ISSN (ONLINE): 2395-695X
ISSN (PRINT): 2395-695X

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Vol.3, Special Issue.24, March 2017

# Bridging Based Multiple Output Transistor Optimization

**A. Jones Immanuel**

**PG student, M.E-VLSI Design, Excel College of Engineering and Technology, Namakkal.**

**P.Narmatha**

**Assistant Professor, Department of ECE, Excel College of Engineering and Technology, Namakkal.**

**Abstract:** Transistor count minimization has a significant importance as large scale integration technology approaches its drastic end line limits. Here, we have presented a computer-aided design synthesis tool that tries to minimize the number of transistors required to implement a given multiple-output logic function. The proposed transistor-level synthesis approach goes beyond the traditional series-parallel design style and allows for extensive bridging. It starts from a sum-of-products expression for each output, allowing also for don't care terms, and produces a transistor network with a small number of transistors to implement all outputs jointly under a user-specified bound on the number of transistors in series to avoid long charge/discharge paths. Experimental results on previously examined multi output functions and case studies (full adder, Gray/binary counter and seven-segment display) demonstrate the benefit of the approach.

**Keywords: Low-power design, Multiple Output Functions, Transistor Optimization, Gate-level approach**

## I.INTRODUCTION

### 1.1 GENERAL

The current trend towards low-power design is mainly driven by two forces, the growing demand for long-life autonomous portable equipment and the technological limitations of high-performance VLSI systems. For the first category of products, low-power is the major goal for which speed and dynamic range might have to be sacrificed. High speed and high integration density are the objectives for the second application category, which has experienced a dramatic increase of heat dissipation that is now reaching a fundamental limit. These two forces are now merging as portable equipment grows to encompass high-throughput computationally intensive products such as portable computers and cellular phones.

An integrated circuit or monolithic integrated circuit (also referred to as IC, chip, or microchip) is an electronic circuit manufactured by the patterned diffusion of trace elements into the surface of a thin substrate of semiconductor material. Additional materials are deposited and patterned to form interconnections between semiconductor devices. Integrated circuits are used in virtually all electronic equipment today and have revolutionized the world of electronics. Computers, mobile phones, and other digital appliances are now inextricable parts of the structure of modern societies, made possible by the low cost of production of integrated circuits. ICs were made possible by experimental discoveries showing that semiconductor devices could perform the functions of vacuum tubes and by mid-20th-century technology advancements in semiconductor device fabrication.

T-Spice is a complete design capture and simulation solution that provides accuracy and convergence with market-proven reliability. To transform your ideas into designs, you must be able to simulate large circuits quickly and with a high degree of accuracy. That means you need a simulation tool that offers fast run times, integrates with your other design tools, and is compatible with industry standards. To transform your ideas into designs, you must be able to simulate large circuits quickly and with a high degree of accuracy. That means you need a simulation tool that offers fast run times, integrates with your other design tools, and is compatible with industry standards

### 1.2 OBJECTIVE OF THE PROJECT

The main objective of my project is to minimize the count of the transistors in the everyday used processing chips and thereby to reduce the power consumption in a conservative manner.

By using the bridging technique the amount of transistors used in the circuit gets reduced with reduced power dissipation and minimized chip size.

### 1.3 SCOPE OF THE PROJECT

The technique used in here reduces not only the count of the transistors used but also reduces the power loss and obviously power consumption. Thereby the corresponding duty cycle of the overall operation also gets reduced.

## II LITERATURE SURVEY

Vinicius N. Possani, Vinicius Callegaro proposed that new graph-based method able to generate optimized transistor networks. Our approach presents a structural algorithm to avoid greedy strategies during part of the generation process. NSP arrangements may be achieved, which leads to a reduction in the total switch count. Different from the approach presented in [11], this new method delivers the networks not only applying transistor sharing, but also considering topological information during the optimization. Finally, this paper improves the previous method described in [15], in which only the NSP kernel concept was effectively introduced.

