

CDMA as a Multiprocessor Interconnect Strategy

Robert H. Bell, Jr. Chang Yong Kang Lizy John Earl E. Swartzlander, Jr.
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX 78712-1084

Abstract

A binary CDMA bus is proposed as a communications interconnect for multiprocessor systems. The binary CDMA bus is a digital bus which incorporates spread-spectrum technology to encode multiple data streams in parallel onto the same physical interconnect. Mean value analysis is used to show that, for resource bandwidth-limited applications, the binary CDMA bus can deliver a throughput speedup over a split-transaction bus as large numbers of processors are interconnected, giving more scalable performance without additional end-to-end physical bus links. By monitoring bus utilization, the binary CDMA bus can be dynamically activated to alleviate contention and queueing delays experienced by a conventional bus. The inherently parallel access of the bus makes it useful for streaming data in DSP applications.

1: Introduction

High-performance servers are often implemented as shared-memory multiprocessor machines because of their flexibility in handling numerous data transactions simultaneously when transactions make use of data in orthogonal address spaces, while also efficiently supporting a coherency protocol when data is shared [1]. The performance of a shared-memory multiprocessor system often depends on the characteristics of the bus interconnect used to share data among the processors and memory modules in the system [1-2]. The ideal case is a fully-interconnected set of processors and memory modules. Interconnection between processors allows cache-to-cache transfers (“interventions”) of modified data without waiting for memory access.

However, the number of links in a fully-interconnected system grows proportionally to the square of the number of nodes [2]. In a VLSI environment using a synchronous bus interconnect, the number of pins and wires necessary to support a fully-connected bus structure becomes prohibitive. In addition, the increased number of connections lowers the overall yield of the interconnection and on-chip bus selection logic, thereby decreasing the overall fault-tolerance of the system [3-4].

When the network becomes something less than fully-interconnected, such as in a ring or cube interconnect, the performance degrades because of the increasing path-length of the communication between nodes [1]. Contention

for fewer available communication points causes longer queueing delays. The bus utilization increases, and throughput suffers.

Contention and queueing delays associated with processor interventions and memory access may prohibit the scalability of a multiprocessor system in which processors and memory share a common bus [1], generating diminishing performance returns as more processors are connected to the same bus structure. At the same time, the growing gap between processor speeds of over a gigahertz and synchronous interconnect speeds of a couple of hundred megahertz or less causes a larger percentage of system performance loss to be attributed to the memory subsystem and interconnect.

There has been some recent interest in using high-bandwidth communications protocols to meet the interconnect needs of processing systems. One such technique is Code-Division Multiple-Access (CDMA). CDMA is a spread-spectrum technique which encodes information prior to transmission onto a communications medium, permitting simultaneous use of the medium by separate information streams. In [5-6], a multiple-valued CDMA scheme is examined as an interconnect for parallel processing systems. In [7], CDMA on a fiber-optic interconnect operating at 10 gigahertz speeds is proposed as a local area network interconnect. In [8], CDMA is investigated as a means of more-fully interconnecting artificial neural network circuits since those circuits are known to suffer from a combinational explosion of nodes. The basic idea of these works is that pin counts and interconnect wiring can be drastically reduced by using CDMA encoding and an appropriate interconnection strategy.

In this paper we describe the binary CDMA bus interconnect and quantify the benefits of the technique for a multiprocessor system. In section 2, CDMA technology is examined and the binary CDMA bus is proposed. In section 3, the binary CDMA bus model and the analytical tool used to quantify its performance are described and performance results are presented. In section 4, adaptive techniques for making best use of the binary CDMA bus are discussed. Section 5 presents some conclusions.

2: CDMA Technology

CDMA technology relies on the principle of codeword-orthogonality, such that when multiple codewords are summed, they do not interfere completely with each other

at every point in time and can be separated without loss of information. Efficient digital codewords are comprised of a series of bits which can be compactly generated by a linear-feedback shift register (LFSR) configured to generate a sequence which is maximal. A maximal sequence is generated when the polynomial represented by the shift-register is primitive, which means that the sequence only repeats every $(2^N - 1)$ clocks, where N is the number of bits in the shift register.

