Design and Development of Low Power Clock and Data Recovery Circuit for Asynchronous Network on Chips

T. Nagalaxmi[1], E. Sreenivasa Rao[2], P. Chandrasekhar[3]

[1]Research Scholar, Department of Electronics and Communication Engineering, Osmania University, Hyderabad, Telangana, India- 500007
[2]Professor, Department of Electronics and Communication Engineering, Vasavi College of Engineering, Hyderabad, Telangana, India-500089
[3]Professor, Department of Electronics and Communication Engineering, Osmania University, Hyderabad, Telangana, India- 500007
e-mail: tnagalaxmi@stanley.edu.in

***Abstract*-Today, the current buffered router Synchronous Network on Chip architecture consumes significant chip area and power. Therefore, based on biased routing, buffer-less routers have recently been predicted as a possible solution, but they suffer from contiguous port assignments, slow critical paths, and increased latency. Asynchronous Network on Chip architecture emerges as the best option for avoiding glitches and consuming less power. A clock and data recovery circuit is used to recover the clock signal from the router-generated data and reduce power consumption in a Multiprocessor System on Chip. This paper proposes a clock and data recovery circuit design for an asynchronous Network on Chip. The proposed 4x4 Mesh router architecture implemented in this paper can process 64-bit of data samples with a depth of 64. When comparing the proposed architecture with the existing NoC architecture, the proposed architecture has shown a power reduction of 5 times. The proposed architecture has consumed a total power of 0.103W. The proposed architecture was designed using Verilog-HDL and synthesized using Xilinx Vivado 2019.2 and implemented on the Zed board (Zynq evaluation and development Kit)**

***Index Terms*-Asynchronous Network on Chip, Clock and data recovery circuit, Metastability, Low power, Multiprocessor System on Chip.**

I. INTRODUCTION

With the continuous development of integrated circuit manufacturing technology, a single chip can integrate dozens or even hundreds of microprocessors. The complex communication between on-chip multi-cores makes the traditional bus-based on-chip communication structure a major performance bottleneck. In order to adapt to the growth of communication complexity, the network-on-chip has become the standard communication architecture of the current multi-core on-chip. According to the purpose of multi-core systems on a chip, they can be divided into two categories: multi-core microprocessor on-chip (chip multiprocessor, CMP) and multi-core system-on-chip (multiprocessor system-on-chip MPSoC) [1]. The former is to integrate many microprocessor units on a single chip to realize a multi-core multi-process processing system; while the latter is to realize one or more embedded systems on a chip by integrating many processing units that work together but with different functions on a single chip.

There are two implementations of Network on Chip (NoC): Synchronous NoC implemented by Synchronous circuits and Asynchronous Network on Chip implemented by asynchronous circuits [2]. Most current on-chip networks are Synchronous on-chip networks: communication between network nodes is driven by a single clock or multiple clocks. A few are asynchronous NoCs: communication between nodes is not driven by a clock but is controlled by a local handshake protocol due to the complexity of asynchronous circuit design and a lack of complete tool support. On average, the research on Asynchronous Network on Chip (NoC) is significantly behind that on Synchronous NoC. However, this does not diminish the importance of Asynchronous NoC has many advantages that Synchronous NoC does not have:

- The Asynchronous circuit has no clock tree and does not generate any dynamic power consumption when there is no actual data transmission.
- The device delay jitter caused by the uncertainty of the production process parameters is one of the main reasons for the decline of the current Synchronous integrated circuit yield rate. At the same time, changes in the circuit operating environment caused by low supply voltage, on-chip noise, and on-chip hot spots also make it difficult for the static timing analysis of the circuit. The handshake protocol used for the Asynchronous NoC is not sensitive to delay. At the same time, the asynchronous circuit does not need timing analysis to ensure the correctness of its function.
- The network interface of the Asynchronous on-chip network is unified. Each node can adjust its own clock frequency and supply voltage as needed.
- Since the Asynchronous circuit does not require static timing analysis, the integration

Digital Object Identifier 10.29292/v17i3.640

process of the on-chip network and network nodes is shortened.
- Asynchronous circuits have better electromagnetic compatibility and resistance to side-channel attacks.

The International Semiconductor Technology Development Roadmap estimates that by 2024, 49% of on-chip global signals will be delivered by asynchronous circuits.

International Semiconductor Technology Development Division-ITRS (The International Technology Roadmap for Semiconductors) predicts that the integration level of a single chip will reach billions of transistors, and the feature size will be reduced to 50mm; the clock frequency can reach 10GHz. The delay of the global signal may exceed the clock cycle; thus it increases the difficulty of a global synchronization design: the design of on-chip communication resources will replace computing resources as the main part of the design [4]. The crosstalk coupling, antenna effects, and transient errors caused by the scaling down of the process will be difficult to solve. NoC can solve the above problems to a certain extent and narrow the gap between design and production.

Aiming at the disadvantages of the high instantaneous power consumption of a synchronous clock network and the large communication delay of an asymmetric waterfall network and a symmetric waterfall network, this paper proposes and designs a Clock and Data Recovery (CDR) circuit for a NoC. The problems of communication delay and instantaneous power consumption are better solved. The rest of the paper is as follows: The section-2 describes the literature survey of asynchronous NoC and CDR circuits. The major contribution of the paper: low power CDR circuit for Asynchronous NoC is proposed in section-3. The result of the proposed architecture is discussed in section-4 and the conclusion is in section-5.

2. LITERATURE REVIEW

This section describes the various Asynchronous NoC architectures for MPSoC. The literature survey on Asynchronous NoC is carried out based on the structure and topology of the network, Synchronous and Asynchronous interface design, flow control methods, quality of services, routing algorithm, and low power design.

