EFFICIENT ARCHITECTURE FOR PROCESSING OF TWO INDEPENDENT DATA STREAMS USING RADIX-2 FFT

Artham Lasya[1] Dr.Shailendra Mishra

lasya116@gmail.com[1] mishra22feb@rediffmail.com[2]

[1]PG Scholar, VLSI,Bharath Institute ofEngineering and Technology, Mangalpally, Ibrahimpatnam, R.R District, India.
[2] Professor,Dept of ECE,Bharath Institute of Engineering and Technology, Mangalpally, IbrahimpatnamR.R.District, India.

Abstract:

Most of the applications require simultaneous computation of multiple independent fast Fourier transform (FFT) operations with their outputs in natural order. Therefore, this brief presents a novel pipelined FFT processor for the FFT computation of two independent data streams. The proposed architecture is based on the multipath delay commutator FFT architecture. It has N/2-point decimation in time FFT and N/2-point decimation in frequency FFT to process the odd and even samples of two data streams separately. The main feature of the architecture is that the bit reversal operation is performed by the architecture itself, so the outputs are generated in normal order without any dedicated bit reversal circuit. The bit reversal operation is performed by the shift registers in the FFT architecture by interleaving the data. Therefore, the proposed architecture requires a lower number of registers and has high throughput.

Key words*:Bit reversal, bit reversed order, fast Fouriertransforms (FFT), multipath delay commutator (MDC) FFT,and normal order.*

I. INTRODUCTION:

Radix-2 FFT Algorithm:

Let us consider the computation of the $N = 2^v$ point DFT by the divide-and conquer approach. We split the N-point data sequence into two $N/2$-point data sequences $f_1(n)$ and $f_2(n)$, corresponding to the even-numbered and odd-numbered samples of $x(n)$, respectively, that is,

$$f_1(n) = x(2n)$$
$$f_2(n) = x(2n+1), \qquad n = 0,1,\ldots,\frac{N}{2}-1$$

This efficiency of the FFT is at a maximum when the length of the sequence is a power of 2, i.e., N=2pN=2p, with pp a positive integer. The complexity of FFT algorithms is O(Nlog2 N), while calculating the DFT by the canonical expression would cost O(N2)operations. It happens that FFTs can be performed using DFTs of even and odd points, which is called a “Decimation-In-Time” (DIT) FFT, or they can be decomposed using a first-half/second-half approach, which is called a "Decimation-In-Frequency" (DIF) FFT.

Decimation-In-Time (DIT) FFT:

Below is the schema of calculations for a 8-point FFT using DIT. Notice that

the X[k] are output in correct order, but the x[n]are pre-ordered:

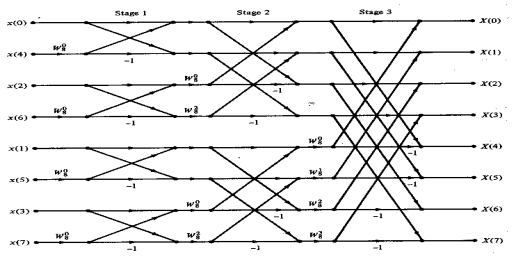

Fig1. Eight-point decimation-in-time FFT algorithm.

"Decimation-In-Frequency" (DIF) FFT:

The schema of calculations for a 8-point FFT using DIF. The x[n] values are input in order, but the FFT X[k]is output in an incorrect order. They have to be re-ordered.

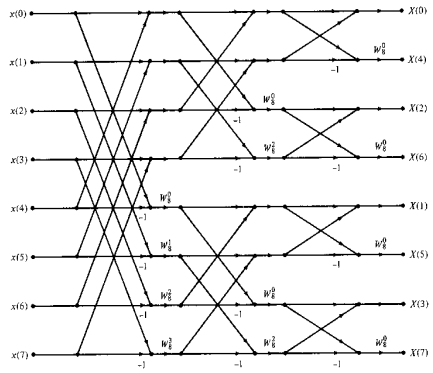

Fig2 *N* = 8-piont decimation-in-frequency FFT algorithm.

