

# CERTIFICATION OF APPROVAL

## APPLICATION OF BOOLEAN ALGEBRA: CIRCUIT THAT CONTROLS THE CEILING LIGHT AND BUZZER IN THE AUTOMOBILE


By

**Mohd Rashdin Bin Ab. Rashid**

A project dissertation submitted to the  
Electrical & Electronics Engineering Program  
Universiti Teknologi PETRONAS  
In partial fulfillment of the requirement for the  
Bachelor of Engineering (Hons)  
(Electrical & Electronics Engineering)

Dr Muhammad Asif Sadiq  
Senior Lecturer,  
Electrical & Electronic Engineering  
Academic Block No 22  
Universiti Teknologi PETRONAS  
Bendera Seri Begawan  
41700 Tronoh, Perak Darul Ridzuan, MALAYSIA

Approved by,

  
(Dr. Muhammad Asif Sadiq)  
Project Supervisor

t  
TK  
7888.4  
M 697  
2004

1. Electronic digital computers..Circuits  
-- Design
2. Logic, Symbolic and mathematical
3. Algebra, Boolean
4. EEE -- Thesis

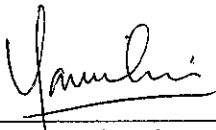
UNIVERSITI TEKNOLOGI PETRONAS

TRONOH PERAK

JUNE 2004

## CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the reference and acknowledgements, and that the originality work contained herein have not been undertaken or done by unspecified sources or person



---

Mohd Rashdin Bin Ab. Rashid

## ABSTRACT

In the present day, technology is drastically enhanced approximately after every eight months, new innovation establishes and electrical circuit gets more complex time to time. The project “Circuit that Controls the Ceiling Light and Buzzer in the Automobile” is a first step to study the similarity between operations of logical connectives and operations of switching devices. In this project, a powerful technique, named, Boolean algebra is introduced to simplify complex circuit. Complex circuit can then be easily simplified and designed by this approach. The project is mainly based on Boolean algebra and digital circuit design. The ceiling light control circuit and buzzer control circuit are then created after the simplification is done by Boolean algebra technique and then implemented it on programmable logic devices. The success of this project is highly dependent on what methods are used. Preliminary design, testing and implementation will be the key for project smoothness. Lastly, this project proves that there is a simpler way to design a very complicated circuit and how mathematic is related to circuit design.

## **ACKNOWLEDGEMENT**

I would like to acknowledge the contribution of my colleague and lecturer at Universiti Teknologi Petronas, Perak, who contributed to this final year project. Without their support this project may not have to success. Those who contributed directly or indirectly to this project require special thanks here. The continued support through all phases of this project development by Electrical and Electronics engineering department was valuable for the completion and printing of this work.

My special gratitude goes to Dr. Muhammad Asif for his guidance as the project supervisor. His guidance and suggestions were invaluable to me. My appreciation is extended to Cik Azira for helping me out with the PAL Trainer and also Cik Hawa who aid me to the electronics part selecting. Not to forget, I would like to thanks all my friends who have been supporting from the beginning of this project. Thank you so much.

## TABLE OF CONTENTS

CERTIFICATION	. . . . .	ii
ABSTRACT	. . . . .	iii
ACKNOWLEDGEMENT	. . . . .	iv
TABLE OF CONTENTS	. . . . .	v
LIST OF FIGURES	. . . . .	vii
LIST OF TABLES	. . . . .	vii
CHAPTER 1:	<b>INTRODUCTION</b> . . . . .	1
	1.1 Background of Study . . . . .	1
	1.2 Problem statement . . . . .	2
	1.3 Objectives and Scope of Study . . . . .	3
CHAPTER 2:	<b>LITERATURE REVIEW / THEORY</b> . . . . .	4
	2.1 Boolean Algebra . . . . .	5
	2.2 Karnaugh Map . . . . .	8
CHAPTER 3:	<b>METHODOLOGY / PROJECT WORK</b> . . . . .	12
	3.1 Procedure Identification . . . . .	12
	3.2 Tool Required . . . . .	13
	3.3 Project Procedure . . . . .	14
CHAPTER 4:	<b>RESULT AND DISCUSSION</b> . . . . .	16
	4.1 Circuit Design for Ceiling Light . . . . .	16
	4.2 Circuit Design for Buzzer . . . . .	19
	4.3 Implementation . . . . .	23
CHAPTER 5:	<b>CONCLUSION AND RECOMMENDATION</b> . . . . .	24
	5.1 Conclusion . . . . .	24
	5.2 Recommendation . . . . .	25

<b>REFERENCES</b>	.	.	.	.	.	.	.	26
<b>APPENDIX</b>	.	.	.	.	.	.	.	28

## LIST OF FIGURES

Figure 2.1: Flowchart for determining the minimum sum-of-products Using K-map [1] . . . . .	9
Figure 2.2: Digital network circuit (example) . . . . .	11
Figure 3.1: Project Flow chart . . . . .	15
Figure 4.1: Logic gate for doors . . . . .	18
Figure 4.2: Logic gates design for switch and doors . . . . .	19
Figure 4.3: Logic gate for buzzer design . . . . .	22
Figure 4.4: Final design of combination circuit . . . . .	22
Figure 4.5: Implementation using PLDs . . . . .	23

## LIST OF TABLES

Table 2.1: Basic operation of Boolean algebra . . . . .	5
Table 2.2: Example of truth table . . . . .	10
Table 2.3: K-map (example) . . . . .	11
Table 4.1: Truth table for doors . . . . .	17
Table 4.2: Truth table for switch and function of doors . . . . .	18
Table 4.3: Truth table for buzzer . . . . .	20
Table 4.4: Four Variables K-map . . . . .	21

# **CHAPTER 1**

## **INTRODUCTION**

This research paper is written for Final Year Research Project (EEB 5034) course, which is compulsory to be undertaken by all final year Electrical & Electronics Engineering students. The project is required to be completed within two semesters.

### **1.1 Background of Study**

This final year project (FYP) is a research-based problem. It requires student to do research on real world problem, which would motivate to produce practical solution. It also gives great advantages to use tools and techniques of problem-solving in order to overcome the matters. The experience that will gain, enable student to participate and prepare for future employment.

Claude Shannon noticed an analogy between the operations of logical connectives and the switching devices. Based on his theory, in a very simple circuit, the current can/cannot flow if the switch is closed/open. In this project, a circuit that controls the interior lighting and buzzer of the automobile is designed. In this problem, a discussion is made to the very powerful technique, Boolean algebra, which is very effective in designing the complicated circuits in a very simple way. [1]

Later on, by using the result obtained form the previous part, a new circuit is invented which includes doors, switch, head light, hand brake and ignition key. This shows how a new system can be designed by applying the Boolean algebra technique.



