SPARClite—The ASIC Companion
Application Note 8

Fujitsu Microelectronics, Inc.

Table of Contents

SPARClite—The ASIC Companion

Introduction

Within the embedded market, each new generation of products demands ever-increasing levels of performance and functionality at an ever-decreasing cost. The need for hardware acceleration, product customization, and high levels of integration have caused discrete components and other programmable devices to give way to custom ASICs, and microcontrollers to give way to high-performance 32-bit microprocessors.

The SPARClite MB8683x family of embedded processors is designed to function as a companion for the ASIC, providing the ability to differentiate a given product line without having to change the ASIC each time. For example, the MB8683x family offers products that scale in both the amount of on-chip cache and operating frequency. In addition, the MB86832 and MB86831 are pin compatible, eliminating many of the redesign issues often encountered when upgrading from one processor to another.

High-performance processors such as SPARClite are now often embedded within the product itself. This reduces the reliance on, and in some cases eliminates the need for, a host computer. One example would be digital cameras that are able to download pictures directly to a printer without the need for host intervention.

The SPARClite MB8683x family offers high levels of performance and integration, low-cost, and binary compatibility across the product line. All of this in an open architecture with a large installed base.

The potential applications for high-performance, high-integration, low cost CPU/ASIC-based solutions are endless. Some examples include:

- Digital still and video cameras
- Set-top boxes
- Hubs for Small Office Home Office (SOHO)
- Finger print recognition
- Video conferencing
- Printers, scanners, and fax machines
- Digital Versatile Disc (DVD) players
- Cellular and cordless telephones
- Smart smoke detectors
- Home control of lights and temperature
- Home and portable audio systems
- Metering devices
- Home appliances (washers, microwave ovens, answering machines, etc.)

This application note discusses the SPARClite MB8683x family of 32-bit microprocessors and how these devices excel in ASIC-based environments. Many of the SPARClite features are geared toward augmenting ASIC-based applications where both performance and cost are an issue. These features are discussed throughout this document using a typical digital camera as a design example.

Block Diagram Example

Figure 1 shows a block diagram of a typical digital camera application using SPARClite. The SPARClite processor works in conjunction with a custom ASIC to perform all necessary functions. In this diagram the ASIC interfaces to the CCD array, converting the CCD array data and performing pixel interpolation. The processor reads the converted information, manipulates the data as directed by software, then maps the data to the LCD display.

In Figure 1, the ASIC handles functions such as CCD interface and pixel processing, LCD display control, JPEG compression and decompression, serial port interface for downloading pictures to a peripheral device, compact flash interface for on-camera temporary image storage, and a PCMCIA interface for off-camera non-volatile image storage.

The features of the SPARClite MB8683x allow for easy interfacing to each of the modules in Figure 1. An on-chip DRAM controller offers a glueless interface to the memory array. Programmable chip-select logic and an on-chip address decoder allows for easy interfacing to peripheral devices such as EEPROM, DRAM, UART, and ASIC. A robust interrupt handler offers per-interrupt programmability and supports up to 15 external interrupts. A three-pin bus arbitration protocol is included which allows other bus masters such as an ASIC to assume control of the bus. At the same time they use the resources of the MB8683x, such as the DRAM controller and programmable chip selects.

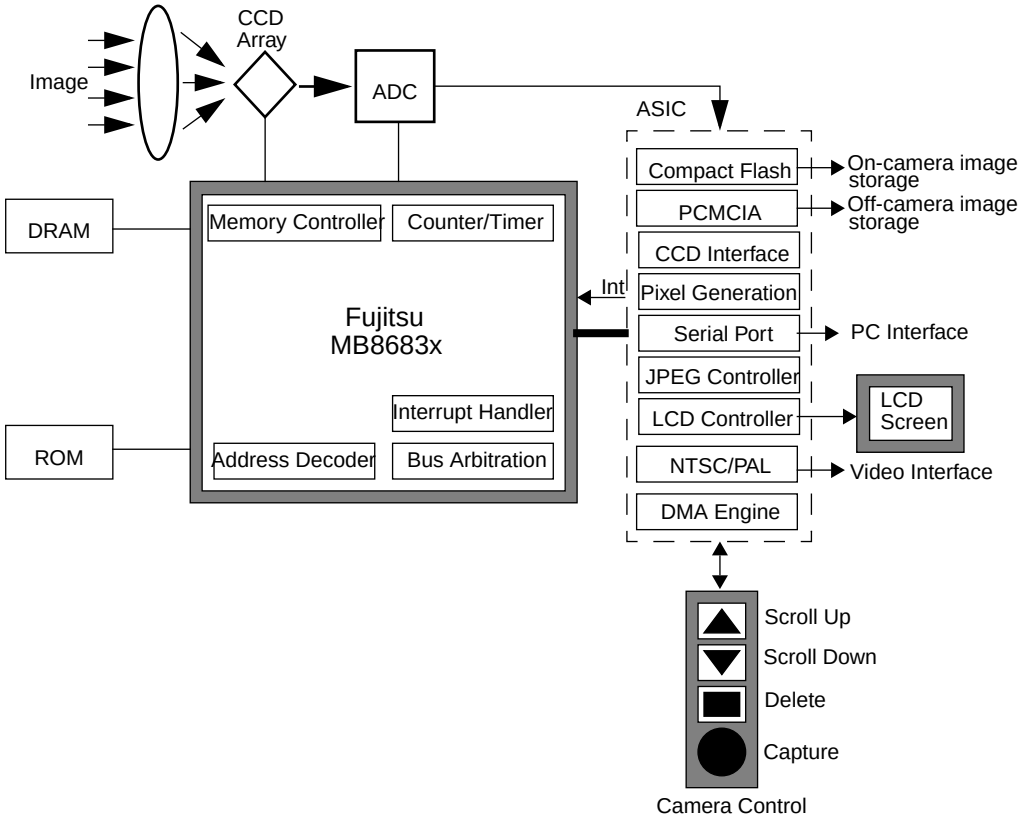

Figure 1. Typical Image Processing Application

SPARClite Family Overview

The SPARClite MB8683x family is a series of RISC microprocessors that offers high performance and high integration for a wide range of embedded applications. Each member of the SPARClite MB8683x family offers the following general features:

- SPARC V8E code compatible.
- Hardware multiplier for faster multiplication.
- 136 working registers controlled via register-windowing for faster task switching.
- 2-way set associative instruction and data caches. Both can be locked on a per-line basis for storage of time-critical code and data.
- Flexible on-chip DRAM controller supports multiple bus widths and DRAM types, eliminating the need for external logic.
- On-chip interrupt controller allows for the programmability of both level and priority for each interrupt.
- Power-down management circuitry provides for reduced power consumption in power-sensitive applications.
- Flexible addressing scheme supports sixteen 256-MByte address spaces. Address Space Identifiers (ASI) distinguish between protected and unprotected space.
- SCAN instruction for bit manipulation.

All SPARClite instructions are 32-bits and are aligned on 32-bit boundaries in memory. There are only three basic instruction formats, all featuring uniform placement of opcode and register address fields. Only load and store instructions access memory or I/O.

Figure 2 shows a block diagram of the MB8683x family. The cache sizes differ between products.

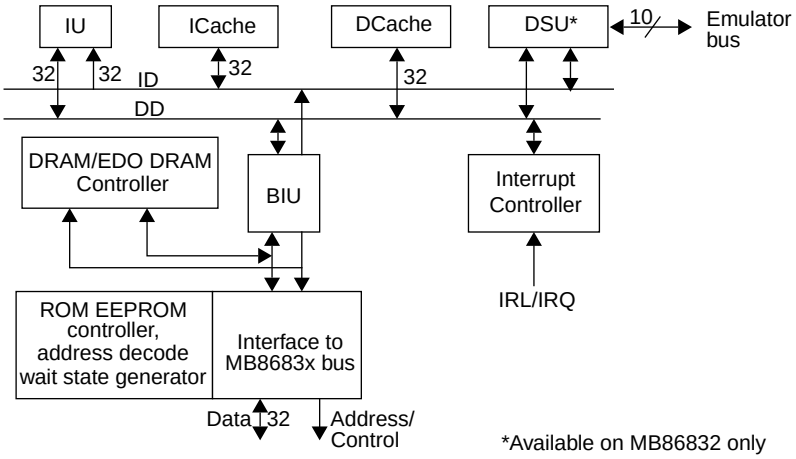

Figure 2. MB8683x Block Diagram

SPARClite Family Product Offerings

MB86833

The MB86833 is the lowest-cost 32-bit RISC microprocessor on the market today, with a price-performance point approaching that of 16-bit microcontrollers. The MB86833 operates at 66 MHz and contains a 1 kbyte instruction cache and 1 kbyte data cache. The contents of each cache can be locked on a per-line basis. Power management logic allows for extended operation in battery-operated environments. The on-chip DRAM controller helps reduce cost by providing a fully functional DRAM interface that does not require any external logic to operate.

MB86832

The MB86832 contains the largest caches in the family with 8 kbytes of data and 8 kbytes instruction, and operates at speeds of 66, 80, and 100 MHz. The contents of each cache can be locked on a per-line basis. The MB86832 is pin-compatible with the MB86831, providing an upgrade path that does not require any redesign to the board. The MB86832 incorporates an on-chip debug support unit (DSU) for connection to an in-circuit emulator.

MB86831

The MB86831 is the midrange product in the MB8683x family, offering higher operating frequencies and larger caches than the MB86833, but smaller caches than the MB86832. The MB86831 contains 4 kbytes of instruction cache and 2 kbytes of data cache and operates at speeds of 66, 80, and 100 MHz. The contents of each cache can be locked on a per-line basis.

Table 1 gives an overview of features for each member of the MB8683x family.

Table 1. SPARClite Product Family Overview

Parameter	MB86831	MB86832	MB86833
Clock Frequency	66/80 MHz	66/80/100 MHz	66 MHz
Instruction Cache Size	4 kbytes	8 kbytes	1 kbyte
Data Cache Size	2 kbytes	8 kbytes	1 kbyte
Data Bus Width	32-bit	32-bit	32-bit
Max. Address Bus Width	28-bit	28/32-bit	24/28-bit
Memory Controller	EDO/FPM DRAM	EDO/FPM DRAM	EDO/FPM DRAM
Interrupt Controller	8 channel	8 channel	8 channel
Hardware Multiplier	Yes	Yes	Yes
Packaging	176 SQFP	176 SQFP	144 LQFP

The following sections describe some of the features of the MB8683x and give examples of how these features can be used in the ASIC-based digital camera design shown in Figure 1.

