Optimized Reconfigurable Network Topology in NoC

S.Anitha, B.Hema,M.Meena, R.Muthulakshmi

Anna University – University College of Engineering Dindigul, Tamilnadu India

Email: anishasakthi21@gmail.com

Abstract: The network-on-chip (NoC) technology is becoming essential for interconnecting the cores in multicore processor. While NoCs result in noticeable performance boost over conventional bus systems, they consume a non-negligible fraction of the system power. In this paper, we proposed reconfigurable architecture for networks-on-chip (NoC) which dynamically changing inter-router connections according to traffic pattern. In the topology design minimizing the power consumption and hop count plays an important role. A NoC router is composed of input ports, output ports, and a switching matrix (to connect the input ports to output ports). The performance metrix such as power consumption is calculated for TDM and ALU applications. The router architecture is modeled using ISim simulator and we calculate the power using xilinx power estimator. The results shown that when compared with the existing system our proposed system gives reduction in power and also better performance.

Keywords: Multicore, network on Chip, xilinx power estimator, performance evaluator.

1. INTRODUCTION

Due to advances in circuit technology and performance limitation in wide- issue in Chip-Multiprocessors (CMP) or muticore technology has become the mainstream in CPU designs. A multicore processor is a single computing component with two or more independent cores. Multicore processors are widely used across many application domains including general-purpose, embedded. network, digital signal processing(DSP), and graphics.

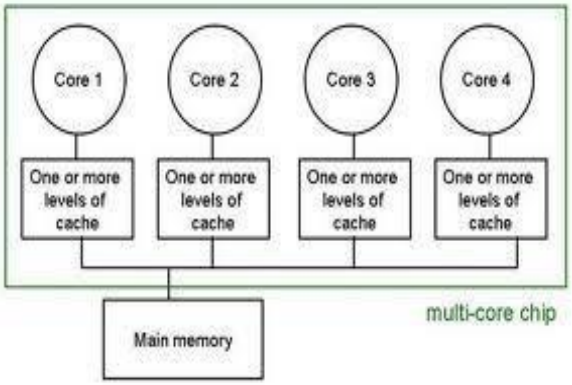


Fig.1.1. Multicore chip

Multicore processors are the new direction to manufacturers for achieving high performance on complex system. Using multiple cores on a single chip is advantageous in raw processing power, but nothing comes for free. With additional cores, power consumption and heat dissi-pation become a concern and must be simulated before lay-out to determine the best floorplan which distributes heat across the chip, while being careful not to form any hot spots. Before multicore

processors the performance increase from generation to generation was easy to see, an increase in fre-quency.This model broke when the high frequencies caused processors to run at speeds that caused increased power consumption and heat dissipation at detrimental levels.

The following isn't specific to any one multicore design, but rather is a basic overview of multi-core architecture. Although manufacturer designs differ from one another, multicore architec-tures need to adhere to

certain aspects. The basic configuration of a microprocessor is seen in following fig..

