Dual-Channel Multiplier for Piecewise-Polynomial Function Evaluation for Low-Power 3-D Graphics

Dina M. Ellaithy, Magdy A. El-Moursy, Amal Zaki, and Abdelhalim Zekry

***Abstract*— A dual-channel multiplier (DCM) for energy efficient second-order piecewise-polynomial function evaluation for 3-D graphics applications is presented in this paper. The performance of the evaluation process is highly dependent on the design of the multiplication and squaring structure. A novel hardware implementation for polynomial evaluation is presented. The proposed approach compensates the complex multipliers by using DCM which reduces the hardware complexity. The DCM scheme performs complex functions with power-efficient and area-efficient approach. The multiplier reduces the hardware computational effort in the piecewise polynomial approximation with uniform or nonuniform segmentation. For large operand input size, a multiplier adder converter and a dedicated radix-4 squaring unit are also proposed. These units achieve the least power consumption compared to previous approaches with large input word size. Comparison with general purpose multiplication has shown reduction in power, and delay by up to 36%, and 50%, respectively. The proposed technique exhibits up to 93% saving in power consumption compared to the current traditional schemes.**

***Index Terms*— Dual channel multiplier (DCM), graphical processing unit (GPU), low power, multiplier adder converter (MAC), piecewuise-polynomial evaluation, radix-4 squaring unit.**

I. Introduction

GRAPHICAL processing units (GPUs) have a wide variety of applications in different fields such as compute art, engineering, science, medicine, entertainment, advertising, visualization, military, and graphical user interface. The growth in the GPUs applications has led to evolution in the hardware design to handle the needed increase in performance [1]. The general purpose GPUs are used to perform intensive computations on the same hardware covering large sector of applications. GPUs are mainly composed of several unified shaders. Each unified shader contains general purpose arithmetic unit and special function unit (SFU) that is used for computing special transcendental and algebraic functions not provided by the general function unit [1]. Functions such as sine, cosine, reciprocal, logarithm, exponential, and compound functions are computed by SFUs.

The main design constraint in the handheld devices is power dissipation. To extend the battery lifetime, low-power schemes are essential at all stages of design. SFUs handle the complex operations which are required for graphics applications [2]. Most of the power is consumed by those heavy arithmetic functions [3]. Three main algorithms have been adopted to evaluate various functions in hardware including direct lookup table algorithm, polynomial and rational algorithm, and table-based piecewise-polynomial (PWP) algorithm [2], [4]–[17]. Large lookup tables are used to store the approximated value of the function in the direct lookup algorithm. The direct lookup table method is appropriate for low accuracy function evaluation due to the exponential increase in area with the increase in the input size [2], [4]. There is tradeoff between the lookup table size and the hardware cost. Fast execution can be obtained with smaller hardware. Low lookup table size is exploited in the polynomial and rational algorithm but with excessive hardware complexity [5]. High-degree polynomial is used in this algorithm to approximate the function. As a result, large number of multiplications and additions leads to high power consumption and long execution times. As a compromise between the direct lookup table algorithm and the polynomial algorithm, the table-based PWP is employed [6]–[18]. PWP has low overhead and small table size for hardware computation which makes that algorithm the most attractive for function evaluation. The approximation coefficients of the polynomial are stored in small lookup table. PWP algorithm gives attractive tradeoff between the computation error and hardware cost. Several PWP evaluation techniques are proposed in literature.

The most hardware expensive part of the PWP evaluation architecture is the multiplier. Focusing on low power, dual-channel multiplier (DCM) with low-cost hardware for efficient GPU is proposed in this paper as shown in Fig. 1. Compared to the traditional PWP evaluation techniques, the proposed DCM technique has improved power-delay-product (PDP) with compact area. Furthermore, for large input operand size, the main advantage of logarithmic number system (LNS) is exploited. Multiplication is reduced to addition using logarithmic conversion in this paper. The multiplier-free architecture achieves low-power dissipation with small area and low conversion error. After a brief background of previous work in Section II, the DCM scheme is explained in Section III combined with the quadratic PWP evaluation. In Section IV,

Manuscript received March 19, 2018; revised August 6, 2018 and October 8, 2018; accepted November 21, 2018. *(Corresponding author: Magdy A. El-Moursy.)*

D. M. Ellaithy is with the Microelectronics Department, Electronics Research Institute, Cairo 12622, Egypt (e-mail: dina_elessy@eri.sci.eg).

M. A. El-Moursy is with the Microelectronics Department, Electronics Research Institute, Cairo 12622, Egypt, and also with the Integrated Circuits Verification Systems Division, Mentor, A Siemens Business, Cairo 12622, Egypt (e-mail: magdy_el-moursy@mentor.com).

A. Zaki, deceased, was with the Microelectronics Department, Electronics Research Institute, Cairo 12622, Egypt.

A. Zekry is with the Electronics and Communications Department, Ain Shams University, Cairo, Egypt (e-mail: aaazekry@hotmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2018.2889769

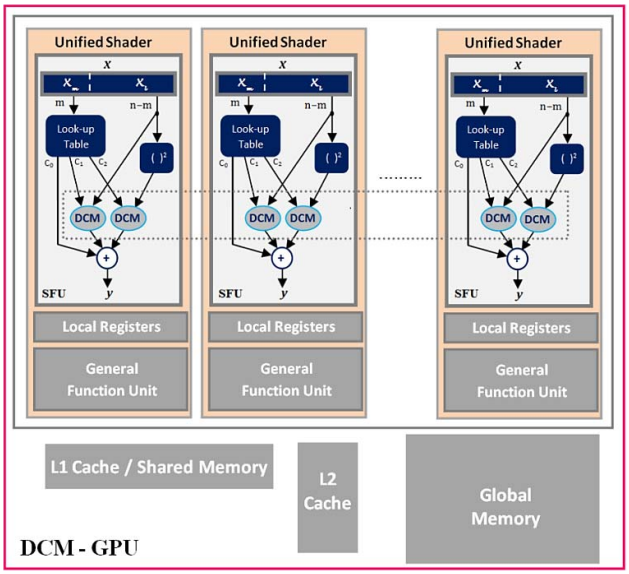

Fig. 1. Core structure of the DCM-GPU.

