I J C T A, 9(4), 2016, pp. 1793-1802

Remote Digital Circuit Design and Functionality Verification Through WIFI

R. Bharathikannan[1], S. Ravi[2], M. Anand[3]

ABSTRACT

In [12], the functionality verification of library circuits is done through WIFI. In [12] predefined library circuits are verified through WIFI. The aim of this paper is to build and verify digital logic circuits through WIFI from remote area. Boolean expression with corresponding input values are used to build the required functionality at remote end. The Boolean expression is checked for syntax error on both Android processor and Arduino board. The proposed technique solves the verification problem by function extraction of digital circuit. To avoid overloading a digital circuit Arduino preset limits to the source voltages which are accessible to digital circuits. Scalability is achieved by dynamic reusability of library circuits.

Keywords: Arduino, WIFI, Build Circuit, Verification, Boolean Expression.

1. INTRODUCTION

Simulator only replaces the physical digital circuit. But Physical digital circuit is more important. Most existing system is used for fixed digital circuit design, the proposed system remotely assemble digital logic circuit simultaneously from library circuits in much the same way as they do in traditional designing process of digital circuit. Android processor controls arduino process by providing Boolean expression. Virtual keyboard in android processor are used to control and read the output from digital circuits by means of remote control. The time-sharing method used allows simultaneous access of Arduino processor through WIFI. The Arduino Digital board is always accessible through WIFI. Remotely build digital circuit enables user to learn the difference between real digital circuit and simulators based on mathematical models. The proposed system is asynchronous in time and space. The proposed system uses an efficient algebraic model of the circuit, with logic gates represented by algebraic expressions, while correctly modeling signals as Boolean variables. Build digital circuit remotely from a distance without involving wires or conductors in the build process is proposed.

2. RELATED WORKS

[1] In this article, a self-reconfigurable constant multiplier suitable for LUT-based FPGAs able to reload the constant in runtime is proposed. The configuration time of the proposed architecture is shorter than the partial reconfiguration times required by FPGA devices. Thus, the multiplier proposed in this paper could replace the fixed point multiplier in the floating-point architecture, being only necessary to provide the value of the constant in a floating-point format where there is information of both mantissa and exponent.

[2] In this paper augment test generation procedures to detect faults using traditional steady state Boolean analysis is proposed. The cause for the non-existence of tests for the faults not detected in benchmark

[1] Research Scholar, ECE Department, Dr. M.G.R. Educational and Research Institute, Chennai, *Email: bharathikannan_r@yahoo.co.in*

[2] Professor & Head, ECE Department, Dr. M.G.R. Educational and Research Institute, Chennai, *Email: ravi_mls@yahoo.com*

[3] Professor, ECE Department, Dr. M.G.R. Educational and Research Institute, Chennai, *Email: harshini.anand@gmail.com*

circuits is investigated. The results are presented on larger ISCAS-89 benchmark circuits to illustrate the effectiveness of the proposed methods to generate tests to detect TSOP faults and the results of analysis for the non-existence of tests for the remaining faults undetected by Boolean tests.

[3] In this paper Boolean difference is used to shown how the Boolean difference is used to analyze the effect of errors on the outputs of logic circuits. Examples are given of error detection problems, analysis of redundant logic, and the generation of diagnostic sequences. A related point is that error analysis is reduced to the simplification of a Boolean expression. This simplification operation is one which is well known to logic designers. The result is that now it is possible to do an analysis of errors in logic with very little necessity of learning new mathematical techniques.

[4] In this paper a new approach to perform functional testing that eliminates the need to have a pre-existing model of a circuit board while automating much of the development process. The resulting test approach exhibits massive parallelism combined with extremely compact size that facilitates novel testing approaches not possible with current generation or planned test equipment. The result of this investigation is the conceptualization of a test methodology where we bring nanoscale probes and measurement equipment (e.g., logic state detection, DC, AC, frequency, pulse, etc.) directly to the IC and circuit board signal path.

[5] In this book, chapter 3 design for some fundamental functions circuits of small size and small depth is discussed. The design methods are important, since they are quite general. In order to estimate the value of digital circuits for all functions $f \in Bn\ m\ n - 1$ gates and depth (log n) are mentioned.

[6] The paper presents an algebraic approach to functional verification of gate-level, integer arithmetic circuits. Simulation is performed on arithmetic circuits synthesized and mapped onto standard cells using ABC system. The results demonstrate scalability of the method to large arithmetic circuits, such as multipliers, multiply-accumulate, and other elements of arithmetic data paths with up to 512-bit operands and over 2 Million gates. Solving the verification problem for highly optimized bit-level circuits, synthesized with commercial tools, remains a challenge, as these tools are more aggressive in removing such redundancy.

