

## Research Article

# A Reconfigurable Systolic Array Architecture for Multicarrier Wireless and Multirate Applications

H. Ho,<sup>1</sup> V. Szwarc,<sup>1</sup> and T. Kwasniewski<sup>2</sup>

<sup>1</sup>Terrestrial Wireless Systems, Communications Research Centre, 3701 Carling Avenue, BOX 11490, Station H, Ottawa, ON, Canada K2H 8S2

<sup>2</sup>Department of Electronics, Carleton University, 1125 Colonel By Drive, Ottawa, ON, Canada K1S 5B6

Correspondence should be addressed to H. Ho, huong.ho@crc.ca

Received 26 November 2008; Accepted 23 June 2009

Recommended by Neil Bergmann

A reconfigurable systolic array (RSA) architecture that supports the realization of DSP functions for multicarrier wireless and multirate applications is presented. The RSA consists of coarse-grained processing elements that can be configured as complex DSP functions that are the basic building blocks of Polyphase-FIR filters, phase shifters, DFTs, and Polyphase-DFT circuits. The homogeneous characteristic of the RSA architecture, where each reconfigurable processing element (PE) cell is connected to its nearest neighbors via configurable switch (SW) elements, enables array expansion for parallel processing and facilitates time sharing computation of high-throughput data by individual PEs. For DFT circuit configurations, an algorithmic optimization technique has been employed to reduce the overall number of vector-matrix products to be mapped on the RSA. The hardware complexity and throughput of the RSA-based DFT structures have been evaluated and compared against several conventional modular FFT realizations. Designs and circuit implementations of the PE cell and several RSAs configured as DFT and Polyphase filter circuits are also presented. The RSA architecture offers significant flexibility and computational capacity for applications that require real time reconfiguration and high-density computing.

Copyright © 2009 H. Ho et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. Introduction

In multicarrier wireless and multirate applications, DFT and Polyphase filters are two key functions frequently employed in the processing of digital signals. Orthogonal Frequency Division Multiplexing (OFDM) utilized in numerous multicarrier communication systems including Digital Audio/Video Broadcasting (DAB/DVB), personal area networks (PAN), and mobile communications [1–5] depends on the efficient realization of the DFT. Furthermore, hardware solutions that offer computational flexibility of  $N$ -point DFTs for OFDM modulation-demodulation of subcarriers enable the systems to operate in a multimode and multistandard environment. For example, the transmission modes of DAB involve FFT computations of 2048, 1024, 512, or 256 subcarriers while DVB-T uses 2K and 8K FFTs for data modulation/demodulation. On the other hand, dedicated DFT hardware solutions for OFDM modulators, where the number of subcarriers is not a power of two,

have been considered such as the processor for the Digital Multimedia Broadcast standard (DMB-T) described in [6]. A solution to the multimode multistandard problem could be realized through the implementation of reconfigurable  $N$  point DFT/IDFT processors, where  $N$  is not restricted to be a power of two.

Polyphase filters have been effectively exploited in a broad range of applications including digital signal interpolation and decimation for speech and image processing as well as multicarrier communications [7–11]. The combination of Polyphase Filter banks with FFTs has resulted in computationally efficient Group Demultiplexer realizations [10]. Multirate signal processing or communication applications incorporating polyphase filter banks frequently need to be reconfigurable [12, 13].

For software defined radio (SDR) [14] applications, digital signal processing is usually carried out using a general purpose processor or digital signal processor (DSP). These processors allow the loading of different software for

different system configurations. The need to handle signals with different data rates and protocols as well as multicarrier communications points to the desirability of incorporating hardware reconfigurability in SDR platforms.

Multicarrier processors based on Polyphase-DFT and IDFT-Polyphase architectures offer a computationally effective solution to the multiplexing and demultiplexing of the composite FDM carriers. The application of such multicarrier processing across a frequency band of interest could provide a potential solution to the problem of dynamic channel selection for SDR. Thus, a flexible hardware platform where real time reconfiguration of Polyphase-DFT and IDFT-Polyphase functions is supported may provide a computationally effective solution for high data rate SDR applications as compared to one based on multiple processors.

FFT-based algorithms have been widely used to realize hardware efficient high-throughput DFT circuits [15–18]. However, hardware FFT implementations are typically difficult to reconfigure due to the problems associated with signal routing between successive butterfly stages.

Several flexible FFT architectures have been proposed for the processing of  $N$  sample input sequences where  $N$  is a power of two [19, 20]. The main feature of these architectures is that the circuits can be extended to support different input sequence sizes. The architecture proposed in [19] consists of a number of identical processing elements connected in a pipeline fashion. This architecture offers some flexibility in terms of configurability, but it requires significant logic resources, and the throughput is limited by the technology constraints inherent in a serial I/O design. The architecture proposed in [20] is modular and consists of identical pipelined butterfly building blocks whose number depends on the size of the transform. However, connections among the butterfly modules include feedback and long signal links, making reconfiguration difficult.

Other systolic array architectures have also been proposed for the computation of  $N$ -point DFTs, where  $N$  is a composite number consisting of two or more cofactors [21–24]. The architectures proposed in [22, 23] offer an efficient way to calculate  $N$ -point DFTs based on an  $N_1$ -by- $N_2$  array where  $N$  is a product of two numbers ( $N = N_1 * N_2$ ). The base-4 algorithm proposed in [21] is implemented in systolic array architecture that offers an efficient way to compute  $N$ -point DFTs. The base-4 algorithm supports DFT computations for  $N$  input sequences where  $N$  is a multiple of 256. The systolic array proposed in [24] supports DFT computation for  $N$ -point DFT where  $N = M^2$ . These architectures share a common characteristic, namely, that the input and output data sequences are reordered throughout the computation process.

DFT circuit designs reflecting vector-matrix multiplication result in modular structures that facilitate circuit reconfiguration and can be realized for input sequences of arbitrary length. However, circuit implementation based on direct DFT computation requires  $N$  vector multiplications, each of which consists of  $N$  complex multiplications. To reduce hardware requirements for DFT circuit implementations, systolic array designs based on Horner's rule have also been proposed [25, 26]. The limitation of all of the

above mentioned systolic array architectures is that they have been designed specifically for DFT/FFT computations and do not support other functions. Reconfigurable systolic array architectures that support different DSP functions including DFT/FFT have been documented in [27–29] and will be considered in the context of current work in later sections.

In this paper, a scalable systolic array architecture which can be configured to function as an FIR filter, Filter bank, phase shifter, DFT/IDFT, and Polyphase-DFT circuit is presented. The architecture offers computational flexibility and allows DSP functions to be executed in parallel, in pipelined configurations, or, alternately, the DSP function can be partitioned and processed sequentially by shared elements of the systolic array.

For DFT configurations, the RSA architecture supports processing of  $N$  input sequences of arbitrary length. To reduce hardware requirements for the  $N$  point DFT implementation, the sum of complex products inherent in the computation is regrouped so as to reduce the number of complex multiplications at the cost of increasing the number of additions. The arithmetic configuration required to compute a partial sum of complex products can be mapped on a single PE cell. Circuit implementations based on a combination of common subexpression and bit-slice arithmetic (CSE-BitSlice) techniques have also been incorporated in the PE design to achieve low-complexity and high-throughput performance [30].

The paper is organized as follows. In Section 2, systolic array architecture capable of supporting Polyphase-FIR filtering and DFT computations is presented. Mapping techniques aimed at achieving low-complexity DFT processors on the systolic arrays are also outlined. Configurations for parallel and sequential DFT processor implementations are also presented in this section. In Section 3 complex multiplication requirements are considered for DFTs with  $N = 4n$ ,  $N = 4n+2$  or  $N = 2n+1$  using the proposed factorization approach to achieve hardware reduction. Section 4 presents a comparison of hardware complexity and performance for DFT circuits mapped on the RSA and several modular FFT structures reported in literature. Section 5 deals with circuit design, FPGA implementation, and simulation results for the RSA, the PE cell, and representative RSA circuits configured to implement DFT processors and polyphase filters. In Section 6, the RSA is evaluated and compared to representative configurable systolic arrays reported in literature in terms of flexibility and performance. Summary and conclusions are presented in Section 7.

