ISSN 2277-2685
IJESR/Jan-Mar. 2025/ Vol-15/Issue-1s/201-212
Yarakala Sreeja *et. al.*, / International Journal of Engineering & Science Research

Advanced FIFO Structure For Router In Bi-Noc

Ms. B Eleena, Yarakala Sreeja, Boya Srija, Gona Teja Sree
[1] Assistant professor, Electronics and Communication Engineering, BRECW
[2,3,4]B.Tech Students, Department of Electronics and Communication Engineering, BRECW

Abstract
Network on chip (NoC) becomes a promising solution for intercommunication infrastructure in System on Chip (SoC) as traditional methods exhibit severe bottlenecks at intercommunication among processor elements. However, designing of NoC is majorly complex because of lot of issues raise in terms of performance metrics such as system scalability, latency, power consumption and signal integrity. This paper discussed issues of memory unit in router and thereafter, proposing advanced memory structure. To obtain efficient data transfer, FIFO buffers are implemented in distributed RAM and virtual channels for FPGA based NoC. An advanced FIFO based memory units are proposed in NoC router and the performance is evaluated in Bi-directional NoC (Bi-NoC). The major motivation of this paper is to reduce burden of router while improving FIFO internal structure. Toenhance the speed data transfer, Bi-NoC with a self-configurable intercommunication channel is proposed. The Simulations and synthesis results are proven guaranteed throughput, predictable latency, and fair network access highly provided when compared to recent works.

Keywords: Bi-NoC; FIFO; Virtual Channel; Switch Allocator; Router; SoC.

1-INTRODUCTION

System on chip (SOC) is a complex interconnection of various functional elements. It creates communication bottleneck in the gigabit communication due to its bus-based architecture. Thus, there was need of system that explicit modularity and parallelism, network on chip possess many such attractive properties and solve the problem of communication bottleneck. It basically works on the idea of interconnection of cores using on chip network.

The communication on network on chip is carried out by means of router, so for implementing better NOC, the router should be efficiently design. This router supports four parallel connections at the same time. It uses store and forward type of flow control and Fsm Controller deterministic routing which improves the performance of router. The switching mechanism used here is packet switching which is generally used on network on chip.

In packet switching the data the data transfers in the form of packets between cooperating routers and independent routing decision is taken. The store and forward flow mechanism are best because it does not reserve channels and thus does not lead to idle physical channels. The arbiter is of rotating priority scheme so that every channel once get chance to transfer its data. In this router both input and output buffering are used so that congestion can be avoided at both sides.

A router is a device that forwards data packets across computer networks. Routers perform the data "traffic direction" functions on the Internet. A router is a microprocessor- controlled device that is connected to two or more data lines from different networks. When a data packet comes in on one of the lines. The router reads the address information in the packet to determine its ultimate destination. Then, using information in its routing table, it directs the packet to the next network on its journey. The router is a”

Four Port Network Router" has a one input port from which the packet enters. It has three output ports where the packet is driven out. Packet contains 3 parts. They are Header, data and frame check sequence.

Packet width is 8 bits and the length of the packet can be between 1 byte to 63 bytes. Packet header contains three fields DA and length. Destination address (DA) of the packet is of 8 bits. The switch drives the packet to respective ports based on this destination address of the packets. Each output port has 8-bit unique port address If the destination address of the packet matches the port address, then switch drives the packet to the output port, Length of the data is of 8 bits and from 0 to 63. Length is measured in terms of bytes. Data should be in terms of bytes and can take anything. Frame check sequence contains the security check of the packet. It is calculated over the header and data.

2-LITERATURE SURVEY

A router is a device that forwards data packet between computer networks, creating an overlay internetwork. A router is connected to two or more data lines from different networks. When a data packet comes in one of the lines, the router reads the address information in the packet to determine its ultimate destination.

Then, using information in its routing table or routing policy, it directs the packet to the next network on its journey. Routers perform the "traffic directing" functions on the Internet. A data packet is typically forwarded from one router to another through the networks that constitute the internetwork until it reaches its destination node.

Routers may also be used to connect two or more logical groups of computer devices known as subnets, each with a different sub-network address. The subnets addresses recorded in the router do not necessarily map directly to the physical interface connections. Forwarding an IP datagram generally requires the router to choose the address and relevant interface of the next-hop router or (for the final hop) the destination host.

In Transmission Control Protocol/Internet Protocol (TCP/IP) networking, routers are used to interconnect the hardware and software used on different physical network segments called subnets. Routers are also used to forward IP packets between each of the subnets. Determine the physical layout of your network, including the number of routers and subnets you need, before proceeding with the instructions in this guide.

Routers may provide connectivity within enterprises, between enterprises and the Internet, and between internet service providers (ISPs) networks. The largest routers (such as the Cisco CRS-1 or Juniper T1600) interconnect the various ISPs, or may be used in large enterprise networks.

Smaller routers usually provide connectivity for typical home and office networks. Other networking solutions may be provided by a backbone Wireless Distribution System (WDS), which avoids the costs of introducing networking cables into buildings. All sizes of routers may be found inside enterprises.

The most powerful routers are usually found in ISPs, academic and research facilities. Large businesses may also need more powerful routers to cope with ever increasing demands of internet data traffic.

3-ROUTER DESIGN SPECIFICATION

Router is a packet-based protocol. Router drives the incoming packet which comes from the input port to output ports based on the address contained in the packet. The router has a one input port from which the packet enters. It has three output ports where the packet is driven out. The router has an active low synchronous input resetn which resets the router.

