1-1-1978

A cassette recorder interface design with high immunity to tape speed variations.

Richard G. McMahon

A CASSETTE RECORDER INTERFACE DESIGN WITH HIGH IMMUNITY TO TAPE SPEED VARIATIONS

by

Richard G. McMahon

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Electrical Engineering

Lehigh University

1978

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering.

August 14, 1978
(date)

Professor in Charge

Chairman of Department

CONTENTS

ACKNOWLEDGEMENT

I would like to thank the following persons for their contributions to the successful completion of the thesis:

- My wife, Linda, for her patience during my many hours of absence while I was building and debugging the circuit. In addition, for typing the manuscript and drawing the illustrations

- Thomas Carpenter, for his advice during the construction of the circuit board and for the use of his computer to run the experiments

- Arthur Larky, my thesis advisor, for his definition of the circuit concept and his guidance throughout the project

ILLUSTRATIONS

ABSTRACT

A low cost interface between a cassette tape recorded and a S-100 bus which is used in several microcomputer systems has been designed and evaluated. The system uses an Intel 8251 USART (Universal Syncronous/Asyncronous Receiver/ Transmitter) to manage the data transmission, insert sync characters, generate and check parity, as well as corresponding with the CPU to accept commands and convey the current status of the system.

The coding format for the system is pulse width modulation which uses an asymetrical waveform to distinguish between logical "1" and a logical "0". Logical "1" is present when the signal is high immediately following the positive going edge of the clock pulse and for a period of approximately two-thirds of the time between positive going clock pulses. Conversely a logical "0" is present when the signal remains high for one-third clock period immediately following the positive going clock edge.

A Fourier analysis was done to determine what bandwidth the recorder would have to have to transmit data to any particular rate. The analysis indicated that a recorder and interface circuitry with a bandwidth of approx-

imately eight times the clock rate of the data transmission might be acceptable.

The data is decoded by counting down in an up/down counter while the data signal is high and counting back up while the data signal is low; therefore, a carry out indicates that the time during which the signal remained high was less than one-half of the data interval which indicates that a logical "0" is present. The absense of a carry out indicates that a logical "1" is present.

Because the data is extracted from the ratio of the high and low level duration during the data interval short term variations (within the data interval) can be fifty percent of the shorter of the two durations. In addition, because the clock signal is integrated into the data signal, the only restrictions on the long term variations of the recorder speed are the modulus of the up/down counter and the bandwidth of the system.

Chapter 1

INTRODUCTION

Sales of small microprocessor-based computers are experiencing rapid growth to the computer hobbyist for home use. One of the primary problems is to have a low cost, reliable method for the storage of data and programs.

The most popular method for off-line storage is to use audio cassette recorders. There are many formats that can be used to store and retrieve data from the tape, but by far the ones most commonly used in low cost microcomputer systems are the frequency shifting techinques because they can be implemented with a minimal amount of hardware and decoding of the retrieved data is simply a matter of filtering to determine which frequency is present. The systems are usually implemented with the aid of a UART (Universal Asyncronous Receiver/Transmitter). These circuits use some rather complex analog circuitry to recover the receiver clocks and to generate the 16X clocks required for the USART and, therefore, maintaining correlation between the received data and the recovered clocks has been a problem in many designs. Typically, a manufacturers specification will require that the speed variation of the tape drive be less than plus or minus five percent. This specification means that, in general, inexpensive recorders cannot be used;

instead, of one of the higher quality medium prices recorders is required.

The design proposed in this paper employs a more powerful controller (USART instead of UART) and uses digital techniques to decode the receiver data and recover the receiver clock. The long term speed variations of the tape drive should be limited only by the modulous of the up/down counter used in the receiver and the bandwidth of the tape recorder. The short term speed variation (jitter) allowed should be about one-third of a cycle.

Chapter 2

DESIGN CONSIDERATIONS

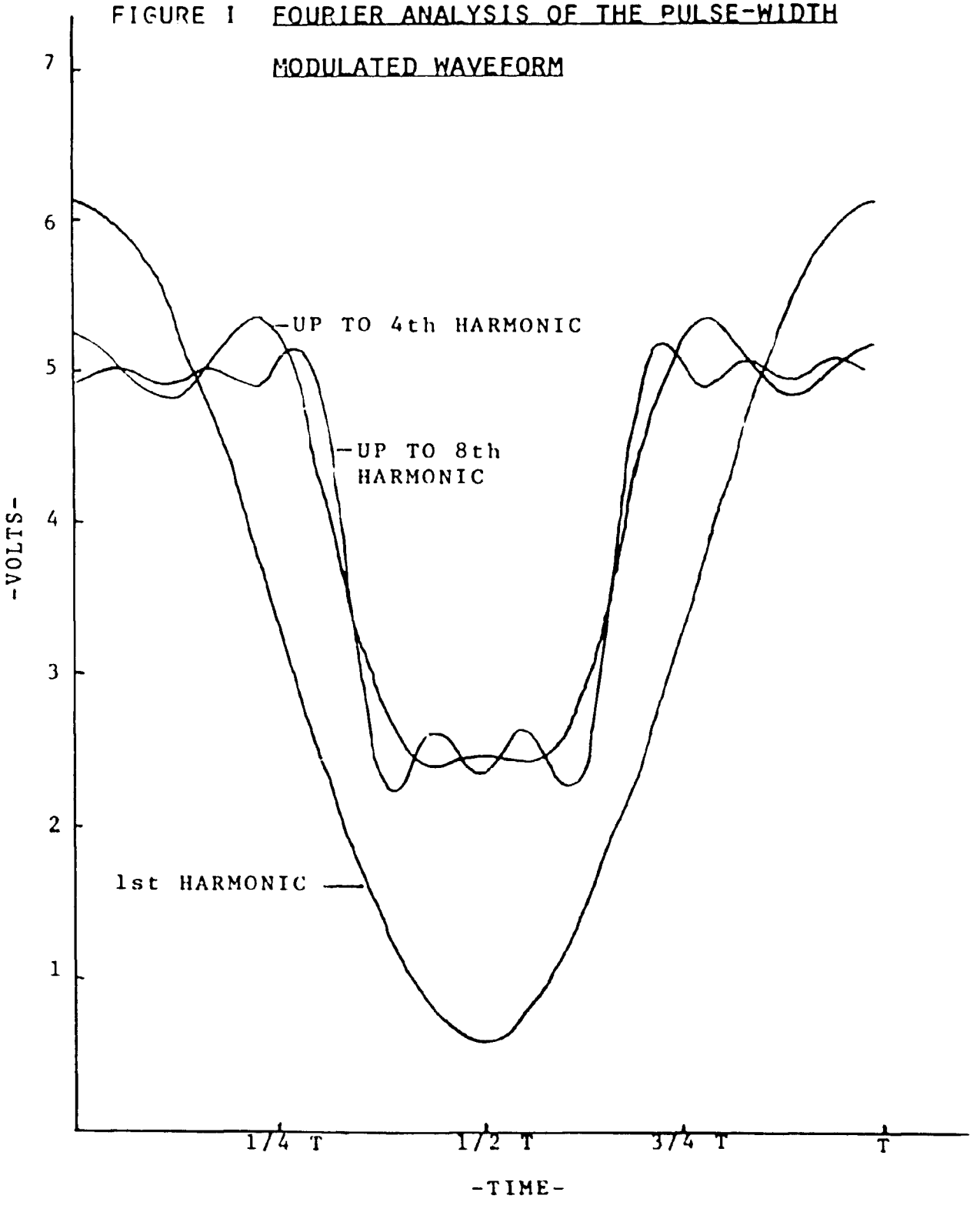
2.1 Bandwidth Requirements-

The limiting component with respect to bandwidth is the cassette recorder. A Fourier analysis of the bandwidth required to record and retrieve the pulse-width modulated signal has been done (Appendix A). The shape of the signal including all harmonics up to the first, fourth and eighth harmonic are shown in Figure 1. The analysis indicates that a bandwidth that is eight times the data rate should give a relatively good reproduction of the recorded data. Therefore, a baud rate of 400 bps should be feasible with a recorder with a bandwidth of approximately 3200 Hz.

2.2 Description and Operation of the USART-

The USART (Iniversal Syncronous/Asyncronous Receiver/ Transmitter) is capable of operating with a wide variety of communications formats. The USART can be used in either syncronous or asyncronous modes. It has a variable character length of five to eight bits. It can generate and receive odd parity, even parity, or no parity. In the asyncronous mode, the USART generates a start bit and

FIGURE 1 FOURIER ANALYSIS OF THE PULSE-WIDTH MODULATED WAVEFORM



one, one and one-half, or two stop bits for each character. In the syncronous mode, the USART can be syncronized either internally or externally. If the internal sync mode is selected, the USART will generate and transmit one or two sync characters at the beginning of each block of data and will hunt for a sync character(s) on the receiver line to initiate reception of data. All of these options are completely software programable.

After the mode has been selected, the CPU must issue a command instruction ($C/\overline{D}=1$, $\overline{CS}=0$, $\overline{WR}=0$) which tells the USART what operation to perform next. The command instruction is an eight bit character as defined in Figure 2. The USART also maintains a status register that can be polled by the CPU to determine the current status of the USART. The status register has the format shown in Figure 3 and can be read using a status read command ($C/\overline{D}=1$, $\overline{CS}=0$, $\overline{RD}=0$). The generalized operational flowchart for the USART is shown in Figure 4. The functional description of the Input/Output pins is in Appendix C. The design that is described in this paper uses the internal syncronous mode with two sync characters and one parity bit at the end of each character, but these choices are completely optional and the system will support other choices.

