Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Obsolete
Core Processor	CPU32+
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	25MHz
Co-Processors/DSP	Communications; CPM
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10Mbps (1)
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	-
Package / Case	357-BGA
Supplier Device Package	357-PBGA (25x25)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68360czp25l

BCLRO—This active-low open-drain output indicates that one of the QUICC internal bus masters is requesting the external bus master to release the bus.

CONFIG1—See 2.1.13 Initial Configuration Pins (CONFIG) for the description.

RAS2—See 2.1.5.1 Chip Select/Row Address Select (CS6–CS0/RAS6–RAS0) for the description.

2.1.9 System Control Signals

The QUICC uses these signals to recover from an exception. Refer to Section 4 Bus Operation for more information on these signals.

2.1.9.1 SOFT RESET (RESETS). This active-low, open-drain, bidirectional signal is used to initiate reset. An external reset signal (as well as a reset from the SIM60) resets the QUICC as well as all external devices. A reset signal from the CPU32+ (asserted as part of the RESET instruction) resets external devices only—the internal state of the CPU32+ is not affected; other on-chip modules are reset, but the configuration is not altered. When asserted by the QUICC, this signal is guaranteed to be asserted for a minimum of 512 clock cycles. For more information see 4.7 Reset Operation.

2.1.9.2 HARD RESET (RESETH). This active-low, open-drain, bidirectional signal is used to initiate reset. An external hard reset signal (as well as an hard reset from the SIM60) resets the QUICC as well as all external devices and the internal state of the CPU32+; other on-chip modules are reset as well as the QUICC configuration. When asserted by the QUICC, this signal is guaranteed to be asserted for a minimum of 512 clock cycles. For more information see 4.7 Reset Operation.

During a hard reset, the address, data, and bus control pins are all three-stated. The BG pin output is the same as that on the BR input. The general-purpose I/O pins are all configured as inputs. The NC4–NC1 pins are undefined outputs. The XTAL, CLKO1, and CLKO2 pins are active outputs, except for CLKO1 which does not oscillate while the on-chip PLL is attaining a lock. The RESETS pin is an output.

2.1.9.3 HALT (HALT). This active-low, open-drain, bidirectional signal is asserted to suspend external bus activity, to request a retry when used with BERR, or to perform a single-step operation. As an output, HALT indicates a double bus fault by the CPU32+.

2.1.9.4 BUS ERROR (BERR). This active-low, open-drain, bidirectional signal indicates that an invalid bus operation is being attempted or, when used with HALT, that the bus master should retry the current cycle.

2.1.10 Clock Signals

These signals are used by the QUICC for controlling or generating the system clocks. Refer to Section 6 System Integration Module (SIM60) for more information on these clock signals.

2.1.10.1 SYSTEM CLOCK OUTPUTS (CLKO2–CLKO1). These output signals reflect the general system clock and are used as the bus timing reference by external devices. CLKO1

5.3.1 M68000 Family Compatibility

It is the philosophy of the M68000 Family that all user-mode programs should execute unchanged on a more advanced processor and that supervisor-mode programs and exception handlers should require only minimal alteration.

The CPU32+ can be thought of as an intermediate member of the M68000 family. Object code from an MC68000 or MC68010 may be executed on the CPU32+, and many of the instruction and addressing mode extensions of the MC68020 are also supported.

5.3.1.1 NEW INSTRUCTIONS. Two instructions have been added to the M68000 instruction set: LPSTOP and TBL.

5.3.1.2 LOW-POWER STOP (LPSTOP). In applications where power consumption is a consideration, the CPU32+ can force the device into a low-power standby mode when immediate processing is not required. The low-power mode is entered by executing the LPSTOP instruction. The processor remains in this mode until a user-specified or higher level interrupt or a reset occurs.

5.3.1.3 TABLE LOOKUP AND INTERPOLATE (TBL). To maximize throughput for real-time applications, reference data is often precalculated and stored in memory for quick access. The storage of sufficient data points can require an inordinate amount of memory. The TBL instruction uses linear interpolation to recover intermediate values from a sample of data points, thus conserving memory.

When the TBL instruction is executed, the CPU32+ looks up two table entries bounding the desired result and performs a linear interpolation between them. Byte, word, and long-word operand sizes are supported. The result can be rounded according to a round-to-nearest algorithm or returned unrounded along with the fractional portion of the calculated result (byte and word results only). This extra precision can be used to reduce cumulative error in complex calculations. See 5.3.4 Using the TBL Instructions for examples.

5.3.1.4 UNIMPLEMENTED INSTRUCTIONS. The ability to trap on unimplemented instructions allows user-supplied code to emulate unimplemented capabilities or to define special-purpose functions. However, Motorola reserves the right to use all currently unimplemented instruction operation codes for future M68000 enhancements. See 5.5.2.8 Illegal or Unimplemented Instructions for more details.

5.3.2 Instruction Format and Notation

All instructions consist of at least one word. Some instructions can have as many as seven words, as shown in Figure 5-6. The first word of the instruction, called the operation word, specifies instruction length and the operation to be performed. The remaining words, called extension words, further specify the instruction and operands. These words may be immediate operands, extensions to the effective address mode specified in the operation word, branch displacements, bit number, special register specifications, trap operands, or argument counts.

tiated while an instruction is executing. Several bus error stack format organizations are utilized to provide additional information regarding the nature of the fault.

First, any register altered by a faulted-instruction EA calculation is restored to its initial value. Then a special status word (SSW) is placed on the stack. The SSW contains specific information about the aborted access—size, type of access (read or write), bus cycle type, and function code. Finally, fault address, bus error exception vector number, PC value, and a copy of the SR are saved.

If a bus error occurs during exception processing for a bus error, an address error, a reset, or while the processor is loading stack information during RTE execution, the processor halts. This simplifies isolation of catastrophic system failure by preventing processor interaction with stacks and memory. Only assertion of $\overline{\text{RESET}}$ can restart a halted processor.

