FILE

I.O.S.

INTERNAL DOCUMENT 176

PRECISION ECHO-SOUNDER TRACKER

C.G. Flewellen

NATURAL ENVIRONMENT RESEARCH COUNCIL

INSTITUTE OF OCEANOGRAPHIC SCIENCES

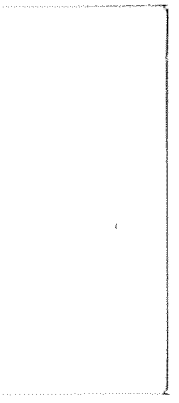
INSTITUTE OF OCEANOGRAPHIC SCIENCES

Wormley, Godalming,
Surrey GU8 5UB
(042-879-4141)

(Director: Dr. A. S. Laughton, FRS)

Bidston Observatory,
Birkenhead,
Merseyside L43 7RA
(051-653-8633)
(Assistant Director: Dr. D. E. Cartwright)

Crossway,
Taunton,
Somerset TA1 2DW
(0823-86211)
(Assistant Director: M. J. Tucker)



PRECISION ECHO-SOUNDER TRACKER

C.G. Flewellen

INTERNAL DOCUMENT NO. 176

1983

Institute of Oceanographic Sciences,
Wormley, Godalming, Surrey GU8 5UB.

PRECISION ECHO-SOUNDER TRACKER

INTRODUCTION

This is a micro-processor depth digitiser and bottom tracking system. The aim of the tracker is to produce a smoothed estimate of the depth including phase information every two minutes and to internally save (as long as the mains is on) up to 34 hours worth of depths in metres with their associated times. A remote entry terminal allows the watch-keeper to provide a starting time and at any time to preset the tracker to a new depth should it have lost the bottom signal. Marker pulses sent to the P.E.S. recorder show the start and end of the tracker's sampling window and should hover about the echo. In deep water when several trans-mission pulses are in the water-path it will still be necessary for the watch-keeper to turn gating on and off otherwise the tracker is sure to lose the bottom in the transmission reverberation. However, with gating on the computer will recognise any sweep containing a transmission pulse and not use this information for tracking, i.e. the sampling window will not be moved. On other sweeps in the gating cycle the echo can be followed through the transmission.

OPERATION

Switch on the remote keyboard and check that it is plugged in to PEST. If it is not connected PEST will expect to get its initial time from a terminal connected to the external serial port and will wait for this. If the remote keyboard is connected, switch on the printer and PEST and the program will start executing with an assumed time of 00.00 Day 1 1983. Enter the time as four digits of hours and minutes followed by the day number as three digits. An incorrect entry will cause the error pattern to be displayed for a few seconds. Try again. A correct entry will be redisplayed with a space inserted between time and day number. After a few seconds this will disappear as it is transmitted to PEST. To get

the time correctly synchronised press ENTER TIME on the minute as entry causes the seconds to be cleared. While waiting for the minute the entry display will time-out to be replaced by the regular depth transmission from PEST. The entry can be recalled with the RCL key or cleared with the C/E key. Now a starting depth can be entered. This may be from 1 to 4 digits but not zero nor must it exceed 8999. A correct entry will cause the display to shift to the left hand side with leading zeroes added if necessary. This will soon disappear as it is transmitted.

A new depth can be entered at any time should PEST lose track. An indication of the success of tracking is provided by a pair of lines that should bracket the echo on the echo-sounder. These lines will be present at switch-on but can be removed by pressing the red button on PEST. Successive presses turns them on and off. In addition the time will be printed out on the Mufax every six minutes and the smoothed depth which is transmitted to the ship's logging system will be displayed every two minutes below the two-minute mark it corresponds to.

PEST, as yet, has no control of the gating of the echo-sounder but does need to know when gating is on. At least one Mufax has been modified to provide an external contact closure in parallel with the gating switch.

PRINCIPLE OF OPERATION

The zero-depth pulse from the echo-sounder starts a count-down to a sampling window. The audio output from the recorder is then digitised and stored till the end of the window is reached. The stored signal is examined for the strongest leading edge of the bottom signal with a bias towards finding it near the centre of the window. The count-down delay for the next sweep is then adjusted to centre the window more nearly on the strongest echo. This adjustment is filtered to avoid an erratic response to a noisy signal. Every sweep a new depth value is

computed from the position of the centre of the window and these are stored away so that on the odd minute, two minutes worth of depths centred on the previous even minute can be filtered. It is this value which is displayed on the recorder and with its associated time is transmitted to the ship's computer.

PROGRAM DESCRIPTION

The main program starts by calling a subroutine that initialises parallel and serial ports and sets up a counter/timer to provide the 1 kHz sampling frequency and a one second pulse for the internal clock. Various locations in RAM are then cleared including the error flags. A number of other flags are initially set and the interrupt mask pattern is preset to allow sweep trigger and one second pulse interrupts. With all interrupts disabled a request is made for the time to be entered from the control terminal. However, if the remote keyboard is plugged in and running the program will start executing immediately with a default time and depth. A whole set of constant parameters are then transferred to the working locations and then the internal interrupt flag is cleared. Sampling of the signal is achieved through interrupts so the main program need only loop round until it detects that the sampling done flag has been set. In this loop error messages if any are printed, the spot depth is displayed and transferred to the remote terminal. The front panel switches and the remote depth entry are checked for any change and the time is regularly checked. Every even minute the spot depth and time are printed and on the hour a heading as well. Each spot depth is stored in an array which on the odd minute is applied to a filter which smooths two minutes worth of depths centred on the even minute. The filter output is then transferred to another array where the time is attached to it. An attempt is made to send a batch of recent depths to the ship's logger and if unsuccessful a note is made that the depths are pending and an attempt will be made to send them next time. Up to 34 hours may be backed up like this.

SUBROUTINE DESCRIPTIONS

INIT

Program the parallel and serial ports and the interval timer. Set up clock constants and variables.

Three 8+8 bit parallel I/O ports are programmed to define data directions and the use to be made of their control lines.

(1) ADC PIA at $8004 all 16 bits are defined as I/Ps. Twelve from the ADC and four as spare logic I/Ps. No control functions are selected at this time.

(2) DAC PIA at $8008 fourteen bits are made O/Ps. Ten to the DAC and four to the edge connector to drive LEDs. CA2 is programmed to interrupt on a negative transition every 1 second to advance the internal software clock. CB1 is programmed to interrupt on a positive-going start of sweep trigger.

(3) The A side of the PIA at $8020 is later programmed to provide BCD communication among local LCD display and thumbwheel switches.

The B side of this PIA is set to provide 7 bits out to the Anadex printer and 1 bit (the most significant) in to indicate the readiness of the printer.

CA2 is programmed as an O/P generating a low strobe pulse as data is written to the printer.

(4) A triple counter/timer appears at $8040. Only one of these is actually needed to delay before sampling the echo, but use is made of the other two sections:-

Counter 0 divides by 1000 from the CPU clock at 1 MHz.

Counter 1 counts down from counter 0 O/P to provide 1 second for the clock.

Counter 2 counts down using the 1 kHz from counter 0 to generate an end of delay interrupt.

(5) The control terminal (during development) uses a serial interface at $9002. During use this interfaces with the ship's computer system.

(6) Initial time information is copied from ROM to RAM including a list of limit values, i.e. 60 for seconds and minutes, 24 for hours, etc.

Communications

There are a number of I/P and O/P routines to cope with the control terminal, the parallel printer, front panel display and local depth entry switches and a serial (RS232) bus to communicate with a remote entry terminal.

These are organised as three ports in order to share various I/O processing routines such as the output of HEX or BCD, fetching a character with a time-out and ASCII string printing.

Port 0 uses the original serial port of the utility routine and uses subroutines in that routine.

Port 1 is an extra serial port for communicating with the remote keyboard.

Port 2 is a parallel output port to drive the Anadex printer.

INPRT

Calls INCH or ZINCH depending on the port number specified by PORTNO.

OUTPRT

Calls OUTCH, ZOUTCH or PRINTC depending on the port number.

GETPRT

Calls GET or ZGET.

PRINTC

X Reg is saved then loaded with a value that gives about 1/2 sec delay in a wait loop if the printer is not ready. If it is the character is sent. A C/R is followed by a L/F to cause the printer to print its buffer contents. The X Reg is then recovered and the routine returns with the character sent in Acc. A.

PRINTS

The character string pointed to by the X Register is sent to the port until the terminating byte 04 is reached.

ZOUTHL etc.

A set of routines to print left or right half of a byte in HEX and

hence a whole byte, whole byte and space or a double byte and space. These routines will also work with packed BCD digits.

ZINCH

Input a character from port 1. The parity bit is masked off and the program drops through to echo the character.

ZOUTCH

Output a character to port 1.

GET

Port 0 is examined for waiting character but if none only about two character times (at 2400 baud) are wasted before the subroutine returns with a null character.

ZGET

Does the same as GET but for port 1.

CLOCK

The interrupt service will call this subroutine every second. Before the time is advanced and if the previous seconds were even the new depth value is stored on a rotating stack starting at $1000. After storing a new depth in two bytes the pointer is advanced but if it goes beyond the end of the stack it is reset to the start so that only the most recent two minutes worth of depths is retained.

The time is advanced one second. If the seconds reach their limit of 60 the seconds are cleared and the routine advances round the loop and increments the minutes. If they reached their limit then ... etc.

On January 1st the days would be reset so zero day number is changed to 1.

CVBTD

Converts a double byte of binary in the X Register into a terminated string of ASCII digits. Starting with 10,000 successive powers of ten are repeatedly subtracted from the binary number. The number of times each power GOES is counted then converted into an ASCII character which

is added to the string result. The string is terminated with 04 and the X Reg. is pointed to its beginning ready for a call to print.

INDEC

Fetches four decimal digits via INDIG and packs them into the two bytes DIG and DIG+1. A non-digit value causes a return. If more than 4 digits are entered only the last 4 will be retained.

INDIG

Fetches a character from the input port. If it is a digit it is converted to binary and the carry bit is cleared. A character other than a digit will cause the carry to be set.

TWOMIN

This routine is called regularly whenever the processor has spare time and between interrupts. The depth and time are printed at the beginning of the even minute. The EVNMIN flag set initially and during the odd minute allows the routine to test the minutes for even. As soon as the minutes become even the depth and time are printed but then EVNMIN is cleared, preventing further printing until minutes have been odd.

PRDAT

Print two minute data. First a check is made for zero minutes HRHEAD then hours, minutes and seconds are printed. The depth (internally in msecs) is scaled to metres then converted into a decimal/ASCII string and printed followed by the estimated phase and an optional signed slope value.

HRHEAD

The flag HRFLG plays the same role here as EVNMIN preventing repeated printing while the minutes are still zero.

HEADING

If the minutes have just become zero an hourly heading is printed followed by the day no. and year and column headings.

ERRS

There are seven RAM locations which can be set to signal up to seven error conditions. Pointers are set to the beginning of these error codes and to a list of error messages. The routine examines each code in turn for a non-zero value when it prints the error message and clears the error code. It is possible for all seven errors to occur at the same time.

GETTM

Get the time from the control terminal. After printing the request for the time 4 groups of decimal digits separated by any non-digit character (except ESC) are copied into the clock variables in the order hours, minutes, day number and year. A mistake can be corrected with ESC which returns to the beginning of the routine. On completion the thank you message is sent.

ADDX

Add DATSIZ, the width of the sampling window, to the index register.

ADDA

Add Acc. A to the index register. The two's complement number is extended into Acc. B.

INTERRUPT SERVICE ROUTINE

All peripheral interrupts cause the program to be vectored to here. When a transition occurs on a peripheral control line an internal flag is set regardless of whether the device was enabled to interrupt. It is thus necessary to keep a flag byte up-to-date to identify which device is enabled to request an interrupt. As each of 4 possible devices is interrupt enabled its flag bit is set. When the device is disabled the flag bit is cleared. These flags are O/P to the edge connector each time an interrupt occurs for scope monitoring.

On an interrupt each of the peripheral internal flags is shifted into Acc. A which is then anded with the flag byte so that only enabled interrupts are acknowledged. On a trigger, ADC end-of-conversion or

end of delay interrupt the appropriate routine is executed and then the peripheral flags are tested again in case either another interrupt has occurred during the servicing or in the event that two interrupts had been simultaneous. Every trigger pulse the clock timer is reloaded to count 500 msecs. This keeps the internal clock synchronised with the trigger rate and phases the 1 second pulses so they occur a quarter and three quarters of the way across the echo-sounder sweep.

A dot-matrix display on the recorder is initiated when required on the one second interrupt, either on the left or right hand side of the paper depending on the position of the bottom echo. The clock timer is reloaded with 1000 msecs and the clock subroutine is called to advance the time.

If no flags are set a return from interrupt is executed. Since interruption can only occur if the interrupt mask is cleared and this is automatically set when the interrupt occurs there is no danger of the service routine being interrupted itself.

TRIG

On receipt of a trigger TRGFLG is set. This can be cleared in the main program and tested if synchronism is required. If the delay is counting down when the trigger occurs the error EXTRA TRIGGER is flagged and the timer is stopped and its interrupt capacity disabled. A count is kept of the number of false triggers that occur consecutively. If the delay was not active the false trigger count is cleared.

If the sampling done flag is cleared and the delay value is non-zero the delay timer is loaded, its interrupt enabled and then started. Had the delay been zero then the sample count DATCNT is initialised from DATSIZ and sampling interrupts enabled. In order to mark the position of the tracking window on the recorder the Mufax mark bit is set low. (Negative logic is used).

TIM

When the count-down finishes sampling is initialised as above when the delay value was zero.

Note that the delay value is a program variable not the instantaneous counter value. In fact the counter is never read.

SAM

The program arrives here when sampling is enabled and an ADC conversion has just been completed. However, the sampling done flag must have been cleared, if not then sampling is disabled - likewise if the number of samples required happens to be zero.

Setting the Mufax mark bit clears the pulse sent to the recorder. Thus the start of the tracking window has been indicated.

Should the display mode flag be set data is not sampled but instead subroutine RASTAS is called to output one dot of a depth or time display on the recorder. When the display mode flag is clear a data sample is taken from the ADC.

When after subsequent calls to SAM the sampling is finished the number of signal overloads or the lack of them is used to calculate the binary gain ranging value to be applied during the next sweep. This acts as a course AGC reducing 12 bits from the ADC to 8 bits by left shifting up to 4 times. TVG is applied prior to this to eliminate the surface reverberation and filter ringing immediately after the transmission. When DATCNT is 2 indicating that this is the penultimate sample the Mufax mark is turned on. This is turned off again one sample later to mark the end of the sampling window on the recorder.

ADJUST

The delay value having been adjusted to shift the sampling window to track the echo, half the window width is then added to give the position of the echo (in msecs) from the beginning of the same sweep. Multiples of 2000 msecs are added for each phase value greater than 1. Phase 1 corresponding conventionally to 0-1500 metres. A phase of less than 1 or greater than 6 generates the error PHASE OUT OF RANGE and no adjustment is made. Finally an offset of 8 samples is added because in the course of

edge detection the data was shifted ahead this amount. The result is the absolute depth but expressed in msecs.

EDGE

This is the edge detecting routine. The algorithm is a correlation of 2N + 1 (N = 8) samples with a ramp that runs from -8 to +8. To speed it up the process is made recursive. To prepare for running the algorithm the first 17 samples are added up and the gradient is set to zero. The oldest sample can be considered to have been multiplied by -8 thus adding 8 times the oldest cancels this out. Adding 9 times the new sample will achieve the effect of multiplying by a ramp running from -7 to +9. Subtracting the running sum gets the desired result. It is now necessary to update the sum by subtracting the oldest and adding the newest samples. The gradient is scaled up by eight but if negative is set to zero. This is stored back at the beginning of the array giving it a -8 sample offset. To avoid problems at the end of the array the correlation stops 17 samples from the end and the rest of the array is cleared.

TVG

Applies Time-Varied-Gain at the beginning of the sweep to remove the transmission reverberation. If the gating on flag is set application of TVG is skipped. It is first necessary to find where the sample is in relation to the start of sweep. DATCNT is a decreasing sample counter and is thus a measure of how far from the end of the window the sample is. Subtracting this from DATSIZ the window width and adding the delay gives the required sample position. If this results in a value greater than 2000, 2000 is subtracted. If the sample is within the first 72 it is set to zero. Within the range 72 to 200 (128 samples) the echo is ramped up by multiplying by the fraction - twice sample position from 72 divided by 256. This has the effect of ramping it up linearly. When the edge detection algorithm is applied it is not likely to see a significant edge in this

region due to the gentle rise. Turning off TVG in the gating mode will allow the tracker to follow an echo through the transmission provided it only processes on sweeps when transmission has been gated off.

PEAK

This routine looks for the highest sample in an array that has been passed through the edge detecter and thus consists of rounded peaks. It is only necessary to find that sample which is greater than its previous neighbour and greater than or equal to the next sample. As each local peak is found it is weighted according to its distance from the centre of the window. The weighting function is one in the middle decreasing linearly to zero at either end. This local peak is compared with any previous one recorded and displaces it if it is bigger. At the same time the offset of the peak from a previously set prediction is compared with any earlier offset and displaces it if the offset is smaller. When the routine has run its course we have two peak values and their positions - the largest overall after weighting and that one (often the same) which is nearest to the prediction. The prediction is normally the centre of the window but could be shifted either way to provide a bias when the bottom is sloping.

WEIGHT

Weight a peak value, one in the centre to zero at the ends of the sampling window. As the window width may be altered weight must work with whatever width exists. (Unlike TVG which using binary tricks to simplify the weighting operation). If the peak position is before the centre then that position value can be used in the weighting. If it is after then its position in the window is subtracted from the window size. The weighting algorithm then multiplies the peak value by position from ends of window and divides by the window width. The operands for the division are scaled down if necessary to suit the division subroutine.

SHFTGT

Adjusts the delay to move the sampling window. Given the processed echo position relative to the start of the window subtracting half the window size provides a signed offset from the centre of the window. Adding this to the present delay provides a new delay (which may go out of range). A negative result i.e. the delay moves back before the transmission is corrected by adding 2000 and the phase value is decreased by one. A value of delay greater than 1999 has 2000 subtracted while the phase is advanced by one. If the delay jumps like this subsequent filtering would cause a large transient so the adjusted delay value is used to "pre-charge" the filter accumulator - AVDPTH.

CVDTB

Convert from decimal to binary. Four decimal digits are converted to a double byte of binary, starting with the most significant digit 1000 in binary is added to a results accumulator while the digit is decremented to zero. The process is repeated for the remaining digits with 100, 10, and 1s being accumulated. The result is returned in Acc.s B-A.

CVRTMS

This routine converts manually entered absolute depths in metres into a sample count in msecs. from the start of sweep plus a phase value. A number of multiples of 1500 are subtracted until the result is less than 1500. The number of times less one that 1500 goes is the phase. The remainder is scaled by 4/3 to convert from metres to msecs. and returned in the index register.

Multiplication and Division

The hardware multiplier has internal registers for data, control and results and appears in the memory between $FC and $FF. The multiplier does an unsigned multiply of two single bytes to produce a double byte product. The divider divides a double unsigned byte by a single byte and

generates one byte of quotient and the remainder. The chip is designed to operate as one of up to four to give high resolution results but can be used alone.

Arguments are passed to these arithmetic routines by pushing them onto the stack before calling the subroutine and then using the index register as a pointer to the operands then the results.

MPYDBL

Double byte times single using the internal accumulation. The result will run to three bytes but the least significant is not returned.

DIVDBL

Divides two bytes by a single byte and returns the quotient and remainder. If an overflow occurs the carry bit is set.

MPY1X3

Multiplies three bytes by a one byte multiplier and returns a four byte product. Use is made of the internal accumulation.

MPY2X2

Two by two multiplier.

FILT

A single pole recursive filter to smooth double byte delay values. The algorithm being:-

$$Y(N + 1) = (X(N)*COEF1 + Y(N))*COEF2$$

the coefficients being trimmed to produce zero DC offset to avoid producing any drift. Y(N) or AVDPTH in the subroutine is preset to a new value when the delay is changed discontinuously to prevent a serious transient being generated.

INBCD

Gets four digits from the front-panel thumbwheel switches and stores them unpacked in BCDIN. The switches are scanned starting from the least significant end and since negative logic is used on the BCD bus for incoming data, the BCD values must be complemented.

LCD

Displays up to eight digits on the front panel. The high 4 bits of the byte written to the interface select the digit position while the low 4 bits contain the BCD value. After the byte is written an all ones byte is sent to latch the data into the display latch/decoders.

INDPTH

Get a depth (in metres) from the front panel. This routine is called regularly in a loop after processing has been completed and during the wait for data to be sampled and so will respond to any depth entry. However, instead of having the program pause in an entry wait loop it fetches the depth by calling subroutine INBCD and then compares this depth with the previous saved value. Only if the value is different is this value saved and further processed. A call to CVDTB converts the four digits of unpacked BCD into a double byte of binary. CVRTMS changes this into msecs. into the sweep plus the phase value. Subtracting from the echo position the half-width of the sampling window gives us the delay value required to centre the window on this depth. This subtraction may put the delay value out of bounds and so it is tested for negative or greater than one sweep. The result is the new delay and is also stored in the delay filter variable AVDPTH to prevent a transient.

DISP1

The most recent depth NWDPTH is displayed. NWDPTH is the absolute (including phase) depth but in msecs. It is converted to metres by multiplying by three quarters, converted into decimal digits CVBTD then sent to the display routine LCD.

DPFILT

Depths are stored every 2 seconds in a rotating store holding just over two minutes worth of data. On the odd minute these must be smoothed to provide a good estimate of the depth on the two minute mark. The filter is a correlator with a reference wave-form that is a leger-weighted sinc. function and thus gives low-pass characteristic.

With the coefficients chosen the cut-off frequency is a 0.1 Hz.

As the coefficients are symmetric about their centre only half the array is needed. Pointers are set to the present oldest depth in the rotating depth stack and to the beginning of the coefficient array. As both coefficients and depths may be negative a sign flag preserves the sign so an unsigned multiply can be performed. (A negative depth could occur if in a later development it was decided to subtract a mean depth from the whole array before filtering, thus improving the arithmetic resolution). As each multiplication is completed, the sign information is used to add the product to or subtract it from the accumulating answer. After the middle coefficient has been used the pointer to the coefficients reverses direction. When the end of the depth store is reached the corresponding pointer is reset to the start.

STORDP

Every odd minute stores the filtered depth in a new array. After the depth has been converted to metres in packed BCD it is stored at the next pair of locations on the depth and time stack. Next come the year, day number, hour and the least significant bit of the minutes is initially set to indicate that the depth has not yet been sent to the ship's logger. The stack pointer is then advanced eight places until, after about 34 hours, the stack pointer is reset to the start of the stack and old depths start getting over-written. If the Mufax display is enabled MFINIT is called to set up for a display and the depth digits are transferred to the Mufax display store.

SRCHDP

After the previous routine the stack pointer will be pointing to the next available set of locations on the stack which will either be clear (the whole array was cleared after the power came on) or contain the oldest depth still in residence. From here and right round to the most recent depth entry the array is searched for any block with the least

significant bit of the minutes set. This depth and time are sent to the ship's logger. (The minutes are made even for this purpose). If the depth is acknowledged the least significant bit of the minutes is cleared and the search continues. If the depth has not been received the bit is left set and the routine returns to try again later.

XMIT

An attempt is made to send a two minute depth and time to the ship's logger. The carry bit is cleared if this is successful.

PROCSS

Process the newly acquired echo. Edge is called to differentiate the echo signal and remove trailing waveform edges. Peak identifies the prominent peak from the edge routine. The peak (may not be the biggest) that is nearest to a prediction is also returned. A decision is made to choose one or the other (at present the largest is chosen). The position of this peak is then used as the input to SHFTGT to move the sampling window to centre it over the echo. The resulting delay value is smoothed by FILT before becoming the new delay value to be used in the next sampling session. Subroutine ADJUST converts the new depth into an absolute depth though still expressed in msecs.

GATEST

Tests whether, when the gating is on, the present sweep contains a transmission pulse. The position of the sampling window is tested to see if it overlaps the start of sweep. If not or if gating is off, the carry bit is set. Otherwise samples from the data array are accumulated from the start of sweep position up to a maximum of twenty samples or to the end of the array. DIVDBL is called to convert the sum to an average. When this is compared with a pre-set threshold the carry will be set if it is less otherwise a clear carry will indicate that a transmission pulse has been found on this sweep.

RASTAS

An ASCII string terminated by a carriage return is converted dot by dot and line by line into a display on the recorder. The dot look-up table caters for the character set:- 0123456789<=>:;? though a blank is generated for "?". Two parameters KWIDTH and KLINES allow the number of dots per pixel to be chosen and the aspect ratio adjusted.

SEND

Transmits the depth to the remote keyboard each time a new one is produced. Port 1 is tested for CLEAR-TO-SEND and if it is not ready an error message will be printed. To prevent continuous printing of errors when the keyboard is not connected an error count of 256 occurs before the message is printed again. If the keyboard is available and ready then 4 digits of depth are transmitted followed by two spaces. (Actually "?" is sent as the keyboard will decode this as a blank). Two digits of seconds are then sent mainly to provide an indication that the keyboard is active. A carriage return is sent to terminate the sequence. A response to this is expected within two character times otherwise error 6 is flagged. If the response is a carriage return then the hand-shaking is complete. The receipt of a "D" or a "T" indicates that the keyboard has a new depth or time to send. The four digits of depth received are then treated as though their input had been local. The digits of time are used to preset the software clock whose seconds are cleared. A carriage return terminator is expected otherwise the error RX ERROR is flagged.

HARDWARE DESCRIPTION

The Motorola 6800 system comprises three boards.

(1) 16K of dynamic memory configured to occupy the 1st quarter of the address space.

(2) The processor board.

(3) An I/O board.

Additionally the front panel display is mounted on a board containing latches and interfacing.

(1) The memory is a standard block of dynamic RAM that is refreshed by on-board circuitry by stealing the occasional clock cycle.

(2) The 6800 CPU has 16 address lines, 8 data lines and a hand-full of control lines that are buffered on the CPU board. The 1 MHz clock provides two clock phases to drive the CPU and can be momentarily stopped to allow for memory refreshing. A one-of-eight decoder generates negative logic select signals from the top 4 address lines:-

$8000 Selects the I/O board

$9000 Selects the serial port used to transmit to the ship's logger

$A000 Selects 128 bytes of scratch-pad memory used by the monitor program

$B000 Is not used

$C000 The non-volatile RAM containing PEST firm-ware

$D000 Is not used

$E000 Selects the serial port used by the remote keyboard

$F000 Selects an EPROM holding the monitor firm-ware

$000-$3FFF is where the RAM board appears but with one exception, the multiplier/divider is addressed by $00FC-$00FF.

A 614.4 kHz crystal oscillator and binary divider chain provides a number of baud-rates for the serial interfaces. The I/Ps and O/Ps of the serial interfaces are buffered to meet RS232C line requirements.

(3) The I/O board comprises three parallel interface adapters (PIAS), a triple counter/timer and analogue circuitry. These are all memory-mapped

devices and appear at addresses as follows:-

$8004-$8007 The ADC interface with 12 lines in from the ADC

$8008-$800B The DAC interface with 10 lines out to the DAC

$8020-$8023 This provides a BCD bus to the display and the local depth entry switches

The printer uses one half of this PIA

$8040-$8043 Comprises three counter data registers and a common control register for the counter/timer

1 kHz from counter 0 (dividing by 1000 from the CPU clock) provides the convert trigger for the ADC, though provision is made for an external sampling clock. This division is reset every trigger pulse.

The signal from the echo-sounder is a base-band signal and is thus full-wave rectified before being applied to a 300 Hz four pole anti-aliasing filter. A sample and hold circuit is triggered by the 1 kHz ADC clock to hold the signal during digitising.

Tenth-inch pitch edge connectors connect to BNCs for analogue signals, a loom to a 15 way socket for the printer and a loom to the front panel.

These boards are fed from two 5 volt power cards, one of which also provides + and - 15 volts for the analogue circuitry.

REMOTE TERMINAL

This is an intelligent keyboard which communicates with the main processor via an RS232C bus. A Motorola 6803 CPU scans a 16 key keyboard, refreshes an eight digit LED display, receives depths from the tracker and can transmit back a new depth or time.

A remote terminal is used to satisfy three requirements:-

(1) So as not to tie up a typewriter or VDU to communicate with the main processor and for the convenience of the watch-keeper. The main processor can be mounted on a 19 inch rack out of the way.

(2) To transmit depths and times to the ship's logger.

REMOTE KEYBOARD FOR P.E.S.T.

Program Description

A power-on reset starts the program running at INIT which programs the serial and parallel ports. The serial interface is enabled to interrupt on receipt of a character. The internal (to the CPU) parallel port is programmed to provide eight bits out - seven to drive 8 LED displays.

Two registers and a buffer are maintained by the program and these are initially clear by filling them with hexidecimal ØF. With the interrupt mask cleared the program enters a loop in subroutine NRMDSP only exitted when a key is pressed (and regularly left to service interrupts). The contents of DATA1 holding a depth, transmitted from PEST, are displayed continuously on the LEDs.

When a digit key is pressed the digit value is decoded from a table (to adapt the key layout to a calculator style) and left shifted into the entry register DATA2. After each digit DATA2 is displayed for several seconds until either another key is pressed or the time is up, in which case we return to the regular display of the contents of DATA1. If the key were not a digit but hexidecimal, A, B, E or F routine COMMND decodes the command and jumps to an appropriate routine.

While this is proceeding, characters sent from the tracker on the serial line cause interrupts and these characters are stored away in a buffer. Line-feed codes are ignored as are carriage-returns when the buffer is still empty. When a carriage-return terminates a string of digits or the buffer becomes full anyway the up to eight ASCII characters in the buffer have their four most significant bits cleared before they are copied into DATA1 to be regularly displayed on return from interrupt. If either the depth entered flag DPENFG or the time entered flag TMENFG are non-zero then the contents of DATA2, the entry register, are converted to ASCII and transmitted back to PEST. The depth comprises 4 digits preceded by D and the time string is 7 digits long (with included space) and is preceded by T.

After either string a carriage-return is sent as would be the case if neither flag had been set.

Command A

Enter depth.

The subroutine DPFMAT is called to check the format of the contents of DATA2. If the depth is zero or greater than 9999 or if the entry register were filled with blanks an error pattern is displayed for a few seconds, the entry register is cleared and a normal display resumed. A correct entry causes the depth to be shifted to the left-hand end of the display with leading zeroes added as necessary. The flag DPENFG is then set to signal to the interrupt service that a depth is waiting. The entry register is displayed until transmission of the contents causes it to be blanked. The program then jumps back to continue the normal display.

Command B

Enter time.

The format is checked for legal time and a day number in the range 1-365. The flag TMENFG is then set. The procedure is then the same as for Command A.

Command E

Recall entry.

This results in a return to the entry register display loop with the time-out count reset.

Command F

Clear entry.

DATA2 is filled with blanks and this is displayed for a brief time, i.e. the display is seen to go blank for half a second before the normal display loop is entered.

Hardware Description

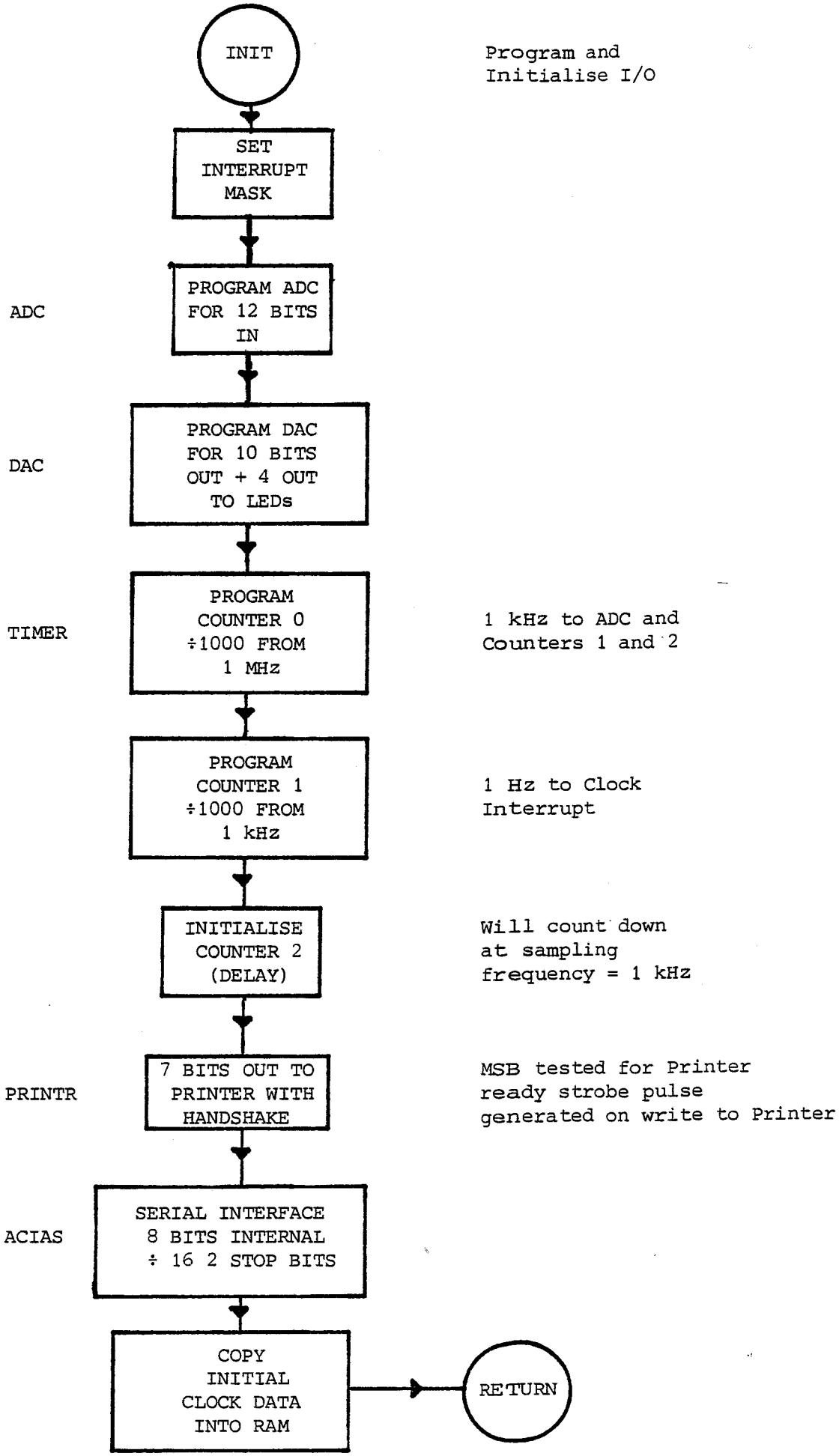
The central processor used is the Motorola MC6803, a version of the 6800 that is intended to be used with a minimum of extra components. There

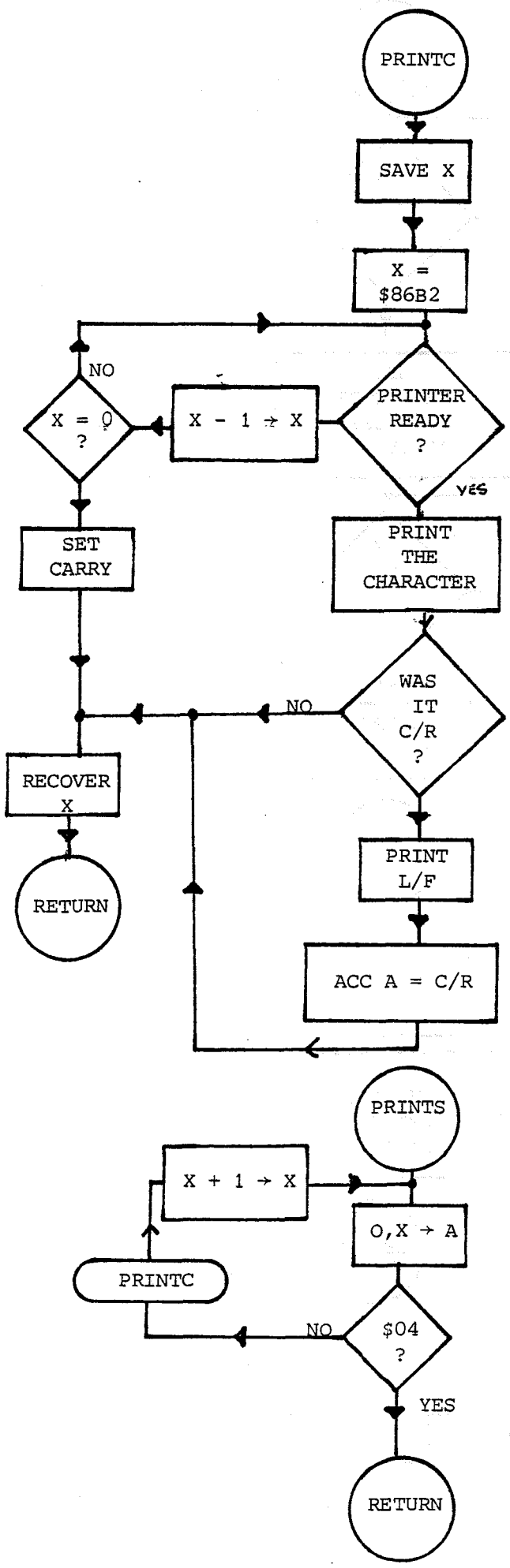
is 128 bytes of RAM in the chip plus a serial and an eight bit parallel ports. The data lines are multiplexed with the low order address lines and an eight bit latch (74LS373) is used with the address strobe line to separate address and data. A one-of-eight decoder driven by the 3 highest address lines generates low level chip selects on 8K boundaries. Seven of the parallel port outputs provide digit select and BCD outputs to drive an array of 8 multiplexed LEDS. An additional parallel port (MC6821) is used to read 4 bits from a 16 key keyboard encoder. (Originally it was intended to use a parallel for communicating with PEST hence the use of an extra parallel port chip).

Internal clock circuitry requires only an external crystal to provide the basic clock rate. The crystal was chosen to also provide a frequency input for a baud-rate generator chip. A timer provides a power-on reset. This reset also disables the EPROM containing the program to prevent a bus contention at switch on and pulls P22, the clock input line for the serial interface, low as the chip has optional operating modes which must be selected during a reset. The RS232C lines TXD, RXD and RTS are buffered by line drivers and receivers.

In addition to the program in the EPROM there is a utility program that was used in development and uses two of the chip select lines to start and stop an external counter which then asserts a non-maskable interrupt to aid program tracing.

