AN INTERACTIVE FRAMEWORK FOR TEACHING FUNDAMENTALS OF DIGITAL LOGIC DESIGN AND VLSI DESIGN

Brahmasree Battina

Thesis Prepared for the Degree of

MASTER OF SCIENCE

UNIVERSITY OF NORTH TEXAS

August 2014

APPROVED:

Gayatri Mehta, Major Professor
Kamesh Namuduri, Committee Member
Hyoung Soo Kim, Committee Member
Shengli Fu, Chair of Department of Electrical Engineering
Costas Tsatsoulis, Dean of the College of Engineering
Mark Wardell, Dean of the Toulouse Graduate School

Battina, Brahmasree. *An Interactive Framework for Teaching Fundamentals of Digital Logic Design and VLSI Design*. Master of Science (Electrical Engineering), August 2014, 47 pp., 42 figures, 6 tables, references, 18 titles .

Integrated Circuits (ICs) have a broad range of applications in healthcare, military, consumer electronics etc. The acronym VLSI stands for Very Large Scale Integration and is a process of making ICs by placing millions of transistors on a single chip. Because of advancements in VLSI design technologies, ICs are getting smaller, faster in speed and more efficient, making personal devices handy, and with more features. In this thesis work an interactive framework is designed in which the fundamental concepts of digital logic design and VLSI design such as logic gates, MOS transistors, combinational and sequential logic circuits, and memory are presented in a simple, interactive and user friendly way to create interest in students towards engineering fields, especially Electrical Engineering and Computer Engineering. Most of the concepts are explained in this framework by taking the examples which we see in our daily lives. Some of the critical design concerns such as power and performance are presented in an interactive way to make sure that students can understand these significant concepts in an easy and user friendly way.

ACKNOWLEDGEMENTS

I am very thankful to my major professor Dr. Gayatri Mehta for her encouragement, suggestions, guidance and constant support for this research work. Her support and guidance assisted me in successfully completing this work.

I am thankful to the committee members: Dr. Kamesh Namuduri and Dr. Hyoung Soo, Kim for spending their valuable time and for their support.

I am also thankful to my family for being with me and for supporting me throughout my life.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This chapter presents the importance of digital logic design and VLSI design and prepares a background for the work presented in the thesis. In this thesis, we present a framework that is designed to help students understand the fundamental concepts of digital logic design and VLSI design in an interactive way. The acronym VLSI stands for "Very Large Scale Integration" and is a process of making ICs (Integrated Circuits) by placing millions of transistors on a single chip. ICs have broad range of applications in healthcare, military etc. As the VLSI design technology is improving, ICs are getting smaller in size and faster in speed, making personal digital devices handy, comfortable to use, and with more features. The next section will explain the key objective in doing this thesis work.

1.1. Objective of the Work

This work presents a framework that is designed to present the fundamental concepts of digital logic design and VLSI design in a simple, interactive, and user friendly way in order to attract students towards engineering fields, especially Electrical Engineering and Computer Engineering.

The concepts are presented in an interactive way and are easy to understand. This framework can be used to make this area of study easy to understand and attractive to students. Previously, many people have worked in this field and are conducting extensive research in developing interactive learning environments for several engineering subjects throughout the world. In paper [6], the authors discussed an Interactive Learning Environment for making the fundamental concepts of Power Electronics and Electrical Machines easy and interesting to the students. In paper [7], the authors discussed an interactive learning environment for solid mechanics course in engineering. This platform is developed using Adobe Flash software and is implemented in the university engineering academic program to make the concepts of solid mechanics easy for students by teaching them in more interesting and interactive way. These experts are working on expanding the work to other engineering courses as part of the research program called "Smart Sparrow" [15]. In paper [8], the authors discussed an interactive E-learning platform to teach the concepts of Hydraulic Engineering to the

university students. People from the Massachusetts Institute of Technology (MIT) have designed a web-based circuit simulator called "Circuit Lab" that helps electrical engineering students simulate their electronic circuits online [10].

In paper [16], the author's discussed about a Windows based interactive learning aid called "WinLogiLab", for elementary digital logic design. In paper [17], the authors discussed an interactive teaching aid using breadboard for digital logic design. In paper [18], authors discussed an interactive learning environment for VLSI design course.

1.2 Organization of Thesis Work

In chapter 2, ideal and non-ideal characteristics of a CMOS transistor are discussed; with emphasis on how these concepts can be presented in an easy and interactive way. Chapter 3 is a brief introduction to logic gates, and their implementation into this framework is presented in detail. Chapter 4 presents an introduction to the combinational circuits like multiplexer, decoder, and encoder and how these concepts are implemented in this framework. Chapter 5 is an introduction to sequential circuits and an example of Finite State Machine (FSM) is presented. Chapter 6 is an introduction to power consumption, techniques to reduce power consumption, calculation of activity factor, and calculating the delay in circuits. Chapter 7 is a brief introduction to types of adders and how they are implemented in this framework is presented. In chapter 8, memories and their implementation is presented. Finally, Chapter 9 presents conclusion and future work.

CHAPTER 2

MOS TRANSISTORS

The MOS transistors are the basic building blocks of any VLSI circuit. There are two types of MOS transistors, NMOS and PMOS. NMOS transistors are made by adding N type impurities to a P type substrate and in converse, the PMOS transistor is made by adding P type impurities to the N type substrate. The electrons are the majority charge carriers in NMOS transistors and holes are the majority carriers in PMOS transistors. The NMOS transistor is considered as a four terminal device with controlling gate terminal, substrate or body terminal, and two other terminals, source and drain stacked in N-type materials. The source and drain terminals have free electrons and the body has holes.

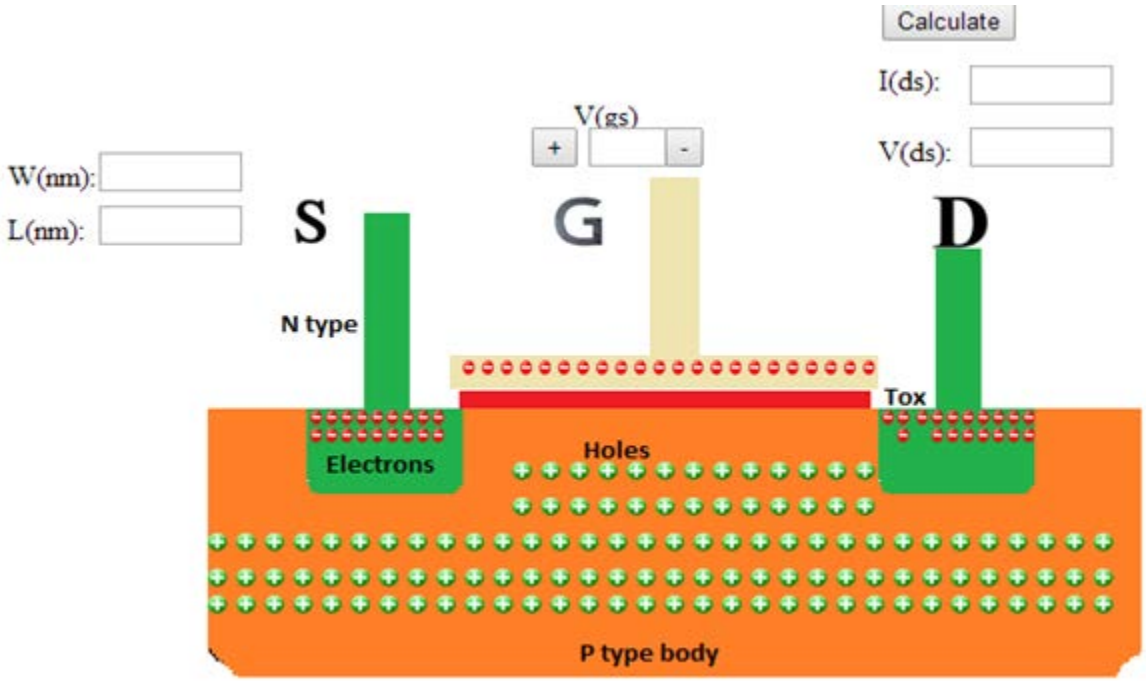


Figure 2.1 NMOS transistor in Accumulation mode

- In figure 2.1, the green '+' sign represents the holes and red '-' sign represents the electrons. The source, gate, and drain terminals are marked S, G, and D respectively. Text boxes are provided to input the width, length, gate to source voltage (V_{gs}), and drain to source voltage (V_{ds}), which are required to calculate drain current I_{ds}.

- If the user applies a negative (<0V) voltage to the gate terminal, then the negatively charged electrons will be at the gate and the positively charged holes are attracted towards the region below the gate terminal. This is called Accumulation mode.
- If the user applies a slightly positive voltage (>0V) to the gate terminal, the gate will attract some holes, and the holes which are in the body are repelled directly from the region below the gate, creating some gap in between them. This occurrence is called the Depleting mode.

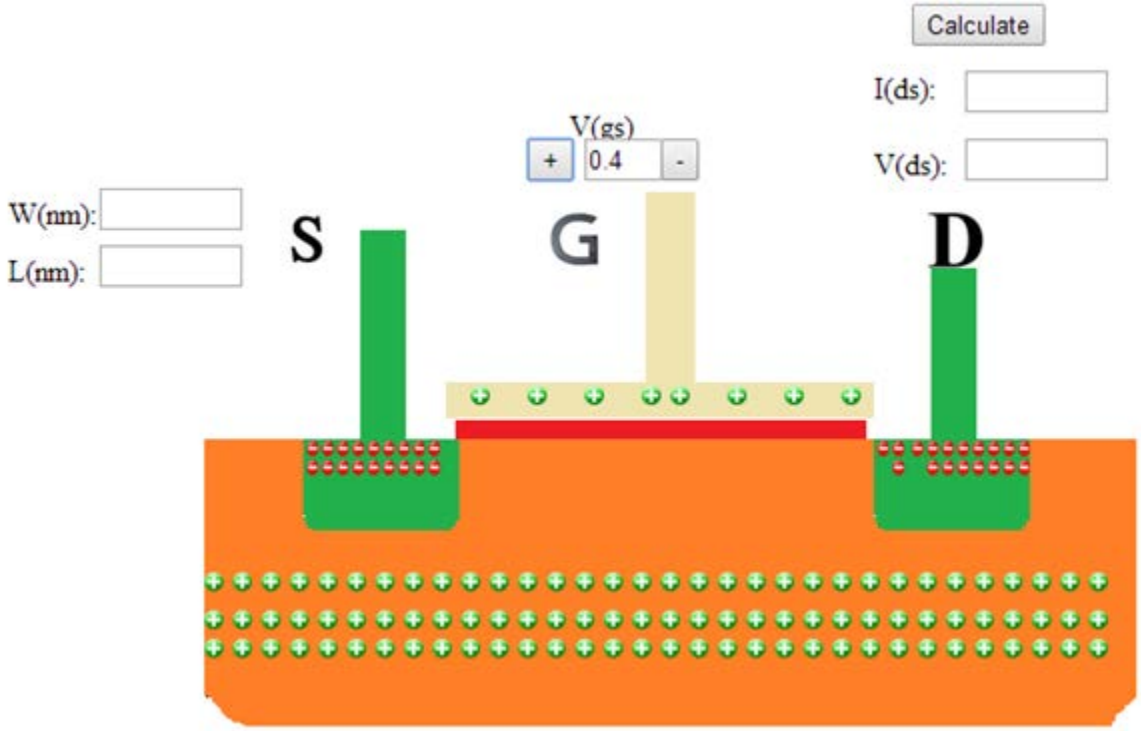


Figure 2.2 NMOS transistor in Depleting mode

- If the user apply a more positive voltage greater than the critical threshold voltage (V_t), then the gate attracts more positive charges. The free electrons of the source and drain are attracted towards the region beneath the gate and form a conductive path of electrons from source to drain called Inversion layer, and this mode is called Inversion mode [Figure 2.3].

2.1. Ideal I-V Characteristics of NMOS Transistor

Based on the gate to source voltage (V_{gs}), the transistor has three regions of operation: Cutoff, Linear, and Saturation.

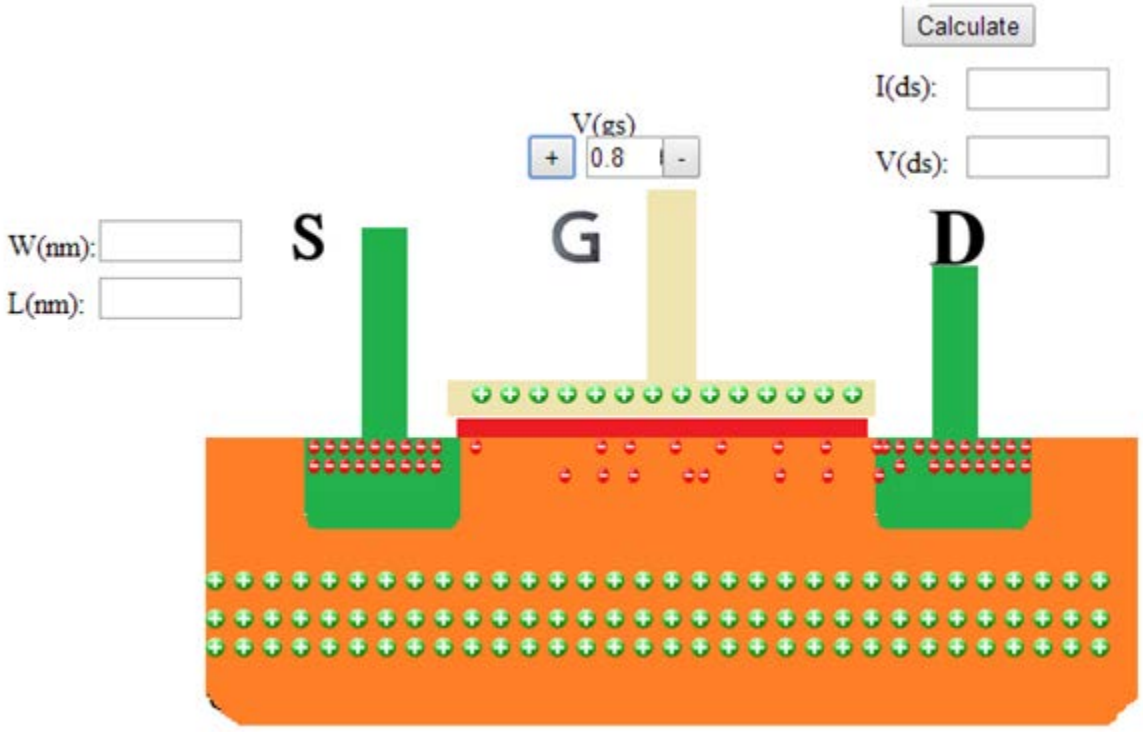


Figure 2.3 NMOS transistor in Inversion mode

- If the source is considered to be grounded (V_{gs}=0 or $V_{gs} < V_t$), the two junctions between the body-source and body-drain are reverse biased and hence there is no conducting path between source and drain. So no current flows from source to drain and hence the transistor is said to be in OFF (0 v) state. This region of operation is called a Cutoff region [Figure 2.2].
- As the voltage $V_{gs} > V_t$, a free flow of inversion electrons will form a conducting path between the source and the drain called a channel. Hence, when the current flows from source to drain, the transistor is said to be in ON (5 v) state. This region of operation is called linear region. The number of charge carriers in a channel and the conductivity both increase as we increase the gate voltage. The potential difference between source and drain is $V_{ds} = V_{gs} - V_{gd}$ [Figure 2.4].

- If 'V_{gs}' still goes on increasing, at some point the channel is no longer inverted near the drain and is pinched off. However, there will be conduction because of the drift of electrons under the influence of positive drain voltage. As the electrons come to the end of the channel they are forced to the depletion region and are accelerated towards the drain. The increasing controlling voltage will no longer affect the current conduction. This region of operation is called a Saturation region [Figure 2.5].