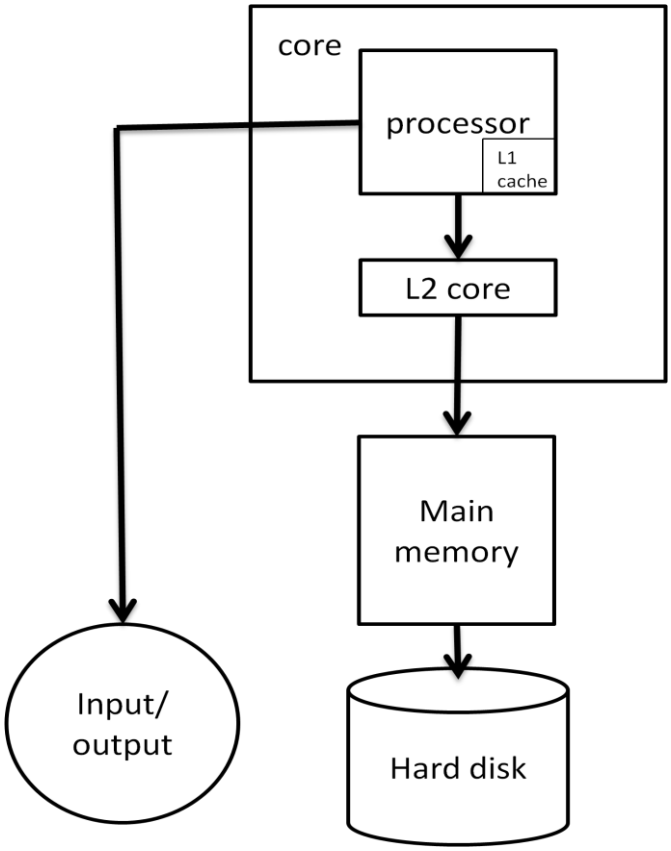


Fig.1.2 Generic Modern Processor Configuration

2. LITERATURE SURVEY

In [1] S. Bell et al... proposed, the TDN can be used directly by the processor or the DMA engine to access another tile's L2 cache. This mechanism allows tiles' L2 caches to be aggregated as a distributed L3 cache. The UDN and IDN networks are directly accessible by the processor ALU by mapping the networks within the register space. The UDN is used for user-level program communication. The IDN is used for communication with I/O interfaces. The OS also uses the IDN for inter-tile OS communication. IDN/UDN traffic is message- based and variable-length. Message-based communication introduces the risk of head-of-line blocking. Algorithms may require messages to be processed in an order different from the order received. To support this, messages are tagged to allow the receiver tile to sort messages into multiple queues. Messages on the IDN and UDN networks are tagged to specify in which one of seven destination queues they should be placed. Two additional catch-all queues handle any non-matching tags. A shared single port SRAM stores the message words from the two software- visible networks to allow message reordering. Small per flow queues provide a direct connection to the ALU. To reduce latency, the RAM is bypassed if the per- flow queues are not full. RAM allocation is controlled via a free-list, and head and tail pointers. The balance of RAM utilization between IDN and UDN is software controllable by defining the percent allocation allowed per network. Oversubscription by both networks allows the RAM to allocate on a first-come first-served basis. The message words are stored in linked lists with the head and tail pointers in the control logic and the RAM storing the next pointers along with the data.

In [2] M. B. Taylor et al...proposed the replicated tile design saved us considerable time in all phases of the project: design, RTL Verilog coding, resynthesis, verification, placement, and back-end flow. Our design

supports the glueless connection of up to 64 Raw chips in any rectangular mesh pattern, creating virtual Raw systems with up to 1,024 tiles. We intend to use this ability to investigate Raw processors with hundreds of tiles. We think that reaching the point at which a Raw tile is a relatively small portion of total computation could change the way we compute. We can imagine computation becoming inexpensive enough to dedicate entire tiles to prefetching, gathering profile data from neighbor tiles, translating (say for x86 emulation) and dynamically optimizing instructions, or even to simulating traditional hardware structures like video Ramdacs. The idea of creating architectural analogs to pins, gates, and wires will ultimately lead to a class of chips that can address a great range of applications. It takes a lot of imagination to envision a 128-tile Raw processor, how fast a full-custom version would clock, or how a more sophisticated compute processor design could affect the overall system. It is our hope that the Raw research will provide insight for architects who are looking for new ways to build processors that leverage the vast resources and mitigate the considerable wire delays that loom on the horizon.

In [3] M. Meeuwsen et al...proposed **,** the challenge of globally synchronous systems, Design difficulty due to high clock frequencies, long clock wires, and large circuit parameter variations, High clock power consumption and lack of flexibility to independently control clock frequencies. The GALS clocking style addresses these challenges. Global wires are a concern, Their length doesn't shrink with technology scaling, assuming the chip size remains the same, AsAP uses nearest neighbor communication. AsAP's key features, Chip multiprocessor

reduced complexity processors and interconnect 0.66 mm2 per processor in 0.18 µm. Fully independent clocking per processor. High performance and energy efficient, 475 MHz standard cell implementation in 0.18 µm, 32 mW average application power at 1.8 V, 2.4 mW average application power at 0.9 V, 116 MHz. Complex multi-task applications. C compiler and auto-mapping tool. . Each processor contains two dual-clock FIFOs, Rd/Wr in separate clock domains ,Gray coded Rd/Wr address across clock domains.