## 2. Systolic Array Architecture for DFT and Polyphase Filtering Computations

The DFT and IDFT can be expressed as follows:

$$Y_{\text{DFT}}(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}, \quad k = 0, \dots, N-1, \quad (1)$$

$$y_{\text{IDFT}}(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi nk/N}, \quad n = 0, \dots, N-1. \quad (2)$$

As can be seen from (1) and (2), each output depends on the sum of partial products generated by the multiplication of the input sequence terms and the known coefficients. Direct DFT /IDFT computation requires a total of  $N^2$  complex multiplications and  $N(N - 1)$  complex additions. For large values of  $N$ , the number of multiplications and additions required for both DFT and IDFT can be rather substantial. However, the number of complex arithmetic operations can be reduced by the application of a factorization and term aggregation technique. For each output, the factorization process first identifies terms that multiply a common coefficient. These terms are then summed together, and the result is multiplied by the common coefficient. This process is repeated and the partial products are accumulated to produce the individual outputs. Furthermore, due to the property of the DFT algorithm, the individual partial products can be used to determine the partial results of one or three other outputs depending upon whether  $N$  is odd or even. The factorization technique can be repeated to generate all  $N$  outputs of the  $N$ -point DFT/IDFT. Thus it is possible to significantly reduce the number of complex multiplications required to compute DFT/IDFT as indicated in some detail in Appendices A and B for even and odd values of  $N$ , respectively. Indeed, a total of only  $MK$  complex multiplications are required to compute an  $N$ -point DFT where the parameter  $M$  is equal to  $N/4$ ,  $(N + 2)/4$ , or  $(N - 1)/2$  and parameter  $K$  is equal to  $N/4$ ,  $N/2$ , or  $(N - 1)/2$  for  $N$  a multiple of four,  $N$  even, and  $N$  odd, respectively. This factorization approach offers the advantage of reducing the overall number of complex multiplications without resorting to data recursion. The data routing and aggregation inherent in this technique make it particularly suitable for a systolic array implementation incorporating circuit scaling, expansion, and reconfiguration features.

A  $T$ -tap FIR filter is described by:

$$y_{\text{FIR}}(n) = \sum_{i=0}^{T-1} x(n-i)a(i). \quad (3)$$

The output can be alternately defined as the product of two vectors, namely, an input vector consisting of the  $x(n-i)$  data samples and a vector reflecting the  $a(i)$  filter coefficients. Thus an array of  $T$  multipliers and  $T - 1$  adders could be employed to meet the computational requirements of a  $T$ -tap FIR filter.

The DFT, on the other hand, corresponds to the product of the input data vector and a matrix consisting of the DFT coefficients. For a DFT realization, an  $M$ -by- $K$  array structure consisting of  $MK$  complex multipliers is required to compute an  $N$ -point DFT. Figure 1 depicts a systolic array of processing elements (PE) that can be used for parallel computation of an  $N$ -point DFT based on the expressions in Appendices A and B. This array provides an  $N$  output sequence at every clock cycle. Each processing element in this array structure can simultaneously compute four real multiplications. If  $N$  is a multiple of four, each set of four complex input data shifted into the PE is combined and the partial result is multiplied by the factored coefficient. The set of four partial results generated is then added to

the results shifted in from the PE cell in the previous row. As shown in Figure 1, each PE cell in the first row of the array combines the set of input data  $X(0)$ ,  $X(N/4)$ ,  $X(N/2)$ , and  $X(3N/4)$  based on the expression  $S_1 + (-j)^k S_2$  described in (A.3) in the Appendix A where  $S_1$  and  $S_2$  are defined in (A.4), respectively. Similarly, each PE cell in row  $n$  ( $n = 1, \dots, N/4 - 1$ ) combines the set of input data  $X(n)$ ,  $X((N/2) - n)$ ,  $X(N/2 + n)$ , and  $X(N - n)$ . The sum or difference generated is then multiplied by the real and imaginary part of the factored coefficients represented by  $a(n)$  and  $b(n)$  as described in (A.3). The partial products generated from the multiplication process are added to results shifted in from PE cells in previous row. The partial products generated for  $Y(k)$  can be used to calculate for partial results of  $Y(N/2 - k)$ ,  $Y(N/2 + k)$  and  $Y(N - k)$  simultaneously since all four multiplications involve the same expressions and coefficients as shown in the second part of (A.3) and (A.5). Thus, the PE computes the multiplication-accumulation on a set of four input data and produces a set of four partial results based on one complex multiplication.

If  $N$  is not a multiple of four, each PE produces two partial results based on the multiplication of the factored coefficient by the sum generated from the set of complex input data. If  $N$  is even, the set of input data consists of two or four complex data as described in (A.8) and (A.9) in Appendix A. If  $N$  is odd, a set of two complex input data is shifted into each PE as shown in (B.2) and (B.4) in Appendix B.

Since each PE cell is connected to its immediate neighbors, the systolic array structure in Figure 1 supports DFT computation of  $N$  input sequences where data sequences are sequentially shifted in. There are two possible mapping schemes for sequential computation of an  $N$ -point DFT/IDFT based on the systolic array architecture depicted in Figure 1. The block diagram in Figure 2 depicts a parallel input-serial output mapping scheme where all  $N$  input sequences are shifted in simultaneously, and a maximum of 4 outputs is shifted out at every clock cycle. Based on this configuration, an  $N$  output sequence is shifted out in  $N/4$  clock cycles if  $N$  is a multiple of four. If  $N$  is not a multiple of four, the set of  $N$  output sequence is shifted out in  $N/2$  clock cycles.

The mapping shown in Figure 3 depicts a sequential synchronous structure where the inputs and outputs operate off the master clock. This serial input-parallel output mapping scheme shifts 4 input sequences into the processing elements at every clock cycle where partial results are combined with data generated in previous clock cycles. All  $N$  output sequences are available in  $N/4$  or  $N/2$  clock cycles for the cases of  $N$  even and odd, respectively.

The sequential structures in Figures 2 and 3 could provide higher data throughput if the dimension of the array is expanded. If the parallel inputs-serial outputs structure is consisting of  $M$  columns of PE cells, it would shift out the complete set of  $N$  output sequences after  $(N/4)M$  clock cycles. Similarly, the serial inputs-parallel outputs structure would shift out all  $N$  output sequences after  $(N/4)K$  clock cycles, based on  $K$  rows of PE cells.

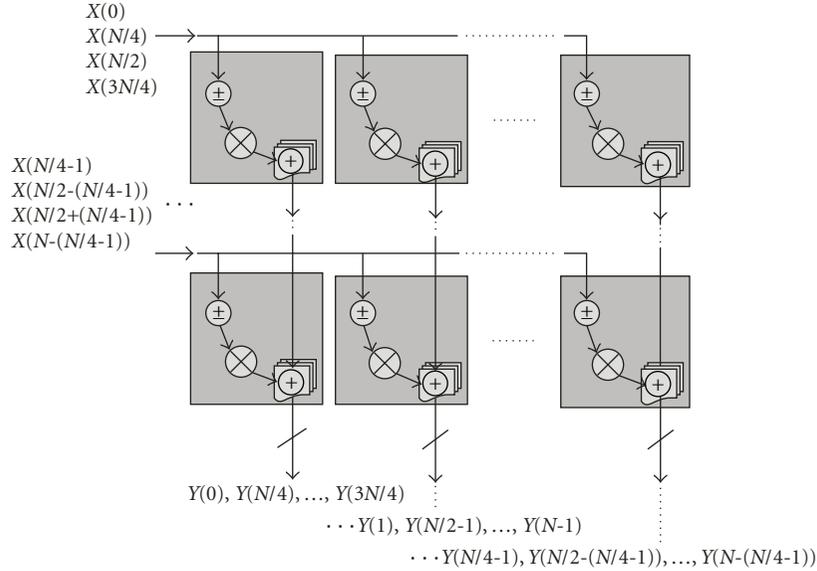


FIGURE 1: Systolic array structure for parallel computation of an  $N$ -point DFT.

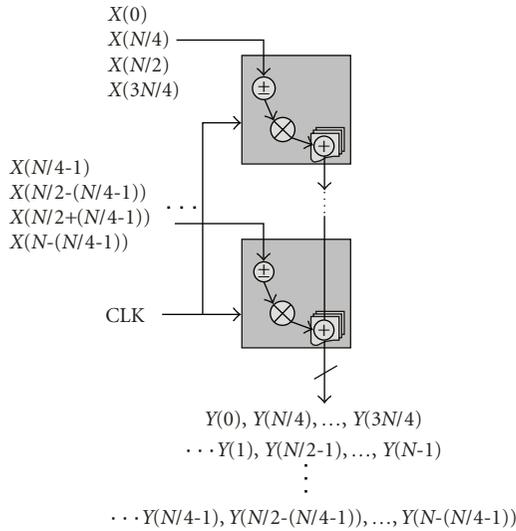


FIGURE 2: DFT processor: Parallel in-Serial out configuration.

