Analysis of various arbitration algorithms to reduce switching delay for NoC design

Cammillus. S[1], Shanmugavel. S[2], A. Justin Diraviam[3]
[1,2]*Department of Electronics and Communication Engineering,*
National Engineering College, Kovilpatti, Tamil Nadu, India.
[3]*Department of Computer Science Engineering,*
Rajas Institute of Technology, Nagercoil, Tamil Nadu, India.
[1]*cammillus@gmail.com*, [2]*sethusvel@gmail.com*,[3]*jusma@rediffmail.com*

Abstract

On recent growth of System-on-Chip (SoC), Network-on-Chip used to interconnect with inside gadgets and interfaces SoC with outside world. Arbiter is most vital block of Network - on – Chip, routes the incoming data to destination port based on the priority. Many high-speed networks used to provide services like IPTV need to routes the packets onto the output port with minimum latency. Low chip area, critical delay, and power consumption are the expectations of the arbitration algorithm for on-chip bus communication. Using this scheme, good performance is accomplished and the results are indicated. To meet the design necessities and the expected performance, diverse philosophies of arbitration are proposed. Round robin arbiter, Matrix arbiter, and Modified Index - based Round Robin (MIRR) arbiter mechanisms are analysed in this paper. The arbiter was executed on FPGA and incorporated utilising XILINX 13.2 version.

Keywords: *Network-on-chip, System-on-Chip, IPTV, Arbitration*

1. Introduction

As with the growth of CMOS innovations, it is conceivable to actualize roughly one billion transistors on a single chip. This headway in the small-scale gadgets expects the coordination of different parts of a processing framework or some other electronic framework on a single Integrated Circuit (IC) to implement a complete System on a Chip (SoC). SoC has programmable components such as processor cores or using explicit IP cores, On-chip memory, I/O gadgets. Increase demand in SoC (System on Chip) cores, Network-on-Chip (NoC) shown in figure 1 plays a vital role in building Scalable on-chip systems. Micro-architecture design of the NoC router [1] [2] has increased the significant impact on performance and efficiency.

Set top box, a house hold device which serves as the head end device of the electronic gadgets in home. Analog set top box that supports DTH services uses tuner circuit that switch between channels. The recent innovation in STB and the requirements of digital transmission like IPTV, the STB includes coders and private video recorder (PVR) for digital content storage. Merging multimedia content with core network and recent innovation in SoC [3], STB has re modified to face the growing needs of digital world.

Switching between channels becomes a great challenge and that induce various delay at user end. Hence ensuring good QoS at user terminal is alarming challenge in modern IPTV applications. Various delays [4] are listed in the table 1.

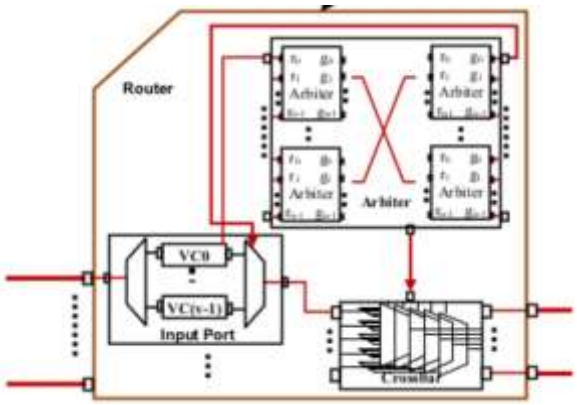

Fig 1. NoC Architecture

The advancement in VLSI technology and Silicon on Chip (SoC), various encoders are embedded in the SoC decodes the video and audio data as claimed by the user. Network-on-chip, the key component involved in networking the components of set top box with rest of world. Popular switching methods like round robin, Matrix-arbiter, Index-based round robin arbiter are used in the set top box that maps signals from input port to output port.

Table 1 Various delay during switching process in STB

Type of delay for digital transmission	Delay in ms
Channel request delay	5 to 10
IGMP leave/Join	100 to 200
Synchronizing delay	1 to 2
Video buffering delay	1 to 2
STB delay	150

The role of the arbiter is to map the necessary inputs with outputs (see fig. 4). Packets of different input port contend for the same output port. This determination is made by the arbiter, which chooses the output port for the requesting incoming signal at input port. The input contributions of the arbiter will be called from the neighboring router (for all directions) and the outputs of routing logic.