Flexible Memory Interface

For all designs that use DRAM, the overall price-performance point of a product is often affected by the volatility of the DRAM market. Devices that are relatively inexpensive one month can be prohibitively expensive the next.

To achieve different price-performance points for a given product, a single product family may require different speeds, sizes, or even types of DRAM. In order to use the same ASIC in each of these

product revisions, the ASIC would be required to support all of these DRAM options, requiring a complex DRAM controller that might require large amounts of on-chip real estate.

The on-chip DRAM controllers in the MB8683x family provide high levels of flexibility in the memory design and eliminate the costs associated with an external ASIC-based DRAM controller.

This level of flexibility is very useful during the design phase of a product, when it is not known which type, speed, or width of DRAM will provide the most cost-effective solution. The SPARClite memory controller provides the following features that are discussed in the following sections:

- Programmable Bus Width (8/16/32 bit)
- Fast Page Mode (FPM) support
- Extended Data Out (EDO) support
- Programmable chip selects
- Non-cachable memory space support
- Burst mode
- EPROM/PROM Interface

Programmable Bus Width

The programmable bus width feature allows the memory data bus to be configured as 8-, 16-, or 32-bits wide. Normally, 8- or 16-bit widths are chosen to allow the use of low cost memory devices. For example, the low-end offering for a product can use slower and lower-cost 8-bit devices, and the high-end product can use fast 32-bit devices.

The programmable bus width feature also allows for multiple memory bus widths in the same system. This is important in designs that incorporate both 8-bit devices such as EPROMs or UARTs, and 16- or 32-bit DRAM, SRAM, Ethernet controllers, or other peripheral devices.

In order to minimize the amount of external logic required for interfacing to 8- and 16-bit ROM devices, the byte enable bits (BE3:0) are encoded on ROM accesses to reflect the two least-significant-bits (LSB) of a byte address or the LSB of a 16-bit address. Therefore, address bits A[27:2] can be concatenated with the BE3:0 bits to form a complete ROM address.

The number of wait states is programmable for each chip select. For example, the memory array may require only one wait state. However, a slow 8-bit ROM may continue to drive data for two or three system clocks after the address has been removed. In this case the processor can add wait states to compensate for the delay. The MB8683x provides a special pin (IDLEEN) to support this function.

DRAM Control Signals

The SPARClite MB8683x DRAM controller contains three main groups of control signals: RAS, CAS, and DWE. The SPARClite is a *Big Endian* architecture, so the behavior of the DRAM control signals and byte enables is different from a *Little Endian* architecture. Refer to the *Endianness* section for a definition of *Big Endian*.

Each member of the MB8683x family contains four CAS lines. In a 32-bit memory system, CAS[0] controls byte 0, which is the most-significant byte in a Big Endian architecture, and CAS[3] controls byte 3, or the least-significant byte.

In a 16-bit memory system, only one half of the data bus is connected to the memory array. CAS[2] controls the most-significant byte (byte 2), and CAS[3] controls the least-significant byte (byte 3). CAS[0] and CAS[1] are not used in a 16-bit memory system.

Table 2 shows how the CAS lines are assigned for 32-bit and 16-bit memory systems.

Table 2. Configuring CAS Lines

Memory Bus Width	CAS[x] Line	Byte Order	Data Bus Pins
32	0	0 (MSB)	D[31:24]
	1	1	D[23:16]
	2	2	D[15:8]
	3	3 (LSB)	D[7:0]
16	0	not used	
	1	not used	
	2	2 (MSB)	D[15:8]
	3	3 (LSB)	D[7:0]

Depending on the architecture of the memory array, the MB86832 and the MB86831 can both allow the writing of data to the RAM to be controlled with either the CAS[3:0] lines or the DWE[3:0] lines. Two configurations are provided: 4 CAS lines and 1 DWE line or 4 DWE lines and 1 CAS line. Bit 7 of the *DRAM Configuration* register selects between these two write modes.

Figure 3 shows a 32-bit memory with a CAS-controlled write mechanism using 1 Mbit x 8 DRAM's.

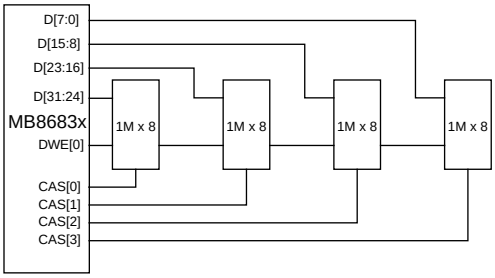

Figure 3. CAS-controlled Write

If a CAS-controlled write architecture is selected, as shown in Figure 10, the DWE[3:0] signals are all output with the same value whenever a write operation is driven onto the bus. Therefore, although only DWE[0] is shown, any of the DWE[3:0] pins can be used to connect to the memory array.

A 16-bit CAS-controlled write architecture looks identical to the right half portion of Figure 10. Data bits D[15:0] are used to transfer data, and CAS[3:2] control the writing of data to the memory array. CAS[1:0] and D[31:16] are not used in a 16-bit memory system.

A 32-bit DWE-controlled write architecture also looks similar to Figure 10, except that CAS[3:0] are replaced with DWE[3:0] respectively, and DWE[0] is replaced with CAS[0]. This design is shown in Figure 4.

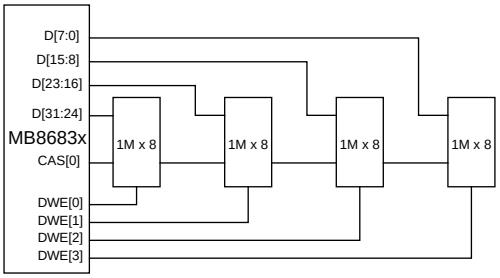

Figure 4. DWE-controlled Write

If a DWE-controlled write architecture is selected, the CAS[3:0] signals are all output with the same value whenever a write cycle is driven onto the bus. Therefore, although only CAS[0] is shown, any of the CAS[3:0] pins can be used to connect to the memory array.

A 16-bit DWE-controlled write architecture looks identical to the right half portion of Figure 11. Data bits D[15:0] are used to transfer data, and DWE[3:2] control the writing of data to the memory array. DWE[1:0] and D[31:16] are not used in a 16-bit memory system.

Figures 3 and 4 are shown only for clarity. In most cases the processor does not connect to the physical DRAM devices themselves, but rather to a memory module such as a common 72-pin Fast Page Mode (FPM) SIMM. Figure 5 shows a connectivity diagram of how the MB8683x would interface to this module.

Note that the $\overline{\text{DOE}}$ signal is not used on an FPM SIMM and therefore would be a no-connect from the processor. Extended Data Out (EDO) modules do use the $\overline{\text{DOE}}$ signal but tie it to ground so that it is always enabled.

DRAM Types

There are both cost and performance issues related to choosing either FPM or EDO DRAMs. FPM DRAMs have the same architecture as standard DRAMs but offer lower access times by allowing the RAS signal to remain asserted as long as successive memory accesses are to the same DRAM page. Figure 6 shows a timing diagram of a Fast Page Mode (FPM) read cycle using the SPARClite DRAM controller. This diagram assumes a 33 MHz bus speed and a 60 ns memory access time. The numbers in parentheses indicate the clock number. The other values above the clock indicate the number of nanoseconds elapsed since the start of the cycle.

In Figure 6, the processor drives the row address based on the rising edge of clock 1 (0 ns). This address is guaranteed to be valid within 20 ns (t_{AVD}) after the rising edge of the clock. RAS is driven by the processor in the following clock and is guaranteed to be valid 15 ns (t_{RVD}) after the rising edge of clock 2 (30 ns). CAS is driven two clocks later and conforms to the same 15 ns valid delay time from the rising edge of clock 4 (90 ns). This means that CAS should be valid at 105 ns into clock 4.

In a 60 ns DRAM, the minimum RAS to CAS delay time is 20 ns. This means that once RAS is asserted, CAS cannot be asserted until at least 20 ns later. This parameter is not an issue in this diagram.

Because the DRAM is asynchronous, data is driven out of the DRAM relative to the falling edge of CAS. Once CAS is asserted, the maximum delay for the DRAM to drive data onto the bus is 15 ns (t_{CAC}). If CAS is asserted at 105 ns, this means that the DRAM should drive data by 120 ns, which is the rising edge of clock 5.

The processor latches the data at the rising edge of clock 6 (150 ns).

