ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A Self-Checking Approach for FSM Based Design on the Replication of One-Hot Code

Modi Mahendra
Department of VLSI & ESD,
SKR College of Engineering & Technology,
Konduru Satram [V], Manubolu [M],
S.P.S.R.Nellore [Dt], Andhra Pradesh, India.

Y.V.Krishna Reddy
Associate Professor,
Department of VLSI & ESD,
SKR College of Engineering & Technology,
Konduru Satram [V], Manubolu [M],
S.P.S.R.Nellore [Dt], Andhra Pradesh, India.

Abstract:

As technology scales, the protection of Finite State Machines' (FSMs) states against single event upset (SEU) and multiple bit upsets (MBUs) becomes more difficult. In this paper, a self-checking approach to enhance the SEU/MBUs immunity of FSMs' states by replicating One-Hot code times for state encoding is presented. This approach can correct less than bit-flip faults in the state register per cycle. Bit-flips are treated as random events and modeled by applying Poisson distribution. Two characteristics of this approach are obtained through probability analysis: first, this approach performs better with the increase of , whereas worse when an FSM contains more states; second, this approach can offer more enhanced reliability than Binary or One-Hot state encoding with Triple Modular Redundancy (TMR). The former characteristic leads to the further improvement of this approach which is called state-reforming. The reliabilities of this proposed approach and its state-reformed solutions, as well as are all evaluated through simulations of fault injections.

Index Terms:

FSMs, MBUs, mean-time-between-failures, Poisson distribution, SEU.

I. INTRODUCTION:

Vending Machines are used to dispense various products like Coffee, Snacks, and Cold Drink etc. when money is inserted into it. Vending Machines have been in existence since 1880s. The first commercial coin operated machine was introduced in London and England used for selling post cards.

In this paper a new approach is proposed to design an FSM based Vending Machine with auto-billing features. The machine also supports a cancel feature means that the person can withdraw the request and the money will be returned back to the user. The user will get a bill of total number of products delivered with total price. This machine can be used at various places like Hotels, Restaurants and food streets. This reduces the time and cost.

1.1.OBJECTIVE :

In This Project I Implement the self-checking approach to enhance of FSMs' states is presented. This approach replicates One-Hot code times for state encoding and has the ability to correct less than bit-flip faults in the state register per cycle. In This Project I Improved solutions state-reforming is proposed, which combine several One-Hot codes and then replicate them for state encoding. State-reforming solutions can offer higher reliabilities by reducing the number of flip-flops needed for storing state values. In this paper the process of four state (user Selection, Waiting for money insertion, product delivery and servicing) has been modeled using MEALY Machine Model.

1.2.MOTIVATION:

Nowadays, Vending Machines are well known among Japan, Malaysia and Singapore. The quantity of machines in these countries is on the top worldwide. This is due to the modern lifestyles which require fast food processing with high quality. This paper describes the designing of multi select machine using Finite State Machine Model with Auto-Billing Features. Finite State Machine (FSM) modeling is the most crucial part in developing proposed

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The vending machines are more accessible and practical than the convention purchasing method. Nowadays, these can be found everywhere like at railway stations selling train tickets, in schools and offices vending drinks and snacks , in banks as ATM machine and provides even diamonds and platinum jewelers to customers. The FPGA based machine is also more flexible, programmable and can be re-programmed. But in microcontroller based machine, if one wants to enhance the design, he has to change the whole architecture again but in FPGA user can easily increase the number of products.

The documentation of the project mainly consist two parts one is design FSM machine and second one is draw the state diagram of one hot vending machine .Implementation of the project using model sim 6.5e and synthesis by using XILINX ISE 10. This documentation helps one to understand the complete functional verification process of complex ASICs an SoC's and it gives opportunity to try the latest verification methodologies, programming concepts like Object Oriented Programming of Hardware Verification Languages and sophisticated. EDA tools, for the high quality verification. The last decade assertion-based verification (ABV) proved itself to be an efficient verification methodology based on instrumenting the design with assertions.These assertions are checked by methods including simulation, emulation, and formal verification. Assertions also document the design and play the role of executable comments.

