ZILOG SCC
Z8030/Z8530
QUESTIONS AND ANSWERS

This document addresses the most commonly asked questions about Zilog's SCC. These questions fall into the following five categories:

- Hard ware Considerations
- Interrupt s and Polling
- Asychronous mode

- Sychronous Mode
- Miscellaneous Questions

HARDWARE CONSIDERATIONS

This section includes questions and answers on the hardware interface, the clocks, the FIFO, special modes (Local Loopback, DPLL, Manchester), and internal timing consideration.

Hardware (Includes DMA Interface)

Q. What is the SCC transistor count?
A. Approximately 6000 gates, or 18,000 transistors.

Q. What is the difference between the Z8030 and the Z8530?
A. The Z8030 and Z8530 are packaged from the same die. The multiplexed bus (Z8030) or non-multiplexed bus (Z8530) version of the chip is selected at packaging time by an internal bonding option.

Q. Can /AS be active only when the Z8030 is being accessed and High all other times?
A. Since the interrupt pending bits (IPs) are updated on address strobes, interrupts will not occur unless /AS is continuous.

Q. How do /WR and /CE interact on the Z8530?
A. /WR and /CE are ANDed to enable a transparent latch. Data is latched on the falling edge when both /CE and /WR go Low.

Q. How many register pointers does the Z8530 have?
A. The SCC has only one register pointer for both channels. The SIO (Z844X) has two, one for each channel.

Q. Do you have to write to the pointer with the Z8530 to access WR0 or RR0?
A. No. Both registers are accessed automatically without first writing to the pointer.

Q. Does /CE (/CS) have to be High during an interrupt acknowledge cycle?
A. No.

Q. Does the SCC support full duplex DMA?
A. The SCC allows full duplex DMA transfers by using the DTR/REQ and W/REQ as two separate DMA control lines for transmit request and receive request on each channel.

Q. When using full duplex DMA, how do you program W/REQ?
A. W/REQ should be programmed for receive and DTR/REQ pin should be programmed for transmit.

Q. Can both channels make simultaneous DMA requests?
A. Yes.

Q. Do you have to reset the SCC in hardware?
A. No. A software reset is the same as a hardware reset, (WR9 CO). It also does not matter whether the Z8030 is in shift right or shift left mode because the address is the same in either.

HARDWARE CONSIDERATIONS (Continued)

Q. Do you need to clear the reset bit in WR0 after a software reset?

A. The reset is clocked with PCLK; so it must be active during reset.

Q. How long after a hardware reset should you wait before programming the SCC.

A. Four PCLKs.

Q. Why does the SCC initialization require that the External Status Interrupts be reset twice?

A. Because of the possibility of noise causing an interrupt pending bit (IP) to be set. The second reset guarantees that the latch is clear. If the latch is closed high and the external signal is low, the first reset will open the latch at the high-to-low transition causing an interrupt.

Clocks

Q. Does PCLK have to have a 50% duty cycle?

A. The duty cycle doesn't have to be 50% as long as the minimum specification is met.

Q. Can the SCC PCLK be stretched?

A. Yes, as long as the pertinent specification is met. However, this could cause a problem if PCLK is used to generate the bit rate.

Q. The bit rate generator is driven from what sources?

A. It may be driven from the RTxC pin or PCLK, or from a crystal.

Q. How do you connect a bit rate crystal to the SCC?

A. A crystal can be connected between RTxC and SYNC to supply the clock if the SCC is programmed for WR11 D7-1.

Q. What is the crystal specification?

A. It is a fundamental, parallel resonant crystal. For further details see the "Design Considerations Using Quartz Crystals with Zilog's Components" Application Note.

Q. Can RTxC on both channels be driven from the same crystal.

A. No. A separate crystal should be used for each channel. The crystal should be connected between /SYNC and RTxC of the respective channels. The alternate solution may be to use crystal on one channel and reflect the clock out of the TRxC output and feed it into another channel.

