

TMS9903 同步通信控制器

TMS9903 概述

TMS9903 同步通信控制器相当于增加了同步和 SDLC 功能的 TMS9902 异步通信控制器。虽然该芯片被称为同步通信控制器，但也具有异步通信功能。该芯片的所有信号与 TTL 兼容。

TMS9903 占用 32 个 CRU 位，编号大的 CRU 位写入控制寄存器，并从状态寄存器读出。编号小的 CRU 位构成双向可变宽度内部数据总线，通过这条数据总线 CPU 可从接收缓冲器读数据，或者可读出三个循环冗余字中的一个。CPU 可向发送缓冲器、参数寄存器或两个同步寄存器中的一个写入，也可输出包含在两个循环冗余字之一的数据。

TMS9903 可读出三个循环冗余字。发送和接收逻辑在程序控制下为发送和接收信息各算出一个循环冗余字。只有在 SDLC 状态，接收逻辑才抽出接收帧的循环冗余字，并保存在一个可读出的寄存器里。发送和接收逻辑各有自己的缓冲器。数据从发送缓冲器传输到发送移位寄存器，从这里经由 TxD 串行地输出。用户有一个字符的发送时间用来向发送缓冲器写另一个字符，如不写，就出现脱空。接收逻辑在接收移位寄存器组装字符，组装完送到接收缓冲器。也有一个字符的接收时间读出接收缓冲器的内容，否则就出现接收溢出。

TMS9903 内的数据缓冲器都是 9 位宽，这样就有向 8 位字符附加一个奇偶位的余地。状态寄存器是 23 位宽，控制寄存器是 20 位宽，参数寄存器是 12 位宽。由于 TMS9903 与 TMS9900 系列微处理器之间的 CRU 接口特性，这些奇特的位长不会发生问题。

同步 1 和同步 2 寄存器保存同步字符。在某些规约里这两个寄存器可保存专用的控制字符。发送逻辑在信息的开头或脱空之后可输出这两个寄存器之一或两个的内容。接收逻辑利用同步 1 寄存器的内容来检测所收到的数据流中的同步字符。

用户应当通过参数寄存器位的设定来指定所接收数据的每个字符的位数。当接收逻辑在接收移位寄存器中装配字符时，它利用了每字符位数的规定，这个规定实际上就是什么时候开始被装配。如果改变每字符位数的规定，则此改变要到下一个接收字符边界才被确认。用户在参数寄存器里规定的每字符位数不适用于发送逻辑，也不适用于同步 1 寄存器和同步 2 寄存器。对这三个寄存器来说，用户写入寄存器的位数也就是要发送的数据位数。最新装入同步寄存器的数据决定着发送两个同步字符时的字符长度。

正如接收器的情况，用户可逐个字符地改变发送字符的长度。随着将每个字符从发送缓冲器移到发送移位寄存器，发送逻辑将每个字符位数的规定附在发送移位寄存器中的数据上。因此，若接着改变每个发送字符的位数——即向发送缓冲器装不同字长的字——并不会影响已经在发送移位寄存器中的字符。

TMS9903 实际上支持三种工作方式：异步、同步和 SDLC/HDLC。异步方式适用于 RS-232C 和 RS-449EIA 标准规约。同步方式可基本实现 IBM 的标准单同步或双同步规约。TMS9903 可在点到点的 SDLC 或 HDLC 系统中运行，也适用于 SDLC 环形方式。

TMS9903 引脚安排

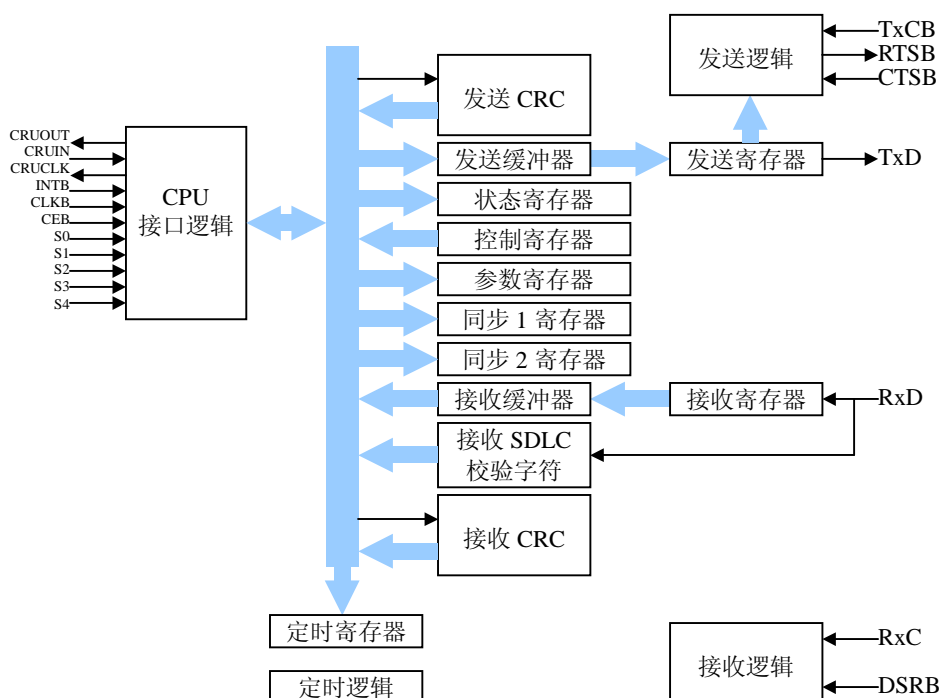
引脚图

INTB	<--	1	20	---	VCC (+5V)
TxD	<--	2	19	<--	CEB
RxD	-->	3	18	<--	CLKB
CRUIN	<--	4	17	<--	CRUCLK
RTSB	<--	5	16	<--	S0 (MSB)
CTSB	-->	6	TMS9903 15	<--	S1
DSRB	-->	7	14	<--	S2
CRUOUT	-->	8	13	<--	S3
VSS (GND)	---	9	12	<--	S4 (LSB)
TxCB	-->	10	11	<--	RxC

