AN1704

Switch Fabric Implementation Using Shared Memory

Prepared by: Lakshmi Mandyam and B. Kinney

INTRODUCTION

Whether it be for the World Wide Web or for an intra–office network, today's businesses and services are heavily dependent on the telecommunications infrastructure. Computers and networks are inextricably linked in business today. Just as processors have increased their speed in instructions per second, users have demanded more bandwidth in megabits to gigabits per second from their networks. Many high–speed technologies have emerged or are emerging including: FDDI, CDDI, Fast Ethernet, Gigabit Ethernet, and Asynchronous Transfer Mode (ATM). Only ATM has the ability to integrate voice, video, and data over a single medium and requires a great deal of high–speed bandwidth. One of the major ways that switch, router, and hub manufacturers have been able to differentiate their products has been in the fabric and buffer management. The shared or distributed cell or frame fabric as well as the input and/or output buffers determine:

1. The scalability of the network.
2. The non–blocking capacity of the network.
3. The throughput experienced by the users.
4. Each user's service levels.

These issues are common across all of the network devices. In the carrier network, ATM is widely used because it reduces the number of network overlays and provides seamless integration of LAN and WAN equipment. The switch fabric buffering scheme is of major importance to the flexibility and adaptability of the network. For the end user, new markets are being developed that require fast efficient buffering schemes; such as multimedia support, bandwidth–on–demand, and high–speed data transfer for internet and access to the corporate headquarters.

For the data LAN networks, Gigabit Ethernet provides an easy migration to gigabit speeds by upgrading the existing legacy Ethernet 10BaseT and 100BaseT systems. The higher throughput and the need to match different speeds on inputs and outputs also requires a great deal of shared–memory fabric and buffering.

SWITCH FABRIC

The function of the common block of memory or shared fabric is to store–then–transfer cells or frames between other functional blocks in the switch/router. Examples of the functions of fabric are:

1. Cell or frame buffering.
2. Traffic concentration and multiplexing.
3. Redundancy for fault tolerance.
4. Support of multi–casting or broadcasting.
5. Cell scheduling based on delay priorities.
6. Selective cell discarding based on loss priorities and congestion level.
7. Congestion monitoring.

There are many types of fabric architectures and choosing one usually depends on where the switch will exist in the network and the amount of traffic it will be required to carry. In reality, fabric implementation is often a combination of two or more basic architectures. Figure 1 illustrates the variety of possible switch fabric architectures.

The type of architecture that is common to the majority of network devices is the shared fabric memory (see Figure 2). Freescale has several products available for shared memory implementation including Late Write, BurstRAM, and a new product, NetRAM. There will be slightly different implementations for each design, but the basic functionality will remain the same.

In shared fabric implementations, fast SRAMs are used to buffer data before transmitting it. The memories are organized in banks to create a shared–memory block of 576 bits wide for ATM and 64 bytes wide for Ethernet. The depth of the fabric will be determined by the bandwidth required of the ports for input and output. With dynamic allocation, the advantage of shared fabric is that it enhances the flexibility of determining network throughput and guarantees that data will not be blocked or dropped as it passes through the switch.

Motorola's new NetRAM, a dual port SRAM with configurable input/output data ports is designed specifically for the communications market to deliver optimal performance for networking devices that demand write/read/write cycles. In this application note, two implementations of switch fabric will be discussed using the NetRAM.

1. The "One–Way" switch fabric implementation with separate input and output ports.
2. The "Snoop" switch fabric implementation with two dual–ports: each capable of supporting input and output.