[7] In this book, standard functional building blocks for digital design are discussed. How programmable logic is used to simplify designs, reduce cost, and accommodate last minute modifications is described. This book defines how PLD and FPGA used for design a circuit by writing sort of program. Boolean expression to logic circuit is described in this book.

[8] In this chapter other kinds of relations (these will all be binary relations here), particularly ones that impose an order of one sort or another on a set is explored. This will used to investigate certain order-structures (posets, lattices) and to introduce an abstract type of algebra known as Boolean Algebra. The Quine-McCluskey method of simplifying Boolean functions is described. The principle of duality of Boolean expressions is described.

[9] Arduino board architecture, pin details interfacing with WIFI and communicate with other devices are discussed.

[10] Android app development, layout, coding example to develop android processor is described.

[11] In this paper, reachability analysis of sequential circuits is proposed. This paper presents a novel semi-formal approach which combines the advantages of simulation and formal methods to traverse the state space of the FSMs. Algorithm proposed in this paper produces a large amount of parallel random pattern simulation can efficiently traverse partial state space, and formal methods serves as the important role to compensate the insufficiency of the first stage. The measurement of controllability of a gate, as guidance in the backward logic implication, is achieved.

3. PROPOSED SYSTEM

Arduino board and android processor connects with WIFI. The library circuits to build digital circuits are connected with Arduino digital board. The App allows for emulating maximum of 6 input digital circuits by providing the Boolean expression. The App sends this data to the Arduino board through WIFI channel and the Arduino board dynamically constructs this circuit using the library circuits connected to it and inputs could be provided through the android processor and the outputs could be verified.

BLOCK DIAGRAM

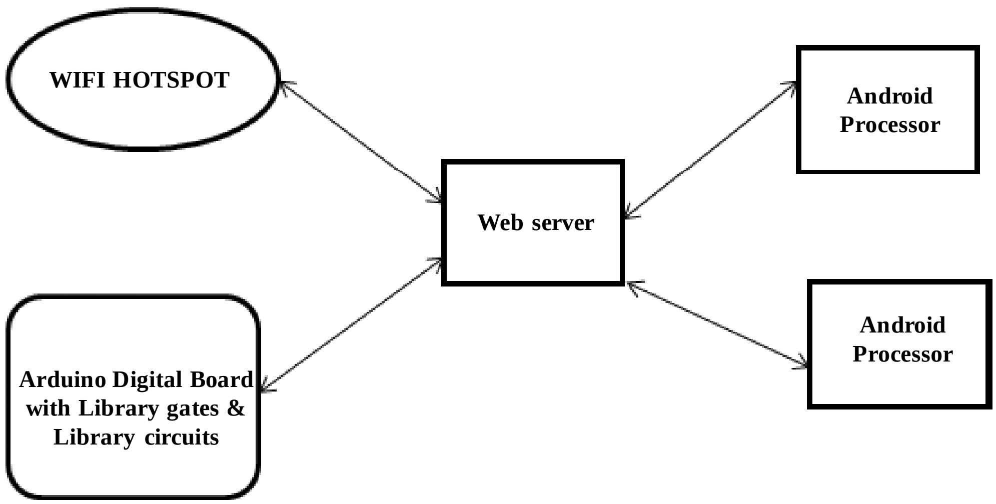


Figure 1: Block Diagram of proposed system

4. PROPOSED SYSTEM COMPONENTS

4.1. Android processor

On the Android processor, there are buttons to change Boolean expression and input for the digital circuits, and it is easily allows user to change them. Evaluate button bring up output from Arduino board and a virtual keyboard will appear for modifying input values. The URL consist the hostname or IP address of WIFI.

4.2. Arduino board

Arduino board is a switching matrix consist relays and instrument connectors. This make connection between the pins of the Arduino digital board and the library gates are controlled from the Android processor. It is possible to assemble a circuit by engaging a number of relays. Arduino board does the wiring to assemble the digital logic circuits.

4.3. WIFI Hotspot

The WIFI Hotspot is provided by the WIFI Module that is connected to the Arduino board. WIFI module is affordable to Arduino. It uses SPI communication for fast data transmission. WIFI board consist regulator and level shifter to allow 3 or 5V logic level.

4.4. SPI Connection

For SPI connection between WIFI module and Arduino is enabled by connecting CLK, MISO (Data out of the microcontroller), MOSI (Data in from a controller) and CS pins of WIFI module to Arduino pins respectively Digital 13, Digital 12, Digital 11 and Digital 10.