5.5.2.3 ADDRESS ERROR. Address error exceptions occur when the processor attempts to access an instruction, word operand, or long-word operand at an odd address. The effect is much the same as an internally generated bus error. The exception processing sequence is the same as that for bus error, except that the vector number refers to the address error exception vector.

Address error exception processing begins when the processor attempts to use information from the aborted bus cycle. If the aborted cycle is a data space access, exception processing begins when the processor attempts to use the data, except in the case of a released operand write. Released write exceptions are delayed until the next instruction boundary or attempted operand access.

An address exception on a branch to an odd address is delayed until the PC is changed. No exception occurs if the branch is not taken. In this case, the fault address and return PC value placed in the exception stack frame are the odd address, and the current instruction PC points to the instruction that caused the exception.

If an address error occurs during exception processing for a bus error, another address error, or a reset, the processor halts.

5.5.2.4 INSTRUCTION TRAPS. Traps are exceptions caused by instructions. They arise from either processor recognition of abnormal conditions during instruction execution or from use of specific trapping instructions. Traps are generally used to handle abnormal conditions that arise in control routines.

The TRAP instruction, which always forces an exception, is useful for implementing system calls for user programs. The TRAPcc, TRAPV, CHK, and CHK2 instructions force exceptions when a program detects a run-time error. The DIVS and DIVU instructions force an exception if a division operation is attempted with a divisor of zero.

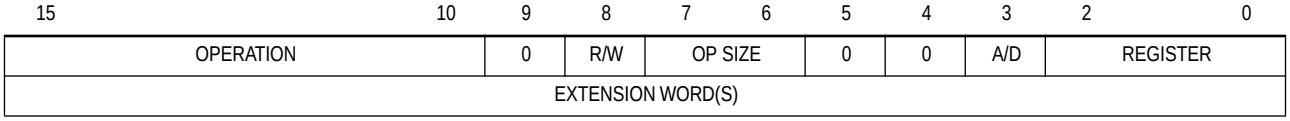
Exception processing for traps follows the regular sequence. If tracing is enabled when an instruction that causes a trap begins execution, a trace exception will be generated by the instruction, but the trap handler routine will not be traced. (The trap exception will be processed first, then the trace exception.)

BKPT_TAG should be timed to the bus cycles since it is not latched. If extended past the assertion of FREEZE, the negation of BKPT_TAG appears to the CPU32+ as the first DSCLK.

DSCLK, the gated serial clock, is normally high, but it pulses low for each bit to be transferred. At the end of the seventeenth clock period, it remains high until the start of the next transmission. Clock frequency is implementation dependent and may range from DC to the maximum specified frequency. Although performance considerations might dictate a hardware implementation, software solutions can be used provided serial bus timing is maintained.

5.6.2.8 COMMAND SET. The following paragraphs describe the command set available in BDM.

5.6.2.8.1 Command Format. The following standard bit .command format is utilized by all BDM commands.



Bits 15–10—Operation Field

The operation field specifies the commands. This 6-bit field provides for a maximum of 64 unique commands.

R/W Field

The R/W field specifies the direction of operand transfer. When the bit is set, the transfer is from theCPU to the development system. When the bit is cleared, data is written to the CPU or to memory from the development system.

Operand Size

For sized operations, this field specifies the operand data size. All addresses are expressed as 32-bit absolute values. The size field is encoded as listed in Table 5-22..

Table 5-22. Size Field Encoding

Encoding	Operand Size
00	Byte
01	Word
10	Long
11	Reserved

Address/Data (A/D) Field

The A/D field is used by commands that operate on address and data registers. It determines whether the register field specifies a data or address register. One indicates an address register; zero indicates a data register. For other commands, this field may be interpreted differently.

Instruction		Head	Tail	Cycles
ABCD	Dn, Dm	2	0	4(0/1/0)
ABCD	–(An), –(Am)	2	2	12(2/1/1)
SBCD	Dn, Dm	2	0	4(0/1/0)
SBCD	–(An), –(Am)	2	2	12(2/1/1)
ADDX	Dn, Dm	0	0	2(0/1/0)
ADDX	–(An), –(Am)	2	2	10(2/1/1)
SUBX	Dn, Dm	0	0	2(0/1/0)
SUBX	–(An), –(Am)	2	2	10(2/1/1)
CMPM	(An)+, (Am)+	1	0	8(2/1/0)

5.7.2.8 SINGLE OPERAND INSTRUCTIONS. The single operand instruction table indicates the number of clock periods needed for the processor to perform the specified operation using the specified addressing mode. The total number of clock cycles is outside the parentheses. The numbers inside parentheses (r/p/w) are included in the total clock cycle number. All timing data assumes two-clock reads and writes.

Instruction		Head	Tail	Cycles
CLR	Dn	0	0	2(0/1/0)
CLR	⟨CEA⟩	0	2	4(0/1/X)
NEG	Dn	0	0	2(0/1/0)
NEG	⟨FEA⟩	0	3	5(0/1/X)
NEGX	Dn	0	0	2(0/1/0)
NEGX	⟨FEA⟩	0	3	5(0/1/X)
NOT	Dn	0	0	2(0/1/0)
NOT	⟨FEA⟩	0	3	5(0/1/X)
EXT	Dn	0	0	2(0/1/0)
NBCD	Dn	2	0	4(0/1/0)
NBCD	⟨FEA⟩	0	2	6(0/1/1)
Scc	Dn	2	0	4(0/1/0)
Scc	⟨CEA⟩	2	2	6(0/1/1)
TAS	Dn	4	0	6(0/1/0)
TAS	⟨CEA⟩	1	0	10(0/1/1)
TST	⟨FEA⟩	0	0	2(0/1/0)

X = There is one bus cycle for byte and word operands and two bus cycles for long-word operands. For long-word bus cycles, add two clocks to the tail and to the number of cycles.
Timing is calculated with the CPU32+ in 16-bit mode

5.7.2.9 SHIFT/ROTATE INSTRUCTIONS. The shift/rotate instruction table indicates the number of clock periods needed for the processor to perform the specified operation on the given addressing mode. Footnotes indicate when to account for the appropriate EA times. The number of bits shifted does not affect the execution time, unless noted. The total num-

0 = The CP is ready to receive a new command.
1 = The CR contains a command that the CP is currently processing. The CP clears this bit at the end of the command execution or after reset.

