DEVELOPMENT OF THE MINITEL PERIPHERAL PORT ADAPTOR (MPPA)

By Johann Strauss

Submitted in part fulfillment of the requirements laid down

for the

Master's Diploma

in the School of Electrical Engineering

at the Cape Technikon

September 1992

Saponet Research And Development Centre

Telkom

DECLARATION

I declare that the contents of this thesis represents my own work and the opinions contained here are my own. It has not been submitted before for any examination at this or any other institute.

J.Strauss

(Signature)

Abstract

This thesis describes how an adaptor board was developed to enable serial devices, such as modems and serial printers operating on RS-232 signals, to be used in conjunction with the MINITEL terminal. Furthermore it enables parallel Centronix interfaces to be used in conjunction with the MINITEL terminal.

The revolutionary 87C751 microprocessor was fully researched, and implemented in the project.

Two marketable products emerged during the course of the project:

1. The 8031-processor solution
2. The 87C751-processor solution

Opsomming

Hierdie verhandeling beskryf hoe 'n apparaat ontwikkel is wat series-werkende toestelle, soos modems en series-drukkers wat RS-232 seining gebruik, in staat stel om aan die Minitel terminaal gekoppel te word. Verder stel dit ook parallel Centronix koppelvlakke in staat om aan die Minitel terminaal gekoppel te word.

Die revolusionere 87C751 mikroverwerker is ten volle nagevors en in hierdie projek geimplementeer.

Twee bemarkbare produkte het voortgespruit uit hierdie projek:

1. Die 8031 oplossing
2. Die 87C751 oplossing

CONTENTS

1. Objective.

The objective of this project was to design and build a marketable product that would promote the MINITEL terminal by overcoming the following limitations:

a) The single serial port that the MINITEL is manufactured with.
b) The TTL orientated design of the MINITEL.

The above-mentioned limitations restricted the choice of peripheral devices that the Minitel could communicate with.

The improved MINITEL was to comply to the following specifications:

1. Enable the usage of serial RS-232 printers or modems.
2. Enable the usage of parallel CENTRONIX printers.
3. Have the capability of buffering at least 8K of data during parallel operation.
4. Have as small as practically possible physical dimensions.
5. Be as reasonably priced as possible to make it affordable for all potential MINITEL users.

2. Introduction

The "MINITEL" terminal as it is known in South Africa is manufactured by ALCATEL in France.

The main objective of MINITEL is to provide a cheap service to the South African public, which will make it possible for anyone to afford BELTEL and other services offered by TELKOM SA, such as EASY-ACCESS.

Services such as BELTEL, were amongst other things implemented to promote the X25 Packet Switched Network of TELKOM SA. This service is a major source of income for Telkom. It is therefore of extreme importance to Telkom that the MINITEL terminal be made as attractive as possible to the South African public.

A major disadvantage of the Minitel terminal is that its peripheral port supports only serial communication at TTL levels (+5V and 0V). Very few printers and modems operate at these levels and a large number of potential Minitel users are already in possession of a parallel printer which unfortunately cannot interface to the Minitel terminal.

The need therefore arose to develop a cheap adaptor, that could fit onto the peripheral port of the MINITEL terminal and would make it possible to connect to serial devices such as modems via a RS-232 interface or to communicate with a PC via the COMMS port.

In effect the MINITEL would operate as a cheap, RS-232 compatible, asyncronous terminal. This would also make the MINITEL terminal compatible to most serial printers on the market.

The most important feature of such an adaptor would be to make it possible for the MINITEL terminal to print to any parallel printer.

Two solutions are described in this thesis:

1. The first is the hardware solution, utilising an 8031 microprocessor with a relatively small assembler program in EPROM. Data buffering is done by including 8K of RAM into the design. Configuration is done by means of DIP switches. The PC-board is mounted on the peripheral backplane of the MINITEL terminal and once installed, only the serial and parallel port connectors are accessable to the user.

2. The second is the software solution, utilising only an 87C751 microprocessor with an assembler program. All data buffering is done by utilising internal RAM of the processor. XON/XOFF flow-control is implemented.
Configuration is done by means of software routines, which transmit messages to the MINITEL screen prompting the user for setup information. The adaptor then processes the instructions entered by the user on the MINITEL keyboard.

Throughout the remainder of this document there will be distinguished between the two versions by referring to either the 8031 or the 87C751 model.

In cases where no distinction is made between the two versions it should be assumed that that part of the operation is identical for both versions.

3. Operation of the MPPA.

3.1 The 8031 Model.

3.1.1 Configuration.

Configuration is done by setting the dipswitches according to the table in the operating instructions. This must be done while the adaptor is powered down. After powering up the MPPA, it tests the settings of the dip switches and sets up the speed and other selectable parameters accordingly.

3.1.2 Serial Operation.

During serial operation, data received from the serial interface is converted from RS-232 to TTL levels to be compatible with the main Minitel logic levels.
Data transmitted from the main Minitel circuitry is converted from TTL levels to RS-232 levels and then transmitted via the serial interface.
Asynchronous operation for speeds up to 9600 Bps is supported.

3.1.3 Parallel Operation.

The built-in UART of the 8031 processor is used to capture serial data from the Minitel and convert it to parallel format.
Each byte from the SBUF register is then temporarily sent to the external RAM buffer. During idle periods, bytes are fetched from RAM and placed on the parallel databus via Port 1 of the 8031.

Bytes are then latched into the parallel device by pulsing the STROBE line.

During parallel operation the BUSY line from the parallel device is continually monitored. When the BUSY line goes active, no data is sent to the parallel device, whilst additional data from the Minitel is buffered in the 8k buffer. If the 8k buffer fills up then additional data from the Minitel will overwrite the buffered data, beginning at the lowest RAM location. No direct flowcontrol exists between the Minitel and the MPPA and therefore a maximum of 8k data can be stored.

3.2 The 87C751 Model.

3.2.1 Configuration.

After power-up, the Minitel speed defaults to 1200Bps.
To simplify use of the adaptor, configuration therefore takes place at 1200Bps.

After power up the MPPA waits for 5 seconds to enable the Minitel to warm up before configuration commences. It then transmits pre-programmed messages to the Minitel screen, prompting the user for input from the Minitel keyboard. Since the Minitel currently only supports even parity, the parity bit of each byte needs to be adjusted before it is transmitted to the Minitel. The byte is then clocked out to the Minitel in serial format. After transmis-

sion of the complete message, the MPPA waits for a response from the user. This response is shifted into a register (bit by bit), interpreted by the MPPA and according to the response received the correct flags are set internally. In the case of an invalid response from the user, the previous message is retransmitted. After the user has completed configuration of the adaptor he is prompted to set up the speed of the Minitel the same as the MPPA and the operational routine starts.

3.2.2 Serial Operation.

During serial operation, data received from the serial interface is converted from RS-232 to TTL levels and sent to P0.0 on the processor. The parallel port is disabled and data is copied from P0.0 to P0.1 which in turn is connected to RXD on the Minitel. Data transmitted from the Minitel terminal is converted from TTL levels to RS-232 levels and then sent to TXD on the serial interface.

3.2.3 Parallel Operation.

Serial data received from the Minitel terminal is shifted into a register (bit by bit) by the processor and once a full byte has been received, it is transmitted to the parallel data bus via Port 3 on the processor. The strobe line is then pulsed to latch the data into the parallel device's receive buffer.

During parallel operation the BUSY line from the parallel device

is continuously monitored. If the BUSY line goes active, data from the Minitel is stored in the internal RAM of the processor. When the busy line goes inactive, the data stored in internal RAM is transmitted to the parallel device. While the internal RAM contains buffered data all subsequent data bytes from the Minitel is sent to internal RAM from where it is transmitted to the parallel port. This is done to ensure that "first in" data from the Minitel is "first out" to the parallel port. Since parallel transmission from the processor to the parallel device is considerably faster than serial reception from the Minitel, the internal RAM buffer is rapidly depleted and transmission can commence normally to the parallel device.

In the case of printers, a carriage return would cause the busy line to go active for a short period during which the internal RAM will fill up partially and the process described in the previous paragraph takes effect.

Most printers have a data buffer of their own. If, however, this buffer becomes full the printer will activate the BUSY line until a certain percentage of its buffer has been emptied before it accepts additional data. In the event of this happening the internal RAM buffer, being very small (approximately 25 bytes), will also fill up. When the MPPA detects that its internal buffer is almost full it waits for the startbit of the next byte from the Minitel and while clocking that byte into internal RAM it clocks out an XOFF to the Minitel's RXD pin, sharing the timing used to clock in the byte from the Minitel. (This is full-duplex operation.) Once the BUSY line goes inactive, the MPPA empties

its internal buffer and then transmits an XON to the Minitel. The Minitel will then continue with its transmission.

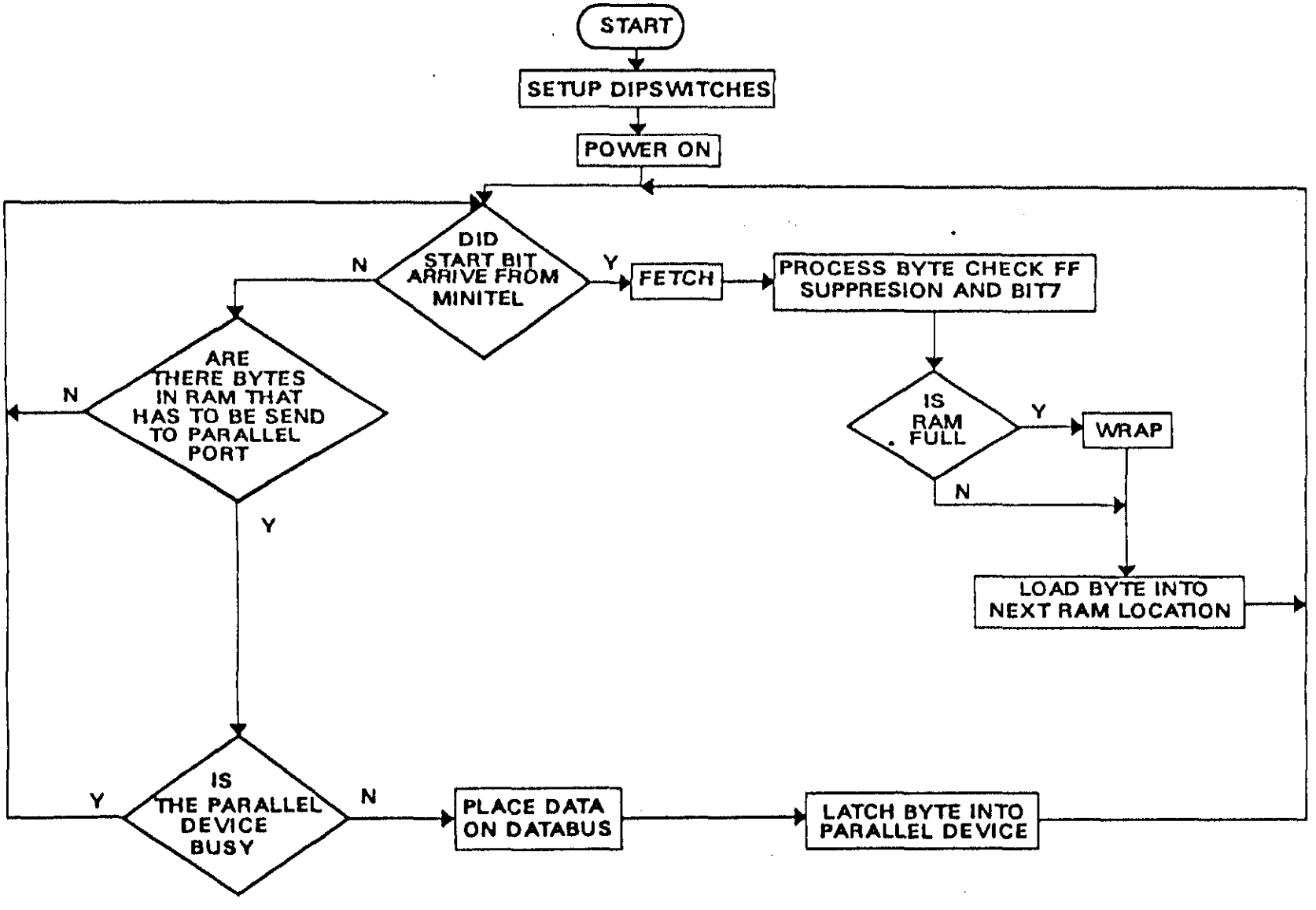


FIGURE 1: 8031 MODEL PARALLEL OPERATION

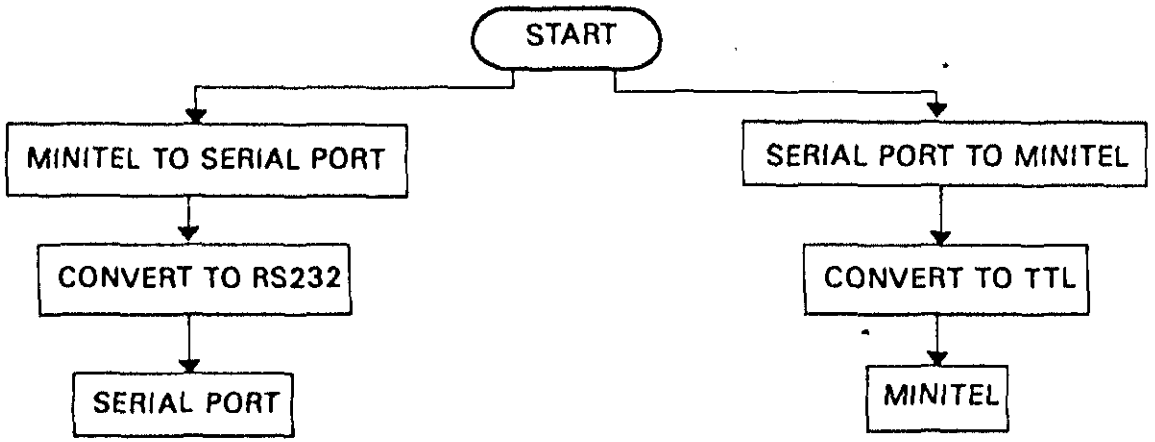


FIGURE 2: 8031 MODEL SERIAL OPERATION

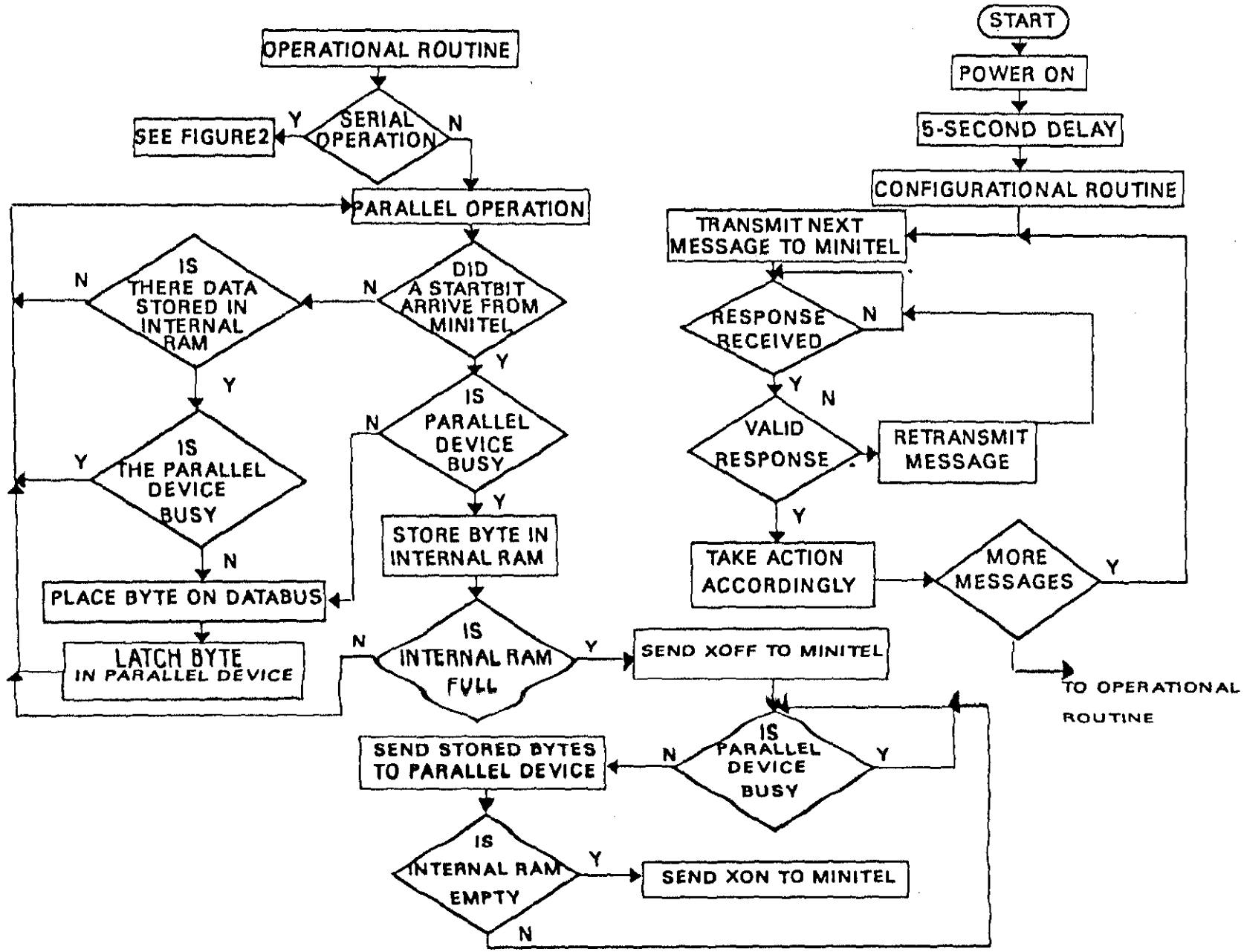


FIGURE 3: 87C751 MODEL CONFIGURATION AND OPERATION

4. The Hardware Design.

4.1 The Power Supply.

The adaptor requires +5V for correct operation. Since the MINITEL provides a +8.5V supply voltage, a voltage regulator was required to provide the required voltage. For this purpose the 7805 voltage regulator was used. The 7805 is a 5V (1A) voltage regulator.

4.2 The Parallel Printer Interface.

The printer will not accept any input unless the SELECT input line is in the proper state. The INITIALISE printer line is used to initialise the printer when the system is powered up. The correct signal must be on the line for at least 50 micro seconds. The data to the printer is placed on the eight DATA_BIT lines and the STROBE line is pulsed. The printer processes the data and sends a pulse back to the ACKNOWLEDGE line. When the acknowledge pulse is received by the processor , another character may be sent to the printer. Instead of waiting for the ACKNOWLEDGE pulse the processor may check the busy line. The processor may send characters to the printer as long as the printer is not busy. The OUT-OF-PAPER input indicates to the processor that the printer is out of paper. The SELECT input indicates to the computer that the printer is on line. The ERROR input indicates to the computer that the printer is off line, out of paper, or in another error state.

THE BUSY SIGNAL ALSO REFLECTS ALL THESE ERROR CONDITIONS.

Certain printers are sensitive to the AUTO FEED line. This line can be used to force the printer to generate an automatic line feed each time it prints a carriage return.

4.3 The Serial Port.

4.3.1 The Minitel Serial Port.

The Minitel serial port consists of the following pins:

Transmit Data (TTL voltage levels)

Receive Data (TTL voltage levels)

Signal ground

4.3.2 The RS-232 Serial Port.

The RS-232 serial port (typically a serial printer or ASCII terminal) has the following relevant pins :

Transmit Data (RS-232 voltage levels)

Receive Data (RS-232 voltage levels)

Signal Ground

4.3.3 Conversion Of Serial Port Voltage Levels.

In both the 8031 and 87C751 models, the MAX232 line driver/receiver chip was used to convert voltage levels from TTL to RS-232 and vice versa. It is relatively cheap and very convenient to use in this application since it is dirven from a single 5V supply.

4.4 The 8031 Model.

4.4.1 The 8031 Microprocessor.

An 8031 8-bit microprocessor was used which provides extensive on-chip support for one-bit variables as a separate data type, allowing direct bit manipulation and testing in control and logic systems that require Boolean processing. The Address/Data is shared on Port 0 and the lower address-byte is latched by the 74LS573. The processor specifications are:

- 128 Bytes Internal Data memory
- 32 I/O Lines (Four 8-bit ports) Providing 5 interrupts
- 2x16-bit Timers/Event Counters
- 64K Program Memory (Only 8K was used in this application)
- 5V operating voltage
- 3.5 to 12.0 MHz Oscillator Freq.(External clock source)

To minimise the use of external chips, the processor was fully utilised and not one pin was left unused.
A15 was used to chip-select either the 8K external RAM or 8K EPROM.
RXD was used for reception of data from the Minitel (Using serial interrupt)

The INT0 pin was used to pulse the STROBE line and INT1 pin was used to monitor the status of the BUSY signal.
T0, T1, A14, A13 and TXD was used to setup the speed, 8 or 7 bit data and formfeed suppression by means of dipswitches that are connected to GND when ON and connected to +5V via 10K pull-up

resistors when OFF.

A0-A12 was used to address external 8K RAM and 8K EPROM.

Port 1 connects to the parallel port databus.

An 11.0592 MHz Crystal was used as external clock source. The use of this value made the programming of timers to provide correct Baudrates considerably easier.

4.4.2 The 8K External RAM.

A 6264 8K static RAM chip was used for data buffering on the MPPA.

4.4.3 The 8K EPROM

An assembler program was written in a 2764 8K EPROM.

4.5 The 87C751 Model.

The Philips 87C751 offers the advantages of the 8051 architecture in a small package and at low cost.

The 87C751 microcontroller contains a 2Kx8 EPROM, a 64x8 RAM, 19 I/O lines, a 16-bit auto-reload counter/timer, a five-source, fixed priority level interrupt structure, an on-chip oscillator and an inter-integrated circuit serial bus interface.

The on-board inter-integrated circuit bus interface facilitates I/O and RAM expansion, access to EEPROM and processor-to-processor communication. This feature was not used in the design of the MPPA.

Other features of the 87C751 are:

Small package size (24-pin 300mm "skinny DIP")

Low power consumption

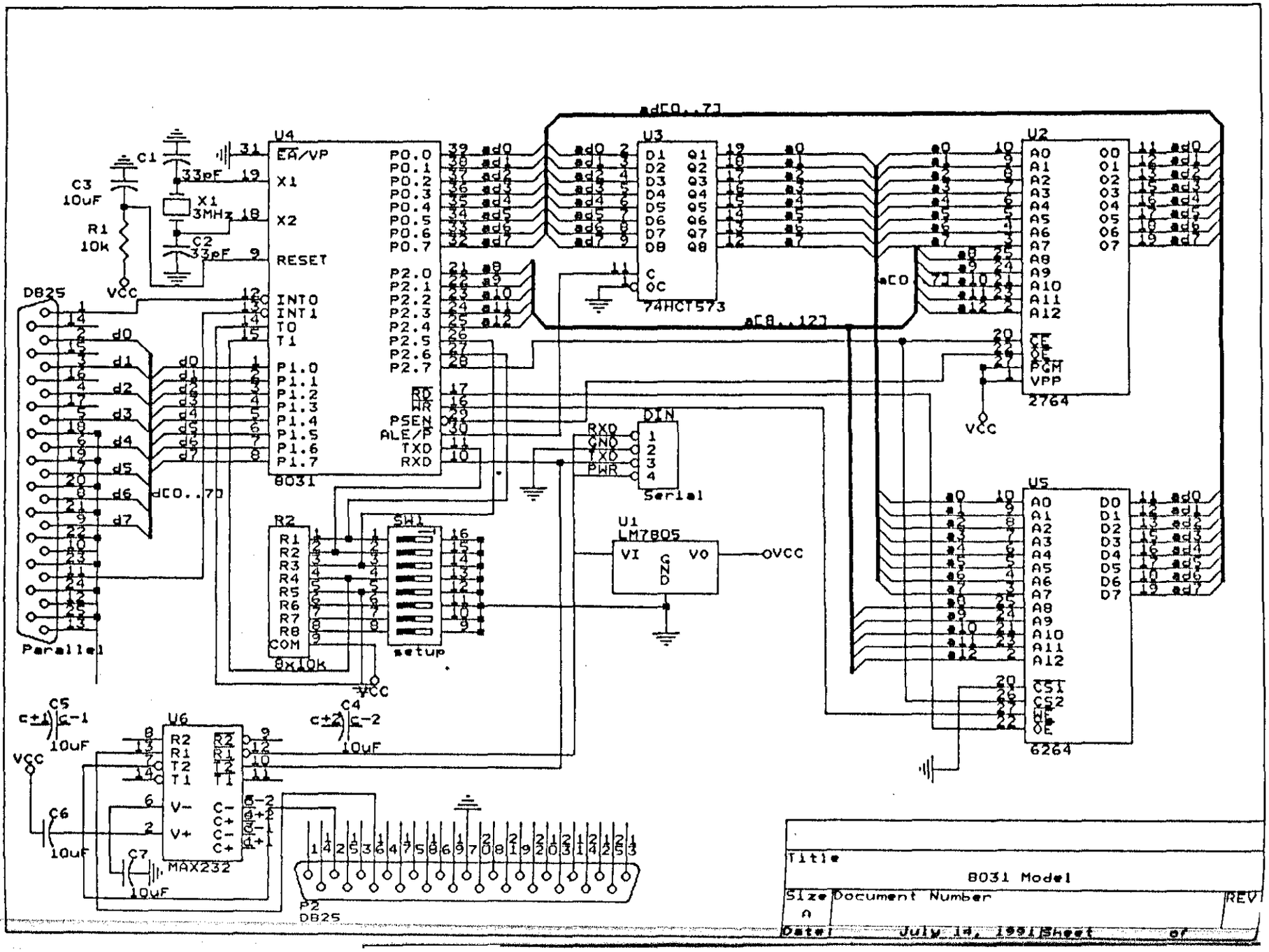
Fixed-rate timer

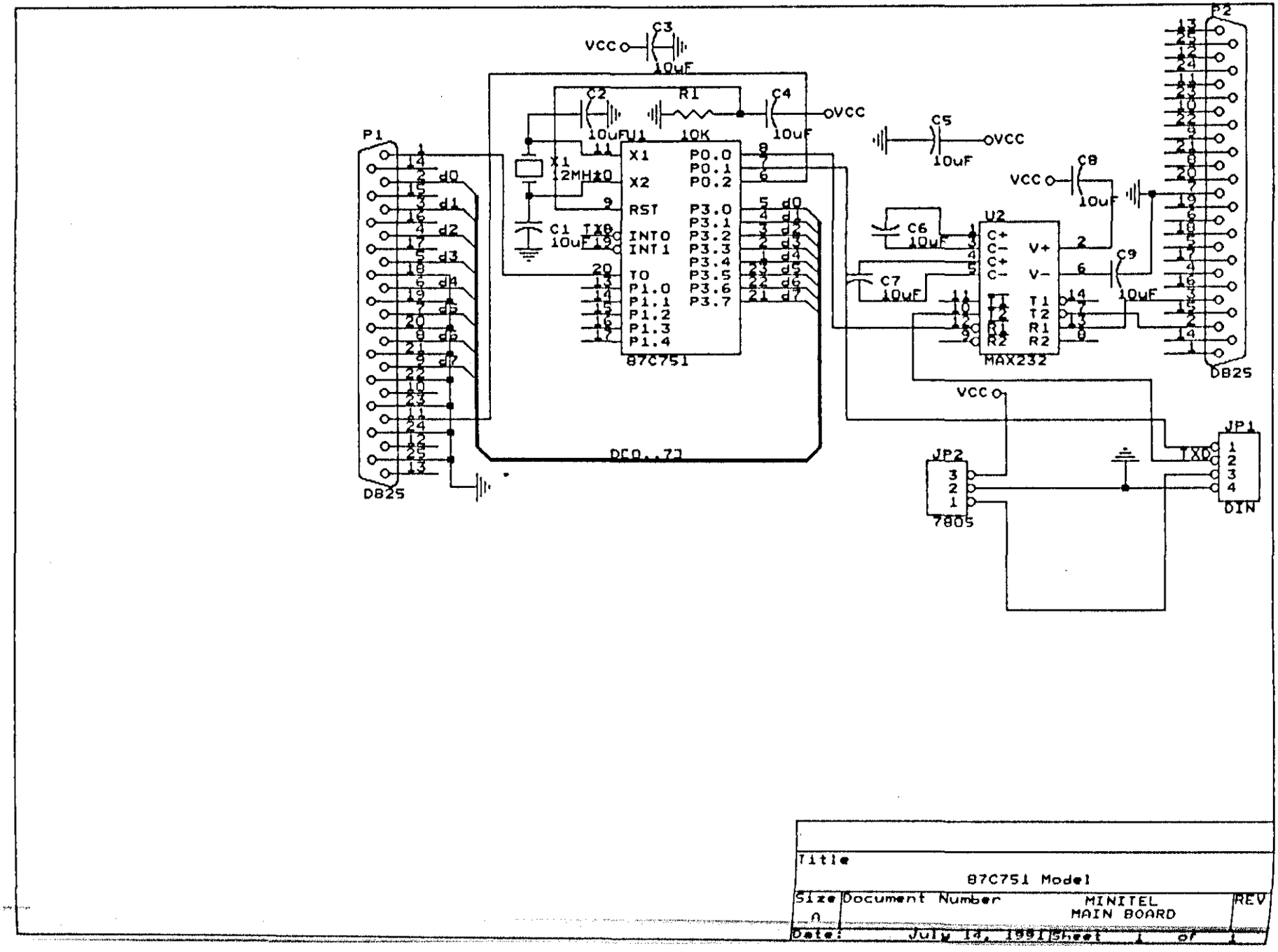
Boolean processor

Since the 87C751 has 2k EPROM on board it eliminated the need for external program memory. Furthermore the external RAM used in the 8031 model was eliminated by using internal RAM buffering and XON/XOFF flow control.

