Examples of Improperly and Properly Labeled Figures, Tables, and Code

The examples and guidelines provided here can be applied to "Results" (printouts of schematics, screen captures, tables, code corresponding to the weekly Labs) as well as documents where the graphics are integrated into the body of the report. You can place more than one figure/table on a page.

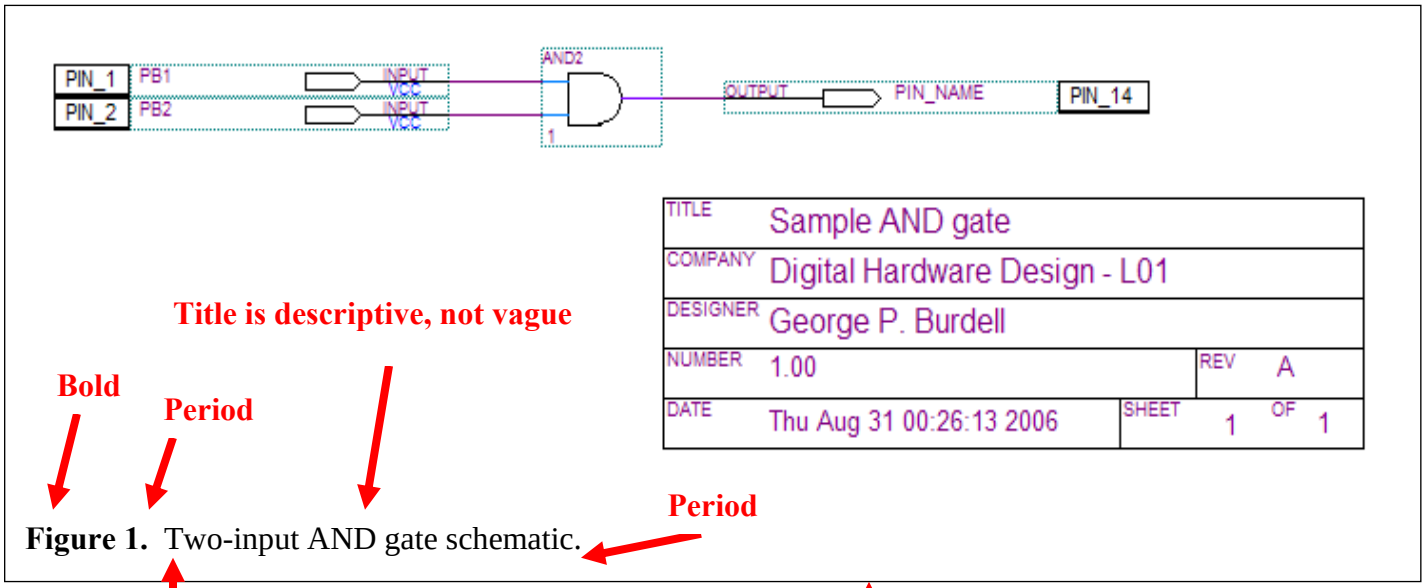
All figures were generated in Quartus software.

Example #1. Properly formatted schematic and title box

Note: The schematic is large enough so that all pin numbers, inputs, and outputs can be clearly seen. There is no way to resize the title box in Quartus, so don't worry that it seems a bit overpowering. The title box should be placed in the lower right corner of the figure. The caption needs to provide a descriptive name for the circuit, not just mention the step in the lab assignment where the circuit is created.

Examples of vague figure caption:

Figure 1. Circuit schematic.
Figure 2. Lab step 4 circuit.



Staple Here

Example #2. Properly formatted landscaped schematic and title box

You can orient your figures using portrait or landscape. However, when including a landscaped figure with other Results, label the figure so that the title caption is on the right-hand side of the page when the page is vertical. The staple would still be in the left corner. This configuration is standard and allows the reader to simply rotate the vertical page clockwise to read the landscaped figure horizontally.

