INTRODUCTION

The transfer of digital data over relatively long distances is generally accomplished by sending the data in serial form thru a single communications channel using one of two general transmission techniques; asynchronous or synchronous. Synchronous data transmission requires that a clock signal be transmitted with the data in order to mark to location of the data bits for receiver. A specified clock transition (either rising or falling) marks the start of each data bit interval as shown in Figure 1. In addition, special synchronization data patterns are added to the start of the transmission in order for the receiver to locate the first bit of the message. with synchronous transmission, each data bit must follow contiguously after the sync word, since one data bit is assumed for every clock period or a fixed multiple of clock periods.

With asynchronous transmission, a clock signal is not transmitted with the data and the characters need not be contiguous, In order for the receiver to properly recover the message, the bits are grouped into data characters (generally from 5 to 8 bits in length) and synchronizing start and stop elements are added to each character as shown in Figure 2.

The start element is a single logic zero (space) data bit that is added to the front of each character. The stop element is a logic one (mark) that is added to the end of each character. The logic one (mark) level is maintained until the next data character is ready to be transmitted. (Asynchronous transmission is often referred to as start-stop transmission for obvious reasons). Although there is no upper limit to the length of the stop element, there is a lower limit that depends on the system characteristics. Typical lower limits are 1.0, 1.42 or 2.0 data bit intervals, although

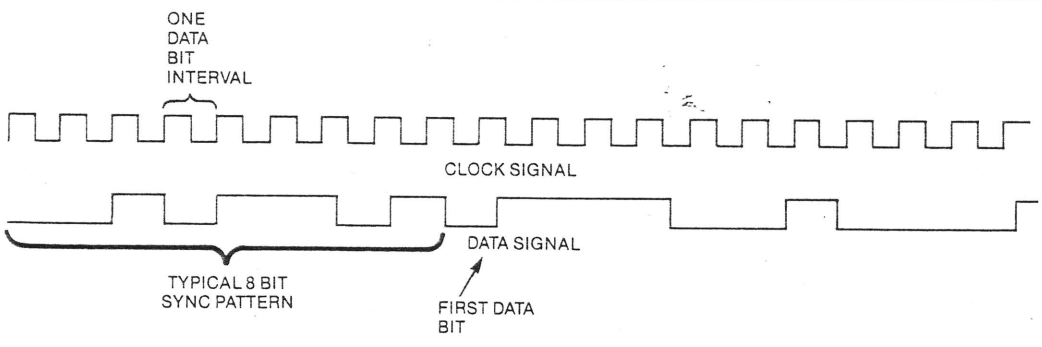

Figure 1. SYNCHRONOUS DATA

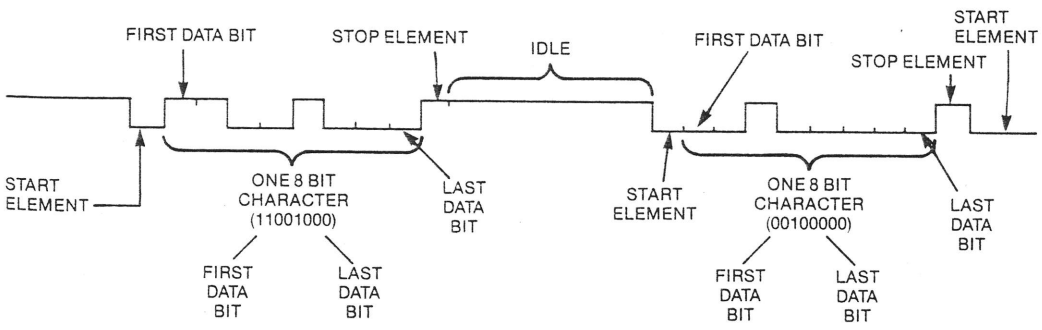

Figure 2. ASYNCHRONOUS DATA

most modern systems use 1.0 or 2.0. The negative going transition of the start element defines the location of the data bits in one character. A clock source at the receiver is reset by this transition and is used to locate the center of each data bit.

The rate at which asynchronous data is transmitted is usually measured in baud, where a baud is defined to be the reciprocal of the shortest signal element (usually one data bit interval). It is interesting to note that the variable stop bits is what makes the baud rate differ from the bit rate. For synchronous transmission, each data element is equal to the clock period therefore the baud rate equals the bit rate. The same is true for asynchronous transmission if the stop element is always one bit in duration (this is referred to as isochronous transmission). However, when the stop code is longer than one bit, as shown in Figure 3, the baud rate differs from the bit rate.

Each character in Figure 3 is 11 data bit intervals in length, and if 15 characters are transmitted per second, then the shortest signal element (one data bit interval) is 66.6 ms/11 = 6.06 ms; giving a rate of 1/6.06 ms = 165 baud. However, since only 10 bits of information (8 data bits, one start bit and 1 stop bit) are transmitted every 66.6 msec, the bit rate is 150 bit/sec. (Even though the stop element lasts for two data intervals, it still is only one bit of information.)

There are several reasons for using asynchronous transmission. The major reason is that since a clock signal need not be transmitted with the data, transmission equipment requirements are greatly simplified. (Note, however, that an independent clock source is still required at both the transmitter and receiver). Another advantage of asynchronous transmission is that characters need not be contiguous in time, but are transmitted as they become available. This is a very valuable feature when transmitting data from manual entry devices such as a keyboard. The major disadvantage of asynchronous transmission is that it requires a very large portion of the communication channel bandwidth for the synchronizing start and stop elements (a much smaller portion of the bandwidth is required for the sync words used in synchronous transmission).