Data packet moves in to the input channel of one port of router by which it is forwarded to the output channel of other port. Each input channel and output channel have its own decoding logic which increases the performance of the router. Buffers are present at all ports to store the data temporarily.

The buffering method used here is store and forward. Control logic is present to make arbitration decisions. Thus, communication is established between input and output ports. According to the destination path of data packet, control bit lines of FSM are set.

The movement of data from source to destination is called switching mechanism

4-FOUR PORT ROUTER ARCHITECTURE

Router Architecture:

The Four Router Design is done by using of the three blocks. the blocks are 8-Bit Register, Router controller and output block. the router controller is design by using FSM design and the output block consists of three fifo's combined together the fifo's are store packet of data and when u want to data that time the data read from the FIFO's.

In this router design has three outputs that is 8-Bit size and one 8_bit data port it using to drive the data into router we are using the global clock and reset signals, and the err signal and suspended data signals are outputs of the router. the FSM controller gives the err and suspended_data_in signals. these functions are discussed clearly in below FSM description.

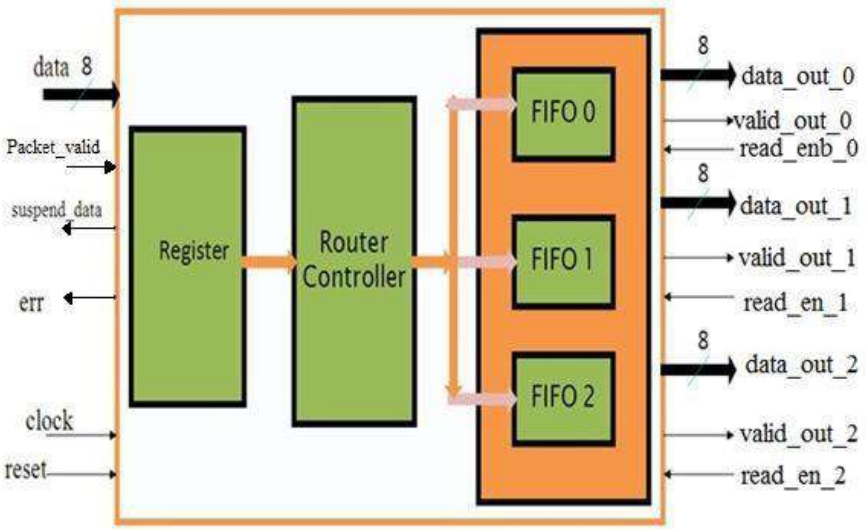


Figure- 4.1 Four Port Router Architecture

The router_reg module contains the status, data and parity registers for the Network router_1x3. These registers are latched to new status or input data through the control signals provided by the fsm_router. There are 3 FIFO for each output port, which stores the data coming from input port based on the control signals provided by fsm_router

The Router blocks Diagram shown below fig…

Router blocks are

- Register
- Router controller (FSM)
- FIFO Output Block

5-SOFTWARE AND HARDWARE COMPONENTS

Software used for the Project:

Verilog:

In the semiconductor and electronic design industry, Verilog is a hardware description language (HDL) used to model electronic systems. Verilog HDL is most commonly used in the design, verification, and implementation of digital logic chips at the register-transfer level of abstraction. It is also used in the verification of analog and mixed-signal circuits.

Hardware description languages such as Verilog differ from software programming languages because they include ways of describing the propagation of time and signal dependencies (sensitivity). There are two assignment operators, a blocking assignment (=), and a non-blocking (<=) assignment. The non-blocking assignment allows designers to describe a state-machine update without needing to declare and use temporary storage variables. Since these concepts are part of Verilog's language semantics, designers could quickly write descriptions of large circuits in a relatively compact and concise form. At the time of Verilog's introduction (1984), Verilog represented a tremendous productivity improvement for circuit designers who were already using graphical schematic capture software and specially written software programs to document and simulate electronic circuits.

The designers of Verilog wanted a language with syntax similar to the C programming language, which was already widely used in engineering software development. Like C, Verilog is case-sensitive and has a basic preprocessor (though less sophisticated than that of ANSI C/C++). Its flow keywords (if/else, for, while, case, etc.) are equivalent, and its operator precedence is compatible. Syntactic differences include variable declaration (Verilog requires bit-widths on net/reg types), demarcation of procedural blocks (begin/end instead of curly braces {}), and many

A Verilog design consists of a hierarchy of modules. Modules encapsulate design hierarchy, and communicate with other modules through a set of declared input, output, and bidirectional ports. Internally, a module can contain any combination of the following: net/variable declarations (wire, reg, integer, etc.), concurrent and sequential statement blocks, and instances of other modules (sub-hierarchies). Sequential statements are placed inside a begin/end block and executed in sequential order within the block. However, the blocks themselves are executed concurrently, making Verilog a dataflow language.

Verilog's concept of 'wire' consists of both signal values (4-state: "1, 0, floating, undefined") and strengths (strong, weak, etc.). This system allows abstract modeling of shared signal lines, where multiple sources drive a common net. When a wire has multiple drivers, the wire's (readable) value is resolved by a function of the source drivers and their strengths.

Xilinx-ISE:

Xilinx ISE (Integrated Software Environment) is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. The ISE® Design Suite is the Xilinx® design environment, which allows you to take your design from design entry to Xilinx device programming. With specific editions for logic, embedded processor, or Digital Signal Processing (DSP) system designers, the ISE Design Suite provides an environment tailored to meet your specific design needs.