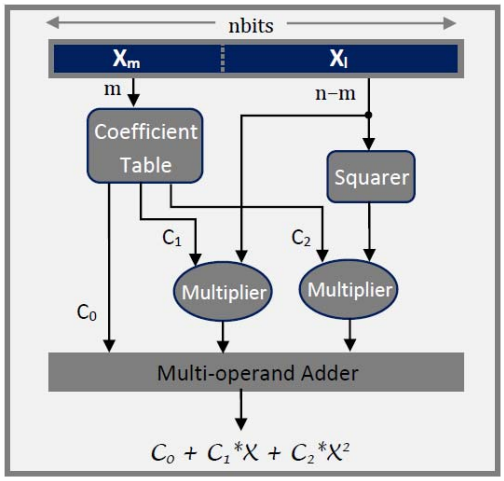

Fig. 2. Hardware architecture of the conventional quadratic PWP function evaluation.

a novel architecture of the quadratic PWP evaluation with radix-4 squaring unit is presented. Hardware implementation and performance evaluation of the proposed techniques beside comparison with earlier work are presented in Section V. Finally, some conclusions are obtained in Section VI.

II. Background

A good tradeoff between the table size and the hardware complexity is given by the PWP function evaluation approach. This approach has been widely used [6]–[18]. In PWP, the interval for the computed function is divided into subsegments. An appropriate low degree polynomial is employed to approximate the computed function in each subsegment. A small size lookup table is used to store the approximation coefficients for each subsegment. The divided subsegments of the PWP function evaluation can be uniformly or nonuniformly distributed. The address count for the approximation coefficients becomes simpler when the evaluation range is equally divided [6]–[10]. Also, uniform segmentation requires low-cost hardware implementation compared to nonuniform segmentation which requires more hardware for address remapping [11], [12]. However, nonuniform segmentation can be more efficient with functions that have high nonlinearity [13].

The hardware architecture of the quadratic PWP function evaluation is shown in Fig. 2. The input operand with n bits is split into two portions. The most significant bits m are used to select the approximation coefficients of each segment. The total number of segments equals 2^m. The function evaluation is approximated by the least significant bits $(n - m)$ which are the second portion of the input operand bits. The approximation coefficients C_0, C_1, and C_2 are determined for each segment using mathematical algorithms. Minimax and Chebyshev are the two primary algorithms that are exploited to determine the approximation coefficients of the quadratic PWP evaluation. Approximations nearby the optimal least maximum approximation are determined by the Chebyshev algorithm [10], [18]. Unlike Chebyshev algorithm, the Minimax algorithm provides better approximation [6]–[9]. Therefore, the Minimax algorithm is the most popular approximation algorithm. Large number of researches adopted the Minimax algorithm for better approximation [6]–[9], [11], [12], [16], [17]. However, Chebyshev is used to speed up the coefficients generation process [10].

The approximation coefficients of the piecewise quadratic polynomial are chosen in order to optimize the overall performance at the expense of precision. Researches have been carried out to simplify the hardware realization of the PWP evaluation [6]–[10], [14], [16]. However, these techniques are only concentrated on the optimization of the coefficients, overlooking the cost of the hardware.

The function evaluation data path as shown in Fig. 2 contains two multipliers. Low-power and low-cost GPUs with enough processing speed are needed in handheld devices. This brings the need of embedding low power multiplier into the PWP. Most of the previous work focuses on speeding up the multiplication process in order to have high performance [6], [12], [16]. However, the requirements in GPU architecture for portable applications have dictated the need for low-power designs. This paper presents different design methods for low power and area efficient PWP function evaluation for low-power 3-D graphics applications. The serial bit schemes which handle one input bit in one clock cycle are well-known with their simplicity and low hardware cost. They are ideal for low-power applications. These characteristics of serial schemes have been exploited in the proposed DCM in this paper. In the Section III, the proposed DCM scheme for cutting down power consumption and area is described.

As the input size grows, the approximated coefficients as well as the hardware complexity and the power consumption are increased incredibly in the conventional schemes. Large input operand size parallel multiplier has quite complex structure. Therefore, for large operand size, the precision requirements have to be shrunk [10], [17], [18]. In this paper,

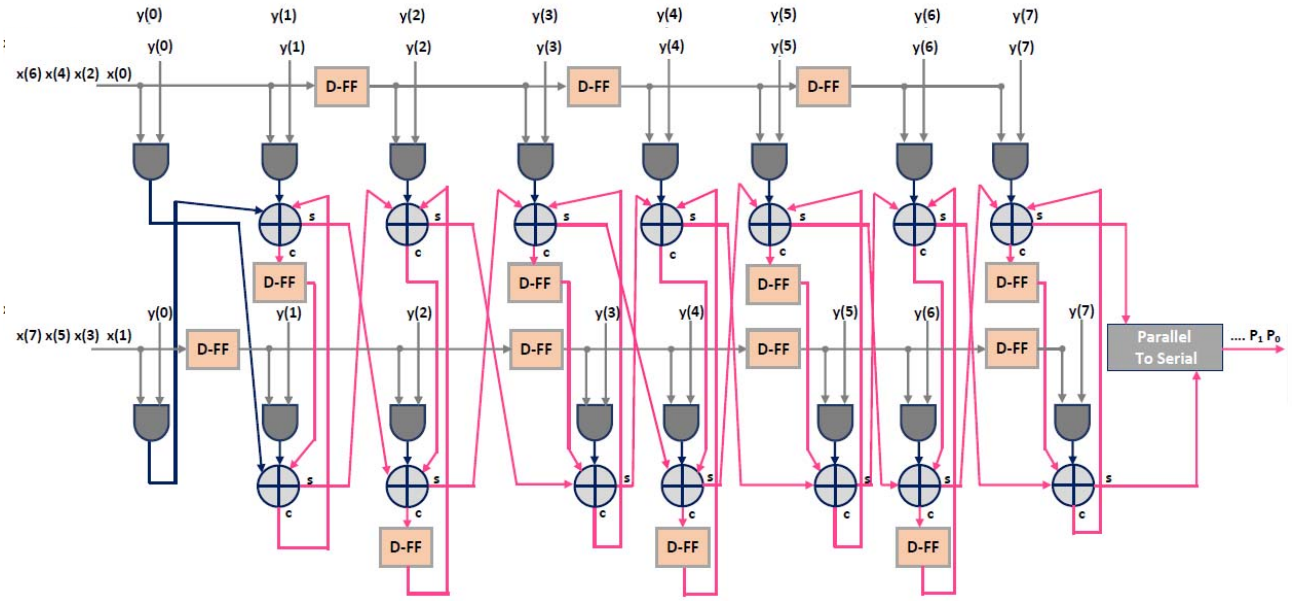

Fig. 3. Block diagram of the proposed 8 bit DCM.

