Folded 32 BIT Carry Skip Adder Using RCA Unit Without Nucleus Stage

Harika Kothapalli[1], Prem Kumar Medapati[2]
[12]*Shri Vishnu College for Women. Vishnupur, Bhimavaram, West godavari district, Andhra pradesh,*

Abstract- In this paper, folded carry skip adder was implemented and demonstrated. This results in low energy consumption when compared with hybrid carry skip adder. In this carry skip unit was replaced by a prediction block named carry feed forward which gives us intermediately predictions and helps eliminating prediction block and nucleus stage and minimizes the complexity of architecture. This reduces the delay factor of the circuit. The resultant simulations were implemented using XILINX- verilog, The analysis of results represents low power consumption through critical path.

IndexTerms **-** Carry Skip adder, low energy consumption, hybrid carry skip adder, prediction block, carry feed forward.

I. INTRODUCTION

Cryptography An adder is a digital circuit that performs addition of numbers. In many computers and other kinds of processors adders are used in the arithmetic logic units or ALU. They are also utilized in other parts of the processor, where they are used to calculate addresses, table indices, increment and decrement operators, and similar operations.

Although adders can be constructed for many number representations, such as binary-coded decimal or excess-3, the most common adders operate on binary numbers. In cases where two's complement or ones' complement is being used to represent negative numbers, it is trivial to modify an adder into an adder–subtractor. Other signed number representations require more logic around the basic adder.

In this section, a short description of the adder architecture and the exact time delay (T) and area (A) complexity based on unit gate model is presented. In the unit gate model each gate has a gate-count of one and a gate-delay of one excluding XOR and XNOR gates having gate counts and gate delays of two, while the gates with more than 2 inputs, the gate-counts and gate-delays can be computed in terms of the ones given for the gates with two inputs; also, inverters and buffers are ignored.

The standard carry generate-propagate logic is used to reduce the critical delay of the adder while blocks of RCAs are used for lesser power consumption. In our design, the generate propagate logic balances the delay and the number of inputs to the skip logic limits the critical path delay.

The 32-bit carry-skip adder design presented in this paper uses a combination of RCAs together with carry-skip logic (SKIP), carry-generate logic (CG), and group generate propagate logic (PG). The complete adder is divided into a number of variable-width blocks. Both the carry generation and skip logic use AOI and OAI circuits. The width of each block is limited by the target delay T. Each block is further divided into sub-blocks. A sub-block may contain additional levels of sub-blocks in a recursive manner. The lowest-level sub-block is formed by a number of variable width RCAs.

Power dissipation is major problem in the electronics device so the goal of this project is to analyze and compare the performance of Carry skip adder at different input voltages. Various filter designs are found in adder application so it is required to efficiently compute the multiply and accumulate operations. Various techniques have been proposed to design multiplexer which are efficient in terms of performance, low power consumption and area.

II. LITERATURE REVIEW

Since there is a finite rise/fall time for both pMOS and nMOS, during transition, for example, from off to on, both the transistors will be on for a small period of time in which current will find a path directly from VDD to ground, hence creating a short-circuit current. Short-circuit power dissipation increases with rise and fall time of the transistors.

An additional form of power consumption became significant in the 1990s as wires on chip became narrower and the long wires became more resistive. CMOS gates at the end of those resistive wires see slow input transitions. During the middle of these transitions, both the NMOS and PMOS logic networks are partially conductive, and current flows directly from VDD to VSS. The power thus used is called crowbar power. Careful design which avoids weakly driven long skinny wires ameliorates this effect, but crowbar power can be a substantial part of dynamic CMOS power.