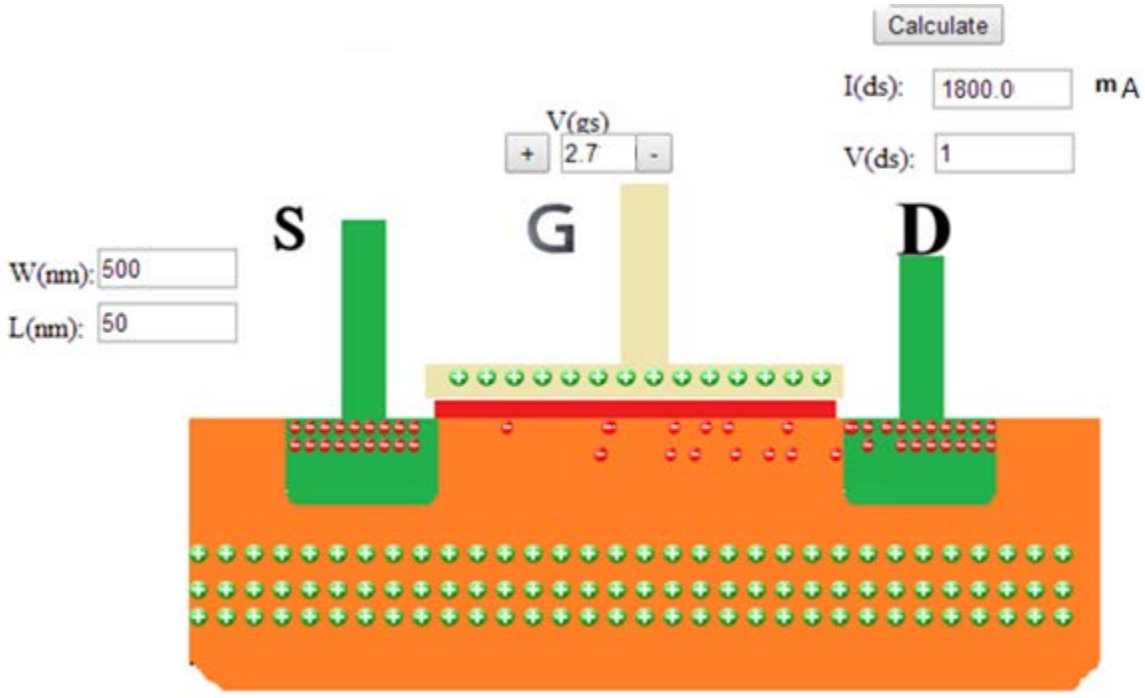


Figure 2.4 NMOS transistor operating in the linear region

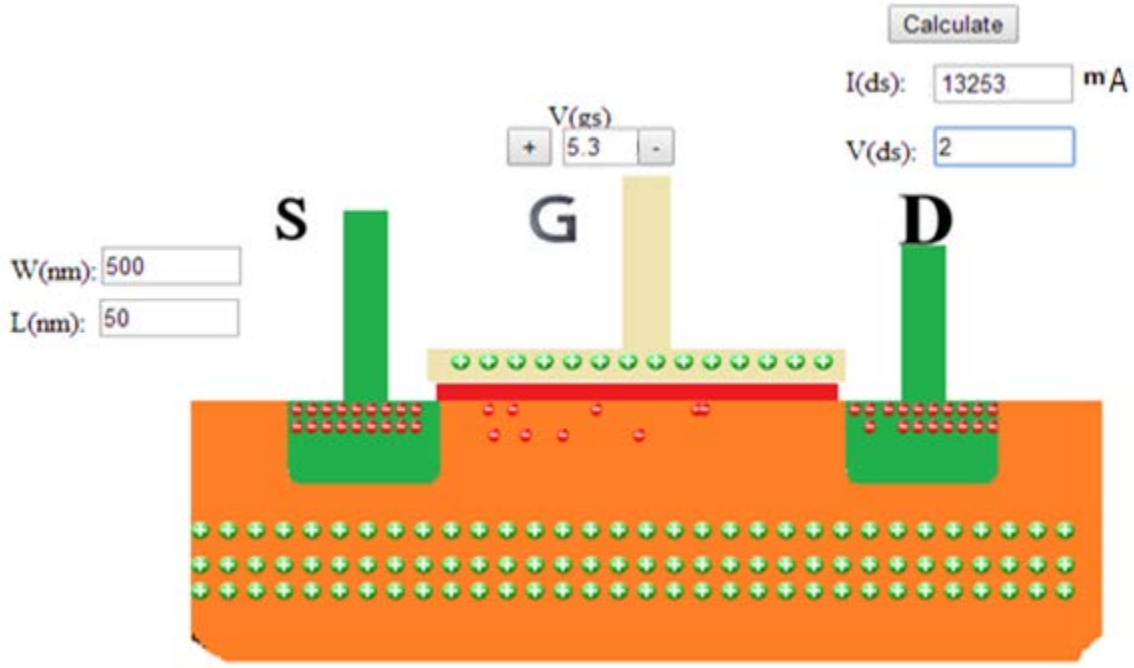


Figure 2.5 NMOS transistor operating in Saturation region

2.2. Non Ideal I-V Characteristics of MOS Transistors

In the ideal characteristics, performance degradation effects such as High field effects, Chanel length modulation, Threshold voltage effects, the temperature effect, and Leakage effects are ignored. The velocity saturation and mobility degradation are the high field effects.

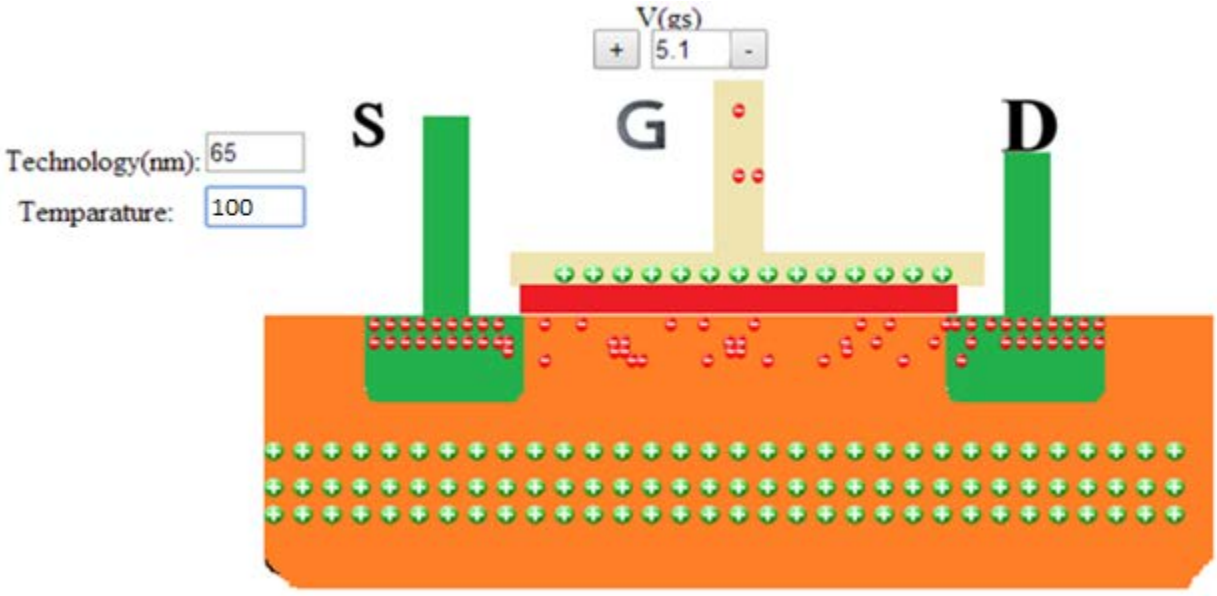


Figure 2.6 High field effects of NMOS transistor

- In figure 2.6, two text boxes are provided for the user to select the Technology node size and the temperature. When set to high vertical field strength (V_{gs}/T_{ox}) and high temperature, the charge carriers in the channel are scattered around because of the collision among themselves and with the Silicon Oxide surface and are thus slowed down. This effect is called Mobility Degradation.
- If the voltage at the drain V_{ds} is increasing, the depletion region around the drain is also increasing, which will effectively decrease the length of the channel. This is called the Channel Length Modulation [Figure 2.7].
- Increasing the voltage between the source and body (V_{sb}), will increase the threshold voltage. This is called Body Effect. Increasing the voltage at the drain terminal will automatically increase the current and hence decrease the threshold voltage. This effect is called Drain Induced Barrier Lowering (DIBL).

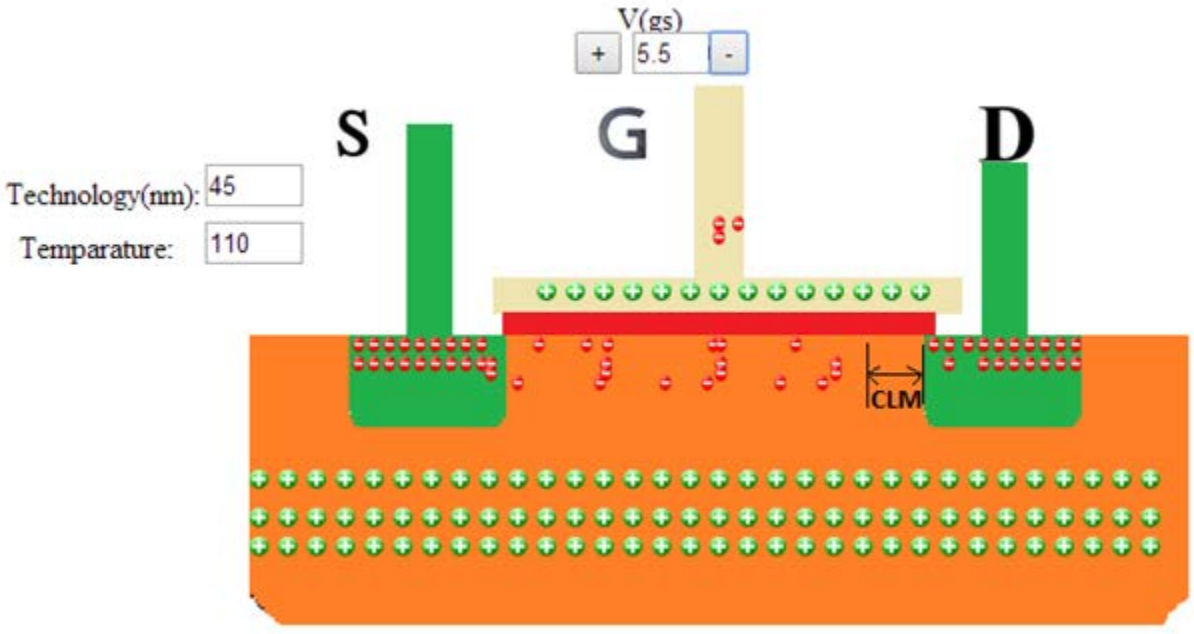


Figure 2.7 Channel Length Modulation

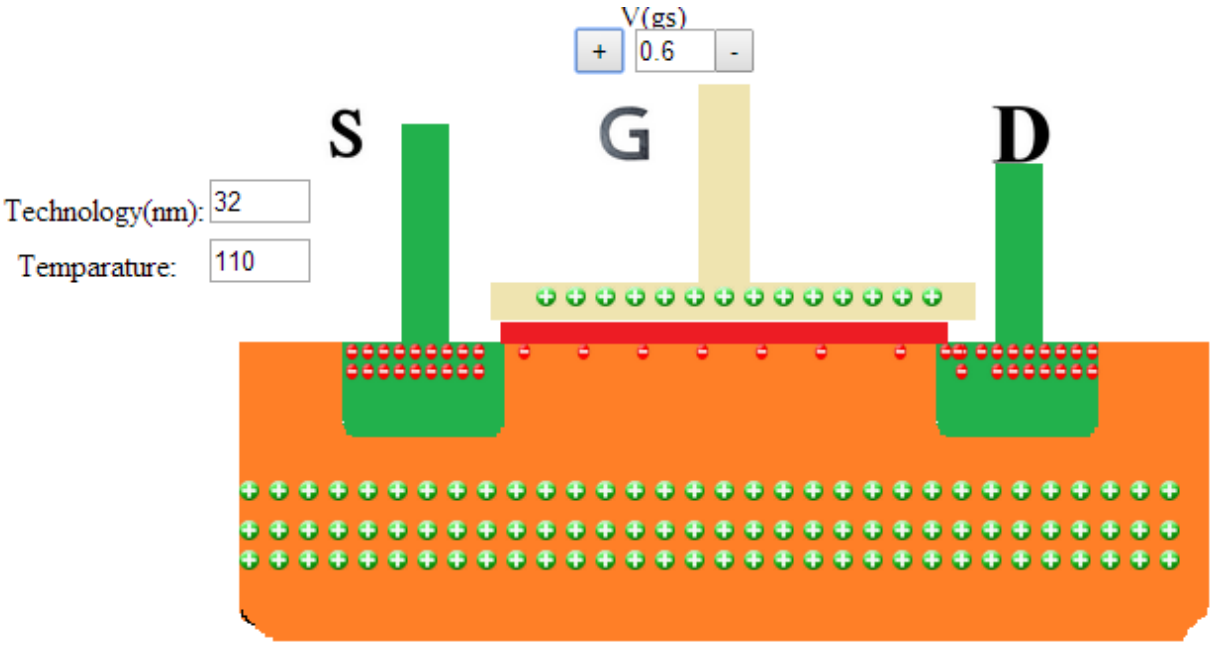


Figure 2.8 Threshold voltage effect

- In figure 2.8, there is a decrease in threshold voltage. The channel is starting to form at V_{gs} =0.6v by itself. This channel forming even at low threshold voltage is due to an increase in temperature and decrease in the oxide thickness.
- As one can see in the figure 2.6, figure 2.7, and figure 2.9, at high V_{gs}, the electrons from the channel are escaping through the oxide towards the gate and causing a small gate current called Gate leakage. This effect is because of a reduction in the oxide thickness and increase in temperature. The overall performance of the transistor will be degraded as the temperature increases.

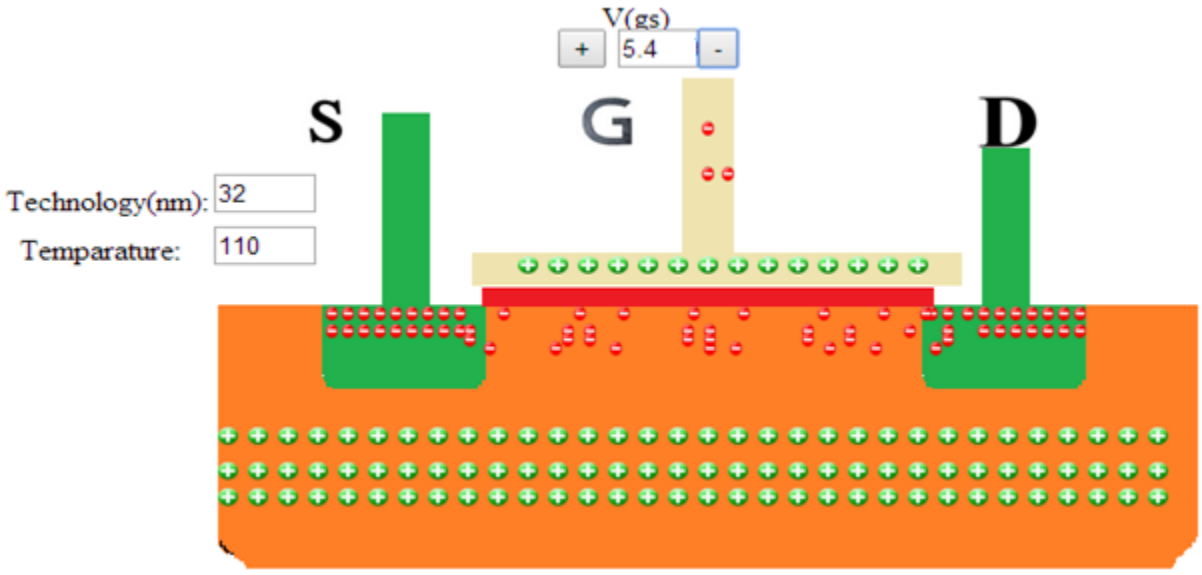


Figure 2.9 Gate leakage

CHAPTER 3

LOGIC GATES

The logic gates are built using the CMOS transistors, which are the basic components in the digital circuits. A logic gate is a physical device which implements the Boolean function or which performs logical operations on the given logical inputs and gives a single logical output. The logic gates have only two input levels: high and low. The high level is represented by a binary 1, and the low level is represented by a binary 0. The input level 1 is considered as a true case, and the input level 0 is considered as a false case. The basic logic gates are: AND, OR, NOT, NAND, NOR.

3.1 AND Gate

The AND gate implements a logical conjunction operation. The AND gate is a two or more input function that produces a single output which is high (1) only if all the inputs are high (1). In all other input cases the output of an AND gate is low (0). A two input AND gate works according to the truth table shown below.

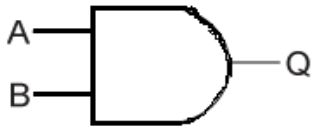


Figure 3.1 Symbol of AND gate

Input A	Input B	Output Q
0	0	0
0	1	0
1	0	0
1	1	1

Table 3.1 Truth table of AND gate

The same two input and gate can be explained in a simpler and interactive way.