## 1.2 Problem Statement

Presently, most of the electrical circuits are very complex and difficult to be designed. This problem can be overridden by introducing Boolean algebra, which is a very powerful technique in developing complex logical circuit.

Recently, people in industry prefer designing logical circuit by using hit and trial methods. If the new or modification project is same, copy on the existing design is applied. The disadvantages of these methods are:

- No standard procedure
- Re-do and re-design are time consuming
- Result still cannot be finalized
- Lead to poor accuracy

However, in this project, all the problems stated above can be overcome by using Boolean algebra method. This provides a standard procedure for designing any circuit. Then results are simplified and lead to final robust design. After simplifying the lengthy Boolean equation, logical circuits are ready to be designed.

Design will start with a very basic digital circuit involves of input from five doors of the car and one output for ceiling light. It is also includes the switch to control the light on, auto and off mode. Then this application is extended by creating a new system that includes hand brake, head light and ignition key as switches to control a buzzer operation inside the automobile.

### **1.2.1 Significant of the project**

Design a circuit that controls the interior light and buzzer in automobile, for example, can be replaced the existing circuit in some automobiles. By using Boolean algebra approach, a new digital circuit will be designed. The purpose of this approach is to obtain the simplified circuit.

### **1.3 Objectives and Scope of Study**

- To study the similarity between the operations of logical connectives and the operation of switching devices.
- To simplify complicated logic circuit with the aid of Boolean algebra.

The project mainly involves Boolean algebra simplification skill. Hence, student must be able to convert English sentences to Boolean expressions. Thus constructing a circuit diagram and implementing it to the real world project would give a better understanding to the student. In addition, knowledge in software and hardware wiring is also essential for the best result of project. PLDs (Programmable Logic Devices) application is applied here for the project advancement.

This project is a relevant project for Electrical and Electronics Engineering student because it is related with digital logic circuit which has been covered in Digital Electronics course. Methodology for this project is one application of digital electronics circuit design. The basic procedure in designing digital circuit can be applied to all working environment.

Simplification of circuit is very important in designing a electrical circuit which can save a lot of time and money on hardware maintenance and troubleshooting. Hopefully this project can be a stepping-stone in designing a very complex circuit.

## CHAPTER 2

### LITERATURE REVIEW AND/OR THEORY

Literature review and theory come from various sources such as books, internet, lecturers and journal. The main theory mostly apply in this project is Boolean algebra. History states that, Boolean algebra had developed by George Boole in 1874 and he used it to solve problem in mathematical logic. Then a century later, came out Claude Shannon, was the first person who applied Boolean algebra to design switching networks in 1939. [1]

Although the technology used to implement digital logic circuit has changed significantly from years to years, but the fundamental principles of logic design have not. Truth tables and state tables are still used to specify the behavior of logic networks and Boolean algebra is still the basic mathematical tool for logic design. Even though programmable logic devices (PLDs) is already in used instead of individual logic gates, reduction of logic equations is still necessary in order to fit equations into a small number of PLDs. [2]

For this project, the main knowledge is on some basic theory of Boolean algebra including Boolean algebra basic operation, Boolean expression and truth tables, algebraic simplification of switching expressions, commutative, associative and distributive laws, simplification theorem, inversion, duality and Consensus Theorem.[1]

Furthermore, project initiator must be able to convert English sentences to Boolean expressions. This theorem is the beginning of project achievement and can be done by creating the truth table. From the truth table, Boolean equation is derived. Algebraic technique is done to get the final simplified equation. However, the main problem arises how to obtain the final minimum solution. In order to overcome this difficulty, Karnaugh map method is applied because it provides a systematic method for simplifying switching functions. [3]

## 2.1 Boolean Algebra

The basic mathematics needed for the study of logic design of digital circuit is Boolean algebra. Claude Shannon who applied Boolean algebra to design switching networks in 1939 had proved it. Boolean variables, for example X and Y is used to represent the input or output of a switching network. [1]

### 2.1.1 Basic Operations

The basic operations of Boolean algebra are AND, OR, and NOT (complement).[1]

AND			
$0 \cdot 0 = 0$	$0 \cdot 1 = 0$	$1 \cdot 0 = 0$	$1 \cdot 1 = 1$
OR			
$0 + 0 = 0$	$0 + 1 = 1$	$1 + 0 = 1$	$1 + 1 = 1$
NOT			
$0 = 1$		$1 = 0$	

Table 2.1: Basic operation of Boolean algebra

The law of ordinary algebra which is also applies to Boolean algebra.

#### 1. Law of complementarity [1]

- a.  $X + X' = 1$
- b.  $XX' = 0$

#### 2. Commutative laws [1]

- a.  $XY = YX$
- b.  $X + Y = Y + X$

#### 3. Associative laws [1]

- a.  $(XY)Z = X(YZ)$
- b.  $(X + Y) + Z = X + (Y + Z) = X + Y + Z$

#### 4. Distributive laws [1]

- a.  $X(Y + Z) = XY + XZ$
- b.  $X + YZ = (X + Y)(X + Z)$

The following theorems are useful in simplifying Boolean expressions:

**5. Simplification theorem [1]**

- a.  $XY + XY' = X$
- b.  $X + XY = X$
- c.  $(X + Y')Y = XY$
- d.  $(X + Y)(X + Y') = X$
- e.  $X(X + Y) = X$
- f.  $XY' + Y = X + Y$

These simplification theorems are very useful during circuit design stage. Complex logical expression can be reduced and lead to a very simple expression with same operations. This means that less logical term are used. As a consequence, it increases the logic reaction time by gates and saving money. [4]

One theorem from algebraic simplification is Consensus theorem. This theorem is very useful in simplifying Boolean expression. This theorem can eliminate redundant term and simplify the uses of gates. Here are some examples of Consensus theorem:

**6. Consensus theorem [1]**

- a.  $XY + X'Z + YZ = XY + X'Z$
- b.  $(X + Y)(X' + Z)(Y + Z) = (X + Y)(X' + Z)$

**2.1.2 Conversion of English Sentences to Boolean Equation**

Logic design problems are often stated in terms of one or more English sentences. The first step in designing a logic circuit is to translate the sentences into Boolean equations. The second step is to break down each sentence into phrase and associate a Boolean variable with each phrase. It is easy if a phrase has “true” or “false” value. [6]