2.1. Structure and topology of the network

Asynchronous Network Interface Controllers (NICs) are generally globally asynchronous and locally synchronous (GALS) systems. In the asynchronous Network on Chip as shown in Fig. 1, the asynchronous circuit implementation portion is represented in a grey colour.

Each network node is a local system, which can have its own microcontroller, memory, bus, and functional unit, which is generally realized by a synchronous circuit. The NoC consists of an Asynchronous router and network interfaces within the network nodes. The network interface packages the data into packets and transmits them to the router or restores the received packets into data. At the same time, the network interface is also responsible for data transfer and format conversion between Synchronous circuits and Asynchronous networks.

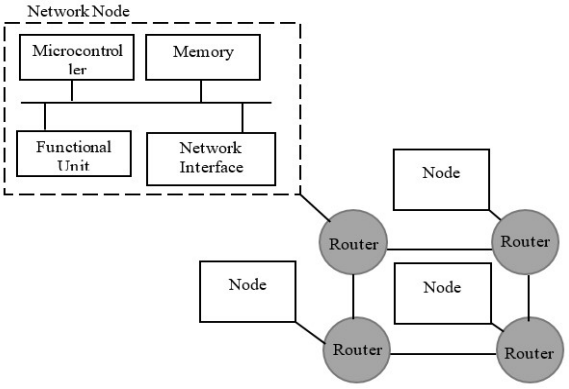


Fig. 1 Asynchronous Network on Chip system

In theory, all network topologies can be used for Asynchronous on-chip networks, but Asynchronous on-chip networks mostly use tree [5] and mesh [6], topologies. Tree topology is mostly used to connect network nodes with similar functions, similar to traditional multi-master buses, but with better scalability and throughput [7]. The use of tree topology benefits from the efficient arbitration circuit of asynchronous circuits. Compared with complex arbitration circuits in synchronous circuits, such as round-robin arbitration [8], asynchronous arbitration circuits use a large number of simple MUTEX gates, which are small in area and high in speed. In the Chain network, an arbitration circuit contains only one MUTEX gate and two RS cache units [9]. However, there is currently no efficient asynchronous distribution circuit. The multi-resource arbitration circuit [10] is the only asynchronous allocation circuit that fits the Speed-Independent (SI) delay assumption model, but its area and delay grow dramatically with the number of requests or resources. A large number of Wide Tree [11] and Clos [12] network structures for CMP and parallel computing have not yet been adopted by any asynchronous on-chip network.

The mesh topology is the most used topology for asynchronous on-chip networks [6, 13]. Compared with tree topology, mesh topology has two advantages: rich network resources and highly modular design. The abundance of network resources is reflected in two points: First, routers and connection channels are evenly distributed in the mesh topology, and their total amount of resources increases linearly with the number of network nodes. Second, there are multiple paths available between any node. Compared with the tree topology with only one path, the mesh topology has higher network redundancy [14] and can more easily guarantee communication service quality. The modular design of the mesh topology benefits from its direct network connection. Each network node is connected to a router with fixed structure and function.

Digital Object Identifier 10.29292/v17i3.640

Routers can be packaged as hard modules and used in different systems. In a tree network, the routers close to the root node have to bear larger data throughput rate and thus have a larger data bandwidth [14]. The entire communication network must be designed as whole and each router is individually configured according to the communication requirements of the nodes.

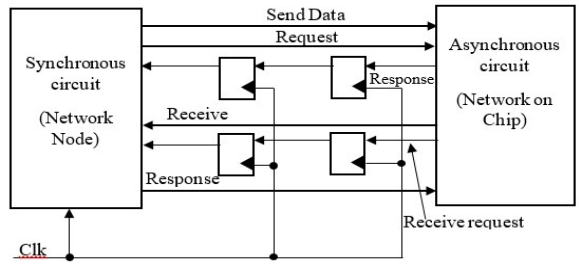
2.2. Synchronous and asynchronous interface design

Synchronous and asynchronous interfaces are part of a network interface. Nodes in a network-on-chip must communicate with the network-on-chip through a network interface. The network interface is responsible for packetizing the data sent by the network node and sending it to the network or disassembling the received a message to the network node. In the asynchronous network-on-chip, the network nodes are often synchronous circuits, so the network interface unit needs to provide a unified Synchronous and Asynchronous interface to realize the connection between the synchronous network node and the Asynchronous network. Commonly used Synchronous and Synchronous interface methods can be divided into three types:

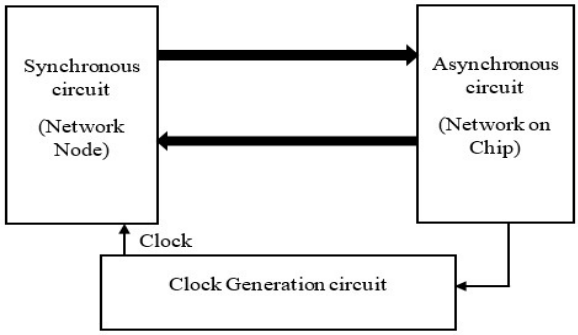
(1). Simple synchronizer interface: Two-flop synchronization circuits have been widely used in multi-clock synchronization circuits to reduce metastability errors. The same approach can be used for synchronous/asynchronous interfaces [15]. After using the synchronous circuit on all control signals from asynchronous to synchronous, the synchronous circuit can safely sample asynchronous data. This interface method is simple to implement and consumes less area and energy, but because it increases the delay on the control signal, the transmission delay is large and the throughput rate is limited. The simple synchronizer interface is shown in Fig. 2a.

