MUSIC SEMICONDUCTORS

Application Note AN-N15

USING THE MUSIC ATMCAM™

INTRODUCTION

The forces that are driving the need for high-bandwidth data transmission include multimedia, worldwide computer networking available to millions of individuals, networked commercial services for companies, the move from textual to graphical representation of information, and expensive transportation. ATM is seen as a medium through which these data communication needs can be satisfied.

For the developers of ATM equipment, networks, and services there are some major challenges that must be faced. Some aspects of the ATM specifications call for massive throughput capabilities. For example, multicasting of real-time video data impose very severe demands on available bandwidth.

With the need to satisfy the demands of ATM, there is a constant search for hardware acceleration of processes that software approaches can not practically handle. CAMs have been in common use in Ethernet bridges, routers, and switching for a number of years. The technology provides a convenient way of providing effective lookup into sparsely occupied address space through the mechanism of non-contiguous addressing. The MU9C4320L ATMCAM is especially targeted to addressing the ATM header translation, which is explained in this application note.

ATM VERSUS ETHERNET LOOKUP REQUIREMENTS

MUSIC Semiconductors is the leading supplier of CAM for LAN applications. Briefly exploring the differences in the associative lookup functions for Ethernet and Token Ring applications allows readers familiar with MUSIC Semiconductors LANCAM devices to get an insight into the philosophy behind the ATMCAM.

The header of an Ethernet packet contains a source address field and a destination address field, each a 48-bit physical address. These addresses are allocated by the IEEE. In contrast, the header of an ATM cell contains not physical, but virtual connection information in the form of a Virtual Path identifier (VPI) and a Virtual Channel identifier (VCI). As a cell is passed through a network each network device maps an incoming VPC/VCI to an outgoing VPI/VCI. This is called the header translation process.

As the VPI/VCI values can be reused across a network, their field width is substantially less than the 48-bit Ethernet address width. In an ATM User-Network interface (UNI) cell structure, the VPI field is 8 bits and the VCI field is 16 bits. In an ATM Network Node interface (NNI) cell structure, the VPI field is 12 bits and the VCI field is 16 bits. These 24 to 28 bits are conversely handled within the 32-bit width of the ATMCAM.

A further difference in the way cells are routed through an ATM network is that there are two types of connection: a Virtual Path connection (VPC) and a Virtual Channel connection (VCC). In the case of the VPC, the routing is based on the VPI field alone, whereas the VPC routing is based on both the VPI and VCI fields. The ATMCAM provides a mechanism for supporting both types of connection without performance degradation.

The LANCAM has a programmable memory core that can be set to an area of CAM and an area of RAM across the data field on 16-bit boundaries. Each location has an associative area in which comparison is performed with the incoming data, and an area of associated memory accessed when a match occurs at a specific location. In the Ethernet environment, the LANCAM is often divided into 48 bits of CAM and 16 bits of RAM. The 16 RAM bits per location typically hold port and aging information. In an ATM system, the associated data widths are often much greater than few bits, and the associated data cannot be held practically in the same physical memory array as the associative data. Therefore the ATMCAM generates an address pointer to access associated data in an external RAM. This leaves the ATM system designer able to specify a lookup table of any width.

ATMCAM OVERVIEW

The MU9C4320L ATMCAM is a 4K x 32 content-addressable memory (CAM) that is designed to provide high-speed VPI/VCI lookup to accelerate the header translation function in ATM switches. It includes several features specifically targeted at the ATM environment.

MUSIC Semiconductors, the MUSIC logo, the phrase “MUSIC Semiconductors”, and LANCAM are registered trademarks of MUSIC Semiconductors. MUSIC and ATMCAM are trademarks of MUSIC Semiconductors.

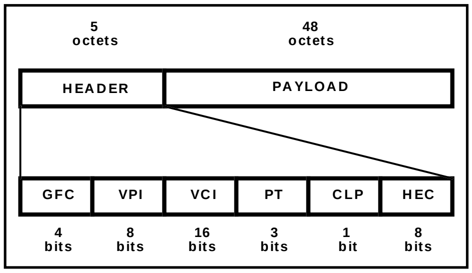

Figure 1: ATM UNI Cell Structure

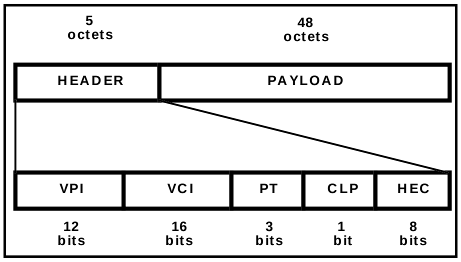

Figure 2: ATM NNI Cell Structure

GFC	**Generic Flow Control**	**PT**	**Payload Type**
VPI	**Virtual Path Identifier**	**CLP**	**Cell Loss Priority**
VCI	**Virtual Channel Identifier**	**HEC**	**Header Error Check**

Referring to Figure 3 on page 3, the data path to and from the device is 32 bits, providing ample width to accommodate the VPI and VCI fields from the ATM cell header. This DQ31–0 data path has one extra bit, the Validity bit, /VB, which is used to write and read validity information to and from individual locations in the CAM array.

The operation performed by the ATMCAM is controlled through the AC11–0 lines. The control states divide into the following categories: Read/Write memory, Read/Write register, Data Move, Comparison, Set Validity, Address Register control, VP Table Control, and Initialization.

Data on the DQ lines has a number of possible sources and destinations within the ATMCAM, including the CAM array and the register set. The Comparand register is loaded with the value for comparison, although data on the DQ lines can be compared directly with the contents of the CAM array and can optionally be written to the Comparand register during comparison.

The results of comparison are output on the Page address, PA3–0, and Active Address outputs, AA11–0, and are qualified by the Match flag, /MF. The address formed by PA:AA is used to access external RAM and lookup associated data. The associated data contains the outgoing VPI/VCI and other connection related information.

The ATMCAM has seven mask registers so comparison can be made to operate only on selected bits within the data field. When a Compare cycle is executed with a mask register specified, any bits set HIGH in the mask register will force a true comparison in the corresponding bits of the data field. Therefore, only bits that correspond to LOW values in the mask register will enter into comparison. One example of where masking can be used in the ATMCAM is when only the 24 bits of the VPI/VCI are needed for comparison. In this case a masked Compare cycle can be executed using one of the mask registers set to FF000000H.

The Configuration register sets the long-term operating conditions of the device, and includes the Page address that is used to provide high-order address bits in a system with more than one ATMCAM.

The Status register holds results of a memory cycle, whether it be a Read, Write, or Compare cycle. These results are available on the pins, but the Status register provides the designer with a convenient method of access through software.

The ATMCAM has two modes of operation, software control and hardware control. The Status register is intended primarily for use in software control mode. In software control mode, which is selected through the Configuration register, the control states are loaded into the Instruction register, and the Device Select register can be used to select a single device or multiple devices in a multiple ATMCAM system through software.

Although the software control mode of operation can be convenient, it does have an impact on the performance of the system. Hardware control mode provides a much faster mode of operation and is the preferred choice for speed-critical systems.

Random access to the CAM array is supported either by direct addressing or by indirect addressing. For direct addressing, the address is input on the AC11–0 lines with the Address Valid line, /AV, LOW to distinguish it from a control state. Alternatively for indirect addressing, the address is loaded into the Address register in a previous cycle, and is used as a pointer to the location to be read or written. Indirect addressing supports DMA transfers through Read or Write cycle control states that cause the contents of the Address register to be incremented or decremented automatically for each Read or Write cycle.

As an alternative to random access Write cycles to the CAM array, the ATMCAM supports Write at Next Free Address (NFA) cycles. Under this circumstance, there is no need to provide an explicit address, instead it is generated by the device from the highest-priority empty location. The vertical cascading mechanism ensures that Write at NFA cycles operate across a system containing more than one ATMCAM.

Multiple ATMCAMs can be connected together using the vertical cascading facility to increase the overall depth of the CAM system. Vertical cascading uses two daisy chains, one for comparison match and prioritization between devices, and one for full indication and writing at next free address within the system. This scheme allows vertical cascading to any practical depth. Although there is a small time penalty per cascaded device, this is usually absorbed in the system timing. In the most time-critical applications the ATMCAM is designed to ease external prioritization.

The ATMCAM is controlled through the Control bus, AC11–0, two Chip Select lines, /CS1 and /CS2, and the Write line, /W. The /AV line is used to distinguish between control and address on the AC lines, a LOW level indicating an address. The Output Enable line, /OE, is used to enable the PA:AA bus. All of these signals and incoming data are latched by the Chip Enable line, /E. Chip Enable is the main timing input to the ATMCAM. All input signals have a setup and hold time with respect to the falling edge of /E, and all output signals are timed from the falling or rising edge of /E.

The rising edge of /E is significant in the timing of match results after a Compare cycle. The PA:AA outputs and the Match flag outputs do not change during a Compare cycle until after /E has gone HIGH. This fact is important because it allows a degree of pipelining. Access into the external RAM can continue while a new comparison is underway in the CAM. As long as /E is LOW the address generated by the previous cycle will remain stable on the PA:AA bus.

Another important feature of the ATMCAM is its ability to handle both VP and VC connections through the use of the VP Table. The VP Table is a single-bit 4K-deep memory array used to flag the presence of a VP connection. The VP Table only needs to be a maximum of 4K deep because the UNI VPI field is 12 bits (see Figure 2.) Therefore, only one device in a multiple ATMCAM system is needed to hold the full VP Table. A VP Table match is given a lower priority than a CAM match, so the table is held in the lowest-priority device in the system.

