(IJRST) 2016, Vol. No. 6, Issue No. III, Jul-Sep **e-ISSN: 2249-0604, p-ISSN: 2454-180X**

DESIGN AND IMPLEMENTATION OF HIGH SPEED AND LOW POWER CARRY SKIP ADDER

***Mercy P M R, **M.Vellapandian**
**PG Scholar, VLSI Design,*
Sardar Raja College of Engineering, Alangulam, Tirunelveli, Tamil Nadu
***Head Of The Department of ECE,*
Sardar Raja College of Engineering, Alangulam, Tirunelveli, Tamil Nadu.

ABSTRACT

The most timing critical part of logic design usually contains one or more arthimetic operations, in which addition is commonly involved. In VLSI applications, area, delay and power are the important factors which must be taken into account in the design of a fast adder. The Carry-skip adder reduces the time needed to propagate the carry by skipping over groups of consecutive adder stages, is known to be comparable in speed to the carry look-ahead technique while it use less logic area and less power. In this paper, a design of 8-bit Carry skip Adder by various existing logic styles are to be compared quantitatively and qualitatively by performing detailed using Xilinx v13.0.

INTRODUCTION

Multipliers are key components of many high performance systems such as FIR filters, microprocessors, digital signal processors, *etc*. A system‘s performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest clement in the system. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. As a result, a whole spectrum of multipliers with different area, speed constraints have been designed with fully parallel Multipliers at one end of the spectrum and fully serial multipliers at the other end. With the increasing level of device integration and the growth in complexity of microelectronic circuits, reduction of power dissipation has also come to the fore as a primary design goal. While power efficiency has always been desirable in electronic circuits, only recently has it become a limiting factor for a broad range of applications, requiring consideration early on in the design process. Power dissipation limitations come in two flavors. The first is related to cooling considerations when implementing high performance systems. High speed circuits dissipate large amounts of energy in a short amount of time, generating a great deal of heat as a by-product. This heat needs to be removed by the package on which integrated circuits are mounted. Heat removal may become a limiting factor if the package (PC board, system enclosure, heat sink) cannot adequately dissipate this heat, or if the required thermal components are too expensive for the application. The second failure mode of high-power circuits relates to the increasing popularity of portable electronic devices. Laptop computers, pagers, portable video players and cellular phones all use

(IJRST) 2016, Vol. No. 6, Issue No. III, Jul-Sep **e-ISSN: 2249-0604, p-ISSN: 2454-180X**

batteries as a power source, which by their nature provide a limited time of operation before they require recharging. To extend battery life, low power operation is desirable in integrated circuits. Furthermore, successive generations of applications often require more computing power, placing greater demands on energy storage elements within the system. Technology improvements in the last few decades have succeeded in reducing power consumption. Trends such as using CMOS instead of bipolar devices, and reduction in feature size of lithographic processes have served to reduce power dissipation, although other objectives, namely high integration and speed, were the primary goals of such improvements.

EXISTING SYSTEM

CONCEPT OF VL-ADDER

Carry Propagation in a Carry-Select Adder

In a conventional design of M-bit square-root carry-select adder (CSA), the input bits are divided into $\approx \sqrt{2M}$ carry-select stages (CSSs), as shown in Fig. 3.1. One CSS includes carry generation blocks, a multiplexer (MUX) for carry selection, and a sum generation block. In CSS_i, two carry-out signals C_{o0} and C_{o1} of every bit are precalculated by assuming that $C_{in,i}$, the carry-in signal of CSS_i, is 0 or 1, respectively. C_{o0} or C_{o1} is then selected by $C_{in,i}$, the actual carry-out signal $C_{out,i-1}$ of CSS_{i-1}, the previous CSS.

The carry-out signal of the MSB adder in CSS_i is sent out as the carry-out signal $C_{out,i}$ of CSS_i, which is also the carry-in signal $C_{in,i+1}$ of CSS_{i+1} , the next CSS. Usually, the number of input bits for each CSS increases linearly from the least significant stage to the most significant stage, and the first CSS includes two single-bit adders, one full adder (FA), and one half adder (HA). Hence, in a conventional square-root CSA with CSSs, is chosen as (3+n) n/2 $\approx M$. In particular, when M=64, n ≈ 10. There are multiple critical carry propagation paths in CSA, each of which starts from the first bit adder of every CSS, crosses the MUXes of the sequential stages, and ends at the sum generation block of the last stage. The longest delay of CSA D_{CSA} can be calculated by

$$D_{CSA} = D_{setup} + q_0 D_{carry} + n D_{mux} + D_{sum} \quad (3.1)$$

where D_{carry}, D_{mux} and D_{sum} are the delays of carry generation circuitry, MUX, and sum generation circuitry, respectively; q_0 is the number of input bits in the first and D_{setup} is the setup time of CSA, which is the time taken to create the intermediate signals G(generation) and P(propagation). The delay of a CSA heavily depends on the input vectors, e.g., the carry propagation through the MUX chain of CSA. Let m_i denote the number of bits in CSS_i.