7.2.1 Command Register Examples

To perform a complete reset of the CP, the value $8001 should be written to the CR. Following this command, the CR will return the value $0000 in two clocks.

To execute an ENTER HUNT MODE command to SCC3, the value $0381 should be written to the CR. While the command is executing, the CR will return the value $0381. When the command has been completely executed, the CR will return the value $0380.

7.2.2 Command Execution Latency

The worst-case command execution latency is 120 clocks. The typical command execution latency is about 40 clocks.

7.3 DUAL-PORT RAM

The CPM has 2560 bytes of static RAM configured as dual-port memory. The dual-port RAM memory map is shown in Figure 7-2, and a block diagram is shown in Figure 7-3.

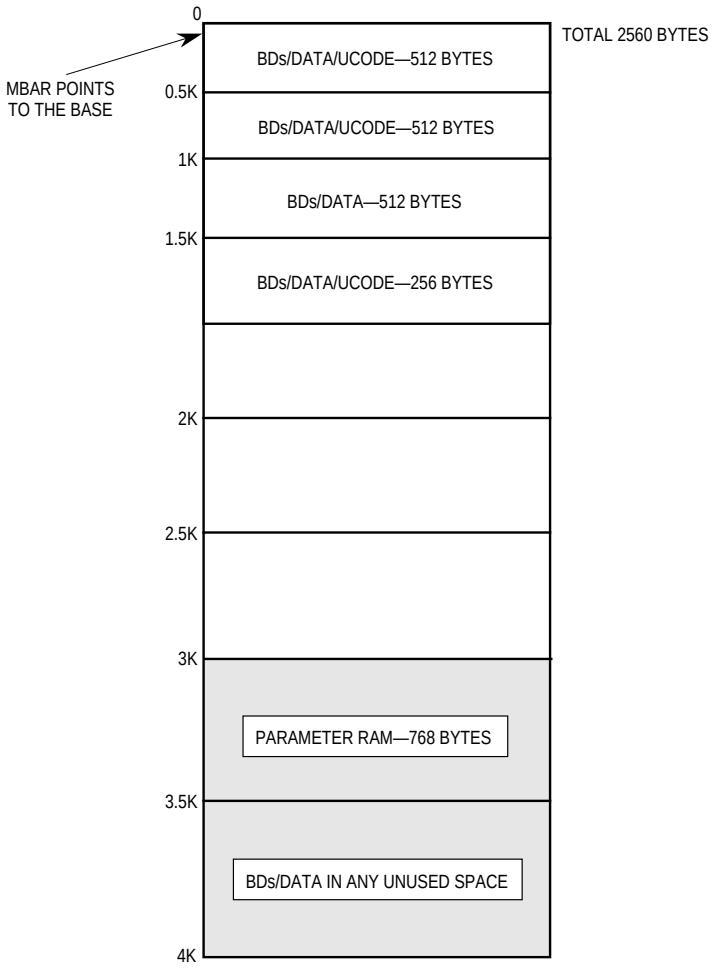


Figure 7-2. Dual-Port RAM Memory Map

ICLK—Input Clock Source for the Timer

00 = Internally cascaded input.
For TMR1, the timer 1 input is the output of timer 2.
For TMR3, the timer 3 input is the output of timer 4.
For TMR2 and TMR4, this selection means no input clock is provided to the timer.
01 = Internal general system clock.
10 = Internal general system clock divided by 16.
11 = Corresponding TIN pin: TIN1, TIN2, TIN3, or TIN4 (falling edge).

GE—Gate Enable

0 = The $\overline{\text{TGATE}}$ signal is ignored.
1 = The $\overline{\text{TGATE}}$ signal is used to control the timer.

7.5.2.4 TIMER REFERENCE REGISTERS (TRR1, TRR2, TRR3, TRR4). Each TRR is a 16-bit, memory-mapped, read-write register containing the reference value for the timeout. TRR1–TRR4 are set to all ones by reset. The reference value is not reached until TCN increments to equal TRR.

7.5.2.5 TIMER CAPTURE REGISTERS (TCR1, TCR2, TCR3, TCR4). Each TCR is a 16-bit register used to latch the value of the counter. TCR1–TCR4 appear as memory- mapped, read-only registers to the user. TCR1–TCR4 are cleared by reset.

7.5.2.6 TIMER COUNTER (TCN1, TCN2, TCN3, TCN4). Each TCN is a 16-bit, memory-mapped, read-write up-counter. A read cycle to TCN1–TCN4 yields the current value of the timer, but does not affect the counting operation. A write cycle to TCN1–TCN4 sets the register to the written value, causing its corresponding prescaler to be reset.

NOTE

Write operation to this register while the timer is not running may not update the register correctry. User should always use timer refrence register to define desired count value.

7.5.2.7 TIMER EVENT REGISTERS (TER1, TER2, TER3, TER4). Each TER is a 16-bit register used to report events recognized by any of the timers. On recognition of an output reference event, the timer sets the REF bit in the TER, regardless of the corresponding ORI in the TMR. The capture event will be set only if enabled by the CE bits in the TMR. TER1–TER4, which appear to the user as memory-mapped registers, may be read at any time.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—														REF	CAP

A bit is reset by writing a one to that bit (writing a zero does not affect a bit's value). More than one bit may be reset at a time. Both bits must be reset before the timer will negate the interrupt to the CPM interrupt controller. This register is cleared by reset.

Bits 15–2—Reserved

7.8 SERIAL INTERFACE WITH TIME SLOT ASSIGNER

The SI connects the physical layer serial lines to the four SCCs and two SMCs (see Figure 7-19). In its simplest configuration, the SI allows the four SCCs and two SMCs to be connected their own set of individual pins. Each SCC or SMC that connects to the external world in this way is said to connect to an NMSI. In the NMSI configuration, the SI provides a flexible clocking assignment for each SCC and SMC from a bank of external clock pins and/or internal baud rate generators.

