DEVICE OPERATION AND SYSTEM IMPLEMENTATION OF THE ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (MC6850)

Prepared by
Karl Fronheiser
Computer Applications Engineering

This application note provides ACIA operational information beyond that included in the data sheet, specifically, information on power-on reset/master reset operation and status register operation. System implementation examples and their associated software are also illustrated and discussed. One of these examples is a data communication application using the MC6860 Modem.

DEVICE OPERATION AND SYSTEM IMPLEMENTATION OF THE ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (MC6850)

INTRODUCTION

Microcomputer systems must be provided with an efficient means of communicating with peripheral equipment such as modems, teletypes, CRT terminals, and keyboard/printers. The microcomputer manipulates parallel data byte information at high speeds relative to the slow speed asynchronous data format required for communicating with peripherals. Therefore, an efficient interface adapter to convert the processor parallel data byte information into a serial asynchronous data format and vice-versa is a highly desirable system function. This relieves the microprocessor of this time-consuming task. A device providing the above data formatting/interface function is the MC6850 Asynchronous Communications Interface Adapter (ACIA). One side of the ACIA is directly compatible with Motorola's MC6800 (MPU) microprocessor bus while the other side is compatible with peripherals that use an asynchronous data format.

The asynchronous data format characteristics are used by the ACIA to establish bit and character synchronization in the absence of a clock that has been pre-synchronized to the data. An asynchronous data format consists of a serial bit stream with the data bits preceded by a start bit and followed by one or more stop bits. The ACIA converts a character which was serially received from peripheral equipment to a parallel byte with the start, stop, and parity bits deleted from the character. Also, the parallel bytes from the microprocessor are converted to a serial form with start, stop, and optional parity bits appended to the character. Performing these functions in hardware outside of the processor enables the microprocessor to more efficiently communicate with peripheral equipment by using a minimum of software overhead.

The ACIA consists of control, status, transmit data and receive data registers; data bus buffers; transmit and receive shift registers; and peripheral control as shown in the block diagram of Figure 1. Since basic operational information on the ACIA is contained in the ACIA data sheet, this application note will provide additional information to supplement the data sheet with a minimum of repetitive information. The first section of this note provides a description of the operation of the transmitter and receiver portions of the ACIA with reference to appropriate timing diagrams. The second section covers the aspects of the power-on and master reset functions for initialization of the ACIA. The third section covers a detailed description of the ACIA status register bits. The fourth section covers a system implementation of the ACIA as a data communications link in a microcomputer based system. The last section provides examples of the software requirements for initializing the ACIA, and the transmit/receive subroutines for the transmission of data. Additional application information on Motorola's MPU family is available in the "M6800 Microprocessor Applications Manual."

TRANSMITTER/RECEIVER OPERATION

This section covers the internal transmitter/receiver operation of the ACIA as well as the timing relationship between characters being transmitted or received and their associated status register bits. It should be noted that prior to the transmission and/or reception of data, the ACIA must be initialized as described in the "Power-on Reset/Master Reset" section.

Data is transferred to/from the four internal registers of the ACIA on the trailing transition (negative edge) of the signal on the enable input (E). For example, a write data command (RS = 1, R/W = 0) transfers data into the transmit data register on the trailing transition of the enable input signal. In a typical MPU based system, the enable input signal is generated from the ANDing of the Valid Memory Address (VMA) and $\phi 2$.

Transmitter

In a typical transmitting sequence, a character is written into the Transmit Data Register (TDR) if a status read operation indicated the TDR was Empty (TDRE). The write data command (trailing edge of the enable pulse) causes the TDRE status bit to go "low" indicating a transmitter data register full condition. During an idling (absence of data transmission) condition, the transfer of data from the TDR to the transmit shift register will take place within one data bit time. This results in a delay (due to internal operation of the ACIA) in the transmission of the character from the Transmit Data Output with respect to the write data command of one to two data bit times as shown in Figure 2. The trailing edge of the internal transfer signal returns the TDRE status bit to a "high" level indicating a Transmitter Data Register empty condition. The transmitter shift register serializes the data and transmits the data bits, starting with data bit D0, preceded by a start bit and followed by one or two stop bits. Also, internal parity (odd or even) can be optionally added by the ACIA and will occur between the last data bit and the first stop bit.

