SERIAL COMMUNICATION CONTROLLER (SCC™): SDLC MODE OF OPERATION

Understanding the transactions which occur within a Serial Communication Controller operating in the SDLC mode simplifies working in this complex area.

INTRODUCTION

Zilog's SCC (Serial Communication Controller) is a popular USART (Universal Synchronous/Asynchronous Receiver/Transmitter) device, used for a wide range of applications. For instance, Macintosh systems use the SCC as a standard communication controller device. There are several different types of devices in the SCC family. The family consists of the Z8530 NMOS SCC, the Z85C30 CMOS SCC, the Z85230 ESCC (Enhanced SCC), Z85233 EMSCC (Mono Enhanced SCC), and Superintegration devices such as the Z181 ZIO™ and Z182 ZIP™.

Since the SCC may be used in many different ways, it may not be easy to understand all the transactions involved between the CPU and the SCC. In particular, the SDLC mode of operation is highly complicated, and many transactions are involved to make it work properly. This application note describes the sequence of events which occurs in the SDLC mode of operation.

The following sequences of events are covered:

- SDLC Transmission
- SDLC receive
  - With Receive Interrupts on all received characters or Special Conditions
  - With Receive Interrupts on First Character or Special Condition
  - With Receive Interrupts on Special Conditions only
- Receiving Back-to-back Frame under DMA control
- SDLC Loop mode

Each section explains the transmit/receive process for packets with the following characteristics:

- Initial state is mark idle
- Address field has 81H
- Control field has 42H
- Two bytes of I-field, 42H and 0FFH
- After the closing flag, mark idling

Note:
This application note describes the SCC, but not the ESCC. The ESCC, since it incorporates enhancements like deeper FIFOs and SDLC mode supporting logic, handles the packets much more simply than the SCC. Refer to the section on CMOS SCC and ESCC of this appnote for more general information on the ESCC.

SDLC TRANSMIT

Figure 1 shows the time chart for the transmitting SDLC packet under interrupt control. When transmit is engaged, data is shifted out of the transmitter on the falling edge of the transmit clock. Transitions on the diagram are shown coincident with TxCLK fall — in actual practice, there are some associated delay times (which are specified in the data sheet).