FLOW CHARTS


INIT
Program and
Initialise I/O
SET
INTERRUPT
MASK
ADC
PROGRAM ADC
FOR 12 BITS
IN
DAC
PROGRAM DAC
FOR 10 BITS
OUT + 4 OUT
TO LEDs
TIMER
PROGRAM
COUNTER 0
÷1000 FROM
1 MHz
1 kHz to ADC and
Counters 1 and 2
PROGRAM
COUNTER 1
÷1000 FROM
1 kHz
1 Hz to Clock
Interrupt
INITIALISE
COUNTER 2
(DELAY)
Will count down
at sampling
frequency = 1 kHz
PRINTR
7 BITS OUT TO
PRINTER WITH
HANDSHAKE
MSB tested for Printer
ready strobe pulse
generated on write to Printer
ACIAS
SERIAL INTERFACE
8 BITS INTERNAL
÷ 16 2 STOP BITS
COPY
INITIAL
CLOCK DATA
INTO RAM
RETURN

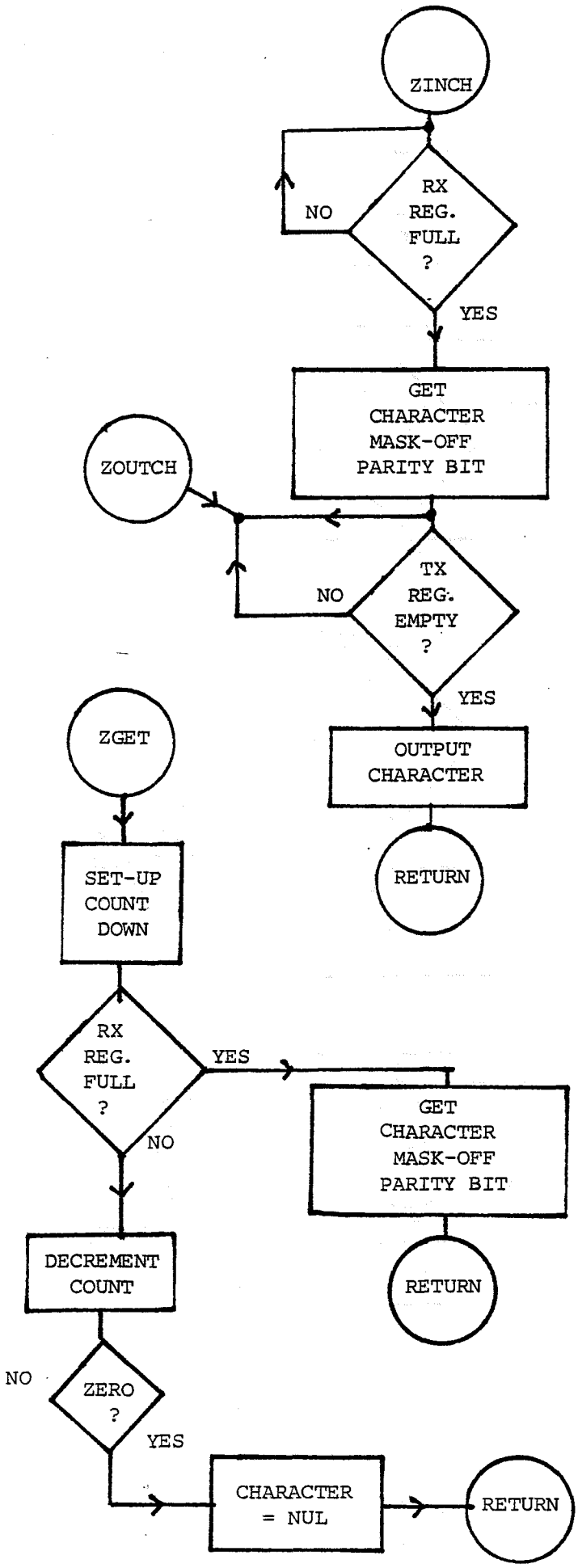


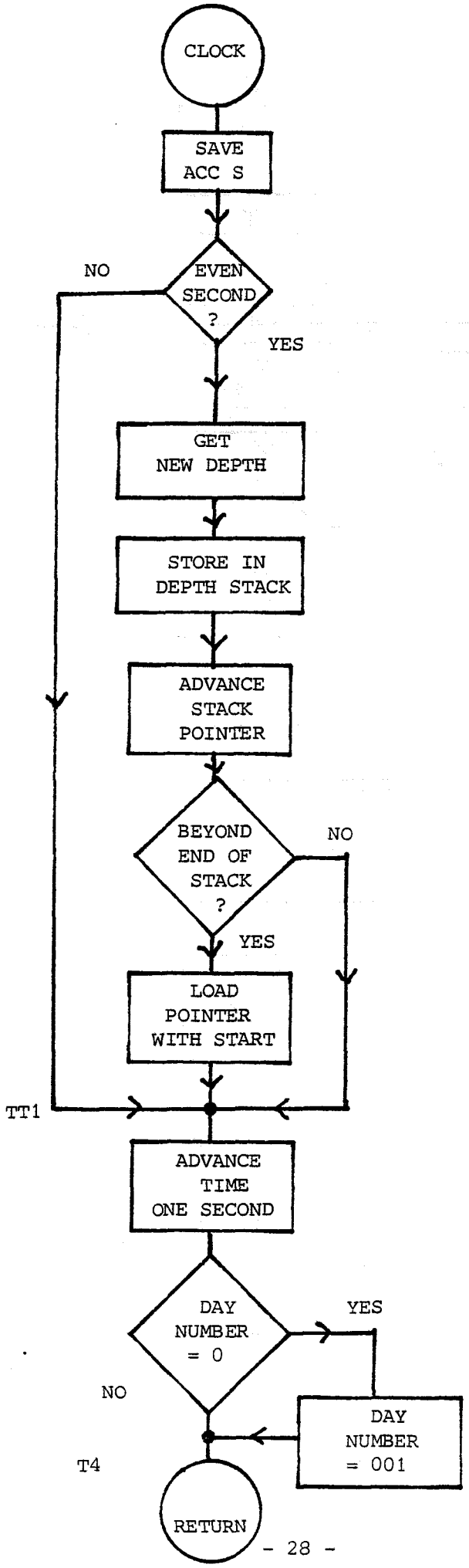
PRINT A CHARACTER FROM ACC. A

TIME-OUT DELAY COUNT

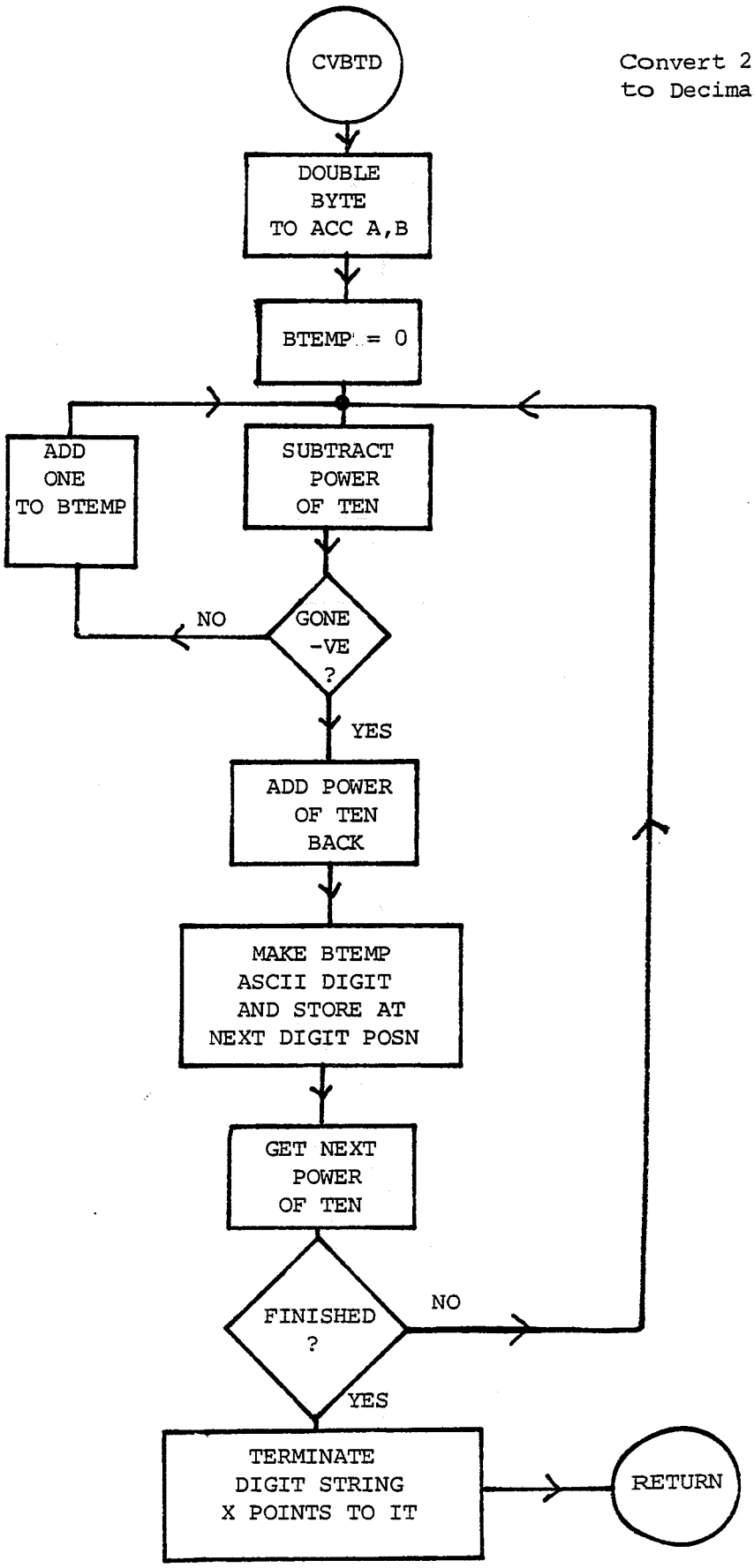
FOLLOW C/R WITH L/F SO THAT PRINTER PRINTS ITS BUFFER STORE CONTENTS

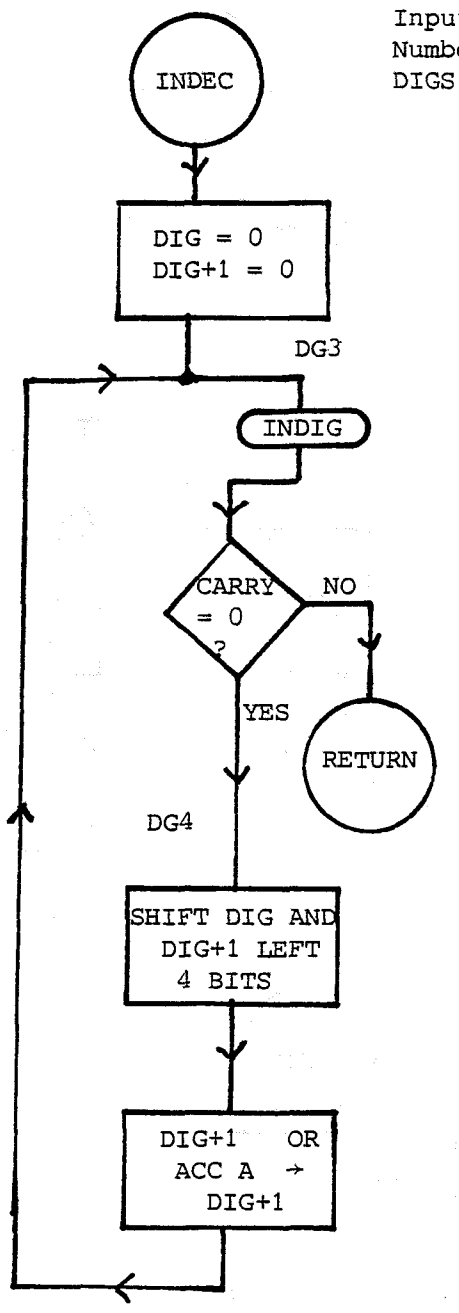
PRINT A STRING. X POINTS TO START OF STRING. $04 TERMINATES IT.


ZINCH
RX REG. FULL ?
NO
YES
GET CHARACTER MASK-OFF PARITY BIT
ZOUTCH
TX REG. EMPTY ?
NO
YES
OUTPUT CHARACTER
RETURN
ZGET
SET-UP COUNT DOWN
RX REG. FULL ?
YES
NO
GET CHARACTER MASK-OFF PARITY BIT
RETURN
DECREMENT COUNT
ZERO ?
NO
YES
CHARACTER = NUL
RETURN

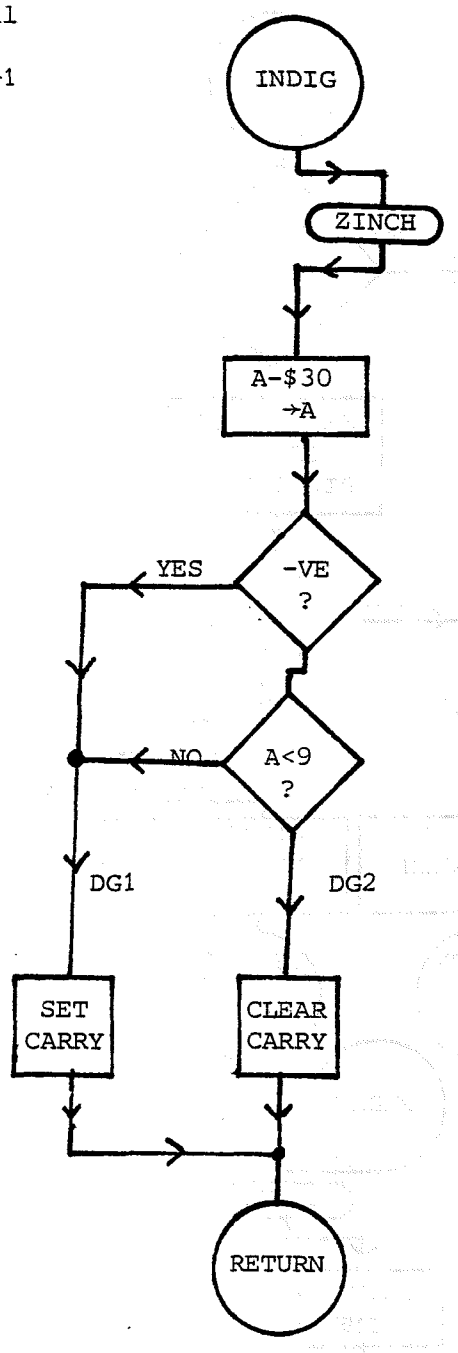

CLOCK
SAVE ACC S
EVEN SECOND ?
NO
YES
GET NEW DEPTH
STORE IN DEPTH STACK
ADVANCE STACK POINTER
BEYOND END OF STACK ?
NO
YES
LOAD POINTER WITH START
TT1
ADVANCE TIME ONE SECOND
DAY NUMBER = 0
YES
NO
DAY NUMBER = 001
T4
RETURN

Convert 2 Bytes Binary
to Decimal Digit String

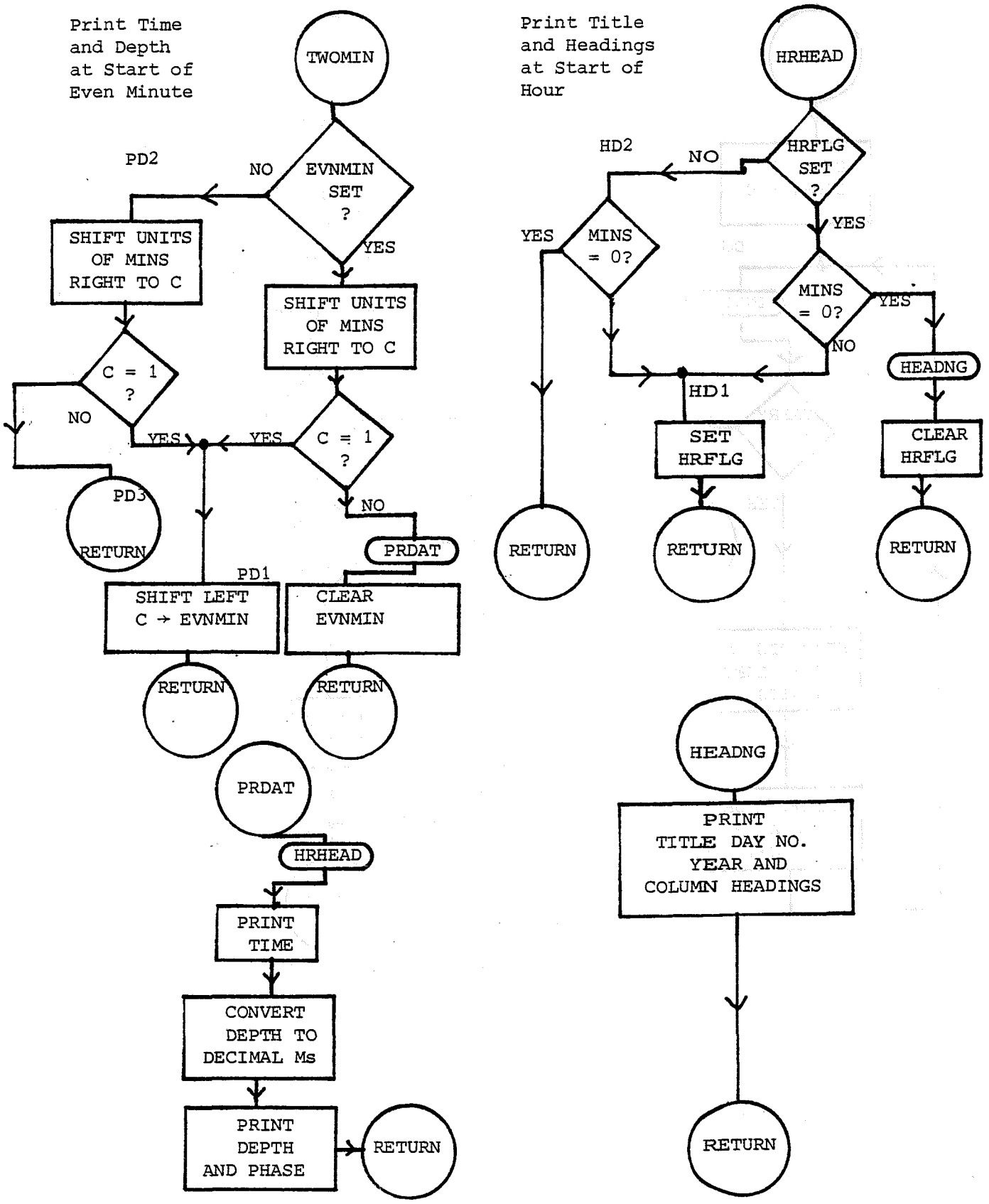


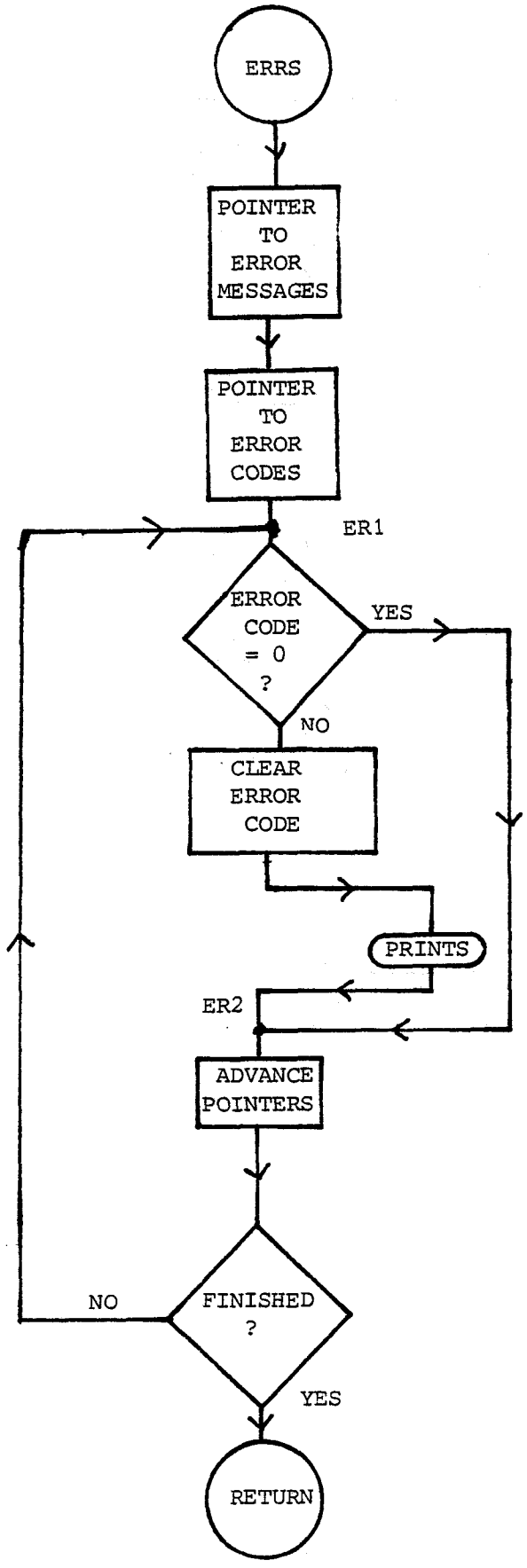


Input Decimal Number to DIGS - DIGS+1



Input Character into Acc A

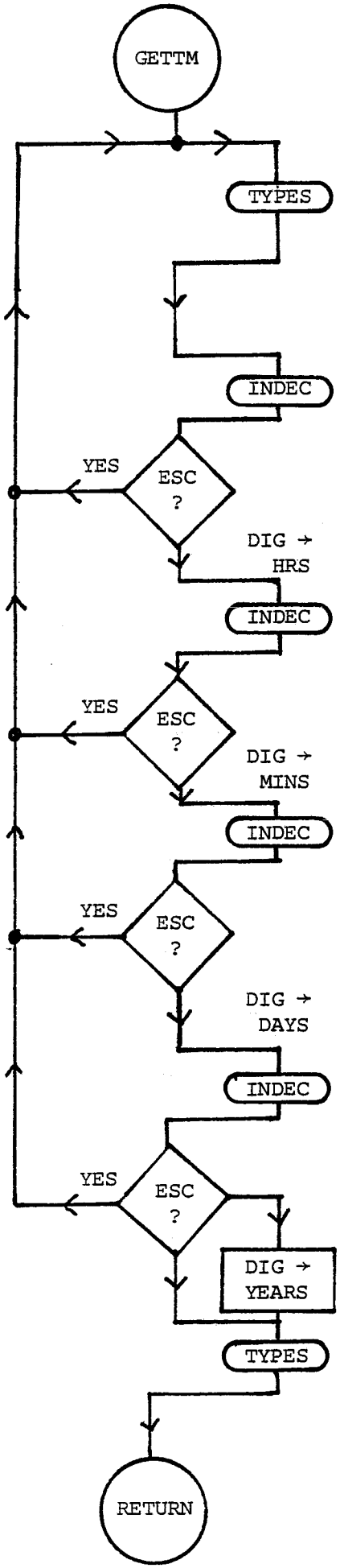

Print Time and Depth at Start of Even Minute
TWOMIN
EVNMIN SET ?
NO
YES
PD2
SHIFT UNITS OF MINS RIGHT TO C
SHIFT UNITS OF MINS RIGHT TO C
C = 1 ?
NO
YES
YES
C = 1 ?
NO
PD3
RETURN
PRDAT
PD1
SHIFT LEFT C → EVNMIN
CLEAR EVNMIN
RETURN
RETURN
PRDAT
HRHEAD
PRINT TIME
CONVERT DEPTH TO DECIMAL Ms
PRINT DEPTH AND PHASE
RETURN
Print Title and Headings at Start of Hour
HRHEAD
HRFLG SET ?
NO
YES
HD2
YES
MINS = 0?
MINS = 0?
YES
NO
HEADNG
HD1
SET HRFLG
CLEAR HRFLG
RETURN
RETURN
RETURN
HEADNG
PRINT TITLE DAY NO. YEAR AND COLUMN HEADINGS
RETURN



Any of Seven Error Codes may be Non-Zero

Make sure it is only printed once

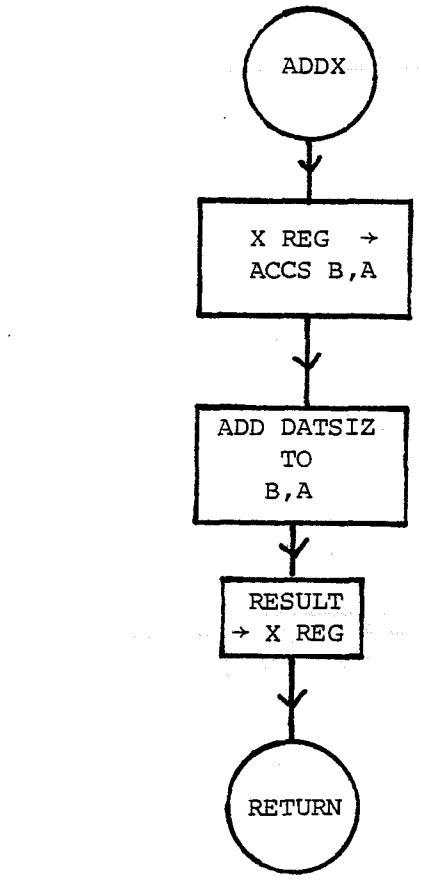
Print Error Message



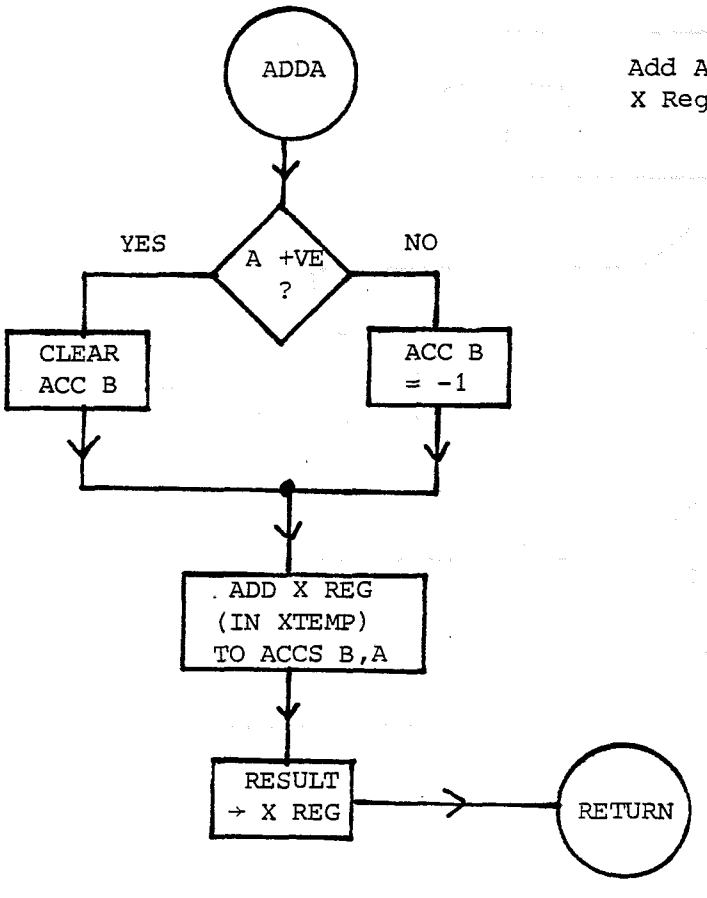
Get time
from Keyboard

Type "Time Please ...

Type "Thank You"

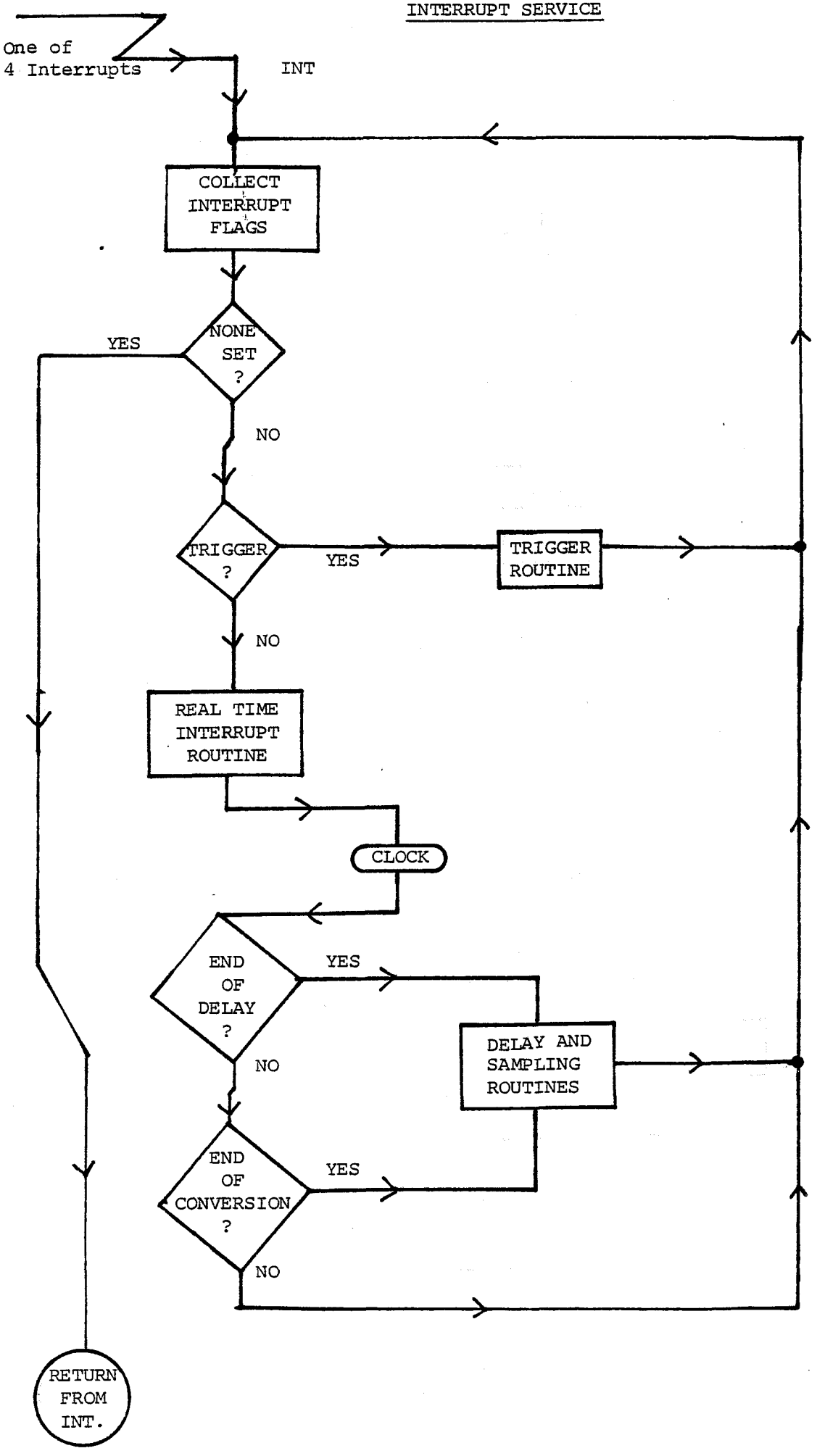


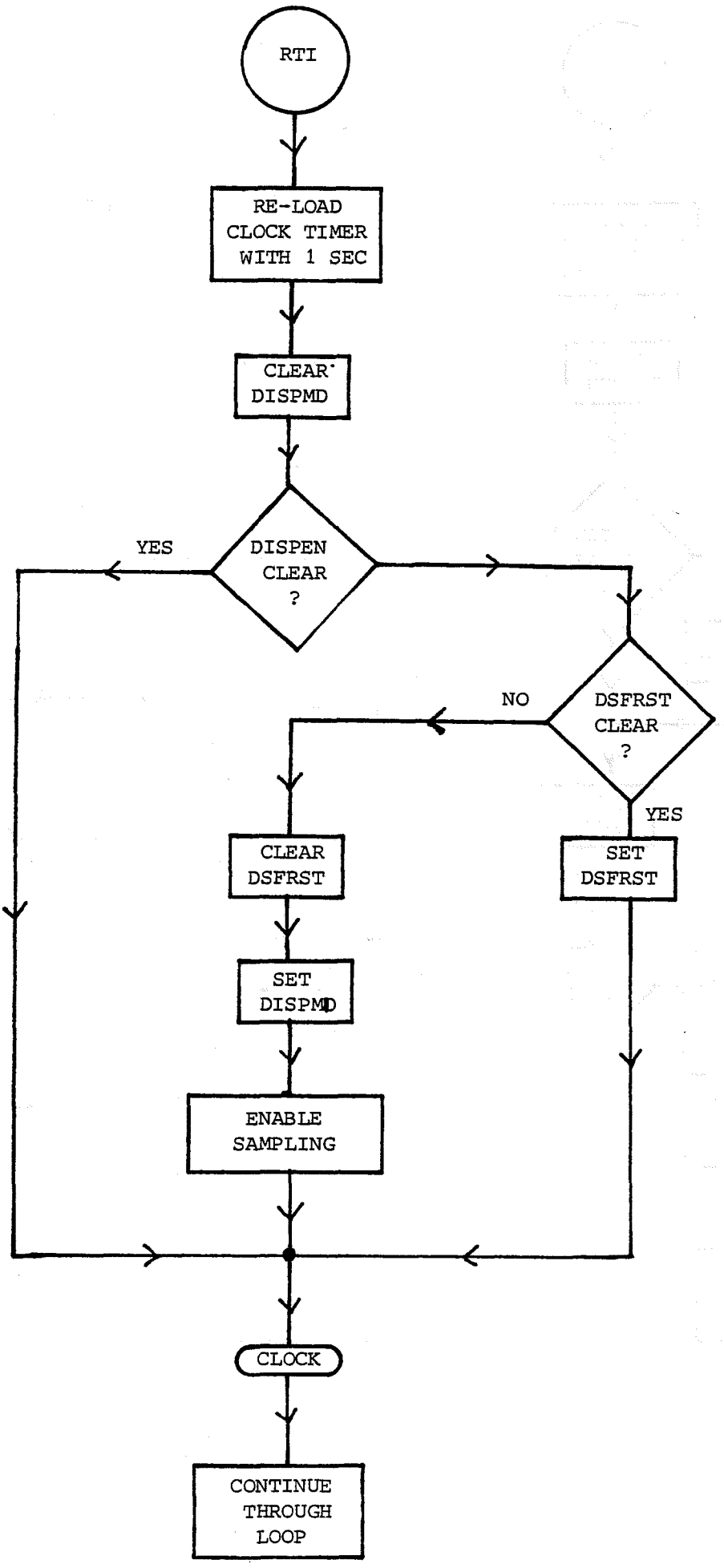
Add DATSIZ to X Reg

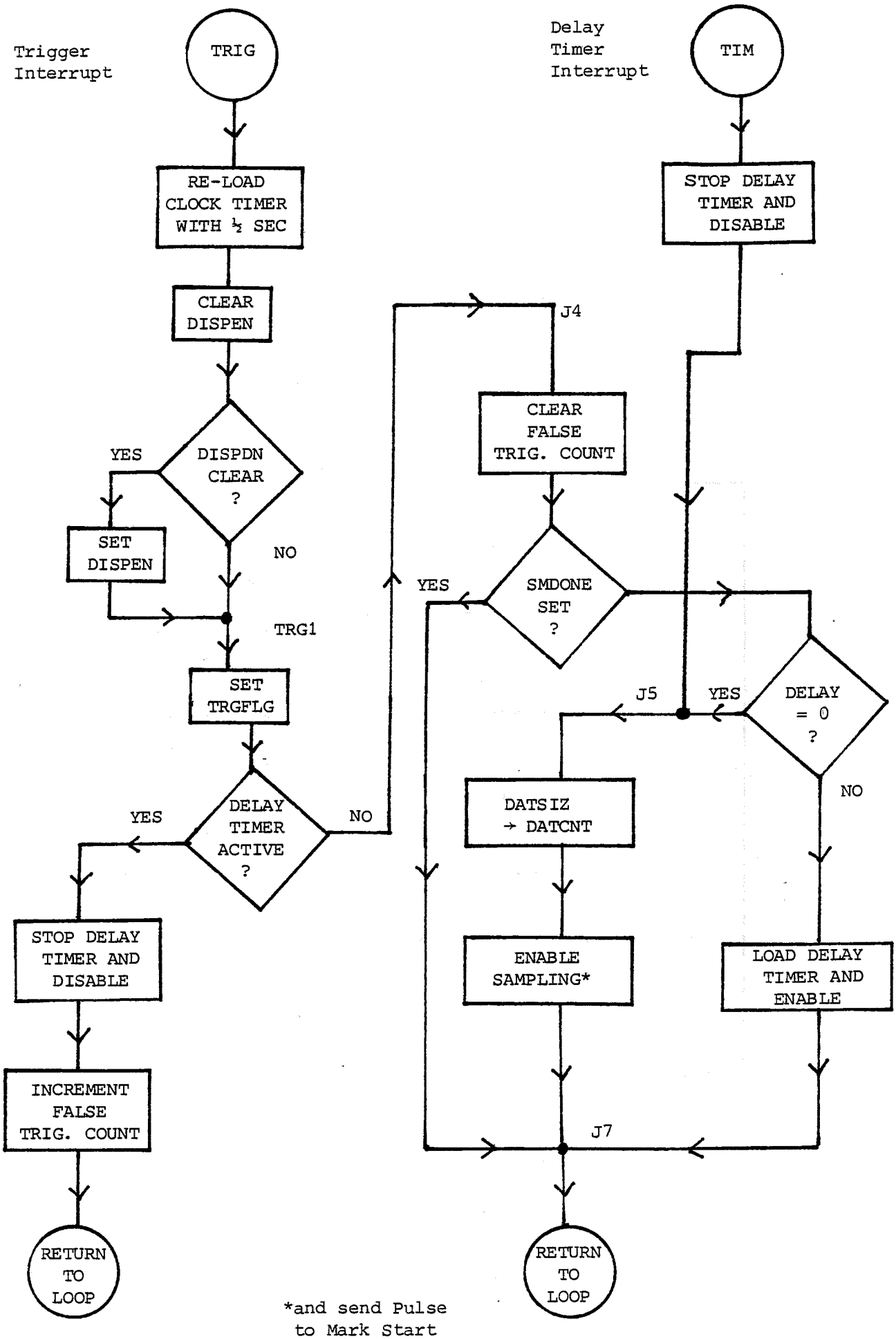


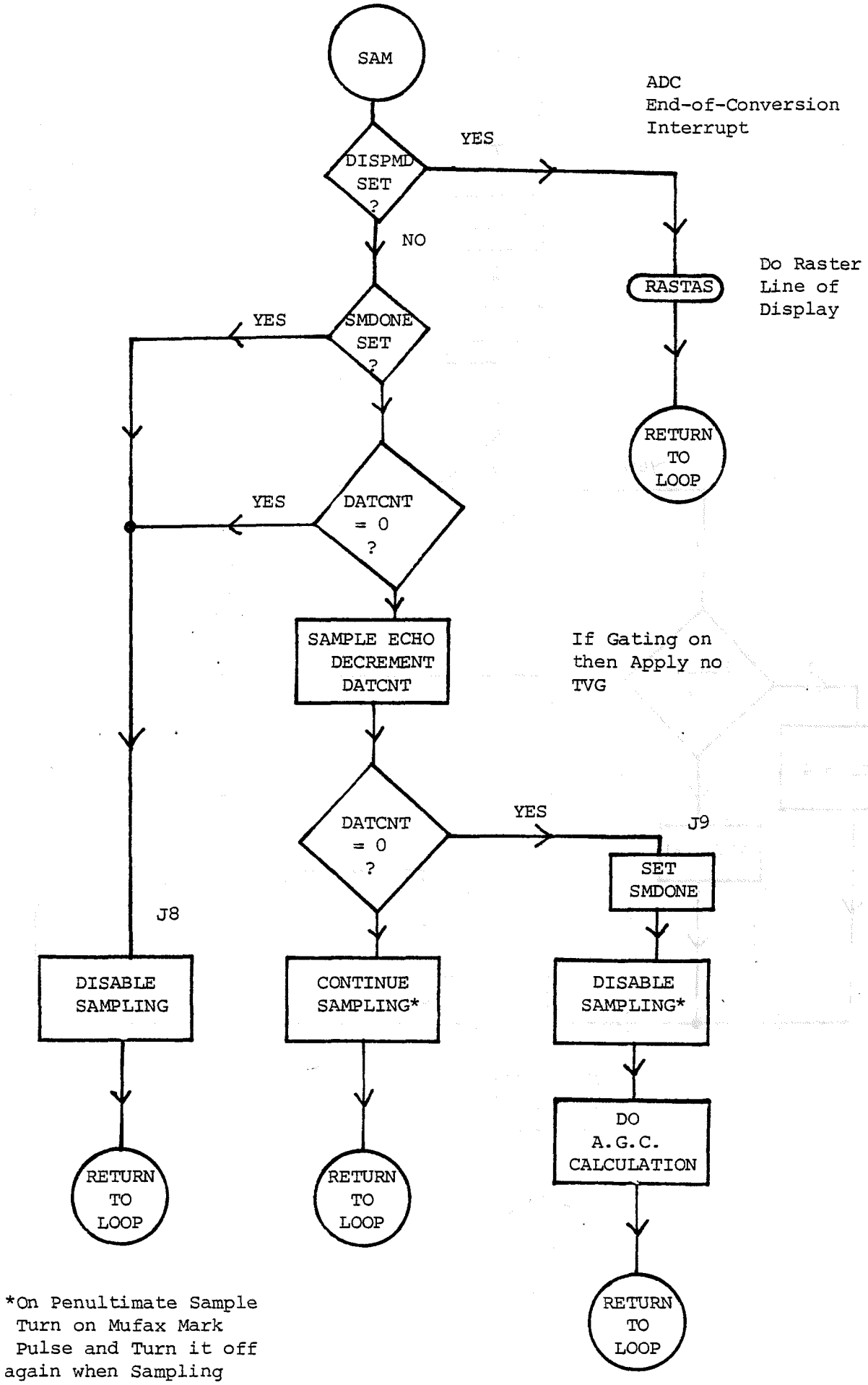
Add Acc A to X Reg (or Subtract if Negative)

