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	33MHz
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/mc68302pv33c

The MC68302 can also be used in applications such as board-level industrial controllers performing real-time control applications with a local control bus and an X.25 packet network connection. Such a system provides the real-time response to a demanding peripheral while permitting remote monitoring and communication through an X.25 packet network.

1.2 FEATURES

The features of the IMP are as follows:

- On-Chip HCMOS MC68000/MC68008 Core Supporting a 16- or 8-Bit M68000 Family-System
- IB Including:
  - —Independent Direct Memory Access (IDMA) Controller with Three Handshake Signals: $\overline{\text{DREQ}}$, $\overline{\text{DACK}}$, and $\overline{\text{DONE}}$.
  - —Interrupt Controller with Two Modes of Operation
  - —Parallel Input/Output (I/O) Ports, Some with Interrupt Capability
  - —On-Chip 1152-Byte Dual-Port RAM
  - —Three Timers Including a Watchdog Timer
  - —Four Programmable Chip-Select Lines with Wait-State Generator Logic
  - —Programmable Address Mapping of the Dual-Port RAM and IMP Registers
  - —On-Chip Clock Generator with Output Signal
  - —System Control:
    - Bus Arbitration Logic with Low-Interrupt Latency Support
    - System Status and Control Logic
    - Disable CPU Logic (M68000)
    - Hardware Watchdog
    - Low-Power (Standby) Modes
    - Freeze Control for Debugging
    - DRAM Refresh Controller
- CP Including:
  - —Main Controller (RISC Processor)
  - —Three Independent Full-Duplex Serial Communications Controllers (SCCs)
  - —Supporting Various Protocols:
    - High-Level/Synchronous Data Link Control (HDLC/SDLC)
    - Universal Asynchronous Receiver Transmitter (UART)
    - Binary Synchronous Communication (BISYNC)
    - Synchronous/Asynchronous Digital Data Communications Message Protocol (DDCMP)
    - Transparent Modes
    - V.110 Rate Adaption
  - —Six Serial DMA Channels for the Three SCCs
  - —Flexible Physical Interface Accessible by SCCs Including:
    - Motorola Interchip Digital Link (IDL)
    - General Circuit Interface (GCI, also known as IOM[3]-2)
    - Pulse Code Modulation (PCM) Highway Interface

[3] IOM is a trademark of Siemens AG

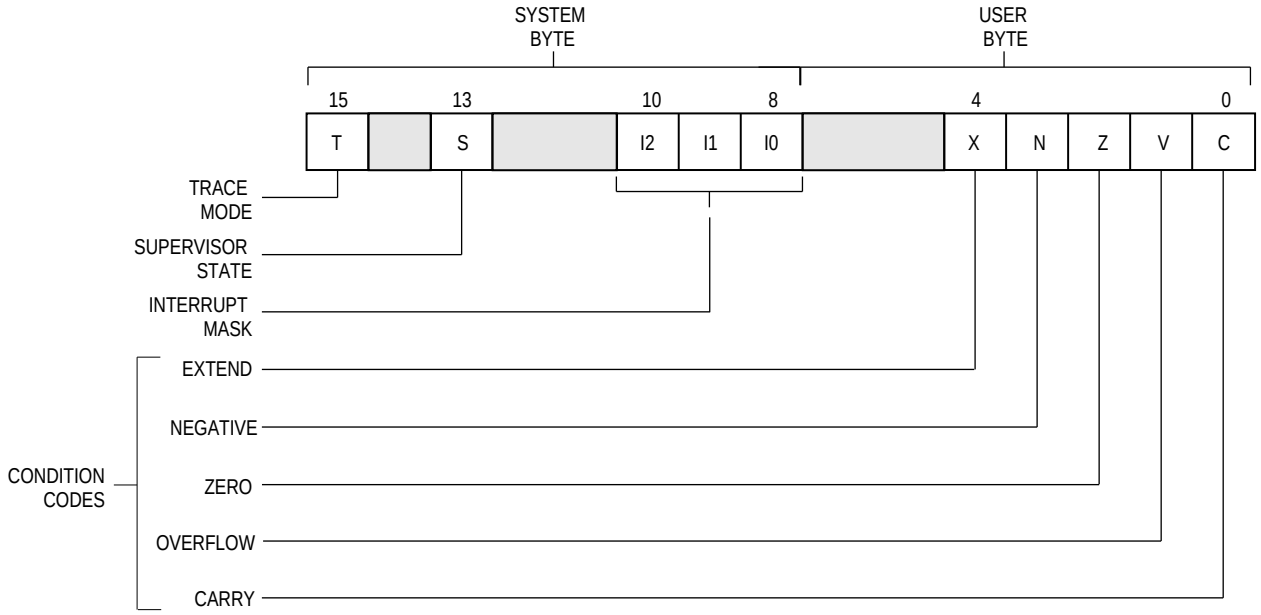


Figure 2-2. M68000 Status Register

2.2 INSTRUCTION SET SUMMARY

The five data types supported by the M68000 on the MC68302 are bits, binary-coded decimal (BCD) digits (4 bits), bytes (8 bits), words (16 bits), and long words (32 bits).

In addition, operations on other data types, such as memory addresses, status word data, etc., are provided for in the instruction set. Shown in Table 2-1, the 14 flexible addressing modes include six basic types:

- Register Direct
- Register Indirect
- Absolute
- Immediate
- Program Counter Relative
- Implied

The capability to perform postincrementing, predecrementing, offsetting, and indexing is included in the register indirect addressing modes. Program counter relative modes can also be modified via indexing and offsetting.