- In figure 3.2, the two inputs A and B are represented by two switches (small lines above the main circuit) respectively. Based on connecting or disconnecting the switches to the circuit by clicking on them, the bulb will turn on or turn off. The high and low output levels are represented by the bulb in on state and the bulb in off state, respectively.

Figure 3.2 Two input AND gate

- When the user clicks on switch A, it gets connected to the circuit (switch is closed). But the switch B is still not connected (open). So the bulb will be in off state and vice versa.

Figure 3.3 (a) AND gate with switch A closed

Figure 3.3 (b) AND gate with switch B closed

- The bulb will be in on state only when both the switches A and B are closed (connected to the circuit, both inputs high), making a conducting path.

Figure 3.4 AND gate with both switches closed

3.2 OR Gate

The OR gate implements logical disjunction operation. The OR gate is a two or more input function and produces a single output which is low (0) only when all the inputs are low

(0). In all other input cases it produces a high (1) output. A two input OR gate works according to the truth table shown below.

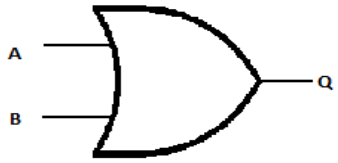


Figure 3.5 Symbol of OR gate

Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	1

Table 3.2 Truth table of OR gate

The same two input OR gate can be explained in a simple and interactive way.

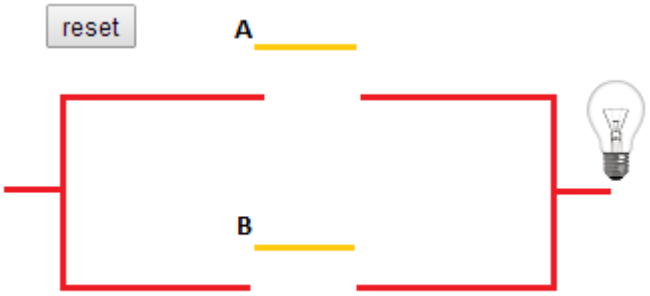


Figure 3.6 Two input OR gate

- As in figure 3.6, the two inputs A and B are represented by two switches respectively. The bulb will be turned on or turned off, based on connecting or disconnecting the switches by clicking on them. The high and low output levels are represented by the bulb in on state and the bulb in off state, respectively.
- When switch A is connected, there will be a conducting path, even though switch B is open, the bulb will be ON and vice versa. The bulb will be ON also when both the switches A and B are connected.

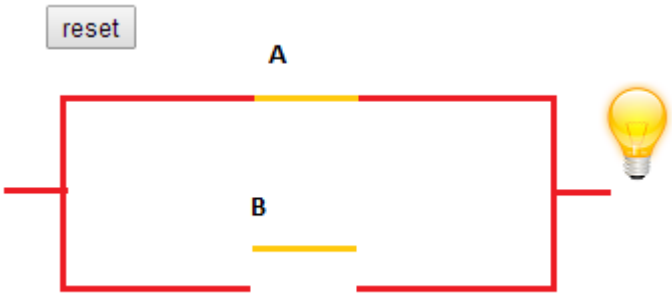


Figure 3.7 (a) OR gate with switch A closed

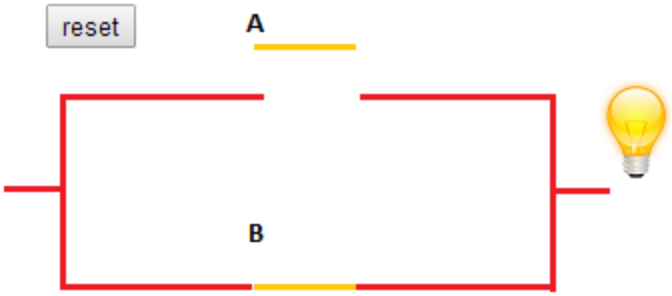
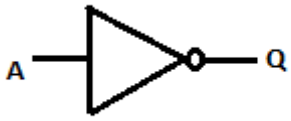


Figure 3.7 (b) OR gate with switch B closed

Figure 3.7 (c) OR gate with both the switches closed

3.3 NOT Gate

The NOT gate is often referred to as Inverter, implements logical negation operation. The NOT gate is a single input function that produces a single output, which is high (1) if the input is low (0), and low (0) if the input is high (1). A simple NOT gate can be shown as below.



Input A	Output Q
0	1
1	0

Figure 3.8 Symbol of NOT gate

Table 3.3 Truth table of NOT gate

- The input A is represented by the switch and can be connected or disconnected to the circuit by clicking on the switch. The high and low output levels are represented by the bulb in on state and the bulb in off state, respectively.

Figure 3.9 NOT gate

Figure 3.10 NOT gate with switch an opened.

CHAPTER 4

COMBINATIONAL CIRCUITS

The combinational circuits are the circuits that do not require memory because the output at any instant of time depends only on the present states of inputs. So the combinational logic circuits can also be called as time-independent logic. Multiplexers, Decoders and Encoders are the commonly used combinational circuits in many digital systems.

4.1 Multiplexer

A multiplexer is commonly called a data selector because it accepts many input signals and gives only one output signal based on the select line. So a multiplexer is a multiple input - single output device. A multiplexer with $2\hat{}n$ input will have 'n' select lines. So a multiplexer with 1 select line will have 2 inputs, with 2 select lines will have 4 inputs, and with 3 select lines will have 8 inputs.

- As shown in figure 4.1, the 8 inputs to the multiplexer are represented by the different colored balls. The drop down menu will be the select line to select the input that needs to be transferred to the output.
- If the yellow ball is to be the output, then just select the option 2 in the select line using drop down menu. To select the next input, reset the multiplexer by clicking on the reset button and choose the other option.

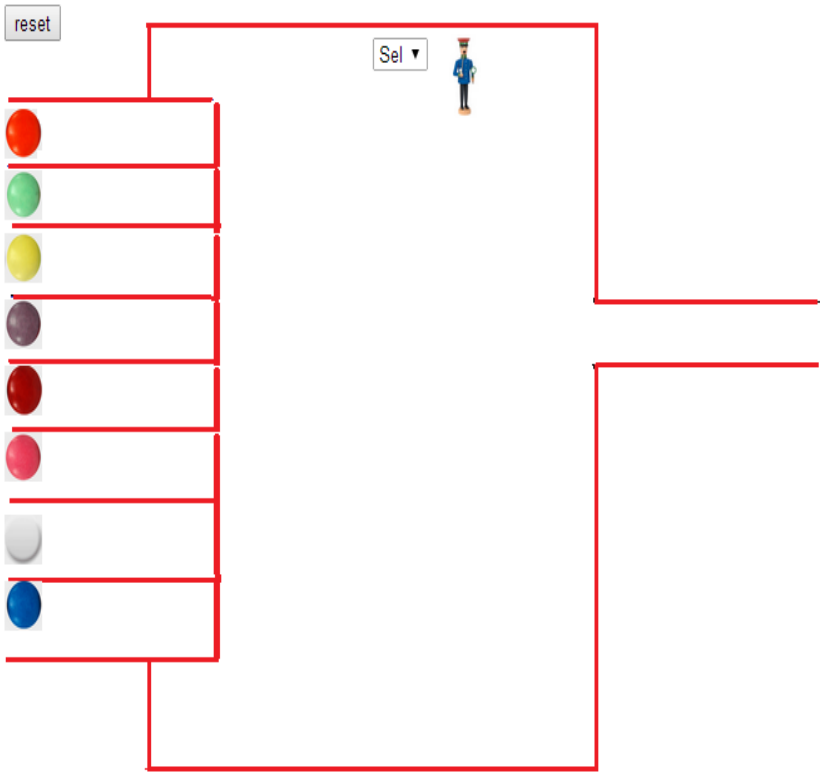


Figure 4.1 Multiplexer

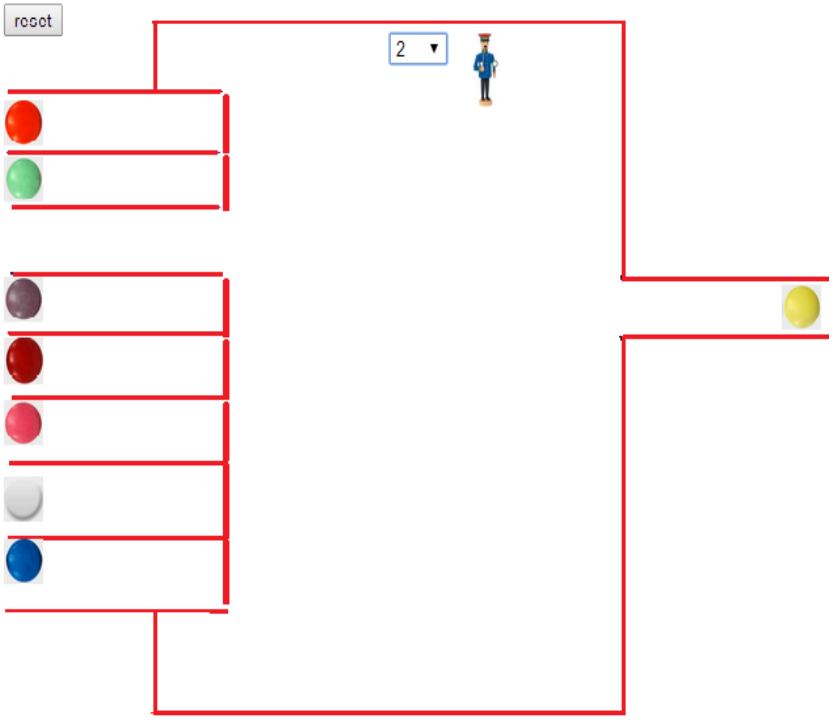


Figure 4.2 Multiplexer with selected input

Inputs			
A2	A1	A0	Output D
0	0	0	0 (red ball)
0	0	1	1 (green ball)
0	1	0	2 (yellow ball)
0	1	1	3 (violet ball)
1	0	0	4 (crimson ball)
1	0	1	5 (pink ball)
1	1	0	6 (white ball)
1	1	1	7 (blue ball)

Table 4.1 Truth table of a Multiplexer

4.2 Decoder

The decoder will take a fewer number of inputs and gives a greater number of outputs. Usually a decoder with 'n' number of inputs gives the $2\hat{}n$ number of outputs. So the decoder with 2 inputs will give 4 outputs, and with 3 inputs gives 8 outputs. The decoders are multiple input - multiple output devices. The smaller decoders can be cascaded in order to build the bigger decoders. These decoders will have a fourth binary input called enable. This enable input is useful to select one of the two decoders at a time. The decoders are extensively used for decryption purposes and in the memory circuits to select the row and column address.

Inputs			Outputs			
En	A	B	Q3	Q2	Q1	Q0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Table 4.2 Truth table of 2-to-4 Decoder

This concept of a 2-to-4 decoder circuit can be explained in a simple and interactive way.

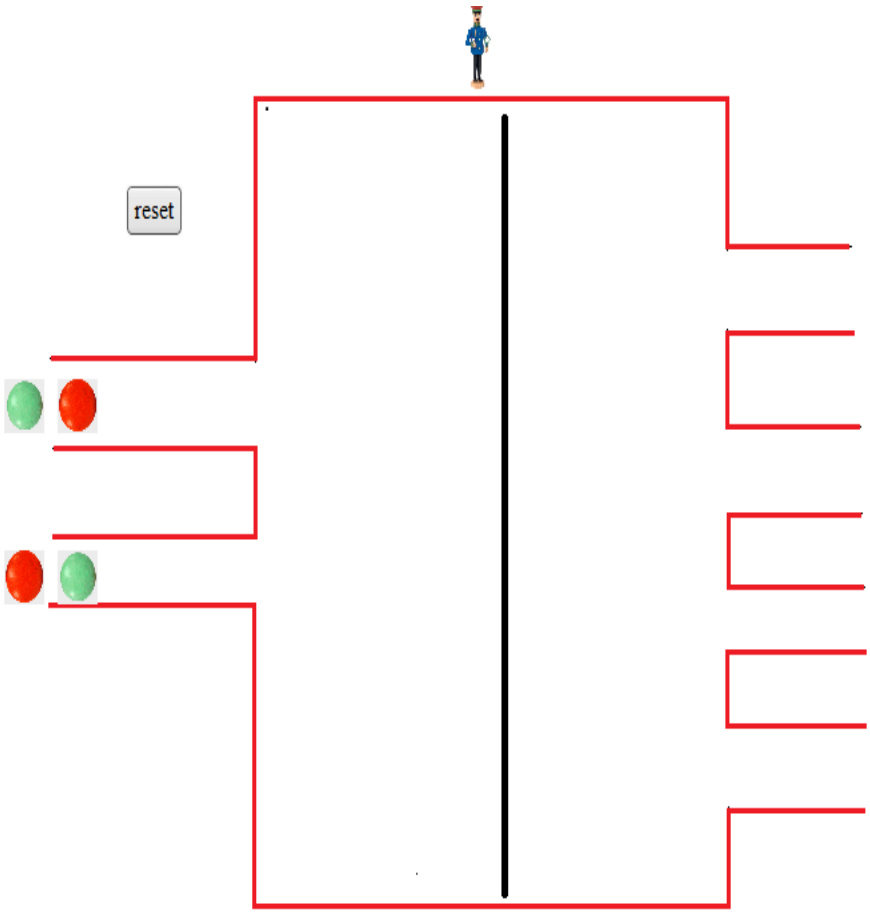


Figure 4.3 2-to-4 Decoder

- In figure 4.3, the inputs are shown in the red and green balls, which represent a logic 0 and logic 1 respectively. One can choose any input combinations by clicking on any one of the balls in each input line. When the user click on the input balls, the logic 1 and logic 0 automatically appear beside them.
- The black line in the middle of the diagram is the Enable signal and is used to enable or disable the decoder. This means the decoder works only if the user enables it by removing the black line by clicking on the small human picture on the top of the diagram.
- For example, to select 0 and 1 as inputs, click on the red ball in the first input line and green ball in the second input line.

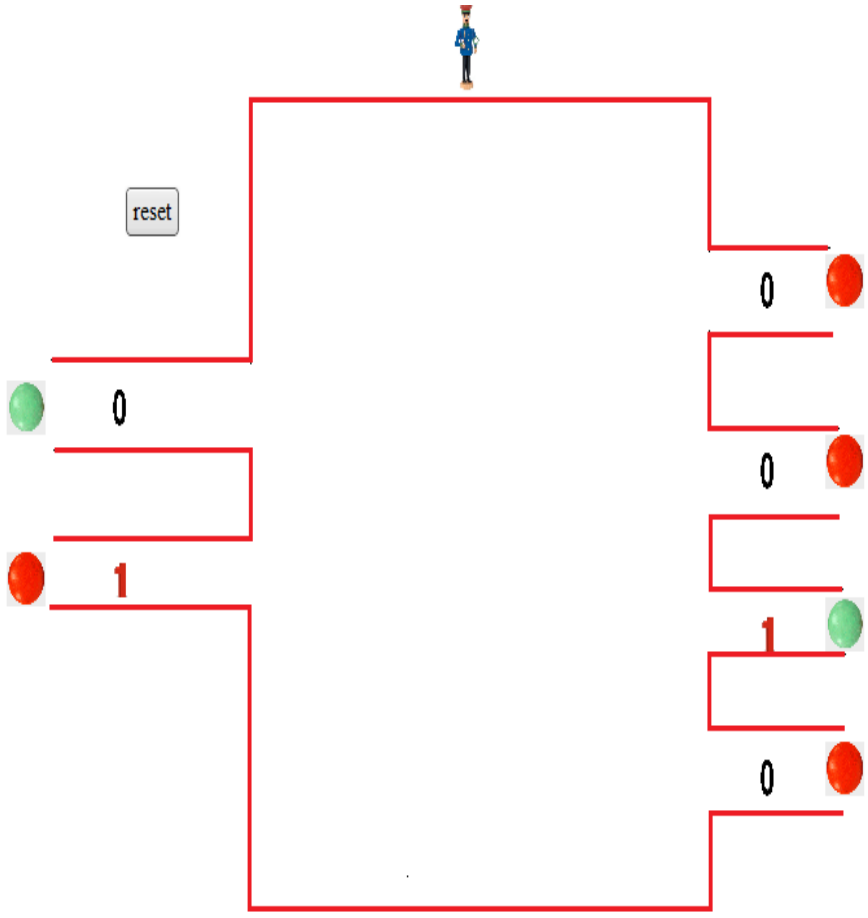


Figure 4.4 Decoder with set of inputs

- To choose other set of inputs, reset the decoder by clicking on the reset button and click on the inputs required.