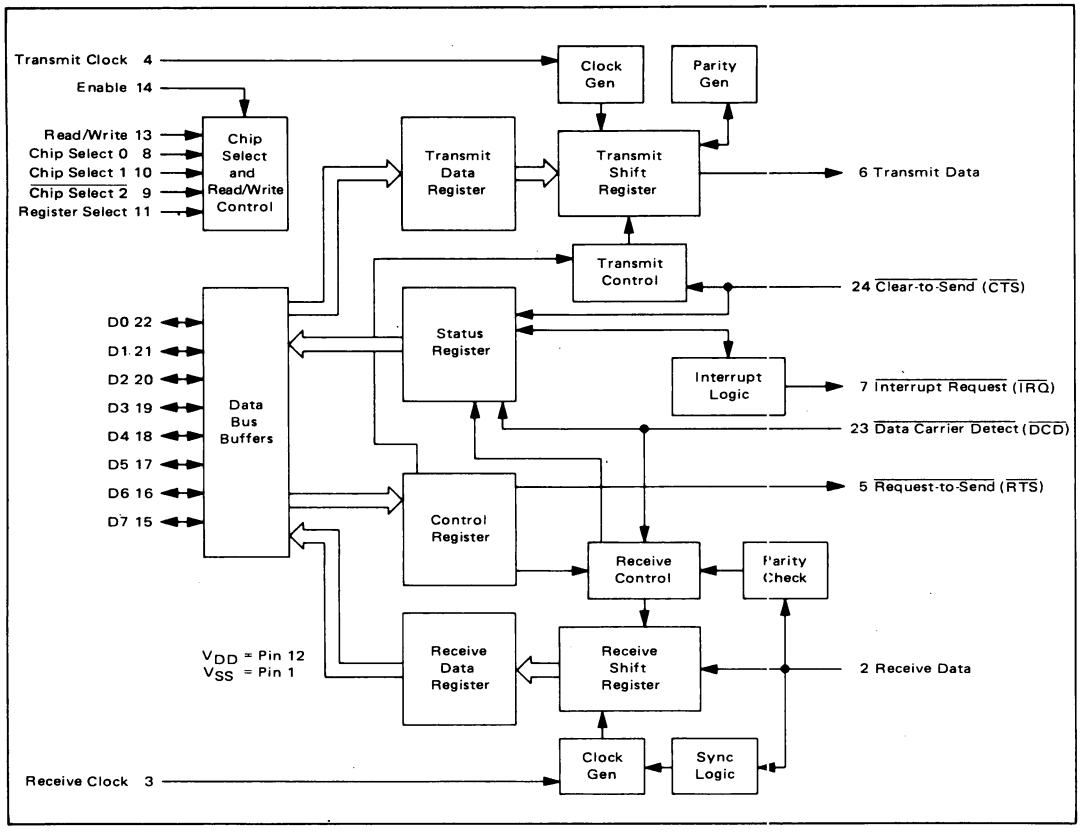


FIGURE 1 – ACIA Block Diagram

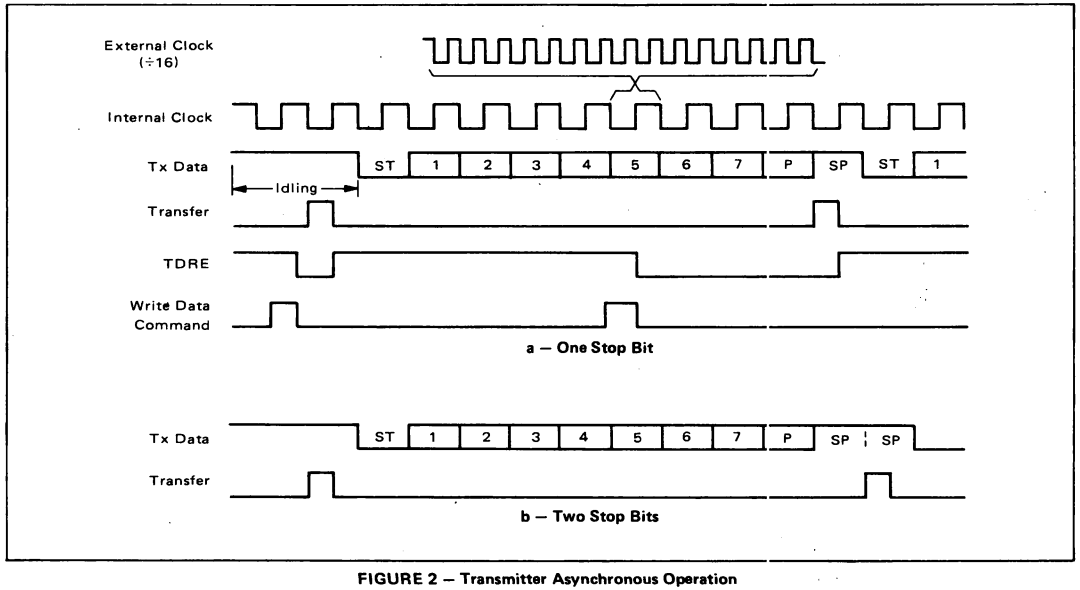


FIGURE 2 – Transmitter Asynchronous Operation

The start, data, and stop bits are shifted out of the transmit shift register on the negative transition of the external transmit clock which is coincident with the negative transition of the internal clock. Selection of the external clock frequency is based on the data transmission rate and clock division ratio of the ACIA. For example, a data transmission rate of 300 bits/s requires an external clock frequency of 300 Hz in the ÷1 mode and 4800 Hz in the ÷16 mode (16 times the data rate). There is no requirement on the duty cycle of the transmitter clock except with respect to the minimum clock pulse width specification listed on the data sheet.

After the first character has been loaded into the TDR, the status register can be read again to check for a Transmit Data Register empty condition and the current peripheral status. If the transmit data register is empty, another character can be written into the TDR even though the first character is still being shifted out of the shift register, due to double buffering being used within the ACIA. Referring to Figure 2, the second character is transferred to the transmit shift register during the last stop bit time of the first character resulting in a contigious transmission of characters (isochronous transmission). If the second character is not written into the TDR prior to the last stop bit time of the character being transmitted, the transmitter will return to an idling condition at the end of that character time.

During the transmission operation, word length and stop bit select may be changed any time except during the internal transfer time without affecting the character being transmitted. The even/odd parity select will immediately affect the character presently being transmitted. Also, changes in word length and parity select will effect the reception of data by the receiver.

Since the control register containing the above functions is common to both the transmitter and receiver sections, these functions for the transmitter must be changed when the receiver is not receiving data, i.e., idling. This control register consideration must also be adhered to for transmission between a local transmitter and a remote receiver.

Receiver

In many asynchronous data communications systems, the data is transmitted in a random manner without any additional synchronization signal. Therefore, the start and stop elements of the asynchronous characters are used to establish both bit and character synchronization. The receiver generates an internal clock that is synchronized to the data from an external clock source (Rx Clock). As with the transmitter portion, the selection of the external clock frequency is based on the received data transmission rate and clock division ratio of the ACIA. For example, a data transmission rate of 300 bits/s requires an external clock frequency of 4800 Hz (16 times the data rate) in the ÷16 mode, and 19,200 Hz (64 times the data rate) in the ÷64 mode. (The ÷1 mode requires external synchronization and is explained separately in a following paragraph.)

Bit synchronization in the ÷16 and ÷64 modes is initiated by the leading mark-to-space transition of the start bit. The start bit on the receiver data input is sampled during the positive transitions of the external clock as shown in Figure 3. If the input remains at a "low" level for a total of 9 separate samplings in the ÷16 mode or

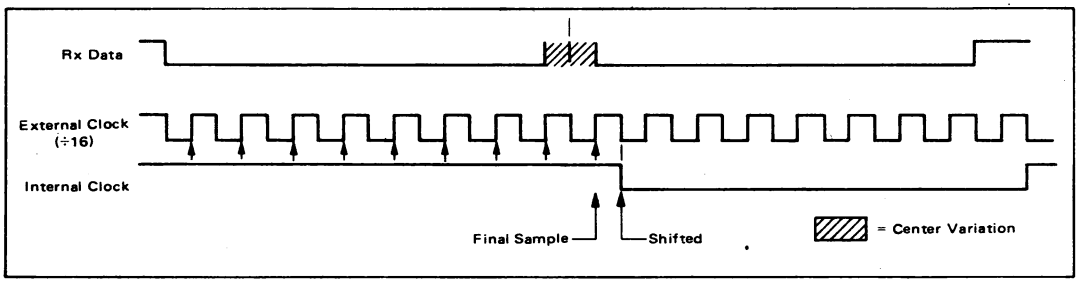


FIGURE 3 – Receiver Start Bit Detection (÷16 and ÷64 Modes)

