Grading Summary for ECE 2031 Lab Reports

Summary of Lab Reports

Lab reports for ECE 2031 are collections of images, tables, and code that were produced during the design, implementation, and test of digital hardware. Each result is formatted and captioned as if it were to be included in a full technical lab report, but the reports do *not* include the paragraph-style descriptions of background, procedure, or conclusions that such a document would normally contain. The purpose of the report is

1. to show that you completed the lab steps,
2. to show that you understand what was done and its context within digital hardware engineering,
3. to practice effectively capturing technical content for presentation to others, and
4. to practice effectively summarizing technical information.

Each lab's report is scored out of 50 points based on:

- Accurately describing the presented results.
- Clearly and concisely presenting the information.
- Adhering to the document formatting requirements.

These categories reflect the fact that engineers and scientists strive for more than just accuracy; we also ensure that information is presented to our audience as clearly and effectively as possible.

Lab Report Grading

The lab reports are evaluated in the following categories:

Technical Accuracy and Completeness

Technical errors in the lab report, both in the figures and the captions, will count in this category. For example, describing some assembly code as VHDL is technically incorrect. Presenting a schematic that should have pin annotations but does not is technically incomplete.

Information Effectiveness

Technical communication has a strong emphasis on the clear presentation of information. For ECE 2031 lab reports, this category requires the following:

- Present graphical information clearly.
- Include enough information in the caption for the user to understand the purpose of the result.
- Keep the caption concise and clear.

Document Formatting

All technical documents have some amount of formatting specification. The formatting specification for ECE 2031 lab reports follows a subset of IEEE formatting guidelines. Formatting requirements for lab reports are detailed later in this document.

Missing Results

You should submit results for everything that you complete in each lab (according to the “capture this” boxes in the lab manual). If the report does not include a result a result that is expected based on your check-offs, a 10-point penalty will be applied. However, there is no requirement to complete all of a lab; again, you should submit results for everything that you *do* complete in each lab. This will not immediately penalize the lab report, because a result is only considered ‘missing’ if you did complete that part of the lab but did not include a result in the report. However, a lab report cannot be graded without some results to evaluate, so if the report contains fewer than three results, the maximum score will be reduced to 40 if the report only contains two results, 20 if the report only contains one result, or 0 if the report contains no results.

Tips for Writing Effective Captions/Titles

This information often refers to the document's user instead of reader because technical documents are functional objects. You are designing and producing something that has utility, and like with anything that you create as an engineer or computer scientist, your goal is to maximize the utility for the user.
Something that greatly increases the utility of a document containing figures is to caption those figures with relevant and helpful information.

- The purpose of a figure caption is to provide the user with useful information and context for the figure.
- Each caption should be self-contained; i.e. the reader should be able to understand the figure and caption without referring to something else in the document and especially without any external information. This does not mean that other parts of the document cannot be mentioned, but that the caption must still be sufficient if that reference was removed.
- Every figure caption in a document will be unique, because its purpose is to provide information specifically about that figure.
- There is more than one way to effectively describe a figure.

For the lab reports in this class, keep in mind that even though you are not required to write a full report (with a procedure, etc.), you are writing captions as if the figures were in such a document. That requires that you think about what a full report would contain and how the current figure fits into that.

Included here in quotes are example snippets, ***not* full captions**.

For All Results

1. Before writing a caption, consider the purpose of the result in the larger context of the lab and course.
    a. What is the theme of the lab?
    b. How do the lab steps that created this result fit into that theme?
2. Describe the result in a way that highlights its purpose within the overall lab and report.
    a. How does this result, rather than anything else that could have been captured, show what was done in the lab procedure?
    b. How does this result differ from other, similar results in the report?
    c. What are the important aspects of this result? What should the user pay attention to?

