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	Active
Core Processor	EC000
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	16MHz
Co-Processors/DSP	-
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	132-BQFP Bumpered
Supplier Device Package	132-PQFP (24.13x24.13)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68306fc16b

— Sales Offices —

UNITED STATES

ALABAMA, Huntsville	(205) 464-6800
ARIZONA, Tempe	(602) 897-5056
CALIFORNIA, Agoura Hills	(818) 706-1929
CALIFORNIA, Los Angeles	(310) 417-8848
CALIFORNIA, Irvine	(714) 753-7360
CALIFORNIA, Rosevlle	(916) 922-7152
CALIFORNIA, San Diego	(619) 541-2163
CALIFORNIA, Sunnyvale	(408) 749-0510
COLORADO, Colorado Springs	(719) 599-7497
COLORADO, Denver	(303) 337-3434
CONNECTICUT, Wallingford	(203) 949-4100
FLORIDA, Maitland	(407) 628-2636
FLORIDA, Pompano Beach/ Fort Lauderdale	(305) 486-9776
FLORIDA, Clearwater	(813) 538-7750
GEORGIA, Atlanta	(404) 729-7100
IDAHO, Boise	(208) 323-9413
ILLINOIS, Chicago/Hoffman Estates	(708) 490-9500
INDIANA, Fort Wayne	(219) 436-5818
INDIANA, Indianapolis	(317) 571-0400
INDIANA, Kokomo	(317) 457-6634
IOWA, Cedar Rapids	(319) 373-1328
KANSAS, Kansas City/Mission	(913) 451-8555
MARYLAND, Columbia	(410) 381-1570
MASSACHUSETTS, Marborough	(508) 481-8100
MASSACHUSETTS, Woburn	(617) 932-9700
MICHIGAN, Detroit	(313) 347-6800
MINNESOTA, Minnetonka	(612) 932-1500
MISSOURI, St. Louis	(314) 275-7380
NEW JERSEY, Fairfield	(201) 808-2400
NEW YORK, Fairport	(716) 425-4000
NEW YORK, Hauppauge	(516) 361-7000
NEW YORK, Poughkeepsie/Fishkill	(914) 473-8102
NORTH CAROLINA, Raleigh	(919) 870-4355
OHIO, Cleveland	(216) 349-3100
OHIO, Columbus Worthington	(614) 431-8492
OHIO, Dayton	(513) 495-6800
OKLAHOMA, Tulsa	(800) 544-9496
OREGON, Portland	(503) 641-3681
PENNSYLVANIA, Colmar	(215) 997-1020
Philadelphia/Horsham	(215) 957-4100
TENNESSEE, Knoxville	(615) 690-5593
TEXAS, Austin	(512) 873-2000
TEXAS, Houston	(800) 343-2692
TEXAS, Plano	(214) 516-5100
VIRGINIA, Richmond	(804) 285-2100
WASHINGTON, Bellevue	(206) 454-4160
Seattle Access	(206) 622-9960
WISCONSIN, Milwaukee/Brookfield	(414) 792-0122

Field Applications Engineering Available
Through All Sales Offices

CANADA

BRITISH COLUMBIA, Vancouver	(604) 293-7605
ONTARIO, Toronto	(416) 497-8181
ONTARIO, Ottawa	(613) 226-3491
QUEBEC, Montreal	(514) 731-6881

INTERNATIONAL

AUSTRALIA, Melbourne	(61-3)887-0711
AUSTRALIA, Sydney	(61(2)906-3855
BRAZIL, Sao Paulo	55(11)815-4200
CHINA, Beijing	86 505-2180
FINLAND, Helsinki	358-0-35161191
Car Phone	358(49)211501
FRANCE, Paris/Vanves	33(1)40 955 900
GERMANY, Langenhagen/ Hanover	49(511)789911
GERMANY, Munich	49 89 92103-0
GERMANY, Nuremberg	49 911 64-3044
GERMANY, Sindelfingen	49 7031 69 910
GERMANY,Wiesbaden	49 611 761921
HONG KONG, Kwai Fong	852-4808333
Tai Po	852-6668333
INDIA, Bangalore	(91-812)627094
ISRAEL, Tel Aviv	972(3)753-8222
ITALY, Milan	39(2)82201
JAPAN, Aizu	81(241)272231
JAPAN, Atsugi	81(0462)23-0761
JAPAN, Kumagaya	81(0485)26-2600
JAPAN, Kyushu	81(092)771-4212
JAPAN, Mito	81(0292)26-2340
JAPAN, Nagoya	81(052)232-1621
JAPAN, Osaka	81(06)305-1801
JAPAN, Sendai	81(22)268-4333
JAPAN, Tachikawa	81(0425)23-6700
JAPAN, Tokyo	81(03)3440-3311
JAPAN, Yokohama	81(045)472-2751
KOREA, Pusan	82(51)4635-035
KOREA, Seoul	82(2)554-5188
MALAYSIA, Penang	60(4)374514
MEXICO, Mexico City	52(5)282-2864
MEXICO, Guadalajara	52(36)21-8977
Marketing	52(36)21-9023
Customer Service	52(36)669-9160
NETHERLANDS, Best	(31)49988 612 11
PUERTO RICO, San Juan	(809)793-2170
SINGAPORE	(65)2945438
SPAIN, Madrid	34(1)457-8204
or	34(1)457-8254
SWEDEN, Solna	46(8)734-8800
SWITZERLAND, Geneva	41(22)7991111
SWITZERLAND, Zurich	41(1)730 4074
TAIWAN, Taipei	886(2)717-7089
THAILAND, Bangkok	(66-2)254-4910
UNITED KINGDOM, Aylesbury	44(296)395-252

FULL LINE REPRESENTATIVES

COLORADO, Grand Junction	
Cheryl Lee Whltely	(303) 243-9658
KANSAS, Wichita	
Melinda Shores/Kelly Greiving	(316) 838 0190
NEVADA, Reno	
Galena Technology Group	(702) 746 0642
NEW MEXICO, Albuquerque	
S&S Technologies, Inc.	(505) 298-7177
UTAH, Salt Lake City	
Utah Component Sales, Inc.	(801) 561-5099
WASHINGTON, Spokane	
Doug Kenley	(509) 924-2322
ARGENTINA, Buenos Aires	
Argonics, S.A.	(541) 343-1787

HYBRID COMPONENTS RESELLERS

Elmo Semiconductor	(818) 768-7400
Minco Technology Labs Inc.	(512) 834-2022
Semi Dice Inc.	(310) 594-4631

SECTION 2
SIGNAL DESCRIPTION

This section contains a brief description of the input and output signals, with reference (if applicable) to other sections which give greater detail on its use. Figure 2-1 provides a detailed diagram showing the integrated peripherals and signals, and Tables 2-1–2-7 provides a quick reference for determining a signal's name, mnemonic, its use as an input or output, active state, and type identification.

