Order this document by
AN2060/D

AN2060

Application Note
MPC860SAR Microprocessor ATM CAM Interface Application

V1.0 -- Initial release

V1.1 -- August 5, 1998 -- Fixed bugs found during Verilog verification: (1) changed match port logic for $\overline{\text{MS}}$ to non-inverting, (2) corrected error in register programming in ‘Insert Value’, ‘Delete Value’. V1.2 1/02 Reformat.

OVERVIEW

The MPC860SAR supports demultiplexing of a received ATM cell stream into memory buffers for each receive connection. In order to do this, the MPC860SAR must resolve identification information in the ATM cell headers into an internally-used connection number. Header information considered in doing this can include VPI, VCI, GFC, and PTI. The MPC860SAR designer selects exactly which bits to consider by setting a header mask register.

There are three methods for resolving cell header information into connection numbers, including (1) a simple internal lookup table, (2) a two-level external address compression tables, and (3) an external content-addressable memory (CAM). The first method is adequate for up to 32 virtual circuit connections (VPCs or VCCs). The latter two methods can support up to 65536 virtual circuit connections. Of these two, the address compression method is lower-cost, but it (1) is less flexible, because it requires the designer to limit the VPI/VCI supported in order to limit the size of the tables, and (2) carries some performance penalty. The address compression method is acceptable if the designer can control the assignment of VPI/VCI, and can accept the slight performance limitations. However, to support the maximum number of virtual circuit connections with 100% flexibility in assignment of VPI/VCI with no performance penalty, a CAM is required.

This document demonstrates the CAM interface. The CAM selected for this application is the Motorola’s 4Kx64 MCM69C232, which can support up to 4096 virtual circuit connections. If more connections are required, larger Freescale CAMs or multiple CAMs can be used with a similar interface.

The following topics are covered:

1. **Physical interface**,
2. **MPC860SAR register configuration,**
3. **MCM69C232 register configuration,**
4. **Transaction timing diagrams.**

PHYSICAL INTERFACE

The interface consists of two ports, the Control Port and the Match Port. The behavior of these ports is very different, and in actual fact they are independent. However, the MPC860SAR will only access one port at a time, as it has only one external bus interface.

The main components of the interface include:

1. Reset and clocking signals,
2. Control port, mapped to a chip-select controlled by the General Purpose Chip Select Machine (GPCM),
3. Match port, mapped to either: A) a chip-select controlled by one of the MPC860SAR's User Programmable Machines (UPMs), or B) a chip-select controlled by the GPCM (with extra glue logic).

NOTE

> The interface described assumes a 50MHz MPC860SAR operating in full-speed bus mode. The fundamentals of the interface remain the same for any other speed of MPC860SAR and for half-speed bus mode, if memory access wait-states are modified appropriately.

The following diagram shows the recommended interface for reset, clocking, and the Control Port between the MPC860SAR and the MCM69C232 CAM.

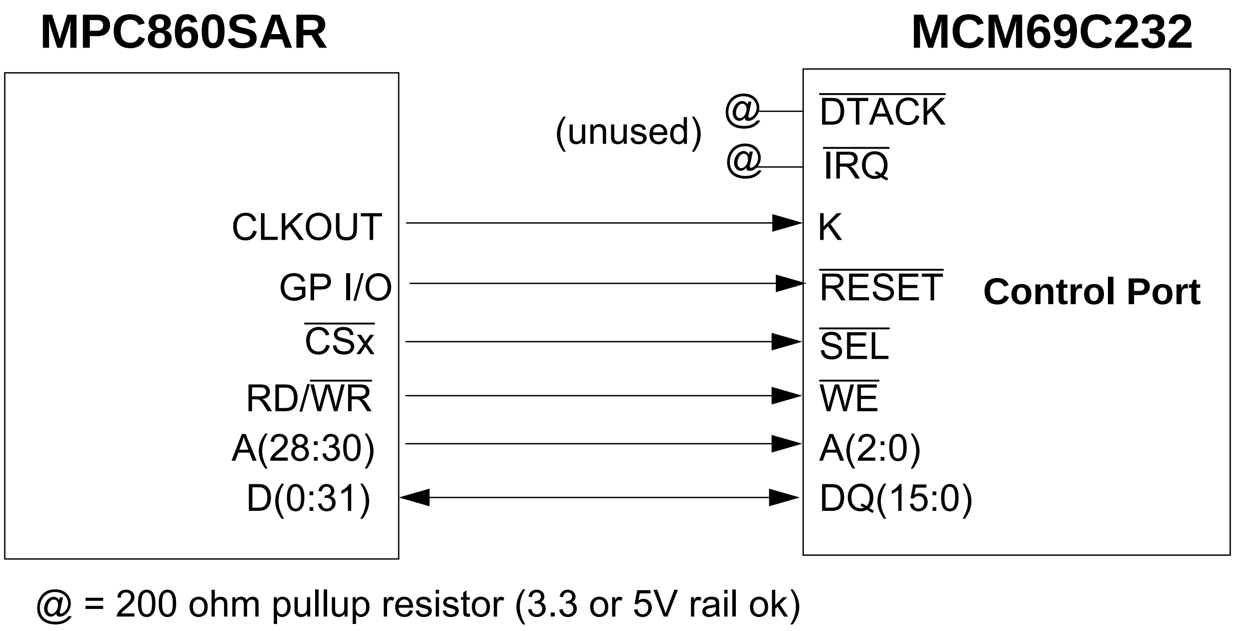

Figure 1. Reset, clocking, and control port interface

Reset and Clocking

The reset input for the CAM is driven by a general-purpose output from the MPC860SAR. This allows the MPC860SAR to reset the CAM in software, for initialization or fault recovery.

The clock input for the CAM is driven by the MPC860SAR's CLKOUT output. At time of writing, the maximum clock frequency of the MCM69C232 CAM is 50MHz.

Control Port

The control port is a 16-bit read/write port. The only control signals required are $\overline{\text{SEL}}$ and $\overline{\text{WE}}$. $\overline{\text{SEL}}$ asserted with $\overline{\text{WE}}$ negated indicates a read; $\overline{\text{SEL}}$ asserted with $\overline{\text{WE}}$ asserted indicates a write. This memory controller signal behavior can be accomplished via the General Purpose Chip Select Machine (GPCM). A chip-select signal ($\overline{\text{CSx}}$) of the MPC860SAR functions as $\overline{\text{SEL}}$, and the RD/$\overline{\text{WR}}$ signal of the MPC860SAR functions as $\overline{\text{WE}}$ of the MCM69C232 CAM.

The control port is only accessible via 16-bit accesses. Therefore, the CAM address lines A(2:0) map to the A(28:30) lines of the MPC860SAR, and A31 of the MPC860SAR is not used. Also, since only 16-bit writes will be performed, $\overline{\text{WEx}}$ signals for specific byte lanes are not required. [The $\overline{\text{WEx}}$ signals of the MPC860SAR cannot be used for the $\overline{\text{WE}}$ signal of the MCM69C232 CAM, because the $\overline{\text{WE}}$ signal must be valid before $\overline{\text{SEL}}$ asserts.]

As a 16-bit memory, the control port maps to data lines D(0:15) of the MPC860SAR's data bus.

The control port will only be accessed by the MPC860SAR during initialization of the CAM and when the MPC860SAR adds or removes connections from the CAM. These tasks are performed by the user's software, not by the MPC860SAR's ATM microcode.

