Department of Computing
Course 112 Hardware
First Year Laboratory Assignment

Dates for the session 2003-2004:

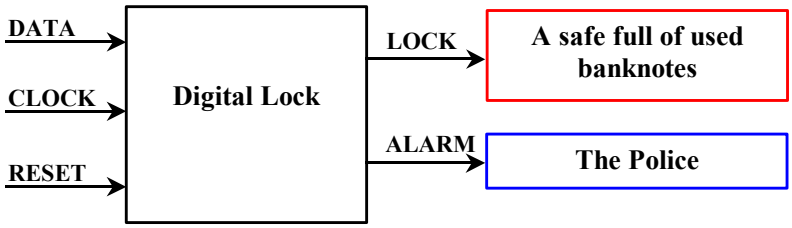
Hand out Date:	13th January 2004
Hand in deadline (electronic and written report):	17.00 Monday 26th January 2004
Laboratory Lecture	13th January 12.00 room 308
Laboratory Sessions	Tuesday 16.00 - 18.00
	Thursday 09.00 - 11.00
	Friday 14.00 - 16.00

There will be a limited amount of tutorial help during the laboratory sessions so try to start the exercise as soon as possible. On the last friday the tutors may be too busy to see everybody. If you are stuck at other times you can mail David Adams for help (dja@doc.ic.ac.uk)

The hardware laboratory is designed to introduce you to a modern digital design tool called Xilinx. The Xilinx software is available on all the Windows2000 machines in the laboratory.

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) = 88 mod 30 = 28 = 11100

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

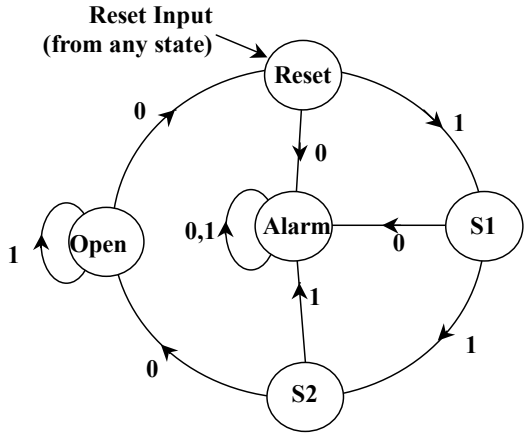
2. Digital Design

The design procedure will be demonstrated for a three-digit sequence, say 110 (remember that your design problem has five digits!).

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 by using either a preset or clear function on the D-Q flip flops.



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 which are a small subset of those supported by Xilinx.

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

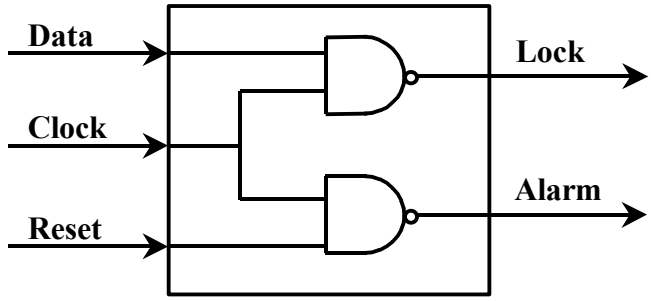
The *RESET* input to the lock mechanism can be achieved by using the CLEAR or PRESET input for the D-type flip-flops FDC or FDP. Depending on the state (flip-flop outputs) assigned to the

RESET state, you may choose to use flip-flops with CLEAR, 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 with using Xilinx. It doesn't solve the problem, but you can easily tell whether you have a correct implementation. Go through the schematic construction, simulation and hard copy generation for this sample circuit, so you get the mechanics out of the way. Decide whether the results make sense.



You can test the above circuit and your final design by using an appropriate test vector file. A test vector file is one that can be read by Xilinx, and contains a specification of the inputs over a period of time. There are eight test vector files named vect00_03.CMD, vect04_07.CMD, ..., vect28_31.CMD each containing four digital sequences starting with a reset; 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. You should check that your circuit places a 0 on the lock output only on the correct sequence only.

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).

5. Getting Started With Xilinx

In the following description, "Click" means click the left mouse button, and "Click right" means click the right mouse button. Use can any Windows 2000 machine to run Xilinx.