The discrete Fourier transform (DFT) X(k)of an N-point sequence x(n)is defined by

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \qquad k = 0, 1, \ldots, N-1$$
$$W_N = e^{-j(2\pi/N)}.$$

Instead of direct implementation of the computationally intensive DFT, the FFT algorithm is used to factorize a large point DFT recursively into many small point DFTs such that the overall operations involved can be drastically reduced. There are two well-known types of FFT algorithms called decimation-in-time (DIT) and decimation-infrequency (DIF) FFT which can be derived from each other by transposition.

Fast Fourier transform (FFT) is one of the most commonly used operations in the wireless communication applications, such as orthogonal frequency division multiple (OFDM) accesses, ultra wideband, digital video broadcast terrestrial, and signal processing application as well. A family of pipelined FFT architectures in which single-path delay feedback (SDF) and multipath delay commutator (MDC) are very popular. In certain circuits are proposed to reorder the bit reversed FFT output into normal order. The bit reversal circuits are proposed for different radices, a similar structure is proposed for variable length FFT, whose register complexity isN. These circuits are suitable for bit reversing the data from the pipelined FFT architecture. However, only the bit reversal structures are proposed, the bit reversal circuit is integrated to FFT architecture; as a result, the total register requirement of the design is reduced from 5N/2to2N. Two-, four-, and eight-parallel pipelined radix-2k decimation in frequency (DIF) feed forward FFT architectures are proposed and they need extra N registers to generate the output in natural order. Moreover, these two-, four-, and eight-parallel FFT architectures can start its operation only when x(n+N/2), x(n+3N/4),and x(n+7N/8)samples arrive, respectively. Therefore, hardware is underutilized and additional registers are required to store the first N/2, 3N/4, and 7N/8 samples.

II. LITERATURE SURVEY:

There are FFT architectures, which can handle multiple independent data streams.

However, all the data streams are processed by a single FFT processor. In four independent data streams are processed one by one. Similarly, eight data streams are processed at two domains. Thus, the outputs of multiple data streams are not available in parallel. In order to simultaneously process the data streams, more than one FFT processors need to be employed. In one to four data streams are processed using multiple data paths for wireless local area network application. Data of different data streams are interleaved to process them simultaneously. In low complexity FFT architectures are proposed but these architectures can process only real-valued signals (signals only with real part). Moreover, they generate two outputs per clock cycle and these outputs are not in natural order. Thus, most of the recent architectures require bit reversal structures to generate the outputs in natural order.

Disadvantages

•Performance is low
• usage of hardware element is high

We introduce a novel pipelined quick Fourier change (FFT) building design which is equipped for delivering the yield succession in typical request. Consolidated single delay commutator-input (SDC-SDF) radix2 pipelined quick Fourier change structuralengineering, which incorporateslog2 N −1 SDCstages, and 1 SDF stage. A solitary way defer commutator preparing component (SDC PE) has been proposed interestingly, It spares a mind boggling snake contrasted and the ordinary radix-2 butterfly unit. The new pipelined construction modeling can be assembled utilizing the proposed preparing component. The proposed construction modeling can prompt 100% equipment usage and 50% decrease in the general number of adders and multipliers needed in the ordinary pipelined FFT outlines. In request to create the yield grouping in typical request, we likewise show a touch reverser, which can accomplish a 50% diminishment in memory use and complex postponement memory 2N + 1.5 log2 N −1.5 .The existing system is to design the pipelined radix-2 based FFT architecture. This architecture is to consist of more no of multiplier architecture, so it affects the system performance level. The existing radix process is to modify the butterfly level for 16- point architecture and to implement the carry save adder architecture for multiplier adder section. The existing architecture is to analysis overall path delay for the real and complex multiplier system process. The existing system is single path delay commutator feed-back based FFT architecture and this work is to optimize the number of multiplier count level. Existing system is to identify the critical path section for overall 16-point radix-2 FFT architecture and to analysis weighted section for multiplier process. Existing system is to consist of bit reversal, adder/subtractor, and real multiplier and register architecture and to reduce the overall logical gate for proposed method. Today's electronic systems mostly run on batteries thus making the designs to be hardware efficient and power efficient. Application areas such as digital signal processing, communications, etc. employ digital systems which carryout complex functionalities. Hardware efficient and power efficient architectures for these systems are most require to achieve maximum performance. FastFourier Transform (FFT) is one of the most efficientwaysto implement Discrete Fourier Transform (DFT) due to its reduced usage of arithmetic units. DFT is one of those primary tools that are used for the frequency analysis of discrete time signals and to represent a discrete time sequence in frequency domain using its spectrum samples. The SDC PE consists of a data commutator, area add/sub unit,and an