4.3 Encoders

Encoders will take many inputs and give out a fewer number of outputs. Usually, an encoder with 2^n number of inputs gives 'n' number of outputs. So encoders are multiple input - multiple output devices. A 4-to-2 encoder is explained here in a simple way.

- As shown in figure 4.5, the inputs are shown in red and green balls, which represent a logic 0 and logic 1 respectively. Any combination of the inputs can be chosen by clicking on the relevant colored ball in each input line.

Inputs					Outputs	
En	A3	A2	A1	A0	Q1	Q0
1	0	0	0	1	0	0
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	1	0	0	0	1	1

Table 4.3 Truth table of 4-to-2 Encoder

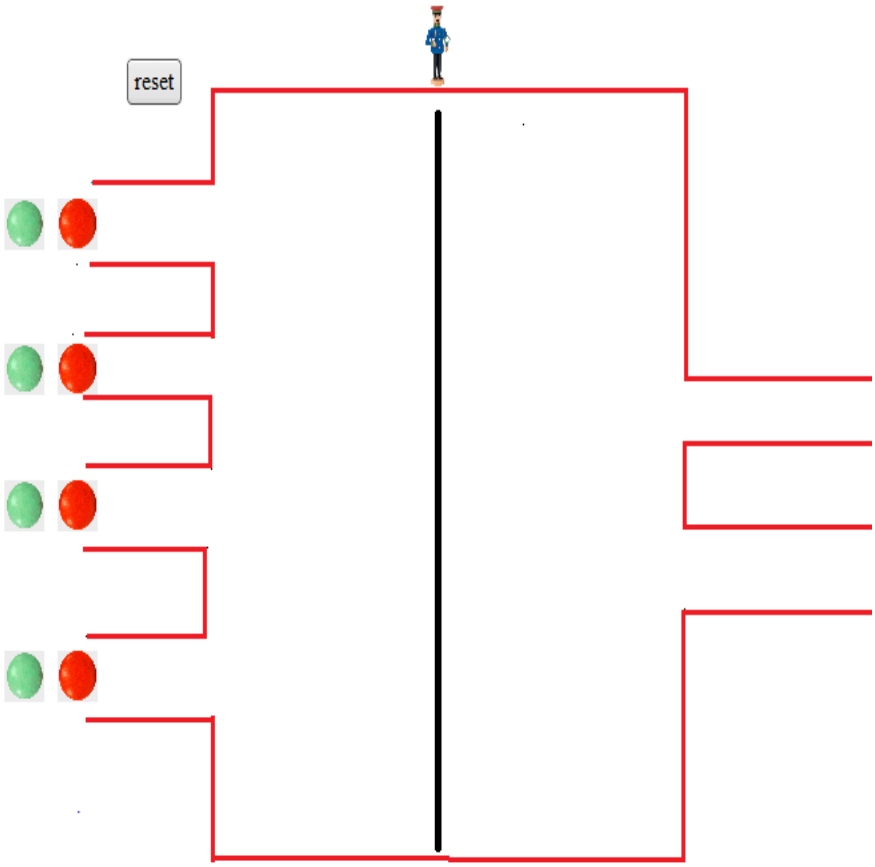


Figure 4.5 4-to-2 Encoder

- The encoder can disabled or enabled by clicking on the small human picture on the top of the diagram. The black line in the middle of the diagram is the enable signal and if this line is hidden, then the encoder is said to be enabled.
- For example, to select 1, 0, 0, and 0 as inputs, click on the green, red, red and red balls on the each input line respectively [Figure 4.7].

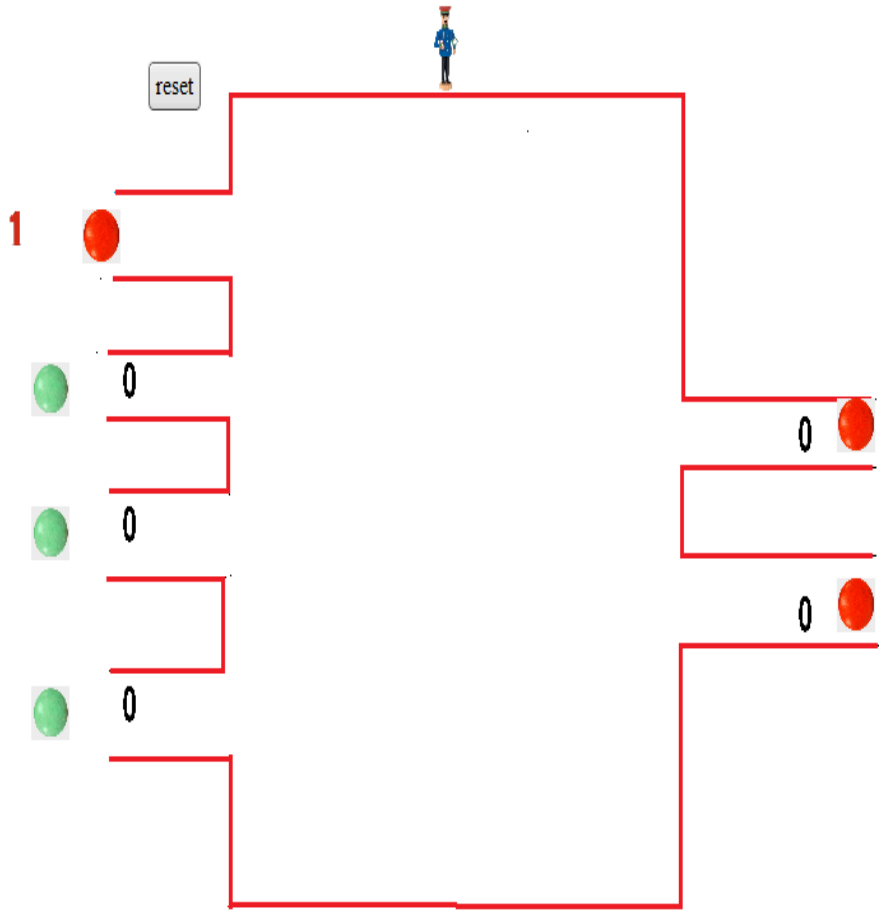


Figure 4.6 Encoder with set of inputs

- To choose other set of inputs, reset the encoder by clicking on the reset button and choose the inputs as required.

CHAPTER 5

SEQUENTIAL CIRCUITS

The sequential circuits require memory because the output of a sequential circuit depends not only on the present states of input, but also on the previous states of inputs. The digital circuits will consist of both the combinational and sequential circuits. The sequential circuits are also called as time-dependent circuits. Finite State Machines (FSM) are commonly used sequential circuits.

5.1 Finite State Machines (FSM)

Finite State Machine (FSM) as the name indicates is a logic circuit or a computational model, which has a finite number of states. The FSM can be in only one state at a given time called the current state, and the state changes only at the intervals triggered by an event and is called transition. FSMs are playing a key role in artificial intelligence and machine automation. Elevator control and Vending machines are the best examples of the FSM which we see in our daily lives. The vending machine example is explained in more detail using an interactive model. The basic principle of vending machines is to vend the item selected by the user if he or she inserts enough money.

- In figure 5.1, the money can be sent to machine by clicking on the dollar bills shown. The text box at the bottom of the diagram will show the amount of money user sent in.
- The vending options along with their prices are shown on the display board. After putting in the money, click on the price button of the item required. The machine will check if the amount of money sent in is enough to vend the selected item. If the money is enough, then the item is transferred to the outlet for the user to collect.
- The remaining money is shown in the text box and the user can choose another item or user can end the transaction by pressing the END button. When the user press END button, if there is any remaining money, it can be collected in the bottom outlet and the machine reset itself and is ready for the next user.