The M68000 instruction set is shown in Table 2-2.

Some basic instructions also have variations as shown in Table 2-3.

Special emphasis has been placed on the instruction set to simplify programming and to support structured high-level languages. With a few exceptions, each instruction operates

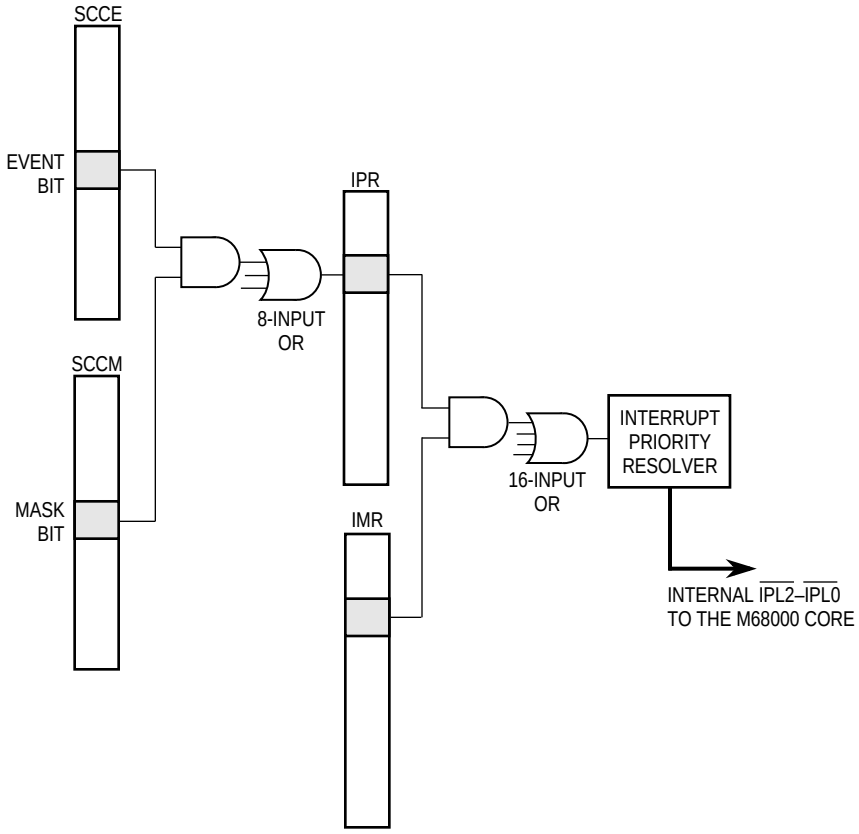


Figure 3-3. Interrupt Request Logic Diagram for SCCs

3.2.4 Interrupt Vector

Pending EXRQ interrupts and unmasked INRQ interrupts are presented to the M68000 core in order of priority. The M68000 core responds to an interrupt request by initiating an interrupt acknowledge cycle to receive a vector number, which allows the core to locate the interrupt's service routine.

If an INRQ source generated the interrupt, the interrupt controller always provides the vector corresponding to the highest priority, unmasked, pending interrupt. If an EXRQ source generated the interrupt, three options are available to generate the vector.

Option 1. By programming the GIMR, the user can enable the interrupt controller to provide the vector for any combination of EXRQ interrupt levels 1, 6, and 7. This is available regardless of whether normal or dedicated mode is selected. Whenever a vector is provided by the interrupt controller, $\overline{\text{DTACK}}$ is also provided by the interrupt controller during that interrupt acknowledge cycle. $\overline{\text{DTACK}}$ is an output from the IMP in this case.

The IMP can generate vectors for up to seven external peripherals by connecting the external request lines to $\overline{\text{IRQ7}}$, $\overline{\text{IRQ6}}$, $\overline{\text{IRQ1}}$, PB11, PB10, PB9, and PB8. PB11, PB10, PB9, and PB8 are prioritized within level 4.

3.2.5 Interrupt Controller Programming Model

The user communicates with the interrupt controller using four registers. The global interrupt mode register (GIMR) defines the interrupt controller's operational mode. The interrupt pending register (IPR) indicates which INRQ interrupt sources require interrupt service. The interrupt mask register (IMR) allows the user to prevent any of the INRQ interrupt sources from generating an interrupt request. The interrupt in-service register (ISR) provides a capability for nesting INRQ interrupt requests.

3.2.5.1 Global Interrupt Mode Register (GIMR)

The user normally writes the GIMR soon after a total system reset. The GIMR is initially $0000 and is reset only upon a total system reset. If bits V7–V5 of the GIMR are not written to specify an interrupt vector prior to the first interrupt condition, the interrupt controller will pass the vector $0F (the uninitialized interrupt vector), regardless of the interrupt source.

15	14	13	12	11	10	9	8	7 5	4 0
MOD	IV7	IV6	IV1	—	ET7	ET6	ET1	V7–V5	RESERVED

MOD—Mode

0 = Normal operational mode. Interrupt request lines are configured as $\overline{IPL2}$–$\overline{IPL0}$.
1 = Dedicated operational mode. Interrupt request lines are configured as $\overline{IRQ7}$, $\overline{IRQ6}$, and $\overline{IRQ1}$.

IV7—Level 7 Interrupt Vector

This bit is valid in both normal and dedicated modes.