optimum complex multiplier unit. In order to minimize the arithmetic resource of the SDC PE, the most significant factor is to maximize the arithmetic resource utilization vi are ordering the data sequences of the above three units. In the stage t, the data commutator shuffles its input data (Node−A) to generate a new data sequence (Node−B), whose index difference is N/2t, where t is the index of stage. We propose a novel pipelined FFT architecture which produces the output data in normal order. Using the proposed combined single path delay commutator (SDC)and single-delay feedback(SDF), the new pipelined architecture achieves 100%hardware utilization In order to produce the output sequence in normal order, a bit reverser saving 50% of memory requirement is proposed. Compared with the conventional pipelined FFT designs, the proposed architecture reduces the number of adders, multiplier s and hardware implementation complexity.

III. PROPOSED SYSTEM

3.1 Proposed pipelined FFT architecture:

The idea of computing an N-point FFT using two N/2-point FFT operations with additional one stage of butterfly operations is

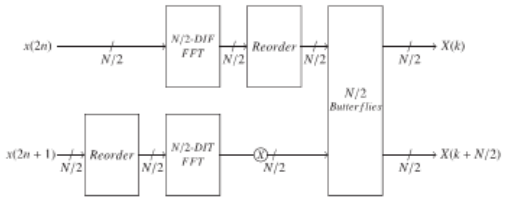

Fig.3. Idea of the proposed method

Shown in Fig. 3, which is not the exact architecture but provides the methodology? The reordering blocks in Fig. 3.1 are merely present to state that the N/2 odd samples (x(2n+1)) are reordered before the N/2-point DIT FFT operation and N/2evensamples(x(2n)) are reordered after the N/2-point DIF FFT operation. In order to compute the N-point DIT FFT from the outputs of two N/2-point FFTs, additional one stage of butterfly operations are performed on the results of the two FFTs. Thus, the outputs generated by additional butterfly stage are in natural order. For the purpose of simplicity, the proposed 16-point FFT architecture in Fig. 2 is explained. It has two eight-point MDC FFT architectures to process two data streams. The delay commutator units present at the left side of SW1 dissociate the odd and even samples. The shift registers in the delay commutator units, which receive inputs, are used to bit reverse the odd input samples. These shift registers are called reordering shift registers (RSRs). The RSR in the last stage store outputs from the eight-point DIF FFT and bit reverses them. The BF2 carries out two-parallel butterfly operations between the bit reversed data in the RSR in the last stage and outputs from the eight-point DIT FFT. Thus, the upper and lower BF2 in the last stage generate the FFT outputs of the first and the second data streams in normal order. The two data paths from SW2 are combined together, so the word length of the data path in last stage is twice and so thick lines are used for representing the data paths and registers.

3.1.1Operation of the Proposed Architecture

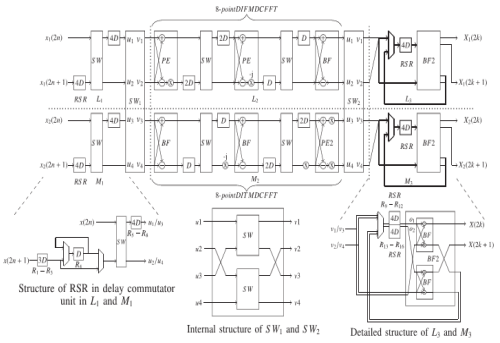

Fig.4.Proposed 16-point radix-2 FFT architecture with outputs in natural order.

