www.ijcrt.org © 2022 IJCRT | Volume 10, Issue 10 October 2022 | ISSN: 2320-2882

IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Concurrent Error Detectable Carry Select Adder with Easy Testability For 16-Bits.

MAKALA SAI PRIYA

STUDENT

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

Abstract—A concurrent error detectable adder with easy testability is proposed. The proposed adder is based on a multi-block carry select adder. Any erroneous output of the adder caused by a fault modeled as a single stuck-at fault can be detected by comparing the predicted parity of the sum with the parity of the sum, i.e., the XORed value of the sum bits, and comparing the duplicated carry outputs. The adder is also testable with only 10 input patterns under single stuck-at fault model. This property eases detection of a fault before the occurrence of a second fault. Both the concurrent error detectability to detect erroneous results and the easy testability to find a fault during operation are important for realizing reliable systems. Both the concurrent error detectability and the easy testability of the proposed adder are proven. A 32-bit adder has been designed. Its hardware overhead is about 70 percent. Its concurrent error detectability and 100 percent test coverage through the 10 patterns has been confirmed by fault simulation.

I.INTRODUCTION TO VLSI

1.1 Very-large-scale integration: Very-large-scale integration (VLSI) combines thousands of transistors into one chip. It began within the Seventies wherever the complicated semiconductor and communication technologies were being developed. micro chip is that the basic example of the VLSI .

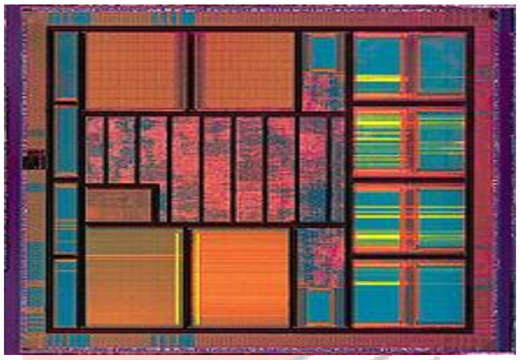


Fig1.1 A VLSI integrated-circuit die

1.2 History

During the 1920's, several inventors tried devices that were meant to regulate this in solid state diodes and convert them into triodes. Success, was a word that is being inhabited till once war II, throughout that the arrange to improve semiconducting material and semiconducting material crystals. With the invention of transistors at Bell labs, in 1947, the sphere of physics have developed a replacement power intense vacuum tubes to solid state devices.

With the invention of the tiny effective semiconductor unit within the era of 50s, engineers had an inspiration of constructing the new advanced circuits.With the rise of the circuits the complexness of understanding the devices square measure surged to associate degree extent. Another one was with the scale of the circuits. Any complicated circuit can largely rely on the speed, likewise laptop. The additional the elements the additional the complexness and the less speed of the system.

Jack Kilby within the state of American state Instruments had return up with an answer to the current drawback in 1958. His plan was to create stone elements of semiconductor material. Kilby given his new plan to his superiors. He was allowed to make a check version of his circuit. In Sept 1958, his initial microcircuit got prepared. though the primary microcircuit was outstanding however had some problems . By creating all the components out of constant block of fabric and adding the metal required to attach them as a layer on high of it, there was no additional want for individual separate elements. No additional wires and elements had to be assembled manually. The circuits may well be created smaller and therefore the producing method may well be machine-controlled. From here the thought of desegregation all elements on one semiconducting material wafer came into existence and that junction rectifier to development in little Scale Integration(SSI) in early Sixties, Medium Scale Integration(MSI) in late Sixties, giant Scale Integration(LSI) and in early Eighties VLSI ten,000s of transistors on a chip (later 100,000s & currently 1,000,000s).

1.3 Developments

The first semiconductor chips control 2 transistors every. succeeding advances additional additional and additional transistors, and, as a consequence, additional individual functions or systems were integrated over time. the primary integrated circuits control solely many devices, maybe as several as 10 diodes, transistors, resistors and capacitors, creating it attainable to fabricate one or additional logic gates on one device.Now well-known retrospectively as small-scale integration (SSI), enhancements in technique junction rectifier to devices with many logic gates, referred to as medium-scale integration (MSI). any enhancements junction rectifier to large-scale integration (LSI), i.e. systems with a minimum of one thousand logic gates. Current technology has emotional way past this mark and today's microprocessors have several several gates and billions of individual transistors.

