



PERFORMANCE EVALUATION OF SPLIT RADIX-2 LOW POWER AND HIGH SPEED FFT PROCESSOR

Sainathreddy. T¹, Komalendra. T¹, Ravi. T² and Mathan. N²

¹Electronics and Communication Engineering, Sathyabama University, Chennai, India

²Department of Electrical and Computer Engineering, Sathyabama University, Chennai, India

E-Mail: redrysainath60@gmail.com

ABSTRACT

Split-radix fast Fourier transform (SRFFT) is mainly for the implementation of a low-power FFT processor. FFT is an exceedingly proficient system to diminish calculation time and furthermore enhances the execution. In FFT calculations, SRFFT has less number of math Operations. With the approach of new innovation in the fields of VLSI and correspondence, there is likewise a constantly developing interest for rapid preparing and low range plan. Twiddle variables is required in FFT tending to processors. The flag stream chart of SRFFT is the same as radix-2 FFT thus regular tending to plans is utilized as a part of SRFFT. Notwithstanding, it has uncalled for game plan of Twiddle variables, and precludes the application from claiming radix 2 address era strategies. We demonstrate that SRFFT can be figured by utilizing a changed radix 2 butterfly unit. The proposed configuration accomplishes more than 20% lower control utilization while figuring a 1024-point complex-esteemed change. The proposed calculation comprises of blended radix butterflies, whose structure is more customary than the regular split radix calculation.

Keywords: SRFFT, FFT, radix 2, VLSI, power consumption.

1. INTRODUCTION

The Fast Fourier Transform (FFT) is a standout amongst the most imperative and essential calculations in the advanced flag preparing range. Since the revelation of FFT, numerous variations of the FFT calculation have been created, for example, radix-2 and radix-4 FFT. The another variation of FFT calculation called split-radix FFT (SRFFT). Their calculation requires minimal number of duplications and augmentations among all the known FFT calculations. Since math operations essentially add to general framework control utilization, SRFFT is a decent possibility for the usage of a low-control FFT processor. When all is said in done, all the FFT processors can be sorted into two principles bunches: pipelined processors or shared-memory processors. Cases of pipelined FFT processors. A pipelined engineering gives high throughputs, however it requires more equipment assets in the meantime. One or numerous pipelines are frequently actualized, each comprising of butterfly units and control rationale. Interestingly, the mutual memory-based engineering requires minimal measure of equipment assets to the detriment of slower throughput. Cases of such processors. In the radix-2 shared-memory design, the FFT information is composed into two memory banks. At each clock cycle, two FFT information are given by memory banks and one butterfly unit is utilized to handle the information. At the following clock cycle, the figuring results are composed back to the memory banks and supplant the old information. The extent of this brief is constrained to the common memory design. In the common memory design, a productive tending to plot for FFT information and additionally coefficients (called twiddle components) is required. For the settled radix FFT, past works of this subject can be found in [5] and [6]. For split-radix FFT, it traditionally includes a L-formed butterfly datapath whose unpredictable shape has uneven latencies and makes booking troublesome. In this short,

we demonstrate that the SRFFT can be registered by utilizing an adjusted radix-2 butterfly structure. Our commitment comprises of mapping the split-radix FFT calculation to the mutual memory design, utilizing the lower multiplicative intricacy of the calculation to lessen the dynamic power and creating two novel twiddle figure tending to plans for the split-radix FFT.

2. RADIX-2 AND SPLIR-RADIX FFT ALGORITHMS

Fast Fourier Transform (FFT) has turned out to be practically universal and most imperative in fast flag handling. Utilizing this change, signs can be moved to the recurrence area where separating and relationship can be performed with less operation. It has been generally utilized as a part of correspondences and radar applications [11]. Late advances in chip innovation have expanded field programmable entryway cluster (FPGA) assets and as an outcome, the calculation of complex calculations, for example, the FFT, can be executed on a programmable device. The continuous prerequisite of FFT and the profoundly adaptable FPGA frame an awesome mix and enhances the speed of FFT handling to meet the fast of the cutting edge flag preparing. There are different FFT calculations, for example, radix-2, radix-4, radix22, split-radix, wino grad and numerous more. Radix-2 calculation is the least complex one, yet its estimation of expansion and increase is more than radix-4's. Though being more proficient than radix-2, radix-4 just can handle 4n-point FFT. The most alluring equipment arranged calculation will be that it has a similar number of non-minor increases at similar positions in the SFG as of radix-4 calculations, however has a similar butterfly structure as that of radix-2 calculations. In this paper, an equipment situated radix-22 calculation is contrasted and the traditional radix-2 calculation, which has the radix-4.

