ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse144942020.pdf
https://doi.org/10.30534/ijatcse/2020/144942020

Analysis and implementation of SDF Radix-2 FFT processor using VERILOG Hardware Description Language

Phuong H. Lai[1], Manh Hoang[2], Viet Q. Tran[2], Tung V. Nguyen[2], Thien V. Truong[2], and Phong H. Nguyen[3]

[1]Dept. of Computing Fundamentals, FPT University, Hanoi, Vietnam, phuonglh17@fe.edu.vn
[2]ICT Department, FPT University, Hanoi, Vietnam, manhhhe130294@fpt.edu.vn,viettqse06178@fpt.edu.vn, tungnvhe130151@fpt.edu.vn, thientvse04522@fpt.edu.vn
[3]Center of Machine Vision & Signal Analysis, University of Oulu, Finland; phong.nguyen@oulu.fi

ABSTRACT

This paper will study a novel system on chip (SoC) design for fast Fourier transform (FFT) module. We first explain the role and position of FFT module in a digital intelligent system. Then, the discrete Fourier transform (DFT) and decimation in frequency (DIF) Radix-2 butterfly FFT algorithm is explained in detail, mathematically. In addition, the analysis of a simple pipeline FFT processor and a single-path delay feedback pipeline FFT processor based on SDF Radix-2 algorithm are discussed.Finally,the implementation and verification of proposed FFT processor are performed VERILOG hardware description language (HDL).

Key words: Fourier transform, pipeline processor, VERILOG HDL, FPGA, System on Chip design.

1. INTRODUCTION

Nowadays, digital intelligent systems can be found anywhere around us it helps our living more comfortable with many types of useful application. One of technologies inside intelligent system is digital signal processing (DSP) and it is no doubt to say that our daily lives are powered by signal processing technologies [1]. In DSP, the Fourier transform (FFT) is one of most fundamental and important building blocks. Fourier transform has a long history, in 1822 Joseph Fourier showed that some functions could be written as an infinite sum of harmonics. Similarly, some researches showed that any periodic signal can be approximated by a sum of many sinusoids at harmonic frequencies of signal with appropriate amplitude and phase. Hence, Fourier transform is used to convert signal from time domain to frequency domain and vice versa.

Fast Fourier transform(FFT) simply is an algorithm for efficiently calculation of discrete Fourier transform. It has a long history improvement from Fourier transform. Basically, the most general and modern version FFT can be noted in 1965 by James Cooley and John Tukey. There are many applications of FFT we can note such as in application specific integrated circuit (ASIC) [2], digital signal processing (DSP) [3], general of quantum gate [4], etc. FFT performs efficiently but there are many parameters should be considered and there is the trade-off to choose those parameters depending on application and its requirements [5].Generally, FFT algorithm can be classify into decimation in time (DIT) or decimation in frequency (DIF) [6]. In addition, for implementation ofFFT processor, we can choose Radix-*k* module where *k* is the size of arithmetic unit [7]. Consequently, single-path or multi-path structure are available [8].

The purpose of our paper is to design an FFT processor which can be flexible and useful to be as intellectual logic core for many systems on chip.The decimation in frequency Radix-2 FFT algorithm is chosen to be explained. And we will use pipeline FFT processor and single path delay feedback pipeline processor for our design.The research is conducted by VERILOG codes running on **Model Simulation 10.4a** tool student version. The data and constant values were calculated by **MATLAB R2020a**.The organization of this research is as follows.In Section 2, we explain the role and position of FFT block on digital system. The analysis and design of pipeline FFT processor is explained in Section 3. In Section 4, the implementation and verification of FFT processor design are discussed. Finally, conclusion is givenin Section 5.

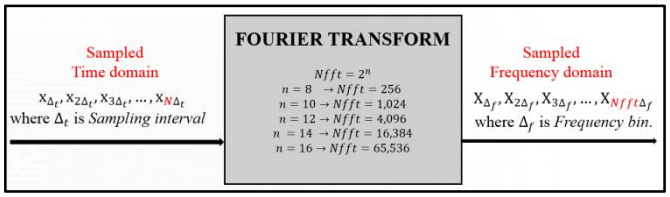

Figure 1: Overall model of **Fourier** transform.