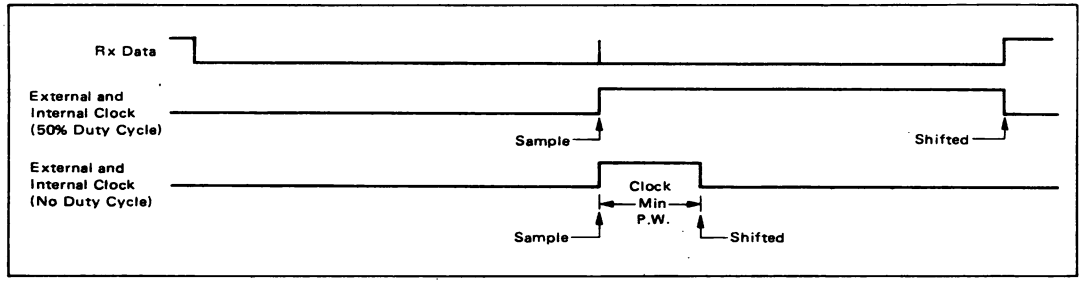


FIGURE 4 – Clock Requirement for ÷1 Mode

33 samplings in the ÷64 mode, which is equivalent to more than 50% of a bit time, the bit is assumed to be a valid start bit. This start bit is shifted into the shift register on the negative edge of the internal clock. Once a valid start bit has been detected, bit and character synchronization are obtained and the remaining bits are shifted into the shift register at their approximate midpoints.

If the receiver input returns to a mark state during the start bit sampling period, this false start bit is ignored and the receiver resumes looking for the mark-to-space transition of a valid start bit; this technique is referred to as false start bit deletion. The ACIA monitors the start bit on an incremental sampling basis rather than on a continuous sampling basis. This technique is a desirable feature for operation within a noisy environment and stems from the fact that a noise pulse occurring anywhere in a continuous sampling technique would initialize the monitoring logic; whereas in an incremental sampling technique, the noise pulse must occur during the sample to initialize the monitoring logic. The receiver will repeat this process for synchronization of each character in the message.

Divide-by-1 mode selection will not provide internal bit synchronization within the receiver. Therefore, the external receive clock must be synchronized to the data under the following considerations. The sampling of the start bit occurs on the positive edge of the external clock and the start bit is shifted into the shift register on the negative edge of the external clock, as shown in Figure 4. For higher reliability of sampling, the positive transition of the external clock (sampling point) should occur at the approximate midpoint of the bit interval. There is no requirement on the duty cycle of the external receive clock except that the clock must meet the minimum pulse width requirement as noted on the ACIA data sheet.

After the start bit has been detected, the remaining portion of the character being received is checked for parity, framing, and overrun errors. The complete reception of the character produces a "high" on the Receiver Data Register Full (RDRF) status bit, indicating that the receiver data register is full (Figure 5). The received character is transferred to the Receive Data Register (RDR) with the start, stop, and parity bits stripped from the character. At the same time, any receive data errors (parity, overrun, framing) are available in the status register in accordance with the status register definitions. The RDR is oriented such that the first data bit received is available on the D0 output. The receiver is double buffered so that one character may be read from the data register as another character is being received in the shift register. During the reception of data characters, the absence of the first stop bit of the character will not result in the receiver losing character synchronization but will indicate a framing error. The above receive process is repeated for each character in the total message.

POWER-ON RESET/MASTER RESET

The ACIA contains an internal power-on reset circuit to detect the power line turn-on transition and to hold the ACIA in a reset state until initialization is complete to prevent any erroneous output transitions from occurring. In addition to initializing the transmitter and receiver sections, the power-on reset circuit holds the CR5 and CR6 bits of the control register at a logic 0 and logic 1, respectively. When CR5 = 0 and CR6 = 1 as defined by the ACIA data sheet, the Request-to-Send ($\overline{\text{RTS}}$) output is held "high" and an interrupt from the transmitter is disabled. The power-on reset logic is sensitive to the shape

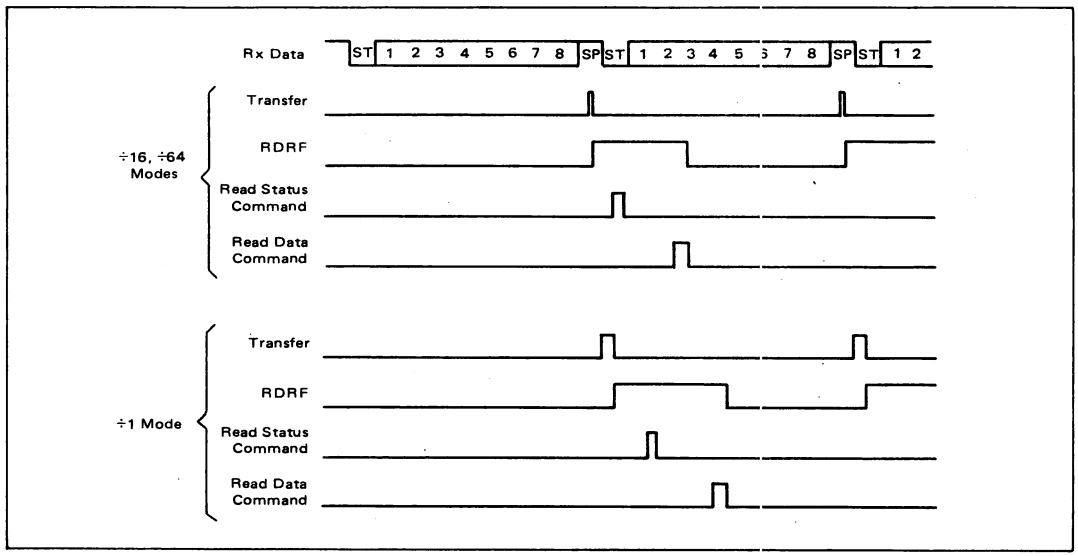


FIGURE 5 – Receiver Asynchronous Operation

TABLE 1 – Reset Functions

Status Register	POWER-ON RESET b7 b6 b5 b4 b3 b2 b1 b0 0 0 0 0 X X 0 0	MASTER RESET (Release Power-On Reset) b7 b6 b5 b4 b3 b2 b1 b0 0 0 0 0 X X 0 0	MASTER RESET (General) b7 b6 b5 b4 b3 b2 b1 b0 0 0 0 0 X X 0 0
$\overline{IRQ}$ Output	1	1	1
$\overline{RTS}$ Output	1	1	X
Transmit Break Capability	Inhibit	Inhibit	Optional
Internal: RIE	0	X	X
TIE	0	0	X

Held by Power-On Reset — Defined by Control Register

(X-Independent of Reset function)

of the V_{DD} power supply turn-on transition. To insure correct operation of the reset function, the power turn-on transition must have a positive slope throughout its transition. The conditions of the status register and other outputs during a power-on reset or software master reset are shown in Table I.