2. SYSTEM DESIGNING
Existing System:

•Binary Encoding
•Gray encoding

2.1 Binary Encoding:

The Binary Method for assigning states is a common method to use. It counts up in binary starting from 0 and going up assigning each state the next number. It will use Log2 bits to assign the states. Binary is a common method to use because it is easy to think of what the bit code will be for each state. Students often learn binary coding as their first encoding sequence because of this reason.However, it is very inefficient. There are a minimum number of bits used to encode the machine, but the encoding equations derived for each bit are often large and complex.

The equations derived for table 1 is:
Q0+ = (Q0' * Q1) + (X * Q0') + (Q0 * Q1' * X') (2.1)
Q1+ = (X' * Q1') + (X' * Q0) + (X * Q0' * Q1) (2.2)

Because of this reason, different methods have been created that shorten the glue logic needed between the different gates.

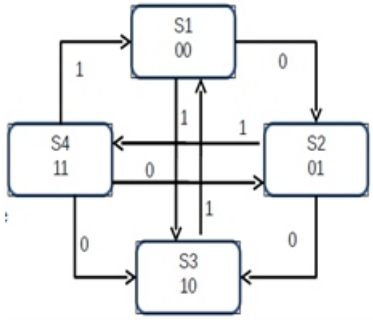


Figure 2.1 : FSM Design with Binary State Code

2.1.2 Gray Encoding:

The reflected binary code, also known as Gray code after Frank Gray, is a binary numeral system where two successive values differ in only one bit (binary digit). The reflected binary code was originally designed to prevent spurious output from electromechanical switches. Today, Gray codes are widely used to facilitate error correction in digital communications such as digital terrestrial television and some cable TV systems.The binary-reflected Gray code list for n bits can be generated recursively from the list for n−1 bits by reflecting the list, concatenating the original list with the reversed list, prefixing the entries in the original list with a binary 0, and then prefixing the entries in the reflected list with a binary 1. For example, generating the n = 3 list from the n = 2 list:

2-bit list:	00, 01, 11, 10	
Reflected:		10, 11, 01, 00
Prefix old entries with 0:	000, 001, 011, 010,	
Prefix new entries with 1:		110, 111, 101, 100
Concatenated:	000, 001, 011, 010,	110, 111, 101, 100

Table 2.1 Example Gary Code

The one-bit Gray code is G1 = (0, 1). This can be thought of as built recursively as above from a zero-bit Gray

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

code G0 = { Λ } consisting of a single entry of zero length. This iterative process of generating Gn+1 from Gn makes the following properties of the standard reflecting code clear:

•Gn is a permutation of the numbers 0... 2n−1. (Each number appears exactly once in the list.)
•Gn is embedded as the first half of Gn+1.
•Therefore the coding is stable, in the sense that once a binary number appears in Gn it appears in the same position in all longer lists; so it makes sense to talk about the reflective Gray code value of a number: G(m) = the m-th reflecting Gray code, counting from 0.
•Each entry in Gn differs by only one bit from the previous entry. (The Hamming distance is 1.)
•The last entry in Gn differs by only one bit from the first entry. (The code is cyclic.)

These characteristics suggest a simple and fast method of translating a binary value into the corresponding Gray code. Each bit is inverted if the next higher bit of the input value is set to one. This can be performed in parallel by a bit-shift and exclusive-or operation if they are available: the nth Gray code is obtained by computing.

2.2. PROBLEMS IN EXISTING METHOD:

If we design a complex controllers it contains the different (2 0r 3) FSM blocks If we use binary Or gray code state encoding methods the circuit is not working properly because same state encoding techniques are used here. They can take a long time to compile the entire program. Also the dependency issues can be slightly more difficult to decipher. They are big files, and take up more space. They are usually enabled with all the options on the install, and thus the file isn't as well tailored for your system, and it will run slower (this is one of the reasons why source based distorts like genitor are so fast).

2.3. PROPOSED METHOD
One-Hot Encoding

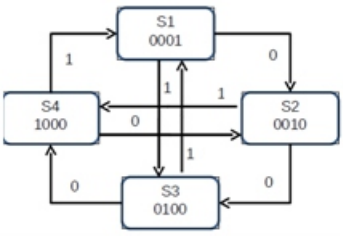


Figure 2.3: FSM Design with One Hot State Code

Using the standard method, the following equations can be easily derived by looking at the 1 on each next state. For state N0, the 1's are on rows 3 and 6. By looking at the current states and the inputs on those rows, the equation for N0 is:

N0=(X'*Q0'*Q1*Q2'*Q3')+ (X*Q0'*Q1'*Q2*Q3') (2.3)

