Multiplier Structures for Low Power Applications in Deep-CMOS

Dursun Baran, Mustafa Aktan and Vojin G. Oklobdzija*
School of Electrical and Computer Engineering, University of Texas at Dallas
Richardson, TX, 75080-3021
*Klipsh School of Electrical Engineering, New Mexico State University, Las Cruces, NM 88003
Email: {dursun, mustafa, vojin}@acsel-lab.com

***Abstract*—Energy-efficient serial and parallel multiplier structures are explored to see their suitability in the low and ultra low power design regimes. 16×16-bit serial and state-of-art parallel multipliers are compared in 45nm CMOS. A multiplier structure is proposed by optimizing the architecture, gate sizes and the voltage supply. The proposed structure provides 15% more throughput as compared to two-cycle parallel multiplier with the same energy consumption for high speed applications. In the low speed design region, it provides 3.7X energy reduction compared to the serial multiplier.**

I. Introduction

The most efficient multiplier structure will vary depending on the throughput requirement of the application. The first step of the design process is the selection of the optimum circuit structure [8]. There are various structures to perform the multiplication operation starting from the simple serial multipliers [5] to the complex parallel multipliers [3, 4, 10]. Any speed improvement in the multiplier will improve the operating frequency of the digital signal processors or can be traded for energy by optimizing circuit sizes and the voltage supply.

The serial multiplier implements the multiplication in *n* cycles for an *n*×*n*-bit multipliers with least hardware [5]. The *n*×*n*-bit serial multiplier requires a *2n* bit adders and shift registers to implement the multiplication. The core logic of the serial multiplier is the adder and an extensive analysis about binary adders is provided in [9]. An efficient adder design provides considerable energy and delay improvement to serial multiplier. The parallel multipliers are commonly used in high performance digital signal processors [1-4]. They require more hardware compared to the serial multiplier in order to provide performance improvements. There are various ways to implement the parallel multipliers. The simplest way of implementing parallel multiplier is to generate all partial products and reduce them to rows of carry and sum signals. The final step is the addition of the generated carry and sum signals. Booth encoding [1, 2] is widely used in parallel multipliers to reduce the number of generated partial products. As an example, Booth2 algorithm reduces the number of partial products to half by scanning the triplets of the multiplier bits [2]. This reduction is obtained at the expense of a more complex partial product generation circuitry [10]. In [10], the authors showed that Booth encoding provides energy reduction at the relaxed delay targets. As the delay target is getting stricter, Non-Booth multipliers become a more energy-efficient structure [10, 11]. After the partial product generation block, the partial products will be reduced to two rows, a row of sum and a row of carry signals. This reduction is implemented using different algorithms [3, 4, 6]. The number of stages in Wallace tree [3] is proportional to the logarithm of the number of partial products. In TDM (Three Dimensional Reduction Method) [6], the reduction tree is treated as one N^{th}-order compressor that guarantees the global optimum. In this method, the fast input is connected to slow output and vice versa. The addition of the generated sum and carry signals are done in the final adder. An efficient final adder structure was proposed in [7] to exploit the difference in the input signal arrival times.

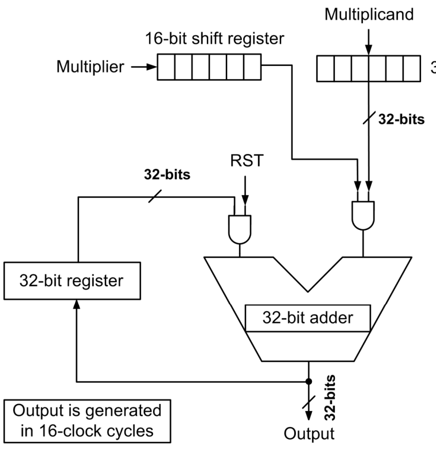
The serial structures are mostly preferred for the ultra low power applications since the speed is not a primary concern. However, the fast structures can be slowed down by optimizing the design parameters such as the gate sizes and the voltage supply. In this paper, we explore the widely used serial and the parallel multiplier structures in energy versus throughput space. In addition, the voltage supply is scaled down to extend the possible design space for each structure. The paper is organized as follows. In Section II, a summary of serial and parallel multipliers are given. The proposed structure is also given in the same section. The results are shown in Section III and the conclusion is made in the Section IV.

