NXP USA Inc. - KMC68302AG25C Datasheet

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	M68000
Number of Cores/Bus Width	1 Core, 8/16-Bit
Speed	25MHz
Co-Processors/DSP	Communications; RISC CPM
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	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/kmc68302ag25c

try is $0F0; however, the actual BAR is a 16-bit value within the BAR entry and is located at $0F2.

After a total system reset, the on-chip peripheral base address is undefined, and it is not possible to access the on-chip peripherals at any address until BAR is written. The BAR and the SCR can always be accessed at their fixed addresses.

NOTE

> The BAR, SCR and CKCR registers are internally reset only when a total system reset occurs by the simultaneous assertion of RESET and HALT. The chip-select (CS) lines are not asserted on accesses to these locations. Thus, it is very helpful to use CS lines to select external ROM/RAM that overlaps the BAR and SCR register locations, since this prevents potential bus contention. (The internal access (IAC) signal may also be used to prevent bus contention.)

NOTE

> In 8-bit system bus operation, IMP accesses are not possible until the low byte of the BAR is written. Since the MOVE.W instruction writes the high byte followed by the low byte, this instruction guarantees the entire word is written.

Do not assign other devices on the system bus an address that falls within the address range of the peripherals defined by the BAR. If this happens, $\overline{\text{BERR}}$ is generated (if the address decode conflict enable (ADCE) bit is set) and the address decode conflict (ADC) bit in the SCR is set.

The BAR is a 16-bit, memory-mapped, read-write register consisting of the high address bits, the compare function code bit, and the function code bits. Upon a total system reset, its value may be read as $BFFF, but its value is not valid until written by the user. The address of this register is fixed at $0F2 in supervisor data space. BAR cannot be accessed in user data space.

15 13	12	11											0
FC2–FC0	CFC	BASE ADDRESS											
		23	22	21	20	19	18	17	16	15	14	13	12

Bits 15–13—FC2–FC0

The FC2–FC0 field is contained in bits 15–13 of the BAR. These bits are used to set the address space of 4K-byte block of on-chip peripherals. The address compare logic uses these bits, dependent upon the CFC bit, to cause an address match within its address space.

NOTE

> Do not assign this field to the M68000 core interrupt acknowledge space (FC2–FC0 = 7).

Table 2-8. Parameter RAM

Base + 580 • • • Base + 5BF		SCC2 SCC2	Specific Protocol Parameters
Base + 5C0 • • • Base + 5FF			Reserved (Not Implemented)
Base + 600	4 Word	SCC3	RxBD0
Base + 608	4 Word	SCC3	RxBD1
Base + 610	4 Word	SCC3	RxBD2
Base + 618	4 Word	SCC3	RxBD3
Base + 620	4 Word	SCC3	RxBD4
Base + 628	4 Word	SCC3	RxBD5
Base + 630	4 Word	SCC3	RxBD6
Base + 638	4 Word	SCC3	RxBD7
Base + 640	4 Word	SCC3	TxBD0
Base + 648	4 Word	SCC3	TxBD1
Base + 650	4 Word	SCC3	TxBD2
Base + 658	4 Word	SCC3	TxBD3 ##
Base + 660	3 Word	SMC	Reserved
Base + 666	Word	SCC1	RxBD
Base + 668	Word	SCC1	TxBD
Base + 66A	Word	SCC2	RxBD
Base + 66C	Word	SCC2	TxBD
Base + 66E #	6 Word	SMC1–SMC2	Internal Use
Base +67A	Word	SCP	Rx/TxBD
Base +67C	Word	SCC1–SCC3	BERR Channel Number
Base +67E #	Word	CP	MC68302 Revision Number
Base + 680 • • • Base + 6BF		SCC3 SCC3	Specific Protocol Parameters
Base + 6C0 • • • Base + 7FF			Reserved (Not Implemented)

# Modified by the CP after a CP or system reset.

## Tx BD 4, 5, 6, and 7 are not initially available to SCC3. (See 4.5.5 Buffer Descriptors Table for information on how they may be regained.)

In addition to the internal dual-port RAM, a number of internal registers support the functions of the various M68000 core peripherals. The internal registers (see Table 2-9) are memory-mapped registers offset from the BAR point1616er and are located on the internal M68000 bus.

$\overline{\text{DTACK}}$ generation occurs under the same constraints as the chip-select signal—if the chip-select signal does not activate, then neither will the $\overline{\text{DTACK}}$ signal.

Chip select 0 has the special property of being enabled upon system reset to the address range from 0 to 8K bytes. This property allows chip select 0 to function as the “boot ROM” select on system start-up. $\overline{\text{DTACK}}$ is initially enabled for six wait states on this chip select.

External masters may use the chip-select logic on the IMP during an external master access to external memory/peripherals. In this case, the external master chip-select timing diagram (see Figure 6-15) must be used. Since the chip-select logic is slightly slower when using external masters, an optional provision can be made to add an additional wait state to an external access by an external master. See the EMWS bit in the SCR for more details (3.8.3 System Control Bits).