(2). Stoppable clock interface: The asynchronous on-chip network divides the chip into local clock domains that are independent of each other, and network nodes can use independent clock domains. The stoppable clock interface introduces the data request signal of the asynchronous circuit into the local clock generation circuit. When asynchronous data acquisition and when the sample and the clock collide, the clock generation circuit suspends the clock to ensure correct data sampling. This approach completely avoids metastability errors and can complete data transfer in a single clock cycle with the lowest propagation delay [16]. With the dynamic frequency and voltage control, network nodes are using a stoppable clock interface have minimal energy consumption. However, the clock period of the local clock should be greater than the clock tree delay. The maximum clock frequency is limited by the clock tree delay. At the same time, the output clock of the local clock generation circuit is unstable and easy to affect by noise and the working environment, the clock accuracy is low. The stoppable clock interface is shown in Fig. 2b

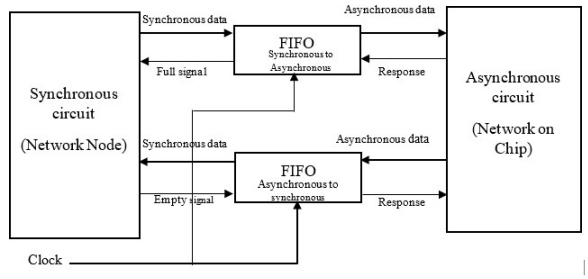
(3). Data buffer (first-in-first-out, FIFO) interface: Asynchronous FIFOs have been used extensively in synchronous circuits. This approach can also be used for a network interfaces but uses asynchronous rather than synchronous handshake protocol on the asynchronous circuit side [17]. This method has high data throughput but is affected by buffer length, its transmission delay, area, and energy consumption are the largest among all methods. The data buffer interface is shown in Fig. 2c.



(a) Synchronous Interface



(b) Stoppable clock Interface



(c) Data buffer Interface

Fig. 2 Synchronous and Asynchronous Interface methods in NoC

This article only discusses a network-on-a-chip using asynchronous routers. Some papers refer to distributed synchronous networks as GAL's systems [2], [18], but these networks do not directly use Asynchronous circuits, which is inconsistent with the Asynchronous circuit-based NoC concept discussed in this paper.

2.3. Flow Control Methods

Asynchronous NICs have fewer flow control methods than synchronous NICs. A small number of asynchronous on-chip networks use circuit switching to provide stable communication bandwidth and transmission delay, most of the others use packet switching. Tree networks and simple mesh networks with communication requirements are often used in the

basic Wormhole Routing method. Virtual Channels (VCs) are mostly used in mesh networks that need to provide high bandwidth guarantees [19]. Throughput networks can use Spatial Division Multiplexing (SDM) flow control methods. The throughput improvement of VC in an asynchronous network-on-chip is not as good as that of a synchronous network-on-chip. In order to ensure the correct execution of the handshake protocol, they exchange within the router. The swap unit cannot pre-allocate the path before the path is released so the swap arbitration and data transfer cannot be performed in parallel. Asynchronous VC routers are required for each. The switching unit is reconfigured with each flit rather than a message, which improves the configuration frequency of the switching unit. Since exchange arbitration and data transfer cannot be performed in parallel, the transfer times for microchips are significantly longer and network speeds drop, offsetting most of the throughput gains from virtual channels. Compared with VC, SDM has higher network throughput performance. SDM uses virtual circuits to transmit messages [20]. An exclusive part of a virtual circuit channel bandwidth is responsible for transmitting the entire message. Since multiple packets can pass through the same data channel through different virtual circuits, SDM is the same as VC can solve the problem of network congestion. But SDM does not introduce additional exchange arbitration time. Each virtual circuit has its own part of the bandwidth, routing. The switching unit within the switch only needs to be reconfigured for newly arrived packets, not the flit.

2.4. Quality of Service

Asynchronous on-chip networks, especially mesh on-chip networks, mainly serve MPSoC systems. Network nodes with different functions often have different communication quality requirements. There are generally three measures of communication quality: bandwidth, packet transmission delay, and delay jitter. A constant bandwidth network ensures that a portion of the data flow has constant average bandwidth. Hard real-time networks ensure that the maximum packet transmission delay of a part of the data flow is less than a hard deadline [21]. Soft real-time networks guarantee that the average transmission delay of a portion of the data stream is less than a predetermined deadline. A delay jitter is used to measure the controllability of packet arrival time. Small delay jitter means that the delay distribution of the packet has a small variance, and the packet receiver can better predict the arrival time of the packet. Due to the lack of a global clock, the asynchronous on-chip network cannot define time slices, and the time-division multiplexing flow control commonly used in the synchronous on-chip network is difficult to apply to the asynchronous on-chip network to provide hard real-time services. Asynchronous on-chip networks mostly use VCs to provide soft real-time bandwidth-constant services [22]. By setting different priorities for different VCs and using the priority arbitration mechanism, packets with high priorities can take priority over VCs to obtain constant bandwidth and a small average transmission delay. However, the average transmission delay and delay jitter are affected by the network load and increase with the increase of the data injection amount.

2.5. Routing Algorithm

Although theoretically asynchronous NICs can use any routing algorithm, limited by the complexity of router implementation, asynchronous NICs mostly use deterministic routing algorithms [23], and only a smaller number of asynchronous on-chip networks started using adaptive routing algorithms to increase network reliability. There are two main types of deterministic routing algorithms used in asynchronous on-chip networks: source routing and XY routing [24]. Source routing requires the originating node of the message to pre-determine the communication path and store the path in the header microchip of the message, while XY routing only needs to store the address of the destination node of the message in the header microchip, and the router will automatically specify the address according to the address path. These two routing algorithms do not require routers to perform complex logical operations, so they can be easily implemented by asynchronous circuits. Source routing can be used with any topology, while XY can only be used with regular mesh networks. When the scale of the network is large, source routing will consume more packet space to mark long network paths. At this time, the packet space load of XY routing is relatively small.

