A New CDMA Based NOC Architecture with CTC Scheme

S. Bhuvaneshwari *(ME-VLSI DESIGN)*
ECE Department, Syed Ammal Engineering College, Ramanathapuram
Mr. J. Sakubar Sadiq M.E., (Phd)
Associate professor, Syed Ammal Engineering College, Ramanathapuram

Abstract- *Network on Chip (NoC) is a promising solution for on-chip communications to enable integrating various processors and on-chip memories into a single chip. As a high performance on-chip communication method, the code division multiple access (CDMA) technique has been recently applied to NoCs. This project proposes an efficient, (NoC) architecture for a hard real-time multiprocessor platform. The NoC implements a best message-passing and Connection Then Credit (CTC) communication between processor cores which Propose an approach for computing the end-to-end delay bound of individual variable bit-rate flows in an First Input First Output multiplexer with aggregate scheduling under weighted round robin (WRR) has an appropriate weight allocation can guarantee the total delay bound minimization in the network. In which data are transferred in a pipelined fashion, from the local memory of the sending core to the local memory of the receiving core, without any dynamic arbitration. The proposed router coded in VHDL and simulated using Xilinx 12.1.*

Keyword: *Code Division Multiple Access (CDMA), Weighted Round Robin (WRR), Connection Then Credit (CTC), End-to-End delay bound.*

I. INTRODUCTION

Future Generations of systems-on-chip (SoCs) will consist of heterogeneous multicore architectures with a main general purpose processor, possibly, itself consisting of multiple processing cores, and many task specific subsystems that are programmable and/or configurable [1], [2]. These subsystems, which are also composed of several cores, will provide more power efficient support to important classes of embedded applications across multiple use case scenarios [3]–[5]. Heterogeneity is the combined result of hardware specialization, reuse of intellectual property modules, and the application of derivative design methodologies [6]. Programmability makes it possible to upgrade dedicated software and add the support of new applications and features that were not included at the chip design time.

Most on-chip networks in practice are general-purpose and use routing algorithms such as dimension-ordered routing and minimal adaptive routing. To support permutation traffic patterns, on-chip permutation networks using application-aware routings are needed to achieve better performance compared to the general-purpose networks. These application-aware routings are configured before running the applications and can be implemented as source routing or distributed routing. However, such application-aware routings cannot efficiently handle the dynamic changes of a permutation pattern, which is exhibited in many of the application phases.

The difficulty lies in the design effort to compute the routing to support the permutation changes in runtime, as well as to guarantee [9] the permutated traffics. This becomes a great challenge when these permutation networks need to be implemented under very limited on-chip power and area overhead. Reviewing on-chip permutation networks (supporting either full or partial permutation) with regard to their implementation shows that most the networks employ a packet-switching mechanism to deal with the conflict of permuted data [3]–[6].

For a given flow, we study the maximum interference of all other flows based on the network calculus. This models have been defined and validated under round robin (RR) policy. The RR policy uses the same service level for each connection while multiple service levels allow to better adapt to the application requirements by providing a different bandwidth and the latency guarantees. A weighted RR (WRR) scheduling policy assigns the weights to concurrent communications to define the multiple service levels. Higher service levels have greater weights and do not preempt the lower ones. It is important for designers to find the appropriate weights in WRR policy such that the corresponding service levels can support the QoS requirements for each communication connection.

Here we proposes a Connection-then-credits (CTC) is a novel end-to-end flow control protocol to handle message-dependent deadlocks in best-effort networks-on-chip (NoC) for embedded multicore systems-on chip (SoCs). CTC is based on the classic end-to-end credit-based flow control protocol but differs from it because it uses a network interface microarchitecture where a single credit counter and a single input data queue are shared among all possible communications. This architectural simplification reduces the area occupation of the network interfaces and increases their design reuse; for instance, the same network interface can be used to connect a core independently of the number of incoming and outgoing communications. CTC, however, requires a handshake preamble to initialize the credit counter in the sender network interface based on the buffering capacity of the receiver network interface.

II. RELATED WORKS

In previous, a standard-basis based encoding/decoding method to leverage the performance and cost of CDMA NoCs in area, power assumption, and network throughput. In the transmitter module, source data from different senders are separately encoded with an orthogonal code of a standard basis and these coded data are mixed together by an XOR operation. Then, the sums of data can be transmitted to their destinations through the onchip communication infrastructure. In the receiver module, a sequence of chips is retrieved by taking an AND operation between the sums of data and the corresponding orthogonal code. After a simple accumulation of these chips, original data can be reconstructed. We implement our encoding/decoding method and apply it to a CDMA NoC with a star topology. Compared with the state-of-the art Walsh-code-based (WB) encoding/decoding technique, our method achieves up power saving and area saving together with decrease of encoding/decoding latency.