In CDMA communications, the sender modulates the data bit with the specific $(2^N - 1)$ code bits (“chips”) unique to a particular receiver. The modulated bits from different sources are then summed together using a parallel counter. The result ranges from $-(2^N - 1)$ to $(2^N - 1)$ and is transmitted. The signal passes through a circuit which correlates the signal with the codeword at the receiver. If the autocorrelation result is greater than a threshold value, a bit has been detected in the receiver.

CDMA has been widely used for wireless communications. The summed mixture goes through an up-conversion process which translates the frequency to a higher band which is appropriate for the medium. With CDMA on a digital bus interconnect, there is no need for up-conversion or down-conversion, which alleviates circuit design complexity. However, the nature of the summation prevents the result from being purely binary because the summed result can assume various values other than 1 or 0. To address this, [5] uses an analog capacitance bus in which values are represented by the charge amount stored in the capacitor. In [6], the values are represented by a multi-valued logic system. By departing from a purely binary representation additional circuit complexity can result.

2.1: The CDMA Digital Bus

In a digital CDMA bus bits are represented as +1 or -1 after modulation with the LFSR sequence and summation. This implies that $\log_2(N)$ pure information bits require $\log_2(N + 1)$ transmission bits. As an example, if we allow three data sources to modulate onto the bus simultaneously, the representation must be able to accommodate values from -3 to 3, or six values plus 0, instead of 0 to 3, or 4 values. For each data bit represented by the bus, $\log_2(7)$ or three bits represent the bus value. For 255 processors, 9 summer output bits of information can represent the 511 possible summations from -255 to 255. The top half of Figure 1 shows a LFSR modulation scheme. Figure 2 shows a data flow example of mixing and separating two binary data bits (with multi-level output). Two wires could represent the output values of -1, 0 and 1 instead of serializing the sum.

2.2: The Binary CDMA Bus

The binary CDMA bus is shown in Figure 1. It avoids multiple-valued representations by encoding the summation. The summations that are less than 0 are ignored. The summer counts the number of chips that are greater than 0 and transmits the binary equivalent. For 255 processors, only summations from 0 to 255, or 8 equivalent information bits, are needed. At the receiver side the original summation from -255 to 255 is reconstructed since, for transmitted value P , the original summation is $(2P - 255)$. Essentially, an original 0 summation does not contribute to the resulting correlation function. If the original 0 summation was -255 (all processors send a -1 to the summer), a

