An Effective Redix-2 FFT Processor Based On Parallel Processing of Input Data Streams

Y. Varsha, MS.P.Rajeshwari, Mr.B.Sivakumar,
PG Scholar, Assistant Professor, Professor & HOD,
Department of Electronics and Communication Engineering,
Ellenki Institute of Engineering and Technology,
Patelguda, Patancheru, Medak (Dist), Telangana, India

Abstract—Nowadays, many 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 an N/2-point decimation in time FFT and an 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.

Keywords— Bit reversal, bit reversed order, fast Fourier transform (FFT), multipath delay commutator (MDC) FFT, normal order.

I. INTRODUCTION

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 wide-band, digital video broadcast terrestrial, and signal processing application as well. A family of pipelined FFT architectures is discussed in which single-path delay feedback (SDF) and multipath delay commutator (MDC) are very popular. There are applications, such as image processing, array signal processing, multiple-input–multiple-output OFDM, and so on, in which more than one data stream need to be processed. Therefore, simultaneous multiple FFT operations are required and a dedicated bit reversal circuit is also required to generate the outputs in natural order.

There are FFT architectures, which can handle multiple independent data streams. However, all the data streams are processed by a single FFT processor. 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 [3], 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. Nevertheless, the architectures do not have any specific bit reversal circuit. 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 is $N$. 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 $5N/2$ to $2N$. Two-, four-, and eight-parallel pipelined radix-$2^k$ 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 $x(n+N/2), x(n+3N/4), \text{ and } x(n+7N/8)$ samples arrive, respectively. Therefore, hardware is underutilized and additional registers are required to store the first $N/2, 3N/4, \text{ and } 7N/8$ samples. Modified MDC FFT architectures with a new data scheduling method and a rearranging structure are proposed in which the drawbacks of [10] are eliminated. The architectures operated at the frequency of incoming sample rate but the architecture operates half the clock frequency to generate the same. Thus, the throughput of the architecture is doubled, if all the architectures are operated at the same speed. Similarly, a combined single path delay commutator SDF FFT architecture with I/O in natural order is proposed in which the bit reversal is carried out only with $N/2$ registers. However, its throughput is low and the required number of register is high. 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. The proposed architecture is designed to process two independent data streams simultaneously with less amount of hardware. The odd inputs, which are in natural order, are first bit reversed and then they are processed by $N/2$-point decimation in time (DIT) FFT. The even samples are directly processed by $N/2$-point DIF FFT, so its outputs are in bit reversed order. Therefore, the outputs of $N/2$-point DIF FFT are bit reversed. The outputs of the two $N/2$-point FFTs are further processed by the two-parallel butterflies to generate the outputs of $N$-point FFT in natural order. The bit reversing is carried out by the scheduling registers, which are actually used to delay the samples for performing the butterfly operations. Thus, the FFT architecture does not use any dedicated circuit to bit reverse the data. As a result, the proposed architecture requires less number of registers than the prior FFT designs.

II. 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 shown in Fig. 1, which is not the exact architecture but provides the methodology. The reordering blocks in Fig. 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 $N/2$ even samples ($x(2n)$) 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 SW$_{1}$ 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 SW$_{2}$ 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.

A. Operation of the Proposed Architecture

The FFT architecture in Fig. 2 is divided into six levels ($L_1, L_2, L_3, M_1, M_2$, and $M_3$). The RSR registers in the levels $L_1$ and $M_1$ reorder the odd input data and the RSR registers in the levels $L_3$ and $M_3$ reorder the partially processed even data. The eight-point DIF and DIT FFT operations are performed in the levels $L_2$ and $M_2$, respectively. The data from $L_1$ and $M_1$ can be forwarded to $L_2$ and $M_2$, respectively, or vice versa with the help of SW$_1$. Similarly, the data from $L_2$ and $M_2$ can be forwarded to $L_3$ and $M_3$, respectively, or vice versa with the help of SW$_2$. SW$_1$ and SW$_2$ have two switches (SW) to swap the data path and propagate the data to different levels. During the normal mode, the switches (SW$_1$ or SW$_2$) pass the data at $u_1, u_2, u_3, \text{and } u_4$ to $v_1, v_2, v_3, \text{and } v_4$, respectively. However, during the swap mode, the switches (SW$_1$ or SW$_2$) pass the data at $u_1, u_2, u_3, \text{and } u_4$ to $v_3, v_4, v_1, \text{and } v_2$, respectively. SW$_1$ is in the swap mode during the first $N/2$ clock cycles and it is in the normal mode during $N/2 + 1$ to $N$. On the other hand, SW$_2$ is in the normal mode during the first $N/2$ clock cycles and it is in the swap mode during $N/2 + 1$ to $N$. Thus, SW$_1$ and SW$_2$ 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 $L_j$ and $L_{j+1}$ or $M_j$ and $M_{j+1}$ (where $y$ can be 1 or 2), then the switches (SW$_1$ or SW$_2$) are in the normal mode, and if there is transition of data between $L_j$ and $M_{j+1}$ or $M_j$ and $L_{j+1}$, then the switches (SW$_1$ or SW$_2$) are in the swap mode. Like other control signals in the design, the control signals to the switches SW$_1$ and SW$_2$ 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 $X_1$ and $X_2$. The odd and even samples of two input streams are disassociated by the delay commutator units in $L_1$ and $M_1$ ($X_1$ is disassociated into $\{E_1(i,j),O_1(i,j)\}$, respectively, and $X_2$ is disassociated into $\{E_2(i,j),O_2(i,j)\}$). 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 $[x(0),x(2),x(4)\ldots]$ is defined as $E(1,j)$ and the odd set of input data $[x(1),x(3),x(5)\ldots]$ is defined as $O(1,j)$. $E(2,j)/O(2,j)$ 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 as $E(3,j)/O(3,j)$, which are fed to the third level for 16-point FFT computation. Table 1 explains the operation of FFT and the data propagation through different levels.