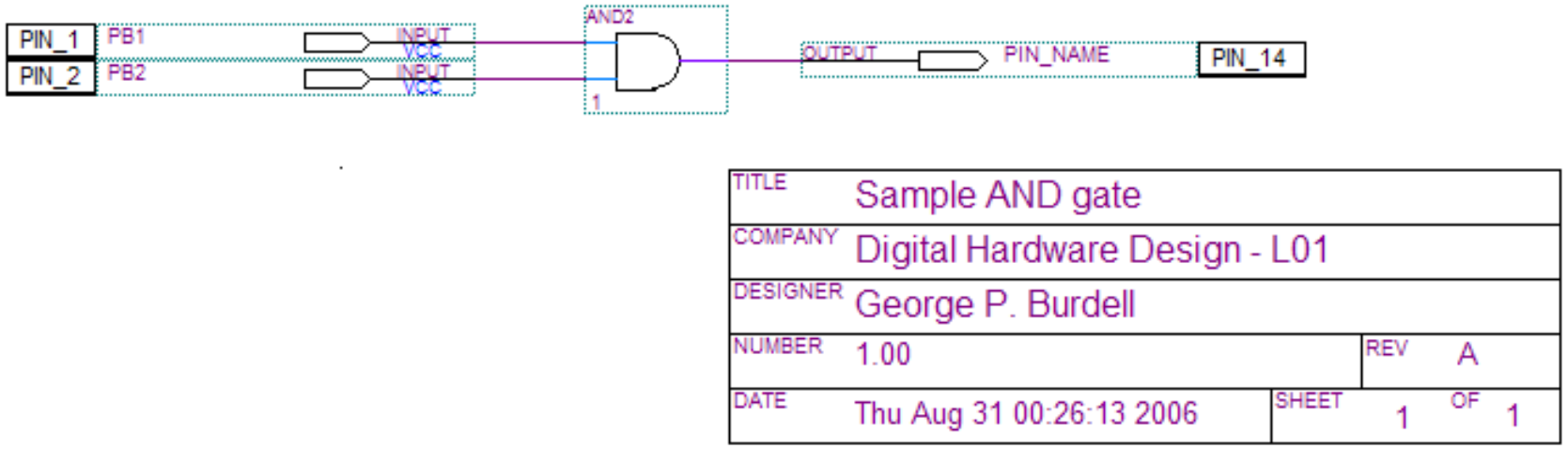


Figure 2. Two-input AND gate schematic.

Example #3. Improperly formatted schematic with grid showing

The following figure is **improperly** formatted because the dotted grid is showing in the background of the figure. Quartus turns on the grid by default, but it should be turned off. To turn off the grid, the option is (with the Block Diagram window open) View >> Show Guidelines.

