© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

Journal homepage: http://penerbit.uthm.edu.my/ojs/index.php/ijie
ISSN : 2229-838X e-ISSN : 2600-7916

The International Journal of Integrated Engineering

Booth Algorithm with Implementation of UART Module using FPGA

Abd Kadir Mahamad[1,2*], Azmi Sidek[3], Sharifah Saon[1,2], Kenneth Wong Fatt Kong[4], Iqbal A. Khan[5]

[1]Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Johor, MALAYSIA

[2]Internet of Things Focus Group, Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Johor, MALAYSIA

[3]Centre for Diploma Studies,
Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Johor, MALAYSIA

[4]Iconix Consulting Sdn Bhd, No. A-3-2,
Northpoint Offices,Mid Valley City,No. 1, Medan Syed Putra Utara,59200 Kuala Lumpur, MALAYSIA

[5]Department of Electrical Engineering,
Umm Al-Qura University, Makkah, SAUDI ARABIA

*Corresponding Author

DOI: https://doi.org/10.30880/ijie.2020.12.02.018
Received 29 December 2019; Accepted 27 January 2020; Available online 28 February 2020

Abstract: FPGA gives high level of flexibility to the user to rapidly construct and test any hardware. It has a lot of gates which are used depending upon the hardware to be implemented. These project aims at designing Booth multiplier using VHDL for signed bit multiplication in FPGA for high speed operations, developed and implemented of UART module required to enable two-way communication between the DE-2 board and computer. It is also designed GUI interface using MATLAB for sending data and enable the output of the process result to be displayed. The Booth multiplier was implemented using the algorithm in both signed and unsigned number and the input and output of the multiplication was successfully achieved and confirmed through simulation. The GUI was implemented and tested, which UART module also performed well for transmitting and receiving of 8-bit width data. In general, the objective of this project was successfully achieved, which, the result of the component part were able to be tested.

Keywords: FPGA, Booth multiplier, UART, VHDL, GUI

1. Introduction

VHDL is related to the hardware-based programming language usually implemented in high speed integrated circuit such CPLD, FPGA and even ASIC. The implementation in FPGA is closely related to the type of platform used and its processor, which contribute with speed of processing, storage capabilities, flexibility and other parameters (Gu,2008).

Multiplication and division are arithmetic operations that been explored by many researchers several decades ago. Earlier basic multiplication operations were perform using ALU's adders. Since then, methods to implement

*Corresponding author:kadir@uthm.edu.my

multiplication operations become more advanced and austere (Akbar *et al*., 2018) -(Bhore *et al*., 2017), especially in digital computing and digital electronics.

Multiplication operation mainly have three (3) stages. The first stage is generating the partial products, which generated through an array of AND gates. For second stage is reducing the partial products by using partial product reduction schemes. Finally, the product is obtained by adding a partial product (Kuang *et al*., 2009) which can produce in signed and unsigned numbers.

The implementation of multiplier in FPGA as an individual function has been tested numerously and proven that the multiplication of two integers is valid even with signed numbers. It is tested using manual input (platform devices) or simulated using various bit setting without considering interconnection to external devices. Therefore, it crucial to examine the proof of concept of multiplication process when connect to external hardware.

Interconnection of DE-2 board to external devices such as computer, which acts as data input, demanding additional module to be built. This designed module is to enable data to be received by DE-2 board which later process and transmitting the result to the external devices. This module integrates baud rate generator, UART receiver and UART transmitter in order to create UART module that enable data communication between external devices with DE-2 board (Pong, 2008). The project is further enhanced by developing GUI MATLAB-based interface that enable two consecutive input data to be sent to DE-2 board, processed input data and then display the result.

2. Booth Multiplier

Nowadays, digital system has been used in many applications to improve daily life because of it benefits compared with analog system. Due to development of digital system, many complex devices of microprocessor, microcontroller or microchip had been designed. So, the high speed and efficient multiplier system is important for new device of microprocessor. Basically, multiplication process requires large area, long latency and consume considerable power. Therefore low-power multiplier design has been considering during VLSI system design. Fast multipliers are great importance in digital signal processing as well as in the general-purpose processors today. The basic multiplication principle is twofold i.e., evaluation of partial products and accumulation of the shifted partial products. Therefore, the application of Booth Multiplier is one of the most important parts which can affect the overall of devices performance and circuits.

