Diyala Journal of Engineering Sciences

ISSN 1999-8716

Printed in Iraq

Vol. 04, No. 02 , pp. 141-149 , December 2011

DESIGN OF A SOFTWARE SYSTEM FOR FINITE STATE MACHINE (FSM)

Hussein Abdulameer Abdulkadhim Assist Lecturer E-mail: h_a_meer@yahoo.com

Taqwa Flayyih Hasan Assist Lecturer E-mail: tota75m@yahoo.com

Computer and Software Eng. Dept./College of Engineering/Diyala University

(Received:11/5/2010 ; Accepted:19/9/2010)

Abstract:- There are several ways used to represent the circuits and system operations for design and analyses. Finite state machine (FSM) is one of the ways that used for representing the operations of many circuits and systems in electronic engineering, computer engineering ,..etc by graph. Finite state machine is very simple machine in design. It consists of sets of input symbols, output symbols and states that required to design it. Also, in fact, the function of input symbols and output symbols with the present state to give the next state must be found. In this paper, a software simulator is implemented by using Visual Basic programming language to simulate the Finite State machine (design and operation). This simulation represents general examples. However, the software may be used to learn the student concepts of FSM and its operations.

Keywords: Design, Software, System, Finite State Machine (FSM).

1. INTRODUCTION

The concept of Finite state machine is one application of directed graph that is more useful than non directed graph, the directed graph denoted as "D" consists of two things:-

1-Aset V whose elements are called vertices, point or node.

2-Aset A of ordered pairs of vertices called arcs.

We may view a digital computer as a machine which is in a certain "internal state" at any given moment. The computer "read" an input symbol and then "print" an output symbol and change its "state". The output symbol depends solely upon the input symbol and the internal state of machine and internal state of the machine depends solely upon the preceding state of the machine and the preceding input symbol. The number of states input symbols and output symbols are assumed to be finite. We can define finite state machine and we can also be used to describe the effect of certain computer programs and other "non hardware" system.[(1)(2)]

2- FSM THEORY

FSM basically consists of states and state-in actions. Researchers[(3)] have divided FSM models into two types according to number of output states and output predictability. These types are called non-deterministic FSM, and deterministic FSM, accordingly. In addition, there are other classifications such as the dependence of output states. The direction of state-transition routes is also a distinguishing property for a FSM. FSM optimization to determine the minimum number of states that have the same function is an active research subject[(3)(4)].

A finite state machine (FSM) is a model or behavior composed of finite number of state, transitions between those state and actions. Finite state machine is abstract model of

machine with a primitive internal memory. The concept of FSM with an example logic function that depends on its history of inputs to determine its output so the complete process of transforming a specification of function thought a variety of equivalent representation, resulting in actual implementation or gates and flip-flops.

In general, FSM is denoted as (M) and it consists of five elements:-

1-A finite set A of input symbols.
2-A finite set S of internal state.
3-A finite set Z of output symbols.
4-A next state function F from S X A ⟶ S.
5-An output function g from S X A ⟶ Z.

Now we can address the procedure of finite state machine design as follows[1][2]:

* First step is to understand the problem, FSM is often described in terminal of English-language specification of its behavior it is important that you interpret this description in an unambiguous meter For FSM try some input sequences to be sure you understand the condition under which the various output generated.
* Second step obtains an abstract representation of FSM. Once you understand the problem .You must place it in a form that is easy to manipulate by the procedures for implementing the FSM. A state diagram is one algorithmic state machine and specification in hardware description languages [1][2]. State diagrams provide a graphical approach to the design of an FSM [5].
* Third step performs state minimization step deriving the abstract representation often results in a description that has too many states path through the state machine can be eliminated because of their output/input behaviors is duplicated by other functionally equivalent path this is a new step which is not needed in the simpler counter design process.
* Fourth step performs state assignment in counter the state and output were identical and did not need to worry about encoding a particular state output are derived from the bit stored in state flip-flops (plus the input).
* Fifth step is to choose flip-flops types for implementing the FSM state, and this is identical to the decision in the counter design procedure[1] .

Summarily, The Finite State Machine (FSM) is a model of behavior composed of a _nite number of states, transitions between those states, and optionally actions. The transitions between the states are managed by the transition function depending on then input symbol (event) [6].