By using the same method, the equations for the other next states are:

N1=(X'*Q0'*Q1'*Q2*Q3')+ (X*Q0'*Q1'*Q2'*Q3) (2.4)
N2=(X'*Q0'*Q1'*Q2'*Q3)+ X'*Q0*Q1'*Q2'*Q3') (2.5)
N3=(X*Q0'*Q1*Q2'*Q3') + (X*Q0*Q1'*Q2'*Q3') (2.6)

Only one of these states are Hot at any given time. This means if Q0 is 1, then Q1, Q2 and Q3 are all 0. Likewise with all bits, if one is a 1, then all others are 0. Because of this, the equations can be reduced to:

N0 = (X' * Q1) + (X * Q2) (2.7)
N1 = (X' * Q2) + (X * Q3) (2.8)
N2 = (X' * Q3) + (X' * Q0) (2.9)
N3 = (X * Q1) + (X * Q0) (2.10)

Input	Current State				Next State			
X	Q0	Q1	Q2	Q3	N0	N1	N2	N3
0	0	0	0	1	0	0	1	0
0	0	0	1	0	0	1	0	0
0	0	1	0	0	1	0	0	0
0	1	0	0	0	0	0	1	0
1	0	0	0	1	0	1	0	0
1	0	0	1	0	1	0	0	0
1	0	1	0	0	0	0	0	1
1	1	0	0	0	0	0	0	1

Table2.2: FSM Sate Encoding Transition Table

This type of encoding uses more bits to encode the states, but reduces the logic needed to glue the bits together. It is a very common coding to use, especially for many states.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

One Hot Encoding is an encoding sequence that uses as many registers as there are states. With One Hot Encoding, only one of the bits are 1 or "Hot" at any given state. All the other bits are 0. This makes sure only two bits change when moving from one state to the next. With a maximum of two bits changing at any given time less power is consumed. In addition, One Hot Encoding reduces the logic needed to implement each bit. However, it also uses many more logic gates to implement the state design. One Hot is often used where there are many states that need to be implemented in the design. Using binary encoding, the complexity of the design grows with each bit added. With One Hot, the number of flip-flops grows with each state added, but the complexity of each equation does not. Because of this, it is more difficult to accidentally mess up the logic needed to implement the bits. K-Maps are not needed for this type of encoding either. Instead, a truth table is created and used to find the equation for each bit.

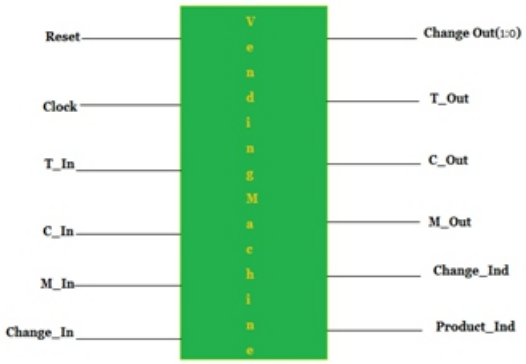


Figure 2.4 Basic One Hot Vending Machine

3.ONE HOT -VENDING MACHINES:

Vending Machines are used to dispense various products like Coffee, Snacks, and Cold Drink etc. when money is inserted into it. Vending Machines have been in existence since 1880s. The first commercial coin operated machine was introduced in London and England used for selling post cards. The vending machines are more accessible and practical than the convention purchasing method. Nowadays, these can be found everywhere like at railway stations selling train tickets, in schools and offices vending drinks and snacks , in banks as ATM machine and provides even diamonds and platinum jewelers to customers. The FPGA based machine is also more flexible, programmable and can be re-programmed. But in microcontroller based machine, if one wants to enhance the design, he has to change the whole architecture again but in FPGA user can easily increase the number of products.

