A Study of Large Width Unsigned Multipliers on FPGAs

Ahmed Sayed and Mohamed Aly

***Abstract*—Multiplication is an important fundamental operation in most signal and image processing applications. High definition image processing has put a huge demand on fast and massive data processing and shrinking the CMOS process made the silicon real estate available to provide for such massive data processing building blocks. We compare large width multipliers from an architecture point of view, maximum clock frequency, latency, throughput, resource usage, power consumption. We use a flopped combinational baseline multiplier for our comparison and we use the same FPGA platform to be fair in our analysis. We mention some remarks and conclude that shift and add is the best.**

***Index Terms*—Computer arithmetic, FPGA, low power, multiplier, verilog.**

I. Introduction

With the increased reliance on mobile devices in our lives and with this trend growing with time, the importance of battery life becomes more and more critical. Also with the reduction of feature sizes in CMOS technology more silicon real estate is available to add functionality or improve current services. These factors mandate a detailed study of the different design tradeoffs that a design engineer would consider while trying to achieve specific product goals.

Usually a design engineer has the freedom to choose the details of the design that serves the functionalty required at the target frequency. There is the typical tradeoff between area, speed and power consumption. Customers always expect more services and functionality of their mobile devices. With more transistors being squeezed in the same area the more functionalty we can add to the product, this limits how much real estate we can spend on improving current designs and services.

This cramming of transitors and adding new features puts very high emphasis on low power consumption for each and every design component to prolong operating time, especially with the increase of leakage as technologies shrink. Power consumption is composed of three components; static, dynamic and leakage power consumptions as shown in (1) below.

$$P_{Consumption} = P_{Leakage} + P_{Static} + P_{Dynamic} \quad (1)$$

CMOS processes do not exhibit static power consumption. Current technology processes suffer from leakage and this power component will continue to grow as the process shrinks. The dynamic power consumption is dependent on the switching events at every node of the circuit. One part of it is short circuit current while transistors switching and the other is the charging and discharging of load capacitances. Good choice of gate sizes would reduce the short circuit component dramatically as a result of sharp transitions of signals across the circuit. The charging and discharging of capacitance part is shown below in (2).

$$P_{Dynamic} = \sum \alpha \, C_{load} \, V^2 \, F \quad (2)$$

Equation 2 states that any node that switches within the circuit will consume power directly proportional to the capacitance charged the supply voltage and that node's swing voltage. This said, how often this node's load charges or discharges identifies its share of the total dynamic power consumption of the circuit and this is represented in the activity factor α*F which represents the toggle rate.

Power analysis can be either done staticaly or dynamicaly. That is to say, either with establishing the toggle rate of each node staticaly based on known inputs toggle rates, or using explicit simulations to run the circuit and capture the internal switching dynamicaly for a specific simulation.

Static power analyses are usually used for estimating average power. Dynamic power analysis can be used to measure the average power over a specific period and for a specific test case(s). Having insights in specific test cases that are of special interest for example one usage mode that dominates the duration of operation; would result in better estimates. Dynamic power analyses are also used for peak power estimation when there is high confidence that a specific test case has proven to consume maximum power, this is very helpful in power grid design.

In the wireless mobile communications area, especially after mobile video being on the rise, average power consumption is of significant importance. It is advertised as the battery operating time expectancy for different mobile devices, hence we focus on average power in our study. Multiplication is inherent in the hardware implementatation of any algorithm, be it in the signal, image processing or communications arenas, therefore details and optimizations of different multiplier architectures are of prime importance in the wireless communications field.

FPGAs have proven to be the fastest prototyping platform for any integrated circuits application as shown in [1], [2]. Power consumption estimations are very much more accurate in FPGAs than in front end ASICs. Therefore, an FPGA platform has been chosen for our study.