NOTE

The terms **assertion** and **negation** will be used extensively. This is done to avoid confusion when dealing with a mixture of “active low” and “active high” signals. The term 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. The term negate or negation is used to indicate that a signal is inactive or false.

additional information about the interaction between $\overline{HALT}$ and $\overline{RESET}$, refer to **3.5 Reset Operation** and for more information on $\overline{HALT}$ and $\overline{BERR}$, refer to **3.4 Bus Error and Halt Operation**.

Processor assertion of $\overline{HALT}$ indicates a double bus fault condition. This condition is unrecoverable; the MC68306 must be externally reset to resume operation.

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

This three-state, bi-directional signal defines the data bus transfer as a read or write cycle. The R/$\overline{W}$ signal relates to the data strobe signals described in the following paragraphs.

2.1.13 Upper And Lower Data Strobes ($\overline{UDS}$, $\overline{LDS}$)

These three-state, bi-directional signals and R/$\overline{W}$ control the flow of data on the data bus. Table 2-9 lists the combinations of these signals, the corresponding data on the bus, and the $\overline{OE}$, $\overline{LW}$, and $\overline{UW}$ signals. When the R/$\overline{W}$ line is high, the processor reads from the data bus. When the R/$\overline{W}$ line is low, the processor drives the data bus. When another bus master controls the bus, the $\overline{UDS}$, $\overline{LDS}$, and R/$\overline{W}$ pins become inputs and the $\overline{OE}$, $\overline{LW}$, and $\overline{UW}$ signals are still decoded as shown in Table 2-9.

Table 2-9. Data Strobe Control of Data Bus

$\overline{UDS}$	$\overline{LDS}$	R/$\overline{W}$	D8–D15	D0–D7	$\overline{OE}$	$\overline{UW}$	$\overline{LW}$
High	High	—	No Valid Data	No Valid Data	High	High	High
Low	Low	High	Valid Data Bits 15–8	Valid Data Bits 7–0	Low	High	High
High	Low	High	No Valid Data	Valid Data Bits 7–0	Low	High	High
Low	High	High	Valid Data Bits 15–8	No Valid Data	Low	High	High
Low	Low	Low	Valid Data Bits 15–8	Valid Data Bits 7–0	High	Low	Low
High	Low	Low	Valid Data Bits 7–0*	Valid Data Bits 7–0	High	High	Low
Low	High	Low	Valid Data Bits 15–8	Valid Data Bits 15–8*	High	Low	High

*These conditions are a result of current implementation and may not appear on future devices.

2.1.14 Upper-Byte Write ($\overline{UW}$)

This signal is a combination of R/$\overline{W}$ low and $\overline{UDS}$ low for writing the upper-byte of a 16-bit port. This signal simplifies memory system design by explicitly signalling that data is valid on the upper portion of the data bus on a write operation. $\overline{UW}$ is also decoded for external bus masters.

The bus request from the granted device should be negated after $\overline{BGACK}$ is asserted. If another bus request is pending, $\overline{BG}$ is reasserted within a few clocks, as described in **3.3 Bus Arbitration Control**. The processor does not perform any external bus cycles before reasserting $\overline{BG}$.

3.3 BUS ARBITRATION CONTROL

All asynchronous bus arbitration signals to the processor are synchronized before being used internally. As shown in Figure 3-16, synchronization requires a maximum of one and a half cycles of the system clock. The input asynchronous signal is sampled on the falling edge of the clock and is valid internally after the next rising edge.

This synchronization scheme is used for all other asynchronous inputs also: $\overline{RESET}$, $\overline{HALT}$, $\overline{DTACK}$, $\overline{BERR}$, $\overline{IPL2}$–$\overline{IPL0}$.

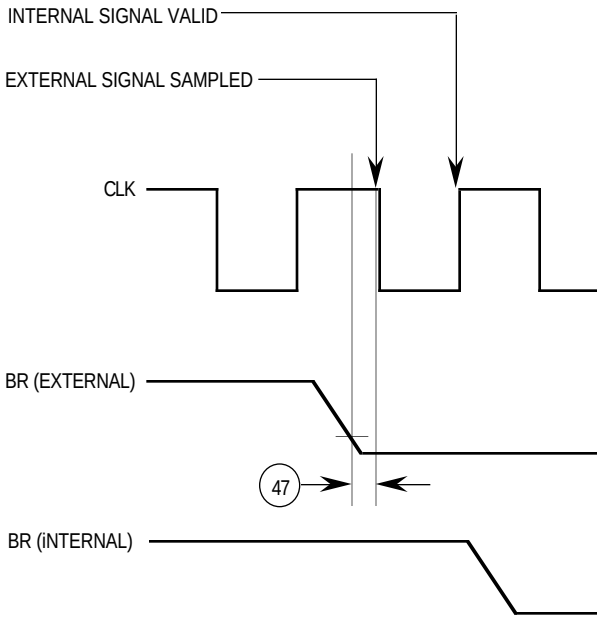


Figure 3-16. External Asynchronous Signal Synchronization

Bus arbitration control is implemented with a finite state machine (see Figure 3-17). In Figure 3-17, input signals R and A are the internally synchronized versions of $\overline{BR}$ and $\overline{BGACK}$. The $\overline{BG}$ output is shown as G, and the internal three-state control signal is shown as T. If T is true, the address, data, and control buses are placed in the high-impedance state when $\overline{AS}$ is negated. All signals are shown in positive logic (active high), regardless of their true active voltage level. State changes (valid outputs) occur on the next rising edge of the clock after the internal signal is valid.

A timing diagram of the bus arbitration sequence during a processor bus cycle is shown in Figure 3-18. The bus arbitration timing while the bus is inactive (e.g., the processor is performing internal operations for a multiply instruction) is shown in Figure 3-19.

Figures 3-18, 3-19, and 3-20 apply to processors using 3-wire bus arbitration. Figures 3-21, 3-22, and 3-23 apply to processors using 2-wire bus arbitration.