Q. How do you select a crystal frequency?

A. Time constant: (Clock Frequency/2 x Bit rate x clock factor) - 2. Two examples are given below:

For PCLK = 3.6864 MHz			For PCLK = 3.9936 MHz		
Bit Rate	**TC**	**Error**	**Bit Rate**	**TC**	**Error**
38400	46	-	19200	102	-
19200	94	-	9600	206	-
9600	190	-	7200	275	12%
7200	254	-	4800	414	-
4800	382	-	3600	553	.06%
3600	510	-	2400	830	-
2400	766	-	2000	996	.04%
1200	1534	-	1800	1107	.03%
			1200	1662	-
			600	3326	-
			300	6654	-
			150	13310	-
			134.5	14844	.0007%
			110	18151	.0015%
			75	26622	-
			50	39934	-

Q. Why are there different Clock factors?

A. These clock factors enable the SCC to sample the center of the data cell. In the 16x mode, the SCC divides the bit cell into 16 counts and samples on count 8. Clock factors are generally only used with Asynchronous modes.

Q. How is the error in the receive/transmit clock reduced?

A. The ideal way to reduce this error is by adjusting the crystal frequency such that only an integer value of TC is yielded when the equation is used.

Q. What are the maximum transfer rates?

A. The following table shows the PCLK rates (in bps).

	4 MHz	6 MHz	8 MHz	10 MHz	16 MHz	20 MHz
Asynchronous mode:						
External clock						
6x mode (no BRG)	250K	375K	500K	635K	1M	1.25M
BRG						
16x mode (TX + 0)	62.5K	93.75K	125K	156.5K	250K	312.5K
Synchronous mode:						
Using external clock	1M	1.5M	2M	2.5M	4M	5M
Using DPLL, FM encoding	250K	375K	500K	625K	1M	1.25M
Using DPLL, MRZ/NRZI encoding	125K	187.5K	250K	312.5K	500K	625K
Using DPLL, FM, BRG	62.5K	93.75K	125K	156.25K	250K	312.5K
Using DPLL, NRZ/NRZI, BRG	32.25K	46.88K	62.5K	78.125K	125K	156.25K

Q. Can the maximum transfer rate using an external clock be achieved?

A. Yes, but it is not trivial. In order to achieve the maximum rate on transmit, the SCC should have a dedicated processor or DMA. For example, at a 1 MHz rate, a byte must be loaded into the SCC every 8 microseconds. To achieve the maximum rate on receive, requires that the receive clock and the SCC PCLK be synchronized. (RTxC to PCLK setup time at maximum rate in the Product Specification.) It is probably easier to use a slightly faster PCLK SCC, or back off slightly from the maximum rate.

FIFO

Q. How do you avoid an overrun in the received FIFO?

A. The receive buffer must be read before the recently received data character on the serial input is shifted into the receive data FIFO. This FIFO is three bytes deep. Thus, if the buffer is not read, the fifth character just arrived causes an overrun condition. There is no bit that can be set or reset to disable the buffering.

Q. What happens when you read an empty FIFO?

A. You read the last character in the buffer.

Q. When the FIFO gets locked due to an error condition, can it still receive?

A. The SCC continues to receive until an overrun occurs.

Q. Assuming that there are characters available in the FIFO, what happens to them if the receiver goes into the hunt mode?

A. They will remain in the FIFO until they are either read by the CPU or DMA, or until the channel is reset.

SPECIAL MODES
(LOCAL, LOOPBACK, DPLL, MANCHESTER)

Q *How are the Local, Loopback, and Auto Echo modes implemented?*

A. The TxD and RxD pins are connected through drivers. If both modes are simultaneously enabled, then Auto Echo overrides.

