Numerical Braille Display
Robert Walsh
Prof. Risbud
EE150
10/22/2012

DUE: Tuesday 10/22 5 PM (by Tuesday 10/16 5 PM for extra credit)

Hardware Design Assignment #1: Braille

Braille is a system which allows a visually challenged person to assess alphanumerics by feeling a pattern of raised dots. In this exercise, we shall deal with the digits only, so assume that only the Binary Coded Decimal (BCD) numbers 0 through 9 will be input to your circuit. You are to design and build a circuit which converts BCD to Braille based on the table below.

Table of BCD to Braille correspondence for the BCD digits 0 through 9: (The assumption is that some electromechanical device exists that raises a dot for each LED that is on.)

DATA INPUTS	D C B A (###)	0000 (zero)	0001 (one)	0010 (two)	0011 (three)	0100 (four)	0101 (five)	0110 (six)	0111 (seven)	1000 (eight)	1001 (nine)
BRAILLE PATTERN	W X	⚡ ⚡		⚡ ⚡	⚡	⚡	⚡		⚡	⚡	⚡
	Y Z	⚡ ⚡	⚡ ⚡		⚡	⚡		⚡		⚡	⚡

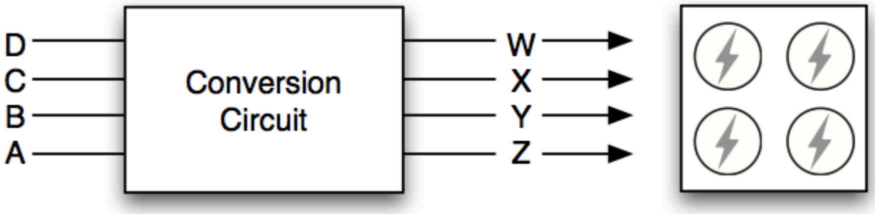
WORK ALONE AND DESIGN CAREFULLY:

- Use DIP switches to provide the inputs D, C, B & A.
- The LEDs must light when the corresponding variable is a ⚡ .
- The LEDs must be oriented in the patterns as seen in the boxes above.
- You may only use the six basic gates (AND, OR, NAND, NOR, NOT & XOR).
- You have a limited supply of parts, so your design must fit within one of each of the following types: quad dual input AND, NAND, OR, NOR, XOR and hex INVERTER
- Circuits that are designed and/or constructed poorly will be penalized.

DOCUMENTATION & NOTES:

- You must use 8.5" x 11" paper for all of your documentation.
- You must include the Karnaugh maps on the following page as part of your documentation.
- You must provide an explanation of any "non-standard" boolean reductions/minimizations.
- You must provide a logic, board and schematic (circuit) diagram to support your design.
- You must provide a bulleted list of issues or problems that you encountered.

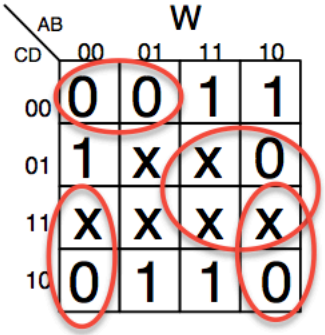
Block diagram of the system you will be designing and constructing



The Cooper Union for the Advancement of Science and Art — Fall 2012

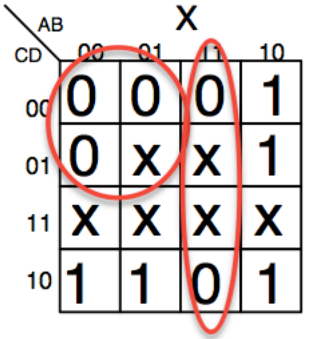
Digital Logic Design: ECE150: Prof. Risbud — HARDWARE DESIGN ASSIGNMENT #1