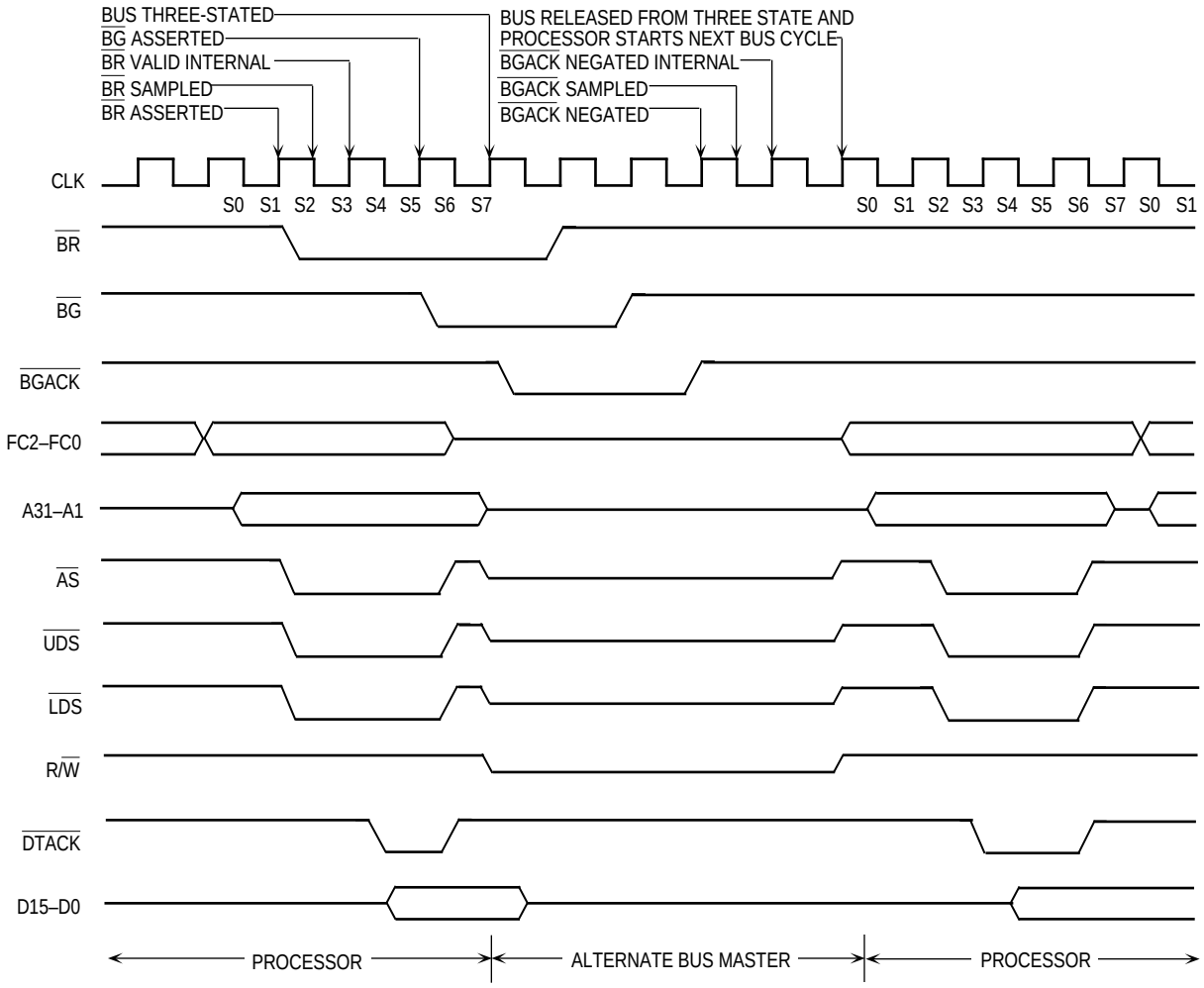


Figure 3-18. Three-Wire Bus Arbitration Timing Diagram—Processor Active

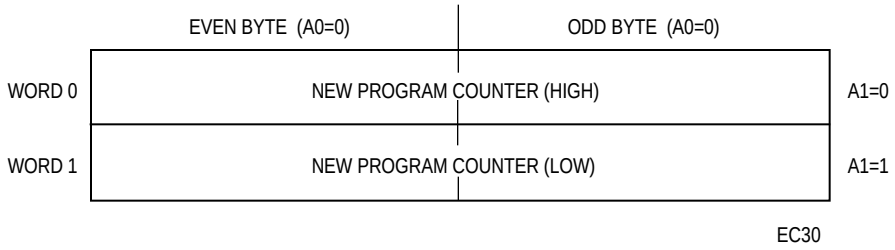


Figure 4-5. Exception Vector Format

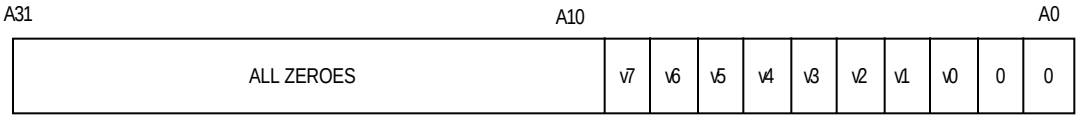


Figure 4-6. Address Translated from 8-Bit Vector Number

The actual address on the address bus is truncated to the number of address bits available on the bus of the particular implementation of the M68000 architecture. In the EC000 core, this is 24 address bits. The memory map for exception vectors is shown in Table 4-5.

The vector table is 512 words long (1024 bytes), starting at address 0 (decimal) and proceeding through address 1023 (decimal). The vector table provides 255 unique vectors, some of which are reserved for trap and other system function vectors. Of the 255, 192 are reserved for user interrupt vectors. However, the first 64 entries are not protected, so user interrupt vectors may overlap at the discretion of the systems designer.

Port B pins can be individually programmed as either IRQ, $\overline{\text{IACK}}$ or parallel port signals. To use any of the port B pins PB7–PB4 as interrupt request signals (IRQ6, IRQ5, IRQ3, IRQ2) be sure the bit is programmed as an input. Interrupt enables are provided for each interrupt level.

To use any of the port B pins PB3–PB0 as $\overline{\text{IACK6}}$, $\overline{\text{IACK5}}$, $\overline{\text{IACK3}}$, or $\overline{\text{IACK2}}$, program the port data bit and the autovector bit to zero.

To use any of the port B pins PB3–PB0 as port inputs, ensure that the autovector bit is one.

Open-drain or open-source operation can be emulated by programming the appropriate fixed data (e.g. 0 = open-drain) and toggling the direction control. PB7–PB4 pins can be programmed as outputs even when enabled as interrupt inputs, allowing inputs to be tested or emulated if the interrupt is open-drain or open-source. The active interrupt level is the inverse of the IX register bit.

5.2.5.1 PORT PINS REGISTER

FFFFFFF4/5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
RESET: PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0

SUPERVISOR ONLY

The port pin register bits are the data at the port pins, regardless of pin direction. The port pins register is read-only, writes are ignored.

5.2.5.2 PORT DIRECTION REGISTER

FFFFFFF2/3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PADIR7	PADIR6	PADIR5	PADIR4	PADIR3	PADIR2	PADIR1	PADIR0	PBDIR7	PBDIR6	PBDIR5	PBDIR4	PBDIR3	PBDIR2	PBDIR1	PBDIR0
RESET: 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SUPERVISOR ONLY

The port direction register bits determine the direction of data flow at the port pins.

PADIR7–0—Port A Direction Register Bit 7–0

This bit determines the direction of data flow at port A pins 7 through 0.

0 = Input.
1 = Output.

PBDIR7–0—Port B Direction Register Bit 7–0

5.2.7.1 DRAM REFRESH REGISTER. The refresh timer is a programmable period counter that generates a refresh request every 16 to 4096 EXTAL periods, programmable in 16 EXTAL period increments.

FFFFFFFC

15	14	13	12	11	10	9	8
RR7	RR6	RR5	RR4	RR3	RR2	RR1	RR0
RESET: U	U	U	U	U	U	U	U

SUPERVISOR ONLY

RR7–0—Refresh Rate Period

The value set in this field supplies the refresh rate for the DRAM controller. The refresh rate can be calculated from the equation:

Period = (16 × (register value +1)) × EXTAL

Where:

EXTAL is the crystal period in nanoseconds and period is in nanoseconds.