Circuit Schematics

1. If you built the circuit to perform some function or task, include it in the caption
    a. If it is an arbitrary logic function, include the expression; e.g. "Y =/A+/B·C." For hardware-oriented Boolean expressions (which is usually the case for this class), use "+" for OR, "·" for AND, and overbar (or preceding "/" if unable to create overbar) for NOT.
    b. If it was created to serve some purpose, concisely describe it; e.g. "Output 'happy' is active when either input 'playing' or 'eating' is active," or "4-bit full adder"
2. If there is a particularly important feature of the schematic, include it in the caption
    a. "Showing the path with worst-case propagation delay"
    b. "with clock c5 attached to device dff3"

What to **AVOID**

1. Naming every component or listing every detail of the schematic
    - a. Unless it is relevant to the result, avoid anything like “with two AND gates and three NOT gates”
    - b. State machine schematics should not include the Boolean expression(s) of next-state logic unless there is something notable about those expressions.
2. Relying on other figures or documents for descriptions to make sense
    - a. E.g. “Schematic circuit for previous truth table” or “Lab 1 circuit schematic”

Oscilloscope Captures

1. Explain where the signals come from.
    - a. “breadboard circuit using HCT chips”
    - b. “unknown circuit in <filename> running on a DE10-Standard board” [note: referring to an external file is acceptable as long as the caption would still make sense without it]
2. If relevant, describe what type of signal is shown
    - a. “A square/sine/sawtooth wave”
    - b. “Clock”, “data”, “input”, “output”, etc.
3. If measuring something, describe what is being measured
    - a. “The rise time of”
    - b. “The period of”
4. If a required parameter is not immediately apparent in the capture, include its value in your caption
    - a. “duty cycle of 33%”
5. If you have multiple signals, differentiate between them by color/position
    - a. “V1 (dark blue) and V2 (light blue)”
    - b. “Input (top) and output (bottom)”

Tables

1. Describe the table contents. If it's a truth table, explain the logic or purpose
    - a. “Truth Table of Y =/A+/B·C”
    - b. “Propagation Delays of Texas Instruments ICs”
2. Include any special characteristics
    - a. “with Highest Propagation Delays Highlighted”

Simulation Waveforms and Logic Analyzer Captures

1. Describe what is being simulated or captured
    - a. “Circuit implementing …”
    - b. “Decoder for I/O devices”
2. Include important features of the simulation
    - a. “for all possible input combinations”
    - b. “receiving data values 0x34 and 0x22”
    - c. “showing expected final value of 0xFFFF”
    - d. “with decoded opcode values for the first three instructions of *example.asm*”

Appendices

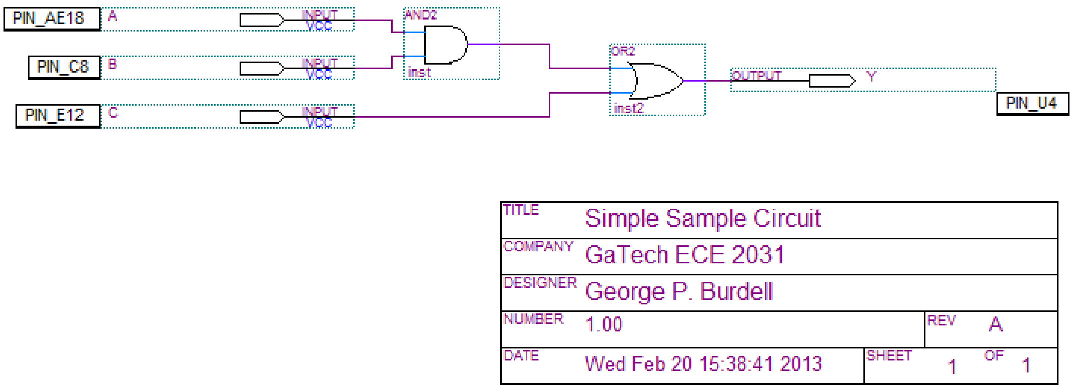
1. Describe what is included in the appendix
    - a. “Assembly code to shift…”
    - b. “VHDL implementing a…”

Lab Report Formatting Requirements

Overall Document Formatting

- Results must be in the order listed in the content requirements, with the exception that any appendices always go at the end of the report. Again, this matches what would be a full lab report document, just without the body text.
- If a formatting detail is not included here, choose something reasonable (or consult with an instructor). For example, use a typical font such as Calibri or Times New Roman, and use the typical size of 11-12.
- Figures and tables are enumerated independently (i.e., there can be a Figure 1 and a Table 1).