2. ROLE OF FFT MODULE ON DIGITAL SYSTEMS

A. Overview of Fourier transform

In digital signal processing, Fourier transform is a linear transformation of the vector in time domain **{x}** to the vector in frequency domain **{X}** where equation is given as:

$$\mathbf{X} = \boldsymbol{F}_N \mathbf{x}$$

$$\leftrightarrow \mathbf{X}[k] = W_N^{k\times 0} \times \mathbf{x}[0] + W_N^{k\times 1} \times \mathbf{x}[1] + \cdots + W_N^{k\times(N-1)} \times \mathbf{x}[N-1],$$

where

$$\mathbf{F}_N = \begin{bmatrix} W_N^{0\times 0} & W_N^{0\times 1} & W_N^{0\times 2} & \cdots & W_N^{0\times(N-1)} \\ W_N^{1\times 0} & W_N^{1\times 1} & W_N^{1\times 2} & \cdots & W_N^{1\times(N-1)} \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W_N^{(N-1)\times 0} & W_N^{(N-1)\times 1} & W_N^{(N-1)\times 2} & \cdots & W_N^{(N-1)\times(N-1)} \end{bmatrix}_{N\times N} \quad (1)$$

and W_N is the roots of unity, W_N^{kt} is also called as the twiddle factors which is calculated as:

$$W_N = e^{-j\frac{2\pi}{N}} = cos\left(\frac{2\pi}{N}\right) - jsin\left(\frac{2\pi}{N}\right),$$
$$W_N^{kt} = e^{-j\frac{2\pi}{N}kt} = cos\left(\frac{2\pi}{N}kt\right) - jsin\left(\frac{2\pi}{N}kt\right).$$

The above matrix form transformation is called *Discrete Fourier transform*(DFT).For efficiently calculation the DFT, fast Fourier transform (FFT) is a novel algorithm which reduced the complexity calculation from $O(N^2)$ to $O(N \times logN)$.The overall model of Fourier transform is given as Figure 1 and the efficiency comparison between DFT and FFT is given as Figure 2.

- The FFT is Simply an Algorithm for Efficiently Calculating the DFT
- Computational Efficiency of an N-Point FFT:
  - DFT: N^2 Complex Multiplications
  - FFT: $(N/2)\ \log_2(N)$ Complex Multiplications

N	DFT Multiplications	FFT Multiplications	FFT Efficiency
256	65,536	1,024	64 : 1
512	262,144	2,304	114 : 1
1,024	1,048,576	5,120	205 : 1
2,048	4,194,304	11,264	372 : 1
4,096	16,777,216	24,576	683 : 1

Figure 2: Comparison between FFT and DFT.

B. FFT module on digital system

FFT are widely used for applications in engineering, science, music, science, and mathematical, it is included in top 10 algorithms of 20th century by IEEE magazine computing in science and engineering. Basically, the input signal such as image, speech, pattern, temperature, sound, are sensing by some sensor system. Then, those signals will pass through some amplifier to stable in some amplitudes and they will be sampled by Analog-Digital converter (ADC) with appropriate parameters. The output of ADC is a discrete time signal which can be performed by digital filter before processing by FFT module. The outputs of FFT module is the signal in spectrum domain, which will be stored and processing in Microcontroller unit (MCU). The sully process can be viewed as Figure 3.

C. DIF Radix-2 Butterfly FFT algorithm

FFT has a long history with many versions, a modern generalization FFT algorithm was invented by J. Cooley and J. Tukey in 1965 that is applicable when length *N* is power of 2. Basically, the idea of FFT is to break down a DFT block into many smaller DFTs along with twiddle factors (complex roots of unity). When *N* is power of 2, the smallest DFT block is Radix-2 butterfly which is easy implemented in Hardware. Figure 4 shows a Radix-2 butterfly unit where W is a complex number, x and y are real input, X and Y are complex output. Since we have the complex numbers on Butterfly unit, an adaptive Hardware description should be considered, it will be mentioned in next section.

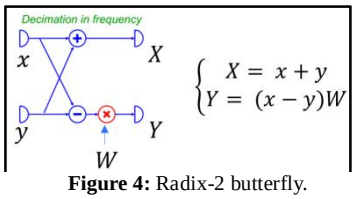

Figure 4: Radix-2 butterfly.