different hardware architectures are adopted for large input size and efficient power saving with higher speed multiplier-free architecture. The LNS is used to design a low-power and high-speed piecewise quadratic polynomial evaluation. Reduction of the degree of the operation is achieved by reducing multiplication into addition [19]–[26]. The cutting down in the switching activity leads to reduction in the power dissipation [19]–[27]. In this paper, the multiplier-adder-converter (MAC) scheme is introduced and is compared with recent techniques in terms of power, area, and delay. Since the proposed scheme is generic, it can be used in combination with other methods for improving the performance of the PWP evaluation. The MAC scheme for efficient power consumption and high-speed function-evaluation is presented in Section IV.

III. Proposed Dual-Channel-Multiplier Scheme

It is the age of the handheld devices which directs all attempts to cut down the amount of power and to shrink the area as much as possible. Both of these requirements fit with the features of serial algorithm [28]–[34]. The straightforwardness of the key cell of serial multipliers makes its implementation perfect for VLSI. The hardware of such multipliers can simply be extended by repeating one cell. The complete structure of the quadratic PWP contains two main multipliers which are typically implemented by booth multipliers with a different radix [12], [16]. In Section III-A, low-power and low-area DCM architecture is proposed based on a serial algorithm.

A. DCM Architecture

The proposed DCM architecture is shown in Fig. 3. The DCM scheme has simple and uniform construction. x and y are the serial input and the parallel input, respectively, and p is the final product. Two serial input bits are processed each clock cycle. Even index numbers (x_6, x_4, x_2, x_0) of the serial input are directed to the upper channel while, odd index numbers (x_7, x_5, x_3, x_1) are processed in the lower channel. Concurrently, the pairs are transferred and handled on the same clock phase. Starting with the first clock cycle, the partial products (PP_0) are generated

$$PP_0 = \begin{cases} y_0x_0, & y_1x_0 \\ y_0x_1. \end{cases} \tag{1}$$

The partial product (y_0x_1) is added to the partial product (y_1x_0) and propagated to the output. Also, the partial product (y_0x_0) is directly propagated to the output. The least significant two bits of the product (P_0 and P_1) are generated simultaneously in the following equations:

$$P_0 = y_0x_0 \tag{2}$$

$$P_1 = y_0x_1 + y_1x_0. \tag{3}$$

In the next clock cycle, the serial input data have been moved one phase to the right after each delay element.

Note that the generated carry from addition remains to be added for the exact final result. The current partial products which are generated for the upper channel are (y_0x_2), (y_1x_2), (y_2x_0), and (y_3x_0) as the following equation. For the lower channel, the partial products (y_0x_3), (y_1x_1), and (y_2x_1) are produced as in the following equation:

$$PP_1 = \begin{cases} y_0x_2, & y_1x_2, & y_2x_0, & y_3x_0 \\ y_0x_3, & y_1x_1, & y_2x_1. \end{cases} \tag{4}$$

Starting from the partial product (y_0x_2) in the upper channel, two complete additions are carried out. The partial product (y_0x_2) is added to (y_1x_1) then is added to (y_2x_0) and the final result becomes (P_2). Three complete additions are performed starting from the partial product (y_0x_3) in the lower channel. The partial product (y_0x_3) is added to (y_1x_2) then is added to (y_2x_1) and at last is added to (y_3x_0).

								y_7	y_6	y_5	y_4	y_3	y_2	y_1	y_0
								x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
								y_7x_0	y_6x_0	y_5x_0	y_4x_0	y_3x_0	y_2x_0	y_1x_0	y_0x_0
							y_7x_1	y_6x_1	y_5x_1	y_4x_1	y_3x_1	y_2x_1	y_1x_1	y_0x_1	
						y_7x_2	y_6x_2	y_5x_2	y_4x_2	y_3x_2	y_2x_2	y_1x_2	y_0x_2		
					y_7x_3	y_6x_3	y_5x_3	y_4x_3	y_3x_3	y_2x_3	y_1x_3	y_0x_3			
				y_7x_4	y_6x_4	y_5x_4	y_4x_4	y_3x_4	y_2x_4	y_1x_4	y_0x_4				
			y_7x_5	y_6x_5	y_5x_5	y_4x_5	y_3x_5	y_2x_5	y_1x_5	y_0x_5					
		y_7x_6	y_6x_6	y_5x_6	y_4x_6	y_3x_6	y_2x_6	y_1x_6	y_0x_6						
	y_7x_7	y_6x_7	y_5x_7	y_4x_7	y_3x_7	y_2x_7	y_1x_7	y_0x_7							
p_{15}	p_{14}	p_{13}	p_{12}	p_{11}	p_{10}	p_9	p_8	p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

Fig. 4. Partial product of multiplying two 8 bit.