Asynchronous transmission over a simple twisted wire pair can be accomplished at moderately high baud rates (10K baud or higher depending on the length of the wire, type of the line drivers, etc.) while it is generally limited to approximately 2K baud over the telephone network. Other types of asynchronous transmission can be as high as 218K baud. When operating over the telephone network, a modem is required to convert the data pulses to tones that can be transmitted through the network.

One of the major limiting factors in the speed of asynchronous transmission is the distortion of the signal elements. Distortion is defined as the time displacement between the actual signal level transmission and the nominal transition ($\Delta\dagger$), divided by the nominal data bit interval (See Figure 4).

The nominal data bit interval is equal to the reciprocal of the nominal transmission baud rate and all data transitions should ideally occur at an integer number of intervals from the negative transition of the start bit. Actual data transitions may not occur at these nominal points in time as shown in the lower waveform of Figure 4. The distortion of any bit transition is equal to $\Delta\dagger$ x NOMINAL BAUD RATE.

This distortion is generally caused by frequency jitter and frequency offset in the clock source, used to generate the actual waveform as well as transmission channel, noise, etc. Thus, the amount of distortion that can be expected on any asynchronous signal depends on the device used to generate the signal and the characteristics of the communication channel over which it was sent. Electronic signal generators can be held to less than 1% distortion while electromechanical devices (such as a teletype) typically generate up to 20% distortion. The transmission channel may typically add an additional 5% to 15% distortion.

The distortion previously described referred only to a single character as all measurements were referenced to the start element transition of that character. However, there may also be distortion between characters when operating at the maximum possible baud rate (i.e., stop elements are of minimum length).

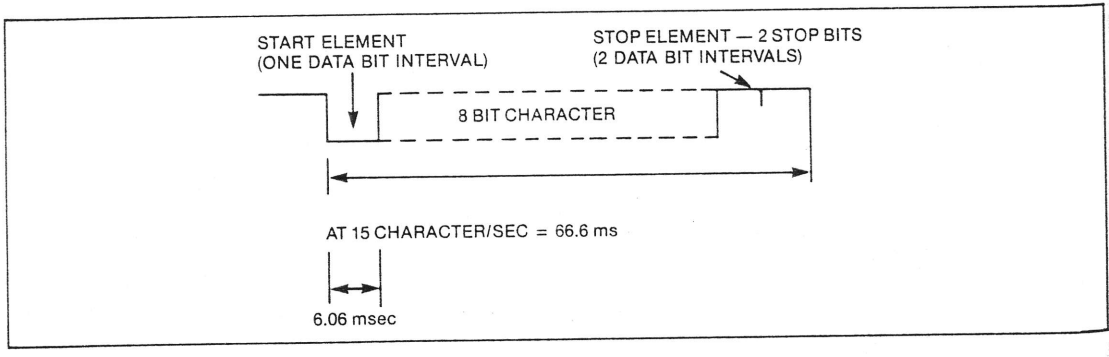

Figure 3.

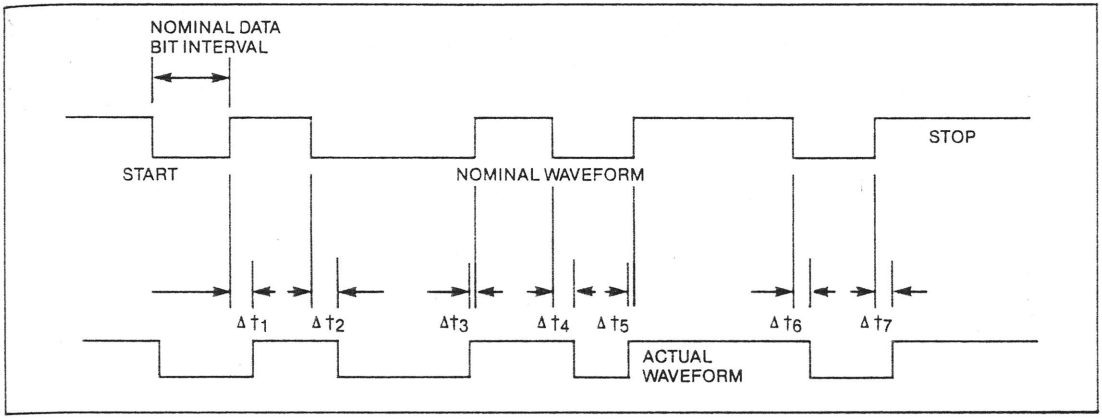

Figure 4A.

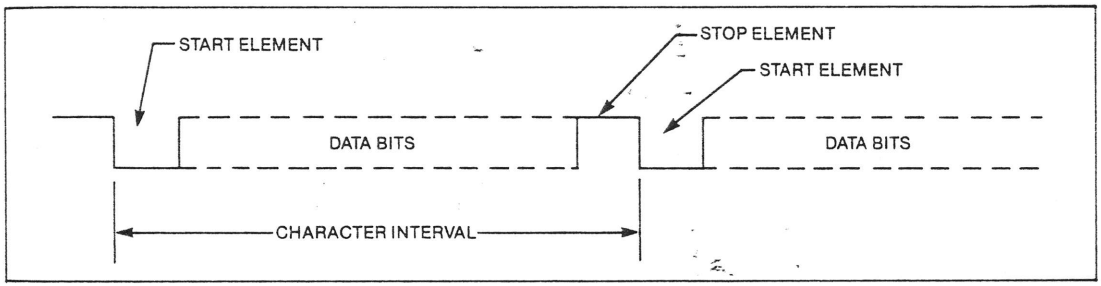

Figure 4B.

This type of distortion is usually measured by the minimum character interval as shown in Figure 4B.