The P1.7 pin was used to STROBE data into the parallel device.
Port 3 was used as parallel data-bus.
The BUSY line status was monitored on the P0.2 pin.

In the 87C751 model the speed, number of databits and formfeed suppression is configured from the Minitel keyboard and therefore the dipswitches and resistor pack used on the 8031 model was eliminated.


U4
8031
EA/VP
X1
X2
RESET
INT0
INT1
T0
T1
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0
P0.1
P0.2
P0.3
P0.4
P0.5
P0.6
P0.7
P2.0
P2.1
P2.2
P2.3
P2.4
P2.5
P2.6
P2.7
RD
WR
PSEN
ALE/P
TXD
RXD
C1
33pF
C2
33pF
X1
3MHz
C3
10uF
R1
10k
VCC
DB25
Parallel
d0
d1
d2
d3
d4
d5
d6
d7
d0..7
R2
8x10k
R1
R2
R3
R4
R5
R6
R7
R8
COM
SW1
setup
U3
74HCT573
D1
D2
D3
D4
D5
D6
D7
D8
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
C
OC
ad0..7
a0
a8..12
aC0
U2
2764
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
O0
O1
O2
O3
O4
O5
O6
O7
CE
OE
PGM
VPP
U5
6264
D0
D1
D2
D3
D4
D5
D6
D7
CS1
CS2
WE
OE
DIN
Serial
RXD
GND
TXD
PWR
U1
LM7805
VI
GND
VO
OVCC
C5
10uF
C4
10uF
C6
10uF
C7
10uF
U6
MAX232
R2
R1
T2
T1
V-
V+
C-
C+
P2
DB25
Title
8031 Model
Size
A
Document Number
REV
Date: July 14, 1991 Sheet of



Title	87C751 Model		
Size: A	Document Number	MINITEL MAIN BOARD	REV
Date: July 14, 1991	Sheet 1 of 1		

5. The Software Design.

5.1 The 8031 Model.

The CYS8051 package was used to write and assemble the program.

5.1.1 The External Memory Map

EPROM	0000h - 7FFFh
External RAM	8000h - A000h

5.1.2 The Internal RAM Memory Map.

R0 - R7	00h - 07h
STACK	08h - 7fh

5.1.3 The Initialization Routine.

During the initialisation routine, all registers are initialised:

5.1.3.1 The PCON register.

The double Baudrate bit SMOD is set.

5.1.3.2 The IE Register.

The global interrupt enable bit EA is set .

The serial port interrupt enable bit ES is set.

This enables the processor to react on serial interrupts.

5.1.3.3 The TCON Register.

Timer 1 Run control bit TR1 is set.
Interrupt 0 Type control bit is set. (falling edge)

5.1.3.4 The TMOD Register.

Timer 1 is set up as an 8-bit auto-reload timer. The reload value is held by register TH1.

5.1.3.5 The SCON Register.

SM0 and SM1 are both set in order to set up the serial port as an 8-bit UART.
SM2 is set in order to not activate the RI flag unless a valid stopbit was received.
REN is set to enable reception on the RXD pin.

5.1.3.6 The PSW Register.

Bits RS0 and RS1 are both 0, which allocates internal RAM location 00h - 07h to R0 - R7 and lets the SP point to 08h.

5.1.2 The Speed Configuration Routine.

During this routine, the setting of the speed-dipswitches are

checked and the TH1 register is loaded with the correct value.

The following formula is used to calculate the value for TH1:
Baudrate = 2xSMOD/32 x (oscillator frequency) / 12x[256-(TH1)]

5.1.3 The Serial Interrupt Vector Routine.

When a startbit is received from the Minitel on the TXD pin of the 8031, a serial interrupt occurs.
The byte is loaded into the ACC, the dipswitches are checked to determine whether 7-bit Or 8-bit data is to be transferred to the parallel device and whether formfeed suppression takes place. The data is then stored in the next available RAM location. A check is done to see whether the external RAM is full. If it is full the next byte to be stored will overwrite the first RAM location and an overflow flag is set.

5.1.4 The Load Byte Routine.

This routine fetches the next outstanding byte from the external RAM and sends it to the parallel device if the BUSY line is inactive. The byte is latched into the parallel device after placing it on the databus. A check is done to determine whether the last RAM location has been transmitted. If true then the byte stored in the first location will be the next byte transmitted and an overflow flag is set.

5.2 The 87C751 Model.

The CYS8051 assembler package was used to write the assembler program.

Since the CYS8051 package does not support the 87C751, precautions had to be taken in order to correctly assemble the program for usage in the 87C751 microprocessor:

1. Because the 87C751 has only one external, 16-bit timer with auto-reload, the different counter registers are referred to as addresses in memory.

TL	TIMER LOW BYTE	8AH
TH	TIMER HIGH BYTE	8CH
RTL	TIMER LOW RELOAD VALUE	8BH
RTH	TIMER HIGH RELOAD VALUE	8DH

2. The timer must be started with the following command:

   SETB 8CH

3. The timer is stopped with the following command:

   CLR 8CH

Other differences between the 8031 and 87C751 are:

1. The internal RAM of the 87C751 is only 64 bytes.
2. The available EPROM memory of the 87C751 is 2k bytes.

5.2.1 The Internal RAM Memory Map

The RS0 and RS1 bits in the PSW register are both set to 0 and therefore 00h - 07h are allocated to registers R0 - R7.

00h	-	02h		R0	- R2
03h	-	2Bh		Internal data buffering	
2Ch	-	2Fh		Bit variables	
3Oh	-	34h		Byte variables	
35h	-	3Fh		Stack	

Note that no external memory map is used in the 87C751 model. (The MOVX command is not supported)

5.2.2 The 5-Second Delay Routine.

One of the problems encountered, was the fact that the MPPA started sending the first configuration screen to the Minitel screen before the Minitel has properly warmed up. This resulted in a loss of characters. To enable the Minitel to warm up sufficiently before starting configuration, a 5-second delay was incorporated into the script.

5.2.3 The Configuration Routine.

Since the Minitel speed defaults to 1200Bps, configuration is done at 1200Bps.

5.2.3.1 Transmission to Minitel.

All the messages to be sent to the Minitel are stored in EPROM. At the start of the program the datapointer (DPTR) is directed to the first byte of the message to be transmitted. That byte is

fetched and placed in the accumulator.
The parity bit is then adjusted to EVEN parity since the Minitel defaults to this setting. Reception from the Minitel is disabled during transmission.

The start bit is then transmitted to the Minitel after which the rest of the bits are shifted out of a data-register at regular intervals, controlled by the timer interrupts.
The following routine is used :

```
MOV A,XMTDAT
RRC A
MOV XMTDAT,A
MOV TXX,C
```

A register is used to count the number of bits transmitted, in order to determine when a full byte has been sent. Once the complete byte has been transmitted, the DPTR points to the next byte and the procedure is repeated. The end of a message is indicated by 0H. Once a 0H is loaded into the accumulator transmission stops and reception is enabled.

5.2.3.2 Setting Up The Timer for Transmission.

The transmission of a byte starts by clearing the TXD pin to the Minitel, directly after which the timer is started. The timer start value is the same as the timer reload value (ie. BAUDVAL). This value is calculated as follows:

1 Timer interval = (Oscillator frequency) x 12

= (1/11.0592MHz) x 12

= 1.0851 uSec

1 Bit time @ 1200Bps = 1/1200

= 0.833 mSec

BAUDVAL = -(1 Bit time @ 1200Bps) / (1 Timer interval)

= -1.0851 E-6 / 0.833 E-3

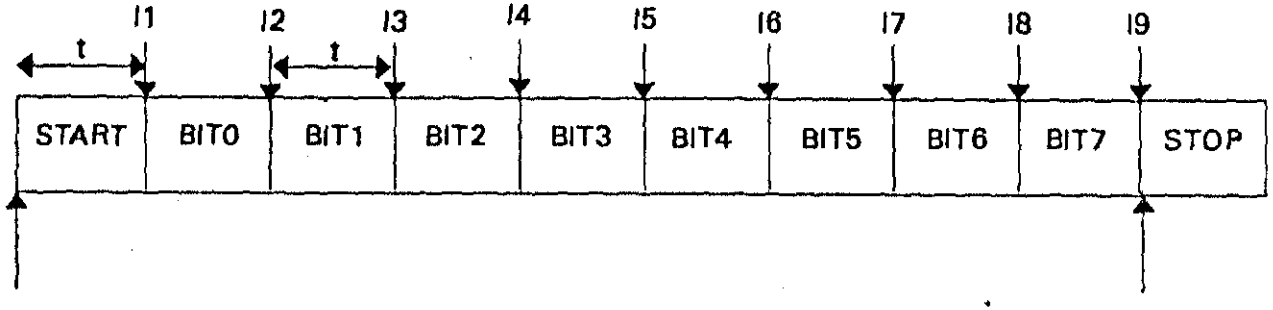
= -768

The '-' sign is necessary since the timer interval counter is an up-counter. This implies that this counter counts from FD00h to FFFFh before a timer interrupt occurs. (See figure 6)

5.2.2.3 Reception from the Minitel.

Once a full message has been transmitted to the Minitel screen, the program waits for a response from the Minitel keyboard.
An external interrupt indicates the arrival of a startbit from the Minitel. At this point further external interrupts are disabled and the internal timer is started. Each bit received is

TRANSMISSION TO MINITEL



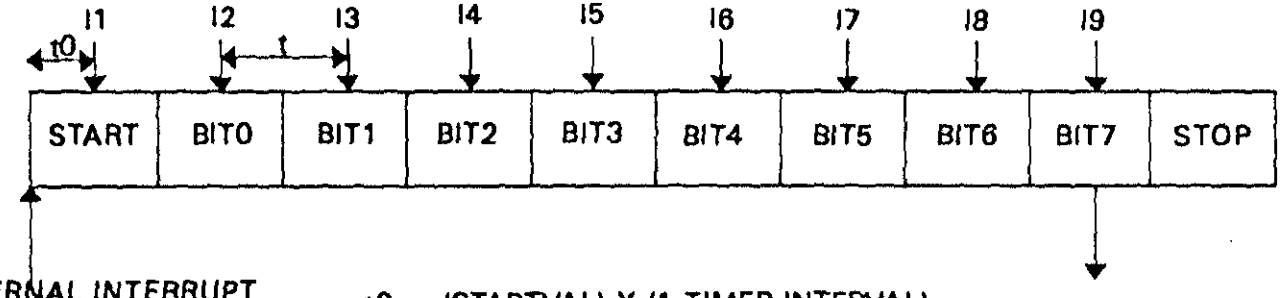
CLR TXD
START TIMER

I# = TIMER INTERRUPT NO.#

SET TXD
STOP TIMER

FIGURE6

RECEPTION FROM MINITEL



EXTERNAL INTERRUPT

START TIMER

t0 = (STARTVAL) X (1 TIMER INTERVAL)

t = (-BAUDVAL) X (1 TIMER INTERVAL)

I# = TIMER INTERRUPT NO#

STOP TIMER

FIGURE 7

clocked and shifted into a register as follows:

```
MOV A,RCVDAT
MOV C,P1.5
RRC A
MOV RCVDAT,A
```

5.2.2.4 Setting Up The Timer For Reception.

Reception of a byte starts with an external interrupt caused by the startbit. The startvalue of the timer (STRTVAL) causes the timer to wait until the middle of the startbit before it issues the first interrupt. It then uses the reload value (BAUDVAL) to provide interrupts at regular intervals (1 bit time).

STRTVAL is calculated as follows :

$$BAUDVAL/2 - DELAY$$

where DELAY is the number of timer intervals that would equal the time it takes the program to start the timer, after it receives an external interrupt.

This procedure ensures that bits are sampled at the point when half the bit has been received. (See figure 7)

Once again a count register is used to determine when the full byte has been received. On reception of a complete byte, the byte is decoded to determine whether it was a valid response or not. (Once again parity has to be taken into account.)

The program then reacts to the response by setting the necessary flags and then starts the transmission of the next message to the Minitel. Once the configuration is complete a flag is set which will pass control over to the second half of the program ie. "The Operational Routine". This flag is also used in the interrupt vector routines to indicate which routine to execute in the case of an interrupt. (The interrupt routines are different for the Configuration and Operational routines.)

5.2.4 The Operational Routine.

5.2.4.1 Serial Operation.

During serial operation the status of the Minitel's transmit is continually monitored and copied to the MAX232 line driver.
This is done by means of a JNB command in conjunction with a SETB/CLR command.

5.2.4.2 Parallel Operation.

All the bit-variables that are set up during the configuration routine is checked and the MPPA is set up accordingly.
During the operational routine serial data is received from the Minitel. The process of reception is the same as in the Configuration Routine except that the timer "start" and "reload" values are calculated according to the speed selected for operation.
Once a full byte has been received and the BUSY signal from the Centronix interface is inactive, the byte is transmitted to the

parallel port by placing the byte on the parallel databus and then pulsing the STROBE line with a CLR bit/SETB routine.

In the event of a byte being ready to be sent to the parallel interface and the BUSY line is active, the data has to be buffered.
R1 is incremented, the value stored in R1 then points to the internal RAM address where the byte has to be stored. If this value exceeds 20H, the program detects that the internal RAM is full. The byte currently being handled is then stored in internal RAM location 20H and the XOF flag is set. When the next external interrupt occurs the program detects that XOF is set and
starts to write an XOFF to the Minitel, using the same timing as the byte being received. (The byte received will be half of the bit time out of phase with the XOFF being transmitted).
Once the XOFF has been transmitted, the XOF flag is cleared and the XF flag is set which indicates to the program that an XOFF has been transmitted to the Minitel. The program will now be idle until the BUSY signal goes inactive. When this occurs, the internal RAM buffer is flushed to the parallel device, the XF flag is cleared and the XON flag is set. This flag indicates to the program that an XON must be sent to the Minitel. The timer is then started and the XON is clocked out to the Minitel. Once the XON has been transmitted, the XON flag is cleared and the XN flag is set. The XN flag indicates to the program that an XON has been sent to the Minitel and all registers and bits are set up for reception from the Minitel. The XN bit is then cleared.

5. The Program Listings.

5.1 The 8031 Model Program Listing.

```
The Cybernetic Micro Systems 8051 Family Assembler, Version 3.04
                                                              25-11-92
0000            ORG 0H
                ;%S
0000 020062             LJMP BEGIN

0023            ORG 23H
0023 020026             LJMP NEW

                NEW:
0026 C0E0               PUSH ACC
0028 C0D0               PUSH PSW
002A C082               PUSH DPL
002C C083               PUSH DPH
002E 8882               MOV DPL,R0      ;LOAD DPTR WITH NEXT RAM ADDRESS
0030 8983               MOV DPH,R1
0032 E599               MOV A,SBUF      ;LOAD RECEIVED BYTE INTO ACCUMULATOR
0034 026200             LJMP 6200H      ;LOAD A13 AND A14 WITH 1'S

                CNT1:
0037 D29C               SETB REN        ;ENABLE RECEPTION
0039 C298               CLR RI
003B D083               POP DPH         ;RESTORE REGISTERS FROM STACK
003D D082               POP DPL
003F D0D0               POP PSW
0041 D0E0               POP ACC
0043 32                 RETI            ;RETURN INTERRUPT

                NEXT:

0044 F0                 MOVX @DPTR,A    ;LOAD BYTE INTO NEXT RAM LOCATION
0045 A3                 INC DPTR        ;INCREMENT DATAPOINTER
0046 A882               MOV R0,DPL      ;STORE CONTENTS OF DATAPOINTER
0048 A983               MOV R1,DPH
004A B80008             CJNE R0,#00H,CONT  ;IS RAM FULL?
004D B9A005             CJNE R1,#0A0H,CONT
0050 7980               MOV R1,#80H        ;POINT TO FIRST RAM LOCATION
0052 EE                 MOV A,R6           ;INDICATE OVERFLOW
0053 F4                 CPL A
0054 FE                 MOV R6,A

                CONT:
0055 D29C               SETB REN        ;ENABLE RECEPTION
0057 D083               POP DPH         ;RESTORE REGISTERS FROM STACK
0059 D082               POP DPL
005B D0D0               POP PSW
005D D0E0               POP ACC
005F C298               CLR RI
0061 32                 RETI            ;RETURN INTERRUPT

                BEGIN:
0062 758780             MOV PCON,#80H
0065 75A890             MOV IE,#90H
0068 75B800             MOV IP,#00H
006B 758841             MOV TCON,#41H
006E 758920             MOV TMOD,#20H
0071 759870             MOV SCON,#70H
```

```
0074 1200D2             LCALL SPEED       ;CALL SPEED SUBROUTINE
0077 7800               MOV R0,#00H       ;DPTR=8000H
0079 7980               MOV R1,#80H
007B 908000             MOV DPTR,#8000H
007E 75D000             MOV PSW,#00H
0081 AA82               MOV R2,DPL        ;MARK LOCATION OF NEXT PRINT BYTE
0083 AB83               MOV R3,DPH
0085 7E00               MOV R6,#00H
0087 7F00               MOV R7,#00H
0089 7590FF             MOV P1,#0FFH      ;INITIALISE PORTS 1 AND 3
008C 75B0FF             MOV P3,#0FFH

                WAIT:
008F EA                 MOV A,R2
0090 C3                 CLR C
0091 98                 SUBB A,R0
0092 400C               JC BACK2          ;IS THERE A BYTE FOR PRINTER?
0094 C3                 CLR C
0095 EB                 MOV A,R3
0096 99                 SUBB A,R1
0097 4007               JC BACK2
0099 EE                 MOV A,R6
009A 9F                 SUBB A,R7
009B 7010               JNZ BACK3
009D 02008F             LJMP WAIT

                BACK2:
00A0 C3                 CLR C
00A1 EA                 MOV A,R2
00A2 98                 SUBB A,R0
00A3 4008               JC BACK3
00A5 C3                 CLR C
00A6 EB                 MOV A,R3
00A7 99                 SUBB A,R1
00A8 4003               JC BACK3
00AA 02008F             LJMP WAIT

                BACK3:
00AD C3                 CLR C
00AE 30B303             JNB P3.3,LOAD1    ;IS PRINTER BUSY?
00B1 02008F             LJMP WAIT

                LOAD1:
00B4 8A82               MOV DPL,R2        ;LOAD DPTR WITH LOCATION OF NEXT
                                          ;PRINT BYTE
00B6 8B83               MOV DPH,R3
00B8 E0                 MOVX A,@DPTR      ;SEND BYTE TO PARALLEL BUS
00B9 F590               MOV P1,A
00BB C2B2               CLR P3.2          ;LATCH BYTE INTO PRINTER
00BD A3                 INC DPTR          ;STORE LOCATION OF NEXT PRINTER BYTE
00BE AA82               MOV R2,DPL
00C0 AB83               MOV R3,DPH
00C2 BA0008             CJNE R2,#00H,CONT1  ;IS THIS THE LAST RAM LOCATION?
00C5 BBA005             CJNE R3,#0A0H,CONT1
00C8 7B80               MOV R3,#80H         ;WRAP
00CA EF                 MOV A,R7            ;INDICATE OVERFLOW
```

```
00CB F4                 CPL A
00CC FF                 MOV R7,A

                CONT1:
00CD D2B2               SETB P3.2
00CF 02008F             LJMP WAIT

                SPEED:
00D2 30B504             JNB P3.5,SPEED1       ;9600BPS ?
00D5 758DFA             MOV TH1,#0FAH
00D8 22                 RET

                SPEED1:
00D9 30B107             JNB P3.1,SPEED2       ;4800BPS ?
00DC 30B40B             JNB P3.4,SPEED3
00DF 758DF4             MOV TH1,#0F4H
00E2 22                 RET

                SPEED2:
00E3 30B408             JNB P3.4,SPEED4       ;2400BPS ?
00E6 758DE8             MOV TH1,#0E8H
00E9 22                 RET

                SPEED3:
00EA 758DD0             MOV TH1,#0D0H         ;1200BPS
00ED 22                 RET

                SPEED4:
00EE 758D40             MOV TH1,#40H          ;300BPS ?
00F1 22                 RET

6200            ORG 6200H
6200 30A502             JNB P2.5,FORM         ;8-BIT DATA ?
6203 547F               ANL A,#7FH            ;MASK BIT 7 WITH 0

                FORM:
6205 30A606             JNB P2.6,NEXT1        ;SUPPRESS FORMFEEDS?
6208 B40C03             CJNE A,#0CH,NEXT1     ;IS THIS A FORMFEED?
620B 020037             LJMP CNT1

                NEXT1:
620E 020044             LJMP NEXT
                ;%E
0000            END
```

```
The Cybernetic Micro Systems 8051 Family Assembler, Version 3.04
                                                              25-11-92



;%T   Symbol Name                    Type Value

BACK2 . . . . . . . . . . . . . . . L 00A0
BACK3 . . . . . . . . . . . . . . . L 00AD
BEGIN . . . . . . . . . . . . . . . L 0062
CNT1. . . . . . . . . . . . . . . . L 0037
CONT. . . . . . . . . . . . . . . . L 0055
CONT1 . . . . . . . . . . . . . . . L 00CD
FORM. . . . . . . . . . . . . . . . L 6205
LOAD1 . . . . . . . . . . . . . . . L 00B4
NEW . . . . . . . . . . . . . . . . L 0026
NEXT. . . . . . . . . . . . . . . . L 0044
NEXT1 . . . . . . . . . . . . . . . L 620E
SPEED . . . . . . . . . . . . . . . L 00D2
SPEED1. . . . . . . . . . . . . . . L 00D9
SPEED2. . . . . . . . . . . . . . . L 00E3
SPEED3. . . . . . . . . . . . . . . L 00EA
SPEED4. . . . . . . . . . . . . . . L 00EE
WAIT. . . . . . . . . . . . . . . . L 008F

;%Z

00 Errors  (0000)
```