A systolic array structure consisting of a  $1$ -by- $T$  array of PE cells could be used to realize a polyphase filter bank consisting of four  $T$ -tap FIR filters. Figure 4 depicts the mapping of such a filter bank onto an array structure consisting of  $T$  PE cells cascaded together. The input and output signals shown in Figures 2 to 4 are all consistent with the complex data format.

Since each PE cell in the systolic array contains four real multipliers, it could also be used to realize a phase shifter.

### 3. Estimation of DFT Complexity

In Appendices A and B it is shown that, by factoring and aggregating terms that multiply a common coefficient, the

total number of complex multiplications required for an  $N$ -point DFT can be reduced significantly. For  $N = 4n$ , the four output sequences  $y(k)$ ,  $y(N/2 - k)$ ,  $y(N/2 + k)$ , and  $y(N - k)$  for  $k = 1, 2, \dots, (N/4) - 1$  can be calculated concurrently using only  $N/4$  complex multiplications. Thus, for parallel computation, the total number of complex multiplications required to compute all  $N$  output sequences where  $N = 4n$  is  $(N/4)^2$ . For  $N = 4n + 2$  and  $N = 2n + 1$ , each set of output sequences  $y(k)$  and  $y(N - k)$  can be calculated concurrently with  $(N+2)/4$  and  $(N-1)/2$  complex multiplications, respectively. Thus, the total number of complex multiplications required is  $((N+2)/4) * (N/2)$  and  $((N-1)/2)^2$  for DFT computation of input sequences for which  $N = 4n + 2$  and  $N = 2n + 1$ , respectively. The total number of complex operations required to compute an  $N$  point DFT/IDFT is summarized in Table 1. From Table 1 we see that for the three cases presented the smallest number of complex multiplications occurs when  $N$  is a multiple of four.

### 4. Comparison of Complexity and Throughput of RSA and Reconfigurable DFT/FFT Architectures

In this section, the number of complex multipliers contained in representative reconfigurable DFT/FFT structures is evaluated and compared with a DFT circuit configured on the RSA architecture. The comparison focuses on complex multipliers as they require significantly more hardware resources than adders. Throughput performance and DFT dimensionality constraints are also considered for each structure. The RSA structure used in this comparison has serial inputs-parallel outputs architecture similar to the one depicted in Figure 3. Table 2 presents throughput and complex multiplier requirements for the RSA and for circuit

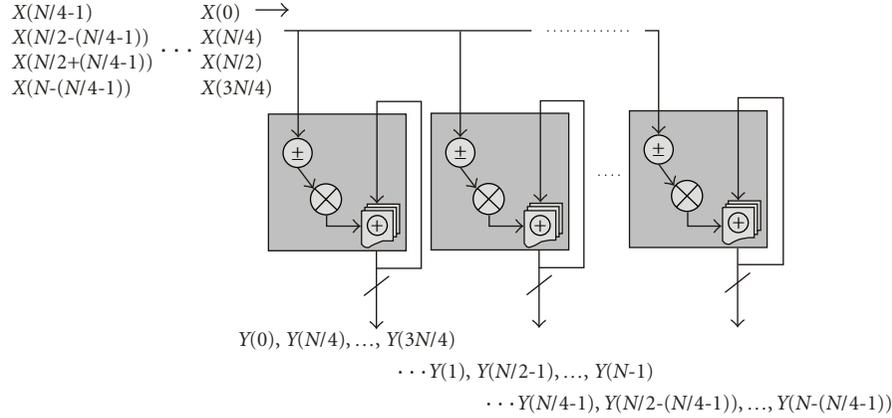
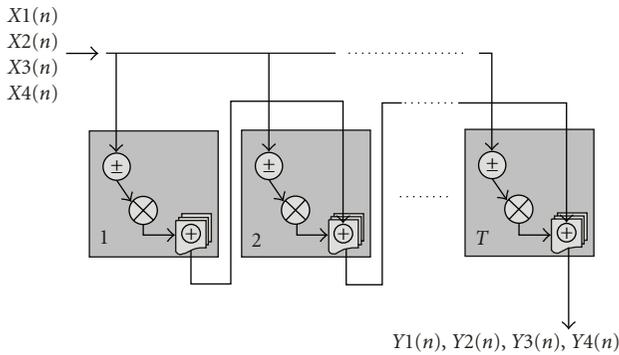


FIGURE 3: DFT Processor: Serial in-Parallel out configuration.

TABLE 1: Number of complex operations for DFT/IDFT based on proposed factorization technique.

Arithmetic Operations	$N = 4n$	$N = 4n + 2$	$N = 2n + 1$
Complex multiplier	$\left(\frac{N}{4}\right)^2$	$\frac{(N+2)}{4} * \left(\frac{N}{2}\right)$	$\left(\frac{(N-1)}{2}\right)^2$
Complex addition	$\left(\left(\frac{3N}{2}\right) + 6\right) \left(\frac{N}{4}\right)$	$\left(\left(\frac{3N}{2}\right) - 1\right) \left(\frac{N}{2}\right)$	$2N \left(\frac{N-1}{2}\right)$


 FIGURE 4: Systolic array architecture for a polyphase filter bank consisted of four  $T$ -tap FIR filters.

designs based on a 2D-Systolic array [24], base-4 [21], the RADComm [28], the CSoC [29], and the modular architectures presented in [19]. However, only the RADComm and RSA circuits can support DFT configurations for which  $N$  is an odd number. Throughput timing is defined as the number of clock cycles between corresponding outputs of any two sequential  $N$  point DFTs.

Figures 5 and 6 illustrate the number of clock cycles per  $N$ -point DFT computation and the number of complex multipliers required by the RSA and representative DFT implementations identified in Table 2. From Figures 5(a) and 5(b) it is apparent that the RSA-based DFT requires more complex multipliers than the base-4. However, it provides higher throughput. Furthermore, the number of complex multipliers required by the RSA circuit is a quarter

of the number in a 2D-systolic structure for any value of  $N$ . However, the latter provides a higher throughput for DFTs for which  $N$  is greater than 256. Thus, the RSA is more compact and provides a higher throughput than the 2D-systolic array structure for  $N$  smaller than 256. The Modular architecture on the other hand requires more complex multipliers and has a lower throughput than the RSA. Moreover, the base-4, the 2D-systolic, and the Modular structures impose specific restrictions on the dimensionality of  $N$ . Specifically, for the 2D systolic and the Modular architectures,  $N$  is limited to values that are a power of two, whereas, for the base-4 architecture  $N = 256n$ . For purpose of this comparison, the RSA was configured to support DFTs for which  $N$  is a power of 2 or a multiple of 4.

In contrast, the RADComm circuit supports DFT implementations for  $N$  even or odd. However, in comparison with the RSA, it requires four times the number of multipliers if  $N = 4n$  and twice as many otherwise. Furthermore, for  $N$  even, the RADComm circuit has one fourth the throughput of the RSA and, for  $N$  odd, the throughput is one half. Figure 6 provides throughput results and hardware requirements in terms of the number of complex multipliers for the RSA and the RADComm circuit.

The CSoC circuit incorporates 16 PE cells per cluster together with fixed arithmetic blocks, which can be configured to perform 4 complex butterfly computations. For a 256-point FFT, approximately 500 clock cycles are needed to perform the complex butterfly computations. Indeed, for FFTs with input sequences for which  $N > 256$  the throughput is considerably lower than that for the other implementations identified in Table 2.

TABLE 2: Throughput and complex multiplier requirements for reconfigurable DFT implementations.

Architecture	Number of complex Multipliers	Throughput: Clk. Cycles per $N$ point DFT	Constraints on $N$
2D Systolic [24]	$N$	$N_3 + N_4 + 1$	$N = N_3 * N_4 (N_3 * N_4 = 2^n)$
Base-4 [21]	$\frac{N_3}{4}$	$\frac{N}{4} + \frac{N_4^2}{4} + 4N_4 + 28$	$N = N_3 * N_4 (N_3 * N_4 = 256n)$
Modular [19]	$\frac{N}{2}$	$N$	$N = 2^n$
CSoc [29]	4	$2\left(\frac{N}{8}\right) \log 2^n$	$N = 2^n$
RADComm [28]	$N$	$N$	$N$
RSA [Proposed]	$\frac{N}{4}$	$\frac{N}{4}$	$N = 4n$
	$\frac{(N+2)}{4}$	$\frac{N}{2}$	$N = 4n + 2$
	$\frac{(N-1)}{2}$	$\frac{(N-1)}{2}$	$N = 2n + 1$

## 5. RSA Circuit Design, Implementation, and Simulation Results

The RSA circuit design is based on systolic array architecture consisting of PEs interconnected via SWs as depicted in Figure 7. Both the PEs and SWs can be reconfigured dynamically with the former as an arithmetic processor and the latter as a flexible router linking the neighboring PE cells. The RSA shifts reconfiguration and input signals into the PEs and SWs on separate data bus which enables the circuit to continue its operation while the reconfiguration is in process.

