An Efficient Voltage and Frequency Division Methods for Multiprecision Multiplier on FPGA

Mrs. N.S.Labhade (Assistant Prof.)[1], Sandip Chougule[2], Santosh Gadikar[3], Sunil Sharma[4]

(Dept. of E&TC, ICOER Wagholi, Pune)

Abstract— *In this paper, we present a reconfigurable multiprecision(MP) multiplier that contains variable precision ,parallel processing, Dynamic voltage scaling (DVS)and dedicated MP operands scheduling to provide desired performance for various operating conditions. The proposed reconfigurable multiplier works independent smaller precision also it works in parallel to perform higher precision multiplications according to conditions or user requirements of voltage or frequency the dynamic voltage/frequency scaling unit configures to works at proper precision and frequency. In this paper we design a multiplier circuit that consumes less power and reduces area overhead by parallel processing .for this instead of using 16bit or 32bit operands we used single 8bit or twin parallel bit multiplication operation . in multipresicion multiplier small multiplications causes complex structures which results in unwanted signal generation and increases complexity of circuitary ,for this we used 8bit multiplier for 16 bit and 32bit multiplication hence to reduce power consumption and according to runtime workload we combine multiprecision multiplier with dynamic voltage scaling(DVS).The DVS technique consists look up tables(LUT) and on chip critical path replica approach .The LUT allows the supply voltage according to the voltage frequency relationship stored in LUT.*

Keywords— Multiprecision, Reconfigurable.

I. INTRODUCTION

A nowadays Consumer demand for increasingly portable and high performance multimedia and communication products imposes strong constraints on the power consumption of individual internal components [1]. Of these multipliers perform one of the most frequently encountered arithmetic operations in digital signal processors (DSPs) [2]. For embedded applications, it has become essential to design more power-aware multipliers. Given that the fairly complex structure and interconnections, multipliers can exhibit a large number of unbalanced paths, resulting in glitch generation and propagation [3]. This unwanted switching activity can be overcomed by balancing internal paths through the combination of architectural and transistor-level optimization techniques. In addition to that equalizing internal path delays, dynamic power reduction can also be achieved by monitoring the effective dynamic range of the input operands so as to disable unused sections of the multiplier, and/or shorten the output product at reduced precision. It is possible because, in most sensor applications, the actual inputs do not always occupy the entire magnitude of its word-length. The operations in lower precisions are the most frequently required. In contrast to that most of today's full-custom DSPs and application-specific integrated circuits (ASICs) are designed for a fixed maximum word-length so as to accommodate for the worst case scenario. Therefore, an 8-bit multiplication computed on a 32-bit Booth multiplier would result in unnecessary switching activity and power loss.

Several works had been done on this word-length optimization. [1] Proposed system of multipliers of different precisions with each optimized to cater for a particular scenario. Each pair of incoming operands is routed to a smallest multiplier that can compute the result to take advantage of the lower energy consumption of the smaller circuit. This ensemble of point systems is reported to consume the least power but this can increase the chip area. for this, [5] proposed to share and reuse some functional modules within the ensemble. In [3], an 8-bit multiplier is reused for 16-bit multiplication, adding scalability without large area coverage. [5] Extended this method by implementing pipelining to further improve the multiplier's performance. A more flexible approach is proposed in [15], with several multiplier elements grouped together to provide higher precisions and reconfigurability. Reference [7] analyzed that the overhead associated to such reconfigurable multipliers. Combining multiprecision (MP) with dynamic voltage scaling (DVS) can provide reduction in power consumption by adjusting the supply voltage according to circuit's run-time workload rather than fixing it to cater for the worst case scenario [4]. Conventional DVS techniques consist of lookup table (LUT) and on-chip critical path replica approaches [17]–[19]. In addition, the critical path may change as a result of the varying supply voltage or process or temperature variations. Recent DVS approaches can overcome the limitations of the conventional DVS [24], because it completely removes worst case safety margins and error-tolerant DVS techniques can further aggressively reduce power consumption. In this paper, we propose a lower power reconfigurable multiplier architecture which combines MP with an error-tolerant DVS approach. [25]

II. WORK FLOW

The objective of this paper is to implement an efficient Dynamic voltage and frequency scaling (DVFS) for multi precision multiplier on reconfigurable FPGA (spartan3).

To Implement of DVFS for multiplier block is on FPGA and which is offering good performance at low power supply voltage 3.3V to 1.8V and frequency 32MHz to 8MHz.To avoid unnecessary switching activity dynamic voltage and frequency scaling is performed.