To speed up designs, manufacturers have switched to constructions that have lower voltage thresholds but because of this a modern NMOS transistor with a Vth of 200 mV has a significant subthreshold leakage current. Designs (e.g. desktop processors) which include vast numbers of circuits which are not actively switching still consume power because of this leakage current. Leakage power is a significant portion of the total power consumed by such designs. Multi-threshold CMOS (MTCMOS), now available from foundries, is one approach to managing leakage power. With MTCMOS, high Vth transistors are used when switching speed is very important, while low Vth transistors are used in speed sensitive paths. Further technology advances that use even thinner gate dielectrics have an additional leakage component because of current tunnelling through the extremely thin gate dielectric. Using high-k dielectrics instead of silicon dioxide that is the conventional gate dielectric allows similar device performance, but with a thicker gate insulator, thus avoiding this current. Leakage power reduction using new material and system designs is critical to sustaining scaling of CMOS.

ADDERS are a key building block in arithmetic and logic units (ALUs) [1] and hence increasing their speed and

reducing their power/energy consumption strongly affect the speed and power consumption of processors. There are many works on the subject of optimizing the speed and power of these units, which have been reported in [2]–[9]. Obviously, it is highly desirable to achieve higher speeds at low-power/energy consumptions, which is a challenge for the designers of general purpose processors.

One of the effective techniques to lower the power consumption of digital circuits is to reduce the supply voltage due to quadratic dependence of the switching energy on the voltage. Moreover, the subthreshold current, which is the main leakage component in OFF devices, has an exponential dependence on the supply voltage level through the drain-induced barrier lowering effect [10]. Depending on the amount of the supply voltage reduction, the operation of ON devices may reside in the superthreshold, near-threshold, or subthreshold regions. Working in the superthreshold region provides us with lower delay and higher switching and leakage powers compared with the near/subthreshold regions. In the subthreshold region, the logic gate delay and leakage power exhibit exponential dependences on the supply and threshold voltages. Moreover, these voltages are (potentially) subject to process and environmental variations in the nanoscale technologies. The variations increase uncertainties in the aforesaid performance parameters. In addition, the small subthreshold current causes a large delay for the circuits operating in the subthreshold region [10].

Recently, the near-threshold region has been considered as a region that provides a more desirable tradeoff point between delay and power dissipation compared with that of the subthreshold one, because it results in lower delay compared with the subthreshold region and significantly lowers switching and leakage powers compared with the superthreshold region. In addition, near-threshold operation, which uses supply voltage levels near the threshold voltage of transistors [11], suffers considerably less from the process and environmental variations compared with the sub-threshold region.

There are many adder families with different delays, power consumptions, and area usages. Examples include ripple carry adder (RCA), carry increment adder (CIA), carry skip adder (CSKA), carry select adder (CSLA), and parallel prefix adders (PPAs). The descriptions of each of these adder architectures along with their characteristics may be found in [1] and [13]. The RCA has the simplest structure with the smallest area and power consumption but with the worst critical path delay. In the CSLA, the speed, power consumption, and area usages are considerably larger than those of the RCA.

The PPAs, which are also called carry look-ahead adders, exploit direct parallel prefix structures to generate the carry as fast as possible [14]. There are different types of the parallel prefix algorithms that lead to different PPA structures with different performances.

As an example, the Kogge–Stone adder (KSA) [15] is one of the fastest structures but results in large power consumption and area usage. It should be noted that the structure complexities of PPAs are more than those of other adder schemes [13], [16]. The CSKA, which is an efficient adder in terms of power consumption and area usage, was introduced in [17]. The critical path delay of the CSKA is much smaller than the one in the RCA, whereas its area and power consumption are similar to those of the RCA.

In addition, the power-delay product (PDP) of the CSKA is smaller than those of the CSLA and PPA structures [19]. In addition, due to the small number of transistors, the CSKA benefits from relatively short wiring lengths as well as a regular and simple layout [18]. The comparatively lower speed of this adder structure, however, limits its use for high-speed applications.

In this thesis, given the attractive features of the CSKA structure, we have focused on reducing its delay by modifying its implementation based on the static CMOS logic. The concentration on the static CMOS originates from the desire to have a reliably operating circuit under a wide range of supply voltages in highly scaled technologies [10]. The proposed modification increases the speed considerably while maintaining the low area and power consumption features of the CSKA. In addition, an adjustment of the structure, based on the variable latency technique, which in turn lowers the power consumption without considerably impacting the CSKA speed, is also presented. To the best of our knowledge, no work concentrating on design of CSKAs operating from the super threshold region down to near-threshold region and also, the design of (hybrid) variable latency CSKA structures have been reported in the literature.