A priority structure exists within the chip-select block. For a given address, the priority is as follows:

1. Access to any IMP internal address (BAR, dual-port RAM, etc.)
   No chip select asserted.
2. Chip Select 0
3. Chip Select 1
4. Chip Select 2
5. Chip Select 3

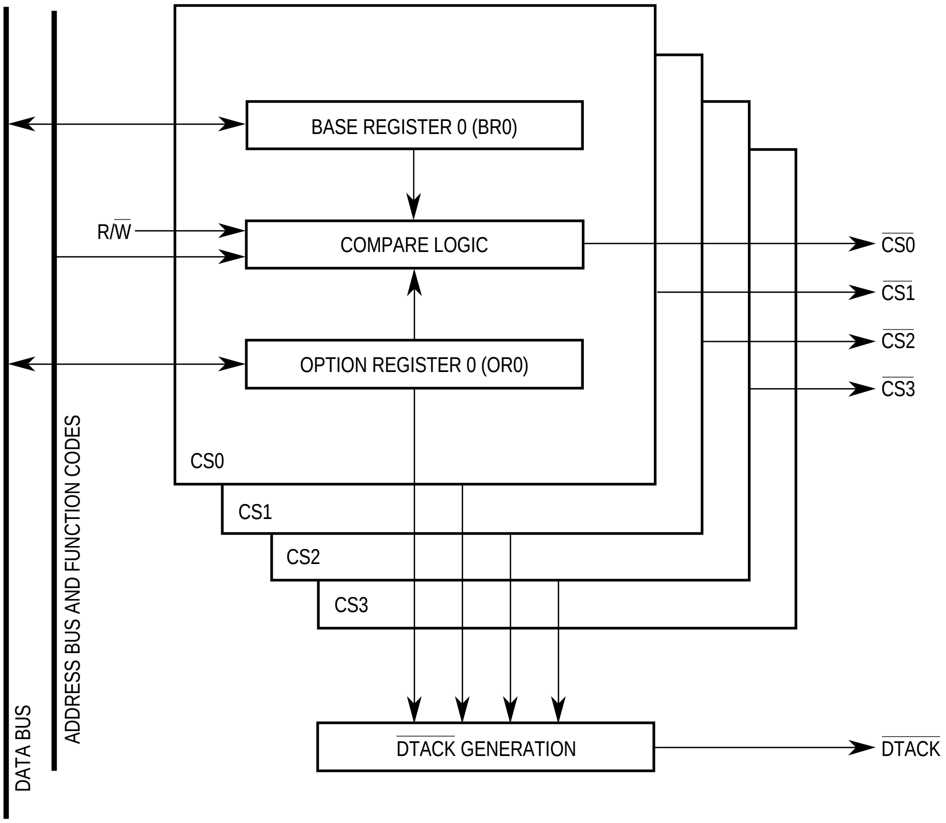


Figure 3-9. Chip-Select Block Diagram

The user should not normally program more than one chip-select line to the same area. When this occurs, the address compare logic will set address decode conflict (ADC) in the system control register (SCR) and generate BERR if address decode conflict enable (ADCE) is set. Only one chip-select line will be driven because of internal line priorities. CS0 has the highest priority, and CS3 the lowest. BERR will not be asserted on write accesses to the chip-select registers.

If one chip select is programmed to be read-only and another chip select is programmed to be write-only, then there will be no overlap conflict between these two chip selects, and the ADC bit will not be set.

When a bus master attempts to write to a read-only location, the chip-select logic will set write protect violation (WPV) in the SCR and generate BERR if write protect violation enable (WPVE) is set. The CS line will not be asserted.

NOTE

The chip-select logic is reset only on total system reset (assertion of RESET and HALT). Accesses to the internal RAM and registers, including the system configuration registers (BAR and

FLG—Command Semaphore Flag

The bit is set by the M68000 core and cleared by the CP.

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 command execution. Note that the execution of the STOP TRANSMIT or RESTART TRANSMIT commands may not affect the TXD pin until many clocks after the FLG bit is cleared by the CP due to the transmit FIFO latency.

4.3.1 Command Execution Latency

Commands are executed at a priority higher than the SCCs, but less than the priority of the DRAM refresh controller. The longest command, the ENTER HUNT MODE command, executes in 41 clocks. All other commands execute in less than 20 clocks. The maximum command latency is calculated as follows:

- Command execution time (41 or 20) +
- 25 clocks if DRAM refresh controller is used +

—205 clocks if any SCC is enabled with BISYNC; or
—165 clocks if any SCC is enabled with Transparent; or
—165 clocks if any SCC is enabled with HDLC; or
—150 clocks if any SCC is enabled with UART; or
—140 clocks if any SCC is enabled with DDCMP; else
—0

For example, if HDLC and UART modes are used on the SCCs with the DRAM refresh controller operating, the maximum command latency is 41 + 25 + 165 = 231 clocks = 13 µs at 16.67 MHz. The equations assume that the DRAM refresh cycle occurs once during the command latency. Note that commands are typically given only in special error-handling situations and that the typical latency is much less than the worst case value.

4.4 SERIAL CHANNELS PHYSICAL INTERFACE

The serial channels physical interface joins the physical layer serial lines to the three SCCs and the two SMCs. (The separate three-wire SCP interface is described in 4.6 Serial Communication Port (SCP).)

The MC68302 supports four different external physical interfaces from the SCCs:

1. NMSI—Nonmultiplexed Serial Interface
2. PCM—Pulse Code Modulation Highway
3. IDL—Interchip Digital Link
4. GCI—General Circuit Interface

The most generic physical interface on the MC68302 is the nonmultiplexed serial interface (NMSI). The NMSI consists of seven of the basic modem (or RS-232) signals: TXD, TCLK, RXD, RCLK, $\overline{\text{RTS}}$, $\overline{\text{CTS}}$, and $\overline{\text{CD}}$. Each SCC can have its own set of NMSI signals as shown in Figure 4-3. In addition to the NMSI, the baud rate generator clocks may be output on separate BRG pins as shown. All NMSI2 pins are multiplexed with parallel I/O pins. The user may choose which NMSI2 pins are used by the SCC2 and which are used as parallel I/O. On NMSI3, the TXD, TCLK, RXD, and RCLK pins are multiplexed with parallel I/O lines;

crystal, an externally supplied UART clock on the TCLK or RCLK pins can be as high as 6.67 MHz, giving a maximum baud rate of 417 kbaud.

