

PBuf: An On-Chip Packet Transfer Engine for MONARCH

Rashed Zafar Bhatti

Information Sciences Institute
University of Southern California
Marina Del Rey, CA 90292 USA
bhatti@isi.edu

Craig Steele

Exogi LLC
Flagstaff, AZ USA
steele@exogi.com

Jeff Draper

Information Sciences Institute
University of Southern California
Marina Del Rey, CA 90292 USA
draper@isi.edu

Abstract—This paper describes the architecture and implementation of an on-chip packet interface/router called the Packet Buffer (PBuf) employed in the MORphable Networked microARCHitecture (MONARCH). This work provides a brief overview of MONARCH and its subsystems to provide motivation for the PBuf design. MONARCH employs a hierarchy with various levels of address spaces. To connect the subsystems and keep the network complexities low, communication packets undergo an address translation process while passing across the address space boundaries. The PBuf provides protected translation in the midst of superior and inferior address spaces while also serving as an on-chip packet switching router. Additional features, such as its 6 memory to memory block transfer (MMBT) engines, enable it to provide high rate data transfer capabilities.

I. INTRODUCTION

Polymorphic computing architectures [1], [2], [3] represent a class of computing architectures and processing systems aiming to revolutionize implementations of embedded computing systems to configure for different granularities and kinds of parallelism. MORphable Networked micro-ARCHitecture (MONARCH) [4], a polymorphous computing architecture, targets stream programming models like those supported by the Imagine Processor [5] and also achieves conventional threaded level parallelism like TRIPS [1]. To combine these two very different computing paradigms on a multi-core platform requires a unique interconnection network for data, command and control communication. To fulfill this unique requirement, MONARCH employs a dual bidirectional ring network topology, with 256-bit wide channels, for its advantages of high bandwidth, area efficiency and scalability. The hierarchical interconnection network of MONARCH is composed of two types of ring networks: (1) a node ring that connects node elements within a centrally arbitrated network of a local 4GB address space and (2) a chip-level ring that connects the twelve heterogeneous nodes with a 128GB global address space for each node.

Each PBuf serves as a bridge component between its localized node ring and the chip-level ring. As such, the PBuf performs many critical functions more than just on-chip routing between nodes; it also provides a protected user-level communication mechanism. The PBuf contains many features that serve this purpose very well. The initial IBM Cu-08 MONARCH implementation of the PBuf router provides a throughput of over 40 GB/s in a modest area.

Sections II and III give a brief overview of the MONARCH architecture and a description of the MONARCH node components and node interconnect. Sections IV and V present the design and implementation details of the PBuf, and Section VI concludes the paper.

II. MONARCH OVERVIEW

The MONARCH architecture distinguishes itself from other Polymorphous Computing Architecture (PCA) systems by unifying two radically different architecture types into a single flexible VLSI device. The DIVA PIM [6] architecture and High Performance Processing System (HPPS) architecture are combined in MONARCH to support threaded as well as stream processing applications. The Field Programmable Compute Array (FPCA), the internal computation engine of HPPS, was developed to perform efficient stream processing. By modifying basic computational structures in the FPCA to support Wide Word [8] functionality, the arithmetic and memory clusters of FPCA can “morph” between stream and thread modes. MONARCH has also adapted HPPS high bandwidth I/O and fault tolerance features to facilitate sensor input as well as to enable tiling of multiple chips. Fig. 1 gives an architectural view of MONARCH. Memory clusters and arithmetic units grouped into core tiles of the FPCA shown in Fig. 1 are used in streaming based applications. Six of these core tiles around the periphery of the FPCA are coupled with attached RISC processors to support WideWord threaded mode control. It is the morphability of the arithmetic cluster that it can be configured either as a streaming engine or threaded WideWord engine that primarily makes MONARCH a polymorphous architecture. There are several special nodes,

as shown Fig. 1, some of which provide access to external devices such as external DRAM and Rapid I/O interfaces. MONARCH implements two types of on-chip point-to-point packet switched ring networks, namely (1) Node Ring Network and (2) PIRX Ring Networks. The chip-level PIRX ring connects the twelve heterogeneous nodes and is composed of two sets of two unidirectional rings. While not imposed by the hardware, one set may be used for threaded data traffic and the other for streaming data traffic.