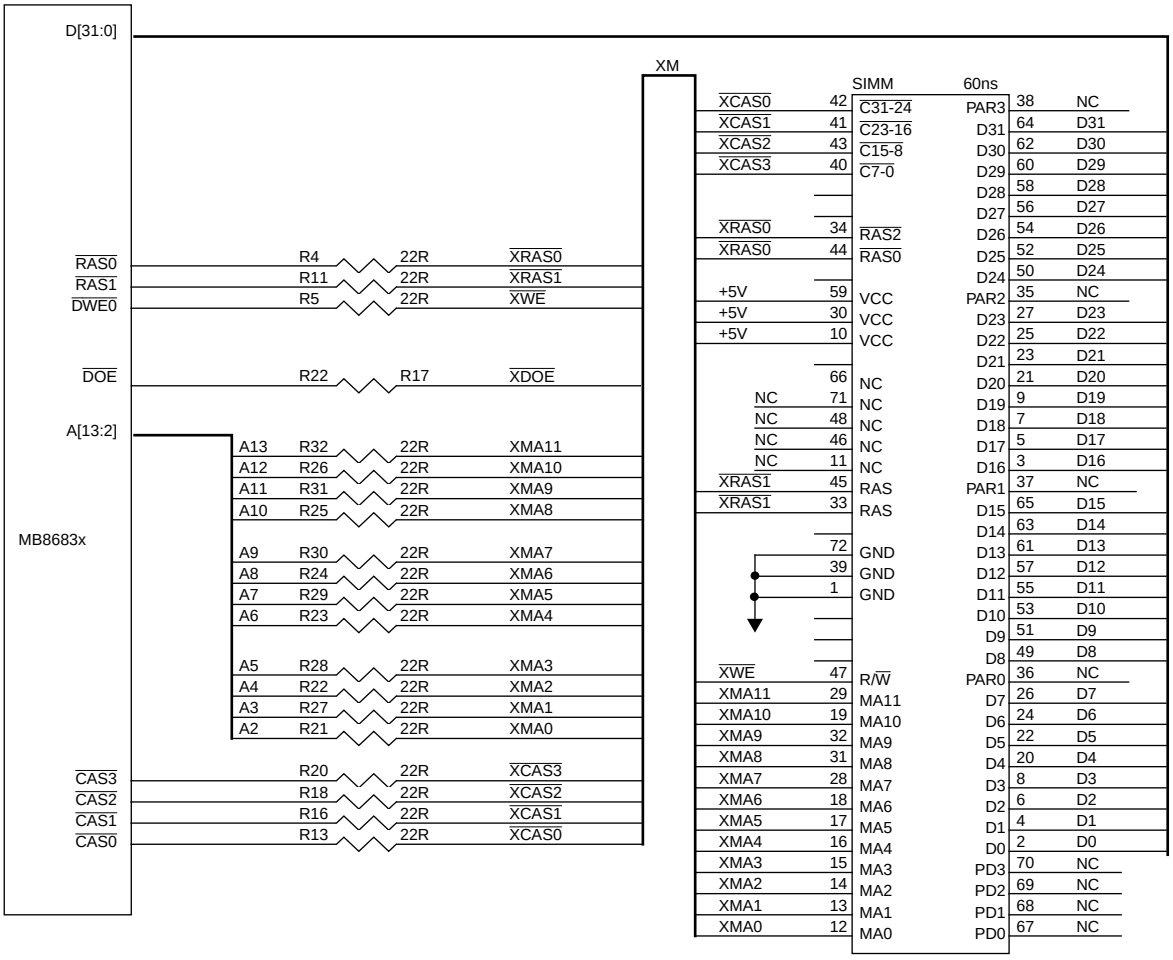

Figure 5. 72-pin SIMM Interface Example

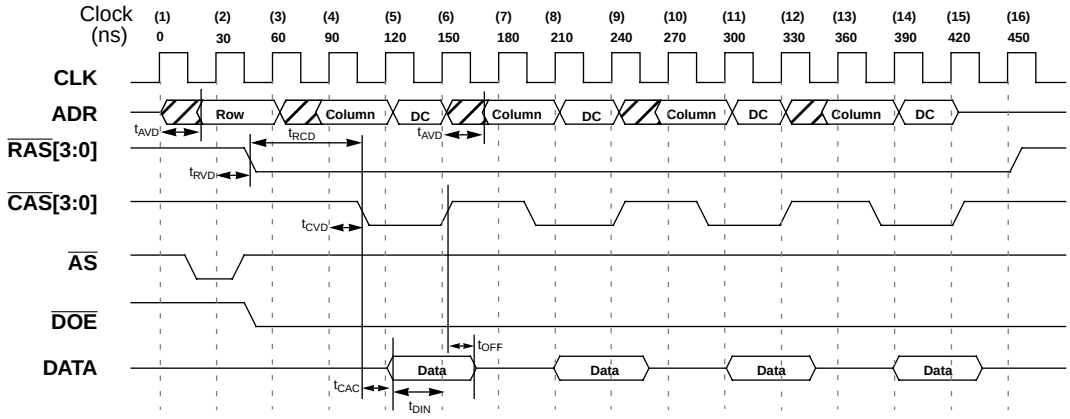

Figure 6. Processor-Initiated Fast Page Mode Read Timing Diagram

This allows data to be valid almost an entire clock (30 ns) before it is latched by the processor, easily meeting the processor's 14 ns data setup time requirement.

Once CAS is deasserted by the processor at the rising edge of clock 6, the DRAM can continue to drive data for a maximum of 13 ns (t_{OFF}).

Figure 3 shows four consecutive read cycles to the same DRAM page, hence RAS need only be asserted once during the operation. If the memory operation requires access to a different DRAM page, RAS must be deasserted and then reasserted after the appropriate precharge times are met. In a 60 ns DRAM, the minimum precharge time is 40 ns (t_{RP}) or two clock cycles in a 33 MHz system.

Table 3 lists the timing values referenced in Figure 6. All values are in nanoseconds.

The write cycle is similar to the read cycle except that data is driven by the processor at the same time as the row address and AS. Figure 7 shows a timing diagram for a fast page mode write cycle using a 60 ns DRAM and a 33 MHz bus speed.

Table 3. CPU-Initiated FPM Read Timings

Symbol	Definition	Min (ns)	Max (ns)
t_{AVD}	Address valid delay		20
t_{RVD}	RAS valid delay		15
t_{RCD}	RAS to CAS delay	20	
t_{CVD}	CAS valid delay		15
t_{CAC}	CAS to data valid		15
t_{OFF}	CAS to data invalid		13
t_{CP}	CAS precharge	10	

The processor drives the address, data, and AS in clock 1 (0 ns). The maximum valid delay time of 20 ns from the rising edge of clock 1 is the same as for the read cycle. At the start of clock 2, the processor drives RAS. Although the DRAM does not require any address setup time relative to RAS assertion (address valid setup time to RAS asserted = 0 ns), RAS cannot be driven in the same clock as address because the maximum valid delay for address is 20 ns from the rising edge of the clock, and the maximum valid delay for RAS is 15 ns from the rising edge of the clock. RAS cannot be asserted before address is valid, which could potentially occur if both are asserted in the same clock. This is a requirement of the DRAM.

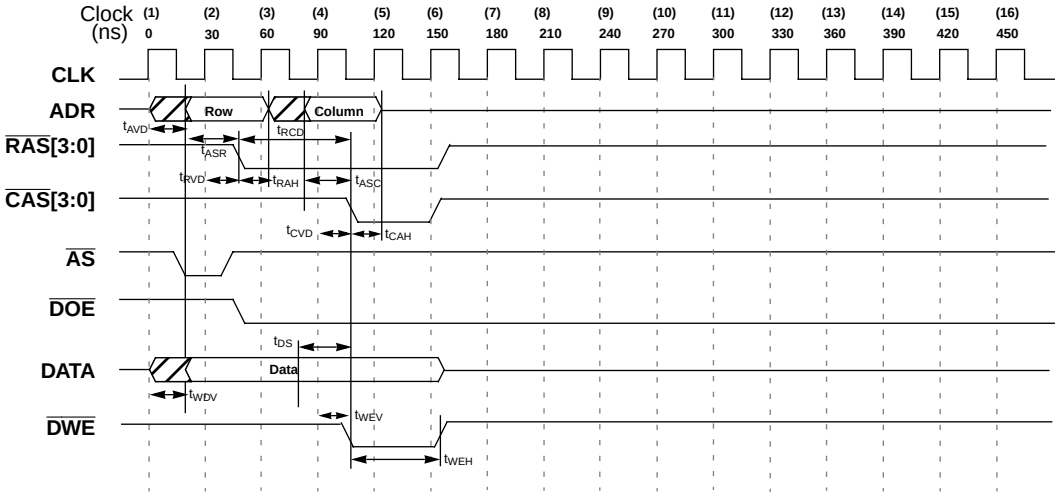

Figure 7. Processor-Initiated Fast Page Mode Write Timing Diagram

Table 4 shows the timing values referenced in Figure 7.

Table 4. CPU-Initiated FPM Write Timing

Symbol	Definition	Min (ns)	Max (ns)
t_{AVD}	Address valid delay		20
t_{RVD}	RAS valid delay		15
t_{RCD}	RAS to CAS delay	20	
t_{CVD}	CAS valid delay		15
t_{ASR}	Address setup to RAS asserted	0	0
t_{RAH}	Row address hold time	10	
t_{ASC}	Address setup to CAS asserted	0	0
t_{CAH}	Column address hold time	10	
t_{DS}	Data setup to CAS assertion	0	0
t_{DH}	Data hold from CAS assertion	10	
t_{WDV}	Write data valid from rising edge		15
t_{WEV}	DWE valid from rising edge		15
t_{WES}	Write enable setup from CAS asserted	0	0
t_{WEH}	Write enable hold from CAS asserted	10	

The DWE and CAS signals each have a 15 ns maximum delay from the rising edge of the clock. According to the DRAM specifications, the setup time from DWE valid to CAS valid is 0 ns. Therefore it is possible to assert both signals in the same clock in Figure 7.

The data setup time for CAS assertion is also 0 ns according to the DRAM specification. This timing parameter is not an issue since data is driven by the processor very early on in the cycle. However, the DRAM requires that data remain valid at least 10 ns after CAS is asserted. In Figure 7 data is shown as being driven for the shortest possible time.

As with any clock-based DRAM controller, such as SPARClite, control signals are driven relative to a clock instead of relative to one another. This means there is a minimum assertion time between signals that is defined by the granularity of the system clock.

EDO DRAM Support

EDO DRAMs incorporate changes to some of the DRAM control signals that allow for increased data throughput. In EDO DRAMs, because data remains valid on the output pins until the next falling edge of CAS, the latching of data in the processor can be overlapped with the next column address precharge. Consequently, burst transfers can be executed up to 25% more quickly.

Figure 8 shows a timing diagram of a 60 ns EDO DRAM 4-transfer read cycle with a bus speed of 33 MHz.

In Figure 8, the rising edge of RAS in clock 1 (0 ns) indicates that the new access is to a different DRAM page, thereby causing a RAS precharge. The MB8683x defines the RAS precharge time as the minimum DRAM RAS precharge time (t_{RP}) + 1 clock cycle. In a 60 ns EDO DRAM the minimum RAS precharge time is 40 ns. At a 30 ns clock cycle this is slightly more than 1 clock. Therefore RAS is held inactive during clock 2 and asserted in clock 3.

RAS is asserted a maximum of 15 ns (t_{RVD}) after the rising edge of clock 3, which is an MB8683x requirement. Row address hold time

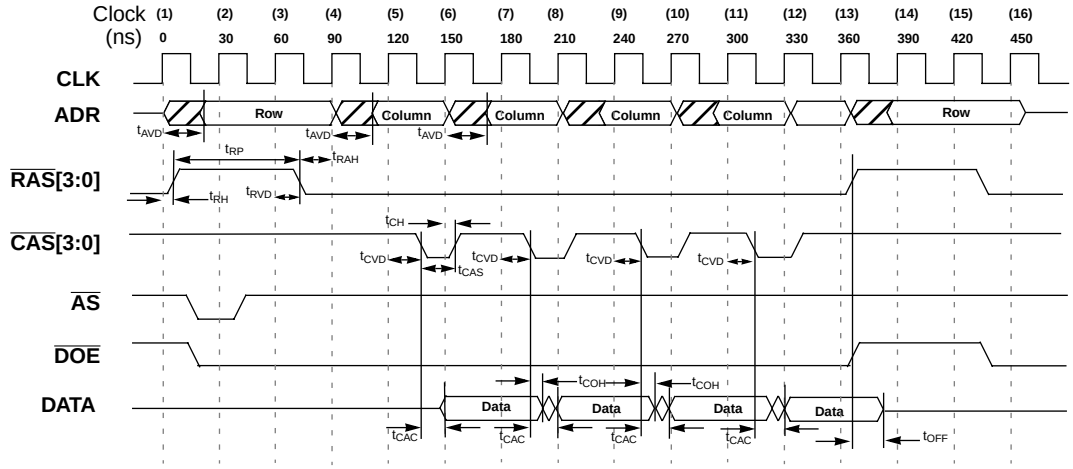

Figure 8. Processor-Initiated EDO Read Timing Diagram

(t_{RAH}) is 10 ns. Since the row address is driven through the end of clock 3, this hold time is easily met.

