Proceedings of Engineering & Technology (PET)
pp. 61-66

Typical Design of Low Power Finite state Machine

Siwar Ben Haj Hassine, Mehdi Jemai ,Bouraoui Ouni and hatem samaali

***Abstract*— As power dissipation has become a high-priority cost metric, designers and researchers have increased their efforts to understand its sources and minimize its impact. In this paper, we propose a new approach that aims to reduce the power consumption when designing a Finite State Machine. Our approach has been tested and compared to classical designs. Design results have shown significant reduction of 74.63% in dynamic energy consumption.**

***Keywords*—FSM; power consumption; power optimization.**

I. INTRODUCTION

A finite-state machine is a mathematical model of computation largely used to design both: computer programs and sequential logic circuits. It is conceived as an abstract machine that can be in one of a finite number of states. The machine is in only one state at a time. The state at any given time is called the current state. At the clock event, the FSM transits from a current state to a next one if the condition which is related to the current state is true. In other circumstances when the condition is false, the FSM returns back to the current state (same state). One may, nevertheless, wonder why the FSM leaves the current state and returns back to the exactly same state. In this paper, we have introduced a design method that blocks all transitions from current state to current state. To meet this aim, our approach is based on blocking the master clock in case where a next state equals to a current one. We illustrate this idea on the following example.

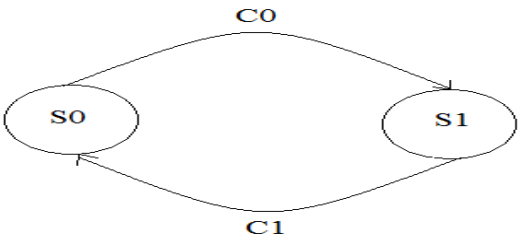

Figure 1. Illustrative example

We assume that the current state is S0. According to our approach, if C0 is not triggered then the clock will be blocked. If C0 is activated then a transition from S0 to S1 takes place and S1 becomes the current state. By the same way, a transition from S1 to S0 will only takes place if C1 is triggered. Otherwise, the clock will be blocked. The simulation results are shown in Fig. 2.

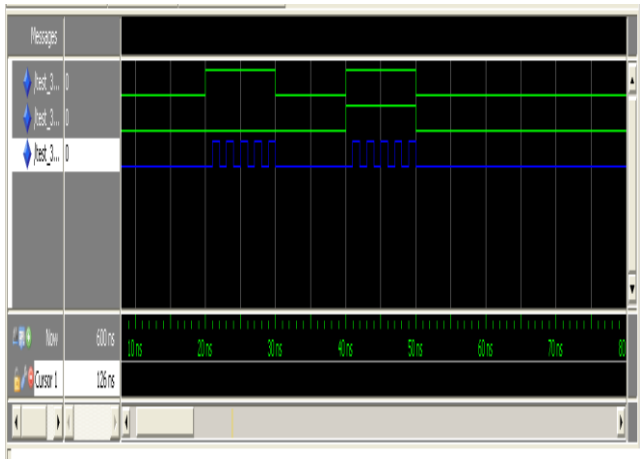

Figure 2. Simulation results

The Fig. 2 shows that the clock was blocked until 21 ns. At 21 ns, the condition C0 becomes true so our approach activates automatically the clock, and the FSM goes from S0 to S1. After that, our approach re-blocks the clock until C1 will be true.

Using Fig. 2, let us calculate the dynamic power dissipated by the FSM in the time interval T= [0 ns, 50ns]

Dynamic Energy dissipated by classical approach is:

$$Edis=Pdis \times T= T \times (A\times C \times V^2 \times F) \quad (1)$$

$$Edis=Pdis \times T= 50\times (A\times C \times V^2 \times F) \quad (2)$$

Where A refers to the percentage of active logic gates; C is the total capacitance load, V presents the supply voltage, F is the execution frequency and T refers to the time.

