http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 5 (1), 198-203, 2006
ISSN 1812-5638

Serial ALU Simulation with Timing and Signal Constraints

[1,2]Muhammad Kamran, [1]Shi Feng and [3]Sana Qureshi
[1]School of Information Science and Technology,
Beijing Institute of Technology, Beijing 100081, China
[2]Department of Electrical Engineering, University of Engineering and Technology,
Lahore-54890, Pakistan
[3]COMSATS Institute of Information and Technology, Lahore, Pakistan

Abstract: This study presents mathematical model and optimization in switching actions using timing and signal constraints while performing arithmetic and logical operations in serially designed ALU. Signal and timing constraints are inevitable to describe in VLSI circuit design in order to get perfect synthesis, post simulation and implementation on FPGA. For the simulation and verification of our idea, VHDL programming and MODEL SIM tool is utilized. Our proposed study not only corresponds to different arithmetic and logical operations but also depicts the concepts of serial to parallel data conversion through FSM.

Key words: Mathematical modeling, data conversion, FSM, time constraint, simulation

INTRODUCTION

Serial ALU designed shows optimization in components utilization, in switching actions to justify its use in processors designed for image processing task as it causes not only area reduction but also power dissipation is minimized. Serial ALU with finite state machine is introduced by Kamran and Feng[1]. The application of serial operations in digital system is to develop high performance Field Programmable VLSI Processor (FPVLSI). Serial ALU is also used as a programmable array for sequence analysis[2]. Although serial operation makes the process slow but reduces the number of components and power consumption in design. Our approach is to introduce a new method to control all arithmetic and logic manipulations by appending Finite state Machine (FSM) after careful investigation regarding timing cycles and control signal constraints. Moreover, serial ALU is planned to be used in 3DWT processor to achieve some encouraging results for future research. A digital circuit is generally defined as to contain data flow part and control flow part. In data flow path part the functional units include ALU, buses for interconnection, registers, multipliers etc. There is library of functional units, out of which most appropriate allocation is made to get feasible and optimal solution. That is selection is done to reduce the time delay and the area occupied by the components. There are different issues in HLS to be given attention like stability of the system, power consumption, time delay and the area occupied. Many researchers have floated their ideas and got considerably positive results form their research for the optimization of the system. FSM is also designed in ASIC design and for FPGA for controlling data flow process to optimize power dissipation[3]. Finite state machine may be clock operated or clock less, depending upon the designer and system requirement. Present study was carried on asynchronous FSM after taking care of timing hazards and glitches in the system. These hazards are eliminated successfully by Kenneth and David[4,5].

MATHEMATICAL MODEL OF SERIAL ALU

Mathematical model starts with the definition of input and out put vectors, to be applied to device are shown in block diagram, as shown in Fig. 1[1]. Mathematical model is considered as first step to design any digital system which can be further verified by using hard descriptive language and helping simulation tools.

Considering Fig. 1, we can interpret following definitions to develop and understand mathematical model of our design already proposed by Kamran and Feng[1].

I, J and X are input vectors applied to ALU.

$$\text{Input vectors; } I = \begin{bmatrix} i_0 \\ i_1 \\ i_2 \\ i_3 \end{bmatrix}, J = \begin{bmatrix} j_0 \\ j_1 \\ j_2 \end{bmatrix}, X = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix};$$

Corresponding Author: Muhammad Kamran, Room No. 931, Central Teaching Building, Beijing Institute of Technology, Baishiqio Road No. 7, Haidian District, Beijing-100081, China

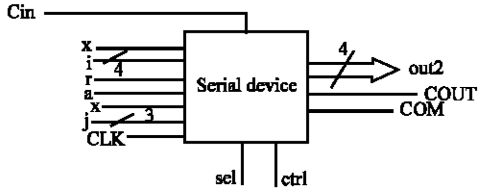

Fig. 1: Block diagram of serial arithmetic and logic device

And "r" represents reset input for restarting FSM while "a" is the input to full adder used for arithmetic operations. OUT2 is 4 bit out put of FSM carried by a register with COM =0 or 1; when COM=0 means operation is not yet completed and FSM is in process and if COM is equals to 1, indicates FSM has finished its operation and new data can be entertained. COUT depicts the possible carry out during arithmetic operation; there are input lines, n selection lines and U single output in basic definition of Multiplexer. μ.

