MC68851

Technical Summary

32-Bit Paged Memory Management Unit

The MC68851 is a high-performance paged memory management unit (PMMU) designed to efficiently support a demand-paged virtual memory environment with the MC68020 32-bit microprocessor. Implemented using VLSI technology and Motorola's advanced HCMOS fabrication process, the MC68851 is optimized to perform very fast logical-to-physical address translations, to provide a comprehensive access control and protection mechanism, and to provide extensive support for paged virtual systems. The following features are included on the MC68851:

- Fast Logical-to-Physical Address Translation
- Hierarchical Protection Mechanism with up to Eight Levels of Protection
- Full 32-Bit Logical and Physical Addresses with 4-Bit Function Code
- Wide Selection of Page Sizes from 256 Bytes to 32K Bytes
- Fully Associative, 64-Entry, On-Chip Address Translation Cache
- Automatic Update of the On-Chip Translation Cache from External Translation Tables
- Multiple Tasks Supported Simultaneously for Fast Task Switching
- M68000 Family Coprocessor Interface
- MC68020 Instruction Set Extensions
- Instruction Breakpoints for Software Debug and Program Control
- Support for Logical and/or Physical Data Cache
- Support for Multiple Logical and/or Physical Bus Masters

SYSTEM FUNCTION AND CONFIGURATION

The primary system functions of the MC68851 are to provide logical-to-physical address translation, to monitor and enforce the protection/privilege mechanism, and to support the breakpoint operations used for system debug. The MC68851 also supports the M68000 Family coprocessor interface to simplify processor/coprocessor communication.

In a simple microprocessor-based system, the central processor unit (CPU) is connected directly to memory. No memory mapping or protection functions are provided, and the addresses generated by the CPU identify the physical locations to be accessed. Any location in the address space that does not contain a memory device cannot be used by the CPU. This type of system is unsuitable for execution of multiple, concurrent tasks since no mechanism exists to protect the memory of one task from corruption by another task. It is unsuitable for hosting virtual systems that allow uniform use of a virtual address space that is larger than the physical address space represented by the memory devices, and it is also unsuitable for providing a separate unique address space for each task in the system.

The MC68851 is designed to provide the mapping and protection facilities needed to construct a multitasking, demand-paged virtual system. To build such a system, the address bus is divided into two sections separated by the MC68851 (see Figure 1). The 'logical' address is output by the processor and is monitored by the MC68851 on its logical address inputs. The MC68851 performs translation and privilege checking on the logical address and, if valid, outputs the translated 'physical' value on the physical address bus where it is used to access memory or other physical devices. Using this configuration, all accesses to physical devices are controlled by the MC68851; tasks can be prevented from accessing the resources owned by other tasks. Also, under control of an operating system with virtual capabilities, the logical-to-physical mapping functions of the MC68851 allow tasks to utilize the entire address space of the CPU without knowledge of the physical attributes of the system.

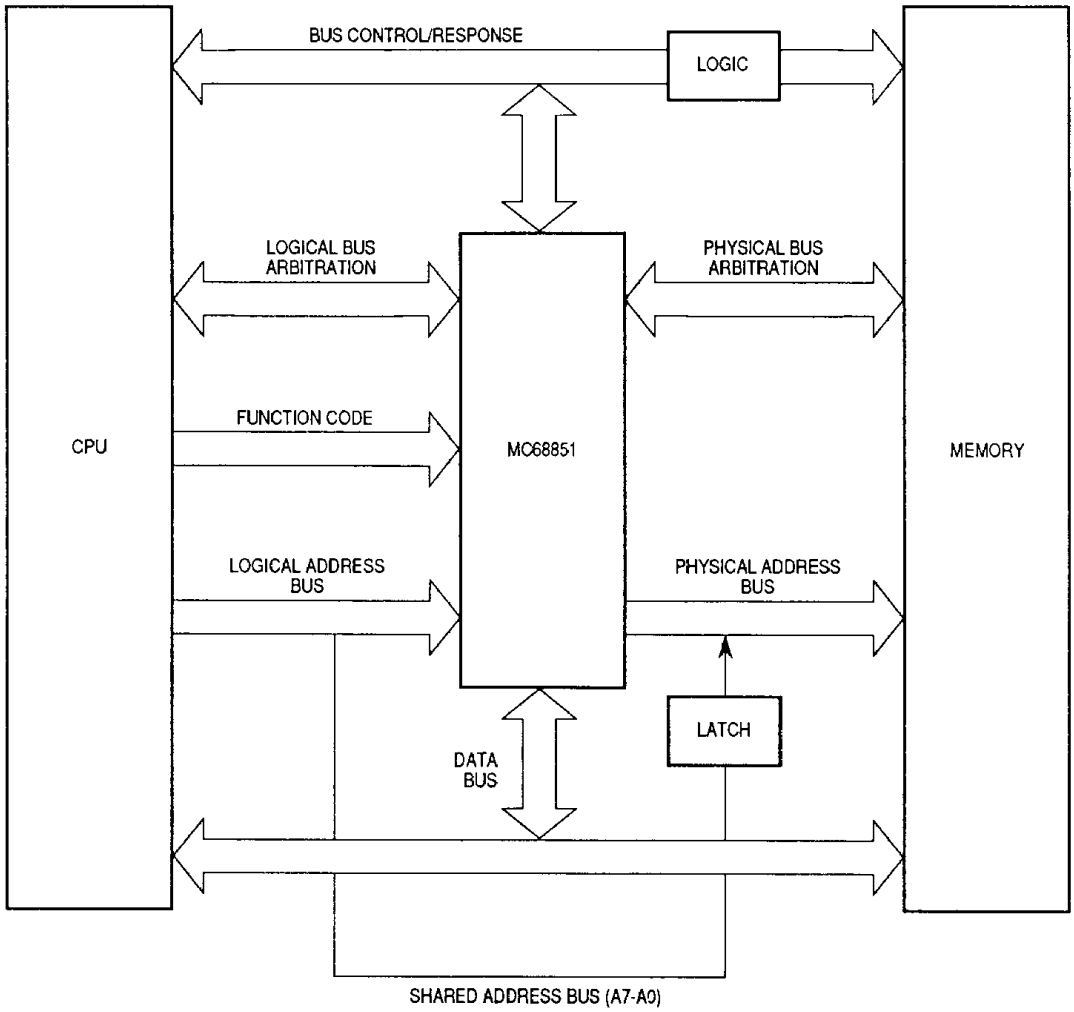


Figure 1. System Block Diagram

ADDRESS TRANSLATION

The address translation facility of the MC68851 is a comprehensive mechanism that provides logical-to-physical mapping of up to a 4-Gbyte logical address space with no software assistance from the CPU. The address translation mechanism is fully implemented in hardware to minimize any penalty in system performance for the mapping functions. The address translation mechanism provides full logical-to-physical mapping in less than one clock cycle for a very high percentage of all bus cycles. The functional timing for these translations is shown in Figure 2. Physical address strobe is asserted one clock after the assertion of logical address strobe with translation completed and access rights checked for the valid physical address.

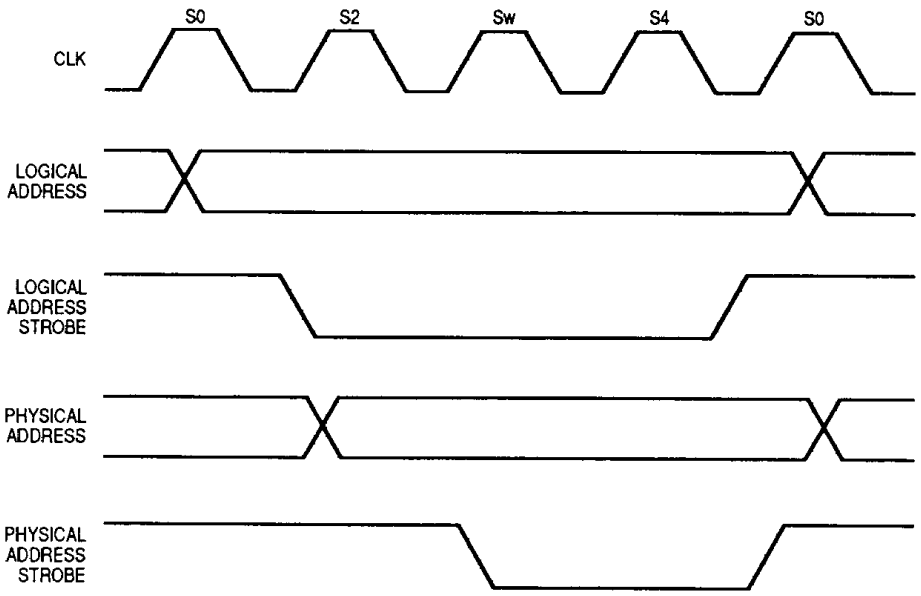
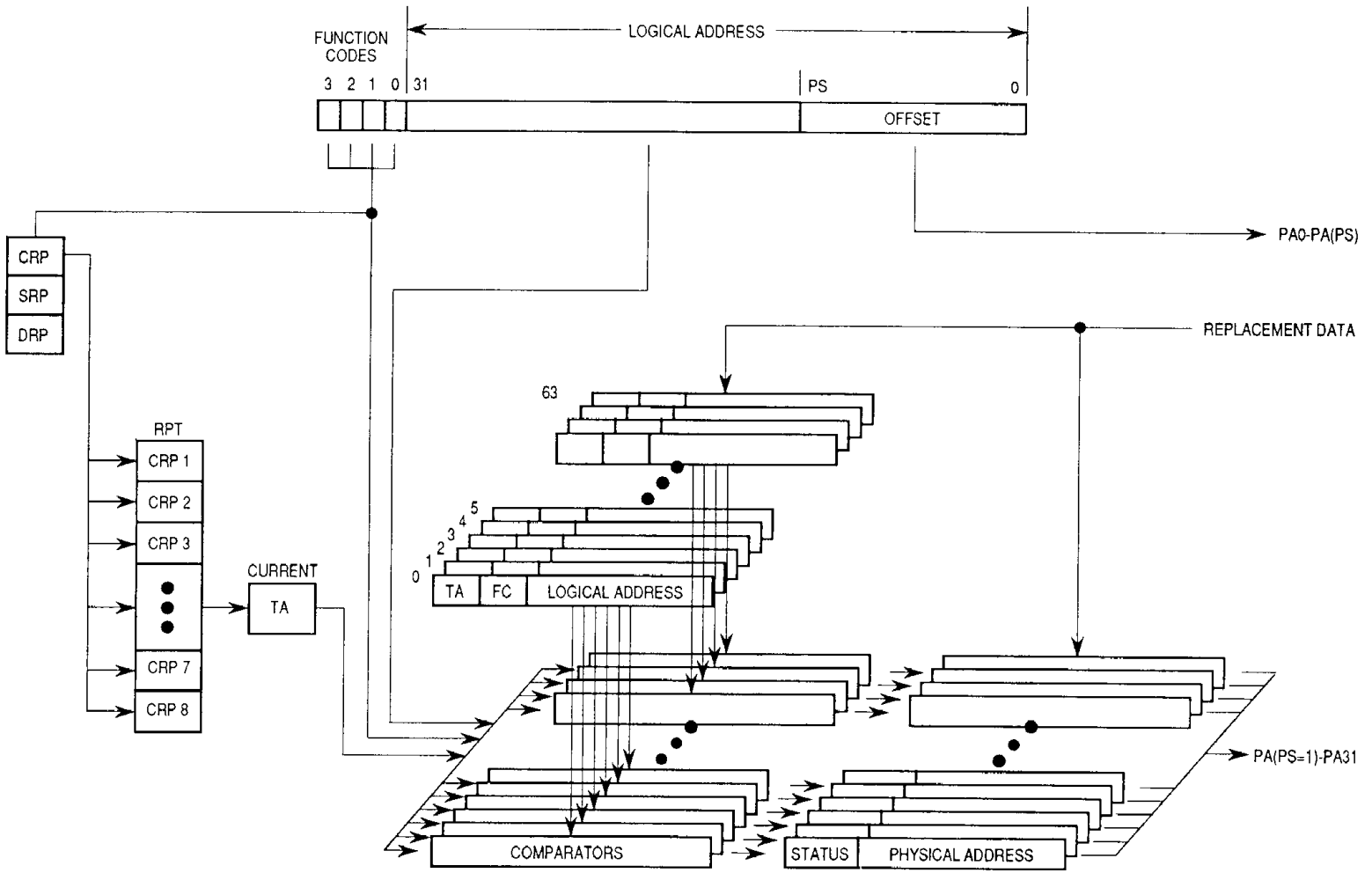


Figure 2. Address Translation Functional Timing

5

To perform the translation functions shown in Figure 2, the MC68851 contains a high-speed memory that stores recently used logical-to-physical address translations. This memory, the address translation cache (ATC), is a 64-entry, fully associative array containing logical addresses and their corresponding physical translations (see Figure 3). When a bus cycle is initiated by a logical master, the logical address and function code is input to the ATC where it is simultaneously compared against all current entries. If one of the ATC entries matches (a 'hit'), the ATC drives the stored physical address onto the physical address bus. If the MC68851 detects no exceptional conditions (e.g., write violation), it then asserts the physical address strobe ($\overline{PAS}$).



PAGE SIZE = 2^{PS} = 256/512/1K/2K/4K/8K/16K/32K BYTES

STATUS INCLUDES: ATC ENTRY LOCKED OR BUS ERRORED, CACHE INHIBIT, MODULE DESCRIPTOR GATE, MODIFIED, AND WRITE PROTECT BITS

Figure 3. ATC with Task Alias, RPT, and Caches

In addition to the address mappings, each entry in the ATC also contains bits that describe the protection information for that mapping (e.g., read-only), a data cache inhibit indicator (used, for instance, when memory-mapped I/O registers must not be cached), a lock-entry flag (used to freeze individual ATC translation entries), and history information used by the MC68851 (used by the ATC replacement algorithm to keep active translations in the ATC).

To improve utilization of the MC68851 address translation cache in a multitasking environment, translation descriptors for multiple tasks can simultaneously reside in the ATC. To control this, the logical portion of each ATC entry has three additional bits (a 'task alias'), that are included in the compare operation to determine if a cache hit has occurred. The task alias, which identifies one of eight tasks that may have translation descriptors simultaneously resident in the ATC, is used as an extension to the logical address during the cache comparison operation.

The task alias works in conjunction with the MC68851 root pointers and the root pointer caching function of the root pointer table (RPT). When the MC68020 starts a new task, the operating system loads the MC68851 CPU root pointer (CRP) register with the pointer needed for translations of the new task. The CRP then contains the address, in physical memory, of the root of the translation table for the currently executing task. The RPT is a table of eight recently used CRPs maintained on-chip by the MC68851. Each of the eight entries in the RPT has a unique 3-bit task alias associated with it. Since these three bits are included in each ATC entry, eight tasks can be distinguished among all ATC entries.

When the operating system initiates a new task or restarts a suspended one, it writes a value to the CRP register, identifying the location of the translation table for that task. When this value is written, it is compared against all entries currently in the RPT. This process is fast because the RPT is also a cache. If no RPT match is found, then a new entry is made in the RPT, and the task alias associated with that entry is assigned to the current task. When the RPT is full, the new entry overwrites the oldest RPT entry, and this task alias becomes associated with the new task. In this case, the MC68851 automatically flushes any entries in the ATC that are currently identified with this task alias since they are associated with the old task.