3- THE MAIN WINDOW OF THE SOFTWARE

The main window of the FSM software is shown in Figure (1). It contains two parts "Enter data part and display part":

1- ***Part one***: input FSM table; this part consists of edit boxes for enter data; and buttons for processing as shown in the figure above. These are as below:

i) Edit boxes:

a)"Enter number of state in FSM": this edit box to enter the number of state such as (1,2,3,…). In this software, the number of state is limits from 1 to 5 states that is if the user enter number 3, a massage box for correct enter will appear as shown in figure (2). If the user enter an character or zero, an massage box error for wrong enter will appear as shown in figure(3) .

b)"*present state*": this edit box designed to enter the present state.

c)"*input*": this edit box designed to enter the input symbol such as q0,q1,….etc.

d)"*output*": this edit box designed to enter the output symbol.

e)"*next state*": this edit box designed to enter the next state such as q0,q1,….etc.

ii) Buttons:

a)"*save current state*": this button designed to save the raw of data the entered in the edit boxes explained above. when the user press on this button, the data will save in matrix inside the software as raw.

b)"*enter next state*": this button designed to enter the next raw of data explained above.

c)"*view FSM*": this button designed to display the data that entered and saved in matrix. (see figure (4))

d)"*draw FSM*": this button designed to display the FSM diagraph depends on the data entered.(see figure (5))

e)"*draw all FSM*": this button designed to display some examples saved in the software for one state , two states, ….,five states.

f) "*Exit*": this button designed to exit from the software when the user wants to exit.

g) "Clear All": this button deigned to clear all edit boxes and screens.

2- ***Part two***: "display part": this part divided to two screens:

a) The first screen is design to display the FSM state table saved in the inside matrix.

b) The second screen is to display the FSM state diagraph.

However, the flowchart of the process is shown in the figure (6).

4- EXPERIMENTS AND RESULTS

Example one: [7] if the user enter the number of state = 1, the input symbols ={a,b},and the output symbols={x,y}.

The state table for the function of the next state is shown in table (1).

The state diagram for one state is shown in figure (7). If the user apply the above example on the software then the results will shown as in figure (8).

Example two: [7]if the user enter the number of state = 5 ,the input symbols ={a,b},and the output symbols={x,y}.

The state table for the function of the next state is shown in table (2)

The state diagram for one state is shown in figure (9). If the user applies the above example on the software then the results will shown as in figure (10).

However, by the same way, the user can apply the 2,3,4 states in the software and the results will shown in the figures (11, 12, 13).

5- CONCLUSION

1-In general, the five factors must be valid to represent any circuit or system in FSM. However, in several circuits and systems, the input and/or output may be not presents.

2-It is possible to use any type of symbols for input/output and state, also, possible to use binary numbers.

3-Software implementation depends upon the graphic tools (functions) contained in the programming language. These tools will effect directly on the implementation and modification. In other side, the programmer should be use carefully that's tools.

4-The number of (input, output) symbols and states are finite but it is possible to make it infinite by modify the software.

5-The software implemented may be used to teach the student the concept and ideas of FSM.

6-This software was simple and it considers as a primary version of implements integrated software about FSM.

REFERENCES

1. Shum series, (1986), "Discrete Mathematic", MacGro-hill and sons Inc..
2. Kolman, Bernard, and C, Robert, (1984), "Discrete Mathematic Structure for Computer Science", Busby Prentice Hall Inc..

3. Deyin, JIN ,and Yuji, ITO, (October 2008), "Application of Finite State Machine Theory to the Simulation of Reversed Non-Linear Hysteretic Relationships", TSINGHUA SCIENCE AND TECHNOLOGY ISSN 1007-0214 08/67, Volume 13, Number S1, pp46-52, http://qhxb.lib.tsinghua.edu.cn/myweb/english/2008/2008es1/46-52.pdf
4. F. Wagner, (2006), "Modeling Software with Finite State Machines: A Practical Approach", New York, USA: Auerbach Publications, ISBN 0-8493-8086-3. http://is.ifmo.ru/.../**modelingsoftwarewithfinitestatemachines**apracticalapproach.pdf
5. Hendry, Dr DC, (March 9, 2006), " Finite State Machine Design", internet document, http://www.abdn.ac.uk/~eng186/eg3560/**presentation12**.pdf.
6. Zolt_an Juh_asz, _Ad_am Sipos, and Zolt_an Porkol_ab, (2005), "Implementation of a Finite State Machine with Active Libraries in C++?", H-1117 Budapest, P_azm_any P_eter s_et_any 1/C. http://aszt.inf.elte.hu/~gsd/s/cikkek/alistair/active.pdf
7. Mano, M. Morris, (1999), "Digital Design", Second Edition, Prentice, Hall Inc.