However, the main feature of the SI is its TSA. The TSA allows any combination of SCCs and SMCs to multiplex their data together on either one or two TDM channels. TDM is used in this manual as the generic term that describes any serial channel that is divided into channels separated by time. Common examples of TDMs are the T1 lines in Japan and the United States and the CEPT lines in Europe.

Even if the TSA is not used in its intended capacity, it may still be used to generate complex waveforms on four output pins. For instance, these pins can be programmed by the TSA to implement stepper motor control or variable duty cycle and period control on these pins. Any programmed configuration may be changed on the fly.

7.8.1 SI Key Features

The two major features of the SI are the TSA and the NMSI. The TSA contains the following features:

- Can Connect to Two Independent TDM channels. Each TDM May Be One of the Following:

  —T1 or CEPT Line
  —PCM Highway
  —ISDN Primary Rate
  —ISDN Basic Rate—IDL
  —ISDN Basic Rate—GCI
  —User-Defined Interfaces
- Independent, Programmable Transmit and Receive Routing Paths
- Independent Transmit and Receive Frame Syncs Allowed
- Independent Transmit and Receive Clocks Allowed
- Selection of Rising/Falling Clock Edges for the Frame Sync and Data Bits
- Supports 1× and 2× Input Clocks (i.e., 1 or 2 Clocks per Data Bit)
- Selectable Delay (0–3 Bits) Between Frame Sync and Frame Start
- Four Programmable Strobe Outputs and Two (2×) Clock Output Pins
- 1- or 8-Bit Resolution in Routing, Masking, and Strobe Selection
- Supports Frames Up to 8192 Bits Long
- Internal Routing and Strobe Selection Can Be Dynamically Programmed
- Supports Automatic Echo and Loopback Mode for Each TDM

receiver is not in hunt mode and a SYNC character has been received, the receiver will discard this character if the valid bit is set.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V	0	0	0	0	0	0	0	SYNC							

NOTE

When using 7-bit characters with parity, the parity bit should be included in the SYNC register value.

7.10.20.8 BDLE-BISYNC DLE REGISTER. The 16-bit, memory-mapped, read-write BDLE register is used to define the BISYNC stripping and insertion of the DLE character. When the BISYNC controller is in transparent mode and an underrun occurs during message transmission, the BISYNC controller inserts DLE-SYNC pairs until the next data buffer is available for transmission.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V	0	0	0	0	0	0	0	DLE							

When the BISYNC receiver is in transparent mode and a DLE character is received, the receiver discards this character and excludes it from the BCS if the valid bit is set. If the second (next) character is a SYNC character, the BISYNC controller discards it and excludes it from the BCS. If the second character is a DLE, the BISYNC controller will write it to the buffer and include it in the BCS. If the character is not a DLE or SYNC, the BISYNC controller will examine the control characters table and act accordingly. If the character is not in the table, the buffer will be closed with the DLE follow character error (DLE) bit set. If the valid bit is not set, the receiver will treat the character as a normal character.

NOTE

When using 7-bit characters with parity, the parity bit should be included in the DLE register value.

7.10.20.9 TRANSMITTING AND RECEIVING THE SYNCHRONIZATION SEQUENCE. The BISYNC channel can be programmed to transmit and receive a synchronization pattern. The pattern is defined in the DSR. The length of the SYNC pattern is defined in the SYNL bits in the GSMR. The receiver synchronizes on the synchronization pattern that is located in the DSR. If the SYNL bits specify a non-zero synchronization pattern, then the transmitter sends the entire contents of the DSR prior to each frame, starting with the LSB first. Thus, the user may wish to repeat the desired SYNC pattern in the other DSR bits as well.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
4-BIT SYNC															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8-BIT SYNC															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-BIT SYNC															