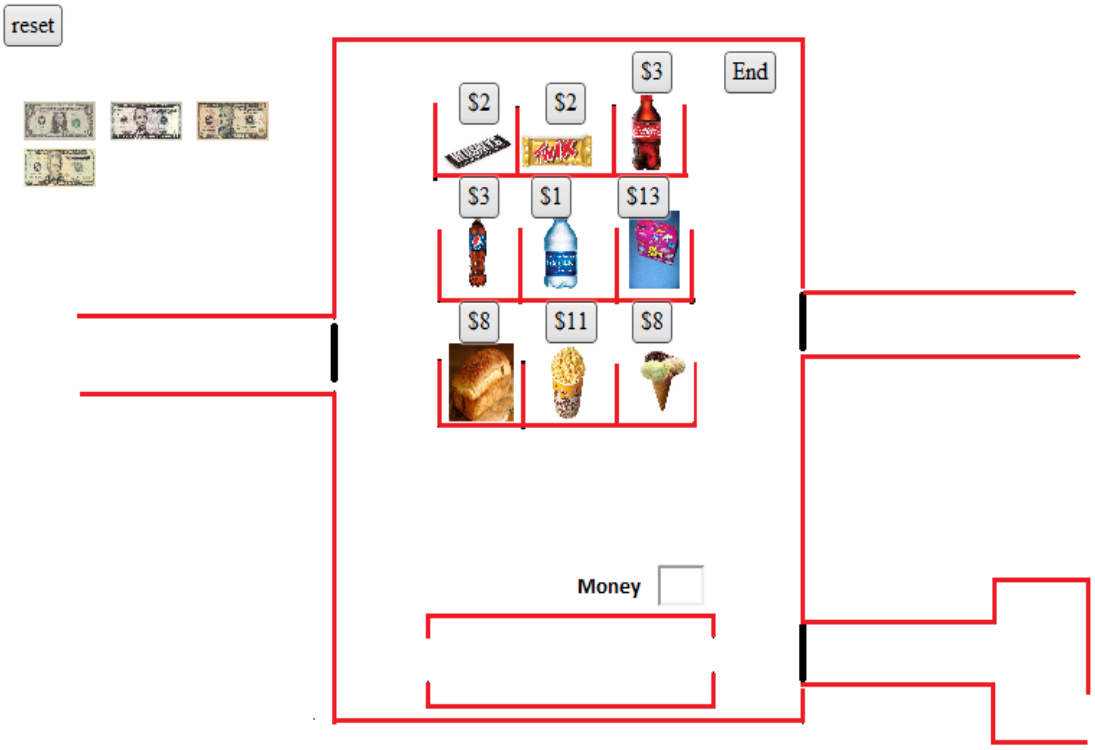


Figure 5.1 Vending machine example

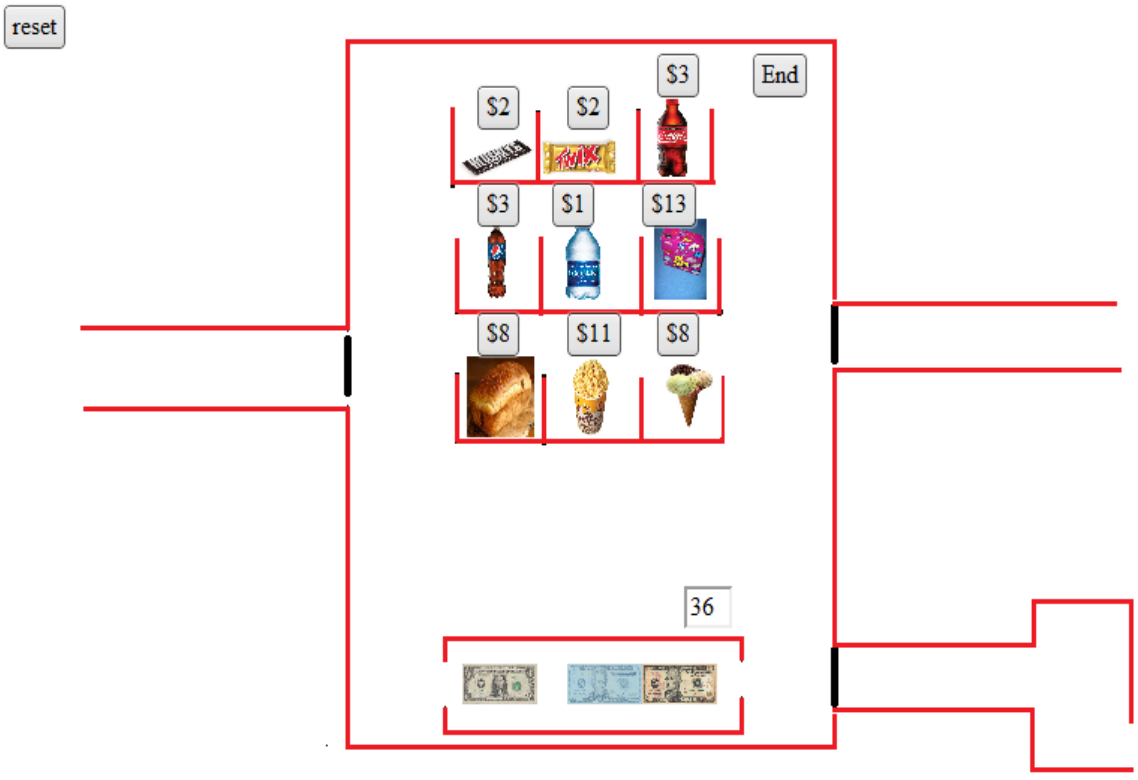


Figure 5.2 (a) Operation of Vending machine

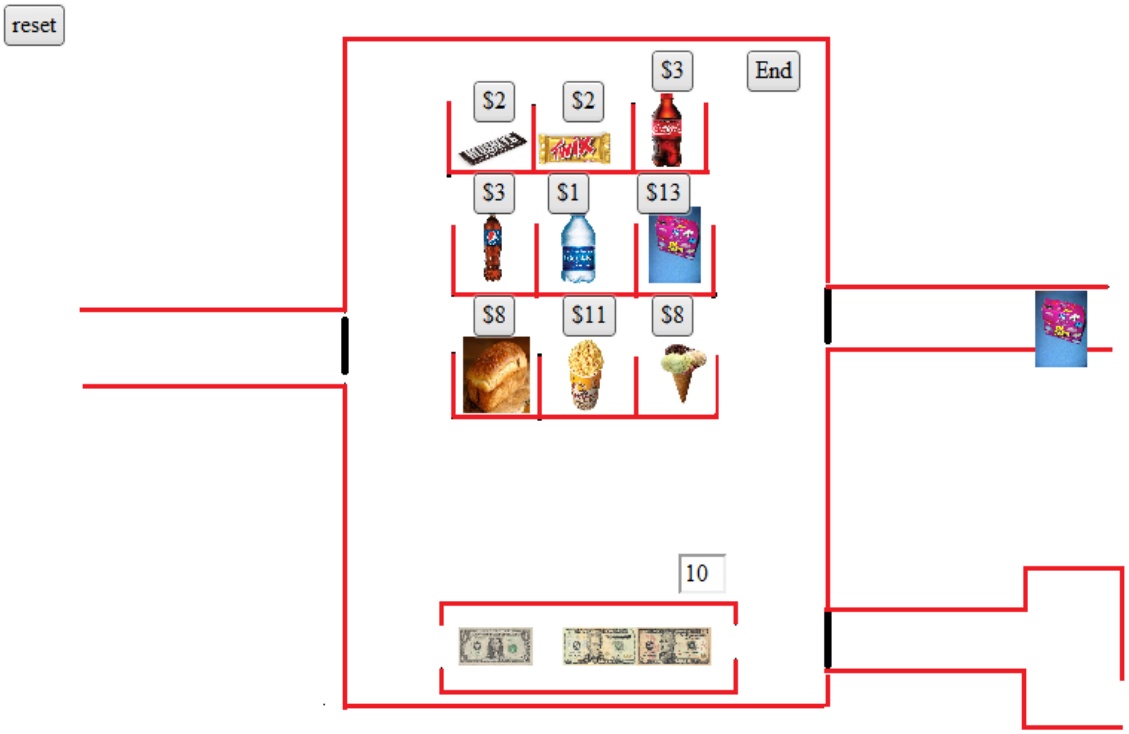


Figure 5.2 (b) Operation of Vending machine

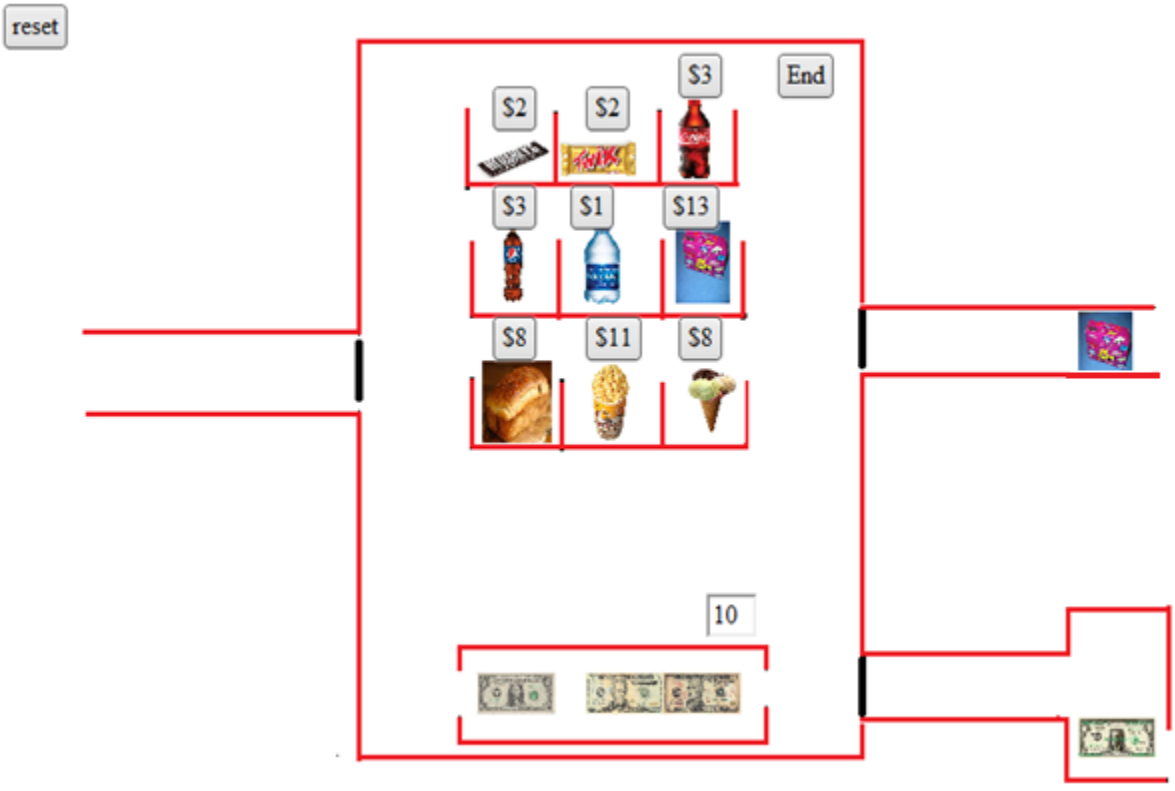


Figure 5.2 (c) Operation of Vending machine

- For example, as shown in figure 5.2 (a), user put $36 in the machine and chooses two packets of cookies ($13+$13=$26). As there is enough money, cookies are at the outlet for the user to collect. Again if the user selects a popcorn, which costs $11, as there is not enough money (only $10) [Figure 5.2 (c)], it is not vended. So when user presses the End button, the remaining $10 can be collected at the money outlet and the machine resets itself for the next user.

CHAPTER 6

POWER

As the technology is scaled down and the transistors are becoming smaller and smaller, the power consumption has become the major consideration for today's chip designers. There are many aspects leading to high power consumption and the designers are exploring new techniques in order to reduce the power consumption. The power dissipation in the circuit is of two types: Dynamic power dissipation and Static power dissipation.

The charging and discharging of the load capacitance (switching power) and short circuit current lead to the dynamic power dissipation. The dynamic power dissipation occurs mostly when the chip is doing work. Switching power constitutes for most of the dynamic power dissipation. Short circuit current occurs when both pull up and pull down networks are momentarily ON at once while the input is switching.

Static power dissipation is mainly because of the sub threshold leakage through OFF transistors, gate leakage through gate dielectric, junction leakage through source/drain diffusions and contention current. The static power dissipation occurs when the circuit is in an idle state. In order to reduce the power consumption, the designers follow different techniques such as Clock Gating, Power Gating, and Dynamic Voltage/Frequency scaling and reducing the capacitance.

6.1 Clock Gating

As mentioned previously, switching power constitutes most of the dynamic power consumption; Clock Gating effectively reduces the switching. There will be many register blocks in the circuit and all the blocks may not be used at all intervals of time. Giving the clock signal and an enable logic as inputs to the AND gate in order to turn off the clock to the idle blocks is called Clock gating. This technique is efficient because a clock has a high switching activity and gating the clock to the input registers of a block prevents registers from switching and stops all activity in the downstream logic. A simple and interactive model of clock gating with four register blocks is shown below.

- In figure 6.1, there are 4 register blocks, which are connected to the clock by an enable switch (EN). If the blocks are in black color, they are said to be OFF and when in red are said to be ON.

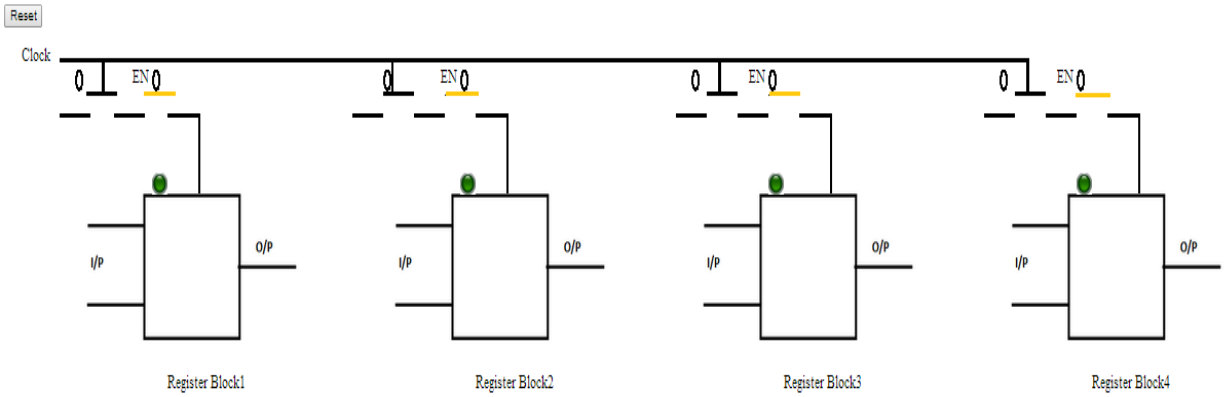


Figure 6.1 Clock Gating

- The clock can be connected by clicking on the clock line. Required blocks can be connected or disconnected to the clock by clicking on the enable switch (EN). When the user clicks on clock and enable switch, logic 0 and logic 1 will appear beside them.

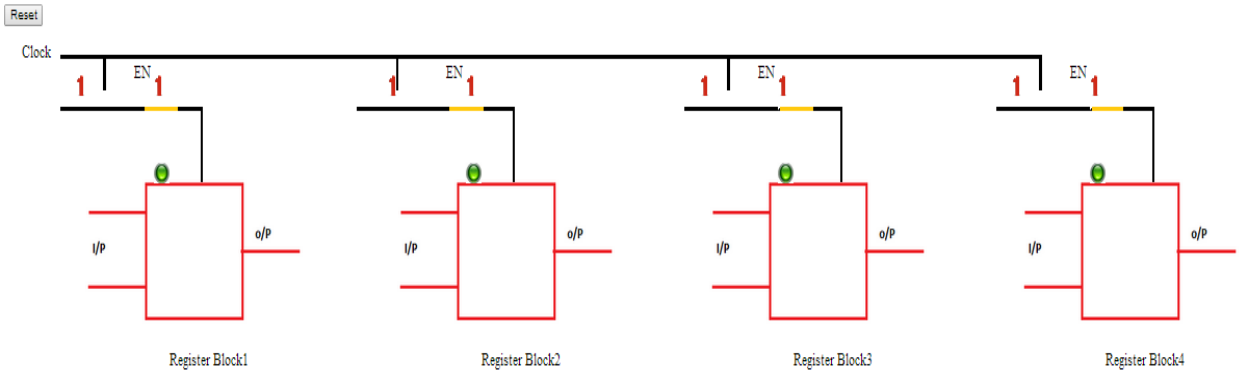


Figure 6.2 (a) All the blocks are connected

- As shown in figure 6.2(b), all the blocks are connected to the clock because the enable switch is closed for all blocks. If the user need to switch off the clock for block 2, then user need to click on the enable switch to open.
- So the user can turn off the clock to any required block by clicking on enable switch and can be connected by clicking again.

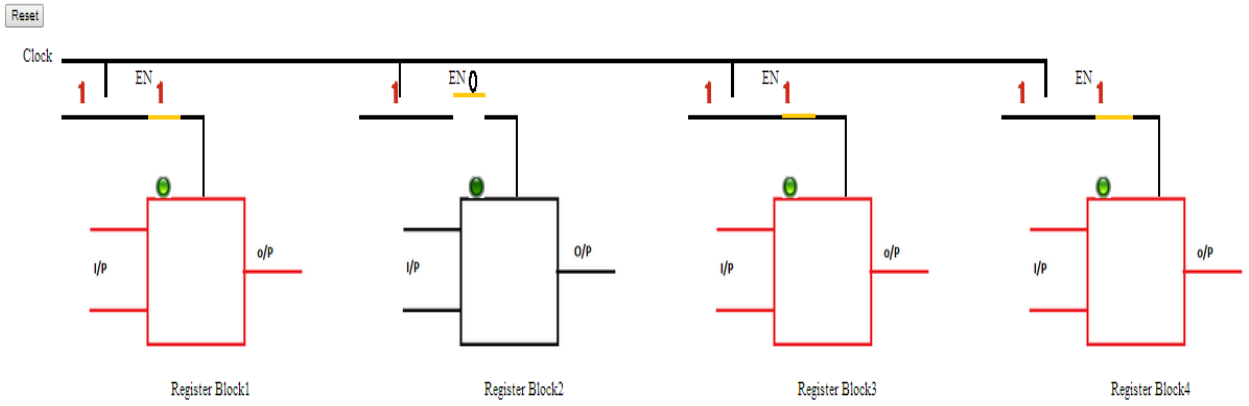


Figure 6.2 (b) Block 2 is off

6.2 Power Gating

Power Gating is the technique used to reduce the static power consumption. There will be many functional blocks in a circuit. Turning off the power supply to certain blocks that are in sleep mode is called Power Gating. In power gating technique functional blocks in the circuit receive power from a virtual voltage (V_{ddv}) rail, which instead is connected to real voltage (V_{dd}). When the functional block is active, the transistor acting as a switch is turned on, which connects the virtual voltage rail to the real voltage (V_{dd}) rail. When the block is in sleep mode, the transistor is turned off and virtual voltage terminal is floating to a voltage below 0.

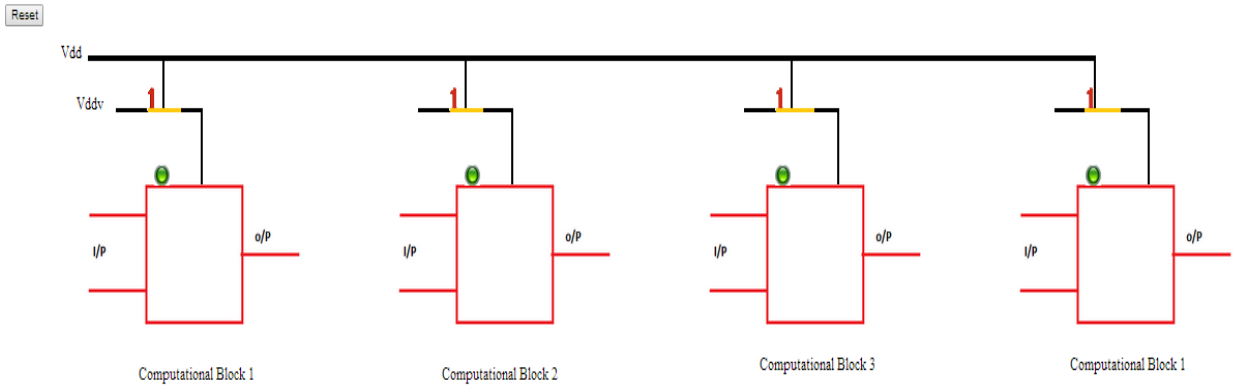


Figure 6.3 Power Gating

- In figure 6.3, the 'V_{ddv}' is connected to the real 'V_{dd}' by a transistor switch as shown. The switch can be opened or closed by clicking on it. When the switch is closed, the computational block will be on and is represented in red and when the switch is opened, the 'V_{ddv}' is disconnected from the 'V_{dd}' and the block is off represented in black.

- If one needs to turn off the power supply to the block 3, then they need to click on the switch at block 3 to get open (the block is turned black indicating it is not connected to power supply) and again to turn on the power supply, click on the opened switch so it gets closed and make the connection with the V_{dd} .

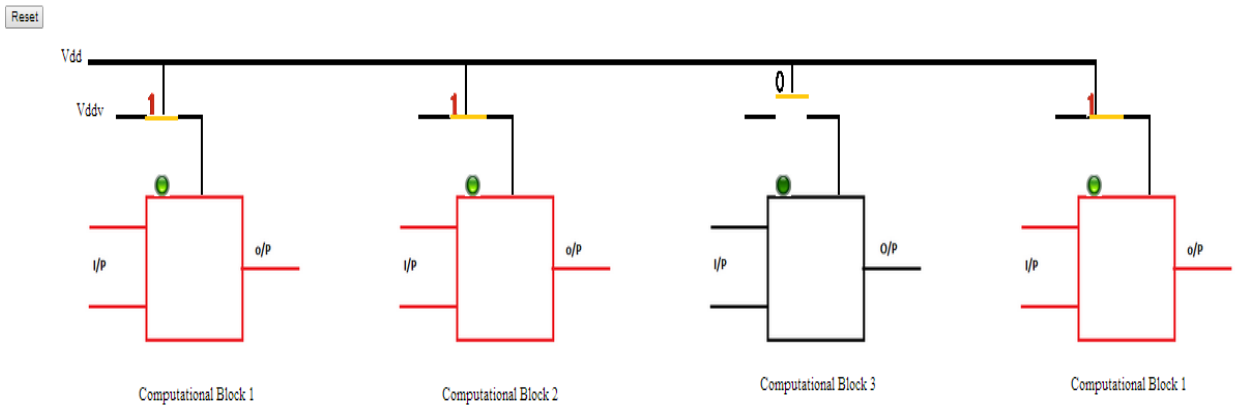


Figure 6.4 Block 3 is turned off

6.3 Activity Factor

Activity Factor is the probability that a circuit node switches from 0 to 1. This is calculated by using switching probability. This probability depends on the logic function. If the signal is a clock, then the activity factor is 1. The reduction in the activity factor reduces the switching power and hence the dynamic power dissipation. If a system clock frequency is 'f', then the switching frequency is 'αf' where α is the Activity factor and is given by

$$\alpha_i = p_i * \bar{p}_i$$

Where p_i is the probability that node '*i*' is 1 and $\bar{p}_i$ =1 - p_i is the probability that node '*i*' is 0.

- A 5x5 matrix array architecture is presented to calculate the activity factor of the circuits as big as having 5 stages and 5 branches in each stage.
- The number of stages and number of branches in each stage can be selected from the dropdown menu. Then there appears dropdown menus for selecting the logic gates (AND, OR, NAND, NOR, XOR) required to build the circuit.

- There appear two text boxes in which user needs to provide input probabilities on the left side of each gate and two text boxes for showing output probability and activity factor on the right side for each gate.
- Now, on entering the values of the input probabilities in those text boxes, output probability and the activity factor for each gate is calculated automatically and will be appearing in the text boxes provided.

