UNIVERSITY OF NAIROBI

PROJECT TITLE: ELEVATOR CONTROL CIRCUIT

PROJECT INDEX: PRJ 045

BY:

NAME: MASILA JANE MWELU

REG. NUMBER: F17/10455/2003

SUPERVISOR: PROF. MWANGI

EXAMINER: DR. MANG'OLI

Project report submitted in partial fulfillment of the requirement for the award of the degree of
Bachelor of Science in Electrical and Electronic Engineering of the University of Nairobi

20th May, 2009

Department of Electrical and Information Engineering

Dedication

To my loving family and friends

Acknowledgments

To the Lord Jesus; The source of all grace that empowers us to be all that He created us to be; thank you for imparting the wisdom and knowledge necessary to get this project done. To you belongs all glory and honor.

To my supervisor, Prof. Mwangi; Thank you for providing unwavering support when I was on this project. Your consistent and constant guidance was a tremendous source of motivation.

To all lecturers, in the Electrical and information Department(U.O.N) for impacting us with seamless knowledge.

To my colleagues; for helping me whenever in doubt.

Abstract

In the design procedure of the elevator control circuit, the controller- Datapath approach was used. In this approach, all the functional and memory are concentrated in the Datapath while the control signals are generated by a simpler sequential machine(controller). To specify the control sequence and data processing tasks of the designed digital system, a hardware algorithm has been adopted and the corresponding Algorithmic State Machine chart has been used.

The control and display logic has been designed with Small Scale Integration(SSI) and Medium Scale Integration(MMI) logic components.

The interface to the power circuit has been identified and its behavior has been simulated with Circuit Maker Pro software tool.

Table of contents

Chapter 4

Chapter 5

Chapter 1
Introduction

For the last one hundred years, the availability of space for construction of the houses and offices within towns and cities has continued to shrink. This has given the way to the construction of multi-storey buildings based in the use of steel and iron as support structures. However this buildings would not have been of use if an access technology such as that provided by elevators was not available. In addition, the elevator must provide a special link between the demand by the users and available access. In this chapter an overview of elevator technology is presented and issues pertaining to safety are also discussed.

The objective of this project is to design a controller for an elevator that serves three floors. The problem statement is formulated as an algorithmic state machine that can be implemented with small scale integration(SSI) and medium scale integration(MSI) logic modules as a sequential logic circuit. A brief discussion of the sequential logic is thus presented.

1.1 Elevator technology

Elevators are especially important in tall structures like skyscrapers, where climbing stairs to get to top floors would be very difficult. The car of an elevator, in which people ride, is attached to guard rails inside a tall, empty space called a shaft.

1.1.2 Elevator designs

There are two types of elevators in common use today.

- Hydraulic type elevators
- Roped (cable type) elevators

1.1.2.1 Hydraulic elevators systems

They are designed to lift a car using a hydraulic ram using a fluid-driven piston mounted inside a cylinder. The cylinder is connected to a fluid-pumping system.

The hydraulic system has three key parts;

- A tank(which is the fluid reservoir)
- A pump(powered by an electric motor)
- A valve

Hydraulic elevators operation:

The pump forces fluid from the tank into a pipe leading to the cylinder. When the valve is opened, the pressurized fluid will take the path of least resistance and return to the fluid reservoir. But when the valve is closed, the pressurized fluid has nowhere to go except into the cylinder. As the fluid collects in the cylinder, it pushes the piston up, lifting the elevator car. When the car approaches the right floor, the control system sends a signal to the electric motor to gradually shut off the pump. With the pump off, there is no more fluid flowing into the cylinder, but the fluid that is already in the cylinder cannot escape (it can't flow backward through the pump, and the valve is still closed). The piston rests on the fluid, and the car stays where it is.

To lower the car, the elevator control system sends a signal to the valve. The valve is operated electrically by a basic solenoid switch. When the solenoid opens the valve, the fluid that has collected in the cylinder can flow out into the fluid reservoir. The weight of the car and the cargo pushes down on the piston, which drives the fluid into the reservoir. The car gradually descends. To stop the car at a lower floor, the control system closes the valve again.

1.1.2.2 The cable system

It is the most popular elevator design, in this system the car is raised and lowed by traction steel ropes. The ropes are connected to the elevator car, and looped around a pulley. The pulley grips the ropes, so that when the electric motor rotates the pulley, the ropes move too.

When the motor turns one way, the pulley raises the elevator and when the motor turns the other way, the pulley lowers the elevator.

There are two types of cable elevators:

- Geared
- Gearless

Geared versus gearless elevators

- In geared elevators, the motor rotates the sheaves directly.
- In gearless elevators, the motor turns a gear train that rotates the sheave.

Roped elevators are much more efficient than hydraulic elevators, they also have more safety systems.

1.1.3 Elevator safety mechanism

The first elevators in use were not especially safe because once in a while a cable would break, and a car, pulled by gravity, would come crashing down. Safety devices were soon added, though, to keep such disasters from occurring. Additional ropes attached to cars and powerful metal "jaws" that grip guard rails keep elevators from falling if their main cables break. Other safety devices keep elevators from moving when their doors are still open and from traveling too fast. Automatic switches in the shaft allow an elevator to hurry past unwanted floors, or to slow and stop when a chosen floor is reached, unlocking its doors to admit and release passengers.

With out safety mechanisms no one would dare use elevator systems. Therefore safety is a mandatory issue that needs special attention. Elevators are built with several safety systems that keep them in position.

1.1.3.1 The rope system:

This is the line of defense. One rope can support the weight of the elevator car. In essence, elevators are built with multiple ropes, if one of the ropes snaps, the rest will hold the elevator up.

Roped elevator cars have in built braking systems, that grab onto the guide rails when the car moves too fast.

1.1.3.2 Speed Governor:

When the elevator moves too quickly, braking is activated by a governor. A governor rope is connected to the elevator car, so it also moves when the car moves up and down. As the car speeds up, so does the governor, therefore controlling the speed of the elevator car.

1.1.3.3 Electromagnetic Brakes

This brakes engage when the elevator comes to a stop. When the elevator moves too fast its brought to a stop by the automatic braking.

The elevator must provide a link between the controller and the mechanical parts, the controller can be implemented using sequential logic circuits. A brief discussion of the sequential logic circuits is therefore presented.

1.2 Sequential logic circuits

In a sequential logic circuit, the outputs depend not just on the current values of the inputs, but also on the past values of the inputs. The circuit has memory. Sequential circuits can do two things that combinational circuits can not:

- Recognize sequences of inputs
- Generate sequence of outputs.

A sequential circuit consists of a combinational circuit to which the storage elements are connected to form a feedback path. The storage elements are devices capable of storing binary information. The binary information stored in these elements at any given time defines the state of the sequential circuit at that time. The sequential circuit receive binary from the information from the external inputs. These inputs together with the present state of the storage element, determine the binary value of the outputs.

A sequential circuit can be made by adding feedback to a combinational circuit as shown in figure 1.1 .In this circuit, depending on the design of the combinational logic, the value of the output Y, will depend both on the two inputs A and B and the internal feedback signal. With one feedback signal the circuit can have two behaviours for any particular values of the inputs from the outside: one with the feedback signal = 0, and one with the feedback signal = 1. the circuit has two states, with different behaviours. A circuit with two feedback signals can have up to four different states, and one with n feedback signals, up to 2^n states.

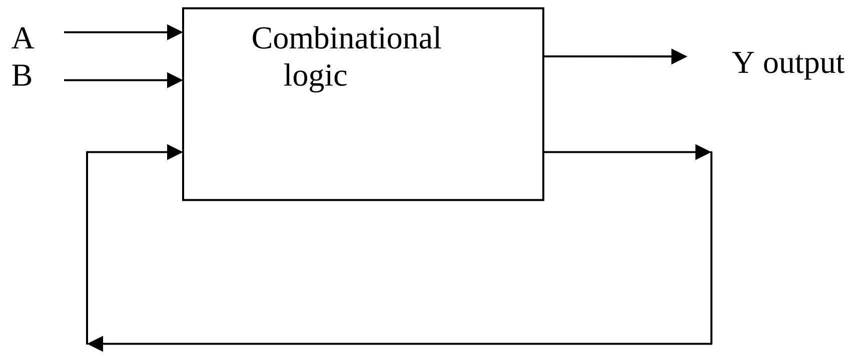

Figure 1.1. A Sequential logic circuit

1.2.1 Synchronous sequential logic

In an asynchronous circuit, the state of the circuit , i.e. the value of the feed back signal(s), can change at any time in response to an input change. Asynchronous circuits are tricky to design, and so most sequential logic circuits are designed in a slightly different way, as synchronous circuits. In a synchronous sequential circuit, changes of state are only allowed to happen at times synchronous to a special timing signal, called the clock.
The simplest synchronous circuit is a one-bit storage element, which is also referred to as a flip-flop. There are four popular types of flip-flops i.e. JK, D, T and SR flip-flops.