This paper described a new graph-based method to generate optimized switch networks. The method is able to generate arrangements more general than the usual series-parallel associations. It results in a minimization of switch count when compared to previously published techniques. It is known that a transistor count reduction in a logic gate may lead to better results in terms of signal delay propagation and power consumption. However, these associated gains were not explicitly investigated in this work, and they are left as a future work at gate, library and circuit design level

Giacomo Buonanno, Donatella Sciuto, and Renato Stefanelli proposed two distinct techniques for multiple output functions synthesis at the switching level for static CMOS technology. The methodology can be applied to general multiple output gates. Furthermore, given the design independence of the P-net and the N-net, the two nets can be synthesized using different techniques, Le., one with the A-network topology and the other with the A-topology. Different synthesis algorithms have been analyzed and compared. Those achieving always the optimal results require long computation times. To reduce their computation times, heuristic techniques have been proposed, but in this case, minimal synthesis cannot always be guaranteed. Electrical simulations have shown that the performances of such gate structures are comparable to their corresponding classical implementations. Hence, the area advantage of such gate interconnections has a minimum impact on performances. Algorithms for efficient layout generation of the proposed structures are being studied. The first results show that an improvement similar to the one obtained at the transistor level can be obtained also at the layout level. Work is in progress to define design techniques valid also for those, transistors whose inputs are not directly connected to the transistor gate, e.g., pass transistors

Cristiana Bolchini, Giacomo Buonanno, Donatella Sciuto, and Renato Stefanell proposed a new methodology for multiple-output junctions synthesis at transistor level is presented. The final network produces the defined output values by creating a set of connections among source, ground and output nodes and implementing specific sub circuits constituting each single function. Area minimization and timing constraints are $pres of merit for the quality of the proposed solution. Application results for a set of random and generated functions are also reported.

Synthesis of multiple-output functions provides cost effective results when the outputs are implemented by sharing the sub circuits constituting each one of them. The sharing is provided by implementing a global network to generate all outputs either by direct connection to source and ground or through connections to other "supplied" outputs. This strategy aims at minimizing the cost of the final implementation of the function, i.e. the number of transistors at the CMOS level, also guaranteeing the network performances in terms of output signal delays. Results show that the new methodology is competitive since it is characterized by a low number of transistors with respect to the traditional approach with which timing performances are comparable, and provides a faster network than SIS, with respect to which transistor costs are only slightly better (although the improved algorithm presently under development would further reduce the number of transistors).

Giacomo Buonanno, Donatella Sciuto, and Renato Stefanelli proposed the design methodology of static CMOS gates for multiple output junctions. Two techniques for minimization of multiple output functions at the switching level are introduced. These techniques are based on innovative transistor interconnection structures named Delta and Lambda networks. Design examples on double output functions are provided. It is shown that the two techniques can be combined together, if necessary, to obtain further area reductions.

They also have presented two distinct techniques for multiple output functions synthesis at the switching level for static CMOS technology. The methodology, although described only by simple two output functions, can be applied to general multiple outputs gates. Furthermore, given the design independence of the P-net and the N-net, the two nets can be synthesized using different techniques, i.e. one with the Delta network topology and the other with the Lambda topology. Simulations are being made in order to study the effects of multiple transitions of the inputs and delay propagations on the outputs. Work is in progress to define design techniques valid also for those transistors whose inputs are not directly connected to the transistor gate, e.g. pass transistors.

Alan Mishchenko, Satrajit Chatterjee and Robert Brayton have presented a technique for preprocessing combinational logic before technology mapping. The technique is based on the representation of combinational logic using And-Inverter Graphs (AIGs), a networks of two-input ANDs and inverters. The optimization works by alternating DAG-aware AIG rewriting, which reduces area by sharing common logic without increasing delay, and algebraic AIG balancing, which minimizes delay without increasing area. The new technology-independent flow is implemented in a public-domain tool ABC. Experiments on large industrial benchmarks show that the proposed methodology scales to very large designs and is several orders of magnitude faster than SIS and MVSIS while offering comparable or better quality when measured by the quality of the network after mapping.