5.2 The 87C751 Program Listing

```
The Cybernetic Micro Systems 8051 Family Assembler, Version 3.04
                                                        25-11-92
                ;*****************************************************************
                ;*                                                               *
                ;*          THE MINITEL PERIPHERAL PORT INTERFACE MODULE          *
                ;*                                                               *
                ;*****************************************************************
                ;
                ;
                ;
                ;
                ;
                ;*****************************************************************
                ;****************DECLARATION OF VARIABLES*************************
                ;*****************************************************************

FD00 =          BAUDVAL     EQU     -768    ;BIT TIME AT 1200Bps
FE93 =          STRTVAL     EQU     -365    ;HALF OF BIT TIME AT 1200Bps
0030 =          XMTDAT.     DATA     30H    ;DATA TO BE TRANSMITTED
0034 =          RCVDAT      DATA     34H    ;DATA RECEIVED
0033 =          BITCNT      DATA     33H    ;COUNTS THE BITS RECEIVED
0032 =          LOOPCNT     DATA     32H
0031 =          CNT         DATA     31H    ;COUNT REG.FOR 5 SEC DELAY
0077 =          TXFLAG      BIT      77H    ;INDICATES TRANSMISSION
0076 =          RXFLAG      BIT      76H    ;INDICATES RECEPTION
0075 =          RXERR       BIT      75H    ;INDICATES ERROR CONDITION
0074 =          RCVRDY      BIT      74H    ;
0073 =          GO          BIT      73H    ;INDICATES THAT TRANSMISSION OF
                                            ;SUBSEQUENT SCREENS COMMENCES.
0081 =          TXX         BIT      P0.1   ;TRANSMIT DATA PIN
0072 =          FIN         BIT      72H    ;INDICATES THE INITIAL
                                            ;/FINAL SETUP SCREEN.
0071 =          TSET        BIT      71H    ;INDICATES THAT THE SPEED
                                            ;SCREEN IS DISPLAYED
0078 =          TWO         BIT      78H    ;INDICATES 2400 BPS
007F =          THR         BIT      7Fh    ;INDICATES 300 BPS
007E =          ONE         BIT      7EH    ;INDICATES 1200 BPS
007D =          FOU         BIT      7DH    ;INDICATES 4800 BPS
007C =          NIN         BIT      7CH    ;INDICATES 9600 BPS

0070 =          PPS         BIT      70H    ;INDICATES THAT THE PORT SELECT
                                            ;SCREEN IS DISPLAYED
007A =          RAM1        BIT      7AH    ;IF SET INDICATES THAT CONFIGURA-
                                            ;TION IS COMPLETE AND THAT THE
                                            ;MODULE IS NOW READY FOR OPERATION.

0079 =          PAR         BIT      79H    ;INDICATES PARALLEL OPERATION

006F =          FORM        BIT      6FH    ;INDICATES THAT THE FORMFEED SCREEN
                                            ;IS DISPLAYED
007B =          FFON        BIT      7BH    ;IF SET THEN SUPPRESS FF'S. IF
                                            ;CLEARED THEN PRINT FF'S.
006E =          REP         BIT      6EH    ;INDICATE THAT CURRENT SCREEN MUST
                                            ;BE REPEATED.
006D =          RAM         BIT      6DH    ;INDICATES THAT SETUP IS COMPLETE
006C =          FORMF       BIT      6CH    ;INDICATES THAT THE FORMFEED STATUS
```

```
                                        ;IS TO BE DISPLAYED
006B =          PORT1       BIT     6BH ;INDICATES THAT THE TYPE OF TRANS-
                                        ;MISSION IS TO BE DISPLAYE
006A =          FIN1        BIT     6AH ;INDICATES THAT FINAL MESSAGE IS TO
                                        ;BE DISPLAYED

0069 =          DEL         BIT     69H ;IF SET INDICATES THAT THE 5-SECOND
                                        ;WARMUP DELAY IS STILL IN PROGRESS


            ;*****************************************************************
            ;********************THE INTERRUPT VECTOR ROUTINES********************
            ;*****************************************************************


0000        ORG 0H
            ;%S
0000 0115           .   AJMP RESET                  ;GOTO INITIALISATION
                                                    ;ROUTINE

                                .
0003            ORG 03H
0003 307A02             JNB RAM1,SET1                   ;CHECK WHETHER CONFIG-
                                                        ;URATION IS COMPLETE
0006 C170               AJMP EEXINT0                    ;GOTO OPERATIONAL
                                                        ;INTERRUPT ROUTINE
0008 2183               SET1: AJMP EXINT0               ;GOTO CONFIGURATION
                                                        ;INTERRUPT ROUTINE

000B            ORG 0BH
000B 20696A             JB DEL,DELAY1                   ;CHECK WHETHER WARMUP
                                                        ;DELAY IS STILL IN
                                                        ;PROGRESS
000E 307A02             JNB RAM1,SET2                   ;CHECK WHETHER CONFIG-
                                                        ;URATION IS STILL IN
                                                        ;PROGRESS
0011 C1CD               AJMP TTIMR0                     ;GOTO      OPERATIONAL
                                                        ;TIMER INTERRUPT ROU-
                                                        ;TINE.


0013 2159   SET2:       AJMP TIMR0                      ;GOTO    CONFIGURATION
                                                        ;TIMER INTERRUPT ROU-
                                                        ;TINE.
```

```
                ;*****************************************************************
                ;*******************THE INITIALISATION ROUTINE********************
                ;*****************************************************************

0015 758135     RESET:      MOV SP,#35H                 ;STACK STARTS AT 35H
0018 75D000                 MOV PSW,#00H                ;CLEAR  THE  PROGRAM
                                                        ;STATUS WORD
001B 758800                 MOV TCON,#00H               ;LOAD TIMER CONTROL
                                                        ;REGISTER WITH 0H
001E 75A883                 MOV IE,#83H                 ;LOAD INTERRUPT ENABLE
                                                        ;REGISTER WITH 83H

                    ;NOTE : ALL DEFINED BITS ARE INITIALISED AS ZEROS

0021 C276                   CLR RXFLAG
0023 C277                   CLR TXFLAG
0025 C275                   CLR RXERR
0027 C274                   CLR RCVRDY
0029 C273                   CLR GO
002B C272                   CLR FIN
002D C271                   CLR TSET
002F C278                   CLR TWO
0031 C27F                   CLR THR
0033 D27E                   SETB ONE                    ;SET SPEED UP FOR 1200
                                                        ;BAUD
0035 C27D                   CLR FOU
0037 C27C                   CLR NIN
0039 C270                   CLR PPS
003B C27A                   CLR RAM1
003D D279                   SETB PAR                    ;SET PORT UP FOR PARAL-
                                                        ;LEL OPERATION
003F C26F                   CLR FORM
0041 D27B                   SETB FFON                   ;SUPPRESS FORMFEEDS
0043 C269                   CLR DEL
0045 C26E                   CLR REP
0047 C26D                   CLR RAM
0049 C26C                   CLR FORMF
004B C26B                   CLR PORT1
004D C26A                   CLR FIN1
004F 753147                 MOV CNT,#47H                ;SETUP CNT REGISTER FOR
                                                        ;47 TIMER INTERRUPTS

                ;*SETUP TIMER START VALUE AND TIMER RELOAD VALUE FOR 5-SEC DELAY*

0052 7A00                   MOV R2,#00H
0054 8A8C                   MOV 8CH,R2
0056 7A01                   MOV R2,#01H
0058 8A8A                   MOV 8AH,R2
005A 7A00                   MOV R2,#00H
005C 8A8D                   MOV 8DH,R2
005E 7A01                   MOV R2,#01H
0060 8A8B                   MOV 8BH,R2
```

```
                ;*****************************************************************
                ;************************THE 5 SECOND DELAY***********************
                ;*****************************************************************

                    DELAY:
0062 D269                       SETB DEL            ;5-SEC DELAY IN PROGRESS
0064 D28C                       SETB 8CH            ;START TIMER
0066 E531                       MOV A,CNT           ;LOAD ACCUMULATOR WITH
                                                    ;THE NUMBER OF REMAINING
                                                    ;INTERRUPTS
0068 B400F7                     CJNE A,#0,DELAY     ;EXIT DELAY ROUTINE WHEN
                                                    ;CNT REGISTER REACHES
                                                    ;ZERO.
006B C28C                       CLR 8CH             ;STOP TIMER
006D C269                       CLR DEL             ;INDICATE THAT CONFIGU-
                                                    ;RATION CAN COMMENCE.
006F C2A8                       CLR EX0             ;DISABLE RECEPTION
0071 D272                       SETB FIN            ;INDICATES THAT INITIAL SETUP
                                                    ;MESSAGE IS DISPLAYED  TO
                                                    ;REACT CORRECTLY WHEN RESPONSE
                                                    ;IS RECEIVED AFTER MESSAGE HAS
                                                    ;BEEN SENT.
0073 900335                     MOV DPTR,#MSG1      ;POINT TO MESSAGE 1
0076 2111                       AJMP PRT            ;GOTO   ROUTINE   THAT
                                                    ;STARTS OUTPUT TO MINI-
                                                    ;TEL SCREEN


                    DELAY1:
0078 1531                       DEC CNT             ;DELAY TIMER INTERRUPT
                                                    ;ROUTINE
007A 32                          RETI


                    LOOP1:
007B 207302                     JB GO,GGO1          ;IS BIT GO SET?
007E 212A                       AJMP LOOP2

                    GGO1:
0080 C2A8                       CLR EX0             ;DISABLE RECEPTION
0082 C273                       CLR GO
0084 307208                     JNB FIN,MES2        ;IS BIT FIN SET ?
0087 206E0D                     JB REP,MES4         ;IS BIT REP SET ?
008A 900335                     MOV DPTR,#MSG1      ;POINT DPTR TO MSG1
008D 2111                       AJMP PRT


                    MES2:
008F 30710C                     JNB TSET,MES5       ;IS BIT TSET SET ?
0092 9002B4                     MOV DPTR,#MSG2      ;POINT DPTR TO MESG2
0095 2111                       AJMP PRT


                    MES4:
0097 C26E                       CLR REP
```

```
0099 900335                MOV DPTR,#MSG1     ;POINT DPTR TO MESG1
009C 2111                  AJMP PR


                   MES5:
009E 306F05                JNB FORM,MES6      ;IS BIT FORM SET ?
00A1 900319                MOV DPTR,#MSG3     ;POINT DPTR TO MESG3
00A4 2111                  AJMP PRT


                   MES6:
00A6 306D05                JNB RAM,MES7       ;IS BIT RAM SET ?
00A9 9004FD                MOV DPTR,#MSG6     ;POINT DPTR TO MESG6
00AC 2111                  AJMP PRT


                   MES7:
00AE 9004A7                MOV DPTR,#MSG5     ;POINT DPTR TO MESG5
00B1 805E                  SJMP PRT


                   LOGO:
00B3 206C33                JB FORMF,LOGO1     ; FORMFEED MESSAGE ?
00B6 206B45                JB PORT1,LOGO2     ;
00B9 207E0C                JB ONE,LOGO01
00BC 207810                JB TWO,LOGO02
00BF 207F14                JB THR,LOGO03
00C2 207D18                JB FOU,LOGO04
00C5 207C1C                JB NIN,LOGO05


                   LOGO01:
00C8 900384                MOV DPTR,#MSG101
00CB D26C                  SETB FORMF
00CD 8042                  SJMP PRT


                   LOGO02:
00CF 90038D                MOV DPTR,#MSG102
00D2 D26C                  SETB FORMF
00D4 803B                  SJMP PRT

                   LOGO03:
00D6 900396                MOV DPTR,#MSG103
00D9 D26C                  SETB FORMF
00DB 8034                  SJMP PRT


                   LOGO04:
00DD 90039E                MOV DPTR,#MSG104
00E0 D26C                  SETB FORMF
00E2 802D                  SJMP PRT


                   LOGO05:
```

```
00E4 9003A7                 MOV DPTR,#MSG10
00E7 8028                   SJMP PRT


                LOGO1:
00E9 207B09                 JB FFON,LOGO101
00EC 9003D9                 MOV DPTR,#MSG107
00EF C26C                   CLR FORMF
00F1 D26B                   SETB PORT1
00F3 801C                   SJMP PRT



                LOGO101:
00F5 9003B0                 MOV DPTR,#MSG106
00F8 C26C                   CLR FORMF
00FA D26B                   SETB PORT1
00FC 8013                   SJMP PRT

                LOGO2:
00FE 207909                 JB PAR,LOGO201
0101 D26A                   SETB FIN1
0103 C26B                   CLR PORT1
0105 900456                 MOV DPTR,#MSG109
0108 8007                   SJMP PRT


                LOGO201:
010A D26A                   SETB FIN1
010C C26B                   CLR PORT1
010E 900403                 MOV DPTR,#MSG108



                PRT:
0111 C2A8                   CLR EX0             ;DISABLE RECEPTION
0113 516A                   ACALL MESS
0115 3136                   ACALL XMTBYTE
0117 D2A8                   SETB EX0            ;ENABLE RCEPTION
0119 306A04                 JNB FIN1,PRT1       ;IS BIT FIN SET?
011C C26A                   CLR FIN1
011E 8003                   SJMP PRT2


                PRT1:
0120 207211                 JB FIN,LOOGOO


                PRT2:
0123 306D04                 JNB RAM,LOOP2       ;IS BIT RAM SET
0126 D27A                   SETB RAM1
0128 A1B1                   AJMP RRESET         ;GOTO OPERATIONAL ROUTINE
```

```
                LOOP2:
012A 208C02                     JB TR0,IEN
012D D2A8                       SETB EX0


                IEN:
012F 3073F8                     JNB GO,LOOP2
0132 017B                       AJMP LOOP1


                LOOGOO:
0134 01B3                       AJMP LOGO


              ;************************THE TRANSMIT ROUTINE*********************


                XMTBYTE:
0136 2076FD                     JB RXFLAG,$             ;CHECK WHETHER RECEPTION
                                                        ;IS IN PROGRESS
0139 313F                       ACALL RSXMT             ;GOTO THE BIT ROUTINE
013B 2077FD                     JB TXFLAG,$             ;IS TRANSMISSION IN
                                                        ;PROGRESS

                ;PROGRAM WAITS FOR TIMER INTERRUPT WHILE TXFLAG IS SET.

013E 22                         RET

             ;*********************THE BIT TRANSMISSION ROUTINE****************


                RSXMT:
013F F530                       MOV XMTDAT,A            ;FETCH BYTE TO BE XMITTED
0141 75330A                     MOV BITCNT,#10          ;LOAD BITCOUNTER


             ;**************LOAD THE TIMER WITH THE VALUE FOR 1200 BAUD**


0144 7AFD                       MOV R2,#0FDH
0146 8A8C                       MOV 8CH,R2
0148 7A00                       MOV R2,#00H
014A 8A8A                       MOV 8AH,R2
014C 7AFD                       MOV R2,#0FDH
014E 8A8D                       MOV 8DH,R2
0150 7A00                       MOV R2,#00H
0152 8A8B                       MOV 8BH,R2              ;TIMER SETUP COMPLETED


0154 C281                       CLR TXX                 ;TRANSMIT STARTBIT
```

```
                EXINT0:
0183 75330A                 MOV BITCNT,#10       ;COUNT TEN BITS
0186 7A93                   MOV R2,#LOW STRTVAL ;LOAD TIMER WITH VALUE FOR
                                                 ;1200 BPS
0188 8A8A                   MOV 8AH,R2
018A 7AFE                   MOV R2,#HIGH STRTVAL
018C 8A8C                   MOV 8CH,R2
018E 7AFD                   MOV R2,#HIGH BAUDVAL
0190 8A8D                   MOV 8DH,R2
0192 7A00                   MOV R2,#LOW BAUDVAL
0194 8A8B                   MOV 8BH,R2
0196 753400                 MOV RCVDAT,#0        ;INITIATE THE RXD REGISTER
0199 C2A8                   CLR EX0              ;DISABLE EXTERNAL INTERRUPTS
019B C275                   CLR RXERR
019D D28C                   SETB 8CH             ;START TIMER
019F D276                   SETB RXFLAG          ;RECEPTION IN PROGRESS
01A1 32                     RETI


                RXBIT:
01A2 D53302                 DJNZ BITCNT,RXBUSY  ;WAS THE STOP BIT RECEIVED ?
01A5 8010                   SJMP RXBITEX


                RXBUSY:
01A7 E533                   MOV A,BITCNT
01A9 B40902                 CJNE A,#9,RXNEXT    ;IS THIS THE STOPBIT
01AC 80B7                   SJMP T0EX2


                RXNEXT:
01AE E534                   MOV A,RCVDAT
01B0 A295                   MOV C,P1.5          ;LOAD THE NEXT BIT INTO RCVDAT
01B2 13                     RRC A
01B3 F534                   MOV RCVDAT,A
01B5 80AE                   SJMP T0EX2


                RXBITEX:
01B7 C276                   CLR RXFLAG           ;RECEPTION COMPLETE
01B9 E534                   MOV A,RCVDAT
01BB 207209                 JB FIN,FINISH        ;WAS THIS THE INITIAL/FINAL
                                                 ;MESSAGE
01BE 20711E                 JB TSET,SPEED        ;WAS THIS THE SPEED MESSAGE
01C1 206F78                 JB FORM,FF           ;WAS THIS THE FORMFEED MESSAGE

01C4 207009                 JB PPS,PORT1         ;;WAS THIS THE PORT MESSAGE



                FINISH:
01C7 B45908                 CJNE A,#59H,OUT1    ;'Y' RECEIVED ?
```

```
01CA C272                       CLR FIN
01CC D26D                       SETB RAM            ;INDICATE TRANSMISSION OF LAST
                                                    ;MESSAGE
01CE 4166                       AJMP OUT

                    PORT1:
01D0 4152                       AJMP PORT

                    OUT1:
01D2 B44E06                     CJNE A,#4EH,OUT5    ;'N' RECEIVED ?
01D5 C272                       CLR FIN
01D7 D271                       SETB TSET           ;SPEED MUST BE SET UP
01D9 4166                       AJMP OUT


                    OUT5:
01DB D26E                       SETB REP            ;MESSAGE HAS TO BE
                                                    ;REPEATED
01DD 4166                       AJMP OUT


                    SPEED:
01DF B43910                     CJNE A,#39h,FOUR    ;9600 ?
01E2 D27C                       SETB NIN
01E4 C27D                       CLR FOU
01E6 C278                       CLR TWO
01E8 C27E                       CLR ONE
01EA C27F                       CLR THR
01EC C271                       CLR TSET
01EE D26F                       SETB FORM           ;FORMFEED MESSAGE IS NEXT
01F0 4166                       AJMP OUT


                    FOUR:
01F2 B4B410                     CJNE A,#0B4H,TWEE   ;4800 ?
01F5 D27D                       SETB FOU
01F7 C27C                       CLR NIN
01F9 C278                       CLR TWO
01FB C27E                       CLR ONE
01FD C27F                       CLR THR
01FF C271                       CLR TSET
0201 D26F                       SETB FORM          ;FORMFEED MESSAGE IS NEXT
0203 4166                       AJMP OUT


                    TWEE:
0205 B4B20E                     CJNE A,#0B2H,EEN    ;2400 ?
0208 D278                       SETB TWO
020A C27C                       CLR NIN
020C C27D                       CLR FO
020E C27F                       CLR THR
0210 C271                       CLR TSET           ;FORMFEED MESSAGE IS NEXT
0212 D26F                       SETB FORM
0214 4166                       AJMP OUT
```

```
                EEN:
0216 B4B110                 CJNE A,#0B1H,DRIE ;1200 ?
0219 D27E                   SETB ONE
021B C27C                   CLR NIN
021D C278                   CLR TWO
021F C27D                   CLR FOU
0221 C27F                   CLR THR
0223 C271                   CLR TSET
0225 D26F                   SETB FORM     ;INDICATE FORMFEED MESSAGE IS NEXT
0227 4166                   AJMP OUT



                DRIE:
0229 B4333A                 CJNE A,#33H,OUT ;300 ?
022C D27F                   SETB THR
022E C27C                   CLR NIN
0230 C278                   CLR TWO
0232 C27E                   CLR ONE
0234 C27D                   CLR FOU
0236 D26F                   SETB FORM     ;INDICATE FORMFEED MESSAGE IS NEXT
0238 C271                   CLR TSET
023A 4166                   AJMP OUT



                FF:
023C B45908                 CJNE A,#59H,OUT3 ;'Y' RECEIVED ?
023F D27B                   SETB FFON        ;SUPPRESS FORMFEEDS
0241 C26F                   CLR FORM
0243 D270                   SETB PPS         ;INDICATE PORT MESSAGE IS NEXT
0245 4166                   AJMP OUT

                OUT3:
0247 B44E1C                 CJNE A,#4EH,OUT  ;'N' RECEIVED ?
024A C27B                   CLR FFON         ; DO NOT SUPPRESS FORMFEEDS
024C C26F                   CLR FORM
024E D270                   SETB PPS         ;INDICATE PORT MESSAGE IS NEXT
0250 4166                   AJMP OUT

                PORT:
0252 B45008                 CJNE A,#50H,OUT4 ; 'P' RECEIVED ?
0255 D279                   SETB PAR         ; PARALLEL OPERATION
0257 C270                   CLR PPS
0259 D272                   SETB FIN         ;INITIAL/FINAL MESSAGE IS
                                             ;NEXT
025B 4166                   AJMP OUT



                OUT4:
025D B45306                 CJNE A,#53H,OUT  ;'S' RECEIVED ?
```

```
0260 C279                       CLR PAR          ; SERIAL OPERATION
0262 C270                       CLR PPS
0264 D272                       SETB FIN         ;INITIAL/FINAL MESSAGE IS
                                                 ;NEXT

                OUT:
0266 D273                       SETB GO          ;SET TRANSMISSION START FLAG
0268 2163                       AJMP T0EX1


             ;********************THE FETCH BYTE ROUTINE*******************


                MESS:
026A C0E0                       PUSH ACC         ;SAVE CONTENTS OF ACCUMULATOR
026C 7800                       MOV R0,#0        ;R0 VALUE INDICATES WHICH
                                                 ;BYTE IN THE MESSAGE IS
                                                 ;CURRENTLY TRANSMITTED.


                MES1:
026E E8                         MOV A,R0
026F 93                         MOVC A,@A+DPTR   ;LOAD BYTE TO XMIT TO MINI-
                                                 ;TEL

            ;*******************THE PARITY ROUTINE*****************************

             ;COUNT BITS IN BYTE THAT ARE SET. IF IT IS AN ODD NUMBER THEN

             ;BIT7 MUST BE SET. IF IT IS AN EVEN NUMBER THEN BIT7 MUST BE

             ;RESET.

0270 7F00                       MOV R7,#00H
0272 30E001                     JNB ACC.0,BIT0          ;IS BIT 0 = 0 ?
0275 0F                         INC R7




                BIT0:
0276 30E101                     JNB ACC.1,BIT1          ;IS BIT 1 = 0 ?
0279 0F                         INC R7


                BIT1:
027A 0F                         INC R7


                BIT2:
027B 30E301                     JNB ACC.3,BIT3          ;IS BIT 3 = 0 ?
027E 0F                         INC R7
```

```
                BIT3:
027F 30E401             JNB ACC.4,BIT4          ;IS BIT 4 = 0 ?
0282 0F                 INC R7

                BIT4:
0283 30E501             JNB ACC.5,BIT5          ;IS BIT 5 = 0 ?
0286 0F                 INC R7

                BIT5:
0287 30E601             JNB ACC.6,BIT6          ;IS BIT 6 = 0 ?
028A 0F                 INC R7

            ;CHECK IF THERE IS AN EVEN OR ODD NUMBER OF ONES IN THE BYTE

                BIT6:
028B BF0104             CJNE R7,#01,BIT7
028E D2E7               SETB ACC.7              ;SET THE PARITY BIT
0290 41A9               AJMP PARITY

                BIT7:
0292 BF0304             CJNE R7,#03,BIT8
0295 D2E7               SETB ACC.7              ;SET THE PARITY BIT
0297 41A9               AJMP PARITY

                BIT8:
0299 BF0504             CJNE R7,#05,BIT9
029C D2E7               SETB ACC.7              ;SET THE PARITY BIT
029E 41A9               AJMP PARITY

                BIT9:
02A0 BF0704             CJNE R7,#07,BIT10
02A3 D2E7               SETB ACC.7              ;SET THE PARITY BIT
02A5 41A9               AJMP PARITY

                BIT10:
02A7 C2E7               CLR ACC.7               ;CLEAR THE PARITY BIT


02A9 B40003     PARITY: CJNE A,#0,SEND      ;CHECK IF THIS IS THE END OF
                                            ;THE MESSAGE
02AC D0E0               POP ACC
02AE 22                 RET

                SEND:
02AF 3136              ACALL XMTBYTE        ;START TO CLOCK OUT THE BYTE
                                            ;SERIALLY
02B1 08                 INC R0              ;MOVE POINTER TO NEXT BYTE OF
```

```
The Cybernetic Micro Systems 8051 Family Assembler, Version 3.04
                                                                    25-11-92

                                                                ;MESSAGE
02B2 80BA                       SJMP MES1


            ;**********************THE MESSAGES*********************************
            ;
            ;
            ;############# 0A = LF
            ;############# 8D = CR
            ;############# 0C = FF


                MSG2:
02B4 8D 0C                      DB 08DH, 0CH
02B6 45 4E 54                   DB 'ENTER SPEED'
02B9 45 52 20 53 50 45 45 44
02C1 8D 0A 0A                   DB 8DH, 0AH, 0AH
02C4 20 39 20                   DB ' 9 = 9600Bps'
02C7 3D 20 39 36 30 30 42 70 73
02D0 8D 0A                      DB 8DH, 0AH
02D2 20 34 20                   DB ' 4 = 4800Bps'
02D5 3D 20 34 38 30 30 42 70 73
02DE 8D 0A                      DB 8DH, 0AH
02E0 20 32 20                   DB ' 2 = 2400Bps'
02E3 3D 20 32 34 30 30 42 70 73
02EC 8D 0A                      DB 8DH, 0AH
02EE 20 31 20                   DB ' 1 = 1200Bps'
02F1 3D 20 31 32 30 30 42 70 73
02FA 8D 0A                      DB 8DH, 0AH
02FC 20 33 20                   DB ' 3 = 300Bps'
02FF 3D 20 33 30 30 42 70 73
0307 8D 0A                      DB 8DH, 0AH
0309 45 4E 54                   DB 'ENTER CHOICE : ',0H
030C 45 52 20 43 48 4F 49 43 45 20
0316 3A 20 00


                MSG3:
0319 8D 0C                      DB 8DH,0CH
031B 53 55 50                   DB 'SUPPRESS FORMFEEDS? (Y/N)',0H
031E 50 52 45 53 53 20 46 4F 52 4D
0328 46 45 45 44 53 3F 20 28 59 2F
0332 4E 29 00


                MSG1:
0335 8D 0C                      DB 8DH, 0CH
0337 43 4F 4E                   DB 'CONFIGURATION OF THE INTERFACE MODULE'
033A 46 49 47 55 52 41 54 49 4F 4E
0344 20 4F 46 20 54 48 45 20 49 4E
034E 54 45 52 46 41 43 45 20 4D 4F
0358 44 55 4C 45
035C 8D 0A 0A                   DB 8DH, 0AH, 0AH, 0AH, 0AH, 0AH, 0AH, 0AH, 0AH
035F 0A 0A 0A 0A 0A 0A
0365 43 55 52                   DB 'CURRENT CONFIGURATION : SPEED=',0H
```

```
0368 52 45 4E 54 20 43 4F 4E 46 49
0372 47 55 52 41 54 49 4F 4E 20 3A
037C 20 53 50 45 45 44 3D 00


                MSG101:
0384 31 32 30                   DB '1200 Bps',0H
0387 30 20 42 70 73 00


                MSG102:

038D 32 34 30                   DB '2400 Bps',0H
0390 30 20 42 70 73 00


                MSG103:
0396 33 30 30                   DB '300 Bps',0H
0399 20 42 70 73 00


                MSG104:
039E 34 38 30                   DB '4800 Bps',0H
03A1 30 20 42 70 73 00


                MSG105:
03A7 39 36 30                   DB '9600 Bps',0H
03AA 30 20 42 70 73 00


                MSG106:
03B0 8D 0A                      DB 8DH,0AH
03B2 20 20 20                   DB '                    FF SUPPRESS ON',0H
03B5 20 20 20 20 20 20 20 20 20 20
03BF 20 20 20 20 20 20 20 20 20 20
03C9 20 46 46 20 53 55 50 50 52 45
03D3 53 53 20 4F 4E 00



                MSG107:
03D9 8D 0A                      DB 8DH, 0AH
03DB 20 20 20                   DB '                    FF SUPPRESS OFF',0H
03DE 20 20 20 20 20 20 20 20 20 20
03E8 20 20 20 20 20 20 20 20 20 20
03F2 20 46 46 20 53 55 50 50 52 45
03FC 53 53 20 4F 46 46 00


                MSG108:
0403 8D 0A                      DB 8DH, 0AH
0405 20 20 20                   DB '                    PARALLEL PORT'
0408 20 20 20 20 20 20 20 20 20 20
0412 20 20 20 20 20 20 20 20 20 20
```

```
041C 20 50 41 52 41 4C 4C 45 4C 20
0426 50 4F 52 54
042A 8D 0A 0A                   DB 8DH, 0AH, 0AH, 0AH, 0AH, 0AH, 0AH, 0AH
042D 0A 0A 0A 0A 0A
0432 49 53 20                   DB 'IS THE CONFIGURATION CORRECT? (Y/N)',0
0435 54 48 45 20 43 4F 4E 46 49 47
043F 55 52 41 54 49 4F 4E 20 43 4F
0449 52 52 45 43 54 3F 20 28 59 2F
0453 4E 29 00


                MSG109:
0456 8D 0A                      DB 8DH, 0AH
0458 20 20 20                   DB '                                SERIAL PORT'
045B 20 20 20 20 20 20 20 20 20 20
0465 20 20 20 20 20 20 20 20 20 20
046F 20 53 45 52 49 41 4C 20 50 4F
0479 52 54


                MSG4:
047B 8D 0A 0A                   DB 8DH,, 0AH, 0AH, 0AH, 0AH, 0AH, 0AH, 0AH
047E 0A 0A 0A 0A 0A
0483 49 53 20                   DB 'IS THE CONFIGURATION CORRECT? (Y/N)',0
0486 54 48 45 20 43 4F 4E 46 49 47
0490 55 52 41 54 49 4F 4E 20 43 4F
049A 52 52 45 43 54 3F 20 28 59 2F
04A4 4E 29 00


                MSG5:
04A7 8D 0C                      DB 8DH, 0CH
04A9 53 45 4C                   DB 'SELECT PARALLEL OR SERIAL OPERATION'
04AC 45 43 54 20 50 41 52 41 4C 4C
04B6 45 4C 20 4F 52 20 53 45 52 49
04C0 41 4C 20 4F 50 45 52 41 54 49
04CA 4F 4E
04CC 8D 0A                      DB 8DH, 0AH
04CE 50 20 3D                   DB 'P = PARALLEL'
04D1 20 50 41 52 41 4C 4C 45 4C
04DA 8D 0A                      DB 8DH, 0AH
04DC 53 20 3D                   DB 'S = SERIAL'
04DF 20 53 45 52 49 41 4C
04E6 8D 0A                      DB 8DH,0AH
04E8 4D 41 4B                   DB 'MAKE SELECTION (P/S)',0H
04EB 45 20 53 45 4C 45 43 54 49 4F
04F5 4E 20 28 50 2F 53 29 00


                MSG6:
04FD 8D 0C                      DB 8DH, 0CH
04FF 43 4F 4E                   DB 'CONFIGURATION OF THE INTERFACE MODULE'
0502 46 49 47 55 52 41 54 49 4F 4E
050C 20 4F 46 20 54 48 45 20 49 4E
```

```
0516 54 45 52 46 41 43 45 20 4D 4F
0520 44 55 4C 45
0524 8D 0A                      DB 8DH, 0AH
0526 49 53 20                   DB 'IS NOW COMPLETE.'
0529 4E 4F 57 20 43 4F 4D 50 4C 45
0533 54 45 2E
0536 8D 0A 0A                   DB 8DH, 0AH, 0AH
0539 52 45 46                   DB 'REFER TO THE "MINITEL USERS GUIDE", IF'
053C 45 52 20 54 4F 20 54 48 45 20
0546 22 4D 49 4E 49 54 45 4C 20 55
0550 53 45 52 53 20 47 55 49 44 45
055A 22 2C 20 49 46
055F 8D 0A                      DB 8DH, 0AH
0561 4E 45 43                   DB 'NECESSARY, AND CONFIGURE THE PERIPHERAL
0564 45 53 53 41 52 59 2C 20 41 4E
056E 44 20 43 4F 4E 46 49 47 55 52
0578 45 20 54 48 45 20 50 45 52 49
0582 50 48 45 52 41 4C
0588 50 4F 52                       DB 'PORT OF THE MINITEL TERMINAL. (SEE P.28)',0H
058B 54 20 4F 46 20 54 48 45 20 4D
0595 49 4E 49 54 45 4C 20 54 45 52
059F 4D 49 4E 41 4C 2E 20 28 53 45
05A9 45 20 50 2E 32 38 29 00
```

```
                RRESET:
05B1 20790E             JB PAR,BEGIN            ;PARALLEL OR SERIAL OPERATION
05B4 75A800             MOV IE,#00H             ;DISABLE INTERRUPTS

                SERIAL:
05B7 308004             JNB P0.0,SEREAL         ;COPY P0.0 TO P0.1
05BA D281               SETB P0.1
05BC 80F9               SJMP SERIAL

                SEREAL:
05BE C281               CLR P0.1
05C0 80F5               SJMP SERIAL

                BEGIN:                          ;DEFINE BITS AND REGISTERS
0077 =                  RRXFLAG BIT 77H
0076 =                  RRXERR  BIT 76H
0075 =                  RRCVRDY BIT 75H
0074 =                  XOF BIT 74H
0073 =                  XON BIT 73H
0072 =                  XF BIT 72H
0071 =                  XN BIT 71H
```

```
0070 =                  XNO BIT 70H
05C2 758135             MOV SP,#35H
05C5 75A883             MOV IE,#083H
05C8 758800             MOV TCON,#00H
05CB C277               CLR RRXFLAG
05CD C276               CLR RRXERR
05CF C275               CLR RRCVRDY
05D1 C273               CLR XON                 ;FLOW CONTROL FLAGS
05D3 C274               CLR XOF
05D5 C272               CLR XF
05D7 C271               CLR XN
05D9 C270               CLR XNO
05DB 75D000             MOV PSW,#00H
05DE 7590FF             MOV P1,#0FFH            ;INITIALISE PORTS 1 AND 3
05E1 75B0FF             MOV P3,#0FFH
05E4 D281               SETB P0.1               ;MINITEL RXD
05E6 7803               MOV R0,#03H             ;INTERNAL RAM POINTERS
05E8 7903               MOV R1,#03H
05EA 307C10             JNB NIN,WAIT
05ED 7AFE               MOV R2,#0FEH            ;SETUP TIMER REGS. FOR 9600BPS
05EF 8A8A               MOV 8AH,R2
05F1 7AFF               MOV R2,#0FFH
05F3 8A8C               MOV 8CH,R2
05F5 7AFF               MOV R2,#0FFH
05F7 8A8D               MOV 8DH,R2
05F9 7AA3               MOV R2,#0A3H
05FB 8A8B               MOV 8BH,R2

                WAIT:
05FD 208CFD             JB 8CH,WAIT         ;IS TIMER RUNNING ?
0600 208BFA             JB 8BH,WAIT         ;IS EXTERNAL INTERRUPT SERVICED ?
0603 D113               ACALL LOAD
0605 307309             JNB XON,BITE
0608 D271               SETB XN               ;INDICATE THAT XON IS XMITTE
060A D28C               SETB 8CH              ;START TIMER
060C 75310A             MOV CNT,#0AH
060F A1FD               AJMP WAIT

                BITE:
0611 A1FD               AJMP WAIT

                LOAD:
0613 20962A             JB P1.6,EXLOAD            ;EXIT IF PRINTER IS BUSY
0616 08                 INC R0
0617 E9                 MOV A,R1
0618 98                 SUBB A,R0                 ;IS THERE BUFFERED DATA
0619 5012               JNC LOAD2
061B C3                 CLR C
061C C2A8               CLR EX0
061E 7803               MOV R0,#03h
0620 7903               MOV R1,#03H
0622 D2A8               SETB EX0
0624 307219             JNB XF,EXLOAD           ;HAS MINITEL RECEIVED XOFF
0627 D273               SETB XON                ;INDICATE THAT BUFFER IS CLEAR
0629 C272               CLR XF
```

```
06A7 8A8C               MOV 8CH,R2
06A9 7AFD               MOV R2,#0FDH
06AB 8A8D               MOV 8DH,R2
06AD 7A00               MOV R2,#00H
06AF 8A8B               MOV 8BH,R2
06B1 C1C3               AJMP SSPEED

                TTHREE:                     ;IS SPEED 300BPS ?
06B3 7A23               MOV R2,#23H
06B5 8A8A               MOV 8AH,R2
06B7 7AFA               MOV R2,#0FAH
06B9 8A8C               MOV 8CH,R2
06BB 7AF4               MOV R2,#0F4H
06BD 8A8D               MOV 8DH,R2
06BF 7A00               MOV R2,#00H
06C1 8A8B               MOV 8BH,R2

                SSPEED:
06C3 753000             MOV XMTDAT,#0H
06C6 C2A8               CLR EX0         ;DISABLE EXTERNAL INTERRUPTS
06C8 D28C               SETB 8CH        ;START TIMER
06CA D277               SETB RRXFLAG    ;INDICATE RECEPTION IS IN PROGRESS
06CC 32                 RETI

                TTIMR0:
06CD F532               MOV LOOPCNT,A   ;STORE CONTENTS OF A IN LOOPCNT
06CF 927F               MOV THR,C       ;STORE CONTENTS OF CARRY IN THR
06D1 A295               MOV C,P1.5      ;STORE NEXT BIT FROM MINITEL IN PAR
                                        ;VIA CARRY FLAG.
06D3 9279               MOV PAR,C
06D5 207709             JB RRXFLAG,RRXBI

                TT0EX1:
06D8 C28C               CLR 8CH         ;STOP TIMER

                TT0EX2:
06DA C270               CLR XNO         ;INDICATE THAT XON HAS BEEN TRANSMITTED
06DC A27F               MOV C,THR       ;RESTORE CARRY FLAG
06DE E532               MOV A,LOOPCNT   ;RESTORE ACCUMULATOR
06E0 32                 RETI

                RRXBIT:
06E1 D5311E             DJNZ CNT,RRXBUSY
06E4 208106             JB P0.1,EXIT9
06E7 D281               SETB P0.1
06E9 C271               CLR XN
06EB D270               SETB XNO

                EXIT9:
06ED 2070E8             JB XNO,TT0EX1
06F0 C277               CLR RRXFLAG
06F2 C275               CLR RRCVRDY
06F4 09                 INC R1
06F5 E530               MOV A,XMTDAT
06F7 F7                 MOV @R1,A
```

```
6F8 E9                  MOV A,R1
6F9 B42002              CJNE A,#20H,XLOAD   ;IS INTERNAL RAM FULL ?
6FC D274                SETB XOF            ;INDICATE THAT XOFF MUST BE SENT

                XLOAD:
6FE D2A8                SETB EX0
700 80D6                SJMP TT0EX1

                RRXBUSY:
702 E531                MOV A,CNT
704 B4090A              CJNE A,#9,RRXNEXT
707 207403              JB XOF,OK5
70A 307102              JNB XN,EXIT8

                OK5:
70D C281                CLR P0.1

                EXIT8:
70F 80C9                SJMP TT0EX2

                RRXNEXT:
711 E530                MOV A,XMTDAT
713 A279                MOV C,PAR
715 13                  RRC A
716 F530                MOV XMTDAT,A
718 307102              JNB XN,OK6
71B F146                ACALL TIMR2

                OK6:
71D 307400              JNB XOF,EXIT

                EXIT7:
720 80B8                SJMP TT0EX2

                TIMR1:
722 E531            MOV A,CNT
724 B40803          CJNE A,#8,LLL
727 D281            SETB P0.1
729 22              RET

                LLL:
72A B40603          CJNE A,#6,QQQ
72D C281            CLR P0.1
72F 22              RET

                QQQ:
730 B40403          CJNE A,#4,RRR
733 D281            SETB P0.1
735 22              RET

                RRR:
736 B40303          CJNE A,#3,SSS
739 C281            CLR P0.1
73B 22              RET
```

```
                SSS:
073C B40106             CJNE A,#1,EXIT
073F D281               SETB P0.1
0741 D272               SETB XF
0743 C274               CLR XOF

                EXIT:
0745 22                 RET

                TIMR2:
0746 E531               MOV A,CNT
0748 B40804             CJNE A,#8,LLLL
074B D281               SETB P0.1
074D E169               AJMP EXIT6

                LLLL:
074F B40704             CJNE A,#7,PPPP
0752 C281               CLR P0.1
0754 E169               AJMP EXIT6

                PPPP:
0756 B40604             CJNE A,#6,QQQQ
0759 C281               CLR P0.1
075B E169               AJMP EXIT6

                QQQQ:
075D B40404             CJNE A,#4,RRRR
0760 D281               SETB P0.1
0762 E169               AJMP EXIT
                RRRR:
0764 B40302             CJNE A,#3,EXIT6
0767 C281               CLR P0.1

                EXIT6:
0769 22                 RET

                ;%E
0000            END
```

