International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 06, June-2015

Advancement of Memory-based Hardware for Efficient Resource-Constraint Digital Signal Processing Systems

Nandita Jaiswal
M.Tech (VLSI), Department of Electronics & Communication Engineering
Hindustan College of Science & Technology
Farah, Mathura, India

***Abstract*— Rapid advancement in very large scale integration (VLSI) technology and hardware performance of digital devices have paved way to efficient memory-based computing systems as alternative to the conventional logic-only computing in order to meet the stringent constraint and growing requirements of the digital signal processing (DSP) systems in different application environments. Several algorithms and architectures have been suggested in the past to reduce the area and time complexities of commonly encountered computation-intensive cores of DSP functions by memory-based computing. Different scientific programs with high efficiency, faster operation and better performance gain have been developed. However most of the scientific programs and applications remain compute-bound in today's scenario and there is an urge to develop many more algorithms and architecture for flexible design of area-delay-power-efficient systems for various DSP applications.**

I. Introduction

Digital signal processing (DSP) is considered as the major component of the digital revolution that is currently taking place around the world. The increasing popularity of digital technology, in the recent years, have not only made the DSP applications more prevalent in daily use, but also the algorithms are subjected to more stringent specifications to meet the basic constraints of the application environments. As a natural follow up of the situation, significant research interest has been observed, in the recent years, for developing improved algorithms and architectures to design the DSP systems with less power dissipation, higher speed performance and less area complexity. But due to mutually conflicting behavior of these constraints, it has been noticed that one has to trade one or more aspects to meet a more important requirement [1]. Architectural solutions can be obtained to trade area for time and power or to trade time for area and cost, but it is difficult to minimize the cost, area, delay and power all together in a given architecture. Several efforts have been made to minimize the arithmetic complexities of the algorithms in order to reduce the overall area-delay-power complexities [2].

Algorithms pertaining to the DSP operations are basically computation-intensive, and most of their applications are of hard-real-time by nature [3]. Apart from that, the DSP systems are very often used in small portable devices which depend mostly on limited battery power [4]. The rigid constraints on size and cost do not usually leave scope for a cooling arrangement in these systems, while the system reliability falls to half for every 10 to 20 degree Celsius rise in temperature [5]. The general-purpose machines, however, very often do not meet the speed-requirement of real-time applications and size-constraints of many portable systems. It is, therefore, important to design dedicated very large-scale integration (VLSI) chips for fast and efficient computation of the DSP applications. Efforts have been made to derive modular VLSI for the fast DSP algorithms, based on recursive decomposition [6], [21]. Although, these algorithms require less number of arithmetic operations, they involve complicated routing and large design-time due to their irregular signal-flow graphs. Moreover, the accuracy of fixed-point implementation of these algorithms degrades as a result of successive truncation during the recursive decomposition process. Similarly, the VLSI realization of time-recursive algorithms also suffers from numerical problems and involves difficulty of pipelineability and increased hardware-complexity [22], [23]. It is also observed further that the algorithms optimized for software-implementation, in general, are not well-suited for dedicated hardware-implementation. Parallel algorithms and architectures are, therefore, imperative for efficient realization of DSP functions in VLSI structures. Appropriate algorithm design has a major role on developing a hardware entity that can meet the system requirements and specification. Not only it should necessarily lead to reduction of computational-complexity, but also should facilitate maximization of concurrency by exploiting the possible parallelism to achieve high-throughput performance. Moreover, the architecture should be developed synergetic with the underlying algorithms to derive a cost effective and area-time-power efficient optimal VLSI.

As the scaling in silicon technology has progressed over the last four decades, the semiconductor memory has become cheaper, faster and more efficient in terms power dissipation. Memory-based designs consequently are gaining substantial popularity in the DSP application space. Most of the DSP algorithms involve repetitive multiply accumulate operations, and the multipliers, the not only consume most of the resources of the system but also involve most of the computation-time. Significant research have, therefore, been made in the past two decades for efficient multiplier less implementation of DSP systems, which can be classified into three-basic categories, e.g., adder-based implementation, CORDIC implementation and memory-based implementation [7]–[12]. The objective of this paper is to highlight some of the current trends in memory technology along with some development of algorithms and architectures for memory-based hardware design to handle the multiple conflicting constraints of DSP applications.

The rest of the paper is organized as follows: A brief overview of VLSI implementation of DSP algorithms is presented in Section-II. The two important developments on algorithmic aspects and architectural approach of memory-based computing systems are described in Section-III. Some of the current trends in growth of memory technology resulting from different application environments are briefly discussed in the Section-IV and conclusion of the paper are presented finally in Section-V.