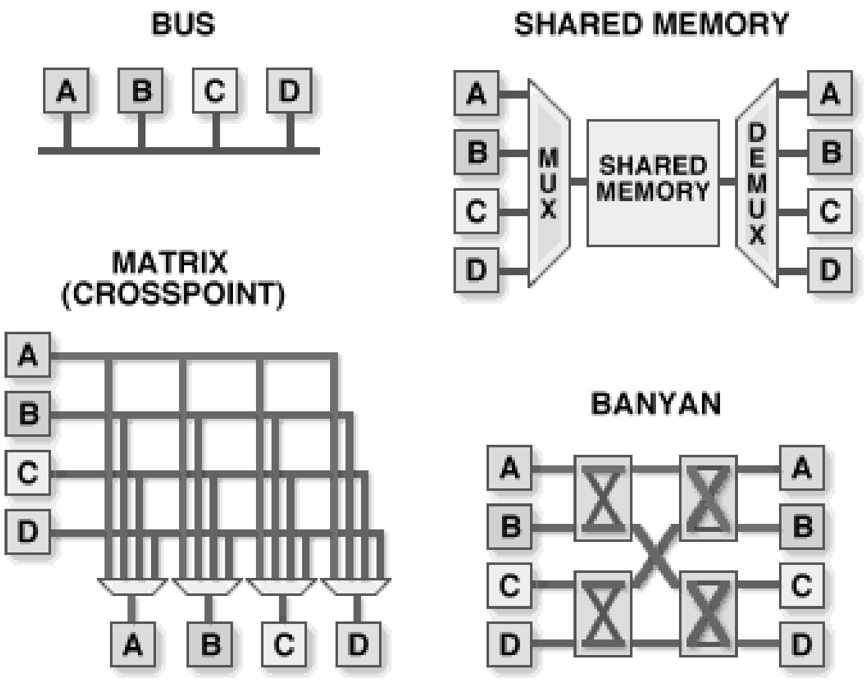

Figure 1. Types of Fabric Architectures

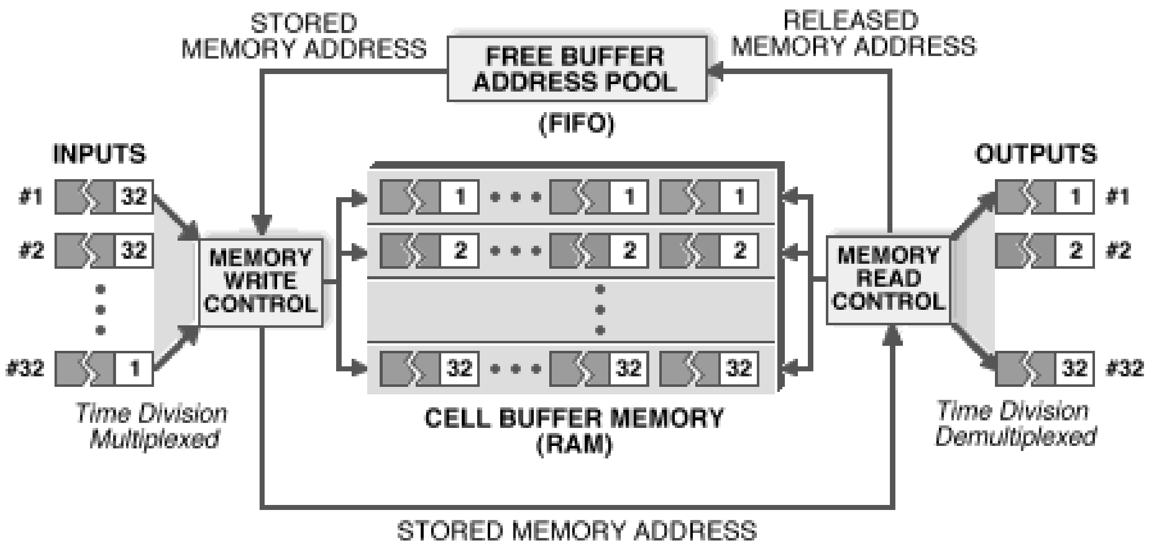

Figure 2. Shared Memory Switch Fabric

"ONE–WAY" SWITCH FABRIC

In the first implementation (Figure 2), an ATM example realizes output buffering and can support multiple qualities of service (different priorities) and multicast/broadcast. Inputs to the common shared memory fabric are time–division multiplexed (TDM) so that only one input port is granted permission to deposit a cell into the shared memory. As the cell traverses the memory write controller, the output port destination is decoded and used to determine which output queue is to receive the cell. A free buffer address is generated to provide a write address for the cell in the fabric. In addition, this address is "linked" to the tail of the output queue managed by the memory read controller.

Cells leaving the fabric are time–division demultiplexed so that only one output port has access to the fabric. Arbitration is usually necessary, because there may be many cells competing for access to the output port. It is the responsibility of the memory read controller to determine which cell wins the arbitration and is forwarded to the output port. Once the cell has been transmitted, the available fabric address is returned to the free fabric–address pool.

