AP-283

APPLICATION NOTE

Flexibility in Frame Size with the 8044

PARVIZ KHODADADI
APPLICATIONS ENGINEER

November 1990

Order Number: 292019-001

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained from:

Intel Corporation
P.O. Box 7641
Mt. Prospect, IL 60056-7641

or call 1-800-879-4683

FLEXIBILITY IN FRAME SIZE WITH THE 8044

CONTENTS

PAGE

CONTENTS

CONTENTS

1.0 INTRODUCTION

The 8044 is a serial communication microcontroller known as the RUPI (Remote Universal Peripheral Interface). It merges the popular 8051 8-bit microcontroller with an intelligent, high performance HDLC/SDLC serial communication controller called the Serial Interface Unit (SIU). The chip provides all features of the microcontroller and supports the Synchronous Data Link Control (SDLC) communications protocol.

There are two methods of operation relating to frame size:

1) Normal operation (limited frame size)
2) Expanded operation (unlimited frame size)

In Normal operation the internal 192 byte RAM is used as the receive and transmit buffer. In this operation, the chip supports data rates up to 2.4 Mbps externally clocked and 375 Kbps self-clocked. For frame sizes greater than 192 bytes, Expanded operation is required. In Expanded operation the external RAM, in conjunction with the internal RAM, is used as the transmit and receive buffer. In this operation, the chip supports data rates up to 500 Kbps externally clocked and 375 Kbps self-clocked. In both cases, the SIU handles many of the data link functions in hardware, and the chip can be configured in either Auto or Flexible mode.

The discussion that follows describes the operation of the chip and the behavior of the serial interface unit. Both Normal and Expanded operations will be further explained with extra emphasis on Expanded operation and its supporting software. Two examples of SDLC communication systems will also be covered, where the chip is used in Expanded operation. The discussion assumes that the reader is familiar with the 8044 data sheet and the SDLC communications protocol.

1.1 Normal Operation

In Normal operation the on-chip CPU and the SIU operate in parallel. The SIU handles the serial communication task while the CPU processes the contents of the on-chip transmit and receiver buffer, services interrupt routines, or performs the local real time processing tasks.

The 192 bytes of on-chip RAM serves as the interface buffer between the CPU and the SIU, used by both as a receive and transmit buffer. Some of the internal RAM space is used as general purpose registers (e.g. R0–R7). The remaining bytes may be divided into at least two sections: one section for the transmit buffer and the other section for the receive buffer. In some applications, the 192 byte internal RAM size imposes a limitation on the size of the information field of each frame and, consequently, achieves less than optimal information throughput.

Figure 1 illustrates the flow of data when internal RAM is used as the receive and transmit buffer. The on-chip CPU allocates a receive buffer in the internal RAM and enables the SIU. A receiving SDLC frame is processed by the SIU and the information bytes of the frame, if any, are stored in the internal RAM. Then, the SIU informs the CPU of the received bytes (Serial Channel interrupt). For transmission, the CPU loads the transmitting bytes into the internal RAM and enables the SIU. The SIU transmits the information bytes in SDLC format.

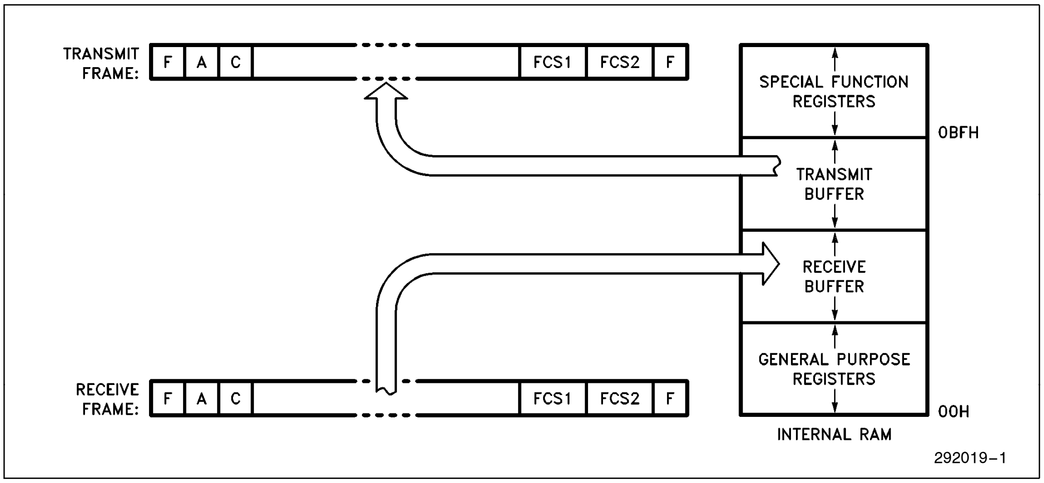


Figure 1. Transmission/Reception Data Flow Using Internal RAM

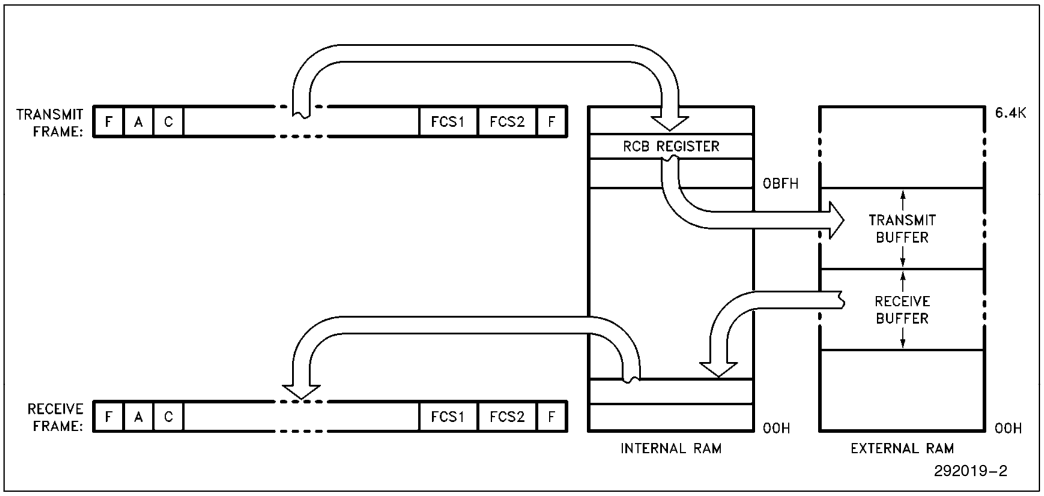


Figure 2. Transmission/Reception Data Flow Using External RAM

1.2 Expanded Operation

In Expanded operation the on-chip CPU monitors the state of the SIU, and moves data from/to external buffer to/from the internal RAM and registers while reception/transmission is taking place. If the CPU must service an interrupt during transmission or reception of a frame or transmit from internal RAM, the chip can shift to Normal operation.

There is a special function register called SIUST, the contents of which dictate the operation of the SIU. Also, at data rates lower than 2.4 Mbps, one section of the SIU, in fixed intervals during transmission and reception, is in the "standby" mode and performs no function. The above two characteristics make it possible to program the CPU to move data to/from external RAM and to force the SIU to repeat or skip some desired hardware tasks while transmission or reception is taking place. With these modifications, external RAM can be utilized as a transmit and receive buffer instead of the internal RAM.

Figure 2 graphically shows the flow of data when external RAM is used. For reception, the receiving bytes are loaded into the Receive Control Byte (RCB) register. Then, the data in RCB is moved to external RAM and the SIU is forced to load the next byte into the RCB register - The chip believes it is receiving a control byte continuously. For transmission, Information bytes (I-bytes) are loaded into a location in the internal RAM and the chip is forced to transmit the contents of this location repeatedly.

Discussion of expanded operation is continued in sections 4 and 5. First, however, sections 2 and 3 describe features of the 8044 which are necessary to further explain expanded operation.

2.0 THE SERIAL INTERFACE UNIT

2.1 Hardware Description

The Serial Interface Unit (SIU) of the RUPI, shown in Figure 3, is divided functionally into a Bit Processor (BIP) and a Byte Processor (BYP), each sharing some common timing and control logic. The bit processor is the interface between the SIU bus and the serial port pins. It performs all functions necessary to transmit/receive a byte of data to/from the serial data line (shifting, NRZI coding, zero insertion/deletion, etc.). The byte processor manipulates bytes of data to perform message formatting, transmitting, and receiving functions. For example, moving bytes from/to the special function registers to/from the bit processor.

The byte processor is controlled by a Finite-State Machine (FSM). For every receiving/transmitting byte, the byte processor executes one state. It then jumps to the next state or repeats the same state. These states will be explained in section 3. The status of the FSM is kept in an 8-bit register called SIUST (SIU State Counter). This register is used to manipulate the behavior of the byte processor.

As the name implies, the bit processor processes data one bit at a time. The speed of the bit processor is a function of the serial channel data rate. When one byte of data is processed by the bit processor, a byte boundary is reached. Each time a byte boundary is detected in the serial data stream, a burst of clock cycles (16 CPU states) is generated for the byte processor to execute one state of the state machine. When all the procedures in the state are executed, a wait signal is asserted to terminate the burst, and the byte processor waits for the next byte boundary (standby mode). The lower the data rate, the longer the byte processor will stay in the standby mode.

2.2 Reception of Frames

Incoming data is NRZI decoded by the on-chip decoder. It is then passed through the zero insertion/deletion (ZID) circuitry. The ZID not only performs zero insertion/deletion, but also detects flags and Go Aheads (GA) in the data stream. The data bits are then loaded into the shift register (SR) which performs serial to parallel conversion. When 8 bits of data are collected in the shift register, the bit processor triggers the byte processor to process the byte, and it proceeds to load the next block of data into the shift register. The serial data is also shifted, through SR, to a 16-bit register called "FCS GEN/CHK" for CRC checking. The byte processor takes the received address and control bytes from the SR shift register and moves them to the appropriate registers. If the contents of the shift register is expected to be an information byte, the byte processor moves them through a 3-byte FIFO to the internal RAM at a starting location addressed by the contents of the Receive Buffer Start (RBS) register.

2.3 Transmission of Frames

In the transmit mode, the byte processor relinquishes a byte to the bit processor by moving it to a register called RB (RAM buffer). The bit processor converts the data to serial form through the shift register, performs zero bit insertion, NRZI encoding, and sends the data to the serial port for transmission. Finally, the contents of the FCS GEN/CHK and the closing flag are routed to the serial port for transmission.

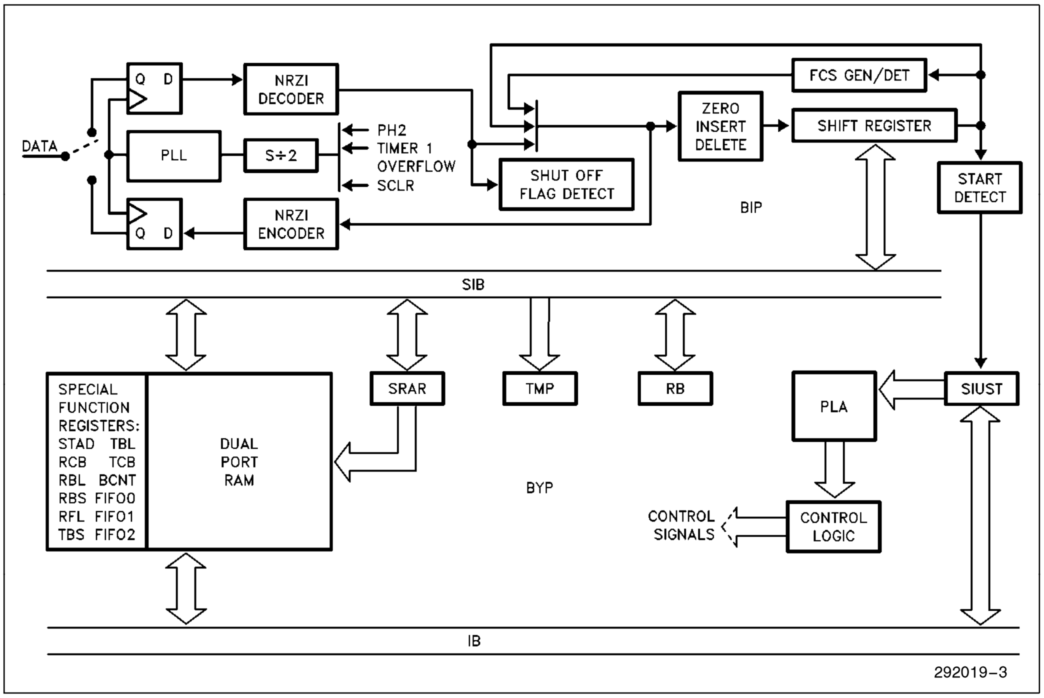


Figure 3. SIU Block Diagram

3.0 TRANSMIT AND RECEIVE STATES

The simplified receive and transmit state diagrams are shown in Figures 4 and 5, respectively. The numbers on the left of each state represent the contents of the SIUST register when the byte processor is in the standby mode, and the instructions on the right of each state represent the "state procedures" of that state. When the byte processor executes these procedures the least three significant bits of the SIUST register are being incremented while the other bits remain unchanged. The byte processor will jump from one state to another without going into the standby mode when a conditional jump procedure executed by the byte processor is true.

3.1 Receive State Sequence

When an opening flag (7EH) is detected by the bit processor, the byte processor is triggered to execute the procedures of the FLAG state. In the FLAG state, the byte processor loads the contents of the RBS register into the Special RAM (SRAR) register. SRAR is the pointer to the internal RAM. The byte processor decrements the contents of the Receive Buffer Length (RBL) register and loads them into the DMA Count (DCNT) register. The FCS GEN/CHK circuit is turned on to monitor the serial data stream for Frame Check Sequence functions as per SDLC specifications.

Assuming there is an address field in the frame, contents of the SIUST register will then be changed to 08H, causing the byte processor to jump to the ADDRESS state and wait (standby) for the next byte boundary. As soon as the bit processor moves the address byte into the SR shift register, a byte boundary is achieved and the byte processor is triggered to execute the procedures in the ADDRESS state.

In the ADDRESS state the received station address is compared to the contents of the STAD register. If there is no match, or the address is not the broadcast address (FFH), reception will be aborted (SIUST = 01H). Otherwise, the byte processor jumps to the CONTROL state (SIUST = 10H) and goes into standby mode.

The byte processor jumps to the CONTROL state if there exists a control field in the receiving frame. In this state the control byte is moved to the RCB register by the byte processor. Note that the only action taken in this state is that a received byte, processed by the bit processor, is moved to RCB. There is no other hardware task performed, and DCNT and SRAR are not affected in this state.

The next two states, PUSH-1 and PUSH-2, will be executed if Frame check sequence (NFCS = 0) option is selected. In these two states the first and second bytes of the information field are pushed into the 3-byte FIFO (FIFO0, FIFO1, FIFO2) and the Receive Field Length register (RFL) is set to zero. The 3-byte FIFO is used as a pipeline to move received bytes into the internal RAM. The FIFO prevents transfer of CRC bytes and the closing flag to the receive buffer (i.e., when the ending flag is received, the contents of FIFO are FLAG, FCS1, and FCS0.) The three byte FIFO is collapsed to one byte in No FCS mode.

In the DMA-LOOP state the byte processor pushes a byte from SR to FIFO0, moves the contents of FIFO2 to the internal RAM addressed by the contents of SRAR, increments the SRAR and RFL registers, and decrements the DCNT register. If more information bytes are expected, the byte processor repeats this state on the next byte boundaries until DMA Buffer End occurs. The DMA Buffer End occurs if SRAR reaches 0BFH (192 decimal), DCNT reaches zero, or the RBP bit of the STS register is set.

The BOV-LOOP state, the last state, is executed if there is a buffer overrun. Buffer overrun occurs when the number of information bytes received is larger than the length of the receive buffer (RFL > RBL). This state is executed until the closing flag is received.