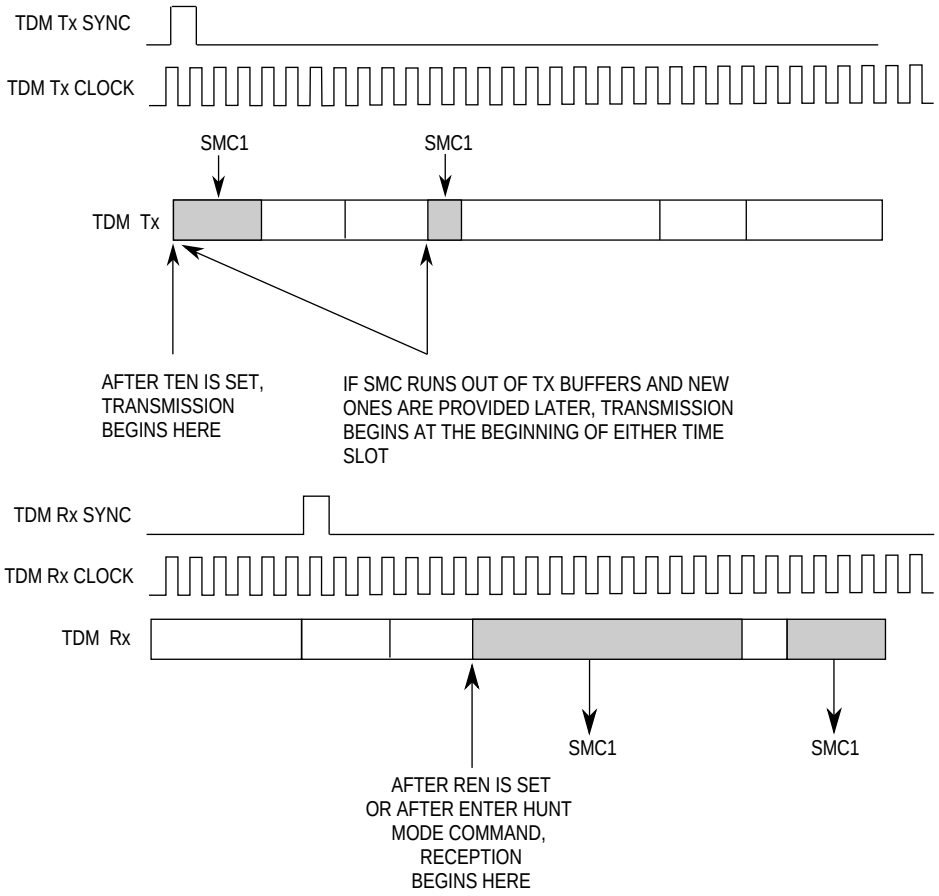


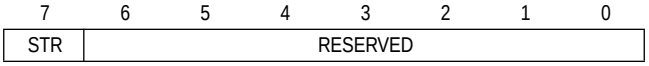
Figure 7-79. Synchronization with the TSA

Once the REN bit is set in SMCMR, the first time slot after frame sync causes the SMC receiver to achieve synchronization. Data will begin to be received immediately, but only during the defined receive time slots. The receiver will continue to receive data during its defined time slots until the REN bit is cleared by the user. If the ENTER HUNT MODE command is executed, the receiver will lose synchronization, close the current buffer, and resynchronize to the first time slot after the frame sync.

Once the TEN bit is set in SMCMR, the SMC waits for the transmit FIFO to be loaded, before attempting to achieve synchronization. Once the transmit FIFO is loaded, synchronization and transmission begin on the first bit of the first time slot after the frame sync. Idles (ones) are transmitted until data begins transmission.

If the SMC runs out of transmit buffers and a new transmit buffer is provided later, idles will be transmitted during the gap between data buffers, and data transmission from the later data buffer will begin at the beginning of an SMC time slot, but not necessarily the first time slot after the frame sync. Thus, if the user wishes to maintain a certain bit alignment beginning with the first time slot, the user should always make sure that at least one Tx BD is always ready and that no underruns occur. Otherwise, the SMC transmitter should be disabled and reenabled. See 7.11.5 Disabling the SMCs on the Fly for a description of how to

7.12.5.2 SPI COMMAND REGISTER (SPCOM). The SPCOM is an 8-bit read-write register that is used to start SPI operation.



Bits 6–0—Reserved.

These bits should be written with zeros by the user.

STR—Start Transmit

When the SPI is configured as a master, setting the STR bit to one causes the SPI controller to start the transmission and reception of data from/to the SPI transmit/receive buffers (if they are configured as ready by the user).

When the SPI is configured as a slave, setting the STR bit to one when the SPI is idle (between transfers) causes the SPI to load the transmit data register from the SPI transmit buffer and start transmission as soon as the next SPI input clocks and select signal are received.

The STR bit is cleared automatically after one system clock cycle.

7.12.5.3 SPI PARAMETER RAM MEMORY MAP. The SPI parameter RAM area (see Table 7-16) begins at the SPI base address. This area is used for the general SPI parameters. The user will notice that it is similar to the SCC general-purpose parameter RAM.

Table 7-16. SPI Parameter RAM Memory Map

Address	Name	Width	Description
SPI Base + 00	**RBASE**	Word	Rx BD Base Address
SPI Base+ 02	**TBASE**	Word	Tx BD Base Address
SPI Base+ 04	**RFCR**	Byte	Rx Function Code
SPI Base+ 05	**TFCR**	Byte	Tx Function Code
SPI Base+ 06	**MRBLR**	Word	Maximum Receive Buffer Length
SPI Base+ 08	RSTATE	Long	Rx Internal State
SPI Base+ 0C		Long	Rx Internal Data Pointer
SPI Base+ 10	RBPTR	Word	Rx BD Pointer
SPI Base+ 12		Word	Rx Internal Byte Count
SPI Base+ 14		Long	Rx Temp
SPI Base+ 18	TSTATE	Long	Tx Internal State
SPI Base+ 1C		Long	Tx Internal Data Pointer
SPI Base+ 20	TBPTR	Word	Tx BD Pointer
SPI Base+ 22		Word	Tx Internal Byte Count
SPI Base+ 24		Long	Tx Temp

NOTE: The items in boldface should be initialized by the user.

Certain parameter RAM values (marked in boldface) need to be initialized by the user before the SPI is enabled; other values are initialized by the CP. Once initialized, the parameter

7	6	5	4	3	2	1	0
RES	RES	RES	MOT	FC3–FC0			

FC3-0 —Function Code 3-0

These bits contain the function code value used during this SDMA channel's memory accesses. It is suggested that the user write bit FC3 with a one to identify this SDMA channel access as a DMA-type access. Example: FC3-FC0 = 1000 (binary). Do not write the value 0111 (binary) to these bits.

MOT—Motorola

This bit should be set by the user to achieve normal operation.

0 = DEC (and Intel) convention is used for byte ordering. Swapped operation. Also called little-endian byte ordering. The bytes stored in each buffer word are reversed as compared to the Motorola mode.

1 = Motorola byte ordering. Normal operation. Also called big-endian byte ordering. As data is received from the serial line and put into the buffer, the most significant byte of the buffer word contains data received earlier than the least significant byte of the same buffer word.

