International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 06, June 2015
Available at http://internationaljournalofresearch.org

Implementation of Graph Theory and Graph Coloring Using User Defined Simulation

Ajay Kaushik[1]& Dr. Suruchi Gautam[2]
[1] M tech, Kurukshetra University
[2]Associate Professor, Computer Science departmrnt, Delhi University

ABSTRACT –

Graph coloring is a technique used widely in many applications. It is used to divide political areas using different colors. In the proposed paper we will build a used defined finite state machine, simulate the finite state machine. We will design different states of the finite state machine in form of graph. Vertices of the graph will represent different states of the finite state machine and edges of the graph will show the state transition from one state to another state. We will actually see the transition in states by means of changing colors in the graph representing finite state machine.

CHAPTER – 1

INTRODUCTION

Graph Coloring

Graph Coloring is an assignment of colors (or any distinct marks) to the vertices of a graph. Strictly speaking, a coloring is a proper coloring if no two adjacent vertices have the same color

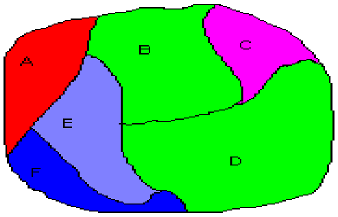

Figure 1 – Graph coloring

- Many problems can be formulated as a graph coloring problem including Time Tabling, Scheduling, Register Allocation, Channel Assignment
- A lot of research has been done in this area so much is already known about the problem space.

K-Coloring

- A k-coloring of a graph G is a mapping of V(G) onto the integers 1..k such that adjacent vertices map into different integers.

A k-coloring partitions V(G) into k disjoint subsets such that vertices from different subsets have different colors

Edge Coloring

- Given a graph G=(V,E) how can we color the edges such that edges that share a vertex do not share a color.

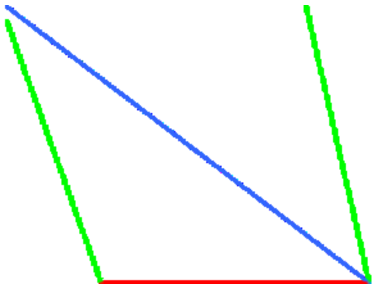

Figure 2 – Edge coloring

Existing Results

- **Vizing's theorem**
  - any graph with a maximum vertex degree of δ can be edge colored using at most $\delta + 1$ colors.

Vertex degree: number of edges entering into the edge

Line Graph

- An Edge Coloring Problem can be formulated as a Vertex Coloring Problem.
- Let L(G) be an auxiliary graph and G be the graphs which we are trying to color. L(G) contains a vertex for every edge in G. There is an edge in L(G) drawn between two vertices if their associated edges in G share a vertex.

1. Finite State System

An FSM can be represented by a state transition diagram, a directed graph whose vertices correspond to the states of the machine and whose edges correspond to the state transitions; each edge is labeled with the input and output associated with the transition. Suppose that the machine is currently in state 1. Upon input b, the machine moves to states 2 and output 1. Equivalently, an FSM can be represented by a state table with one row for each state and one column for each input symbol (Gillespie, 2008). For a combination of a present state and input symbol, the corresponding entry in the table specifies the next state and output. A state machine represents a system as a set of states, the transitions between them, along with the associated inputs and outputs. So, a state machine is a particular conceptualization of a particular sequential circuit. State machines can be used for many other things beyond logic design and computer architecture. (Hasheim, 2012) Any Circuit with memory is a Finite State Machine. Even computers can be viewed as huge FSMs. Design of FSMs Involves: Defining states, Defining transitions between states, Optimization /Minimization. (Hasheim, 2012)

CONDITION/STATE	STATE 1	STATE 2	…	STATE i	…	STATE p
CONDITION 1						
CONDITION 2		STATE 3				
….						
CONDITION j				STATE K		
….						
CONDITION m						

Figure 3 – State transition of a Finite state Machine

State Diagram

Illustrates the form and function of a state machine. Usually drawn as a bubble-and-arrow diagram.

State

A uniquely identifiable set of values measured at various points in a digital system. (Kumar, 2013)

Next State

The state to which the state machine makes the next transition, determined by the inputs present when the device is clocked.

Branch

A change from present state to next state.(Kumar, 2013)

Mealy Machine

A state machine that determines its outputs from the present state and from the inputs.