If the value loaded into the CRP register matches an entry in the RPT, then the MC68851 already has a task alias assigned to identify those ATC entries that belong to the new task; therefore, no ATC entries are flushed since they can be reused for address translations of the new task. The eight entries of the RPT and the 64 entries of the ATC are well balanced. When a task is restarted, the

ATC usually holds some of its entries and the task can begin execution immediately. Without task aliasing, the new task would be delayed while the needed translation descriptors are loaded into the ATC from the translation table in memory.

In addition to the CRP, the MC68851 maintains exclusive root pointers for the supervisor (SRP) and for DMA-type devices and coprocessors (DRP) requiring logical-to-physical address translations.

ADDRESS TRANSLATION TABLES

When a logical bus master (e.g., the processor) initiates a cycle that does not have a corresponding translation resident in the ATC, the MC68851 performs bus operations to load the mapping for that cycle from the translation tables in memory pointed to by the relevant root pointer. To perform this search operation, the MC68851 simultaneoulsy aborts the logical bus cycle, signals the master to retry the operation, and requests mastership of the logical bus. Upon indication that the logical bus is free, the MC68851 completes the bus arbitration sequence, assumes mastership of the bus, and begins to search the translation tables pointed to by the relevant root pointer to locate the needed translation descriptor that describes the page accessed by this logical address. After loading the required translation descriptor, the MC68851 returns control of the bus to the logical master to retry the previous bus cycle, which can be checked for access rights and properly translated by the MC68851.

The operation of searching the translation tables and reloading the ATC is called a table search. The MC68851 automatically searches the translation tables when a translation misses in the ATC, utilizing only hardware without any software assistance from the operating system. Using hardware, the MC68851 has significantly minimized the table search overhead when compared to previous memory management schemes that required software assistance.

The translation tables supported by the MC68851 have a tree structure. The root of a translation table tree is pointed to by one of the three root pointer registers: CRP, SRP, or DRP. Table entries at the higher levels of the tree (pointer tables) contain pointers to other tables. Entries at the leaf level (page tables) contain page descriptors. All addresses contained in the translation table entries are physical addresses. The three root pointers allow the MC68851 to manage three simultaneous activities: the CRP points at the translation table tree for the currently executing task, the SRP points to the operating system's translation table, and the DRP manages the space of an alternate bus master, which could be a DMA controller.

Figure 4 illustrates the structure of the MC68851 translation tables. Several determinants of the detailed table structure are software selectable. The first level of lookup in the table normally uses the function codes as an index, but this feature can be suppressed. The function codes are control signals output by the MC68020 that distinguish memory accesses as supervisor/user and program/data space accesses. The function codes can be used by the MC68851 to separate the address map into address spaces that protect any supervisor space from being corrupted by a faulty user program and/or to prevent a data access from corrupting a program space. The logical address can be limited from 17 to 32 bits (inclusive) to control the necessary size of the corresponding translation table. The number of levels in the table indexed by the logical address can be set from one to four, and up to 15 logical address bits can be used as an index at each level.

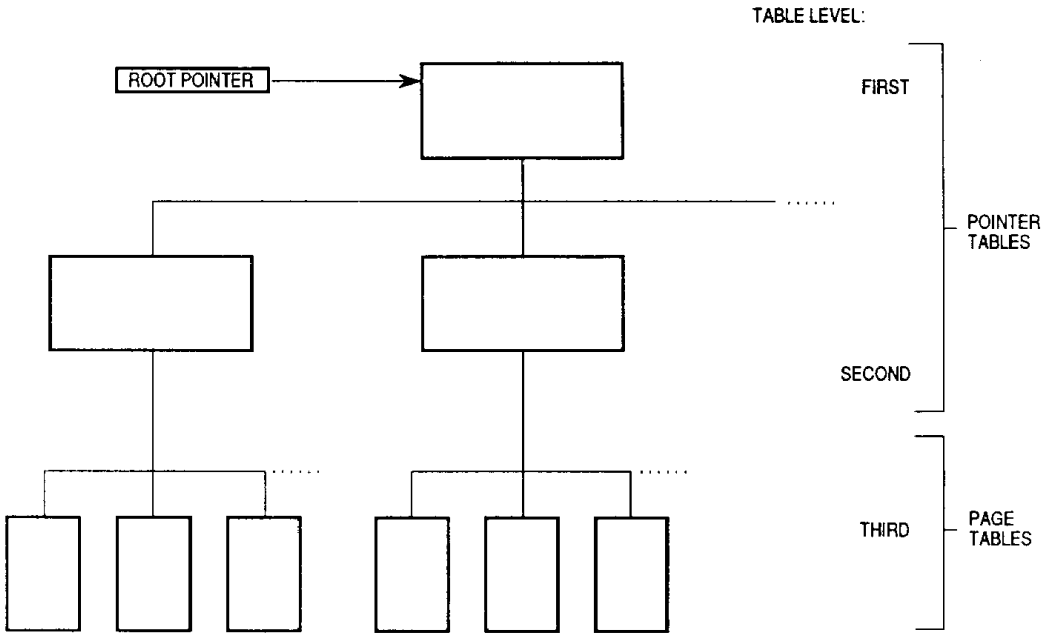


Figure 4. Translation Table Structure

PROTECTION MECHANISM

The MC68851 supports a comprehensive protection mechanism that facilitates implementation of fully protected systems. In addition to the option of enforcing the distinction of user and supervisor modes normally found in an M68000 system, the MC68851 also supports the access level mechanism that provides finer granularity of protection within the user address spaces.

The access level mechanism subdivides the logical address spaces of user mode operations into one, two, four, or eight level(s) of privilege. Routines operating at different access levels can have different privileges to memory, and a facility is provided to closely control changes in access level.

The access level for a bus cycle is encoded in the highest order (zero, one, two, or three) bits of the logical address generated by the CPU. The access level mechanism, when enabled, compares this value against the current access level as specified in the current access level (CAL) register. The current access level specifies the highest privilege level that a task may assume at that time. If the privilege level presented by the bus cycle is more privileged than the current level allowed, then the cycle is requesting a privilege in excess of its rights and is aborted by the MC68851.

In the MC68851 protection scheme, the privilege associated with a task is specified by its access level. Smaller values for access levels represent higher privilege levels. In a system using eight access levels, level zero is the highest privilege in the hierarchy, and level seven is the lowest.

To access code and/or data requiring a higher level of privilege than that possessed by the current task, the MC68851 supports the MC68020 module call (CALLM) and return (RTM) instructions that allow a routine to transfer execution control to a module operating at the same or higher level of privilege and to return from that module after completion of the module function. When the MC68020 executes a CALLM instruction that requests an increase in access level, the MC68020 automatically communicates with the MC68851 access level mechanism via access level control CPU space cycles to determine if the requested change is valid. The MC68851 checks the request against a module descriptor for that operation and indicates the validity of that request to the MC68020. The RTM instruction operates similarly except that control is always passed from a task to a task of the same or lesser privilege.

BREAKPOINTS

The MC68851 provides a breakpoint acknowledge facility to support the MC68020 and other processors with an on-chip cache. When the MC68020 encounters a breakpoint instruction, it executes a breakpoint acknowledge bus cycle by reading from a predetermined address in the CPU address space. The MC68851 decodes this address and responds either by providing a replacement opcode for the breakpoint opcode and completing the bus cycle normally (asserting the data transfer and size acknowledge outputs) or by terminating the bus cycle with an exception (asserting bus error to initiate illegal instruction exception processing). The MC68851 can be programmed to signal the illegal instruction exception on every breakpoint or to provide the replacement opcode n times ($1 \leq n \leq 255$) before signaling the exception. With eight sets of breakpoint registers, the MC68851 simultaneously supports eight breakpoints.

Debugging an MC68020-based system using an on-chip cache is simplified with the MC68851 breakpoint support. Programmers typically use a debug monitor when debugging programs, which can place up to eight breakpoints in the program's code when using MC68851 breakpoint support. The debug monitor replaces each target instruction with a breakpoint instruction, passing the target opcode to the MC68851 with a skip count. During execution of the program, the MC68020 encounters the various breakpoint instructions. The MC68851 automatically provides the correct target opcode to the MC68020 for each execution of the corresponding breakpoint instruction and counts down the skip count to zero. The MC68020 executes the substituted instruction as if it were in program memory, and the program continues with only a small overhead to retrieve the substituted opcode. Once the skip count is exhausted during breakpoint processing, the debug monitor regains control through the illegal instruction exception handler.

COPROCESSOR INTERFACE

The M68000 Family coprocessor interface is an integral part of the design of the MC68020 microprocessor, the MC68881 floating-point coprocessor, and the MC68851 PMMU. The coprocessor interface allows execution of special-purpose instructions that are logical extensions to the microprocessor. Each coprocessor (e.g., MC68851 or MC68881) has an instruction set that reflects its special function. These instructions can be executed by placing the instruction opcode and parameters in the MC68020 instruction stream. The MC68020 decodes the coprocessor instruction and performs bus communication with the coprocessor registers, specifying the nature of the action to be taken. Both the MC68020 and the coprocessor execute portions of the instruction, depending on which processor is best suited to handle a particular task.

The interchange of information and division of responsibility between the processor and the coprocessor are controlled by the coprocessor interface, which is transparent to the user/programmer. The addition of a coprocessor unit to an MC68020 system is a logical extension that simply complements the instruction set of the processor.

The coprocessor interface is designed to be flexible, functional, and expandable. The interface is intended to support current M68000 Family devices and future extensions to the Motorola coprocessor family as well as user-defined coprocessors for single or multiple coprocessor systems.

MC68851 INSTRUCTIONS

The MC68851 implements an extension to the M68000 Family instruction set using the coprocessor interface. These instructions provide control functions for 1) loading and storing of PMMU registers, 2) testing access rights and conditionals based on the results of this test, and 3) PMMU control functions. The MC68851 instruction set extension to the MC68020 instruction set provides a programming model that allows the registers and functions implemented by the MC68851 to appear to the programmer as available on-chip with the MC68020. The programming model is shown in Figure 5.

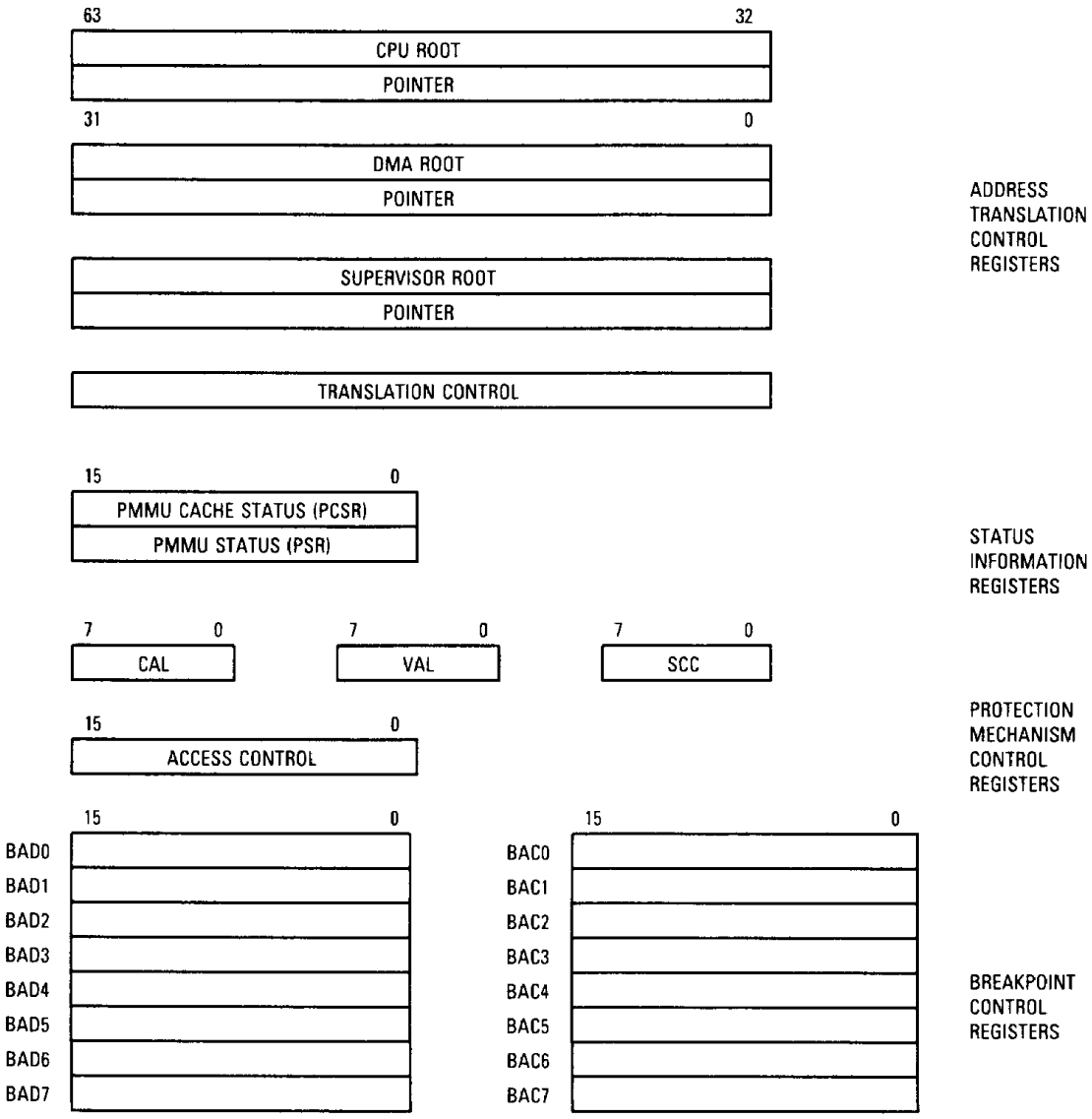


Figure 5. MC68851 Programming Model

The instruction set extensions for the MC68851 are as follows:

PMOVE Moves data to/from MC68851 register.

PVALID Compares access rights of a logical address against the current access level and traps if address requires a higher privilege than allowed. This instruction can be used by a routine to verify that an address passed to it by a calling routine is a valid address.

PTESTR Searches translation tables and loads the status and access rights information of a logical address used for a read cycle into the MC68851 status register. This instruction allows the operating system to quickly determine the cause of faults generated by a read cycle from a particular logical address.

PTESTW Searches translation tables and loads the status and access rights information of a logical address used for a write cycle into the MC68851 status register. This instruction allows the operating system to quickly determine the cause of faults generated by a write cycle to a particular logical address.

PLOADR Searches translation tables and loads the ATC with a translation for the specified logical address used for a read cycle. The history information in the external translation tables is updated to reflect that the physical page corresponding to the logical address has been used.

PLOADW Searches translation tables and loads the ATC with a translation for the specified logical address used for a write cycle. The history information in the external translation tables is updated to reflect that the physical page corresponding to the logical address has been modified.

PFLUSH Flushes translation cache entries by logical address, function code, or function code and effective address. The PFLUSH instructions allow the operating system to easily remove entries from the ATC after making modifications to the external translation tables.

PFLUSHA Flushes all entries from the translation cache.

PFLUSHR Flushes RPT and translation cache entries by root pointer.

PFLUSHS Flushes entries from the ATC by logical address and/or function code, including globally shared entries.

PSAVE Saves the internal state of the MC68851 to support fast context switching and MC68020 virtual memory/virtual machine capabilities.

PRESTORE Restores the internal state of the MC68851 stored by the PSAVE instruction.

PBcc Branches conditionally on MC68851 condition. The conditional instructions provide the operating system with a means to control program flow by MC68851 conditions.

PDBcc Tests MC68851 condition, decrements a CPU register, and branches.

PScc	Sets operand according to MC68851 condition.
TRAPcc	Traps on MC68851 condition.