At just once, there was a shot to call and calibrate varied levels of large-scale integration on top of VLSI. Terms like ultra-large-scale integration (ULSI) were used. however the large variety of gates and transistors out there on common devices has rendered such fine distinctions moot. Terms suggesting bigger than VLSI levels of integration are not any longer in widespread use.

As of early 2008, billion-transistor processors square measure commercially out there. this can be expected to become additional commonplace as semiconductor fabrication moves from this generation of sixty five nm methodes to consecutive forty five nm generations (while experiencing new challenges like inflated variation across process corners). A notable example is Nvidia's280 series GPU. This GPU is exclusive within the indisputable fact that most of its one.4 billion transistors square measure used for logic, in distinction to the Itanium, whose giant semiconductor unit count is essentially thanks to its twenty four MB L3 cache. Current styles, in contrast to the earliest devices, use in depth style automation and automatic logic synthesis to get out the transistors, enabling higher levels of complexness within the ensuing logic practicality. bound superior logic blocks just like the SRAM (Static Random Access Memory) cell, however, square measure still styleed by hand to make sure the very best potency (sometimes by bending or breaking established design rules to get the last little bit of performance by mercantilism stability). VLSI technology is moving towards radical level miniaturisation with introduction of NEMS technology. Alot of issues got to be sorted out before the transition is truly created.

1.4 VLSI Technology

Gone area unit the times once immense computers manufactured from vacuum tubes Saturday buzzing in entire dedicated rooms and will do concerning 360 multiplications of ten digit numbers in an exceedingly second. tho' they were publicized because the quickest computing machines of that point, they sure enough don't stand an opportunity in comparison to the fashionable day machines. modern-day computers have gotten smaller, faster, and cheaper and a lot of power economical each progressing second. however what drove this change? the total domain of computing ushered into a brand new dawn of electronic shrinking with the appearance of semiconductor semiconductor unit by physicist (1947-48) then the Bipolar semiconductor unit by William Shockley (1949) within the Bell Laboratory.

Since the invention of the primary IC (Integrated Circuit) within the sort of a Flip Flop by Jack Kilby in 1958, our ability to pack a lot of and a lot of transistors onto one chip has doubled roughly each eighteen months, in accordance with the Moore's Law. Such exponential development had ne'er been seen in the other field and it still continues to be a serious space of analysis work.

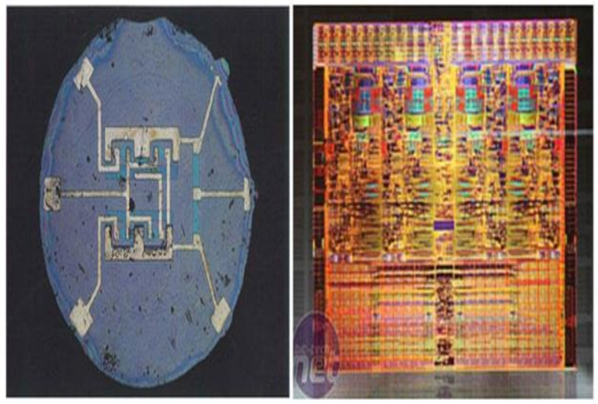


Fig 1.2 A comparison: First Planar IC (1961) and Intel Nehalem Quad Core Die

II.INTRODUCTION TO ADDERS

2.1 Motivation

To humans, decimal numbers area unit simple to understand and implement for acting arithmetic. However, in digital systems, like a micro chip, DSP (Digital Signal Processor)or ASIC (Application-Specific Integrated Circuit), binary numbers area unit a lot of pragmatic for a given computation. this happens as a result of binary values area unit optimally economical at representing several values.Binary adders area unit one in every of the foremost essential logic components at intervals a digital system. additionally, binary adders also are useful in units aside from Arithmetic Logic Units (ALU),such as multipliers, dividers and memory addressing. Therefore, binary addition is crucial that any improvement in binary addition may end up in an exceedingly performance boost for any ADPS and, hence, facilitate improve the performance of the complete system.

The major downside for binary addition is that the carry chain. because the breadth of the input quantity will increase, the length of the carry chain will increase. Figure 2.1 demonstrates associate degree example of associate degree 8- bit binary add operation and the way the carry chain is affected. this instance shows that the worst case happens once the carry travels the longest doable path, from the smallest amount vital bit (LSB) to the foremost vital bit (MSB). so as to boost the performance of carry-propagate adders, it's doable to accelerate the carry chain, however not eliminate it. Consequently, most digital designers usually resort to putting together quicker adders once optimizing a pc design, as a result of they have a tendency to line the essential path for many computations.

