Implementation of a Shared Buffer ATM Switch Embedded Scalable Pipelined Buffer Memory

Gab Joong Jeong*

가변형 파이프라인방식 메모리를 내장한 공유버퍼 ATM 스위치의 구현

정갑중*

ABSTRACT

This paper illustrates the implementation of a scalable shared buffer asynchronous transfer mode (ATM) switch. The designed shared buffer ATM switch has a shared buffer of a pipelined memory which has the access time of 4 ns. The high-speed buffer access time supports a possibility of the implementation of a shared buffer ATM switch which has a large switching capacity. The designed switch architecture provides flexible switching performance and port size scalability with the independence of queue address control from buffer memory control. The switch size and the buffer size of the designed ATM switch can be reconfigured without serious circuit redesign. The designed prototype chip has a shared buffer of 128-cell and 4 x 4 switch size. It is integrated in 0.6um, double-metal, and single-poly CMOS technology. It has 80MHz operating frequency and supports 640Mbps per port.

요 약

본 논문은 가변형 공유 버퍼 ATM 스위치의 구조 및 VLSI 구현에 관한 연구이다. 본 논문에서 설계한 단일 칩 공유 버퍼 ATM 스위치는 4ns접근속도의 가변형 파이프라인 방식 공유 버퍼를 내장하고 기존의 공유 버퍼 ATM 스위치들이 가지는 메모리 사이클 시간 제한을 해결한다. 내장 버퍼의 가변성을 이용하여 유연한 스위칭 성능을 지원하고 버퍼 메모리 제어와 주소 큐 제어의 독립성을 이용하여 포트 사이즈의 가변성을 제공한다. 제안된 ATM 스위치는 스위치 사이즈와 버퍼 사이즈의 가변성을 이용하여 복잡한 회로의 재설계 없이 용량 및 성능을 재구성할 수 있다. 0.6um CMOS 기술로 설계된 칩은 동작 주파수 80MHz, 640Mbps/port, 4 x 4 Switch Size를 지원한다.

키워드

initial latency, pipelined memory, ATM switch, scalability, shared buffer

*경주대학교 컴퓨터전자공학부

접수일자 : 2002. 3. 13

Ⅰ. Introduction

Asynchronous transfer mode (ATM) has been selected as the multiplexing and switching technique for use in the Broadband Integrated Services Digital Network (B-ISDN). Switch element design is an important issue to provide efficient and reliable transport for various services in the B-ISDN. There have been extensive researches in ATM switch architectures. We can classify ATM switches globally according to the location of buffers which are named input buffer, output buffer, cross point buffer, and shared buffer [1]. Shared buffer ATM switch minimizes the amount of buffers needed to achieve a specific cell loss rate without the head-of-line (HOL) effect which limits switch throughput to 60% [2]-[4]. However, in the shared buffer approach, the shared single port memory of an N x N switch must operate 2N times faster than I/O port speed. As the access time of memory is physically limited, the approaches in [2] and [3] are not scalable. Several ATM switch architectures have been introduced to improve the memory access time of a large shared buffer [5], [6]. These architectures have divided a large memory buffer into small memory banks. They have solved the problem of buffer access time restriction but needed high speed switching circuit matrix and complex address control. The architecture in [5] has the HOL effect although the throughput reaches 98% for random traffic. In bursty traffic, its throughput will be degraded less than 98% caused by the HOL effect. The number of memory bank is fixed for the architecture in [6]. These architectures can not provide the scalability of shared buffer for flexible switching performance, as memory bank control is not independent of address control.

In this paper, we have proposed and designed a reconfigurable shared buffer ATM switch [7], [8]. The proposed switch has embedded 4ns scalable pipelined buffer memory [9]-[11], and breaks through the restriction of buffer memory cycle time for shared buffer ATM switch. The proposed switch gives scalability in switch size and buffer size, as queue address control is independent of buffer memory control. It supports the flexible switching performance by resizing the shared buffer capacity with little circuit redesign. The proposed switch supports serial address control scheme for the reconfiguration. We have divided a large buffer memory into small memory banks and configured a 2-D array of the memory banks. The 2-D array configuration of the shared buffer can be scalable by adding additional memory banks and decoders. Gated clock in each memory bank minimizes the increase of power consumption according to the increase of the array configuration. The scalable pipelined buffer memory is considered as one large buffer by address controller. We have provided port size expandability with designed switch elements.

In Section II, the reconfigurable shared buffer ATM switch architecture is introduced. This section describes the characteristics of the proposed switch, logical output queue structure, and the characteristics of the scalable pipelined buffer memory. Section III describes VLSI implementation, detailed circuit design, and the gated clocking scheme to minimize the power consumption of the pipelined buffer memory. The characteristics of the designed prototype chip and its advantages and disadvantages are described in Section IV. We have discussed the simulation environment of switching performance and the simulation results of the designed chip in this section. Section V draws conclusions and future works.

II. Switch Architecture

1. Reconfigurable Shared Buffer Switch Architecture

The concept of a general shared buffer ATM switch architecture is shown in Fig. 1. Input cells are multiplexed by MUX block and stored in shared buffer memory. The MUX carries out serial-to-parallel conversion. Stored cells are read from shared buffer memory. They are demultiplexed by DMX block and transferred to their own destination output ports. The DMX block carries out parallel-to-serial conversion. Shared buffer switch shares one large buffer. However, each output port has its own logical output queue individually. In this case, one shared buffer supports all input and output cells. It has to operate at 2NV speed when each port speed is V. If the switch size is large, the operation speed of shared buffer must increase in the proportion of switch size.

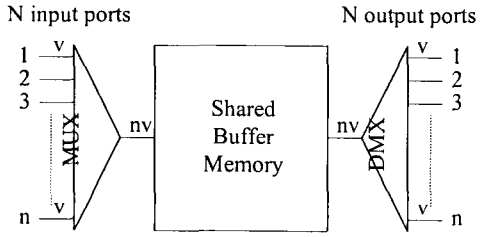


Fig. 1 General concept of a shared buffer ATM switch architecture.