The paper is organized as follows; in Section II we go over the different architectures considered in our study. Section III covers the comparison of implementation results with respect to specific metrics. Section IV covers our observations, remarks and recommendations; and we wrap

Manuscript received September 10, 2012; revised November 29, 2012.

The authors are with the Varkon Semiconductos (email: asayed@varkonsemi.com; mohamed.aly@varkonsemi.com).

up with our conclusion.

II. Architectures Considered

Traditional multipliers like in [3]-[5] can prove to have prohibitive area, speed and/or power consumption for large widths, this has been alluded to in [6]. In our current study we focus on multipliers that are well suited for large bit widths.

The following are five multiplier architectures considered in the current study, they can be easily found in the literature so we will only mention a brief note about each. References [7]-[13] have more detailed information about large width multipliers, further references are mentioned as we go.

A. Unsigned CombinationalF

The unsigned flopped combinational "UCombiF" design is our baseline for our comparison study. It heavily relies on FPGA DSP elements unlike other architectures. DSP elements are supposed to be intentionally designed for this purpose. This design focuses on the regular approach of using the DSP via the "*" element in the Verilog code as shown in Fig. 1.

```
always@(multiplier, multiplicand)
  begin
    product<=multiplier*multiplicand;
  end
```

Fig. 1. Code snippet of UCombiF

The "UCombiF" multiplier is a dual stage pipelined version of the combinational multiplier so as to establish the maximum clock frequency of the design.

B. Divide and Conquer

Divide and conquer "DC" [14] is used when we need to minimize the clock period. It also uses DSP elements, but it divides both multiplier and multiplicand into two segments to calculate partial products concurrently to reduce clock period.

C. Karatsuba-Ofman

Karatsuba-Ofman multiplier "KO" [15]-[16] is used in the same cases as "Divide and Conquer", but "KO" further reduces the number of used DSP elements.

D. Broadcast

Broadcast "BC" multiplication [17] reduces the number of multipliers used to only one, on the down side it reduces throughput. It does this by segmenting both multiplier and multiplicand into (order) k segments then calculates iteratively partial products.

E. Unisigned Shift and Add

The unsigned shift and add multiplier "USA" [18] follows the basic multiplication that humans go through when multiplying any two multi-digit numbers. This is particularly useful when the number of DSPs on an FPGA has been exhausted or limited for more critical functions.

III. FPGA Design and Implementation Results

The designs of "UCombiF", "DC", "KO" and "BC" and "USA" unsigned multipliers 4×4, 8×8, 16×16, 32×32, 64×64 and 128×128-bit are implemented using Verilog and on a Xilinx Virtex-6 FPGA, device XC6VLX760, package: FF1760, speed grade: -1, time constraint: 2ns, and using the Xilinx ISE 13.2 design tool suite. Using the same FPGA device for all designs and reports is a way of setting a fair plane level of comparison among the different architectures.

A. The Hardware Resource Usage Results

For our study we only consider register, LUT, slice, DSP and RAM usage since these are the most important hardware resources which FPGA digital designers are interested in.

After having a closer look at Fig. 2 we notice that the best from LUT usage point of view is the "UCombiF" multiplier and the worst one is the "BC" multiplier with the highest k.

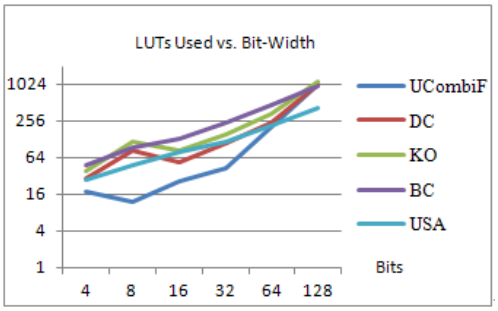

Fig. 2. LUTs usage (log scale)

It is noticeable from Fig. 3 that the best one from register usage point of view is the "USA" multiplier and the worst one is the "DC" multiplier.

