A low power radix-2 FFT accelerator for FPGA

Conference Paper · November 2015

DOI: 10.1109/ACSSC.2015.7421167

Soumak Mookherjee
Florida State University
7 PUBLICATIONS 38 CITATIONS
SEE PROFILE

Linda Sumners DeBrunner
Florida State University
103 PUBLICATIONS 567 CITATIONS
SEE PROFILE

Victor Debrunner
Florida State University
202 PUBLICATIONS 1,753 CITATIONS
SEE PROFILE

A Low Power Radix-2 FFT Accelerator for FPGA

Soumak Mookherjee, Linda DeBrunner, Victor DeBrunner

Electrical & Computer Engineering, Florida State University

***Abstract*—This paper presents a low power FFT accelerator using Radix-2 algorithm with 8-parallel multi-path delay commutator. Hardware accelerator can achieve better performance and throughput compared to software FFT routine. Thus, FFT accelerators are used in many DSP processor. In this paper a Radix-2 Multipath Delay Commutator (R2MDC) FFT accelerator is designed with 8-parallel processing of input samples. The hardware utilization of the architecture is 100% requiring only 4 parallel butterflies. It increases throughput to eight times of the traditional R2MDC. Thus, it can achieve similar throughput while running at one eighth of the clock frequency for a regular MDC accelerator while roughly increasing the gate capacitance by 4 times. Thus, it can operate at a lower power than the regular Radix-2 MDC accelerator. We implement our design on Xilinx Virtex FPGA and measure area, frequency, latency, throughput and power. We show that our design can operate at a similar rate while reducing the power by 25% on FPGA compared to R2MDC.**

***Index Terms*—FFT, DIF, DSP, R2MDC, 8-parallel, accelerator**

I. Introduction

The Fast Fourier Transform is a popular algorithm in the field of Digital Signal Processing. It has wide application in digital communication specially Orthogonal Frequency Division Multiplexing (OFDM) system, video broadcasting, speech and image processing. DSP processors are used in many of these application. Nowadays, there is a huge demand of mobile devices including smart phones, tablets and wearables. In these type of devices, software solution is not beneficial as the power consumption of software solution is considerably higher than the dedicated hardware and latency is high for software. Thus, we need specially designed hardware to address these issues. FFT accelerators are widely used in DSP processors for this purpose. Pipelined architecture is specially suitable for this purpose as they provide high throughput and low power as well as low latency.

There are two popular pipelined architecture for FFT: Single Delay Feedback (SDF) and Multi-path Delay Commutator (MDC). SDF architecture was presented in [1]. In SDF, each stage has a feedback loop where some of the outputs of butterfly are fed back to the memory of the same stage. The hardware utilization of SDF is 100%. On the other hand, in MDC which was described in [2], processed samples from one stage is always passed to the next stage. The hardware utilization of MDC is 50%. In some real-time applications such as OFDM or ultra-wide band (UWB) system, where high throughput is a requirement, it is important to be able to process the input samples in parallel. Also, it is a challenge to process several samples of input sequence in parallel when they are received in parallel. Thus, parallel pipelined FFT architectures have become popular in recent times. In this paper, we propose an 8-parallel Radix-2 MDC(R2MDC) architecture, which is more power efficient than the regular Radix-2 MDC and the 2-parallel Radix-2 proposed in [3] and suitable for using as an accelerator. Also, it has simpler butterfly structure and simpler control than the corresponding Radix-4 architecture.