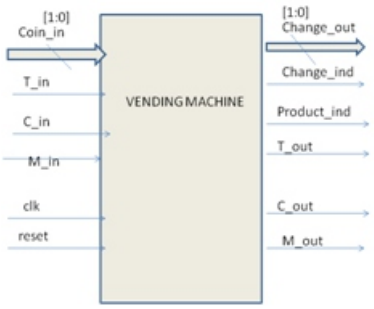
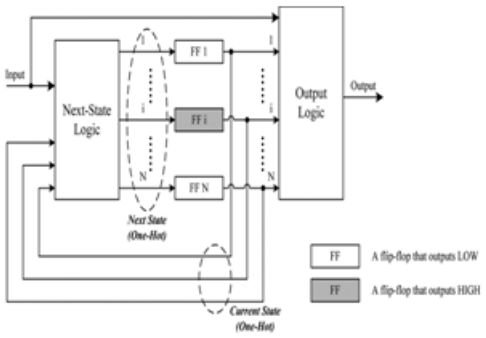


Figure 3.1 One Hot Vending Machine

In this paper a new approach is proposed to design an FSM based Vending Machine with auto-billing features. The machine also supports a cancel feature means that the person can withdraw the request and the money will be returned back to the user. The user will get a bill of total number of products delivered with total price. This machine can be used at various places like Hotels, Restaurants and food streets. This reduces the time and cost.

3.2.OPERATION OF VENDING MACHINE:

I. When the user puts in money, money counter tells the control unit, the amount of money inserted in the Vending Machine.
II. When the user presses the button to purchase the item that he wants, the control unit turns on the motor and dispenses the product if correct amount is inserted.
III. If there is any change, machine will return it to the user.
IV. The machine will demand for servicing when the products are not available inside the machine.



ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

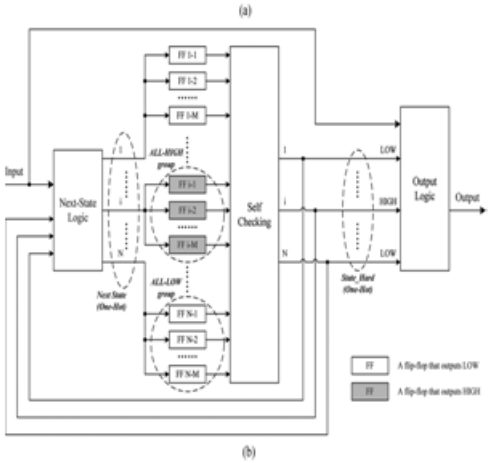


Figure 3.2 : FSM Next and Present State Logic

3.3.IMPLEMENTATION OF VENDING MACHINE:

In this paper a state diagram is constructed for the proposed machine which can vend 3 products that is coffee, Milk, Tea. Three select (select1, select2, select3) inputs are taken for selection of products. Select1 is used for the selection of Tea. Similarly select2, select3 are used for coffee and Milk respectively. Rs_1,Rs_2 and rs_5 inputs represents rupees 1,2,5 Coins respectively. A Reset input is also used when the user wants to withdraw his request and also the money will be returned through the return output. Return, product and change are the outputs. Return and change vectors are seven bits wide.

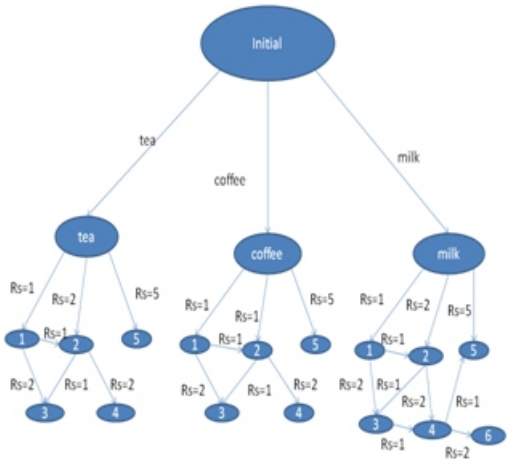


Figure 3.3 Proposed Vending Machine State Diagram

Money is an in/out signal which can be updated with the total money of all products delivered at a time. Money signal is seven bits wide. Money_ count is an internal signal which can be updated at every transition. This signal is also seven bits wide. If the inserted money is more than the total money of products then the change will be returned through the change output signal. The products with their prices are shown by table 1. There are also two input signal clk and reset. The machine will work on the positive edge of clock and will return to its initial state when reset button is pressed. The proposed vending machine is designed using FSM modeling and is coded in Verilog HDL language.