An implementation of switch fabric using NetRAM in a "one–way" path is shown in Figure 3. Input modules or line cards are connected to the NetRAM data inputs, and output modules or line cards are connected to the data outputs. The read address port and write address ports are connected to the external controller or application specific integrated circuit (ASIC).

As noted previously noted, the ASIC generates the memory address for the cell to be stored until the destination output port becomes available. The store address is sent to the AY port of NetRAM and the cell is written from the input port (DQY) to the fabric after the write enable signal (WY) is activated. The timing diagram is shown in Figure 4.

When a destination output port for the data cell becomes available, the ASIC recalls the read address of where it had previously stored the cell and sends it to the AX port on NetRAM. After the read is performed, the data is available at the output port (DQX) two internal clock cycles later providing output enable (GX) is activated at that time. Notice that the internal clock is running at twice the external clock. Because of this, board frequencies can be maintained at frequencies equal to or less than 83 MHz while the device delivers performance equivalent to a conventional 166 MHz memory.

NetRAM is a pipelined part with reads occurring before writes. Therefore, if a read and a write occur at the same address on the same clock edge, the read data available from the location will be the cell or frame that was previously in memory and the write data will be written after the read has occurred. Thus, NetRAM resolves potential address contention situations without the need of external arbitration logic or special ASICs to make sure the data is transferred without corruption.

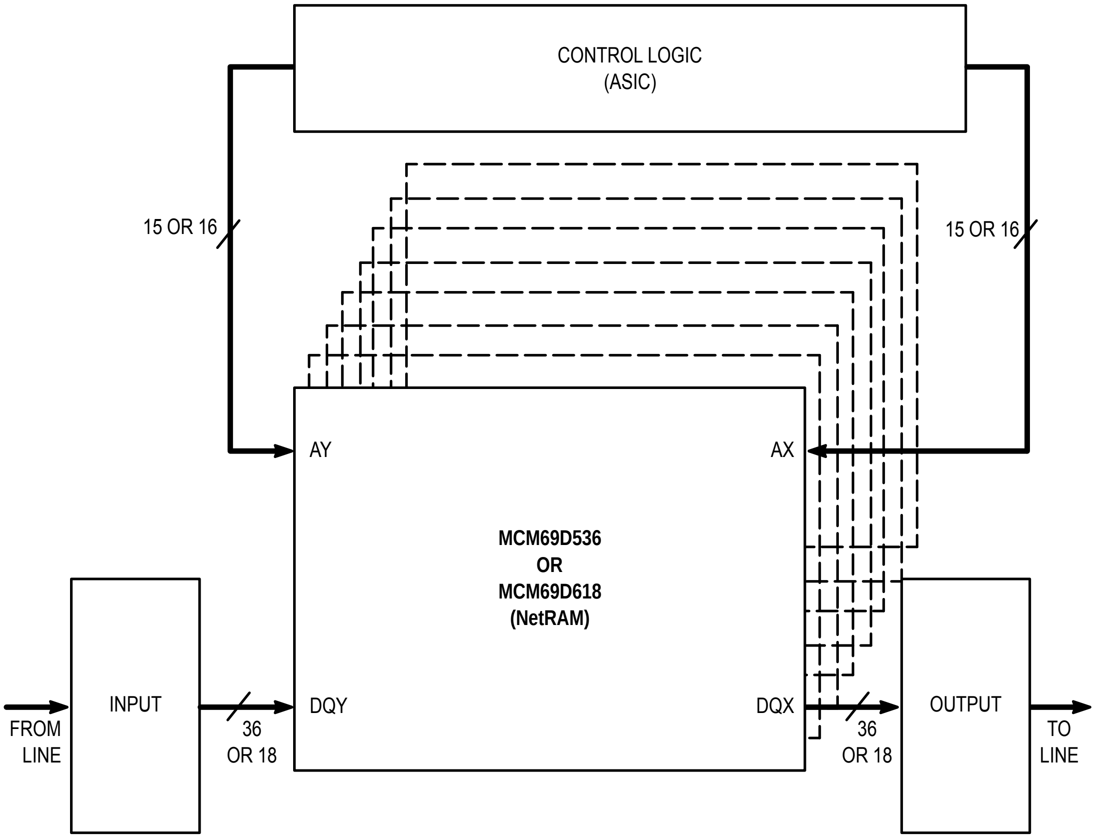

Figure 3. "One–Way" Switch Fabric Implementation Using NetRAM as a Separate I/O Device