2.6. Low Power Design

Asynchronous circuits are not in all cases energy efficient than synchronous circuits, only when network utilization is low do, they consume less energy than normally powered synchronous circuits. When the network is running at full load, the energy consumption of the asynchronous circuit is similar to that of the synchronous circuit that improves the power supply, that is to say, it consumes more energy than the synchronous circuit with a normal power supply. Asynchronous network-on-chip is not suitable for MPSoC which needs to run at full capacity [25].

The power consumption of asynchronous circuits has a faster growth rate than that of synchronous circuits: using asynchronous networks to transmit data per unit length consumes more energy. A closer analysis showed that a single-track pipeline is more energy efficient than a multi-track pipeline and that the two-phase handshake protocol consumes less energy than the four-phase handshake protocol. Asynchronous NICs for low-power designs should use single-rail pipelines with two-phase handshakes whenever possible [26].

Dynamic voltage control can also be used in asynchronous routers to reduce leakage current in asynchronous circuits. Similar to synchronous, asynchronous circuits can also operate at low supply

voltages. Since the swing of the signal becomes smaller at low supply voltage, the dynamic power consumed by the circuit is correspondingly reduced, but the circuit speed becomes slower [27]. Unlike synchronous circuits, low supply voltages do not affect the function of asynchronous circuits. Dynamically controlling the voltage of an asynchronous circuit is safer. Since the asynchronous network-on-chip is more suitable for low-load network applications, the additional network delay introduced by the use of dynamic voltage control will not cause a large functional impact. After examining the finest existing NoC designs in the literature, it was discovered that while all NoC designs are focused on performance, power is also a primary concern in the current situation. Furthermore, with asynchronous NoC, the router will just transfer data; no clock will be transferred via the NoC router. This structure necessitates the requirement of a clock and data recovery circuit at the receiver, which increases the area and power consumption of the device connected to the NoC router. To address the aforementioned issues, in the proposed design, created a router structure based on FSM to lower the power consumption of the NoC, as well as a clock and data recovery circuit to reduce the power consumption and area of the device connected to the NoC.

3. PROPOSED METHOD

This section describes the proposed low power CDR circuit for asynchronous NoC. The proposed method is explained in two sections: CDR for Asynchronous NoC and Asynchronous NoC architecture for MPSoC.

3.1. Clock and Data recovery circuit

The clock data recovery circuit (CDR) is an important circuit in the data transmission system. In a communication system, the data stream is generally transmitted in the channel without a clock, and the receiving end needs to recover the effective digital signal from the noise-containing signal. The function of the clock data recovery circuit is to find the clock and phase sampling data so that the data recovery is the most stable and reliable. The traditional phase-locked loop is an effective method for tracking the clock and phase. It has the advantages of tracking zero frequency difference of clock and fixed tracking phase difference and has been widely used in much clock recovery. However, for burst data signals, the clock data recovery circuit using a traditional phase-locked loop often cannot meet the requirements of fast synchronization, a large phase jitter can easily lead to loss of lock, and the locking time of the phase-locked loop method is very long. Aiming at the transmission system of a burst data signal, an all-digital fast-locking improved lead-lag clock data recovery method is proposed, which can quickly determine the sampling clock.

3.1.1. Design principle and circuit

This circuit is mainly composed of an edge detector, a frequency capture device, a phase tracker, and a synchronous clock generator. The block diagram of Clock and Data recovery circuit is presented in Fig 3. Compared with the traditional super-pre-lag phase-locked loop method, this circuit adds a frequency catcher and improves the synchronous clock generator. The frequency catcher counts the local high-frequency clock period between two adjacent data edges, and selects the minimum value within a period of time as the basic oscillation period of the lead-lag phase-locked loop synchronous clock. The synchronous clock generator increases the oscillation period of the synchronous clock when encountering a leading pulse, and reduces the oscillation period when encountering a lagging pulse, and clears the synchronous clock counter after encountering a data edge. To prevent entering metastability, in the edge detector, the circuit uses a local high frequency clock.

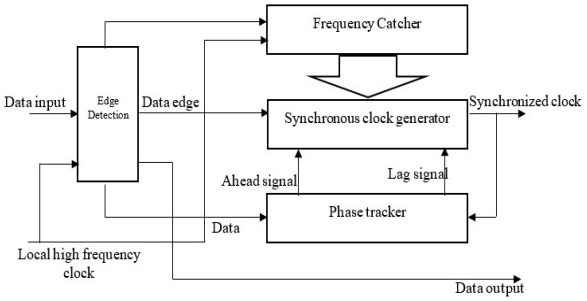
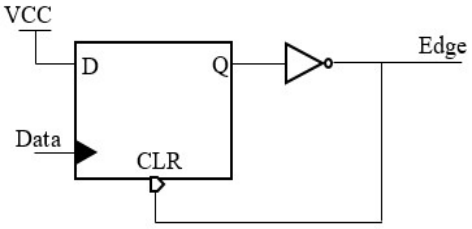


Fig. 3 Block diagram of Clock and Data recovery circuit principle

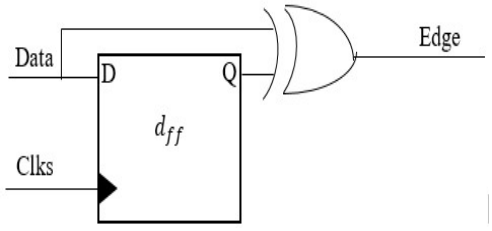
3.1.2. Circuit Implementation

This circuit can be realized on the Zynq SoC platform and uses Xilinx Zynq -7000 series to complete the circuit design.