The internal ACIA power-on reset logic must be released prior to the transmission of data by performing a software master reset function via the control register. Control Register bits CR0 and CR1 are used to program a master reset condition while the remaining control register bits provide other functions in accordance with the ACIA data sheet. The internal power-on reset logic will inhibit any change in bits CR5 and CR6 of the control register. Therefore, the control word that generates the master reset function clearing the internal power-on reset will not change the $\overline{RTS}$ output or the Internal Transmit Interrupt Enable (TIE), as reflected in Table I. Also, the state of the Receiver Interrupt Enable (RIE) bit of the control register has no external effect because the receiver is initialized by the master reset function.

After master reset of the ACIA, the programmable control register can be set for a number of options such as variable clock divider ratios, variable word length, one or two stop bits, parity (even, odd, or none). Also, bits CR5 and CR6 of the control register are no longer inhibited and can now be programmed for several options as defined on the ACIA data sheet.

During the initialization of the ACIA, the master reset function can be optionally used to establish a communications link without generating an interrupt from the transmitter or receiver sections. For example, the first control word, XXXXXX11-LSB (X = don't care) resets the power-on reset logic. To maintain a reset condition, the second control word, X01XXX11-LSB holds the transmitter and receiver in a reset state and produces a "low" on the $\overline{RTS}$ output. The $\overline{RTS}$ output may be used to enable a local modem. The local modem, upon detection of a remote modem's carrier, will generate a "low" on the Clear-to-Send ($\overline{CTS}$) input of the ACIA. Since the $\overline{CTS}$ bit of the status register reflects the present status of the $\overline{CTS}$ input, the establishment of the communications link can be verified by reading the status register of the ACIA. For a more detailed description of this system application, refer to the system implementation section.

STATUS REGISTER

ACIA status information is available to the MPU through the bus interface by means of the ACIA Status Register. Status information comes from three sources: the receiving section, the transmitting section, and the peripheral status inputs.

Receiver Status

Receive Data Register Full (RDRF), Bit 0 – A logic "high" level on the RDRF bit indicates that data has been transferred to the Receive Data Register and that the received data can be read from the ACIA. Reading the Receive Data Register causes the RDRF status bit to go "low", as shown in Figure 5. A "low" on the Data Carrier Detect ($\overline{DCD}$) input enables the RDRF status bit to be generated from a Receive Data Register full condition. A "high" on the $\overline{DCD}$ input or a master reset condition will force the RDRF status bit to a "low" state until the $\overline{DCD}$ input returns to a "low" state. This is independent of the state of the status register $\overline{DCD}$ bit.

Transmitter Status

Transmit Data Register Empty (TDRE), Bit 1 – The write data command (see Figure 2) causes the TDRE status bit to go "low", indicating a data register full condition. An internal transfer signal transfers the data from the Transmit Data Register to the Transmit Shift Register and causes the TDRE bit to go "high", indicating a Transmit Data Register Empty condition as shown in Figure 2. The TDRE bit contains the present status of the Transmit Data Register when the Clear-to-Send ($\overline{CTS}$) input is in a "low" state.

Peripheral Status

Data Carrier Detect ($\overline{DCD}$), Bit 2 – A "high" level on the $\overline{DCD}$ input, indicating a loss of carrier causes: (1) the $\overline{DCD}$ status bit to go "high"; (2) the RDRF bit to be inhibited ("low"); and (3) immediate initialization of the receiver. When the Receive Interrupt Enable (RIE) is set, a loss of carrier will cause: (1) an interrupt to occur ($\overline{IRQ}$ output goes "low"), and (2) the $\overline{IRQ}$ status register bit to go "high". The characteristics of the $\overline{DCD}$ status bit and the associated $\overline{IRQ}$ status bit are as follows, with reference to the six segments in Figure 6, where each

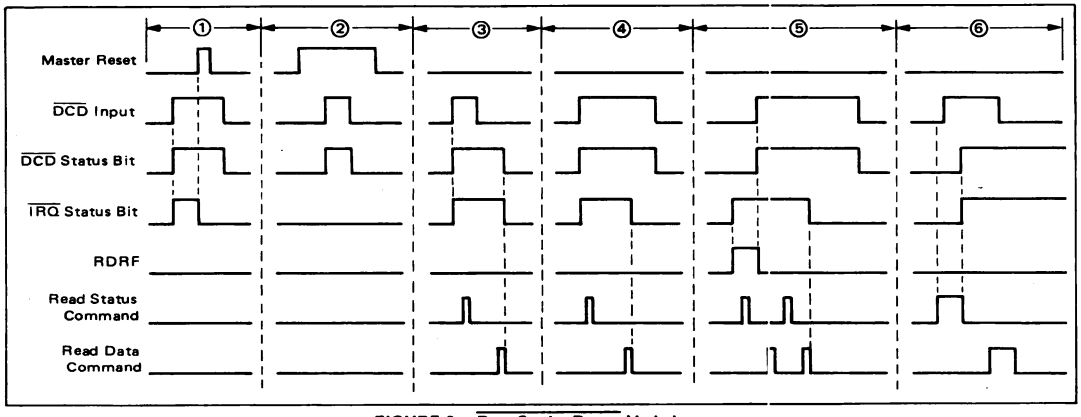


FIGURE 6 – Data Carrier Detect Variations

segment represents a specific condition. (Note: The IRQ output is the inverse logic level of the IRQ status bit.) Segment (1) – A master reset of the ACIA resets the interrupt status bit (IRQ) generated by a loss of carrier. Segment (2) – If the DCD input goes "high" during a master reset condition, the DCD status bit will reflect the state of the DCD input. Segment (3) – After an interrupt has occurred from a loss of carrier, the IRQ and DCD status bits (provided the DCD input has returned to a "low" level) are reset by first reading the Status Register and then reading the Data Register. Segment (4) – If the DCD input remains "high" after a read status and a read data, the IRQ bit will be cleared but the DCD status bit remains "high" and will follow the state of the DCD input. Segment (5) – If a read status occurs when the DCD input is "low" followed by a loss of carrier (DCD input goes "high") prior to the read data command, this read data command will not reset either the IRQ or DCD status bits. The next read status followed by a read data will reset the IRQ status bit. Segment (6) – A transition of the DCD input during a read status or read data command is not recognized until the trailing edge of the read command. The DCD input to the ACIA must be tied "low" if it is not used.

Clear-to-Send (CTS), Bit 3 – The CTS status bit continuously reflects the state of the CTS input. A "high" on the CTS input will inhibit the TDRE status bit and associated interrupt status bit (IRQ). The CTS input has no effect on a character being transmitted from the shift register or the character in the Transmit Data Register (the transmitter is not initialized). Also, the CTS bit is not affected by a master reset. The CTS input to the ACIA must be tied "low" if it is not used.