II. Multiplier Structures

In this section, the state-of-art structures for serial and the parallel multipliers are summarized. In order to have a fair comparison, unsigned multipliers are selected which are extensively used in floating point arithmetic units. Also, the same analysis can be repeated for the signed arithmetic.

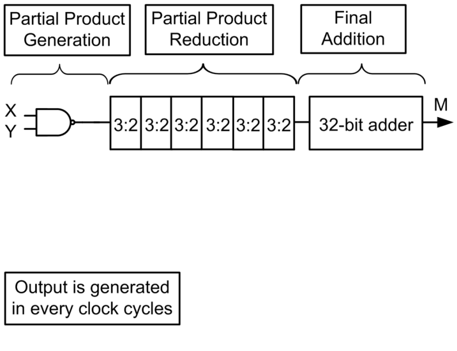
A. Serial Multipliers

Serial multipliers are simply working on the principle of shift and add algorithm [5]. An efficient implementation of the serial multiplier is given in Fig. 1. Static 32-bit Ling adder



and2	64
aoi21	37
aoi22	16
dff	32
inv	35
mux21	32
nand2	28
nor2	54
oai21	27
oai22	15
xnor2	32
xor2	64
*TOTAL : 436 (shift registers are not reported)	
Logic depth : 7	

Fig. 1. The block diagram of the 16×16-bit serial multiplier.


Partial Product Generation
Partial Product Reduction
Final Addition
X
Y
3:2
3:2
3:2
3:2
3:2
3:2
32-bit adder
M
Output is generated in every clock cycles

aoi21	15
aoi22	12
inv	288
mux21	162
nand2	456
nand3	183
nor2	27
nor3	123
oai21	19
oai22	11
xnor2	77
xor2	234
cmp32_p1	187
TOTAL : 1794	
Logic depth : 21	

Fig. 2. The critical path of the 16×16-bit Non-Booth parallel multiplier.

implemented with a sparse-2 carry tree is used [9]. A 32-bit register is used to feed the current sum to the input of the adder circuitry over an AND operation. The RST input is designed to reset internal residue of the circuit before the start of a new multiplication operation. A 32-bit shift register is used to generate the *pseudo*-partial products (Each time make a left shift and feeds a ZERO for 16-bit *multiplicand* input). The real partial product is generated after ANDing the *pseudo*-partial products with the corresponding bit of the *multiplier* input. The final output is generated in 16 clock cycles for a 16×16-bit multiplier since there are 16 partial products that need to be summed up.

B. Single Cycle Parallel Multiplier

Serial multipliers offer the advantage of lower hardware cost at the expense of the low throughput. For the demanded applications the parallel multipliers are widely used to satisfy the throughput requirement. The Non-Booth multipliers implemented with TDM approach is selected as a representative of the parallel multipliers. This structure outperforms the other parallel structures in the strict delay targets [10]. The critical path of the 16×16-bit Non-Booth multiplier is shown in Fig. 2. The depth is 21 logic stages and the number of logic elements is 1794 as shown in Fig. 2.

The inversion stage in the partial product generation block is merged to the partial product reduction stage to reduce the number of stages sitting on the critical path [10]. There are many different implementation styles for the 3:2 compressors that are used to reduce the generated partial products to the rows of carry and sum signals. The schematic of the used 3:2 compressor is given in Fig. 3. The slow inputs are connected to fast outputs and the fast inputs are connected to the slow outputs to take advantage of the delay difference between the carry and the sum outputs.