The Booth algorithm (Tariq et al., 2016)- (Chaitanya *et al*., 2019) performs the operation by generates a 2n-bit product and make both positive and negative 2's complement n-bit operands uniformly. As a basis operation of this algorithm, consider a multiplication operation which multiplier is positive and has a single block of 1s. To derive the product, it could add the total number of a single block of 1s appropriately shifted versions of the multiplicand. However, the operation can reduce the number of required operations by regarding this multiplier as the difference between two numbers.

In the Booth algorithm, −1 time the shifted multiplicand is selected when operation moving from 0 to 1, and +1 times the shifted multiplicand is selected when operation moving from 1 to 0, as the multiplier is scanned from right to left. The Booth algorithm (Carl *et al*., 2012)-(William, 2010) clearly covers to any number of blocks of 1s in a multiplier, including the situation where single 1 is considered as a block.

The Booth algorithm has two special features. First, this algorithm handles both positive and negative multipliers uniformly. Second, it gains efficiency in the number of additions required when the multiplier has a several large blocks of 1s.

3. Methodology

In this project, VHDL is the main programming language used to develop Booth algorithm and UART module. The implementation of Booth algorithm in VHDL is designed based on the flow chart illustrated in Fig. 1 which specify the input to be 8-bit width regardless the sign notation. There are two (2) operations that should be known before attempting Booth algorithm (Wang *et al*., 2008; Hussin *et al*., 2008);

- *Right-Shift Circulant (RSC)*: RSC is simply shifting the bit, in a binary string, to the right, 1-bit position and take the last bit in the string and append it to the beginning of the string, and
- *Right-Shift Arithmetic (RSA)*: RSA is addition of two binary number together and shift the result to the right, 1-bit position.

Prior to implementing Booth multiplication algorithm, these acronym need to be identified; the product Register or Accumulator (A), multiplicand (M), multiplier (Q) and variable x and y to count for the bit length M and Q. The first step is to setup the following step:

- convert the two operands into binary,
- determine which operand has the least transitions between bits, and
- set Q, equal to that operand which has the least transition bit and M, equal to the other operand.

Followed by setting up four columns as shown in Fig.2, with the initial value is set to 0. The functionality of each columns are as follows;

- *1st column is* ***A***; this column holds the results from each step in the algorithm,

- *2nd column is **M***; this column holds the overflow from A when right-shifting,
- *3rd column is **Q***; this holds Q operand. This will show each RSC step, and
- *4th column is **Q-1***; this holds the least significant bit from *Q* before RSC.