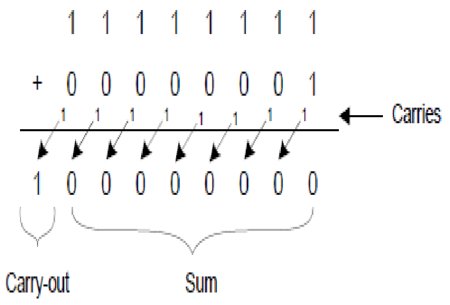


Figure 2.1: Binary Adder Example.

The binary adder is that the essential part in most digital circuit styles together with digital signal processors (DSP) and micro chip knowledge path units. As such, in depth analysis continues to be centered on up the ability delay performance of the adder.In VLSI implementations, parallel-prefix adders area unit best-known to own the simplest performance. Reconfigurable logic like Field Programmable Gate Arrays (FPGAs) has been gaining in quality in recent years as a result of it offers improved performance in terms of speed and power over DSP-based and microprocessor-based solutions for several sensible styles involving mobile DSP and telecommunications applications and a big reduction in development time and price over Application Specific microcircuit (ASIC) styles.

The power advantage is very vital withthe growing quality of mobile and transportable natural philosophy, that build in depth use of DSP functions. However, due to the structure of the configurable logic androuting resources in FPGAs, parallel-prefix adders can have a special performance than VLSI implementations. above all, hottest FPGAs use a fast-carry chain that optimizes the carry path for the straightforward Ripple Carry Adder (RCA).In this paper, the sensible problems concerned in coming up with and implementing tree-based adders on FPGAs area unit delineated . many tree-based adder structures area unit enforced and characterised on a FPGA and compared with the Ripple Carry Adder (RCA) and therefore the Carry Skip Adder (CSA). Finally, some conclusions and suggestions for up FPGA styles to modify higher tree-based adder performance area unit given.

2.2 Carry-Propagate Adders

Binary carry-propagate adders are extensively printed, heavily assaultive issues associated with carry chain drawback. Binary adders evolve from linear adders, that have a delay more or less proportional to the breadth of the adder, e.g. ripple-carry adder (RCA),to logarithmic-delay adder, like the carry-lookahead adder (CLA). There area unit some further performance enhancing schemes, together with the carry-increment adder and therefore the Ling adder which will any enhance the carry chain, however, in terribly massive Scale Integration (VLSI) digital systems, the foremost economical manner of giving binary addition involves utilizing parallel-prefix trees, this happens as a result of they need the regular structures that exhibit exponent delay.

2.3 Research Contributions

The implementations that are developed during this treatise facilitate to enhance the planning of Carry choose adders and their associated computing architectures. This has the potential of impacting several application specific and general purpose pc architectures. Consequently, this work will impact the styles of the many computing systems, still as impacting several areas of engineers and science. during this paper, the sensible problems concerned in coming up with and implementing Carry choose adders on FPGAs area unit delineate. many carry choose adder structures area unit enforced and characterised on a FPGA and compared with the CSLA with Ripple Carry Adder (RCA) and therefore the CSLA with Binary Excess device. Finally, some conclusions and suggestions for rising FPGA styles to modify higher carry choose adder performance area unit given.

III.BINARY ADDER SCHEMES

Adders area unit one amongst the foremost essential parts in digital building blocks, however, the performance of adders become a lot of important because the technology advances. the matter of addition involves algorithms in formal logic and their several circuit implementation. Algorithmically, there area unit linear-delay adders like ripple-carry adders (RCA),which area unit the foremost easy however slowest. Adders like carry-skip adders (CSKA),carry-select adders (CSLA) and carry-increment adders (CINA) area unit linear-based adders with optimized carry-chain and improve upon the linear chain inside a ripple-carry adder. Carry-lookahead adders (CLA) have exponent delay and presently have evolved to parallel-prefix structures. alternative schemes, like Ling adders, NAND/NOR adders and carry-save adders will facilitate improve performance still.

This chapter provides background info on architectures of adder algorithms. within the following sections, the adders area unit characterised with linear gate model, that could be a rough estimation of the quality of real implementation. though this analysis methodology is deceptive for VLSI implementers, such sort of estimation will offer comfortable insight to know the planning trade-offs for adder algorithms.

3.1 Binary Adder Notations and Operations

As mentioned antecedently, adders in VLSI digital systems use mathematical notation. therein case, add is completed bit by bit victimization Boolean equations. think about an easy binary add with 2 n-bit inputs A;B and a one-bit carry-in cinalong side n-bit output S.