The MCM69C232 includes an $\overline{\text{IRQ}}$ line which can optionally be used to alert the user of certain conditions within the CAM. This function is available to the designer, but it is not absolutely necessary and is therefore not demonstrated in this design. $\overline{\text{IRQ}}$ is pulled up.

The $\overline{\text{DTACK}}$ signal is unused and pulled up. The control port is actually an asynchronous port which responds with $\overline{\text{DTACK}}$ when commands are completed. However, since the match port is not being used simultaneously with the control port, the behavior of this interface is actually very deterministic. Since speed of this control interface is not important to the system, it is possible to design this interface for worst-case timing of this asynchronous port, without monitoring $\overline{\text{DTACK}}$. If 200ns are allowed to elapse between control port accesses, then the user can be assured that each individual command has completed before issuing the next command. For a 50MHz system, this would entail programming the GPCM for the control port's chip-select to 10 wait-states. This holds true for all commands except INITIALIZE TABLE. If the user desires to use the INITIALIZE TABLE command, then the MCM69C232 CAM can be configured via its Interrupt Register to signal an $\overline{\text{IRQ}}$ when this command is complete. This $\overline{\text{IRQ}}$ signal could be used to signal an interrupt to the MPC860SAR upon completion of this command.

Match Port

The following sections describe two different options for the interface between the MPC860SAR and the MCM69C232 CAM.

Option 1: Match port interface using UPM

The following diagram shows the interface between the MPC860SAR and the match port of the MCM69C232 CAM using a chip-select controlled by one of the User Programmable Machines (UPMs) of the MPC860SAR. The advantages of this interface are:

1. Uses less glue logic,
2. Enables optimized timing of the interface by using an external acknowledge signal.

j

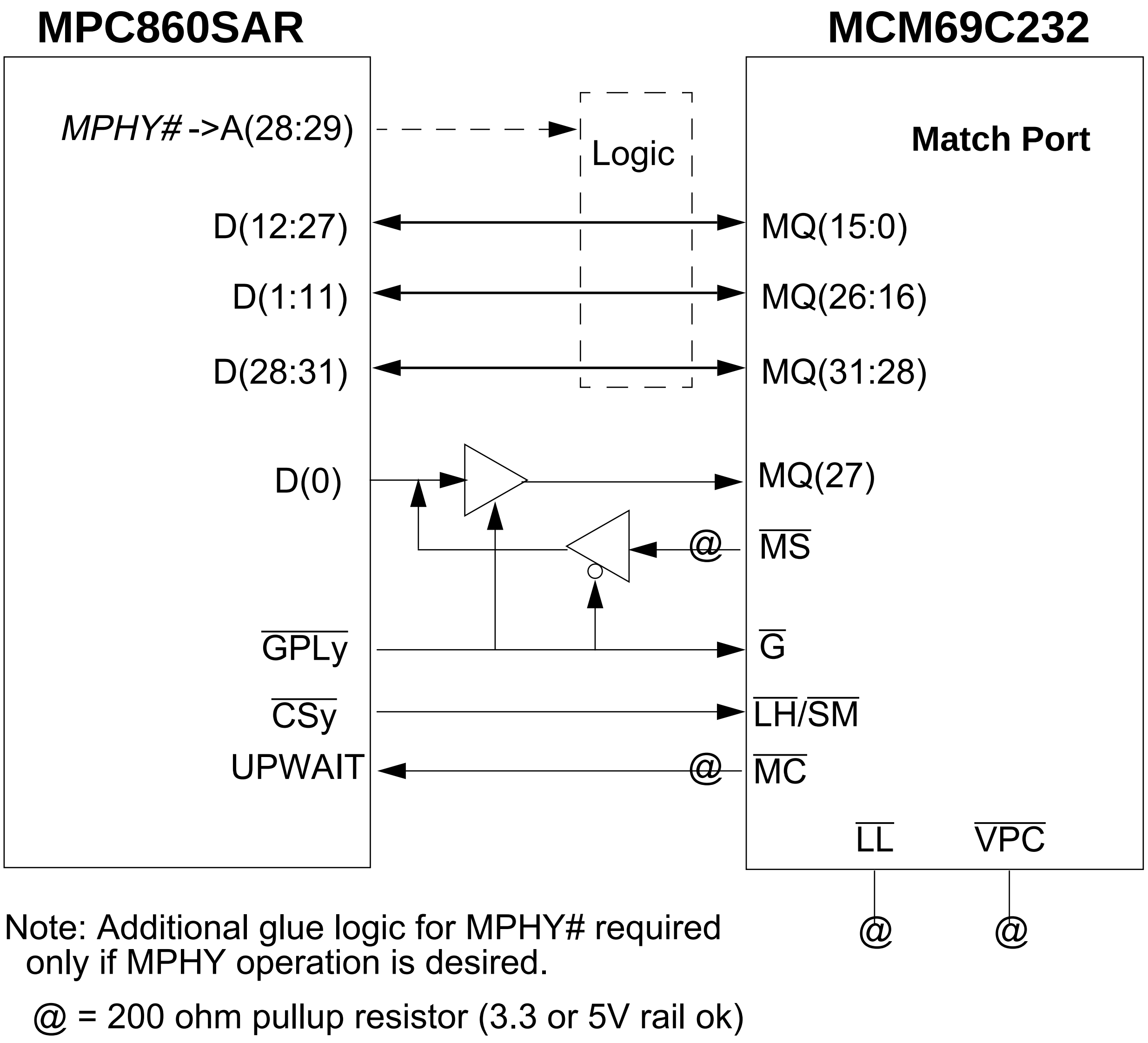

Figure 2. Match port interface using UPM

The match port is a 32-bit read/write port. Accesses to the match port of the MCM69C232 from the MPC860SAR consist of a write cycle followed be a read cycle. The write cycle drives match data to the CAM, which latches the data when the LH/SM signal asserts. The read cycle reads the return data from the CAM (if there is a successful match). Completion of a match attempt is indicated by the MC signal; success of a match is indicated by the MS signal. If a match is successful, return data will be driven by the MCM69C232 CAM as controlled by the output enable signal G. If the control port is not being accessed during a match port access (which will not happen with the MPC860SAR), the MCM69C232 CAM will assert the MC signal in eight clocks. Thus, the maximum number of clocks required for the read cycle is eight; however, it is possible that the read cycle could be even shorter, since other intervening bus activity between the CAM write and read could provide the required eight clocks of latency. Because the match port is in the data path of the receive cell traffic, it is desirable to optimize the timing of its accesses, rather than to set the cycle length to meet worst-case timing of this transaction (as was done with the control port). This behavior is accomplishable via the User Programmable Machine (UPM) of the MPC860SAR's memory controller.

The UPM’s $\overline{\text{CSy}}$ and $\overline{\text{GPLy}}$ signals are used as the $\overline{\text{LH}}/\overline{\text{SM}}$ and $\overline{\text{G}}$ signals, respectively. The UPM is programmed to assert $\overline{\text{CSy}}$ during write cycles and $\overline{\text{GPLy}}$ during read cycles. For read cycles, the $\overline{\text{MC}}$ signal is used as a UPWAIT signal to the UPM. When the $\overline{\text{MC}}$ signal asserts, the UPWAIT signal will be read as negated and the read cycle will be allowed to terminate.

The match port is only accessible via 32-bit accesses. All match data written will be 32-bit operands; all return data read will be 32-bit operands. Thus, the byte-select $\overline{\text{BS}}$(0:3) signals are not used in this interface.