0 = Internal vector. The interrupt controller will provide the vector number for a level 7 interrupt during the interrupt acknowledge cycle.
1 = External vector. The interrupt controller will not provide the vector number for a level 7 interrupt.

IV6—Level 6 Interrupt Vector

This bit is valid in both normal and dedicated modes.

0 = Internal vector. The interrupt controller will provide the vector number for a level 6 interrupt during the interrupt acknowledge cycle.
1 = External vector. The interrupt controller will not provide the vector number for a level 6 interrupt.

IV1—Level 1 Interrupt Vector

This bit is valid in both normal and dedicated modes.

0 = Internal vector. The interrupt controller will provide the vector number for a level 1 interrupt acknowledge cycle.
1 = External vector. The interrupt controller will not provide the vector number for a level 1 interrupt.

Port A Control Register(PACNT

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

0 = I/O 1 = Peripheral

Port A Data Direction Register(PADDR)

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

0 = Input 1 = Output

Port A Data Register(PADAT)

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

Port B Control Register(PBCNT)

15	8	7	6	5	4	3	2	1	0
RESERVED		CB	CB	CB	CB	CB	CB	CB	CB

0 = I/O 1 = Peripheral

Port B Data Direction Register(PBDDR)

15	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		DB	DB	DB	DB	DB	DB	DB	DB	DB	DB	DB	DB

0 = Input 1 = Output

Port B Data Register(PBDAT)

15	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PB	PB	PB	PB	PB	PB	PB	PB	PB	PB	PB	PB

Figure 3-6. Parallel I/O Port Registers

3.4 DUAL-PORT RAM

The CP has 1152 bytes of static RAM configured as a dual-port memory. The dual-port RAM can be accessed by the CP main controller or by one of three bus masters: the M68000 core, the IDMA, or an external master. The M68000 core and the IDMA access the RAM synchro-

transmission, message transmission is aborted after the contents of the FIFO (up to three bytes) are transmitted. The TBD# is not advanced. No new BD is accessed, and no new messages are transmitted for this channel. Upon receipt of this command, the transmitter aborts the message transmission (if currently transmitting) and then transmits SYN1–SYN2 pairs or IDLEs as determined by the DDCMP mode register.

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

RESTART TRANSMIT Command

The RESTART TRANSMIT command re-enables the transmission of characters on the transmit channel. This command is expected by the DDCMP controller after a STOP TRANSMIT command, after a STOP TRANSMIT command followed by the disabling of the channel in its SCC mode register, or after a transmitter error (underrun or CTS lost during data or maintenance message header fields). The DDCMP controller will resume transmission from the current transmitter BD number (TBD#) in the channel's transmit BD table.

If the channel 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 in the 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 DDCMP controller to abort reception of the current message, generate an RBD interrupt (if enabled) as the buffer is closed, and enter hunt mode. In hunt mode, the DDCMP controller continually scans the input data stream for the SYN1–SYN2 sequence on synchronous links. Then for synchronous or asynchronous links, the DDCMP controller scans the input bytes for the starting byte of one of the messages. After receiving the command, the current receive buffer is closed, and the CRC is reset. Message reception continues using the next BD.

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.

4.5.14.6 DDCMP Control Character Recognition

The DDCMP controller can recognize three special control characters. These characters are used to synchronize the message and allow the DDCMP controller to function in a DMA-controlled environment.

DSYN1—DDCMP Sync Character Register

The 8-bit DSYN1 register should be written with the same value that was written in the SYN1 byte of the data synchronization register (DSR). DSYN1 is a memory-mapped read-write register.

NOTE

For correct operation of DDCMP, DSYN1, SYN1, and SYN2 must be the same value.

DSOH—DDCMP SOH Register

The 8-bit DSOH register is used to synchronize data messages by the DDCMP controller. When the DDCMP controller is not in hunt mode (byte synchronization is now established), it searches for the SOH character to start processing data messages. The DDCMP controller transfers the header and the data fields of the message to the buffer, checks the header and data CRCs, counts the data field up to the value contained in the header byte count field, and compares the header address field against the user-defined addresses. The DSOH register is a memory-mapped read-write register.

DENQ—DDCMP ENQ Register

The 8-bit DENQ register is used to synchronize control messages by the DDCMP controller. When the DDCMP controller is not in hunt mode (byte synchronization is established), it searches for the ENQ character to start processing control messages. The DDCMP controller transfers the message to the buffer, checks the CRC, and compares the message address field against the user-defined addresses. The DENQ register is a memory-mapped read-write register.

DDLE—DDCMP DLE Register

The 8-bit DDLE register is used to synchronize maintenance messages by the DDCMP controller. When the DDCMP controller is not in hunt mode (byte synchronization is established), it searches for the DLE character to start processing the maintenance messages. The DDCMP controller transfers the header and the data fields of the message to the buffer, checks the header and data CRCs, counts the data field up to the value contained in the header byte count field, and compares the header address field against the user-defined addresses. The DDLE register is a memory-mapped read-write register.

4.5.14.7 DDCMP Address Recognition.

Each DDCMP controller has five 16-bit registers to support address recognition: one mask register and four address registers (DMASK, DADDR1, DADDR2, DADDR3, and DADDR4). The DDCMP controller reads the message address from the receiver, masks it with the user-defined DMASK bits, and then checks the result against the four address register values. A one in DMASK indicates a bit position where a comparison should take place; a zero masks the comparison. For 8-bit address comparison, the high byte of DMASK should be zero.

