FPGA IMPLEMENTATION OF HIGH SPEED AND LOW POWER SPECULATIVE ADDER

V.Aparajita[1], N. Krishna kumari[2], S. Ramesh[3]

[1, 2] Student, Dept. of Electronics and Communication Engineering, Bapatla Engineering College, Andhra Pradesh, India
[3] Asst. Prof, Dept. of Electronics and Communication Engineering, Bapatla Engineering College, Andhra Pradesh, India

---***---

Abstract — *High speed adders are highly desirable in present day scenario where power also plays equal role. This paper displays carry-lookahead adder (CLA) based configuration of the contemporary inexact-speculative adder (ISA) which is further fine-grain pipelined that is addition of registers along its critical path and thereby, upgrading the process of addition by decreasing the delay of operation and enhancing the frequency of operation. The registers we used are nothing but D-Flip-flops which are clock gated in order to reduce the power consumption. Functional verification and hardware implementation for various configurations of the suggested ISA is to be carried out on field-programmable gate array(FPGA) platform.*

The synthesis and post layout simulation of the proposed ISA is carried out in FPGA using vivado hls for power analysis. Implementation of pipelining has reduced the delay up to 6ns compared to non-pipelined architecture and it has also reduced power up to 4w.

Key Words: **Inexact speculative adder, Carry lookahead adder, pipelining, Field programmable gate array, very-large scale-integration.**

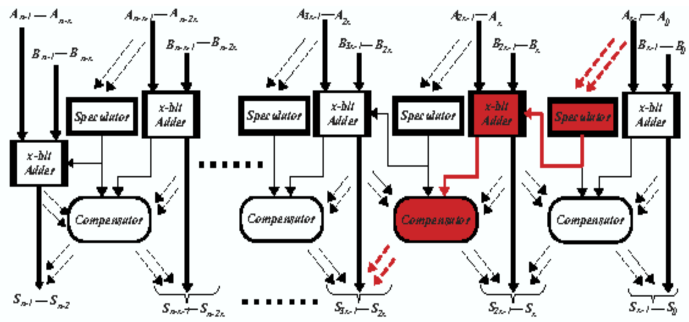
1. INTRODUCTION

Speed is one of the important factor along with the utilizing of less power for the adders in the present day scenario rather than the exact result. For this we prefer highly optimized adders which require less delay and low power and this paper presents an adder of exactly this type.

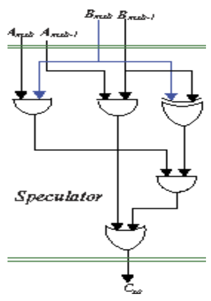
With acceptable degradation in accuracy and performance it is possible to design high speed and low power adder using speculation technique[4]. Accuracy is the major compromise to be done to improve power and speed by speculation. Thereby these adders are referred to as Inexact speculative adders. Various adders are reported in the references from [5]-[9] but accuracy is considered as the major constraint in these adders and concentrated more on improving accuracy of the results. However there is a chance to improve the speed of the adders by retaining a minimum error in the result. So our contributions in this work as follows:

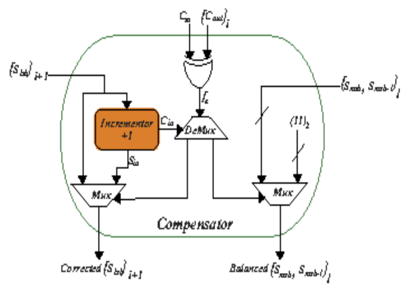
(1) Design of carry lookahead adder based inexact speculative adder.

(2) Thereafter this adder is fine grain pipelined to reduce the critical path delay and also enhancing the speed of operation. FPGA implementation of 8, 16 and 32 bit versions of proposed and suggested architectures are carried out, obtained the post, place and route results and compared. clock signal fed to various stages of the pipelined ISA-architecture has been gated to reduce the power consumption. Synthesis and post-layout simulation of the clock gated ISA has been performed in FPGA using vivado hls.