To generate control signal, Scanning of input data is carried out.

To Address the issues of fix length ASIC's.

The following figure shows the operation flow of the system.

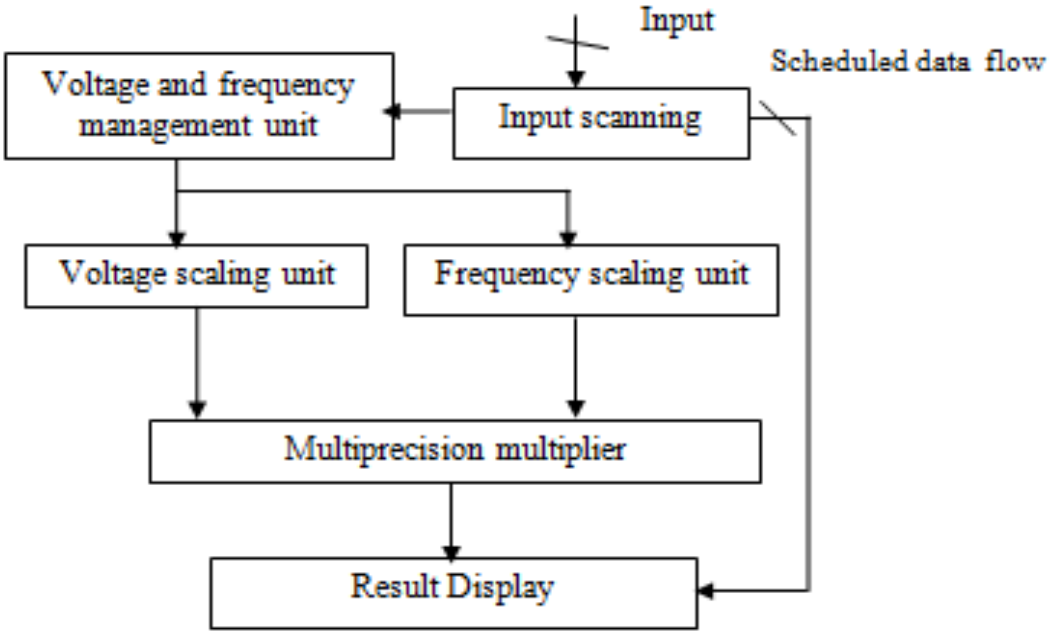


FIG. 1 WORK FLOW OF MULTIPRECISION MULTIPLIER

III. METHODOLOGY

3.1 Scanning

Scanning is vital step for making system dynamic. This is the first and important step, by this step range of input is detected. After the scanning it will generate the control signal 1, 2, 3 for 8bit, 16bit, and 32bit data.

3.2 Voltage Scaling

Here, frequency scaling is used to conserve the power. For stable operation of voltage required is determined by the frequency at which circuit is clocked, and can be reduced by if the frequency of the system is reduce.

3.3 Frequency Scaling

In an integrated circuits(IC) at the time of implementation manufacturer kept the fixed word length that is not used in all the application. To avoid this voltage scaling is important according to workload to system.

3.4 Multiplication

The Multiplier is controlled by three external signals. Once the input is scanned and detected the operating voltage and frequency are tuned automatically. The voltage range and frequency of 1.2-3.3V and 32-8MHz is achieved for full functionality respectively

