International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 09 | Sept - 2021 ISSN: 2582-3930

PERFORMANCE AND ANALYSIS OF PROGRAM COUNTER

SNEHA V L
Department of Electronics and communications
RV College of Engineering
Mysore Road Bangalore Karnataka 560059

Dr KIRAN V
Professor, Department of Electronics and communications
RV College of Engineering
Mysore Road Bangalore Karnataka 560059

Abstract **-** The paper presents the design of the program counter and design constraints are analysed. Two approaches namely, 1) finite state machine logic-based and 2) Behavioural model have been employed. The designs have been implemented and a comparison between the two design methodologies has been made. The finite state machine logic-based design uses flip flops and multiplexers, while the behavioural model design employs the incremented circuit using logic gates. The incremented based logic used here is carry lookahead adder because the delay in this adder is less when compared to other adders. The power consumed by the program counter designed using FSM-based logic is compared with incremented based design. The designs have been implemented using Cadence tools and simulated using 180nm technology files.

KEYWORDS: Program counter, finite state machine (FSM), cadence, incremented.

I. INTRODUCTION

The increase in the size of the circuit also facilitates complicate designs. These complex design increases its area which in turn result in increased power dissipation even while reducing the delay and enhancing the speed of operation of the complex circuits. Hence, the focus of designers is towards realizing a higher speed of operation for more complex circuits, and all the while attempting for low power operation potential in all the blocks of the design.[1]

The delivery of instructions is controlled by the active edge of the clock and its performance explains the executing speed of the program counter which points to the address of the successive instruction to be executed. Hence, the processor speed can be improved by reducing the time taken by the processor to point to the address of the instructions. Hence, the design of individual parts of design becomes an important factor while realizing the processor design meant to achieve low power and high-speed architectures.[2]

The schematic of program counter is created using cadence virtuoso tool that uses 180nm technology files which then simulates and synthesizes and provides netlist. The flow used here are implementation and analysis. In implementation flow, the schematic is generated in Virtuoso Schematic Editor and then assign schematic to layout which then performs physical design. Previously shifting to the physical design, synthesis is carried out using genus tool. From the synthesis, the constraints such as area, power, and quality of report are obtained. After obtaining the correctness of functionality with the help of netlist from synthesis the design is analysed.

This paper explains the design of the processor called the *program counter*. This paper provides two different approaches to the design of a program counter, namely, 1) with the Finite State Machine (FSM) based conventional program counter designed using flip-flops, and 2) with the use of combinational circuits which is an incremented circuit. Comparisons are made in terms of design constraints [3].

A counter is an example of a register. Usually, in the register each combination is stockpiled in the collection of flip-flops include the counter [4], which is the output pattern and is called as state of the counter. The sequence in which the states counts is mentioned as a counting sequence by using finite state machine [5]. The same design can be implemented by incremented based logic that uses adder circuit [6][7][8]. The adder circuit used here is carry lookahead adder because of its less delay when compared to other adders and when area and complexity is not concerned[9]. The proposed synchronous 3-bit program counter is implemented using Cadence EDA tool and this tool delivers sophisticated structures and design constraints. The

II PROGRAM COUNTER:

A program counter is a CPU register that presents in processor of computer which contain the address of the next instruction that is going to execute from the memory. Here the program counter used is 3-bit up counter which counts the sequence till the clock is running. The program counter is designed in two ways 1) by using finite state machine 2) incremented based design.

In finite state machine, the counting happens from one state to another state and sine the designed program counter is 3-bits, all 8 input combinations are considered as 8 states. The functionality is defined with the help of behavioural modelling which contains procedural statements. For a given input combination or state, the counting starts in sequence.

In incremented based design, the structure of 3-bit counter is designed and schematic is generated in cadence. From the structure, the functionality is defined with the help of behavioural modelling. The netlist is obtained after performing synthesis.

III FINITE STATE MACHINE BASED DESIGN:

In this section, Verilog code for FSM based program counter is designed to describe the functionality on up counting of states.

The program counter initiates by resetting the program counter to 0.

1) The input a is checked, and when it's found low, the FSM drives to the idle state, which disappears the output, setting the Program counter to 0.

2) If input is high, the program counter checks for active edge of clock. When there is active edge of the clock state increments.

In this manner the program counter keeps on checking for input signal and active edge of the clock.

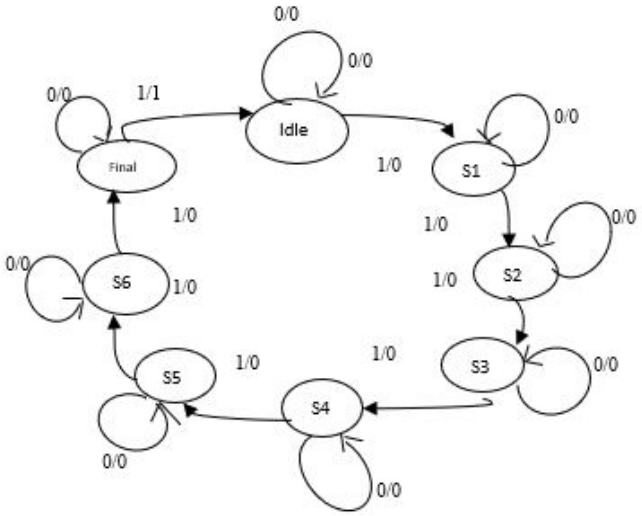

Fig.1: State diagram of FSM based program counter

The Fig.1 shown above describes the counting sequence from initial position based on the input given and up counting till last input combination and returns back to initial position.