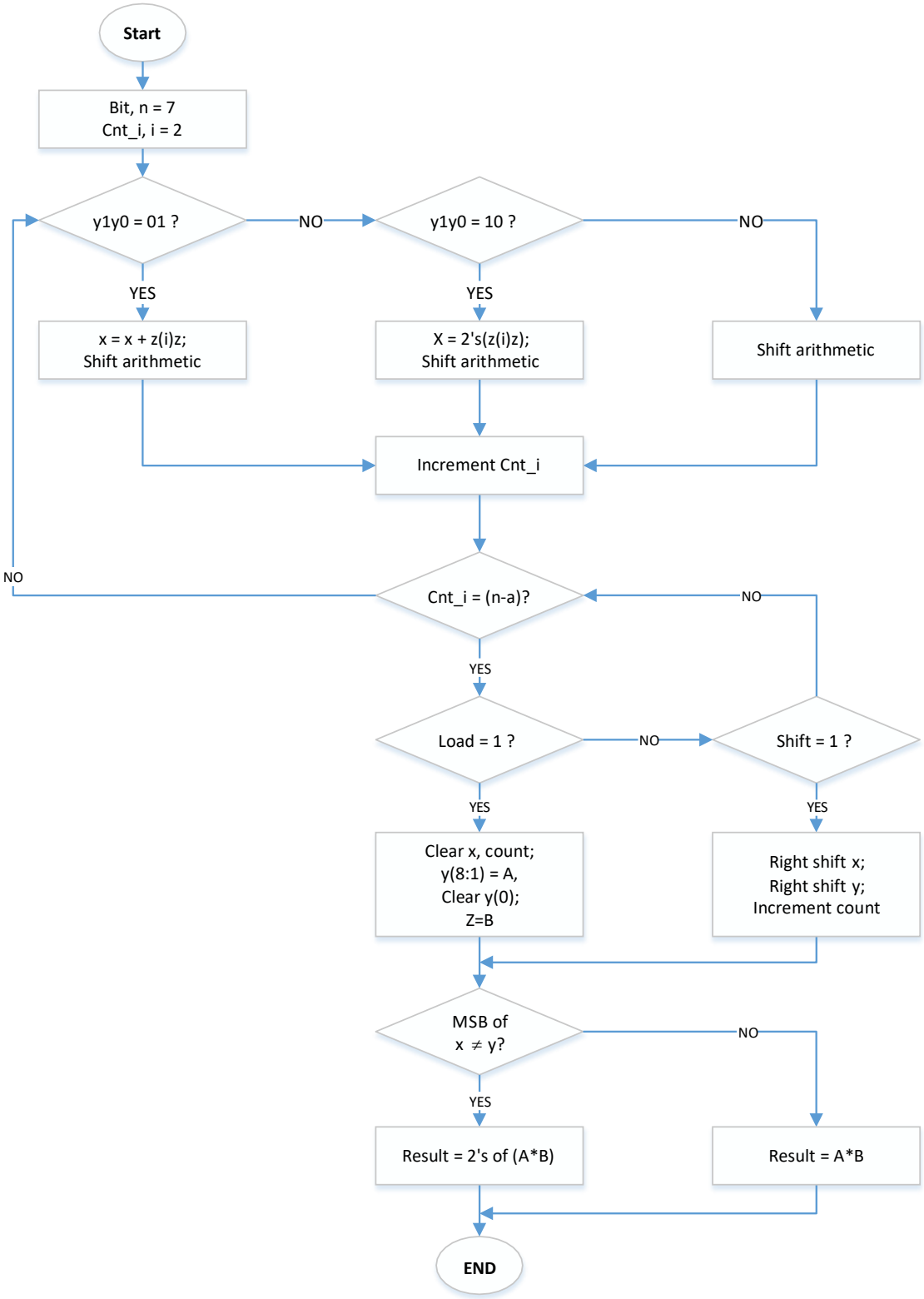

Fig. 1 - Booth Multiplier algorithm

A	*M*	*Q*	*Q-1*	Operation

Fig. 2 - Columns setup

The next step is to analyze the least significant bit of *Q* and the bit in *Q-1*. Refer Table 1 for the consequence action:

- if the string is 01, add *M* to the value in *A* and right-shift the result,
- if the string is 10, subtract *M* (Add 2's compliment of *M*) from the value in A and right-shift the result,
- if the string is 00, take no action, and
- if the string is 11, right-shift the value in *A*, to 1-bit position.

Table 1 - Operation summary

Q	*Q-1*	Operation
0	0	Shift Only
1	1	Shift Only
1	0	Subtract (A) and shift
0	1	Add (A) and shift

The final step is to implement RSC for *Q*. Repeat the previous steps until the *Q* has been RSC to its original position. Two 8-bit shift register is required for Booth multiplier operation to perform serial data communication in order to synchronizes the ASCII data encryption before feeding it the multiplier. The 8-bit shift register is design by cascaded D flip-flop to shift the data 1-bit at every clock cycle. The decrypted ASCII code is then feed to parallel load with modulo-8 which trigger control of data out to the parallel out. The integration of shift register, modulo-8 counter and parallel load to Booth multiplier circuit, completes the data processing from serial input and ready to be sent back to PC by parallel output.