引脚说明

引脚	引脚名称	类型	说明
CRUIN	至 CPU 的数据输出	输出	使用方法和 TMS9901 完全相同
CRUOUT	由 CPU 来的数据输入	输入	使用方法和 TMS9901 完全相同
CRUCLK	CRU 数据选通	输入	使用方法和 TMS9901 完全相同
CEB	器件使能	输入	使用方法和 TMS9901 完全相同
S0~S4	CRU 位地址	输入	使用方法和 TMS9901 完全相同
CLKB	同步时钟	输入	通常是 TMS9904 CLK3 的反相信号，控制数据传输的速率
DSRB	数据设备就绪显示	输入	通用输入信号，其电平表示在状态寄存器位 27 内。可由程序安排由低向高跳变时产生中断请求，但是本身不参与开启发送或接收逻辑的操作
RTSB	请求发送指示	输出	当发送逻辑开启时，该线输出低电平
CTSB	清除发送指示	输入	当接收到该线的低电平后，开始发送数据
RxD	串行数据输入	输入	串行数据在该线输入
RxC	串行数据输入时钟	输入	数据在该线从低向高转换时被采样
TxD	串行数据输出	输出	串行数据在该线输出
TxCB	串行数据输出时钟	输入	数据在该线从高向低转换时发送出去
INTB	至 CPU 的中断请求	输出	
VCC, VSS	电源和参考地		

TMS9902 功能逻辑图



TMS9903 控制寄存器和状态寄存器

TMS9903 同步通信控制器控制寄存器位的分配					
位号	方式			功能	
	异步	同步	SDLC	写入 1	写入 0
31	•	•	•	器件复位	
30	•	•	•	清除发送器 (禁止中断)	清除接收器 (禁止中断)
29	•	•	•	清除发送 CRC 寄存器 (复位为 0)	清除接收 CRC 寄存器 (复位为 0)
28		•		清除接收到的同步 1 字符 (仅限于双同步方式)	
			•	禁止发送逻辑的零位插入	
27		•	•	将 CRU 位 0~9 的数据装入同步 2 寄存器	
26		•		将 CRU 位 0~9 的数据装入同步 1 寄存器 (仅适用于使用同步 1 寄存器的同步方式)	
			•	将 CRU 位 0~15 位读收到的校验字符	状态寄存器 CRU 位 13 复位 (校验字符缓冲器满)。若校验字符溢出, 则 CRU 位 12 复位; 若零插入检测错, 则 CRU 位 10 复位
25	•	•	•	将输出到 CRU 位 0~8 的数据装入发送缓冲器, 更新发送 CRC。通过 CRU 位 0~15 选择要读出的发送 CRC	状态寄存器位 22 和 17 复位
24	•	•	•	用下一个输出到 CRU 位 0~9 的数据更新发送 CRC, 读 CRU 位 0~15 的发送 CRC	

TMS9903 同步通信控制器控制寄存器位的分配				
位号	方式			功能
	异步	同步	SDLC	写入 1 / 写入 0
23	•			脱空期间发送断开（低电平输出）。在将新数据装入发送缓冲器之前将此位复位为 0，以结束脱空
		•		指明脱空方式的脱空选择
			•	脱空后发送同步 2 寄存器内容（HDLC 异常终止一般为 7F16）
		•	•	脱空后异常终止发送，并置状态寄存器位 23（仅限一般情况）
22		•	•	开启异常终止中断，置状态寄存器位 23 和 18（仅限一般情况）
21	•	•	•	开启数据组交换中断，使状态寄存器位 29 和 20 复位
20	•	•	•	开启定时器中断，使状态寄存器位 25、24 和 19 复位
19	•	•	•	开启发送缓冲器空中断
18	•	•		开启接收缓冲器满中断，使状态寄存器位 21 和 11 复位
			•	开启接收缓冲器满、接收校验字符缓冲器满和接收异常终止中断，使状态寄存器位 21、14、11 和 9 复位
17	•	•	•	通过 RTSB 输出反相信号，禁止自动 RTSB 控制逻辑
16	•	•	•	开启发送器逻辑
15	•	•	•	测试方式
14	•	•	•	将 CRU 位 0~11 的数据装入控制寄存器
13	•	•	•	将 CRU 位 0~7 的数据装入定时器逻辑
12	•	•	•	用下一个输出到 CRU 位 0~9 的数据更新接收 CRC，读 CRU 位 0~15 的接收 CRC