The FFT architecture in Fig. 4 is divided into six levels (L1, L2, L3, M1, M2,andM3). The RSR registers in the levels L1andM1reorder the odd input data and the RSR registers in the levels

L3andM3reorder the partially processed even data. The eight-point DIF and DIT FFT operations are performed in the levels L2andM2, respectively. The data from L1andM1can be forwarded to L2 and M2, respectively, or vice versa with the help of SW1. Similarly, the data fromL2andM2can be forwarded toL3andM3, respectively, or vice versa with the help of SW2.SW1and SW2have two switches (SW) to swap the data path and propagate the data to different levels. During the normal mode, the switches (SW1or SW2) pass the data atu1, u2, u3,and u4tov1, v2, v3,andv4, respectively. However, during the swap mode, the switches (SW1 or SW2)pass the data at u1, u2, u3,andu4to v3, v4, v1,andv2, respectively. SW1 is in the swap mode during the first N/2 clock cycles and it is in the normal mode during N/2+1toN. On the other hand, SW2 is in the normal mode during the first N/2 clock cycles and it is in the swap mode during N/2+1toN.Thus,SW1and SW2 are in different modes at any time and change their modes for every N/2 clock cycles. Moreover, if there is transition of data between Ly andLy+1orMyandMy+1(where y can be 1 or 2), then the switches (SW1or SW2) are in the normal mode, and if there is transition of data between Ly and My+1 or My and Ly+1, then the switches(SW1or SW2) are in the swap mode. Like other control signals in the design, the control signals to the switches SW1 and SW2 are externally provided and these switch control signals swap at every N/2 clock cycles. The two input streams to the FFT processor are represented as X1 and X2. The odd and even samples of two input streams are disassociated by the delay commutator units in L1 and M1 (X1 is disassociated into {E1(i,j),O1(i,j)}, respectively, and X2 is disassociated into {E2(i,j),O2(i,j)}). In these representations, i defines the nature of the data and j defines the number of the data set whose FFT has to be computed. The even set of input data [x(0),x(2),x(4)...]isdefinedasE(1,j) and the odd set of input data [x(1),x(3),x(5)...]isdefinedasO(1, j). E(2,j)/O(2, j) is the set of scheduled or ordered even/odd data, which are ready to be fed to eight-point DIF/DIT FFT. The outputs of eight-point DIF/DIT FFT are defined asE(3,j)/O(3, j), which are fed to the third level for 16-point FFT computation. Table I explains the operation of FFT and the data propagation through different levels.

1) The first eight samples of X1 are loaded into the registers (4Din the upper and lower arms of delay commutator unit) in L1. After eight clock cycles, the switch (SW1)issetin the normal mode and the first eight samples of X2are loaded into the registers (4D)inM1. Simultaneously, E1(1,1) (even samples of X1) is forwarded from L1 to L2 as E1(2,1) to perform the eight-point FFT operation. The odd samples of X1 and X2 are bit reversed by the RSR in L1 and L2, respectively.

2) After eight clock cycles, the positions of the switches SW1and SW2are set in the swap mode and the normal mode, respectively. The odd samples (O1(1,1))ofX1are forwarded from L1to M2asO1(2,1)and the even samples (E2(1,1))of X2is forwarded from M1 to L2asE2(2,1). Simultaneously, E1(2,1)is forwarded from L2to L3asE1(3,1)and reordering is performed.

3) After eight clock cycles, SW1and SW2are set in the normal mode and the swap mode, respectively. The odd samples of X2(O2(1,1)) are forwarded from M1 to M2 as O2(2,1) and O1(2,1) is forwarded from M2 as O1(3,1) to L3 where the butterfly operations with E1(3,1) corresponding to the last stage (of the data stream X1) are performed. In the meantime,E2(2,1) from L2 is forwarded to M3 as E2(3,1) and reordering is performed in the RSR.

4) After eight clock cycles, the switch (SW2) is set to normal position to allow the partially processed odd samples (O2(3,1)) from M2to M3and perform the butterfly operations of the

last stage (of the data stream X2). Instead of using radix-2 FFTs, as shown in Fig. 2, any higher radix FFTs architecture can be used. In Fig. 5, two radix-2^3 64-point FFTs are used to realize 128-point FFT whose multiplier complexity is 4(log 8 (N/2)−.5) and working is almost the same as the 16-point FFT. The multiplier complexity of N-point radix-k FFT algorithm is 4(logk(N/2)−.5).