5.2.7.2 DRAM BANK CONFIGURATION REGISTER (HIGH HALF). The DRAM configuration registers are not affected by any reset, and must be explicitly programmed. This applies to both banks, whether used or not. Unused banks must be disabled to prevent interference with other address decodes.

FFFFFFE4/5 (DR1), FFFFFFE0/1 (DR0)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRA31	DRA30	DRA29	DRA28	DRA27	DRA26	DRA25	DRA24	DRA23	DRA22	DRA21	DRA20	DRA19	DRA18	DRA17	DRW
RESET: U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

SUPERVISOR ONLY

DRA31–DRA17—DRAM Bank Address

This bit field selects the base address for DRAM bank.

DRW—DRAM Write

This bit determines whether write cycles are permitted to DRAM bank space. If read and write cycles are both inhibited, the DRAM bank is inhibited.

0 = Write cycles are inhibited to DRAM bank space
1 = Write cycles are permitted to DRAM bank space

NOTE

Never perform a TAS instruction to DRAM if the DRAM is configured as write-only.

5.2.7.3 DRAM BANK CONFIGURATION REGISTER (LOW HALF)

FFFFFFE6/7 (DR1), FFFFFFE2/3 (DR0)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRR	DRFC 6	DRFC 5	—	—	DRFC 2	DRFC 1	—	DRM3	DRM2	DRM1	DRM0	DRSZ2	DRSZ1	DRSZ 0	DRDT
RESET: U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

SUPERVISOR ONLY

DRR—DRAM Read

This bit determines whether read cycles are permitted to DRAM bank space. If read and write cycles are both inhibited, DRAM bank is inhibited.

0 = Read cycles are inhibited to DRAM bank space
1 = Read cycles are permitted to DRAM bank space

DRFC6, 5, 2, 1—DRAM Bank Function Code 6, 5, 2, 1 Enable

This bit determines which function code accesses are permitted to DRAM bank space. If all function code cycles are inhibited, the DRAM bank is inhibited.

0 = Function code n cycles are inhibited to DRAM bank space
1 = Function code n cycles are permitted to DRAM bank space

DRM3–0—DRAM Bank Address Match

This field determines which DRAM bank address bits must match address bits for DRAM bank to occur. DRA bits not included in match must be set to zero, or else DRAM bank is inhibited.

0000 = A31–A17 ignored in DRAM bank address match
0001 = A31 must match DRA31; A30–A17 ignored in DRAM bank address match
0010 = A31–A30 must match DRA31–DRA30; A29–A17 ignored in DRAM bank address match
.....
1111 = A31–A17 must match DRA31–DRA17 in DRAM bank address match

Table 5-4 shows the entire range of address bits that must match for a DRAM bank to occur.

6.3 OPERATION

The following paragraphs describe the operation of the baud rate generator, transmitter and receiver, and other functional operating modes of the serial module.

6.3.1 Baud Rate Generator

The baud rate generator consists of a crystal oscillator, baud rate generator, and clock selectors (see Figure 6-3). The crystal oscillator operates directly from a 3.6864-MHz crystal or from an external clock of the same frequency. Baud rates are selected by programming the clock-select register (DUCSR) for each channel.

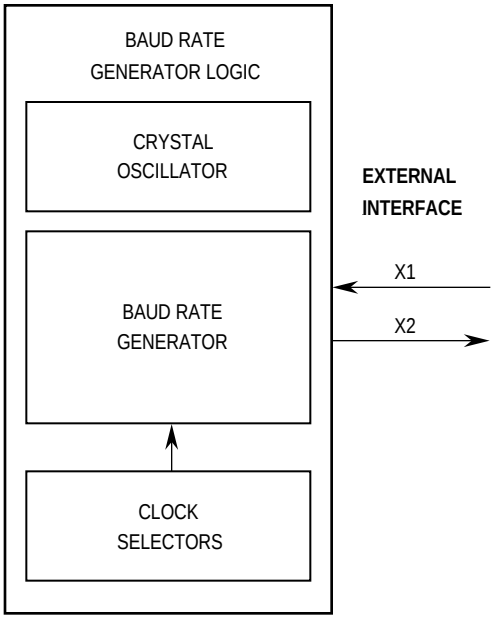
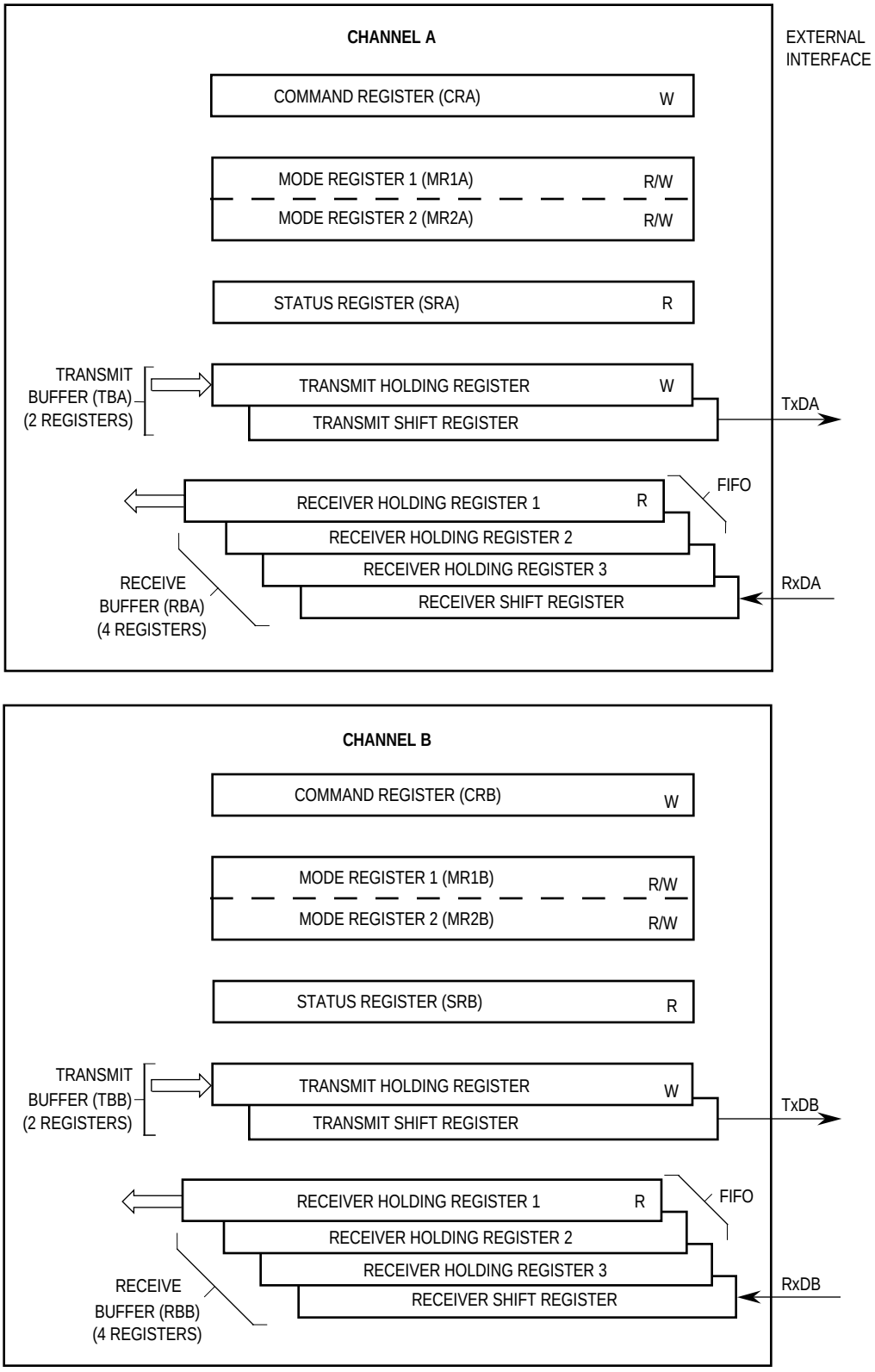