Flip-flops record states while clock controls the transitions from one state to the next. These are called finite state machines because they can have at most a finite number of states.

1.2.2 Types of State Machines

There are two types of finite state machines that can be built from sequential logic circuits:

- Moore machine
- Mealy machine

In the Moore state machine shown in figure 1.3, the outputs depend only on the internal state and any inputs that are synchronized with the circuit.

In the Mealy state machine, the outputs are determined by both the internal state and by inputs that are not synchronized with the circuit, as indicated in figure 1.4

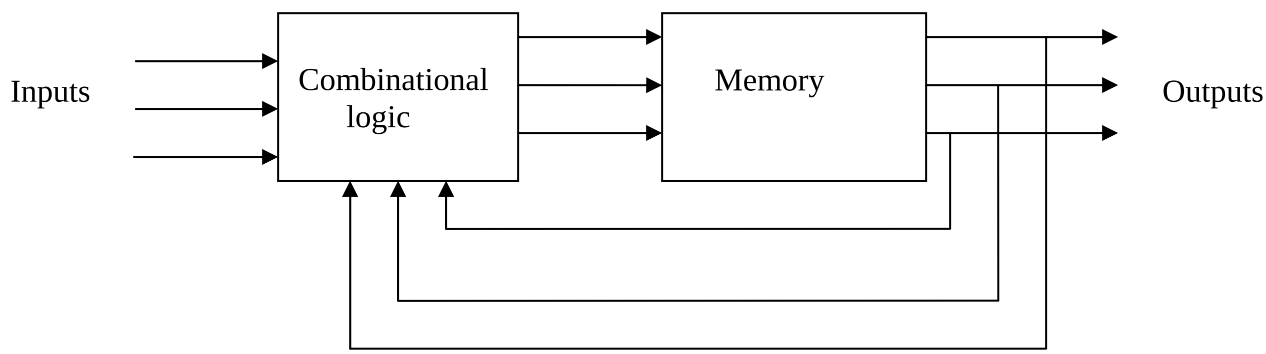

Figure 1.3 Moore state machine

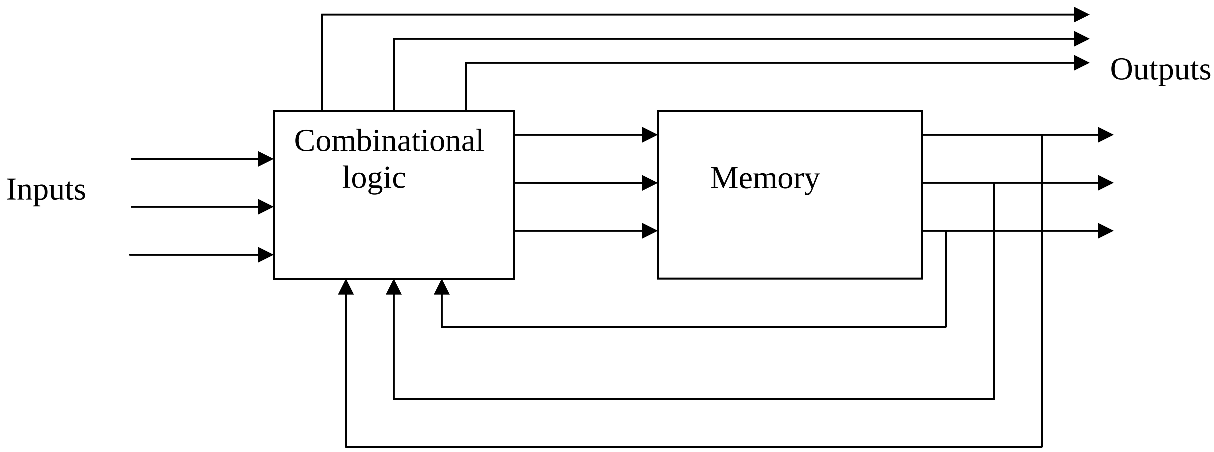

Figure 1.4 Mealy state machine

Chapter 2

Algorithmic State Machine(ASM)

The logic design of a digital system can be divided into two distinct parts. One part is concerned with the design of the digital circuits that perform the data processing operations. The other part is concerned with the design of control circuits that determines the sequence in which the various actions are performed.

The relationship between the control logic and the data processing in a digital system is as shown in fig. 2.1.

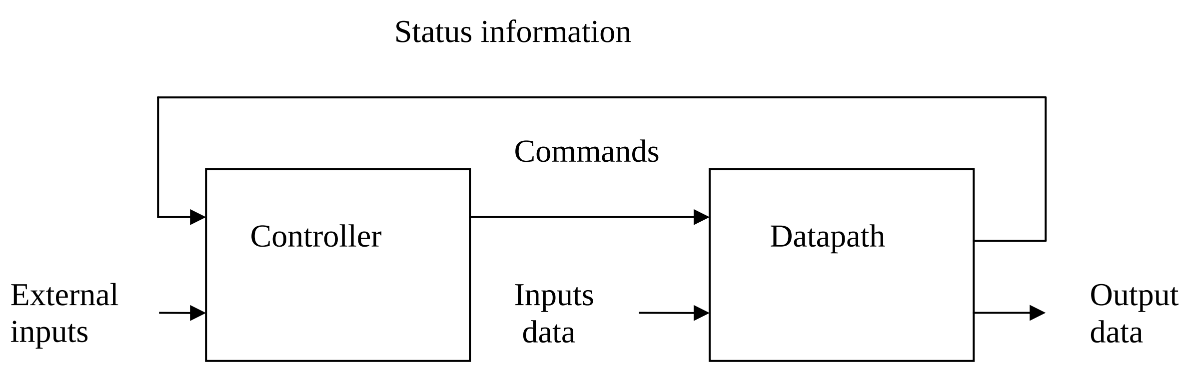

Figure 2.1 Partitioning of a digital system

The Algorithmic State Machines are similar to finite state machines, yet they are more expressive in that, during each state (i.e. clock pulse), register operations may be specified and performed during the transition.(i.e next positive clock transition). Fig. 2.2 shows a model of an Algorithmic State Machine.

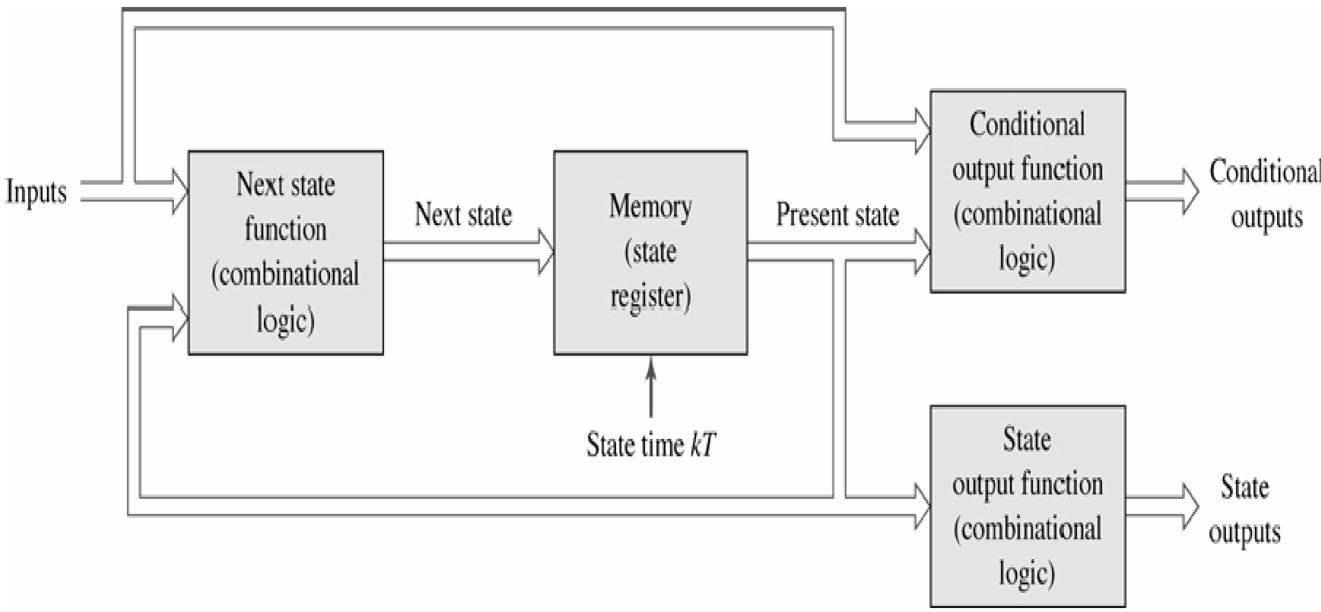

Figure 2.2 Model of an Algorithmic State Machine

Algorithmic State Machine Chart

The ASM chart is a convenient tool that translates the system specifications to an information diagram in which the sequence of operations and the conditions necessary for execution are indicated. It is a special type of flow chart describing the sequential operations of a digital system. The ASM chart consists of a set of one or more blocks. Each block may be viewed as a directed simple graph that has three kinds of nodes.