Q. Can the SCC transmit when the Auto Echo mode is enabled?

A. No, the transmitter is logically disconnected from the TxD pin.

Q. Can the Digital Phase Lock Loop (DPLL) be used with NRZ?

A. The DPLL simply generates the receive clock which is the same for both NRZ and NRZI.

Q. Do you have to use the DPLL with NRZI and FM encoding?

A. If the DPLL is not used, a properly phased external clock must be supplied.

Q. What is the error tolerance for the DPLL?

A. The DPLL can only tolerate a + or - 1/32 deviation in frequency, or about 3%.

Q. Can you receive and transmit between two channels on the same SCC using the DPLL to generate both the transmit and receive clocks?

A. To transmit and receive using the same clock, you need to divide the transmit clock by 16 or 32 to be the same rate for transmitting and receiving, because the DPLL requires a divide-by-16 or -32 on the receiver, depending on the encoding. An external divide-by-16 or -32 is required, and can be connected by outpouring the bit rate generator on the /TRxC pin, through the external divide circuit, and back in the /RTxC pin as an input to the transmitter.

Q. How fast will Manchester be decoded?

A. The SCC can decode Manchester data by using the DPLL in the FM mode and programming the receiver for NRZ data. Hence, the 125K bit/s is the maximum rate for decoding at 8MHz SCC. A circuit for encoding Manchester is available from Zilog.

Q. When will the Time Constant be loaded into the BRG counter?

A. After a S/W reset or a Zero Count is reached.

Q. How to run NRZ data using the DPLL?

A. Use NRZI for DPLL (WR14) but set to NRZ (WR10).

INTERNAL TIMING

Q. When does data transfer from the transmit buffer to the shift register?

A. About 3 PCLK's after the last bit is shifted out.

Q. How long does it take for a write operation to get to the transmit buffer?

A. It takes about 5 PCLK's for the data to get to the buffer.

Q. What is Valid Access Recovery Time?

A. Since WR/ and RD/ (AS/ and DS/ on the Z8030) have no phase relationship with PCLK, the circuitry generating these internal control signals must provide time for metastable conditions to disappear. This gives rise to a recovery time related to PCLK.

Q. How long is Valid Access Recovery Time?

A. On the NMOS SCC, the recovery time is 4 PCLK's, while on the CMOS SCC, the recovery time is 3-3.5 PCLK's.

Q. Why does the Z8030 require that the PCLK be "at least 90% of the CPU clock frequency for Z8000?"

A. If the clocks are within 90%, then the setup and hold times will be met. Otherwise, the setup and hold times must be met by the user.

Q. Does Valid Access Recovery Time apply to all successive accesses to the SCC?

A. Any access to the SCC requires that the recovery time be observed before a new access. This includes reading several bytes from the receive FIFO, accessing separate bytes on two different channels, etc. When using DMA or block transfer methods, the recovery time must be considered.

Q. Do the DMA request and wait lines on the SCC take the Valid Access Recovery time into account before they make a request?

A. No, they are not that intelligent. The user must take this into account, and program the DMA accordingly. For example, by inserting wait states during the memory access between SCC accesses, which will lengthen the time in between SCC accesses, or by requiring the DMA to release the bus between accesses to the SCC, to prevent simultaneous data requests from two channels from violating the recovery time.

Q. What happens if Valid Access Recovery Time is violated?

A. Invalid data can result.

Q. Does Valid Access Recovery Time affect the interrupt acknowledge cycle?

A. No. The interrupt vector is put on the bus by the SCC during the interrupt acknowledge cycle, but does not require any recovery time.

Q. Why can some systems violate the recovery time by 1 or 2 PCLK's without affecting the data to the SCC?

A. This violation may or may not matter to the SCC. This phase relationship between PCLK, /RD, /WR, (/AS, /DS for Z8030) can by ASYNC. The SCC requires some time internally to synchronize these signals. The electrical specs for the SCC indicate a recovery time, which is the worst case maximum.

INTERRUPT CONSIDERATIONS

Q. What conditions must exist for the SCC to generate an interrupt request?

A. Interrupts must be enabled (MIE = 1 and IE = 1). The Interrupt Enable Input (IEI) must be high. The interrupt pending bit (IP) must be set and its interrupt under service bit (IUS) must be reset. No interrupt acknowledge cycle may be active.