SIGNAL DESCRIPTION

The following paragraphs provide a brief description of the input and output signals of the MC68851 PMMU. The signals are functionally grouped as shown in Figure 6.

NOTE

Assertion and **negation** are used exclusively to avoid confusion when dealing with a mixture of active-low and active-high signals. **Assert** or **assertion** is used to indicate that a signal is active or true, independent of whether that level is represented by a high or low voltage. **Negate** or **negation** is used to indicate that a signal is inactive or false.

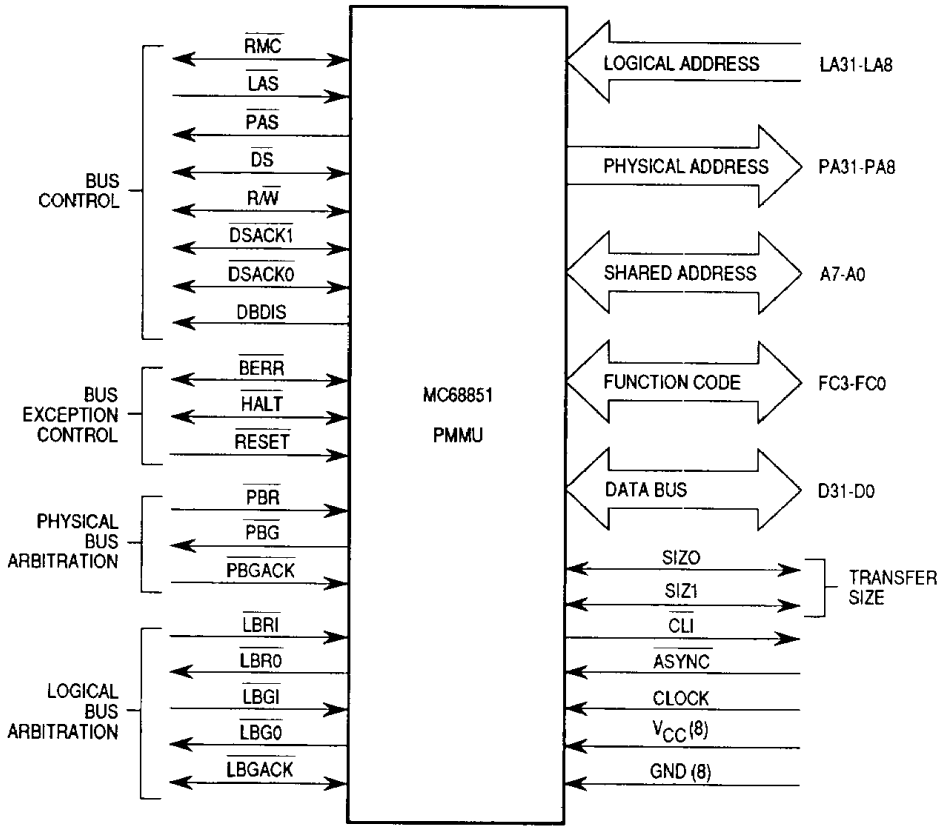


Figure 6. Functional Signal Groups

LOGICAL ADDRESS BUS (LA8–LA31)

The MC68851 accepts a logical address for translation or for internal operations on these inputs. The logical address bus should be connected to the address outputs of all logical bus masters.

If the logical address is less than 32 bits as determined by the translation control register, the unused bits are ignored and should be tied to a constant voltage level (either V_{CC} or ground).

PHYSICAL ADDRESS BUS (PA8–PA31)

These three-state outputs provide the physical address for address translations and MC68851-initiated bus operations.

SHARED ADDRESS BUS (A0–A7)

The use of these three-state, bidirectional signals is shared between the functions of the logical and physical buses. When the MC68851 is performing address translations, these signals are input for the MC68851 to monitor the entire logical address in the event that a CPU space cycle accesses one of its registers. When the MC68851 is the bus master, these pins output the low-order eight bits of the physical address. With the inclusion of A0–A7, both the logical and physical buses have a 32-bit (4-Gbyte) linear addressing range.

FUNCTION CODE (FC0–FC3)

These three-state, bidirectional signals indicate the address space of the current bus cycle. When the MC68851 is performing translations, these signals provide the address space being accessed by the current logical bus master. The MC68851 uses the function code associated with a bus cycle as an extension to the logical address when creating entries in the address translation cache. The function code may also be used as an index from the root pointer in the first level of a translation table search.

The 4-bit function code consists of the three function code outputs of the M68000 Family processor and a fourth bit that indicates that a DMA access is in progress.

When the MC68851 is bus master, it drives the function code pins as outputs with a constant value of FC3–C0 = $5, indicating the supervisor data space.

DATA BUS (D0–D31)

These three-state, bidirectional signals provide the general-purpose data path between the MC68851 and other devices. This bus can be dynamically sized through use of the $\overline{\text{DSACKx}}$ signals, transferring 8, 16, 24, or 32 bits of information during a bus cycle. The most significant byte of the data bus is D24–D31.

In systems that do not use the MC68020 (or any other 32-bit CPU) as the main processor, the width of the data bus used to communicate between the processor and the MC68851 may be fixed at 16 or 8 bits. In such systems, the dynamic bus sizing mechanism still functions, but the maximum amount of data transferred in a single cycle is limited to the bus size. In either case, the processor data bus is aligned towards the high-order portion of the MC68851 data bus — that is, an 8-bit master is connected to D24–D31, and a 16-bit master is connected to D16–D31.

When the $\overline{\text{RESET}}$ signal is asserted, the MC68851 inputs configuration information from the least significant byte of the data bus (D0–D7). This information determines the bus size for coprocessor operations, sets the decision time for determining whether or not an ATC hit has occurred, determines whether the $\overline{\text{CLI}}$ signal is asserted for all MC68851-initiated bus operations, and sets the timing for $\overline{\text{PAS}}$ and $\overline{\text{DS}}$ assertion during table searches.

TRANSFER SIZE (SIZ0, SIZ1)

These three-state, bidirectional signals are used in conjunction with the dynamic bus sizing capabilities of the MC68851. When the MC68851 is the bus master, the SIZx signals are outputs; when the MC68851 is accessed as a slave, these signals are inputs. Otherwise, the SIZx signals are ignored. Regardless of the state (input or output) of these signals, they indicate the number of bytes remaining to be transferred during the current operand cycle. An operand cycle is a bus cycle or sequence of bus cycles required to transfer a complete operand.

BUS CONTROL SIGNALS

The logical and physical bus control signals are described in the following paragraphs.

Read-Modify-Write ($\overline{\text{RMC}}$)

This three-state, bidirectional signal is used to indicate that the bus cycle in progress is an indivisible read-modify-write cycle. This signal is asserted for

the duration of the read-modify-write sequence and should be used as a bus lock to ensure integrity of cycle operation.

When the MC68851 is translating addresses, the assertion of $\overline{RMC}$ by the logical bus master indicates that the master is performing a read-modify-write cycle and that a write operation to the same operand is likely to follow. When $\overline{RMC}$ is asserted during a read cycle, the MC68851 performs access and privilege checking for that cycle as if it were a write cycle to prevent the operation from aborting after having partially completed the write portion of the cycle. In addition, physical bus arbitration is suspended once the physical bus cycle for the address translation is initiated.

When the MC68851 is bus master, $\overline{RMC}$ can be asserted to indicate that the operation in progress should not be interrupted by other bus traffic; hence, all arbitration for the physical bus is suspended by the MC68851 when this signal is asserted.

Logical Address Strobe ($\overline{LAS}$)

The assertion of this input indicates that the logical bus master has driven the logical address bus, function code, and R/$\overline{W}$ valid. When the MC68851 is being accessed as a slave, the assertion of $\overline{LAS}$ also indicates that the SIZx signals are driven valid.

Physical Address Strobe ($\overline{PAS}$)

This three-state output is asserted when the MC68851 has driven a valid address on the physical address bus. When the MC68851 is master of the logical bus, the assertion of $\overline{PAS}$ also indicates that the function code, R/$\overline{W}$, and SIZx signals are valid.

Data Strobe ($\overline{DS}$)

This bidirectional, three-state signal is used to control the flow of information on the data bus.

When the MC68851 is selected by the CPU, $\overline{DS}$ is an input indicating that the MC68851 should drive the data bus on a read cycle or that the CPU has placed valid data on the bus during a write cycle.

When the MC68851 is the bus master, $\overline{DS}$ indicates that the slave device should drive the data bus during a read cycle or that the MC68851 has placed valid data on the bus during a write cycle.

The data strobe is ignored for the purposes of address translation.

Read/Write (R/$\overline{W}$)

This bidirectional, three-state signal is used to indicate the direction of transfer for a bus cycle. When the MC68851 is translating addresses, the state of the R/$\overline{W}$ signal is input to support write-protection checking.

When the MC68851 register set is accessed by the CPU for an operation, the R/$\overline{W}$ output by the CPU determines the direction of data transfer. If the signal is asserted (low), the MC68851 latches data from the data bus at the termination of the cycle. If the signal is negated (high), the MC68851 outputs data on the data bus and signals that the transfer is complete.

When the MC68851 is bus master, the R/$\overline{W}$ signal is driven as an output. A high indicates a read from an external device; a low indicates a write to an external device.

Data Transfer and Size Acknowledge ($\overline{DSACK0}$, $\overline{DSACK1}$)

These bidirectional, three-state signals, whether used as inputs or outputs, are normally used to terminate a bus cycle and to indicate the port size of the responding device.

When the MC68851 register set is accessed by the CPU, the $\overline{DSACKx}$ signals are output to indicate that valid data has been or will be placed on the data bus for a read cycle or that data has been accepted from the data bus for a write cycle. Note that the relationship between $\overline{DSACKx}$ and data is dependent on the operating mode of the MC68851. When operating in the synchronous mode, the MC68851 drives the data bus on the same clock edge that $\overline{DSACKx}$ is asserted. Otherwise, the MC68851 drives the data bus a minimum of one clock period before asserting the $\overline{DSACKx}$ signals.

The $\overline{DSACKx}$ signals are monitored as inputs when the MC68851 arbitrates for the logical bus. After receiving a bus grant from the CPU, the MC68851 waits until $\overline{LBGACK}$, $\overline{LAS}$, and both $\overline{DSACKx}$ signals are negated before asserting $\overline{LBGACK}$ to ensure that the previous slave device has released connection from the bus.

When the MC68851 is executing bus cycles as the physical bus master, the $\overline{\text{DSACKx}}$ signals are inputs to indicate the completion of a data transfer and the port size of the external device being accessed. During a read cycle, when the MC68851 recognizes $\overline{\text{DSACKx}}$, it latches the data and then terminates the bus cycle; during a write cycle, when the MC68851 recognizes $\overline{\text{DSACKx}}$, the bus cycle is terminated. When operating as bus master, the MC68851 synchronizes the $\overline{\text{DSACKx}}$ inputs and allows skew between the two inputs of up to one-quarter of a clock.

Data Buffer Disable (DBDIS)

This active-high output provides an enable to external data buffers connected to the MC68851 data bus.

When the logical bus master reads the contents of one of the MC68851 registers, the MC68851 drives the data bus with the required operand. Typical systems directly connect the MC68851 data bus with that of the main processor, and the combined bus is buffered before being routed to a large number of physical address space devices. To avoid contention, the buffers between the MC68851/CPU bus and the bus driving the physical memory must be disabled when the MC68851 drives the bus. The MC68851 provides the control necessary to perform this function with the DBDIS signal.

In addition, DBDIS performs a function similar to the function of the MC68020 $\overline{\text{DBEN}}$ signal. DBDIS can be used to control data bus transceivers to avoid contention between the transceivers and the MC68851 data bus drivers during table search operations.

Finally, DBDIS is driven during reset to isolate the MC68851 data bus while configuration information is being input.

BUS EXCEPTION CONTROL SIGNALS

The following paragraphs describe the bus exception control signals for the MC68851.

Reset ($\overline{\text{RESET}}$)

Assertion of this input signals the MC68851 to disable the address translation mechanism, clear all breakpoints, set the internal state to idle, and input configuration information from the data bus.

Halt ($\overline{\text{HALT}}$)

$\overline{\text{HALT}}$ is a bidirectional, three-state signal.

When the MC68851 is the logical bus master, $\overline{\text{HALT}}$ is an input, and assertion of $\overline{\text{HALT}}$ stops all MC68851 bus activity at the completion of the current bus cycle. When the MC68851 has been halted using this input, all control signals, with the exception of bus arbitration outputs, are placed in their inactive states, and the physical address bus remains driven with the value used during the previous bus cycle. Bus arbitration functions normally when the MC68851 is halted.

When the MC68851 is translating addresses, $\overline{\text{HALT}}$ is used as an output in conjunction with $\overline{\text{BERR}}$ and/or $\overline{\text{LBRO}}$ to signal the current logical bus master to perform either a relinquish and retry or a relinquish operation. During address translation, the assertion of $\overline{\text{HALT}}$ by an external device does not effect translation operations of the MC68851.

5

Bus Error ($\overline{\text{BERR}}$)

This bidirectional, three-state signal is used to indicate that a bus cycle should be terminated due to abnormal conditions.

When the MC68851 is bus master, $\overline{\text{BERR}}$ is an input, and assertion of $\overline{\text{BERR}}$ by an external device signals a problem with the curent bus cycle. These problems may be the result of 1) nonresponding devices or 2) various other application-dependent errors (e.g., parity errors).

When the MC68851 is translating addresses, $\overline{\text{BERR}}$ is used as an output to the logical bus master. $\overline{\text{BERR}}$ is asserted by the MC68851 for the following conditions:

1. The BERR bit is set in the matched ATC entry.
2. A write or read-modify-write cycle is attempted to a write-protected page.
3. An instruction breakpoint is detected, and the associated count register is zero or is disabled.
4. As a portion of the relinquish and retry operation if:
   a. the required address mapping is not resident in the ATC,
   b. a write operation occurs to a previously unmodified page,
   c. a read from the response coprocessor interface register (CIR) causes a suspended PLOAD or PTEST instruction to be restarted,
   d. a module call operation references a descriptor that does not have a corresponding entry in the ATC.

5. Read-modify-write cycle is attempted and a corresponding descriptor with appropriate status is not resident in the ATC.
6. The access level protection mechanism detects an access.

$\overline{BERR}$ interacts with $\overline{HALT}$ to determine if the current bus cycle should be retried or aborted.

CACHE LOAD INHIBIT ($\overline{CLI}$)

During address translation, this three-state output is asserted by the MC68851 if the matched address translation cache entry has the cache inhibit (CI) bit set. Assertion of this output signals to external caches that the data associated with the current bus cycle is noncachable.

To maintain the distinction between CPU space and other address spaces (for example, supervisor program, etc.), the MC68851 does not assert $\overline{PAS}$ for CPU space cycles. $\overline{CLI}$ is used to generate a CPU space address strobe during CPU space cycles that do not access the MC68851. $\overline{CLI}$ is asserted on the falling edge of the clock and external qualification of $\overline{CLI}$ with $\overline{LAS}$ and a CPU space indicator provides a CPU space address strobe. CPU space cycles that access the MC68851 registers are decoded internally and generate no physical bus activity. Note that, if the MC68851 is not master of the physical bus, $\overline{CLI}$ is not asserted until ownership of the physical bus is returned to the MC68851.

When the MC68851 is performing table search operations, it continuously asserts $\overline{CLI}$ to prevent caching of translation table information. This function can be suppressed during reset configuration if desired.

ASYNCHRONOUS CONTROL ($\overline{ASYNC}$)

When a logical bus master does not present logical bus control signals with the exact timing specifications of the MC68020, this input must be driven, with appropriate setup and hold times, to inform the MC68851 that input synchronization must occur.