Fig. 2 shows the architecture of the reconfigurable shared buffer ATM switch proposed in this paper. The proposed switch consists of a rotation buffer, an address controller, and a scalable pipelined buffer. The rotation buffer performs the MUX and DMX functions in Fig. 1. Before an input cell arrives at an input port of switch, the 5-byte header of standard 53-byte ATM cell recommended by ITU is translated by new header and additional routing information, routing tag, is appended to the input cell. The routing tag represents the destination output port of input cell. The proposed switch handles 64-byte cell that has 11-byte routing tag and standard 53-byte ATM cell. The 64-byte cell can support various additional functions easily in the future and minimize the hardware complexity of controller because the cell size is multiples of 2.

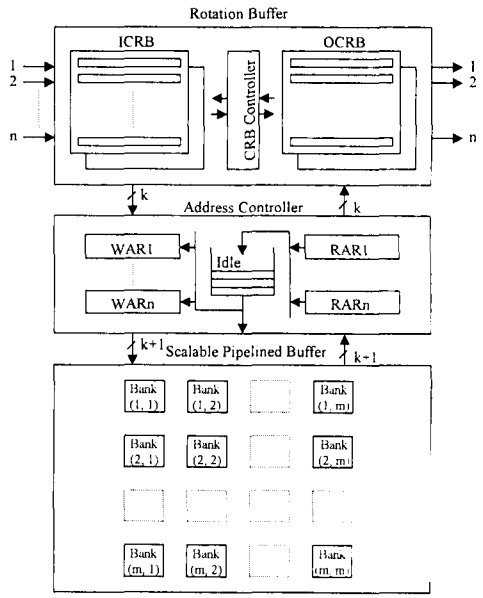


Fig. 2 Reconfigurable shared buffer ATM switch architecture.

The rotation buffer communicates with the address controller by k-bit parallel data. It consists of input cell rotation buffer (ICRB) and output cell rotation buffer (OCRB). The ICRB converts input cell into k-bit parallel data. The OCRB converts output cell into serial data fitted for the port data width. The address controller has an idle address queue and pointers that represent virtual output queues of each output port. It manages idle addresses and output queues. The idle addresses can be elements of all output queues for sharing the buffer. The address controller communicates

with the shared scalable buffer by k+1-bit parallel data. The one additional bit represents the next cell address of an output queue. In the scalable pipelined buffer, we have divided a large buffer memory into small memory banks and configured a 2-D array of the small memory banks. We can scale the 2-D array configuration to enlarge the shared buffer size without serious circuit redesign by adding additional memory banks and decoders. That means switching performance can be scalable for the target switch size. The operating speed of the shared buffer memory has to increase in proportion to the switch size for a given word length of the buffer memory. The required memory cycle time T of the proposed ATM switch is determined as follows:

$$T = CellArrivalTime/[(CellSize/k)x2xN]$$
$$= k/(PortSpeedx2xN).$$

2. Queue Structure

Fig. 3 shows the queue structure of the proposed switch architecture. The idle address queue consists of an idle address pool, an idle address queue head register (IAQHR), and an idle address queue tail register (IAQTR). The IAQHR and the IAQTR indicate the head and tail of the idle address queue. One output queue of an output port consists of a pair of read address register (RAR) and write address register (WAR). The RAR1 and WAR1 represent the head and tail of the output queue of output port 1. The RARn and WARn represent the head and tail of the output queue of output port n.

Head of each output queue indicates next output cell to be transferred through each output port. After one output cell is read from the shared buffer using a RAR for cell output, the value of RAR is overwritten by new next output cell address that comes from the current output cell of its output port at every cell output cycle of the port.

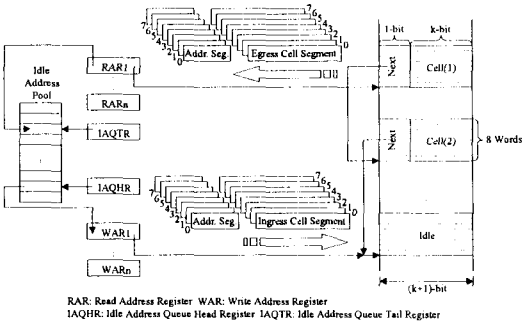


Fig. 3 Queue structure.

Simultaneously, the address for current output cell, the value of used RAR, is stored again in the idle address queue located at the address of IAQTR. The IAQTR is increased by 1 after storing one freed address. The selected RAR or WAR is updated by shift-left operation in serial. If n-bit address is used, n-bit shift-left operation is needed for updating the RAR or WAR completely. The shift-left operation is synchronized by writing or reading clock cycles of the shared buffer for one cell.

Tail of each output queue indicates next cell address that is idle for next input cell to be transferred through each destination output port. Input cell is stored at the location indicated by a WAR that is selected by its destination. Simultaneously, the selected WAR is overwritten by new next cell address that comes from the idle address queue. After reading out one new next cell address from idle address pool using the IAQHR, the value of the IAQHR is increased by 1.

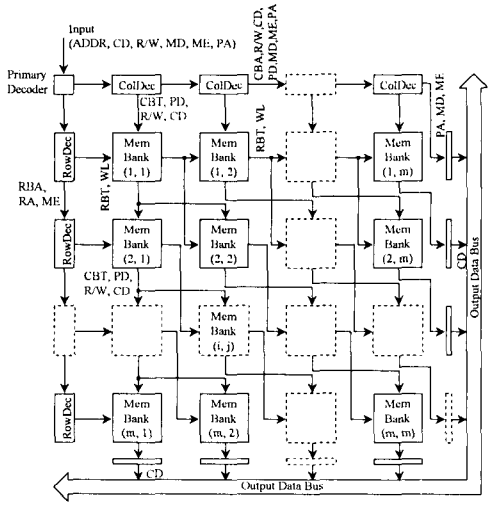


Fig. 4 The architecture of scalable pipelined buffer

3. Scalable Pipelined Buffer

The scalable pipelined buffer in the proposed switch is designed by high-speed pipelined memory that has a 2-D array of SRAM based small memory banks. The architecture of the scalable pipelined buffer is shown in Fig. 4. It consists of a primary decoder, a column decoder, a row decoder, a small memory bank, and an output buffer. The 2-D array configuration is determined by the required buffer size that affects switching performance. The primary decoder decodes parts of input address and generates special pipeline control signals according to the algorithm in Fig. 5. It generates column and row branch addresses (CBA and RBA) which are used in column and row decoders. It generates another special signal, pipeline depth (PD), which is used in each SRAM based small memory bank.

```
Algorithm: Primary Decoding (RBA, CBA, PD, ADDR)
    x = log2M bits of ADDR;
    y = another log2M bits of ADDR;
    Temp = x - y;
    if( Temp >= ZERO ) {
            RBA = | Temp |;
            CBA = 0;
            PD = x;
    } else {
            RBA = 0;
            CBA = | Temp |;
            PD = y;
    }
```