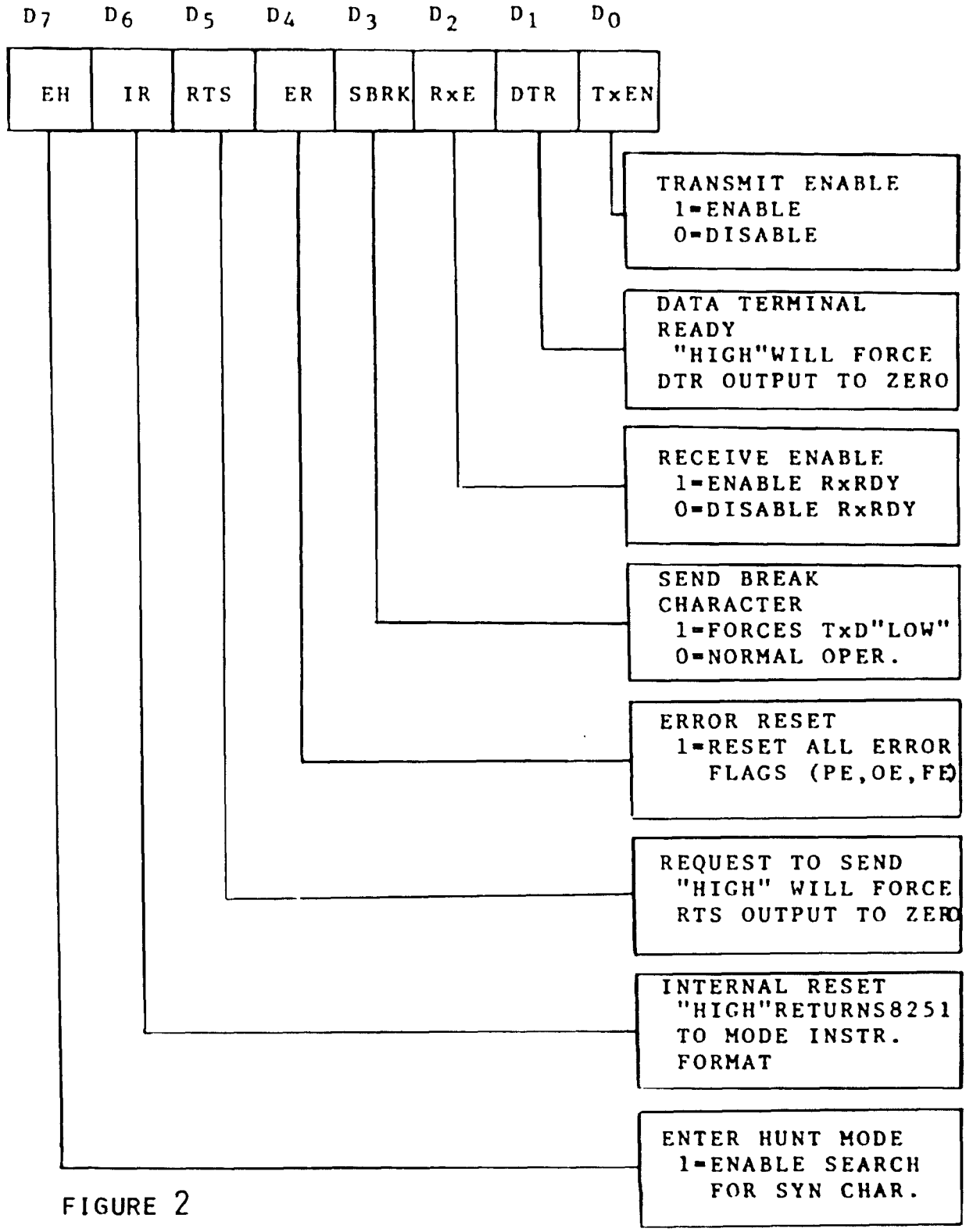


FIGURE 2

COMMAND INSTRUCTION FORMAT

source: ref. 7

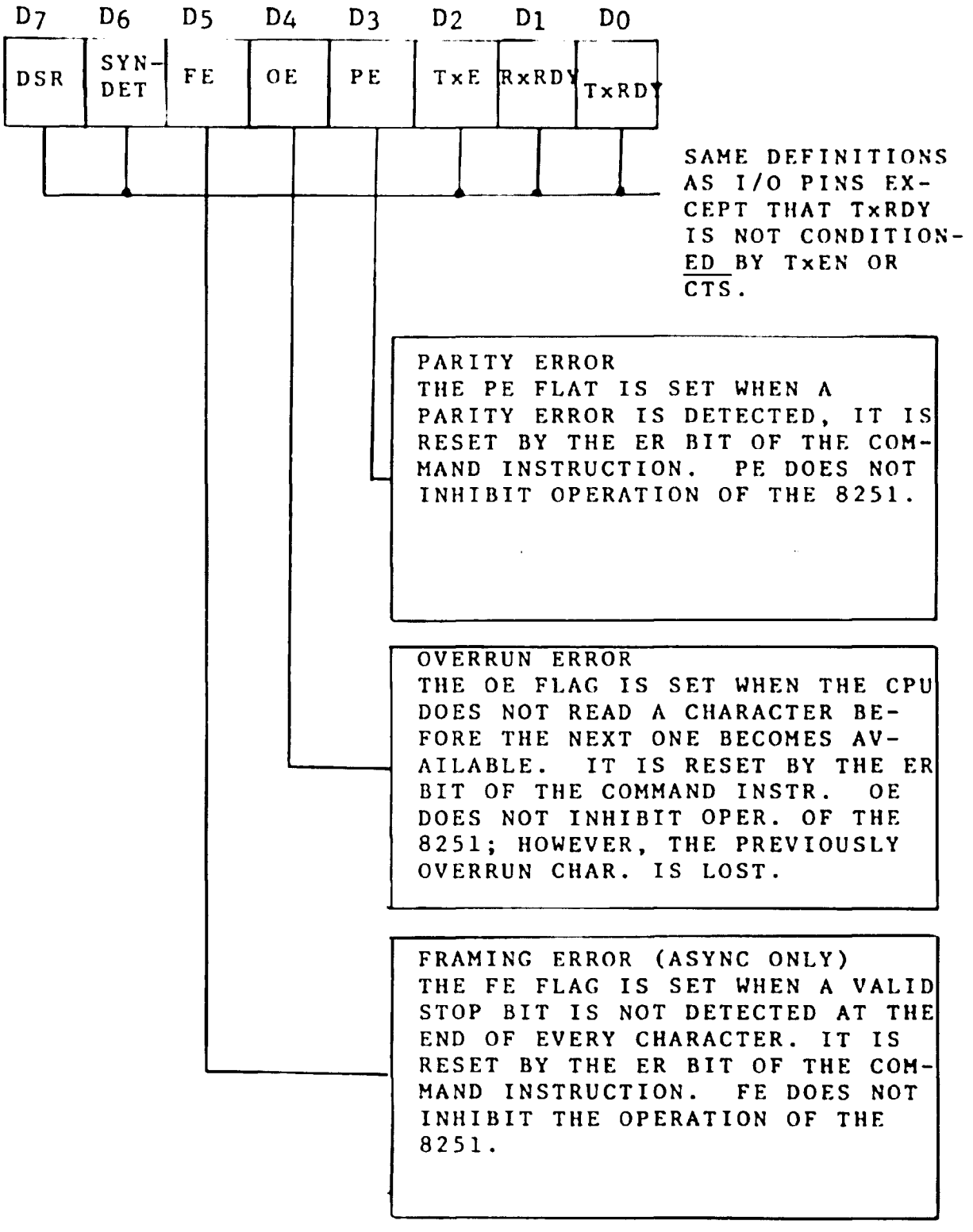


FIGURE 3 STATUS REGISTER FORMAT

source: ref. 7

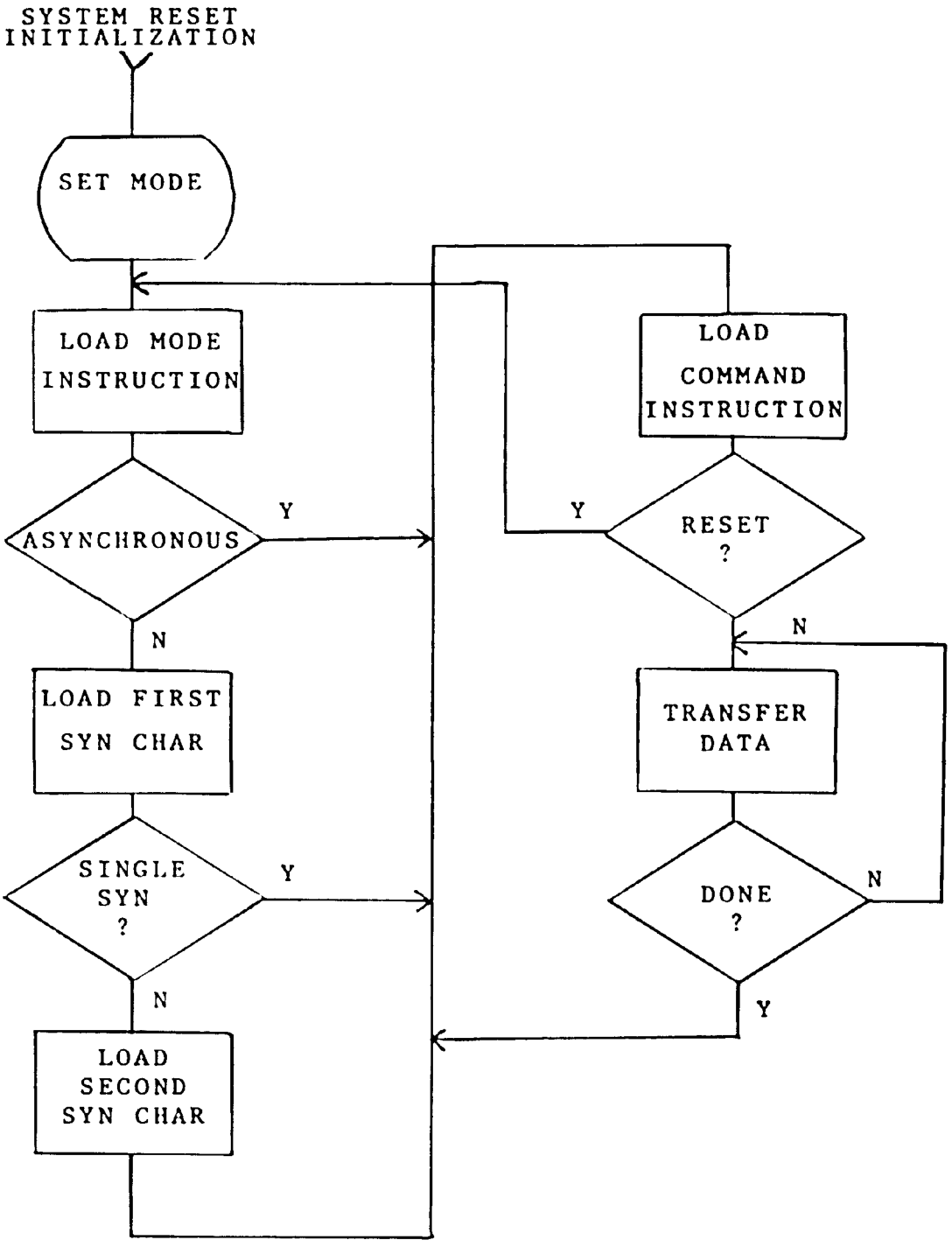


FIGURE 4 INITIALIZATION FLOWCHART

source: ref. 7

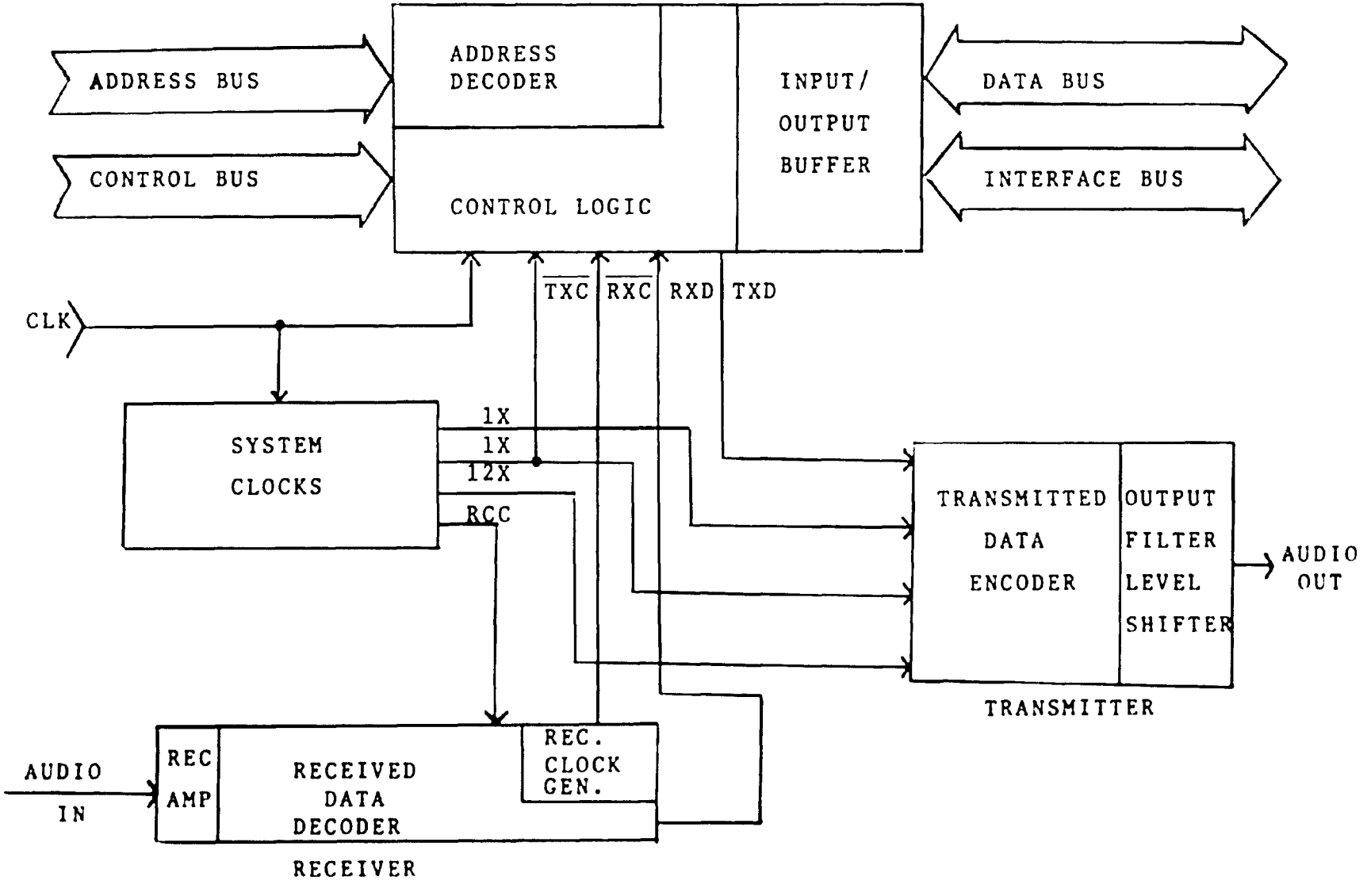
Chapter 3

THEORY OF OPERATION

3.1 Functional Description-

The system can be segmented into four functional blocks. These are: Clock Generator, Interface and Control Logic, Transmitter and Receiver which are interconnected as shown in Figure 5. The clock generator provides the transmitter and USART with the 1X clock that determines the data rate for the transmitter. It also supplies a 12X clock that is used by the transmitter to generate the PWM (pulse-width modulated) data. A third clock is generated to be used by the receiver to count the relative length of the data pulse. This third clock is selected from the available signals in the clock generator circuit to be as close to 100X as possible without exceeding 100X. This will allow long term speed variations of at least one-hundred and fifty percent without exceeding the capabilities of the Up/Down counters in the receiver circuitry. The interface and control circuitry is built around the Intel 8251 USART. The function of this block is to converse with the CPU and convert parallel data from the computer bus to serial data for the transmitter and accept serial data from the transmitter and convert

FIGURE 5 SYSTEM BLOCK DIAGRAM

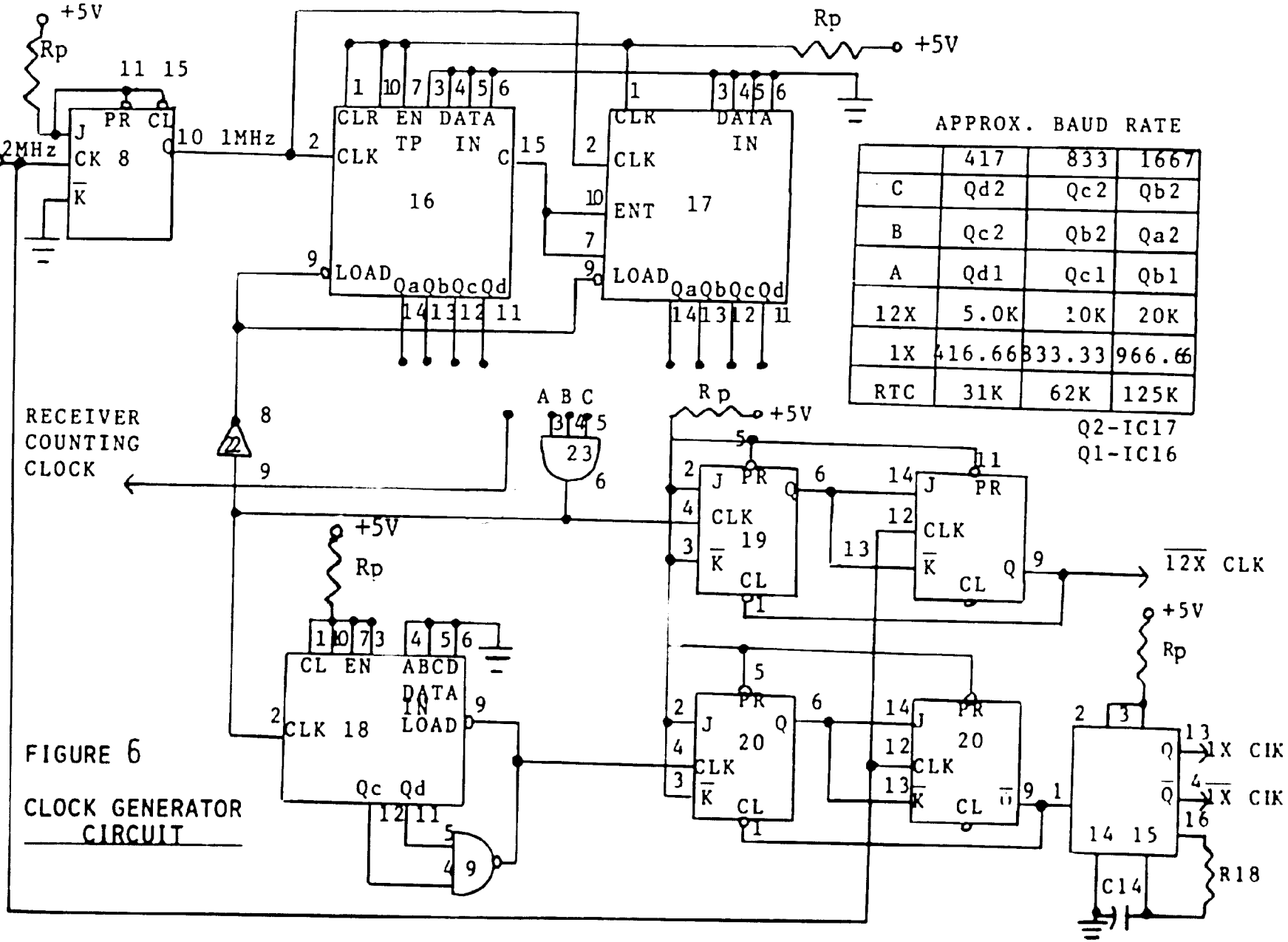