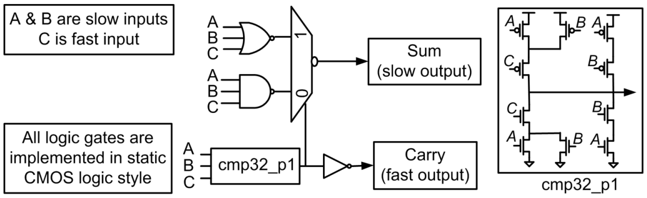
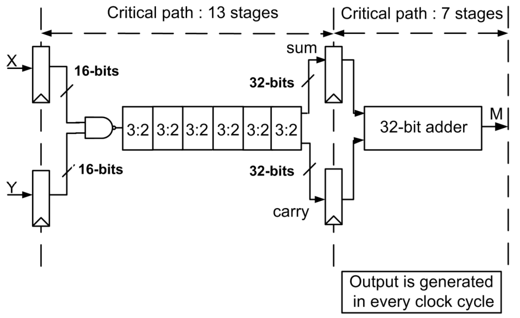


Fig. 3. Schematic of the 3:2 compressors used in the implementation of parallel multipliers. It consists of two logic stages for both outputs.



aoi21	23
aoi22	16
dff	64
inv	307
mux21	158
nand2	464
nand3	171
nor2	39
nor3	111
oai21	27
oai22	15
xnor2	81
xor2	250
cmp32_p1	175
TOTAL : 1901	
Logic depth : 13	

Fig. 4. The block diagram of the 16×16-bit two-cycle Non-Booth parallel multiplier (conventional approach).

The final adder of the single cycle multiplier is implemented by taking advantage of the signal arrival profile of the sum and the carry signals. The signals being in the middle bit positions come later than the signals at end bit positions [7]. The final adder of the single cycle multiplier is implemented as follows; ripple carry adder for the first four bits (<3:0>), 24-bit Ling adder implemented with a sparse-2 carry tree, conditional ripple carry adder for the last 4-bits. (The compressor shown in Fig. 3 is used in ripple carry adders). This partitioning provides the lowest energy at the same performance as compared to other possible scenarios.

C. Two Cycle Parallel Multiplier (Conventional Design)

The critical path of the single cycle multiplier consists of 21 logic stages. This restricts the operating frequency of the overall system. In order to improve the throughput of the parallel multiplier, the multiplier circuit is divided into two cycles. The conventional way of the pipelined multiplier implementation is the separating the circuit at the end of the partial product reduction stage. At the end of the partial product reduction block, there are two 32-bit outputs namely carry and sum signals. These signals will be fed to the final adder to generate the final output. The block diagram of the conventional two-cycle parallel multiplier is given in Fig. 4. The critical paths of each stage become 13 and 7 respectively. 32-bit Ling adder with a sparse-2 carry tree is used to implement the final addition [9].

D. Two Cycle Parallel Multiplier (Proposed Design)

The critical paths of the first and the second pipeline stages are unbalanced in the conventional design of the

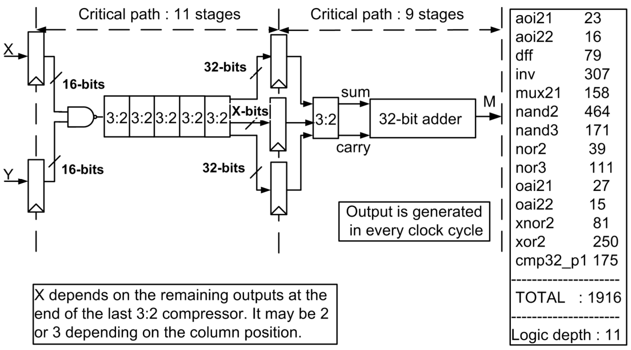


Fig. 5. The block diagram of the 16×16-bit two-cycle Non-Booth parallel multiplier (proposed approach).

