

Design of dynamically reconfigurable fully optimized low power FFT architecture for MC-CDMA receiver

R. Sakthivel^{a)} and Harish M Kittur

VLSI Division, VIT University, Vellore-632014, India

a) rsakthivel@vit.ac.in

Abstract: This work aims for providing low power, high performance receiver architecture for MC-CDMA receiver. We provide a dynamically reconfigurable FFT architecture to switchover from 64 points to 16 points based on the channel parameters(i.e.)delay spread, which otherwise would have been designed for the worst case FFT length which are our potential source of power reduction. Clock gating with FFT point reconfigurability technique is used to switchover from low length to high length FFT or vice versa. This work explores different possible ways of power reduction in the major internal blocks of FFT based on the data dependency. The simulation results are compared in terms of power dissipation, area and performance with the existing system. Results show that we could achieve a overall power reduction of more than 50% and improvement in the performance to about 24% with slight increase in area of 4.13%. The proposed architecture is modeled using Verilog HDL, simulated using NCLaunch of Cadence and synthesized with TSMC 180 nm and also with 45 nm Technology.

Keywords: dynamically reconfigurable, delay spread, clock gating and power dissipation

Classification: Integrated circuits

References

- [1] W. Han, A. T. Erdogan, T. Arslan and M. Hasan: IEEE Int. Symp. Circuit and Systems (ISCAS) **5** (2005) 5274.
- [2] W. Han, A. T. Erdogan, T. Arslan and M. Hasan: IEEE Signal Processing Systems Workshop (SIPS) (2004).
- [3] K. Maharatna, E. Grass and U. Jagdhold: IEEE J. Solid-State Circuits **39** [3] (2004) 484.
- [4] W. Han, A. T. Erdogan, T. Arslan and M. Hasan: ETRI J. **30** [3] (2008) 451.
- [5] M. B. Bevan: IEEE J. Solid-State Circuits **34** [3] (1999) 380.

1 Introduction

The Fast Fourier Transform (FFT) is one of the fundamental operations in the field of DSP, telecommunications, speech and image processing, etc. Recently, the FFT is used as one of the key components in OFDM based wideband communication systems. It is desirable for wireless receivers to adapt their operation rather than only be targeted for fixed or worst case scenario. These receivers will have to be flexible enough to accommodate various operating conditions while meeting computational requirements and simultaneously achieve low-power consumption. The combined requirement on providing high-performance and flexibility with low-power consumption is the focus of the architecture implemented in this paper. The commutator and complex multiplier contribute a dominating part of the whole power consumption and acts as a leading actor with the increase of FFT size. The complex multiplier has high switching activity between successive coefficients fed to it and by coefficient ordering its power consumption can be reduced.

2 Previous works

In [1], the authors proposed a low-power FFT architecture based on multirate signal processing and asynchronous circuit technology. The communication is localized, and the sharing of the global memory is eliminated. To reduce the number of operations in FFT and thus reduce power consumption, some researchers use shifters and adders to replace the complex multiplications by some special constant coefficients. The authors of [2] employ seven shift-and-add units to carry out seven multiplications in parallel, each by a constant coefficient. Papers [2, 3] proposed a multiplier less architecture based on common sub expression sharing, which replaces the complex multipliers in FFTs, and a low-power commutator architecture, which reduces the number of memory accesses. In addition to the pipelined FFT, the parallel-pipelined FFT is a good solution for applications requiring high throughput and high power efficiency. This paper aims for developing an FFT Processor with re-configurability, low-power and high performance. The approach is to implement basic parts that can be combined together to form a reconfigurable architecture, such that these parts should be reusable and the architecture should be adjustable in several parameters. In our case for e.g. FFT size (in case of FFT Processor) [5].