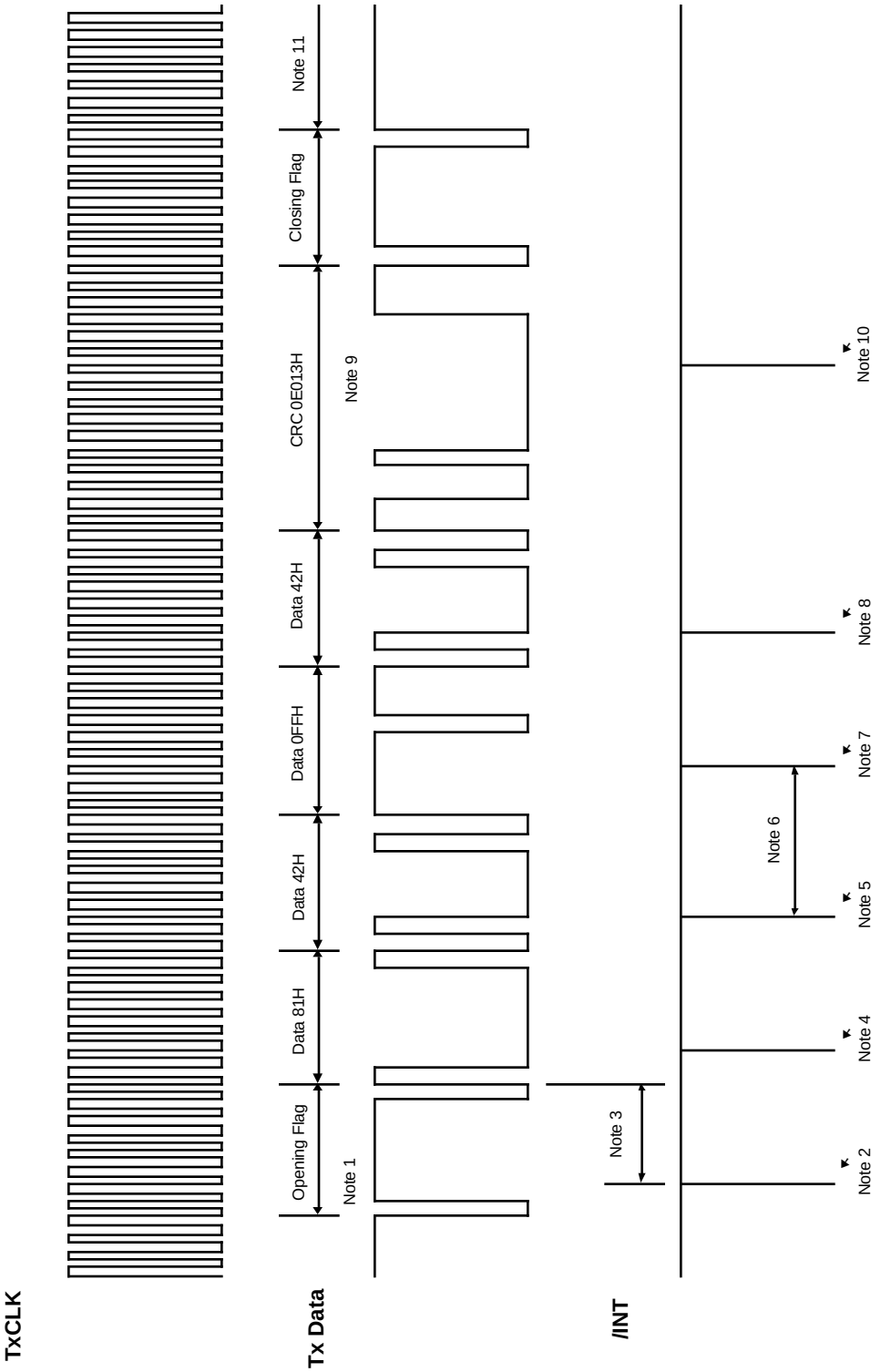

Figure 1. Typical SDLC Trsnsmission Sequence

Notes on Figure 1:

1. The SCC has two possible idle states, Mark idle (contiguous logic 1) or Flag idle (repeating flag pattern 7EH). In this figure, the SCC has to be switched to flag idle in order to send the opening flag of the frame. Care must be taken not to put the first data byte (in this case, address 81H) into the Transmit Buffer too soon after the switchover from Mark idle to Flag idle has been made; otherwise, the data may be loaded into the Transmit Shift Register before the flag is loaded. To ensure that this cannot happen, a delay must be executed before the first data byte is put into the buffer. The delay time is dependent on the data rate and a safe minimum duration is 8 bit-times.

2. Transmit Buffer Empty Interrupt for 81H. At this point the data has just been transferred to the Transmit Shift Register and data 42H is written to the Transmit Buffer.

3. The time between the first data byte being transferred to the Shift Register and the first bit appearing at the TxD pin is always six bit-times.

4. Transmit Buffer Empty Interrupt for data 42H. Data 0FFH is written to the Transmit Buffer at this point.

5. Transmit Buffer Empty Interrupt for data 0FFH. Data 42H is written to the Transmit Buffer at this point.

6. The time between interrupts depends on the data character length and the number of zero insertions in the character. For 8 bits/character it can vary between 8 and 10 bit-times. The particular instance shown corresponds to the single zero insertion when the byte 0FFH is transmitted.

7. Transmit Buffer Empty Interrupt for data 42H. Since this is the last byte to be transmitted, the Reset Transmit Interrupt Pending command is issued instead of writing another byte to the Transmit Buffer.

8. Transmitter Underrun/EOM Interrupt. This occurs when both the Transmit Shift Register and the Transmit Buffer are empty. It is an External/Status interrupt. The data sent when this occurs is summarized in the table below:

Abort/Flag on Underrun bit	Tx Underrun/EOM Latch State when Underrun occurs	Data Sent
0	Reset	CRC and Flag
1	Reset	Abort and Flag
0	Set	Flags
1	Set	Flags

9. The transmitted CRC is 16 bits long provided that there are no zero insertions. In theory it could be as long as 19 bits.

10. The last interrupt generated occurs after the CRC is shifted out of the transmitter and a flag is loaded to be sent. It is a Transmit Buffer Empty Interrupt. If another frame is to be transmitted, the first character of the next frame can be loaded. The two frames will then be separated by a single flag (Back-to-back frame).

11. If the SCC is set up for mark on idle and a new character is not loaded when the last interrupt occurs, only a single flag is sent.

SDLC RECEIVE

There are several different ways to receive a SDLC packet on the SCC; by polling, by Interrupts and by DMA. The SCC has the following four Receive Interrupt Modes:

- Disabled. This should be used in the Polling mode.
- Interrupts on all received characters or Special Conditions. This mode should be used for normal interrupt-driven operation.
- Interrupts on First Character or Special Condition. This mode is intended for received data transfer by the DMA, and enables the DMA when the interrupt is received by the First Character of the packet.
- Interrupt on Special Condition only. This mode allows the DMA to free-run and keep transferring data to the buffer. This is an ideal mode for the CMOS SCC as well as the ESCC with Status FIFO enabled, because the Status FIFO can give byte count and error status without interrupting data transfer operations.

Each of the four cases is covered in this application note except Receive Interrupt disabled. For polling, the basic operation is identical to that used for “interrupt on all characters or Special Condition” mode. Instead of waiting for an interrupt, polling Reads Registers to determine if service is needed or not.

On the SCC, data is sampled by the receiver on the rising edge of the receive clock. Set-up and hold times for RxD with respect to RxC are specified in the product specifications.

In general, receiver status changes are triggered by RxC. In the following Figures, they are shown as being coincident with this edge — in actual practice, there are some associated delay times (which are specified in the data sheet).

RECEIVE INTERRUPTS ON ALL RECEIVE CHARACTERS OR SPECIAL CONDITIONS

SCC is placed in this mode by programming Bit D4-3 of WR1 to 10. Once programmed in this mode, the SCC generates interrupts whenever character(s) are in the receive buffer or when Special Conditions occur. This mode is the most common operational mode.