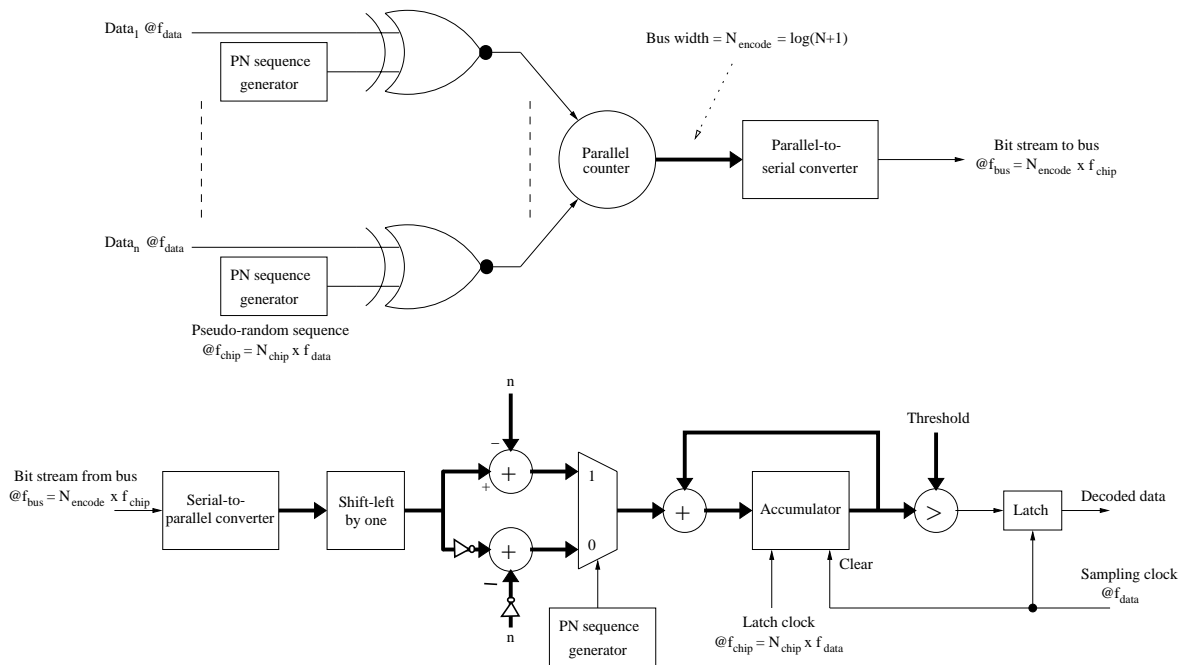


Figure 1: The Binary CDMA Digital Bus

zero is now transmitted to the receiver, and $(2^{*0} - 255)$ yields the original -255. In this example, $\log_2(256)$ or 8 summer bits per data bus bit comprise the actual number of wires in the CDMA bus.

2.3: Binary CDMA Bus Latency and Busy Time

The summer adds an amount of latency to the bus connection proportional to the binary log of the number of sources plus a complexity factor:

$$LAT_{summer} = f + \log_2 N$$

Because of the modulation of the data bit with the CDMA chips, the latency to transmit is a function of the specific code used to modulate the bit:

$$LAT_{chips} = K(N)$$

Since the demodulation takes as long as the number of chips, there is no additional latency associated with the accumulation of bits in the demodulator. Therefore the total latency on the CDMA bus can be represented as:

$$LAT_{CDMA} = K(N) + f + \log_2 N$$

This latency is in addition to any latency seen on a typical bus for a particular system configuration. Assuming no pipelining of demodulation operations, the bus busy time equals the number of chips. Since values placed on the bus must be synchronized to guarantee a specific minimal interference level, an additional bus busy time of approximately half the number of chips is assumed for each source waiting for the bus:

$$BSY_{CDMA} = \frac{3K(N)}{2}$$

2.4: CDMA Codewords

Since the code length is the bandwidth multiplying factor and the number of available codes is the bandwidth dividing factor, it is desirable to have a large number of available codes for a small number of chips. True orthogonal codes, such as the Walsh codes, are not attractive since they offer a relatively small number of codes per chip. The

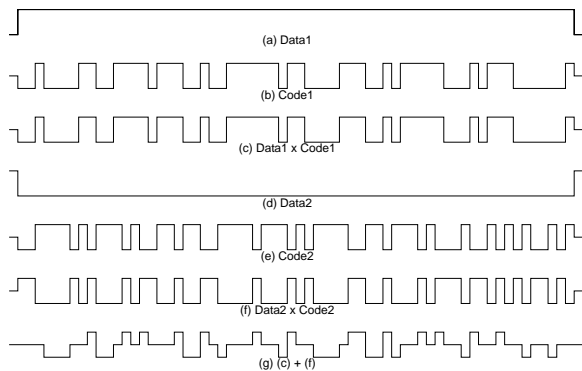


Figure 2: Mixing two binary data bits

codes known as the “Kasami Codes from the Large Set” have several interesting features:

- The degree of the code is a positive even integer.
- The degree must not be a factor of 4.
- For degree n , the length of the code c is $2^n - 1$.
- For degree n , $2^{n/2}(2^n + 1)$ different codes are available.
- The full cross correlation between two codes can be minimized.

It would be optimal for any pair of modulated datawords to exhibit the minimum full cross correlation of -1. Finding the Kasami code offsets is not a trivial task. Also, the minimum cross correlation of -1 is not tolerable because each summed code may contribute -1 miscorrelation and the sums of those miscorrelations may be greater than the true autocorrelation result. To avoid this problem, all the codes are appended with one additional chip of the same value. This offsets the full cross correlation of -1 and makes it 0, at the cost of an extra cycle of latency.