Fig. 5 The algorithm of primary decoding

Each column decoder compares the CBA, which is transferred by the primary decoder or previous column decoder, with zero. If it is true, column decoder generates one bit signal, column branch trigger (CBT), and transfers the CBT to adjacent memory bank with cell data (CD) in vertical. The column decoder transfers the PD to adjacent memory bank in vertical, and the next CBA and PD, which are decreased by 1 from its input CBA and PD, to the next column decoder with the CD in horizontal. Each row decoder compares RBA, which is transferred by the primary decoder or previous row decoder, with zero. If it is true, row decoder generates row branch trigger (RBT) signal and transfers the RBT with pre-decoded word-lines to adjacent memory bank in horizontal. The row decoder transfers the next RBA, which is decreased by 1 from its input RBA, to next row decoder in vertical. The CBT and RBT signals go to low when the memory enable (ME) propagated by the address controller is low.

Each memory bank transfers input data to next adjacent memory banks when the CBT or RBT is high according to the systolic fashioned three-directional data flow. It compares the PD with zero. If CBT and RBT are high and PD is zero in a memory bank, the memory bank accesses its own internal memory cells according to the 'R/W' signal. The memory bank transfers

the next PD, which is decreased by 1 from its input PD, to the next adjacent memory banks. Output buffers located at the last pipeline stage transfer valid data to data bus. Only one output buffer is activated to transfer the CD at every cycle when the both signals, CBT and RBT are high and the 'R/W' signal indicates 'read' mode. In the 'read' mode, the output data of the scalable pipelined buffer come out after the initial latency of M + 3 cycles for M x M array configuration from the read address input cycle. In 'write' mode, the input data is stored during M + 3 cycles. The scalable pipelined buffer does not need any redundant cycles for the mode change between 'read' and 'write'. Timing diagram of the buffer is shown in Fig. 6. It shows the initial latency of 7-cycle for 4 x 4 array configuration of small memory banks.

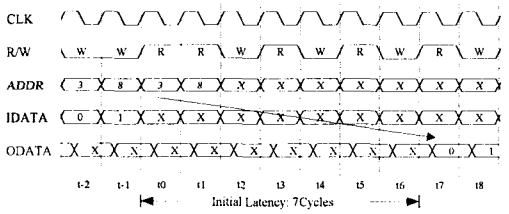
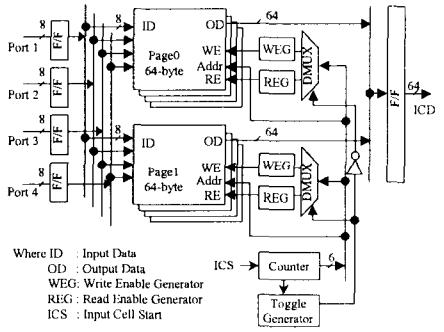


Fig. 6 Input and output timing of the scalable pipelined buffer for 4 x 4 array configuration

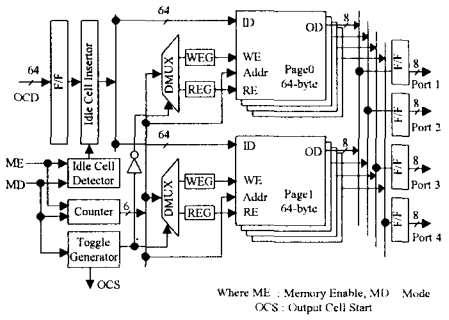
III. VLSI Implementation

1. Cell Rotation Buffer

Fig. 7 shows cell rotation buffer that consists of input cell rotation buffer (ICRB) and output cell rotation buffer (OCRB). The ICRB converts the 64 words of 8-bit data into the 8 words of 64-bit data. The OCRB converts the 8 words of 64-bit data into the 64 words of 8-bit data. They have two cell pages individually for buffering input and output cells. Two pages of them operate exclusively. One page of the ICRB loads 8-bit input data per port at each clock cycle. Another page of the ICRB delivers the loaded ATM cell by 64-bit data at each clock cycle to address controller. It transfers 4 ATM cells to the address controller during 32-cycle. The two cell-pages exchange the roles of each other at every 64-cycle. Two pages of the OCRB operate like the two pages of ICRB except that the OCRB receives data from the shared buffer memory through the address controller after the initial latency of the scalable pipelined buffer memory. Output cell start signal, 'OCS' is delayed by the initial latency of the shared buffer from input cell start signal, 'ICS'. If an output port has no ATM cell to transfer, idle cell indication is inserted by the OCRB.



(a)



(b)

Fig. 7 Block diagram of the cell rotation buffer: (a) ICRB, (b) OCRB

2. Address Controller

Designed address controller is shown in Fig. 8. It has read address register (RAR) and write address register (WAR) which have four address registers individually for four logical output queues. It has 1-kbit idle address queue for the scalable pipelined buffer memory of 128-cell. It has a 6-bit master counter (MC) to control 'write' and 'read' modes. The MSB bit of the MC indicates 'write' and 'read' modes. Three LSB bits of the MC are used for the address fraction of one cell operation. One cell is written into the shared scalable buffer during an 8-cycle and is read from the buffer during an 8-cycle.