II. Comparison Study Of VLSI Implementation Of DSP Algorithms

In general, we can obtain a significant improvement in any computational structure of a VLSI implementation by appropriately restructuring the algorithmic computational structure of a DSP algorithm. This is sometimes called *algorithmic engineering* [13]. In order to efficiently make use of this restructuring approach, it is necessary to have a clear architectural target. However, restructuring the forward and inverse discrete cosine transform (DCT)/discrete sine transform (DST) algorithms in such a way that we can obtain an efficient structure that allows the use of the memory-based implementation techniques is a challenging design problem. The DCT and DST [14]–[16] are orthogonal transforms, which are represented by basic functions used in many signal processing applications, especially in speech and image transform coding [17]. These transforms are good approximations to the statistically-optimal Karhunen–Loeve transform (KLT) [16], [17]. The choice of the DCT or DST depends on the statistical properties of the input signal, which in the case of image processing is subject to relatively fast changes. The DCT provides better results for a wide class of signals. However, there are other statistical processes, such as the first-order Markov sequences with specific boundary conditions, for which the DST is a better solution. Also, for low correlated input signals, DST provides a lower bit rate [17]. There are some applications as in [18] and [19], where both the DCT and DST are involved. Thus, a very large-scale integration (VLSI) structure that allows the use of both the DCT and DST is desired. Since both the DCT and DST are computationally intensive, many efficient algorithms have been proposed to improve the performance of their implementation, but most of these are only good software solutions. For hardware implementation, appropriate restructuring of the classical algorithms or the derivation of new ones that can efficiently exploit the embedded parallelism is highly desirable. In order to obtain an optimal hardware implementation, it is necessary to treat the development of the algorithm, its architecture, and implementation in a synergetic manner.

Fast DCT and DST algorithms, based on a recursive decomposition, result in butterfly structures with a reduced number of multiplications, but lead to irregular architectures with complicated data routing and large design time, due to the structure of their signal flow graphs, even though efforts have been made to improve their regularity and modularity as in [20] and [21]. Also, the successive truncations involved in a recursive decomposition structure lead to a degradation in accuracy for a fixed point implementation. The VLSI structures based on time-recursive algorithms [21]–[24] are not suitable for pipelining due to their recursive nature, and suffer from numerical problems, which can severely compromise their low hardware complexity.

The data movement and transfer play a key role in determining the efficiency of a VLSI implementation of the hardware algorithms [26]–[29]. This is the reason why regular computational structures such as the circular correlation and cyclic convolution lead to efficient VLSI implementations [26]–[28] using modular and regular architectural paradigms such as the distributed arithmetic [11] and systolic arrays [30]. These structures also avoid complex data routing and management, thus leading to VLSI implementations with reduced complexity, especially when the transform length is sufficiently large.

Systolic arrays [30] represent an appropriate architectural paradigm that leads to an efficient VLSI implementation due to its regularity and modularity, with simple and local interconnections between the processing elements (PEs); at the same time, they yield a high-performance by exploiting concurrency through pipelining or parallel processing. However, a large portion of the chip is consumed by the multipliers, putting a severe limitation on the allowable number of PEs that could be included.

The memory-based techniques [27], [28], [11], [31] are known to provide improved efficiency in the VLSI implementation of DSP algorithms through increased regularity, low hardware complexity and higher processing speed by efficiently replacing multipliers with small ROMs as in the distributed arithmetic (DA) or in the look-up table approach. The DA is popular in various digital signal processing (DSP) applications dominated by inner-product computations, where one of the operands can be fixed. It uses ROM tables to store the pre-computed partial sums of the inner product. Such a scheme has been adopted to implement several commercial products due to its efficiency in VLSI implementation [32], [33]. However, the main problem is that the ROM size increases exponentially with the transform size, thus rendering the technique impractical for large transform sizes. Moreover, due to the feedback connection in the accumulator stage, the structure obtained is difficult to pipeline.

In [27], a new memory-based implementation technique that combines some of the characteristics of the DA and systolic array approaches has been proposed. When one of the operands is fixed, one can efficiently replace the multipliers by small ROMs that contain the pre-computed results of the multiplication operations. If the size of the ROM is small, a significant increase in the processing speed can be obtained since the ROM access time is considerably smaller than the time required for a multiplication. The resulting VLSI structures are easy to pipeline allowing an efficient combination of the memory-based implementation techniques with the systolic array concept. Using the partial sums technique [32], it has been shown that the size of the ROM necessary to replace a multiplier can be further reduced to half at the cost of an extra adder [27].

Most of the reported unified systolic array-based VLSI designs [24], [25], [35], [36] obtain the flexibility of computing DCT/DST and/or inverse DCT/inverse DST (IDCT/IDST) by feeding the different transform coefficients into the hardware structure. They cannot use efficiently the memory-based implementation techniques since they are not able to use the constant property of the coefficients, namely that for both the DCT and DST, the coefficients are the same and are fixed for each processor. Moreover, they use an additional control module to manage the feeding of the transform coefficients into the VLSI structure, and have a high I/O cost.