INTERRUPT SERVICE

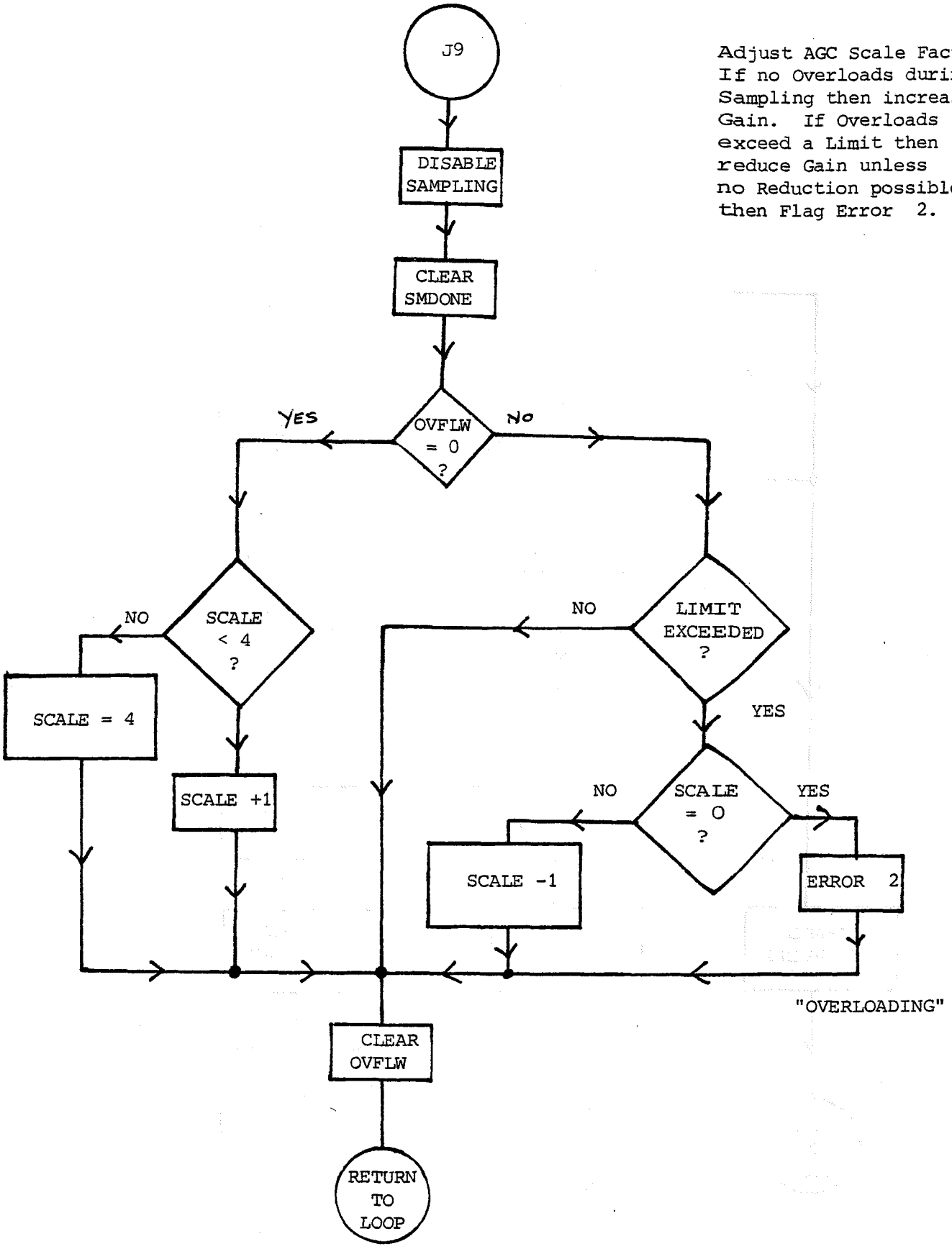



RTI
RE-LOAD CLOCK TIMER WITH 1 SEC
CLEAR DISPMD
DISPEN CLEAR ?
YES
NO
DSFRST CLEAR ?
YES
CLEAR DSFRST
SET DSFRST
SET DISPMD
ENABLE SAMPLING
CLOCK
CONTINUE THROUGH LOOP

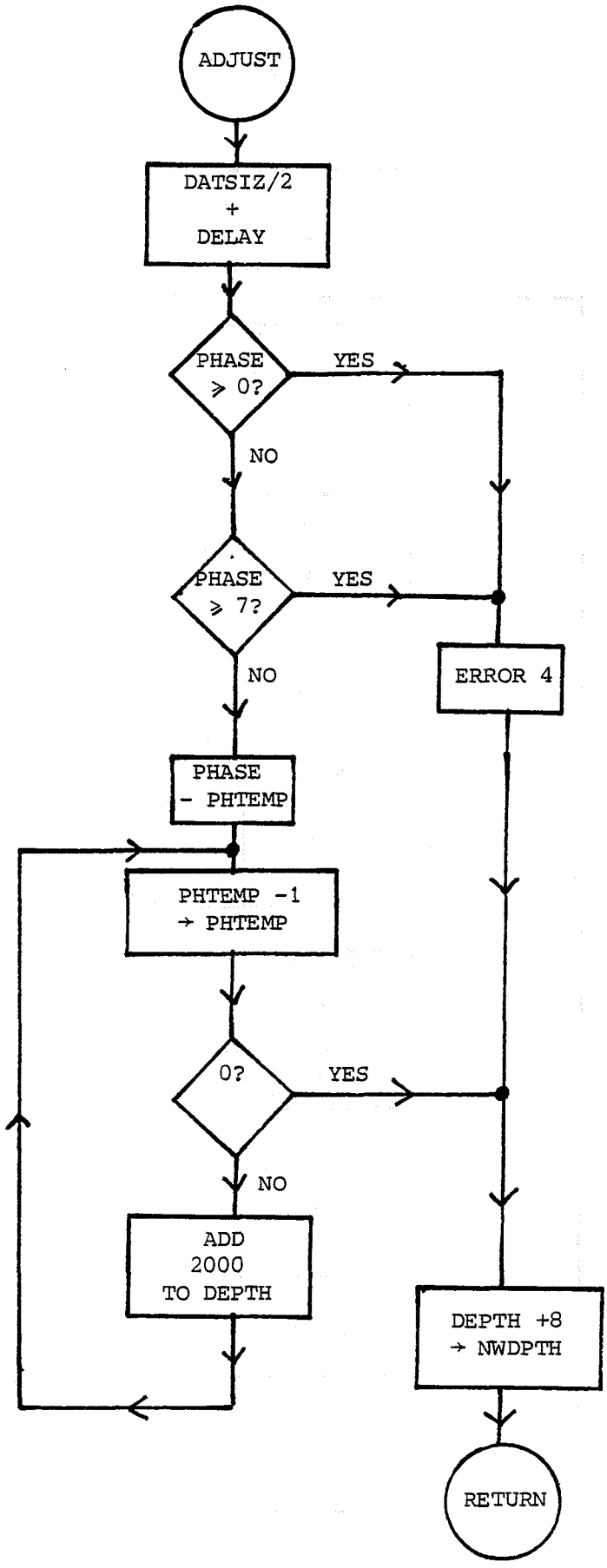

Trigger Interrupt
TRIG
RE-LOAD CLOCK TIMER WITH ½ SEC
CLEAR DISPEN
DISPDN CLEAR ?
YES
SET DISPEN
NO
TRG1
SET TRGFLG
DELAY TIMER ACTIVE ?
YES
NO
STOP DELAY TIMER AND DISABLE
INCREMENT FALSE TRIG. COUNT
RETURN TO LOOP
Delay Timer Interrupt
TIM
STOP DELAY TIMER AND DISABLE
J4
CLEAR FALSE TRIG. COUNT
SMDONE SET ?
YES
J5
YES
DELAY = 0 ?
NO
DATSIZ → DATCNT
ENABLE SAMPLING*
LOAD DELAY TIMER AND ENABLE
J7
RETURN TO LOOP
*and send Pulse to Mark Start of Sampling

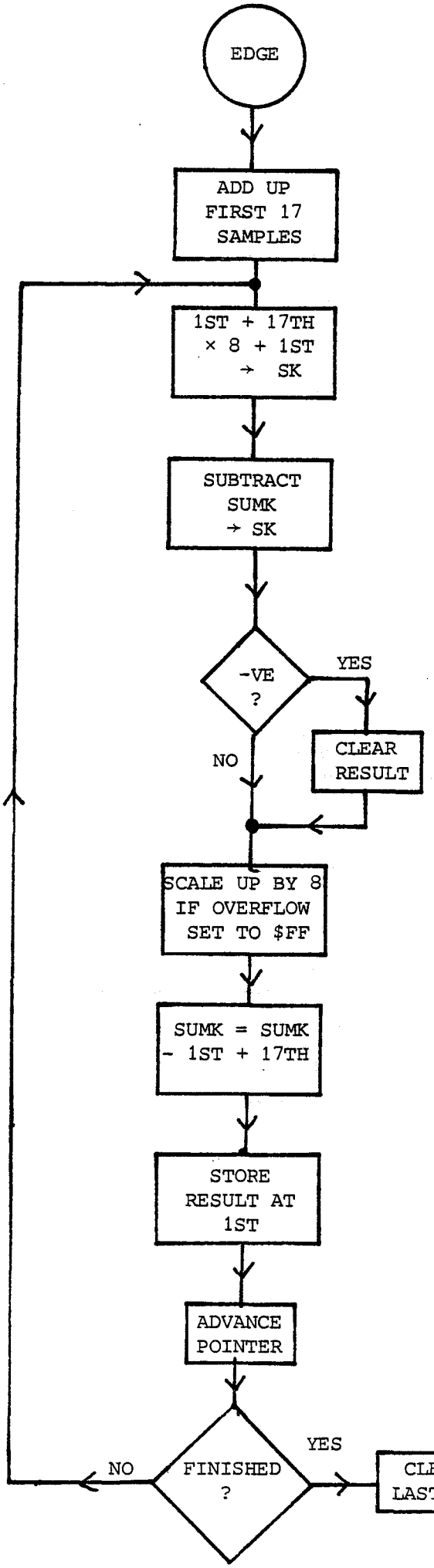


*On Penultimate Sample
Turn on Mufax Mark
Pulse and Turn it off
again when Sampling
finished.



Adjust AGC Scale Factor. If no Overloads during Sampling then increase Gain. If Overloads exceed a Limit then reduce Gain unless no Reduction possible then Flag Error 2.


ADJUST
DATSIZ/2
+
DELAY
PHASE
⩾ 0?
YES
NO
PHASE
⩾ 7?
YES
NO
ERROR 4
PHASE
- PHTEMP
PHTEMP -1
→ PHTEMP
0?
YES
NO
ADD
2000
TO DEPTH
DEPTH +8
→ NWDPTH
RETURN



Detect Leading Edge of Echo

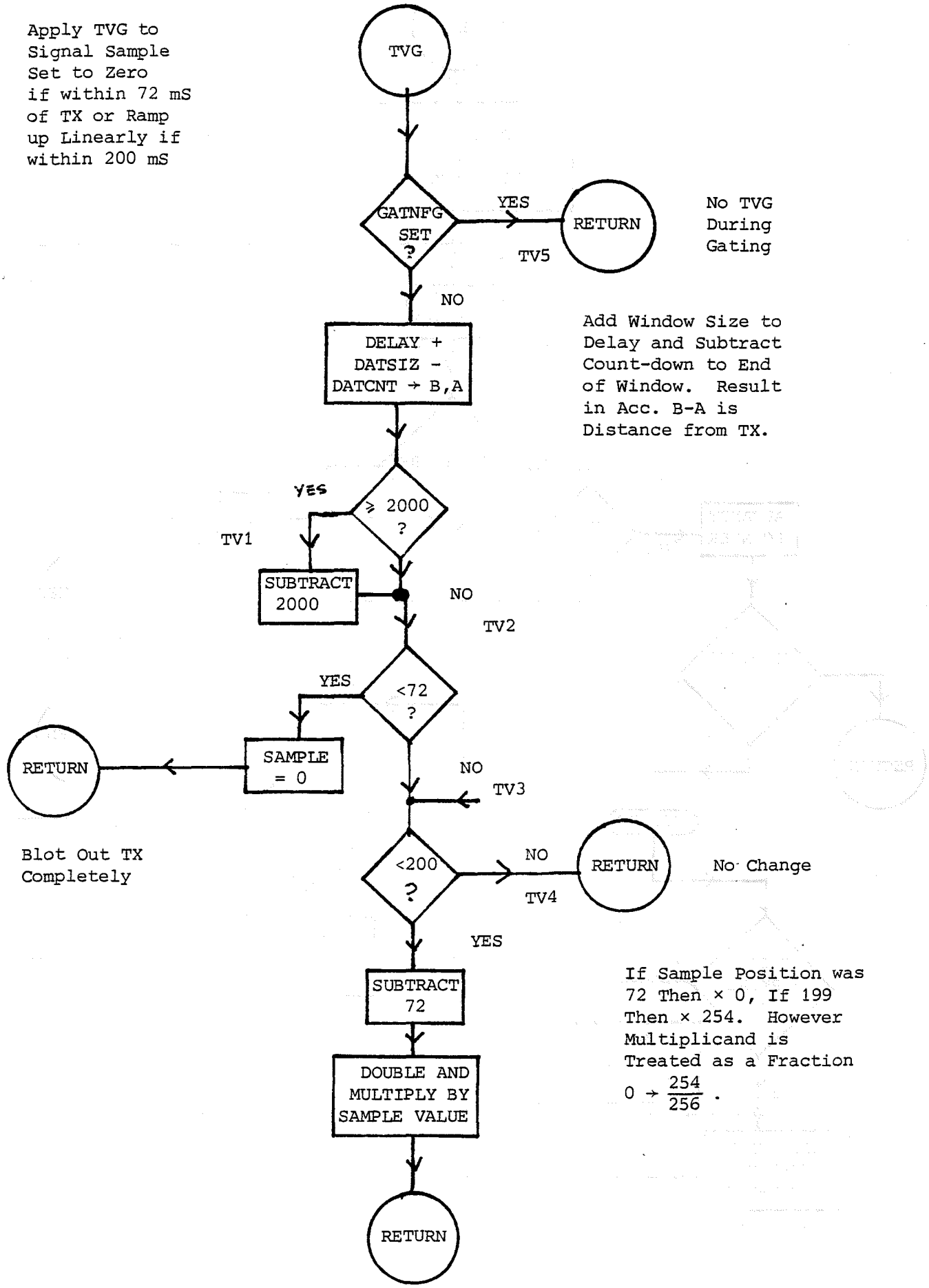
Initialise SUMK to Avoid GLITCH at Start

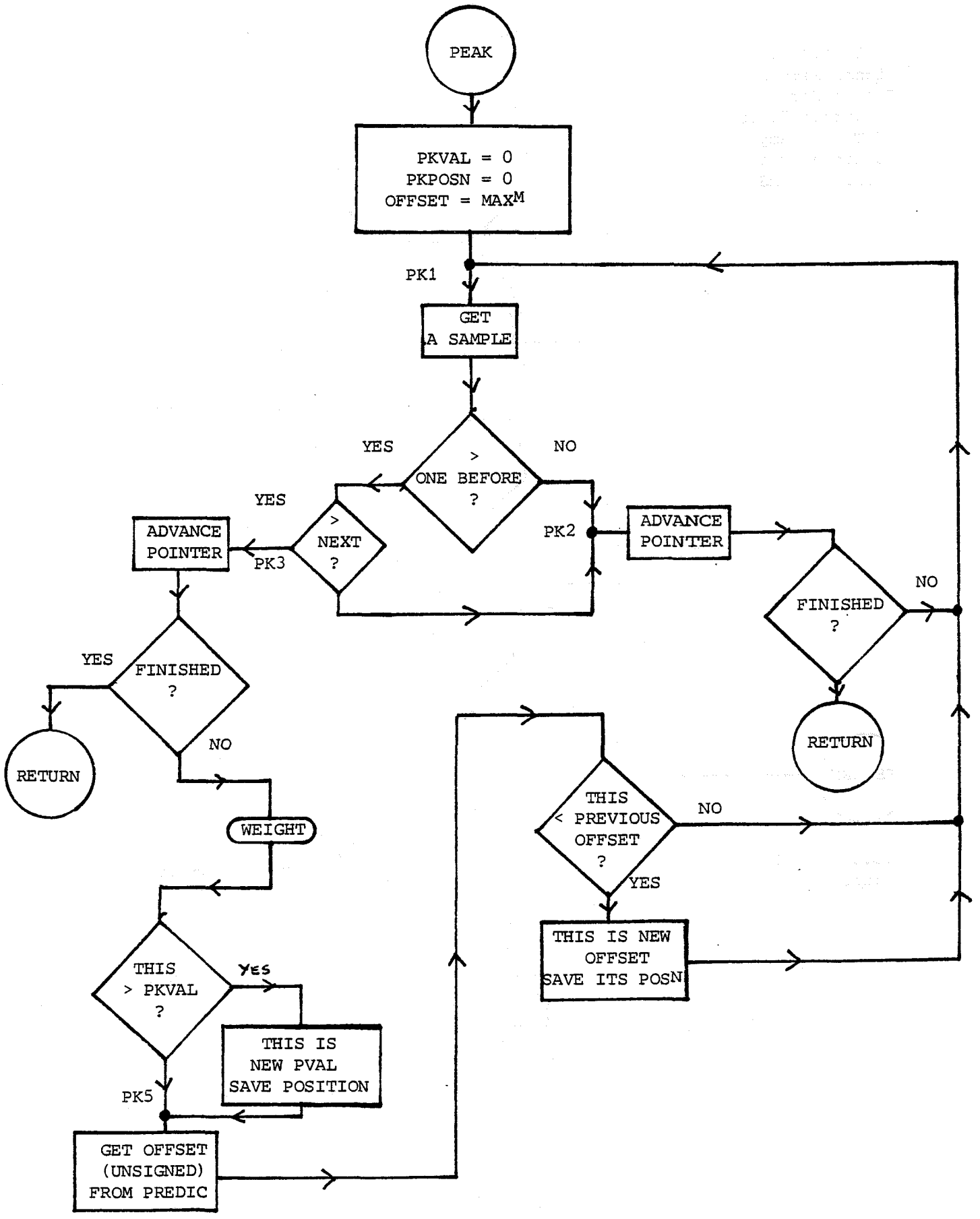
Running Correlation

Half-Wave Rectify. We Only Want Rising Edges

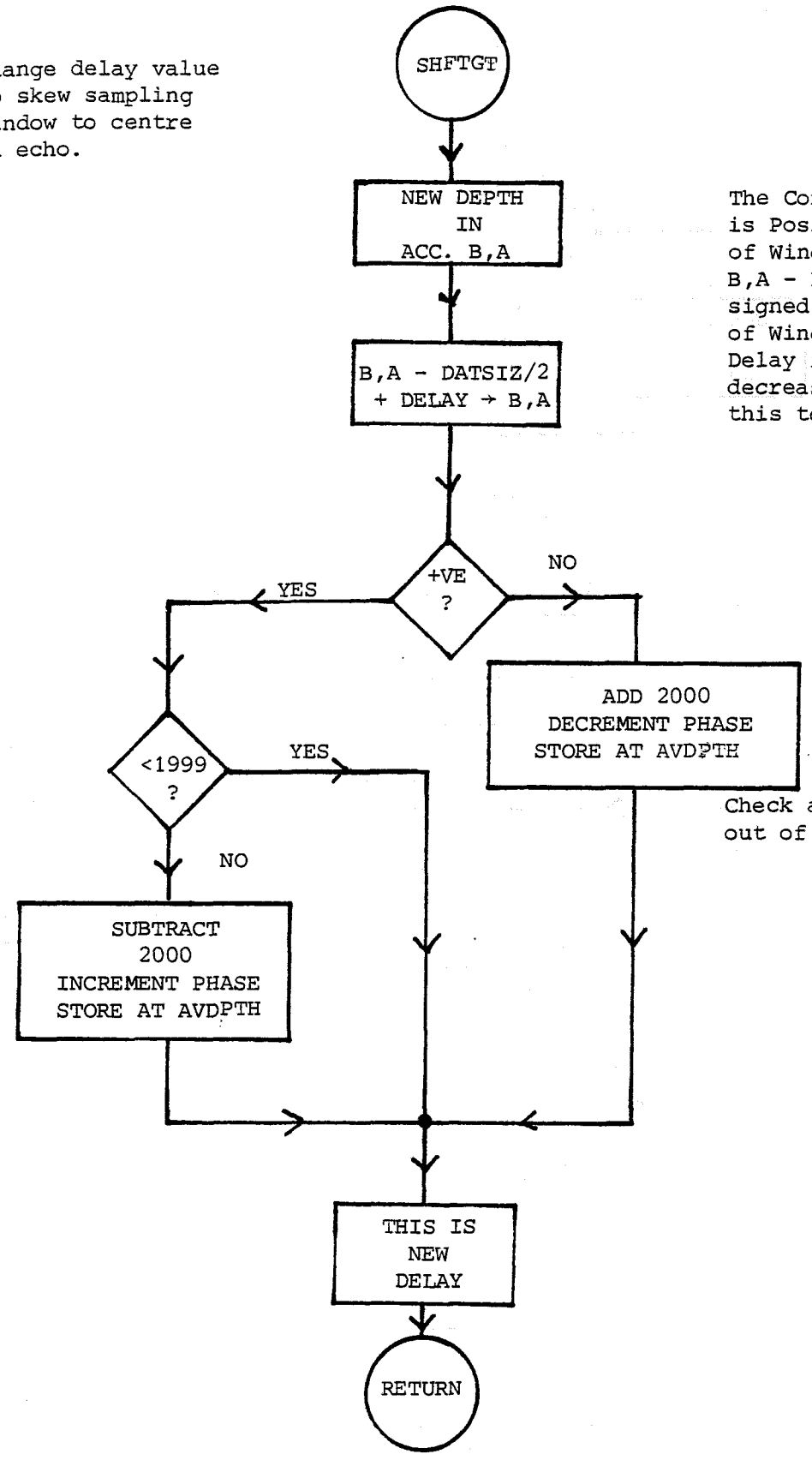
Update Running Sum

Result Put Back into I/P Array 8 Places Early (Corrected for in 'ADJUST')


Apply TVG to Signal Sample Set to Zero if within 72 mS of TX or Ramp up Linearly if within 200 mS
TVG
GATNFG SET ?
YES
RETURN
TV5
No TVG During Gating
NO
DELAY + DATSIZ - DATCNT → B,A
Add Window Size to Delay and Subtract Count-down to End of Window. Result in Acc. B-A is Distance from TX.
YES
≥ 2000 ?
TV1
SUBTRACT 2000
NO
TV2
YES
<72 ?
RETURN
SAMPLE = 0
NO
TV3
Blot Out TX Completely
<200 ?
NO
RETURN
TV4
No Change
YES
SUBTRACT 72
DOUBLE AND MULTIPLY BY SAMPLE VALUE
If Sample Position was 72 Then × 0, If 199 Then × 254. However Multiplicand is Treated as a Fraction 0 → 254/256 .
RETURN

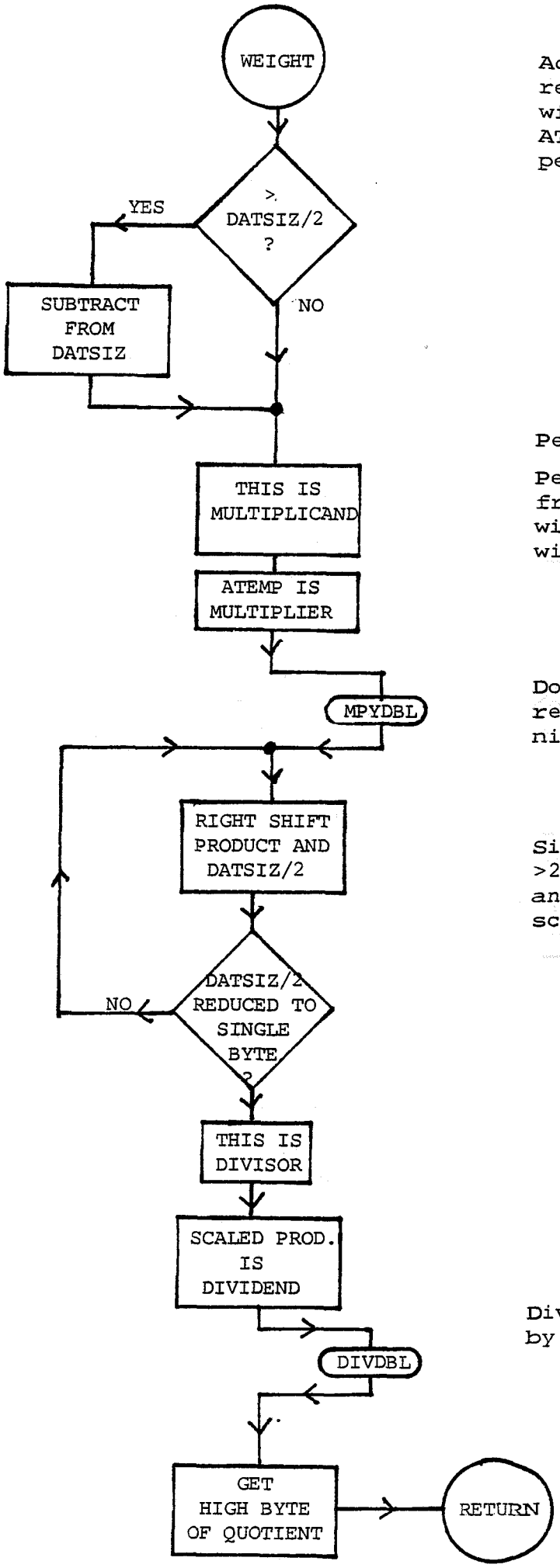

PEAK
PKVAL = 0
PKPOSN = 0
OFFSET = MAXM
PK1
GET A SAMPLE
> ONE BEFORE ?
YES
NO
> NEXT ?
YES
ADVANCE POINTER
PK3
PK2
ADVANCE POINTER
FINISHED ?
NO
RETURN
FINISHED ?
YES
NO
RETURN
WEIGHT
THIS < PREVIOUS OFFSET ?
NO
YES
THIS IS NEW OFFSET SAVE ITS POSN
THIS > PKVAL ?
YES
THIS IS NEW PVAL SAVE POSITION
PK5
GET OFFSET (UNSIGNED) FROM PREDIC

Change delay value to skew sampling window to centre on echo.



The Contents of Acc. B,A is Position from Start of Window.
B,A - DATSIZ/2 is signed Offset from Centre of Window.
Delay is increased/ decreased by adding this to it.

Check and Correct if out of Range.



Acc. B,A contains relative position within window. ATEMP contains peak amplitude.

Perform:-

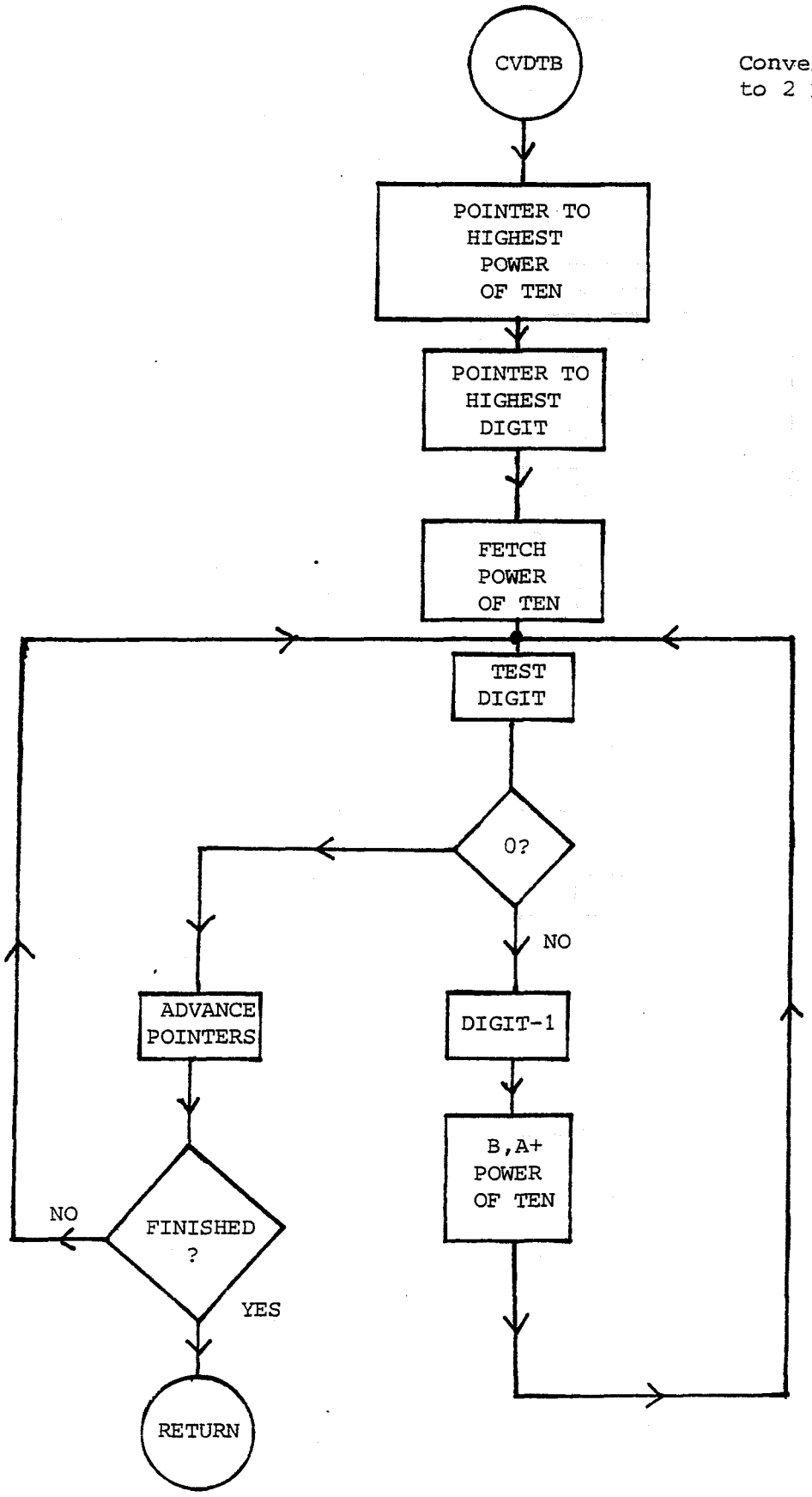
Peak Value × Distance from centre of window ÷ half window width.

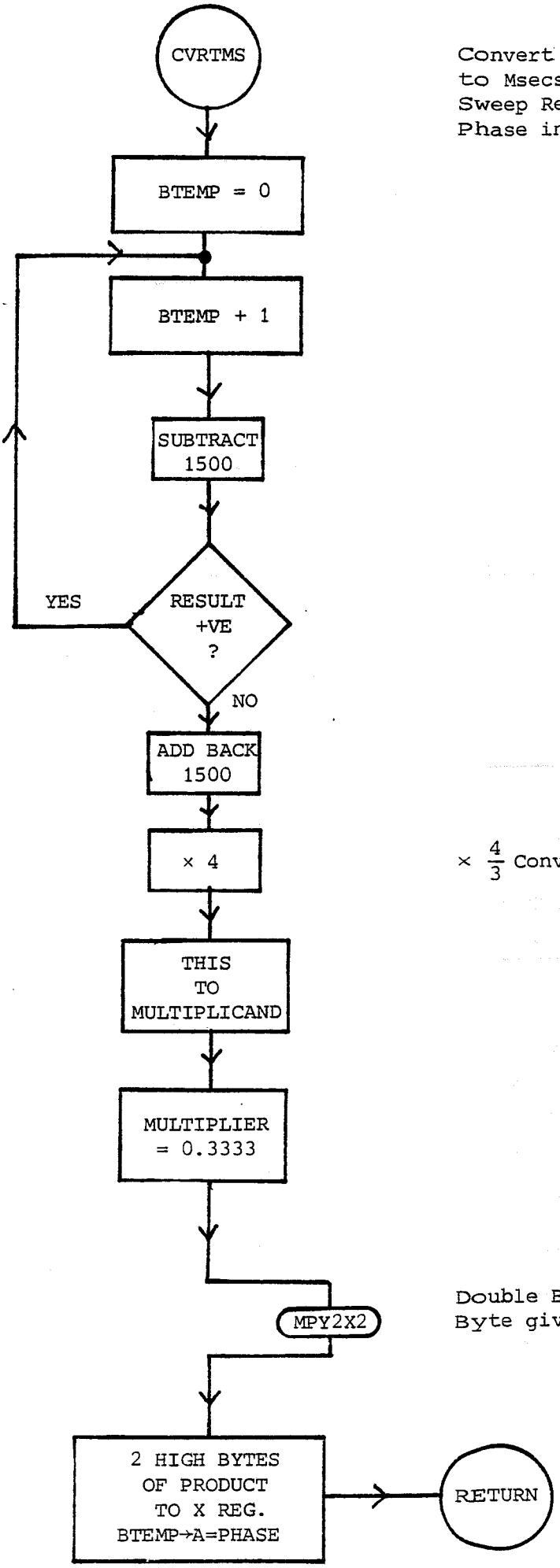
Double Byte × Single retain 2 most significant bytes.

Since DATSIZ/2 may be >255 both Numerator and Denominator are scaled down.

Divide Double Byte by Single Byte.

Convert 4 Decimal Digits
to 2 Bytes Binary.

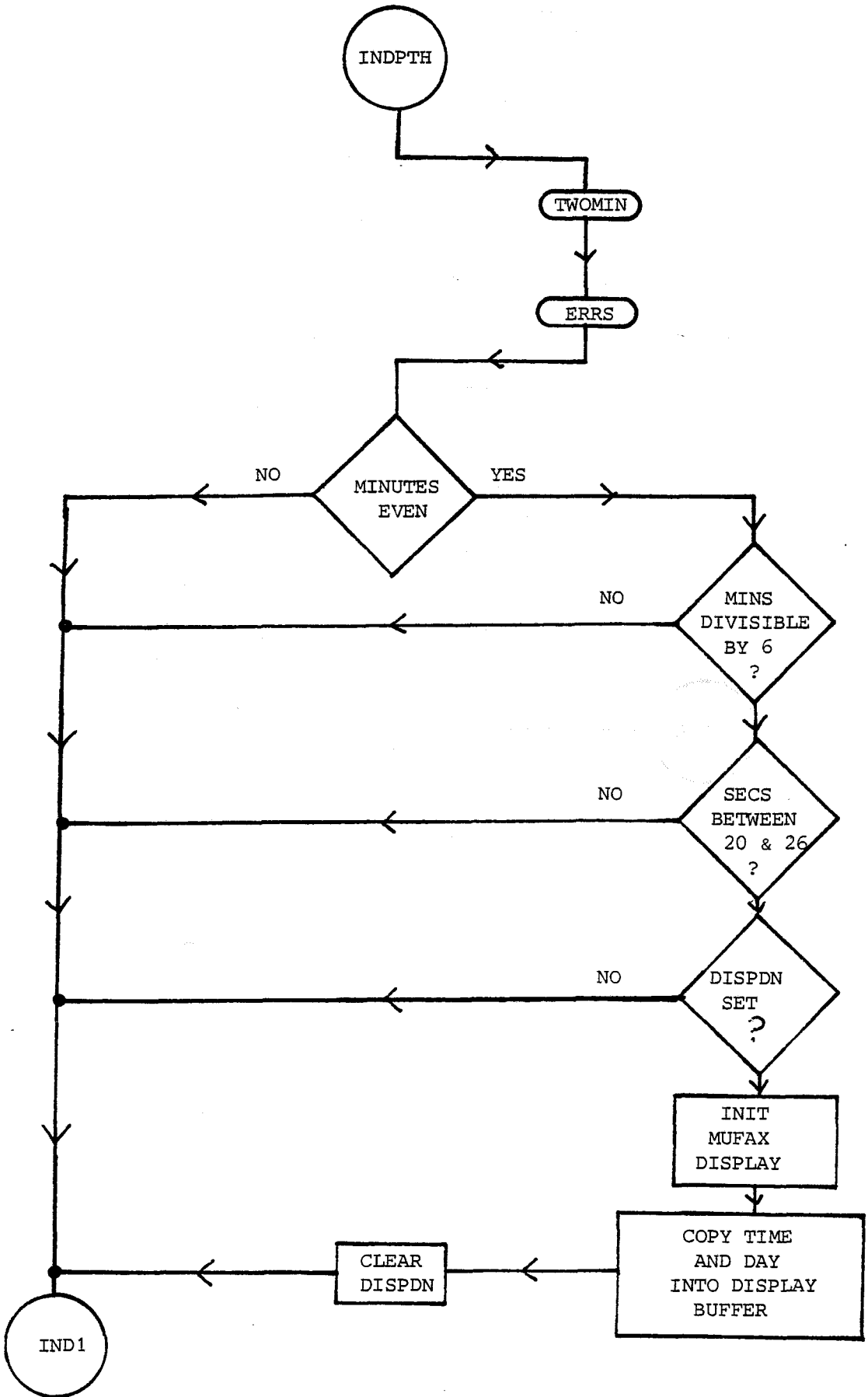


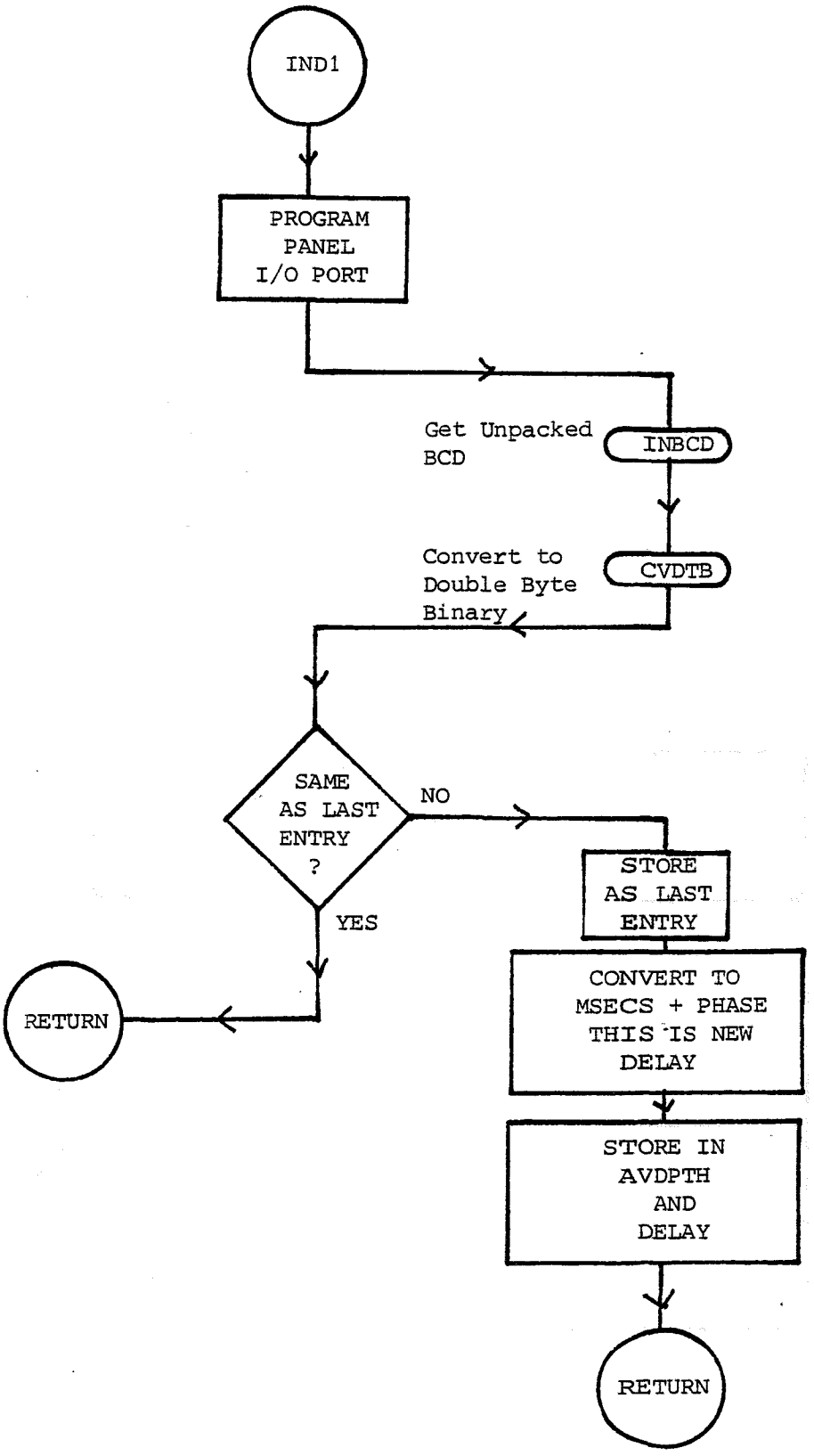


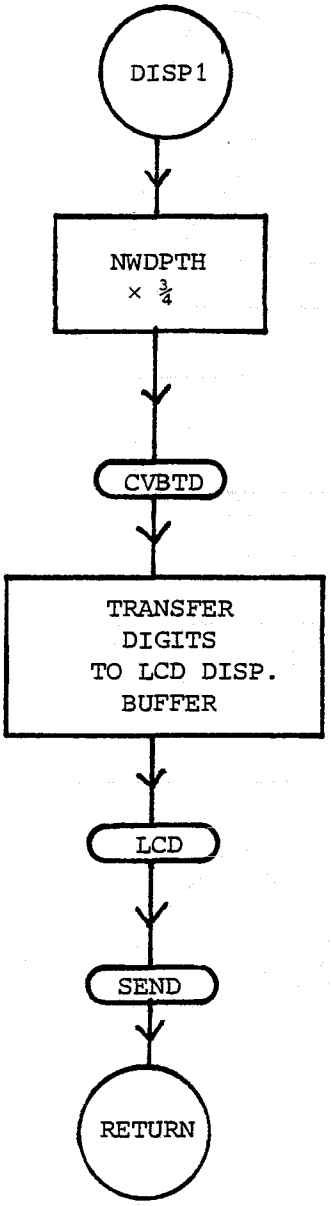
Convert Depth in Metres to Msecs. Re-Start of Sweep Result in X Reg. Phase in Acc. A.