The ATMCAM has two match flags, /MV and /MF, which distinguish between a CAM match and a VP Table match. The VPI field is used as an explicit address into the VP Table. If there is no CAM match, and the addressed location in the VP Table holds a LOW logic level, the system will respond with a VP Table match. The PA:AA bus carries the VP Table

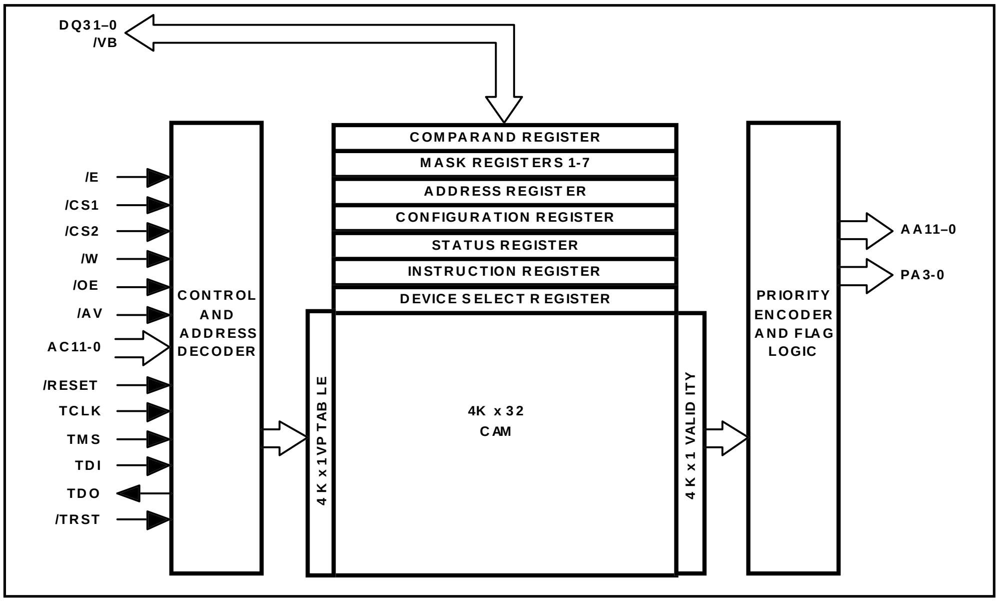

Figure 3: ATMCAM Block Diagram

Page address concatenated with the match address of the VP Table location. The use of the VP Table is optional and is selected through a setting in the Configuration register.

The ATMCAM is initialized through hardware by pulling the /RESET input line LOW, or through software by executing the RESET control state. There is a small difference between the two reset methods: some fields in the Device Select register and the Configuration register remain unchanged after a software reset, thereby preserving certain operational conditions such as Page address.

The ATMCAM has a full JTAG diagnostic interface that allows automatic testing of both the circuit wiring and the full functionality of the device. The JTAG circuitry is designed in accordance with the IEEE Standard Test Access Port and Boundary-scan Architecture, IEEE Standard 1149.1-1990 and IEEE Standard 1149.1a-1993.

Pin Descriptions

DQ31–0 (Data Bus, Input/Output)
The main data bus into and out of the ATMCAM.

AC11–0 (Address/Control Bus, Input)
Address and Control Bus. The state of the /AV line selects the type of input. The input is interpreted as an address when /AV is LOW.

AA11–0 (Active Address, Output)
Depending on the most recent memory cycle, the Active address is one of the CAM Match address, the VP Table address, the Next Free address or the random access address.

PA3–0 (Page Address, Output)
Page Address information output on the PA lines at the same time, and under the same conditions, as the AA lines are active.

/E (Chip Enable, Input)
Chip enable is the main and synchronizing control for the ATMCAM.

/CS1, /CS2 (Chip Select 1, Chip Select 2, Inputs)
The /CS1 and /CS2 inputs enable the ATMCAM. If either /CS1 OR /CS2 are LOW, the device is selected.

/W (Write Enable, Input)
The /W input determines the direction of data transfer on the DQ bus and modifies the control state on the AC lines.

/OE (Output Enable, Input)
The /OE input enables the PA:AA outputs.

/AV (Address Valid, Input)
The /AV input determines whether the AC lines carry address or control information. When /AV is LOW, the AC lines convey a memory address.

/VB (Validity Bit, Three-state, Input/Output)
The /VB line conveys validity information to and from the ATMCAM.

/MV (Match Valid, Output)
After a Compare cycle, the /MV line indicates whether there has been a valid match in either the CAM Array or the VP Table.

/MF (Match Flag, Output)
The /MF output indicates whether a valid match has occurred in the CAM array during the previous Comparison cycle.

/MI (Match Input, Input)
The /MI input receives match information from the next higher priority ATMCAM in a vertically cascaded system to provide system-level prioritization.

/FF (Full Flag, Output)
The /FF output indicates when all the memory locations have their validity bits set valid (LOW).

/FI (Full Input, Input)
The /FI input receives full information from the next higher priority ATMCAM in a vertically cascaded system to provide system-level full information.

/MM (Multiple Match, Open Drain Input/Output)
The /MM line indicates whether there is a multiple match within the system. It is usually interpreted as an error condition.

/RESET (Reset, Input)
The /RESET input is used to reset the ATMCAM to a known state, including setting all CAM locations to empty.

TCLK, TMS, TDI, TDO, /TRST (JTAG Test Pins, Input/Output)
The JTAG standard Test Access Port interface. See IEEE Standard 1149.1-1990 and IEEE Standard 1149.1a-1993.

Vcc, GND (Positive Power Supply, Ground)
The main power supply connections to the ATMCAM.

HARDWARE INTERFACE

A simple way of controlling the ATMCAM is by memory mapping its address and control space into the address space of a local processor. Figure 4 shows a simplified system architecture using a generic processor and an ATMCAM. There are two areas of random access memory, one mapped to the address space of the processor, and the other addressed by the pointer generated by the ATMCAM. This second area of memory holds associated data such as translated header information.

Figure 5 shows details of the memory mapping. Twelve processor address lines convey address and control information to the AC11–0 inputs of the ATMCAM, and one more address line is fed to its /AV input. The address line serves to separate the address and control space into two areas of the local processor address space.

The address space below 1MB is mapped to the RAM, while the address space above 1MB is mapped to the CAM. Reading or writing to the lower half of the CAMs address space will perform random access to the CAM array, while accessing the upper half will execute control states. The simple decoding scheme shown in Figure 5 will create CAM address aliases that can be avoided by using a tighter decoding structure for the upper-order address lines from the local processor.

In Figure 5, note that the RAM area is not the associated data memory addressed by the output from the ATMCAM. It is system RAM addressed by the local processor. The associated data memory addressed by the ATMCAM is not shown in Figure 5 for the sake of clarity.

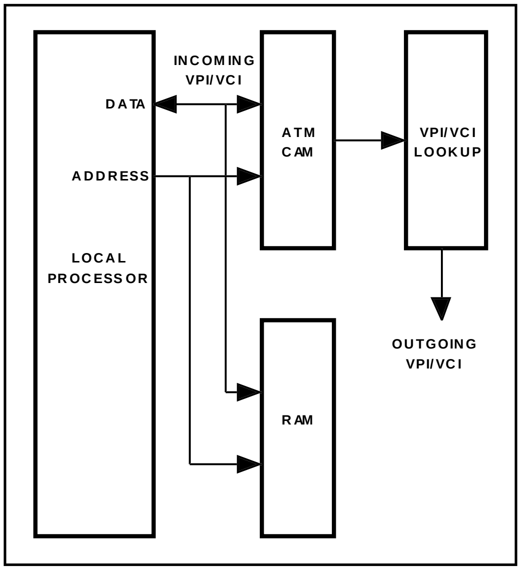

Figure 4: Simplified ATMCAM System Block Diagram

In the following code examples, and throughout this document, the Intel™ assembly language conventions are used, and are of the general form:

OPCODE destination, source

where OPCODE is the instruction mnemonic, destination is the destination of any data transfer, and source is the source of any data transfer.

Following are two examples of how the mapping of address and control of the ATMCAM works in this architecture. The first example executes a Compare cycle using the data

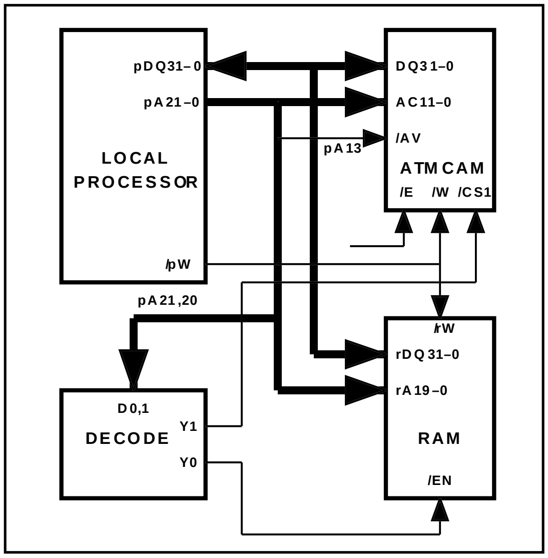

Figure 5: ATMCAM System Architecture

102000H	
101FFFH 101000H	CAM CONTROL SPACE
100FFFH 100000H	CAM ADDRESS SPACE
0FFFFFH 000000H	RAM

Table 2: ATMCAM System Memory Map

on the DQ bus, and writes that data to the Comparand register, the comparison is masked by Mask register 3. The value to be compared, the incoming VPI/VCI, is preloaded into the EAX register of the local processor. The /AV line is HIGH during the ATMCAM Write cycle, so the value on the AC lines is interpreted as a control state and not an address.