At present, the commonly used edge detectors include asynchronous detection and synchronous detection. Asynchronous detection uses D flip-flops and NOT gates to realize the edge detection function, as shown in Fig. 4a. Synchronous detection uses flip-flops and XOR gates to realize the detection function. Function, as shown in Fig. 4b, when the data sampled by the D flip-flop skips an edge, a high level of one clock cycle will appear at the edge end. This edge detector selects the synchronization detection structure.



(a) Asynchronous edge detection sub circuit



(b) Synchronous edge detection sub circuit

Fig. 4 Edge detector

The frequency capture device is completed with a counter, and its structural block diagram is shown in Fig 5. The local high-frequency clock between two data edges is counted, compared with the previously stored value, and if the count value is less than the stored value, the previously stored value is replaced with the counter value. The phase tracker uses the "AND" of the data edge detected by the synchronous clock and the edge detector as the leading signal, and the "AND" of the non-sum data edge of the synchronous clock as the lagging signal. Since the synchronous clock and the edge detector are both connected with the local high-frequency time clock is synchronized, so the phase tracking circuit is synchronized with the local high-frequency clock.

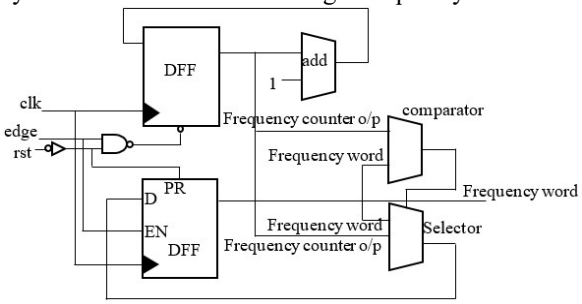


Fig. 5 Structural block diagram of frequency capture

In order to increase the locking frequency range, the lead-lag adjustment of the synchronous clock generator does not use the subtraction gate and an additional gate method in the traditional lead-lag phase-locked loop method. After obtaining the data frequency from the frequency capturer, if it is judged that the data edge is ahead, changing the maximum value T_{data} of the synchronous clock cycle counter to $T_{data} - 1$, and if it is judged that the data edge is lagging, increase the maximum value T_{data} of the synchronous clock cycle counter to $T_{data} + 1$.

To reduce the synchronous clock jitter, in this circuit, the synchronous clock is not generated by counting the local high-frequency clock overflow and then inverting the phase. Instead, it is stipulated that when the count value is less than or equal to 1/2 of the frequency word, the synchronous clock output is low. When the value of the counter is greater than 1/2 of the frequency word, let the synchronous clock output will be high. The step of the synchronous clock cycle using this method is one local high-frequency clock cycle, while the step of the synchronous clock cycle generated by the overflow inversion method is two local high-frequency clock cycles, so the time synchronization clock of this circuit is less jitter.

3.2. CDR integration with Asynchronous NoC

This section describes the integration of the CDR circuit with asynchronous NoC. The integration of the CDR circuit which is described in section 3.1.2 with the Asynchronous NoC circuit as presented in Fig 6.

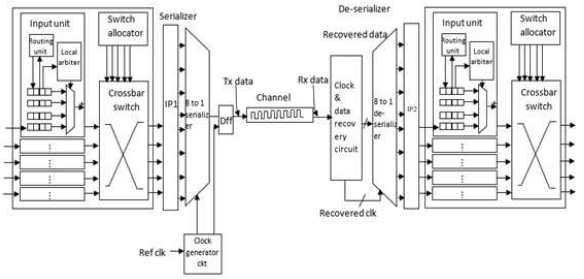


Fig. 6 CDR circuit integration with Asynchronous NoC

A general two-dimensional mesh is assumed for the network topology. The internal operation of the router is explained as follows:

3.2.1. Flow control method

There are two types of NoC flow control methods: the Wormhole method, which has a buffer smaller than the packet size, and the Virtual Cut-Through method, which has a buffer equal to the packet size. In the Virtual Cut Through method, the flow is calculated in bucket units. In general, NoC, the wormhole method is adopted to reduce the number of buffers, but Wormhole starvation occurs frequently when priority control is performed on a per flit basis. For that reason, the Virtual Cut-Through method is used to perform priority control on a packet-by-packet basis in routers for real-time systems.

3.2.2. Routing Algorithm

NoC routing algorithms can be categorized into two types: fixed routing, in which a route is uniquely determined for one destination, and adaptive routing, in which a route is dynamically determined according to network conditions. Since the order of packets varies in type of routing, a large buffer is required on the receiving node side. Therefore, in this research, dimension-order routing is considered, which is the most common fixed routing.

3.2.3. Router architecture

The architecture of a router for real-time systems is shown in Fig.6. The router has five input and output ports, four of which are connected to neighbouring routers and one to the core. The Input Unit stores the received packets in the buffer calculates the output port by the Routing Unit and arbitrates the input virtual channel by the Local Arbiter. The Switch Allocator performs Crossbar Switch arbitration, and the Crossbar Switch passes packets to each output port according to the result of the Switch Allocator. The output port of each router is integrated with the CDR circuit. In between the output port and CDR circuit, the serializer circuit is used to convert the parallel data into serial data. At the other end, a de-serializer is used to convert serial data into parallel data. This entire operation is called SerDes which is mostly used in a high-speed communication system.

Packet transfer processing is divided into the following steps:

Routing Computation (RC) /Local Arbitration (LA).

RC: Calculate the output port from the destination address included in the header information.
LA: Arbitrate input virtual channel requests to the Switch Allocator based on priority.
Switch Allocator (SA): Routes buckets through the Crossbar Switch based on priority arbitrate for an Input Unit.
Switch Traversal (ST): Transmits packets from Input Units selected by the Switch Allocator.