3. ROUTER ARCHITECTURE DESIGN

A NoC router is composed of a number of input ports (connected to shared NoC channels), a number of output ports (connected to possibly other shared channels), a switching matrix connecting the input ports to the output ports, and a local port to access the IP core connected to this router. Herein, we use the terms router and switch as synonymous, but the term switch can also mean the internal switch matrix that actually connects the router inputs to its outputs .In addition to this physical connection infrastructure, the router also contains a logic block that implements the flow control policies (routing, arbiter, etc.) and defines the overall strategy for moving data though the NoC.

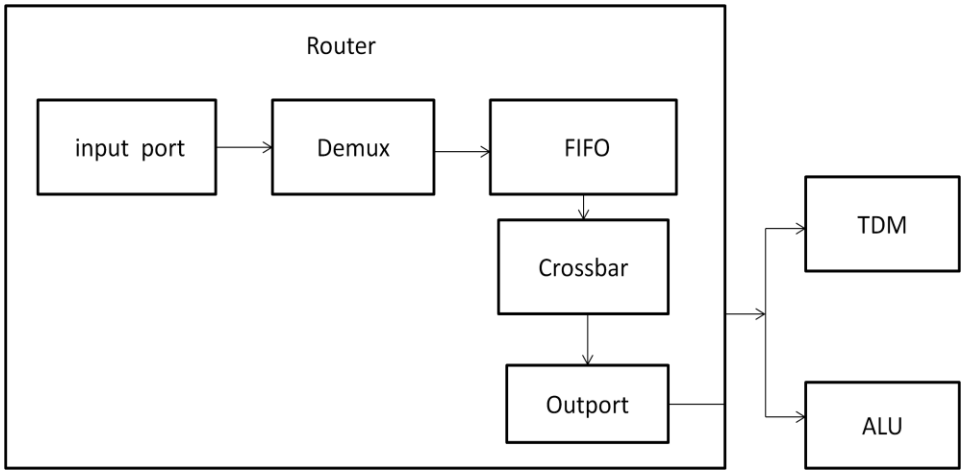
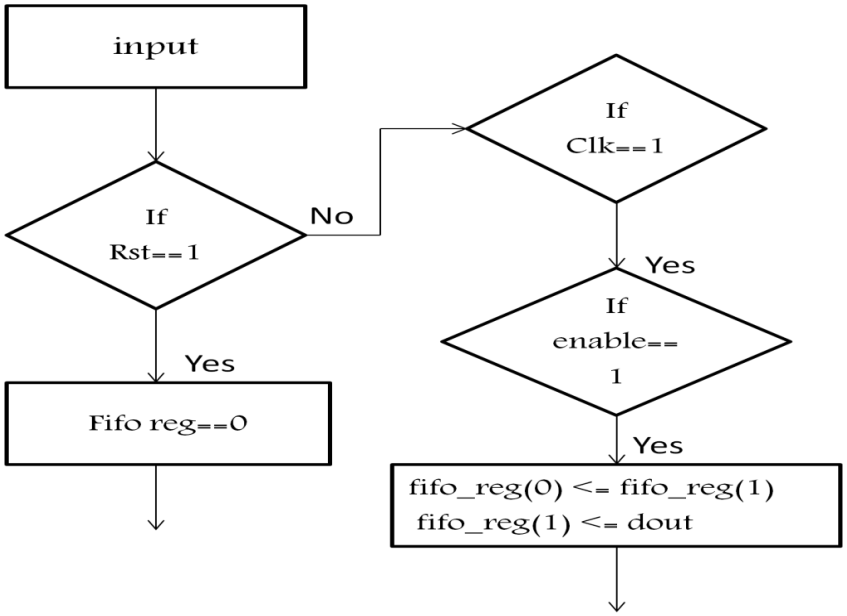