Moore Machine

A state machine that determines its outputs from the present state only. (Singh et. al., 2012)

On a well-drawn state diagram, all possible transitions will be visible, including loops back to the same state. From this diagram it can be deduced that if the present state is State 5, then the previous state was either State 4 or 5 and the next state must be either 5, 6, or 7. .(Singh et. al., 2012)

Moore and Mealy Machines

Both these machine types follow the basic characteristics of state machines, but differ in the way that outputs are produced. (Kumar, 2013)

Moore Machine

Outputs are independent of the inputs, i.e. outputs are effectively produced from within the state of the state machine. (Kumar, 2013)

Mealy Machine

Outputs can be determined by the present state alone, or by the present state and the present inputs, i.e. outputs are produced as the machine makes a transition from one state to another. (Aljeaid et. al., 2014)

SIMULATION - The component-based approach is an important design principle in software and systems engineering. In order to document, specify, validate, or verify components, various formalisms that capture behavioral aspects of component interfaces have been proposed. These formalisms capture assumptions on the inputs and their order, and guarantees on the outputs and their order. (Hasheim, 2012) For closed systems (which do not interact with the environment via inputs or outputs), a natural notion of refinement is given by the simulation preorder. For open systems, which expect inputs and provide outputs, the corresponding notion is given by the alternating simulation preorder. (Ivan et. Al., 2011) Under alternating simulation, an interface A is refined by an interface B if, after any given sequence of inputs and outputs, B accepts all inputs that A accepts, and B provides only outputs that A provides. (Hasheim, 2012) The alternating simulation preorder is a Boolean notion. Interface A either is refined by interface B, or it is not. However, there are various reasons for which the alternating simulation can fail, and one can make quantitative distinctions between these reasons. For instance, if B does not accept an input that A accepts (or provides an output that A does not provide) at every step, then B is more different from A than an interface that makes a mistake once, or at least not as often as B. (Cerny et al., 2000)

CHAPTER - 2

LITERATURE SURVEY -

WORK DONE BY DIFFERENT RESEARCHERS ON GRAPH AND FINITE STATE MACHINE IN PREVIOUS YEARS

RESEARCHER	OBJECTIVE	METHODOLOGY	RESULT
Evan (1976)	A special class of graph, the "systems net," is defined The set of all systems nets is broken down into equivalence classes,	shown to be partially ordered by a particular type of condensation, the "simplification."	every class of systems net is shown to have a unique maximal simplification, and applications of the concepts of "systems net" and

			"simplification" to automaton modeling are discussed.
Delphine Lautier, (2012),	The graph theory, which is especially relevant for the study of high-dimensional financial data. We illustrate the advantages of this method in the context of systemic risk in derivative markets, a main subject nowadays in finance.	We offer a pedagogical introduction to the use of the graph theory in finance and to some tools provided by this method. As we focus on systemic risk, we first examine correlation-based graphs in order to investigate markets integration and inter/cross-market linkages. We then restrain the analysis to a subset of these graphs, the so-called "minimum spanning trees.	We study their topological and dynamic properties and discuss the relevance of these tools as well as the robustness of the empirical results relying on them.
R.A. Proctor, (1991)	Development in micro computer software packages is examined as they impinge on the area of creative problem solving in business. I	Computer modeling	A review is provided of different computer aids to creative problem solving and an overview is given of the different approaches to management games. Many of the different kinds of management game are amenable to computerization.
Dolęga et al. (2012)	To enable the correct selection of the radiofrequency thermal ablation (RFTA) process parameters for an individual patient by applying a computer modeling of RFTA.	professional package of *FLUX3D* to generate the geometric models	The computational results show that the RFTA algorithm is effective in solving this practical problem. The computational results show that the selection of the type of electrodes used in the RFTA process is as important as the correct selection of the process parameters, i.e. voltage and frequency.
G. Southern, (1986)	Demonstrates the techniques of CAPM by means of manual simulation.	Computer simulation model	the CAPM system is presently installed on the ACT Sirius and the IBM personal

			microcomputers. It consists of a series of basic programs and data files in four modules. There is wide scope for using both versions in degree courses in mechanical and production engineering and production management.
Brian Leaned, (1993)	Provides an introduction to simulation, and discusses the use of a modern simulation environment.	simulation tools	The gap is even further reduced if the manager understands something of simulation terminology and methods. simulation is a tool which can aid managers in policy making and decision making.
McWaters et al. (1994)	Describes how simulations were executed for all combinations of eight fabrics and three contact surfaces, and presents the experimental results obtained for similar conditions and fabrics.	Computer simulation model	Proves the validity of the computer model by comparing the experimental results with those obtained by simulation. Describes how the computer model could be used to choose the optimum diameter of a fabric feeder picking roller.

