IMPLEMENTATION OF 64-BIT PARALLEL SUBTRACTOR USING XILINX IP CORES FOR FAST COMPUTING

B.Khaleelu Rehman[1], G.Vallathan[2], Vetriveeran Rajamani[3], Mohammad Salauddin[4], Mudasar Basha[5]

[1]Associate Professor Dept of ECE, Nalla Malla Reddy Engg College, Hyderabad.afridi.1156@gmail.com

[2]Associate Professor, Dept. of ECE, Geethanjali College of Engineering & Technology, Hyd
[3]Associate Professor Grade-1, School of Electronics Engineering (SENSE), Vellore Institute of Technology, Vellore.[1]
[4]Associate Professor Dept of ECE, JBIET,Hyderabad
[5]Assistant Professor, Dept. of ECE, B.V.R.I.T Narsapur, Hyderabad

Abstract. The main objective of the paper is the part of Arithmetic Logical circuits (ALU) i.e Subtractor circuit.64 bit parallel subtractor circuit by using core generator Xilinx IP core and conventional approach is simulated and synthesized with the most advanced Xilinx FPGAs.The proposed system uses less delay as compared to the existing system.RTL, placement and routing, Hardware device utilization parameters, simulation results and power reports are verified by using VHDL programming language. Xilinx I-sim simulator is used for functional verification and timing verification.. X-power analyzer is used to calculate the power.

Keywords. FPGA, VHDL, Xilinx IP.

1.Introduction*:* A N-bit parallel subtractor, subtract the 2 inputs A (Minuend) and B (Subtrahend) as shown in figure 1.FA_1 is the first full adder. Similarly, FA_2 is the second full adder and FA_n is the nth full adder block.Ci_1 is the initialized carry input of the first full adder .$S_{1,}$ $S_{2.........,}$ Sn-$_{1,}$ $S_{n,}$ is the sum of the respective full adders. $CO_{1,}$ $CO_{2.............}CO_{n-1,}$ $CO_{n,}$ is the carry of the respective full adders. A_1 is the LSB of Minuend and $B_1$is the complement of Subtrahend. The two’s complement of a number can be done by converting the binary format into its equivalent 1’s complement. Here 1’s complement is to negate the binary number. Here, by appending ‘1’ to the LSB bit of 1’s complement, 2’s complement notation can be attained [1]. By using basic gates, the 1’s complement of ‘B’ can be attained through the NOT logic gate & ‘1’ is added throughout the carry to get the two’s complement notation of ‘B_1’. Further, this is added to ‘A_1’ to perform the subtraction. The value of C_{i1}=’1’ This procedure will continue till the final full adder like ‘FA_n’ and it utilizes the carry bit ‘C_n’ to include with its input ‘A_n’ as well as 2’s complement of ‘B_n’ to produce the final output bit with final carry bit ‘C_{out}’. From the figure, it can be seen that the first full adder carry FA1 is the input to the second full adder FA2 carry and the FA2 carry is the input to the third full adder FA3 carry [2].To design a 64-bit parallel subtractor the designer requires 64 full adders which are connected in a cascaded manner. For example, the delay of the first full adder is 10ns [3] then to generate the 64th Full adder it requires 10ns×64=640ns to generate the final output

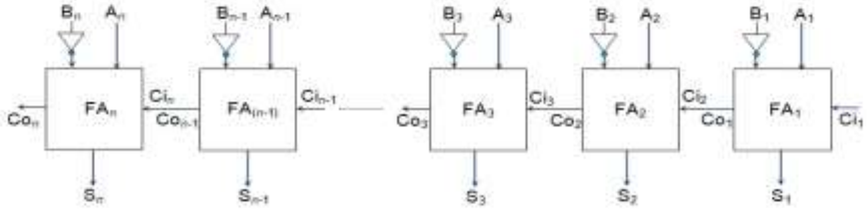
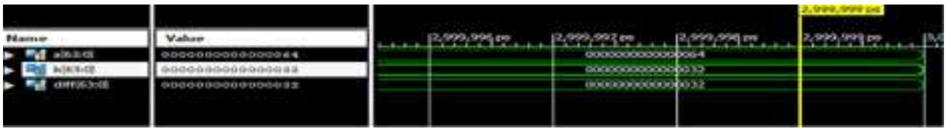