The NoC router [5] [6] comprises of control tables, cross bar switch and memories. Multiplexer, the key component in cross bar switch, maps the input to the requested output port. For 'n' channels, the i^{th} channel occupies the output port when a grant signal is triggered with a grant ID (grant_id). Without the grand ID the channel is blocked in accessing the output port. The grant ID is used to decode the grant signal. For a channel to gain the output port many arbitration techniques were proposed. Fixed priority arbitration (see fig. 2a), assuming the channel r_0 has high priority and the subsequent

channels have lower priority than r_0. The order of priority as assumed as $r_0>r_1>r_2>r_3>r_4>\ldots>r_n$ where n is the last channel contending for the output port. The priority of the channel is decided with the user interest. Assuming the channel r_0, been watched by the user, then channel r_1 gets the next highest priority for the user to view next.

Figure 2 depicts the channels are arranged in linear fashion. When the priorities of the channels are not linear to the order of the channels in the list then the channels are arranged with variable priority (see fig. 2b). If channel r_2 have highest priority, then multiplexer M2 is selected and if no channels are asserted with high priority, then multiplexer M0 will hold the output port (see fig. 3). The channel that has priority, outputs the data over the output port ensured by the corresponding multiplexer. If no channels gains priority, the output port holds the channel that is ongoing.

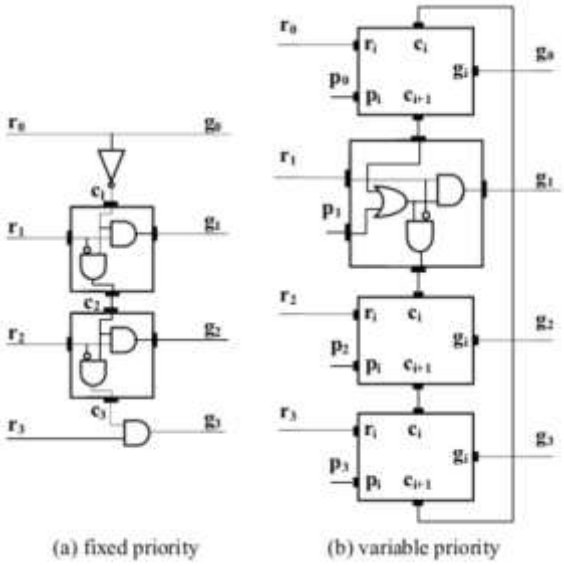

Fig. 2 Priority based arbitration

Based on the priority we set and analyzing the inputs we can uncontrive the data to the port and can reduce data congestion and data loss. If more than one demand arises for a port arbitration works to limit the port to one user otherwise data sprits occurs normally.

Arbiters [7] [8] can be classified as weak, strong or FIFO arbiters regarding fairness. For a weak fairness arbiter, resources are allocated on every demand. For strong fairness arbiter, the resources are granted equally as request originates. Resources are granted as first come first served basis for the request uses FIFO fairness arbitration method. Generally, arbitration is classified as fixed and variable priority architectures [9]. The priority of the request is linear for fixed architecture. A pearly iterative arbiter priority of the request can be reverted from cycle to cycle.

The round-robin architecture is easy to implement and starvation free. The structure of round robin arbiter expands as the demand at input port increases which result in huge chip area, higher power utilization, and hair-trigger path delay. Due to the architectural complexity, the hair-trigger path wait of arbiter dominates input-port and crossbar switch. The speed of arbitration relates the maximum frequency F_{max} of the NoC router.