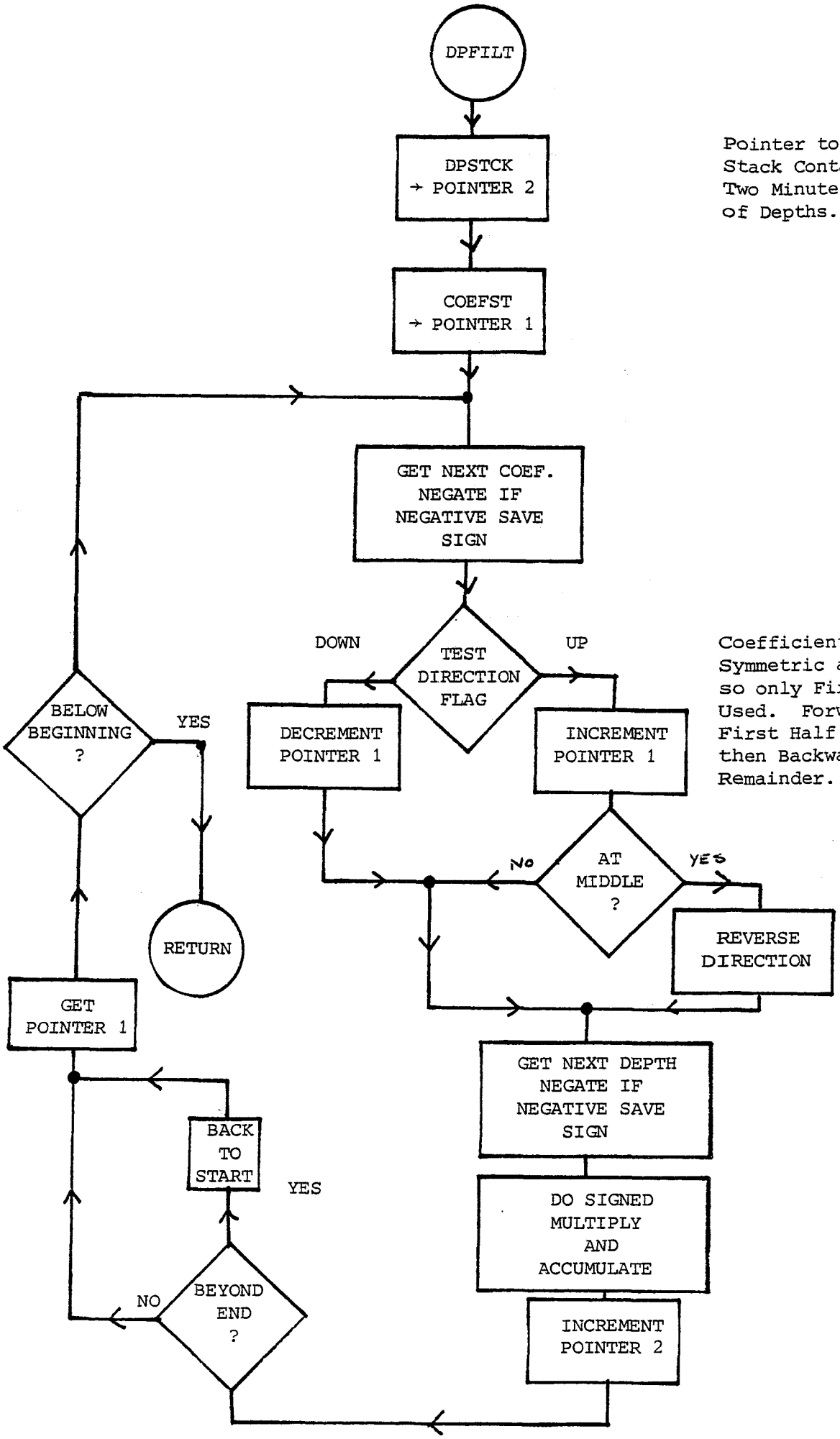
$\times \frac{4}{3}$ Converts Metres to Msecs.

Double Byte × Double Byte gives 4 Byte Product.


INDPTH
TWOMIN
ERRS
MINUTES EVEN
NO
YES
MINS DIVISIBLE BY 6 ?
NO
SECS BETWEEN 20 & 26 ?
NO
DISPDN SET ?
NO
INIT MUFAX DISPLAY
COPY TIME AND DAY INTO DISPLAY BUFFER
CLEAR DISPDN
IND1

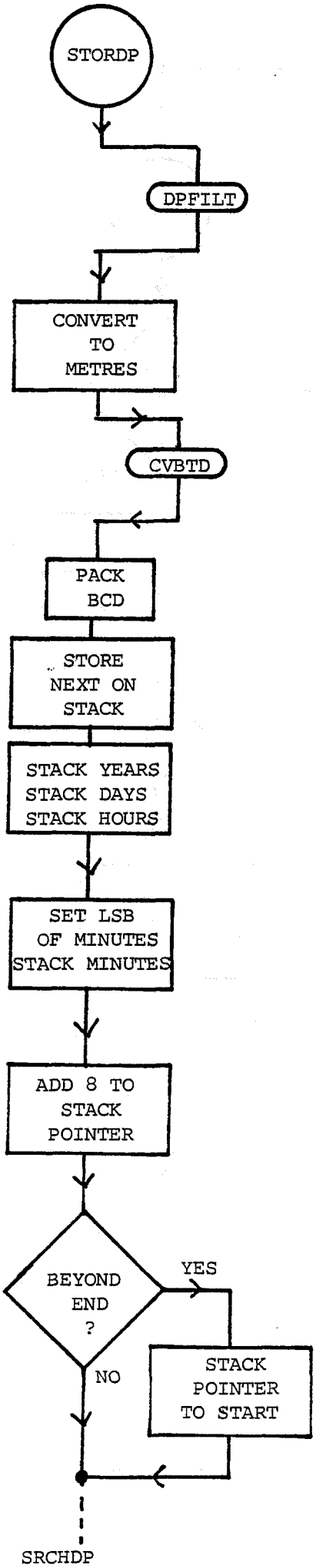

IND1
PROGRAM PANEL I/O PORT
Get Unpacked BCD
INBCD
Convert to Double Byte Binary
CVDTB
SAME AS LAST ENTRY ?
NO
YES
RETURN
STORE AS LAST ENTRY
CONVERT TO MSECS + PHASE THIS IS NEW DELAY
STORE IN AVDPTH AND DELAY
RETURN


DISP1
NWDPTH
× ¾
CVBTD
TRANSFER
DIGITS
TO LCD DISP.
BUFFER
LCD
SEND
RETURN



Pointer to Rotating Stack Containing Two Minutes Worth of Depths.

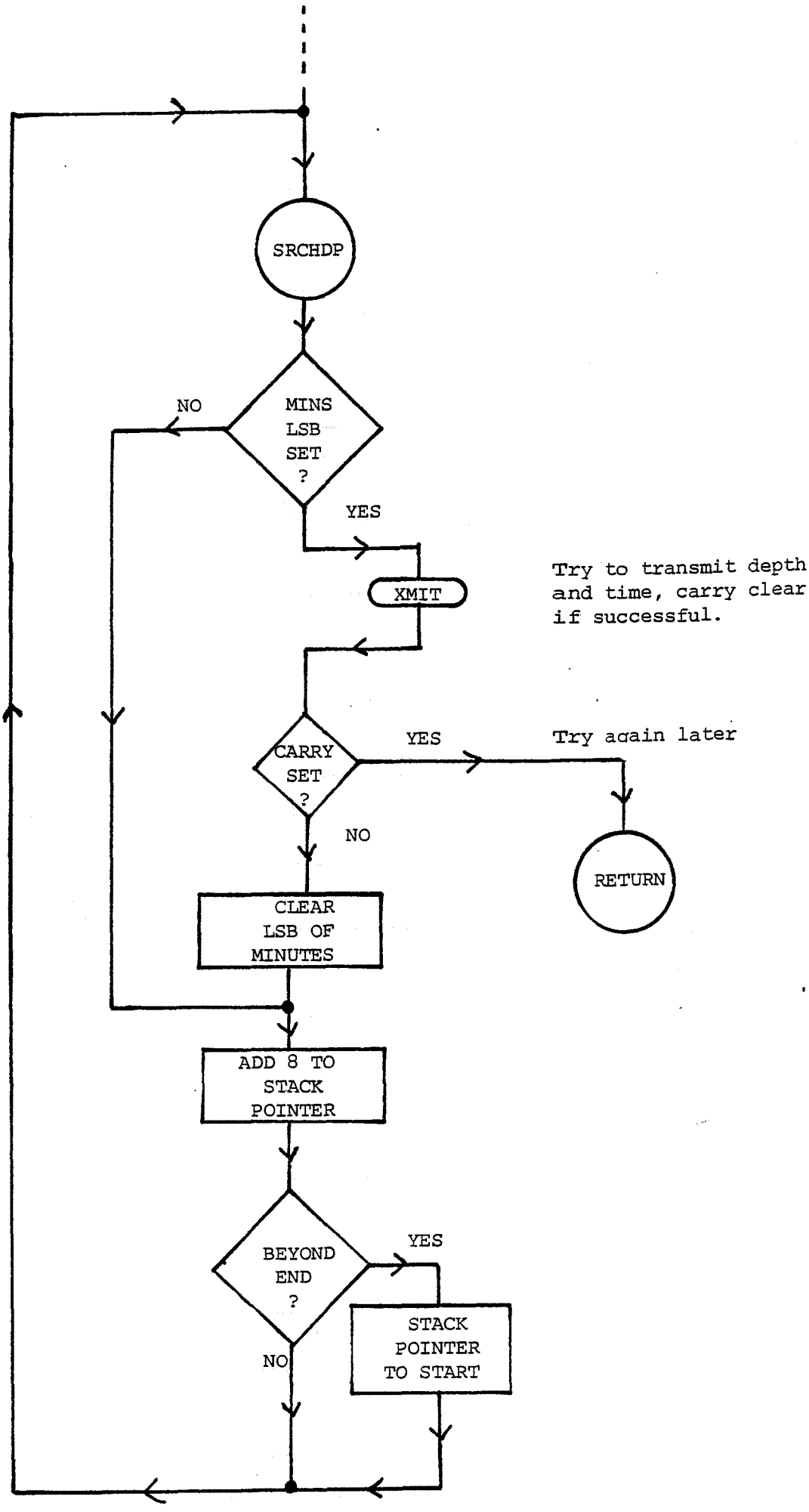
Coefficients are Symmetric about Middle so only First Half are Used. Forwards during First Half of Filter then Backwards for Remainder.

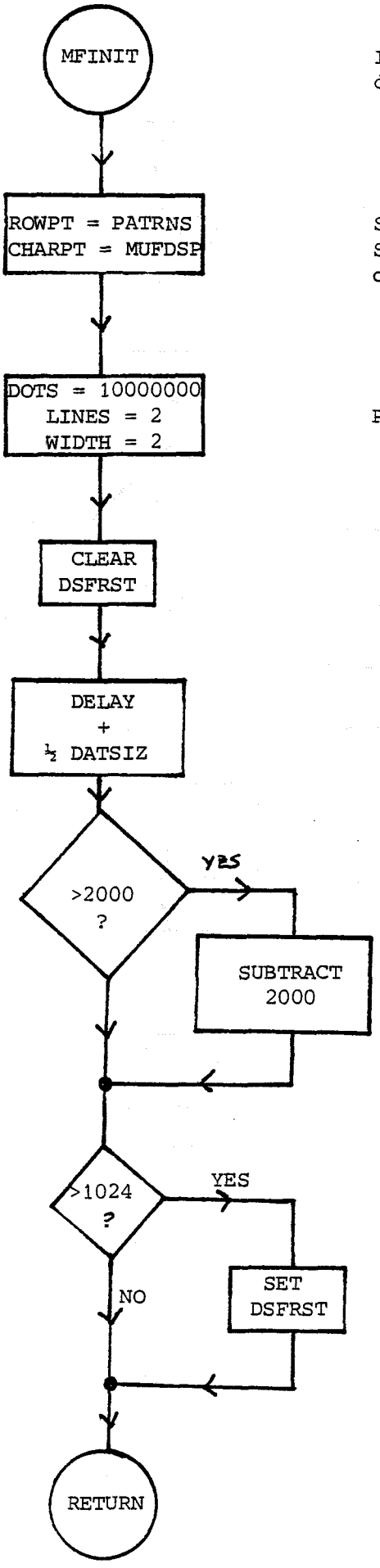


Get two Minutes of Most Recent Depths and Filter.

Stack contains about 34 hours of two minute depths in groups of eight bytes:-
2 Bytes - Depth
2 Bytes - Year
2 Bytes - Day No.
1 Byte - Hours
1 Byte - Mins

LSB of Minutes is set to flag a depth not yet sent to ship's logger. This is clear when the depth has been sent and acknowledged.

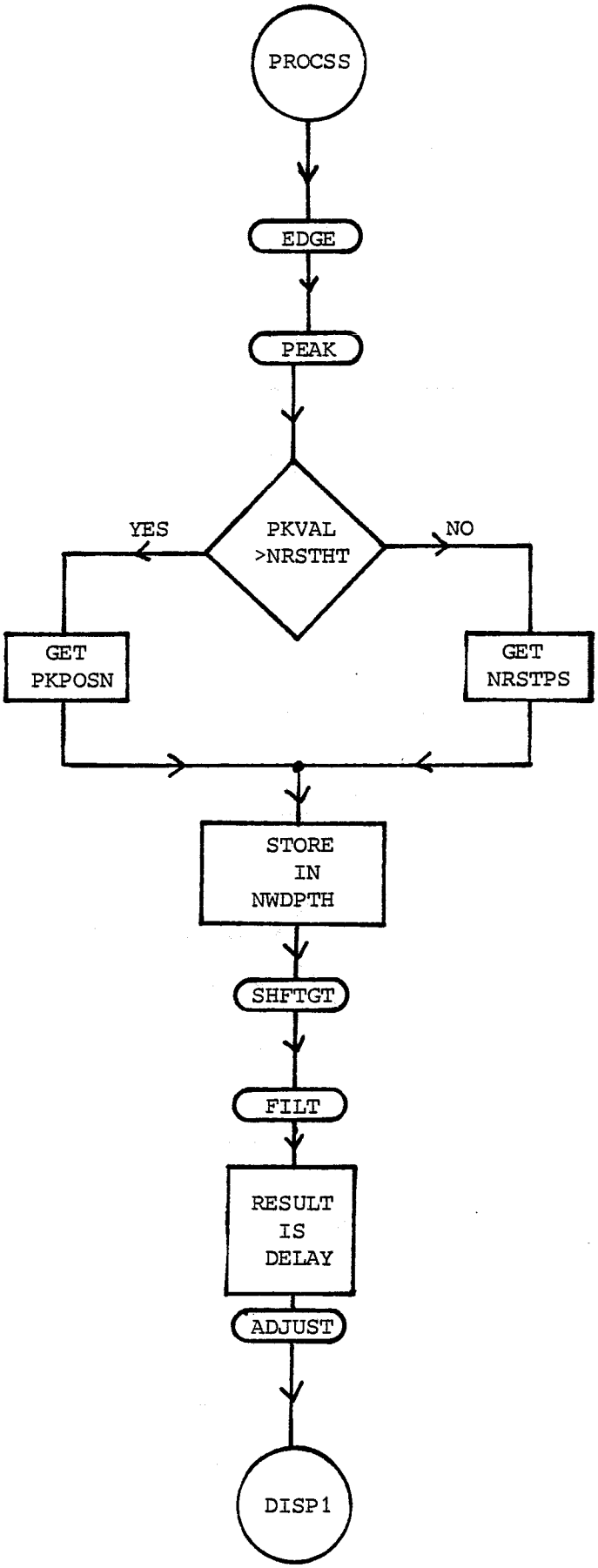

SRCHDP
MINS LSB SET ?
NO
YES
XMIT
Try to transmit depth and time, carry clear if successful.
CARRY SET ?
YES
Try again later
RETURN
NO
CLEAR LSB OF MINUTES
ADD 8 TO STACK POINTER
BEYOND END ?
YES
STACK POINTER TO START
NO

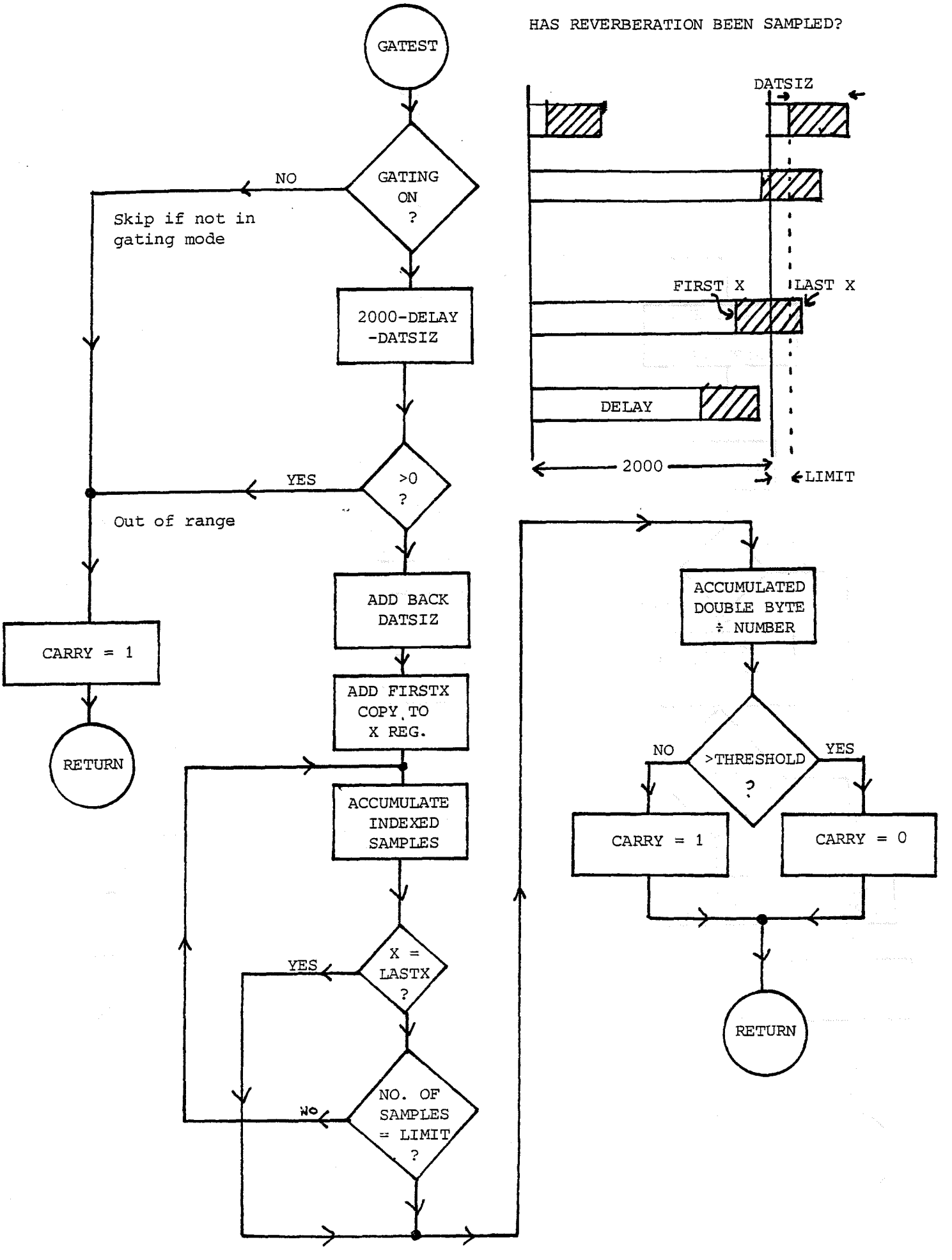


Initialise Mufax raster display.

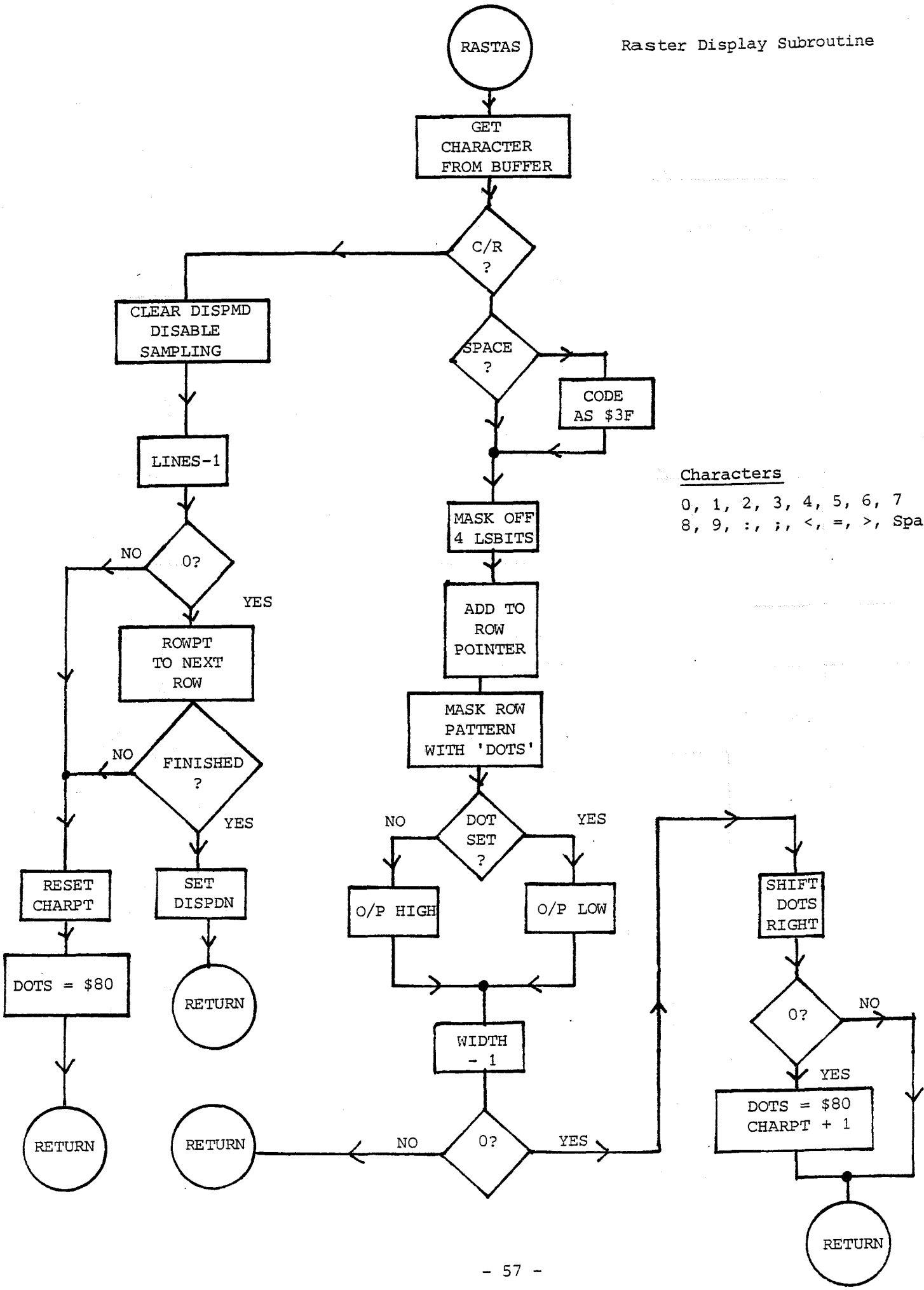
Start of dot patterns.
Start of raster display character buffer.

Pixel size set to 2 × 2.


PROCSS
EDGE
PEAK
PKVAL
>NRSTHT
YES
NO
GET
PKPOSN
GET
NRSTPS
STORE
IN
NWDPTH
SHFTGT
FILT
RESULT
IS
DELAY
ADJUST
DISP1

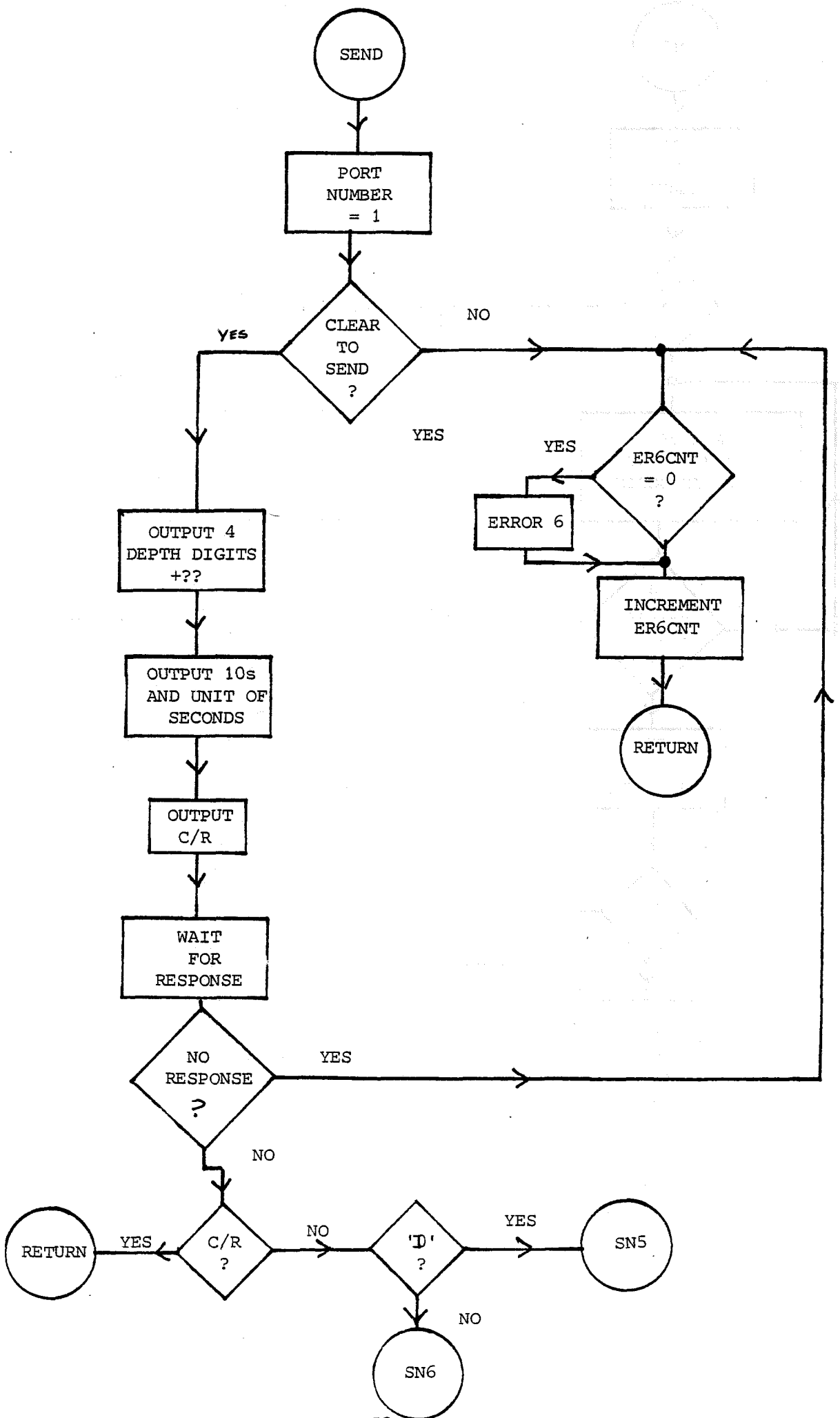

GATEST
GATING ON ?
NO
Skip if not in gating mode
2000-DELAY -DATSIZ
>0 ?
YES
Out of range
CARRY = 1
RETURN
ADD BACK DATSIZ
ADD FIRSTX COPY TO X REG.
ACCUMULATE INDEXED SAMPLES
X = LASTX ?
YES
NO. OF SAMPLES = LIMIT ?
NO
HAS REVERBERATION BEEN SAMPLED?
DATSIZ
FIRST X
LAST X
DELAY
2000
LIMIT
ACCUMULATED DOUBLE BYTE ÷ NUMBER
>THRESHOLD ?
NO
YES
CARRY = 1
CARRY = 0
RETURN

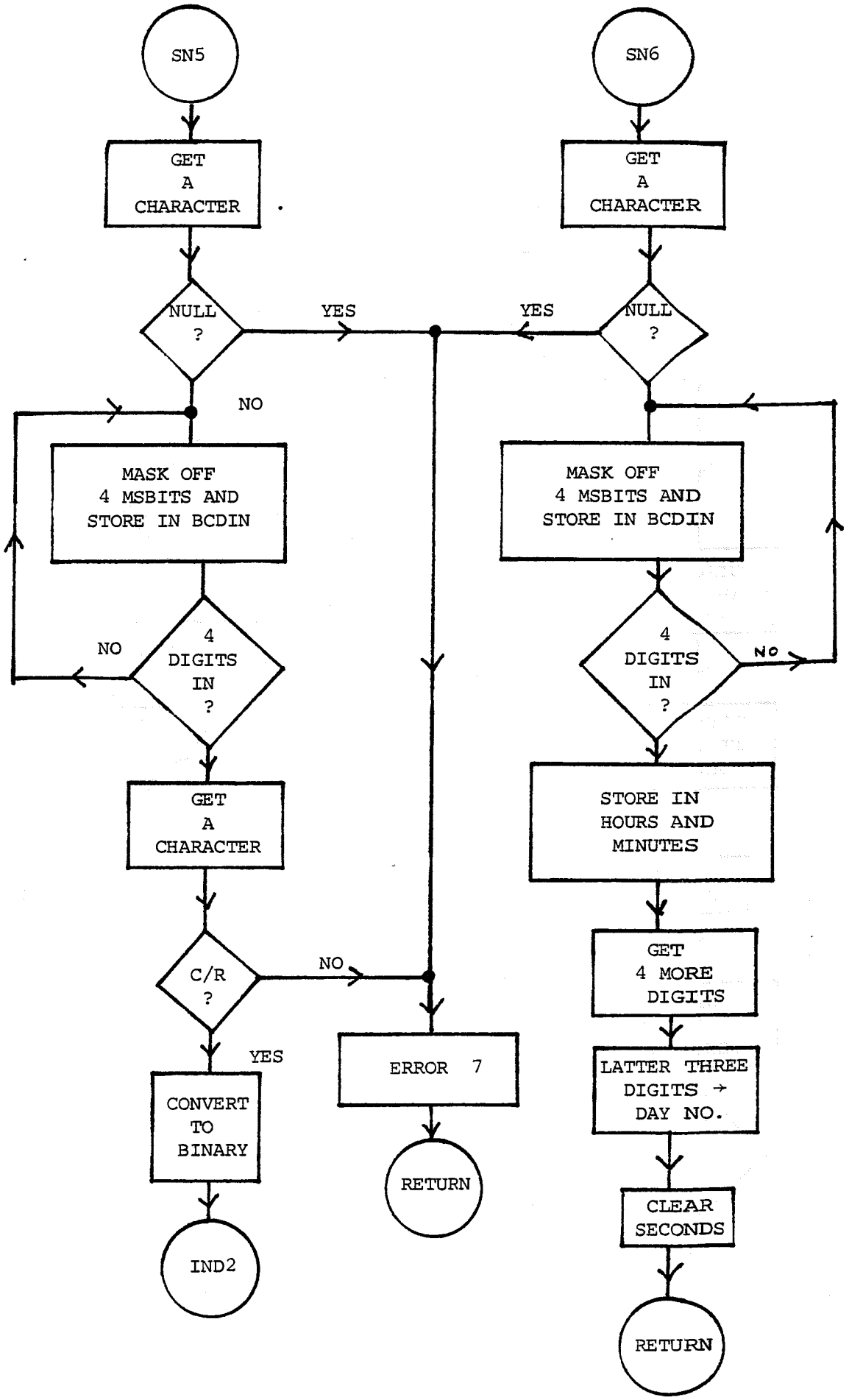
Raster Display Subroutine



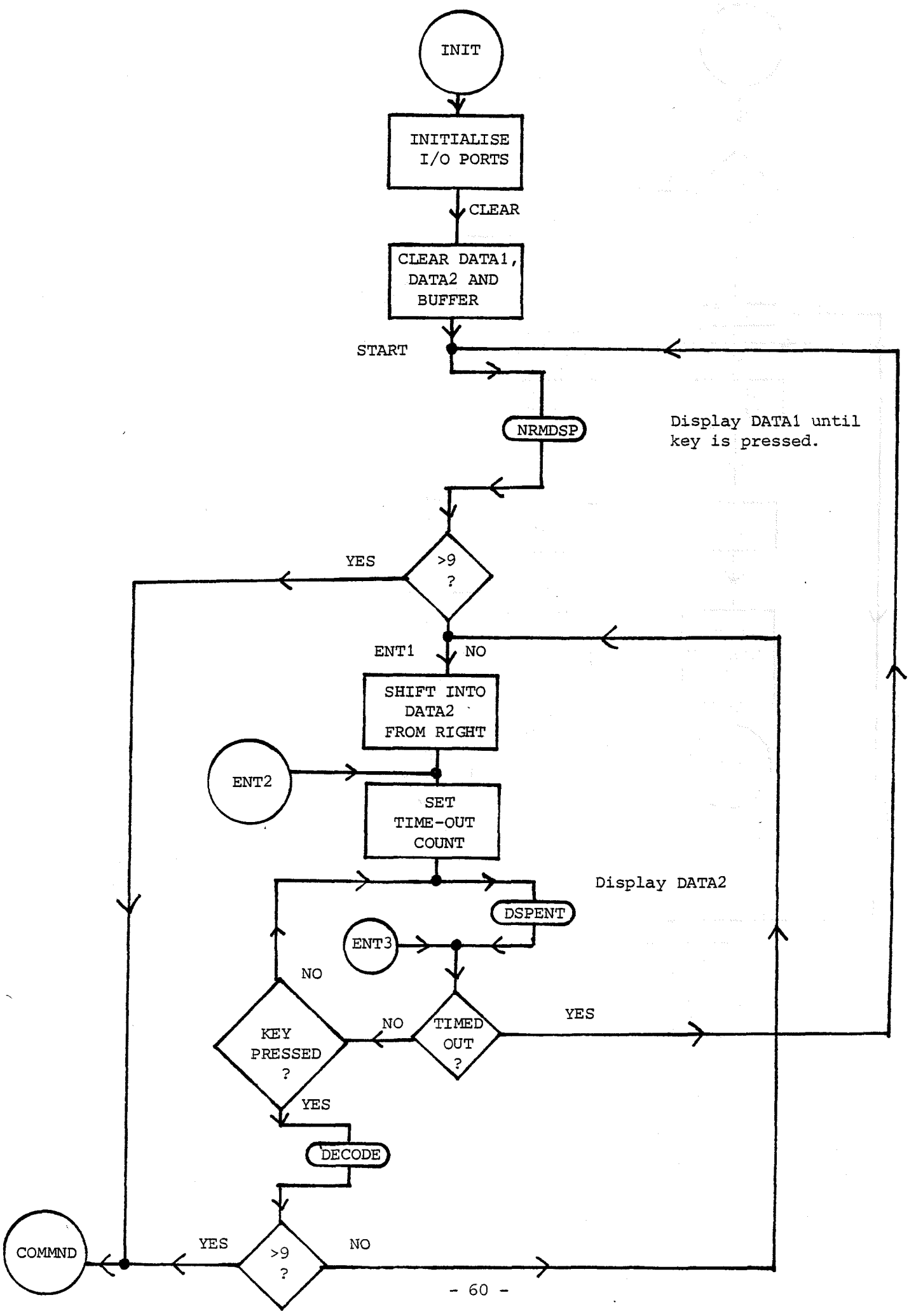
Characters

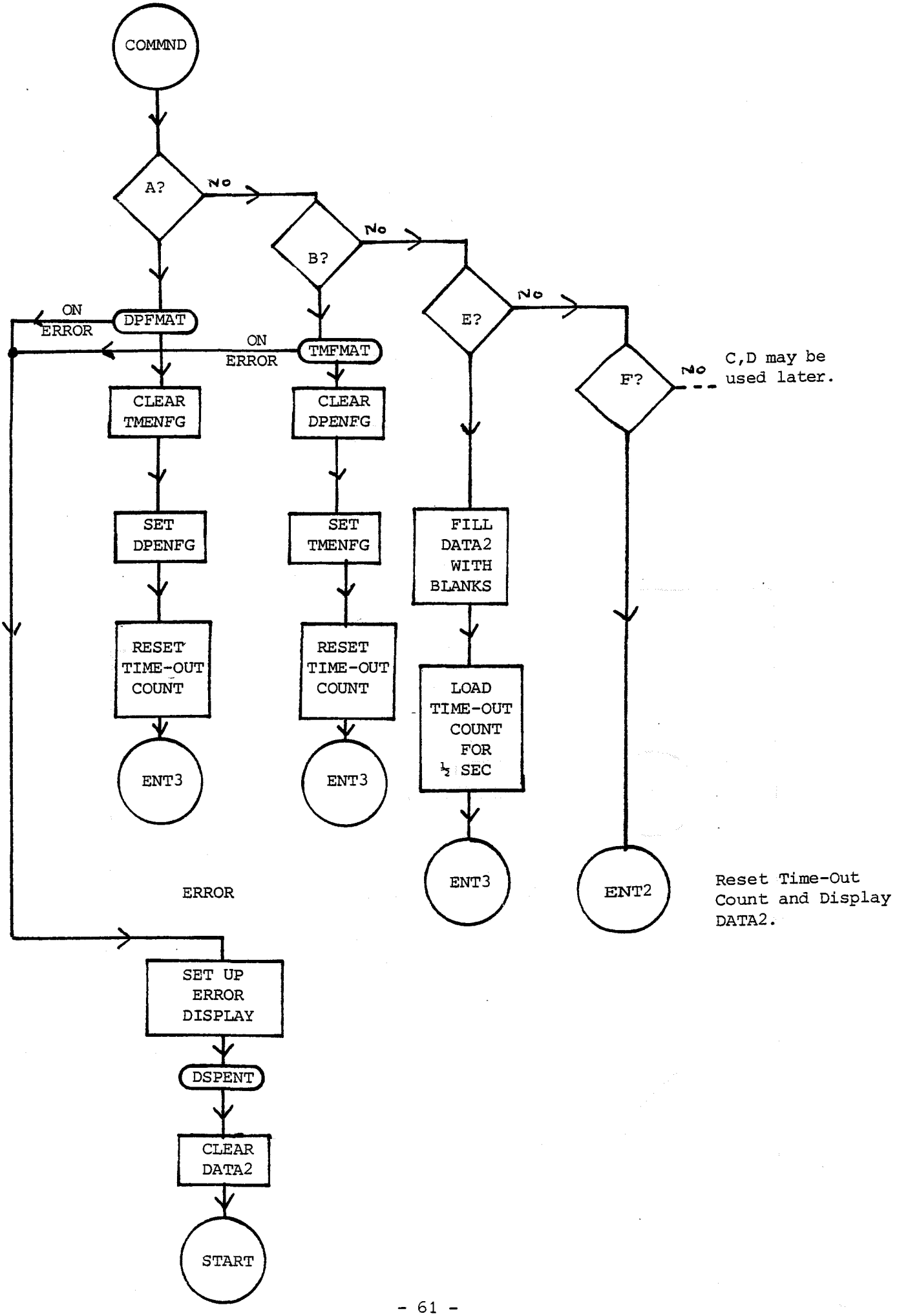
0, 1, 2, 3, 4, 5, 6, 7
8, 9, :, ;, <, =, >, Space.

