SCC IN BINARY SYNCHRONOUS COMMUNICATIONS

INTRODUCTION

Zilog's Z8030 Z-SCC Serial Communications Controller is one of a family of components that are Z-BUS® compatible with the Z8000™ CPU. Combined with a Z8000 CPU (or other existing 8- or 16-bit CPUs with nonmultiplexed buses when using the Z8530 SCC), the Z-SCC forms an integrated data communications controller that is more cost effective and more compact than systems incorporating UARTs, baud rate generators, and phase-locked loops as separate entities.

The approach examined here implements a communications controller in a Binary Synchronous mode of operation, with a Z8002 CPU acting as controller for the Z-SCC.

One channel of the Z-SCC is used to communicate with the remote station in Half Duplex mode at 9600 bits/second. To test this application, two Z8000 Development Modules are used. Both are loaded with the same software routines for initialization and for transmitting and receiving messages. The main program of one module requests the transmit routine to send a message of the length indicated in the 'COUNT' parameter. The other system receives the incoming data stream, storing the message in its resident memory.

DATA TRANSFER MODES

The Z-SCC system interface supports the following data transfer modes:

- **Polled Mode.** The CPU periodically polls the Z-SCC status registers to determine the availability of a received character, if a character is needed for transmission, and if any errors have been detected.
- **Interrupt Mode.** The Z-SCC interrupts the CPU when certain previously defined conditions are met.
- **Block/DMA Mode.** Using the Wait/Request (/W//REQ) signal, the Z-SCC introduces extra wait cycles to synchronize data transfer between a CPU or DMA controller and the Z-SCC.

The example given here uses the block mode of data transfer in its transmit and receive routines.

SYNCHRONOUS MODES

Three variations of character-oriented synchronous communications are supported by the Z-SCC: Mono-sync, Bisync, and External Sync (Figure 1). In Monosync mode, a single sync character is transmitted, which is then compared to an identical sync character in the receiver. When the receiver recognizes this sync character, synchronization is complete; the receiver then transfers subsequent characters into the receiver FIFO in the Z-SCC.

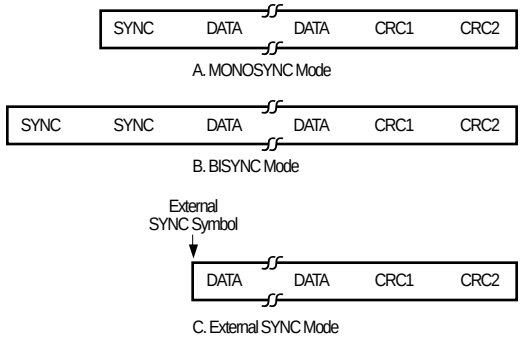

Figure 1. Synchronous Modes of Communication

Bisync mode uses a 16-bit or 12-bit sync character in the same way to obtain synchronization. External Sync mode uses an external signal to mark the beginning of the data field; i.e., an external input pin (SYNC) indicates the start of the information field.

In all synchronous modes, two Cyclic Redundancy Check (CRC) bytes can be concatenated to the message to detect data transmission errors. The CRC bytes inserted in the transmitted message are compared to the CRC bytes computed to the receiver. Any differences found are held in the receive error FIFO.

SYSTEM INTERFACE

The Z8002 Development Module consists of a Z8002 CPU, 16K words of dynamic RAM, 2K words of EPROM monitor, a Z80A SIO providing dual serial ports, a Z80A CTC peripheral device providing four counter/timer channels, two Z80A PIO devices providing 32 programmable I/O lines, and wire wrap area for prototyping. The block diagram is depicted in Figure 2. Each of the peripherals in the development module is connected in a prioritized daisy-chain configuration. The Z-SCC is included in this configuration by tying its IEI line to the IEO line of another device, thus making it one stop lower in interrupt priority compared to the other device.

Two Z8000 Development Modules containing Z-SCCs are connected as shown in Figure 3 and Figure 4. The Transmit Data pin of one is connected to the Receive Data pin of the other and vice versa. The Z8002 is used as a host CPU for loading the modules' memories with software routines.

The Z8000 CPU can address either of the two bytes contained in 16-bit words. The CPU uses an even address (16 bits) to access the most-significant byte of a word and an odd address for the least-significant byte of a word.