Notes on Figure 2:

1. The receiver is usually in hunt mode when waiting for a frame. When the opening flag is received, an External/Status Interrupt is generated, indicating the change from hunt mode to sync mode.
2. The /SYNC output follows the state of the sync register comparison output. The comparison is done on a bit by bit basis, so the /SYNC pin is only active for one bit-time. /SYNC goes active one bit-time after the last bit of the sync character is sampled at the RxD pin.
3. A Receive Character Available Interrupt is generated 11 bit-times 8 bits for the shifter and a 3-bit delay) after the last bit of the character is sampled at the RxD pin. The status bits corresponding to that character must be read before the data character is read from the Receive Buffer. This interrupt is for data 81H.
4. Receive Character Available Interrupt for data 42H.
5. Receive Character Available Interrupt for data 0FFH.
6. Receive Character Available Interrupt for data 42H.
7. Receive Character Available Interrupt for the first CRC byte. The SCC treats the CRC as data, since the SCC does not yet distinguish a difference between CRC and data!
8. The closing flag is recognized two bit-times before the second CRC byte is completely assembled in the Receive Shift Register. As soon as it is recognized, a Special Condition interrupt is generated. The EOF bit is set at this point and the CRC error bit can be checked. The six least significant bits of the second CRC byte are present at the top of the first CRC byte. The status information must be read before the second CRC byte is read from the buffer. The CRC bytes should be discarded. The CRC checker is automatically reset for the next frame.
9. External/Status interrupt for the Sync/Hunt change. This occurs when the SCC recognizes an Abort (Marking line) and forces the receiver into hunt mode. The SCC can be programmed so that the Abort itself generates an interrupt if required. If flag idle was set, this interrupt will not occur.

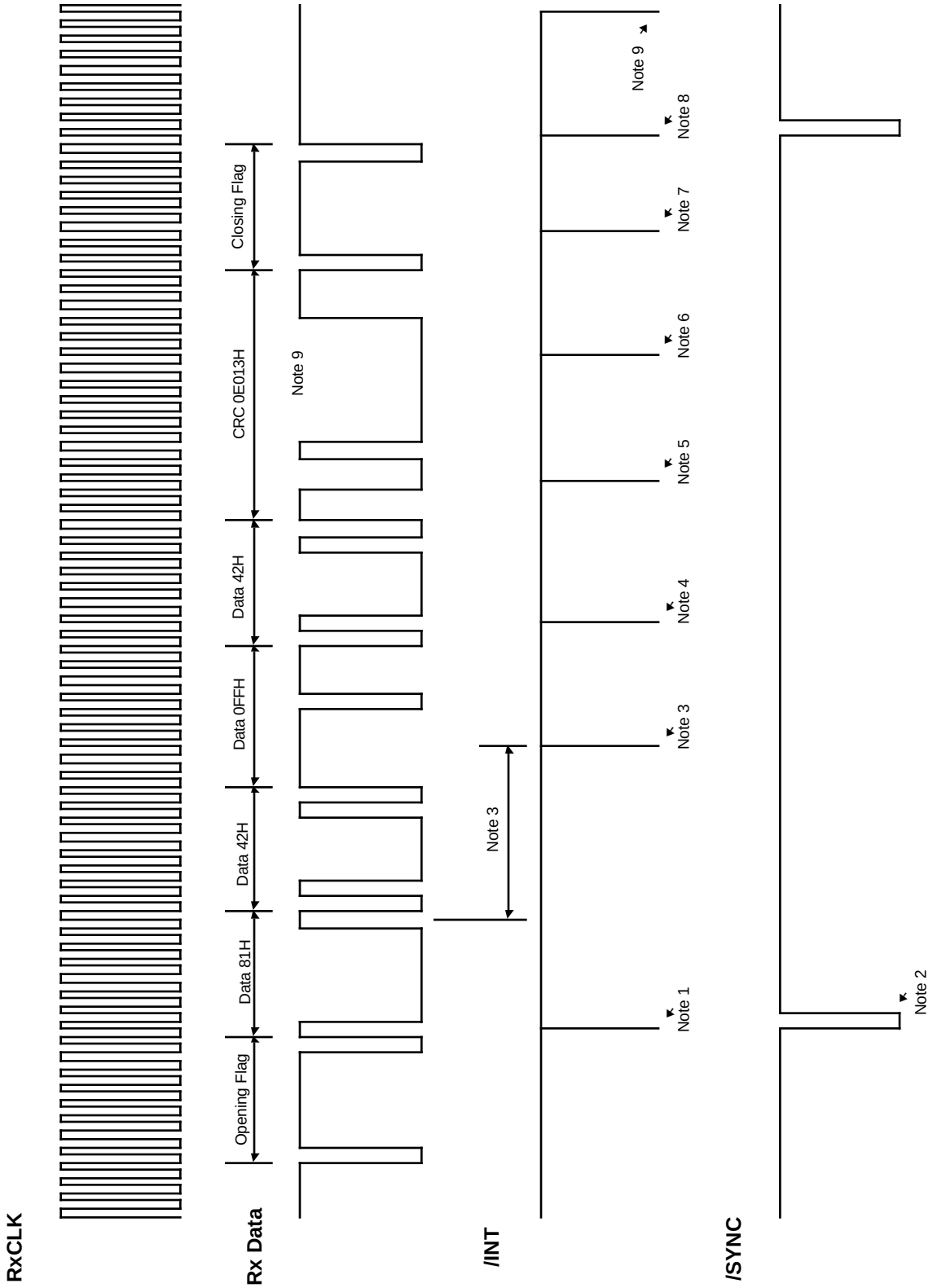

Figure 2. Typical SDLC Receive Sequence with Receive Interrupts on all Received Characters or Special Condition

