2012 25th International Conference on VLSI Design

Energy-Efficient Application Mapping in FPGA through Computation in Embedded Memory Blocks

Anandaroop Ghosh, Somnath Paul and Swarup Bhunia
Department of Electrical Engineering and Computer Science
Case Western Reserve University, Cleveland, OH, USA
Email: $axg468@case.edu$

***Abstract*—FPGAs have emerged as the preferred prototyping and accelerator platform for diverse application domains such as digital signal processing (DSP), security and multimedia, which often impose real-time performance requirements. Most applications in these domains demand efficient implementation of complex datapaths or functions e.g. transcendental functions, which are spatially mapped in the configurable logic or embedded DSP blocks of a FPGA device. Requirement of elaborate computational resources to realize these operations impose a major barrier to energy efficiency. In this paper, we propose to use embedded memory blocks in FPGA for computing to significantly improve energy efficiency of the applications which are dominated by complex datapaths and/or functions. Complex operations are decomposed or fused into large multiple input/output lookup tables (LUTs); mapped to embedded memory blocks and evaluated through memory access over single or multiple cycles. Different parts of an application are selectively mapped into memory or logic/DSP blocks in a heterogeneous mapping framework to maximize energy efficiency. We explore optimal energy configuration of embedded memory for mapping operations of varying input size and develop a complete mapping flow including decomposition, fusion and packing. Effectiveness of the proposed flow is evaluated for a set of applications using a commercial state-of-the-art FPGA system (Altera Stratix IV). Finally, the proposed framework is extended to drastically trade-off energy versus accuracy at run time for common signal processing applications.**

***Index Terms*—FPGA, Embedded RAM, Memory Based Computing, Energy-Efficiency, Energy-Accuracy Trade-off.**

I. Introduction

Reconfigurable computing platforms, such as Field Programmable Gate Array (FPGA), are increasingly used in embedded applications (such as DSP, multimedia, security and graphics) due to the flexibility in application mapping, reduced design cost and largely improved time-to-volume. However, these platforms are well-known to suffer from poor energy efficiency, primarily due to large overhead of their elaborate programmable interconnect (PI) fabric. Currently PI accounts for 80% of power and 60% of delay in FPGAs at nanoscale process technologies [3]. There is a growing need to address the power/energy issues in reconfigurable platforms while retaining their performance and flexibility advantages.

FPGA vendors such as Xilinx and Altera, as well as researchers from academia have investigated various device engineering options (such as low-k dielectric, multiple device thresholds) as well as architecture-level techniques (e.g. clustered architecture) to improve the energy consumption of the FPGA devices [6]. These optimization approaches however, cannot provide adequate solution to reduce the energy requirement for many compute-intensive applications. Besides, as the interconnect delay does not scale as significantly as logic delay, fine-grained architectures of FPGAs suffer from poor technological scalability of performance and energy.

On the other hand, an efficient application mapping methodology that can drastically reduce the need of PIs for an application while using conventional FPGA architecture can be extremely effective in reducing energy consumption. Such an approach can be attractive to both FPGA vendors and users alike, since it does not require modifications in FPGA hardware and hence saves device design/fabrication cost. Moreover, it is also able to work on legacy hardware. In this paper, we propose a novel application mapping methodology that uses the embedded memory arrays in FPGA for mapping complex compute-intensive parts of an application. We note that modern FPGAs come with large number of embedded memory blocks (EMBs) - e.g. Altera Stratix-IV devices are equipped with 17-33 Megabits (Mb) of block random access memory (RAM) in addition to the one-dimensional lookup tables (LUT) for configurable logic blocks (CLBs). As shown in Fig. 1, driven by aggressive technology scaling, FPGA devices from different vendors are integrating larger amount of RAM with faster access speed in each technology generation. In many applications, large part of embedded memory resources remain unused. Hence, through opportunistic use of these RAM blocks for computation - in particular, to realize complex datapaths or functions, we can significantly improve both energy efficiency and resource utilization compared to conventional logic based implementation.

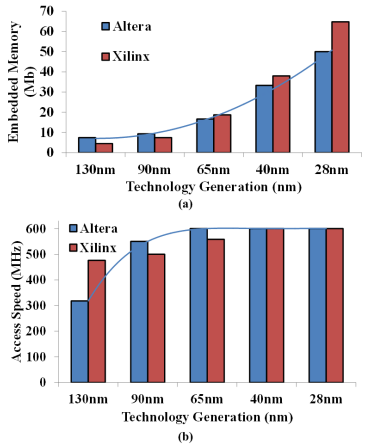

Fig. 1. The trend in embedded memory in FPGA: a) size (Mb), and b) access speed (MHz) for Altera Stratix [9] and Xilinx Virtex [10] series of FPGA devices across technology generations.