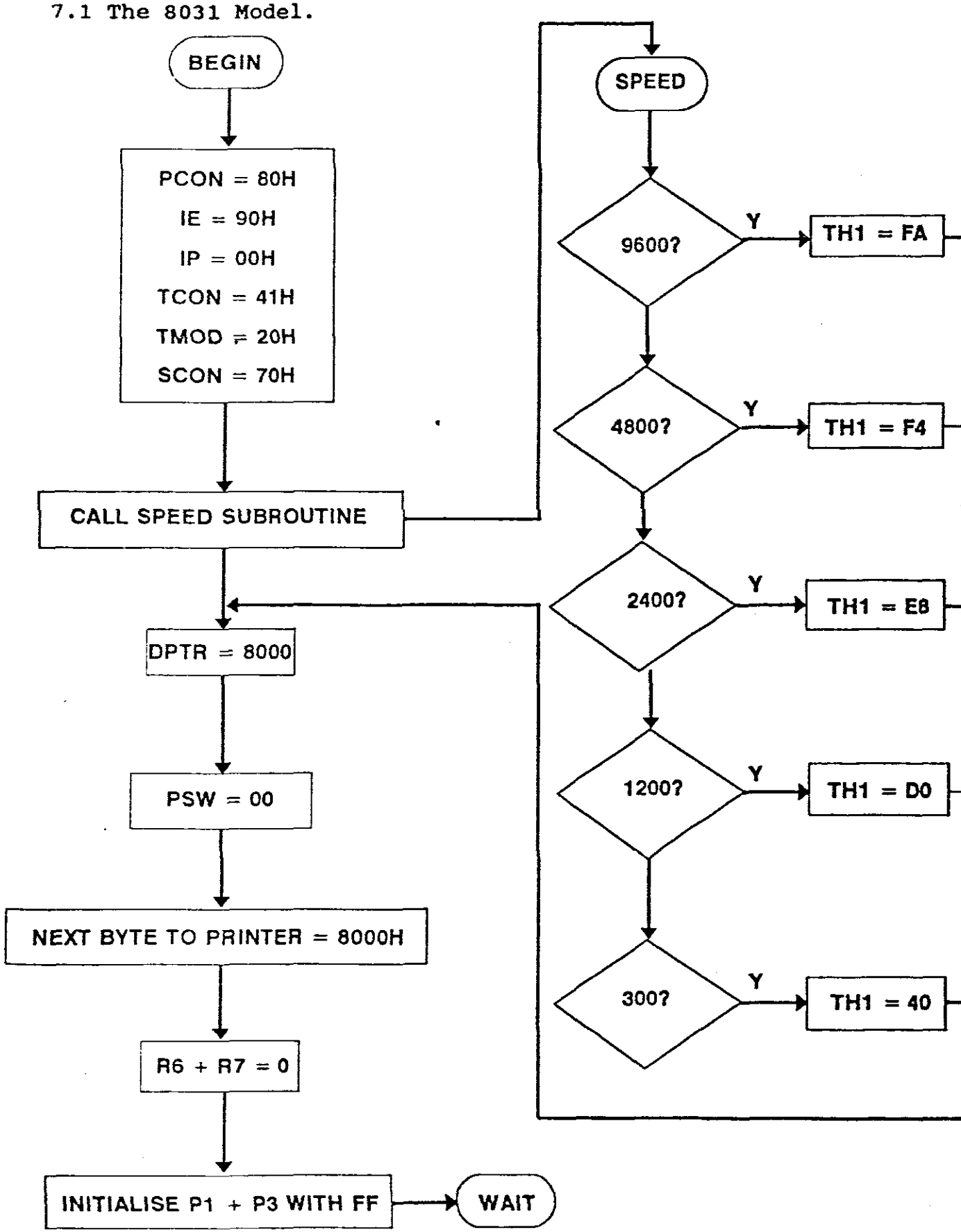
```
;%T   Symbol Name                Type Value

BAUDVAL . . . . . . . . . . . . . I FD00
BEGIN . . . . . . . . . . . . . . L 05C2
BIT0. . . . . . . . . . . . . . . L 0276
BIT1. . . . . . . . . . . . . . . L 027A
BIT10 . . . . . . . . . . . . . . L 02A7
BIT2. . . . . . . . . . . . . . . L 027B
BIT3. . . . . . . . . . . . . . . L 027F
BIT4. . . . . . . . . . . . . . . L 0283
BIT5. . . . . . . . . . . . . . . L 0287
BIT6. . . . . . . . . . . . . . . L 028B
BIT7. . . . . . . . . . . . . . . L 0292
BIT8. . . . . . . . . . . . . . . L 0299
BIT9. . . . . . . . . . . . . . . L 02A0
BITCNT. . . . . . . . . . . . . . D 0033
BITE. . . . . . . . . . . . . . . L 0611
CNT . . . . . . . . . . . . . . . D 0031
DEL . . . . . . . . . . . . . . . B 0069
DELAY . . . . . . . . . . . . . . L 0062
DELAY1. . . . . . . . . . . . . . L 0078
DRIE. . . . . . . . . . . . . . . L 0229
EEN . . . . . . . . . . . . . . . L 0216
EEXINT0 . . . . . . . . . . . . . L 0670
EXINT0. . . . . . . . . . . . . . L 0183
EXIT. . . . . . . . . . . . . . . L 0745
EXIT6 . . . . . . . . . . . . . . L 0769
EXIT7 . . . . . . . . . . . . . . L 0720
EXIT8 . . . . . . . . . . . . . . L 070F
EXIT9 . . . . . . . . . . . . . . L 06ED
EXLOAD. . . . . . . . . . . . . . L 0640
EXLOAD1 . . . . . . . . . . . . . L 0636
F. . . . . . . . . . . . . . . . . L 023C
FON. . . . . . . . . . . . . . . . B 007B
FOUR . . . . . . . . . . . . . . . L 0676
FIN . . . . . . . . . . . . . . . . B 0072
FIN1. . . . . . . . . . . . . . . . B 006A
FINISH. . . . . . . . . . . . . . . L 01C7
FORM. . . . . . . . . . . . . . . . B 006F
FORMF . . . . . . . . . . . . . . . B 006C
FOU . . . . . . . . . . . . . . . . B 007D
FOUR. . . . . . . . . . . . . . . . L 01F2
GO1. . . . . . . . . . . . . . . . L 0080
GO. . . . . . . . . . . . . . . . . B 0073
HEN . . . . . . . . . . . . . . . . L 012F
ILL . . . . . . . . . . . . . . . . L 072A
ILLL. . . . . . . . . . . . . . . . L 074F
LOAD. . . . . . . . . . . . . . . . L 0613
LOAD2 . . . . . . . . . . . . . . . L 062D
LOGO. . . . . . . . . . . . . . . . L 00B3
LOGO01. . . . . . . . . . . . . . . L 00C8
LOGO02. . . . . . . . . . . . . . . L 00CF
LOGO03. . . . . . . . . . . . . . . L 00D6
LOGO04. . . . . . . . . . . . . . . L 00DD
```

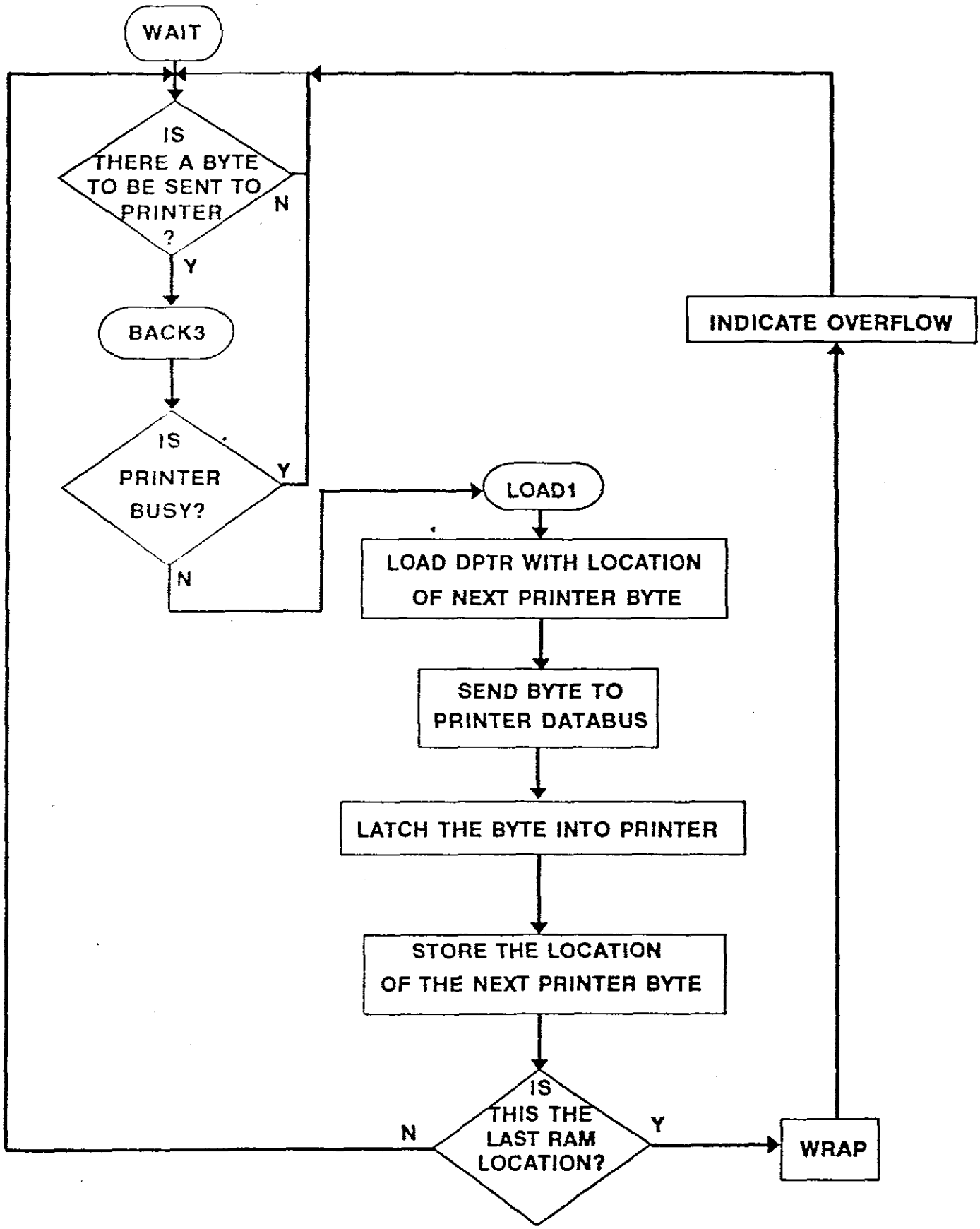
```
.OGO05. . . . . L 00E4
.OGO1 . . . . . L 00E9
.OGO101 . . . . . L 00F5
.OGO2 . . . . . L 00FE
.OGO201 . . . . . L 010A
.OOGOO. . . . . L 0134
.OOP1 . . . . . L 007B
.OOP2 . . . . . L 012A
.OOPCNT . . . . . D 0032
IES1. . . . . L 026E
IES2. . . . . L 008F
IES4. . . . . L 0097
IES5. . . . . L 009E
MES6. . . . . L 00A6
MES7. . . . . L 00AE
MESS. . . . . L 026A
MSG1. . . . . L 0335
MSG101. . . . . L 0384
MSG102. . . . . L 038D
MSG103. . . . . L 0396
MSG104. . . . . L Q39E
MSG105. . . . . L 03A7
MSG106. . . . . L 03B0
MSG107. . . . . L 03D9
MSG108. . . . . L 0403
MSG109. . . . . L 0456
MSG2. . . . . L 02B4
MSG3. . . . . L 0319
MSG4. . . . . L 047B
MSG5. . . . . L 04A7
MSG6. . . . . L 04FD
MIN . . . . . B 007C
OK5 . . . . . L 070D
OK6 . . . . . L 071D
ONE . . . . . B 007E
ONE. . . . . L 069E
OUT . . . . . L 0266
OUT1. . . . . L 01D2
OUT3. . . . . L 0247
OUT4. . . . . L 025D
OUT5. . . . . L 01DB
PAR . . . . . B 0079
PARITY. . . . . L 02A9
PORT. . . . . L 0252
PORT1 . . . . . B 006B
PORTY . . . . . L 01D0
PPPP. . . . . L 0756
PPS . . . . . B 0070
PRT . . . . . L 0111
PRT1. . . . . L 0120
PRT2. . . . . L 0123
QQQ . . . . . L 0730
QQQQ. . . . . L 075D
RAM . . . . . B 006D
RAM1. . . . . B 007A
```

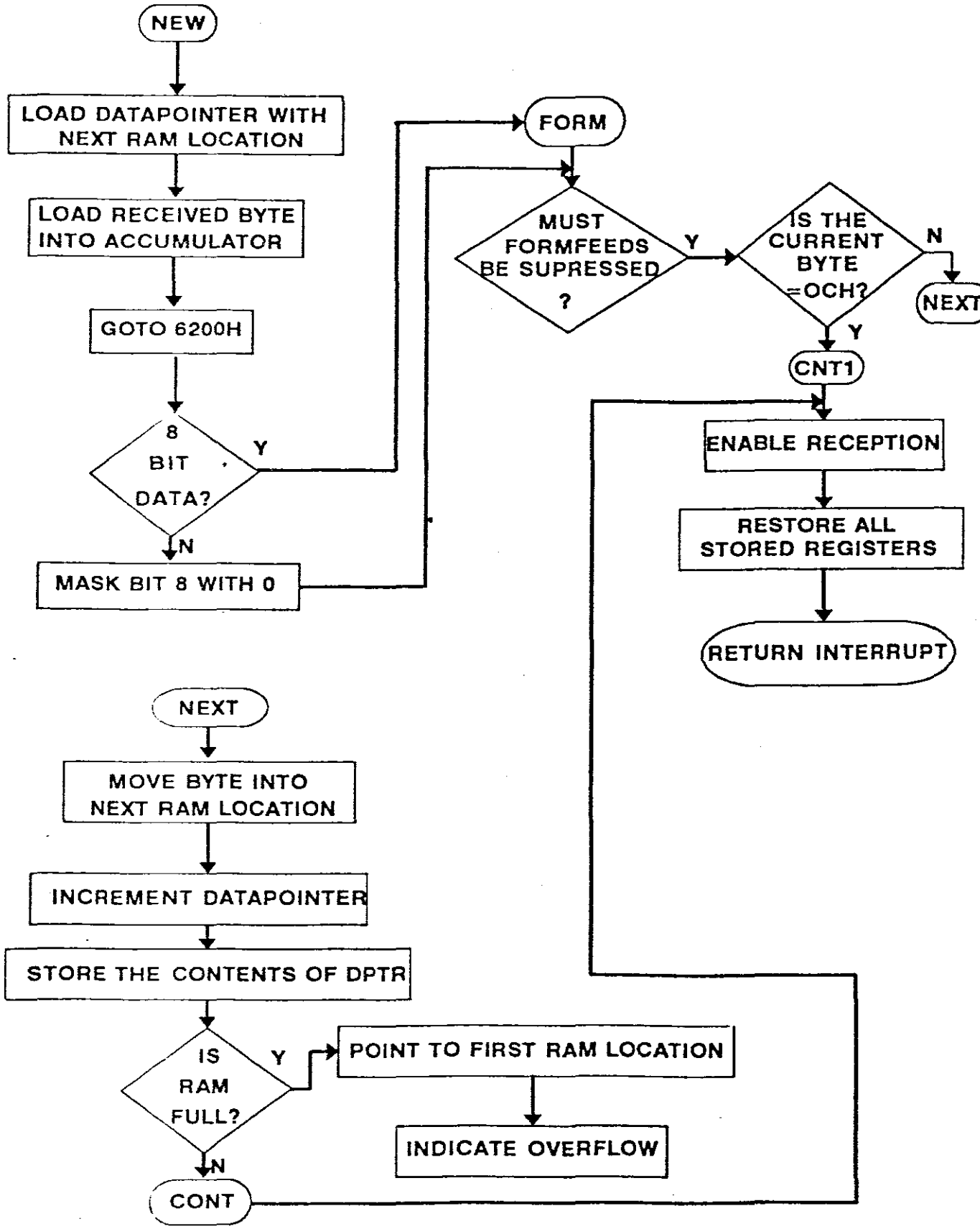
```
RCVDAT. . . . . D 0034
RCVRDY. . . . . B 0074
REP . . . . . B 006E
RRCVRDY . . . . . B 0075
RRESET. . . . . L 05B1
RRR . . . . . L 0736
RRRR. . . . . L 0764
RRXBIT. . . . . L 06E1
RRXBUSY . . . . . L 0702
RRXERR. . . . . B 0076
RRXFLAG . . . . . B 0077
RRXNEXT . . . . . L 0711
RSXMT . . . . . L 013F
RXBIT . . . . . L 01A2
RXBITEX . . . . . L 01B7
RXBUSY. . . . . L 01A7
RXERR . . . . . B 0075
RXFLAG. . . . . B 0076
RXNEXT. . . . . L 01AE
SEND. . . . . L 02AF
SEREAL. . . . . L Q5BE
SERIAL. . . . . L 05B7
SET1. . . . . L 0008
SET2. . . . . L 0013
SPEED . . . . . L 01DF
SSPEED. . . . . L 06C3
SSS . . . . . L 073C
STRTVAL . . . . . I FE93
T0EX1 . . . . . L 0163
T0EX2 . . . . . L 0165
THR . . . . . B 007F
TIMR0 . . . . . L 0159
TIMR1 . . . . . L 0722
TIMR2 . . . . . L 0746
TSET. . . . . B 0071
TT0EX1. . . . . L 06D8
TT0EX2. . . . . L 06DA
TTHREE. . . . . L 06B3
TTIMR0. . . . . L 06CD
TTWO. . . . . L 0689
TWEE. . . . . L 0205
TWO . . . . . B 0078
TXBIT . . . . . L 016A
TXBUSY. . . . . L 0171
TXFLAG. . . . . B 0077
TXNEXT. . . . . L 017A
TXX . . . . . B 0081
WAIT. . . . . L 05FD
XF. . . . . B 0072
XLOAD . . . . . L 06FE
XMTBYTE . . . . . L 0136
XMTDAT. . . . . D 0030
XN. . . . . B 0071
XNO . . . . . B 0070
XOF . . . . . B 0074
```