Fig.3.1. Router design

4. SYSTEM DESIGN

4.1. **TDM design**



Time-Division Multiplexing (TDM) is a convenient method for combining various digital signals onto a single transmission media such as wires, fiber optics or even radio. These signals may be interleaved at the bit, byte, or some other level. The resulting pattern may be transmitted directly, as in digital carrier systems, or passed through a modem to allow the data to pass over an analog network. Digital data is generally organized into frames for transmission and individual users assigned a time slot, during which frames may be sent. If a user requires a higher data rate than that provided by a single channel, multiple time slots can be assigned.Time-division multiplexing (TDM) is a type of digital or (rarely) analog multiplexing in which two or more signals or bit streams are transferred apparently simultaneously as sub-channels in one communication channel, but are physically taking turns on the channel. The time domain is divided into several recurrent timeslots of fixed length, one for each sub-channel. A sample byte or data block of sub-channel 1 is transmitted during timeslot 1, sub-channel 2 during timeslot 2, etc. One TDM frame consists of one timeslot per sub-channel plus a synchronization channel and sometimes error correction channel before the synchronization. After the last sub-channel, error correction, and synchronization, the cycle starts all over again with a new frame, starting with the second sample, byte or data block from sub-channel 1, etc. TDM is all about cost: fewer wires and simpler receivers are used to transmit data from multiple sources to multiple destinations. TDM also uses less bandwidth than Frequency-Division Multiplexing (FDM) signals, unless the bitrate is increased, which will subsequently increase the

necessary bandwidth of the transmission.The TDM is composed of mainly multiplexer and demultiplexer. The TDM is designed with sixteen bit multiplexer and demultiplexer. The sixteen bit input l(15:0) is apply to the multiplexer.

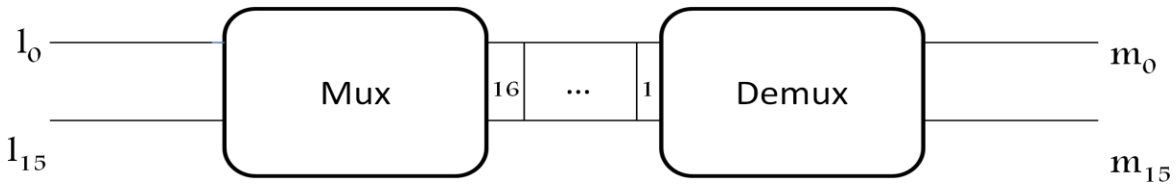


Fig. 4.1. TDM Block Diagram

The output of the demultiplexer is m(15:0) which is generated after the internal process. The internal process is explained in a following flow chart.

Fig.4.2. TDM Implementation

First of all it checks rst input if rst is 1 then the FIFO block is set to zero. If the rst input is 0 then it checks the clk and enable input if both are then it shift the content of the register after the four clk pulses are applied the output m(15:0) is generated.

Advantages of TDM

- Full available cahannel bandwidth can be utilized for each channel.
- Intermodulation distortion is absent
- TDM circuitary is not very complex
- The problem of crosstalk is not severe

4.2. ALU design

An arithmetic logic unit, or ALU is a combinational network that implements a function of its inputs based on either logic or arithmetic operations. An ALU must process numbers using the same formats as the rest of the digital circuit. The format of modern processors is almost always the two's complement binary number representation. Early computers used a wide variety of number systems, including ones' complement, two's complement, sign- magnitude format, and even true decimal systems, with various representation of the digits.The ones' complement and two's complement number systems allow for subtraction to be accomplished by adding the negative of a number in a very simple way which negates the need for specialized circuits to do subtraction; however, calculating the negative in two's complement requires adding a one to the low order bit and propagating the carry. An alternative way to do two's complement subtraction of A−B is to present a one to the carry input of the adder and use ¬B rather than B as the second input. The arithmetic, logic and shift circuits introduced in previous sections can be combined into one ALU with common selection. The inputs to the ALU are the data to be operated on (called operands) and a code from the control unit indicating which operation to perform. Its output is the result of the computation