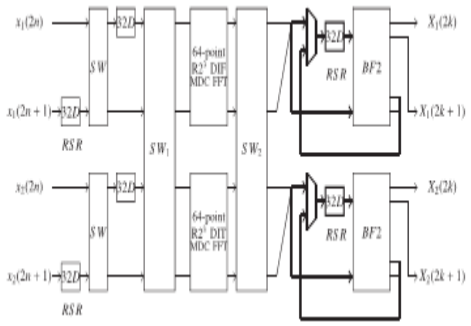

Fig.5. Proposed 128-point radix-2^3FFT architecture

3.1.2 Bit Reversing

The proposed architecture is inspired from the architecture whereN/2 data scheduling registers before the first butterfly unit are used to separate odd samples from the even samples and delay them to generate x(n) andx(n+N/2) in parallel. In the proposed architecture, this data scheduling registers are reused to bit reverse odd samples. Similarly, N/2 data scheduling registers are used before the last butterfly unit to store the partially processed even samples until the arrival of odd samples and here, these registers are reused to bit reverse the partially processed even samples (outputs from DIF FFT). Circuits that use multiplexers and shift registers for bit reversal are proposed..Nis the even power ofr, then the number of registers required to bit reverse N data is (√N−1)2 .IfNis the odd power of r, then the number of registers required to bit reverse Ndata is(√rN−1)(√N/r−1),wherer is the radix of the FFT algorithm. In the proposed architecture, these bit reversal circuits are incorporated in the data scheduling register to perform dual role.

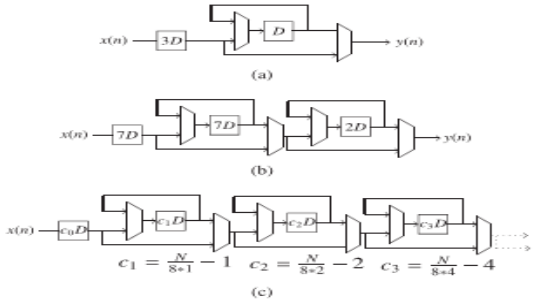

Fig.6.RSR. (a) RSR used in 16-point FFT architecture. (b) RSR used in 64-point FFT architecture. (c) RSR used in N-point FFT architecture.

The RSR used in the 16-point FFT and 64-point FFT architectures is shown in Fig. 6(a) and 6(b), respectively. Actually, these structures are present in the places of the shift registers marked with RSR. Generalized RSR for N-point is shown in Fig. 6(c) in which c0 is N/4−(√N/4−1)2orN/4−(√(Nr)/4−1)(√N/(4r)−1). These registers in c0 do not involve in reordering. The control signals tothe multiplexer in RSR are properly varied to interleave the data.If log2Nis even, log2N−2 multiplexers are required otherwiselog2Nmultipliers are required for bit reversal. For more details on bit reversal, may be referred.

TABLE

DATA FLOW THROUGH DIFFERENT LEVELS

Level	Time —>						
L_1	$E_1(1,1)$	$O_1(1,1)$	$E_1(1,2)$	$O_1(1,2)$			
L_2		$E_1(2,1)$	$E_2(2,1)$	$E_1(2,2)$	$E_2(2,2)$		
L_3			$E_1(3,1)$	$O_1(3,1)$	$E_1(3,2)$	$O_1(3,2)$	
M_1		$E_2(1,1)$	$O_2(1,1)$	$E_2(1,2)$	$O_2(1,2)$		
M_2			$O_1(2,1)$	$O_2(2,1)$	$O_1(2,2)$	$O_2(2,2)$	
M_3				$E_2(3,1)$	$O_2(3,1)$	$E_2(3,2)$	$O_2(3,2)$