III. METHODOLOGY

The sum is found in two stages. The RCA block finds intermediate results that are the first stage and the final result is the result from the incrementation block. The incrementation block is a chain of half adders in which one input is carry of previous stage and other input is intermediate result. Since the carry input is given to incrementation block, there is no carry input for RCA block and hence all the 1st Full adders of RCA block in all stages except the 1st stage can be replaced by half adder. In the first stage sum is calculated by RCA and the carry is propagated to AOI block. The propagated carry is given to both AOI skip logic and also to the incrementation block. Since the RCA blocks of every stage does not requires carry of previous stages all RCA block find the intermediate results simultaneously.

The operation of AOI and OAI stages can be explained by two cases, first case is that if carry is propagated from previous state and second is when carry is not propagated. Consider the case in which carry is propagated from previous stage and all input bits are in carry propagation condition, then all intermediate bits will be 1, hence the output of AND gate will be 1. This one will be propagated to AOI logic. The first gate of AOI compound gate is AND gate. This AND gate will generate 1 only if there is a carry from previous stage is 1. The second gate of AOI compound gate is NOR. So the input to this NOR will be one so it will generate output 0, which is the inverted carry from previous stage.

This is again inverted by a NOT gate and given to incrementation block. So the time of propagation of carry in this case will be the sum of time taken by and gate TAND and time taken by skip logics which can be TOAI or TAOI logic.

If all input bits are in propagation condition and carry from previous stage is zero the output of AND gate in AOI will be zero and hence it have to wait to receive the third input which is from RCA. Same condition exists if the input bits are not in propagation condition. Similar case occurs for OAI block also. The time required for carry propagation if all the intermediate results are not one will be the sum of time taken for carry propagation through the RCA chain TRCA and the time taken for carry to propagate through the skip logic.

Advantage of this model is that for finding carry output of next stage the carry from incrementation block is not required. So the delay for generation of final result does not depends on the delay for carry propagation from one block to other. Also all the intermediate results can be calculated simultaneously by RCA without waiting for the carry from previous block. The carry is calculated based on intermediate results and carry from previous block. The disadvantage of this model is that if all inputs are not in propagation condition then all the intermediate results will not be one, in such cases the skip logic have to wait for the carry from RCA block.

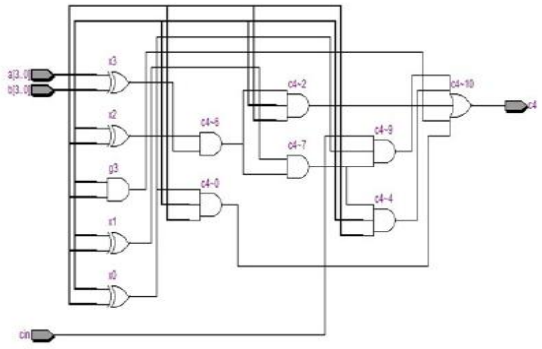

Fig.1: CARRY FEED FORWARD BLOCK

In conventional carry skip adder the carry is skipped only when all input bits are in propagation mode. Only in this case the carry prediction unit will generate an output of one, which is given to multiplexer as select line. Based on this input the multiplexer is selecting any one of the input that is the carry of previous state or the carry of the RCA. If the input bits are not in propagation mode the adder will have to wait until the carry of RCA block is generated. The probability of all bits in propagation mode is very less. The idea of proposed model is that to generate the carry from of the input bits, if input bits are not in propagation condition. The input bits are first fed to a XOR gate. In this step we can reuse the XOR gate which is used by the carry prediction unit. The carry is generated from this XOR outputs by using simple ANDOR logic which is a modified version used in Look ahead carry adder. The block which is used to generate carry is carry feed forward block (Fig.3) and the complete architecture is shown in Fig.4. The carry generated by this carry feed forward block is fed to the input of multiplexer. The multiplexer will select the carry generated from carry feed forward block if the input bits are not in propagation mode. So the delay when input bits are not in propagation condition (eq.6) is the sum of time for propagation through XOR gate and time for propagation through AND-OR logic and time for propagation through the skip logic.