Q. How can the /INTACK signal be synchronized with PCLK?

A. /INTACK needs to be synchronized with PCLK. This can be accomplished by changing /INTACK only on the falling edge of PCLK by using a D flip-flop that is clocked with the inverted PCLK.

Q. Is /CE required during an Interrupt Acknowledge cycle?

A. No.

Q. How long does /INT stay active low when requesting an interrupt?

A. If the SCC is operated in a polled mode, the /INT will remain active until the IP bit is reset. For an interrupt acknowledge cycle, the /INT will go inactive shortly after the falling edge of /RD or /DS when the IUS bit is set.

Q. Can you use the SCC without a hardware interrupt acknowledge?

A. Yes. If you are not using the hardware daisy chain, you don't need to give an interrupt acknowledge. Tie the intack pin high, enable interrupts, and on responding to an interrupt, check RR3 for the cause, and special receive conditions if you are in receive mode. The internal daisy-chain settling time must still be met. (IEI to IEO delay time specification.)

Q. How do you acknowledge an interrupt without a hardware interrupt acknowledge?

A. Reset the responsible interrupt pending bit (IP). The /INT line follows the IP bit.

Q. When are the IP bits cleared?

A. A transmitter empty IP is cleared by writing to the data register. A receive character available IP is cleared by reading the data register. The external/status interrupt IP is cleared by the command Reset Ext/Status Interrupts.

Q. Can the IP bits be set while the SCC is servicing other interrupts?

A. Yes. If the interrupting condition has a higher priority than the interrupt currently being serviced, it causes another interrupt, thus nesting the interrupt services.

Q. Can the IUS bits be accessed?

A. No. They are not accessible.

Q. When do IUS bits get set?

A. The IUS bits are set during an interrupt acknowledge cycle on the falling edge or /RD or /DS.

Q. How do you reset interrupts on the SCC?

A. The interrupt under service bit (IUS) can be reset by the command "Reset Highest IUS" or 38 Hex to WR0. Reset Highest IUS should be the last command issued in the interrupt service routine.

Q. Why is the interrupt daisy chain settle time required?

A. This mechanism allows the peripheral with the highest priority interrupt pending in the hardware interrupt daisy chain to have its interrupt serviced.

Q. Is there still a settle time if the peripherals are not chained?

A. Even if only one SCC is used, there still is a minimum daisy-chain settle time due to the internal chain.

Q. How should the vectors be read when utilizing the /INTACK?

A. /INTACK should be tied to 5 volts through a register. Erroneous reads can result from a floating INTACK. The interrupt vectors can be read after an interrupt from RR2.

Q. How is the vector register different from the other registers?

A. The vector register is shared between both channels. The Write register can be accessed from either channel. Reading "Read Register 2" on Channel A (RR2A) returns the unmodified vector, and RR2B returns the

INTERRUPT CONSIDERATIONS (Continued)

modified vector that includes status. The vector includes the status bit (VIS, WR9) and determines which vector register is put out on the bus during an interrupt cycle.

Q. How do you poll the external/status interupt IP bit?
A. Set the IE bits in WR15 so the conditions are latched and set ext/status master interrupt enable bit in WR1. To guarantee the current status, the processor should issue a Reset External/Status interrupts command in WR0 to open the latches before reading the register. For further details see the SCC Technical Manual, section 3.4.7.

Q. When should the status in RR1 be checked?
A. Always read RR1 before reading the data.

Q. What conditions cause the transmit IP to be set?
A. Either the buffer is empty, or the flag after CRC is being loaded.

Q. How do you tell if you have a Zero Count (ZC) interrupt?
A. This bit is not latched like the other external IP bits. If an external interrupt occurs and none of the other IP bits have changed since the last ext/status interrupt, then the ZC condition caused it. A ZC interrupt will not be generated if there are other ext/status (IP) pending. The ZC stays active for each time only when the count reached zero, approximately two PCLK time periods.