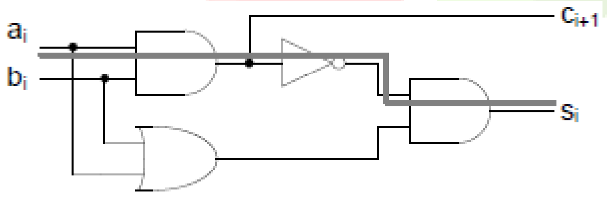


Figure 3.1: 1-bit Half Adder.
Considering true of adding 2 bits, the total s and carry c is expressed

using Boolean operations mentioned higher than.

si = ai^bi

ci+1 = ai.bi

The Equation of ci+1 is enforced as shown in Figure 3.1. within the figure, there's a [*fr1] adder, that takes solely two input bits. The solid line highlights the important path,that indicates the longest path from the input to the output.

Equation of ci+1 is extended to perform full add operation, wherever there's a carry input.

si = ai^ bi ^ ci
ci+1 = ai .bi + ai. ci + bi . ci

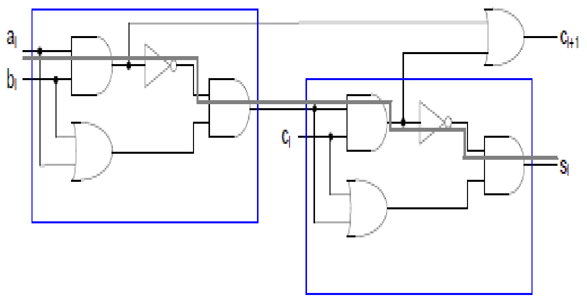


Figure 3.2: 1-bit Full Adder.

A full adder can be built based on Equation above. The block diagram of a 1-bit full adder is shown in Figure 3.2. The full adder is composed of 2 half adders and an OR gate for computing carry-out.

Using Boolean algebra, the equivalence can be easily proven.
To help the computation of the carry for each bit, two binary literals are introduced.
They are called carry generate and carry propagate, denoted by gi and pi. Another literalcalled temporary sum ti is employed as well. There is relation between the inputs and theseliterals.
gi = ai. bi
pi = ai + bi
ti = ai^ bi
where i is an integer and 0 _ i< n.
With the help of the literals above, output carry and sum at each bit can be written as
ci+1 = gi + pi .ci
si = ti^ ci
In some literatures, carry-propagate pi can be replaced with temporary sum ti in order tosave the number of logic gates. Here these two terms are separated in order to clarify theconcepts. For example, for Ling adders, only pi is used as carry-propagate.
The single bit carry generate/propagate can be extended to group version G and P. The following equations show the inherent relations.
Gi:k = Gi:j + Pi:j. Gj-1:k
Pi:k = Pi:j. Pj-1:k
where i : k denotes the group term from i through k.
Using group carry generate/propagate,carry can be expressed as expressed in the following equation.
ci+1 = Gi:j + Pi:j.cj

3.2 Carry-Select Adders (CSLA)

Simple adders, like ripple-carry adders, area unit slow since the carry has got to to travel throughevery full adder block. there's the simplest way to boost the speed by duplicating the hardware dueto the actual fact that the carry will solely be either zero or one. the tactic is predicated on the conditionalsumadder and extended to a carry-select adder. With 2 RCA, every computingthe case of the one polarity of the carry-in, the add may be obtained with a 2x1 multiplexerwith the carry-in because the choose signal. associate example of 16-bit carry-select adder is shown inFigure three.4. within the figure, the adder is sorted into four 4-bit blocks. The 1-bit multiplexorsfor add choice may be enforced as Figure three.5 shows. presumptuous the 2 carry terms area unit used such the carry input is given as a continuing one or 0:

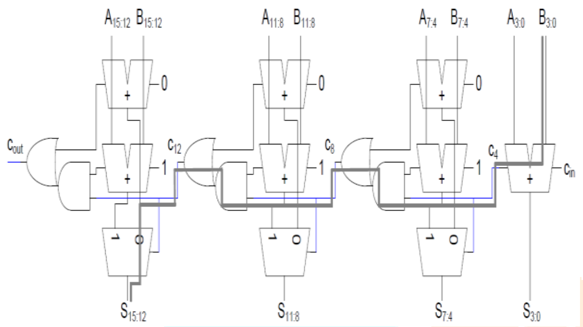


Figure 3.3: Carry-Select Adder.