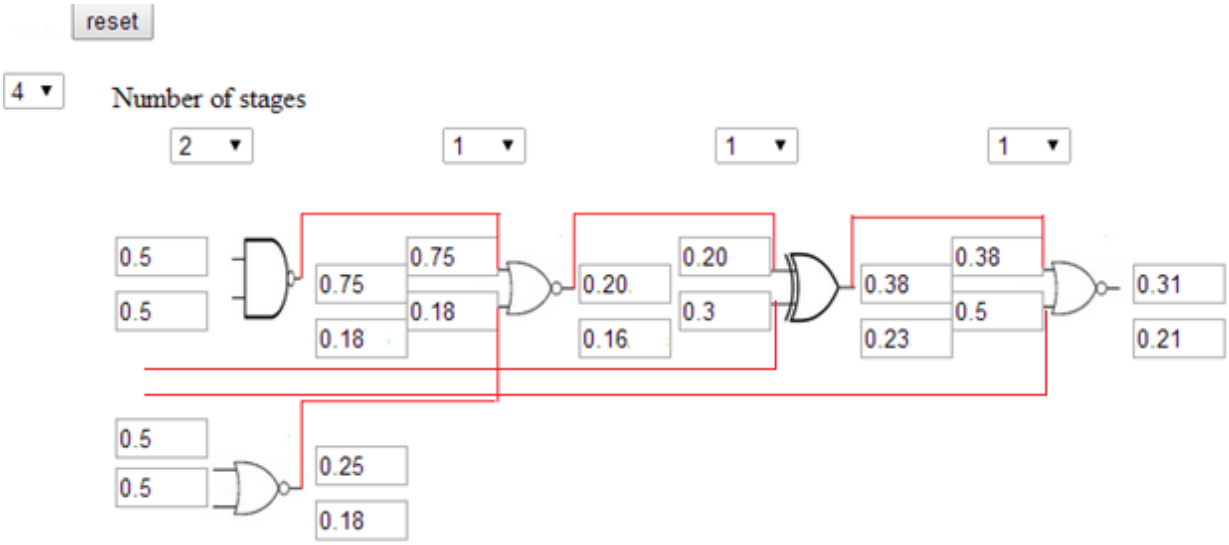


Figure 6.5 Calculating Activity factor

- As shown in figure 6.5, an example circuit is chosen. For the first gate input probabilities were given, the output probability and activity factor are calculated automatically and are shown in the top and bottom text boxes on the right side of the gate respectively. If the output probability of this gate is input to the gate in the next stage, the user just needs to copy the values.

6.4 Delay Calculation

The chip designers have many choices when they are designing a chip. For example, what kind of circuit topology is best for the given function, and how many stages of logic have to be designed to get a minimum delay. These calculations are very important for designing an effective chip in a given time; consequently, delay is one of the major design concerns in chip design. So Logical Effort helps in taking a quick decision regarding these aspects. The Logical

effort of a gate is defined as the ratio of the input capacitance of the gate to the input capacitance of an inverter that can deliver the same output current. The delay in the gate is defined by the following expression:

d=f + p where f=g * h

f=Effort Delay or Stage Effort

p= Parasitic Delay

g=Logical Effort

h= Electrical Effort

The Logical Effort depends on different gates and the number of inputs to the gates and is always 1 for an inverter. The Electrical Effort is the ratio of the output capacitance to that of the input capacitance (C_{out}/C_{in}) and also called as Fan-out. The Parasitic delay is the delay in the gate when it is driving zero load. The parasitic delay changes with the gates and the number of inputs to the gates.

- A 10x10 array architecture is presented to calculate the delay of the circuits as big as having 10 stages and 10 branches in each stage.
- The number of stages and number of branches in each stage can be selected from the drop down menu. Then there appears a dropdown menu's with the option of gates to select (NAND, NOR, XOR, NOT) in order to build the required circuit.
- The data path for calculating the delay can be selected by clicking on the gates. The gates, which are in the data path, will be turned to a red color once user clicks on them and a dropdown menu appears in order to select the number of inputs to each gate.
- There are text boxes provided on the top of the architecture for entering input and output capacitance values.
- In figure 6.6, a 3 stage circuit is selected and each stage has a different number of branches. The logic gates are selected according to the required circuit and the data path is selected by clicking on the gates. Input capacitance of 40 and output capacitance of 80 is given in the text boxes.

- Now if the user clicks on the calculate button, the circuit will calculate the different parameters such as Logical effort, Parasitic Delay, Electrical Effort, Branching factor and Delay of the selected data path and are shown in the output textboxes provided.

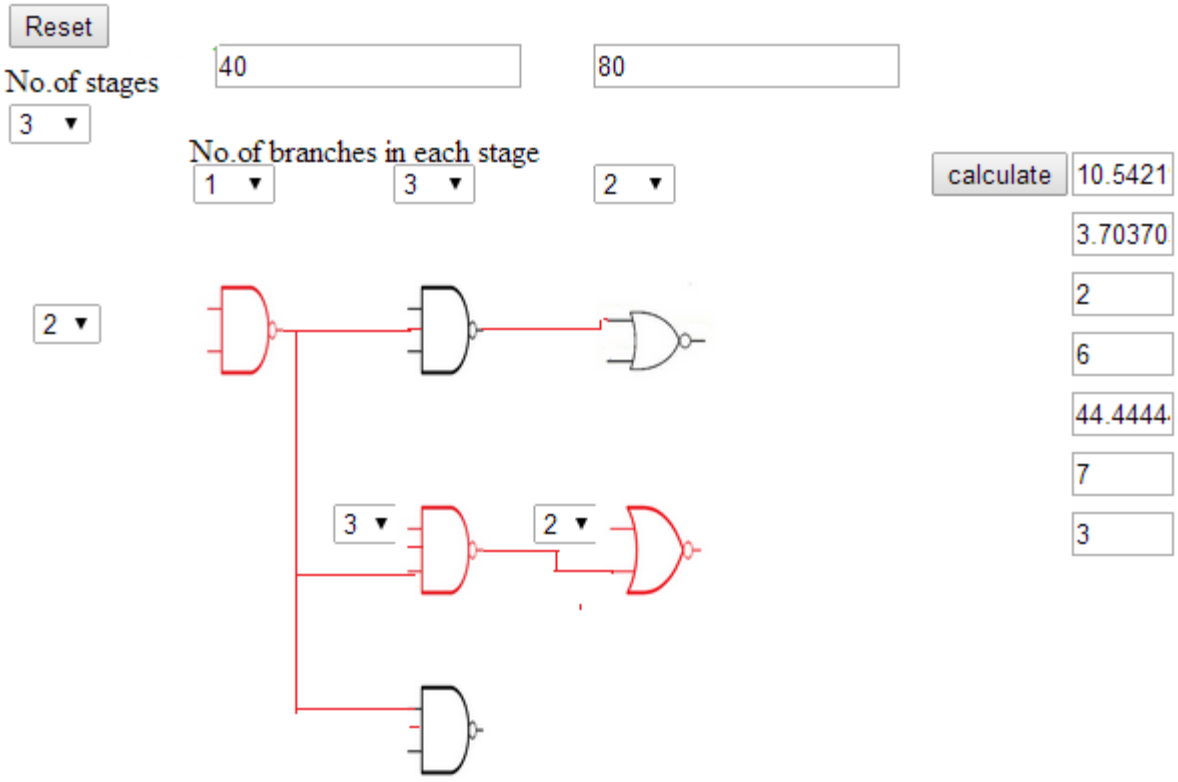


Figure 6.6 Calculating Delay

- In figure 6.6, if the user needs to find the delay of the other data path in the same circuit, then the user should click on the reset button and can proceed as said in earlier steps.

CHAPTER 7

COMPUTATIONAL UNITS

Adders are seen in almost all Arithmetic Logic units (ALU's), in filter designs, address generation and in multipliers (shift and add). Because of this extensive use of adders, the designers are trying constantly to design much faster, more efficient adder architectures from time to time. In multi bit addition, each bit sum will be dependent on the previous bit carry. So the carry of the first bit addition, influences the addition of the next bits. Correspondingly, the adders which perform this kind of addition are called Carry Propagate Adders (CPA). Adders such as Carry Ripple Adder, Carry Select Adder and Carry Look ahead Adder are examples of Carry Propagate Adders.

7.1 Carry Ripple Adder (CRA)

The Carry Ripple Adder (CRA) is a simple adder operating in the most obvious way. If 'C_i' is the carry of the bit '*i*', then 'C_i' is the carry input to the bit *i*+1. The N-stage CRA can be built by cascading the N full adders. Here a 4 bit CRA is presented in a simple and interactive way.

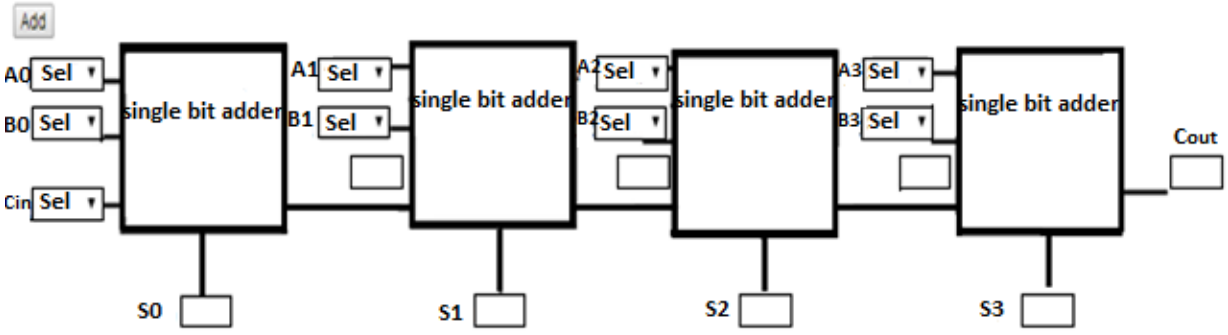


Figure 7.1 Carry Ripple Adder

- In figure 7.1, a 4 bit CRA is built using 4 single bit adders. The inputs A0-A3 and B0-B3 are 4 bits each and a single bit carry input can be selected using the drop down menu.
- All the 4 bits can be given at once or a single bit addition can be done at one time. After each single bit addition, the Add button will be disabled and enabled only after giving the inputs for the next bit addition. All the sum bits are stored in S0-S3 text boxes and the final carry bit is stored in the C_{out} box.

- In order to add the next set of inputs, the CRA has to be reset by clicking on the reset button.

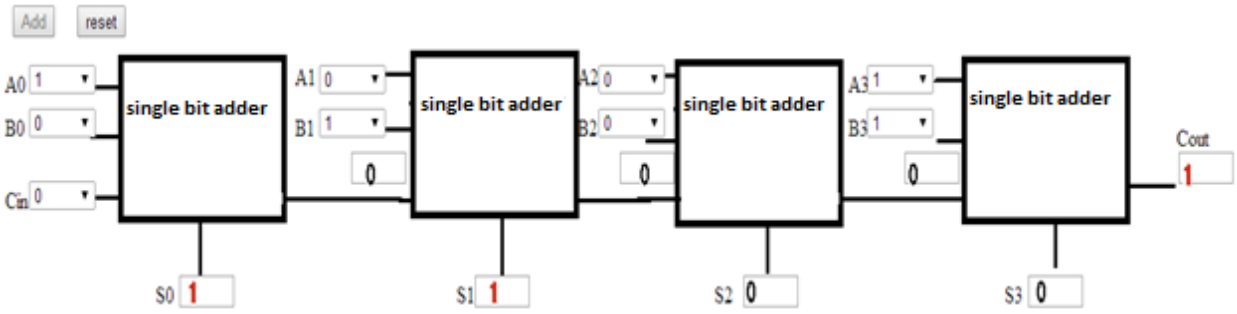


Figure 7.2(a) Carry Ripple Adder is adding all 4 bits at a time

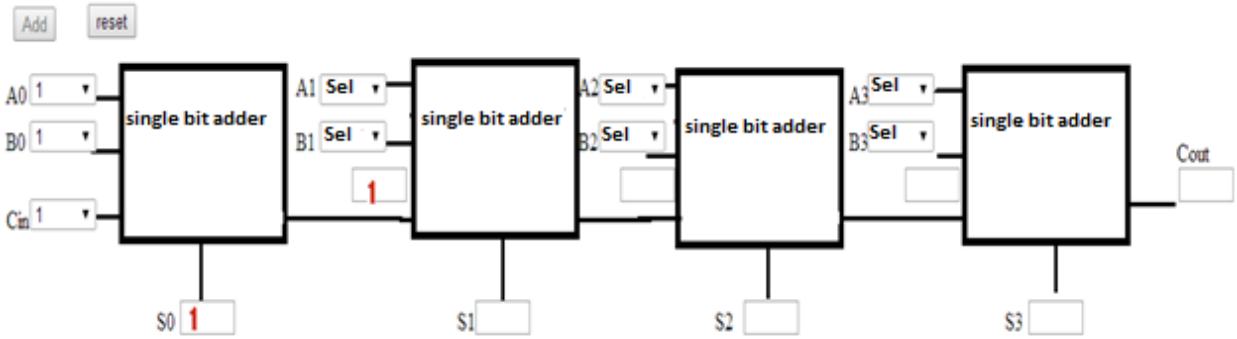


Figure 7.2(b) Carry Ripple Adder adding a single bit at a time

7.2 Carry Look ahead Adder (CLA)

Delay is the main disadvantage of the Carry Ripple Adder and Carry Look ahead Adder is designed to reduce this delay. In CLA sum and carry for each digit is calculated simultaneously. If the carry arrives within a group of digits, then it is propagated to the most significant bit and then continues with the next group. In each bit the carry propagates slowly, but in groups the carry propagates much faster than in that of CRA thereby increasing the speed. A 4 bit CLA is presented in an interactive way as shown.

- As shown in figure 7.3 (a), the inputs A0-A3 and B0-B3 and a single bit carry input can be selected from the drop down menu. The carry look ahead adder employs separate circuitry for carry generation and propagation.

- In order to add the next set of inputs, the CLA has to be reset by clicking on the reset button.

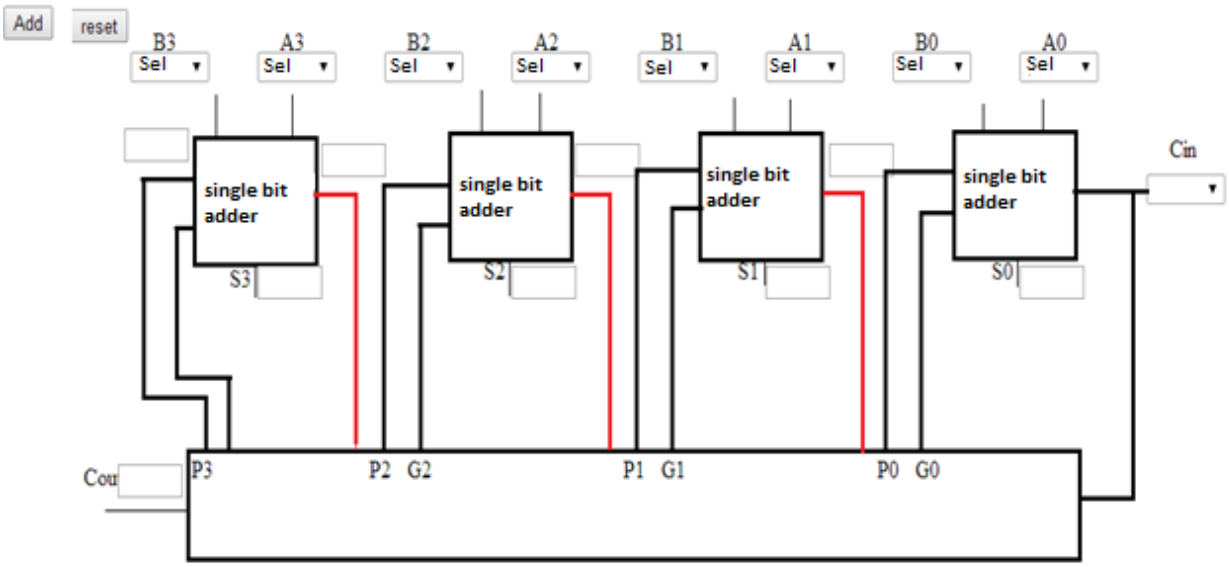


Figure 7.3(a) Carry Look ahead Adder

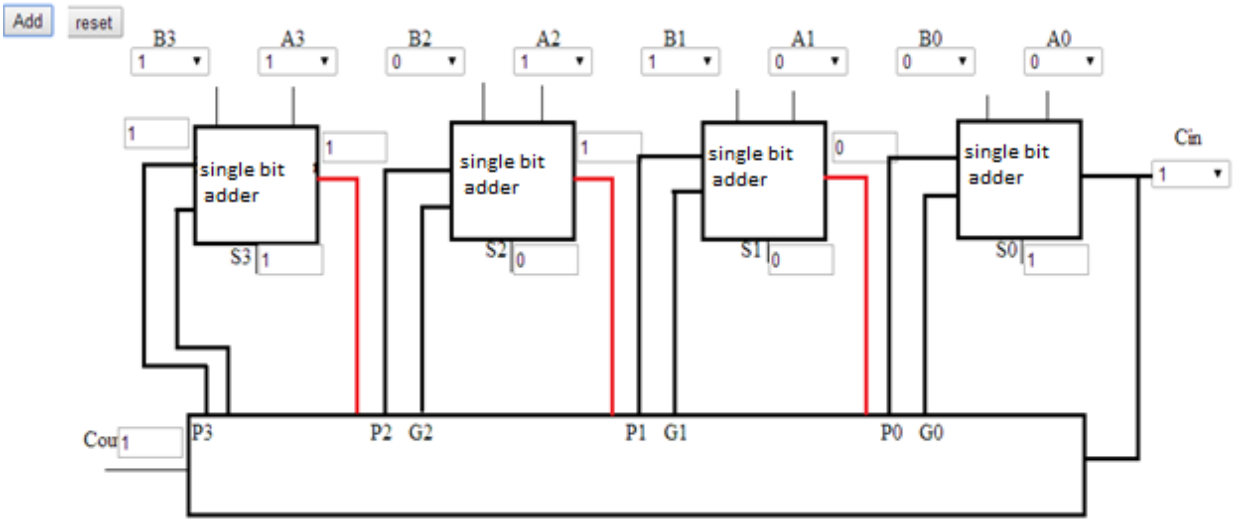


Figure 7.3(b) Carry Look ahead Adder adding all 4 bits at a time

7.3 Carry Select Adder (CSA)

In Carry Select Adder(CSA), for each bit of input two sums and carrys are calculated by considering carry input as 0 in one case and 1 in the other case. Later these sum and carry were given to the multiplexers which are triggered by the real carry input in order to select the output sum. The critical path delay in the carry select adder is less than that of the CRA and

CLA. A 2 bit Carry Select Adder built using 2 single bit full adders is presented in an interactive way.