RECEIVE INTERRUPTS ON FIRST CHARACTER OR SPECIAL CONDITIONS

The sequence of events in this mode is similar to that in "Receive Interrupts on all received characters and Special Conditions", except that it generates Receive Character Interrupt on the first received character only, and subsequent data is read by the DMA.

The SCC is placed in this mode by programming Bit D4-3 of WR1 to 01. Once programmed in this mode, the SCC generates interrupts when it receives the First Character of the packet or a Special Condition occurs. This mode is for operation with the DMA. On the interrupt for the first received character, DMA is enabled. On Special Conditions (either End-of-Message, overrun, or Parity error, — parity on the SDLC is not normal, however), the service routine stops the DMA and starts over again.

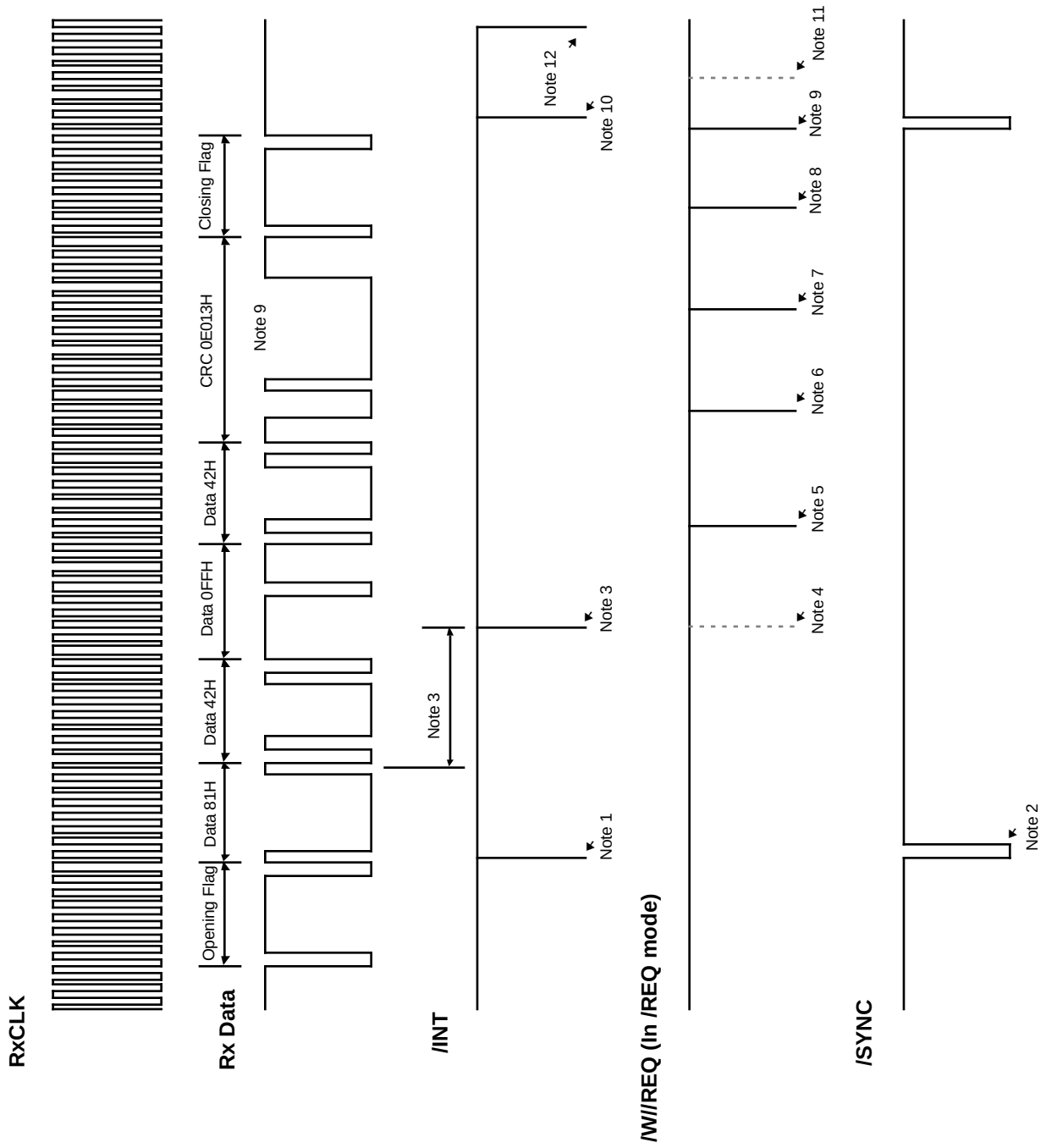

Figure 3. Typical SDLC Receive Sequence with Receive Interrupts on First Character or Special Condition

Notes on Figure 3:

1. The receiver is usually in hunt mode when waiting for a frame. When the opening flag is received, an External/Status Interrupt is generated, indicating the change from hunt mode to sync mode.