4.5.14.8 DDCMP Error-Handling Procedure

The DDCMP controller reports message reception and transmission errors using the channel BDs, the error counters, and the DDCMP event register. The modem interface lines can also be directly monitored with the SCC status register.

Transmission errors:

1. Transmitter Underrun. When this error occurs, the channel terminates buffer transmission, closes the buffer, sets the underrun (UN) bit in the BD, and generates the transmit error (TXE) interrupt (if enabled). The channel will resume transmission after the reception of the RESTART TRANSMIT command. The FIFO size is three bytes.

Data Length

The data length is the number of octets that the DDCMP controller should transmit from this BD's data buffer. It is never modified by the CP. The data length should be greater than zero.

Tx Buffer Pointer

This pointer, which contains the address of the associated data buffer, may be even or odd. The buffer may reside in either internal or external memory.

NOTE

For correct operation of the function codes, the upper 8 bits of the pointer must be initialized to zero.

4.5.14.12 DDCMP Event Register

The SCC event register (SCCE) is referred to as the DDCMP event register when the SCC is configured for DDCMP. It is an 8-bit register used to report events recognized by the DDCMP channel and to generate interrupts. On recognition of an event, the DDCMP controller sets its corresponding bit in this register. Interrupts generated by this register may be masked in the DDCMP mask register.

The DDCMP event register is a memory-mapped register that may be read at any time. A bit is cleared by writing a one (writing zero does not affect a bit's value). More than one bit may be cleared at a time. All unmasked bits must be cleared before the CP will clear the internal interrupt request. This register is cleared at reset.

7	6	5	4	3	2	1	0
CTS	CD	—	TXE	RBK	BSY	TX	RBD

CTS—Clear-To-Send Status Changed

A change in the status of the $\overline{\text{CTS}}$ line was detected on the DDCMP channel. The SCC status register may be read to determine the current status.

CD—Carrier Detect Status Changed

A change in the status of the CD line was detected on the DDCMP channel. The SCC status register may be read to determine the current status.

Bit 5—Reserved for future use.

TXE—Tx Error

An error (CTS lost or underrun) occurred on the transmitter channel.

RBK—Receive Block

A complete block has been received on the DDCMP channel. A block is defined as reception of a complete header, a complete message, or a receiver error condition.

BSY—Busy Condition

A data byte was received and discarded due to lack of buffers. The receiver will enter hunt mode automatically.

Table 5-1. Signal Definitions

Chip Select	CS3–CS0	4
Testing	$\overline{FRZ}$ (2 Spare)	3
V_{DD}		8
GND		13

All pins except EXTAL, CLKO, and the layer 1 interface pins in IDL mode support TTL levels. EXTAL, when used as an input clock, needs a CMOS level. CLKO supplies a CMOS level output. The IDL interface is specified as a CMOS electrical interface.

All outputs (except CLKO and the GCI pins) drive 130 pF. CLKO is designed to drive 50 pF. The GCI output pins drive 150 pF.

5.2 POWER PINS

The IMP has 21 power supply pins. Careful attention has been paid to reducing IMP noise, potential crosstalk, and RF radiation from the output drivers. Inputs may be +5 V when V_{DD} is 0 V without damaging the device.

- V_{DD} (8)—There are 8 power pins.
- GND (13)—There are 13 ground pins.

NOTE

The Input High Voltage and Input Low Voltage for EXTAL and the values for power are specified in the DC Electrical Characteristics.A valid clock signal oscillates between a low voltage of between V_{SS}-0.3 and .6 volts and a high voltage of between 4.0 and V_{DD} volts. This EXTAL signal must be present within 20 mS after V_{DD} reaches its minimum specified level of 4.5 volts.

6.6 DC ELECTRICAL CHARACTERISTICS—NMSI1 IN IDL MODE

Characteristic	Symbol	Min	Max	Unit	Condition
Input Pin Characteristics: L1CLK, L1SY1, L1RXD, L1GR					
Input Low Level Voltage	V_{IL}	-10%	+ 20%	V	(% of V_{DD})
Input High Level Voltage	V_{IH}	V_{DD} - 20%	V_{DD} + 10%	V	
Input Low Level Current	II_L	—	± 10	µA	$V_{in} = V_{ss}$
Input High Level Current	II_H	—	± 10	µA	$V_{in} = V_{DD}$
Output Pin Characteristics: L1TXD, SDS1- SDS2, L1RQ					
Output Low Level Voltage	V_{OL}	0	1.0	V	I_{OL} = 5.0 mA
Output High Level Voltage	V_{OH}	V_{DD} - 1.0	V_{DD}	V	I_{OH} = 400 µA

6.7 AC ELECTRICAL SPECIFICATIONS—CLOCK TIMING

(see Figure 6-1, Figure 6-2, Figure 6-3, and Figure 6-4)

			16.67 MHz		20 MHz		25 MHz		
Num.	**Characteristic**	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**
	Frequency of Operation	f	8	16.67	8	20	8	25	MHz
1	Clock Period (EXTAL) (See note 3)	t_{cyc}	60	125	50	125	40	125	ns
2, 3	Clock Pulse Width (EXTAL)	t_{CL}, t_{CH}	25	62.5	21	62.5	16	62.5	ns
4, 5	Clock Rise and Fall Times (EXTAL)	t_{Cr}, t_{Cf}	—	5	—	4	—	4	ns
5a	EXTAL to CLKO Delay (See Notes 1 and 2)	t_{CD}	2	11	2	9	2	7	ns

NOTE:
1. CLKO loading is 50 pF max.
2. CLKO skew from the rising and falling edges of EXTAL will not differ from each other by more than 1 ns, if the EXTAL rise time equals the EXTAL fall time.
3. You may not stop the clock input at any time.