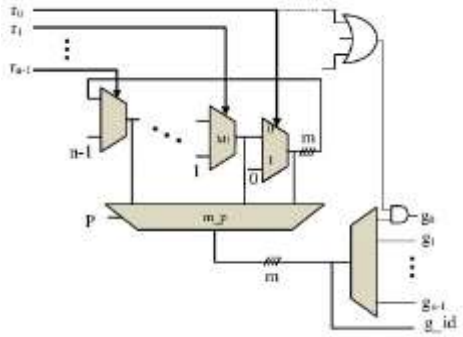

Fig. 3 Variable priority arbitration

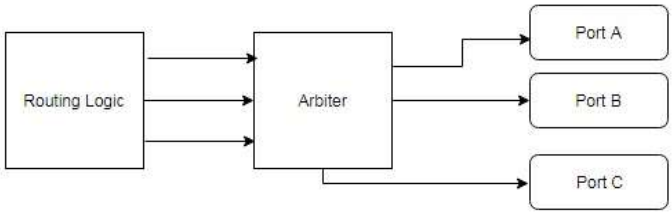

Fig. 4 Arbitration of ports

2. Related Works

The user choice of choosing a desired channel is made possible via set top box. As the user choose a desired channel, a group request to the desired channel and a group leave request to current viewing channel is sent simultaneously. As discussed above the delay due to IGMP leave/join will be 100 to 200 ms and the delay roughly changes on the availability of congestion in the core network.

The delay contributed by the set top box is around 150 ms. The switching delay between channels is the key factor of delay in set top box. Most popular switching methods that used in set top box are discussed below.

2.1. Round Robin Method

The most popular method deployed in conventional set top boxes. The channels are arranged in round fashion. The user can switch between channels as in order as the channels are arranged in the round fashion. Figure 5 shows a four port round robin arbiter. The current channel shifts to the last channel of the list as the user skips to the next channel. The channel has equal probability that it may be accessed. The switching delay increases as with the position of the desired channel in the list. The switching delay is reduced as if the desired channels are arranged adjacent to each other. If the user's choice increases the switching delay increases. If the desired channel is reduced to two or four, the switching delay between channels

will be better. For growing demands, it is highly challenge-able and the method is ruled off as the desired channel increases.

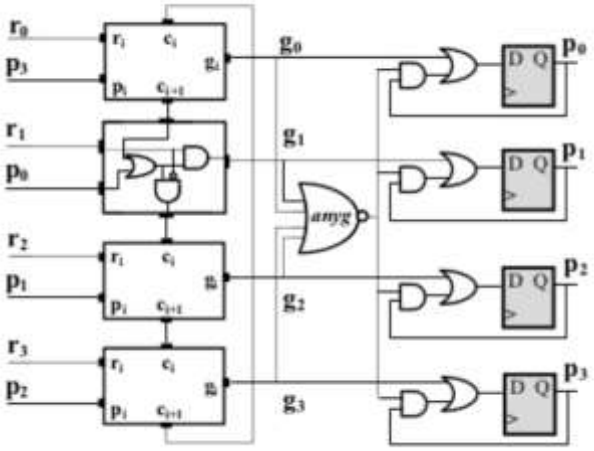

Fig. 5 Four port Round robin arbiter

2.2. Matrix Method

Matrix arbiter, a strong fairness arbiter with four input ports that works with the standard of last-recently-served scheme. A set of six flip-flops are deployed in triangular fashion as shown in figure 6. The lower flip-flops produce the complement version of the upper flip-flop block. Matrix arbiter utilizes a triangular array of p x n state bit which uses last recently served order priority. For the element p x n, request originated by m (row) takes the priority of recently served n (row).