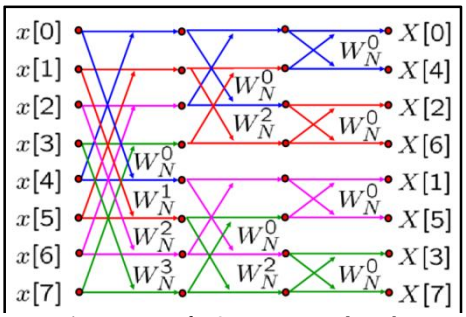

Figure 5: DIF for 8-points **FFT** algorithm.

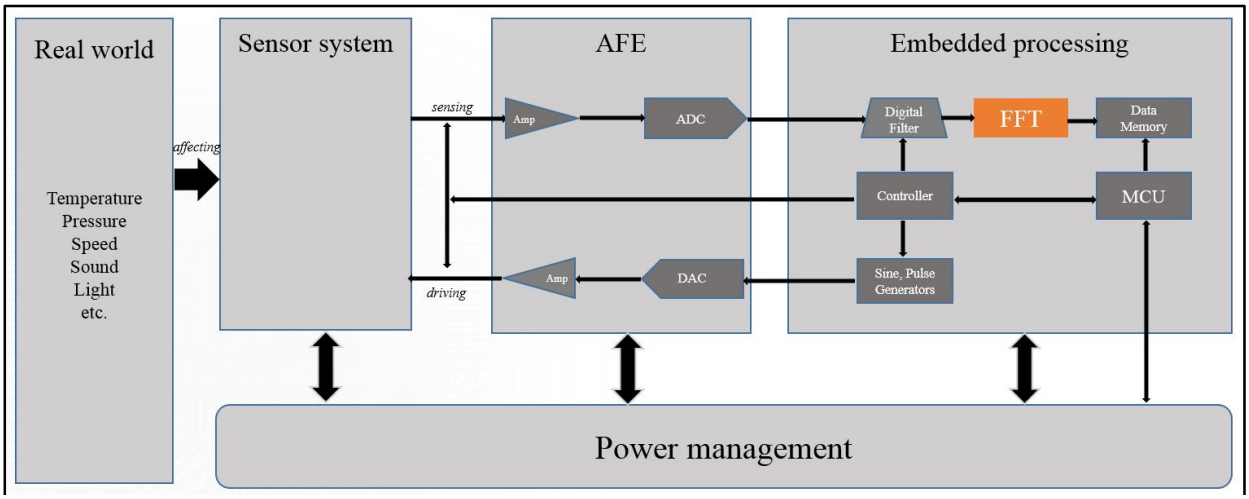

Figure 3 : Position of **FFT** module in an overall digital system.

There are two ways to take the input for Radix-2 butterfly, hence we can have decimation in time (DIT) or decimation in frequency (DIF) FFT algorithm. Through this paper, DIF is using such that in Figure 5.Consequently, the bit-reversal permutation is required at output of FFT module. The equation for DIF is as following:

- X[2m]

$$= \sum_{t=0}^{\frac{N}{2}-1} x[t] \times W_N^{2mt} + \sum_{t=0}^{\frac{N}{2}-1} x\left[t+\frac{N}{2}\right] \times W_N^{2m\left(t+\frac{N}{2}\right)}$$

$$= \sum_{t=0}^{\frac{N}{2}-1} \left(x[t] + x\left[t+\frac{N}{2}\right]\right) \times W_N^{2mt}$$

$$= \sum_{t=0}^{\frac{N}{2}-1} \left(x[t] + x\left[t+\frac{N}{2}\right]\right) \times W_{\frac{N}{2}}^{mt}.$$

- X[2m + 1]

$$= \sum_{t=0}^{\frac{N}{2}-1} x[t] \times W_N^{(2m+1)t} + \sum_{t=0}^{\frac{N}{2}-1} x\left[t+\frac{N}{2}\right] \times W_N^{(2m+1)\left(t+\frac{N}{2}\right)}$$

$$= \sum_{t=0}^{\frac{N}{2}-1} x[t] \times W_N^{2mt} \times W_N^{t} + \sum_{t=0}^{\frac{N}{2}-1} x\left[t+\frac{N}{2}\right] \times W_N^{2mt} \times W_N^{t} \times W_N^{(2m+1)\frac{N}{2}}$$

$$= \sum_{t=0}^{\frac{N}{2}-1} x[t] \times W_N^{2mt} \times W_N^{t} + \sum_{t=0}^{\frac{N}{2}-1} x\left[t+\frac{N}{2}\right] \times W_N^{2mt} \times W_N^{t} \times (-1)$$

$$= \sum_{t=0}^{\frac{N}{2}-1} \left(x[t] - x\left[t+\frac{N}{2}\right]\right) \times W_N^{t} \times W_{\frac{N}{2}}^{mt}.$$