Operating in a synchronous mode, the MC68851 utilizes known signal relationships to perform faster translations. If the logical bus master does not present signals conforming to these relationships (different control strobe timings and/or different operating frequency), it must assert $\overline{ASYNC}$ prior to initiating bus activity.

CLOCK (CLK)

The MC68851 clock input is a TTL-compatible signal that is internally buffered to develop internal clocks for the memory management unit. The clock must conform to minimum and maximum period and pulse width specifications and must be of a constant frequency.

Note that the MC68851 and the logical bus master may operate at different clock frequencies.

PHYSICAL BUS ARBITRATION

The following paragraphs describes the three-wire physical bus arbitration circuitry of the MC68851 used to determine which device in a system is the master of the physical bus. Because the MC68851 is the default master of the physical bus, any other devices requiring access to the bus must arbitrate for mastership.

Physical Bus Request ($\overline{PBR}$)

This input is the wire-OR of the bus request signals from all potential physical bus masters and indicates that some device other than the MC68851 requires mastership of the physical bus.

Physical Bus Grant ($\overline{PBG}$)

This output signal indicates to potential bus masters that the MC68851 will release ownership of the physical bus when the current bus cycle has been completed.

Physical Bus Grant Acknowledge ($\overline{PBGACK}$)

This input signal, which indicates that some other device has become master of the physical bus, should not be asserted until the following conditions have been met:

1. A physical bus grant ($\overline{PBG}$) has been received through the arbitration process;
2. $\overline{PAS}$ is negated, indicating that neither the MC68851 nor the logical bus master is using the physical bus;
3. $\overline{DSACKx}$ are negated, indicating that no external device is still driving the data bus; and

4. $\overline{\text{PBGACK}}$ is negated, indicating that no other device is still claiming bus mastership.

$\overline{\text{PBGACK}}$ must remain asserted as long as any device other than the MC68851 is bus master.

LOGICAL BUS ARBITRATION

The following paragraphs describe the five-wire bus arbitration pins used to determine which device is master of the logical bus.

Logical Bus Request In ($\overline{\text{LBRI}}$)

This input indicates that a device with higher priority than the MC68851 or the current logical bus master requires ownership of the logical bus.

Logical Bus Request Out ($\overline{\text{LBRO}}$)

This output, asserted to inform the processor that the MC68851 requires ownership of the logical bus, is used as a portion of the relinquish operation and the relinquish and retry operation.

The request input to the logical bus arbiter (usually the main processor) should consist of the wire-ORed request inputs to $\overline{\text{LBRI}}$ logically ORed with the $\overline{\text{LBRO}}$ output of the MC68851.

Logical Bus Grant In ($\overline{\text{LBGI}}$)

This input, generated by the MC68020, indicates that the MC68020 will release ownership of the bus at the completion of the current bus cycle or, if an alternate master is currently the owner of the bus, that the MC68020 will not claim the bus after the alternate master has released it.

Logical Bus Grant Out ($\overline{\text{LBGO}}$)

This output indicates that the MC68851 has recognized and synchronized the assertion of $\overline{\text{LBGI}}$ by the MC68020, has detected the assertion of $\overline{\text{LBRI}}$, and is passing the bus grant to an alternate logical bus master or to arbitration prioritization circuitry.

Logical Bus Grant Acknowledge ($\overline{\text{LBGACK}}$)

This bidirectional, three-state signal indicates that a logical bus master other than the CPU has taken control of the logical bus.

This signal is asserted by the MC68851 to indicate that it is the current logical bus master. $\overline{\text{LBGACK}}$ is also monitored as an input to determine when the MC68851 can become bus master.

SIGNAL SUMMARY

Table 1 provides a summary of the signals discussed in the previous paragraphs.

Table 1. Signal Summary

Signal Name	Mnemonic	Input/Output	Active State	Three-State	Driven by MC68851 When
Logical Address Bus	LA8–LA31	Input	High	—	—
Physical Address Bus	PA8–PA31	Output	High	Yes	MC68851 Owns Physical Bus
Shared Address Bus	A0–A7	Input/Output	High	Yes	MC68851 Owns Logical and Physical Buses
Function Codes	FC0–FC3	Input/Output	High	Yes	MC68851 Owns Logical and Physical Buses
Data Bus	D0–D31	Input/Output	High	Yes	Read from MC68851 Registers or MC68851 Write Cycle
Size	SIZ0–SIZ1	Input/Output	High	Yes	MC68851 Owns Logical and Physical Buses
Cache Load Inhibit	$\overline{CLI}$	Output	Low	No	Always
Asynchronous Control	$\overline{ASYNC}$	Input	Low	—	—
Read-Modify-Write Cycle	$\overline{RMC}$	Input/Output	Low	Yes	MC68851 Owns Logical and Physical Buses
Logical Address Strobe	$\overline{LAS}$	Input	Low	—	—
Physical Address Strobe	$\overline{PAS}$	Output	Low	Yes	MC68851 Owns Physical Bus
Data Strobe	$\overline{DS}$	Input/Output	Low	Yes	MC68851 Owns Logical and Physical Buses
Read/Write	R/$\overline{W}$	Input/Output	High/ Low	Yes	MC68851 Owns Logical and Physical Buses
Data Transfer and Size Acknowledge	$\overline{DSACK0}$/$\overline{DSACK1}$	Input/Output	Low	Yes	Access to Address Map Occupied by MC68851 Interface Register Set
Data Bus Disable	DBDIS	Output	High	No	Always
Bus Error	$\overline{BERR}$	Input/Output	Low	Yes	Exceptional Condition is Generated by Address Translation
Halt	$\overline{HALT}$	Input/Output	Low	Yes	Exceptional Condition is Generated by Address Translation
Reset	$\overline{RESET}$	Input	Low	—	—
Physical Bus Request	$\overline{PBR}$	Input	Low	—	—
Physical Bus Grant	$\overline{PBG}$	Output	Low	No	Always
Physical Bus Grant Acknowledge	$\overline{PBGACK}$	Input	Low	—	—
Logical Bus Request In	$\overline{LBRI}$	Input	Low	—	—
Logical Bus Request Out	$\overline{LBRO}$	Output	Low	No	Always
Logical Bus Grant In	$\overline{LBGI}$	Input	Low	—	—
Logical Bus Grant Out	$\overline{LBGO}$	Output	Low	No	Always
Logical Bus Grant Acknowledge	$\overline{LBGACK}$	Input/Output	Low	Yes	MC68851 Has Assumed Mastership of the Logical Bus
Clock	CLK	Input	—	—	—
Power Supply	V_{CC}	Input	—	—	—
Ground	GND	Input	—	—	—

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.5 to +7.0	V
Operating Temperature	T_A	0 to 70	°C
Storage Temperature	T_{stg}	−55 to +150	°C

The device contains protective circuitry against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of voltages higher than maximum-rated voltages to these high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{CC}) enhances reliability of operation.

THERMAL CHARACTERISTICS — PGA PACKAGE

Characteristic	Symbol	Value	Rating
Thermal Resistance — Ceramic			
Junction to Ambient	θ_{JA}	30*	°C/W
Junction to Case	θ_{JC}	15*	°C/W

*Estimated

5

POWER CONSIDERATIONS

The average die-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{I/O}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts — Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications, $P_{I/O} < P_{INT}$ and can be neglected.

An appropriate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273\ °C) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at thermal equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

The total thermal resistance of a package (θ_{JA}) can be separated into two components, θ_{JC} and θ_{CA}, representing the barrier to heat flow from the semiconductor junction to the package (case) surface (θ_{JC}) and from the case to the outside ambient air (θ_{CA}). These terms are related by the equation:

$$\theta_{JA} = \theta_{JC} + \theta_{CA} \quad (4)$$

θ_{JC} is device related and cannot be influenced by the user. However, θ_{CA} is user dependent and can be minimized by such thermal management techniques as heat sinks, ambient air cooling, and thermal convection. Thus, good thermal management on the part of the user can significantly reduce θ_{CA} so that θ_{JA} approximately equals θ_{JC}. Substitution of θ_{JC} for θ_{JA} in equation (1) results in a lower semiconductor junction temperature.

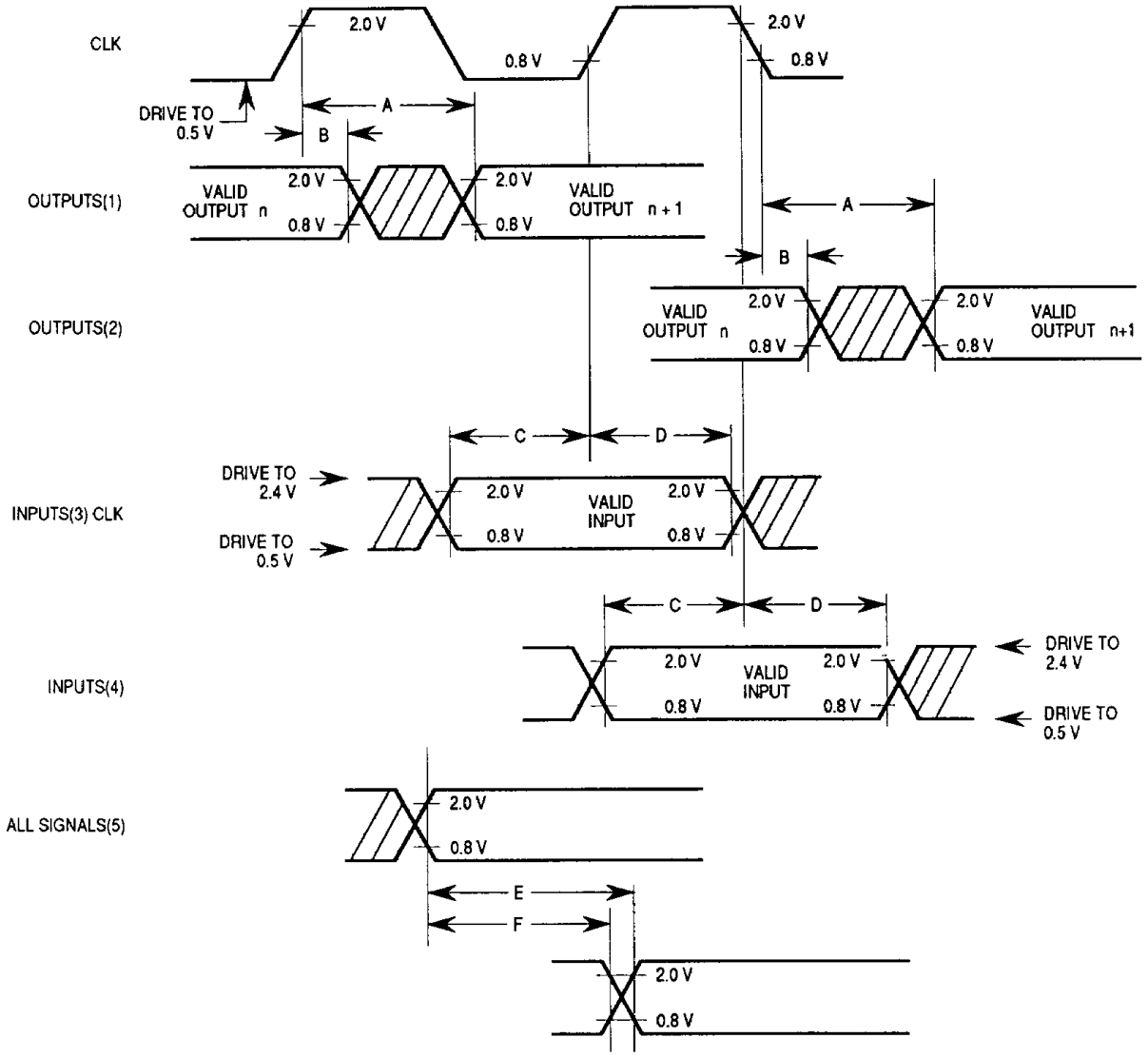
Values for thermal resistance presented in this document, unless estimated, were derived using the procedure described in Motorola Reliability Report 7843, *Thermal Resistance Measurement Method for MC68XXX Microcomponent Devices*, and are provided for design purposes only. Thermal measurements are complex and dependent on procedure and setup. User-derived values for thermal resistance may differ.

AC ELECTRICAL SPECIFICATIONS DEFINITIONS

The AC specifications presented consist of output delays, input setup and hold times, and signal skew times. All signals are specified relative to an appropriate edge of the clock and possibly to one or more other signals.

The measurement of the AC specifications is defined by the waveforms in Figure 7. To test the parameters guaranteed by Motorola, inputs must be driven to the voltage levels specified. Outputs of the MC68851 are specified with minimum and/or maximum limits, as appropriate, and are measured as shown. Inputs to the MC68851 are specified with minimum and, as appropriate, maximum setup and hold times, and are measured as shown. Finally, the measurements for signal-to-signal specifications are also shown.

Note that the testing levels used to verify conformance to the AC specifications does not affect the guaranteed DC operation of the device as specified in the DC electrical specifications.



NOTES:
1. This output timing is applicable to all parameters specified relative to the rising edge of the clock.
2. This output timing is applicable to all parameters specified relative to the falling edge of the clock.
3. This input timing is applicable to all parameters specified relative to the rising edge of the clock.
4. This input timing is applicable to all parameters specified relative to the falling edge of the clock.
5. This timing is applicable to all parameters specified relative to the assertion/negation of another signal.

LEGEND:
A. Maximum output delay specification.
B. Minimum output hold time.
C. Minimum input setup time specification.
D. Minimum input hold time specification.
E. Signal valid to signal valid specification (maximum or minimum).
F. Signal valid to signal invalid specification (maximum or minimum).

Figure 7. Drive Levels and Test Points for AC Specifications

DC ELECTRICAL SPECIFICATIONS (V_{CC} = 5.0 Vdc ± 5%; GND = 0 Vdc; T_A = 0 to 70°C)

Characteristic	Symbol	Min	Max	Unit
Input High Voltage	V_{IH}	2.0	V_{CC}	V
Input Low Voltage	V_{IL}	GND–0.5	0.8	V
Input Leakage Current @ 5.25 V — CLK, $\overline{RESET}$, LA8–LA31, $\overline{LAS}$, $\overline{LBRI}$, $\overline{LBGI}$, $\overline{PBR}$, $\overline{PBGACK}$, $\overline{ASYNC}$	I_{in}	—	10	μA
Hi-Z (Off-State) Input Current @ 2.4 V/0.4 V — $\overline{DSACK0}$, $\overline{DSACK1}$, D0–D31, FC0–FC3, SIZ0/SIZ1, $\overline{PAS}$, $\overline{DS}$, R/$\overline{W}$, $\overline{RMC}$, $\overline{BERR}$, $\overline{HALT}$, $\overline{LBGACK}$	I_{TSI}	—	20	μA
Output High Voltage (I_{OH} = −400 μA) — A0–A7, $\overline{DSACK0}$, $\overline{DSACK1}$, D0–D31, FC0–FC3, SIZ0–SIZ1, $\overline{PAS}$, $\overline{DS}$, R/$\overline{W}$, $\overline{RMC}$, $\overline{BERR}$, $\overline{HALT}$, $\overline{LBGACK}$, PA8–PA31, DBDIS, $\overline{LBRO}$, $\overline{LBGO}$, $\overline{PBG}$, $\overline{CLI}$	V_{OH}	2.4	—	V
Output Low Voltage (I_{OL} = 5.3 mA) — $\overline{DSACK0}$, $\overline{DSACK1}$, $\overline{HALT}$, $\overline{PAS}$, $\overline{DS}$, R/$\overline{W}$, $\overline{RMC}$, $\overline{BERR}$, $\overline{LBGACK}$, DBDIS, $\overline{LBRO}$, $\overline{LBGO}$, $\overline{PGB}$, $\overline{CLI}$	V_{OL}	—	0.5	V
Output Low Voltage (I_{OL} = 3.2 mA) — D0–D31, A0–A7, FC0–FC3, SIZ0–SIZ1, PA8–PA31	V_{OL}	—	0.5	V
Power Dissipation	P_D	—	1.50	W
Capacitance* (V_{in} = 0, T_A = 25°C, f = 1 MHz)	C_{in}	—	20	pF

*Capacitance is periodically sampled rather than 100% tested.