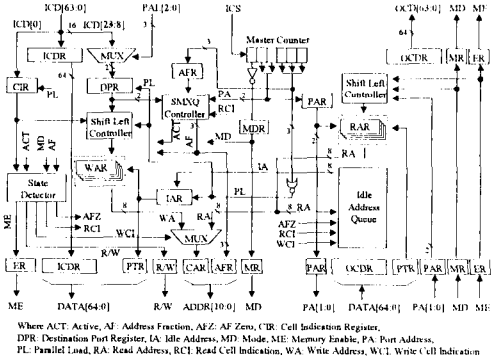


Fig. 8 Block diagram of the address controller

During one cell arriving time of 64-cycle, the address controller of the designed 4x4 switch uses a 32-cycle for 'read' operation and a 32-cycle for 'write' operation. It transfers eight 64-bit data in an 8-cycle for one cell from the ICRB to the buffer. At the same time, the address controller propagates the write address from a selected WAR, the tail pointer of output queue for the destination port of current input cell, to the buffer. The address controller decides the destination port of current input cell when the first 64-bit data of the cell arrive. It selects the destination field in the first 64-bit cell data using a 3-bit signal, port address location (PAL), which is given by user defined external data for switch size expansion such as delta network with designed switch elements. The address controller writes 8-bit idle cell address, which comes from the idle address queue, into the used WAR and simultaneously transfers the idle cell address in serial with the eight 64-bit cell data to the shared buffer. It propagates 65-bit data to the shared buffer that include next cell address.

In 'read' mode, the address controller transfers read addresses, which are the head addresses of output queues, from the four RARs to the scalable pipelined buffer memory one by one for each output port. It transfers the address into cell address register (CAR) after mode (MD) signal goes to low by the MC. The address controller propagates a port address (PA) generated by the MC to the shared buffer with a selected read address among RARs. Used 8-bit read cell address is transferred and restored to the idle address queue in parallel. Cell data (CD), which are read from the buffer memory, come out after the initial latency of the pipelined buffer. MSB of the 65-bit output data of the buffer is loaded in a RAR. Shift-left controller selects one of four RARs with the PA transferred from the buffer. Selected RAR loads the next cell address in serial during an 8-cycle. The access time of the idle address queue can be slow enough to use typical asynchronous memory, as one idle address is treated during an 8-cycle. The timing diagram of input and output cells is shown in Fig. 9.

We have used a buffer sharing scheme which is called sharing with maximum queue length (SMXQ). In the SMXQ, a limit is imposed on the maximum number of buffers to be allocated to each output port. This sharing scheme supports the advantage of buffer sharing like complete sharing (CS) and eliminates buffer hogging by one congested output port that degrades throughput.

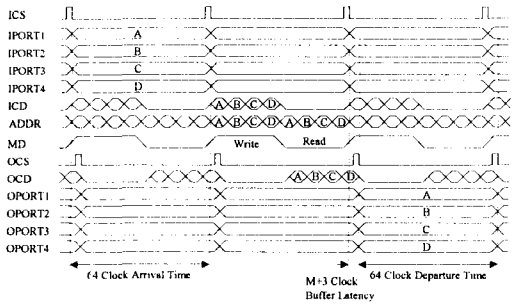


Fig. 9 Timing diagram of input and output cells

It can be implemented with simple control circuit. The buffer sharing scheme of SMXQ provides better performance than the buffer sharing scheme of CS in heavy load [12]. We have adopted Latouche's square root formula in [13] for deciding the maximum queue length of the SMXQ. The square root formula is described in Section IV. The maximum queue length is 64 for the designed prototype chip because the chip has the switch size of 4x4 and the shared buffer size of 128-cell.

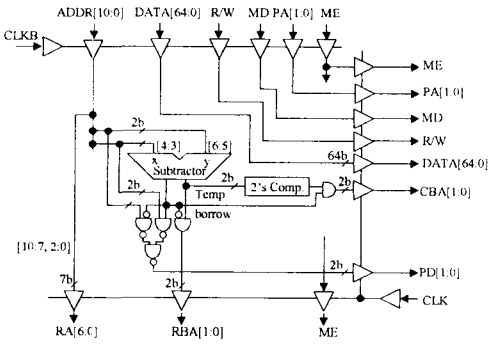
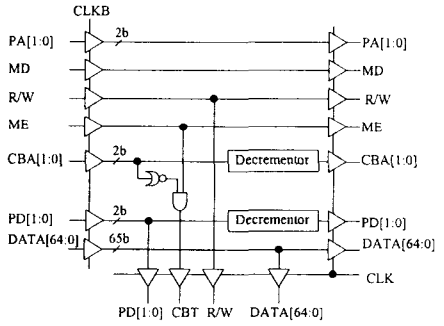


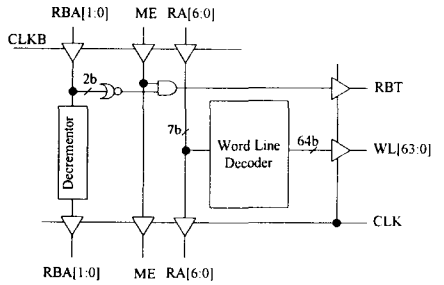
Fig. 10 Block diagram of the primary decoder

3. Scalable Pipelined Buffer Memory

We have designed a scalable pipelined memory for the shared buffer. The block diagram of the pipelined memory is shown in Fig. 4. All of the blocks have input and output latches. All the input latches transfers new input data during the negative level of clock. All the output latches transfer internal data during the positive level of clock. The block diagram of the primary decoder is shown in Fig. 10. It consists of a subtractor, 2's complement block, and some gates for signal selection according to the algorithm in Fig. 5. Signal selection method is very simple. All the pipeline control data, column branch address (CBA), row branch address (RBA), and pipeline depth (PD) are determined by the 'borrow' signal. The bit size of CBA, RBA, and PD is determined by the array configuration of memory banks in the pipelined memory. It is log2M bits for the M x M array configuration. It is designed by the size of 2-bit because the pipelined memory in the prototype chip has the 4x4 configuration of memory banks.



