



Australia's
Global
University



ELEC 2141 Digital Circuit Design

In this assignment you will work on a combination circuit design with a real world application. It will involve applying all the techniques you have learnt over Weeks 1-4 in the course.

The gate input cost should be calculated for the simplified AND-OR-NOT expressions in the design that you implement. You can choose the technology to implement the circuit with (e.g. AND/OR, NAND, multiplexers, etc.). However your design for each module should aim at achieving optimized implementation. Simulate a schematic of your design using Xilinx SE.

Your final submission should include your design procedure, choice of technology implementation, and simulation results. Combine all scanned copies of your handwritten work and electronic documents into one pdf file for the submission on Moodle. This includes all design and simulation materials such as schematic diagrams, simulation outputs and testing fixtures. All Verilog code used to simulate your design will need to be submitted separately. Details of file names, structures to be used will be posted on Moodle.

In your submission file include a scanned copy of a completed and signed assignment submission form as the front page.

Any output of generative AI software used within your assignment must be attributed with full referencing. If the outputs of generative AI such as ChatGPT form part of your submission and is not appropriately attributed, it will be regarded as serious academic misconduct and subject to the standard penalties, which may include 00FL, suspension and exclusion.

*To cite: OpenAI (Year Accessed). ChatGPT. OpenAI. <https://openai.com/models/chatgpt/>

*Please note that the outputs from these tools are not always accurate, appropriate, nor properly referenced. You should ensure that you have moderated and critically evaluated the outputs from generative AI tools such as ChatGPT before submission.

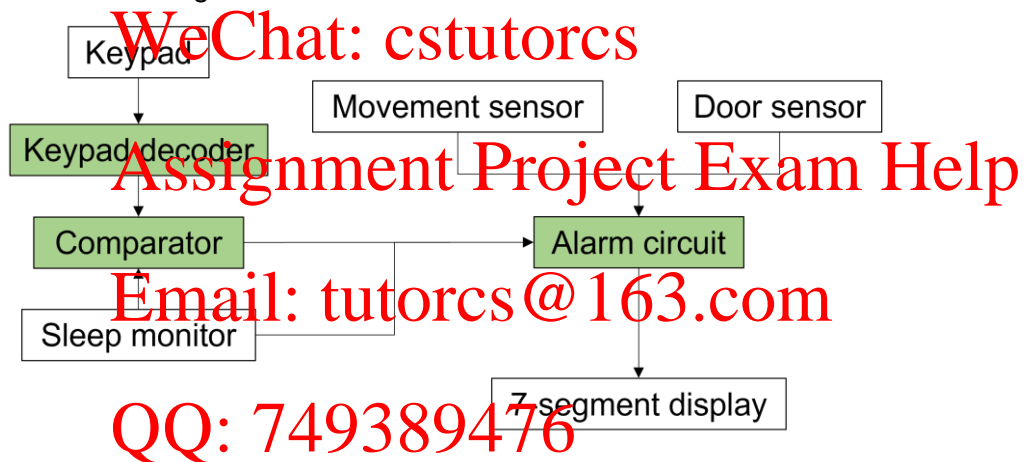
DUE DATE: 27th MARCH, 23:59 PM

Smart Homes for the Elderly

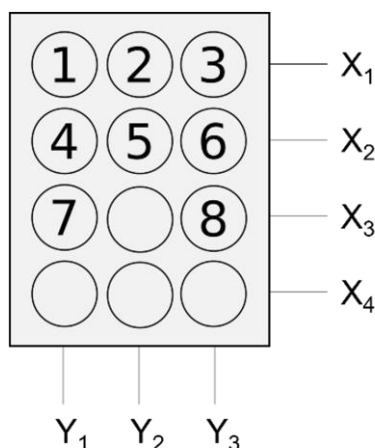
程序代写代做 CS编程辅导

An aged care provider close to UNSW has asked us to develop an automated system that enables them to unobtrusively monitor their elderly residents for risks such as falls and long periods of inactivity. They have acquired activity monitors, sleep monitors that their residents can wear plus installed sensors at their main door. They would like to install a monitoring system as shown in the block diagram below.

The system allows them to monitor whether a resident has slept for, whether the resident is moving, stationary or had a fall, whether a resident tries to access the main door without an accompanying authorised carer. You need to design an alarm circuit which generates alarms and notifications for their staff based on inputs from these sensors. Specifically, you will need to design combinational circuits that implement the three modules in green.



In the system, the user sets the resident's average sleep duration (1-8 hours) using a keypad as below. Each button is identified by row (X_i) and column (Y_i) signals. No two buttons may be pressed at the same time. The empty buttons cannot be pressed. The keypad decoder you need to design will convert the button press into a BCD number.