The unified DA-based implementations of the DCT/DST and IDCT/IDST algorithms based on a general formulation, presented in [37] and [38], also do not exploit the constant property of the transform coefficients in each processor, nor do they benefit from the advantages of the cyclic convolution structures. Thus, the unification is achieved with a lower computational throughput and a higher hardware complexity. In addition, they have the overheads of the bit-serial implementations with parallel-to-serial and serial-to-parallel conversions, and lower processing speeds compared to the bit-parallel ones as they need more than one clock cycle per operation. Moreover, they are difficult to pipeline and are appropriate only for small values of transform length. Using a dual-port ROM-based DA-like realization technique [39] an efficient design strategy to obtain a unified VLSI implementation of DCT/ DST/IDCT/IDST is achieved by an appropriate reformulation of the DCT, DST, IDCT and IDST algorithms, whose transform length is a prime number, so that they retain all the advantages of the cyclic convolution-based implementations. Thus, an efficient unified VLSI structure, wherein a large percentage of the chip area is shared by all the transforms, and which results in a high computing speed with a low hardware complexity, low I/O cost, and a high degree of regularity, modularity and local connectivity, is presented. Many more pioneer work are yet to come in near future to further enhance are resource-constraint requirements for memory-based implementation of DSP systems.

III. Algorithms And Architectures For Memory Based Computing Systems

Most of the DSP algorithms involve repetitive multiply accumulate operations and inner-product computation. Besides, very often the multiplying coefficients (e.g., filter coefficients or transform kernel coefficients) remain constant during the DSP operations. This behavior of DSP algorithms is utilized to realize the memory-based computing systems. There are two basic variants of memory-based computing techniques found to be popularly used. One of the techniques is the direct memory-based implementation of multiplications [12], while the second is based on distributed arithmetic (DA) [11].

A. Design Techniques

1) Direct-Memory-Based: In the direct-memory-based implementations, the multiplications of input values with the fixed coefficients are performed by a look-up-table (LUT) of size 2^L, (L is the word-length) where each of the LUTs contains the pre-computed product values for all possible values of input samples.

2) Distributed Arithmic: The DA principle is used primarily to compute the inner-products by repeated shift-add operations of partial products corresponding to the successive bit-vectors of one of the input vectors. When one of the vectors is invariant, it is possible to store all the partial product values in a memory. In filtering operations and discrete transform evaluation, one of the vectors is derived from the input samples while the other vector is usually fixed (e.g., impulse response of a filter or coefficients of the transform kernel etc). The inner-products, thus, can be computed by using a LUT and a shift-accumulator, by a straight-forward implementation of the DA-principle.

B. Comparison

Each of these memory-based techniques, however, had some advantages and disadvantages over one another.

- Direct-memory-based implementation involves less hardware complexity compared with the DA-based method, when the word-length is less than the transform-length, while the latter involves less hardware-complexity otherwise.
- In case of DA based method, the time-complexity is independent of the transform-size or the number of filter-taps; and depends only on the word-length. In the direct-memory-based implementations, time-complexity is independent of word-length but increases linearly with the transform size.
- To minimize the I/O bandwidth and to have hardware-efficiency of direct memory-based implementation, sinusoidal transforms are usually converted into cyclic convolution form [27]. Since the average computation time and the latency of direct-memory based implementation is high for large transform-lengths, novel algorithms have been proposed in the last a few years to decompose the sinusoidal transforms into multiple number of circular convolution or convolution-like structures of smaller convolution-lengths [29],[34]and[39]–[43]. Such decompositions have resulted in improvement of throughput performance with substantial reduction of hardware and computational latency. New decomposition schemes have also been suggested, similarly, to reduce the computational latency and overall area-delay complexity of direct-memory-based implementation of large order finite impulse response (FIR) filters [44].
- The major disadvantage of the DA-approach is that its memory size increases exponentially with the transform-length or the filter order. Memory-partitioning and multiple memory bank approach along with flexible multi-bit data-access mechanisms are suggested for FIR filtering and inner-product computation in order to reduce the memory-size of DA-based implementation [11], [12]and [45]–[48]. Attempts have also been made to reduce the memory space in DA-based architectures using offset binary coding [49] and group distributed technique [50]. A systolic realization of linear and circular convolution based on coefficient partitioning is suggested in a recent paper for area-delay efficient DA-based systolic architectures [51]. An LUT-less adder-based DA approach has been suggested where memory-space is reduced at the cost of additional adders [52]. Some efforts have also been made for DA-based implementation of recursive filters and reduction of dynamic power by minimization of bit-transitions during the addition of partial results [53], [54]. Based on the DA decomposition

technique, several systolic and systolic-like architectures of discrete sinusoidal transforms are also suggested in the last ten years to have improved area-delay performance over the existing structures [28],[37],[38],[55]and[56]. A few DA-based architectures are suggested for video and multimedia applications and adaptive FIR filtering [57]–[59], while many more DA-based accelerators are expected to come up in the future years.

IV. Different Application Environment Leading To Current Trends In Memory Technology

DSP plays a vital role in digital modulation and demodulation, speech and image data compression, speech recognition, synthesis and equalization, spectral estimation and analysis, along with a wide range of adaptive filtering applications [60], [61]. The DSP functionalities are, therefore, appearing increasingly in electronic systems for wired- and wireless communication, interactive multimedia systems, biomedical instrumentation, military surveillance and target tracking operations, satellite and aerospace control, remote sensing, and in a host of digital consumer products.