(a)



(b)

Fig. 11 Block diagrams of decoders: (a) column decoder, (b) row decoder

Column and row decoders have input and output latches like the primary decoder. The column decoder has two decrementors to decrease PD and CBA by 1, and has a zero detector for the generation of column branch trigger (CBT) to memory bank when the CBA is zero. The column decoder propagates additional data, port address (PA), through the pipeline stages. Address controller accepts the PA propagated by the column decoder for selecting one of four RARs to overwrite its address by the next address. The row decoder has one decrementor to decrease the RBA by 1, and has a zero detector for the generation of row branch trigger (RBT) to memory bank when the RBA is zero. The row decoder has a row address (RA) decoder that pre-decodes the word-lines for fast memory access. Fig. 11 shows the block diagrams of column and row decoders.

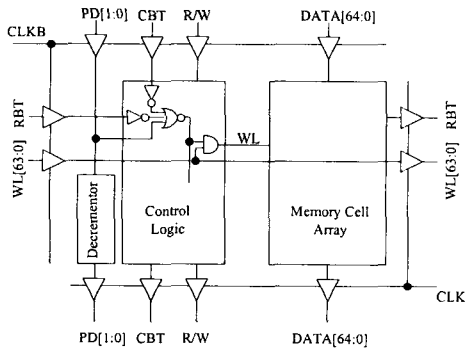


Fig. 12 Block diagram of one memory bank

We have designed a SRAM based small memory bank by 65 bit-lines and 64 word-lines. One cell occupies 8 word-lines. One memory bank can store 8 cells that include 8-bit next cell addresses. We have used 16 memory banks for configuring an array of 4 x 4. The total buffer memory can store 128 cells. Fig. 12 shows the block diagram of one memory bank. It consists of input and output latches, one decrementor to decrease PD by 1, control logic block, and memory cells. The control logic block controls the memory bank operation. The control logic block generates control signals for 'read', 'write', and propagation of data. It controls sense amplifiers and pre-charges the bit-lines of memory columns and the load lines of sense amplifiers to Vdd. The pipelined buffer memory uses the divided word-line technique in [14] but the word-line capacitance is not increased according to the increase of array configuration, since all the memory banks are pipelined by input and output latches. The cycle time of a designed memory bank is 4ns including input and output latch delays with output load capacitance. The circuit diagram of one memory column is shown in Fig. 13. We have used 6-tr SRAM cell and added additional word-line for propagation in the each SRAM cell. The designed single stage sense amplifier is a p-MOS cross-coupled sense amplifier that has low power consumption. In the circuit of the entire memory column, there is no steady state current path to minimize the power consumption in memory circuit. Fig. 14 shows the simulation results of one SRAM based memory bank.

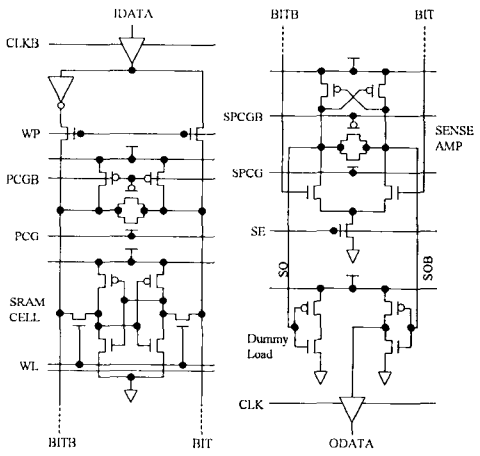


Fig. 13 Circuit diagram of one memory column

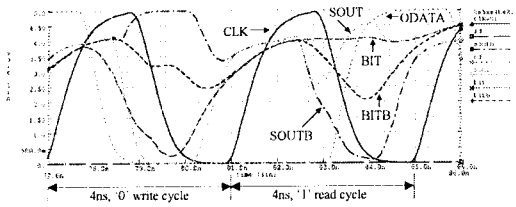


Fig. 14 Simulation results of one SRAM based memory bank

For minimizing power consumption of the pipelined memory, we used a gated clock in each memory bank. The gated clock disables the I/O latches of inactive memory banks. The gated clocking scheme in a memory bank is shown in Fig. 15. We separated the I/O latches in a memory bank by vertical and horizontal latches to reduce clock skew, since vertical and horizontal data are independent in a memory bank. Data transmission is decided by the trigger signals, CBT and RBT, in a memory bank. If the CBT signal is 'low', all vertical data are not loaded in input and output latches. If the RBT signal is 'low', all horizontal data are not loaded in input and output latches. Fig. 16 shows the floorplan and layout of one memory bank.

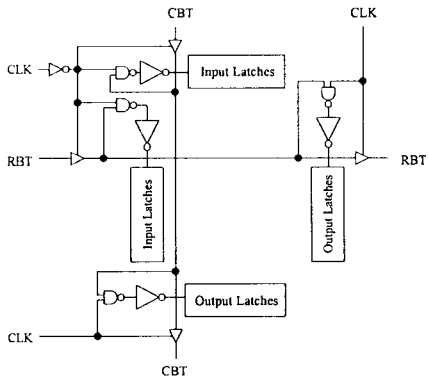


Fig. 15 Gated clocking scheme of one SRAM based memory bank.

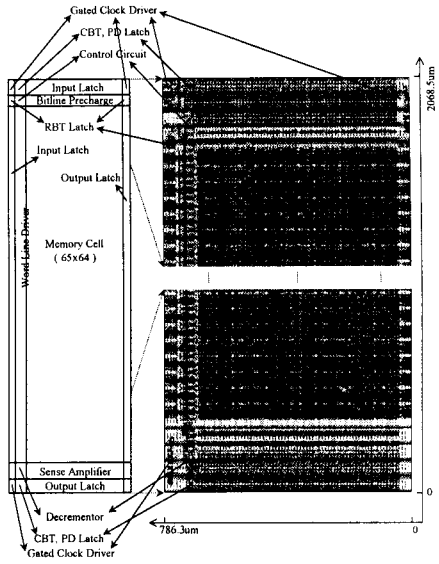


Fig. 16 The floorplan and layout of one memory bank

IV. Simulation Results and Discussions

1. Characteristics of Prototype Chip

The designed prototype, 4 x 4 reconfigurable shared buffer ATM switch chip, is shown in Fig. 17. It has the scalable pipelined memory of 65-Kbit SRAM for the shared buffer of 128-cell with the next cell address in every cell. The prototype chip is designed by 0.6um single-poly double-metal CMOS technology. Core size is 10.6 x 10.6 mm2. The buffer memory is designed by full custom. Inter bank routing area of the pipelined memory takes 15% of total memory area. It can be shrunk with better process technology. General low-power and high-speed circuit techniques of SRAM can be applied to the scalable pipelined memory for increasing memory cycle time and reducing power consumption. In the region of address controller, chip area can be reduced with using random access memory for the idle address queue. Operating frequency of the prototype chip is 80MHz by post layout simulation.

