Jeide, Matthew Date 10/31/24 Period 2

W12 DE 2.1.1 Part A AOI Design: Truth Tables to Logic Expressions

Read the introduction in each section carefully.

INTRODUCTION

The first step in designing a new product is clearly defining the design requirements or design specifications. These design specifications detail all of the features and limitations of the new product.

In digital electronics, the process of translating these design specifications into a functioning circuit starts with the creation of a **truth table**. A truth table is simply a list of all possible binary input combinations that could be applied to a circuit and the corresponding binary outputs that the circuit produces. Once the truth table is complete, a Boolean expression can easily be written directly from the truth table.

In this activity you will learn how to translate design specifications into truth tables, and in turn, write unsimplified logic expressions from these truth tables.

In future activities we will learn how to use Boolean algebra, as well as a graphical technique called Karnaugh mapping, to simplify these logic expressions.

Procedure

Jeide, Matthew Date 10/31/24 Period 2

Truth Tables to Logic Expressions

Observe the truth table and its unsimplified logic expression.

A	B	Z	Minterms
0	0	0	
0	1	1	$\bar{A}B$
1	0	1	$A\bar{B}$
1	1	0	

$$Z = \bar{A}B + A\bar{B}$$

Figure 1. Truth Table and Minterm

1. Using the example as a guide, write the unsimplified logic expression for each of the following truth tables. Though it is not required, you may find it helpful to first write the **Minterm** expression for every place containing a 1 in the output function.

A. What is F_1?

M	N	F_1	Minterm
0	0	1	$F_1 = \overline{MN}$
0	1	0	
1	0	0	
1	1	1	$F_1 = MN$

B. What is F_2?

X	Y	F_2	Minterm
0	0	1	$F_2 = \overline{XY}$

0	1	0	
1	0	1	$F_2 = X\overline{Y}$
1	1	1	$F_2 = XY$

C. What is F_3?

P	Q	R	F_3	Minterm
0	0	0	0	
0	0	1	1	$F_3 = \overline{PQ}R$
0	1	0	0	
0	1	1	0	
1	0	0	1	$F_3 = P\overline{QR}$
1	0	1	0	
1	1	0	1	$F_3 = PQ\overline{R}$
1	1	1	0	

D. What is F_4?

Jeide, Matthew Date 10/31/24 Period 2

J	K	L	F_4	Minterm
0	0	0	0	
0	0	1	1	$F_4 = \overline{J}\overline{K}L$
0	1	0	0	
0	1	1	1	$F_4 = \overline{J}KL$
1	0	0	1	$F_4 = J\overline{K}\overline{L}$
1	0	1	0	
1	1	0	0	
1	1	1	1	$F_4 = JKL$

E. What is F_5? $\overline{W}$ $\overline{X}$ $\overline{Y}$ $\overline{Z}$ Complete all of the Minterms.

W	X	Y	Z	F_5	Minterm
0	0	0	0	0	
0	0	0	1	0	
0	0	1	0	1	$F_5 = \overline{W}\overline{X}Y\overline{Z}$
0	0	1	1	0	
0	1	0	0	1	$F_5 = \overline{W}X\overline{Y}\overline{Z}$
0	1	0	1	1	$F_5 = \overline{W}X\overline{Y}Z$

Jeide, Matthew Date 10/31/24 Period 2

0	1	1	0	0	
0	1	1	1	0	
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	1	$F_5 = W\overline{X}Y\overline{Z}$
1	0	1	1	0	
1	1	0	0	1	$F_5 = WX\overline{YZ}$
1	1	0	1	0	
1	1	1	0	1	$F_5 = WXY\overline{Z}$
1	1	1	1	0	

Logic Expressions to Truth Tables

Now that you have mastered the process of writing an unsimplified logic expression from a completed truth table, let's reverse the process.

Jeide, Matthew Date 10/31/24 Period 2

2.

Create a corresponding truth table for each logic expression.

> **Note:** Some terms in the logic expression may map to more than one place in the truth table.

a. $\mathbf{F_1 = \overline{C}\,\overline{D} + \overline{C}\,D}$
b. $\mathbf{F_2 = \overline{R}\,S\,\overline{T} + R\,\overline{S} + \overline{R}\,\overline{S}\,\overline{T} + S\,T}$
c. $\mathbf{F_3 = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,\overline{C} + A\,\overline{B}\,\overline{C} + A\,C}$

a) $F_1 = \overline{C}$

C	D	F_1
0	0	1
0	1	1
1	0	0
1	1	0

b) $F_2 = S(\overline{RT} + T) + R\overline{S} + \overline{RST}$

$F_2 = S(\overline{RT} + T) + R\overline{S}$, omitted 3rd condition, only true for 000

R	S	T	F_2
0	0	0	1
0	0	1	0
0	1	0	1

Jeide, Matthew Date 10/31/24 Period 2

0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Seat Belt Alarm Circuit

Now that you understand the mechanics of converting from a truth table to a logic expression (and vice-versa), let's revisit a circuit design we were introduced to in Unit 1. Your new car has an audio alarm that buzzes whenever the *door is open* and the *key is in the ignition* or *when the key is in the ignition* and *the seat belt is not buckled*.

> **Note**: This seat belt alarm design is slightly different than the one you created earlier.

3. Using the following variable names and assignment condition complete the truth table that captures the functionality of this audio alarm.

- D: Door → 0 = Door Open / 1 = Door Close
- K: Key → 0 = Key Not in Ignition / 1 = Key in Ignition
- S: Seat Belt → 0 = Not Buckled / 1 = Buckled

Jeide, Matthew Date 10/31/24 Period 2

- B: Buzzer → 0 = Buzzer Off / 1 = Buzzer On

[Double-click on the table to insert values in the image]

D (Door)	K (Key)	S (Seat Belt)	B (Buzzer)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Note: Subsequent activities, you will see that this logic expression can be simplified to:

$$B = \overline{D}\,K + K\,\overline{S}$$

Some of you may have been able to extract this simplified logic expression directly from the specifications. If so, you may be asking yourself, “Why did I have to go through the process of creating the truth table and writing the unsimplified logic expression first? This was a simple problem.” As you progress through this class, the design problems will become more difficult. The process that you are learning now will become invaluable in solving more challenging problems.