it to parallel data for the computer bus. The CPU programs the USART with regard to format of the data (start, stop, parity bits, etc.) and issues command on the type of operation (clear, read or write) to be executed. The CPU then polls the USART by using the status read operation to determine when to receive or transmit data. The transmitter logic takes the serial data outputted from the USART and converts it into the pulse-width modulated data to be stored on the tape. The receiver block takes the audio data in and reshapes and converts it to TTL level and then decodes the PWM data into a serial data stream. The receiver also recovers the clock signal by sampling the rising edge of the PWM data.

3.2 Circuit Design-

3.2.1 Clock Generator (Figure 6)-

The basic clock frequency is the phase two (Ø2) clock from the microcomputer bus. This signal is used to sync all the system clocks to the computer and as the base frequency for the system. The Ø2 clock is reduced to one MHz by the toggle flip-flop (IC-8); the resulting one MHz signal is used as the input to an eight-bit programmable counter (IC16 & 17). The 12X clock rate is selected by ANDing the appropriate counter outputs with IC23. The 12X clock signal is used to reset the eight bit counter with IC22 and is syncronized with the computer with IC19. The 12X



APPROX. BAUD RATE

	417	833	1667
C	Qd2	Qc2	Qb2
B	Qc2	Qb2	Qa2
A	Qd1	Qc1	Qb1
12X	5.0K	10K	20K
1X	416.66	833.33	966.66
RTC	31K	62K	125K

Q2-IC17
Q1-IC16

FIGURE 6

CLOCK GENERATOR CIRCUIT

clock signal is also used to generate the 1X clock by dividing by twelve with IC9. The 1X clock is then syncronized to the computer by IC20. The receiver counting clock is generated by selecting the highest frequency output from the eight bit counter that is not more than one-hundred times the baud rate.