Table (1): State table of example one.

Present state	Input	Output	Next state
q0	a	x	q0
q0	b	y	q0

Table (2): State table of example two.

Present state	Input	Output	Next state
q0	a	y	q1
q1	b	x	q0
q3	a	x	q1
q2	b	x	q2
q0	a	y	q2
q3	a	x	q4
q3	b	y	q2

Fig.(1): The main window.

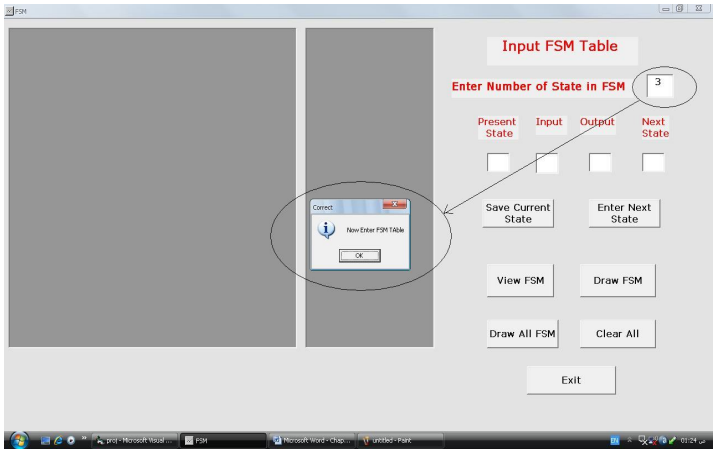


Fig.(2): Amassage box for correct enter.

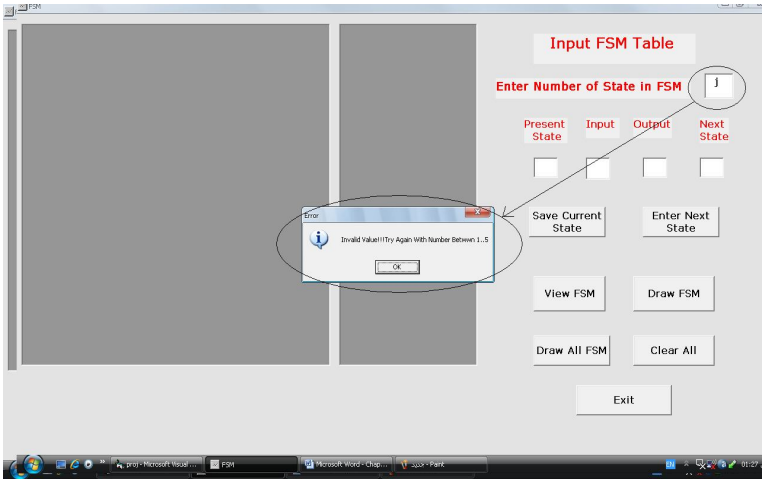


Fig.(3): Error massage box for wrong enter.

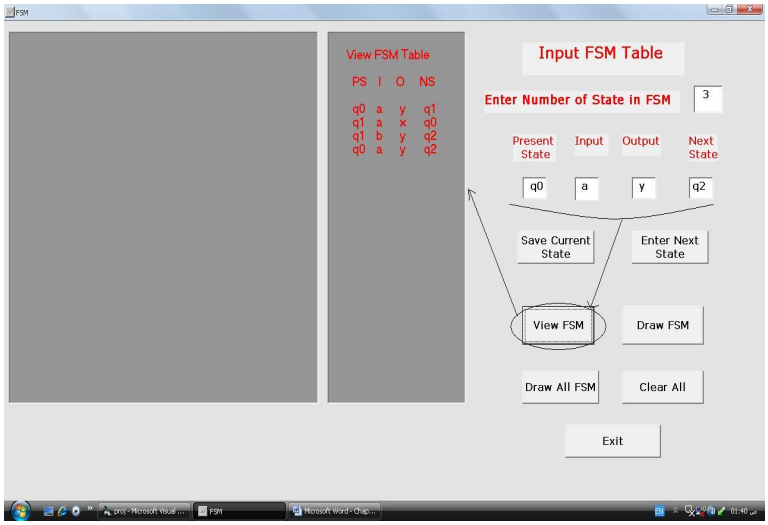


Fig.(4): View FSM button.