(IJRST) 2016, Vol. No. 6, Issue No. III, Jul-Sep e-ISSN: 2249-0604, p-ISSN: 2454-180X

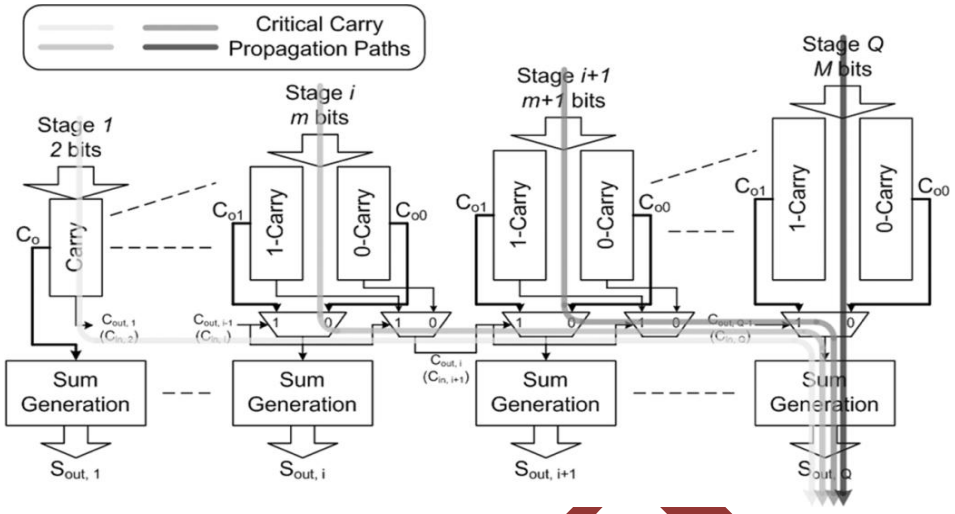


Fig. 1. Carry propagation in CSA

VL-CSA

A new VL CSA architecture called "VL-CSA" is shown in the row "VL-CSA" in Table I. The CSSs of VL-CSA are divided into two separate sequences: sequence I and sequence II. Sequence I and sequence II include k(=7) and m(=6) CSSs, respectively. In each sequence, the first stage CSS starts with a relatively small number of input bits, and the number of input bits to a CSS increases with the stage number of the CSS, as in the case of a conventional CSA. Now, we combine the two sequences of CSSs and label them from 1 to m+k. However, the carry can propagate through the first N_C CSSs in sequence II, i.e., CSS_{k+1} ..... CSS_{k+Nc} only if

$$P_{N_c} = \prod_{i\in\{CCS_{k+1}.....CCS_{k+N_c}\}} P_i = 1 \quad \text{.....} \quad (3.2)$$

where $P_i = A_i \oplus B_i$ is the propagation signal of the adder of bit in CSS_{k+1} CSS_{k+Nc}. A long-latency operation, of which the longest delay is L_{long}, occurs only when the carry can propagate through first N_c CSSs in sequence II. For random inputs, the probability of long-latency operations, denoted by Pr_{long} , *is*

$$Pr_{long} = \prod_{i=1}^{N_c} \frac{1}{2^{m_i}} \quad (3.3)$$

where m_i is the total number of input bits of CSS_{k+1}. When a short-latency operation is detected, say $P_{N_c} = 0$, the carry propagation in VL-CSA is divided into two parts: one from CSS_1 through CSS_{k+Nc} and the other one from CSS_{k+Nc} through the last stage CSS_{k+m}. The longest latency of all short-latency operations, denoted by L_{short}, is

$$L_{short} = \max(D_{max\,1}, D_{max\,2}) \quad (3.4)$$

where $D_{max\,1}$ and $D_{max\,2}$ are the longest delays of the first and the second parts, respectively. We have

$$D_{max\,1} = D_{setup} + q_{0,I}\, D_{carry} + (k+N_c)\, D_{mux} + D_{sum}$$

$$D_{max\,2} = D_{setup} + q_{0,II}\, D_{carry} + m\, D_{mux} + D_{sum} \quad (3.5)$$

$$P_{VL\text{-}CSA}(1+Pr_{long}) < P_{CSA} \quad (3.6)$$

where $P_{VL\text{-}CSA}$ and P_{CSA} are power consumptions of the VL-CSA (at $V_{VL\text{-}CSA}$) and the conventional CSA (at V_{CSA}), respectively. Based on the CSA structure, the numbers of CSSs

in the two sequences of VL-CSA, m and k, satisfy$(2q_{o,I}$+k-1)$\frac{k}{2}$+$(2q_{o,I}$+m-1)$\frac{m}{2}$ ≈M=64 (3.7)

By combining (3.1)–(3.7), a 64-bit VL-CSA with 13 CSSs is designed, as shown in the row "VL-CSA" of Table 3.1. In each sequence, the number of input bits of a CSS roughly increases with the stage number.