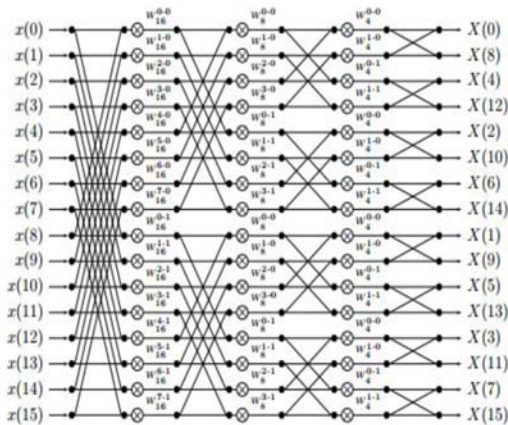


Figure-1. Signal flow graph.

3. MODIFIED RADIX FFT ALGORITHM

The existing FFT processor architecture consists of conventional adder. The proposed low power FFT processor is designed using split radix butterfly units. In the proposed method the carry select adder is used. Using the carry select adder the time taken to analyse the FFT process will reduce.

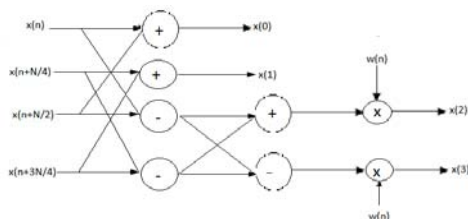


Figure-2. Modified butterfly diagram.

The proposed architecture of 16 bit CSLA using D latch consists of five clusters of bit word size starting from n bit RCA and D latch later on 3b, 4b, 5b, 6b word size simultaneously. Instead of using two separate adders in regular CSLA, in this method only one adder is used to reduce area, power consumption and delay and each of two addition is performed in one clock cycle. In this 16-b. adder the LSB is ripple carry adder which is 2 bit. The upper half of adder i.e. MSB which is 14-b wide works on the principal of clock. Whenever clock goes high the addition for carry input zero is performed and when clock goes low carry is assumed to be zero and addition is stored in adder itself. Carry out from the previous stage i.e., least significant bit adder is used as control signal for multiplexer to select final output carry and sum of the 16-bit adder. If the actual carry input is one, then computed sum and carry latch is accessed and for carry input zero MSB adder is accessed. Cout is the output carry [5].

Instead of using two separate adders in the regular CSLA, in this method only one adder is used to reduce the area, power consumption and delay. Each of the two additions is performed in one clock cycle. This is 16-bit adder in which least significant bit (LSB) adder is ripple carry adder, which is 2 bit wide. The upper half of

the adder i.e., most significant part is 14-bit wide which works according to the clock. Whenever clock goes high addition for carry input one is performed [10]. When clock goes low then carry input is assumed as zero and sum is stored in adder itself. From the Figure. it can understand that latch is used to store the sum and carry for $C_{in}=1$ and $c_{in}=0$. Carry out from the previous stage i.e, least significant bit adder is used as control signal for multiplexer to select final output carry and sum of the 16-bit adder. If the actual carry input is one, then computed sum and carry latch is accessed and for carry input zero MSB adder is accessed. Cout is the output carry.

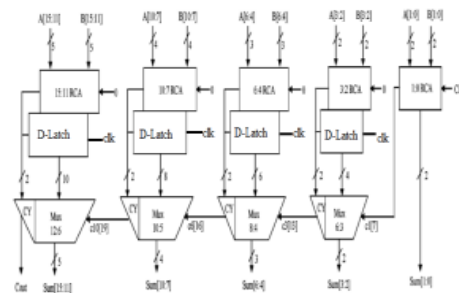


Figure-3. Carry select adder.

The group 2 performed the two bit additions which are a_2 with b_2 and a_3 with b_3 . This is done by two full adder (FA) named FA2 and FA3 respectively. The third input to the full adder FA2 is the clock instead of the carry and the third input to the full adder FA3 is the carry output from FA2. The group 2 structure has five D-Latches in which four are used for store the sum2 and sum3 from FA2 and FA3 respectively and the last one is used to store carry [11]. Multiplexer is used for selecting the actual sum and carry according to the carry is coming from the previous stage. The 6:3 multiplexer is the combination of 2:1 multiplexer. When the clock is low a_2 and b_2 are added with carry is equal to zero. Due to the low clock, the first D-Latch is not enabled. The second D-Latch store the sum with $c_{in}=0$ by using inverted clock enable. When the clock is high, the addition is performed with carry is equal to one. The other D-Latches enabled and store the sum and carry for carry is equal to one. According to the value of c_1 whether it is 0 or 1, the multiplexer selected the actual sum and carry.