**5.1. The Processing Element (PE) Cell.** The PE cell incorporates arithmetic circuits that can be configured to perform a combination of addition and multiplication operations needed by the RSA to execute a broad range of DSP functions, that is, Polyphase-FIR filter, phase shifter, DFT, IDFT, Group Demultiplexer (Polyphase-DFT), and Group Multiplexer (IDFT-Polyphase). The PE consists of three main building blocks: the arithmetic unit (AU), the control unit (CU), and the memory unit (MU). The input signals shifted into the PE cells via eight data buses identified as  $IN_1$  to  $IN_8$  are processed by the AU, and the configuration data is loaded via the CONF bus as indicated in Figure 8. Configuration data consists of control information, saved in the CU, that dictates the PE mode of operation and the MU stored coefficients required by the AU to execute arithmetic computations. The input data is in complex format while the configuration data is in real format.

The eight data buses are partitioned into two sets,  $IN_1$  to  $IN_3$  and  $IN_4$  to  $IN_8$ . Input data routed over the first set is typically multiplied by the stored coefficients while the partial results available from other cells are shifted in via the second set. These partial results are then combined with the results generated by the multiplication process to produce the PE's output at the four output ports depicted in Figure 8.

For DFT implementations for which the data block of length  $N$  is even, input data is partitioned into four input sequences and shifted into the PE cell via the first set of

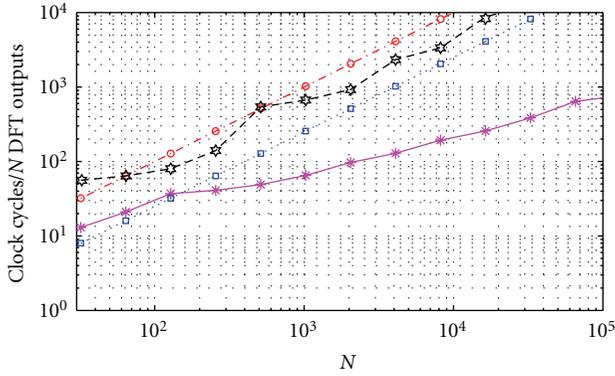
input buses. The results are shifted out as two or four output sequences. Alternately, for the case that  $N$  is odd, the input data and output results consist of two sequences that are clocked in/out synchronously.

For Polyphase/FIR filtering computation, two complex input and output sequences are shifted in and out of the cell at every clock cycle. Partial results from previous cells are shifted over the second input bus of the PE cells and are combined with the partial products generated by the multiplication of input data and the filter coefficients. For phase shifting computation, only one complex input and one complex output sequence need to be shifted in and out of the cell at every clock cycle.

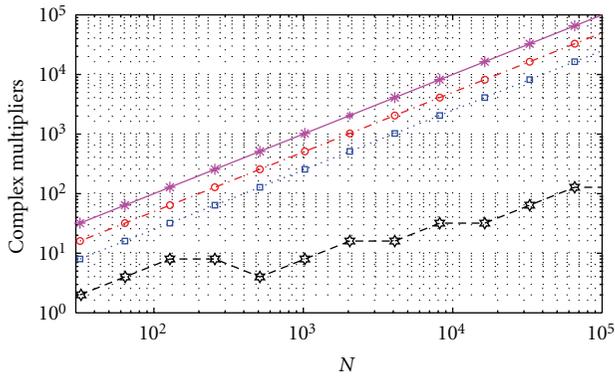
The MU incorporates a block of RAM where coefficients are stored while the CU controls data flow among the sub-blocks within the AU based on the mode of operation selected. A total of nine modes of operation are supported by the CU, which can be configured in real time via the configuration data bus. The AU which contains the PE's arithmetic functions interfaces with MU and CU as depicted in Figure 9.

**5.1.1. The Memory Unit (MU).** The MU is a storage unit where a total of four sets of coefficients, each of which consists of two 9-bit words, can be stored. The four sets of coefficients allow the AU to perform sequential calculations on four different sets of input data. For DFT algorithms requiring a parallel rather than a sequential signal processing implementation, only one set of coefficients is forwarded to the AU. The coefficients can be loaded into the MU or updated via the configuration data bus in real time.

**5.1.2. The Arithmetic Unit (AU).** Figure 10 depicts a block diagram of the AU architecture where each of the multiplier blocks performs two real multiplications involving complex data and the real or quadrature part of the coefficient. Thus, a total of four real multiplication operations can be performed by the PE cell.



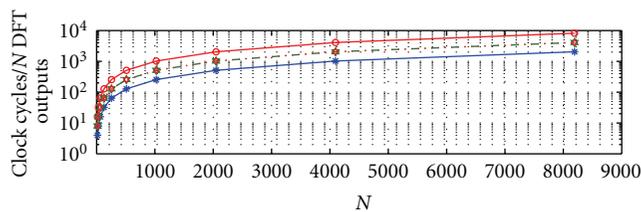
(a)



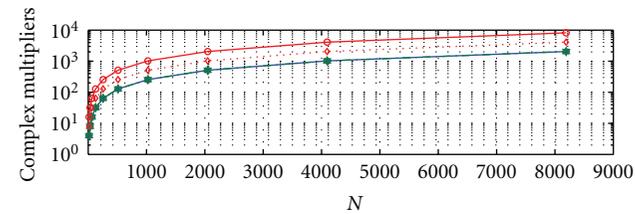
-○- Modular      -\* 2D-syst  
-★- Base-4      -□- RSA

(b)

FIGURE 5: (a) Throughput of RSA, Modular, 2D Systolic, and base-4 based DFT implementations. (b) Complex multiplier requirements of RSA, Modular, 2D Systolic, and base-4-based DFT implementations.



(a)



-○- RADComm      -★- RSA ( $N = 4n + 2$ )  
-□- RSA ( $N = 4n$ )      -◇- RSA ( $N = 2n + 1$ )

(b)

FIGURE 6: Throughput and complex multiplier requirements for the RSA and RADComm implementations.

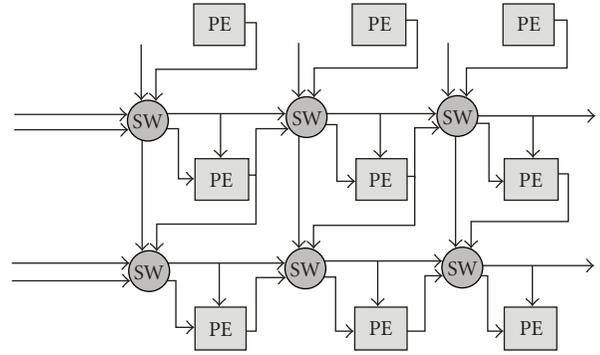


FIGURE 7: Block diagram of the RSA architecture.

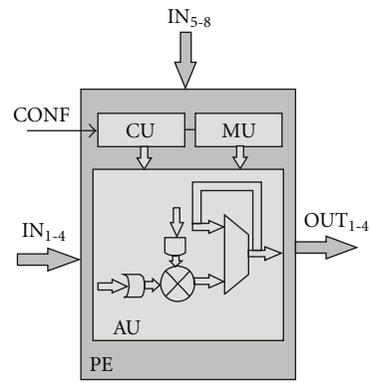


FIGURE 8: Architecture of the PE cell.

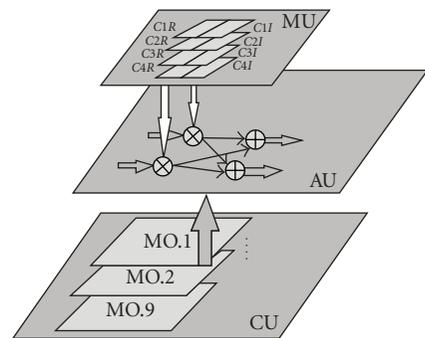


FIGURE 9: Interfaces between the MU, AU, and CU blocks.

The AU can be configured to support several modes of operation, including complex and real multiplication, addition, or a combination of these arithmetic operations. Complex multiplication-accumulation is required for DFT/IDFT implementations while complex multiplication and real multiplication-accumulation are needed to realize phase shifting and FIR filtering, respectively. The complex multiplication-accumulation configuration is shown in Figure 11 while Figures 12 and 13 depict complex multiplication and real multiplication-accumulation respectively.