4. RESULTS AND DISCUSSION

The router is used to connect packet-switched networks or subnets. It serves two main functions:1) It manages traffic between these networks by forwarding the packets of data to the desired IP address, allowing multiple devices to share the same internet connection. But the router doesn't transmit the clock through it. So, this work designed a low-power router that can generate a clock signal and transmit the clock signal. 2) Generate a clock signal from the data sent by the router. So firstly, it needs to serialize the data sent by the router. Then pass the serialized data to the clock data recovery module, which allows the clock signal to be recovered from the data sent by the router. By designing in this manner, a clock signal is obtained from the router itself by consuming low power and less area.

The proposed architecture is designed using Verilog-HDL and synthesized using Xilinx Vivado 2019.2. The proposed architecture is implemented on the Zed board (Zynq evaluation and development Kit) in order to verify the functionality and performance of the proposed architecture in terms of area and power. All the modules of the architecture are designed using Verilog- HDL and the final architecture is designed using a bottom-up approach. The proposed design can process up to 64 different 64-bit data simultaneously and generate serial output.

The simulation result of the proposed architecture is shown in Fig. 7. The simulation results show that the output of the proposed architecture is generated serially. In order to recover the clock, a phase generator is used that will generate multiple clock signals with different phases. In the proposed design, a phase detector that will detect 16 different phases is used. Different phases delayed incoming data signal is ex-ored in original form and shift left form. When both the ex-ored output is equal then the phase of the clock is locked and the locked phase clock is the recovered clock as shown in Fig. 8.

The synthesized design of the proposed architecture as shown in Fig.9.The area utilization summary of the proposed architecture as shown in Fig.10. Proposed architecture is implemented in the Zed board (Zynq evaluation and development Kit). Upon implementation, obtain the area utilization summary in terms of LUTS and Slice registers. The proposed design consumes 33064 LUTS with a utilization percentage of 62.15%. In terms of slice registers, the proposed architecture consumes 25152 slice registers with a utilization percentage of 23.64%. Out of this 23.64%, 20.93% is occupied by the Flip Flop and the rest 2.71% is occupied by the Latches.

Fig. 7 Simulation results showing packet transmissions through a router

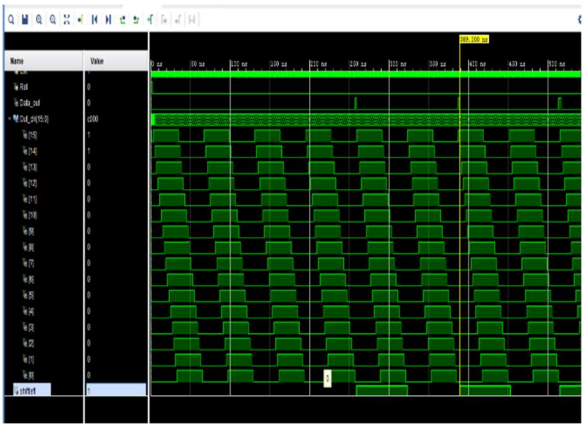


Fig. 8 Simulation results of the proposed architecture

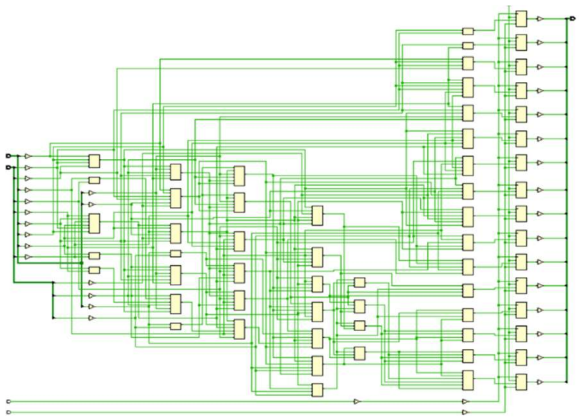


Fig 9. Synthesized design of the proposed architecture

1. Slice Logic

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	33064	0	53200	62.15
LUT as Logic	33064	0	53200	62.15
LUT as Memory	0	0	17400	0.00
Slice Registers	25152	0	106400	23.64
Register as Flip Flop	22272	0	106400	20.93
Register as Latch	2880	0	106400	2.71
F7 Muxes	0	0	26600	0.00
F8 Muxes	0	0	13300	0.00

Fig.10 Area Utilisation summary of the proposed architecture

The total on-chip power of 0.103W as shown in Fig 11. The research work reported in the [18], 2D 4x4 Asynchronous Mesh NoC architecture with a data size of 48-bits and depth of 5 is proposed. In the proposed architecture, 2D 4x4 Asynchronous Mesh NoC architecture with a data size of 64-bits and depth of 64 is proposed. When compared with the research article cited in [2], [18], and [25], the proposed design shows better performance in terms of area and power. The increase in the area is due to the increase in the data depth, data width, and circuit for clock recovery. There is a power reduction of 5 times compared to [18] and 2 times compared to [2] and 9 times compared to [25] as shown in Table 1 and in fig. 12-15.