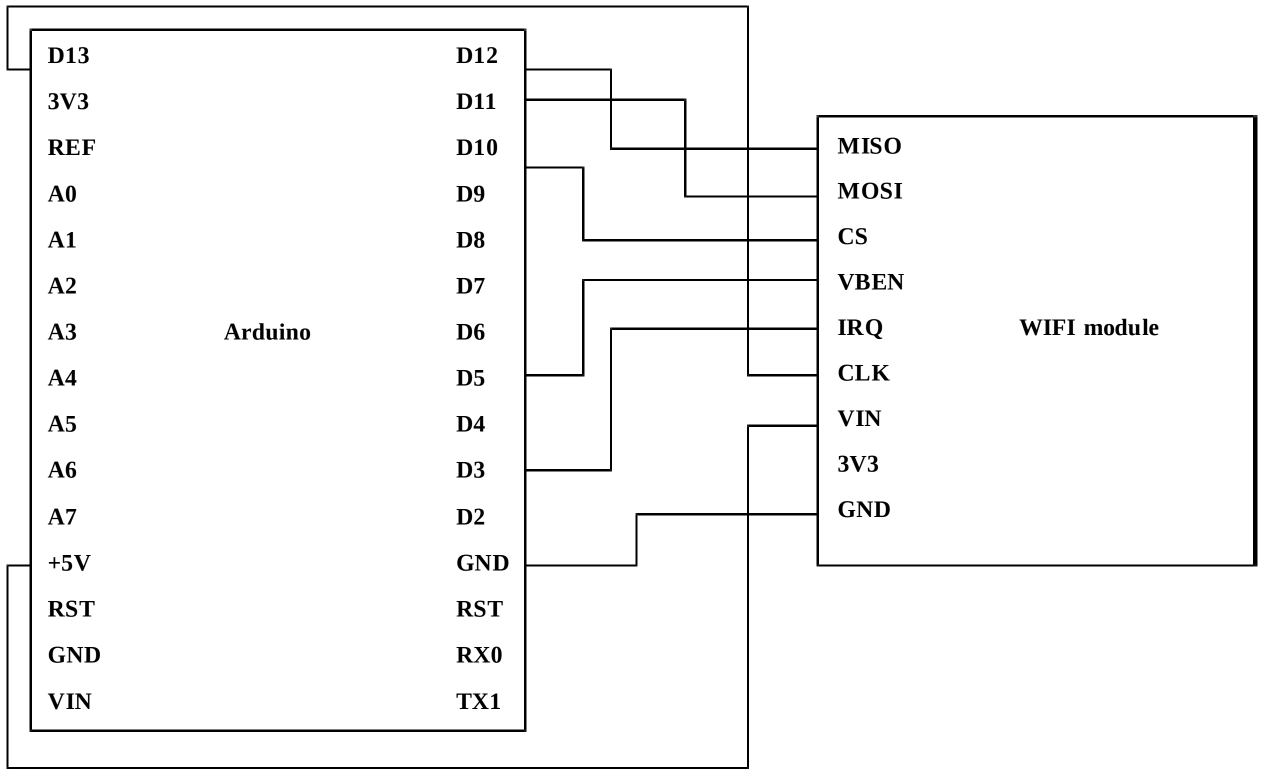


Figure 2: WIFI module connect with Arduino board

Arduino connect with predefined library gate is shown in figure 3. The Arduino Digital board consists of Arduino Nano board and the following atomic library gates: 1. And Gate 2. Nand Gate 3. OR Gate 4. NOT Gate 5. XOR Gate 6. RS flip flop using Nand Gate. Functionality of Arduino board in proposed system is process the expression given by android processor from left to right based on priority and parenthesis. Provides the input to the generated digital circuits connected to Arduino digital board. Arduino reads the output from the digital circuit output pins provide this output back to the Android processor.

Proposed Digital circuit design process

Digital circuit design process is executed in arduino based on two methods.

1. Gate based circuit design
2. Library based circuit design

Gate based circuit design

In gate based circuit design, library gates are used to build new digital logic circuit. The newly build digital circuit added to the arduino as a library circuits.

Library based circuit design

In library based circuit design, newly added library circuits are used to build digital logic circuit. If Boolean expression contains function of any library circuits, then Arduino uses the library circuit. At $time_1$, $expression_1$ executed and the digital circuit is designed by arduino. The designed digital circuit added as library circuit to arduino board. At $time_n$, $expression_2$ needed to be evaluated to build logic circuit by arduino.