Fig. 4 shows slices used by different types of multipliers. We notice that the lowest usage of slices is the "UCombiF" multiplier and the worst one is the "BC" multiplier with the highest k.

The inconsitent trending in Fig. 3 and 4 is due to the sudden use of DSPs going from 8bits to 16bits wide multipliers.

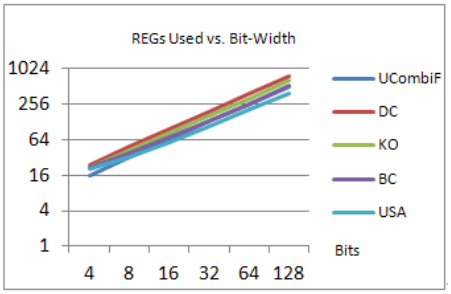

Fig. 3. Registers usage (log scale)

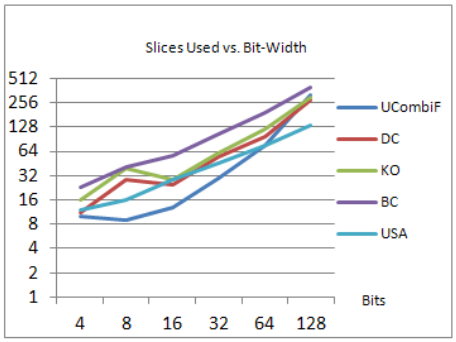

Fig. 4. Slices usage (log scale)

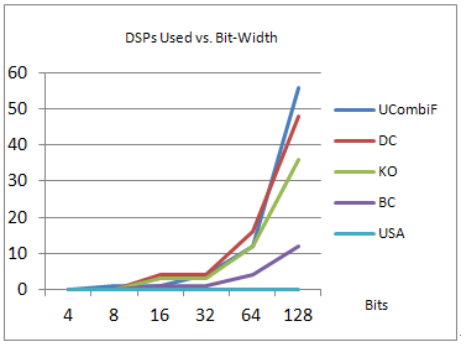

Fig. 5. DSP elements usage

Fig. 5 shows DSP usage for different types of multipliers. As mentioned above the "USA" multiplier does not use any DSPs. Out of the rest of the multipliers we notice that the "BC" multiplier with the highest k is the best from DSP elements usage point of view, and the worst is the "UCombiF." Finally, for RAM usage none of the multipliers uses RAM blocks.

B. *Minimum Clock Period, Latency and Throughput Results*

Fig. 6 shows minimum clock period for different types of multipliers. We notice that the best one from minimum clock period point of view is the "USA" multiplier.

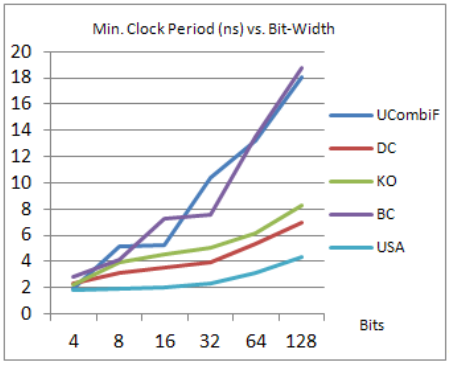

Fig. 6. Minimum clock period

Latency is definitely a function of the pipeline depth which is two for "UCombiF", "DC", "KO" and "USA", even if "BC" is an iterative multiplier, it generates the first bit in two clocks, so the latency as we define it, is two clocks for all the types of multipliers studied.

From Table I we notice that the lowest througput is for "BC" with the highest k, or the "USA" in other cases and all other multipliers are similar.

TABLE I: THROUGHPUT (BITS/CLOCK)

Data width	UCombiF	DC	KO	BC	USA
N	2N	2N	2N	$\frac{2N}{K^2}$	$\frac{2N}{3N+2}$

C. *Power Consumption Results*

We only consider dynamic power consumption for our study at target frequency 500MHz and 12.5% toggle rate on inputs, since leakage is only process, voltage and temperature (PVT) dependent, also CMOS technologies do not exhibit staic power consumption. We break down the dynamic power into clock power and others. This is to differentiate between a heavily registered or pipelined multiplier design and a mostly combinational one.