Column address is driven a maximum of 20 ns (t_{AVD}) after the rising edge of clock 4 (90 ns). Since the maximum output valid delay from the rising edge of the clock is 15 ns with CAS and 20 ns with address (MB8683x specification), CAS cannot be asserted in the same clock as column address. It is delayed by one cycle as shown in Figure 8.

The EDO DRAM guarantees that data will be output within a maximum of 15 ns (t_{CAC}) after CAS is asserted. The CAS valid delay (t_{CVD}) of 15 ns from the rising edge of clock 5 means that data is output by the DRAM at the rising edge of clock 6 ($t_{CVD} + t_{CAC}$ = 30 ns = 1 clock cycle).

As shown in Figure 8, EDO DRAMs allow data to be driven even after the corresponding CAS has been deasserted. This allows CAS accesses to be overlapped. In fact, data continues to be driven by the DRAM until the following CAS is asserted (t_{COH}).

The overlapping of subsequent CAS accesses allows the four transfer reads to be completed in 12 clocks, as opposed to 16 clocks for a FPM read as shown in Figure 6. This equates to approximately a 25% reduction in the total number of clocks required.

Table 5 shows the timing parameters referenced in the EDO read cycle in Figure 8.

Table 5. CPU-Initiated EDO Read Timing

Symbol	Definition	Min (ns)	Max (ns)
t_{AVD}	Address valid delay		20
t_{RVD}	RAS valid delay		15
t_{RH}	RAS hold time	2	
t_{RAH}	Row address hold time	10	
t_{CVD}	CAS valid delay		15
t_{RP}	RAS precharge	40	
t_{CAS}	CAS pulse width	10	
t_{CH}	CAS hold time	2	
t_{CAC}	CAS to data valid		15
t_{COH}	CAS valid to data off	5	
t_{OFF}	Output buffer turn off delay	3	15

Figure 9 shows a timing diagram of a 60 ns EDO DRAM 4-transfer write cycle with a bus speed of 33 MHz.

The write timing diagram is similar to the read timing diagram in Figure 8 except that data is driven by the processor instead of the DRAM (at the same time as column address). The data setup to CAS is 0 ns (t_{DS}). This parameter is not an issue as data is driven at one clock before CAS is asserted.

In Figure 9 the first CAS is driven at 135 ns, or 15 ns after the rising edge of clock 5 (120 ns). This is the maximum output delay time of the MB8683x. The corresponding data is driven through the end of clock 5 and changes based on the rising edge of clock 6. This means that data is driven a minimum of 15 ns after $\overline{CAS}$ is asserted, easily meeting the data hold time from CAS (t_{DS}) of 10 ns.

Table 6 lists the timing values referenced in the EDO write diagram in Figure 9.

Table 6. CPU-Initiated EDO Write Timing

Symbol	Definition	Min (ns)	Max (ns)
t_{AVD}	Address valid delay		20
t_{RVD}	RAS valid delay		15
t_{RH}	RAS hold time	2	
t_{RAH}	Row address hold time	10	
t_{CVD}	CAS valid delay		15
t_{RP}	RAS precharge	40*	
t_{CAS}	CAS pulse width	10	
t_{CH}	CAS hold time	2	
t_{DS}	Data setup to CAS	0	0
t_{DH}	Data hold from CAS	10	

ASIC-Initiated DRAM Cycles

Many applications require that multiple devices in the system have bus master capability. This can include Ethernet controllers, other processors, or the ASIC itself.

If the ASIC is designed for bus mastership, the MB8683x family provides a simple three-pin interface that does not require the ASIC to provide its own internal DRAM or address space control signals. The ASIC requests ownership of the bus by asserting the BREQ input to the MB8683x processor. The processor responds by asserting BGNT, indicating that it has relinquished control of the bus. All relevant processor signals are tri-stated by the MB8683x by the time BGNT is asserted.

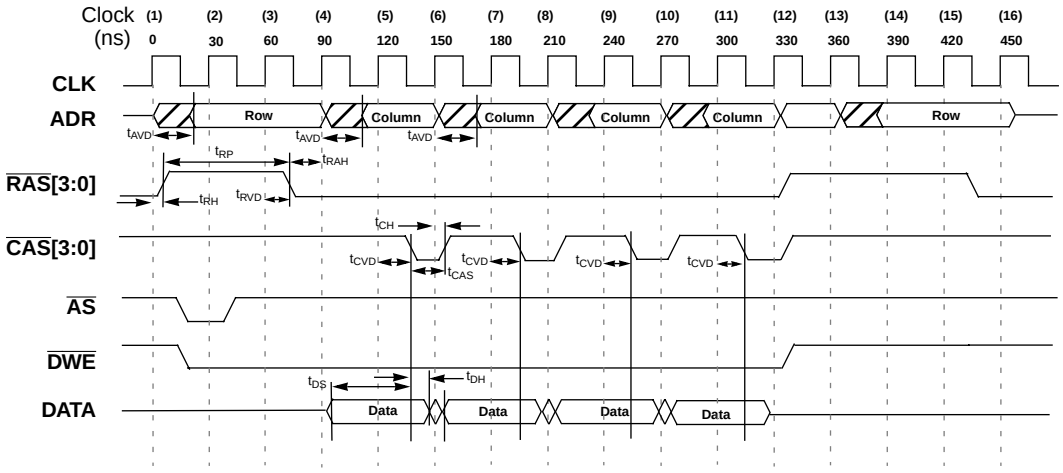

Figure 9. Processor-Initiated EDO Write Timing Diagram

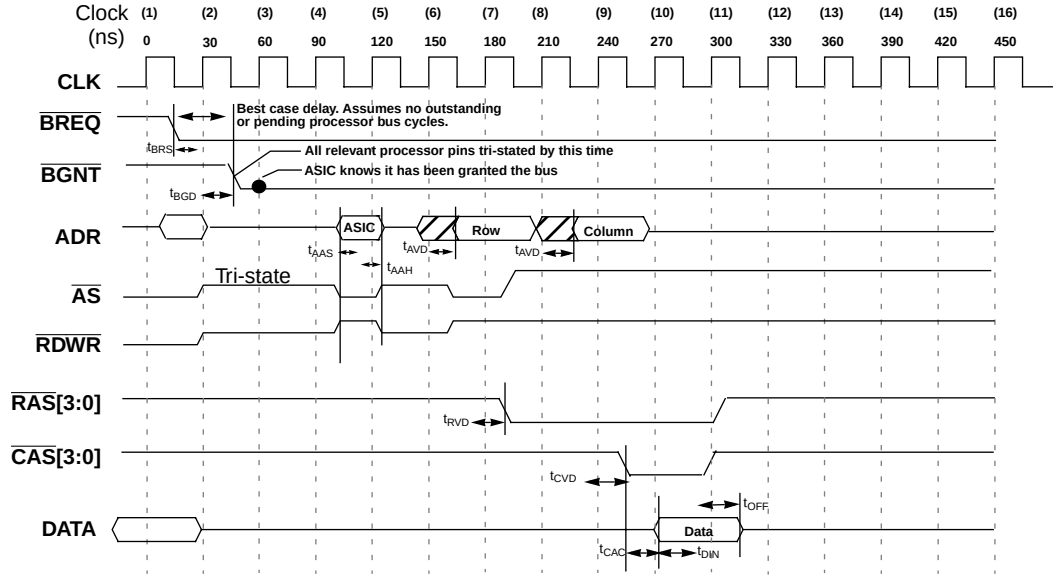

Figure 10. ASIC-Initiated Read Using MB8683x DRAM Controller

Figure 10 shows a timing diagram for an ASIC memory read using the MB8683x DRAM controller. This diagram represents a best-case scenario and assumes that no cycles are currently oustanding on the bus or pending.

If the ASIC is using the DRAM controller in the MB8683x to access memory, then once the ASIC has gained control of the bus, it drives the address and control pins to the processor for one clock. The MB8683x accepts these inputs, then drives the address and appropriate control signals out to the DRAM.

The cycle is initiated when the ASIC asserts BREQ. In Figure 10, a minimum setup time of 12 ns (t_{BRS}) to the rising edge of clock 2 (30 ns) is required by the processor. The processor samples BREQ active at the rising edge of clock 2. If no cycles are currently outstanding on the bus, and none are pending, the processor asserts BGNT in clock 2 (30 ns). However, a maximum BGNT delay time of 18 ns (t_{BGD}) from the rising edge of clock 2 is required by the processor. By the time BGNT is asserted, the processor has tri-stated all relevant address, data, and control pins.

At the rising edge of clock 3 (60 ns) the ASIC samples BGNT active and knows it has been granted control of the bus. One clock later, at the rising edge of clock 4 (90 ns), the ASIC drives address and control signals to the processor for one clock. The output delay for these signals from the rising edge of clock 4 cannot be known. However, the ASIC must meet the 12 ns (t_{AAS}) setup time prior to the rising edge of clock 5 as required by the processor. The required hold time after the rising edge is 2 ns (t_{AAH}). Once the ASIC has provided the address AS and RDWR signals to the processor, it stops driving the bus. The processor requires one clock cycle (clock 5) to process these signals before driving them onto the bus at the rising edge of clock 6 (150 ns).

From the rising edge of clock 6, the cycle is handled just like a standard memory read, as discussed in Figure 6 (processor-initiated FPM read). This includes the driving of the row and column addresses, and the assertion of RAS and CAS.

An internal DRAM ready signal is asserted to indicate to the processor that data is available. The data from the DRAM is accepted by the processor and driven out to the ASIC.

The MB8683x only supports word access for external bus requests to and from memory when the ASIC is using the MB8683x internal DRAM controller.

Table 7 lists the timing parameters referenced in the ASIC-initiated read in Figure 10.

Table 7. ASIC-Initiated FPM Read Timing

Symbol	Definition	Min (ns)	Max (ns)
t_{BRS}	BREQ setup	12	
t_{BGD}	BGNT delay		18
t_{AAS}	ASIC address setup	12	
t_{AAH}	ASIC address hold	2	
t_{AVD}	Address valid delay		20
t_{RVD}	RAS valid delay		15
t_{CVD}	CAS valid delay		15
t_{CAC}	CAS to data valid	15	
t_{DIN}	Input data setup time	14	
t_{OFF}	CAS to data invalid		13

Figure 11 shows a timing diagram for an ASIC memory write using the MB8683x DRAM controller.