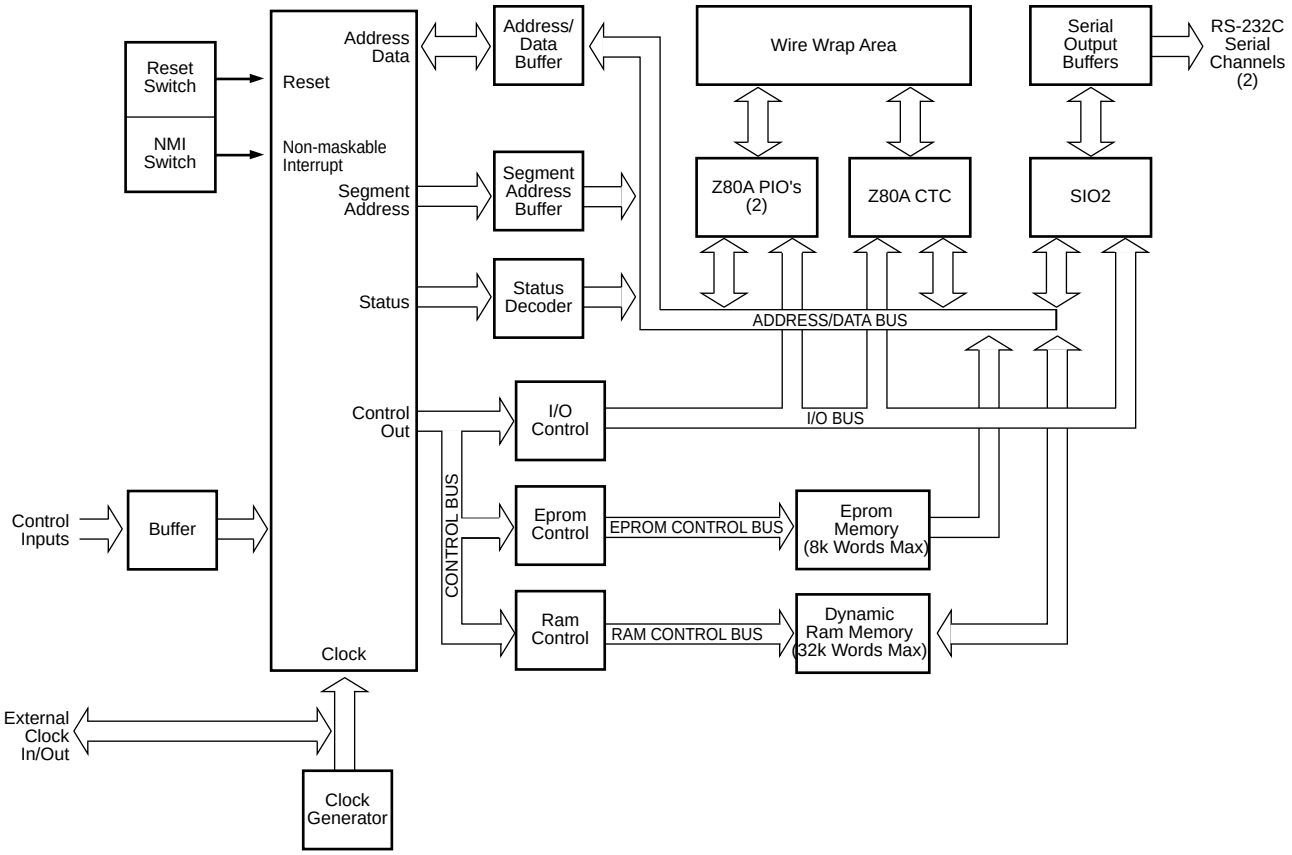

Figure 2. Block Diagram of Z8000 DM

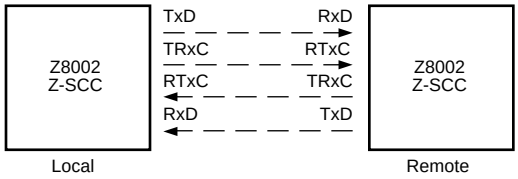

Figure 3. Block Diagram of Two Z8000 Development Modules

SYSTEM INTERFACE (Continued)

