JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

IMPLEMENTATION OF DATA COMPRESSION SCHEMES USING FFT PROCESSORS FOR HIGH SPEED COMMUNICATION SYSTEMS

[1]Sk. Ayesha ,[2] Dr. Mohammad Hayath Rajvee, [3]Jabeena Shaik
[1]PG Scholar,[2]Professor,[3]Associate Professor & HOD, Dept. of E.C.E
[1,2,3]Quba College of Engineering and Technology, Nellore, SPSR Nellore Dist., A.P

ABSTRACT

A high-throughput programmable fast Fourier transform (FFT) processor is designed supporting 16- to 4096-point FFTs and 12- to 2400-point discrete Fourier transforms (DFTs) for 4G, wireless local area network, and future 5G. A 16-path data parallel memory-based architecture is selected as a tradeoff between throughput and cost. To implement a hardware-efficient high-speed processor, several improvements are provided. To maximally reuse the hardware resource, a reconfigurable butterfly unit is proposed to support computing including eight radix-2 in parallel, four radix-3/4 in parallel, two radix-5/8 in parallel, and a radix-16 in one clock cycle. Twiddle factor multipliers using different schemes are optimized and compared, wherein modified coordinate rotation digital computer scheme is finally implemented to minimize the hardware cost while supporting both FFTs and DFTs.

I. INTRODUCTION

The orthogonal frequency-division multiplex (OFDM) modulation technique has been widely exploited in communication systems, such as 802.11, terrestrial digital video broadcasting (DVB-T), and terrestrial digital multimedia broadcasting (DMB-T). However, OFDM modulation involves the discrete Fourier transform (DFT) that needs substantial computation. Today, various FFT processors, such as pipelined or memory-based architectures, have been proposed for different applications. However, for long-size FFT processors, such as the 2048-point FFT, the pipelined architecture would cost more area and power than the memory-based design. Hence, memorybased approach has gained more and more attention recently in FFT processor designs for long-size DFT applications. For the memory-based processor design, minimizing the necessary memory size is effective for area reduction since the memory costs a significant part of the processor. On the other hand, the FFT processor usually adopts on-chip static random access memory (SRAM) instead of external memory. The reason is the high-voltage I/O and the large capacitance in the printer-circuit-board (PCB) trace would increase power consumption for external memory. Besides the power issue, using external memory also increases the PCB-level verification cost for end-product manufacturers. Therefore, it is a trend to use the on-chip SRAM for FFT processors and to conduct FFT optimization for better system-level integration.

II. Literature Survey

To minimize the necessary memory size, an in-place approach [1] is taken for both butterflies output and I/O data. That is, the output data of butterflies are written back to their original location during the computation time. Moreover, for the I/O data, the new input data x[n] would be put in the location of the output data X[n] of the previous FFT symbol. On the other hand, for the memory-based processor, the high-radix structure would be taken to increase the throughput to meet real-time requirements. However, when both in-place policy and highradix structure are adopted simultaneously, it would result in memory conflict problem if the data have not been addressed properly. Until now, there have been several publications on this issue, the in-place approach for multibank memory structure, to optimize the processor design [1]–[3]. However, the approaches mentioned in [1] and [2] only work for fixed radix-r. The approach in [3] only works for radix-2/4. Hence, all of them could not be used for prime-number memory-based DFT processor design, and this still remains a challenge today. Note that a prime-number FFT has been proposed in recent communication standards, e.g., the 3780-point DFT in Chinese direct TV (DMB-T) [4] and the 1536-point DFT in Third Generation Partnership Project Long-Term Evolution [5]. Although, for the prime-number FFT computation, the zero-padding method can be taken to approximate it by computing some 2n-point FFT, it has poor signal-to-quantization-noise-ratio performance than a prime-radix FFT. Therefore, to develop an approach for a multibank in-place addressing algorithm for general size, an FFT is necessary. In this brief, a generalized mixed-radix (GMR) algorithm is proposed to optimize the memory-based FFT processor design. It supports not only in-place policy to minimize the necessary memory size for both butterflies output and I/O data but also multibank memory structure to increase its maximum throughput to satisfy more system applications without memory conflict. After the algorithm is introduced, we take the 3780- point FFT as an illustrative example. Finally, a low-complexity hardware implementation of an index vector generator is also proposed for our algorithm.