3.2.2 Interface and Control Logic (Figure 7)-
The interface between the system and the S-100 bus is controlled by the USART described in 2.2. The system is addressed by using the seven inverters (IC1 & 2) that are connected from address lines A1 thru A7 to IC6 using jumpers J1 thru J7 and ANDed with either the SINP or SOUT signals from the CPU to generate the chip select ($\overline{CS}$) signal for the USART. Address line AØ is used to designate whether the information on the data bus is a data or command byte. The $\overline{WR}$ (write) signal for the USART is obtained directly from the $\overline{PWR}$ signal on the S-100 bus and the $\overline{RD}$ (read) signal is generated by inverting the SINP signal. The Ø2 clock is also obtained directly from the S-100 bus. The reset signal is generated by NORing the $\overline{PRESET}$ and $\overline{EXT}$ $\overline{CLR}$ signal available from the bus. The data bus (D7 thru DØ)is split to interface to the S-100 data input bus (DI7 thru DIØ) and the S-100 data output bus (DO7 thru DOØ) by tri-state buffers of IC6 and 7. The tri-state buffers activate the input data bus by ANDing SINP and $\overline{CS}$.

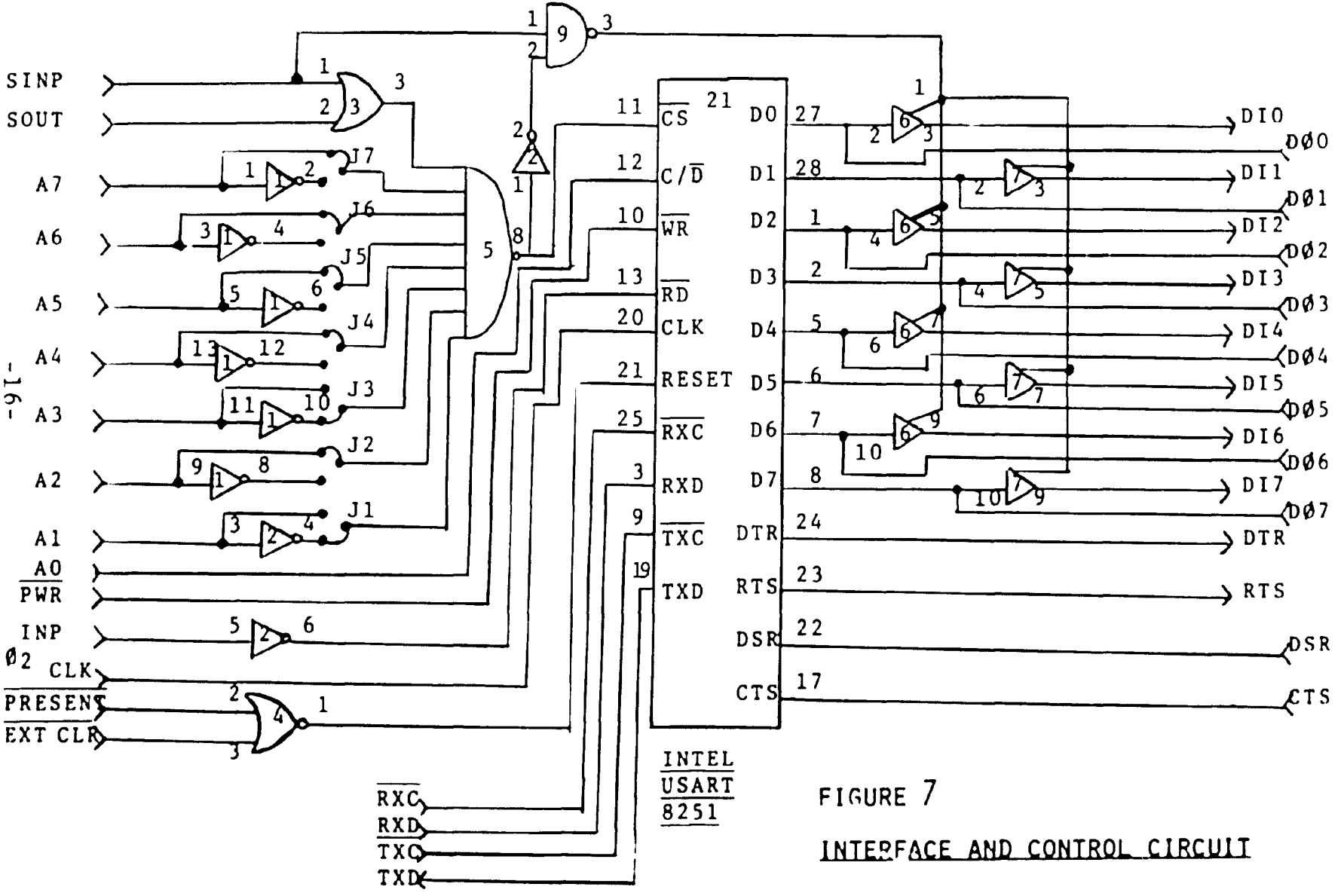


FIGURE 7

INTERFACE AND CONTROL CIRCUIT

The Interface bus contains the two output signals RTS and DTR and their input response signals CTS and DSR as defined in 2.2. $\overline{RXC}$, RXD, $\overline{TXC}$ and TXD are outputs of the other functional blocks.

3.2.3 Transmitter Logic (Figure 8)-

The transmitter logic generates the pulse-width modulated signal by ANDing the inverted serial data with the output of the C flip-flop of the twelve count counter (IC10) and ANDing the non-inverted serial data with the output of the D flip-flop of the twelve count counter. These ANDed signals are ORed and used to trigger the encoded data output of IC8. The resisters R11, R12 and R13 serve to reduce the output voltage level from TTL to 500mv for the auxiliary input to the recorder or 50mv for the MIC input to the recorder. The capacitors C_6 and C_{15} filter out the high and low frequency components respectively. The modulo twelve counter is implemented using a four bit binary counter and reseting by ANDing the $\overline{1X}$ clock and the inverted output signal. The modulo twelve counter is triggered using the $\overline{12X}$ clock. The transmitter timing diagram is shown in Figure 9.

3.2.4 Receiver Wave-Shaping and Logic (Figure 10)-

The first state of IC11 is a high gain amplifier (approx.