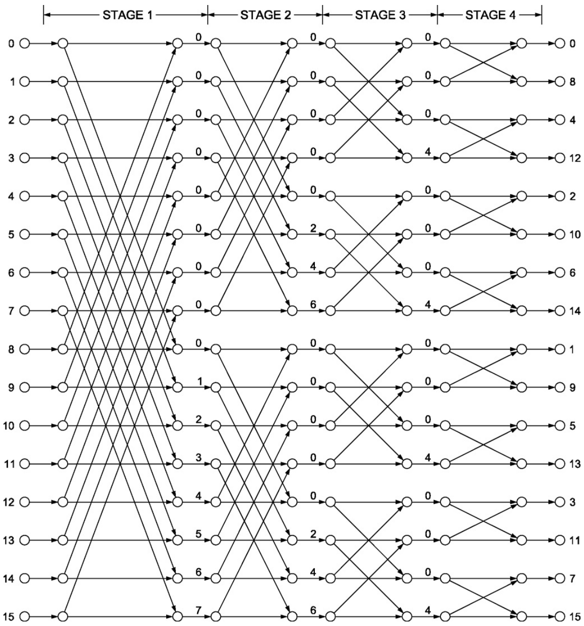

Fig. 1. Signal flow graph for 16-point Decimation in Frequency FFT algorithm

In the paper [4], a novel Radix-2^2 SDF structure was proposed which is still very popular for FFT accelerator. However, when we need high throughput, SDF is not a suitable choice since parallel architecture with SDF requires more area. In [5], a low power FFT architecture had been proposed for WLAN application, but it uses Radix-8 FFT structure. Thus, it is only suitable for input sequence lengths which are in power of eight. In the paper [6], a variable length low power FFT architecture was proposed using Radix-2/4/8 algorithm for OFDM system. It uses SDF architecture which is less efficient for parallel architecture. In the paper [7] 4-parallel Radix-2^4 FFT processor was presented for Ultra-Wide Band (UWB) applications. However, it used Mult-path Delay Feedback (MDF) which takes more hardware resources than a 4-parallel MDC architecture.

In recent time, several parallel pipelined FFT architectures were proposed in [3], but the 8-parallel architecture was not investigated. A power model was presented for the architectures, but no power analysis was done on real hardware. In [8], memory based FFT architecture was proposed using Radix-2^3 algorithm. But, memory based architecture is not useful for high throughput application. A Radix-4 MDC architecture was proposed with parallel datapath in [9], in which the input sequence length should be of power of four. In the paper [10], several parallel architectures were proposed for Radix-2^k. However, the designs do not have regular structure and also the paper does not present the power consumption on real hardware. In this paper, we have implemented our 8-parallel Radix-2 FFT accelerator on real hardware using Xilinx FPGA and measured area, throughput, latency and power consumption.

The dynamic power consumption of a CMOS circuit is given by [11],

$$P_{Dyn} = \alpha.C_L.V_{dd}^2.f_{clk} \qquad (1)$$

where α is the switching probability, C_L the sum of all capacitances that are being charged or discharged, V_{dd} is the supply voltage, and f_{clk} is the clock frequency of the circuit. The switching probability depends on the number of operations within a block including memory access. C_L depends on the hardware complexity of the circuit. It can seen from Eq(1) that if the clock frequency is reduced, the dynamic power consumption will also decrease. Thus, parallel pipeline architecture is an ideal candidate for power reduction.

In the next few sections, we first review the Radix-2 FFT algorithm, then we discuss the basic MDC architecture. In section IV , we present our proposed architecture for 8-parallel Radix-2 MDC architecture. We compare the area, latency, throughput and power of our architecture with those of the R2MDC and the 2-parallel R2MDC in section V. Finally, the contribution of this paper is summarized in section VI.

II. FFT Algorithm

The N-point Discrete Fourier Transform (DFT) of an input sequence *x*[n] is given below,

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk} \qquad (2)$$

where $W_N^{nk} = e^{-j\frac{2\pi}{N}nk}$ is called the twiddle factor. The most common algorithm for efficient computation of DFT when N is power of two is the Fast Fourier Transform (FFT) algorithm proposed by Cooley and Tuckey [12]. In this work, Radix-2 Decimation in Frequency (DIF) Cooley-Tuckey algorithm is chosen for the 64-point FFT computation. We choose Radix-2 as since it offers simple butterfly structure and simpler control and it can process any input sequence lengths which are in power of two. A Radix-4 would only process input sequences with lengths of power of four. In Radix-2 DIF FFT algorithm, N is divided by two in each stage, and butterfly-like operation is performed on two samples in each stage. The number of stages is given by, $v = log_2 N$. The computational complexity of DFT is reduced from $O(N^2)$ to $O(Nlog_2 N)$.

