CMPUT 329: COMPUTER ORGANIZATION AND ARCHITECTURE II
Homework # 3
Issue Date: November 11 2002
Due Date: November 27 2002

CMPUT 329 Honor Code

By turning in the exam solution for grading, I certify that I have worked all the solutions on my own, that I have not copied or transcribed solutions from a classmate, someone outside the class, or from any other source. I also certify that I have not facilitated or allowed any of my classmates to copy my own solutions. I am aware that the violation of this honor code constitutes a breach of the trust granted me by the teaching staff, compromises my reputation, and subjects me to the penalties prescribed in Section 26.1 of the University of Alberta 2002/2003 Calendar.

Edmonton, November, 2002.

StudentID:	Name:

Problem 1	/25
Problem 2	/25
Problem 3	/20
Problem 4	/15
Problem 5	/15
Total	/100

Problem 1 (25 points)

A sequential circuit has an input X and outputs Y and Z. YZ represents a 2-bit binary number equal to the number of pairs of adjacent 1's that have been received as inputs. For example, the input sequence 0110 contains one pair, the sequence 01110 contains two pair, and the sequence 0110111 contains three pairs of adjacent 1's. The circuit *resets* when the total number of pairs of 1's received reaches 4.

Examples:

Cycle:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$X =$	0	1	0	1	1	0	1	1	1	0	0	1	0	1	0	1	0	1	1
$Y =$	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0
$Z =$	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0

Cycle:	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
$X =$	1	0	1	1	0	1	1	0	0	1	0	1	1	1	1	1	1	1	1
$Y =$	0	0	0	0	0	0	1	1	1	1	1	0	0	1	1	0	0	0	1
$Z =$	0	0	0	1	1	1	0	0	0	0	0	0	1	0	1	0	0	1	0

- (5 points) Design a Moore state graph and state table for this circuit (be sure that the network resets as shown in examples). Remember that X is the input, and Y and Z are outputs.
- (10 points) Using the rules presented in class, provide a *good* state assignment for your state machine(s).
- (10 points) Find the minimal implementation for a two-level AND-OR implementation of the combinatorial circuit for the state machine that you designed using the minimal number of J-K flip-flops required to implement your machine.

Problem 2 (25 points) [Conceived by Paul Berube]

Design the controller logic to be used in a new musical toilet. The design must be a finite state machine. The purpose of the controller is to cause classical music to be played when (and only when) the trickle of water would be heard when someone is using a stall in a public washroom. Also, the toilet should automatically flush when the user leaves the stall. There should be a 7 second delay from when a person leaves the stall to when the flush starts.

Figure 1 shows a block diagram of the complete system. You will be asked to design the finite state machine that forms the control unit.

Each washroom stall has three sensors. One sensor detects the state of the door, which is either open or closed. Another sensor detects the presence of a person in the stall reporting

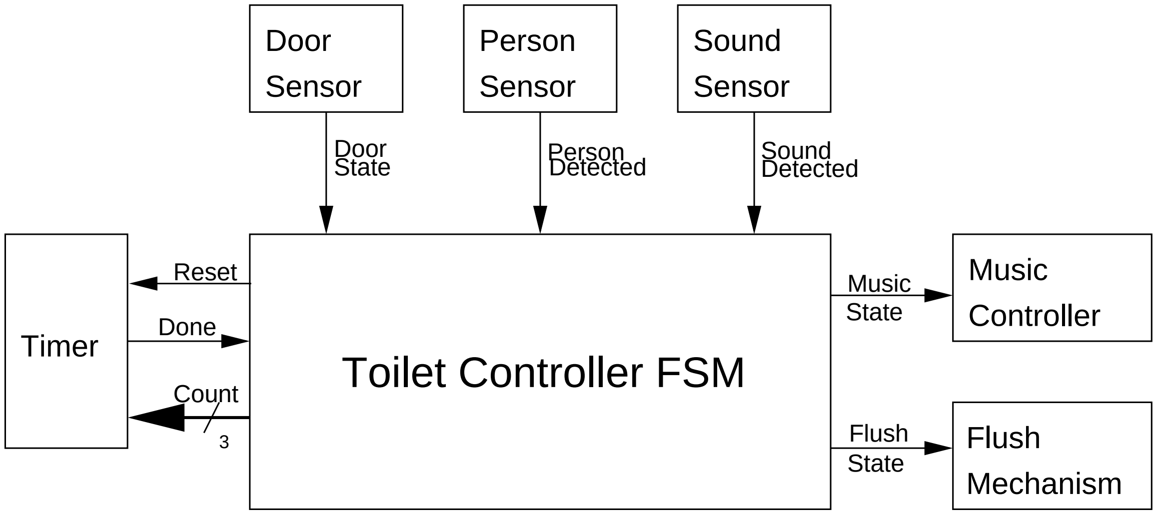


Figure 1: A block diagram for the musical toilet controller.

whether there is or there is not a person in the stall. The third sensor is a sound sensor that asserts a signal when the sound of trickling water is detected.