III. MONARCH NODE OVERVIEW

The basic computing node on a MONARCH chip consists of an interconnection of four components, namely a RISC threaded processor, Embedded DRAM (EDRAM), Packet Buffer (PBuf) and Array Node Bus Interface (ANBI). There is one specialized master node that contains additional components; its details are beyond the scope of this paper. The PBuf is a memory-mapped on-chip network router that supports packet-switched communication and is the primary topic of this paper. The ANBI connects a core tile of the FPCA to the node ring, providing a port for streams to make memory-mapped accesses. The control signals required for mapping of WideWord ALU functionality onto an FPCA core-tile for thread level parallelism are provided by the RISC processor on the node. The MONARCH node thread processor is largely derived from the DIVA [6] PIM processor model [7] and thus supports single-issue, in-order execution, with 32-bit instructions and 32-bit addresses. In contrast to the dedicated WideWord Unit implemented in DIVA [8], the arithmetic cluster is a morphable unit that can be configured to operate independently as a streaming engine or under control of the threaded execution unit as a wide threaded processor. The RISC processor also contains a small 4KB instruction cache (IC) to keep instruction accesses to the memory from interfering with data accesses. A segment-based Address Translation Unit (ATU) [9] for converting virtual to physical addresses is also incorporated into the threaded processor.

The node components are connected by a node ring network, where each channel supports 256 data bits and a 27-bit WideWord address (where a WideWord is defined as 256 bits of data). The node ring is composed of two counter-rotating point-to-point rings. The counter-rotating rings allow low-latency, high-bandwidth transactions for reads and writes between adjacent nodes. The ring runs at the system clock rate and supports a command/address & data protocol.

The node ring supports three types of transactions: read requests, read replies, and writes. Read transactions are split into separate read request and read reply transactions to allow other activity on the interconnect while waiting for returned read data. Read request transactions include a block size field to allow the requesting device to request a target return more than one WideWord's of data for reduced latency. The primary characteristics of the protocol employed by the MONARCH node ring network are as following:

A. Network Arbitration

Source devices make a request to and acquire a grant from a central arbiter to send a command packet. The arbiter ensures a reserved buffer space at the target node device. The reply transaction generated by the target device does not require any arbitration; the requesting device ensures target buffer spaces for expected replies.

B. Flow Control

All node packets complete a traversal from a node device router to the adjacent node device router in a single cycle, and the rings are never stalled. Packets on the node ring have priority over packets waiting for insertion into the node ring. Packets who have acquired a grant from the target device arbiter must wait for a node ring idle cycle to get injected onto the ring. Each device on the node ring is responsible for keeping the node ring operational by either passing through transactions or inserting transactions or idle cycles onto the ring.

C. Routing

All devices follow their respective routing functions. To inject a packet, a source selects a direction on the ring based on the target device. Once a packet is on a ring, it continues in the same direction until it reaches its target; it cannot be switched to the counter-rotating ring. It is the device's responsibility to maintain the order of transactions to which it is the target, e.g., a read following a write to the same address would return the newly written data.

D. Address Space

The 27-bit WideWord address of the node rings allows each device to have an independent address space of 4 GB with a unique 4-bit node ring device ID. This address space is called the inferior address space when viewed from the superior address space of the chip-level PIRX network.