Fig. 1 shows the signal flow graph for radix-2 16-point FFT algorithm using decimation in frequency. At each stage, butterfly operations are performed between samples separated by $N/2^s$ sample points, where *s* is the stage number. The lower edges of the butterflies always have the transmittance -1. The bottom output of the butterfly is multiplied by the appropriate twiddle factor which depends upon the stage number and the sample points. The output of the butterflies are then fed to the next stage. At the final stage, FFT of the input sequence is obtained in bit-reversed order.

For the 64-point FFT, there are 32 twiddle factors. They are selected from the following sets for each stage.

stage1: $(W_0, W_1, W_2, W_3, W_4, ..., W_{31})$
stage2: $(W_0, W_2, W_4, W_6, ..., W_{30})$
stage3: $(W_0, W_4, W_8, ..., W_{28})$
stage4: $(W_0, W_8, W_{16}, W_{24})$
stage5: (W_0, W_{16})
stage6: (W_0)

It can be noticed that in each stage, subsequent twiddle factors differ by 2^{s-1} where *s* is the stage number.

III. Radix-2 MDC FFT Architecture

In this section, we review the Radix-2 MDC FFT architecture which was described in [2]. It is one of the most popular FFT architectures. Fig. 2 shows a 64-point Radix-2 MDC FFT architecture. In this architecture, input samples separated by *N/2* samples are fed to the BFI. This can be achieved by placing *N/2* delay elements before the second input. In the first *N/2* (32 in this case) clock cycle, BFI generates the outputs for the first stage.

Fig. 3 show the internal structure of the butterflies. It is simply made of an adder and a subtractor. It accepts two input signal and generates the sum and the difference of the two inputs. The lower outputs of the butterfly are fed through the complex multiplier and is stored in the delay elements of the next stage. In the complex multiplier, the samples are multiplied by the appropriate twiddle factors stored in the ROM.

The switch acts as a dual port multiplexer. Fig. 4 shows the internal structure of the switch consisting of two multiplexers and a common control signal. When, the control signal *sel* is 0, the switch simply passes the inputs, but when *sel* is 1 the switch swaps the inputs. For the first 32 clock cycle, *sel1* is set to 0. For the next 16 cycles, the *sel1* is also kept to 0, and for the next 16 cycles, the *sel1* is set to 1 so that BFII starts processing the samples which are now 16 cycles apart. The samples are processed through the subsequent stages in the similar manner.

Overall latency for this architecture is *N* clock cycles, i.e. 64 cycles in our case. The hardware utilization is 50%.