The ATMCAM control state that performs the required function is:

```
CMPW DQ {MRnnn}

        with AC0–11 = xxx nnn 011 010 = 0DAH
        xxx = 000, nnn = 011 for Mask register 3
```

First define an absolute value for the instruction mapped to the CAM control space starting at 101000H and offset by the control state value 0DAH. Next execute the MOVD instruction to write the contents of the EAX register to the memory address defined by CMPW_DQ_M3.

```
CMPW_DQ_M3 EQU 1010DAH
MOVD [CMPW_DQ_M3] EAX
```

This instruction assumes that the segment offset applied to the direct address is zero. A better method for selecting masking is presented later. Also using immediate operands cuts down the number of instructions needed.

The second example writes data at a specified address within the CAM. In this case the /AV line is LOW during the ATMCAM Write cycle, so the value on the AC lines is interpreted as an address and not a control state.

Again assume that the data to be written is loaded in the EAX register and location 0 is to be written in the CAM. CAM location 0 maps to 100000H in the processor address space. Therefore, using indirect addressing via the EDI register and executing the following instructions causes the contents of the EAX register to be written to location 0 in the CAM:

```
MOV EDI,100000H
MOV [EDI],EAX
```

Although this is a very simple model, it provides a useful vehicle for demonstrating how to control the ATMCAM through software. All subsequent examples are based on this architecture. It can be easily adapted to other processors and architectures.

For reference, Table 3 on page 8 shows a summary of the ATMCAM control states. Refer to the MU9C4320L data sheet for a full description of these control states.

System Timing Design Considerations

The architecture shown in Figure 5 assumes that certain timing relationships required by the ATMCAM are met. In particular, it is the timing of /E with respect to the data and control lines that is of importance. The detailed timing design of a system is dependent on system specific aspects such as clock speed, processor type, and system memory architecture. Therefore, this discussion concentrates on examining the timing parameters that must be taken into account when designing the ATMCAM into a system, rather than providing a specific design solution.

The timing for the ATMCAM cycles is relative to the /E input. This signal defines the extent of all types of cycle, Read, Write, and Compare. /E registers the control and data inputs on its falling edge. Figure 6 shows the control and data inputs with respect to /E. The signals that are registered are /CS1, /CS2, /W, /AV, VB/, AC11–0, and DQ31–0 on a Write cycle. The setup and hold times for these signals with respect to /E are 3ns.

In the memory mapped architecture of Figure 5, the setup timing requirement means that /E must be delayed by at least 3ns from the time that the address lines are stable PLUS the delay through the decoder to ensure correct registering of the data and control lines.

Data read on the DQ bus is enabled when /E is LOW, as shown in Figure 7. At the beginning of a Read cycle the DQ bus comes out of high-impedance after a delay, and data becomes stable after an access time of a maximum of 70ns from /E going LOW. When /E goes HIGH, the DQ bus returns to its high-impedance state. The data remains stable on the bus for a minimum of 2ns after /E has gone HIGH. Note that the minimum /E LOW time is 75ns, which gives a data capture window of 7ns.

In the architecture shown in Figure 5, /E should be held LOW for the duration of the memory access cycle of the processor to ensure that data is read into the processor correctly. /E is the only input signal that can terminate a cycle because all others are registered as /E goes LOW.

Figure 8 shows the timing relationship between /E and the match address on the PA:AA bus. The /E cycle has executed a comparison in the ATMCAM, but the results of the comparison do not appear on the PA:AA bus until after /E line goes HIGH. This timing allows access to the external RAM to be as long as a full /E cycle. Therefore, access to the associated data in the external RAM is pipelined; while the access with the address generated by the previous Compare cycle is going on in the RAM, the ATMCAM can be performing another comparison. If the processor is able to supply comparands at a suitable rate, Compare cycles can run continuously at the /E cycle rate (up to 70ns).

It is important to note that the PA:AA lines can only change while /E is HIGH. Once /E goes LOW, the PA:AA lines are latched. In a vertically cascaded system, there is a delay in the daisy chain signal propagating from one device to the next. Therefore, the results of comparison are delayed from /E going high to when the daisy chain has settled. The daisy chain must have settled before /E is pulled LOW for the next cycle. The timing details of this requirement are discussed in the Vertical Cascading section.

CONTROL MODES

The ATMCAM has two main modes of operation: Hardware Control and Software Control. There is also a Diagnostic mode that is a variation of Software Control. Hardware Control mode gives the fastest performance and should be preferred in most applications. Software Control mode provides a convenient method of control in less time critical systems.

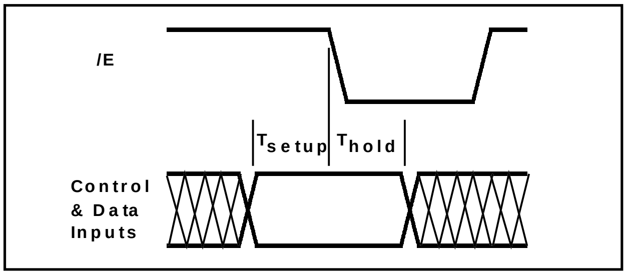

Figure 6: ATMCAM Setup and Hold Timing

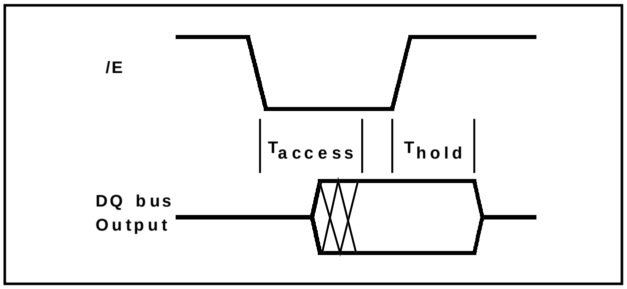

Figure 7: ATMCAM Read Data Access

The default mode is Software Control and is set after a hardware reset; a software reset does not change the mode of operation. The operating mode is then selected through the Configuration register. The Initialization section gives details of how to set the operating mode in the ATMCAM.

Hardware Control

Hardware Control is the normal mode of operation for the ATMCAM. All previous and subsequent sections in this Application Note use Hardware Control in the examples.

Hardware Control is characterized by the following criteria:

- Control states entered through the AC bus
- Random addressing through the AC bus or the Address register
- Results of comparison accessed through the PA:AA bus, /MV, and /MF
- Primary chip select through /CS1 and /CS2

The architecture shown in Figure 5 uses Hardware Control. Note that some Software Control facilities can be used in Hardware Control mode, such as the Status register, and vice versa, for example the Device Select register.

By applying the memory mapping shown in Table 2 on page 5, the ATMCAM control states map into the address space of the local processor and are defined as constants associated with the control states. Note that there is repetition of values between some Read and Write cycles. These values could be defined as a single constant, but for clarity, they are defined separately.

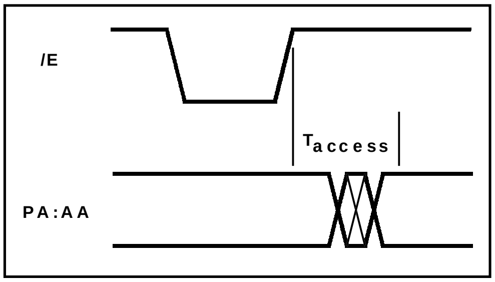

Figure 8: ATMCAM Match Address Access

CONTROL STATE	/W = LOW	/W = HIGH
Write/Read Memory		
A11–0	WR aaa	RD aaa
xxx nnn 000 000	WR [AR] {MRnnn}	RD [AR]
xxx nnn 000 001	WR [NFA] {MRnnn}	RD [HPM]; INC MA
xxx nnn 000 010	WR [HPM] {MRnnn}	RD [HPM]
xxx nnn 000 011	NOP	RD NFA
Write/Read Register		
xxx nnn 000 100	WR AR {MRnnn}	RD AR
xxx nnn 000 101	WR CR {MRnnn}	RD CR
xxx nnn 000 110	WR FR {MRnnn}	RD FR
xxx nnn 000 111	RESERVED	RD SR
xxx nnn 001 000	WR DS {MRnnn}	RD DS
xxx nnn 001 001	WR MRnnn	RD MRnnn
Data Move		
xxx nnn 001 100	MOV [AR],CR {MRnnn}	MOV CR,[AR] {MRnnn}
xxx nnn 001 101	MOV [NFA],CR {MRnnn}	RESERVED
xxx nnn 001 110	MOV [HPM],CR {MRnnn}	MOV CR,[HPM] {MRnnn}
Comparison		
xxx nnn 011 000	CMP CR {MRnnn}	RESERVED
xxx nnn 011 001	CMP DQ {MRnnn}	RESERVED
xxx nnn 011 010	CMPW DQ {MRnnn}	RESERVED
xxx nnn 011 011	INC MA	RESERVED
Set Validity		
xxx xxx 100 000	SET V@[AR]	RD V@[AR]
xxx xxx 100 001	RST V@[AR]	RESERVED
xxx xxx 100 010	RST V@[HPM]	RESERVED
xxx xxx 100 011	RST V@AML	RESERVED
Address Register Control		
xxx xxx 100 100	Inc AR	RESERVED
xxx xxx 100 101	Dec AR	RESERVED
xxx nnn 100 110	WR [AR]+ {MRnnn}	RD [AR]+
xxx nnn 100 111	WR [AR]- {MRnnn}	RD [AR]-
VP Table Control		
xxx xxx 101 000	SET VP@[AR]	RD VP@[AR]
xxx xxx 101 001	RST VP@[AR]	RESERVED
Initialization		
xxx xxx 111 100	WR PA	RESERVED
xxx xxx 111 101	RST FF	RESERVED
xxx xxx 111 111	RST	RESERVED