- **State box**. Represents a state and it is equivalent to a node in a state diagram or a row in a state table. A block has one state box which is symbolically represented as a rectangular box. The state box represents which state the system is in when block is being executed. Within the box is a listing one or more register operations that take place during the next rising clock edge when control passes from that block to the next block/state. The edge leaving the state box may

a) Loop back to itself
b) Be directed towards the state box of a different block, or
c) Be directed to a decision box within the block.

The state is given a symbolic name which is placed at the upper left corner of the box. The binary code assigned to the state is placed at the upper right corner as shown in figure 2.3.

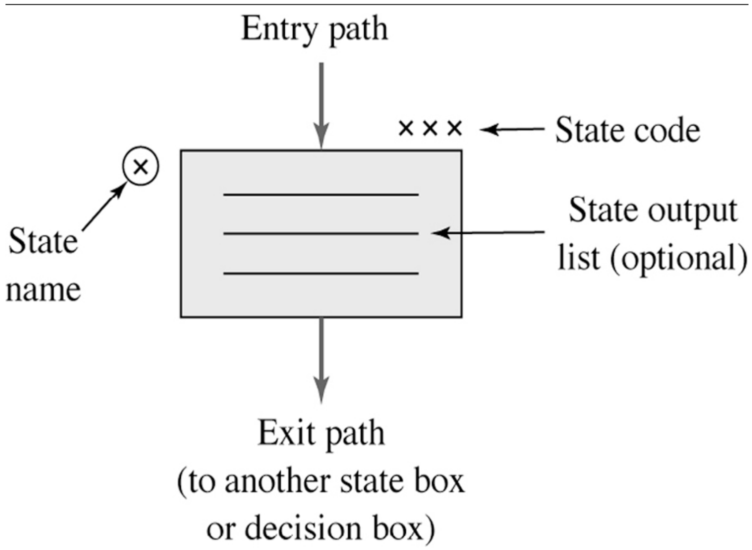

Figure 2.3 State box

- **Decision box**. A decision box is symbolically shaped like a diamond with two or more exit paths, as shown in figure 2.4(a), alternate symbol of a decision box is as illustrated in figure 2.4(b). It describes the effect of an input on the control subsystem. The input condition to be tested is written inside the box. Edges leaving the decision box may

a) Be directed towards the state box of any block,
b) Another decision box within the block, or
c) A conditional block.

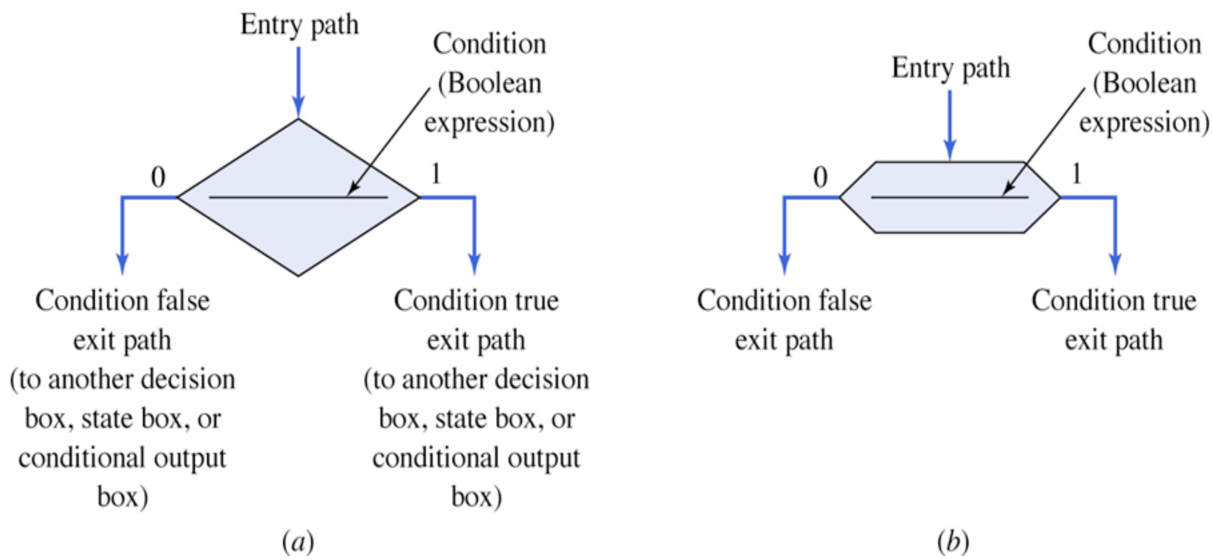

Figure 2.4. Decision boxes

- **Conditional box**. A conditional box is symbolically shaped like a rectangular box with rounded sides as shown in figure 2.5. Within the box are one or more register operations which take place during the next rising clock edge, on condition that execution passes through the box . Edges leaving the conditional box may be

a) Be directed towards the state box of any other block or

b) Another decision box within that block.

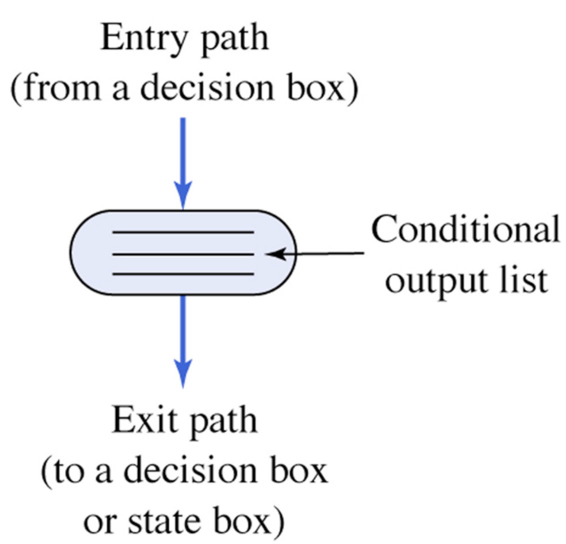

Figure 2.5 Conditional box

The data processing path(Datapath) manipulates data in the registers according to the system requirements. The control logic initiates a sequence of commands to the Datapath. The control logic uses status condition from the Datapath to serve a decision variables for determining the sequence of the control signals.

At any given time, the state of the sequential control initiates a prescribed set of commands. Depending on the status conditions and other external inputs, the sequential control goes to the next state to initiate other operations. The digital circuit that act as the control logic provide a time sequence of signals for initiating the operations in the Datapath and also determine the next state of the subsection. The ASM chart describes the sequence of events as well as the timing relationship between the states of a sequential controller and the events that occur while going from one state to the next. The timing of the Algorithmic State Machine is illustrated in figure 2.6.

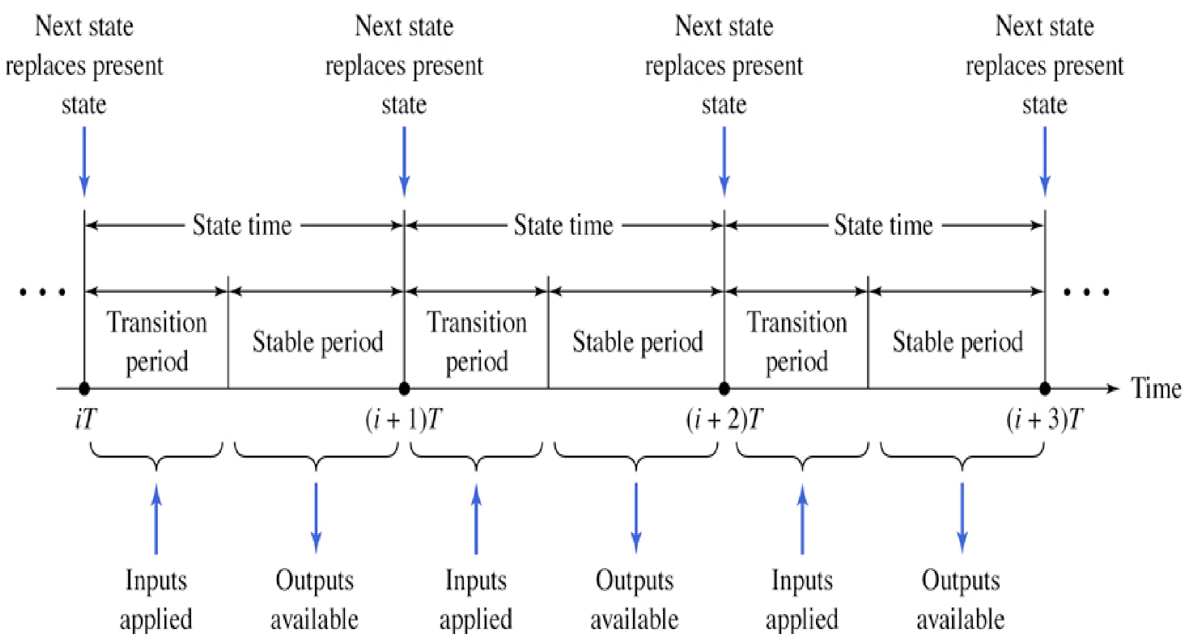

Figure 2.6. Timing of an ASM

The operations specified inside the state boxes and the conditional boxes of each block are performed in the Datapath subsystem while the change from one state to the next is realized in the controller.