Several previous investigations have earlier considered fine-grained application mapping in EMBs inside a FPGA [15], [16]. Although these investigations lay the foundation of using EMBs for computation, they are limited to mapping small Boolean functions in fine-grained applications (e.g. control logic) and do not consider coarse-grained mapping of complex operations (e.g. datapaths or transcendental functions). Moreover, they primarily focus on improving performance and do not analyze the impact on energy dissipation. Finally, they do not study the complete application mapping process and the effectiveness of computation with memory at nanoscale technology nodes using commercial FPGAs.

Compared to existing mapping approaches, the proposed mapping approach achieves significant improvement in energy-efficiency since

1063-9667/12 $26.00 © 2012 IEEE
DOI 10.1109/VLSID.2012.108

IEEE computer society

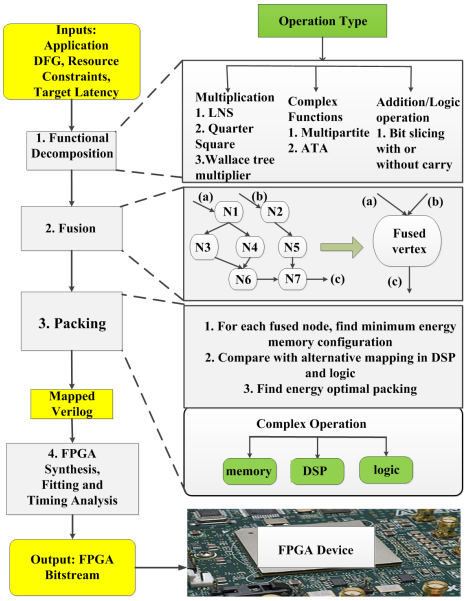

Fig. 2. Proposed application mapping steps using EMBs for computation.

the entire function is computed typically in one or two LUT accesses leading to significant improvement in operational latency and energy due to large reduction in PI overhead. In particular, the key contributions of the paper are:

1) It analyzes the effectiveness of mapping coarse-grained compute-intensive operations in an application to embedded memory blocks in FPGA. It shows that opportunistic mapping of complex datapaths and functions in memory in the form of large multi-input multi-output LUTs can significantly improve energy efficiency. It explores the most energy-efficient memory configuration in a FPGA for mapping operations with varying memory requirements.
2) It then develops a heterogeneous application mapping framework, which combines conventional application mapping in logic and DSP blocks (for DSP-enhanced FPGA devices) with judicious mapping of specific computations in memory. It determines the complete mapping methodology including functional decomposition, which partitions complex functions into multi-input/multi-output LUTs of manageable size, fusion of small operations into a large one, and finally optimal packing of operations into a combination of EMBs and logic array.
3) It validates the effectiveness of the proposed methodology by mapping a number of common scientific and graphics applications on a commercial state-of-the-art FPGA platform (Altera Stratix IV, 40nm process). It compares the performance and energy efficiency for the proposed heterogeneous mapping approach with conventional mapping strategies in FPGA.
4) It shows that computation in memory can be an effective vehicle for energy-accuracy tradeoff at run time. In this case, operand bitwidth for complex operation can be dynamically truncated to achieve exponential reduction in memory space leading to large savings in computation energy. The impact on the quality of service (due to operand truncation) is minimized through choice of optimal values for truncated bits and preferential truncation of the operations. We demonstrate the effectiveness of such dynamic trade-off for a common filtering application.

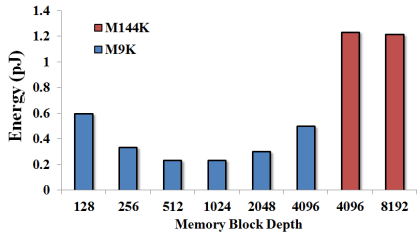

Fig. 3. Variation in energy consumption of a 12-bit input 12-bit output memory with varying memory type and block depth for a commercial FPGA.