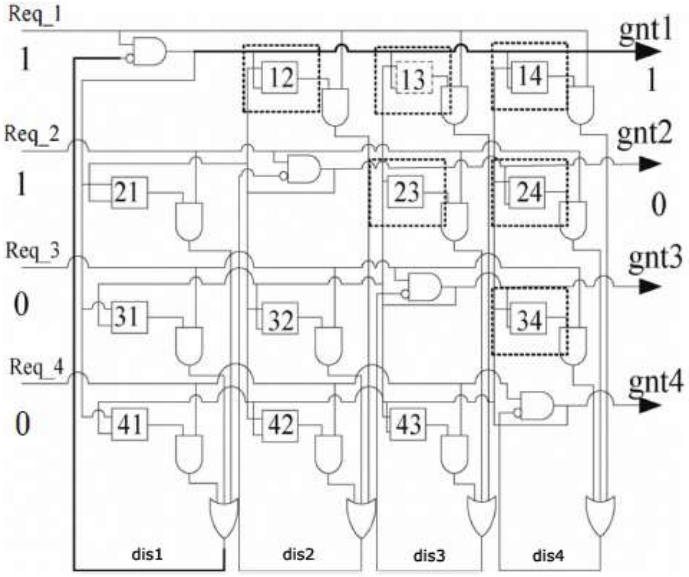

Fig. 6 Matrix-arbiter architecture

The method is more efficient if the channels are limited to 4 or 6. As the number of channels increases the design complexity increases. The switching delay decreases and design complexity increases for more than 6 channels. As a user chooses his desired channel, it get swaps with the current channel and the desired channel will occupy the output port. The previous watched channel will resume the previous position of the current channel.

2.3. Indexed Round Robin Arbiter

The limitation of round robin arbiter was addressed in this IRR arbiter [10]. The channels are given with variable priority and the channel will occupy the output port on higher priority. As the user wishes to skip to another desired channel, the current channel will lose its priority and get updated with the least priority. The IRR method (see figure 7) uses a sequence of multiplexer, de-multiplexer and flip flops as its architectural organization to perform the switching methods. The input port uses virtual channels [11] [12] that need to arbitrate to gain the output port.

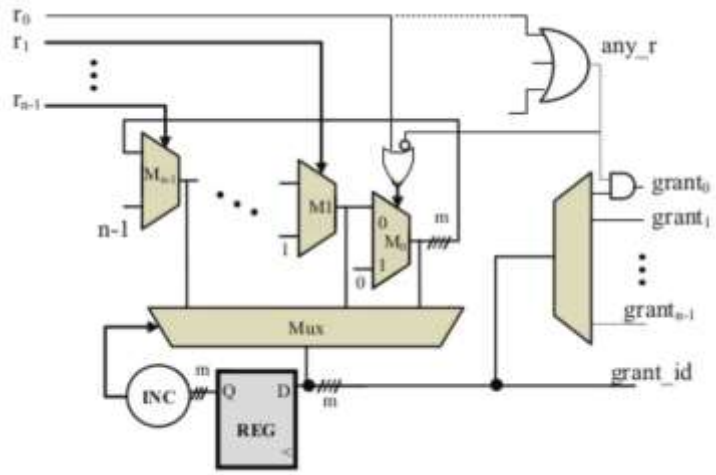

Fig. 7 Index-based round robin arbiter

To accommodate more number of channels, virtual channels are deployed at the input ports of the router. Based on the user interest, the specific channel assigned with high priority than other channels that wins the output port. Virtual channel enlarges the channels at input port [13] based on different combination of user interest. This would reduce the switching delay between channels.

2. Proposed Work

Another recent proposed arbiter model Index-based round robin arbiter (IRR) uses indexed input signals. The signal having high priority gains the output port. The input port uses virtual channel with buffer to accommodate the input signal. The architecture of IRR arbiter is shown in fig. 8. As with the IRR arbiter, the switching speed increases also, the like channels are brought nearby, though all the channels are arranged in a ring fashion. The grant_id that drives the de-multiplexer to generate the grant signal used to map the input port with the output port (see fig 7). The switching delay is still reduced with modified IRR arbitration.

For high speed NoC, the internal clock rate is high, the routers are synchronous locally and cross bar switching between input and output ports are asynchronous

globally. The data flits are stored temporarily onto the buffers which are synchronous with respect to router clock. The consumption of clock events is listed in table 2. The incoming signal is switched onto the output port is determined by RC, VA, SA. The role of different modules is briefed below.