The design for the Datapath requires both the interpretation of the operations and their implementation adopting digital components such as registers, counters, adders and so on. State diagrams can become very messy. In many cases just drawing a state diagram includes assumptions that are not true in general. Perhaps certain cases of the inputs will never happen, hence the corresponding arcs are simply not drawn. Certain cases of the outputs are not significant and sometimes are left out. An algorithmic State machine(ASM) charts offers several advantages over state diagrams;

- Often easier to interpret
- Conditions for a proper state diagram are automatically satisfied
- May be easily converted to other forms

Chapter 3

Implementation

The implementation of the controller and Datapath for a three floor elevator is presented in this chapter. In the implementation a general description of the Datapath, the controller and the relationship in digital system is indicated. The design procedure is specified by means of a digital hardware algorithm, adopting the algorithmic State Machine(ASM) chart.

The Algorithmic State Machine design method is used due to the presence of large number of input signals.

The controller responds to call switches on each floor and floor select switches within the elevator .When the elevator lands on a given floor of the building, signals are generated from the sensor switches .The controller should also generate control signals to move the elevator up or down and generate signals to open or close the door.

3.1 Design specification

The control system monitors the status of the elevator push- button switches, responds to service calls, and displays the floor number.

There is a request push- button on each floor beside the elevator door .When the elevator arrives at a given floor, the door opens for a pre-determined duration to allow passengers on and off and then closes automatically .Inside the elevator there are several push-button switches. These are X0, X1, X2, OPEN and CLOSE push-buttons switches .Pressing the open switch will cause the door to reopen .If none of the switches is activated, then the elevator remains at the last floor serviced. A slow clock signal controls the time that the door remains open when the elevator stops at a floor. The block diagram of the elevator control system is shown in Figure 3.1.

The control logic consists of a state machine which stores a momentary closure of the push-buttons switches and produces the elevator control outputs.

The display logic is basically a decoder that produces outputs to a 7-segment display to indicate either ground, first or second floor.

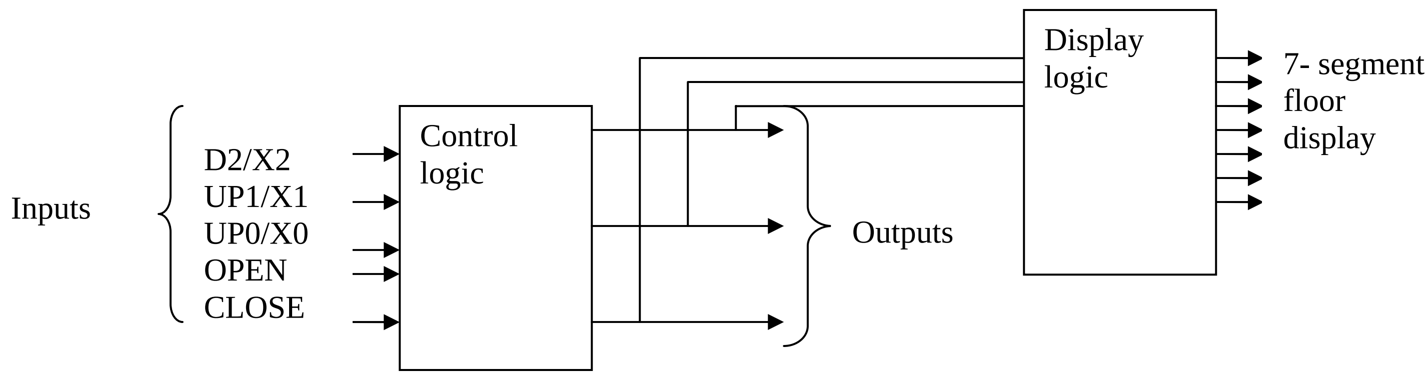

Figure 3.1 Block diagram of the elevator control system.

3.2 Description of the system requirement

If the elevator is on ground floor and the floor requested is ground floor, then the elevator remains waiting on ground floor. When the elevator is on ground floor and the floor requested is first floor, then the elevator is raised up one floor, then if the elevator is on ground floor and the floor requested is second floor, then the elevator is raised up two floors.

When the elevator is on first floor and the floor requested is first floor, then the elevator remains on first floor. If the elevator is on first floor and the floor requested is second floor, then the elevator is raised up one floor, then if the elevator is on first floor and the floor requested is ground floor, then the elevator goes down one floor.

When the elevator is on second floor and the floor requested is second floor, then the elevator remains on second floor. If the elevator is on second floor and the floor requested is first floor, then the elevator goes down one floor. When the elevator is on second floor and the floor requested is ground floor, then the elevator goes down two floors.

3.2.1 Inputs to the state control logic

- X0 Request for the elevator to come to the ground floor. The X0 push-button is located inside the elevator.
- X1 Request for the elevator to come to the first floor. The X1 push-button is located inside the elevator.
- X2 Request for the elevator to come to the second floor. The X2 push-button is located inside the elevator.
- UP0   Request for the elevator to come to the ground floor. The UP0 push-button is located outside the elevator.
- UP1 Request for the elevator to come to the first floor. The UP1 push-button is located outside the elevator.
- UP2 Request for the elevator to come to the second floor. The UP2 push-button is located outside the elevator.
- OPEN Instruction for the door to open when the elevator is not moving. The Open push-button is located inside the elevator.
- CLOSE Instruction for the door to close when the elevator is moving. The Close push-button is located inside the elevator.

3.2.2 Outputs from the state control logic

- Ygf- Ground floor latch output for the push button switches. When one of the push-button switches for the ground floor is pressed, the request is immediately stored in D flip-flop.
- Yff- First floor latch output for the push button switches. When one of the push-button switches for the first floor is pressed, the request is immediately stored in D flip-flop.
- Ysf - Second floor latch output for the push button switches. When one of the push-button switches for the second floor is pressed, the request is immediately stored in D flip-flop.
- Y-tim-Controls the timing of the system.

3.3 The Algorithmic State Machine(ASM) Chart for the controller

The three states of the system are as follows;

- GRD: The elevator is waiting on ground floor with the door open or closed.
- FIR: The elevator is waiting on first floor with the door open or closed.
- SEC: The elevator is waiting on second floor with the door open or closed.

The ASM charts for GRD, FIR and SEC states are shown in figure 3.2, 3.3 and 3.4 respectively. Figure 3.5 shows the interconnected blocks ASM chart describing the sequential operations of the elevator controller. It is composed of;

- State boxes
- Decision boxes
- Conditional boxes

Each block in the ASM chart describes the state of the system during one click pulse interval. The operations in the state and conditional boxes within that block are executed with a common clock pulse while the elevator is in that state. The same clock pulse also transfers the system controller to one of the next states – GRD,FIR or SEC as dictated by the binary values. The control logic is determined from the decision boxes and the required transitions. The decision boxes describes the effect of the inputs on the control subsystem. One exit path is taken if the condition is true and the change from one state to the next is performed in the control logic.

The requirements for the design of the Datapath are specified inside the conditional boxes and the state boxes. The Datapath for the elevator controller consists of a pulser, counters and gates.

When none of the push-button switches is activated, the elevator rests at the last state serviced, waiting for the input signal.

3.3.1 State GRD

If CLOSE = 1, the counter is cleared to 0 and the door is closed.

Then if OPEN = 1, the counter continues to count while the door is open.

When (UP0 or X0) = 1 the system remains in state GRD.

If (UP1 or D1 orX1) = 1 the system goes to state FIR.