Dimitri Kagaris and Themistoklis Haniotakis proposed that the number of transistors required for the implementation of a logic function is a fundamental consideration in digital VLSI design. While the determination of a series-parallel implementation can be straightforward once a simplified Boolean expression of the function is available, this may not be an optimum solution. In this paper, a methodology is developed for minimizing the number of transistors that starts from a sum-of-products expression and utilizes non-series-parallel structures. Experimental results demonstrate the efficiency of the approach.

Thomas C. Bartee proposed an important step in the design of digital machines lies in the derivation of the Boolean expressions which describe the combinational logical networks in the system. Emphasis is generally placed upon deriving expressions which are minimal according to some criteria. A computer program has been prepared which automatically derives a set of minimal Boolean expressions describing a given logical network with multiple-output lines. The program accepts punched cards listing the in out relations for the network, and then prints a list of expressions which are minimal according to a selected one of three criteria. This paper describes the basic design procedure and the criteria for minimality

Alan Mishchenko, Satrajit Chatterjee, and Robert K. Brayton proposed several orthogonal improvements to the state-of-the-art lookup table (LUT)-based field programmable gate array (FPGA) technology mapping. The improvements target the delay and area of technology mapping as well as the runtime and memory requirements. 1) Improved cut enumeration computes all *K*-feasible cuts, without pruning, for up to seven inputs for the largest Microelectronics Center of North Carolina benchmarks. A new technique for on-the-fly cut dropping reduces, by orders of magnitude, the memory needed to represent cuts for large designs. 2) The notion of cut factorization is introduced, in which one computes a subset of cuts for a node and generates other cuts from that subset as needed. Two cut factorization schemes are presented, and a new algorithm that uses cut factorization for delay-oriented mapping for FPGAs with large LUTs is proposed. 3) Improved area recovery leads to mappings with the area, on average, 6%smaller than the previous best work while preserving the delay optimality when starting from the same optimized netlists. 4) Lossless synthesis accumulates alternative circuit structures seen during logic optimization. Extending the mapper to use structural choices reduces the delay, on average, by 6% and the area by 12%, compared with the previous work, while increasing the runtime 1.6 times. Performing five iterations of mapping with choices reduces the delay by 10% and the area by 19% while increasing the runtime eight times. These improvements, on top of the state-of-the-art methods for LUT mapping are available in the package ABC

Vinicius Neves Possani, Vinicius Callegaro,André I. Rei, Renato P. Ribas, Felipe de Souza Marques, and Leomar Soares da Rosa Jr proposed an effective way of improving VLSI circuits. This paper proposes a novel method to automatically generate networks with minimal transistor count, starting from an irredundant sum-of-products expression as the input. The method is able to deliver series–parallel (SP) and non-SP switch arrangements, improving speed, power dissipation, and area of CMOS gates. Experimental results demonstrate expected gains in comparison with related approaches. This paper described an efficient graph-based method to generate optimized transistor (switch) networks. The approach generates more general arrangements than the usual SP associations. Experimental results demonstrated a significant reduction in the number of transistor needed to implement logic networks, when compared with the ones generated by existing related approaches. It is known that the transistor count minimization in CMOS gates may improve the performance, power dissipation, and area of digital ICs. In a general point-of-view, the proposed method produces efficient switch arrangements quite useful to be explored by different IC technologies based on switch theory

## II1 EXISTING SYSTEM

CMOS circuits are constructed in such a way that all PMOS transistors must have either an input from the voltage source or from another PMOS transistor. Similarly, all NMOS transistors must have either an input from ground or from another NMOS transistor. The composition of a PMOS transistor creates low resistance between its source and drain contacts when a low gate voltage is applied and high resistance when a high gate voltage is applied. On the other hand, the composition of an NMOS transistor creates high resistance between source and drain when a low gate voltage is applied and low resistance when a high gate voltage is applied. CMOS accomplishes current reduction by complementing every nMOSFET with a pMOSFET and connecting both gates and both drains together. A high voltage on the gates will cause the nMOSFET to conduct and the pMOSFET to not conduct, while a low voltage on the gates causes the reverse. This arrangement greatly reduces power consumption and heat generation. However, during the switching time, both MOSFETs conduct briefly as the gate voltage goes from one state to another. This induces a brief spike in power consumption and becomes a serious issue at high frequencies.

