Introduction

This exercise uses a different column notation which is like the ones found in digital data books instead of the ones typical of textbooks. Columns are labeled B A for two input functions, preparing for 3 input functions with columns headed C B A and 4 input functions with D C B A column headings. This convention keeps the columns having the same weight while allowing more significant columns to be added. With this convention A is always weighted 2^0 ; B is 2^1 ; C is 2^2 ; D is 2^3 etc.

This two-input exercise is intended to provide you with some vocabulary common to data sheets, and the ability to think digitally. Older gate symbols are used but understanding these will aid in reading the newer symbols when they are encountered. As circuits developed more than a few inputs and outputs a newer concise form of symbols became necessary.

A two-input table as four lines from 0 to 3 in a natural binary count. 2^4 =16 columns with a natural binary count from 0 to 15. This is a small enough amount of material to practice with. Binary expands very fast with a three-input table with the binary representation of natural binary count from 0 to 7 resulting in 256 possible functions! 2^8 = 256 or a count from 0 to 255. If we add one more input to a circuit giving us D C B A columns the binary count is from 0000 to 1111 or 0 to 15. These 16 rows yield 65,536 possible different functions and we are only talking (thinking) about 4 inputs. Thinking about 16 or 32 or 64 inputs is complicated enough that we typically use hexadecimal numbers to handle these large values, but we are still dealing with 1s and 0s or H an L at the bit level.

As a child takes baby steps before learning to walk, then run practice with two input gates in this exercise will show you how to think and comprehend the signal levels on displays and on other chips. This material is derived from my experience in learning to read data books and looking for a way to make the process easier for other students and technicians. A co-worker who was a physics instructor once complained that he was not able to make sense of digital gates and circuits but had tried for several years to understand (comprehend) them. I suggested he take an earlier version of this home for the weekend. It was with excitement and raving on Monday when we met again. Not only did he express confidence in his ability to now think about digital circuits but requested he be allowed to use this with his students. Permission granted of course.

PRACTICE WITH TWO INPUT DIGITAL FUNCTIONS

Table 1

B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Columns B and A may represent a binary count from 0 to 3 when viewed as a two bit binary number: 00, 01, 10, 11.

When viewed vertically columns 0 through 15 represent a natural binary count from 0 to 15 with the top row the least significant bit and the bottom row the most significant bit.

Another viewpoint can be that the above table list all 16 two input functions or two input gates in truth table form. These truth tables can be observed by examining columns A and B along with one of the numbered columns. Each of the numbered columns has a Boolean Function name or verbal description. The columns may also be referenced as a function by number: f_5 as an example.

All have a Boolean algebra expression which can be shown as a formula, usually in two forms. One form is traditional while the other form is obtained by the use of the law of complements and DeMorgan's 1st or 2nd theorem.

Associated with each formula is a schematic logic symbol with most columns having two symbols.

ACTIVITY: For each column do the following:

1. Reproduce the column in the center of the work area adjacent to and to the right of columns A and B to form a truth table. (1) in example table.

2. Identify, where possible the function by the traditional name. (Some of the functions are actually circuits when implemented in small scale logic, depending on the family and therefore may not have a traditional name.) The traditional names are: AND, OR, INVERTER, NAND, NOR, XOR, and XNOR. (2) in example table (if it exists)

Note that the name NOT is absent with INVERTER being preferred by the writer. Use of the term NOT leads to problems in more advanced activities.

3. Show the Boolean algebra expression for the function High and the function Low. Use of the term function here typically is referencing the output column. See (3a) and (3b), order depends on function.

4. Draw a logic symbol for EACH Boolean algebra expression. See (4a) and (4b) again order depends on function. f_0, f_6, f_9, f_{15} are four functions that may not have two expressions or symbols.

5. Write a word description relating input levels, function, and output level using the templates below.

Blanks are to be filled with high or low or H or L as indicated by the function being described. Some functions will not fit either of the templates and modifications may be required as the function cannot be described with AND \OR vocabulary. f_0, f_6, f_9, f_{15} are four functions that will not fit this template.

#5 AND template: If A is ___ AND B is ___ then F will be ___.

#5 OR template: If A is ___ OR B is ___ OR both are ___ then F will be ___.

Note that the terms NAND, and NOR are not used as this system treats these terms as proper names rather than actions.

6. The AND condition identifies one and only one line in the truth table; while the OR condition identifies all but one line. Use → , ← and } { to indicate the line association with the gate symbol and the Boolean algebra expression.