CHAPTER -3

PAPER OBJECTIVES

1. Objective of this paper is to implement graph colorings with simulation of a finite state machine.

2. We want to build a user defined finite state machine, represent finite state machine in form of a graph with vertices of the graph showing different states of the finite state machine and edges of the graph showing the transition between states of a FSM.

3. We will show change in the states with changing color.

4. Trnasition from one state to other state is represented by change in graph color.

5. Entire process is simulated for time t = 50 ms or 100 ms. Time of simulation is user defined.

6. For entire length of simulation time states changes in the output graph and this state transition is represented by changing graph color.

CHAPTER – 4

METHODOLOGY

1. We want represent a finite state machine output in form of a graph

2. Vertices of the graph will show states of a FSM and edges will show state transition between states.
3. We will build a simulation interface for sine wave with appropriate multiplexer and integrator using simulink as a tool.
4. Obtaining the curve for sine wave corresponding to random values as input.
5. Varying parameters of simulator on the go.
6. Generating a chart as output of the integrator
7. Designing a state flow structure in the chart using 3 different states.
8. Designing the tradeoff between 2 basic parameters in the chart.
9. Analyzing the output of finite state machine
10. Output showed when position symbol becomes positive output does not go to 1.
11. This process is completed using a graph structure.
12. If we focus on the state flow chart during simulation we will clearly see that state changes from initial to final using vertices of a graph and vertices changing color according to state transition.
13. At a given instant of time no 2 states will have the same color.
14. Output is obtained and a graph simulated output is analyzed.

CHAPTER – 5

RESULT AND CONCLUSION

1. Here we have build a model for graphical representation of a FSM. Initial structure is build to feed as input in state flow chart . Structure is shown in figure 4