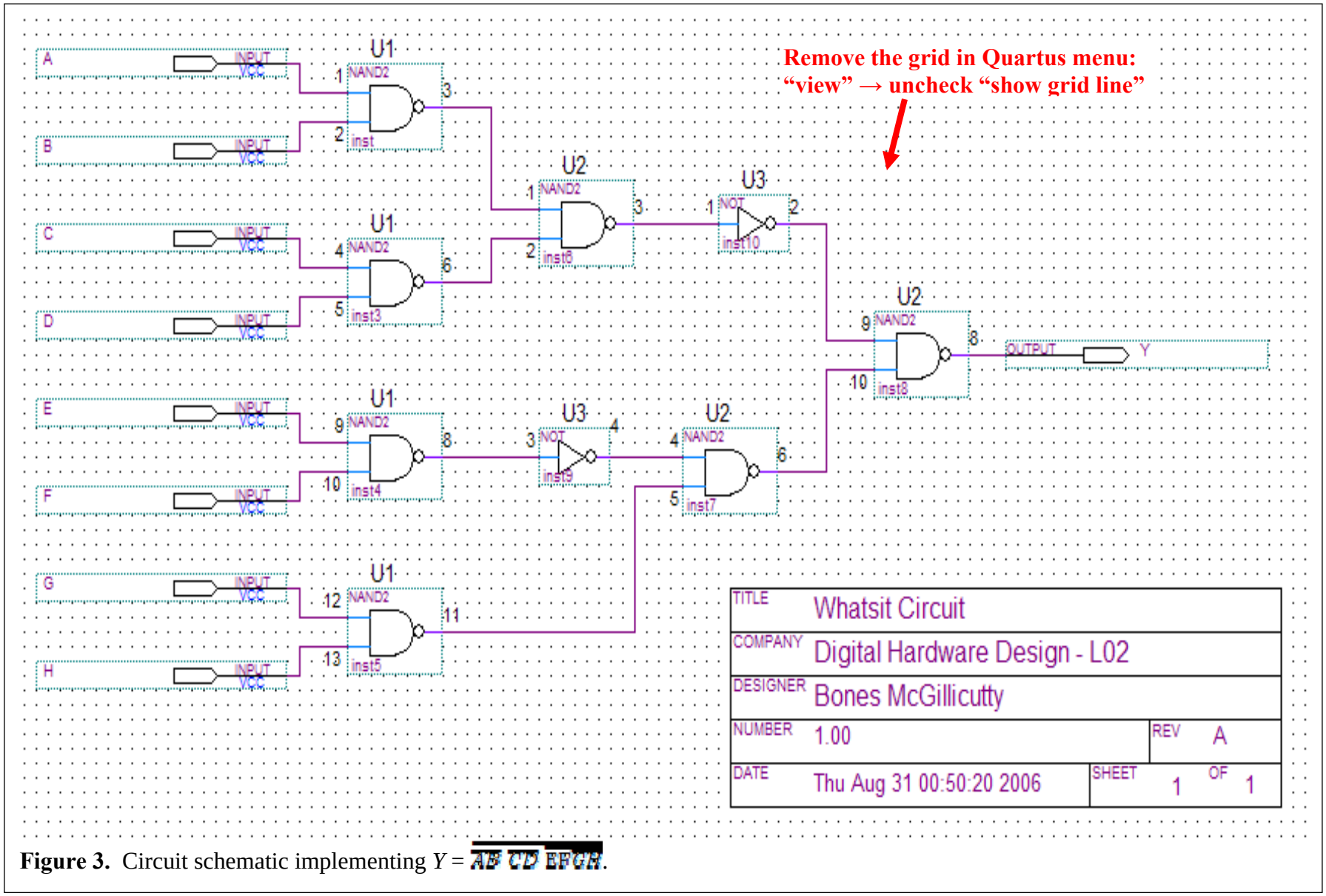


Figure 3. Circuit schematic implementing $Y = \overline{AB\ CD\ EFGH}$.