$$Expression_1 = Library\ circuit_1$$

$$Expression_2 = Expression_1 U\ Expression_{sub(2)}$$

$$= Library\ circuit_1\ U\ Expression_{sub(2)}$$

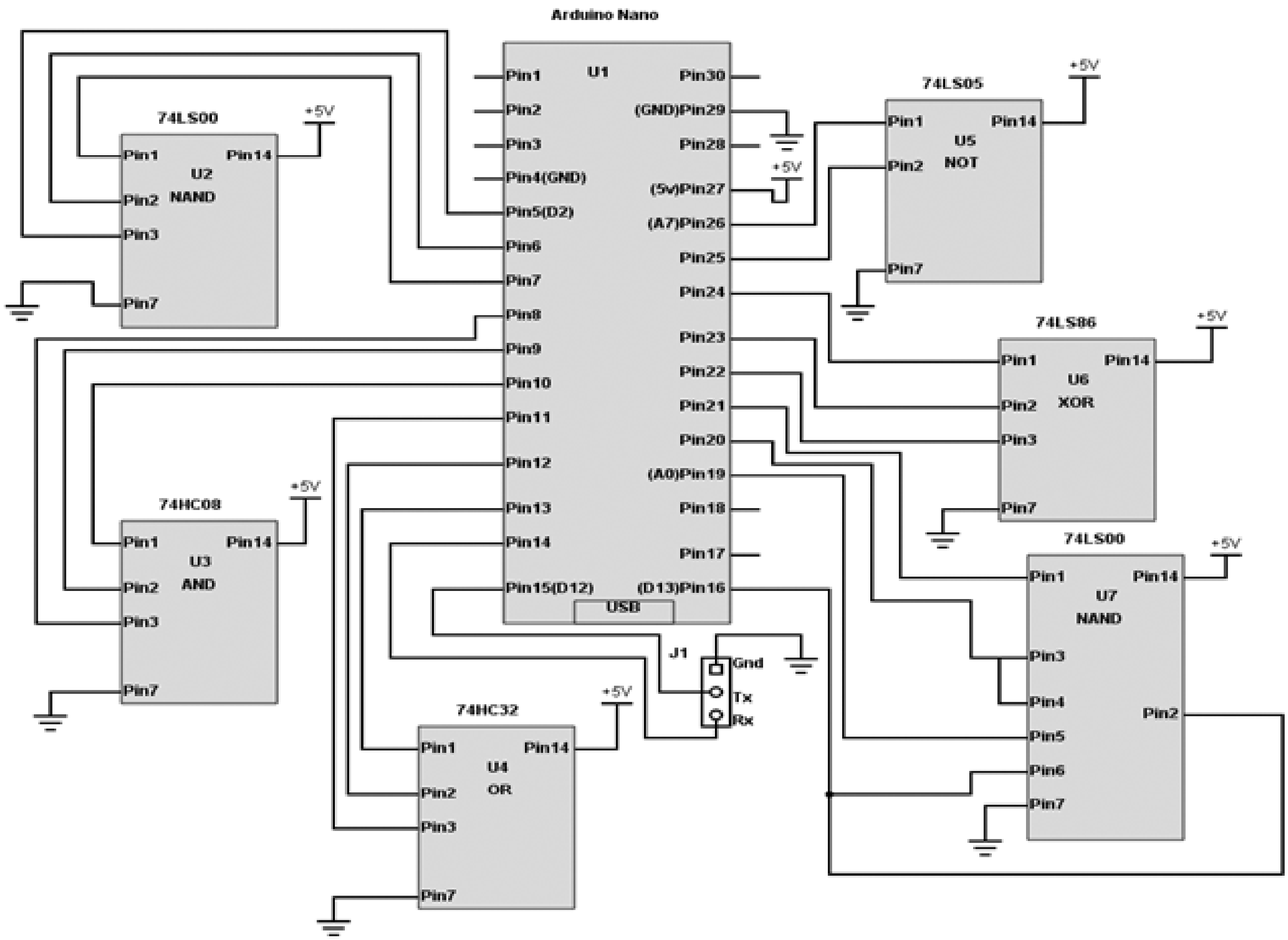


Figure 3:

Pseudo code

Initialize communication in both Arduino and Android processor
Android processor

Android send access request to arduino
if (arduino = idle)
arduino send expression request
else
wait for idle state

Arduino board
loop: execution
expression divided into sub-blocks
select library circuits/library gates
build circuit
end loop
loop1: verification
give i/p to build circuit
o/p send to the Android processor
end loop1
While (expression received)
Check for syntax error
if(error)
send error in expression
else
execution loop
verification loop1
end if

Arduino build digital logic circuit for Expression$_{sub(2)}$ and use the library circuit$_1$ for complete evaluation. The resulted digital logic circuit added as library circuit$_2$.

Expression$_2$ = Library circuit$_2$

Scalability

By library based circuit design, the number of logic gates requirement is reduced. Complex digital circuits are difficult to build. Scalability convert complex circuits as sub blocks and thus to evolve complex circuits. The evaluation is easier by scalability. This implies the proposed system of build digital circuits are scalable, and therefore, the effort required to build circuits of n blocks is invariant to the size of the blocks.

Input Frame Format for Expression building and Verification

Syntax: s <ExpType> <Boolean Expression> e

's' defines the starting of the frame. 'ExpType' defines number to identify the predefined library module. It is always fixed as '0'. 'Boolean Expression' defines Boolean expression substituted by its values Ex: (1+0) *(0 *!1). 'e' defines the end of frame.