3 Proposed reconfigurable system and its architecture

A reconfigurable 64-point R4SDC pipelined FFT processor architecture comprises of three radix-4 stages as shown in Figure 1. Re-configurability is achieved by inserting a multiplexer namely MUX I for selectively enabling the stage-64 and stage-16 for directly routing the input data depending upon the required FFT size. The FFT processor can act as a 16-point processor by feeding the input data directly into stage-16 and clocking down the stage-64. This is accomplished by selecting the input data rather than the output of stage-64 by the external select input S16 of MUX I. Moreover, the gated clock

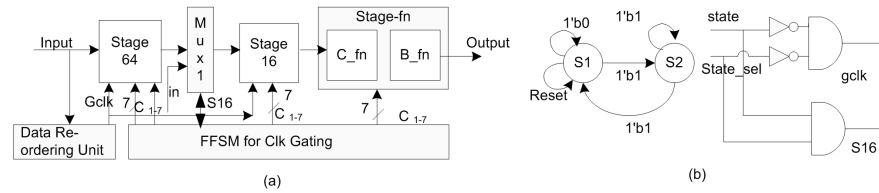


Fig. 1. (a) Architecture of 64-point radix-4 reconfigurable pipelined FFT processor Stage-64: 64-point FFT stage; Stage-16: 16-point FFT stage; C_fn: Commutator structure B_fn: Butterfly structure. (b) FFSM (FFT Finite State Machine).

input (Gclk) to stage-64 is also disabled by the FFT Finite State Machine (FFSM). The basic stage of the FFT processor comprises of a commutator, a butterfly unit and a complex multiplier. The last stage contains just the commutator (*C_fn*) and the butterfly unit (*B_fn*). The FFSM is responsible for generating all the control signals. The data reordering unit explores the data dependencies of the input data and reduces the computation involved in the FFT structure. If the input data follows the following pattern say

1st input - '1011110' and 2nd input - '0000000'

Then the corresponding multiplier and adder blocks in the FFT blocks can be disabled so as to reduce the computation, similarly if the 1st data is followed by '0000001' then the multiplier can be disabled and the multiplicand value can be passed on to the next stage. So these types of data dependencies of the input data have been exploited for the reduction of power in the computational elements of the internal blocks of the FFT architecture.

3.1 Algorithm for dynamic reconfiguration of FFT points

The dynamic reconfigurability of FFT point follows this algorithm

1. Get the variation in the channel parameters like Delay spread (D) and Signal to Noise Ratio (SNR).
2. Let a1 and a2 be the threshold value of SNR and b1 and b2 be the threshold values of delay spread (D).
3. Assign the values of a1 and a2 for the threshold values of SNRs of 16 point and 64 point FFT similarly b1 and b2 values for D based on the channel varying condition.
4. If channel SNR > a1 then enable the 64 point FFT else disable the 64 point and enable only the 16 point FFT.
5. If channel D > b1 then enable the 64 point FFT else disable the 64 point and enable only the 16 point FFT.
6. Repeat step 4 and 5 for different value of varying channel parameters like SNR and D.

3.2 Canonic Signed Digit (CSD)

The complex multiplication in the FFT is done by using CSD and sub expression sharing technique. The advantage of CSD form is that no value has more than $(N+1)/2$ non-zero bits. The filter coefficient constant 5a82 is represented by two's complement format, and 7641 and 30fb are represented by CSD format as follows: 5a82 (0101101010000010), 7641 (1000_10_1001000001), and 30fb (010_1000100000_10_1). The mixed use of CSD and two's complement minimizes the number of addition/shift operations as shown in Table I. We can use the shift-add based implementation of multiplications with the three constants to carry out those non-trivial complex multiplications. According to the previous representation, these multiplications with the three constants are given by:

$$\begin{aligned} 5a82X &= X \ll 1 + X \ll 7 + X \ll 9 + X \ll 11 + X \ll 12 + X \ll 14 \\ 7641X &= X + X \ll 6 - X \ll 9 - X \ll 11 + X \ll 15 \\ 30fbX &= -X - X \ll 2 + X \ll 8 - X \ll 12 + X \ll 14 \end{aligned}$$

Where X represents the input data [4].