### 2.1.3 Truth Table

When constructing a truth table, there will be two standard algebraic form of the function can be derived. [7]

1. Standard sum of products (minterm expansion)
2. Standard products of sum (maxterm expansion)

Simplification of either of these standards form leads directly to an implementation of network using AND and OR gates. Somehow, developing a truth table is a tedious job. If there are a lot of inputs, there will be a very long table. The condition for truth table is  $2^k$  (where k is the number of the input or variable). If there are 4 inputs, the number of condition will be 16, significant to the equation. [1]

### 2.1.4 Simplification Techniques

When the truth table is already developed and all the conditions are filled, then the result for the Boolean equation can be obtained. There are three methods that can be applied for the simplification which are: [8]

1. Algebraic simplification
2. Karnaugh maps<sup>1</sup>
3. Quine-McCluskey Method

Algebraic simplification is normally very difficult if facing a few numbers of inputs. In order to overcome that problem, an application of Boolean algebra which is known as Karnaugh map (often called the K-map) is introduced. Karnaugh map is a useful tool for simplifying and manipulating switching function of three or four variables. However, it is limited only up to six variables. The Quine-McCluskey method is better to use for a large number of variables. This method is base on basic Boolean operation and can be used up to 15 variables. Algebraic simplification comes to at advantage if there are large number of variables and small number of term. There are also several other methods but most of these methods are just variations of these three basic simplification techniques.[9]

---

<sup>1</sup> This tool was introduce in 1953 by Maurice Karnaugh

## 2.2 Karnaugh Maps

Switching functions can be simplified by using the algebraic simplification techniques as describe in basic operation of Boolean algebra. However there are two problems arise when algebraic procedures are used: [9]

1. The procedures are difficult to apply in a systematic way.
2. It is difficult to tell when minimum solution is arrived.

The Karnaugh map overcomes these difficulties by providing systematic method for simplifying functions. Karnaugh map is very useful tool for simplifying and manipulating switching function up to six variables. This method is faster and easier to apply than other simplification methods. [1]

Karnaugh map techniques is directly lead to minimum cost two-level network composed of AND and OR gates. An expression consisting of sum of product terms corresponds directly to two-level network composed of a group of AND gates feeding a single OR gate (see example). [4]

The following procedure can be used to obtain a minimum sum of product from a Karnaugh map. [1]

1. Choose a minterm (a '1')
2. Find all 1's adjacent to that minterm.
3. If a single term covers the minterm of all of the adjacent, that term is essential prime implicant<sup>2</sup>, so select that term.
4. Step 1, 2, and 3 is repeated until all prime implicants have been chosen.
5. Find the minimum set of prime implicants on the map.

---

<sup>2</sup> 1 or any group of 1's which combined together on K-map.

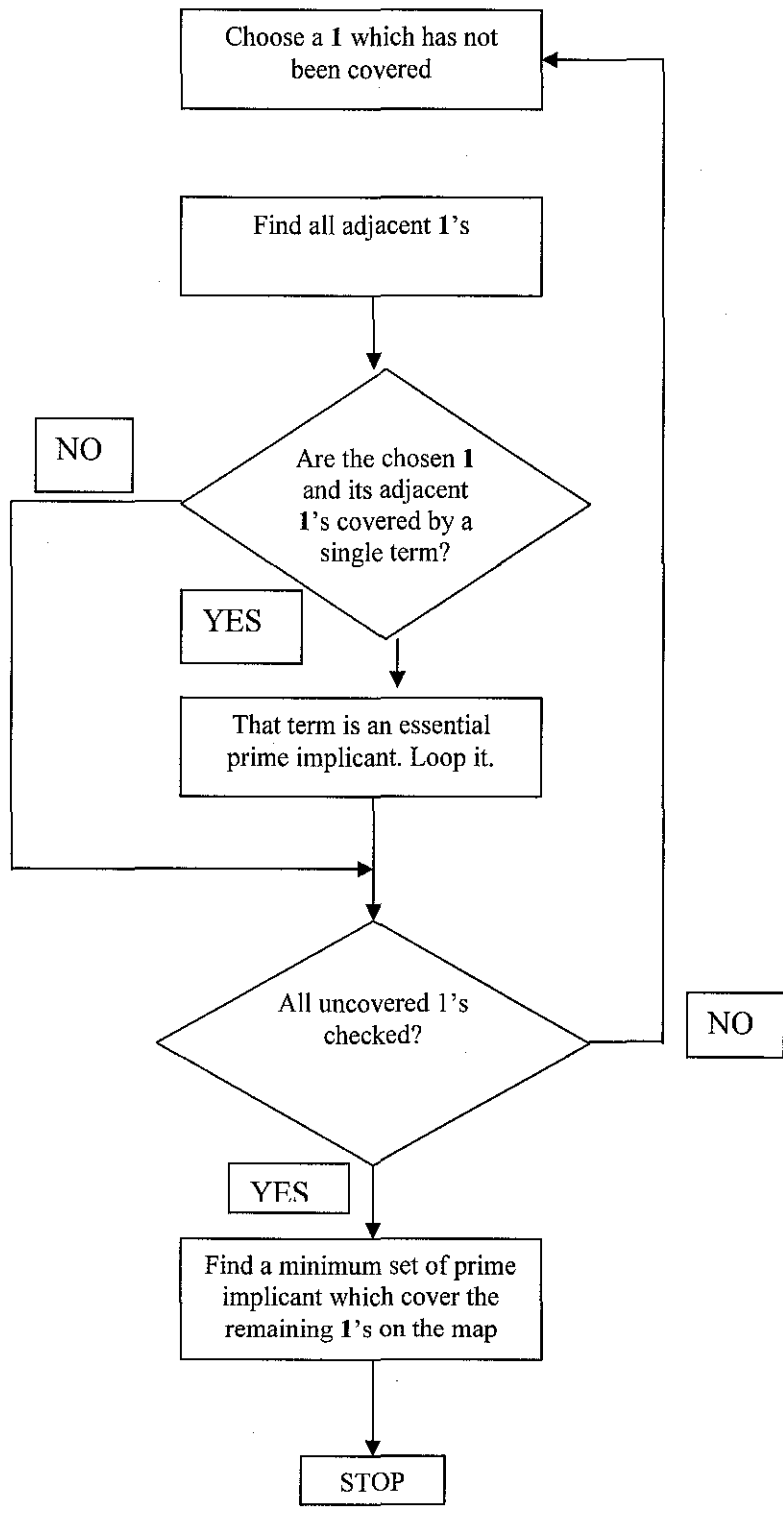


Figure 2.1: Flowchart for Determining the minimum sum-of-products Using K-map [1]