The baud rate using the baud rate generator is (System Clock or TIN1 clock)/(1 or 4)/(Clock Divider + 1)/16. The baud rate using the baud rate generator with an externally supplied clock to the TCLK or RCLK pins is always (TCLK or RCLK)/16.

Table 4-4 shows examples of typical bit rates of asynchronous communication and how to obtain them with the baud rate generator using an internally supplied clock.

Table 4-4. Typical Bit Rates of Asynchronous Communication

Baud Rates	15.36			16.0			16.667		
	DIV4	DIV	Actual Frequency	DIV4	DIV	Actual Frequency	DIV4	DIV	Actual Frequency
150	1	1599	150	1	1666	149.97	1	1735	150.01
300	1	799	300	1	832	300.12	1	867	300.02
600	0	1599	600	0	1666	599.88	0	1735	600.05
1200	0	799	1200	0	832	1200.48	0	867	1200.1
2400	0	399	2400	0	416	2398.08	0	433	2400.2
4800	0	199	4800	0	207	4807.69	0	216	4800.4
9600	0	99	9600	0	103	9615.34	0	108	9556.76
19200	0	49	19200	0	51	19230.8	0	53	19290.5
38400	0	24	38400	0	25	38461.53	0	26	38581.0

4.5.2.2 Synchronous Baud Rate Generator Examples

For synchronous communication (HDLC/SDLC, BISYNC, DDCMP, Transparent, and V.110), the internal clock is identical to the baud rate output. To obtain the desired rate, the user selects the appropriate system clock according to the following equation:

Baud rate = (System Clock or TIN1 Clock)/(Clock Divider + 1)/(1 or 4) according to the DIV4 bit

For example, to get the data rate of 64 kbps, the system clock can be 15.36 MHz, DIV4 = 0, and the Clock Divider = 239. Of course, a 64 kbps rate provided externally on the TCLK or RCLK pins could also be used.

4.5.3 SCC Mode Register (SCM)

Each SCC has a mode register. The functions of bits 5–0 are common to each protocol. The function of the specific mode bits varies according to the protocol selected by the MODE1–MODE0 bits. They are described in the relevant sections for each protocol type. Each SCM is a 16-bit, memory-mapped, read-write register. The SCMs are cleared by reset.

15 6	5	4	3	2	1	0
SPECIFIC MODE BITS	DIAG1	DIAG0	ENR	ENT	MODE1	MODE0

DIAG1–DIAG0—Diagnostic Mode

00 = Normal operation ($\overline{\text{CTS}}$, $\overline{\text{CD}}$ lines under automatic control)
In this mode, the CTS and CD lines are monitored by the SCC controller. The SCC controller uses these lines to automatically enable/disable reception and transmission.

4.5.11.5 UART Command Set

These commands are issued to the command register described in 4.3 Command Set**.**

STOP TRANSMIT Command

After a hardware or software reset and the enabling of the channel by writing the SCC mode register, the channel is in the transmit enable mode and starts polling the first BD in the table approximately every eight transmit clocks.

The channel STOP TRANSMIT command disables the transmission of characters on the transmit channel. If this command is received by the UART controller during message transmission, transmission of that message is aborted. The UART completes transmission of any data already transferred to the UART FIFO (up to three characters) and then stops transmitting data. The TBD# is not advanced.

The UART transmitter will transmit a programmable number of break sequences and then start to transmit idles. The number of break sequences (which may be zero) should be written to the break count register (BRKCR) before this command is given to the UART controller.

The STOP TRANSMIT command must be issued before the SCC mode register is used to disable the transmitter if the transmitter will be reenabled at a later time.

RESTART TRANSMIT Command

The channel RESTART TRANSMIT command re-enables the transmission of characters on the transmit channel. This command is expected by the UART in three situations: after issuing a STOP TRANSMIT command, after issuing a STOP TRANSMIT and then disabling the channel using the SCC mode register, or after transmitter errors (CTS lost). The UART controller will resume transmission from the current transmitter BD number (TBD#) in the channel's Tx BD table.

If the transmitter is being re-enabled, the RESTART TRANSMIT command must be used and should be followed by the enabling of the transmitter in the SCC mode register.

ENTER HUNT MODE Command

After a hardware or software reset and the enabling of the channel by its SCC mode register, the channel is in the receive enable mode and will use the first BD in the table.

The ENTER HUNT MODE command is used to force the UART controller to abort reception of the current message, generate an RX interrupt (if enabled) as the buffer is closed, and enter the hunt mode. The UART controller will resume reception using the next BD once an address character or a single idle character is received. In multidrop hunt mode, the UART controller continually scans the input data stream for the address character. While not in multidrop mode, the UART controller will wait for a single IDLE character. In the UART mode, none of the data received in the FIFO is lost when ENTER HUNT MODE command is issued; however, this command does reset the receive FIFO in other protocols, e.g., HDLC.

If an enabled receiver has been disabled by clearing ENR in the SCC mode register, the ENTER HUNT MODE command must be given to the channel before setting ENR again. Reception will then begin with the next BD.

BRK—Break Character Received

A break character was received.

CCR—Control Character Received

A control character was received (with reject (R) character = 1) and stored in the receive control character register (RCCR).

BSY—Busy Condition

A character was received and discarded due to lack of buffers. The receiver automatically enters hunt mode immediately if in the multidrop mode. The latest that an Rx BD can be made empty (have its empty bit set) and still avoid the busy condition is the middle of the stop bit of the first character to be stored in that buffer.