The $\overline{\text{MS}}$ signal is an active-low indication of a successful match, driven concurrently with $\overline{\text{MC}}$. The MPC860SAR expects an active-low indication of a successful match on D0. However, the MPC860SAR will drive match data out on D0 to DQ31 of the MCM69C232 CAM during writes to the match port. Therefore, the tri-state buffers shown are necessary to set the appropriate signal and drive direction of D0. $\overline{\text{GPLy}}$ is used to gate these tri-state buffers.

The arrangement of the data lines as shown is required to enable the ‘ATM MODE’ functionality of the MCM69C232 CAM. This has some implications to the user’s software, which are explained in the discussion of the ‘SET ATM MODE’ command.

The MCM69C232 CAM’s $\overline{\text{LL}}$ signal is only required if match data written to the CAM requires more than one 32-bit write transaction (i.e. is more than 32 bits). The MPC860SAR’s ATM microcode only performs one 32-bit write, so $\overline{\text{LL}}$ is unused and pulled up.

The $\overline{\text{VPC}}$ signal is only used to signal a VPC match when the MCM69C232 CAM is in ‘ATM MODE’. However, for this interface an explicit VPC indication is not necessary; the return data alone is sufficient. Therefore, the $\overline{\text{VPC}}$ signal is also unused and pulled up.

If Multi-PHY operation of the MPC860SAR is desired, then PHY address signals must also be considered as part of the match data. However, all 32 bits of the data bus are already in use by the VPI, VCI, GFC, PTI, and CLP bits. Therefore, if Multi-PHY operation is used, during a match data write cycle the PHY address will appear on A(28:29) of the MPC860SAR. These address lines should be driven on two of the data bits of the match port, as. This will require some gated tri-state buffers, similar to how the $\overline{\text{MS}}$ and D0 interface was implemented. Because this feature will not be required by the majority of applications, these buffers are not explicitly shown, but are indicated as optional ‘Logic’ in the diagram.

Option 2: Match port interface using GPCM

The following diagram shows the interface between the MPC860SAR and the match port of the MCM69C232 CAM using a chip-select controlled by the General Purpose Chip Selects Machine (GPCM) of the MPC860SAR. The advantage of this interface is that it does not require the use of a UPM, if neither of the

UPMs of the MPC860SAR happen to be available. The disadvantage of this interface is that it requires two additional tri-state buffers.

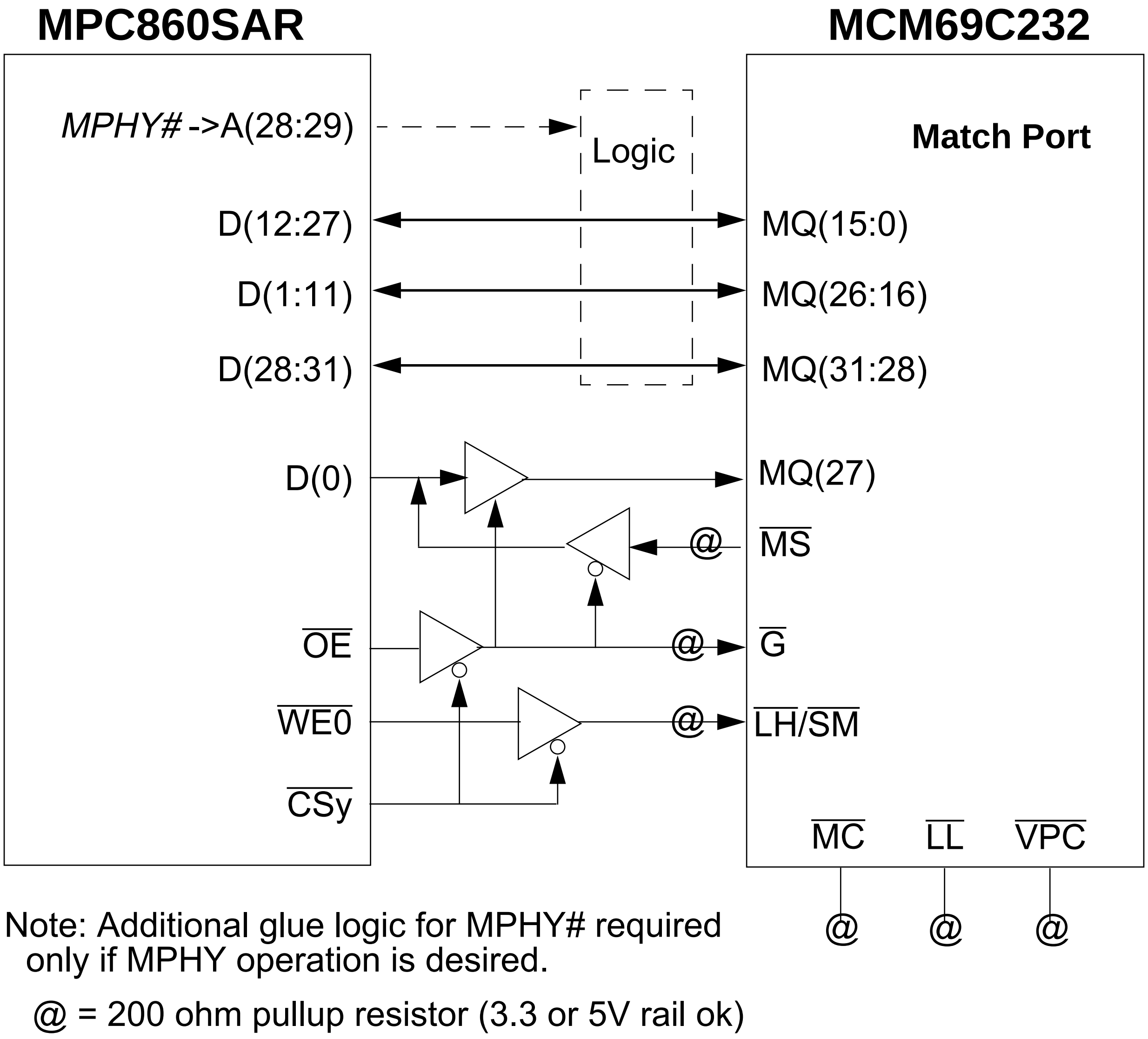

Figure 3. Match port interface using GPCM

The match port is a 32-bit read/write port. Accesses to the match port of the MCM69C232 from the MPC860SAR consist of a write cycle followed be a read cycle. The write cycle drives match data to the CAM, which latches the data when the LH/SM signal asserts. The read cycle reads the return data from the CAM (if there is a successful match). Completion of a match attempt is indicated by the MC signal; success of a match is indicated by the MS signal. If a match is successful, return data will be driven by the MCM69C232 CAM as controlled by the output enable signal G. If the control port is not being accessed during a match port access (which will not happen with the MPC860SAR), the MCM69C232 CAM will assert the MC signal in eight clocks. Thus, the maximum number of clocks required for the read cycle is eight. Thus, it is possible to program a chip-select of the MPC860SAR to accommodate this worst-case timing of the MCM69C232 CAM (as was done with the control port). This behavior is accomplishable via the General Purpose Chip Select Machine (GPCM) of the MPC860SAR's memory controller.

