

VLSI Architecture for Area-Delay-Energy efficiency for ascendable In-Place Computation of 16 POINT FFT on Real Data

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in
Electronics and Communication Engineering

by

P. Vijay (37130318)
N. Harsha Vardhan (37130278)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING**

SCHOOL OF ELECTRICAL AND ELECTRONICS

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI- 600 119**

APRIL - 2021



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by
NAAC Jeppiaar Nagar, Rajiv
Gandhi Salai, Chennai – 600119.

www.sathyabama.ac.in



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of
P.Vijay (Reg.No: 37130318) and **N.Harsha Vardhan** (Reg.No: 37130278)
who carried out the project entitled VLSI Architecture for Area-Delay-Energy
efficiency for ascendable In-Place Computation of 16 POINT FFT on Real Data
under my supervision from November 2020 to March 2021.

Internal Guide
Mrs. K. Srilatha ,M.E, (Ph.D).,

Head of the Department
Dr. T. RAVI, M.E.,
Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

We **P.Vijay** (Register No: 37130318) and N.Harsha Vardhan (Register No: 37130278) hereby declare that the Project Report entitled **VLSI Architecture for Area-Delay-Energy efficiency for ascendable In-Place Computation of 16 POINT FFT on Real Data** done by us under the guidance Mrs. K. Srilatha ,M.E, (Ph.D). is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in ELECTRONICS AND COMMUNICATION ENGINEERING.

DATE :

PLACE: CHENNAI

P.Vijay (37130318)

N.Harsha vardhan(37130278)

SIGNATURE OF CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. NANDHITHA, M.E., Ph.D.**, Dean School of Electrical and Electronics and **Dr. T. RAVI, M.E., Ph.D.**, Head of the Department of Electronics and Communication Engineering for providing me necessary support and details at the right time during the progressive reviews.

We would like to express my sincere and deep sense of gratitude to our Project Guide **Mrs. K. Srilatha ,M.E, (Ph.D).** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT:

Proposed a design approach to develop efficient architecture for in-place RFFT which could be scaled for higher throughput and larger FFT sizes. The energy per sample is less while comparing to other systems. Efficient computation of real-valued fast Fourier transform (RFFT) has received significant attention in recent years thanks to its several applications in conventional digital signal processing and other emerging areas. In- place RFFT architectures are gaining popularity thanks to their lower hardware complexity compared with pipeline architectures. The scaling of in-place RFFT architecture for higher lengths and better throughput may be a challenging issue thanks to increasing access conflict and better memory bandwidth requirement. A design is presented for the development of an area-delay and energy-efficient architecture for in-place RFFT. An in-place fast Fourier transform (FFT) structure consists of a butterfly block which performs a group of butterfly operations in every clock cycle. From the analysis we found that the in-place FFT structures with larger butterfly blocks are more efficient in terms of area-time complexity and energy consumption.

CONTENT		
CHAPTER	TITLE	PAGE No
	ABSTRACT	v
	LIST OF ABBREVIATIONS	vii
	LIST OF FIGURES	viii
	LIST OF TABLES	viii
1	INTRODUCTION	9
2	LITERATURE SURVEY	13
3	EXISTING SYSTEM	17
4	PROPOSED SYSTEM	28
	4.1 Storage unit design of proposed RFFT structure	20
	4.2 Internal structure of multi-channel register bank	21
	4.3 Data-Flow Analysis of Storage Unit	22
	4.4 Storage Unit Design	23
	4.5 Multiplexer	
5	RESULTS AND DISCUSSIONS	28
6	SUMMARY AND CONCLUSION	37
	REFERENCES	40

LIST OF ABBREVIATIONS

VLSI	VERY LARGE SCALE INTEGRATION
IC	INTEGRATED CIRCUIT
RTL	REGISTER TRANSFER LEVEL
HDL	HARDWARE DISCRIPTION LANGUAGE
EDA	ELECTRONIC DESIGN AUTOMATION
FFT	FAST FOURIER TRANSFORM
DFT	DIRECT FOURIER TRANSFORM
RFFT	REAL FAST FOURIER TRANSFORM
ASIC	APPLICATION-SPECIFIC INTEGRATED CIRCUIT
DSP	DIGITAL SIGNAL PROCESSING
OFDM	ORTHOGONAL FREQUENCY DIVISION
MULTIPLEXING	
SU	STORAGE UNIT
BCU	BRIDGE CONTROL UNIT
MCRB	MULTI-CHANNEL REGISTER-BANKS
MOSFET	METAL-OXIDE-SEMICONDUCTOR FIELD EFFECT
TRANSISTOR	
ISE	INTEGRATED SYNTHESIS ENVIRONMENT
SDK	SOFTWARE DEVELOPMENT KIT
UWB	ULTRA WIDE BAND

LIST OF FIGURES

FIGURE No.	DESCRIPTION	PAGE No.
1.1	Vlsi Design Flow	11
1.2	Hierarchy-Structural	12
4.1	Storage unit design of proposed RFFT	
4.2	Internal structure of multi-channel register	22
4.3	Mux 4:1	25
4.4	Multiplexer	27

LIST OF TABLES

TABLE	DESCRIPTION	PAGE NUMBER
4.1	Boolean Table	26
6.1	Comparison	38

CHAPTER 1

INTRODUCTION

FFT is vastly applied in several applications such as spectral estimation, speech and audio processing, image processing, digital communication, wireless sensor network (WSN), radar signal processing, bio- medical. In these applications.

Fast Fourier Transform are of two categories:

pipeline architecture (i.e., single-path delay feedback (SDF) based structures and multi-path delay commutator (MDC) based structures. Processor-memory (PM)-based folded in-place structures.

Generally, all the above mentioned structures maintain of the three main units that are:

- (i) butterfly computation unit (BCU),
- (ii) storage unit (SU), and
- (iii) twiddle factor generation unit (TFU).

The MDC based structures are more tender for pipeline calculation of

FFT thanks to the systematic data- flow and better utilization efficiency of Bus Control Unit(BCU) at the value of limited extent high memory obligation than the SDF-based structures. In PM-based structures, the intra and inter butterfly stages of calculations are saved in BCU ,taken use of In Place Fast Fourier Transform for less-complexity realization. Pipeline and PM-based in-place architectures are of two main variants of FFT structures with disparity of features and selection of the one from these two variants of architectures mostly depends on the space-time restrictions of target application. Several pipeline and PM-based architectures are proposed for efficient implementation of FFT for various applications.