GUI main function is as user interface for data input, and acquire process data from DE-2, thus displaying the result. GUI is developed by using a MATLAB & SIMULINK R2009b version. The layout of GUI is creating with the two text boxes for user to insert the numbers to be multiplied in. The push-button of SEND is created to enable the transmission of the arithmetic operation to be sent to FPGA device via RS232 serial communication port. Then executed the multiplication process and the result is sent back to MATLAB via same medium. The result of multiplication had been programmed to be displayed in RESULT text-box in MATLAB GUI. To initiate the display result process, the COMPUTE push-button is created. The layout of the developed GUI is shown as in Fig. 3.

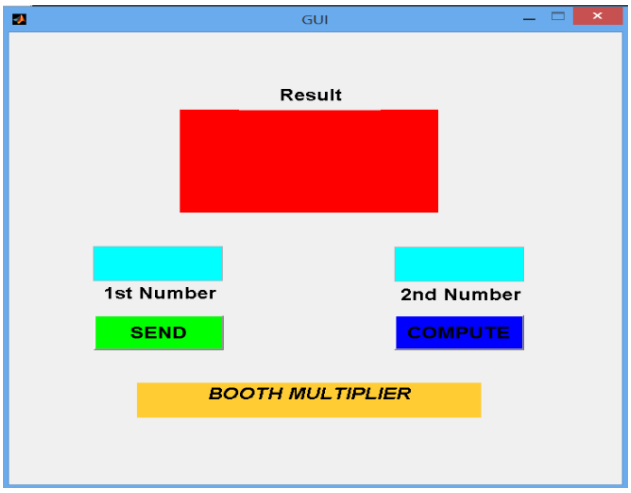

Fig. 3 - Layout of MATLAB GUI

Virtual serial port to initiate the connection between MATLAB and virtual terminal in Proteus Professional v.8 third party software are for verification purpose only. The MATLAB script code is modified to test the connection and transmit data from MATLAB GUI (Input Number 1 & Input Number 2) to serial port RS232. The numbers should be appeared in virtual terminal Proteus if the successful connection is established. Same case, for receiving data from serial communication port, it can be set any number from virtual terminal and the number should be appeared in the RESULT text-box GUI MATLAB. Virtual serial port emulator is downloaded from open-source but with the limited time usage. Fig. 4 shows the illustration on how the connection verification is done.

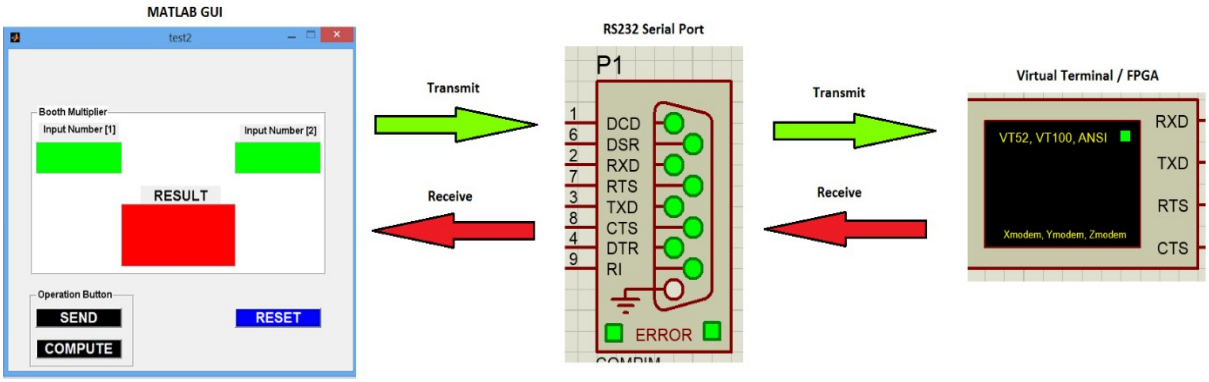

Fig. 4 - The graphical view of connection verification between MATLAB GUI and serial communication port RS232

Universal asynchronous receiver and transmitter (UART) is a system that transfer parallel data through serial cable. In this case, data is transmitted from PC through RS232 serial cable and process for arithmetic operation in DE-2 board. UART is responsible of converting received data into data configuration recognize by the processor. The complete system of UART consists of baud generator to synchronize the data received and process, UART receiver to process data received from PC, FIFO for buffering received/transmit data for next process and UART transmitter to process transfer to external devices (Bhor *et al*., 2014) -(Govindrao, 2011).

