text"/>	<input type="text"/>
4	<input type="text"/>	<input type="text"/>
5	<input type="text"/>	<input type="text"/>
6	<input type="text"/>	<input type="text"/>
7	<input type="text"/>	<input type="text"/>
8	<input type="text"/>	<input type="text"/>
9	<input type="text"/>	<input type="text"/>

Add Row

Delete Row

Set Input Vectors & Execute

Step
3:

Fig. 2.4 – Step appearing after pressing “Set clock pin”

- The user may opt out from giving an input if the input is just repeating for some number of clock cycles where the system will take the value as the value of previous clock cycle but user will have to give the input where the input is changing.
- User can add and delete rows, if required, and can extend upto maximum a total of 256 rows.

Step 3: Provide input vectors & Execute Design

Clock Cycle	reset	vehicle_sensor
0	0	0
2		1
5	1	0
7	0	1
11	1	0
16	0	1
26	1	0
31	0	1
32	1	0

Set Input Vectors & Execute


Powered by IPintentiq™

Fig. 2.5 – Input the appropriate test vectors by giving appropriate value of clock cycle and press “Set Input Vectors & Execute” button

- These inputs along with the clock pin are combined together to form a 16-bit where the rest of the bits are zeros.

Step 3: Provide input vectors & Execute Design

Clock Cycle	reset	vehicle_sensor
0	0	0
2		1
5	1	0
7	0	1
11	1	0
16	0	1
26	1	0
31	0	1
32	1	0

 Working..

Powered by IPintentic™

Fig. 2.6 - User clicks on “Set Input Vectors & Execute”

- Output in VCD format is now available to the user for download.
- A log is also displayed which is actually the compilation report of quartus.

Step 3: Provide input vectors & Execute Design

Clock Cycle	reset	vehicle_sensor
0	0	0
2		1
5	1	0
7	0	1
11	1	0
16	0	1
26	1	0
31	0	1
32	1	0

[Download Value Change Dump \(VCD\) file for the execution here.](#)

Powered by IPintentic™

Fig. 2.7 – VCD file ready for download

- Please check at the end of the log for possible error.
- If no error then download the VCD file.

11	1	0
16	0	1
26	1	0
31	0	1
32	1	0

Add Row Delete Row

Set Input Vectors & Execute

Download Value Change Dump (VCD) file for the execution

FPGA Execution Output

```

Info: Assembler is generating device programming file
Info: Quartus II Assembler was successful. 0 errors,
Info: Peak virtual memory: 378 megabytes
Info: Processing ended: Thu Jul 19 11:57:18 2012
Info: Elapsed time: 00:00:06
Info: Total CPU time (on all processors): 00:00:05
Warning: Skipped module PowerPlay Power Analyzer due to the assignment FLOW_ENABLE_POWER_ANALYZER
Info: Quartus II Full Compilation was successful. 0 errors, 36 warnings
Info: Evaluation of Tcl script TCLScript_final.tcl was successful
Info: Quartus II Shell was successful. 0 errors, 36 warnings
Info: Peak virtual memory: 224 megabytes
Info: Processing ended: Thu Jul 19 11:57:22 2012
Info: Elapsed time: 00:00:32
Info: Total CPU time (on all processors): 00:00:29
  
```

Powered by IPintento™

Fig. 2.8 – Download VCD file if no error

- This VCD file can then be viewed on the GTKWave waveform viewer and is illustrated below.
- Goto File>Open New Tab