TX—Tx Buffer

A buffer has been transmitted over the UART channel. If CR = 1 in the Tx BD, this bit is set no sooner than when the last data bit of the last character in the buffer begins to be transmitted. If CR = 0, this bit is set after the last character was written to the transmit FIFO.

RX—Rx Buffer

A buffer has been received over the UART channel. This event occurs no sooner than the middle of the first stop bit of the character that causes the buffer to be closed.

4.5.11.17 UART MASK Register

The SCC mask register (SCCM) is referred to as the UART mask register when the SCC is operating as a UART. It is an 8-bit read-write register with the same bit formats as the UART event register. If a bit in the UART mask register is a one, the corresponding interrupt in the event register will be enabled. If the bit is zero, the corresponding interrupt in the event register will be masked. This register is cleared upon reset.

4.5.11.18 S-Records Programming Example

In the following paragraphs, an example of a downloading application is given that utilizes an SCC channel as a UART controller. The application performs downloads and uploads of S records between a host computer and an intelligent peripheral through a serial asynchronous line.

The S records are strings of ASCII characters that begin with `S' and end in an end-of-line character. This characteristic will be used to impose a message structure on the communication between the devices. Note that each device may also transmit XON and XOFF characters for flow control, which do not form part of the program being uploaded or downloaded.

The UART mode register should be set as required, with the freeze (FRZ) bit cleared and the enable transmitter/receiver (ENT, ENR) bits set. Receive buffers should be linked to the receive buffer table with the interrupt (I) bit set. For simplicity, assume that the line is not multidrop (no addresses are transmitted) and that each S record will fit into a single data buffer.

ceeds the length of the data buffer, the HDLC controller will fetch the next BD in the table and, if it is empty, will continue to transfer the rest of the frame to this BD's associated data buffer.

During this process, the HDLC controller will check for a frame that is too long. When the frame ends, the CRC field is checked against the recalculated value and is written to the data buffer starting with the first address byte. The data length written to the last BD in the HDLC frame is the length of the entire frame. This enables HDLC protocols that “lose” frames to correctly recognize the frame-too-long condition. The HDLC controller then sets the last buffer in frame bit, writes the frame status bits into the BD, and clears the empty bit. The HDLC controller next generates a maskable interrupt, indicating that a frame has been received and is in memory. The HDLC controller then waits for a new frame. Back-to-back frames may be received with only a single shared flag between frames. Also, flags that share a zero will be recognized as two consecutive flags.

4.5.12.3 HDLC Memory Map

When configured to operate in HDLC mode, the IMP overlays the structure shown in Table 4-7 onto the protocol-specific area of that SCC parameter RAM. Refer to 2.8 MC68302 Memory Map for the placement of the three SCC parameter RAM areas and to Table 4-2 for the other parameter RAM values.

Table 4-8. HDLC-Specific Parameter RAM

Address	Name	Width	Description
SCC Base + 9C SCC Base + 9E SCC Base + A0 # SCC Base + A2 # SCC Base + A4 SCC Base + A6	RCRC_L RCRC_H C_MASK_L C_MASK_H TCRC_L TCRC_H	Word Word Word Word Word Word	Temp Receive CRC Low Temp Receive CRC High Constant ($F0B8 16-Bit CRC, $DEBB 32-Bit CRC) Constant ($XXXX 16-Bit CRC, $20E3 32-Bit CRC) Temp Transmit CRC Low Temp Transmit CRC High
SCC Base + A8 # SCC Base + AA # SCC Base + AC # SCC Base + AE # SCC Base + B0 #	DISFC CRCEC ABTSC NMARC RETRC	Word Word Word Word Word	Discard Frame Counter CRC Error Counter Abort Sequence Counter Nonmatching Address Received Counter Frame Retransmission Counter
SCC Base + B2 # SCC Base + B4	MFLR MAX_cnt	Word Word	Max Frame Length Register Max_Length Counter
SCC Base + B6 # SCC Base + B8 # SCC Base + BA # SCC Base + BC # SCC Base + BE #	HMASK HADDR1 HADDR2 HADDR3 HADDR4	Word Word Word Word Word	User-Defined Frame Address Mask User-Defined Frame Address User-Defined Frame Address User-Defined Frame Address User-Defined Frame Address

# Initialized by the user (M68000 core).

NOTE

> An incorrect initialization of C_MASK may be used to “force” receive CRC errors for software testing purposes. The transmit CRC will not be affected.

4.5.12.4 HDLC Programming Model

The M68000 core configures each SCC to operate in one of four protocols by the MODE1–MODE0 bits in the SCC mode register (SCM). MODE1–MODE0 = 00 selects HDLC mode.

By setting its SCC mode register (SCM), any of the SCC channels may be configured to function as a DDCMP controller. The DDCMP link can be either synchronous (by programming the MODE1–MODE0 bits of the SCC mode register to DDCMP) or asynchronous (by programming the MODE1–MODE0 bits of the SCC mode register to asynchronous and setting the DDCMP bit in the UART mode register). The DDCMP controller handles the basic functions of the DDCMP protocol in both cases.

The SCC in DDCMP mode can work in either IDL, GCI, PCM highway, or NMSI interfaces. When the SCC is used with a modem interface (NMSI), the serial outputs are connected directly to the external pins. The modem interface uses seven dedicated pins: transmit data (TXD), receive data (RXD), receive clock (RCLK), transmit clock (TCLK), carrier detect (CD), clear to send (CTS), and request to send (RTS). Other modem lines can be supported through the parallel I/O pins.

The DDCMP controller consists of separate transmit and receive sections whose operations are asynchronous with the M68000 core and may be either synchronous or asynchronous with respect to the other SCCs. Each clock can be supplied either from the baud rate generator or externally. More information on the baud rate generator is available in 4.5.2 SCC Configuration Register (SCON).

