BOOST YOUR SYSTEM PERFORMANCE USING THE ZILOG ESCC™

For greater testability, larger interface flexibility, and increased CPU/DMA offloading, replace the SCC with the ESCC™ Controller... and utilize the ESCC to its full potential.

INTRODUCTION

This App Note (Application Note) describes the differences between the SCC (Z8030/8530, Z80C30/85C30) and ESCC (Z80230/85230). It outlines the procedures in utilizing the ESCC to its full potential. Application details such as Schematics and Program Listings are not included since these materials are in our various application support products.

Note: The author assumes the audience has fundamental Datacommunications knowledge and basic familiarity with Zilog SCC products.

Notes: All Signals with a preceding front slash, "/", are active Low, e.g.: B//W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

Connection	Circuit	Device
Power	V_{CC}	V_{DD}
Ground	GND	V_{SS}

ESCC/SCC DIFFERENCES

The differences between the ESCC and SCC are shown below:

ESCC ENHANCEMENT	PERFORMANCE BENEFITS
1. Extended Read Enable of Write Registers	- Improves Testability - Ability to examine SDLC status on-the-fly
2. Hardware Improvement	
- Modified WRITE Timing	- Improves Interface to 80X86 CPU
- Modified DMA Request on	- Improves Interface DMA-driven system
- Transmit Deactivation Timing	
3. Throughput improvement	
- Deeper Transmit FIFO	- Reduces TBE Interrupt Frequency by 3/4
- Deeper Receive FIFO	- Reduces RCA Interrupt Frequency by 3/4
- FIFO Interrupt Level	- Flexibility in Adapting CPU Workload
4 SDLC End Of Frame Improvement	
- Automatic RTS Deassertion after Closing Flag	- Reduces CPU and DMA Controller Overhead after End Of Frame
- Automatic Opening Flag Transmission	
- Automatic TxD Forced High in SDLC with NRZI Encoding When Marking Idle After End Of Frame	- Allows Optimal SDLC Line Utilization
- Improvement to Allow Transmission of Back-to-Back Frames with a Shared Flag	
- Status FIFO Anti-Lock Feature in DMA-Driven System	

The differences between the ESCC and SCC are summarized by a new register, WR7' (Figure 1).

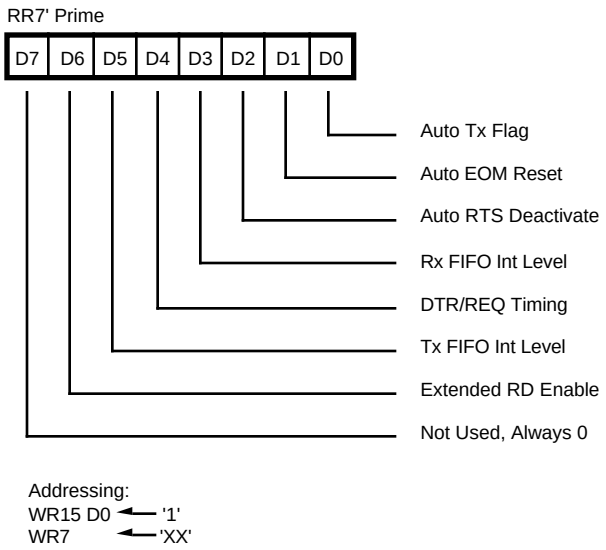


Figure 1. WR7' Definition

The advantages of the new features are illustrated in the following examples.

One of the features that is offered by the ESCC, but not the SCC, is Extended Read Enable. Write Register values from the WR3, WR4, WR5, WR7', and WR10 can be examined in the ESCC but not the SCC. This feature improves system testability. It is also crucial for SCC/ESCC differentiation and allows generic software structures for all SCC/ESCC devices.

Flowchart 1 (Figure 2) shows a generic software structure applicable for all SCC/ESCC initializations. Flowchart 2 (Figure 3) suggests a method for determining which type of SCC/ESCC™ device is in the socket. This software structure helps the development of software drivers independent of the device type.