The request and grant mechanism is identical to that in Figure 10. The main difference is that data is being driven by the processor at the same time as row address. $\overline{DWE}$ is asserted one clock after CAS for the reasons mentioned in the discussion of Figure 4 (processor-initiated FPM write).

Table 8 lists the timing parameters referenced in Figure 11.

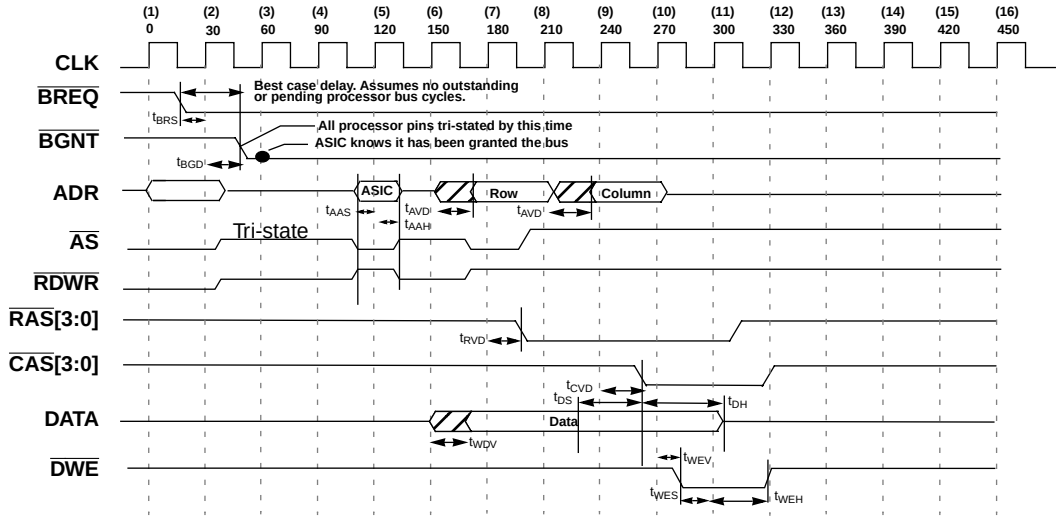

Figure 11. ASIC-Initiated Write Using MB8683x DRAM Controller

Table 8. ASIC-Initiated FPM Write Timing

Symbol	Definition	Min (ns)	Max (ns)
t_{BRS}	BREQ setup	12	
t_{BGD}	BGNT delay		18
t_{AAS}	ASIC address setup	12	
t_{AAH}	ASIC address hold	2	
t_{AVD}	Address valid delay		20
t_{RVD}	RAS valid delay		15
t_{CVD}	CAS valid delay		15
t_{DS}	Data setup time to CAS valid	0	0
t_{DH}	Data hold time from CAS valid	10	
t_{WEV}	DWE valid		15
t_{WES}	DWE setup	0	
t_{WEH}	DWE hold from rising edge	10	
t_{WDV}	Data valid from rising edge		20

In addition to the BREQ and BGNT signals discussed above, a third signal, PBREQ, can be asserted by the processor while in slave mode to indicate to the current bus master that it is requesting immediate control of the bus. This signal is asserted when the write buffer is full, or when the MB8683x needs to do an instruction or data fetch from memory.

ASIC-Based DRAM Controller

The fastest memory interface would be to have the ASIC incorporate its own DRAM controller. The ASIC would still have to arbitrate for the bus, but once granted control of the bus, it would drive its own address and DRAM control signals. Using the same 60 ns DRAM and 33 MHz bus speed, the fastest possible read cycle would require 4 clocks as shown in Figure 12. This figure assumes the following:

- Since the row address setup time to RAS active is 0 ns, both of these signals can be asserted in the same clock if the output timing is tightly controlled. This operation requires one clock cycle.
- Since the column address setup time to CAS active is 0 ns and the $\overline{OE}$ setup time to CAS valid is also 0, all of these signals can be asserted in the same clock if the output timing is tightly controlled. This operation requires one clock cycle.
- If CAS and OE are asserted within 5 ns of the rising edge of the clock, which is pretty much the best case scenario, data would be available 15 ns after CAS is asserted as specified by the DRAM. This would make data available at 50 ns. This operation requires one clock cycle.
- The following rising clock edge occurs at 60 ns. This allows a maximum of a 10 ns setup time for data. However, this time does not meet the 14 ns minimum setup time required by the processor. Therefore data cannot be latched by the processor until the following clock cycle. Data is latched on the 4th clock of the operation.

Figure 12 shows how this fast memory cycle might look. Table 9 shows the relative memory performance between the three type of memory accesses.

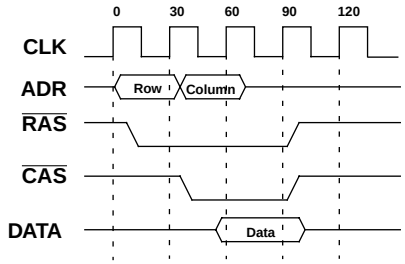

Figure 12. ASIC-Initiated Read Using On-Chip ASIC DRAM Controller

Table 9. Relative Memory Performance (in Clocks)

Cycle	MB8683x as Bus Master (1st Data)	ASIC as Bus Master Using MB8683x DRAM Controller (1st Data)	ASIC as Bus Master with Fast On-Chip DRAM Controller (1st Data)
FPM Read	7	12	4
EDO Read	6	—	—
FPM Write	6	11	3
EDO Write	5	—	—

Burst Mode

When a bus request requires more than a single data transfer, the MB8683x can perform a 4-transfer burst cycle. Burst cycles allow for the maximum data rate by eliminating the unnecessary driving of the address bus. In a burst cycle, only the first address of the transfer is driven by the processor. External logic, or the DRAM device itself, calculates the remaining three addresses based on a fixed subblock ordering protocol.

In non-burst mode, an address must be driven by the processor to access each desired memory location. Burst mode allows the processor to generate only one address to access four memory locations, thereby reducing CPU overhead and facilitating the ability to return the contents of an entire data cache line in only one memory access.

For example, 32-bytes is the line size for the MB86831 and MB86832 data caches. A 32-bit wide non-interleaved memory array means that each time the memory is accessed, 32-bits are returned. But in burst mode, 128-bits (32-bits x 4 accesses) are returned. This number equals one half of the 32-byte line size required by the processor. Therefore, a cache line can be filled in only two memory accesses.

In the MB8683x, burst mode transfers are initiated on either an instruction or data cache miss and are supported only for instruction fetches and data cache loads. Stores are not supported.

Table 10 shows the sequence in which memory is accessed during a burst transaction. The values shown represent the state of the two least-significant address bits. In a 32-bit memory array this would equate to address bits A3 and A2.

Table 10. Burst-Mode Sequence

1st Address	2nd Address	3rd Address	4th Address
00	01	10	11
01	00	11	10
10	11	00	01
11	10	01	00

Although the burst-mode sequence may look complex, it is really quite simple. For example, in a 32-bit wide memory, the addresses increment by complementing the value of A2 after every transfer and complementing the value of A3 after every other transfer, regardless of the starting address.

In order to use burst mode, the memory system must be able to support burst transfers. Burst-mode accesses from PROM/EPROM space are not supported in 8- or 16-bit bus mode.

The MB8683x incorporates a simple two-signal interface for burst transactions. The BMREQ signal is output by the processor to inform the memory array that the upcoming transaction is a 4-transfer burst. The BMACK signal is driven by the memory array to inform the processor that it can support burst-mode transfers. However, if the burst mode bit in the *Bus Control* register is set, the memory system need not return BMACK. In this case BMREQ is asserted on either an instruction of data cache miss, but does not require BMACK to be sampled active before the transaction begins.

Programmable Chip Selects

Each peripheral device in a design resides within a given address space. Decode logic must decode the address on the bus, determine which peripheral is being accessed, and respond accordingly. The types of peripherals and the amount of memory space each requires can easily change within different revisions of the same product, requiring maximum flexibility of the memory controller. The chip select logic incorporated in SPARClite eliminates the need for this function in the ASIC.

The SPARClite MB8683x architecture contains six programmable chip selects, each with its own address decoder, wait state generator, and bus-width selector. The chip selects are accessed by load and store instructions. The corresponding chip select output is asserted whenever the value on the address bus falls within a programmable address range specified in the corresponding *Address Range Specifier* register. There are five *Address Range Specifier* registers available which correspond to chip selects CS[5:1]. The first chip select CS[0] is hardwired to zero and is normally used for ROM accesses. Each register supports up to a 23-bit address with an 8-bit address space identifier (ASI).

In addition to the *Address Range Specifier* registers, each chip select also contains a corresponding *Address Mask* register which is used to mask certain bits of the address. Only those bits of the *Address Range Specifier* register are compared for which the corresponding mask bits are zero.

Non-Cachable Memory Support

The SPARClite MB8683x allows for portions of memory space to be made non-cachable by overlapping the programmable memory ranges of successive chip selects. This function is useful in designs where the ASIC can also access memory.

For example, the memory ranges corresponding to programmable chip selects CS4 and CS5 can be overlapped, allowing MB8683x space to be cached, while the ASIC data space within the same DRAM can be forced by hardware or software to be non-cacheable.

EPROM Interface

The MB8683x can be interfaced directly to an EPROM with no external glue logic. ROM devices tend to operate at slow speeds relative to the rest of the system and hence can require a fairly long period of time to turn off their data drivers. The MB8683x provides the IDLEEN pin to deal with this issue. For example, the EPROM is always tied to chip select 0 (CS0). If IDLEEN is asserted, the next cycle following the access to CS0 address space is delayed for two clock cycles to give the EPROM adequate time to stop driving the bus. This signal can be tied either high or low. Figure 13 shows a connectivity diagram of an EPROM interface.

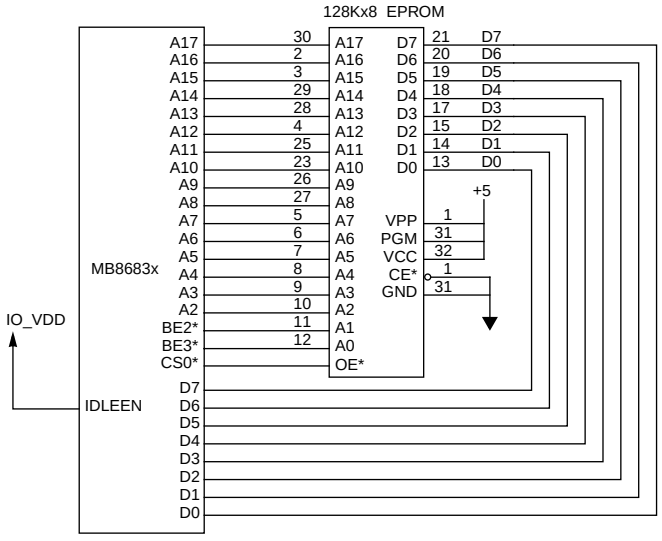

Figure 13. EPROM Interface Example

In this example the EPROM chip enable signal is always asserted. When the device is accessed the processor asserts the CS0 output signal. This signal is in turn used to enable the output buffers of the EPROM. Once the operation is completed and CS0 is deasserted, the next cycle will be delayed by two clocks if the IDLEEN signal is tied to IO_VDD.

