UNIVERSITI TEKNOLOGI MALAYSIA

Getting Started with Quartus

Muhammad Mun'im Ahmad Zabidi, Musa Mohd Mokji, Izam Kamisian, Norhafizah Ramli

Abstract
This article introduces the Quartus software, where to find it, install it and check out the installation with a simple design.

[1] *Department of Electronic and Computer Engineering, School of Electrical Engineering, Faculty of Engineering*
*__Corresponding author__: munim@utm.my

1. What is Quartus

Quartus II is a software package for design and development of digital system based on Altera FPGA and CPLD devices. The main tasks simulation and verification activities. The designed model are simulated for functional and timing verification on Quartus II before integrated with the actual Altera devices for hardware verification. Quartus can also be used even without access to the actual Altera devices, if we only want to go up to the simulation level.

In this tutorial, Quartus II 13.1 Web Edition will be used. This software supports both 32-bits and 64-bits operating system. Users may also use other version that meet their system requirement and the targeting Altera device.

Quartus is available in versions 13.0sp1 through 20.4 (up to March 2021). All previous version are discontinued. Up until version 16, the software was called Quartus II. After Intel took over Altera in 2016, three changes were made:

- Intel branding in the software
- The Altera Quartus II is now called Intel® Quartus® Prime and Quartus II Web Edition become Quartus Prime Lite Edition
- Introduction of the Pro edition which supports the partial reconfiguration capability of the new Arria 10 FPGA

Table 1 summarizes the differences between pro and non-pro versions. As the version numbers get bigger, the software becomes more complex and require more computer resources. Version 13.0sp1 is the only version to support the Cyclone II low-cost FPGA. If you are not using Cyclone II, go for version 13.1. Install higher versions only if you installation problems with these two versions.

Table 1. *Quartus Prime editions.*

Edition	Device Support	Cost
Pro Edition	Focus on top-of-the-line devices	30 day trial
Standard Edition	Widest device support	30 day trial
Lite/Web Edition	Entry-level	Free

2. Getting Quartus

This section shows the steps to download version 13.1 of Quartus II Web Edition.

The first step to downloading is to point your browser to `http://fpgasoftware.intel.com/13.1/?edition=web`.

Figure 1. *Choose a version.*

You can download the full Quartus II Web Edition bundle which is 4.4GB. This will download gigabytes worth of device support which you will never use.

Download Only the Necessary Files

Since you are going to use only the CPLD chip, it is better to download only the necessary files.

1. Choose the *Individual Files* tab as shown in Figure 2.

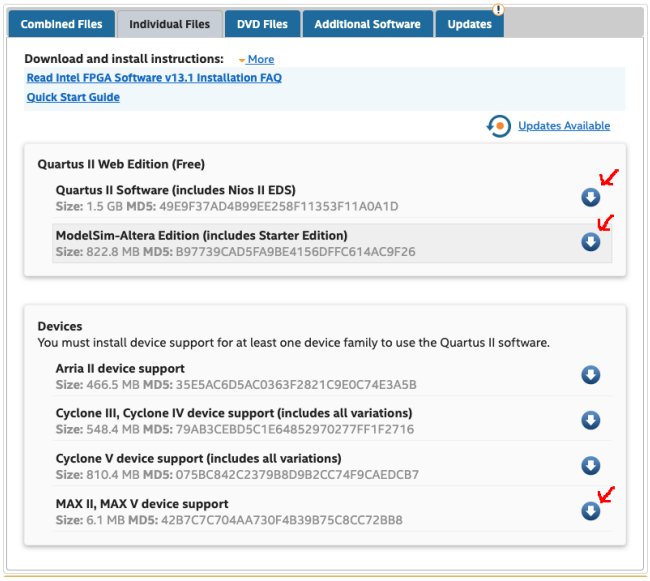

Figure 2. *Download the marked files.*

2. Download Quartus II Software
3. ModelSim-Altera Edition
4. Download MAX II device support for using EPM240. Do not download any other files.

4. You will be asked to enter your password if you have not logged in. And you may have to create an account if you have not done so. If everything is fine, you will download less than 2.6 GB instead of 4.4 GB for the full package. You should have:

File	Size
QuartusSetupWeb-13.1.0.162.exe	1.58 GB
ModelSimSetup-13.1.0.162.exe	864 MB
max_web-13.1.0.162.qdz	6.4 MB

Installing Quartus

For installation, double-click on the *QuartusLiteSetup* executable file and choose the default values.