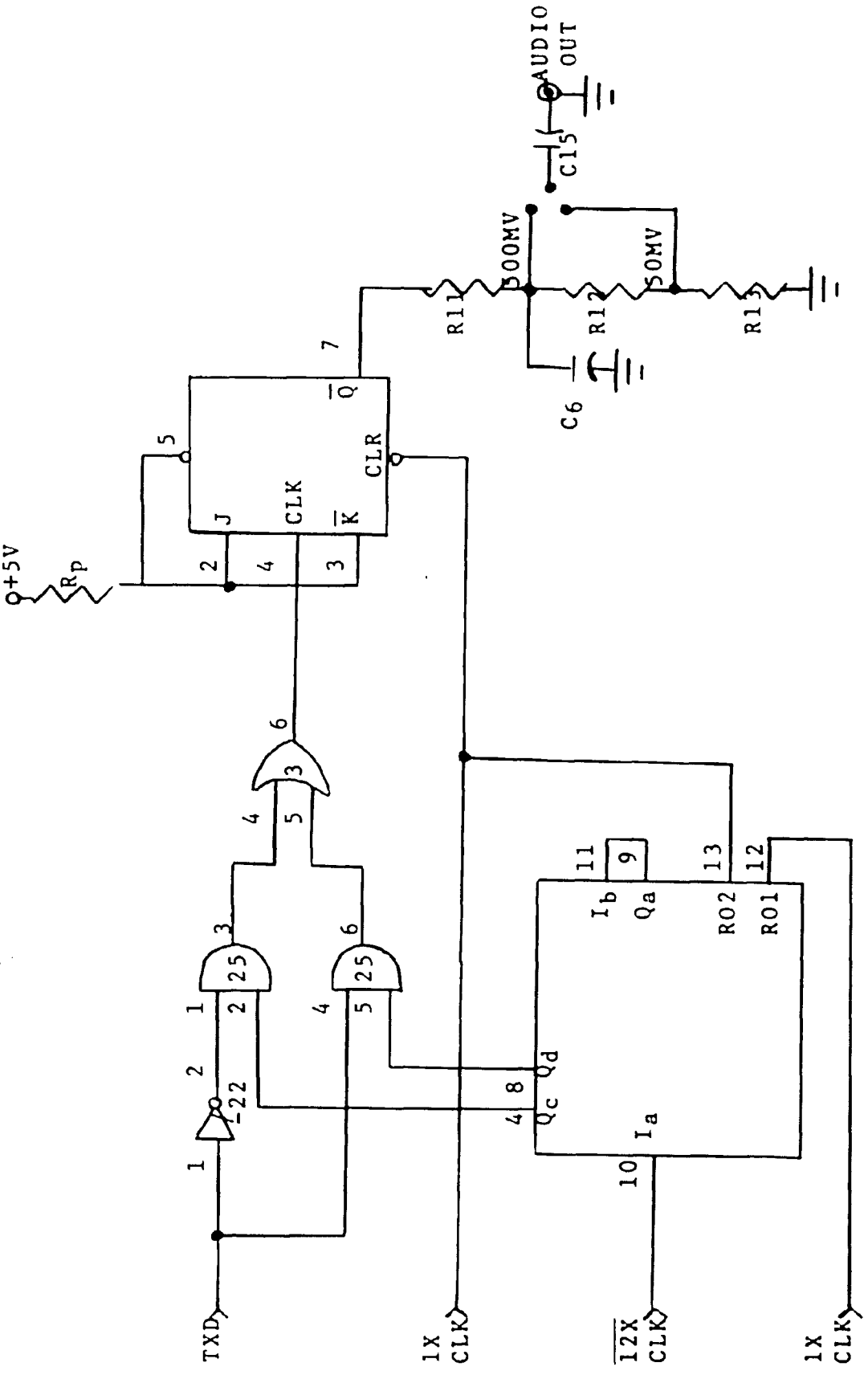


FIGURE 8 TRANSMITTER CIRCUIT

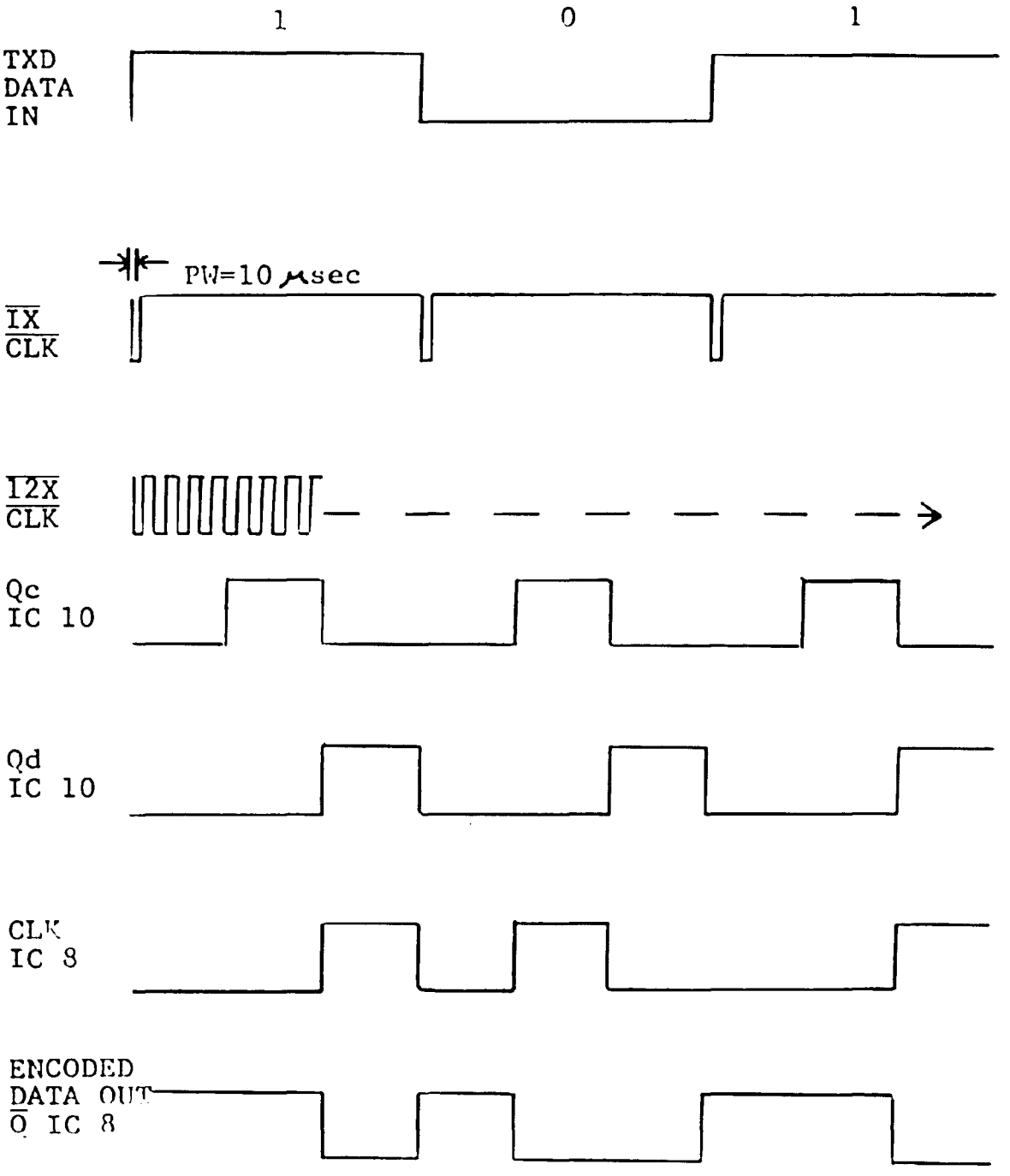


FIGURE 9 TRANSMITTER TIMING DIAGRAM

100). The current into the base of T3 causes C_2 to discharge (C_2 charges through R3) and the voltage changes on the gate of T1 make it act like a variable resistor which completes the negative feedback loop for the automatic gain control (AGC). The second state of IC11 is a comparator and performs the audio to digital conversion. The feedback resistor R9 in conjunction with R8 establishes the level on the positive input terminal. This level, whether positive or negative, is the threshold voltage which the negative input must exceed in order for the output to switch levels. The first inverter IC2) serves to buffer the amplifiers and the second inverter supplies the required negated signal for the down count on the first up/down counter (IC12). The two NAND gates (IC9) are used to gate the receiver counter clock with the high and low level of the encoded data. The up/down counters (IC12&13) are preset to binary sixteen and the high portion of the PWM data pulse enables the count down at the receiver counter clock rate (Approx. 60x baud rate). The counters start counting back up when the PWM signal goes low, therefore, if the PWM signal is low longer than it is high the counter will output a carry which signifies that a zero is present. The carry out signal is syncronized to the incoming data stream by IC14, which inverts the data and delays the data to the USART by one data clock width.

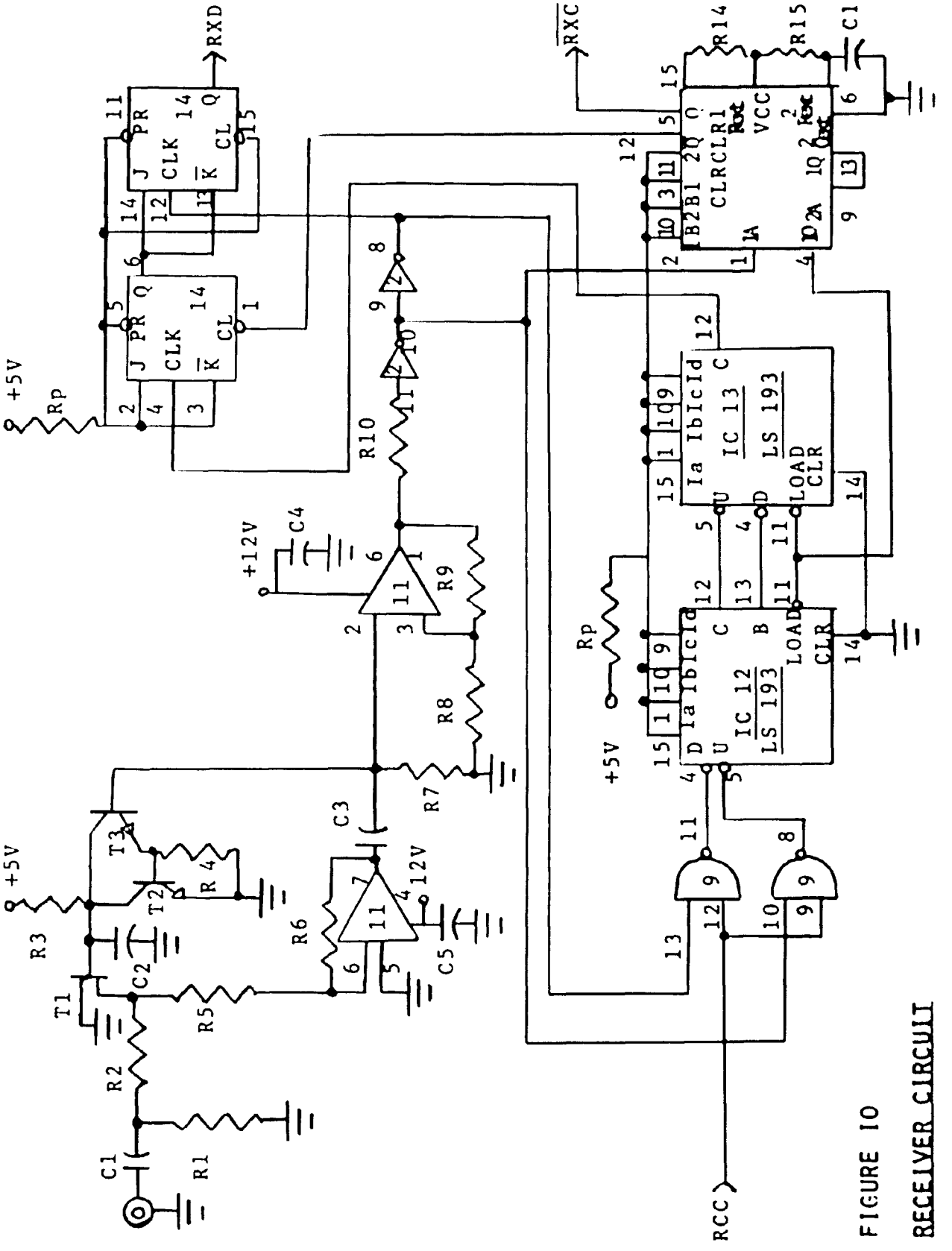


FIGURE 10

RECEIVER CIRCUIT

If the incoming data is a logical "1" the count down interval will be longer than the count up interval and carry out will not be present; which indicates that a logical "1" has been received. The receiver clock is generated by the two one-shots of IC15. The first one-shot introduces a 40 nsec delay to allow the second flip-flop in IC 14 to set-up. The second one-shot is configured to give a pulse-width of approximately 10 μsec because the USART requires that the receiver clock pulse width be at least twelve times the width of the phase two clock. The complete timing for the receiver circuit is shown in Figure 11. The system power distribution is shown in Figure 12. The complete parts list is shown in Table 1.

3.3 System Software-

The system software can be broken into three major subroutines, namely: USRUN- which is the controlling program, UREAD- which is the subroutine that reads data from tape and UWRITE- which is the subroutine that writes data onto the tape.