Dynamic Energy by our approach is:

$$Edis= T\times (Pdis1+Pdis2+Pdis3+Pdis4) \quad (3)$$

Where: Pdis1: Power dissipated in the interval [0 ns, 21 ns]

Pdis2: Power dissipated in the interval [21 ns, 30 ns]

Pdis3: Power dissipated in the interval [30ns, 41 ns]

Pdis4: Power dissipated in the interval [41 ns, 50 ns]

$$Pdis1=Pdis3= 0 \quad (4)$$

Because the clock does not function, the frequency equals 0.

$$Pdis2=Pdis4= T\times (A\times C \times V^2 \times F) \quad (5)$$

$$Edis= 9 \times (A\times C \times V^2 \times F) + 9 (A\times C \times V^2 \times F) \quad (6)$$

$$Edis= 18 \times (A\times C \times V2\times F) \quad (7)$$

Siwar Ben haj hassine e-mail: siwar.haj.hassine@gmail.com
Mehdi Jemai e-mail: jmehdie@gmail.com
Bouraoui Ouni e-mail: ouni_bouraoui@yahoo.fr
Hatem samaali e-mail: samaali_hatem@yahoo.fr
Laboratory of Electronic and Microelectronic, University of Monastir, Monsatir 5000, Tunisia.

ISSN: 2356-5608

Hence, based on (2) and (7), in this example, our approach provides a gain of 64% in dynamic energy compared to classical approach.

II. RELATED WORK

FSM are widely used to resolve problems like, parsing, electronic design automation, communication protocol and other engineering applications. The clock is the major consumer of power in the FSMs. For that designers have maid they efforts to solve that dilemma. For instance, many clock gating methods have been introduced. For example, authors in [1] have presented a new design that reduces energy, electromagnetic iteration and clock skew. They have used a single-edge triggered flip-flops-SET-FF to synthesize synchronous pipelined electronic systems which operates on both edges of the clock signal. Besides, authors in [2] have introduced a method that provides a significant gain in energy consumption while designing an electronic system by a synchronous FSM. In that approach, they have needed only a timer in order to modify the state of the processor to a low power state. Moreover, in [3] authors have optimized the number of clocked transistors in order to minimize the capacity of the clock load to have a low power clocking system. Practically speaking, they have proposed a clocked pair shared flip-flop that meant to diminish the number of local clocked transistors. In addition, authors in [4] have come up with a new model of power gated design of FSM. For that they have used Genetic Algorithm to solve the dilemma of bi-partitioning and encoding which leads to effective dynamic power consumption. In [5] designers have developed a new FSM that runs at high clock frequency and consumes less power. In fact, this concept has been used to design a PFM FSM in a field programmable gate array and in a 65 nm CMOS technology. Finally, in [6] authors have presented a design based on synchronous FSM to optimize dynamic power consumption in digital systems. To do that, they have applied a methodology where energy consumption is a concern from the first stages of the design cycle.

III. CLASSICAL DESIGN

Often, the first design step of FSM is to attribute to each state a binary code. So far, all classical methods as well as computer aided design synthesis tools have provided an FSM design by the following way, Fig. 3.

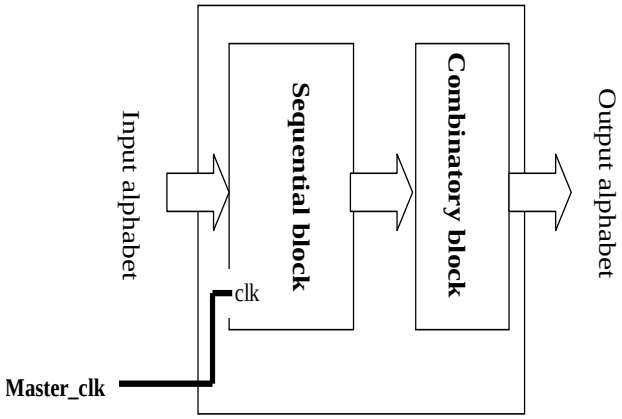

Figure 3. Classical design of FSM

A. Sequential block

The inputs of this block are the master clock and the input alphabet. The outputs of this block are the binary of the current state.

B. Combinatory block

This block decodes the current state into the outputs. The instant the current state changes, those changes ripple through this block, and almost instantaneously the output gets updated.

IV. PROPOSED DESIGN

Our approach is based on adding a new device called "clock controller". The clock controller produces a new clock called Clk_cont. The later is used to handle the FSM, in other words it replaces the master clock, Fig. 4. We note that this modification on the master clock does not change the functionality of the FSM. In other word the FSM keeps run correctly.