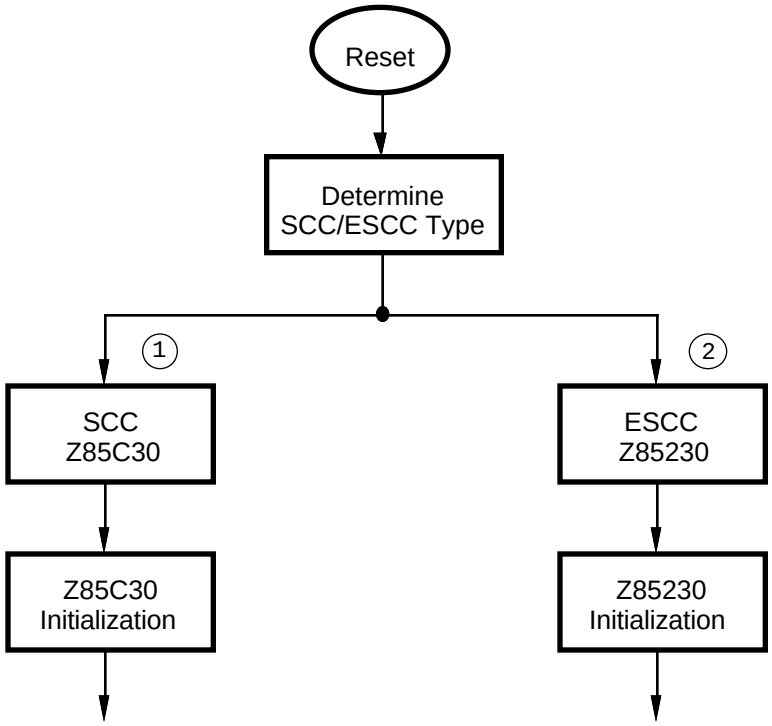


Figure 2. Generic SCC/ESCC Drivers

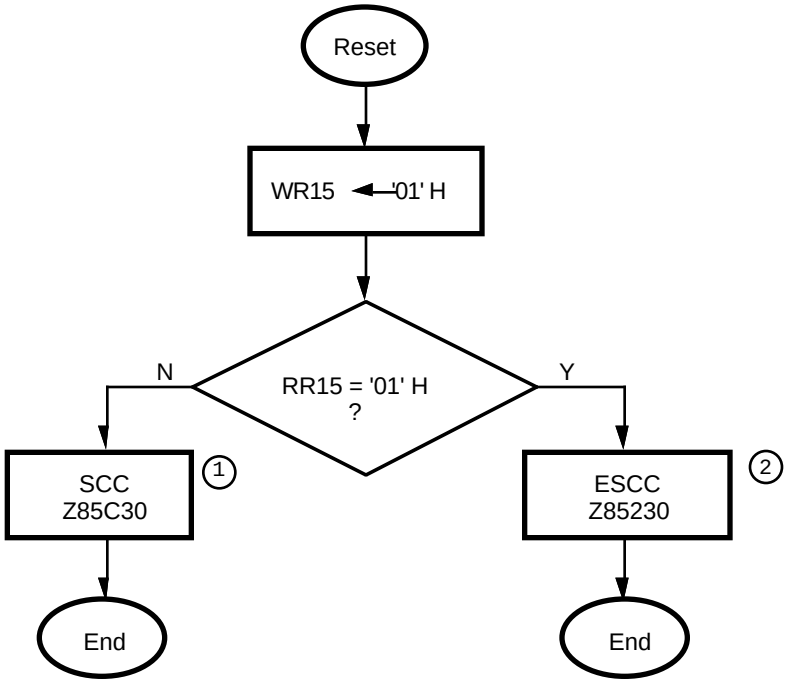
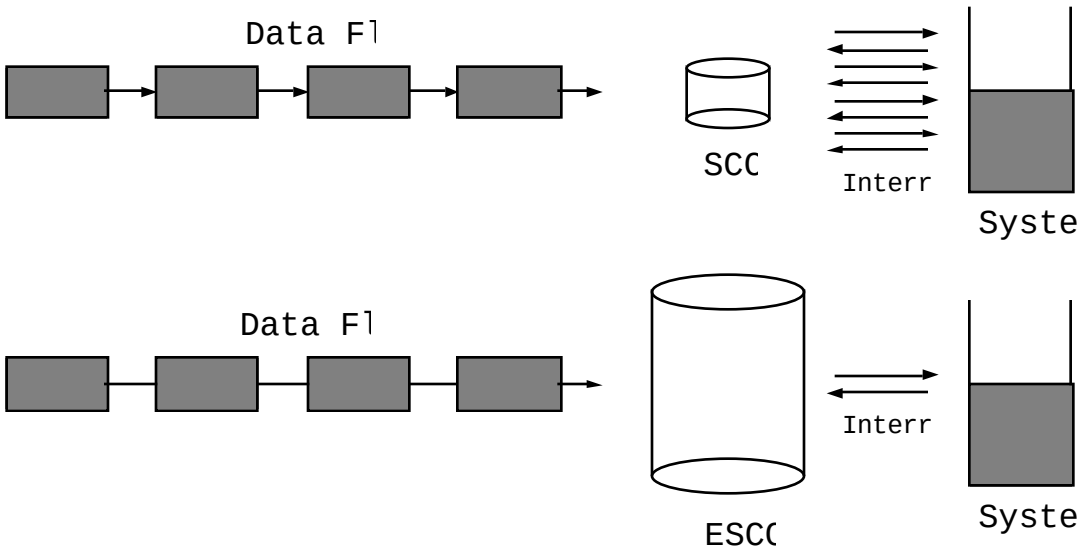


Figure 3. SCC/ESCC Differentiation Flowchart

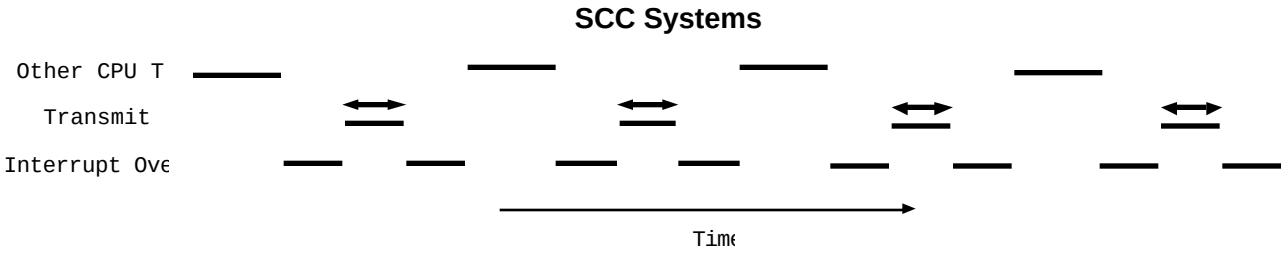
ESCC SYSTEM BENEFITS

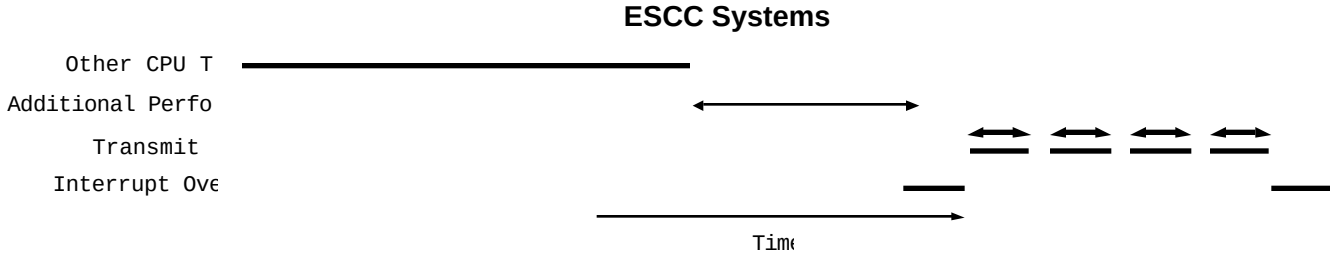
The Software Overhead sets the System Performance Limits. The ESCC's deeper FIFOs and other features significantly reduce the software overhead for each channel. This allows:

- More Channels Per System
- Faster Data Rates on Channels
- More CPU bandwidth available for other tasks
- Lower CPU Costs



Interrupt Frequency Reduction





ESCC Reduces System Workload and Allows Extra Performance

TRANSMIT FIFO INTERRUPT

In the ESCC, transmit interrupt frequencies are reduced by a deeper Transmit FIFO and the revised transmit interrupt structure. If the WR7' D5 Transmit FIFO Interrupt Level bit is reset, the transmit interrupt is generated when the entry location of the FIFO is empty, i.e., more data can be written. This is downward compatible with a SCC Transmit Interrupt since the SCC only has a one-byte transmit buffer instead of a four-byte Transmit FIFO.

If WR7' D5 is set, the transmit buffer empty interrupt is generated when the transmit FIFO is completely empty. Enabling the transmit FIFO interrupt level, together with polling the Transmit Buffer Empty (TBE) bit in RR0, causes significant transmit interrupt frequency reduction. Transmit data is sent in blocks of four bytes (algorithm is illustrated in Figure 4). This helps to offload those systems which have long interrupt latency or a fully loaded Operating System.

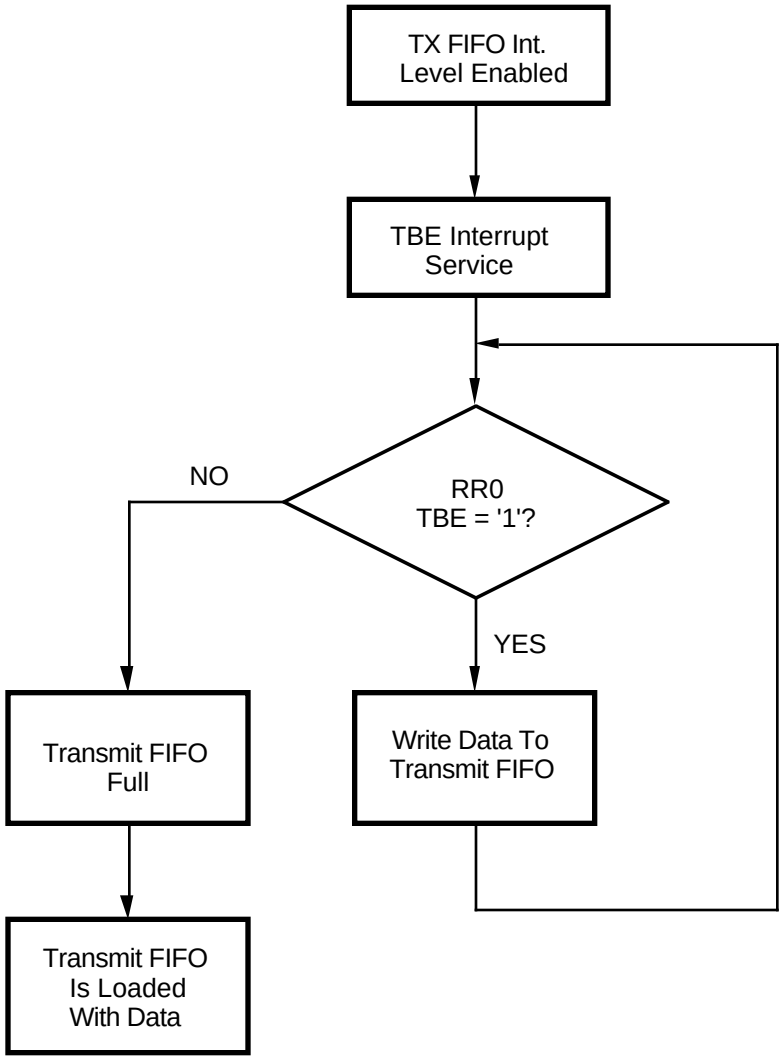


Figure 4. Flowchart of Transmit Interrupt Service Routine to Reduce Transmit Interrupt Frequencies

RECEIVE FIFO INTERRUPT

In the ESCC, receive interrupt frequencies are reduced due to a deeper Receive FIFO and the revised receive interrupt structure.

If WR7' D3 Receive FIFO Interrupt Level bit is reset, the ESCC generates the receive character available interrupt on every received character. This is compatible with SCC Receive Character Available Interrupt. If WR7' D3 is set, the Receive Character Available Interrupt is triggered when the Receive FIFO is half full; the first four locations from the entry are still empty. By enabling the receive FIFO interrupt level, together with polling the Receive Character Available (RCA) bit in RR0, the receive interrupt frequencies are reduced significantly. Receive data is read in blocks of four bytes (Figure 5). This would help to offload systems which have a long interrupt latency and heavily loaded Operating Systems.