If (D2 or X2) = 1 the system goes to state SEC.

3.3.2 State FIR;

If CLOSE = 1, the counter is cleared to 0 door the door is closed.

If OPEN = 1, the counter continues to count while the door is open.

If (X0UP1 or D1 orX1) = 1 the system remains in state FIR.

If (UP0 or X0) = 1 the system goes to state GRD.

If (D2 or X2) = 1 the system goes to state SEC.

3.3.3 State SEC.

If CLOSE = 1, the counter is cleared to 0 and the door is closed.

If OPEN = 1, the counter continues to count while the door is open.

If (UP0 or X0) = 1 the system goes to in state GRD.

If (UP1 or D1 orX1) = 1 the system goes to state FIR.

If (D2 or X2) = 1 the system goes to state SEC.

CLOSE is the complement of OPEN.

3.3.4 Timing Sequence

The timing for all the flip-flops in the controller is controlled by a clock, The clock pulses are applied to all the flip-flops in the control logic.

Each block in the ASM chart describes the state of the system during one clock pulse interval. All the operations that are specified within the block occur in synchronism during the edge transition of the same clock pulse while the system changes from one floor to the next. The operations specified within the state and conditional boxes in the block are performed in the Datapath subsection. The ASM chart for the elevator consists of three blocks. Change from one state to the next is performed in the control logic, this is presented pictorially in figure 2.6

