IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 5, Ver. II (Sep.- Oct. 2017), PP 08-13
www.iosrjournals.org

Packet Transfer in DLH Networks

Milen Angelov [1], Nadezhda Ruskova [2]

[1]*(Computer Science and Engineering Department, Technical University of Varna, Bulgaria)*
[2]*(Computer Science and Engineering Department, Technical University of Varna, Bulgaria)*

Abstract: *An effective implementation of message passing between nodes in multiprocessing systems is a key factor in achieving high performance and efficiency in their operation. The flow-control mechanism in a router, one of the fundamental functional nodes in MPP systems, allocates buffers and channel bandwidth for their data units – flits or packets. In this paper, we present an approach for packet transfer in a DLH network over full-duplex channels via cut-through flow control, implemented for the presented input-buffer architecture, routing algorithm, arbitration, distribution and packet commutation in the routers. With this sort of transmission, a high degree of channel utilization and throughput is achieved.*

Keywords: *Cut-Through Flow Control, DLH Network, Flow Control Digit (flit), MPP Computer*

Date of Submission: 30-08-2017 Date of acceptance: 16-09-2017

I. Introduction

Parallel computers with tens of thousands of nodes are used in many areas because applications on one hand require high speed of execution achieved by parallel work of many nodes, but on the other hand they use only a small part of the continuously increasing number of processor elements of supercomputers [1]. The data transmission between the nodes of MPP systems using static interconnection networks greatly affects their performance and efficiency [2, 3, 4]. The applied solutions for message passing between nodes depend on network topology, architecture of the routers, routing algorithms and the flow-control mechanism [5]. Of significant interest in this aspect is the implementation of effective data transfer in multiprocessor systems based on Double - Loop Hypercube (DLH) network topology [6, 7]. An architecture of high-speed routers for DLH topology is discussed in this paper. This architecture provides minimal delays for exchange of packets between nodes of the parallel computers without deadlock. The main principles and requirements which the solution is based on are the following:

Interconnection network: DLH static hypercube network topology with asynchronous communication and decentralized control. The routers determine paths for data packets on the basis of given communication functions. We assume that each hypercube in a DLH network has 256 nodes. This means that the router in each node must have 12 full-duplex channels: 8 channels for the hypercube, one channel for its ring, two channels for ring direction and one channel to its own node.

Routing: The routing must be minimally adaptive, allowing selection of one among several paths with minimum length [8, 5]. Local and time conditions on nodes also influence the choice of path in each router.

Packet transfer: Each message is divided into packets before being sent. Packets, in turn, consist of flits (flow control digits). Each packet consists of a header, body and tail and contains: one header flit, zero, one or several body flits and one tail flit. The information in the header flit determines its path [8]. The format of the packet is not a subject of analysis for the router. The transmission by one router of different packets at the same time is not multiplexed onto a single physical channel. The flits are transmitted through the output channel in one clock cycle.

Packet's route: The route of a packet consists of an ordered sequence of resources, which can be queues, communication elements and communication links. The destination address in the header flit, the routing algorithm and the current state of the network define the further route of the packet. After receiving a header flit in some queue of an input channel, the router defines an output channel depending on resource state (the busy buffers and the state of the output channels) using the algorithm for minimal adaptive routing, described in [8]. For the current distribution of input queues to the output channels, the iSLIP algorithm with round-robin arbiters is used [9, 5, 10]. In the case of simultaneous sending of several packets by different nodes in the network, a different route to the destination of each packet is set. These different routes interlace and use common resources. During its travel from the sender to the receiver, each packet dynamically occupies and releases needed resources.

A router architecture and its implementation for static DLH networks are discussed in the paper. In section 2 we introduce pipelined router architecture of three stages. The structure and control of router's input buffer based on pool of eigth FIFO queues, as well as the basic control signals, are presented in section 3. The functioning of the router is based on iSLIP algorithm and output channels arbitration (chapter 4). Hardware implementation for modification of data packets headers during the routing and arbitration process in the router's second stage is given in chapter 5.