II. Use of Embedded Memory Blocks for Computation

It is well-known that a majority of scientific and graphics applications include a set of common compute-intensive kernels, which essentially constitute of basic mathematical operations such as addition, multiplication as well as complex functions such as sine, cosine, reciprocal, arctan, square root, exponentiation, and logarithm [1]. Traditionally, in a FPGA framework the transcendental functions are mapped using CORDIC approach [4] or Taylor series expansion [5], which either require large number of computing resources or suffers from large latency. Evaluation of these functions by holding the output response of a function as LUT in the embedded memory array is an attractive solution in terms of performance and energy consumption. For larger bitwidth, efficient decomposition techniques (as proposed in [7] and [8]) have also been employed. The focus of our work is to achieve energy efficient mapping of applications dominated by complex functions using EMBs. The following computations have been identified to be amenable for mapping to memory.

1) Any function satisfying the maximum LUT input size in a FPGA device. The function can be a regular one (such as $Sine(x)$ where x has a maximum resolution of 16-bits) or any arbitrary function obtained by fusing many simple ones.
2) Complex datapath having constant coefficients like two constant coefficient multiplications followed by an addition.
3) Any function that is easily bit-sliceable like logic and Galois field operations and some datapath operations like addition, magnitude comparison, multiplication, etc.
4) Functions where operands can be truncated with minimal degradation of output quality, e.g. DSP or multimedia applications.
5) Functions which are amenable to decomposition using a different number system. For example, a multiplication of two operands can be converted to a set of single-operand logarithmic operations followed by an arithmetic addition.

III. Application Mapping Methodology

In this section, we describe the proposed application mapping methodology which uses a combination of EMBs and other FPGA resources to maximize energy efficiency for a given input application.

A. Mapping Flow

Figure 2 shows the proposed application mapping methodology. A comprehensive software flow has been developed in C language which implements the proposed methodology. The input to our application mapping flow is a Verilog netlist containing a data flow graph (DFG) representation of the target circuit. Major steps of the mapping process are described below.

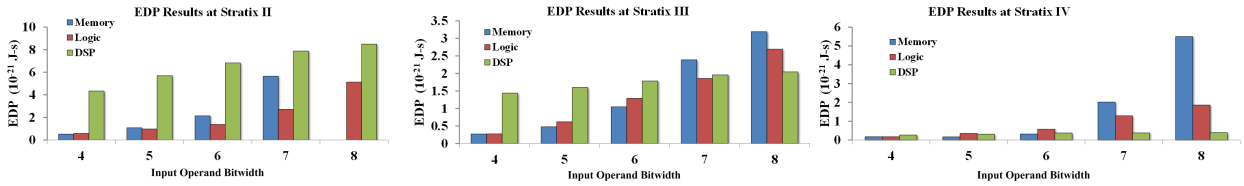

Fig. 4. Comparison of EDP with varying input resolution for constmult-add in Altera Stratix II, Stratix III and Stratix IV series of devices.

TABLE I
EDP IMPROVEMENTS WITH THE PROPOSED HETEROGENEOUS MAPPING APPROACH COMPARED TO MAPPING IN LOGIC AND DSP BLOCKS FOR THREE GENERATIONS OF ALTERA FPGA FAMILIES

Input bitwidth	Stratix II		Stratix III		Stratix IV	
	% Impr. over Logic	% Impr. over DSP	% Impr. over Logic	% Impr. over DSP	% Impr. over Logic	% Impr. over DSP
4	10.26	87.90	1.16	81.18	1.12	33.80
5	-11.58	80.96	22.90	70.14	52.92	47.20
6	-54.62	68.64	18.74	41.34	44.37	14.04
7	-108.30	28.24	-28.60	-21.88	-56.83	-431.72
8	*	*	-18.51	-56.30	-195.79	-1299.69

*cannot place due to lack of memory resources

1) Functional Decomposition: In this step, nodes of the input DFG which violate the maximum LUT input constraint are decomposed into multiple LUTs with varying input sizes to limit the memory resource requirement. The maximum LUT input size is determined by the available memory sizes of the underlying device, for e.g. 16 in the case of Stratix IV devices. The decomposition step employs function-specific decomposition process, such as bit-slicing for large addition and logarithmic number system (LNS) for multiplication. For mapping complex transcendental functions, we employ multi-partite decomposition [8] whenever feasible or Addition Table Addition (ATA) approaches [7] for other functions. Nodes which satisfy the maximum LUT input constraint are not decomposed.