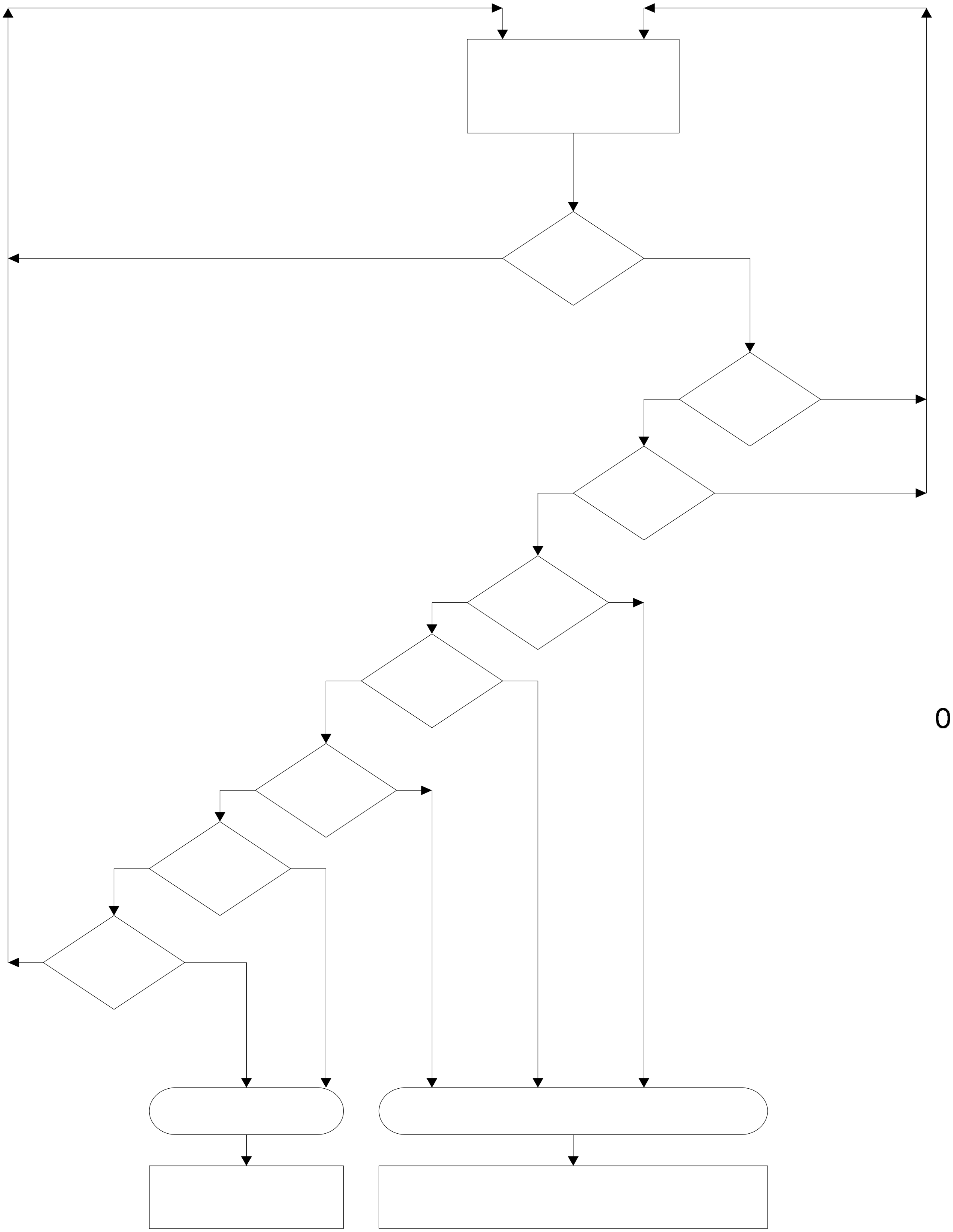

Figure 3.2 State GRD ASM chart

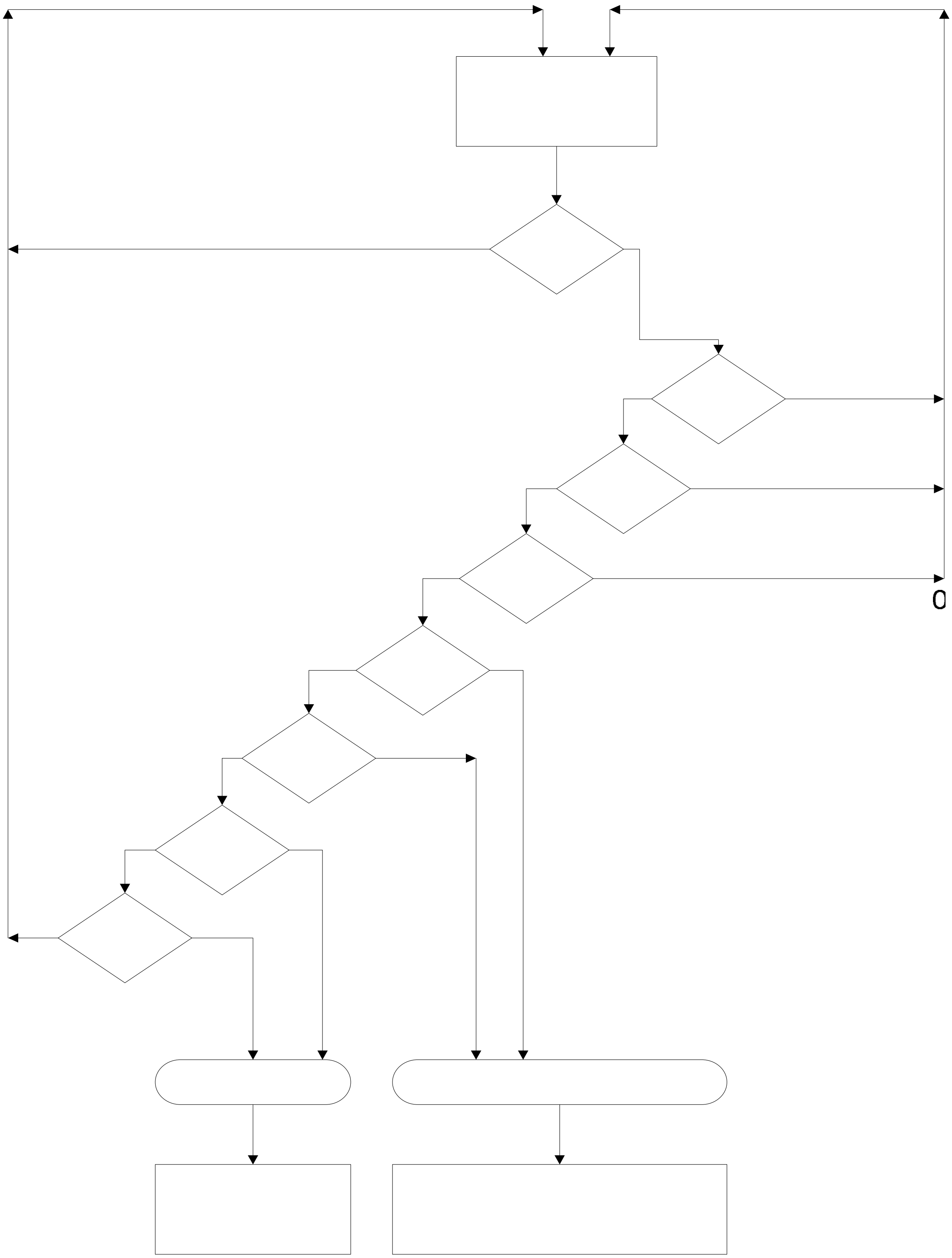

Figure 3.3: State FIR ASM chart

0

0

UP0

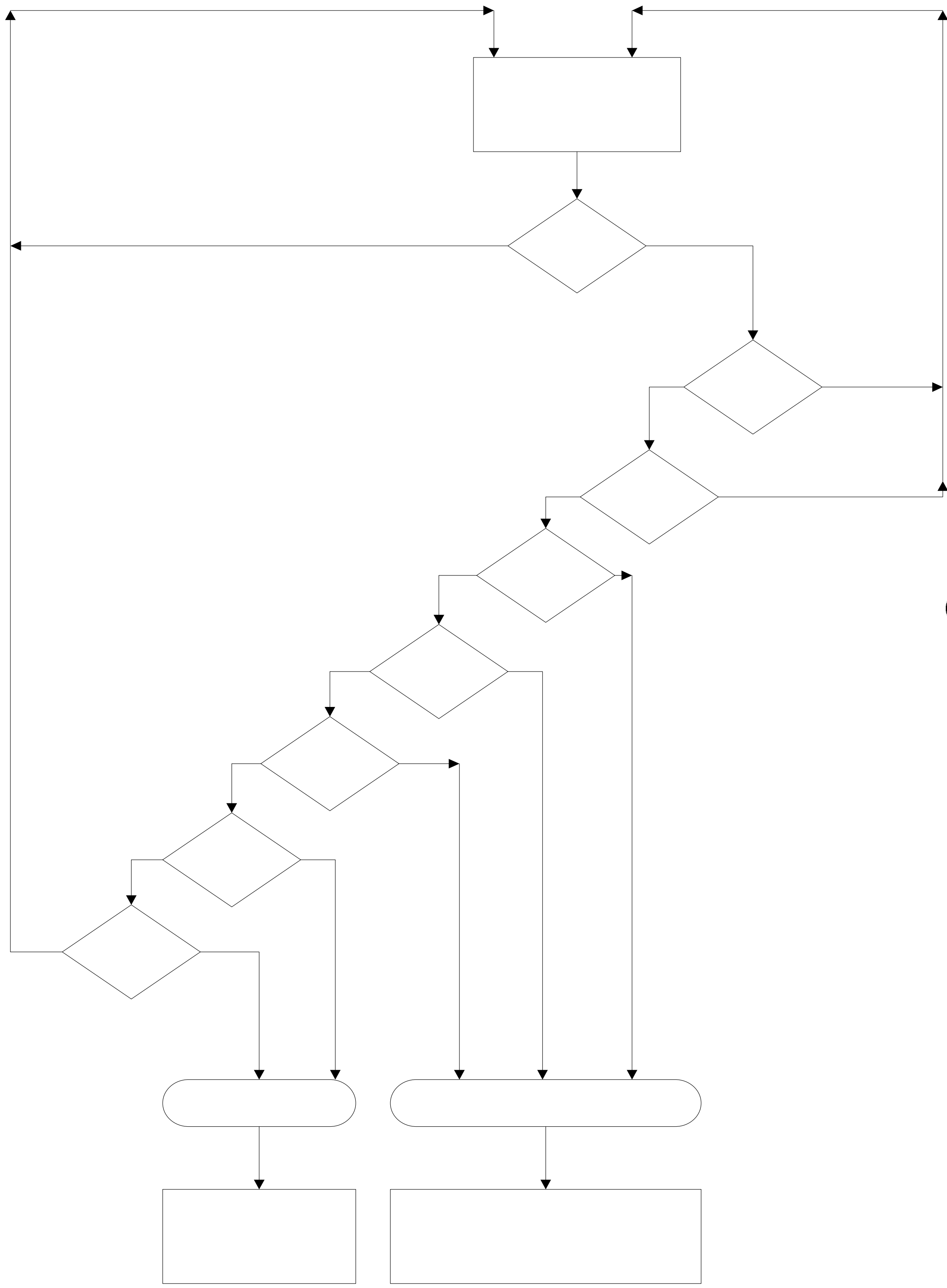

Figure3.4 :   State SEC ASM chart

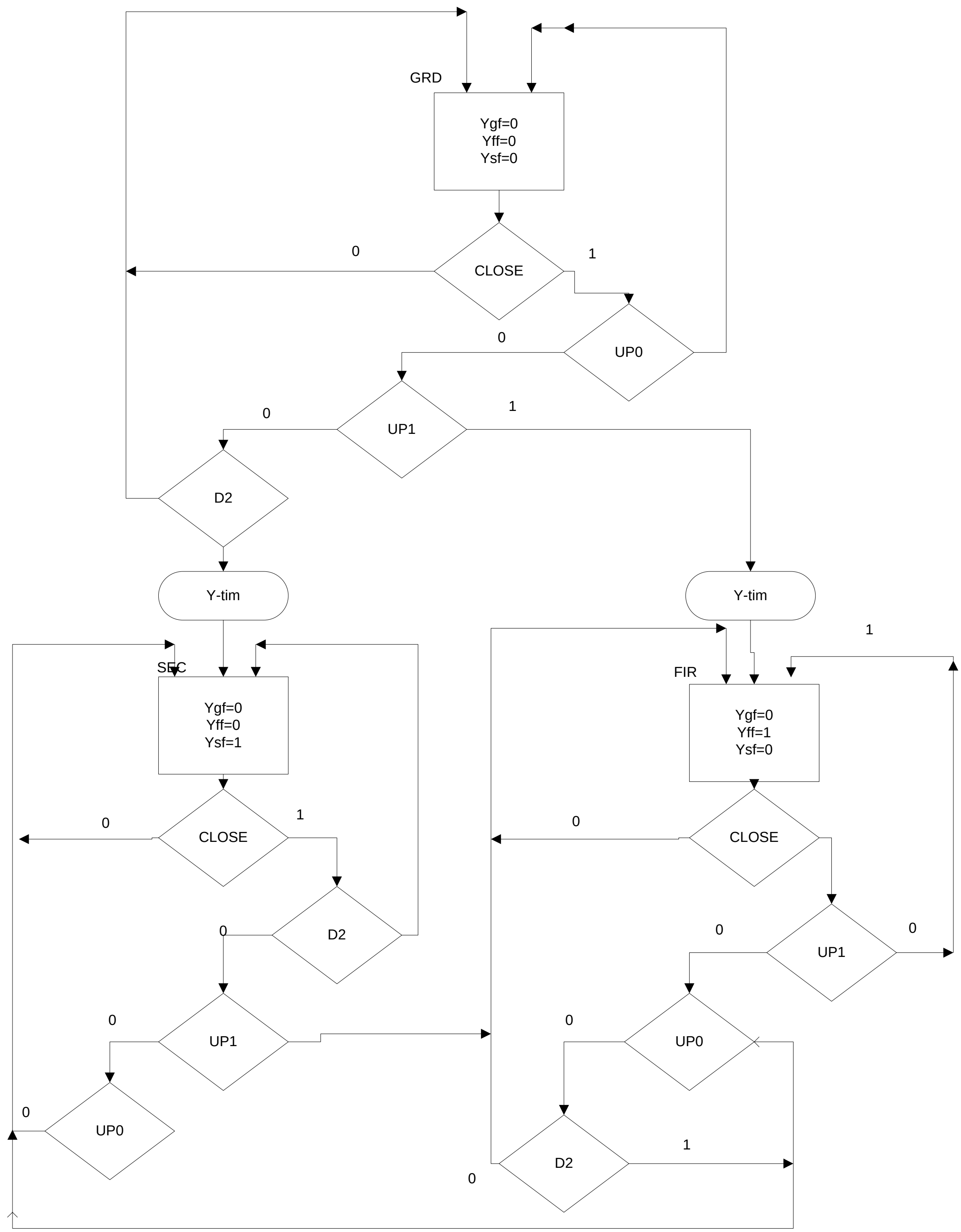

Figure 3.5; interconnected ASM chart

3.3.5 Datapath

The requirements for the design of the Datapath are specified inside the state and conditional boxes. The Datapath for the elevator controller consists of a pulser, three counters and gates. The pulser continuously generates pulses into the counters. The 4-bit binary counters i.e. counter 1 and counter 2 as shown in figure 3.6, are used to determine the stability of the pulse, to generate a low frequency of about one pulse per second to observe slow changes in the digital signal.