The sensors in the system provide the following outputs from the door sensor, activity band and sleep monitor as described in the table below. They are the inputs to the alarm circuit you need to design.

Sensor	Outputs
Door (D_1D_0)	Closed (00), authorised door open (11), unauthorised door access (10)
Movement (M_1M_0)	Stationary (00), moving (11), fall detected (10), no activity for 2+ hrs (01)
Sleep ($S[3:0], N$)	Sleep duration in 4 bits (0-15 hrs), sensor connected (Y/N)

The comparator generates a two-bit output CM coded as in the table below by comparing the average sleep duration set using the keypad to the sleep duration from the sleep monitor.

CM	Comparator output
00	Less than or equal to setpoint
01	More than setpoint

The system uses the comparator outputs from the sensors to generate the outputs as below.

Output	Door	Movement	Sleep	Sensor
Error (E) = 1 if all conditions across any of these rows are true, else 0	Closed	Moving	Sleep duration less than or equal to setpoint	Connected
			Sleep duration more than setpoint	Connected
	Authorised door open	Moving	Sleep duration less than or equal to setpoint	Connected
			Sleep duration more than setpoint	Connected
		Stationary	Sleep duration less than or equal to setpoint	Connected
			Sleep duration more than setpoint	Connected
Check (C) = 1 if all conditions across any of these rows are true, else 0	Closed	Fall detected	Sleep duration less than or equal to setpoint	Connected
			Sleep duration more than setpoint	Connected
		No activity for 2+ hrs	Awake	Connected
		Stationary	Sleep duration more than setpoint	Connected
	Authorised door open	Moving		Disconnected
				Connected
Alarm (A) = 1 if all conditions across any of these rows are true, else 0		Fall detected	Awake	Connected
	Unauthorised door access			
SEVEN 7-bit signal	Used to indicate the resident's status on a 7-segment display. The letter E on the display indicates an error in the system, C the resident needs to be checked upon and A there is an alarm that needs immediate attention			

To summarise, you need to design a combinational circuit with these three components:

1. A circuit that decodes the keys pressed in the keypad into a 4-bit digit to obtain the average sleep duration,
2. A magnitude comparator that compares the real-time sleep duration from the sleep monitor to the average sleep duration and
3. An alarm circuit that outputs E, C, A and SEVEN.

You can choose different ICs to implement each component with (AND/OR, NAND, multiplexers, etc.).



Note: A magnitude comparator determines the relative magnitude of two binary numbers, with applications in numerous digital systems. In general, a comparator performs a magnitude comparison of two binary numbers A and B and provides three outputs, $A < B$, $A = B$, $A > B$. Comparators are available in IC form with different bit comparing configurations, e.g. 4-bit, 8-bit. More than one comparator can also be connected in cascade arrangement to perform comparison of numbers of longer lengths.

To compare the two binary numbers, first the most significant bits (MSBs) are compared. If the MSBs are equal, then the next significant bits are compared. However, if the MSBs are not equal, then either A is greater than or less than B , hence further comparison is not needed. For example, when comparing two 2-bit numbers given by $A = A_1A_2$ and $B = B_1B_2$. If $A_1 \neq B_1$, then $A > B$ if $A_1 = 1$, $B_1 = 0$ or $A < B$ if $A_1 = 0$, $B_1 = 1$. If the MSBs are equal, then the next significant bits A_0 and B_0 are compared to decide whether $A < B$, $A = B$, $A > B$. The comparator overall generates three outputs, L, E and G, corresponding to whether $A < B$, $A = B$, $A > B$.

You will need to design a comparator that compares the four bits in BCD number and then use it to compare the average sleep duration set using the keypad to the patient's current sleep duration.

Optional: A hardware implementation of your design of the keypad decoder on a breadboard. If you are interested, please sign up at the link posted on Moodle so we can arrange access to open labs in week 5 where you can test your circuit. Note this option will be available to a maximum of 38 students only. Students who do not attend the open lab in Week 5 will no longer be eligible to receive bonus marks and a call will be made for other students interested in this option. You will be able to test your implementation using the BCD to 7-segment decoder you will design in the Week 4 lab. You will need to show your validated test design at the end of week 6. A hardware implementation of the keypad decoder will be worth up to 20% (of this assignment) bonus marks and will be marked on whether it performs as per the specifications and efficiency of design.

- Week 4 Friday 12pm: signups for hardware implementation available on Moodle.
- Week 5 Friday: Open labs for testing.
- Week 6 Friday: Marking.