E. Deadlock Prevention

Each node device always guarantees a priority channel

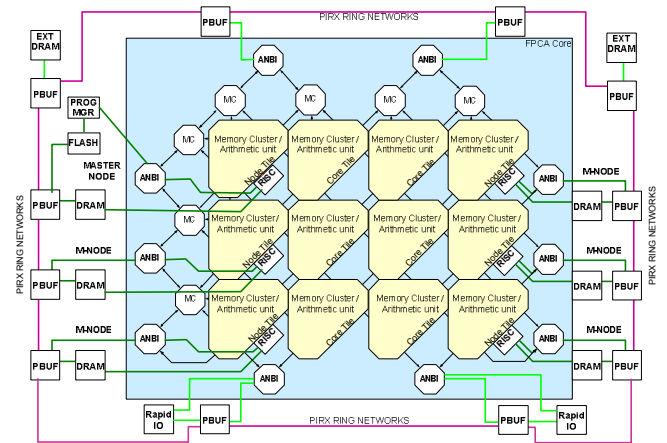


Figure 1. Architectural block diagram of MONARCH

for reply packets to bypass any number of command packets already in the node ring interface queues. Since the replies do not have to arbitrate for a target buffer, they can be injected into the ring as soon as an idle cycle is available on the ring.

IV. PBUF DESIGN

The PBuf is perhaps the most critical component of the entire MONARCH system. It provides a high-bandwidth on-chip communication fabric for command, address and data communication. In addition to just switching packets between a node ring and the chip-level PIRX ring, the PBuf performs many additional tasks like (1) broadcast communication to quickly load the EDRAMs with the programs, (2) address translation for packet moving from inferior to superior address space, (3) implementing system and user level memory access protection, (4) maintaining several memory mapped system level Control and Status Registers (CSR), (5) monitoring system and network performances, (6) communicating occurrence of special events to desired parts of the system through doorbells and (7) high data rate Memory to Memory Block Transfer (MMBT). Fig. 2 shows an implementation block diagram of the PBuf. This section first describes the PIRX portion of the PBuf, which provides the chip-level routing function, and then emphasizes some of the more eminent features of the PBuf.

A. MONARCH PIRX Ring Network

The Processing-In-Memory (PIM) Routing Component (PiRC) [10] is extended to meet the on-chip network communication of MONARCH and is named PIRX. The PIRX router supports two copies of a high-performance wide data ring network to support command/address & data communication across the entire chip. One ring, the streaming network, is shown in the Fig. 2 as a shadow of the other ring, the threaded network. These rings are two identical networks. The remainder of this paper simply describes one of the networks, but it's important to remember that the PIRX actually provides two copies.

For each network, the PIRX implements four unidirectional point-to-point one-dimensional routed network channels, with identical signals and protocols. The four unidirectional channels of each of a collection of PIRX devices are connected to produce a bidirectional ring that connects the MONARCH nodes. PIRX network signals are generally identical to the node ring signals with the exception that the PIRX also supports a broadcast type of packet. The following are primary features of the each PIRX network:

1) Credit Based Flow Control Scheme

The PIRX arbitration scheme uses multiple circulating credit rings for target buffer arbitration that are counter-rotating with respect to the direction of the associated data ring. This avoids the long random wires that would result from simply replicating the node ring central arbitration

scheme to extend the same concept to a chip-spanning scale. An independent flow-control operation of the two PIRX networks is always maintained over two distinct rings, i.e. threaded and streaming. That is, flow-control credits are always associated with a particular ring, threaded or streaming. After system reset each PBuf issues a number of credits on its credit rings equal to the number of buffer slots it has in its target (input) FIFO's. Initially half of the credits are sent out in each direction. No initiating PBuf injects a packet into a ring without having previously acquired a target buffer credit from the flow-control credit ring for that target PBuf. This condition is applied with no exception even to the reply packets. The sum of the number of target (input) buffer credits stored in the associated target PBuf credit counter, the number of credits circulating on the associated flow-control ring, the number of credits grabbed by all possible initiator PBufs, and the number of data packets in flight destined for a specific target buffer pool is an invariant equal to the number of buffers in that target buffer pool.