Function 8 (f_8) will be used to demonstrate:

(1)

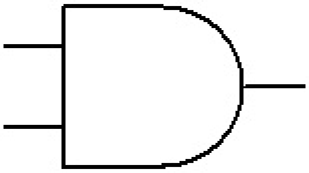
B	A	f_8=AND (2)
0	0	0 }
0	1	0 }
1	0	0 }
1	1	1

(6a) ←

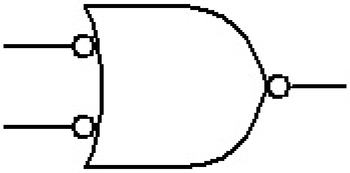
(6b) at } location

A*B = f (3a)

$\overline{A} + \overline{B} = \bar{f}$ (3b)



(4a)



(4b)

(5a) If A is H AND B is H then f will be H

(5b) If A is L OR B is L OR both are L then f will be L

Steps showing the DeMorgan algebra process for changing 3a to 3b above.

A *B = f Complement both sides of expression $\overline{A * B} = \bar{f}$

Break bar and change sign $\overline{A} + \overline{B} = \bar{f}$

In general the process of negating (or complementing) both sides of and expression provides an opportunity to use DeMorgan's first or second theorem remembering that two bars over an expression or single variable removes both bars leaving the variable indicating a High or H state.

The above is an example of DeMorgan's first law AND to OR equivalence.

The second law does the same for OR to AND equivalence.

INHIBIT – ENABLE VOCABULARY --------- APPLICATION OF GATES

Reference: Table of Two Input Functions; any Truth Table

Terms and Symbols

CONTROL a. Typically the most significant column in the truth table.

b. The input on a gate considered to be the controlling input.

DATA, DATA IN, SIGNAL Level changes on inputs other than the control line(s)

INHIBIT The level applied to the control input which results in no change at the output even though the data input is changing.

ENABLE The level applied to the control input which results in a change at the output when the data input is changing.

DATA OUT Level changes at the output which is identical with the input data level changes.

$\overline{\text{(Data Out)}}$ Complemented data out; Inverted data; Level changes at the output which are the complements of the input data level changes.

INHIBITED OUTPUT LEVEL IOL The constant level found at the output of an inhibited gate.

Information to discover from this exercise:

A. If basic gate shape is AND

1.____level enables

2.____level inhibits

B. If basic gate shape is OR

1.____level enables

2.____level inhibits

C. Condition of data out: Is it Data Out or $\overline{\text{(Data Out)}}$?

D. Inhibited output level: Is IOL H or L ?

Example for Inhibit-Enable Control Vocabulary

	control	Data in	Data out	f 8 AND			control	Data in	Data out	f 7 NAND
inhibit	0	0	0	IOL=0		inhibit	0	0	1	IOL = 1
inhibit	0	1	0	IOL=0		inhibit	0	1	1	IOL = 1
enable	1	0	0	Data out =		enable	1	0	1	Data out =
enable	1	1	1	Data in		enable	1	1	0	$\overline{(\text{Data In})}$

Procedure:

1. Copy truth table of desired function.

2. Split truth table into upper and lower halves based on most significant column changing from 0 to 1.

3. Assign term CONTROL to most significant column.

4. Assign term DATA IN to next column.

5. Assign DATA OUT to function column.

6. Examine DATA IN and DATA OUT columns. Assign the term INHIBIT to the half where Data IN is a 0 and 1 and the DATA OUT does not change. Assign the term ENABLE to the other half of the table. Write these terms on the left side of the table.

7. Examine the value in the DATA OUT column for the inhibited half of the table. This is the inhibited output level IOL.

8. Compare the DATA IN with the DATA OUT in the enabled half of the table. If they are the same then DATA IN = DATA OUT else DATA OUT is the complement of DATA IN (inverted data).

9. XOR and XNOR are important special cases. Attempt to do the above and modify the above vocabulary to accommodate these two functions. (f_6 and f_9)

MATCHING EXERCISE TRUTH TABLE → BOOLEAN EXPRESSION

Insert f_n in blank. $\overline{B}$ is read as B low or low B .