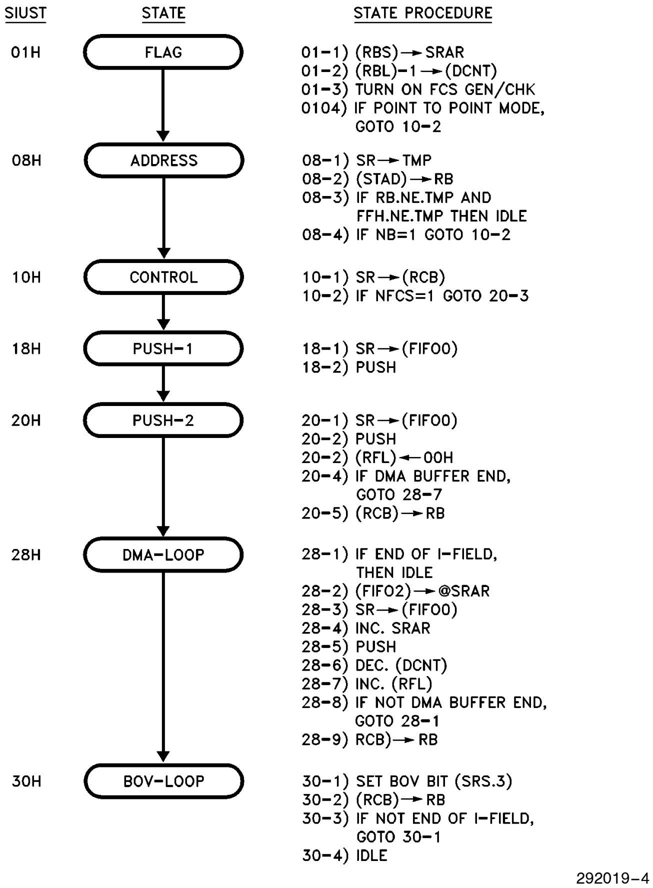
At the end of reception, if the FCS option is used, the closing flag and the FCS bytes will remain in the 3-byte FIFO. The contents of the RCB register are used to update the NSNR (Receive/Send Count) register. The SIU updates the STS register and sets the serial interrupt.

3.2 Transmit State Sequence

Setting the RTS bit puts the SIU in the transmit mode. When the CTS pin goes active, the byte processor goes into START-XMIT state. In this state the opening flag is moved into the RAM Buffer (RB) register. The byte processor jumps to the next state and goes into the standby mode.

If the Pre-Frame Sync (PFS) option is selected, the PFS1 and PFS2 states will be executed to transmit the two Pre-Frame Sync bytes (00H or 55H). In these two states the contents of the Pre-Frame Sync generator are sent to the serial port while the Zero Insertion Circuit (ZID) is turned off. ZID is turned back on automatically on the next byte boundary.

If the PFS option is not chosen, the byte processor jumps to the FLAG state. In this state, the byte processor moves the contents of TBS into the SRAR register, decrements TBL and moves the contents into the DCNT register. The byte processor turns off the ZID and turns on FCS GEN/CHK. The contents of FCS GEN/CHK are not transmitted unless the NFCS bit is



292019-4

Figure 4. Receive State Diagram

set. If a frame with the address field is chosen, it moves the contents of the STAD register into the RB register for transmission. At the same time, the opening flag is being transmitted by the bit processor.

In the ADDRESS (SIUST = A0H) and CONTROL (SIUST = A8H) states, TCB and the first information byte are loaded into the RB register for transmission, respectively. Note that in the CONTROL state, none of the registers (e.g. DCNT, SRAR) are incremented, and ZID and FCS GEN/CHK are not turned on or off.

The procedures in the DMA-LOOP state are similar to the procedures of the DMA-LOOP in the receive state diagram. The SRAR register pointer to the internal RAM is incremented, and the DCNT register is decremented. The contents of DCNT reach zero when all the information bytes from the transmit buffer are transmitted. A byte from RAM is moved to the RB register for transmission. This state is executed on the following byte boundaries until all the information bytes are transmitted.

The FCS1 and the FCS2 states are executed to transmit the Frame Check Sequence bytes generated by the FCS generator, and the END-FLAG state is executed to transmit the closing flag.

The XMIT-ACTION and the ABORT-ACTION states are executed by the byte processor to synchronize the SIU with the CPU clock. The XMIT-ACTION or the ABORT-ACTION state is repeated until the byte processor status is updated. At the end, the STS and the TMOD registers are updated.

The two ABORT-SEQUENCE states (SIUST = E0H and SIUST = E8H) are executed only if transmission is aborted by the CPU (RTS or TBF bit of the STS register is cleared) or by the serial data link (CTS signal goes inactive or shut-off occurs in loop mode.)

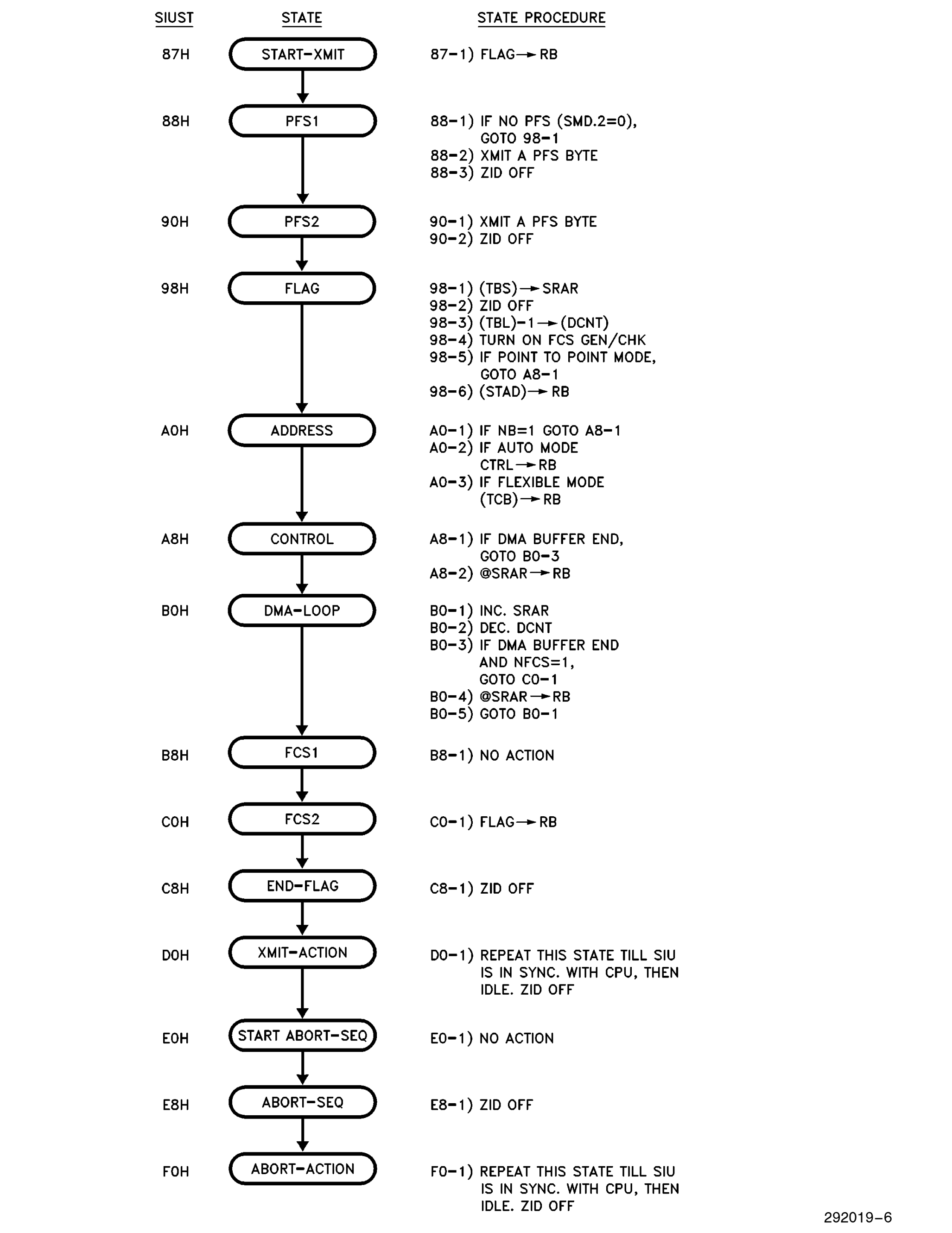


Figure 5. Transmit State Diagram

4.0 TRANSMISSION/RECEPTION OF LONG FRAMES (EXPANDED OPERATION)

In this application note, a frame whose information field is more than 192 bytes (size of on-chip RAM) is referred to as a long frame. The 8044 can access up to 64000 bytes of external RAM. Therefore, a long frame can have up to 64000 information bytes.

4.1 Description

During transmission or reception of a frame, while the bit processor is processing a byte, the byte processor, after 16 CPU states, is in the standby mode, and the internal registers and the internal bus are not used. The period between each byte boundary, when the byte processor is in the standby mode, can be used to move data from external RAM to one of the byte processor registers for transmission and vice versa for reception. The contents of the SIUST register, which dictate the state of the byte processor, can be monitored to recognize the beginning of each SDLC field and the consecutive byte boundaries.

By writing into the SIUST register, the byte processor can be forced to repeat or skip a specific state. As an example, the SIU can be forced to repeatedly put the received bytes into the RCB register. This is accomplished by writing E7H into the SIUST register when the byte processor goes into the standby mode. The byte processor, therefore, executes the CONTROL state at the next byte boundary.

For transmission, the byte processor is put in the transmit mode. When transmission of a frame is initiated, the user program calls a subroutine in which the state of the byte processor is monitored by checking the contents of the SIUST register. When the byte processor reaches a desired state and goes into standby, the CPU loads the first byte of the internal RAM buffer with data and moves the byte processor to the CONTROL state. The routine is repeated for every byte. At the end, the program returns from the subroutine, and the SIU finishes its task (see application examples).

For reception, a software routine is executed to move data to external RAM and to force the SIU to repeat the CONTROL state. The CONTROL state is repeated because, as shown in the receive state diagram, the only action taken by the byte processor, in the CONTROL state, is to move the contents of SR to the RCB register. None of the registers (e.g. SRAR and DCNT) are incremented. A similar comment justifies the use of the CONTROL state for transmission. In the transmit CONTROL state, contents of a location in the on-chip RAM addressed by TBS is moved to RB for transmission.

4.2 SIU Registers

To write into the SIUST register, the data must be complemented. For example, if you intend to write 18H into the SIUST register, you should write E7H to the register. The data read from SIUST is, however, true data (i.e. 18H).

Read and write accesses to the SIUST, STAD, DCNT, RCB, RBL, RFL, TCB, TBL, TBS, and the 3-byte FIFO registers are done on even and odd phases, respectively. Therefore, there is no bus contention when the CPU is monitoring the registers (e.g. SIUST), and SIU is simultaneously writing into them.

There is no need to change or reset the contents of any SIU register while transmitting or receiving long frames, unless the byte processor is forced to repeat a state in which the contents of these registers are modified. Note that the SRAR register can not be accessed by the CPU; therefore, avoid repeating the DMA-LOOP states. If SRAR increments to 192, the SIU will be interrupted and communication will be aborted.

4.3 Other Possibilities

The internal RAM, in conjunction with an external buffer (RAM or FIFOs), can be used as a transmit and receive buffer. In other words, Expanded and Normal operation can be used together. For example, if a frame with 300 Information bytes is received and only 255 of them are moved to an external buffer, the remaining bytes (45 bytes) will be loaded into the internal RAM by the SIU (assuming RBL is set to 45 or more). The contents of RFL indicate the number of bytes stored in the internal RAM. For transmission, the contents of the external buffer can be transmitted followed by the contents of the internal buffer.

If the internal RAM is not used, contents of the RBL register can be 0 and contents of the TBL register must be set to 1. The contents of the TBS register can be any location in the internal RAM.

The transmission and reception procedures for long frames with no FCS are similar to those with FCS. The exception is the contents of the SIUST register should be compared with different values since the two FCS states of the transmit and receive flow charts are skipped by the byte processor.

If a frame format with no control byte is chosen, a location in the RAM addressed by TBS should be used for transmission as with control byte format. The FIFO can be used for reception. The STAD register can be used for transmission if no zero insertion is required.

If the RUPI is used in Auto mode (see Section 5), it will still respond to RR, RNR, REJ, and Unnumbered Poll (UP) SDLC commands with RR or RNR automatically, without using any transmit routine. For example, if the on-chip CPU is busy performing some real time operations, the SIU can transmit an information frame from the internal buffer or transmit a supervisory frame without the help of CPU (Normal operation).

Maximum data rate using this feature is limited primarily by the number of instructions needed to be executed during the standby mode.

Transmission or reception of a frame can be timed out so that the CPU will not hang up in the transmit or receive procedures if a frame is aborted. Or, if the data rate allows enough time (standby time is long enough), the CPU can monitor the SIUST register for idle mode (SIUST = 01H).

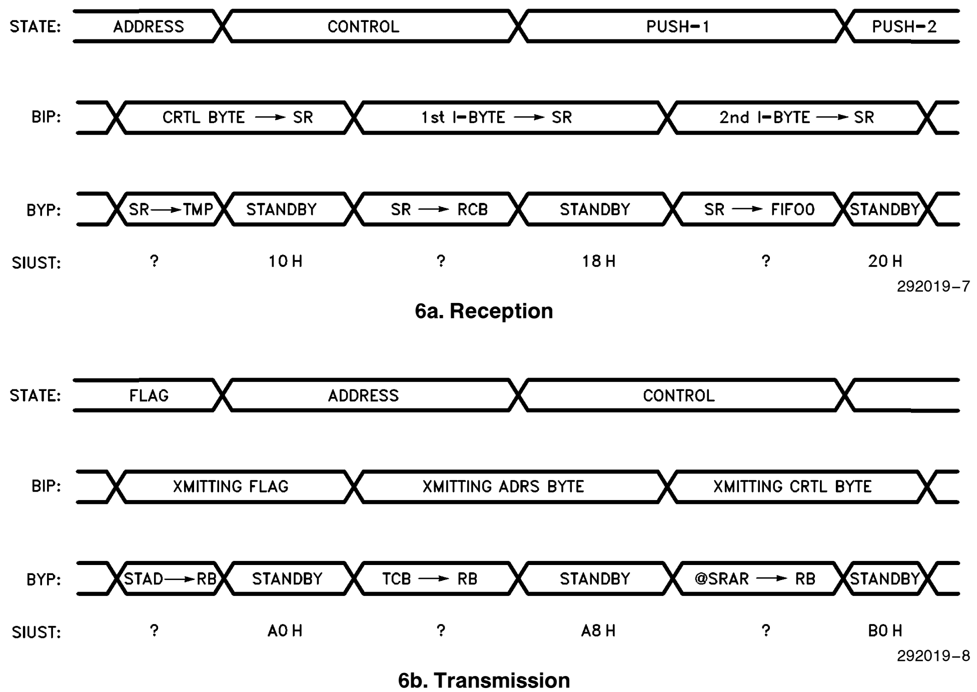
It is also possible to transmit multiple opening or closing flags by forcing the byte processor to repeat the END-FLAG state.

4.4 Maximum Data Rate in Expanded Operation

Assuming there is no zero-insertion/deletion, the bit processor requires eight serial clock periods to process one block of data. The byte processor, running on the CPU clock, processes one byte of data in 16 CPU states (one state of the state diagrams). Each CPU state is two oscillator periods. At an oscillator frequency of 12 MHz, the CPU clock is 6 MHz, and 16 CPU states is 2.7 μs. At a 3 Mbit rate with no zero-insertion/deletion, there is exactly enough time to execute one state per byte (16 states at 6 MHz = 8 bits at 3M baud). In other words, the standby time is zero.

Figure 6 demonstrates portions of the timing relationship between the byte processor and the bit processor. In each state, the actions taken by the processors, plus the contents of the SIUST register, are shown. When the byte processor is running, the contents of SIUST are unknown. However, when it is in the standby mode, its contents are determinable.

The maximum data rate for transmitting and receiving long frames depends on the number of instructions needed to be executed during standby, and is propor-



6a. Reception

6b. Transmission

Figure 6. Portions of the BIP/BYP Timing Relationship

tional to the oscillator frequency. The time the byte processor is in the standby mode, waiting for the bit processor to deliver a processed byte, is at least equal to eight serial clock periods minus 16 CPU states. If an inserted zero is in the block of data, the bit processor will process the byte in nine serial clock periods.

The equation for theoretical maximum data rate is given as:

$$(2TCLCL) \times (16 \text{ states}) + (\# \text{ of instruction cycles}) \times (12TCLCL) = (8TDCY) \qquad \text{Equation (1)}$$

Where: TCLCL is the oscillator period.
TDCY is the serial clock period.

At an oscillator frequency of 12 MHz and baud rate of 375 Kbps, about 18 instruction cycles can be executed when the byte processor is in the standby mode. At a 9600 baud rate, there is time to execute about 830 instruction cycles—plenty of time to service a long interrupt routine or perform bit-manipulation or arithmetic operations on the data while transmission or reception is taking place.

5.0 MODES OF OPERATION

The 8044 has two modes: Flexible mode and Auto mode. In Auto mode, the chip responds to many SDLC commands and keeps track of frame sequence numbering automatically without on-chip CPU intervention. In Flexible mode, communication tasks are under control of the on-chip CPU.

5.1 Flexible Mode

For transmission, the CPU allocates space for transmit buffer by storing values for the starting location and size of the transmit buffer in the TBS and the TBL registers. It loads the buffer with data, sets the TBF and the RTS bits in the STS register, and proceeds to perform other tasks. The SIU activates the RTS line. When the CTS signal goes active, the SIU transmits the frame. At the end of transmission, the SIU clears the RTS bit and interrupts the CPU (SI set).