Table 3.1Numbers of Input Bits of CSSs in a 64-Bit Standard CSA And VL-CSA (and a Modified VL-CSA)

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13
Conv. CSA	2	2	3	4	6	7	8	9	11	12	/	/	/
VL-CSA	2	2	3	4	6	7	7	***4***	***3***	5	6	7	8

Detection of Long Latency for VL-CSA

In our 64-bit VL-CSA design, the input bits of and - (see Table 3.1) are used to detect the long-latency operations. The detection logic can be expressed as follows:

LONG_OP=$A_{31} \oplus B_{31}, A_{32} \oplus B_{32} \ldots\ldots\ldots\ldots\ldots\ldots A_{37} \oplus B_{37}$ (3.8)

A long-latency operation occurs only when LONG_OP = 1. The corresponding circuit, called carry length detection circuit (CLDC), is shown in Fig. 3.2. When an LONG OP signal is pulled up ''high,'' a gating signal is triggered to disable the clock (Gen CLK) switching in the following clock cycle, as shown in Fig. 3.3 Here, signal TH ADJ is fixed at ''low.''

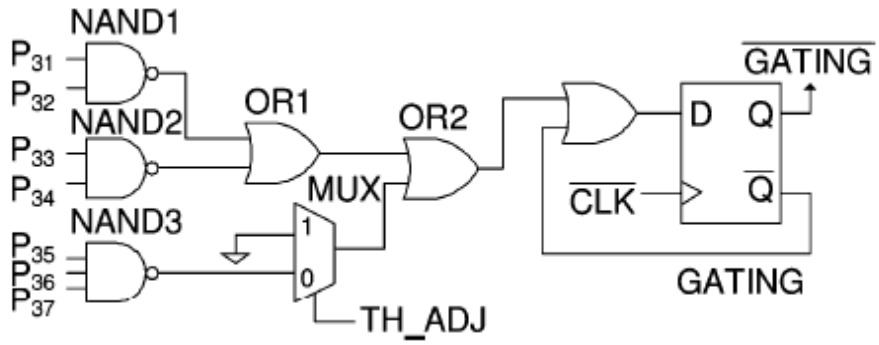


Fig 2 CLDC Design of VL-CSA

PROPOSED SYSTEM

General Description of the Proposed Structure

The structure is based on combining the concatenation and the incrementation schemes [13] with the Conv-CSKA structure, and hence, is denoted by CI-CSKA. It provides us with the ability to use simpler carry skip logics. The logic replaces 2:1 multiplexers by AOI/OAI compound gates (Fig. 3). The gates, which consist of fewer transistors, have lower delay, area, and smaller power consumption compared with those of the 2:1 multiplexer [37]. Note that, in this structure, as the carry propagates through the skip logics, it becomes

complemented. Therefore, at the output of the skip logic of even stages, the complement of the carry is generated. The structure has a considerable lower propagation delay with a slightly smaller area compared with those of the conventional one. Note that while the power consumptions of the AOI (or OAI) gate are smaller than that of the multiplexer, the power consumption of the proposed CI-CSKA is a little more than that of the conventional one.