Output Frame Format for Expression building and Verification

Syntax: s <ExpType> <Status> <out> e

's' defines the starting of frame. 'ExpType' defines to number identify the predefined library module. It is always fixed as '0'. 'Status' defines status of expression building it is either '0' for Success or '1' in case of syntax error in Boolean expression. 'Output' defines output of the expression, it is either '0' or '1'. 'e' defines end of Frame.

The code flow of Arduino for Expression building and verification is shown in figure 4.

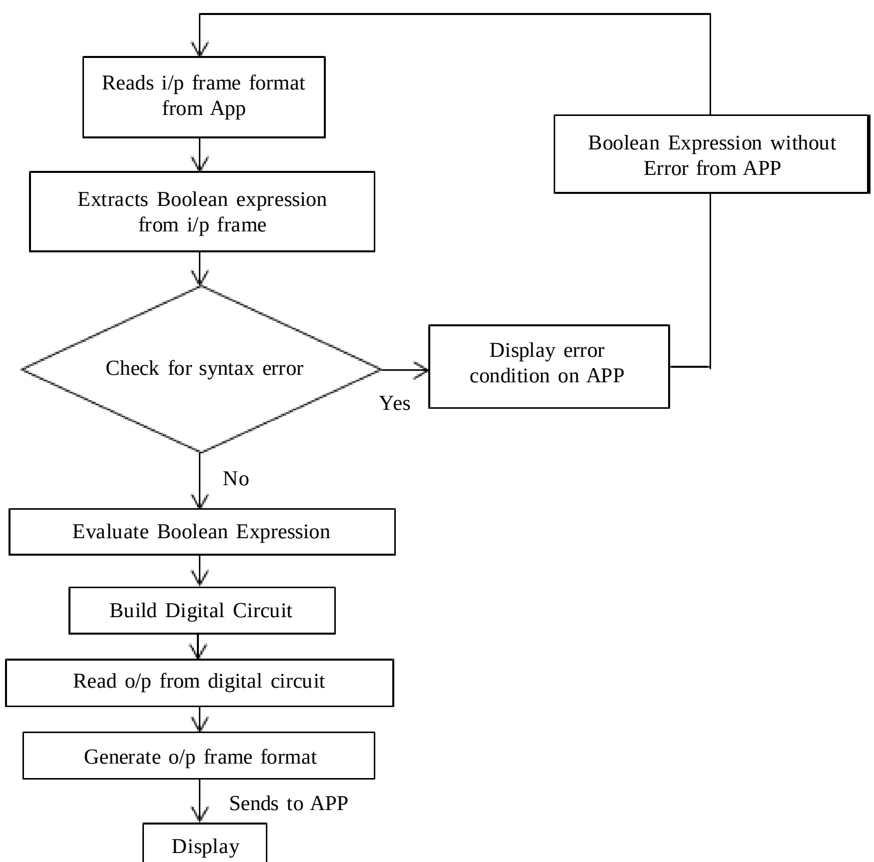


Figure 4:

The Code Flow of Android processor for Expression building and Verification is shown in figure 5.