5

AC ELECTRICAL SPECIFICATIONS — CLOCK INPUT (V_{CC} = 5.0 Vdc ± 5%; GND = 0 Vdc; T_A = 0 to 70 °C; see Figure 8)

No.	Characteristic	Symbol	MC68851RC12		MC68851RC16		MC68851RC20		Unit
			Min	Max	Min	Max	Min	Max	
	Frequency of Operation	f	8.0	12.5	8.0	16.67	10	20	MHz
1	Cycle Time	t_{cyc}	80	125	60	125	50	100	ns
2, 3	Clock Pulse Width	t_{CL}, t_{CH}	32	87	24	95	19	81	ns
4, 5	Clock Rise and Fall Time	t_{Cr}, t_{Cf}	—	5	—	5	—	5	ns

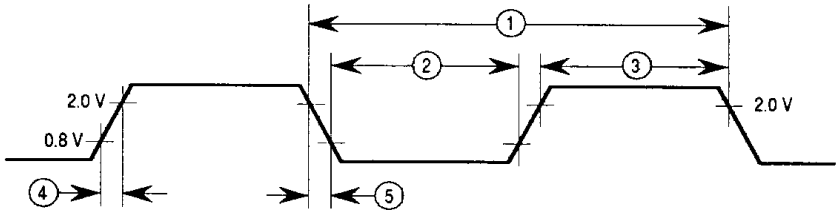


Figure 8. Clock Input Timing Diagram

AC ELECTRICAL SPECIFICATIONS — ALL BUS OPERATIONS (V_{CC} = 5.0 Vdc ± 5%; GND = 0 Vdc; T_A = 0 to 70°C; see Figures 9–17)

No.	Characteristic	Mode	MC68851RC12		MC68851RC16		MC68851RC20		Unit
			Min	Max	Min	Max	Min	Max	
6	Clock High to FC, Size, $\overline{RMC}$, Physical Address, Shared Address Valid (see Note 1 and Figures 9 and 10)	T[2]	0	40	0	30	0	25	ns
7	Clock High to FC, Size, $\overline{RMC}$, Data-Out, Physical Address, Shared Address High Impedance (see Figure 14)	T	0	40	0	30	0	25	ns
8	Clock High to FC, Size, $\overline{RMC}$, Physical Address, Shared Address Invalid (see Figures 9 and 10)	T	0	—	0	—	0	—	ns
9	Clock Transition to $\overline{PAS}$ Asserted (see Figures 9 and 10)	T	0	35	0	25	0	20	ns
9A[11]	$\overline{PAS}$ to $\overline{DS}$ Assertion (Read) (Skew) (see Figure 9)	T	−20	20	−15	15	−10	10	ns
9B	Clock Transition to $\overline{DS}$ Asserted (see Figure 10)	T	0	40	0	30	0	25	ns
11[16]	FC, Size, $\overline{RMC}$, Physical Address, Shared Address Valid to $\overline{PAS}$, $\overline{DS}$ Asserted (see Figures 9 and 10)	T	20	—	15	—	10	—	ns
12	Clock Low to $\overline{PAS}$ Negated (see Figures 9 and 10)	T	0	35	0	25	0	20	ns
12A	Clock Low to $\overline{DS}$ Negated (see Figures 9 and 10)	T	0	40	0	30	0	25	ns
13	$\overline{PAS}$, $\overline{DS}$ Negated to FC, Size, $\overline{RMC}$, Physical Address, Shared Address Invalid (see Note 20 and Figures 9 and 10)	T	20	—	15	—	10	—	ns
14	$\overline{PAS}$, $\overline{DS}$ (Read) Width Asserted (see Figures 9 and 10)	T	120	—	100	—	85	—	ns
14A	$\overline{DS}$ Width Asserted (Write) (see Figure 10)	T	50	—	40	—	35	—	ns
15	$\overline{PAS}$, $\overline{DS}$ Width Negated	T	50	—	40	—	35	—	ns
16	Clock High to $\overline{PAS}$, $\overline{DS}$, R/$\overline{W}$, DBDIS High Impedance (see Figure 9)	T	0	40	0	30	0	25	ns
17	$\overline{PAS}$, $\overline{DS}$ Negated to R/$\overline{W}$ Invalid (Read or Write) (see Figures 9 and 10)	T	20	—	15	—	10	—	ns
18	Clock High to R/$\overline{W}$ High (Read) (see Figure 9)	T	0	40	0	30	0	25	ns
20	Clock High to R/$\overline{W}$ Low (Write) (see Figure 10)	T	0	40	0	30	0	25	ns
21	R/$\overline{W}$ High to $\overline{PAS}$ Asserted (see Figure 9)	T	20	—	15	—	10	0	ns
22	R/$\overline{W}$ Low to $\overline{DS}$ Asserted (Write) (see Figure 10)	T	90	—	75	—	60	—	ns
23	Clock High to Data-Out Valid (see Figures 10 and 15)	T:O	0	40	0	30	0	25	ns

AC ELECTRICAL SPECIFICATIONS — ALL BUS OPERATIONS

(Continued)

No.	Characteristic	Mode	MC68851RC12		MC68851RC16		MC6851RC20		Unit
			Min	Max	Min	Max	Min	Max	
25	$\overline{DS}$ Negated to Data-Out Invalid (see Figures 10 and 15)	T/O	20	—	15	—	10	—	ns
26	Data-Out Valid to $\overline{DS}$ Asserted (see Figure 10)	T	20	—	15	—	10	—	ns
27	Data-In Valid to Clock Low (Data Setup) (see Figure 9)	T	10	—	5	—	5	—	ns
27A	$\overline{BERR}$-i/$\overline{HALT}$-i Asserted to Clock Low (Late $\overline{BERR}$/$\overline{HALT}$ Setup Time) (see Note 3 and Figure 9)	T	25	60[4]	20	45[4]	15	35	ns
29	$\overline{DS}$ Negated to Data-In Invalid (Data-In Hold Time) (see Figure 9)	T	0	—	0	—	0	—	ns
29A	$\overline{DS}$ Negated to Data-In High Impedance (see Figure 9)	T	0	80	0	60	0	50	ns
31[12]	$\overline{DSACKx}$ Asserted to Data-In Valid (see Figure 9)	T	—	60	—	50	—	40	ns
31A[13]	$\overline{DSACKx}$ Asserted to $\overline{DSACKx}$ Valid (Assertion Skew) (see Figures 9 and 10)	T	—	20	—	15	—	10	ns
32	$\overline{RESET}$ Input Transition Time (see Figure 17)	X	—	2	—	2	—	2	Clk Per
33	Clock Low to $\overline{PBG}$ Asserted (see Figure 14)	X	0	40	0	30	0	25	ns
34	Clock Low to $\overline{PBG}$ Negated (see Figure 14)	X	0	40	0	30	0	25	ns
35A	$\overline{PBR}$ Asserted to $\overline{PBG}$ Asserted ($\overline{RMC}$ Not Asserted) (see Figure 14)	T	1.5	3.5	1.5	3.5	1.5	3.5	Clk Per
35B	$\overline{PBR}$ Asserted to $\overline{PBG}$ Asserted ($\overline{RMC}$ Not Asserted) (see Figure 14)	M	1.5	5.5	1.5	5.5	1.5	5.5	Clk Per
36	$\overline{PBR}$ Negated to $\overline{PBG}$ Negated (Transient or Spurious Request) (see Figure 14)	X	1.5	3.5	1.5	3.5	1.5	3.5	Clk Per
37	$\overline{PBGACK}$ Asserted to $\overline{PBG}$ Negated (see Figure 14)	X	1.5	3.5	1.5	3.5	1.5	3.5	Clk Per
39	$\overline{PBG}$ Width Negated (see Figure 14)	X	1.5	—	1.5	—	1.5	—	Clk Per
39A	$\overline{PBG}$ Width Asserted (see Figure 14)	X	1.5	—	1.5	—	1.5	—	Clk Per
40A	Clock High to DBDIS Negated (Read) (see Figure 9)	T	0	40	0	30	0	25	ns
40B	Clock Low to DBDIS Negated (Write (T)) (Read (O)) (see Figure 10)	T/O	0	40	0	30	0	25	ns
41A	Clock Low to DBDIS Asserted (Read (T)) (Write (O)) (see Figures 9 and 15)	T/O	0	40	0	30	0	25	ns
41B	Clock High to DBDIS Asserted (Write) (see Figures 10 and 15)	T	0	40	0	30	0	25	ns
43	$\overline{PBGACK}$ Negated to $\overline{PAS}$, Physical Address Impedance Change (see Note 10 and Figure 14)	X	0.5	2.5	0.5	2.5	0.5	2.5	Clk Per

5

AC ELECTRICAL SPECIFICATIONS — ALL BUS OPERATIONS

(Continued)

No.	Characteristic	Mode	MC68851RC12		MC68851RC16		MC68851RC20		Unit
			Min	Max	Min	Max	Min	Max	
44	R/$\overline{W}$ Asserted to DBDIS Negated (Read or Write) (see Figures 9 and 10)	T	20	—	15	—	10	—	ns
45A	DBDIS Width Negated (Read) (see Figure 9)	T	80	—	60	—	50	—	ns
45B[15]	DBDIS Width Negated (Write) (see Figure 10)	T	160	—	120	—	100	—	ns
46	R/$\overline{W}$ Width Asserted (Read or Write) (see Figures 9 and 10)	T	180	—	150	—	125	—	ns
47A	Asynchronous Input Setup Time to Sampling Clock Edge (see Figures 9, 10, and 14)	T	10	60[4]	5	45[4]	5	35	ns
47B	$\overline{PAS}$, $\overline{DS}$ Negated to Asynchronous Input Negated (see Figures 9 and 10)	T	0	100	0	80	0	65	ns
48[14]	$\overline{DSACKx}$-i Asserted to $\overline{BERR}$-i/$\overline{HALT}$-i Asserted (Late Bus Error or Retry) (see Figures 9 and 10)	T	—	40	—	30	—	25	ns
53	Data-Out Hold from Clock High (see Figure 10)	T/O	0	—	0	—	0	—	ns
53A	Data-Out Hold from $\overline{LAS}$ Negated (see Figure 15)	O	0	—	0	—	0	—	ns
55	R/$\overline{W}$ Low to Data Bus Impedance Change (see Figure 10)	T	40	—	30	—	25	—	ns
56	DBDIS Asserted to Data Bus Impedance Change (see Figure 15)	O	15	—	15	—	15	—	ns
56A	Data Bus Impedance Change to DBDIS Negated (see Figure 15)	O	0	—	0	—	0	—	ns
59	DBDIS High to R/$\overline{W}$ Low (see Figure 9)	T	20	—	15	—	10	—	ns
60	$\overline{BERR}$-i Negated to $\overline{HALT}$-i Invalid (Hold Time for Retry) (see Figure 9)	T	0	—	0	—	0	—	ns
63	$\overline{DS}$ Negated to DBDIS Asserted (Write) (see Figure 10)	T	20	—	15	—	10	—	ns
64	DBDIS Negated to Data Bus Impedance Change (Write) (see Figure 10)	T	20	—	15	—	10	—	ns
65	Clock Low to $\overline{LBGACK}$-o, $\overline{LBGO}$ Asserted (see Figures 12 and 13)	X	0	40	0	30	0	25	ns
66	Clock Low to $\overline{LBGACK}$-o, $\overline{LBGO}$ Negated (see Figures 12 and 13)	X	0	40	0	30	0	25	ns
67	$\overline{LBGACK}$-o Asserted to $\overline{LBRO}$ Negated (see Figure 12)	T	20	60	15	45	10	35	ns
68	$\overline{LBGACK}$-o Asserted to $\overline{CLI}$ Asserted (see Figure 12)	T	0	80	0	60	0	50	ns
69	$\overline{CLI}$ Negated to $\overline{LBGACK}$-o Negated (see Figure 12)	T	0	80	0	60	0	50	ns
70	$\overline{LBGACK}$-o Asserted to DBDIS Asserted (see Figure 12)	T	−20	20	−15	15	−10	10	ns

AC ELECTRICAL SPECIFICATIONS — ALL BUS OPERATIONS

(Continued)

No.	Characteristic	Mode	MC68851RC12		MC68851RC16		MC68851RC20		Unit
			Min	Max	Min	Max	Min	Max	
71	$\overline{LBGI}$ Asserted to $\overline{LBGO}$ Asserted (see Figure 13)	X	1.5	12.5[8]	1.5	12.5[8]	1.5	12.5[8]	Clk Per
72	$\overline{LBRI}$ Negated to $\overline{LBGO}$ Negated (see Figure 13)	X	1.5	3.5	1.5	3.5	1.5	3.5	Clk Per
73	$\overline{LBGO}$ Width Asserted (see Figure 13)	X	40	—	30	—	25	—	ns
74	$\overline{LBGO}$ Width Negated (see Figure 13)	X	40	—	30	—	25	—	ns
75	$\overline{LBGI}$ Negated to $\overline{LBGO}$ Negated (see Figure 13)	X	1.5	3.5	1.5	3.5	1.5	3.5	Clk Per
77	$\overline{LBGI}$ Asserted to $\overline{LBGACK}$-o Asserted (see Figure 12)	T	1.5	3.5	1.5	3.5	1.5	3.5	Clk Per
79	$\overline{RESET}$ Width Asserted (V_{CC} Active and Stable) (see Figure 17)	X	512	—	512	—	512	—	Clk Per
79A	$\overline{RESET}$ Width Asserted (V_{CC} Stable > 512 Clocks) (see Figure 17)	X	10	—	10	—	10	—	Clk Per
80	$\overline{RESET}$ Asserted to Bus Control Signals Negated (V_{CC} Active and Stable) (see Figure 17)	X	0	4	0	4	0	4	Clk Per
81	$\overline{RESET}$ Negated to $\overline{LAS}$ Asserted (see Figure 17)	X	4	—	4	—	4	—	Clk Per
82	$\overline{RESET}$ Negated to Mode Select Data Invalid (Hold) (see Figure 17)	X	0	—	0	—	0	—	ns
83	Mode Select Data Valid to $\overline{RESET}$ Negated (Setup) (see Figure 17)	X	2	—	2	—	2	0	Clk Per
84	Clock Transition to $\overline{HALT}$/$\overline{BERR}$/$\overline{LBRO}$ Asserted (Logical Master Relinquish and Retry) (see Figure 11)	M	0	40	0	30	0	25	ns
86A	$\overline{LAS}$ Asserted to $\overline{HALT}$/$\overline{BERR}$/$\overline{LBRO}$ Asserted (Logical Master Relinquish and Retry) (see Figure 12)	MS	0.5	1.5[7]	0.5	1.5[7]	0.5	1.5[7]	Clk Per
86B	$\overline{LAS}$ Asserted to $\overline{HALT}$/$\overline{BERR}$/$\overline{LBRO}$ Asserted (Logical Master Relinquish and Retry) (see Figure 12)	MA	.5	3.0[6]	.5	3.0[6]	.5	3.0[6]	Clk Per
89	$\overline{HALT}$ Negated to $\overline{LBGACK}$-o Asserted (see Figure 12)	M	20	60	15	45	10	35	ns
90	$\overline{LAS}$ Negated to $\overline{BERR}$-o Negated (Termination of Relinquish and Retry) (see Figure 12)	M	0	40	0	30	0	25	ns
91	Logical Address, FC, $\overline{RMC}$, R/$\overline{W}$ Valid to Clock High (Setup) (see Figure 11)	MS/OS	40	—	30	—	25	—	ns
92A	Logical Address, FC, $\overline{RMC}$, R/$\overline{W}$ Valid to $\overline{LAS}$ Asserted (see Figures 11, 15, and 16)	MS/OS	20	—	15	—	10	—	ns
92B	Logical Address, FC, $\overline{RMC}$, R/$\overline{W}$ Valid to $\overline{LAS}$ Asserted (see Figures 13, 15, and 16)	MA/OA	0	—	0	—	0	—	ns

