THE BAYBRIDGE DQDB MAC CHIP

by

Ting Y. Kao, Frederick L. Burghardt, Wee-Liang Heng,
and My T. Le

Memorandum No. UCB/ERL M92/92

27 August 1992

THE BAYBRIDGE DQDB MAC CHIP

by

Ting Y. Kao, Frederick L. Burghardt, Wee-Liang Heng,
and My T. Le

Memorandum No. UCB/ERL M92/92

27 August 1992

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

THE BAYBRIDGE DQDB MAC CHIP

by

Ting Y. Kao, Frederick L. Burghardt, Wee-Liang Heng,
and My T. Le

Memorandum No. UCB/ERL M92/92

27 August 1992

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

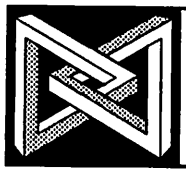
The BayBridge DQDB MAC Chip

Submitted for publication June 1992

Ting Y. Kao
Frederick L. Burghardt
Wee-Liang Heng
My T. Le

Abstract

This paper describes the BayBridge DQDB MAC chip. The MAC chip is a CMOS VLSI device that implements the connectionless Medium Access Control protocol of the Distributed Queue Dual Bus network (IEEE 802.6) at 155Mbps (STS3) with multiple CPE capability. After a short overview of DQDB, we discuss the function and logical implementation of the MAC in detail, followed by a description of the physical implementation including technology, softtools, floorplan, and critical performance issues. Since the chip is a high speed device constructed of standard library subcomponents, speed optimizations were employed to achieve the target performance. We describe these optimizations and their rationale. Finally, we present a short concluding section on design for testability followed by test results on the prototype device.



The Bay Bridge

Project Report: 11 Revision: 2.0
Department of Electrical Engineering
and Computer Sciences
University of California at Berkeley

1 INTRODUCTION

1.1 The BayBridge Project

The BayBridge DQDB Medium Access Control (MAC) chip was built as part of the University of California at Berkeley BayBridge project. The aim of BayBridge is to design and implement a high speed, high throughput encapsulating bridge between Fiber Distributed Data Interface (FDDI), a local area network, and Switched Multi-Megabit Data Service (SMDS), a public network service offered by telecommunication carriers (figure 1) [1].

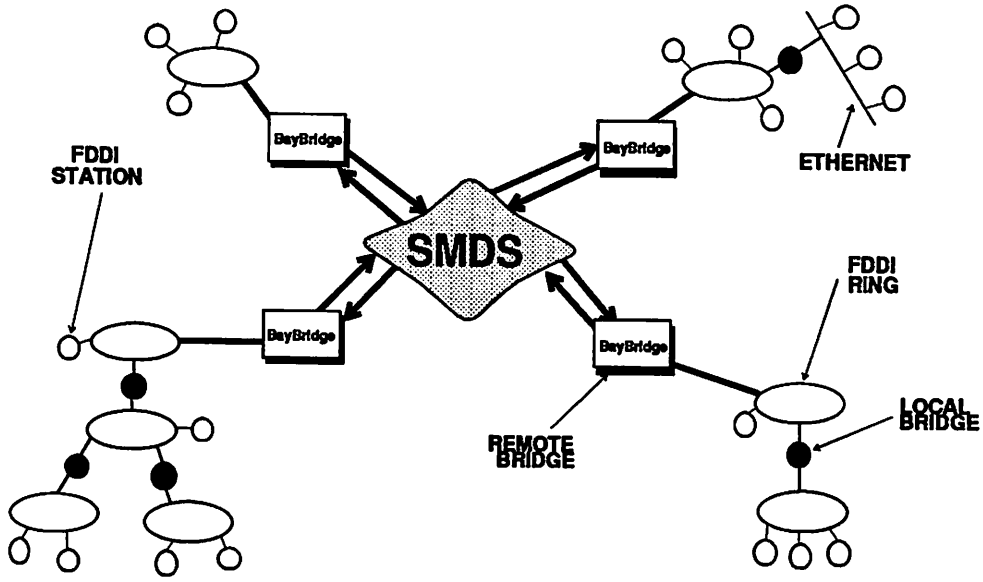


Figure 1: The BayBridge Environment

The BayBridge will interconnect FDDIs via SMDS. The bridge architecture is defined in four major blocks: the Bridge Board, the Host Interface, the FDDI Interface and the SMDS Interface (figure 2). The DQDB MAC chip is part of the SMDS Interface which provides communication between the SMDS public network and the bridge board according to the Distributed Queue Dual Bus (DQDB - IEEE 802.6) [2] protocol. It is a networking device which provides MAC sublayer service to the Logical Link Control sublayer. It is responsible for establishing connections and ensuring error-free data transfer across the channel. The MAC is implemented in accordance with the DQDB protocol in a general DQDB network and will henceforth be discussed in that context [3].

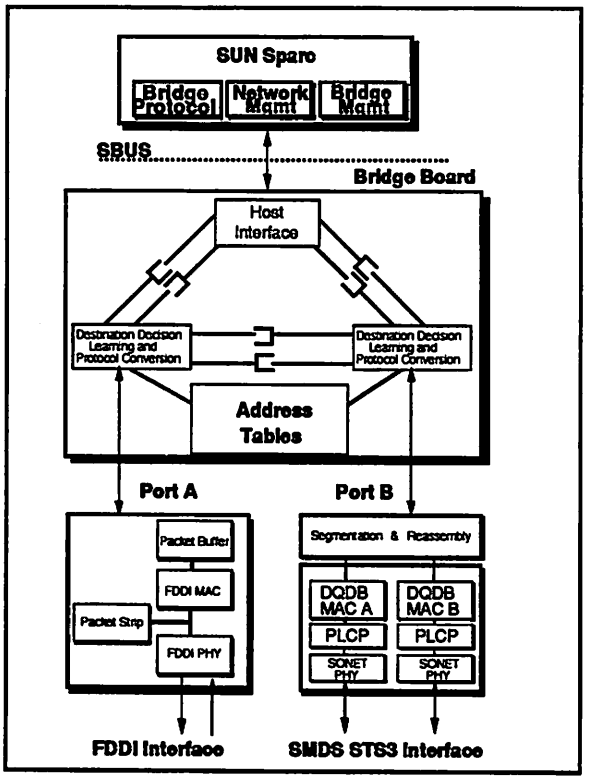


Figure 2: The BayBridge Architecture

The main features of the DQDB MAC Chip include:

- Implements Queued Arbitrated (QA) Functions of the Medium Access Control (MAC) sublayer protocol of the IEEE 802.6 standard (Metropolitan Area Network)
- Operates at STS-3 (155.520Mbps) and DS3 (44.736Mbps) rates
- Implements Bandwidth Balancing protocol
- Allows pre-loading of bandwidth balancing and timeout constants
- Provides on-chip MID lookup table
- Provides Scanning function
- 8-bit interface with Physical Layer
- 32-bit interface with the Segmentation and Reassembly board
- Provides TTL compatible I/O
- Requires a single 5V power supply

Note: For consistency, the term *Message* is used to describe the SMDS Level 3 PDU (Initial MAC PDU in 802.6 terminology) while the term *Cell* refers to the SMDS Level 2 PDU (Derived MAC PDU in 802.6 terminology).

1.2 Definitions

- Cell: The basic DQDB data unit. A cell is 53 bytes in length with a 7-byte header, 44 bytes of payload, and a 2-byte trailer.
- Message: A collection of cells, not necessarily contiguous. A message is segmented into cells or reassembled from cells.
- Message Identifier (MID): A ten bit field in the cell header, identifying the message of which the cell is a part.
- Beginning of Message (BOM): The first cell in a message.
- Continuation of Message (COM): All cells between a BOM and EOM.
- End of Message (EOM): The last cell in a message.
- Single Segment Message (SSM): A single-cell message.

- Access Control Field (ACF): The first byte of a cell, containing queue information and cell status (full or empty).
- Network Control Information (NCI): 4 bytes immediately following ACF, specifying the network configuration. In this implementation, NCI contains a constant specifying a Connectionless, Queue-Arbitrated network.

2 SMDS (DQDB) INTERFACE OVERVIEW

DQDB is a high speed cell based network using a distributed queueing algorithm. It is a collection of nodes connected by two unidirectional buses with opposite directions of data flow; information flows from upstream nodes to downstream nodes. Each bus has a head node that is located upstream of all other nodes on that bus, and produces empty cells that can be used by downstream nodes to send messages. Referring to figure 3, node 2 must use bus A to send messages to node 3, and must use bus B to send messages to node 1.

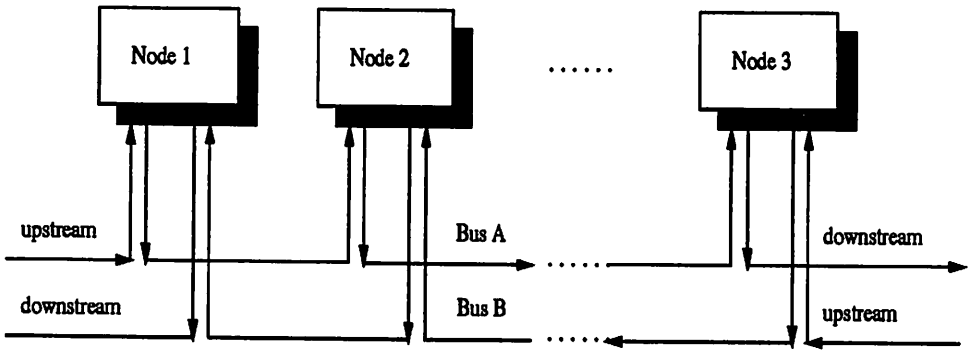


Figure 3: DQDB Network Topology

A node in the DQDB network is comprised of a host and two identical interfaces, one connected to bus A, the other to bus B. The interfaces and buses are symmetric. There are two boards in The BayBridge SMDS interface: a Segmentation and Reassembly (SAR) board and the SMDS Board which includes two DQDB MAC chips with an external Contents Addressable Memory (CAM), and the Physical Layer logic [4].

A cell addressed to a node travels from the physical medium through the Physical Layer where the bit serial stream is converted into bytes. The MAC processes the cell in byte format, verifying destination, removing DQDB header and trailer, testing for errors, and formatting the cell payload into words. The

SAR board accepts the words, reassembles a collection of cells into a message, and passes the message to the host. A message produced by the host is segmented into cells by the SAR board, passed to the MAC where DQDB header and trailer are added, and transmitted to the physical medium via the Physical Layer when all DQDB queue requirements are met.

The MAC interfaces with the SAR board, the PLCP board, a MAC connected to the complementary bus (the **complementary MAC**), and a CAM which stores the network address of the node. These interfaces are shown in (figure 4).