7. The Flowcharts.

7.1 The 8031 Model.

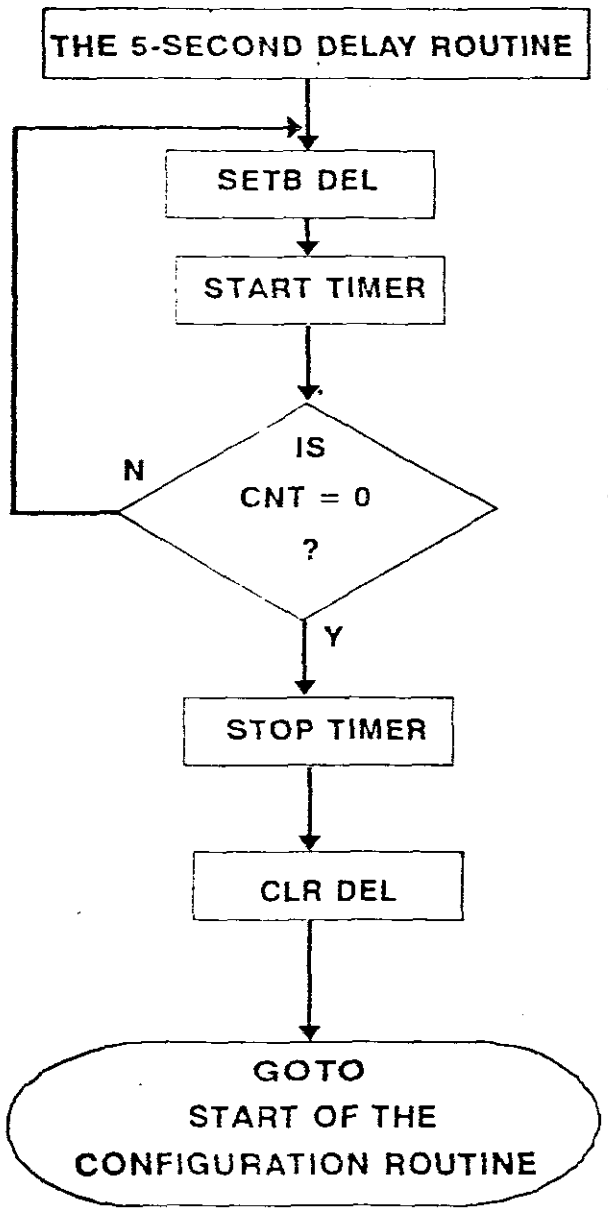


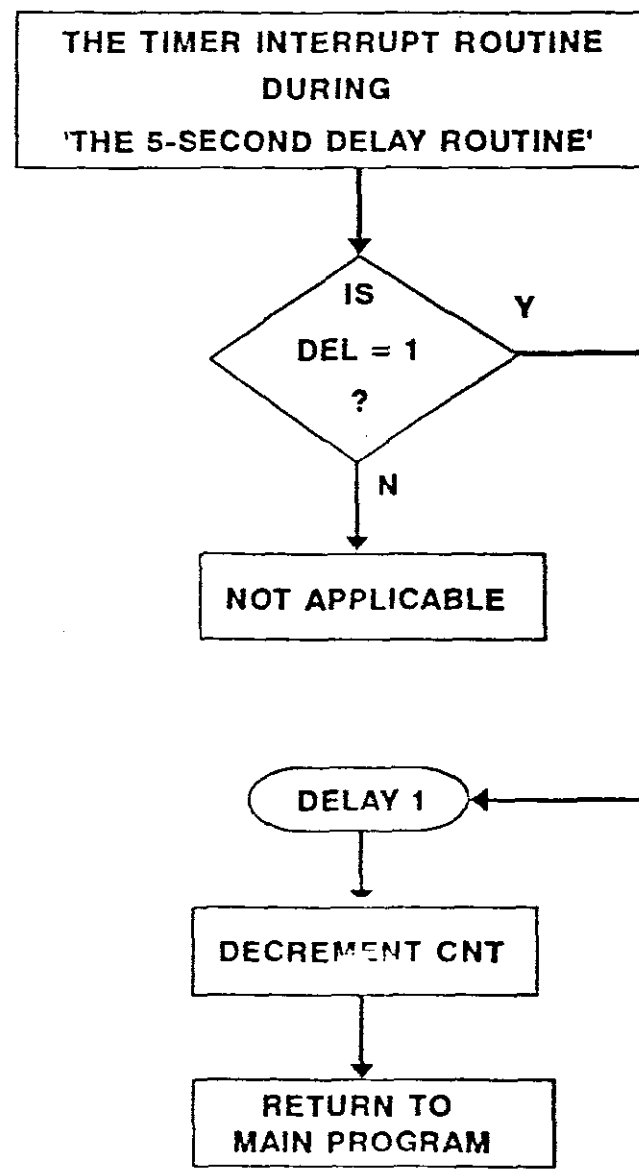

WAIT
IS THERE A BYTE TO BE SENT TO PRINTER ?
N
Y
BACK3
IS PRINTER BUSY?
Y
N
LOAD1
LOAD DPTR WITH LOCATION OF NEXT PRINTER BYTE
SEND BYTE TO PRINTER DATABUS
LATCH THE BYTE INTO PRINTER
STORE THE LOCATION OF THE NEXT PRINTER BYTE
IS THIS THE LAST RAM LOCATION?
N
Y
WRAP
INDICATE OVERFLOW

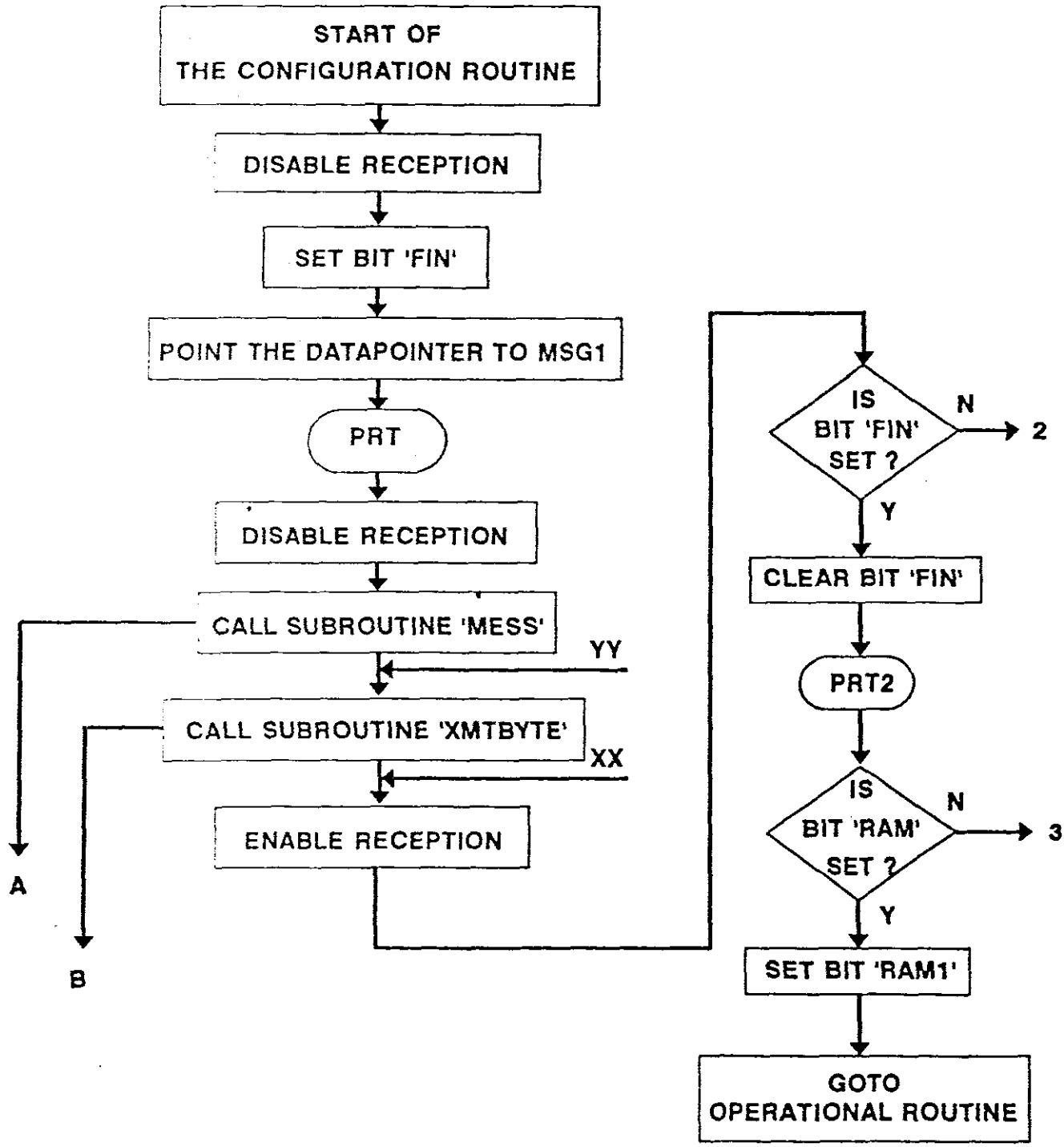

NEW
LOAD DATAPOINTER WITH NEXT RAM LOCATION
LOAD RECEIVED BYTE INTO ACCUMULATOR
GOTO 6200H
8 BIT DATA?
Y
N
MASK BIT 8 WITH 0
FORM
MUST FORMFEEDS BE SUPRESSED ?
Y
IS THE CURRENT BYTE =OCH?
N
NEXT
Y
CNT1
ENABLE RECEPTION
RESTORE ALL STORED REGISTERS
RETURN INTERRUPT
NEXT
MOVE BYTE INTO NEXT RAM LOCATION
INCREMENT DATAPOINTER
STORE THE CONTENTS OF DPTR
IS RAM FULL?
Y
N
POINT TO FIRST RAM LOCATION
INDICATE OVERFLOW
CONT

7.2 The 87C751 Model.

7.2.1 The Configuration Routine.

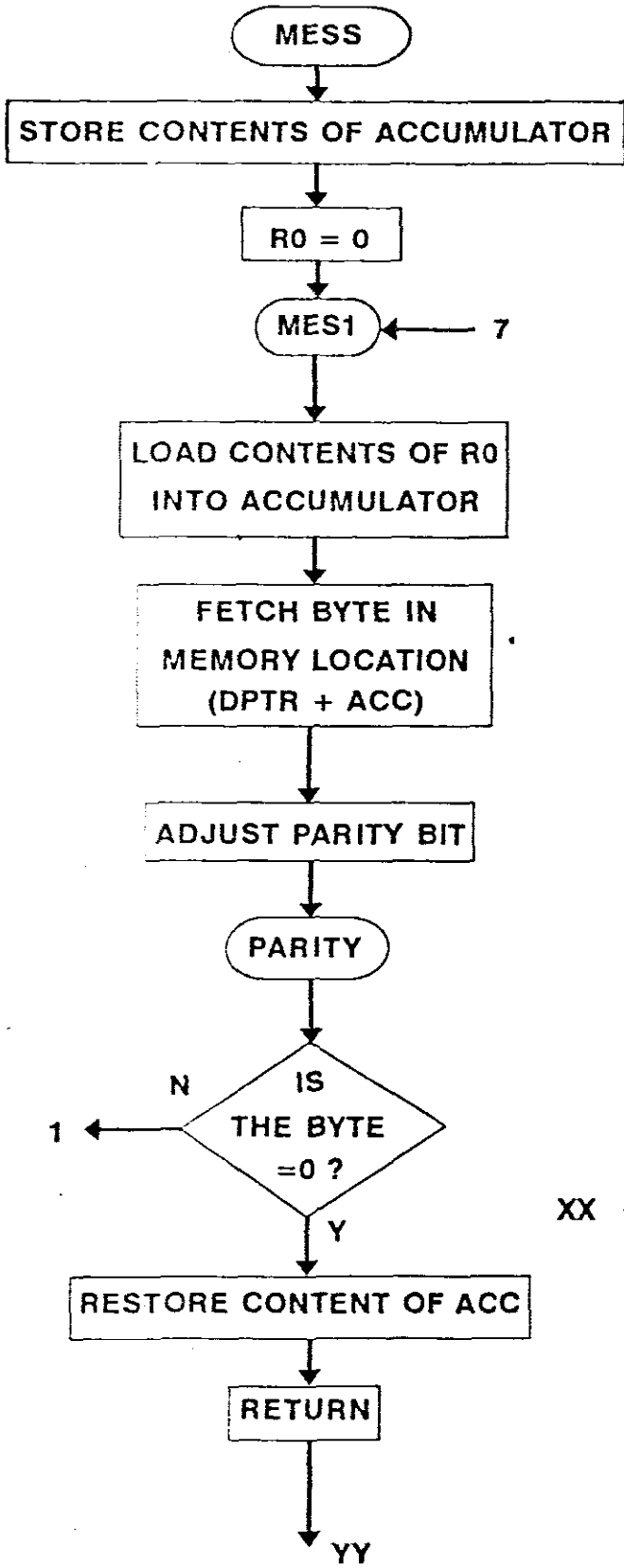


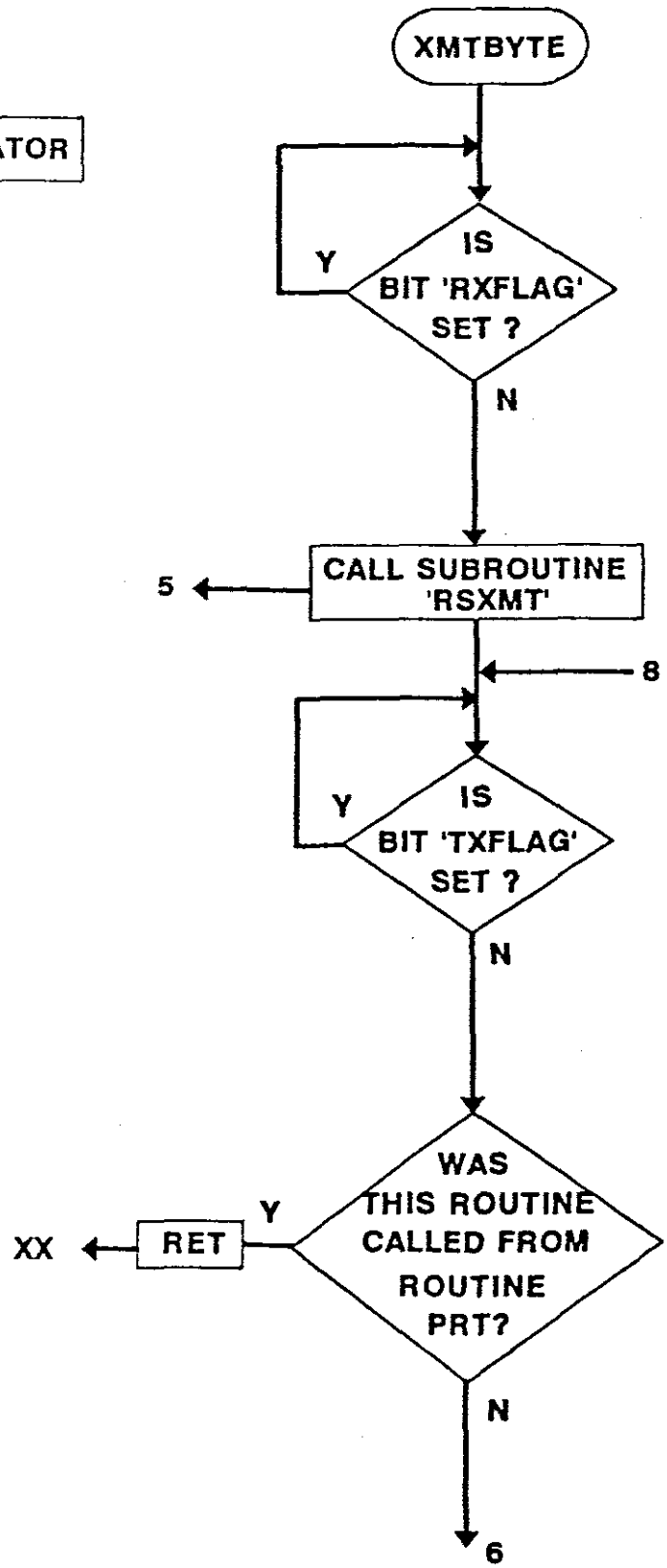


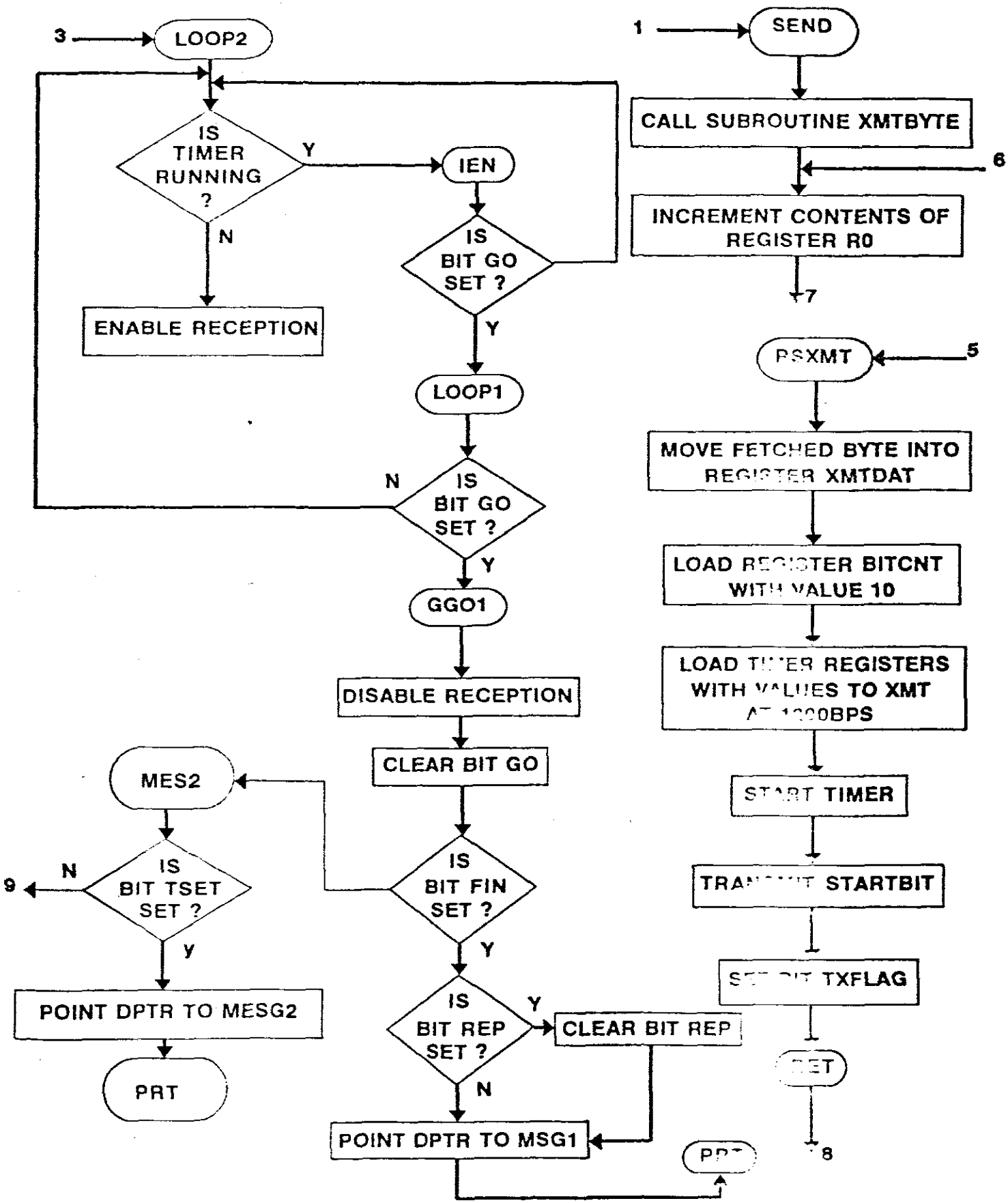


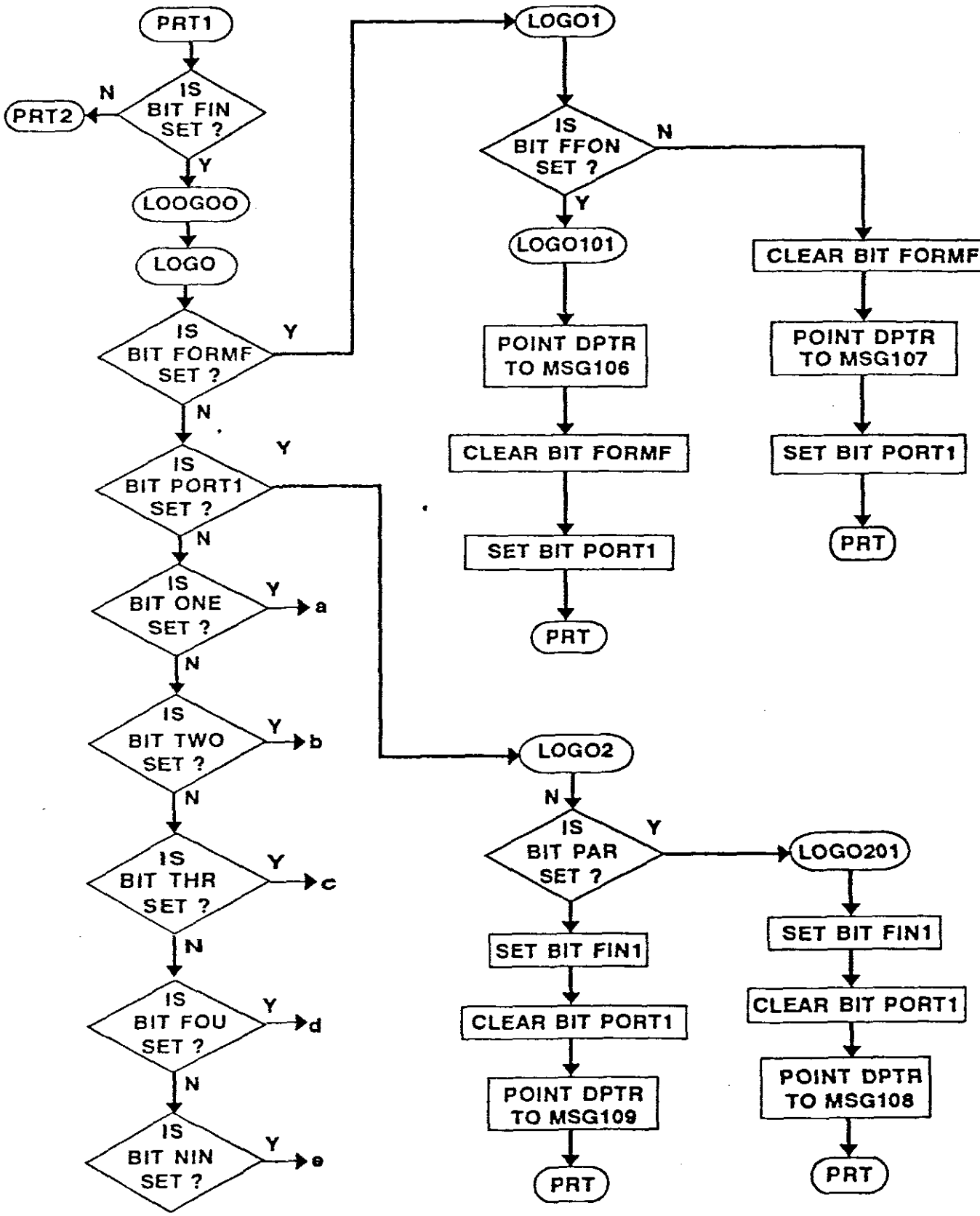
A = SUBROUTINE 'MESS'

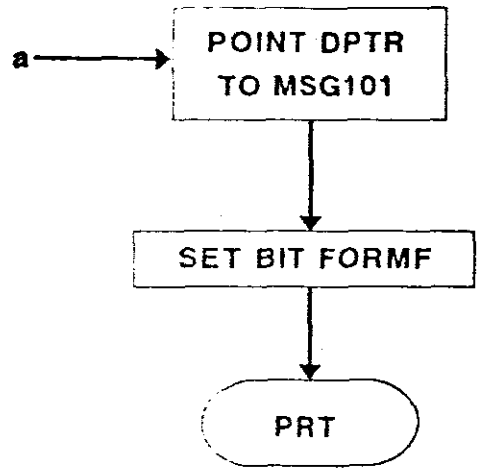
B = SUBROUTINE 'XMTBYTE'