$$\Rightarrow \mu(2^n;n;U(1));$$

μ1 and μ2 are the definition of our multiplexers while μ3 is the definition of 3rd multiplexer respectively used in our design.

$$\mu 1(i0, i1, i2, i3; s0, s1; U2) = \begin{cases} u2 = i0: \text{when } s0 = s1 = '0'; \\ u2 = i1: \text{when } s0 = '0' \& s1 = '1'; \\ u2 = i2: \text{when } s0 = '1' \& s1 = '0'; \\ u2 = i3: \text{when } s0 = '1' \& s1 = '1'; \end{cases}$$

$$\mu 2(d0, d1, d2, d3; s0, s1; U3) = \begin{cases} u3 = d0: \text{when } s0 = s1 = '0'; \\ u3 = d1; \text{when } s0 = '0' \& s1 = '1'; \\ u3 = d2; \text{when } s0 = '1' \& = '0'; \\ u3 = d3; \text{when } s0 = '1' \& s1 = '1'; \end{cases}$$

Where, d (i) are the internal signals in our design applied as inputs to multiplexer 2.
Similarly, multiplexer 3 is defined as follows;

$$\mu(u2, u3; ctrl; z) = \begin{cases} u2 \\ u3 \end{cases} \text{for ctrl} = '0' \text{ or } '1' \text{ respectively}$$

FSM is five tuple and can be defined mathematically; FSM: A = (Q, ∑, δ, q0, F), while other signals and units are described in the same pattern z = F(μ3(ctrl, FA, LU)); is the input to FSM transmitted through Multiplexer 3 carrying the information of Full adder or Logic unit in the design. The definition of remaining components is as follows;

$$FA = f1(Cin, \mu, i); \&$$
$$LU = f2(xi, j);$$
$$FSM = f3(Q, z, S0, S5, step(clk));$$

Clock controls the state flow.

Although parallel ALU operation will be faster than our serial device but with the increase in input vectors, increment in number of components (FA) is observed and system becomes uneconomical with respect to area and more power is consumed. While in serial ALU single FA is utilized irrespective of the number of bits to be applied on the input of the circuit. Presently, our objective is to perform as many operations with serial device as we do with parallel device to obtain most optimized design. Moreover, with the reduction in the number of components, we can manage to complete our design with minimum power consumption[1,6].

TIMING CONSTRAINTS

Figure 1 is the block diagram of serial ALU giving interpretation of mathematical model. Serial ALU is not only optimized with respect to cost and area but also with respect to the power consumption which occurs due to flow of current or due to switching action in different parts of the circuit. Using serial input data, the circuit is considerably optimized especially at large number of input vectors. Our proposed paper is completed in four steps; mathematical model, minimum number of components, serial to parallel data conversion through FSM and timing constraint to activate correct signal at right time.

$$P_{switching} = CfVdd^2 \qquad (1)$$

Equation 1 is the generalized formula for power dissipation in any circuit or device.
For arithmetic circuit scheduling only one full adder is used for (i) Transfer (ii) Increment (iii) Addition (iv) subtraction operations. The constraint is to keep proper delay between start and end of process as we are utilizing single hardware resource for the purpose. We utilized data flow graph (DFG) by Kamran and Feng[1] to present all arithmetic and logic operations with their scheduling and allocation. Each operation/node of DFG is of 3 tuple (D, T, R). D is the delay prior to start of next operation, T is the time taken for the completion of operation and R is the functional unit selected. Every node OPi in DFG represents an operation which is supposed to be executed at time Ti finished after an operation delay Di. At all nodes calculations to be done before specified time which s tarted after a zero moment;