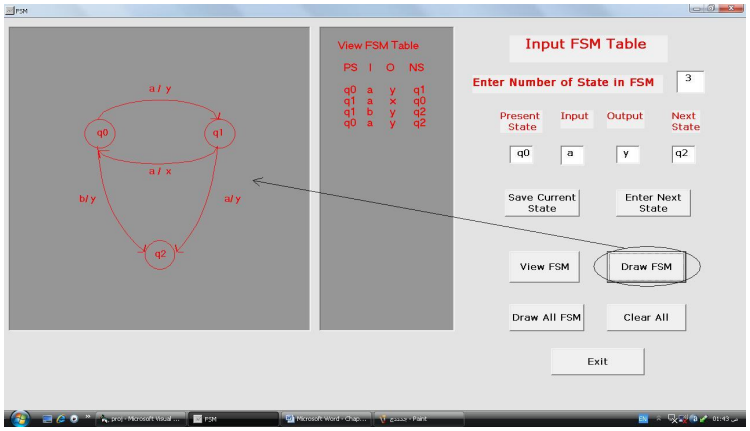


Fig.(5): Draw FSM button.

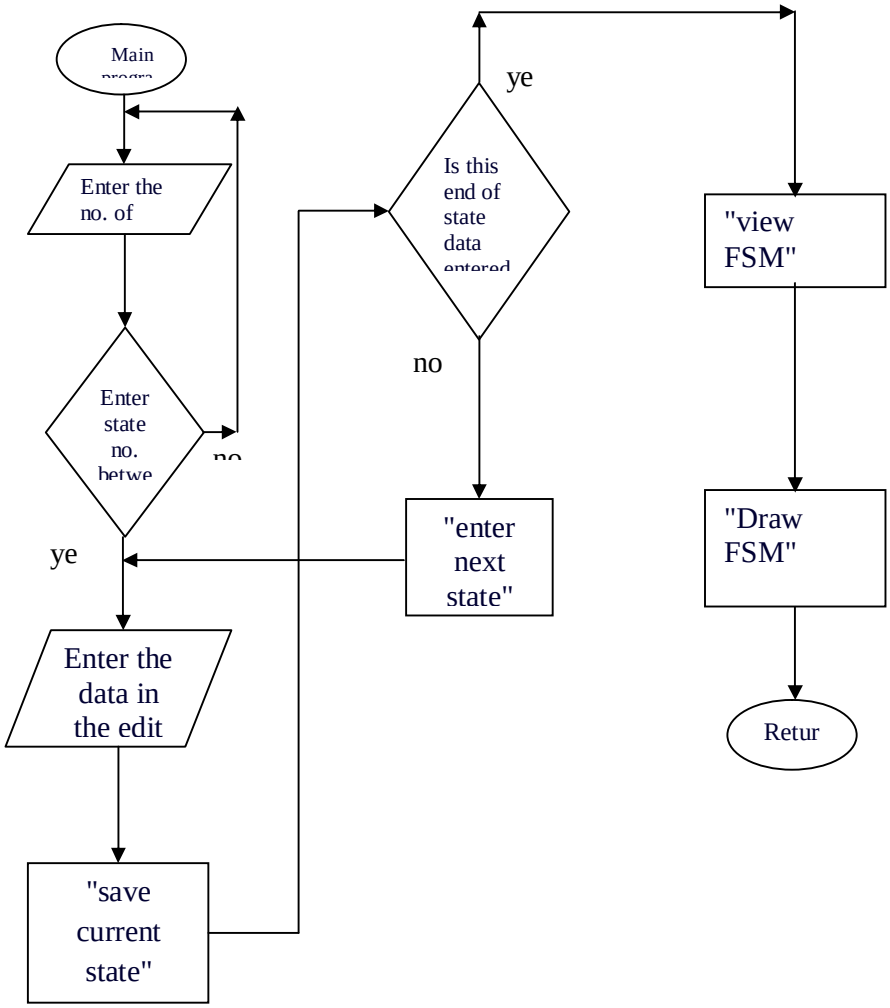
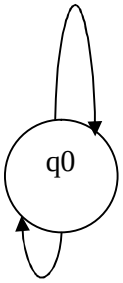


Fig.(6) : Flowchart of the FSM system.



a/x

b/y

Fig.(7): State diagram.