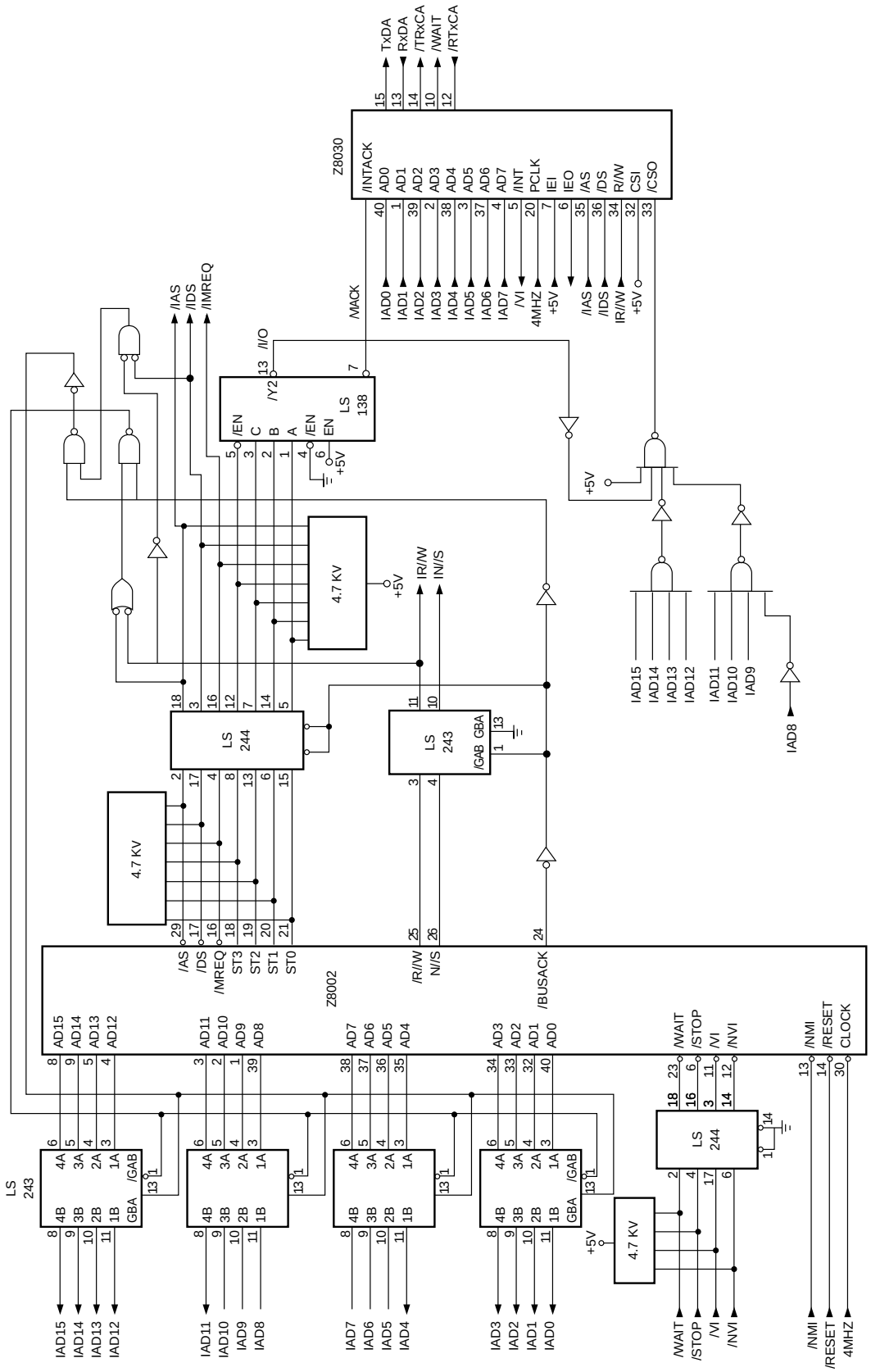

Figure 4. Z8002 with SCC

When the Z8002 CPU uses the lower half of the Address/Data bus (AD0-AD7 the least significant byte) for byte read and write transactions during I/O operations, these transactions are performed between the CPU and I/O ports located at odd I/O addresses. Since the Z-SCC is attached to the CPU on the lower half of the A/D bus, its registers must appear to the CPU at odd I/O addresses. To achieve this, the Z-SCC can be programmed to select its internal registers using lines AD5-AD1. This is done either automatically with the Force Hardware Reset command in WR9 or by sending a Select Shift Left Mode command to WR0B in channel B of the Z-SCC. For this application, the Z-SCC registers are located at I/O port address 'FExx'. The Chip Select signal (/CS0) is derived by decoding I/O address 'FE' hex from lines AD15-AD8 of the controller. The Read/Write registers are automatically selected by the Z-SCC when internally decoding lines AD5-AD1 in Shift Left mode. To select the Read/Write registers automatically, the Z-SCC decodes lines AD5-AD1 in Shift Left mode. The register map for the Z-SCC is depicted in Table 1.

INITIALIZATION

The Z-SCC can be initialized for use in different modes by setting various bits in its Write registers. First, a hardware reset must be performed by setting bits 7 and 6 of WR9 to one; the rest of the bits are disabled by writing a logic zero.

Bisync mode is established by selecting a 16-bit sync character, Sync Mode Enable, and a Xl clock in WR4. A data rate of 9600 baud, NRZ encoding, and a data character length of eight bits are among the other options that are selected in this example (Table 2).

Note that WR9 is accessed twice, first to perform a hardware reset and again at the end of the initialization sequence to enable the interrupts. The programming sequence depicted in Table 2 establishes the necessary parameters for the receiver and the transmitter so that, when enabled, they are ready to perform communication tasks. To avoid internal race and false interrupt conditions, it is important to initialize the registers in the sequence depicted in this application note.