For reception, the CPU allocates space for receive buffer by loading the beginning address and length of the receive buffer into the RBS and RBL registers, sets the RBE bit, and proceeds to perform other tasks. The SIU, upon detection of an opening flag, checks the next received byte. If it matches the station address, it will load the received control byte into RCB, and received information bytes into the receive buffer. At the end of reception, if the Frame Check Sequence (FCS) is correct, the SIU clears RBE and interrupts the CPU.

5.2 Auto Mode

In the Auto mode, the 8044 can only be a secondary station operating in the SDLC "Normal Response Mode". The 8044 in Auto mode does not transmit messages unless it is polled by the primary.

For transmission of an information frame, the CPU allocates space for the transmit buffer, loads the buffer with data, and sets the TBF bit. The SIU will transmit the frame when it receives a valid poll-frame. A frame whose poll bit of the control byte is set, is a poll-frame. The poll bit causes the RTS bit to be set. If TBF were not set, the SIU would respond with Receive Not Ready (RNR) SDLC command if RBP = 1, or with Receive Ready (RR) SDLC command if RBP = 0. After transmission RTS is cleared, and the CPU is not interrupted.

For reception, the procedure is the same as that of Flexible mode. In addition, the SIU sets the RTS bit if the received frame is a poll-frame (causing an automatic response) and increments the NS and NR counts accordingly.

6.0 APPLICATION EXAMPLES

Two application examples are given to provide additional details about the procedures used to transmit and receive long frames. In the first application example, procedures to construct receive and transmit software routines for the point-to-point frame format are described. The point-to-point frame has the information field and the FCS field enclosed between two flags (see Figure 7). In the second example software code is generated for reception and transmission of the standard SDLC frame. The SDLC frame has the pattern: flag, address, control, information, FCS, flag.

The first example focuses on the construction of transmit and receive code which allow the chip to transmit and receive long frames. The second example shows how to make more use of the 8044 features, such as the on-chip phase locked loop for clock recovery and automatic responses in the Auto mode to demonstrate the capability of the 8044 to achieve high throughput when Expanded operation is used.

6.1 Point-to-Point Application Example

A point-to-point communication system was developed to receive and transmit long frames. The system consists of one primary and one secondary station. Although multiple secondary stations can be used in this

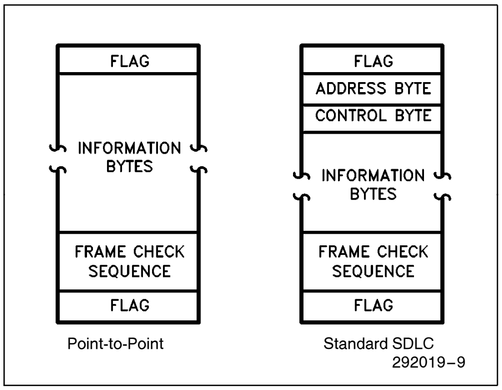


Figure 7. Point-to-Point and Standard SDLC Frame Formats

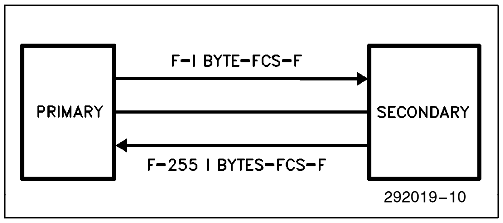


Figure 8. Secondary Responses to Primary Station Commands

system, one secondary is chosen to simplify the primary station's software and focus on the long frame software code. Both the primary and the secondary stations are in Flexible mode and the external clock option is used for the serial channel. The maximum data rate is 500 Kbps. The FCS bytes are generated and checked automatically by both stations.

6.1.1 POLLING SEQUENCE

The polling sequence, shown in Figure 8, takes place continuously between the primary and the secondary stations. The primary transmits a frame with one information byte to the secondary. The information byte is used by the secondary as an address byte. The secondary checks the received byte, and if the address matches, the secondary responds with a long frame. In this example, the information field of the frame is chosen to be 255 bytes long. Since there is only one secondary station, the address always matches. Upon successful reception of the long frame, the primary transmits another frame to the secondary station.

6.1.2 HARDWARE

The schematic of the secondary station is given in Figure 9. The circuit of the primary station is identical to the secondary station with the exception of pin 11 (DATA) being connected to pin 14 (T0). In the primary station, the 8044 is interrupted when activity is detected on the communication line by the on-chip timer (in counter mode). This is explained more later. The serial clock to both stations is supplied by a pulse generator. The output of the pulse generator (not shown in the diagram) is connected to pin 15 of the 8044s. Since the two stations are located near each other (less than 4 feet), line drivers are not used.

The central processor of each station is the 8044. The data link program is stored in a 2Kx8 EPROM (2732A), and a 2Kx8 static RAM (AM9128) is used as the external transmit and receive buffer. The RTS pin is connected to the CTS pin. For simplicity, the stations are assumed to be in the SDLC Normal Respond Mode after Hardware reset.

6.1.3 PRIMARY STATION SOFTWARE

The assembly code for the primary station software is listed in Appendix A. The primary software consists of the main routine, the SIU interrupt routine, and the receive interrupt routine. The receive interrupt routine is executed when a long frame is being received.

In the flow charts that follow, all actions taken by the SIU appear in squares, and actions taken by the on-chip CPU appear in spheres.

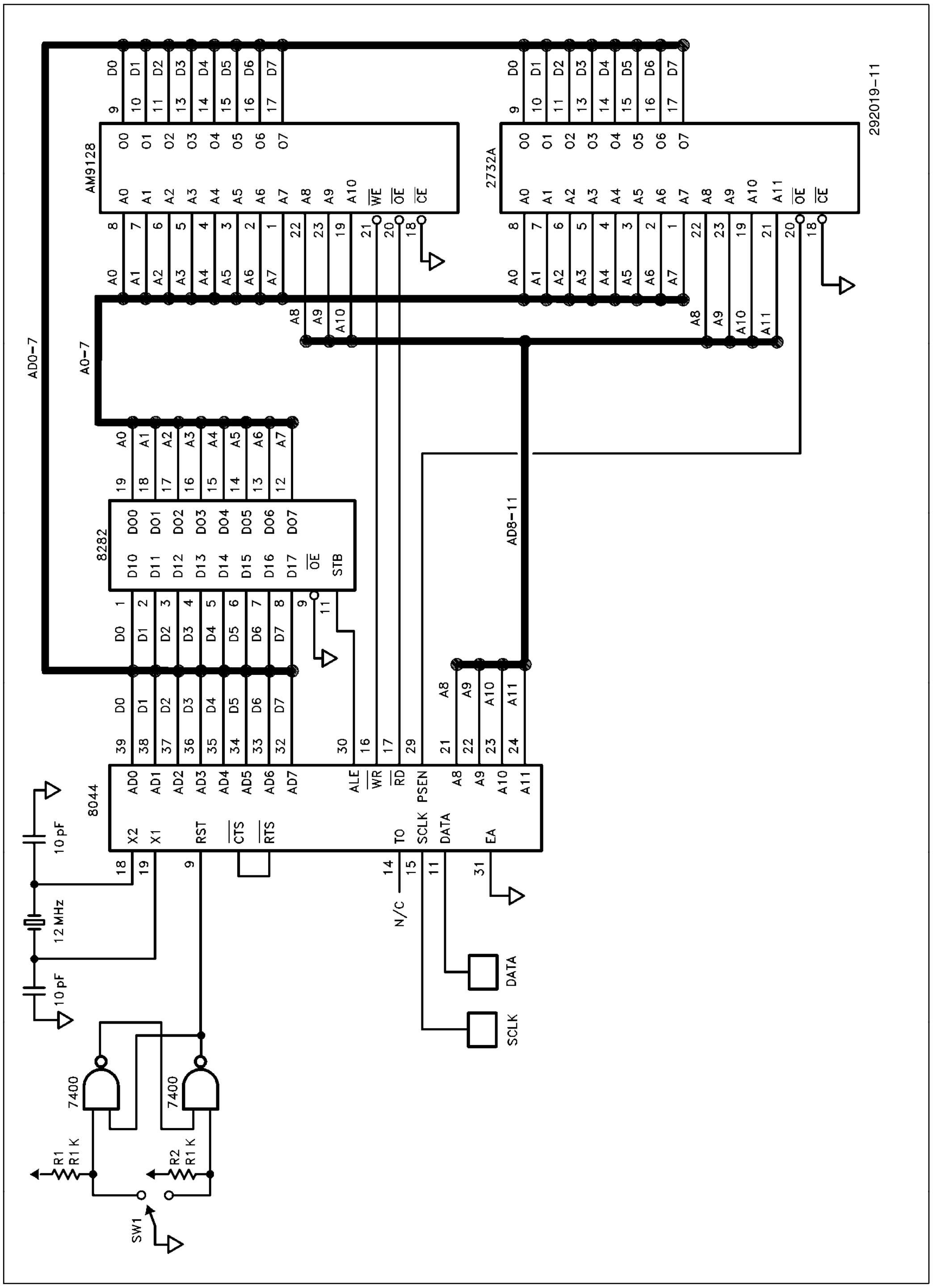


Figure 9. Secondary Station Hardware

Main Routine

First, the chip is initialized (see Figure 10). It is put in Flexible mode, externally clocked, and "Flag-Information Field-FCS-Flag" frame format. Pre-Frame Sync option (PFS = 1) and automatic Frame Check Sequence generation/detection (NFCS = 0) are selected. The on-chip transmit buffer starts at location 20H and the transmit buffer length is set to 1. This one byte buffer contains the address of the secondary station. There is no on-chip receive buffer since the long frame being received is moved to the external buffer. The RTS, TBF, and RBE bits are set simultaneously. Setting the RTS and TBF bits causes the SIU to transmit the contents of the transmit buffer.

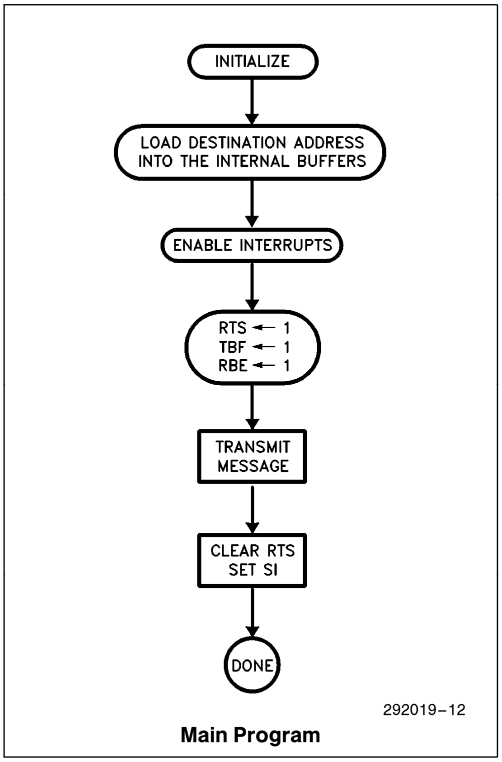


Figure 10. Primary Station Flow Charts

SIU Interrupt Routine

After transmission of the frame, the SIU interrupts the on-chip CPU (SI is set). In the SIU interrupt service routine, counter 0 is initialized and turned on (see Figure 11). The user program returns to perform other tasks. After reception of the long frame, the SIU interrupt routine is executed again. This time, RTS, TBF, and RBE are set for another round of information exchange between the two stations.

SIU never interrupts while reception or transmission is taking place. The SIU registers are updated and the SI is set (serial interrupt) after the closing flag has been received or transmitted. An SIU interrupt never occurs if the receive interrupt routine or the transmit subroutine is being executed.

Setting the RBE bit of the STS register puts the RUPI in the receive mode. However, the jump to the receive interrupt routine occurs only when a frame appears on the serial port. Incoming frames can be detected using the Pre-Frame Sync. option and one of the CPU timers in counter mode. The counter external pin (T0) is connected to the data line (pin 11 is tied to pin 14). Setting the PFS (Pre-Frame Sync.) bit will guarantee 16 transitions before the opening flag of a flame.

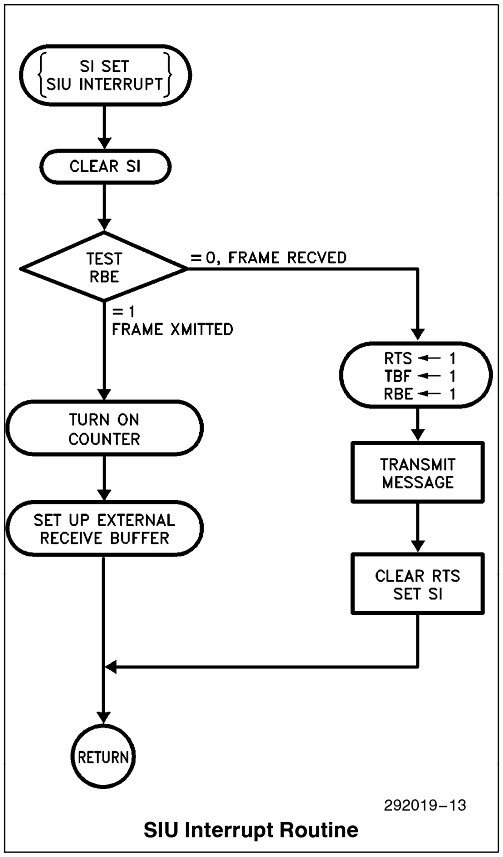


Figure 11. Primary Station Flow Charts

The counter registers are initialized such that the counter interrupt occurs before the opening flag of a frame. When the PFS transitions appear on the data line, the counter overflows and interrupts the CPU. The CPU program jumps to the timer interrupt service routine and executes the receive routine. In the receive routine, the received frame is processed, and the information bytes are moved to the external RAM. Note that the maximum count rate of the 8051 counter is $1/24$ of the oscillator frequency. At 12 MHz, the data rate is limited to 500 Kbps.

Another method to detect a frame on the data line and cause an interrupt is to use an external "Flag-Detect" circuit to interrupt the CPU. The "Flag Detect" circuit can be an 8-bit shift register plus some TTL chips. If this option is used, the RUPI must operate in externally clocked mode because the clock is needed to shift the incoming data into the shift register. With this option, the maximum data rate is not limited by the maximum count rate of the 8051 counter.

Receive Interrupt Routine

In Normal operation, the byte processor executes the procedures of the FLAG state, jumps to the CONTROL state without going into the standby mode, and executes 10–2 procedure of the state (see Figure 4). It then jumps to the PUSH-1 state and goes into the standby mode. At the following byte boundaries, the byte processor executes the PUSH-1, PUSH-2, and DMA-LOOP states, respectively. The receive interrupt routine as shown in the flow chart of Figure 12 and described below forces the byte processor to repeatedly execute the CONTROL state before the PUSH-1 state is executed. The following is the step by step procedure to receive long frames:

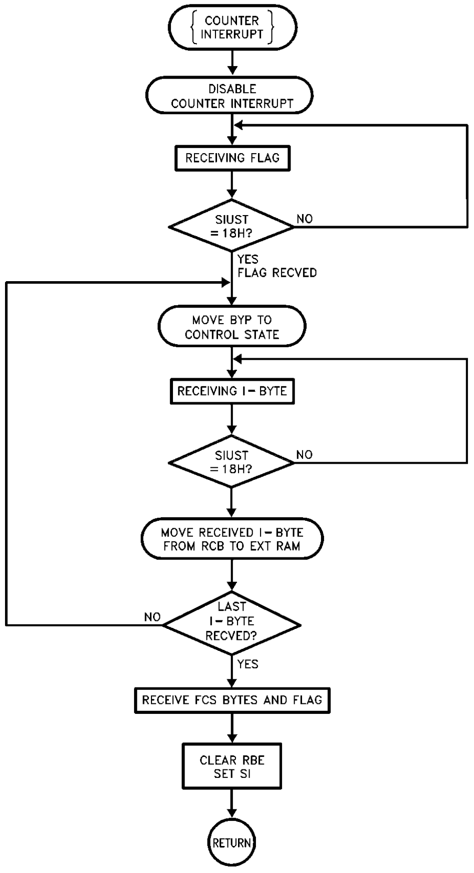
1) Turn off the CPU counter and save all the important registers. Jump to the receive interrupt routine, execution of the instructions to save registers, and initialization of the receive buffer pointer take place while the Pre-Frame Sync bytes and the opening flag are being received. This is about three data byte periods (48 CPU cycles at 500 Kbps).
2) Monitor the SIUST register for standby in the PUSH-1 state (SIUST = 18H). When the SIUST contents are 18H, the byte processor is waiting for the first information byte. The bit processor has already recognized the flag and is processing the first information byte.
3) In the standby mode, move the byte processor into the CONTROL state by writing "EFH" (complement of 10H) into the SIUST register. When the next byte boundary occurs, the bit processor has processed and moved a byte of data into the SR register. The byte processor moves the contents of SR into the RCB register, jumps to the PUSH-1 state (SIUST = 18H), and waits.
4) Monitor the SIUST register for standby in the PUSH-1 state. When the contents of SIUST becomes 18H, the contents of RCB are the first information byte of the information field.
5) While the byte processor is in the standby mode, move the contents of RCB to an external RAM or an I/O port.
6) Check for the end of the information field. The end can be detected by knowing the number of bytes transmitted, or by having a unique character at the end of information field. The length of the information field can be loaded into the first byte(s) received. The receive routine can load this byte into the loop counter.
7) If the byte received is not the last information byte, move the byte processor back to standby in the CONTROL state and repeat steps 4 through 6. Otherwise, return from the interrupt routine.