Xilinx Design Flow Overview:

The following steps are involved in the realization of a digital system using Xilinx FPGAs, as illustrated by the following figure.

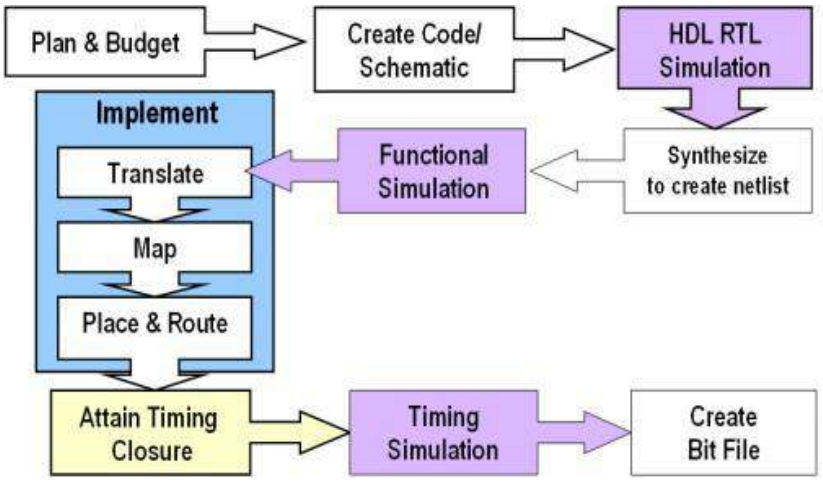


Figure 5.1: Overview of the various steps involved in the design flow of a digital system

Hardware used for the Project

FPGA:

FPGA implementations have the potential to be parallel using a mixture of these two forms. For example, the FPGA could be configured to partition the image and distribute the resulting sections to multiple pipelines all of which could process data concurrently. Such parallelization is subject to the processing mode and hardware constraints of the system.

The Advantage of Using FPGAs:

Image processing is difficult to achieve on a serial processor. This is due to the large data set required to represent the image and the complex operations that need to be performed on the image. Consider video rates of 25 frames per second, a single operation performed on every pixel of a 768X576 color image (Standard PAL frame) equates to 33 million operations per second. FPGA consists of a matrix of logic blocks that are connected by an interconnect network. Both the logic blocks and the interconnect network are reprogrammable allowing application specific hardware to be constructed, while at the same time maintaining the ability to change the functionality of the system with ease. As such, an FPGA offers a compromise between the flexibility of general-purpose processors and the hardware-based speed of ASICs.

6-RESULTS AND DISCUSSIONS

Creating a New Project

Xilinx Tools can be started by clicking on the Project Navigator Icon on the Windows desktop. This should open up the Project Navigator window on your screen. This window shows the last accessed project.

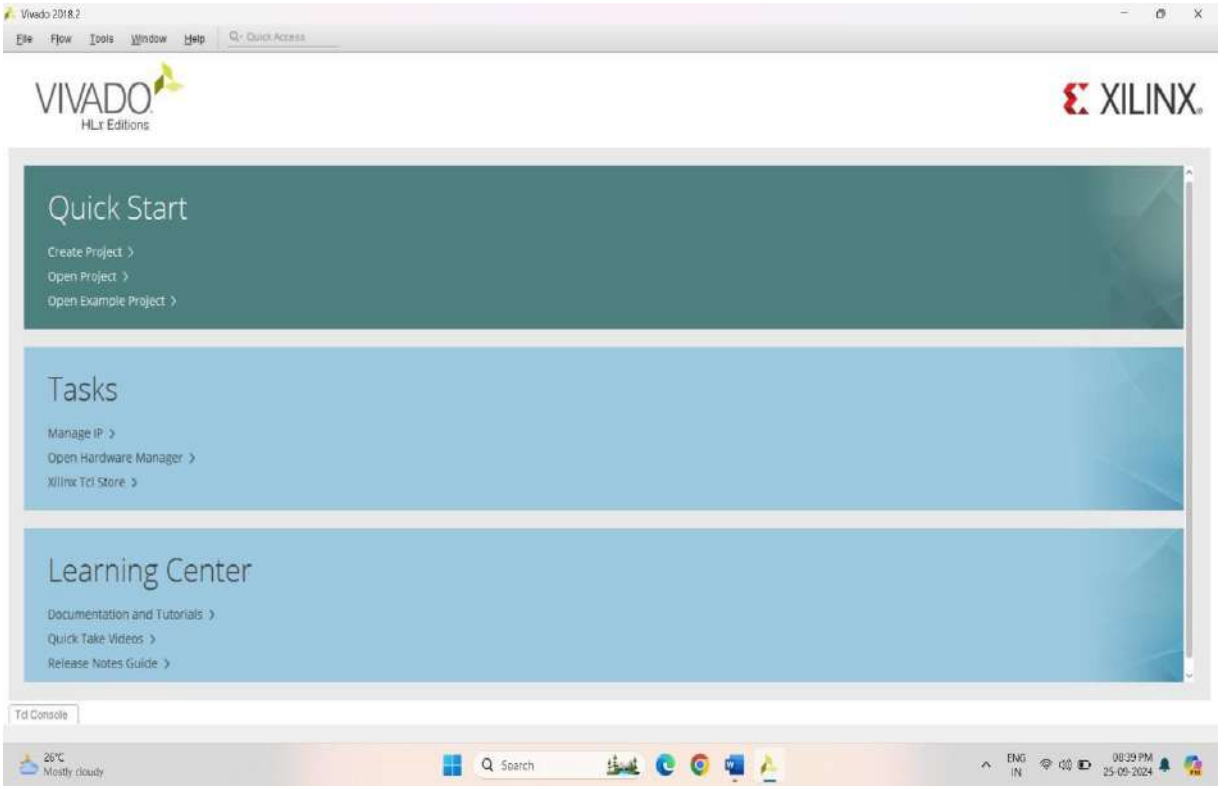


Fig 6.1: Vivado Project Navigator window (snapshot from vivado software)