**Example:**

This example uses 4 logic status; A, B C, and D. The output is represented as F. Conversion of English sentences are skipped since it is left to the user to decide about the output. Results for the truth table are assumed as below:

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Table 2.2: Example of truth table

Result from the truth table (Algebraic simplification):

$$\begin{aligned}
 F &= m(3, 7, 11, 15)^3 \\
 &= A'B'CD' + A'BCD' + AB'CD' + ABCD' \\
 &= CD' (A'B' + A'B + AB' + AB) \dots\dots\dots 4.a \\
 &= CD' [A'(B+B') + A(B'+B)] \dots\dots\dots 1.a \\
 &= CD' [A'(1) + A(1)] \dots\dots\dots 1.a \\
 &= CD'[1] \\
 &= \mathbf{CD'}
 \end{aligned}$$

---

<sup>3</sup> Denoted the number of rows in truth tables

Another method that gives the same result is Karnaugh map method.

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

Table 2.3: K-map (example)

$$F = CD'$$

K-map application is rather fast in generating the results as compared to Boolean simplification theorem. However, this technique is limited to 6 variables only. Both of these methods give the same result for logic network.

The final result can be shown as below:

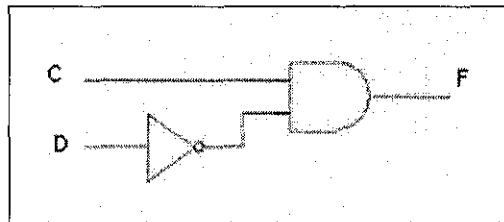


Figure 2.2: Digital network circuit (example)

The final result shown above is represented with logic gates diagram. In other word the result can be expressed in other configuration such as transistors, relays, switches, ladder diagram and etc. The decision of using these devices depends on the working environment of certain projects.

## CHAPTER 3

### METHODOLOGY

#### 3.1 Procedure Identification

For this project, the main objective is to design a circuit that control the ceiling light and buzzer of automobile by discovering the way how switching network is operated. At the same time, Boolean algebra technique is used to get the same result as the switching network application.

This project is based on designing a circuit that control ceiling light and buzzer in automobile due to the fact that it is the effective methodology that can start before going into more complex circuit. The problem is quite challenging because the inputs are consisted of five doors of the car, hand brake, head light, ignition key and manual switch. Hence, the outputs are a ceiling light and a buzzer. Below are the suggested steps that will be considered to deal with the stated problem

- 1) Problem analysis
  - Research and studies about Boolean algebra applications and digital circuit design.
  
- 2) Requirement analysis
  - Become skilled at digital circuit designing.
  - Able to simplify Boolean expression and lead to simplify complex circuits
  - Able to use Programmable Logic Devices (PLDs) to design a digital circuit.
  
- 3) Design
  - Design the digital circuit to control the ceiling light and buzzer by using Boolean algebra technique.

- 4) Testing
  - Start testing the circuit that was developed using Boolean algebra simplification method.
  
- 5) Implementation
  - Implement the circuit into a prototype.

### **3.2 Tool Required**

For the project, student needs related equipment and software to ensure the project can run smoothly. Here is the list of items required for the project.

Hardware:

- Personal computer or laptop.
- Wiring tools: - Stripper, iron solder, cutter and etc.
- Electronics parts: - resistors, switches, LEDs or buzzer.
- Flight Electronics PAL Trainer
- PLDs

Software (optional):

- Multisim
- Electronic Workbench
- Pspice

### 3.3 Project Procedure

When designing a digital circuit, there is a certain procedure which must be followed for the project smoothness. This involves identifying the input and the output works and PLDs programming. The steps are:

a) Conversion from English to Boolean state

At this stage, user needs to know how many inputs and outputs are required. Then under what condition, input will activate the output. For example, light will be turn off when all doors are closed or OFF switch is selected.

b) Develop a Truth table

When all conditions are already cleared, a truth table is developed according to how many variables it has. The more inputs, the longer truth table must be developed.

c) Simplification technique

At this stage, Boolean equation is already obtained. The task is to simplify the equation. There are 3 options to use like algebraic simplification, Karnaugh map and Quine-McCluskey method. For this project only algebraic simplification and Karnaugh map are considered.

d) Final logic design

Simplified Boolean equation obtained from previous work will be used as a final result. Then it is up to the user what method to use for the final design. This can be PLDs, transistors, logic gates, relays, or programmable logic controller (PLC). PLDs is chosen as the final implementation for this project.