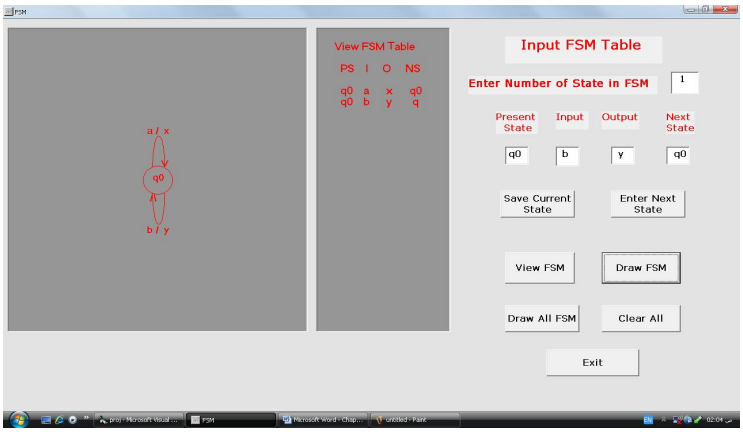


Fig.(8): Draw FSM diagram.

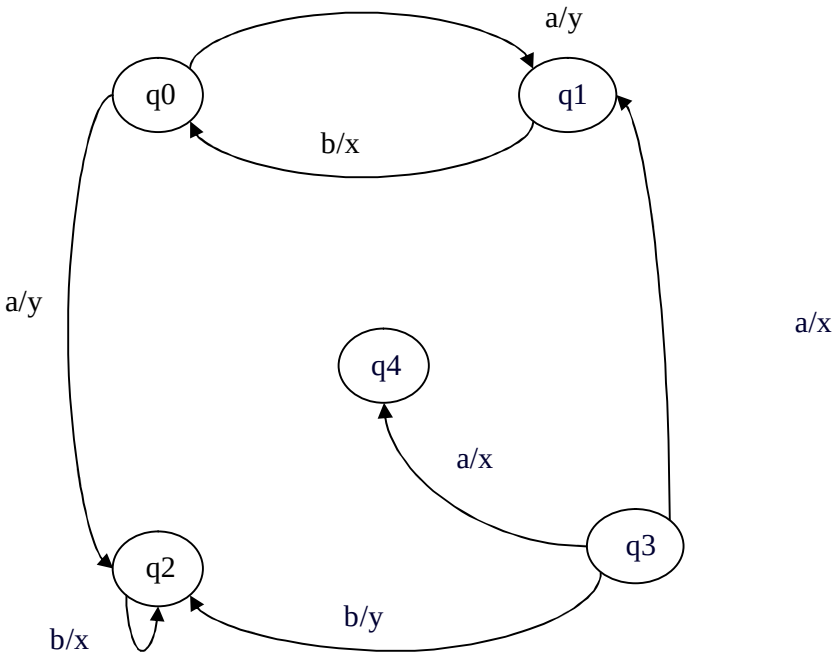


Fig.(9): State diagram of example two.

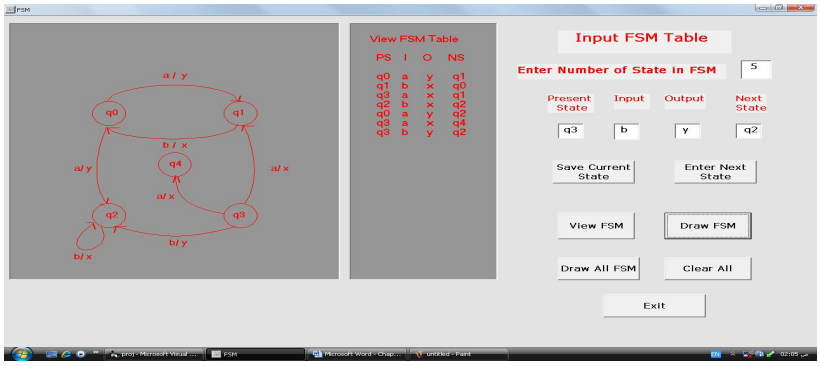


Fig.(10): Applying of example two.