4. DESIGN METHODOLOGY:

The state diagram mainly consists of four states (User Selection, Waiting for the money nsertion, product delivery and servicing (when product_not_available='1')). Initially when the reset button is pressed, the machine will be ready for the users to select the product. This state is the initial state of the design. After this the user will select the product to be dispensed. This state can be one of the select1, select2, select3 and select 4. The machine can accept only two types of notes i.e. rupees 1/- and 2/-. Let us suppose that the user selects sel1 input. The machine will firstly check that whether the products are available in the machine or not. After this the control unit will move to the waiting state, where it will wait for the money to be inserted. Then if rupees 1/- note is inserted then the machine will go to state_1 and wait until the desired money is inserted. And if rupees 2/- note is inserted the machine will move to state_2 and then wait until 30/- rupees are inserted to the machine. When the desired amount is inserted the machine will go to the snacks state and snacks will be delivered at the product output.

If products are not available in the machine then the control unit will demand for servicing and after service the machine will get reset. There is also an additional feature of withdrawing the request if the user doesn't want to take the product. When cancel button is pressed then the money inserted will be returned to the user through the return output. A money count signal is used for calculating the total money inserted in the machine. And if the money inserted is more than the money of the product then the extra change will be returned to the user. The total amount of the product taken at a time is shown by the money signal. Similarly the user can select and get the other products following the above procedure.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

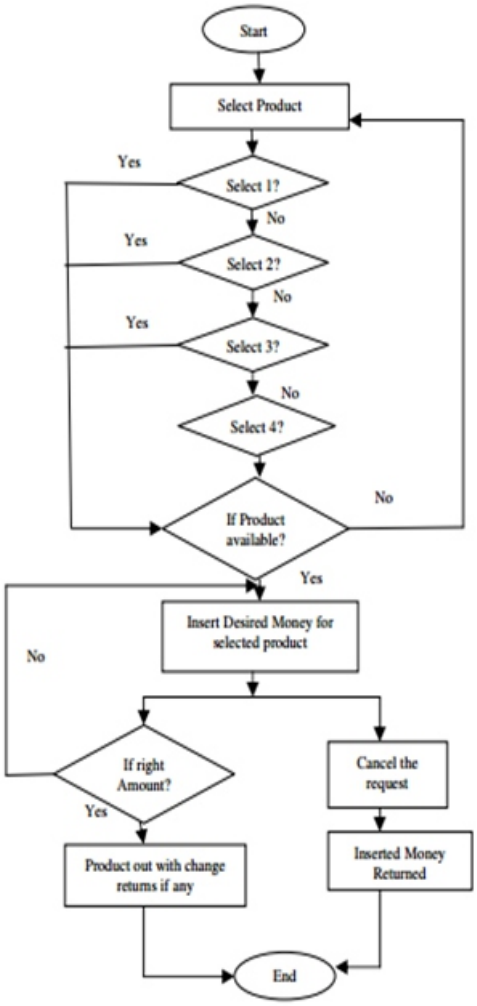


Figure 4.1 Flow Chart For Proposed Vending Machine

Description of states:

The selection of products and all the states are given below.

• When initialize=>
•money_count=0;
•Change=0;
•Product=0;
• When select1=>Sel1&!sel2&!sel3
When product_available=1=>
nx_st1<= waiting1;
When product_available=0 =>
nx_st1<= service1;
• When waiting1=>
When rs_1&!rs_2=> nx_st1<=state_1;
When !rs_1&rs_2=> nx_st1<= state_2;
Change=0; product=0;
When money_count>=3nx_st1<= Tea;
• When state_1=>
Rs_1=1 & rs_2=0;
Change=0; Product=0;
Money_count=money_count+11;

5. SIMULATION RESULTS:

The state diagram shown in figure is simulated using Modelsim 6.5 e. Simulation Waveforms for the selection of Three products like Coffee, Tea, Milk is shown in figure. respectively with servicing feature when products are not available in the machine and change return features when the money inserted is more than the money of the product.Let us take an example that the user wants to take Tea .When one selects sel1 button, the machine will check that whether the products are available or not, if available then it will go to the waiting state and wait for total money insertion. If rs_1 note is inserted it will go to state_1 and if rs_2 note is inserted it will so to state_2 and check whether money_count>=3 or not. If the money_count >- 3 then machine will go to state snacks and vend the product.