The final result of the previous sum is (P_3). The carry bits which are produced by the full adders are propagated forward and added to the next partial products. The multiplication process is repetitive

$$P_2 = y_0x_2 + y_1x_1 + y_2x_0 \tag{5}$$

$$P_3 = y_0x_3 + y_1x_2 + y_2x_1 + y_3x_0. \tag{6}$$

The next partial products are generated by the same procedure. After eight clock cycles, the product of 8-bit by 8-bit multiplication is achieved. The partial product architecture for 8-bit by 8-bit multiplication is shown in Fig. 4. A 1-bit multiplication is performed by an AND gate and a 1-bit addition is performed by a full adder. Unlike the traditional twin serial/parallel multipliers [30]–[34], double throughput is achieved in DCM with minimum data latches. Hence, the multiplier requires n (number of input bits) clock cycles to complete the multiplication process. The DCM includes parallel-to-serial and serial-to-parallel converters to handle the input and to compute the parallel result, respectively. For large word size, additional cells can be added in the hardware implementation to perform the multiplication. Each cell includes two AND gates, two full adders, and 2-D flip flops. Compared to different previous architectures [30]–[34], DCM uses less hardware. Also, the proposed DCM includes only full adders and delay units. This makes the proposed architecture consumes significant lower power than traditional architecture with booth multiplier. The 8-bit and 16-bit DCM are implemented in 90-nm CMOS process in Section V to demonstrate the efficiency of the architecture.

B. DCM With Quadratic Piecewise Polynomial Function Evaluation

In this paper, traditional parallel multipliers are replaced with DCM scheme in the PWP function evaluation architecture as shown in Fig. 5. The control unit is responsible for determining the start and the end of the multiplication process and also for managing the function evaluation. The hardware architecture of the quadratic PWP evaluation is simplified using DCM. In the performance evaluation Section V, great savings in power and area are achieved using DCM.

As the input operand size increases, the size of the hardware of the quadratic PWP function evaluation grows. A multiplier adder converter (MAC) is proposed in Section IV to overcome the increase in hardware cost for large operand input size.

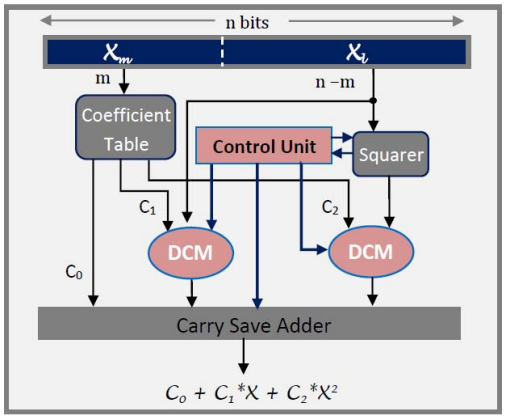

Fig. 5. Architecture of the DCM for the quadratic PWP function evaluation.

IV. Proposed Multiplier–Adder–Converter Quadratic PWP Function Evaluation

In order to reduce the hardware cost, truncated multipliers have been utilized to decrease the number of formed partial products. The least significant lines of the partial products are not generated. Hence, the number of the utilized adders and logic cells are reduced. Consequently, savings in the overall area and power consumption using the truncated multipliers are achieved. However, truncated multipliers complicate the error analysis and decrease the accuracy. It is useful to reduce the hardware implementation by replacing the multiplication with addition. Consequently, faster and more power efficient systems are acquired. However, additions and subtractions become more complex operations in LNS. The main limitation of LNS is that not all basic operations can be performed within the logarithmic field. Thus, PWP approximations are utilized for the nonlinear operations in the logarithmic domain [35]–[38]. On the other hand, tasks are switched and logarithmic domain can be involved in the PWP function evaluation architecture. LNS is adopted in MAC and is used in the evaluation of quadratic PWP structure design.

The proposed scheme is designed and simulated. Power saving of MAC is achieved by completely suppressing multiplication process. The structure of MAC is shown in Fig. 6. The key blocks of MAC are the logarithmic and antilogarithmic converters. The inputs go directly to the logarithmic converters then the addition is carried out. The obtained result is converted from the logarithmic domain using antilogarithmic converter. Large reductions in power consumption and area are obtained as a result of decreasing the process degree from multiplication to addition.

A. Proposed MAC

The MAC is composed of two logarithmic converters, an adder, and an antilogarithmic converter as shown in Fig. 6. The implementation of the logarithmic and antilogarithmic converters controls the MAC performance. Several works have been proposed to optimize the performance of the transformation [19]–[23]. To take full advantage of LNS, logarithmic and antilogarithmic converters are implemented using the double

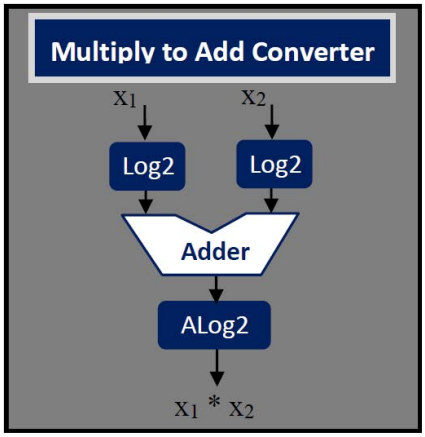

Fig. 6. Proposed MAC scheme.

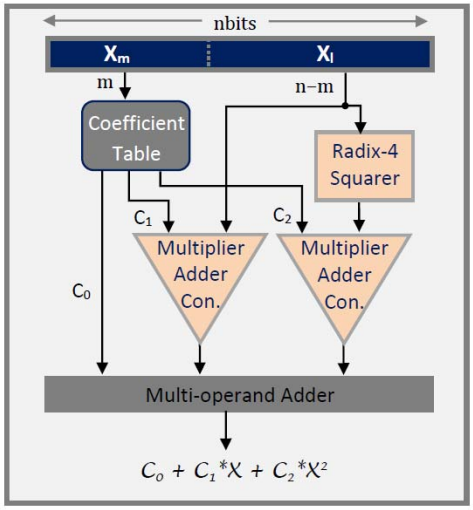

Fig. 7. Quadratic PWP block diagram with the proposed MAC.

logarithmic technique [19]. Lower error ranges with saving in power and area are provided by the proposed transformation. The converters are employed in the MAC implementation. The addition is implemented using the basic carry propagate adder (CPA) for efficient power saving. For speeding up the addition, carry look ahead adder can be used. The architecture of the proposed quadratic PWP function evaluation is presented in Section IV-B.