The third counter, counter 3 is a down counter with parallel load, this counter determines the open /close door operations, in this counter the binary number decreases by one for each input pulse. An initial binary number(D3D2D1D0 = 0111) is transferred into the counter 3 prior to the count operation. The count starts at Q3Q2Q1Q0 = 0111 and decrements by 1 after every clock pulse.

When the count reaches Q3Q2Q1Q0 = 0000, counter 1 and 2 are cleared and counter 3 stops counting(door closes). This operation requires two NOR gates and a NAND gate to guarantee that the counters are cleared when Q3Q2Q1Q0 = 0000. If OPEN = 1 counter3 is set and the count continues, then if CLOSE = 1 count stops. Therefore door opens for seven seconds to allow passengers on or off and then closes automatically. Pressing the open push button inside the elevator will cause the door to reopen or stay open longer than the preset time as long as the elevator is not moving. Sequence of operation for the counter is shown in table 3.1

The elevator controller circuit operates the motor, which moves the elevator or does not move it. The output signals are send to the motor to control its behaviour.

The following are the transitions which take place;

- Immediate floor up
- Immediate floor down
- Jump one floor up
- Jump one floor down
- Up through immediate floor
- Down through immediate floor

Table 3.1

Sequence of Operation for Counter 3

Outputs Q3 Q2 Q1 Q0	Number of input Pulses received.	Condition
0 1 1 1	0	Door Open
0 1 1 0	1	Door Open
0 1 0 1	2	Door Open
0 1 0 0	3	Door Open
0 0 1 1	4	Door Open
0 0 1 0	5	Door Open
0 0 0 1	6	Door Open
0 0 0 0	7	Door Closed

The elevator moves to immediate floor up when present state is GRD and next state is FIR or present state is FIR and next state is SEC.

Immediate floor up = (Q0 and Q1) or (Q1 and Q2)

= (Q0.Q1)+(Q1.Q2)

The combinational logic required for this operation consists of two AND gates and one NOR gate.

The elevator moves to immediate floor down when current state is FIR and next state is GRD or when current state is SEC and next state is FIR.

Immediate floor up = (Q1 and Q0) or (Q2 and Q1)

= (Q1.Q0)+(Q2.Q1)

The combinational logic required for this operation consists of two AND gates and one NOR gate.

The elevator jumps one floor up when present state is GRD and next state is SEC, this operation requires one AND gate.

Jump one floor up = (Q0 and Q2)

= (Q0.Q2)

The elevator jumps one floor down present state is SEC and next state is GRD this requires one AND gate.

Jump one floor down = (Q2 and Q0)

= (Q2.Q0)

The elevator goes up through immediate floor when the current state is GRD and there are two requests, the first from state SEC and the other from FIR. The gate required for this operation is a three input AND gate.

Up through immediate floor = (Q0 and Q1 and Q0)

= (Q0.Q1.Q2)

Up through immediate floor = (Q2 and Q1 and Q0)

=(Q2.Q1.Q0)

A diagram showing the Datapath for the controller is shown in figure 3.6

3.3.6 State table

A state table for a controller is a list of present state, inputs and their corresponding next state and outputs. The ASM chart was converted into a state table from which the sequential circuit of the controller was designed.

State GRD, FIR and SEC are assigned the following binary values; GRD = 001, FIR = 010 ,SEC = 100. There are ten inputs and three outputs. The inputs are taken from the conditions in the decision boxes and the outputs are equivalent to the present state of the controller.

There are many don’t-care input conditions that have been included in the state table. Three flip flops are needed and they are labeled Ysf, Yff and Ygf.

The state table corresponding to the ASM chart is shown in table 3.2.

Table 3.2

State Table for the controller

Present State			Inputs								Next State			Outputs		
Ysf	Yff	Ygf	CLOSE	UP0	X0	UP1	DI	X1	D2	X2	Ysf	Yff	Ygf	SEC	FIR	GRD
0	0	1	X	X	X	0	0	0	0	0	0	0	1	0	0	1
0	1	0	1	1	1	0	0	0	0	0	0	0	1	0	1	0
1	0	0	1	1	1	0	0	0	0	0	0	0	1	1	0	0
0	0	1	1	0	0	1	1	1	0	0	0	1	0	0	0	1
0	1	0	X	0	0	X	X	X	0	0	0	1	0	0	1	0
1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0
0	0	1	1	0	0	0	0	0	1	1	1	0	0	0	0	1
0	1	0	1	0	0	0	0	0	1	1	1	0	0	0	1	0
1	0	0	X	0	0	0	0	0	X	X	1	0	0	1	0	0

3.3.7 Control Logic

The procedure used to design the sequential circuit starting from a state table is as follows;

- Chose the type of flip-flops to be used for the relationship.
- Determined the minimum number n, of bistables required from the relationship.
- Derived the simplified flip-flop input equations and output equations.
- Drew the logic diagram.

By inspecting the state table, the next states are equal to the corresponding inputs. Thus the input equations are obtained directly from the state table. The D flip-flop receives the designation from its ability to hold data in its internal storage. The next state of the D flip-flop is equal to the input prior to the application of a clock pulse. The use of D type memory elements simplified the design due to the ease in formulating the design equations from the state table.

The control logic diagram is as illustrated in figure 3.7.

3.3.8 Display logic

The display unit is basically a decoder consisting of combinational logic that produce seven outputs to a 7-segment display, to indicate either ground, first or second floor.

The complete controller circuit diagram is as shown in figure 3.8.