Opening a project Select File->New Project to create a new project. This will bring up a new project window on the desktop. Fill up the necessary entries as follows

New Project Initiation

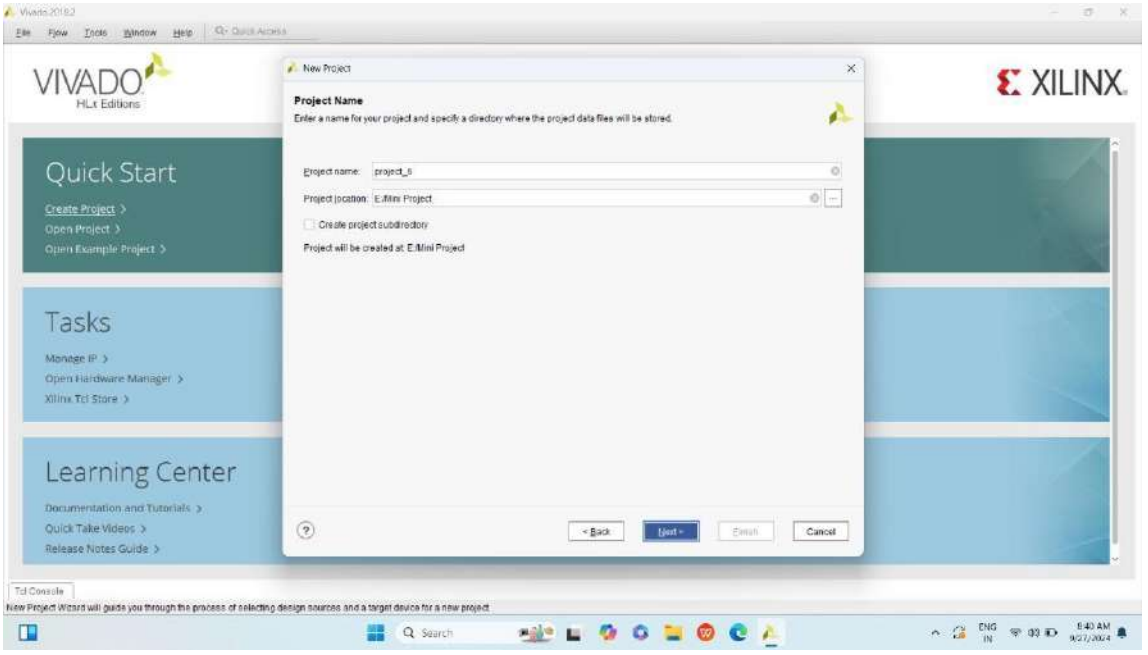


Fig 6.2: New Project Initiation window (snapshot from Vivado software)

6.3 Adding source code into the project

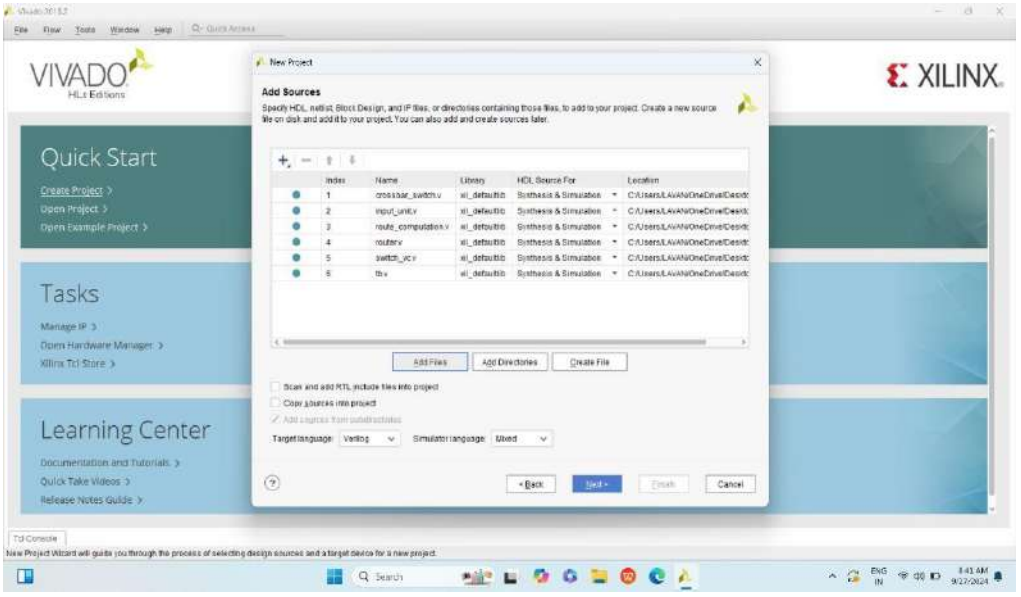


Fig 6.3: Adding source code into the project (snapshot from Vivado software)

In this step we need to add the codes according to the block diagram that is proposed system in the vivado.