- As shown in figure 7.4 (a), the inputs A0-A1 and B0-B1 and a single bit carry input can be selected from the drop down menu as shown.
- In figure 7.4 (b), the top full adder will add the input bits considering carry input as 0, while the bottom adder adds considering carry input as 1. Both sum and carry bits in both cases are given to a multiplexer which has real carry input as the select line.

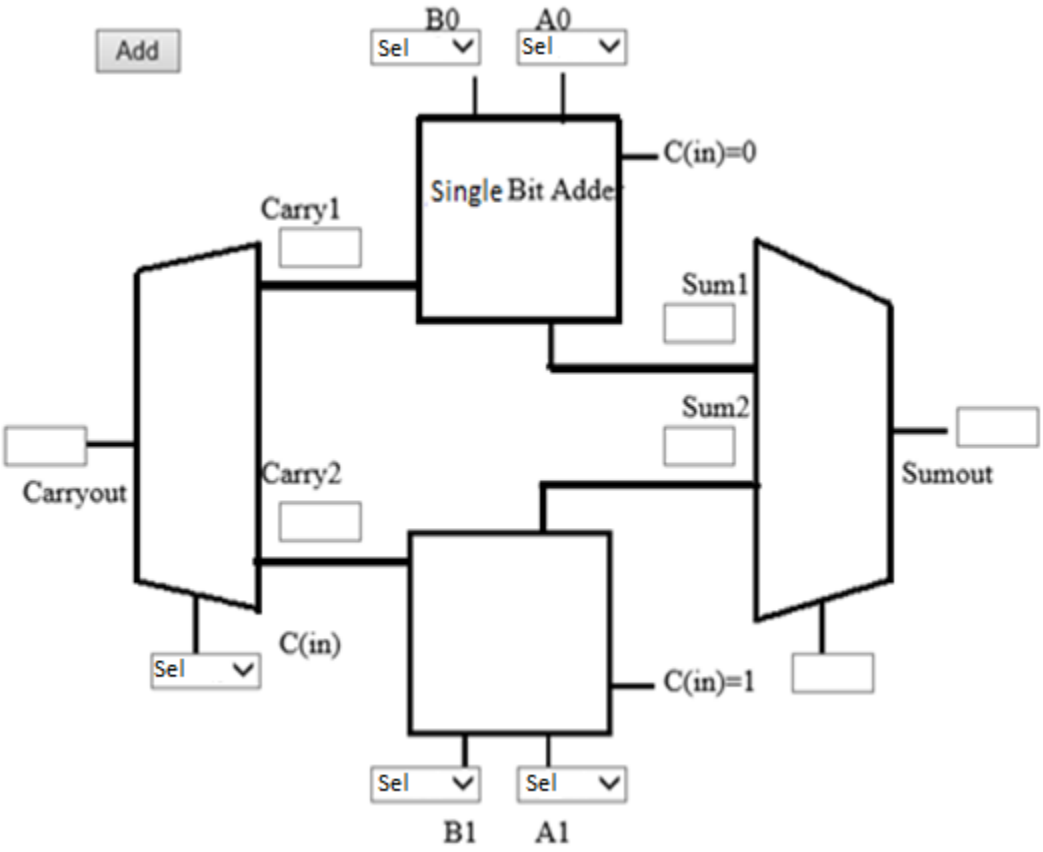


Figure 7.4(a) Carry Select Adder

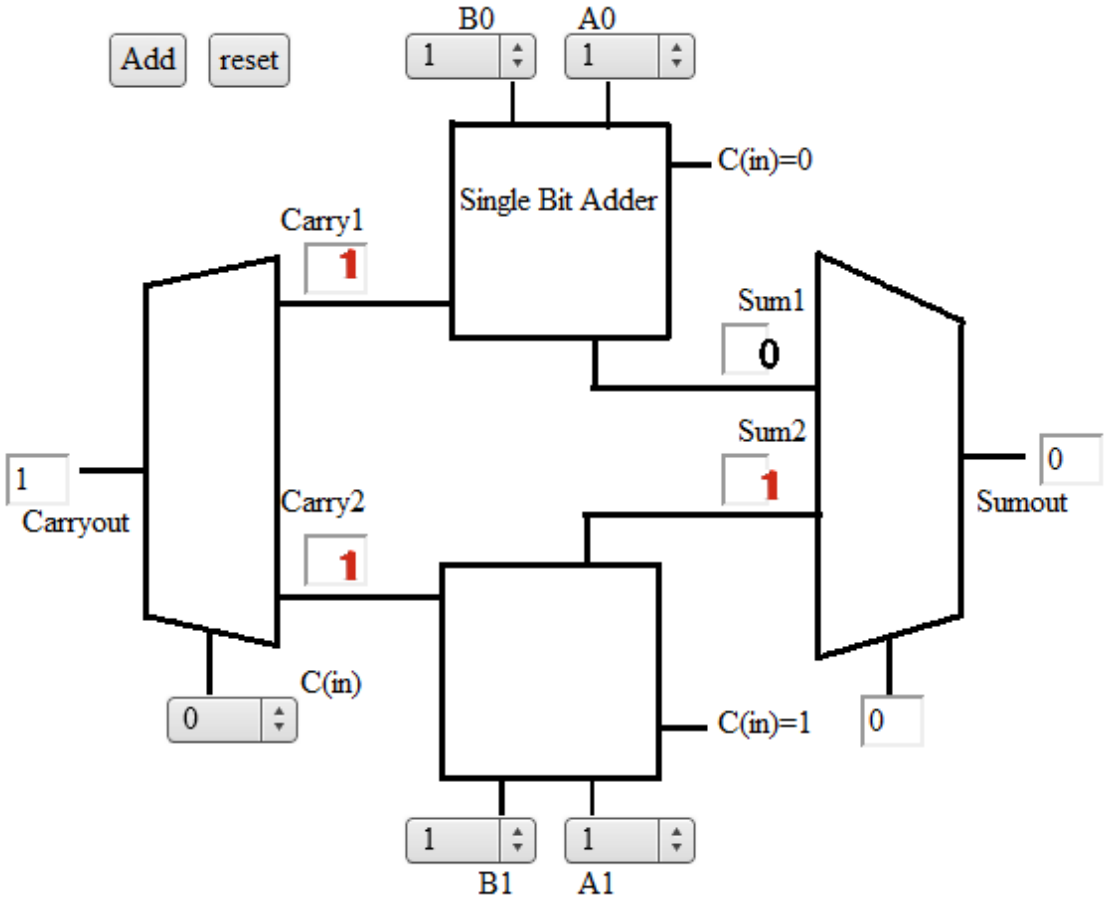


Figure 7.4(b) Carry Select Adder adds all the bits at a time

CHAPTER 8

MEMORY

Memory chips are found everywhere in personal computers, cell phones, music devices, and tablet PCs to store vast amounts of data. In general, memory architecture consists of a row decoder and a column decoder. The row decoder selects the row address and the column decoder selects the column address. The memory array consists of memory cells. Each memory cell may have one or more ports. In the memory, each port may be read only, write only, or can be both readable and writable. A memory array contains 2^k words of 2^m bits each. During the write operation, the data that has to be written is given to the data lines and the memory cell in which it has to be written is selected using the row and column decoders. In the same way, during the read operation, the cell address from which data has to be read is selected using the decoders.

8.1 Content Addressable Memory (CAM)

The Content Addressable Memory is similar to the general memory in read and write operations, but it also performs matching operations. The CAM will have a key line and any data given to this key line is matched with the data throughout the memory array and triggers the match line if found. This memory is used in high speed search applications such as artificial neural networks, and such database systems as finger print searching and DNA matching.

- In figure 8.1, an 8x8 memory array is presented with read and write operations. The row and column address can be selected using the 3-to-8 decoders. The inputs to the decoders can be given by dropdown menu as shown.
- As shown in figure 8.2, the memory cell at the selected address turns to red. Suppose if an input of "000" is given to both the row and column decoders, then the first memory cell in the first row is selected. If the input is "010" to row decoder and "011" to column decoder, then the fourth memory cell in the third row is selected.

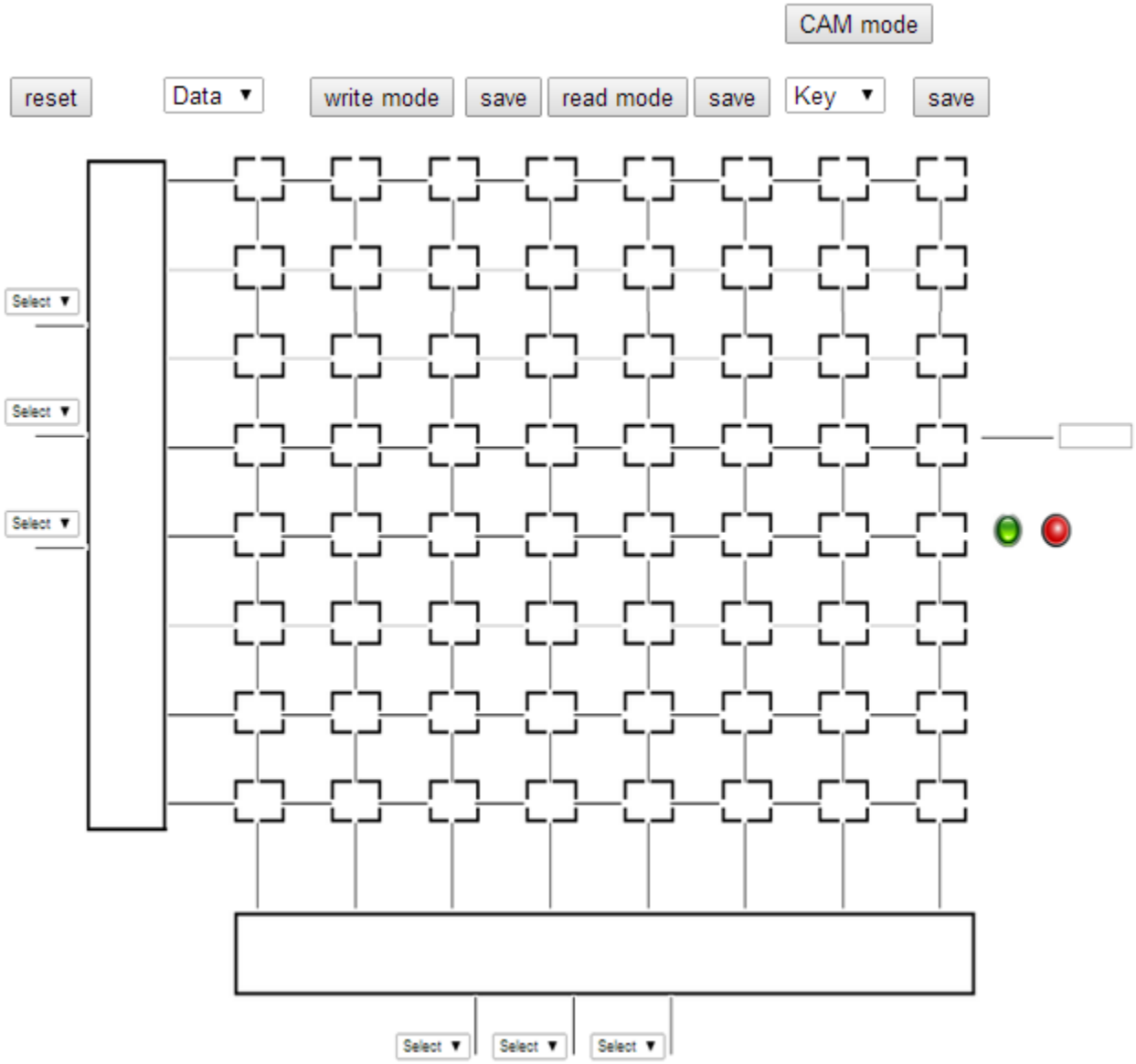


Figure 8.1 Memory Array

- If data has to be written to a memory location, click on the memory mode button, and the button gets disabled. Then select the data from the drop down menu. Then an address is selected using the decoders. Press the save button before attempting to write the next data [Figure 8.3].
- If data has to be read from a memory location, click on the read mode button, and the button gets disabled. Then select the address using decoders. The read data appear in the data output textbox. Click on the save button before attempting to read from another location [Figure 8.4].

- If any data need to be searched in the memory array, click on the CAM mode button and select the data to be searched in the key dropdown menu. The CAM mode automatically scans the memory array and if found the green light appears (hit) and if not found red light appears (miss). Click on the save button before searching for other data [Figure 8.5].