2. The /SYNC output follows the state of the sync register comparison output. The comparison is done on a bit by bit basis, so the /SYNC pin is only active for one bit-time. /SYNC goes active one bit-time after the last bit of the sync character is sampled at the RxD pin.

3. A Receive Character Available Interrupt is generated 11 bit-times after the last bit of the character is sampled at the RxD pin. In this mode, enable the DMA on this interrupt. This interrupt is for data 81H.

4. If SCC's DMA request function has been enabled, /REQ becomes active here.

5. DMA request for data 42H.

6. DMA request for data 0FFH.

7. DMA request for data 42H.

8. DMA request for the first CRC byte. The SCC treats the CRC as data, since the SCC does not yet distinguish a difference between CRC and data!

9. DMA request for the second CRC byte. The closing flag is recognized two bit-times before the second CRC byte is completely assembled in the Receive Shift Register. As soon as it is transferred to the Receive Buffer, it generates a DMA request.

10. This interrupt is EOF (End of Frame), a Special Condition interrupt. This will not occur until the DMA has read the 2nd CRC byte from the Receive Buffer. When it occurs, the Receive Buffer is locked and no more DMA requests can be generated until the Receive Buffer is unlocked by issuing the Error Reset command. Before issuing this command, all of the status bits required (e.g., the CRC error status) must be read, and the last two bytes read by the DMA discarded. The enable interrupt on next Receive Character command must be sent to the SCC so that the next character (i.e., the First Character of the next frame) will produce an interrupt. If this is not done, the character will generate a DMA request, not an interrupt.

    Should a Special Condition occur within the data stream (i.e., for a condition other than EOF) the /INT pin will not go active until the character with the Special Condition has been read by the DMA.

11. DMA request for 2nd CRC bytes. This occurs when the EOF interrupt service routine has not disabled the DMA function of the SCC, and did not read the data after unlocking the buffer by issuing an Error Reset command.

12. External/Status Interrupt for the Sync/Hunt change. This occurs when the SCC recognizes an Abort (Marking line) and forces the receiver into hunt mode. The SCC can be programmed so that the Abort itself generates an interrupt if required. If flag idle was set, this interrupt would not occur.

RECEIVE INTERRUPTS ON SPECIAL CONDITIONS ONLY

The sequence of event in this mode is similar to that for "Receive Interrupts on first received character or Special Condition," except it will not generate Receive Character Available interrupt at all. This mode is designed for operations where the DMA is pre-programmed, or the application does not have enough time to set up DMA transfer on First Character interrupt.

The SCC is placed in this mode by programming Bit D4-3 of WR1 to 11. Once programmed in this mode, the SCC generates interrupts when Special Conditions occur. On Special Condition (either End-Of-Message or overrun/Parity error, if enabled), corrective action can be taken for that packet.

The SDLC Frame Status Buffer (not available on the NMOS version) is very useful in this mode. First of all, set DMA to transfer several packets. The SDLC Frame Status Buffer holds information which tells you how many bytes were in the received packet and reports whether or not error conditions (overrun/CRC error/parity error) have occurred.

The sequence of events in this mode is identical to the "Receive Interrupts on First Character or Special Condition" mode (Figure 3); Note 3, however, does not apply, and Note 4 should read as follows for this case:

Note 4 in Receive Interrupts on Special Condition only mode:
DMA request for data 81H. The DMA function of the SCC should be enabled by this time frame.

RECEIVING BACK TO BACK FRAME IN RECEIVE INTERRUPTS ON SPECIAL CONDITION ONLY MODE

"Back to Back" frame means there are two frames separated with only one flag — the closing flag of the previous packet also acts as the opening flag of the following packet. Receiving such packets is identical to receiving a single packet, except that the sequence of events happens in a short time around the shared flag.

Assuming SCC is running under Receive Interrupts on Special Condition only mode (under DMA Control), a typical sequence of events is shown in Figure 4. It is identical to that used for "Receive Interrupts on Special Condition Only" mode, with the addition of another following packet.

Notes on Figure 4:

1. DMA request data before 0FFH.

2. DMA request for data 0FFH.

3. DMA request for data 42H.

4. DMA request for the first CRC byte. The SCC treats the CRC as data, since the SCC does not yet distinguish a difference between CRC and data!

5. DMA request for the second CRC byte. The closing flag is recognized two bit-times before the second CRC byte is completely assembled in the Receive Shift Register. As soon as it is transferred to the Receive Buffer, it generates a DMA request.