4. MODIFIED SPLIT-RADIX FFT ALGORITHM

The split-radix calculation required less aggregate increase and include operations than any past energy of-two calculation. For a period numerous FFT specialists thought it to be ideal as far as aggregate multifaceted nature, yet considerably more productive varieties have all the more as of late been found by Johnson and Frigo. The split-radix calculation can be inferred via watchful examination of the radix-2 and radix-4 flow graphs as in Figure-1 underneath. While in many spots the radix-4 calculation has less nontrivial twiddle elements, in a few places the radix-2 really needs twiddle elements exhibit in



the radix-4 structure or those twiddle variables disentangle to duplication by $-i$, which really requires just increments. By blending radix-2 and radix-4 calculations fittingly, a calculation of lower unpredictability than either can be inferred.

The proposed configuration is contrasted and the two ordinary shared-memory designs. Our two proposed address era calculations of the twiddle components are comparable and consequently, we just execute calculation 1, which is inside the ROM address generator. The address era of RAM depends on [6] and datapath width is 32 b. The three FFTs are incorporated under the limitation of 100 MHz in Xilinx ISE 14.7 focusing for Spartan-6 XC6SLX4 gadget. The re-enactment results are appeared in Table III. Power is measured by Xilinx XPower analyzer utilizing the exchanging movement trade organize document recorded in an adequate long re-enactment time. For a 1024-point FFT, the proposed calculation could accomplish more than 20% lower control however nearly keeps up the static power utilization. In the given engineering, when the FFT estimate builds, a bigger RAM and ROM size is required, however the butterfly unit does not change. The constraint of the proposed configuration is the vast assets utilized as a part of the butterfly unit. This restriction could be expelled by utilizing diverse butterfly structures for augmentations and duplications. Table IV demonstrates the power utilization of every segment for a 1024-point FFT. The lessening of element power utilization is because of the way that multipliers and ROM banks are empowered just when vital. We have likewise orchestrated the outline utilizing OSU gsc145 nm library in Cadence RTL compiler. All the three FFTs can keep running over 200 MHz. The library does not have a memory licensed innovation, and the recollections are built utilizing essential cells and flip-flops. The execution consequences of a 1024-point FFT are appeared in Table V. Countless are utilized to execute the memory banks, which turn into the most power hungry segment in the plan. Contrasted and the radix-2 tending to plans in [5] and [6], our tending to technique requires extra 2S-1-piece memory. Be that as it may, the SRFFT calculation has the sporadic flag stream diagram and makes the control of such processors more troublesome than the settled radix ones. In spite of the fact that a product answer for the ordering issue has been given in [9], the ordering plan is intended for the L butterfly structure, which is not reasonable for the equipment execution because of its uneven latencies. Some past works, for example, [10] utilize query tables to take care of the ordering issue. Clearly the proposed calculation requires altogether less memory than the query table approach. Contrasted and a pipelined SRFFT design, for example, split-radix single-way postpone criticism (SRSDF) given in the common memory engineering offers altogether lessened equipment cost and power utilization to the detriment of slower throughput. For a N-point FFT, SRSDF requires $\log_4 N - 1$ multipliers and $4 \log_4 N$ adders.

5. RESULTS AND DISCUSSIONS

The proposed method is developed in Xilinx platform. These are designed in Verilog HDL, simulated in Xilinx ISim simulator and synthesized using Xilinx XST. These architecture are implemented and analysed in Xilinx Spartan3 FPGA. Figure-4 shows the output of counter, Figure-5 shows the output of ROM, Figures 6, 7, 8 shows the output of SRFFT.

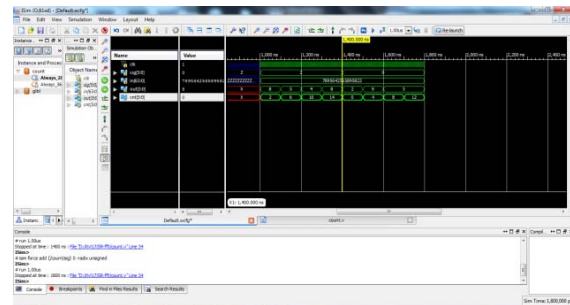


Figure-4. Simulation output of counter.

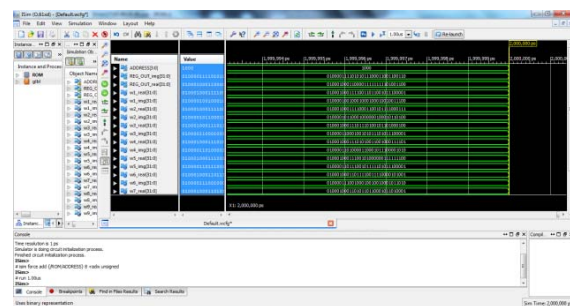


Figure-5. Simulation output of ROM.