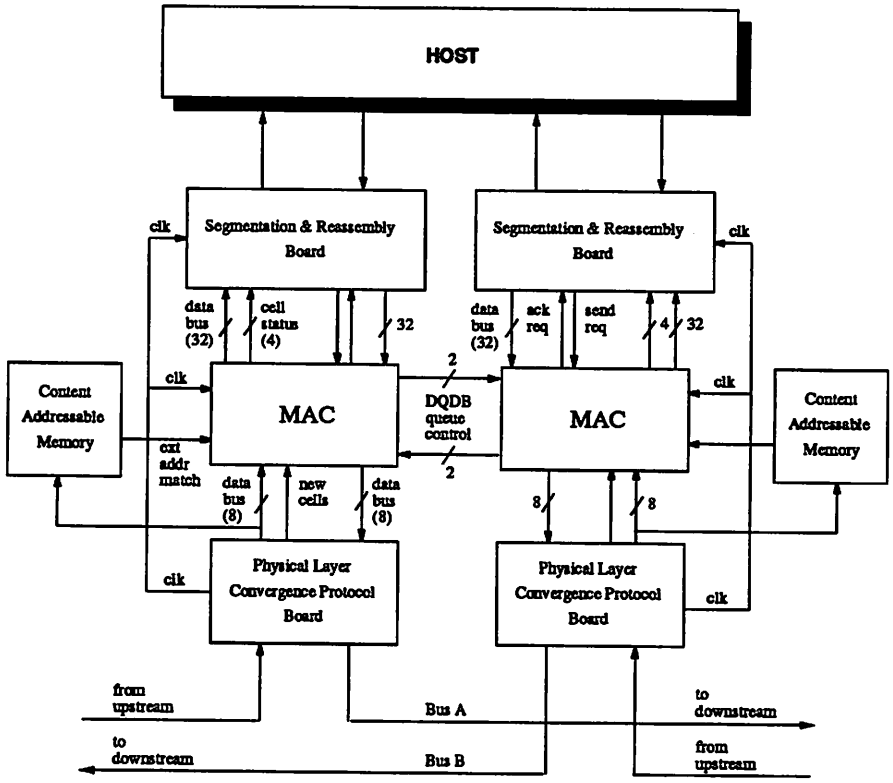


Figure 4: Major Functional Blocks of a Node in a DQDB Network

3 MAC FUNCTIONS

The MAC performs three basic tasks: implementing the DQDB protocol, sending cells, and receiving cells.

3.1 DQDB Protocol Management

The DQDB access protocol determines when an Empty cell is available for sending. This protocol is implemented locally in each MAC: the chip must remember its place in the sending queue. To accomplish this task, the MAC continuously keeps track of the number of downstream MACs on the send queue. When the SAR board signals that a cell is ready for transmission, the MAC *queues* itself by loading an internal counter with the number of downstream MACs currently waiting to send. The MAC then allows the corresponding number of empty cells to pass unused. When the proper number of empty cells have passed, the MAC replaces the next empty cell with a new cell built around data from the SAR board.

In addition to maintaining a count of downstream MACs waiting to send, the MAC must reserve its place in the queue and prevent the upstream MACs from using all empty cells. Because the buses are unidirectional, a MAC can only communicate directly with downstream MACs on its own bus. To send a signal to an upstream MAC, the MAC must use the complementary bus (opposite direction of data flow) in an indirect manner. The communication is accomplished in three steps:

1. The MAC wishing to send asks its complementary MAC (connected to the complementary bus) to set the request bit of a passing cell,
2. Downstream MACs on the complementary bus detect the set request bit, and
3. The downstream MACs assert a signal that tells their respective complementary MACs to queue the request.

The DQDB protocol is not entirely fair since the speed at which a node is able to process a send request is related to the location of the node on the bus if the bus span a distance that is more than the equivalent of one 53-byte cell. A *Bandwidth Balancing Protocol* attempts to correct for this unfairness by allowing the user to specify a constant for controlling the number of consecutive cells a MAC is allowed to send. If a MAC exceeds this number, it must release the next empty cell regardless of send status.

3.2 Sending

When the SAR board has data to send, it asserts a Send Request signal to the MAC. In accordance with the DQDB protocol, the MAC waits for a suitable

Empty cell. Once a cell is available, the MAC asserts an Acknowledge signal to the SAR board and accepts 48 bytes of data which represent a standard 53 byte cell less ACF and NCI fields, and with CRC set to zero. The MAC processes this 48 bytes by prepending ACF and NCI information and replacing the zeros in the Cyclic Redundancy Check (CRC) field with the CRC of the new cell. The Empty cell arriving from the PLCP is replaced by the new cell, and the new cell is transmitted to the PLCP board with a busy status. This process is repeated until the entire message has been sent.

3.3 Receiving

If the MAC is not sending, it returns each cell unchanged to the Physical Layer for downstream retransmission. If the cell is not empty, the MAC *copies* it and checks for transmission errors by performing a CRC. In the meantime, the MAC determines if the cell is intended for the node. In the case of a BOM or SSM cell, the first 8 bytes of payload which indicates the *Destination Address* must match an address stored in the external CAM. For a COM or EOM cell, the MID must be marked valid in the internal MID table which indicates that the cell is a part of a message that is currently being copied by this node. In all cases, an incoming cell is passed to the SAR board along with synchronization signals. If address or MID validation is successful and the cell is error-free, status signals are asserted that instruct the SAR board to accept the cell for reassembly. Otherwise, the data is ignored.

4 LOGICAL IMPLEMENTATION

The MAC chip is designed as a network of interacting blocks, each performing a specific function. Each block consists of one or more controllers implemented as Moore finite state machines and associated data paths made from a combination of registers, counters, shifters, SRAM, and combinational logic.

Four blocks handle essential functions: the DQDB block, the Send block, the Receive block, and the MID Table Management block. The DQDB block implements the DQDB protocol, specifying when an Empty cell can be used for sending. The Send block performs the send function. The Receive block, together with the MID Table Management block, performs the receive function.

4.1 DQDB Block

The DQDB block (figure 5) utilizes four FSMs and three counters to implement the DQDB protocol.

The Request Counter keeps track of the number of Send Requests issued by MACs downstream from the node. The Count Down Counter keeps track of the number of Empty cells the MAC must pass before it can send. When the SAR board issues a send request to the MAC, the DQDB-SAR FSM broadcasts the send request to all upstream nodes by signaling the complementary MAC to set the Request bit of a passing cell. In addition, it initiates the count down process by loading the current value of the Request Counter into the Count Down Counter and clearing the Request Counter.

The Bandwidth Balancing Counter, along with the Bandwidth Balancing constant, implements the Bandwidth Balancing protocol. The Bandwidth Balancing counter keeps track of the number of consecutive cells sent by the MAC while the bandwith balancing constant is set by the user to specify the number of consecutive cells that a node is allowed to send.

The DQDB-QUEUE FSM processes the Request bit of the incoming cell. If the Request bit is set, the FSM signals the complementary MAC to increment its Request Counter (pass an Empty cell unchanged). If the Request bit is not set and the complementary MAC has data to send, the DQDB-QUEUE FSM initiates a procedure to set the Request bit which will cause a cell to be reserved by all downstream nodes.