Fig. 11 Power Utilisation summary of the proposed architecture

Table 1. Comparison of proposed work with the existing works

References	Mesh Size	Data Width	Data Depth	Area (LUTs)	Power (W)
[2]	4x4	-	-	-	0.211
[18]	4x4	48	5	3544	0.56
[25]	3x3	64	-	-	0.940
Proposed	4x4	64	64	33064	0.103

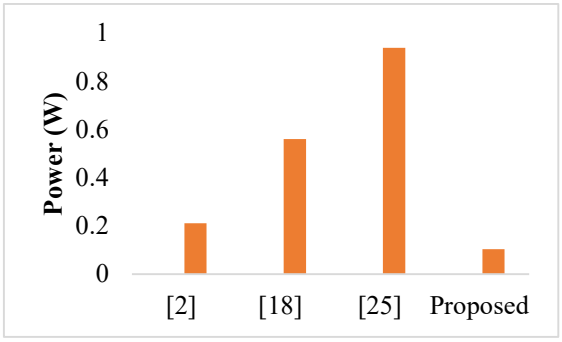


Fig. 12 Power of proposed work with the existing works

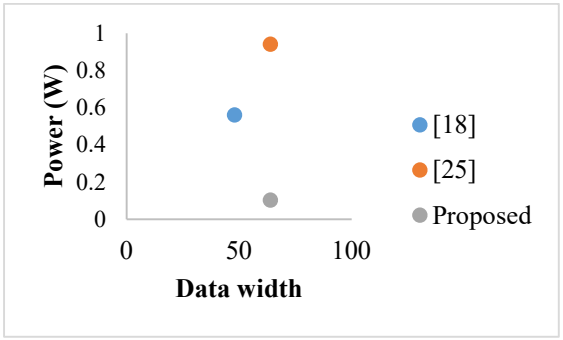


Fig. 13 Data width vs. Power

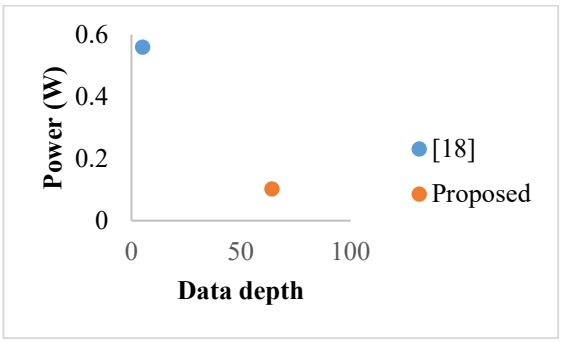


Fig. 14 Data depth vs. Power

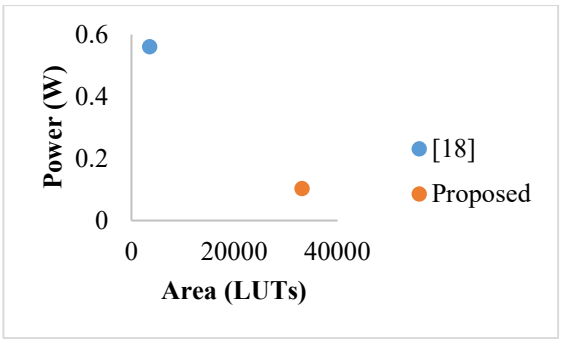


Fig. 15 Area vs. Power

5. CONCLUSION

In this, 2D 4x4 Mesh NoC architecture without buffer is proposed in this paper. In this proposed NoC router, a low-power Clock and Data Recovery circuit is designed for Asynchronous NoC in order to recover the clock from the router-transmitted output. The proposed design efficiently handles medium to large network loads. The data is transmitted parallel out of the router and then processed serially into and out of the Clock and Data Recovery circuit. By transmitting data serially, we achieve an advantage of a smaller number of conducting wires, hence reducing the cost of the interface and simple interface to the receiver. The proposed design exhibits less power consumption of 0.103W compared to the existing systems and can process 64-bit of data samples with a data depth of 64.

ACKNOWLEDGMENT

The authors wish to thank CV Raman Heterogeneous Computing Lab, Department of ECE, University College of Engineering (A), Osmania University, Hyderabad for providing a supporting environment for carrying out this work.

REFERENCES

[1] Torres, L., Benoit, P., Sassatelli, G., Robert, M., Clermidy, F., & Puschini, D. (2011). An introduction to multi-core system on chip–trends and challenges. Multiprocessor System-On-Chip, 1-21. DOI: 10.1007/978-1-4419-6460-1_1

[2] Imai, M., Van Chu, T., Kise, K., &Yoneda, T. (2016). The synchronous vs. asynchronous NoC routers: an apple-to-apple comparison between synchronous and transition signalling asynchronous designs. In 2016 Tenth IEEE/ACM International Symposium on Networks-On-Chip (NOCS) (pp. 1-8). IEEE. DOI: 10.1109/NOCS.2016.7579330

[3] Bhardwaj, K., & Nowick, S. M. (2018). A continuous-time replication strategy for efficient multicast in asynchronous NoCs. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 27(2), 350-363. DOI: 10.1109/TVLSI.2018.2876856

[4] Nayak, P. (2021). A Study of Technology Roadmap for Application-Specific Integrated Circuit (Doctoral dissertation, Rice University). url: https://hal.archives-ouvertes.fr/hal-03313367

[5] Ax, Johannes, N. Kucza, M.Porrmann, U.Rueckert and T.Jungeblut, “Aysnchronous network-on-chips(NoCs) for resource efficient many-core architecture”, Asynchronous Circuit Applications, 2020. DOI: 10.1049/PBCS061E_ch8