The GPCM’s $\overline{\text{CSy}}$ serves as a gaiting signal for the MPC860SAR’s $\overline{\text{WE0}}$ and $\overline{\text{OE}}$ signals, which serve as the MCM69C232 CAM’s $\overline{\text{LH}}/\overline{\text{SM}}$ and $\overline{\text{G}}$ signals, respectively. The selection of $\overline{\text{WE0}}$ for the gated $\overline{\text{LH}}/\overline{\text{SM}}$ signal was arbitrary; any of the $\overline{\text{WE}}$(0:3) signals could be used.

The match port is only accessible via 32-bit accesses. All match data written will be 32-bit operands; all return data read will be 32-bit operands.

The $\overline{\text{MS}}$ signal is an active-low indication of a successful match, driven concurrently with $\overline{\text{MC}}$. The MPC860SAR expects an active-low indication of a successful match on D0. However, the MPC860SAR will drive match data out on D0 to DQ31 of the MCM69C232 CAM during writes to the match port. Therefore, the tri-state buffers shown are necessary to set the appropriate signal and drive direction of D0. The gated $\overline{\text{OE}}$ signal is used to gate these tri-state buffers.

The arrangement of the data lines as shown is required to enable the ‘ATM MODE’ functionality of the MCM69C232 CAM. This has some implications to the user’s software, which are explained in the discussion of the ‘SET ATM MODE’ command.

The $\overline{\text{MC}}$ signal is only required for dynamic handshaking on this match port interface. However, this interface has been programmed for worst-case timing, instead of using dynamic handshaking. Therefere, $\overline{\text{MC}}$ is not used and is pulled up.

The MCM69C232 CAM’s $\overline{\text{LL}}$ signal is only required if match data written to the CAM requires more than one 32-bit write transaction (i.e. is more than 32 bits). The MPC860SAR’s ATM microcode only performs one 32-bit write, so $\overline{\text{LL}}$ is unused and pulled up.

The $\overline{\text{VPC}}$ signal is only used to signal a VPC match when the MCM69C232 CAM is in ‘ATM MODE’. However, for this interface an explicit VPC indication is not necessary; the return data alone is sufficient. Therefore, the $\overline{\text{VPC}}$ signal is also unused and pulled up.

If Multi-PHY operation of the MPC860SAR is desired, then PHY address signals must also be considered as part of the match data. However, all 32 bits of the data bus are already in use by the VPI, VCI, GFC, PTI, and CLP bits. Therefore, if Multi-PHY operation is used, during a match data write cycle the PHY address will appear on A(28:29) of the MPC860SAR. These address lines should be driven on two of the data bits of the match port, as. This will require some gated tri-state buffers, similar to how the $\overline{\text{MS}}$ and D0 interface was implemented. Because this feature will not be required by the majority of applications, these buffers are not explicitly shown, but are indicated as optional ‘Logic’ in the diagram.

MPC860SAR REGISTER CONFIGURATION

ATM Parameters

The following parameters must be set in the parameter RAM for each ATM channel that will use the CAM:

Receiver State (SRSTATE) -- Sets the operating mode of the receiver. Set EXT=1 (extended channel mode), and set ACP=0 (CAM used for channel lookup).

CAM Address (CAMADD) -- Set this parameter to the address of the match port of the CAM in the MPC860SAR’s memory map. CAMADD must be divisible by 16. The CAM is mapped by the memory controller to a 64KB region.

Header Mask (HMASK) -- This mask determines which bits of the header (GFC, VPI, VCI, PTI, CLP) are used for channel number lookup in the CAM. The user should set these bits in accordance with their desired application.

Memory-Mapping of Control Port

The control port is mapped to a chip-select signal which is controlled by the GPCM of the MPC860SAR’s memory controller. The smallest memory region mappable to a chip-select is 64KB; this minimum memory size should be selected for the chip-select. The assertion of the chip-select signal has been programmed to delay by 1.5 clock cycles (via the TRLX and ACS(0:1) bits of the OR) in order to assure that, during write cycles, data is valid before $\overline{\text{CSx}}$ ($\overline{\text{SEL}}$) asserts to the MCM69C232 CAM.

The programming of the registers for this chip-select should be as follows:

Base Register (BRx)

0:16 BA(0:16) --- Sets the base address for the CAM’s control port. As only address lines A(28:30) are used by the CAM’s control port, this will map the16-byte register set of the MCM69C232 CAM throughout the 64KB region. Set these bits to point to an unused 64KB region of the memory map.

17:19 AT(0:2) -- Can optionally be used to limit access to certain address types. This feature is normally unused, and these bits can be set to zero.

20:21 PS(0:1) -- The control port is a 16-bit port. These bits must be set to binary 10.

22 PARE -- Used to enable parity generation/checking for this chip-select. Set this bit to zero.

23 WP -- Used to write-protect the memory mapped to this chip-select. Set this bit to zero.

24:25 MS(0:1) -- Selects the mechanism used for controlling this chip-select. In this case, it will be the GPCM. Set these bits to binary 00.

26:30 Reserved

31 V -- Activates this chip-select. Set this bit to one.

Option Register (ORx)

0:16 AM(0:16) --- Sets the size of the memory by selecting which bits will be used in address-comparison. Set the memory size of the chip-select to the 64KB minimum by setting these bits to all ones.

17:19 ATM0:2) -- Mask used with the AT(0:2) bits of the BRx to enable address-type protection for this memory. This feature is normally unused. Set these bits to zero to disable this feature.

20 CSNT/SAM -- In this case, this bit sets the chip-select negate timing for this chip-select. Set this bit to zero.

21:22 ACS(0:1)/G5LA,G5LS -- In this case, these bits set the chip-select assertion timing for this chip-select. Set these bits to binary 11.

23 BI -- Enables/disables bursting to this chip-select. The control port is not burstable, so set this bit to one.

24:27 SCY(0:3) -- If the GPCM is used for this chip-select, these bits set the number of wait-states for this chip-select. The wait-states should be set to provide a minimum 200ns cycle time when accessing the control port. For a 50MHz bus, the cycle length should be 10 clocks. Setting TRLX=1

below causes the wait-states programmed by SCY[0:3] to double. Therefore, to set a ten-clock cycle, (SCY[0:3]x2)+2 equals ten. Therefore, set these bits to binary 0100.
28 SETA -- Selects whether $\overline{TA}$ is generated internally or externally. Select internal by setting this bit to zero.
29 TRLX -- Sets relaxed timing for this chip-select. Set this bit to one .
30 EHTR -- Inserts a null cycle for memories with long disable times. Set this bit to zero.
31 Reserved

Memory-Mapping of Match Port

The match port is mapped to a chip-select which is controlled by the MPC860SAR's memory controller. The memory controller can control this chip-select with either the UPM or the GPCM, as previously shown. Register programming for both options is presented here.

Option 1: Match port interface using UPM

This option uses one of the User Programmable Machines (UPM) of the MPC860SAR's memory controller to control the match port. This can be either UPMA or UPMB; in the register descriptions below, y is used as a place-holder. The smallest memory region mappable to a chip-select is 64KB; this minimum memory size should be selected for the chip-select. The programming of the registers for this chip-select should be as follows:

Base Register (BRy)

0:16 BA(0:16) --- Sets the base address for the CAM's match port. As no address lines are used by the CAM's match port, this will map the match port of the MCM69C232 CAM throughout the 64KB region. Set these bits to point to an unused 64KB region of the memory map. This base address must match the value set in CAMADD in the MPC860SAR's ATM parameters.
17:19 AT(0:2) -- Can optionally be used to limit access to certain address types. This feature is normally unused, and these bits can be set to zero.
20:21 PS(0:1) -- The match port is a 32-bit port. These bits must be set to binary 00.
22 PARE -- Used to enable parity generation/checking for this chip-select. Set this bit to zero.
23 WP -- Used to write-protect the memory mapped to this chip-select. Set this bit to zero.
24:25 MS(0:1) -- Selects the mechanism used for controlling this chip-select. In this case, it will be the UPMA or UPMB. Set these bits to binary 10 or binary 11.
26:30 Reserved
31 V -- Activates this chip-select. Set this bit to one.