Cache Architecture

Caches play an important role in the overall performance of most applications. For example, the majority of image processing applications use high-repitition count loops that offer greatly increased performance when run out of a cache as opposed to memory.

Figure 14 shows an example of a high-repetition count loop such as histogramming used for pixel processing with and without the cache.

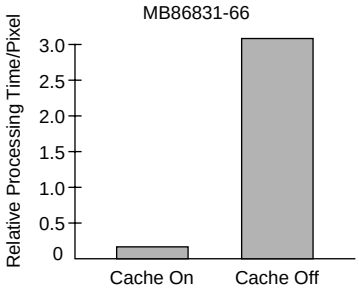

Figure 14. Cache Performance Example

For critical or frequently executed code, the SPARClite offers a "locking"mechanism. Cache locking allows critical code or data segments to be locked in the caches. The locked contents can be updated on a store hit, but cannot be selected for replacement on a miss.

The cache sizes vary for each member in the MB8683x family, allowing for different price-performance points within a product line with little or no redesign required.

The MB8683x family supports two cache locking modes:

- Local lock mode
- Global lock mode

Local cache locking dynamically locks selected instructions or data entries into the appropriate cache. This allows repetitive loop routines to be locked into the cache without the danger of being overwritten on a cache line fill.

In image processing applications, for example, the performance of filtering operations can be significantly improved by locking the convolution matrices and other frequently used constants (coefficients) in the data cache. The convolution process modifies or extracts information on an image based on the proximity of pixels to one another and applies them either in a spatial or frequency domain. Spatial examples include:

- Low-pass filtering, which blurs the image.
- High-pass filtering, which sharpens the image.

- Edge detection, which sharpens the contrast between the different constituents of an image.

A typical 3 x 3 convolution requires the following instruction mix:

- 9 multiplies
- 12 additions
- 3 loads
- 1 store
- 1 shift
- 4 compares

This highly-repetitive loop is used for each pixel processed and can be "locally" locked in the cache.

Figure 15 shows the instruction mix profile for the 3 x 3 convolution.

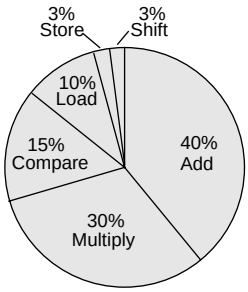

Figure 15. Instruction Mix Profile

To facilitate the fast execution of repetitive loops in image processing and other applications, each member of the SPARClite MB8683x family incorporates a fast hardware multiplier, allowing an 8x8 multiply-accumulate to be executed in three clock cycles.

The program code for most image-processing operations makes use of a limited number of subroutines which can be repetitively called thousands of times during an operation. Such code fragments or inner loops can be locked in the instruction cache. The locking feature for both the data and instruction caches in the SPARClite ensures deterministic response and the highest performance for critical or frequently recurring routines employed by most image processing applications.

Global cache locking allows the entire contents of either cache to be locked. With the entire cache locked, no entries in that cache can be replaced. Global cache locking is useful when executing a large code segment that must remain intact in the cache.

Interrupt Control

It is common in most designs to have multiple peripheral devices able to generate interrupts. In the digital camera block diagram in Figure 1, interrupts can be generated from numerous sources, including the CCD array, serial port, PCMCIA interface, NTSC/PAL interface, LCD controller, etc.

However, depending on the programmability of the device's interrupt logic, a situation could arise where different devices are generating different types of interrupt signals, some of which may be active HIGH, active LOW, rising-edge triggered, or falling-edge triggered.

The MB8683x family contains two modes that allow for up to 16 external interrupts. One mode allows each of the eight external interrupt signals to be dedicated to a given peripheral, allowing for a maximum of eight interrupts. Each interrupt channel can be independently programmed to respond to any of the previously mentioned interrupt types, thereby eliminating the need for any external interrupt glue logic. Figure 16 shows a connectivity diagram for this mode.

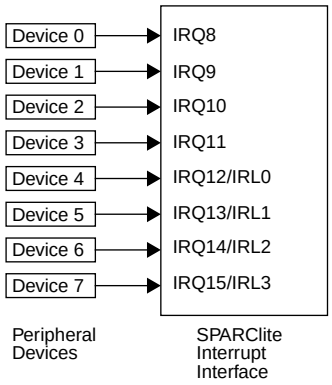

Figure 16. Dedicated Interrupt Mode

A second mode encodes the lower 4 interrupt pins, allowing for a total of 16 external interrupts. However, this mode requires external encode logic. Figure 17 shows a connectivity diagram for this mode.

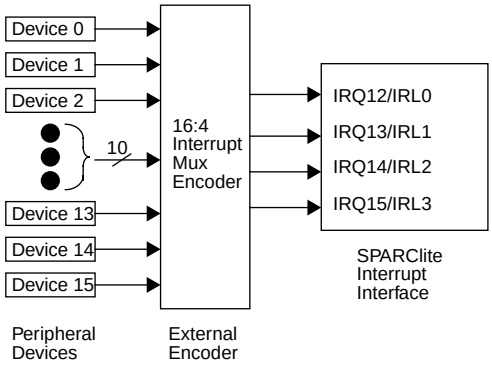

Figure 17. Encoded Interrupt Mode

In the MB8683x architecture, an interrupt is defined as a type of *trap*. There are 32 basic trap types defined in order of priority. The 15 possible external interrupts (encoded interrupt mode, see Figure 15) reside in priority levels 17 through 31 of the trap table. This means that the highest priority interrupt (15) is actually at priority level 17 of 32 in the trap hierarchy.

For simplicity, the MB8683x family contains a single vector trapping facility. All traps and interrupts go to a single location specified by the upper 20 bits of the *Trap Base* register. This register contains the base address of the trap table. The next 8 bits are the *trap type* field that distinguishes up to 256 different trap types. The *trap type* field is used as an offset into the trap table. This table contains the first four instructions of each service routine. From the trap table, control is transferred to the appropriate trap handler via a Jump and Link (JMPL) or Branch All (BA) instruction.

Figure 18 shows a block diagram of the interrupt mechanism.

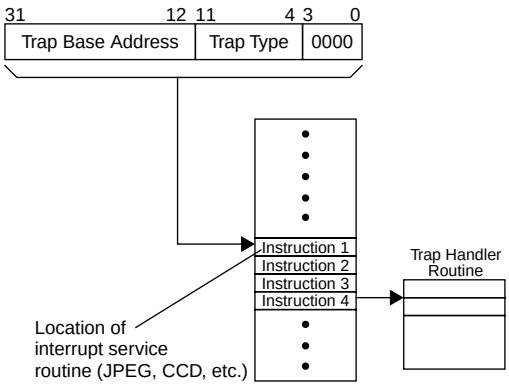

Figure 18. Interrupt Block Diagram

When an external interrupt is generated it is latched by the processor. However, the amount of elapsed time before the interrupt is serviced depends on the location of the interrupt service routine and whether any other interrupts are currently being serviced. Assuming that no other interrupts or traps are pending, processor hardware performs the following sequence once an interrupt is sampled:

- Latches the state of the external interrupt signals
- Writes the trap type number into the *tt* field in the *Trap Base* register
- Saves the current processor mode (user or supervisor) by copying the current value of the *Supervisor* (S) bit of the *Processor Status* register into the PS bit.
- Enters supervisor mode by setting the *Supervisor* bit in the *Processor Status* register
- Disables traps by clearing the Enable Trap (ET) bit in the *Processor Status* register
- Saves the window of the interrupted routine by decrementing the Current Window pointer (CWP)
- Stores the current Program Counter and Next Program Counter values in r[17] and r[18] of the new window
- Transfers control to the address specified by the *Trap Base* register.

Note that interrupts are serviced in Supervisor mode.

Asynchronous traps require an additional cycle to be recoginzed by the processor versus synchronous traps. For example, a synchronous interrupt that is asserted at least one nanosecond before the rising edge of the clock causes the seven operations listed above to be performed by hardware. If the trap table and ISR routines are located in the on-chip cache, the first instruction of the interrupt service routine is fetched from the trap table within seven clocks after the interrupt is sampled active. Figure 19 shows a timing diagram of the trap sequence.

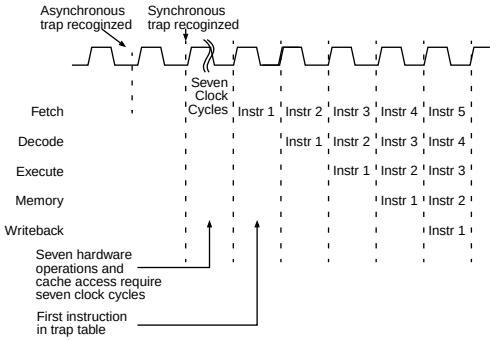

Figure 19. Interrupt Latency Diagram

Therefore, in a best-case scenario using a synchronous interrupt, where the interrupt service routine is already in the cache, the routine begins execution seven clocks after the interrupt is recognized by the processor. Once an interrupt service routine has started, it is allowed to finish, regardless of whether any higher-priority traps are taken.

The trap table and ISR routines can also be located in either DRAM of ROM. If DRAM is used, software can either move the appropriate interrupt service routine into the cache, or allow it to execute out of DRAM. The amount of time required to handle the interrupt depends on the size of the routine.

Register Windowing

The latency required for context switching can have a profound effect on overall system performance. To minimize the context switching latency, MB8683x processors incorporate a register window architecture.

During a context switch, the *Current Window Pointer* (CWP) can be decremented to point to another window without the need to waste processor cycles storing parameters or data values to another location.

Register windowing decreases the time required for subroutine calls. Parameters are passed between processes by writing to the *out* registers of the current window, which are the same as the *in* windows of the adjacent register. The *SAVE* instruction decrements the CWP, making the parameters available to the next process without having to move any data. Because the interrupt routine has its own window, there is no need to save context before executing the *TRAP* instruction.

The register windowing mechanism contains 128 general purpose registers. They are organized into 8 overlapping windows, each having 24 switchable registers available and access to 8 additional global registers, for a total of 136 registers.