VLSI:

Very-large-scale integration (VLSI) is an integrated chip (IC) with a thousands of transistors on chip. VLSI started within 1970s the time when difficult semiconductor and communication Technology was being developed. The microprocessor is a VLSI device. Prior the initiation of VLSI technology, most of the ICs had slight set of functions they might perform. Electronic circuit may contain a rom, ram, cpu and other glue logic. Very Large Scale Integration(VLSI) lets IC designers append all of those into one chip.

The industry has achieved an outstanding growth over the previous couple of decades, mainly thanks to the rapid advances in large scale integration technologies and system design applications.

The present cutting-edge technologies such as high resolution and low bit-rate video and cellular communications give the end-users a excellent amount of applications, processing power as well as portability. This tendency is expected to grow swiftly, with essential implications on VLSI design and systems design.

VLSI Design Flow

The Very Large Scale Integration(VLSI) IC circuit design flow is given in the figure below. The different stages of design are numerated and therefore the blocks present processes within the design flow.

Specifications takes place first, they relate abstractly, the functionality, interface, and therefore the architecture of the digital IC circuit to be designed.

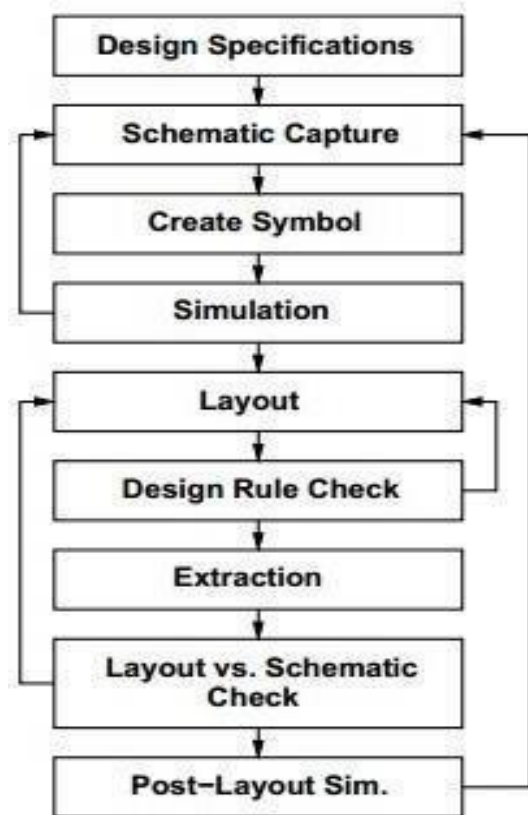


Fig 1.1 Vlsi Design Flow

Behavioral description is created to research the planning with regard to functionality, performance, compliance to given standards, and other statements.

RTL description is achieved using HDLs. This RTL description is simulated to test functionality. From here onwards we'd like the assistance of EDA tools.

RTL description is then converted to gate-level netlist with the use of logic synthesis tools. A gate level netlist may be a description of the circuit in terms of gates and connections between them, which are made in such how that they meet the timing, power and area specifications.

Finally, a physical layout is formed , which can be verified then sent to fabrication.

Y Chart

The Gajski-Kuhn Y-chart mostly a model, that captures considerations in designing semiconductor devices. Each of the domains are often divided into levels of abstraction, using concentric rings.

At the highest level (outer ring), we consider the architecture of the chip; at the lower levels (inner rings), we successively refine the planning into finer detailed implementation – Generating a structural description through a behavioral one is done through the procedure of high-level synthesis or logical synthesis.

Generating a physical description from a structural one is done through layout synthesis.

Design Hierarchy-Structural

The design hierarchy is based on the principle of "Divide and Conquer." it's done as dividing the particular task into smaller tasks until it gets to its simplest level. This process is best suited due to the final evolution of the design has become so simple that its manufacturing becomes easier.

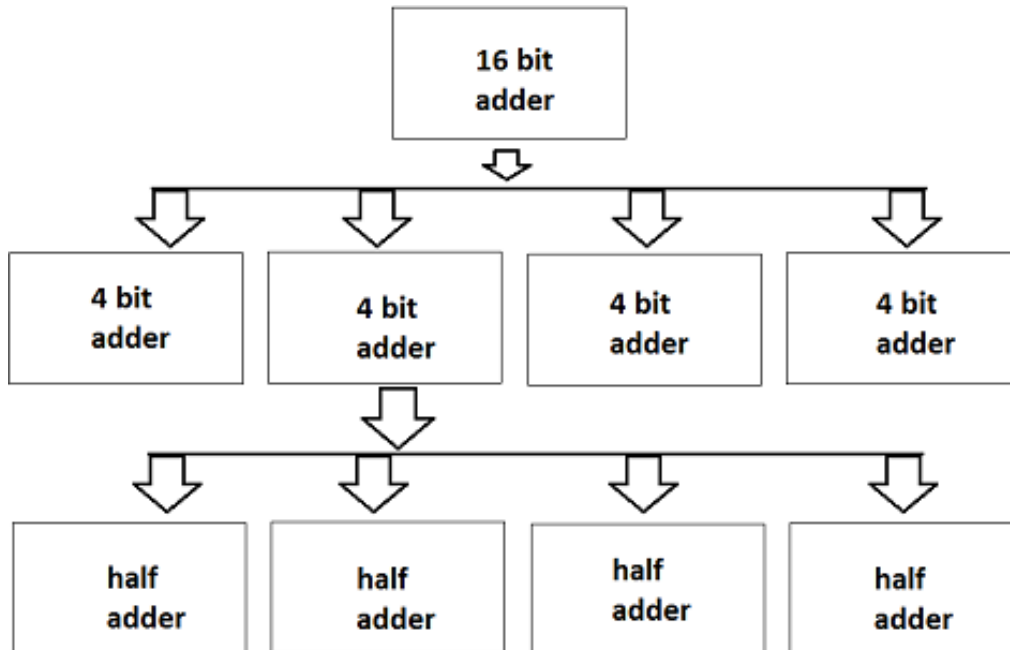


Fig 1.2 Hierarchy-Structural

Here, the entire chip of 16 bit adder is split into four modules of 4-bit adders. Dividing the 4-bit adder as the 1-bit adder or half adder. Adder addition is the easiest designing process and its internal circuit is also easy to fabricate on the chip. Now, connecting all the last four adders, we will design a 4-bit adder and moving on, we will design a 16-bit adder.