Image/Figure Formatting



TITLE	Simple Sample Circuit		
COMPANY	GaTech ECE 2031		
DESIGNER	George P. Burdell		
NUMBER	1.00	REV	A
DATE	Wed Feb 20 15:38:41 2013	SHEET	1 OF 1

Figure 1. Schematic with FPGA pin assignments of circuit implementing $Y = (A \cdot B) + C$.

Bold

Period

Figure and label are centered on page*

Period at end

Capitalize initial letter of first word, proper nouns, and acronyms

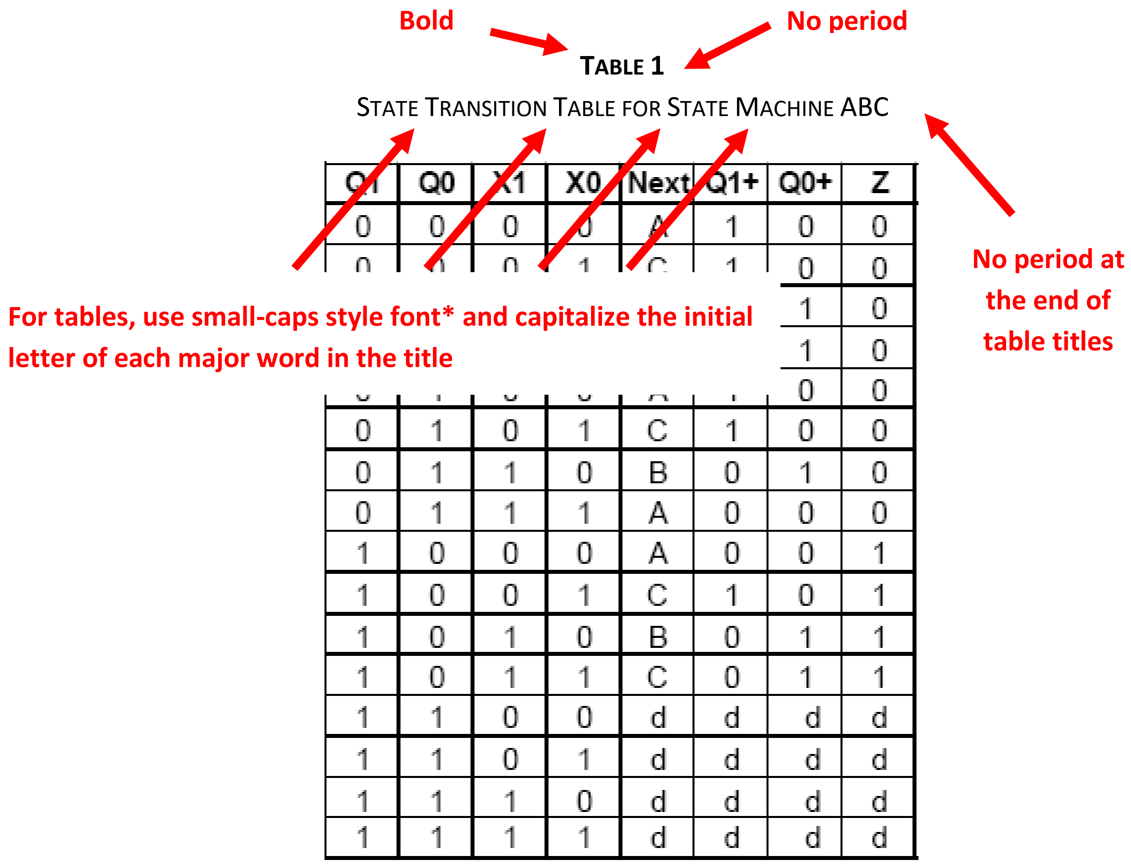
*** Labels that fit on one line are centered. Labels that require two or more lines are left-aligned. This is common in IEEE formatting requirements.**

Code Formatting

- Code and other text that fits on one page can be formatted as a figure. Longer code should be included in an appendix.
- Code should always be in a fixed-width font and single-spaced. Note that some document editors (like Word) default to having extra space between paragraphs, which should be removed (or the text converted to use line breaks instead of paragraph breaks).