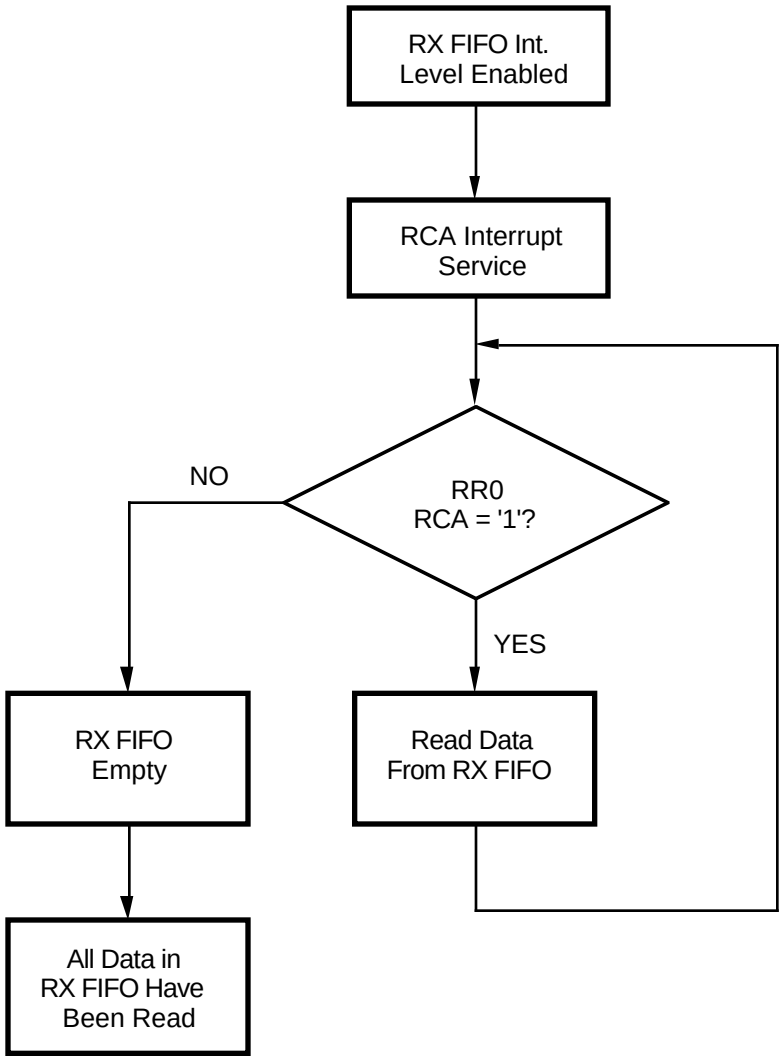


Figure 5. Flowchart of Receive Interrupt Service Routine to Reduce Receive Interrupt Frequencies

AUTOMATIC /RTS DEASSERTION

Several SDLC enhancements are provided in the ESCC. The ESCC allows automatic /RTS deassertion at End Of Frame (EOF). The automatic /RTS deassertion is enabled by setting WR7' D2. If ESCC is programmed for SDLC mode and the Flag-On-Underrun bit (WR10 D2) is reset, with the RTS bit (WR5 D1) reset, /RTS is deasserted automatically at the last bit of the closing flag. It is triggered by the rising edge of the Transmit Clock (TxC - Figures 6 and 7).

/RTS is normally used in SDLC for switching the direction of line drivers. Automatic /RTS deassertion allows optimal line switching without any software intervention. The typical procedures are as follows:

1. Enable Automatic /RTS Deassertion
2. Before frame transmission, set RTS bit
3. Enable frame transmission
4. Reset RTS bit
5. RTS pin deassertion is delayed until the last rising TxC edge closing flag.

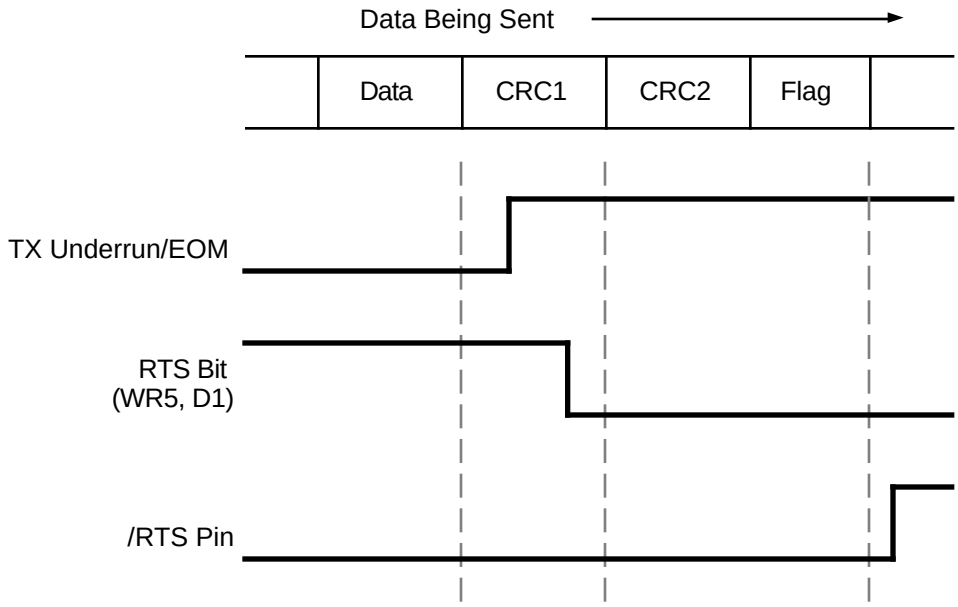


Figure 6. /RTS Deassertion Timing

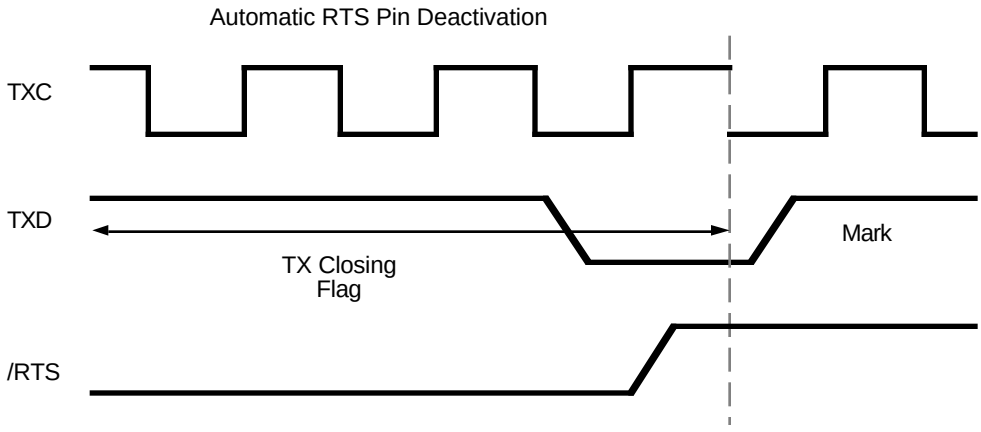


Figure 7. /RTS Deassertion Sequence

AUTOMATIC OPENING FLAG TRANSMISSION

When Auto Tx Flag (WR7', D0) is enabled, the ESCC automatically transmits a SDLC opening flag before transmitting data. This removes:

1. Requirements to reset the mark idle bit (WR10 D3) before writing data to the transmitter, or;

2. Waiting for eight bit times to load the opening flag.

TxD Forced High In SDLC With NRZI Encoding When Marking Idle After End Of Frame

When the ESCC is programmed for SDLC mode with NRZI encoding and mark idle (WR10 D6=0,D5=1,D3=1), TxD is automatically forced high when the transmitter goes to the mark idle state at EOF or when Abort is detected. This feature is used in combination with the automatic SDLC opening flag transmission to format the data packets between successive frames properly without any requirement in software intervention.

Status FIFO Enhancement

ESCC SDLC Frame Status FIFO implementation has been improved to maximize ESCC ability to interface with a DMA-driven system (Technical Manual, 4.4.3). The Status FIFO and its relationship with RR1, RR6 and RR7 is shown in Figure 8.

Other special conditions (e.g., Overrun) generates special receive conditions and lock the Receiver FIFO (Figures 9 and 10).

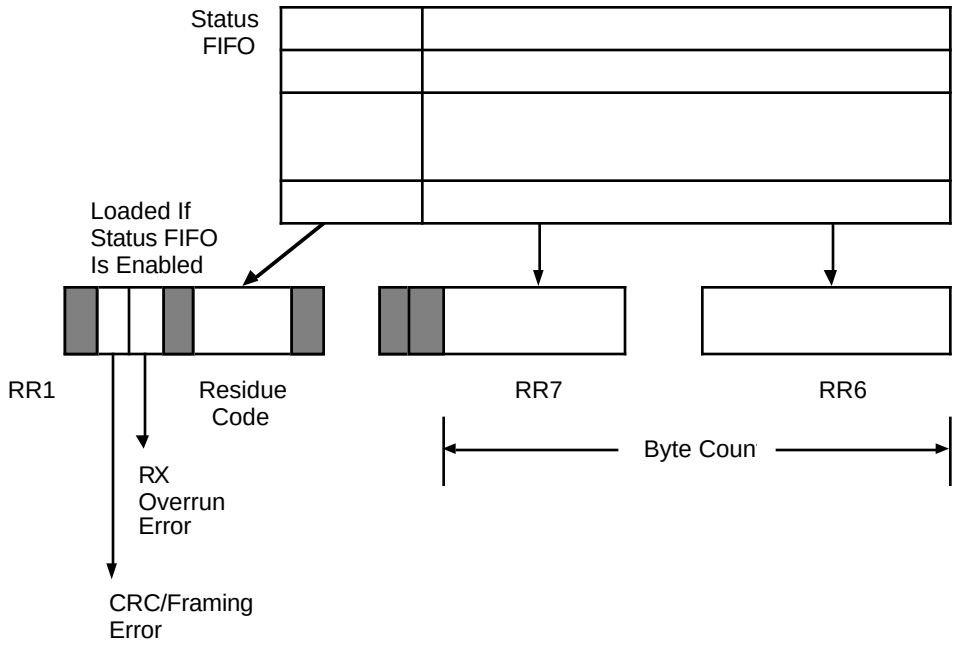


Figure 8. Status FIFO

SDLC Frame Status FIFO enhancement is enabled by setting WR15 D2. If it is enabled when EOF is detected, byte count and status from the Status FIFO are loaded into RR6, RR7 and RR1. This is used in DMA-driven systems. Historically, EOF is treated as a special condition. Special condition interrupts are triggered if any one of the below interrupts is enabled:

1. Receive Interrupt on First Character or Special Condition.
2. Interrupt on All Receive Characters or Special Conditions.
3. Special Receive Condition Only.

If 1 or 3 (above) is enabled, the data FIFO is locked after the interrupt is serviced by reading RR1 in the Status FIFO, as shown in Figure 11. This is commonly used in a DMA-driven system to avoid delivering useless information (e.g., EOF) to the data buffer. Locking the data FIFO is not desirable in systems with long interrupt latency and high data rate communications. The reason is the ERROR RESET command is necessary to unlock the FIFO. Data from the next frame may be lost if ERROR RESET fails to issue early.

This drawback is improved in the ESCC for a DMA driven system. By enabling interrupts on "Special Receive Conditions only" and SDLC status FIFO, EOF is treated differently from other special conditions. When EOF status reached the exit location of the FIFO:

1. A "Receive Data Available" interrupt is generated to signal that EOF has been reached.
2. Receive Data FIFO is not locked.

Because of these changes, the data from the next frame is securely loaded and the system processes the EOF interrupt. The only responsibility of the software is issuing the Reset Highest IUS before resuming normal operation (Figure 12).