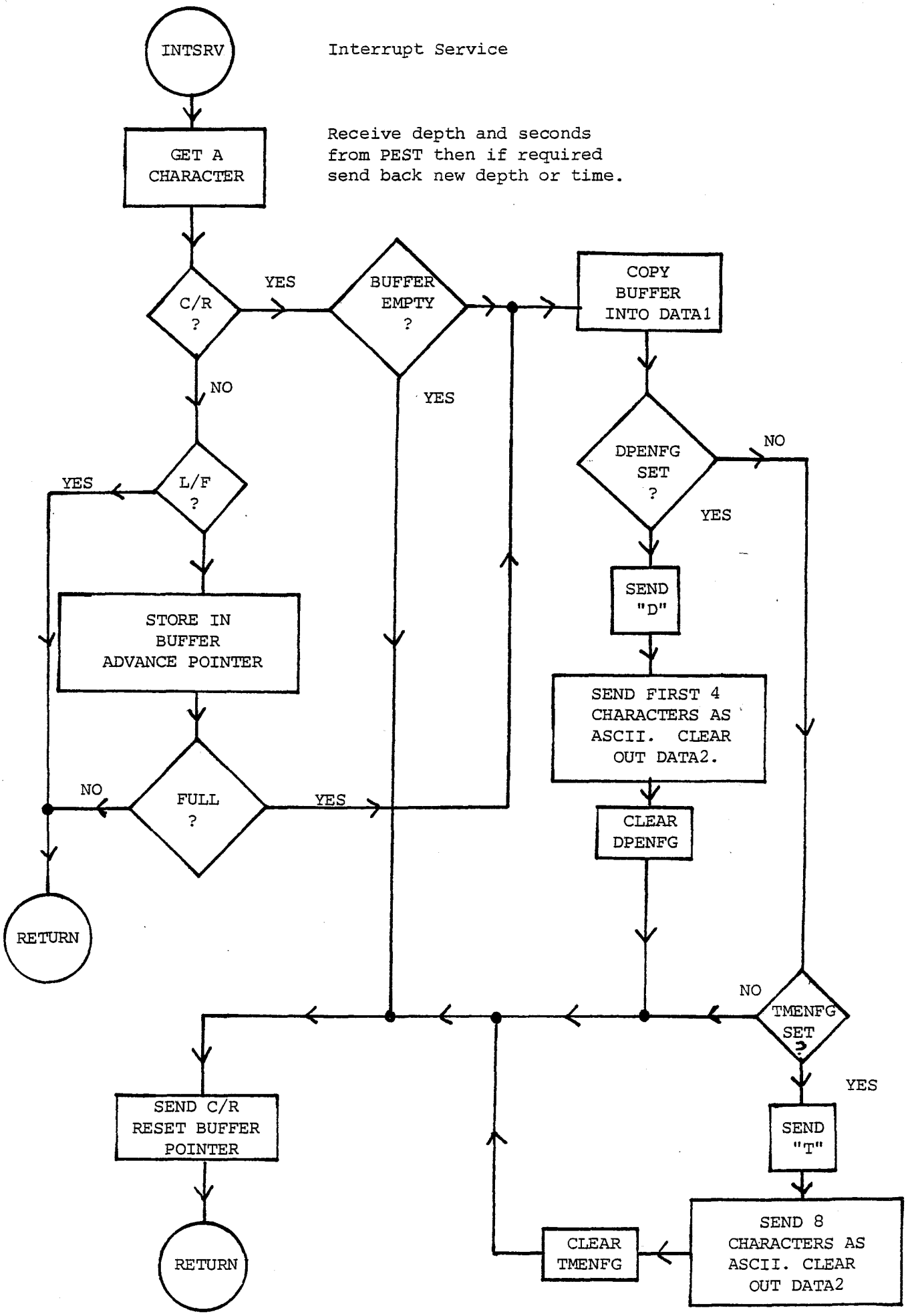

SEND
PORT NUMBER = 1
CLEAR TO SEND ?
YES
NO
YES
ER6CNT = 0 ?
YES
ERROR 6
INCREMENT ER6CNT
RETURN
OUTPUT 4 DEPTH DIGITS +??
OUTPUT 10s AND UNIT OF SECONDS
OUTPUT C/R
WAIT FOR RESPONSE
NO RESPONSE ?
YES
NO
C/R ?
YES
RETURN
NO
'D' ?
YES
SN5
NO
SN6


SN5
GET A CHARACTER
NULL ?
YES
NO
MASK OFF 4 MSBITS AND STORE IN BCDIN
4 DIGITS IN ?
NO
GET A CHARACTER
C/R ?
NO
YES
CONVERT TO BINARY
IND2
ERROR 7
RETURN
SN6
GET A CHARACTER
NULL ?
YES
MASK OFF 4 MSBITS AND STORE IN BCDIN
4 DIGITS IN ?
NO
STORE IN HOURS AND MINUTES
GET 4 MORE DIGITS
LATTER THREE DIGITS → DAY NO.
CLEAR SECONDS
RETURN

REMOTE


INIT
INITIALISE I/O PORTS
CLEAR
CLEAR DATA1, DATA2 AND BUFFER
START
NRMDSP
Display DATA1 until key is pressed.
YES
>9 ?
ENT1
NO
SHIFT INTO DATA2 FROM RIGHT
ENT2
SET TIME-OUT COUNT
Display DATA2
DSPENT
ENT3
NO
KEY PRESSED ?
NO
TIMED OUT ?
YES
YES
DECODE
COMMND
YES
>9 ?
NO


COMMND
A?
No
B?
No
E?
No
F?
No
C,D may be used later.
ON ERROR
DPFMAT
ON ERROR
TMFMAT
CLEAR TMENFG
CLEAR DPENFG
SET DPENFG
SET TMENFG
FILL DATA2 WITH BLANKS
RESET TIME-OUT COUNT
RESET TIME-OUT COUNT
LOAD TIME-OUT COUNT FOR ½ SEC
ENT3
ENT3
ENT3
ENT2
Reset Time-Out Count and Display DATA2.
ERROR
SET UP ERROR DISPLAY
DSPENT
CLEAR DATA2
START


INTSRV
Interrupt Service
GET A CHARACTER
Receive depth and seconds from PEST then if required send back new depth or time.
C/R ?
YES
BUFFER EMPTY ?
COPY BUFFER INTO DATA1
NO
YES
DPENFG SET ?
NO
YES
L/F ?
YES
SEND "D"
STORE IN BUFFER ADVANCE POINTER
SEND FIRST 4 CHARACTERS AS ASCII. CLEAR OUT DATA2.
NO
FULL ?
YES
CLEAR DPENFG
RETURN
NO
TMENFG SET ?
YES
SEND C/R RESET BUFFER POINTER
SEND "T"
RETURN
SEND 8 CHARACTERS AS ASCII. CLEAR OUT DATA2
CLEAR TMENFG