The minimum character interval distortion is generally specified as the percentage of a nominal data bit interval that any character interval may be shortened from its nominal length. Since many of the same parameters that cause distortion of the data bits are also responsible for the character length distortion, the two distortions are often equal. However, some systems may exhibit character interval distortions of up to 50% of a data bit interval. This parameter is important when operating at the maximum baud rate since the receiver must be prepared to detect the next start bit transition after the minimum character interval.

Asynchronous receivers operate by locating the nominal center of the data bits as measured from the start bit negative going transition. However, due to receiver inaccuracies, the exact center may not be properly located. In electromechanical devices such as teletypes, the inaccuracy may be due to mechanical tolerances or variations in the power line frequency. With electronic receivers, the inaccuracies are due to frequency offset, jitter and resolution of the clock source used to find the bit centers. (The bit centers are located by counting clock pulses). For example, even if the receiver clock had no jitter or offset, and it was 16 times the baud rate, then the center of the bit could only be located within 1/16 of a bit interval (or 6.25%) due to clock resolution. However, by properly phasing the clock, this tolerance can be adjusted so that the sampling will always be within ±3.125% of the bit center. Thus, signals with up to 46.875% distortion could be received. This number (the allowable receiver input distortion) is often referred to as the receiver distortion margin. Electromechanical receivers have distortion margins of 25 to 30%. The receiver must also be prepared to accept a new character after the minimum character interval. Most receivers are specified to operate with a minimum character interval distortion of 50%.

TR1863/65 OPERATION

TR1863/65 is designed to transmit and receive asynchronous data as shown in Figure 5. Both the transmitter and the receiver are in one MOS CHIP, packaged in a 40 lead ceramic DIP. The array is capable of full duplex (simultaneous transmission and reception) or half duplex operation.

The transmitter basically disassembles parallel data characters into a serial asynchronous data system. Control lines are included so that the characters may be 5, 6, 7 or 8 bits in length, have an even or odd parity bit, and have either one or two * stop bits. Furthermore, the baud rate can be set anywhere between DC and 218K baud (3.5 MHz clock) by providing a transmit clock at 16 times the desired baud rate.

* 1-1/2 with 5 bit code

The receiver assembles the asynchronous characters into a parallel data character by searching for the start bit of every character, finding the center of every data bit, and outputting the characters in a parallel format with the start, parity and stop bits removed. Three error flags are also provided to indicate if the parity was in error, a valid stop bit was not decoded or the last character was not unloaded by the external device before the next character was received (and therefore the last character was lost). The receiver clock is set at 16 times the transmitter baud rate.

Both the transmitter and receiver have double character buffering so that at least one complete character interval is always available for exchange of the characters with the external devices. This double buffering is especially important if the external device is

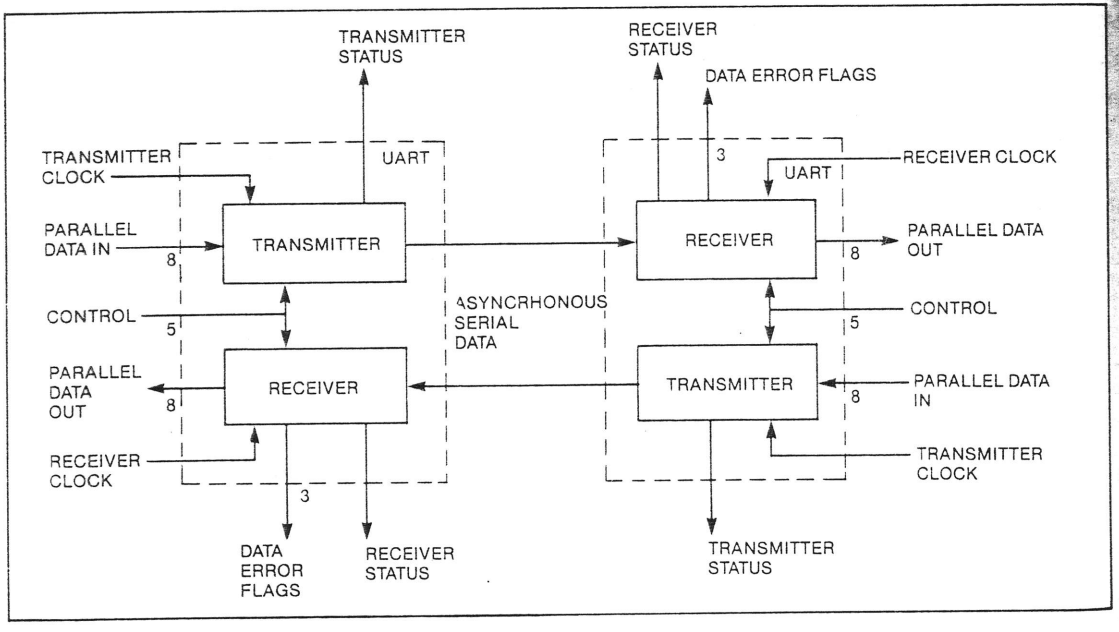

Figure 5.

a computer, since this provides a much longer permissible interrupt latency time (the time required for the computer to respond to the interrupt).

The status of the transmitter buffer and the receiver buffer (empty or full) is also provided as an output.

Another feature of the UART is that the control information can be strobed into the transmitter and receiver and stored internally. This allows a common bus from a computer to easily maintain the controls for a large number of transmitter/receivers.

The UART data and error flag outputs are designed for direct compatibility with bus organized systems. This feature is achieved by providing completely TTL compatible Three-state outputs (no external components are required). Three-state outputs may be set to a logic one or logic zero when enabled, or set to an open circuit (very high impedance) when disabled. A separate control line is provided to enable the data outputs and another one to enable the error flags so that the data outputs can be tied to a separate bus from the flag outputs.

The TR1865 has internal pullups connected to its inputs making it TTL compatible, while the TR1863 requires external pullups to be connected to its input pins.

