DOI: 10.35598/mcfpga.2021.003

Using Benchmark Tests for Research State Memory Encoding in Finite State Machine

Olga Zaichenko
ORCID 0000-0003-4936-2785
dept. Design and Operation of Electronic Devices
Kharkiv National University of Radioelectronics
Kharkov, Ukraine
olha.zaichenko@nure.ua

Nataliia Zaichenko
ORCID 0000-0001-9798-7136
dept. Microelectronics, Electronic Devices and Apliances
Kharkiv National University of Radioelectronics
Kharkov, Ukraine
nataliia.zaichenko@nure.ua

Pavlo Galkin
ORCID 0000-0002-0558-6448
dept. Design and Operation of Electronic Devices
Kharkiv National University of Radioelectronics
Kharkov, Ukraine
pavlo.halkin@nure.ua

Roman Tsekhmistro
ORCID 0000-0003-3628-3658
dept. Media Endineering and Information Radioelectronic System
Kharkiv National University of Radioelectronics
Kharkov, Ukraine
roman.tsektmistro@nure.ua

***Abstract*—The synthesizer will automatically assign the state memory codes based on the most effective use of the target technology (e.g., binary, gray code, one-hot) in FPGA. But exists alternative, when user by himself choose type of memory encoding. There was considered user defined state coding method for Quartus Altera.**

***Keywords*—FPGA, finite state machine, state memory, memory coding, synthesizer, one-hot, sequential, johnson, optimization, speed of performance**

I. Introduction

Finite state machines can be easily modeled using the behavioral constructs in FPGA. Within the VHDL state machine model, three processes are used to describe each of the functional blocks: state memory, next state logic, and output logic[1].

The model of the state memory of the FSM using a process describes the behavior of the D-Flip-Flops in the FSM that are holding the current state on their Q outputs. Each time there is a rising edge of the clock, the current state is updated with the next state value present on the D inputs of the D-Flip-Flops. This process must also model the reset condition. At all other times, the process will simply update current_state with next_state on every rising edge of the clock. The process model is very similar to the model of a D-Flip-Flop. This is as expected since this process will synthesize into one or more D-Flip-Flops to hold the current state. The sensitivity list contains only clock and reset, and assignments are only made to the signal current_state. The following syntax shows how to model the state memory of this FSM example.

The synthesizer will automatically assign the state codes based on the most effective use of the target technology (e.g., binary, gray code, one-hot). But exists alternative, when user by himself choose type of memory encoding.

The aim of the work is a comparative study of the possibilities of using the state coding methods to reduce increase the speed of the logic circuit of the finite state automaton.

II. Types of Memory Encoding

There are variants of memory encoding. 1. "One-hot." A separate trigger is used to encode each state. The number of triggers is equal to the number of states of the machine. At any given time, only one trigger can have a single value. To form the value of each trigger, a logical equation is used, in which the number of terms is equal to the number of transitions to the corresponding state. 2. "Sequential". The synthesizer finds long sequences of states in the machine, consisting of unconditional transitions, and encodes the states within them with consecutive binary codes of minimum sufficient bit size. As a result, the address inputs of the Look-Up Table (LUT) elements are not fed to the input signals of the machine, and only the current state code is fed, which usually has a small bit compared to the number of input signals. Sequential state coding provides more optimal filling of static memory cells of LUT elements and reduces the number of unused cells. 3. Johnson. State coding is performed using Johnson code. Each value of this code contains only one continuous sequence of single bits, and any two adjacent values in an ordered sequence of values differ by only one bit. Johnson's code is a cyclic code with an excess and reduces the number of electrical interference caused by the simultaneous switching of several bits of the register circuit. Thus, when using the Johnson code to encode the states of the machine, the number of triggers involved will be greater than in the case of sequential encoding. 4. "Gray." State coding is performed using Gray code, in which two adjacent values in an ordered sequence of values differ in the value of one binary digit, and the number of bits coincides with the number in the case of sequential coding. Like Johnson's code, Gray's code should be used to encode state chains, because each automatic transition in such a chain will be accompanied by a change of only one bit in the machine's memory register. 5. "Auto". The synthesizer chooses one of the above coding methods at its discretion based on the analysis of the VHDL description of the machine. The choice of coding method also depends on other settings (for example, on the leading optimization strategy - hardware costs or speed), but the generalized approach is as follows: if the machine contains a small number of states, one-hot coding is used; with an average number of states, the Johnson code is used; with a large number of states, the Gray code is used.

III. Using Benchmark Tests for Compare Speed of Performance for Different State Memory Encoding

It is known the input project is synthesized into a circuit that consists of logical elements of logical blocks in a FPGA. Quartus converts a system description in one of the hardware description languages (HDL from Hardware Description Language) into a set of microcircuit-independent functional and storage elements. After the synthesis is completed, information can be obtained on the number of LUT-elements and triggers required for its implementation, as well as the estimated maximum frequency of operation [2].