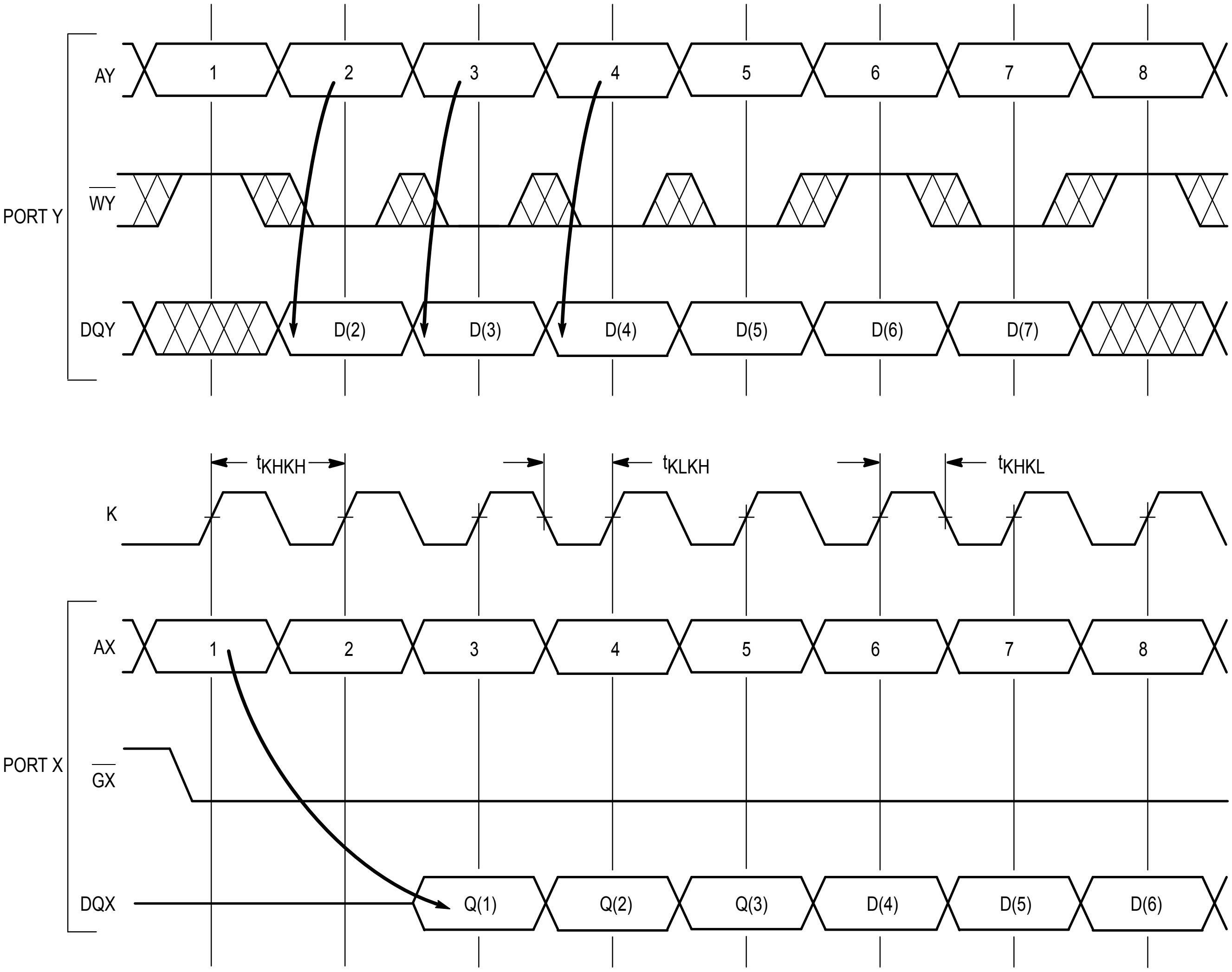

Figure 4. Read/Writing Timing Diagram for “One Way” Switch Fabric

There are additional functions available to the designer. The pass–through function allows the data to be sent from the input to the output, bypassing the memory. Consider cycle t_4: pass–through (PTY) and write enable (WY) are asserted and as a result, data on NetRAM input port Y (DQY) gets written to address A(4) and is passed through to the output at port X (DQX) and becomes available on the output at cycle t_6. In a situation where the system is trying to get data from the input port to the output port as soon as possible, the pass–through function saves two cycles of system time over a RAM without the pass–through function. Without this function, to send data from memory input to memory output, the user would have to wait until data was written to the AY address and then send a read address to the AX address port.