(a)





(b) (c)

Fig 1: Basic block diagram of *n*-bit conventional inexact-speculation adder (ISA). (b) Gate-level circuit representation of speculator block. (c) Digital architecture of compensator block.

2. INEXACT SPECULATIVE ADDER

In the proposed architecture, we have segregated the *n*-bit input into 4-bit blocks (i.e., the value of *x* = 4 in Fig. 1) and each of these blocks is fed as operands to the *x*-bit adder. Unlike the conventional ISA architecture, the adder unit has been replaced with 4-bit CLA to further enhance the speed of operation. Explanation of different blocks of the adder are as follows:

a) Adder and Speculator blocks: Consider two n-bit operands are $A=\{A_0, A_1, A_2, ----A_{n-1}\}$ and $B=\{B_0, B_1, B_2----B_{n-1}\}$. whereas the sum, carry-in and carry-out are expressed as $S=\{S_0, S_1, S_2----S_{n-1}\}$, C_{in} and C_{out} respectively. Speculator block is based on CLA logic to speculate the output carry for each 4-bit adder block. Speculation is carried out for last two msb bits of each block. Subsequently, the input carry for each speculator block is 0 or 1 which introduces positive or negative errors respectively. The output carry, which is denoted as C_{s0} from each speculator block is fed as an input carry for the adder block succeeding it. So each 4-bit adder block need not wait for the input carry from the preceding 4- bit adder block. Instead, all such adder blocks perform simultaneous additions on receiving input carries from the concerned speculator blocks. Speculator block computes carry based on the equation shown below:

$$P_i = A_i \oplus B_i$$
$$G_i = A_i \cdot B_i$$
$$C_{i+1} = G_i + A_i \cdot B_i$$

This block is situated along the critical path of ISA architecture; however, it doesn't produce much delay as it computes the carry for only two bits. On the other hand, adder block performs addition of 4-bit input blocks using CLA logic based on

$$S_i = A_i \oplus B_i$$

Here, the local sum obtained from each adder block is not the exact output because the addition has been performed using speculated carry inputs. Correction or Balancing of such sum value is carried out by the compensator block.

b)Compensator Block: Fig.1(c) shows the digital architecture for compensator block used in the ISA adder. This block compares carry from each 4-bit adder block with the corresponding speculated carry using a XOR gate. Thereafter, the output from XOR gate generates an error flag *(f_e)* triggers the activation of one of the two compensation techniques: error correction or reduction. If the XOR-gate output is '0' then the local sum is directly passed to the final output. Similarly, if the XOR gate gives '1' then this indicates that an error has occurred which can be either positive or negative. A positive error indicates a speculation of '0' instead of '1' and, hence, induces too low sum and negative error indicates speculation of '1' instead of '0' which induces too high sum. The components of compensation block involved in the overall critical path of ISA are the XOR gate, de-multiplexer and multiplexer.

3. FINE-GRAIN PIPELINED ARCHITECTURE

In the conventional ISA architecture, let us assume that the combinational delay of 4-bit adder, speculator and compensator blocks to be $\partial_{4b-adder}$, ∂_{spec} and ∂_{comp} respectively. In this architecture, carry in is speculated for each 4-bit adder block and based on this adder block calculates the local sum. Thereafter, the error speculation is detected by comparing speculated carry in and prior carry out from 4-bit adder. Subsequently, compensator block performs the correction and balancing operation. Thus, the critical path of the conventional ISA architecture includes delays of adder of the i^{th} instant and the speculator plus compensator delays of $(i+1)^{th}$ instant and the equation is given as

$$\partial_{critical} = (\partial_{4b-adder})_i + (\partial_{spec})_{i+1} + (\partial_{comp})_{i+1}$$

The detailed version of the critical path is given by

$$\partial_{critical} = (\partial_{4b-adder})_i + (\partial_{xor+or+and})_{i+1} + (\partial_{xor+demux+mux})_{i+1}$$

Where ∂_{xor}, ∂_{and}, ∂_{or}, ∂_{demux}, ∂_{mux} are the combinational delays of logical AND, OR and XOR respectively. The speculator, compensator, 4-bit adder and overall design is feed forward VLSI architecture. If we carefully analyze and pipeline these blocks then we may reduce the critical path delay and gain fast result.