Figure 3. *Quartus II Web Edition splash screen.*

3. Designing with Quartus

Before you can start experimenting, you should understand how Quartus does its work. Figure 4 gives a overview. Note that Quartus software can do much more than what is shown here.

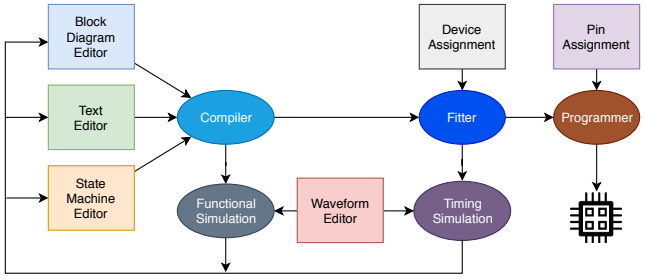

Figure 4. *Design flow using Quartus.*

The square boxes are Quartus editor or dialog boxes which you can use to enter designs or settings. They are:

- **Block diagram editor** for design entry using gates and logic blocks
- **Text editor** for design entry using hardware design language, usually Verilog
- **State machine editor** for entering a state machine
- **Waveform editor** for preparing an input waveform before simulation
- **Device assignment** is a simple dialog box to specify the target device
- **Pin assignment** is a dialog box for setting the input/output pins on the target device

The ovals are internal Quartus operations that execute commands:

- **Compiler** converts a design into a *netlist*
- **Fitter** configures the internal wiring on a target device to the netlist to obtain a *bitstream*
- **Functional simulator** performs *device-independent* simulation based on logic function only without considering the physical device aspects
- **Timing simulator** performs *device-specific* simulation which includes delay estimation. This simulation is slower than functional so it is usually done after a circuit has passed functional simulation.
- **Programmer** download the bitstream into the target device

For simple experiments with digital logic, only two boxes are used: the Block Diagram Editor and the Waveform Editor.

4. Example Design

This tutorial uses a simple circuit to check out the Quartus installation.

A combinational logic circuit shows the results for three people who vote. The three people are labeled as ***A***, ***B***, and ***C***. If at least two of them vote yes, then the output V becomes HIGH. Else if only one or none of them vote yes, the output becomes LOW.

The truth table of the problem is given in Table 2.

Table 2. *Voter truth table.*

Input			Output
A	***B***	***C***	***V***
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

There are several possible Boolean equations that can be derived. After simplifying the equation for ***V***, we get

$$V = AB + BC + AC$$

We will now enter this circuit in Quartus and simulate its function.