III. PROPOSED METHOD

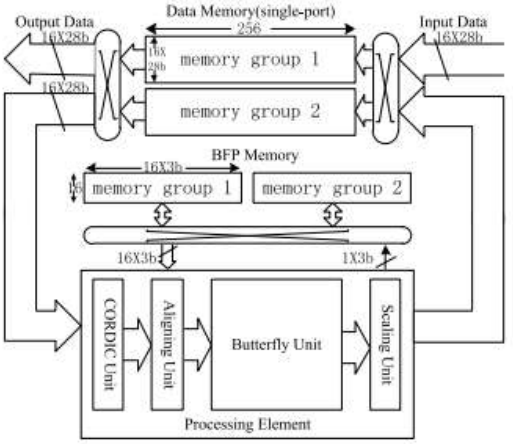


Fig. 1. Structure of the FFT processor

As shown in Fig. 1, the proposed FFT processor consists of: 1) a ping-pong data memory including two 16-bank 28-bit single-port memory groups (28-bit data width including 14-bit real part and 14-bit image part); 2) a block floating point (BFP) [5] memory that stores the exponents of the data in BFP format; and 3) a processing element including a CORDIC unit, an aligning unit, a butterfly unit, and a scaling unit. The CORDIC unit conducts TF multiplications. The aligning unit and the scaling unit conduct aligning operations and scaling operations, respectively, for BFP operations. To support different radices, multiple butterfly units have been proposed. A unified butterfly unit in [21] supports radix-2, -3, -4, -5, and -7 butterfly operations by reusing the hardware adders and multipliers. The enhanced delay element matrix unit in [17] supports radix-2, -3, -4, -5, -8, -9, -16, and -25 butterfly operations by using the 2-D DFT factorization method. The high radix small butterfly (HRSB) unit in [18] supports radix-2, -3, -4, -5, -8, -9, -12, -15, -16, and -25 butterfly operations by using a two-stage multipath delay commutator unit. According to the design space exploration in Section II, the butterfly unit we proposed is based on the radix-16 butterfly unit and is reformulated to support 2-, 3-, 4-, 5-, 8-, and 16-point radices. As shown in Fig. 3, the butterfly unit contains two processing element (PE)-A units, a PE-B unit, two PE-C units, and several switch networks. The butterfly unit supports one radix-16, two radix-5/8 in parallel, four radix3/4 in parallel, or eight radix-2 operations in parallel.

Fast Fourier Transform and Inverse Fast Fourier Transform are the most efficient and fast algorithms to calculate the Discrete Fourier Transform and Inverse Discrete Fourier Transform respectively. Fast Fourier Transform/Inverse Fast Fourier Transform is mostly used in many communication applications like Digital Signal Processing and the implementation of this is a growing research. From the last years, OFDM became an important one in FFT algorithms and is going to be implemented. The efficient multiple access method for Bandwidth in digital communications is OFDM (Engels, 2002; Nee & Prasad, 2000). Many of nowadays OFDM technique can be used in most important wireless communication systems: Digital Audio Broadcasting (DAB) (World DAB Forum, n.d.), Digital Video Broadcasting (DVB), Wireless Local Area Network (WLAN), Wireless Metropolitan Area Network (WMAN) and Multi Band – OFDM Ultra Wide Band (MB–OFDM UWB). Moreover, this method is also utilized in important wired applications like Asymmetric Digital Subscriber Line (ADSL) or Power Line Communication (PLC). Every communication system must have both Transmitter and Receiver. At the Transmitter side, IFFT is used for modulating signal, which depends on the OFDM system and at the Receiver side, FFT is used for demodulating signal. The FFT/IFFT are the important modules in OFDM transceivers. From this we can say that, the most parts of OFDM systems are, IFFT can be used at the transmitter side where as viterbi decoder can be used at the receiver side (Maharatna et al., 2004). The FFT is the second calculative huge part in the receiver. Therefore, the implementation of the FFT and IFFT must be designed to achieve the required throughput with the reduced area and delay.The demanding requirements of modern OFDM transceivers lead, in many cases, to the implementation of special–purpose hardware for the most critical parts of the transceiver. Thus, it is common to find the FFT/IFFT implemented as a Very Large Scale Integrated (VLSI) circuit. The techniques applied to the FFT can be applied to the IFFT as well. Moreover, the IFFT can be easily obtained by manipulating the output of a FFT processor. Therefore, the discussion in this chapter concentrates on the FFT without loss of generality. The inverse discrete Fourier transform can be found using Which can be expressed as Where is called the twiddled factor We can see that the difference between the inverse discrete Fourier and forward Fourier transform is the twiddled factor and the division by 1/N is called the twiddled factor.

IV.EXTENSION :

MIMO-OFDM defines multi input and multi output orthogonal frequency division multiplexing which is dominated over 4G AND 5G wireless communications. The word multi input and multi output defines which is capable of sending multiple signals over multi antennas and orthogonal frequency division multiple multiplexing divides the radio channels into largely spaced sub channels so that the data can be used in communication system without loss in reliability. In earlier MIMO is used with a combination of time division multiple access, code division multiple access, but MIMO with OFDM is much famous for its high data rate, high message deliver capacity, high throughput. For these reasons only it is familiar at wires LAN and some standard networks at mobile communications. We investigate techniques to implement the direct recoding of the sum of two numbers in its Modified Booth (MB) and Wallace form.