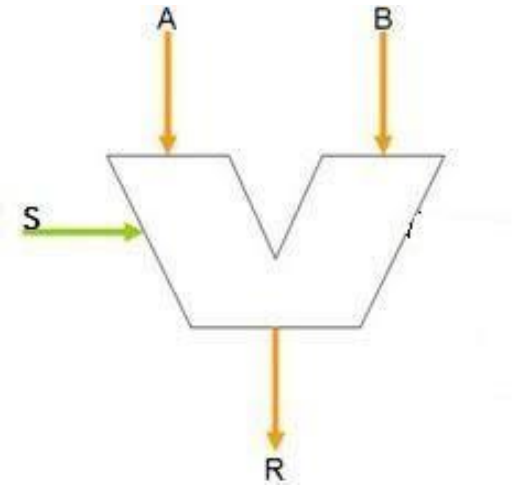


Fig.4.3. ALU Design

An *n*-bit ALU typically has two input words, denoted by $A = An - 1, , A0$ and $B = Bn - 1, , B0$. The output word is denoted by $R = Rn, Rn - 1, , R0$, where the high-order output bit, *R*n, is actually the carry-out. In

addition, there is a carry-in input *F*0.Besides data inputs and outputs, an ALU must have control inputs to specify the operations to be performed. In addition, there are operation selection inputs, *S*i, which determine the particular logic or arithmetic function to be performed. When *S*= 00, the operation is addition ; when *s* = 01, subtraction operation is indicated; when S=10 the operation is NOT logic function;for others the operation is NAND.

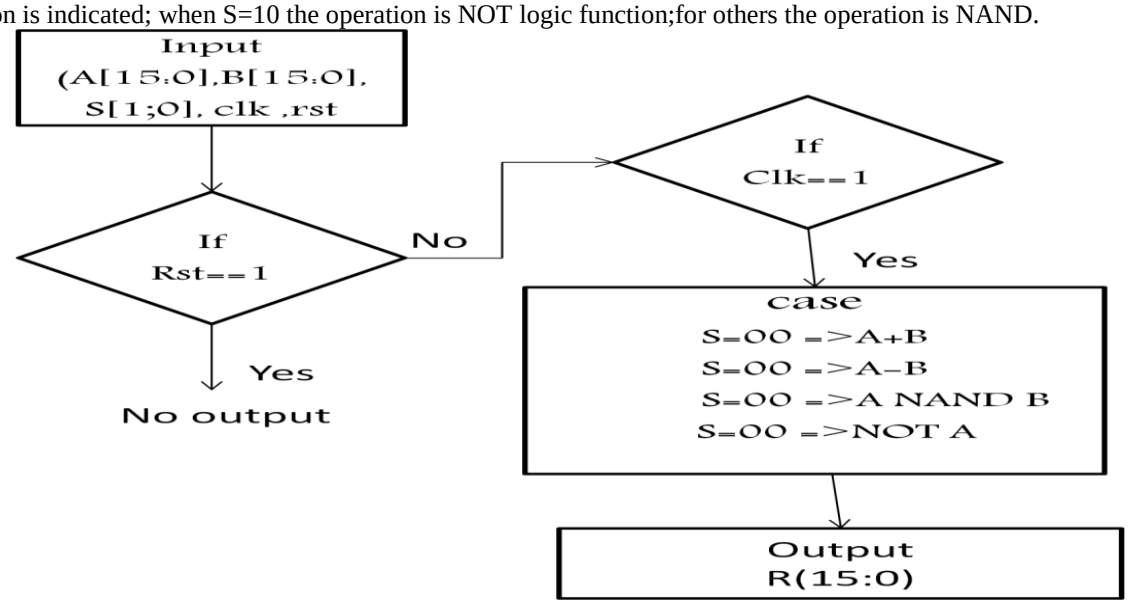


Fig.4.4. ALU implementation

The sixteen bit inputs A,B and the two bit S(selection line) are given to the ALU according to the clk and rst signal status then it checks the selection line based on this value the ouput R[15:0] is generated.

5. EXPERIMENTAL ENVIRONMENT

5.1. Software description:

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 ISE[1] (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.