$\overline{A}$ is read as A low or low A

$\overline{f}$ is read as f low or low f

a____A = f

b____ $\overline{B}$ = f

c____$\overline{B}$ *$\overline{A}$ = f

d____0=f

e____$\overline{B}$ * A = f

f____ B = $\overline{f}$

g____$\overline{B}$ + A = $\overline{f}$

h____($\overline{B}$ *$\overline{A}$) + (B * A) = f

i____B * $\overline{A}$ = f

j____ B * $\overline{A}$ =$\overline{f}$

k____$\overline{A}$ = f

l____($\overline{B}$ * A) + (B *$\overline{A}$) = f

m____ B +A =$\overline{f}$

n____ B * A =$\overline{f}$

o____$\overline{B}$ + A =f

p_____1 = f

q_____$\overline{B}$ * A = $\overline{f}$

r____ B + $\overline{A}$ = $\overline{f}$

s_____$\overline{B}$ = $\overline{f}$

t____ A = $\overline{f}$

u_____ B + $\overline{A}$ = f

v_____ B = f

w_____$\overline{B}$ * $\overline{A}$ = $\overline{f}$

x_____ wired low; tied low

y_____$\overline{B}$ + $\overline{A}$ = f

z_____ wired high; tied hi

aa_____ B + A= f

bb_____$\overline{B}$ + $\overline{A}$ = $\overline{f}$

cc_____ B *A = f

dd_____$\overline{A}$ = $\overline{f}$

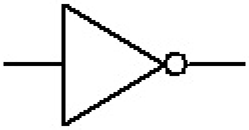


Figure 1 INVERTER active low output

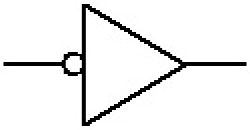


Figure 2 INVERTER active high output

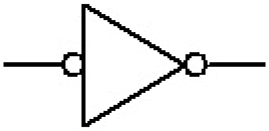


Figure 3 BUFFER active low

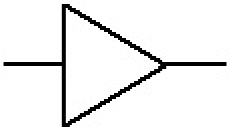


Figure 4 BUFFER active high

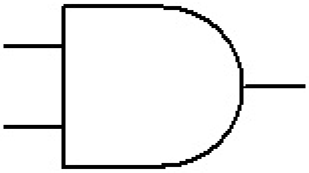


Figure 5 AND

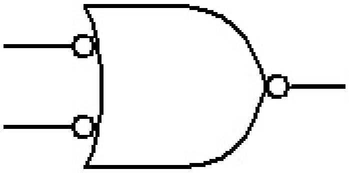


Figure 6 DeMorgan AND

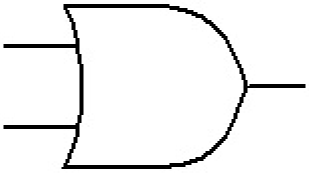


Figure 7 OR

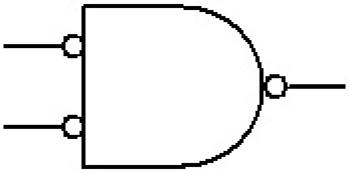


Figure 8 DeMorgan OR

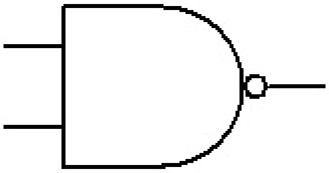


Figure 9 NAND

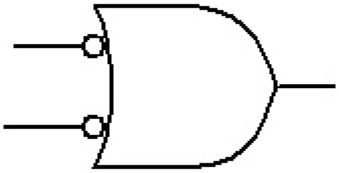


Figure 10 DeMorgan NAND

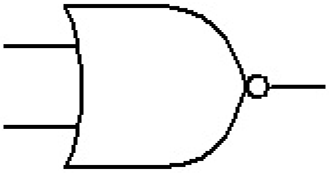


Figure 11 NOR

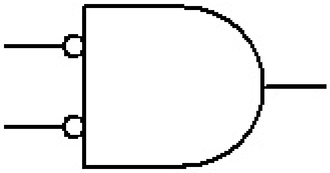


Figure 12 DeMorgan NOR

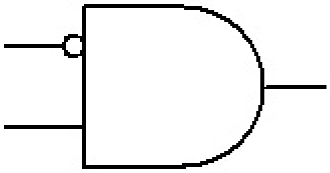


Figure 13 $\overline{a}$ * b = f

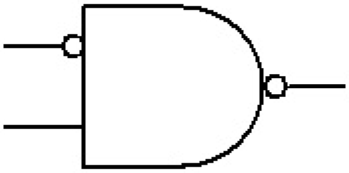


Figure 14 $\overline{a}$ * b = $\overline{f}$

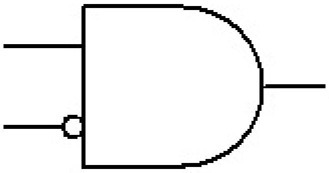


Figure 15 a * $\overline{b}$ = f