3. ANALYSIS OF PIPELINE FFT PROCCESSOR

A. Analysis of simple pipeline FFT processor

Based on DIF radix-2 butterfly FFT algorithm, the suitable and popular way to implement in hardware is pipelined architecture to achieve higher speed. The block diagrams of a simple pipeline FFT processor is given in Figure 6. For 8-points FFT, we need three stages. In stage 1,there are two delay modules, one Radix-2 module, and one multiplication with twiddle value.The delay-4 module is used to set 4 inputs signal come to the upper input of Radix-2 module and remaining 4 inputs in lower input of Radix-2. The upper output of Radix-2 will directly to next stage, in contrast the lower outputs will come to another delay-4 module and take the multiplication with corresponding twiddle values which was stored in ROM. Similarity in stage 2 and stage 3, the difference is that delay-2 module is used in stage 2 and delay-1 module is used in final stage. Finally, the output will be the values signal in frequency domain but in permutation order as discussion in Section 2.

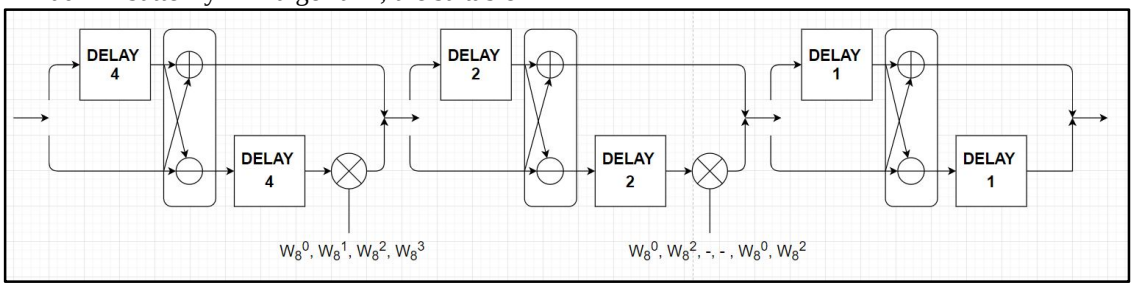

Figure 6: System model for simple 8-points **FFT** processor.

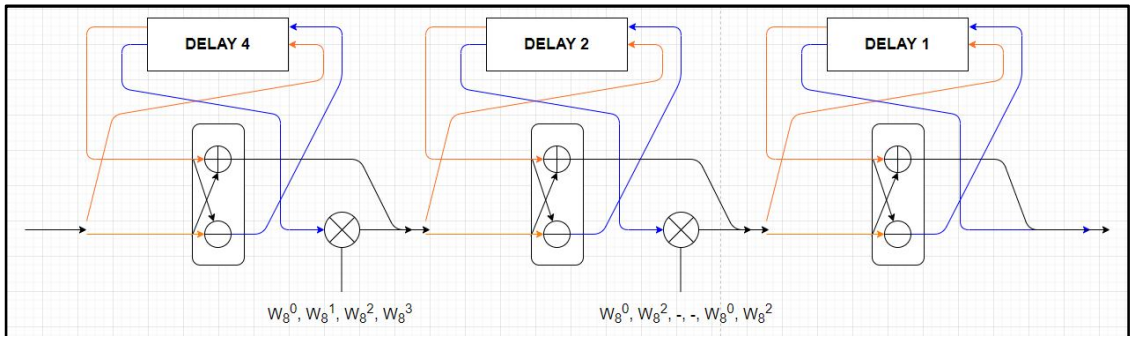

Figure 7: System model for single-path delay feedback 8-points **FFT** processor.