Table 2 Clock events for various modules

Event	Clock event required	Module(s) involved
Router	2(squeezed) - Clubbed	
Arbitration	1(Parallel)	RC, VA, Decoder, output-arbiter

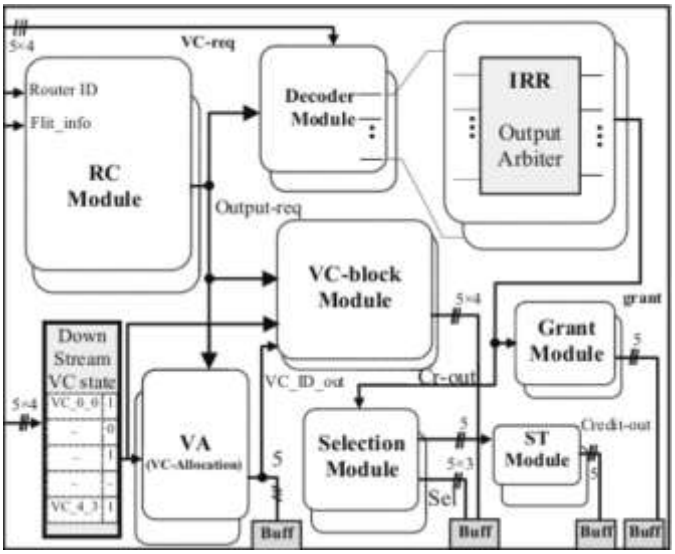

Fig. 8 Arbiter sub modules

RC module does the routing computation for the incoming flit on understanding the header flit. It pushes the packet to queue of VCA module. The Virtual channel allocation (VCA) involves dynamic queues that store the incoming flits. For the request originated by the flit, the arbiter issues with proper address to access the output of the cross bar switch. It provides input to the VC block module that issues grant signal for the incoming signal to cross over the output port.

Switch Allocation accepts 'n' inputs and arbitrates among the input signals and select r_i for service. The module issues grant signal g_i to the asserted signal. Switch Transversal helps to transverse between the cross bar switches. It issues credit_out signal to the outgoing signal to the downstream module. The decoder module holds the information about the winner VC that passes to the output port.

Selection module creates credit and selection address (Sel) for the cross bar multiplexer that drives the ST module. These signals are used to relate to the output port of the router. Grant signal that is generated by the Grant module makes the signal to move out the input port and furthermore over the cross bar switch to output port.

IRR arbiter gains the information about the VC winner and generate grant and grant_ID signal. The IRR arbiter involves IRR_read and IRR_write pointer. The

signal issued with grant_ID is read by the IRR_read pointer and maps to the output port, where the information is written into the output buffer by IRR_write pointer. De-multiplexer in the arbiter is used to generate grant signal. Critical path (i.e.) delayed path is experienced in the output of the de-multiplexer of the arbiter.

Hence MIRR uses a decoder that eliminates the critical path at the arbiter output. Fig 9 shows the MIRR arbiter that increases the switching process than IRR. The buffer holds the output that was triggered by the request line (any_r) and the decoder generates the grant signal. If the request line is not available the buffer holds the previous output. The switching speed increases than IRR. Critical path is eliminated and consume minimum power. The working process to map the input signal to output port is given below.

- The incoming flit with credit_in signal, saved in the buffer. IRR_write pointer points the flit VC_ID and the concern slot bit is set.
- The flit VC_ID is allotted with request signal that pointed by the IRR_read pointer is moved to the output buffer.
- RC module fixes the output port for the flit to transverse. The VA module verifies the status of its downstream VC module to identify the free virtual channel for the flit to transverse.
- Decoder and the output arbiter identifies the RC output, VC request signal and arbitrates to the output port.
- Selection module chooses the related address of the cross bar switch for the flit to transverse.
- SA module output triggers the Grant module to issue grant signal for the winner VC.
- If any flit looses the switch arbitration VC block issues block signal that blocks the flit to cross over its output port.
- Output arbiter issues Cr-out signal for the flit that wins its output port. ST module releases credit-out signal for the flit after two clock cycles that makes the flit to exit.
- Grant signal makes the flit to occupy the output port. Sel signal used to choose the specified output port asserted by the input signal.
- If the flit misses the credit-out signal, VC block issues a block signal that blocks the flit in accessing the cross bar switch.