B. Quadratic PWP Function Evaluation With MAC

A novel design of second-degree PWP function evaluation is provided in this section. The traditional multiplier technique is replaced with the multiplier-free MAC scheme in the PWP approximation architecture as shown in Fig. 7. For large operand input size, MAC directly handles the limitations of hardware cost unlike the conventional second-degree PWP function evaluation which requires high implementation cost.

In addition, a radix-4 squarer is proposed. As shown in the PWP function evaluation structure, the squarer is located in the critical path. Therefore, to speed up the overall performance, the proposed radix-4 squarer can be realized by operating on each two bits simultaneously. The number of partial products is reduced to half. Instead of multiplying by 0 or 1, multiplying by 0, +1, +2, or +3 is performed in radix-4. The proposed squarer exhibits higher speed than the traditional squarer that is utilized in PWP [39]. Hardware implementation and performance results for the proposed schemes are presented and compared with the existing techniques in Section V.

TABLE I
EIGHT-bit MULTIPLIER, 90-nm CMOS IMPLEMENTATION AT 200 MHz

Scheme	Power (mW)	Area (μm^2)	Delay (ns)	Power-Delay Product (pJ)
Array Multiplier	0.27089	1456.67	2.39	0.64743
Booth radix-4 Multiplier	0.28537	1336.72	2.25	0.64208
Booth radix-8 Multiplier	0.30867	1564.08	1.80	0.55561
Wallace Multiplier	0.28560	1527.23	1.95	0.55692
Serial Multiplier	0.01590	646.80	9.92(0.62*16)	0.15773
Proposed DCM	0.01979	838.88	4.96(0.62*8)	0.09816

V. HARDWARE IMPLEMENTATION AND PERFORMANCE EVALUATION

The proposed and all prior schemes are implemented in VHDL. Then the implementation is compiled using Synopsys Design Compiler 90-nm CMOS technology, 1-V supply voltage standard cell library, operating at 200 MHz.

A. Different Multipliers Implementation

The implementation of the most popular five types of multipliers is compared with the proposed DCM approach. Different numbers of bits are used in the comparison. The results of 8-bits and 16-bits are reported in Tables I and II, respectively. The power, area, and delay circuit characteristics are included in columns 2–4, respectively. Combining both power and delay values provides the measure of energy [power-delay product (PDP)] which is listed in the last column. Note that the serial multiplier needs $2N$ clock cycles to achieve the multiplication while the DCM needs only N clock cycles. Hence, the delay of 8-bit serial multiplier equals to the delay of one clock cycle multiplies 16 as reported in the Table I. Similarly, the 8-bit DCM delay is 8-times the clock period delay. The 38% reduction in PDP is achieved compared to serial multiplier. From Tables I and II, the implementation cost (power, area, delay) for parallel multipliers is considerably increased as the input size N is doubled from 8 bit to 16 bit. Less power and area characterize DCM. Compared to the fully parallel multipliers, DCM provides cutting down in area and power by up to

TABLE II
SIXTEEN-bit MULTIPLIER, 90-nm CMOS IMPLEMENTATION AT 200 MHz

Scheme	Power (mW)	Area (μm^2)	Delay (ns)	Power-Delay Product (pJ)
Array Multiplier	1.40	8229.65	4.50	6.300
Booth radix-4 Multiplier	1.07	5311.60	4.13	4.419
Booth radix-8 Multiplier	1.14	6130.10	3.59	4.093
Wallace Multiplier	1.15	6773.76	3.24	3.726
Serial Multiplier	0.0298	1299.09	42.56(1.33*32)	1.267
Proposed DCM	0.0391	1716.96	26.24(1.64*16)	1.026

37% and 93%, respectively. On the other hand, one clock cycle is required for parallel multipliers to complete multiplication while eight clock cycles are required for the DCM. However, the PDP is improved by at least 80% in the proposed DCM. DCM is optimum for low-energy applications such as hand-held devices applications that require low power consumption. Thus, DCM is efficient in designing PWP function evaluation since it provides simple and energy efficient design structure.

It can be observed that for radix-8 booth multiplier, instead of multiplying by 0 or 1, multiplying by 0, ± 1, ± 2, ± 3, or ± 4 are performed. Hence, extra hardware is needed to handle the generation of $\pm 3^*$ Multiplicand which leads to large power consumption and area, although the reduction in partial products provides high-speed output. All additions for the radix-4 and radix-8 booth multipliers use Wallace adder tree to compress the partial products to compute the final results. In contrast, serial multiplier has a different performance due to its different structures. Huge reduction in power and area are achieved by the serial multiplier while completely sacrificing speed. The proposed DCM approach exhibits better tradeoff between area, power, and delay. Hence, compared to conventional different types of multipliers, the proposed DCM exhibits a cutting in energy requirements by at least 70%. The proposed DCM is used in the quadratic PWP architecture for high reduction in energy. The hardware implementation results of the second-degree PWP with DCM are presented in Section V-B with comparison with the conventional techniques.

B. Quadratic PWP With DCM Implementation

Quadratic PWP evaluation for the reciprocal and the $1/(1+x)$ functions are created. Different algorithms for obtaining and optimizing the approximation coefficients have been proposed [6]–[16]. For the $1/(1+x)$ function, the approximation coefficients of Qoutb *et al.* [12] and Pineiro *et al.* [16] are utilized to compare different interpolator structures. Uniform segmentation algorithm Pineiro *et al.* [16] uses radix-4 booth multiplier in the structure of the interpolator. For high speed, Qoutb *et al.* [12] have proposed a high radix booth multiplier which considerably increases the consumed power. By switching the standard arithmetic units of PWP function evaluation to the proposed DCM scheme, significant saving in hardware size is achieved and performance is improved. Synopsys Design Compiler using 90-nm CMOS technology, 1-V supply voltage standard cell library is used to synthesis the proposed quadratic PWP approximation design and all previous schemes. The implementation results for the various designs that evaluate the $1/(1+x)$ function are shown in Table III. Also, a comparison between the DCM structure and the optimized quadratic interpolator Sadeghian *et al.* [10] is performed using the same approximation coefficients for the 1/x function. The results are reported in Table IV.