T=TXOR+TAND+TOR+TMUX---- Eq.(6)

Delay in conventional adder was the delay of RCA chain that is reduced to delay of AND and OR logic. The proposed model can generate the carry without much delay when the input bits are not in propagation condition. A conventional carry skip adder skip carry only if the bits are in propagation condition, and in other cases the delay will be proportional to number of XOR gate the carry propagates. Since the carry is generated by the carry feed forward mechanism we does not require carry from the RCA chain. So the last full adder in this RCA chain is replaced by two XOR gates. The two AND gates in the last full adder of each RCA block can be removed, which helps to reduce the area. For generating the carry the carry feed forward block requires 4-AND gates and 1-OR gate. Out of this 4-AND gates two will be compensated by removal of two AND gate from last full adder of RCA chain. So only two AND gates and one OR gate will be more in the proposed architecture. The proposed model is more optimized version of CSKA for high speed applications.

In C.I.CSKA carry skip adder the carry is skipped only when all the intermediate inputs are one. This happens only when all input bits are in propagation mode. If this case is not happed the second gate in OAI-AOI compound gate will wait for the carry generated by the RCA block. If the carry starts propagating from the first adder then the carry have to propagate through all eight XOR gates and the delay in this case will the very high.

The proposed CFF-C.I.CSKA uses the same CFF mechanism which is used by CFF-CSKA. That is to generate the carry from of the input bits. The input bits are first fed to a XOR gate. The output of this XOR gate is fed to AND-OR logic to generate the carry of each RCA block. The block which is used to generate carry is carry feed forward block (Fig.3) and the complete proposed architecture is shown in Fig.1. The carry generated by this carry feed forward block is fed to the second gate of AOI-OAI skip logic.

In cases where the input bits are not in propagation mode the CFF block will generate the carry output using ANDOR logic, from the XOR ed input bits. The CFF block requires two three input AND gate and one OR gate. Since the carry provided to each RCA block is zero, the number of AND gates in carry feedforward block can be limited to three. Since the last full adder is modified with out AND gates, the newly added block requires only one AND and one OR gate. By adding 4-XOR gates, one AND and one OR gate per block it is possible to make the C.I.CSKA to skip carry in all conditions.

Since the carry input to each RCA block is zero, the first full adder in each RCA block can be replaced by half adders. Also the carry generated by RCA block is not required because the carry feedforward block generates the carry. Hence the last full adder of RCA chain is modified in such a way that the two

AND gates required for the generation of output carry is removed. The carry generated by CFF block helps the second gate in the AOI-OAI logic to decide the carry of next block. The advantage is that if the bits are not in propagation mode the adder will have to wait until the carry of RCA block is generated. This architecture has so many advantages. In CI-CSKA finding carry output of each stage the carry from incrementation block is not required. In CFF CI-CSKA the carry for next block does not requires the carry from incrementation block as well as from RCA block. The carry from RCA block is generated by CFF block which can reduce the delay (E.7).

TMAX=TXOR+TAND+TOR+TAOI(or)TOAI-------(7)

Thus the carry is calculated based on intermediate results and carry from previous block if all intermediate bits are in propagation condition. If not the carry to next state is find from CFF mechanism. The disadvantage of this model is that for carry calculation we requires 4- XOR gates, three AND gates and one OR gate per block. Since the first full adder in each RCA block is replaced by half adder and the last full adder of RCA chain is modified in such a way that the two AND gates required for the generation of output carry is removed the increase in area due to feed forward block can be compensated to a greater extend. This proposed CFF-CI-CSKA is an area efficient model which compensates for two XOR gates and two AND gates. CFF-CI-CSKA can be used for high speed applications.