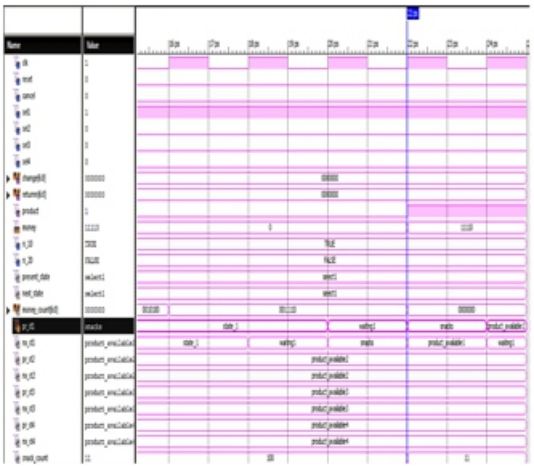


Figure 5.1: Simulation waveform for Coffee Selection

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

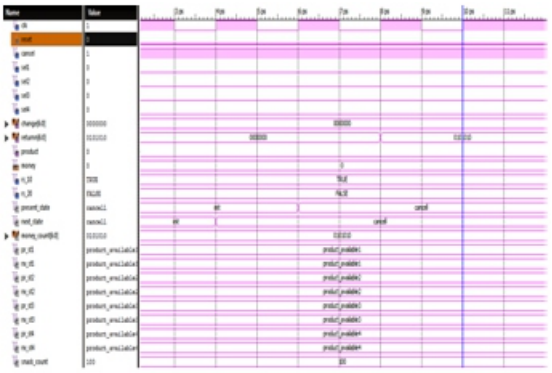


Figure 5.2: Simulation waveform for Tea Selection

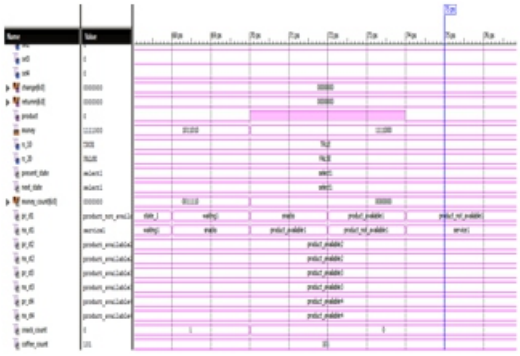


Figure 5.3 : Simulation wave for Milk Selection

5.2 SYNTHESIS REPORTS :

5.2.1 BINARY ENCODING :

JKJH Project Status			
Project File:	jkjh.ise	Current State:	Synthesized
Module Name:	fsmdesign_binary	• Errors:	No Errors
Target Device:	xc3s100e-4tq144	• Warnings:	1 Warning
Product Version:	ISE 9.2i	• Updated:	Thu Aug 1 13:34:40 2013

JKJH Partition Summary
No partition information was found.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	22	960	2%
Number of Slice Flip Flops	27	1920	1%
Number of 4 input LUTs	39	1920	2%
Number of bonded IOBs	14	108	12%
Number of GCLKs	1	24	4%

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Thu Aug 1 13:34:45 2013	0	1 Warning	7 Infos
Translation Report					

Figure 5.4: Synthesis waveform for Binary Code

5.2.2 ONE HOT ENCODING:

JKJH Project Status			
Project File:	jkjh.ise	Current State:	Synthesized
Module Name:	fsmdesign_onehot	• Errors:	No Errors
Target Device:	xc3s100e-4tq144	• Warnings:	23 Warnings
Product Version:	ISE 9.2i	• Updated:	Thu Aug 1 13:31:05 2013

JKJH Partition Summary
No partition information was found.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	10	960	1%
Number of Slice Flip Flops	13	1920	0%
Number of 4 input LUTs	18	1920	0%
Number of bonded IOBs	14	108	12%
Number of GCLKs	1	24	4%

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Thu Aug 1 13:31:02 2013	0	23 Warnings	6 Infos
Translation Report					

Figure 5.5: Synthesis waveform for One-hot Code

RTL Schematic :

One Hot Encoding:

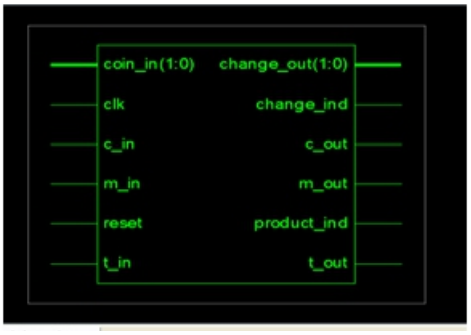


Figure 5.6 RTL View of One Hot Vending Machine

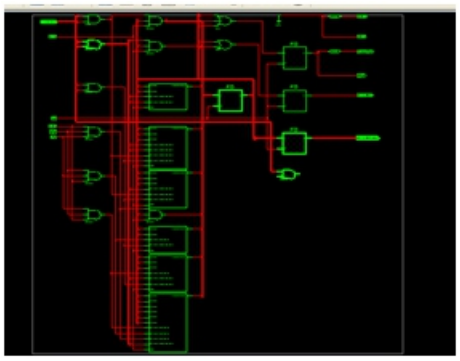


Figure 5.7 RTL Schematic of One Hot Vending Machine

TIMING REPORT

Clock Information:

Clock Signal | Clock buffer(FF name) | Load |

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

clk | BUFGP | 13 |

Asynchronous Control Signals Information:

No asynchronous control signals found in this design
Timing Summary:
Speed Grade: -4
Minimum period: 3.263ns (Maximum Frequency: 306.466MHz)
Minimum input arrival time before clock: 4.959ns
Maximum output required time after clock: 4.310ns
Maximum combinational path delay: No path found
Timing Detail:
Data Path: m_in to state_0

Cell:in->out	Gate fanout	Net Delay	Delay	Logical Name (Net Name)
IBUF:I->O	4	1.218	0.762	m_in_IBUF (m_in_IBUF)
LUT3:I0->O	1	0.704	0.455	next_state<0>128 (next_state<0>1_map11)
LUT3_L:I2->LO	1	0.704	0.104	next_state<0>147_SW0 (N142)
LUT4:I3->O	1	0.704	0.000	next_state<0>1601 (N133)
FDS:D		0.308		state_0
Total		4.959ns (3.638ns logic, 1.321ns route)		
		(73.4% logic, 26.6% route)		

Binary Coded Design :

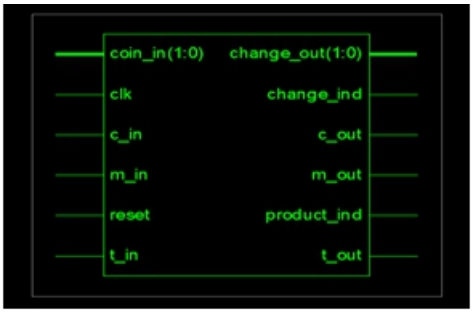


Figure 5.8 RTL View of Binary Coded Vending Machine

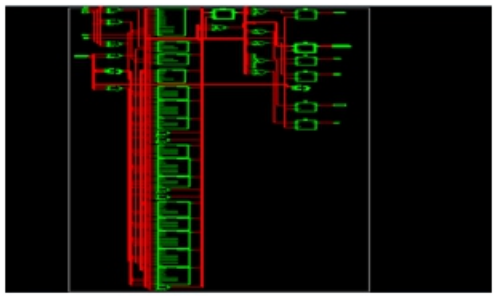


Figure 5.9 RTL Schematic of Binary Coded Vending Machine

==========
==========
* Final Report *
==========
==========
Final Results
RTL Top Level Output File Name : fsmdesign_binary.ngr
Top Level Output File Name : fsmdesign_binary
Output Format : NGC
Optimization Goal : Speed
Keep Hierarchy : NO
Design Statistics
IOs : 14
Cell Usage :
BELS : 39
LUT2 : 2
LUT3 : 11
LUT3_L : 1
LUT4 : 20
LUT4_D : 1
LUT4_L : 4
FlipFlops/Latches : 27
FD : 15
FDR : 7
FDS : 5
Clock Buffers : 1
BUFGP : 1
IO Buffers : 13
IBUF : 6
OBUF : 7
==========
==========
Device utilization summary:

Selected Device : 3s100etq144-4
Number of Slices: 22 out of 960 2%
Number of Slice Flip Flops: 27 out of 1920 1%

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

```
Number of 4 input LUTs:                39  out of   1920
2%
 Number of IOs:                        14
 Number of bonded IOBs:                14  out of    108
12%
 Number of GCLKs:                       1  out of     24    4%
=========================================
=================================


TIMING REPORT
Clock Information:
-----------------------------------+------------------------+------
-+
Clock Signal                       | Clock buffer(FF name)  |
Load  |
-----------------------------------+------------------------+------
-+
clk                                | BUFGP                  | 27    |
-----------------------------------+------------------------+------
-+
Timing Summary:
---------------
Speed Grade: -4
     Minimum period: 3.474ns (Maximum Frequency:
287.853MHz)
   Minimum input arrival time before clock: 5.060ns
   Maximum output required time after clock: 4.283ns
   Maximum combinational path delay: No path found
```