CHAPTER 2

LITERATURE SURVEY:

An Area Efficient 1024-Point Low Power Radix-2² FFT Processor With Feed-Forward Multiple Delay Commutators

The given paper gives a novel radix-2² multiple delay commutator architecture using the perks of the radix-2² algorithm, like as plain butterflies and low memory obligation. Therefore, it is more hardware efficient when implementing parallelism for higher throughput using multiple delay commutators or feed-forward data paths. Here, we propose an improved input scheduling algorithm based upon memory to eliminate energy required to shift data along the delay lines. A 1024-point FFT processor with two parallel data paths is implemented in 65-nm CMOS process technology. The FFT processor occupies an area of 3.6 mm², successfully operates in the supply voltage range from 0.4-1 V and the maximum clock frequency of 600 MHz. For low voltage, high performance applications, the processor is able to operate at 400 MHz and consumes 60.3 mW or 77.2 nJ/FFT generating 800 M samples/s at 0.6 V supply.

Area-Delay-Energy Efficient VLSI Architecture for Scalable In-Place Computation of FFT on Real Data

Efficient computation of real-valued fast Fourier transform (RFFT) has received significant attention in recent years due to its several applications in conventional digital signal processing and other emerging areas. In-place RFFT architectures are gaining popularity due to their lower hardware complexity compared with pipeline architectures. But the scaling of in-place RFFT architecture for higher lengths and higher throughput is a challenging issue due to increasing memory access conflict and higher memory bandwidth requirement. In this paper, a design approach is presented to develop an area-delay and energy-efficient architecture for in-place RFFT. Generally, an in-place fast Fourier transform (FFT) structure consists of a butterfly block which performs a set of butterfly operations in every clock cycle. From complexity analysis we find that in-place FFT structures with

larger butterfly blocks are more efficient in terms of area-time complexity and energy consumption. The resolution of memory access conflict is however more challenging for higher butterfly block sizes. Therefore, we have analyzed the data-flow and memory footprint of in-place RFFT architectures for different throughput requirements, and based on that, we have proposed here a strategy to partition the storage unit into several banks of smaller sizes (without increasing the overall memory size) to resolve the memory access conflicts by concurrent data-swapping between the banks. Synthesis result shows that the proposed structure with butterfly block of size 4 and 8 involves (~44% and ~57%) less area- delay product and (~54% and ~57%) less energy per sample than those of existing similar structure on average for different FFT lengths, respectively.

Design and Implementation of Flexible and Reconfigurable SDF-Based FFT Chip Architecture With Changeable-Radix Processing Elements

In this paper, we propose a flexible and reconfigurable changeable-radix fast Fourier transform (FFT) hardware architecture. It aims to support 48 different FFT sizes and up to 4096 FFT points, which are defined in current 3GPP-LTE communication system. The built-up design structure is primarily constructed on a radix-5 2 basis of single-path delay feedback FFT and up to 18 various changeable radices of FFT processing. A design technique of switchable FIFO usage approach is developed to efficiently manage FIFO arrangement for 48 FFT modes. In addition, a design technique of coarse and fine rotating is designed to effectively reduce twiddle-factor circuit area. By using TSMC 40-nm CMOS technology, an FFT ASIC implementation only has a core area occupation of

0.414 mm² and consumes 49.8 mW in average at maximal working frequency of 526.32 MHz. This innovative design work is competitive as compared to current state-of-the-art works, especially in terms of circuit area cost and power/energy performance evaluation.

Area-Time Efficient Architecture of FFT-Based Montgomery Multiplication

The modular multiplication operation is the most time-consuming operation for

number-theoretic cryptographic algorithms involving large integers, such as RSA and Diffie-Hellman. Implementations reveal that more than 75 percent of the time is spent in the modular multiplication function within the RSA for more than 1,024-bit moduli. There are fast multiplier architectures to minimize the delay and increase the throughput using parallelism and pipelining. However such designs are large in terms of area and low in efficiency. In this paper, we integrate the fast Fourier transform (FFT) method into the McLaughlin's framework, and present an improved FFT-based Montgomery modular multiplication (MMM) algorithm achieving high area-time efficiency. Compared to the previous FFT-based designs, we inhibit the zero-padding operation by computing the modular multiplication steps directly using cyclic and nega-cyclic convolutions. Thus, we reduce the convolution length by half. Furthermore, supported by the number-theoretic weighted transform, the FFT algorithm is used to provide fast convolution computation. We also introduce a general method for efficient parameter selection for the proposed algorithm. Architectures with single and double butterfly structures are designed obtaining low area-latency solutions, which we implemented on Xilinx Virtex-6 FPGAs. The results show that our work offers a better area-latency efficiency compared to the state-of-the-art FFT-based MMM architectures from and above 1,024-bit operand sizes. We have obtained area-latency efficiency improvements up to 50.9 percent for 1,024-bit, 41.9 percent for 2,048-bit, 37.8 percent for 4,096-bit and 103.2 percent for 7,680-bit operands. Furthermore, the operating latency is also outperformed with high clock frequency for length-64 transform and above.

Novel design of FFT using high radix butterfly of complex valued data

The Discrete Fourier Transform (DFT) is an important technique in the field of Digital Signal Processing (DSP) and Telecommunications, especially for applications in Orthogonal Frequency Division Multiplexing (OFDM) systems. The Fast Fourier Transform (FFT) is an efficient algorithm to compute the DFT and its inverse. The FFT processor plays a key role in the field of communication

systems such as Digital Video or Audio Broadcasting, Wireless LAN with Standards of IEEE 802.11, High Speed Digital Subscriber Lines etc. In this paper involves the implementation of a area efficient 8-point, 16-point, 32-point, 64- point and 128-point single path delay feedback (SDF) and folding technique using radix-2 DIT FFT algorithm. The proposed algorithm are used in radix-2 butterfly in all stage. The proposed algorithm is area efficient and consumed delay in previous algorithm. The all design are implementation vertex-2p device family Xilinx software.

VLSI architecture for FFT using radix-2 butterfly of complex valued data

The Discrete Fourier Transform (DFT) is an important technique in the field of Digital Signal Processing (DSP) and Telecommunications, especially for applications in Orthogonal Frequency Division Multiplexing (OFDM) systems. The Fast Fourier Transform (FFT) is an efficient algorithm to compute the DFT and its inverse. The FFT processor plays a key role in the field of communication systems such as Digital Video or Audio Broadcasting, Wireless LAN with Standards of IEEE 802.11, High Speed Digital Subscriber Lines etc. In this paper involves the implementation of a area efficient 8-point, 16-point, 32-point, 64- point and 128-point single path delay feedback (SDF) and folding technique using radix-2 DIT FFT algorithm. The proposed algorithms are used in radix-2 butterfly in all stage. The proposed algorithm is area efficient and consumed delay in previous algorithm. The all design are implementation vertex-2p device family Xilinx software.

CHAPTER 3

EXISTING SYSTEM

- It is observed that a pair of 2-point butterflies (referred to as butterfly block of size 2) is used in the existing RFFT designs.