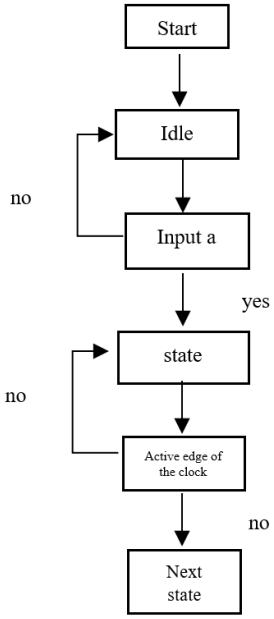

Fig.2: ASM flow chart of program counter

The Fig.2 shown above describes the algorithmic state machine of the designed 3-bit program counter

IV INCREMENTED BASED PROGRAM COUNTER:

In this section the program counter is designed using logic gates. The incremented based design usually contains D-flipflops and multiplexers for selecting lines. This idea of design reduces the speed and increases the operating frequency. And this operation is similar to that of FSM based design.

In this paper, design is proposed by employing adder circuits and therefore adder such as carry look ahead adder (CLA) is used for minimizing delay at the cost of increased complexity and area.

The full custom design of 3-bit CMOS based logic is designed as shown in Fig.3 and the adder circuit is derived from following equations:

B0=AO'
B1=A1^A0
C1=A1&A0
B2=C1^A2
C0=A2&C1

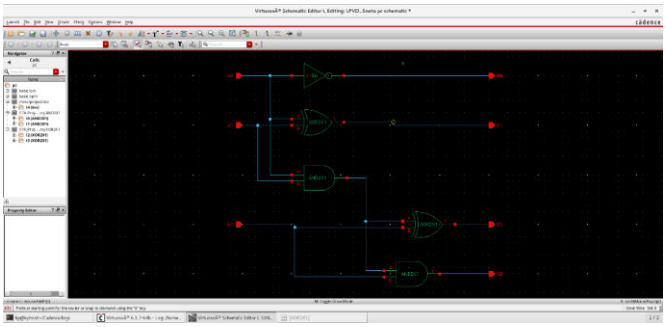

Fig.3: Adder circuit of incremented based program counter

The Fig.3 shown above describes incremented based logic circuit of 3-bit program counter. This is a adder circuit and the type of adder used is carry lookahead adder that has less delay usually when compared to other adders when area is not a concerned.

V SIMULATION AND SYNTHESIS RESULTS:

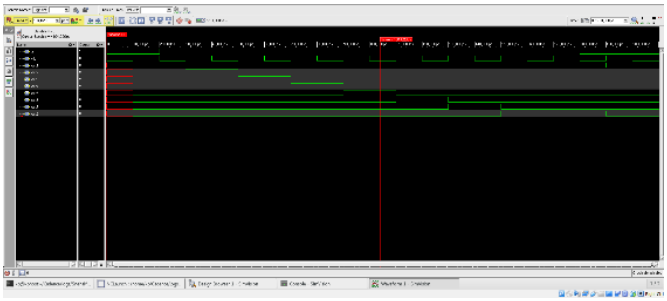

Fig.4: Simulation result of FSM based program counter

The behavioural modelling is generated for FSM based program counter and Fig.4 shows the simulation result obtained.
As we can observe from the simulation result that counting starts from 000 and increments one state per time for the active edge of the clock.

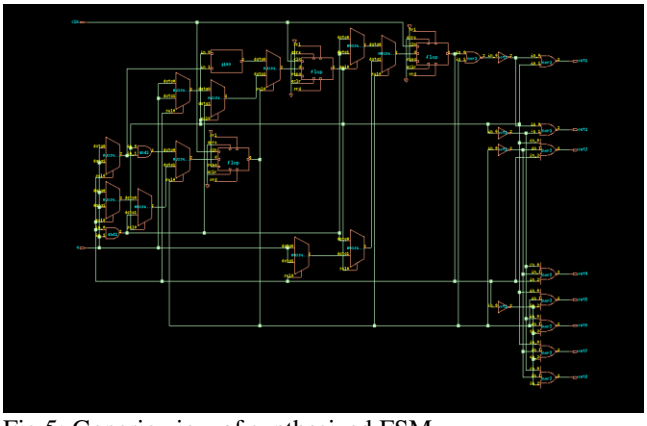

Fig.5: Generic view of synthesized FSM.