B. Analysis of SDF pipeline FFT processor

Since the pipeline FFT processor obtains a higher speed but the cost on resources, several methods to save resource are proposed. In this proposed system, instead of using two delay modules for each stage, only one delay module is used by the help of delay feedback as given in Figure 7.

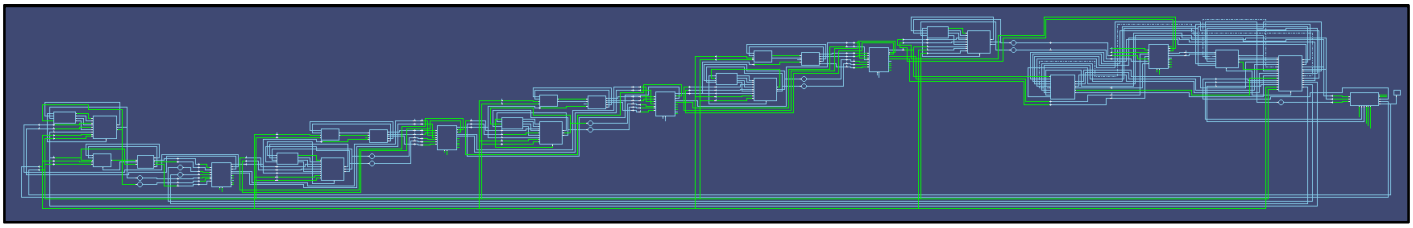

Figure 8: Schematic tracer of 32-points **FFT** pipeline processor.

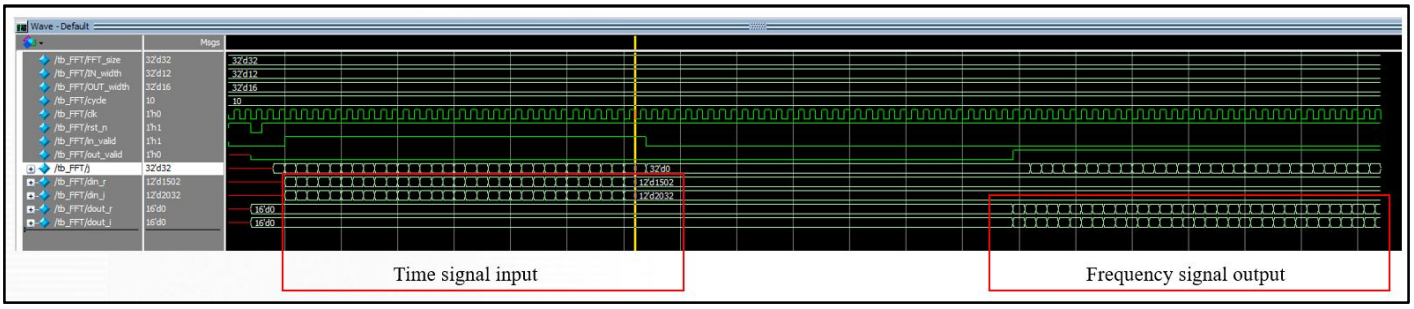

Figure 15: Timing diagram for a testbench of 32-points **FFT** pipeline processor.

4. IMPLEMENTATION OF FFT PROCCESSOR USING VERILOG HDL

A. Implementation of basic modules

There are numerous modules to implement to build up the full FFT processor. Overall, the proposed FFT processor design is given as Figure 8 and the list of all sub-instance can be found asFigure10. The schematic tracer of all sub-modules is clearly given as Figure 11 for Radix-2 module, Figure 12 for ROM module (twiddle factor), Figure 13 for shift-register module (delay).As analysis in previous section, the Radix-2 modules are just an arithmetic operation with some modification for complex number. The constant number of twiddles are calculated by MATLAB tool and are stored in Read-only-memory (ROM). The delay modules are just the shift-registers with suitable parameter.

One of their common problems can be shown here is the representation of real number and their arithmetic operators art in Hardware description language. The fixed point and second complementation are used to represented real number in HDL where 8-length fraction bits. The complex number is separated into two parts, one is for real part and another is for image part. As a result, the real number multiplication can be implemented as Figure 9.

B. Verification of proposed FFT processor

A test bench for verification of proposed FFT processor is made. Therein, the input is stored in txt file where input txt file consisted of 32 value. The 32-points input data and expected 32-points output data are calculated by MATLAB 2020 version software.