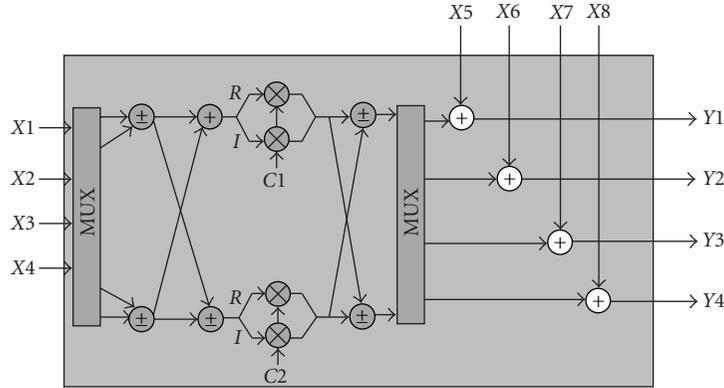


FIGURE 10: Block diagram of the AU architecture.

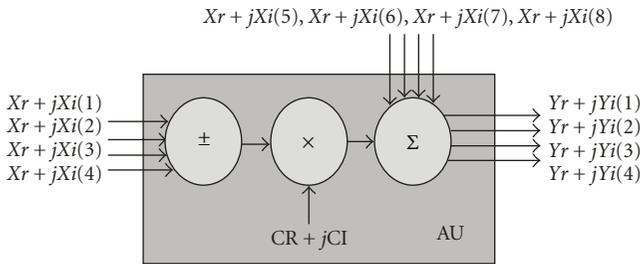


FIGURE 11: AU configuration for complex multiplication-accumulation.

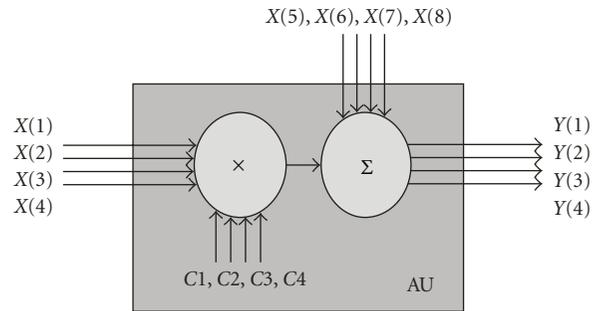


FIGURE 13: AU configuration for real multiplication-accumulation.

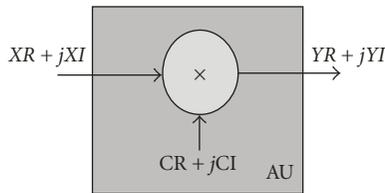


FIGURE 12: AU configuration for complex multiplication.

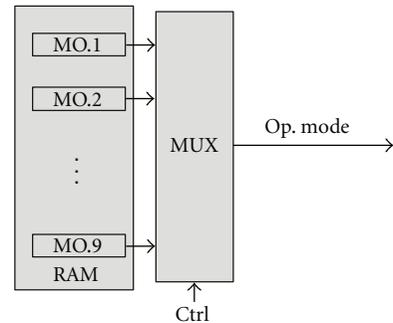


FIGURE 14: Block diagram of the control block in the CU.

5.1.3. *The Control Unit (CU).* The CU, depicted in Figure 14, is responsible for the PE modes of operation, which are loaded in via the configuration data bus. It multiplexes the PE cell's three clock signals, namely,  $clk$ ,  $clk2$ , and  $clk4$  to synchronize the input data and coefficients based on parallel or sequential mode of computation. The clock signal  $clk2$  has a frequency twice the frequency of  $clk$  and  $clk4$  has a frequency four times the frequency of the  $clk$  signal. If the circuit is configured to compute in sequential mode, the data input is controlled by the  $clk$  signal while the computation inside the AU is controlled by the signal  $clk2$  if two complex multiplications are performed, or  $clk4$  if four complex multiplications are performed. The CU incorporates a multiplexer and a RAM block that supports nine modes of operation MO.1 to MO.9, which dictate the flow of signals in the AU in accordance with the identified target function. The first five modes of operation MO.1 to MO.5 are allocated for DFT configurations for  $N = 8n$  ( $n$ : even or odd),  $N$  is a multiple of four,  $N$  is even, and  $N$  is odd,

respectively. The next two modes of operation are allocated for phase shifters configured to operate in sequential fashion where two (MO.6) or four (MO.7) complex multiplications are performed. The modes of operation MO.8 and MO.9 are reserved for FIR filters configured to operate in parallel and sequential modes, respectively.

5.2. *The Switch (SW).* The switches route signals from the RSA's inputs to the PE cells, between the PE cells, and from the PE cells to the RSA's output [31]. In this section, primary signals are defined as signals at the input and output of the RSA circuit as well as signals at the output of SWs that are routed to other SW cells as indicated in Figure 15. Secondary signals are defined as signals that originate at the output of

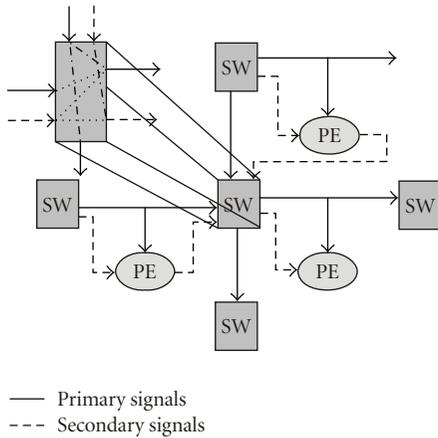


FIGURE 15: Connections between SW and PE cells.

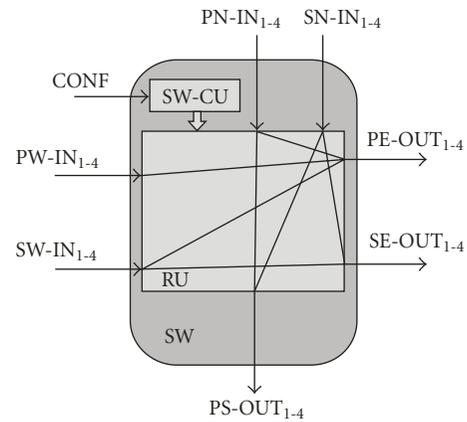


FIGURE 16: Block diagram of the SW.

a PE cell and connect to the adjacent SW. Similarly, signals that originate at the output of an SW and connect to the adjacent PE cell are defined as secondary signals. Each SW interfaces with four neighboring SWs and three nearby PE cells. The SW shifts primary signals out to its neighboring SWs on its east side and south side. It also routes the primary and secondary signals to the PE cell on its east side. The primary signals routed to neighboring PE cells contain data to be further operated on by the PE's CU. The secondary signals shifted to these PE cells are partial results from the previous stage to be combined with results generated by the multiplication process. Figure 15 depicts interconnections between an SW and its neighboring SW and PE cells.

The SW has two types of input interfaces, namely, data and control. The first, which consists of sixteen data buses in complex format, supports data that is shifted in from system inputs or data that is shifted out of the neighboring PE cell. The sixteen data buses are grouped into two sets of input data representing the primary and secondary signals. The primary input data consists of eight, 24 bit data signals of which four are shifted in to the SW from the north side and the other four are shifted in from the west side. The secondary input data consists of eight, 34 bit data signals of which four are shifted in to the SW from the west side and the other four are coming from the north side of the SW. The control input interface is dedicated to reconfiguration data that determines the modes of operation for the SW. Reconfiguration data can be loaded in to the switch in real time via the CONF bus. Signal outputs are shifted out of the SW via twelve data buses represented by two sets of primary and one set of secondary data buses, all in complex format.

As shown in Figure 16, each set of the primary output buses consists of four data signals that are shifted out to PE cells to the east and to the south side of the SW. The set of secondary buses consists of four data signals that are shifted out to the PE cell on the east side of the SW.

The SW consists of two building blocks: the routing unit (RU) and the control unit (SW-CU). The SW-CU contains control data that is used to direct the flow of signals from the inputs to the outputs of the switch. The SW-CU controls the set of clock signals  $clk$ ,  $clk2$  and  $clk4$  that are used to

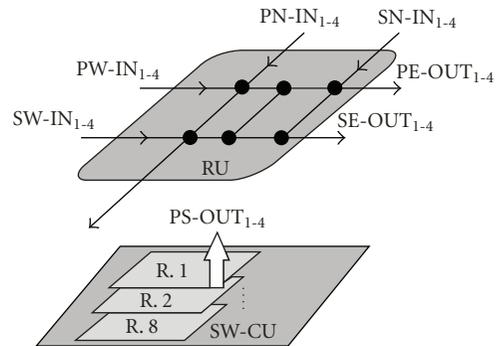


FIGURE 17: Dataflow between the SW-CU and RU building blocks inside the SW.

synchronize the routing of signals between the inputs and outputs of the SW for sequential and parallel modes of operation. The relationships between the signals  $clk$ ,  $clk2$ , and  $clk4$  have been defined in the Section 5.1.3. The SW-CU consists of a multiplexer and a RAM block. The RAM block contains eight sets of routing data, each of which represents a specific routing configuration. The multiplexer selects one of the eight routing modes based on a 3-bit control bus. The control mode is loaded via the configuration data bus in real time if necessary. Figures 17 and 18 show the data flow between the SW-CU and RU building blocks inside the SW and the block diagram of the SW-CU building block, respectively.