3.3 Common sub expression sharing

Common sub expression sharing shares the sub expression among several multiplication-accumulation operations in order to reduce the total number of operations [4]. This approach is very effective for reducing the hardware cost of multiple constant multiplications, especially for the filter-like operation. As discussed in section 2.2, the number of operations required for the computation of 5a82X, 7641X, and 30fbX are shown in Table I. In all nontrivial coefficient multiplications, the proportions of the multiplications referring to 5a82, 7641, and 30fb are 50%, 25%, and 25%, respectively. Hence, the average operations for a nontrivial coefficient multiplication are 4.5 additions, 2.5 subtractions, and 7 shifts. The weights A_i are the filter coefficients.

$$Y(i) = \sum_{i=0}^n A_i X_{n-i} \quad (1)$$

The block diagram of multiplier less unit is depicted in Fig. 2 (a). Only data that has to be multiplied with non-trivial complex coefficients is fed into the shift-and-add units. Two shift-and-add units are needed for both the real part (X_r) and the imaginary part (X_i). There are two single-bit control signals, s_6 and s_7 , in the multiplier less unit. Signal s_6 indicates whether the input data corresponds to a nontrivial complex coefficient. When signal s_7 is asserted to logic 1 state, the real and imaginary parts of the input data are swapped, and the imaginary part is inverted. Otherwise, the swap unit passes the input data unchanged. Here, in the multiplier less unit, 22 adders are used to substitute the four real multipliers in the complex multiplier unit. The data control unit passes the data to the multiplierless unit based on the input sequence. If subsequent input is zero it disables the complete unit and provides the result '0'. If '1' follows it simply gives the coefficient result thereby reducing the computational power.

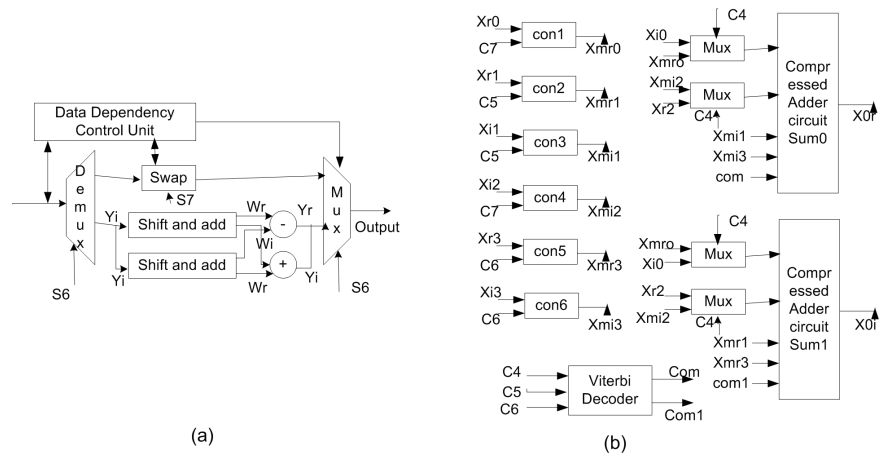


Fig. 2. (a) Block diagram of the multiplier less unit in the 16-point R4SDC FFT (b) Proposed Low Power Butterfly Architecture based on [4].

Table I. Operations required before and after common sub expression sharing [4].

Operation	Before Common sub expression			After Common sub expression		
	5a82X	7641X	30fbX	5a82X	7641X	30fbX
Addition	5	2	2	2	1	0
Subtraction	0	2	3	0	1	2
Shift	6	4	4	3	3	2

3.4 IDR commutator

The proposed architecture uses the same IDR commutator discussed in Ref. [4], which uses dual port RAMs as FIFO elements, however, the interconnection topology among the RAM blocks is different from that of the conventional approach. The RAM blocks (DM0, DM1,...DM5) are appropriately enabled and disabled so as to provide low power. Table II illustrates which RAM blocks are enabled for write access during each period. It can be seen that there are at most three RAM blocks selected in a given period. For stage t , when M_t is equal to 1, new N_{t-1} input data is processed. The first N_t data will be written into DM0. The previous N_t data stored in DM0 will be read out and written into DM2 for vacating space for the new data. The same applies for DM2 and DM4. The other three RAM blocks (DM1, DM3 and DM5) will be disabled for write access during this period. For $M_t = 0$ and 3, the number of RAM blocks enabled is two, because the previous data stored in DM2 and DM3 are no longer needed for subsequent outputs. Therefore, during the four periods, each RAM is enabled $5/3$ times on average. Whereas, for Dual-port RAM(DR) and Triple-port RAM(TR) architectures this corresponds to 4 and $10/3$ times respectively. Hence our new commutator architecture is significantly more power efficient compared to other commutator architectures.