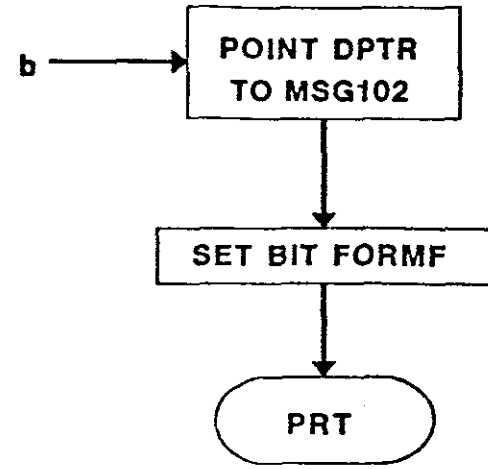

MESS
STORE CONTENTS OF ACCUMULATOR
R0 = 0
MES1
7
LOAD CONTENTS OF R0 INTO ACCUMULATOR
FETCH BYTE IN MEMORY LOCATION (DPTR + ACC)
ADJUST PARITY BIT
PARITY
N
IS THE BYTE =0 ?
1
Y
RESTORE CONTENT OF ACC
RETURN
YY

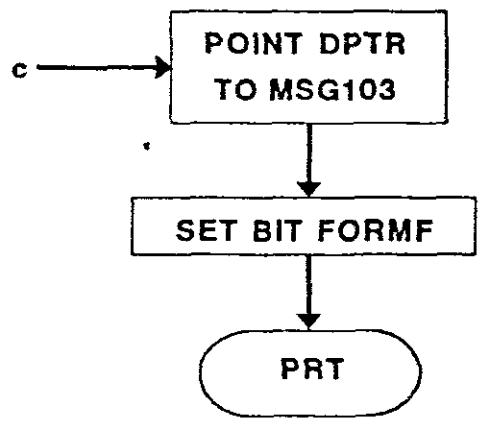

XMTBYTE
Y
IS BIT 'RXFLAG' SET ?
N
5
CALL SUBROUTINE 'RSXMT'
8
Y
IS BIT 'TXFLAG' SET ?
N
WAS THIS ROUTINE CALLED FROM ROUTINE PRT?
Y
XX
RET
N
6

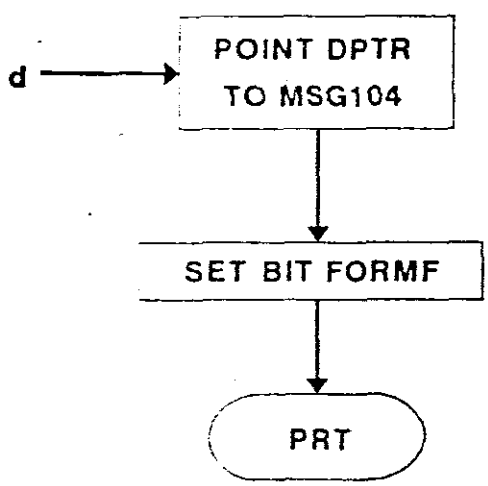

3
LOOP2
IS TIMER RUNNING ?
Y
N
IEN
IS BIT GO SET ?
Y
ENABLE RECEPTION
LOOP1
N
IS BIT GO SET ?
Y
GGO1
DISABLE RECEPTION
CLEAR BIT GO
IS BIT FIN SET ?
Y
IS BIT REP SET ?
Y
CLEAR BIT REP
N
POINT DPTR TO MSG1
MES2
N
IS BIT TSET SET ?
9
y
POINT DPTR TO MESG2
PRT
1
SEND
CALL SUBROUTINE XMTBYTE
6
INCREMENT CONTENTS OF REGISTER R0
7
PSXMT
5
MOVE FETCHED BYTE INTO REGISTER XMTDAT
LOAD REGISTER BITCNT WITH VALUE 10
LOAD TIMER REGISTERS WITH VALUES TO XMT
START TIMER
TRANSMIT STARTBIT
SET BIT TXFLAG
RET
8

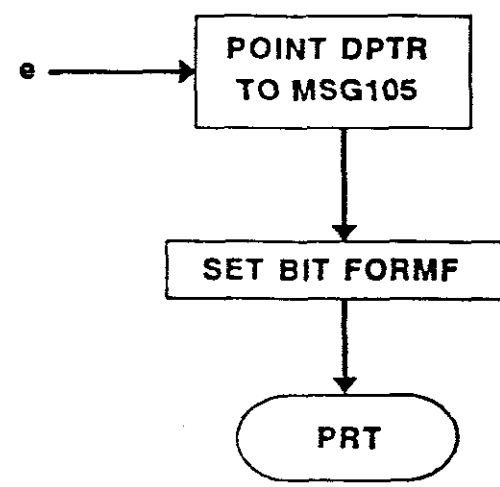

PRT1
IS BIT FIN SET ?
N
PRT2
Y
LOOGOO
LOGO
IS BIT FORMF SET ?
Y
N
IS BIT PORT1 SET ?
Y
N
IS BIT ONE SET ?
Y
a
N
IS BIT TWO SET ?
Y
b
N
IS BIT THR SET ?
Y
c
N
IS BIT FOU SET ?
Y
d
N
IS BIT NIN SET ?
Y
e
LOGO1
IS BIT FFON SET ?
N
Y
LOGO101
POINT DPTR TO MSG106
CLEAR BIT FORMF
SET BIT PORT1
PRT
CLEAR BIT FORMF
POINT DPTR TO MSG107
SET BIT PORT1
PRT
LOGO2
N
IS BIT PAR SET ?
Y
SET BIT FIN1
CLEAR BIT PORT1
POINT DPTR TO MSG109
PRT
LOGO201
SET BIT FIN1
CLEAR BIT PORT1
POINT DPTR TO MSG108
PRT

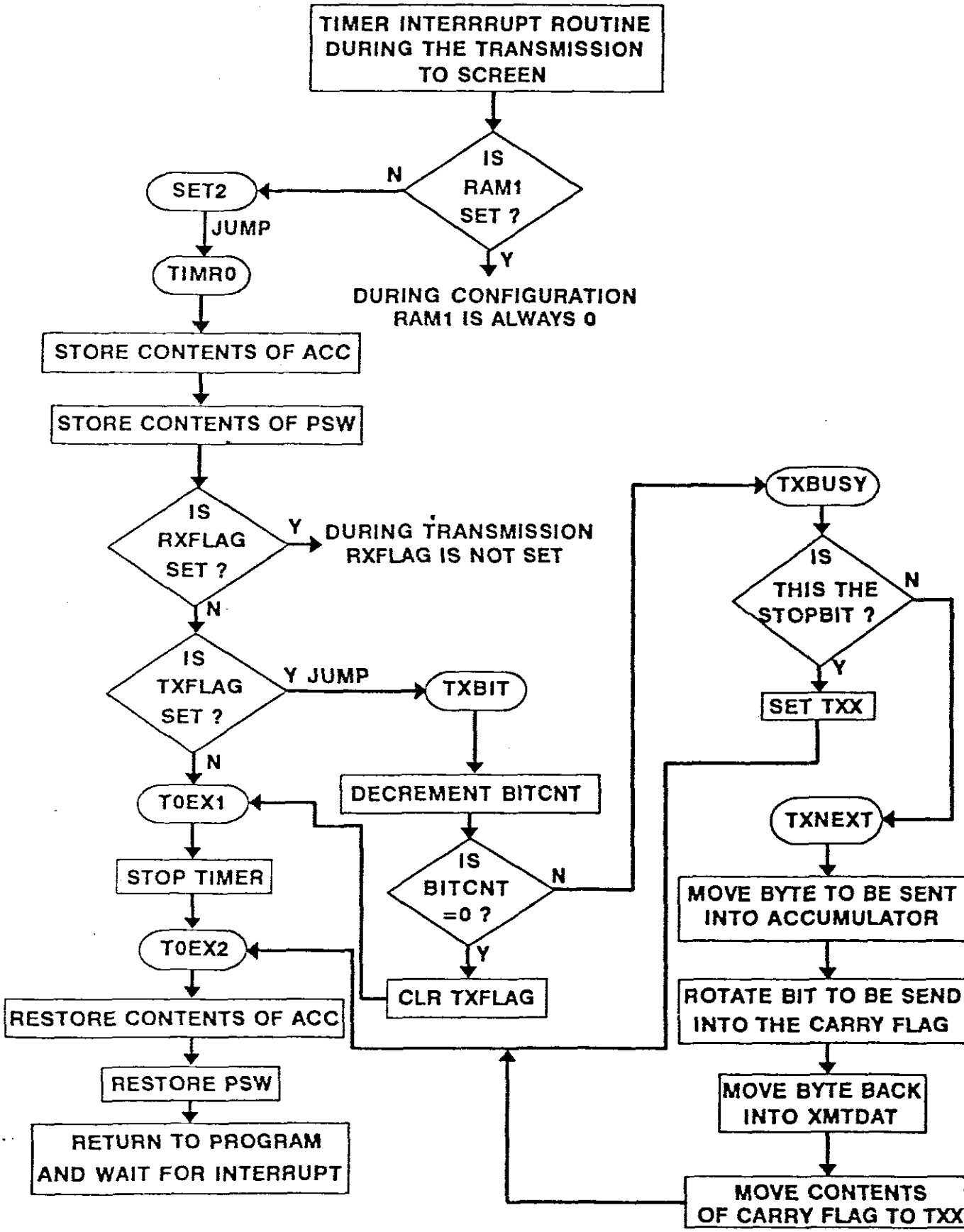

a
POINT DPTR
TO MSG101
SET BIT FORMF
PRT

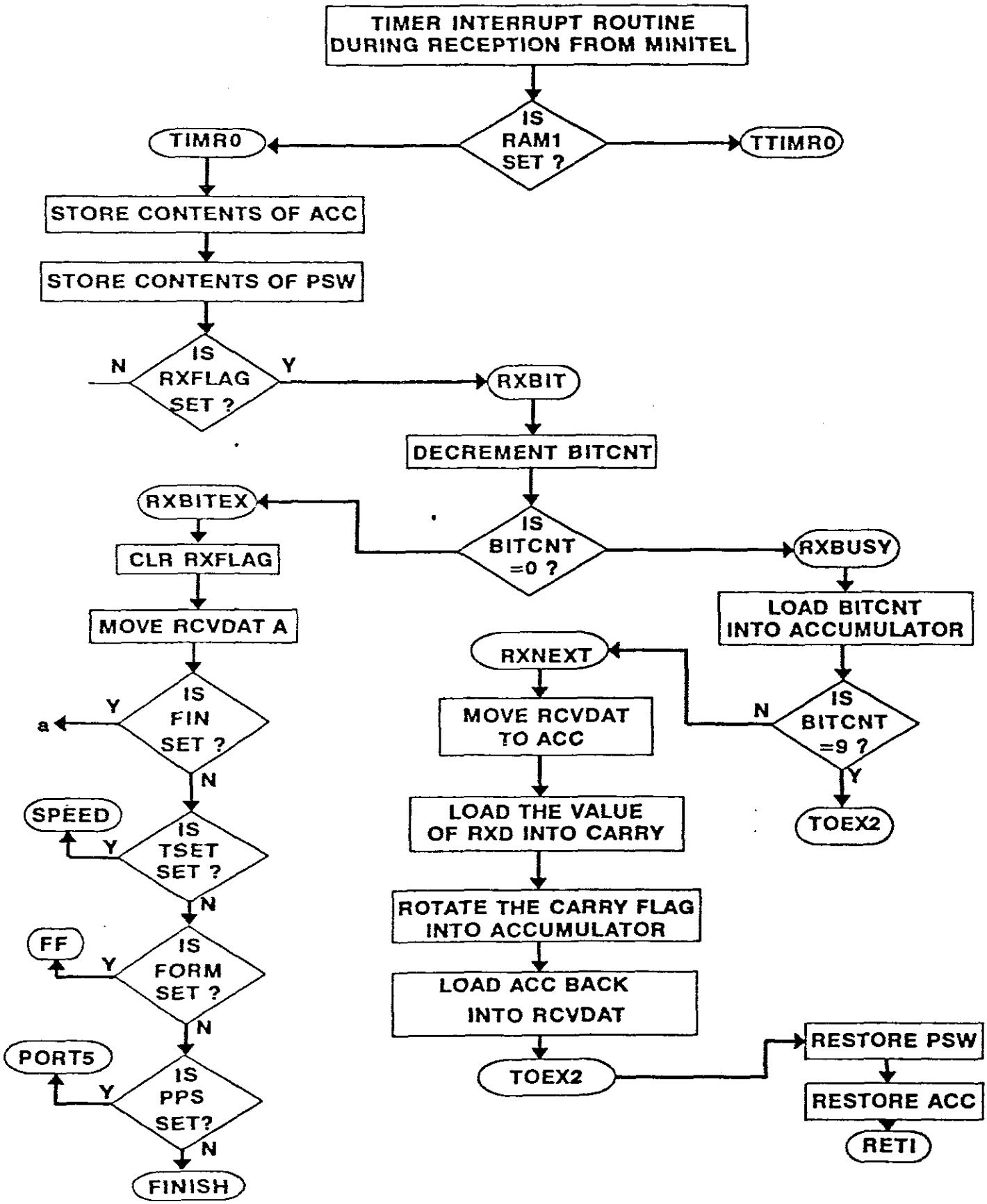

b
POINT DPTR
TO MSG102
SET BIT FORMF
PRT

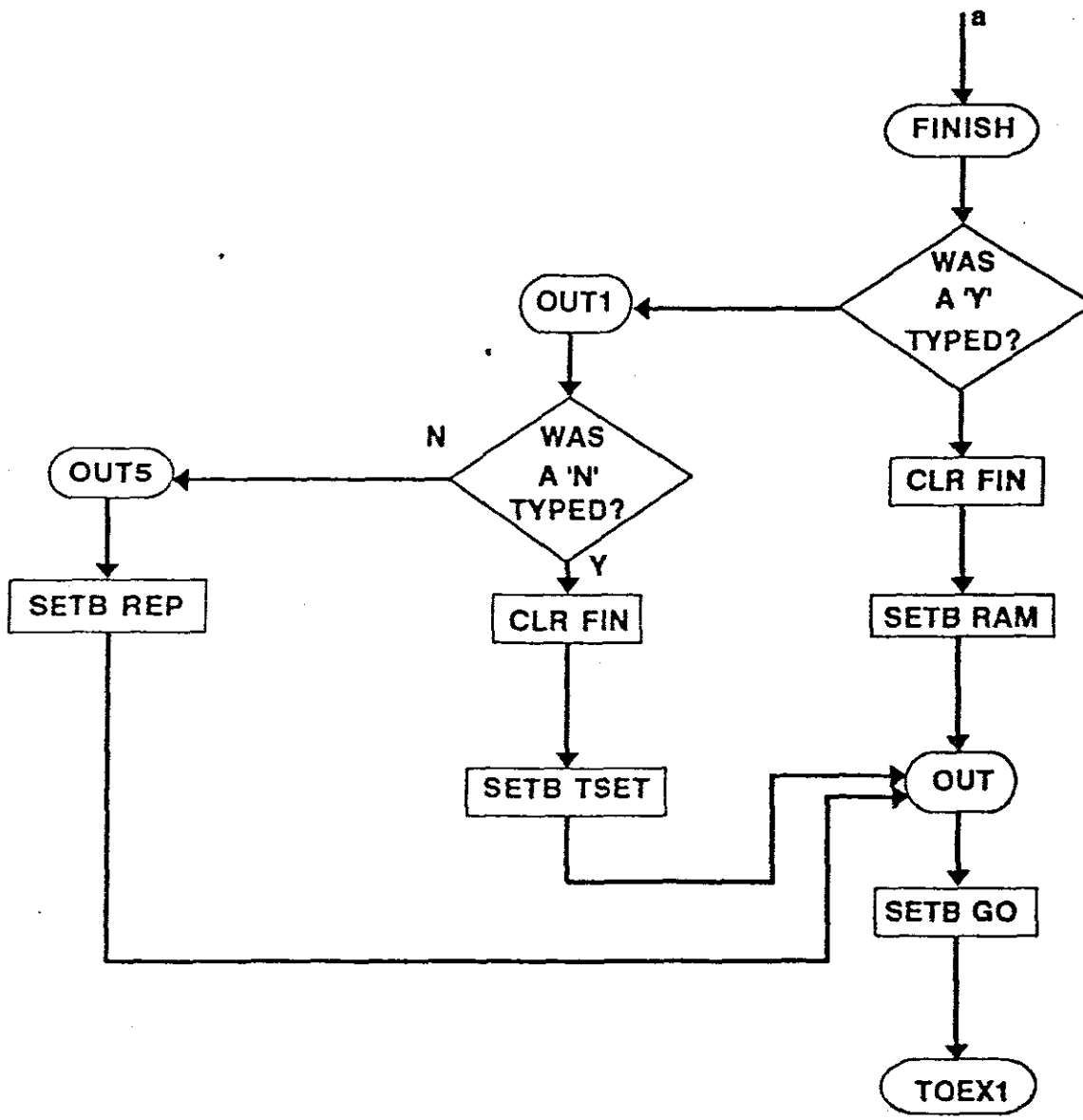

c
POINT DPTR
TO MSG103
SET BIT FORMF
PRT

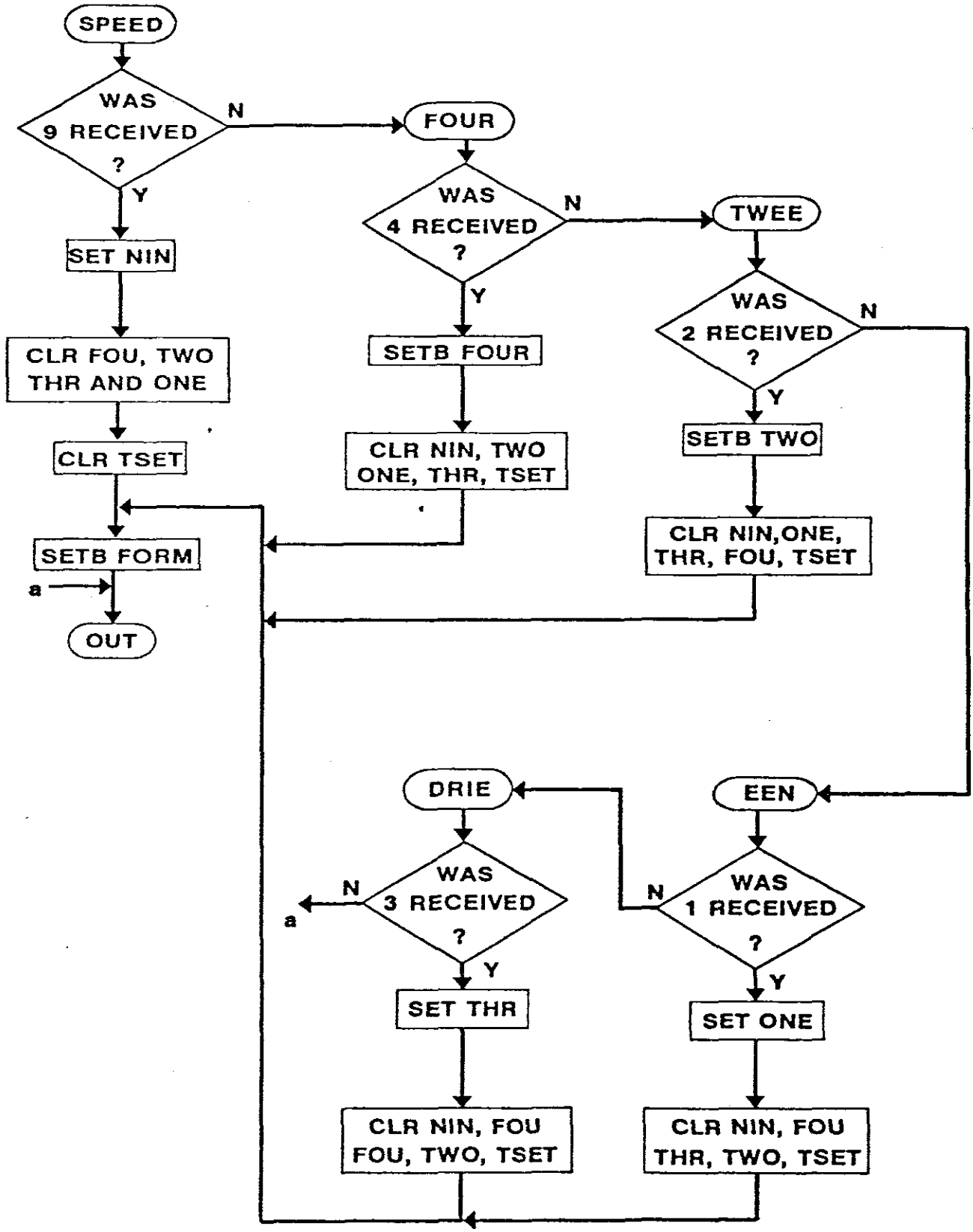

d
POINT DPTR
TO MSG104
SET BIT FORMF
PRT

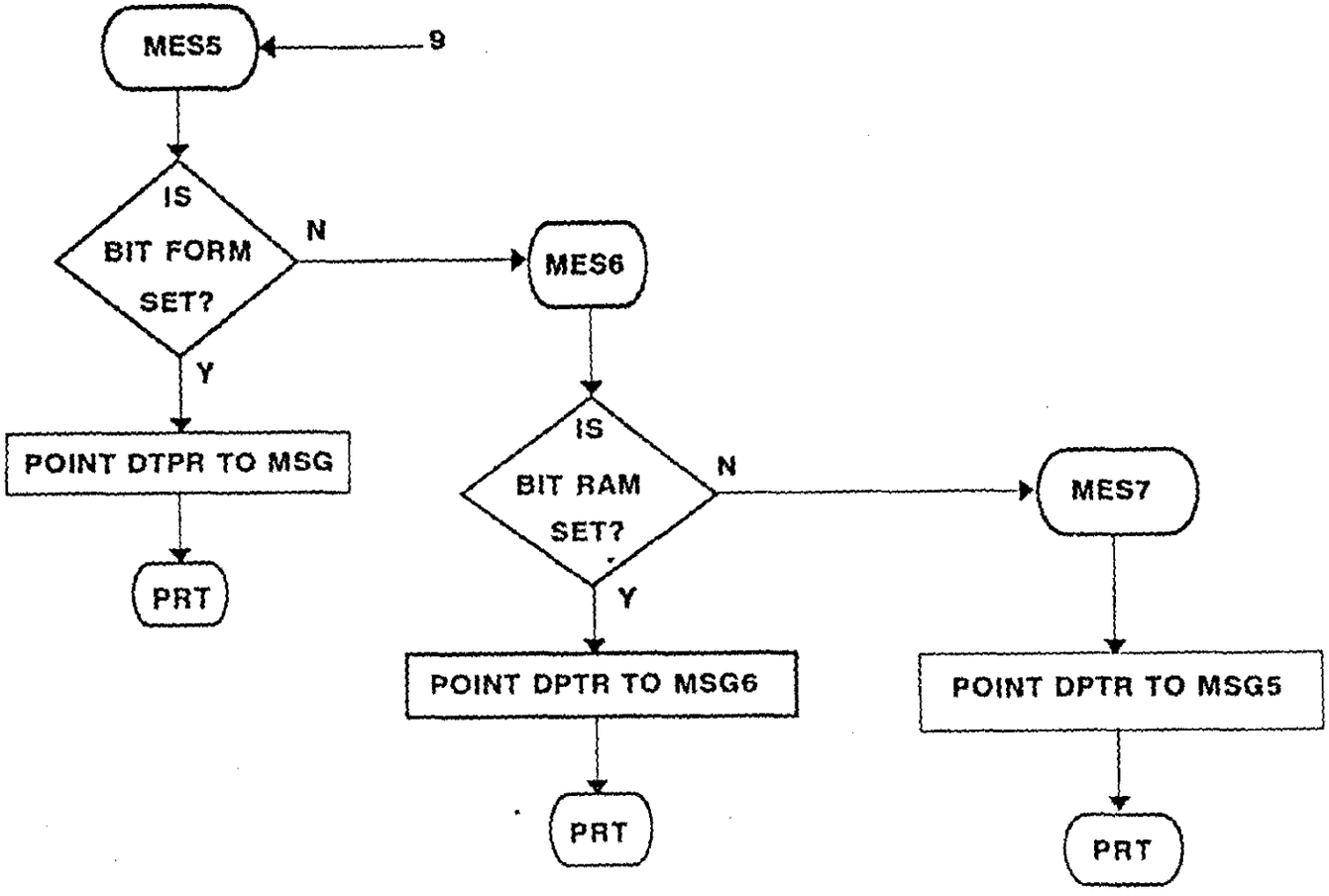

e
POINT DPTR
TO MSG105
SET BIT FORMF
PRT

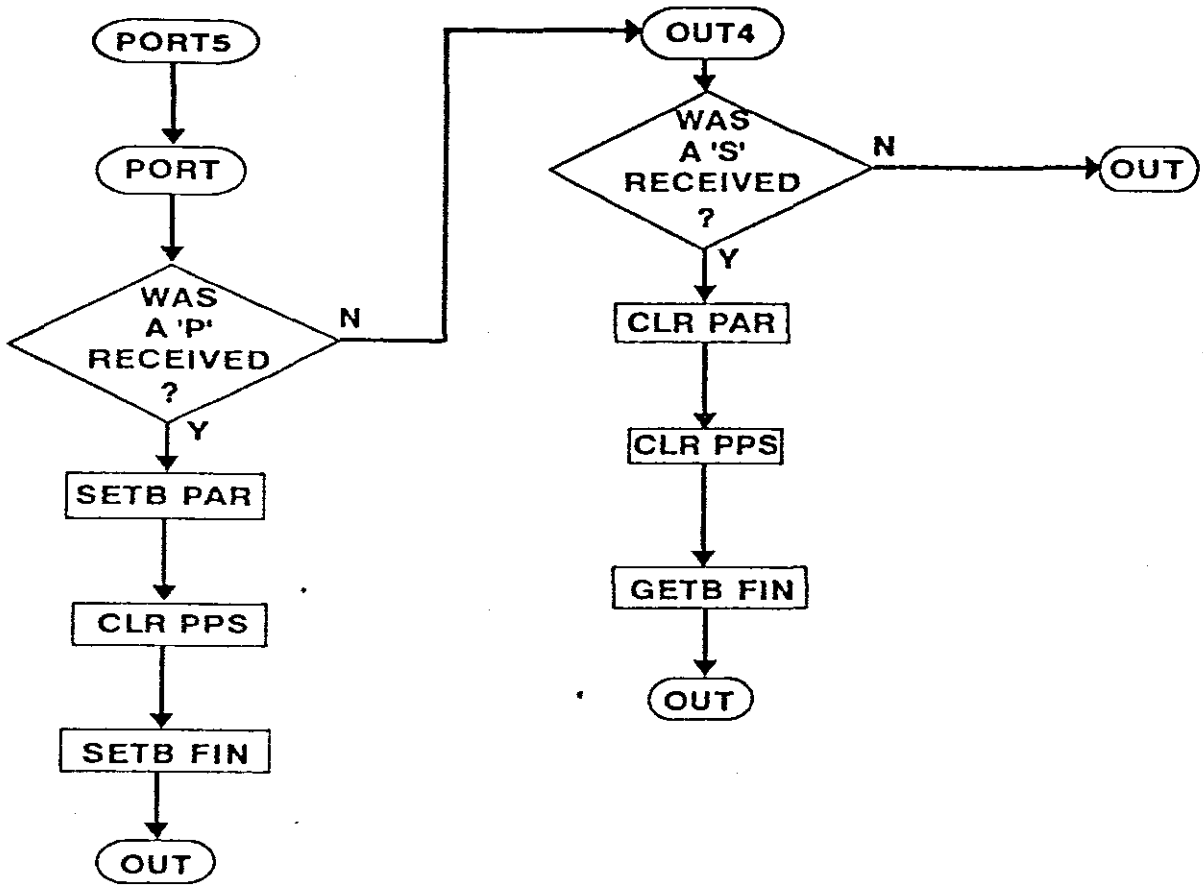

TIMER INTERRRUPT ROUTINE DURING THE TRANSMISSION TO SCREEN
IS RAM1 SET ?
N
Y
SET2
JUMP
TIMR0
DURING CONFIGURATION RAM1 IS ALWAYS 0
STORE CONTENTS OF ACC
STORE CONTENTS OF PSW
IS RXFLAG SET ?
Y
DURING TRANSMISSION RXFLAG IS NOT SET
N
IS TXFLAG SET ?
Y JUMP
TXBIT
N
T0EX1
DECREMENT BITCNT
STOP TIMER
IS BITCNT =0 ?
N
Y
T0EX2
CLR TXFLAG
RESTORE CONTENTS OF ACC
RESTORE PSW
RETURN TO PROGRAM AND WAIT FOR INTERRUPT
TXBUSY
IS THIS THE STOPBIT ?
N
Y
SET TXX
TXNEXT
MOVE BYTE TO BE SENT INTO ACCUMULATOR
ROTATE BIT TO BE SEND INTO THE CARRY FLAG
MOVE BYTE BACK INTO XMTDAT
MOVE CONTENTS OF CARRY FLAG TO TXX