Upon returning from the receive interrupt routine, the byte processor automatically executes the PUSH-1, PUSH-2, and DMA-LOOP before it stops. This causes the remaining information bytes (if any) to be stored in the internal RAM at the starting location specified by the contents of RBS register. At the end of the cycle, the closing flag and the CRC bytes are left in the FIFO. The RFL register will be incremented by the number of bytes stored in the internal RAM. Then, the STS and NSNR registers are updated, and an appropriate response is generated by the SIU.

The software to perform the above task is given in Table 1. In this example, the number of instruction cycles executed during standby is 12 cycles.

Table 1. Codes for Long Frame Reception

```
Receive Codes                                                  Cycles
                    •          •
                    •          •
                    •          •
       REC:         CLR        TR0
                    MOV        A,#18H
       WAIT1:       CJNE       A,SIUST,WAIT1
       NEXTI:       MOV        SIUST,#0EFH .......................2
                    MOV        A,#18H ............................1
       WAIT2:       CJNE       A,SIUST,WAIT2......................2
                    MOV        A,RCB..............................1
                    MOVX       @DPTR,A ...........................2
                    INC        DPTR...............................2
                    DJNZ       R5,NEXTI...........................2
                    RETI                                    ________
       END                                                 12 Cycles
```



292019–14

Receive Interrupt Routine

Figure 12. Primary Station Flow Charts

6.1.4 SECONDARY STATION SOFTWARE

The assembly code for the secondary station software is given in Appendix A. The secondary station contains the transmit subroutine which is called for transmission of long frames.

Main Routine

As shown in the secondary station flow chart (Figure 13), the external transmit buffer (external RAM) is loaded with the information data (FFH, FEH, FDH, . . .) at starting location 200H. The internal transmit buffer (on chip RAM) starts at location 20H (TBS = 20H), and the transmit buffer length (TBL) is set to 1. The on-chip CPU, in the transmit subroutine, moves the information bytes from the external RAM to this one byte buffer for transmission. The receive buffer starts at location 10H and the receiver buffer length is 1. This buffer is used to buffer the frame transmitted by the primary. The received byte is used as an address byte.

The Secondary is configured like the Primary station. It is put in Flexible mode, externally clocked, Point-to-point frame format. The PFS bit is set to transmit two bytes before the first flag of a frame. The RBE bit is set to put the chip in receive mode. Upon reception of a valid frame, the SIU loads the received information byte into the on-chip receive buffer and interrupts the CPU.

SIU Interrupt Routine

In the serial interrupt routine, the RBE bit is checked (see Figure 14). Since RBE is clear, a frame has been received. The received Information byte is compared with the contents of the Station Address (STAD) register.

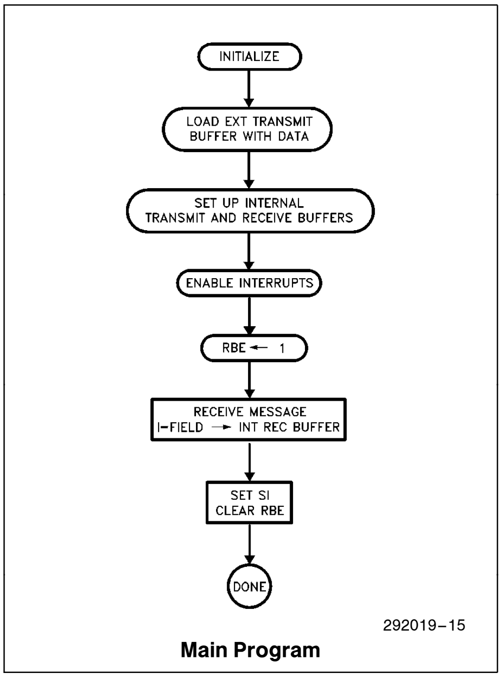

INITIALIZE
LOAD EXT TRANSMIT BUFFER WITH DATA
SET UP INTERNAL TRANSMIT AND RECEIVE BUFFERS
ENABLE INTERRUPTS
RBE ← 1
RECEIVE MESSAGE I-FIELD → INT REC BUFFER
SET SI CLEAR RBE
DONE
292019-15
Main Program

Figure 13. Secondary Station Flow Charts

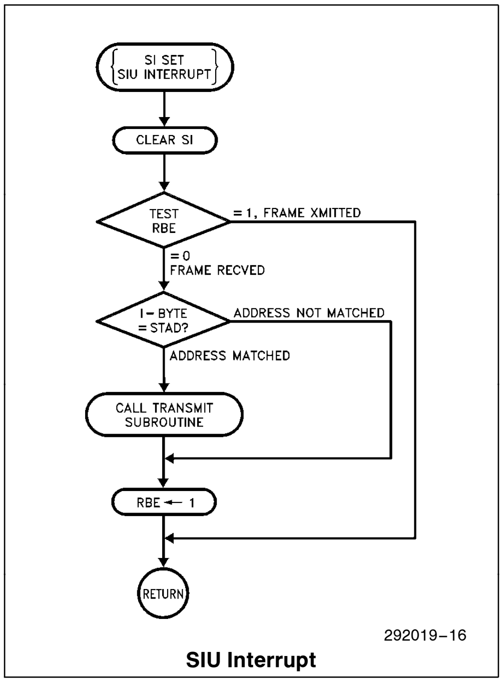


Figure 14. Secondary Station Flow Charts

If they match, the secondary will call the transmit subroutine to transmit the long frame. Upon returning from the transmit subroutine, the RBE bit is set, and program returns from the SIU interrupt. After transmission of the closing flag, SIU interrupt occurs again. In the interrupt routine, the RBE is checked. Since the RBE is set, the program returns from the SIU interrupt routine and waits until another long frame is received.

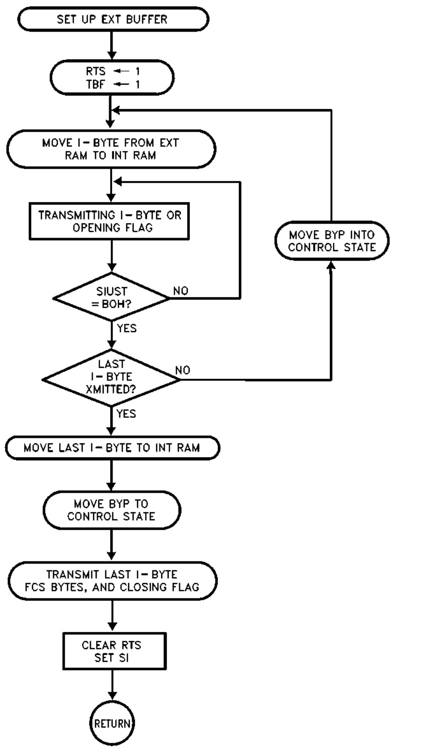
If the secondary were in Auto mode, the chip must be ready to execute the transmit routine upon reception of a poll-frame; otherwise, the chip automatically transmits the contents of the internal transmit buffer if the TBF bit is set, or transmits a supervisory command (RR or RNR) if TBF is clear.

Transmit Subroutine

In Normal operation the byte processor executes the START-TRANSMIT state and jumps to the PFS1 state. While the bit processor is transmitting some unwanted bits, the byte processor executes the PFS1 state and jumps to the standby mode in the PFS2 state. While the bit processor is transmitting the first Pre-Frame Sync byte, the byte processor executes the PFS2 state and jumps to the standby mode in the FLAG state. The FLAG state is executed when the bit processor begins to transmit the second Pre-Frame Sync byte. When the flag is being transmitted, the byte processor executes the 98-1, 98-2, 98-3, and 98-4 procedures of the FLAG state, and jumps to execute the A8-1 procedure of the CONTROL state. When the opening flag is transmitted, the contents of RB are the first information byte. (See transmit State diagram.)

In the transmit subroutine (see Figure 15), the byte processor is forced to repeat the CONTROL state before the DMA-LOOP state. In the CONTROL state, the contents of a RAM location addressed by the TBS register are moved to the RB register. The following is the step by step procedure to transmit long frames:

1) Put the chip in transmit mode by setting the RTS and TBF bits.
2) Move an information byte from external RAM to a location in the internal RAM addressed by the contents of TBS.


SET UP EXT BUFFER
RTS ← 1
TBF ← 1
MOVE I-BYTE FROM EXT RAM TO INT RAM
TRANSMITTING I-BYTE OR OPENING FLAG
SIUST = BOH?
NO
YES
LAST I-BYTE XMITTED?
NO
YES
MOVE BYP INTO CONTROL STATE
MOVE LAST I-BYTE TO INT RAM
MOVE BYP TO CONTROL STATE
TRANSMIT LAST I-BYTE FCS BYTES, AND CLOSING FLAG
CLEAR RTS SET SI
RETURN

292019–17

Transmit Subroutine

Figure 15. Secondary Station Flow Charts

3) Monitor the SIUST register for the standby mode in the DMA-LOOP state (SIUST = B0H). When SIUST is B0H, the opening flag has been transmitted, and the first information byte is being transmitted by the bit processor.
4) If there are more information bytes, move the byte processor back to the CONTROL state, and repeat steps 2 through 4. Otherwise, continue.
5) Move byte processor to the Standby mode in the CONTROL state (SIUST = A8H) and return from the subroutine.

The byte processor automatically executes the remaining states to send the FCS bytes and the closing flag. After the completion of transmission, SIU updates the STS and NSNR registers and interrupts the CPU.

If the contents of the TBL register were more than 1, the SIU transmits (TBL) − 1 additional bytes from the internal RAM at starting address (TBS) + 1 because it executes the DMA-LOOP state (TBL) − 1 additional times. The byte processor should not be programmed to skip the DMA-LOOP state, because the transmission of FCS bytes is enabled in this state.

The maximum baud rate that can be used with these codes is calculated by adding the number of instruction cycles executed, during the standby mode, between each byte boundaries (see Table 2).

Using Equation 1, the maximum data rate, based on the transmit software, is 509 Kbps; However, the maximum count rate of the counter limits the data rate to 500 Kbps.

Table 2. Codes for Long Frame Transmission

```
Transmit Codes                                                Cycles
                        •           •
                        •           •
                        •           •
          TRAN:         MOV         DPTR,#200H
                        MOV         R5,#0FFH
                        SETB        TBF
                        SETB        RTS
          LOOP:         MOVX        A,@DPTR
                        MOV         @R1,A
                        MOV         A,#0B0H
          WAIT1:        CJNE        A,SIUST,WAIT1...................2
                        INC         DPTR............................2
                        MOVX        A,@DPTR ........................2
                        MOV         @R1,A ..........................1
                        DJNZ        R5,NEXTI........................2
                        MOV         SIUST,#57H
                        RET
          NEXTI:        MOV         SIUST,#57H .....................2
                        MOV         A,#0B0H ........................1
                        JMP         WAIT1...........................1
          END
                                                           ---------
                                                           13 Cycles
```

6.2 Multidrop Application

Performance of long frame in addition to the features of the 8044 are described using a simple multidrop communication system in which three RUPIs, one as a master and the other two as secondary stations, transmit and receive long frames alternately (see Figure 16). All stations perform automatic zero bit insertion/deletion, NRZI decoding/encoding, Frame Check Sequence (FCS) generation/detection, and on-chip clock recovery at a data rate of 375 Kbps.

The primary and the secondary station's software code is given in Appendix B. These programs, for simplicity, assume only reception of information and supervisory frames. It is also assumed that the frames are received and transmitted in order. All stations use very similar transmit and receive routines. This code is written for standard SDLC frames (see Figure 7).

6.2.1 POLLING SEQUENCE

The primary station, in Flexible mode, transmits a long frame (for this example, 255 I-bytes), polls one of the secondary stations, and acknowledges a previously received frame simultaneously (see Figure 17). Both secondary stations, in Auto mode, detect the transmitted frame and check its address byte. One of the secondary stations receives the frame, stores the Information bytes in an external RAM buffer, and transmits the same data back to the primary. After reception of the frame, the primary polls and transmits a long frame to the other secondary station which will respond with the same long frame.

6.2.2 HARDWARE

The schematic of the secondary station hardware is shown in Figure 18. The primary station's hardware is similar to the secondary station's hardware. The exception is in secondary stations only, where the RTS signal is inverted and tied to the interrupt 0 input pin (INT0). In the primary station, RTS is tied to CTS. At each station, software codes are stored in external EPROM (2732A). Static RAM (2Kx8) is used as external transmit/receive buffer. There is no hardware handshaking done between the stations. The serial clock is extracted from the data line using the on-chip phase locked loop.