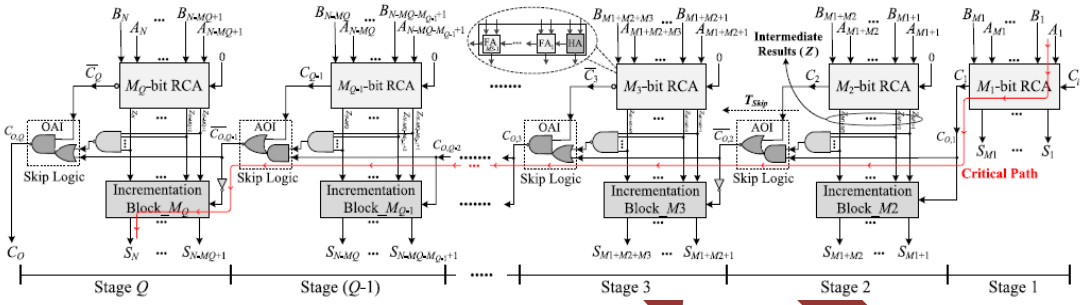


Fig 3 CI-CSKA Structure

This is due to the increase in the number of the gates, which imposes a higher wiring capacitance (in the noncritical paths). Now, we describe the internal structure of the proposed CI-CSKA shown in Fig. 4.1 in more detail. The adder contains two N bits inputs, A and B, and Q stages. Each stage consists of an RCA block with the size of M_j (j = 1, . . . , Q). In this structure, the carry input of all the RCA blocks, except for the first block which is C_i, is zero (concatenation of the RCA blocks). Therefore, all the blocks execute their jobs simultaneously. In this structure, when the first block computes the summation of its corresponding input bits (i.e., S_{M1}, . . . , S_1), and C_1, the other blocks simultaneously compute the intermediate results [i.e., $\{Z_{Kj+Mj}$, . . . , Z_{Kj+2}, $Z_{Kj+1}\}$ for $K_j = \sum_{r=1}^{1-1} M_r (1 = 2 \dots \dots . . Q)$, and also C_j signals. In the proposed structure, the first stage has only one block, which is RCA. The stages 2 to Q consist of two blocks of RCA and incrementation. The incrementation block uses the intermediate results generated by the RCA block and the carry output of the previous stage to calculate the final summation of the stage.

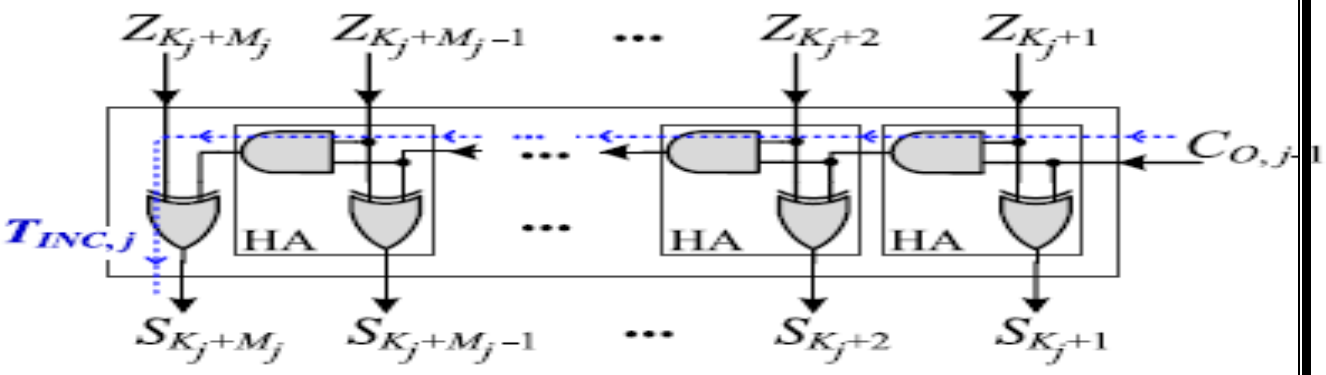


Fig. 4 Internal structure of the j th incrementation block, $K_j = \sum_{r=1}^{1-1} M_r (1 = 2 \dots \dots . . Q)$.

Stage Sizes Consideration

Similar to the Conv-CSKA structure, the proposed CI-CSKA structure may be implemented with either FSS or VSS. Here, the stage size is the same as the RCA and

incrementation blocks size. In the case of the FSS (FSS-CI-CSKA), there are Q = N/M stages with the size of M. The optimum value of M, which may be obtained using (11), is given by