The operating frequency can be enhanced by the speed optimization of control circuit. The cycle time of the designed buffer memory is 4ns. It can support 640Mbps/port 8 x 8 switch enough, since the 640Mbps 8 x 8 switch requires 6.25ns as the cycle time of a shared buffer by the equation of a required memory cycle time in Section II. The high-speed memory cycle time is one of the advantages of the proposed ATM switch. Priority control and multicasting can be solved with adding additional queues. Estimated power consumption of the designed prototype chip with a gated clock is 3.3W at the supply voltage of 5V for the 640Mbps/port 4 x 4 ATM switch. Table 1 shows the characteristics of the prototype chip.

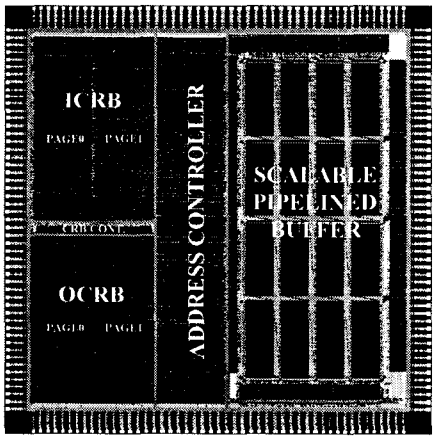


Fig. 17 The full chip layout of 4 x 4 prototype ATM switch with the shared buffer of 128-cell

Table 1. Characteristics of prototype chip

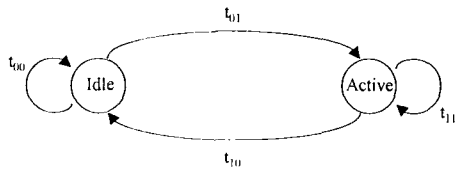
Process Technology	0.6μm 2-Metal 1-Poly CMOS
Core Size	10.6 x 10.6 mm^2
Switch Size	4 x 4
Buffer Size	66560-bit (4 x 4 x 4160-bit)
Transistors	1-Million
Max Buffer Cycle Time	4ns (Simulated)
Operating Frequency	80MHz (Simulated)
Throughput	2.5Gbps
Power Dissipation	3.3W at 80MHz (Estimated)
Power Supply	5V

2. Switching Performance

We have simulated to evaluate the performance of the designed prototype chip for random and burst cell arrival processes with uniform distribution to the output ports of the switch. For random (Poisson) cell arrival process, input cells are randomly generated according to the offered load (P*) arriving at each input port, and are uniformly distributed to the output ports of the switch. The load matrix is:

$$P = \begin{bmatrix} P^*/4 & P^*/4 & P^*/4 & P^*/4 \\ P^*/4 & P^*/4 & P^*/4 & P^*/4 \\ P^*/4 & P^*/4 & P^*/4 & P^*/4 \\ P^*/4 & P^*/4 & P^*/4 & P^*/4 \end{bmatrix}$$

We have used a simple two-state Markov chain model for burst cell arrival process. Fig. 18 shows the model. It stochastically alternates between an active state and an idle state with transition probabilities, t00, t01, t10, and t11. A cell arrival occurs at the offered load of P* during a time slot if the Markov chain is in active state at the beginning of the time slot, and no arrival occurs otherwise. In bursty traffic, there can be no idle state transition when consecutive new bursty cell stream is occurred in the active state. L is defined as the average burst length. We let all entries in the load matrix of uniform bursty traffic be identical as those in the load matrix of uniform random traffic. In burst matrix, we let Li,j = L* for all i, j. L* is the mean burst length of each input port.



Where $t_{01} = \frac{P^*}{L \cdot (1 - P^*)}$, $t_{00} = 1 - t_{01}$, $t_{10} = \frac{1}{L}$, $t_{11} = 1 - t_{10}$.

Fig. 18 Two-state Markov model for bursty traffic

In our experiments, the simulation run is repeated 10 times with independent random number streams at every offered load. We have increased the offered load from 0.9 to 1.0 for uniform random traffic and from 0.5 to 1.0 for uniform bursty traffic. The length of each simulation run is 10-million cell cycles after 1-million initial cell cycles for the steady state of the switch. We have adopted to eliminate the buffer hogging effect by sharing with maximum queue length (SMXQ). We have supposed all output ports have fair sharing by the buffer sharing scheme of SMXQ. We have calculated the mean cell loss probability as the mean value of 10 cell loss probabilities of each output port. We have used the variance of the mean cell loss probability to get a confidence interval of 90%. We let the maximum queue length for SMXQ, X, for each output $j(1 \le j \le N)$, are calculated using the following Latouche's square root formula for

$$SMXQ: \; X_j = \lfloor B/\sqrt{N} \rfloor \forall_j$$

giving their best performance in the experiment [13].

We have compared the SMXQ, implemented in the prototype chip, with the buffer sharing scheme of complete sharing (CS). Fig. 19 shows the cell loss probability in uniform random traffic for the designed 4 x 4 ATM switch embedded the shared buffer of 128-cell. The cell loss probability in uniform random traffic is near 10-9 at the offered load of 0.9. It satisfies the requirement of an ATM exchange [2]. Fig. 20 shows the cell loss probability in uniform bursty traffic with the offered load of 0.95. It shows that SMXQ provides better performance in comparison with CS in heavy load for the designed switch configuration.