Option Register (ORy)

0:16 AM(0:16) --- Sets the size of the memory by selecting which bits will be used in address-comparison. Set the memory size of the chip-select to the 64KB minimum by setting these bits to all ones.

17:19 ATM0:2) -- Mask used with the AT(0:2) bits of the BRx to enable address-type protection for this memory. This feature is normally unused. Set these bits to zero to disable this feature.

20 CSNT/SAM -- In this case, this sets the initial state of the address-multiplexing for this chip-select. Set this bit to zero.

21:22 ACS(0:1)/G5LA,G5LS -- In this case, these bits set the behavior of the GPL5 line for this UPM. GPL5 is not used in this interface. Set these bits to binary 01.

23 BI -- Enables/disables bursting to this chip-select. The match port is not burstable, so set this bit to one.

24:27 SCY(0:3) -- If the GPCM is used for this chip-select, these bits set the number of wait-states for this chip-select. The GPCM is not used for this chip-select, so these bits can be set to zero.

28 SETA -- Selects whether $\overline{\text{TA}}$ is generated internally or externally. Select internal by setting this bit to zero.

29 TRLX -- Sets relaxed timing for this chip-select. This is a don't-care for the UPM. Set this bit to zero.

30 EHTR -- Inserts a null cycle for memories with long disable times. This is a don't-care for the UPM. Set this bit to zero.

31 Reserved

Machine (A or B) Mode Register (MyMR)

0:7 PTA(0:7) -- Sets the period of the refresh timer for this UPM. The CAM doesn't need refresh, so this is a don't-care. Set these bits to zero.

8 PTAE -- Enables/disables the refresh timer for this UPM. Set this bit to zero.

9:11 AMy(0:2) --Sets the address multiplexing size for this UPM. Address muliplexing is not used for the CAM. Set these bits to binary 000.

12 Reserved

13:14 DSA(0:1) -- Sets the period of the disable timer between accesses to this UPM. No delay is needed, so set these bits to binary 00.

15 Reserved

16:18 G0CLA -- Determines which address line is put out on $\overline{\text{GPL0}}$, if this feature is used. This features is not used, so set these bits to binary 000.

19 GPL_y4DIS -- Enables/disables $\overline{\text{GPL\_y4}}$, allowing this signal to be used as UPWAIT. UPWAIT is required for this interface. Set this bit to one.

20:23 RLFy(0:3) --Sets the number of iterationsof loops in the UPM's read patterns. Loops are not used by the UPM patterns, so these bits can be set to binary 0000.

24:27 WLFy(0:3) --Sets the number of iterationsof loops in the UPM's write patterns. Loops are not used by the UPM patterns, so these bits can be set to binary 0000.

28:31 TLFy(0:3) --Sets the number of iterationsof loops in the UPM's refresh patterns. Refresh is not performed for this UPM, so these bits can be set to binary 0000.

UPM RAM words

The following patterns should be programmed in to the UPM RAM words for the appropriate UPM (A or B). Only the single-beat read and single-beat write patterns are used. The burst patterns, refresh pattern, and exception patterns do not need to be initialized.

For a 50MHz bus, the UPM RAM words for the UPM RAM words should be programmed to:

Single-beat read:

```
0xFFFFCC04
0xFFF3DC04
0XFFF3CC00
0XFFF7CC05
```

Single-beat write:

```
0xFFFFCC04
0x0FFFCC04
0X3FFFCC00
0xFFFFCC05
```

Option 2: Match port interface using GPCM

This option uses the General Purpose Chip Select Machine (GPCM) of the MPC860SAR's memory controller to control the match port. The smallest memory region mappable to a chip-select is 64KB; this minimum memory size should be selected for the chip-select. The programming of the registers for this chip-select should be as follows:

Base Register (BRy)

0:16 BA(0:16) --- Sets the base address for the CAM's match port. As no address lines are used by the CAM's match port, this will map the match port of the MCM69C232 CAM throughout the 64KB region. Set these bits to point to an unused 64KB region of the memory map. This base address must match the value set in CAMADD in the MPC860SAR's ATM parameters.

17:19 AT(0:2) -- Can optionally be used to limit access to certain address types. This feature is normally unused, and these bits can be set to zero.

20:21 PS(0:1) -- The match port is a 32-bit port. These bits must be set to binary 00.

22 PARE -- Used to enable parity generation/checking for this chip-select. Set this bit to zero.

23 WP -- Used to write-protect the memory mapped to this chip-select. Set this bit to zero.

24:25 MS(0:1) -- Selects the mechanism used for controlling this chip-select. In this case, it will be the GPCM. Set these bits to binary 00.

26:30 Reserved

31 V -- Activates this chip-select. Set this bit to one.

Option Register (ORy)

0:16 AM(0:16) --- Sets the size of the memory by selecting which bits will be used in address-comparison. Set the memory size of the chip-select to the 64KB minimum by setting these bits to all ones.

17:19 ATM0:2) -- Mask used with the AT(0:2) bits of the BRx to enable address-type protection for this memory. This feature is normally unused. Set these bits to zero to disable this feature.

20 CSNT/SAM -- In this case, this sets the timing of the chip-select negation. Set this bit to one (thus programming the chip-select to negate one clock early).

21:22 ACS(0:1)/G5LA,G5LS -- In this case, these bits set the timing of the chip-select assertion. Set these bits to binary 10 (thus programming the chip-select to assert 1/4 clock late).

23 BI -- Enables/disables bursting to this chip-select. The match port is not burstable, so set this bit to one.

24:27 SCY(0:3) -- Since the GPCM is used for this chip-select, these bits set the number of wait-states for this chip-select. Cycle length should be eight clocks. Setting TRLX=1 below causes the wait-states programmed by SCY[0:3] to double. Therefore, to set an eight-clock cycle, (SCY[0:3]x2)+2 equals eight. Therefore, set these bits to binary 0011.

28 SETA -- Selects whether $\overline{TA}$ is generated internally or externally. Select internal by setting this bit to zero.

29 TRLX -- Sets relaxed timing for this chip-select. Set this bit to one (thus providing an extra clock before chip-select assertion).

30 EHTR -- Inserts a null cycle for memories with long disable times. This isnot necessary for the MCM69C232 CAM. Set this bit to zero.

31 Reserved

MCM69C232 CAM REGISTER CONFIGURATION

Operation Register

Refer to MCM69C232 documentation for the complete command set.

'SET GLOBAL MASK REGISTER' COMMAND

[WARNING: The sense of this mask register is opposite of normal mask registers;1 indicates ignore this bit, 0 indicates use this bit].

Set the lower 32 bits to 1; these lower 32 bits of the CAM will contain the channel number, and will be returned upon a successful match. Set the upper 32 bits to $00000000 (i.e. do not mask the match data presented to the CAM). Masking of unwanted bits will be done by the MPC860SAR's microcode via the MPC860SAR's HMASK register, so it is not necessary to do it with the Global Mask Register..