II. Router Architecture

The basic router structure consists of a switch, buffers, registers, arbiters, controllers and communication channels. By means of the switch and the buffers, the router connects the input and output channels and in this way forms paths for transmission of data units. The communication technique implemented in a router largely determines the router's architecture [11, 5]. The cut-through flow control mechanism is used here, which allows switching in time and space with a pipelined passing of the packets along their routes from sources to receivers. Each of the input channels of the router has FIFO queues. We further assume that each channel has eight queues.

Figure 1 shows the pipelined version of the basic input-buffered router architecture with cut-through flow control. This architecture provides independent data paths from each queue to the various output channels. The data transmission is implemented by a 3- stage pipeline. These stages are: 1) Flit Write – writing of a flit into a selected input channel's queue. 2) Routing, Arbitration and Path Setting – forming a data path through the switch. 3) Switch Traversal – passing through the switch. The header flit of the packet must pass all stages and take a path from the input channel's FIFO queue through the switch to the selected output channel. All other flits are passed through the first and third pipeline stages. The second pipeline stage can be divided into two different stages: Routing and Arbitration, and Path Setting. In practice, however, this does not matter, because for the chosen technique, packet delay is limited by the delay of the header flit, which is passed through all pipeline stages. The tail flit of the packet releases the resources used for the routing of the packet once it has passed through.

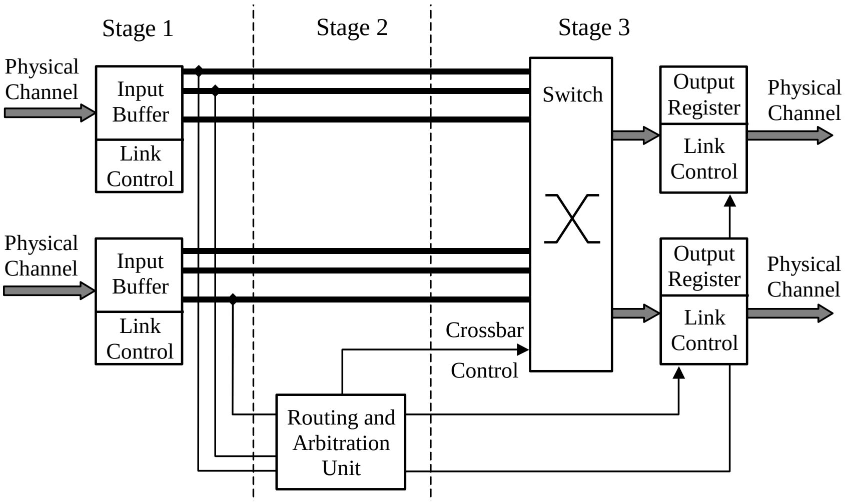

Fig. 1. Pipelined version of a basic input-buffered router with cut-through flow control

III. Input Buffer

Figure 2 shows the architecture of an input buffer of a router with cut-through technology for switching [12], which can be used to form data paths of the packets in a DLH network. Some signals for explaining how the buffer works are shown. This is described in detail below.

The following principles and decisions are used in this architecture: 1) The input buffer is based on a pool of eight FIFO queues. Each queue, whose structure is also shown in figure 3, has its own separate local arbiter. The queue can be used to store only a single packet of a length not greater than the number of its storage elements. 2) The number of FIFO queues in the input buffer does not depend directly on the number of output channels of the router (Fig. 2 shows a buffer with eight queues). 3) The signals for flow-control relate to packets, not to flits. 4) Each of the queues has two states: busy and free indicated by flag B. The state of one input channel depending on the states of the queues in the pool of its buffer can be: a) busy (CHAN_BUSY = 1), when all queues are busy; b) heavily loaded (CHAN_LOAD = 1), when only one or two queues are free; c) lightly loaded (CHAN_LOAD = 0), when more than two queues are free. The state of the input channel is transmitted to the previously visited router via the signals CHAN_BUSY, CHAN_LOAD and PACK_WAIT. 5) The states of the queues are saved in the control automata of each of them (not discussed here) and are fixed into the register B_FIFO_STATUS via the signal WR_B_RG. The PRIORITY ENCODER block always selects the first available queue. The queue can receive each packet that arrives on the corresponding link into the input buffer, no matter which output channel it is directed to. Starting to receive a packet, the corresponding queue changes its state to busy (FIFO_BUSY=1). 6) Each output channel has its own round-robin arbiter (Fig. 4) which allows balanced servicing of the input queues. Thus, conflicts and competitions may happen only for output channels and high performance is assured. 7) The FIFO queues are directly connected to the cross-bar switch of the router (Fig. 1), which makes it possible to send more than one pack-et from one input buffer to the output channels in a same interval of time. 8) Once a header flit has been received by a FIFO queue, it sends request to the local arbiter, which starts routing and arbitration phase for the corresponding packet.