A. The inputs of our clock controller

The inputs of the clock controller are: the binary code of the current state, the input conditions and the master clock.

B. Architecture of clock controller

The architecture is based on multiplexer and two-input AND logic gate.

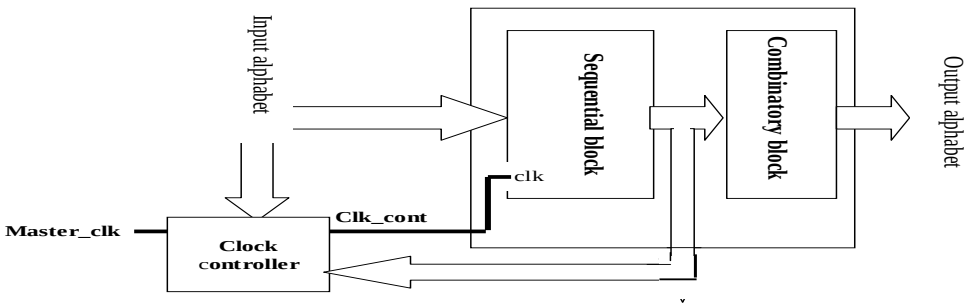

Figure 4. Proposed design

1) The multiplexer

The multiplexer has 2^p data inputs and p data address. The address inputs are used to select which input will be send to the output. In our approach, p equals to the number of bits used in the binary code. The address inputs of the multiplexer are mapped to the outputs of sequential block. The output of the multiplexer is mapped to a two-input AND logic gate. The data inputs of the multiplexer are mapped to the input conditions such as: Input (i) of the multiplexer is mapped to the condition related to binary code (i).

2) Two-input AND logic gate

Two-input AND logic gate running as:

$$Clk\_cont=S \text{ AND } clk \quad (8)$$

S is the output of the multiplexer and clk is the master clock.

C. The output of our clock controller

The clock controller has only one output which is the Clk_cont (provided by the two-input AND logic gate). The later will be the clock of the sequential block.

V. Experiment

To further clarify our approach, we have applied it on the FSM below (Fig. 5) then compared it with the traditional one.

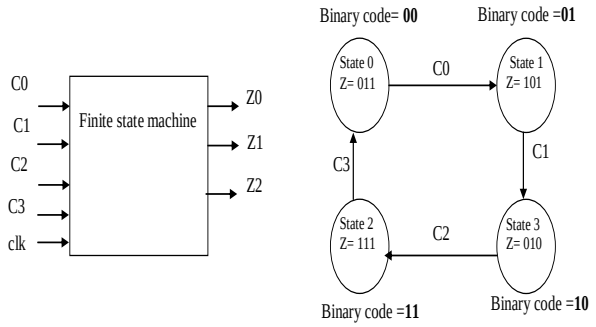

Figure 5. Finite state machine

A. Designing of clock controller

In our example the number bits used in the binary code= 2, hence p=2.

1) The multiplexer

Address inputs = p=2, data inputs: 2^p=4. The address inputs of the multiplexer are mapped to the outputs of sequential block such as Input data (i) of the multiplexer is mapped to the condition related to the binary code (i); hence input data (0) is mapped to C0, input data (1) is mapped to C1, input data (2) is mapped to C2 and the input data (3) is mapped to C3. The output S of the multiplexer is mapped to the two-input AND logic gate, Fig. 6.

2) Two-input AND logic gate

Two-inputs AND logic gate running as Clk_cont = S AND clk. S is the output of the multiplexer and clk is the master clock.

B. Simulation results

Fig. 7 represents the simulation results provided by our approach. We can definitely deduce through an analysis of the Fig. 7 that our approach has kept the correct functionality of the FSM. Furthermore, Fig 7 shows that Clk_cont is activated only when the FSM needs it.