The DQDB-REQ and DQDB-MAC FSMs provide an interface to the complementary MAC. The DQDB-REQ FSM registers Bandwidth Requests and the DQDB-MAC FSM increments the Request Counter, when signalled by the complementary MAC. Structurally, these two state machines are identical.

4.2 Send Block

The Send block is comprised of the Send FSM and its associated datapath (figure 6). The Send FSM controls the interface between the MAC and the Segmentation section of the SAR board. If an incoming cell will not be used to send data, the Send block multiplexes the cell unaltered from the Receive block directly to the Physical Layer (the Request bit may be set if a Bandwidth Request is pending). If Send conditions are met, the Monitor FSM signals the Send FSM to begin moving data from the SAR board into the From-SAR FIFO shifter.

The wake up signal also serves as an acknowledgment to the SAR board that data is being accepted. The SAR will place new data onto the interface every four clock cycles thereafter. The ACF field of the incoming Empty cell is modified to show a busy status and returned to the Physical Layer, and a hardwired NCI constant is multiplexed into the stream following the modified ACF field. The third multiplexer channel directs remaining data from the From-SAR FIFO to the PLCP board. The new CRC is calculated and inserted while the CRC field is still within the From-SAR FIFO. When the Send block initiates a Send operation, it notifies the DQDB block so that queue status can be maintained.

4.3 Receive Block

The Receive block (figure 7) is implemented with a datapath, assorted logic, and three finite state machines: the Receive FSM, the Send-To-SAR FSM, and the Monitor FSM. A Busy cell is processed by the Receive FSM while an Empty cell is processed by the Monitor FSM. If transmitting a cell is not in progress, the Send-To-SAR FSM passes the contents of the cell along with the synchronization and status signals to the SAR.

Every incoming cell is loaded into the Receive FIFO shifter; the major part of the Receive data path. If the incoming cell is busy, then the Receive FSM performs various tests, including NCI pattern matching and CRC calculation while data is moving through the Receive FIFO shifter. In the case of a BOM or SSM cell, an address match must be performed by the external CAM to determine status before the cell can be passed to the SAR board. The time needed to read the entire Destination Address (8 bytes) from a BOM or SSM cell plus the time required to perform the external address match results in a latency between data reception from the Physical Layer and the initial cell status signals transmitted to the SAR board. This latency defines the depth of the Receive FIFO shifter. Since the Receive FSM is synchronized to the first byte of incoming cells, the latency generated by the address match procedure makes controlling the SAR board interface with the Receive FSM impractical. Therefore, data exchange with the SAR board is performed by the Send-To-SAR FSM. The Receive FSM sends a wake up signal to the Send-To-SAR FSM during processing of every Busy cell. Immediately after wake-up, the Send-To-SAR FSM makes a first determination of cell validity by testing for an address match in the case of a BOM or SSM cell, or a MID table match in the case of an EOM or COM cell, then proceeds to regulate data transfer in 32 bit words from the receive FIFO shifter to the MAC/SAR interface along with synchronization and appropriate status signals.

The Monitor FSM plays a central coordinating role. If an incoming cell is Empty

and there is no data to send, it decrements the Request Counter if the request counter is greater than zero, and loads the Bandwidth Balancing Counter with the Bandwidth Balancing constant. If an incoming cell is Empty and there is data to send but all send conditions are not met, it decrements the Count Down Counter if that counter is greater than zero, and loads the Bandwidth Balancing Counter. If all requirements for a Send operation are met, the Monitor FSM asserts a wake up signal to the send block and returns to an idle state. If an incoming cell is Busy, it performs no function (the Receive FSM is independent).

4.4 MID Table Block

When a message is segmented for transmission under DQDB, only the first cell (BOM) of the message will contain the 8-byte Destination Address due to high storage overhead (18% of payload). The remaining cells of the message are associated with the destination node via the 10-bit MID located in the cell header. In a DQDB network, messages are not guaranteed to arrive in contiguous cells. Therefore, when a BOM cell with a matching network address is received, the MAC must record its MID so that following COM cells and the terminating EOM cell can be associated with that message.

The MID Table block keeps track of active messages. It consists of a MID table, associated state machines, counters, and logic. The MID Table is a 1024x2 SRAM that stores a two bit time stamp at an address formed by the MID. The MID table block performs two functions: MID Table entry update and MID table entry timeout.

The MID Table is shared by the update and timeout functions. Access for updates is performed during the arrival of the first half of the cell, and access for timeouts is performed during the last half.

Figures 8 and 9 show the organization of the update and timeout subblocks, respectively.

4.4.1 MID Table Entry Update

Upon initialization of the MAC, zeros are written into all entries of the MID Table SRAM. When a new cell arrives, the external CAM is employed to test for a Destination Address match. If a match is found, the cell is a BOM and the cell is Busy, the MID Update FSM will apply a non-zero time stamp to the MID Table location addressed by the MID.

For all cells, the MID Update FSM stores the Cell Type and MID and initiates an SRAM read cycle. For COM and EOM cells, if the entry is non-zero, the *MID table valid* signal is asserted to indicate that the MID is active and the cell belongs to a message currently undergoing reassembly by the SAR. Although BOM and SSM cell MID values are tested, the result is ignored because the validity of these cells is determined by the Destination Address. By definition, a SSM cell requires no MID table entry.

When the SRAM read cycle is performed for subsequent cells with the same MID, the *MID table valid* signal will be asserted until the MID entry is invalidated by the timeout algorithm described below.

4.4.2 MID Table Entry Timeout

Since the DQDB protocol allows out of order transmission, the possibility exists that, for a particular message, the EOM cell may arrive before a COM cell. As a consequence, MID Table entries are not invalidated on arrival of the EOM cell; the MID decays independently. The current time stamp is kept in a two bit counter that cycles from one to three. A time out occurs when the difference between the MID Table entry and the current timestamp equals two - the MID Table entry is written with zeros, and the MID becomes invalid. Subsequent cells with the same MID will not be used for reassembly unless the entry is revalidated by a BOM cell.