pipelined parallel multiplier as shown in Fig. 4. One level of the 3:2 compressors is moved from first pipeline stage to second one to balance critical paths of pipeline stages. The same circuit structures are used as in the conventional design. The critical paths become 11 and 9 for the first and the second pipeline stages respectively as shown in Fig. 5. Hence, a better throughput is obtained from the proposed multiplier. However, the number of registers will be higher and it will increase the hardware cost. The gate sizes in the first pipeline stages will be reduced since the delay target becomes more relaxed for the first pipeline stage and this provides energy reduction. There is a trade-off between the energy saving obtained from the first pipeline stage and the energy cost for the extra registers and the gate size increase in the second pipeline stage. The best design is expected to happen when the critical paths of the first and the second pipeline stages are equal if there is no functional and clock storage element limitations. However, extra registers are required when a level of 3:2 compressors is moved from first pipeline stage to second one. Therefore, the first pipeline stage will have more complex critical path than the second stage when the pipelining is implemented at the optimum place. Also, the proposed design has less glitch since it has more hardware in the second pipeline stage.

III. RESULTS

16×16-bit serial multiplier, single cycle Non-Booth parallel multiplier, two-cycle Non-Booth parallel multipliers are implemented in 45nm CMOS. Inputs have a drive strength equivalent to *30-minimum sized inverters.* Outputs are loaded with *30-minimum sized inverters (or 180-minimum sized inverters for heavily loaded case)*. The circuit sizing optimization is an iterative process and implemented in MATLAB. The best possible delays are obtained using Logical Effort sizing for each design. Then, the minimum energy solution is found by using all minimum sized gates. The delay range for each design is determined based on the Logical Effort delay points and the all minimum sized delay points. Then each design is sized for the minimum energy under the pre-determined delay targets. Then, the delay range is extended towards larger values to explore a bigger space. The energy and delay estimation method proposed in [8] is used to quickly estimate the energy and delay for every sizing solution. In order to make an accurate estimation, the technology characterization is performed. A summary of the technology characterization is given in Table-I. When the voltage supply is reduced from its nominal value of 1.1V, some of the technology characterization parameters will also change. A summary of the timing parameters at reduced voltage supplies are given in Table-II.

TABLE I. TECHNOLOGY CHARACTERIZATION SUMMARY

Technology	45nm
Voltage Supply [V]	1.1
[a]W_{min}	100nm
Tau [pS]	4.1
FO4 [pS]	22.54
[b]INV Input Capacitance [fF]	0.381
Wire Capacitance [fF/um]	0.2
Bit Picth [um]	2
Temperature [^{o}C]	25

a. Minimum sized transistor width in 45nm CMOS

b. Input capacitance of minimum sized inverter

TABLE II. TIMING PARAMETERS AT REDUCED VOLTAGES

Voltage Supply [V]	1.1	1.0	0.9	0.8	0.7	0.6	0.5
Tau [pS]	4.1	4.6	5.5	7.0	10.0	17.2	42.4
FO4 [pS]	22.54	25.35	30.46	38.86	55.50	95.52	237.1

At nominal voltage supply, all designs are optimized and the energy versus the number of multiplications per second is given in Fig. 6. The throughput is directly proportional to *1/Delay* and the throughput will be a better metric since the serial multiplier is not generating outputs at every clock cycles. For the application that requires high throughput, the proposed two cycle parallel multiplier is the best structure at the nominal voltage supply for lightly loaded case. As the throughput requirement is reduced, the serial multiplier starts to become a more energy-efficient structure. Notice that every structure reaches a minimal energy point that corresponds to all minimum sized solution. After this point, if the operating frequency of the multiplier is relaxed further, the energy will start to increase because of the increase in the leakage energy. The optimum multiplier structure is also found for the heavily loaded designs. For this purpose, the path gain is increased to 6 by increasing load to *180-minimum sized inverters*. The results are provided in Fig. 7. As shown from the figure, the proposed structure is still the most energy-efficient one at high throughput design regimes.

Voltage supply is another design parameter used to explore the energy-delay trade-offs in digital circuits. As an example, fast structures can be slowed down by reducing the voltage supply. The energy and delay trade-offs coming from the voltage scaling are explored for the selected multiplier structures. When the voltage supply is scaled to 0.5V, the pipelined structures start to provide more throughput with the lowest possible energy consumption as shown in Fig. 8. The proposed two-cycle multiplier provides 13.5 times more throughput at the lowest possible energy of 352fJ as compared to the serial multiplier.