2) Fusion: Fusion involves opportunistic reduction in the total number of operations after decomposition (represented as vertices in the DFG) by combining multiple operations into a single operation which can be suitably mapped in a 2D LUT. Nodes in the modified DFG obtained after decomposition are clustered to achieve maximum input cone sharing [11] without violating the maximum LUT input (M) and output constraint (N).

3) Packing: The fused DFG is packed or allocated into different functional units like CLB, DSP or memory. We follow a greedy heuristic-based approach by packing each operation to a resource which provides optimal energy. In case of memory, the LUT contents of the functional blocks are also generated along with the fused DFG. Finally after packing, the allocated resources and their interconnections are converted to appropriate verilog netlist which consists of the required resources and the EMB contents which is readily mappable into the actual FPGA device with resource and I/O constraints.

B. Energy-Efficient Configuration of RAM Blocks

In the packing step we explore the optimal energy configuration of the EMBs for all the memory blocks instantiated in any application. For energy-efficient configuration of RAM blocks in a FPGA, we explore the design space for a given input and output size by varying memory block type and block depth [14].

Figure 3 shows the variation in energy consumption by varying the block type and depth of a 12-bit input 12-bit output memory. We can observe that the optimal configuration is obtained for a block type of M9K and block depth of 1024.

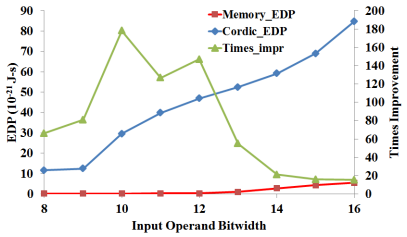

Fig. 5. Variation in EDP with varying input resolution for CORDIC based operations.

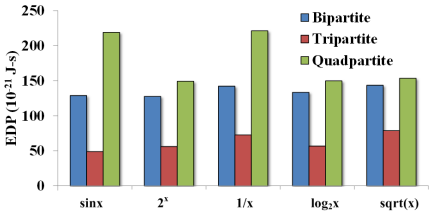

Fig. 6. EDP trends with bipartite, tripartite and quadpartite decompositions.

C. Mapping Complex Datapath in Memory

For smaller bitwidths, datapaths with large number of combinational levels are expected to have significantly better energy efficiency while mapping in memory compared to normal logic or DSP based mapping. To analyze this effect, we mapped a common datapath for DSP applications, namely two constant coefficient multiplications followed by an addition (constmult-add). We found optimal memory configurations for mapping it to memory and obtained energy and delay results across different technology generations, namely Stratix II, Stratix III and Stratix IV. We compared these results with the corresponding mapping results in logic and DSP blocks. Figure 4 compares the energy delay product (EDP) for varying input bitwidths in case of three alternative mapping schemes. Table I shows the corresponding EDP improvement compared to mapping only in logic and DSP. For Stratix IV, we observe EDP improvement for up to 6-bit input resolution, as shown in Table I.

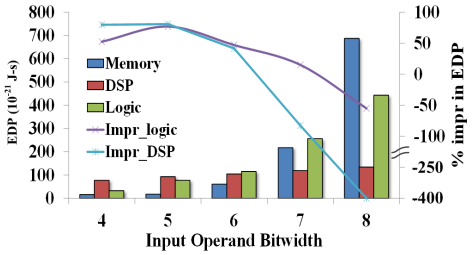

Fig. 7. Variation in EDP with varying input resolution for 8-tap FIR filter.

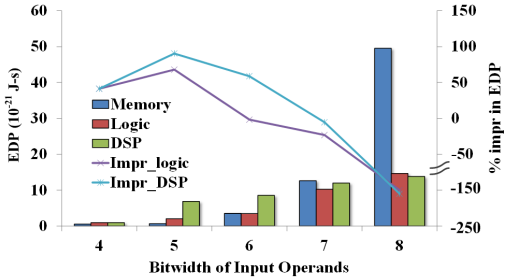

Fig. 8. Variation in EDP with varying input resolution for DWT.

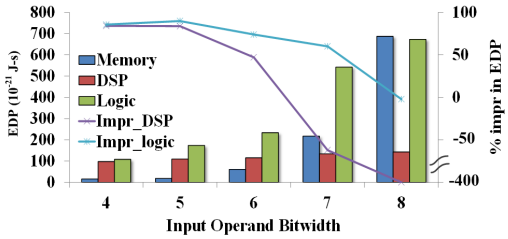

Fig. 9. Variation in EDP with varying input resolution for coherence calculation of an 8-point cluster.