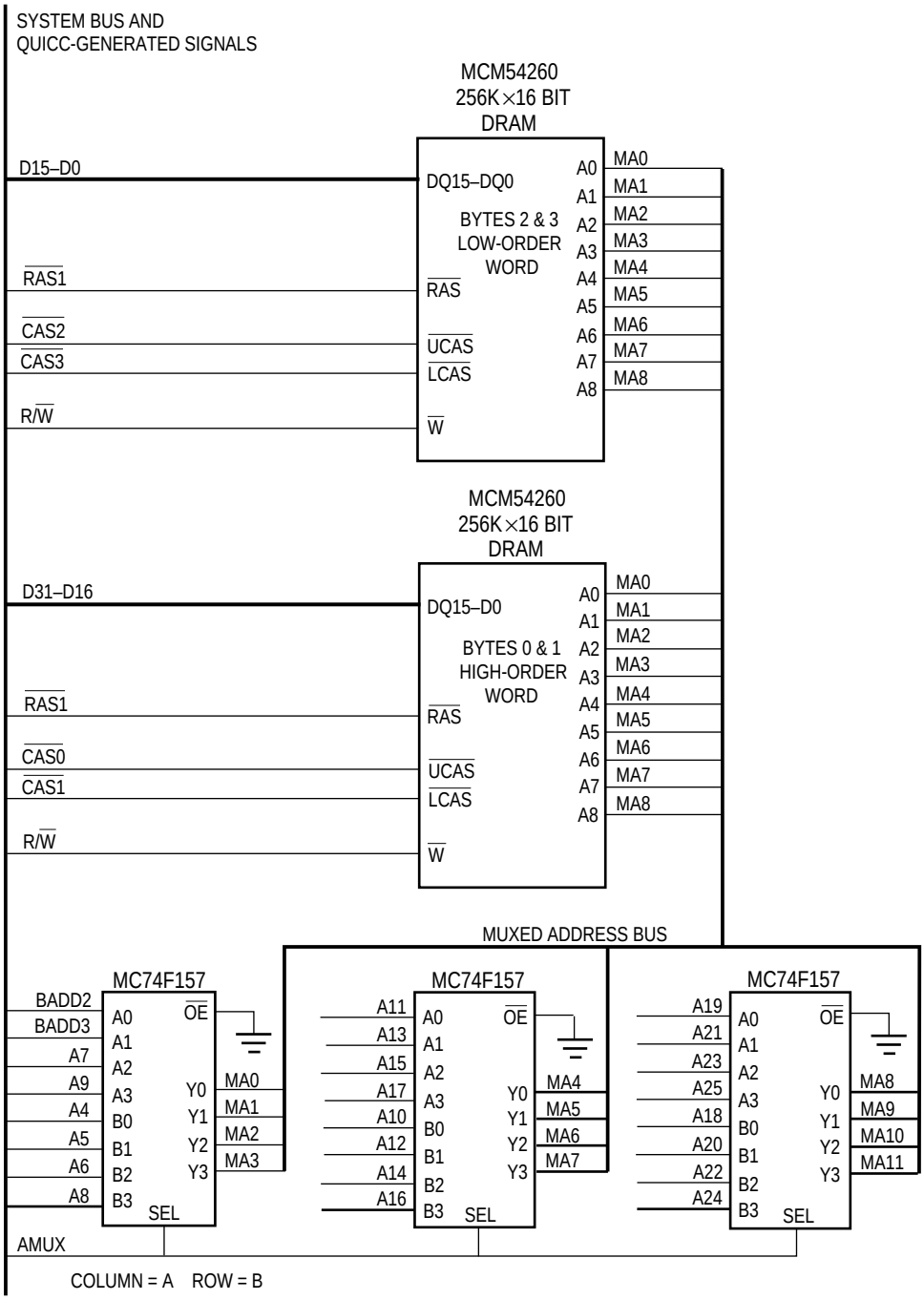
Res—Reserved. Should be set to zero by the user.

7.13.8.16 RECEIVER BUFFER DESCRIPTOR POINTER (RBPTR). The receiver buffer descriptor pointer (RBPTR) points to the next BD that the receiver will transfer data to when it is in IDLE state, or to the current BD during frame reception. After a reset or when the end of BD table is reached, the CP initializes this pointer to the value programmed in the RBASE entry. Although RBPTR need never be written by the user in most applications, it may be modified by the user when the receiver is disabled.

7.13.8.17 CENTRONICS RECEIVER PROGRAMMING MODEL. The host configures the PIP to operate as a Centronics controller by programming the PIP Configuration register (PIPC). Timing attributes (ACK pulse width and the timing between ACK and BUSY) are set by programming the PIP Timing Parameters register (PTPR). The receive errors are reported through the Rx BD.

7.13.8.18 CENTRONICS CONTROL CHARACTERS. The Centronics receiver has the capability to recognize special control characters. These characters may be used when the Centronics functions in a message oriented environment. Up to eight control characters may be defined by the user in the Control Characters Table. Each of these characters may be either written to the receive buffer (upon which the buffer is closed and a new receive buffer taken) or rejected. If rejected, the character is written to the Received Control Character Register (RCCR) in internal RAM and a maskable interrupt is generated. This method is useful for notifying the user of the arrival of control characters that are not part of the received messages.

The Centronics receiver uses a table of 16-bit entries to support control character recognition. Each entry consists of the control character, a valid bit, and a reject character bit.



NOTE: MA11–MA9 not required but allows future expansion.

Figure 9-16. 1-Mbyte DRAM Bank—32 Bits Wide

9.4.3 Software Configuration

The following paragraphs discuss a number of key points for to a software writer desiring to initialize the system. The points discussed are those that are required to enable the previously mentioned hardware configurations.

if DRAM is used elsewhere in the system. The QUICC does not support bursting by the MC68EC030.