NAME: Robert Walsh EMAIL: Walsh3@cooper.edu



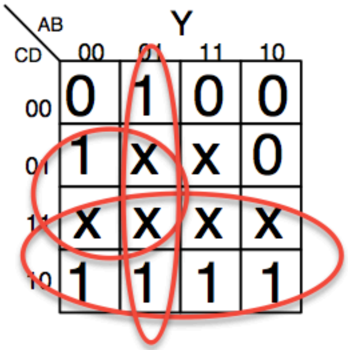
W

CD \ AB	00	01	11	10
00	0	0	1	1
01	1	x	x	0
11	X	X	X	X
10	0	1	1	0



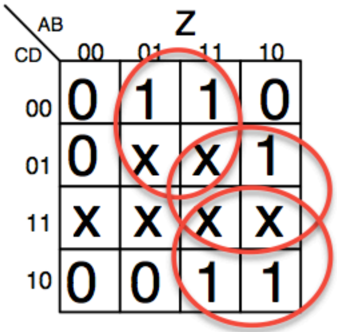
X

CD \ AB	00	01	11	10
00	0	0	0	1
01	0	x	x	1
11	X	X	X	X
10	1	1	0	1



Y

CD \ AB	00	01	11	10
00	0	1	0	0
01	1	x	x	0
11	X	X	X	X
10	1	1	1	1



Z

CD \ AB	00	01	11	10
00	0	1	1	0
01	0	x	x	1
11	X	X	X	X
10	0	0	1	1

BE VERY NEAT & SHOW ALL WORK FOR W,X,Y & Z. STAY WITHIN THE SPACE PROVIDED BELOW

W = (B+C')(C+(AD)'(A'D')')

X = (A'C')'(AB)'

Y = C+(A+(B+D)')'

Z = BC'+(A'+(D+C)')'

*** * * * * DO NOT WRITE BELOW HERE * * * * ***

IN ADDITION, YOU WILL BE GRADED ON YOUR COMPLIANCE WITH THE FOLLOWING LIST:

- □ Outputs are displayed on simple LED/resistor detectors driven up to Vcc. Do not expect the gates to "source" the current that is necessary for the output circuit to operate.
- □ Color code your circuit (for example, consistently use red wire for connections to Vcc (power), black wire for connections to GND (ground), and some scheme for your inputs and outputs.
- □ Do not "cut down" resistors or LEDs. These are the only components that are not required to be flat on the board.
- □ Do NOT cross wires over chips. Run wires around them, flat onto the board.
- □ Bends in wires must be a "soft" or "neat" 90 degrees.
- □ All wiring must be flat on the board.
- □ Keep leads as short as possible.

OTHER NOTES:

THE LOGIC

W: The output ‘W’ was found by the Karnaugh Map looping seen on the previous page. The looping actually yields (B+C’)(A’+D’)(A+C+D). In order to reduce this expression, the second term was strategically manipulated. By changing the second term to (A’+D’+C), a C is conveniently factored out of the second and third terms. Adding this C only has an effect on the “Don’t Care” states, and does not affect the logic for inputs from 0000 to 1001. After ‘C’ was factored, the two terms containing ‘A’ and ‘D’ were manipulated using Demorgan’s Law in order to reduce the number of “OR” expressions. The final expression for ‘W’ was (B+C’)(C+(AD)’(A’D’)’).

X: The output ‘X’ was found by the Karnaugh Map looping seen on the previous page. The looping yields (A+C)(A’+B’). These two terms were manipulated using Demorgan’s Law. The final output for ‘X’ was (A’C’)’(AB)’.

Y: The output ‘Y’ was found by the Karnaugh Map looping seen on the previous page. The looping yields C+A’B+A’D. By factoring out the A’, the expression reduces to C+A’(B+D). The expression on the right was then manipulated using Demorgan’s Law, in order to yield C+(A+(B+D)’)’.

Z: The output ‘Z’ was found by the Karnaugh Map looping seen on the previous page. The looping yields BC’+AD+AC. The ‘A’ was then factored out of the two rightmost terms, resulting in BC’+A(D+C). The term on the right was then manipulated using Demorgan’s Law in order to fit into the chip limitation requirements. The final output ‘Z’ was BC’+(A’+(D+C)’)’.

The combination of outputs W, X, Y, and Z, resulted in 19 logic gates. This includes 4 OR gates, 4 AND gates, 4 NAND gates, 4 NOR gates, and 3 NOT gates.

PROJECT SETBACKS

There were several setbacks during the construction of the Numerical Braille Display. Initially three of the karnaugh maps were looped incorrectly. Luckily two of these mistakes were found before the construction of the circuit was even initiated. However, 'W' was constructed in entirety before the improperly looped karnaugh map was discovered. Needless to say, the 'W' output did not work as required. Another main issue occurred during the construction of 'Y' output. A bad spool of brown wire was used, and it took awhile to find the strands that had breaks in them.