Table Formatting

Table titles are always centered. If more than one line is needed, keep each line approximately the same length by manually inserting line breaks.



Bold

No period

Table 1

State Transition Table for State Machine ABC

Q1	Q0	X1	X0	Next	Q1+	Q0+	Z
0	0	0	0	A	1	0	0
0	0	0	1	C	1	0	0
						1	0
						1	0
0	1	0	0	A	1	0	0
0	1	0	1	C	1	0	0
0	1	1	0	B	0	1	0
0	1	1	1	A	0	0	0
1	0	0	0	A	0	0	1
1	0	0	1	C	1	0	1
1	0	1	0	B	0	1	1
1	0	1	1	C	0	1	1
1	1	0	0	d	d	d	d
1	1	0	1	d	d	d	d
1	1	1	0	d	d	d	d
1	1	1	1	d	d	d	d

For tables, use small-caps style font* and capitalize the initial letter of each major word in the title

No period at the end of table titles

*** “Small caps” is a text style that most document editors, including Word, can apply. Search for instructions for your chosen document editor if you are unfamiliar with it.**

Tables do not have a caption underneath them like figures do. The table title describes the table contents.

Appendix Formatting

- Appendices always go at the end of a document. See example at the end of this document.
- Appendices have a cover page with the appendix enumerator and description centered vertically and horizontally on the page, where the enumeration is 'A', 'B', 'C', etc.

Common Lab Report Errors

Common Effectiveness Errors

Vague or incomplete title or caption.
Caption should describe the content and purpose the result, and contain enough information to be understood by itself. See caption tips earlier in this document for examples.

Spelling and grammar errors or ineffective wording.
Proofread your document for mechanical errors as well as word-choice and sentence structure. Use technical words correctly. Use the simplest wording that conveys the required information.

Failure to crop out unnecessary or useless areas from a screen capture or figure, such as toolbars, scroll bars, and empty space.
Include only relevant information. Put thought into the effective presentation of visual information.

Stretched or skewed images.
Preserve the original aspect ratio of technical images.

Graphics or in-image labels too small.
Ensure that all necessary information in a figure is legible. This could result in technical errors if required information is no longer discernible.

Common Formatting Errors

Improper capitalization in title/caption.

Missing or incorrect use of punctuation in title/caption.

Missing or incorrect use of bold.

Code not in fixed-width font, or with extra line spaces.

Document, page(s), or items in incorrect location or order.

Common Technical Errors

Describing figures in technically-incorrect ways, such as referring to assembly code as VHDL.

Missing title box on schematics. Schematics should always have a title block, and unless you are specifically capturing a cropped area of the schematic, the title block should be included in the capture.

Incomplete or ineffective comments in code, including missing header lines.

Additional Examples of Lab Results

Example with many errors

The following figure is the same circuit as in Figure 1, but with many errors.

fig 4 - step 5 state machine Resoult

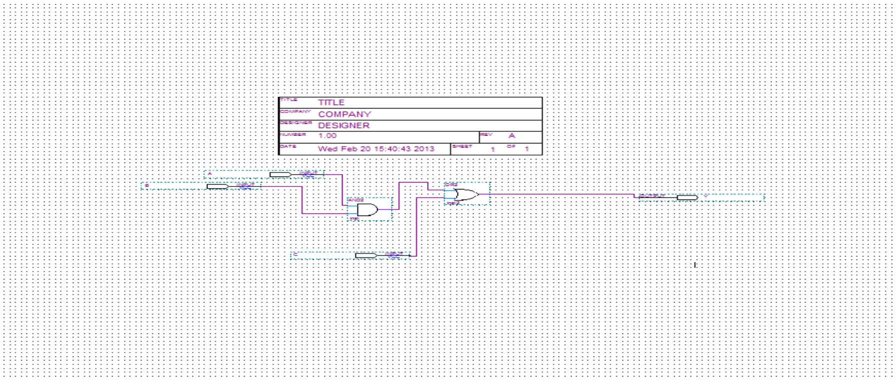


Figure formatting errors:

1. Stretched horizontally
2. Not centered on page

Figure effectiveness errors:

1. Improper cropping
2. Grid in background (dots)
3. Too small to read
4. Title block in wrong place
5. Excessively messy layout

Figure technical errors:

1. Title block not filled out
2. No pin numbers (if required for result)

Caption formatting errors:

1. In wrong place – above image and not centered
2. ‘Figure’ not bold, capitalized, or spelled out
3. Figure number out of order (should be 2 in this document)
4. No period after ‘4’ (hyphen used instead)
5. ‘Step’ not capitalized and 'Result' unnecessarily capitalized.
6. Spelling error
7. No period at end

Caption effectiveness errors:

1. Does not describe result content or purpose
2. Is not self-contained (requires lab manual to make sense)

Caption technical errors:

1. Calls the combinational circuit a state machine.

Landscape pages within a portrait document are stapled here; i.e. this page gets rotated clockwise when read. Fully-landscape documents are stapled in the top-left.

Example of proper landscape page

You can orient your figures using portrait or landscape. Use landscape if the image is too large to fit in portrait and remain legible. When including a landscaped figure with other results, label the figure so that the caption is on the right-hand side of the page when the page is vertical. This configuration is standard and allows the reader to turn the previous pages away from them instead of towards them.

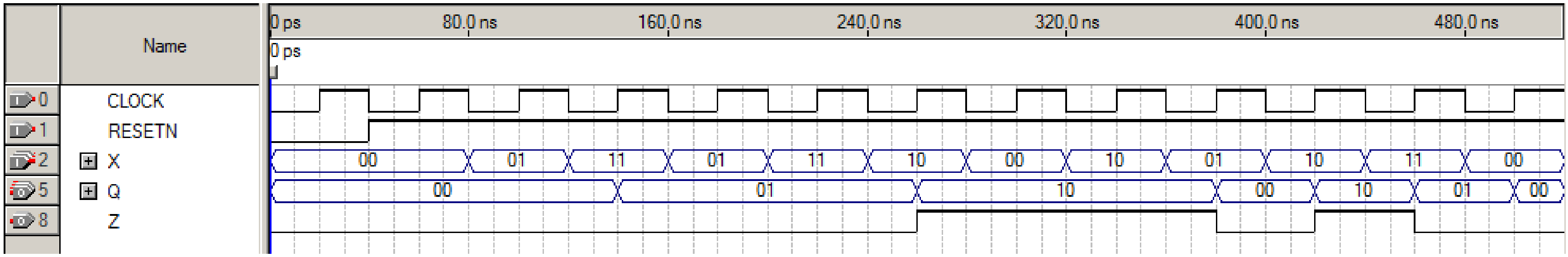


Figure 3. Functional simulation waveform of state machine with three states, 2-bit input X, and 1-bit output Z. The input vector provides 100% coverage of state transitions to prove correct behavior of the state machine, including assertion of Z in state '10'.

If producing a document intended to be viewed electronically (which is true of your lab reports), ensure that any landscape pages are properly oriented in the pdf.

A document may contain landscape pages, but the document overall must be portrait. It is not acceptable to create a document that is entirely landscape.

Example of code less than one page in length

Figure 4 is an example of properly formatted code that spans **less than a page**. Always place header comments in the first lines, as shown below. Be sure to use a `fixed-width font` such as `Courier New` for code, and remove extra space between lines (Microsoft Word defaults to extra line space for newlines)

```
-- ORGATE.VHD (VHDL)
-- This code produces a negative-logic OR circuit
-- George P. Burdell
-- ECE2031 L01
-- 01/31/2009

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;

ENTITY orgate IS
PORT(
      PB1, PB2 : IN STD_LOGIC;
      LED : OUT STD_LOGIC);
END orgate;

ARCHITECTURE a OF orgate IS
BEGIN
      LED <= NOT(NOT PB1 OR NOT PB2);
END a;
```

Code always contains header lines

Code that is less than one page is formatted as a figure

Figure 4. VHDL code used to turn off an LED when either of two active-low pushbuttons are pressed.

