Department of Computing
Course 112 Hardware
First Year Laboratory Assignment
Spring 2002

This is the description of a hardware design exercise which you will be expected to work on for two weeks from the 14th to the 28th January on any of the Windows2000 machines. The design software should be running on all the Windows2000 machines at all times. You will be working with a design tool called XILINX

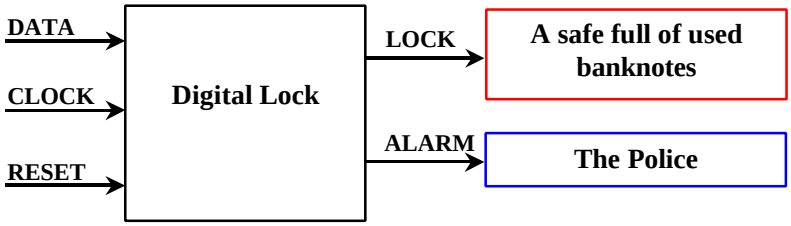
Important instructions on how to use XILINX will be given in the Laboratory Lecture at 14.00 in Room 311 on Monday 14th January 2002.

You will have priority during the laboratory sessions on Tuesday 16.00 to 18.00, Thursday 10.00 to 12.00 and Friday 10.00 to 12.00. Some tutorial help will be available at these times. Please try to make a start early in the first week. If you leave it till the last day you may find that the tutors are too busy to give you any help!

Electronic Submission Deadline: Monday 28th January at 6.00pm
Written project Submission Deadline (in the SGO): Friday 1st February at 5.00pm

1. The Design Problem

Design a three-input, two-output sequential digital circuit which functions as a digital locking mechanism.



If a logic level 0 appears on the LOCK output then the safe is unlocked. If a logic level 1 appears on the ALARM line then the police are summoned. When a logic 1 appears on the RESET line, the mechanism is put into the known "reset" state in which the LOCK output is set to logic level 1 (locked). In this state the lock mechanism is waiting for a specific five binary digit sequence of 1s and 0s to appear, one binary digit for each CLOCK pulse. If the correct sequence of digits is sent then the LOCK output becomes logic 0 and the safe opens. The safe remains open as long as the DATA input line is at logic 1. When logic 0 appears on the DATA input the system returns to the reset state.

If a wrong sequence of digits is input then the ALARM output signal becomes a logic 1 and it remains so for any sequence of values appearing at the DATA input. The only way to remove the Alarm signal is to set the RESET input line to 1.

Your design should use a personalised sequence number. On the web there is a list of students with a unique number allocated to each. This same number was used for the coursework in term 1. Your individual binary sequence number can be generated using the formula:

Binary Sequence = Binary Equivalent of (1 + MyNumber mod 30)

For example, if my sequence number is 87, I get (1 + 87 mod 30) = 1 + 27 = 28 = 11100

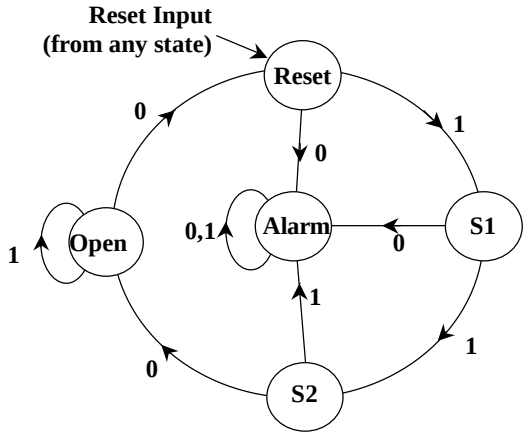
If my sequence number is 8, I get (1 + 8 mod 30) = 1 + 8 = 9 = 01001. The sequence is to be input with the most significant bit comming first.

2. Digital Design

This is an example using three digits, but remember that your design problem is for five digits! The design procedure will be demonstrated for a three-digit sequence, say 110.

2.1 The State Diagram

The operation of the digital lock mechanism can be described by the following Moore state diagram.



Note that a 1 on the RESET input takes the finite state machine to the reset state from any state. We will handle this input separately from the DATA input. We first solve the problem of the sequencing and then consider how to achieve the reset function.