The DDCMP controller key features are as follows:

- Synchronous or Asynchronous DDCMP Links Supported
- Flexible Data Buffers
- Four Address Comparison Registers with Mask
- Automatic Frame Synchronization
- Automatic Message Synchronization by Searching for SOH, ENQ, or DLE
- CRC16 Generation/Checking
- NRZ/NRZI Data Encoding
- Maintenance of Four 16-Bit Error Counters

4.5.14.1 DDCMP Channel Frame Transmission Processing

The DDCMP transmitter is designed to work with almost no intervention from the M68000 core (see Figure 4-35).

When the M68000 core enables the DDCMP transmitter and the link is synchronous, it starts transmitting SYN1–SYN2 pairs (programmed in the data synchronization register) or IDLEs as determined in the DDCMP mode register. The DDCMP controller polls the first buffer descriptor (BD) in the channel's transmit BD table. When there is a message to transmit, the DDCMP controller fetches the data from memory and starts transmitting the message (after first transmitting the SYN1–SYN2 pair when the link is synchronous).

E—Empty

0 = This bit is cleared by the CP to indicate that the data bits associated with this BD are now available to the M68000 core.

1 = This bit is set by the M68000 core to indicate that the data bits associated with this BD have been read.

NOTE

> Additional data received will be discarded until the empty bit is set by the M68000 core.

Bits 14–6—These bits are reserved and should be set to zero by the M68000 core.

C/I—Command/Indication Channel Data

Bits 1–0—The CP always writes these bits with zeros.

4.7.4.4 SMC2 Transmit Buffer Descriptor

In the IDL mode, this BD is identical to the SMC1 transmit BD. In the GCI mode, SMC2 is used to control the C/I channel.

15	14	6	5	2	1	0
R	RESERVED		C/I		0	0

R—Ready

0 = This bit is cleared by the CP after transmission to indicate that the BD is now available to the M68000 core.

1 = This bit is set by the M68000 core to indicate that the data associated with this BD is ready for transmission.

Bits 14–6—Reserved for future use; should be set to zero by the user.

C/I—Command/Indication Channel Data

Bits 1–0—These bits should be written with zeros by the M68000 core.

4.7.5 SMC Interrupt Requests

SMC1 and SMC2 send individual interrupt requests to the IMP interrupt controller when one of the respective SMC receive buffers is full or when one of the SMC transmit buffers is empty. Each of the two interrupt requests from each SMC is enabled when its respective SMC channel is enabled in the SPMODE register. Interrupt requests from SMC1 and SMC2 can be masked in the interrupt mask register. See 3.2 Interrupt Controller for more details.