According to the requirement of different application environments, memory technology has advanced in a wide and diverse manner. Radiation hardened memories for space applications, wide temperature memories for automotive, high reliability memories for biomedical instrumentation, low power memories for consumer products, and high-speed memories for multimedia applications are under continued development process to take care of the special needs [62], [63]. Although traditionally memory has remained as an integral part of general purpose computers as a subsystem to store programs and data, it has undergone a lot of transformation in terms of its hierarchical organization and access mechanism. Interestingly also the concept of memory as a standalone subsystem is being replaced by embedded memories those are integrated as part within the processor chip to derive much higher bandwidth between a processing unit and a memory macro with much lower power consumption [64]. To achieve overall enhancement in performance of computing systems and to minimize the bandwidth requirement, access delay and power dissipation, either the processor has been moved to memory or the memory has been moved to processor in order to place the computing-logic and memory elements at closest proximity to each other [65].

According to International Technology Roadmap for Semiconductors (ITRS) [5] system complexities dramatically increase with the amount of software in embedded systems and the rapid adoption of multi-core SOC architectures. Not only is software dominating overall design effort as shown in Fig.1, but hardware dependent software that is tightly coupled to hardware and required functionality, must be eventually handled by an SOC integration and verification process that is still hardware-centric today. Methodological aspects are rapidly becoming much harder than tools aspects as enormous system complexity can be realized on a single die, but exploiting this potential reliably and cost-effectively will require a roughly 50 times increase in design productivity over what is possible today.

Briefly, we discuss here some of the current trends in memory technology which appear to be very much in favor of efficient realization of dedicated and reconfigurable memory-based computing systems for DSP applications. Some of the interesting projections of ITRS pertaining to the time during which research, development, qualification/pre-production and continuous improvement should be taking place for the potential solution, which involves significant technological innovation are shown in Fig. 2, 3 and 4 for logic, DRAM and Non-Volatile memory technology respectively. The industry faces a major overall challenge due to the sheer number of major logic technological innovations required over the next five years: enhanced mobility and high-field transport, high-κ/metal gate stack (which are already implemented but requiring continuous improvement with scaling), ultra-thin body fully depleted SOI, and multi-gate MOSFETs, with quasi-ballistic transport. Future innovations in logic technology are: Enhanced transport with alternate channels: III-V or/and Germanium, Enhanced transport with alternate channels: CNT, Nanowire, grapheme and Non-CMOS Logic Devices and Circuits/Architectures as depicted in Fig. 2.

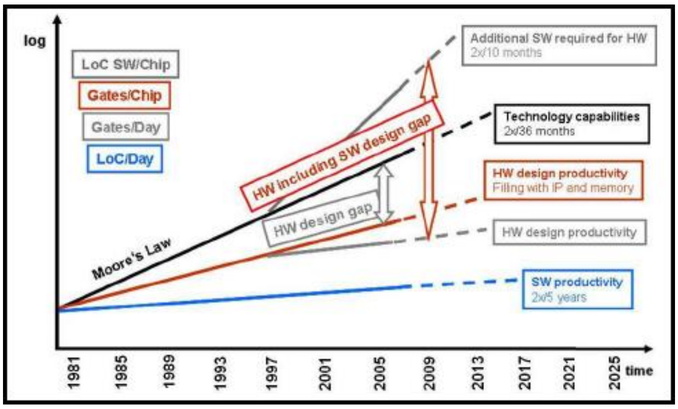
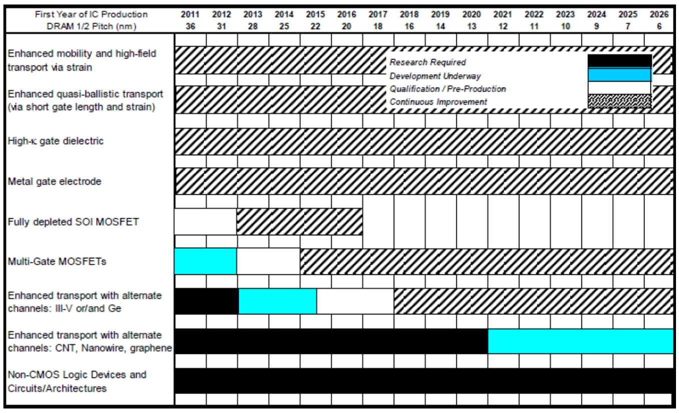


Fig. 1. Hardware and Software Design Gaps versus Time



First Year of IC Production DRAM 1/2 Pitch (nm)	2011 36	2012 31	2013 28	2014 25	2015 22	2016 20	2017 18	2018 16	2019 14	2020 13	2021 12	2022 11	2023 10	2024 9	2025 7	2026 6
Enhanced mobility and high-field transport via strain	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Enhanced quasi-ballistic transport (via short gate length and strain)	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
High-κ gate dielectric	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Metal gate electrode	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Fully depleted SOI MOSFET	□	□	▨	▨	▨	▨										
Multi-Gate MOSFETs	🟦	🟦	□	□	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Enhanced transport with alternate channels: III-V or/and Ge	■	■	🟦	🟦	🟦	□	□	▨	▨	▨	▨	▨	▨	▨	▨	▨
Enhanced transport with alternate channels: CNT, Nanowire, graphene	■	■	■	■	■	■	■	■	■	■	🟦	🟦	🟦	🟦	🟦	🟦
Non-CMOS Logic Devices and Circuits/Architectures	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Research Required	■
Development Underway	🟦
Qualification / Pre-Production	□
Continuous Improvement	▨