6. This interrupt is EOF (End of Frame), a Special Condition Interrupt. This will not occur until the DMA has read the 2nd CRC byte from the Receive Buffer. When it occurs the Receive Buffer is locked and no more DMA requests can be generated until the Receive Buffer is unlocked by issuing the Error Reset command. Before this command is issued, all of the status bits required (e.g., the CRC error status) must be read, and the last two bytes read by the DMA discarded. The Enable Interrupt on Next Receive Character command must be sent to the SCC so that the next character (i.e., the First Character of the next frame) will produce an interrupt. If this is not done, the character will generate a DMA request, not an interrupt.

   On unlocking the Receive Buffer after the EOF interrupt, no initialization is required with respect to the receiver. All characters have been removed by the DMA and the receiver is ready for the next frame. While the Buffer is locked the SCC can receive 2 7/8 characters (8 bits/character) before there is a danger of the receiver overrunning. The only way that this can be specified is by referencing it to the falling edge of the request for the last CRC byte. This time is a worst case minimum of 33 bit-times (possibly more if there are any characters with inserted zeros). As soon as the Buffer is unlocked an additional 8 (minimum) bit-times become available because the top byte of the Buffer is freed up.

7. DMA request for second CRC byte. This occurs when the EOF interrupt service routine has not disabled the DMA function of the SCC, and fails to read the data after unlocking the FIFO by issuing Error Reset command.

8. DMA request for data 01H.

9. MA request for data 03H.

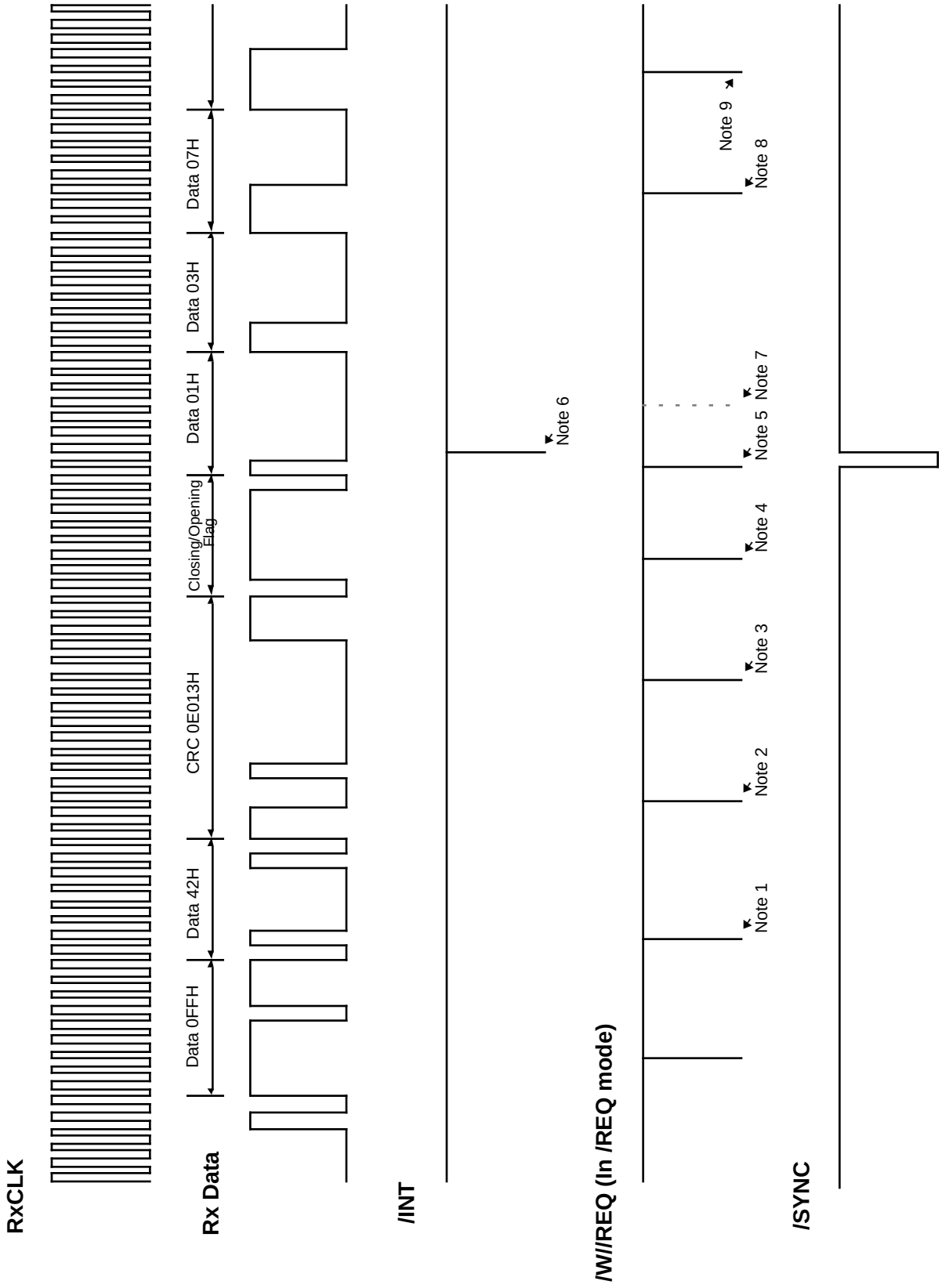

Figure 4. Receiving "Back to Back" frame with Receive Interrupts on Special Condition Only Mode (DMA Controlled)

THE SDLC LOOP MODE

The SDLC Loop mode is one of the protocols used in the ring configuration network topology. The typical network configuration is shown in Figure 5. As shown, there is one Master (or primary) station and several slave (or secondary) stations. This figure does not have a clock connection, but each station's transmit clock must be synchronized to the master SCC. This can be done by feeding the clock using a separate clock line, or by using Phase Locked Loop (PLL) to recover the clock.