5

AC ELECTRICAL SPECIFICATIONS — ALL BUS OPERATIONS

(Continued)

No.	Characteristic	Mode	MC68851RC12		MC68851RC16		MC68851RC20		Unit
			Min	Max	Min	Max	Min	Max	
93A	$\overline{LAS}$ Negated to Logical Address, FC, RMC, R/$\overline{W}$ Invalid (Sync Mode) (see Figures 11, 15, and 16)	MS/OS	20	—	15	—	10	—	ns
93B	$\overline{LAS}$ Negated to Logical Address, FC, RMC, R/$\overline{W}$ Invalid (Async Mode) (see Figures 13, 15, and 16)	MA/OA	0	—	0	—	0	—	ns
95	Logical Address Valid to Physical Address Valid (Translation Cache Hit or CPU Space Cycle) (see Figures 11, 13, 15, and 16)	M	0	50[19]	0	45[19]	0	38	ns
96	Size, Shared Address Valid to $\overline{LAS}$ Asserted (Access to MC68851 Register) (see Figures 15 and 16)	OS	20	—	15	—	10	—	ns
97	$\overline{LAS}$ Negated to Size, Shared Address Invalid (Access to MC68851 Register) (see Figures 15 and 16)	OS	20	—	15	—	10	—	ns
100	$\overline{LAS}$ Asserted to Clock Low (Setup Time) (see Figure 11)	MS	40	—	30	—	25	—	ns
103	$\overline{LAS}$ Width Asserted (see Figures 11 and 13)	M	1.5	—	1.5	—	1.5	—	Clk Per
104	$\overline{LAS}$, $\overline{DS}$ Width Negated (see Figure 11)	MS	0.5	—	0.5	—	0.5	—	Clk Per
104A	$\overline{LAS}$, $\overline{DS}$ Width Negated (see Figure 13)	MA	30	—	20	—	10	—	ns
105	$\overline{ASYNCH}$ Asserted to $\overline{LAS}$, $\overline{DS}$ Asserted (see Figure 13)	M	1.5	—	1.5	—	1.5	—	Clk Per
106	$\overline{LAS}$, $\overline{DS}$ Negated to $\overline{ASYNC}$ Negated (see Figures 11 and 13)	M	0	—	0	—	0	—	ns
107	$\overline{ASYNC}$ Negated to $\overline{LAS}$, $\overline{DS}$ Asserted (For Synchronous Next Cycle) (see Figure 11)	M	1.5	—	1.5	—	1.5	—	Clk Per
108	Data Valid to $\overline{DS}$ Asserted (Write Setup Time to MC68851) (see Figure 16)	O	0	—	0	—	0	—	ns
109A	$\overline{DS}$ Negated to Data Invalid (Write Hold Time to MC68851) (see Figure 16)	O	0	—	0	—	0	—	ns
109B	$\overline{LAS}$ Negated to Data High Impedance (see Figure 16)	O	0	80	0	60	0	50	ns
110	$\overline{DSACKx}$-o Asserted to $\overline{DSACKx}$-o Valid (see Figures 15 and 16)	O	0	20[18]	0	15[18]	0	10[18]	ns
111	Clock High to $\overline{DSACKx}$-o Asserted (see Figures 15 and 16)	O	0	40	0	30	0	25	ns
112A	$\overline{LAS}$ Asserted to $\overline{DSACKx}$-o Asserted (see Figures 15 and 16)	OS	2.0	23	2.0	23	2.0	23	Clk Per
112B	$\overline{LAS}$ Asserted to $\overline{DSACKx}$-o Asserted (see Figures 15 and 16)	OA	2.0	26	2.0	26	2.0	26	Clk Per
113	$\overline{LAS}$ Negated to $\overline{DSACKx}$-o, $\overline{BERR}$-o Negated (see Figures 15 and 16)	O	0	40	0	30	0	25	ns
114	$\overline{LAS}$ Negated to $\overline{DSACKx}$-o, $\overline{BERR}$-o High Impedance (see Figure 15)	O	0	60	0	40	0	30	ns

AC ELECTRICAL SPECIFICATIONS — ALL BUS OPERATIONS (Concluded)

No.	Characteristic	Mode	MC68851RC12		MC68851RC16		MC68851RC20		Unit
			Min	Max	Min	Max	Min	Max	
115	Clock Low to $\overline{PAS}$ Asserted (see Figure 11)	MS	0[5]	35[5]	0[5]	25[5]	0[5]	20[5]	ns
116	Clock Transition (Rising or Falling Edge) to PAS Asserted (see Figures 11 and 13)	M	0[5]	35[5]	0[5]	25[5]	0[5]	20[5]	ns
116A	Clock Low to $\overline{CLI}$ Asserted (see Figures 11 and 15)	M	0[5]	40[5]	0[5]	30[5]	0[5]	25[5]	ns
117	$\overline{LAS}$ Asserted to $\overline{PAS}$ Asserted (Synchronous Translation with ATC Hit) (see Figure 11)	MS	0.5[7]	1.5[7]	0.5[7]	1.5[7]	0.5[7]	1.5[7]	Clk Per
118	$\overline{LAS}$ Negated to $\overline{PAS}$ Negated (see Figures 11 and 13)	M	0	20	0	15	0	10	ns
119	Physical Address Valid to $\overline{PAS}$ Asserted (see Figure 11)	M	20	—	15	—	10	—	ns
120A	$\overline{PAS}$ Negated to Physical Address Invalid (see Figure 13)	MS	15	—	10	—	10	—	ns
120B	$\overline{PAS}$ Negated to Physical Address Invalid (see Figure 13)	MA	0	—	0	—	0	—	ns
121	$\overline{LAS}$ Asserted to $\overline{PAS}$ Asserted (Asynchronous Operation Only) (see Figure 13)	M	0.5[6]	3.0[9]	0.5[6]	3.0[9]	0.5[6]	3.0[9]	Clk Per
122	$\overline{LAS}$ Negated to $\overline{PAS}$ High Impedance (PBR Asserted by Alternate Physical Master) (see Figures 11 and 14)	M	—	80	—	60	—	50	ns
123	Physical Address Valid to $\overline{CLI}$ Asserted (CPU Space Cycle Not Accessing MC68851) (see Figures 15 and 16)	M	20	—	15	—	10	—	ns
124A	$\overline{LAS}$ Asserted to $\overline{CLI}$ Asserted (CPU Space Cycle Not Accessing MC68851) (see Figures 15 and 16)	MA	0.5[21]	3.0[21]	0.5[21]	3.0[21]	0.5[21]	3.0[21]	Clk Per
124B	$\overline{LAS}$ Asserted to $\overline{CLI}$ Asserted (CPU Space Cycle Not Accessing MC68851) (see Figures 15 and 16)	MS	0.5[21]	1.5[21]	0.5[21]	1.5[21]	0.5[21]	1.5[21]	Clk Per
126	$\overline{LAS}$ Negated to $\overline{CLI}$ Negated (CPU Space Cycle Not Accessing MC68851) (see Figures 15 and 16)	M	0	40	0	30	0	25	ns
127	$\overline{CLI}$ Negated to Physical Address Invalid (CPU Space Cycle Not Accessing MC68851) (see Figures 15 and 16)	MS	10	—	5	—	5	—	ns
128	$\overline{PAS}$ Asserted to $\overline{CLI}$ Asserted (Not CPU Space Access) (see Figure 11)	M	—	20	—	15	—	10	ns
129	$\overline{PAS}$ Negated to $\overline{CLI}$ Negated (Not CPU Space Access) (see Figure 11)	M	5	40	5	30	5	25	ns

NOTES:

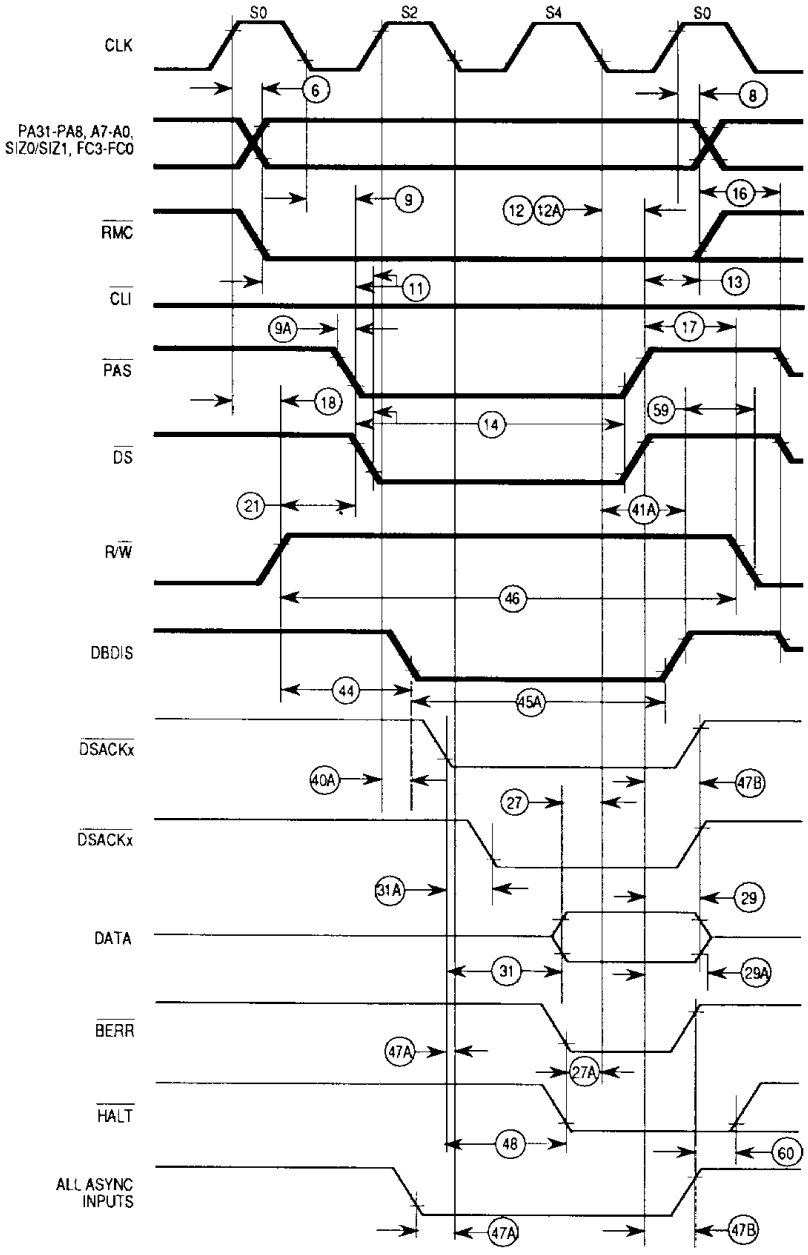
1. In this specification, the terms "high," "low," "asserted," "negated," "valid," and "invalid" are used frequently to describe a signal state. For inputs to the MC68851, high indicates that the signal conforms to the V_{IH} voltage specification; low indicates that the V_{IL} specification is satisfied. Similarly, a MC68851 output is high if it conforms to the V_{OH} specification and 'low' if it conforms to the V_{OL} parameter. An active-low input (output) is asserted if it satisfies the respective V_{IL} (V_{OL}) requirements and is negated if it satisfies the V_{IH} (V_{OH}) specification. A signal is valid if it conforms to either the voltage high or the voltage low specifications and is an appropriate value for the current

operation (for example, R/$\overline{W}$ should output a valid low during an MC68851-initiated write cycle). A signal is 'invalid' if it either does not conform to the V_H or V_L specifications or is an inappropriate value for the current operation.

2. To better understand the parameters given, a mode identification is included with each specification: "X" indicates that this specification is valid in any operating mode whatever; "T" indicates that the MC68851 is the current bus master and is performing a table search operation; "M" indicates that the MC68851 is mapping translations for the current bus master with "MS" indicating that the master is operating synchronously with the MC68851, "MA" indicating an asynchronous master, and "MX" indicating that the parameter is valid for any type of logical master; "O" indicates that the parameter is valid for operations which access the MC68851 internal registers.
3. Due to the numerous MC68851 signals used as inputs in one operating mode and as outputs in another, some attempt has been made to clarify whether a particular signal is acting as an input or as an output in cases where ambiguity is possible. The suffix "-o" indicates that the signal is an output of the MC68851; the suffix "-i" indicates that this signal is an input to the MC68851.
4. The maximum value for parameter #47A is specified so the system designer can deterministically identify the clock edge on which an asynchronous input to the MC68851 will be recognized. Any signal that meets the minimum specified setup time to an appropriate clock edge (rising/falling) and does not exceed the maximum time is guaranteed to be recognized as asserted on that edge. Signals that do not meet the minimum setup time may or may not be recognized; signals that exceed the maximum specified setup time may be recognized on the previous rising/falling clock edge.
5. The actual assertion delay from the low-going clock edge that causes the strobe(s) to assert includes the time specified in the parameter plus any additional delay specified on D3/D4 during MC68851 configuration at reset.
6. The actual assertion delay from the assertion of $\overline{LAS}$ when mapping in the asynchronous mode is the time specified in the parameter plus any additional delay specified on D3/D4 during MC68851 configuration at reset.
7. The actual assertion delay from the assertion of $\overline{LAS}$ is the time specified in the parameter plus any additional delay specified on D3/D4 during MC68851 configuration at reset. This specification has a range of one clock period to allow for cases in which the CPU exhibits a best-case (minimum) assertion delay for the $\overline{LAS}$ signal relative to the clock while the MC68851 $\overline{PAS}$ or $\overline{CLI}$ outputs exhibit worst-case (maximum) assertion delays. When operating in the synchronous translation mode, the MC68851 asserts $\overline{PAS}$ ($\overline{CLI}$) on the falling edge of the clock (plus additional specified delay) one clock period after the CPU drives $\overline{LAS}$.
8. The worst-case assertion delay for this specification can be reduced to 5.5 clock periods if the early processing startup mode of operation is disabled.
9. This maximum can be reduced to 2.5 clock periods if the $\overline{LAS}$ high time (negated period) is one clock period or greater.
10. This specification also applies to the signals A0–A7, FC0–FC3, SIZ0/SIZ1, and $\overline{RMC}$ if the MC68851 is awaiting the negation of $\overline{PBGACK}$ to initiate or complete a table search operation.
11. This number can be reduced to ±5 ns if the strobes have equal load.
12. If the asynchronous setup time (#47) requirements are satisfied, the $\overline{DSACKx}$ low to data setup (#31) and $\overline{DSACKx}$ low to $\overline{BERR}$ low setup time (#48) can be ignored. The data must only satisfy the data-in to clock low setup time (#27) for the following clock cycle. $\overline{BERR}$ must only satisfy the late $\overline{BERR}$ low to clock low setup time (#27A) for the following clock cycle.
13. This parameter specifies the maximum-allowable skew between $\overline{DSACK0}$ to $\overline{DSACK1}$ asserted or $\overline{DSACK1}$ to $\overline{DSACK0}$ asserted. Specification #47 must be met by either $\overline{DSACK0}$ or $\overline{DSACK1}$.
14. In the absence of $\overline{DSACKx}$, $\overline{BERR}$ is an asynchronous input using the asynchronous input setup time (#47).
15. DBDIS may stay asserted on consecutive write cycles (e.g., a retry of an MC68851 write operation).
16. Actual value depends on the clock input waveform.
17. This number can be reduced to 5 ns if $\overline{CLI}$ and $\overline{PAS}$ have equal loading.
18. This specification is valid only if the loading of the $\overline{DSACKx}$ outputs are equal (±50 pF).
19. This specification can be reduced to 35 or 50 ns at 16.67 and 12.5 MHz, respectively, for those bits of the logical address that are not translated by the MC68851. This includes all bits of the logical address, if the MC68851 translation mechanism is disabled, and all bits, LA*n*, of the logical address (page size 2^m) such that $n \leq m$.
20. This specification also applies to the signals A0–A7, FC0–FC3, and SIZ0/SIZ1 if the MC68851 is granting physical bus mastership to an alternate device during a table search operation.
21. The actual assertion delay from the assertion of $\overline{LAS}$ is the time specified in the parameter plus a delay derived from the reset configuration. Although the reset configuration allows additional strobe delay in half-clock increments, $\overline{CLI}$ is always asserted relative to the falling edge of the clock and can only be delayed by full-clock increments. Therefore, if a one or two half-clock delay is specified in the reset configuration, $\overline{CLI}$ assertion is delayed by one full clock.