The block diagram of UART complete system is as illustrated in Fig. 5. UART transmitter is a shift register that loads data in parallel and then shifts it out, bit by bit at a specific baud rate. UART receiver, on the other hand, shifts in data, bit by bit and then reassembles the data. The serial line is '1' when it is idle. The transmission starts with a *start bit,* which is '0', followed by *data bits* (in this case, 8-bit width) and an optional *parity bit,* and ends with *stop bits,* which are '1'.

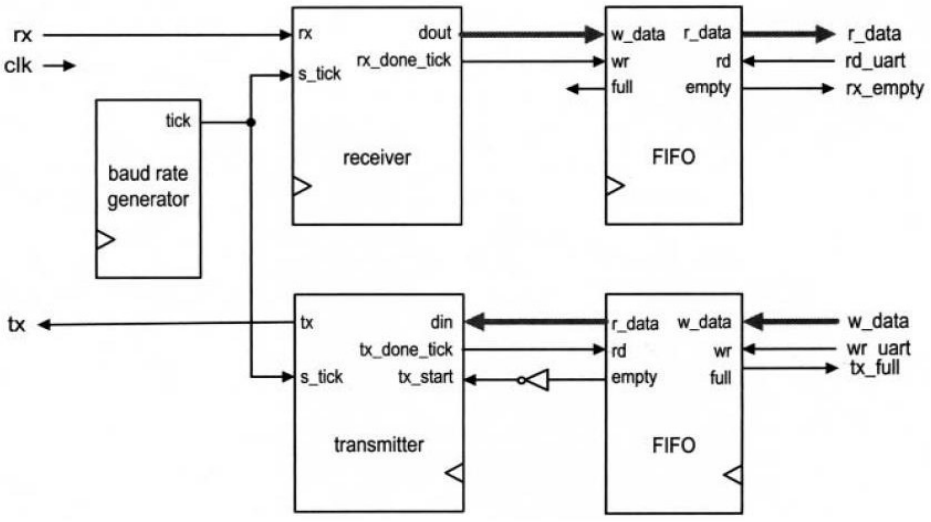

Fig. 5 - Complete system of UART

Fig. 6 shows Booth Multiplier in RTL representation. The operation and function of this multiplier can be referred as in Fig. 1. Meanwhile, Fig. 7 shows the complete UART system configuration in RTL representation which consists of transmitter and receiver. This RTL representation is designed as operation which explained as in Fig. 5.