4. Analytical comparison

Here we perform hardware analysis to compare the expected performance and hardware overhead of previously mentioned round robin arbiter with our proposed MIRR shown in figure 9. The subjective parameters of an arbiter circuit are speed, area and power consumption. The speed of an arbiter relay on the delay time or maximum clock frequency. (F_{max}). The clock frequency of an arbiter relies upon the longest delay (critical path) between two registers clocked simultaneously. Synopsys 90nm Digital Standard Cell Library is used to derive the electrical parameters of the logic gates. The power consumption depends on both static and dynamic powers. The critical path delay between two registers of each circuit is used to calculate the speed of the arbiter. MIRRA results in low chip area, delay and power consumption rather than the previous design.

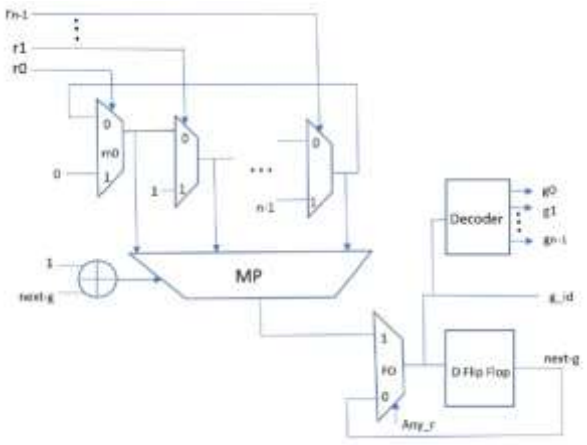

Fig 9. Modified Index Round robin Arbiter

5. Tools Used

Analysis of HDL design and synthesis, Xilinx ISE (Integrated Synthesis Environment) is used. It enables to perform timing analysis, synthesize designs, and examine RTL schematic diagrams.

.

6. Results and Discussions

Fig 10 shows the simulation output of the matrix arbiter. When the particular channel is specified, the arbiter makes the necessary steps and switches the requested channel to its output port.