Comparison Table :

FSM State Encoding	Area in Slices	Delay
Binary	22	5.06ns
One hot	10	3.43ns

Table 5.1 : Synthesis and Delay Comparison

CONCLUSION & FUTURE SCOPE:

It takes more time and money initially to use these methods to reduce the number of states and logic to implement these states. Because of this, when designing a simple logic circuit, these methods are not needed. If a complex design with many states is required however, it takes more time and money in the end to not these methods. When dealing with state machines that have many inputs and states, the circuit can become very large and complex.

The more complex the circuit, the easier it is to make a mistake in the logic. The present One Hot Encoding FSM vending machine controller is implemented using FSMs with the help of Xilinx ISE Design Suite 9.2 State machines based vending Systems enhances productivity, reduces system development cost, and accelerates time to market. Also FSM based vending machine give fast response and easy to use by an ordinary person. The designed machine can be used for many applications and we can easily enhance the number of selections. Compare to Binary Encoded based Design One- Hot Method is High Speed Shown in Compassion Table 5.1

REFERENCES:

[1] Fauziah Zainuddin, Norlin Mohd Ali, Roslina Mohd Sidek, Awanis Romli, Nooryati Talib & Mohd. Izham Ibrahim (2009) “Conceptual Modeling for Simulation: Steaming frozen Food Processing in Vending Machine” International Conference on Computer Science and Information Technology, University Malaysia Pahang, pp.145-149.

[2] C. N. Hadjicostis and G. C. Verghese, “Coding approaches to fault tolerance in linear dynamic systems,” IEEE Trans. Inf. Theory, vol. 51, no. 1, pp. 210–228, Jan. 2005.

[3] M. J. Gadlage, R. D. Schrimpf, B. Narasimham, B. L. Bhuva, P. H. Eaton, and J. M. Benedetto, “Effect of voltage fluctuations on the single event transient response of deep submicron digital circuits,” IEEE Trans. Nucl. Sci., vol. 54, no. 6, pp. 2495–2499, Dec. 2007.

[4] S. M. Jahinuzzaman, M. Sharifkhani, and M. Sachdev, “An analytical model for soft error critical charge of nanometric SRAMs,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 17, no. 9, pp. 1187–1195, Sep. 2009.

[5] Horn, and Delton T, “Electronic Components A Complete Reference for Project Builders”, McGraw-Hill/Tab Electronics, 1991.

[6] Janick Bergeron, “Writing Testbenches: Functional Verification of HDL Models”, Kluwer Academic Publishers, 2009.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[7] Www.Testbench.In

[8] Www.Ovmworld.Org

[9] Www.Systemverilog.Org

[10] Www.Cisco.Com/Go/Isr

[11] Www.Vmmcentral.Com

About Authors:

Modi Mahendra Was Born In Musthapuram [V] Ananthasagaram[M], S.P.S.R.Nellore [Dt], Andhra Pradesh, India. He Received The B.Tech Degree In Electronics And Communication Engineering From Priyadarsini Institute Of Technology Affiliated To Jnt University, Anantapur In 2013 And Pursuing M.Tech Degree In Vlsi & Esd From Skr College Of Engineering & Technology, Konduru Satram [V], Manubolu [M], S.P.S.R.Nellore [Dt], Andhra Pradesh, India, Affiliated To Jnt University, Anantapur.

Mr.Y.V.Krishna Reddy Was Born In Andhra Pradesh, India. He Received The B.Tech Degree In Elecronics And Communication Engineering From Audhishankara College Of Engineering And Technology Affiliated To Jnt University, Anantapur And M.Tech Degree In Embedded System From Nova College Of Engineering & Technology, Ibrahimpatnam, Telangana Affiliated To Jnt University-Hyderebad.. He Is Working As Associate Professor In Dept. Of Electronics And Communication Engineering And Specialization In Embedded System Design At Skr College Of Engineering & Technology, Konduru Satram [V], Manubolu [M], S.P.S.R Nellore [Dt], Andhra Pradesh, India.