Table 1. Register Map

Address (hex)	Write Register	Read Register
FE01	WR0B	RR0B
FE03	WR1B	RR1B
FE05	WR2	RR2B
FE07	WR3B	RR3B
FE09	WR4B	
FE0B	WR5B	
FE0D	WR6B	
FE0F	WR7B	
FE11	B DATA	B DATA
FE13	WR9	
FE15	WR10B	RR10B
FE17	WR11B	
FE19	WR12B	RR12B
FE1B	WR13B	RR13B
FE1D	WR14B	
FE1F	WR15B	RR15B
FE21	WR0A	RR0A
FE23	WR1A	RR1A
FE25	WR2	RR2A
FE27	WR3A	RR3A
FE29	WR4A	
FE2B	WR5A	
FE2D	WR6A	
FE2F	WR7A	
FE31	A DATA	A DATA
FE33	WR9	
FE35	WR10A	RR10A
FE37	WR11A	
FE39	WR12A	RR12A
FE3B	WR13A	RR13A
FE3D	WR14A	
FE3F	WR15A	RR15A

INITIALIZATION (Continued)

The Z8002 CPU must be operated in System mode in order to execute privileged I/O instructions, so the Flag Control Word (FCW) should be loaded with System/Normal (S//N), and the Vectored Interrupt Enable (VIE) bits set. The Program Status Area Pointer (PSAP) is loaded with address %4400 using the Load Control instruction (LDCTL). If the Z8000 Development Module is intended to be used, the PSAP need not be loaded by the programmer as the development modules monitor loads it automatically after the NMI button is pressed.

Table 2. Programming Sequence for Initialization

Register	Value (hex)	Effect
WR9	C0	Hardware reset
WR4	10	x1 clock, 16-bit sync, sync mode enable
WR10	0	NRZ, CRC preset to zero
WR6	AB	Any sync character “AB”
WR7	CD	Any sync character “CD”
WR2	20	Interrupt vector “20”
WR11	16	Tx clock from BRG output, TRxC pin = BRG out
WR12	CE	Lower byte of time constant = “CE” for 9600 baud
WR13	0	Upper byte = 0
WR14	03	BRG source bit = 1 for PCLK as input, BRG enable
WR15	00	External interrupt disable
WR5	64	Tx 8 bits/character, CRC-16
WR3	C1	Rx8 bits/character, Rx enable (Automatic Hunt mode)
WR1	08	RxInt on 1st char & sp. cond., ext. int. disable)
WR9	09	MIE, VIS, Status Low

Since VIS and Status Low are selected in WR9, the vectors listed in Table 3 will be returned during the Interrupt Acknowledge cycle. Of the four interrupts listed, only two, Ch A Receive Character Available and Ch A Special Receive Condition, are used in the example given here.

