Assignment

CSCU9V4

Systems

The 'Why-Aye' Pad Controller

Student ID 2628030

Spw00004

Spring/2019

# Task Marking Scheme Marks distribution

**PART I: Systems**

Build the truth table **/5**

1. Derive algebraic expressions: **/20**
   1. Sum-of-minterms expressions /5
   2. Simplification of the expression for Clock Wise /15
2. Draw the CW circuit **/5**
3. Convert CW expression to NAND-only form **/10**
4. Demonstrate NAND-only version is correct with a truth table **/5**
5. Draw the NAND-only version of the CW circuit **/5**

**PART 2: C Programming**

1. Controller Simulator **/20**
2. Simulate Sum-of-Minterms Boolean Expressions **/15**
3. Draw on a Bitmap File **/10**
4. Produce a well presented submission **/5**

## TOTAL **/100\***

(\*This assignment constitutes 40% of the overall module assessment.)

**Part 1: Systems**

**Task 1: Build the Truth Table**

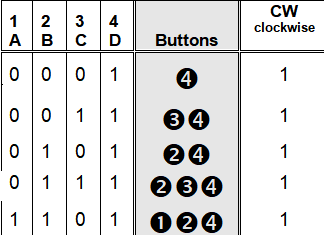
Create the truth table for the controller. Assume that the buttons when pressed individually or in combination create a 4 bit binary number, between 0000 (010) and 1111 (1510). These four bits are represented for convenience as A,B,C and D as shown below, with D representing the least significant bit. The code for each button combination can be taken to be its ordinality in the table given above.

The table will thus have this form:

button combination code effect on sprite dictated by 0/1 values on these six wires

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1A** | **2B** | **3C** | **4D** | **Buttons** | **CW**  **clockwise** | **ACW**  **anti-clockwise** | **L**  **left** | **R**  **right** | **U**  **up** | **D**  **down** |
| 0 | 0 | 0 | 0 | none | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | ➍ | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | ➌ | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | ➌➍ | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | ➋ | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | ➋➍ | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | ➋➌ | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | ➋➌➍ | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | ➊ | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | ➊➍ | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | ➊➌ | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | ➊➌➍ | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | ➊➋ | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | ➊➋➍ | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | ➊➋➌ | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | ➊➋➌➍ | 0 | 0 | 0 | 0 | 0 | 0 |

**Task 2: Deriving Sum-of-Minterms Boolean Expressions**



CW = 

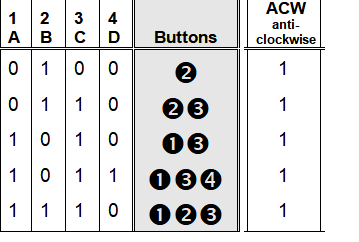
CW = 

CW = 

CW = 

CW = 

CW = ++++



ACW = 

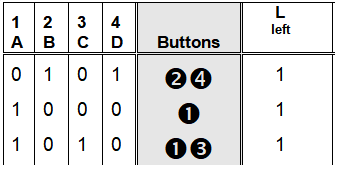
ACW = 

ACW = 

ACW = 

ACW = 

ACW = ++++

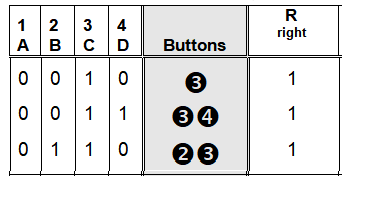


L= 

L= 

L= 

L = ++

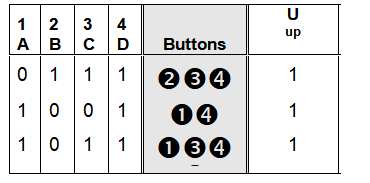


R= 

R= 

R= 

R = ++

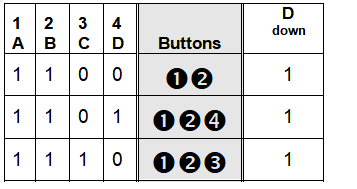


U= 

U= 

U= 

U= ++



D= 

D= 

D= 

D= ++

a) Using your truth table from Task 1, create six **sum-of-minterms** expressions, one for each output column:

1. CW = ++++

2. ACW = ++++

3. L= ++

4. R= ++

5. U = ++

6. D = ++

b) Using **Boolean Algebra** simplify the minterm expression for **CW** (clockwise) as far as you can, showing each major step that you take. The rule you are applying at each step should also be made clear (e.g. "distributive law", "", etc).

CW = ++++

CW = ()+()+ ~ Distributive law / ( = 1) complement law

CW = (1)+(1)+ ~ (A.1 = A) Identity law

CW = ++

CW = ()+ ~ Distributive law

CW = (1)+ ~ ( = 1) complement law

CW = + ~ (A.1 = A) Identity law

CW = () ~ Distributive law

CW = () ~ () = () Distributive law

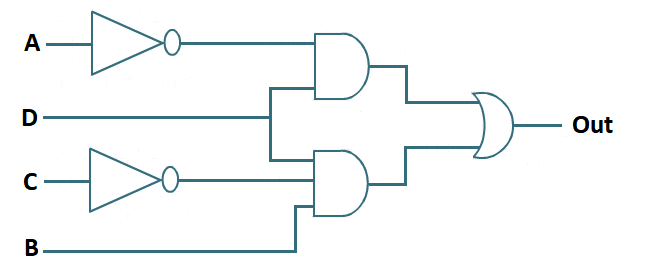
( = 1) complement law

CW = ()

CW = 

**Task 3: Draw the circuit for simplified CW logic/expression**

Using your simplified expression for the **CW** (clockwise)logic **only**, draw the logic circuit using AND, OR, NOT and other logic gates, as required.