当向 CRU 位 31 写入 0 或 1 时，整个器件复位。所有中断被禁止，所有标志和寄存器选择位复位为 0（控制寄存器位 14 和状态寄存器 22 例外，它们被置 1），这就使第一个数据要装入参数寄存器，同时状态寄存器处于发送缓冲区空状态。TMS9903 复位和初始化需要进行下列步骤：1) 向 CRU 位 31 写 1 或 0，使整个器件复位，并开启参数寄存器；2) 装填参数寄存器（CRU 位 0~11），以确定操作方式和传送方式；3) 向 CRU 位 30 和 29 写 11，使发送器和发送器 CRC 逻辑初始化；4) 向 CRU 位 30 和 29 写 00，使接收器和接收 CRC 逻辑初始化。

TMS9903 读出时的寄存器寻址				
控制寄存器				CRU 位寻址的寄存器
26	25	24	12	
1	0	0	0	CRU 位 31~16 由状态寄存器来；CRU 位 15~0 由接收 SDLC 帧的 CRC 来
0	1	0	0	CRU 位 31~16 由状态寄存器来；CRU 位 15~0 由发送信息计算的 CRC 来
0	0	1	0	CRU 位 31~16 由状态寄存器来；CRU 位 15~0 由发送信息计算的 CRC 来
0	0	0	1	CRU 位 31~16 由状态寄存器来；CRU 位 15~0 由接收信息计算的 CRC 来
0	0	0	0	CRU 位 31~9 由状态寄存器来；CRU 位 15~0 由接收缓冲器来

值得注意的是，在 SDLC 状态用户可以读两个循环冗余校验字符：第一个字符是在程序控制下由相应接收帧的接收逻辑算出的；第二个字符是在帧的末尾接收到的。信息段的最后的 16 位是接收到的循环冗余字符。要读出接收到的循环冗余字符，就要使控制寄存器位 26 置为 1。要读出通过相应接收帧算出的循环冗余字符，就要使控制寄存器的位 12 置为 1。若收到的是有效信息，这两个循环冗余字符就是相同的。在同步和异步状态中，没有指定的信息结束符，而是将接收到的数据流中的某控制字理解为信息结束符。此时，前面收到的两个数据字符被理解为接收到循环冗余字符。用户的程序必须将这两个数据字符（被理解为接收到的 CRC 字符 以及计算出来的校验字符）相比较。后者是在控制寄存器位 12 置 1 之后，从接收逻辑读出的。

TMS9903 写入时的寄存器寻址							
控制寄存器							CRU 位 11~0 寻址的寄存器 (CRU 位 31~12 总是寻址控制寄存器)
27	26	25	24	14	13	12	
1	0	0	0	0	0	0	同步 2 寄存器 (最多 10 位)
0	1	0	0	0	0	0	同步 1 寄存器 (最多 10 位)
0	0	1	0	0	0	0	发送缓冲寄存器 (最多 9 位)
0	0	1	1	0	0	0	发送 CRC 寄存器 (最多 10 位)
0	0	0	0	1	0	0	参数寄存器 (12 位)
0	0	0	0	0	1	0	定时寄存器 (8 位)
0	0	0	0	0	0	1	接收 CRC 寄存器 (最多 10 位)
0	0	0	0	0	0	0	发送缓冲寄存器 (最多 9 位)

当向参数寄存器最高位 (位 11) 写入时, 控制寄存器位 14 被自动复位。当用户对控制寄存器中的任何其他寄存器选择位置 1 时, 它便保持置位状态直到用户使它复位为止。若参数寄存器选择位 (控制寄存器位 14) 被置位, 而用户又希望写入另一个可寻址单元时, 则必须在置另一个选择位为 1 时使控制寄存器位 14 为 0。在 SDLC 方式用户只能写入同步 2 寄存器。仅在同步方式下才能写入同步 1 寄存器——而且只有在使用同步 1 寄存器的各种同步方式下才能写入。

直接器件控制位包括发送器控制和接收器控制。

首先是发送器, 在发送器清零后, 置控制寄存器位 16 为 1, 从而开启发送逻辑。发送逻辑保持开启状态, 直至用户使该位复位为 0。在发送一个字符的过程中发送逻辑不会自行禁止。若正在发送一个字符中途向控制寄存器位 16 写 0, 则该字符仍继续发送, 只有在发送完毕后, 发送逻辑才被禁止。在复位之后, 若不向控制寄存器位 17 写入, 则 RTSB 输出信号电平便自动地由发送器逻辑控制。一旦向控制寄存器位 16 写 1 从而开启发送器逻辑, RTSB 即输出低电平。RTSB 维持低电平, 直至向控制寄存器位 16 写 0 从而禁止发送器逻辑。但是, 只要用户向控制寄存器位 17 写了信号, 就会立即禁止 RTSB 输出电平的自动控制。这时, RTSB 输出电平就成为控制寄存器位 17 的反相状态。

有两种方法可将被发送的字符包括在循环冗余字符计算之中。若通过使控制寄存器位 25 置 1 来选择发送缓冲器, 则写入发送缓冲器的字符就被包括在发送循环字符的计算之中。若将发送缓冲器选作缺席写单元 (即控制寄存器中没有地址位被置 1), 则在发送循环冗余字符计算中不会包含要写入发送缓冲器的字符, 除非使控制寄存器位 24 置为 1, 然后向发送 CRC 逻辑输出该字符。这就是说, 利用控制寄存器位 24 就可以向发送缓冲器或发送 CRC 逻辑写入, 但不能同时向两者写入。发送循环冗余字符的计算如果包含长串连接的字符, 则使用控制寄存器位 25。如果发送循环冗余字符的计算中要有选择地包含和排除某些字符, 则使用控制寄存器位 24。