Let us now calculate the dynamic power in the interval of time t= [0ns to 59 ns]. We note that we have neglect the energy dissipated by the two-input AND logic gate

1) Dynamic Energy dissipated by classical approach

$$Edis=Pdis\times T= T\times (A\times C\times V^2\times F) \quad (9)$$

$$Edis=Pdis\times T= 59\times (A\times C\times V^2\times F) \quad (10)$$

2) Dynamic Energy by our approach

$$Edis= T\times (Pdis1+Pdis2+Pdis3+Pdis4+Pdis5+Pdis6+ Pdis7+ Pdis8+Pdis9) \quad (11)$$

Since T=59 ns

$$Edis= 59\times (Pdis1+Pdis2+Pdis3+Pdis4+Pdis5+Pdis6+ Pdis7+ Pdis8+ Pdis9) \quad (12)$$

Pdis1: Power dissipated in the interval [0ns, 10 ns]
Pdis2: Power dissipated in the interval [10ns, 12 ns]
Pdis3: Power dissipated in the interval [12 ns, 22 ns]
Pdis4: Power dissipated in the interval [22 ns, 24 ns]
Pdis5: Power dissipated in the interval [24 ns, 30 ns]
Pdis6: Power dissipated in the interval [30 ns, 32 ns]
Pdis7: Power dissipated in the interval [32 ns, 42 ns]
Pdis8: Power dissipated in the interval [42 ns, 44 ns]
Pdis9: Power dissipated in the interval [44 ns, 59 ns]

Since Pdis1=Pdis3=Pdis5=Pdis7=Pdis9= 0 (13)
(frequency =0)

$$Edis= Pdis2\times T2+ Pdis4\times T4+ Pdis6\times T6+ Pdis8\times T8$$

Since T2=T4=T6=T8=T=2 ns

and Pdis2=Pdis4=Pdis6 = Pdis8= (A× C ×V^2 × F)

Edis= 8 × (A× C ×V^2 × F) (14)

Based on (12) and (14), our approach produces an important gain of 86.44% in dynamic energy compared to classical design.