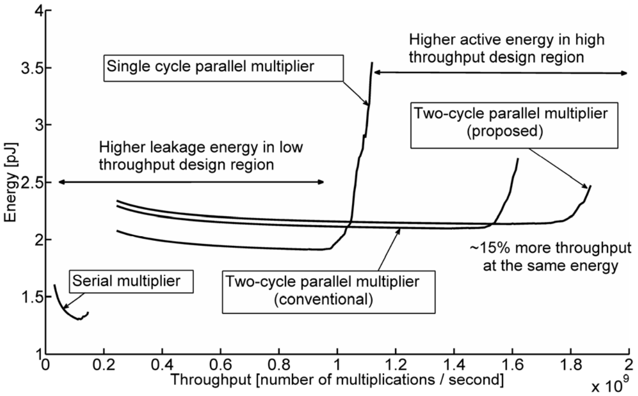


Fig. 6. Energy versus throughput at 1.1V voltage supply in 45nm CMOS (output load is 30*min_sized inverter and path gain is 1).

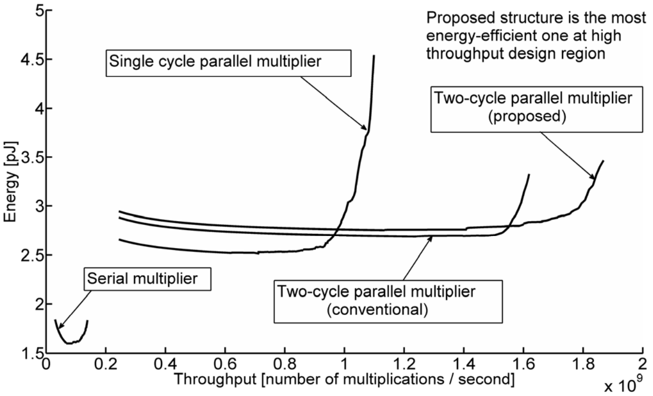


Fig. 7. Energy versus throughput at 1.1V voltage supply in 45nm CMOS (output load is 180*min_sized inverter and path gain is 6).

When all gates are minimum sized, each design reaches to a minimum in energy consumption. The delay cannot be reduced further by optimizing gate sizes. The further delay relaxation is obtained from the voltage scaling. For this purpose, voltage is scaled when all minimum sized gates are used in each design and the output is loaded with *30-minimum sized inverters*. The results are given in Fig. 9. The proposed multiplier provides the same throughput as the serial multiplier in the low throughput applications with considerably lower energy consumption. As an example, the proposed multiplier provides 1.26×10^8 multiplications per second with 352fJ (3.7X energy reduction) at a voltage supply of 0.5V. The same throughput is possible with the serial multiplier operating at the 1.1V and consumes 1306fJ energy.

IV. Conclusion

The energy-throughput trade-offs coming from the circuit sizing, voltage scaling and the structural optimizations are explored for multiplier circuits. It is shown that the parallel multipliers will be more energy-efficient than the serial multiplier in low throughput design regimes by optimizing voltage supply and the gate sizes. A two-cycle Non-Booth parallel multiplier that is obtained by balancing the complexities of pipeline stages is proposed. The proposed design outperforms the parallel structures and it also provides up to 3.7X energy reduction for ultra-low power applications compared to low cost serial multiplier.

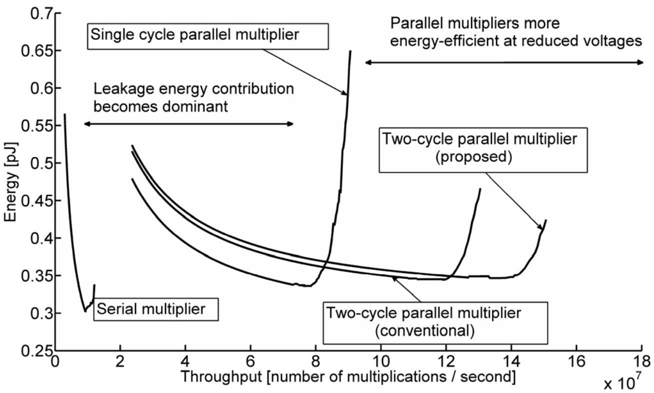


Fig. 8. Energy versus throughput at 0.5V voltage supply in 45nm CMOS (output load is 30*min_sized inverter and path gain is 1).