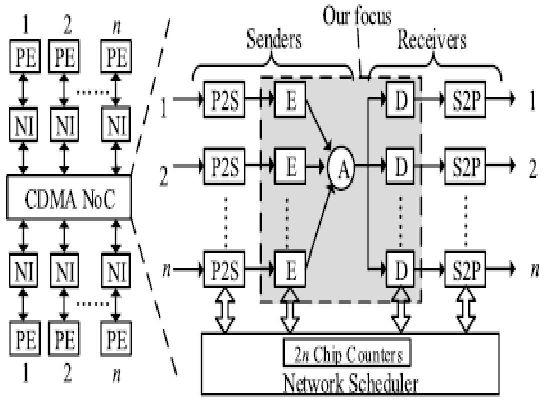

Fig: star topology CDMA technique to NoC

The basic structure of applying CDMA technique to NoC with a star topology is shown in Fig. In this figure, a PE executes tasks of the application and network interface (NI) divides data flows from PE into packets and reconstruct data flows by using packets from NoC. In the sender, packet flits from NI are transformed to a sequential bit stream via a parallel-to-serial (P2S) module. This bit stream is encoded with an orthogonal code in the Encoding module (E in Fig.). The coded data from different encoding modules are added together in the Addition module (A in Fig). Then, the sums of data chips are transmitted to receivers. In the receiver, Decoding modules (D in Fig. 1) reconstruct original data bits from the sums of data chips. Then these sequential bit streams are transformed t packet flits by serial-to-parallel (S2P) modules. Finally, these packet flits are transferred to NI. In the CDMA NoC, network scheduler receives the transmitting requests from senders and assigns proper spreading codes to the senders and requested receivers. Note that all-zero codeword is assigned to nodes having no data to transmit/ receive. Moreover, when there are multiple senders requesting the same receiver, the scheduler will apply an arbitration scheme, for example, round-robin. The chip counters calculate how many orthogonal chips are used in one encoding/decoding operation. Each node needs two chip counters, one for the sender and the other for the receiver. Note that packet flits from NI can also be transformed to multiple bit streams in the P2S module to make tradeoffs between power/area cost and packet transfer latency, and the scheduler should provide a bit-synchronous scheme to maintain the orthogonality of the transmitted channels.

Here focuses on a new encoding/decoding method for CDMA NoCs, we discuss some general concerns in scaling, topology, routing, and traffic pattern.

A. Scaling

The CN (CDMA NoC) can be scaled to different network sizes using two basic methods. In the direct scaling method, the length of orthogonal code will increase with the number of PEs and thus is more suitable for small-size NoCs (e.g., CDMA NoC with several PEs. In contrast, the luster-based scaling method, by which each cluster has several PEs and clusters are connected with each other, can be used to scale the network hierarchically to any required size .

B. Topology

Although a CDMA node cluster may be limited in the star topology, any other topologies can also be obtained by using the cluster-based scaling method. For example, Kim *et al*. developed a hierarchical star topology, Lee and Sobelman developed a mesh topology, and so on.

C. Routing

There exist various incremental and global routing schemes for CDMA N oCs. Consider an incremental routing, where the routing schemes are related to the packet formats. In general, the packet header contains the destination PE address. The source CN checks the destination address, determines the next-hop CN or PE, and allocates a corresponding spread code for the packet encoding and decoding to reach the right output port. The next-hop CN continues the process until the destination PE is reached.

D. Traffic Patterns

For the CDMA NoC, the influence of traffic patterns has been discussed before, and some real applications have also been mapped onto the CDMA NoCs to show their advantages over the packet-switched NoCs

III. PROPOSED SYSTEM

In this paper, we proposes a Connection-then-credits (CTC) is a novel end-to-end flow control protocol to handle message-dependent deadlocks in best-effort networks-on-chip (NoC) for embedded multicore systems-on chip (SoCs). CTC is based on the classic end-to-end credit-based flow control protocol but differs from it because it uses a network interface microarchitecture where a single credit counter and a single input data queue are shared among all possible communications. This architectural simplification reduces the area occupation of the network interfaces and increases design reuse; for instance, the same network interface can be used to connect core independently of the number of incoming and outgoing communications. CTC, however, requires a handshake preamble to initialize the credit counter in sender network interface based on buffering capacity of the receiver network interface.