**Task 4: Convert the CW expression to use NAND gates only**

Use **de-Morgan's theorem** to convert your expression for the CW circuit so that it uses **NAND gates only.** You should clearly show each step that you take in doing this in your report.

CW = 

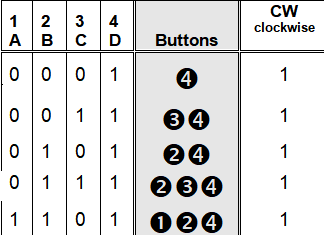
CW =  ~ Invert both sides of the gate

CW =  ~ Invert expression

CW =  ~ Change the sign (operator) from + to .

**Task 5: Demonstrate that the NAND version of CW is correct**

Use a truth table to show that the NAND version you now have for CW agrees with the original output. Your table should have the general form shown below:

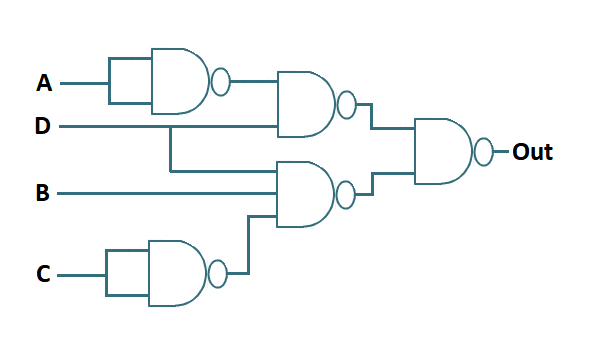


A, B, C and D are the original inputs and column number 1 is the column for CW (clockwise) taken from the table created in Task 1. Columns 2 to 5 (you will probably need more columns than this) represent the steps taken as you incrementally deduce (from the left to right) the results for your **NAND version** of the expression. You should have sufficient intermediate steps in your table (for example, showing , then  as separate columns) with the final, right-most column representing the final value for CW. This should match up with the original CW in column 1 (if it does not, and you cannot find where you have gone wrong, you will still get credit for the attempt).

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1A** | **2B** | **3C** | **4D** | **CW**  **clockwise** |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |

**Task 6: Draw the NAND-only version of the CW circuit**

Using the NAND-only circuit expression derived in Task 4, draw the NAND-only version of the CW circuit. Use the same basic format for the diagram as suggested in Task 3. If task 5 shows up an error in your NAND-only version, you should still draw the circuit, errors included; you will get credit for a correct rendition of the expression.



**Part 2: C Programming**

In Part 2 of the assignment, you are required to build a simulator for 'Why-Aye' Pad Controller using C programming. There are a number of tasks within this part with increasing difficulty.

**NOTE: Programs must compile to be considered for assessment.** Generally speaking, work is graded primarily along the following criteria:

* Scope – the extent to which code implements the features required/specified.
* Correctness – the extent to which code is consistent with the specs and bug-free.
* Design – the extent to which code is well written, i.e. clearly, efficiently, elegantly, or logically.
* Style – the extent to which code is readable, e.g. comments, indentation, apt naming.

**Task 7: ‘Why-Aye’ Pad Controller simple simulator**

In this task you need to write a simple ‘C’ program that performs following operations:

1. Accept input from your keyboard either numbers (1,2,3,4) or keyboard arrow keys representing four buttons in the controller.
2. On pressing different combinations of keys it should display texts on the console as given in Figure 1 on page 3.
3. Improve your sub-task 7.b) by displaying graphical representation on the console. For example, if your key combination indicates ‘Move left’ you can display a **←** shape (which is composed of a number of simple characters like ‘\*’, ‘+’, ‘-‘, ‘x’, or ‘<’) on your console. Be innovative for different outputs.

**Task 8: Simulate Sum-of-Minterms Boolean Expressions**

In this task you need to modify the program created in Task 7 and emulate the Sum-of-Minterms Boolean Expressions you have derived in Task 2. You need to follow the instructions below:

1. Your keyboard input should be stored in four Boolean variables, such as A, B, C, D
2. You should write equations for CW, ACW, L, R, U & D using the variables in previous step and C logical operators.
3. Finally use the output of CW, ACW, L, R, U & D to produce the same output as shown in Figure 1.

**Task 9: Draw on a Bitmap File**

[This is an optional component and should only be attempted after other tasks are complete.]

Create a bitmap image file of a plain colour (called *“mydraw.bmp”*). Modify the bitmap file to let it display the graphics you draw in the console window in Task 8. You should not change the file name.

Apart from doing some extra reading on your own, you may refer to the following webpage about how to write code to read, write and crop BMP image files:

<https://engineering.purdue.edu/ece264/17au/hw/HW15>

**Task 10 Produce your submission report**

The tasks outlined above should be carried out and your results presented in the form of a brief report. This means word processing text and creating diagrams using a drawing tool, and not submitting hand-drawn diagrams and tables. Your submission should be well-presented, spell checked where necessary and proof read, and in the order of the tasks requested. Up to 5% of the marks are available for fulfilling this criteria. Please do not forget to include the Mark Sheet from this assignment as the first page of your submission.

**Outcomes**

This work is designed to test your understanding of logic circuits, truth tables, Boolean algebra and your skill in C programming. Marks are awarded for correct and accurate use of these logic design techniques, efficient coding and producing a clear and structured report.

**Submission**

You will need to submit your work on Canvas as a zipped file bearing your university username (3 letters + 5 digits, e.g., xyz00001.zip). This zip file should contain a folder also named after your username. Place all the files created during the assignment in this folder.

* This is an individual piece of work.
* **Please ensure that the cover Mark Sheet (i.e page 1) is included in the report.**

**The deadline for this assignment is**

**5:00 pm on Friday 5th April 2019**