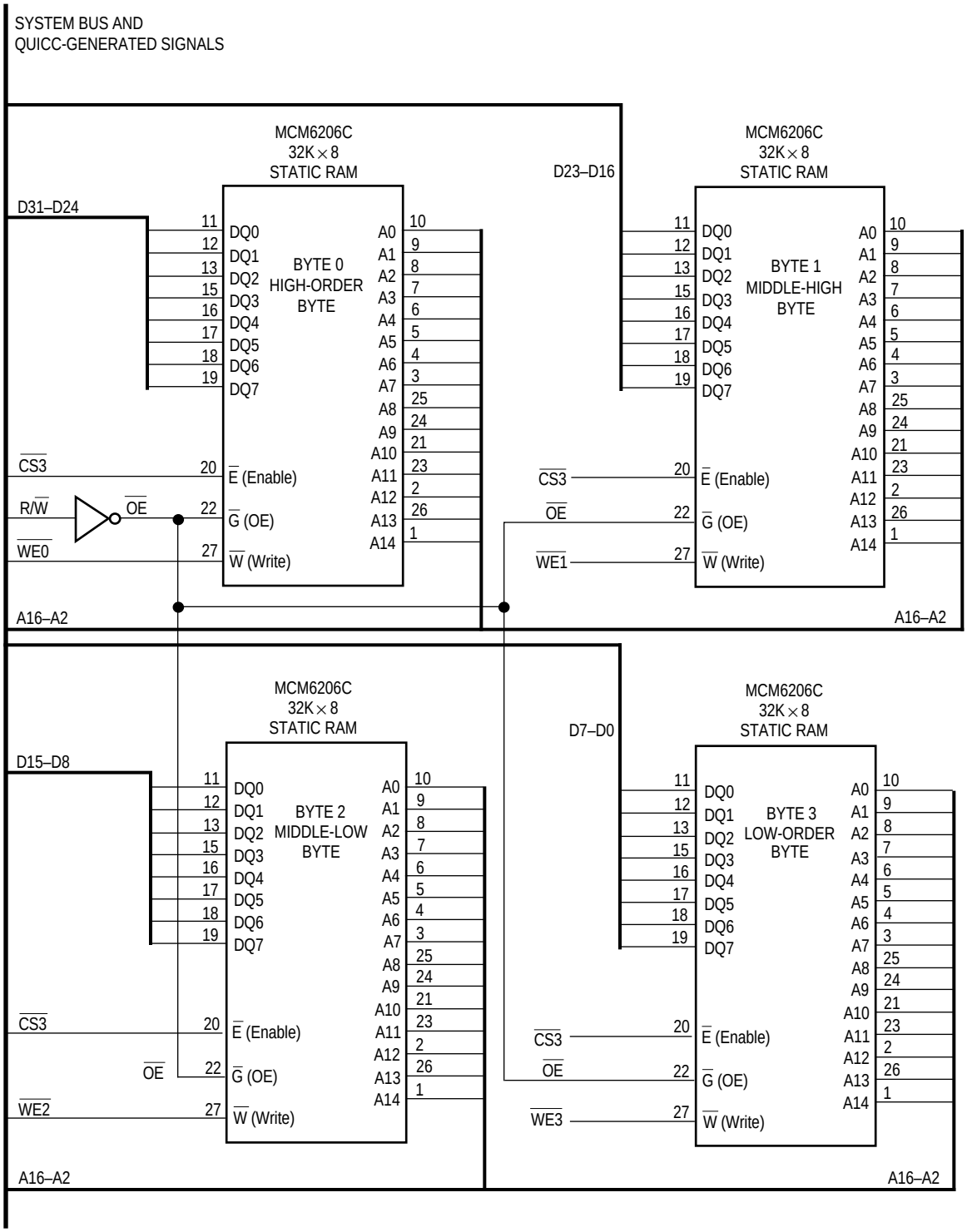


Figure 9-30. 128-Kbyte Static RAM Bank—32 Bits Wide

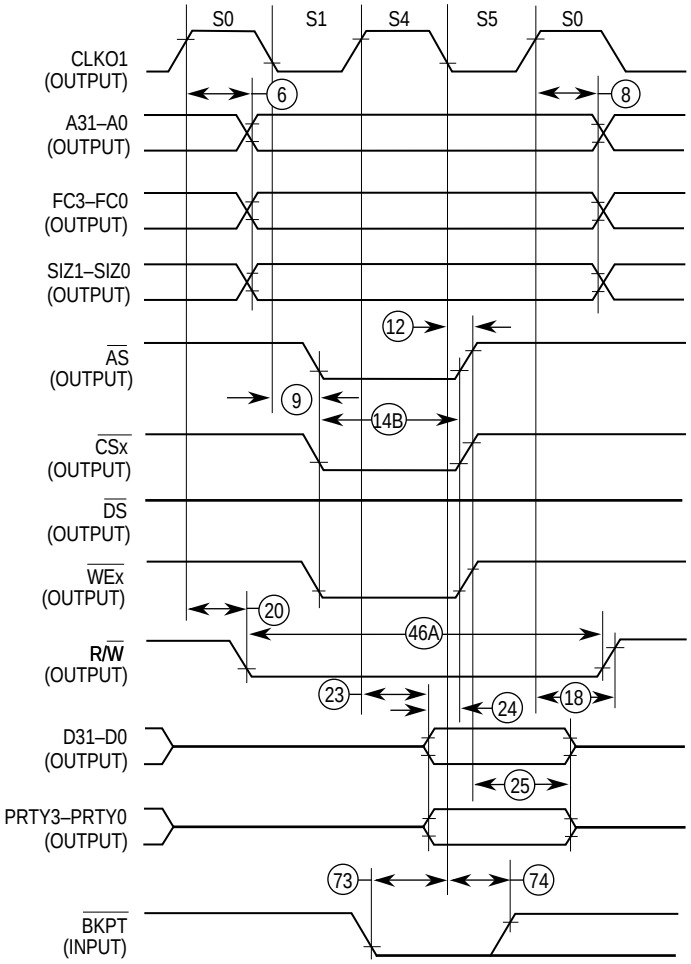
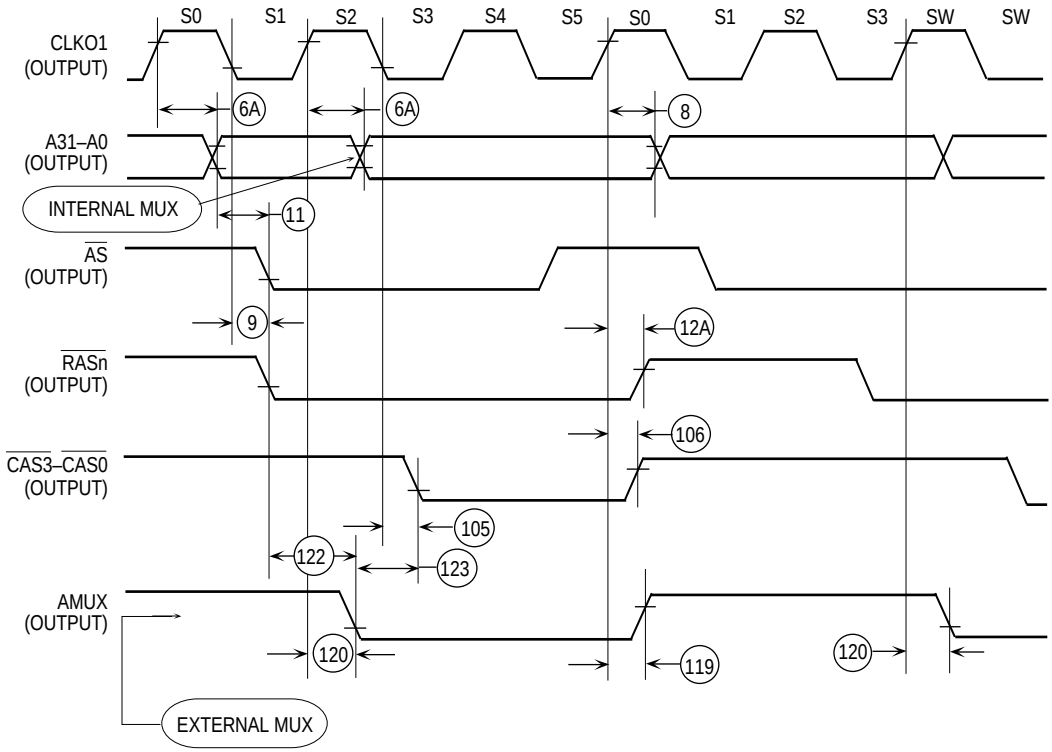