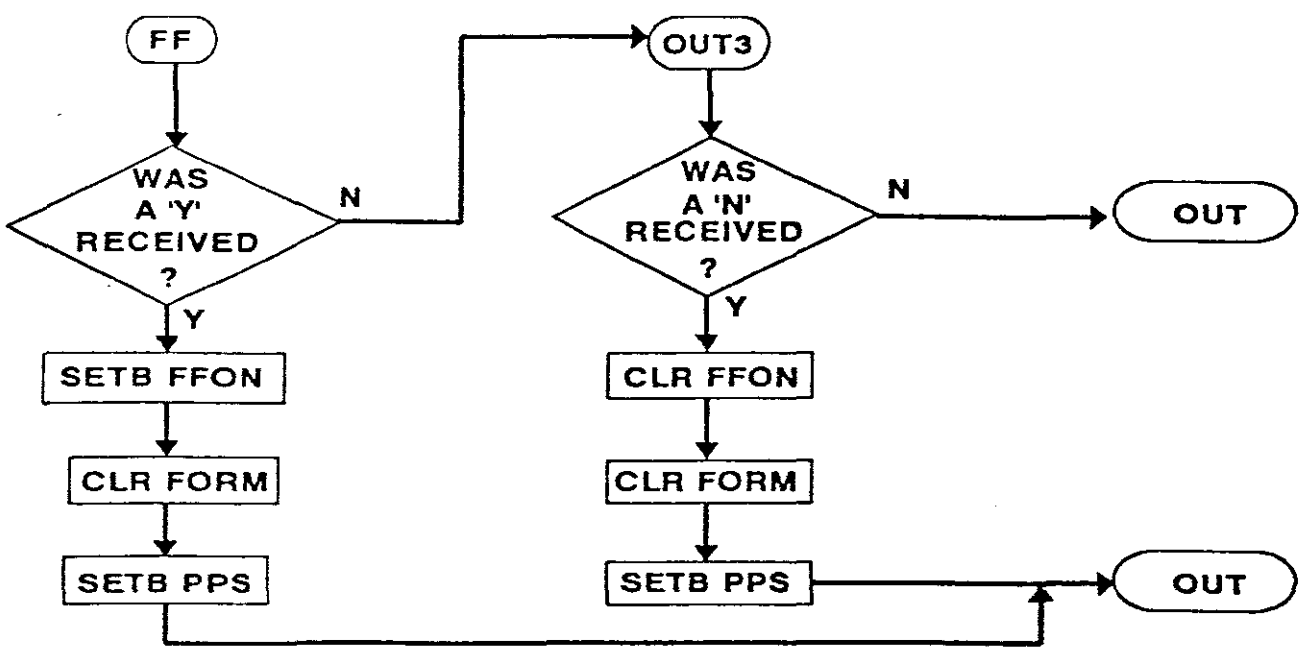

TIMER INTERRUPT ROUTINE DURING RECEPTION FROM MINITEL
IS RAM1 SET ?
TIMR0
TTIMR0
STORE CONTENTS OF ACC
STORE CONTENTS OF PSW
IS RXFLAG SET ?
N
Y
RXBIT
DECREMENT BITCNT
IS BITCNT =0 ?
RXBITEX
CLR RXFLAG
MOVE RCVDAT A
RXBUSY
LOAD BITCNT INTO ACCUMULATOR
IS BITCNT =9 ?
N
Y
RXNEXT
MOVE RCVDAT TO ACC
LOAD THE VALUE OF RXD INTO CARRY
ROTATE THE CARRY FLAG INTO ACCUMULATOR
LOAD ACC BACK INTO RCVDAT
TOEX2
TOEX2
RESTORE PSW
RESTORE ACC
RETI
IS FIN SET ?
Y
a
N
IS TSET SET ?
SPEED
Y
N
IS FORM SET ?
FF
Y
N
IS PPS SET?
PORT5
Y
N
FINISH

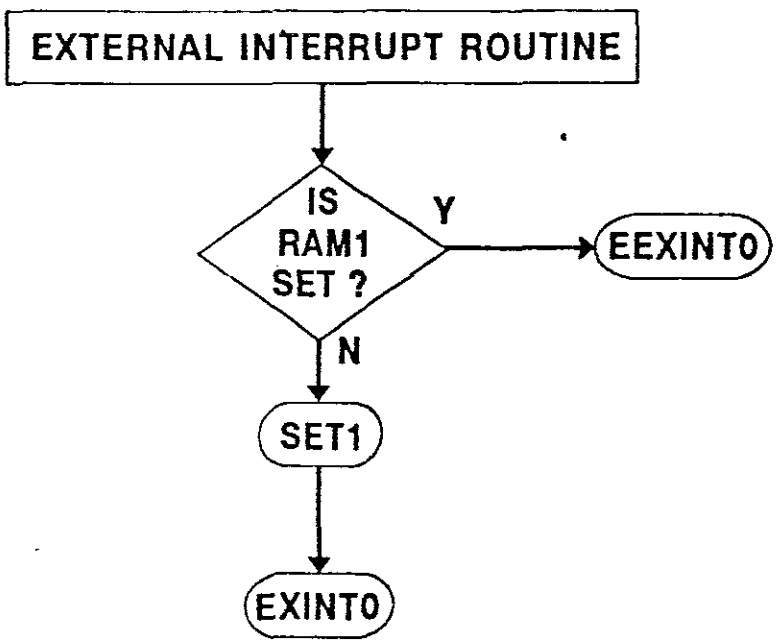

a
FINISH
WAS
A 'Y'
TYPED?
OUT1
N
WAS
A 'N'
TYPED?
OUT5
Y
CLR FIN
SETB REP
SETB RAM
SETB TSET
OUT
SETB GO
TOEX1

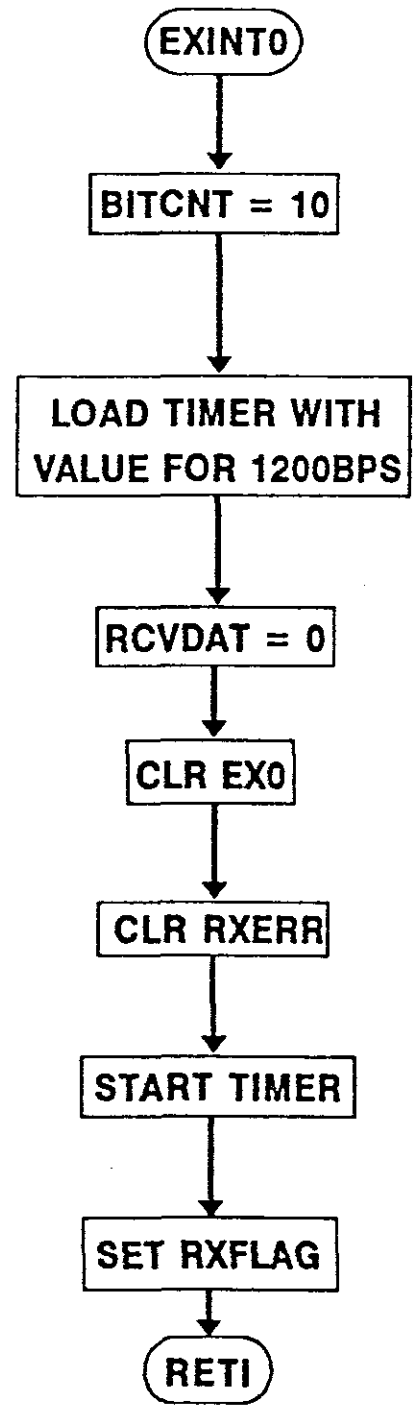

SPEED
WAS 9 RECEIVED ?
N
Y
FOUR
WAS 4 RECEIVED ?
N
Y
TWEE
WAS 2 RECEIVED ?
N
Y
SET NIN
CLR FOU, TWO THR AND ONE
CLR TSET
SETB FOUR
CLR NIN, TWO ONE, THR, TSET
SETB TWO
CLR NIN,ONE, THR, FOU, TSET
SETB FORM
a
OUT
DRIE
EEN
WAS 3 RECEIVED ?
N
a
Y
WAS 1 RECEIVED ?
N
Y
SET THR
SET ONE
CLR NIN, FOU FOU, TWO, TSET
CLR NIN, FOU THR, TWO, TSET


MES5
9
IS
BIT FORM
SET?
N
Y
MES6
POINT DTPR TO MSG
PRT
IS
BIT RAM
SET?
N
Y
MES7
POINT DPTR TO MSG6
POINT DPTR TO MSG5
PRT
PRT

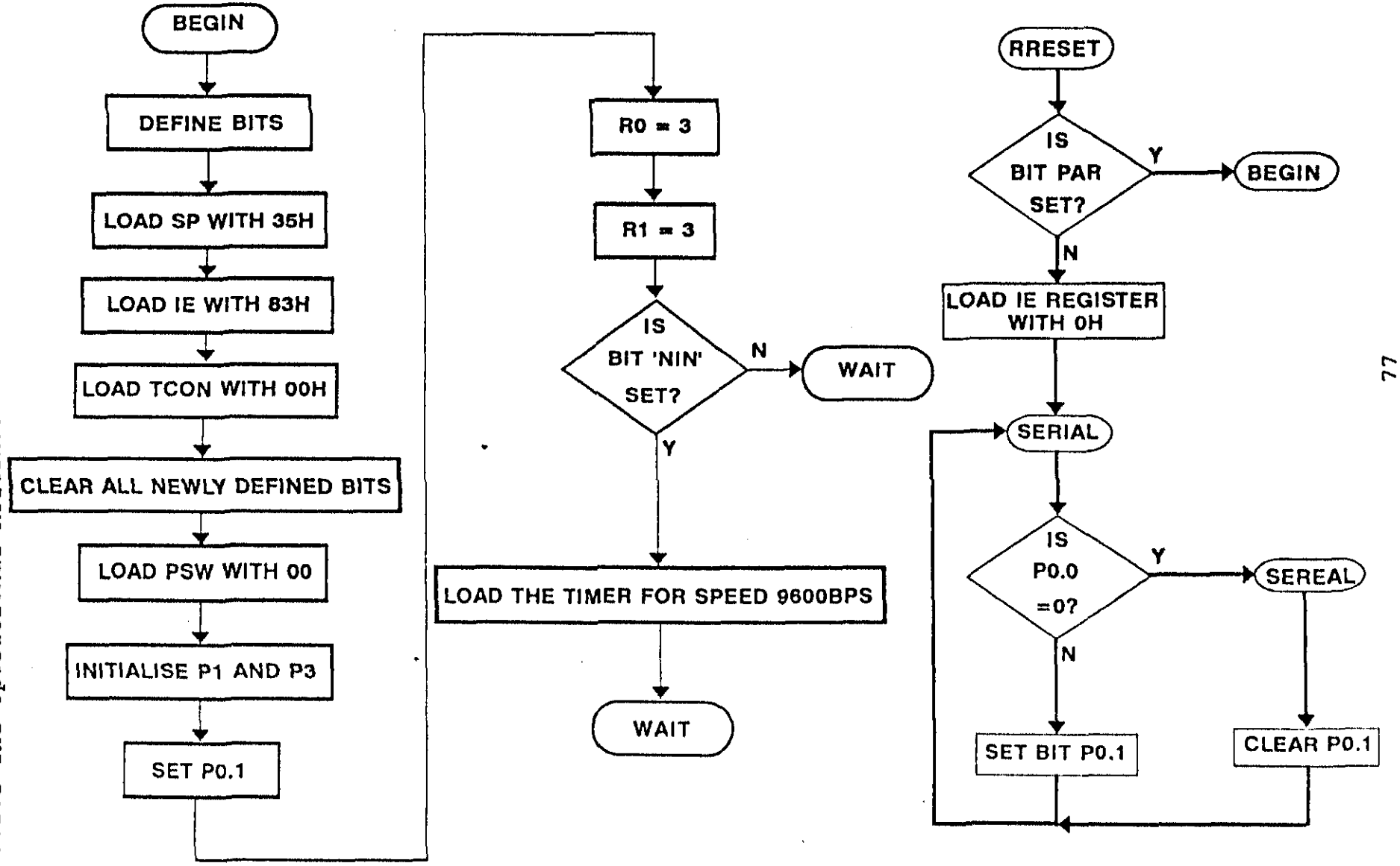

PORT5
PORT
WAS A 'P' RECEIVED ?
N
Y
SETB PAR
CLR PPS
SETB FIN
OUT
OUT4
WAS A 'S' RECEIVED ?
N
OUT
Y
CLR PAR
CLR PPS
GETB FIN
OUT


FF
WAS A 'Y' RECEIVED ?
N
Y
SETB FFON
CLR FORM
SETB PPS
OUT3
WAS A 'N' RECEIVED ?
N
OUT
Y
CLR FFON
CLR FORM
SETB PPS
OUT

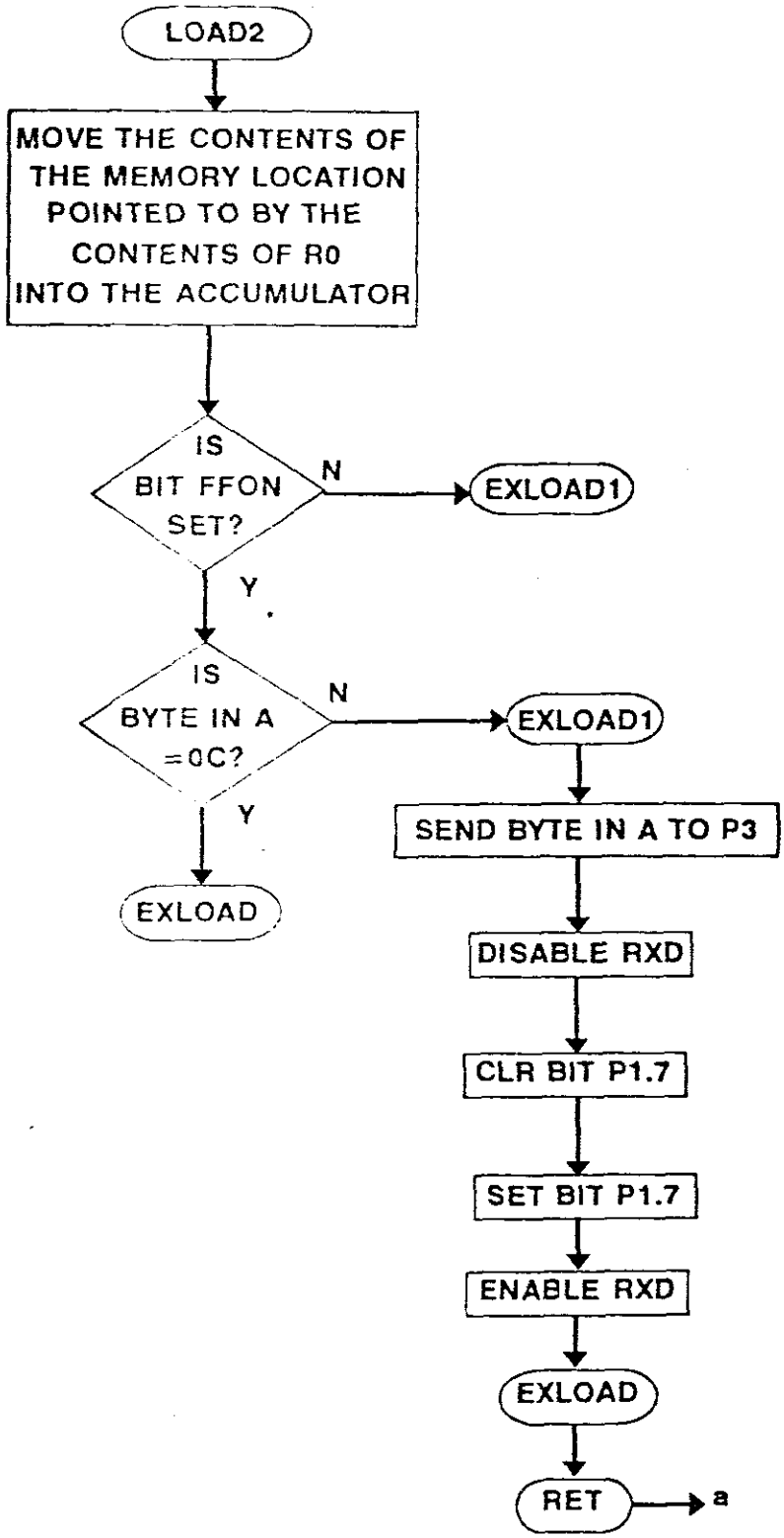

EXTERNAL INTERRUPT ROUTINE
IS RAM1 SET ?
Y
EEXINT0
N
SET1
EXINT0

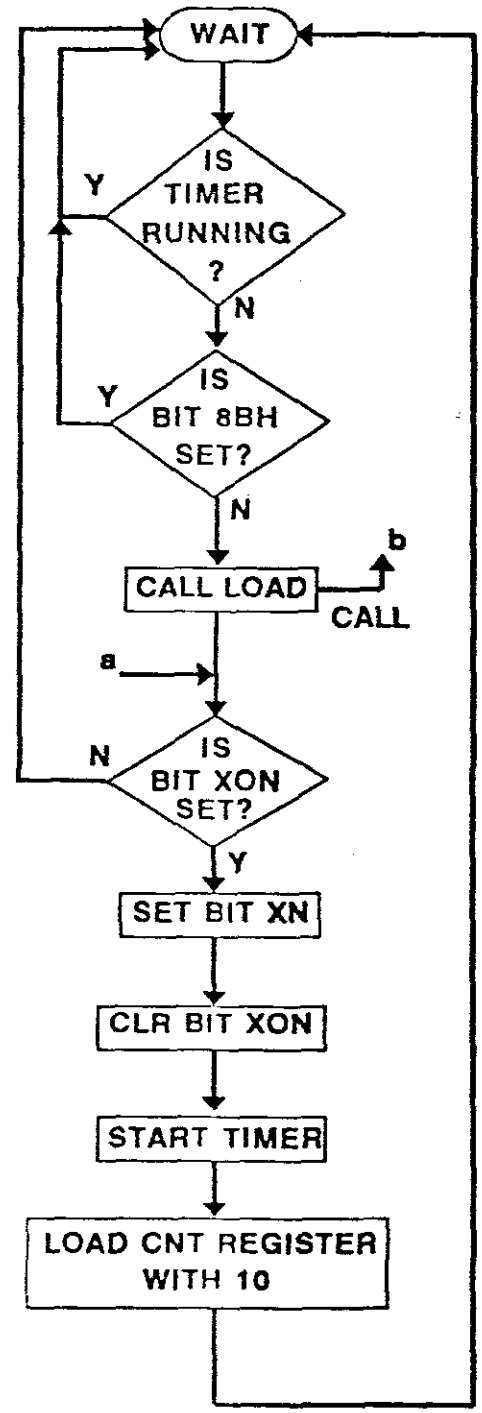

EXINT0
BITCNT = 10
LOAD TIMER WITH VALUE FOR 1200BPS
RCVDAT = 0
CLR EX0
CLR RXERR
START TIMER
SET RXFLAG
RETI

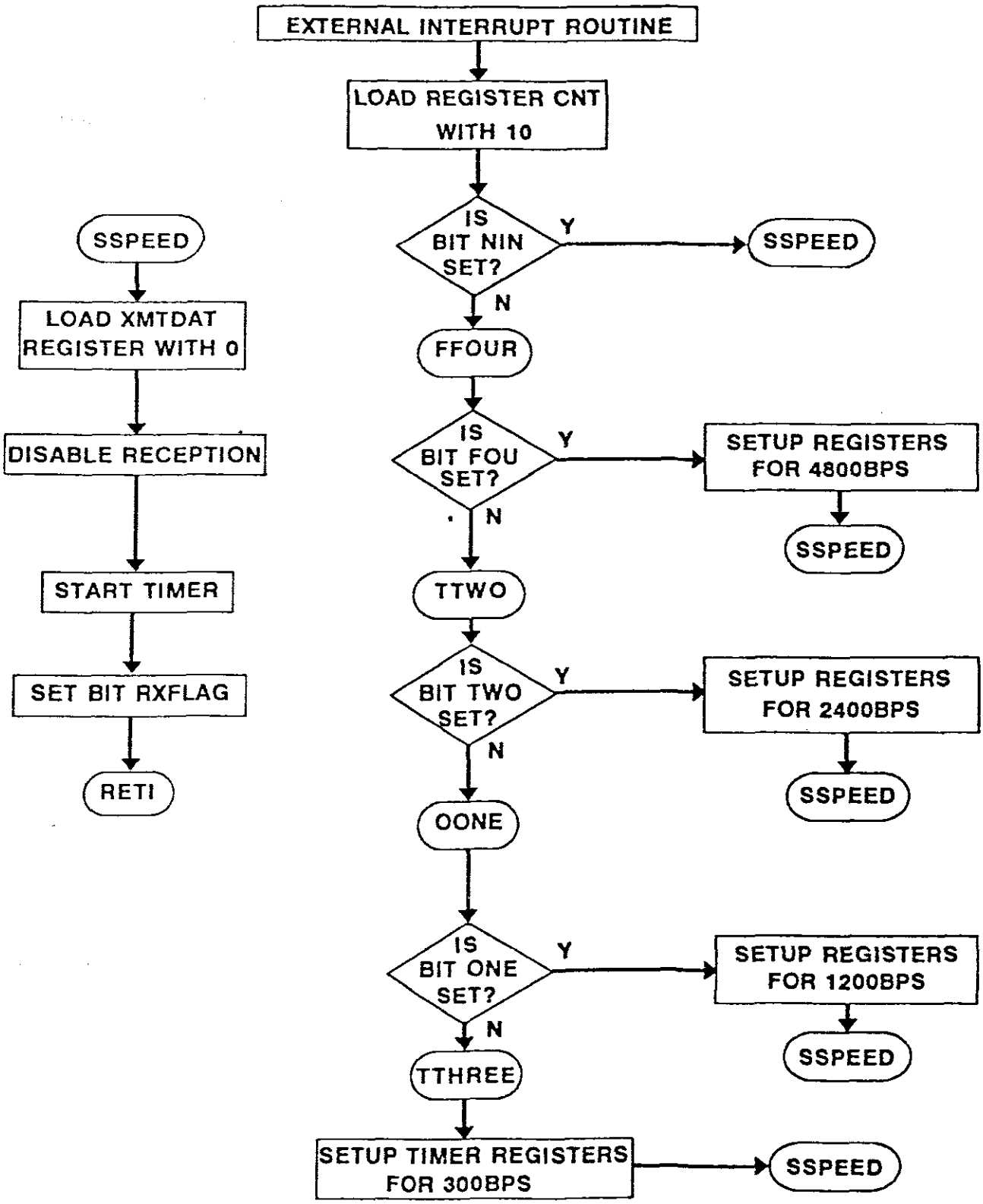
THE OPERATIONAL ROUTINE
AFTER CONFIGURATION