5. Specifying Project Settings

1. Launch Quartus by double-clicking on the icon.

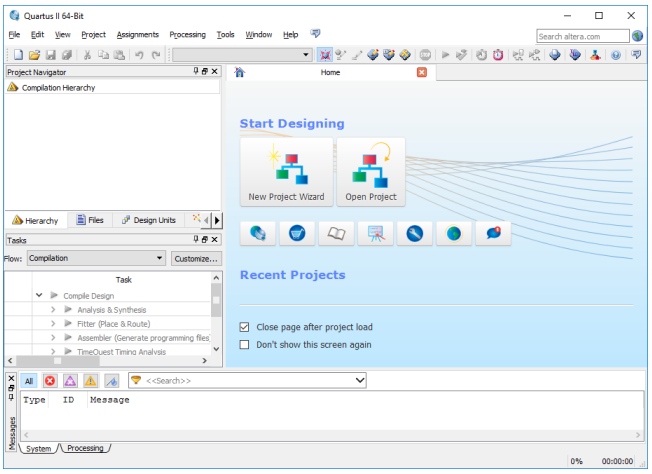

Figure 5. *Quartus II Web Edition welcome screen.*

2. From the top menu, choose **File ➥ New Project Wizard**. This will bring us to the introduction page.

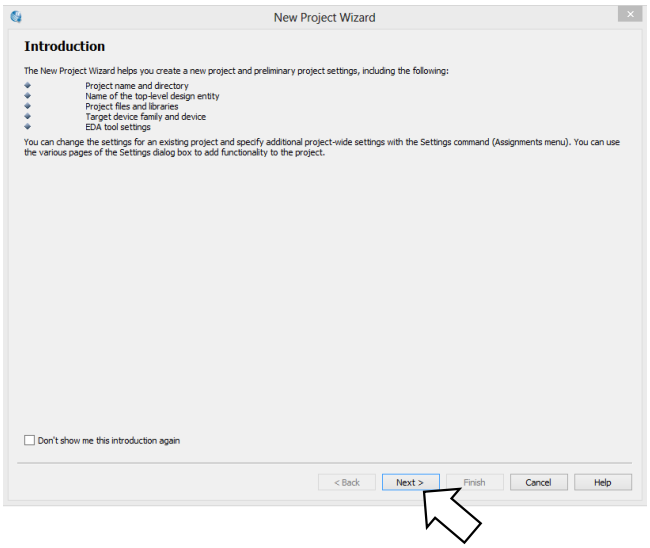

Figure 6. *New project Introduction page.*

3. Click Next >. This brings us to page 1.

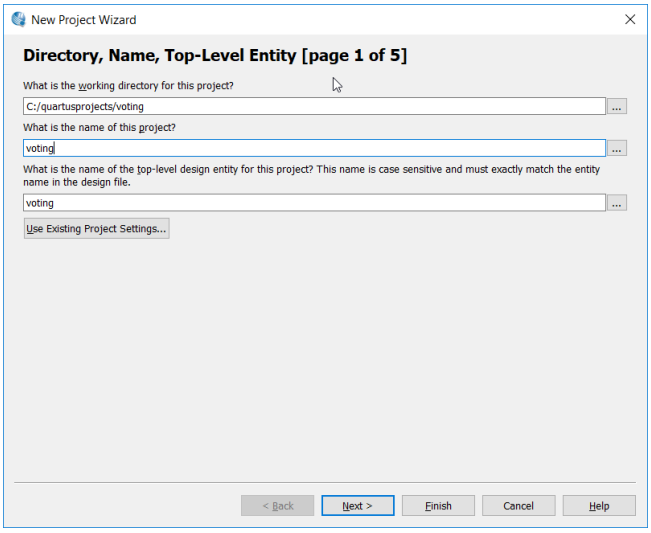

Figure 7. *Quartus II Web Edition welcome screen.*

Working directory is the location of your project files. *We strongly suggest to change the directory settings* to avoid problems if you have to reinstall Quartus and to make it easy for backing up your data. You can create a working directory at the root level of your hard disk (e.g. C:\quartusprojects) or inside your documents folder. Once you have decided on the working directory for all your Quartus projects, create a subdirectory for your current design (e.g. Voting).

Name is the name of your project. Simply use the same name at the design directory.

Top level entity specifies the master file in your design. Unless you have a specific reason, just use the project name.

4. Click Finish. Notice the voting entity as shown in Figure 8.

The Quartus program has 4 windows:

- **Project navigator**: displays the design files in your working directory and their relationships
- **Tasks**: shows the progress of design tasks
- **Workspace**: is the area where you enter the design
- **Messages**: displays messages from the tasks e.g. compiler, fitter, simulator