$$M_{opt} = \sqrt{\frac{N(T_{AOI} + T_{OAI})}{2(T_{CARRY} + T_{AND})}}$$

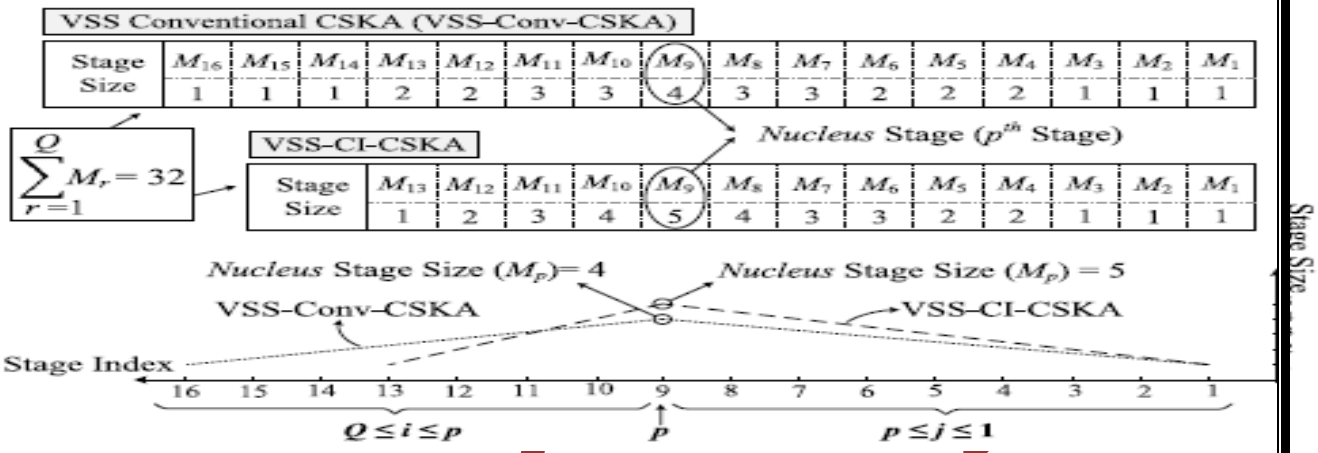


Fig. 5 Sizes of the stages in the case of VSS for the proposed andconventional 32-bit CSKA structures in 45-nm static CMOS technology.

4.3 Proposed Hybrid Variable Latency CSKA Structure

The basic idea behind using VSS CSKA structures was based on almost balancing the delays of paths such that the delay of the critical path is minimized compared with that of the FSS structure [21]. This deprives us from having the opportunity of using the slack time for the supply voltage scaling. To provide the variable latency feature for the VSS CSKA structure, we replace some of the middle stages in our proposed structure with a PPA modified in this paper. It should be noted that since the Conv-CSKA structure has a lower speed than that of the proposed one, in this section, we do not consider the conventional structure. The proposed hybrid variable latency CSKA structure is shown in Fig. 4.5 where an Mp-bit modified PPA is used for the pth stage (nucleus stage). Since the nucleus stage, which has the largest size (and delay) among the stages, is present in both SLP1 and SLP2, replacing it by the PPA reduces the delay of the longest off-critical paths.

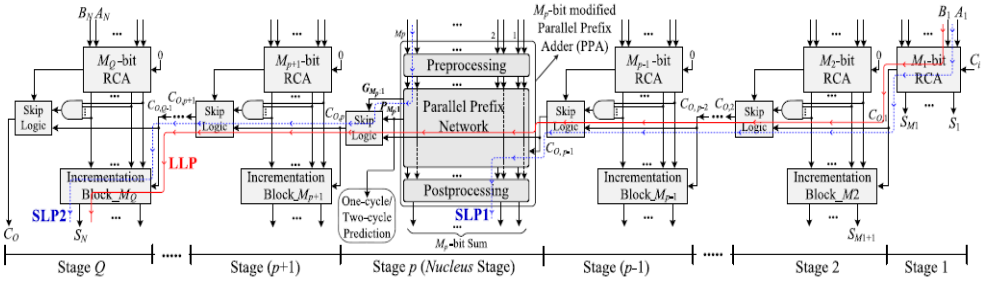


Fig 6 Structure of the proposed hybrid variable latency CSKA

Thus, the use of the fast PPA helps increasing the available slack time in the variable latency structure. It should be mentioned that since the input bits of the PPA block are used in

the predictor block, this block becomes parts of both SLP1 and SLP2. In the proposed hybrid structure, the prefix network of the Brent–Kung adder [39] is used for constructing the nucleus stage (Fig. 7).