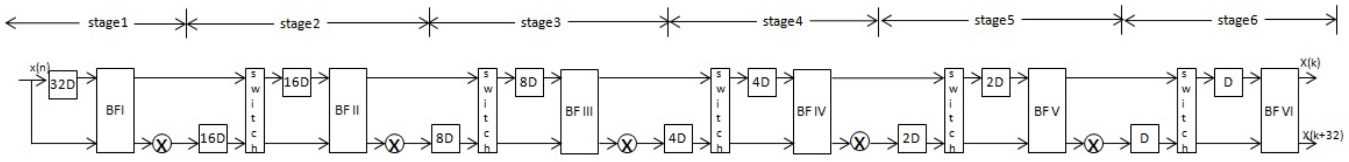

Fig. 2. 64-point R2-MDC FFT architecture

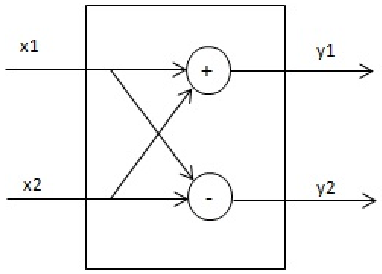

Fig. 3. Butterfly structure of Radix-2 MDC FFT architecture

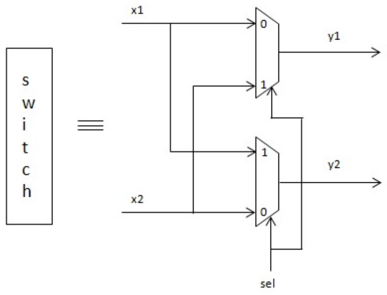

Fig. 4. The internal structure of the Switch

IV. Proposed 8-parallel Radix-2 MDC FFT Accelerator

In this section, we present our proposed architecture which is capable of 8-parallel data processing. Fig. 5 show the block diagram of the proposed architecture. In this design, eight input samples are applied at the first stage in parallel. However, only four parallel datapaths are used. By efficient usage of the delay elements and data scheduling, we achieve 8-parallel processing of the samples. The hardware utilization of the design is 100% and the latency is N/8 clock cycles.

A. Operation

In the first stage, samples x[8n] and x[8n+4] are processed in datapath I, samples x[8n+1] and x[8n+5] are processed in datapath II and so on. The delay elements are placed in a way so that inputs to the first butterflies are separated by 32 samples. After first stage, outputs from the upper port of the butterfly from datapath III and IV are moved to the datapath I and II respectively. Similarly, outputs from the multiplier from datapath III and IV are moved to the datapath I and II respectively. Thus, upper two datapaths now contain the first 32 samples and lower two datapaths contain the last 32 samples.

In the second stage, the samples that are processed through the butterflies are separated by 16 samples. The outputs from lower ports of the butterflies of datapaths I and III are fed to the datapath II and IV respectively. Similarly, the outputs from upper ports of the butterflies of datapaths II and IV are fed to the datapath I and III respectively. Thus, after this stage the top datapath would have the first 16 samples, second datapath would have the second set of 16 samples and so on. The remaining stages for each datapath are exactly same for regular Radix-2 MDC datapath.

B. Control signals

Generating the control signals is simple. It can be done by a simple five bit counter ($b_4b_3b_2b_1b_0$). The counter is incremented by 1 in each clock cycle. The multiplexer control signals are the same across the four datapaths. They are generated as follows,

sel1= b_2; *sel2*= b_1;
sel3= b_0; *sel4*= $b_2\bar{b}_1 + b_2\bar{b}_0 + \bar{b}_2b_1b_0$;
sel5= $b_1 \oplus b_0$; *sel6*= $\bar{b}_0$

C. ROM Address Generator

The twiddle factors are stored in the ROMs. The address generation for the ROMs are simple. We use the same 5-bit counter. However, in stages *1* and *2* different twiddle factors are selected for different datapaths. These can be easily generated too. We present below how different addresses are related. We use notation $addr_{LS}$ for denoting the address for datapath *L* and stage *S*. These can be easily verified from twiddle factor sets presented in section 2.

$addr_{21} = addr_{11} + 1; addr_{31} = addr_{11} + 2;$
$addr_{41} = addr_{11} + 3; addr_{22} = addr_{12} + 2;$
$addr_{32} = addr_{12}; addr_{42} = addr_{22};$

V. Results & Discussions

The proposed FFT accelerator has ben designed in VHDL. For comparison, we have also implemented the R2MDC and the 2-parallel R2MDC as described in [3] using VHDL. The wordlength is chosen as 16. The designs are synthesized for Xilinx FPGA using ISE 14.4 version. The target device is chosen to be Virtex-5 xc5vlx50-3ff1153. Although the generated outputs are in bit-reversed order, bit reverse circuit is not considered for the discussion presented here.