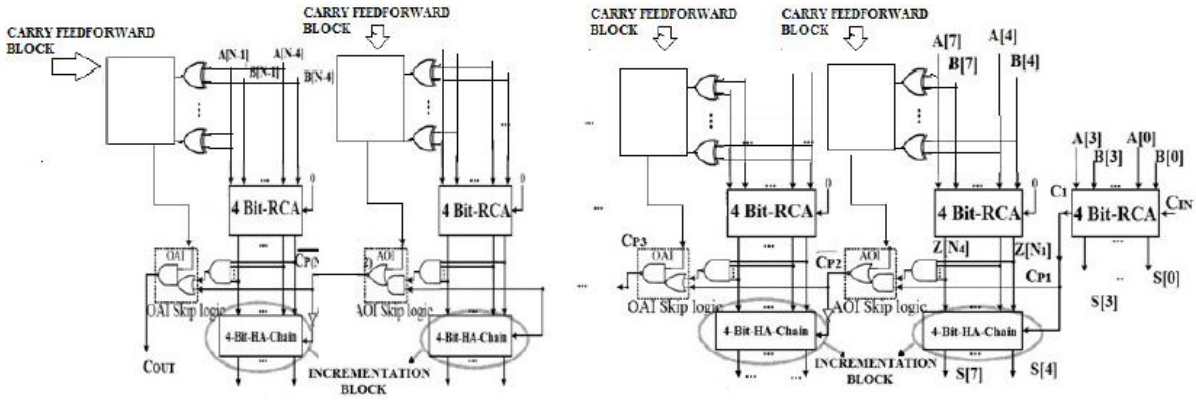

Fig.1: Proposing CI-FOLDED-CSKA ARCHITECTURE

Each 4-bit CSA block is formed by cascading four FAs. As the mirror circuit only produces *Co*. So, if we want to connect two FAs, we need to invert it back to *Co*. However, this can slow down the whole circuit because it will add more inverters to the carry propagation path. Fortunately, if we invert all *A*, *B* and *Ci* signals, from the equations (1, 2, 3), value of *P* does not change while values of both *S* and *Co* are inverted. From this observation, the 4-bit CSA block can be formed as depicted, no inverter is needed at the carry outputs of each FA. The carry is skipped if $P\_ = P3P2P1P0 = 1$ (as shown in Fig. 1). To avoid a high fan-in circuit that can make the $Ion/Io\ f\ f$ ratio low at subthreshold voltages, the logic *P*_ is formed from only 2-input NAND and NOR gates as shown in Fig. 4(a) (instead of an 4-input AND gates). The 2-input NAND and NOR gates are shown in Fig. 4(b) and (c) with transistor sizing in order that they have worst-case pullup and pulldown times equivalent to a minimum size balanced inverter (which was shown in Fig. 2(b)).

The carry out of a 4-bit CSA block is gotten from a 2-input MUX. This MUX uses a transmission gate with buffer (by an inverter) at output as shown in Fig. 6. Benefit of the output buffer here is two-fold. First, it makes *Cout* stronger at output of each 4-bit CSA block. Second, it avoids the case when *C*0 can be skipped and travels through all eight 4-bit blocks without any intermediate buffer. This will seriously degrade the final carry output signal due to weak driving current (so it will not meet the robustness requirement of $VOL\ \_\ 0.1VDD$ and $VOH\ \_\ 0.9VDD$ even at high supply voltages), and also slows down the overall speed of the adder.