Key:

aaa	Address	CR	Comparand Register	MRnnn	Mask Register nnn
AML	All Matching Locations	DS	Data Select Register	NFA	Next Free Address
AR	Address Register	HPM	Highest Priority Match	PA	PageAddress
[AR]+	Autoincrement AR	FF	Full Flag	SR	Status Register
[AR]-	Autodecrement AR	FR	Configuration Register	V	Validity
CMP/W	Compare/and Write	MA	Match Address	VP	VP Table Bit

Table 3: Control State Summary

```
        ;MEMORY CYCLES
WR_at_AR EQU 101000H                ;WRITE CYCLE
WR_at_NFA EQU 101001H               ;WRITE CYCLE
WR_at_HPM EQU 101002H               ;WRITE CYCLE
NOP EQU 101003H                     ;WRITE CYCLE
RD_at_AR EQU 101000H                ;WRITE CYCLE
RD_at_HPM_INC_MA EQU 101001H        ;READ CYCLE
RD_at_HPM EQU 101002H               ;READ CYCLE
RD_NFA EQU 101003H                  ;READ CYCLE

        ;REGISTER CYCLES
WR_AR EQU 101004H                   ; WRITECYCLE
WR_CR EQU 101005H                   ;WRITE CYCLE
WR_FR EQU 101006H                   ;WRITE CYCLE
WR_DS EQU 101008H                   ;WRITE CYCLE
WR_MR EQU 101009H                   ;WRITE CYCLE
RD_AR EQU 101004H                   ;READ CYCLE
RD_CR EQU 101005H                   ;READ CYCLE
RD_FR EQU 101006H                   ;READ CYCLE
RD_SR EQU 101007H                   ;READ CYCLE
RD_DS EQU 101008H                   ;READ CYCLE
RD_MR EQU 101009H                   ;READ CYCLE

        ;DATA MOVE CYCLES
MOV_at_AR_CR EQU 10100CH            ;WRITE CYCLE
MOV_at_NFA_CR EQU 10100DH           ;WRITE CYCLE
MOV_at_HPM_CR EQU 10100EH           ;WRITE CYCLE
MOV_at_AR_MR EQU 10100FH            ;WRITE CYCLE
MOV_CR_at_AR EQU 10100CH            ;READ CYCLE
MOV_CR_at_HPM EQU 10100EH           ;READ CYCLE
MOV_MR_at_AR EQU 10100FH            ;READ CYCLE
        ;COMPARISON
CMP_CR EQU 101018H                  ;WRITE CYCLE
CMP_DQ EQU 101019H                  ;WRITE CYCLE
CMPW_DQ EQU 10101AH                 ;WRITE CYCLE
INC_MA EQU 10101BH                  ;WRITE CYCLE

        ;SET VALIDITY CYCLES
SET_V_at_AR EQU 101020H             ;WRITE CYCLE
RST_V_at_AR EQU 101021H             ;WRITE CYCLE
RST_V_at_HPM EQU 101022H            ;WRITE CYCLE
RST_V_at_AML EQU 101023H            ;WRITE CYCLE
RD_V_at_AR EQU 101020H              ;READ CYCLE
        ;ADDRESS REGISTER CONTROL CYCLES
INC_AR EQU 101024H                  ;WRITE CYCLE
DEC_AR EQU 101025H                  ;WRITE CYCLE
WR_at_AR_INC EQU 101026H            ;WRITE CYCLE
WR_at_AR_DEC EQU 101027H            ;WRITE CYCLE
RD_at_AR_INC EQU 101026H            ;READ CYCLE
RD_at_AR_DEC EQU 101027H            ;READ CYCLE

        ;VP TABLE CONTROL CYCLES
SET_VP_at_AR EQU 101028H            ;WRITE CYCLE
RST_VP_at_AR EQU 101029H            ;WRITE CYCLE
RD_VP_at_AR EQU 101028H             ;READ CYCLE
```

```
        ;INITIALIZATION
WR_PA EQU 10103CH                   ;WRITE CYCLE
RST_FF EQU 10103DH                  ;WRITE CYCLE
RST EQU 10103FH                     ;WRITE CYCLE
```

These constants correspond to the control states shown in Table 3. Note that there are several control states that use a mask register. To cut down the number of constants needed to define the control states, the mask register offset value is added to the control state constant.

To define the mask register constants, note that the bits of the instruction that define the mask register to use are DQ8–6. A value of 000 means ‘No Mask’, 001 means ‘Use Mask Register 1’ etc. Therefore, the mask register constants are offset to give a value that is added to the instruction value to produce the required masking function.

```
;MASK REGISTER OFFSETS
        MR_0 EQU 000H - NO MASK
        MR_1 EQU 040H
        MR_2 EQU 080H
        MR_3 EQU 0C0H
        MR_4 EQU 100H
        MR_5 EQU 140H
        MR_6 EQU 180H
        MR_7 EQU 1C0H
```

The ATMCAM is controlled by sequences of MOV instructions executed in the local processor and addressed to the memory space that is mapped to the control space of the ATMCAM. Here are some examples:

```
;WRITES DATA IN EAX REGISTER TO NEXT FREE
;ADDRESS IN CAM ARRAY
                MOVD [WR_at_NFA],EAX
;WRITES NEW COMPARAND TO EAX REGISTER AND
;MOVES IT TO THE COMPARAND REGISTER
                MOVD EAX,NEW_COMPARAND
                MOVD [WR_CR],EAX
;INITIATE COMPARISON
                MOVD [CMP_CR],EAX
```

Note that in the last line of the code segment the contents of the EAX register are used as dummy data. The ATMCAM control state CMP CR{Mnnn} does not have a data transaction on the DQ bus, even though it is performed through a Write cycle. The incoming data on the DQ bus is ignored by the ATMCAM.

A more efficient way to achieve the same result is to load the comparand and to do the compare in a single cycle. The last two instructions in the previous code can be combined into the single instruction:

```
;MOVES CONTENTS OF EAX REGISTER TO COMPARAND
;REGISTER AND INITIATES COMPARISON
        MOVD [CMPW_DQ],EAX
```

To use a mask register, the mask offset is added to the control state constant. The combined value creates the control state with the mask activated. A mask offset value of zero implies 'No Mask', therefore by simply not adding an offset, the control state is executed without using a mask register. The previous instruction is executed using Mask register 7 as follows:

```
;MOVES CONTENTS OF EAX REGISTER TO COMPARAND
;REGISTER AND INITIATES COMPARISON WITH MASK
;REGISTER 7
        MOVD [CMPW_DQ+MR_7],EAX
```

Reading information into the local processor from the ATMCAM is done in a similar way:

```
;READS THE CONTENTS OF THE ATMCAM STATUS
;REGISTER INTO THE EAX REGISTER
                MOVD EAX,[RD_SR]
```

The principles described here can be applied to all the functionality of the ATMCAM. The examples that are included in the sections following the Software Control section use this memory mapped architecture as the platform for the code segments.

Software Control

Software Control is characterized by the following criteria:

- Control states entered through the Instruction register
- Random addressing through the Address register only
- Results of comparison accessed through the Status register
- Primary chip select through the Device Select register

In Software Control mode the AC11–0 lines are not used, instead the instruction is loaded from the DQ11–0 lines into the Instruction register. The /AV line is used to distinguish between instruction and data on the DQ lines. When it is HIGH, the data on the DQ lines is interpreted as an instruction and is loaded into the Instruction register.

The instructions are directly analogous to the control states for any operation that does not involve data transfer on the DQ31–0 lines. These instructions are executed during the same cycle as the instruction is loaded.

To distinguish between Read and Write control states, DQ12 is used to indicate which type of instruction should be executed. This mechanism is necessary because the /W line is used to control the instruction load. When DQ12 is LOW, the Write cycle instruction is executed; when DQ12 is HIGH the Read cycle instruction is executed.

When the instruction expects data to be written or read from the DQ31–0 lines, that instruction is loaded into the Instruction register during the Write cycle with /AV HIGH, then the next data Read or Write cycle with /AV LOW executes the previously loaded instruction using the DQ31–0 bus for the data transaction.

The instruction is persistent, in other words, if no other instruction is loaded into the Instruction register, subsequent data transactions with the /AV line LOW will be executed according to the instruction currently loaded in the Instruction register.

Therefore, instructions that involve data transactions on DQ31–0 and are executed on a subsequent Read or Write cycle with the /AV line LOW, are all of the Read/Write memory and Read/Write register instructions, Read Validity, Write DQ3–0 to Page address in Configuration register of Highest Priority Empty device, and Set Full flag. All other instructions are executed in a single cycle with the state of DQ12 being interpreted as the state of the /W line during the equivalent hardware control state.

Once a Compare cycle has been executed, the results are read back through the Status register. Although there is an instruction that reads the Status register, Software Control mode provides a short cut: a Read cycle with the /AV line HIGH accesses the Status register.