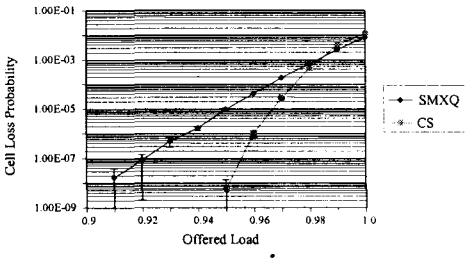


Fig. 19 Cell loss probability of SMXQ and CS in uniform random traffic for 4 x 4 ATM switch with the shared buffer of 128-cell

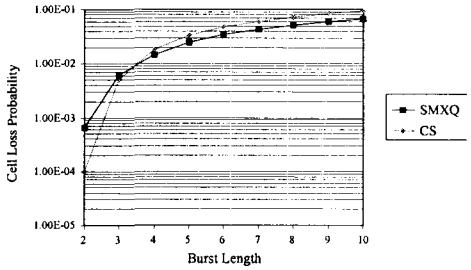


Fig. 20 Cell loss probability of SMXQ and CS in uniform bursty traffic with the offered load of 0.95 for 4 x 4 ATM switch with the shared buffer of 128-cell

3. Scalability

Scalability is one of the advantages of the proposed shared buffer ATM switch that supports versatile switching performance. The shared buffer size is enlarged with increasing the array configuration of the scalable pipelined memory without the variation of memory cycle time. The increase of the shared buffer size is accomplished by adding additional memory banks and column and row decoders without serious circuit redesign. The scalability is an advantage for fast migration to better process technology, too. The proposed ATM switch realizes easily to scale the switch size with adding additional rotation buffer and counter bit to indicate each port, since the address controller uses serial writing and reading of the next cell address of each output queue and the

address control is independent of the buffer control. The cycle time of the scalable pipelined memory will be increased by better process technology and high-speed circuit techniques for SRAM. The high-speed memory cycle time solves the restriction of memory cycle time for enlarging switch size, which a shared buffer ATM switch has.

We have simulated with reconfigured switch size (S) and buffer size (B) to show the scalability. The proposed switch is reconfigured by 8 x 8 switch with the shared buffer of 200-cell and 16 x 16 switch with the shared buffer of 288-cell. Fig. 21 shows the cell loss probability in uniform random traffic. Fig. 22 shows the cell loss probability in uniform bursty traffic with fixed burst length of 8. The array configuration of the scalable pipelined memory is 6 x 6 for the buffer size of 288-cell and 5 x 5 for the buffer size of 200-cell. In the view of cell loss probability, those three configurations have similar performance in random and bursty traffic. In the view of silicon area, we have estimated that the 16 x 16 ATM switch with buffer size of 288-cell occupies 14.8mm x 14.8mm with the same technology of the prototype chip, as one small memory bank size is 0.8 mm x 2.1 mm. With using 0.35um CMOS technology, we can estimate the one chip 16 x 16 ATM switch with buffer size of 288-cell is realizable in the area of 8 x 8 mm2. The memory cycle time of 4ns is sufficient to support 160Mbps/port 16 x 16 ATM switch even if we divide a cell into sixteen 32-bit data for addressing each cell in a large buffer over 256-cell. One chip 640Mbps/port 16 x 16 ATM switch is realizable when we use better process technology and faster circuit to support 3.1 ns cycle time of the scalable pipelined memory. In this case, we need to treat 16-bit address data with eight 2-bit address data and not to divide a cell into sixteen 32-bit data for addressing each cell in a large buffer over 256-cell.

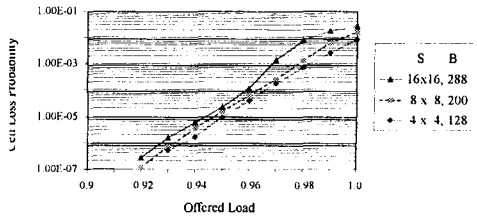


Fig. 21 Cell loss probability of enlarged switch size (S) and buffer size (B) in uniform random traffic

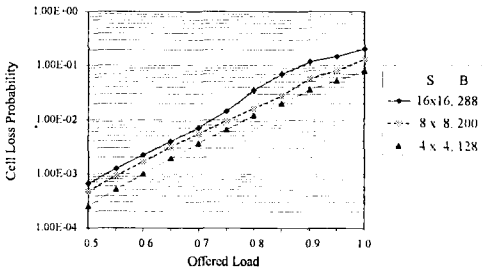


Fig. 22 Cell loss probability of enlarged switch size (S) and buffer size (B) in uniform bursty traffic with the fixed burst length of 8

4. Power Consumption

In the proposed switch architecture, the power consumption arises mainly in the scalable pipelined buffer memory. The power consumption increases seriously according to the increase of the array configuration of the pipelined memory. We have estimated the power consumption of the designed chip by the method of [15]. The power consumption of the pipelined memory is one of disadvantages for enlarging the array configuration. The increase of power consumption followed by the increase of the configuration size of buffer memory array can be reduced by disabling input and output latches in inactive memory banks with a gated clock. Fully active memory banks are M among M x M memory banks because only one diagonal data path is fully active for one operation. Therefore, we can estimate that the reduction rate

of on-chip power consumption of the pipelined memory is 1/M approximately by the gated clock for M x M configuration of the buffer memory bank array. Fig. 23 shows the comparison of the estimated entire power consumption for the proposed ATM switch architecture with gated and non-gated clock in the pipelined memory. We have compared the power consumption of 4 x 4 ATM switches that have 8 x 8 (512-cell) and 4 x 4 (128-cell) configurations of the pipelined memory to show the power minimization using the gated clock.