For N equal 64 or more processors, n is equal to 6 and c is equal to 63, plus 1 for the miscorrelation offset gives $K(N)$ equal 64. The LAT_{CDMA} overhead is estimated to be 75 cycles and BSY_{CDMA} is 96 cycles. For fewer than 64 processors, the codeword lengths are equal to the number of processors since the code degree must not be a factor of 4, so the total latency is set to the number of processors, N , plus $\log_2(N)$ for the summer.

3: Performance Analysis

We wish to determine if there are any circumstances in which a multiprocessor system using a binary CDMA bus performs better than a typical split-transaction bus carrying the same number of processors. The base system model shown in Figure 3 is a typical configuration for a high-end data server. It consists of multiple 375 MHz processors and a memory module with a split-transaction bus interconnect. The processors are configured with a 32KB L1 instruction cache, 64KB L1 data cache, and an 8MB off-chip combined data and instruction L2 cache. A 4-cycle bus provides access to memory, which contains 992 interleaved banks. The split-transaction bus is modelled to run three times slower than the processor. Bus address arbitration and a typical MESI coherency protocol are modelled.

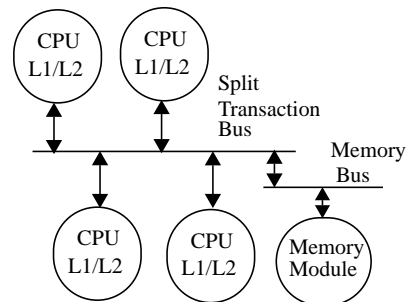


Figure 3: Base Model with Split-Transaction Bus

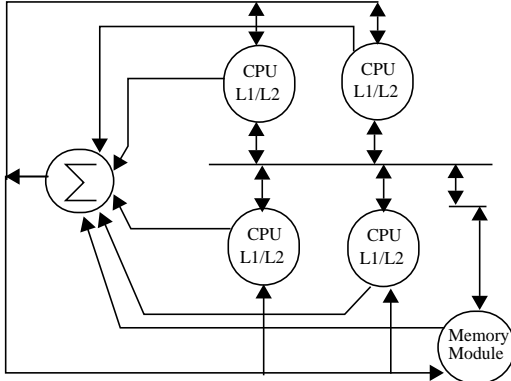


Figure 4: Model with the Binary CDMA Bus

Figure 4 shows the base model augmented with the binary CDMA bus. The sigma represents the CDMA summer, to which the processors and memory module send data after modulating it with the codewords unique to receivers. The summer is not a switch; it is a flow-through parallel counter synchronized for all processors, and therefore it need not be centralized. It can be distributed near groups of processors to reduce wiring costs. Depending on cycle time, the particular configuration modelled may require additional latency. The output of the distributed summer is fed back to all processors and the memory module on a single bus. Only data is sent on the CDMA interconnect. The same bus address arbitration and coherency protocol as the base system are modelled.

There are many possibilities for how a CDMA bus can be used to achieve multiple useful channels. In our model, small sets of memory banks are assumed able to configure a unique codeword for demodulation from any sender. The receiver must know and be able to configure a demodulator for the sender he wishes to receive data from [7]. This knowledge can be transferred in the address phase when the address is requested and granted on the address bus. An index to a codeword can be communicated on the address bus along with the request response. The inherent parallel access of the bus makes it a natural choice for data streaming applications. After orthogonal channels to memory are set up, DSP processors can stream data in parallel to and from separate memory spaces without contention from the other streams. The optimal scheme depends on the application and is a subject for further research.

The model contains both split-transaction and CDMA buses so that hybrid configurations can be studied. In a system with large numbers of processors, not all processors may be able to receive data from all other processors due to silicon area limitations, so most cache-to-cache transfers occur on the split-transaction bus. Since memory is made up of large numbers of DRAM chips, demodulator area costs are a smaller fraction of the total area.

3.1: Mean Value Analysis

Performance comparisons were carried out using an analytical modelling tool similar to the one described in

[1]. The inputs to the tool are the system topology, routing, path information, and instruction miss rates. The topology indicates how the processors and memory are connected to each other using the bus and are modelled as queuing resources. The routing describes probabilistically how processor miss operations move around the topology. The path information give the specific latencies and busy times associated with instruction miss routing. The miss rates include load and store core miss rates per instruction to the memory subsystem, shared and shared-intervention data rates, writeback rates, I/O access rates, and data claim rates. For these experiments the rates used are similar to those expected for a large multiprocessor system [1]. For a typical database workload, data loads may be 30% and stores may be 20% of the instructions respectively, and miss rates to the system interconnect may be less than 5%.