IV. RESULTS

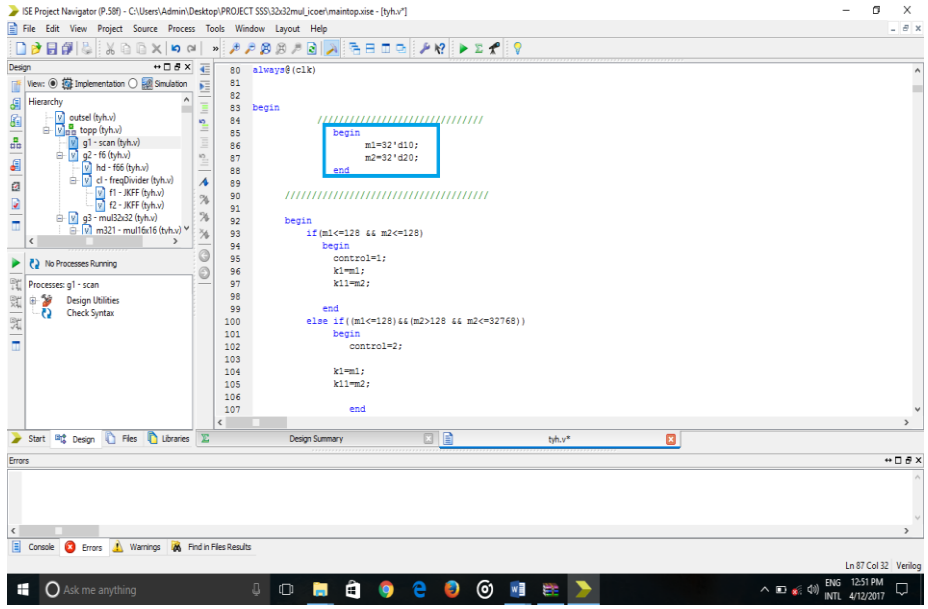
TABLE 1
VOLTAGE AND FREQUENCY DIVISION

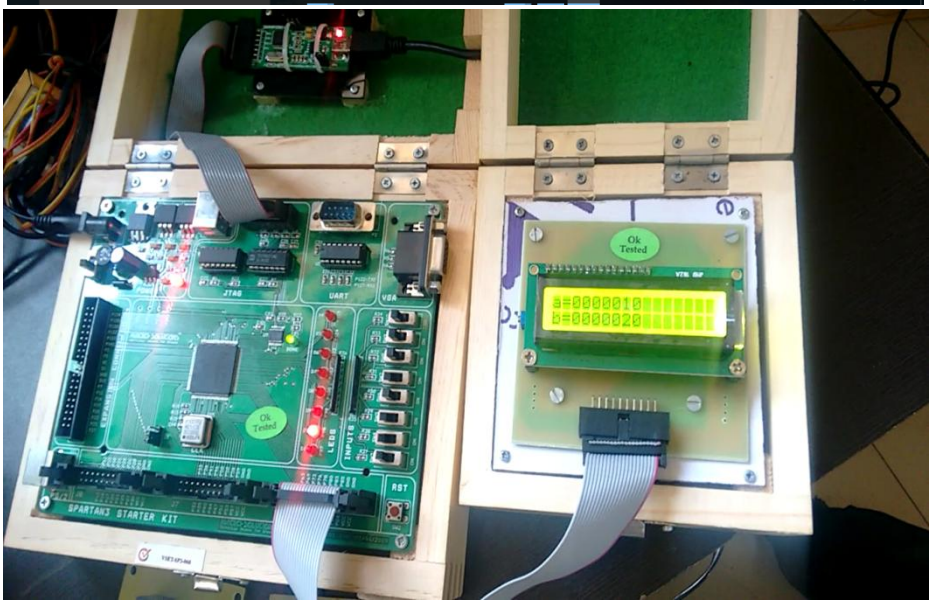
Scanning range	Voltage(v)	Frequency(MHz)
8*8	1.8	25
16*16	2.8	50
32*32	3.3	100

TABLE 2
MULTIPLICATION RESULTS

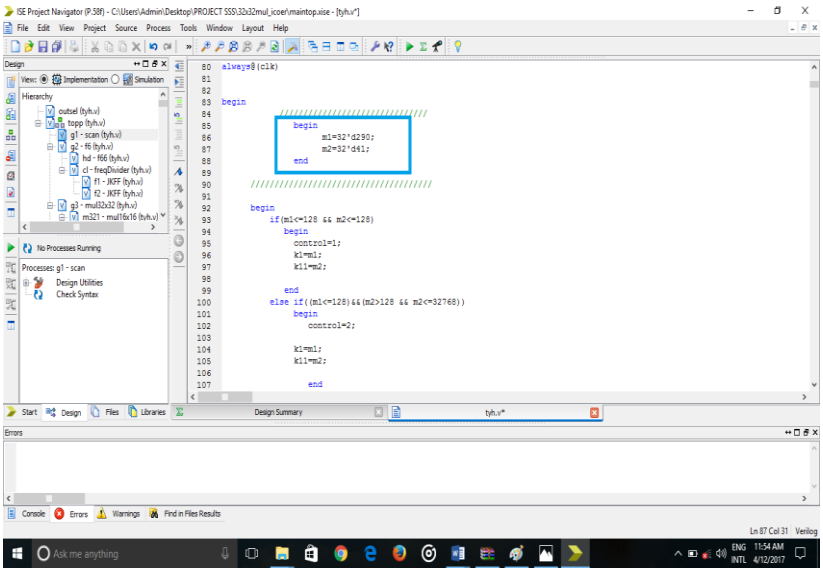
	8*8	16*16	32*32
A	10	290	70000
B	20	41	20
Result	200	11890	1400000