Q. How do you poll the bits in RR3A?
A. Enable interrupts in WR1 and disable MIE before polling.

Q. What happens when the SCC is programmed to interrupt on transmit buffer empty and also to request DMA activity on transmit buffer empty?
A. This would not be a wise thing to do. The interrupt would occur but the DMA could gain control of the bus and remove the interrupting condition before the interrupt acknowledge could take place. When the CPU recovers control of the bus and starts the interrupt acknowledge cycle, bus confusion results because the peripheral no longer has a reason to interrupt.

Q. Will IP bit (s) for external status be cleared by the Reset Ext/Status Interrupt?
A. Yes.

ASYNCHRONOUS MODE

Q. *Can the Sync Character Load Inhibit function strip characters in Asynchronous mode if not disabled?*

A. Yes. If not disabled it will strip any characters which match the value in the sync character register. Always disable this function in asynchronous mode (WR3, bit D1).

Q. *What controls the DTR/WREQ pin?*

A. The DTR pin follows the D7 bit in WR5 (inverse) as a Data Terminal Ready pin, or it is a DMA request line (WREQ). The bit can be set or reset by writing to WR5.

Q. *How is the Asynchronous mode selected?*

A. The Asyn mode is selected by programming the number of stop bits in write register 4.

Q. *How are receiver breaks handled?*

A. The SCC should monitor the break condition and wait for it to terminate. When the break condition stops, the single NULL character in the receive buffer should be read and discarded.

Q. *Where can you get the DTR input if the DTR/REQ pin is being used for DMA?*

A. The SYNC can be used as an input if operating in the Async mode. It will cause an interrupt on both transitions.

Q. *When a special condition occurs due to a parity error, will a receive interrupt for that byte still be generated?*

A. No. In the case of Receive interrupt on Special Condition Only mode, the interrupt will not occur until after the character with the special condition is read. In the case of Receive Interrupt on All Characters or Special Condition Only mode, the interrupt is generated on every character whether or not it has a special condition.

Q. *In the Auto Enable mode, what happens when CTS/ goes inactive (high) in the middle of transferring a byte?*

A. If the Auto Enable mode is selected, the CTS/ pin is an enable for the transmitter. So, when CTS/ is inactive, transmit stops immediately.

Q. *Can X1 clock mode really be used for the Async operation?*

A. X1 mode cannot be used unless the receive and transmit clocks are synchronized. Using a synchronous modem is one way of satisfying this requirement.

Q. *When does the FIFO buffer lock on an error condition?*

A. The receive data FIFO gets locked only in cases where the following receiver interrupt modes are selected:

– Receive Interrupt on Special Condition only

– Receive Interrupt on First Character or Special Condition

In both of these modes, the Special Condition interrupt occurs after the character with the special condition has been read. The error status has to be valid when read in the service routine. The Special Condition locks the FIFO and guarantees that the DMA will not transfer any characters until the Special Condition has been serviced.

SYNCHRONOUS MODES
(SDLC, HDLC, BYSYNC, AND MONOSYNC MODES INCLUDED)

Q. For what are the cyclical redundancy check (CRC) residue codes used?

A. The residue codes provide a secondary method to check the reception of the message.

Q. Why is the second byte of the CRC incorrect when read from the receiving SCC?

A. The second byte of the CRC actually consists of the last two bits of the first byte or CRC, and the first six bits of the second byte of CRC.

Q. How does the SCC send CRC?

A. The SCC can be programmed to automatically send the CRC. First, write the first byte of the message to be sent. This guarantees the transmitter is full. Then reset the Transmit Underrun/EOM latch (WR0 10). Write the rest of the data frame. When the transmit buffer under-runs, the CRC is sent. The following table describes the action taken by the SCC for the bit-oriented protocols:

Tx Underrun EOM Latch Bit	Abort/Flag Bit	Action Upon Tx Underrun	Comment
0	0	Sends CRC + Flags	Valid Frame
0	0	Sends Abort + Flags	Aborted Frame
1	X	Sends Flags	Software CRC

The SCC sets the Tx Underrun/EOM latch when the CRC or Abort is loaded into the shift register for transmission. This event causes an interrupt (if enabled).