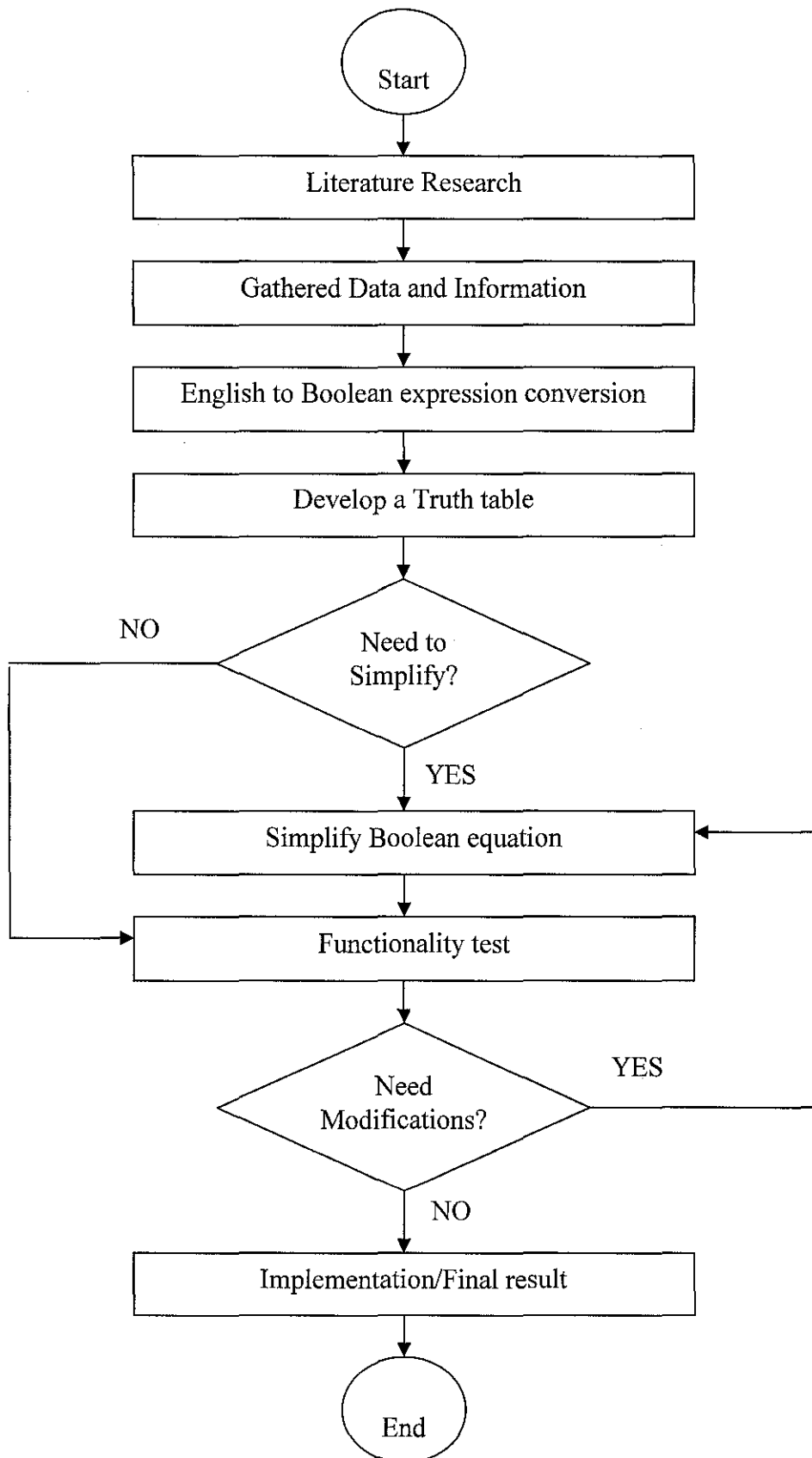


Figure 3.1: Project Flow chart

## CHAPTER 4

### RESULT AND DISCUSSION

#### 4.1 Circuit Design for Ceiling Light

The first part in designing logic circuit is by applying the conversion of English sentences to Boolean expression.

1. Ceiling light inside an automobile will only light off when all doors are closed and manual switch is at AUTO position.
2. When manual switch is at ON position the light will always on.
3. When the manual switch is at OFF position the light will always off.

First of all, variables must be identified. There are five doors including the boot and a switch that can be positioned at ON, Auto, OFF. Firstly the doors condition should be solved since there are many variables. In binary form, closed door is represented as '1' and open door by '0'. For the output (P), light on is represented as '1' and light off is '0'. From this point, it is already predicted that the light is on for most condition and is off for only one condition. In order to avoid difficulty, it is better to reverse the output once it is already occurred. Below is an example of truth table for the doors. Car doors are represented as B (driver's door), C (co-driver's door), D (right back door), E (left back door), and K (boot).

B	C	D	E	K	P	P'
0	0	0	0	0	1	0
0	0	0	0	1	1	0
0	0	0	1	0	1	0
0	0	0	1	1	1	0
0	0	1	0	0	1	0
0	0	1	0	1	1	0
0	0	1	1	0	1	0
0	0	1	1	1	1	0
0	1	0	0	0	1	0
0	1	0	0	1	1	0
0	1	0	1	0	1	0
0	1	0	1	1	1	0
0	1	1	0	0	1	0
0	1	1	0	1	1	0
0	1	1	1	0	1	0
0	1	1	1	1	1	0
1	0	0	0	0	1	0
1	0	0	0	1	1	0
1	0	0	1	0	1	0
1	0	0	1	1	1	0
1	0	1	0	0	1	0
1	0	1	0	1	1	0
1	0	1	1	0	1	0
1	0	1	1	1	1	0
1	1	0	0	0	1	0
1	1	0	0	1	1	0
1	1	0	1	0	1	0
1	1	0	1	1	1	0
1	1	1	0	0	1	0
1	1	1	0	1	1	0
1	1	1	1	0	1	0
1	1	1	1	1	0	1

Table 4.1: Truth table for doors

Hence the Boolean expression is:

$$P' = m(32)$$

$$P' = B \& C \& D \& E \& K$$

$$P' = BCDEK$$

$$P = (BCDEK)'$$



Since this expression is true for only one condition, therefore algebraic simplification or Karnaugh map are not used. The final result for the doors condition can easily be obtained. At this stage, the equation is then converted to logic gate for the implementation. Below is the structure of the gate:

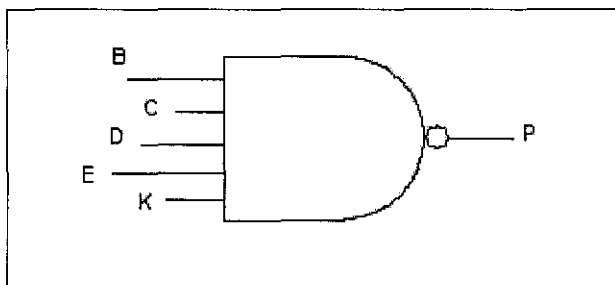


Figure 4.1: Logic gate for doors

The logic circuit for doors is already prepared. Now it is time to combine this with the manual switch. First, assume that P is input for the light. Then manual switch is represented as A. The binary condition for the switch is '1' at ON and '0' at AUTO mode. The OFF condition can be removed from the logic since it is connected directly to the output itself. Assume that F represents the output for light. By using conversion of English sentences to Boolean expressions, the following truth table is developed:

A	P	F
0	0	0
0	1	1
1	0	1
1	1	1

Table 4.2: Truth table for switch and function of doors

$$F = m(2, 3, 4)$$

$$F = A'P + AP' + AP$$

$$F = A'P + A(P' + P) \dots\dots\dots 1.a$$

$$F = A'P + A \dots\dots\dots 5.f$$

$$F = A + P$$

$$F = A + (BCDEK)'$$

This equation can also be obtained by using Karnaugh map. Now it is clear what the final equation. Since this equation uses the minimum gates and hence no need to be simplified. Below is the diagram for the logic combinations.