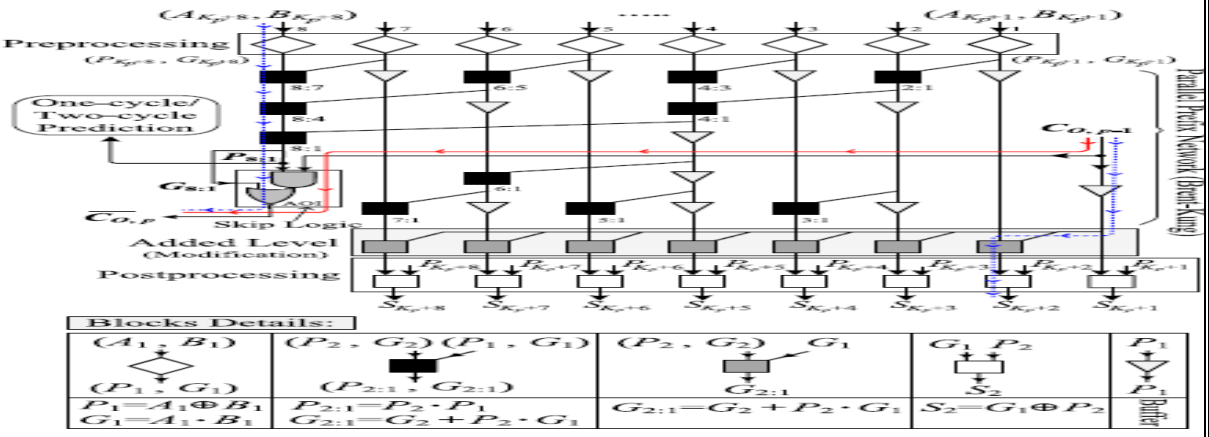
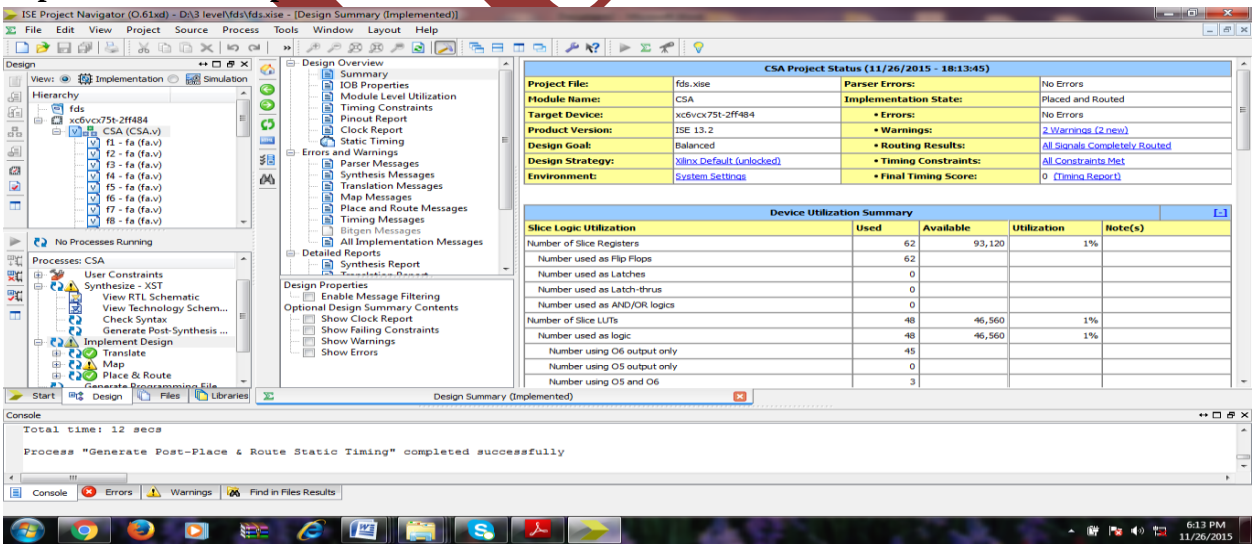


Fig 7 Internal structure of the pth stage of the proposed hybrid variable latency CSKA. M_p is equal to 8 and K_p

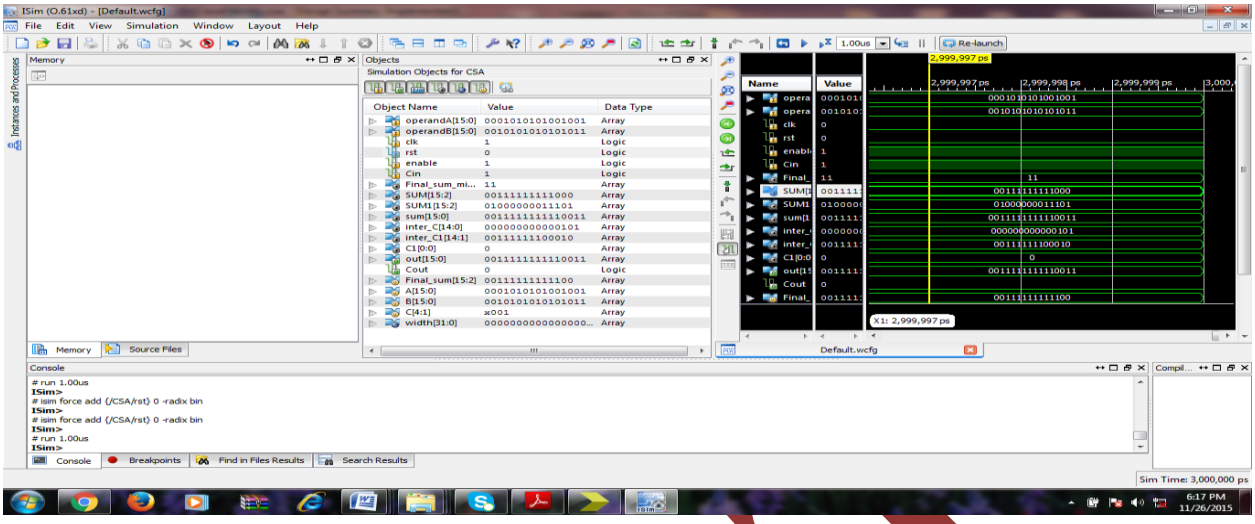
One the advantages of the this adder compared with other prefix adders is that in this structure, using forward paths, the longest carry is calculated sooner compared with the intermediate carries, which are computed by backward paths. In addition, the fan-out of adder is less than other parallel adders, while the length of its wiring is smaller [14]. Finally, it has a simple and regular layout. The internal structure of the stage p, including the modified PPA and skip logic, is shown in Fig. 7.