The write–and–pass–through function allows the data to be sent from the input to the output, and the data also is written to memory in the normal fashion. Figure 5 shows both pass–through and write–and–pass–through timing for moving data from port Y to port X.

NetRAM’s advantage over conventional single address and common I/O RAMs is that it allows reads and writes to be performed at different addresses during the same clock cycle. This dual address feature gives the NetRAM a very high throughput of this one–way fabric of 2.98 Gbits/sec which is sufficient to support up to OC–12 throughput. In addition, most fabric implementations bank several memories together in parallel. In the example given, assume that 16 memories are banked together and thus the total available throughput would be 16 x 2.98 Gbits/sec or 47.8 Gbits/sec.

Consider the flexibility of this part if the designer were to double up on the data input and data output**.** The configurable data ports allow the designer to read simultaneously from both ports in the same clock cycle providing a throughput of approximately 6 Gbits/sec with one device. With a 576–bit wide memory block and 16 NetRAMs connected in parallel (Figure 6), the maximum throughput becomes approximately 96 Gbits/sec. This is one of NetRAMs advantages when configured in the “snoop” switch fabric implementation as described in the following section.

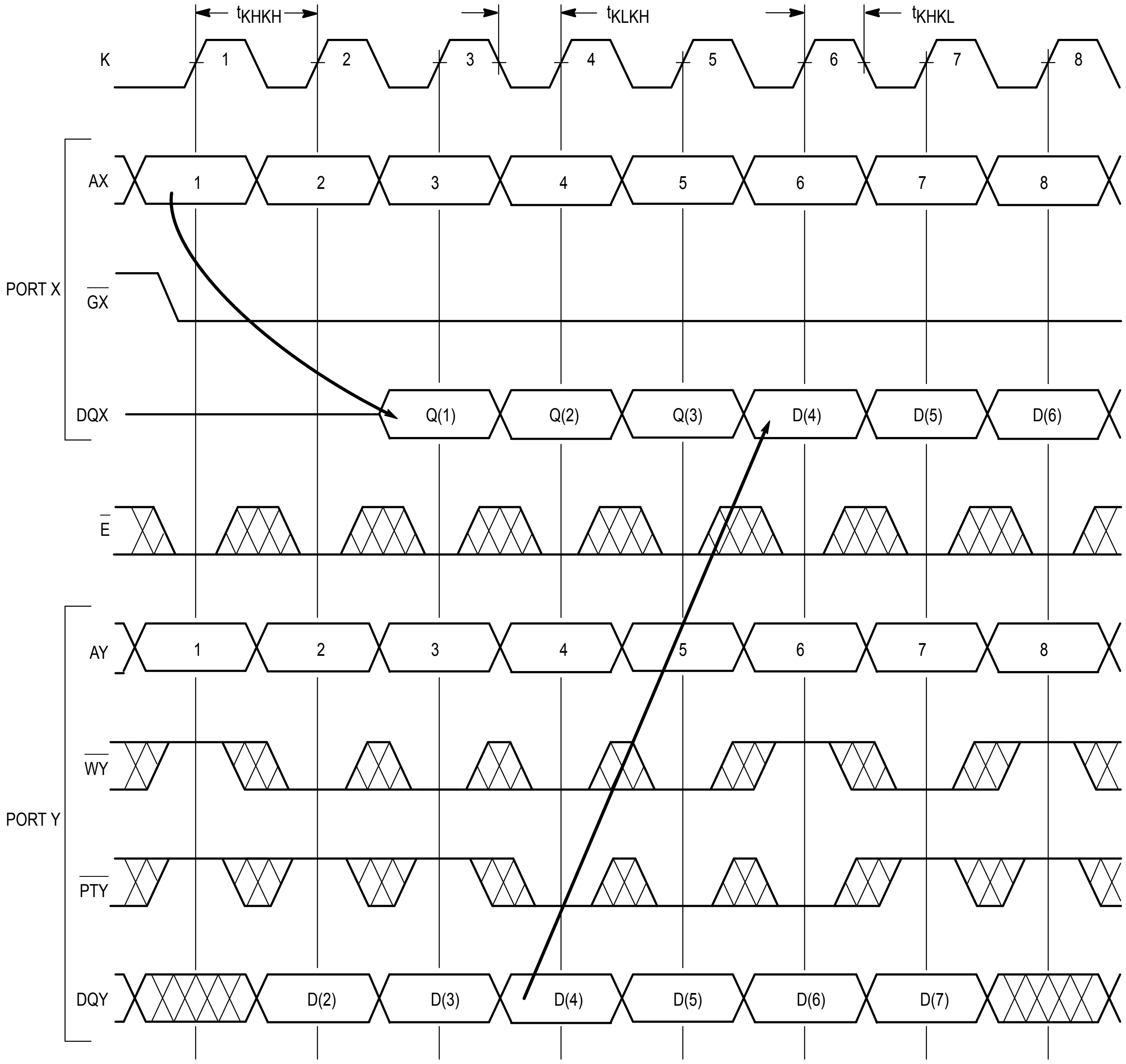

Figure 5. Write and Pass–Through Timing for “One Way” Switch Fabric

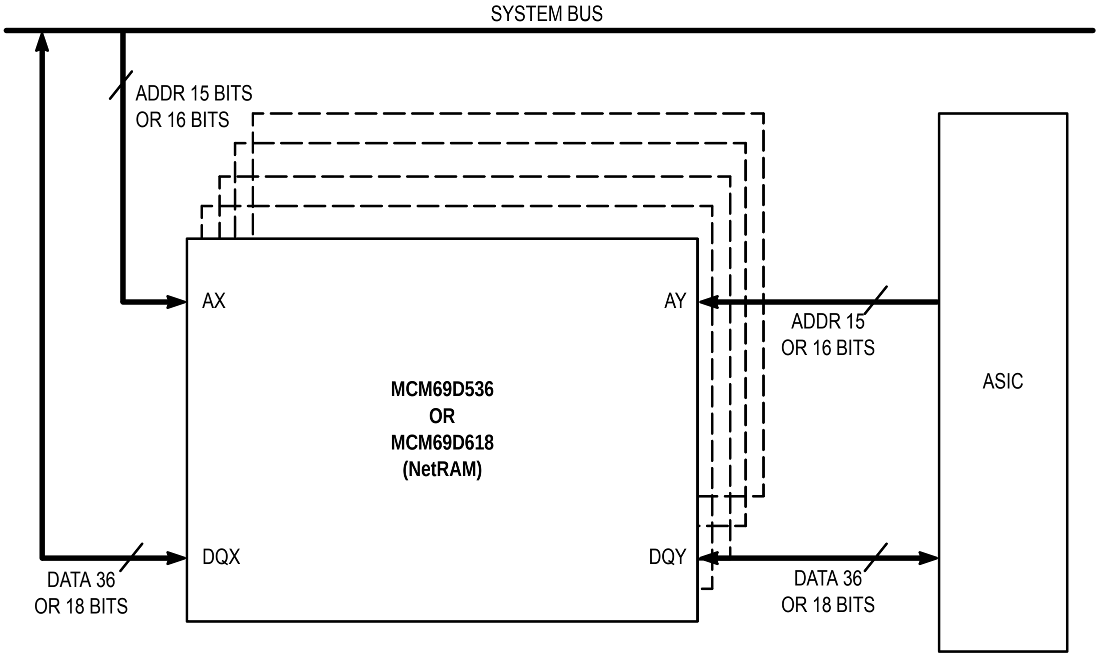

Figure 6. "Snoop" Switch Fabric Using NetRAMs as Dual I/O Devices

"SNOOP" SWITCH FABRIC

The "snoop" switch fabric implementation of NetRAM is shown in Figure 6. The NetRAM is configured as a common I/O with the system bus utilizing port DQX for storage and retrieval of cells or frames. While at the same time, on the ASIC dual port side (DQY), the system is able to "snoop" into any address to retrieve any data that the designer wants to verify — such as for Ethernet destination addresses — or to modify — such as the VPI/VCI headers for ATM. It is common for the designer to include error checking bits or other intellectual property data that the system needs to have available as soon as the cell or frame begins to arrive in the shared memory. This makes it possible for decisions to be made as early as possible in the data input cycle, such as the next destination or priority of service. The ability to write back data at any address is one of the major advantages.