NOTE

Figures 9–17 are intended to provide parametric timing information for the MC68851. Effort has been made to ensure that the diagrams provide correct functional signal relationships. However, not all relationships depicted are valid operations for the MC68851 (e.g., during a CPU space access (see Figure 14), accesses to the MC68851 will not cause assertion of $\overline{\text{CLI}}$).



NOTE: The clock signal is always depicted with a normal-width line.
LEGEND:
- Indicates that the signal is driven by the MC68851
- Indicates that the signal is driven by the main processor
- Indicates that the signal is driven by an external device or alternate bus master

Figure 9. MC68851-Initiated Read Cycle

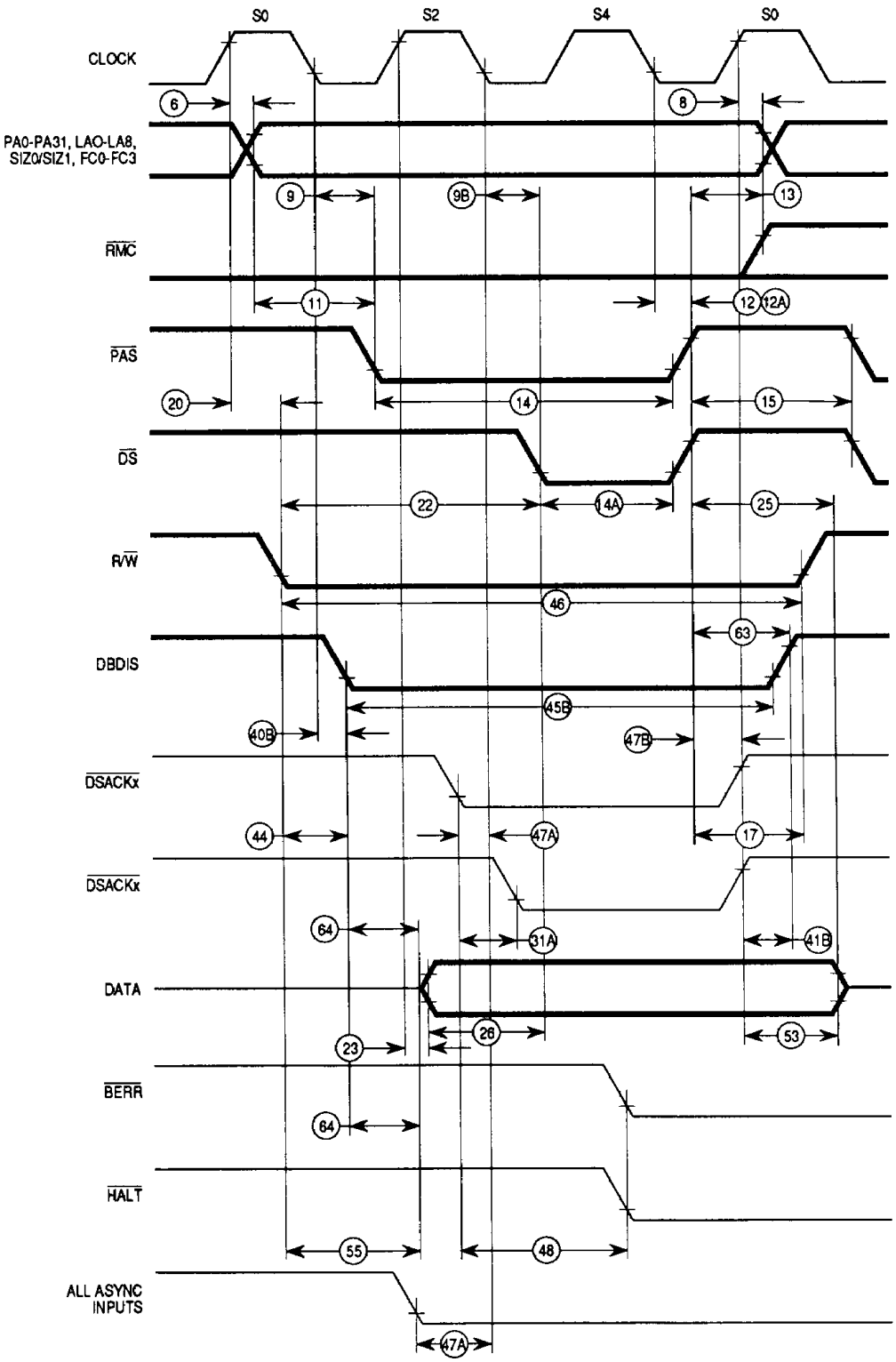


Figure 10. MC68851-Initiated Write Cycle

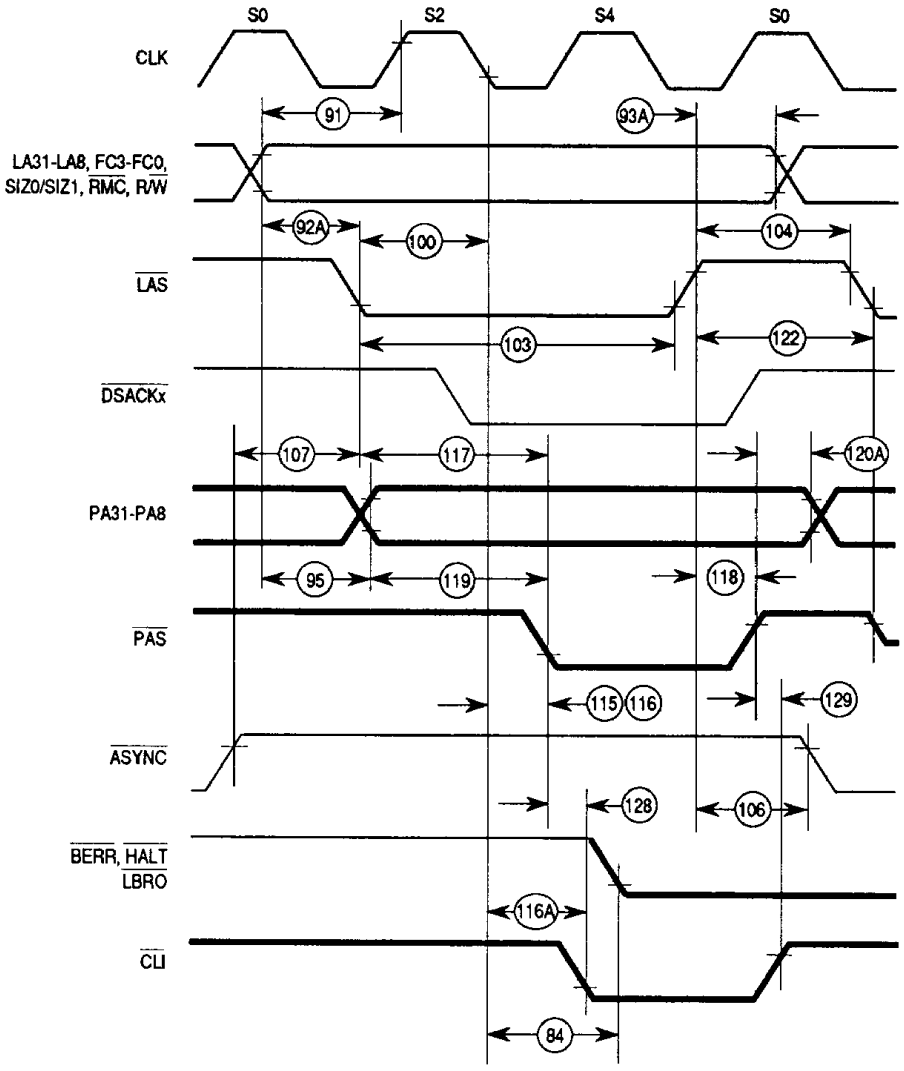


Figure 11. Synchronous Mode Translation

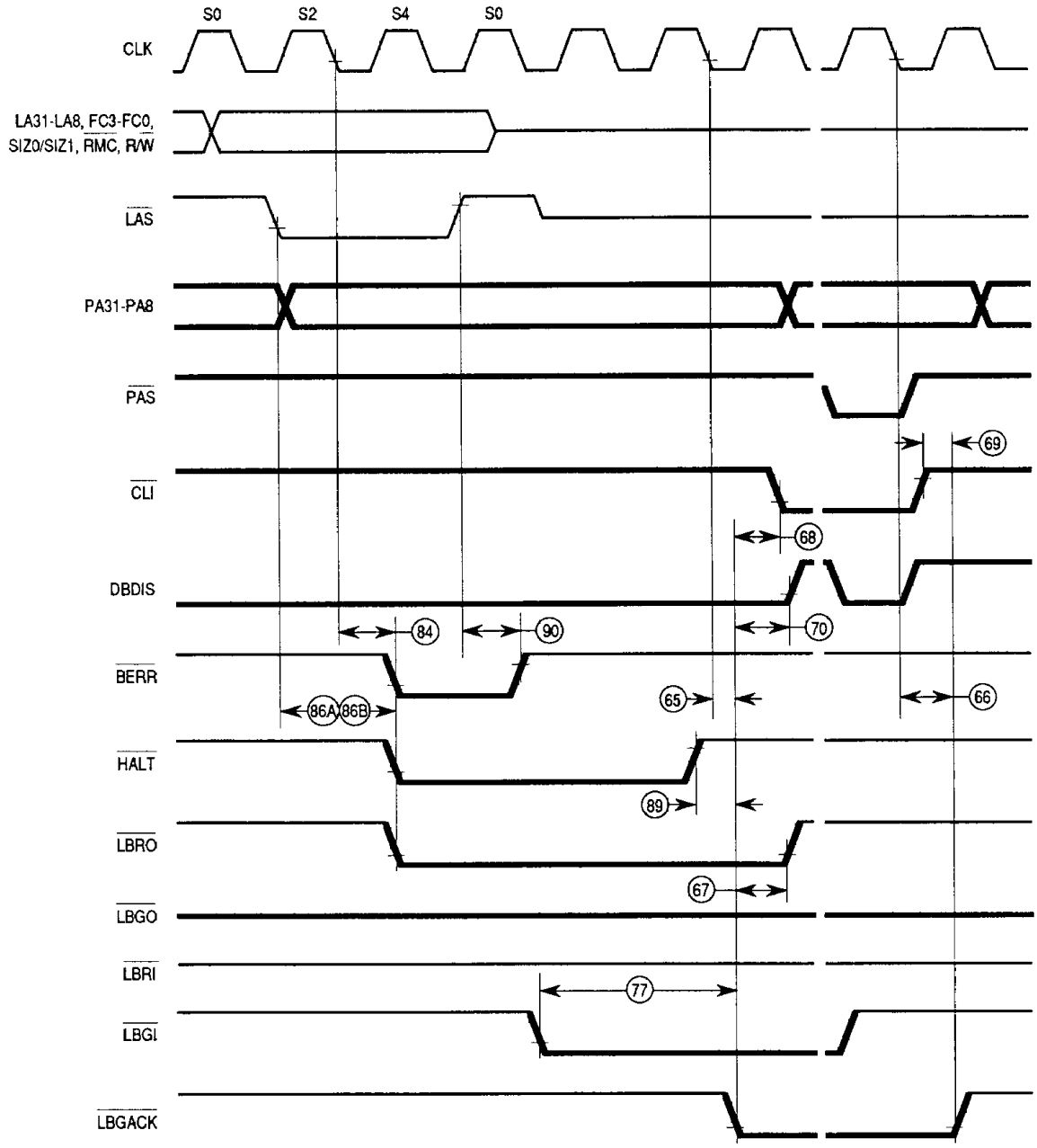


Figure 12. Logical Master Relinquish and Retry Timing Diagram

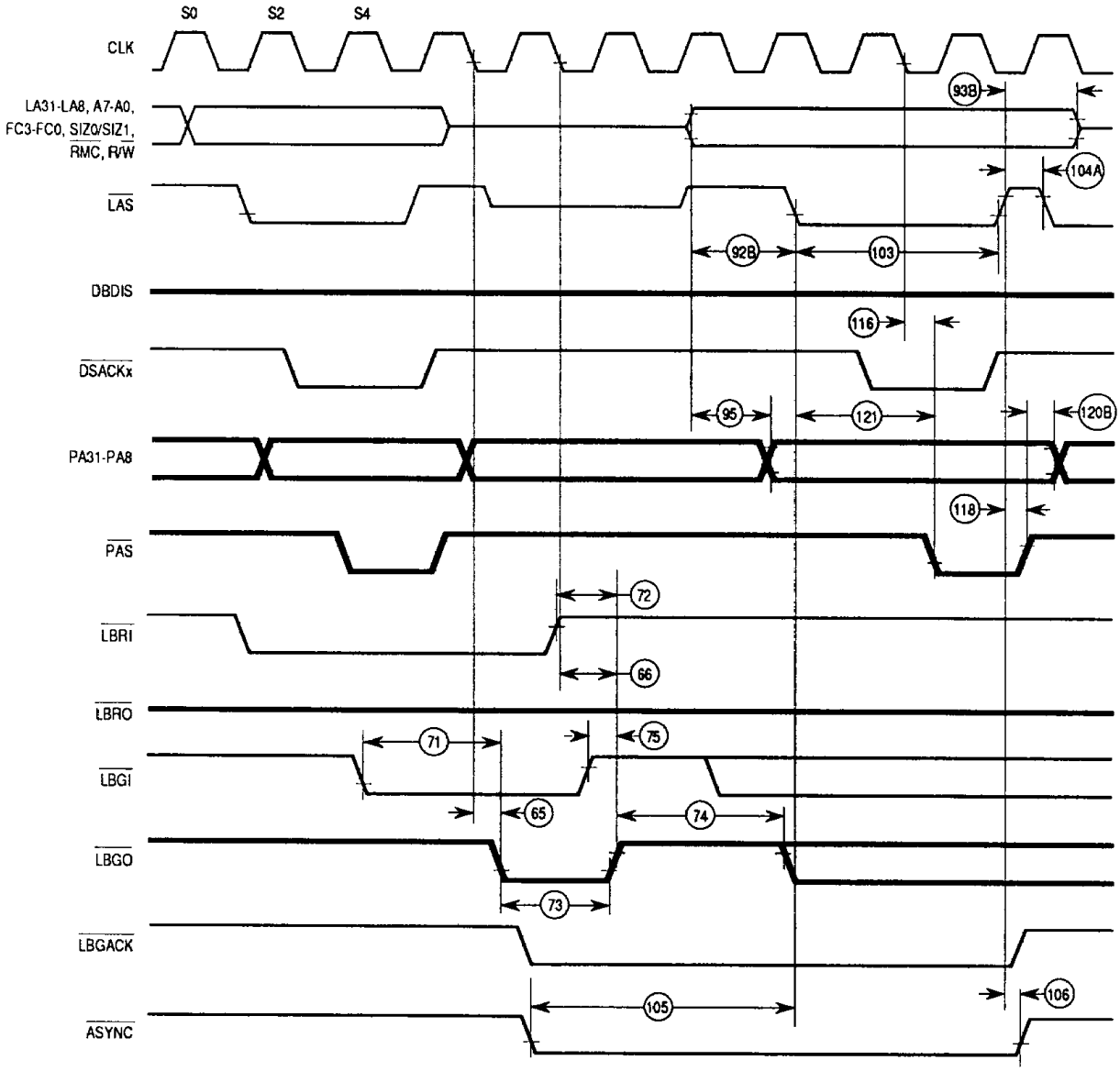
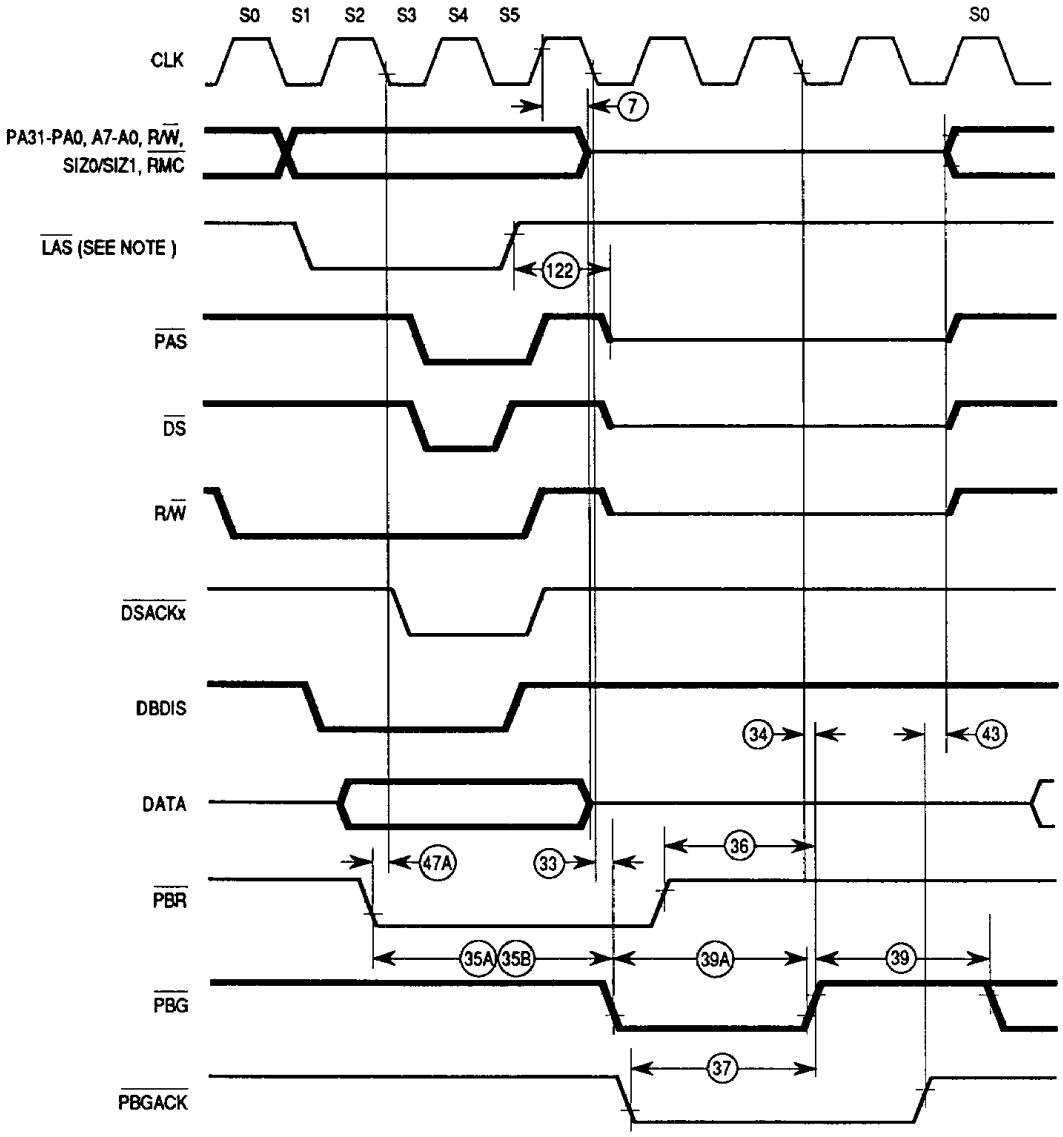