Figure 9 on page 11 shows an architecture that is suitable for Software Control mode. Note that it is simpler than the Hardware Control architecture shown in Figure 5. There is a simplified memory mapping scheme, which is shown in Table 4.

In Figure 9 the ATMCAM is mapped into the lower 2 MB of the local processor address space. That area is divided into two with the processor address line pA20 being connected to the /AV line. The first megabyte of processor address space maps the ATMCAM instruction execution, while the second megabyte maps instruction loading. Note that a tighter decoding scheme could map the ATMCAM into only two locations.

Using the architecture in Figure 9 for the software model, following are some examples that show how to control the ATMCAM in Software Control mode.

First, define the memory locations that are mapped to perform instruction load and instruction execute.

```
        :CONTROL LOCATIONS
                I_LOAD EQU 100000H
                I_EXEC EQU 000000H
```

Next define all the instructions that can be loaded and executed in the ATMCAM. The Write cycles have DQ bit 12 set LOW, and the READ cycles have DQ bit 12 set HIGH.

```
        ;MEMORY CYCLES
WR_at_AR EQU 0000H                   ;WRITE CYCLE
WR_at_NFA EQU 0001H                  ;WRITE CYCLE
WR_at_HPM EQU 0002H                  ;WRITE CYCLE
NOP EQU 0003H                        ;WRITE CYCLE
RD_at_AR EQU 1000H                   ;READ CYCLE
RD_at_HPM_INC_MA EQU 1001H           ;READ CYCLE
RD_at_HPM EQU 1002H                  ;READ CYCLE
RD_NFA EQU 1003H                     ;READ CYCLE

        ;REGISTER CYCLES
WR_AR EQU 0004H                      ;WRITE CYCLE
WR_CR EQU 0005H                      ;WRITE CYCLE
WR_FR EQU 0006H                      ;WRITE CYCLE
WR_DS EQU 0008H                      ;WRITE CYCLE
WR_MR EQU 0009H                      ;WRITE CYCLE
```

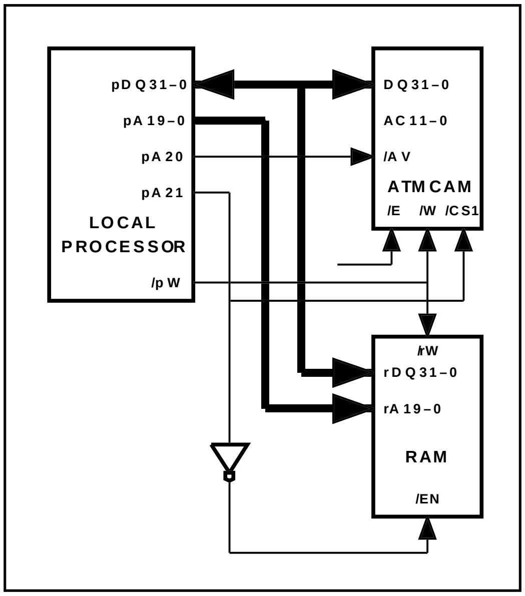

Figure 9: ATMCAM System Architecture for Software Control Mode

```
RD_AR EQU 1004H                      ;READ CYCLE
RD_CR EQU 1005H                      ;READ CYCLE
RD_FR EQU 1006H                      ;READ CYCLE
RD_SR EQU 1007H                      ;READ CYCLE
RD_DS EQU 1008H                      ;READ CYCLE
RD_MR EQU 1009H                      ;READ CYCLE

        ;DATA MOVE CYCLES
MOV_at_AR_CR EQU 000CH               ;WRITE CYCLE
MOV_at_NFA_CR EQU 000DH              ;WRITE CYCLE
MOV_at_HPM_CR EQU 000EH              ;WRITE CYCLE
MOV_at_AR_MR EQU 000FH               ;WRITE CYCLE
MOV_CR_at_AR EQU 100CH               ;READ CYCLE
MOV_CR_at_HPM EQU 100EH              ;READ CYCLE
MOV_MR_at_AR EQU 100FH              ;READ CYCLE

        ;COMPARISON
CMP_CR EQU 0018H                      ;WRITE CYCLE
CMP_DQ EQU 0019H                      ;WRITE CYCLE
CMPW_DQ EQU 001AH                     ;WRITE CYCLE
INC_MA EQU 001BH                      ;WRITE CYCLE

        ;SET VALIDITY CYCLES
SET_V_at_AR EQU 0020H

        ;WRITE CYCLE
RST_V_at_AR EQU 0021H                 ;WRITE CYCLE
RST_V_at_HPM EQU 0022H                ;WRITE CYCLE
RST_V_at_AML EQU 0023H                ;WRITE CYCLE
RD_V_at_AR EQU 1020H                  ;READ CYCLE

        ;ADDRESS REGISTER CONTROL CYCLES
INC_AR EQU 0024H                      ;WRITE CYCLE
DEC_AR EQU 0025H                      ;WRITE CYCLE
WR_at_AR_INC EQU 0026H                ;WRITE CYCLE
WR_at_AR_DEC EQU 0027H                ;WRITE CYCLE
RD_at_AR_INC EQU 1026H                ;READ CYCLE
RD_at_AR_DEC EQU 1027H                ;READ CYCLE
```

500000H	
4FFFFFH 300000H	RAM (alias)
2FFFFFH 200000H	RAM
1FFFFFH 100000H	CAM - load Instruction Register
0FFFFFH 000000H	CAM - execute instruction

Table 4: ATMCAM System Memory Map for Software Control Mode

```
        ;VP TABLE CONTROL CYCLES
SET_VP_at_AR EQU 0028H          ;WRITE CYCLE
RST_VP_at_AR EQU 0029H          ;WRITE CYCLE
RD_VP_at_AR EQU 1028H           ;READ CYCLE

        ;INITIALIZATION
WR_PA EQU 003CH                 ;WRITE CYCLE
RST_FF EQU 003DH                ;WRITE CYCLE
RST EQU 003FH                   ;WRITE CYCLE
```

Next define the mask register constants in the same way as for Hardware Mode:

```
        ;MASK REGISTER OFFSETS
                MR_0 EQU 000H
                MR_1 EQU 040H
                MR_2 EQU 080H
                MR_3 EQU 0C0H
                MR_4 EQU 100H
                MR_5 EQU 140H
                MR_6 EQU 180H
                MR_7 EQU 1C0H
```

Controlling the ATMCAM is simply a matter of writing instruction constants to the location mapped to the Instruction register. If the instruction does not need a data transaction it executes immediately.

For example, the following processor instruction performs a comparison between the current contents of the Comparand register and the ATMCAM memory array:

```
MOVD [I_LOAD],CMP_CR
```

and to perform the same comparison, but masking the comparison with Mask register 3:

```
MOVD [I_LOAD],CMP_CR+MR_3
```

Both of the previous instructions execute in the ATMCAM in a single cycle because they do not need a data transaction on the DQ bus.

When the comparand is presented on the DQ bus, two processor cycles are needed. The first one loads the instruction into the ATMCAM, but does not execute until the next instruction, which provides the data for comparison. The ATMCAM automatically distinguishes between instructions that execute in a single cycle and those that need a subsequent data transaction. In the following code, the value for comparison is held in the EAX register of the local processor:

```
MOVD [I_LOAD],CMP_DQ
MOVD [I_EXEC],EAX
```

There is an ATMCAM control state that reads the Status register in the same way as any other register is read:

```
RD SR
```

However, in Software Control mode loading and executing this instruction would take two cycles:

```
MOVD [I_LOAD],RD_SR
MOVD EAX, [I_EXEC]
```

Therefore, Software Control mode provides a short cut to accessing the Status register, where the results of a comparison, and other information, are available. A Read cycle with the /AV line HIGH accesses the Status register in a single cycle. This means that reading from the I_LOAD location will access the Status register directly:

```
MOVD EAX,[I_LOAD]
```

This local processor instruction loads the EAX register with the contents of the ATMCAM Status register. Note that in Software Diagnostic mode, this instruction accesses the contents of the ATMCAM Instruction register instead of the Status register. This function is for diagnostic purposes only and would not normally be used.

One feature of the ATMCAM is its ability to be selected either through the hardware Chip Select lines, /CS1 and /CS2, or through software control through the Device Select register. When the Device Select register contains the same value as the Page address and the SELECT ENABLE bit of the Device Select register is LOW the device is then selected. Table 5 shows the Device Select Register Bit assignments.

If any one of the selection mechanisms is selected then the device is selected:

```
If (/CS1=LOW) or (/CS2=LOW) or
        (DS8=LOW) and (DS3–0=PA3–0)
        then device_selected = true
else device_selected = false;
```

The idea behind the device select mechanism is to cut down the number of select lines needed in a multiple ATMCAM system. One chip select line is connected to all the /CS1 inputs for broadcast control. For example, for initiating a comparison across the whole system. The second chip select line, /CS2, is intended for the selection of individual devices within the system for random access, or for segmenting the CAM array.

The Device Select register provides a way to access individual devices through software without the need for the individual connections to the /CS2 inputs. This is a suitable selection method for non-time-critical applications where preserving control lines is more important than raw speed, as is the case in systems using Software Control mode.

The Device Select register is used by broadcasting the Page address of the device to be selected, and the Enable Select bit set LOW, to the Device Select registers of all devices, enabled by /CS1 being LOW. Subsequent cycles with /CS1 HIGH will only operate in the device whose DS3–0 bits match with its PA3–0 bits. At any time the selection can be overridden by performing a cycle with the broadcast /CS1 lines pulled LOW.