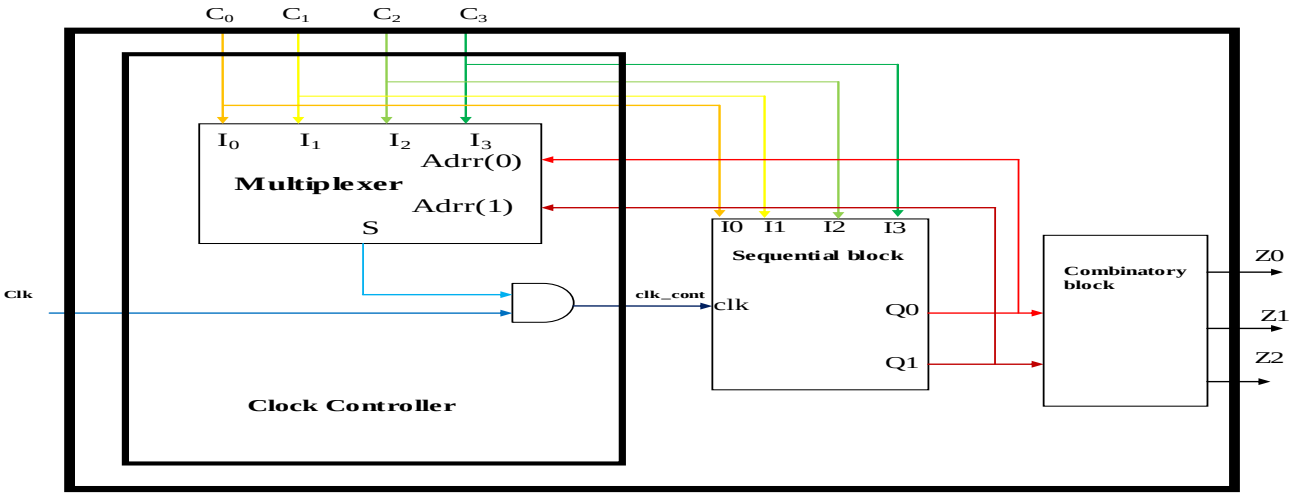

Figure 6. Clock controller

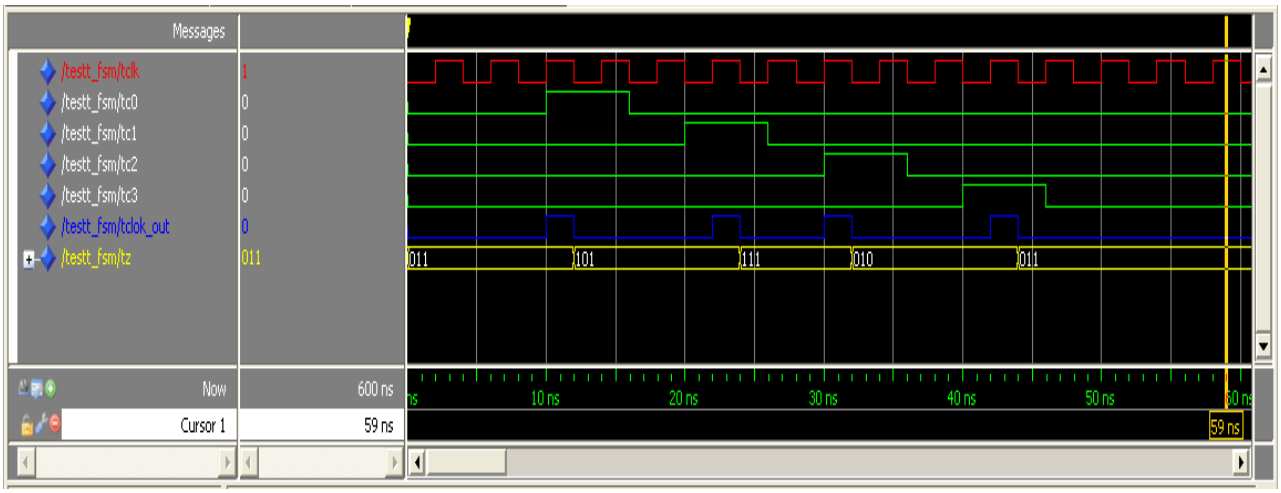

Figure 7. Simulation result: (Red wave: master clock, blue wave: Clk_out, yellow wave: outputs)

C. Physical results

In this part, we intend to implement a comparative study between our approach and the traditional one. To put that into practice, we have used different tools consisting of an HP computer, windows 7 as an operative system, ISE 12.3 tools and a XiIinx nexys 2 based on Spartan 3E as an electronic board. In this part, we have executed the FSM during 59 seconds.

Table I. design results

	Classical design	**Proposed design**
Dynamic energy (J)	0,32396	0,08216

According to table 1, our approach produces a gain of 74.63% in dynamic energy compared to classical design.

VI. CONCLUSION

Unfortunately, FPGA synthesis tools such XST of Xilinx, Leonardo spectrum of mentor graphic and Quartus of altera, still neglect the optimization of power dissipation through the restriction of the synthesis-optimization goal to either the speed (performance) or the area (resources). Hence, in this paper we have developed a design method that aims to build a low power finite state machine (FSM).

REFERENCES

[1] Oliveira D.L, Curtinhas T., Faria L.A., Romano .L," Design of synchronous pipeline digital systems operating in double-edge of the clock", Circuits and Systems (LASCAS), 2013 IEEE Fourth

Latin American Symposium on Feb. 27 2013-March 1 2013.pp 1-4.

[2] Renaux D.P.B, Pottker F, " Power Reduction on Embedded Systems Achieved by a Synchronous Finite State Machine Design Technique ", Computing System Engineering (SBESC). 2012 Brazilian Symposium on 5-7 Nov. 2012. pp 71-76.

[3] Peiyi Zhao, McNeely.J, Weidong Kuang , Nan Wang , Zhongfeng Wang, "Design of Sequential Elements for Low Power Clocking System", Very Large Scale Integration (VLSI) Systems, IEEE Transactions on May 2011.pp 914-918.

[4] Choudhury, Priyanka; Pradhan, Sambhu Nath . " An Approach for Low Power Design of Power Gated Finite State Machines Considering Partitioning and State Encoding Together ". Journal of Low Power Electronics August 2012. pp. 452-463

[5] Lindholm,C. "High frequency and low power semi-synchronous PFM state machine", Circuits and Systems(ISCAS), 2011 IEEE International Symposium

[6] D. P. B. Renaux, F. Pöttker, " Power Reduction in Embedded Systems Using a Design Methodology Based on Synchronous Finite State Machines", Embedded Systems: Design, Analysis and Verification IFIP Advances in Information and Communication Technology Vol. 403,2013, pp 61-72.