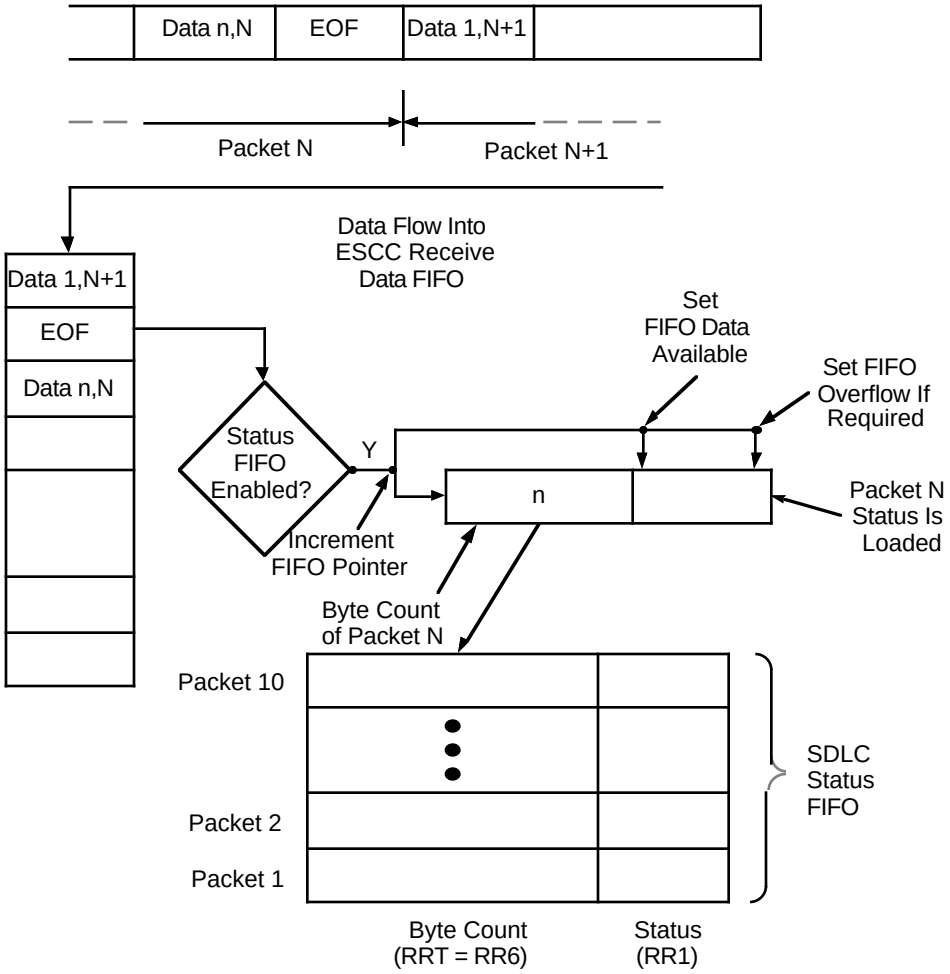
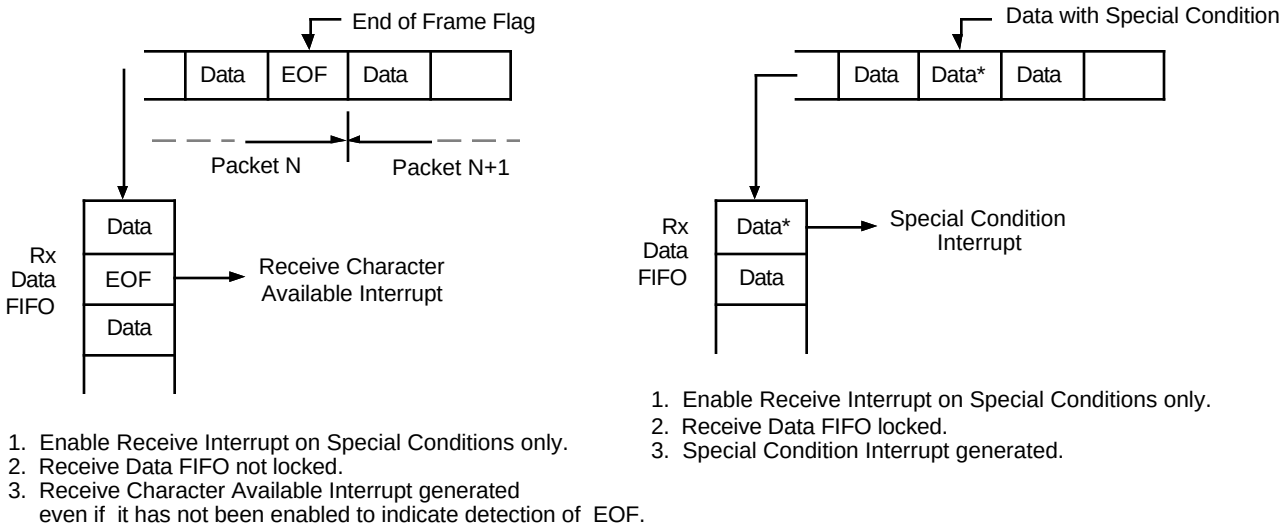


Figure 9. Status FIFO Operation at End Of Frame

AUTOMATIC OPENING FLAG TRANSMISSION (Continued)



1. Enable Receive Interrupt on Special Conditions only.
2. Receive Data FIFO not locked.
3. Receive Character Available Interrupt generated even if it has not been enabled to indicate detection of EOF.

1. Enable Receive Interrupt on Special Conditions only.
2. Receive Data FIFO locked.
3. Special Condition Interrupt generated.