Fig. 1 N-bit parallel subtractor adder block diagram

The I-sim simulated figure as shown in fig 2 depicts One input a [63:0] is the hexadecimal format "0000000000000064" and its base-10 equivalent is 100.similarly the input b[63:0] its hexadecimal is "0000000000000032" and its decimal value is "50".The output diff[63:0] is the hexa equivalent is "0000000000000032" and its base-10 equivalent is 50.The MCPD for the parallel subtractor circuit is 13.146ns through which **1.376ns** for the logic and 11.770ns for the routing the design. Total REAL time to Xst completion requires 44.00secs and total CPU time for finishing time is 44.19secs. Total memory usage of the design 4698524 kilobytes. The table 1.1 shows the hardware device utilized report of the 64-bit parallel subtractor. The number of slices Luts used for the design are 95 out of 17600 available and the bonded Input output blocks are 192 out of 100 available. The synthesis report, RTL view, and simulation waveforms have been targeted with FPGA Zynq XC7Z010 [4] hardware works on 28nm technology and has the ARM processing system [5] and programming logic.

2. Proposed Work: In this digital design era engineers use HDLs for description of any complex logical function. Let us assume that the digital design engineer needs the ALU to design, then one of the components in the ALU design is the subtractor circuit. ALU is used to performs all the arithmetic and logical operations. The operations involved such as subtraction, addition [6] , division and multiplication for signed and unsigned numbers. Logical operations such as logical shifting (left shift, right shift). This paper focuses on the 64 bit parallel subtractor circuit. For designing the parallel subtrator circuit each and every time the engineers have to redesign and code has to be developed at each time they were using .One solution to end up with the above problem is by using Xilinx IP cores[7]. In Xilinx IP cores a piece of code is written for performing a specific task and there by saves the designers time effectively.

2.1 64-BIT PARALLEL SUBTRACTOR: To design a 64-bit parallel subtractor. Below figure 3 shows the Xilinx IP core wizard. The left side shows the FPGA [8] device Zynq XC7Z010 Right side shows the IP(core generator & Architecture wizard). The figure 2.2 shows the Xilinx adder/Subtractor 11.0 IP core.



Name	Value
a[63:0]	0000000000000064
b[63:0]	0000000000000032
diff[63:0]	0000000000000032

Fig 2 64-bit parallel subtractor simulation

.

Synthesis Report summary		
Logic used	Used	Available
No. of Slice Luts	95	176000
No. of Fully used FF-pairs	0	95
No. of bounded IOBs	100	192

Table 1 64 bit parallel subtractor synthesis report

Adder & Subtractor IP core is used for adding and subtracting the 256-bit unsigned and signed numbers. Apart from adder and subtractor it also performs adder and subtractor in the single design with one control input.