5.2. XILINX Power Estimator Tool

The Xilinx Power Estimator (XPE) spreadsheet is a power estimation tool typically used in the pre-design and pre-implementation phases of a project. XPE assists with architectureevaluation, device selection, appropriate power supply components, and thermal management components specific for the application. XPE requires a licensed- version of Microsoft Excel 2003 or Microsoft Excel 2007 to beinstalled.Microsoft Excel 2010 is not officially supported in this release of XPE. The accuracy of XPE is dependent on two primary sets of inputs:

- Device utilization, component configuration, clock, enable, and toggle rates, and other information enter into the tool
- Device data models integrated into the tool

Power estimation for programmable devices like FPGAs is a complex process, since it is highly dependent on the amount of logic in the design and the configuration of that logic.To produce accurate estimates,

the power estimation process requires accurate input values, such as resource utilization, clock rates, and toggle rates.

The XPE Toolbar

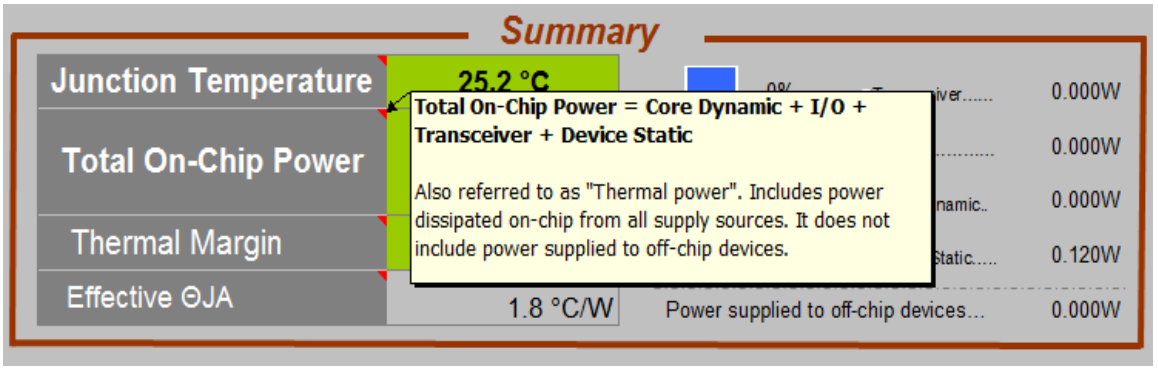


Fig.5.1. **Comment Indicators**

Junction Temperature

Estimated junction temperature as the design operates. Each device operates within a temperature grade specified in the datasheet. The background for this cell turns orange when the value is outside the operating range (timing may be affected) and turns redwhen outside the absolute maximum temperature (device damage possible). Thebackground color turns light blue when the value is set by user.

6. RESULTS AND DISCUSSION

The following screenshot represents the simulation result of our proposed system.