Framing Error (FE), Bit 4 – A framing error indicates the absence of the first stop bit of a character resulting from a loss of character synchronization, faulty transmission, or a "break" (all spaces) condition. If one of the above conditions is present, the internal receiver transfer signal will cause the FE bit to go "high". The next internal transfer signal will cause the FE status bit to be updated for the error status of the next character, as shown in Figure 7. A "high" on the DCD input or a master reset

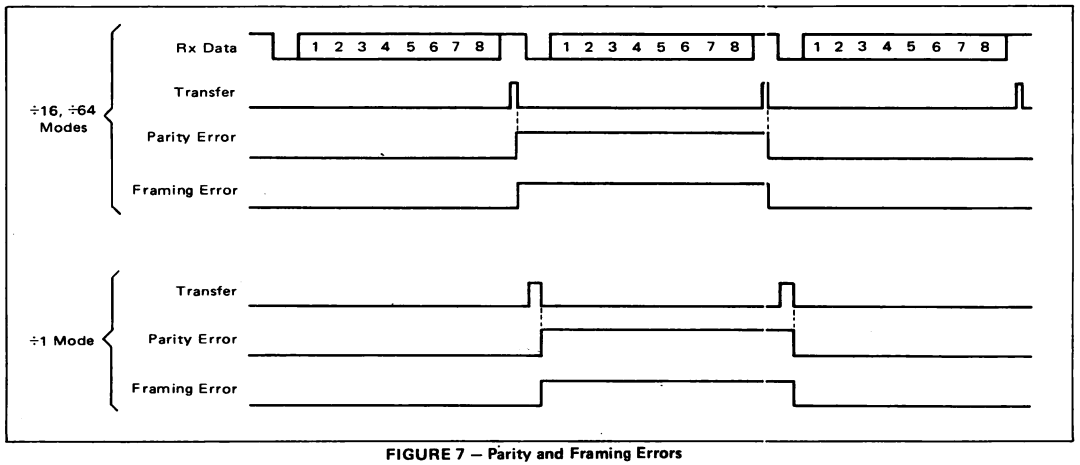


FIGURE 7 – Parity and Framing Errors

will disable and reset the FE status bit.

Overrun Error (OVRN), Bit 5 – A "high" state on the OVRN status bit indicates that a number of characters were received but not read from the Receive Data Register, resulting in the loss of a character/or characters. The OVRN status bit is set when the last character prior to the overrun condition has been read. The read data command forces the RDRF and OVRN status bits to go "high" if an overrun condition exists. The next read data command causes the RDRF and OVRN status bits to return to a "low" level. During an overrun condition, the last character in the Receive Data Register that was not read subsequent to the overrun condition is retained since the internal transfer signal is disabled. Figure 8 illustrates the timing

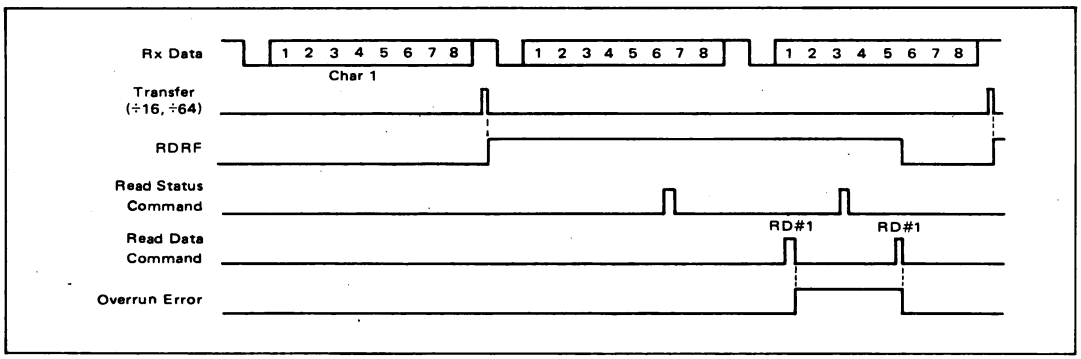


FIGURE 8 – Overrun Error

events during an overrun error condition. A "high" state on the $\overline{DCD}$ input or a master reset disables and resets the OVRN status bit.

Parity Error (PE), Bit 6 – If the parity check function is enabled, the internal transfer signal causes the PE status bit to go "high" if a parity error condition exists. The parity error status bit is updated by the next internal transfer signal, as shown in Figure 7. A "high" state on the $\overline{DCD}$ input or a master reset disables and resets the PE status bit.

Interrupt Request ($\overline{IRQ}$), Bit 7 – A "high" level on the $\overline{IRQ}$ status bit may be generated from three sources: transmitter, receiver, and loss of carrier. (a) Transmitter – if the transmitter interrupt enable (TIE) is active, the state of the TDRE status bit is reflected by the $\overline{IRQ}$ status bit (refer to TDRE, Bit 1); (b) Receiver – if the internal receiver interrupt enable (RIE) is active, the state of the RDRF status bit is reflected by the $\overline{IRQ}$ status bit (refer to RDRF, Bit 0); (c) Data Carrier Loss – a loss of carrier (logic "high" level) on the $\overline{DCD}$ input generates an interrupt on the $\overline{IRQ}$ status bit if RIE is active (refer to $\overline{DCD}$, Bit 2).

The above status information is accumulated in a random asynchronous manner. Because of the asynchronous nature for updating status, it is possible that the status word will change before, during, or after the reading of the status register. This presents no problem during a status read cycle because no automatic status reset will occur. The response of the system to a status word will depend upon the status bit read. Should a status change not be registered, it can be read during the next read status cycle.

SYSTEM IMPLEMENTATION

In a microcomputer based system, an address map of the system identifies the block of memory allocated for the system program, stack storage location, interrupt locations, peripheral addresses, etc. The ACIA requires two addresses in the MPU system for addressing its four registers: control, status, transmit, and recieve. To select a register within the ACIA requires the appropriate logic levels on the chip select inputs (CS0, CS1, $\overline{CS2}$), register select input (RS), and read/write control input (R/W). The R/W output line provided by the MPU (MC6800) is used to control writing to and reading from peripheral interface devices or memory. In addition, the R/W control selects one of the read or write registers in the ACIA. A combination of the chip selects and register select inputs can be used to minimize the amount of address decoding logic required for each peripheral. For example, the four Boolean combinations of address lines A14 and A15 select blocks of memory locations as shown in Figure 9. Assigning these blocks specific functions such as RAM, ROM, and peripheral devices forms a memory map of the system. In this example, the peripheral devices are assigned to addresses between 8000 and BFFF (hexadecimal notation). Assigning address bit A15 to CS0 and address bit A14 to $\overline{CS2}$ selects a peripheral device when A15 = "1" and A14 = "0". Since the ACIA requires two addresses, the use of address bit A0 for the RS input assigns two consecutive addresses for selection of the ACIA's four internal registers. Connecting the CS1 input to one of the other address lines allows the selection of 13 different peripherals without any additional decoding logic.