D. Mapping Complex Functions in Memory

Conventionally complex transcendental functions are computed in FPGA using CORDIC based approaches [12] where the latency of the computation increases linearly with the increase in input resolution. On the other hand, a LUT storing the output response of a function is capable of evaluating the function in one or few lookup operations, providing tremendous improvement in terms of energy consumption and latency. In a Stratix IV FPGA, for up to 16-bit input resolution, any function can be mapped in a single lookup table due to adequate embedded memory resources. Figure 5 shows the improvement in energy efficiency or EDP of LUT based approach with respect to normal CORDIC based uni-variable function realization in Stratix IV devices. The EDP improvement is minimum (15X) in case of 16-bit operands and maximum (178X) in case of 10-bit input operands. If the size of the input bitwidth does not match with the size of the lookup table that can be supported, then we need to employ efficient decomposition techniques, such as, multipartite and ATA method [7] in order to minimize the memory requirement at the cost of graceful increase in latency. These decomposition techniques have been investigated in order to achieve energy-efficient mapping of complex transcendental functions. Figure 6 shows the energy efficiency results in case of Stratix IV using bipartite, tripartite and quadpartite decompositions [8] of transcendental functions with 24-bit fixed point inputs. In general, tripartite decompositions give the most energy optimal mapping for 24-bit input uni-variable functions as shown in Fig. 6.

IV. Application Mapping Results

In this section, we evaluate the effectiveness of the proposed approach in improving energy efficiency for several common applications, which require complex datapaths and/or functions.

A. 8-tap FIR Filter

Implementation of a 8-tap FIR includes eight constant coefficient multiplications followed by the addition of the product terms. The EDP trends in Stratix IV for 8-tap FIR filter is shown in Fig. 7 with variation in input bitwidth. In case of the proposed heterogeneous mapping of memory and logic, significant EDP improvement is achieved for up to 7-bit inputs (16.7% improvement for 7-bit) when compared to logic only implementation and 6-bit inputs (41.2% improvement for 6-bit input) when compared to a heterogeneous implementation of logic and DSP blocks.

B. Calculation of Approximation Coefficient in DWT

The datapath for the approximation coefficient computation in a discrete wavelet transform (DWT) consists of two constant coefficient multiplications of the odd samples followed by one addition. After a single level of truncation, the intermediate output is added to the even sample of the current level. The EDP results with the variation in input bitwidth is shown in Fig. 8 and compared with that of logic and DSP based implementation in Stratix IV. EDP improvements are observed for up to 6-bit input width compared to both logic only (nearly 1% improvement for 6-bit input) and a heterogeneous implementation of logic and DSP (58.7% improvement in EDP for 6-bit input).

C. Coherence Calculation in a Cluster

For calculation of coherence in a cluster, first, the absolute distance between the different points allocated to a particular cluster from its centroid is computed and then the distances for all the points from their corresponding cluster centroid are squared and added which define the coherence of the cluster. Memory based computation in coherence calculation of an 8-point cluster is extremely helpful and gives significant advantage for up to 6-bit input over a DSP based implementation (47.4% improvement for 6-bit input) and up to 7-bit input over a logic only implementation (58.2% improvement for 7-bit input) The advantages in EDP over different bitwidth of inputs are shown in Fig. 9 for a Stratix IV platform.

D. Newton Raphson Method of Root Finding

Newton Raphson's (NR) method is widely used for the evaluation of roots for polynomial functions. Consider a 3rd order polynomial given by

$$f(x) = a_3x^3 + a_2x^2 + a_1x + a_0 \quad (1)$$

NR method employs the following iteration formula in order to move closer to the actual root:

$$X_{n+1} = X_n - \frac{f(x)}{f'(x)} = X_n - \frac{a_3X_n^3 + a_2X_n^2 + a_1X_n + a_0}{3a_3X_n^2 + 2a_2X_n + a_1} \quad (2)$$

For a 24-bit fixed point input, the proposed framework takes 106 KB of memory through the ATA based decomposition method [7]. The energy and EDP improvements achieved with respect to conventional mapping in FPGA is shown in Table II. The results of conventional mapping for all applications are obtained by choosing the better EDP results between logic and DSP-based implementation.