'SET ATM MODE' COMMAND

The match port interface has been designed to enable 'ATM MODE' of the MCM69C232. In 'ATM MODE', the MCM69C232 CAM can perform matching on either a VCC or VPC basis.

In order to enable matching via VPC in 'ATM MODE', the order of the data bits of the MPC860SAR and the MCM69C232 CAM had to be arranged in a particular fashion. This is because the MCM69C232 expects the VCI to be on its data lines MQ(15:0). The MPC860SAR's match data places the VCI on data bits 12-27, which would correspond to data lines MQ(19:4) if the data bus was connected normally.

The data lines are effectively shifted in the MPC860SAR to MCM69C232 CAM match port interface, such that:

1. D(12:27) connect to MQ(15:0)
2. D(1:11) connect to MQ(26:16)
3. D0 connects to MQ27 (and to $\overline{MS}$, as shown in the diagram)
4. D(28:31) connect to MQ(31:28)

There are software implications of this data-bit arrangement which pertain to the 'INSERT VALUE', 'DELETE VALUE', and related commands.

'INSERT VALUE', 'DELETE VALUE', AND RELATED COMMANDS

The arrangement of the data bits in this interface requires the user to use the I/O registers in the following manner when issuing 'INSERT VALUE', 'DELETE VALUE', and related commands:

I/O Register 3: Bits15-13=PTI, Bit 12=CLP, Bits 11-0=GFC/VPI

I/O Register 2: Bits 15-0=VCI

I/O Register 1: Upper order bits of channel number (match return data), shifted right by 4

I/O Register 0: Lower order bits of channel number (match return data), shifted left by 12

For example, in order to route NNI cells with GFC/VPI = hex 5F2, VCI = hex 9A26, PTI = binary 010, CLP = 0 to channel number hex 3752, write I/O Reg 3 = hex 45F2, I/O Reg 2 = hex 9A26, I/O Reg 1 = hex 2000, I/O Reg 0 = hex 0375.

Because of the arrangement of the data lines, the match data (i.e. 3752) will be returned to the MPC860SAR via the match port on data lines D(16:31).

‘SET FAST ENTRY MODE’ COMMAND VERSUS ‘SET BUFFERED ENTRY MODE’ COMMAND

Buffered Entry mode is recommended for normal operation. Fast Entry mode could be used at initialization, but using Fast Entry mode requires the use of an INITIALIZE TABLE command, issues with which are described below.

‘INITIALIZE TABLE’ COMMAND

The physical interface to the control port of the MCM69C232 CAM uses normal worst-case timing of 200ns per command transaction in order to avoid monitoring the $\overline{\text{DTACK}}$ signal. However, the 200ns worst-case timing does not apply to the INITIALIZE TABLE command; it could be longer. If the user desires to use the INITIALIZE TABLE command, then the MCM69C232 CAM can be configured via its Interrupt Register to signal an $\overline{\text{IRQ}}$ when this command is complete. This $\overline{\text{IRQ}}$ signal could be used to signal an interrupt to the MPC860SAR upon completion of this command.

Error Code Register

Check for error conditions in Error Code Register after control operations.

Interrupt Register

Though not shown in the interface, interrupts (signalled via $\overline{\text{IRQ}}$) can be used to signal the MPC860SAR of certain conditions. This can be configured in the Interrupt Register.

TRANSACTION TIMING DIAGRAMS

Control Port Timing

The access to the control port is performed by the user software, for initialization of the CAM and for adding and deleting entries in the CAM.

For a 50MHz bus, the timing for the read and write accesses is as shown below:

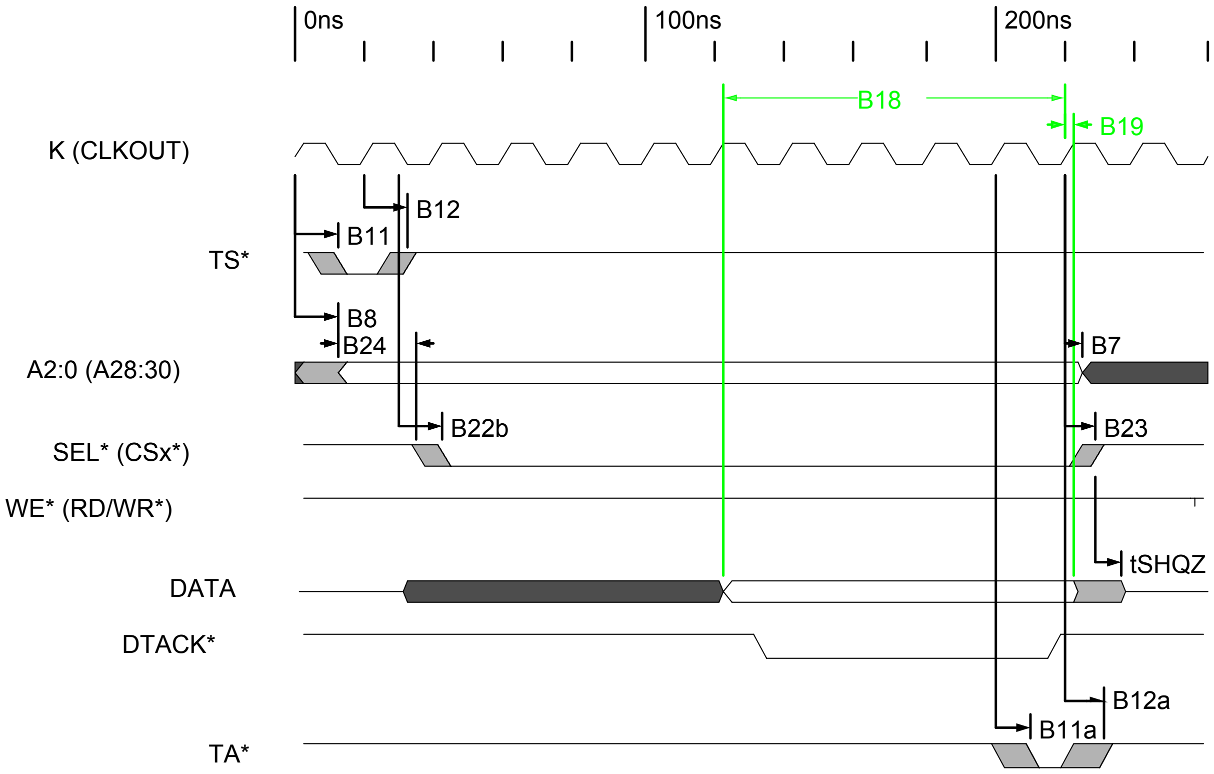

Figure 4. MCM69C232 CAM Control Port Read Access

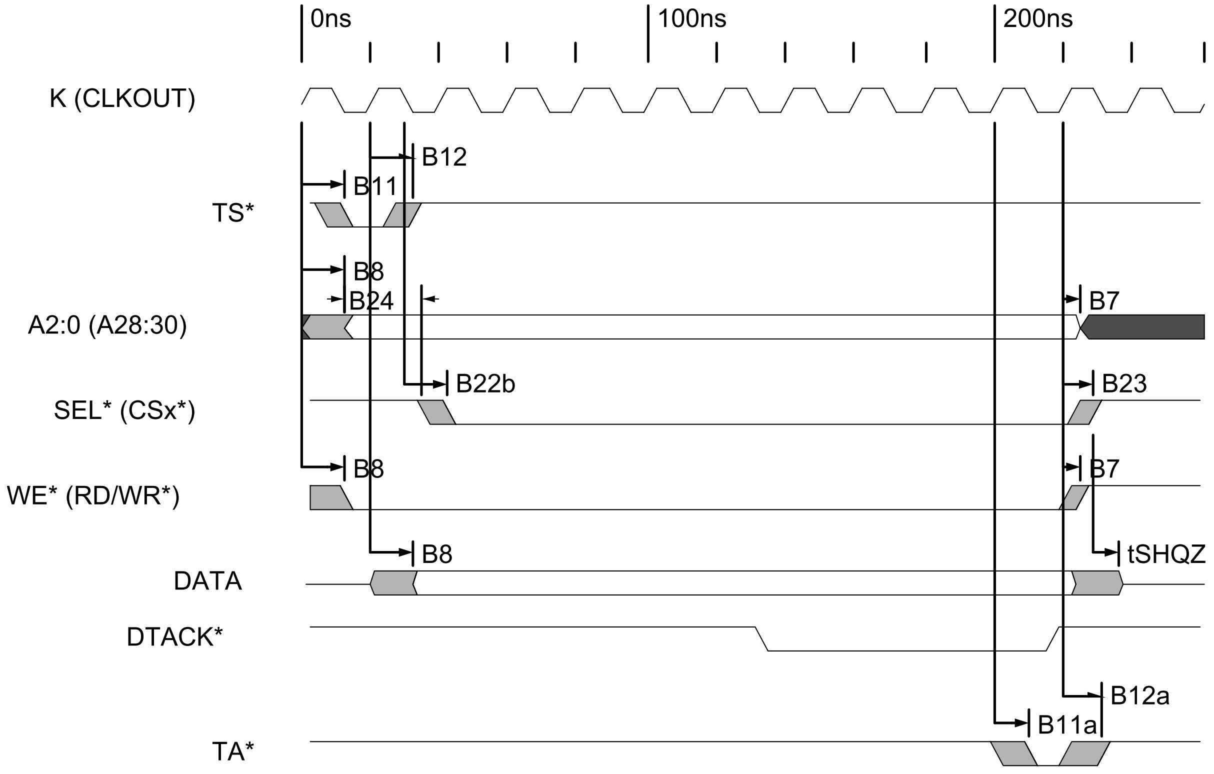

Figure 5. MCM69C232 CAM Control Port Write Access

Match Port Timing

The access to the match port is performed by the MPC860SAR's ATM microcode. The microcode performs a DMA write (of match data) followed by a DMA read (of return data from the CAM). These DMA accesses are not necessarily back-to-back; other higher-priority bus masters can take the bus between these DMA accesses.

Option 1: Match port interface using UPM

For a 50MHz bus, the timing for the write and read accesses is as shown below:

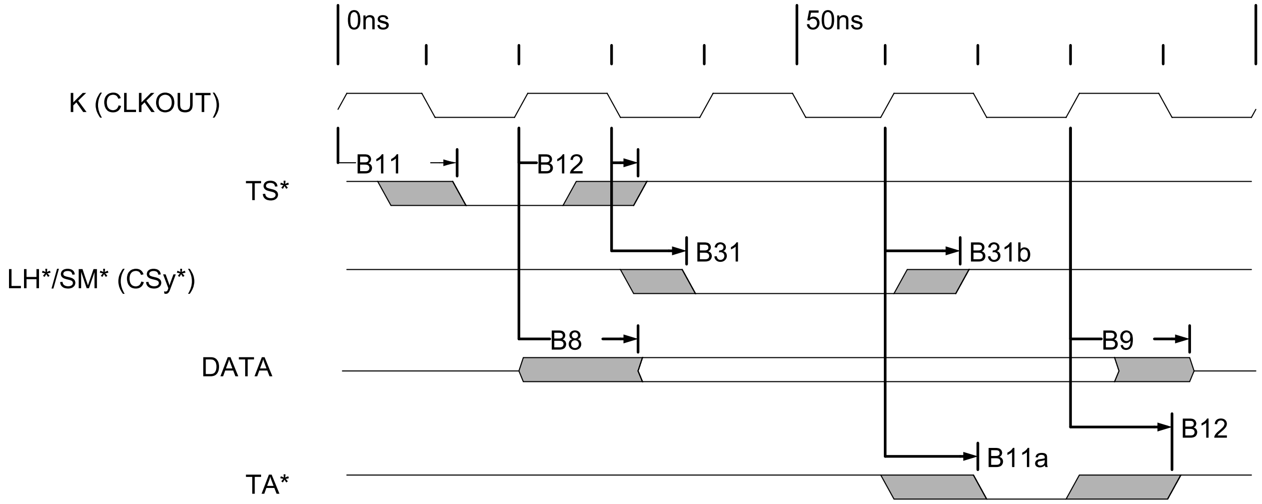

Figure 6. MCM69C232 CAM Match Port Write Access using UPM (writing match data to CAM

)

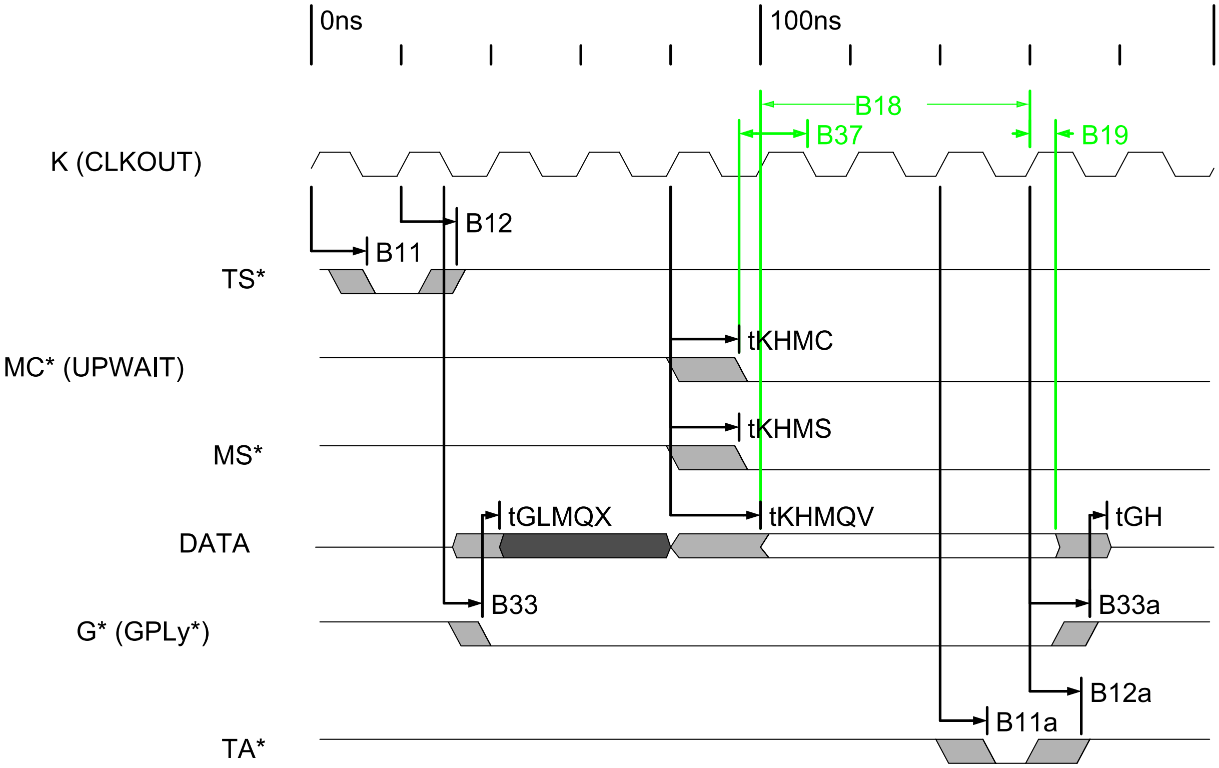

Figure 7. MCM69C232 CAM Match Port Read Access using UPM (reading return data from CAM)

Option 2: Match port interface using GPCM

For a 50MHz bus, the timing for the write and read accesses is as shown below. The timing diagrams include a propagation delay of 3 to 8 ns through the external logic, shown as tPD..