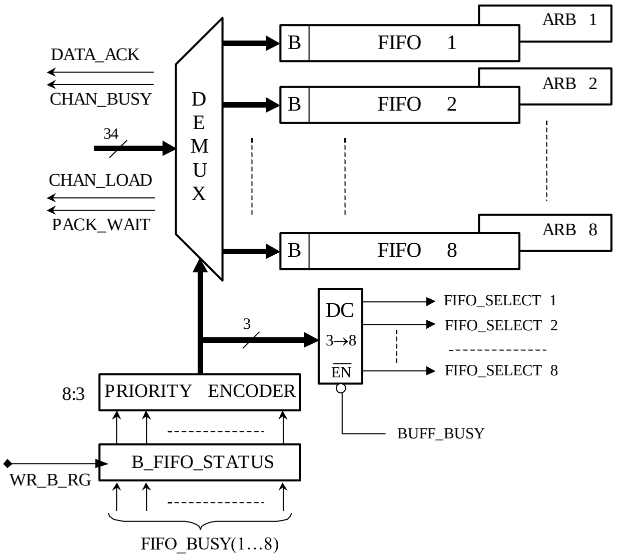

Fig. 2. Architecture of an input buffer with 8 queues

FIFO queue: The basic structure of an input buffer's FIFO queue consists of a demultiplexer, a circular buffer, a multiplexer and a control block. The width of the input buffer is 34 bits [D33 .. D0]. The width of each flit is 32 bits [D31 .. D0]. Bits D33 and D32 encode a type of the flit – Header, Body or Tail. The signals EXT_CLK, VALID_DATA, PACK_WAIT, CHAN_BUSY and DATA_ACK, shown in Fig. 2 and Fig. 3, are used for packet transfer synchronization. The queue is designed to work with only one packet with a maximum size (L+2) flits. The transfer of a packet starts when there is a free and selected queue (PACK_WAIT=1) and when both sides of the transfer are ready after negotiation via the aforementioned synchronization signals [13]. The whole packet is accepted without breaks and interruptions between flits unless a hardware problem occurs.

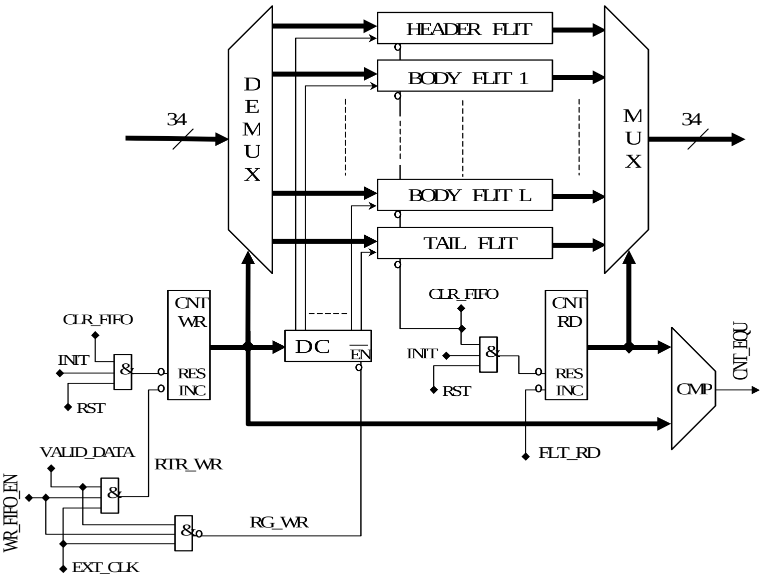

Fig. 3. Architecture of an input buffer FIFO queue

Immediately after receiving a header flit, the queue sends a request to its arbiter to start a phase for routing and arbitration. The FIFO queue allows reading and writing flits simultaneously. A flit is written using the external signal EXT_CLK with the help of the counter CNT_WR and decoder DC. Reading is done using the signal FLT_RD and counter CNT_RD. The signal CNT_EQU indicates an empty queue and is set after reading the last flit.