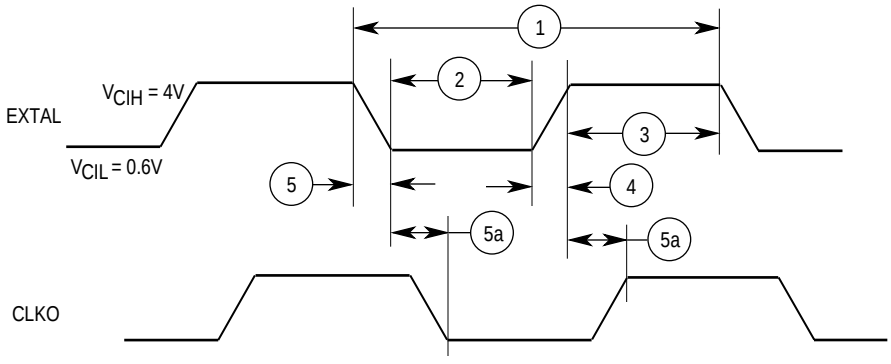
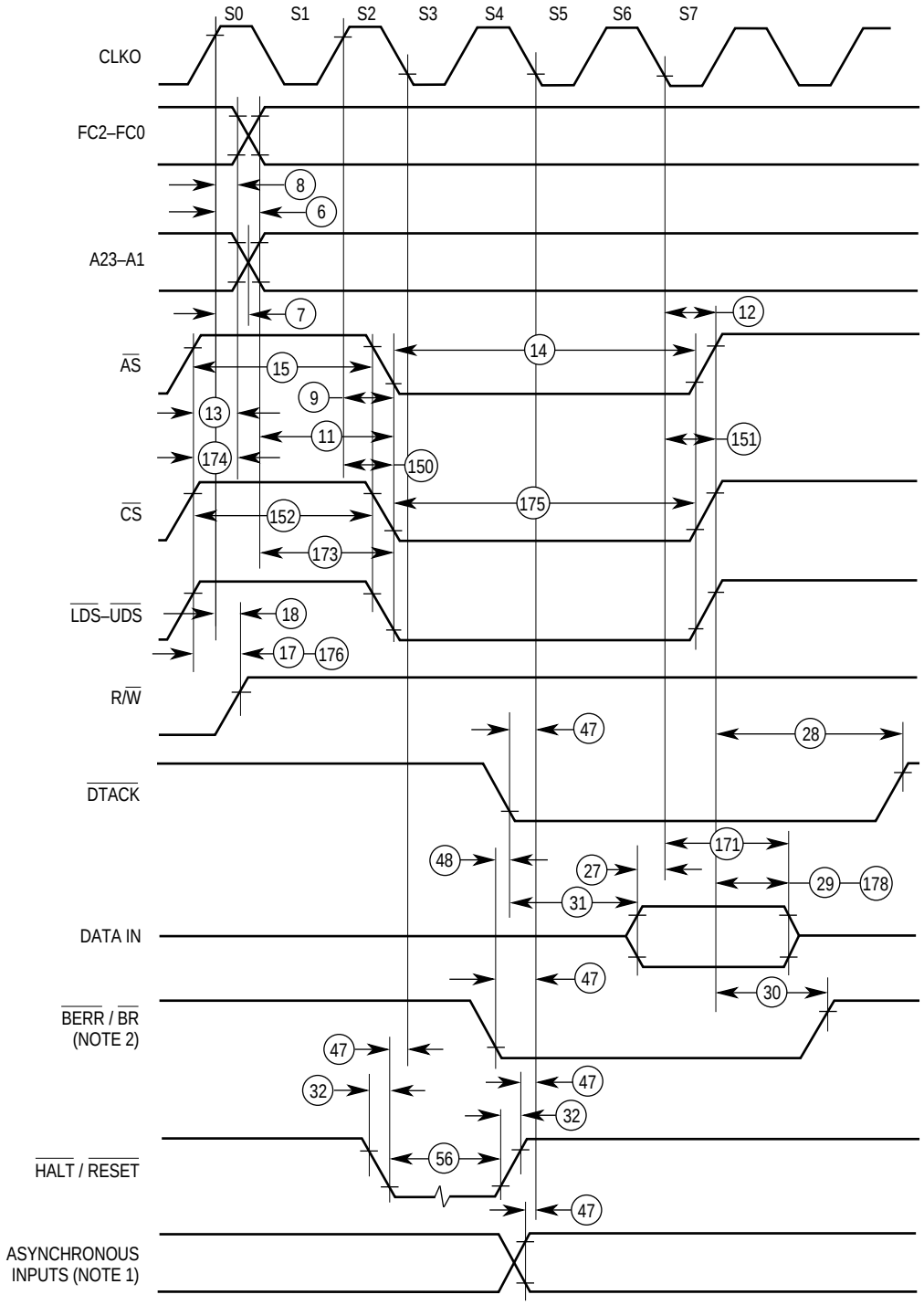


Figure 6-1. Clock Timing Diagram



NOTES:
1. Setup time for the asynchronous inputs IPL2–IPL0 guarantees their recognition at the next falling edge of the clock.
2. BR need fall at this time only to insure being recognized at the end of the bus cycle.
3. Timing measurements are referenced to and from a low voltage of 0.8 volt and a high voltage of 2.0 volts, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall is linear between 0.8 volts and 2.0 volts.