The difficulty lies in the fact that the same project can be placed in the FPGA in different ways, and there are millions of these ways. Some placement and routing is better, others are worse. The main criterion for the quality of the resulting system is the maximum frequency at which the project can operate with a given arrangement of elements and with a given routing of links. It is influenced by the length of the links between the blocks and the number of programmable switches between them. To use advanced settings that impact the synthesis of design it is necessary to access to settings. To click Assignments > Settings > Compiler Settings > Advanced Settings (Synthesis). The Optimization mode setting enables various combinations of these settings to achieve design goals.

```
.i 2
.o 1
.p 11
.s 4
-0 st0 st0 0
11 st0 st0 0
01 st0 st1 -
0- st1 st1 1
11 st1 st0 0
10 st1 st2 1
1- st2 st2 1
00 st2 st1 1
01 st2 st3 1
0- st3 st3 1
11 st3 st2 1
```

Fig. 1. Benchmarklion in KISS2 format [5]

As it turned out, not every program is suitable for the analysis of efficiency, for example, the vending machine [1] showed the same performance for all types of encoding. So we had to turn to special state machines, benchmarks. Benchmarks have their own names [2], for example, “lion”. Their number will reach more than 50 pieces. Initially, they are recorded in the KISS2 format (fig.1) The KISS2 format is a very popular text format for describing the nehavior of a control units. A KISS2 file is devided into two parts: header and a state transition table. The header contains generic parameters of control unit, i.e. the number of inputs, the number of outputs, the number of states and the number of terms.

```
.i <number of inputs>
.o <number of outputs>
.p <number of products>
.s <number of states used>
.r <reset state>
<input> <current-state > <next-state> <output>
.
.
.
<input> <current-state > <next-state> <output>
```

Fig. 2. Example of headers KISS2[4]

Due to application of TimeQuest Timing Analyzer from Tools Menu Quartus Altera[7] there was estimated time of performance for hybrid of finite state machine from[1] and [5,6]. The results from Report Fmax Summary are for Auto and One-Hot Encoding - 854.7MHz, for Sequential and Minimal Bits 1000 MHz, .Johnson and Gray encoding 1001 MHz, which partially coincide with results from [2].

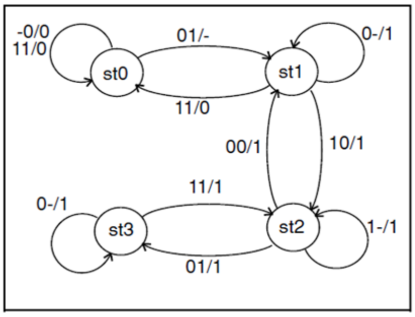

Fig. 3. Graph of the lion benchmark [6]

IV. Conclusion

One can use his own User-Encoded style, or select One-Hot, Minimal Bits, Gray, Johnson, Sequential, or Auto (Compiler-selected) encoding [3]. The best speed of performance has one-hot encoding.

References

[1] B. J. LaMeres, ”Introduction to Logic Circuits and Logic Design with VHDL,” – Springer, 2019, 503 p.

[2] A. A. Barkalov, I. Ya. Zeleneva, E. R. Tatolov. "Analysis of the efficiency of state coding methods in the synthesis of Mealy automata on FPGA,” Science of the Donetsk National Technical University, series of Problems of modeling and design automation, 2011, Vipusk 10 (197) pp. 1-6.(in Russian)

[3] Advanced Synthesis Settings Dialog Box. [Online]. Available: https://www.intel.com/content/www/us/en/programmable/quartushelp/17.0/mapIdTopics/mwh1465495270874.htm [Accessed: 10- Jun- 2021]

[4] A. Barkalov, L. Titarenko, M. Kolopienczyk, K. Mielcarek, G.Bazydlo, Logic synthesis for FPGA-based finite state machines. Springer, 2016.

[5] LGSynth93, International Workshop on logic synthesis benchmark suite (LGSynth93).TAR, Benchmarks test. [Online]. Available:http://www.cbl.ncsu.edu:16080/benchmarks/LGSynth93/LGSynth93.tar [Accessed: 10- Jun- 2021]

[6] H. Kubátová, “Finite state machine implementation in FPGAs”. In Design of Embedded Control Systems Springer, Boston, MA. , 2005, pp. 175-184

[7] Getting started with the TimeQuest Niming Analyzer [Online]. Available: https://www.youtube.com/watch?v=bFmTHLZ3DGs [Accessed: 10- Jun- 2021]