When the DCM scheme is used in the second-degree interpolator for function evaluation, great reductions in hardware, power consumption, and area are achieved. For the $1/(1+x)$ function, the DCM achieves up to 85%, 46%, and 63% savings in power, area, and energy, respectively, compared to Pineiro *et al.* [16]. Comparison with Qoutb *et al.* [12]

TABLE III
QUADRATIC PWP FUNCTION EVALUATION FOR THE $1/(1+x)$ FUNCTION AT 200 MHz

Technique	Power (mW)	Area (μm^2)	Delay (ns)	Power-Delay Product (pJ)
Proposed Quadratic with DCM	0.08641	1712.26	7.68	0.664
Pineiro et al. [16] [Booth radix-4 Multiplier]	0.59707	3206.56	3.01	1.797
Reduction (%)	85.53%	46.60%	2.55 times increase	63.07%
Qoutb et al. [12] [Booth high radix Multiplier]	0.62331	3456.66	2.80	1.745
Reduction (%)	86.14%	50.46%	2.74 times increase	62%

TABLE IV
QUADRATIC PWP FUNCTION EVALUATION FOR THE $1/x$ FUNCTION AT 200 MHz

Technique	Power (mW)	Area (μm^2)	Delay (ns)	Power-Delay Product (pJ)
Proposed Quadratic with DCM	0.2012	3884.72	33	6.639
Masoud et al. [10]	2.4023	12604.37	4.37	10.498
Reduction (%)	91.60%	69.18%	7.55 times increase	36.76%

TABLE V
THIRTY-TWO-bit MAC MULTIPLIER COMPARISON AT 100 MHz

Scheme	Power (mW)	Area (μm^2)	Delay (ns)	Power-Delay Product (pJ)
Wallace Multiplier	2.50	27636	7.44	18.600
Booth radix-4 Multiplier	1.88	23105	8.34	15.679
Booth radix-8 Multiplier	2.09	24176	6.64	13.877
Proposed MAC	1.34	26869	3.7	4.958

TABLE VI
THIRTY-TWO-bit SQUARER COMPARISON AT 100 MHz

Scheme	Power (mW)	Area (μm^2)	Delay (ns)
Totadri et al. [39]	1.16	13005.78	8.98
Proposed radix-4 squarer	1.78	19927.71	6.86

exhibits a reduction by at least 86%, 50%, and 62% in power consumption, area, and PDP, respectively. Furthermore, 91% less power, 70% less area, and 37% less energy are achieved by DCM compared to Masoud *et al.* [10] for the function $1/x$.

C. Proposed MAC Associated With Quadratic PWP Implementation

The 32-bit MAC is also implemented and synthesized using Synopsys Design Compiler based on 90-nm CMOS technology, 1-V supply voltage standard cell library, at 100 MHz. From Fig. 6, multiplication is replaced with addition in logarithmic field. The addition is implemented using CPA to obtain the summing result that directly goes into the antilogarithmic converter. The output from the antilogarithmic converter is the multiplication result. The 9-region logarithmic converter with 8-region antilogarithmic converter is employed to give a simple hardware implementation [19]. Correspondingly, 32-bit radix-4 and radix-8 booth multipliers are implemented using the same technology for comparison. The hardware performance results are summarized in Table V. Area, power, delay, and PDP are listed in columns 2–5, respectively. From the results in Table V, a significant drop in power and delay is achieved by the proposed MAC which would be valuable for implementing PWP function evaluation with large input operand size. Moreover, a radix-4 squarer is implemented and compared to the conventional squarer that is usually utilized in PWP evaluation [39]. The comparison is presented in Table VI. Speeding up the squarer is the result of reducing the number of partial products using radix-4 algorithm. Hardware resources are decreased while the implementation is improved according to the power and delay values in Table VII. Savings by up to 36% in power, 44% in delay, and 64% in energy are achieved with MAC compared to booth radix-8 multiplier in Table V. Quadratic PWP function evaluation with MAC achieves reduction in power, delay, and energy by at least 36%, 40%, and 61% compared to the traditional architecture with radix-4 booth multiplier, respectively, with an increase in the approximation error by less than 0.04%.

TABLE VII
THIRTY-TWO-bit QUADRATIC PWP FUNCTION EVALUATION FOR THE $1/(1+x)$ FUNCTION AT 100 MHz

Technique	Power (mW)	Area (μm^2)	Delay (ns)	Power-Delay Product (pJ)
Prior work [Booth radix-4 Multiplier]	4.25	52768.69	9.41	39.992
Reduction (%)	36.00	-7.35	40.00	61.57
Prior work [Booth radix-8 Multiplier]	4.61	54584.43	8.38	38.631
Reduction (%)	41.00	-4.16	32.58	60.22
Proposed Quadratic with MAC	2.72	56955.25	5.65	15.368

VI. CONCLUSION

The implementation of PWP evaluation in the SFU of the GPUs can be highly improved by boosting the performance of multiplication and squaring unit which are the basic components in the evaluation process. Exploiting the serial/parallel algorithm, an energy efficient DCM is proposed in this paper. Comparisons with the well-known multiplication schemes have demonstrated savings in area, power, and energy by at least 46%, 85%, and 63%, respectively. Moreover, for large operand input size, MAC is utilized to replace the traditional parallel multiplier scheme. MAC is proposed to perform the computation of the overall approximated polynomial without the need for multiplication in SFUs. The proposed scheme can implement different functions using simple hardware design. Also, high-speed dedicated squaring unit is proposed. It can be readily applied to any number of bits. PWP function evaluation by MAC is implemented using 90-nm CMOS technology. Eliminating the need of multiplication in the evaluation process leads to reduction in power by 36%, and delay by 40%. These features make the proposed designs more suitable for GPU applications. Complex designs generally incur long delay in PWP function evaluation, so a simple structure is a good choice. The overall energy of the proposed PWP function evaluation is reduced by at least 61% compared to previous work.

REFERENCES

[1] Y.-J. Chen *et al.*, "A 130.3 mW 16-core mobile GPU with power-aware pixel approximation techniques," *IEEE J. Solid-State Circuits*, vol. 50, no. 9, pp. 2212–2223, Sep. 2015.