The peripheral side of the ACIA provides a means by which a microcomputer can efficiently control a peripheral device requiring an asynchronous serial data format. This format is generally used in (but not confined to) low and medium transmission rate systems – 1800 bps and below.

Peripheral	8000 – BFFF	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X
RAM	4000 – 7FFF	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X
RAM	0000 – 3FFF	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X
ACIA #1	8400 – 8401	CS0	$\overline{CS2}$	0	0	0	CS1	0	0	0	0	0	0	0	0	0	RS
ACIA #2	8020 – 8021	CS0	$\overline{CS2}$	0	0	0	0	0	0	0	0	CS1	0	0	0	0	RS

X = 1 or 0
CS0 = 1
$\overline{CS2}$ = 0
CS1 = 1
RS = 1 or 0

FIGURE 9 – Address Specification

A teletype is an example system which has a transmission rate of 110 bps or 10 char/s. An interface device is required between a teletype and an ACIA to convert the TTL compatible levels of the ACIA to the 20 mA current loop of the teletype. A non-complemented data teletype interface requires an optical coupler (4N33) with the addition of logic inverters, as shown in Figure 10. Other teletype options such as RS-232 can be easily interfaced to the ACIA with RS-232 interface devices (MC1488, MC1489), as shown in Figure 11.

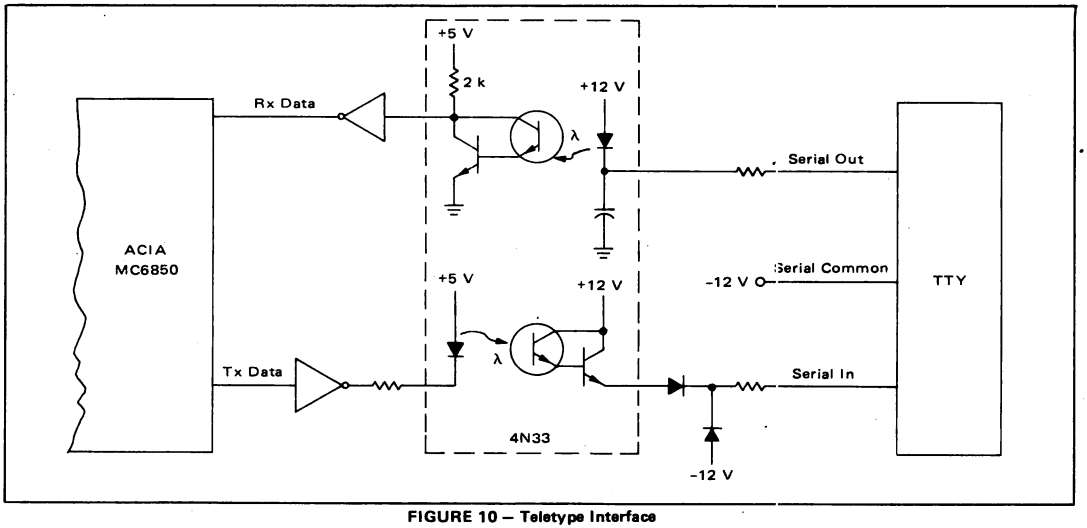


FIGURE 10 – Teletype Interface

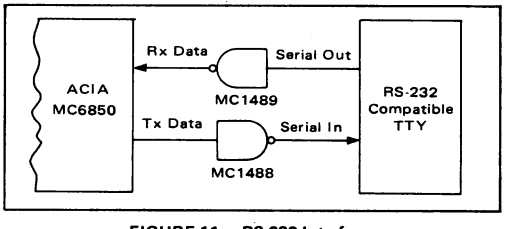


FIGURE 11 – RS-232 Interface

The ACIA provides a means to control a modem for the transmission of data to and from a remote terminal, such as a teletype, over the telephone lines. The modem function can be implemented with a low speed modem, MC6860, as shown in Figure 12. The MC6860 modem uses a frequency shift keying (FSK) technique for the transmission of data up to a maximum data rate of 600 bps. A typical system consists of a local modem and a similar modem at the remote terminal. The local modem converts the digital data from the ACIA to analog form for transmission over the telephone lines. Likewise, analog data received from the remote modem is converted back to digital form by the local modem for use by the microcomputer system via the ACIA. Refer to application notes, AN-731, AN-747 and the MC6860 data sheet for a detailed description of the modem and its operation.

Since the MC6860 does not provide automatic dialing, the telephone channel must be established by manual means or through the use of external automatic dialing equipment. The procedure for accomplishing the handshaking