FFT size N	BFB size $(L/2)$	MULT (L)	ADD $(3L/2)$	REG (N)	CT (ccs)	MUE (%)
32	2	4	6	32	40	12.5
	4	8	12	32	20	25
	8	16	24	32	10	50
	16	32	48	32	5	100
64	2	4	6	64	96	6.25
	4	8	12	64	48	12.5
	8	16	24	64	24	25
	16	32	48	64	12	50
128	2	4	6	128	224	3.125
	4	8	12	128	112	6.25
	8	16	24	128	56	12.5
	16	32	48	128	28	25
256	2	4	6	256	512	1.56
	4	8	12	256	256	3.12
	8	16	24	256	128	6.25
	16	32	48	256	64	12.5

Hardware and Time complexities of In-place RFFT structure

- Because of in-place nature of the computation, the storage unit complexity increases linearly only with FFT length and gives significantly to the total area for lengths higher than 32.
- During a clock cycle, the RFFT format of butterfly-block size 2 uses only 4 memory words out of N words stored in the storage unit.
- This results in low utilization of SU. For small butterfly blocks the combinational logic is a less part of storage place and the storage is highly low used.
- By increasing the butterfly block size, then with a slight rise in all-inclusive hardware complexity, memory can be well utilized as well as the throughput, area-delay product, and power- performance can also be improved substantially.
- The grading of in-place RFFT architecture for greater lengths and greater butterfly block sizes, however, is a difficult matter because of growing memory access dispute and large memory bandwidth requirement.

CHAPTER 4:

PROPOSED SYSTEM

- A design approach is presented for the development of area-delay-energy efficient for the ascendable in-place computation of 16-point FFT on real data.
- An in-place Fast Fourier transform (FFT) structure comprises of a butterfly block that performs a pack of butterfly operations in each clock cycle.
- Analyzed the data-flow as well as memory trace of in-place RFFT architectures for various throughput prerequisites, and depending on that, a strategic plan for partitioning the storage unit (SU) into various branches of small sizes (without increase in the whole memory size) to sort out the memory access disputes by simultaneous data-swapping between the banks.

- While comparing to other multiplier our system has more energy per sample
- The area of complexity is less in our system
- Execution speed is also high while comparing to existing one