IV. Routing and Arbitration

Figure 4 shows a block diagram of the router's allocator. For clarity, the demultiplexers and the queues of the input channels are also shown in Fig. 4, although they are not a part of the allocator. The allocator uses the iSLIP algorithm and minimal adaptive routing. Every input queue and every output channel has local arbiters. Once a FIFO queue accepts the header flit of a packet, the phase for routing and arbitration starts. It consists of three steps: Request-Grant-Accept (RGA). At the beginning of a time interval, each of the arbiters of the input queues, which are not involved in matching, can send one or more requests to the round-robin arbiters (RRAs) of the output channels. The number of requests depends on the possible routes along which the incoming packet to the queue can continue its path to its destination. Each of the output RRAs (if its channel is not busy and has at least one request) selects the request with highest priority according to its ring priority pointer PTR and returns a grant to its sender. If an input arbiter receives one or more acknowledgements, it selects the one with the highest priority via logic for fixed priority based on the local conditions of the network (the state of the input channels of the next neighbouring routers) and responds with an accept to the corresponding RRA. An input/output pair of a FIFO queue and an output channel is formed. The PTR of the selected RRA is reinitialised.

Through the switch of the router (not shown in Fig. 4) the data path for the packet transmission from selected queue to the output channel is formed. After the transmission of the whole packet, the resources are released and can be used in subsequent allocations.

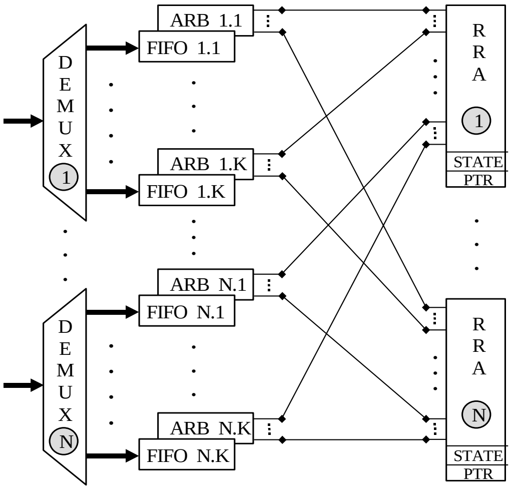

Fig. 4. Block diagram of the router's allocator and its arbiters

It is important to emphasize that all arbiters work in parallel. Conflicts and competitions may arise only at output channels. This is one of the factors for providing maximum bandwidth through the router. For one time interval, the iSLIP algorithm performs one iteration. That is enough, since in the technology for cut-through flow control, the routing and arbitration phase is done only for the header flit of the packet and therefore the number of requests to the output channels is much smaller com-pared to the Wormhole (WH) communication technique. Depending on the current state of the input queues, the output channels and the content of pointer PTR in RRA, in a current time interval the allocator may perform zero, one or more matches. After several iterations and realized matches, the round priority pointers in the output RRAs are desynchronized [14]. This leads to better results of the comparisons during the next iterations and the algorithm converges to a distribution with a high throughput

V. Packet Transfer

The format of the header flit of a packet which forms the route of the packet is shown in figure 5. It consists of the following fields: 1) Destination address DEST_ADR, which consists of three parts: EI [D20] – external or internal ring of the DL network, DLNetwork [D19 .. D8] – destination address of a hypercube in the DL network, BCH [D7 .. D0] – destination address of a node in the hypercube of the DLH network. 2) dT [D31

.. D21] – field for mismatch divided into three parts, LOOP [D31] – indicator for a reached ring in the DL network, LR [D30 .. D29] – direction of moving, indicator for reached hypercube in DL network, DIF_BCH [D28 .. D21] – indicator for the reached node in the target hypercube of DLH network. 3) TF [D33 .. D32]: flit type – Header, Body or Tail.