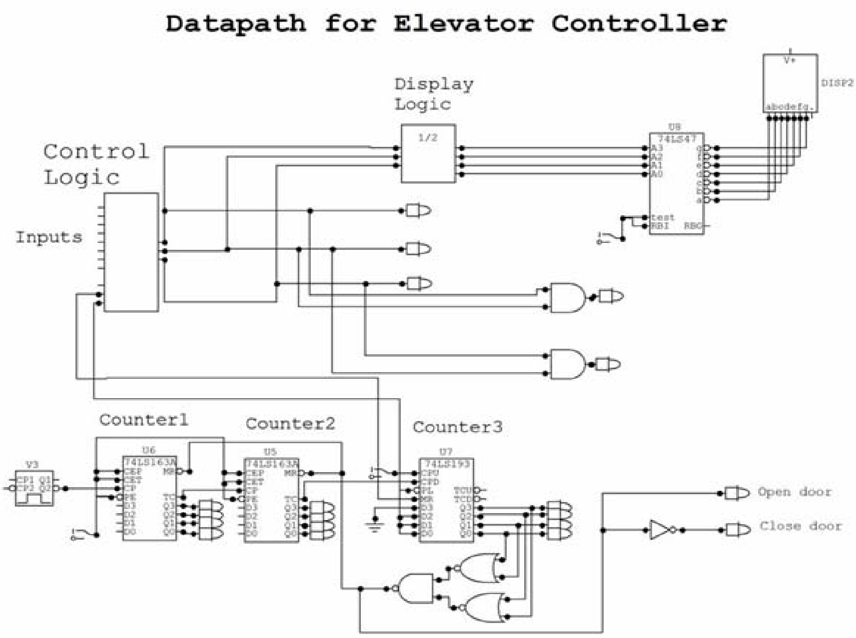

Figure 3.6 Datapath

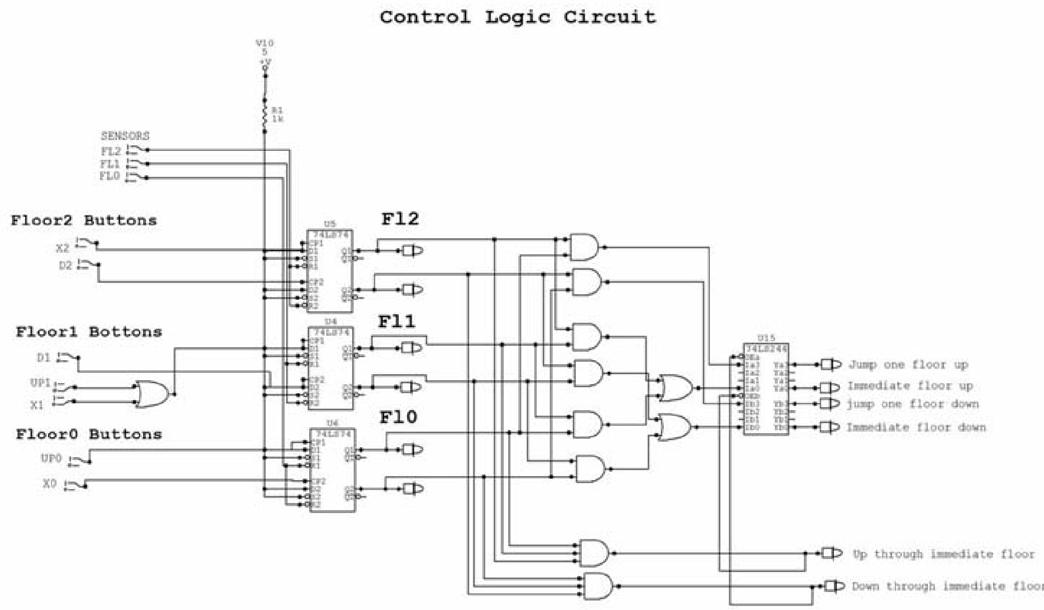

Figure 3.7 Control logic circuit

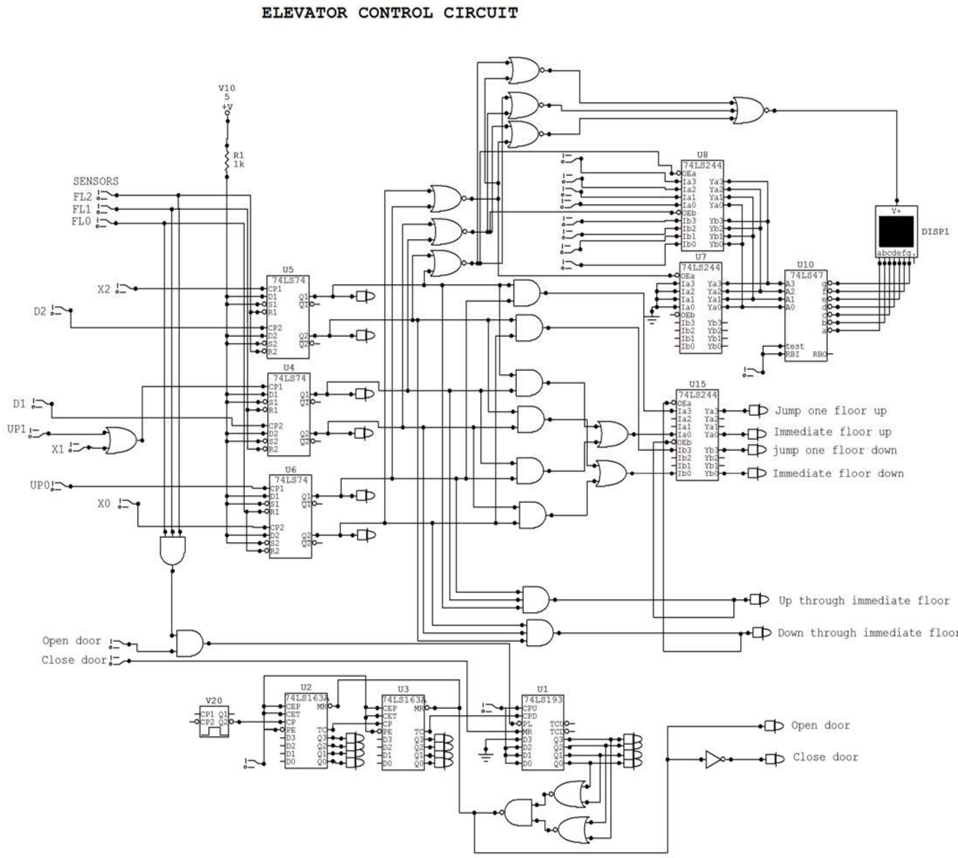

Figure 3.8 Complete controller circuit

CHAPTER 4

Results and Discussion

In this chapter, the results that were obtained by computer simulation are presented and discussed. The simulation software used was the Circuit Maker which has most of the SSI and MSI components in the library .

In the simulation the elevator floor push-buttons were simulated as logic switches and the output indication displays used were;

- A seven- segment indicator was used for displaying any of the three decimal digits 0 through 2. The display shows a “0” when the system is in ground floor. The display shows a “1” when the system is in first floor and a “2” when the system is in second floor state. The inputs to the display logic come from the state control logic.

- The logic displays were used to display the output signals.

4.1 Controller response

The elevator controller receives requests from the call switches from each floor and floor select switches and then displays outputs accordingly, as shown in table 4.1. All possibilities were taken into consideration and analysis was done for each input.

The output signals are interpreted as follows;

When the elevator arrives in a new state the previous state is cleared, a pulse was generated that resets the previous state. Immediately the previous state is cleared the display indicates the current state, as shown in table 4.2. A delay is derived from the counter, pressing the open door button causes the door to reopen or stay open for longer as long as the elevator is not moving. A slow clock signal of 1Hz controls the time the door remains open when the elevator stops at a floor. The door operation outputs are synchronized with the clock. Table 4.3 shows the states of these synchronized outputs.

Table 4.1

Tabulated results for the controller

Present State Q3 Q2 Q1	Inputs X0,UP0,X1,UP1,D1,X2,D2	Outputs Q3 Q2 Q1	Details
0 0 1	X0 or UP0	0 0 1	Elevator remains on ground floor
0 0 1	X1 or UP1 or D1	0 1 0	Elevator is raised to immediate floor up
0 0 1	X2 or D2	1 0 0	Elevator jumps one floor up
0 1 0	X1 or UP1 or D1	0 1 0	Elevator remains on first floor
0 1 0	X0 or UP1	0 0 1	Immediate floor down
0 1 0	X2 or UP2	1 0 0	Immediate floor up
1 0 0	X0 or UP0	0 0 1	Elevator jumps one floor down
1 0 0	X1 or UP1 or D1	0 1 0	Elevator goes to immediate floor down
1 0 0	X2 or UP2	1 0 0	Elevator remains on second floor

Table 4.2

Display logic response

Outputs Q3 Q2 Q1	Display Details
0 0 1	Display a “0”
0 1 0	Display a “1”
1 0 0	Display a “2”

Table 4.3

Door output variables

Output Variable	Transition
DOOR	0 = Close 1 = Open

4.2 Discussion of Results

The behaviour of the controller agreed with the expected theoretical results. For example when the elevator is waiting on the ground floor with the door open. If, after a period of seven seconds established by the clock, there is no request from the push buttons, the door closes and the elevator remains waiting on the ground floor.

When the system is in GRD state and there is a request from second floor button, the system bypasses the FIR state and goes to SEC state on the next clock pulse. During the transition the door closes and the elevator moves upward. The door remains closed until an arrival signal is received when the elevator arrives in second floor, which activates a switch and opens the door.

When the system is in SEC state, it remains there until there is a request from first floor or ground floor buttons, which causes the system to go down. Alternatively, it remains in SEC state until there is a request from X2 or D2 or the OPEN button is pressed(X2 + D2 + OPEN), which causes the door to open. The current floors are displayed accordingly.

Chapter 5

Conclusion and Recommendation

5.1 Conclusion

In modern digital system the synthesis of finite state machine is impractical for their complexity, thus specialized methods have to be implemented. In this design of the elevator control circuit, implementation was performed by adopting the Algorithmic State machine procedure. The method was of great importance because of its simplicity and procedural steps that specify how to obtain a solution to a problem.

The sequential logic process was employed in the design, adoption of D type memory elements simplified the design because the input functions could be obtained directly from the state table. Design with D flip-flops proved simple and clear.

The necessary outputs from the control logic and display were identified. The results that were obtained from the simulation agreed with the theoretical expectation.

5.2 Recommendations for further work

The proposed elevator control can be extended as part of future study in the following ways;

- Practical implementation of the actual circuit, however changes are expected in the results due to electronic glitches, noise or transmission line effects.
- The elevator may find other ways to accommodate the need of passengers without delays caused by the “up and down” philosophy. This may be replaced by more interactive devices which responds to voice commands or electronic badges worn by employees on specific floors.
- Elevators which shall anticipate surges at various times of the day, such as shift changes or lunch breaks. This problem can be better solved by the use of microprocessor based controller. The microcontroller in charge of this floors can send more cars at a specific time to handle the sudden demand.

- Improving safety in the elevator- The measured weight of the elevator car can be used by the controller to precisely implement special dispatching strategies such as bypassing floor calls when the elevator is full.
- For a large number of floors, the need to use a microcontroller such as 8051 can be adopted.

References

1. M. Morris Mano. Digital Design
   Third Edition, prentice Hall USA, 2002

2. B. R Thomas. Digital Design with Standard MSI and LSI
   Wiley, New York. 1975

3. Mwangi Elijah. Foundation of Digital Electronics
   Department of Electrical and Information Eng; University of Nairobi Press, 1998

4. J. R Gibson. Electronic Logic Circuits
   First Edition, Edward Arnold, 1979

5. Internet sites: www.freepatentsonline.com/467405.html as at 12th May 2009.
   www.fyecw.com/system_ruibo/uploadfiles/2007455619251.pdf as at 4th May 2009.