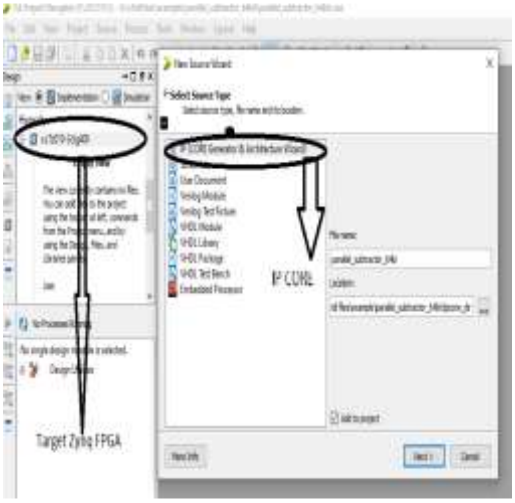


Fig 3 Xilinx ISE IP core

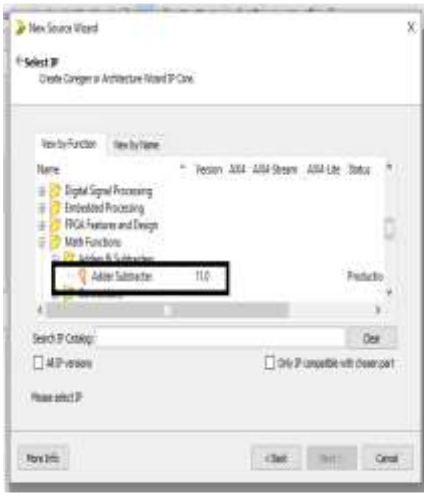
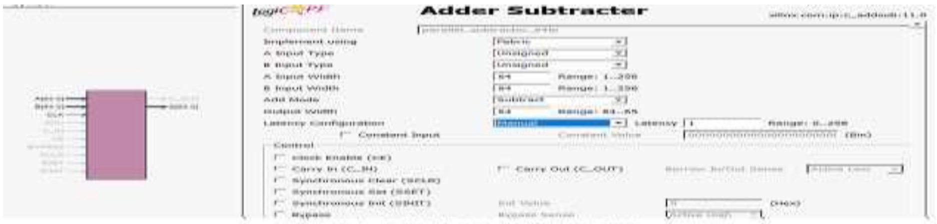


Fig. 4 Xilinx adder/Subtractor IP core.

Figure 2.3 is the 64-bit unsigned parallel subtractor. The figure IP core shown below has two GUI windows. The left GUI part represents the symbolic representation of IP

A[63:0] is one 64-bit input minuend, B[63:0] is another input(Subtrahend), and CLK is



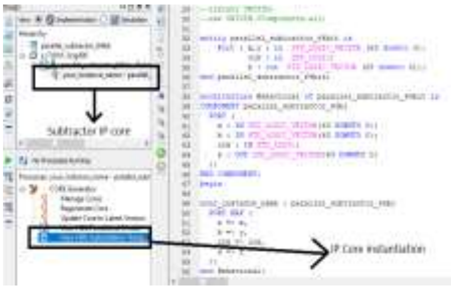
Component Name parallel_subtractor_64b
Implement using Fabric
A Input Type Unsigned
B Input Type Unsigned
A Input Width 64 Range: 1..256
B Input Width 64 Range: 1..256
Add Mode Subtract
Output Width 64 Range: 64..65
Latency Configuration Manual Latency 1 Range: 0..256
□ Constant Input Constant Value 0000000000000000000000 (Bin)
Control
□ Clock Enable (CE)
□ Carry In (C_IN) □ Carry Out (C_OUT) Borrow In/Out Sense Active Low
□ Synchronous Clear (SCLR)
□ Synchronous Set (SSET)
□ Synchronous Init (SINIT) Init Value 0 (Hex)
□ Bypass Bypass Sense Active High

Fig 2.3 64-bit parallel Subtractor

the clock input. The 3 inputs shown are enabled and the remaining ports are disabled the inputs are “ADD” which is used when we select add/subtract module. “C_{in}” is the carry input. And clock enable is representing with “CE”. The outputs are S[63:0], difference output of 64 bits enabled and C_{Out} carry out disabled. The disable pins are optional pins. For any N bit parallel subtractor the carry input should be ‘1’ and the carry output is the output from the previous Full adder

The right side has the component selection. The component selection has 2 options for implementing the design one Fabric and another DSP, through which Fabric uses 64 LUTs and 64 Flip flops.DSP48 is not implemented for 64-bit subtraction. The maximum subtraction is allowed 47 bits with the use of 1 DSP48 processor inside the inbuilt FPGA hardware.