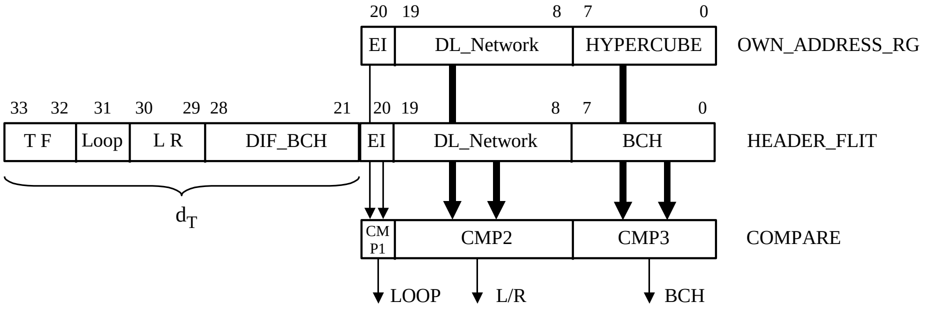

Fig. 5. Format of the packet header flit and comparison between own current node and node destination addresses

Immediately after taking a header flit into an input FIFO queue, the content of its field DEST_ADR containing the destination address is compared with the contents of a special register OWN_ADDRESS_RG of the router. During the initialization of the router, this register is set with the address of the DLH network's node which is served by the router.

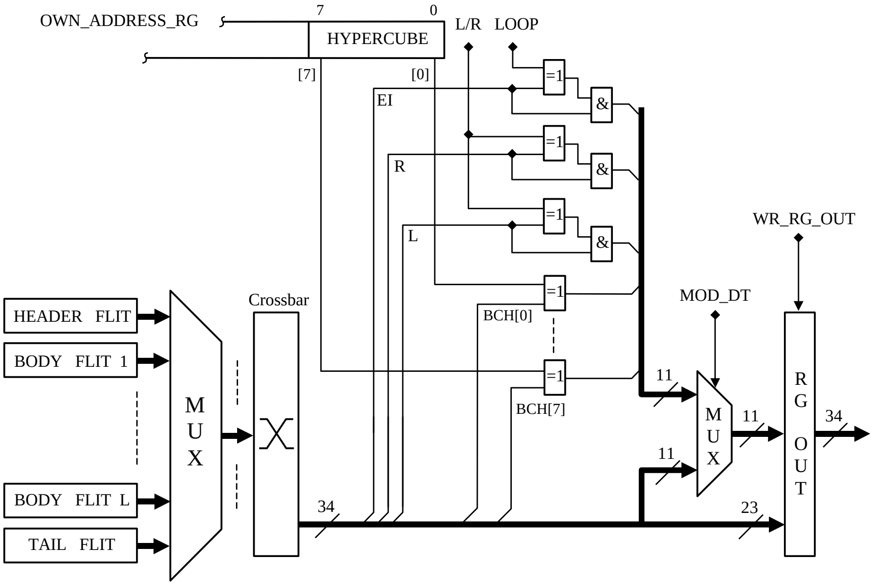

Fig. 6. Scheme for modification of mismatch field dT

The comparison is done by three comparators, one for each of the three parts of the destination address, described above. The comparators generate a three-bit result, which is then analyzed and used to form the path of the packet. A high level of the bits LOOP, L/R and BCH indicates equality of the corresponding fields compared: 1) (DEST_ADR = OWN_ADR) means that the current node is the target of the packet. The active queue of the input channel sends a request to the output channel of its own node to receive the packet. 2) If (DEST_ADR ≠ OWN_ADR) the RRAs of the output channels are sent as many requests as there are high bits in the mismatch field dT. After forming a route for the packet and establishing a path through the switch, dT is modified depending on the selected output channel before the header flit to be sent to the next router. This modification scheme is shown in figure 6. When the header flit passes through the established route, its dT field is modified using the contents of the HYPERCUBE field of OWN_ADDRESS_RG, some XOR and AND logic and a multiplexer. The signals that control the flit's modification are: LOOP – the result of the comparison of the current and destination ring of the DL network, L/R – the result of the comparison of the current and destination hypercube in the DL network, BCH [7..0] – the destination address of the hypercube in the DLH network, MOD_DT – control of the multiplexer from the automaton of the output RRA. All other flits of the packet are passed without modification. Each fixed flit is made available for receiving by the next router by the signal WR_RG_OUT in register RG_OUT of one output channel [1].