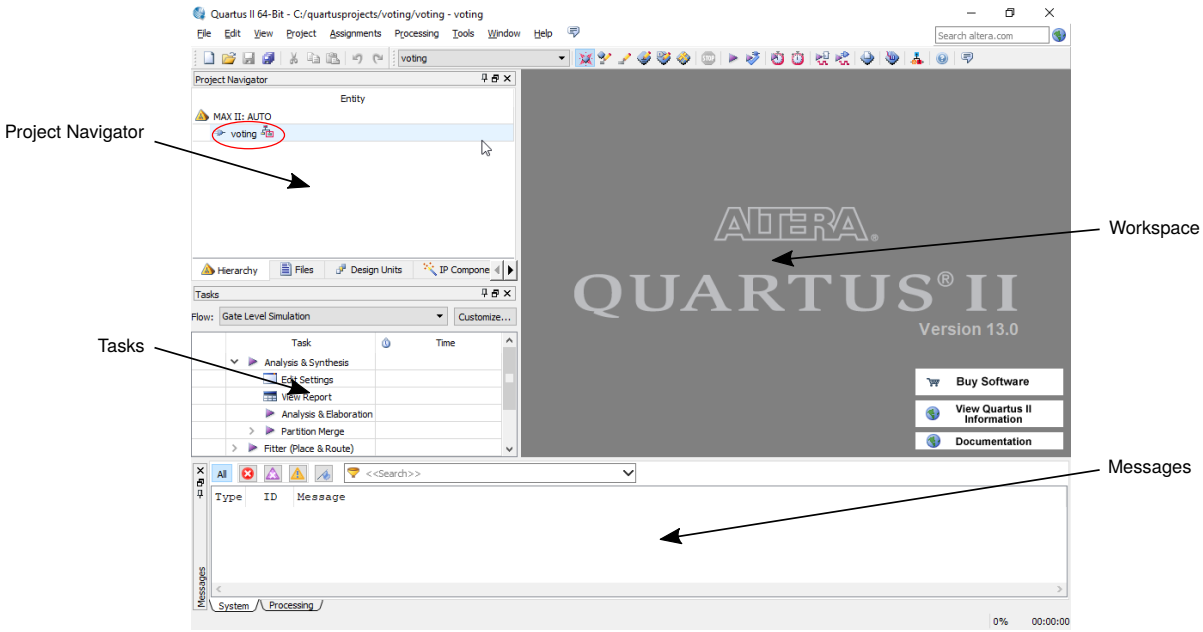

Figure 8. *Quartus layout.*

6. Schematic Entry

In the design entry step you create a schematic or Block Design File (.bdf) that is the top-level design.

1. From the top menu, choose **File ➥ New ➥ Block Diagram/Schematic File** to create a new file (see Figure 9) then click OK.

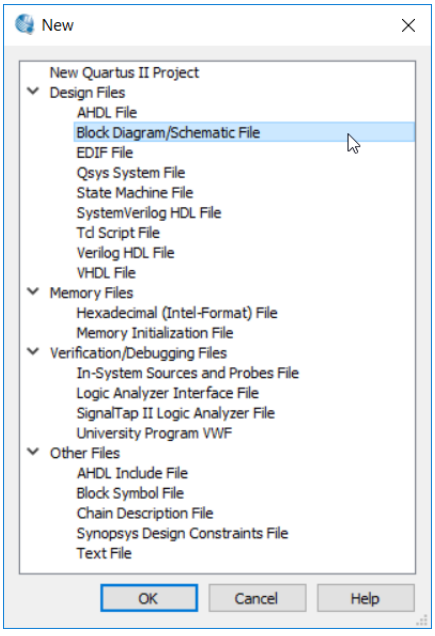

Figure 9. *New BDF.*

2. The schematic entry window will appear on the working space as shown.

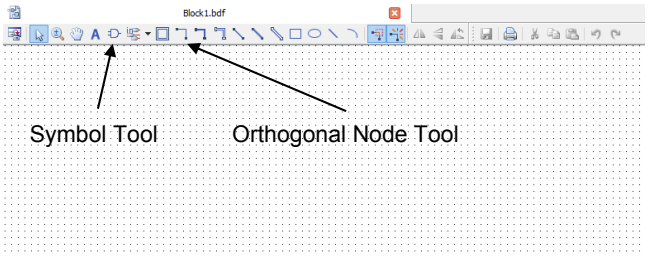

Figure 10. *Quartus schematic editor.*

3. Click Symbol Tool. A pop-up dialog box will appear as shown in Figure 11.

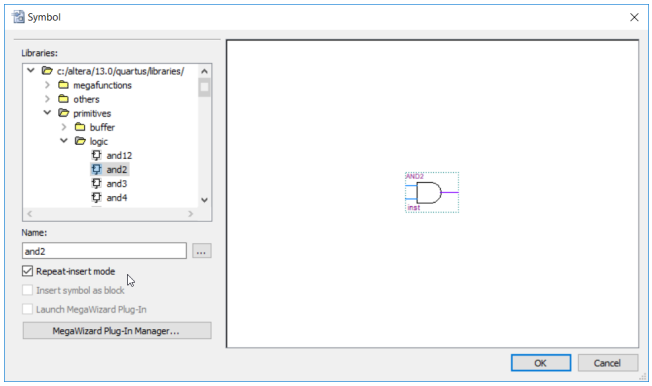

Figure 11. *Add new symbol.*

4. Expand `c:/altera/13.0/quartus/libraries`, expand `primitives` followed `logic`.

5. Select `and2` and then check Repeat-insert mode and then click OK.

6. Place the three (3) `and2` gates as shown in Figure 12 by clicking on the desired placing area and then press the ESC keyboard key to return back to the default arrow cursor.