Example #4. Properly formatted schematic with dotted grid turned off

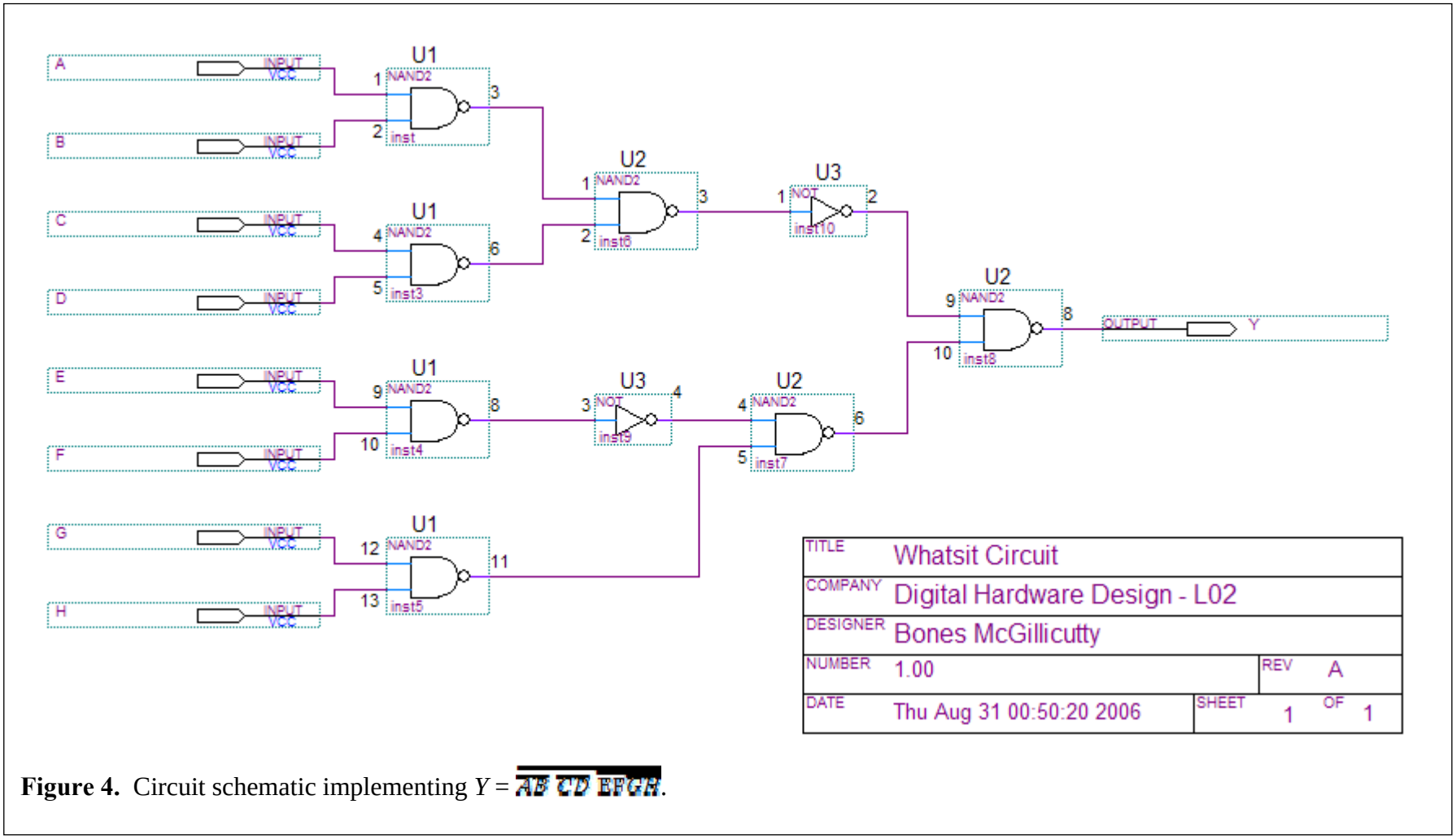


Figure 4. Circuit schematic implementing $Y = \overline{AB}\ \overline{CD}\ \overline{EFGH}$.

Examples #5-6. Properly labeled and formatted code (less than one page in length)

Figures 5 and 6 are examples of **properly** formatted code that span **less than a page**. Always place the title heading on the first five lines of every code submitted as shown below (i.e. "--ORGATE.VHD (VHDL)…").

```
-- 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 labeled as a figure and boxed

Figure 5. Negative-logic OR gate VHDL code.