UART DESCRIPTION

Figure 6 is a block diagram of the transmitter portion of the UART. Data can be loaded into the Transmitter Holding Register whenever the Transmitter Holding Register Empty (THRE) line is at a logic one, indicating the the Transmitter Holding Register is empty. The data is loaded in by strobing the Transmitter Holding Register Load (THRL) line to a logic zero. The data is automatically transferred to the Transmitter Register as soon as the Transmitter Register becomes empty. The desired start, stop and parity bits are then added to the data and serial transmission is started. The number of stop bits and the type of parity bit is under control of the Control Register. The state of the control lines is loaded into the Control Register when the Control Register Load (CRL) line is strobed to a logic one. The 5 control lines allow 24 different character formats as shown in Table 1. These 24 formats cover almost all of the transmission schemes presently in use.

A Master Reset (MR) input is provided which sets the transmitter to the idle state whenever this line is strobed to a logic one. In addition, a Status Flag Disconnect (SFD) line is provided. When this signal is at a logic one, the THRE output is disabled and goes to a high impedance. This allows the THRE outputs of a number of arrays to be tied to the same data bus.

Figure 7 illustrates the relative timing of the transmitter signals. After power turn-on, the master reset should be strobed to set the circuits to the idle state. The external device can then set the transmitter register data inputs to the desired value and after the data inputs are stable, the load pulse is applied. The data is then automatically transferred to the Transmitter Register where the start, stop and parity (if required) bits are added and transmission is started. This process is then repeated for each subsequent character as they become available. The only timing requirement for the external device is that the data inputs be stable during the load pulse (and 20 nsec after).

Table 1.
CONTROL DEFINITION

CONTROL WORD					CHARACTER FORMAT			
WLS2	WLS1	PI	EPE	SBS	START BIT	DATA BITS	PARITY BIT	STOP BITS
0	0	0	0	0	1	5	ODD	1
0	0	0	0	1	1	5	ODD	1.5
0	0	0	1	0	1	5	EVEN	1
0	0	0	1	1	1	5	EVEN	1.5
0	0	1	x	0	1	5	NONE	1
0	0	1	x	1	1	5	NONE	1.5
0	1	0	0	0	1	6	ODD	1
0	1	0	0	1	1	6	ODD	2
0	1	0	1	0	1	6	EVEN	1
0	1	0	1	1	1	6	EVEN	2
0	1	1	x	0	1	6	NONE	1
0	1	1	x	1	1	6	NONE	2
1	0	0	0	0	1	7	ODD	1
1	0	0	0	1	1	7	ODD	2
1	0	0	1	0	1	7	EVEN	1
1	0	0	1	1	1	7	EVEN	2
1	0	1	x	0	1	7	NONE	1
1	0	1	x	1	1	7	NONE	2
1	1	0	0	0	1	8	ODD	1
1	1	0	0	1	1	8	ODD	2
1	1	0	1	0	1	8	EVEN	1
1	1	0	1	1	1	8	EVEN	2
1	1	1	x	0	1	8	NONE	1
1	1	1	x	1	1	8	NONE	2

The UART Transmitter output will have less than 1% Distortion at baud rates of up to 218K baud (assuming the Transmitter Register Clock is perfect) and is, therefore, compatible with virtually all other asynchronous receivers.

Figure 8 is a block diagram of the Receiver portion of the UART. Serial asynchronous data is provided to the Receiver Input (RI). A start bit detect circuit continually searches for a logic one to logic zero transition while in the idle (logic one) state. When this transition is located, a counter is reset and allowed to count until the center of the start bit is located. If the input is still a logic zero at the center, the signal is assumed to be a valid start bit and the counter continues to count to find the center of all subsequent data and stop bits. (Verification of the start bit prevents the receiver from assembling an erroneous data character when a logic zero noise