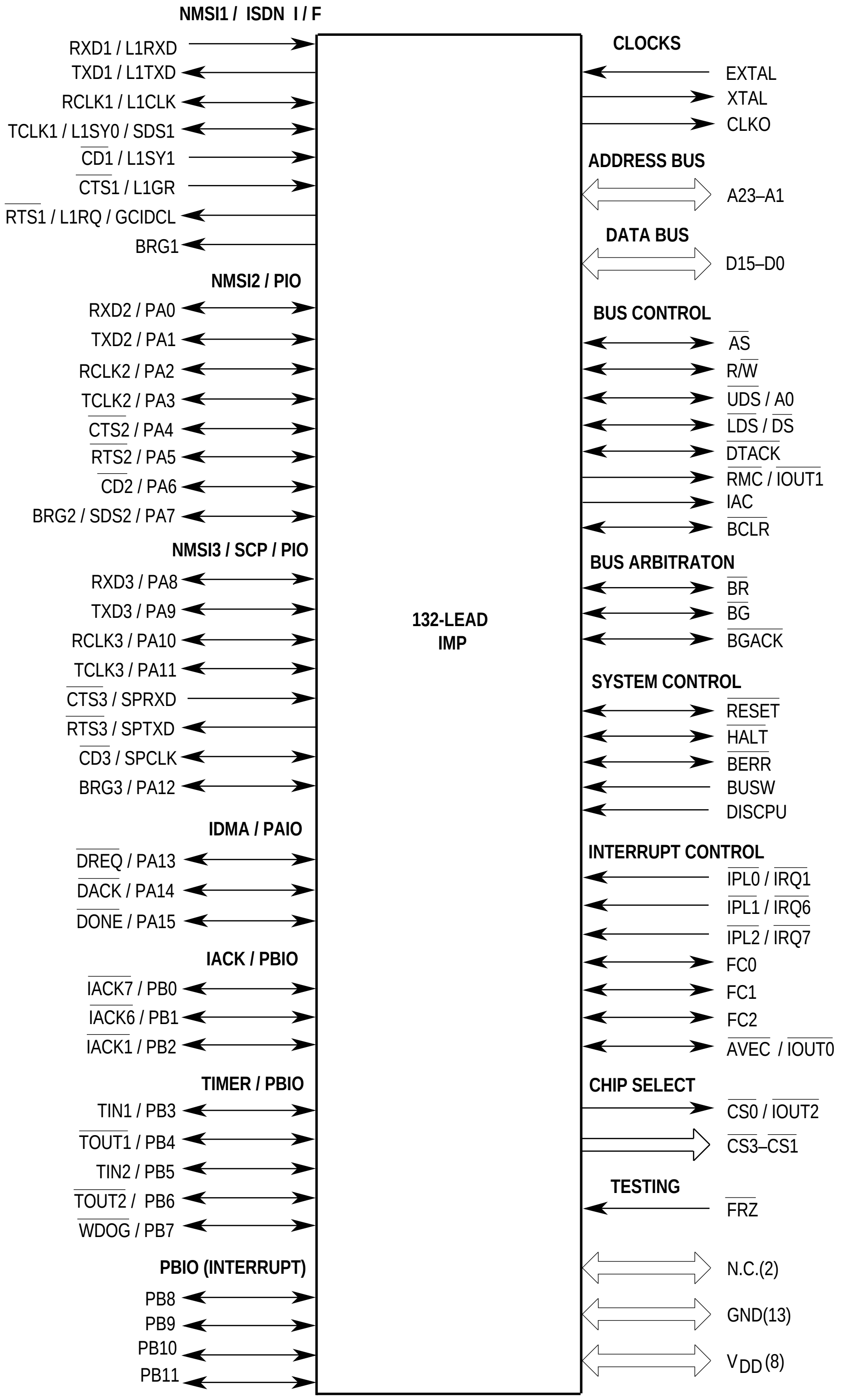


Figure 5-1. Functional Signal Groups

triggers. TCLK2 acts as the SCC2 baud rate generator output if SCC2 is in one of the multiplexed modes.

- RXD2/PA0
- TXD2/PA1
- RCLK2/PA2
- TCLK2/PA3
- $\overline{\text{CTS2}}$/PA4
- $\overline{\text{RTS2}}$/PA5
- $\overline{\text{CD2}}$/PA6
- SDS2/PA7/BRG2

Table 5-9. Baud Rate Generator Outputs

Source	NMSI	GCI	IDL	PCM
SCC1	BRG1	BRG1	BRG1	BRG1
SCC2	BRG2	TCLK2	TCLK2	TCLK2
SCC3	BRG3	TCLK3	TCLK3	TCLK3

NOTE: In NMSI mode, the baud rate generator outputs can also appear on the RCLK and TCLK pins as programmed in the SCON register.

5.15 NMSI3 PORT OR PORT A PINS OR SCP PINS

The NMSI3 port or port A pins or SCP pins are shown in Figure 5-12.

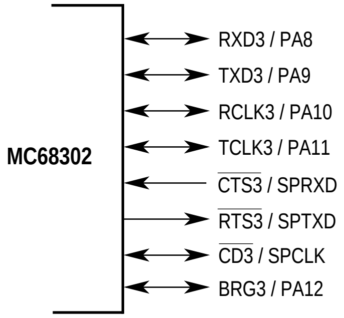
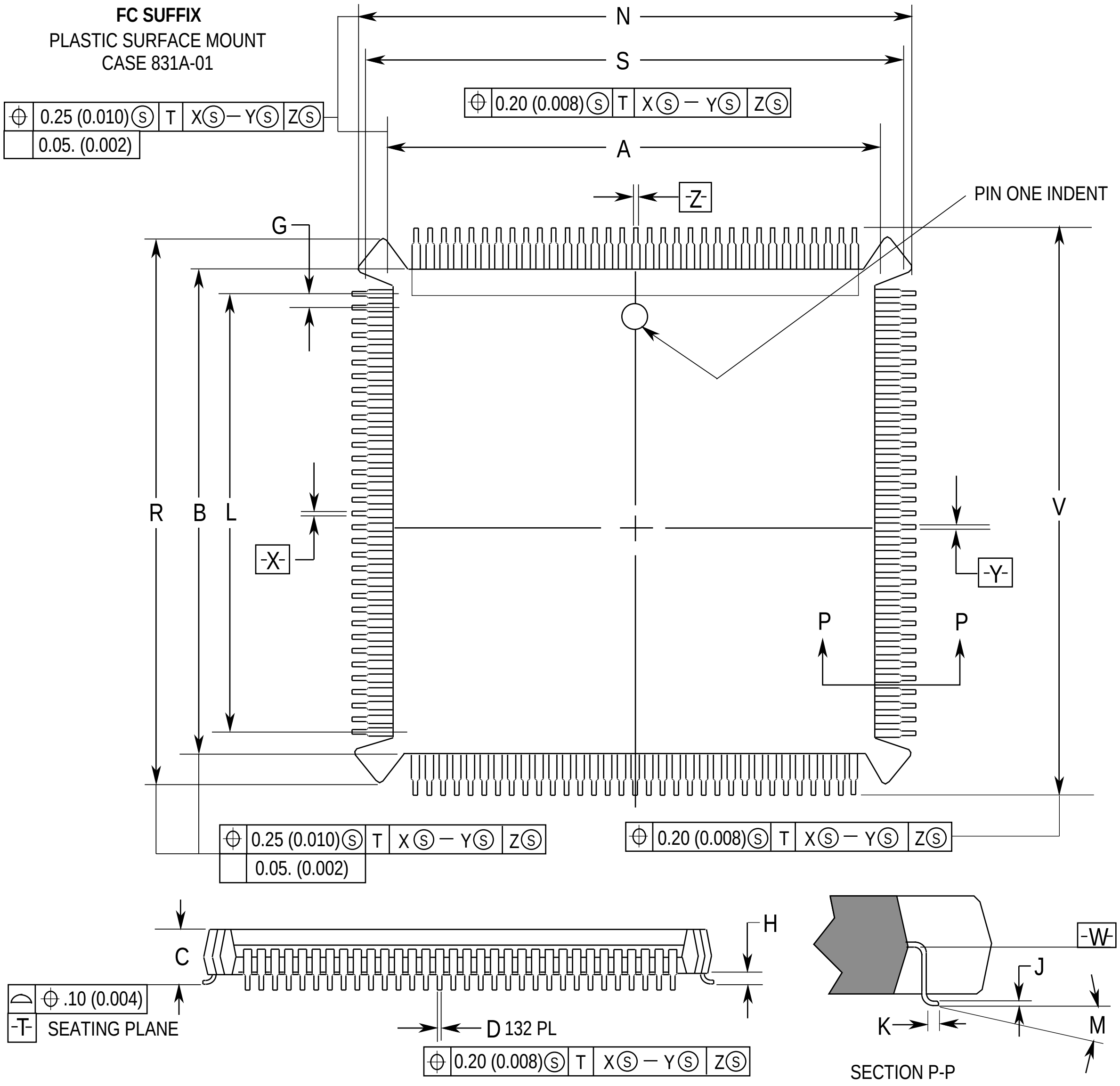