BLOCK DIAGRAM

4.1 Storage unit design of proposed RFFT structure

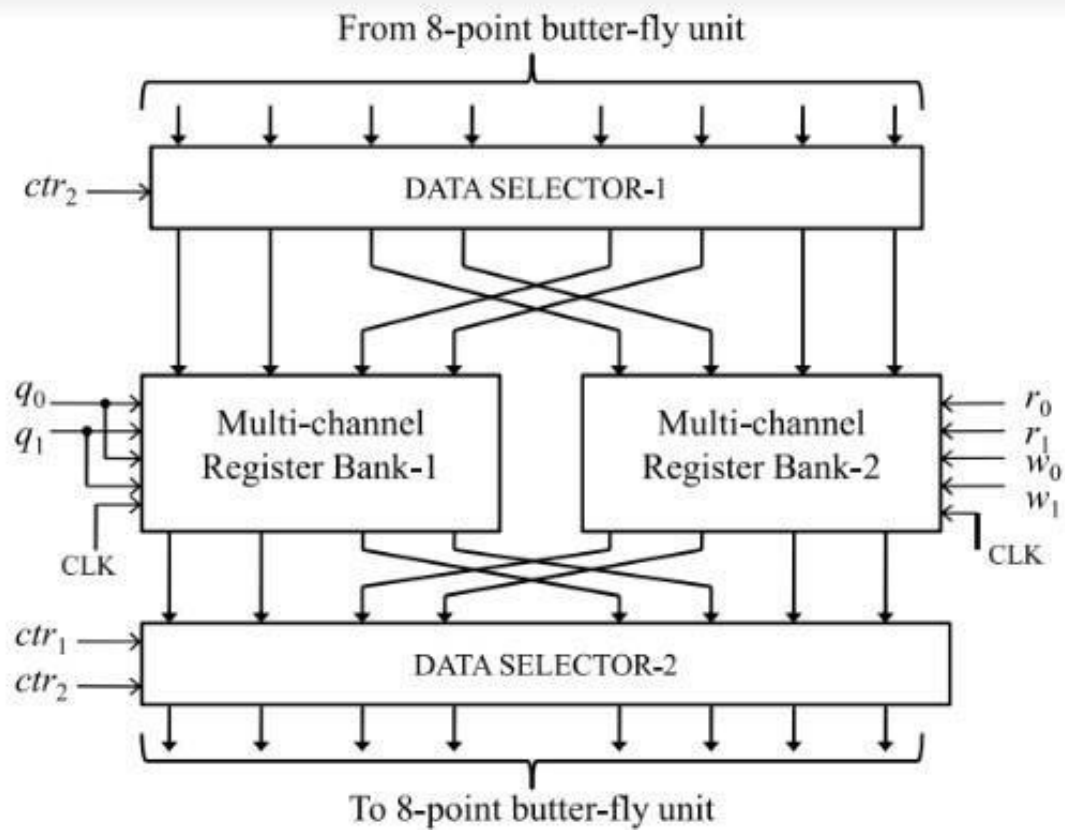


Fig 4.1 Storage Unit Design of Proposed RFFT structure

4.2 Internal structure of multi-channel register bank

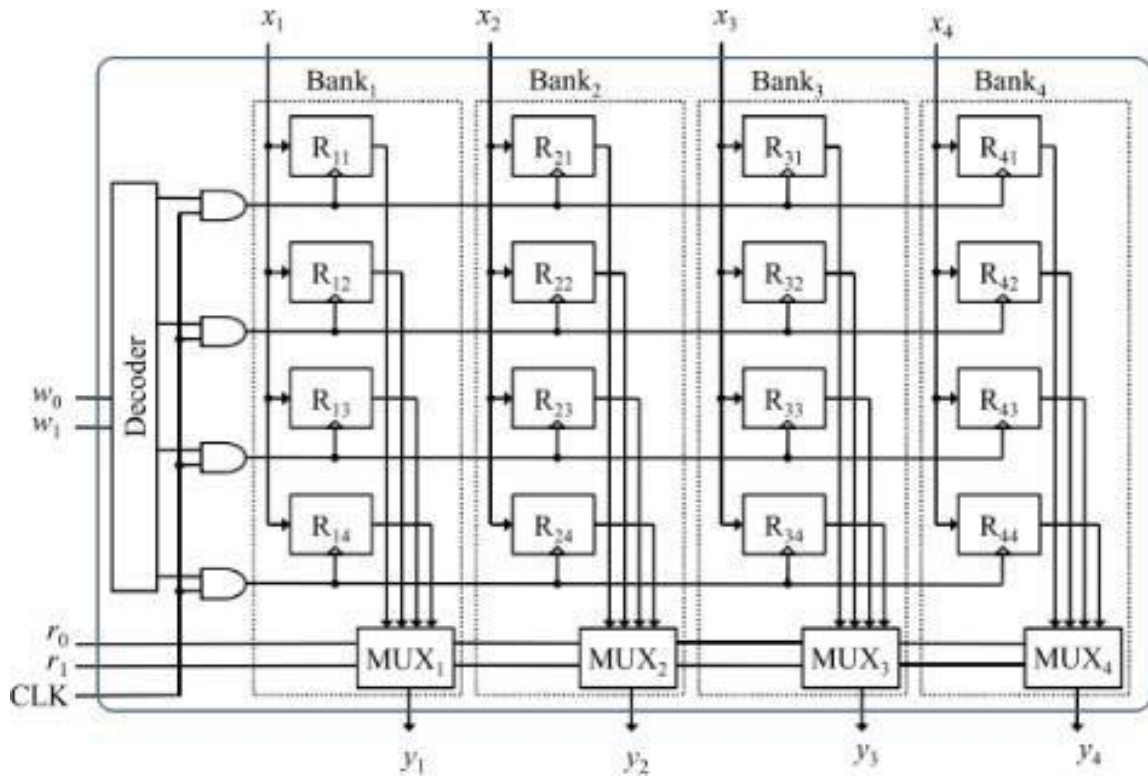


Fig 4.2 Internal Structure of Multi-Channel Register Bank

CONTENT:

4.3 Data-Flow Analysis of Storage Unit

- The BCU for butterfly block size ($L/2$) takes (N/L) clock cycles to complete the computations of one butterfly stage and takes $(N/L)\log_2 N$ clock cycles to complete all the butterfly operations for N -point FFT.
- Butterfly blocks size $(L/2) = 4$ and FFT size $N = 32$ as a sample to examine the input-output data-flow resulting in memory access disputes and essential data-swapping instances.

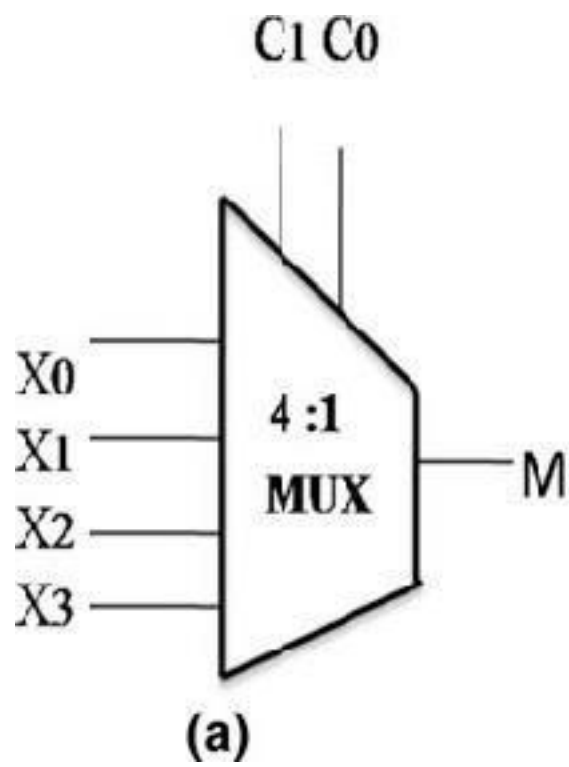
- The input-output data-flow of other butterfly block sizes can also be undertaken in similar manner. Computations of each butterfly stages of 32-point FFT is performed in a set of 4 successive clock cycles and the entire computation is performed in 20 clock cycles.
- Throughout the last 4 clock cycles of period of 20 clock cycles the FFT elements are calculated out as output of the BCU while samples of the next input sequence are stored in the SU during the same period.

4.4 Storage Unit Design

- The design of SU of the proposed structure for 32-point RFFT and butterfly block-size 4 is shown in Figure
- It has two multi-channel register-banks (MCRBs) as well as two data-selectors.
- The MCRB executes various memory banks in one unit that uses a common address decoding logic for all the banks.
- Internal structure of multi-channel register bank for butterfly block-size 4 and $N = 32$ and Storage unit design of proposed RFFT structure for butterfly block-size 4 and $N = 32$

4.5 MULTIPLEXER

- Multiplexer is a combinational logic circuit used to select only one input among several inputs based on selection lines.
- A multiplexer can act as digital switch.
- A multiplexer is also known as data selector.
- A mux there can be 2^n inputs, n selection lines and only one output.
- Multiplexers can either be digital circuits made of high speed logic gates or binary data.



C1	C0	M
0	0	X0
0	1	X1
1	0	X2
1	1	X3

(b)

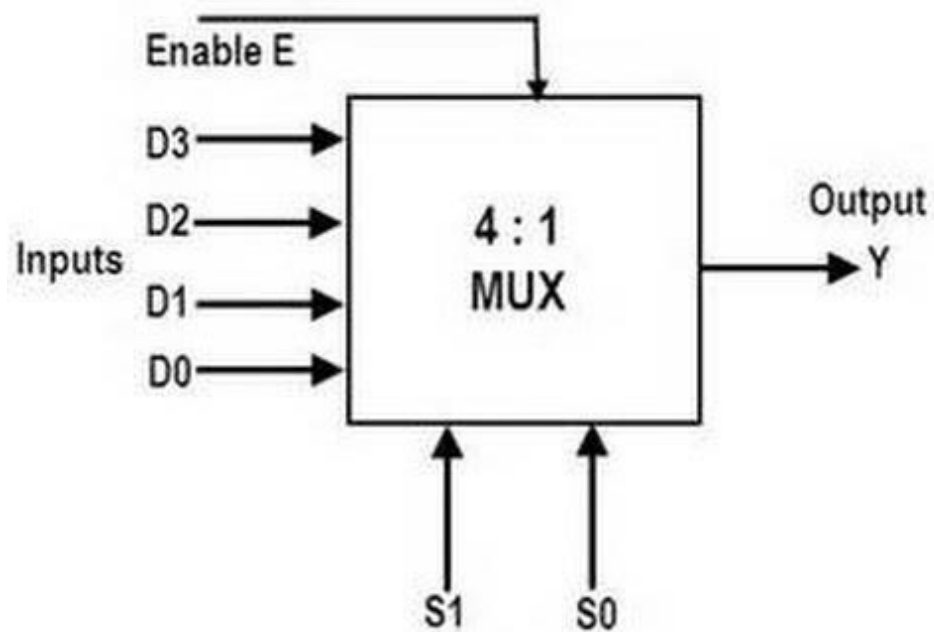


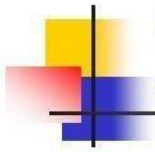
Fig 4.3 4:1 MUX

BOOLEAN TABLE:

Select Data Inputs			Output
S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

Table 4.1 Boolean Table

MULTIPLEXER:

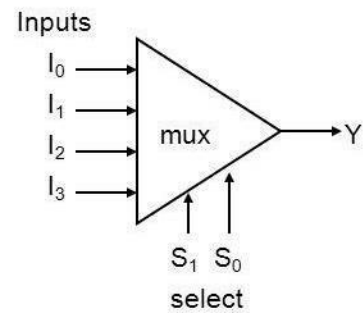
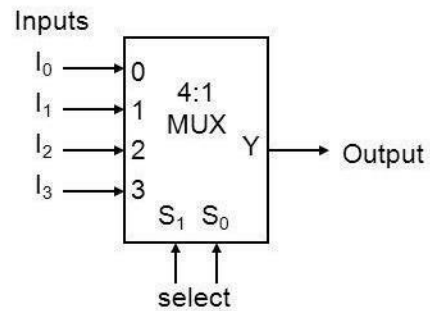


Multiplexer

- Truth table for a 4-to-1 multiplexer:

I_0	I_1	I_2	I_3	S_1	S_0	Y
d_0	d_1	d_2	d_3	0	0	d_0
d_0	d_1	d_2	d_3	0	1	d_1
d_0	d_1	d_2	d_3	1	0	d_2
d_0	d_1	d_2	d_3	1	1	d_3

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



Multiplexer

47

Fig 4.4 Multiplexer

CHAPTER 5: RESULTS AND DISCUSSION

SOFTWARE REQUIREMENTS:

- XILINX ISE DESIGN SUITE 14.5
- MODELSIM

XILINX ISE:

Xilinx ISE (Integrated Synthesis Environment) may be a discontinued software tool from Xilinx for synthesis and breakdown of HDL designs, which mainly targets improvement embedded firmware for Xilinx FPGA and CPLD microcircuit (IC) product families. Use of the previous released edition from October 2013 endure for in-system programming of estate hardware designs including earlier FPGAs and CPLDs otherwise opened by the restored design tool, Vivado Design Suite.

ISE makes the developer to synthesize ("compile") their designs, perform timing analysis, Inquire RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. Other components shipped with the Xilinx ISE involves carried the Embedded Development Kit (EDK), a Software Development Kit (SDK) and ChipScope Pro. The Xilinx ISE is mainly used for circuit anthesis and style , Although ISIM or the ModelSim logic simulator is used for system-level authentication.

As regularly practiced in the commercial electronic design automation sector, Xilinx ISE is tightly-coupled to the architecture of Xilinx's own chips (the internals of which are highly proprietary) and can't be used with FPGA parts from other dealers. Given the high proprietary nature of the Xilinx hardware product lines, it's highly impossible to use opensource alternatives to tooling involve from Xilinx, although as of 2020, some attempts are being made.

SIMULATION:

System-level measuring also be conducted aligning with ISIM or the ModelSim logic

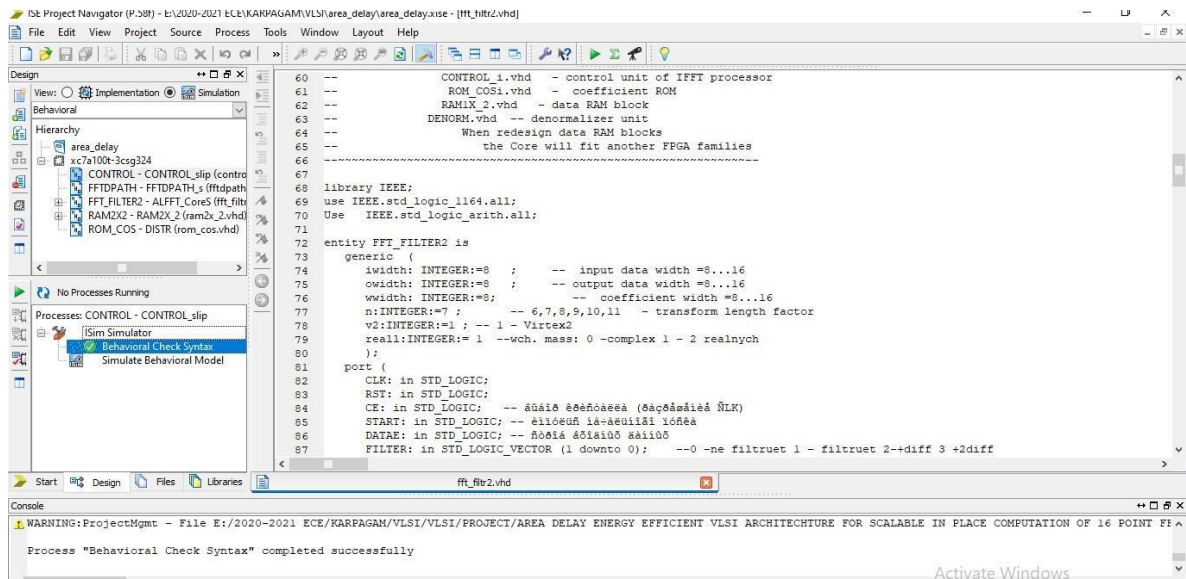
simulator, and such test programs must even be in HDL languages. Test bench programs may involve simulated input waveforms, or monitors which observe and authenticate the outputs of the device under test.

ModelSim or ISIM could also be used to perform the subsequent sorts of simulations:

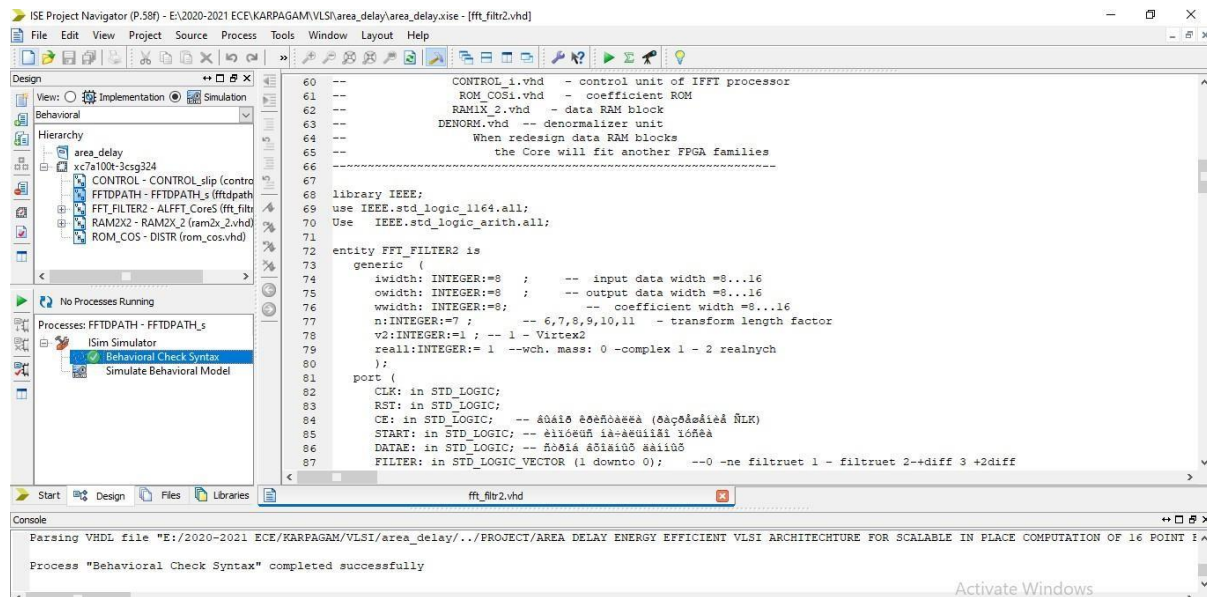
- Logical verification, to assure the module produces desired results
- Behavioral verification, to authenticate logical and timing issues
- Post-place & route simulation, to verify behavior after employment of the module in the conquerable logic of the FPGA

OUTPUT:

Control path output:



FFT OUTPUT:



FFT FILTER OUTPUT:

The screenshot displays the ISE Project Navigator interface. The left pane shows the project hierarchy with the following components:

- area_delay
 - xc7a100t-3csg324
 - CONTROL - CONTROL_slip (contro
 - FFDPATH - FFDPATH_s (fftdpath
 - FFT_FILTER2 - ALFFT_Core5 (fft_filt
 - RAM2X2 - RAM2X_2 (ram2x_2.vhd)
 - ROM_COS - DISTR (rom_cos.vhd)

The right pane shows the source code for `fft_filt2.vhd`:

```

60 --          CONTROL_1.vhd -- control unit of IFFT processor
61 --          ROM_COS1.vhd -- coefficient ROM
62 --          RAM1X_2.vhd -- data RAM block
63 --          DENORM.vhd -- denormalizer unit
64 --          When redesign data RAM blocks
65 --          the Core will fit another FPGA families
66 -----
67
68 library IEEE;
69 use IEEE.std_logic_1164.all;
70 use IEEE.std_logic_arith.all;
71
72 entity FFT_FILTER2 is
73   generic (
74     iwidth: INTEGER:=8 ; -- input data width =8...16
75     owidth: INTEGER:=8 ; -- output data width =8...16
76     wwidth: INTEGER:=8; -- coefficient width =8...16
77     n:INTEGER:=7 ; -- 6,7,8,9,10,11 -- transform length factor
78     v2:INTEGER:=1 ; -- 1 - Virtex2
79     reall:INTEGER:= 1 --wch. mass: 0 -complex 1 - 2 realnych
80   );
81   port (
82     CLK: in STD_LOGIC;
83     RST: in STD_LOGIC;
84     CE: in STD_LOGIC; -- a0a16 a0e0a0e0a (a0a0a0a1e0 NLK)
85     START: in STD_LOGIC; -- a10e0u ia-a0u11a1 1oñea
86     DATAE: in STD_LOGIC; -- a0e1a a01a1u0 a11u0
87     FILTER: in STD_LOGIC_VECTOR (1 downto 0); --0 -ne filtruet 1 - filtruet 2--diff 3 +2diff

```

The bottom pane shows the console output:

```

Parsing VHDL file "E:/2020-2021 ECE/KARPAGAM/VLSI/area_delay/./PROJECT/AREA DELAY ENERGY EFFICIENT VLSI ARCHITECHTURE FOR SCALABLE IN PLACE COMPUTATION OF 16 POINT 1
Process "Behavioral Check Syntax" completed successfully

```

ISE Project Navigator (P.58) - E:\2020-2021 ECE\KARPGAM\VLSI\area_delay\area_delay.xise - [fft_filt_2.vhd]

File Edit View Project Source Process Tools Window Layout Help

Design

View: ☐ Implementation ☒ Simulation

Behavioral

Hierarchy

- area_delay
 - xcf1a100t-3csg324
 - CONTROL - CONTROL_slip (contro)
 - FFTDPATH - FFTDPATH_s (fftdpath)
 - FFT_FILTER2 - ALFFT_CoreS (fft_filt)
 - RAM2X2 - RAM2X_2 (ram2x_2.vhd)
 - ROM_COS - DISTR (rom_cos.vhd)

No Processes Running

Processes: RAM2X2 - RAM2X_2

- ISim Simulator
- ☒ Behavioral Check Syntax
- Simulate Behavioral Model

```

60 --          CONTROL_1.vhd   - control unit of IFFT processor
61 --          ROM_COS1.vhd   - coefficient ROM
62 --          RAM1X_2.vhd   - data RAM block
63 --          DENORM.vhd    -- denormalizer unit
64 --          When redesign data RAM blocks
65 --          the Core will fit another FPGA families
66
67
68 library IEEE;
69 use IEEE.std_logic_1164.all;
70 Use IEEE.std_logic_arith.all;
71
72 entity FFT_FILTER2 is
73   generic (
74     iwidth: INTEGER:=8 ;    -- input data width =8...16
75     owidth: INTEGER:=8 ;    -- output data width =8...16
76     width: INTEGER:=8;      -- coefficient width =8...16
77     n:INTEGER:=7 ;          -- 6,7,8,9,10,11 - transform length factor
78     v2:INTEGER:=1 ;        -- 1 - Virtex2
79     reall:INTEGER:=1 --wch. mass: 0 -complex 1 - 2 realnych
80   );
81   port (
82     CLK: in STD_LOGIC;
83     RST: in STD_LOGIC;
84     CE: in STD_LOGIC;      -- 00000 0000000000 (0000000000 0000)
85     START: in STD_LOGIC;  -- 000000 000000 000000 000000
86     DATAE: in STD_LOGIC; -- 00000 000000 000000 000000
87     FILTER: in STD_LOGIC_VECTOR (1 downto 0); --0 -ne filteruet 1 - filteruet 2--diff 3+2diff