没有同发送器开启位 (控制寄存器位 16) 类似的接收器开启控制位。一旦接收逻辑清零, 接收逻辑就被开启, 并开始对经由 Rx D 而到达的数据采样。当每个字符被组装完毕后, 即传送给接收缓冲器。若接收循环冗余字符的计算中要包括接收到的字符, 则程序逻辑必须在从接收缓冲器中读出该字符之后将它输送给接收 CRC 逻辑。当对控制寄存器位 12 置 1 时, 送到 CRC 位 0~9 的数据将传达给接收 CRC 逻辑。

CRC 逻辑不一定要连接发送器或接收器。循环冗余计算寄存器可以脱离发送逻辑或接收逻辑而使用。

测试方式位 (控制寄存器位 15) 在通常情况下被复位为 0。若置此位为 1, 则发生下列动作:

- (1) Tx D 连接 Rx D;
- (2) RTSB 连接 CTSB, DSRB 维持低电平;
- (3) Tx CB 和 Rx C 都同定时器逻辑时钟相连, 它运行的速率是常规速率的 32 倍。

在 TMS9903 中定时器决定了测试方式中的接收与发送数据的速率。

在异步方式下, 当置控制寄存器位 23 为 1 时, 一出现脱空, 发送逻辑就将在 Tx D 上输出一个连续的低电平 (断开) 信号。需要注意的是, 若控制寄存器位 23 为 1 后, 若不出现脱空, 则不出现异常。一旦出现脱空, 则在置控制寄存器位 23 为 0 而结束断开之前, 不能将新的数据写入发送缓冲器。若控制寄存器位 23 复位为 0, 则在脱空之后, Tx D 输出连续的高电平信号。任何时候都可通过向发送缓冲器装入数据来重新开始发送——此时, Tx D 上的高电平输出结束, 而按照参数寄存器中所规定的异步规约选择来发送下一个字符。

TMS9903 同步通信控制器状态寄存器位的分配					
位号	方式			功能	复位条件
	异步	同步	SDLC		
31	•	•	•	1=中断挂起	无中断挂起
30	•	•	•	1=1 个或多个寄存器装入控制标志置位	无控制标志置位
29	•	•	•	1=DSRB 或 CTSB 或自动 RTSB 信号电平改变	输出到 CRU 位 21
28	•	•	•	CTSB 输入反相	
27	•	•	•	DSRB 输入反相	
26	•	•	•	自动控制下的 RTSB 电平。若 RTSB 在程序控制下，则发送器处于有效状态	
25	•	•	•	1=定时器减到 0	输出到 CRU 位 30
24	•	•	•	1=定时器出错。当定时器减到 0，位 25 已为 1	输出到 CRU 位 20
23		•	•	1=脱空后异常终止	输出到 CRU 位 22
22	•	•	•	1=发送缓冲器空	输出 0 到 CRU 位 25
21	•	•	•	1=接收缓冲器满	输出到 CRU 位 18
20	•	•	•	1=中断请求伴随 RTSB、DSRB 或 CTSB 信号电平改变（位 29=1）	位 29=0 或输出到位 21
19	•	•	•	1=中断请求伴随定时器倒数到 0（位 25=1）	位 25=0 或输出到位 20
18		•	•	1=中断请求伴随异常终止（位 23=1）	位 23=0 或输出到位 22
17	•	•	•	1=中断请求伴随发送缓冲器空	输出 0 到 CRU 位 25
16	•			1=中断请求伴随接收缓冲器满（位 21=1）	无有效中断条件
		•		1=中断请求伴随接收缓冲器满（位 21=1）	
			•	1=中断请求伴随接收缓冲器满（位 21=1）或收到异常终止（位 14=1）或收到闭标志和收到的校验字符准备读出（位 13=1）	
15	•	•	•	RxD 输入电平	
14	•			1=检测到起始位	收到终止位
			•	1=收到异常结束	输出到 CRU 位 18
13	•			1=收到第一个字符数据位	收到终止位
			•	1=收到关标志，可读出校验字符	输出 0 到 位 26
12	•			1=检测出接收帧出错	收到无错字符
			•	1=检测出溢出错误，即接收数据溢出前一帧的校验字符	输出到 位 26
11	•	•	•	1=检测出接收溢出错误	输出到 位 18
10	•	•		1=检测出接收奇偶校验错	收到有效字符
			•	1=检测出零插入错	输出到 位 26
9	•	•		1=最新收到的字符中任何接收错误	收到有效字符
			•	1=检测出标志	输出到 位 18

在 SDLC 方式下，接收零插入错意指接收到了五个连续的 1，后面跟一个标志字符，两者之间没有插入应有的 0。SDLC 方式下还可能出现一种反常而有趣的错误，若在读出前一帧的循环冗余校验字之前，若接收到新的一帧数据，则出现接收 CRC 溢出错误，将使状态位 12 置位。