```
/*
  ORGATE.V (Verilog)
  This code produces a negative-logic OR circuit
  George P. Burdell
  ECE2031 L01
  01/31/2001
*/

module orgate (PB1, PB2, LED);

// Port Declarations
     input PB1, PB2;
     output LED;

// Concurrent Assignment
     assign LED = !(!PB1 | !PB2);

endmodule
```

Figure 6. Negative-logic OR gate VHDL code.

Example #7. Properly formatted table

Shown below is an example of a **properly** labeled table. The title is at the top of the textbox and the **first letter of each word is capitalized**. Furthermore, there is **no period placed after the last word in the title**. This is the opposite of the caption seen in figures (i.e., it is placed in the lower left corner and the caption is in sentence case). As with all the figures, the caption should provide a descriptive name for the source of the data, such as "State Machine ABC" in this case.

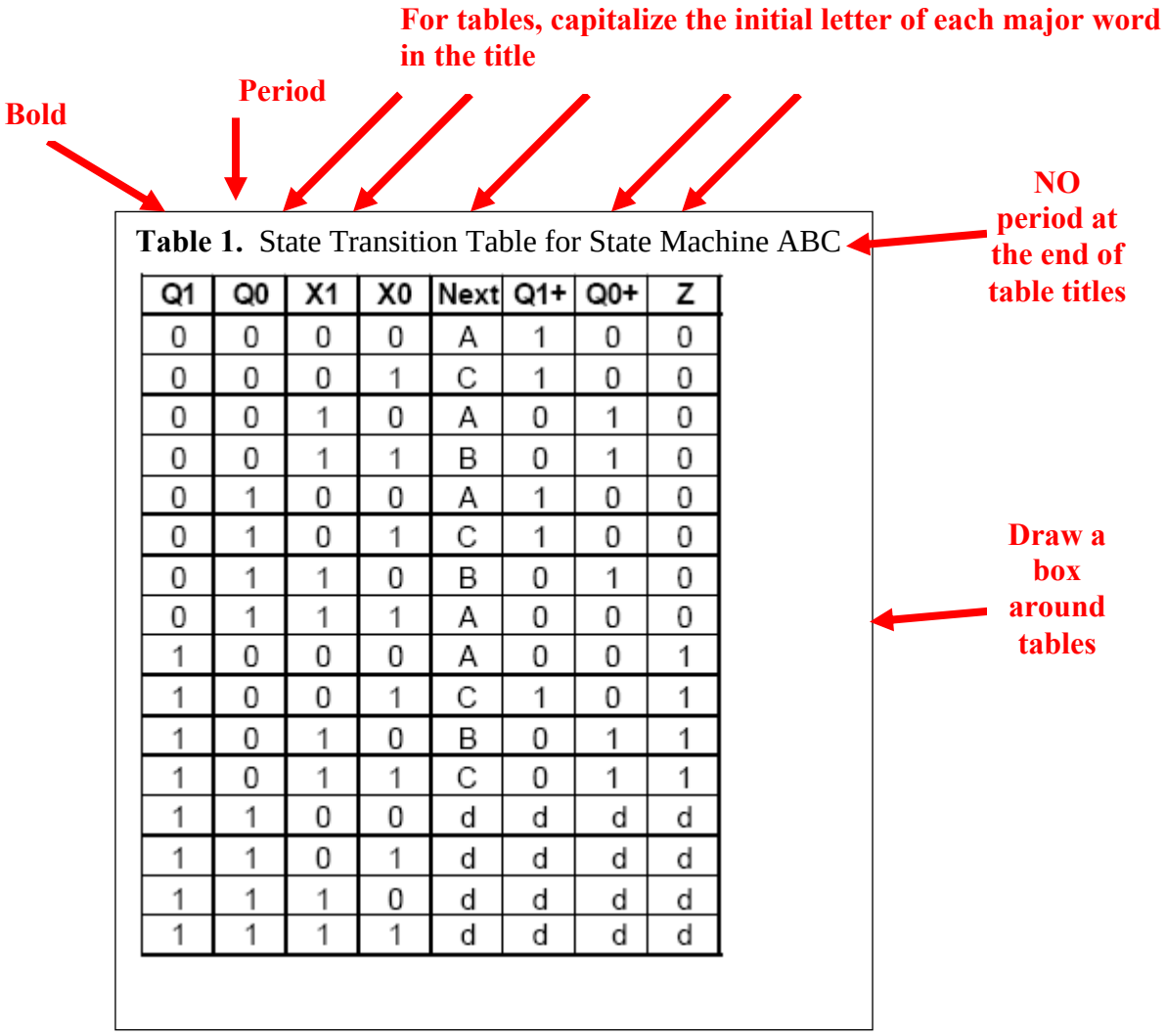


Table 1. State Transition Table for State Machine ABC

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

Example #8. Properly formatted waveform

Shown below is an example of a **properly** formatted waveform. Notice that a significant portion of the waveform is shown to see the activity of the signals changing. Moreover, notice the **scrollbar and window header are cropped out.** The caption needs to tell the reader what sort of waveform is shown (usually either functional or timing), and the name of the circuit that was used to create the results.