Fig. 2. Logic Potential Solutions

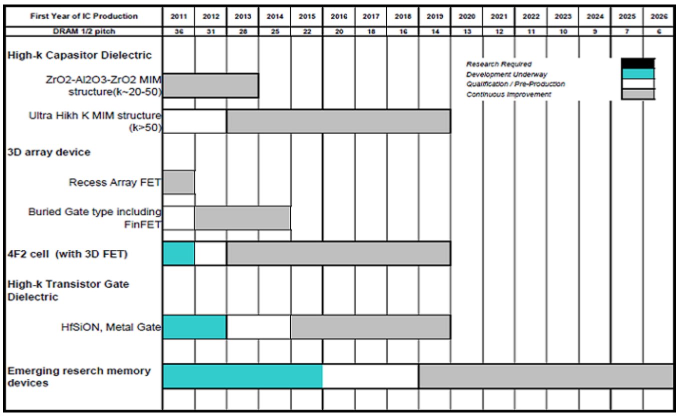


Fig. 3. DRAM Potential Solutions

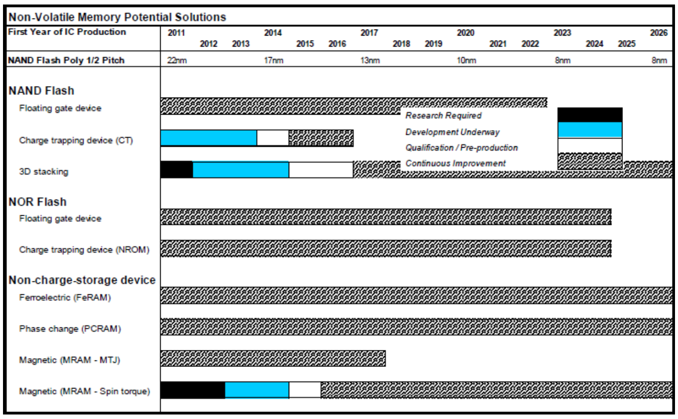


Fig. 4. Non-Volatile Memory Potential Solutions

As the DRAM storage capacitor gets physically smaller with scaling, dielectric materials having high relative dielectric constant (κ) will be needed. Therefore metal-insulator-metal (MIM) capacitors have been adopted using high-κ (ZrO2/Al2O/ZrO2) as the capacitor of 40-30's nm half-pitch DRAM and this material evolution will be continued and ultra high-κ (perovskite κ > 50 ~ 100) material will be released. Also, the physical thickness of the high-κ insulator should be scaled down to fit the minimum feature size. Due to that, capacitor 3-D structure will be changed from cylinder to pillar shape. On the other hand, with the scaling of peripheral CMOS devices, a low-temperature process flow is required for process steps after formation of these devices. This is a challenge for DRAM cell processes which are typically constructed after the CMOS devices are formed, and therefore are limited to low-temperature processing. The other big topic is 4F2 cell migration. As the half-pitch scaling become very difficult, it is impossible to sustain the cost trend. The most promising way to keep the cost trend and increasing the total bit output by generation is changing the cell size factor (a) scaling (where a = [DRAM cell size]/[DRAM half pitch]2). Currently 6F2 (a = 6) is the majority. To migrate 6F2 to 4F2 cell is very challenging. For example, vertical cell transistor must be needed but still a couple of challenges are remaining. All in all, maintaining sufficient storage capacitance and adequate cell transistor performance are required to keep the retention time characteristic in the future. And their difficult requirements are increasing to continue the scaling of DRAM devices and to obtain the bigger product size (i.e. > 16 GB). In Fig. 3, the DRAM potential solutions are listed, but many future technologies will be necessary for 30 nm half-pitch or less and these future technologies are still unknown[5].

Non-volatile memories are used in a wide range of applications, some standalone and some embedded, with varying requirements that depend on the application. The memory array architecture and signal sensing method also differ for different applications. The technical challenges are difficult, and in some cases fundamental physics limitations may be reached before the end of the current roadmap. For charge storage devices, the number of electrons in the storage node, whether for single level logic cells (SLC) or multi-level logic cells (MLC), needs to be sufficiently high to maintain stable threshold voltage against statistical fluctuation, and cross talk between neighboring bits must be reduced while the spacing between neighbors decreases. Meanwhile, data retention and cycling endurance requirements must be maintained, and in some cases even increased for new applications. Non-charge-storage devices also may face fundamental limitations when the storage volume becomes small such that random thermal noise starts to interfere with signal. A host of nonvolatile random access memory, such as NAND Flash, NOR Flash, Phase change memory (PCRAM), Ferroelectric RAM (FeRAM) and Magnetic RAM (MRAM) are emerging at present, which would possibly provide faster and easier access mechanism, would consume lesser power, and can be embedded directly into the structure of the microprocessor or can be integrated with the functional elements of a dedicated processor.