Application

Communication system

Dsp processors

Advantage

Low area

Low usage of memory

Low power

RESULTS

RTL SCHEMATIC:

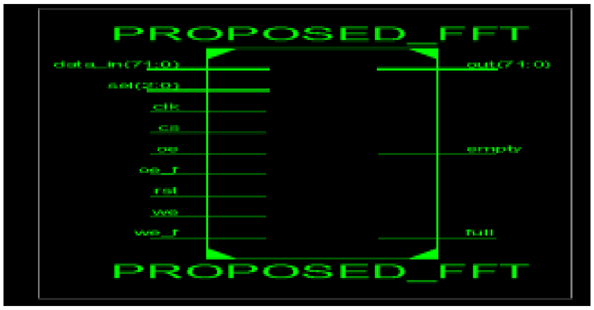
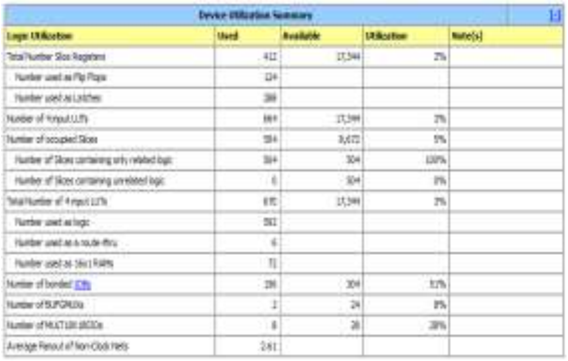
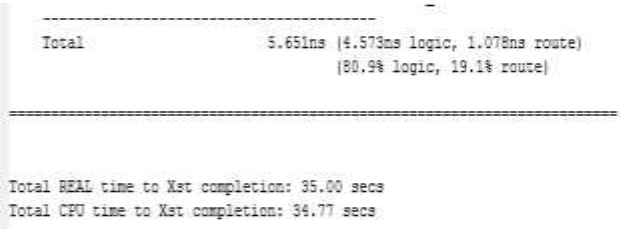


Fig 2: RTL Schematic of Final Output



Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Total Number Slice Registers	412	17,344	2%	
Number used as Flip Flops	124			
Number used as Latches	288			
Number of 4 input LUTs	864	17,344	2%	
Number of occupied Slices	504	8,672	5%	
Number of Slices containing only related logic	304	304	100%	
Number of Slices containing unrelated logic	0	304	0%	
Total Number of 4 input LUTs	670	17,344	3%	
Number used as logic	592			
Number used as a route-thru	6			
Number used as 16x1 RAMs	72			
Number of bonded IOBs	156	304	51%	
Number of BUFGMUXs	1	24	4%	
Number of MULT18X18SIOs	8	28	28%	
Average Fanout of Non-Clock Nets	2.61			

Fig 3: Device Utilization Summary

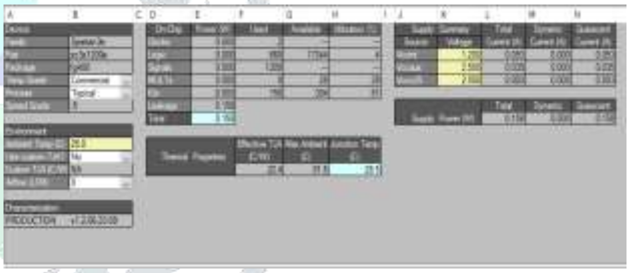


```
   -----------------------------------------
   Total                      5.651ns (4.573ns logic, 1.078ns route)
                                      (80.9% logic, 19.1% route)

=========================================================================


Total REAL time to Xst completion: 35.00 secs
Total CPU time to Xst completion: 34.77 secs
```

Fig 4: Delay in Time



Device	
Family	[illegible]
Part	xc3s1200e
Package	fg400
Temp Grade	Commercial
Process	Typical
Speed Grade	[illegible]
Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	0
Characterization	
PRODUCTION	[illegible]