Figure 6. TRANSMITTER BLOCK DIAGRAM

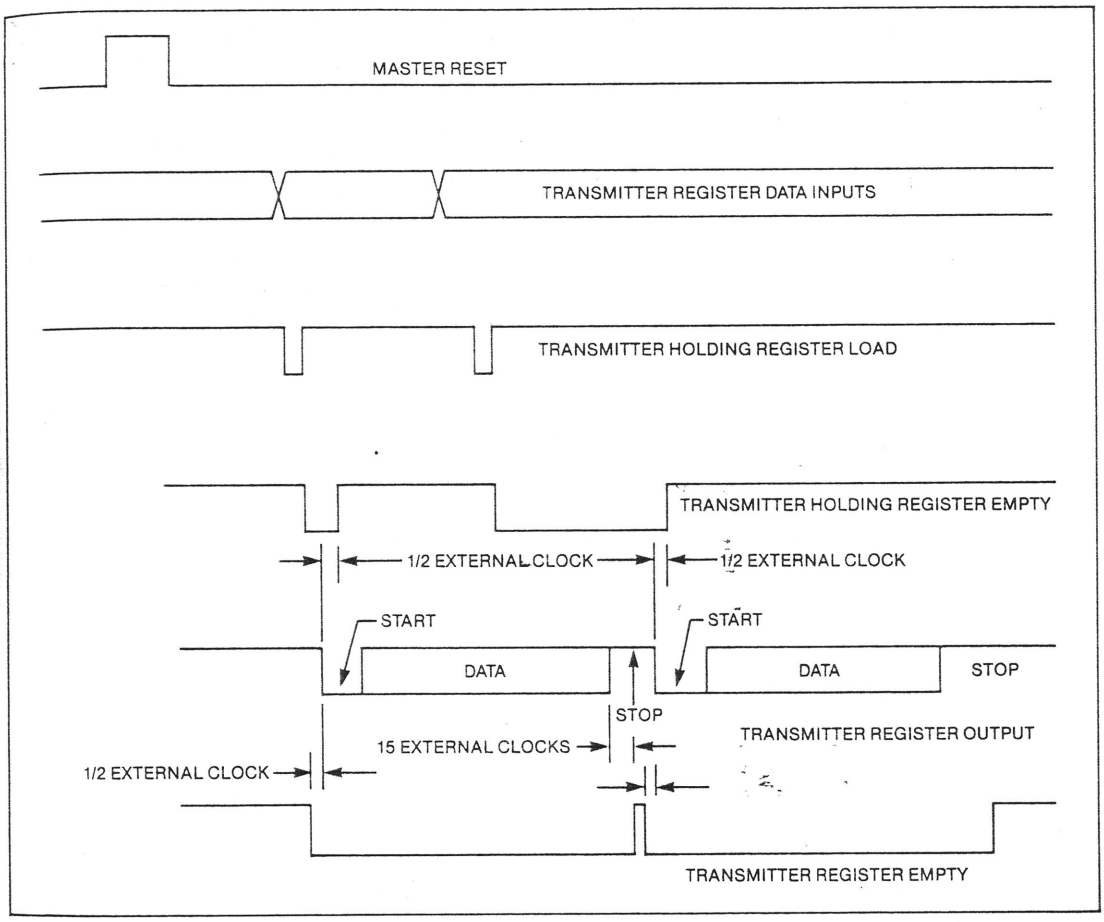

Figure 7. TRANSMITTER TIMING DIAGRAM

spike is presented to the Receiver Input). The Receiver is under control of the Control Register described in the previous paragraph. This register controls the number of data bits, number of stop bits, and the type of parity as described in Table 2. The word length gating circuit adjusts the length of the Receiver Register to match the length of the data characters. A parity check circuit checks for even or odd parity if parity was programmed. If parity does not check a Parity Error signal will be set to a logic one and this signal will be held until the next character is transferred to the Holding Register. A circuit is also provided that checks the first stop bit of each character. If the stop bit is not a logic one, the Framing Error line will be set to a logic one and held until the next character is transferred to the Holding Register. This feature permits easy detection of a break character (null character with no stop element). As each received character is transferred to the Holding Register, the Data Received (DR) line is set to a logic one indicating that the external device may sample the data output. When the external device samples the output, it should strobe the Data Received Reset (DRR) line to a logic zero to reset the DR line. If the DR line is not reset before a new character is transferred to the Holding Register (i.e., a character is lost) the Overrun Error line will be set to a logic one and held until the next character is loaded into the Holding Register. The timing for all of the Receiver functions is obtained from the external Receiver Register Clock which should be set at 16 times the baud rate of the transmitter.