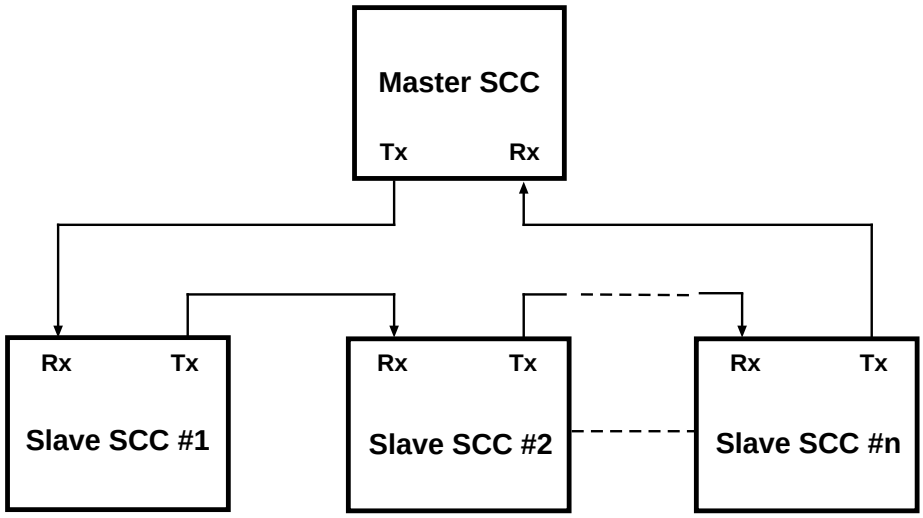

Figure 5. SDLC Loop Mode Configuration

This mode is similar to the normal SDLC mode, other than that secondary stations are not allowed to freely send out packets. When a secondary station wants to send a packet, it needs to wait for a special pattern to be received. The pattern is called EOP (End Of Poll), and consists of a 0 followed by seven 1s on the transmission line (as data, it is 11111110). This pattern resembles the SDLC Flag pattern (7EH; 0111110), except the last bit has been changed to a 1 thus turning this pattern into a flag.

Upon network initialization, secondary station TxD and RxD connections use gate propagation delay. On the first EOP, a secondary station inserts one bit -time delay between RxD and TxD, and relays RxD input to TxD. When it has a message to send out, it waits for an EOP. When it detects EOP in this phase, it changes the last bit of the EOP to zero, making it a Flag, then begins to send its own message. From this point on, normal SDLC transmission modes apply. Packets conclude with Mark idle, identifying it as an EOP pattern. The secondary station then reverts back to one bit delay mode.

Figure 6 illustrates this mode's sequence of events. To simplify the example, this figure assumes there is one Master station and one Slave station. If there are more Slave stations, there will be additional one bit time delay per station after the network has initialized for loop mode of operation.