Setting up your Machine

1. Ensure that your fileserver space is mounted as the H: drive. Although the work is done under the Windows 2000 OS you will need to use Linux for the final submission.
2. Create a directory in your home space for your designs (eg use H:\xilinx\designs2).
3. The simulation test vector files are held on the server in a folder called Xilinx\test_files. To access them from Windows, directory \\fs-ug\Lab should be mounted as a drive. Using the windows explorer:
   Click on - map network drive - in the Tools menu.
   In the Drive box enter X: (or whatever letter you want to use)
   In the path box, enter \\fs-ug\Lab
   Click OK

The test vector files set up for this exercise are named vec00_03.cmp to vec28_31.cmp. Copy them all to your directory by dragging and dropping.

Starting the Tools

From the Windows Start menu select Programs, Lab, Xilinx Foundation 4 and Click on Project Manager. The project manager window will appear. If this is the first time the tools have been started a dialogue box will appear offering to create a new project. Click OK. Set the directory path to the one you created for your design (H:\xilinx\designs2) and name your design d_lock Ensure the library is set to XC4000E (bottom left dialog) and schematic button is set, leave the rest as default. Click OK.

If you experience problems starting Xilinx try another PC and email David Adams (dja@doc.ic.ac.uk) with the name of the suspect PC.

Entering your design.

1. The right hand window shows various icons. Click on the schematic editor icon (shown as a gate symbol in design entry box). This will open the schematic editor window. You will see a column of icons on the left. Pass the cursor over the icons and pause for a moment to see a description of each one.
2. Click on the one marked with an AND symbol to obtain the symbol toolbox. This lists the components available in the selected libraries. For this exercise you are limited to the 15 types of component listed above the handout.
3. Select each component required and move the cursor to a suitable spot on the sheet, click to place it. Select the arrow tool from the left hand menu to move it about. Place other components as required.
4. Input (IPAD) and output (OPAD) terminals should be placed for the 5 external connections to your circuit.
5. The interconnections can be made using the "Draw Wires" tool (the button below the symbol toolbox on the left hand side). Alternatively wires (and terminals) with like names are assumed to be connected together. Using this feature can simplify the appearance of your diagram.

To name a wire

1. Using the standard crosshair (selected with the arrow button on top of the left hand button bar) point to the wire and click right. A menu will appear, select net properties then enter/edit the name.
2. For Component pins remember to add a small stub of wire if there is no other wire connected. First connect a wire to the pin add a small stub and click right, then click the "Add label" button type in the name and enter.
3. Terminal pins will automatically be connected to any wires with the same name.

Note that you must use the names DATA, RESET and CLOCK for the input wires and LOCK and ALARM for the output wires in order to make the simulation work. For all other wires adopt a sensible naming convention to help you when debugging your circuit. For example if as part of your state sequencing logic you have a wire corresponding to a tern Q1·Q2' choose a name line NQ1NQ2

To Edit Components, wires or terminals

First select the component then click right, a menu will appear. Click properties.

Once you have entered your design you can use Xilinx to simulate it.

1. Click on the SIM button, and the Logic simulator will appear. Click on the File menu and select the Run Script File command. A file selector will appear. Look in the H:\xilinx\ folder and select the file which contains your lock combination. Click on the OK button and your design will then be simulated using the file selected.
2. Wait for the simulation to complete. The waveforms obtained should now be displayed. If the wave forms appear as one bus, click on the button called "buses" (5th button from left above the waveform display) to display the individual signals.

Debugging your design

To display individual wires in the simulated waveforms click the "select component" button (5th button from left above the waveform display) to display a list of the named wires. Double click to add a wire to the waveform list. When the circuit is re-simulated the waveforms will be displayed. By clicking on the waveforms displayed a cursor can be set to any time in the simulation. The corresponding states will be displayed in the schematic editor window.

Printing

In the schematic editor the default the sheet size is very large. To change this, Click Page setup in the File menu and select a more suitable size. For the final printout you can scale the design to fit the printer page.

6. Required Submission

Paper report

You are required to submit a written report in which you describe your solution to this design problem, showing your finite state machine, state transition table, Karnaugh maps, final circuit printout, waveform printout and your calculation of the total silicon area.

Electronic submission of your design

Submission must be done in the Linux environment. First change to your design directory \xilinx\ and type:-

```
tar zcf   Design.tar.gz  designs2
submit hw2
```

Confirmation will be sent to you by email.