During processing of a cell, the MID Timeout FSM checks the value of the Timeout Period Counter. If it is not zero, it is decremented and no other action is taken. If it is zero, the FSM loads the Timeout Period Counter with the value of the Timeout Period constant (set by the user in the same way as the Bandwidth Balancing constant), performs a timeout operation on the MID Table entry if the difference between the MID Table value and the Timestamp is two, and increments the MID Table Address Counter. When the value of the MID Table Address Counter is equal to the number of entries in the MID Table (every 1024 cells), the current Timestamp Counter is incremented. Thus, the Timeout Period constant controls the rate at which entries in the MID Table decay.

5 PHYSICAL IMPLEMENTATION

5.1 Technology and Tools

The MAC chip was fabricated in 2.0 micron N-well CMOS through the MOSIS service. The selection of a mature technology was driven primarily by a requirement for first-run success. In addition, the design floorplan was sufficiently compact such that die area was not a concern. Overall die size is approximately *7.5mm x 7.4mm*, and the die is packaged in *132 lead pin grid arrays (PGAs)*. All 132 connections are utilized, although four are non-essential test outputs.

The initial MAC chip design was produced as a class project in VLSI design. The design softtools used in that class were from the Octtools suite, a Berkeley CAD development effort. Octtools offers a full set of capabilities for realizing designs in silicon such as visual editing, standard cell placement, logic minimization, and global routing. As work on the chip continued after the course ended and as the designers gained experience, many of the tasks performed in Octools were transferred to LagerIV [5] - an alternate set of design tools that offers all the capabilities of Octtools with significant improvements in several areas. LagerIV was developed by a group of universities and corporations, and is administered by U.C. Berkeley.

In the final design phases, algorithm implementation, standard cell placement and routing, and floorplanning were accomplished using the Octtools V4.0 programs BDSYN [6], Bdnet, Wolfe, and Bdnet, respectively, with assistance from the visual editor VEM [7]. Global routing, verification, and generation of .ext and .cif were performed with Magic, part of the LagerIV suite. The program Ext2sim was used to produce *.sim* format files, and Irsim was the primary simulation tool. Both Ext2sim and Irsim are from the LagerIV release, although Irsim was developed at Stanford University.

All logic elements except those used in the MID table SRAM are standard cells from the Scalable CMOS (SCMOS) Standard Cell Library V2.2 administered by the Institute for Technology Development, Advanced Microelectronics Division (ITD/AuE), and developed under contract to U.C. Berkeley as library support for LagerIV. The SRAM is also part of the LagerIV distribution.

The pad family is *m2c_2u* developed by Paul Cohen at the Massachusetts Microelectronics Center.

5.2 Layout

The logical organization described in section 4 translates well into several natural physical subblocks (figure 10). The Receive FIFO Shifter and a 32 bit Buffer are placed as a single high aspect ratio block adjacent to the MAC/SAR interface output pads to minimize interconnect. Similarly, the From-SAR Shifter is located near the MAC/SAR input pads. The shape of the From-SAR Shifter was dictated by the adjacent SRAM.

The DQDB, Receive and Send circuitry is combined into a single large block (the MAC subsection) primarily because of the complexity of interconnect and timing, and secondarily because of the convenience of Wolfe in comparison with global routing tools. The MID Table logic is combined into a smaller block (the MID subsection) near the SRAM. The functional division between MID and MAC subsections is so sharp that only a single timing independent signal (*MID table valid*) is necessary for communication between the subsections. It is this inherent division that allows a very high degree of physical independence between the two major logic sections.

Two identical clock drivers are incorporated into a single small block, and the four primary clock signals *(ϕ1, ϕ2, ϕ1n, ϕ2n)* from each are distributed globally. The clock signals are assigned to subsections such that loads were equalized and potential clock skew problems are minimized.

5.3 Standard Cells

The ITD/AuE standard cell library contains a wide variety of logic primitives and memory leaf cells. For simplicity, only two memory cell types are used: a level sensitive D-type flip-flop with reset (larf310) and an infrequently used edge triggered latch (dfrf301). Nearly all counter, register, datapath, and FSM state is built from the larf310 cell because of its small size and relatively low propogation delay (approximately 5nS nominal with 1pF load).

5.4 Performance

One of the primary objectives in the design of the MAC chip was to produce a device capable of processing data at STS3 rate (155.520Mbps). To achieve this speed, the chip must function at a nominal clock rate of 19.44MHz (51.44nS per cycle). Since the clock architecture is non-overlapping two-phase, combinational logic plus state element propogation delay must be less than approximately 20nS when dfrf301s are used and approximately 40nS when larf310s are used because

these level-sensitive memory elements can tolerate instability at the data input throughout most of the *clock asserted* half cycle. The longest critical paths include the large 10-bit counters, where a carry signal must propogate serially. Although serial carry propogation will not yield optimally low delays, use of standard cells makes carry look-ahead schemes expensive in terms of area. To overcome this difficulty, counter logic was hand-designed such that the carry path requires only one gate (3-input NAND or 3-input NOR) per bit. This method yields an overall propogation delay through the carry chain of approximately 28nS. Due to some additional logic and a state element needed to form the counter state, simulations show that the most complex ten bit counter will operate correctly to about 44.8nS per cycle, for a maximum operating speed of approximately 22.3MHz. The counters are the limiting factor for chip performance.

5.5 Optimization

As mentioned above, the critical paths of the chip include the the wide DQDB counters. In addition, the larger FSMs impose a significant delay during output signal generation. Normally, the FSM must generate stable control signals and the counter state must change in response to those control signals within one clock period. It is difficult, however, to design large FSMs and counters (such as the receive FSM and the request counter) which will meet that requirement. To overcome this difficulty, the MAC employs pipelining on critical paths.