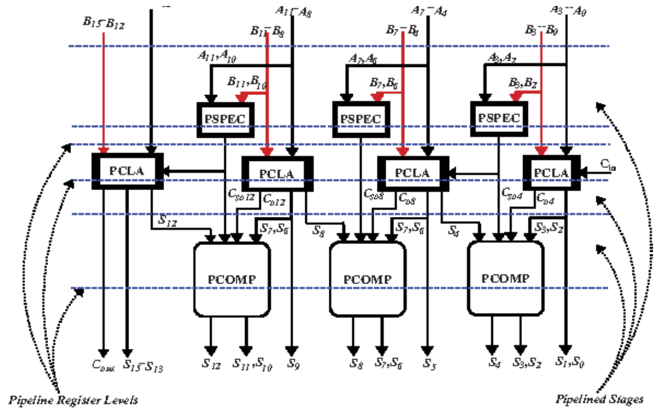


Fig 2: Pipelined VLSI architecture of the proposed ISA for n=16bits and x=4bits with five pipeline stages.

Pipelining process here is explained using n=16bit ISA architecture. Even the value of n increases the critical path is unaffected because the value of x is always a 4-bit and hence the adder, speculator and compensator blocks remain unchanged. In the above fig 2 the proposed architecture is replaced by the pipelined speculator (PSPEC), pipelined compensator (PCOMP) and the pipelined CLA (PCLA). Sub blocks PSEPEC, PCOMP, PCLA contain the pipelined stages. Overall architecture of the ISA adder has been designed with five pipelined stages and there are six levels of registers included in this design as shown in fig 2.In this case the number of pipeline stages remains constant and on increasing the width of operands then the bit widths of the operands, retaining the same critical path delay.