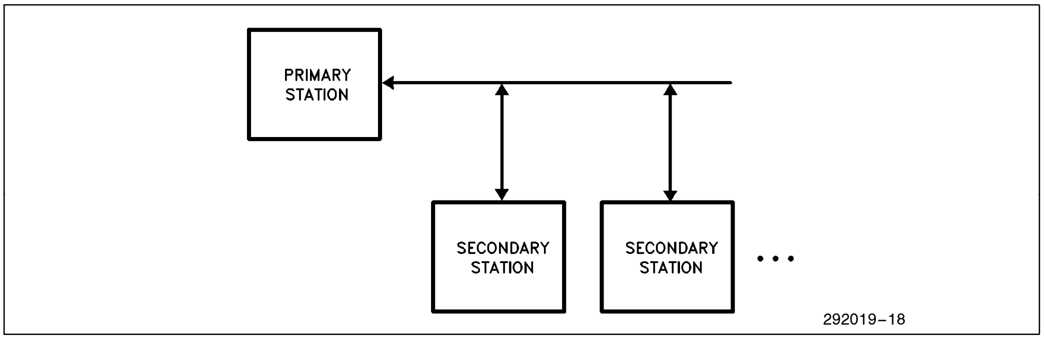


Figure 16. SDLC Multidrop Application Example

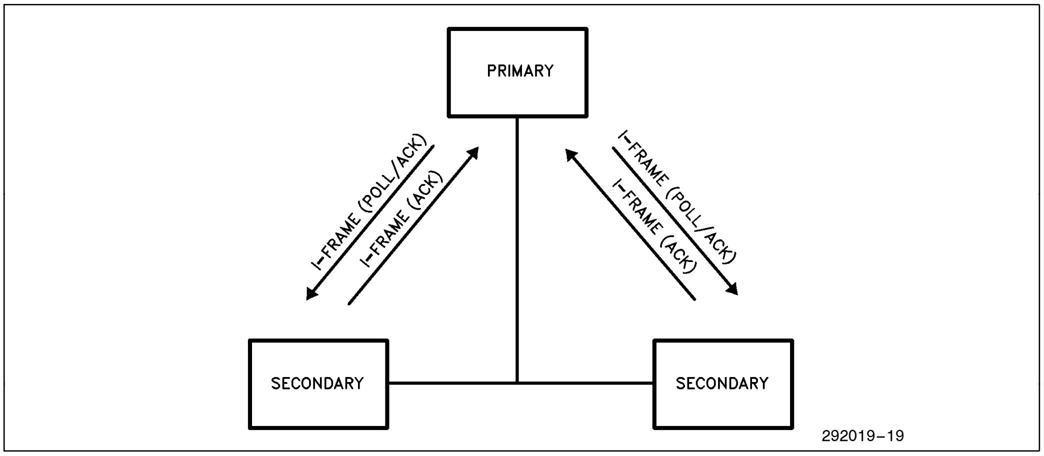


Figure 17. Polling Sequence Between the Primary and Secondary Stations

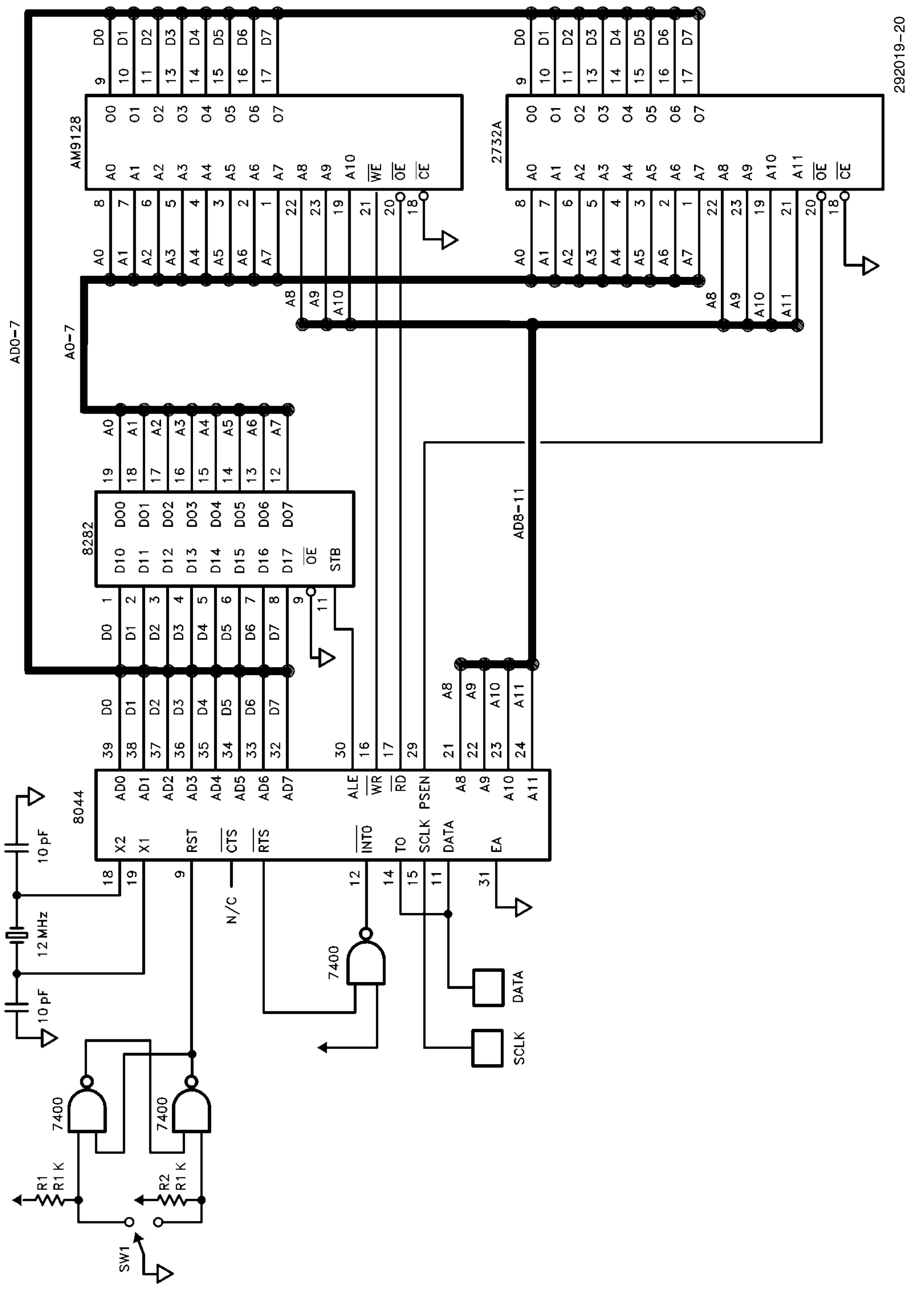


Figure 18. Secondary Station Hardware

6.2.3 PRIMARY SOFTWARE

Main Routine

During initialization (see Figure 19), the 8044 is set to Flexible mode, internally clocked at 375 Kbps, and configured to handle standard SDLC frames. The on-chip receive and transmit buffer starting addresses and lengths are selected. The external transmit buffer is chosen from physical location 200H to location 2FFH (255 bytes). The external transmit buffer (external RAM) is loaded with data (FFH, FEH, FDH, FCH, . . . 00H). Timer 0 is put in counter mode and set to priority 1. The counter register (TL0) is loaded such that interrupt occurs after 8 transitions on the data line. The Pre-Frame Sync option (setting bit 2 of the SMD register) is selected to guarantee at least 16 transitions before the opening flag of a frame.

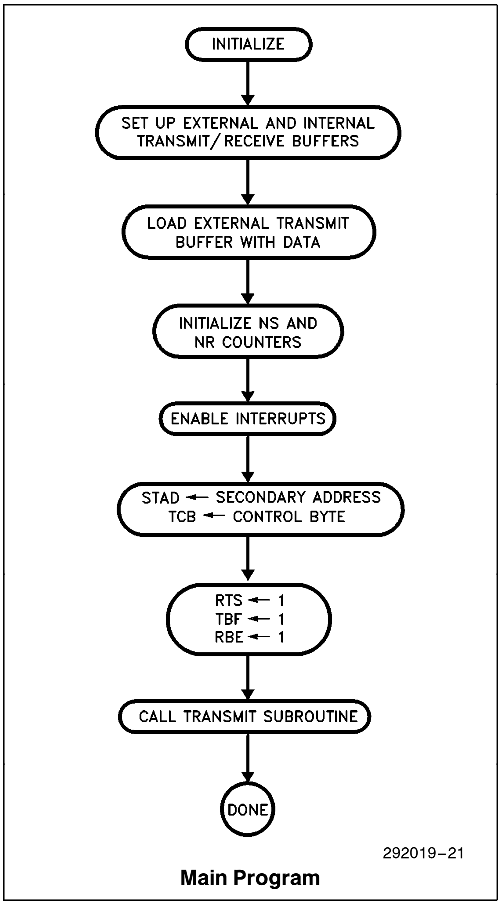


Figure 19. Primary Station Flow Charts

The station address register (STAD) is loaded with address of one of the secondary stations. The RTS, TBF, and RBE bits of the STS register are simultaneously set and a call to the transmit routine follows. The transmit routine transmits the contents of the external transmit buffer. At the end of transmission, RTS and TBF are cleared by the SIU, and SIU interrupt occurs. In Flexible mode, SIU interrupt occurs after every transmission or reception of a frame.

SIU Interrupt Routine

In the SIU interrupt service routine (see Figure 20), SI is cleared and the RBE bit is checked. If RBE is set, a long frame has been transmitted. The first time through the SIU interrupt service routine, the RBE test indicates a long frame has been transmitted to one of the secondary stations. Therefore, the Counter is initialized

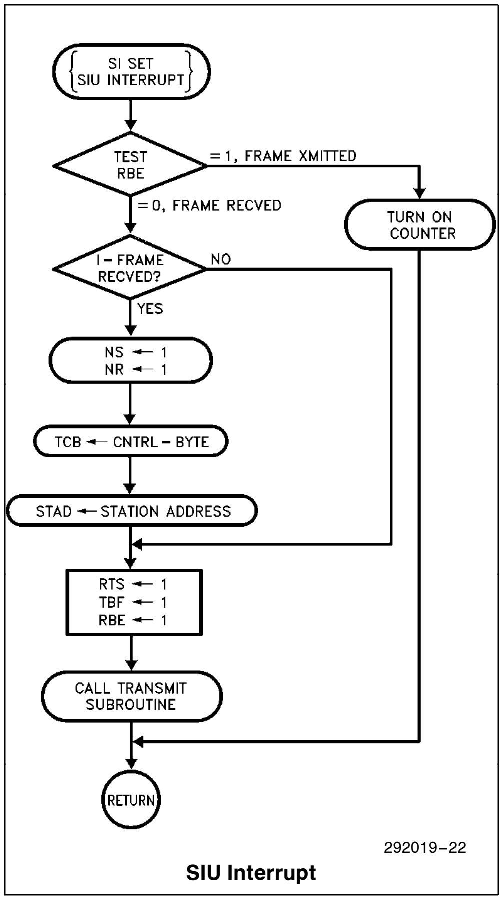


Figure 20. Primary Station Flow Charts

and turned on. The program returns from the interrupt routine before a frame appears on the communication channel.

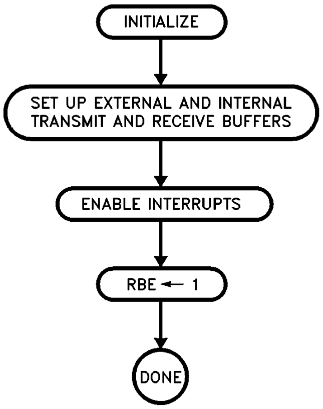
When a frame appears on the communication line, counter interrupt occurs and the receive routine is executed to move the incoming bytes into the external RAM. After reception of the frame and return from the receive routine, SIU interrupt occurs again.

In the SIU interrupt routine, RBE is checked. Since the RBE bit is clear, a frame has been received. Therefore, the appropriate NS and NR counters are incremented and loaded into the TCB register (two pairs of internal RAM bytes keep track of NS and NR counts for the two secondary stations). Transmission of a frame to the next secondary station is enabled by setting the RTS and the TBF bits. The chip is also put in receive mode (RBE set), and a call to transmit routine is made. After transmission, SIU interrupt occurs again, and the process continues.

6.2.4 SECONDARY SOFTWARE

Main Routine

Both secondary stations have identical software (Appendix B). The only differences are the station addresses. Contents of the STAD register are 55H for one station and 44H for the other.



292019–23

Main Program

Figure 21. Secondary Station Flow Charts

During initialization, the chip is set to Auto mode, standard SDLC frame, and internally clocked at 375 Kbps (see Figure 21). Internal buffer registers: RBS, RBL, TBS, and TBL are initialized. The RBE bit is set and the counter 0 is turned on.

The secondary is configured to transmit an Information frame every time it is polled. The RTS pin is inverted and tied to INT1 pin. External interrupt 1 is enabled and set to interrupt on low to high transition of the RTS signal. This will cause an interrupt (EX1 set) after a frame is transmitted. In the interrupt routine the CTS pin is cleared to prevent any automatic response from the secondary. If the CTS pin were not disabled, the secondary station would respond with a supervisory frame (RNR) since the TBF is set to zero by the SIU due to the acknowledge. In the SIU interrupt routine, the CTS pin is cleared after the TBF bit is set. If this option is not used, the primary should acknowledge the previously received frame and poll for the next frame in two separate transmissions.

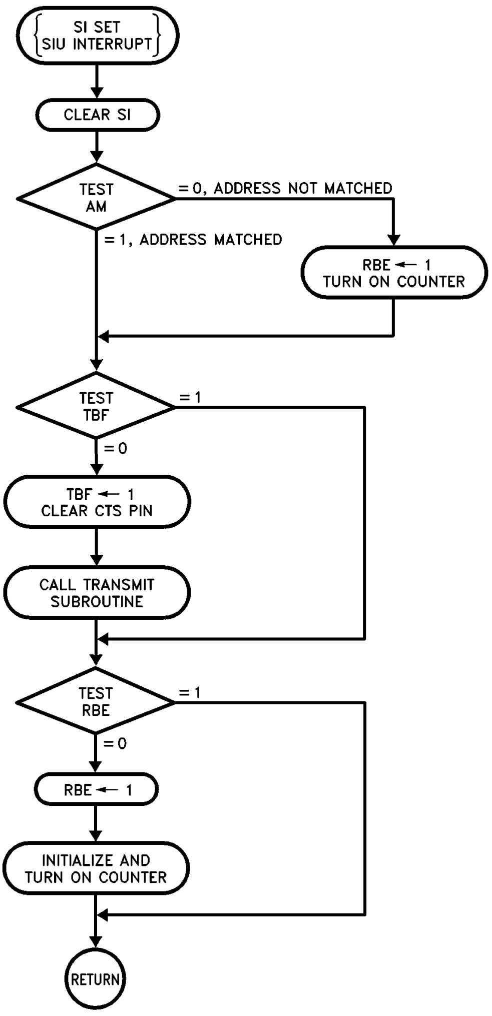
SIU Interrupt Routine

When a frame is received, counter 0 interrupt occurs and the receive routine is executed (see Figure 22). If the incoming frame is addressed to the station, the information bytes are stored in external RAM; Otherwise, the program returns from the receive routine to perform other tasks. At the end of the frame, SIU interrupt occurs. In Auto mode, SIU interrupt occurs whenever an Information frame or a supervisory frame is received. Transmission will not cause an interrupt. In the SIU interrupt service routine, the AM bit of the STS is checked.

If AM bit is set, the interrupt is due to a frame whose address did not match with the address of the station. In this case, NFCS, AM, and the BOV bits are cleared, the RBE bit is set, the counter 0 is initialized and turned on, and program returns from the interrupt routine.

If AM bit is not set, a valid frame has been received and stored in the external RAM. TBF bit is set, CTS pin is activated, counter 0 is disabled and a call to transmit routine is made which transmits the contents of external transmit buffer. This frame also acknowledges the reception of the previously received frame (NS and NR are automatically incremented). Upon return from the transmit routine RBE is set and counter 0 is turned on, thereby putting the chip in the receive mode for another round of data exchange with the primary.

Note that, if the second station is in receive mode, and the counter is enabled and turned on, the CPU will be interrupted each time a frame is on the communication channel. If the frame is not addressed to the secondary station, the chip enters the receive routine, executes only a few lines of code (address comparison) and returns to perform other tasks. This interrupt will not occupy the CPU for more than two data byte periods (43 microseconds at 375 Kbps). At the end of the frame, the BOV bit is set by the SIU, and the SIU interrupt occurs. In the SIU interrupt service routine,



292019–24

SIU Interrupt

Figure 22. Secondary Station Flow Charts

the RBE bit is set and the counter is turned on which put the chip back in the receive mode.

6.2.5 RECEIVE INTERRUPT ROUTINE

Assembly code for the receive interrupt routine can be found in both primary and secondary software (Appendix B). The receive interrupt routine of the primary station is very similar to that of the primary station in example 1. In the following two sections the receive and transmit routine of the secondary stations are discussed.

In the receive interrupt service routine (see Figure 23), counter 0 is turned off, important registers are saved, receive buffer starting address and receive buffer length of the external RAM are set (do not confuse the external RAM settings with that of the internal RAM buffer.)

After reception of an opening flag, the byte processor jumps to the ADDRESS state and waits until the bit processor processes and moves the receiving address byte to SR. Then, the byte processor is triggered to execute the state. In the secondary stations, the CPU monitors the SIUST register for the ADDRESS state (SIUST = 08H). When the ADDRESS state is reached, the byte processor is moved to the next state (CONTROL state), and the ADDRESS state is skipped. Therefore, when the address byte is moved to SR, the byte processor executes the CONTROL state rather than the ADDRESS state and then jumps to the PUSH-1 state. The execution of the CONTROL state causes the contents of SR (the received address byte) to be loaded into the RCB register.

The CPU checks the contents of RCB with the contents of the STAD (Station Address) register. If they match, the receive routine continues to store the received Information bytes in the external RAM buffer; Otherwise, the byte processor is moved to the very last state (BOV-LOOP), and the program returns from the routine to perform other tasks. The byte processor executes the BOV-LOOP state in each byte boundary until the closing flag of the frame is reached. It then sets the BOV bit and interrupts the CPU (serial interrupt SI set). In the serial interrupt routine the counter 0 is turned back on, and the station is reset back to the receive mode (RBE set).

In Normal operation, in the ADDRESS state, the received address byte is automatically compared with the station address. If they match, the byte processor executes the remaining states; otherwise, the byte processor goes into the idle mode (SIUST = 01H) and waits for the opening flag of the next frame. In the expanded operation, this state is skipped to avoid idle mode. If the byte processor went into the idle mode, clocks which run the byte processor would be turned off, and the byte processor can not be moved to any other states by the CPU. When the byte processor is in idle mode, counter 0 can not be turned on immediately because counter interrupt occurs on the same frame, and program returns to the receive routine and stays there.

If the address byte matches the station address, the byte processor is moved to the CONTROL state again. This time, after execution of the CONTROL state the contents of RCB are the received control byte.