Table II. RAMS selection in different period [4].

Mt	0	1	2	3
RAMs Selected	DM1,DM3	DM0,DM2,DM4	DM1,DM3,DM5	DM0,DM2

Table III. Comparison of Power, Area and Timing.

Architecture	Multiplierless (mW)	Commutator (mW)	Butterfly (mW)	16 Point FFT (mW)	Area (um ²) 16 Point	Timing (ns)
CFFTWR *	15.27	6.52	8.43	64.24	193267.2	8.32
WHAND [2]	4.54#	3.75	3.46	40.9	-	-
Ahmet T [4]	8.55#	2.83	2.99	30.20	-	-
PFFTW **	1.17	2.83	1.43	14.68	201432.1	6.29

* Conventional FFT without Reconfiguration; ** Proposed FFT with Reconfiguration; # With Multiplier.

3.5 Proposed low power butterfly architecture

The architecture showed in Fig. 2 (b) aims to reduce power by removing unwanted sub blocks from previous architecture [4]. This architecture was constructed based on Radix-4 decimation in time. Input data's are two's complemented based on the control signals c4, c5 and c6. This architecture uses viterbi decoder which is efficient in the aspect of power and compressed adders are used in order to perform the summation. Multiplexers are removed by replacing select lines and two's complement blocks are used instead of using one's complement.

4 Simulation and results discussion

The proposed architecture has been modeled with Verilog HDL and the system level simulation has been done using NC Launch of Cadence. The design has been synthesized with 180 and 45 nm (slow.lib) tech file from TSMC. The backend design has been completed using Cadence SOC Encounter with 180 nm tech file from TSMC. The Nano routed view of our design has been shown in the Fig. 3 (a) and the power comparison of the proposed FFT sub blocks with some of the existing work is shown in Fig. 3 (b). The selection of FFT size depends on the value of the delay spread, for the delay spread produced by De-spreading module in the combiner block between 350 ns to 900 ns we can transmit it using 64 point FFT. Whereas for delay spread above 900 ns 64-point FFT switches to 16 point FFT by using clock gating technique to save the power consumption. Table III compares the area, power and speed of the proposed reconfigurable 16 point FFT system with the previous existing work. Table IV shows the evaluation results of reconfigurable and fixed 16, 64 points FFT architecture with 180 nm and 45 nm technology. The FFT blocks have been optimized individually and their comparison with the conventional architectures is shown in Fig. 4.

5 Conclusions

In this paper, a new reconfigurable FFT architecture has been designed and it is made efficient in the aspect of power and speed by applying data reorder-