This implementation takes away the need to have a separate path to separate out this critical information from the data flow, because the entire cell or frame is placed into the shared memory, thus reducing chip count and board space.

The "snoop" dual–ported fabric can be configured as a 64K x 18 and 32K x 36 in a dual–ported configuration, which is larger than any other dual–ports on the market and the first of many dual–ported products that will be available from Freescale.

Consider the timing diagram in Figure 7. Assume that in cycle t_2, the cell or frame D(2) gets written to address A(2) in the fabric. In cycle t_3, this cell data can be "snooped" or read from address A(2) by ASIC using address port AY, and become available on data bus (DQY) at the end of cycle t_4 and during cycle t_5. The modifications to this data can be made by ASIC and written back to the fabric in cycle t_6. Simultaneously, the AX port (cell transfer side) can continue to either read or write to the fabric with deselects inserted between the read and write transitions to eliminate corruption of the data.

COMPARISON TO BURST SRAM

The computer market's extremely high volume usage of the burst SRAM determined there would be multi–vendor supply and competitive pricing for the burst SRAMs. For the write–once, read–many functions of level–2 cache, burst SRAMs were perfect. For the cost and supply reasons, some communications designers have attempted to use them in switch fabric. The common characteristic of burst SRAMs is that they have one address port and a common input/output port. As a result, during one clock cycle, the device can either read an address location or write to an address location. In addition, when the common I/O must be turned from the read to write, a deselect cycle must be inserted into the timing, which results in one wait state. It is even worse for the turning of the bus from write to read since two deselect cycles must be inserted to make sure there are no conflicts on the common I/O bus. This translates to a utilization of the clock cycles to data in or out, of between 50 to 70%, depending on the pattern used to access the data in the fabric. For this reason, the burst SRAM parts are not efficient for the write/read/write cycles of the communications market, and this limitation becomes more critical for the gigabit and terabit bandwidths of switches and switch/routers. NetRAM can be configured as a separate I/O device or a dual I/O device which avoids the overall system performance penalties imposed by the need to insert deselect cycles with a common I/O burst SRAM.