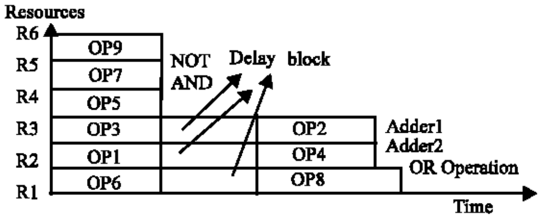

Fig. 2: Resource allocation of our arithmetic and logic unit example

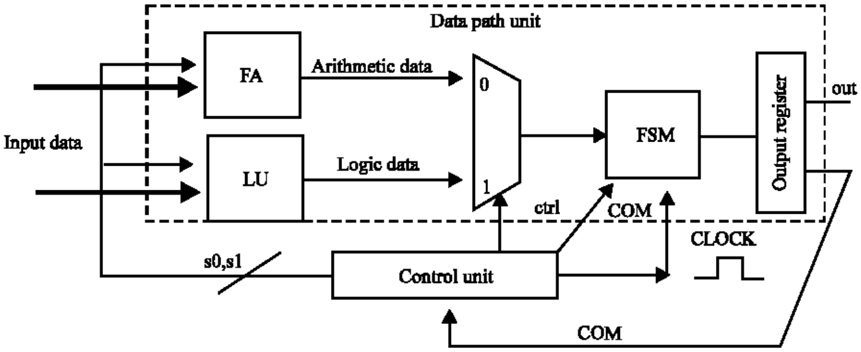

Fig. 3: Data flow path and control signal

$$T_{end} \ge T_i + D_i \quad \& \quad T_{zero} \le T_i \qquad (2)$$

If an operation OPk depends upon OPi then condition should be fulfilled.

$$T_{zero} \le T_i + D_i \le T_3 \quad \& \quad T_4 + D_4 \le T_{end};$$
$$T_1 + D_1 \le T_4 \quad \& \quad T_3 + D_3 \le T_4$$

The constraint is given by the fact that one resource can be used for one operation at a time. For our optimally designed ALU with 9 specified operations 6 resources are needed to complete all operations as shown in Fig. 2[1]. Delay block represents the time taken by the functional block to be reused for the same operation for example same adder is utilized twice for the completion of operation in resource R2 and R3. Moreover, if time taken by any operation is T1 and this operation or functional unit is to be utilized again, then considerable time should be given between FU to be used again. This is elaborated in Eq. 2. This is the responsibility of the Scheduling operation to use proper functional at proper time. This problem of High Level Synthesis (HLS) is Non deterministic Polynomial (NP-hard) and special algorithms are proposed to get most appropriate scheduling, allocation and binding.

CONTROL SIGNAL CONSTRAINTS

The designed device has multiple operations to be performed. The sequence of operation is already discussed by Kamran and Feng[1]. In continuation to our previous work, a new idea of control circuit and signal constraints is added to become more precise about our design and the time to execute the signal operation.

Figure 3 is the data flow and control signal diagram deduced from Fig. 4 which is serial ALU proposed and tested earlier by Kamran and Feng[1]. S0 and S1 are the control signals applied to the Multiplexer 1 and Multiplexer 2 to select arithmetic and logic operation at the same time. Input data will be processed after these multiplexers and will be processed through Full Adder (FA) and Logic Unit (LU) and stop temporarily prior to another Multiplexer in the design. Control unit will come in action again to deliver a signal "ctrl" to select one of the data to come out and moved to FSM. FSM will process the serial data one bit/positive edge of clock with time period of 20 ns in our case. The best thing in utilizing serial circuit for data manipulation is that the number of components in the circuit will remain same independent of the length of data. Despite of the fact of spending more time in this circuit, we can manage to complete our operation with out making circuit complex with respect to the components. "ctrl" signal plays a vital role in order

Table 1: Logic analysis of designed circuit with corresponding output

Ctrl (Control)	Carry_in	S1	S0	Output
0	0	0	0	a+x (Addition)
0	0	0	1	a+$\overline{x}$(Subtract with borrow)
0	0	1	0	a+1 (Increment)
0	0	1	1	a (Transfer a)
0	1	0	0	a+x+1 (Addition with carry)
0	1	0	1	a+ $\overline{x}$+1 (Subtrection)
0	1	1	0	a-1 (Decrement)
0	1	1	1	a+1 (Increment)
1	X	0	0	a ^ b (AND)
1	X	0	0	a || b (OR)
1	X	1	0	a XOR b
1	x	1	1	NOT b