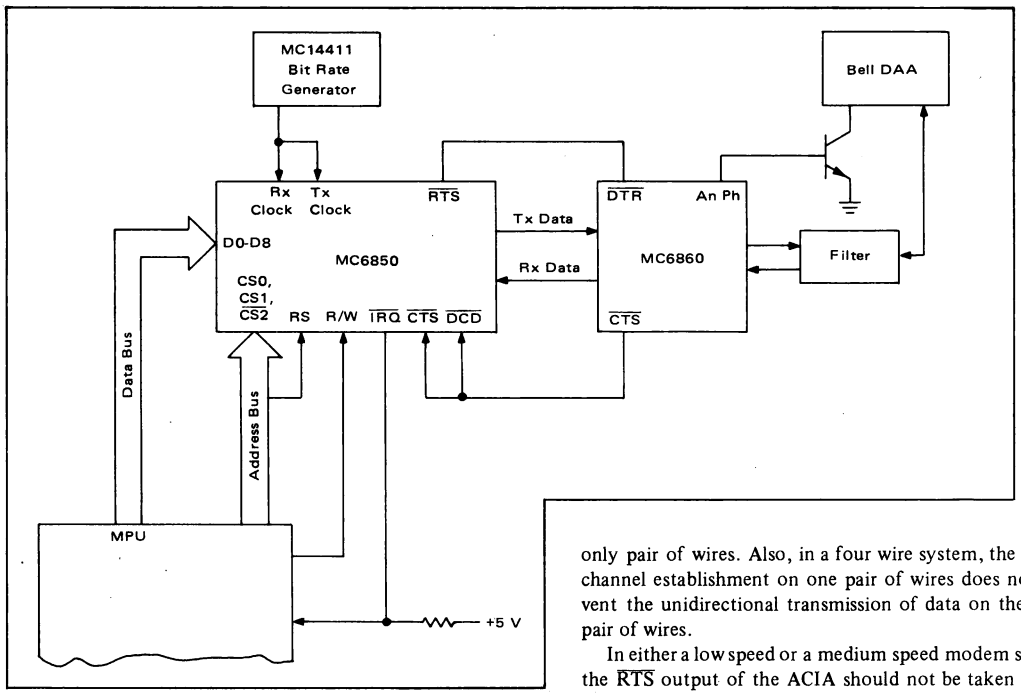


FIGURE 12 – ACIA to Modem Interface

between the local modem and remote modem after the telephone channel has been established is as follows. Under program control, the local modem is enabled via the Request-to-Send (RTS) output of the ACIA which is connected to the Data Terminal Ready (DTR) input of the modem. Since the remote modem has answered the phone due to a ring detection, it has transmitted back a handshaking carrier frequency to the local modem. Upon the detection of the remote modem carrier frequency, the local modem enables its Clear-to-Send (CTS) output. The CTS output of the modem is tied directly to the CTS input of the ACIA and the state of this input is available as a status bit. Therefore, under program control, the completion of the handshaking between the local and remote modem can be verified by reading the status register. After modem handshaking is completed, data can be transmitted and received over the telephone lines under program control of the microcomputer system. Since a low speed modem such as the MC6860 provides only a CTS output, the CTS and DCD inputs of the ACIA in this example were tied together such that a communications link disconnect could be detectable in either the transmitting or receiving subroutines. The software section of this note provides additional information on the handling of the CTS and DCD status bits.

Medium speed modem systems may independently utilize both the CTS and DCD inputs. In a four wire system, the CTS input indicates the status of a transmit-only pair of wires and the DCD input indicates the status of a receive-only pair of wires. Also, in a four wire system, the loss of channel establishment on one pair of wires does not prevent the unidirectional transmission of data on the other pair of wires.

In either a low speed or a medium speed modem system, the RTS output of the ACIA should not be taken "high" until the last character is completely received by the remote system. However, the ACIA does not provide a word complete output indicating that the last character loaded has been completely transmitted out so that the modem may be disabled. The word complete function can be generated by loading a "dummy" character into the ACIA and then reading the status register for a transmit data register empty condition indicating that the "dummy" character has been transferred to the shift register. This provides an indication that half the stop bit of the last data character has been completely transmitted. Taking the RTS "high" does provide a means for disabling the local modem, but care should be taken to ensure that the last character has been read by the remote system prior to disabling the modems.

As the microcomputer system is expanded with more peripheral devices requiring more processing time, it becomes increasingly difficult to service each peripheral in the time available. One method to increase the efficiency of the system is to use an interrupt driven system. In an interrupt driven system, each interface adapter of the MPU family has an interrupt output (IRQ) that is wire-ORed to the other interface adapters to generate a common interrupt to the MPU. An interrupt from any of the interface adapters causes the MPU to jump to an interrupt address after it has finished processing its present instruction. The contents of the interrupt address contains the address of the subroutine to service the interrupt. The MPU then executes the interrupt routine, which samples the status register of each interface adapter. The ACIA provides an IRQ status bit that is located in the D7 position of the

status register (sign position for numbers) such that only one MPU instruction, BMI (Branch if minus), is required to determine if the transmitter or receiver portion of that particular ACIA was generating the interrupt. Once it is determined that an ACIA is generating the interrupt, the TDRE and RDRF status bits can be checked within their individual subroutines to determine the specific reason for interrupt. The control register can be programmed to inhibit an interrupt from either the transmitter or receiver portions depending on the intended use of the ACIA.

An MC14411 CMOS Bit Rate Generator (BRG), which has 16 standard communication clock rates available, provides a clock source for the ACIA. The receiver and transmitter sections of the ACIA have separate clock inputs to provide independent transmission rates, if desired.

SOFTWARE

Since the internal registers of the ACIA and other MPU interface adapters look like memory locations to the MPU, there is no need for special instructions in the MPU instruction set when using interface adapters. The MPU instructions most commonly used for writing information into the ACIA and reading information out of the ACIA are the store (STA) and load (LDA) instructions, respectively. A store instruction causes the read/write (R/W) output of the MPU to go "low" while a load instruction causes the R/W output to go "high". Assigning consecutive addresses with address bus bit A0 tied to the ACIA Register Select input (RS) along with the R/W input allows access of one of the four ACIA internal registers in accordance with Table II. For example, an STA instruction

ADDRESS* LOCATION (Hexadecimal Notation)	STA INSTRUCTION (R/W = 0)	LDA INSTRUCTION (R/W = 1)
8400 RS = A0 = 0	Control Register	Status Register
8401 RS = A0 = 1	Transmit Register	Receive Register

*A0 tied to RS

TABLE 2 – ACIA Register Selection

to address 8400 (hexadecimal notation) performs a write to the ACIA control register whereas an LDA instruction to the same address performs a read of the ACIA status register.

Software Initialization Routine

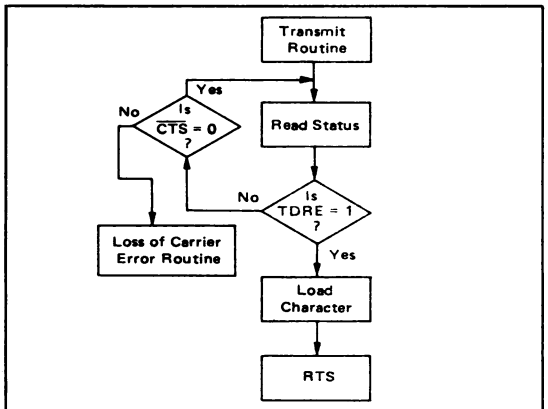
The ACIA must be initialized prior to transmitting and receiving data. During a power-on transition, an internal power-on chip reset (latch) holds the $\overline{IRQ}$ output "high" to prevent the ACIA from interrupting the MPU or transmitting erroneous information (Ref. Table I). The power-on reset function is released by master resetting the ACIA. A master reset is accomplished by storing a word with bits B0 and B1 equal to "one" into the Control Register. After master resetting, the control register is programmed to set the counter divide ratio, word length, parity, interrupt control, etc., which completes the initialization of the ACIA.

Transmit and Receive Software Routines

After completion of initialization, the ACIA can then be used for transmitting and receiving data. Due to the length of data messages, the transmission of data is normally handled in subroutines to reduce the duplication of instructions. Typical examples for transmit and receive subroutines, flow diagrams, and source statements are shown in Figures 13 and 14, respectively.