Q. In SDLC, when do you reset the CRC generator and checker?

A. The Reset TxCRC Generator command should be issued when the transmitter is enabled and idling (WR0). This needs to be done only once at initialization time for SDLC mode.

Q. How can you make sure that a flag is transmitted after CRC?

A. Use the external status end of message (EOM) interrupt to start the CRC transmission, then enable the transmit buffer empty interrupt. When you get the interrupt, it means that the buffer is empty, a flag is loaded in the shift register, and you can send the next packet of information.

Q. If the SCC is idling flags, and a byte of data is loaded into the transmit buffer, what will be transmitted?

A. Data takes priority over flags and will be loaded in the shift register and transmitted.

Q. Since data is preferred, can this cause a problem?

A. This allows you to append on the end of a message, but it can cause problems with DMA. A character could be transmitted without an opening flag. To make sure that a flag has been transmitted, watch for the W/REQ line to toggle when the flag is loaded into the shift register.

Q. Can you gate data by stretching the receive clock?

A. You can hold the clock until you have valid data. There are no maximum specs on the RxC period, and the edges are used to sample the data. If there are no edges, no data is sampled.

Q. How do you synchronize the DPLL in SDLC mode?

A. There are two methods to synchronize the DPLL. Supply at least 16 transitions at the beginning of each message so the DPLL has time to make adjustments, or use the DPLL search mode in WR14 to cause the SCC to synchronize on first transition. The first edge must be guaranteed to be a cell boundary.

Q. In SDLC, is the flag and address stripped-off?

A. No, only the flag is stripped. The address will be the 1st character received.

Q. Does IBM® SDLC specify parity?

A. No.

Q. Can the SCC include parity in SDLC mode?

A. Yes. It is appended at the end of the character.

Q. How does the SCC operate in transparent mode?

A. The transparentness, as defined by IBM SNA, should be provided by the software. The SCC does not perform any automatic insertion and deletion of link control nor does it automatically exclude the characters from the CRC calculation. This also applies to other high level protocols.

Q. When does the Abort function take effect?

A. The abort takes place immediately by inserting eight consecutive 1's.

Q. Can the SCC detect multiple aborts?

A. The SCC searched for seven consecutive 1's on the receive data line for the abort detection. This condition may be allowed to cause an external status interrupt. After these seven 1's are received, the receiver automatically enters Hunt mode, where it looks for flags. So, even if more than seven 1's are received in case of multiple aborts, only the first sequence of 1's is significant.

Q. How do you send an end of poll (EOP) flag in SDLC loop mode?

A. To send the EOP message, simply toggle the bit which idles flags or ones to mark flags, then mark ones. This produces a zero and more than seven 1's; an EOP condition.

Q. When the SCC is programmed for 6 bit sync, how are bits sent?

A. Six bits are sent. The 12-bit sync character sends 12 bits.

Q. Do sync patterns (or flags) in data transmissions get stripped and still cause interrupts?

A. All leading sync patterns (and all flags) are automatically stripped if the Sync Character Load Inhibit feature is programmed. Any data stripped from the transmission stream cannot cause a receive character available interrupt but may cause other interrupts (such as External/Status for Sync/Hunt and special receive condition for EOM).

Q. How are the sync characters sent at the beginning of a Bisync frame?

A. Load the transmit buffer with the first byte and the sync characters are automatically sent out.

Q. How can you determine when the flag has been completely sent?

A. There are several ways to determine if the flag has been completely sent. This allows the transmitter to be shut off, or in half duplex the line can be turned around. This requires a little work by the user because the SCC does not know when the last flag bit has been shifted our. The following are some suggestions:

– Once the flag is loaded into the transmit shift register, start an external clock. Use the baud rate generator as the counter.

– Tie the transmit line into DCD or an available input pin, and watch for a zero, or end of flag. If you are running half-duplex, use the local loopback mode and watch for the flag to end.