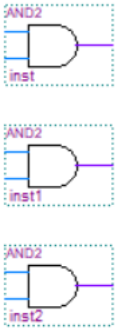

Figure 12. *Add 3 and2 gates.*

7. Another way to add symbols is by clicking anywhere on the schematic editor working space. The Symbol dialog box similar to Figure 11 will appear. At this time, insert an `or3` gate as shown in Figure 13.

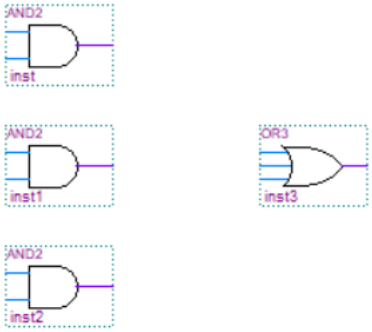

Figure 13. *Add or3 gate.*

8. Next we will add the input/output pins. In the Symbol dialog box, expand `c:/altera/13.0/quartus/libraries`, expand `primitives` followed by `pin`.

9. Select `input` then click OK.

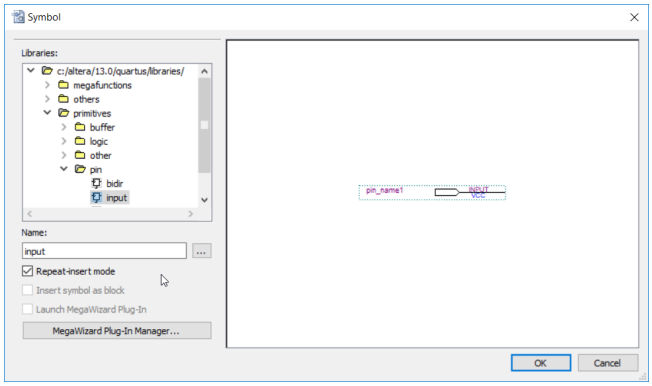

Figure 14. *Adding an `input` pin.*

10. Place the three (3) unit **input** pins.

11. Repeat step 9 and 10 for an **output** pin. By now we should have our design entry such shown in Figure 15.

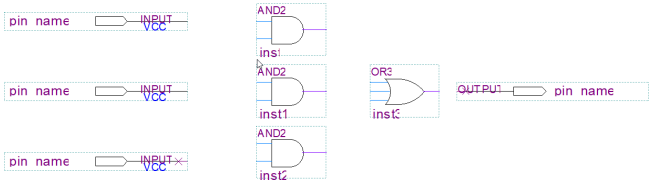

Figure 15. *All gates entered.*

12. Right-click on the top most input pin and click on Properties. We should see a Pin Properties pop-up dialog box (Figure 16).

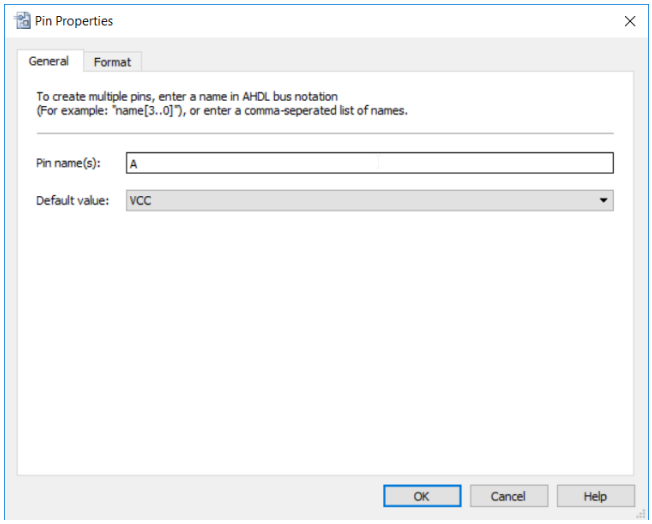

Figure 16. *Pin Properties dialog box.*

13. Rename the pin name as ***A***. By default, the default value will set to VCC if not, select VCC as shown in Figure 16. Click OK.

14. Repeat the Step 13 for pin ***B*** and ***C*** and output pin ***V***

15. Using the Orthogonal Node Tools on the schematic toolbars Make the circuit connection for the Boolean equation. We should see the schematics similar to Figure 17.