Table 3. Interrupt Vectors

Vector (hex)	PS Address* (hex)	Interrupt
28	446E	Ch A Transmit Buffer Empty
2A	4472	Ch A External Status Change
2C	4476	Ch A Receive Char. Available
2E	447A	Ch A Special Receive Condition

* “PS Address” refers to the location in the Program Status Area where the service routine address is stored for that particular interrupt, assuming that PSAP has been set to 4400 hex.

TRANSMIT OPERATION

To transmit a block of data, the main program calls up the transmit data routine. With this routine, each message block to be transmitted is stored in memory, beginning with location 'TBUF'. The number of characters contained in each block is determined by the value assigned to the 'COUNT' parameter in the main module.

To prepare for transmission, the routine enables the transmitter and selects the Wait On Transmit function; it then enables the wait function. The Wait On Transmit function indicates to the CPU whether or not the Z-SCC is ready to accept data from the CPU. If the CPU attempts to send data to the Z-SCC when the transmit buffer is full, the Z-SCC asserts its Wait line and keeps it Low until the buffer is empty. In response, the CPU extends its I/O cycles until the Wait line goes inactive, indicating that the Z-SCC is ready to receive data.

The CRC generator is reset and the Transmit CRC bit is enabled before the first character is sent, thus including all the characters sent to the Z-SCC in the CRC calculation, until the Transmit CRC bit is disabled. CRC generation can be disabled for a particular character by resetting the TxCRC bit within the transmit routine. In this application, however, the Transmit CRC bit is not disabled, so that all characters sent to the Z-SCC are included in the CRC calculation.

The Z-SCC's transmit underrun/EOM latch must be reset sometime after the first character is transmitted by writing a Reset Tx Underrun/EOM command to WR0. When this latch is reset, the Z-SCC automatically appends the CRC characters to the end of the message in the case of an underrun condition.

Finally, a five-character delay is introduced at the end of the transmission, which allows the Z-SCC sufficient time to transmit the last data byte, two CRC characters, and two sync characters before disabling the transmitter.

RECEIVE OPERATION

Once the Z-SCC is initialized, it can be prepared to receive data. First, the receiver is enabled, placing the Z-SCC in Hunt mode and thus setting the Sync/Hunt bit in status register RR0 to 1. In Hunt mode, the receiver is idle except that it searches the incoming data stream for a sync character match. When a match is discovered between the incoming data stream and the sync characters stored in WR6 and WR7, the receiver exits the Hunt mode, resetting the Sync/Hunt bit in status register RR0 and establishing the Receive Interrupt On First Character mode. Upon detection of the receive interrupt, the CPU generates an Interrupt Acknowledge cycle. The Z-SCC sends to the CPU vector %2C, which points to the location in the Program Status Area from which the receive interrupt service routine is accessed.

The receive data routine is called from within the receive interrupt service routine. While expecting a block of data, the Wait On Receive function is enabled. Receive data buffer RR8 is read, and the characters are stored in memory locations starting at RBUF. The Start of Text (%02) character is discarded. After the End of Transmission character (%04) is received, the two CRC bytes are read. The result of the CRC check becomes valid two characters later, at which time, RR1 is read and the CRC error bit is checked. If the bit is zero, the message received can be assumed correct; if the bit is 1, an error in the transmission is indicated.

Before leaving the interrupt service routine, Reset Highest IUS (Interrupt Under Service), Enable Interrupt on Next Receive Character, and Enter Hunt Mode commands are issued to the Z-SCC.

If a receive overrun error is made, a special condition interrupt occurs. The Z-SCC presents the vector %2E to the CPU, and the service routine located at address %447A is executed. The Special Receive Condition register RR1 is read to determine which error occurred. Appropriate action to correct the error should be taken by the user at this point. Error Reset and Reset Highest IUS commands are given to the Z-SCC before returning to the main program so that the other lower priority interrupts can occur.