RESULTS AND DISCUSSION

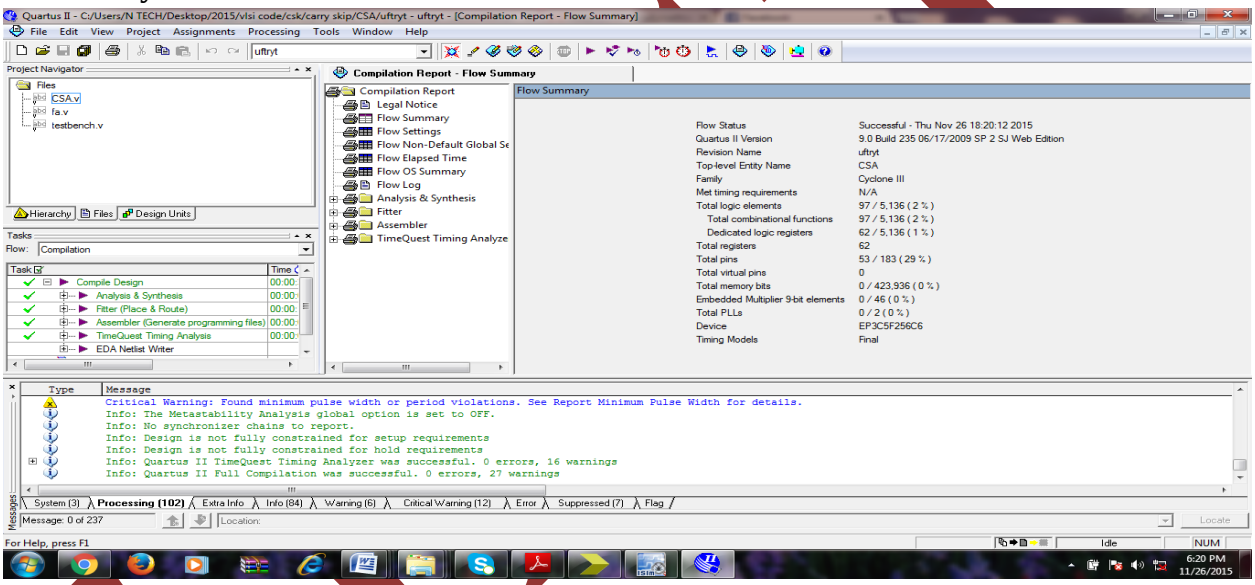
Implementation output without Error



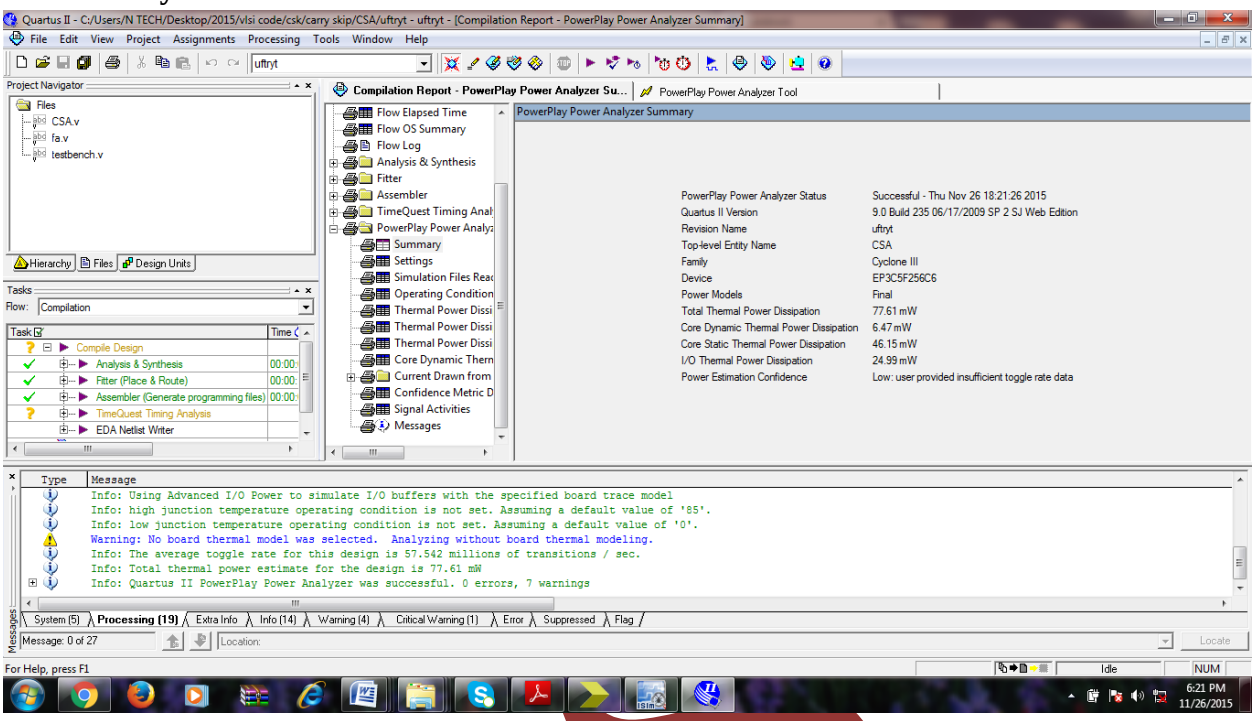
Output waveform



Area Analysis



Power Analysis



CONCLUSION

In this project, a static CMOS CSKA structure called CI-CSKA was proposed, which exhibits a higher speed and lower energy consumption compared with those of the conventional one. The speed enhancement was achieved by modifying the structure through the concatenation and incrementation techniques. In addition, AOI and OAI compound gates were exploited for the carry skip logics. The efficiency of the proposed structure for both FSS and VSS was studied by comparing its power and delay with those of the Conv-CSKA, RCA, CIA, SQRT-CSLA, and KSA structures. The results revealed considerably lower PDP for the VSS implementation of the CI-CSKA structure over a wide range of voltage from super-threshold to near threshold. The results also suggested the CI-CSKA structure as a very good adder for the applications where both the speed and energy consumption are critical. In addition, a hybrid variable latency extension of the structure was proposed. It exploited a modified parallel adder structure at the middle stage for increasing the slack time, which provided us with the opportunity for lowering the energy consumption by reducing the supply voltage.

REFERENCES

[1] I. Koren, Computer Arithmetic Algorithms, 2nd ed. Natick, MA, USA: A K Peters, Ltd., 2002.

[2] R. Zlatanovici, S. Kao, and B. Nikolic, “Energy–delay optimization of 64-bit carry-lookahead adders with a 240 ps 90 nm CMOS design example,” IEEE J. Solid-State Circuits, vol. 44, no. 2, pp. 569–583, Feb. 2009.