Figure 5-12. NMSI3 Port or Port A Pins or SCP Pins

These eight pins can be used either as the NMSI3 port or as the NMSI3 port (less three modem lines) and the SCP port. If the SCP is enabled (EN bit in SPMODE register is set), then the three lines are connected to the SCP port. Otherwise, they are connected to the SCC3 port.

Each of the port A I/O pins can be configured individually to be general-purpose I/O pins or a dedicated function in NMSI3. When they are used as the NMSI3 pins, they function exactly

7.2.2 Plastic Surface Mount (PQFP)



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
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: INCHES
3. DIM 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. DIM S AND V TO BE DETERMINED AT SEATING PLANE, DATUM -T-.
7. DIM A, B, N AND R TO BE DETERMINED AT DATUM PLANE -W-.

A nibble register write is made by writing one byte with the following format:

7							0
0	A2	A1	A0	D3	D2	D1	D0

A3–A0 Address Register
D3–D0 Data Register

A byte register read is made by writing two bytes with the following format:

1	1	1	1	A3	A2	A1	A0

A3–A0 Address Register

X	X	X	X	X	X	X	X

X–Don't Care

Data read from the register will be received during the second transaction.

A nibble register read is made by writing one byte with the following format:

7							0
1	A2	A1	A0	X	X	X	X

A2–A0 Address Register
X– Don't Care

Data read from the register will be received during the second transaction.

NOTE

The SCP_EN signal must be asserted prior to each SCP transaction and negated after completion.

D.6.13 Additional IMP To S/T Chip Connections

In addition to the IDL bus and the SCP bus, two discrete signals connect the MC145475 S/T chip to the MC68302 (see Figure D-17).

$\overline{\text{IRQ}}$ —The active-low signal sends an interrupt request from the MC145475 to the MC68302 core. This is an active-low signal that is asserted when one or more of the following events occurs:

— Change in the received information state (INFO n) of the S/T receiver.
— Multiframe reception.
D-channel collision.

Each event can be masked and/or cleared by a write/read operation on the corresponding register. The $\overline{\text{IRQ}}$ signal can be connected to the IRQ1 pin of the MC68302 to generate a level 1 interrupt.

$\overline{\text{RESET}}$—This active-low signal initializes the MC145475, forces all internal state machines to the initial state, and forces all internal nibble and byte registers (except BR4 and

D.8.5 Transparent Mode with the NMSI Physical Interface

NMSI has two independent data signals, TXD and RXD, and two independent clocking signals, TCLK and RCLK. TCLK and RCLK may be individually chosen to be generated internally or externally to the MC68302.

NMSI also has three control signals: $\overline{\text{RTS}}$, $\overline{\text{CTS}}$, and $\overline{\text{CD}}$. First, let's discuss their properties in general. The SCC forces $\overline{\text{RTS}}$ low when it is ready to transmit data, but the SCC waits until it sees $\overline{\text{CTS}}$ is low before doing this. After the frame has been transmitted, the $\overline{\text{RTS}}$ signal is negated (high). The $\overline{\text{CTS}}$ signal should stay low during the entire time $\overline{\text{RTS}}$ is low, or transmission is aborted and a $\overline{\text{CTS}}$ lost error is indicated in the transmit buffer descriptor (Tx BD). On the receiving side, the CD signal going low tells the MC68302 to gate data into this SCC. Once low, $\overline{\text{CD}}$ should remain low for the entire frame, or reception is terminated and a $\overline{\text{CD}}$ lost error is signaled in the receive buffer descriptor (Rx BD).

Sometimes the $\overline{\text{CTS}}$ and $\overline{\text{CD}}$ input functions described above are not appropriate for an application. In this case, the software operation mode in the SCC mode register (SCM) can be chosen by programming the DIAG1-DIAG0 bits. In the software operation mode, as far as the SCC is concerned, $\overline{\text{CTS}}$ and $\overline{\text{CD}}$ are always low. However, the real value of the $\overline{\text{CTS}}$ and $\overline{\text{CD}}$ lines externally can be read in the SCCS register once the transmitter and receiver are enabled, and changes in these lines can generate interrupts via the SCCE register. Software operation mode does not affect $\overline{\text{RTS}}$ because, since $\overline{\text{RTS}}$ is an output, $\overline{\text{RTS}}$ can always be ignored by the external logic.

In totally transparent mode (and also BISYNC mode), the $\overline{\text{CD}}$ signal can become a synchronization input. When discussing the $\overline{\text{CD}}$ signal during totally transparent mode in this document, $\overline{\text{CD}}$ will be referred to as "$\overline{\text{CD}}$ (sync)". The totally transparent mode is initiated by setting the EXSYN bit in the SCM. With EXSYN set, a high-to-low transition on $\overline{\text{CD}}$ (sync) defines the start of *both transmission and reception* of transparent mode frames. Subsequent high and low transitions of $\overline{\text{CD}}$ (sync) have *no* effect on the reception of data. The only way to reinitiate the SYNC process is to issue an ENTER HUNT MODE command to the channel, and then force another high-to-low transition on $\overline{\text{CD}}$ (sync).