E. Solution to Schrodinger Equation 1-D

Finding solution of a time-independent Schrodinger wave equation for arbitrary periodic potentials is a common scientific application. Similar to many other scientific applications, it is dominated by transcendental functions. For single dimension, the function is given by:

$$\psi_n(x) = \sqrt{\frac{2}{L}} sin(\frac{n\pi x}{L}) \tag{3}$$

A FPGA based evaluation of this function using the proposed mapping approach for 24-bit fixed-point input operand requires 193.96 KB of memory. Due the increased latency of CORDIC based computation, the proposed framework has significant improvement in energy and EDP over conventional mapping in FPGA, as shown in Table II.

F. Lighting in Computer Graphics

Lighting is a very well-known compute-intensive kernel in computer graphics, which involves computing several transcendental functions in parallel. The algorithm for lighting is as shown below:

$$\begin{aligned} I = K_a + c((K_d + K_s sin(v,t))sin(l,t) - \\ K_s exp[sln(cos(v,t))]cos(l,t)) \end{aligned} \tag{4}$$

where $c = not(msb(l.n))$ and K_a, K_d and K_s are the ambient, diffuse and specular coefficients, s is the surface shininess, l, n, v, t are the light, normal, view and tangent unit vectors on the surface and $sin(v,t)$ denotes the sine of the angle between the vectors v and t. The actual inputs to the computing framework are the vectors v, t and l whereas the coefficients or the constants are assumed to be known beforehand. We find that for a 24-bit fixed point input, proposed computing framework requires 335.84 KB of memory. The energy and EDP improvements achieved with respect to conventional mapping in FPGA is shown in Table II.

G. Gaussian Random Noise Generator

Generation of random numbers with Gaussian probability distribution is used in wide range of applications, such as simulation of economic systems, scientific applications like molecular dynamics, and cryptography. Popular Box-Muller Method of generating the Gaussian noise involves evaluation of three complex transcendental functions, namely $f1$, $f2$, $f3$ where $f1 = \sqrt{-ln(u_1)}$, $g1 = \sqrt{2}\sin(2\pi u_2)$ and $g2 = \sqrt{2}\cos(2\pi u_2)$ where $u1$ is 24-bit and $u2$ is 16-bit wide. The input bitwidth determines the maximum possible value of the Gaussian sample. Here it is assumed that that the input bitwidth is 40 bits and the output bitwidth is 24 bits. The total memory requirement is 270.76 KB. The energy and EDP improvements achieved with respect to conventional mapping in FPGA is shown in Table II.

V. Dynamic Energy Accuracy Trade-Off

As demonstrated in the previous sections, heterogeneous mapping can provide significant improvement in energy consumption for specific datapath bitwidth in case of applications dominated by complex datapaths or functions. As a result, such a heterogeneous mapping methodology can be utilized to trade-off energy consumption and accuracy requirement dynamically for DSP and multimedia applications. In this case, operand bitwidth can be reduced at run time through truncation such that the memory requirement in mapping the complex datapaths/functions can be exponentially reduced leading to large savings in energy. Clearly, operand truncation would have impact on the output quality. However, judicious choice of bit values assigned at truncated bit positions and a non-uniform truncation approach that aggressively truncates inputs of only the less critical components can lead to modest impact on output. Next, we describe the truncation approaches in details.

1) Uniform Truncation: First, we study the effect of truncating inputs uniformly, i.e. changing input resolution uniformly across the board to see the effect on primary output values of a FIR filter. We truncate inputs at all taps equally and generate the LUTs by optimal value allocation at the truncated bits. The designed filter is essentially a low-pass equiripple 32-tap FIR filter with a passband frequency of 9.6 kHz and a stopband frequency of 12 kHz. The original filter input operand resolution is 8 bits. Three bits of the inputs, starting from the LSB have been allocated different values and the impact in the output mean square error (MSE) for different value allocation at the inputs is shown in Fig. 10. The inputs applied to the filter is an impulse input of magnitude $8'$b11111111. It can be inferred that for 3-bit truncation, LUT representation assuming $3'$b011 or $3'$b100 values at the truncated positions provides minimum error at the output.

2) Preferential Truncation: To minimize the effect on output quality due to truncation while maintaining the benefit of energy reduction, we propose a preferential truncation approach, which truncates specific inputs more aggressively than others. This leverages on the observation that most DSP applications have some components which are more critical than others in terms of output quality [17]. Hence, unlike the uniform truncation approach discussed earlier, we propose truncating the inputs of a less significant component more aggressively than the significant ones.