The flowchart for USRUN is illustrated in Figure 13. The program starts by clearing the accumulator and outputting "zeros" three times to the USART to be certain that the input buffer is cleared. After a reset operation, the MODE instruction is loaded into the USART. The mode instruction is the eight bit word 00111100 which indicates

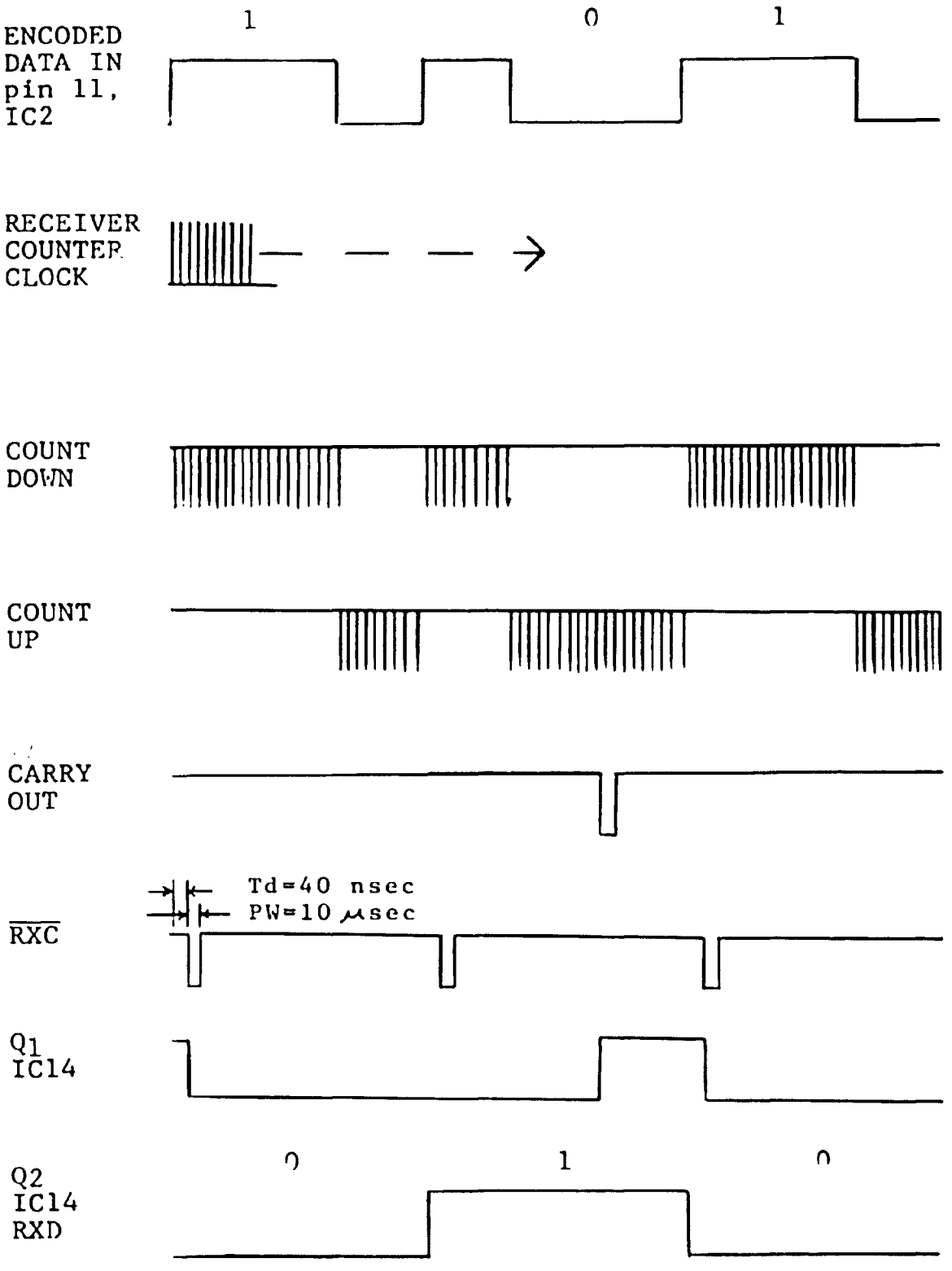


FIGURE II RECEIVER TIMING DIAGRAM

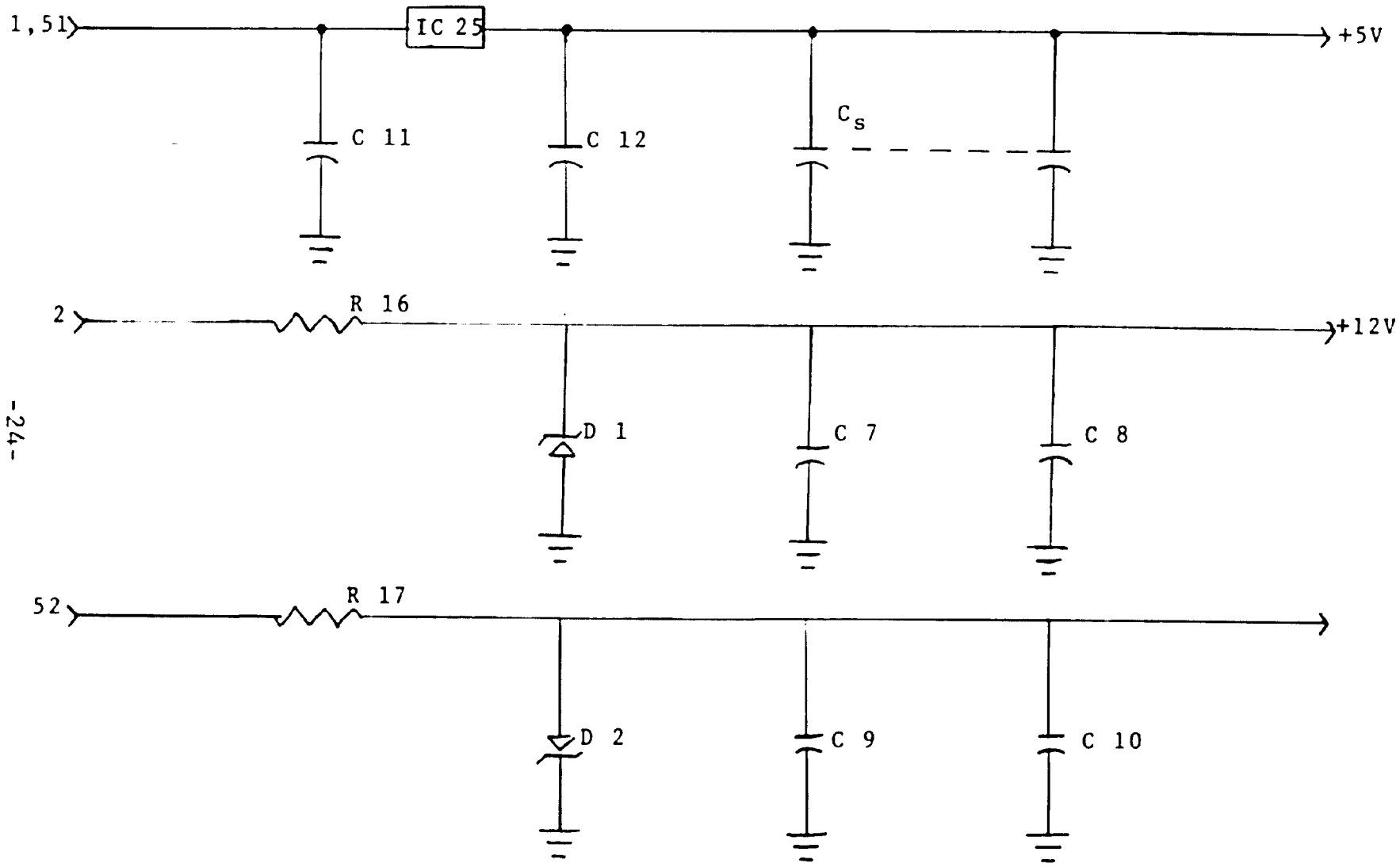


FIGURE 12 POWER DISTRIBUTION CIRCUIT

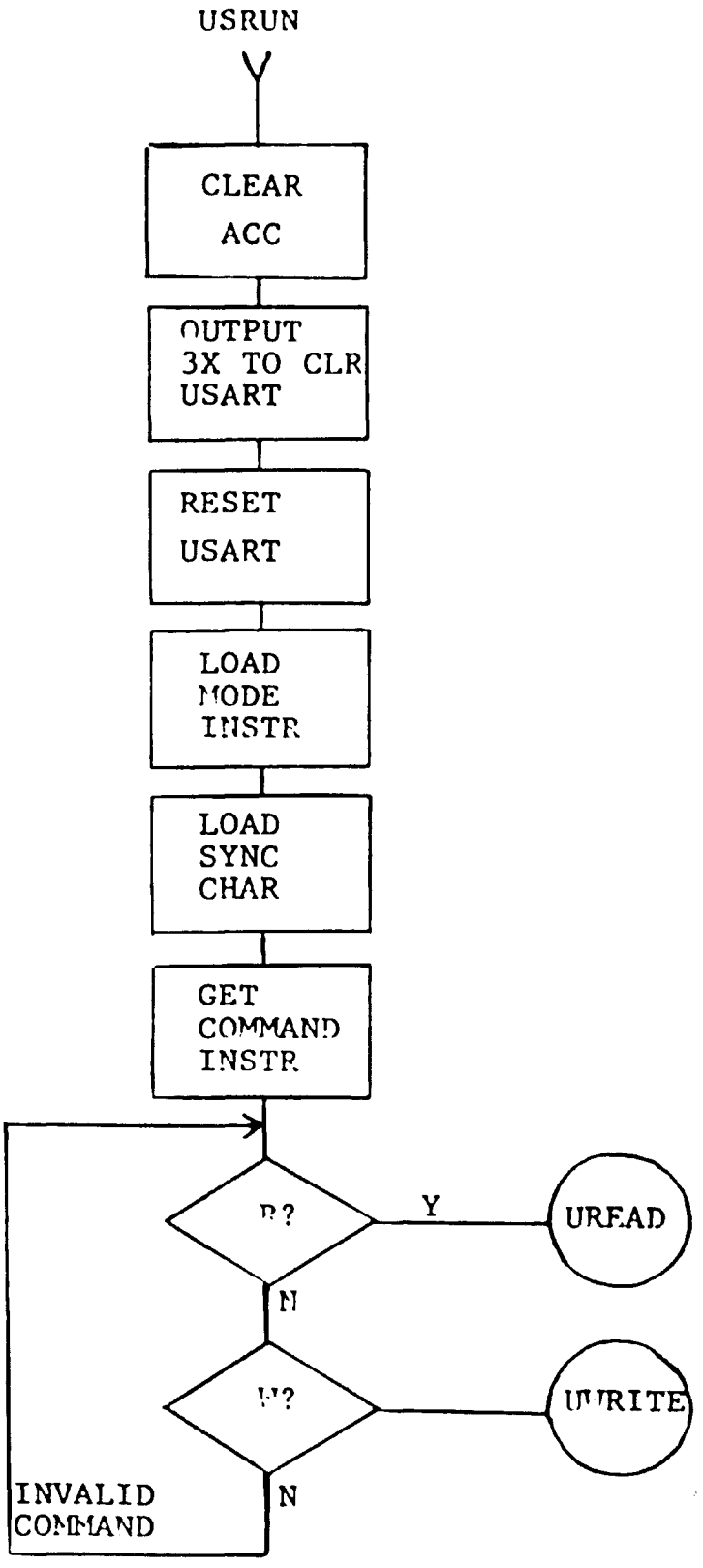


FIGURE 13 CONTROL PROGRAM (USRUN)

syncronous mode, 8-bit character length, even parity and internal sync. Next, the program loads the two sync characters into the USART and fetches the command instruction from a specified memory buffer. The command instruction is either an ASCII "R" or "W". The program checks for either "R" or "W" and, if either are present, transfers system control to UREAD or UWRITE respectively.