Mean value analysis arrives at memory subsystem cycles-per-core-request probabilistically based on the instruction rates, which determine the intensity of the requests on the various routing paths. The cache request response time is determined from the latencies along the request path multiplied by the probabilities of moving along those paths. These are input into equations for the request response times. The bus utilization is determined from the response times, which is used to determine the request waiting times, which in turn give cycles per request. The cycles per request multiplied by the requests per instruction provides a CPI for the system. Throughput (in MIPS) is determined by dividing the assumed processor clock frequency by the CPI multiplied by the number of processors N in the system:

$$MIPs = \left(\frac{f_{CPU}}{CPI} \cdot N \right) / (1 \times 10^6)$$

3.2: Performance Results

Figure 5 shows MIPS for three systems: a pure split-transaction bus, a hybrid bus allowing up to 32 CDMA intervention connections between processors, and a hybrid bus allowing up to 32 CDMA intervention connections and up to 32 CDMA memory writeback connections to

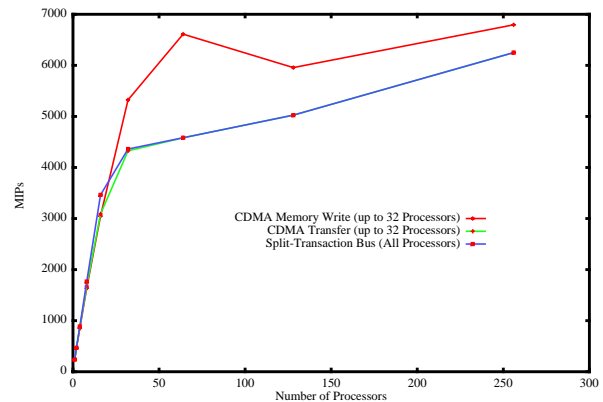


Figure 5: MIPS for up to 32 Processors

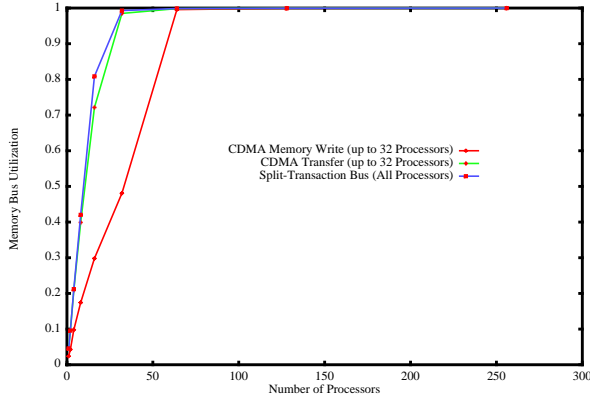


Figure 6: Memory Bus Utilization

memory. The results are similar for 16 or fewer processors, but quickly diverge for 32 or more. This can be understood from Figure 6, which shows that the memory bus utilization is still below 50% when 32 processors use the CDMA bus for memory writebacks. The application is memory-bound since CDMA-based interventions alone never help performance. For CDMA intervention writebacks, the dip in MIPS moving from 64 to 128 processors is due to the average decrease in performance as more non-CDMA bus processors are added. Figure 7 shows the enhanced throughput when up to 256 processors use CDMA for all memory-related operations. Figure 8 shows that the memory bank utilization increasing for 256 processors as the memory bus bottleneck is broken. Again, with 32 processors using CDMA, average utilization decreases as more non-CDMA processors are added.

4: Adaptive Binary CDMA Bus Systems

Figure 7 shows that the binary CDMA bus is most effective when a conventional bus pushes a resource utilization to its limit. Used in conjunction with a conventional bus, it can maximize performance as bus utilization changes. As utilization increases past a threshold, more processors can be dynamically switched to the CDMA bus for intervention and memory access. As the high-latency CDMA bus saturates, no more processors would be switched to it. The cost includes double the wires and added logic complexity.