In Figure 3.3, each two adjacent 4-bit blocks utilizes a carry relationship

ci+4 = c0 i+4 + c1 i+4 . ci

The relationship can be verified with properties of the group carry generate/propagate and c0i+4 can be written as

c0i+4 = Gi+4:i + Pi+4:i . 0
= Gi+4:i

Similarly, c1i+4 can be written as

c1i+4 = Gi+4:i + Pi+4:i . 1
= Gi+4:i + Pi+4:i

Then

c0i+4 + c1i+4 .ci = Gi+4:i + (Gi+4:i + Pi+4:i) .ci
= Gi+4:i + Gi+4:i .ci + Pi+4:i .ci
= Gi+4:i + Pi+4:i .ci
= ci+4

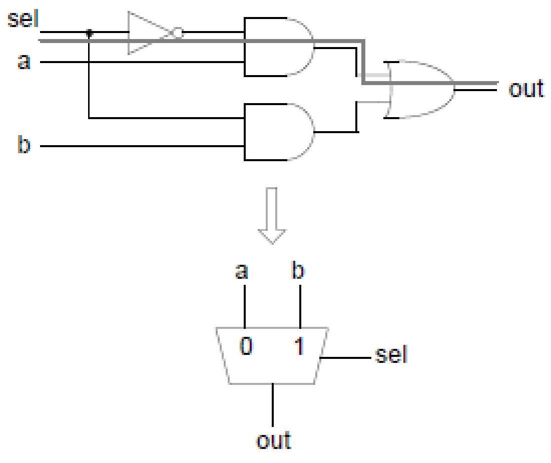


Figure 3.4: 2-1 Multiplexor.

Varying the number of bits in each group can work as well for carry-select adders. temporary sums can be defined as follows.

s0 i+1 = ti+1 .c0i

s1i+1 = ti+1 .c1i

The final sum is selected by carry-in between the temporary sums already calculated.

si+1 = cj.s0i+1 + cj.s1i+1

Assuming the block size is fastened at r-bit, the n-bit adder consists of k teams ofr-bit blocks, i.e. n = r x k. The important path with the primary RCA incorporates a delay of (4r + 5)/\ from the input to the carry-out, and there area unit k - two blocks that follow, every with a delay of4/\ for carry to travel through. the ultimate delay comes from the multiplexor, that incorporates a delay of 5/\, as indicated in Figure 3.4. the whole delay for this CSEA is calculated as

tcsea = 4r + 5 + 4(k - 2) + 5/\
= {4r + 4k + 2}/\

The area can be estimated with (2n - r) FAs, (n - r) multiplexors and (k - 1) AND/ORlogic. As mentioned above, each FA has an area of 9 and a multiplexor takes 5 units ofarea. The total area can be estimated

9(2n - r) + 2(k - 1) + 4(n - r) = 22n - 13r + 2k – 2

IV. PROPOSED MODEL

Configuration

Fig. 2a shows a style of the projected adder. additionally to operands X and Y , it receives their parities PX and pY , i.e., the XORed worth of X and therefore the XORed worth of Y . additionally to total output S, it produces the expected parity note of the total output and 2 carry outputs cn and c0 n. we have a tendency to prohibit the block length nk to be even. Any incorrect output of the adder caused by a fault sculptured as one stuck-at fault are often detected by comparison the expected parity note with the parity of total output S and comparison cn and c0 n.

One inconsistency within the 2 input pairs, i.e., a combine of X and PX or a combine of Y and pY , may also be detected. every addition block except block0 has 2 carry inputs. The addition block receives 2 operands Xk and Yk, and produces total result Sk. additionally to those inputs and output, it receives 2 carry inputs cink and cin0 k, and produces parity of carries pCk and 2 carry outputs coutk and cout0 k. coutk and cout0 k area unit connected to cinkþ1 and cin0 kþ1, severally.

Fig. 2c shows gate-level styles of HA’, INC’, and MUX’. each of HA’ and INC’ have 2 carry outputs to get coincident error detectability. One carry bit is employed for addition, and therefore the alternative is employed for parity prediction. hour {angle|HA|angular distance}’ ANd INC’ area unit designed in order that effects of one stuck-at fault in an INC’ or its ascendant HA ne'er seem in each its total output and one in all its carry signals at the same time as a result of such a faulty operation generates an incorrect total result and a expected parity in line with the incorrect total result. In MUX' and INC', we have a tendency to use XOR gates to boost testability. as a result of XOR gates stop masking of effects of a fault, effects of a fault are often determined simply. In every addition block, AN X-OR circuit is placed for each bit position of INC0 k and INC1 k except the foremost important position and therefore the least important position. For parity output pCk , 2 intermediate candidates area unit calculated within the 2 rows of INCs.