2) Deadlock Prevention

The PBuf implements independent target buffer

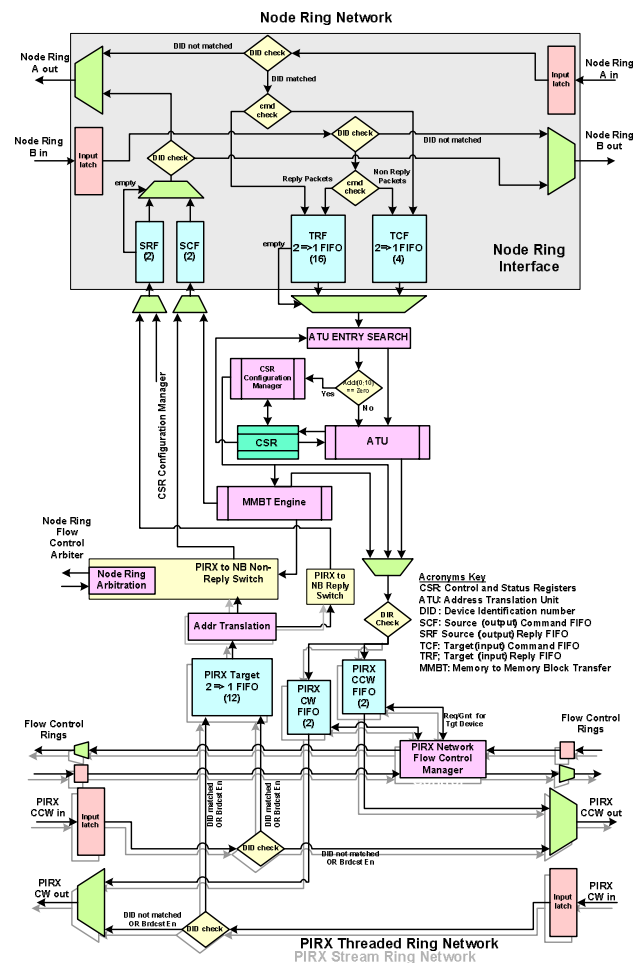


Figure 2. PBUF implementation block diagram

allocation pools and associated flow control for both threaded and streaming PIRX networks. Reply packets get priority over command packets when routed from an inferior address space (node rings network) to one of the superior address spaces (PIRX network) through the PBuf. This prevents any possible blocking of reply packets and thus prevents deadlock. Similarly reply packets moving from superior address spaces (PIRX) to inferior address spaces (node ring) get priority over command packets.

3) Address Space

Each threaded and streaming PIRX ring operates with a 32-bit address space. This makes a 128 GB superior address space for each MONARCH node as viewed from the PIRX Network. Each PBuf connected through the PIRX network has a unique 4-bit PIRX ID which is used for point-to-point packet routing and flow control arbitration.

4) Routing

An ascending packet from a node ring to the PIRX network selects a target PIRX ring and target PIRX device as a part of its address translation. Packets in a ring have priority over packets which may be awaiting injection into the ring from an attached PBuf. The PIRX rings are non-blocking and packets are propagated either forward along the ring or extracted to the target PBuf on each clock cycle. Packets in transit in a ring are propagated without delays until they are extracted from the ring at their target destination by the router, when it detects a match of the target ID with the PIRX router ID. Descending packets from the PIRX ring to a node ring use part of their address to select the target node ring device and arbitrate according to the node ring protocol described above.

5) Broadcast

A broadcast mechanism is implemented to facilitate a fast system initialization through the efficient duplication of boot-time code to multiple node EDRAMs. At boot-time a “broadcast enable” signal in each PBuf enables its associated PIRX to receive and copy broadcast packets that do not match its PIRX router ID. The broadcast packet is taken off the ring by the target PBuf whose PIRX router ID appears in the ID field of the broadcast packet. A received broadcast packet is converted to a write command packet and passed onto the node ring.

B. PBuf Interface Features

The remaining paragraphs of this section describe some of the more interesting features of the interface capability of the PBuf.