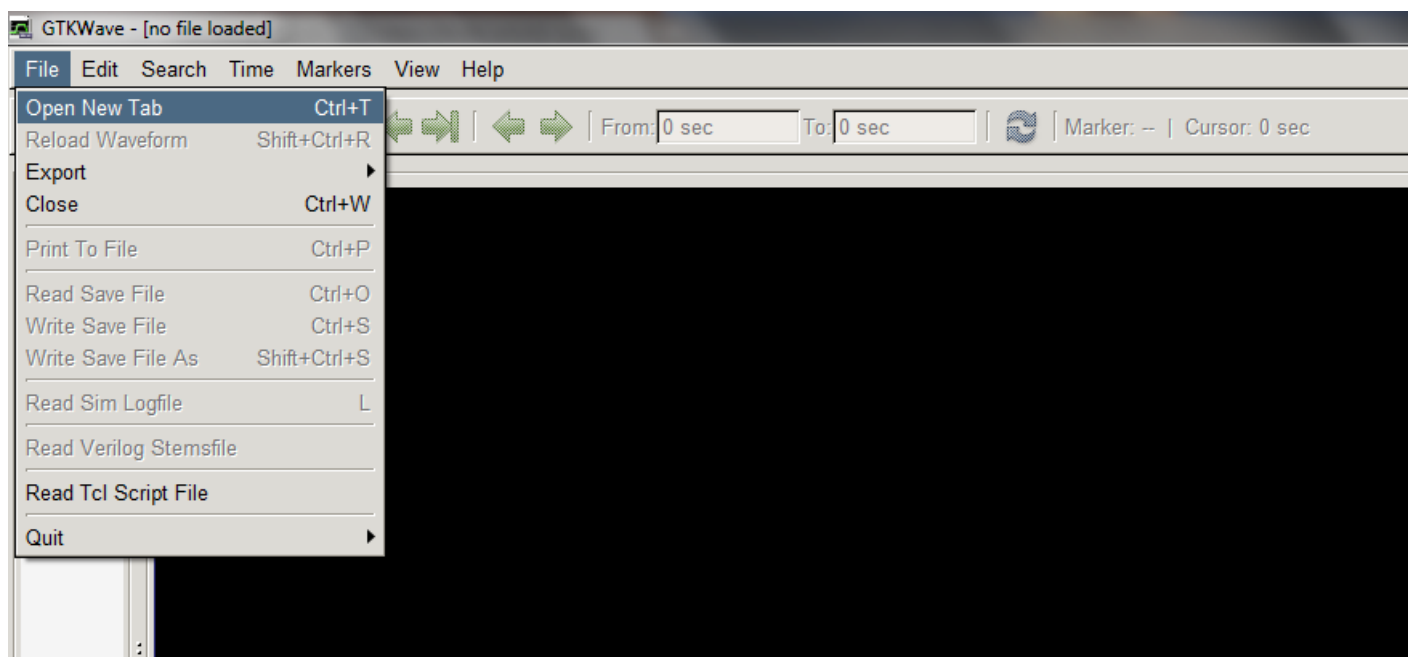


Fig. 2.8 – Open GTKWave waveform viewer (refer to GTKWave installation manual provided)

- Select the file from the directory in which .vcd file is saved.

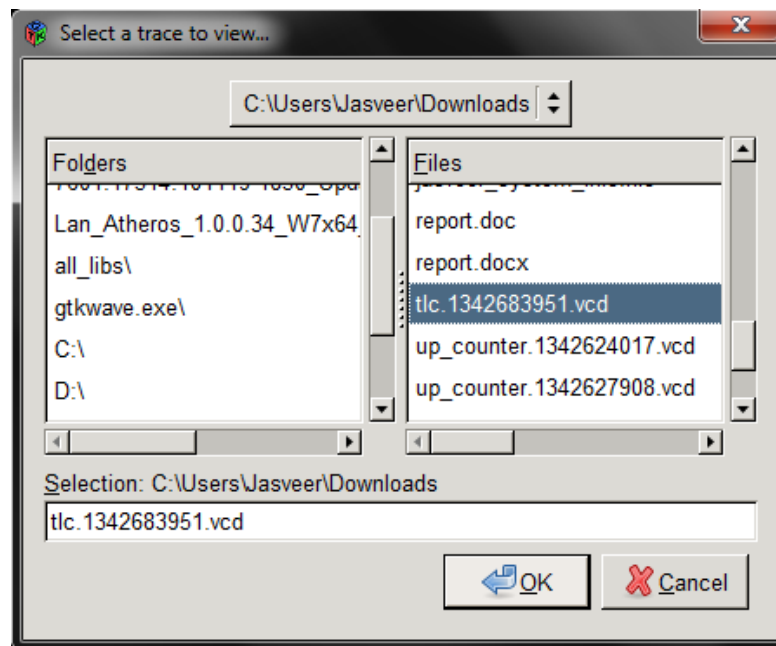


Fig. 2.9 – Select vcd file to be viewed

- The top level module name is now displayed on the top left pane of the GTKWave window.