A CMOS transistor (or device) has four terminals: gate, source , drain , and a fourth terminal that we shall ignore until the next section. A CMOS transistor is a switch. The switch must be conducting or on to allow current to flow between the source and drain terminals (using open and closed for switches is confusing—for the same reason we say a tap is on and not that it is closed ). The transistor source and drain terminals are equivalent as far as digital signals are concerned—we do not worry about labeling an electrical switch with two terminals. Therefore what happens when an input is connected to both a PMOS transistor and an NMOS transistor is being explained as follows.

When the voltage of input A is low, the NMOS transistor's channel is in a high resistance state. This limits the current that can flow from Q to ground. The PMOS transistor's channel is in a low resistance state and much more current can flow from the supply to the output. Because the resistance between the supply voltage and Q is low, the voltage drop between the supply voltage and Q due to a current drawn from Q is small. The output therefore registers a high voltage. On the other hand, when the voltage of input A is high, the PMOS transistor is in an OFF (high resistance) state so it would limit the current flowing from the positive supply to the output, while the NMOS transistor is in an ON (low resistance) state, allowing the output from drain to ground. Because the resistance between Q and ground is low, the voltage drop due to a current drawn into Q placing Q above ground is small. This low drop results in the output registering a low voltage. In short, the outputs of the PMOS and NMOS transistors are complementary such that when the input is low, the output is high, and when the input is high, the output is low. Because of this behavior of input and output, the CMOS circuit's output is the inverse of the input.

The power supplies for CMOS are called $V_{DD}$ and $V_{SS}$, or $V_{CC}$ and Ground (GND) depending on the manufacturer. $V_{DD}$ and $V_{SS}$ are carryovers from conventional MOS circuits and stand for the drain and source supplies. These do not apply directly to CMOS, since both supplies are really source supplies. $V_{CC}$ and Ground are carryovers from TTL logic and that nomenclature has been retained with the introduction of the 54C/74C line of CMOS.

## IV PROPOSED SYSTEM

In the proposed system, the transistor count of the circuit is being reduced with the bridging technique. Most existing transistor-level techniques concern single-output functions (sometimes referred to also as “super gate” design). Some propositions of how to handle multiple-output functions at the transistor level have also been made in the past. In this paper presented is a complete technique that has been developed for multiple-output transistor-level synthesis. The demand for high speed processing has been increasing as a result of expanding computer and signal processing applications. One of the key arithmetic operations in such application is multiplication, and the development of fast multiplier circuits has been a subject of interest over decades. Reducing time delay and power dissipation are very essential requirements for many applications. Achieving these critical requirements may result, in some cases, in increasing the area of the design by a considerable factor. This technique, computes a low-cost placement for the next product term by trying to place it on a “bridge” between two already constructed branches of the transistor network, while at the same time avoiding the creation of any invalid paths for any of the outputs in the network.

The tool can also be parameterized to find transistor configurations with a user-specified bound on the number of transistors in series in order to avoid long delays. An additional benefit of the approach is that it allows the handling of don’t cares (for an initial treatment of don’t cares in transistor-level design. The corresponding diagram for the Transistor level is shown below:

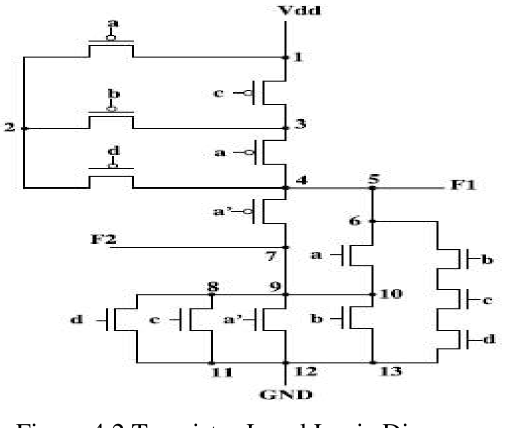

Figure 4.2 Transistor Level Logic Diagram

An FADD has two 1-bit number inputs *a* and *b* and a carryin *c*, and produces a 1-bit sum Sum and a carry-out *C*out. The sum-of-minterms expressions for the outputs are respectively,Sum*(a, b, c) = _m(*1*,* 2*,* 4*,* 7*)* and *C*out*(a, b, c) = m(*3*,* 5*,* 6*,* 7*)*. A manually optimized design computes *C*out as *C*out = *ab* + *ac* + *bc* and then reuses *C*out to compute Sum as Sum = *abc*+*(a*+*b*+*c)(C*out*)* This yields a standard CMOS implementation with 28 transistors [22] (implementations with transmission gates or pass transistor logic are not considered here) with the maximum number of transistors in series being 4. The application of the technique on the FADD with the number of transistors in series bound set to *H* = 3 yielded the CMOS implementation as shown in Fig. 9. As shown by the analytical listing of the conducting paths in Table I, the obtained logic equation for *C*out is *C*out = *ac*+*abc*' +*a*'*bc* and for Sum it is Sum = *a*'*bc*' +*a*'*b*'*c*+*ab*'*c*' +*abc*. This implementation has a total of 22 transistors (11 for the pMOS network and 11 for the nMOS) excluding negated literals. We note that the system does not differentiate between negated and non-negated literals in the same way that, for example, the K-map method does not differentiate between them either in computing the minimum two-level expression. That is,the system does not try to explicitly minimize the number of negated literals in the resulting network graph. Here three inverters (six transistors in CMOS implementation) are needed which brings the total number of transistors to 22 + 6 = 28, which is the same as the manually optimized implementation. However, the complements of *a* and *b* are usually available as they are normally the outputs of registers/flip-flops. In that case, only one additional inverter is needed to obtain the complement of *C*out so that the same FADD can be used in subsequent cascading stages. This brings the total transistor count to 22+2 = 24 yielding a reduction of four transistors (14.3%). The maximum number of transistors in series is 4 (due to the inverter for the complement of *C*out) which is the same as the number of transistors in series in the manually optimized implementation.

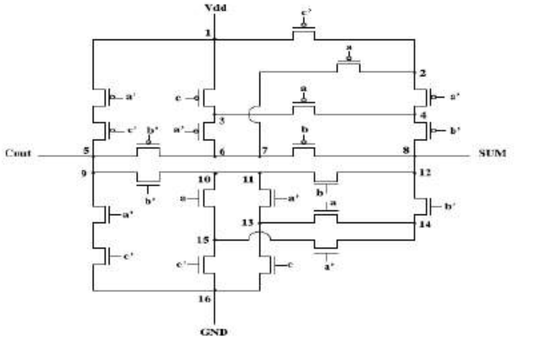

Figure 4.4 Reduced transistor count full adder circuit diagram

## V. SIMULATION

The software used here is T-Spice ,which is a complete design capture and simulation solution that provides accuracy and convergence with market-proven reliability. To transform our ideas into designs, we must be able to simulate large circuits quickly and with a high degree of accuracy. That means we need a simulation tool that offers fast run times, integrates with our other design tools, and is compatible with industry standards. To transform our ideas into designs, we must be able to simulate large circuits quickly and with a high degree of accuracy. That means we need a simulation tool that offers fast run times, integrates with our other design tools, and is compatible with industry standards. T-Spice lets us precisely characterize circuit behavior using virtual data measurements, Monte Carlo analysis, and parameter sweeping. For greater efficiency and productivity, T-Spice puts us in control over our simulation process with an easy-to-use graphical interface and a faster, more intuitive design environment

## SIMULATION OUTPUT

The following image shows the conventional circuit diagram of a full adder with an increased number of transistor count in its circuit, which depicts a complex appearance of the circuit. Here the complete transistor assembly increases the complexity.