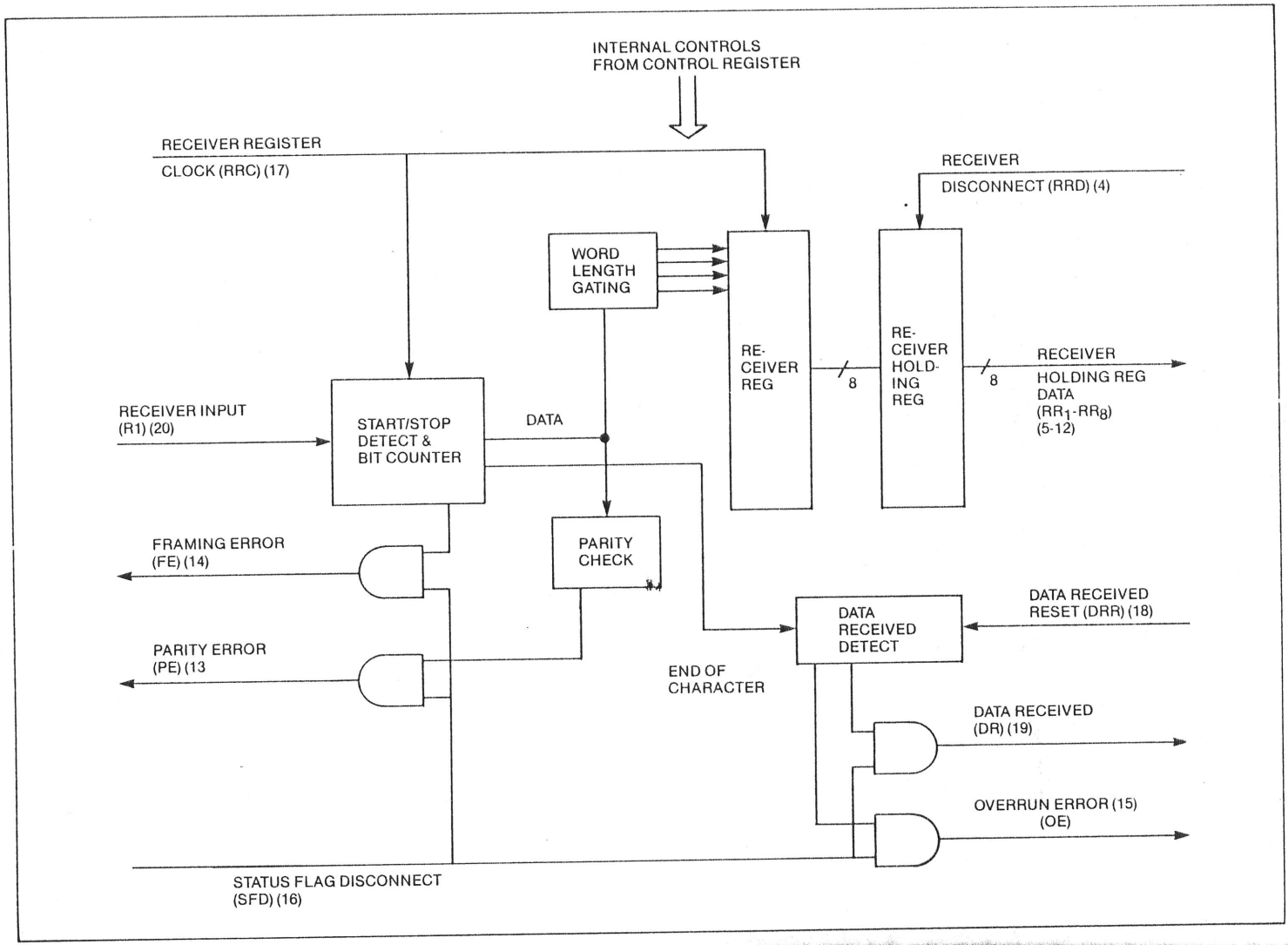

Figure 8. RECEIVER BLOCK DIAGRAM

Figure 9 illustrates the relative timing of the Receiver signals. A Master Reset strobe places the unit in the idle mode and the Receiver then begins searching for the first start bit. After a complete character has been decoded, the data output and error flags are set to the proper level and the Data Received (DR) line is set to a logic one. Although it is not apparent in Figure 9, the data outputs are set to the proper level one half clock period before the DR and error flags, which are set in the center of the first stop bit. The Data Received Reset pulse resets the DR line to a logic zero. Data can be strobed out at any time before the next character has been disassembled.

The UART Receiver uses a 16X clock for timing purposes. Furthermore, the center of the start bit is defined as clock count 7-1/2. Therefore, if the receiver clock is a symmetrical square wave as shown in Figure 10, the center of the bits will always be located within ±3.125% (assuming a perfect input clock) thus giving a receiver margin of 46.875%.

In Figure 10, the start bit could have started as much as one complete clock period before it was detected, as indicated by the shaded area of the negative going transition. Therefore, the exact center is also unknown by the shaded area around the sample point. This turns out to be ± 1/32 = ±3.125%.

If the receiver clock is not perfect, then the receiver distortion margin must be further reduced. For example, if the clock had 1.0% jitter, 0.1% offset and the positive clock pulse was only 40% of the clock cycle; then, for a 10 element character, the clock would add:

1.0% (Jitter)	+ (0.1% x 10) (Offset)	+ 0.1 (1/16) (Non-symmetrical Clock)	= 2.3% Distortion

(The frequency offset was multiplied by the number of elements per character since the offset is cumulative on each element.

Since a clock with these characteristics is very easy to obtain, it is apparent that a receiver operating margin of slightly over 45% is very easy to achieve when using the UART. Furthermore, this margin is sufficient for virtually all existing transmitters and modems presently in use.

The UART also begins searching for the next start bit exactly in the center of the first stop bit so that minimum character distortions of up to 50% can be accepted.