参数寄存器

TMS9903 同步通信控制器参数寄存器功能				
位号	方式			说明
	异步	同步	SDLC	
11	•	•	•	0, 以输入钟速发送/接收; 1, 以输入钟速÷32, 发送接收; 采用补零 NRZI 编码
10~9	•	•	•	00, CRC-16($X^{16}+X^{15}+X^2+1$); 01, CRCC-12($X^{12}+X^{11}+X^3+X^2+X+1$); 10, 修正 CRCC -16($X^{16}+X^{14}+X+1$); 11, CRC-CCIT($X^{16}+X^{12}+X^5+1$)
8~6	•	•	•	000, 一般同步; 001, SDLC; 010, 单同步; 011, 双同步; 100, 未使用; 101, 异步, 两停止位; 110, 异步, 一停止位; 111, 未使用
5~4	•	•	•	00, 无奇偶校验; 01, 无奇偶校验; 10, 偶校验; 11, 奇校验
			•	00, 点到点; 01, 回路主站; 10, 回路从站-备用; 11, 回路从站-现役
3	•	•	•	0, CLKB 除以 3 产生定时器时钟; 1, CLKB 除以 4 产生定时器时钟
2~0	•	•	•	000, 5 位/字符; 001, 6 位/字符; 010, 7 位/字符; 011, 8 位/字符; 100, 9 位/字符

有三种同步方式选择和一种 SDLC 方式选择。这四种选择共用同步字符逻辑。

TMS9903 同步与 SDLC 方式同步字符与脱空选择						
参数寄存器 CRU 位号			方式	同步字符	脱空填充字符	
					控制寄存器 CRU 位 23=0	控制寄存器 CRU 位 23=1
8	7	6				
0	0	0	同步-一般	无	异常终止	[同步 2 寄存器]
0	0	1	SDLC	7E	异常终止	[同步 2 寄存器]
0	1	0	同步-单同步	[同步 1 寄存器]	[同步 2 寄存器]	[同步 2 寄存器]
0	1	1	同步-双同步	[同步 1 寄存器]- [同步 2 寄存器]	[同步 1 寄存器]- [同步 1 寄存器]	[同步 2 寄存器]- [同步 1 寄存器]

SDLC 发送逻辑: 每帧必须以标志字符开始。7E 总是作为开头的同步字符。之后就置控制寄存器位 23 为 0, 因为 SDLC 帧内不允许脱空。如果出现了脱空, 则发送器将异常终止, 输出连续的高电平信号, 并置相应的状态位。为了发送一个有效的信息结束字符, 必须读出计算的发送校验字 (通过控制寄存器位 23 或 25 选择), 置控制寄存器位 23 为 1, 将标志字符 (7E) 装入同步 2 寄存器, 将算出的发送校验字符作为两个数据字节输出。然后, 允许脱空出现。当脱空出现时, 同步 2 寄存器的内容被输出。由于同步 2 含有标志字符, 故通过发送信息校验字符和关标志来结束该帧, 如 SDLC 规约所要求的那样。除此之外, 还有一种办法可以结束帧的传输。不用允许脱空和从同步 2 寄存器输出帧的关标志, 就可以禁止 SDLC 的 0 插入, 其方法是: 先向控制寄存器位 28 写 1, 然后将关标志作为简单的一列 8 位数据字符输出。

如果用 HDLC 规约运行 TMS9903, 则必须输出 7F 作为异常终止字符。为了在发送脱空之后获得有效的 HDLC 异常终止, 必须向同步 2 寄存器写 HDLC 异常终止字符, 然后, 在帧传输过程中使控制寄存器位 23 置 1。若出现脱空, 则从同步 2 寄存器输出 HDLC 异常终止字符。

在检测到新的帧时, SDLC 接收逻辑按照标志字符 7E (它也是专用的同步字符) 自行同步。所以, 上表所示的控制寄存器位 23 的设定以及脱空填充字符选择项不适用于这一情况。当接收逻辑检测出另一个标志字符时, 它就认为已收到帧的关标志, SDLC 接收逻辑也能检测出异常终止。如果选通, 它就置相应的状态标志和产生中断请求。

表中所列的三种同步方式及其脱空填充字符选择项, 使用户可以在程序的控制下选用常见的任何一种同步规约方案。

外同步在信息首部不使用引导同步字符。用户可以通过选择一般的同步方案来实现这一规约。对于 TMS9903 发送逻辑, 在选通发送器之前一定要保证 CTSB 是低电平。一旦 RTSB 转为低电平, 就开始信息传输。接收该发送信息的装置可以将此 RTSB 输出用作外同步输入。TMS9903 接收逻辑将用 DSRB 调制解调输入作为它的外部同步信号。向 TMS9903 发送信号的装置在它开始传输信息之前必须产生低电平的 DSRB 输入, 控制 TMS9903 接收逻辑的程序必须通过查询相应的状态寄存器位来检测低电平的 DSRB 输入, 一旦检测到 DSRB 有效, 即开始接收。

单同步方式在发送信息的首部输出同步 1 寄存器的内容,并通过将接收字符跟同步 1 寄存器的内容相符时,接收逻辑就认为检测到了新信息。当出现脱空时,输出一个同步字符。将适当的字符装入同步 2 寄存器用户便可以发送和检测任何脱空填充字符。

双同步方式在发送信息的首部输出两次同步 1 寄存器的内容。当两个相连的字符跟同步 1 寄存器的内容相符时,接收逻辑就认为检测到了新信息。在某些情况下,脱空之后要发送两个同步字符,但按照标准的双同步规约,脱空之后要输出 DLE-SYN 字符的组合。当控制寄存器位 23 等于 0 时,TMS9903 将从同步 1 寄存器输出两个同步字符。为了满足双同步规约的要求,得将 DLE 字符装入同步 2 寄存器,将 SYN 字符装入同步 1 寄存器,使控制寄存器位 23 置 1。其他双同步逻辑(特别是专用控制字符组合的产生与检测)都必须由监控程序处理。