[2] S. F. Hsiao, P.-H. Wu, C.-S. Wen, and P. K. Meher, "Table size reduction methods for faithfully rounded lookup-table-based multiplierless function evaluation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 5, pp. 466–470, May 2015.

[3] H. Kim, B.-G. Nam, J.-H. Sohn, J.-H. Woo, and H.-J. Yoo, "A 231-MHz, 2.18-mW 32-bit logarithmic arithmetic unit for fixed-point 3-D graphics system," *IEEE J. Solid-State Circuits*, vol. 41, no. 11, pp. 2373–2381, Nov. 2006.

[4] S.-F. Hsiao, C.-S. Wen, Y.-H. Chen, and K.-C. Huang, "Hierarchical multipartite function evaluation," *IEEE Trans. Comput.*, vol. 66, no. 1, pp. 89–99, Jan. 2017.

[5] I. Koren and O. Zinaty, "Evaluating elementary functions in a numerical coprocessor based on rational approximations," *IEEE Trans. Comput.*, vol. 39, no. 8, pp. 1030–1037, Aug. 1990.

[6] D. De Caro, N. Petra, and A. G. M. Strollo, "High-performance special function unit for programmable 3-D graphics processors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 9, pp. 1968–1978, Sep. 2009.

[7] S. A. Tawfik and H. A. H. Fahmy, "Algorithmic truncation of minimax polynomial coefficients," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, p. 2427.

[8] G. Cao, H. Du, P. Wang, Q. Du, and J. Ding, "A piecewise cubic polynomial interpolation algorithm for approximating elementary function," in *Proc. 14th Int. Conf. Comput.-Aided Design Comput. Graph.*, Aug. 2015, pp. 57–64.

[9] A. G. M. Strollo, D. De Caro, and A. Petra, "Elementary functions hardware implementation using constrained piecewise-polynomial approximations," *IEEE Trans. Comput.*, vol. 60, no. 3, pp. 418–432, Mar. 2011.

[10] M. Sadeghian, J. E. Stine, and E. G. Walters, "Optimized linear, quadratic and cubic interpolators for elementary function hardware implementations," *Electronics*, vol. 5, no. 2, p. 17, Apr. 2016.

[11] S.-F. Hsiao, H.-J. Ko, Y.-L. Tseng, W.-L. Huang, S.-H. Lin, and C.-S. Wen, "Design of hardware function evaluators using low-overhead nonuniform segmentation with address remapping," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 5, pp. 875–886, May 2013.

[12] A.-E. G. Qoutb, A. M. El-Gunidy, M. F. Tolba, and M. A. El-Moursy, "High speed special function unit for graphics processing unit," in *Proc. 9th Int. Design Test Symp. (IDT)*, Dec. 2014, pp. 24–29.

[13] D.-U. Lee, R. C. C. Cheung, W. Luk, and J. D. Villasenor, "Hierarchical segmentation for hardware function evaluation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 1, pp. 103–116, Jan. 2009.

[14] D. De Caro, E. Napoli, D. Esposito, G. Castellano, N. Petra, and A. G. M. Strollo, "Minimizing coefficients wordlength for piecewise-polynomial hardware function evaluation with exact or faithful rounding," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 5, pp. 1187–1200, May 2017.

[15] S.-F. Hsiao, H.-J. Ko, and C.-S. Wen, "Two-level hardware function evaluation based on correction of normalized piecewise difference functions," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 5, pp. 292–296, May 2012.

[16] J.-A. Pineiro, S. F. Oberman, J.-M. Müller, and J. D. Bruguera, "High-speed function approximation using a minimax quadratic interpolator," *IEEE Trans. Comput.*, vol. 54, no. 3, pp. 304–318, Mar. 2005.

[17] D. De Caro, N. Petra, A. G. M. Strollo, F. Tessitore, and E. Napoli, "Fixed-width multipliers and multipliers-accumulators with min-max approximation error," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 9, pp. 2375–2388, Sep. 2013.

[18] E. G. Walters, III, "Linear and quadratic interpolators using truncated-matrix multipliers and squarers," *Computers*, vol. 4, no. 4, pp. 293–321, Nov. 2015.

[19] D. M. Ellaithy, M. A. El-Moursy, G. H. Ibrahim, A. Zaki, and A. Zekry, "Double logarithmic arithmetic technique for low-power 3-D graphics applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 7, pp. 2144–2152, Jul. 2017.

[20] C.-W. Liu, S.-H. Ou, K.-C. Chang, T.-C. Lin, and S.-K. Chen, "A low-error, cost-efficient design procedure for evaluating logarithms to be used in a logarithmic arithmetic processor," *IEEE Trans. Comput.*, vol. 65, no. 4, pp. 1158–1164, Apr. 2016.

[21] D. De Caro, M. Genovese, E. Napoli, N. Petra, and A. G. M. Strollo, "Accurate fixed-point logarithmic converter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 7, pp. 526–530, Jul. 2014.

[22] R. R. Selina, "VLSI implementation of piecewise approximated antilogarithmic converter," in *Proc. Int. Conf. Commun. Signal Process.*, Apr. 2013, pp. 763–766.

[23] C.-T. Kuo and T.-B. Juang, "Lower-error antilogarithmic converters using binary error searching schemes," *Int. J. Innov. Technol. Exploring Eng.*, vol. 3, no. 7, pp. 95–101, Dec. 2013.

[24] V. Mahalingam and N. Ranganathan, "Improving accuracy in Mitchell's logarithmic multiplication using operand decomposition," *IEEE Trans. Comput.*, vol. 55, no. 12, pp. 1523–1535, Dec. 2006.

[25] Z. Babić, A. Avramović, and P. Bulić, "An iterative logarithmic multiplier," *Microprocessors Microsyst.*, vol. 35, no. 1, pp. 23–33, Feb. 2011.

[26] S. E. Ahmed and M. B. Srinivas, "An improved logarithmic multiplier for media processing," *J. Signal Process. Syst.*, pp. 1–14, Mar. 2018.

[27] A. Avramović, Z. Babić, D. Raič, D. Strle, and P. Bulić, "An approximate logarithmic squaring circuit with error compensation for DSP applications," *Microelectron. J.*, vol. 45, no. 3, pp. 263–271, Mar. 2014.