CPU investigates the type of received frame by checking the received control byte. If the receiving frame is not an information frame (i.e. Supervisory frame), execution of receive routine will be terminated to free the

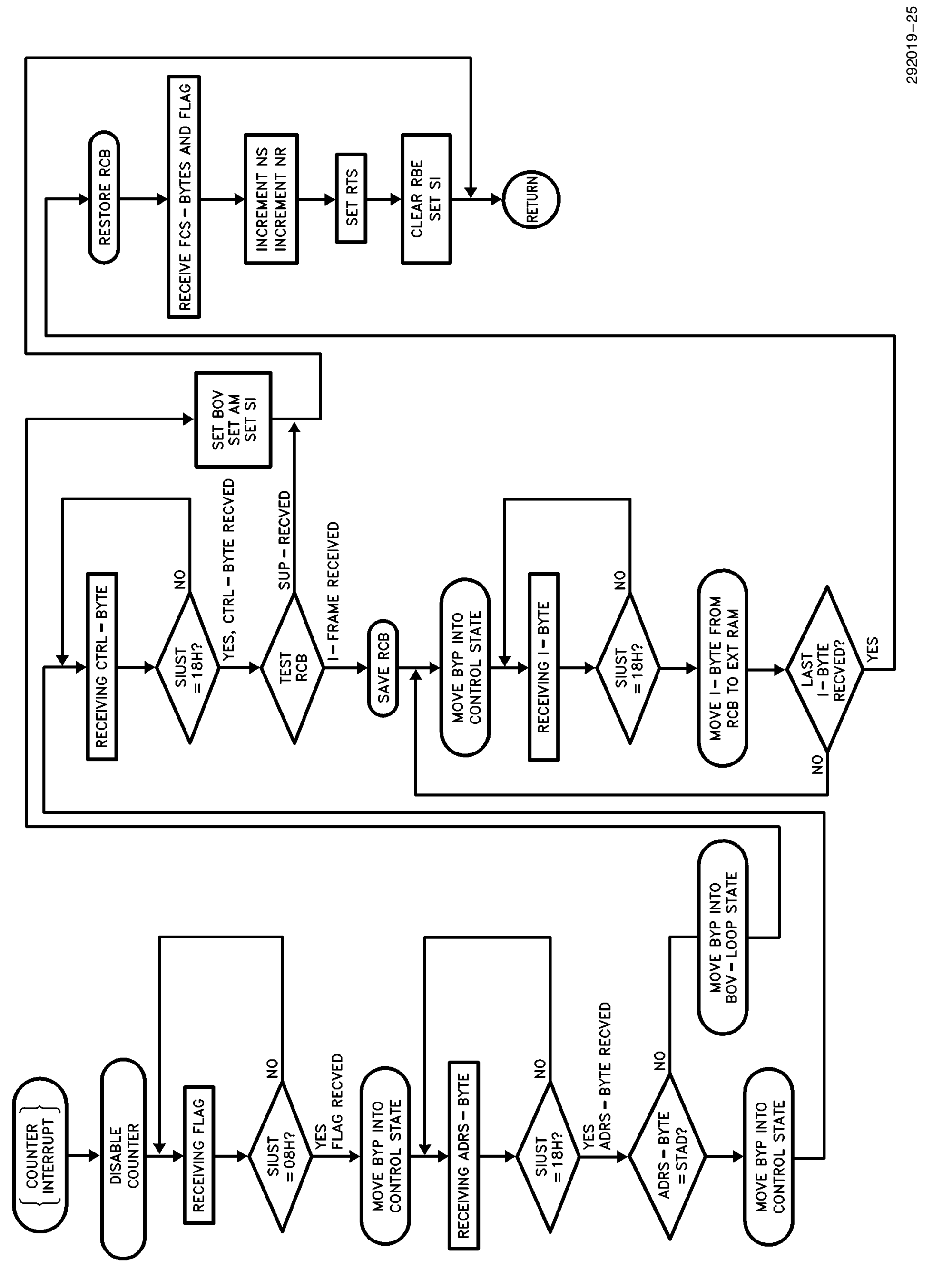


Figure 23. Receive Flow Chart (secondary station)

CPU. In Auto mode, the SIU checks the control byte and responds automatically in response to the supervisory frame.

After the control byte is received, it is saved in the stack. The byte processor is moved to the CONTROL state so that the next incoming byte will also be loaded into the RCB register. The byte processor remains in CONTROL state until a byte is processed by the bit processor and moved to SR. The byte processor is then triggered to move the contents of SR to the RCB register. The CPU monitors SIUST and waits until the first Information byte is loaded into the RCB register.

When byte processor reaches the PUSH-1 state (SIUST = 18H), RCB contains the first Information byte. The byte is moved to external RAM (receive buffer), and the byte processor is moved back to the CONTROL state. The process continues until all of the Information bytes are received. When all the Information bytes are received, the program returns from the routine. The byte processor automatically goes through the remaining states, updates the STS register, and interrupts the CPU as it would in Normal operation.

6.2.6 TRANSMIT SUBROUTINE

The transmit subroutine codes can be found in the primary and the secondary software (Appendix B). The transmit subroutines of the Primary and secondary stations are identical. A call to transmit routine is made when the RTS and TBF bits of the STS register are set. In Auto mode, RTS is set automatically upon reception of a poll-frame (poll bit of the control byte is set).

In the transmit routine (see Figure 15), the starting address and the transmit buffer length of the external buffer are set. Then the CPU monitors the SIUST register for CONTROL state (SIUST = A8H). In the CONTROL state the bit processor transmits the control byte, while the byte processor goes into the standby mode after it has moved the contents of a location in the internal RAM addressed by the contents of Transmit Buffer Start (TBS) register to the RB register.

While the control byte is being transmitted and the byte processor is in standby, the CPU moves an Information byte from external RAM to the internal RAM location addressed by TBS. The byte processor is then moved to CONTROL state. This will cause the byte processor, in the next byte boundary, to move the contents of the same location in the internal RAM to the RB register (see transmit state diagram.)

When this byte is being transmitted, the byte processor jumps to the DMA-LOOP state (SIUST = B0H) and waits. When the DMA-LOOP state is reached (CPU monitors SIUST for B0H), the CPU loads the next Information byte into the same location in the internal RAM and moves the byte processor to the CONTROL state before it gets to execute the DMA-LOOP state. Note that the same location in the internal RAM is used to transmit the subsequent Information bytes.

When all the Information bytes from the external RAM are transmitted, the byte processor is free to go through the remaining states so that it will transmit the FCS bytes and the closing flag.

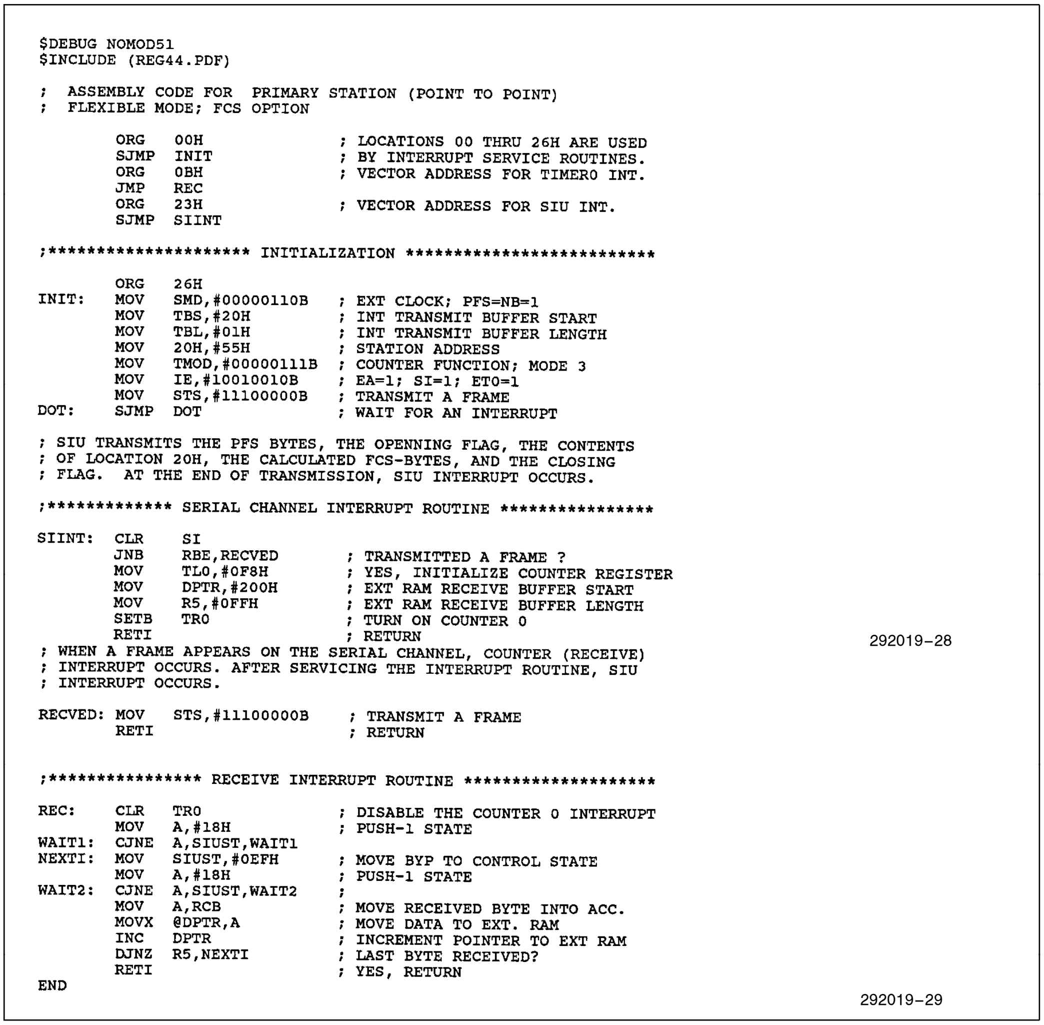
7.0 CONCLUSIONS

The RUPI, with addition of only a few bytes of code, can accept and transmit large frames with some compromise in the maximum data rate. It can be used in Auto or Flexible mode, with external or internal clocking, automatic CRC checking, and zero bit insertion/deletion. In addition, almost all of the internal RAM is available to be used as general purpose registers, or in conjunction with the external RAM as transmit and receive buffers.

All in all, this feature opens up new areas of applications for this device. Besides transmitting/receiving long frames, it may now be possible to perform arithmetic operations or bit manipulation (e.g. data scrambling) while transmission or reception is taking place, resulting in high throughput. Transmission of continuous flags and transmission with no zero insertion are also possible.

In addition to unlimited frame size, an on-chip controller, automatic SDLC responses, full support of SDLC protocol, 192 bytes of internal RAM, and the highest data rate in self clocked mode compared to other chips make this product very attractive.

APPENDIX A
LISTING OF SOFTWARE MODULES FOR APPLICATION EXAMPLE 1

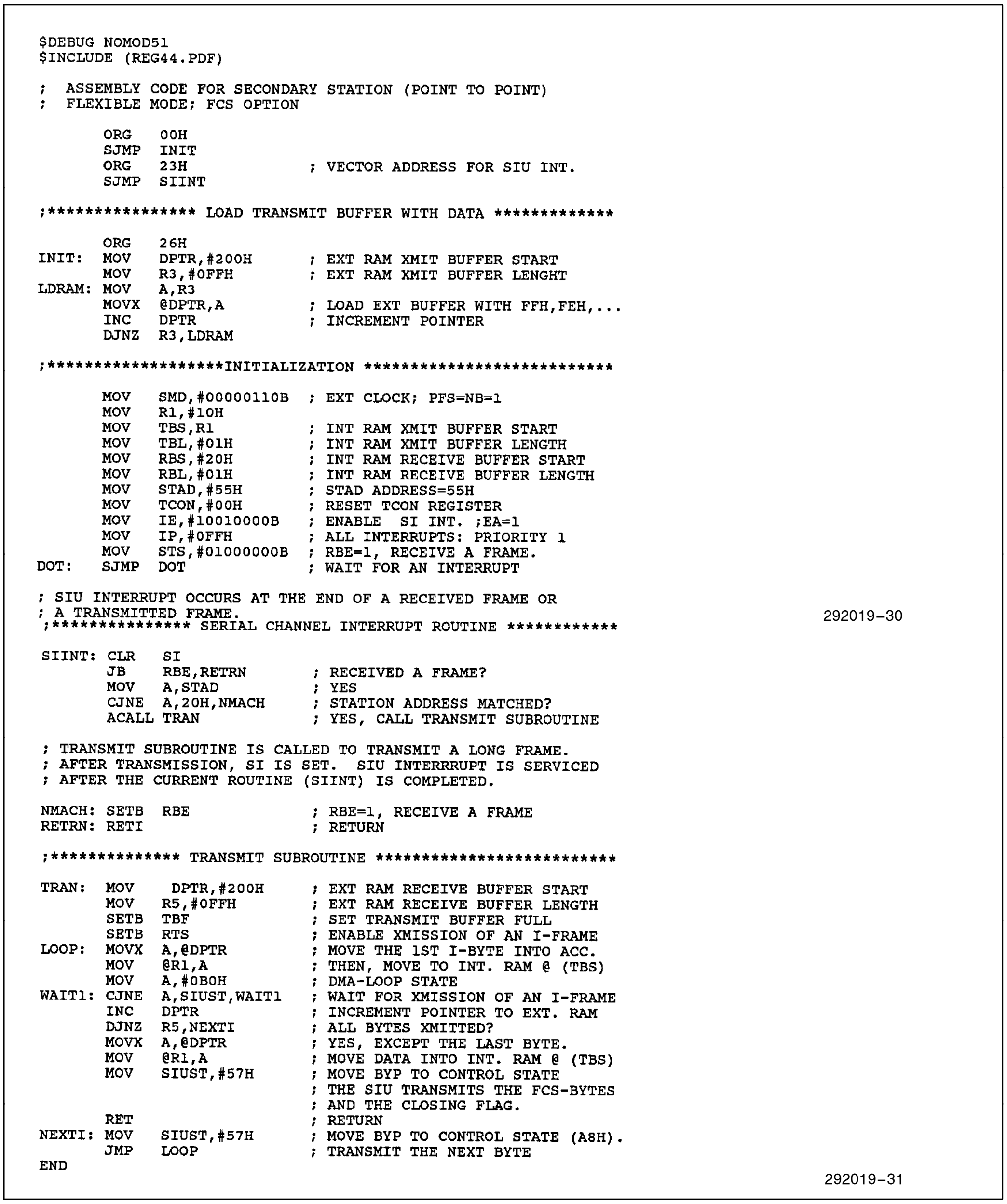


```
$DEBUG NOMOD51
$INCLUDE (REG44.PDF)

;  ASSEMBLY CODE FOR  PRIMARY STATION (POINT TO POINT)
;  FLEXIBLE MODE; FCS OPTION

        ORG   00H              ; LOCATIONS 00 THRU 26H ARE USED
        SJMP  INIT             ; BY INTERRUPT SERVICE ROUTINES.
        ORG   0BH              ; VECTOR ADDRESS FOR TIMER0 INT.
        JMP   REC
        ORG   23H              ; VECTOR ADDRESS FOR SIU INT.
        SJMP  SIINT

;********************** INITIALIZATION ***************************

        ORG   26H
INIT:   MOV   SMD,#00000110B   ; EXT CLOCK; PFS=NB=1
        MOV   TBS,#20H         ; INT TRANSMIT BUFFER START
        MOV   TBL,#01H         ; INT TRANSMIT BUFFER LENGTH
        MOV   20H,#55H         ; STATION ADDRESS
        MOV   TMOD,#00000111B  ; COUNTER FUNCTION; MODE 3
        MOV   IE,#10010010B    ; EA=1; SI=1; ET0=1
        MOV   STS,#11100000B   ; TRANSMIT A FRAME
DOT:    SJMP  DOT              ; WAIT FOR AN INTERRUPT

; SIU TRANSMITS THE PFS BYTES, THE OPENNING FLAG, THE CONTENTS
; OF LOCATION 20H, THE CALCULATED FCS-BYTES, AND THE CLOSING
; FLAG.  AT THE END OF TRANSMISSION, SIU INTERRUPT OCCURS.

;************* SERIAL CHANNEL INTERRUPT ROUTINE ****************

SIINT:  CLR    SI
        JNB    RBE,RECVED       ; TRANSMITTED A FRAME ?
        MOV    TL0,#0F8H        ; YES, INITIALIZE COUNTER REGISTER
        MOV    DPTR,#200H       ; EXT RAM RECEIVE BUFFER START
        MOV    R5,#0FFH         ; EXT RAM RECEIVE BUFFER LENGTH
        SETB   TR0              ; TURN ON COUNTER 0
        RETI                    ; RETURN
```

292019-28

```
; WHEN A FRAME APPEARS ON THE SERIAL CHANNEL, COUNTER (RECEIVE)
; INTERRUPT OCCURS. AFTER SERVICING THE INTERRUPT ROUTINE, SIU
; INTERRUPT OCCURS.

RECVED: MOV   STS,#11100000B    ; TRANSMIT A FRAME
        RETI                    ; RETURN


;**************** RECEIVE INTERRUPT ROUTINE ********************

REC:    CLR   TR0              ; DISABLE THE COUNTER 0 INTERRUPT
        MOV   A,#18H           ; PUSH-1 STATE
WAIT1:  CJNE  A,SIUST,WAIT1
NEXTI:  MOV   SIUST,#0EFH      ; MOVE BYP TO CONTROL STATE
        MOV   A,#18H           ; PUSH-1 STATE
WAIT2:  CJNE  A,SIUST,WAIT2    ;
        MOV   A,RCB            ; MOVE RECEIVED BYTE INTO ACC.
        MOVX  @DPTR,A          ; MOVE DATA TO EXT. RAM
        INC   DPTR             ; INCREMENT POINTER TO EXT RAM
        DJNZ  R5,NEXTI         ; LAST BYTE RECEIVED?
        RETI                   ; YES, RETURN
END
```

