International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 6, June - 2014

ASIC Implementation of Two Stage Pipelined Multiplier

Hitender Singh Rathore
VLSI Design,
VIT University, Vellore

Heena Dua
VLSI Design,
VIT University, Vellore

Sivanantham S
Assistant Professor,
VIT University, Vellore

***Abstract*—Multiplication is one of the mostly used operations in all of the devices. This paper presents an efficient implementation of a pipelined multiplier designed with two stage pipelining and performed backend designing using Encounter tool provided by Cadence. This design is compared with shift and add multiplier, power and timing analysis for this design are performed using Cadence tool. A range of multipliers architectures are available based on applications. The pipelined circuits are designed so that we can achieve high performance for a system as they can be operated at higher frequency. From the implementation results, it is verified that two stage pipelined circuit is faster by a factor of two times than the non pipelined circuit. But we get an area and power overhead. Therefore, this circuit can be used where the performance of system is of major concern.**

Keywords: Multiplier; Verilog; Pipelining; ASIC; Encounter.

I. INTRODUCTION

In today's era everybody wants high performance system where power is not considered the major factor because speed and power of the circuits cannot be optimized simultaneously. As if we want to reduce the power consumption then the system would become slow. Therefore power dissipation and speed of the circuits cannot go hand in hand. Hence, in this paper the multiplier is designed which is application specific that is which will be used for high performance system only because the power and area of the circuit are increased as compared to non- pipelined circuits. Now these days, multipliers are being used as one of the fundamental blocks. They play an important role in today's digital signal processing, FIR filters, microprocessors and various other applications. A system's performance is generally determined by the performance of the multiplier. So they are highly preferred, as they offer high speed, low power consumption and regularity in layouts. There are different types of multipliers like booth multiplier, serial multiplier, it depends for which application we want to use. Here we have used shift and add multiplier for non-pipelining circuit and inserted two stages of pipelining in the same multiplier for the comparison in between them. Multipliers are widely used in DSP operations such as convolution for filtering, correlation and filter banks for multi rate signal processing. Without multipliers, no computations can be done in DSP applications. For that reason, multipliers are chosen for pipelining in our proposed design.

II. DESCRIPTION OF MULTIPLIERS

We have taken the 8*8 bit shift and add multiplier which is designed by Verilog Code. The algorithm followed in the shift and add multiplier is that first we need to generate partial products, that are added together in order to give final output. After generation of each partial product multiplicand is shifted by one bit in order to perform next bit multiplication operation. We have one control block which checks for existence of multiplicand bit if it is one then only multiplication is performed else we need to set partial product equal to zero as there is no need to perform this bit multiplication.

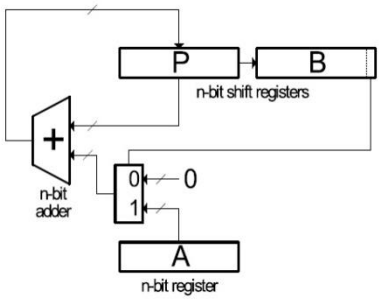
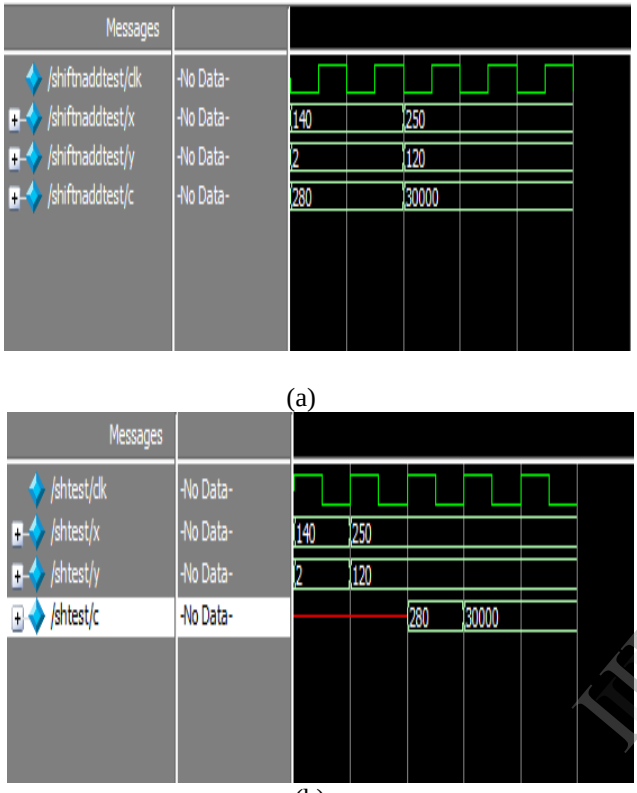


Fig1: Architecture of Shift and Add Multiplier

The architecture of shift and add multiplier is shown in figure1 above, this is most common architecture used to explain the working of a basic multiplier. The two stage pipelining is implemented in this shift and add multiplier for high performance as the basic principle of pipelining is that it divides the work into segments and each segment can execute its operation concurrently with other segments. When a segment completes an operation, it passes the result to the next segment in the pipeline and fetches the next operation from the preceding segment. The final results of each instruction emerge at the end of the pipeline in rapid succession. Pipelining reduces the effective critical path by introducing pipelining latches along the critical data path. It is simple to understand that in same clock period if we need to perform half of the operation than we can decrease the supply voltage or we can change the operating frequency keeping the supply voltage constant. In this paper we kept the supply voltage constant and changed the operating frequency, as this is two stages of pipelining so theoretically the output frequency should be made twice of the previous frequency. After designing this

and performing timing analysis it is seen that for the clock period of approximately 1550ns the same performance of the circuit is achieved whereas in case of the non-pipelining circuit the operating frequency was 3100. The pipelining in the circuit is achieved by adding registers in between them and designed in Verilog code. Then the circuit is simulated in order to be verified with a sample test-bench, the results are as follows



(a)

(b)

Fig2: Comparison of outputs of multipliers

III. DESIGN OF MULTIPLIER

The architecture of our designed pipeline multiplier is shown in figure 3, in which the multiplier is divided in two parts. For first four bits it will be calculating the partial products in one clock cycle and in next clock cycle it will calculate the partial products for last four bits and then these partial products are given as input to an adder block which will generate the final output consisting of 16 bits of resultant.