The left MUX' selects one in all them in keeping with the worth of carry input cink, and therefore the X-OR circuit at the output of the MUX' produces the parity of the carry bits together with cin0 k. With the obtained pCk .

Concurrent Error Detectability

Effects of one stuck-at fault in AN INC' or its ascendant HA' ne'er seem in each its add output and one in every of its 2 carry signals at the same time as represented in Section three.1. Therefore, impact of a fault in AN INC' or its ascendant HA' seems on either (1) one in every of the 2 carry signals of the INC', (2) the add signal, or (3) all the 3 signals (two carries and therefore the sum). In any case, any incorrect result caused by a fault are often detected by examination the anticipated parity notation with the parity of S and examination cn and c0 n. just in case (1), one in every of the 2 carry signals of INC' is employed for addition, and therefore the alternative is employed for parity prediction. once the carry signal for parity prediction is incorrect, solely one-bit of carry bits for the parity prediction is affected and therefore the add result's correct. Thus, incorrect results caused by the fault are often detected. once the carry signal for addition is incorrect, we tend to let ck;j be the carry that's stricken by the fault occurred at AN INC'. Then, the error poignant ck;j induces conjointly a slip at the add signal sk;j, and if it's not propagated in any subsequent positions, the error is detected as a result of the carry signal stricken by the fault isn't used for parity prediction.

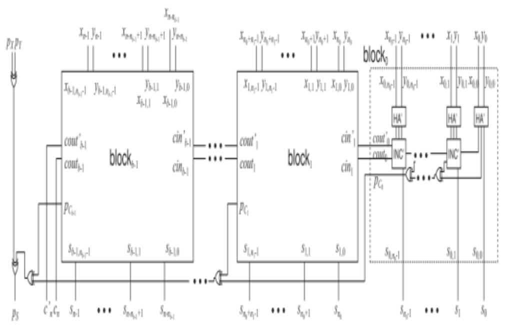


Fig. 2(a). Concurrent error detectable adder with easy testability

In case (2), a small amount of the total result's inverted and also the parity of the total result's completely different from the parity of the proper one. On the opposite hand, the expected parity is calculated properly as a result of all the carry bits used for the prediction ar correct. The impact of a fault is detected by examination the expected parity with the parity of the total result. just in case (3), all of the carry bits and also the total bit from the INC' ar incorrect. as a result of each of the 2 carry bits ar incorrect, there's no inconsistency among the obtained total bits and carry bits used for parity prediction within the higher positions than the position of the faulty INC'. within the lower positions of the INC', tho' carry bits for parity prediction ar correct, the total bit from the INC' is inverted.

Therefore, parity of the total result's completely different from the expected parity. within the adder, every adder block has 2 carry inputs cink and cin0 k. If there's miscalculation on cink then sk;0 are incorrect. more a lot of the error on cink also can induce errors on many alternative total outputs (let United States of America say at alphabetic character total outputs). Also, equally to the arguments given just in case (1), it'll additionally produce errors at alphabetic character carry signals c0 k;i. Thus, there'll be alphabetic character þ one incorrect total outputs and alphabetic character incorrect carry signals c0 k;i, and primarily based to identical arguments as those given just in case (1) these errors also will be detected. The output of associate XOR or a MUX' affects only 1 total or carry bit or the expected parity.

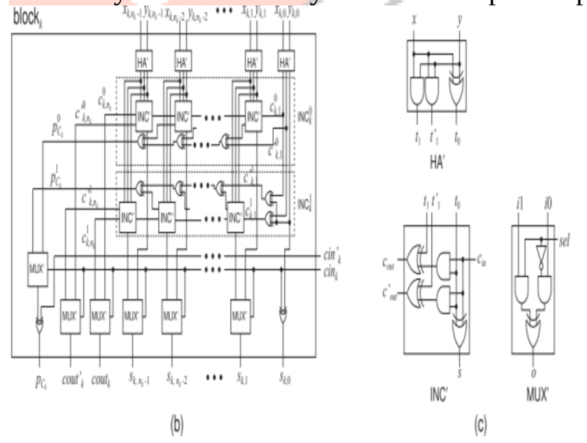


Fig 2: the design of block (k>1) for the adder (b), and the gate-level designs of HA', INC' and MUX' (c).

