Chapter 6
Frequency and Baud Rate Generation

FREQUENCY GENERATION

The Am9513 is capable of synthesizing frequencies with a high degree of precision. For square wave outputs, the counter's Mode register should be set for a TC Toggled output. In this configuration, the output will generate a complete cycle for every two TCs the counter generates. Thus, if the spacing between TCs is controlled by the Load register, the TC Toggled output period (in number of source pulses) will be twice the Load register contents.

A 50% duty cycle output can be generated by operating the counter in Mode D. In this mode, the time between TCs is controlled by the Load register contents. The Hold register is not used in controlling the counter. Figure 6-1 shows an Am8080A/8085A routine for configuring Counter 3 as a baud rate generator.

Some applications require square wave outputs with variable duty cycles. This can be achieved by operating the counter in Mode J. In this mode the reload sources alternate between the

```
0100              00010             ORG 100H
                  00020             ;
                  00030             ;EQU'S FOR AM9513 PORTS
                  00040             ;
0012 =            00050   CMDPRT    EQU     12H       ;COMMAND PORT
0010 =            00060   DATAPRT   EQU     10H       ;DATA PORT
                  00070             ;
                  00080             ;
                  00090             ;
                  00100             ;THIS SUBROUTINE CONFIGURES COUNTER 3 ON
                  00110             ;THE AM9513 AS A BAUD RATE GENERATOR. THE
                  00120             ;INTERNAL OSCILLATOR FREQUENCY IS DIVIDED
                  00130             ;DOWN BY TWICE THE CONTENTS OF VARIABLE
                  00135             ;'FACTOR' AND A SQUARE WAVE OUTPUT IS
                  00137             ;GENERATED.
                  00140             ;
                  00150   BAUD:
                  00160             ;SET COUNTER MODE AND LOAD REGISTERS.
                  00170             ;
0100 3EE0         00171             MVI     A,0E0H    ;ENABLE DATA POINTER
0102 D312         00172             OUT     CMDPRT    ;SEQUENCING.
0104 3E03         00180             MVI     A,03H     ;SET DATA POINTER TO
0106 D312         00190             OUT     CMDPRT    ;COUNTER 3 MODE REG.
0108 3E22         00200             MVI     A,22H     ;LOWER BYTE
010A D310         00210             OUT     DATAPRT
010C 3E0B         00220             MVI     A,0BH     ;UPPER BYTE
010E D310         00230             OUT     DATAPRT
                  00240             ;
                  00250             ;SET COUNTER 3 LOAD REG.
                  00260             ;
0110 3A2301       00270             LDA     FACTOR    ;LOWER BYTE
0113 D310         00280             OUT     DATAPRT
0115 3A2401       00290             LDA     FACTOR+1          ;UPPER BYTE
0118 D310         00300             OUT     DATAPRT
                  00310             ;
                  00320             ;LOAD AND ARM COUNTER
                  00330             ;
011A 3E44         00340             MVI     A,44H     ;LOAD COUNTER 3 COMMAND
011C D312         00350             OUT     CMDPRT
011E 3E24         00360             MVI     A,24H     ;ARM COUNTER 3 COMMAND
0120 D312         00370             OUT     CMDPRT
0122 C9           00380             RET
                  00390             ;
                  00400             ;INTERNAL OSCILLATOR FREQUENCY WILL BE
                  00410             ;DIVIDED BY TWICE CONTENTS OF
                  00420             ;VARIABLE 'FACTOR'.
                  00422             ;
0123 1000         00430   FACTOR:   DW      16        ;DIVIDE BY 32
                  00432             ;
                  00440             ;A CRYSTAL OF FREQUENCY 4.915200
                  00450             ;MHZ. WHEN DIVIDED BY 32 WILL YIELD
                  00460             ;AN OUTPUT FREQUENCY OF 9600 X 16 =
                  00470             ;153600 HZ. FOR 9600 BAUD SERIAL
                  00480             ;COMMUNICATIONS.
                  00490             ;
0125              00500             END
```