Figure 20 shows how register file windowing is implemented in the SPARClite architecture.

The *in* and *out* registers are used primarily for passing parameters to subroutines and receiving results from them. The *local* registers are used for storing values whose lifetime is typically not longer than the lifetime of the routine being executed. The *in* registers of Window 7 are shared with the *out* registers of Window 0.

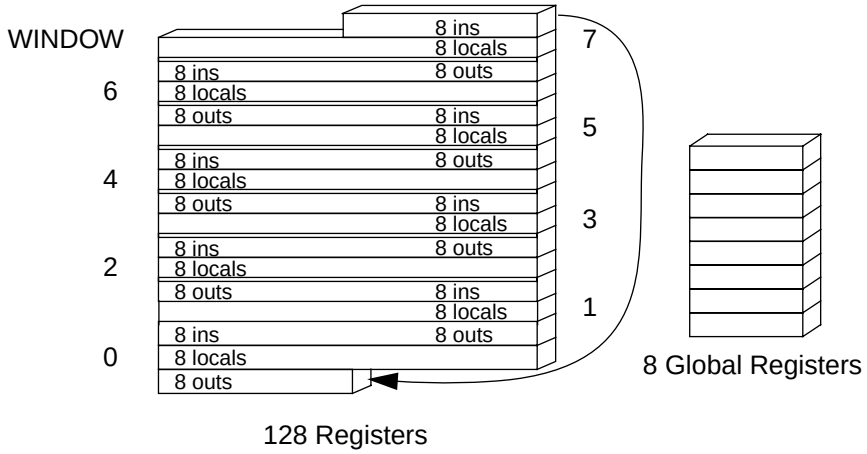

Figure 20. SPARClite Register Windowing Mechanism

Parameters are passed between subroutines by writing to the *Out* registers of the current window, which are the same as the *In* registers of the adjacent window. A special SAVE instruction decrements the CWP, making the parameters available to the next subroutines without having to move any data.

When an interrupt occurs, the processor automatically decrements the CWP, giving the interrupt routine its own window, eliminating the need to save the current context before executing the trap instruction.

For interrupt priority programming, three modes are provided. Priority modes 1 and 2 support varying numbers of external and internal interrupts and adhere to a fixed interrupt priority scheme. Priority mode 0 causes the interrupt to bypass the priority encoder and proceed directly to the integer unit (IU).

The register windowing mechanism in the SPARClite architecture is well-suited for image processing applications. For example, a typical imaging application requiring a 7 x 7 convolution requires a total of 98 registers: 49 to maintain the coefficients and 49 to maintain pixel color channel (RGB) integer data. The large number of registers in the SPARClite architecture allows this operation to be performed without any memory accesses. In a typical 32-register RISC processor, all of the data that overflows the register set would have to be brought in from the cache at the cost of one instruction per load.

Clocking Mechanism

Depending on the types of peripheral devices used, one design can require multiple clock frequencies. ASICs typically do not run at the same speed as the processor, and other slower 8-bit peripherals such as ROM's and UART's can require a clock frequency that is only a fraction of the processor frequency. Each member of the MB8683x family includes a clock multiplier that allows the processor core to run at much higher frequencies than the system bus. The slower bus frequency allows for the use of low-cost RAM, ROM, and peripheral devices.

In the MB8683x family the maximum bus speed is 40 MHz. The processor multiplies the input clock frequency by a ratio based on the state of the CLKSEL[2:0] pins at reset to generate the processor core frequency, as shown in Table 11.

Table 11. Selecting the Clock Ratio

CLKSEL[2:0]	Internal Clock Multiplied by
100	1
101	2
110	3
111	4
011	5

Each member of the MB8683x family can multiply the clock frequency by a maximum value of 5. In the MB86833, this allows for a very low input frequency that is ideal for low-cost, lower-performance designs. For example, a 66 MHz core frequency using a x5 clock multiplier allows for an input clock of only 13.2 MHz.

The MB86833 operates only at 66 MHz. The speed of the external bus should be designed so that when this value is multiplied internally, it equals 66 MHz. For example, a 33 MHz input clock can be used as the bus frequency, then multiplied by 2 internally, to derive 66 MHz.

Figure 21 shows an example of the MB8683x clock distribution mechanism with an input frequency of 25 MHz.

Figure 22 shows a connectivity diagram for the clock pins of the MB8683x. The CLKSEL[2:0] pins are connected to jumpers and can be selected to provide one of five clock ratios as shown in Table 11. Connecting the jumper drives a low voltage onto that pin. If there is no jumper connected, the weak pullup resistor drives a high voltage level onto the pin.

The CLKEXT pin can be used to bypass the internal PLL. A high voltage level causes the PLL to be bypassed. This pin should be jumpered for normal operation, indicating a low voltage level. Refer to Figure 21 for a functional representation of the PLL mechanism.

Endianness

The SPARClite MB8683x is a Big Endian architecture. This means that, in a 32-bit data word, byte 0 occupies the most-significant byte of the word (data bits D[31:24]).

Each member of the MB8683x family contains a 32-bit data bus, which contains four bytes. This means that address bits A1 and A0 are not available as external signals. Rather, 4-byte enable signals BE[3:0] are used to select one of these 4 bytes. In a 32-bit system, the address increments by 4 each time, as shown in Figure 23.

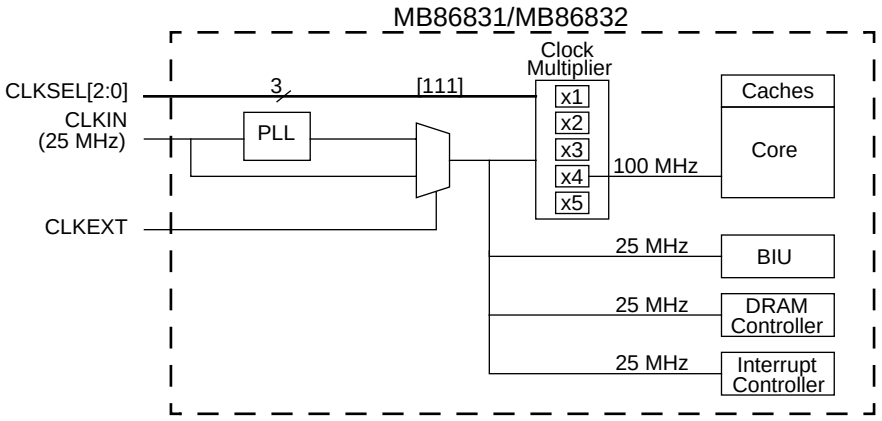

Figure 21. MB8683x Clock Distribution Mechanism

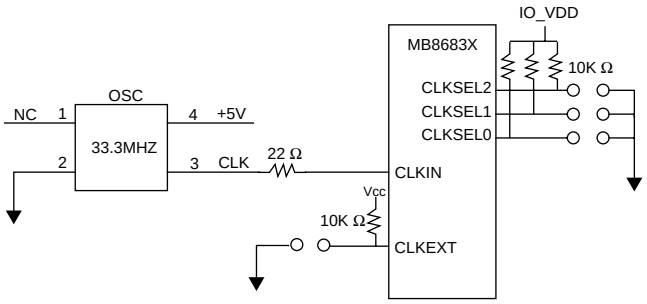

Figure 22. Clock Connectivity Diagram

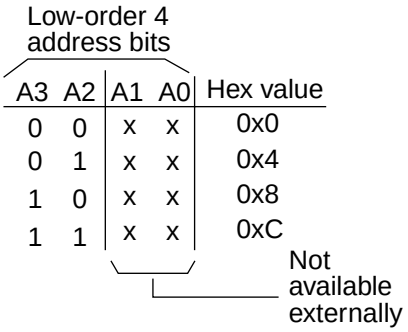

Figure 23. Incrementing Addresses

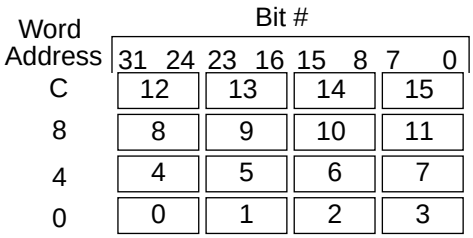

Figure 24. Big Endian Byte Ordering

Figure 24 shows an example of Big-Endian byte ordering in a 32-bit memory system. The number inside each box indicates the byte.

Once a 32-bit location is accessed, the BE[3:0] signals are used to indicate which bytes of the word are to be written. On a read, all of the byte enables are always asserted, and the entire 32-bit value is fetched, regardless of which bytes are actually valid.

Power Management

Power management is very important in battery-operated applications where there is a need to conserve power, but not at the expense of performance. The MB8683x family contains a SLEEP mode of operation where only the main clock remains active, thus lowering power consumption by approximately 90% to 10 mA. For example, at 66 MHz in SLEEP mode, the MB86831 consumes only 33 mW.

All members of the MB8683x family can be configured to operate off of a 3.3 volt supply and interface directly to 3.3 volt peripherals. However, 5 volts can also be applied to the I/O pins, allowing 5 volt tolerant peripherals and memory to be used in the same system.

The MB8683x family contains three external power management pins: PDOWN, WAKEUP, and FLOAT. Each of these can be used in conjunction with an ASIC to provide power management capability to the entire system.

If the ASIC supports bus-mastering, it is possible for the ASIC to access memory and other system peripherals while the MB8683x is in SLEEP mode. This can be accomplished using the following procedure:

1. Software executes code that places the MB8683x in SLEEP mode.
2. ASIC waits for the MB8683x to assert PDOWN, indicating that it is in SLEEP mode.
3. The RAS and CAS lines in the MB8683x are switched to ASIC control.
4. The ASIC asserts FLOAT to the MB8683x, causing the processor to float all address, data, and control lines.
5. ASIC now has control of the bus as long as the processor is in SLEEP mode.

To wake up the processor:

1. The ASIC relinquishes control of the bus and drives RAS and CAS low, placing the DRAM in self-refresh mode.
2. The ASIC deasserts the FLOAT signal and waits 2 clocks for the bus to stabilize.
3. The ASIC asserts the WAKEUP signal and waits for the processor to respond by deasserting PDOWN.
4. The processor is now out of SLEEP mode and has regained control of the bus.

Note that since the IRC clock is connected to the CPU core resource clock, the IRC clock stops when the CPU is placed in SLEEP mode. In addition, the IRC logic is suspended when the processor asserts the Bus Grant (BGNT) signal.

Migration Path

Factors such as cost, performance, and functionality all play a role in choosing the type of processor for a given application. The SPARClite MB8683x family offers many features that allow for easy migration between processors.