6.4 Selecting the Board Required

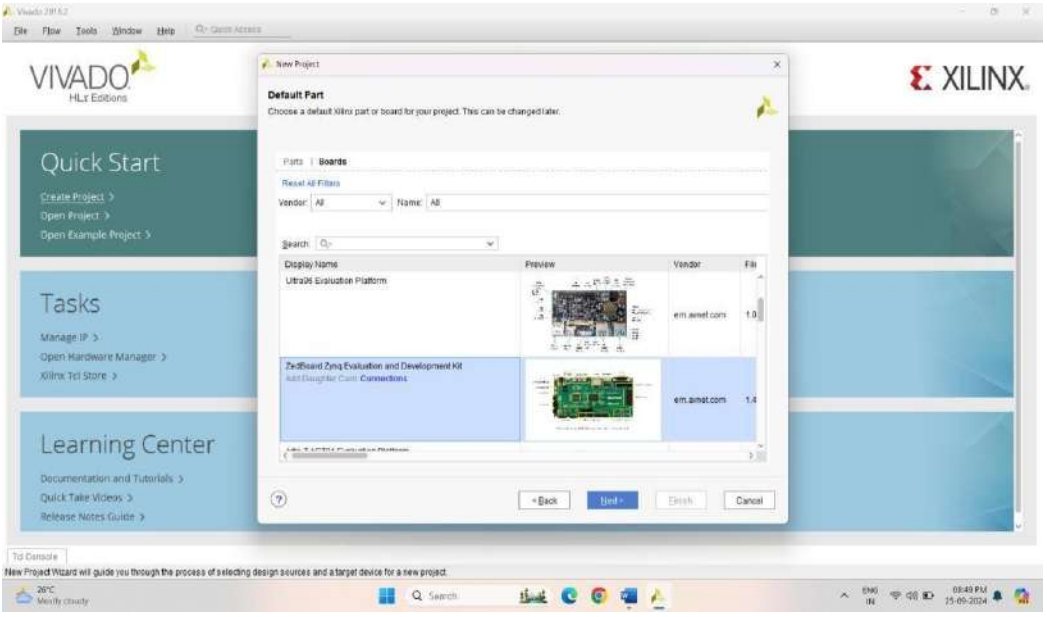
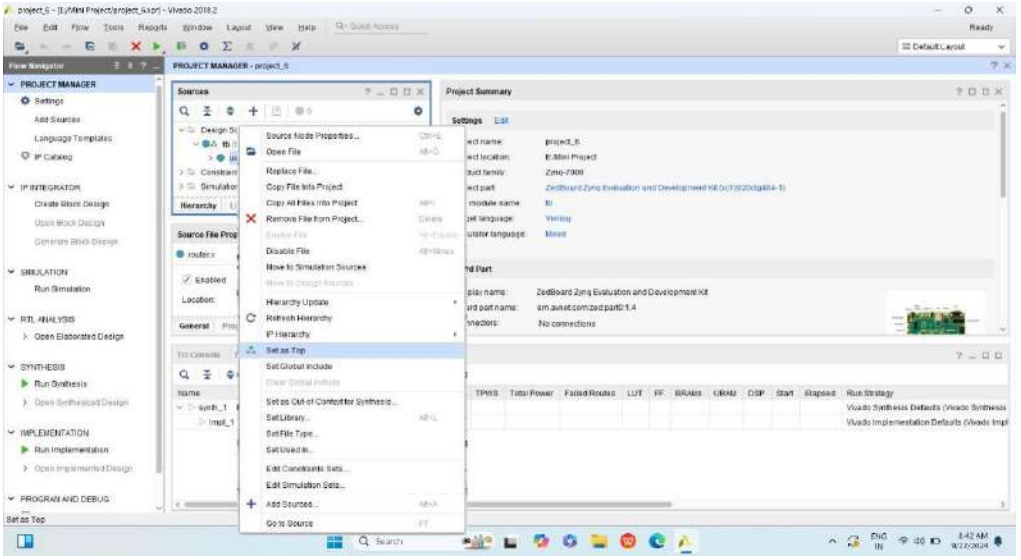


Fig 6.4: Selecting the Board Required (snapshot from Vivado software)

In this step we need to select zed board zynq evaluation and development kit for the dumping of the code into this



board to get the better performance of the vivado software and click on the next.

This zeb board is more efficient than any other boards also we can use the other but with the good power efficiency and delay and area