Figure 6-1. Baud Rate Generation with the Am9513 in an Am8080A/8085A System

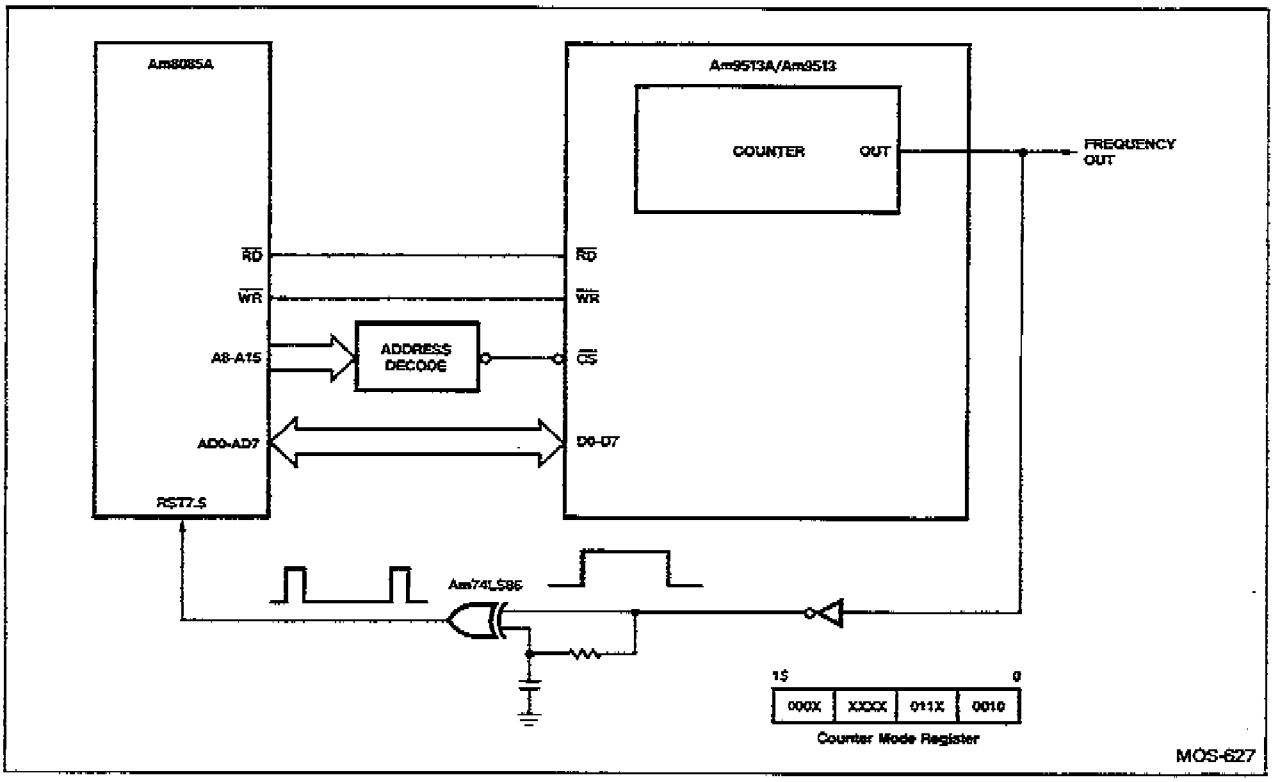

Figure 6-2. Updating Load and Hold Register Values in Real-Time

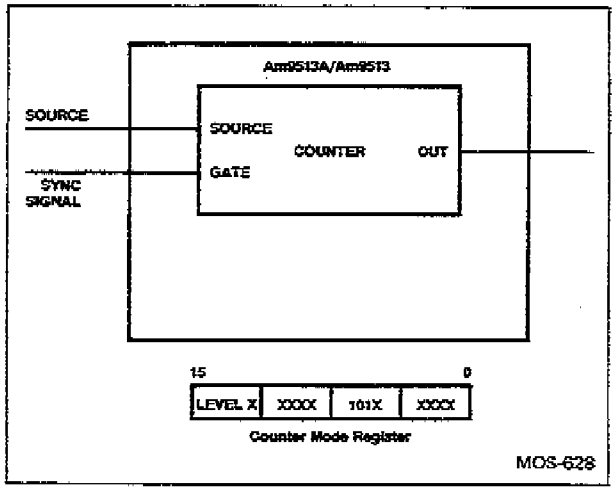

Figure 6-3. External Synchronization of Frequency Counters: Method 1

Load and Hold registers. This allows one polarity of the output to be controlled by the Load register contents and the other output polarity to be controlled by the Hold register. Fine resolution of the output duty cycle can be achieved by adjusting the relative values of the two registers.