Figure 6-2. Read Cycle Timing Diagram

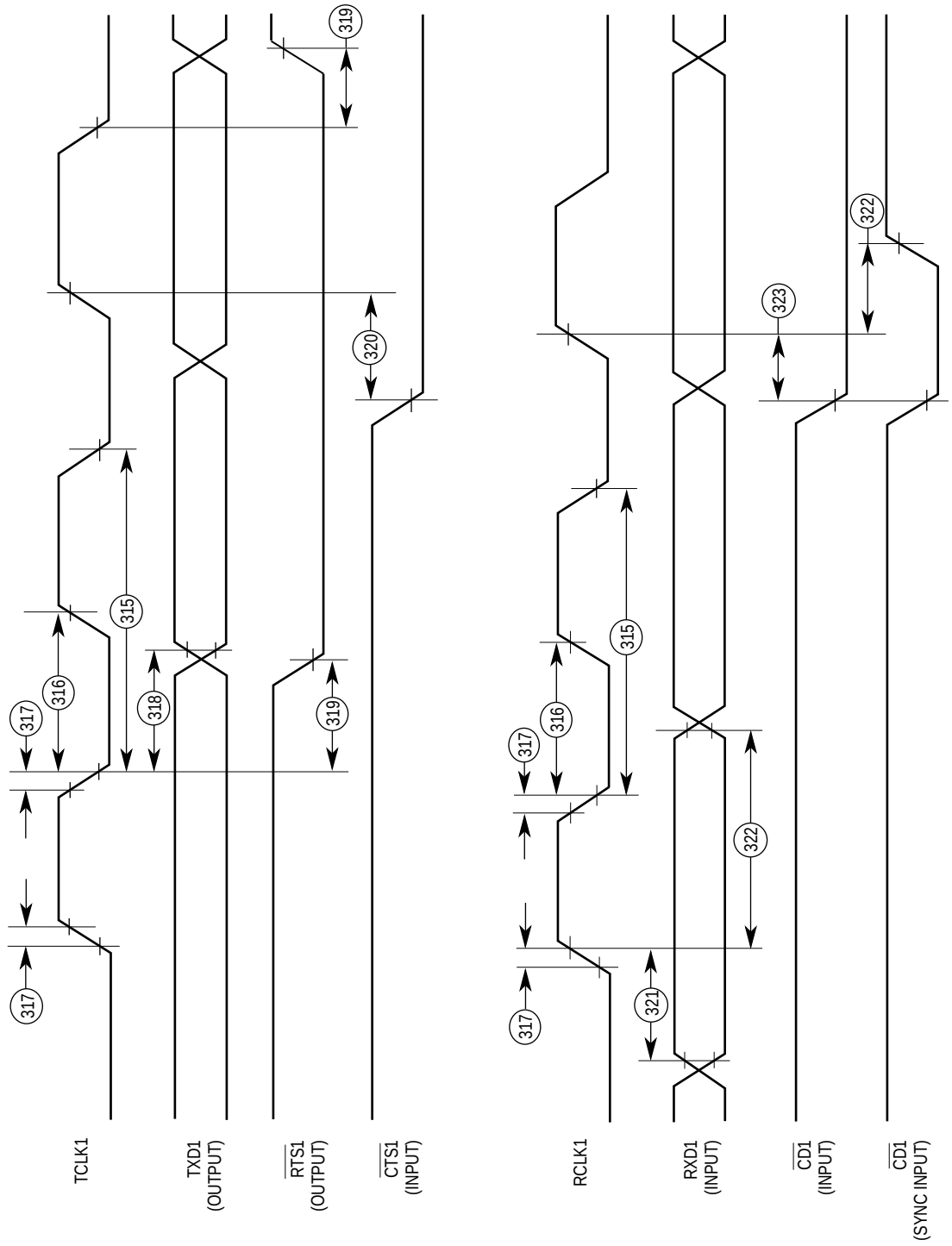


Figure 6-24. NMSI Timing Diagram

APPENDIX B
DEVELOPMENT TOOLS AND SUPPORT

B.1 MOTOROLA SOFTWARE OVERVIEW

A software development package is offered as a set of independent modules which provides the following features:

- IMP Chip Evaluation
  A development board, described in B.5 302 Family ADS System, can be used as a prototyping vehicle for user code.
- IMP Chip Drivers
  Written in C, the source code of the IMP drivers is available on electronic media. These drivers were developed to support the needs of the protocol implementations described below.
- Protocol Implementations
  Modules implementing common ISO/OSI layer 2 and 3 protocols are available under license. These are in the form of binary relocatable object code as well as source code written in C.
- Portability
  All of the modules may be ported to different MC68302 implementations and combined with user-developed code. Well-defined software interface documentation is available for all provided modules.

B.2 MOTOROLA SOFTWARE MODULES

Chip driver routines written in C illustrate initialization of the IMP, interrupt handling, and the management of data transmission and reception on all channels.

In addition to the chip drivers, protocol modules are provided. Layer 2 modules include LAPB and LAPD. (Support for layer 2 BISYNC and DDCMP functions are also provided with the chip driver package.) The layer 3 module supported is the X.25 packet layer protocol.

Since the modules do require some minimal operating system services, they are written to use our EDX operating system. Use of EDX with the protocol modules is not required as long as some other operating system support is provided by the user.