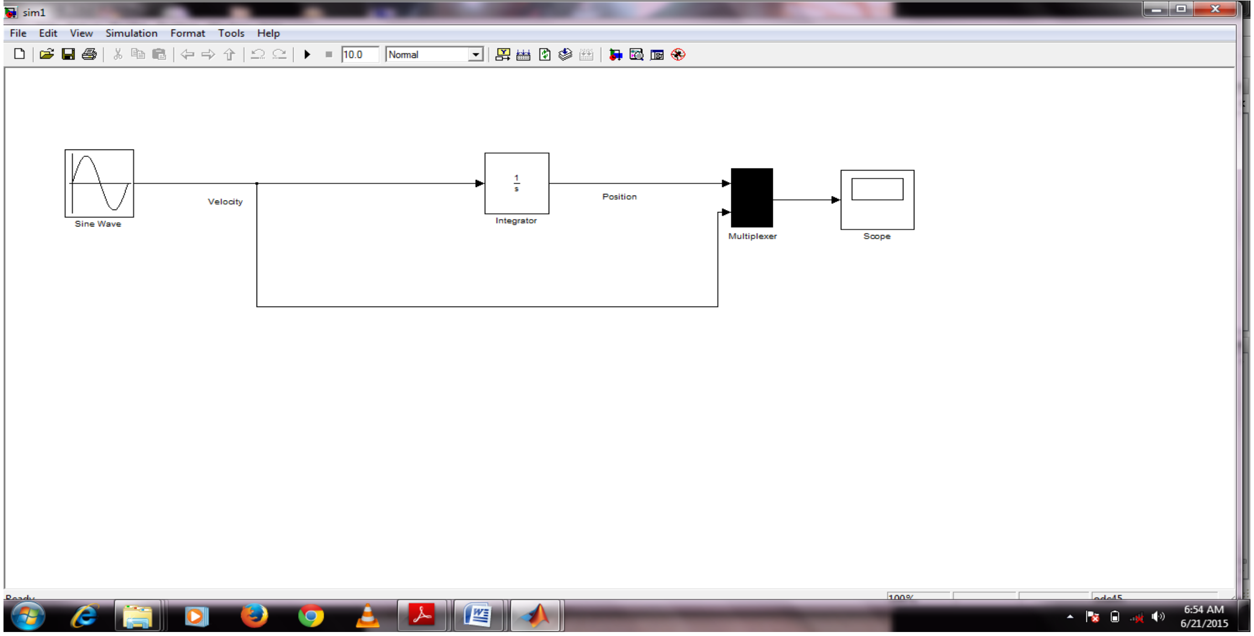

Figure 4 – Initial Structure of a FSM

2. When we run this model for time t = 10 units we get the output shown in figure 3

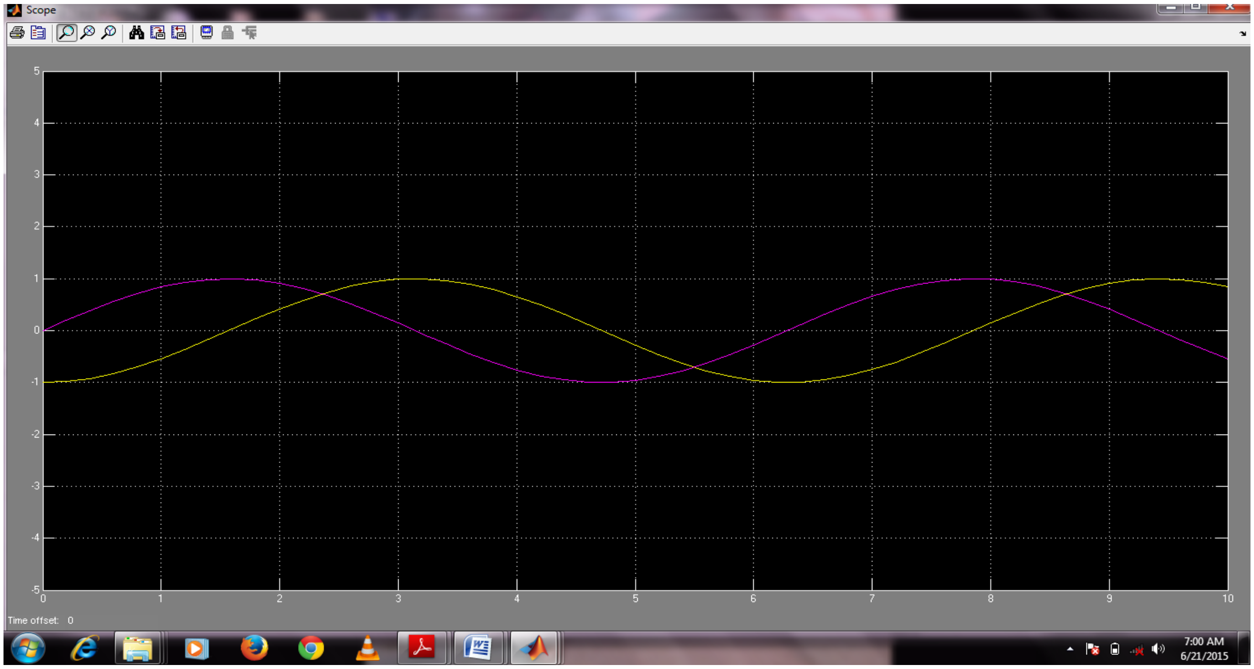

Figure 5 – Initial finite state output which

3. Here yellow curve = Velocity sine wave
4. pink curve = Position waveform (Integral form of velocity sine wave).
5. State flow chart is build in form of a graph . Structures are shown in figure 6 and 7.