6.5 New Project Summary

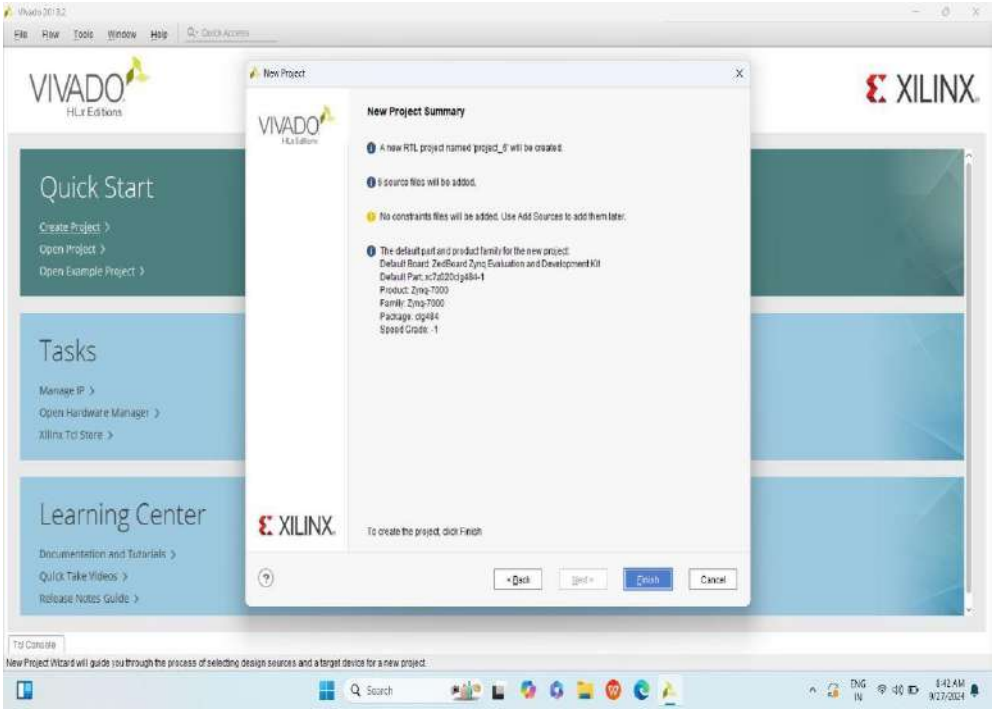


Fig 6.5: New Project Summary (A Snapshot from Vivado Software)

Make sure that all the files are available with green marks and then click on ok to continue. Then a window will open which shows that to create the project we need to click on finish.**6.6** Editor Window

Fig 6.6: Editor Window (ASnapshot from Vivado Software)

Select the uut _ POSIT multiplier and set as the Top in the editor window and open elaborated

design to get the RTL diagram and run synthesis and implementation

6.7 Code

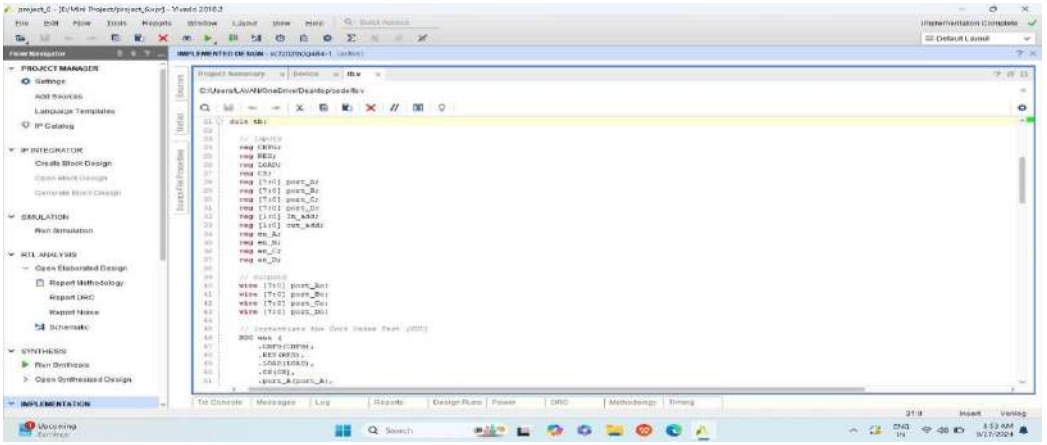


Fig 6.7: Code for fifo (A Snapshot from Vivado Software)