**5.3. The RSA Circuit for DFT Computations.** For parallel configuration, a 4-by-4 array that can implement a 16-point, as well as a 9-point DFT is depicted in Figure 19. Indeed, parallel DFT implementations for the case that  $N$  equals 10 and 12 can be realized on RSAs of 3-by-5 and 3-by-3 PE cells, respectively. Each of these arrays could also be configured to perform DFT computations for longer input sequences using serial modes of operation. In fact, the 4-by-4 array could also be configured to perform parallel input-serial output computation of a 14-point DFT for which the complete output sequence would be available every two clock cycles.

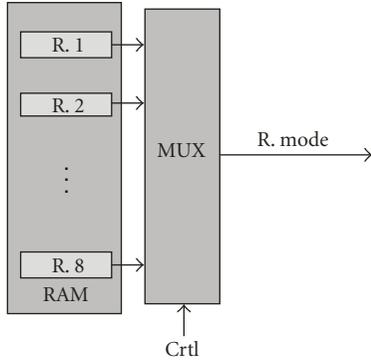


FIGURE 18: Block diagram of the SW-CU.

The 4-by-4 array could also be configured to implement a serial input-serial output realization of a 256-point DFT. The RAM block in each PE cell has to be expanded to support a set of 256 coefficients, and a complete output sequence would be available every 256 clock cycles.

*5.4. The RSA Circuit for Polyphase Filter Implementation.* An RSA circuit consisting of a 1-by-8 array of PE cells, as shown in Figure 20, can be configured to realize a 2-channel, 8-tap Polyphase filter.

The configured filter bank consists of four 8-tap FIR filters, each of which operates independently. Two complex input data sequences are simultaneously shifted into the array. Partial results generated by the multiplication of these inputs by the two coefficients in each PE cell are shifted to the next PE cell to be combined with partial products computed in that cell. The filter outputs are available at the output of the last PE cell.

*5.5. Circuit Implementations and Simulation Results.* In this section, circuit implementation and simulation results for the PE and the RSA structures for parallel computation of  $N$ -point DFTs, for  $N$  equal to 9, 10, 12, and 16 as well as a Polyphase filter bank consisting of four 8-tap FIR filters are presented. Additional results for RSA circuit implementations including a 32-point sequential DFT, 8-channel Polyphase filter, and 8-channel Polyphase-DFT Group Demultiplexer have been presented in [31].

For the implementations presented here, the PE complex data inputs are 24-bit wide and the intermediate result inputs to the PEs are 34-bit wide. Real coefficients are represented by 9-bit words while the complex coefficients in the DFT circuit consist of 18-bit words with 9-bit real and 9-bit quadrature. Two's complement number representation has been used throughout, and the arithmetic computations have been based on fixed-point representation. The RSA structures have been designed in VHDL, and MATLAB models of the configured functions have been used in circuit functional simulations. The VHDL designs based on a generic library which can be synthesized and mapped onto an FPGA or an ASIC. The circuit implementations presented have been synthesized and mapped onto Xilinx's xc5v1x330-2ff1760 FPGAs [32].

TABLE 3: PE performances and hardware requirements.

PE	Logic resources	Clock frequency (MHz)
Slice FF	413 (1%)	
4-input LUT	3431 (1.5%)	55

TABLE 4: Throughput and hardware requirements for representative DFT circuits.

$N$	Slice FF	4-Input LUT	Throughput(Gbps)
9	15706 (7.6%)	61821 (29.8%)	11.88
10	14717 (7.1%)	58031 (28%)	13.2
12	8612 (4.2%)	28383 (13.7%)	15.84
16	15706 (7.6%)	61821 (29.8%)	21.12

TABLE 5: Throughput and hardware requirement for the 2-channel, 8-tap Polyphase filter.

Polyphase filter	Logic resources	Throughput (Gbps)
Slice FF	7797 (3.8%)	
4-Input LUT	31194 (15.3%)	2.64

PE circuit simulations show that it can shift out four data samples per clock cycle at a clock frequency of 55 MHz. Thus the 4-by-4 RSA circuit computes a 16-point DFT in 18 ns and similarly the 4-by-4, 3-by-5 and 3-by-3 RSA circuits output 9, 10, and 12 point DFTs every 18 ns. For the Polyphase filter circuit, the 1-by-8 array offers a throughput of up to 55 MSPS per filter or 110 MSPS for the two filters. Simulation results for the PE cell are shown in Table 3 and for the various RSA-based circuit implementations in Tables 4 and 5.

## 6. Flexibility and Performance Comparison of Reconfigurable Architectures

The RSA was designed to incorporate reconfigurability of the array's PEs and SWs to enable the mapping and seamless interfacing of selected computationally intensive DSP functions such as those present in multicarrier baseband processors, that is, FIR filters and DFTs. The reconfigurability of the arithmetic processor functionality and signal routing within the array enables the RSA to be configured so that certain DSP functions can be computed in parallel or in sequential modes, thus providing a better match between the throughput and logic resource requirements. Thus a row of 8 PEs could, for example, be configured as four 8-tap FIR filters or one 32-tap FIR filter. Similarly an 8-by-8 array of PEs could be configured to implement a 32-point DFT or, alternately, at 1/8th the throughput, a single row of 8 PEs could be configured to operate in sequential mode with the same functionality. The latter configuration is possible as no data index mapping is required by the DFT-configured RSA circuit, hence it is feasible to time share the PEs.

In this section the configuration flexibility and performance of the RSA and representative configurable architectures reported in literature [27–29] will be considered. The features of interest are the circuit's architecture, ability to

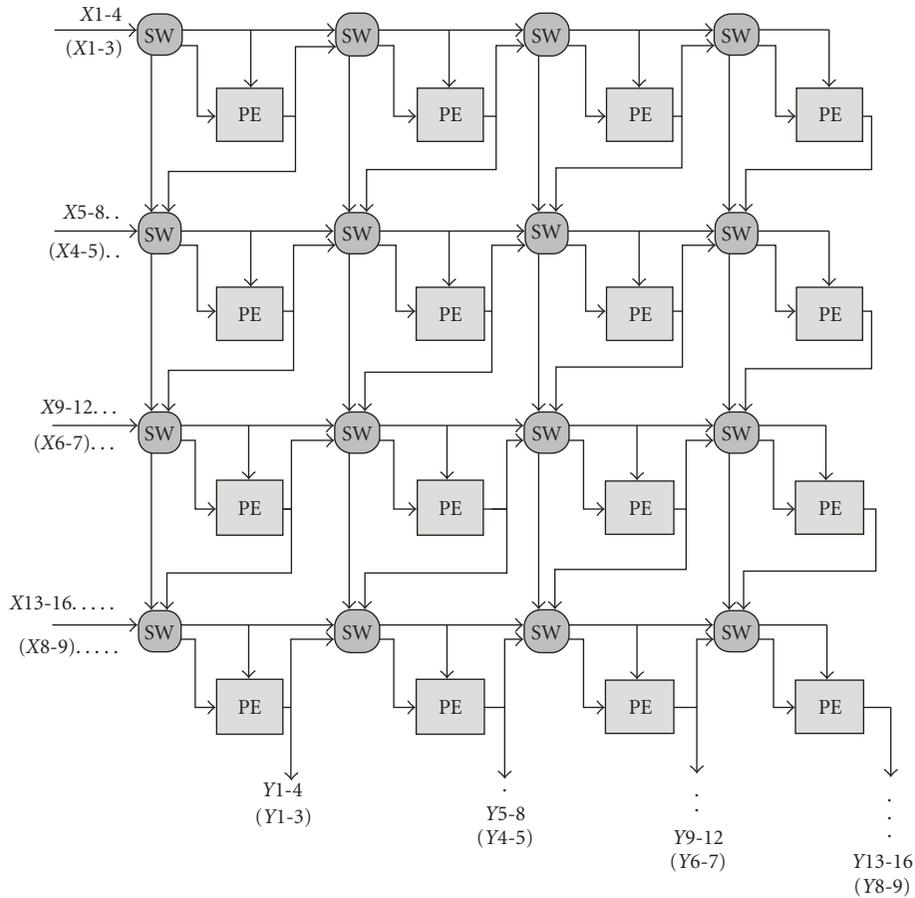


FIGURE 19: Circuit structure for parallel computation of 16- and 9-point DFTs.