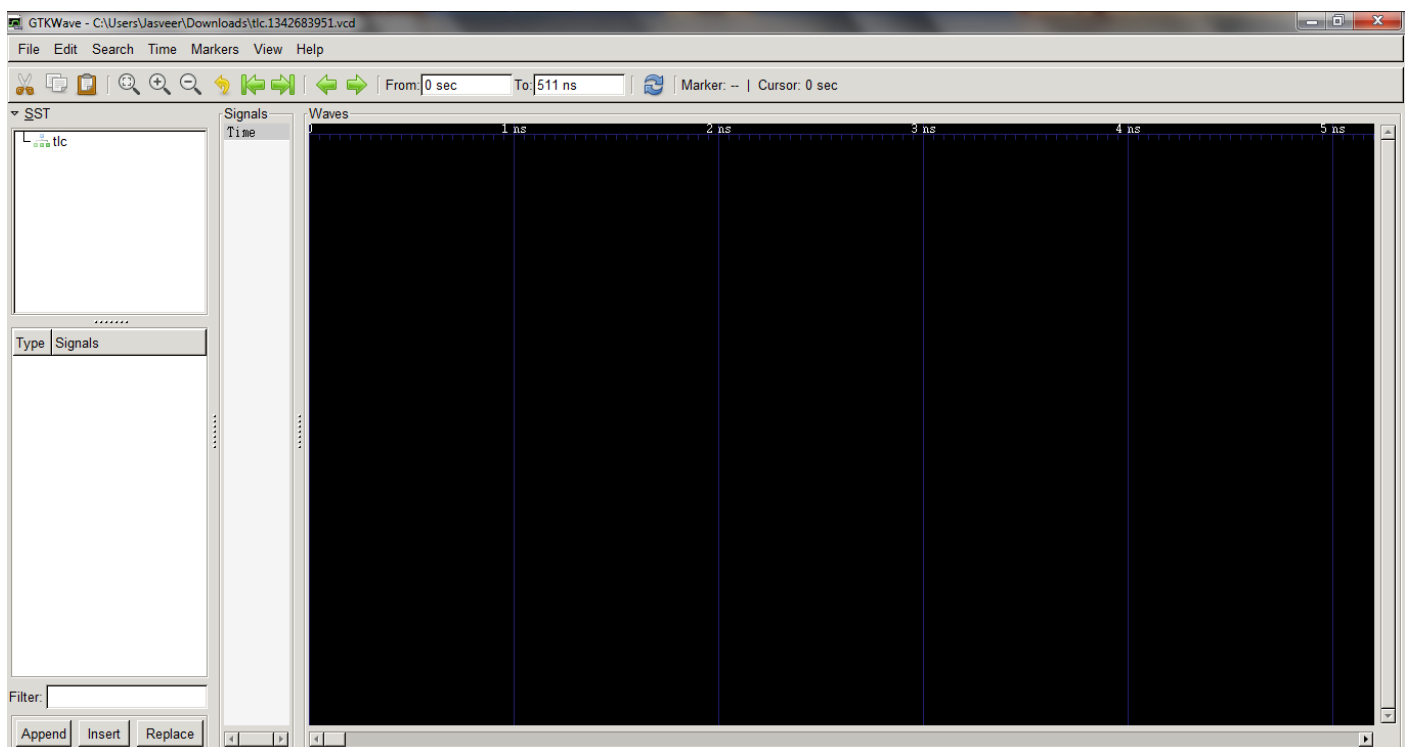


Fig. 2.10 – Select the top level module name displayed on the top left pane of the GTKWave window

- After selecting the top level module name displayed on the top left pane of the GTKWave window, the signals can be seen on the bottom left pane of the GTKWave window.