In the proposed FFT architecture, the first N/4andthenext N/4 odd input data to DIF FFT

are separately bit reversed as they are required in parallel. Thus, N/4-point bit reversing algorithm is enough and the number of registers required to bit reverseN/4data is either (√N/4−1)2or(√(Nr)/4−1)(√N/(4r)−1)depending upon the power of two. In Fig. 2, the RSR (R1–R4)inM1 bit reverses the first N/4 odd input data [x(1),x(3),x(5),andx(7)]and store Them inR5–R8[x(1),x(5),x(3),andx(7)]. After that, the next N/4 odd input data x(9),x(11),x(13),andx(15)]are bit reversed in R1–R4 [x(9),x(13),x(11), and x(15)], which is explained in Table II. The delay commutator unit inL1 and M1 feeds the bit reversed odd input samples to u1andu2, andu3andu4 (in SW1), respectively. Similarly, in M3,theRSR(R9–R12) bit reverses the first N/4 output data [X(0),X(2),X(4), and X(6)]and the RSR (R13–R16) bit reverses the next N/4 output data [X(8),X(10),X(12), and X(14)] of DIT FFT separately, which is explained in Table III. Thus, the RSR inL3andM3bit reverses the partially processed even data samples fromv1 andv2,andv3 and v4(in SW2), respectively, and feeds to BF2 (via o1ando2).

TABLE-II

BIT REVERSAL OPERATION IN THE LEVELS L_1 AND M_1

Clk	$x(2n)$	$x(2n+1)$	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	u_3	u_4
0	x(0)	x(1)	-	-	-	-	-	-	-	-	-	-
1	x(2)	x(3)	x(1)	-	-	-	x(0)	-	-	-	-	-
2	x(4)	x(5)	x(3)	x(1)	-	-	x(2)	x(0)	-	-	-	-
3	x(6)	x(7)	x(5)	x(3)	x(1)	-	x(4)	x(2)	x(0)	-	-	-
4	x(8)	x(9)	x(7)	x(5)	x(3)	x(1)	x(6)	x(4)	x(2)	x(0)	x(0)	x(8)
5	x(10)	x(11)	x(9)	x(7)	x(5)	x(3)	x(1)	x(6)	x(4)	x(2)	x(2)	x(10)
6	x(12)	x(13)	x(11)	x(9)	x(7)	x(3)	x(5)	x(1)	x(6)	x(4)	x(4)	x(12)
7	x(14)	x(15)	x(13)	x(11)	x(9)	x(7)	x(3)	x(5)	x(1)	x(6)	x(6)	x(14)
8	-	-	x(15)	x(13)	x(11)	x(9)	x(7)	x(3)	x(5)	x(1)	x(1)	x(9)
9	-	-	-	x(15)	x(13)	x(11)	-	x(7)	x(3)	x(5)	x(5)	x(13)
10	-	-	-	-	x(15)	x(11)	-	-	x(7)	x(3)	x(3)	x(11)
11	-	-	-	-	-	x(15)	-	-	-	x(7)	x(7)	x(15)

TABLE-III

BIT REVERSAL OPERATION IN THE LEVELS L_3 AND M_3

Clk	v_3	v_4	R_9	R_{10}	R_{11}	R_{12}	R_{13}	R_{14}	R_{15}	R_{16}	o_1	o_2
0	X(0)	X(4)	-	-	-	-	-	-	-	-	-	-
1	X(1)	X(5)	X(0)	-	-	-	X(4)	-	-	-	-	-
2	X(2)	X(6)	X(1)	X(0)	-	-	X(5)	X(4)	-	-	-	-
3	X(3)	X(7)	X(2)	X(1)	X(0)	-	X(6)	X(5)	X(4)	-	-	-
4	-	-	X(3)	X(2)	X(1)	X(0)	X(7)	X(6)	X(5)	X(4)	X(0)	X(8)
5	-	-	-	X(3)	X(2)	X(1)	-	X(7)	X(6)	X(5)	X(4)	X(12)
6	-	-	-	-	X(3)	X(2)	-	-	X(7)	X(6)	X(2)	X(10)
7	-	-	-	-	-	X(3)	-	-	-	X(7)	X(6)	X(14)

IV. RESULTS

SIMULATION RESULT:

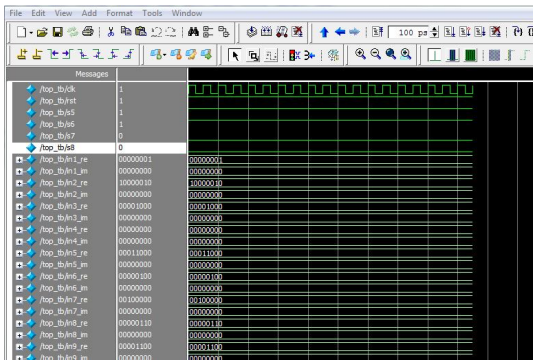

Fig7: Simulation for top module

SYNTHESIS RESULTS:

The developed project is simulated and verified their functionality. Once the functional verification is done, the RTL model is taken to the synthesis process using the Xilinx ISE tool. In synthesis process, the RTL model will be converted to the gate level netlist mapped to a specific technology library.