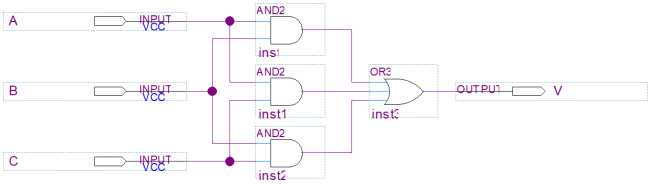

Figure 17. *All pins renamed, and gates connected.*

16. From the top menu, choose **File ➥ Save As...**

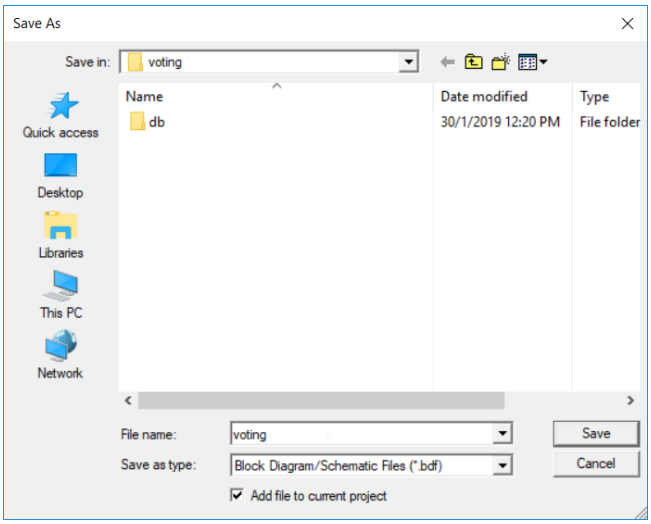

Figure 18. *Save the file.*

18. From the top menu, choose **Processing ➥ Start Compilation** . We can also simply click on Start Compilation button on the toolbar. The compilation report is shown in Figure 19. For this tutorial, ignore all warnings. Click OK.

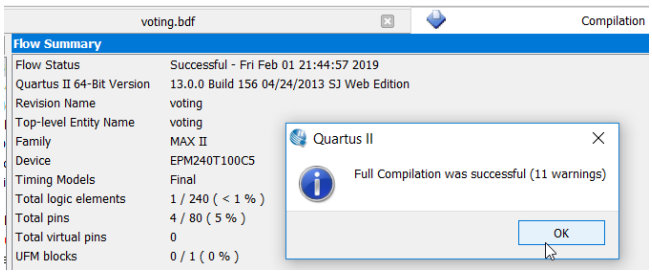

Figure 19. *Compilation success.*

7. Simulation

In order to simulate the design, we will enter the input simulation waveform.

1. Choose **File ➥ New ➥ University Program VWF** to create a new file (see Figure 20) then click OK.

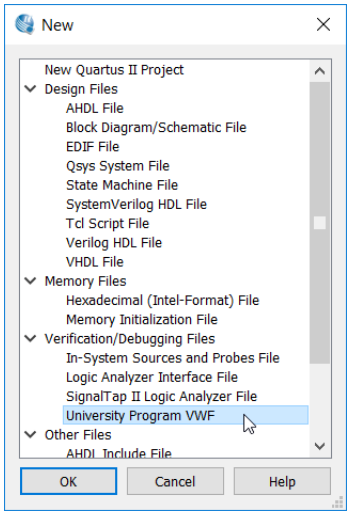

Figure 20. *Create new waveform file.*

2. Simulation waveform editor window will appear such shown in Figure 21.

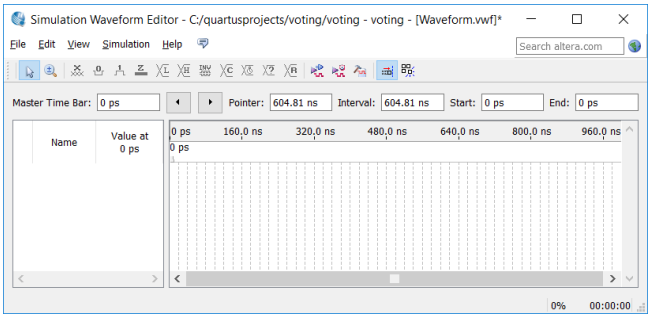

Figure 21. *Blank waveform.*

3. Go to **View ➥ Fit in Window** to get the Vector Waveform at the workspace. We can also adjust the view at our convenience through Zoom In and Zoom Out tool.