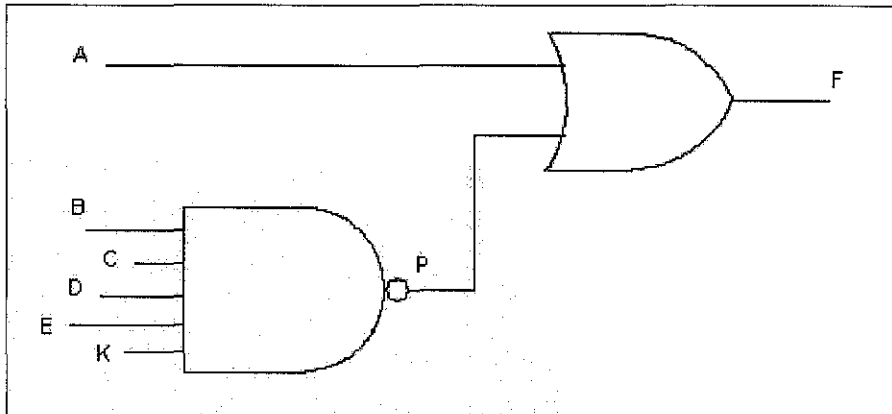


Figure 4.2: Logic gates design for switch and doors

The logic gates shown in the above figure is still not the final circuit. However, the structure of the gates must be maintained. At this stage, it is up to the user to decide what kind of implementation is needed for the final result. For this project, it will be implemented on PLDs.

#### 4.2 Circuit Design for Buzzer

This design is a further application of the circuit that controls ceiling light in automobile. The procedure is still same but the conditions are different. For this design the variables involve are B (driver's door), G (head light), H (hand brake), I (ignition key) and output J (buzzer). The final implementation is a combination of the previous design which is used to show how PLDs can be utilized to eliminate messy wiring and logic gates consumption.

As usual, the first thing to do for designing the circuit is that, the English sentences must be converted to Boolean expression. This English sentence is an example of what is the purpose of the project. Below is the example of this conversion from English sentences to Boolean expression.

1. Buzzer will alarm if hand brake is not pulled or head light is on when driver left the car after parking.
2. Buzzer will not buzz if the ignition key remains in the switch or the driver door is still closed.

From these sentences, truth table can be developed. Before that, all variable must be identified in binary. '1' represents that the head light is on and '0' for off. Similarly, '1' represents the state that the hand brake is pulled and '0' when it is released. For ignition key, '1' represents the state when key is still in ignition and vice-versa. Closed driver's door is represented by '1' and open by '0'. The output is '1' when buzzer starts buzzing and vice-versa. Below is the result for the truth table:

G	H	I	B	J
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Table 4.3: Truth table for buzzer

$$J = m(1, 9, 13)$$

$$J = G'H'I'B' + GH'I'B' + GHI'B'$$

$$J = I'B' (G'H' + GH' + GH) \dots\dots\dots 4.a$$

$$J = I'B' [G'H' + (G(H + H'))] \dots\dots\dots 1.a$$

$$J = I'B' [G'H' + G(1)]$$

$$J = I'B' [G'H' + G] \dots\dots\dots 5.f$$

$$J = I'B' (G + H') \dots\dots\dots 4.a$$

$$\mathbf{J = GI'B' + H'I'B'}$$

This is the final equation after the simplification. The same equation can also be obtained by using the Karnaugh map method.

J	GH	00	01	11	10
I B	00	1		1	1
	01				
	11				
	10				

Table 4.4: Four Variables K-map

$$\mathbf{J = GI'B' + H'I'B'}$$

The final gates combination is shown in the following diagram:

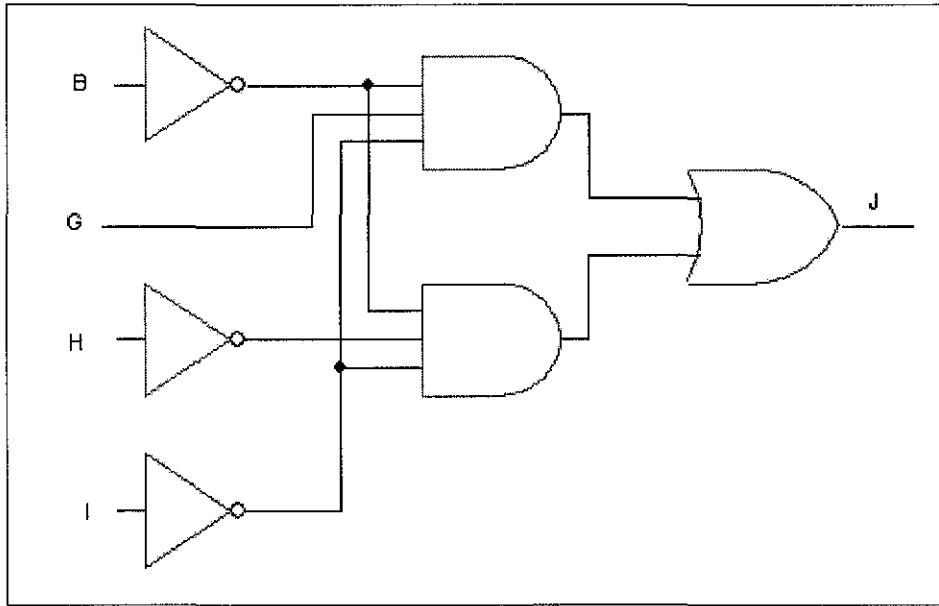


Figure 4.3: Logic gate for buzzer design

At this stage, the logic circuit controls two outputs. One output for the light and one for the buzzer. The circuit is already simplified and ready to be implemented in the PLDs. Below is the diagram of the combination of the circuits.

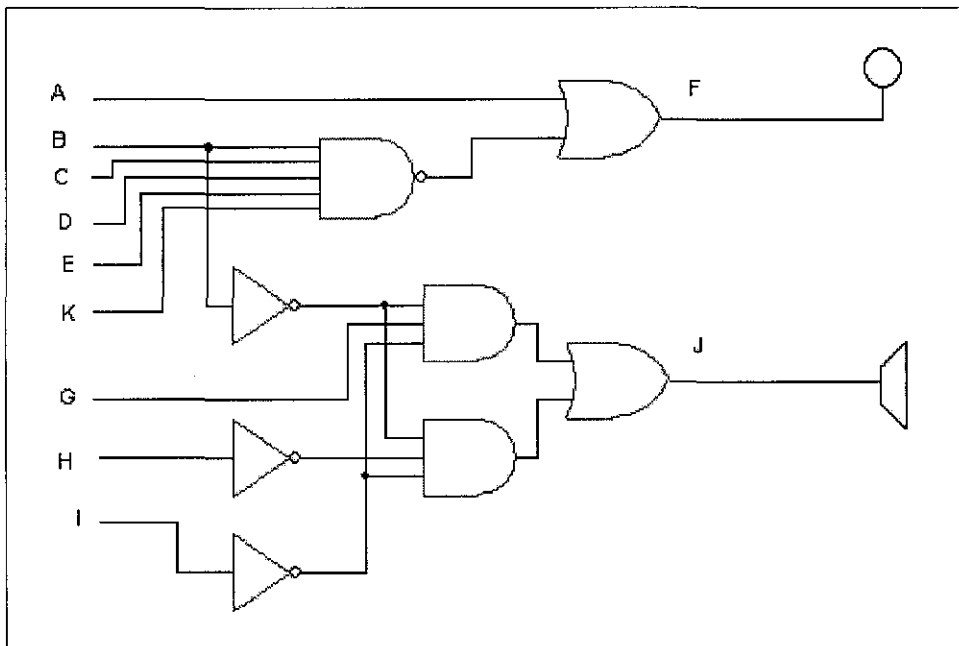


Figure 4.4: Final design of combination circuit.