Figure 5 shows the 64-bit parallel subtractor Intellectual property core instantiation the IP core is shown in the highlighted portion in the left side window and below that window, there is the HDL instantiation. The instantiation HDL is with .vho [9] extension an xco file is created and then the instantiation can be done in the VHDL code before “begin” the component declaration and after “begin” the HDL instantiation can be done which is seen in the right side VHDL code. The Register Transfer Level (RTL) diagram of the 64-bit parallel subtractor IP is shown in figure 6. After writing the code and the compilation process completed Xilinx generates a file with extension .ngr for the Register Transfer Level (RTL) schematic. ”clk” is the input signal clock pulse, X(63:0) is one of the input minuends to the subtractor y(63:0) is another input to the adder(subtrahend).z(63:0) is the output difference. Figure 2.3 and 6 are prototype and RTL diagrams respectively.



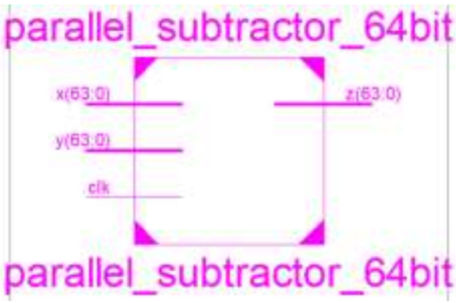
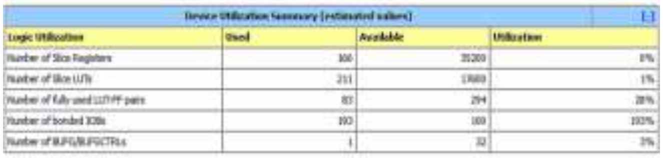


Fig 5 64-bit parallel subtractor IP core. Fig. 6 RTL schematic of 64-bit instantiation parallel subtractor IP core.

Figure 2.6 shows the Xilinx I-SIM simulation [12] timing diagram of the 64-bit parallel subtractor. By default, Xilinx has its own simulator I-Sim but one can install the modelsim[10] also, a product of mentor graphics and X- sim simulator from Xilinx Vivado will also give the same waveforms

One of the input for the above fig 2.6 in the timing window is x[63:0] the Hexadecimal its value “0000000000001086” its decimal value is 4230. Similarly, the input y[63:0] and its Hexa value is “0000000000000934” its value in the decimal is 2356. The output Z[63:0] is the Hexadecimal “0000000000000752” its decimal value is 1874.CLK is the clock pulse.

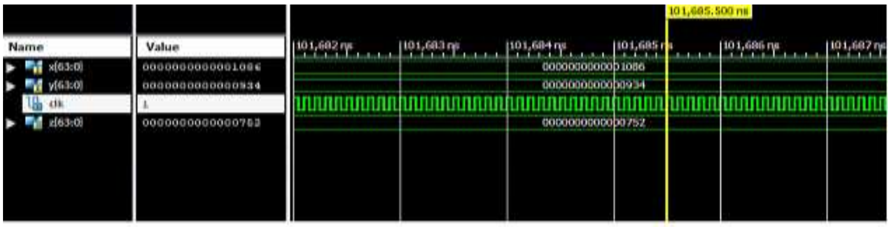
The table 2.1 shows the hardware device utilization report of the parallel subtractor Ip-core with respect to the Zynq-7 series FPGA board. The number of slice flip-flops, 4 inputs Slice Luts, I/O blocks, Buffer memories are 166,211,83,193,1 the MCPD for the parallel subtaractor is **1.158** ns through which 0.878ns for the logic to be completed, 0.279ns for the routing. Total REAL time is 37.00secs and total CPU time is 36.43secs. Overall design memory is 4740892 kilobytes. Thus the proposed method gives less delay as compared to the normal subtractor approach.



Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	166	35200	0%
Number of Slice LUTs	211	17600	1%
Number of fully used LUT-FF pairs	83	294	28%
Number of bonded IOBs	193	100	193%
Number of BUFG/BUFGCTRLs	1	32	3%

Table 2.1 64 bit parallel subtrator IP core synthesis report



Name	Value
x[63:0]	0000000000001086
y[63:0]	0000000000000934
clk	1
z[63:0]	0000000000000752

Fig. 2.6 64 bit parallel subtrator simulation using IP Core

3. CONCLUSION:

The HDL implementation of the parallel subtractor circuit using Xilinx Intellectual Property core generation method and conventional approach technique is compared, Its RTL view, synthesis report, power report and timing waveforms has been tested with recent Zynq-7 XC7Z010 series FPGAs which works on 28nm technology.The FPGA parameters like slice Luts,IOBs,LUT-FF pairs, Registers are tabulated and this device has the ARM processing system and programming logic. The proposed design gives less delay as compared to the conventional approach. The IP core approach method gives 1.158 ns of delay. The total power consumption of the design is 0.56 W by using an X power analyzer.

4. REFERENCES

[1] Jaiswal, Manish Kumar, and Ray CC Cheung. "High-performance FPGA implementation of double precision floating point adder/subtractor." International Journal of Hybrid Information Technology 4.4 (2011): 71-80.

[2] Rudnicki, Kamil, and Tomasz P. Stefanski. "IP core of coprocessor for multiple-precision-arithmetic computations." 2018 25th International Conference" Mixed Design of Integrated Circuits and System"(MIXES). IEEE, 2018.

[3] Lee, Boseon, and TaewonSuh. "Towards Characterization of Modern FPGAs: A Case Study with Adders and MIPS CPU." The Journal of Korean Association of Computer Education 16.3 (2013): 99-105

[4] Basha, Mudasar, M. Siva Kumar, Vemulapalli Sai Pranav, and B. KhaleeluRehman. "Approach to Find Shortest Path Using Ant Colony Algorithm." In Intelligent Communication, Control, and Devices, pp. 1243-1252. Springer, Singapore, 2018.

[5] Crockett, Louise H., et al. The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc. Strathclyde Academic Media, 2014.

[6] Kumar, Adesh, Gaurav Verma, Mukul Kumar Gupta, Mohammad Salauddin, B. KhaleeluRehman, and Deepak Kumar. "3D Multilayer Mesh NoC Communication and FPGA Synthesis." Wireless Personal Communications 106, no. 4 (2019): 1855-1873.

[7] Reddy, K. Venkata Siva, P. Vishnu Kumar, K. Maheswari, and B. KhaleeluRehman. "Design and Verification of 16-Bit Vedic Multiplier Using 3: 2 Compressors and 4-Bit Novel Adder." In Proceeding of International Conference on Intelligent Communication, Control, and Devices, pp. 723-732. Springer, Singapore, 2017.

[8] Rehman, B. Khaleelu, Adesh Kumar, and Paawan Sharma. "Modeling and Simulation of ECG Signal for Heartbeat Application." Intelligent Communication, Control, and Devices. Springer, Singapore, 2018. 503-511.

[9] Lysaght, Patrick, et al. "Enhanced architectures, design methodologies, and CAD tools for dynamic reconfiguration of Xilinx FPGAs." 2006 International Conference on Field Programmable Logic and Applications. IEEE, 2006.

[10] Mohammad, Salauddin, R. Seetha, S. Jayamangala, B. KhaleeluRehman, and D. Kumar. "Implementation of FM-Based Communication System with 3-Level Parallel Multiplier Structure for Fast Transmission Using FPGA." In Intelligent Communication, Control, and Devices, pp. 619-626. Springer, Singapore, 2020.

[11] Rehman, B. K., Kumar, A., Mohammad, S., Basha, M., & Reddy, K. V. S. (2020). Detection of DTMF by Using Goertzel Algorithm and Optimized Resource-Sharing Approach. In Intelligent Communication, Control and Devices (pp. 851-856). Springer, Singapore.