4. Go to **Edit ➥ insert** and click insert node or bus. A pop-up dialog box will appear as shown in Figure 22.

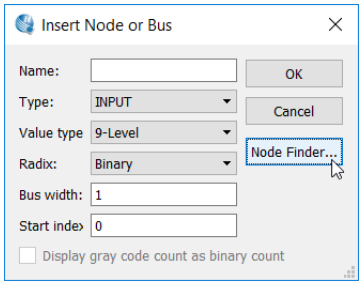

Figure 22. *Insert node dialog.*

5. Click Node Finder.. . A pop-up box will appear as shown in Figure 23.

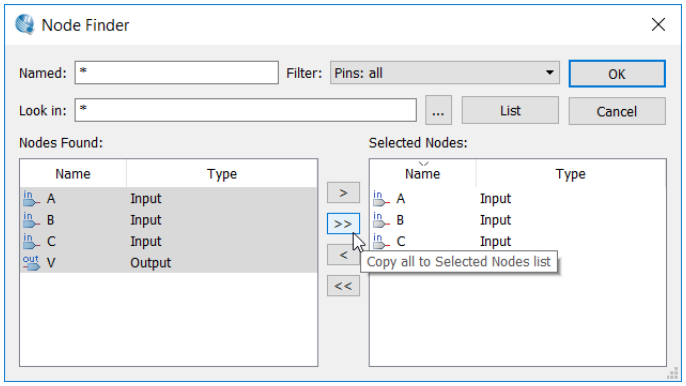

Figure 23. *Node finder dialog.*

6. Follow these steps:

   (a) Set the Filter to Pins:all
   (b) Click List
   (c) Click the >> button
   (d) Click OK on the Node Finder window

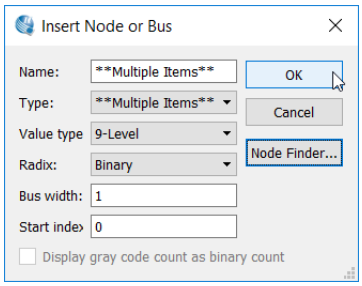

Figure 24. *Adding multiple nodes.*

7. We will be brought back at the Insert Node or Bus box as shown in Figure 24, click OK. By stage we will see voting inputs ***A***, ***B*** and ***C*** all goes LOW at time 0ns to 100ns while voting output "V" is the output to be determined as shown in Figure 25.

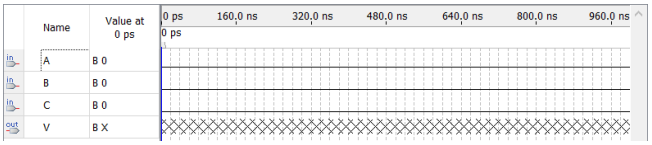

Figure 25. *Waveform after inserting nodes.*

8. Highlight ***A*** by clicking on the input port symbol of the port name (Figure 26.

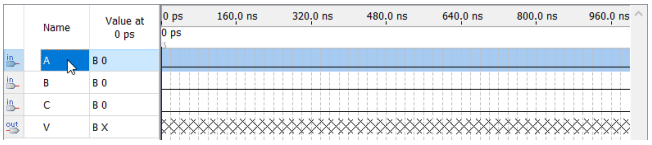

Figure 26. *Waveform after selecting input A.*

9. Click the Overwrite Clock icon on the toolbar and set the period of signal ***A*** to 1000 ns.

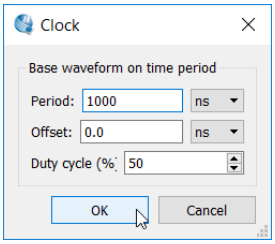

Figure 27. *Set period of B to 1000 ns.*

10. Click input port symbol of input ***B*** to highlight the whole frame of input ***B*** as shown in Figure 28.

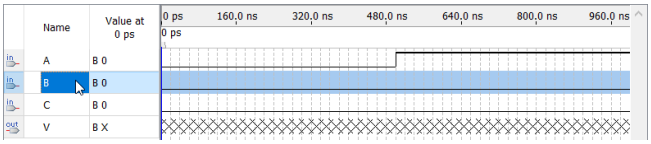

Figure 28. *Waveform after selecting input B.*

10. Next, click overwrite clock icon on the toolbar. pop-up window will appear as shown in Figure 29 fill in period 500ns then click OK.