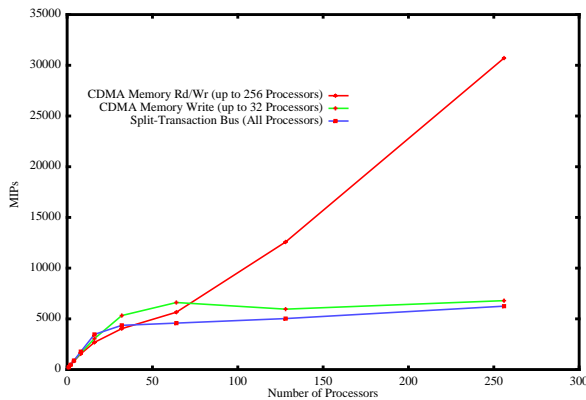


Figure 7: MIPS for up to 256 Processors

5: Conclusions

We described the binary CDMA bus. A multiprocessor system model was developed and the binary CDMA bus performance was compared to that of a split-transaction bus. While contention and queuing delay may saturate the conventional bus, parallel binary CDMA bus access can break the resource bottleneck and improve performance by trading wiring complexity with more silicon area for bit modulation and demodulation.

6: References

- [1] Men-Chow Chiang, Gurinder S. Sohi, "Evaluating Design Choices for Shared Bus Multiprocessors in a Throughput-Oriented Environment," IEEE Transactions on Computers, Vol. 41, No. 3, March 1992, pp. 297-317.
- [2] Earl E. Swartzlander, Jr., VLSI Signal Processing Systems, Kluwer Academic Publishers, Boston, 1986, pp. 160-179.
- [3] Earl E. Swartzlander, Jr., "VLSI, MCM, and WSI: A Design Comparison," IEEE Design and Test of Computers, July-September 1998, pp. 28-34.
- [4] Earl E. Swartzlander, Jr., "A WSI Macrocell Fault Circumvention Strategy," International Conference on WSI, San Francisco, CA, January 29-31, 1991, pp. 90-96.
- [5] Ryuji Yoshimura, Tan B. Koat, Toru Ogawa, Shingo Hatanaka, Toshimasa Matsuoka, Kanji Taniguchi, "DS-CDMA Wired Bus with Simple Interconnection Topology for Parallel Processing system LSIs," International Conference on Parallel Processing, 1985, pp. 370-371.
- [6] Yasushi Yuminaka, Osamu Katoh, Yoshisato Sasaki, Takafumi Aoki, Tatsuo Higuchi, "An Efficient Data Transmission Technique for VLSI Systems Based on Multiple-Valued Code-Division Multiple Access," Proceedings of IEEE International Symposium on Multiple-Valued Logic, 2000, pp. 430-437.
- [7] Paul M. Wexelblat, "An alternative Addressing Scheme for Conventional CDMA Fiber-Optic Networks Allows Interesting Parallel Processing Capabilities," Proceedings of the International Conference on Parallel and Distributed Systems, 1996, pp.248-255.
- [8] J.C. Herrero, "CDMA and TDMA Based Neural Nets," Proceedings of the Sixth Brazilian Symposium on Neural Nets, 2000, pp. 78-83.

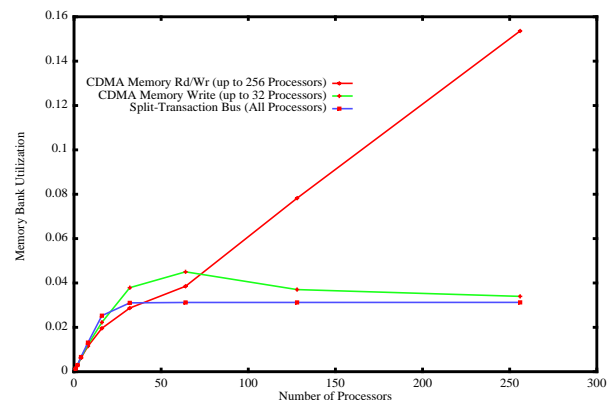


Figure 8: Memory Bank Utilization