Fig. 7 shows clock dynamic power consumed. We notice that the best one from clock dynamic power consumption perspective is the "USA" multiplier for high bitwidths. The inconsistent trending is still under investigation.

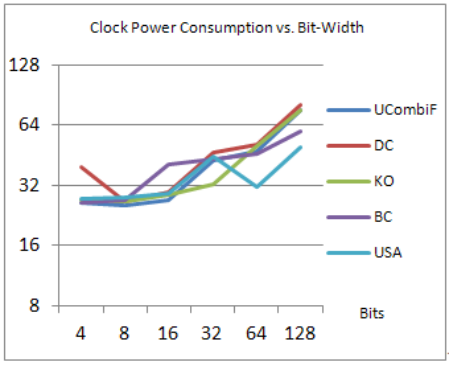

Fig. 7. Clock dynamic power consumption (log scale)

A closer look at Fig. 8 shows that the best one from others dynamic power consumption is the "USA" multiplier and the worst is the "KO" multiplier.

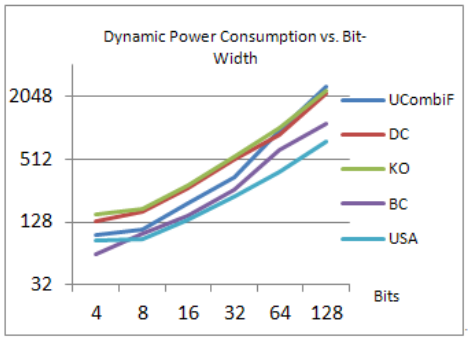

Fig. 8. Others dynamic power consumption (log scale)

IV. REMAKRS AND RECOMMENDATIONS

In this paper we have demonstrated results of FPGA based architectures for "UCombiF", "DC", "KO", "BC" and "USA" 4×4, 8×8, 16×16, 32×32, 64×64 and 128×128-bit unsigned multipliers using Verilog. The designs were implemented on XC6VLX760 Virtex-6 FPGA device using the Xilinx ISE 13.2 design tool suite. These are some remarks:

1) From an area perspective, "UCombiF" highly utilizes area usage on the other side it uses more DSP elements than others. The "BC" is the worst in this category.
2) It is worth noting that the "USA" multiplier starts to be the best in very high bitwidths from an area point of view.
3) From a speed perspective, "USA" has the lowest minimum clock period "DC" is second best. The worst is the "UCombiF."
4) From a clock and others dynamic power consumption the "USA" seems to dominate at higher bitwidths with more investigation to be exerted in the low bitwidth range. The worst is the "KO."

V. Conclusion

We presented a comparative study of different large width unsigned multipliers using the same FPGA device. It is obvious from our study that on the Viertix6 FPGA the unsigned shift and add "USA" multiplier outperforms all other architectures that have been reported for large bit width designs. Future work will explore this finding on other FPGAs and also in the ASIC design area.

References

[1] Maxfield, "The design warrior's guide to FPGA devices, tools and flows," *Newnes Publishers*, MA, 2004.

[2] S. Hauck, "The roles of FPGAs in reprogrammable systems," in *Proc. of the IEEE*, vol. 86, no. 4, pp. 615 – 638, April 1998.

[3] Koren, "Computer arithmetic algorithms," *A K Peters/CRC Press*, Natick, Mass, USA, 2nd edition, Dec. 2001.

[4] Bewick, *Fast multiplication: algorithm and implementation*, Ph.D. Dissertation, Stanford University, February 1994.

[5] S. Kumar, K. Forward, and M. Palaniswami, "A fast-multiplier generator for FPGA," in *Proc. of the 8th International Conference VLSI Design*, New Delhi, Jan. 1995.