In some applications it is necessary to adjust the output frequency in real time. Figure 6-2 shows a circuit which generates an interrupt on each edge of the output waveform. The host microprocessor's interrupt routine would update the Load and/or Hold register contents to adjust the period of the next output cycle. Updating the registers by means of an interrupt routine ensures the register contents are stable when the counter reaches TC and reloads from the register.

The Am9513 has provisions for synchronizing a counter's output frequency to an external signal. To accomplish synchronization, the counter should be operated in Mode Q, as shown in Figure 6-3. For normal counter operation the gate input is held active. While the gate is active, the counter will count to TC repeatedly. When it is necessary to synchronize the counter to an external signal, the gate should be driven inactive, which will stop counting, and then driven active again. On the active-going gate edge, the current counter contents will be saved in the Hold register. This value may be used by the microprocessor as an indication of how out-of-sync the counter was. On the first source edge after the gate edge, the counter will be reloaded from the Load register. Counting will start (assuming the gate remains active) on the second source edge after the gate edge. It can be seen that the gate edge effectively resets the counter, synchronizing it to the gate signal.

Synchronization of the counter to an external event can also be accomplished by issuing a LOAD command to the counter. This will cause the Load or Hold register contents to be transferred into the counter. The selection of whether the Load or Hold register is used as a reload source depends on the counter's operating mode. Note that data sheet parameters TEHWH, TWHEH, TGVWH and TWHGV must be met to ensure successful synchronization. See Appendix A for further details.

Applications which need to generate Frequency-Shift Keying outputs can achieve this function by operating the counter in Mode V. This mode uses the level of the gate input to select the