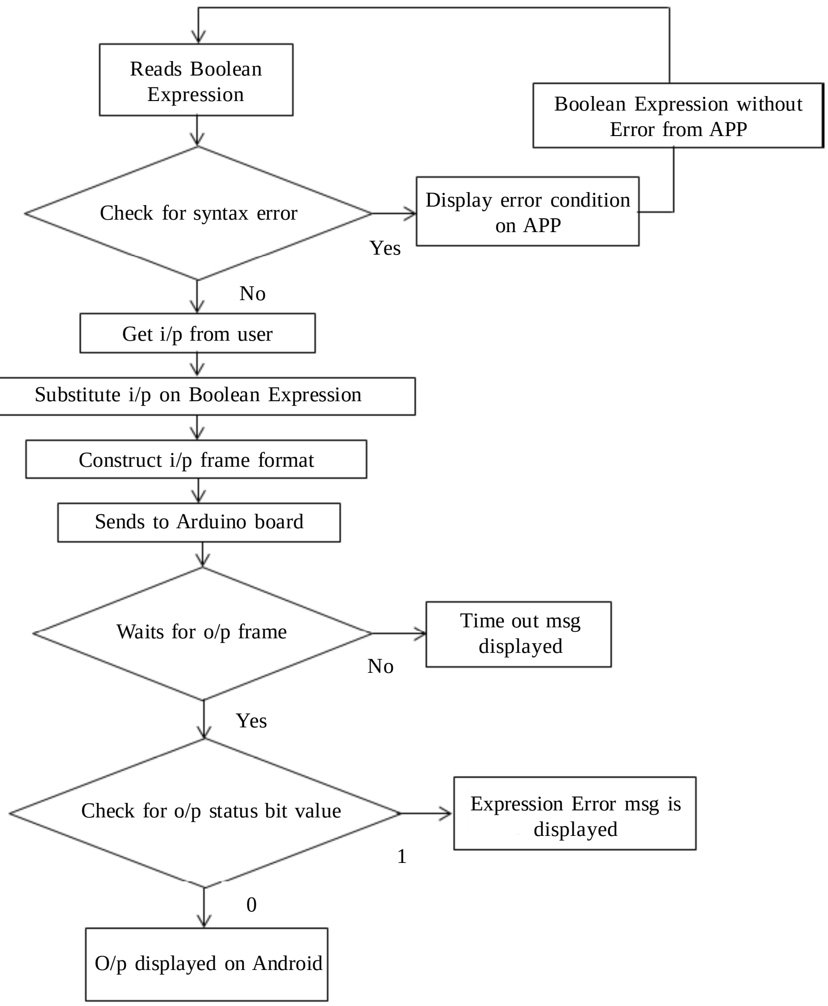


Figure 5:

4.5. Boolean expression

A Boolean expression is the relationship between Boolean variables which is used to design digital circuits using Logic Gates. Every logic circuit is described using the Boolean expression. Logical operators such as ~, * and + is used for logical gates respectively NOT, AND and OR gates. The Boolean expression should be formed only by logical operators and input variable names (A, B, C and D etc.). Boolean operations evaluated by precedence order of NOT, AND OR. Expressions with brackets () are always evaluated ûrst, before the precedence order.

Boolean expression error check

i. Boolean expression with constants (0 and 1) consider as syntax error.

ii. Consider Boolean expression (A+C) +AB. Android processor returns syntax error for this Boolean expression. Because of ignoring AND (*) symbol considered as syntax error. The correct Boolean Expression is (A + C) + A*B.

4.6. Evaluating Boolean expressions

In evaluation of Boolean expressions, initially, the number of inputs to the circuits and library circuit required is calculated. The digital circuit is built by the Arduino using parenthesis and order of precedence.

Consider Y = A*D + !(A + !(B + C)), number of inputs are four and two NOT gate, three 2-input OR gate and one 2-input AND gate are the required library circuits. Input values for the Boolean expression is given by the APP and applied on the Boolean expression.

1. Parenthesis has highest priority in (B + C). OR gate executes addition operation between B and C and the result is represented as X.
2. Parenthesis has second highest priority in (A + !X). In that NOT operation has highest priority than OR. So, NOT gate executes the inversion operation of X and the result is considered as Y.
3. OR gates execute addition operation between A and Y and then result inverted by NOT gate.
4. AND operation has the next highest priority so AND operation between A and D is executed. Obtaining result is added with step 3 result.

Implemented hardware setup

The connection setup includes

1. Connect Transmitter of WIFI module to Receiver of Arduino digital board
2. Connect receiver of WIFI module to transmitter of Arduino digital board
3. Connect ground of WIFI module to ground of Arduino digital board
4. Connect Power Adaptor of WIFI Module to Power Socket

5. SIMULATION SETUP AND RESULTS

5.1. Expression Builder and Verification

In android processor, if the start builder process enabled by the user, then request sent to Arduino for act as receiver by webserver. The Boolean expression and corresponding inputs is given to the android