Figure 6-3. Baud Rate Generator Block Diagram

6.3.2 Transmitter and Receiver Operating Modes

The functional block diagram of the transmitter and receiver, including command and operating registers, is shown in Figure 6-4. The paragraphs that follow contain descriptions for both these functions in reference to this diagram. For detailed register information, refer to **6.4 Register Description and Programming**.



NOTE:
R/W = READ/WRITE
R = READ
W = WRITE

Figure 6-4. Transmitter and Receiver Functional Diagram

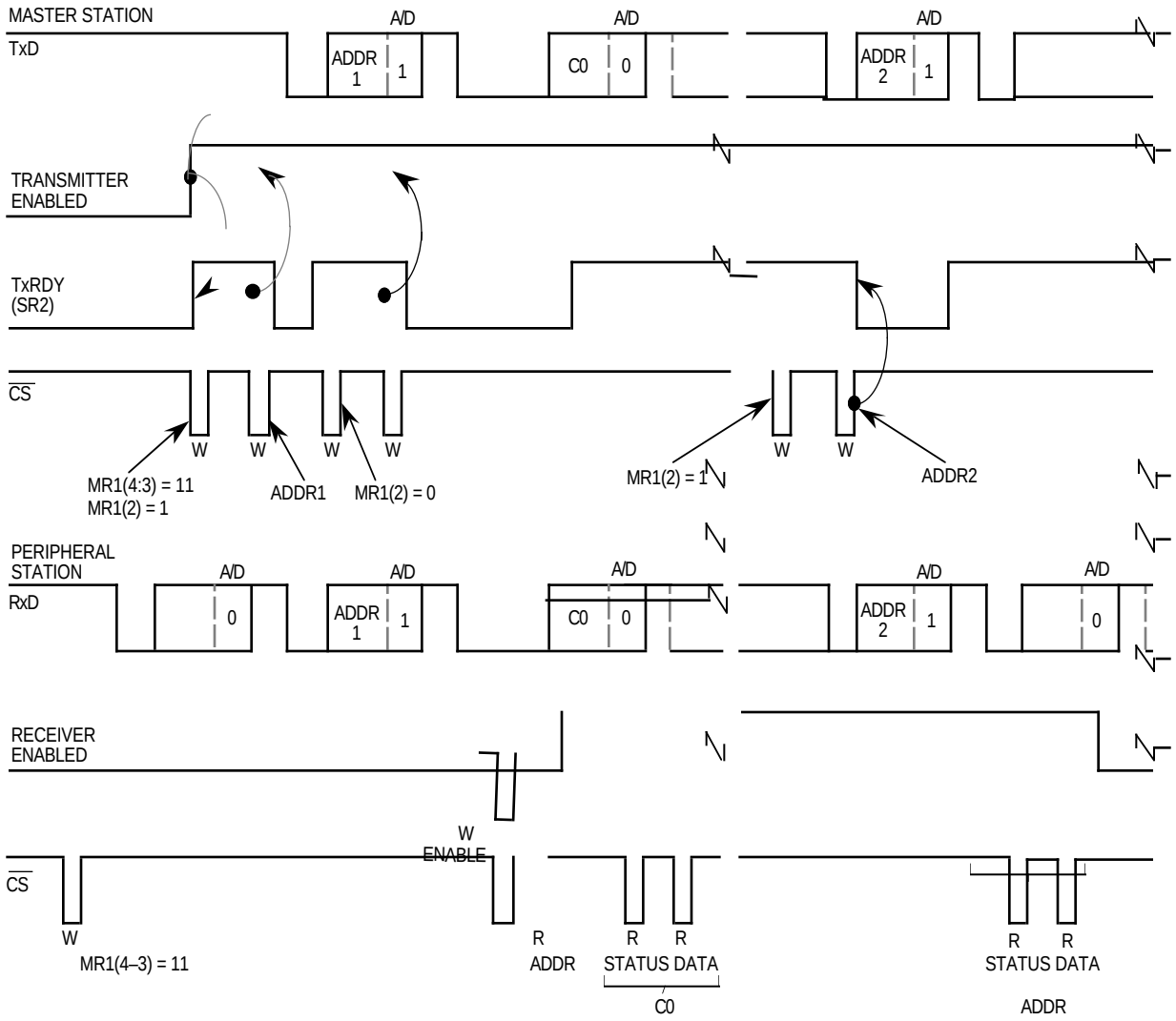


Figure 6-8. Multidrop Mode Timing Diagram

A transmitted character from the master station consists of a start bit, a programmed number of data bits, an address/data (A/D) bit flag, and a programmed number of stop bits. The A/D bit identifies the type of character being transmitted to the slave station. The character is interpreted as an address character if the A/D bit is set or as a data character if the A/D bit is cleared. The polarity of the A/D bit is selected by programming bit 2 of the DUMR1. The DUMR1 should be programmed before enabling the transmitter and loading the corresponding data bits into the transmit buffer.

In multidrop mode, the receiver continuously monitors the received data stream, regardless of whether it is enabled or disabled. If the receiver is disabled, it sets the RxRDY bit and loads the character into the receiver holding register FIFO stack provided the received A/D bit is a one (address tag). The character is discarded if the received A/D bit is a zero (data tag). If the receiver is enabled, all received characters are transferred to the CPU via the receiver holding register stack during read operations.

channel A mode register pointer points to DUMR1. The pointer is set to DUMR1 by RESET or by a set pointer command, using control register A. After reading or writing DUMR1A, the pointer points to DUMR2A.

DUMR1A, DUMR1B

7	6	5	4	3	2	1	0
RxRTS	RxIRQ	ERR	PM1	PM0	PT	B/C1	B/C0
RESET:							
0	0	0	0	0	0	0	0

Read/Write

RxRTS—Receiver Request-to-Send Control

1 = Upon receipt of a valid start bit, RTSx is negated if the channel's FIFO is full. RTSx is reasserted when the FIFO has an empty position available.
0 = The receiver has no effect on RTSx.