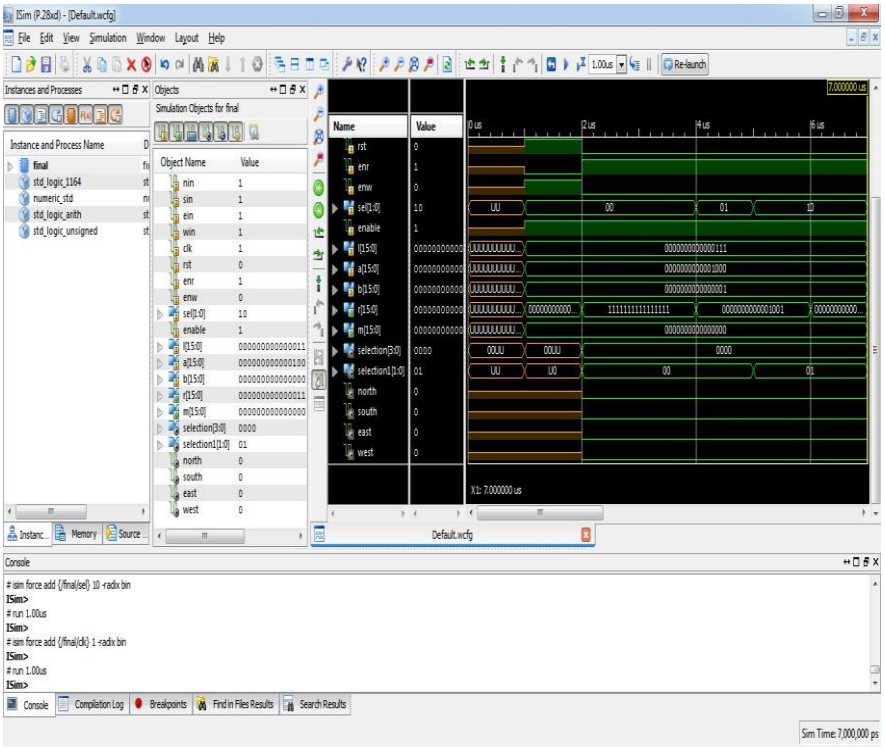


FIG.6.1. Output Screen Shot

In this, Four inputs(nin,sin,ein,win) with the control signals are given to the router. Based on the clock variation we get the output. From the router output the two single bit input is given to TDM and ALU. Then 16 bits of single input is given to TDM and 16 bits of two inputs are given to the ALU. Based on the selection lines and clock variation TDM produces output and ALU performs the operation.

6.1.Performance analysis

In this we compare the existing and proposed power levels. Finally we compare the performance of our proposed system. The following graphs are representing various power comparision from power estimator. The FIG.9 represents the comparision of total power and static power. From this comparision Our proposed method gives 0.091W power consumption which is less when compared with the existing system.

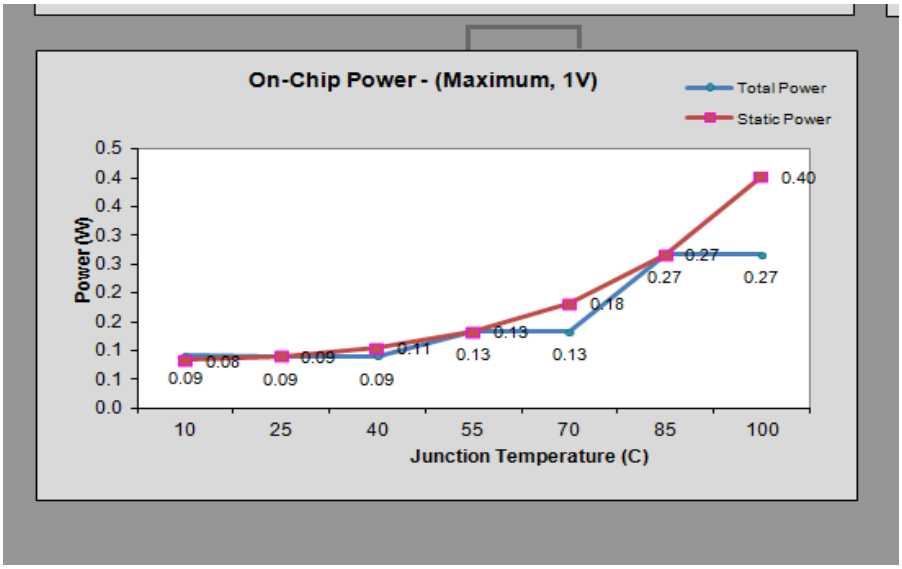


Fig.6.2. On Chip Power (Maximum,1V)

7. CONCLUSION

We proposed a reconfigurable router architecture for networks-on-chip (NoC) with the combination of optimization algorithm. Our overall design flow consists of two phases namely, router designing, and the estimation of power consumption. In the first phase, the design of router is done by the fuzzy logic algorithm. And the output of the router is used to the applications such as TDM and ALU. In second phase the power consumption for the design router is estimated. The results shows that the resultant power for our router design reduces the power consumption and gives better performance for NoC architectures.

REFERENCES

[1] S. Bell, B. Edwards, J. Amann, R. Conlin, K. Joyce, V. Leung, J. MacKay, M. Reif, L. Bao, J. Brown, M. Mattina, C.-C. Miao, C. Ramey, D. Wentzlaff,

[2] W. Anderson, E. Berger, N. Fairbanks, D. Khan, F. Montenegro, J. Stickney, and J. Zook, "Tile64 - processor: A 64-coresoc with mesh interconnect," in Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International, Feb. 2008, pp. 88–598.

