CSCU9V4: Systems

Assignment 1 - Spring 2021
The Logic Circuit Controller

Student ID:

Task Marking Scheme	Marks distribution

PART I: Systems

1. Build the truth table .... **/5**
2. Derive algebraic expressions:
    a. Sum-of-minterms expressions .... **/12**
    b. Simplification of the expression for Clock Wise .... **/18**
    c. Simplification of the expression using K-Map .... **/15**
3. Draw the CW circuit .... **/10**
4. Convert CW expression to NAND-only form .... **/15**
5. Demonstrate NAND-only version is correct with a truth table .... **/10**
6. Draw the NAND-only version of the CW circuit .... **/10**
7. Produce a well presented submission .... **/5**

TOTAL **/100***

(*This assignment constitutes 25% of the overall module assessment.)

Assignment 1 Semester: Spring Year: 2020-21

Module Title:	*Systems*	**Module Code:**	CSCUMV4
Tutor:	*Mr. Gamal Mohamed*	**Division:**	*Computing Science*

Submission date: **4:00 pm, Friday 5th March 2021**

Nature of Task & Brief Description:

The 'Why-Aye' Pad controller has a six-core wire connecting it to the Wi-i console and each wire carries a Boolean/binary value of 0 or 5 volts (or 0 and 1 for convenience). The combination of six values on the wires controls the output. Inside the controller there is some circuitry to create the correct 0 or 1 value on each wire for each button combination and it is your job to design part of that circuitry.

Read descriptions of all TASKS and the assignment brief before you start working on this.

Word Count:	n/a	**Assessment Weighting:**	**25 %**
Objectives	**This assignment is designed to address the following learning outcome of the module** : ▪ *Design and manipulate simple logic circuits using* truth tables, *Boolean algebra and K-Map.* ▪ *Design implement circuits with universal NAND gates.*		
Task Details:	*The finished assignment must:* *a) Complete all tasks as described below.* *b) Clearly be identified and be the student's own work.*		

Assignment Specification

The Wi-i™ *Controller*

Number 910 Fulchester Road in Newcastle is a famous address in the games industry. Whilst other provincial games manufacturers took on the big guns in the games industry with products such as the *Phony PlaySatan* to compete with the *iPhone*, and the Welsh assembly-supported *X-Cist 360* and '*GameBoyo*' (as 'leeked' to the press recently), a small business operating out of a terraced house in the North East of England decided to call its revolutionary new product after the digits in their house number. Thus was born the *NineTenDoor Wi-i* (pronounced, as expected in Newcastle, "*Why-Aye*"). This somewhat radical new console was designed to be used by inebriated players, and so had a ruggedised minimalist controller with only 4 buttons, as shown opposite. This was dubbed the '*Eee!*', reflecting the uncalled-for enthrall when first seen by a drunk game-player.

Games soon appeared for this new console, such as "*You're Me Best Mate!*" a bus-station-platform game, and "*Down The Bigg Market*", a multi-person, free-for-all, shoot and stab 'em up. "*Are You Looking At Me?*" and "*You've Spilled Me Pint!*" soon followed, followed by ports of popular games such as "*Spore*" but with just the first level available, as deemed appropriate. *Shoot 'em Up's*, educational games warning of the perils of drunken drug abuse.

The controller's buttons are designed to be used singly, or in combination with one or two of the others. There are 14 possible combinations with 2 'no operations' - it was decided that pressing all four simultaneously was probably not only impossible for a drunkard, but insensible in terms of game play. The buttons control a sprite in a game (we're not concerned with the game's details here) in the following way:

no buttons	nothing happens!
➊	move left (L)
➋	rotate anti-clockwise (ACW)
➌	move right (R)
➍	rotate clockwise (CW)
➊+➋	move down (D)
➊+➍	move up (U)
➊+➌	rotate anticlockwise (ACW) & move left (L) simultaneously
➋+➍	rotate clockwise (CW) & move left (L) simultaneously
➋+➌	rotate anti-clockwise (ACW) & move right (R) simultaneously
➌+➍	rotate clockwise (CW) & move right (R) simultaneously
➊+➋+➌	rotate anticlockwise (ACW) & move down (D) simultaneously
➊+➌+➍	rotate anticlockwise (ACW) & move up (U) simultaneously
➊+➋+➍	rotate clockwise (CW) & move down (D) simultaneously
➋+➌+➍	rotate clockwise (CW) & move up (U) simultaneously
➊+➋+➌+➍	nothing happens!