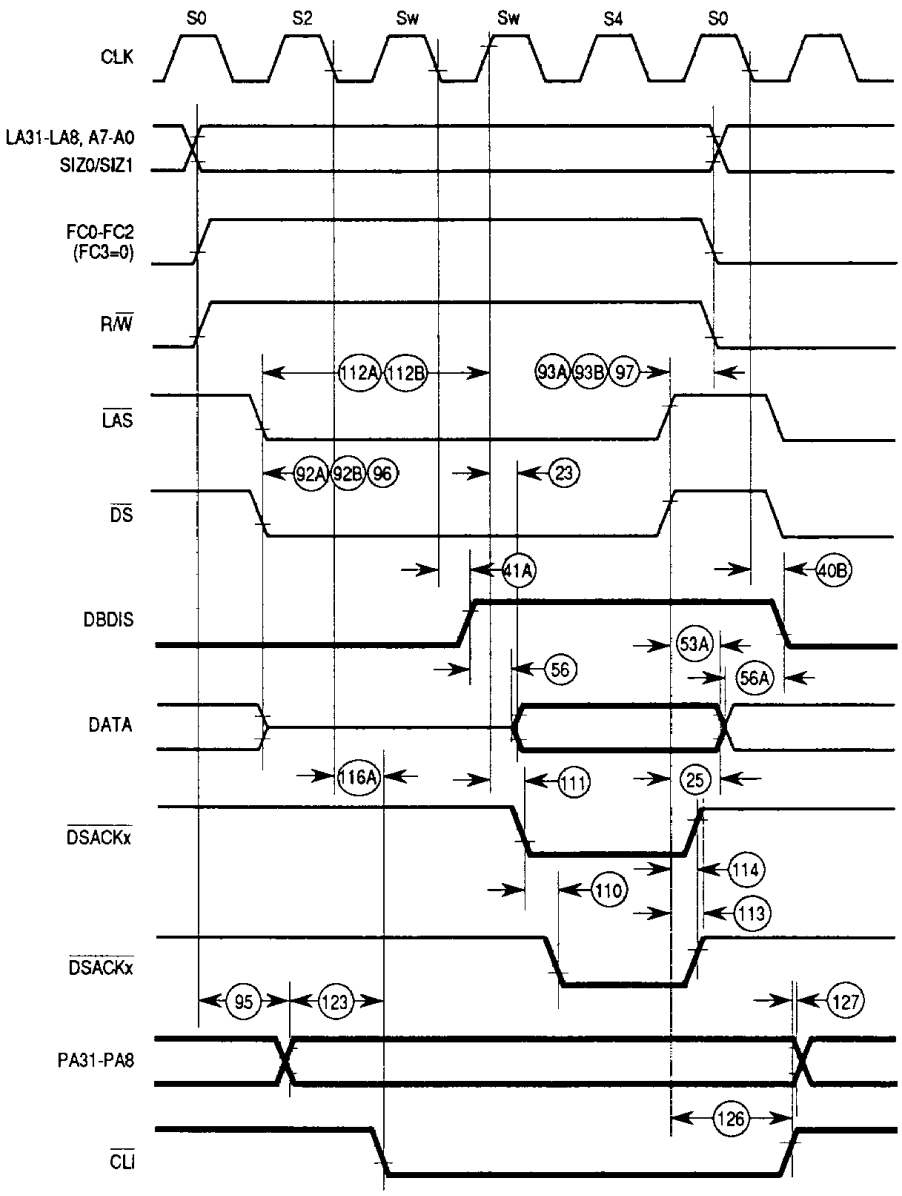


Figure 13. Logical Bus Arbitration by Asynchronous Master Timing Diagram



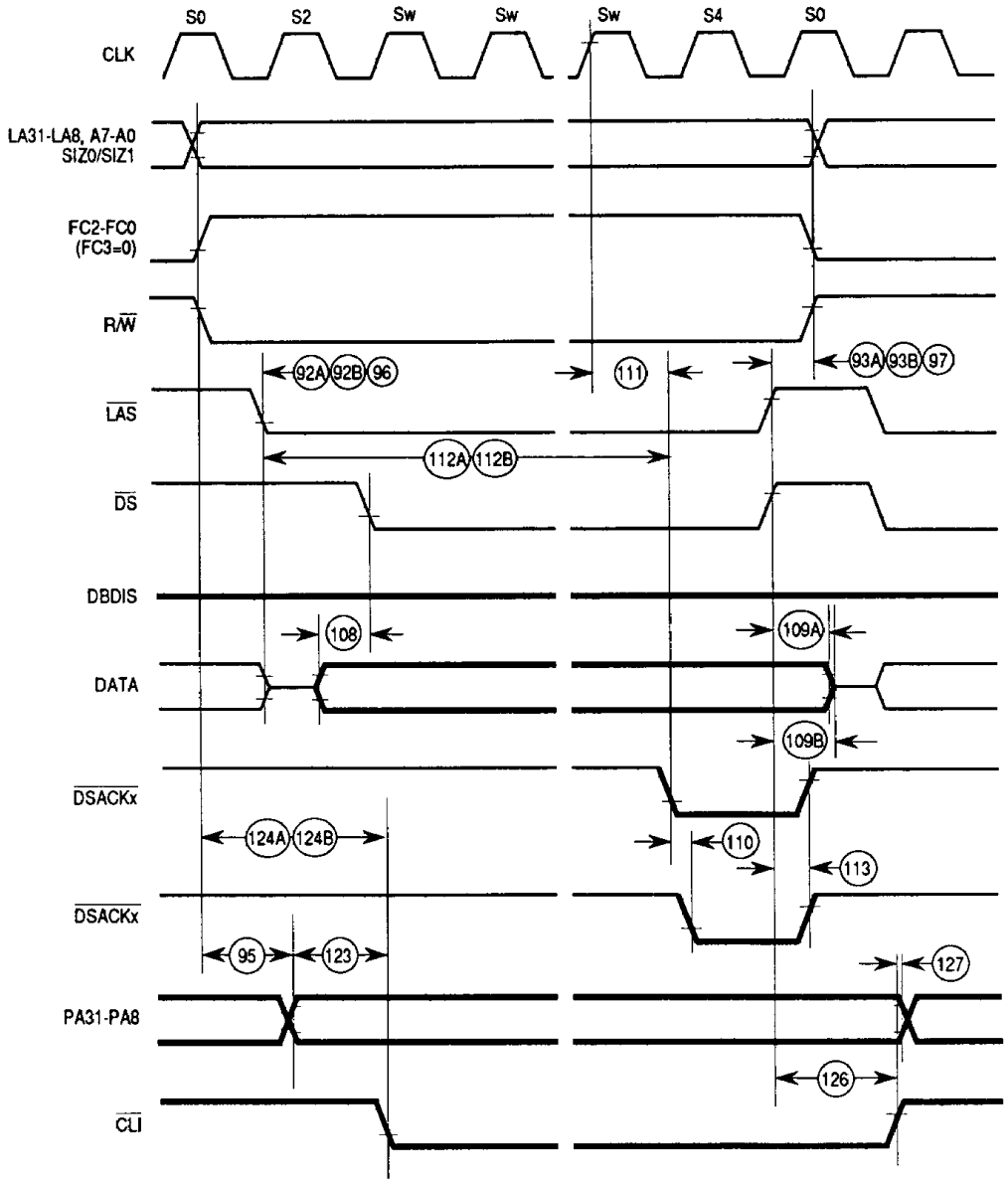
NOTE: This depiction of LAS is inconsistant with the MC68851 bus cycle depicted, but is included to show signed relationships when the MC68851 is performing a translation

Figure 14. Physical Bus Arbitration Timing Diagram



NOTE: $\overline{CLI}$ asserted by MC68851.

Figure 15. CPU Space Read from MC68851 or Other Coprocessor Timing Diagram



NOTE: CLI asserted by MC68851.

Figure 16. CPU Space Write to MC68851 or Other Coprocessor Timing Diagram

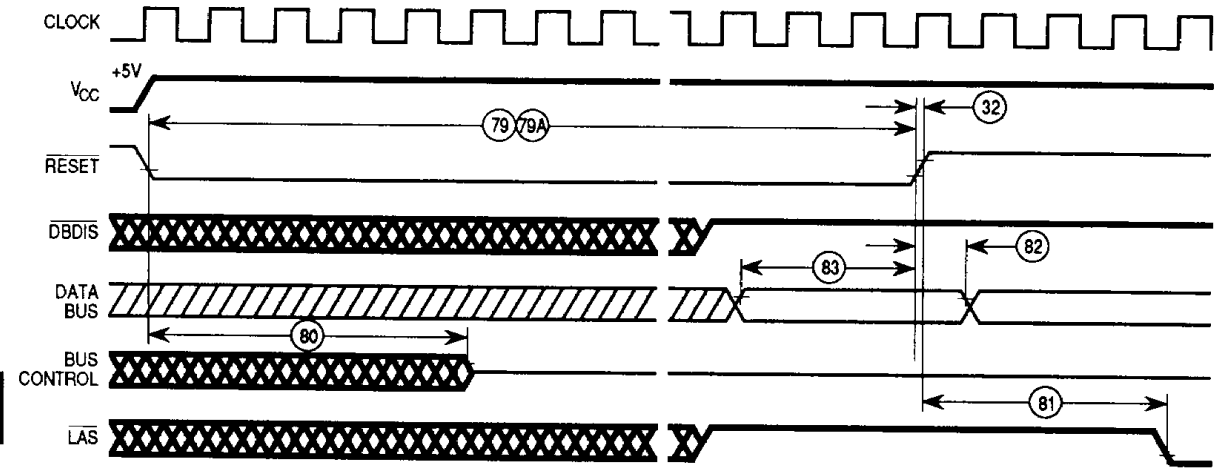


Figure 17. Reset and Mode Select Timing Diagram

124-LEAD PIN GRID ARRAY

The V_{CC} and GND pins are separated into four groups to provide individual power supply connections for the address bus buffers, data bus buffers, and all other output buffers and internal logic.

Pin Group	V_{CC}	GND
Physical Address	D5, G2, J4	E4, G3, K5
Logical Address, Internal Logic	M7	L7
D0–D31	E10,0G12, K9	D9, G11, J10
Internal Logic, Clocks	B7	C7