1) The first eight samples of $X_1$ are loaded into the registers (4D in the upper and lower arms of delay commutator unit) in $L_1$. After eight clock cycles, the
mal mode and the first eight samples of X2 are loaded into the registers (4D) in M1. Simultaneously, $E1(1, 1)$ (evensamples 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 SW1 and SW2 are set in the swap mode and the normal mode, respectively. The odd samples ($O1(1, 1)$) of X1 are forwarded from L1 to M2 as $O1(2, 1)$ and the even samples ($E2(1, 1)$) of X2 is forwarded from M1 to L2 as $E2(2, 1)$. Simultaneously, $E1(2, 1)$ is forwarded from L2 to L3 as $E1(3, 1)$ and reordering is performed.

3) After eight clock cycles, SW1 and SW2 are 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 M2 to M3 and 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 high-radix FFTs architecture can be used. In Fig. 3, two radix-23 64-point FFTs are used to realize 128-point FFT whose multiplier complexity is $4(\log(8N/2) - .5)$ and working is almost the same as that of the 16-point FFT. The multiplier complexity of N-point radix-k FFT algorithm is $4\log k (N/2) - .5$.

B. Bit Reversing: The proposed architecture is inspired from the architecture in [7] where N/2 data scheduling registers before the first butterfly unit are used to separate odd samples from the even samples and delay them to generate $x(n)$ and $x(n + N/2)$ 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 in [7] and here, these registers are reused to bit reverse the partially processed even samples (outputs from DIF FFT). In [8], circuits that use multiplexers and shift registers for bit reversal are proposed. According to [8], if $N$ is the even power of $r$, then the number of registers required to bit reverse $N$ data is $((\sqrt{N} - 1)/2)$. If $S$ is the odd power of $r$, then the number of registers required to bit reverse $N$ data is $((\sqrt{N} - 1)/((\sqrt{N} - 1)/r) - 1)$, where $r$ 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. The RSR used in the 16-point FFT and 64-point FFT architecture is shown in Fig. 4(a) and (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. 4(c) in which $c0 < N/4 = (\sqrt{N}/4 - 1/2)$ or $N/4 < (\sqrt{Nr}/4 - 1)/((\sqrt{N}/(4r) - 1)$). These registers in $c0$ do not involve in reordering. The control signals to the multiplexer in RSR are properly varied to interleave the data. If $\log_2 N$ is even, $\log_2 N - 2$ multiplexers are required otherwise $\log_2 N$ multipliers are required for bit reversal. In the proposed FFT architecture, the first $N/4$ and the next $N/4$ 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 reverse N/4 data is either $((\sqrt{N} - 1)/2$ or $((\sqrt{Nr}/4 - 1)/((\sqrt{N}/(4r) - 1)$ depending upon the power of two. In Fig. 2, the RSR ($R1$–$R4$) in M1 bit reverses the first $N/4$ odd input data [$x(1)$, $x(3)$, $x(5)$, and $x(7)$] and stores them in $R5$–$R8$ [$x(1)$, $x(5)$, $x(3)$, and

Available online: https://edupediapublications.org/journals/index.php/IJR/
After that, the next \( N/4 \) odd input data \( [x(9), x(11), x(13), \text{and } x(15)] \) are bit reversed \( R1–R4 \) \( [x(9), x(13), x(11), \text{and } x(15)] \). The delay commutator unit in \( L1 \) and \( M1 \) feeds the bit-reversed odd input samples to \( u1 \) and \( u2 \), and \( u3 \) and \( u4 \) (in SW1), respectively. Similarly, in \( M3 \), the RSR (\( R9–R12 \)) bit reverses the first \( N/4 \) output data \( [X(0), X(2), X(4), \text{and } X(6)] \) and the RSR (\( R13–R16 \)) bit reverses the next \( N/4 \) output data \( [X(8), X(10), X(12), \text{and } X(14)] \) of DIT FFT separately, which is explained in Table III. Thus, the RSR in \( L3 \) and \( M3 \) bit reverses the partially processed even data samples from \( v1 \) and \( v2 \), and \( v3 \) and \( v4 \) (in SW2), respectively, and feeds to BF2 (via \( o1 \) and \( o2 \)).

![Fig. 2. Proposed 16-point radix-2 FFT architecture with outputs in natural order.](image)

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

![Fig. 3. Proposed 128-point radix-2\(^3\) FFT architecture.](image)

Fig. 3. Proposed 128-point radix-2\(^3\) FFT architecture.

\[
\begin{align*}
\frac{N}{4} &= 16 \\
\frac{N}{8} &= 8 \\
\frac{N}{32} &= 4 \\
\frac{N}{256} &= 1
\end{align*}
\]

![Fig. 4. 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.](image)

Fig. 4. 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.

## III. SIMULATION RESULTS
V. CONCLUSION

This transient has given a unique FFT processor whose outputs area unit generated within the world. The projected method will process 2 freelance information streams at the same time and makes it appropriate for several high-speed periods of time applications.

The bit reversal circuit gift in previous styles is eliminated by integration 2 FFT processors and also the registers, that area unit gift within the design area unit reused for bit reversal. As a result, the necessity of extra registers to bit reverse the outputs is avoided. Moreover, the projected design provides outturn on top of the previous architectures. These attributes create the projected FFT processor superior in sense of hardware quality and performance.

REFERENCES