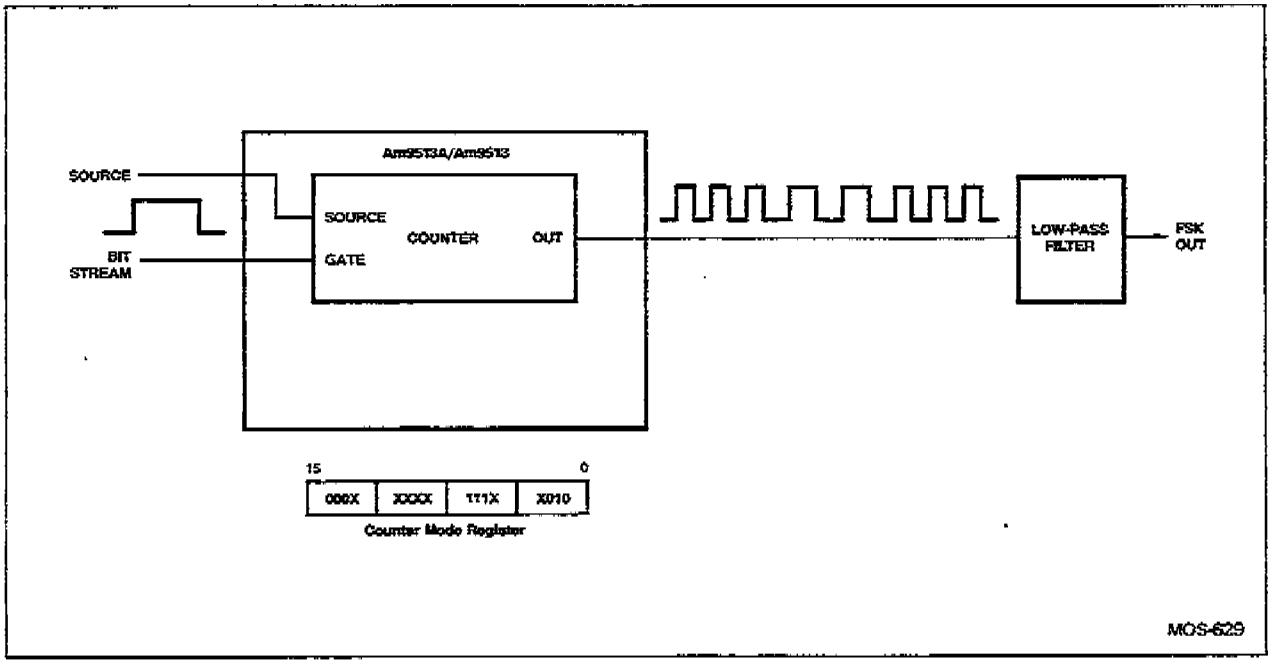

Figure 6-4. Frequency Shift Keying

counter reload source for each TC. For Frequency-Shift Keying (FSK) applications, the serial bit stream should be input to the counter's gate and the counter output periods (as set by the Load and Hold register contents) should be selected to be less than the shortest high or low time on the incoming bit stream. The FSK output can be shaped by a low-pass filter to remove high order harmonics. The output can then be transmitted over a telecommunications link or stored on an audio tape recorder for a low-cost data acquisition system. Demodulation of the FSK signal can be performed using standard techniques.

AUTO BAUD RATE GENERATOR

Many systems require an asynchronous data link that is typically implemented using an Am9513 System Timing Controller and Am8251/Am9551 USART. Since these systems are often shipped without accompanying CRT consoles, they may be set at an arbitrary baud rate to cater for common terminals, usually by means of a buffered bank of dip switches and associated address decode logic. A system capable of automatically sensing the attached terminal baud rate at power-up would save not only switch adjustment by the end user (frequently after dismantling the processor) but also reduce component count, board space and assembly effort at the expense of a few lines of code. Such an example usage of the Am9513 is presented here, using about 160 bytes of code (Am8080/Am8085).

The principle of the baud rate determination algorithm is unrefined but effective. The Am9513 baud rate generator is set to provide the maximum rate anticipated, perhaps 19,200 baud. The system user first enters a carriage return, thus allowing the terminal rate to be measured by counting the pseudo characters received by the system within a short time frame. Some of these characters will be flagged with framing errors by the Am8251/Am9551, although these errors may be safely ignored for our purposes.

The pseudo characters seen by the system, for a given terminal baud rate and asynchronous character format, may be accurately predicted in advance. For example, a 4800 baud terminal will produce the four-character sequence ⟨78⟩⟨7E⟩⟨00⟩⟨7E⟩. However, if the number of pseudo characters received is used as the primary parameter the measurement becomes much more tolerant to skew and variation in the number of bits per word. In practice, many other characters in addition to a carriage return provide correct measurement. The "capture range" of characters entered may be increased by reducing the number of allowable baud rates.

Am8080/Am8085 SAMPLE LISTING

The Am8080/Am8085 code sequence shown in Figure 6-5 initializes the Am8251/Am9551 and the Am9513 to provide an asynchronous data link at 19,200 baud. The received characters are decoded and the new baud rate time constant calculated by using an indexed sequential search. To reduce code, the entire initialization of the Am9513 and Am8251/Am9551 is then recalled using the calculated baud rate time constant. Notice that this practice avoids problems with asynchronous changes of the Am9513 load register when counting in close proximity to TC.

AmZ8000 SAMPLE LISTING

The Z8000 code sequence shown in Figure 6-6 performs the same action as the previous example with the additional facility of changing the Am8251/Am9551 Mode control word for differing baud rates. For example, at 9600 baud frequently 10 bits per byte are utilized compared to 11 bits per byte at 110 baud.

For both examples, successful decoding of the baud rate is indicated by the appearance of the prompt (">") on the users terminal.