1) Address Translation Mechanism

The inferior address space of the PBuf can be mapped separately for read/write and broadcast command (non-reply) packets and reply packets ascending from inferior address space to superior address space. The PBuf address translation unit (ATU) implements a number of translation entry registers for non-reply type of packets and a separate set of ATU entry registers for reply packets. Through these ATU entry registers an address window of an inferior address

space can be mapped to an address window of a superior address space. The size of a non-reply window can be set in a range of 256B to 2GB, whereas reply windows are kept of fixed sizes. The fixed size reply windows simplify the address translation process for reply packets. The ATU entries carry all necessary fields that specify address window mapping for address translation purposes along with some additional access protection field. All packets ascending from the inferior address space of a node ring to the superior address of the PIRX network undergo a translation, in which fields of the node packet are used to search for an ATU mapping. In case of an unmapped address or access protection violation the PBuf generates an exception for the local RISC processor on the node. The exception can also be configured to initiate a remote doorbell sent to a remote PBUF via the PIRX; then the remote PBUF generates a doorbell exception for its local RISC. This mechanism allows a master RISC processor in the system to monitor exceptions occurring in remote nodes.

2) Control and Status Registers (CSR)

All PBuf architectural control and status registers including ATU entry registers are mapped to the lower 1MB window of its inferior address. The CSR's include device status registers, exception control registers, watchdog timer for stall monitoring of RISC, doorbell registers, and MMBT transfer control registers etc. These registers can only be accessed through the node ring network. Remote devices can access these registers by mapping the CSR address space of a remote PBuf through their ATU entries.

3) Exception mechanisms

The PBuf generates three kinds of exceptions (1) two types of ATU exceptions, (2) doorbell exception (3) watchdog timer (WDT) exception. The first two types of exceptions are mask-able and can be polled by local or remote device. The watchdog timer is a free running counter that counts when the local RISC processor is stalled and remains at a preset value otherwise. An unmask-able exception is generated when this timer expires. A PBuf can be configured to initiate a remote exception packet for any exception to inform a remote device about the exception.

4) Doorbells Exception Mechanism

A PBuf implements one 64-bit doorbell exception source register, to allow 64 doorbells. Each doorbell is mapped to a distinct inferior address called the doorbell trigger space. Doorbell exceptions are used for notifications to a RISC processor associated with a PBuf, typically by assertion of an exception signal to the directly attached RISC processor or by polling status by a remote processor.

C. MMBT (Mechanism and throttling)

It is obvious from Fig. 1 that various instances of the PBuf are advantageously situated to provide a memory-to-memory block-transfer (MMBT) function, also known as a memory-to-memory direct-memory-access(DMA) controller function. The split-transaction mechanism of a remote read in the PIRX and node ring network protocols requires a

minimum of two packets per read operation, which makes a data “pull” model of computation more costly in terms of bus traffic compared to a “push” model which produces write packets for data transfer. The PBuf location allows it to read from memory on a local node, e.g., EDRAM, or external DRAM, performing the bidirectional communication on a physically smaller and possibly less congested node ring network, and then push the data to its destination via the possibly more congested PIRX or external network, reducing overall network cost. Latency of a read/reply transaction is lower on a local node ring, allowing higher bandwidth of a transfer without deep pipelining of multiple read requests and consequent increased buffering requirements. Locating an MMBT in the PBuf gives a single-design alternative to implementing a similar “push” DMA function in the EDRAM, external DRAM and RapidIO controller interfaces.

Each PBuf implements 6 MMBT engines that can generate read requests concurrently and arbitrate for the associated node ring providing a capability of reading from its inferior address space and writing to its superior address space. Each MMBT engine is capable of moving a strided-address group of WideWords from a device on the inferior address space to the superior address space with a distinct address stride. Each MMBT engine can individually be triggered to start its block transfer function and can be configured to initiate a remote exception to indicate end of transfer. The PBuf implements a fully programmable throttling mechanism that suspends injection of read requests from the MMBT to avoid unnecessary congestion for regular traffic received from the associated PIRX rings.

D. Performance monitoring.

The PBUF contains four independent reset-able counters to record various internal or external events. Various performance related events can be multiplexed to enable the count of each counter. These performance monitoring counters can be employed to monitor all kinds of traffic passing through the associated network. Run-time software can exploit these memory-mapped architectural counters for traffic load balancing of associated networks.