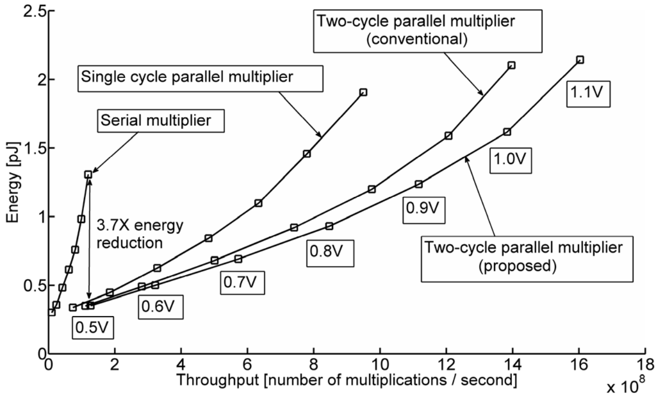


Fig. 9. All minimum sized designs at reduced voltages in 45nm CMOS (output load is 30*min_sized inverter and path gain is 1).

References

[1] A. D. Booth, "A Signed Binary Multiplication Technique", *Quarterly J. Mechanical Applications in Math*., vol. 4, part 2, pp. 236-240, 1951.

[2] Q. L. Macsorley, "High Speed Arithmetic in Binary Computers", *IRE Proc.*, vol. 49, pp. 67-91, Jan. 1961.

[3] C. S. Wallace, "A Suggestion for a Fast Multiplier", *IEEE Trans. Computers*, vol. 13, no. 2, pp. 14-17, Feb. 1964.

[4] L. Dadda, "Some schemes for Parallel Multipliers", *Alta Frequenza*, vol. 34, pp. 349-356, Mar. 1965.

[5] B.Parhami, *Computer arithmetic algorithms and hardware designs*, Oxford Univ. Press, 2000.

[6] V. G. Oklobdzija, D. Villeger and S.S. Liu, "A Method for Speed Optimized Partial Product Reduction and Generation of Fast Parallel Multipliers Using an Algorithmic Approach", *IEEE Trans. Computers*, vol. 45, no. 3, pp. 294-306, Mar. 1996.

[7] V. G. Oklobdzija and D. Villeger, "Improving Multiplier Design Using Improved Column Compression Tree and Optimized Final Adder in CMOS Technology", *IEEE Trans. VLSI*, vol. 3, no. 2, pp. 292-301, June 1995.

[8] Oklobdzija, V.G.; Zeydel, B.R.; Dao, H.Q.; Mathew, S.; Krishnamurthy, R.; , "Comparison of high-performance VLSI adders in the energy-delay space," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* , vol.13, no.6, pp. 754- 758, June 2005.

[9] Zeydel, B.R.; Baran, D.; Oklobdzija, V.G.; , "Energy-Efficient Design Methodologies: High-Performance VLSI Adders," *Solid-State Circuits, IEEE Journal of* , vol.45, no.6, pp.1220-1233, June 2010.

[10] Baran, D; Aktan, M; Oklobdzija, V.G.; , "Energy-Efficient Implementation of Parallel CMOS Multipliers with Improved Compressors," 16th *ACM/IEEE International Symposium on Low-power Electronics and Design* , pp. 147-152 , August 2010.

[11] D. Villeger and V. G. Oklobdzija, "Evaluation Of Booth Encoding Techniques For Parallel Multiplier Implementation", *Electronics Letters*, Vol. 29, No. 23, pp. 2016-2017, 1993.