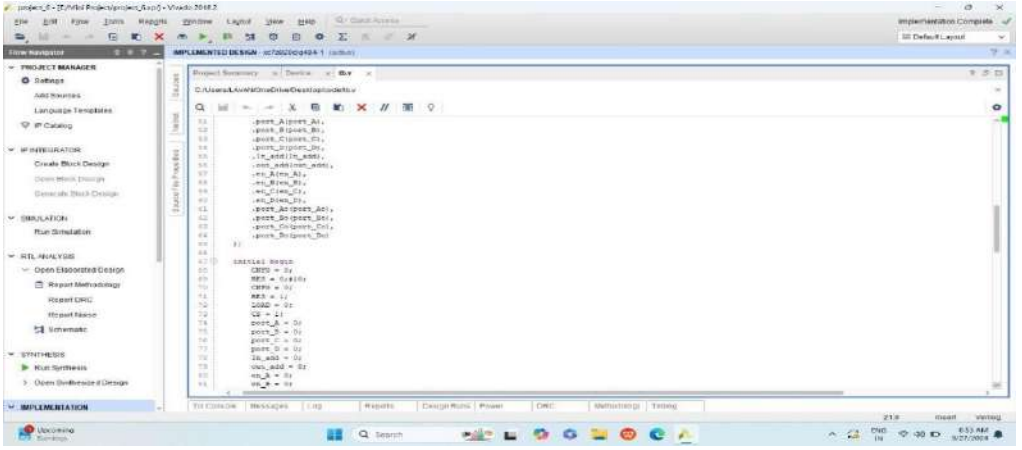


Fig 6.7.1 code

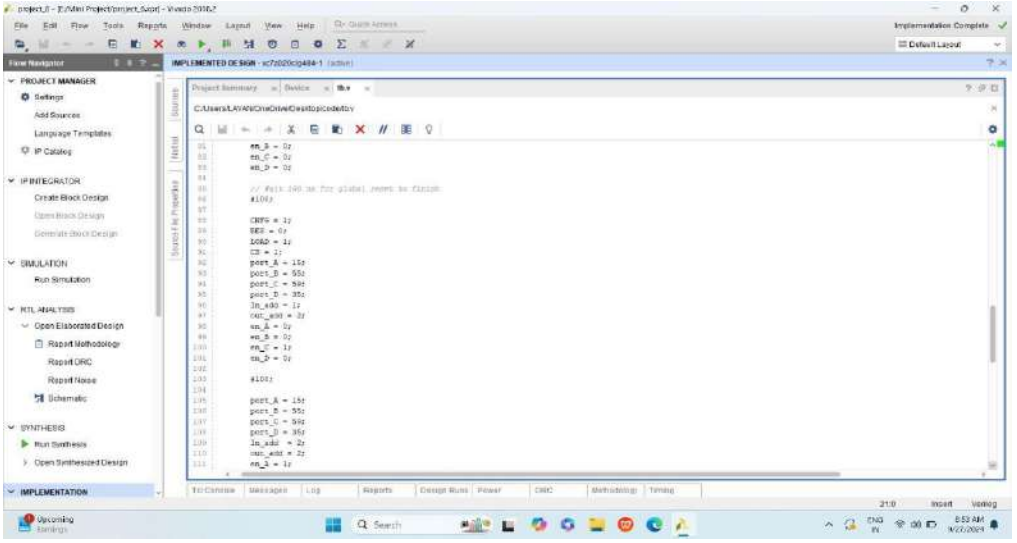


Fig 6.7.2 code

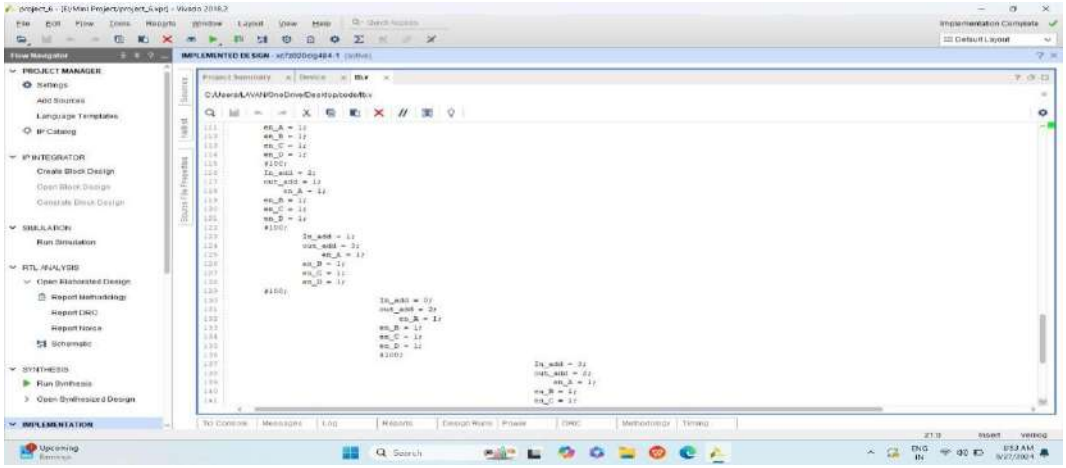


Fig 6.7.3

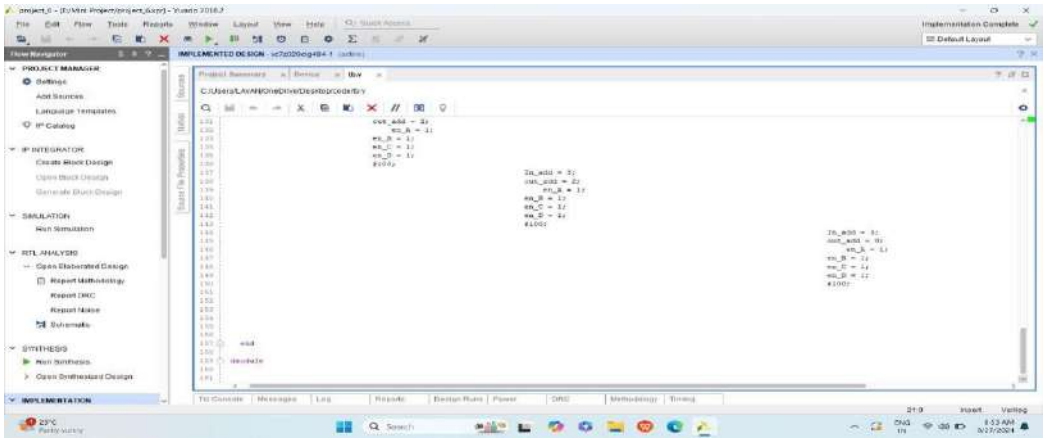


Fig 6.7.4

7-SIMULATION AND SYNTHESIS REPORT

RTL And Technology Schematics

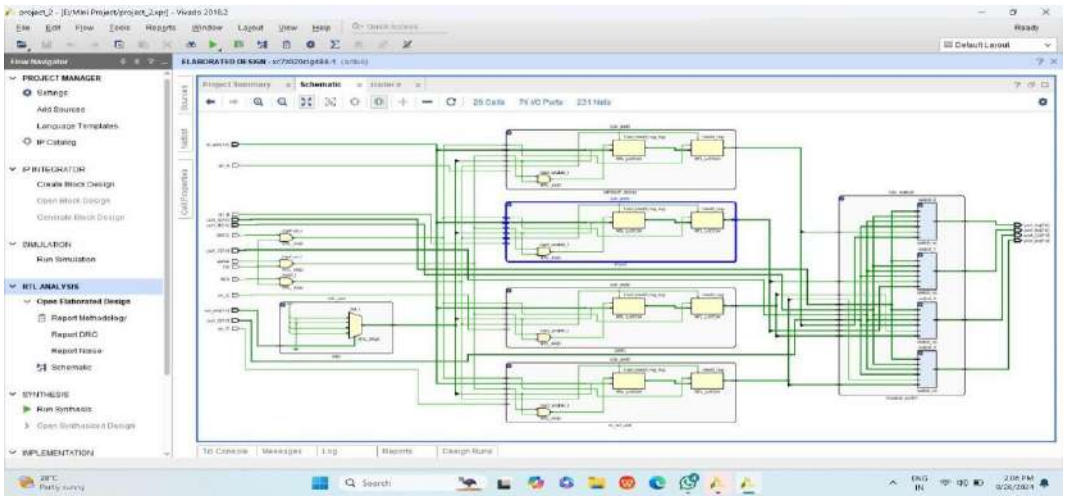


Fig 7.1: RTL (ASnapshot from Vivado Software) RTL: Register Transfer Level

These are the RTL Diagram from the open elaborated design

This image shows a schematic design in the Vivado tool for FPGA development. It displays multiple blocks connected via signal lines, with some modules like multiplexers (MUX) and logical gates (AND/OR) clearly visible. The green and black lines represent the interconnections or nets between different blocks. The left panel shows the project structure, with options for design elaboration, simulation, and synthesis steps.

8-CONCLUSION

An advanced FIFO structure based NoC is simulated and synthesized in Xilinx 14.7 ISE and implemented Vertex-6 FPGA device to analyze the performance in terms of occupied area, latency, power consumption and throughput. Single router is designed initially and then designed mesh based NoC to realize the memory utilization of FPGA. Fig.4 indicates that Register Transfer Level (RTL) schematic of single NoC router which is composed with input and output ports, arbiter, crossbar and channel control modules. The figure also describes the utilizations in terms of memory units each component individually. Each module of NoC designed using Verilog Hardware Description Language (HDL) separately and integrated as one module. An advanced queued buffer is. Virtual channels are created between routers when data flit is block in case of physical channel is not available therefore data packet latency is reduced as well as deadlock error avoided. The implementation results are improved in terms of resource utilization when compared with existing work. In future, NoC based processors are used at Artificial Intelligence applications. The performance NoC is needed to be improved by advancing router components because the power consumption increased through virtual channels at advanced FIFO structure.

Many future work directions are inspired by this paper including exploiting the mathematical properties of the code space to find additional nonorthogonal codes and boost the CDMA interconnect capacity and exploring more architectural optimizations of the OCI crossbar. Studying the robustness of CDMA interconnects and its enhancement techniques will be one of the prior future research points. Moreover, we plan to investigate using the OCI-based routers in different network topologies, evaluate their performance using standard benchmarks, and study their suitability for various applications.