SOFTWARE

Software routines are presented in the following pages. These routines can be modified to include various versions of Bisync protocol, such as Transparent and Nontransparent modes. Encoding methods other than NRZ (e.g., NRZI, FM0, FM1) can also be used by modifying WR10.

APPENDIX

SOFTWARE ROUTINES

plzasm 1.3

```
LOC  OBJ CODE  STMT SOURCE    STATEMENT
                1             BISYNC MODULE
                    $LISTON   $TTY
                    CONSTANT
                    WR0A      :=    %FE21       !BASE ADDRESS FOR WR0 CHANNEL A!
                    RR0A      :=    %FE21       !BASE ADDRESS FOR RR0 CHANNEL A!
                    RBUF      :=    %5400       !BUFFER AREA FOR RECEIVE CHARACTER!
                    PSAREA    :=    %4400       !START ADDRESS FOR PROGRAM STAT AREAS!
                    COUNT     :=    12          !NO. OF CHAR FOR TRANSMIT ROUTINE!
0000                GLOBAL MAIN PROCEDURE
                    ENTRY
0000 7601                     LDA   R1, PSAREA
0002 4400
0004 7D1D                     LDCTL PSAPOFF,R1  !LOAD PSAP
0006 2100                     LD    RO,#%5000
0008 5000
000A 3310                     LD    RI(#%IC),R0 !FCW VALUE(%5000) AT %441C FOR VECTORED!
000C 001C                                       !INTERRUPTS!
000E 7600                     LDA   R0,REC
0010 00F4'
0012 3310                     LD    RI(#%76),R0 !EXT. STATUS SERVICE ADDR. AT %4476 IN!
0014 0076                                       !PSA!
0016 7600                     LDA   R0, SPCOND
0018 011E'
00IA 3310                     LD    R1(#%7A),R0 !SP.COND.SERVICE ADDR AT %447A IN PSA!
001C 007A
001E 5F00                     CALL  INIT
0020 0034'
0022 5F00                     CALL  TRANSMIT
0024 00A6'
0026 E8FF                     JR    $
0028 02         TBUF:         BVAL  %02         !START OF TEXT!
0029 31                       BVAL  '1'         !BVAL MEANS BYTE VALUE. MESSAGE CHAR.!
002A 32                       BVAL  '2'
002B 33                       BVAL  '3'
002C 34                       BVAL  '4'
002D 35                       BVAL  '5'
002E 36                       BVAL  '6'
002F 37                       BVAL  '7'
0030 38                       BVAL  '8'
0031 39                       BVAL  '9'
0032 30                       BVAL  '0'
0033 31                       BVAL  '1'
0034                          END   MAIN
```

INITIALIZATION ROUTINE FOR Z-SCC

0034		GLOBAL ENTRY	INIT	PROCEDURE	
0634	2100		LD	R0, #15	!NO.OF PORTS TO WRITE TO!
0036	000F				
0038	7602		LDA	R2, SCCTAB	!ADDRESS OF DATA FOR PORTS!
003A	004E'				
003C	2101	ALOOP:	LD	R1, #WR0A	
003E	FE21				
0040	0029		ADDB	RL1, @R2	
0042	A920		INC	R2	
0044	3A22		OUTIB	@Rl, @R2,R0	!POINT TO WR0A,WR1A ETC THRO LOOP!
0046	0018				
0048	8D04		TEST	R0	!END OF LOOP?!
004A	EEF8		JR	NZ, ALOOP	!NO, KEEP LOOPING!
004C	9E08		RET		
004E	12	SCCTAB:	BVAL	2*9	
004F	CO		BVAL	%C0	!WR9=HARDWARE RESET!
0050	08		BVAL	2*4	
0051	10		BVAL	%10	!WR4=X1 CLK, 16 BIT SYNC MODE!
0052	14		BVAL	2*10	
0053	00		BVAL	0	!WRIO=CRC PRESET ZERO, NRZ,16 BIT SYNC!
0054	0C		BVAL	2*6	
0055	AB		BVAL	%AB	!WR6=ANY SYNC CHAR %AB!
0056	0E		BVAL	2*7	
0057	CD		BVAL	%CD	!WR7=ANY SYNC CHARR %CD!
0058	04		BVAL	2*2	
0059	20		BVAL	%20	!WR2=NT VECTOR %20!
005A	16		BVAL	2*11	
005B	16		BVAL	%16	!WR11=TxCLOCK & TRxC OUT=BRG OUT!
005C	18		BVAL	2*12	
005D	CE		BVAL	%CE	!WR12= LOWER TC=%CE!
005E	IA		BVAL.	2*13	
005F	00		BVAL	0	!WR13= UPPER TC=01
0060	1C		BVAL	2*14	
0061	03		BVAL	%03	!WRI4=BRG ON, ITS SRC=PCLK!
0062	1E		BVAL	2*15	
0063	00		BVAL	%00	!WRI5=NO EXT INT EN.!
0064	0A		BVAL	2*5	
0065	64		BVAL	%64	!WR5= TX 8 BITS/CHAR, CRC-16!
0066	06		BVAL	2*3	
0067	Cl		BVAL	&CI	IWR3=RX 8 BITS/CHAR, REC ENABLE!
0068	02		BVAL	2*1	
0069	08		BVAL	%C1	!WR1=RxINT ON 1ST OR SP COND! !EXT INT DISABLE!
006A	12		BVAL	2*9	
006B	09		BVAL	%09	!WR9=MIE, VIS, STATUS LOW!
006C		END INIT			