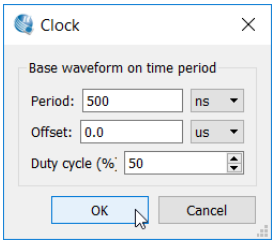

Figure 29. *Set period of B to 500 ns.*

11. Click port symbol of input ***C*** to highlight the whole frame of the input ***C*** as shown in Figure 30.

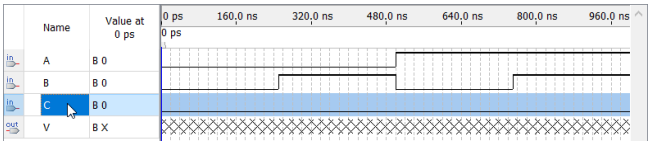

Figure 30. *Waveform after selecting input C.*

12. Next, click a pop-up box will appear as shown in Figure 31. Fill -up transition occurrence to count every 125ns, click OK.

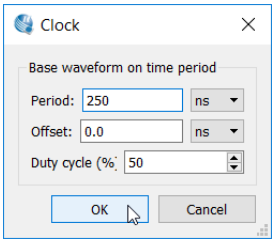

Figure 31. *Set count to 125 ns.*

13. Go to File click Save as. A pop-up box will appear as shown in Figure 32, click Save

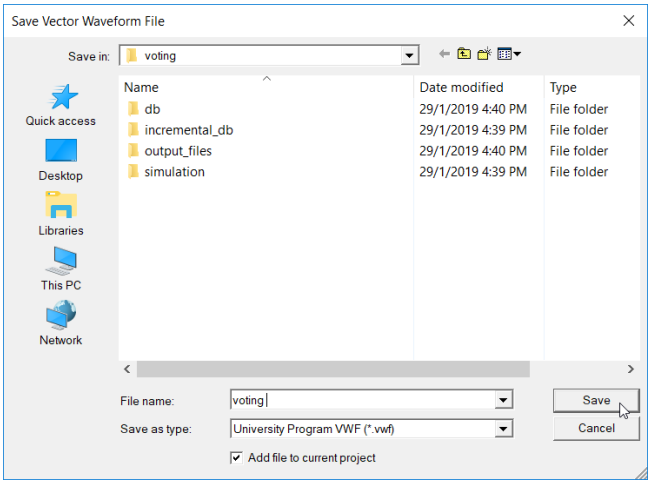

Figure 32. *Save the vector waveform.*

14. Click run functional simulation icon on the toolbar. If the simulation is successful, a new window will appear showing the result of simulated waveforms.

15. With corresponding input ***A***, ***B*** and ***C***, notice that ***V*** is responding in accordance to our Boolean equation. Therefore, out design is now functional verified.

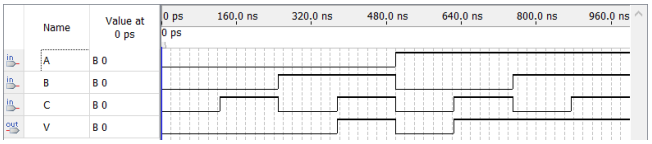

Figure 33. *Functional simulation output.*

16. Next, click run timing simulation icon on the toolbar the result as indicated in Figure 34. For simple simulations such as this circuit, timing requirements is not important. However, timing requirements are critical in more complex designs.

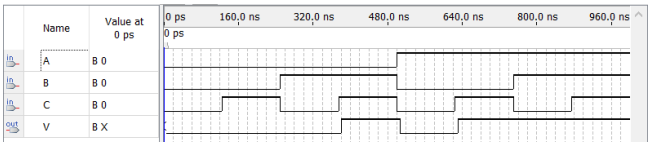

Figure 34. *Timing simulation output.*

If we look closely or if we zoom in, we can see the difference at the ***V*** output. In the functional simulation output, ***V*** changes immediately after receiving input changes. In the timing simulation, there is some delay.

Acknowledgments

Thanks to Siti Nursyuhada binti Mahsahirun and Zulkifli Md. Yusof, both of Faculty of Manufacturing, Universiti Malaysia Pahang.

References

[1] Munim Zabidi, Izam Kamisian, and Ismahani Ismail. *The Art of Digital Design*. 2019.

[2] *Introduction to the Quartus® II Software*. Version 10. Altera. 2010. URL: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/manual/intro_to_quartus2.pdf.

[3] *My First FPGA Design Tutorial*. Altera. July 2008. URL: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/tt/tt_my_first_fpga.pdf.