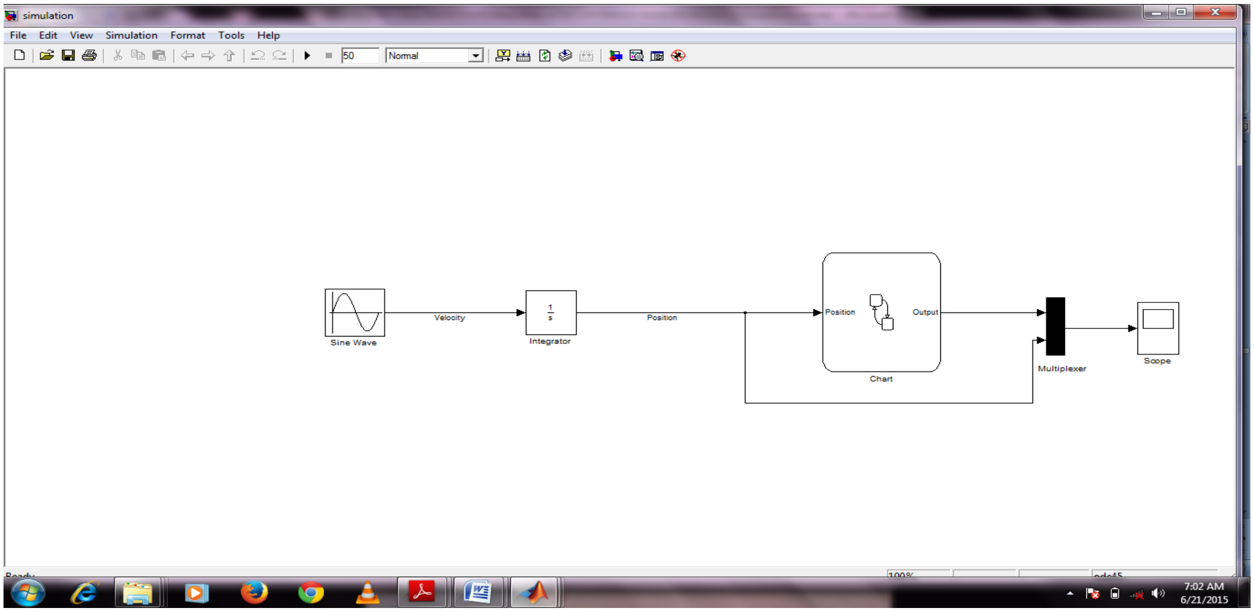

Figure 6 – State flow structure containing state flow chart

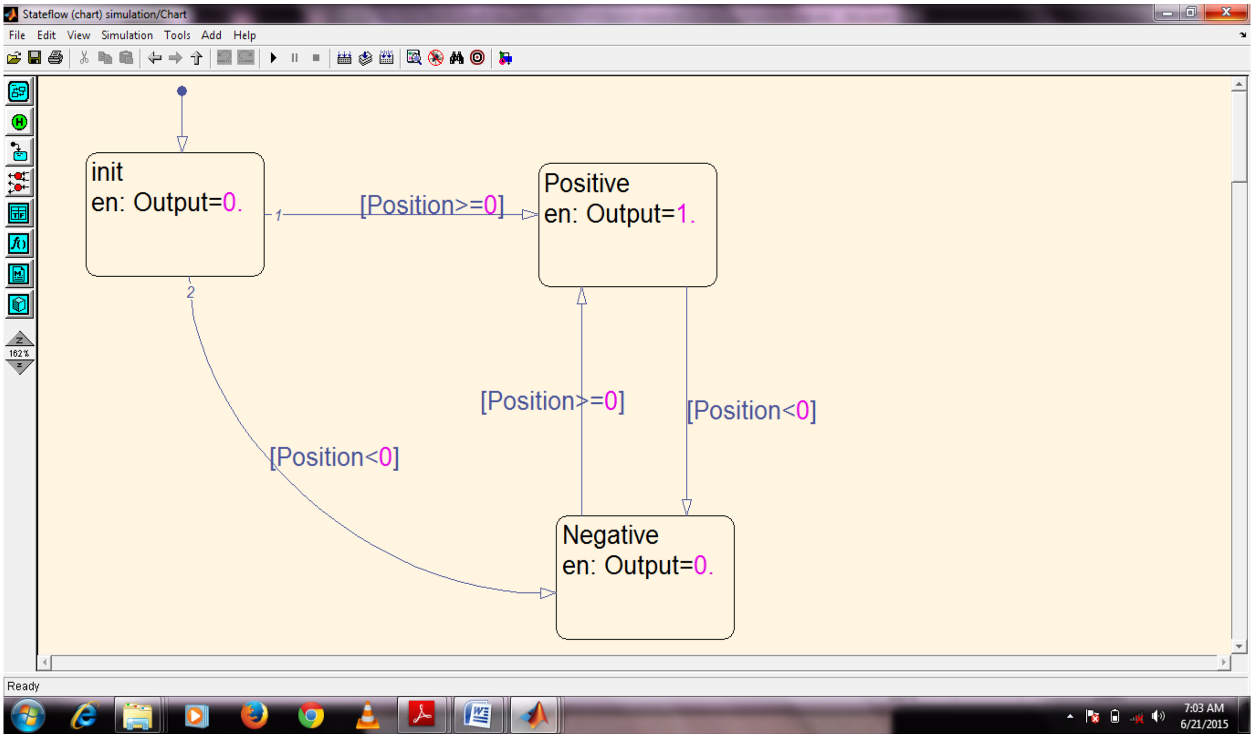

Figure 7 – state flow chart in graphical form

6. Now we will simulate the structure . states of the FSM will change states with changing colors of the vertices. This is shown in figure 8, 9 and 10