– Allow an abort, although this destroys the last character. Be sure to send a dummy character - then idle flags after the abort latch is set.

Q. How do the DMA W/REQ lines operate?

A. DMA request lines follow the state of the transmit buffer.

Q. How does the SCC handle messages less than four bytes in length?

A. A 4-byte message consists of an address, control word, no data, and 2 bytes of CRC. SDLC defines messages of less than 4-bytes as an error. It is not defined how the SCC will react, however, as tested by a SCC user, 4-, 3-, and 2-byte messages cause an interrupt on end of frame, but a 1 byte message does not cause an interrupt.

MISCELLANEOUS QUESTIONS

Q. Can the SCC support MARK and SPACE parity in async?

A. The SCC can transmit-end the equivalent of MARK parity by setting WR4 to select two STOP bits. The receiver always checks for only one STOP bit; therefore, the receiver does not verify the MARK parity bit.

The SCC (and products using the SCC cell) does notsupport SPAC parity for transmitting or receiving. The Zilog USC Family of serial datacom controllers do support odd, even, mark, & space parity types.

Q. Since both D7 and D1 bits in RR0 are not latched, it is possible that the receiver detected an Abort condition, set D7 to 1, initiated an external/status interrupt and before the processor entered the service routine, termination of the abort was detected, which reset the Break Abort bit . Currently in the TM (page 7-20), the description for Bit1: Zero Count states if the interrupt service routine does not see any changes in the External/Status conditions, it should assume that a zero count transition occurred when in fact, an Abort condition occurred and was missed. What could be done to correct this and not miss the fact that an Abort occurred?

A. Very few people actually use the Zero Count interrupt. This interrupt is generated TWICE during each bit time and is usually used to count a specific number of bits that are sent or received. If this interrupt is not used by your customer, then what is said in the TM about the Zero Count is true for the Abort Condition. If no other changes occurred in the external/status conditions and the Zero Count is not used, then the source of the interrupt was the Abort condition.

Q. Can the SCC resynchronize independent clocks (at the same frequency, but could be out of phase), one for Rx data and one for Tx data?

A. No, the two clocks are independent of each other. However, the SCC provides a special transmitter-to-receiver synchronization function that may be used to guarantee that the character boundaries for the received and transmitted data are the same.

Q. When is EOM and EOF asserted?

A. EOM is asserted when it detects depletion of data in the Tx buffer; EOF is asserted when it detects a closing flag.

Q. After powering up the SCC, are the reset values in the write and read registers guaranteed?

A. No. You must perform a hardware or software reset.

Q. Can you read the status of a write register, such as the MIE bit in WR9?

A. No, in order to retain the status of a write register, you must keep its status in a separate memory for later use. However, the only exception is that WR15 is a mirror image of RR15. Also, the ESCC has a new feature to allow the user to read some of the write registers (see the ESCC Product Specification or Technical Manual for more details).

Q. Is there a signal to indicate that a closing SDLC flag is completely shifted out of the TxD pin? This is needed to indicate that the frame is completely free of the output to allow carrier cut off without disrupting the CRC or closing flag.

A. No, the only way to find this timing is to count the number of clocks from Tx Underrun Interrupt to the closing flag. The ESCC contains the feature by deasserting the /RTS pin after the closing flag. Upgrade to the ESCC!

Q. Does the SCC detect a loss of the receive clock signal?

A. No, if the clock stops, the SCC senses that the bit time is very long. Use a watch-dog timer to detect a loss in the receive clock signal.

Q. Is there any harm in grounding the "NO CONNECT" (NC) pins in the PLCC package (pin #17,18,28,36)?

A. These NC pins are not physically connected inside the die. Therefore, it is safe to tie them to ground.

Q. Can the SCC be used as a shift register in one of the synchronous modes with only data sent to the Tx register with no CRC and no sync characters?

A. CRC is optional in Mono-, Bi-, and External Sync Modes only. The sync characters can be stripped out via software.