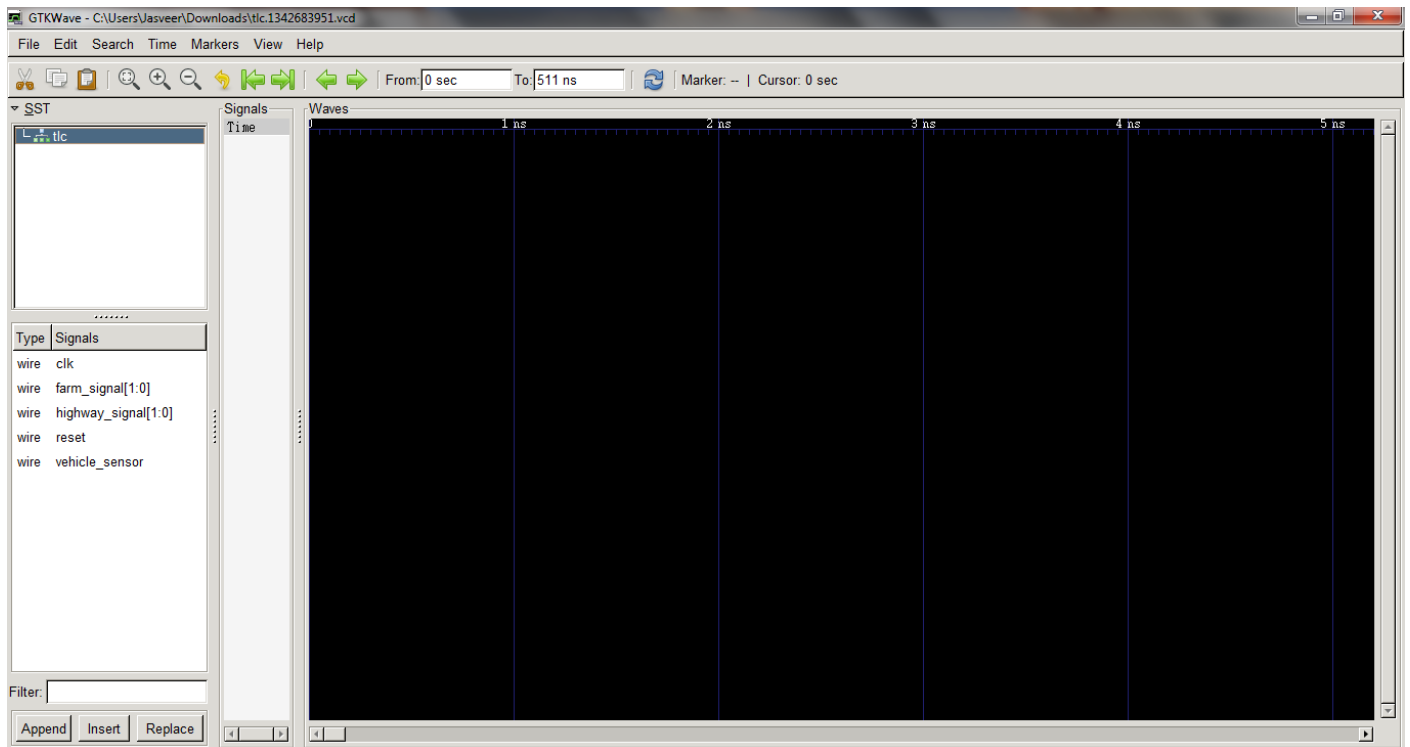


Fig. 2.11 – Open GTKWave waveform viewer (refer to GTKWave installation manual provided)

- Select all of the signals and press the button Append.