The third issue occurred when using the NOT gate. On the side of the chip containing the VCC pin, it was initially though that all the input pins were output pins, and vice versa. However some simple debugging quickly fixed this issue. The final issue was the most severe of all the others. The circuit was shorted unintentionally several times. Most of the time, this problem was quickly caught, but two or three times, the circuit started to smell like it was burning. No damage was done to any of the physical components.

ACKNOWLEDGMENTS

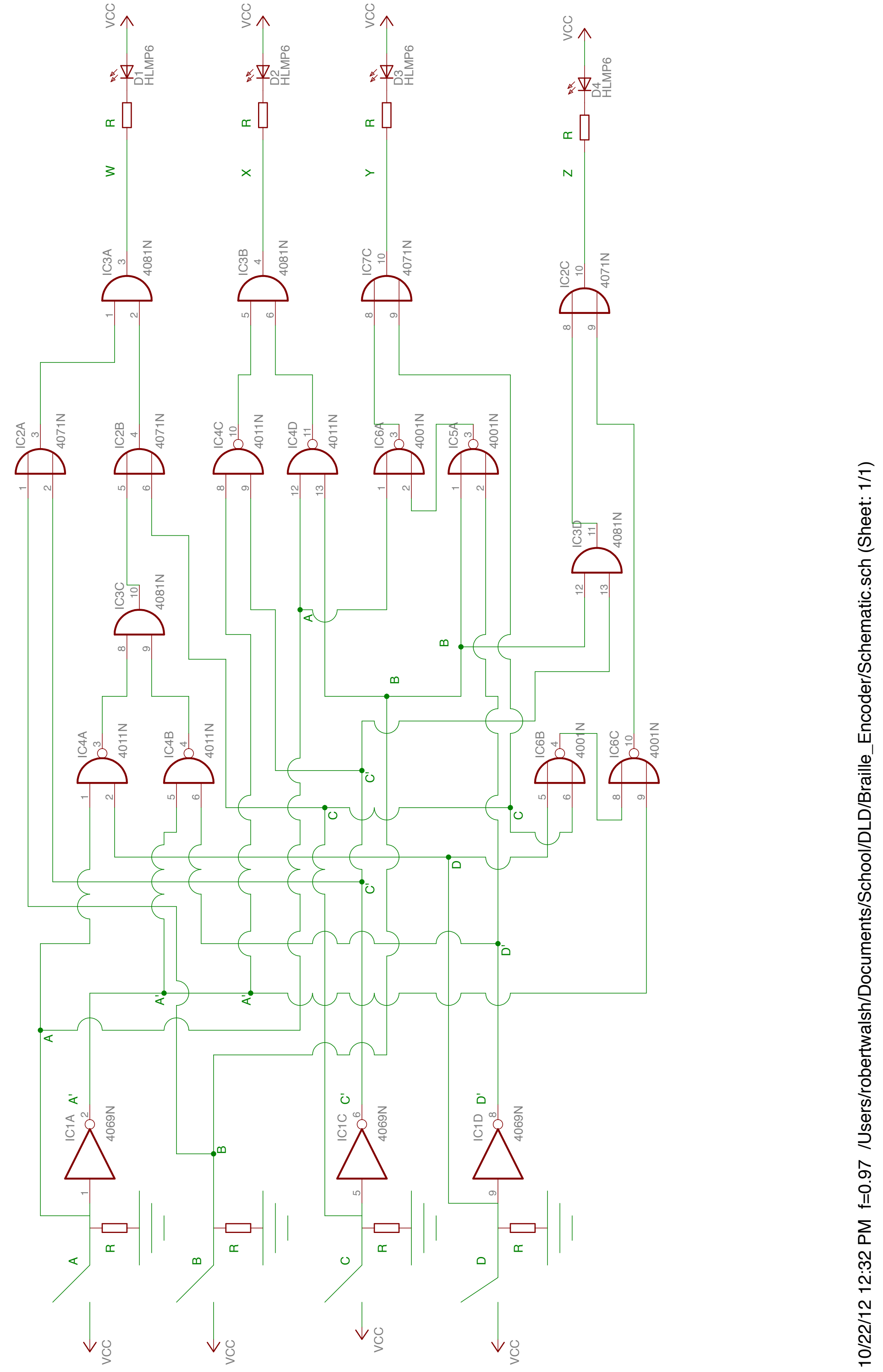
Credit must be given to Euguene Sokolov and Sheryan Resutov for explaining several uses of XOR in the circuit. Although the XOR chip was stripped from the current design, the knowledge will be useful at some point in the future.