Figure 10 shows a hardware configuration suitable for using the Device Select register. The system has four vertically cascaded ATMCAMS, with all lines common between them. Details of vertical cascading are discussed in a later section, and this discussion focuses only on the software device selection mechanism.

Address bits pA21 and pA22 are combined through an OR gate to generate /CS1 for the ATMCAMs, and are combined through a NAND gate to generate an enable for the RAM. This decoding leaves an empty space in the memory map between 200000H and 5FFFFFH.

This empty address space can be used to access the ATMCAM device selected through its Device Select register. The instruction constants are offset into that address space by adding the address of the starting location to the instruction constant.

First, define the offset into the empty memory area:

```
DS_OFFSET EQU 200000H
```

Now assume that the Page Address values of the four ATMCAMs have already been set to 0H, 1H, 2H, and 3H. Next define the device select and deselect values for the CAMs:

```
SEL_CAM0 EQU 000H
SEL_CAM1 EQU 001H
SEL_CAM2 EQU 002H
SEL_CAM3 EQU 003H
DESEL_CAMS EQU 100H
```

Note that bit 8 of the select values is set LOW, and bit 8 of the deselect value is set HIGH. When bit 8 is HIGH, bits 3–0 are 'don't care'.

The page address value for CAM1 is broadcast to the Device Select registers of all ATMCAM devices, with the Select Enable bit set LOW.

```
MOVD [I_LOAD],WR_DS
MOVD [I_EXEC],SEL_CAM1
```

All subsequent reads and writes to the empty address space will access ATMCAM number 1 because its Page address and Device Select values match, and the Select Enable bit is set LOW. By adding the offset to the I_LOAD and I_EXEC constants, an instruction loaded into the Instruction register will be executed in the one selected device only. For example, when executed after the two previous instructions, the following code runs a Compare cycle in CAM1 only.

```
MOVD [I_LOAD+DS_OFFSET],CMP_CR
```

The instruction is similar to one that requires a data transaction on the DQ bus. Here data in the EAX register of the local processor is compared with the contents of CAM1, and is also written to the Comparand register of CAM1:

```
MOVD [I_LOAD+DS_OFFSET],CMPW_DQ
MOVD [I_EXEC+DS_OFFSET], EAX
```

The same function, but with the comparison masked by Mask register 3, is executed in CAM1 using the following instructions in the local processor.

```
MOVD [I_LOAD+DS_OFFSET],CMPW_DQ+MR_3
MOVD [I_EXEC+DS_OFFSET], EAX
```

To return to normal operation, the Device Select registers are loaded with the value that sets the Select Enable bit HIGH. This instruction is broadcast to all devices in the normal control space of the ATMCAMs, thereby causing /CS1 to go LOW during the cycle.

DS31–9	RESERVED (set to zero)
DS8	0 Enable Select 1 Disable Select
DS7–4	RESERVED (set to zero)
DS3–0	Device Select Value Selected when DS3–0=PA3–0 AND DS8=LOW

Table 5: Device Select Register Bit Assignments

```
MOVD [I_LOAD],WR_DS
MOVD [I_EXEC],DESEL_CAMS
```

Care must be taken to ensure that no ATMCAM remains enabled through its Data Select register while accessing the RAM. Under such a circumstance, all data to and from the RAM would be mistakenly interpreted as control or data by the enabled ATMCAM.

Once an instruction has been loaded into the Instruction register of an ATMCAM, it is persistent, and any data transfer will be executed in the device according to that instruction. For example the following instructions load zero into the two next free addresses:

```
MOVD [I_LOAD],WR_at_NFA
MOVD [I_EXEC],0H
MOVD [I_EXEC],0H
```

Controlling the Validity Bit

The Validity Bit line, /VB, conveys validity information to and from a location when it is written to or read. When the Validity Bit is LOW at a particular location, that location contains valid contents. Only locations whose validity bits are set LOW will enter into comparison. On Reset, the validity bits of all locations are set HIGH, and the ATMCAM is said to be empty.

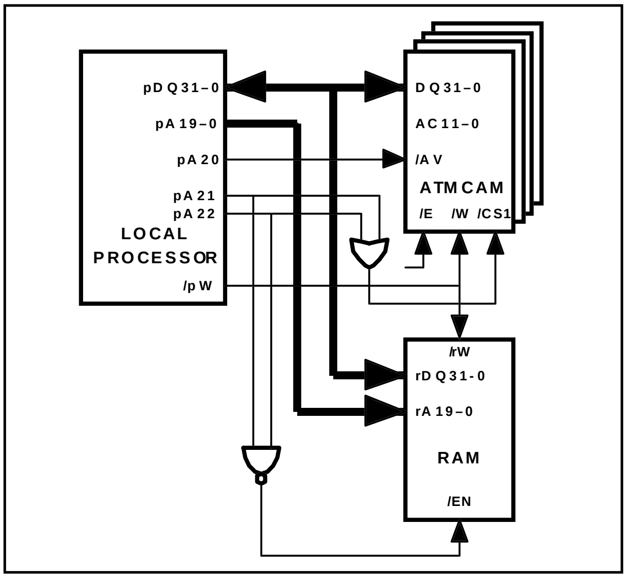

Figure 10: ATMCAM System for using Data Select Register

The Validity Bit is also important in generating the address for Write at Next Free Address cycles. All the validity bits are fed into the Priority Encoder after a Write cycle to determine which is the highest-priority empty location. This mechanism allows the ATMCAM to fill gracefully using Write at Next Free Address cycles even when the filled locations are non-contiguous through the CAM array.

So far, there has been no discussion as to how to control the Validity Bit. It behaves like an extra data bit for all control states that access memory locations either by direct or indirect addressing. For register and other cycles, the line is ignored.

In system architectures that use state machine control, or a local processor with a data bus wider than 32 bits, the /VB line can be tied to one of the high-order data bits. It is set and reset by Write cycles as part of the data field.

When setting up a connection in the ATMCAM, and writing a new VPI/VCI to the CAM array, the /VB line is set LOW during the Write cycle. When tearing down a connection, the validity bit can be reset by running a comparison, and then executing the instruction that resets the Validity Bit at the highest-priority matching location. The following code, which uses the Hardware Control mode of Figure 5, assumes that the VPI/VCI value to be removed is present in the EAX register of the local processor.

```
MOVD [CMP_DQ],EAX
MOVD [RST_V_at_HPM],EAX
```

Note that there is no data transfer associated with the Reset Validity Bit control state, so the contents of the EAX register in the previous second instruction is dummy data.

7FFFFFH 700000H	RAM (alias)
6FFFFFH 600000H	RAM
5FFFFFH 200000H	Empty Address Space
1FFFFFH 100000H	CAM - load Instruction Register
0FFFFFH 000000H	CAM - execute instruction

Table 6: ATMCAM System Memory Map for using Data Select Register

There still remains the problem of how to control the Validity Bit from a 32-bit processor. The simple answer is to drop the most significant ATMCAM data line, DQ31, in favor of /VB. In other words, DQ31 remains unconnected, and /VB connects to the local processor line pDQ31. Because the GFC/VPI/VCI field width is only 28 bits, dropping one data bit is acceptable, and there are still three general purpose bits available, though these may also be masked off or remain unconnected.

When one or more DQ lines remain unconnected, compare operations need to use a mask register with its most significant bit, or bits, set HIGH to prevent random data in DQ31 from upsetting the results of comparison. Using this method, the Validity Bit is read and written from the most significant bit of the local processor data bus. It is essential to provide a source of /VB because it is fundamental to the operation of the ATMCAM.

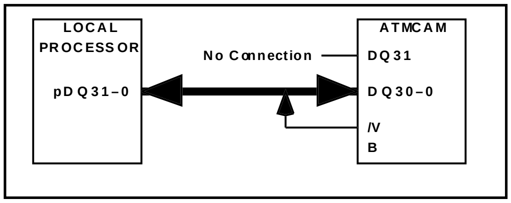

Figure 11: Controlling the Validity Bit from a 32-bit Processor

Active Address Interface

The Active Address outputs (AA11–0), and the Page Address outputs (PA3–0) is the address of the memory location where the most recent transaction occurred, whether it was a Read, Write, or Compare cycle. The type of address on the PA:AA bus depends on the type of the last memory cycle. Table 7 lists the active address type for all possible types of memory cycle.

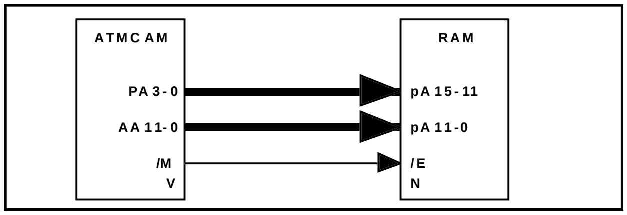

Figure 12: Addressing External RAM with the PA:AA Bus

Compare Cycle:
1. If Match in CAM: CAM Match Address
2. If Match in VP Table: VP Table Match Address
3. No Match: All '1's

Write at Next Free Address Cycle:
1. If system was not full: Address at which Write took place
2. If system was full: All '1's

Random Access Read or Write Cycle (direct or indirect addressing):
1. Address at which Write took place
2. Broadcast Write enables all PA:AA lines (select with /OE)

Random Access Read or Write to VP Table:
1. VP Table address at which Write took place
2. Only PA:AA lines of lowest-priority device active

Notes:
1. PA:AA lines are only active when /OE is LOW
2. All other control states leave PA:AA unchanged
3. Match flags hold the results of the most recent comparison regardless of the address type on PA:AA

Table 7: Active Address Output for Different Memory Cycle Types

The PA:AA bus is used to address external RAM for lookup of outgoing header information, as shown in Figure 12. As data is written into the ATMCAM, whether by direct, indirect, or Write Next Free addressing, the address written is output on the PA:AA lines. This address allows the lookup data to be written into the equivalent location in the external RAM. That same address is output after a Compare cycle with that data as the comparand. Therefore, the ATMCAM takes care of address generation for both random access and associative cycles.