Figure 1: Button control details

YOUR TASKS

The controller logic circuit needs to be designed. The controller has a six-core wire connecting it to the Wi-i console and each wire carries a boolean/binary value of 0 or 5 volts (or 0 and 1 for convenience). The combination of six values on the wires controls the sprite. Inside the controller there is some circuitry to create the correct 0 or 1 value on each wire for each button combination and it is your job to design part of that circuitry. This assignment consists of multiple tasks as described below.

Each task has a maximum percentage mark allocated to it - this is repeated on the front sheet of this assignment (which should be attached to your final submission). You will be given full credit for each task done correctly, and pro-rata credit where you have not managed to complete a task or have made a mistake/omission.

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 (0_{10}) and 1111 (15_{10}). 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*

A	B	C	D	Buttons	CW clockwise	ACW anti-clockwise	L left	R right	U up	D down
0	0	0	0	none						
0	0	0	1	❶						
0	0	1	0	❷						
0	0	1	1	❸						
0	1	0	0	❹						
0	1	0	1	❶+❷						
0	1	1	0	❶+❹						
0	1	1	1	❶+❸						
1	0	0	0	❷+❹						
1	0	0	1	❷+❸						
1	0	1	0	❸+❹						
1	0	1	1	❶+❷+❸						
1	1	0	0	❶+❸+❹						
1	1	0	1	❶+❷+❹						
1	1	1	0	❷+❸+❹						
1	1	1	1	❶+❷+❸+❹						

Task 2: Deriving Sum-of-Minterms Boolean Expressions

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 =

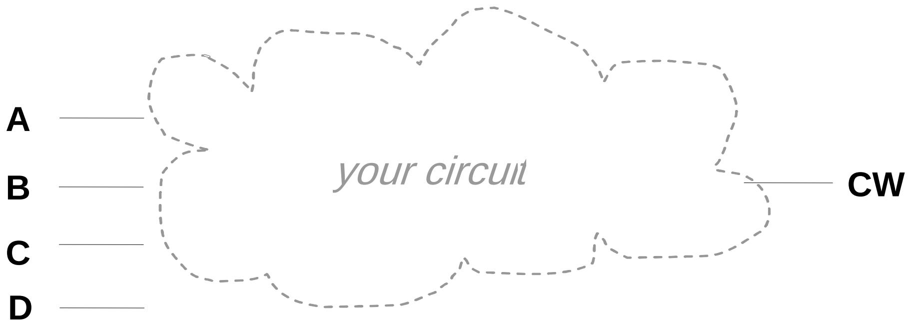
b) Using **Boolean Algebra** simplify the minterm expression for **CW** (clockwise) as far as you can, showing each major step that you take. The rule you are applying at each step should also be made clear (e.g. "distributive law", "$A + \overline{A} = 1$", etc).

c) Now using the **Karnaugh Map (K-Map)** simplify the minterm expression for origianal **CW** (clockwise) in Task 2.a) showing and describing the groupings etc. You may need to do further simplification with Boolean algebra in order to get the most optimised solution.

Hint: The answer from b) and c) will likely to be same.

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 (See task 7 for further instruction on how to draw). Your diagram should have the following format:



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.

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:

column number:				1	2	3	4	..	..	..	Last
A	**B**	**C**	**D**	**CW**							**CW**

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 last but one (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 $\overline{B}$, then $\overline{A \bullet \overline{B}}$ as separate columns) with the final, right-most column representing the final value for you NAND version 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).

Hint: Using a spreadsheet program such as Excel is a convenient way of building a truth table.

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.

Task 7 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. 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 K-map. Marks are awarded for correct and accurate use of these logic design techniques and producing a clear and structured report.

Submission

You will need to submit your work on Canvas as a PDF file bearing your university username (3 letters + 5 digits, e.g., xyz00001.pdf).

This is an individual piece of work.

Plagiarism

Work which is submitted for assessment must be your own work. Plagiarism means presenting the work of others as though it were your own. The University takes a very serious view of plagiarism, and the penalties can be severe (ranging from a reduced mark in the assessment, through a fail mark for the module, to expulsion from the University for more serious or repeated offences). We check submissions carefully for evidence of plagiarism and pursue those cases we find. Further details can be found here:
https://www.stir.ac.uk/about/professional-services/student-academic-and-corporate-services/academic-registry/academic-policy-and-practice/quality-handbook/academic-integrity-policy-and-academic-misconduct-procedure/

Late submission

If you cannot meet the assignment hand-in deadline and have good cause, please see **Mr. Gamal** to explain your situation and ask for an extension. Coursework will be accepted up to seven days after the hand-in deadline (or expiry of any agreed extension) but the mark will be lowered by three marks per day or part thereof. After seven days the work will be deemed a non-submission.