RECEIVE ROUTINE

RECEIVE A BLOCK OF MESSAGE
THE LAST CHARACTER SHOULD BE EOT (%04)

006C		GLOBAL ENTRY	RECEIVE	PROCEDURE	
006C	C828		LDB	RL0,#428	!WAIT ON RECV.!
006C	3A86		OUTB	WR0A+2,RL0	
0070	FE23				
0072	6000		LDB	RL0,%A8	
0074	00AB				
0076	3A86		OUTB	WR0A+2,RL0	!ENABLE WAIT 1ST CHAR,SP.COND. INT!
0078	FE23				
007A	2101		LD	Rl,#RR0A+16	
007C	FE31				
007E	3Cl8		INB	RL0,@R1	!READ STX CHARACTER!
0080	C8C9		LDB	RL0,#%C9	
0082	3AB6		OUTB	WR0A+6,RL0	!Rx CRC ENABLE!
0084	FE27				
0086	2103		LD	R3,#RBUF	
0088	5400				
008A	3C18	READ:	INB	RL0,@R1	!READ MESSAGE!
008C	2E38		LDB	@R3,RL0	!STORE CHARACTER IN RBUF!
008E	AB30		DEC	R3,#l	
0090	0A08		CPB	RL0,#%04	!IS IT END OF TRANSMISSION ?!
0092	0404				
0094	EEFA		JR	NZ,READ	
0096	3C18		INB	RL0,@R1	!READ PAD1!
0098	3C18		INB	RL0,@R1	!READ PAD2!
009A	3A84		INB	RL0,RR0A+2	!READ CRC STATUS!
009C	FE23				
		! PROCESS CRC ERROR IF ANY, AND GIVE ERROR RESET COMMAND IN WR0A!			
009E	C800		LDB	RL0,#0	
00A0	3A86		OUTB	WR0A+6,RL0	!DISABLE RECEIVER!
00A2	FE27				
00A4	9E08		RET		
00A6		END RECEIVE			