According to the ITRS projections, embedded memories will continue to have dominating presence in the system on chips (SoC) content, which may exceed 90% of total SoC content in the next few years. It has also been found that the transistor packing density of memory devices is not only high but also increasing much faster than the transistor density of logic devices. Apart from these, the memory-based implementations are more regular compared with the multiply-accumulate structures. Memory-based computing systems have many other advantageous features from their architectural point of view, as listed in the following:

- Memory-based computing systems have potential for high-throughput and reduced-latency hardware implementation since the memory-access time is usually very much shorter compared with multiplication time.

- Memory-based designs are expected to involve much less dynamic power consumption due to minimal switching activities associated in obtaining the output product/inner product values by memory read operations.

- Apart from that, memory-based designs have a lot of scope to have flexible implementation to scale the throughput to match the temporal requirement of the applications.

From the above observations it is very much apparent that memory-based computations would have greater potential for resulting in compact and cheaper computing structures. In application specific SoCs, memory-based computing system would, therefore, be a promising alternative to the conventional logic-only implementation, where appropriate combination of logic-based arithmetic circuits and memory-based computing elements may be integrated together for dedicated implementation of DSP functionalities.

V. Conclusion

The current trends of advancement of VLSI technology indicate reasonable scope to have area-delay-power-efficient memory-based computing systems which may have potential to meet the growing requirements of the DSP systems in various application environments. Several algorithms and architectures have been suggested in the literature to reduce the area and time-complexities of commonly encountered computation intensive cores of DSP functions by memory-based computing, but many more novel algorithms and architectures need to be developed to design flexible area-delay-power-efficient systems for DSP applications of various domains. Memory elements and logic-elements can be integrated together to form more compact functional elements, and novel memory access schemes may also be explored to maximize the power efficiency and speed-performance as well.

References

[1] F. Vahid and T. Givargis, Embedded System Design : a Unified Hardware/ Software Introduction. New York: Wiley, 2002.

[2] K. K. Parhi, VLSI Digital Signal Procesing Systems: Design and Implementation.

[3] S.-M. Kuo, Real-Time Digital Signal Processing : Implementations and Applications. Hoboken, NJ: John Wiley, 2006.

[4] S. Sheng, A. Chandrakasan, and R. W. Brodersen, "A portable multimedia terminal," IEEE Communications Magazine, vol. 30, no. 12, pp. 64–75, Dec. 1992.

[5] ITRS(2012). International technology roadmap for semiconductors 2012update , tech. report. [Online]. Available: http://public.itrs.net/

[6] H. S. Hou, "A fast recursive algorithm for computing the discrete cosine transform," IEEE Trans. Acoust., Speech, and Signal Process., vol. 35, no. 10, pp. 1455–1461, Oct. 1987.

[7] M. D. Macleod and A. G. Dempster, "Multiplierless FIR filter design algorithms," IEEE Signal Processing Letters, vol. 12, no. 3, pp. 186–189,Mar. 2005.

[8] D. L. Maskell, J. Leiwo, and J. C. Patra, "The design of multiplierless FIR filters with a minimum adder step and reduced hardware complexity," in Proc. 2006 IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006, May 2006, p. 4.

[9] Y. H. Hu, "CORDIC-based VLSI architectures for digital signal processing," IEEE Signal Processing Magazine, vol. 9, no. 3, pp. 16–35, July 1992.

[10] M. Kuhlmann and K. K. Parhi, "A high-speed CORDIC algorithm and architecture for DSP applications," in 1999 IEEE Workshop on Signal Processing Systems, 1999. SiPS 99, Oct. 1999, pp. 732–741.

[11] S. A. White, "Applications of the distributed arithmetic to digital signal processing: A tutorial review," IEEE ASSP Magazine, vol. 6, no. 3, pp. 5–19, July 1989.

[12] H.-R. Lee, C.-W. Jen, and C.-M. Liu, "On the design automation of the memory-based VLSI architectures for FIR filters," IEEE Trans. Consumer Electronics, vol. 39, no. 3, pp. 619–629, Aug. 1993.

[13] J. G. McWhirter, "Algorithmic engineering in adaptive signal processing," Proc. Inst. Elect. Eng. F, vol. 139, no. 3, pp. 226–232, Jun. 1992.

[14] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Trans. Comput., vol. C-23, no. 1, pp. 90–94, Jan. 1974.

[15] H. B. Kekre and J. K. Solanki, "Comparative performance of various trigonometric unitary transforms for transform image coding," Int. J.Electron., vol. 44, no. 3, pp. 305–315, 1978.

[16] A. K. Jain, "A fast Karhunen–Loeve transform for a class of random processes," IEEE Trans. Comm., vol. COM-24, no. 10, pp. 1023–1029, Oct. 1976.

[17] Fundamentals of Digital Image Processing. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[18] K. Rose, A. Heiman, and I. Dinstein, "DCT/DST alternate-transform of image coding," IEEE Trans. Comm., vol. 38, no. 1, pp. 94–101, Jan. 1990.