This feature can be used for flow control to prevent overrun in the receiver by using the RTSx output to control the CTSx input of the transmitting device. If both the receiver and transmitter are programmed for RTS control, RTS control will be disabled for both since this configuration is incorrect. See **6.4.1.17 Mode Register 2** for information on programming the transmitter RTSx control.

RxIRQ—Receiver Interrupt Select

1 = FFULL is the source that generates IRQ.
0 = RxRDY is the source that generates IRQ.

ERR—Error Mode

This bit controls the meaning of the three FIFO status bits (RB, FE, and PE) in the DUSR for the channel.

1 = Block mode—The values in the channel DUSR are the accumulation (i.e., the logical OR) of the status for all characters coming to the top of the FIFO since the last reset error status command for the channel was issued. Refer to **6.4.1.5 Command Register (DUCR)** for more information on serial module commands.
0 = Character mode—The values in the channel DUSR reflect the status of the character at the top of the FIFO.

NOTE

ERR = 0 must be used to get the correct A/D flag information when in multidrop mode.

PM1–PM0—Parity Mode

These bits encode the type of parity used for the channel (see Table 6-1). The parity bit is added to the transmitted character, and the receiver performs a parity check on incoming data. These bits can alternatively select multidrop mode for the channel.

DBB—Delta Break B

1 = Enable interrupt
0 = Disable interrupt

FFULLB—Channel B FIFO Full

1 = Enable interrupt
0 = Disable interrupt

TxRDYB, TxRDYA—Transmitter Ready

1 = Enable interrupt
0 = Disable interrupt

CTR/TMR_RDY—Counter/Timer Ready

1 = Enable interrupt
0 = Disable interrupt

DBA—Delta Break A

1 = Enable interrupt
0 = Disable interrupt

FFULLA—Channel A FIFO Full

1 = Enable interrupt
0 = Disable interrupt

6.4.1.12 COUNT REGISTER: CURRENT MSB OF COUNTER (DUCUR). This register holds the most-significant byte of the current value in the counter/timer. It should only be read when the counter/timer is in counter mode and the counter is stopped. See **6.3.5 Counter/Timer** for further information.

6.4.1.13 COUNT REGISTER: CURRENT LSB OF COUNTER (DUCLR). This register holds the least-significant byte of the current value in the counter/timer. It should only be read when the counter/timer is in counter mode and the counter is stopped. See **6.3.5 Counter/Timer** for further information.

6.4.1.14 COUNTER/TIMER UPPER PRELOAD REGISTER (DUCTUR). This register holds the eight most-significant bits of the preload value to be used by the conter/timer in either the count or timer mode. The minimum value that can be loaded on the concatenation of DUCTUR with DUCTLR is 0002 (hex). This register is write only and cannot be read by the CPU.

6.4.1.15 COUNTER/TIMER LOWER PRELOAD REGISTER (DUCTLR). This register holds the eight least-significant bits of the preload value to be used by the conter/timer in either the count or timer mode. The minimum value that can be loaded on the concatenation of DUCTUR with DUCTLR is 0002 (hex). This register is write only and cannot be read by the CPU.

6.4.2 Programming

The basic interface software flowchart required for operation of the serial module is shown in Figure 6-10. The routines are divided into three categories:

- Serial Module Initialization
- I/O Driver
- Interrupt Handling

6.4.2.1 SERIAL MODULE INITIALIZATION. The serial module initialization routines consist of SINIT and CHCHK. SINIT is called at system initialization time to check channel A and channel B operation. Before SINIT is called, the calling routine allocates two words on the system stack. Upon return to the calling routine, SINIT passes information on the system stack to reflect the status of the channels. If SINIT finds no errors in either channel A or channel B, the respective receivers and transmitters are enabled. The CHCHK routine performs the actual channel checks as called from the SINIT routine. When called, SINIT places the specified channel in the local loopback mode and checks for the following errors:

- Transmitter Never Ready
- Receiver Never Ready
- Parity Error
- Incorrect Character Received

6.4.2.2 I/O DRIVER EXAMPLE. The I/O driver routines consist of INCH and OUTCH. INCH is the terminal input character routine and gets a character from the channel receiver. OUTCH is used to send a character to the channel transmitter.

6.4.2.3 INTERRUPT HANDLING. The interrupt handling routine consists of SIRQ, which is executed after the serial module generates an interrupt caused by a channel A change-in-break (beginning of a break). SIRQ then clears the interrupt source, waits for the next change-in-break interrupt (end of break), clears the interrupt source again, then returns from exception processing to the system monitor.