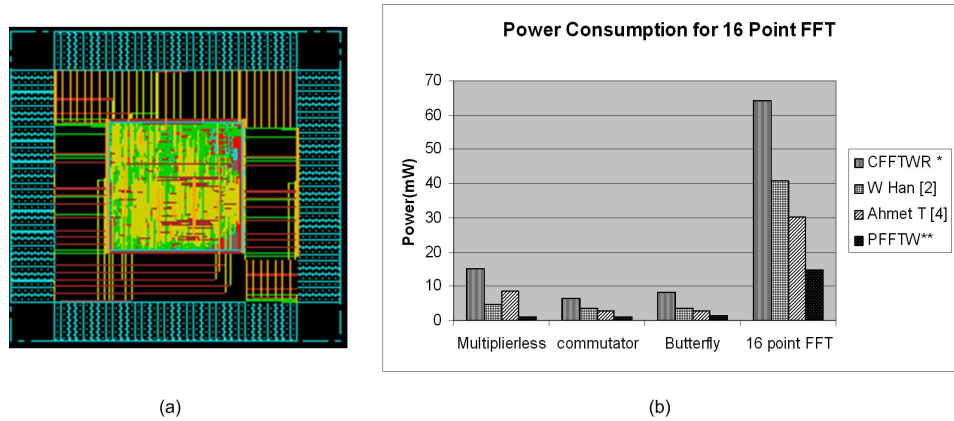
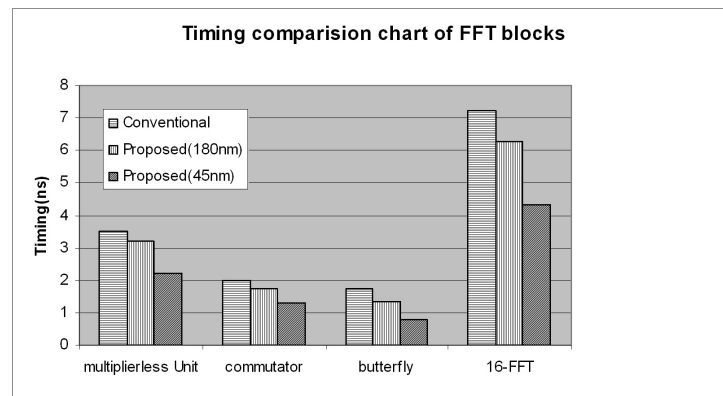
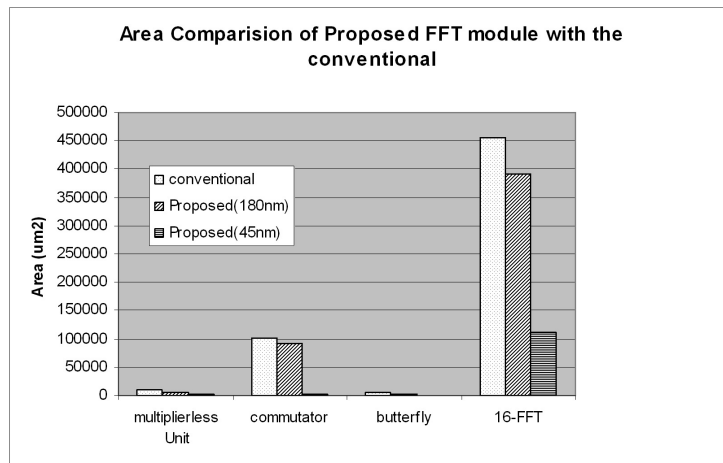


Fig. 3. (a) Complete IC after Nano Routing (b) Power comparison chart.



(a)



(b)

Fig. 4. (a) Timing comparison chart of proposed FFT blocks with the conventional (b) Area comparison of the proposed FFT blocks with the conventional.

ing, clock gating and selective enabling and disabling of FFT blocks based on the delay spread, SNR and BW. The data dependency is being explored for the reduction of computation in the internal blocks of the FFT structure.

Table IV. Evaluation of Dynamic Reconfiguration.

Points of FFT	Design Type	Power	Area	Timing
		(mW)	(μm^2)	(ns)
16.Point FFT	Without Dynamic Reconfiguration (180 nm) (CFFTWR)	64.24	193267.2	8.32
	With Dynamic Reconfiguration(180 nm)	14.68	201432.1	6.29
	With Dynamic Reconfiguration(45 nm)	0.76	3414.45	4.32
64.Point FFT	Without Dynamic Reconfiguration (180 nm)	169.29	202022.7	7.5
	With Dynamic Reconfiguration(180 nm)	47.23	251814.6	9.2
	With Dynamic Reconfiguration(45 nm)	2.14	4134.28	6.9

The comparison result is made for 16 point FFT; further the design is made, simulated and synthesized for 64 point with re-configurability. The simulation results shows more than 50% power reduction with 24% increase in the speed at the cost of 4% increase in the area for 16.Point Reconfigurable FFT architecture.