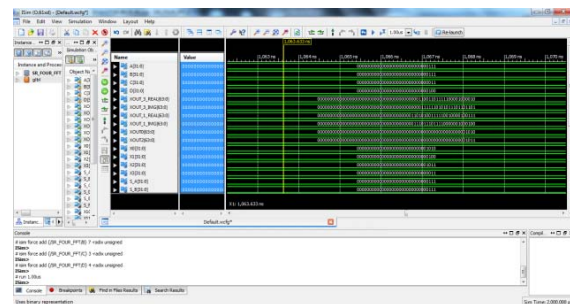


Figure-6. Simulation output of split radix FFT



Figure-7. Top level block of split radix FFT.

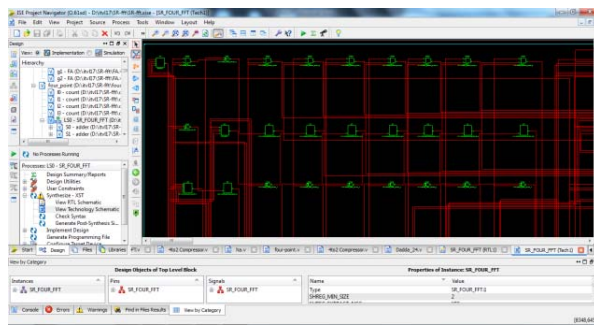


Figure-8. Top level block in FPGA.

6. CONCLUSIONS

The proposed SRFFT strategy lessens the dynamic power utilization to the detriment of more equipment assets. We additionally show two tending to plans for both the insignificant and nontrivial twiddle elements. Since SRFFT has the base number of augmentations contrasted and different sorts of FFT, the outcomes could be more ideal in the feeling of skimming point operations.

REFERENCES

- [1] Harpreet Singh Dhillon and Abhijit Mitra. 2008. A Reduced-bit Multiplication Algorithm for Digital Arithmetic. *International Journal of Computational and Mathematical Sciences*, February. pp. 64-69.
- [2] Sumit Vaidya and Depak Dandekar. 2010. Delay-power performance comparison of multipliers in VLSI circuit design. *International Journal of Computer Networks & Communications (IJCNC)*. 2(4).
- [3] Pramod Kumar Mehe, Basant Kumar Mohanty, Sujit Kumar Patel, Soumya Ganguly and Thambipillai Srikanthan. 2015. Efficient VLSI Architecture for Decimation-in-Time Fast Fourier Transform of Real-Valued Data. *IEEE Transactions on Circuits and Systems-I: Regular Papers*. 62(12).
- [4] M. Ayinala, Y. Lao, and K. K. Parhi. 2013. An in-place FFT architecture for real-valued signals. *IEEE Trans. Circuits Syst. II, Exp. Briefs*. 60(10): 652-656.
- [5] M. Micheal Priyanka and T. Ravi. 2015. Survey on role of Memristor in electronics. *Proceedings of International Conference on Control Instrumentation Communication and Computational Technologies, ICCICCT*. pp. 738-744.
- [6] J. Chen, J. Hu, S. Lee and G. E. Sobelman. 2015. Hardware efficient mixedradix-25/16/9 FFT for LTE systems. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* 23(2): 221-229.
- [7] L. G. Johnson. 1992. Conflict free memory addressing for dedicated FFT hardware. *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.* 39(5): 312-316.
- [8] D. Cohen. 1976. Simplified control of FFT hardware. *IEEE Trans. Acoust., Speech, Signal Process.* 24(6): 577-579.
- [9] X. Xiao, E. Oruklu and J. Saniie. 2008. An efficient FFT engine with reduced addressing logic. *IEEE Trans. Circuits Syst. II, Exp. Briefs*. 55(11): 1149-1153.
- [10] M. Manoranjani and T. Ravi. 2015. Multithreshold CMOS sleep stack and logic stack technique for digital circuit design. *ARPN Journal of Engineering and Applied Sciences*. 10(10): 4550-4556.
- [11] Ravi. T. 2016. 10-Nanometer carbon nano tube field effect transistor based high celerity transposed polyphase decimation filter. *Materials Today: Proceedings*. 3(6): 1799-1807.
- [12] H. V. Sorensen, M. T. Heideman, and C. S. Burrus. 1986. On computing the split-radix FFT. *IEEE Trans. Acoust., Speech Signal Process.* 34(1): 152-156.
- [13] J. Kwong and M. Goel. 2012. A high performance split-radix FFT with constant geometry architecture. in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, Dresden, Germany. pp. 1537-1542.
- [14] W.-C. Yeh and C.-W. Jen. 2003. High-speed and low-power split-radix FFT. *IEEE Trans. Signal Process.* 51(3): 864-874.