Figure 10-9. Fast Termination Write Cycle



NOTE: All timing is shown with respect to 0.8-V and 2.0-V levels.

Figure 10-24. DRAM: Page-Mode—Page-Miss

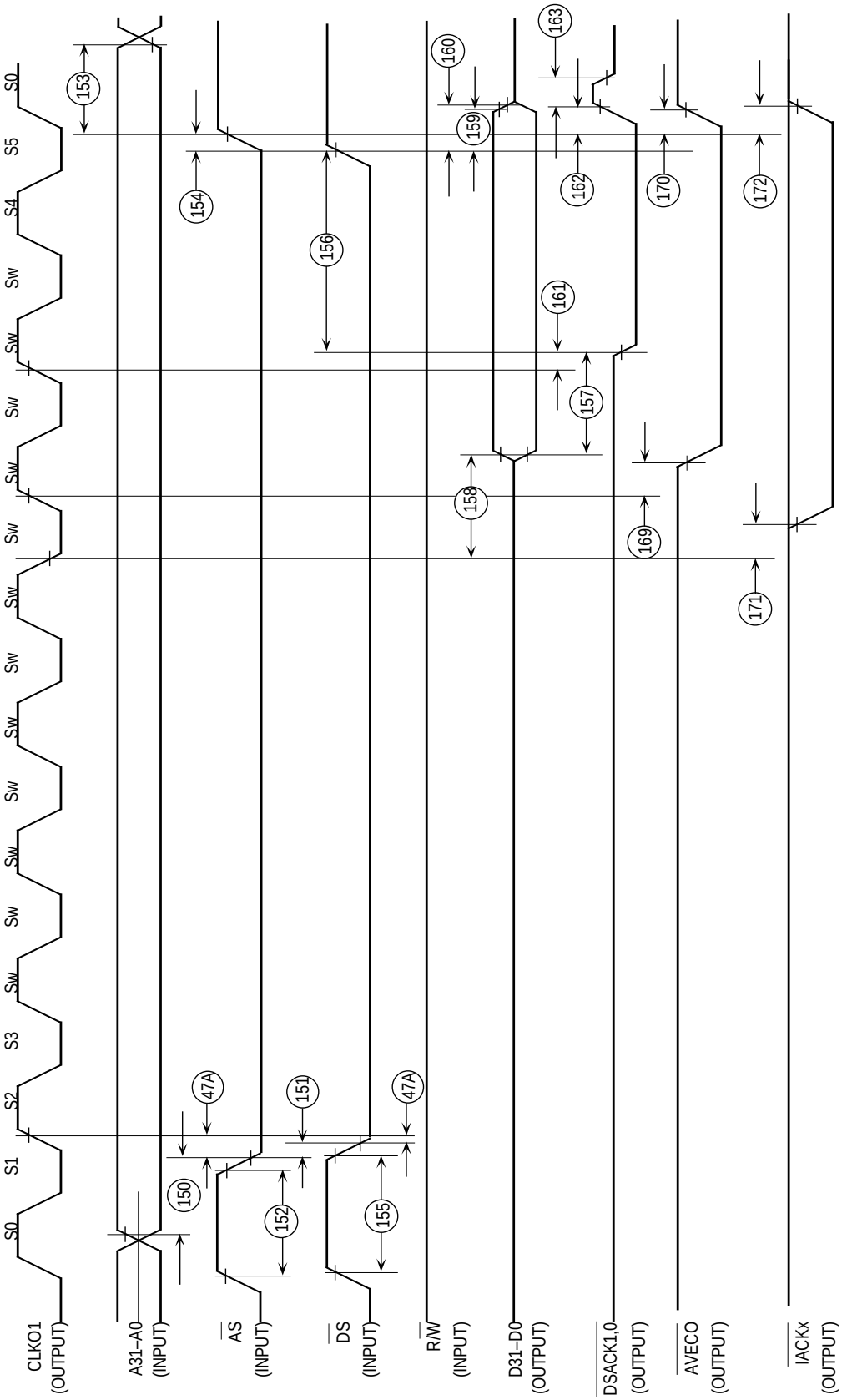


Figure 10-29. External MC68030/MC68360 Asynchronous IACK Cycle Timing Diagram