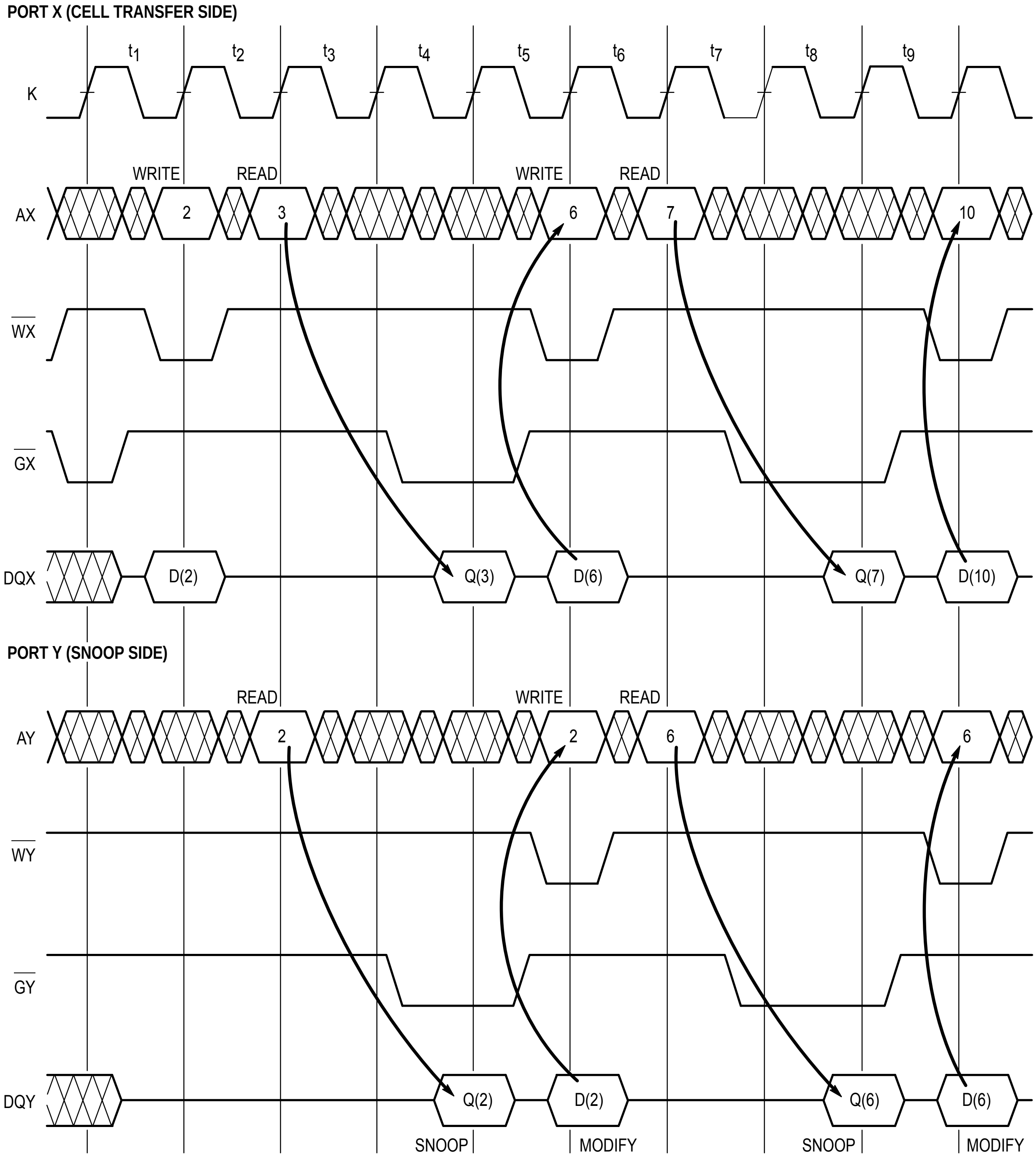

Figure 7. Timing for Snoop Fabric Implementation

SYSTEM PERFORMANCE ADVANTAGES OF NetRAM

NetRAM has the ability to do reads and writes in the same clock cycle with separate addresses for each port, empowering the designer to create higher–speed switch designs than would be possible with a conventional SRAM. Even with the dramatic increases in system frequency and speed that the burst SRAMs have made available, the designer is still not able to get the throughput that is possible with the dual–port NetRAM product while maintaining lower system frequencies.

The communications designer needs to be able to write/read/write a great deal of the time and the burst SRAMs were created for the burst reads and writes of the computing applications. In applications such as datacom switches, which frequently transition from reads to writes, burstRAMs perform inefficiently, causing dead bus cycles between operations. This increases design complexity and reduces the overall system performance. In addition, the dual port feature of the NetRAM enables the designer to simultaneously read and write to different addresses in each clock cycle. Even with recently announced products that read and write in back to back cycles without deselects, the throughput would be one–half that of the NetRAM.

CONCLUSION

The first members of Motorola's NetRAM family will be available to meet your high–performance communications system memory requirements. The MCM69D536 is a 3.3–volt synchronous 32K x 36 dual port SRAM with pipelined reads. This product is also available in the 64K x 18 configuration as the MCM69D618. The NetRAM offers a high–performance memory solution with frequencies as high as 83 MHz and clock–to–data access times of 6 ns. This device also offers the flexibility of being used as either a dual I/O or a separate I/O device. The various configurations of the NetRAM are shown below.

Product Family Configurations

Part Number	Dual Address	Single Address	Dual I/O	Separate I/O
MCM69D536	✓	Note 1	✓	Note 2
MCM69D618	✓	Note 1	✓	Note 2

NOTES:

1. Tie AX and AY address ports together for the part to function as a single address part.
2. Tie $\overline{GX}$ high and $\overline{WX}$ low for DQX to be inputs and tie $\overline{WY}$ high and GY low for DQY to be outputs.

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.
Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

freescale™
semiconductor