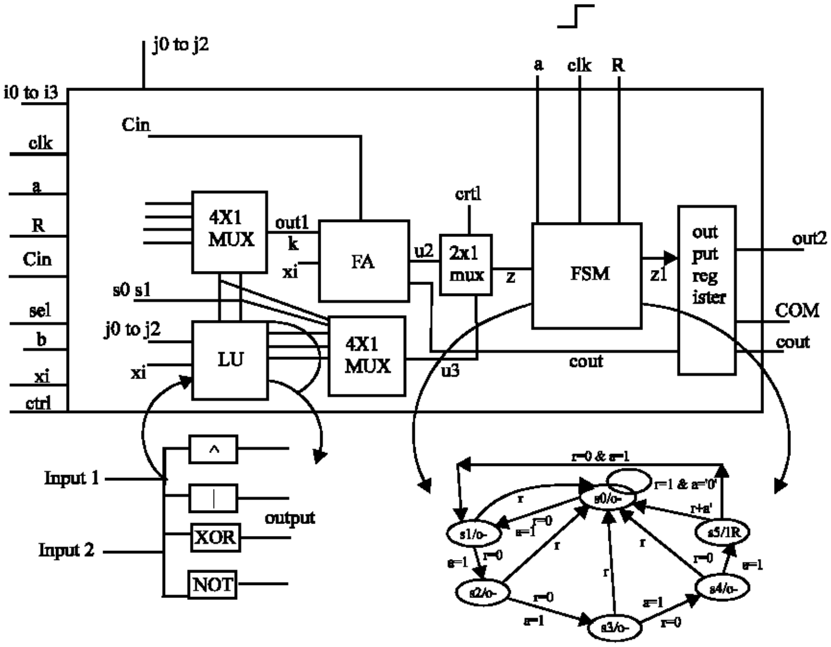

Fig. 4: Circuit diagram of serial arithmetic and logic device with LU functions and FSM state diagram

to get out put with co precision. For 4 clock cycles ctrl is kept='0' as to get arithmetically operated data, while it is kept='1' for next four clock cycles for output register to contain logically operated data. The COM signal will automatically be '1' after four clock cycles and after some certain delay for next data processing its value will be '0'. Transition of COM control signal enables us to inform regarding new data processing is started or not.

We tested our circuit for two 4 bit data inputs. FSM not only synchronizes the data transmitting processes but also performs serial to parallel data conversion. The FSM will require 5 states including S0, the starting state with 4 clock cycles. The clock cycles and states are incremented with the increase in the length of input data. The control signal COM indicates the termination of FSM operation and when its value='1', shows that all data is processed and system is reset for new data values (Fig. 4).

EXPERIMENTAL RESULTS

We tested our circuit for two numbers of 4 bits each. As discussed earlier the component complexity will not increase even if we increase the data input size. So the circuit can be tested with out any component change for large data processing elements. The circuit is designed to verify. Table 1 representing arithmetic and logical operations. Extended arrow in Fig. 5 indicates signal COM=1, so data operation is finished and output register contain the arithmetic or logic operation out put. It is proposed here, to design test bench to first get arithmetic out put for 4 clock cycles and of course with the application of appropriate control signals. After the termination of arithmetic operations with output register with 4 bit value and COM signal high, we apply 4 clocks more after "RESET" the system and perform 4 logical operations.