### 4.3 Implementation

Since the result for final circuit is already obtained, the next task is how to implement the circuit to a suitable method such as PLDs. PLDs is a very useful method to implement the circuit. PLDs can greatly reduce the number of logic devices which appear in final design. PLDs can also reduce chip count, therefore saving money and often improving the speed of the design. Other advantage is that it is easy to change if there is a demand of alteration. PLDs is the best method to implement logic gates. [11]

This PLDs chip is very reliable because it is rewritable as many times as user desired. Below is a diagram showing the circuit<sup>4</sup> in PLDs.

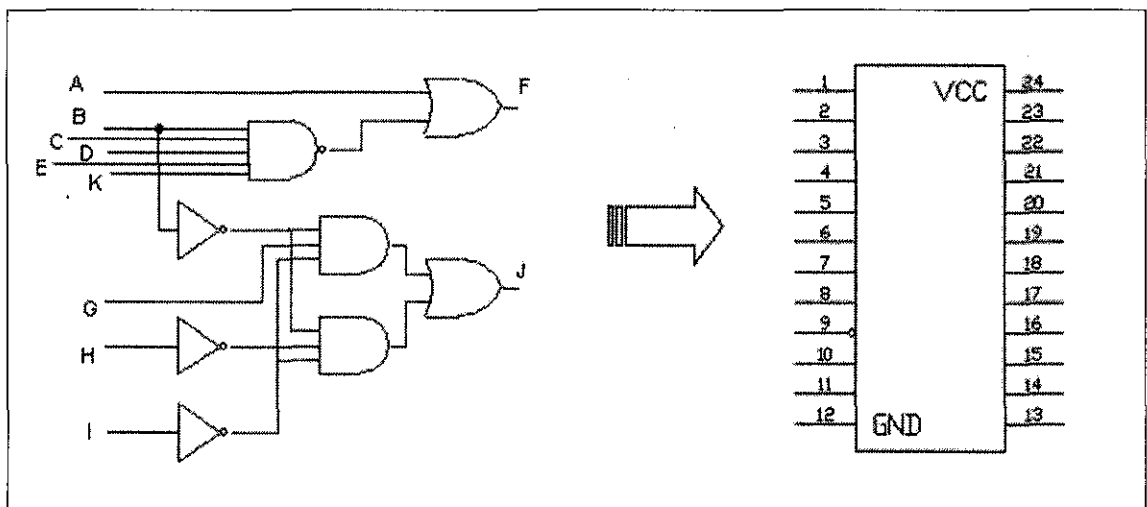


Figure 4.5: Implementation using PLDs

By using PLDs, digital circuit can be created or programmed inside the Integrated Circuit (I.C) chip. In the figure 4.5, the right diagram is the inner logics inside the integrated circuit which is on the left side. This technique will save a lot of wiring and number of logic gates.

<sup>4</sup> the circuit is programmed and then written on PLDs

## **CHAPTER 5**

### **CONCLUSION AND RECOMMENDATION**

#### **5.1 Conclusion**

Implementation of the project is a good exposure to the student to discover new skills, new ideas and apply the knowledge gained in the project, throughout the real working life problem. The main objectives for this project; study the similarity between the operations of logical connectives and the operation of switching devices and simplify the circuit by using Boolean algebra approach are already met. Since the logical circuit and Boolean algebra are much related, this approach is very suitable for designing any digital circuit involving logics. Theory and application of the software hardware design are the key factors that lead to the success of this project.

For overall conclusion of this project, circuit can be easily design with a standard procedure provided by Boolean application. This standard procedure leads to the final circuit design which is already simplified. A lot of time can be saved from this approach instead of trial and error method. Hence, the number of gates used is minimized and the outcome is cost saving. Less number of gates used directly reduce the number of wiring and then easy for maintenance and modification.

## 5.2 Recommendation

### 5.2.1 Software

Throughout the whole work of this project, it is found that to derive a Boolean expression was quite challenging. It was noticed that, if number of variables become larger, then the number of conditions also large. That makes it difficult to find the final Boolean expression.

$$\text{Condition} = 2^k$$

k = number of variable

The condition will become larger if k is increased. It is also observed that, the Boolean approach, K-map, applicable for a maximum of 6 variables is introduced. When the number of variables is large, a new method is introduced to overcome this problem. The result is to use the Quine-McCluskey method to solve equation for larger number of variables.

For future work expansion, it is preferred that a computer software could be design by applying the Quine-McCluskey method for the ease of Boolean equation. The rough idea is that, by using Visual Basic programming, for any number of variables, just key in the truth table, computer will automatically generate the result (Boolean equation).

### 5.2.2 Hardware

PLDs is not the very high-end technology at this time but it's quite useful when dealing with logic gates. However, the problem is when programming the PLDs, there are too many information that should be keyed in. A new technology that can overcome this problem is a PIC microcontroller. This microcontroller is very reliable and it's programming also simple. This kind of chip can perform a lot of operations at the same time. PIC microcontroller has many advantages than the PLDs.



## REFERENCES

- [1] Charles H. Roth Jr, *Fundamental of Logic Design 4<sup>th</sup> Edition*, PWS, 1995.
- [2] J.R. Nowicki and L.J Adam, *Digital Circuit*, 1990.
- [3] Alan B. Marcovitz, *Introduction to Logic Design*, McGraw Hill, 2002.
- [4] Stephen Brown and Zvonko Vranesic, *Fundamental of Digital Logic Design*, McGraw Hill, 2003.
- [5] Marcovitz, *Introduction to Logic Design*, McGraw Hill, 2002.
- [6] Roger L. Tokheim, *Digital Electronics - Principles and Applications*, 5<sup>th</sup> Edition, 1998.
- [7] John M. Yarbough, *Digital Logic – Applications and Designs*, WEST, 1997.
- [8] Tsutomu Sasa, *Switching Theory For Logic Synthesis*, KAP, 1999.
- [9] Richard S. Sandige, *Modern Digital Design*, McGraw Hill, 1990.
- [10] Institute of Electrical and Electronics Engineers, *IEEE Standard Graphic Symbols for logic function (ANSI/IEEE Std 1991)*, New York, 1991.
- [11] *Programmable Logic Devices Manual*, Flight Electronics International Ltd, 1994.