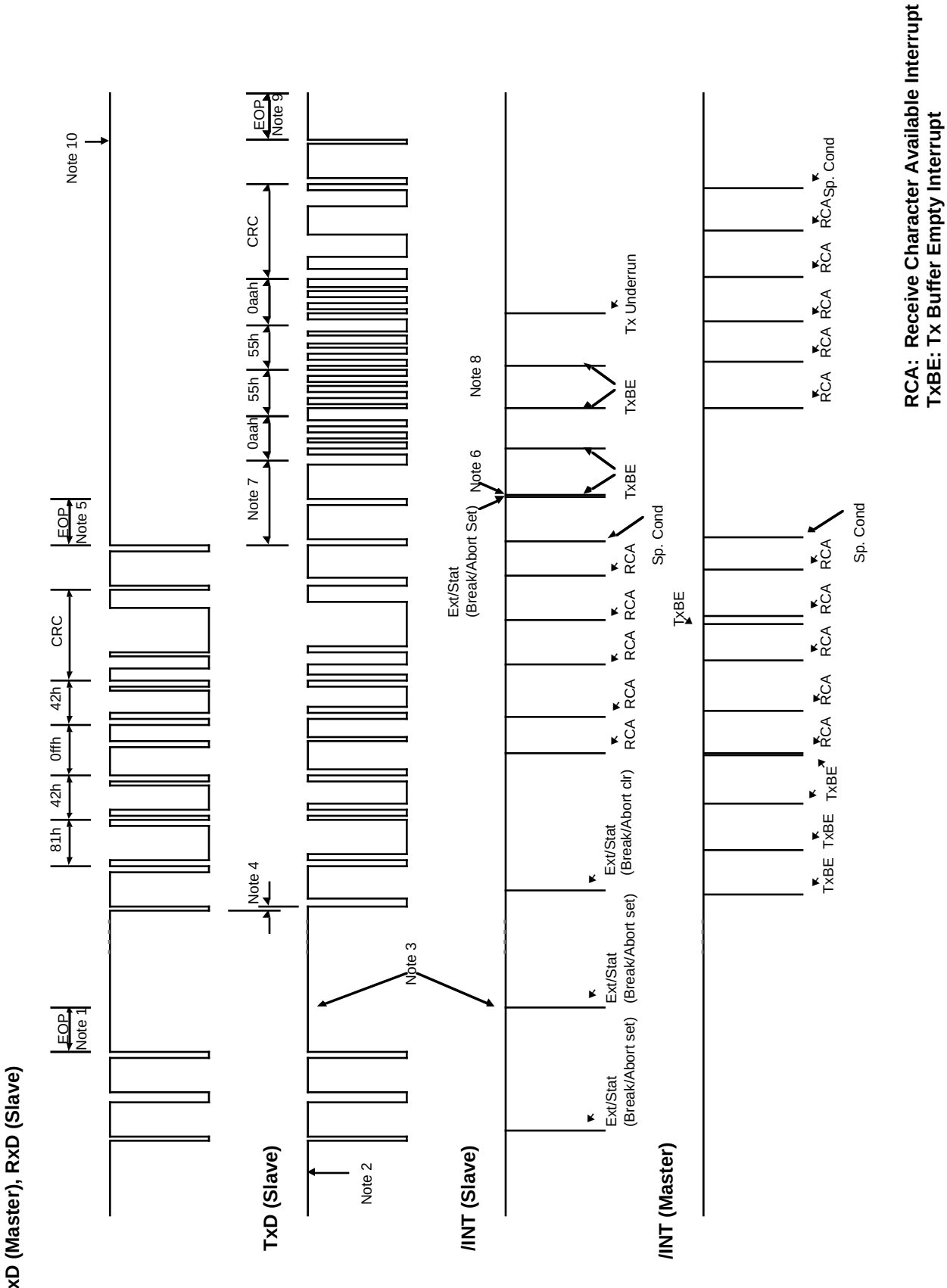

Figure 6. SDLC Loop Mode

THE SDLC LOOP MODE (Continued)

Notes on Figure 6:

1. The master SCC sends EOP by switching from flag on idle to mark on idle
2. At initialization, all Slave stations were set up for SDLC loop mode At this point, the Slave station connects its RxD pin to TxD pin with gate propagation delay, and starts to monitor Rx data for the EOP sequence.
3. On receiving the EOP, the slave generates an External/Status Interrupt with Break/Abort bit set. A one bit time delay is inserted between RxD and TxD. (The GAOP,Go active on Poll, bit should be reset at this point to avoid unexpected loop entry by the Slave transmitter.) The Slave's on-loop bit is set and the receiver is in hunt mode.
4. Note that there is a one bit time delay between received data and transmitted data.
5. When the Slave wants to transmit it must first receive an EOP and have GAOP set.
6. On receiving an EOP, the Slave interrupts with Break/Abort clear. The EOP is converted to a flag, the loop sending bit is set, and the transmitter will send flags until data is written into the Transmit Buffer.
7. Note that the flags overlap.
8. When the slave has sent all of its data the GAOP flag should be cleared so that the CRC is sent on underrun.
9. When the closing flag has been sent the Slave reverts to a one bit delay, which produces another EOP.
10. The master must keep its output marking until its receiver has received all frames sent by secondaries.

CMOS SCC AND ESCC

The discussion above applies to the NMOS SCC and the CMOS SCC without the SDLC Frame Status FIFO feature. The CMOS version and the ESCC have a SDLC Frame Status FIFO for easier handling of the SDLC mode of operation. The SDLC Status FIFO is designed for DMA controlled SDLC receive for high speed SDLC data transmission, or for systems whose CPU interrupt processing is not fast.

This FIFO is able to store up to 10 packets' worth of byte count (14-bit count) and status information (Overrun/Parity/CRC error status). To use this feature, simply enable this FIFO and let DMA transfer data to memory. While DMA is transferring received data to the memory, the CPU will check the FIFO and locate the data in memory, as well as the status information of the received packet.

Other ESCC enhancements make it easier to handle the SDLC mode of operation. These include:

- Deeper FIFO (4 Bytes Transmit, and 8 Bytes receive)
- Automatic Opening Flag transmission
- Automatic EOM reset
- Automatic /RTS deactivation
- Fast /DTR//REQ mode
- Complete CRC reception
- Receive FIFO Antilock feature
- Programmable DMA and interrupt request level
- Improved data setup time specification

For more details on these functions, please refer to the SCC/ESCC Technical manual and related documents.

CONCLUSION

This application note describes the basic operation of the SCC in SDLC operational modes. With minor variations, most of these operations also apply to the CMOS SCC with Status FIFO enabled and the ESCC.