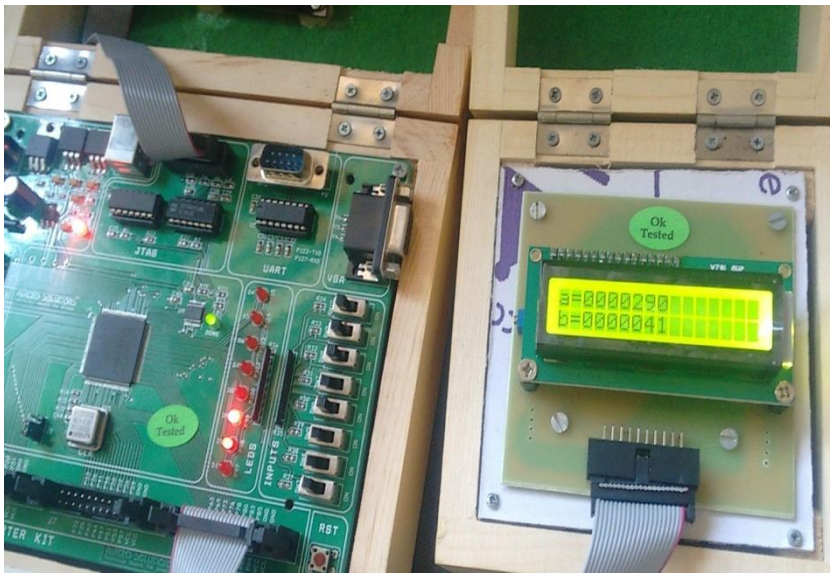
4.1 8 * 8 Bit Multiplication

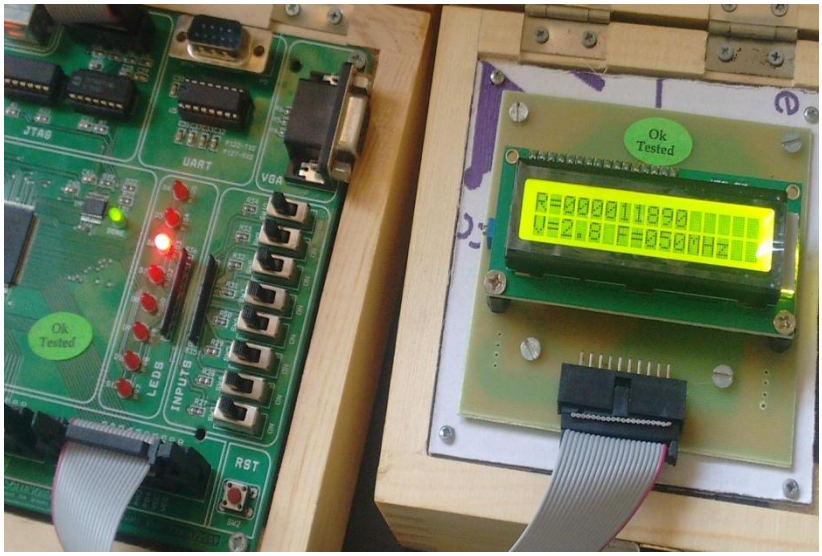




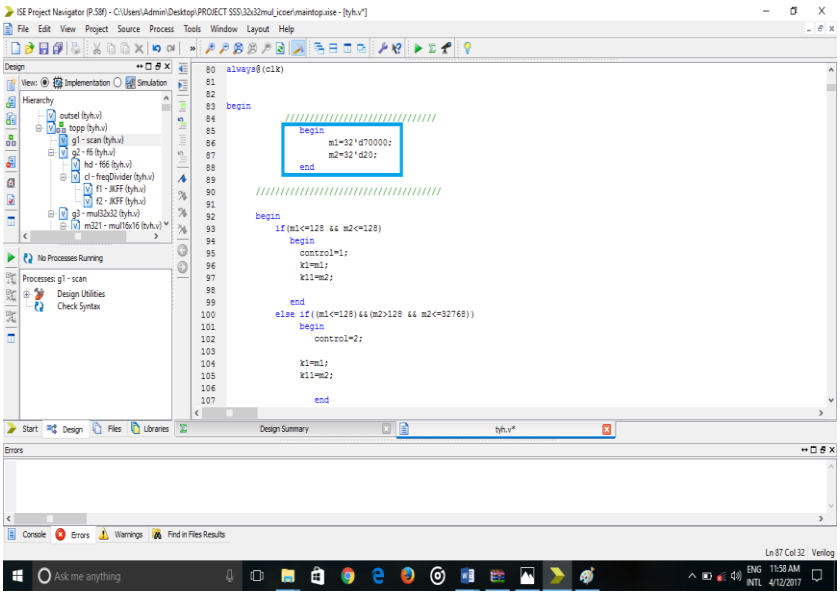
4.2 16 * 16 Bit Multiplication

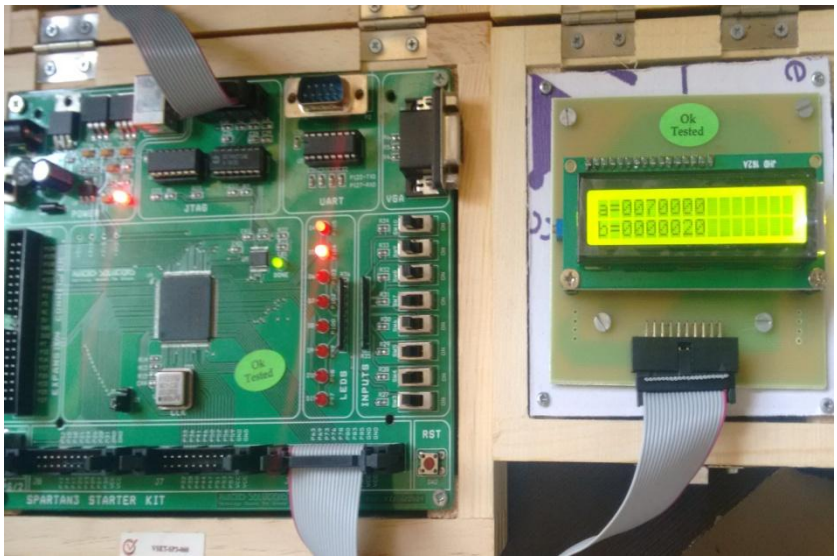


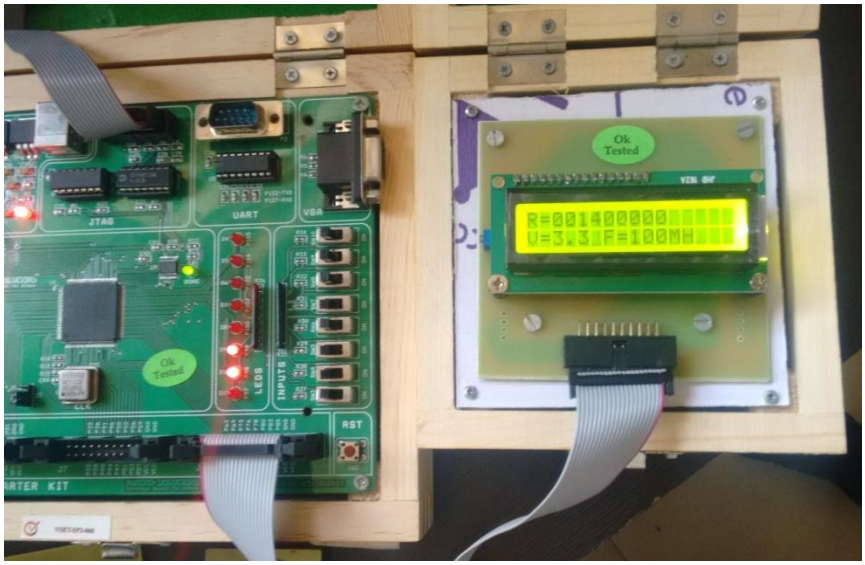




4.3 32 * 32 Bit Multiplication







V. CONCLUSION

Today energy efficient devices are the world's necessity. This paper presented techniques for efficient use of hardware resources power on FPGA. Here dynamically adjusting the frequency and voltage according to bit width of input data by scaling technique. As input is known the system can save the power by disabling the unused selection of multiplier

REFERENCES

[1] R. Min, M. Bhardwaj, S.-H. Cho, N. Ickes, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan, "Energy-centric enabling technologies for wireless sensor networks," IEEE Wirel. Commun., vol. 9, no. 4, pp. 28–39, Aug. 2002.

[2] T. Kuroda, "Low power CMOS digital design for multimedia processors," in Proc. Int. Conf. VLSI CAD, Oct. 1999, pp. 359–367

[3] F. Carbognani, F. Buergin, N. Felber, H. Kaeslin, and W. Fichtner, "Transmission gates combined with level-restoring CMOS gates reduce glitches in low-power low-frequency multipliers," IEEE Trans. Very Large Scale Integer. (VLSI) Syst., vol. 16, no. 7, pp. 830–836, Jul. 2008.