In pipelining, a control signal generated by an FSM is latched into a register and the advance of the counter state is delayed until the next clock cycle, at which time the FSM can be generating its next control signal. The throughput is the same as if the operation were performed within one clock cycle, but the data is delayed by one clock cycle. This allows one full cycle each for FSM signal generation and carry propogation in the counters.

Because of constraints imposed by Octtools, state machines were generated using standard cells rather than Programmable Logic Arrays (PLAs). In order for the FSMs to achieve the required speed, large FSMs were subdivided as the design matured to reduce the number of logic layers and improve signal generation. The receive, send, and monitor FSMs were originally implemented as one state machine because all were synchronized to the physical layer. This optimization required a trade off of die area for performance, but the resulting floorplan was well within acceptable limits.

The DQDB counters are implemented as 10 bit counters as opposed to the 16 bit counters specified by the IEEE 802.6 document. Because only 10 bits are allocated for the MID, the maximum number of nodes on the DQDB bus is

limited to 1024. Furthermore, since the SAR board will issue a new send request only when the previous cell has been sent, each node on the bus will contribute only one request to the bus at any time. Consequently, the maximum number of requests on the bus is 1023: only 10 bits are needed for the counters.

6 PROTOTYPE TESTING

6.1 Design for Testability

Because the MAC is a highly sequential device, a large number of state elements are present in the design. To make effective testing possible, a serial scan path was included that passes through all state elements except the major data paths. Each state element is encapsulated into a small scan block with data, scan, and mode inputs, a data output, and a scan output. Several varieties of scan blocks were developed to accomodate level sensitive and edge triggered state elements on either clock phase. These scan blocks are strung together in a serial chain to create the scan path. The scan path is accessed by a single scan input, a single scan output, and a mode input that is common to all scan blocks. The major data paths are not included on the scan path for two reasons: the data paths are themselves scan paths in nature and can be tested independently, and the area expense of adding scan circuitry to the data paths was prohibitive.

When the mode input is high (+5.0V), the scan path is configured as a long shift register (188 bits). When the mode input is low (0V), the scan path is transparent. By scanning in vectors, state machines can be loaded with any state, and counters and registers can be set with any value so that when normal operation resumes very specific conditions are present. The scan path can be switched in or out on the fly.

During simulation, the scan path was not instrumental in MAC development because all internal nodes were accessible via software. However, hardware testing was greatly facilitated by extensive use of the scan path. Virtually 100% testability was achieved, including functional verification of all state machines, registers, counters, SRAM cells, and combinational logic blocks. This high degree of visibility can not be reasonably obtained with a vector set applied to the normal bus interfaces due to the extremely large number of vectors required.

6.2 Test Results

Thirty-two devices were received from MOSIS, and of that number twenty-four (75%) passed all electrical tests at a nominal clock speed of 20MHz (effective STS3). Functional testing was performed with a Tektronix DAS9100 series benchtop test system installed with 91PXX pattern generator modules and 91A32 aquisition modules. A GPIB interface between the DAS and an HP workstation was used to automate large and complicated test procedures requiring many pattern files. Timing tests were performed with an HP16500 digital logic analyzer/oscilloscope. The chip has met all design requirements.

7 Acknowledgements

We thank Professors Pravin Varaiya and Jean Walrand for their invaluable advice and support. Special thanks to Nick McKeown, Richard Edell and Glenn Serre for their knowledge, support and friendship.

References

[1] N. McKeown, R. Edell, M. Le, "Architecture and Performance of The Bay-Bridge: A High Speed Bridge/Router between FDDI and SMDS," Submitted to this issue of JSAC, June 1992

[2] "P802.6 DQDB Subnetwork of a Metropolitan Area Network Draft Standard," *IEEE Project 802 Committee P802.6/D15*, Oct. 1990.

[3] F. L. Burghardt, T. Y. Kao, and M. T. Le, "DQDB MAC Chip Datasheet," *Bay Bridge Project Report 10, Rev 2.0*, Jan. 1992.

[4] M. Le, N. McKeown, R. Edell, "A High Performance SMDS Interface at STS-3c Rate," Submitted to this issue of JSAC, June 1992

[5] "Lager Tool Set," Scalable CMOS Standard Cell Library 2.2, Mar. 1990.

[6] R. Segal, "BDSYN User's Manual," University of California, Berkeley, Mar. 1990.

[7] P. B. Cohen, "Symbolic Editing with VEM," Massachusetts Microelectronics Center, 1989.