控制寄存器位 28 为上表所列的方案增加了一定的灵活性。但是,这个控制位只适用于 SDLC 和双同步方式。在 SDLC 状态,当控制寄存器位 28 复位为 0 时,TMS9903 发送逻辑在每发送五个相连的 1 之后就插入一个 0。只要置控制位 28 为 1,就禁止了 SDLC 方式下的零插入。在双同步方式下,当控制寄存器位 28 置 1 时,收到的与同步 1 寄存器的内容相符的接收字符即被废弃,这就使用户能够剔除收到的脱空同步字符。

参数寄存器位 5 和位 4 对同步方式和 SDLC 方式起着不同的作用。在同步和异步方式下,参数寄存器的位 5 和位 4 用于指明奇偶校验的有无和类型。如有奇偶校验,将自动地为发送的数据字符产生奇偶校验位,为收到的数据检验奇偶校验位。但奇偶校验不适用于同步 1 和同步 2 寄存器的内容。如果用户想发送带奇偶校验的同步字符,则必须给这些寄存器的内容增加奇偶校验位,每个同步寄存器都是 10 位宽,所以用户可以向能指定的最前字符(9 位)增加一个奇偶校验位。由于接收逻辑将所有接收字符看作数据,故接收逻辑将自动地校验所收到的同步字符的奇偶性。在 SDLC 方式,参数寄存器位 5 和位 4 用来指明回路或非回路方式,实际上它们指明了 EOP 字符(7F)的处理方式。在点到点的配置形态下,EOP 字符无关紧要,从而被忽略。作为回路主站,发送逻辑不管 EOP 字符,但是,接收逻辑将 EOP 字符当作帧的关标志。这是需要的,因为回路主站环绕回路发送的查询 EOP 字符最终要被作为回路从站发送的最后一帧的关标志。对不发送数据而能接收数据的 SDLC 回路从站,选择了回路从站待命方式。与此相反,SDLC 回路中要向主站发送的从站,则要选择回路从站现役方式。在回路从站待命方式下,TMS9903 就在重发被接收的数据之前插入一位延迟。只要不在电气上使从站脱开 SDLC 回路,待命从站方式就照管在线的下一个回路从站的定时和规约需求,但是,若要从 SDLC 回路电气脱开了 TMS9903 从站,则要求外部逻辑使上游数据绕过电气脱开的从站。当它电气上连通时又断开旁路将从站纳入。如果 TMS9903 在 SDLC 回路中起从站作用,那么通常就使它工作于回路从站现役方式。仅在从站刚进入回路尚未同步(未收到 EOP)时,才转换到回路从站待命方式。在回路从站现役方式,TMS9903 接收逻辑将寻找下一个 EOP 字符。一旦接收到 EOP 字符,它将会将此字符转换为一个标志,此标志将成为该站发送给主站信息帧的开标志。只要 TMS9903 在回路从站现役方式工作,它就要继续捕获接收 EOP 字符,并在此之后发送信息帧。当 TMS9903 没有帧要继续发送时,就应当使它保持回路从站现役方式,同时,通过使控制寄存器位 16 复位为 0 的办法来关闭发送器。

参数寄存器位 0、1 和 2 在有奇偶校验时,指明的数据位数并不包括奇偶校验位在内。若同步字符和控制字符是 8 位宽,则在同步方式不能指定少于 8 位的字符。这是因为,在接收同步或控制字符时,接收逻辑不能自动从每个指定位数的字符转换成 8 位字符。而且,程序控制接收逻辑不能进行这种转换,因为它要直到字符进入接收缓冲器之后才知道已收到同步或专用字符,但此时已太晚了,已不能进行变换了。

参数寄存器位 11 使用户可以按发送和接收时钟速率(或以其 1/32 的速率)来发送和接收数据。在一般情况下,这是一种标准的同步方式选择。就 TMS9903 来说,它在所有方式(同步、异步及 SDLC)都是可用的。此位在作为 SDLC 回路从站工作过程中应复位为 0。在同步或 SDLC 工作过程中,若每隔 32 个时钟脉冲(参数寄存器位 11 为 1)就对数据采样,则采取串行数据的 NRZI 编码和解码。即是说,数据信号改变状态用以发送一个 0,或者保持原状态不变,用以表示一个 1。

参数寄存器位 9 和 10 用来指定循环冗余算法,(这种算法用于发送和接收逻辑)。CRC-16 是用于同步和异步规约的正规算法。CRC-12 是用于有 6 位字符的同步和异步规约的算法。修正的 CRC-16 常用于标准双同步规约。CRC-CCIT 是标准的 SDLC 算法。

TMS9903 中断逻辑

TMS9903 内产生中断请求的条件有七个，其中三个结合起来产生一个中断请求状态。所以，七个中断产生条件有五种中断请求状态，如下所示：

TMS9903 中断请求状态表			
状态寄存器 条件位号	控制寄存器 中断选通位号	状态寄存器 中断位号	中断
29	21	20	DSRB、CTSB 或自动 RTSB 电平变更
25	20	19	定时器时间到
22	19	17	发送缓冲器空
23	22	18	发送异常结束
接收缓冲器满 21	18	16	接收中断
SDLC 帧结束 13			
接收异常结束 14			