References

[**1**] Q. Xu, T. Mytkowicz, and N. S. Kim, “Approximate computing: A survey,” IEEE Des. Test., vol. 33, no. 1, pp. 8–22, Feb. 2016.

[**2**] A. G. M. Strollo, D. De Caro, E. Napoli, N. Petra and G. Di Meo, "Low-Power Approximate Multiplier with Error Recovery using a New Approximate 4-2 Compressor," 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 2020, pp. 1-4, doi: 10.1109/ISCAS45731.2020.9180767.

[3] Y. Zhao, T. Li, F. Dong, Q. Wang, W. He and J. Jiang, "A New Approximate Multiplier Design for Digital Signal Processing," 2019 IEEE 13th International Conference on ASIC (ASICON), Chongqing, China, 2019, pp. 1-4, doi: 10.1109/ASICON47005.2019.8983437.

[4] Z. Yang, J. Han, and F. Lombardi, “Approximate compressors for error-resilient multiplier design,” in Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFTS), Oct. 2015, pp. 183–186.

[5] A. Kumar, R. K. Chintakunta, S. Kumar, K. Jamal and S. E. Ahmed, "Approximate Multiplier Architectures for Error Resilient Applications,"

2021 IEEE International Symposium on Smart Electronic Systems (iSES), Jaipur, India, 2021, pp. 89-92, doi: 10.1109/iSES52644.2021.00031.

[6] C. Jyothi, K. Gayathri, S. Karunamurthi, S. Veeramachaneni and N. M. S, "Area Efficient Approximate 4–2 Compressor for Multiplier Design," 2020 IEEE India Council International Subsections Conference (INDISCON), Visakhapatnam, India, 2020, pp. 231-235, doi: 10.1109/INDISCON50162.2020.00055.

[7] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra and G. D. Meo, "Comparison and Extension of Approximate 4-2 Compressors for Low-Power Approximate Multipliers," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 9, pp. 3021-3034,
Sept. 2020

[8] G. Park, J. Kung and Y. Lee, "Design and Analysis of Approximate Compressors for Balanced Error Accumulation in MAC Operator," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 7, pp. 2950-2961, July 2021

[9] C.-H. Lin and I.-C. Lin, “High accuracy approximate multiplier with error correction,” in Proc. 31st ICCD Conf., Oct. 2013, pp. 33–38.

[10] M. Ha and S. Lee, “Multipliers with approximate 4–2 compressors and error recovery modules,” IEEE Embedded Syst. Lett., vol. 10, no. 1, pp. 6–9, Mar. 2018.

[11] D. R. Gandhi and N. N. Shah, "Comparative analysis for hardware circuit architecture of Wallace tree multiplier," 2013 International Conference on Intelligent Systems and Signal Processing (ISSP), Vallabh Vidyanagar, India, 2013, pp. 1-6, doi: 10.1109/ISSP.2013.6526864