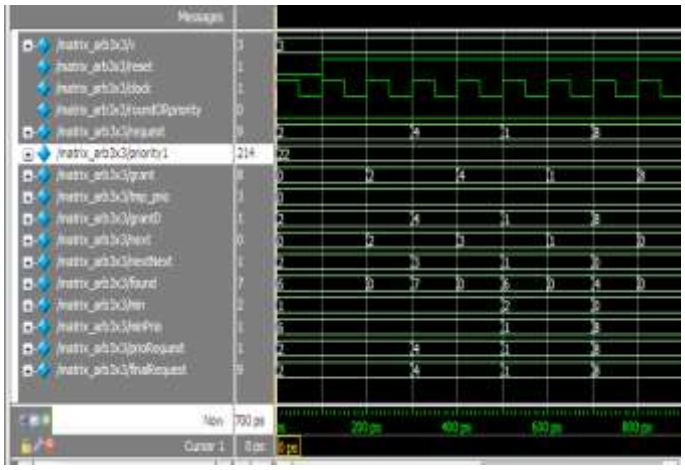

Fig 10 Simulation result for the Matrix Arbiter

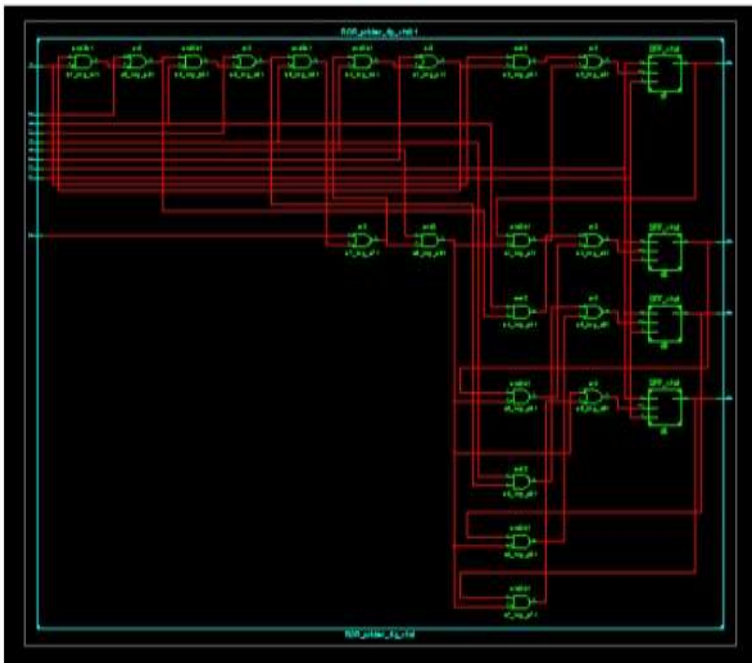

Fig.11 Screenshot of Modified Index round Robin Arbiter

The gate level diagram of Modified Index round robin arbiter is shown in figure 11. The comparison of parameters such as area and power is depicted in figure 12. The parameters show the largest resurgence compared to the round robin technique. The area and the power required by the matrix arbiter are less compared to that of the round robin arbiter.

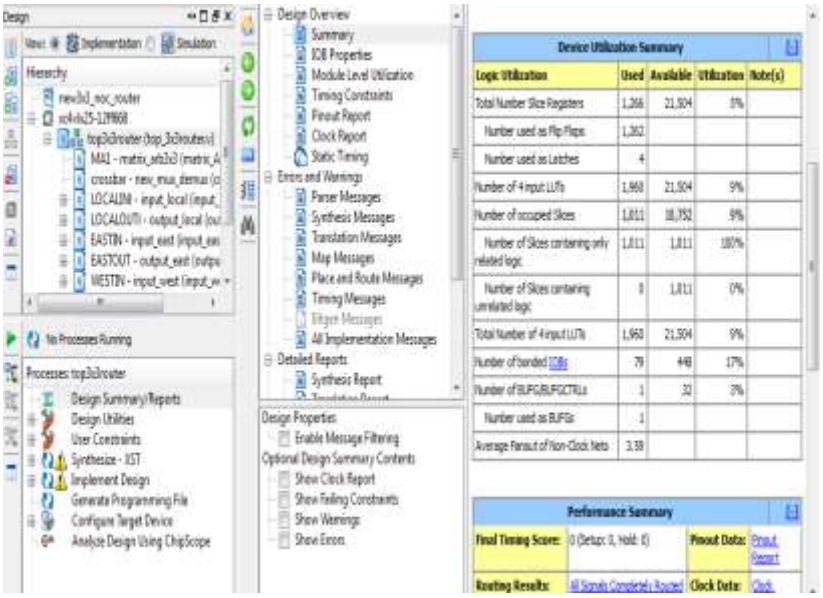

Fig. 12 Synthesis output for the various model

```
Device utilization summary:
---------------------------

Selected Device : 7a30tcsg324-3


Slice Logic Utilization:
 Number of Slice Registers:                4  out of  42000     0%
 Number of Slice LUTs:                     8  out of  21000     0%
    Number used as Logic:                  8  out of  21000     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:      12
   Number with an unused Flip Flop:        8  out of     12    66%
   Number with an unused LUT:              4  out of     12    33%
   Number of fully used LUT-FF pairs:      0  out of     12     0%
   Number of unique control sets:          1

IO Utilization:
 Number of IOs:                           14
 Number of bonded IOBs:                   14  out of    210     6%

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:                 1  out of     32     3%
```

Fig. 13 Synthesis output for MIRRA chip area

```
=========================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'
  Total number of paths / destination ports: 4 / 4
-------------------------------------------------------------------------
Offset:              0.645ns (Levels of Logic = 1)
  Source:            d1/q (FF)
  Destination:       op0 (PAD)
  Source Clock:      clk rising

  Data Path: d1/q to op0
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDR:C->Q             2   0.361   0.283  d1/q (d1/q)
     OBUF:I->O                0.000          op0_OBUF (op0)
    ----------------------------------------
    Total                     0.645ns (0.361ns logic, 0.283ns route)
                                      (56.0% logic, 44.0% route)

=========================================================================
```

Fig. 14 Synthesis output for MIRRA delay

The chip area, delay and power consumption of Modified Index round robin arbiter are illustrated in fig. 12 – 14. Table 3 and 4 depicts the comparison of various arbiters. Hence the proposed architecture explores good features compared to other arbitration and fast switching schemes that support for high-speed routers.