292019-29



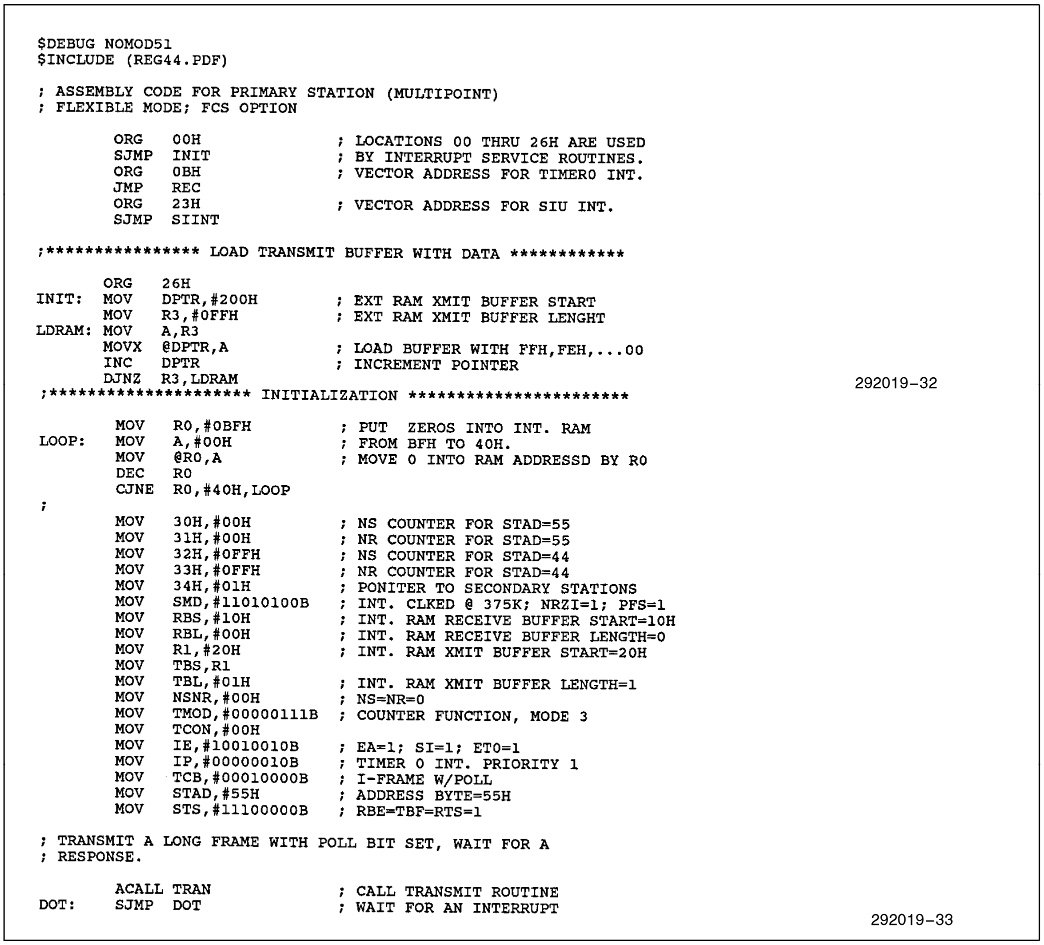
```
$DEBUG NOMOD51
$INCLUDE (REG44.PDF)

;  ASSEMBLY CODE FOR SECONDARY STATION (POINT TO POINT)
;  FLEXIBLE MODE; FCS OPTION

       ORG   00H
       SJMP  INIT
       ORG   23H             ; VECTOR ADDRESS FOR SIU INT.
       SJMP  SIINT

;***************** LOAD TRANSMIT BUFFER WITH DATA *************

       ORG   26H
INIT:  MOV   DPTR,#200H      ; EXT RAM XMIT BUFFER START
       MOV   R3,#0FFH        ; EXT RAM XMIT BUFFER LENGHT
LDRAM: MOV   A,R3
       MOVX  @DPTR,A         ; LOAD EXT BUFFER WITH FFH,FEH,...
       INC   DPTR            ; INCREMENT POINTER
       DJNZ  R3,LDRAM

;*******************INITIALIZATION ***************************

       MOV   SMD,#00000110B  ; EXT CLOCK; PFS=NB=1
       MOV   R1,#10H
       MOV   TBS,R1          ; INT RAM XMIT BUFFER START
       MOV   TBL,#01H        ; INT RAM XMIT BUFFER LENGTH
       MOV   RBS,#20H        ; INT RAM RECEIVE BUFFER START
       MOV   RBL,#01H        ; INT RAM RECEIVE BUFFER LENGTH
       MOV   STAD,#55H       ; STAD ADDRESS=55H
       MOV   TCON,#00H       ; RESET TCON REGISTER
       MOV   IE,#10010000B   ; ENABLE  SI INT. ;EA=1
       MOV   IP,#0FFH        ; ALL INTERRUPTS: PRIORITY 1
       MOV   STS,#01000000B  ; RBE=1, RECEIVE A FRAME.
DOT:   SJMP  DOT             ; WAIT FOR AN INTERRUPT

; SIU INTERRUPT OCCURS AT THE END OF A RECEIVED FRAME OR
; A TRANSMITTED FRAME.
```

292019–30

```
;*************** SERIAL CHANNEL INTERRUPT ROUTINE ************

SIINT: CLR   SI
       JB    RBE,RETRN       ; RECEIVED A FRAME?
       MOV   A,STAD          ; YES
       CJNE  A,20H,NMACH     ; STATION ADDRESS MATCHED?
       ACALL TRAN            ; YES, CALL TRANSMIT SUBROUTINE

; TRANSMIT SUBROUTINE IS CALLED TO TRANSMIT A LONG FRAME.
; AFTER TRANSMISSION, SI IS SET.  SIU INTERRRUPT IS SERVICED
; AFTER THE CURRENT ROUTINE (SIINT) IS COMPLETED.

NMACH: SETB  RBE             ; RBE=1, RECEIVE A FRAME
RETRN: RETI                  ; RETURN

;************** TRANSMIT SUBROUTINE **************************

TRAN:  MOV    DPTR,#200H     ; EXT RAM RECEIVE BUFFER START
       MOV   R5,#0FFH        ; EXT RAM RECEIVE BUFFER LENGTH
       SETB  TBF             ; SET TRANSMIT BUFFER FULL
       SETB  RTS             ; ENABLE XMISSION OF AN I-FRAME
LOOP:  MOVX  A,@DPTR         ; MOVE THE 1ST I-BYTE INTO ACC.
       MOV   @R1,A           ; THEN, MOVE TO INT. RAM @ (TBS)
       MOV   A,#0B0H         ; DMA-LOOP STATE
WAIT1: CJNE  A,SIUST,WAIT1   ; WAIT FOR XMISSION OF AN I-FRAME
       INC   DPTR            ; INCREMENT POINTER TO EXT. RAM
       DJNZ  R5,NEXTI        ; ALL BYTES XMITTED?
       MOVX  A,@DPTR         ; YES, EXCEPT THE LAST BYTE.
       MOV   @R1,A           ; MOVE DATA INTO INT. RAM @ (TBS)
       MOV   SIUST,#57H      ; MOVE BYP TO CONTROL STATE
                             ; THE SIU TRANSMITS THE FCS-BYTES
                             ; AND THE CLOSING FLAG.
       RET                   ; RETURN
NEXTI: MOV   SIUST,#57H      ; MOVE BYP TO CONTROL STATE (A8H).
       JMP   LOOP            ; TRANSMIT THE NEXT BYTE
END
```

292019–31

APPENDIX B
LISTING OF SOFTWARE MODULES
FOR APPLICATION EXAMPLE 2

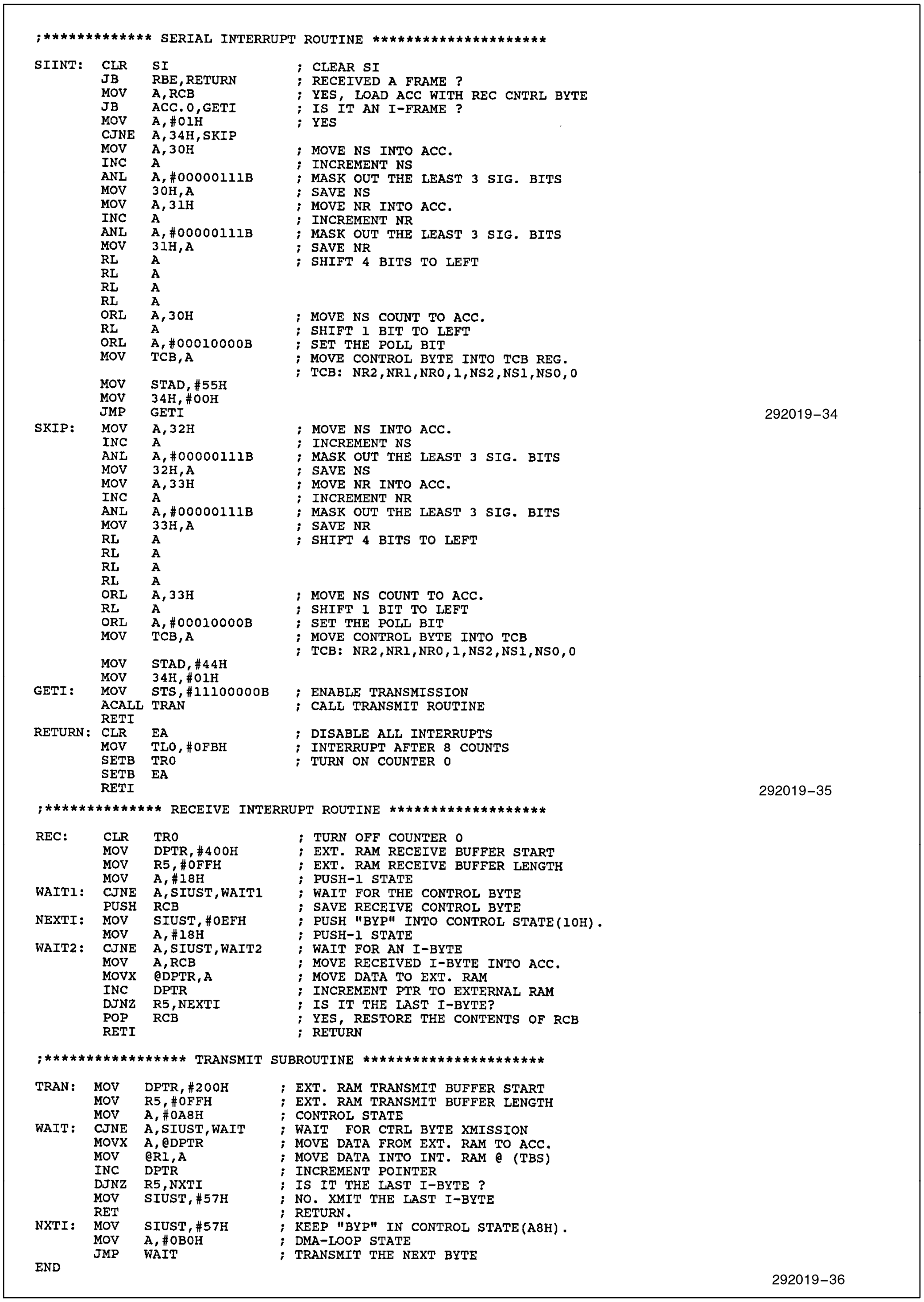


```
$DEBUG NOMOD51
$INCLUDE (REG44.PDF)

; ASSEMBLY CODE FOR PRIMARY STATION (MULTIPOINT)
; FLEXIBLE MODE; FCS OPTION

        ORG   00H               ; LOCATIONS 00 THRU 26H ARE USED
        SJMP  INIT              ; BY INTERRUPT SERVICE ROUTINES.
        ORG   0BH               ; VECTOR ADDRESS FOR TIMER0 INT.
        JMP   REC
        ORG   23H               ; VECTOR ADDRESS FOR SIU INT.
        SJMP  SIINT

;**************** LOAD TRANSMIT BUFFER WITH DATA ************

       ORG   26H
INIT:  MOV   DPTR,#200H         ; EXT RAM XMIT BUFFER START
       MOV   R3,#0FFH           ; EXT RAM XMIT BUFFER LENGHT
LDRAM: MOV   A,R3
       MOVX  @DPTR,A            ; LOAD BUFFER WITH FFH,FEH,...00
       INC   DPTR               ; INCREMENT POINTER
       DJNZ  R3,LDRAM
;******************** INITIALIZATION ***********************
```

292019-32

```
        MOV   R0,#0BFH          ; PUT  ZEROS INTO INT. RAM
LOOP:   MOV   A,#00H            ; FROM BFH TO 40H.
        MOV   @R0,A             ; MOVE 0 INTO RAM ADDRESSD BY R0
        DEC   R0
        CJNE  R0,#40H,LOOP
;
        MOV   30H,#00H          ; NS COUNTER FOR STAD=55
        MOV   31H,#00H          ; NR COUNTER FOR STAD=55
        MOV   32H,#0FFH         ; NS COUNTER FOR STAD=44
        MOV   33H,#0FFH         ; NR COUNTER FOR STAD=44
        MOV   34H,#01H          ; PONITER TO SECONDARY STATIONS
        MOV   SMD,#11010100B    ; INT. CLKED @ 375K; NRZI=1; PFS=1
        MOV   RBS,#10H          ; INT. RAM RECEIVE BUFFER START=10H
        MOV   RBL,#00H          ; INT. RAM RECEIVE BUFFER LENGTH=0
        MOV   R1,#20H           ; INT. RAM XMIT BUFFER START=20H
        MOV   TBS,R1
        MOV   TBL,#01H          ; INT. RAM XMIT BUFFER LENGTH=1
        MOV   NSNR,#00H         ; NS=NR=0
        MOV   TMOD,#00000111B   ; COUNTER FUNCTION, MODE 3
        MOV   TCON,#00H
        MOV   IE,#10010010B     ; EA=1; SI=1; ET0=1
        MOV   IP,#00000010B     ; TIMER 0 INT. PRIORITY 1
        MOV   TCB,#00010000B    ; I-FRAME W/POLL
        MOV   STAD,#55H         ; ADDRESS BYTE=55H
        MOV   STS,#11100000B    ; RBE=TBF=RTS=1

; TRANSMIT A LONG FRAME WITH POLL BIT SET, WAIT FOR A
; RESPONSE.

        ACALL TRAN              ; CALL TRANSMIT ROUTINE
DOT:    SJMP  DOT               ; WAIT FOR AN INTERRUPT
```

292019-33



```
;************* SERIAL INTERRUPT ROUTINE *********************

SIINT:  CLR   SI                 ; CLEAR SI
        JB    RBE,RETURN         ; RECEIVED A FRAME ?
        MOV   A,RCB              ; YES, LOAD ACC WITH REC CNTRL BYTE
        JB    ACC.0,GETI         ; IS IT AN I-FRAME ?
        MOV   A,#01H             ; YES
        CJNE  A,34H,SKIP
        MOV   A,30H              ; MOVE NS INTO ACC.
        INC   A                  ; INCREMENT NS
        ANL   A,#00000111B       ; MASK OUT THE LEAST 3 SIG. BITS
        MOV   30H,A              ; SAVE NS
        MOV   A,31H              ; MOVE NR INTO ACC.
        INC   A                  ; INCREMENT NR
        ANL   A,#00000111B       ; MASK OUT THE LEAST 3 SIG. BITS
        MOV   31H,A              ; SAVE NR
        RL    A                  ; SHIFT 4 BITS TO LEFT
        RL    A
        RL    A
        RL    A
        ORL   A,30H              ; MOVE NS COUNT TO ACC.
        RL    A                  ; SHIFT 1 BIT TO LEFT
        ORL   A,#00010000B       ; SET THE POLL BIT
        MOV   TCB,A              ; MOVE CONTROL BYTE INTO TCB REG.
                                 ; TCB: NR2,NR1,NR0,1,NS2,NS1,NS0,0
        MOV   STAD,#55H
        MOV   34H,#00H
        JMP   GETI
```