All modules communicate with each other using message passing. This technique simplifies the interfaces between the different modules and enables them to interface with other user-added modules with relative ease. Detailed descriptions of the software interfaces are available.

```
* Set up PACNT, PBCNT, etc., ignore for this example, only SCC1 is used
* Select Serial Interface Mode: normal operation, NMSI mode
          MOVE.W    #0,SIMODE          ;Same as default after reset

*** SCC1 Initialization ***
*  Interrupt Vector: SCC1 interrupt handler is at INT_VEC = $31000
*  v7-v5 =5, v4-v0 = $ad -> vector = $ad -> Exception vector = ($ad<<2) = $2b4
          MOVE.L    #INT_VEC, $02B4

* Determine Configuration
* Use Baud Rate Generator for transmit and receive, Rate is 130 kbps.
          MOVE.W    #$07E,SCON1

* Select SCC Mode
* HDLC, Loopback mode, CRC16, RTS negate between frames, NRZ mode.
          MOVE.W    #$10,SCM1

* Set up Parameter RAM
          MOVE.W    #0,FCR_1           ; Clear RFCR and TFCR
          MOVE.W    #$08,MRBLR_1       ; Max Buffer Length = 8
          MOVE.W    #$F0B8,CMSKL_1     ; 16 bit CRC
          MOVE.W    #$070,MFLR_1       ; Max Frame Length = $70 bytes
          MOVE.W    #0,HMASK_1         ; Do not check address
          MOVE.W    #0,DISFC_1         ; Clear the counter
          MOVE.W    #0,CRCEC_1         ; Clear the counter
          MOVE.W    #0,ABTSC 1         ; Clear the counter
          MOVE.W    #0,NMARC_1         ; Clear the counter
          MOVE.W    #0,RETRC_1         ; Clear the counter

* Clear Event Register
          MOVE.B    #$FF,SCCE1

* Determine Maskable Interrupt Events by setting SCCM
* Allow the following interrupt: TXE, RFX, TXB, and RFB
          MOVE.B    #$1B,SCCM1

* Clear M68000 data registers
          CLR.L     D0
          CLR.L     D1
          CLR.L     D2
          CLR.L     D3
          CLR.L     D4
          CLR.L     D5

***Prepare Buffer Descriptors ***
*SCC1 Rx Buffer Descriptors Initialization values before execution:
*00700400 D000 0000 0003 0000 D000 0000 0003 0010
*00700410 D000 0000 0003 0020 D000 0000 0003 0030
*00700420 D000 0000 0003 0040 D000 0000 0003 0050
*00700430 D000 0000 0003 0060 F000 0000 0003 0070
          LEA.L     RXBD_01,A0         ;A0 points to the first RXBD of SCC1
          LEA.L     RXBF_01,A1         ;A1 points to the first buffer
          MOVE.W    #$D000,D1          ;D1 is used for setting the status of BD
*                                      ;Empty = 1, External = 1, Int=1
          MOVE.W    #$F000,D2          ;D2 is for the last BD, Wrap = 1
```

D.6.1 M68000 Core

The M68000 core processor on the MC68302 is instruction and timing compatible with the standard MC68000 (16-bit) or MC68008 (8-bit) versions of the M68000 Family. The core supports bus lock during read-modify-write cycles, a low latency interrupt mechanism, and bus width configuration. It does not support the older M6800 peripherals.

D.6.2 Communications Processor

The communication processor consists of a RISC processor, three serial communication controllers (SCCs), six DMA channels for the three SCCs, a programmable physical interface, a programmable serial communication port (SCP), and two serial management controllers (SMCs). The RISC processor, a separate processor from the M68000 core processor, is dedicated to the service of the SCCs, SCP, and SMCs.

The MC68302 supports three, full-duplex, independent SCCs, which support HDLC, UART, BISYNC, DDCMP, and V.110 protocols as well as transparent mode.

The physical interface supports a standard nonmultiplexed interface for each of the three SCCs (TXD, RXD, TCLK, RCLK, CTS, RTS, and CD) as well as several multiplexed modes.In multiplexed modes, up to three SCCs can be time-multiplexed onto the same serial channel. The multiplexed modes include IDL, GCI, and PCM highway.

The SCP is a full-duplex, synchronous, character-oriented channel that provides a three-wire interface. It is used to control and program SPI-type devices. The SCP implements a subset of Motorola's SPI interface.

The two SMCs are used to exchange control information multiplexed with the 2B + D data in the IDL or GCI buses.

D.6.3 System Integration Block

The system integration block incorporates general-purpose peripherals that eliminate the glue logic found in most M68000 systems. It includes an independent DMA controller (IDMA), an interrupt controller, parallel I/O ports, 1152-byte dual-port RAM, two timers, one watchdog timer, chip-select lines and wait-state generation logic, a bus arbiter, low power modes, core disable logic, on-chip clock generator, and a hardware watchdog.

D.6.4 IDL Bus

The IDL was developed to maximize the portability of the various chips required in an ISDN system. It provides a consistent interface definition across which a family of ISDN chips will be able to transport data (see Figure D-12).

The IDL bus connects one IDL master to one or more IDL slaves. The IDL bus timing is driven by the master device.

The bus signals are as follows:

- CLOCK — always provided from the master to the slave. It provides the bit timing for the data traveling across the IDL.
- SYNC —provides the framing for the IDL. The SYNC occurs once per 125-µsec frame and is active high one full clock cycle in the bit immediately preceding the data transaction.
- TXDATA — drives the data from one chip to another. The line is in high impedance when no data transaction occurs.
- RXDATA — the input line that receives data from the TXDATA of another part.