Therefore, the impact of a fault in them is detected by examination the expected parity with the parity and examination 2 carry outputs of the adder. Note that inconsistency of 1 of the input operands associate degreed its parity input causes an incorrect results of the expected parity as a result of the parity input is employed for the parity prediction. Therefore, it's additionally potential to observe inconsistency of X and post exchange or inconsistency of Y and pY once there are not any faults within the adder. The projected adder is appropriate for systems mistreatment parity-based error detection as shown in Fig. 3. The parity-based error detection of arithmetic circuits was employed in real styles [21], e.g.,

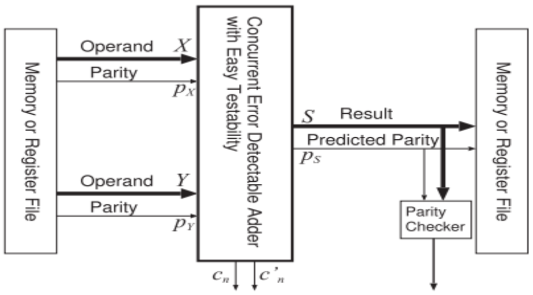


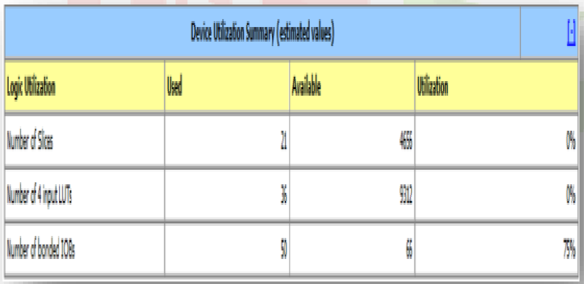
Fig. 3. Example of a datapath circuit with the proposed adder in a system using parity-based error detection

Parities fed from a memory or a register file square measure used as commissary and pY for operands X and Y , and therefore the foreseen parity obtained by the projected adder is employed for the bit of the result. Any inaccurate output of the adder is detected by observant the parity checker of the system and scrutiny 2 carry outputs cn and c0.

V.SIMULATION RESULTS

All the results of arithmetic operation outputs with their RTL schematics which are executed in Xilinx ISE tool are

Adder without MUX



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	21	4656	0%
Number of 4 input LUTs	36	9312	0%
Number of bonded IOBs	50	66	75%

Timing Summary:

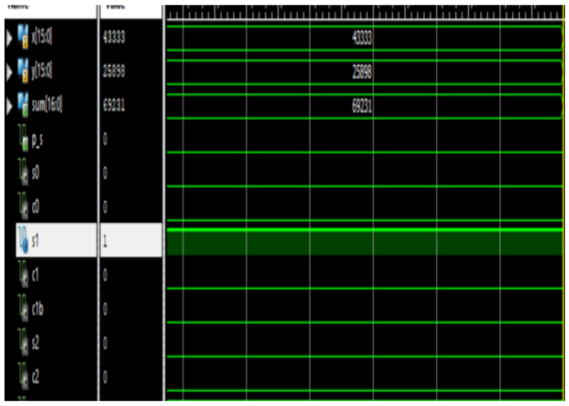
Speed Grade: -5

Minimum period: No path found

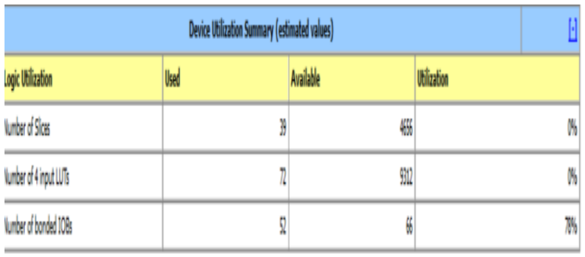
Minimum input arrival time before clock: No path found

Maximum output required time after clock: No path found

Maximum combinational path delay: 22.006ns



Adder with mux



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	39	4656	0%
Number of 4 input LUTs	72	9312	0%
Number of bonded IOBs	52	66	78%

Timing Summary:

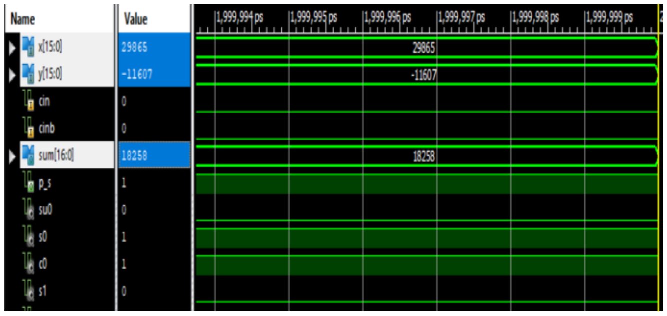
Speed Grade: -5

Minimum period: No path found

Minimum input arrival time before clock: No path found

Maximum output required time after clock: No path found

Maximum combinational path delay: 23.110ns



VI.CONCLUSION

A synchronic error detectable adder with straightforward testability is planned. The planned adder is predicated on a multi-block carry choose adder. It receives parities of 2 operands additionally to the operands, and produces foretold parity of the add result and 2 carry outputs additionally to the add result. Any inaccurate output of the adder by a fault sculptural as one stuck-at fault is detected by parity checking and comparison of the 2 carry outputs. The adder is additionally testable with ten patterns below single stuck-at fault model. This property eases testing of the adder to seek out a fault before the incidence of a second fault. For future intelligent autonomous systems like autonomous cars, each synchronic error detectability and simple testability for detection a fault throughout operation before fatal accidents ar crucial. additional investigations of circuits with each of synchronic error detectability and testability and handling numerous faults like delay faults and soft errors ar fascinating for realizing reliable system.

REFERENCES

[1] J. H. Stathis, “Reliability limits for the gate insulator in CMOS technology,” IBM J. Res. Develop., vol. 46, no. 2/3, pp. 265–286, 2002.

[2] J. Srinivasan, S. Adve, P. Bose, and J. Rivers, “The impact of technology scaling on lifetime reliability,” in Proc. Int. Conf. Depend. Syst. Netw., Jun. 2004, pp. 177–186.

[3] D. K. Schroder and J. A. Babcock, “Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing,” J. Appl. Phys., vol. 94, no. 1, pp. 1–18, 2003.

[4] M. Nicolaidis, “Carry checking/parity prediction adders and ALUs,” IEEE Trans. Very Large Scale Integr. Syst., vol. 11, no. 1, pp. 121–128, Feb. 2003.

[5] B. Kumar and P. Lala, “On-line detection of faults in carry-select adders,” in Proc. Int. Test Conf., Sep. 2003, pp. 912–918.

[6] D. Vasudevan and P. Lala, “A tecinique for modular design of self-checking carry-select adder,” in Proc. 20th IEEE Int. Symp. Defect Fault Tolerance VLSI Syst., Oct. 2005, pp. 325–333.

[7] N. Kito and N. Takagi, “Low-overhead fault-secure parallel prefix adder by carry-bit duplication,” IEICE Trans. Inf. Syst., vol. E96-D, no. 9, pp. 1962–1970, Sep. 2013.

[8] J. Rivers, M. Gupta, J. Shin, P. Kudva, and P. Bose, “Error tolerance in server class processors,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 30, no. 7, pp. 945–959, Jul. 2011.

[9] J. R. Black, “Electromigration - A brief survey and some recent results,” IEEE Trans. Electron Devices, vol. 16, no. 4, pp. 338–347, Apr. 1969.

[10] C. K. Hu, R. Rosenberg, H. S. Rathore, D. B. Nguyen, and B. Agarwala, “Scaling effect on electromigration in on-chip Cu wiring,” in Proc. IEEE Int. Interconnect Technol. Conf., May 1999, pp. 267–269.

[11].Jamal, K., Chari, K. M., & Srihari, P. (2019). Test pattern generation using thermometer code counter in TPC technique for BIST implementation. Microprocessors and Microsystems, 71, 102890.

[12].Jamal, K., Srihari, P., Chari, K. M., & Sabitha, B. (2018). Low power test pattern generation using test-per-scan technique for BIST implementation. ARPN Journal of Engineering and Applied Sciences, 13(8).

[13].Jamal, K., & Srihari, P. (2016). Low power TPC using BSLFSR. International Journal of Engineering and Technology (IJET), 8(2), 759-e.

[14].Jamal, K., & Srihari, P. (2015, January). Analysis of test sequence generators for built-in self-test implementation. In 2015 International Conference on Advanced Computing and Communication Systems (pp. 1-4). IEEE.

[15].Jamal, K., Srihari, P., & Kanakasri, G. (2016). Test Vector Generation using Genetic Algorithm for Fault Tolerant Systems. International Journal of Control Theory and Applications (IJCTA), 9(12), 5591-5598.