[6] M. Aly and A. Sayed, "A study of signed multipliers on FPGAs," in *proc. of ICEDSA*, Malaysia, Nov. 5-6, *2012*.

[7] Quan J. Davis, S. Devarkal, and D. Buell, "High-level synthesis for large bit-width multipliers on FPGAs: a case study," in *Proc. of the 3rd IEEE/ACM/IFIP International Conference on Hardware/software Codesign and System Synthesis*. pp. 213 - 218, New York, 2005.

[8] C. Kumar and R. Saravanan, "VLSI design for low power multiplier using full adder," *European Journal of Scientific Research*, ISSN 1450-216X, vol. 72, no.1, pp. 5-16, 2012.

[9] N. Nedjah and L. Mourelle, "A reconfigurable recursive and efficient hardware for karatsuba-of man's multiplication algorithm," in *Proc. of IEEE Conference on Control Applications*, pp. 1076, Istanbul. June 2003.

[10] S. Gao, D. Al-Khalili, and N. Chabini, "Optimized realizations of large-size two's complement multipliers on FPGAs," in *Proc. of the IEEE Northeast Workshop on Circuits and Systems*. pp. 494–497, Montreal, Aug. 2007.

[11] S. Gao, N. Chabini, D. Al-Khalili, and P. Langlois, "Optimized realizations of large integer multipliers and squarer using embedded block," *IET Computers & Digital Techniques*, vol. 1, pp. 9-16, 2007.

[12] S. Gao, N. Chabini, and D. Al-Khalili, "256×256-bitmultiplier using multi-granular embedded DSP blocks in FPGAs," in *Proc. Joint 6th International IEEE Northeast Workshop on Circuits and Systems*, pp. 253-256, Montreal, June 2008.

[13] S. Gao, D. Al-Khalili, and N. Chabini, "Efficient scheme for implementing large size signed multipliers using multi granular embedded DSP blocks in FPGAs," *International Journal of Reconfigurable Computing*, vol. 2009, January 2009, no.1. Hindawi Publishing Corp, New York, United States.

[14] S. Dasgupta, C. Papadimitriou, and U. Vazirani, *Algorithms*, ISBN 0073523402. McGraw-Hill Higher Education, September 13, 2006.

[15] M. Machhout, M. Zeghid, W. El Hadj Youssef, B. Bouallegue, A. Baganne, and R. Tourki, "Efficient large numbers karatsuba-ofman multiplier designs for embedded systems," *IJECE*, vol. 4, no. 9, 2009.

[16] S. Mishra and M. Pradhan, "Implementation of karatsuba algorithm using polynomial multiplication," *IJCSE*, ISSN: 0976-5166. vol. 3, no. 1, Feb -Mar 2012.

[17] Quan D. Buell, J. Davis, and S. Devarkal, "Design exploration of 192-bit elliptic curve adder on the starbridge HC-36 system," in *Katz, R. (ed.), Proc. of MAPLD-2003 Military Applications of Programmable Logic Devices*, NASA Office of Logic Programming, 2003.

[18] C. N. Marimuthu *et al*., "Low power shift and add multiplier design," *International Journal of Computer Science and Information Technology*, vol. 2, no. 3, June 2010.

Ahmed T. Sayed was born in Cairo in 1972, got his BSc. from Cairo University from the electronics and communications department, in 1995 with Honors. He received his M.Sc. in computer engineering in 1997 from the University of Louisiana at Lafayette with Honors. He worked for Intel Corp. for ten years. He received his PhD from the University of California, Davis in 2007, (atsayed@ucdavis.edu). He has taught at several public and private universities in Egypt. He joined IBM Cairo in Sep. 2008 as a research scientist. He holds one patent with Intel and two with IBM. Joined Varkon Semiconductors as a digital design manager starting Feb. 2011. He is interested in low power digital design and parallel programming for signal processing.