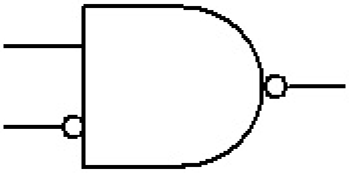


Figure 16 a * $\overline{b} = \overline{f}$

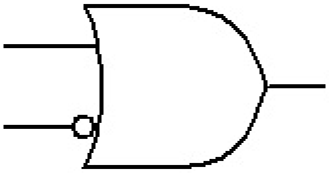


Figure 17 $a + \overline{b} = f$

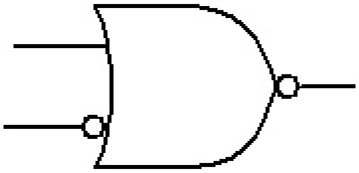


Figure 18 $a + \overline{b} = \overline{f}$

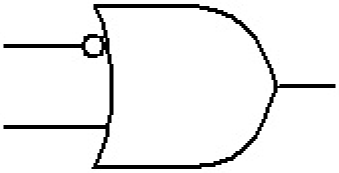


Figure 19 $\overline{a} + b = f$

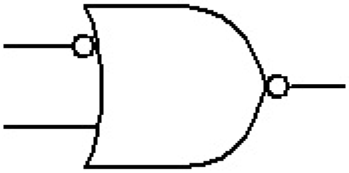


Figure 20 $\overline{a} + b = \overline{f}$

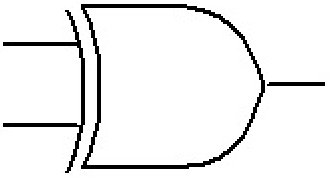


Figure 21 XOR

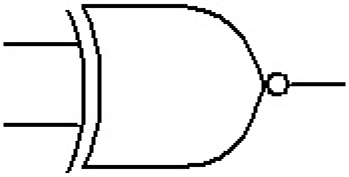


Figure 22 XNOR

SOLUTION KEY

MATCHING EXERCISE TRUTH TABLE → BOOLEAN EXPRESSION

Insert f_n in blank. $\overline{B}$ is read as B low or low B .

$\overline{A}$ is read as A low or low A

$\overline{f}$ is read as f low or low f

a _f_{10}___ A = f

b_f_3___ $\overline{B}$ = f

c_f_1_____$\overline{B}$ *$\overline{A}$ = f

d_f_0_____0=f

e_f_2___$\overline{B}$ * A = f

f_f_3___ B = $\overline{f}$

g_f_4___$\overline{B}$ + A = $\overline{f}$

h_f_9___($\overline{B}$ *$\overline{A}$) + (B * A) = f

i_f_4___B * $\overline{A}$ = f

j_f_{11}___ B * $\overline{A}$ =$\overline{f}$

k_f_5___$\overline{A}$ = f

l_f_6___($\overline{B}$ * A) + (B *$\overline{A}$) = f

m_f_1___ B +A =$\overline{f}$

n_f_7___ B * A =$\overline{f}$

o_f_{11}___$\overline{B}$ + A =f

p__f_{15}___1 = f

q__f_{13}___$\overline{B}$ * A = $\overline{f}$

r__f_2__ B + $\overline{A}$ = $\overline{f}$

s__f_{12}___$\overline{B}$ = $\overline{f}$

t__f_5__ A = $\overline{f}$

u__f_{13}___ B + $\overline{A}$ = f

v__f_{12}___B = f

w__f_{14}___$\overline{B}$ * $\overline{A}$ = $\overline{f}$

x___f_0__ wired low; tied low

y ___f_7__$\overline{B}$ + $\overline{A}$ = f

z___f_{15}__ wired high; tied hi

aa__f_{14}___ B + A= f

bb__f_8___$\overline{B}$ + $\overline{A}$ = $\overline{f}$

cc__f_8__ B *A = f

dd__f_{10}___$\overline{A}$ = $\overline{f}$

SHORT FORM ANSWER KEY

a f_{10}; b f_3; c f_1; d f_0; e f_2; f f_3; g f_4; h f_9; I f_4; j f_{11}; k f_5; l f_6; m f_1; n f_2; o f_{11}; p f_{15}; q f_{13}; r f_2; s f_{12}; t f_5; u f_{13}; v f_{12}; w f_{14}; x f_0; y f_7; z f_{15}; aa f_{14}; bb f_8; cc f