Example of code that spans more than one page

Appendix enumerator on its own line

Appendix A

Example Long Code Implementing a PS/2 Keyboard Interface

Everything in small-caps font with initial letters capitalized

When code is longer than one page, treat it as an appendix by creating a cover page (like this one). The appendix title includes an enumerator (A, B, etc.) and a title that describes the contents of the appendix. Then include all of the pages of code after the cover sheet. If you have multiple sets of code, you would have several Appendices (A, B, C, etc.).

The title on the appendix cover page should be centered as shown above.

Appendices ALWAYS go at the end of the document. That is why they are called appendices (they are "appended" to the document).

Remember to use fixed-width font for code and to remove extra space between lines.

Code should always be commented so that the reader understands how the code works.

Do not allow the document editor to wrap lines of code, because at best it makes the code difficult to read, and at worst, produces invalid code (for example, if a comment wraps to a new line, the second line might be interpreted as a separate, invalid, line of code). Adjust indention sizes, line lengths, and/or font size as needed.

```
--PS2KEY.VHD
--Basic PS/2 keyboard interface with clock filtering
--George P. Burdell
--ECE 2031 L01
--01/31/2009

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;
USE IEEE.STD_LOGIC_ARITH.all;
USE IEEE.STD_LOGIC_UNSIGNED.all;

ENTITY keyboard IS
   PORT(
      keyboard_clk, keyboard_data,
         clock_25Mhz, reset, read : IN STD_LOGIC;
      scan_code : OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
      scan_ready : OUT STD_LOGIC);
END keyboard;

ARCHITECTURE a OF keyboard IS
   SIGNAL INCNT : std_logic_vector(3 downto 0);
   SIGNAL SHIFTIN : std_logic_vector(8 downto 0);
   SIGNAL READ_CHAR : std_logic;
   SIGNAL INFLAG, ready_set : std_logic;
   SIGNAL keyboard_clk_filtered : std_logic;
   SIGNAL filter : std_logic_vector(7 downto 0);
BEGIN

PROCESS (read, ready_set)
BEGIN
   IF read = '1' THEN scan_ready <= '0';
   ELSIF ready_set'EVENT and ready_set = '1' THEN
   scan_ready <= '1';
   END IF;
END PROCESS;


--This process filters the raw clock signal coming from the
--keyboard using a shift register and two AND gates
Clock_filter: PROCESS
BEGIN
   WAIT UNTIL clock_25Mhz'EVENT AND clock_25Mhz= '1';
   filter (6 DOWNTO 0) <= filter(7 DOWNTO 1) ;
   filter(7) <= keyboard_clk;
   IF filter = “11111111” THEN keyboard_clk_filtered <= '1';
   ELSIF filter= “00000000” THEN keyboard_clk_filtered <= '0';
   END IF;
END PROCESS Clock_filter;
```

```
--This process reads in serial data coming from the terminal
PROCESS
BEGIN
WAIT UNTIL (KEYBOARD_CLK_filtered'EVENT AND KEYBOARD_CLK_filtered='1');
IF RESET='1' THEN
      INCNT <= “0000”;
      READ_CHAR <= '0';
ELSE
   IF KEYBOARD_DATA='0' AND READ_CHAR='0' THEN
      READ_CHAR<= '1';
      ready_set<= '0';
   ELSE

      -- Shift in next 8 data bits to assemble a scan code
      IF READ_CHAR = '1' THEN
         IF INCNT < “1001” THEN
            INCNT <= INCNT + 1;
            SHIFTIN(7 DOWNTO 0) <= SHIFTIN(8 DOWNTO 1);
            SHIFTIN(8) <= KEYBOARD_DATA;
            ready_set <= '0';
            -- End of scan code character, so set flags and exit loop
         ELSE
            scan_code <= SHIFTIN(7 DOWNTO 0);
            READ_CHAR <='0';
            ready_set <= '1';
            INCNT <= “0000”;
         END IF;
      END IF;
   END IF;
END IF;
END PROCESS;
END a;
```