Figure 10. SDLC Status FIFO Anti-Lock

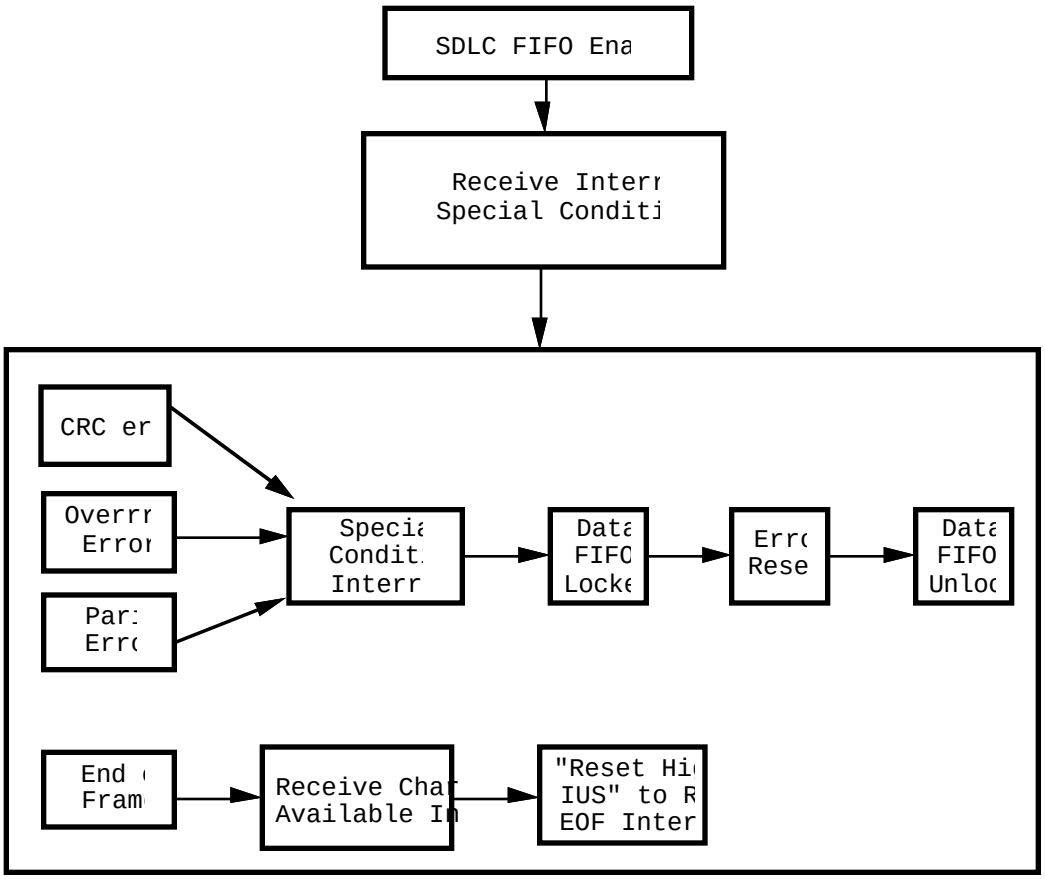


Figure 11. Receive Interrupt Mechanism 1

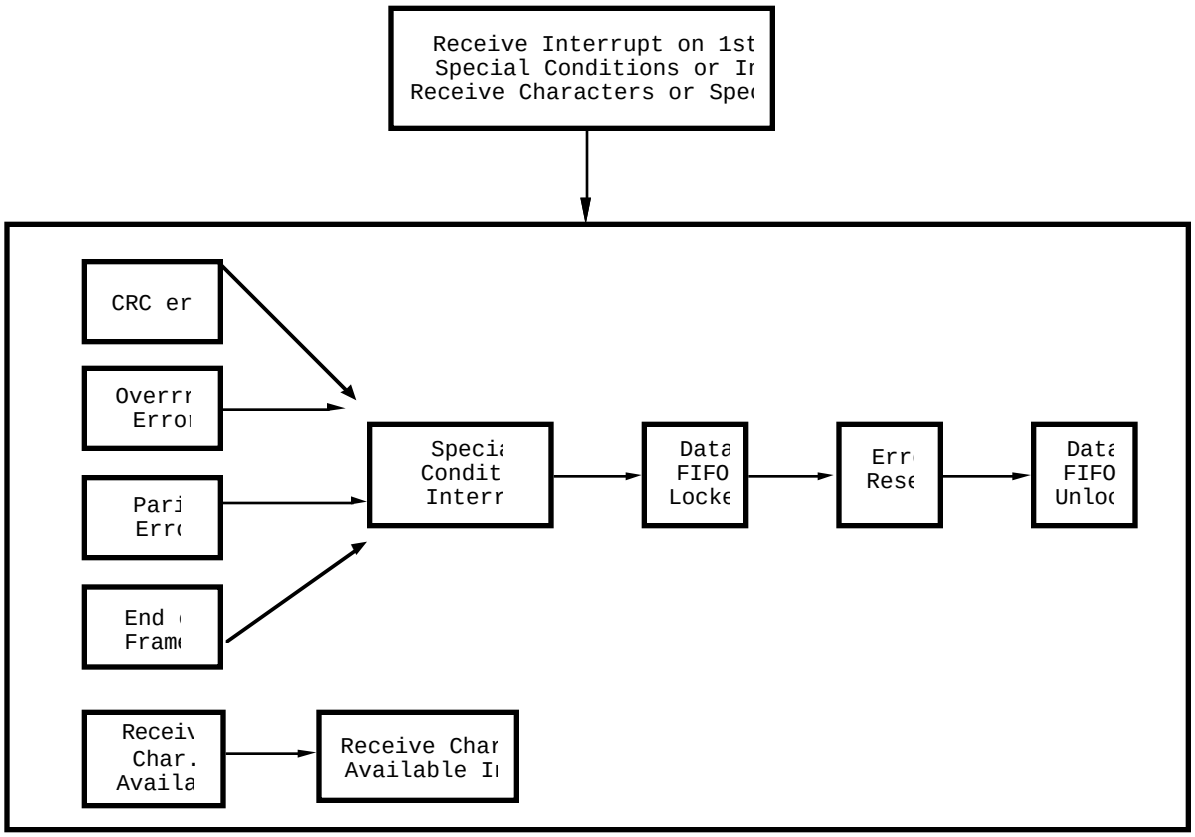


Figure 12. Receive Interrupt Mechanism 2

DMA Request on Transmit Deactivation Timing /DTR//REQ.

Timing implementation in the ESCC has been improved to make it more compatible with the DMA cycle timing (Reference Tech Manual, Section 2.5.2; DMA Request on Transmit).

Transmission of Back-To-Back Frames with a Shared Flag.

The ESCC provides facilities to allow transmission of back-to-back frames with a shared flag between frames (Figure 13).

In the ESCC, if the Automatic End Of Message (EOM) Reset feature is enabled (WR7' D1=1), data for a second frame is written to the transmit FIFO when Tx Underrun/EOM interrupt has occurred. This allows application software sufficient time to write the data to the transmit FIFO while allowing the current frame to be concluded with CRC and flag.