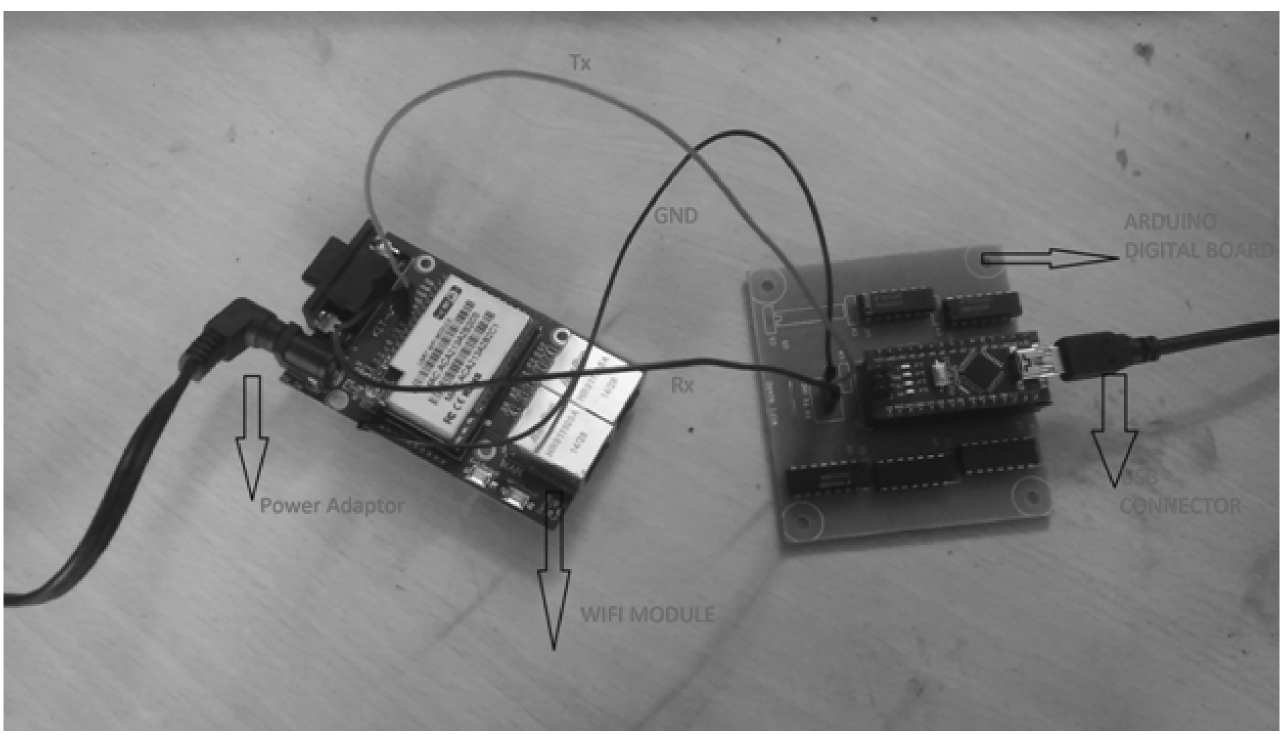


Figure 6:

processor. The input frame format constructed sends to Arduino digital board through WIFI. The corresponding circuit builds by Arduino board. The output frame is received from arduino and sent to the android process through WIFI.

5.2. Implemented results

In android processor, start builder is selected. The Boolean expression (A*B)+(C#D!) is entered in android processor. The arduino interconnects the expression and implements logically the circuit shown in figure 7. The inputs (A, B, C and D) for Boolean expression are given respectively as 0, 1, 1 and 0. If the evaluate button is clicked the output is displayed on android processor. These are explained in figure 8.

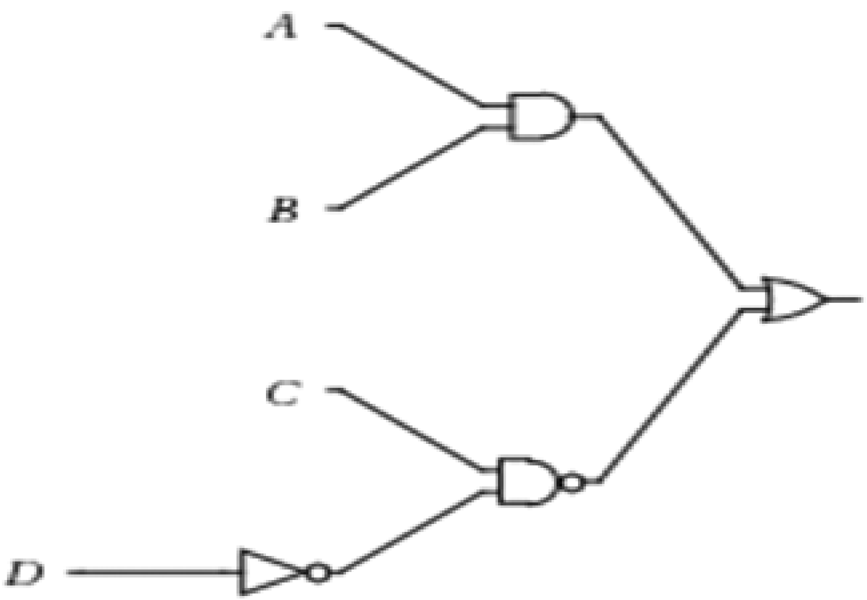
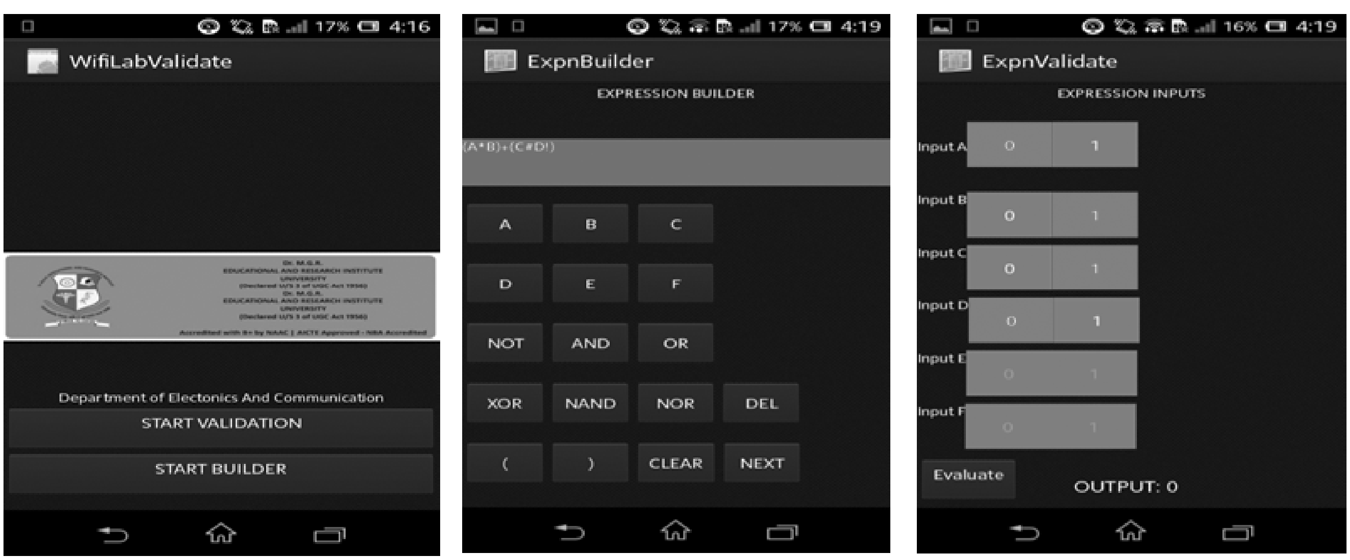