D.6.6 IMP/IDL Interconnection

The MC68302 directly connects to the IDL bus with no glue logic. The MC68302 is an IDL slave (accepts IDL timing from the bus). In the application described, the IDL master device is the MC145475 S/T interface chip (see Figure D-14).

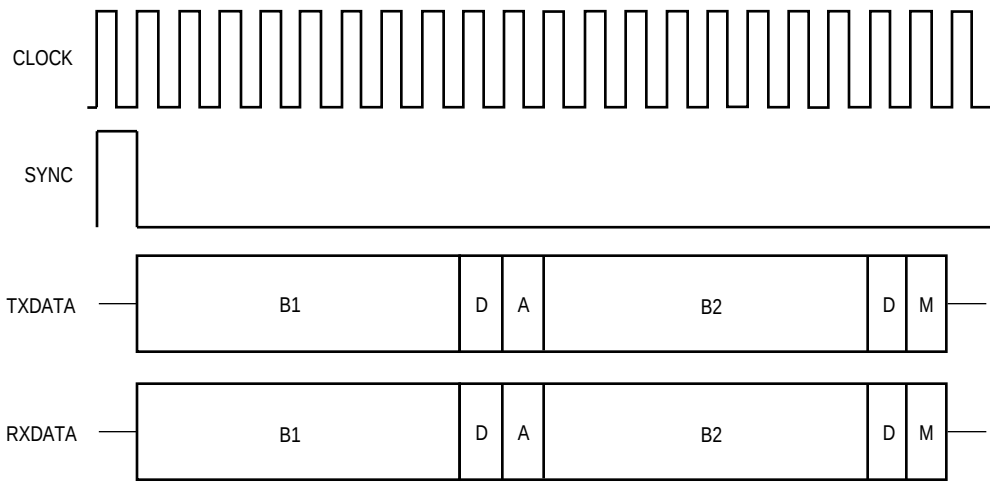
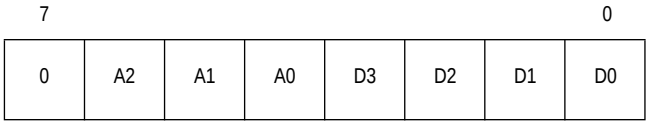


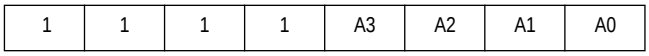
Figure D-13. IDL Frame Structure

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

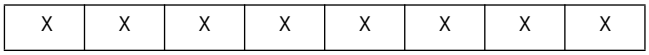


A3–A0 Address Register
D3–D0 Data Register

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



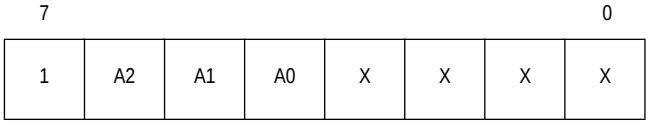
A3–A0 Address Register



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:



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

TX—Tx Buffer

0 = No interrupt.
1 = A buffer has been transmitted on the UART channel (set only if the I bit in the Tx buffer descriptor is set).

RX—Rx Buffer

0 = No interrupt.
1 = A buffer was received on the UART channel (set only if the I bit in the Rx buffer descriptor is set).

E.2.1.2.5 UART Mask Register (SCCM). This 8-bit register is located at offset $88A (SCC1), $89A (SCC2), and $8AA (SCC3) on D15-D8 of a 16-bit data bus. The SCCM is used to enable and disable interrupt events reported by the SCCE. The mask bits correspond to the interrupt event bit shown in the SCCE. A bit should be set to one to enable the corresponding interrupt in the SCCE.

7	6	5	4	3	2	1	0
CTS	CD	IDL	BRK	CCR	BSY	TX	RX

E.2.1.2.6 UART Status Register (SCCS). This 8-bit register is located at offset $88C (SCC1), $89C (SCC2), and $8AC (SCC3), on D15-D8 of a 16-bit data bus. The SCCS register reflects the current status of the RXD, $\overline{CD}$, and $\overline{CTS}$ lines as seen by the SCC.

7	6	5	4	3	2	1	0
—	—	—	—	—	ID	CD	CTS

ID—Idle Status on the Receiver Line (valid only when the ENR bit is set and the receive clock is running)

0 = Receiver Line is not idling.
1 = Either $\overline{CD}$ is not asserted or the receiver line is idling while $\overline{CD}$ is asserted.

$\overline{CD}$—Carrier Detect Status Changed (valid only when the ENR bit is set and the receive clock is running)

0 = $\overline{CD}$ is asserted.
1 = $\overline{CD}$ is not asserted.

$\overline{CTS}$—Clear-To-Send Status Changed (valid only when the ENT bit is set and the transmit clock is running)

0 = $\overline{CTS}$ is asserted.
1 = $\overline{CTS}$ is not asserted.

E.2.1.3 GENERAL AND UART PROTOCOL-SPECIFIC PARAMETER RAM. Each SCC has 32 words of parameter RAM used to configure receive and transmit operation, store temporary parameters for the CP, and maintain counters. The first 14 words are general parameters, which are the same for each protocol. The last 18 words are specific to the protocol selected. The following sections discuss the parameters that the user must initialize to configure the UART operation.