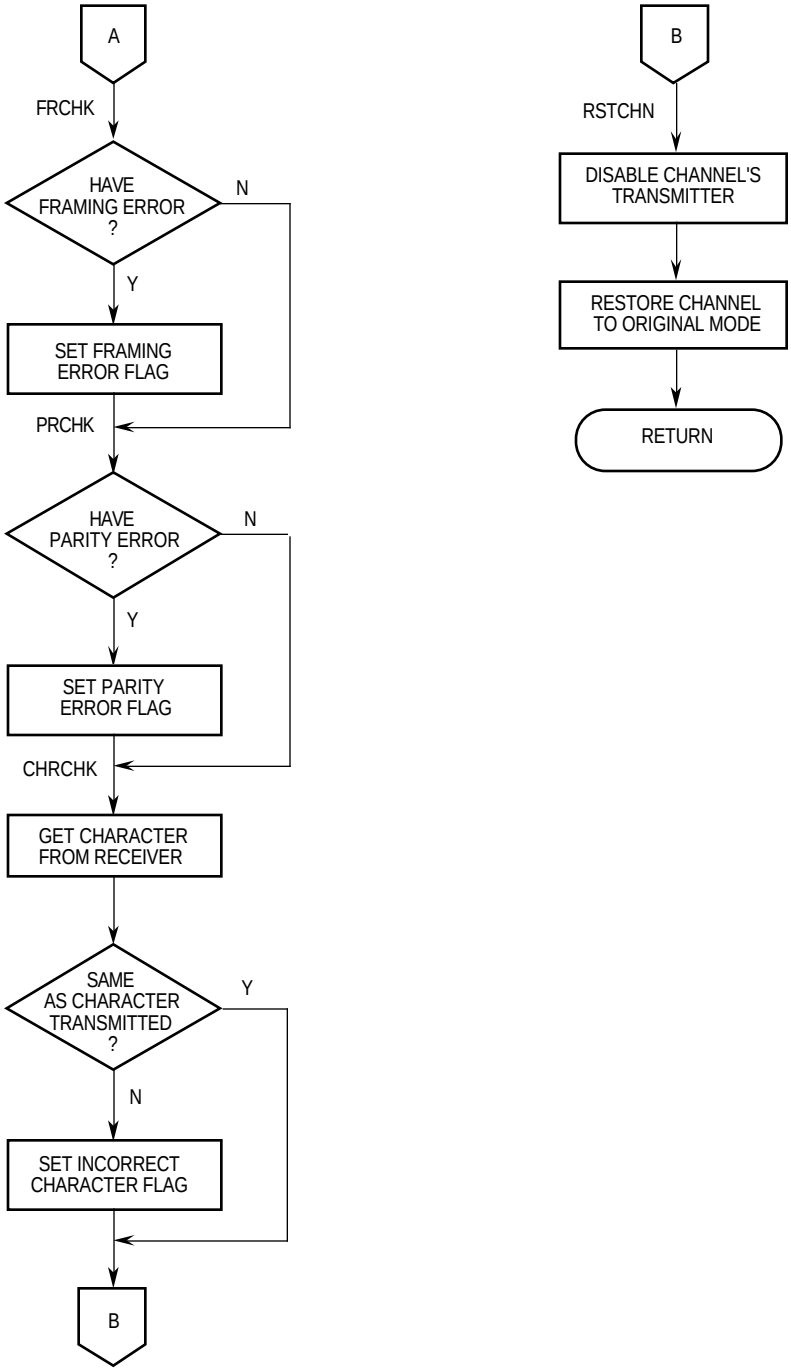


Figure 6-10. Serial Module Programming Flowchart (3 of 5)

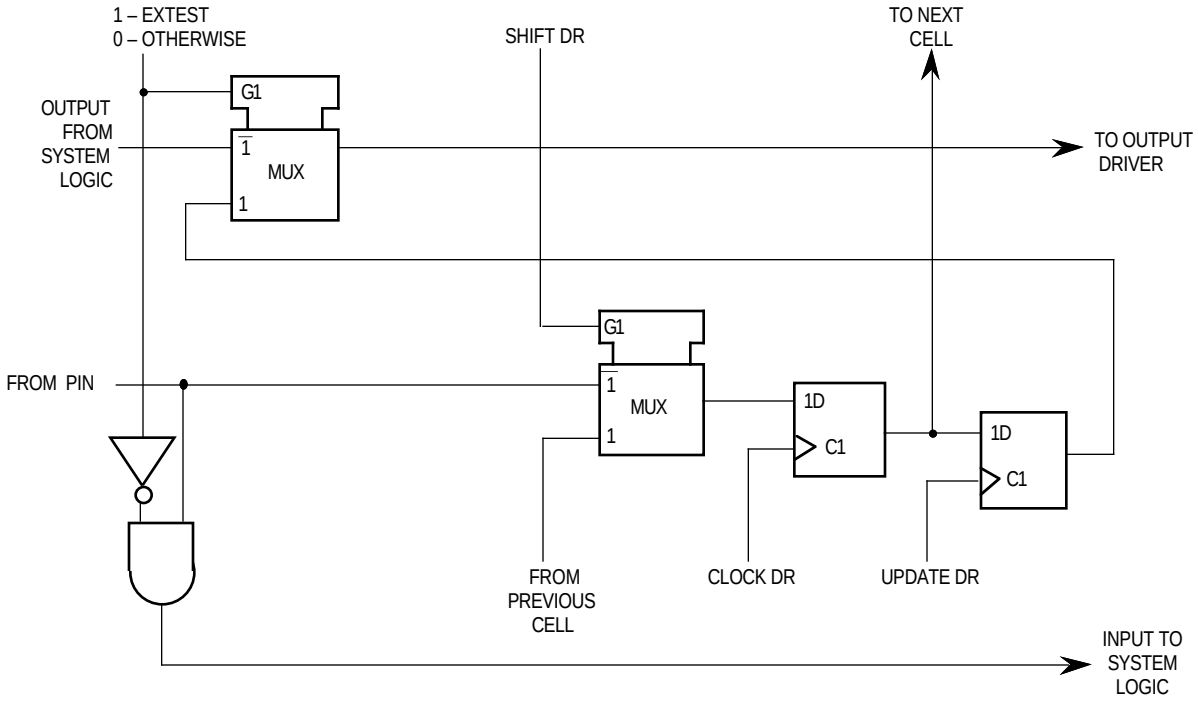
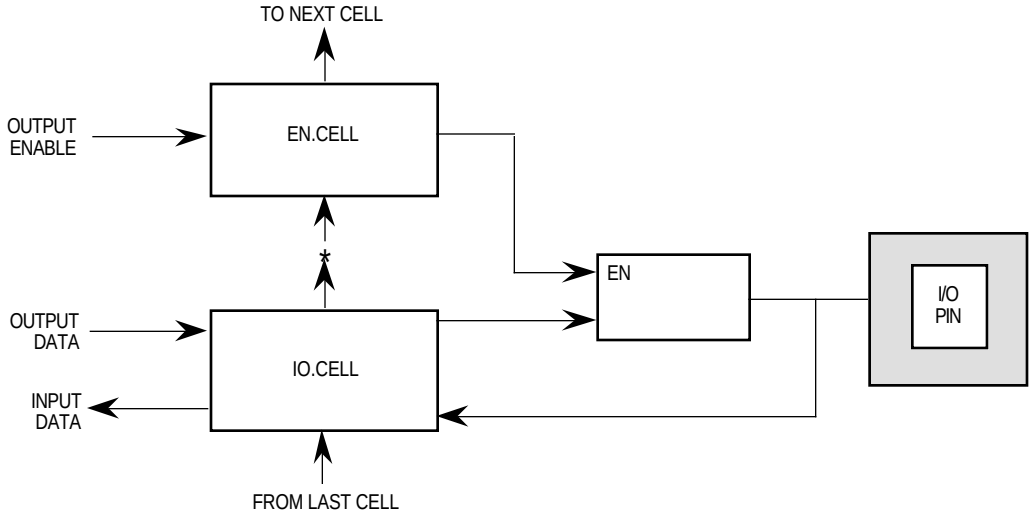


Figure 7-7. Bidirectional Cell (IOx0.Cell)



NOTE: More than one IO.Cell could be serially connected and controlled by a single En.Cell.

Figure 7-8. General Arrangement for Bidirectional Pins

7.4 INSTRUCTION REGISTER

The MC68306 IEEE 1149.1 implementation includes the three mandatory public instructions (EXTEST, SAMPLE/PRELOAD, and BYPASS), the optional public ID instruction, plus one additional public instruction (CLAMP) defined by IEEE 1149.1. The

NOTE

Since there is no internal synchronization between the IEEE 1149.1 clock (TCK) and the system clock (CLKOUT), the user must provide some form of external synchronization to achieve meaningful results.

The second function of SAMPLE/PRELOAD is to initialize the boundary scan register output bits prior to selection of EXTEST. This initialization ensures that known data will appear on the outputs when entering the EXTEST instruction.

7.4.3 BYPASS (010, 101, 111)

The BYPASS instruction selects the single-bit bypass register as shown in Figure 7-9. This creates a shift-register path from TDI to the bypass register and, finally, to TDO, circumventing the 124-bit boundary scan register. This instruction is used to enhance test efficiency when a component other than the MC68306 becomes the device under test.