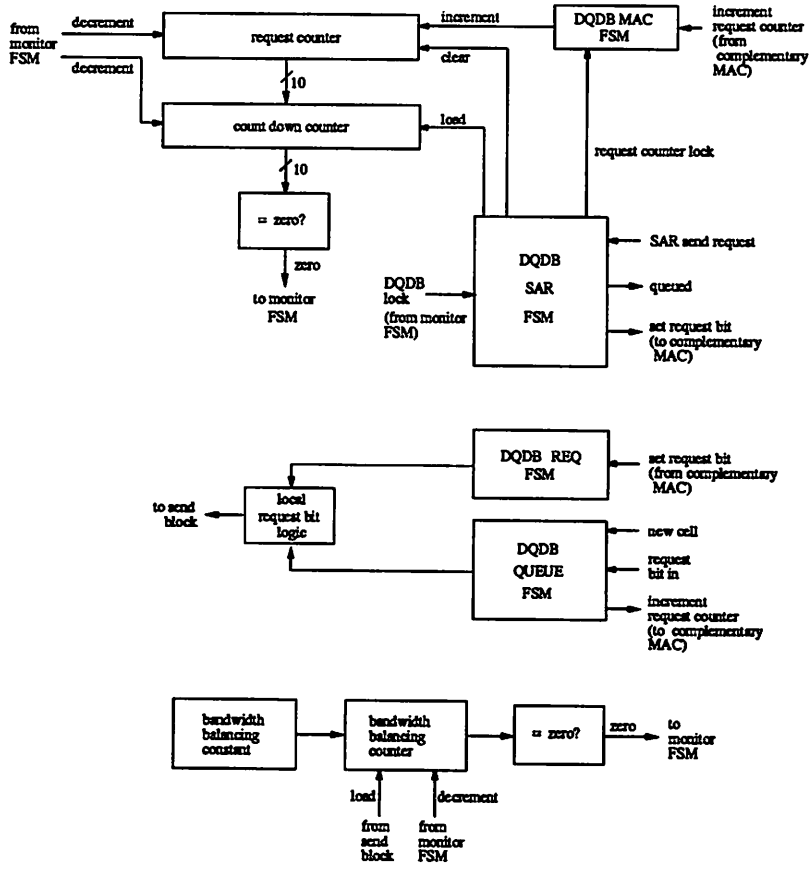


Figure 5: DQDB Block

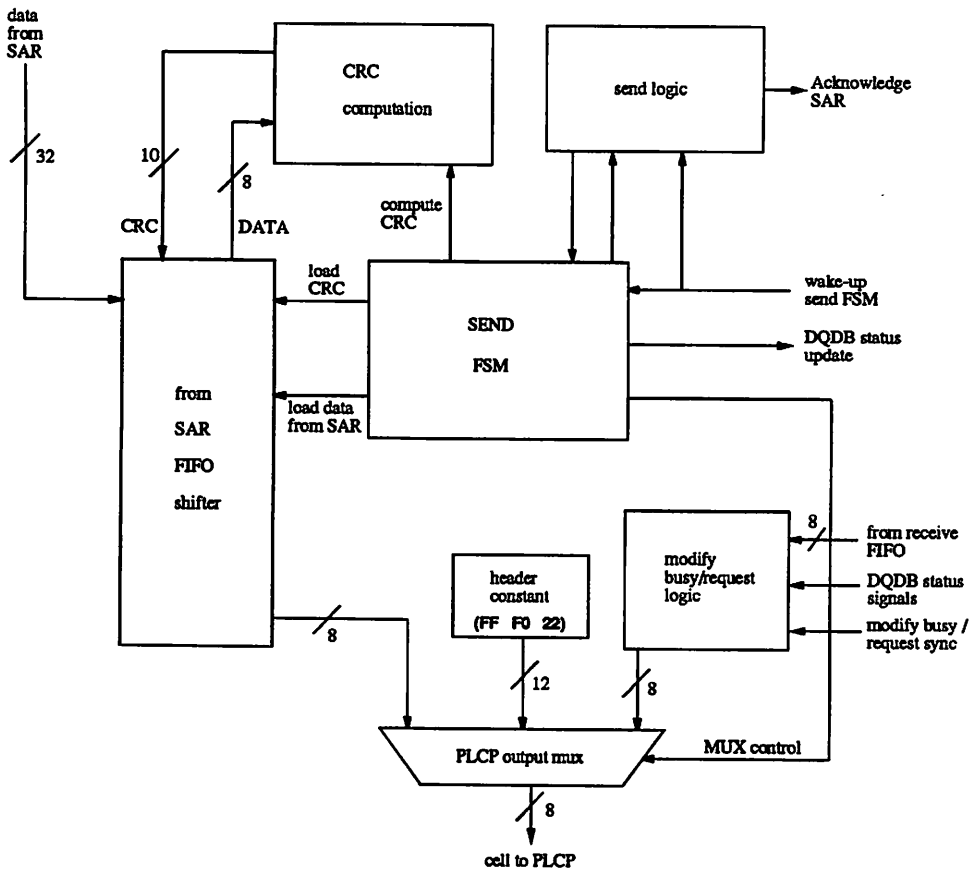


Figure 6: Send Block

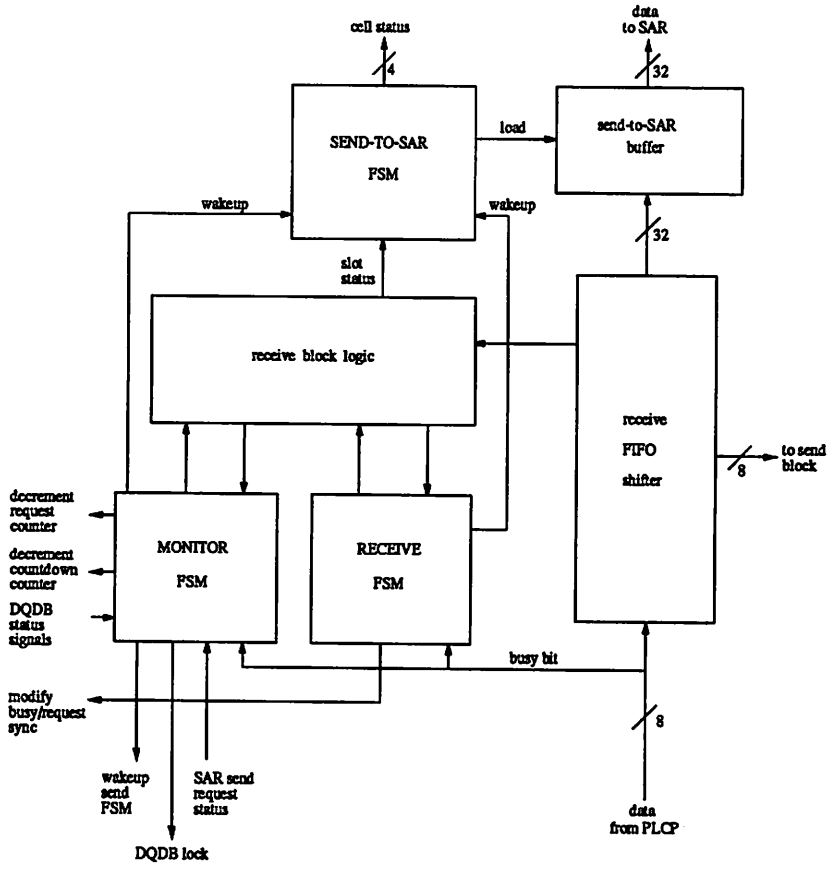


Figure 7: Receive Block