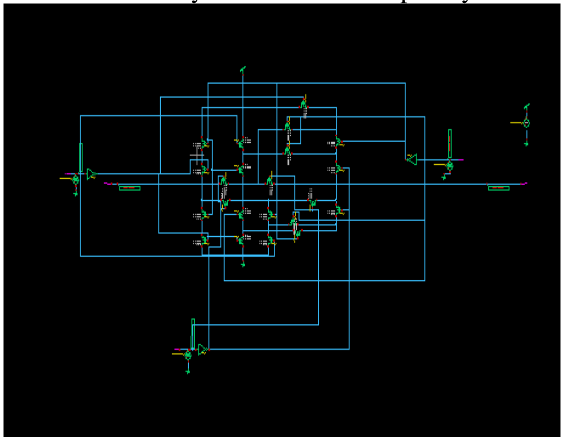

**Figure 6.1 Conventional circuit of full adder**

The following shows the gate level implementation of the full adder circuit, which shows the simplicity of the structure and the proficiency of the output.

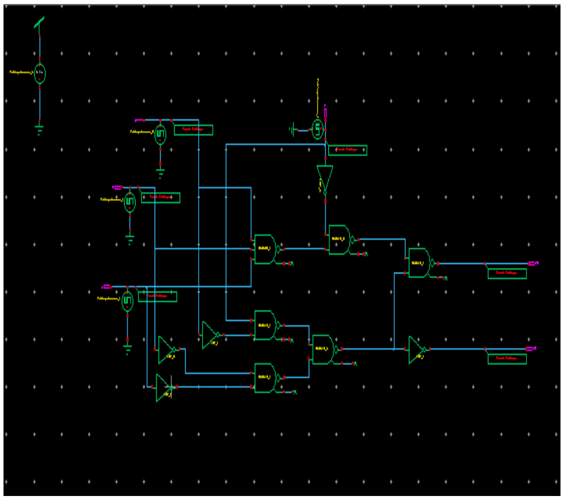

**Figure 6.3 Multioutput full adder circuit**

## VI. CONCLUSION

Hence the issue of customized multiple-output transistor-level synthesis with reduced transistor counts which is very important as integration technology reaches its limits has been addressed. Also presented is a CAD tool that starts from a sum-of-products expression with possible don’t care terms for each output and tries to minimize the number of transistors needed jointly for all outputs under a user-specified bound $H$ on the number of transistors in series. Experimental results show that the system can obtain transistor networks with significant savings over mapped networks obtained by gate level tools such as ABC as well as networks obtained by other approaches. In future work, I have planned to use the tool in conjunction with a clustering/partitioning method to resynthesize multi-input/multi-output subcircuits of a given circuit for reduced transistor count implementations.

## VII. REFERENCES

[1] P. Agrawal, V. D. Agrawal, and N. N. Biswas (2005), “Multiple output minimization,” in *Proc. 22nd Design Autom. Conf.*, Piscataway, NJ, USA, pp. 674–680.

[2] S. Bampi and R. Reis (2011), “Challenges and emerging technologies for system integration beyond the end of the roadmap of nano-CMOS,” in *VLSI-SoC: Technologies for Systems Integration* (IFIP Advances in Information and Communication

Technology), vol. 360. Berlin, Germany: Springer, pp. 21–33.

[3] T. C. Bartee (2002), "Computer design of multiple output logical networks," *IRE Trans. Electron. Comput.*, vol. 10, no. 1 pp. 21–30.

[4] L. A. M. Bennett (2000), "Synthesis of multioutput logic networks using spectral techniques," *IEE Proc. Comput. Digit. Tech.*, vol. 142, no. 4, pp. 241–248.

[5] Berkeley Logic Synthesis and Verification Group. (2014). *ABC: A System for Sequential Synthesis and Verification*.

[6] N. N. Biswas and B. Gurunath (1986), " An algorithm for the optimal minimization of programmable logic arrays," *Int. J. Electron.*, vol. 60, no. 6, pp. 709–725.