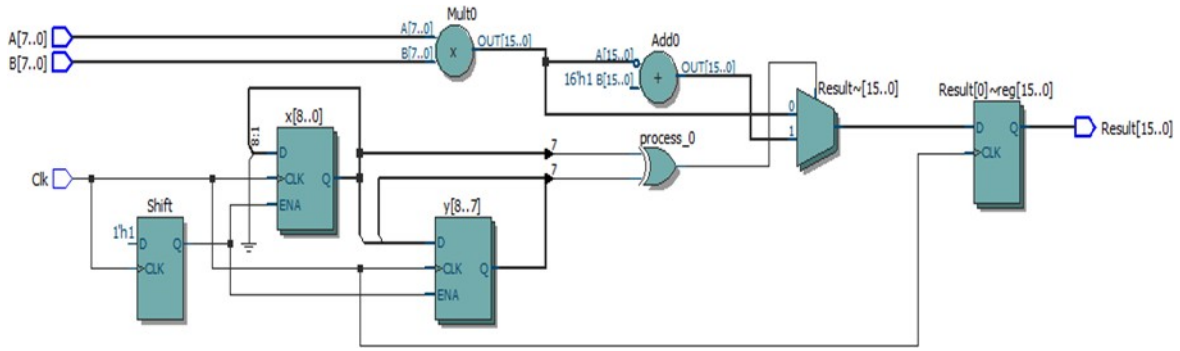

Fig. 6 - Booth multiplier RTL representation

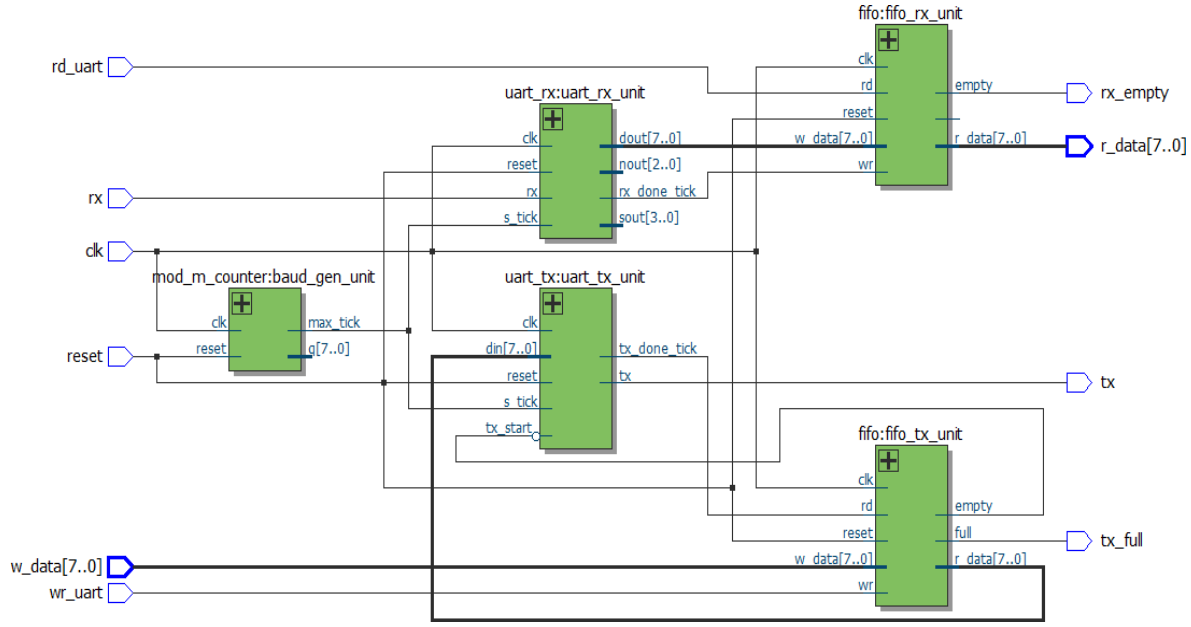

Fig. 7 - Complete UART system configuration

4. Design Analysis

The simulation of Booth multiplier is carried out using signed decimal, where the multiplication is carried out upon the detection of clock cycle rising edge. The simulation output result is as depicted as in Fig. 8. When A = –16 and B = – 15 is multiplied, the result displayed is 240, similar operation is also carried out using A = 17 and B = 57, which resulted the required value of 969 using Booth algorithm. It shows that the algorithm implemented is capable of multiplying a signed value of 8-bit input.

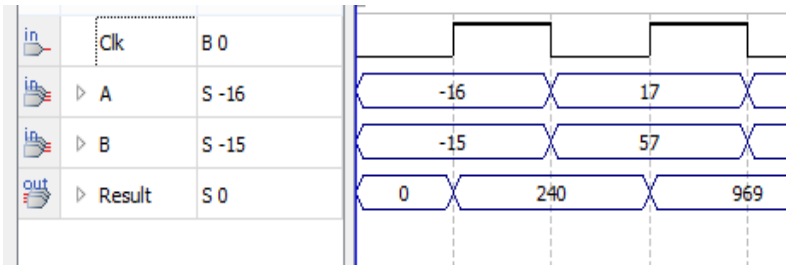

Fig. 8 - Booth multiplication simulation and operation

MATLAB GUI is tested by embedding the script code for multiplication process to test that text-boxes and push-button are operated normally. It has been set, the two numbers in Input Number 1 and Input Number 2 text-boxes and push the push-button SEND to see the outcomes. The result of multiplication is displayed in RESULT text-box. Fig. 9 show that MATLAB GUI operates successfully.