TMS9903 没有中断请求优先权仲裁逻辑。在出现一个或几个能请求中断的条件时，若中断已被选通，则 INTB 输出低电平，状态寄存器位 31 置 1。响应 TMS9903 中断请求的中断服务子程序此时必须查询状态寄存器位，以便确定哪个中断请求是有效的。程序逻辑负责所有的中断优先权仲裁。通常适用于串行 I/O 设备的中断优先权有：

- (1) 最高优先级。接收缓冲器满（状态寄存器位 16 和 21 置位）。
- (2) 发送缓冲器空（状态寄存器位 17 和 22 置位）。
- (3) 调制解调器电平信号变化（状态位 20 和 29 置位）。
- (4) 检测出接收异常终止信号（状态寄存器位 16 和 14 置位）。
- (5) 发送器异常终止（状态寄存器 18 和 23 置位）。
- (6) 检测出 SDLC 帧结束字符（状态寄存器位 16 和 13 置位）。
- (7) 最低优先级。定时器中断（状态寄存器位 19 和 25 置位）。

TMS9903 初始化程序逻辑

任何 TMS9903 操作。第一步通常都是使设备初始化。下面是必需的步骤：

- (1) 向控制寄存器位 31 写 0 或 1，使设备复位。
- (2) 输出相应的参数寄存器设定值。
- (3) 向控制寄存器位 18~22 输出数据，以选通相应的中断。
- (4) 在同步和 SDLC 方式时，将适当的代码装入同步 2 和/或同步 1 寄存器，这两个寄存器在异步方式中不用。
- (5) 使接收逻辑初始化，向控制寄存器位 30 写 0。若用循环冗余，则通过向控制寄存器位 29 写 0 来预置接收 CRC 逻辑。这些步骤完成后，接收逻辑开始工作，组装接收数据。
- (6) 向控制寄存器位 30 写 1 来初始化发送逻辑。若用循环冗余，则向控制寄存器位 29 写 1 来预置发送 CRC 逻辑。此时，发送逻辑已初始化，但尚未开启。
- (7) 等到用户向控制寄存器位 16 写 1 从而开启了发送器时，发送逻辑才开始工作。在开启发送器时，用户还应将数据装入发送缓冲器。

TMS9903 异步操作

选择异步方式时，数据要跟奇偶校验位（选用的话）一起发送，外加一个终止位（按参数寄存器的设定）。每当发送缓冲器变空时，若发送缓冲器中断被开启，就产生中断请求，同时，使相应的状态位置 1。用户有一个字符的时间，在这个时间内，通过输出另一个字符来实现响应，否则就出现脱空。脱空之后，将输出连续的高（传号）信号，或连续的低（断开）信号，这取决于控制寄存器位 23 的设置。

在开始一个接收操作时，要对起始位检测状态（状态寄存器位 14）采样，以识别新接收信息的首部。该状态不能产生中断请求。为了处理接收的字符，要使用接收缓冲器满中断请求逻辑。当字符被接收后，程序逻辑必须在一个字符时间内从接收缓冲器读出字符，并通过同时读出错状态寄存器的各位来校验任何一个异步接收错误状态。接收数据和状态可一起读出，方法是从 TMS9903 读 CRU 位 0~15。

不存在与 TMS9903 的异步操作有关的特殊程序设计因素。反之，监控程序逻辑必须满足任何其他规约的要求。

TMS9903 同步操作

大部分与单同步（Monosync）和双同步（Bisync）规约有关的逻辑必须由用来控制 TMS9903 发送与接收操作的监控程序提供。唯一由 TMS9903 本身提供的逻辑功能是各种同步寄存器的可编程选择，出错及正常状态指示，以及字符长度定义。

有关同步字符的选用问题，可参见前面关于参数寄存器的讨论。有关所指示的状态的介绍，可参见状态寄存器和中断逻辑的讨论。

TMS9903 SDLC 操作

在 SDLC 规约中，每一帧的第一个字节是地址字段，第二个字节是控制字段。在 HDLC 规约中，地址字段可任意长，而控制字段可以是一字节长或二字节长。SDLC 规约的某些变态在控制字段后面插入了一个逻辑控制字段。这种逻辑控制字段可任意长。地址字段、控制字段和逻辑字段字符都是八位宽。信息段字符可以是任意数据位宽。在多字节地址或逻辑控制字段中的字节数目，可通过检查特定的字符位来决定。例如，某个规约可能规定：地址字段的最后一个字节在低位里有 1；而所有前面的字节低位都是 0。

TMS9903 没有用来处理地址字段、控制字段或逻辑控制字段的片上逻辑。设备程序设计可指定每个字符的位数——这种位数的指定可从一个字符变到另一个字符。此外就无其他特长了。而且，监控程序必须考虑主站或从站的逻辑。主站的监控程序必须发送从站的地址，并将收到的地址理解为标志着信息帧的来源。从站的监控程序必须在每帧的首部发送自己的地址，并检查接收帧首部的地址，以便判定该帧的其余部分是否应该读入或忽略不管。

在收到控制字段（或逻辑控制字段）的最后一个字节时，程序逻辑必须先改变参数寄存器中每字位数的规定，然后处理信息段的第一个字符（该信息字段将使用不同的字符长度）。由于参数寄存器中每字符的位数规定只适用于接收器逻辑；用户通过输送给发送缓冲器的位数就可指明发送字符的长度。这样，如果规约需要从一种字符长度变为另一种字符长度就很简单了。接收的字符跟同时发送的字符也可具有不同的位数。