Additional baud rates may be allowed by entering the appropriate data in the two data tables (TABLE and RATES). Reducing the number of baud rates available will allow a greater number of characters to give correct operation, although the program is designed to receive a carriage return. As listed, the 11 standard baud rates between 75 and 19,200 baud are allowed.

```
;       Coded for Am8080/Am8085 - also runs on Z80

A_DATA  EQU     0D8H    ; Am9513 date port
A_CTRL  EQU     0DAH    ; Am9513 command port
B_DATA  EQU     0DCH    ; Am8251/Am9551 data port
B_CTRL  EQU     0DDH    ; Am8251/Am9551 command port
RX_RDY  EQU     2       ; Am8251/Am9551 rx ready mask
TX_RDY  EQU     1       ; Am8251/Am9551 tx ready mask
MAXTIM  EQU     5000    ; Time for worst-case reception or, 75 baud
RESETU  EQU     040H    ; Official Am8251/Am9551 software reset
MODE    EQU     0CEH    ; 2 stop, /16, no parity, 8 data bits
COMND   EQU     037H    ; tx/rx enable, dtr, rts

        INCLUDE B:EQUS.MAC      ; Listed in Appendix 7
        INCLUDE B:8080.MAC      ; Listed in Appendix 3

BAUD:
        ; The system monitor calls 'BAUD' to set up console io

        LHLD RATES      ; Start with max rate known to program
        CALL INIT       ; Set up Am9513, Am8251/Am9551
BAUD_1:
        IN   B_CTRL     ; Read console status
        ANI  RX_RDY
        JZ   BAUD_1     ; Wait for 1st character

        MVI  E,0        ; Zero character counter
        LXI  B,MAXTIM   ; Initialise timer
BAUD_2:
        DCX  B          ; Move and test timer status
        MOV  A,B
        ORA  C
        JZ   BAUD_3     ; Timeout, go decode data

        IN   B_CTRL     ; Read console status
        ANI  RX_RDY
        JZ   BAUD_2     ; Wait for a character

        IN   B_DATA     ; Discard pseudo character
        INR  E          ; Adjust character counter
        JMP  BAUD_2     ; Repeat
BAUD_3:
        ; Perform serial index search for reqd baud rate const
        ;
        ; Entry:  E holds # pseudo characters received

        MOV  A,E
        LXI  H,TABLE-1  ; Byte pointer, decode table
        LXI  D,RATES-2  ; Word pointer, rate constant table
BAUD_4:
        INX  D          ; Move the pointers
        INX  D
        INX  H
        CMP  M          ; Check the decode entry
        JNC  BAUD_4     ; Not found yet
```

Figure 6-5. Am8080/8085 Auto Baud Rate Generator

```
        ; DE points to the required time constant


        LDAX D          ; Put time constant in HL
        MOV  L,A
        INX  D
        LDAX D
        MOV  H,A
        CALL INIT       ; Re-program both Am9513 and Am8251/Am9551
BAUD_5:
        ; Print a log-on prom . to indicate success

        IN   B_CTRL     ; Read console status
        ANI  TX_RDY
        JZ   BAUD_5     ; Wait till ready - good practice

        MVI  A,'>'      ; Send the prompt
        OUT  B_DATA
        RET

INIT:
        ; Reset and set Am9513 channel 1 to baud rate generator.
        ; Mode: TC toggle, Mode D, count bin/down, SRC 1
        ; Note: For Am9513 crystal use F1. SRC 1 used here for testing.
        ; Reset and set Am8251/Am9551 to usual async uart
        ;
        ; Entry: HL contains baud rate time constant

        RESET
        MODE_REG 1,TC_TOGGLE,DOWN,BINARY,MODE_DEF,SRC_1,RISE,NO GATE
        POINT    1,LOAD_
        MOV  A,L        ; Set up counter 1 LOAD register
        OUT  A_DATA
        MOV  A,H
        OUT  A_DATA     ; That was the upper byte
        LOAD 1          ; Transfer the value
        ARM  1          ; Turn on the baud rate generator

        ; Safe now to set up the Am8251/Am9551 uart

        MVI  A,Ø        ; Ensure no sync problems
        OUT  B_CTRL
        OUT  B_CTRL
        OUT  B_CTRL
        MVI  A,RESETU   ; Am8251/Am9551 software reset
        OUT  B_CTRL
        MVI  A,MODE     ; Set uart mode
        OUT  B_CTRL
        MVI  A,COMND    ; Set uart command register
        OUT  B_CTRL
        IN   B_DATA     ; Clear any rubbish around
        RET
```