V. IMPLEMENTATION DETAILS OF PBUF

The MONARCH chip is targeted to IBM Cu-08 (90 nm) technology. The technology provides multiple threshold libraries which allows designer to trade off power for speed. The PBuf design presented in this paper was implemented as RTL-level VHDL and synthesized with the standard threshold cell libraries. IBM’s custom designed monolithic register arrays are used for ATU entry registers and large sized target FIFOs. Overall an instance of the PBuf occupies around 1177107 cells of area. The aggregate area occupied by the twelve PBufs needed for one MONARCH chip is less than 4% of the entire MONARCH chip area.

VI. CONCLUSION

This paper briefly describes MONARCH chip’s architecture that integrates two radically different computing paradigms into a single multi-core chip platform. The PBuf component described in this paper is crucial for high-bandwidth protected user-level communication between many components on the MONARCH chip. More than just an on-chip packet router, the PBuf translates packets between multiple address spaces. Its location in the network allows it to perform other important system level functions like MMBT, network performance monitoring, watchdog monitoring and doorbells. Each PBuf provides over 40 GB/s throughput to each of its neighbors, while the area of a PBuf implementation is negligible compared to the entire MONARCH chip.

REFERENCES

- [1] K. Sankaralingam, R. Nagarajan, H. Liu, C. K. Kim, D. Burger, S. W. Keckler, C. R. Moore, “Exploiting ILP, TLP, and DLP Using Polymorphism in the TRIPs Architecture”, International Symposium on Computer Architecture (ISCA), Jun 2003.
- [2] K. Mai, T. Paaske, N. Jayasena, R. Ho, W. Dally, M. Horowitz, “Smart Memories: A Modular Reconfigurable Architecture”, International Conference on Supercomputing, June 2000.
- [3] M. B. Taylor, J Kim, et al, “The Raw Microprocessor: A Computational Fabric for Software Circuits and General Purpose Programs”, IEEE Micro, Mar 2002.
- [4] J. Granacki and M. Vahey, “MONARCH: A Morphable Networked micro-ARCHitecture”, High Performance Embedded Computing Workshop, October 2002.
- [5] Ujval J. Kapasi, William J. Dally, Scott Rixner, John D. Owens, and Bruce Khailany “The Imagine Stream Processor”, Proceedings of the IEEE International Conference on Computer Design, September 2002.
- [6] J. Draper, J. Chame, M. Hall, C. Steele, T. Barrett, J. LaCoss et al, “The Architecture of the DIVA Processing In Memory Chip,” International Conference on Supercomputing, June 2002.
- [7] Jeffrey Draper, Jeff Sondeen, Sumit Mediratta, Ihn Kim, “Implementation of a 32-bit RISC Processor for the Data-Intensive Architecture Processing-In-Memory Chip”, *Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors*, July 2002.
- [8] Jeffrey Draper, Jeff Sondeen, Chang Woo Kang, “Implementation of a 256-bit WideWord Processor for the Data-Intensive Architecture (DIVA) Processing-In-Memory (PIM) Chip”, *Proceedings of the 28th European Solid-State Circuit Conference*, September 2002.
- [9] Herming Chiueh, Jeffrey Draper, Sumit Mediratta, Jeff Sondeen, “The Address Translation Unit of the Data-Intensive Architecture (DIVA) System”, *Proceedings of the 28th European Solid-State Circuit Conference*, September 2002.
- [10] Sumit D. Mediratta, Craig Steele, Jeff Sondeen, Jeffrey Draper, “An Area-efficient and Protected Network Interface for Processing-In-Memory Systems”, To appear at International Symposium on Circuits and Systems, May 2005.
- [11] Yan Zhang, Irwin, M.J., “Power and performance comparison of crossbars and buses as on-chip interconnect structures”, Signals, Systems, and Computers, 1999. Conference Record of the Thirty-Third Asilomar Conference.