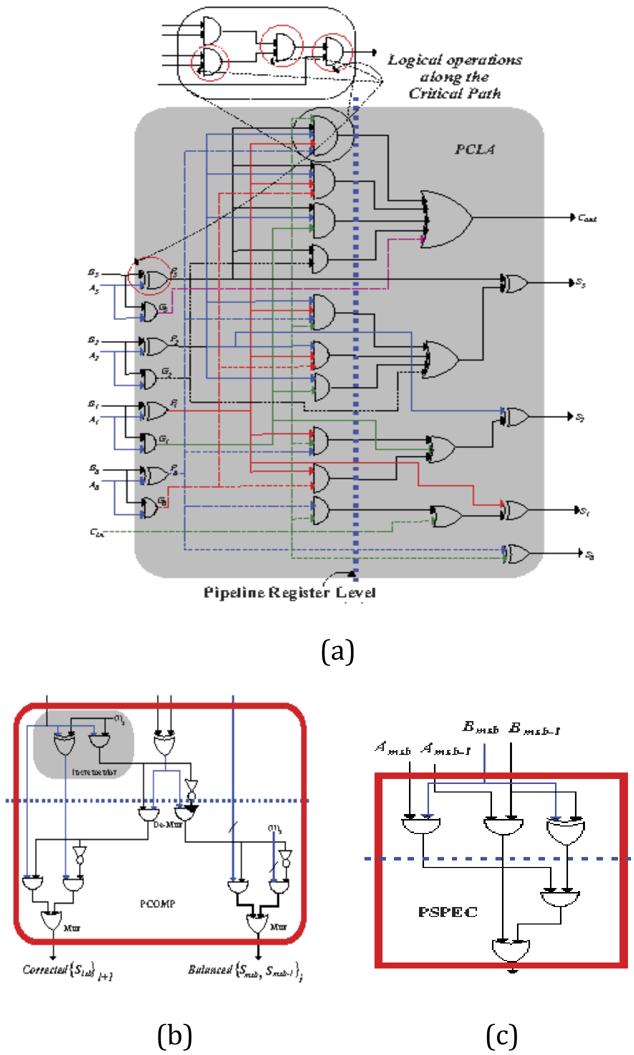


Fig 3: Gate level circuit of (a) 4-bit pipelined carry lookahead adder (b) pipelined compensator(PCOMP) (c) pipelined speculator(PSPEC)

Fig 3 shows the gate level designs of the sub blocks and their respective pipelined stages. From the figures we can conclude that the critical path of the suggested pipelined architecture lies in PCLA and it includes only one XOR and three AND gate input delays. Therefore the equation for delay is given by

$$\partial_{crit-prop} = \partial_{clk-qff} + \partial_{setup-ff} + 3 * \partial_{and} + \partial_{xor}$$

This includes clock-q delay and the setup time required to launch and capture flip-flops respectively. Thus maximum clock frequency can be obtained by the inverse of the delay.

4. EXPERMENTAL RESULT AND COMPARISION

This section presents the functional verification and board level implementation of non-pipelined and pipelined ISA. Subsequently the post-simulation results are compared.

a) FPGA implementation

In this work, the proposed and suggested ISA adder-architecture has been coded in hardware descriptive language (HDL) and then simulated as well as synthesized in ISE 14.7 design suite. We have synthesized this architecture for three different configurations : *n* = 8-bit, *n* = 16-bit and *n* = 32-bit. After the successful syntax check and synthesis of the design, the generated net-lists are placed and routed (P&R) on Spartan-3E version of Xilinx FPGA board. Then after the timing information and the number of devices utilized are also calculated for both the adders.

The maximum operating frequency is 127.7MHz.This value is 69.29%, 57.4%, 56.69% better than the clock frequencies achieved by 8-bit,16-bit,32-bit non-pipelined adders. The comparison of exact area occupied in terms of LUT's and the power required of these adders are possible by synthesizing as well as laying out FPGA for each of these adders.

b) Power and area analysis

In the digital circuit design, pipelining is the process of shortening the delay in critical path at the cost of area which is predominated by the registers used to create pipeline stages in the design. Therefore, the suggested ISA architecture that is deep pipelined definitely requires extra registers in comparison with the proposed non-pipelined ones. On the other side, we have divided the suggested ISA architecture into different stages by pipelining it. Now, this makes our architecture suitable for clock-gating. In the new design, we have gated the clock signal that is fed into every stage. On doing this, the ideal stages of our architecture can be deferred from the clock switching which significantly reduces the power consumption. Such gating is valid only during the beginning and ending sessions of the addition process. On the starting of addition, later pipeline stages (towards the output side) of the design are ideal and these stages can be clock gated. Unlike towards the end of addition process, earlier stages (near the input side) of the design seem to be ideal and are clock-gated. For example: pipeline stages five , four, three and two are ideal while the process is being carried out in the first stage when the addition begins. Similarly, first stage will be ideal while rest keeps processing data when addition is towards the completion. However, while the adder is in-between the process of adding continuous stream of data then there is no point of gating the clock because all the stages are busy performing the operations.

In order to obtain number of LUT's required and amount of power consumed by both the adders, this work includes synthesis and post layout simulation results of three configurations of both pipelined and non-pipelined adders for the purpose of comparison.