TRANSMIT ROUTINE

SEND A BLOCK OF DATA CHARACTERS
THE BLOCK STARTS AT LOCATION TBUP

OA6		GLOBAL ENTRY	TRANSMIT	PROCEDURE	
00A6	2102		LD	R2, #TBUF	!PTR TO START OF BUFFER!
00AB	0028'				
00AA	C86C		LDB	RL0, #%6C	
00AC	3AB6		OUTB	WR0A+10, RL0	!ENABLE TRANSMITTER!
00AE	FE2B				
00B0	C800		LDB	RL0, #%00	!WAIT ON TRANSMIT!
00B2	3A86		OUTB	WR0A+2 , RL0	
00B4	FE23				
00B6	C888		LDB	RL0, #%88	
00B8	3AB6		OUTB	WR0A+2, RL0	!WAIT ENABLE, INT ON 1ST & SP COND!
00BA	FE23				
00BC	C880		LDB	RL0, #%80	
00BE	3A86		OUTB	WR0A, RL0	!RESET TxCRC GENERATOR!
00C0	FE21				
00C2	2101		LD	R1, #WR0A+16	!WR8A SELECTED!
00C4	FE31				
00C6	C86D		LDB	RL0, #%6D	
00C8	3A86		OUTB	WR0A+10, RL0	!Tx CRC ENABLE!
00CA	FE2B				
00CC	2100		LD	R0, #1	
00CE	0001				
00D0	3A22		OTIRB	@Rl, @R2,R0	!SEND START OF TEXT!
00D2	0010				
00D4	C8C0		LDB	RL0, #%C0	
00D6	3AB6		OUTB	WR0A, RL0	!RESET TxUND/EOM LATCH!
00D8	FE21				
00DA	2100		LD	R0, #COUNT-1	
00DC	000B				
00DE	3A22		OTIRB	@Rl, @R2, R0	!SEND MESSAGE!
00E0	0010				
00E2	C804		LDB	RL0, #%04	
00E4	3EI8		OUTB	@R1, RL0	!SEND END OF TRANSMISSION CHARACTER!
00E6	2100		LD	R0, #1670	!CREATE DELAY BEFORE DISABLING!
00E8	0686				
00EA	F081	DEL:	DJNZ	R0, DEL	
00EC	C800		LDB	RL0, #0	
00EE	3AB6		OUTB	WR0A+10, RL0	!DISABLE TRANSMITTER!
00F0	FE2B				
00F2	9E0B		RET		
00F4		END TRANSMIT			

RECEIVE INT. SERVICE ROUTINE

00F4		GLOBAL ENTRY	REC	PROCEDURE	
00F4	93F0		PUSH	@RI5, R0	
00F6	3A84		INB	RL0, RR0A	!READ STATUS FROM RR0A!
00F8	FE21				
00FA	A684		BITB	RL0, #4	!TEST IF SYNC HUNT RESET!
00FC	EE02		JR	NZ, RESET	!YES CALL RECEIVE ROUTINE!
00FE	5F00		CALL	RECEIVE	
0100	006C’				
0102	C808	RESET:	LDB	RL0, #%08	
0104	3A86		OUTB	WR0A+2, RL0	!WAIT DISABLE!
0106	FE23				
0108	C8D1		LDB	RL0, #%D1	
010A	3A86		OUTB	WR0A+6, RL0	!ENTER HUNT MODE!
010C	FE27				
010E	C820		LDB	RL0, #%20	
0110	3A86		OUTB	WR0A, RL0	!ENABLE INT ON NEXT CHAR!
0112	FE21				
0114	C838		LDB	RL0, #%38	
0116	3A86		OUTB	WR0A, RL0	!RESET HIGHEST IUS!
0118	FE21				
011A	97F0		POP	R0, @RI5	
011C	7B00		IRET		
0IIE		END REC			

SPECIAL CONDITION INTERRUPT SERVICE ROUTINE

```
011E                GLOBAL
                    ENTRY       SPCOND      PROCEDURE
011E    93F0                    PUSH        @Rl5, R0
0120    3A84                    INB         RL0, RR0A+2     !READ ERRORS!
0122    FE23
                                !PROCESS ERRORS!
0124    C830                    LDB         RL0, #%30
0126    3A8B6                   OUTB        WR0A, RL0       !ERROR RESET!
0128    FE21
012A    C808                    LDB         RL0, #%08
012C    3A86                    OUTB        WR0A+2, RL0     !WAIT DISABLE, RxINT ON 1ST OR SP COND.!
012E    FE23
0130    C0D1                    LDB         RL0, #%D1
0132    3A86                    OUTB        WR0A+6, RL0     !HUNT MODE, REC. ENABLE!
0134    FE27
0136    C838                    LDB         RL0, #%38
0138    3A86                    OUTB        WR0A, RL0       !RESET HIGHEST IUS!
013A    FE21
013C    97F0                    POP         R0, @Rl5
013E    7B00                    IRET

0140                END SPCOND

                    END BISYNC
```

o errors
Assembly complete