In the SCC, Transmission of Back-to-Back Frames is more difficult because (Figure 14):

1. Data cannot be written to the transmitter at EOF until a Transmit Buffer Empty interrupt is generated after CRC has completed transmission.

2. Automatic EOM Reset is not available in the SCC. Application software has to issue the “Reset Tx/Underrun EOM” command manually. The software overhead limits the next frame data to deliver immediately after the preceding frame has been concluded with CRC and Flag.

AUTOMATIC OPENING FLAG TRANSMISSION (Continued)

Requirements: Automatic EOM Reset and Automatic Opening Flag features are enabled.

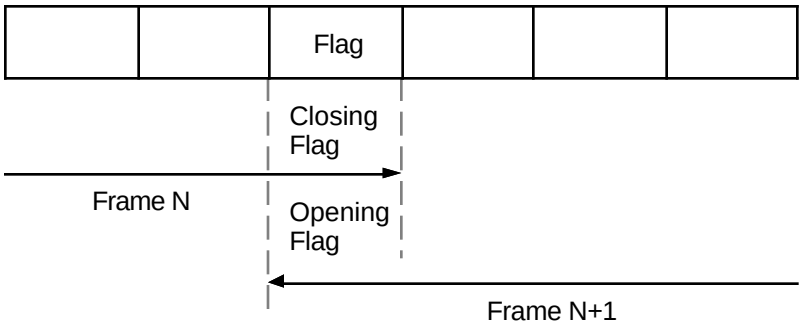


Figure 13. Transmission of Back-to-Back Frames with a Shared Flag

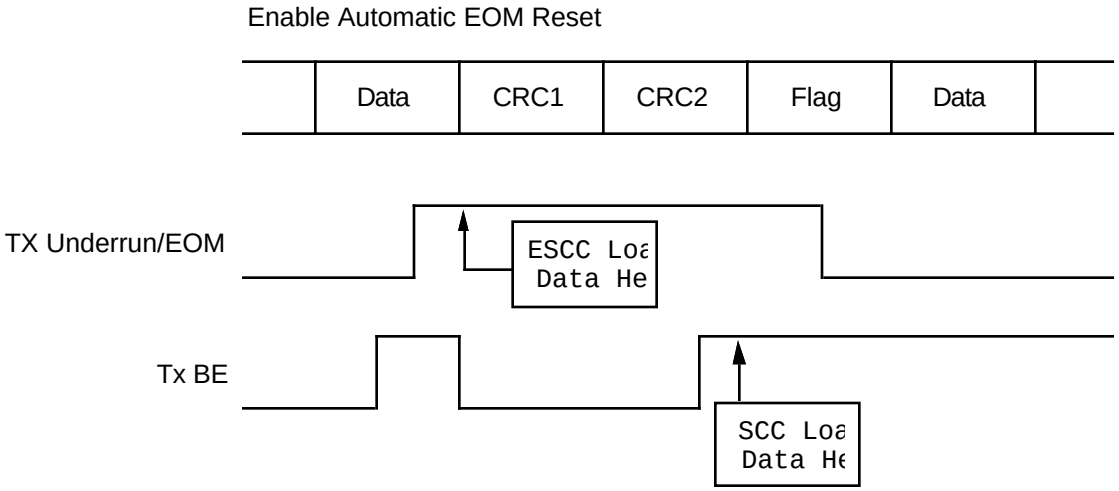
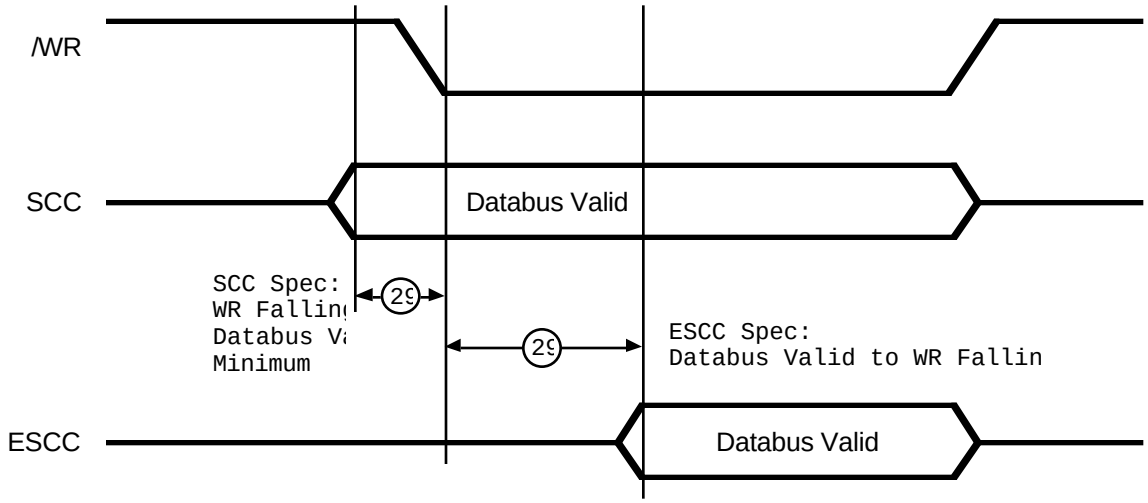


Figure 14. Operation of Shared Flag Transmission

MODIFIED WRITE TIMING

In the SCC write cycle, the SCC assumes the data is valid when /WR is asserted (Figure 15). This assumption is not valid for some CPUs, e.g., the Intel 80X86. The /WR signal from this CPU needs to delay for one more clock to initiate the write cycle. Additional hardware is required.

In the ESCC, write cycle timing has been modified so that data becomes valid a short time after write (approx. 20 ns). Therefore, if the data pins from the Intel CPU are connected directly to the ESCC, no additional logic is required.



Databus latched after falling edge of WR saves external logic required to delay WR until databus is valid. Typically needed with Intel CPUs.

Figure 15. Modified Write Timing