

Name: _____

ECET 331 – Digital Integrated Circuits

Lab 2 Two-Bit Adder with Karnaugh Mapping

Objective: Students successfully completing this lab exercise will accomplish the following objectives:

1. Learn the basics of binary addition.
2. Use Karnaugh maps to generate simplified Boolean expressions for the outputs of a circuit which adds two 2-bit binary numbers.
3. Construct the circuit which implements the expressions described in objective 2 above using discrete logic devices.

Equipment: Digital trainer, logic gates (7400, 7402, 7404, 7408, 7432, others as needed).

Lab Report: A formal lab report will be required for labs 2 and 3. Reports will be due one week after lab 3 has been performed. Lab handouts complete with tabulated data, calculations, schematics and diagrams, should be added as attachments to your formal report.

Procedure:

1. Complete the table below so that the outputs, Carry, S1 and S0 (bits 1 and 0 of the sum, respectively), represent the three bits which would result from the addition of two 2-bit inputs, A(1,0) and B(1,0).

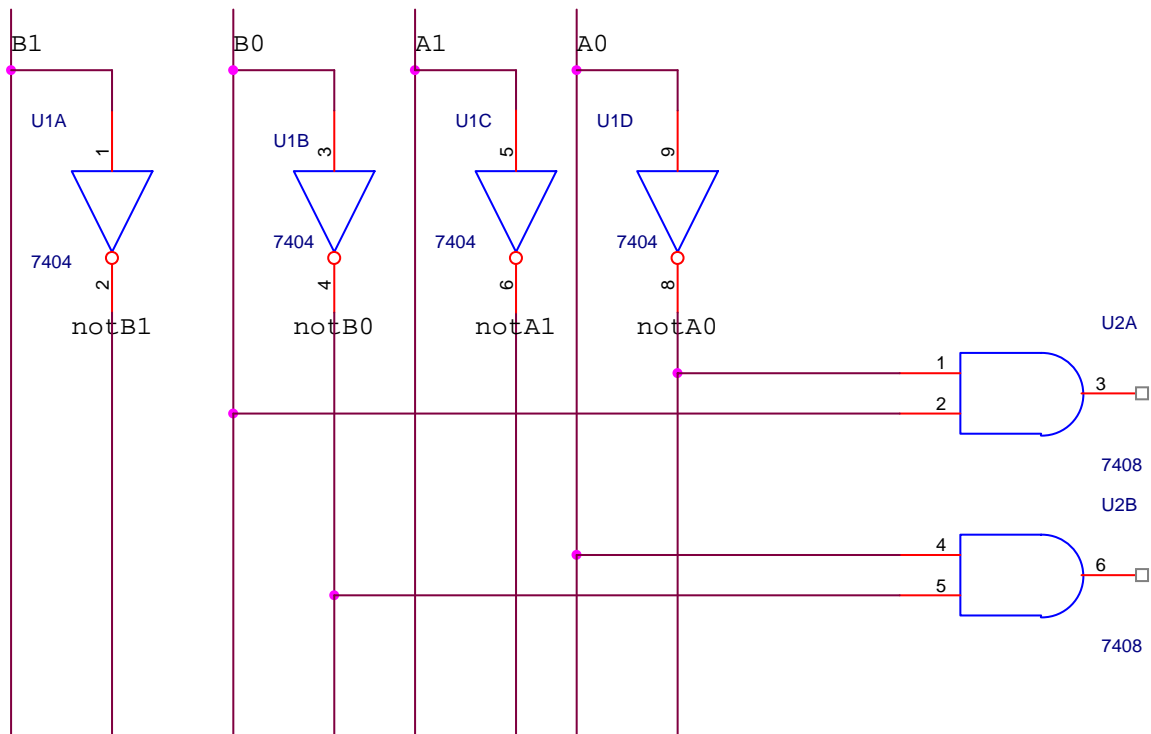
B1	B0	A1	A0	Carry	S1	S0
0	0	0	0	0	0	0
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1	1	0	0
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1	1	1	0

2. Generate a K-map and the resultant minimized Sum-Of-Products (SOP) expression for each of the three outputs (Carry, S1, S0). Show your work on separate sheets.
3. Generate the corresponding minimized Product-Of-Sums (POS) expressions.
4. Draw an initial schematic for the three outputs using bus form (double-rail inputs). See Figure 3.17, page 81 in your textbook for an example. Use the expressions from either step 2 or step 3 above.

5. Convert the bus form schematic to a wiring diagram (See the sample shown below in Figure 1) using the minimum number of packages. Remember that there are multiple gates per package. Use a single 7404 inverter to generate the complements of the inputs in the bus form.
- Generate the diagram for S_0 first using AND (7408) and OR (7432) gates.
 - Generate the diagram for Carry next using AND (7408) and OR (7411, 7432) gates. Remember that the OR function is associative.
 - Finally, generate the diagram for S_1 . You may find it takes fewer devices if you switch use a NAND-NAND structure. We only have two-input OR gates available in the lab while we have 8-input NAND gates available as well as 2- and 3-input devices. Remember that the NAND function is also associative.
6. Construct the circuit. Connect the three outputs to LEDs. Have the instructor verify your working circuit.

Instructor's Signature: _____ Date: _____

Figure 1: Sample Wiring Diagram



Questions for Lab Report:

Relevant Theory / Background Information:

- What is the nature of each of the three forms for describing a logic function: logic diagram, truth table, and Boolean expression?
- What is SOP form of a Boolean expression and how can it be extracted from a truth table?
- What is POS form of a Boolean expression and how can it be extracted from a truth table?
- What is a Karnaugh map?
- How can a Karnaugh map be used for simplifying a Boolean expression?
- What is a wiring diagram and what is it used for? How is it different from a logic diagram.
- What is an adder?

Experimental Data / Analysis:

- What are your POS and SOP expressions for each term of the 2-bit adder output and how did you develop and simplify them?
- How many gates (and gate packages) would it take to construct each of your expressions if you were not able to simplify the SOP and POS expressions and you were forced to use only AND, OR, and NOT gates? A table might be a useful way to present this information.
- Which set of expressions (SOP or POS) did you choose to construct and why?
- How many gates (and gate packages) did it take you to construct each of your chosen expressions after simplification? The table mentioned in the bullet point above might be a good place to present this data for a side-by-side comparison.
- Was your implementation successful?
- Were there any drawbacks to your chosen implementation?