Referring to the transmit subroutine, the contents of the ACIA status register are loaded into the accumulator of the MPU. Under program control, a character is stored into the ACIA for transmission if the transmitter data register is empty. Control is then returned from the subroutine back to the main program by an RTS instruction. If the transmitter data register is not empty (TDRE = 0) indicating the transmit data register is full or that the $\overline{CTS}$ input is "high", inhibiting the TDRE status, the $\overline{CTS}$ status information which was previously loaded into the accumulator should be checked for its condition. This step is not required when the $\overline{CTS}$ input is permanently held "low". In a system using a modem, a "high" on the $\overline{CTS}$ input indicates that the modem data carrier is not present or was lost, requiring the re-establishment of the communications channel. A "low" on the $\overline{CTS}$ status register bit indicates the TDR is full and allows the status register to be re-read in a loop manner until the TDR becomes empty. When a TDR empty indication occurs, the character stored in the TDR and control is returned to the main program.

Referring to the receive subroutine, there is a similarity of its flow diagram to the transmit routine. The first step in the receive routine is to load the contents of the status register into the accumulator of the MPU. If the receive data register is not full (RDRF = 0), it indicates that the register is empty or that the receiver is inhibited by the $\overline{DCD}$ input being "high". Therefore, the $\overline{DCD}$ status bit which was previously loaded into the accumulator should be checked under program control for its condition (this step is not required when the $\overline{DCD}$ input is permanently held "low"). In a medium speed modem system, a "high" on the $\overline{DCD}$ input during character reception indicates that the receive carrier was lost and the communications channel would have to be re-established. The $\overline{DCD}$ status bit is reset back to a "low" state when: (1) the $\overline{DCD}$ input has returned "low"; (2) by a master reset; or (3) by reading the Receive Data Register after having read the status register. If the $\overline{DCD}$ status bit is "low", the status is re-read in a loop manner until the receive data register is full. When a logic "1" is read from the RDRF status bit position (B0), indicating that a character was received, the status regarding the framing, overrun, and parity errors of the received character is available. A received character status error could provide for re-transmission of the message, or implement error correction techniques. If there are no errors in the character received, the Receive Data Register is read and control is returned from the



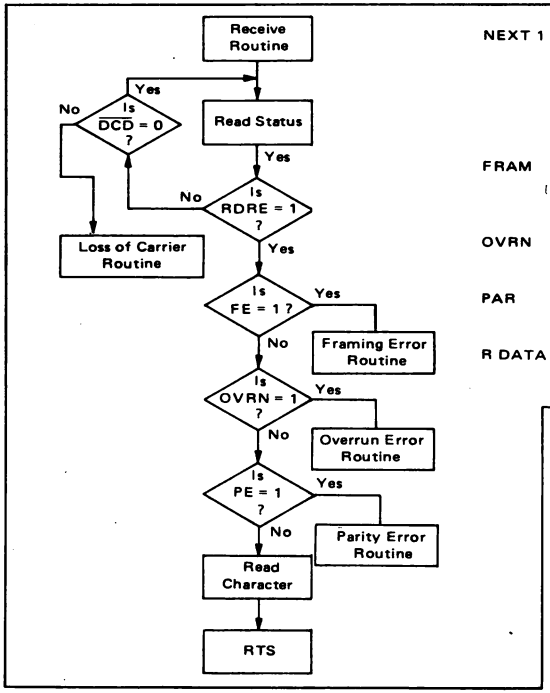

```
NEXT    LDA A STACON    Load Status
        ASR A
        ASR A           Shift TDRE Bit to C-Bit Position
        BCS TX Data     Check TDRE Bit
        ASR A
        ASR A           Shift CTS Bit to C-Bit Position
        BCC NEXT        Check CTS
        BR ERROR 1      Carrier Loss-Branch to Error Routine
TX Data STA B TXRX      Store Character in ACIA
        RTS             Return from Subroutine
```

FIGURE 13 – Flow Diagram and Source Statements for Transmit Subroutine

subroutine to the main program via an RTS instruction.

In an interrupt driven system, the ACIA can be programmed to generate an interrupt from the transmitter or receiver sections independently. For example, an interrupt from only the transmitter section can be achieved by enabling the transmitter interrupt enable (CR5 = 1, CR6 = 0), and disabling the receiver interrupt enable (CR7 = 0). This results in an interrupt being generated when the Transmit Data Register is empty (TDRE = 1). Therefore, the condition of the Transmit Data Register Empty bit is known and there is no need to examine the condition of this bit as shown in the transmit data subroutine in Figure 13.

As demonstrated in the program examples, only the STA and LDA instructions are required to access one of the four internal registers within the ACIA. However, there are other instructions such as the CMP (compare) instruction that can be used with the ACIA. Since the registers in the ACIA are either write-only or read-only registers, the MPU instructions that perform an automatic rewrite should not be used with the ACIA; this would result in the selection of two registers from one instruction. The MPU instructions that should not be used during ACIA operation are: ASL, ASR, COM, DEC, INC, LSR, NEG, ORA, ROR, and ROL.




```
NEXT 1  LDA A STACON    Load Status
        ASR A           Shift RDRF Bit to C-Bit Position
        BCS FRAM        Check RDRF Bit
        ASR A
        ASR A           Shift DCD Bit to C-Bit Position
        BCC NEXT 1      Check DCD Bit
        BR ERROR 2      Carrier Loss – Branch to Error Routine
FRAM    ASR A
        ASR A           Shift FE Bit to C-Bit Position
        BCC OVRN        Check FE Bit
        BR ERROR 3      Framing Error – Branch to Error Routine
OVRN    ASR A           Shift OVRN Bit to C-Bit Position
        BCC PAR         Check OVRN Bit
        BR ERROR 4      Overrun Error-Branch to Error Routine
PAR     ASR A           Shift PE Bit to C-Bit Position
        BCC R Data      Check PE Bit
        BR ERROR 5      Parity Error-Branch to Error Routine
R DATA  LDA B TXRX      Load B Register with Data
        RTS             Return From Subroutine
```

FIGURE 14 – Flow Diagram and Source Statements for Receive Subroutine

CONCLUSION

The ACIA provides a cost effective approach for adding asynchronous data communications links to computer, minicomputer, and microcomputer systems. The memory-like registers of the ACIA enable a processor to transmit and receive data via the ACIA without the need of special I/O instructions. The ACIA also provides modem I/O control for the transmission of data to remote sites over the telephone network. Included has been a detailed discussion on the ACIA status register along with software program examples, such that the ACIA user can effectively and efficiently apply this part in his data communications system.