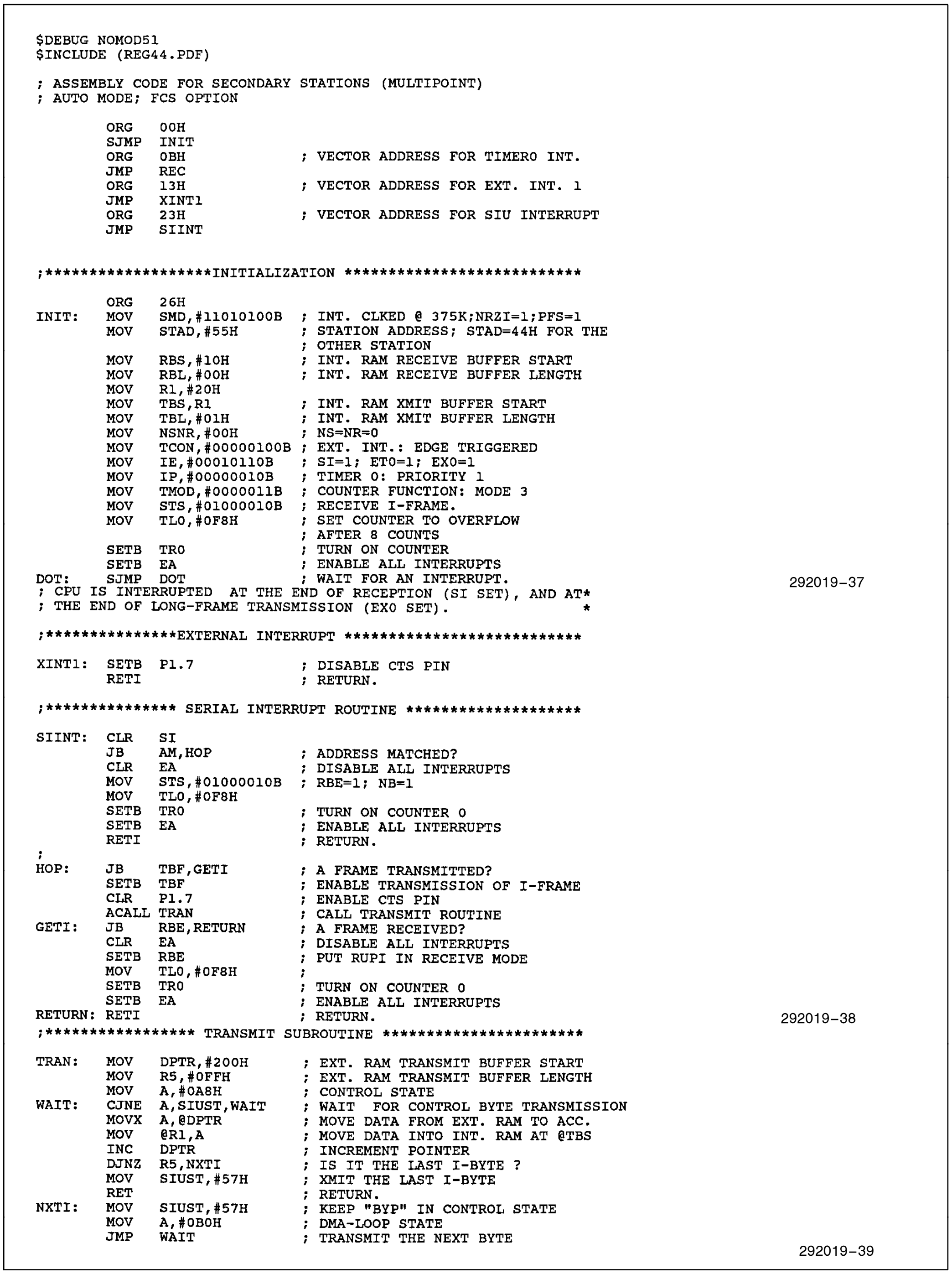
292019–34

```
SKIP:   MOV   A,32H              ; MOVE NS INTO ACC.
        INC   A                  ; INCREMENT NS
        ANL   A,#00000111B       ; MASK OUT THE LEAST 3 SIG. BITS
        MOV   32H,A              ; SAVE NS
        MOV   A,33H              ; MOVE NR INTO ACC.
        INC   A                  ; INCREMENT NR
        ANL   A,#00000111B       ; MASK OUT THE LEAST 3 SIG. BITS
        MOV   33H,A              ; SAVE NR
        RL    A                  ; SHIFT 4 BITS TO LEFT
        RL    A
        RL    A
        RL    A
        ORL   A,33H              ; MOVE NS COUNT TO ACC.
        RL    A                  ; SHIFT 1 BIT TO LEFT
        ORL   A,#00010000B       ; SET THE POLL BIT
        MOV   TCB,A              ; MOVE CONTROL BYTE INTO TCB
                                 ; TCB: NR2,NR1,NR0,1,NS2,NS1,NS0,0
        MOV   STAD,#44H
        MOV   34H,#01H
GETI:   MOV   STS,#11100000B     ; ENABLE TRANSMISSION
        ACALL TRAN               ; CALL TRANSMIT ROUTINE
        RETI
RETURN: CLR   EA                 ; DISABLE ALL INTERRUPTS
        MOV   TL0,#0FBH          ; INTERRUPT AFTER 8 COUNTS
        SETB  TR0                ; TURN ON COUNTER 0
        SETB  EA
        RETI
```

292019–35

```
;************** RECEIVE INTERRUPT ROUTINE *******************

REC:    CLR   TR0                ; TURN OFF COUNTER 0
        MOV   DPTR,#400H         ; EXT. RAM RECEIVE BUFFER START
        MOV   R5,#0FFH           ; EXT. RAM RECEIVE BUFFER LENGTH
        MOV   A,#18H             ; PUSH-1 STATE
WAIT1:  CJNE  A,SIUST,WAIT1      ; WAIT FOR THE CONTROL BYTE
        PUSH  RCB                ; SAVE RECEIVE CONTROL BYTE
NEXTI:  MOV   SIUST,#0EFH        ; PUSH "BYP" INTO CONTROL STATE(10H).
        MOV   A,#18H             ; PUSH-1 STATE
WAIT2:  CJNE  A,SIUST,WAIT2      ; WAIT FOR AN I-BYTE
        MOV   A,RCB              ; MOVE RECEIVED I-BYTE INTO ACC.
        MOVX  @DPTR,A            ; MOVE DATA TO EXT. RAM
        INC   DPTR               ; INCREMENT PTR TO EXTERNAL RAM
        DJNZ  R5,NEXTI           ; IS IT THE LAST I-BYTE?
        POP   RCB                ; YES, RESTORE THE CONTENTS OF RCB
        RETI                     ; RETURN

;***************** TRANSMIT SUBROUTINE **********************

TRAN:  MOV   DPTR,#200H       ; EXT. RAM TRANSMIT BUFFER START
       MOV   R5,#0FFH         ; EXT. RAM TRANSMIT BUFFER LENGTH
       MOV   A,#0A8H          ; CONTROL STATE
WAIT:  CJNE  A,SIUST,WAIT     ; WAIT  FOR CTRL BYTE XMISSION
       MOVX  A,@DPTR          ; MOVE DATA FROM EXT. RAM TO ACC.
       MOV   @R1,A            ; MOVE DATA INTO INT. RAM @ (TBS)
       INC   DPTR             ; INCREMENT POINTER
       DJNZ  R5,NXTI          ; IS IT THE LAST I-BYTE ?
       MOV   SIUST,#57H       ; NO. XMIT THE LAST I-BYTE
       RET                    ; RETURN.
NXTI:  MOV   SIUST,#57H       ; KEEP "BYP" IN CONTROL STATE(A8H).
       MOV   A,#0B0H          ; DMA-LOOP STATE
       JMP   WAIT             ; TRANSMIT THE NEXT BYTE
END
```

292019–36

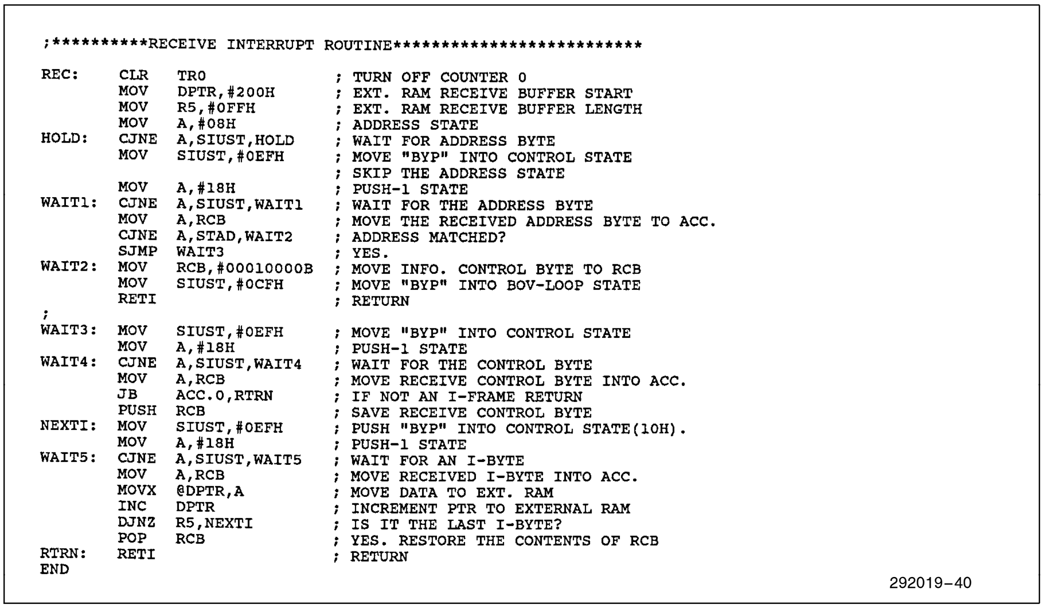


```
$DEBUG NOMOD51
$INCLUDE (REG44.PDF)

; ASSEMBLY CODE FOR SECONDARY STATIONS (MULTIPOINT)
; AUTO MODE; FCS OPTION

        ORG   00H
        SJMP  INIT
        ORG   0BH               ; VECTOR ADDRESS FOR TIMER0 INT.
        JMP   REC
        ORG   13H               ; VECTOR ADDRESS FOR EXT. INT. 1
        JMP   XINT1
        ORG   23H               ; VECTOR ADDRESS FOR SIU INTERRUPT
        JMP   SIINT


;*******************INITIALIZATION ***************************

        ORG   26H
INIT:   MOV   SMD,#11010100B  ; INT. CLKED @ 375K;NRZI=1;PFS=1
        MOV   STAD,#55H       ; STATION ADDRESS; STAD=44H FOR THE
                              ; OTHER STATION
        MOV   RBS,#10H        ; INT. RAM RECEIVE BUFFER START
        MOV   RBL,#00H        ; INT. RAM RECEIVE BUFFER LENGTH
        MOV   R1,#20H
        MOV   TBS,R1          ; INT. RAM XMIT BUFFER START
        MOV   TBL,#01H        ; INT. RAM XMIT BUFFER LENGTH
        MOV   NSNR,#00H       ; NS=NR=0
        MOV   TCON,#00000100B ; EXT. INT.: EDGE TRIGGERED
        MOV   IE,#00010110B   ; SI=1; ET0=1; EX0=1
        MOV   IP,#00000010B   ; TIMER 0: PRIORITY 1
        MOV   TMOD,#0000011B  ; COUNTER FUNCTION: MODE 3
        MOV   STS,#01000010B  ; RECEIVE I-FRAME.
        MOV   TL0,#0F8H       ; SET COUNTER TO OVERFLOW
                              ; AFTER 8 COUNTS
        SETB  TR0             ; TURN ON COUNTER
        SETB  EA              ; ENABLE ALL INTERRUPTS
DOT:    SJMP  DOT             ; WAIT FOR AN INTERRUPT.
; CPU IS INTERRUPTED  AT THE END OF RECEPTION (SI SET), AND AT*
; THE END OF LONG-FRAME TRANSMISSION (EX0 SET).              *

;***************EXTERNAL INTERRUPT ***************************

XINT1:  SETB  P1.7            ; DISABLE CTS PIN
        RETI                  ; RETURN.

;*************** SERIAL INTERRUPT ROUTINE ********************

SIINT:  CLR   SI
        JB    AM,HOP          ; ADDRESS MATCHED?
        CLR   EA              ; DISABLE ALL INTERRUPTS
        MOV   STS,#01000010B  ; RBE=1; NB=1
        MOV   TL0,#0F8H
        SETB  TR0             ; TURN ON COUNTER 0
        SETB  EA              ; ENABLE ALL INTERRUPTS
        RETI                  ; RETURN.
;
HOP:    JB    TBF,GETI        ; A FRAME TRANSMITTED?
        SETB  TBF             ; ENABLE TRANSMISSION OF I-FRAME
        CLR   P1.7            ; ENABLE CTS PIN
        ACALL TRAN            ; CALL TRANSMIT ROUTINE
GETI:   JB    RBE,RETURN      ; A FRAME RECEIVED?
        CLR   EA              ; DISABLE ALL INTERRUPTS
        SETB  RBE             ; PUT RUPI IN RECEIVE MODE
        MOV   TL0,#0F8H       ;
        SETB  TR0             ; TURN ON COUNTER 0
        SETB  EA              ; ENABLE ALL INTERRUPTS
RETURN: RETI                  ; RETURN.
;***************** TRANSMIT SUBROUTINE ***********************

TRAN:   MOV   DPTR,#200H      ; EXT. RAM TRANSMIT BUFFER START
        MOV   R5,#0FFH        ; EXT. RAM TRANSMIT BUFFER LENGTH
        MOV   A,#0A8H         ; CONTROL STATE
WAIT:   CJNE  A,SIUST,WAIT    ; WAIT  FOR CONTROL BYTE TRANSMISSION
        MOVX  A,@DPTR         ; MOVE DATA FROM EXT. RAM TO ACC.
        MOV   @R1,A           ; MOVE DATA INTO INT. RAM AT @TBS
        INC   DPTR            ; INCREMENT POINTER
        DJNZ  R5,NXTI         ; IS IT THE LAST I-BYTE ?
        MOV   SIUST,#57H      ; XMIT THE LAST I-BYTE
        RET                   ; RETURN.
NXTI:   MOV   SIUST,#57H      ; KEEP "BYP" IN CONTROL STATE
        MOV   A,#0B0H         ; DMA-LOOP STATE
        JMP   WAIT            ; TRANSMIT THE NEXT BYTE
```

292019-37

292019-38

292019-39



```
;**********RECEIVE INTERRUPT ROUTINE***************************

REC:     CLR   TR0              ; TURN OFF COUNTER 0
         MOV   DPTR,#200H       ; EXT. RAM RECEIVE BUFFER START
         MOV   R5,#0FFH         ; EXT. RAM RECEIVE BUFFER LENGTH
         MOV   A,#08H           ; ADDRESS STATE
HOLD:    CJNE  A,SIUST,HOLD     ; WAIT FOR ADDRESS BYTE
         MOV   SIUST,#0EFH      ; MOVE "BYP" INTO CONTROL STATE
                                ; SKIP THE ADDRESS STATE
         MOV   A,#18H           ; PUSH-1 STATE
WAIT1:   CJNE  A,SIUST,WAIT1    ; WAIT FOR THE ADDRESS BYTE
         MOV   A,RCB            ; MOVE THE RECEIVED ADDRESS BYTE TO ACC.
         CJNE  A,STAD,WAIT2     ; ADDRESS MATCHED?
         SJMP  WAIT3            ; YES.
WAIT2:   MOV   RCB,#00010000B   ; MOVE INFO. CONTROL BYTE TO RCB
         MOV   SIUST,#0CFH      ; MOVE "BYP" INTO BOV-LOOP STATE
         RETI                   ; RETURN
;
WAIT3:   MOV   SIUST,#0EFH      ; MOVE "BYP" INTO CONTROL STATE
         MOV   A,#18H           ; PUSH-1 STATE
WAIT4:   CJNE  A,SIUST,WAIT4    ; WAIT FOR THE CONTROL BYTE
         MOV   A,RCB            ; MOVE RECEIVE CONTROL BYTE INTO ACC.
         JB    ACC.0,RTRN       ; IF NOT AN I-FRAME RETURN
         PUSH  RCB              ; SAVE RECEIVE CONTROL BYTE
NEXTI:   MOV   SIUST,#0EFH      ; PUSH "BYP" INTO CONTROL STATE(10H).
         MOV   A,#18H           ; PUSH-1 STATE
WAIT5:   CJNE  A,SIUST,WAIT5    ; WAIT FOR AN I-BYTE
         MOV   A,RCB            ; MOVE RECEIVED I-BYTE INTO ACC.
         MOVX  @DPTR,A          ; MOVE DATA TO EXT. RAM
         INC   DPTR             ; INCREMENT PTR TO EXTERNAL RAM
         DJNZ  R5,NEXTI         ; IS IT THE LAST I-BYTE?
         POP   RCB              ; YES. RESTORE THE CONTENTS OF RCB
RTRN:    RETI                   ; RETURN
END
```

292019–40

INTEL, SUPPLY FILLER

INTEL CORPORATION, 2200 Mission College Blvd., Santa Clara, CA 95052; Tel. (408) 765-8080

INTEL CORPORATION (U.K.) Ltd., Swindon, United Kingdom; Tel. (0793) 696 000

INTEL JAPAN k.k., Ibaraki-ken; Tel. 029747-8511

Printed in U.S.A./xxxx/1296/B10M/xx xx