```

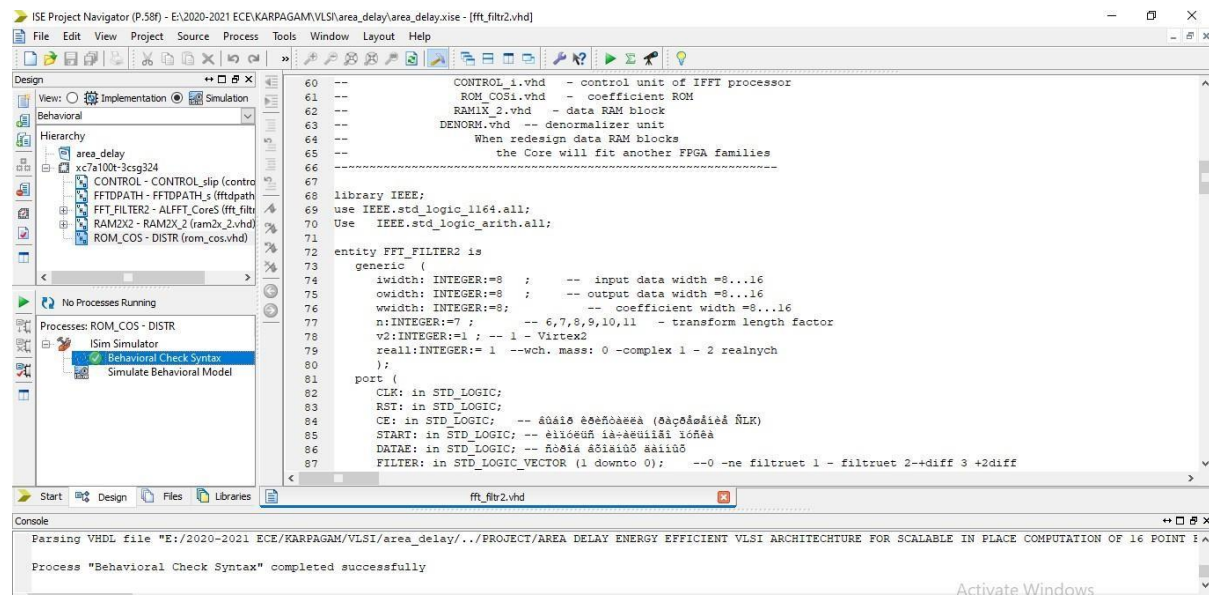
fft_filt_2.vhd

Console

Parsing VHDL file "E:\2020-2021 ECE\KARPGAM\VLSI\area_delay\PROJECT\AREA DELAY ENERGY EFFICIENT VLSI ARCHITECTURE FOR SCALABLE IN PLACE COMPUTATION OF 16 POINT

Process "Behavioral Check Syntax" completed successfully

ROM OUTPUT:



MODELSIM OUTPUT:



CHAPTER 6: SUMMARY AND CONCLUSION

COMPARISON		
	PROPOSED	EXISTING
AREA	36nm	90nm
POWER	1.2w	1.0w
SPEED	100ps	23ps

Table 6.1 Comparison

CONCLUSION:

A design approach to develop efficient architecture for in-place RFFT which could be scaled for higher throughput and larger FFT sizes. The proposed structure for butterfly block size 4 and 8, involves significantly less EPS than the existing designs. The proposed design for butterfly block size 16 can meet the throughput requirement of Ultra wide band (UWB) and WSN applications while the proposed design for butterfly block sizes 8 can able to meet the throughput requirement of WSN and medical implantable devices.

REFERENCES:

- [1] E. H. Wold and A. M. Despain, "Pipeline and parallel-pipeline FFT processors for VLSI implementations," *IEEE Trans. Comput.*, vol. C-33, no. 5, pp. 414–426, May 1984.
- [2] Y.-N. Chang and K. K. Parhi, "An efficient pipelined FFT architecture," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 50, no. 6, pp. 322–325, Jun. 2003.
- [3] H. Sorensen, D. Jones, M. Heideman, and C. Burrus, "Real-valued fast Fourier transform algorithms," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, no. 6, pp. 849–863, Jun. 1987.
- [4] S.-F. Hsiao and W.-R. Shiue, "Design of low-cost and high-throughput linear arrays for DFT computations: Algorithms, architectures, and implementations," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 11, pp. 1188–1203, Nov. 2000.
- [5] M. Garrido, K. K. Parhi, and J. Grajal, "A pipelined FFT architecture for real-valued signals," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 12, pp. 2634–2643, Dec. 2009.
- [6] M. Ayinala, M. Brown, and K. K. Parhi, "Pipelined parallel FFT architectures via folding transformation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 6, pp. 1068–1081, Jun. 2012.
- [7] A. Wang and A. P. Chandrakasan, "Energy-aware architectures for a realvalued FFT implementation," in *Proc. Int. Symp. Low Power Electron. Design*, Aug. 2003, pp. 360–365.

- [8] H.-F. Chi and Z.-H. Lai, "A cost-effective memory-based real-valued FFT and Hermitian symmetric IFFT processor for DMT-based wire-line transmission systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 6, May 2005, pp. 6006–6009.
- [9] L. G. Johnson, "Conflict free memory addressing for dedicated FFT hardware," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 39, no. 5, pp. 312–316, May 1992.
- [10] M. Ayinala, Y. Lao, and K. K. Parhi, "An in-place FFT architecture for real-valued signals," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 10, pp. 652–656, Oct. 2013. $3ADP = area \times MCP \times (N/L) \log_2 N$ $4EPS = power \times MCP \times [(N/L) \log_2 N] / N$
- [11] B. G. Jo and M. H. Sunwoo, "New continuous-flow mixedradix (CFMR) FFT processor using novel in-place strategy," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 5, pp. 911–919, May 2005.
- [12] P. K. Meher, B. K. Mohanty, S. K. Patel, S. Ganguly, and T. Srikanthan, "Efficient VLSI architecture for decimation-in-time fast Fourier transform of real-valued data," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 12, pp. 2836–2845, Dec. 2015.
- [13] S. R. Kuang, J. P. Wang, and C. Y. Guo, "Modified Booth multipliers with a regular partial product array," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 5, pp. 404–408, May 2009.
- [14] R. M. Jiang, "An area-efficient FFT architecture for OFDM digital video broadcasting," *IEEE Trans. Consume. Electron.*, vol. 53, no. 4, pp. 1322–1326, Nov. 2007.
- [15] L. Yang, K. Zhang, H. Liu, J. Huang, and S. Huang, "An efficient locally pipelined FFT processor," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 7, pp. 585–589, Jul. 2006.