In the Table I, we compare the complexity and performance of the proposed 8-parallel architecture with that of R2MDC and 2-parallel R2MDC described in [3]. It can be seen that

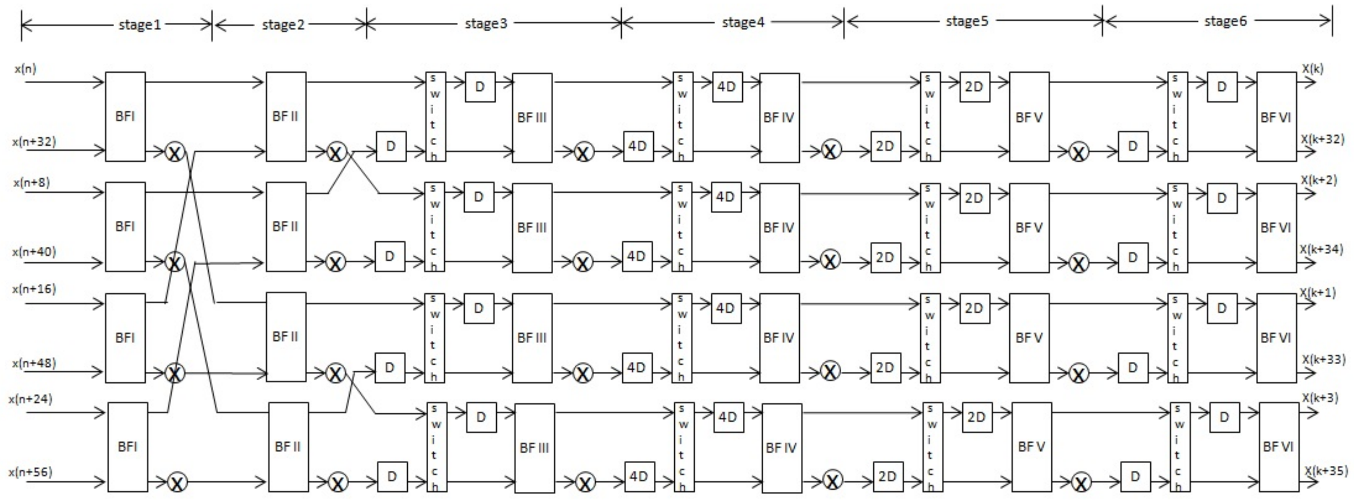

Fig. 5. Proposed 64-point 8-parallel R2MDC FFT architecture

the number of complex adder and complex multiplier of the proposed design is four times the other two designs. However, the number of delay elements is almost similar. The latency of the proposed architecture is one eighth of the latency of the R2MDC while it is one fourth of the latency of the R2MDC 2-parallel. Thus, the proposed FFT architecture greatly reduced the latency of the system. The throughput of the proposed system is increased by eight times than that of the R2MDC, while compared to 2-parallel R2MDC it is increased by a factor of four.

For comparison on power consumption, we fix the throughput of all three system at the same rate and estimate the power by Xilinx Power Analyzer tool. The synthesis results along with power estimation are presented in table II. It can be seen that proposed design consumes less power than the other two designs. As stated before, the number of adders and multiplier of the proposed design are four times more than that of the R2MDC and 2-parallel the R2MDC. Thus, the capacitance of the circuit is roughly increased the 4 times, but since the throughput is also increased by 8, the power consumption should decrease by 50% compared to the R2MDC according to Eq(1). However, it can be seen from the table that power reduction in FPGA is not 50%, that is because power of FPGA depends on various parameters, for example, the number of slices used, number of IO banks occupied etc.

The synthesis result is consistent with our complexity analysis. The reduction in frequency is attributed to complex interconnection in FPGA and possible overhead in ROM access for four datapaths in our proposed architecture.