ISA configurations	8-bit NPLA	8-bit PLA	16-bit NPLA	16-bit PLA	32-bit NPLA	32-bit PLA
FPGA family	Spartan-3E	Spartan-3E	Spartan-3E	Spartan-3E	Spartan-3E	Spartan-3E
FPGA device	Xc-7a100tcsg324-1L	Xc-7a100tcsg324-1L	Xc-7a100tcsg324-1L	Xc-7a100tcsg324-1L	Xc-7a100tcsg324-1L	Xc-7a100tcsg324-1L
4-ip LUT's	13	31	22	72	46	149
Crit.path delay(ns)	11.292	7.888	13.672	7.888	13.762	7.888
Max.clk freq (MHz)	88.592	126.77	73.141	126.77	72.66	126.77
Power consumed(w)	5.533	4.95	10.596	8.94	21.865	17.634

Table 1: Comparison of post P&R results obtained from FPGA implementations of 8, 16, 32-bit pipelined and non-pipelined ISA designs.

The 32-bit pipelined ISA consumes a total power of 17.634w and it can be observed that this architecture consumes 4.235w lesser power than the non-pipelined ISA and this is possible only due to the implementation of clock-gating technique and also due to pipelining the critical path delay of 32-bit non-pipelined adder is 13.762ns where as for pipelined one it is 7.888 ns and hence due to pipelining the delay is reduced by 6ns. The above table shows the comparison of total area consumed in terms of LUT's, power consumed, critical path delay and the maximum clock frequency of 8-bit,16-bit,32-bit non-pipelined as well as pipelined adders. Thereby the adder presented has area degradation in comparison with the non-pipelined one.

5. CONCLUSION

In this paper we presented the high-speed and low- power version of ISA design. This architecture is fine-grain pipelined and clock-gated to reduce delay as well as to reduce power consumption respectively. Experimental results showed that the modified architecture operate at a maximum frequency of 127.72MHz in FPGA. Subsequently a 32-bit pipelined architecture consumes power of 17.634w.Thereby, such design would definitely play significant role in the design of contemporary as well as future electronic devices for IoE and many other applications. However, the area issue can be resolved to some extent by using lower technology nodes in the design process.

6. REFERENCES

[1] Behzad Razavi, "Cognitive Radio Design Challenges and Techniques," IEEE Journals of Solid-State Circuits (JSSC), vol. 45, no. 8, pp. 1542- 1553, 2010.

[2] Gyanendra Prasad Joshi, Seung Yeob Nam and Sung Won Kim, "Cog- nitive Radio Wireless Sensor Networks: Applications, Challenges and Research Trends," Sensors, vol. 13, no. 9, pp. 11196-11228, 2013.

[3] D. Blaauw et al., "IoT Design Space Challenges: Circuits and Systems," IEEE Symposium on VLSI Technology (VLSI-Technology): Digest of Technical Papers, pp. 1-2, 2014.

[4] T. Liu and S. L. Lu, "Performance Improvement with Circuit-level Speculation," 33rd Annual IEEE ACM International Symposium on Mi- croarchitecture (MICRO-33), pp. 348-355, 2000.

[5] N. Zhu, W.-L. Goh, and K.-S. Yeo, "An Enhanced Low-power High-speed Adder For Error-tolerant Application," 12th International Symposium on Integrated Circuits (ISIC), pp. 69-72, 2009.

[6] M. Weber, M. Putic, H. Zhang, J. Lach, and J. Huang, "Balancing Adder for Error Tolerant Applications," IEEE International Symposium on Circuits and Systems (ISCAS), pp. 3038-3041, 2013.

[7] N. Zhu, W.-L. Goh, G. Wang, and K.-S. Yeo, "Enhanced Low-power High-speed Adder for Error-tolerant Application," IEEE International SoC Design Conference (ISOCC), pp. 323-327, 2010.

[8] Y. Kim, Y. Zhang, and P. Li, "An Energy Efficient Approximate Adder with Carry Skip for Error Resilient Neuromorphic VLSI Systems," IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 130- 137, 2013.

[9] Vincent Camus, Jeremy Schlachter and Christian Enz, "Energy-Efficient Inexact Speculative Adder with High Performance and Accuracy Control," IEEE International Symposium on Circuits and Systems (ISCAS), pp. 45- 48, 2015.