[6] He, A., Feng, G., Jilin, Z., & Wu, J. (2019). An asynchronous mesh NoC-based booth multiplication. IET Circuits, Devices & Systems, 13(1), 73-78. DOI: 10.1049/iet-cds.2018.0058
[7] Akhtar, M. Nishat, Q. Azam, T.A. Almohamad, J.M. Saleh, E.A.Bakar, and A.A.Janvekar, "An Overview of Multi-Core Network-On-Chip System to Enable Task Parallelization Using Intelligent Adaptive Arbitration", Intelligent Manufacturing and Mechatronics, pp. 15-38, 2021. DOI: 10.1007/978-981-16-0866-7_2
[8] Karthikeyan, A., & Kumar, P. S. (2018). GALS implementation of randomly prioritized buffer-less routing architecture for 3D NoC. Cluster Computing, 21(1), 177-187. DOI: 10.1007/s10586-017-0979-0
[9] Abdelhadi, A. M., & Li, H. (2021, August). Enabling Mixed-Timing NoCs for FPGAs: Reconfigurable Synthesizable Synchronization FIFOs. In 2021 31st International Conference on Field-Programmable Logic and Applications (FPL) (pp. 312-318). IEEE. DOI: 10.1109/FPL53798.2021.00062
[10] Singh, M., &Nowick, S. M. (2000, April). High-throughput asynchronous pipelines for fine-grain dynamic data paths. In Proceedings Sixth International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC 2000) (Cat. No. PR00586) (pp. 198-209). IEEE. DOI: 10.1109/ASYNC.2000.837017
[11] Abad, P., Puente, V., & Gregorio, J. A. (2009, February). MRR: Enabling fully adaptive multicast routing for CMP interconnection networks. In 2009 IEEE 15th International Symposium on High Performance Computer Architecture (pp. 355-366). IEEE. DOI: 10.1109/HPCA.2009.4798273
[12] Kao, Y. H., Yang, M., Artan, N. S., & Chao, H. J. (2011). CNoC: high-radix clos network-on-chip. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 30(12), 1897-1910. DOI: 10.1109/TCAD.2011.2164538.
[13] Muhammad, S. T., Saad, M., El-Moursy, A. A., El-Moursy, M. A., & Hamed, H. F. (2019). CFPA: Congestion aware, fault tolerant and process variation aware adaptive routing algorithm for asynchronous Networks-On-Chip. Journal of Parallel and Distributed Computing, 128, 151-166. DOI: 10.1016/j.jpdc.2019.03.001
[14] Li, F., Liu, W., Gao, W., Liu, Y., & Hu, Y. (2022). Design and Reliability Analysis of a Novel Redundancy Topology Architecture. Sensors, 22(7), 2582. DOI: 10.3390/s22072582
[15] Taraate, V. (2022). FPGA Design and Guidelines. In Digital Logic Design Using Verilog (pp. 385-401). Springer, Singapore. DOI: 10.1007/978-981-16-3199-3_17.
[16] Attia, Brahim, W.Chouchenne, A.Zitouni, and R.Tourki, "A Stoppable clock based Approach for Low Power Network Interface Design in a Network on Chip", International Journal of Computer Science Issues (IJCSI) Vo.10, no. 3 pp.70, 2013. URL: www. IJCSI.org.
[17] Maurya, S., Sharma, S., & Saroj, J. (2019). Design of High-Performance Mixed-Clock FIFO. In Advances in System Optimization and Control (pp. 23-31). Springer, Singapore. DOI: 10.1007/978-981-13-0665-5_3
[18] Patil, Trupti and S.Anuradha, "Design and implementation of asynchronous NOC architecture with buffer-less router", Materials Today: Proceedings 49, pp. 756-763, 2022. DOI: 10.1016/j.matpr.2021.05.282
[19] Dimitrakopoulos, G., Psarras, A., &Nicopoulos, C. (2022, June). Virtual-Channel Flow Control across Mesochronous Clock Domains. In 2022 11th International Conference on Modern Circuits and Systems Technologies (MOCAST) (pp. 1-4). IEEE. DOI: 10.1109/MOCAST54814.2022.9837772
[20] Farrokhbakht, H., Gratz, P. V., Krishna, T., San Miguel, J., & Jerger, N. E. (2022, April). Stay in your Lane: A NoC with Low-overhead Multi-packet Bypassing. In 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA) (pp. 957-970). IEEE. DOI: 10.1109/HPCA53966.2022.00074
[21] Panem, C., Rane, U. V., & Gad, R. S. (2022). Data of multilayer mesh NoC performance analysis for throughput and delay over FTP and CBR applications. Data in Brief, 42, 108196. DOI: 10.1016/j.dib.2022.108196
[22] Felicijan, T. (2004). Quality-of-service (QoS) for asynchronous on-chip networks. The University of Manchester (United Kingdom).
[23] Gogoi, A., Ghoshal, B., Sachan, A., Kumar, R., & Manna, K. (2022). Application driven routing for mesh based network-on-chip architectures. Integration, 84, 26-36. DOI: 10.1016/j.vlsi.2021.12.008
[24] Lallas and Efthymios N, "An Evaluation of Routing Algorithms in Traffic Engineering and Quality of Service Provision of Network on Chips", Engineering 13, no. 01 pp. 1, 2021.DOI: 10.4236/eng.2021.131001
[25] Ax, Johannes, N. Kucza, Marten Vohrmann, T. Jungeblut, M. Porrmann, and U. Rückert, "Comparing synchronous, mesochronous and asynchronous NoCs for GALS based MPSoCs", In 2017 IEEE 11th International Symposium on Embedded Multicore/Many-core Systems-On-Chip (MCSoC), pp. 45-51. IEEE, 2017. DOI: 10.1109/MCSoC.2017.19.
[26] Wasif, A.Sandy, Salma Hesham, G.Diana, K.Hofmann, and A.Mohamed Abd Ghany, "Double-Layer Energy Efficient Synchronous-Asynchronous Circuit-Switched NoC", Electronics 10, no. 15, pp. 1821, 2021. DOI: 10.3390/electronics10151821
[27] Dashkin, Ruslan and Rajit Manohar, "General Approach to Asynchronous Circuits Simulation Using Synchronous FPGAs", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2021. DOI: 10.1109/TCAD.2021.3131546