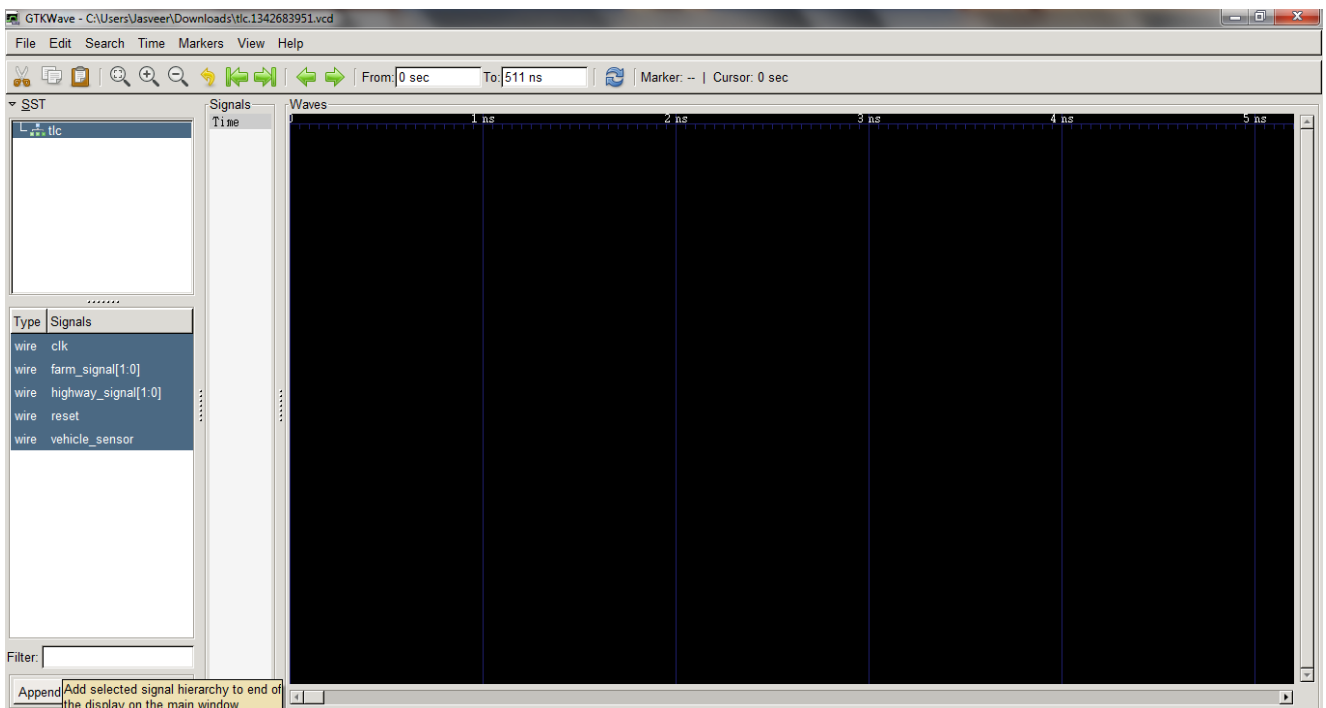


Fig. 2.12 – Select all of them and press append

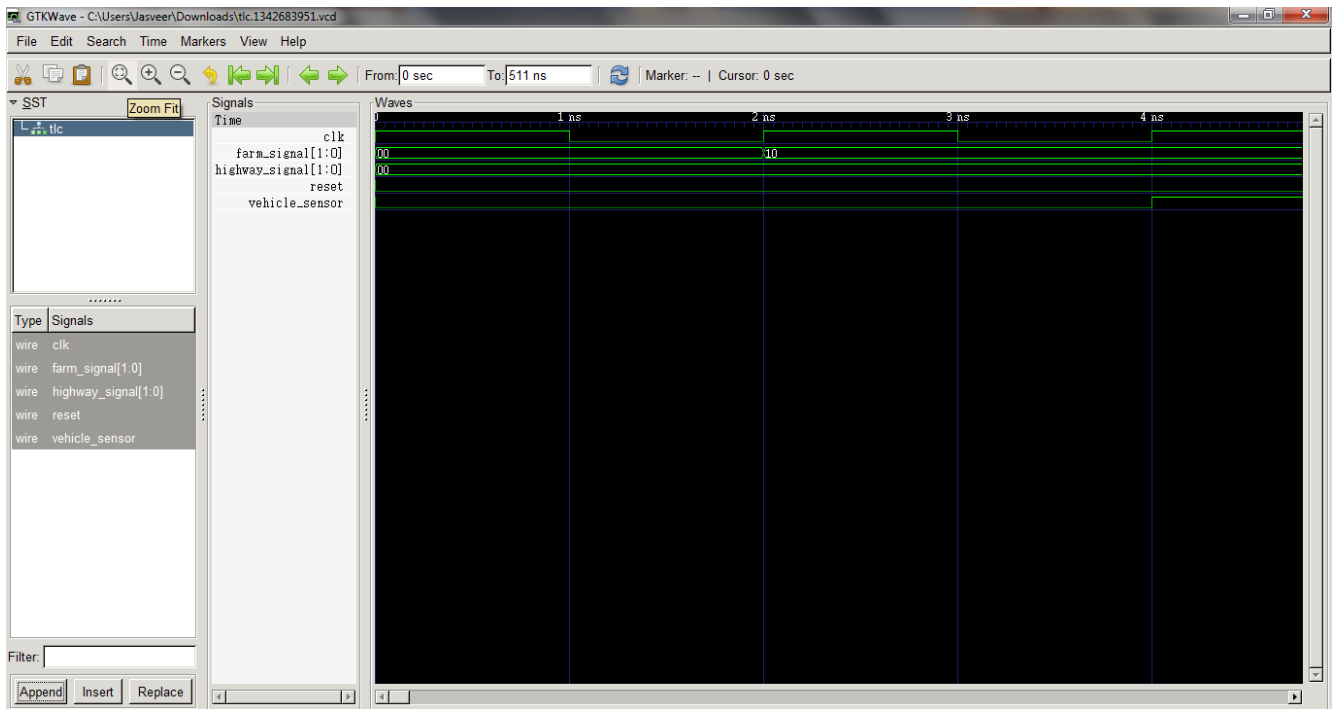





Fig. 2.12 – The waveform can be seen now

- Use the button,  called Zoom-fit to see the whole 256 clock cycles on your screen.
- Use the button,  called Zoom-in to see larger view i.e. less no. of cycles on your screen.
- Use the button,  called Zoom-out to see the large no. of cycles on your screen.