At each system clock, the value at each line is called and come into our proposed system. As given in Figure 15, we need 32-clocks for all inputs come into the system and 2*32=64 clocks for all the FFT values appear at output. The output of proposed system can be found at two columns at the middle of Figure 14. The comparison shows that there is a non-considerable error with expected result which come from the fractional part is set to 8-bits length as Figure 9.

```
module FloatMultiplier(
     input wire signed [23:0] a, b,
     output wire signed [23:0] product
);
    wire signed [31:0] temp;
    assign temp    = a*b;
    assign product = temp[31:8];

endmodule
```

Figure 9: Real number multiplication for proposed **FFT** processor.

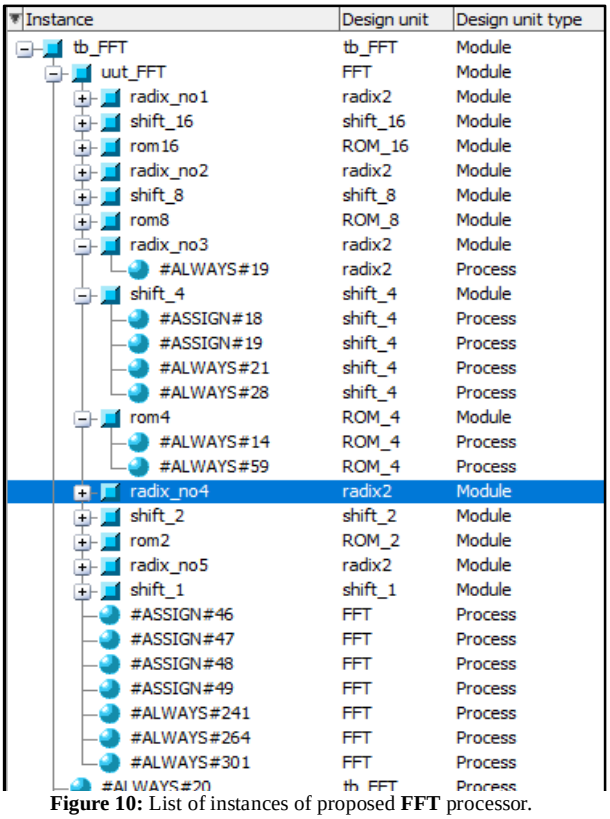

Figure 10: List of instances of proposed **FFT** processor.

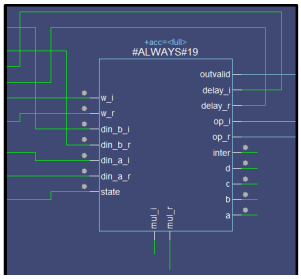

Figure 11: Radix-2 modules of proposed FFT processor.

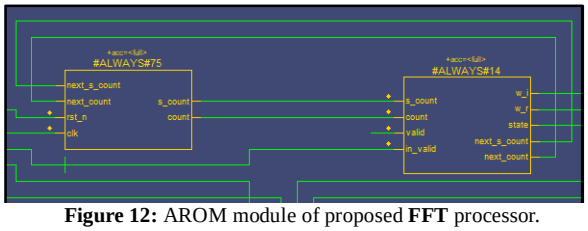

Figure 12: AROM module of proposed **FFT** processor.

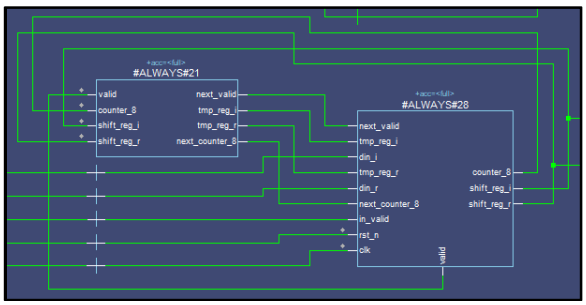

Figure 13: Adelay module of proposed FFT processor.