UREAD is the subroutine that reads data from the tape and transfers it to a preselected area in RAM. The flowchart for UREAD is shown in Figure 14. UREAD starts by clearing the "R" from the command instruction buffer and getting the starting and ending addresses from the receive address memory buffers. The program then loads the instruction command into the USART following the format of Figure 2. The instruction command for the read operation is the eight bit word 10010110 following the format of Figure 3. This instructs the USART to set the $\overline{DTR}$ output, enable the receiver, reset the error flags and enter the hunt mode. The program then waits until the receiver has received and decoded its sync characters at which time, by using the status read operation, the CPU will know that the receiver is receiving valid data and is syncronized. The program then polls the USART until the TXRDY flag is set. The CPU then reads data from the USART and increments the starting address register. This

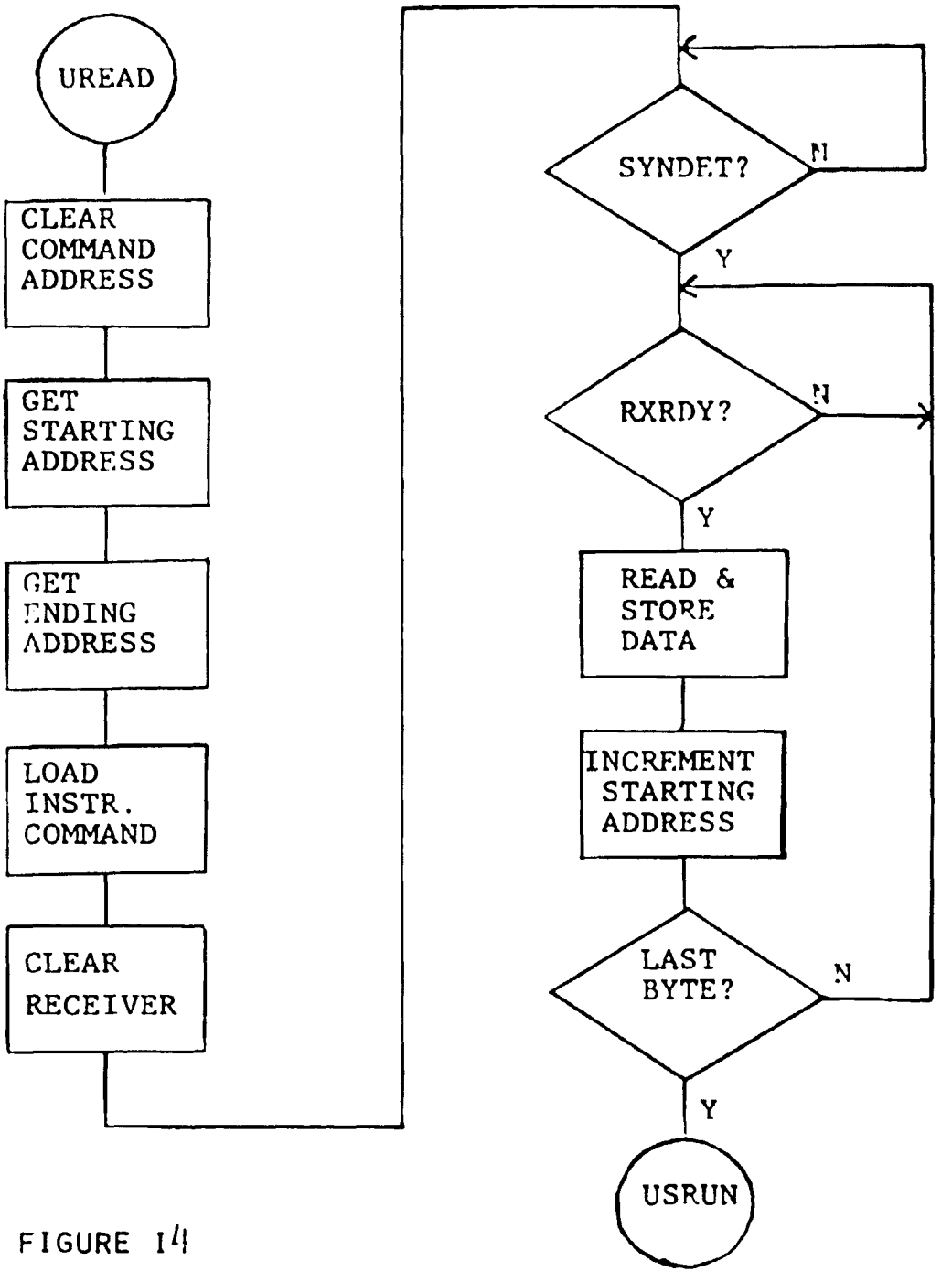


FIGURE 14

READ DATA PROGRAM (UREAD)

sequence of operations is repeated until the starting address is equal to the ending address which indicates that the last data byte has been transferred. The system control is then returned to USRUN.

UWRITE is the program that controls the system when parallel data from memory is being serially transmitted to the recorder. UWRITE starts by clearing the "W" from the command buffer and fetching the start and finish addresses from memory. The program then loads the instruction command which is the eight bit word 00110001. This command tells the USART to enable $\overline{\text{RTS}}$, reset error flags, and enable the transmitter. The program then gets the first data byte from memory and polls the USART until the TXRDY flag is set. UWRITE then outputs the data byte and increments the starting address. When the last byte has been transmitted, UWRITE returns control to USRUN.

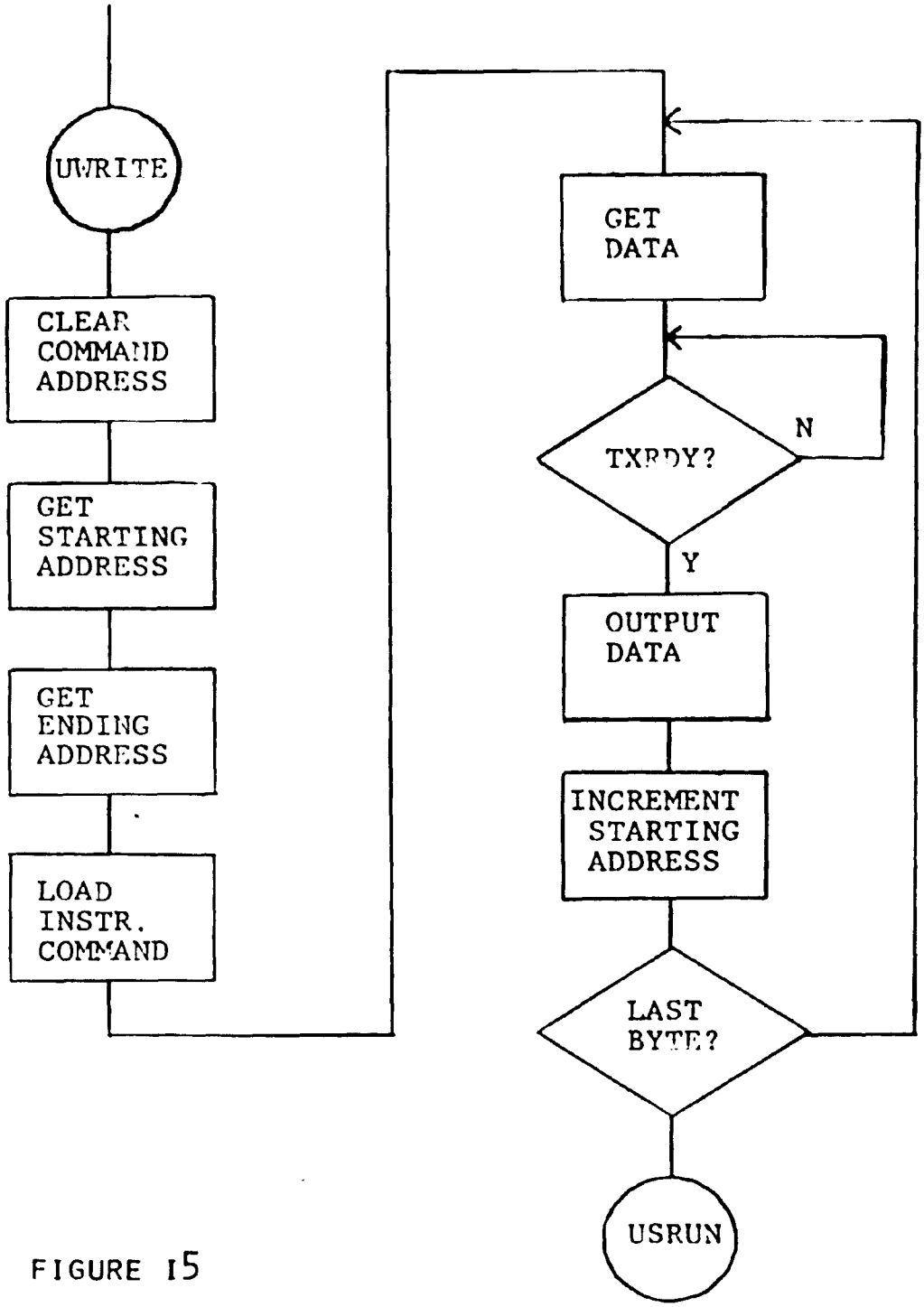


FIGURE 15

WRITE DATA PROGRAM (UWRITE)

Chapter 4

SYSTEM EVALUATION

4.1 Bandwidth and Voltage Level Tolerance-

As stated previously, the bandwidth limiting portion of the system is the recorder. The system was evaluated by running it at two different baud rates (417 and 833), and by varying the output voltage from the recorder. The technique used to evaluate the system was to transmit a continuous stream of AAH (10101010) words into the tape and read the tape while checking for errors. The evaluation was done using two recorders, an inexpensive (approximately $20) Concord, Model F-50 recorder and a relatively expensive (approximately $200) Optisonics, Model 10-6401-01 recorder.

Both recorders performed well and were able to record and play back ten minutes of error free data transmission at either baud rate with the output voltage held constant at two volts. This corresponds to an error rate of less than one part in 250,000 for the 413 baud rate and one part in 500,000 for the 833 baud rate.

When the output voltage level of the recorder was varied, there was a significant difference between the performance of the two recorders. The reshaped encoded data received from the tape had random pulses mixed-in with

the data pulses at output voltage levels of less than approximately 300 mv for the inexpensive recorder. On the other hand, the higher quality recorder did not exhibit the problem until the voltage level was reduced to about 200 mv. As the output voltage level was increased, and the output amplifier of the recorder became saturated (flattening of the peaks of the output waveform), the narrow portion of the data interval began to spread. This distortion of the data occurred at about 3.5 volts (pp) for the low priced recorder, and approximately 10 volts (pp) for the better recorder.