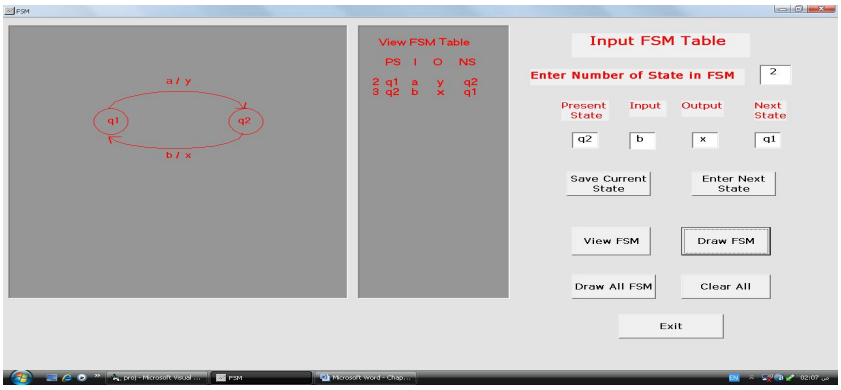


Fig.(11): Applying of 2-states.

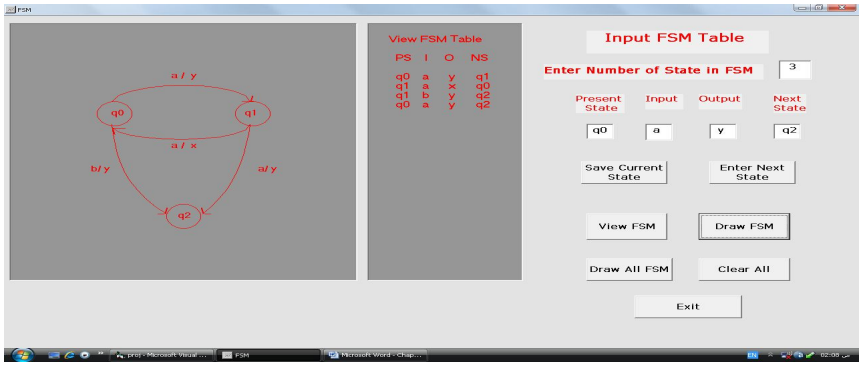


Fig.(12): Applying of 3-states.

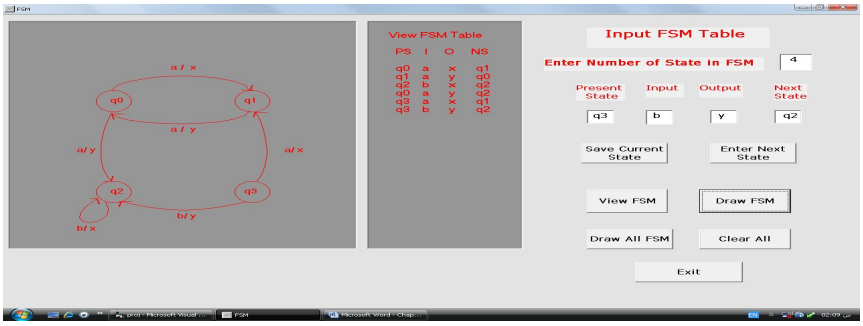


Fig.(13): Applying of 4-states.

تصميم نظام برمجي لماكنة الحالة المنتهية

حسين عبد الأمير عبد الكاظم
مدرس مساعد

تقوى فليح حسن
مدرس مساعد

قسم هندسة الحاسبات والبرمجيات /كلية الهندسة / جامعة ديالى

الخلاصة

هناك طرق عديدة مستخدمة لتمثيل عمل الدوائر والأنظمة لغرض التصميم والتحليل. ماكنة الحالة المنتهية هي إحدى الطرق التي تستخدم لتمثيل عمل دوائر وأنظمة عدة في الهندسة الالكترونية, هندسة الحاسوب,...الخ باستخدام الرسم. ماكنة الحالة المنتهية هي ماكنة بسيطة جدا من ناحية التصميم. تتألف من مجموعة من رموز المدخلات, رموز المخرجات والحالات المطلوبة لتصميمها. كذلك في الحقيقة دالة رموز المدخلات و رموز المخرجات مع الحالة الحالية لإعطاء الحالة المستقبلية يجب أن تكون موجودة. في هذه الورقة, تم بناء برنامج باستخدام لغة البرمجة الفجيول بيسك لنمذجة ماكنة الحالة المنتهية (من ناحية التصميم والعمل). في هذا النموذج تم تمثيل عدة أمثلة عامة وعلى أية حال فان البرنامج ممكن استخدامه لتعليم الطلبة مفهوم وعمل الماكنة.