The variation in the stopband ripple due to zeroing different tap coefficients is shown in Fig. 11. It shows the sensitivity of different taps to output quality. We observe that the middle taps are more critical than the side ones. Based on this observation, less truncation is applied to the more significant taps while more aggressive truncation is applied at the non-critical taps in order to achieve minimum impact on output quality at iso-energy consumption. In particular, the middle tap inputs are truncated by one bit, the inputs corresponding to the 1st and 2nd order coefficients are truncated by 2 and 3 bits, respectively, and so on. We refer this preferential truncation scheme as $Config.1$. We also explore two other variations of preferential truncations. In the second ($Config.2$) and third schemes ($Config.3$), 2 and 3 bits are truncated respectively at the middle with increasing truncation at the sides.

Figure 12 presents the output MSE and energy consumption trade-off for different truncation scenarios in case of a 32-tap FIR filter. Figure 12(a) compares between two uniform truncation approaches. Zero value allocation denotes blind truncation with $1'$b0 at the truncated bits. The three design points in case of zero value allocation denote $1'$b0, $2'$b0 and $3'$b0 truncations at primary inputs. Similarly, optimal value allocation denote that optimal bit values are assigned at the truncated bits ($1'$b0, $2'$b01, $3'$b011, respectively). We observe that optimal value allocation gives lesser MSE compared to zero value

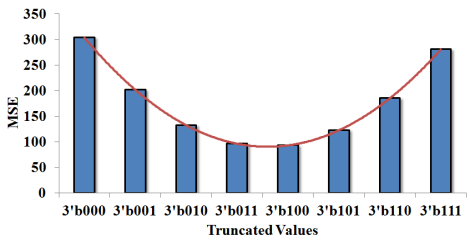

Fig. 10. Variation in mean square error (MSE) with different value assignments at the truncated bits for a 32-tap FIR filter.

TABLE II
COMPARISON OF ENERGY, LATENCY AND EDP FOR SEVERAL COMMON SCIENTIFIC AND GRAPHICS APPLICATIONS FOR 24-BIT FIXED POINT INPUTS

Applications	Conventional Mapping in FPGA			Proposed Mapping Framework				%Impr. in Energy
	Latency (ns)	Energy (pJ)	EDP ($10^{-18}J-s$)	Latency (ns)	Energy (pJ)	EDP ($10^{-18}J-s$)	Memory (KB)	
Polynomial evaluation (3rd order)	49.0	379.1	18.9	7.6	53.7	0.4	106.0	85.8
Schrodinger equation (1D)	274.4	1258.1	345.2	25.0	368.7	9.2	193.9	70.7
Lighting (computer graphics)	380.0	3684.0	1400.2	90.0	1540.4	138.6	335.8	58.2
Gaussian noise generator	180.0	232.3	41.8	27.6	157.2	4.3	270.8	32.3

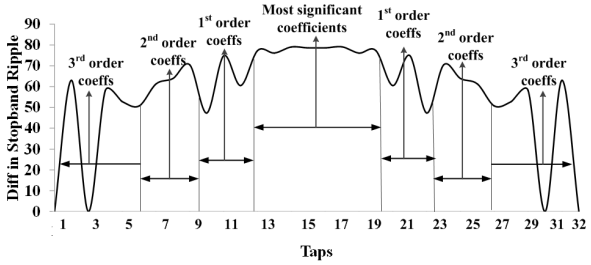

Fig. 11. Variation in stopband ripple magnitude by zeroing different coefficients of a 32-tap FIR filter.

allocation but consumes the same amount of energy.

Figure 12(b) compares between uniform (with optimal value assignment) and preferential truncation approaches. The three design points for preferential truncation correspond to $Config.1$, $Config.2$ and $Config.3$ as described above. Preferential truncation can provide significant improvement in output quality at similar energy levels compared to optimal value allocation. In particular, for $Config.2$, the proposed preferential truncation methodology results in 87% and 96.9% improvement in output MSE over optimal value and zero value allocation respectively at iso-energy consumption.

VI. CONCLUSION

We have presented a novel application mapping process for FPGA that exploits the embedded memory blocks in FPGA for computation to minimize the overhead of programmable interconnects. We show that the proposed mapping process can significantly improve energy efficiency for many applications which require complex datapath, complex functions or both. Scalability of the proposed mapping to different bitwidth is addressed with detailed analysis. Software architecture to perform functional decomposition, fusion of operations and packing of individual coarse-grained operations is presented. The effectiveness of the proposed mapping has been validated for a commercial FPGA platform, namely Altera Stratix IV FPGA. Finally, computation in embedded memory in FPGA is extended to dynamically trade-off between energy and accuracy through use of judicious bitwidth truncation at run time. With emergence of novel high-density memory technologies- both volatile and non-volatile-FPGA devices are expected to integrate more memory with improved performance, which can significantly benefit the proposed approach. Future work will include extending the framework to other domain of applications, support for different data format and precision, and more efficient functional decomposition.