VI. Conclusion

The proposed approach for data transfer is suitable for parallel computers with DLH network topologies. It is based on a minimal adaptive routing algorithm, specific assumptions about the router architecture, and the chosen flow-control mechanism. Its main advantages are:

1. Full connectivity between FIFO queues of the input channels and output ports is implemented, which allows 100% utilization of the output channels.
2. The header flit of a packet travels through a minimal number of stages in a router, thus delays of the packets along their routes are minimized.
3. Absence of deadlocks.
4. A high degree of channel utilization while transferring one or several packets in a particular direction is achieved. The time an input queue spends switching between two packets is a small fraction of the transfer time.

References

[1] TOP500 Supercomputer Sites. http://www.top500.org

[2] M. Pütz. Cray Aries Interconnect. Superior scalability and performance, ZKI Workshop 2014, Kaiserlautern

[3] C. Minkenberg Interconnection Network Architectures for High Performance Computing, Advanced Computer Networks, 2013

[4] H Miyazaki., Y. Kusano, N. Shinjou, F. Shoji, M. Yokokawa, T. Watanabe. Overview of the K computer System, *FUJITSU Sci. Tech. J., 48 (3)*, 2012, 255–265

[5] W. Dally, B. Towles. *Principles and Practices of Interconnection Networks* (Morgan Kaufmann Press, San Francisco, 2004)

[6] Y. Liu, J. Han, H. Du. A Hypercube-based Scalable Interconnection Network for Massively Parallel Computing, *Journal of Computers, 3 (10),* 2008, 58-65

[7] M. Ahamad, M. Husain Literature Review: Convey the Data in Massive Parallel Computing, *International Journal of Innovative Research in Information Security (IJIRIS) ISSN: 2349-7017, 2 (9),* November 2015

[8] M. Angelov. Packet Routing in MPP Computers with DLH Network Topologies, *Computer Science and Technologies, ISSN: 1312-3335, 12 (1)*, 2014, 64-69,(http://cs.tu-varna.bg/images/spisanie_knt/cse_journal_1_2014.pdf).

[9] M. Angelov. Router Packet Arbitration in MPP Computers with DLH Network Topologies, *Computer Science and Technologies, ISSN: 1312-333513, 13 (1)*, 2015, 38-45. (http://cs.tu-varna.bg/images/spisanie_knt/cse_journal_1_2015.pdf)

[10] J. Duato, S. Yalamanchili, L. *Ni Interconnection networks. An Engineering Approach* (Morgan Kaufmann, San Francisco, CA, 2003)

[11] M. Angelov. Routers for MPP Computers, Using Direct Communications Networks, *Proc. Of International Conference AUTOMATICS AND INFORMATICS'2014, 2014*, Sofia, Bulgaria, ISSN 1313-1869

[12] M. Angelov: Structure and Control of Buffer for Router Input Channel for MPP Computers, *Computer Science and Technologies ISSN 1312-3335,12 (1)*, 2014, 71-76 (http://cs.tu-varna.bg/images/spisanie_knt/cse_journal_1_2014.pdf).

[13] M. Angelov. A Variant for Cut-Through Flow Control in a Router for MPP Computers, *Computer Science and Technologies, ISSN: 1312-333513, 13 (1)*, 2015, 46-54. (http://cs.tu-varna.bg/images/spisanie_knt/cse_journal_1_2015.pdf)

[14] N. McKeown, T. E. Anderson, A Quantitative Comparison of Scheduling Algorithms for Input Queued Switches, *Computer Network and ISDN Systems, 30 (24)*, 1998, 2309–2326

Milen Angelov . "Packet Transfer in DLH Networks ." IOSR Journal of Computer Engineering (IOSR-JCE) , vol. 19, no. 5, 2017, pp. 08–13.