Table 2: Listing of FSM data transfer (VHDL result)

Time	Delta	Clock	"a"	"x"	"r"	Out2	COM	Reg	State
0 ns	0	u	u	u	u	u	u	u	S0
10 ns	0	1	0	1	1	u	0	u	S0
30 ns	1	1	1	0	0	???1	0	????1	S1
50 ns	1	1	1	1	0	??11	0	??11	S2
70 ns	1	1	1	0	0	?011	0	?011	S3
90 ns	1	1	0	1	0	1111	0	1111	S4
110 ns	1	1	0	1	0	1111	1	1111	S5

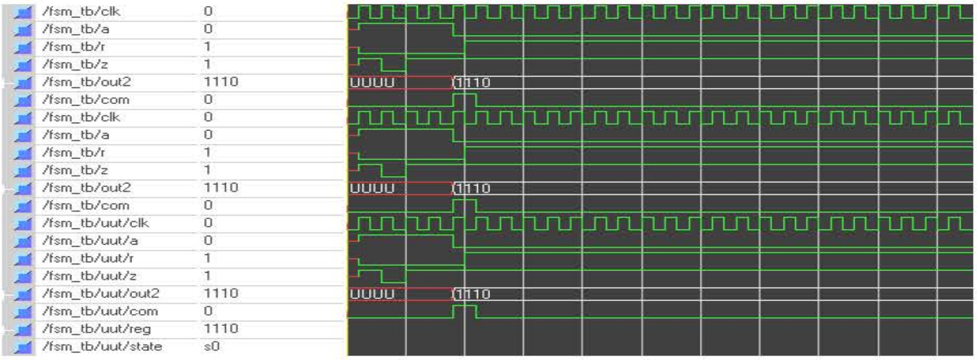

Fig. 5: Model SIM result for FSM operation

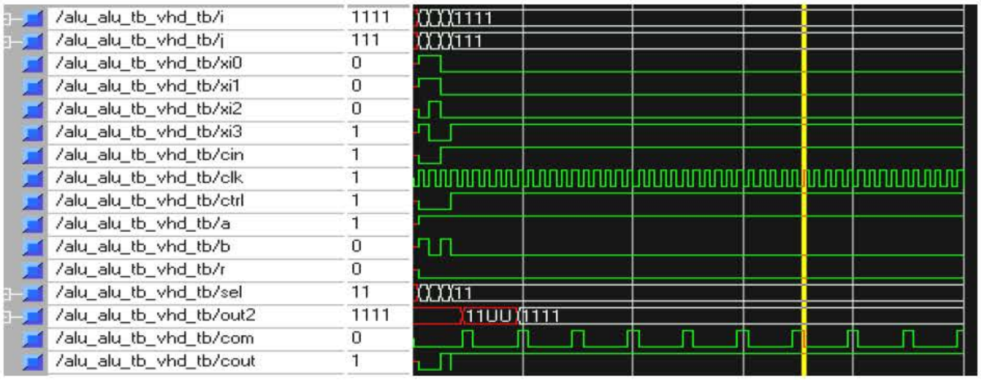

Fig. 6: Simulation result of serial ALU

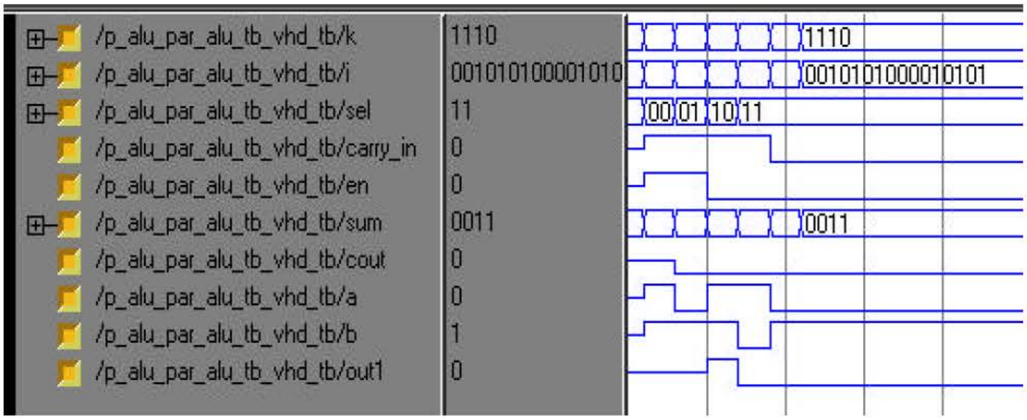
/p_alu_par_alu_tb_vhd_tb/k 1110
/p_alu_par_alu_tb_vhd_tb/i 001010100001010
/p_alu_par_alu_tb_vhd_tb/sel 11
/p_alu_par_alu_tb_vhd_tb/carry_in 0
/p_alu_par_alu_tb_vhd_tb/en 0
/p_alu_par_alu_tb_vhd_tb/sum 0011
/p_alu_par_alu_tb_vhd_tb/cout 0
/p_alu_par_alu_tb_vhd_tb/a 0
/p_alu_par_alu_tb_vhd_tb/b 1
/p_alu_par_alu_tb_vhd_tb/out1 0
1110
0010101000010101
00
01
10
11
0011