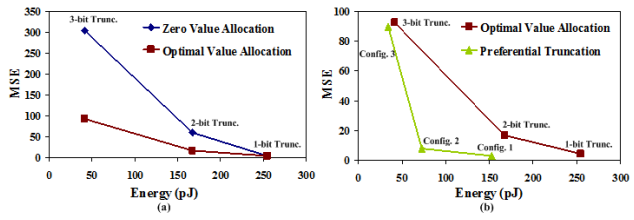

Fig. 12. MSE and energy consumption trade-off for a 32-tap FIR filter using three variations of heterogeneous mapping approach - (a) zero vs. optimal value allocation, (b) optimal value allocation vs. preferential truncation.

REFERENCES

[1] "The landscape of parallel computing research: A view from Berkeley" [Online] http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html.

[2] T. Good and M. Benaissa, "AES on FPGA from thr Fastest to the Smallest," em Cryptographic Hardware and Embedded Systems, 2005.

[3] A. Rahman, S. Das, A. P. Chandrakasan and R. Reif, "Wiring Requirement and Three-Dimensional Integration Technology for Field Programmable Gate Arrays," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 11, No. 1, 2003.

[4] W.B. Ligon III, G. Monn, D. Stanzione, F. Stivers and K.D. Underwood, "Implementation and Analysis of Numerical Components for Reconfigurable Computing," *Aerospace Applications Conference*,1999.

[5] C. Brunelli, H. Berg and D. Guevorkian, "Approximating sine functions Using Variable Precision Taylor Polynomials," *IEEE Workshop on Signal Processing Systems*, 2009.

[6] J. Lamoureux and W. Luk, "An Overview of Low Power Techniques for Field Programmable Gate Arrays," *NASA/ESA Conference on Adaptive Hardware and Systems*, 2008.

[7] W.F. Wong and E. Goto, "Fast Evaluation of Elementary Functions in Single Precision," *IEEE Transactions on Computers*, Vol. 44, No. 3, 1995.

[8] F. D. Dinechin and A. Tisserand, "Multipartite Table Methods," *IEEE Transactions on Computers*, Vol. 54, No. 3, 2005.

[9] "Stratix FPGA: Low Power, High Performance" [Online] http://www.altera.com/devices/fpga/stratix-fpgas/stratix/stratix/stx-index.jsp

[10] "Xilinx Virtex series of FPGAs" [Online] http://www.xilinx.com/products/index.htm

[11] S. Paul, S. Chatterjee, S. Mukhopadhyay and S. Bhunia, "A Circuit-Software Co-Design Approach for Improving EDP in Reconfigurable Frameworks," *International Conference on Computer-Aided Design*, 2009.

[12] T. Lang and E. Antelo, "High Throughput CORDIC Based Geometry Operations for 3D Computer Graphics," *IEEE Transactions on Computers*, Vol. 54, No. 3, 1997.

[13] D.U. Lee, W. Luk, J.D. Villasenor and P.Y.K. Cheung, "A Gaussian Noise Generator for Hardware Based Simulations," *IEEE Transactions on Computers*,Vol. 53, No. 12, 2004.

[14] R. Tessier, V. Betz, D. Neto, A. Egier and T. Gopalsamy, "Power-Efficient RAM Mapping Algorithms for FPGA Embedded Memory Blocks," *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, Vol. 26, No. 2, 2007.

[15] J. Cong and S. Xu, "Technology Mapping for FPGAs with Embedded Memory Blocks," *International Symposium on Field Programmable Gate Arrays*, 1998.

[16] S. Wilton, "SMAP: Heterogeneous Technology Mapping for Area Reduction in FPGAs with Embedded Memory Arrays," *International Symposium on Field Programmable Gate Arrays*, 1998.

[17] N. Banerjee, J.H. Choi and K. Roy, "A Process Variation Aware Low Power Synthesis Methodology for fixed point FIR filters," *International Symposium on Low Power Electronics and Design*, 2007.