Only one device in a multiple CAM system will respond to being enabled by the /OE line, except in the case of a broadcast Write cycle. Under this circumstance, the designer must ensure that only one /OE line is pulled LOW to avoid bus contention. In all other cases the /OE lines can all be pulled LOW together.

Vertical Cascading

The ATMCAM supports vertical cascading through a simple daisy chain scheme. Match, multiple match, and full conditions are fed from device to device in the chain, giving system level indication of these conditions. It also provides device level prioritization.

The daisy chaining works to any practical depth, although it does involve a small timing overhead. In most cases the overhead can be absorbed, but in very large systems, the match conditions would be better handled by external prioritizing logic. Figure 13 shows a vertically cascaded system of ATMCAMs.

The Match Flag daisy chain has an Match Input line, /MI, and a Match Flag output, /MF. The /MF output of one device is connected to the /MI line of the next device in the daisy chain. The first in the chain is the highest-priority device, and the last in the chain is the lowest-priority. Therefore, the device-level prioritization is hardwired.

If the /MI line is HIGH, an ATMCAM assumes that there is no higher-priority device that contains a match, therefore it is free to condition the /MF output based on its own match results. If, on the other hand, the /MI line is LOW, the ATMCAM assumes that there is at least one higher-priority device that contains a match, so it ignores its own match conditions, and sets its /MF LOW. Table 8 shows the truth table for Match Flag cascading.

The Match Flag daisy chain is an important mechanism for controlling access to an ATMCAM in a system containing multiple devices. For example, after a Compare cycle, only the device containing the highest-priority match will be able to output its match address value on the PA:AA bus. All other devices in the daisy chain will have their PA:AA lines held in high impedance, regardless of their match results and the state of their /OE lines. Also, a broadcast Read Status register, will only be responded to by the device containing the highest-priority match.

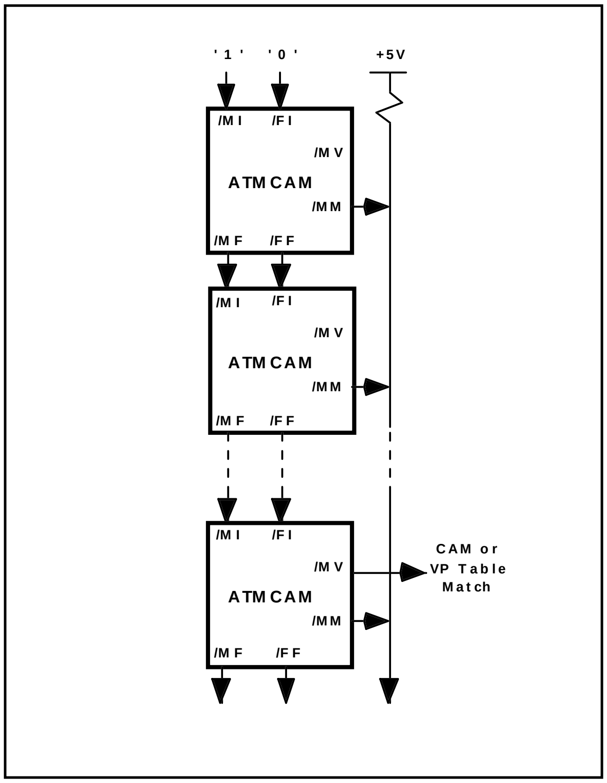

Figure 13: Vertically Cascaded System of ATMCAMs

/MI	Match?	/MF	Access to Device?
LOW	NO	LOW	NO
LOW	YES	LOW	NO
HIGH	NO	HIGH	NO
HIGH	YES	LOW	YES

Table 8: Match Flag Truth Table

Results from a Compare cycle are unlatched by /E being HIGH. Therefore, the PA:AA value and the /MF and /MV flags do not come available until a delay after /E goes HIGH, as shown in Figure 8. Once the /MF line is stable, the daisy chain can start to resolve. In the worst case the match occurs in the highest-priority device, and must propagate the whole length of the daisy chain before the PA:AA buses and system-level Match flag are valid.

An important consideration in the design of a multiple ATMCAM system is that /E must remain HIGH until the daisy chain prioritization is resolved, because the Match flag conditions and the PA:AA buses are latched while /E is LOW. If the daisy chain is long, then the /E HIGH time must be extended, causing the overall /E cycle time to be extended.

The daisy chain settling time, t_{dc}, is the sum of the Match Flag access time from /E going HIGH, t_{EHMV} and the propagation delay through the daisy chain. So the total worst-case daisy chain settling time from /E going HIGH is:

$$t_{dc} = t_{EHMV} + (n-1)\, t_{MIVMV}$$

where t_{MIVMV} is the propogation delay from /MI to /MF within single device. The data sheet figures for these parameters on a -90 device are:

t_{EHMV}	19ns
t_{MIVMV}	7ns

/FI	Device Full?	/FF	Write Next Free Address?
LOW	NO	HIGH	YES
LOW	YES	LOW	NO
HIGH	NO	HIGH	NO
HIGH	YES	HIGH	NO

Table 9: Full Flag Truth Table

The parameters for a minimum /E Compare cycle are:

t_{ELEL}	90ns	
t_{ELEH}	75ns	
t_{EHEL}	30ns	(derived value)

Therefore, any more than one device in a system will require t_{EHEL} to be increased by 4µs plus 7ns per additional device. Note: data sheet parameters are subject to change; please check with MUSIC for the latest data sheet revision.

In many systems the /E HIGH time will be some convenient multiple of the system clock. In which case there may be some slack in the timings, allowing the daisy chain to settle in the allocated time without having to extend the /E HIGH time. In other cases there may not be such slack, and a decision must be made as to whether the performance of the system is compromised by having to extend the /E cycle.

If the daisy chain settling time proves to be unacceptable, then the alternative is to use external prioritization. Figure 14 shows a system that uses an external priority encoder instead of the Match Flag daisy chain. Each match flag is fed into one input to the priority encoder that returns a prioritized enable signal to the ATMCAM with the highest-priority match.

The outputs from the priority encoder are fed into the respective ATMCAMs /OE and /CS inputs. The /OE lines serve to enable the PA:AA bus in the device containing the highest-priority match. The /CS line need only be connected if the designer wants to run Read Status Register cycles, or to access the highest-priority memory location. Under these circumstances the devices must be selected individually.

Since the devices all generate local match results in parallel, the priority encoder needs to resolve the prioritization in less than the /E HIGH time less the setup time for the /CS input.

The other daisy chain connects the individual Full flags throughout the system. This information is needed to run Write at Next Free address Cycles across the entire system. The /FF line from one device is fed into the /FI line of the next device. Writes at Next Free address fill from the top device down the chain.

If the /FI line to an ATMCAM is LOW, it assumes that all higher-priority devices are full. In which case, if it is not full itself, it will respond to Write at Next Free address Cycles. If the /FI line is HIGH, then the device assumes that there is at least one location free in a higher-priority device, and will not respond to Write at Next Free address Cycles. Table 9 shows the truth table for Full flag cascading.

Multiple match cascading is not done by a daisy chain, but by an open-drain output that is pulled LOW when either an individual device has a multiple match, or when it has a match and the /MI line is LOW.

Neither of these other cascaded signals are as time critical as the Match flag. The System Full condition, which is flagged by the /FF line from the least significant devices, usually represents an error condition, because higher level software would detect that the capacity of the ATM switch had been reached. Similarly, the multiple match flag usually indicates an error because all VPI/VCIs should be unique.

Initialization

The ATMCAM requires a hardware reset by providing a LOW on the /RESET pin after power up to ensure that it is in correct operating condition. The reset initializes the registers and CAM array. All CAM locations are set empty (Validity Bit set HIGH); the Comparand register, Mask registers 1-7, Address register, Instruction register, and Next Free Address register are all set to zero. These conditions also prevail after a software reset. A software reset is initiated through the control state RST.

The reset settings in the Configuration register, Status register, and Device Select register are more intricate, and reflect the default operating conditions of the ATMCAM. Table 10 shows the reset conditions for these registers for both hardware and software reset. Note the differences between the two reset types: for a software reset there is no change to the states set in the Configuration register for Control mode, Lowest-Priority CAM, Enable VP Table, VP Table Address mask, VP Table Page address, Page Address value, or in the Device Select register for Device Select value.