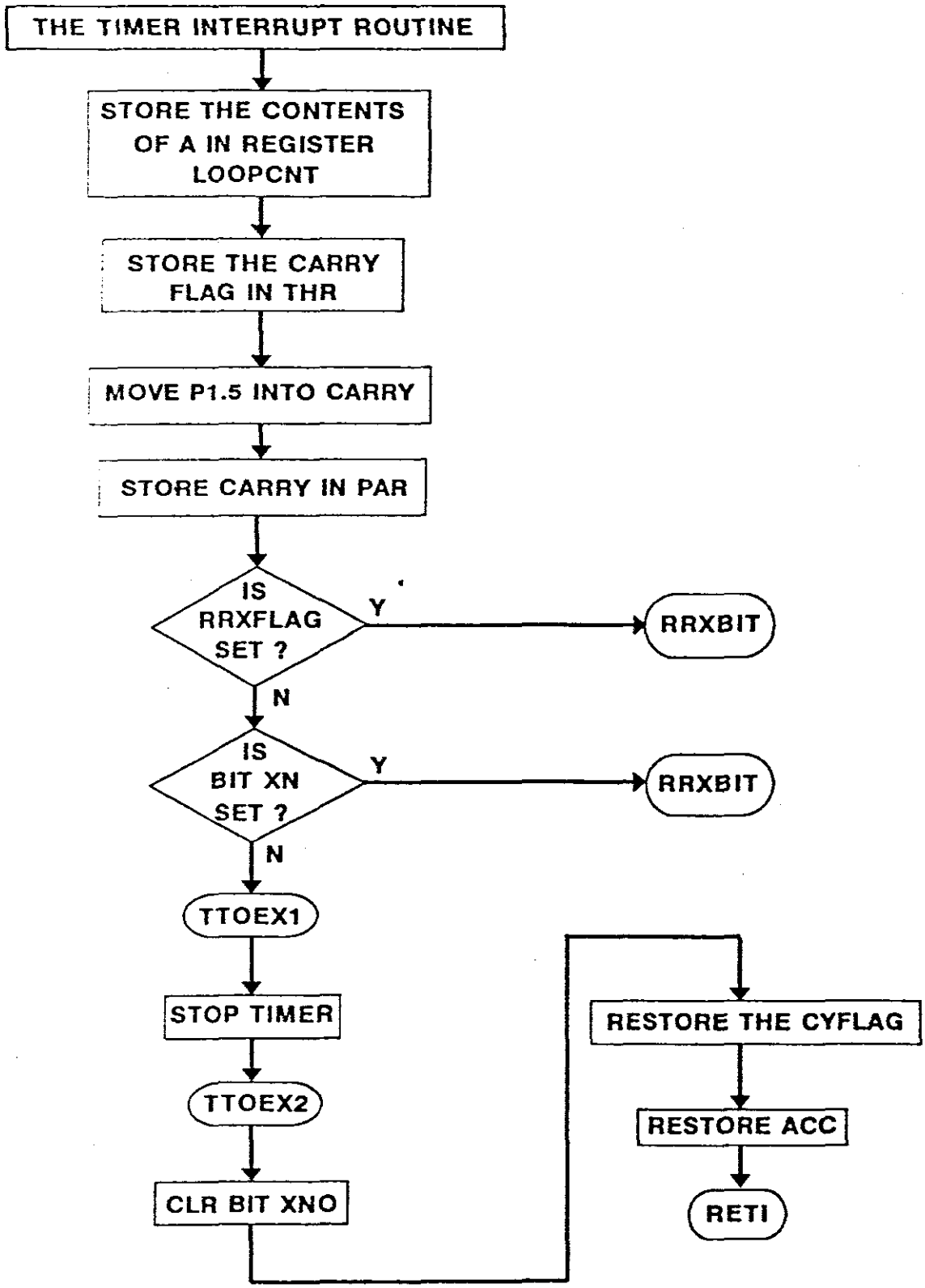


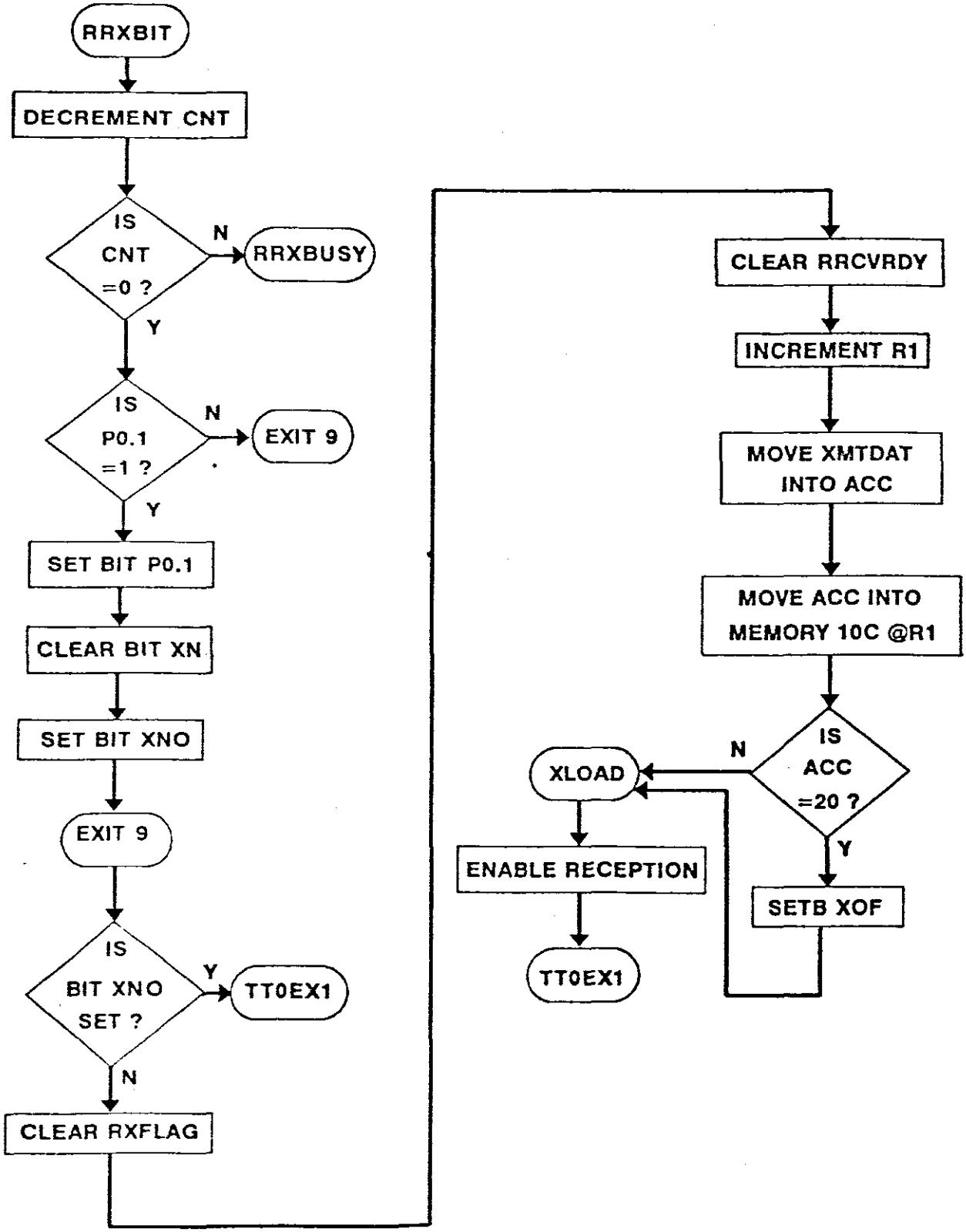
7.2.2 The Operational Routine.

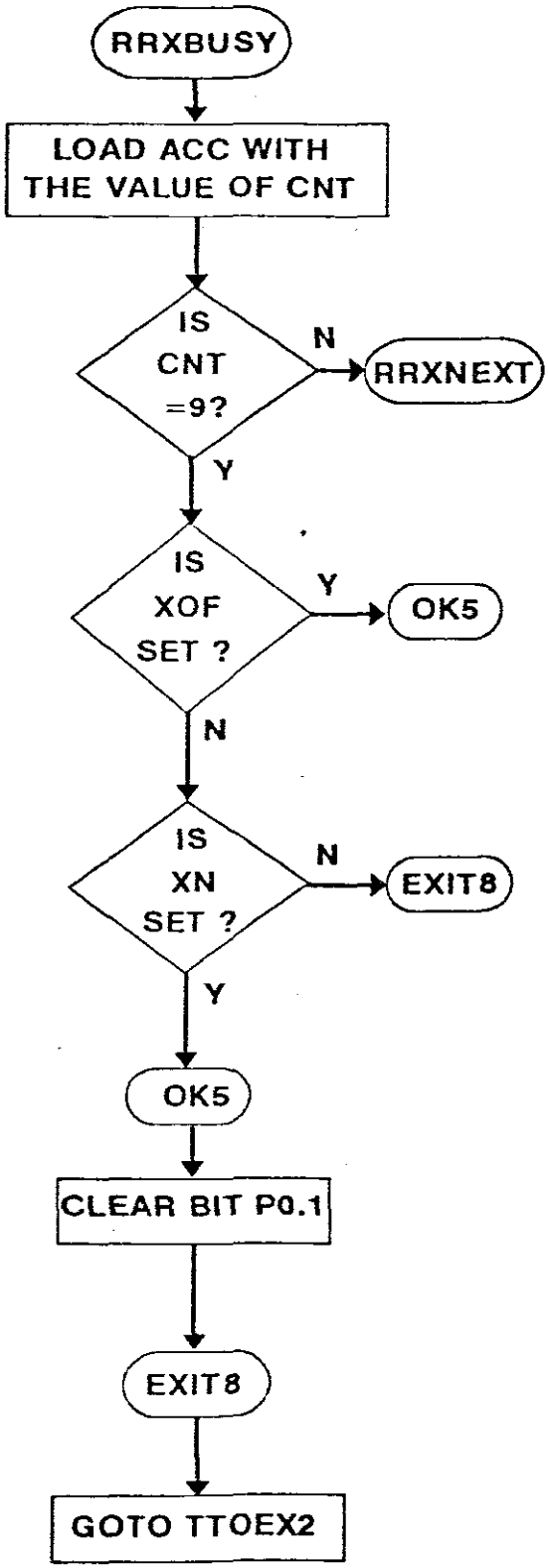

LOAD2
MOVE THE CONTENTS OF THE MEMORY LOCATION POINTED TO BY THE CONTENTS OF R0 INTO THE ACCUMULATOR
IS BIT FFON SET?
N
EXLOAD1
Y
IS BYTE IN A =0C?
N
EXLOAD1
Y
EXLOAD
SEND BYTE IN A TO P3
DISABLE RXD
CLR BIT P1.7
SET BIT P1.7
ENABLE RXD
EXLOAD
RET
a

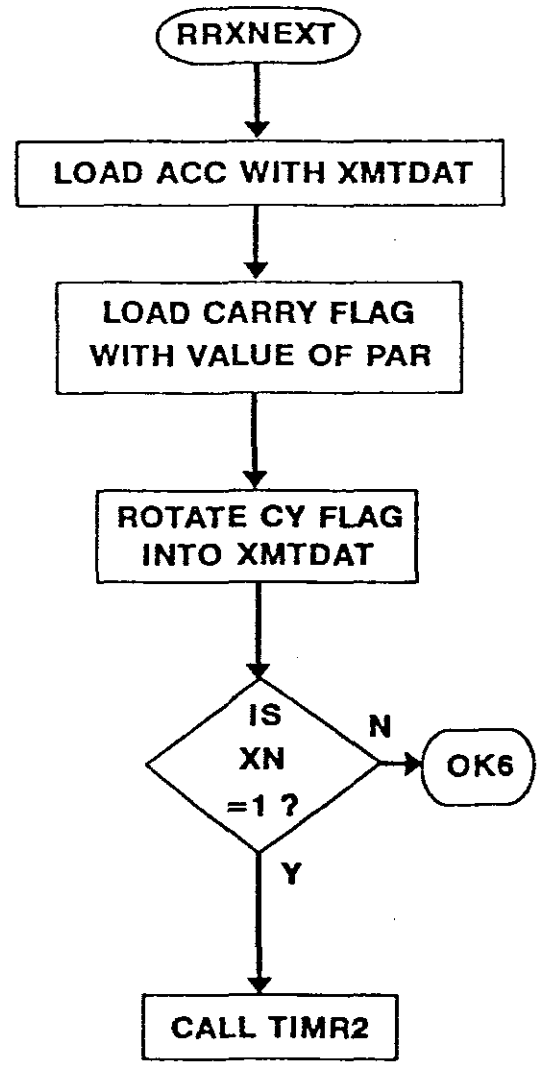

WAIT
IS TIMER RUNNING ?
Y
N
IS BIT 8BH SET?
Y
N
CALL LOAD
b
CALL
a
IS BIT XON SET?
N
Y
SET BIT XN
CLR BIT XON
START TIMER
LOAD CNT REGISTER WITH 10

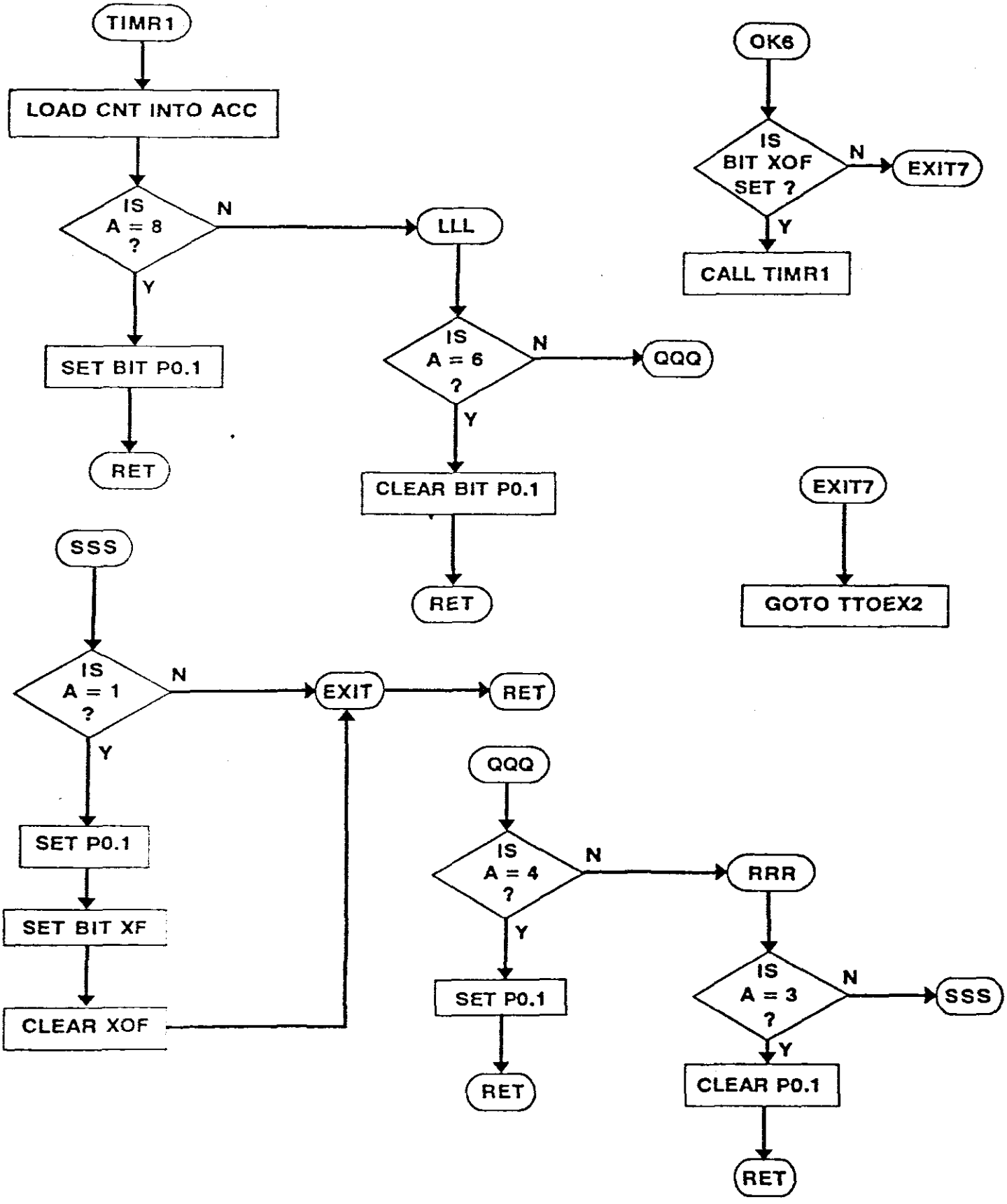

EXTERNAL INTERRUPT ROUTINE
LOAD REGISTER CNT WITH 10
IS BIT NIN SET?
Y
SSPEED
N
FFOUR
IS BIT FOU SET?
Y
SETUP REGISTERS FOR 4800BPS
SSPEED
N
TTWO
IS BIT TWO SET?
Y
SETUP REGISTERS FOR 2400BPS
SSPEED
N
OONE
IS BIT ONE SET?
Y
SETUP REGISTERS FOR 1200BPS
SSPEED
N
TTHREE
SETUP TIMER REGISTERS FOR 300BPS
SSPEED
SSPEED
LOAD XMTDAT REGISTER WITH 0
DISABLE RECEPTION
START TIMER
SET BIT RXFLAG
RETI

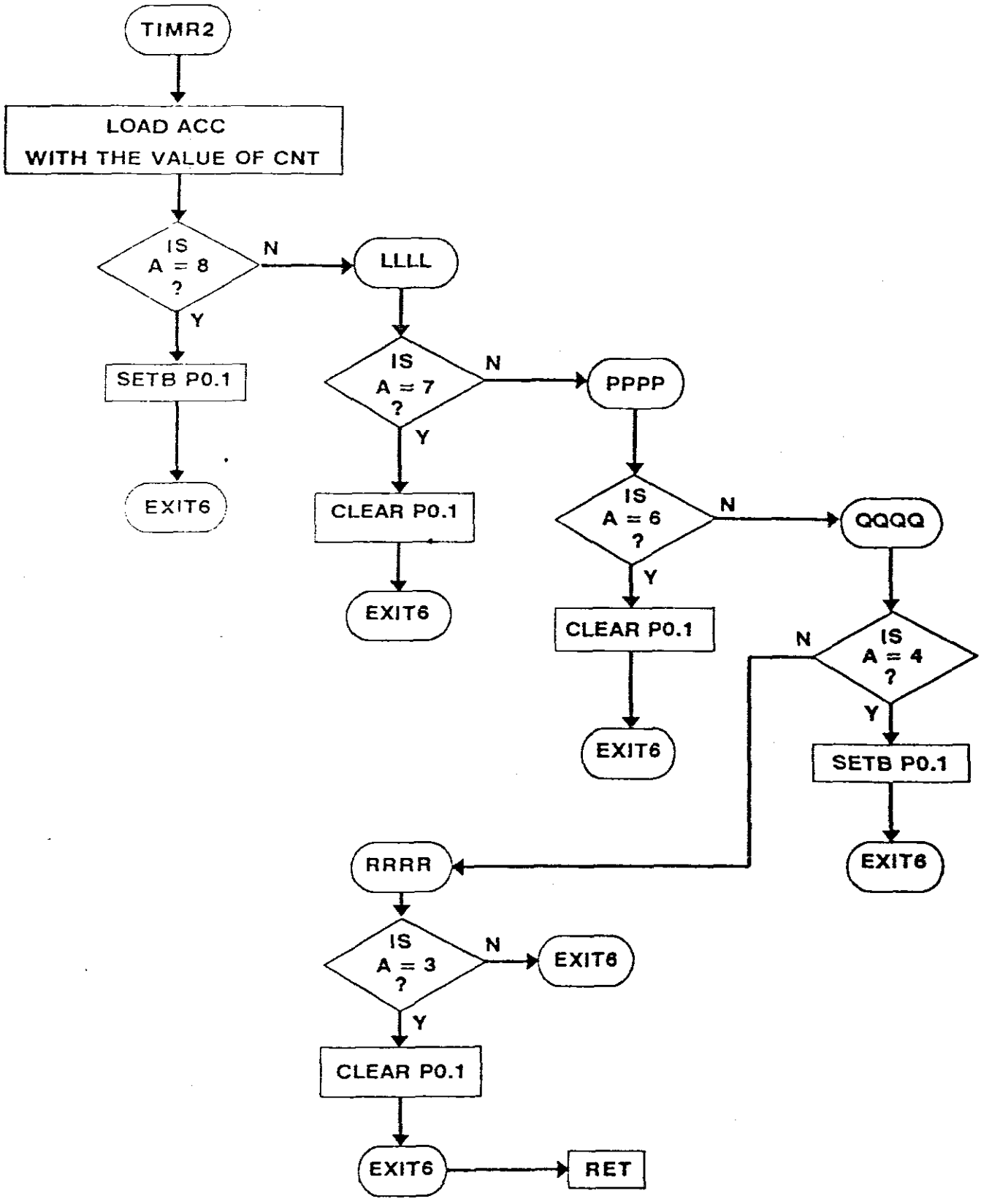

THE TIMER INTERRUPT ROUTINE
STORE THE CONTENTS OF A IN REGISTER LOOPCNT
STORE THE CARRY FLAG IN THR
MOVE P1.5 INTO CARRY
STORE CARRY IN PAR
IS RRXFLAG SET ?
Y
RRXBIT
N
IS BIT XN SET ?
Y
RRXBIT
N
TTOEX1
STOP TIMER
TTOEX2
CLR BIT XNO
RESTORE THE CYFLAG
RESTORE ACC
RETI


RRXBIT
DECREMENT CNT
IS CNT =0 ?
N
RRXBUSY
Y
IS P0.1 =1 ?
N
EXIT 9
Y
SET BIT P0.1
CLEAR BIT XN
SET BIT XNO
EXIT 9
IS BIT XNO SET ?
Y
TT0EX1
N
CLEAR RXFLAG
CLEAR RRCVRDY
INCREMENT R1
MOVE XMTDAT INTO ACC
MOVE ACC INTO MEMORY 10C @R1
IS ACC =20 ?
N
XLOAD
Y
SETB XOF
ENABLE RECEPTION
TT0EX1


RRXBUSY
LOAD ACC WITH THE VALUE OF CNT
IS CNT =9?
N
RRXNEXT
Y
IS XOF SET ?
Y
OK5
N
IS XN SET ?
N
EXIT8
Y
OK5
CLEAR BIT P0.1
EXIT8
GOTO TTOEX2


RRXNEXT
LOAD ACC WITH XMTDAT
LOAD CARRY FLAG WITH VALUE OF PAR
ROTATE CY FLAG INTO XMTDAT
IS XN =1 ?
N
OK6
Y
CALL TIMR2


TIMR1
LOAD CNT INTO ACC
IS A = 8 ?
N
Y
SET BIT P0.1
RET
LLL
IS A = 6 ?
N
QQQ
Y
CLEAR BIT P0.1
RET
OK6
IS BIT XOF SET ?
N
EXIT7
Y
CALL TIMR1
EXIT7
GOTO TTOEX2
SSS
IS A = 1 ?
N
EXIT
RET
Y
SET P0.1
SET BIT XF
CLEAR XOF
QQQ
IS A = 4 ?
N
RRR
Y
SET P0.1
RET
IS A = 3 ?
N
SSS
Y
CLEAR P0.1
RET


TIMR2
LOAD ACC
WITH THE VALUE OF CNT
IS
A = 8
?
N
Y
LLLL
SETB P0.1
EXIT6
IS
A = 7
?
N
Y
PPPP
CLEAR P0.1
EXIT6
IS
A = 6
?
N
Y
QQQQ
CLEAR P0.1
EXIT6
IS
A = 4
?
N
Y
SETB P0.1
EXIT6
RRRR
IS
A = 3
?
N
Y
EXIT6
CLEAR P0.1
EXIT6
RET

8. The Operating Instructions.

8.1 The 8031 Model.

Set up the the dipswitches according to the following table:

Suppress formfeeds: switch 1 = off

8-Bit data: switch 2 = on

Speed (Bps):

	switch 3	switch 4	switch 5
9600	off	off	off
4800	on	off	off
2400	on	off	on
1200	on	on	off
300	on	on	on

Insert the 5-pin DIN plug of the adaptor into the 5-pin socket of the MINITEL peripheral port. Engage the bottom of the adaptor board into the peripheral slot and push it forward until the top clicks home.

The MINITEL terminal should now be switched on and the peripheral port should be set up according to the instructions in the Minitel Users Guide.

8.2 The 87C751 Model.

Insert the 5-pin DIN plug into the 5-pin DIN socket of the MINITEL terminal.

Switch the terminal on and press "FNCT MEM" to enter LOCAL MODE. The program in the 87C751 waits for 5 seconds and then transmits the following message to the MINITEL screen:

" CONFIGURATION OF THE INTERFACE MODULE
CURRENT CONFIGURATION : SPEED=1200Bps
FF SUPPRESS ON
PARALLEL PORT
IS THE CONFIGURATION CORRECT ? (Y/N)"

Enter either "Y" to indicate that the configuration is correct or "N" to indicate that the configuration is not correct. Any other character entered will cause the prompt to be re-transmitted to the MINITEL.

If "N" is entered the program will transmit the following message to the MINITEL screen:

" ENTER SPEED
9 = 9600Bps
4 = 4800Bps
2 = 2400Bps
1 = 1200Bps
3 = 300Bps "

Press '9' for operation at 9600Bps, '4' for operation at 4800Bps, '2' for operation at 2400Bps, '1' for operation at 1200Bps or '3' for operation at 300Bps.

Any other key will cause the program to retransmit the message to the MINITEL screen.

After pressing a valid key, the program will transmit the following message to the MINITEL screen:

" SUPPRESS FORMFEEDS? (Y/N)"

Press 'Y' to suppress formfeeds or 'N' to print them.
Any other key will cause the program to retransmit the message.

After pressing a valid key, the following message will be transmitted to the MINITEL screen:

" SELECT PARALLEL OR SERIAL OPERATION
   P = PARALLEL
   S = SERIAL
MAKE SELECTION (P/S) "

Press 'P' for parallel operation or 'S' for serial operation.
Any other key will cause the program to retransmit the message.
After pressing a valid key, the operator is prompted to indicate whether the configuration is correct. If 'Y' is pressed the following message is displayed :

" CONFIGURATION OF THE INTERFACE MODULE IS NOW COMPLETE.
  REFER TO THE 'MINITEL USERS GUIDE', IF NECESSARY AND CONFIGURE THE PERIPHERAL PORT OF THE MINITEL TERMINAL. (SEE P.28) "

Hereafter the adaptor enters operational mode.

It should be noted that the SOFTWARE SOLUTION assumes 7-BIT data with even parity since the MINITEL terminal does not support any other data format at present.

The PARALLEL or SERIAL selection activates either the parallel port or the sèrial port.

To change the configuration the MINITEL terminal should be reset, which will force the adaptor module to enter the configuration mode of operation.

9. Problems Encountered

9.1 Hardware Problems

1. False triggering of the timing signals was encountered. This problem was solved by using decoupling capacitors. (Both models)

2. Even though 5V was placed on A14 and A13, the program would not read a '1' on those pins until '1s' was eventually written into the pin latches. (8031 model)

9.2 Software Problems

1. Parallel operation at 9600BPS (87C751 model) initially produced errors on the output to the printer.
A 'race' problem was then diagnosed. When the internal RAM buffer is full and an XOFF has to be transmitted, the time available to sample one bit of the XOFF to the Minitel, is only 104 microseconds. The length of one clock cycle is 180 nanoseconds. One machine cycle is available between bits. (72 2-byte instructions or 48 3-byte instructions)
Solution: All routines involved in this process were optimised. Two of the measures taken to solve this problem illustrates clearly how critical the timing is:

1. PUSH PSW was replaced with MOV THR,C. (Since only the Carry flag was used in the PSW Register, it was not necessary to store the whole PSW, but only the Carry flag). Valuable time was recovered by this measure.

2. PUSH ACC was substituted with a MOV instruction. Once again saving valuable time.

10. Results.

In order to test the serial ports of the two models the TXD and RXD pins on the external RS-232 connector were connected by means of a loop plug. In the case of both models it was found that all characters transmitted from the MINITEL keyboard was successfully echoed back to the screen.

The parallel' port was tested by connecting a printer with a Centronix interface to it. The screen of the MINITEL was then filled with characters. In ASCII mode the MINITEL screen displays a maximum of 2000 characters. 'CTRL P', which is the command to print the screen to a printer, was then repeatedly typed from the keyboard.

In the case of the 8031 Model, the first four print operations completed successfully . Subsequent printing resulted in errors.

In the case of the 87C751 Model, no errors were encountered during repetitive printing.

11. Conclusion.

The operation of the serial port is identical for both the 8031 and the 87C751 models and tested satisfactory in both cases.

The results of the tests performed on the parallel ports of the two models differed. This difference can be attributed to the fact that the 8031 does not implement any flowcontrol and relies on the data buffering only. The buffer fills after printing four pages and subsequent printing results in errors.
In the case of the 87C751 model, XON/XOFF flow control is implemented and therefore no errors occur during printing.

12. Recommendations

The 8031 solution is a very elegant solution, since the board fits neatly into the back of the Minitel terminal. It is also more cost effective than the 87C751 model for small quantities.

The 87C751 model however has the feature of a menudriven configuration routine. It is also guaranteed to not run errors during repetitive printing. If large quantities are to be manufactured, mask programming can be used and in this case the cheaper 83751 can be used.

It is therefore recommended that the 8031 model be used if at all possible, but it should be kept in mind that the 87C751 model becomes feasible in a production environment.

13. Bibliography.

Intel Corporation. 1986. Microcontroller handbook.

Motorola Semiconductors. Data Library Linear IC's Vol.6.

National Semiconductors. Linear Databook.

Maxim Integrated Circuits Databook 1990.

Intel Corporation. 1986. Memory Components Handbook.

Dallas Semiconductor. 1990-1991 Product Handbook.

Philips Components. 1991. 8051-based 8-bit Microcontrollers.

Minitel Reference Guide.

Cybernetic Microsystems. CYS8051 Version 3.0.

RS Components July-October 1990.

14. Acknowledgements.

I wish to thank my colleagues in the Research And Development Centre of SAPONET, in particular Messrs P.Gamble and A.Botha with whom many helpful discussions were held.