TABLE I
COMPLEXITY AND PERFORMANCE COMPARISON OF PROPOSED 8-PARALLEL R2MDC WITH R2MDC AND 2-PARALLEL R2MDC

	R2MDC	2-P R2MDC	8-P R2MDC (Proposed)
Complex Adder	$4log_4N$	$4log_4N$	$16log_4N$
Complex Mult	$2(log_4N-1)$	$2(log_4N-1)$	$8(log_4N-1)$
Complex Memory	$3N/2-2$	$3N/2-2$	$7N/4$
Latency (cycles)	N	N/2	N/8
Throughput (samples/cycle)	1	2	8
Control	simple	simple	simple

TABLE II
AREA, LATENCY, THROUGHPUT AND POWER COMPARISON OF PROPOSED R2MDC WITH R2MDC AND 2-PARALLEL R2MDC

	R2MDC	2-P R2MDC	8-P R2MDC(Proposed)
Slice Registers	804	944	3082
Slice LUT	2061	2578	12645
Total Slice Used	705	800	4004
DSP Blocks	20	20	48
Frequency(Mhz)	46.65	41.37	30
Power(mW)	168	141	125

VI. CONCLUSION

A low power 8-parallel Radix-2 MDC FFT accelerator is designed in this paper. The design was implemented in FPGA and compared with previously presented popular FFT accelerators Radix-2 MDC and 2-parallel Radix-2 MDC architecture. Complexity analysis and performance analysis are performed for all three accelerators. The proposed design is found to occupy four times more hardware resources, but at the same time it reduces the latency and increases the throughput of the system by a factor of eight.

Power analysis is performed and the proposed design is found to consume 25% less power than the traditional R2MDC and 10% less power than the 2-parallel R2MDC. This is achieved by processing 8 samples in parallel although it uses only 4-parallel butterfly structures. Thus, this design is particularly useful for low power applications such as WLAN etc. For future work, we plan to optimize the complex multipliers for more power reduction and optimize the ROM storage as well.

References

[1] E. Wold and A. Despain, “Pipeline and parallel-pipeline FFT processors for VLSI implementations,” *IEEE Transactions on Computers*, vol. C-33, no. 5, pp. 414–426, 1984.

[2] L. R. Rabinar and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[3] M. Ayinala, M. Brown, and K. Parhi, “Pipelined parallel FFT architectures via folding transformation,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 6, pp. 1068–1081, 2012.

[4] S. He and M. Torkelson, “Design and implementation of a 1024-point pipeline FFT processor,” in *Custom Integrated Circuits Conference, 1998. Proceedings of the IEEE 1998*, 1998, pp. 131–134.

[5] C.-T. Lin, Y.-C. Yu, and L.-D. Van, “A low-power 64-point FFT/IFFT design for IEEE 802.11a WLAN application,” in *2006 IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006. Proceedings*, 2006, pp. 4 pp.–4526.

[6] Y.-T. Lin, P.-Y. Tsai, and T. D. Chiueh, “Low-power variable-length fast fourier transform processor,” *Computers and Digital Techniques, IEE Proceedings -*, vol. 152, no. 4, pp. 499–506, 2005.

[7] M. Shin and H. Lee, “A high-speed four-parallel radix-24 FFT/IFFT processor for UWB applications,” in *IEEE International Symposium on Circuits and Systems, 2008. ISCAS 2008*, 2008, pp. 960–963.

[8] S. Langemeyer, P. Pirsch, and H. Blume, “A FPGA architecture for real-time processing of variable-length FFTS,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 1705–1708.

[9] D. Jeon, M. Seok, C. Chakrabarti, D. Blaauw, and D. Sylvester, “Energy-optimized high performance FFT processor,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 1701–1704.

[10] M. Garrido, J. Grajal, M. Sanchez, and O. Gustafsson, “Pipelined radix-feedforward FFT architectures,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 1, pp. 23–32, 2013.

[11] 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.

[12] J. W. Cooley and J. W. Tukey, “An algorithm for the machine calculation of complex fourier series,” *Mathematics of computation*, vol. 19, no. 90, p. 297–301, 1965.