Figure D-23 shows the simplest NMSI transmit case. NTSYN and EXSYN are set to enable transparent mode, and the L bit is set. Software operation mode (DIAG1 = 1 and DIAG0 = 1) is chosen to eliminate using $\overline{\text{CTS}}$ to control transmission. However, since EXSYN = 1, $\overline{\text{CD}}$ becomes $\overline{\text{CD}}$ (sync), and transmission cannot begin until $\overline{\text{CD}}$ (sync) is low (which can be accomplished by grounding $\overline{\text{CD}}$ (sync)). Thus, there will only be a 1-bit delay between $\overline{\text{RTS}}$ being asserted by the transmitter and actual data being transmitted. Since the L bit is set, $\overline{\text{RTS}}$ is negated after the last byte in the frame.

In the preceding example, if multiple buffers had been ready with their L bits cleared, $\overline{\text{RTS}}$ would have remained asserted, and the next buffer's data would have begun immediately. If multiple buffers had been ready with their L bits set, $\overline{\text{RTS}}$ would have been asserted again after a delay of at least 17 idle bits on the line (the exact number of bits is load dependent).

The L1SY1-L1SY0 signals determine 1) when clocks are sent to the particular SCC and 2) when the synchronization signal is sent to the SCC. If the L1SY1-L1SY0 signals are not active, then the SCC is not being clocked. The rising edge of L1SY1-L1SY0 starts the clocking and sends an internal synchronization pulse to the SCC. These facts are very important in determining when the first byte of real data from a buffer will be transmitted onto the PCM.

In transparent mode, if data is not ready to transmit, $FFs will be sent during the time slot. Once data transfer begins, data will be clocked out during every clock of the time slot. When the time slot ends, the SCC will wait without being clocked until the next time slot arrives. Similarly, on the receive side, data and the clock will only be presented to the SCC when that SCC's time slot is active.

When using transparent mode with PCM, it is often of interest to know exactly when the very first buffer of transmit data will go out. The $\overline{\text{RTS}}$ signal gives an important clue here. Once the $\overline{\text{RTS}}$ signal for an SCC is asserted, the next rising edge of the L1SY1-L1SY0 pins for this channel (i.e., the SCC's next time slot) will begin clocking out the following pattern:

$FF, data1, data2, data3, . . .

where data1 is the first byte of data stored in the transmit data buffer. For example, if the PCM was configured with individual 8-bit time slots for this SCC, $FF would be clocked on the first time slot, data1 on the second, etc. **It is assumed in that this buffer in this example does not immediately follow the previous buffe**r. A string of buffers with their L bits cleared will follow each other immediately without any delay—only the first will have this delay.

From the time the ENT bit is set and a buffer is ready to transmit, it can take a number of serial clocks (usually less than 36) for $\overline{\text{RTS}}$ to be asserted (it could be more for higher data rates). Thus, this clock delay must be taken into account if data transmission delays need to be consistent. The delay can be accounted for by synchronizing the setting of the ENT bit with the time slot itself. If the time slot is long, it should be sufficient to set the ENT bit before one time slot to guarantee data transmission during the next time slot (see Figure D-30). The algorithm can work as follows:

1. After the last buffer is transmitted, give STOP TRANSMIT command.
2. Clear ENT.
3. Give RESTART TRANSMIT command.
4. Set ready bit of next Tx BD to transmit.
5. Generate interrupt to MC68302 on falling L1SY1 /L1SY0 pin.
6. Now that time slot is inactive, set ENT bit.

Rx BD = $5000 $xxxx $xxxx $xxxx (This Rx BD is not yet available.)

13. The final change is to set the ENT and ENR bits in the SCM2 register, causing the transfers to begin.
SCM2 = $603F

With the above configuration, the data in the receive buffers will be as follows:

Buffer1: $FF, data1, data2, ..., datae, dataf.
Buffer2: data10, data11, ..., data17, data18, (and then 7 $FF bytes).

Four events in the SCCE2 event register will be set:

—RX—One or more receive buffers have been used (in this case two).
—TX—One or more buffers have been transmitted (in this case one).
—BSY—Receive data was discarded due to lack of receive buffers (occurred because the third Rx BD was not empty).
—RCH—A word of data has been written to a receive buffer (occurred each time a word was written).

D.8.12 Special Uses of Transparent Mode

The following paragraphs discuss two special cases where transparent mode can be used to extend the capabilities of the MC68302 UART mode.

D.8.12.1 5- OR 6-BIT UART. One special protocol of note that can be accomplished with transparent mode is the building of a 5- or 6-bit UART. The UART on the SCCs offers 7- and 8-bit modes only.

A 5- or 6-bit UART can be accomplished with software and the transparent mode. Software is responsible for inserting and deleting start and stop bits; the transparent mode provides oversampling.

Select transparent mode for 8x the desired bit rate. For every bit of data to transmit, write a byte of data to memory. A character will then be encoded as one byte of zeros for the start bit, a byte of either zeros or ones for every bit in the character, and a byte of ones for each stop bit. When there is no data to send, the transmitter will send out ones during the idle period if the buffer had its L bit set.

Reception is more software intensive. The data is received at 8x the desired rate, and the software must extract the start, stop, and character bits from the data stream. There is no easy way to “byte align” the received data to byte boundaries in memory (as explained in D.8.5 Transparent Mode with the NMSI Physical Interface) without some added external hardware.

D.8.12.2 SYNCHRONOUS UART. A synchronous UART may also be built with transparent mode. As with a 5- or 6-bit UART, all data including start and stop bits must be placed into the transmit buffer and the true data extracted from the raw transparent data that includes start and stop bits. (V.14 applications would also require the detection of deleted stop bits.)