2.2 The State Transition Table

From the State Diagram a State Transition Table can be generated. There are five states; therefore, we need a minimum of three flip-flops whose outputs are Q2, Q1, Q0. We have a free choice to assign the five required states to the eight possible ones. The complexity of the final circuit will depend on these assignments but not in an obvious way. We choose the following ones:

	This State				Next State			
DATA	State	Q2	Q1	Q0	State	D2	D1	D0
0	Reset	0	0	0	Alarm	1	1	1
0	S1	0	0	1	Alarm	1	1	1
0	*	0	1	0	*	*	*	*
0	S2	0	1	1	Open	1	1	0
0	*	1	0	0	*	*	*	*
0	*	1	0	1	*	*	*	*
0	Open	1	1	0	Reset	0	0	0
0	Alarm	1	1	1	Alarm	1	1	1
1	Reset	0	0	0	S1	0	0	1
1	S1	0	0	1	S2	0	1	1
1	*	0	1	0	*	*	*	*
1	S2	0	1	1	Alarm	1	1	1
1	*	1	0	0	*	*	*	*
1	*	1	0	1	*	*	*	*
1	Open	1	1	0	Open	1	1	0
1	Alarm	1	1	1	Alarm	1	1	1

2.3 The Karnaugh Maps

From the flip-flop output table the three design Karnaugh maps can be generated:

D2

DataQ2 \ Q1 Q0	00	01	11	10
00	1	1	1	X
01	X	X	1	0
11	X	X	1	1
10	0	0	1	X

D1

DataQ2 \ Q1 Q0	00	01	11	10
00	1	1	1	X
01	X	X	1	0
11	X	X	1	1
10	0	1	1	X

D0

DataQ2 \ Q1 Q0	00	01	11	10
00	1	1	0	X
01	X	X	1	0
11	X	X	1	0
10	1	1	1	X

D2 = DATA'•Q1'+DATA•Q2 +Q1•Q0
D1 = Q0 + DATA'•Q1'+DATA•Q2
D0 = Q1' + Q2•Q0 + DATA•Q0

Further factorisations and simplifications may be possible at this stage.

2.4. Output Logic Design

You must ensure that the LOCK output signal becomes 0 only when the system is in the Open state, and the safe is to be unlocked. Similarly, the ALARM output should only be 1 when the system is in the Alarm state.

3. Circuit Design

Design now is similar to all the previous exercises, except that there is a larger number of available gates. You are allowed to use the following gate types:

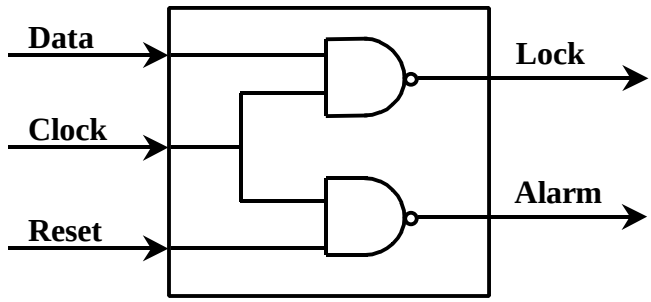
Name	Function	Area	Name	Function	Area
INV	Inverter	15	NOR3		43
AND2	Two input AND	27	OR2		28
AND3	Three input AND	34	OR3		34
AND4		40	XNOR2		44
NAND2	etc.	20	XOR2		44
NAND3		28	FDC	D Flip Flop with reset	110
NAND4		32	FDP	D Flip Flop with preset	100
NOR2		19			

The *RESET* input to the lock mechanism can be achieved by using the RESET or PRESET input for the D-type flip-flops FDC or FDP. Depending on the state (flip-flop outputs) assigned to the RESET state, you may choose to use flip-flops with RESET, or PRESET function only. The ordinary clock action occurs at the leading edge; i.e at the positive rising edge of the clock pulse.

Minimisation here is for silicon area; therefore you are given the approximate area for each gate in the table above. Try to minimise this area as much as possible within your design constraints.

4. Schematic Construction and Testing

We suggest that you implement the simple circuit shown below at first. This is just for practice. It doesn't solve the problem. Go through the schematic construction, simulation, hard copy generation, and the electronic submission for this sample circuit, so you get the mechanics out of the way. Decide whether the results make sense.



Once you have built the circuit schematic, you can test your circuit by the appropriate test vector file. There are eight test vector files named vect00_03.CMD, vect04_07.CMD, ..., vect28_31.CMD which contain testing for four digital DATA sequences; the first includes data sequences for numbers 0, 1, 2, and 3, the next for 4,5,6, and 7, etc.; the last for 28, 29, 30 and 31.

Once you are satisfied that your circuit does work, you should make a hard copy of your circuit as well as your simulation results (waveforms). You have to include these two hard copy sheets with your report. You should also submit your design with the proper submit command.

5. Required Submission

You are required to electronically submit results of your simulation. Also, you are required to submit a written report (of not more than 5 sides plus the two hard copy pages) in which you describe your solution to this design problem, and include the calculation of the total silicon area.