4.2 Speed Variation Tolerance-

The method used to evaluate the effect of drive motor speed variations was to put a 50 ohm trimpot in series with the drive motor, which made continuous motor speed variation possible from zero rpm to full speed. The up/down counter in the decoding circuit was designed to be large enough to allow motor to run as low as twenty-five percent of full speed. This experiment was only done on the low-priced recorder, and errors began to occur in the received data when the motor speed was adjusted below seventy-five percent of full speed. The failure mechanism was very similar to that which occurred at reduced voltage levels, that is, random pulses mixed in with the data pulses.

Chapter 5

CONCLUSION

5.1 Summary-

A cassette recorder interface to an S-100 bus has been designed and evaluated. It works well on both high and low priced recorders, but more care must be taken when adjusting the output voltage level of the recorder if an inexpensive one is used. The system appears to be insensitive to slow speed variations of as much as twenty-five percent. Some further work should be done to evaluate the upper-limit of data transmission rates. In addition, it would be desirable to check the pattern sensitivity of the system by recording some data other than AAH (10101010)

5.2 Interpretation-

The sensitivity to recorder output voltage level is probably due to the signal to noise ratio for low output voltage and due to the drive capability of the output amplifier for the higher voltages. The reason that the data is jumbled at the slower motor speeds could be that the induced head voltage which is proportioned to $d\phi/dt$ (change in magnetic flux per unit time) is reduced because of the longer time interval that elapses while the

data is being read. This point could be evaluated further by varying both output voltage and motor speed simultaneously.

Chapter 6

REFERENCES

1. MITS Inc., ALTAIR 8800 Operators Manual, 1975

2. I. Rampil and J. Breimeir, "The Digital Cassette Subsystem: Part 1, Digital Recording Background and Head Interface Electronics," BYTE, Vol. 2, No. 2, Feb. 1977, pp. 24-31

3. I. Rampil and J. Breimeir, "The Digital Cassette Subsystem: Part 2, Digital Data Formats and System Considerations," BYTE, Vol. 2, No. 3, March 1977, pp.38-48

4. Intel Corp., 8080 Microprocessor Systems User's Manual, September 1975

5. Texas Instruments Inc., The TTL Data Book for Design Engineers, second edition, 1976

6. Signetics Corp., Signetics Analog Manual, 1976

7. L. Smith, "Using the 8251 Universal Syncronous/ Asyncronous Receiver/Transmitter," Intel Application Note AP-16, 1976

Appendix A

FOURIER ANALYSIS

Since both pulse-width modulated signals are the same signal (logical "1" is inverted logical "0" except for the 180^{o} phase inversion. Therefore, this analysis is only done for the logical "1" waveform. The equation used to generate the representation of the signal is:

$$f(x) = a_o/2 + \sum_{n=1}^{\infty} a_n \cos(n\pi t/L) \qquad (1)$$

normalized to a frequency of 1 Hz ($L=\frac{1}{2}$)

$$f(x) = a_o/2 + \sum_{n=1}^{\infty} a_n \cos(2n\pi t) \qquad (2)$$

the coefficients are given by the equations:

$$a_o = 4\int_0^{1/3} (x)\, dx = 4\, x \Big|_0^{1/3} = 4/3 \qquad (3)$$

$$a_n = 4\int_0^{1/3} \cos(2n\pi x)\, dx = 4\, \frac{\sin(2n\pi x)}{2n\pi} \Big|_0^{1/3}$$

$$an = \frac{2\, \sin(2n\pi x/3)}{n\pi} \qquad (4)$$

Equations (2), (3) and (4) were used to generate the waveforms shown in Figure 1.

Appendix B

ESTIMATED SYSTEM COST

Type of Component	No. of Packages Required	Estimated Cost per Package	Estimated Cost for Required Quantity
basic gates	11	$ .35	$ 3.85
tri state buffers	2	.70	1.40
flip-flops	4	.40	1.60
binary counters	4	1.00	4.00
up/down counters	2	1.50	3.00
USART	1	10.00	10.00
voltage regulator	1	1.25	1.25
op amp (dual)	1	.50	.50
resistors	23	.03	.69
capacitors	25	.10	2.50
capacitors (electrolytic)	4	.15	.60
14,16 pin sockets	24	.25	6.00
28 pin socket	1	1.00	1.00
8 pin socket	1	.25	.25
vector board 8800V	1	5.00	5.00
zener diodes	2	1.00	2.00
transistors	3	.30	.90
TOTAL.....			$ 44.54

Table I

SYSTEM PARTS LIST

Part Number	Manufacturers Number
IC 1, 2, 22	74LS04
IC 3	74LS32
IC 4	74LS02
IC 5	74LS30
IC 6, 7	74LS367
IC 8, 14, 19, 20	74LS109
IC 9, 25	74LS00
IC 10	74LS293
IC 11	1458 (Signetics)
IC 12, 13	74LS193
IC 15, 24	74LS123
IC 16, 17, 18	74LS161
IC 21	8251 (Intel)
IC 23	74LS11
IC 26	7805
R 1, 5, 7, 8	10K ohm
R 2	150K "
R 3, 6	1M "
R 4	100K "
R 9	2.2M "
R 10	47K "

Part Number	Manufacturers Number
R 11, 14, 15, 18	5.6K ohm
R 12	470 "
R 13	100 "
R 16, 17	56 "
Rp	2.2K "
C 1	.01 Microfarad
C 2	1.0 "
C 3, 4, 5, 6, 8, 10, 15, S	0.1 "
C 7, 9, 11, 12	35 "
C 13, 14	4700 Picofarad
T 1	2N4360
T 2, 3	2N2222
D 1, 2	
Vector Board 8800V	

Appendix C

DESCRIPTION OF USART INPUT/OUTPUT PINS

pin 1 - DB2 (I/O) - Bit 2 of the data bus

pin 2 - DB3 (I/O) - Bit 3 of the data bus

pin 3 - RXD(I) - Receiver data. Characters are received serially on this pin and are assembled into parallel characters.

pin 4 - GROUND

pin 5 - DB4 (I/O) - Bit 4 of the data bus

pin 6 - DB5 (I/O) - Bit 5 of the data bus

pin 7 - DB6 (I/O) - Bit 6 of the data bus

pin 8 - DB7 (I/O) - Bit 7 of the data bus

pin 9 - $\overline{\text{TXC}}$(1) - Transmitter clock. This clock controls the rate at which characters are transmitted by the USART. In the syncronous mode, RXC is equivalent to the baud rate. In the asyncronous mode, $\overline{\text{TXC}}$ is 1, 16 or 64 times the baud rate and is preselected by the mode instruction. Data is shifted out of the USART on the falling edge of the $\overline{\text{TXC}}$.

pin 10- $\overline{\text{WR}}$(I) - A low on this input causes the USART to accept data on the data bus as either a command or as a data character.

pin 11- $\overline{CS}$(I) - Chip select. A low on this input enables communication between the USART and the CPU.

pin 12- C/$\overline{D}$(I) - Control/Data. A high on this input tells the USART that the current operation is either a command or status read. A low indicates that the operation is either a read or write data.

pin 13- $\overline{RD}$(I) - A low on this input causes the USART to gate either status or data onto the data bus.

pin 14- RXRDY(O) - Receiver ready. This output goes high to indicate that a character has been received and is ready to be transferred to the CPU. RXRDY will only be asserted if the RXE bit in the command instruction has been set. RXRDY can be read by using a status read operation. RXRDY will be reset when a character is read.

pin 15- TXRDY(O) - Transmitter ready. This output signals the CPU that the USART is ready to accept the data character from the CPU. TXRDY can be read by the CPU by using the status read operation. The status bit will be asserted whenever the XMIT DATA/CMD buffer is empty

but the TXRDY pin will only be enabled is $\overline{CTS}$ =0 and TXEN=1. TXRDY will be reset when the USART receives a character from the program.

pin 16- SYNDET(I/O) - Syne detect. This line is used only in the syncronous mode. It is an output in the internal sync mode indicating that the USART has identified a (or two) sync characters on the incoming data line. In the external sync mode, it is used as an input to tell the CPU that syncronization has been established.

pin 17- CTS(I) - Clear to send. A low on this input enables the USART to transmit data. CTS is normally generated in response to $\overline{RTS}$.

pin 18- TXE(0) - Transmitter empty. A high on this output indicates that the parallel to serial converter in the transmitter is empty.

pin 19- TXD(0) - Transmit data. Serial output line of the USART.

pin 20- CLK(I) - The clock input is connected to Ø2 clock (TTL) of the computer and must be 30 times greater than the data rate for the syncronous mode and 4.5 times greater than the data rate in the asyncronous mode.

pin 21- RESET(I) - A high on this input clears the

USART and returns it to the idle mode.

pin 22- $\overline{DSR}$(I) - Data set ready. This input is a general purpose input signal that can be tested as bit 7 of the status register.

pin 23- $\overline{RTS}$(0) - Request to send. General purpose output signal that can be asserted by setting bit 5 of the command instruction.

pin 24- $\overline{DTR}$(0) - Data terminal ready. This is a general purpose output signal that can be set low by setting bit 1 of the command instruction.

pin 25- $\overline{RXC}$(I) - Receiver clock. This clock controls the data rate of characters to be received by the USART. The relationship between the clock and baud rate is the same as for $\overline{TXC}$. Data is sampled by the USART on the rising edge of $\overline{RXC}$.

pin 26- VCC - 5 volt power supply

pin 27- DBØ(I/O) - Bit Ø of the data bus

pin 28- DB1(I/O) - Bit 1 of the data bus

Appendix D

BIBLIOGRAPHICAL DATA: RICHARD G. McMAHON

Mr. McMahon was born on December 9, 1944. His parents are Kathryn A. and John R. McMahon of Philadelphia. Mr. McMahon received a Bachelor of Electrical Engineering Degree from the University of Delaware in Newark, Delaware. From 1970 until 1976, he was employed as an engineer in the Semiconductor Materials Engineering Department at the Western Electric Company in Allentown, PA. In 1976, he was transferred to the Product Line Planning Organization at the same location and in 1978 he was promoted to his current position of Department Chief-Corporate Product Line Planning at the Western Electric Corporate headquarters in New York City. Mr. McMahon is a member of Eta Kappa Nu.