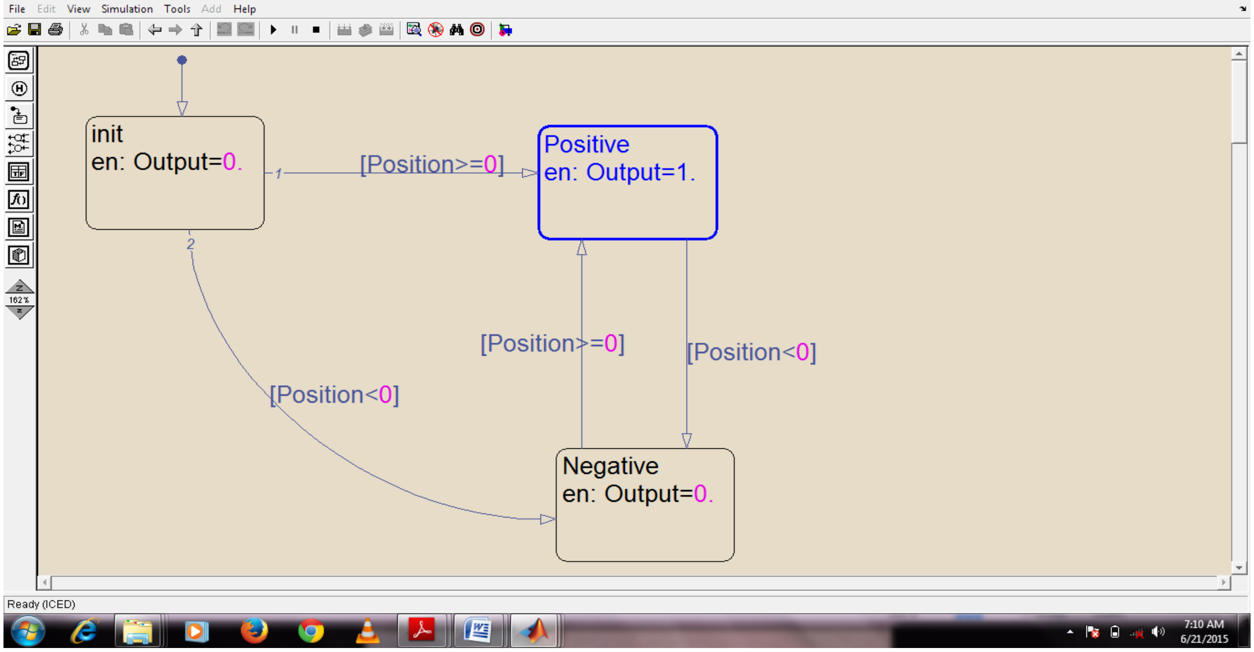

Figure 8 – state transition with changing color of vertices. No 2 states in control have the same color.