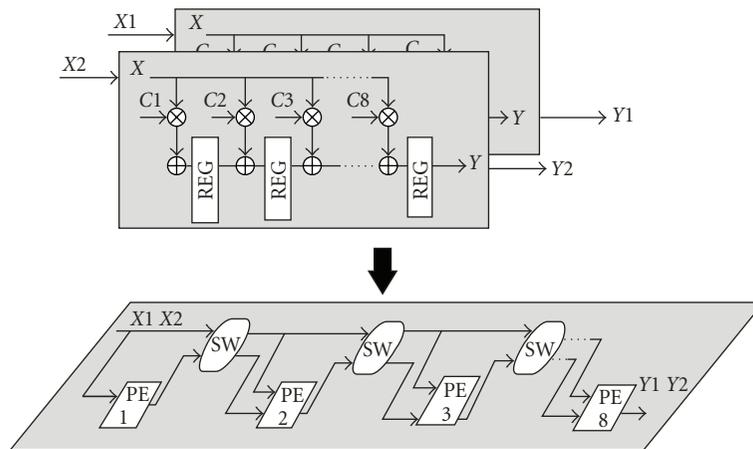


FIGURE 20: Configuration of 2-channel, 8-tap Polyphase filters on an RSA consisting of a 1-by-8 array of PE cells.

realize complex functions, and support of real time reconfigurability, scalability, and performance. Upon consideration of the different architectures, we note that all architectures support FIR filter implementations, however, not all support FFT or DFT circuits. Indeed the DRAW circuit does not support either DFTs or FFTs, and the CSoC supports only FFTs while the RADComm circuit supports DFT implementations based on the Goertzel algorithm. As previously indicated,

the FFT realization restricts unduly the dimensionality of the transform to powers of two. Furthermore, the CSoC circuit with 32 reconfigurable PE cells [29] requires a total of 100 clock cycles to compute a 64-point FFT. Thus, in terms of the ability to support both FIR filters and DFT, the RSA and RADComm circuits meet some of the previously identified requirements. However, the RADComm circuit was not designed for real time reconfiguration as its control

data and coefficients must be shifted in serially during start up and reconfiguration can only be initiated by circuit reset. To address the issue of reconfigurability in real time the RSA was designed to support the routing of configuration and coefficient data in real time.

## 7. Summary

A reconfigurable systolic array architecture that supports computations of DSP functions commonly used in multicarrier and multirate applications has been presented. The architecture consists of a systolic array of identical processing elements and switches that could be configured to perform arithmetic computations required by FIR filters, phase shifters and DFT/IDFT transforms of arbitrary dimensionality. The RSA circuit is reconfigurable in real-time and configuration data can be updated without interrupting the circuit operation. The FPGA implementation of the PE circuit shows that it can perform up to 220 million multiply and accumulate operations per second.

The RSA, a regular structure with PE cells and switches connected to their immediate neighbors, can be expanded and scaled in a seamless fashion. The RSA architecture's flexibility and configurability also makes it suitable for embedded or system on chip implementations.

## Appendices

### A. DFT Computation for $N$ Even

If  $N$  is a multiple of four (1) can be decomposed into a sum of products as follows:

$$\begin{aligned}
 Y(k) = & \left[ x(0) + x\left(\frac{N}{2}\right) e^{-j\pi k} \right. \\
 & \left. + \left( x\left(\frac{N}{4}\right) + x\left(\frac{3N}{4}\right) e^{-j\pi k} \right) e^{-j\pi k/2} \right] \\
 & + \sum_{n=1}^{N/4-1} \left( x(n) + x\left(\frac{N}{2} + n\right) e^{-j\pi k} \right) e^{-j2\pi nk/N} \\
 & + \sum_{n=1}^{N/4-1} \left( x(N-n) + x\left(\frac{N}{2} - n\right) e^{-j\pi k} \right) e^{j2\pi nk/N},
 \end{aligned} \tag{A.1}$$

where  $k = 0, 1, \dots, N-1$  and the coefficients in (A.1) can be defined as follows:

$$\begin{aligned}
 e^{-j2\pi nk/N} &= \left( \cos \frac{2\pi nk}{N} - j \sin \frac{2\pi nk}{N} \right), \\
 e^{-j\pi k} &= (-1)^k, \\
 e^{-j\pi k/2} &= (-j)^k.
 \end{aligned} \tag{A.2}$$

Substituting the terms shown in (A.2) into (A.1) and simplifying, we obtain

$$\begin{aligned}
 Y(k) = & S_1 + (-j)^k S_2 \\
 & + \left( \sum_{n=1}^{N/4-1} (S_3(n) + S_4(n)) * a(n) \right. \\
 & \left. + j \sum_{n=1}^{N/4-1} (S_3(n) - S_4(n)) * b(n) \right),
 \end{aligned} \tag{A.3}$$

where

$$\begin{aligned}
 S_1 &= x(0) + (-1)^k x\left(\frac{N}{2}\right), \\
 S_2 &= x\left(\frac{N}{4}\right) + (-1)^k x\left(\frac{3N}{4}\right), \\
 S_3(n) &= x(N-n) + (-1)^k x\left(\frac{N}{2} - n\right), \\
 S_4(n) &= x(n) + (-1)^k x\left(\frac{N}{2} + n\right), \\
 a(n) &= \cos \frac{2\pi nk}{N}, \\
 b(n) &= \sin \frac{2\pi nk}{N}.
 \end{aligned} \tag{A.4}$$

From (A.3), the output sequences  $Y(N-k)$ ,  $Y(N/2+k)$  and  $Y(N/2-k)$  can be inferred as follows:

$$\begin{aligned}
 Y(N-k) &= S_1 + (j)^k S_2 \\
 & + \left( \sum_{n=1}^{N/4-1} (S_3(n) + S_4(n)) * a(n) \right. \\
 & \left. - j \sum_{n=1}^{N/4-1} (S_3(n) - S_4(n)) * b(n) \right), \\
 Y\left(\frac{N}{2} + k\right) &= S_1 + (-j)^{k+N/2} S_2 \\
 & + \sum_{n=1}^{N/4-1} (-1)^n \\
 & \left( (S_3(n) + S_4(n)) * a(n) \right. \\
 & \left. + j(S_3(n) - S_4(n)) * b(n) \right),
 \end{aligned} \tag{A.5}$$

$$\begin{aligned}
 Y\left(\frac{N}{2} - k\right) &= S_1 + (j)^{k+N/2} S_2 \\
 & + \sum_{n=1}^{N/4-1} (-1)^n \\
 & \left( (S_3(n) + S_4(n)) * a(n) \right. \\
 & \left. - j(S_3(n) - S_4(n)) * b(n) \right),
 \end{aligned}$$

where  $k = 1, 2, \dots, N/4 - 1$ .

For  $k = 0, N/4, N/2, 3N/4$ , the four outputs  $Y(0)$ , and  $Y(N/4), Y(N/2), Y(3N/4)$  are expressed as follows:

$$\begin{aligned}
Y(0) &= \sum_{n=0}^{N-1} x(n), \\
Y\left(\frac{N}{4}\right) &= S_1 + (-j)^{N/4} S_2 \\
&\quad + \left( \sum_{n=1}^{N/4-1} (j)^n S_3(n) + (-1)^n S_4(n) \right), \\
Y\left(\frac{3N}{4}\right) &= S_1 + (-j)^{3N/4} S_2 \\
&\quad + \left( \sum_{n=1}^{N/4-1} (j)^n ((-1)^n S_3(n) + S_4(n)) \right), \\
Y\left(\frac{N}{2}\right) &= S_1 + (-j)^{N/2} S_2 \\
&\quad + \left( \sum_{n=1}^{N/4-1} (-1)^n (S_3(n) + S_4(n)) \right).
\end{aligned} \tag{A.6}$$

If  $N$  is even but not a multiple of four, the output sequence  $y(k)$  is derived from the decomposition of (1) using a similar approach as follows:

$$\begin{aligned}
Y(k) &= \left[ x(0) + x\left(\frac{N}{2}\right) e^{-j\pi k} \right] \\
&\quad + \sum_{n=1}^{N-2/4} \left( x(n) + x\left(\frac{N}{2} + n\right) e^{-j\pi k} \right) e^{-j2\pi nk/N} \\
&\quad + \sum_{n=1}^{N-2/4} \left( x(N-n) + x\left(\frac{N}{2} - n\right) e^{-j\pi k} \right) e^{j2\pi nk/N}.
\end{aligned} \tag{A.7}$$

Upon substituting in (A.6) with the terms defined in (A.2), the expression becomes

$$\begin{aligned}
Y(k) &= S_1 + \left( \sum_{n=1}^{N-2/4} ((S_3(n) + S_4(n)) * a(n)) \right. \\
&\quad \left. + j \sum_{n=1}^{N-2/4} ((S_3(n) - S_4(n)) * b(n)) \right).
\end{aligned} \tag{A.8}$$

The output sequence  $y(N-k)$  can be derived from (A.8) as follows:

$$\begin{aligned}
Y(N-k) &= S_1 + \left( \sum_{n=1}^{N-2/4} ((S_3(n) + S_4(n)) * a(n)) \right. \\
&\quad \left. - j \sum_{n=1}^{N-2/4} ((S_3(n) - S_4(n)) * b(n)) \right),
\end{aligned} \tag{A.9}$$

where  $k = 1, 2, \dots, N/2 - 1$ .