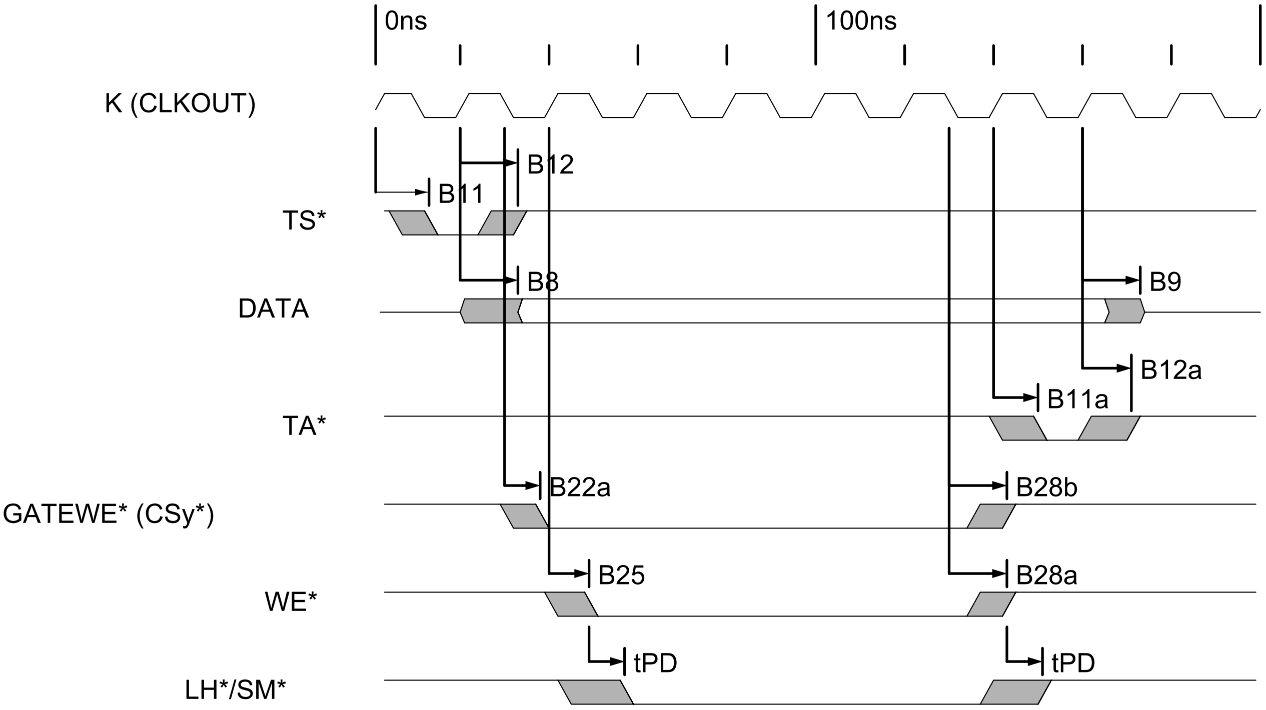

Figure 8. MCM69C232 CAM Match Port Write Access using GPCM (writing match data to CAM)

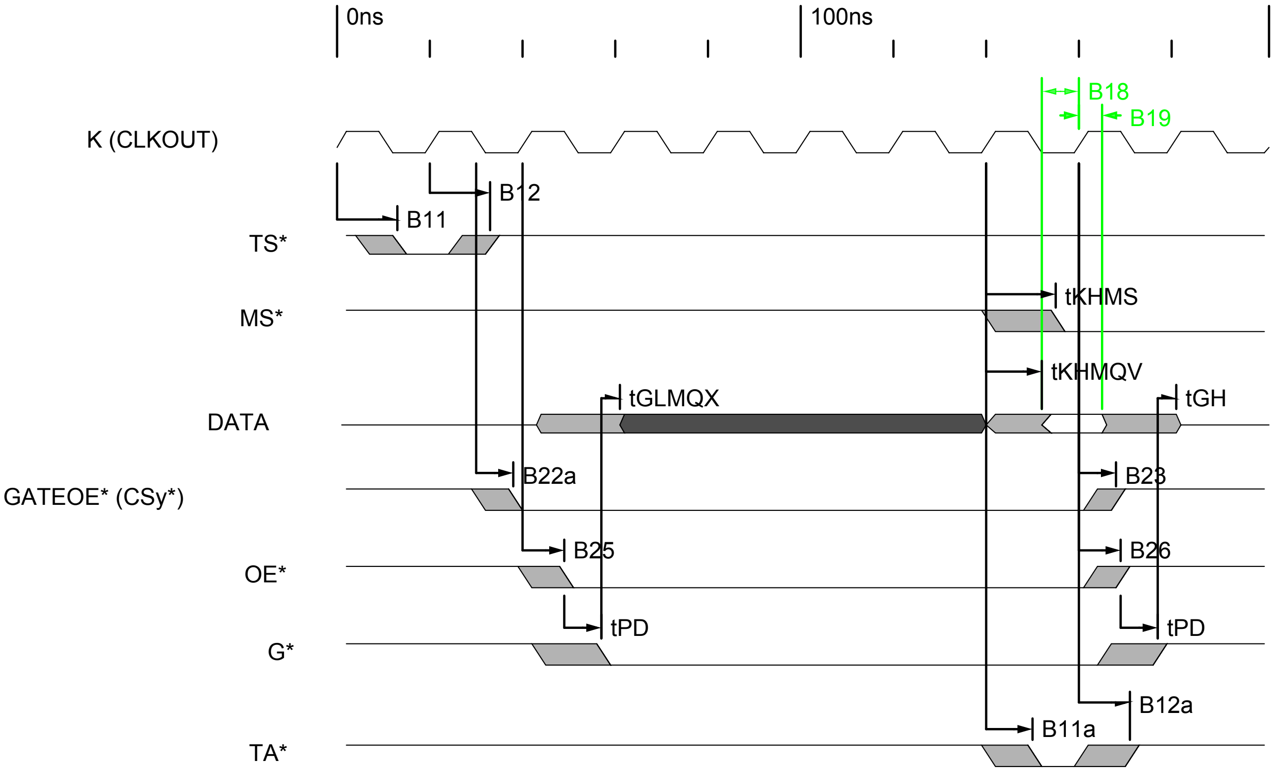

Figure 9. MCM69C232 CAM Match Port Read Access using GPCM (reading return data from CAM)

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