This section discusses the differences between the three processors in the MB8683x family. These differences are listed in Table 12.

Table 12. Differences Between MB8683x Family

Parameter	MB86831	MB86832	MB86833
Clock Speeds (MHz)	66/80/100	66/80/100	66
Instruction Cache Size	4 kbytes	8 kbytes	1 kbyte
Data Cache Size	2 kbytes	8 kbytes	1 kbyte
Address Bus Width	28-bit	28/32-bit	24/28-bit
Number of RAS Signals	4	4	1
Number of DRAM Write Enable Signals	4	4	1
ASISEL signal	No	Yes	Yes
Pin Compatible	With ‘832	With ‘831*	None
Debug Support	No	Yes	No
Pull-up Resistors on CLKSEL2	Yes	Yes	No
Pull-up Resistors on ASISEL	No ASISEL Pin	Yes	Yes

* With the exception of the ASISEL pin

Refer to the *MB8683x Family Overview* section at the beginning of this document for a breakdown of features for each product.

During the design phase of a product is when the ability to emulate a part is the most critical. The MB86832 contains a hardware emulator interface, debug support registers, on-chip breakpoint and single-step logic, and ten dedicated interface signals and logic for supporting In Circuit Emulators (ICE) and debug monitors. Therefore, the MB86832 is a logical choice for an initial design. Each of the following examples assumes that the MB86832 is used in the prototype design and discusses how to design the prototype so that it most closely resembles the production version.

Address Bus Widths

Each member of the MB8683x family supports a different maximum width for the address bus. Understanding these addressing schemes and how they differ allows for an easier migration from one processor to another.

Table 13 gives an overview of how the address busses for each processor are configured.

The MB86831 contains a 28-bit dedicated address bus that supports a 256-Mbyte memory space. The ASI[3:0] pins, which are driven along with each address, allow the user to define up to 16 different 256-Mbyte address spaces with a single 28-bit bus. No optional address signals are supported.

Table 13. Address Bus Widths

Address pins	MB86831	MB86832	MB86833
Dedicated	27:2	27:2	23:2
Optional	None	31:28	27:24
ASISEL pin	No	Yes	Yes
ASI[3:0] pins	Yes	Yes	Yes

The MB86832 also contains a dedicated 28-bit address bus. However, the ASI[3:0] signals are multiplexed with optional address bits A[31:28], yielding a maximum address bus size of 32-bits. The ASISEL signal is used to select between the ASI[3:0] signals and the optional address bits. If ASISEL is high, these pins function as ASI[3:0]. This yields up to 16 different 256-Mbyte address spaces. If ASISEL is low, these bits are driven as optional address bits A[31:28], yielding a single 4-Gbyte address space.

The addressing scheme of the MB86833 is similar to that of the MB86832, except that there are not as many address bits. There are 24 dedicated bits, with four optional bits A[27:24] multiplexed with ASI[3:0], which is the same as in the MB86832. The ASISEL signal selects between the two. Table 14 lists how ASISEL affects the address space of each processor.

Table 14. Configuring Memory Space

ASISEL pin	MB86831	MB86832	MB86833
Low	n/a*	(1) 4 Gbyte space	(1) 256 Mbyte space
High	n/a*	(16) 256 Mbyte spaces	(16) 16 Mbyte spaces

* The MB86831 contains up to, (16) 256 Mbyte address spaces using ASI[3:0]. No optional addresses are supported and the ASISEL signal is not available.

If the MB86831 is the processor to be used in production, the initial design using the MB86832 should tie the ASISEL signal high as this pin is a no-connect in the MB86831. Tying the signal high enables the ASI[3:0] signals, which are supported on the MB868831, and disables optional addresses A[31:28], which are not available on the MB86831. Figure 25 shows how to configure the MB86832 to emulate the MB86831 addressing scheme.

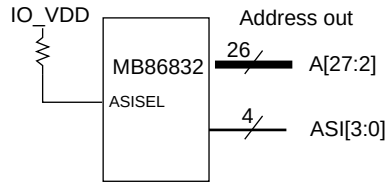

Figure 25. Emulating ’831 Address Space

If the MB86833 is the production processor, the uppermost 4 bits of optional address A[31:28] should not be used because they are not available on the MB86833. This is accomplished by tying the ASISEL signal high to enable ASI[3:0] as these signals are supported on both devices. In addition, A[27:24] should be left unconnected as these bits are only available on the MB86833 when ASISEL is low.

If ASISEL is tied low, the upper address bits on the MB86832 are enabled, and these bits are not available on the MB86833. Figure 26 shows how to configure the MB86832 address to emulate the MB86833 addressing scheme. This configuration enables sixteen 16 Mbyte address spaces.

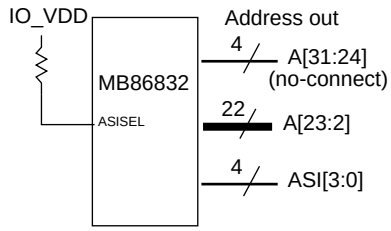

Figure 26. Emulating ’833 Address Space

RAS and DWE Signals

The MB86831 and MB86832 support four row address select signals RAS[3:0] and four write enable signals DWE[3:0]. However, the MB86833 supports only one RAS signal and one DWE signal.

If the MB86833 is targeted as the production processor, the MB86832 based prototype memory array should be designed so that RAS0 is the only RAS line used throughout the entire array.

This is because the MB86833 contains only one RAS line.

The same holds true for the write enable. DWE0 should be the only write enable line used throughout the entire memory array as the MB86833 contains only one DWE signal.

ASISEL Signal

The ASISEL signal is used on the MB86832 and MB86833 processors to select between the ASI[3:0] signals and the uppermost four bits of optional address. Refer to the *Address Bus Width* section for more information.

Pin Compatibility

The MB86832 and MB86831 members of the SPARClite MB8683x family are pin compatible. The only external differences are:

- The ASISEL signal on the MB86832 is a no-connect on the MB86831
- Address bits A[31:28] are not available on the MB86831
- All of the debug support pins on the MB86832 are no-connects on the MB86831

The main internal difference is the size of the caches. Both devices can operate up to 100 MHz and contain the same memory interface. This allows the migration from the MB86832 to the MB86831 to be made without having to redesign the board.

The MB86833 is not pin compatible with the MB86832 or the MB86831. Differences in the address bus and memory interface are discussed in the above sections. To minimize the level of redesign for the production product, it is suggested that the MB86832 board design conform to the resources available on the MB86833. This requires specific layout considerations for the memory array, such as the number of RAS and DWE signals, as noted in the *RAS and DWE Signals* section.

Software and Debug Support

Fujitsu offers a very competitive suite of development tools that allow customers to significantly reduce their time-to-market. There are five complete turnkey development environments for the SPARClite family, giving customers a variety of choices to meet their needs. Each member of the SPARClite family has evaluation boards and Verilog behavioral models available from Fujitsu Microelectronics.

The MB86832 has a built-in emulator bus to support in-circuit emulators up to 40 MHz. These emulators provide an interface to a compiler/assembler tool-chain for source-level debugging. Hardware breakpoint logic supports a single-step instruction flow, two instruction addresses, two data addresses, and maskable data values.

MB8683x System Solutions

Fujitsu offers a three-step system solution for migrating a product from prototype or initial offering to mass-production. These steps are shown below.

Solution I: Separate Components

Step one utiltizes an MB8683x processor along with a microcontroller and other discrete components. This solution provides the lowest customer risk. The microncontroller provides a wide range of peripheral combinations.

Solution II: MB8683x and Custom ASIC

The Fujitsu Systems Solutions team works with the customer on ASIC development. The ASIC integrates the microcontroller and discrete component functions from Solution I. Overall board size is reduced.

Solution III: System-On-a-Chip

This solution incorporates the MB8683x core and ASIC functions from Solution II in one device. This solution offers the best customer price position on the market and is ideal for commodity products. The Fujitsu Systems Solutions team works with the customer during development.

Summary

The MB8683x family of embedded microprocessors are designed to function with ASIC devices to provide a high-performance, high integration solution at low cost. The MB8683x family of microprocessors offers the ability to differentiate a given product line without having to change the ASIC each time.

There are three product offerings that scale in both cost and performance. The MB86833 is the lowest-cost embedded controller on the market today.

The programmable nature of the MB8683x allows for maximum flexibility during the design phase. The on-chip interrupt controller, DRAM controller, and chip-select logic simplifies system design by eliminating the need for these functions externally.

SPARClite—The ASIC Companion

The MB8683x supports multiple DRAM types such as Fast page Mode (FPM) and Extended Data Out (EDO). The interrupt controller supports the selection of both level and priority on a per-interrupt basis, allowing the MB8683x to interface to almost any peripheral.

The memory controller supports features such as programmable bus widths, burst mode, EDO and FPM DRAM, programmable chip selects, and non-cacheable memory support.

The MB8683x family offers a scalable amount of on-chip cache, from the 1k-data/1k- instruction MB86833 to the 8k-data/8k-instruction MB86832. The MB86831 and MB86832 devices incorporate a 2-way set associative cache architecture for both data and instruction and all devices support cache locking.

An on-chip hardware multiplier accelerates many image-processing applications. Most instructions execute in just one clock cycle.

A clock multiplier allows the core to operate at much higher frequencies than the system bus, at up to a 5:1 ratio.

The MB8683x provides a power management mode which is critical in battery-powered applications. The MB86833 consumes only 10 mA when in power-down mode.

The MB86832 contains a built-in emulator bus and provides a 10-pin interface that supports In-Circuit Emulators (ICE) up to 40 MHz.

The MB86831 and the MB86832 are pin compatible, allowing for a seamless upgrade path from one component to another.

The Fujitsu systems solution team provides a strong infrastructure that can assist the customer in migrating from one product solution to another.

References

MB86831 SPARClite Embedded Processor User's Manual

MB86832 SPARClite Embedded Processor User's Manual

MB86833 SPARClite Embedded Processor User's Manual

SPARClite Series 32-bit RISC Embedded Processor Datasheet, MB86831

SPARClite Series 32-bit RISC Embedded Processor Datasheet, MB86832

SPARClite Series 32-bit RISC Embedded Processor Datasheet, MB86833

MB86831 Evaluation Kit Hardware Guide

Image Processing on SPARClite, Embedded Systems Conference Presentation, Roger Brooks, Ph.D.

Printed in U.S.A.

EC-AP-20702-7/98

FUJITSU MICROELECTRONICS, INC.

Corporate Headquarters
3545 North First Street, San Jose, California 95134-1804
Tel: (800) 866-8608 Fax: (408) 922-9179
E-mail: fmicrc@fmi.fujitsu.com Internet: http://www.fujitsumicro.com