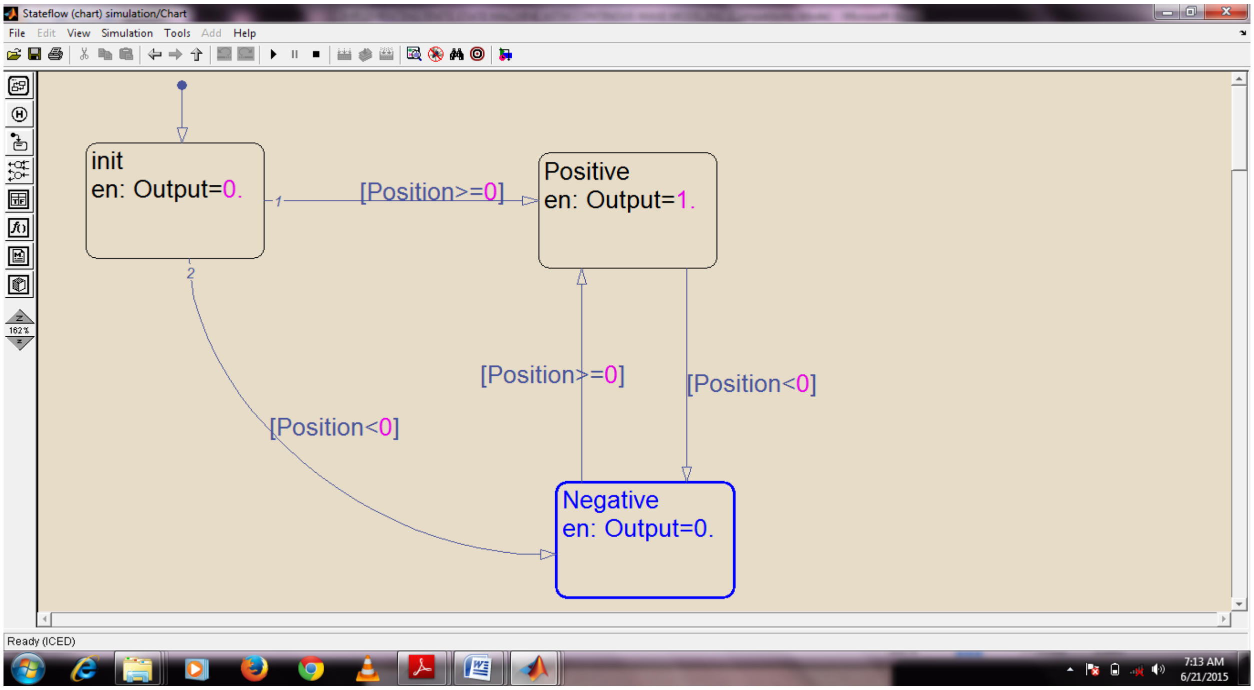

Figure 9 – state transition in a graph

CHAPTER -6
REFERENCES

[1.] Aljeaid et al., 2014, Modelling and Simulation of a Biometric Identity-Based Cryptography, International Journal of Advanced Research in Artificial Intelligence, Vol. 3, No.10.

[2.] Amr Hasheim , Simulation of Radial Variation of Neutral Atoms on Edge Plasma of Small Size Divertor Tokamak, Bekheit, Journal of Modern Physics, 2012, 3, 145-150 February 2012

[3.] Brian Leaned, (1993) "A Guide to Computer-based Simulation", Industrial Management & Data Systems, Vol. 93 Iss: 4, pp.8 – 10

[4.] Cerny,(2009) "Comparison between continuous and discrete frequency domain solution methods for structural steady state calculations with the finite element method", COMPEL - The international journal for computation and mathematics in electrical and electronic engineering, Vol. 28 Iss: 4, pp.1068 – 1080

[5.] Dagmara M. Dołęga, Jerzy Barglik, (2012) "Computer modeling and simulation of radiofrequency thermal ablation", COMPEL - The international journal for computation and mathematics in electrical and electronic engineering, Vol. 31 Iss: 4, pp.1087 – 1095

[6.] G. Southern, (1986) "Teaching the Techniques of Computer Aided Production Management and Simulation Modelling", International Journal of Operations & Production Management, Vol. 6 Iss: 1, pp.15 – 24

[7.] Gillespie, Implementing the Stochastic Simulation Algorithm, Journal of Statistical Software, April 2008, Volume 25, Issue 12

[8.] Haslinda Mohamed Kamar, Computerized Simulation of Automotive Air-Conditioning System: A Parametric Study, IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 1, January 2013

[9.] Ivan et. Al., 2011, Collaborative Systems – Finite State Machines, Informatica Economică, vol. 15, no. 2.

[10.] J.Horta-Rangel (2008) "Computer simulation of a pressure-volume-temperature process", International Journal of Numerical Methods for Heat & Fluid Flow, Vol. 18 Iss: 1, pp.24 – 35

[11.] S.D. McWaters, T.G. Clapp, J.W. Eischen, (1994) "Automated Apparel Processing: Computer Simulation of Fabric Deformation for the Design of Equipment", International Journal of Clothing Science and Technology, Vol. 6 Iss: 5, pp.30 – 38

[12.] R.A. Proctor, (1991) "Can Computers Simulate Managerial Creativity?", Leadership & Organization Development Journal, Vol. 12 Iss: 4, pp.13 – 16

[13.] Singh et. al.,2012, Finite State Testing and Syntax Testing, International Journal of Computers & Technology, Volume 3. No. 1

[14.] Stewart Robinson, (1993) "The Application of Computer Simulation in Manufacturing", Integrated Manufacturing Systems, Vol. 4 Iss: 4, pp.18 – 23