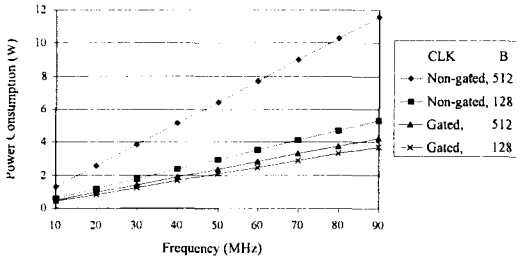


Fig. 23 The comparison of the estimated power consumption for the proposed 4 x 4 ATM switch with gated and non-gated clock in the scalable pipelined buffer memory

V. Conclusion

We have proposed and designed a reconfigurable shared buffer ATM switch in this paper. We have used a scalable pipelined memory for the shared buffer [16]. The switch architecture solves the restriction of memory cycle time in a shared buffer ATM switch. The proposed ATM switch is reconfigurable in switch size and shared buffer size by its structural characteristics as shown above. The designed prototype 4 x 4 ATM switch has the shared buffer of 128-cell. It is integrated in the area of 10.6 mm x 10.6 mm with 0.6 um CMOS technology. The prototype operates at 80MHz by post layout simulation that supports 640Mbps per port. It has the throughput of 2.5Gbps. The memory banks of the scalable pipelined memory are designed by full custom layout. The cycle time of the scalable pipelined memory is 4 ns. It is sufficient for one chip 640Mbps/port 8 x 8 shared buffer ATM switch. We have described the advantages and simulation results of the proposed ATM switch architecture in Section IV. We have shown the feasibility of one chip 640Mbps/port 16 x 16 shared buffer ATM switch by the speed optimization of control circuit and the area optimization using better process technology. In future work, we are researching special functions such as priority control and multicasting for the proposed switch architecture.

References

[1] T. Koinuma and N. Miyaho, "ATM in B-ISDN Communication Systems and VLSI Realization," IEEE J. Solid-State Circuits, vol. 30, no. 4, pp. 341-347, Apr. 1995.

[2] N. Endo, T. Kozaki, T. Ohuchi, H. Kuwahara, and S. Gohara, "Shared Buffer Memory Switch for an ATM Exchange," IEEE Trans. Commun., vol. 41, no. 1, pp. 237-245, Jan. 1993.

[3] Y. Shobatake, M. Motoyama, E. Shobatake, T. Kamitake, S. Shimizu, M. Noda, and K. Sakaue, "A One-Chip Scalable 8*8 ATM Switch LSI Employing Shared Buffer Architecture," IEEE J. Selected Areas Commun., vol. 9, no. 8, pp. 1248-1254, Oct. 1991.

[4] M. G. Hluchyj and M. J. Karol, "Queueing in high-performance packet switching," IEEE J. Selected Areas Commun., vol. 6, no. 9, pp. 1587-1597, Dec. 1988.

[5] H. Kondoh, H. Notani, H. Yamanaka, K. Higashitani, H. Saito, I. Hayashi, S. Kohama, Y. Matsuda, K. Oshima, and M.

Nakaya, "A 622-Mb/s 8x8 ATM Switch Chip Set with Shared Multibuffer Architecture," IEEE J. Solid-State Circuits, vol. 28, no. 7, pp. 808-815, July 1993.

[6] A. Chemarin, A. Andre, A. Botta, J. Majos, J. Rainard, H. Teyssier, and P. Thorel, "A High-Speed CMOS Circuit for 1.2-Gb/s 16x16 ATM Switching," IEEE J. Solid-State Circuits, vol. 27, no. 7, pp. 1116-1121, July 1992.

[7] G. J. Jeong, J. W. Shim, M. K. Lee, and S. H. Ahn, "A Scalable Shared Buffer ATM Switch Embedded SPRAMS," in Proc. IEEE Int. Symp. Circuits and Systems, Monterey, CA, MPA14-5, May 1998.

[8] J. W. Shim, G. J. Jeong, M. K. Lee, and S. H. Ahn, "FPGA Implementation of a Scalable Shared Buffer ATM Switch," in Proc. IEEE Int. Conf. ATM, Colmar, France, pp. 247-251, June 1998.

[9] G. J. Jeong, K. H. Kwon, M. K. Lee, and S. H. Ahn, "High throughput systolic memory architecture using three directional data flows," in Proc. IEEE ICECS, Rodos, Greece, pp. 667-670, Oct. 1996.

[10] K. H. Kwon, G. J. Jeong, M. K. Lee, and S. H. Ahn, "A scalable memory system design," in Proc. Int. Conf. VLSI Design, Hyderabad, India, pp. 257-260, Jan. 1997.

[11] G. J. Jeong and M. K. Lee, "Design of a Scalable Pipelined RAM System," IEEE J. Solid-State Circuits, vol. 33, no. 6, pp. 910-914, June 1998.

[12] F. Kamoun, and L. Kleinrock, "Analysis of Shared Finite Storage in a Computer Network Node Environment Under General Traffic Conditions," IEEE Trans. Commun., vol. COM-28, no. 7, pp. 992-1003, July 1980.

[13] G. Latouche, "Exponential Servers Sharing a Finite Storage: Comparison of Space Allocation Policies," IEEE Trans. Commun., vol. COM-28, no. 6, pp. 910-915, June 1980.

[14] M. Yoshimoto, K. Anami, H. Shinohara, T. Yoshihara, H. Takagi, S. Nagao, S. Kayano, and T. Nakano, "A divided word-line structure in the static RAM and its application to a 64k full CMOS RAM," IEEE J. Solid-State Circuits, vol. SC-18, no. 5, pp. 479-485, Oct. 1983.

[15] D. Liu and C. Svensson, "Power Consumption Estimation in CMOS VLSI Chips," IEEE J. Solid-State Circuits, vol. 29, no. 6, pp. 663-670, June 1994.

[16] G. J. Jeong, "VLSI design of CMOS shared scalable buffer switch for gigabit ATM switching network," PhD dissertation, Yonsei university, 1999.

저 자 소 개

정갑중(Gab Joong Jeong)

1987년 경북대학교 전자공학과 공학사

1989년 경북대학교 대학원 전자공학과 공학석사

1999년 연세대학교 대학원 전자공학과 공학박사

1989년 1월~1999년 3월 LG반도체 책임연구원

1999년 4월~2001년 2월 ETRI 선임연구원

2001년~현재 경주대학교 컴퓨터전자공학부 전임강사

※관심분야 : Packet Switching, QoS, VLSI