PROGRAM LISTINGS

```
1: *
2: *
3: *     P.E.S TRACKING AND DIGITISATION PROGRAM
4: *     ======================================
5: *
6: *     This program running on a motorola 6800 micro-processor
7: *     receives a zero-depth trigger from the precision echo-
8: *     sounder recorder and samples the audio-output during a
9: *     gating time. Time-varied and automatic gain control are
10: *    applied as apropriate to the block of data before echo
11: *    detection is attempted.
12: *
13: *    The "window" of data is weighted linearly - zero at the
14: *    ends and one at the middle. An algorithm is then called
15: *    to differentiate the signal by correlating it with a
16: *    ramp. Negative results are set to zero. The routine
17: *    'PEAK' returns with the position (relative to the start
18: *    of the window) of the largest peak and the peak nearest
19: *    to the position 'PREDIC'.
20: *
21: *    The selected peak position is then used to adjust the
22: *    pre-sampling delay to move the window in order to centre
23: *    it the echo.
24: *
25: *    While the program is running a new depth can be entered
26: *    to force the window to a new position. Every two minutes
27: *    depth and time are printed. Errors are printed as they
28: *    occur.
29: *
30: *    A raster display output of depth and time is available
31: *    for marking a chart recorder. The timing of the display
32: *    insures that the echo is not over-printed.
33: *
34: *    PROGRAM STRUCTURE
35: *    =================
36: *
37: *    a) Subroutines for initialisation, I/O and processing.
38: *    b) The interrupt routine.
39: *    c) The main program which sets and adjusts parameters
40: *       and dictates strategy. This program is appended to
41: *       the rest of the routines to so that it can be easily
42: *       modified during development.
43: *
```

```
                1:          NAM    PEST82
                2:          OPT    NOG,PAG
                3: *
                4: *I/O SPECIFICATIONS
                5: *MULT 8 bit by 8 bit multiplier/divider
                6: *
00FC            7: MULT     EQU    $FC
                8: *
                9: *ADC 12 bit single channel analogue to digital converter
               10: *     This is preceded by a FW rectifier, anti-aliassing
               11: *     filter and sample & hold chip.
               12: *
8004           13: ADC      EQU    $8004
               14: *
               15: *DAC 10 bit digital to analogue converter
               16: *
8008           17: DAC      EQU    $8008
               18: *
               19: *FRONT PANEL BCD SWITCH AND LCD DISPLAY
               20: *
8020           21: PANEL    EQU    $8020
               22: *
               23: *PRINTER ANADEX Parallel printer
               24: *
8021           25: PRINTR EQU    $8021
               26: *
               27: *TIMER 3 independent 16 bit counter/timers
               28: *     Used to divide down from 1 Mhz. to 1Khz. and 1hz.
               29: *     and count down pre-sampling delay.
               30: *
8040           31: TIMER    EQU    $8040
               32: *
               33: *ACIAS Serial port for control terminal.
               34: *
E002           35: ACIAS    EQU    $E002
               36: *
0200           37: MEMBEG EQU    $200          Start of available data storage
               38: *                           memory
1000           39: DPSTRT EQU    $1000         Start of 2 sec. depth storage
2000           40: DPTMST EQU    $2000         Start of buffer for 2 min.
               41: *                           depths.
CF40           42: PATRNS EQU    $CF40
CFC0           43: COEFFS EQU    $CFC0         Depth filter coefficients
A000           44: IOVECT EQU    $A000         I/O interrupt vectoring address in
               45: *                           MINIBUG
               46: *BASE PAGE ALLOCATIONS
               47: *
0058           48:          ORG    $58
0058           49: ER6CNT RMB    1
0059           50: MUFDSP RMB    10            Mufax display buffer
0063           51: XSTOR    RMB    2
0065           52: DISPEN RMB    1             Mufax display enable flag
0066           53: DSFRST RMB    1             Display LHS or RHS flag
0067           54: DISPDN RMB    1             Display done flag
0068           55: CHARPT RMB    2             Character pointer
006A           56: ROWPT    RMB    2          Dot matrix row pointer
006C           57: DISPMD RMB    1             Display or sample mode flag
006D           58: DOTS     RMB    1          Dot mask
006E           59: LINES    RMB    1          Number of line repeats
```

```
006F            60: WIDTH   RMB   1       Number of dot repeats
0070            61: THRHLD  RMB   1       Threshold for TX test.
0071            62: SRCHCT  RMB   1       Number of pending depths sent.
0072            63: RVBSIZ  RMB   2       No. of reverb. samples.
0074            64: DPTMP1  RMB   2       Pointers for stacking 2 min.
0076            65: DPTMP2  RMB   2       depths and times.
0078            66: FILTOP  RMB   4       Depth filter accumulator
007C            67: SIGN    RMB   1       Sign indicator for DPFILT
007D            68: PNTR1   RMB   2       Coeff pointer for DPFILT
007F            69: PNTR2   RMB   2       Depth pointer for DPFILT
0081            70: DIRFLG  RMB   1
0082            71: DPSTCK  RMB   2       Pointer to newest depth
                72: *                     in depth stack
0084            73: LSTENT  RMB   2       Most recent manually entered depth
0086            74: BCDIN   RMB   8       8 unpacked bcd digits. 4 entered
                75: *                     from front panel and 8 for display
008E            76: TXNO    RMB   1       Tx number in gating sequence
008F            77: LOOPCT  RMB   1       Count of number of passes while no
                78: *                     TX.
0090            79: SCTEMP  RMB   1       Temp for AGC gain value
0091            80: NRSTHT  RMB   1       Value of peak nearest to PREDICtion.
0092            81: NRSTPS  RMB   2       Position of above.
0094            82: PREDIC  RMB   2       Predicted echo position. Normally
                83: *                     gate centre.
0096            84: SAVX    RMB   2       Temp for Index reg.
0098            85: OFFSET  RMB   2       Difference between echo position and
                86: *                     PREDICtion.
009A            87: PKVAL   RMB   1       Value of greatest peak.
009B            88: PKPOSN  RMB   2       Position of above.
009D            89: FIRSTX  RMB   2       Pointer to start of data array.
009F            90: LASTX   RMB   2       Pointer to end of data array.
00A1            91: SK      RMB   2       Accumulator used by EDGE.
00A3            92: SUMK    RMB   2       Accumulator used by EDGE.
00A5            93: EGEND   RMB   2       Pointer to last value used by EDGE.
00A7            94: PHTEMP  RMB   1       Temp for phase.
00A8            95: FLSTRG  RMB   1       Flag to indicate extra trigger.
00A9            96: RVRBCT  RMB   2       Decremented reverb. samples
                97: *                     while testing for presence of TX.
00AB            98: TXSAMP  RMB   2       Accumulator for samples during TX.
00AD            99: SMDONE  RMB   1       Hand-shake flag between main prog
               100: *                     and interrupt service.
               101: *                     SET by MAIN to enable sampling.
               102: *                     CLEARED by interrupt service
               103: *                     when sampling over.
00AE           104: FRSTTX  RMB   2       TXSAMP after first attempt to find
               105: *                     TX.
00B0           106: SCNDTX  RMB   2       TXSAMP after second attempt to find
               107: *                     TX.
00B2           108: GATNFG  RMB   1       Gating on flag.
00B3           109: COEF1   RMB   2       First coefficient for FILTer.
00B5           110: COEF2   RMB   2       Second coefficient for FILTer.
00B7           111: AVDPTH  RMB   2       FILTer accumulator.
00B9           112: ATEMP   RMB   1       Temp for acc A.
00BA           113: DELAY   RMB   2       Presampling delay. 1ms. units.
00BC           114: FLAGS   RMB   1       Interrupt routine enable mask.
00BD           115: TRGFLG  RMB   1       Set by occurrence of trigger.
00BE           116: DATSIZ  RMB   2       Size of data sampling window.
00C0           117: DATCNT  RMB   2       Counted down to zero during sampling
00C2           118: OVFLW   RMB   2       Number of overloads during sampling.
00C4           119: SCALE   RMB   1       AGC gain. Binary gain steps.
```

```
00C5            120: DESTAD RMB   2          Destination address of sampled data.
00C7            121: HRFLG  RMB   1          Set after printing hour headings.
00C8            122: ODDMIN RMB   1          Set to cause 2 min. filtering
00C9            123: EVNMIN RMB   1          Clear after printing 2 minute depth.
00CA            124: NWDPTH RMB   2          Up to date depth.
00CC            125: PHASE  RMB   1          Present depth phase.
00CD            126: SLOPE  RMB   1          Gradient of bottom. Samples/sweep.
00CE            127: ERRORS RMB   7          Error signalling flags.
00D5            128: XT     RMB   2          Temp for X reg.
00D7            129: DIG    RMB   2          Location for 4 packed BCD.
00D9            130: BTEMP  RMB   1          Temp for acc B.
00DA            131: DIGITS RMB   6          Ascii digit string and terminator.
00E0            132: YEARS  RMB   2          BCD year.
00E2            133: DAYS   RMB   2          BCD day number.
00E4            134: HRS    RMB   2          BCD hour.
00E6            135: MINS   RMB   2          BCD minutes.
00E8            136: SECS   RMB   2          BCD seconds.
00EA            137:        RMB   10         Clock constants.
00F4            138: XTEMP  RMB   2          Temp for X reg.
00F6            139: SPTEMP RMB   2          Temp for Stack pointer.
00F8            140:        RMB   4
                141: *MULT appears here.
                142: *
                143: *
                144: *      SUBROUTINES
                145: *      ===========
                146: *        INITIALLISE I/O
                147: *
C000            148:        ORG   $C000
                149: *
C000 0F         150: INIT   SEI
C001 7F 8004    151:        CLR   ADC          8 bits in 4 lsbits from ADC
C004 7F 8005    152:        CLR   ADC+1        8 bits in from ADC.
C007 CE 0404    153:        LDX   #$0404       Select data registers.
C00A FF 8006    154:        STX   ADC+2
C00D CE FFFF    155:        LDX   #$FFFF       8 bits to DAC B side.
C010 FF 8008    156:        STX   DAC          2 bits to DAC and 6 logic drives.
                157: *Initialise TIMER
C013 CE 8040    158:        LDX   #TIMER
C016 86 36      159:        LDA A #$36         Select Counter 0
C018 A7 03      160:        STA A 3,X          Binary count. Load 2 bytes.
C01A 86 E8      161:        LDA A #$E8         Low byte first.
C01C A7 00      162:        STA A 0,X
C01E 86 03      163:        LDA A #3           High byte.
C020 A7 00      164:        STA A 0,X          Divide by 1000 from 1Mhz.
C022 86 70      165:        LDA A #$70         Select Counter 1
C024 A7 03      166:        STA A 3,X          Binary count. Load 2 bytes.
C026 86 E8      167:        LDA A #$E8         Low byte first.
C028 A7 01      168:        STA A 1,X          Divide by 1000 from 1Khz.
C02A 86 03      169:        LDA A #3           High byte.
C02C A7 01      170:        STA A 1,X          O/P = 1 sec. pulse for clock.
C02E 86 B0      171:        LDA A #$B0         Prepare counter 2
C030 A7 03      172:        STA A 3,X          for 2 byte load.
C032 CE 1C07    173:        LDX   #$1C07       CA2 +ve edge 1 sec pulse
C035 FF 800A    174:        STX   DAC+2        CB1 +ve edge trigger
C038 86 10      175:        LDA A #$10         Negative logic MUFAX time mark
C03A B7 8008    176:        STA A DAC          initialised high.
                177: *Initialise communications
C03D 86 7F      178:        LDA A #%01111111   7bits out to PRINTER
C03F B7 8021    179:        STA A PRINTR       Msb handshake from PRINTER
```

```
C042 86 2C     180:          LDA A #$2C
C044 B7 8023   181:          STA A PRINTR+2     CA2 1 us strobe pulse
               182: *Copy CLOCK constants into base page.
C047 9F F6     183:          STS   SPTEMP       N.B.interrupts are disabled
C049 8E C05A   184:          LDS   #I2-1
C04C C6 14     185:          LDA B #20
C04E CE 00E0   186:          LDX   #YEARS
C051 32        187: I1       PUL A
C052 A7 00     188:          STA A 0,X
C054 08        189:          INX
C055 5A        190:          DEC B
C056 26 F9     191:          BNE   I1
C058 9E F6     192:          LDS   SPTEMP
C05A 39        193:          RTS
C05B 19 82     194: I2       FDB   $1982
C05D 00 01     195:          FDB   1
C05F 00 00     196:          FDB   0
C061 00 00     197:          FDB   0
C063 00 00     198:          FDB   0
C065 99 99     199:          FDB   $9999,$366,$24,$60,$60
               200: *
               201: *Print a character.
               202: *
C06F DF F4     203: PRINTC STX   XTEMP
C071 CE 86B2   204: P5     LDX   #$86B2
C074 7D 8021   205: P1     TST   PRINTR       Printer busy ?
C077 2B 06     206:        BMI   P2           No
C079 09        207:        DEX                Decrement count
C07A 26 F8     208:        BNE   P1           Try again
C07C 0D        209:        SEC                Still not ready
C07D 20 0D     210:        BRA   P3           Timed-out return with carry set
C07F B7 8021   211: P2     STA A PRINTR       O/P character
C082 81 0D     212:        CMP A #$D          Was it C/R ?
C084 26 06     213:        BNE   P3           No
C086 86 0A     214:        LDA A #$A          Yes. Send L/F as well.
C088 8D E7     215:        BSR   P5           Call again
C08A 86 0D     216:        LDA A #$D          Return with C/R
C08C DE F4     217: P3     LDX   XTEMP
C08E 39        218:        RTS
               219: *
               220: *Print a string up to EOT.
               221: *
C08F 8D DE     222: P4     BSR   PRINTC
C091 08        223:        INX
C092 A6 00     224: PRINTS LDA A 0,X          Enter here
C094 81 04     225:        CMP A #4           EOT?
C096 26 F7     226:        BNE   P4
C098 39        227:        RTS
               228: *
               229: *Input a character from control port.
               230: *
C099 B6 E002   231: INCH   LDA A ACIAS
C09C 47        232:        ASR A              RX reg. full?
C09D 24 FA     233:        BCC   INCH         No  Not yet
C09F B6 E003   234:        LDA A ACIAS+1      I/P character.
C0A2 84 7F     235:        AND A #$7F         Mask off parity
               236: *
               237: *Output a character to control port.
               238: *
C0A4 37        239: OUTCH  PSH B
```

```
C0A5 F6 E002  240: OUTC1  LDA B ACIAS
C0A8 57       241:        ASR B              TX reg. empty?
C0A9 57       242:        ASR B
C0AA 24 F9    243:        BCC   OUTC1
C0AC B7 E003  244:        STA A ACIAS+1      O/P character
C0AF 33       245:        PUL B
C0B0 39       246:        RTS
              247: *
              248: *Get a charcter with time-out
              249: *
C0B1 37       250: GET    PSH B
C0B2 C6 00    251:        LDA B #0
C0B4 B6 E002  252: G1     LDA A ACIAS
C0B7 47       253:        ASR A
C0B8 25 06    254:        BCS   G2
C0BA 5A       255:        DEC B
C0BB 26 F7    256:        BNE   G1
C0BD 4F       257:        CLR A
C0BE 20 05    258:        BRA   G3
C0C0 B6 E003  259: G2     LDA A ACIAS+1
C0C3 84 7F    260:        AND A #$7F
C0C5 33       261: G3     PUL B
C0C6 39       262:        RTS
              263: *
              264: *Output hex.
              265: *
C0C7 44       266: OUTHL  LSR A              O/P left nibble
C0C8 44       267:        LSR A
C0C9 44       268:        LSR A
C0CA 44       269:        LSR A
C0CB 84 0F    270: OUTHR  AND A #$F          O/P right nibble.
C0CD 8B 30    271:        ADD A #'0          Make ASCII
C0CF 81 39    272:        CMP A #'9
C0D1 23 02    273:        BLS   OUT          0 to 9
C0D3 8B 07    274:        ADD A #7           A to F
C0D5 20 98    275: OUT    BRA   PRINTC       Print then return.
C0D7 A6 00    276: OUT2H  LDA A 0,X          O/P byte in HEX
C0D9 8D EC    277:        BSR   OUTHL
C0DB A6 00    278:        LDA A 0,X
C0DD 08       279:        INX
C0DE 20 EB    280:        BRA   OUTHR        Print then return.
C0E0 8D F5    281: OUT4HS BSR   OUT2H        O/P 2 bytes + space.
C0E2 8D F3    282: OUT2HS BSR   OUT2H        O/P 1 byte + space.
C0E4 86 20    283: OUTS   LDA A #$20         O/P space/.
C0E6 20 ED    284:        BRA   OUT          and then return.
              285: *
              286: *Update time.
              287: *
C0E8 37       288: CLOCK  PSH B
C0E9 36       289:        PSH A
C0EA D6 E9    290:        LDA B SECS+1
C0EC 54       291:        LSR B
C0ED 25 18    292:        BCS   TT1
C0EF D6 CA    293:        LDA B NWDPTH       Stack depth every even second.
C0F1 96 CB    294:        LDA A NWDPTH+1
C0F3 DE 82    295:        LDX   DPSTCK
C0F5 E7 00    296:        STA B 0,X
C0F7 A7 01    297:        STA A 1,X
C0F9 08       298:        INX
C0FA 08       299:        INX
```

```
C0FB DF 82     300:          STX    DPSTCK
C0FD 8C 1080   301:          CPX    #4*31+DPSTRT+4
C100 26 05     302:          BNE    TT1
C102 CE 1000   303:          LDX    #DPSTRT
C105 DF 82     304:          STX    DPSTCK
C107 C6 05     305: TT1      LDA B  #5
C109 CE 00E8   306:          LDX    #SECS
C10C A6 01     307: T1       LDA A  1,X
C10E 8B 01     308:          ADD A  #1
C110 19        309:          DAA
C111 A7 01     310:          STA A  1,X
C113 24 02     311:          BCC    T2
C115 6C 00     312:          INC    0,X
C117 A1 0B     313: T2       CMP A  11,X
C119 25 0F     314:          BCS    T3
C11B A6 00     315:          LDA A  0,X
C11D A1 0A     316:          CMP A  10,X
C11F 25 09     317:          BCS    T3
C121 6F 00     318:          CLR    0,X
C123 6F 01     319:          CLR    1,X
C125 09        320:          DEX
C126 09        321:          DEX
C127 5A        322:          DEC B
C128 26 E2     323:          BNE    T1
C12A 7D 00E3   324: T3       TST    DAYS+1        Set day 000 to day 001
C12D 26 08     325:          BNE    T4
C12F 7D 00E2   326:          TST    DAYS
C132 26 03     327:          BNE    T4
C134 7C 00E3   328:          INC    DAYS+1
C137 32        329: T4       PUL A
C138 33        330:          PUL B
C139 39        331:          RTS
               332: *
               333: *Convert double byte in X reg. to BCD.
               334: *as ASCII in DIGITS + EOT.
               335: *
C13A DF F4     336: CVBTD    STX    XTEMP
C13C 96 F4     337:          LDA A  XTEMP
C13E D6 F5     338:          LDA B  XTEMP+1
C140 CE 00DA   339:          LDX    #DIGITS
C143 DF F4     340:          STX    XTEMP
C145 CE C17A   341:          LDX    #CONST
C148 7F 00D9   342: CVDEC1   CLR    BTEMP
C14B E0 01     343: CVDEC2   SUB B  1,X
C14D A2 00     344:          SBC A  0,X
C14F 25 05     345:          BCS    CVDEC3
C151 7C 00D9   346:          INC    BTEMP
C154 20 F5     347:          BRA    CVDEC2
C156 EB 01     348: CVDEC3   ADD B  1,X
C158 A9 00     349:          ADC A  0,X
C15A 36        350:          PSH A
C15B DF F6     351:          STX    SPTEMP
C15D DE F4     352:          LDX    XTEMP
C15F 96 D9     353:          LDA A  BTEMP
C161 8B 30     354:          ADD A  #'0
C163 A7 00     355:          STA A  0,X
C165 32        356:          PUL A
C166 08        357:          INX
C167 DF F4     358:          STX    XTEMP
C169 DE F6     359:          LDX    SPTEMP
```

```
C16B 08        360:          INX
C16C 08        361:          INX
C16D 8C C184   362:          CPX    #CONST+10
C170 26 D6     363:          BNE    CVDEC1
C172 86 04     364:          LDA A  #4            Put EOT on the end.
C174 97 DF     365:          STA A  DIGITS+5
C176 CE 00DA   366:          LDX    #DIGITS
C179 39        367:          RTS
C17A 27 10     368: CONST    FDB    10000,1000,100,10,1
               369: *
               370: *I/P up to 4 BCD digits.
               371: *
C184 7F 00D7   372: INDEC    CLR    DIG
C187 7F 00D8   373:          CLR    DIG+1
C18A 8D 21     374: DG3      BSR    INDIG
C18C 24 01     375:          BCC    DG4            Is it a digit ?
C18E 39        376:          RTS                   No
C18F 78 00D8   377: DG4      ASL    DIG+1
C192 79 00D7   378:          ROL    DIG
C195 78 00D8   379:          ASL    DIG+1
C198 79 00D7   380:          ROL    DIG
C19B 78 00D8   381:          ASL    DIG+1
C19E 79 00D7   382:          ROL    DIG
C1A1 78 00D8   383:          ASL    DIG+1
C1A4 79 00D7   384:          ROL    DIG
C1A7 9A D8     385:          ORA A  DIG+1
C1A9 97 D8     386:          STA A  DIG+1
C1AB 20 DD     387:          BRA    DG3
               388: *
               389: *I/P one digit.
               390: *
C1AD BD C099   391: INDIG    JSR    INCH
C1B0 80 30     392:          SUB A  #'0
C1B2 2B 04     393:          BMI    DG1
C1B4 81 09     394:          CMP A  #9
C1B6 2F 02     395:          BLE    DG2
C1B8 0D        396: DG1      SEC                   Not digit. Return with carry set.
C1B9 39        397:          RTS
C1BA 0C        398: DG2      CLC                   Digit. Return with carry clear.
C1BB 39        399:          RTS
               400: *
               401: *Send a string to console.
               402: *
C1BC BD C0A4   403: TP1      JSR    OUTCH
C1BF 08        404:          INX
C1C0 A6 00     405: TYPES    LDA A  0,X           Enter here.
C1C2 81 04     406:          CMP A  #4            EOT ?
C1C4 26 F6     407:          BNE    TP1
C1C6 39        408:          RTS
               409: *
               410: *Print heading on the hour.
               411: *
C1C7 7D 00C7   412: HRHEAD   TST    HRFLG
C1CA 27 0D     413:          BEQ    HD2
C1CC D6 E7     414:          LDA B  MINS+1
C1CE 26 06     415:          BNE    HD1
C1D0 8D 0C     416:          BSR    HEADNG
C1D2 7F 00C7   417:          CLR    HRFLG
C1D5 39        418:          RTS
C1D6 D7 C7     419: HD1      STA B  HRFLG
```

```
C1D8 39        420:          RTS
C1D9 D6 E7     421: HD2      LDA B MINS+1
C1DB 26 F9     422:          BNE   HD1
C1DD 39        423:          RTS
               424: *
C1DE CE C1F7   425: HEADNG   LDX   #HEAD1
C1E1 BD C092   426:          JSR   PRINTS
C1E4 CE 00E2   427:          LDX   #DAYS
C1E7 BD C0E0   428:          JSR   OUT4HS
C1EA CE 00E0   429:          LDX   #YEARS
C1ED BD C0E0   430:          JSR   OUT4HS
C1F0 CE C21A   431:          LDX   #HEAD2
C1F3 BD C092   432:          JSR   PRINTS
C1F6 39        433:          RTS
C1F7 0A        434: HEAD1    FCB   $A
C1F8 20        435:          FCC   " PEST82 REV1. C.G.FLEWELLEN 1982"
C218 0D        436:          FCB   $D,4
C21A 0D        437: HEAD2    FCB   $D,$A
C21C 20        438:          FCC   "  TIME   DEPTH PH  SLOPE"
C234 0D        439:          FCB   $D,4
               440: *
               441: *Print DEPTH every two minutes.
               442: *
C236 7D 00C9   443: TWOMIN   TST   EVNMIN
C239 27 0F     444:          BEQ   PD2
C23B D6 E7     445:          LDA B MINS+1
C23D 54        446:          LSR B
C23E 25 06     447:          BCS   PD1
C240 8D 13     448:          BSR   PRDAT
C242 7F 00C9   449:          CLR   EVNMIN
C245 39        450:          RTS
C246 79 00C9   451: PD1      ROL   EVNMIN
C249 39        452:          RTS
C24A D6 E7     453: PD2      LDA B MINS+1
C24C 54        454:          LSR B
C24D 24 05     455:          BCC   PD3
C24F 79 00C8   456:          ROL   ODDMIN
C252 20 F2     457:          BRA   PD1
C254 39        458: PD3      RTS
               459: *
C255 BD C1C7   460: PRDAT    JSR   HRHEAD
C258 CE 00E5   461:          LDX   #HRS+1
C25B BD C0E2   462:          JSR   OUT2HS
C25E CE 00E7   463:          LDX   #MINS+1
C261 BD C0E2   464:          JSR   OUT2HS
C264 CE 00E9   465:          LDX   #SECS+1
C267 BD C0E2   466:          JSR   OUT2HS
C26A 96 CA     467:          LDA A NWDPTH
C26C D6 CB     468:          LDA B NWDPTH+1
C26E 58        469:          ASL B
C26F 49        470:          ROL A
C270 DB CB     471:          ADD B NWDPTH+1
C272 99 CA     472:          ADC A NWDPTH
C274 44        473:          LSR A
C275 56        474:          ROR B
C276 44        475:          LSR A
C277 56        476:          ROR B
C278 97 F4     477:          STA A XTEMP
C27A D7 F5     478:          STA B XTEMP+1
C27C DE F4     479:          LDX   XTEMP
```

```
C27E BD C13A  480:         JSR   CVBTD
C281 BD C092  481:         JSR   PRINTS
C284 BD C0E4  482:         JSR   OUTS
C287 CE 00CC  483:         LDX   #PHASE
C28A BD C0E2  484:         JSR   OUT2HS
C28D 86 20    485:         LDA A #$20
C28F D6 CD    486:         LDA B SLOPE
C291 2A 03    487:         BPL   PD4
C293 50       488:         NEG B
C294 8B 0D    489:         ADD A #$D
C296 BD C06F  490: PD4     JSR   PRINTC
C299 17       491:         TBA
C29A BD C0CB  492:         JSR   OUTHR
C29D 86 0D    493:         LDA A #$D
C29F BD C06F  494:         JSR   PRINTC
C2A2 39       495:         RTS
              496: *
              497: *Print error messages.
              498: *
C2A3 CE C350  499: ERRS    LDX   #ERMESS
C2A6 DF F6    500:         STX   SPTEMP
C2A8 CE 00CE  501:         LDX   #ERRORS
C2AB A6 00    502: ER1     LDA A 0,X
C2AD 27 0D    503:         BEQ   ER2
C2AF 6F 00    504:         CLR   0,X
C2B1 DF D5    505:         STX   XT
C2B3 DE F6    506:         LDX   SPTEMP
C2B5 EE 00    507:         LDX   0,X
C2B7 BD C092  508:         JSR   PRINTS
C2BA DE D5    509:         LDX   XT
C2BC 96 F7    510: ER2     LDA A SPTEMP+1
C2BE 8B 02    511:         ADD A #2
C2C0 24 03    512:         BCC   ER3
C2C2 7C 00F6  513:         INC   SPTEMP
C2C5 97 F7    514: ER3     STA A SPTEMP+1
C2C7 08       515:         INX
C2C8 8C 00D5  516:         CPX   #ERRORS+7
C2CB 26 DE    517:         BNE   ER1
C2CD 39       518:         RTS
              519: *
              520: *Get time from K/B
              521: *
C2CE CE C307  522: GETTM   LDX   #MESS1
C2D1 BD C1C0  523:         JSR   TYPES
C2D4 BD C184  524:         JSR   INDEC
C2D7 81 EB    525:         CMP A #$EB          Esc - $30
C2D9 27 F3    526:         BEQ   GETTM
C2DB DE D7    527:         LDX   DIG
C2DD DF E4    528:         STX   HRS
C2DF BD C184  529:         JSR   INDEC
C2E2 81 EB    530:         CMP A #$EB
C2E4 27 E8    531:         BEQ   GETTM
C2E6 DE D7    532:         LDX   DIG
C2E8 DF E6    533:         STX   MINS
C2EA BD C184  534:         JSR   INDEC
C2ED 81 EB    535:         CMP A #$EB
C2EF 27 DD    536:         BEQ   GETTM
C2F1 DE D7    537:         LDX   DIG
C2F3 DF E2    538:         STX   DAYS
C2F5 BD C184  539:         JSR   INDEC
```

```
C2F8 81 EB     540:          CMP A #$EB
C2FA 27 D2     541:          BEQ   GETTM
C2FC DE D7     542:          LDX   DIG
C2FE DF E0     543:          STX   YEARS
C300 CE C343   544:          LDX   #MESS2
C303 BD C1C0   545:          JSR   TYPES
C306 39        546:          RTS
C307 0D        547: MESS1    FCB   $D,$A
C309 54        548:          FCC   "TIME PLEASE (HRS/MINS/DAY/YEAR) "
C329 50        549:          FCC   "PRESS ESCAPE TO RESTART"
C340 0D        550:          FCB   $D,$A,4
C343 0D        551: MESS2    FCB   $D,$A
C345 54        552:          FCC   "THANKYOU"
C34D 0D        553:          FCB   $D,$A,4
               554: *
C350 C3 5E     555: ERMESS   FDB   ERR1,ERR2,ERR3,ERR4,ERR5,ERR6,ERR7
C35E 21        556: ERR1     FCC   "!! EXTRA TRIG."
C36C 0D        557:          FCB   $D,4
C36E 21        558: ERR2     FCC   "!! OVER SAMPLING"
C37E 0D        559:          FCB   $D,4
C380 21        560: ERR3     FCC   "!! OVER LOADING"
C38F 0D        561:          FCB   $D,4
C391 21        562: ERR4     FCC   "!! PHASE IS WRONG"
C3A2 0D        563:          FCB   $D,4
C3A4 21        564: ERR5     FCC   "!! NO TX"
C3AC 0D        565:          FCB   $D,4
C3AE 21        566: ERR6     FCC   "!! CAN'T XMIT"
C3BB 0D        567:          FCB   $D,4
C3BD 21        568: ERR7     FCC   "!! RX ERROR"
C3C8 0D        569:          FCB   $D,4
               570: *
               571: *Add DATSIZ to X reg.
               572: *
C3CA DF F4     573: ADDX     STX   XTEMP
C3CC 96 F5     574:          LDA A XTEMP+1
C3CE D6 F4     575:          LDA B XTEMP
C3D0 9B BF     576:          ADD A DATSIZ+1
C3D2 D9 BE     577:          ADC B DATSIZ
C3D4 97 F5     578:          STA A XTEMP+1
C3D6 D7 F4     579:          STA B XTEMP
C3D8 DE F4     580:          LDX   XTEMP
C3DA 39        581:          RTS
               582: *
               583: *Add acc A to X reg.
               584: *
C3DB 37        585: ADDA     PSH B
C3DC 5F        586:          CLR B
C3DD 4D        587:          TST A
C3DE 2A 01     588:          BPL   ADBA
C3E0 5A        589:          DEC B
C3E1 DF 63     590: ADBA     STX   XSTOR
C3E3 9B 64     591:          ADD A XSTOR+1
C3E5 D9 63     592:          ADC B XSTOR
C3E7 97 64     593:          STA A XSTOR+1
C3E9 D7 63     594:          STA B XSTOR
C3EB 33        595:          PUL B
C3EC DE 63     596:          LDX   XSTOR
C3EE 39        597:          RTS
               598: *
               599: *INTERRUPT SERVICE ROUTINE
```

```
                600: *
C3EF 4F         601: INT     CLR A
C3F0 F6 8006    602:         LDA B ADC+2        Test for EOC
C3F3 58         603:         ASL B
C3F4 49         604:         ROL A
C3F5 F6 800A    605:         LDA B DAC+2        Test for 1 sec. and timer.
C3F8 7D 8008    606:         TST   DAC          Clear timer interrupt flag.
C3FB 58         607:         ASL B
C3FC 49         608:         ROL A
C3FD 58         609:         ASL B
C3FE 49         610:         ROL A
C3FF F6 800B    611:         LDA B DAC+3        Test for trigger.
C402 58         612:         ASL B
C403 49         613:         ROL A
C404 D6 BC      614:         LDA B FLAGS
C406 F7 8008    615:         STA B DAC          O/P to LEDs
C409 94 BC      616:         AND A FLAGS        Mask-off interrupts not enabled.
C40B 26 01      617:         BNE   JO
C40D 3B         618:         RTI                Return from interrupt.
C40E 44         619: JO      LSR A
C40F 24 02      620:         BCC   J1
C411 20 44      621:         BRA   TRIG         Was trigger interrupt.
C413 44         622: J1      LSR A
C414 24 35      623:         BCC   J2
C416 36         624:         PSH A
                625: *Load clock timer for 1 Sec.
C417 C6 E8      626:         LDA B #$E8
C419 F7 8041    627:         STA B TIMER+1
C41C C6 03      628:         LDA B #3
C41E F7 8041    629:         STA B TIMER+1
C421 7F 006C    630:         CLR   DISPMD
C424 7D 0065    631:         TST   DISPEN
C427 27 1E      632:         BEQ   CLK3
C429 7D 0066    633: CLK1    TST   DSFRST
C42C 26 05      634:         BNE   CLK2
C42E 73 0066    635:         COM   DSFRST
C431 20 14      636:         BRA   CLK3         Do display next time
C433 7F 0066    637: CLK2    CLR   DSFRST
C436 73 006C    638:         COM   DISPMD       Do display now
C439 C6 05      639:         LDA B #5
C43B F7 8006    640:         STA B ADC+2        Enable sampling interrupts
C43E 7D 8004    641:         TST   ADC
C441 D6 BC      642:         LDA B FLAGS
C443 CA 08      643:         ORA B #%1000
C445 D7 BC      644:         STA B FLAGS
C447 BD C0E8    645: CLK3    JSR   CLOCK        Advance time
C44A 32         646:         PUL A
C44B 44         647: J2      LSR A
C44C 24 03      648:         BCC   J3
C44E 7E C4CC    649:         JMP   TIM          Was timer end of count interrupt.
C451 44         650: J3      LSR A
C452 24 9B      651:         BCC   INT          Check no further interrupts.
C454 7E C4DF    652:         JMP   SAM          Must be sampling interrupt.
                653: *
                654: *Trigger.
                655: *
C457 7D 8009    656: TRIG    TST   DAC+1        Clear trig. interrupt flag.
C45A C6 F4      657:         LDA B #$F4
C45C F7 8041    658:         STA B TIMER+1      Load TIMER 1 with 1/2 sec.
C45F C6 01      659:         LDA B #1
```

```
C461 F7 8041  660:          STA B TIMER+1
C464 7F 0065  661:          CLR   DISPEN
C467 7D 0067  662:          TST   DISPDN        Display requested?
C46A 26 03    663:          BNE   TRG1
C46C 73 0065  664:          COM   DISPEN        Yes, set display enable
C46F 7C 00BD  665: TRG1     INC   TRGFLG
C472 86 04    666:          LDA A #%0100        Timer mask pattern.
C474 95 BC    667:          BIT A FLAGS         Timer still running?
C476 27 16    668:          BEQ   J4
C478 C6 01    669:          LDA B #1
C47A D7 CE    670:          STA B ERRORS
C47C 7C 8042  671:          INC   TIMER+2       Stop timer.
C47F 7D 8008  672:          TST   DAC           Clear its interrupt flag.
C482 D6 BC    673:          LDA B FLAGS
C484 C4 1B    674:          AND B #%11011       Clear its mask bit.
C486 D7 BC    675:          STA B FLAGS
C488 7C 00A8  676:          INC   FLSTRG        False trigger count.
C48B 7E C3EF  677:          JMP   INT
              678: *
C48E 7F 00A8  679: J4       CLR   FLSTRG
C491 7D 00AD  680:          TST   SMDONE        Sampling to be done?
C494 26 33    681:          BNE   J6
C496 DE BA    682:          LDX   DELAY         Yes
C498 27 1F    683:          BEQ   J5            No delay. Don't start timer.
              684: *Set up timer.
C49A C6 B0    685:          LDA B #$B0
C49C F7 8043  686:          STA B TIMER+3
C49F D6 BB    687:          LDA B DELAY+1       Low byte
C4A1 F7 8042  688:          STA B TIMER+2
C4A4 D6 BA    689:          LDA B DELAY         High byte
C4A6 F7 8042  690:          STA B TIMER+2
C4A9 C6 1F    691:          LDA B #$1F
C4AB F7 800A  692:          STA B DAC+2         Enable timer int.
C4AE 7D 8008  693:          TST   DAC
C4B1 D6 BC    694:          LDA B FLAGS
C4B3 CA 04    695:          ORA B #%0100        Set timer mask bit.
C4B5 D7 BC    696:          STA B FLAGS
C4B7 20 10    697:          BRA   J6
              698: *
              699: *Get ready for sampling
C4B9 DE BE    700: J5       LDX   DATSIZ
C4BB DF C0    701:          STX   DATCNT
C4BD C6 05    702:          LDA B #5            Enable sampling
C4BF F7 8006  703:          STA B ADC+2
C4C2 7D 8004  704:          TST   ADC
C4C5 C6 1B    705:          LDA B #%11011
C4C7 D7 BC    706:          STA B FLAGS
C4C9 7E C3EF  707: J6       JMP   INT
              708: *
              709: *DELAY Here when count-down finished.
              710: *
C4CC 7C 8042  711: TIM      INC   TIMER+2       Stop timer.
C4CF C6 1E    712:          LDA B #$1E          Stop timer.
C4D1 F7 800A  713:          STA B DAC+2
C4D4 7D 8008  714:          TST   DAC
C4D7 D6 BC    715:          LDA B FLAGS         Clear its mask bit.
C4D9 C4 1B    716:          AND B #%11011
C4DB D7 BC    717:          STA B FLAGS
C4DD 20 DA    718:          BRA   J5
              719: *Sample. Here on EOC pulse from ADC.
```

```
C4DF 7D 8004   720: SAM     TST   ADC          Clear int. flag
C4E2 7D 006C   721:         TST   DISPMD       Display mode?
C4E5 27 06     722:         BEQ   SM1
C4E7 BD CC62   723:         JSR   RASTAS       Yes
C4EA 7E C3EF   724:         JMP   INT
C4ED 7D 00AD   725: SM1     TST   SMDONE       No, sampling done?
C4F0 26 47     726:         BNE   J8
C4F2 DE C0     727:         LDX   DATCNT       No, no data to be sampled?
C4F4 27 43     728:         BEQ   J8
               729: *Sample during echo
               730: *Apply AGC and TVG
               731: *
C4F6 D6 C4     732:         LDA B SCALE
C4F8 D7 90     733:         STA B SCTEMP
C4FA B6 8005   734:         LDA A ADC+1        Get high 8 bits.
C4FD F6 8004   735:         LDA B ADC          Get low 4 bits.
C500 BD C638   736:         JSR   TVG
C503 7D 0090   737: J13     TST   SCTEMP
C506 27 13     738:         BEQ   J15          Already maxm. gain.
C508 7A 0090   739:         DEC   SCTEMP
C50B 58        740:         ASL B
C50C 49        741:         ROL A              Double sample.
C50D 24 F4     742:         BCC   J13          Overloaded?
C50F 86 FF     743:         LDA A #$FF         Yes. Set to F.S.
C511 7C 00C2   744: J14     INC   OVFLW        Count overflows.
C514 26 09     745:         BNE   J16
C516 7A 00C2   746:         DEC   OVFLW        Count saturates at $FF
C519 20 04     747:         BRA   J16
C51B 81 FF     748: J15     CMP A #$FF         F.S.?
C51D 27 F2     749:         BEQ   J14          Yes
               750: *Store the sample
C51F DE C5     751: J16     LDX   DESTAD
C521 A7 00     752:         STA A 0,X          Store in array
C523 08        753:         INX                Advance pointer
C524 DF C5     754:         STX   DESTAD
C526 B7 8009   755:         STA A DAC+1        Echo sample to DAC
C529 DE C0     756:         LDX   DATCNT       Decrement data count
C52B 09        757:         DEX
C52C DF C0     758:         STX   DATCNT
C52E 27 1D     759:         BEQ   J9           Zero - finished sampling.
               760: *Continue sampling.
C530 D6 BC     761:         LDA B FLAGS
C532 CA 08     762:         ORA B #%1000       Set sampling mask bit.
C534 D7 BC     763:         STA B FLAGS
C536 7E C3EF   764:         JMP   INT
               765: *
               766: *Disable sampling
C539 C6 04     767: J8      LDA B #4
C53B F7 8006   768:         STA B ADC+2
C53E 7D 8004   769:         TST   ADC
C541 D6 BC     770:         LDA B FLAGS
C543 C4 17     771:         AND B #%10111      Clear its mask bit.
C545 D7 BC     772:         STA B FLAGS
C547 7F 8009   773:         CLR   DAC+1
C54A 7E C3EF   774:         JMP   INT
               775: *
               776: *Sampling finished.
C54D C6 01     777: J9      LDA B #1
C54F D7 AD     778:         STA B SMDONE       Set sampling done flag
C551 C6 04     779:         LDA B #4
```

```
C553 F7 8006  780:         STA B ADC+2
C556 7D 8004  781:         TST   ADC
C559 D6 BC    782:         LDA B FLAGS
C55B C4 17    783:         AND B #%10111      Clear its mask bit
C55D D7 BC    784:         STA B FLAGS
C55F 7F 8009  785:         CLR   DAC+1        Echoed O/P goes to zero.
              786: *Adjust gain for next sampling session
C562 7D 00C2  787:         TST   OVFLW
C565 27 16    788:         BEQ   J19          No overflows
C567 D6 C2    789:         LDA B OVFLW
C569 C1 05    790:         CMP B #5           No. of overflows allowed.
C56B 25 1F    791:         BCS   J21
C56D 7D 00C4  792:         TST   SCALE
C570 26 06    793:         BNE   J18          Gain can be increased
C572 C6 03    794:         LDA B #3
C574 D7 D0    795:         STA B ERRORS+2     "Over loading"
C576 20 14    796:         BRA   J21
C578 7A 00C4  797: J18     DEC   SCALE        Reduce gain
C57B 20 0F    798:         BRA   J21
C57D D6 C4    799: J19     LDA B SCALE
C57F C1 04    800:         CMP B #4           Can gain be increased?
C581 25 06    801:         BCS   J20          Yes
C583 C6 04    802:         LDA B #4           No  set it to maxm.
C585 D7 C4    803:         STA B SCALE
C587 20 03    804:         BRA   J21
C589 7C 00C4  805: J20     INC   SCALE        Increase gain
C58C 7F 00C2  806: J21     CLR   OVFLW
C58F 7E C3EF  807:         JMP   INT          And return
              808: *
              809: *Adjust DEPTH (ms) for delay and PHASE
              810: *
C592 D6 CA    811: ADJUST  LDA B NWDPTH
C594 96 CB    812:         LDA A NWDPTH+1     Get fresh depth
C596 9B BB    813:         ADD A DELAY+1
C598 D9 BA    814:         ADC B DELAY        Add on delay (mS)
C59A 37       815:         PSH B
C59B D6 CC    816:         LDA B PHASE
C59D 2E 07    817:         BGT   ADJ2         Zero or negative?
C59F C6 04    818: ADJ1    LDA B #4
C5A1 D7 D1    819:         STA B ERRORS+3     Phase out of range
C5A3 33       820:         PUL B
C5A4 20 12    821:         BRA   ADJ4
C5A6 C1 06    822: ADJ2    CMP B #6
C5A8 24 F5    823:         BCC   ADJ1         Phase too large
C5AA D7 A7    824:         STA B PHTEMP
C5AC 33       825:         PUL B
C5AD 7A 00A7  826: ADJ3    DEC   PHTEMP       0-1500 m actually phase 1
C5B0 27 06    827:         BEQ   ADJ4
C5B2 8B D0    828:         ADD A #$D0
C5B4 C9 07    829:         ADC B #7           Add 2000 mS
C5B6 20 F5    830:         BRA   ADJ3         Loop until PHTEMP=0
C5B8 8B 08    831: ADJ4    ADD A #8           Adjust for shift due to "EDGE"
C5BA C9 00    832:         ADC B #0           Propagate carry
C5BC D7 CA    833:         STA B NWDPTH
C5BE 97 CB    834:         STA A NWDPTH+1     Put it back corrected.
C5C0 39       835:         RTS
              836: *
              837: *Find leading edges of echo signal.
              838: *Correlate with 17 sample ramp - -8 to +8
              839: *
```

```
C5C1 CE 0000   840: EDGE   LDX    #0
C5C4 DF A1     841:        STX    SK
C5C6 DE 9D     842:        LDX    FIRSTX
C5C8 86 11     843:        LDA A  #17
C5CA BD C3DB   844:        JSR    ADDA
C5CD DF F4     845:        STX    XTEMP
C5CF DE 9D     846:        LDX    FIRSTX
C5D1 4F        847:        CLR A
C5D2 5F        848:        CLR B
C5D3 AB 00     849: E1     ADD A  0,X          Initialise running sum
C5D5 C9 00     850:        ADC B  #0
C5D7 08        851:        INX
C5D8 9C F4     852:        CPX    XTEMP
C5DA 26 F7     853:        BNE    E1
C5DC D7 A3     854:        STA B  SUMK
C5DE 97 A4     855:        STA A  SUMK+1
C5E0 DE 9F     856:        LDX    LASTX
C5E2 86 F0     857:        LDA A  #-16
C5E4 BD C3DB   858:        JSR    ADDA
C5E7 DF A5     859:        STX    EGEND        Edge stops before end of data
C5E9 5F        860:        CLR B
C5EA DE 9D     861:        LDX    FIRSTX       Start algorithm
C5EC A6 00     862: E2     LDA A  0,X
C5EE AB 11     863:        ADD A  17,X
C5F0 C9 00     864:        ADC B  #0
C5F2 48        865:        ASL A               Times 8
C5F3 59        866:        ROL B
C5F4 48        867:        ASL A
C5F5 59        868:        ROL B
C5F6 48        869:        ASL A
C5F7 59        870:        ROL B
C5F8 AB 00     871:        ADD A  0,X
C5FA C9 00     872:        ADC B  #0
C5FC 9B A2     873:        ADD A  SK+1
C5FE D9 A1     874:        ADC B  SK
C600 90 A4     875:        SUB A  SUMK+1
C602 D2 A3     876:        SBC B  SUMK
C604 97 A2     877:        STA A  SK+1
C606 D7 A1     878:        STA B  SK
C608 2A 02     879:        BPL    E3           Set -ve values to 0
C60A 4F        880:        CLR A
C60B 5F        881:        CLR B
C60C 48        882: E3     ASL A               Times 8
C60D 59        883:        ROL B
C60E 48        884:        ASL A
C60F 59        885:        ROL B
C610 48        886:        ASL A
C611 59        887:        ROL B
C612 24 02     888:        BCC    E5           Over loaded?
C614 C6 FF     889:        LDA B  #$FF         Set to F.S.
C616 37        890: E5     PSH B
C617 5F        891:        CLR B
C618 A6 11     892:        LDA A  17,X
C61A A0 00     893:        SUB A  0,X
C61C C2 00     894:        SBC B  #0
C61E 9B A4     895:        ADD A  SUMK+1
C620 D9 A3     896:        ADC B  SUMK
C622 97 A4     897:        STA A  SUMK+1
C624 D7 A3     898:        STA B  SUMK
C626 33        899:        PUL B
```

```
C627 E7 00    900:          STA B 0,X          Over-write with result
C629 5F       901:          CLR B
C62A 08       902:          INX
C62B 9C A5    903:          CPX   EGEND        Finished yet?
C62D 26 BD    904:          BNE   E2           No loop back
C62F C6 11    905:          LDA B #17          Clear rest of array
C631 6F 00    906: E4       CLR   0,X
C633 08       907:          INX
C634 5A       908:          DEC B
C635 26 FA    909:          BNE   E4
C637 39       910:          RTS
              911: *
              912: *Apply TVG in TX range.
              913: *
C638 7D 00B2  914: TVG      TST   GATNFG       Gating mode?
C63B 26 39    915:          BNE   TV5          If yes then don't apply TVG
C63D 37       916:          PSH B
C63E 36       917:          PSH A
C63F 96 BB    918:          LDA A DELAY+1
C641 D6 BA    919:          LDA B DELAY
C643 9B BF    920:          ADD A DATSIZ+1
C645 D9 BE    921:          ADC B DATSIZ
C647 90 C1    922:          SUB A DATCNT+1
C649 D2 C0    923:          SBC B DATCNT
              924: *We now have position of sample re. TX
C64B C1 07    925:          CMP B #7
C64D 22 06    926:          BHI   TV1
C64F 25 08    927:          BCS   TV2
C651 81 D0    928:          CMP A #$D0
C653 25 04    929:          BCS   TV2
C655 80 D0    930: TV1      SUB A #$D0         Result is greater than 2000
C657 C2 07    931:          SBC B #7           therefore subtract 2000
C659 5D       932: TV2      TST B              High byte zero?
C65A 26 09    933:          BNE   TV3          If no then out of TVG range
C65C 81 48    934:          CMP A #72
C65E 24 05    935:          BCC   TV3
C660 31       936:          INS                Repair stack
C661 31       937:          INS
C662 4F       938:          CLR A              0-72 therefore return zero
C663 5F       939:          CLR B
C664 39       940:          RTS
C665 5D       941: TV3      TST B
C666 26 0C    942:          BNE   TV4
C668 81 C8    943:          CMP A #200
C66A 24 08    944:          BCC   TV4
C66C 80 48    945:          SUB A #72          Ramp data 0 - 1
C66E 48       946:          ASL A
C66F 36       947:          PSH A
C670 BD C6E5  948:          JSR   MPYDBL       X 2(sample no.-114)
C673 31       949:          INS
C674 32       950: TV4      PUL A
C675 33       951:          PUL B
C676 39       952: TV5      RTS
              953: *
              954: *PEAK
              955: *Find nearest peak to PREDICtion and largest overall
              956: *
C677 4F       957: PEAK     CLR A
C678 97 9A    958:          STA A PKVAL
C67A 97 9B    959:          STA A PKPOSN
```

```
C67C 97 9C     960:          STA A PKPOSN+1
C67E DE 9F     961:          LDX   LASTX
C680 09        962:          DEX
C681 DF A5     963:          STX   EGEND
C683 CE 7FFF   964:          LDX   #$7FFF       Largest +ve No. to start with
C686 DF 98     965:          STX   OFFSET
C688 DE 9D     966:          LDX   FIRSTX
C68A A6 01     967: PK1      LDA A 1,X
C68C A1 00     968:          CMP A 0,X