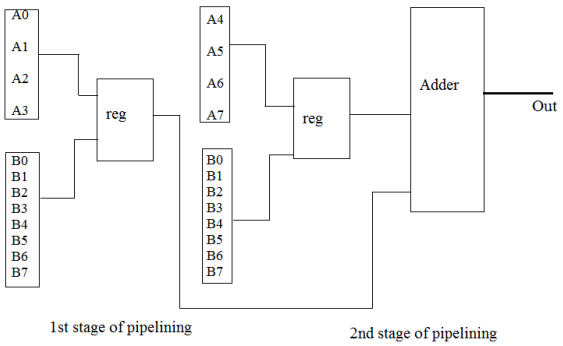


Fig3: Architecture of pipelined multiplier

After this design, we have performed backend operations for synthesizing this design. For that first step is to generate the synthesized netlist, which is given as input to IO file which contains the mapping commands for IO pads and corner pads. In Encounter tool first thing after importing the design is to do Floor planning, in Floor planning we took core ratio as 1 and core utilization as 0.6. Then we have to go for connection of power nets, after that power planning is performed. In power planning we have included rings and strips. After this we go for special route followed by standard cell placement. At last routing is performed in order to connect all the IO pads to internal circuitry. The final Tap out diagram obtained for this design is shown in figure4.

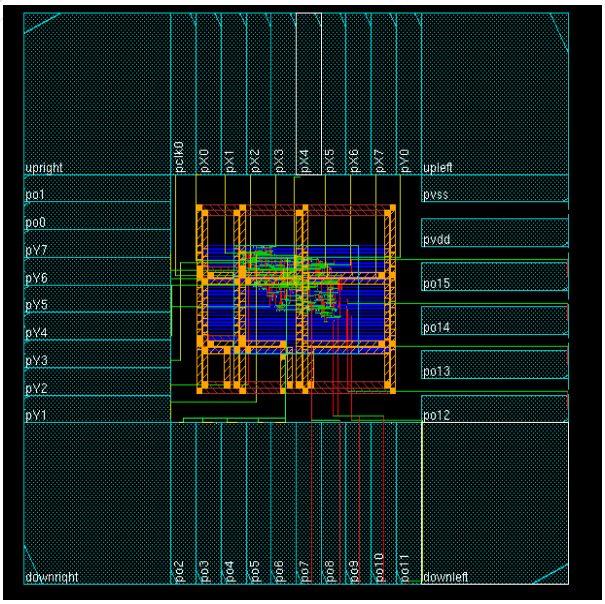


Fig4: Tap out diagram of Pipelined Multiplier

IV. SIMULATION RESULTS

These circuits are implemented using 180nm technology in ASIC. Verilog HDL language is used to describe the functionality of the circuit and for the timing, area and power analysis the Cadence tool is used.

Schemes	Area (cell)	Power (µW)	Frequency(MHz)
Pipelining	312	549.650	645.16
Non-Pipelining	169	465.784	322.580

Table1. Implementation Results of Multipliers

From table1, it is noticed that for the pipelining circuit the frequency is twice as compared with the non pipelining circuit whereas the area and power of the non-pipelined circuit is less as compared to the pipelining circuit because the area increases in the pipelined circuit as we have introduced the registers in between them. Therefore for high speed of the system we have to go with area and power overhead.

V. CONCLUSION

The proposed scheme is implemented for the pipelined circuit and it is tested using 8*8 shift and add multiplier. From the implementation results, it is verified that the pipelined multipliers are faster by a factor of two as compared to non-pipelined multipliers.

REFERENCES

[1.] A.Chandrakasan and R. Brodersen, "Low Power CMOS Digital Design", IEEE J. Solid State Circuits, Vol.27, no.4, pp 473-484, Apr 1992.
[2.] Jia Di_, J.S. Yuan, R. Demara "Improving power-awareness of pipelined array multipliers using two-dimensional pipeline gating and its application on FIR design", Microelectronics Journal 41 (2010) 639–650.
[3.] Ko-Chi Kuo n, Chi-Wen Chou "Low power and high speed multiplier design with row bypassing and parallel architecture ",Microelectronics Journal 41 (2010) 639–650.
[4.] Rengaprabhu Paramasivam, Venkatasubramanian Adhinarayanan, and Seetharaman Gopalakrishnan, "Design and Implementation of Automated Wave-Pipelined Circuit using ASIC", 2012 Third International Conference on Intelligent Systems Modelling and Simulation.
[5.] Refik Sever , Murat Askar "8x8-Bit Multiplier Designed With a New Wave-Pipelining Scheme", IEEE Trans. VLSI Syst., vol. 6, no. 3, pp. 2095-2098, Sep. 2000.