The controller has four outputs. One output controls a music-playing module. When the input to this module is asserted, classical music is played through a speaker. Another output controls the flush mechanism on the toilet. The toilet will flush while the input to the flush mechanism is asserted. Note that the input to the flush mechanism must be asserted for a period of time, to ensure a complete flush. In your design a flush period should be 5 seconds. However the flush mechanism should not start flushing until 7 seconds after the person leaves the stall (which is detected by the door sensor).

To assist with this task, you will use a standard counter powered by a 555 timer IC. The amount of time in seconds to count will be supplied by a 3-bit output from your controller. The counter works as follows: when the counter receives a RESET signal, it deactivates the Done signal and resets its internal counter to count for the period specifies in the Time input (in seconds). For example if the value in the 3-bit time input is 5, the timer's counter will count for the period of five seconds and then activate the Done signal. You should use the same timer unit to obtain the delay before a flush and the length of the flush.

- (10 points) Present your design in the form of a state transition table using meaningful symbolic names for each state of your finite state machine. If your machine becomes too complex, i.e., it will lead to excitation and output equations that have too many inputs to be minimized "by hand", you may decompose it into two separate state machines. If you do decide to design two separate state machines, you must also provide a block diagram specifying the interface between them.

- (5 points) Using the rules presented in class, provide a *good* state assignment for your state machine(s).

- (10 points) Find the minimal implementation for a two-level AND-OR implementation of the combinatorial circuit for the state machine that you designed using the minimal number of D flip-flops required to implement your machine.

Problem 3 (20 points)

For the following State Table:

Present State	**Next State** $X=0$	$X=1$	**Present Output** $X=0$	$X=1$
a	h	c	1	0
b	c	d	0	1
c	h	b	0	0
d	f	h	0	0
e	c	f	0	1
f	f	g	0	0
g	g	c	1	0
h	a	c	1	0

a. (10 points) Reduce the state table to a minimum number of states.

b. (10 points) You are given two identical sequential circuits which realize the above state table. One circuit is initially in state "a" and the other circuit is initially in state "c". Specify an input sequence of length three which could be used to distinguish between the two circuits, and give the corresponding output sequence for each circuit.

Problem 4 (15 points)

According to Mccluskey, a machine with 4 states has three non-equivalent state assignments.

a. (10 points) Enumerate the assignments for the following state table:

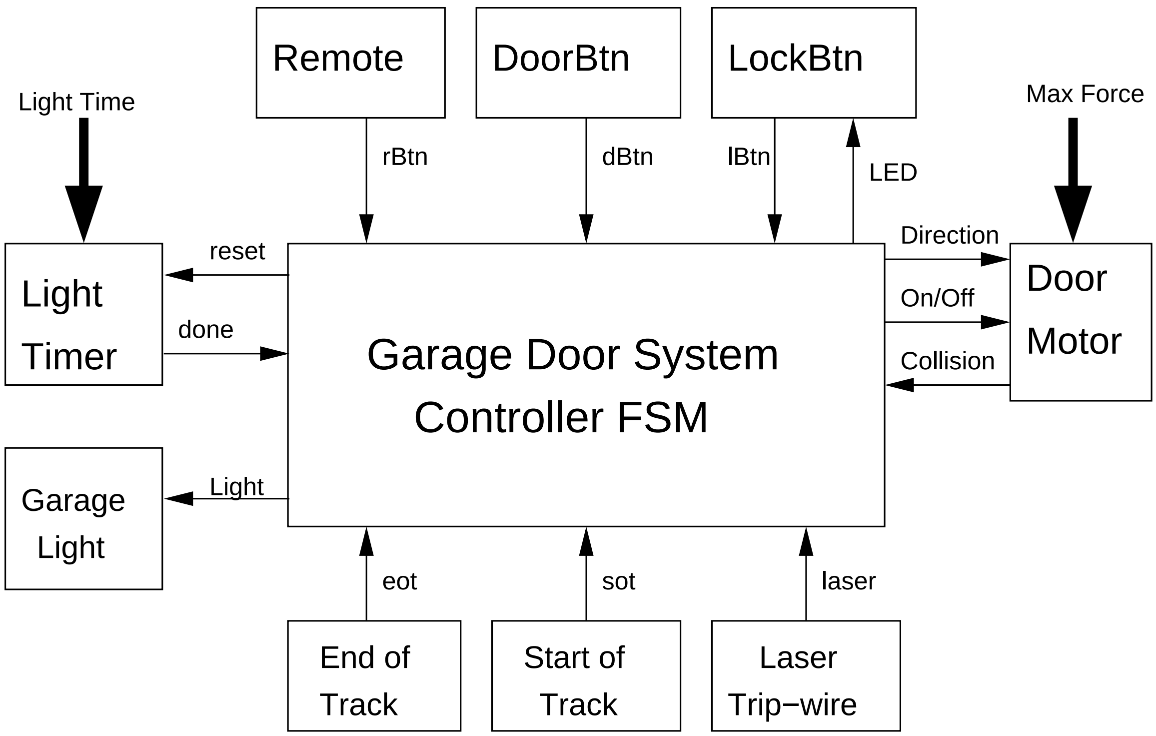
Present State	**Next State** $X_1X_2 = 00$	01	10	11	**Present Output**			
A	A	C	B	D	00	00	00	00
B	B	B	D	D	00	00	10	10
C	C	A	C	A	01	01	01	01
D	B	B	C	A	01	01	10	10