[3] S. K. Mathew, M. A. Anders, B. Bloechel, T. Nguyen, R. K. Krishnamurthy, and S. Borkar, "A 4-GHz 300-mW 64-bit integer execution ALU with dual supply voltages in 90-nm CMOS," IEEE J. Solid-State Circuits, vol. 40, no. 1, pp. 44–51, Jan. 2005.

[4] V. G. Oklobdzija, B. R. Zeydel, H. Q. Dao, S. Mathew, and R. Krishnamurthy, "Comparison of high-performance VLSI adders in the energy-delay space," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 6, pp. 754–758, Jun. 2005.

[5] B. Ramkumar and H. M. Kittur, "Low-power and area-efficient carry select adder," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 2, pp. 371–375, Feb. 2012.

[6] M. Vratonjic, B. R. Zeydel, and V. G. Oklobdzija, "Low- and ultra low-power arithmetic units: Design and comparison," in Proc. IEEE Int. Conf. Comput. Design, VLSI Comput. Process. (ICCD), Oct. 2005, pp. 249–252.

[7] C. Nagendra, M. J. Irwin, and R. M. Owens, "Area-time-power tradeoffs in parallel adders," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 43, no. 10, pp. 689–702, Oct. 1996.

[8] Y. He and C.-H. Chang, "A power-delay efficient hybrid carrylookahead/carry-select based redundant binary to two's complement converter," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 1, pp. 336–346, Feb. 2008.

[9] C.-H. Chang, J. Gu, and M. Zhang, "A review of 0.18 μm full adder performances for tree structured arithmetic circuits," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 6, pp. 686–695, Jun. 2005.

[10] D. Markovic, C. C. Wang, L. P. Alarcon, T.-T. Liu, and J. M. Rabaey, "Ultralow-power design in near-threshold region," Proc. IEEE, vol. 98, no. 2, pp. 237–252, Feb. 2010.