[7] C. Bolchini, G. Buonanno, D. Sciuto, and R. Stefanelli (1995), "A new switching-level approach to multiple-output functions synthesis," in *Proc. Int. Conf. VLSI Design*, New Delhi, India, pp. 125–129.

[8] R. K. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, and A. R. Wang (2007), "MIS: A multiple-level logic optimization system," *IEEE Trans.Comput.-Aided Design Integr. Circuits Syst.*, vol. 6, no. 6, pp. 1062–1081.

[9] R. K. Brayton, G. D. Hachtel, C. T. McMullen, and A. L. Sangiovanni-Vincentelli (2009), *Logic Minimization Algorithms for VLSI Synthesis*. Boston, MA, USA: Kluwer Academic.

[10] R. K. Brayton, G. D. Hachtel, and A. L. Sangiovanni-Vincentelli (1990), "Multilevel logic synthesis," *Proc. IEEE*, vol. 78, no. 2, pp. 264–300.

[11] G. Buonanno, D. Sciuto, and R. Stefanelli (1994), "Innovative structures for CMOS combinational gates synthesis," *IEEE Trans. Comput.*, vol. 43, no. 4, pp. 385–399.

[12] G. Buonanno, D. Sciuto, and R. Stefanelli (1999), "Synthesis of multiple outputs CMOS gates," in *Proc. IEEE Int. Symp. VLSI Design*, New Delhi, India, pp. 51–56.

[13] S.-W. Cheng (2010), "Over complementary MOS logic for don't care conditions," in *Proc. IEEE Asia Pac. Conf. Circuits Syst.*, Kuala Lumpur, Malaysia, pp. 700–703.

[14] O. Coudert (1994), "Two-level logic minimization: An overview," *Integr. VLSI J.*, vol. 17, no. 2, pp. 97–140.

[15] L. S. da Rosa, Jr., *et al.* (2004), "Fast disjoint transistor networks from BDDs," in *Proc. 19th Annu. Symp. Integr. Circuits Syst. Design*, Minas Gerais, Brazil, pp. 137–142.

[16] L. S. da Rosa, Jr., F. R. Schneider, R. P. Ribas, and A. I. Reis (2009), "Switch level optimization of digital CMOS gate networks," in *Proc. IEEE Int. Symp. Qual. Electron. Design*, San Jose, CA, USA, pp. 324–329.

[17] M. R. Dagenais, V. K. Agarwal, and N. C. Rumin (1986), "McBOOLE: A new procedure for exact logic minimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 5, no. 1, pp. 229–238.

[18] G. De Micheli (1994), *Synthesis and Optimization of Digital Circuits*. New York, NY, USA: McGraw Hill.

[19] B. Gurunath and N. N. Biswas (1984), "An algorithm for multiple output minimization," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 8, no. 9, pp. 1007–1013, Sep.

[20] S. J. Hong, R. G. Cain, and D. L. Ostapko (2000), "MINI: A heuristic approach for logic minimization," *IBM J. Res. Develop.*, vol. 18, no. 5, pp. 443–458

## Biography

- **A. Jones Immanuel** is pursuing, Master of Engineering in the discipline of VLSI Design at Excel College of Engineering and Technology, under Anna University, Chennai, India. He did his Bachelor of Engineering in the discipline of Electronics and Communication Engineering at Excel College of Engineering and Technology, Namakkal, under Anna University, Chennai, India.

He has published and presented a number of technical papers in National and International journals and Conferences. He is doing minor research works on various fields like Signals and Systems, Electromagnetics and Remote Sensing, and Microelectronics and Quantum Electronics. He is BEC (Business English Certificate) certified student from Cambridge University. He has obtained outstanding student award, Best Speaker-ICTACT Youth Talk award, Best Event Manager award and several other awards. He has also been the Student's Council President of the Department of Electronics and Communication Engineering of his college, and has also been the Vice-President of the Excel Rotarct club of his college.