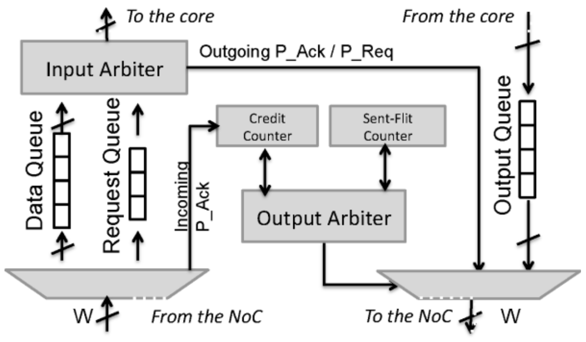

Fig.2 Block diagram of network interface supporting the CTC protocol.

Fig.2 shows the block diagram of a network interface supporting the CTC protocols. The CTC NI contains *two* input queues and *one* single output queue *independently* from the number of network interfaces that may require a connection with this NI or the number of cores that this NI may address. The data queue *Q*in is used to store incoming data flits while the request queue *Q*req is used to store incoming connection requests. Additionally, the *output arbiter* uses two counters to account for the available storage on the connected peer network interface and the number of flits that remain to be sent before terminating the current connection.

The operations of the CTC protocol with three snapshots.

1) Network interfaces NI_0 and NI_2 address NI_1 with two P_REQ packets requesting the transfer of two messages of 100 and 80 flits, respectively.

2) NI_1 selects NI0 to initiate the connection while it stores the other request in the *request queue*. Assuming that $K = 5$ and that the space currently available in the *data queue* is $S = 10$, NI_1 first generates a P ACK containing S flit-credits used to initialize NI0's credit counter with the maximum available storage.

3) Assuming $P\max \geq 10$, a data-packet with 10 flits is injected in the NoC by NI_0 to be later absorbed and stored in the data queue of NI_1. Eventually, NI1 will generate a total of $[(M-S)/K]+1 = 19$ P ACK packets for this connection to enable the transfer of 100 flits from NI_0.

A new CTC end-to-end flow control protocol has been proposed as an area-efficient solution to the message-dependent deadlock problem that characterized by a simpler and more modular network interface architecture. CTC-supporting network interfaces use one single input data queue and one output credit counter. Hence, the overall number of queues per network interface remains equal to two, the total amount of storage is reduced, and the overall network-interface design becomes independent from the communication requirement of the particular core, thus increasing its reusability. On the other hand, any new communication between a pair of peer nodes requires the preliminary completion of a handshake procedure to initialize output credit counter on the producer side (after the connection has been established CTC works in a way similar to the original CB flow protocol). This procedure necessarily increase latency of a message transfer and also reduces network throughput for small messages.

We also propose an approach for computing the end-to-end delay bound of individual variable bit-rate flows in an First Input First Output multiplexer with aggregate scheduling under weighted round robin (WRR) policy. To this end, a network calculus to derive per-flow end-to-end equivalent service curves employed for computing least upper delay bounds (LUDBs) of the individual flows. Since the real-time applications are going to meet guaranteed services with lower delay bounds, the weights inWRR policy to minimize the LUDBs to satisfying the performance constraints.

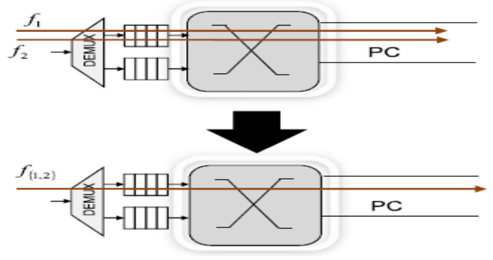

Fig.3: channel sharing and buffer sharing

To derive a delay bound per-flow passing a series of nodes, one simple way is to sum up the delay bounds at each node, which results in a loose delay bound. A theorem called Pay Bursts Only Once is known to give a tighter upper estimate on the delay bounds, when an end-to-end service curve is obtained prior to the delay computations. This accounts for bursts of the tagged flow only once instead of at each link independently. This principle also holds in aggregate scheduling networks. To this end, we propose the two following steps to derive the end-to-end service curve for a tagge dflow.