For  $k = 0, N/2$ , the two outputs  $Y(0), Y(N/2)$  can be obtained based on (A.6).

## B. DFT Computation for $N$ Odd

If  $N$  is odd, the output sequence  $Y(k)$  is derived from decomposition of (1) as follows:

$$\begin{aligned}
Y(k) &= x(0) + \sum_{n=1}^{N-1/2} x(n) e^{-j2\pi nk/N} + \sum_{n=1}^{N-1/2} x(N-n) e^{j2\pi nk/N}, \\
Y(k) &= x(0) + \sum_{n=1}^{N-1/2} \left( (x(N-n) + x(n)) \cos \frac{2\pi nk}{N} \right) \\
&\quad + j \sum_{n=1}^{N-1/2} \left( (x(N-n) - x(n)) \sin \frac{2\pi nk}{N} \right),
\end{aligned} \tag{B.1}$$

$$Y(k) = S_5(n) + \sum_{n=1}^{N-1/2} (S_6 * a(n)) + j \sum_{n=1}^{N-1/2} (S_7 * b(n)), \tag{B.2}$$

where

$$\begin{aligned}
S_5 &= x(0), \\
S_6(n) &= x(N-n) + x(n), \\
S_7(n) &= x(N-n) - x(n).
\end{aligned} \tag{B.3}$$

The output sequence  $Y(N-k)$  can be derived from (B.1) as follows:

$$Y(N-k) = S_5 + \sum_{n=1}^{N-1/2} (S_6 * a(n)) - j \sum_{n=1}^{N-1/2} (S_7 * b(n)), \tag{B.4}$$

where  $k = 1, 2, \dots, (N-1)/2$ .

For  $k = 0$ , the output  $Y(0)$  can be determined based on (A.6).

## References

- [1] ETSI, "Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television," EN 300 744 v1.5.1, 2004.
- [2] ETSI, "Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to Mobile, Portable and Fixed Receivers," EN 300 401 v1.4.1, 2006.
- [3] IEEE STD 802.11a, "High-Speed Physical Layer in 5 GHz Band," 1999, <http://ieee802.org>.
- [4] WPAN Working Group, <http://grouper.ieee.org/groups/802/15>.
- [5] ETSI, "Broadband Radio Access Networks (BRAN); HIPER-LAN Type 2 Physical (PHY) layer," TS 101 475 v1.1.1, April 2000.
- [6] Z. Yang, Y. Hu, C. Pan, and L. Yang, "Design of a 3780-point IFFT processor for TDS-OFDM," *IEEE Transactions on Broadcasting*, vol. 48, no. 1, pp. 57–61, 2002.
- [7] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: a tutorial," *Proceedings of the IEEE*, vol. 78, no. 1, pp. 56–93, 1990.
- [8] N. J. Fliege, *Multirate Digital Signal Processing*, John Wiley & Sons, New York, NY, USA, 1994.

- [9] M. Abo-Zahhad, "Current state and future directions of multirate filter banks and their applications," *Digital Signal Processing*, vol. 13, no. 3, pp. 495–518, 2003.
- [10] M. Re, A. Del Re, and G. Cardarilli, "Efficient implementation of a demultiplexer based on a multirate filter bank for the skyplex satellites DVB system," *Journal of VLSI Design*, vol. 15, no. 1, pp. 427–440, 2002.
- [11] O. Gerek and A. Cetin, "Adaptive polyphase subband decomposition structures for image compression," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1649–1660, 2000.
- [12] A. Mehrnia and B. Daneshrad, "A low-complexity multirate channel selector transmit filter bank with reconfigurable bandwidth," in *Proceedings of the IEEE Aerospace Conference*, pp. 1739–1749, March 2005.
- [13] S. Ramanathan, S. K. Nandy, and V. Visvanathan, "Reconfigurable filter coprocessor architecture for DSP applications," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 26, no. 3, pp. 333–359, November 2000.
- [14] Software Defined Radio Forum, <http://www.sdrforum.org>.
- [15] C.-H. Chang, C.-L. Chang, and Y.-L. Chang, "Efficient VLSI architectures for fast computation of the discrete Fourier transform and its inverse," *IEEE Transactions on Signal Processing*, vol. 48, no. 11, pp. 3206–3216, 2000.
- [16] W.-C. Yeh and C.-W. Jen, "High-speed and low-power split-radix FFT," *IEEE Transactions on Signal Processing*, vol. 51, no. 3, pp. 864–874, 2003.
- [17] K. Maharatna, E. Grass, and U. Jagdhold, "A 64-point Fourier transform chip for high-speed wireless LAN application using OFDM," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 3, pp. 484–493, 2004.
- [18] S. He and M. Torkelson, "Design and implementation of a 1024-point pipeline FFT processor," in *Proceedings of the Custom Integrated Circuits Conference*, pp. 131–134, 1998.
- [19] K. Sapiecha and R. Jarocki, "Modular architecture for high performance implementation of the FFT algorithm," *IEEE Transactions on Computers*, vol. 39, no. 12, pp. 1464–1468, December 1990.
- [20] G. Bongiovanni, "Two VLSI structures for the discrete Fourier transform," *IEEE Transactions on Computers*, vol. 32, no. 8, pp. 750–754, 1983.
- [21] J. G. Nash, "Computationally efficient systolic architecture for computing the discrete Fourier transform," *IEEE Transactions on Signal Processing*, vol. 53, no. 12, pp. 4640–4651, 2005.
- [22] S. Peng, I. Sedukhin, and S. Sedukhin, "Design of array processors for 2-d discrete Fourier transform," *IEICE Transactions on Information and Systems*, vol. E80-D, no. 4, pp. 455–464, 1997.
- [23] H. Lim and E. E. Swartzlander Jr., "Multidimensional systolic arrays for the implementation of discrete Fourier transforms," *IEEE Transactions on Signal Processing*, vol. 47, no. 5, pp. 1359–1370, 1999.
- [24] S. He and M. Torkelson, "A systolic array implementation of common factor algorithm to compute DFT," in *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks*, pp. 374–381, Tokyo, Japan, 1994.
- [25] K. J. Jones, "High-throughput, reduced hardware systolic solution to prime factor discrete fourier transform algorithm," *IEE Proceedings E*, vol. 137, no. 3, pp. 191–196, 1990.
- [26] J. A. Beraldin, T. Aboulnasr, and W. Steenaert, "Efficient one-dimensional systolic array realization of the discrete Fourier transform," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 1, pp. 95–100, 1989.
- [27] A. Alsolaim, *Dynamically reconfigurable architecture for third generation mobile systems*, Ph.D. thesis, Ohio University, 2002.
- [28] E. Grayver and B. Daneshrad, "A reconfigurable 8 GOP ASIC architecture for high-speed data communications," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 11, pp. 2161–2171, 2000.
- [29] S. Wallner, "A configurable system-on-chip architecture for embedded and real-time applications: concepts, design and realization," *Journal of Systems Architecture*, pp. 350–367, 2005.
- [30] H. Ho, V. Szwarc, and T. Kwasniewski, "Hardware optimization of a configurable polyphase-FFT design using common sub-expression elimination," in *Proceedings of the MWS-CAS/NEWCAS Conference*, pp. 5–8, August 2007.
- [31] H. Ho, V. Szwarc, and T. Kwasniewski, "A reconfigurable systolic array SoC design for multicarrier wireless applications," in *Proceedings of the Midwest Symposium on Circuits and Systems (MWSCAS '08)*, August 2008.
- [32] Xilinx Corporation, <http://www.xilinx.com>.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