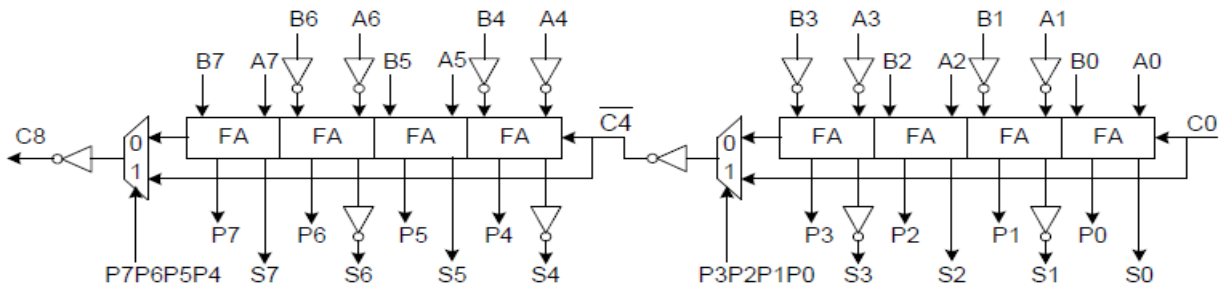

Fig.3: Connection of two 4-bit CSA with carry out inverted to form a 8-bit adder.

However, with this output MUX buffer, the carry out of 4-bit block is inverted. Again, instead of using one more inverter for inverting this carry out before connecting it to the next block, we can build the next 4-bit block with inputs inverted as depicted. These two 4-bit CSA blocks form an 8- bit adder block. The final 32-bit adder is formed by cascading these four 8-bit blocks.

IV. RESULTS AND DISCUSSION

A. Results of Descriptive Statics of Study Variables

TABLE 1: EXISITNG SCHEME:

VDD	POWER(mW)	DELAY
0.7	6.47	4.8
0.8	7.3	4.61
0.9	7.43	4.51
1.0	8.05	4.46
1.1	8.13	4.23

TABLE 2: PROPOSING SCHEME

VDD	POWER(mW)	DELAY
0.7	6.32	4.67
0.8	6.53	4.62
0.9	6.84	4.51
1.0	7.5	4.4
1.1	7.8	4.02

B. Results of Simulation

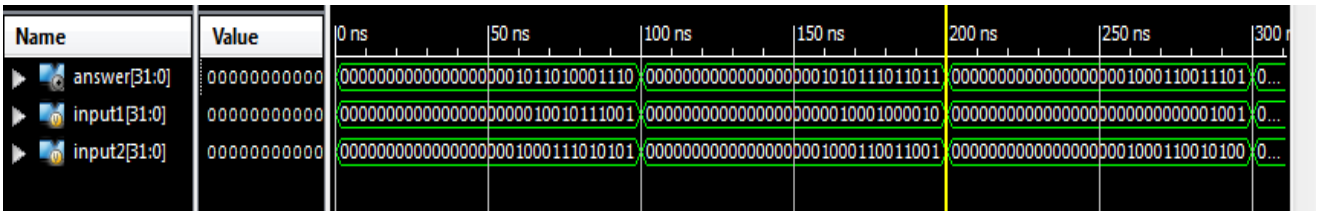

Fig.4: Simulation Results for CSA

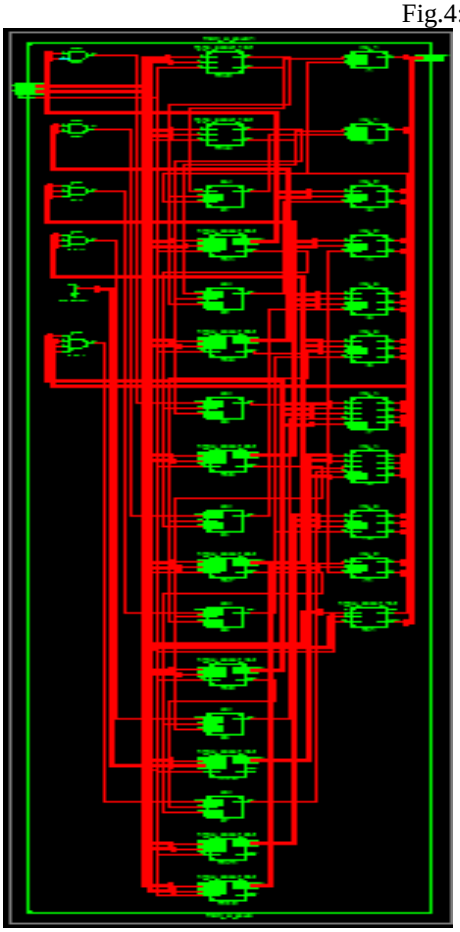

Fig.5: DECRYPTION RTL SCHEMATIC

V. CONLUSION

The conventional carry skip adder can be used to skip carry only if the input bits are in carry propagation condition. In CI-CSKA also the carry can be skipped only if the intermediate results are one. But the proposed structure can skip the carry if the input bits are in carry propagation or carry generation condition. The proposed CFF-CSKA architecture requires only 2-AND gates and one OR gate in addition to skip carry. CFF-CSKA is a speed optimized version of conventional CSKA. The second proposed model was CFF-CI-CSKA. This model incorporates many advantages and can be used for very high speed applications. But the extra circuit added with this architecture consumes large area. Analysis shows that the proposed models consumes more area and power but it has an improved timing. So it can be concluded that the proposed models can be used for high speed applications by the cost of area and power.The adder will have minimum EDP point at a voltage higherthan the transistor threshold. Design and evaluation of an 64-bit adder with minimum EDP point in the subthreshold regionis our future work

VI. REFERENCES

[1]. 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.

[2]. V. G. Oklobdzija, B. R. Zeydel, H. Q. Dao, S. Mathew, and R. Krishnamurthy, "Comparison of highperformance VLSI adders

in the energy-delay space," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 6, pp. 754–758, Jun. 2005.
[3]. Kerur, S.S., Saktivel, R., Kittur, H., Girish, V.A Low power high performance carry select adder (2014) International Journal of Applied Engineering Research, 9 (2), pp. 175-182
[4]. 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.
[5]. M. Lehman and N. Burla, "Skip techniques for high-speed carry propagation in binary arithmetic units," IRE Trans. Electron. Comput., vol. EC-10, no. 4, pp. 691–698, Dec. 1961.
[6]. M. Alioto and G. Palumbo, "A simple strategy for optimized design of one-level carry-skip adders," IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 50, no. 1, pp. 141–148, Jan. 2003.
[7]. High-Speed and Energy-Efficient Carry Skip Adder Operating Under a Wide Range of Supply Voltage Levels. IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 24, NO. 2, FEBRUARY 2016
[8]. M. Lehman and N. Burla, "Skip techniques for high-speed carry propagation in binary arithmetic units," IRE Trans. Electron. Comput. vol. EC-10, no. 4, pp. 691–698, Dec. 1961.
[9]. P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," IEEE Trans. Comput., vol. C-22, no. 8, pp. 786–793, Aug. 1973
[10]. Ragunath, G., Sakthivel, R. Low - power and area - efficient square - Root carry select adders using modified XOR gate (2016) Indian Journal of Science and Technology.
[11]. S.V.Manikanthan, Padmapriya.T, "RECENT TRENDS IN M2M COMMUNICATIONS IN 4G NETWORKS AND EVOLUTION TOWARDS 5G", International Journal of Pure and Applied Mathematics, Vol. 115, No. 8, pp: 623-630, 2017.
[12]. S.V.Manikanthan and V.Rama "Optimal Performance Of Key Predistribution Protocol In Wireless Sensor Networks" International Innovative Research Journal of Engineering and Technology ,ISSN NO: 2456-1983,Vol-2,Issue –Special –March 2017.
[13]. Rajesh.M., and J. M. Gnanasekar. & quot; GC Cover Heterogeneous Wireless Ad hoc Networks.& quot; Journal of Chemical and Pharmaceutical Sciences (2015): 195-200.
[14]. S.V.Manikanthan and T.Padmapriya "Recent Trends In M2m Communications In 4g Networks And Evolution Towards 5g", International Journal of Pure and Applied Mathematics, ISSN NO: 1314-3395, Vol-115, Issue -8, Sep 2017.