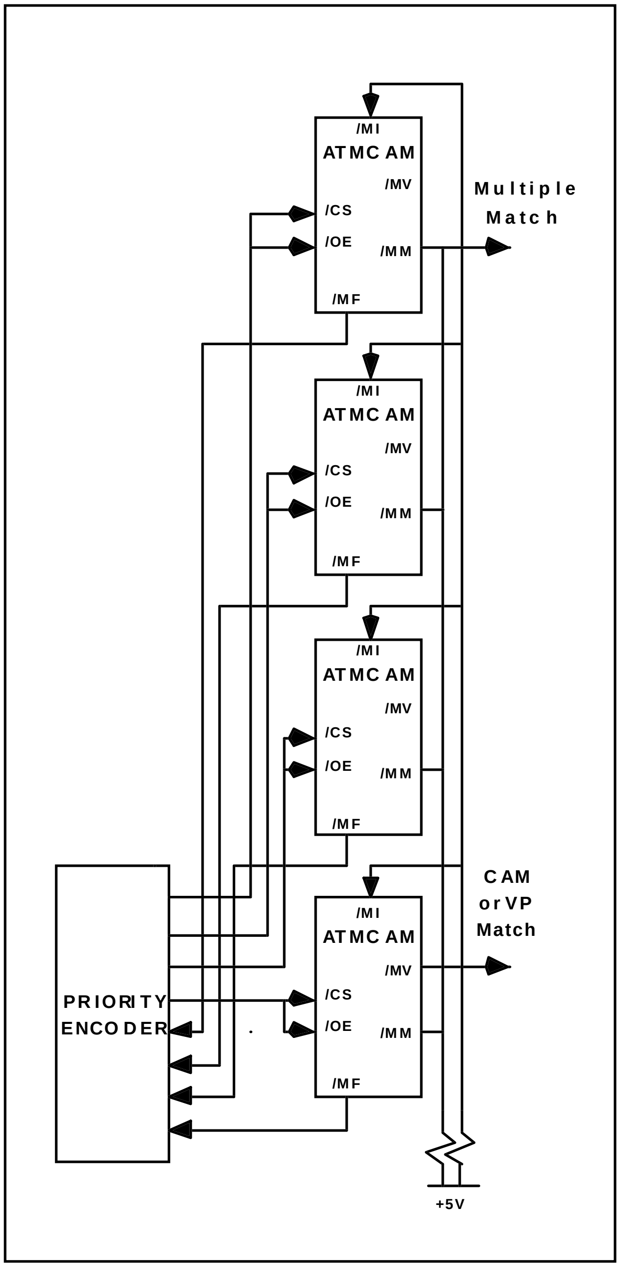

Figure 14: ATMCAM System with External Prioritization

Therefore, the system-level operating conditions of the ATMCAM are preserved after a software reset.

The initialization procedure differs depending on whether the system is designed for Hardware Control mode or Software Control mode, and whether the system contains a single ATMCAM, or more than one. For Hardware Control in a system with a single device, initializing the Configuration register can be done in a single pass. The two changes to the default value are to change from Software Control mode, which is the default, by setting the Configuration Register bits FR27–26 to 00, and to set the device as the lowest-priority CAM, FR25 set to 0. The following code makes these changes, and all subsequent accesses are in Hardware Control mode.

```
MOVD [I_LOAD],WR_FR
MOVD [I_EXEC],01000000H
```

In a multiple ATMCAM system, initialization is done in two steps. First change to Hardware Control mode with a broadcast Software Control sequence as in a single device, and then by a sequence of Configuration Register Write cycles, setting each the Page Address field in each device to its particular value.

Initializing a single ATMCAM, or multiple ATMCAMs, in Software Control mode is similar to Hardware Control mode when individual device selection is through the /CS lines, except that there is no change to the control mode. However, a different situation arises when device selection is done through the Device Select register.

The problem arises because the selection of individual devices is by comparing their Page Address value with the Device Select value. However, after a hardware reset, the Page Address values are all the same, so it is not possible to select an individual device to set its Page Address value.

To overcome this problem, there are two extra control states: WR PA and RST FF. WR PA is broadcast to all devices and writes DQ3–0 to the Page Address field of the Configuration register in the highest-priority empty device; it then sets the Full flag in that device, indicating that it is full. When the WR PA instruction is run the next time, it writes to the next lower-priority device. The sequence continues until all devices are loaded with a unique Page address. Next the RST FF instruction is broadcast, and all the Full flags are reset. The individual ATMCAMs can then be accessed through the Device Select registers. The following code demonstrats this process.

```
MOVD [I_LOAD],WR_PA
MOVD [I_EXEC],0H
MOVD [I_EXEC],1H
MOVD [I_EXEC],2H
MOVD [I_EXEC],3H
MOVD [I_LOAD],RST_FF
```

Lowest-Priority CAM

During initialization the last device in the daisy chain must have its Lowest-Priority CAM (LPC) bit set to zero. This device plays a special role. First, if enabled, it holds the VP Table. Second, it drives the PA:AA bus after a Compare cycle with a default value (all ‘1’s) if there is no match in either the CAM or the VP Table. This action ensures that the PA:AA bus does not float if there is a miss in the system. Similarly, when a Write to Next Free Address cycle is executed into an already full system, the LPC device drives the PA:AA bus with a default value (all ‘1’s).

Using the VP Table

In ATM systems, there are two types of connection, a Virtual Channel Connection (VCC) whose routing is dependent on both the VPI and VCI fields of the ATM cell header, or a Virtual Path Connection (VPC) whose routing is dependent on only the VPI field (or GFC/VPI). Since the GFC/VPI field width is only 12 bits, it can be used to address a 4K-deep table which holds flags to indicate whether that particular value is a VPC.

The ATMCAM contains a VP Table which is a small area of RAM, organized as 4K x 1, which holds VPC flags. Its use is optional, and is selected by setting the Configuration Register bit FR24 LOW. Since it is only 4K deep, a single ATMCAM is required. It resides only in the lowest-priority CAM in a multiple device system. It is given lower priority than a CAM match, which represents a VPI/VCI match, or a VC connection. If a bit at a particular address in the VP Table is set LOW, it indicates that the VPI forming that address is a valid VP connection.

The ATMCAM has two match flags so it can distinguish between a CAM match and a VP Table match or ‘hit’. When it is LOW, the /MV output indicates that there is a valid match in the system. When it is LOW, the /MF output indicates that there has been a match in the CAM array. Table 11 shows the interpretation of the match flags.

Note that the VP Table entries have their own separate Page address. The designer can then locate all VPC lookup information in a separate area of memory. The VP Table Page address is set in Configuration Register bits FR11-8.

Hardware Reset		Software Reset	Function	Reset Condition
Configuration Register				
FR31–29	000	000	Direct Addressing Write Mask	No Direct Address Mask
FR28	0	0	RESERVED	RESERVED
FR27–26	11	No Change	Control Mode	Software Control Mode
FR25	1	No Change	Lowest Priority CAM	Not Lowest Priority
FR24	1	No Change	Enable VP Table	VP Table Disabled
FR23–12	FFFH	No Change	VP Table Address Mask	No VP Table Address Mask
FR11–8	1111	No Change	VP Table Page Address	Zero
FR7–4	0000	0000	RESERVED	RESERVED
FR3–0	1111	No Change	Page Address Value	Zero
Status Register				
SR31–28	1111	1111	/MV, /MF, /MM, /MV	All flags set FALSE
SR27–26	00	00	RESERVED	RESERVED
SR25–24	11	11	Active Address Type	Reset State
SR23–20	0000	0000	RESERVED	RESERVED
SR19–16	1111	1111	Page Address	Zero
SR15–12	0000	0000	RESERVED	RESERVED
SR11–0	111111111111	111111111111	Active Address	All '1's
Device Select Register				
DS31–9	000000H	000000H	RESERVED	RESERVED
DS8	1	1	Enable Select	Enable deselected
DS7–4	0000	0000	RESERVED	RESERVED
DS3–0	1111	No Change	Device Select Value	All '1's

Table 10: Reset Conditions

/MV	/MF	Compare Cycle Result
0	0	Match in CAM Array
0	1	Hit in VP Table
1	0	RESERVED
1	1	No Match

Table 11: Match Flag Indication

The VP Table address can be masked by the value set in Configuration Register bits FR23–12. The mask bits are ANDed with the incoming VP Table Address bits, so a zero in a mask bit forces the corresponding address bit to zero. This facility is useful where, for example, only the VPI information is needed, and the upper-order address bits can be forced to zero. In this case, the VP Table Address mask would contain zeros in the four most significant bit positions, and ones in the rest, and the VP Table would only take 256 locations.

Conclusion

The ATMCAM has been designed to fulfill the needs of the ATM system designer for performing high-speed cell header lookup and translation. It supports both hardware and software control, and VC and VP connections. As ATM systems progress towards 2.5Gbs, the ATMCAM will provide an increasingly necessary solution for hardware acceleration.

MUSIC Semiconductors Agent or Distributor:

MUSIC Semiconductors reserves the right to make changes to its products and specifications at any time in order to improve on performance, manufacturability, or reliability. Information furnished by MUSIC is believed to be accurate, but no responsibility is assumed by MUSIC Semiconductors for the use of said information, nor for any infringement of patents or of other third party rights which may result from said use. No license is granted by implication or otherwise under any patent or patent rights of any MUSIC company.

©Copyright 1998, MUSIC Semiconductors

http://www.music-ic.com
email: info@music-ic.com

USA Headquarters
MUSIC Semiconductors
254 B Mountain Avenue
Hackettstown, New Jersey 07840
USA
Tel: 908/979-1010
Fax: 908/979-1035
USA Only: 800/933-1550 Tech. Support
888/226-6874 Product Info.

Asian Headquarters
MUSIC Semiconductors
Special Export Processing Zone 1
Carmelray Industrial Park
Canlubang, Calamba, Laguna
Philippines
Tel: +63 49 549 1480
Fax: +63 49 549 1023
Sales Tel/Fax: +632 723 62 15

European Headquarters
MUSIC Semiconductors
Torenstraat 28
6471 JX Eygelshoven
Netherlands
Tel: +31 45 5462177
Fax: +31 45 5463663