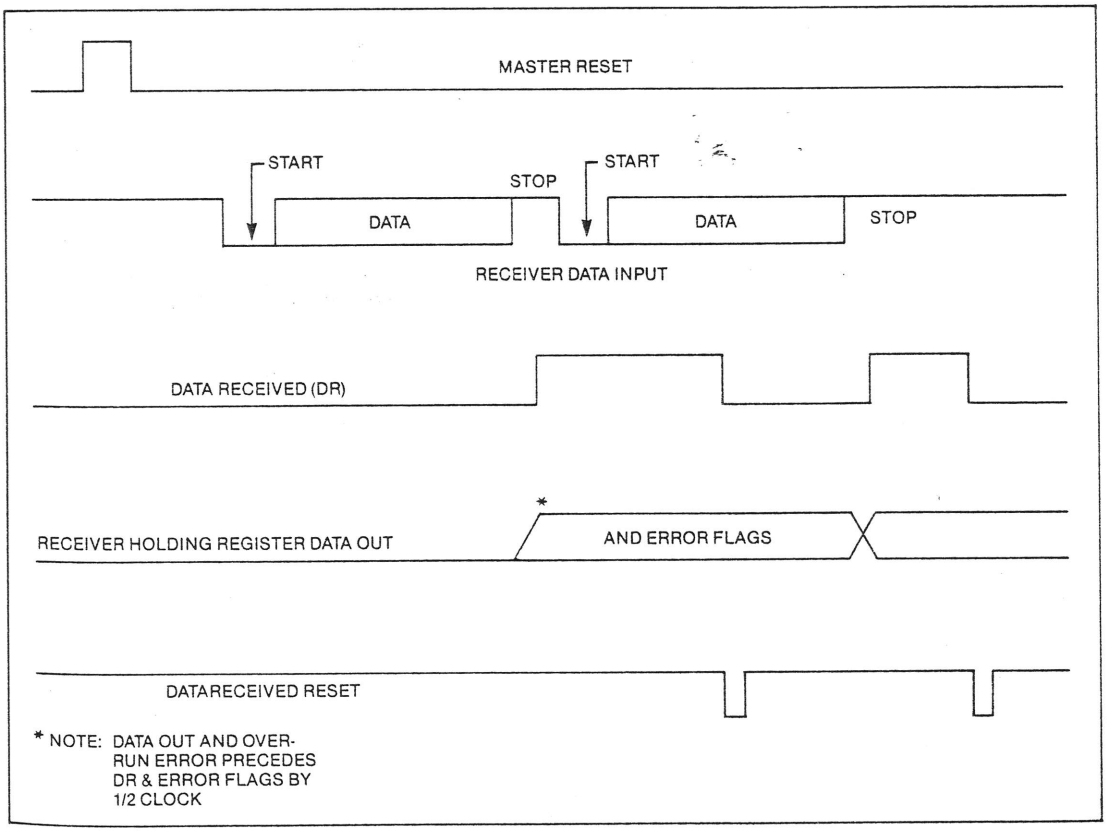

Figure 9. RECEIVER TIMING DIAGRAM

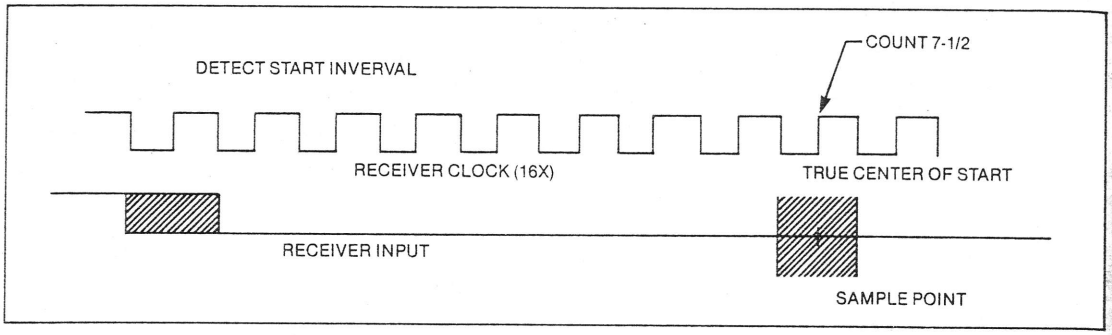

Figure 10.

A break character (null character without a stop bit) will lock the receiver up since it will not begin looking for the next start bit until a stop bit has been received.

TYPICAL UART APPLICATION

The UART is ideally suited for use in distributed computer networks such as is illustrated in Figure 11. One of the primary purposes of the communications controller is to assemble and disassemble the asynchronous characters (required for communication with the data terminals) to/from the parallel data format required by the host computer. Often the communications controller is a micro-computer and character assembly/disassembly is performed by the software. When this is the case, the micro-computer must be interrupted at a rate equal to 8 to 16 times the baud rate of all terminals being handled by the controller. (The actual interrupt rate depends on the amount of distortion that can be experienced on the received characters). When the number of terminals exceeds 8 to 16, even the most powerful micro-computers become overloaded due to the high interrupt rate and the complex algorithms required by the software.

The UART greatly reduces this problem by performing the character assembly/disassembly functions in external hardware as shown in a typical configuration in Figure 12. This solution not only reduces the interrupt rate by a factor of up to 176, but it also greatly reduces the micro-computer load, thus freeing it for other functions.

Since the UART inputs and outputs are TTL compatible, it interfaces directly with virtually all micro-computer I/O busses. In Figure 12, the micro-computer Data Output Bus is connected to the Transmitter Register (TR) inputs and the Control Register inputs. When the micro-computer has a character to transmit, the character is placed on the Data Output bus and the address of the appropriate UART is placed on the Device Address Bus. The Address Decode circuit will output a THRL load pulse under control of the Data Out Strobe from the micro-computer. When the control register should be changed, a new 5 bit control word is placed on the Data Output Bus and along with an appropriate device address which is converted to a CRL load pulse in the Address Decode circuits, again under control of the Data Out Strobe. A THRE Pulse to the Interrupt Request circuit will notify the micro-computer when a new character may be provided to the UART for transmission.

When a character has been received, a DR signal to the Interrupt Request circuit will request an interrupt from the micro-computer. The micro-computer will respond by setting the proper device address and provide a Data in Strobe pulse. The Address-Decode circuit then sets the RRD line and SFD line to the appropriate receiver to enable the Data Outputs onto the mini Data Input Bus. The Data in Strobe from the micro-computer then resets the DR signal with a DRR pulse from the Address Decode circuit.

The UART Transmitter Output (TRO) and Receiver Input (RI) must generally be converted to RS232 levels if they interface with a modem as shown in Figure 12. RS232 is a standard that has been established by the Electronic Industries Association for the interface between data terminals and data communications equipment. RS232-C defines a space as greater than 3 volts and a mark as less than negative 3 volts at the Receiver input. A transmitter output of between 5 and 15 volts is a space while a level between -5 and -15 is a mark. The input/output impedances and signal rise and fall times are also specified by RS232. Fairly simple discrete level translators can be used to convert from the TTL levels to the RS232 levels, or monolithic IC's are also available.

It should be noted that the typical application illustrated in Figure 12 is only one of many and it does not take advantage of many of the UART features. For example, the Status Flags could be tied to a separate interrupt request bus or the TRE output could be used to implement half-duplex operation.