In order to establish the connection with serial communication port RS232, the virtual serial ports must be paired. COM2 is serial port for virtual terminal which is implemented by Proteus. The MATLAB is connected to RS232 via COM1. It can use any available virtual port as long as it does not connect to other application. The MATLAB script code had been embedded to initiate the connection. The result for transmitting the data is shown in Fig. 10. The virtual terminal displayed the Input Number 1 and Input Number 2 which are '67' and '34' with ASCII code of 36 37 33 34 respectively in serial. This prove that MATLAB GUI is successful transmitting the data via RS232.

To verify the data is received to the MATLAB GUI via RS232, the virtual terminal is need to be inserted by any number. Then, the number should be appeared in RESULT text-box in MATLAB GUI. For the verification process, number '1234' was inserted into virtual terminal and the same number is appeared in RESULT text-box GUI MATLAB. Fig. 11 shows the result for receiving process.

Fig. 10 - MATLAB GUI transmit data to virtual terminal via RS232

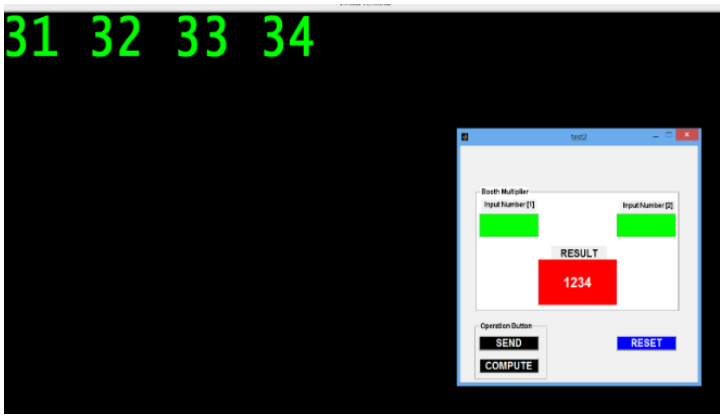

Fig. 11 - Testing result from virtual terminal to MATLAB GUI

5. Conclusion

Due to restriction in the simulation, some of circuits is simulated out of norm protocol to ensure the design circuit works accordingly. The project is successful if it is measured as individual component, but as a whole working circuit, it needs major modification as it need to synchronize UART data protocol with already designed Booth multiplier. Booth multiplier itself works well using sign or unsigned data, which resulted a 16-bit output. The UART receiver and transmitter perform well to receive and transmit 8-bit width data. The UART receiver tested in this project cater only for a single 8-bit width data whereas in this project require two 8-bit width data, on the other hand UART transmitter in this project require 16-bit data transmission instead of only 8-bit width data. The overall system needs to be modified by rearranging the configuration of UART interconnection with Booth multiplier and taking into consideration multiple data input and processing technique.

Acknowledgement

The financial support received from Vot No: U855 and U867, Office for Research, Innovation, Commercialization and Consultancy Management (ORICC), Universiti Tun Hussein Onn Malaysia is gratefully acknowledged.

References

[1] Akbar, A., Adiono, T., Harimurti, S., Putra, T.A.M., (2018). Design of Neuron Net Function using Modified Radix-4 Booth Multiplier with a Flipped Logic Parallel Prefix Adder. International Symposium on Electronics and Smart Devices, 1-6.

[2] Bhor, P.B., Priya, R.A., Malathi, P., (2014). Loading of Soft-Core Processor using Soft Core UART at Run Time. IEEE International Conference on Advances in Engineering & Technology Research (ICAETR - 2014), 1-5.

[3] Bhore, C., Thakare, R.D., (2017). Area efficient and speed performance of pre encoded multiplier by using non redundant radix-4 signed digit encoding. Proceedings of the International Conference on Electronics, Communication and Aerospace Technology, 270-275.

[4] Carl H., Zvonko V., Safwat Z.N.M., (2012). Computer Organization and Embedded Systems. McGraw-Hill Education.

[5] Chaitanya, C.V.S., Sundaresan, C., Venkateswaran, P.R., Keerthana Prasad, (2019). Design of modified booth-based multiplier with carry pre-computation. Indonesian Journal of Electrical Engineering and Computer Science, 13(3), 1048-1055.