1) *Step 1 (Intrarouter ESC):* This step derives intrarouter ESCs for each router through which the tagged flow is passing. Different resource sharing scenarios in each router are distinguished and intrarouter analysis models are built.

2) *Step 2 (Interrouter ESC):* In this step, according to the intrarouter analysis models, we present a set-theoretic approach that recognizes and investigates different contention scenarios that a flow may experience along its routing pathand in turn derive an end-to-end ESC for the tagged flow .

For extending our proposed analytical method to WRR policy, we expand the first step while the second step is unchanged. Similarly, to support some other arbitration policies, only the first step must be modified.

Intrarouter ESC

In this step, we consider three types of resource sharing, including channel and buffer sharing, channel sharing, and buffer sharing.

Channel and Buffer Sharing:

The multiple flows share both the same buffer and channel in the router, and are scheduled as a flow called aggregate flow. An aggregate flow including the tagged flow is named as tagged aggregate flow. In this case, intra-ESC is derived for the tagged aggregate flow instead of the tagged flow. Due to the contention scenarios, we will remove the contention flows from the ESC of a tagged aggregate flow in order to extract the ESC of the tagged flow.

We have extended our earlier proposed methodology for deriving the per-flow delay bounds under RR policy to WRR scheduling and then compared them. We have developed algorithms to automate the analysis steps. It is notable that the proposed methodologies for both the RR and the WRR do not deal with the back pressure, but we have calculated the buffer size thresholds to make sure the back pressure does not occur in the network. Based on these analytical models, we have presented two optimization problems for weight allocation in the WRR scheduling, one for minimizing the total worst case delays and one for minimizing both the total worst case delays and their variance under performance requirements to control the per-flow delay bounds. We have also demonstrated that the proposed model exerts significant impact on communication performance. The algorithm for solving the proposed minimization problems runs very fast.

For the case study, the optimized solution is found within ~1 s. The contribution of this paper is providing a performance evaluation tool for designers to make agood decision at the Design phase. The algorithm can be performed at run time as it is quite fast, but it needs a control infrastructure to get feedback from the network and distribute the results. On the other hand modifying the weights at run time is not easy.

S.No	Parameter	Existing	Proposed
1.	Slice	75	70
2.	LUT	129	119
3.	IOB	55	55

Table 1: Comparison Table

IV. EXPERIMENTAL RESULT

The proposed circuit are simulated and synthesized by using modelsim and xilinx12.1 which occurs low area than the existing. The experimental results are given The simulation results of layout and the waveforms are shown in the fig.4 and fig.5.

Then the RTL schematic of the proposed are shown in fig.5.

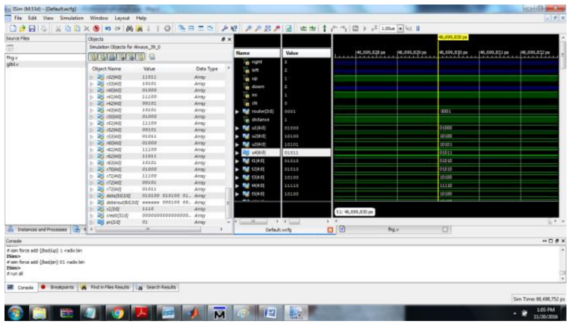

Fig.4 simulation results

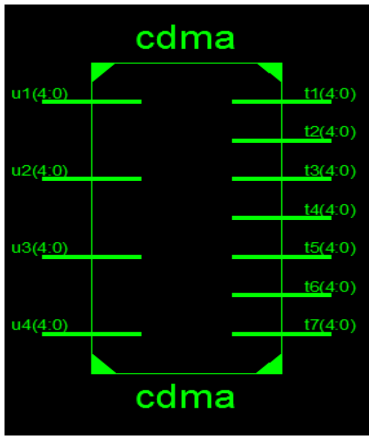

Fig.5 RTL schematic

V. PERFORMANCE ANALYSIS

The Figure given below is shown that there is a considerable reduction based on no of transistors and the performance chart has been shown below in fig.7