[19] U. T. Koc and K. J. R. Liu, "Discrete cosine/sine transform based motion estimation," in Proc. IEEE Int. Conf. Image Processing, vol. 3, 1994, pp. 771–775.

[20] D. Sundararajan, M. O. Ahmad, and M. N. S. Swamy, "Vector computation of the discrete Fourier transform," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 45, no. 4, pp. 449–461, Apr. 1998.

[21] V. Britanak, "DCT/DST universal computational structure and its impact on VLSI design," in Proc. IEEE DSP Workshop, Hunt, TX, Oct. 15–18, 2000.

[22] L-P. Chau and W-C. Siu, "Direct formulation for the realization of discrete cosine transform using recursive filter structure," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 42, no. 1, pp. 50–52, Jan. 1995.

[23] J. F. Yang and C-P. Fang, "Compact recursive structures for discrete cosine transform," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 47, no. 4, pp. 314–321, Apr. 2000.

[24] W. H. Fang and M. L.Wu, "An efficient unified systolic architecture for the computation of discrete trigonometric transforms," in Proc. IEEE Symp. Circuits and Systems, vol. 3, 1997, pp. 2092–2095.

[25] W. H. Fang and M-L. Wu, "Unified fully-pipelined implementations of one- and two-dimensional real discrete trigonometric transforms," IEICE Trans. Fund. Electron. Commun. Comput. Sci., vol. E82-A, no. 10, pp. 2219–2230, Oct. 1999.

[26] C. M. Rader, "Discrete Fourier transform when the number of data samples is prime," Proc. IEEE, vol. 56, no. 6, pp. 1107–1108, Jun. 1968.

[27] J. Guo, C. M. Liu, and C. W. Jen, "The efficient memory-based VLSI array design for DFT and DCT," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 39, no. 10, pp. 436–442, Oct. 1992.

[28] Y. H. Chan and W. C. Siu, "On the realization of discrete cosine transform using the distributed arithmetic," IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 39, no. 9, pp. 705–712, Sep. 1992.

[29] D. F. Chiper, M. N. S. Swamy, M. O. Ahmad, and T. Stouraitis, "A systolic array architecture for the discrete sine transform," IEEE Trans. Signal Process., vol. 50, no. 9, pp. 2347–2353, Sep. 2002.

[30] H. T. Kung, "Why systolic architectures," IEEE Comp., vol. 15, pp. 37–46, Jan. 1982.

[31] S. Yu and E. E. Swartzlander, "DCT implementation with distributed arithmetic," IEEE Trans. Comp., vol. 50, no. 9, pp. 985–991, Sep. 2001.

[32] M. T. Sun et al., "VLSI implementation of a 16_16 discrete cosine transform," IEEE Trans. Circuits Syst., vol. 36, no. 4, pp. 610–617, Apr. 1989.

[33] S. Uramoto et al., "A 100 MHz 2-D discrete cosine transform core processor," IEEE J. Solid-State Circuits, vol. 27, no. 4, pp. 482–498, Apr. 1992.

[34] D. F. Chiper, "A systolic array algorithm for an efficient unified memorybased implementation of the inverse discrete cosine transform," in Proc. IEEE Conf. Image Processing, Kobe, Oct. 1999, pp. 764–768.

[35] S. B. Pan and R-H. Park, "Unified systolic array for computation of DCT/DST/DHT," IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 2, pp. 413–419, Apr. 1997.

[36] L.W. Chang andW. C.Wu, "A unified systolic array for discrete cosine and sine transforms," IEEE Trans. Signal Process., vol. 39, no. 1, pp. 192–194, Jan. 1991.

[37] J-I. Guo and C-C. Li, "A generalized architecture for the one-dimensional discrete cosine and sine transforms," IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 7, pp. 874–881, Jul. 2001.

[38] J. Guo, C. Chen, and C-W. Jen, "Unified array architecture for DCT/DST and their inverses," Electron. Lett., vol. 31, no. 21, pp. 1811–1812, 1995.

[39] D. F. Chiper, M. N. S. Swamy, M. O. Ahmad, and T. Stouraitis, “Systolic algorithms and a memory-based design approach for a unified architecture for the computation of DCT/DST/IDCT/IDST,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 6, pp. 1125–1137, Jun. 2005.

[40] C. Cheng and K. K. Parhi, “A novel systolic array structure for DCT,” IEEE Trans. Circuits Syst-II: Express Briefs, vol. 52, no. 7, pp. 366–369, July 2005.

[41] P. K. Meher, “Systolic designs for DCT using a low-complexity concurrent convolutional formulation,” IEEE Trans. Circuits & Systems for Video Technology, vol. 16, no. 9, pp. 1041–1050, Sept. 2006.

[42] P. K. Meher, J. C. Patra, and M. N. S. Swamy, “New systolic algorithm and array architecture for prime-length discrete sine transform,” IEEE Trans. Circuits Syst. II: Express Briefs, vol. 54, no. 3, pp. 262–266, Mar. 2007.