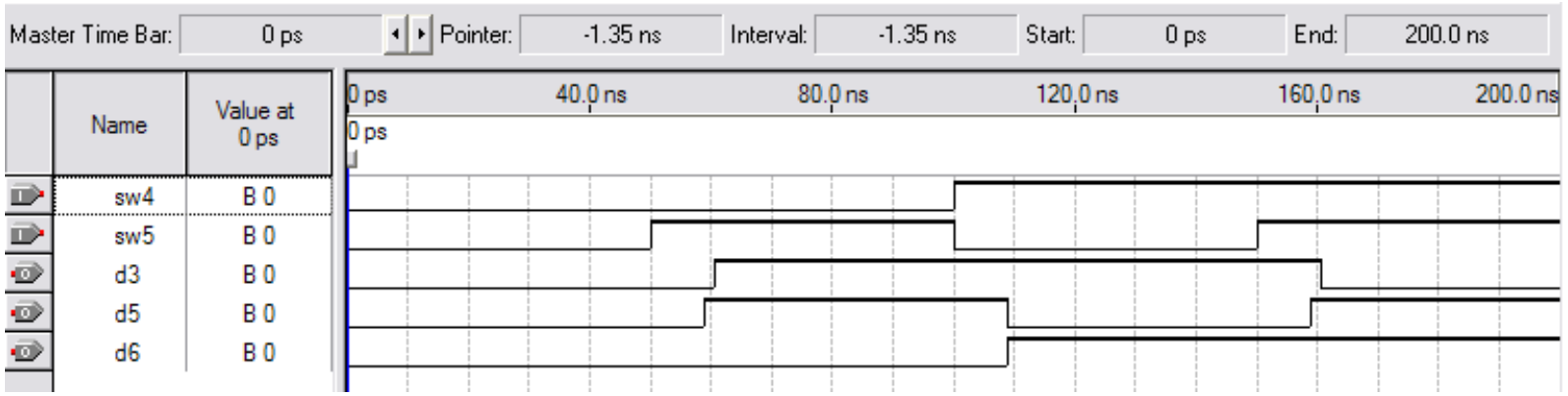


Figure 7. Timing simulation waveform for counter circuit.

Always include a descriptive title, with waveform type and circuit name

Example #9. Properly labeled Karnaugh maps

Shown below are examples of **properly** labeled Karnaugh maps. They can be placed in *either* a figure or a table. Make sure to circle the appropriate terms.

Note: Unless the Lab Manual specifies "hand annotated" or "worksheet," all figures and tables should be computer generated. Hand annotations and worksheets are acceptable only if specified in the Lab Manual.

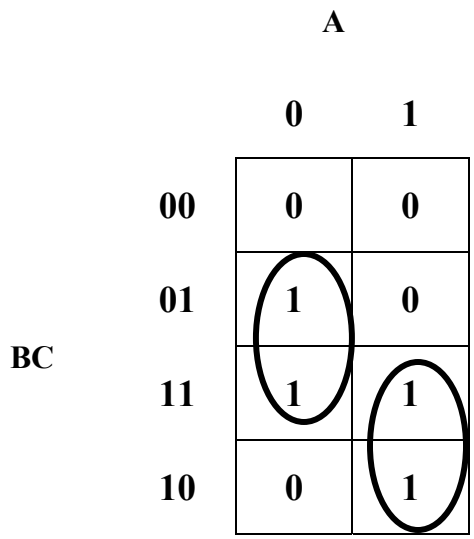


Figure 8. Example of a Karnaugh map implementing $\bar{A}C$ + AB.

Table 2. Karnaugh Map Implementing $\bar{A}$C + AB

BC \ A	0	1
00	0	0
01	1	0
11	1	1
10	0	1

Example #10. Oscilloscope screenshots

Oscilloscope screen captures should be cropped to show only the signals, as well as the information at the bottom that describes the settings of the oscilloscope. In the caption, not only is it important to specify the circuit that produced the signals, the signals need to be identified since there is no