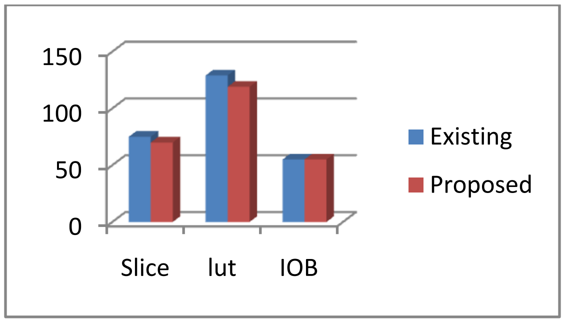

Fig.7 performance analysis of existing and proposed

VI. CONCLUSION

In this paper, a Connection-then-credits (CTC) is a novel end-to-end flow control protocol to handle message-dependent deadlocks in best-effort networks-on-chip (NoC) for embedded multicore systems-on chip (SoCs) is proposed. We also proposed an approach for computing the end-to-end delay bound of individual variable bit-rate flows in an First Input First Output multiplexer with aggregate scheduling under weighted round robin (WRR) policy. The simulation results are experimentally shown by using the tool Xilinx12.1 and modelsim which occurs low area than the which occurs low area than the existing.

References

[1] Evangelia Kasapaki, Martin Schoeberl, *Rasmus BoSørensen* ,Christoph, Kees Goossens and *Jens Sparsø (2016),"*Argo: A Real-Time Network-on-Chip Architecture With an Efficient GALS Implementation", *IEEE Trans.on VLSI Systems,vol. 24, no. 2, pp.479-492.*

[2] Qian Z.-L., .Juan D.-C, Bogdan P., Tsui, C.-Y. Marculescu R. (*2016*),"A support vector regression (SVR)-based latency model for network-on-chip architectures,"*IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 35, no. 3, pp. 471–484.*

[3] Wiklund D. and Liu D.(*2015*), "SoCBUS: Switched network on chip for hard real time embedded systems," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS), p. 78a.*

[4] Müller C.T, Kasapaki .E, Sørensen .R.B, and Sparsø.J (2014), "Synthesis and layout of an asynchronous network-on-chip using standard EDA tools"in *Proc. NORCHIP, pp 1–6.*

[5] Indrusiak. L.S(2014), "End-to-end schedulability tests for multiprocessor embedded systems based on networks-on-chip with priority-preemptive arbitration," *J. Syst. Archit., vol. 60, no. 7, pp. 553–561,*

[6] Sallam M., El-Kharashi M.W., and Dessouky M. (*2014*), "The connectionthen-credit flow control protocol for networks-on-chips: Implementation trade-offs," in *Proc. Int. Workshop Netw. Chip Archit. (NoCArc)*,pp. 25–30

[7] Sparsø J., Kasapaki E., and Schoeberl M. (*2013*), "An area-efficient network interface for a TDM-based network-on-chip," in *Proc. Design, Autom. Test Eur. (DATE), pp. 1044–1047.*

[8] Vidapalapati A.,Vijayakumaran V., Ganguly A., and Kwasinski A.(*2012*), "NoC architectures with adaptive code division multiple access based wireless links," in *Proc. IEEE Int. Trans. Circuits Syst.*, pp. 636–639.

[9] Jafari F,Z.Lu,A. Jantsc, Yaghmaee M.H (*2010*), "Buffer optimization in network-on-chip through flow regulation," *IEEE Trans. Comput.-AidedDesign Integr. Circuits Syst., vol. 29, no. 12, pp. 1973–1986*

[10] Bertozzi*eta .Dl.(2010),"NoC synthesis flow for customized domain specific* multiprocessor systems-on-chip," *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 2, pp. 113–129

[11] N . Concer, L. Bononi, M. Soulie, R. Locatelli, and L. P. Carloni, "The connection-then-credit flow control protocol for heterogeneous multicore systems-on chip," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 29, no. 6, pp. 869–882, Jun. 2010.

[12] F. Jafari, Z. Lu, A. Jantsch, and M. H. Yaghmaee (2010), "Buffer optimization in network-on-chip through flow regulation," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 29, no. 12, pp. 1973 1986,

[13] Y. Qian, Z. Lu, and Q. Dou (2010), "QoS scheduling for NoCs: Strict priority queueing versus weighted round robin," in Proc. 28th Int. Conf. Comput. Design(ICCD), pp. 52–59.

[14] Panades. I.M and Greiner. A (2007), "Bi-synchronous FIFO for synchronous circuit communication well suited for network-on-chip in GALS architectures," In *Proc. IEEE/ACM Int. conf. Netw.-Chip (NOCS)*, pp. 83–92.

[15] Flachs *et al. B(2008), "The micro architecture of the synergistic processor* for a cell processor," *IEEE J. Solid-State Circuits, vol. 41, no. 1, pp. 63–70.*