COLOR SCHEME

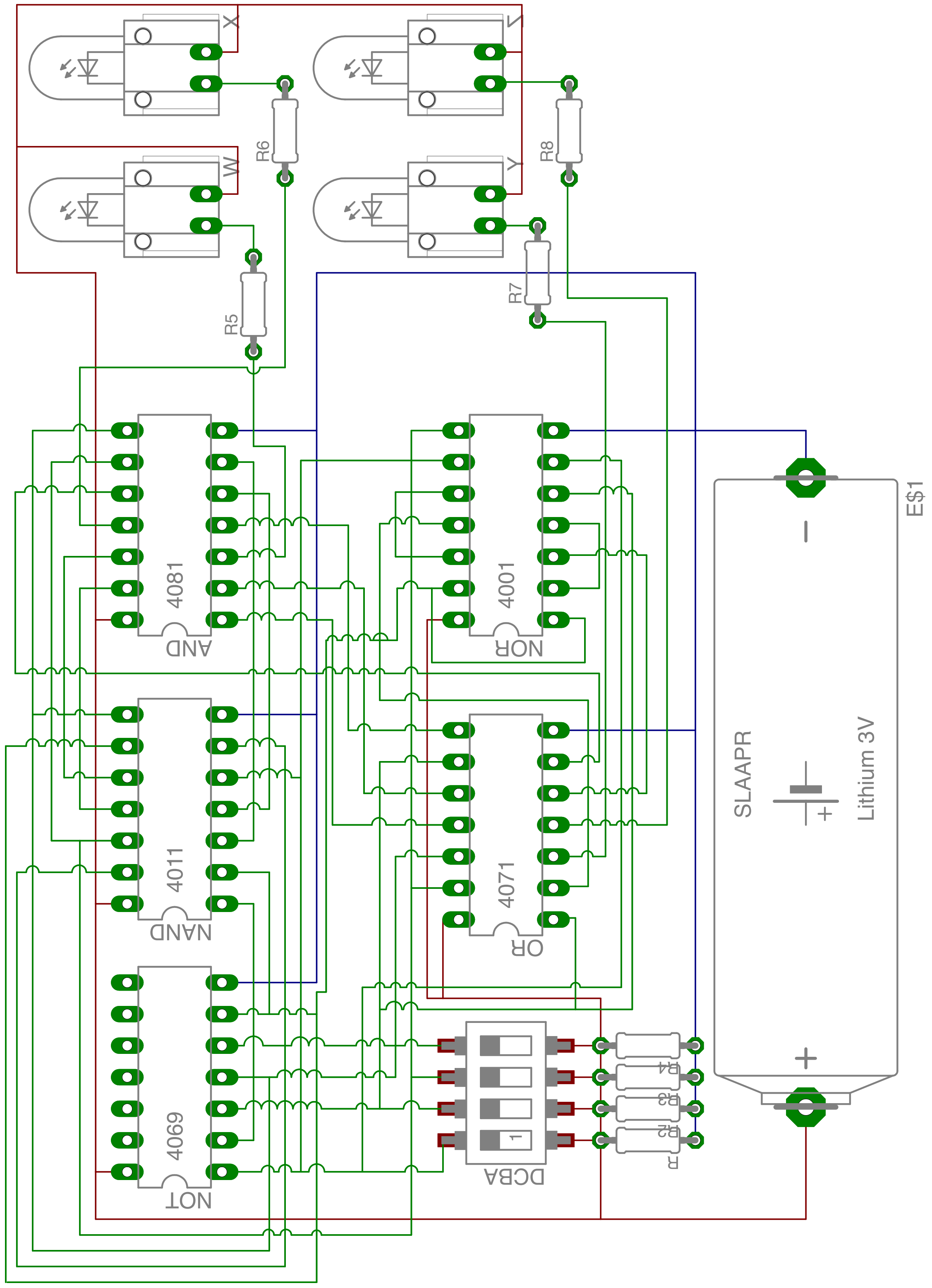
- A – Blue
- B – Purple
- C – Yellow
- D – Brown
- A', C' & D' – Grey
- Preliminary Outputs – Orange
- Final Output – Grey

Logic Diagram

Note: The OR and NOR gates are represented by the AND and NAND gates with the input lines completely piercing the gate



Board Diagram



10/22/12 8:42 PM f=2.30 /Users/robertwalsh/Documents/School/DLD/Braille_Encoder/Board.brd

Board Pictures