## **Internet**

*[www.play-hookey.com/digital](http://www.play-hookey.com/digital)*

*[www.williamson-labs.com/480\\_logic.htm](http://www.williamson-labs.com/480_logic.htm)*

*<http://hyperphysics.phy-astr.gsu.edu/hbase/electronic/diglog.html>*

*<http://renoir.vill.edu/mnt/a/cassel/html/1200/tch3.1-2.html>*

*[www.spsu.edu/cs/faculty/bbrown/web\\_lectures/transistors](http://www.spsu.edu/cs/faculty/bbrown/web_lectures/transistors)*

# APPENDIX

## 1. PLDs Coding

;PALASM Design Description

;----- Declaration Segment -----

TITLE fyp

PATTERN fyp

REVISION

AUTHOR e1616

COMPANY utp

DATE 10/29/03

CHIP \_e1616 PALCE20V8

;----- PIN Declarations -----

PIN 1	A	COMBINATORIAL ; INPUT
PIN 2	B	COMBINATORIAL ; INPUT
PIN 3	C	COMBINATORIAL ; INPUT
PIN 4	D	COMBINATORIAL ; INPUT
PIN 5	E	COMBINATORIAL ; INPUT
PIN 6	K	COMBINATORIAL ; INPUT
PIN 7	G	COMBINATORIAL ; INPUT
PIN 8	H	COMBINATORIAL ; INPUT
PIN 9	I	COMBINATORIAL ; INPUT
PIN 12	GND	; INPUT
PIN 15	/F	COMBINATORIAL ; OUTPUT
PIN 16	/J	COMBINATORIAL ; OUTPUT
PIN 24	VCC	; INPUT

;----- Boolean Equation Segment -----

EQUATIONS

F=/(B\*C\*D\*E\*K) + A ;NAND and OR gates

J=I\*/B\*(G+/H) ;INVERTER OR AND

;----- Simulation Segment -----

SIMULATION

TRACE\_ON A B C D E F G H I J K

;logic principles

SETF /K /A /B /C /D /E

CHECK F

SETF /K /A /B /C /D E

CHECK F

SETF /K /A /B /C D /E

CHECK F

SETF /K /A /B /C D E

CHECK F

SETF /K /A /B C /D /E

CHECK F

SETF /K /A /B C /D E

CHECK F

SETF /K /A /B C D /E

CHECK F

SETF /K /A /B C D E

CHECK F

SETF /K /A B /C /D /E

CHECK F

SETF /K /A B /C /D E

CHECK F

SETF /K /A B /C D /E

CHECK F

SETF /K /A B /C D E

CHECK F

SETF /K /A B C /D /E

CHECK F

SETF /K /A B C /D E

CHECK F

SETF /K /A B C D /E

CHECK F

SETF /K /A B C D E

CHECK F

SETF /K A /B /C /D /E

CHECK F

SETF /K A /B /C /D /E

CHECK F

SETF /K A /B /C /D /E

CHECK F

SETF /K A /B /C /D /E

CHECK F

SETF /K A /B C /D /E

CHECK F

SETF /K A /B C /D /E

CHECK F

SETF /K A /B C D /E

CHECK F

SETF /K A /B C D E

CHECK F

SETF /K A B /C /D /E

CHECK F

SETF /K A B /C /D /E

CHECK F

SETF /K A B /C D /E

CHECK F

SETF /K A B /C D E

CHECK F

SETF /K A B C /D /E

CHECK F

SETF /K A B C /D E

CHECK F

SETF /K A B C D /E

CHECK F

SETF /K A B C D E

CHECK F

SETF K /A /B /C /D /E

CHECK F

SETF K /A /B /C /D E

CHECK F

SETF K /A /B /C D /E

CHECK F

SETF K/A/B/CDE  
CHECK F  
SETF K/A/B C/D/E  
CHECK F  
SETF K/A/B C/D E  
CHECK F  
SETF K/A/B CD/E  
CHECK F  
SETF K/A/B CDE  
CHECK F  
SETF K/A B/C/D/E  
CHECK F  
SETF K/A B/C/D E  
CHECK F  
SETF K/A B/C D/E  
CHECK F  
SETF K/A B C/D/E  
CHECK F  
SETF K/A B C/D E  
CHECK F  
SETF K/A B C D/E  
CHECK F  
SETF K/A B C D E  
CHECK /F  
SETF K A/B/C/D/E  
CHECK F  
SETF K A/B/C/D E  
CHECK F  
SETF K A/B/C D/E  
CHECK F  
SETF K A/B/C D E  
CHECK F  
SETF K A/B C/D/E  
CHECK F  
SETF K A/B C/D E  
CHECK F

CHECK F  
SETF K A /B C D /E  
CHECK F  
SETF K A /B C D E  
CHECK F  
SETF K A B /C /D /E  
CHECK F  
SETF K A B /C /D E  
CHECK F  
SETF K A B /C D /E  
CHECK F  
SETF K A B /C D E  
CHECK F  
SETF K A B C /D /E  
CHECK F  
SETF K A B C /D E  
CHECK F  
SETF K A B C D /E  
CHECK F  
SETF K A B C D E  
CHECK F

SETF /G /H /I /B  
CHECK J  
SETF /G /H /I B  
CHECK /J  
SETF /G /H I /B  
CHECK /J  
SETF /G /H I B  
CHECK /J  
SETF /G H /I /B  
CHECK /J  
SETF /G H /I B  
CHECK /J  
SETF /G H I /B  
CHECK /J

SETF /G H I B  
CHECK /J  
SETF G/H/I/B  
CHECK J  
SETF G/H/I/B  
CHECK /J  
SETF G/H I/B  
CHECK /J  
SETF G/H I B  
CHECK /J  
SETF G H/I/B  
CHECK J  
SETF G H/I B  
CHECK /J  
SETF G H I/B  
CHECK /J  
SETF G H I B  
CHECK /J

TRACE\_OFF

;------