C68E 23 04     969:          BLS   PK2
C690 A1 02     970:          CMP A 2,X
C692 24 06     971:          BCC   PK3
C694 08        972: PK2      INX
C695 9C A5     973:          CPX   EGEND
C697 26 F1     974:          BNE   PK1
C699 39        975:          RTS
C69A 08        976: PK3      INX                This sample is greater then adjacent
               977: *                           ones.
C69B DF 96     978:          STX   SAVX
C69D 9C A5     979:          CPX   EGEND
C69F 26 01     980:          BNE   PK4
C6A1 39        981:          RTS
C6A2 97 B9     982: PK4      STA A ATEMP
C6A4 96 97     983:          LDA A SAVX+1
C6A6 D6 96     984:          LDA B SAVX
C6A8 90 9E     985:          SUB A FIRSTX+1     Find relative position
C6AA D2 9D     986:          SBC B FIRSTX
C6AC 97 D6     987:          STA A XT+1
C6AE D7 D5     988:          STA B XT
               989: *Call WEIGHT to weight signal 0 at edges to 1 in middle
C6B0 BD C76A   990:          JSR   WEIGHT
C6B3 97 B9     991:          STA A ATEMP
C6B5 91 9A     992:          CMP A PKVAL        > than previous greatest?
C6B7 23 06     993:          BLS   PK5
C6B9 97 9A     994:          STA A PKVAL        Yes. It becomes newest maxm.
C6BB DE D5     995:          LDX   XT
C6BD DF 9B     996:          STX   PKPOSN       And save its position
C6BF 96 D6     997: PK5      LDA A XT+1
C6C1 D6 D5     998:          LDA B XT
               999: *Is it closest to PREDICtion?
C6C3 90 95    1000:          SUB A PREDIC+1
C6C5 D2 94    1001:          SBC B PREDIC
C6C7 2A 02    1002:          BPL   PK6          We only want modulus of difference
C6C9 40       1003:          NEG A
C6CA 53       1004:          COM B
C6CB D1 98    1005: PK6      CMP B OFFSET       Have we a smaller OFFSET than
              1006: *                           before?
C6CD 22 12    1007:          BHI   PK8          No
C6CF 25 04    1008:          BCS   PK7
C6D1 91 99    1009:          CMP A OFFSET+1
C6D3 22 0C    1010:          BHI   PK8
C6D5 97 99    1011: PK7      STA A OFFSET+1     Yes Save it
C6D7 D7 98    1012:          STA B OFFSET       This is new OFFSET
C6D9 96 B9    1013:          LDA A ATEMP
C6DB 97 91    1014:          STA A NRSTHT       Save its amplitude
C6DD DE D5    1015:          LDX   XT
C6DF DF 92    1016:          STX   NRSTPS       Save its position
C6E1 DE 96    1017: PK8      LDX   SAVX
C6E3 20 A5    1018:          BRA   PK1          Keep looking
              1019: *
```

```
                    1020: *Multiply double byte by single byte but throw away LSByte
                    1021: *Arguments are passed on stack :_
                    1022: *low byte  multiplicand   (first on stack)
                    1023: *high byte      "
                    1024: *one byte multiplier
                    1025: *Results in :_
                    1026: *Low byte product
                    1027: *high byte    "
                    1028: *unchanged multiplier
C6E5 30             1029: MPYDBL TSX
C6E6 C6 70          1030:        LDA B #$70
C6E8 D7 FC          1031:        STA B MULT
C6EA A6 02          1032:        LDA A 2,X
C6EC 97 FF          1033:        STA A MULT+3
C6EE A6 04          1034:        LDA A 4,X
C6F0 97 FE          1035:        STA A MULT+2
C6F2 C6 79          1036:        LDA B #$79          Start multiplier
C6F4 D7 FC          1037:        STA B MULT
C6F6 01             1038:        NOP                 Wait for it
C6F7 01             1039:        NOP
C6F8 A6 03          1040:        LDA A 3,X
C6FA 97 FE          1041:        STA A MULT+2
C6FC C6 71          1042:        LDA B #$71
C6FE D7 FC          1043:        STA B MULT
C700 01             1044:        NOP
C701 01             1045:        NOP
C702 96 FD          1046:        LDA A MULT+1
C704 A7 03          1047:        STA A 3,X
C706 96 FE          1048:        LDA A MULT+2
C708 A7 04          1049:        STA A 4,X
C70A 39             1050:        RTS
                    1051: *
                    1052: *Divide double byte by single byte.
                    1053: *Dividend and divisor on stack.
                    1054: *Quotient and remainder returned.
C70B 30             1055: DIVDBL TSX
C70C C6 70          1056:        LDA B #$70
C70E D7 FC          1057:        STA B MULT
C710 A6 02          1058:        LDA A 2,X
C712 97 FF          1059:        STA A MULT+3
C714 A6 03          1060:        LDA A 3,X
C716 97 FD          1061:        STA A MULT+1
C718 A6 04          1062:        LDA A 4,X
C71A 97 FE          1063:        STA A MULT+2
C71C C6 72          1064:        LDA B #$72
C71E D7 FC          1065:        STA B MULT
C720 01             1066:        NOP
C721 01             1067:        NOP
C722 96 FE          1068:        LDA A MULT+2
C724 A7 03          1069:        STA A 3,X
C726 96 FD          1070:        LDA A MULT+1
C728 A7 04          1071:        STA A 4,X
C72A 7D 00FC        1072:        TST   MULT
C72D 27 01          1073:        BEQ   DV1
C72F 0D             1074:        SEC
C730 39             1075: DV1    RTS
                    1076: *Shift gate by adjusting delay.
                    1077: *Out of range delay is prevented.
                    1078: *
C731 DE BE          1079: SHFTGT LDX   DATSIZ
```

```
C733 DF F4    1080:          STX    XTEMP
C735 74 00F4  1081:          LSR    XTEMP
C738 76 00F5  1082:          ROR    XTEMP+1
C73B 90 F5    1083:          SUB A  XTEMP+1
C73D D2 F4    1084:          SBC B  XTEMP
C73F 9B BB    1085:          ADD A  DELAY+1
C741 D9 BA    1086:          ADC B  DELAY
C743 2A 0B    1087:          BPL    SH1
C745 8B D0    1088:          ADD A  #$D0
C747 C9 07    1089:          ADC B  #7
C749 7A 00CC  1090:          DEC    PHASE
C74C 97 B8    1091:          STA A  AVDPTH+1
C74E D7 B7    1092:          STA B  AVDPTH
C750 C1 07    1093: SH1      CMP B  #7
C752 22 06    1094:          BHI    SH2
C754 25 0F    1095:          BCS    SH3
C756 81 CE    1096:          CMP A  #$CE
C758 23 0B    1097:          BLS    SH3
C75A 80 CF    1098: SH2      SUB A  #$CF
C75C C2 07    1099:          SBC B  #7
C75E 7C 00CC  1100:          INC    PHASE
C761 97 B8    1101:          STA A  AVDPTH+1
C763 D7 B7    1102:          STA B  AVDPTH
C765 97 BB    1103: SH3      STA A  DELAY+1
C767 D7 BA    1104:          STA B  DELAY
C769 39       1105:          RTS
              1106: *
              1107: *WEIGHT differentiated signal - zero at ends
              1108: *one in middle.
              1109: *
C76A DE BE    1110: WEIGHT   LDX    DATSIZ
C76C DF F4    1111:          STX    XTEMP
C76E 74 00F4  1112:          LSR    XTEMP
C771 76 00F5  1113:          ROR    XTEMP+1
C774 96 D6    1114:          LDA A  XT+1
C776 D6 D5    1115:          LDA B  XT
C778 D1 F4    1116:          CMP B  XTEMP
C77A 22 06    1117:          BHI    WG1
C77C 25 0A    1118:          BCS    WG2
C77E 91 F5    1119:          CMP A  XTEMP+1
C780 23 06    1120:          BLS    WG2
C782 40       1121: WG1      NEG A
C783 53       1122:          COM B
C784 9B BF    1123:          ADD A  DATSIZ+1
C786 D9 BE    1124:          ADC B  DATSIZ
C788 36       1125: WG2      PSH A
C789 37       1126:          PSH B
C78A 96 B9    1127:          LDA A  ATEMP
C78C 36       1128:          PSH A
C78D BD C6E5  1129:          JSR    MPYDBL
C790 31       1130:          INS
C791 33       1131:          PUL B
C792 32       1132:          PUL A
C793 7F 00B9  1133:          CLR    ATEMP
C796 7D 00F4  1134: WG3      TST    XTEMP
C799 27 0D    1135:          BEQ    WG4
C79B 74 00F4  1136:          LSR    XTEMP
C79E 76 00F5  1137:          ROR    XTEMP+1
C7A1 54       1138:          LSR B
C7A2 46       1139:          ROR A
```

```
C7A3 76 00B9 1140:          ROR   ATEMP
C7A6 20 EE   1141:          BRA   WG3
C7A8 D6 B9   1142: WG4      LDA B ATEMP
C7AA 37      1143:          PSH B
C7AB 36      1144:          PSH A
C7AC 96 F5   1145:          LDA A XTEMP+1
C7AE 36      1146:          PSH A
C7AF BD C70B 1147:          JSR   DIVDBL
C7B2 31      1148:          INS
C7B3 32      1149:          PUL A
C7B4 31      1150:          INS
C7B5 39      1151:          RTS
             1152: *
             1153: *Single pole FILTer
             1154: *
C7B6 36      1155: FILT     PSH A
C7B7 37      1156:          PSH B
C7B8 96 B4   1157:          LDA A COEF1+1
C7BA 36      1158:          PSH A
C7BB 96 B3   1159:          LDA A COEF1
C7BD 36      1160:          PSH A
C7BE BD C81D 1161:          JSR   MPY2X2
C7C1 33      1162:          PUL B
C7C2 32      1163:          PUL A
C7C3 31      1164:          INS
C7C4 31      1165:          INS
C7C5 9B B8   1166:          ADD A AVDPTH+1
C7C7 D9 B7   1167:          ADC B AVDPTH
C7C9 36      1168:          PSH A
C7CA 37      1169:          PSH B
C7CB 96 B6   1170:          LDA A COEF2+1
C7CD 36      1171:          PSH A
C7CE 96 B5   1172:          LDA A COEF2
C7D0 36      1173:          PSH A
C7D1 BD C81D 1174:          JSR   MPY2X2
C7D4 33      1175:          PUL B
C7D5 32      1176:          PUL A
C7D6 31      1177:          INS
C7D7 31      1178:          INS
C7D8 97 B8   1179:          STA A AVDPTH+1
C7DA D7 B7   1180:          STA B AVDPTH
C7DC 39      1181:          RTS
             1182: *
             1183: *Multiply 3 bytes by 1. Multiplier,multiplicand
             1184: *and product passed on stack
             1185: *
0070         1186: RESET    EQU   $70
0079         1187: MULCLR   EQU   $79
0071         1188: MULCOM   EQU   $71
             1189: *
C7DD 30      1190: MPY1X3   TSX
C7DE 86 70   1191:          LDA A #RESET
C7E0 97 FC   1192:          STA A MULT
C7E2 A6 02   1193:          LDA A 2,X
C7E4 97 FF   1194:          STA A MULT+3
C7E6 A6 05   1195:          LDA A 5,X
C7E8 97 FE   1196:          STA A MULT+2
C7EA 86 79   1197:          LDA A #MULCLR
C7EC 97 FC   1198:          STA A MULT
C7EE 01      1199:          NOP
```

```
C7EF 01      1200:          NOP
C7F0 96 FE   1201:          LDA A MULT+2
C7F2 A7 05   1202:          STA A 5,X
C7F4 86 70   1203:          LDA A #RESET
C7F6 97 FC   1204:          STA A MULT
C7F8 A6 04   1205:          LDA A 4,X
C7FA 97 FE   1206:          STA A MULT+2
C7FC 86 71   1207:          LDA A #MULCOM
C7FE 97 FC   1208:          STA A MULT
C800 01      1209:          NOP
C801 01      1210:          NOP
C802 96 FE   1211:          LDA A MULT+2
C804 A7 04   1212:          STA A 4,X
C806 96 70   1213:          LDA A RESET
C808 97 FC   1214:          STA A MULT
C80A A6 03   1215:          LDA A 3,X
C80C 97 FE   1216:          STA A MULT+2
C80E 86 71   1217:          LDA A #MULCOM
C810 97 FC   1218:          STA A MULT
C812 01      1219:          NOP
C813 01      1220:          NOP
C814 96 FE   1221:          LDA A MULT+2
C816 A7 03   1222:          STA A 3,X
C818 96 FD   1223:          LDA A MULT+1
C81A A7 02   1224:          STA A 2,X
C81C 39      1225:          RTS
             1226: *
             1227: *Multiply 2 bytes by 2 bytes
             1228: *
C81D 30      1229: MPY2X2 TSX
C81E 86 70   1230:          LDA A #RESET
C820 97 FC   1231:          STA A MULT
C822 A6 03   1232:          LDA A 3,X
C824 97 FF   1233:          STA A MULT+3
C826 A6 05   1234:          LDA A 5,X
C828 36      1235:          PSH A
C829 97 FE   1236:          STA A MULT+2
C82B 86 79   1237:          LDA A #MULCLR
C82D 97 FC   1238:          STA A MULT
C82F 01      1239:          NOP
C830 01      1240:          NOP
C831 96 FE   1241:          LDA A MULT+2
C833 A7 05   1242:          STA A 5,X
C835 86 70   1243:          LDA A #RESET
C837 97 FC   1244:          STA A MULT
C839 A6 04   1245:          LDA A 4,X
C83B 36      1246:          PSH A
C83C 97 FE   1247:          STA A MULT+2
C83E 86 71   1248:          LDA A #MULCOM
C840 97 FC   1249:          STA A MULT
C842 01      1250:          NOP
C843 01      1251:          NOP
C844 96 FE   1252:          LDA A MULT+2
C846 A7 04   1253:          STA A 4,X
C848 86 70   1254:          LDA A #RESET
C84A 97 FC   1255:          STA A MULT
C84C A6 02   1256:          LDA A 2,X
C84E 97 FF   1257:          STA A MULT+3
C850 32      1258:          PUL A
C851 97 FE   1259:          STA A MULT+2
```

```
C853 86 71    1260:          LDA A #MULCOM
C855 97 FC    1261:          STA A MULT
C857 01       1262:          NOP
C858 01       1263:          NOP
C859 96 FE    1264:          LDA A MULT+2
C85B A7 03    1265:          STA A 3,X
C85D A6 02    1266:          LDA A 2,X
C85F 36       1267:          PSH A
C860 96 FD    1268:          LDA A MULT+1
C862 A7 02    1269:          STA A 2,X
C864 86 70    1270:          LDA A #RESET
C866 97 FC    1271:          STA A MULT
C868 32       1272:          PUL A
C869 97 FF    1273:          STA A MULT+3
C86B 32       1274:          PUL A
C86C 97 FE    1275:          STA A MULT+2
C86E 86 79    1276:          LDA A #MULCLR
C870 97 FC    1277:          STA A MULT
C872 01       1278:          NOP
C873 01       1279:          NOP
C874 96 FE    1280:          LDA A MULT+2
C876 AB 04    1281:          ADD A 4,X
C878 A7 04    1282:          STA A 4,X
C87A 96 FD    1283:          LDA A MULT+1
C87C A9 03    1284:          ADC A 3,X
C87E A7 03    1285:          STA A 3,X
C880 24 02    1286:          BCC   MP1
C882 6C 02    1287:          INC   2,X
C884 39       1288: MP1      RTS
              1289: *
              1290: *I/P 4 unpacked BCD digits into "BCDIN"
              1291: *Thumbwheel switch entry completed by push-button.
              1292: *Or enter at BC2 to fetch entry anytime.
              1293: *
C885 C6 B0    1294: INBCD    LDA B #$B0
C887 F7 8020  1295: BC3      STA B PANEL       Select digit 3 (digits 0 - 3)
C88A B6 8020  1296:          LDA A PANEL       Get BCD.
C88D 43       1297:          COM A             Was -ve logic
C88E 84 0F    1298:          AND A #$F         Mask off 4 lsb's
C890 A7 00    1299:          STA A 0,X         Into buffer.
C892 08       1300:          INX
C893 C0 10    1301:          SUB B #$10        Select next highest digit.
C895 C1 70    1302:          CMP B #$70        Finished?
C897 26 EE    1303:          BNE   BC3
C899 39       1304:          RTS
              1305: *
              1306: *Convert 4 un-packed BCD digits
              1307: *to binary in acc.s B-A
              1308: *
C89A 4F       1309: CVDTB    CLR A
C89B 5F       1310:          CLR B
C89C CE C8C9  1311:          LDX   #TENPWR
C89F DF F4    1312:          STX   XTEMP
C8A1 CE 0086  1313:          LDX   #BCDIN
C8A4 6D 00    1314: CVBIN1   TST   0,X
C8A6 27 0E    1315:          BEQ   CVBIN2
C8A8 6A 00    1316:          DEC   0,X
C8AA DF F6    1317:          STX   SPTEMP
C8AC DE F4    1318:          LDX   XTEMP
C8AE AB 01    1319:          ADD A 1,X
```

```
C8B0 E9 00    1320:          ADC B 0,X
C8B2 DE F6    1321:          LDX   SPTEMP
C8B4 20 EE    1322:          BRA   CVBIN1
C8B6 08       1323: CVBIN2   INX
C8B7 DF F6    1324:          STX   SPTEMP
C8B9 DE F4    1325:          LDX   XTEMP
C8BB 08       1326:          INX
C8BC 08       1327:          INX
C8BD 8C C8D1  1328:          CPX   #TENPWR+8
C8C0 27 06    1329:          BEQ   CVBIN3
C8C2 DF F4    1330:          STX   XTEMP
C8C4 DE F6    1331:          LDX   SPTEMP
C8C6 20 DC    1332:          BRA   CVBIN1
C8C8 39       1333: CVBIN3   RTS
C8C9 03 E8    1334: TENPWR   FDB   1000,100,10,1
              1335: *
              1336: *Display 8 unpacked BCD on front panel.
              1337: *
C8D1 CE 0086  1338: LCD      LDX   #BCDIN
C8D4 7F 8022  1339:          CLR   PANEL+2
C8D7 86 FF    1340:          LDA A #$FF
C8D9 B7 8020  1341:          STA A PANEL
C8DC 86 04    1342:          LDA A #4
C8DE B7 8022  1343:          STA A PANEL+2
C8E1 C6 70    1344:          LDA B #$70
C8E3 A6 00    1345: LC1      LDA A 0,X
C8E5 08       1346:          INX
C8E6 84 0F    1347:          AND A #$F
C8E8 1B       1348:          ABA
C8E9 B7 8020  1349:          STA A PANEL
C8EC 86 FF    1350:          LDA A #$FF
C8EE B7 8020  1351:          STA A PANEL
C8F1 C0 10    1352:          SUB B #$10
C8F3 C1 F0    1353:          CMP B #$F0
C8F5 26 EC    1354:          BNE   LC1
C8F7 39       1355:          RTS
              1356: *
              1357: *Convert metres to mS.
              1358: *Result in x reg. PHASE in acc A
              1359: *
C8F8 48       1360: CVRTMS   ASL A
C8F9 59       1361:          ROL B
C8FA 48       1362:          ASL A
C8FB 59       1363:          ROL B
C8FC 36       1364:          PSH A
C8FD 37       1365:          PSH B
C8FE 86 55    1366:          LDA A #$55
C900 36       1367:          PSH A
C901 36       1368:          PSH A
C902 BD C81D  1369:          JSR   MPY2X2
C905 33       1370:          PUL B
C906 32       1371:          PUL A
C907 31       1372:          INS
C908 31       1373:          INS
C909 DE BE    1374:          LDX   DATSIZ
C90B DF F4    1375:          STX   XTEMP
C90D 74 00F4  1376:          LSR   XTEMP
C910 76 00F5  1377:          ROR   XTEMP+1
C913 90 F5    1378:          SUB A XTEMP+1
C915 D2 F4    1379:          SBC B XTEMP
```

```
C917 2A 02    1380:          BPL    CVR1
C919 4F       1381:          CLR A
C91A 5F       1382:          CLR B
C91B 7F 00D9  1383: CVR1     CLR    BTEMP
C91E 7C 00D9  1384: CVR2     INC    BTEMP
C921 80 D0    1385:          SUB A  #$D0
C923 C2 07    1386:          SBC B  #7
C925 24 F7    1387:          BCC    CVR2
C927 8B D0    1388:          ADD A  #$D0
C929 C9 07    1389:          ADC B  #7
C92B 97 F5    1390:          STA A  XTEMP+1
C92D D7 F4    1391:          STA B  XTEMP
C92F DE F4    1392:          LDX    XTEMP
C931 96 D9    1393:          LDA A  BTEMP
C933 39       1394:          RTS
              1395: *
              1396: *Subr. to unpacked time digits.
              1397: *
C934 16       1398: GETLFT   TAB
C935 54       1399:          LSR B
C936 54       1400:          LSR B
C937 54       1401:          LSR B
C938 54       1402:          LSR B
C939 20 01    1403:          BRA    GETR1
C93B 16       1404: GETRGT   TAB
C93C C4 0F    1405: GETR1    AND B  #$F
C93E CB 30    1406:          ADD B  #'0
C940 C1 39    1407:          CMP B  #'9
C942 23 02    1408:          BLS    GETR2
C944 CB 37    1409:          ADD B  #'7
C946 39       1410: GETR2    RTS
              1411: *
              1412: *Get depth from front panel
              1413: *Return if depth entry not changed.
              1414: *Also call TWOMIN and ERRs
              1415: *
C947 BD C236  1416: INDPTH   JSR    TWOMIN
C94A BD C2A3  1417:          JSR    ERRS
              1418: *Decide whether to send time to Mufax.
C94D 96 E7    1419:          LDA A  MINS+1
C94F 8B 94    1420: INDO     ADD A  #$94          Test MINS divisible by 6
C951 19       1421:          DAA
C952 2A FB    1422:          BPL    INDO
C954 8B 06    1423:          ADD A  #6
C956 19       1424:          DAA
C957 26 48    1425:          BNE    IND1
C959 96 E9    1426:          LDA A  SECS+1
C95B 81 20    1427:          CMP A  #$20
C95D 25 42    1428:          BCS    IND1
C95F 81 26    1429:          CMP A  #$26
C961 24 3E    1430:          BCC    IND1
C963 7D 0067  1431:          TST    DISPDN
C966 27 39    1432:          BEQ    IND1
              1433: *Get ready for Mufax time display.
C968 BD CBA3  1434:          JSR    MFINIT        Initialise mufax display
C96B CE 0059  1435:          LDX    #MUFDSP       Transfer to display buffer
C96E 96 E5    1436:          LDA A  HRS+1
C970 8D C2    1437:          BSR    GETLFT        Hours
C972 E7 00    1438:          STA B  0,X
C974 8D C5    1439:          BSR    GETRGT
```

```
C976 E7 01    1440:          STA B 1,X
C978 C6 3A    1441:          LDA B #':
C97A E7 02    1442:          STA B 2,X
C97C 96 E7    1443:          LDA A MINS+1
C97E 8D B4    1444:          BSR   GETLFT       Minutes
C980 E7 03    1445:          STA B 3,X
C982 8D B7    1446:          BSR   GETRGT
C984 E7 04    1447:          STA B 4,X
C986 C6 20    1448:          LDA B #$20
C988 E7 05    1449:          STA B 5,X
C98A 96 E2    1450:          LDA A DAYS
C98C 8D AD    1451:          BSR   GETRGT       Day number
C98E E7 06    1452:          STA B 6,X
C990 96 E3    1453:          LDA A DAYS+1
C992 8D A0    1454:          BSR   GETLFT
C994 E7 07    1455:          STA B 7,X
C996 8D A3    1456:          BSR   GETRGT
C998 E7 08    1457:          STA B 8,X
C99A C6 0D    1458:          LDA B #$D
C99C E7 09    1459:          STA B 9,X
C99E 7F 0067  1460:          CLR   DISPDN       Signal start of display
C9A1 CE 0086  1461: IND1     LDX   #BCDIN
C9A4 7F 8022  1462:          CLR   PANEL+2
C9A7 86 F0    1463:          LDA A #$F0
C9A9 B7 8020  1464:          STA A PANEL
C9AC 86 04    1465:          LDA A #4
C9AE B7 8022  1466:          STA A PANEL+2
C9B1 BD C885  1467:          JSR   INBCD        Get BCD from front panel.
C9B4 BD C89A  1468:          JSR   CVDTB        Convert to binary.
C9B7 D7 F4    1469:          STA B XTEMP
C9B9 97 F5    1470:          STA A XTEMP+1
C9BB DE F4    1471:          LDX   XTEMP
C9BD 9C 84    1472:          CPX   LSTENT       Same as before?
C9BF 27 0B    1473:          BEQ   IND3         Go and display depth.
C9C1 DF 84    1474:          STX   LSTENT       New changed depth entry.
C9C3 BD C8F8  1475: IND2     JSR   CVRTMS       Convert to mS.
C9C6 97 CC    1476:          STA A PHASE
C9C8 DF B7    1477:          STX   AVDPTH
C9CA DF BA    1478:          STX   DELAY
C9CC 39       1479: IND3     RTS
              1480: *Display present depth and send to remote K/B
C9CD D6 CA    1481: DISP1    LDA B NWDPTH
C9CF 96 CB    1482:          LDA A NWDPTH+1
C9D1 48       1483:          ASL A
C9D2 59       1484:          ROL B              Times 2
C9D3 9B CB    1485:          ADD A NWDPTH+1
C9D5 D9 CA    1486:          ADC B NWDPTH       Times 3
C9D7 54       1487:          LSR B
C9D8 46       1488:          ROR A
C9D9 54       1489:          LSR B
C9DA 46       1490:          ROR A              Divide by 4
C9DB D7 F4    1491:          STA B XTEMP        Nett result times 3/4
C9DD 97 F5    1492:          STA A XTEMP+1
C9DF DE F4    1493:          LDX   XTEMP        This is now depth in metres.
C9E1 BD C13A  1494:          JSR   CVBTD        Convert to decimal.
C9E4 DE DB    1495:          LDX   DIGITS+1     Transfer to display buffer.
C9E6 DF 86    1496:          STX   BCDIN
C9E8 DE DD    1497:          LDX   DIGITS+3
C9EA DF 88    1498:          STX   BCDIN+2
C9EC BD C8D1  1499:          JSR   LCD          Display it.
```

```
C9EF BD CCE7 1500:         JSR   SEND        Send it to remote K/B
C9F2 39      1501:         RTS
             1502: *
             1503: *FILTER 2 sec. depths centred on even min.
             1504: *
C9F3 DE 82   1505: DPFILT  LDX   DPSTCK
C9F5 DF 7F   1506:         STX   PNTR2
C9F7 CE CFC0 1507:         LDX   #COEFFS
C9FA DF 7D   1508:         STX   PNTR1
C9FC 7F 0081 1509:         CLR   DIRFLG
C9FF 7F 0078 1510:         CLR   FILTOP
CA02 7F 0079 1511:         CLR   FILTOP+1
CA05 7F 007A 1512:         CLR   FILTOP+2
CA08 7F 007B 1513:         CLR   FILTOP+3
CA0B 7F 007C 1514: CR1     CLR   SIGN
CA0E A6 01   1515:         LDA A 1,X
CA10 E6 00   1516:         LDA B 0,X
CA12 2A 09   1517:         BPL   CR2
CA14 4D      1518:         TST A
CA15 26 01   1519:         BNE   CR6
CA17 5A      1520:         DEC B
CA18 40      1521: CR6     NEG A
CA19 53      1522:         COM B
CA1A 73 007C 1523:         COM   SIGN
CA1D 36      1524: CR2     PSH A
CA1E 37      1525:         PSH B
CA1F 7D 0081 1526:         TST   DIRFLG
CA22 26 0E   1527:         BNE   CRR1
CA24 08      1528:         INX
CA25 08      1529:         INX
CA26 DF 7D   1530:         STX   PNTR1
CA28 8C CFFE 1531:         CPX   #2*31+COEFFS
CA2B 26 09   1532:         BNE   CRR2
CA2D 7C 0081 1533:         INC   DIRFLG
CA30 20 04   1534:         BRA   CRR2
CA32 09      1535: CRR1    DEX
CA33 09      1536:         DEX
CA34 DF 7D   1537:         STX   PNTR1
CA36 DE 7F   1538: CRR2    LDX   PNTR2
CA38 A6 01   1539:         LDA A 1,X
CA3A E6 00   1540:         LDA B 0,X
CA3C 2A 09   1541:         BPL   CR3
CA3E 4D      1542:         TST A
CA3F 26 01   1543:         BNE   CR7
CA41 5A      1544:         DEC B
CA42 40      1545: CR7     NEG A
CA43 53      1546:         COM B
CA44 73 007C 1547:         COM   SIGN
CA47 36      1548: CR3     PSH A
CA48 37      1549:         PSH B
CA49 BD C81D 1550:         JSR   MPY2X2
CA4C 30      1551:         TSX
CA4D 31      1552:         INS
CA4E 31      1553:         INS
CA4F 31      1554:         INS
CA50 31      1555:         INS
CA51 7D 007C 1556:         TST   SIGN
CA54 2A 1A   1557:         BPL   CR4
CA56 96 7B   1558:         LDA A FILTOP+3
CA58 D6 7A   1559:         LDA B FILTOP+2
```

```
CA5A A0 03    1560:          SUB A 3,X
CA5C E2 02    1561:          SBC B 2,X
CA5E 97 7B    1562:          STA A FILTOP+3
CA60 D7 7A    1563:          STA B FILTOP+2
CA62 96 79    1564:          LDA A FILTOP+1
CA64 D6 78    1565:          LDA B FILTOP
CA66 A2 01    1566:          SBC A 1,X
CA68 E2 00    1567:          SBC B 0,X
CA6A 97 79    1568:          STA A FILTOP+1
CA6C D7 78    1569:          STA B FILTOP
CA6E 20 18    1570:          BRA   CR5
CA70 96 7B    1571: CR4      LDA A FILTOP+3
CA72 D6 7A    1572:          LDA B FILTOP+2
CA74 AB 03    1573:          ADD A 3,X
CA76 E9 02    1574:          ADC B 2,X
CA78 97 7B    1575:          STA A FILTOP+3
CA7A D7 7A    1576:          STA B FILTOP+2
CA7C 96 79    1577:          LDA A FILTOP+1
CA7E D6 78    1578:          LDA B FILTOP
CA80 A9 01    1579:          ADC A 1,X
CA82 E9 00    1580:          ADC B 0,X
CA84 97 79    1581:          STA A FILTOP+1
CA86 D7 78    1582:          STA B FILTOP
CA88 DE 7F    1583: CR5      LDX   PNTR2
CA8A 08       1584:          INX
CA8B 08       1585:          INX
CA8C DF 7F    1586:          STX   PNTR2
CA8E 8C 1080  1587:          CPX   #4*31+DPSTRT+4
CA91 26 05    1588:          BNE   CRR3
CA93 CE 1000  1589:          LDX   #DPSTRT
CA96 DF 7F    1590:          STX   PNTR2
CA98 DE 7D    1591: CRR3     LDX   PNTR1
CA9A 8C CFBE  1592:          CPX   #COEFFS-2
CA9D 27 03    1593:          BEQ   CRR4
CA9F 7E CA0B  1594:          JMP   CR1
CAA2 39       1595: CRR4     RTS
              1596: *
              1597: *Filter over +&- 1 Min. depths on the 2 Min. mark
              1598: *Store them in a rolling stack
              1599: *Transfer any unsent depths to the ship's logger.
              1600: *
CAA3 BD C9F3  1601: STORDP   JSR   DPFILT
CAA6 DE 78    1602:          LDX   FILTOP
CAA8 7D 007A  1603:          TST   FILTOP+2
CAAB 2A 01    1604:          BPL   ST1
CAAD 08       1605:          INX
CAAE DF F4    1606: ST1      STX   XTEMP
CAB0 96 F5    1607:          LDA A XTEMP+1
CAB2 D6 F4    1608:          LDA B XTEMP
CAB4 48       1609:          ASL A
CAB5 59       1610:          ROL B
CAB6 9B F5    1611:          ADD A XTEMP+1
CAB8 D9 F4    1612:          ADC B XTEMP
CABA 54       1613:          LSR B
CABB 46       1614:          ROR A
CABC 54       1615:          LSR B
CABD 46       1616:          ROR A
CABE D7 F4    1617:          STA B XTEMP
CAC0 97 F5    1618:          STA A XTEMP+1
CAC2 DE F4    1619:          LDX   XTEMP
```

```
CAC4 BD C13A 1620:          JSR   CVBTD
CAC7 DE 74   1621:          LDX   DPTMP1
CAC9 96 DB   1622:          LDA A DIGITS+1
CACB D6 DC   1623:          LDA B DIGITS+2
CACD C4 OF   1624:          AND B #$OF
CACF 48      1625:          ASL A
CAD0 48      1626:          ASL A
CAD1 48      1627:          ASL A
CAD2 48      1628:          ASL A
CAD3 1B      1629:          ABA
CAD4 A7 00   1630:          STA A 0,X
CAD6 96 DD   1631:          LDA A DIGITS+3
CAD8 D6 DE   1632:          LDA B DIGITS+4
CADA C4 OF   1633:          AND B #$OF
CADC 48      1634:          ASL A
CADD 48      1635:          ASL A
CADE 48      1636:          ASL A
CADF 48      1637:          ASL A
CAE0 1B      1638:          ABA
CAE1 A7 01   1639:          STA A 1,X
CAE3 96 E0   1640:          LDA A YEARS
CAE5 A7 02   1641:          STA A 2,X
CAE7 96 E1   1642:          LDA A YEARS+1
CAE9 A7 03   1643:          STA A 3,X
CAEB 96 E2   1644:          LDA A DAYS
CAED A7 04   1645:          STA A 4,X
CAEF 96 E3   1646:          LDA A DAYS+1
CAF1 A7 05   1647:          STA A 5,X
CAF3 96 E5   1648:          LDA A HRS+1
CAF5 A7 06   1649:          STA A 6,X
CAF7 96 E7   1650:          LDA A MINS+1
CAF9 44      1651:          LSR A
CAFA 0D      1652:          SEC
CAFB 49      1653:          ROL A
CAFC A7 07   1654:          STA A 7,X
CAFE 86 08   1655:          LDA A #8
CB00 BD C3DB 1656:          JSR   ADDA
CB03 8C 4000 1657:          CPX   #DPTMST+$2000
CB06 26 03   1658:          BNE   ST2
CB08 CE 2000 1659:          LDX   #DPTMST
CB0B DF 74   1660: ST2      STX   DPTMP1
             1661: *
             1662: *Get ready to send depth to Mufax.
             1663: *
CB0D 7D 0067 1664:          TST   DISPDN
CB10 27 23   1665:          BEQ   SRCHDP
CB12 BD CBA3 1666:          JSR   MFINIT
CB15 CE 0059 1667:          LDX   #MUFDSP
CB18 86 20   1668:          LDA A #$20
CB1A A7 00   1669:          STA A 0,X
CB1C A7 01   1670:          STA A 1,X
CB1E 96 DB   1671:          LDA A DIGITS+1
CB20 A7 02   1672:          STA A 2,X
CB22 96 DC   1673:          LDA A DIGITS+2
CB24 A7 03   1674:          STA A 3,X
CB26 96 DD   1675:          LDA A DIGITS+3
CB28 A7 04   1676:          STA A 4,X
CB2A 96 DE   1677:          LDA A DIGITS+4
CB2C A7 05   1678:          STA A 5,X
CB2E 86 0D   1679:          LDA A #$D
```

```
CB30 A7 06    1680:          STA A 6,X
CB32 7F 0067  1681:          CLR   DISPDN
              1682: *
              1683: *Search list of depths and times for any not
              1684: *yet sent to ship's logger.
              1685: *
CB35 DE 74    1686: SRCHDP   LDX   DPTMP1
CB37 86 05    1687:          LDA A #5
CB39 97 71    1688:          STA A SRCHCT
CB3B DF 76    1689: ST5      STX   DPTMP2
CB3D A6 05    1690:          LDA A 5,X
CB3F 27 1D    1691:          BEQ   ST3
CB41 A6 07    1692:          LDA A 7,X
CB43 85 01    1693:          BIT A #1
CB45 27 17    1694:          BEQ   ST3
CB47 84 FE    1695:          AND A #$FE          Clear LSB for TX.
CB49 A7 07    1696:          STA A 7,X
CB4B 8D 25    1697:          BSR   XMIT
CB4D 25 06    1698:          BCS   ST4
CB4F 7A 0071  1699:          DEC   SRCHCT
CB52 26 0A    1700:          BNE   ST3
CB54 39       1701:          RTS
CB55 DE 76    1702: ST4      LDX   DPTMP2
CB57 A6 07    1703:          LDA A 7,X
CB59 8A 01    1704:          ORA A #1
CB5B A7 07    1705:          STA A 7,X
CB5D 39       1706:          RTS
CB5E DE 76    1707: ST3      LDX   DPTMP2
CB60 86 08    1708:          LDA A #8
CB62 BD C3DB  1709:          JSR   ADDA
CB65 8C 4000  1710:          CPX   #DPTMST+$2000
CB68 26 03    1711:          BNE   ST6
CB6A CE 2000  1712:          LDX   #DPTMST
CB6D 9C 74    1713: ST6      CPX   DPTMP1
CB6F 26 CA    1714:          BNE   ST5
CB71 39       1715:          RTS
              1716: *
              1717: *
              1718: *Try to send depth and time to ship's logger.
              1719: *
CB72 B6 E002  1720: XMIT     LDA A ACIAS
CB75 85 08    1721:          BIT A #%00001000
CB77 27 02    1722:          BEQ   XM1
CB79 0D       1723:          SEC
CB7A 39       1724:          RTS
CB7B 86 45    1725: XM1      LDA A #'E
CB7D BD C0A4  1726:          JSR   OUTCH
CB80 BD F17C  1727:          JSR   $F17C
CB83 BD F17C  1728:          JSR   $F17C
CB86 BD F17C  1729:          JSR   $F17C
CB89 BD F17E  1730:          JSR   $F17E
CB8C BD F17E  1731:          JSR   $F17E
CB8F 86 0D    1732:          LDA A #$0D
CB91 BD F108  1733:          JSR   $F108
CB94 86 0A    1734:          LDA A #$0A
CB96 BD F108  1735:          JSR   $F108
CB99 BD C0B1  1736:          JSR   GET
CB9C 4D       1737:          TST A
CB9D 26 02    1738:          BNE   XM2
CB9F 0D       1739:          SEC
```

```
CBA0 39        1740:          RTS
CBA1 0C        1741: XM2      CLC
CBA2 39        1742:          RTS
               1743: *
               1744: *Init Mufax display routine RASTAS
               1745: *
0002           1746: KLINES   EQU   2
0002           1747: KWIDTH   EQU   2
               1748: *
CBA3 CE CF40   1749: MFINIT   LDX   #PATRNS
CBA6 DF 6A     1750:          STX   ROWPT
CBA8 CE 0059   1751:          LDX   #MUFDSP
CBAB DF 68     1752:          STX   CHARPT
CBAD 86 80     1753:          LDA A #$80
CBAF 97 6D     1754:          STA A DOTS
CBB1 86 02     1755:          LDA A #KLINES
CBB3 97 6E     1756:          STA A LINES
CBB5 86 02     1757:          LDA A #KWIDTH
CBB7 97 6F     1758:          STA A WIDTH
CBB9 7F 0066   1759:          CLR   DSFRST
CBBC 96 BB     1760:          LDA A DELAY+1
CBBE D6 BA     1761:          LDA B DELAY
CBC0 DE BE     1762:          LDX   DATSIZ
CBC2 DF F4     1763:          STX   XTEMP
CBC4 74 00F4   1764:          LSR   XTEMP
CBC7 76 00F5   1765:          ROR   XTEMP+1
CBCA 9B F5     1766:          ADD A XTEMP+1
CBCC D9 F4     1767:          ADC B XTEMP
CBCE C1 07     1768:          CMP B #7
CBD0 22 06     1769:          BHI   MF0
CBD2 25 08     1770:          BCS   MFF1
CBD4 81 D0     1771:          CMP A #$D0
CBD6 23 04     1772:          BLS   MFF1
CBD8 80 D0     1773: MF0      SUB A #$D0
CBDA C2 07     1774:          SBC B #7
CBDC C1 04     1775: MFF1     CMP B #4
CBDE 25 03     1776:          BCS   MF1
CBE0 73 0066   1777:          COM   DSFRST
CBE3 39        1778: MF1      RTS
               1779: *
               1780: *PROCESS gated signal.
               1781: *Compilation of various processing routines.
               1782: *
CBE4 BD C5C1   1783: PROCSS   JSR   EDGE
CBE7 BD C677   1784:          JSR   PEAK
CBEA 96 91     1785:          LDA A NRSTHT
CBEC 91 9A     1786:          CMP A PKVAL
CBEE 24 06     1787:          BCC   PR1
CBF0 96 9C     1788:          LDA A PKPOSN+1
CBF2 D6 9B     1789:          LDA B PKPOSN
CBF4 20 04     1790:          BRA   PR2
CBF6 96 93     1791: PR1      LDA A NRSTPS+1
CBF8 D6 92     1792:          LDA B NRSTPS
CBFA 97 CB     1793: PR2      STA A NWDPTH+1
CBFC D7 CA     1794:          STA B NWDPTH
CBFE BD C731   1795:          JSR   SHFTGT
CC01 BD C7B6   1796:          JSR   FILT
CC04 97 BB     1797:          STA A DELAY+1
CC06 D7 BA     1798:          STA B DELAY
CC08 BD C592   1799:          JSR   ADJUST
```

```
CC0B 7E C9CD 1800:         JMP   DISP1
             1801: *
             1802: *Test for TX when gating on.
             1803: *
CC0E 7D 00B2 1804: GATEST  TST   GATNFG
CC11 27 4D   1805:         BEQ   GT8
CC13 C6 07   1806:         LDA B #7
CC15 86 D0   1807:         LDA A #$D0
CC17 90 BB   1808:         SUB A DELAY+1
CC19 D2 BA   1809:         SBC B DELAY
CC1B 90 BF   1810:         SUB A DATSIZ+1
CC1D D2 BE   1811:         SBC B DATSIZ
CC1F 2A 3F   1812:         BPL   GT8
CC21 9B BF   1813:         ADD A DATSIZ+1
CC23 D9 BE   1814:         ADC B DATSIZ
CC25 DE 9D   1815:         LDX   FIRSTX
CC27 DF F4   1816:         STX   XTEMP
CC29 9B F5   1817:         ADD A XTEMP+1
CC2B D9 F4   1818:         ADC B XTEMP
CC2D 97 F5   1819:         STA A XTEMP+1
CC2F D7 F4   1820:         STA B XTEMP
CC31 DE F4   1821:         LDX   XTEMP
CC33 7F 00A3 1822:         CLR   SUMK
CC36 7F 00A4 1823:         CLR   SUMK+1
CC39 4F      1824:         CLR A
CC3A 5F      1825:         CLR B
CC3B A6 00   1826: GT1     LDA A 0,X
CC3D 08      1827:         INX
CC3E 5C      1828:         INC B
CC3F 9B A4   1829:         ADD A SUMK+1
CC41 24 03   1830:         BCC   GT2
CC43 7C 00A3 1831:         INC   SUMK
CC46 97 A4   1832: GT2     STA A SUMK+1
CC48 9C 9F   1833:         CPX   LASTX
CC4A 27 04   1834:         BEQ   GT3
CC4C C1 14   1835:         CMP B #20
CC4E 25 EB   1836:         BCS   GT1
CC50 96 A4   1837: GT3     LDA A SUMK+1
CC52 36      1838:         PSH A
CC53 96 A3   1839:         LDA A SUMK
CC55 36      1840:         PSH A
CC56 37      1841:         PSH B
CC57 BD C70B 1842:         JSR   DIVDBL
CC5A 31      1843:         INS
CC5B 32      1844:         PUL A
CC5C 31      1845:         INS
CC5D 91 70   1846:         CMP A THRHLD
CC5F 39      1847:         RTS
CC60 0D      1848: GT8     SEC
CC61 39      1849:         RTS
             1850: *
             1851: *Raster display subroutine.
             1852: *
CC62 DE 68   1853: RASTAS  LDX   CHARPT
CC64 A6 00   1854:         LDA A 0,X
CC66 81 0D   1855:         CMP A #$D
CC68 27 41   1856:         BEQ   RAS4
CC6A 81 20   1857:         CMP A #$20
CC6C 26 02   1858:         BNE   RAS0
CC6E 86 3F   1859:         LDA A #$3F
```

```
CC70 80 30    1860: RAS0   SUB A #'0
CC72 84 0F    1861:        AND A #$F
CC74 DE 6A    1862:        LDX   ROWPT
CC76 BD C3DB  1863:        JSR   ADDA
CC79 A6 00    1864:        LDA A 0,X
CC7B 94 6D    1865:        AND A DOTS
CC7D 27 0B    1866:        BEQ   RAS1
CC7F 96 BC    1867:        LDA A FLAGS
CC81 84 EF    1868:        AND A #$EF
CC83 B7 8008  1869:        STA A DAC
CC86 97 BC    1870:        STA A FLAGS
CC88 20 09    1871:        BRA   RAS2
CC8A 96 BC    1872: RAS1   LDA A FLAGS
CC8C 8A 10    1873:        ORA A #$10
CC8E B7 8008  1874:        STA A DAC
CC91 97 BC    1875:        STA A FLAGS
CC93 7A 006F  1876: RAS2   DEC   WIDTH
CC96 26 12    1877:        BNE   RAS3
CC98 86 02    1878:        LDA A #KWIDTH
CC9A 97 6F    1879:        STA A WIDTH
CC9C 74 006D  1880:        LSR   DOTS
CC9F 26 09    1881:        BNE   RAS3
CCA1 86 80    1882:        LDA A #$80
CCA3 97 6D    1883:        STA A DOTS
CCA5 DE 68    1884:        LDX   CHARPT
CCA7 08       1885:        INX
CCA8 DF 68    1886:        STX   CHARPT
CCAA 39       1887: RAS3   RTS
CCAB 7F 006C  1888: RAS4   CLR   DISPMD
CCAE C6 04    1889:        LDA B #4
CCB0 F7 8006  1890:        STA B ADC+2
CCB3 7D 8004  1891:        TST   ADC
CCB6 D6 BC    1892:        LDA B FLAGS
CCB8 C4 17    1893:        AND B #%10111
CCBA D7 BC    1894:        STA B FLAGS
CCBC 7A 006E  1895:        DEC   LINES
CCBF 26 0E    1896:        BNE   RAS5
CCC1 86 02    1897:        LDA A #KLINES
CCC3 97 6E    1898:        STA A LINES
CCC5 96 6B    1899:        LDA A ROWPT+1
CCC7 8B 10    1900:        ADD A #$10
CCC9 97 6B    1901:        STA A ROWPT+1
CCCB 81 C0    1902:        CMP A #$C0
CCCD 27 0A    1903:        BEQ   RAS6
CCCF CE 0059  1904: RAS5   LDX   #MUFDSP
CCD2 DF 68    1905:        STX   CHARPT
CCD4 86 80    1906:        LDA A #$80
CCD6 97 6D    1907:        STA A DOTS
CCD8 39       1908:        RTS
CCD9 96 BC    1909: RAS6   LDA A FLAGS
CCDB 8A 08    1910:        ORA A #%1000
CCDD 97 BC    1911:        STA A FLAGS
CCDF B7 8008  1912:        STA A DAC
CCE2 86 FF    1913:        LDA A #$FF
CCE4 97 67    1914:        STA A DISPDN
CCE6 39       1915:        RTS
              1916: *
              1917: *Send depth and seconds to remote K/B
              1918: *then accept new depth or time entry
              1919: *
```

```
CCE7 B6 E002 1920: SEND   LDA A ACIAS
CCEA 85 08   1921:        BIT A #%00001000
CCEC 27 0D   1922:        BEQ   SN2
CCEE 7D 0058 1923: SN1    TST   ER6CNT
CCF1 26 04   1924:        BNE   SN10
CCF3 86 06   1925:        LDA A #6
CCF5 97 D3   1926:        STA A ERRORS+5
CCF7 7C 0058 1927: SN10   INC   ER6CNT
CCFA 39      1928:        RTS
CCFB 86 53   1929: SN2    LDA A #'S
CCFD BD C0A4 1930:        JSR   OUTCH
CD00 CE 00DB 1931:        LDX   #DIGITS+1
CD03 A6 00   1932: SN3    LDA A 0,X
CD05 BD C0A4 1933:        JSR   OUTCH
CD08 08      1934:        INX
CD09 8C 00DF 1935:        CPX   #DIGITS+5
CD0C 26 F5   1936:        BNE   SN3
CD0E 86 3F   1937:        LDA A #'?
CD10 BD C0A4 1938:        JSR   OUTCH
CD13 BD C0A4 1939:        JSR   OUTCH
CD16 CE 00E9 1940:        LDX   #SECS+1
CD19 BD F17E 1941:        JSR   $F17E
CD1C CE C34D 1942:        LDX   #MESS2+10
CD1F BD C1C0 1943:        JSR   TYPES
CD22 BD C0B1 1944:        JSR   GET
CD25 4D      1945:        TST A
CD26 27 C6   1946:        BEQ   SN1
CD28 81 0D   1947:        CMP A #$D
CD2A 26 01   1948:        BNE   SN4
CD2C 39      1949:        RTS
CD2D 81 44   1950: SN4    CMP A #'D
CD2F 26 26   1951:        BNE   SN6
             1952: *Remote depth entry
CD31 CE 0086 1953:        LDX   #BCDIN
CD34 BD C0B1 1954: SN5    JSR   GET
CD37 4D      1955:        TST A
CD38 27 66   1956:        BEQ   SN9
CD3A 84 0F   1957:        AND A #$F
CD3C A7 00   1958:        STA A 0,X
CD3E 08      1959:        INX
CD3F 8C 008A 1960:        CPX   #BCDIN+4
CD42 26 F0   1961:        BNE   SN5
CD44 BD C0B1 1962:        JSR   GET
CD47 81 0D   1963:        CMP A #$D
CD49 26 55   1964:        BNE   SN9
CD4B BD C89A 1965:        JSR   CVDTB
CD4E D7 F4   1966:        STA B XTEMP
CD50 97 F5   1967:        STA A XTEMP+1
CD52 DE F4   1968:        LDX   XTEMP
CD54 7E C9C3 1969:        JMP   IND2
             1970: *Remote time entry
CD57 CE 0086 1971: SN6    LDX   #BCDIN
CD5A BD C0B1 1972: SN7    JSR   GET
CD5D 4D      1973:        TST A
CD5E 27 40   1974:        BEQ   SN9
CD60 84 0F   1975:        AND A #$F
CD62 A7 00   1976:        STA A 0,X
CD64 08      1977:        INX
CD65 8C 008A 1978:        CPX   #BCDIN+4
CD68 26 F0   1979:        BNE   SN7
```

```
CD6A 96 86    1980:          LDA A BCDIN
CD6C 48       1981:          ASL A
CD6D 48       1982:          ASL A
CD6E 48       1983:          ASL A
CD6F 48       1984:          ASL A
CD70 9A 87    1985:          ORA A BCDIN+1
CD72 97 E5    1986:          STA A HRS+1
CD74 96 88    1987:          LDA A BCDIN+2
CD76 48       1988:          ASL A
CD77 48       1989:          ASL A
CD78 48       1990:          ASL A
CD79 48       1991:          ASL A
CD7A 9A 89    1992:          ORA A BCDIN+3
CD7C 97 E7    1993:          STA A MINS+1
CD7E CE 0086  1994:          LDX   #BCDIN
CD81 BD C0B1  1995: SN8      JSR   GET
CD84 4D       1996:          TST A
CD85 27 19    1997:          BEQ   SN9
CD87 84 0F    1998:          AND A #$F
CD89 A7 00    1999:          STA A 0,X
CD8B 08       2000:          INX
CD8C 8C 008A  2001:          CPX   #BCDIN+4
CD8F 26 F0    2002:          BNE   SN8
CD91 96 87    2003:          LDA A BCDIN+1
CD93 97 E2    2004:          STA A DAYS
CD95 96 88    2005:          LDA A BCDIN+2
CD97 48       2006:          ASL A
CD98 48       2007:          ASL A
CD99 48       2008:          ASL A
CD9A 48       2009:          ASL A
CD9B 9A 89    2010:          ORA A BCDIN+3
CD9D 97 E3    2011:          STA A DAYS+1
CD9F 39       2012:          RTS
CDA0 86 07    2013: SN9      LDA A #7
CDA2 97 D4    2014:          STA A ERRORS+6
CDA4 39       2015:          RTS
              2016: *
              2017: *
              2018: ****************************************
              2019: *
              2020: *MAIN PROGRAM
              2021: *
CDA5 BD C000  2022:          JSR   INIT          Initiallise I/O and CLOCK
CDA8 CE C3EF  2023:          LDX   #INT
CDAB FF A000  2024:          STX   IOVECT        Set up I/O interrupt add.
CDAE CE 03E8  2025:          LDX   #1000
CDB1 DF BA    2026:          STX   DELAY         Set delay to 1 Sec.
CDB3 4F       2027:          CLR A
CDB4 CE 2000  2028:          LDX   #$2000        DPTMST
CDB7 A7 00    2029: MN0      STA A 0,X
CDB9 08       2030:          INX
CDBA 8C 4000  2031:          CPX   #$4000        DPMTST+$2000
CDBD 26 F8    2032:          BNE   MN0
CDBF CE 00CE  2033:          LDX   #ERRORS
CDC2 A7 00    2034: MN1      STA A 0,X           Clear ERRORS and flags.
CDC4 08       2035:          INX
CDC5 8C 00D5  2036:          CPX   #XT
CDC8 26 F8    2037:          BNE   MN1
CDCA 97 6C    2038:          STA A DISPMD
CDCC 97 65    2039:          STA A DISPEN
```

```
CDCE 97 C4    2040:          STA A SCALE
CDD0 97 A8    2041:          STA A FLSTRG
CDD2 97 CD    2042:          STA A SLOPE
CDD4 97 C8    2043:          STA A ODDMIN
CDD6 97 58    2044:          STA A ER6CNT
CDD8 86 01    2045:          LDA A #1          Set flags.
CDDA 97 C7    2046:          STA A HRFLG
CDDC 97 C9    2047:          STA A EVNMIN
CDDE 97 CC    2048:          STA A PHASE
CDE0 97 67    2049:          STA A DISPDN
CDE2 86 13    2050:          LDA A #%10011     Mask pattern for clock
CDE4 97 BC    2051:          STA A FLAGS       and trigger interrupts.
CDE6 BD C2CE  2052:          JSR   GETTM       Get time and date.
CDE9 CE 1000  2053:          LDX   #DPSTRT
CDEC DF 82    2054:          STX   DPSTCK
CDEE CE 2000  2055:          LDX   #DPTMST
CDF1 DF 74    2056:          STX   DPTMP1
CDF3 DF 76    2057:          STX   DPTMP2
CDF5 CE 0080  2058:          LDX   #$80        No of reverb samples.
CDF8 DF 72    2059:          STX   RVBSIZ
CDFA CE 0100  2060:          LDX   #$100       Initial echo gate length.
CDFD DF BE    2061:          STX   DATSIZ
CDFF CE 0080  2062:          LDX   #$80        PREDICtion = centre of gate.
CE02 DF 94    2063:          STX   PREDIC
CE04 CE 6DB6  2064:          LDX   #$6DB6
CE07 DF B3    2065:          STX   COEF1
CE09 CE B334  2066:          LDX   #$B334
CE0C DF B5    2067:          STX   COEF2       Set up FILTer coefficients.
CE0E 7F 00B2  2068:          CLR   GATNFG
CE11 CE 0200  2069:          LDX   #MEMBEG
CE14 DF C5    2070:          STX   DESTAD
CE16 DF 9D    2071:          STX   FIRSTX
CE18 BD C3CA  2072:          JSR   ADDX        Add DATSIZ
CE1B DF 9F    2073:          STX   LASTX
CE1D DE BA    2074:          LDX   DELAY
CE1F DF B7    2075:          STX   AVDPTH      "Charge up" FILTer
CE21 7F 00AD  2076:          CLR   SMDONE
CE24 01       2077:          NOP
CE25 0E       2078:          CLI               We're off. Interrupts enabled.
CE26 BD C947  2079: MN2      JSR   INDPTH      Waiting for sampling to be done.
CE29 7D 00AD  2080:          TST   SMDONE
CE2C 27 F8    2081:          BEQ   MN2
CE2E CE 0200  2082: MN3      LDX   #MEMBEG     Get ready for next sweep.
CE31 DF C5    2083:          STX   DESTAD
CE33 7F 00AD  2084:          CLR   SMDONE
CE36 B6 8004  2085:          LDA A ADC
CE39 43       2086:          COM A
CE3A 84 01    2087:          AND A #1
CE3C 97 B2    2088:          STA A GATNFG
CE3E BD CC0E  2089:          JSR   GATEST
CE41 24 0E    2090:          BCC   MN5
CE43 BD CBE4  2091:          JSR   PROCSS      Process recent data.
CE46 7D 00C8  2092:          TST   ODDMIN
CE49 27 06    2093:          BEQ   MN5
CE4B BD CAA3  2094:          JSR   STORDP
CE4E 7F 00C8  2095:          CLR   ODDMIN
CE51 BD C947  2096: MN5      JSR   INDPTH
CE54 7D 00AD  2097:          TST   SMDONE
CE57 27 F8    2098:          BEQ   MN5
CE59 20 D3    2099:          BRA   MN3
```

```
2100: *
2101: *
2102:          END
NO ERROR(S) DETECTED
```