[6] El Atre, S.G.M., Alshewimy, M.A.M., (2018). Design and implementation of new delay-efficient/configurable multiplier using FPGA. Proceedings of 12th International Conference on Computer Engineering and Systems, 8-13.

[7] Garg, A., Agrawal, D., Kularia, P., Mehra, A., Rajput, S., (2017). Area efficient modified booth adder based on Sklansky adder. 2nd International Conference for Convergence in Technology, 308-312.

[8] Govindrao, C. H., (2011). Development of PCI Embedded Card useful for Microcontroller Trainer, Diss. Saurashtra University, India.

[9] Gu, Y.F., (2008). FPGA acceleration of molecular dynamics simulations. Diss. Boston University.

[10] Gurgel, Sanches, F., Siqueira, M.D.A., Vallim, M.B.R., (2009). Aspects of Implementation of an Educational Platform for Robot Based FPGA. IX Microelectronic Students Forum SFORUM.

[11] Hussin, R., Md. Shakaff, A.Y., Idris, N., Sauli, Z., Che Ismail, R., Kamarudin, A., (2008). An Efficient Modified Booth Multiplier Architecture. IEEE International Conference on Electronic Design, 1-4.

[12] Kuang, S.R., Wang, J.P., Guo, C.Y., (2009). Modified Booth Multipliers with a Regular Partial Product Array. IEEE Transactions on Circuits and Systems -II: Express Briefs, 56(5), 1549-7747.

[13] Kumar, M.S., Tulasi, S.K., Srinivasulu, N., Krishnam, G.S., Raghuveer, E., Hari Kishore, K., (2018) Improvement of the efficiency of booth multiplier. International Journal of Engineering & Technology, 7 (1.5), 31-36.

[14] Manjusha, K.A., Naresh, B., Arulanath, T.S., (2018). A new architecture of modified booth recorder for add multiply operator using carry save adder. ARPN Journal of Engineering and Applied Sciences, 13(6), 2153-2156.

[15] Miles J.M., Vincent P., (1999). Heuring principles of computer architecture. Prentice Hall.

[16] Pong, P.C., (2008). UART, FPGA prototyping by VHDL examples. New Jersey: Wiley, 163-182.

[17] Prabhu, S., Elakya, V., (2012). Design of modified low power booth multiplier. IEEE on International Conference on Computing, Communication and Applications, 1-6.

[18] Raghuveer, M., (2012). Hardware Implementation of Densely Packed Decimal Encoding-An Optimized Approach Supporting Run-Time User Input. Diss. National Institute of Technology Rourkela,Odisha.

[19] Ravindra P.R., Shanmukha Swamy, M.N., (2012). High Speed Modified Booth Encoder Multiplier for Signed and Unsigned Numbers. IEEE on UKSim 14th International Conference on Computer Modelling and Simulation, 649 – 654.

[20] Rubinfield, L.P., (1975). A Proof of the Modified Booth's Algorithm for Multiplication. IEEE Transactions on Computers, C-24(10), 1014-1015.

[21] Sukowati, A.I., Putra, H.D., Wibowo, E.P., (2016). Usage area and speed performance analysis of booth multiplier on its FPGA implementation. IEEE on International Conference on Informatics and Computing (ICIC), 117–121.

[22] Swetha, T., Srinivas, S., (2019). A novel IEEE-754 floating-point butterfly architecture based on multi operand adders. International Journal of Recent Technology and Engineering, 7(5), 55-60.

[23] Tariq, A.S., Amin, R., Mondal, M.N.I., Hossain, M.A., (2016). Faster implementation of Booth's algorithm using FPGA. 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE), 1¬¬¬-4.

[24] Wang, L.R., Jou S.J., Lee, C.L., (2008). A Well-Structured Modified Booth Multiplier Design. IEEE International Symposium on VLSI Design, Automation and Test (VLSI-DAT), 85-88.

[25] William S., (2010). Computer Organization and Architecture Designing for Performance Eighth Edition. Prentice Hall.