Input		Output by proposed system		Output by software	
Real	Image	Real	Image	Real	Image
0	-1884	-1365	-4063	-1365	-4063
-1700	1674	-5015	989	-5006	978
-1605	262	6212	9417	6210	9398
412	1052	3684	66	3677	66
1363	-6	-7306	3855	-7306	3855
-115	1818	-3352	3926	-3351	3915
-209	-36	-7300	1455	-7292	1432
277	-1932	1735	2359	1732	2356
381	-203	12794	-696	12794	-696
736	-1591	1814	-5628	1805	-5620
-1086	-1760	9439	-470	9432	-473
42	47	-6285	3002	-6277	2992
1346	-1397	-5415	6621	-5414	6622
1889	-145	-1651	-9304	-1654	-9291
-978	-24	-2290	-2366	-2289	-2363
96	-743	-816	-14387	-812	-14366
-227	116	5629	-9067	5629	-9067
-1409	1596	-1502	998	-1496	994
1979	-98	1187	-11132	1189	-11113
-477	-1501	-5779	431	-5769	434
299	681	-6239	-550	-6239	-550
-751	1002	1190	1628	1193	1611
1563	-1492	1851	-20561	1843	-20540
-968	-1465	7572	-1040	7566	-1041
-1552	121	-2446	-4102	-2446	-4102
-723	86	8801	-16525	8795	-16518
-484	393	2864	2347	2872	2351
-1781	-49	5110	-1413	5098	-1406
2043	-1910	-6838	-6800	-6839	-6801
-527	621	-1065	-7552	-1067	-7538
-701	672	-4414	7819	-4415	7817

Figure 14: Outputs comparison between proposed system and software of proposed FFT processor.

5. CONCLUSION

The paper has discussed a system on chip design of pipeline FFT processor. Through the paper, analysis of DIF Radix-2 FFT algorithm is given mathematically and clearly. In addition, simple pipeline FFT processor and SDF pipeline FFT processor are explained. A test bench is given to verify our proposed system.

In SoC, FFT processor is popularly used in many digital signal processing systems. This SoC design can be viewed as an intellectual logic core for flexible and suitable integration in DSP system for many types of digital intelligent system. Due to the VERILOG source codes, MATLAB based calculation of input data in our implementation have numerous lines of codes, we cannot show it clearly in this limited paper. Any requirement is welcomed, and author will share them with appropriated reasons.

ACKNOWLEDGEMENT

This work is supported byFPT University, Hanoi, Vietnam. We would like to thank Dr. Nguyen Manh Duc (G2Touch Company, Republic of Korea) for his useful suggestions, and Dr. Nguyen Ha Phong for his advices.

REFERENCES

1. Nguyen, D.M., Kim, S. "**The fog on Generalized teleportation by means of discrete-time quantum walks on N-lines and N-cycles**", Modern Physics Letters B 33 (23), 1950270, 2019. https://doi.org/10.1142/S0217984919502701
2. Antipas., et. al. "**Performance analysis of MUSIC algorithm for microphone array using FPGA board**". Int. J. of Advanced Trendsin Computer Science and Engineering. vol. 9(3), 2020. https://doi.org/10.30534/ijatcse/2020/152932020
3. B. M. Krishna., et. al. "**FPGA Implementation of Image Cryptology using Reversible Logic Gates**". Int. J. of Advanced Trendsin Computer Science and Engineering. vol. 9(3), 2020. https://doi.org/10.30534/ijatcse/2020/06932020
4. Nguyen, D.M., Kim, S. "**Quantum Key Distribution Protocol Based on Modified Generalization of Deutsch-Jozsa Algorithm in d-level Quantum System**", Int. J. Theor. Phys. vol. 58(1), pp. 71-82, 2019.
5. Nguyen, D.M., Kim, S. "**A novel construction for quantum stabilizer codes based on binary formalism**". Int. J. of Modern Phys. B. vol. 33(8), 2020.
6. Nguyen, N.H., et. al. "**A Pipelined FFT Processor Using an Optimal Hybrid Rotation Scheme for Complex Multiplication: Design, FPGA Implementation and Analysis**".Electronics, Vol. 7(137), 2018.
7. Sudhanshu., et. al. "**FPGA based design and simulation of 32-point FFT through radix-2 DIT algorithm**". Int. J. of Engineering Research $ Technology (IJERT), vol. 5(2), 2016. https://doi.org/10.17577/IJERTV5IS020484
8. K. Malathy., et. al. "**A novel VLSI based radix-2 single path delay commutator (R2SDC) FFT architecture design**". Indian journal of Science and Technology, vol. 9(11), 2016.