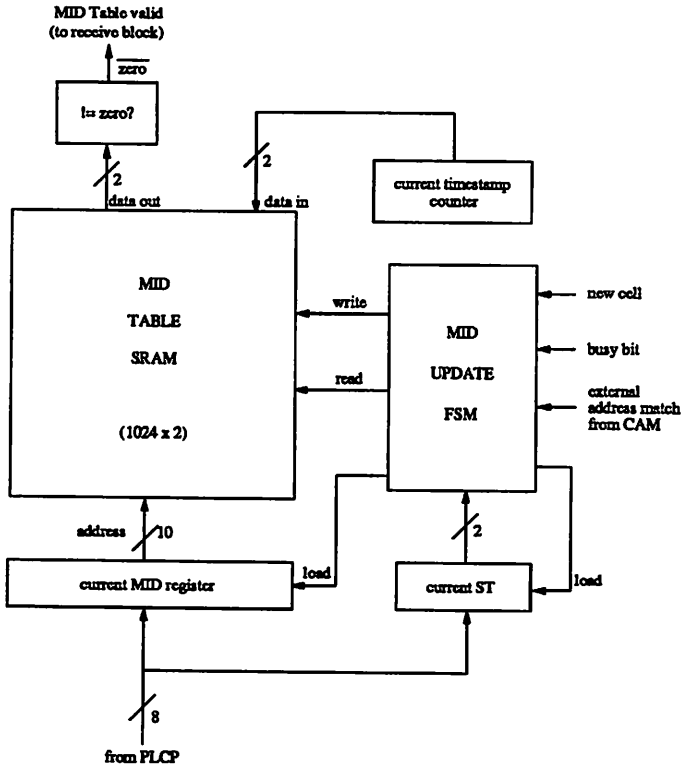


Figure 8: MID Table Entry Update Subblock

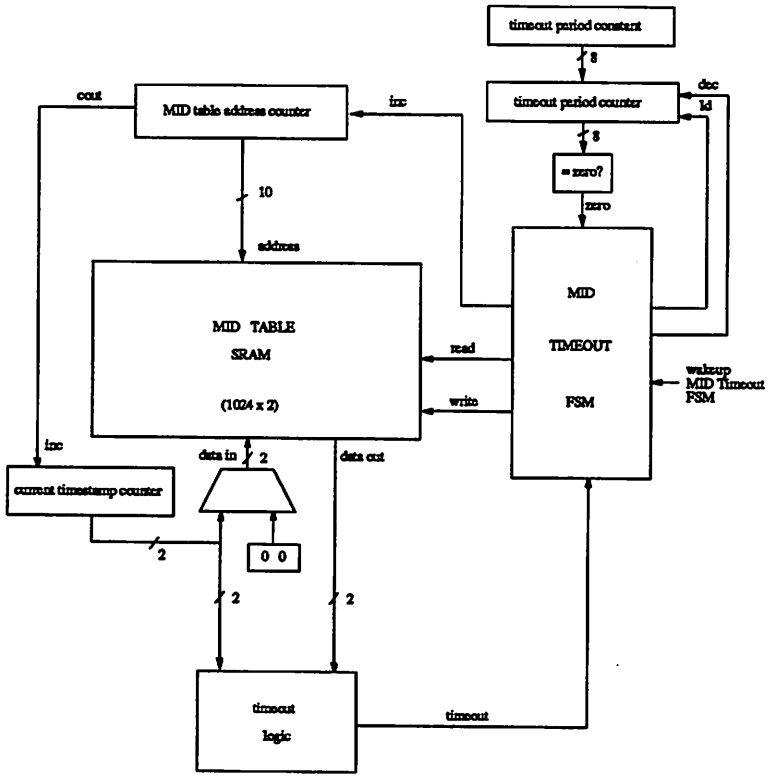


Figure 9: MID Table Entry Timeout Subblock

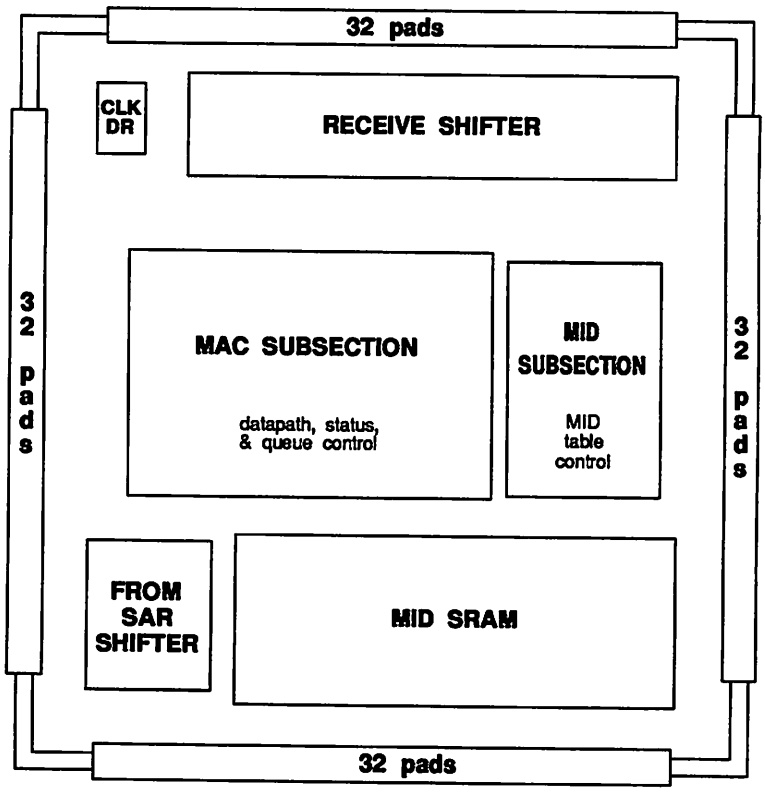


Figure 10: Floorplan