[3] M. B. Taylor, J. Kim, J. Miller, D. Wentzlaff, F. Ghodrat, B. Greenwald,H. Hoffman, P. Johnson, J.-W. Lee, W. Lee, A. Ma, A. Saraf, M. Seneski, N. Shnidman, V. Strumpen, M. Frank, S. Amarasinghe, and A. Agarwal, "The raw microprocessor: a computational fabric for software circuits and general- purpose programs," vol. 22, no. 2, pp. 25–35, Mar. 2002.

[4] Z. Yu, M. Meeuwsen, R. Apperson, O. Sattari, M. Lai, J. Webb, E. Work, D. Truong, T. Mohsenin, and B. Baas, "Asap: An asynchronous array of simple processors," IEEE Journal of Solid State Circuits, vol. 43, no. 3, p. 695, 2008.

[5] S. Mukherjee, P. Bannon, S. Lang, A. Spink, and D. Webb, "The alpha 21364 network architecture," IEEE micro, pp. 26–35, 2002.

[6] A. B. Kahng, B. Li, L.-S. Peh, and K. Samadi, "Orion 2.0: A fast and accurate noc power and area model for early-stage design space exploration," in Proc. DATE '09. Design, Automation. Test in Europe Conference. Exhibition, Aspr. 20–24, 2009, pp. 423–428.

[7] A. K. Mishra, R. Das, S. Eachempati, R. Iyer, N. Vijaykrishnan, and C. R. Das, "A case for dynamic frequency tuning in on-chip networks," in Proceedings of the 42nd International Symposium on Microarchitecture, ser. MICRO 42. New York, NY, USA: ACM, 2009, pp. 292–303.

[8] A. K. Mishra, A. Yanamandra, R. Das, S. Eachempati, R. Iyer, N. Vijaykrishnan, and C. R. Das, "Raft: A router architecture with frequency tuning for on-chip networks," J. Parallel Distrib. Comput., vol. 71, pp. 625–640, May 2011.

[9] U. Y. Ogras and R. Marculescu, "Prediction-based flow control for network- on-chip traffic," in Proc. 43rd ACM/IEEE Design Automation Conference, 2006, pp. 839–844.

[10] G. Chen, F. Li, M. Kandemir, and M. Irwin, "Reducing noc energy consumption through compiler-directed channel voltage scaling," in Proceedings of the 2006 ACM SIGPLAN conference on Programming language design and implementation. ACM New York, NY, USA, 2006, pp. 193–203.

[11] C. Isci, G. Contreras, and M. Martonosi, "Live, runtime phase monitoring and prediction on real systems with application to dynamic power management," in MICRO 39: Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture. Washington, DC, USA: IEEE Computer Society, 2006, pp. 359–370.

[12] Y. S.-C. Huang, K. C.-K. Chou, C.-T. King, and S.-Y. Tseng, "Ntpt: On the end-to-end traffic prediction in the on-chip networks," in Proceedings of the 47th Design Automation Conference (DAC), 2010.

[13] F. Worm, P. Ienne, P. Thiran, and G. De Micheli, "An adaptive lowpower transmission scheme for on-chip networks," in Proceedings of the 15th international symposium on System Synthesis. ACM, 2002, p. 100.

[14] W. Kim, M. Gupta, G. Wei, and D. Brooks, "System level analysis of fast, per-core dvfs using on-chip switching regulators," in High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on. Ieee, 2008, pp. 123–134.

[15] Y. S.-C. Huang, K. C.-K. Chou, and C.-T. King, "Implementation details of atpt-based predictions for low power noc design," Tech. Rep., 2012.

[16] N. Muralimanohar, R. Balasubramonian, and N. Jouppi, "Optimizing nuca organizations and wiring alternatives for large caches with cacti 6.0," in Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture. IEEE Computer Society, 2007, pp. 3–14.