Report and Video Guidelines

Cover Pages and Templates

Each submission *must* include the following on a separate cover page: the course number and name, the lab or assignment number and title, your name, your BCIT ID, your lab set, and the date the document was created.

You may find it helpful to create a document template that you can re-use for future submissions. One is available on the course web site.

Errors

Marks will be deducted for errors, ambiguities or lack of clarity in your report. Common errors include:

- Values without units.
- Missing, incorrect, or redundant titles or headings.
- Ambiguous or unclear text due to poor grammar or punctuation.
- Screen captures showing too little (e.g. missing scales) or too much (e.g. the whole screen).

Program Listings

You will need to include Verilog code in most lab reports. Do *not* paste a screen capture. You *must* include your code as text using a monospaced font with no extra space between lines.

You can do this by applying a style to your text. Paste your code into your report and select it. In Microsoft Word select **Apply styles...** from the **Styles** gallery, enter `HTML Preformatted` in the text box and click on **Apply**. In LibreOffice enter `Preformatted Text` in the style text box. Note that you can enter these style names even if they are not one of the choices in the drop-down box.

Syntax Highlighting (Optional)

You can install an editor such as VSCode along with a Verilog extension such as TerosHDL to auto-indent and add syntax highlighting to your code. Open your System Verilog file, right-click and select "Format Document," then copy-and-paste the formatted text into your word processor or back into Quartus.

Another option is Notepad++. Install with the *Custom* install so you can include the **NppExport** plugin. Open your file with Notepad++, select the text and then click on **Plugins** > **NppExport** > **Copy RTF to clipboard** and paste your code into your word processor.

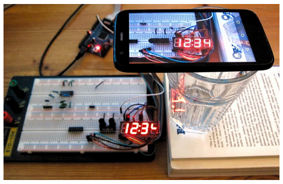
Auto-Indenting Code

An AI chatbot such as ChatGPT or the form on the course web site at **Course Information** > **indent** can indent your code.

Videos

Videos of your lab demonstrations must be viewable on a Windows web browser. Text or digits in the video must be upright.

You can put a phone on top of books or a glass to hold it steady while you record your demonstration...



Re-Encoding Videos

If your video is too large or doesn't display in a browser, you can use Handbrake or Avidemux to reduce its size and convert it to a compatible format:

- With Handbrake, set "Preset:" to "Fast 720p30," set the "Save As:" location and press "Start Encode."
- With Avidemux select (1) Video Output as "MPEG4 AVC (x264)" and (2) Output Format as "MP4 Muxer" and select File/Save. If necessary, pressing the "Configure" and "Filter" buttons will display options to reduce the file size or use Transform / Rotate to make sure things are

upright. You can also change the start (A) and end (B) of the video.

When you're done, check the file size. Drag the file onto an open browser window and check that the file plays in the browser, rather than in in another application such as Windows Media Player or "Films and TV".

Check again that your video it plays in your (Windows) browser and that any digits are upright after submitting it.

Units, Notation and Significant Figures

Numerical results without units are incomplete and will be marked incorrect.

Use SI units and engineering notation. For example, 1.2×10^{-5} F should be written as 12 μF.

You should be familiar with Significant Figures. For example, if your measurements have three significant figures don't give answers with 10 decimal places.

Diagrams

Diagrams must be legible. They may be drawn by hand or created with a drawing or CAD program. Unless otherwise specified, diagrams must be drawn by the student rather than generated by Quartus.

Block Diagrams

You will be required to convert between HDL descriptions and block diagrams. These diagrams are similar to schematics but the blocks represent logic functions rather than components. Inputs should be on the left and outputs on the right.

Ports are symbols for module inputs and outputs. Label with the port signal names.

Buses are multi-bit signals. Draw a slash (/) over the line and write the bus width in bits or the bit indices.

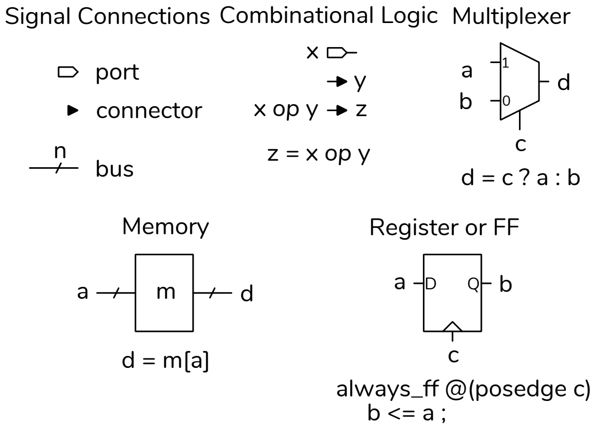
Combinational Logic Use Verilog expressions involving port and other signal names rather than schematic symbols.

Multiplexers correspond to conditional operators. Label the inputs 1 and 0 for true and false.

Memories These function as look-up tables and correspond to HDL arrays. Use a rectangle with the input on the left and the output on the right.

Registers The output of the register or flip-flop is the signal assigned to in an **`always_ff`** statement.

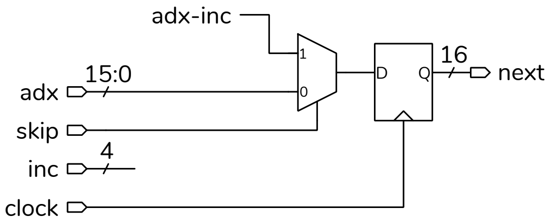
Use symbols similar to the following:



For example, the following Verilog code:

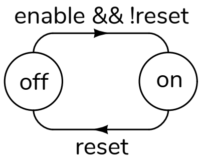
```
module ex61 ( input logic [15:0] adx,
              input logic [3:0] inc,
              output logic [15:0] next,
              input logic skip, clock ) ;

   always_ff @(posedge clock)
     next <= skip ? adx - inc : adx ;

endmodule
```

would be drawn as follows:



State Transition Diagrams

State transition diagrams should show state names (or values) in circles or boxes. State transitions should be shown as arrows labelled with unambiguous logical expressions for the corresponding state transition. An example is:



There are many diagram-drawing programs such as LibreOffice Draw, and diagrams.net as well as online schematic editors provided by vendors such as Scheme-it or EasyEDA. However, if you're not already familiar with using a drawing or schematic capture program it may be faster to draw the diagrams by hand.