[43] P. K. Meher and M. N. S. Swamy, “High-throughput memory-based architecture for DHT using a new convolutional formulation,” IEEE Trans. Circuits Syst. II: Express Briefs, vol. 54, no. 7, pp. 606–610, July 2007.

[44] P. K. Meher, “Low-latency hardware-efficient memory-based design for large-order FIR digital filters,” To appear in Proc. Sixth International Conference on Information, Communications and Signal Processing (ICICS 2007), Dec. 2007.

[45] C.-F. Chen, “Implementing FIR filters with distributed arithmetic,” IEEE Trans. Acoustics, Speech, and Signal Processing, vol. 33, no. 5, pp. 1318–1321, Oct. 1985.

[46] K. Nourji and N. Demassieux, “Optimal VLSI architecture for distributed arithmetic-based algorithms,” in 1994 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1994. ICASSP-94, vol. 2, Apr. 1994, pp. II/509–II/512.

[47] M. Mehendale, S. D. Sherlekar, and G. Venkatesh, “Area-delay tradeoff in distributed arithmetic based implementation of FIR filters,” in Proc. Tenth International Conference on VLSI Design, Jan. 1997, pp. 124–129.

[48] S.-S. Jeng, H.-C. Lin, and S.-M. Chang, “FPGA implementation of FIR filter using M-bit parallel distributed arithmetic,” in Proc. 2006 IEEE International Symposium on Circuits and Systems, ISCAS 2006, May 2006, p. 4.

[49] J. P. Choi, S.-C. Shin, and J.-G. Chung, “Efficient ROM size reduction for distributed arithmetic,” in Proc. IEEE International Symp on Circuits and Syst., 2000. ISCAS, vol. 2, May 2000, pp. 61–64.

[50] H.-C. Chen, J.-I. Guo, T.-S. Chang, and C.-W. Jen, “A memory-efficient realization of cyclic convolution and its application to discrete cosine transform,” IEEE Trans. Circuits Syst for Video Technol., vol. 15, no. 3, pp. 445–453, Mar. 2005.

[51] P. K. Meher, “Hardware-efficient systolization of DA-based calculation of finite digital convolution,” IEEE Trans. Circuits Syst. II: Express Briefs, vol. 53, no. 8, pp. 707–711, Aug. 2006.

[52] H. Yoo and D. V. Anderson, “Hardware-efficient distributed arithmetic architecture for high-order digital filters,” in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP ’05), vol. 5, Mar. 2005, pp. v/125–v/128.

[53] S. Hwang, G. Han, S. Kang, and J. Kim, “New distributed arithmetic algorithm for low-power FIR filter implementation,” IEEE Signal Processing Letters, vol. 11, no. 5, pp. 463–466, May 2004.

[54] Y.-T. Hwang and C.-L. Su, “Parallel and pipelined architecture designs for distributed arithmetic-based recursive digital filters,” in Proc. Workshop on IX, VLSI Signal Processing, Oct-Nov 1996, pp. 35–44.

[55] P. K. Meher, T. Srikanthan, and J. C. Patra, “Scalable and modular memory-based systolic architectures for discrete Hartley transform,” IEEE Trans. Circuits Syst. I: Regualr Papers, vol. 53, no. 5, pp. 1065-1077, May 2006.

[56] P. K. Meher, “Unified systolic-like architecture for DCT and DST using distributed arithmetic,” IEEE Trans. Circuits Syst. I: Regualr Papers, vol. 53, no. 5, pp. 2656–2663, Dec. 2006.

[57] M. Alam, C. A. Rahman, W. Badawy, and G. Jullien, “Efficient distributed arithmetic based DWT architecture for multimedia applications,” in Proc. The 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications, June-July 2003, pp. 333–336.

[58] Z. Abid, W. Wang, and Y. Chen, “Low-power fpga implementation for da-based video processing,” in Proc. 2005 IEEE International Workshop on VLSI Design and Video Technology, May 2005, pp. 361–364.

[59] D. J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson, “Lms adaptive filters using distributed arithmetic for high throughput,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 52, no. 7, pp. 1327–1337, July 2005

[60] S. K. Mitra, Digital Signal Processing : a Computer Based Approach. Boston: McGraw-Hill, 2006.

[61] J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications. Upper Saddle River, NJ: Prentice-Hall, 1996.

[62] B. Prince, “Trends in scaled and nanotechnology memories,” in Proc. IEEE 2004 Conference on Custom Integrated Circuits, Nov. 2005, p. 7.

[63] K. Itoh, S. Kimura, and T. Sakata, “VLSI memory technology: Current status and future trends,” in Proc. 25th European Solid-State Circuits Conference,ESSCIRC ’99, Sept. 1999, pp. 3–10.

[64] T. Furuyama, “Trends and challenges of large scale embedded memories,” in Proc. IEEE 2004 Conference on Custom Integrated Circuits, Oct. 2004, pp. 449–456.

[65] D. G. Elliott, M. Stumm, W. M. Snelgrove, C. Cojocaru, and R. Mckenzie, “Computational RAM: implementing processors in memory,” IEEE Trans. Design & Test of Computers, vol. 16, no. 1, pp. 32–41, Jan-Mar,1999.