The Fig.5 shows the generic view of synthesized FSM based 3-bit program counter which contains D flipflop and muxes

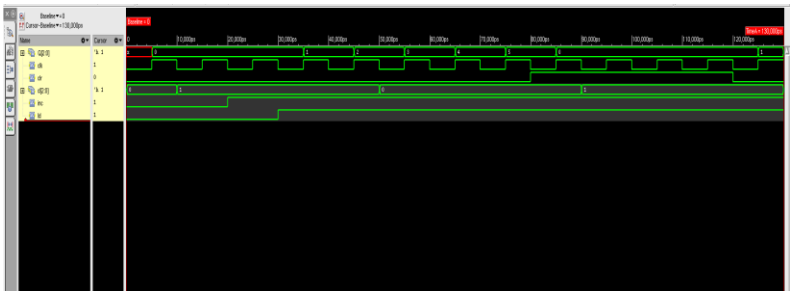

Fig.6: Simulation result of incremented based program counter

The Fig.6 shows the synthesized waveform of incremented based logic circuit of 3-bit program counter that contains combinational gates. This design also depicts the design of adder circuit

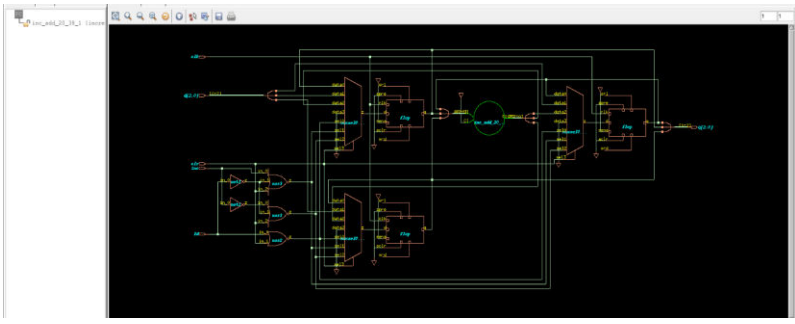

Fig.7: generic view of synthesized program counter

The Fig.7 shows the generic view of synthesized form of incremented based logic of 3-bit program counter.

RESULTS

TABLE I: Timing slack comparison of two ways designed based program counter

Instance	Timing slack
FSM based program counter	38791ps
Incremented based program counter	38499ps

TABLE II: power comparison of two ways designed based program counter

Instance	Cells	Leakage power (nW)	Dynamic power (nW)	Total (nW)
FSM based PC	28	1.109	67319.036	67320.144
Incremented based PC	28	1.275	61486.807	61488.082

VI CONCLUSION:

The program counter is executed using the anticipated finite state machine-based design and the incremented based design. The results from two methods are compared and from the comparison it is observed as follows

1)The delay is less in incremented based design when compared to FSM based design.

2)The total power consumed by the incremented based design is less when compared with FSM based design even though both the design consumes same number of transistors.

The inferences that have been derived out of the comparisons are as follows. The incremented based program counter is advantageous when compared in terms of area, timing and power. But FSM based program counter almost gives accuracy in design. Hence, for the design of high-performance processors, the program counters design should be selected thus liable on the conditions essential for the designing of the processors

REFERENCES

[1] Matrosova, A., & Ostanin, S. (2000, July). Self-checking FSM design with observing only FSM outputs. In *Proceedings 6th IEEE International On-Line Testing Workshop (Cat. No. PR00646)* (pp. 153-154). IEEE.

[2] Divya, M., Belgudri, R., & Bhaaskaran, V. K. (2014, September). Design and analysis of program counter using finite state machine and incrementer based logic. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 581-587). IEEE.

[3] Majumder, A., Singh, P. L., Chowdhury, B., & Kumar, V. (2015, October). Cost efficient realization & synthesis of reversible presettable program counter for processor. In *2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)* (pp. 614-619). IEEE.

[4] Radha, S., Krishna, R. H., Pandi, N. P., Varghese, S., & Nagabushanam, P. (2018, March). Floor planning of 16 bit counter design for health care applications using 180nm technology in cadence tool. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 484-487). IEEE.

[5] Veeramachaneni, S., Avinash, L., & Srinivas, M. B. (2007, August). A novel high-speed binary and gray incrementer/decrementer for an address generation unit. In *2007 International Conference on Industrial and Information Systems* (pp. 427-430). IEEE.

[6] Sakthikumaran, S., Salivahanan, S., Bhaaskaran, V. K., & Kavinilavu, V. (2010). A Very Fast and Low Power Incrementer and Decrementer Circuits. *International Journal of Computer Communication and Information System (IJCCIS) Vol2*, (1-2011), 200-203.

[7] Huang, C. H., Wang, J. S., & Huang, Y. C. (2001, May). A high-speed CMOS incrementer/decrementer. In *ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No. 01CH37196)* (Vol. 4, pp. 88-91). IEEE.

[8] Parandeh-Afshar, H., Afzali-Kusha, A., & Khakifirouz, A. (2006, December). A Very Fast and Low Power Pseudo-Incrementer for Address Bus Encoder/Decoder. In *2006 International Conference on Microelectronics* (pp. 91-94). IEEE.

[9] Rawat, K., Darwish, T., & Bayoumi, M. (2002, August). A low power and reduced area carry select adder. In *The 2002 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002.* (Vol. 1, pp. I-467). IEEE.