b. (15 points) Using this best state assignment, write the equations for the flip-flops and outputs to implement this machine using D flip-flops.

Problem 5 (15 points) [Conceived by Paul Berube]

In this problem you will design the control logic to be used in a garage door opener system. This controller activates a motor to raise or lower the door, and to turn the garage light on.

Here is a block diagram of the garage door system:



The garage door system has several components. The first is the remote door opener. The remote has only a single button. When the button is pressed, the door will open if it is closed, or close if it is open. If the button is pressed while the door is in the process of opening or closing, then it should stop where it is, and reverse directions. When the button is pressed again, the door will resume moving, but in the new direction. There is a wireless receiver in the garage door opener unit that detects the signal from the remote and generates a single-cycle pulse to the controller unit when the remote button is pressed. *You do not need to design the wireless receiver.*

Next there is the wall button panel mounted inside the garage. This panel has two buttons. The first, the door button, is identical to the button on the remote. Use of the remote button and the door button can be intermingled at will. However, the two signals enter the control unit as separate lines. Also on the wall panel there is a lock button. This lock button toggles the lock state of the control unit. If the lock is on, then signals from the remote should be ignored, but signals from the door button should still work. If the lock is off, the signals from both inputs should be acted on. Inside the lock button is a small LED that should be lit when the lock is on.

When a signal is received from any of the buttons, the garage light should go on if it is not on already. The light should stay on for a fixed period of time. This time delay is generated a timer unit. Whenever the timer unit receives a reset signal, it will immediately de-assert the done signal, and will restart counting down from the hard-wired light time. When the count reaches zero, the counter asserts the done pulse signal.

The movement of the garage door is controlled by a motor. This motor receives two signals: ON/OFF and UP/DOWN. The motor will attempt to move the door in the indicated direction as long as it is receiving the ON signal. There are sensors at each end of the garage door track that will assert a signal (one signal for each end of the track) to the controller if the door reaches the end of the track.

For safety, there is a laser trip-wire 4' off the ground across the door opening. If the laser beam is broken, a signal is sent to the controller. If the door is closing, it should immediately reverse directions, to avoid squashing whatever is blocking the beam. If the door is opening, an interruption of the laser beam should not affect the door's operation.

The motor has a force sensor that measures how much force is required to lower the door. If the force required exceeds a preset level, then the motor raises a signal to the controller to indicate that the door has struck an obstacle. If that happens, the door should immediately reverse directions and start opening. If the excess force signal occurs when the door is opening, the motor should simply be turned off.

a. (10 points) Write a VHDL code that specifies the behavior of the state machine that you designed.

b. (5 points) Using this best state assignment, write the equations for the flip-flops and outputs to implement this machine using D flip-flops.

Problem 6 (0 points) (You should not turn in solutions for this, but we will consider that you have done this problem when preparing your final exam).

- Read and understand section 7.5, 7.7, and 7.8 of the Wakerly textbook.
- Solve problems 7.9, 7.14, 7.15, 7.16, 7.28, 7.39, 7.57, and 7.70 of the Wakerly textbook.