Figure 7:



(a) **(b)** **(c)**

Figure 8:

6. CONCLUSION

In this paper, building digital circuits remotely and verifying the designed digital system is implemented. The concept of build-and-verify together by one system improves the designing quality. The hardware implementation gives real behavior verification of build digital circuit through WIFI. Automation of design and verification of digital circuits is achieved. This paper provides a novel solution to digital logic circuit verification by extracting output from its build circuit at the inputs from remote area. If the specification of the digital logic circuit is known, the extracted output will be compared with that specification. Otherwise, the extracted output provides the function of implemented by the circuit. Damage to library circuits and/or instruments is avoided by Arduino which defines maximum source voltages and all circuit loops which are possible to create without overloading any library circuits. Future direction of study shall focus on extracting optimal expression and accordingly optimal circuit verification can aldo be achieved.

REFERENCES

[1] Javier Hormigo, Gabriel Caffarena, Juan P. Oliver And Eduardo Boemo, “Self-Reconfigurable Constant Multiplier for FPGA”, ACM Transactions on Reconfigurable Technology and Systems, Vol. 6, No. 3, Article 14, Publication date: October 2013.

[2] Xijiang Lin, Sudhakar M. Reddy and Janusz Rajski, “Using Boolean Tests to Improve Detection of Transistor Stuck-open Faults in CMOS Digital Logic Circuits”, 2015 28th International Conference on VLSI Design and 2015 14th International Conference on Embedded Systems.

[3] Frederick F. Sellers, Jr, M. Y. Hsiao, and L. W. Bearnson, “Analyzing Errors with the Boolean Difference”, IEEE TRANSACTIONS ON COMPUTERS, JULY 1968.

[4] R. Glenn Wright, Larry V. Kirkland, Marek Zgol, David Adebimpe, Ernest Keenan, and Robert Mulligan, “Novel Approach to Circuit Board Testing”, NSTI-Nanotech 2004, www.nsti.org, ISBN 0-9728422-9-2 Vol. 3, 2004.

[5] The Complexity of Boolean Functions by Ingo Wegener Johann Wolfgang Goethe-Universit¨atCopyright c_ 1987 by John Wiley & Sons Ltd, and B. G. Teubner, Stuttgart.

[6] Maciej Ciesielski, Cunxi Yu, Walter Brown, Duo Liu, “Verification of Gate-level Arithmetic Circuits by Function Extraction”, *DAC ’15*, June 07-11, 2015, San Francisco, CA, USA.

[7] Digital design principles and practices third edition by JOHN. Wakerly. Copyright © 1999 by John F. Wakerly.

[8] Calvin Jongsma, “Discrete Mathematics: Chapter 7, Cosets, Lattices,& Boolean Algebra”, 1-2016.

[9] https://www.arduino.cc/en/Main/ArduinoBoardYun

[10] http://developer.android.com/tools/workflow/index.html

[11] Jung-Tai Tsai Chun-Yao Wang Kuang-Jung Chang “Reachability Analysis of Sequential Circuits”, 978-1-4244-5271-2/10/$26.00 ©2010 IEEE

[12] R. Bharathikannan and Dr. S. Ravi, “APP ENABLED REMOTE TESTING AND VERIFICATION OF DIGITAL CIRCUIT”, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 11, Number 5 (2016) pp. 3299-3303 © Research India Publications. http://www.ripublication.com