RTL SCHEMATIC:

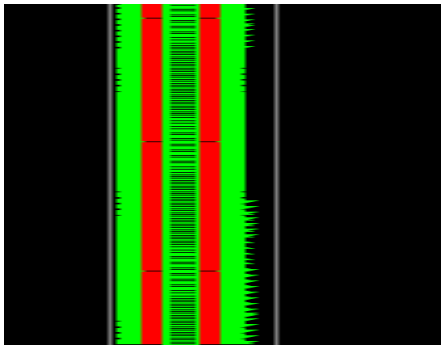

Fig 8:RTL for 64-bit

DESIGN SUMMARY:

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	2649	3584	73%
Number of 4 input LUTs	5045	7168	70%
Number of bonded IOBs	8075	221	3653%
Number of GCLKs	1	8	12%

Fig 9: Device utilization for 128-bit

TIMING REPORT:

```
Timing constraint: Default path analysis
  Total number of paths / destination ports: 17590 / 4096
--------------------------------------------------------------------------------
Delay:               16.745ns (Levels of Logic = 4)
  Source:            s5 (PAD)
  Destination:       Z27i<7> (PAD)

  Data Path: s5 to Z27i<7>
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O            403   0.821   3.978  s5_IBUF (s5_IBUF)
     BUF:I->O             404   0.551   4.323  s5_IBUF_4 (s5_IBUF_4)
     LUT2:I0->O             2   0.551   0.877  m11/Y27i<7>1 (Z27i_7_OBUF)
     OBUF:I->O                  5.644          Z27i_7_OBUF (Z27i<7>)
    ----------------------------------------
    Total                     16.745ns (7.567ns logic, 9.178ns route)
                                       (45.2% logic, 54.8% route)
```

Fig10: Timing report for 128-bit

V. CONCLUSION:

This brief has presented a novel FFT processor whose outputs are generated in the natural order. The proposed processor can process two independent data streams simultaneously, and makes it suitable for many high-speed real-time applications. The bit reversal circuit present in prior designs is eliminated by integrating two FFT processors and the registers, which are present in the architecture, are reused for bit reversal. As a result, the need of additional registers to bit reverse the outputs is avoided. Moreover, the proposed architecture provides throughput higher than the prior architectures. These attributes make the proposed FFT processor superior in sense of hardware complexity and performance.

REFERENCES:

[1] S. He and M. Torkelson, “A new approach to pipeline FFT processor,”in Proc. 10th Int. Parallel Process. Symp., 1996, pp. 766–770.

[2] Y. Chen, Y.-W. Lin, Y.-C. Tsao, and C.-Y. Lee, “A 2.4-Gsample/sDVFS FFT processor for MIMO OFDM communication systems,”IEEE J. Solid-State Circuits, vol. 43, no. 5, pp. 1260–1273,May 2008.

[3] S.-N. Tang, C.-H. Liao, and T.-Y. Chang, “An area- and energyefficient multimode FFT processor for WPAN/WLAN/WMAN systems,”IEEE J. Solid-State Circuits, vol. 47, no. 6, pp. 1419–1435,Jul. 2012.

[4] P. P. Boopal, M. Garrido, and O. Gustafsson, “A reconfigurable FFTarchitecture for variable-length and multi-streaming OFDM standards,”in Proc. IEEE ISCAS, May 2013, pp. 2066–2070.

[5] K.-J. Yang, S.-H. Tsai, and G. C. H. Chuang, “MDC FFT/IFFTprocessor with variable length for MIMO-OFDM systems,”IEEETrans.Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 4, pp. 720–731,Apr. 2013.

[6] Y.-N. Chang, “An efficient VLSI architecture for normal I/O orderpipeline FFT design,”IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 55,no. 12, pp. 1234–1238, Dec. 2008.