Figure 6-5. Am8080/8085 Auto Baud Rate Generator (Cont.)

```
TABLE:
        ; Conversion table - used to calculate an index from the
        ; number of pseudo characters received by the system.
        ; Entries may be adjusted for convenience of use.

        DB      2       ; 19200 baud, nominally 1 char: ie match if < 2
        DB      4       ; 9600  nom 2
        DB      5       ; 4800  nom 4
        DB      7       ; 2400  nom 5
        DB      9       ; 1800  nom 8
        DB      13      ; 1200  nom 9
        DB      23      ; 600   nom 18
        DB      45      ; 300   nom 34
        DB      75      ; 150   nom 68
        DB      108     ; 110   nom 93
        DB      255     ; 75    nom >130

RATES:
        ; Baud rate constant conversion table. The indexed entry
        ; corresponds to the value required for the Am9513 LOAD
        ; register if a 2.4576MHz crystal is used. Notice the one to
        ; one correspondence between the entries in RATES and TABLE.
        ;
        ; Calculate new rates (or different crystals) by:
        ;
        ;       entry := ((crystal/32)/baud_rate)
        ;
        ; All values in Hz. The factor 32 comes from /16 mode in
        ; the Am8251/Am9551 uart and /2 from the Am9513 TC toggle mode.

        DW      4       ; 19200 baud
        DW      8       ; 9600
        DW      16      ; 4800
        DW      32      ; 2400
        DW      43      ; 1800
        DW      64      ; 1200
        DW      128     ; 600
        DW      256     ; 300
        DW      512     ; 150
        DW      698     ; 110
        DW      1024    ; 75

END BAUD

A>
```

Figure 6-5. Am8080/8085 Auto Baud Rate Generator (Cont.)

AmZ8000 SAMPLE LISTING

The Z8000 code sequence shown in Figure 6-6 performs the same action as the previous example with the additional facility of changing the Am8251/Am9551 Mode control word for differing baud rates. For example, at 9600 baud frequently 10 bits per byte are utilized compared to 11 bits per byte at 110 baud.

For both examples, successful decoding of the baud rate is indicated by the appearance of the prompt (")") on the users terminal.

Additional baud rates may be catered for by entering the appropriate data in the two data tables (TABLE and RATES). Reducing the number of baud rates available will allow a greater number of characters to give correct operation, although the program is designed to receive a carriage return. As listed, the 11 standard baud rates between 75 and 19,200 baud are catered for.

```
%       This file allows auto baud-rate sensing of an attached terminal
% by looking at the data returned from a user entered carriage return.
% It assumes the use of an Am9513, Am8251/Am9551 combination.

CONST
A_DATA  =       0FFD8H, % Am9513 data port
A_CTRL  =       0FFDAH, % Am9513 command port
B_DATA  =       0FFDCH, % Am8251/Am9551 data port
B_CTRL  =       0FFDDH, % Am8251/Am9551 command port
RX_RDY  =       1,      % Am8251/Am9551 rx ready bit
TX_RDY  =       0,      % Am8251/Am9551 tx ready bit
MAXTIM  =       5000,   % Time for worst-case reception cr, 75 baud
RESETU  =       040H,   % Official Am8251/Am9551 software reset
MODE_1  =       0CEH,   % 2 stop, /16, no parity, 8 data bits
MODE_2  =       0DFH,   % 2 stop, /16, odd parity, 8 data bits
COMND   =       037H,   % tx/rx enable, dtr, rts
INDEX   =       R1,     % Common index for TABLE and RATES
TIMER   =       R2,     % Timer for worst case reception
COUNT   =       RL3;    % Input psuedo character counter

INCLUDE 'B:EQUSZ.ZSC'     ; % Listed in Appendix 7
INCLUDE 'B:Z8000MAC.ZSC'  ; % Listed in Appendix 5

BAUD:
        % The system monitor calls 'BAUD' to set up console io

        CLR  INDEX          ; % Start with max rate known to program
        CALR INIT           ; % Set up Am9513, Am8251/Am9551
BAUD_1:
        IN   R0,B_CTRL      ; % Read console status
        BIT  R0,RX_RDY      ;
        JR   ZR,BAUD_1      ; % Wait for 1st character

        CLRB COUNT          ; % Zero character counter
        LD   TIMER,MAXTIM   ; % Initialise timer
BAUD_2:
        DEC  TIMER,1        ; % Move and test timer status
        JR   ZR,BAUD_3      ; % Timeout, go decode data

        IN   R0,B_CTRL      ; % Read console status
        BIT  R0,RX_RDY      ;
        JR   ZR,BAUD_2      ; % Wait for a character

        IN   R0,B_DATA      ; % Discard pseudo character
        INCB COUNT,1        ; % Adjust character counter
        JR   BAUD_2         ; % Repeat
BAUD_3:
          ; % Perform serial index search for reqd baud rate const
          ; % Entry:  COUNT holds # pseudo characters received

        LD   INDEX,-2       ; % Word index
BAUD_4:
        INC  INDEX,2        ; % Move the index
        CPB  COUNT,TABLE(INDEX) ; % Check the decode entry
        JR   NC,BAUD_4      ; % Not found yet
```