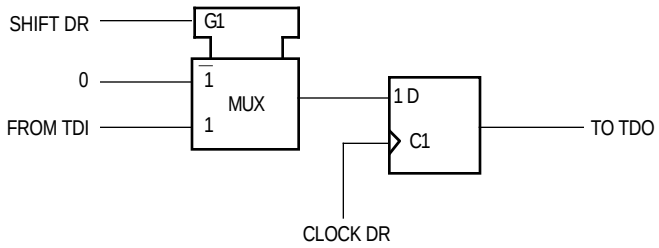


Figure 7-9. Bypass Register

When the bypass register is selected by the current instruction, the shift-register stage is set to a logic zero on the rising edge of TCK in the capture-DR controller state. Therefore, the first bit to be shifted out after selecting the bypass register will always be a logic zero.

7.4.4 CLAMP (011)

When the CLAMP instruction is invoked, the boundary scan multiplexer control signal EXTEST is asserted, and the BYPASS register is selected. CLAMP should be invoked after valid data has been shifted into the boundary scan register, e.g. by SAMPLE/PRELOAD. CLAMP allows static levels to be presented at the MC68306 output and bidirectional pins, like EXTEST, but without the shift latency of the boundary scan register from TDI to TDO.

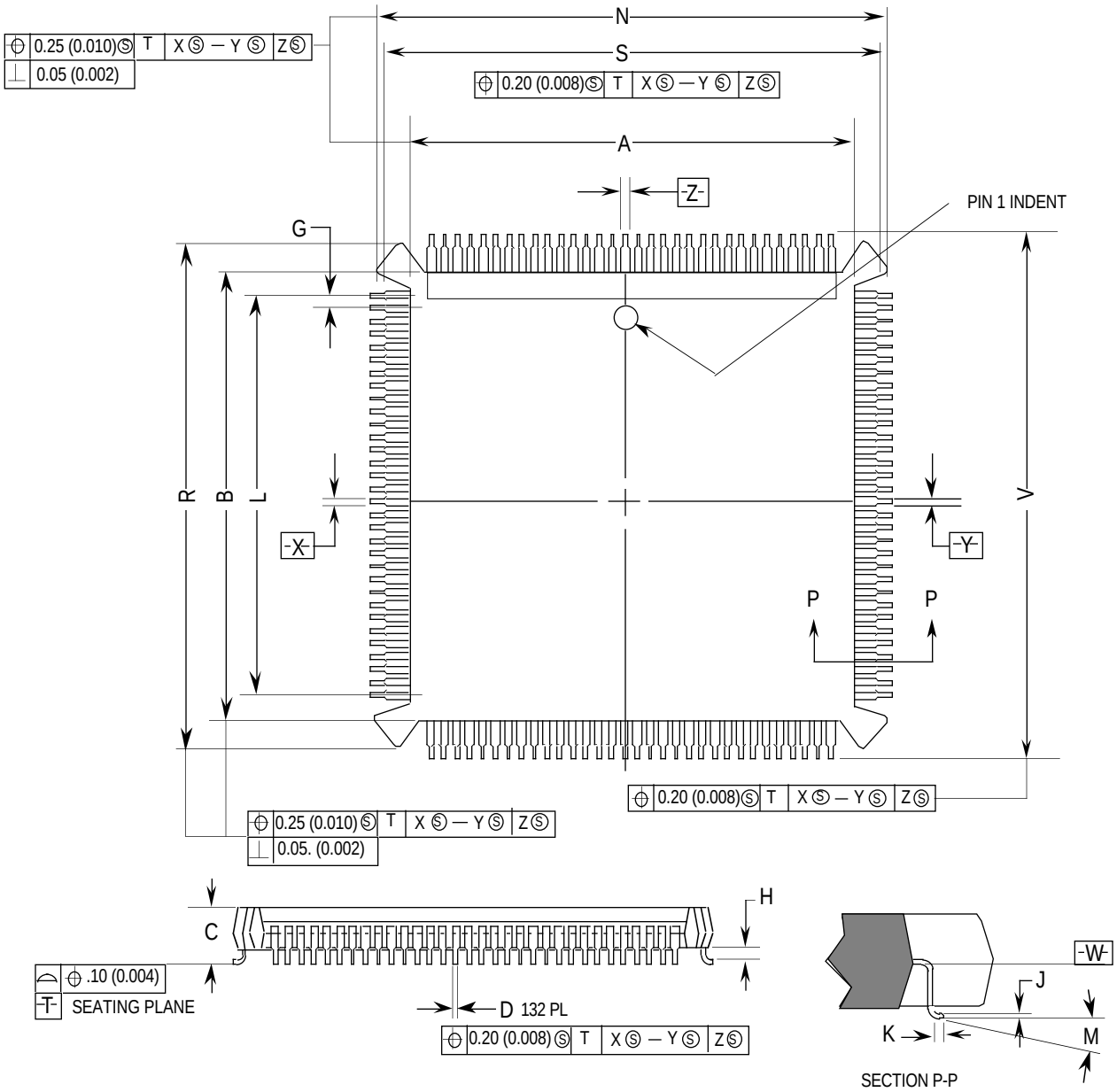
7.5 MC68306 RESTRICTIONS

The control afforded by the output enable signals using the boundary scan register and the EXTEST instruction requires a compatible circuit-board test environment to avoid device-destructive configurations. The user must avoid situations in which the MC68306 output drivers are enabled into actively driven networks. Overdriving the TDO driver when it is active is not recommended.

9.3 PACKAGE DIMENSIONS

132 Pin PQFP (FC Suffix)

CASE 831A-01



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	24.06	24.20	0.947	0.953
B	24.06	24.20	0.947	0.953
C	4.07	4.57	0.160	0.180
D	0.21	0.30	0.008	0.012
G	0.64 BSC		0.025 BSC	
H	0.51	1.01	0.020	0.040
J	0.16	0.20	0.006	0.008
K	0.51	0.76	0.020	0.030
L	20.32 REF		0.800 REF	
M	0°	8°	0°	8°
N	27.88	28.01	1.097	1.103
R	27.88	28.01	1.097	1.103
S	27.31	27.55	1.075	1.085
V	27.31	27.55	1.075	1.085

NOTES:

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH
3. DIMENSIONS A, B, N, AND R DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION FOR DIMENSIONS A AND B IS 0.25 (0.010), FOR DIMENSIONS N AND R IS 0.18 (0.007).
4. DATUM PLANE **-W-** IS LOCATED AT THE UNDERSIDE OF LEADS WHERE LEADS EXIT PACKAGE BODY.
5. DATUMS **-X-** , **-Y-**, AND **-Z-** TO BE DETERMINED WHERE CENTER LEADS EXIT PACKAGE BODY AT DATUM **-W-** .
6. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE, DATUM **-T-**.
7. DIMENSIONS A, B, N AND R TO BE DETERMINED AT DATUM PLANE **-W-** .

INDEX