在 SDLC 方式下的 TMS9903 接收逻辑具有一种很有用的帧结束能力：可将收到的校验字（在该帧的关标志之前，必须是 16 个数据位）自动装入接收 CRC 寄存器。微处理器可读出这个收到的校验字，并拿它同算出的校验字相比较。这个接收 CRC 逻辑还有一些额外的用出。一个有效的 SDLC 帧在开标志和关标志之间必须至少有 32 位，其中包括一个 8 位的控制字段和一个 16 位循环冗余校验字。往往为了传输一个无相应数据的命令或响应，需要发送和接收 32 位信息帧。在这样短的帧内出现的错误可导致复杂的逻辑问题。由于短帧的终止标志可能未被察觉，所以可能难以辨认下一帧的起始和终止标志。但用户可用 TMS9903 的接

收 CRC 逻辑来识别短接收帧。若用户没有收到一个状态指示字（指明接收的校验字符是有效的），则可以认为已收到了一个短帧。

SDLC 规约规定：帧之间可以用单个标志字符分开，此字符作前一帧的关标志和下一帧的开标志；也可用许多个标志字符来分隔两帧。这两种方式都可被 TMS9903 采用。

在介绍参数寄存器的时候已经解释了在脱空之后如何由同步 2 寄存器产生帧关标志，或者禁止零插入，并且将标志字符当作数据来传送。由于用户可以分别指定循环冗余校验计算中包含或不包含某字符，所以对 TMS9903 来说，处理非数据字符（就把它们当作数据字符一样对待）并不困难。用户可以用任何一种方法（以一个或几个标志字符分隔信息帧）来终止一个信息帧。

如果用户用脱空逻辑来产生帧关闭，则在单个标志字符发送时装入下一帧的地址段首字节，这样就会使一个标志能够分开两帧。如果用户使脱空拖延的更久，则继续输出标志字符，直到下一帧的地址段首字节作为数据写入发送缓冲器而开始另一帧为止。如果用户想把标志当作插入零的数据来发送，则用来分隔帧的标志数目严格地说就是一个程序控制的函数，但必须保证不出现脱空。

SDLC 回路中的主站没有特殊的编程要求。若在发送帧关标志的末尾产生异常终止，则标志末尾的 0 位和异常终止的前七位的 1 一起可组成 EOP 字符（环绕该回路传送 EOP 字符就可查询从站）。当该 EOP 字符送回到主站接收逻辑时。它就被当成关标志。

SDLC 回路中的从站在跟回路同步化之前（也就是说，直到接收了 EOP 字符并开始用一个“位延迟”转发为止），应运行于 SDLC 回路的从站待命方式。然后，从站工作方式改变为 SDLC 回路从站的现役方式。在现役条件下，从站查找来自 RxD 的下一个 EOP 字符。若发送器已开启，则 TMS9903 把这个接收到的 EOP 字符转换为它打算发送的信息帧的开标志。控制 SDLC 从站的程序可以用关标志后接异常终止符来结束发送，也可用 EOP 字符结束发送。关标志后接异常终止符，这将产生一个 EOP 字符标志或传送给下方的从站，并在帧之间倍增标志字。关 EOP 字符由下方的相继从站转变为标志或传送给主站，主站把接收逻辑上的 EOP 字符理解为关标志。所以，关 EOP 字符产生一个标志字，用以分隔两个发送帧。

对发送和接收操作中可能出现的正常状态和出错状态的讨论参见前文，但应注意，完全是由监控程序按本机系统逻辑的需要来解释状态位和处理异常终止的。

TMS9903 区间定时逻辑

TMS9903 有一个区间定时器。将定时计数值装入定时寄存器便可预置区间定时器。为了选择定时寄存器作为从 CRU 位 0~7 输出数据的目标，应置控制寄存器位 13 为 1，一经将控制寄存器位 13 清零，则开启了定时器。定时器减 1 的速率取决于参数寄存器位 3 和控制寄存器位 15。

参数寄存器位 3 使得可以对 CLKB 时钟除以 3（参数寄存器位 3=0）或除以 4（参数寄存器位 3=1）产生定时器时钟。每隔 64 个定时器时钟脉冲便进行一次定时器减 1。定时寄存器本身的内容并不更动，而是将定时寄存器的内容移入定时器逻辑，然后在那里减 1。一旦发生“超时”，状态寄存器位 25 便置 1。若定时器中断已经选通，则状态寄存器位 19 在中断请求出现时也置 1。在定时器减到 0 时定时寄存器的内容再次载入，并再次启动减 1 操作。所以装入定时寄存器的值决定了“超时”（time-out）之间的区间。在向定时寄存器装入另外的值之前，原值始终有效。

应向控制寄存器位 20 写 0 来响应“超时”。若在下次“超时”之前仍未响应，则表示定时器出错，状态寄存器位 21 置 1。定时器逻辑经常用于串行 I/O，以产生故障中断，提醒监控程序挂起或出现了自己不能识别的其他出错条件。当通过控制寄存器位 15 写 1 来选择测试方式时，定时器以常规速率的 32 倍运行。这可加速定时器的测试。此外，在测试方式下，定时器作为发送和接收的共同时钟。所以，用户可以对发送器和接收器的测试指定自动波特速率。

资料主要来源

- 《16 位与 32 位微计算机手册》，周继武，黄秦等编译，国防工业出版社。