Figure 6-6. Z8000 Auto Baud Rate Generator

```
        % INDEX is offset for both baud rate constant and Am8251/Am9551
        % commend byte

        CALR INIT           ; % Re-program both Am9513 and Am8251/Am9551
BAUD_5:
        % Print a log-on prompt to indicate success

        IN   R0,B_CTRL      ; % Read console status
        BIT  R0,TX_RDY      ;
        JR   ZR,BAUD_5      ; % Wait till reedy - good practice

        LDB  RL0,'>'        ; % Send the prompt
        OUT  B_DATA,R0      ;
        RET                 ;


INIT:
        % Reset and set Am9513 channel 1 to baud rate generator.
        % Mode: TC toggle, Mode D, count bin/down, SRC 1
        % Note: For Am9513 crystal use F1. SRC 1 used here for testing.
        % Reset and set Am8251/Am9551 to usual async uart
        %
        % Entry: INDEX contains baud rate time constant index


        RESET               ; % Not neccessary if the Am9513 already set
        MODE_REG  1,TC_TOGGLE,DOWN,BINARY,MODE_DEF,SRC_1,RISE,NO_GATE;
        PCINT 1,LOAD_       ;
        LD   R0,RATES(INDEX) ; % Get rate constant
        OUT  A_DATA,R0      ;
        LOAD 1              ;
        ARM  1              ;

        % Safe now to set up the Am8251/Am9551 uart

        CLR  R0             ; % Ensure no sync problems
        OUT  B_CTRL,R0      ;
        OUT  B_CTRL,R0      ;
        OUT  B_CTRL,R0      ;
        LD   R0,RESETU      ; % Am8251/Am9551 software reset
        OUT  B_CTRL,R0      ;
        LD   R0,TABLE(INDEX) ; % Set uart mode
        OUT  B_CTRL,R0      ;
        LD   R0,COMND    ; % Set uart command register
        OUT  B_CTRL,R0      ;
        IN   R0,B_DATA      ; % Clear any rubbish around
        RET                 ;
```

Figure 6-6. 28000 Auto Baud Rate Generator (Cont.)

```
TABLE:
        % Conversion table - used to calculate an index from the
        % number of pseudo characters received by the system.
        % Notice that a different mode byte may be defined for
        % each separate baud rate - 110 baud should use 11 bits.


        BYTE: 2,   MODE_1       ; % 19200 baud, nominally 1 char.
        BYTE: 4,   MODE_1       ; % 9600  nom 2              Match < 2
        BYTE: 5,   MODE_1       ; % 4800  nom 4
        BYTE: 7,   MODE_1       ; % 2400  nom 5
        BYTE: 9,   MODE_1       ; % 1800  nom 6
        BYTE: 13,  MODE_1       ; % 1200  nom 9
        BYTE: 23,  MODE_1       ; % 600   nom 18
        BYTE: 45,  MODE_1       ; % 300   nom 34
        BYTE: 75,  MODE_2       ; % 150   nom 68
        BYTE: 108, MODE_2       ; % 110   nom 93
        BYTE: 255, MODE_2       ; % 75    nom >130


RATES:
        % Baud rate constant conversion table. The indexed entry
        % corresponds to the value required for the Am9513 LOAD
        % register if a 2.4576MHz crystal is used. Notice the one to
        % one correspondence between the entries in RATES and TABLE.
        %
        % Calculate new rates (or different crystals) by:
        %
        %       entry := ((crystal/32)/baud_rate)
        %
        % All values in Hz. The factor 32 comes from /16 mode in
        % the Am8251/Am9551 uart and /2 from the Am9513 TC toggle mode.


        WORD:    4      ; % 19200 baud
        WORD:    8      ; % 9600
        WORD:    16     ; % 4800
        WORD:    32     ; % 2400
        WORD:    43     ; % 1800
        WORD:    64     ; % 1200
        WORD:    128    ; % 600
        WORD:    256    ; % 300
        WORD:    512    ; % 150
        WORD:    698    ; % 110
        WORD:    1024   ; % 75

END.


A>
```

Figure 6-6. Z8000 Auto Baud Rate Generator (Cont.)