Table 3 Synthesis output for MIRRA power consumption

Supply Summary		Total Current (A)	Dynamic Current (A)	Quiescent Current (A)
Source	Voltage			
VCCint	1.200	0.008	0.000	0.008
VCCaux	2.500	0.008	0.000	0.008
Vcco25	2.500	0.002	0.000	0.002
Supply Power(W)		Total	Dynamic	Quiescent
		0.034	0.000	0.034

Table 4 Comparison of different arbitrations methods

Design	Delay(ns)	Area(µm2)	Power Consumption(W)
MIRR	0.645	35	0.034
Matrix	1.070	44	0.092
RoR	1.305	74	0.115

7. Conclusion and Future work

The MIRR arbitration provides fast switching compared to round-robin arbiter and other schemes. The chip area and the power consumption can be considerably reduced as the gates are replaced with less critical elements. The work provokes that the delay is greatly reduced and switching speed increases than the traditional STB that uses RoR arbitration. Rather more channels are allowed on deploying the virtual channels for the input port and hence the switching between alike channels are greatly reduced than the previous methods. The proposed method deploys more channels in its input port as it is restricted in the previous methods. The work enhances high speed communication that claimed for VoD application of IPTV.

References

[1] *Santhosh, G, "Design of Network on Chip with an Arbiter", International Journal of Innovative Research in Computer and Communication Engineering, **(2015)**, 3(9).*

[2] *Fu, Z., & Ling, X, "The design and implementation of arbiters for Network-on-chips", In Industrial and Information Systems (IIS), 2nd International Conference IEEE, **(2010)**, (Vol. 1, pp. 292-295).*

[3] *Gupta, J., & Goel, N, "Efficient bus arbitration protocol for SoC design", In Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), International Conference IEEE, **(2015)**, (pp. 396-400).*

[4] *Ramos, F. M, "Mitigating IPTV zapping delay" IEEE Communications Magazine, **(2013)**, 51(8), 128-133.*

[5] *Khanam, R., Sharma, H., & Gaur, S, "Design a low latency Arbiter for on chip Communication Architecture", In Computing, Communication & Automation (ICCCA), International Conference IEEE, **(2015)**, (pp. 1421-1426).*

[6] *Oveis-Gharan, M., & Khan, G. N, "Reconfigurable On-Chip Interconnection Networks for High Performance Embedded SoC Design", Journal of Systems Architecture, **(2020)**, 101711.*

[7] *Kamal, R., & Arostegui, J. M. M, "RTL implementation and analysis of fixed priority, round robin, and matrix arbiters for the NoC's routers" In Computing, Communication and Automation (ICCCA), International Conference IEEE, **(2016)**, (pp. 1454-1459).*

[8] *Mahalakshmi, S. S., & Cammillus, S, "A Survey on Arbitration Techniques for SoC Design", International Journal of Engineering Technology Science and Research(IJETSR), vol 4, no. 8, **(2017)**, e- ISSN 2394 – 3386.*

[9] *Helal, K. A., Attia, S., Ismail, T., & Mostafa, H, "Priority-select arbiter: An efficient round-robin arbiter", 13th International on New Circuits and Systems Conference (NEWCAS), IEEE, **(2015)**, (pp. 1-4).*

[10] *Oveis-Gharan, M., & Khan, G. N, "Index-based round-robin arbiter for NoC routers", In VLSI (ISVLSI), 2015 IEEE Computer Society Annual Symposium on IEEE **(2015)**, (pp. 62-67).*

[11] *Oveis-Gharan, M., & Khan, G. N, "Dynamic virtual channel and index-based arbitration based Network on Chip router architecture", In High Performance Computing & Simulation (HPCS), International Conference on IEEE, **(2016)**, (pp. 96-103).*

[12] *Oveis-Gharan, M., & Khan, G. N, "Efficient dynamic virtual channel organization and architecture for NoC systems" IEEE Transactions on Very Large Scale Integration (VLSI) Systems, **(2016)**, 24(2), 465-478.*

[13] *Avani, P., & Agrawal, S, "Efficient Dynamic Virtual Channel Architecture for NoC Systems", International Conference on Advances in Computing, Communications and Informatics (ICACCI) IEEE, **(2018)**, (pp. 2502-2507).*