[28] S. G. Smith and B. P. Denyer, "Advanced serial-data computation," *J. Parallel Distrib. Comput.*, vol. 5, no. 3, pp. 228–249, Jun. 1988.

[29] R. Gnanasekaran, "A fast serial-parallel binary multiplier," *IEEE Trans. Comput.*, vol. C-34, no. 8, pp. 741–744, Aug. 1985.

[30] R. C. North and W. H. Ku, "β-bit serial/parallel multipliers," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 2, no. 4, pp. 219–233, May 1991.

[31] P. Larsson-Edefors, "A 965-Mb/s 1.0-μm standard CMOS twin-pipe serial/parallel multiplier," *IEEE J. Solid-State Circuits*, vol. 31, no. 2, pp. 230–239, Feb. 1996.

[32] J. Valls, T. Sansaloni, M. M. Peiró, and E. Boemo, "Fast FPGA-based pipelined digit-serial/parallel multipliers," in *Proc. IEEE Int. Symp. Circuits Syst. VLSI*, May/Jun. 1999, pp. 482–485.

[33] Y.-N. Chang, J. H. Satyanarayana, and K. K. Parhi, "Low-power digit-serial multipliers," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 1997, pp. 2164–2167.

[34] A. Aggoun, A. Ashur, and M. K. Ibrahim, "Bit-level pipelined digit-serial multiplier," *Int. J. Electron.*, vol. 75, no. 6, pp. 1209–1219, May 1993.

[35] Y. Popoff, F. Scheidegger, M. Schaffner, M. Gautschi, F. K. Gürkaynak, and L. Benini, "High-efficiency logarithmic number unit design based on an improved cotransformation scheme," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2016, pp. 1387–1392.

[36] S. Z. M. Naziri, R. C. Ismail, and A. Y. M. Shakaff, "Implementation of LNS addition and subtraction function with co-transformation in positive and negative region: A comparative analysis," in *Proc. 3rd Int. Conf. Electron. Design (ICED)*, Aug. 2016, pp. 1–6.

[37] R. C. Ismail, S. A. Z. Murad, R. Hussin, and J. N. Coleman, "Improved subtraction function for logarithmic number system," *Procedia Eng.*, vol. 53, no. 1, pp. 387–392, Jan. 2013.

[38] P. D. Vouzis, S. Collange, and M. G. Arnold, "Cotransformation provides area and accuracy improvement in an HDL library for LNS subtraction," in *Proc. 10th Euromicro Conf. Digit. Syst. Design Archit., Methods Tools*, Aug. 2007, pp. 85–93.

[39] T. Jayashree and D. Basu, "On binary multiplication using the quarter square algorithm," *IEEE Trans. Comput.*, vol. C-25, no. 9, pp. 957–960, Sep. 1976.

Dina M. Ellaithy received the B.S. degree in electronics and communications engineering and the M.S. degree in electrical and electronics engineering from Ain Shams University, Cairo, Egypt, in 2005 and 2013, respectively, where she is currently working toward the Ph.D. degree in electrical and electronics engineering.

From 2007 to 2013, she was a Teaching Assistant with the Electronics and Communications Department, Modern Academy for Engineering and Technology, Cairo. Since 2014, she has been a Research Assistant with the Electronics Research Institute, Cairo. Her current research interests include low-power circuit design, low-power arithmetic, low-power design and implementation of 3-D graphics processors, analog integrated circuit design, and phase-locked loop design.

Magdy A. El-Moursy was born in Cairo, Egypt, in 1974. He received the B.S. degree (with honors) in electronics and communications engineering and the master's degree in computer networks from Cairo University, Cairo, in 1996 and 2000, respectively, and the master's and Ph.D. degrees in electrical engineering in the area of high-performance VLSI/IC design from the University of Rochester, Rochester, NY, USA, in 2002 and 2004, respectively.

In 2003, he was with STMicroelectronics, Advanced System Technology, San Diego, CA, USA. From 2004 to 2006, he was a Senior Design Engineer at Portland Technology Development, Intel Corporation, Hillsboro, OR, USA. From 2006 to 2008, he was an Assistant Professor with the Information Engineering and Technology Department, German University in Cairo, Cairo. From 2008 to 2010, he was a Technical Lead with the Mentor Hardware Emulation Division, Mentor Graphics Corporation, Cairo. He is currently a Staff Engineer with the Design Creation and Synthesis Division, Mentor Graphics Corporation, and an Associate Professor with the Microelectronics Department, Electronics Research Institute, Cairo. He has authored more than 80 papers, five book chapters, and three books in the fields of high-speed and low-power CMOS design techniques and NoC/SoC. His current research interests include networks-on-chip/system-on-chip, interconnect design and related circuit level issues in high-performance VLSI circuits, clock distribution network design, digital ASIC circuit design, VLSI/SoC/NoC design and validation/verification, circuit verification and testing, embedded systems, and low-power design.

Dr. El-Moursy is an Associate Editor in the Editorial Board of the *Elsevier Microelectronics Journal*, the *International Journal of Circuits and Architecture Design,* and the *Journal of Circuits, Systems, and Computers* and the Technical Program Committee of many IEEE Conferences such as ISCAS, ICAINA, PacRim CCCSP, ISESD, SIECPC, and IDT.

Amal Zaki was the Head of the OBC and Data Handling Subsystem of Satellite Division at Egyptian Space Program in National Authority for Remote Sensing and Space Sciences NARSS. She has been a Chair with the VLSI Department, Electronics Research Institutem Cairo, Egypt, since 2011, where she is a Full Professor. Her current research interests include digital circuit design, computer architecture, mixed-signal VLSI, and MEMS.

Abdelhalim Zekry was a staff member at several rewarded universities. He is currently a Professor of Electronics in the Faculty of Engineering, Ain Shams University, Cairo, Egypt. He has authored over 200 papers. He also supervised over 70 master thesis and 25 Doctorate. His current research interests include microelectronics and electronic applications including communications and photovoltaic.

Prof. Zekry got several prizes for his outstanding research and teaching performance.