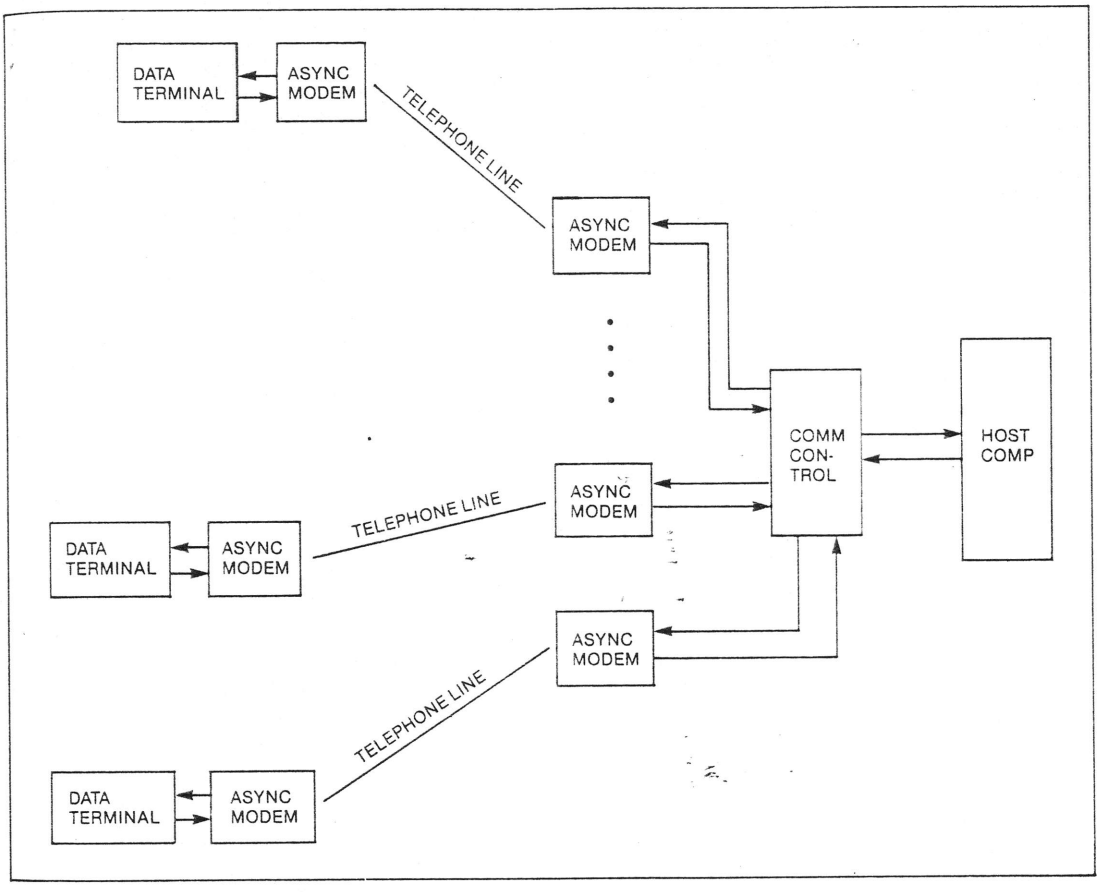
DATA TERMINAL
ASYNC MODEM
TELEPHONE LINE
ASYNC MODEM
COMM CON-TROL
HOST COMP
DATA TERMINAL
ASYNC MODEM
TELEPHONE LINE
ASYNC MODEM
DATA TERMINAL
ASYNC MODEM
TELEPHONE LINE
ASYNC MODEM

Figure 11.

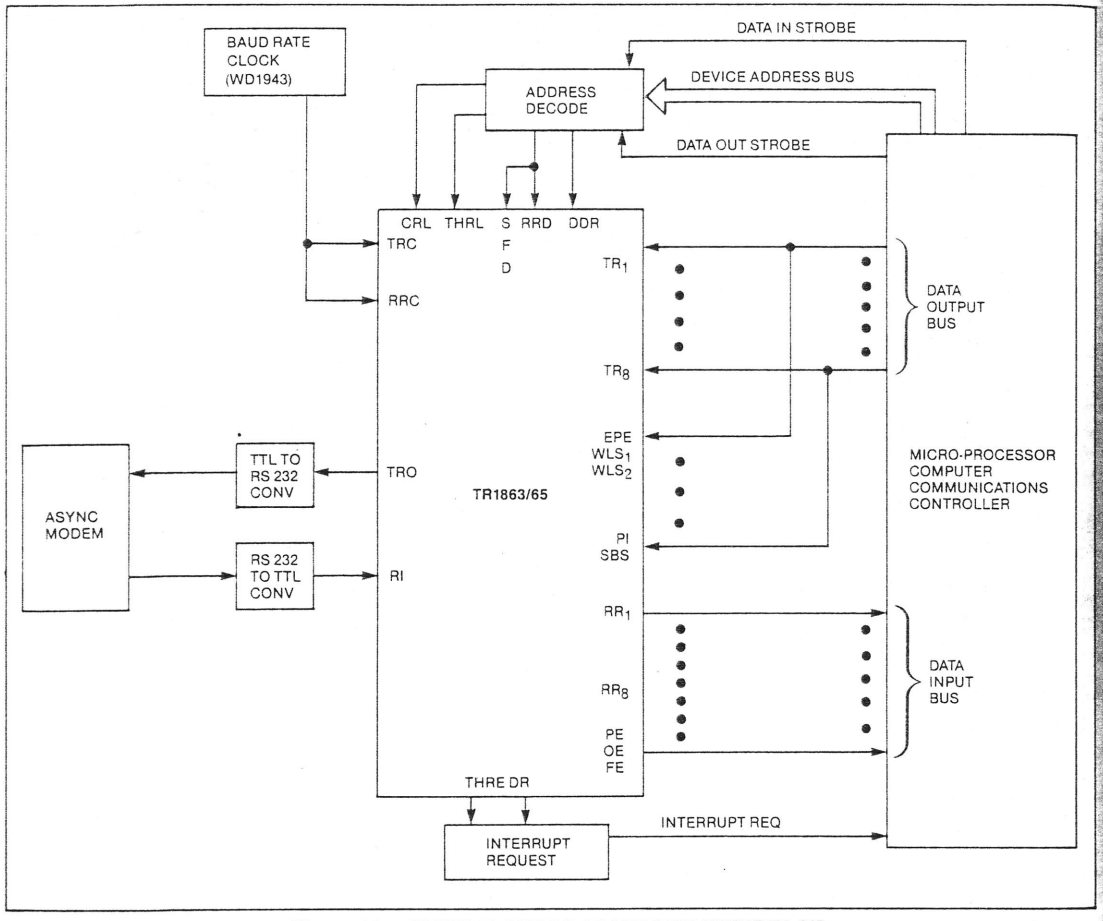

Figure 12. TYPICAL MICROCOMPUTER INTERFACE

Printed in U.S.A.