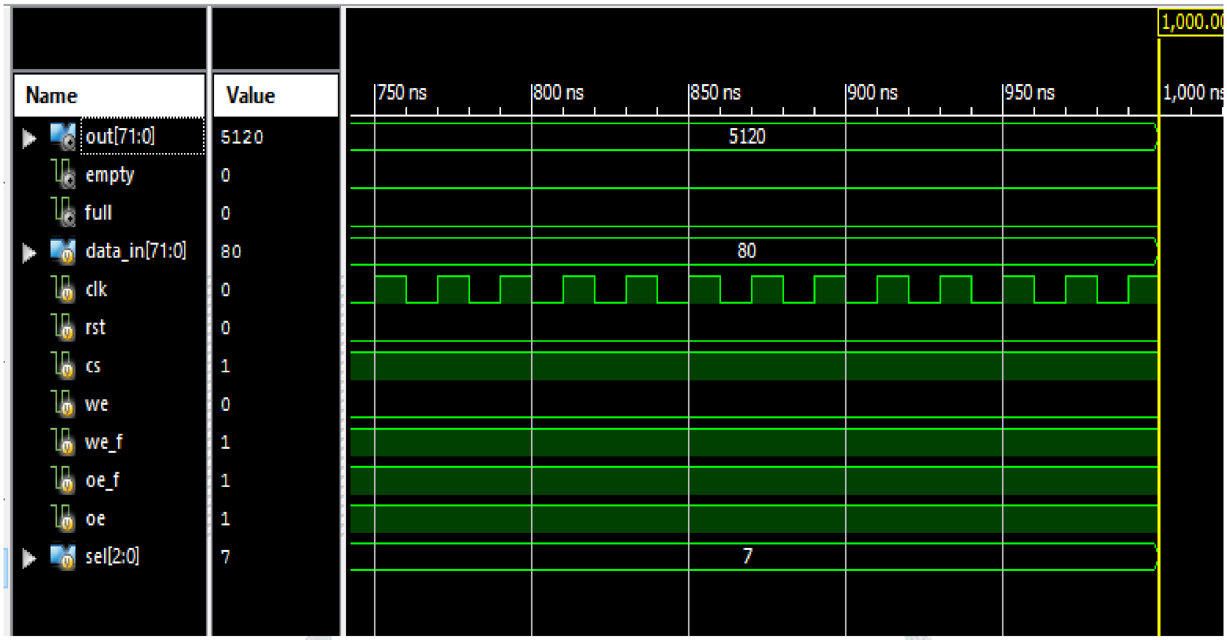
On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	[illegible]	[illegible]	---	---
Logic	[illegible]	[illegible]	17344	[illegible]
Signals	[illegible]	[illegible]	---	---
MULTs	[illegible]	8	28	[illegible]
IOs	[illegible]	[illegible]	304	[illegible]
Leakage	[illegible]			
Total	[illegible]			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	[illegible]	[illegible]	[illegible]

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	[illegible]	[illegible]	[illegible]
Vccaux	2.500	[illegible]	[illegible]	[illegible]
Vcco25	2.500	[illegible]	[illegible]	[illegible]

	Total	Dynamic	Quiescent
Supply Power (W)	[illegible]	[illegible]	[illegible]

Fig 5: Power Supplied



Name	Value
out[71:0]	5120
empty	0
full	0
data_in[71:0]	80
clk	0
rst	0
cs	1
we	0
we_f	1
oe_f	1
oe	1
sel[2:0]	7

Fig 6: Final Simulation Result

V.CONCLUSION

In this work, a high-speed programmable memory-based In this paper, a high-speed programmable memory-based and 16- to 4096-point FFTs for 4G, WLAN, and future 5G. By reusing the hardware resources, a 16-path data parallel butterfly unit is proposed, supporting eight radix-2 in parallel, four radix-3/4 in parallel, two radix-5/8 in parallel, and a radix-16 in one clock cycle. A CORDIC scheme is proposed to support both FFTs and DFTs with high hardware efficiency. An optimized add-based data access scheme is also proposed to support multiple butterfly units at any radix. ASIP designer is used to synthesize this design from high level.

REFERENCES

[1] L. G. Johnson, “Conflict free memory addressing for dedicated FFT hardware,” IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 39, no. 5, pp. 312–316, May 1992.

[2] J. A. Hidalgo, J. Lopez, F. Arguello, and E. L. Zapata, “Area-efficient architecture for fast Fourier transform,” IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 46, no. 2, pp. 187–193, Feb. 1999.

[3] B. G. Jo and M. H. Sunwoo, “New continuous-flow mixed-radix (CFMR) FFT processor using novel in-place strategy,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 5, pp. 911–919, May 2005.

[4] Z.-X. Yang, Y.-P. Hu, C.-Y. Pan, and L. Yang, “Design of a 3780-point IFFT processor for TDS-OFDM,” IEEE Trans. Broadcast., vol. 48, no. 1, pp. 57–61, Mar. 2002.

[5] 3GPP TS 36.201 V8.3.0 LTE Physical Layer—General Description, E-UTRA, Mar. 2009.

[6] C. Burrus, "Index mappings for multidimensional formulation of the DFT and convolution," IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-25, no. 3, pp. 239–242, Jun. 1977.

[7] D. P. Kolba and T. W. Parks, "A prime factor FFT algorithm using high-speed convolution," IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-25, no. 4, pp. 281–294, Aug. 1977.

[8] A. M. Despain, "Very fast Fourier transform algorithms hardware for implementation," IEEE Trans. Comput., vol. C-28, no. 5, pp. 333–341, May 1979.