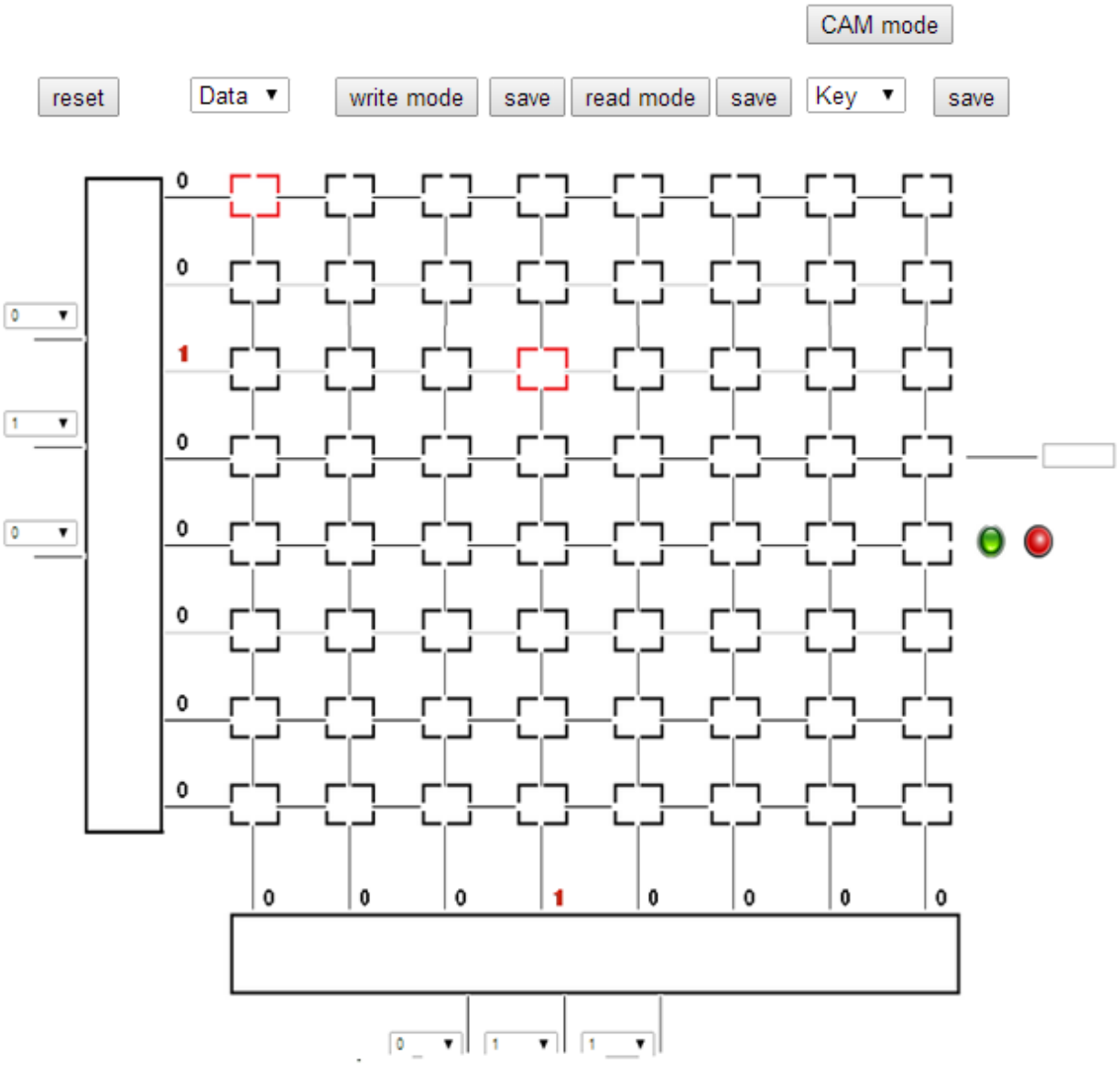


Figure 8.2(a) memory address selection

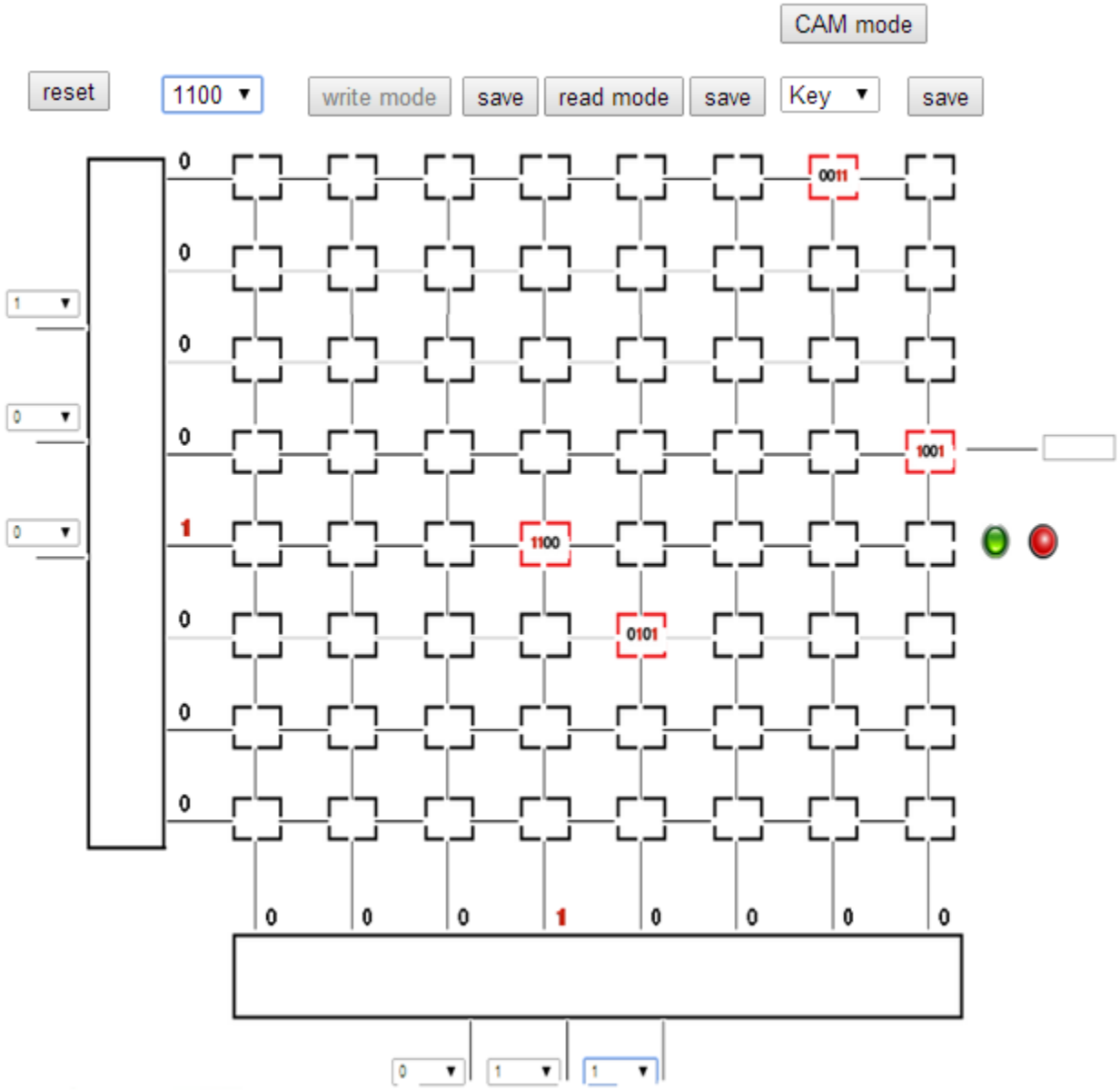


Figure 8.2(b) Memory write operation

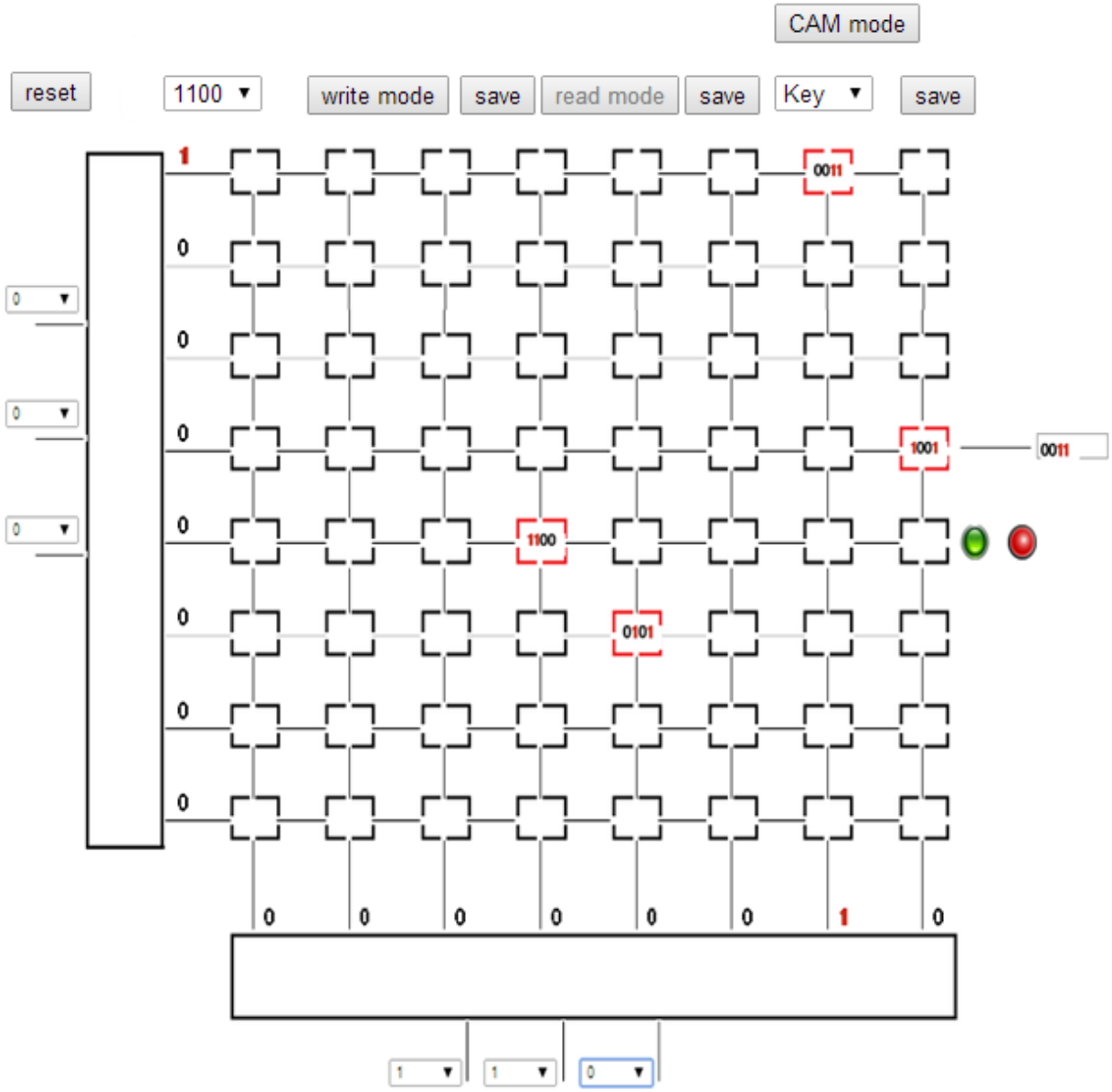


Figure 8.2(c) Memory read operation

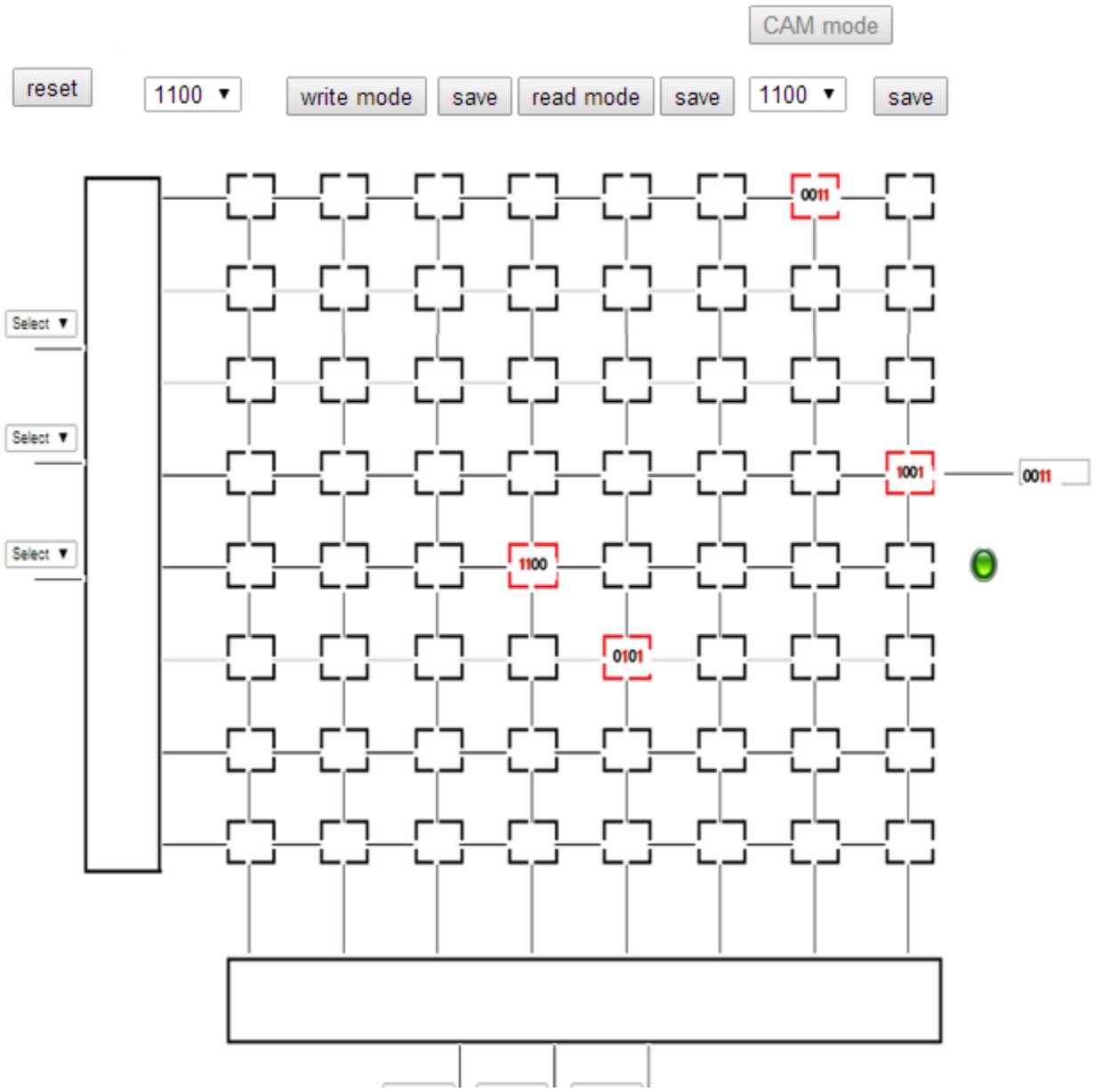


Figure 8.2(d) CAM operation

CHAPTER 9

CONCLUSIONS

In this thesis, an interactive framework is designed in which fundamental concepts of digital logic design are presented in an easy and interactive way to attract and create interest in students towards engineering fields, especially Electrical Engineering and Computer Engineering. Most of the concepts are presented with examples taken from day-to-day lives. Some of the critical design concerns such as power and performance are presented in an interactive way to make sure that students can understand these significant concepts in an easier and user-friendly way.

This work can be extended by adding more advanced concepts of digital logic design and VLSI design to the framework. Some of the real-world problems from the area of digital logic design and VLSI design can also be introduced to students in an interactive way using this framework.

REFERENCES

[1] *Book: CMOS VLSI Design, A circuits and systems perspective (Fourth edition).* Neil H. E Weste, David Money Harris.

[2] *Lectures on - VLSI Design/ Digital IC Design*. G. Mehta. EENG5540: Digital IC Design.

[3] *Book: Java Script - The complete reference (second edition).* Thomas Powell, Fritz Schneider. McGraw-Hill/Osborne © 2004. ISBN: 0072253576.

[4] *Book: foundational Java - Key Elements and Practical Programming*. David parson. Springer 2012. ISBN 978-1-4471-2478-8.

[5] *HTML Tutorial*. http://www.w3schools.com/html/default.asp.

[6] Uwe DROFENIK, Johann W. KOLAR, Peter J. van DUIJSEN and Pavol BAUER: "*New Web-Based Interactive E-Learning in Power Electronics and Electrical Machines"*. Industry Applications Conference, Sept 30, 2001. IEEE (Vol 3). PP: 1858-1865.

[7] BG Prusty, O Ho, and S Ho: "*Adaptive Tutorials using eLearning Platform for Solid Mechanics Course in Engineering".* 20th Australasian Association for Engineering Education Conference University of Adelaide, 6-9 December 2009. ISBN 1 876346 59 0 © 2009.

[8] Yumei Kang, Gengye Chen, Yunhong Cheng, and Lin Xu: "*Design and Realization of Network Teaching Platform Based on E-Learning".* 2008 International Conference on Computer Science and Software Engineering. pp 218.

[9] Zhan Chen, Israel Koren: "*Techniques for Yield Enhancement of VLSI Adders"*. International Conference on Application-Specific Array Processors, IEEE 1995.

[10] "*Circuits Lab - An online circuit simulator"*: https://www.circuitlab.com.

[11] Yuan Taur, Douglas A. Buchanan, Wei Chen, David J. Frank, Khalid E. Ismail, and Shih-Hsien Lo, George A. Sai-Halasz, Raman G. Viswanathan, Hsing-Jen C. Wann, Shalom J. Wind, and Hon-Sum Wong: **"***CMOS Scaling into the Nanometer Regime"*. Proceedings of the IEEE, vol. 85, NO. 4, APRIL 1997.

[12] Ed Grochowski, Murali Annavaram: "*Energy per Instruction Trends in Intel® Microprocessors".*

[13] Massoud Pedram: "*Power Minimization in IC Design: Principles and Applications".* F33615-95-C1627, NSF NYI.

[14] Course: *"Digital Electronics"*. IISC Bangalore, India.

[15] Gangadhara B. Prusty, Carol Russell, Robin Ford, Dror Ben-Naim, Shaowei Ho, Zora Vrcelj, Nadine Marcus, Timothy McCarthy and Tom Goldfinch, Roberto Ojeda, Anne Gardner, Tom Molyneaux, Roger Hadgraft: "*Adaptive Tutorials to target Threshold Concepts in Mechanics –a Community of Practice Approach.* Proceedings of the 2011 AAEE Conference.

[16] Charles Hacker and Renate Sitte: "*Interactive Teaching of Elementary Digital Logic Design with WinLogiLab".* IEEE transaction on education. Vol 47, No 2. May 2004.

[17] Harrison Hilgers, Simon Huynh, and Norman Lee: "*Interactive Breadboard Digital Logic Teaching Aid"*. May 1, 2013.

[18] Jonathan Allen, Christopher J. Therman: "*An Interactive Learning Environment for VLSI Design".* Proceedings of IEEE. Vol 88, No 1. January 2000.