other means for the reader to know how the signals shown correspond with the circuit. It is acceptable for captions to refer to colors to distinguish signals, but only if the document is printed in color.

Note that the background is set to white to save printer ink

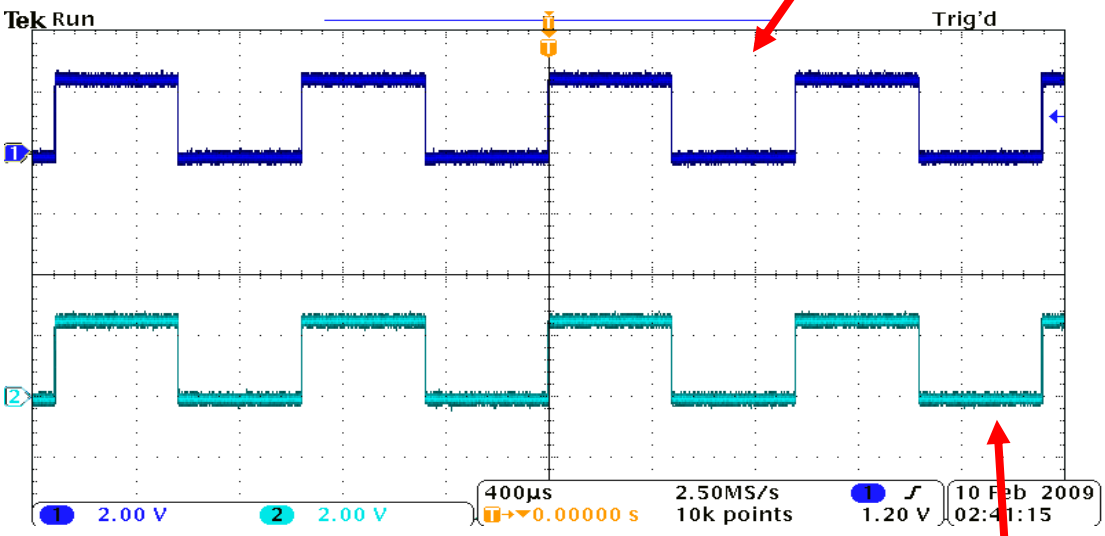


Figure 9. Signals V_a (top) and V_b from state machine ABC.

To separate signals, just specify one as on top or bottom

Display of oscilloscope menu is deactivated

Example #11. Code that spans more than one page

When code spans longer than one page in length, it should be NOT be boxed nor should it be labeled as a figure (i.e, don't put "Figure X: VHDL code" at the bottom of each page). Instead, treat each set of code as a separate Appendix by creating a cover page that describes the code. 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 cover page of each Appendix should be centered as shown below.

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

Appendix A: Example Long Code

```
--ORGATE.VHD
--This VHDL code produces a negative-logic OR circuit
--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';
```

Code always contains header lines

Non-proportional font is used for code formatting (i.e., Courier New)

```
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;
```