Fig. 7: Simulation results of parallel circuit

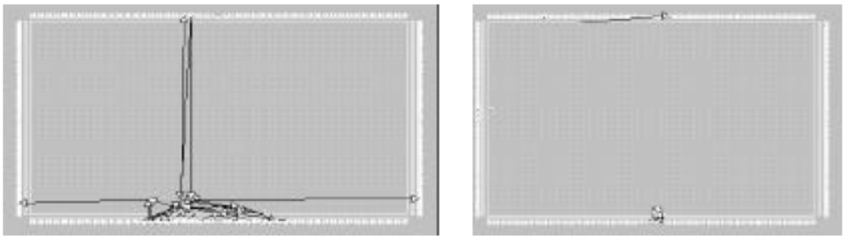

Fig. 8: Placements diagram showing difference of connection completely for parallel and serial ALU

Figure 5 presents the MODEL SIM simulation result of FSM serial clock based operation with the contents of out put register with status of COM 0 to 1 to 0, between two operations. While Fig. 6 is also showing the 4 clock cycle results of serial ALU. Kamran and Feng[1], proposed most optimized parallel ALU with the selection of digital IC's from TTL data sheets, the simulation results are verifying the result Table 1 as shown in Fig. 7, but complexity comparison between Parallel and serial ALU is very obvious on PAR diagram for FPGA standard device, shown in Fig. 8. Table 2 is the FSM operation table with state transition, output register status and other signal obtained during ACTIVE HDL simulation till the completion of one process for four clock pulses.

CONCLUSION AND FUTURE WORK

We successfully implemented an important part of processing element with minimum complexity regarding component selection and with minimum power consumption as this circuit has minimum switching actions as suggestedd by Rabaey and Pedram[6]. The data input variables may be extended with same circuit with compromise on operating time.

- For Future Clock less logic can be implemented for FSM operation in which state change will take place with operation required and status of bit in adjacent state[4,5].
- The circuit will be used in image transfer and compression processing elements.

REFERENCES

1. Kamran, M. and S. Feng, 2005. Optimization in Component Selection, Switching and Process Control of Serial Arithmetic and Logical Device using FSM. ICYCS05 Conference Proceedings, September, Beijing, China.
2. Hirschberg, J.D., R. Hughey, K. Karplus and D. Speck, 1996. Kestrel: A Programmable Array for sequence Analysis: IEEE, August.
3. Sutter, G., E. Todorovich, S. Lopez-Buedo and E. Boemo, 2002. Low-power FSMs in FPGA: Encoding Alternatives. Springer-Verlag Berlin Heidelberg, pp: 363-370.
4. Kenneth, Y.Y. and L.D. David, 1999a. Automatic Synthesis of Extended Burst Mode Circuits: Part-1(Hazard Free Implementation). IEEE Transaction on CAD of Integrated Circuits and Systems, Vol. 18, No.2, February.
5. Kenneth, Y.Y. and L.D. David, 1999b. Automatic Synthesis of Extended Burst Mode Circuits: Part-2(Automatic synthesis). IEEE transaction On CAD of Integrated circuits and systems, Vol. 18, No.2, February.
6. Rabaey, J.M. and M. Pedram, 1996. Low Power Design Methodologies. Kluwer Academic Publisher.