SYMBOL TABLE:

ACIAS	E002	ADBA	C3E1	ADC	8004	ADDA	C3DB
ADDX	C3CA	ADJ1	C59F	ADJ2	C5A6	ADJ3	C5AD
ADJ4	C5B8	ADJUST	C592	ATEMP	00B9	AVDPTH	00B7
BC3	C887	BCDIN	0086	BTEMP	00D9	CHARPT	0068
CLK1	C429	CLK2	C433	CLK3	C447	CLOCK	C0E8
COEF1	00B3	COEF2	00B5	COEFFS	CFC0	CONST	C17A
CR1	CA0B	CR2	CA1D	CR3	CA47	CR4	CA70
CR5	CA88	CR6	CA18	CR7	CA42	CRR1	CA32
CRR2	CA36	CRR3	CA98	CRR4	CAA2	CVBIN1	C8A4
CVBIN2	C8B6	CVBIN3	C8C8	CVBTD	C13A	CVDEC1	C148
CVDEC2	C14B	CVDEC3	C156	CVDTB	C89A	CVR1	C91B
CVR2	C91E	CVRTMS	C8F8	DAC	8008	DATCNT	00C0
DATSIZ	00BE	DAYS	00E2	DELAY	00BA	DESTAD	00C5
DG1	C1B8	DG2	C1BA	DG3	C18A	DG4	C18F
DIG	00D7	DIGITS	00DA	DIRFLG	0081	DISP1	C9CD
DISPDN	0067	DISPEN	0065	DISPMD	006C	DIVDBL	C70B
DOTS	006D	DPFILT	C9F3	DPSTCK	0082	DPSTRT	1000
DPTMP1	0074	DPTMP2	0076	DPTMST	2000	DSPRST	0066
DV1	C730	E1	C5D3	E2	C5EC	E3	C60C
E4	C631	E5	C616	EDGE	C5C1	EGEND	00A5
ER1	C2AB	ER2	C2BC	ER3	C2C5	ER6CNT	0058
ERMESS	C350	ERR1	C35E	ERR2	C36E	ERR3	C380
ERR4	C391	ERR5	C3A4	ERR6	C3AE	ERR7	C3BD
ERRORS	00CE	ERRS	C2A3	EVNMIN	00C9	FILT	C7B6
FILTOP	0078	FIRSTX	009D	FLAGS	00BC	FLSTRG	00A8
FRSTTX	00AE	G1	C0B4	G2	C0C0	G3	C0C5
GATEST	CC0E	GATNFG	00B2	GET	C0B1	GETLFT	C934
GETR1	C93C	GETR2	C946	GETRGT	C93B	GETTM	C2CE
GT1	CC3B	GT2	CC46	GT3	CC50	GT8	CC60
HD1	C1D6	HD2	C1D9	HEAD1	C1F7	HEAD2	C21A
HEADNG	C1DE	HRFLG	00C7	HRHEAD	C1C7	HRS	00E4
I1	C051	I2	C05B	INBCD	C885	INCH	C099
IND0	C94F	IND1	C9A1	IND2	C9C3	IND3	C9CC
INDEC	C184	INDIG	C1AD	INDPTH	C947	INIT	C000
INT	C3EF	IOVECT	A000	J0	C40E	J1	C413
J13	C503	J14	C511	J15	C51B	J16	C51F
J18	C578	J19	C57D	J2	C44B	J20	C589
J21	C58C	J3	C451	J4	C48E	J5	C4B9
J6	C4C9	J8	C539	J9	C54D	KLINES	0002
KWIDTH	0002	LASTX	009F	LC1	C8E3	LCD	C8D1
LINES	006E	LOOPCT	008F	LSTENT	0084	MEMBEG	0200
MESS1	C307	MESS2	C343	MF0	CBD8	MF1	CBE3
MFF1	CBDC	MFINIT	CBA3	MINS	00E6	MN0	CDB7
MN1	CDC2	MN2	CE26	MN3	CE2E	MN5	CE51
MP1	C884	MPY1X3	C7DD	MPY2X2	C81D	MPYDBL	C6E5
MUFDSP	0059	MULCLR	0079	MULCOM	0071	MULT	00FC
NRSTHT	0091	NRSTPS	0092	NWDPTH	00CA	ODDMIN	00C8
OFFSET	0098	OUT	C0D5	OUT2H	C0D7	OUT2HS	C0E2
OUT4HS	C0E0	OUTC1	C0A5	OUTCH	C0A4	OUTHL	C0C7
OUTHR	C0CB	OUTS	C0E4	OVFLW	00C2	P1	C074
P2	C07F	P3	C08C	P4	C08F	P5	C071
PANEL	8020	PATRNS	CF40	PD1	C246	PD2	C24A

PD3	C254	PD4	C296	PEAK	C677	PHASE	00CC
PHTEMP	00A7	PK1	C68A	PK2	C694	PK3	C69A
PK4	C6A2	PK5	C6BF	PK6	C6CB	PK7	C6D5
PK8	C6E1	PKPOSN	009B	PKVAL	009A	PNTR1	007D
PNTR2	007F	PR1	CBF6	PR2	CBFA	PRDAT	C255
PREDIC	0094	PRINTC	C06F	PRINTR	8021	PRINTS	C092
PROCSS	CBE4	RAS0	CC70	RAS1	CC8A	RAS2	CC93
RAS3	CCAA	RAS4	CCAB	RAS5	CCCF	RAS6	CCD9
RASTAS	CC62	RESET	0070	ROWPT	006A	RVBSIZ	0072
RVRBCT	00A9	SAM	C4DF	SAVX	0096	SCALE	00C4
SCNDTX	00B0	SCTEMP	0090	SECS	00E8	SEND	CCE7
SH1	C750	SH2	C75A	SH3	C765	SHFTGT	C731
SIGN	007C	SK	00A1	SLOPE	00CD	SM1	C4ED
SMDONE	00AD	SN1	CCEE	SN10	CCF7	SN2	CCFB
SN3	CD03	SN4	CD2D	SN5	CD34	SN6	CD57
SN7	CD5A	SN8	CD81	SN9	CDA0	SPTEMP	00F6
SRCHCT	0071	SRCHDP	CB35	ST1	CAAE	ST2	CB0B
ST3	CB5E	ST4	CB55	ST5	CB3B	ST6	CB6D
STORDP	CAA3	SUMK	00A3	T1	C10C	T2	C117
T3	C12A	T4	C137	TENPWR	C8C9	THRHLD	0070
TIM	C4CC	TIMER	8040	TP1	C1BC	TRG1	C46F
TRGFLG	00BD	TRIG	C457	TT1	C107	TV1	C655
TV2	C659	TV3	C665	TV4	C674	TV5	C676
TVG	C638	TWOMIN	C236	TXNO	008E	TXSAMP	00AB
TYPES	C1C0	WEIGHT	C76A	WG1	C782	WG2	C788
WG3	C796	WG4	C7A8	WIDTH	006F	XM1	CB7B
XM2	CBA1	XMIT	CB72	XSTOR	0063	XT	00D5
XTEMP	00F4	YEARS	00E0				

```
                     1:          NAM    PSCOEF
                     2:          OPT    NOG,PAG
                     3: *
                     4: *PEST3 RASTER DOT PATTERNS AND DEPTH FILTER
                     5: *COEFFICIENTS.
                     6: *
CF40                 7:          ORG    $CF40
                     8: *
                     9: *ROW 0
CF40                10: PATRNS   EQU    *
CF40 60             11:          FCB    $60,$20,$70,$70,$10,$F8,$30,$F8
CF48 70             12:          FCB    $70,$70,$00,$60,$08,$00,$80,$00
                    13: *ROW 1
CF50 90             14:          FCB    $90,$60,$88,$88,$30,$80,$40,$08
CF58 88             15:          FCB    $88,$88,$60,$60,$10,$00,$40,$00
                    16: *ROW 2
CF60 90             17:          FCB    $90,$20,$08,$08,$50,$F0,$80,$10
CF68 88             18:          FCB    $88,$88,$60,$00,$20,$F8,$20,$00
                    19: *ROW 3
CF70 90             20:          FCB    $90,$20,$70,$30,$90,$08,$F0,$20
CF78 70             21:          FCB    $70,$78,$00,$60,$40,$00,$10,$00
                    22: *ROW 4
CF80 90             23:          FCB    $90,$20,$80,$08,$F8,$08,$88,$40
CF88 88             24:          FCB    $88,$08,$60,$60,$20,$F8,$20,$00
                    25: *ROW 5
CF90 90             26:          FCB    $90,$20,$80,$88,$10,$88,$88,$80
CF98 88             27:          FCB    $88,$10,$60,$20,$10,$00,$40,$00
                    28: *ROW 6
CFA0 60             29:          FCB    $60,$70,$F8,$70,$10,$70,$70,$80
CFA8 70             30:          FCB    $70,$60,$00,$40,$08,$00,$80,$00
                    31: *ROW 7
CFB0 00             32:          FCB    $00,$00,$00,$00,$00,$00,$00,$00
CFB8 00             33:          FCB    $00,$00,$00,$00,$00,$00,$00,$00
                    34: *
                    35: *
                    36: *DEPTH FILTER COEFFICIENTS
                    37: *
CFC0                38: COEFFS   EQU    *
CFC0 00 00          39:          FDB    $0000
CFC2 00 00          40:          FDB    $0000
CFC4 FF D3          41:          FDB    $FFD3
CFC6 FF D5          42:          FDB    $FFD5
CFC8 00 3B          43:          FDB    $003B
CFCA 00 7D          44:          FDB    $007D
CFCC 00 00          45:          FDB    $0000
CFCE FF 43          46:          FDB    $FF43
CFD0 FF 74          47:          FDB    $FF74
CFD2 00 A4          48:          FDB    $00A4
CFD4 01 35          49:          FDB    $0135
CFD6 00 00          50:          FDB    $0000
CFD8 FE 66          51:          FDB    $FE66
CFDA FE DE          52:          FDB    $FEDE
CFDC 01 4A          53:          FDB    $014A
CFDE 02 61          54:          FDB    $0261
CFE0 00 00          55:          FDB    $0000
CFE2 FC EB          56:          FDB    $FCEB
CFE4 FD D4          57:          FDB    $FDD4
CFE6 02 7C          58:          FDB    $027C
CFE8 04 9D          59:          FDB    $049D
```

```
---

PSCOEF                                  SSB MNEMONIC ASSEMBLER

CFEA 00 00      60:         FDB   $0000
CFEC F9 CD      61:         FDB   $F9CD
CFEE FB 7E      62:         FDB   $FB7E
CFF0 05 60      63:         FDB   $0560
CFF2 0A 92      64:         FDB   $0A92
CFF4 00 00      65:         FDB   $0000
CFF6 EE E0      66:         FDB   $EEE0
CFF8 F1 5E      67:         FDB   $F15E
CFFA 16 BC      68:         FDB   $16BC
CFFC 4C 1C      69:         FDB   $4C1C
CFFE 67 EC      70:         FDB   $67EC       MIDDLE
                71: *REMAINDER IS MIRROR IMAGE
                72: *
                73:         END
     NO ERROR(S) DETECTED


   SYMBOL TABLE:
COEFFS  CFC0     PATRNS  CF40
```

```
                    1:          NAM   REMOTE
                    2: *
                    3: *REMOTE ENTRY TERMINAL FOR PES DIGITAL TRACKER
                    4: *
C000                5: KEYBD    EQU   $C000       Keypad
0000                6: LEDS     EQU   $0000       8 digit LED display
0001                7: PORT2    EQU   $1
0010                8: PORT3    EQU   $10
00FC                9: SP       EQU   $FC
00FE               10: IOV      EQU   $FE
                   11: *
0080               12:          ORG   $0080
0080               13: DATA1    RMB   8           Register for PEST depth
0088               14: DATA2    RMB   8           Keyboard entry register
0090               15: BUFFER   RMB   9           Storage for TX from PEST
0099               16: XTEMP    RMB   2
009B               17: TMENFG   RMB   1           Time entered flag
009C               18: DPENFG   RMB   1           Depth entered flag
009D               19: BUFPNT   RMB   2
                   20: *
E000               21:          ORG   $E000
E000 7E E1C4       22:          JMP   INTSRV
E003 8E 00E8       23: INIT     LDS   #$E8        Here on POWER-ON reset.
E006 9F FC         24:          STS   SP            -
E008 86 01         25:          LDA A #1
E00A 97 01         26:          STA A PORT2       Bit 0 as O/P = RTS
E00C 4F            27:          CLR A
E00D 97 03         28:          STA A PORT2+2     Set NOT RTS low
E00F CE 0C1A       29:          LDX   #$0C1A      Enable interrupt on RX
E012 DF 10         30:          STX   PORT3
E014 7F C000       31:          CLR   KEYBD       pragram I/O ports
E017 C6 06         32:          LDA B #6
E019 F7 C002       33:          STA B KEYBD+2
E01C C6 FF         34:          LDA B #$FF
E01E D7 00         35:          STA B LEDS
                   36: *
E020 CE 0080       37: CLEAR    LDX   #DATA1      Fill both registers with
E023 86 0F         38:          LDA A #$F         $F = blank
E025 A7 00         39: CL1      STA A 0,X
E027 08            40:          INX
E028 8C 0099       41:          CPX   #BUFFER+9
E02B 26 F8         42:          BNE   CL1
E02D 7F 009B       43:          CLR   TMENFG
E030 7F 009C       44:          CLR   DPENFG
E033 CE 0090       45:          LDX   #BUFFER
E036 DF 9D         46:          STX   BUFPNT
E038 0E            47:          CLI               Allow serial input interrupt.
E039 BD E085       48: START    JSR   NRMDSP      Normal display routine
E03C 81 09         49:          CMP A #9          display received depth
E03E 22 26         50:          BHI   COMMND      Is it a command key?
E040 CE 0088       51: ENT1     LDX   #DATA2      No. slide entry along one
E043 E6 01         52: ENT4     LDA B 1,X         digit towards left
E045 E7 00         53:          STA B 0,X
E047 08            54:          INX
E048 8C 008F       55:          CPX   #DATA2+7
E04B 26 F6         56:          BNE   ENT4
E04D A7 00         57:          STA A 0,X         Insert new digit at RHS
E04F C6 00         58: ENT2     LDA B #0          Set time-out count
E051 BD E0AD       59: ENT3     JSR   DSPENT      Display entry
```

```
E054 27 E3      60:          BEQ   START        Timed-out?
E056 7D C002    61:          TST   KEYBD+2      No. Then has key been pressed?
E059 2A F6      62:          BPL   ENT3         No
E05B B6 C000    63:          LDA A KEYBD        Yes. Get key code
E05E 84 0F      64:          AND A #$F          Mask LSBits
E060 8D 54      65:          BSR   DECODE       Decode it
E062 81 09      66:          CMP A #9           Command?
E064 23 DA      67:          BLS   ENT1         If no then enter new digit
                68: *
                69: *HERE IF KEY IS A COMMAND
                70: *
E066 80 0A      71: COMMND   SUB A #$A
E068 26 03      72:          BNE   CM1
E06A 7E E188    73:          JMP   ENTDP        Code A. Enter depth
E06D 4A         74: CM1      DEC A
E06E 26 03      75:          BNE   CM2
E070 7E E197    76:          JMP   ENTTM        Code B. Enter time and day no.
E073 4A         77: CM2      DEC A
E074 26 03      78:          BNE   CM3
E076 7E E1A6    79:          JMP   SENDDP       Code C. Send depth
E079 4A         80: CM3      DEC A
E07A 26 03      81:          BNE   CM4
E07C 7E E1AC    82:          JMP   SENDTM       Code D. Send time and Day No.
E07F 4A         83: CM4      DEC A
E080 26 CD      84:          BNE   ENT2         Code E. Recall entry
E082 7E E1B2    85:          JMP   CLRENT       Code F. Clear entry
                86: *
                87: *DISPLAY DEPTH SENT FROM PEST
                88: *
E085 7D C002    89: NRMDSP   TST   KEYBD+2      Check for key entry
E088 2B 07      90:          BMI   RET1         Skip if key pressed
E08A CE 0080    91:          LDX   #DATA1       Otherwise keep displaying
E08D 8D 0A      92:          BSR   DISP
E08F 20 F4      93:          BRA   NRMDSP
E091 B6 C000    94: RET1     LDA A KEYBD        Get key code
E094 84 0F      95:          AND A #$F          Mask LSBits
E096 8D 1E      96:          BSR   DECODE       Decode it
E098 39         97:          RTS
                98: *
                99: *SUBR. TO DISPLAY EIGHT DIGITS
               100: *
E099 C6 F0     101: DISP     LDA B #$F0         Get digit select code
E09B A6 00     102: DSP1     LDA A 0,X          Get digit value
E09D 1B        103:          ABA                Combine them
E09E 97 02     104:          STA A LEDS+2       O/P to LEDs
E0A0 86 FF     105:          LDA A #$FF
E0A2 4A        106: DSP2     DEC A              Hold during count-down
E0A3 26 FD     107:          BNE   DSP2         of Acc. A
E0A5 08        108:          INX
E0A6 C0 10     109:          SUB B #$10         Advance to next digit
E0A8 C1 70     110:          CMP B #$70
E0AA 26 EF     111:          BNE   DSP1
E0AC 39        112:          RTS
               113: *
               114: *DISPLAY ENTRY REGISTER
               115: *
E0AD CE 0088   116: DSPENT   LDX   #DATA2
E0B0 37        117:          PSH B              Acc. B contains time-out
E0B1 8D E6     118:          BSR   DISP         delay
E0B3 33        119:          PUL B
```

```
E0B4 5A          120:          DEC B
E0B5 39          121:          RTS
                 122: *
                 123: *CHANGE KEY CODE TO MATCH CALCULATOR
                 124: *TYPE LAY-OUT
                 125: *
E0B6 DF 99       126: DECODE   STX    XTEMP
E0B8 37          127:          PSH B
E0B9 CE E0C4     128:          LDX    #TABLE
E0BC 16          129:          TAB
E0BD 3A          130:          FCB    $3A          ABX Add Acc. B to X reg.
E0BE A6 00       131:          LDA A 0,X
E0C0 DE 99       132:          LDX    XTEMP
E0C2 33          133:          PUL B
E0C3 39          134:          RTS
                 135: *
E0C4 0A          136: TABLE    FCB    $A,7,8,9,$B,4,5,6,$C,1,2,3,$D,0,$E,$F
E0C5 07 08
E0C7 09 0B
E0C9 04 05
E0CB 06 0C
E0CD 01 02
E0CF 03 0D
E0D1 00 0E
E0D3 0F
                 137: *
```

```
E0D4 CE 0088     138: ERROR    LDX    #DATA2       Generate error display pattern
E0D7 C6 0C       139:          LDA B #$C
E0D9 E7 00       140:          STA B 0,X
E0DB E7 01       141:          STA B 1,X
E0DD E7 02       142:          STA B 2,X
E0DF E7 05       143:          STA B 5,X
E0E1 E7 06       144:          STA B 6,X
E0E3 E7 07       145:          STA B 7,X
E0E5 5F          146:          CLR B
E0E6 E7 03       147:          STA B 3,X          First digit of error no.
E0E8 A7 04       148:          STA A 4,X          Error no.
E0EA C6 00       149:          LDA B #0
E0EC BD E0AD     150: ER       JSR    DSPENT       Display it with time-out
E0EF 26 FB       151:          BNE    ER
E0F1 CE 0088     152:          LDX    #DATA2
E0F4 C6 0F       153:          LDA B #$F          then clear the register
E0F6 E7 00       154: ER1      STA B 0,X
E0F8 08          155:          INX
E0F9 8C 0090     156:          CPX    #DATA2+8
E0FC 26 F8       157:          BNE    ER1
E0FE 8E 00E8     158:          LDS    #$E8         Clean up unused returns
E101 7E E039     159:          JMP    START
                 160: *
                 161: *CHECK FORMAT OF DEPTH ENTRY
                 162: *
E104 CE 0088     163: DPFMAT LDX    #DATA2
```

```
E107 A6 00     164: DPF1   LDA A 0,X        First 4 places should be blanks
E109 27 04     165:        BEQ   DPF2       or zeroes
E10B 81 0F     166:        CMP A #$F        Depth limited to 4 figures
E10D 26 16     167:        BNE   DPF4
E10F 08        168: DPF2   INX
E110 8C 008C   169:        CPX   #DATA2+4
E113 26 F2     170:        BNE   DPF1
E115 A6 00     171: DPF3   LDA A 0,X        Next places may be zeroes
E117 27 06     172:        BEQ   DPF7       or blanks
E119 81 0F     173:        CMP A #$F
E11B 26 0C     174:        BNE   DPF5
E11D 6F 00     175:        CLR   0,X        If blank then set to zero
E11F 08        176: DPF7   INX
E120 8C 0090   177:        CPX   #DATA2+8   Got to end yet?
E123 26 F0     178:        BNE   DPF3       If all zero then
E125 86 01     179: DPF4   LDA A #1         Error #1
E127 20 AB     180:        BRA   ERROR      "FORMAT ERROR"
E129 CE 0088   181: DPF5   LDX   #DATA2     Indicate correct entry by
E12C C6 0F     182:        LDA B #$F        left justifying it
E12E A6 04     183: DPF6   LDA A 4,X
E130 A7 00     184:        STA A 0,X
E132 E7 04     185:        STA B 4,X
E134 08        186:        INX
E135 8C 008C   187:        CPX   #DATA2+4
E138 26 F4     188:        BNE   DPF6
E13A 39        189:        RTS
               190: *
               191: *CHECK FORMAT OF TIME ENTRY
               192: *
E13B CE 0088   193: TMFMAT LDX   #DATA2
E13E A6 01     194:        LDA A 1,X
E140 81 02     195:        CMP A #2         Check tens of hours
E142 22 3F     196:        BHI   TMF3       greater than 2?
E144 26 06     197:        BNE   TMF0       Error 1
E146 A6 02     198:        LDA A 2,X        If tens of hours = 2
E148 81 03     199:        CMP A #3         then units greater than 3?
E14A 22 37     200:        BHI   TMF3
E14C A6 03     201: TMF0   LDA A 3,X        Check tens of minutes
E14E 81 05     202:        CMP A #5
E150 22 31     203:        BHI   TMF3
E152 A6 05     204:        LDA A 5,X        Check hundreds of day no.
E154 81 03     205:        CMP A #3         Greater than 3?
E156 22 2B     206:        BHI   TMF3
E158 26 0E     207:        BNE   TMF1
E15A A6 06     208:        LDA A 6,X        If = 3 are tens
E15C 81 06     209:        CMP A #6         = 6?
E15E 22 23     210:        BHI   TMF3
E160 26 06     211:        BNE   TMF1
E162 A6 07     212:        LDA A 7,X        If yes then are units
E164 81 06     213:        CMP A #6         greater than 6 ?
E166 22 1B     214:        BHI   TMF3
E168 6D 05     215: TMF1   TST   5,X        Go through Day no. again
E16A 26 08     216:        BNE   TMF2
E16C 6D 06     217:        TST   6,X        to check that all digits
E16E 26 04     218:        BNE   TMF2       are not zero
E170 6D 07     219:        TST   7,X
E172 27 0F     220:        BEQ   TMF3
E174 A6 01     221: TMF2   LDA A 1,X        Successful entry
E176 A7 00     222:        STA A 0,X        left justify time
E178 08        223:        INX
```

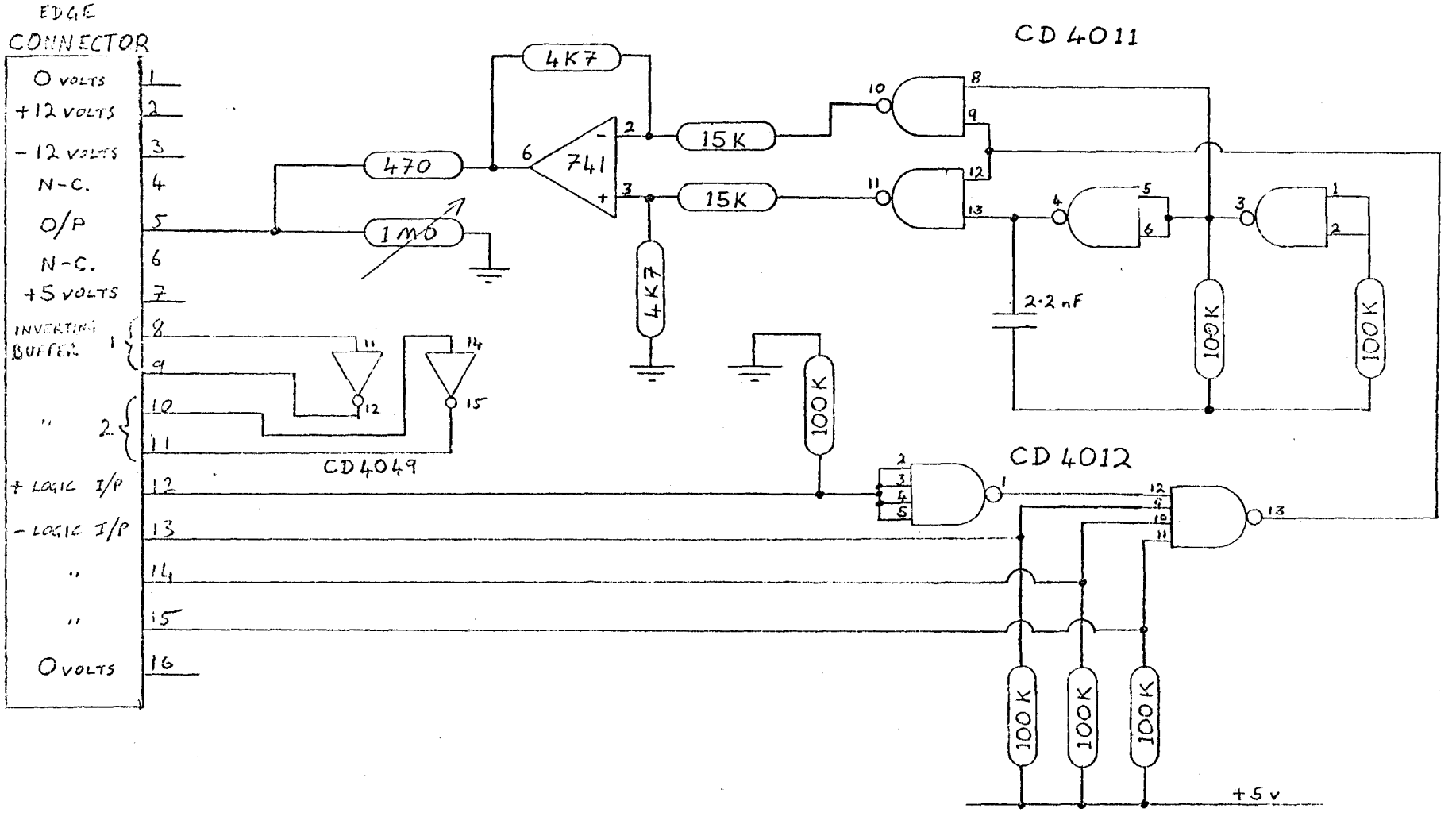
```
E179 8C 008C  224:          CPX   #DATA2+4
E17C 26 F6    225:          BNE   TMF2
E17E 86 0F    226:          LDA A #$F          and put blank between
E180 A7 00    227:          STA A 0,X          time and day no.
E182 39       228:          RTS
E183 86 01    229: TMF3     LDA A #1           No good. "FORMAT ERROR"
E185 7E E0D4  230:          JMP   ERROR
              231: *
E67F          232: OUTHR    EQU   $E67F
E6F0          233: SPC      EQU   $E6F0
              234: *
              235: *ENTER DEPTH
              236: *
E188 BD E104  237: ENTDP    JSR   DPFMAT       First check format
E18B 7F 009B  238:          CLR   TMENFG
E18E 86 FF    239:          LDA A #$FF
E190 97 9C    240:          STA A DPENFG
E192 C6 00    241:          LDA B #0
E194 7E E051  242:          JMP   ENT3
              243: *
              244: *ENTER TIME AND DAY NO.
              245: *
E197 BD E13B  246: ENTTM    JSR   TMFMAT       First check format
E19A 7F 009C  247:          CLR   DPENFG
E19D 86 FF    248:          LDA A #$FF
E19F 97 9B    249:          STA A TMENFG
E1A1 C6 00    250:          LDA B #0
E1A3 7E E051  251:          JMP   ENT3
              252: *
              253: *
              254: *SEND DEPTH
              255: *
E1A6 BD E104  256: SENDDP   JSR   DPFMAT       First check format
E1A9 7E E1B2  257:          JMP   CLRENT
              258: *
              259: *SEND TIME
              260: *
E1AC BD E13B  261: SENDTM   JSR   TMFMAT       First check format
E1AF 7E E1B2  262:          JMP   CLRENT
              263: *
              264: *CLEAR ENTRY REGISTER
              265: *
E1B2 CE 0088  266: CLRENT   LDX   #DATA2
E1B5 C6 0F    267:          LDA B #$F          Fill with blanks
E1B7 E7 00    268: CLR1     STA B 0,X
E1B9 08       269:          INX
E1BA 8C 0090  270:          CPX   #DATA2+8
E1BD 26 F8    271:          BNE   CLR1
E1BF C6 20    272:          LDA B #$20         Display for about 1/2 sec.
E1C1 7E E051  273:          JMP   ENT3
              274: *
              275: *
              276: *INTERRUPT SERVICE
              277: *PROGRAM ARRIVES HERE WHEN PEST TRANSMITS
              278: *A CHARACTER
              279: *
E686          280: OUTCH    EQU   $E686
              281: *
E1C4 7D 0011  282: INTSRV   TST   PORT3+1
E1C7 96 12    283:          LDA A PORT3+2      Get a character
```

```
E1C9 81 0D    284:          CMP A #$D          C/R?
E1CB 27 11    285:          BEQ   CR
E1CD 81 0A    286:          CMP A #$A          L/F?
E1CF 27 0C    287:          BEQ   RETURN
E1D1 DE 9D    288:          LDX   BUFPNT
E1D3 A7 00    289:          STA A 0,X          Store character in buffer
E1D5 08       290:          INX
E1D6 DF 9D    291:          STX   BUFPNT
E1D8 8C 0099  292:          CPX   #BUFFER+9
E1DB 27 08    293:          BEQ   FINI         Buffer full?
E1DD 3B       294: RETURN   RTI                No.Return from interrupt
              295: *
E1DE DE 9D    296: CR       LDX   BUFPNT
E1E0 8C 0090  297:          CPX   #BUFFER
E1E3 27 5B    298:          BEQ   SETUP        Ie.nothing has been sent
E1E5 CE 0080  299: FINI     LDX   #DATA1       Transfer into depth register
E1E8 A6 10    300: FN1      LDA A 16,X
E1EA 84 0F    301:          AND A #$F
E1EC A7 00    302:          STA A 0,X
E1EE 08       303:          INX
E1EF 8C 0088  304:          CPX   #DATA1+8
E1F2 26 F4    305:          BNE   FN1
E1F4 7D 009C  306:          TST   DPENFG
E1F7 27 26    307:          BEQ   FN2
E1F9 86 44    308:          LDA A #'D
E1FB BD E686  309:          JSR   OUTCH
E1FE CE 0088  310:          LDX   #DATA2
E201 C6 0F    311:          LDA B #$F
E203 A6 00    312: LP1      LDA A 0,X
E205 E7 00    313:          STA B 0,X
E207 8B 30    314:          ADD A #$30
E209 BD E686  315:          JSR   OUTCH
E20C 08       316:          INX
E20D 8C 008C  317:          CPX   #DATA2+4
E210 26 F1    318:          BNE   LP1
E212 E7 00    319: LP3      STA B 0,X
E214 08       320:          INX
E215 8C 0090  321:          CPX   #DATA2+8
E218 26 F8    322:          BNE   LP3
E21A 7F 009C  323:          CLR   DPENFG
E21D 20 21    324:          BRA   SETUP
E21F 7D 009B  325: FN2      TST   TMENFG
E222 27 1C    326:          BEQ   SETUP
E224 86 54    327:          LDA A #'T
E226 BD E686  328:          JSR   OUTCH
E229 CE 0088  329:          LDX   #DATA2
E22C C6 0F    330:          LDA B #$F
E22E A6 00    331: LP2      LDA A 0,X
E230 E7 00    332:          STA B 0,X
E232 8B 30    333:          ADD A #$30
E234 BD E686  334:          JSR   OUTCH
E237 08       335:          INX
E238 8C 0090  336:          CPX   #DATA2+8
E23B 26 F1    337:          BNE   LP2
E23D 7F 009B  338:          CLR   TMENFG
E240 86 0D    339: SETUP    LDA A #$D
E242 BD E686  340:          JSR   OUTCH
E245 CE 0090  341:          LDX   #BUFFER
E248 DF 9D    342:          STX   BUFPNT
E24A 3B       343:          RTI
```

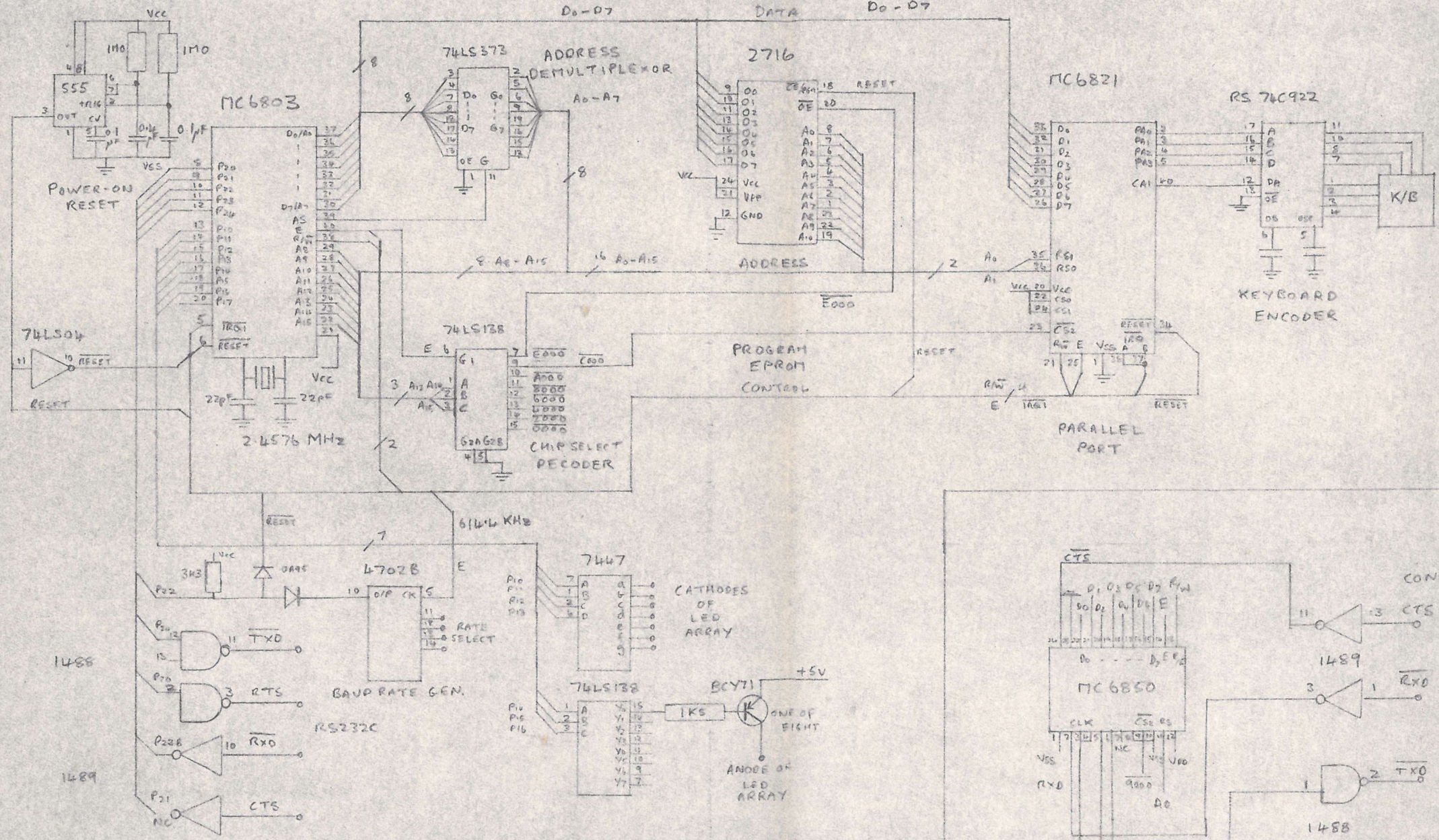
```
                344:          END
    NO ERROR(S) DETECTED


   SYMBOL TABLE:
BUFFER  0090    BUFPNT  009D    CL1     E025    CLEAR   E020
CLR1    E1B7    CLRENT  E1B2    CM1     E06D    CM2     E073
CM3     E079    CM4     E07F    COMMND  E066    CR      E1DE
DATA1   0080    DATA2   0088    DECODE  E0B6    DISP    E099
DPENFG  009C    DPF1    E107    DPF2    E10F    DPF3    E115
DPF4    E125    DPF5    E129    DPF6    E12E    DPF7    E11F
DPFMAT  E104    DSP1    E09B    DSP2    E0A2    DSPENT  E0AD
ENT1    E040    ENT2    E04F    ENT3    E051    ENT4    E043
ENTDP   E188    ENTTM   E197    ER      E0EC    ER1     E0F6
ERROR   E0D4    FINI    E1E5    FN1     E1E8    FN2     E21F
INIT    E003    INTSRV  E1C4    IOV     00FE    KEYBD   C000
LEDS    0000    LP1     E203    LP2     E22E    LP3     E212
NRMDSP  E085    OUTCH   E686    OUTHR   E67F    PORT2   0001
PORT3   0010    RET1    E091    RETURN  E1DD    SENDDP  E1A6
SENDTM  E1AC    SETUP   E240    SP      00FC    SPC     E6F0
START   E039    TABLE   E0C4    TMENFG  009B    TMF0    E14C
TMF1    E168    TMF2    E174    TMF3    E183    TMFMAT  E13B
XTEMP   0099
```

CIRCUIT DIAGRAMS



MUFAX MODULATOR



REMOTE KEYBOARD FOR P.E.S.T.
+ ADDITION TO PEST CPU BOARD

INSTITUTE OF OCEANOGRAPHIC SCIENCES.

ISSUED TO	DRAWING No. I.S.O./
REVISED	DRAWN BY C.C.FLEWELLEN DATE 8/12/82 SHEET No.
	ISSUED FOR ORDER No. FOR UNITS

Woking Dyeline

BLANK SIZE A.3.(297mm x 420mm)

0·157" PITCH 43 WAY DOUBLE-SIDED

8T97 8T97 8T97 8T26 8T26

DATA
ADDRESS
CONTROL
BUSEN

74LS04

DM8130 ADDRESS

DM8160 MULTIPLIER ADDRESS DECODER

CDP1855 MULTIPLIER/DIVIDER

74LS74 POWER-ON AND MANUAL RESET

MEMORY REFRESH CONTROL

MC 6800 CPU

CPU CONTROL

PROGRAM 2532

MONITOR 2716

MC6810 128 BYTES RAM

MC6850 SERIAL INTERFACE

MC 6871A CPU CLOCK

8T96 CONTROL LINE BUFFERS

74LS138 CHIP SELECTS

74LS30 EXTERNAL DATA BUS CONTROL

MC6871A 614·4KH$_z$ BAUD RATE CLOCK

RATE SELECT

RS232C BUFFERS

0·1" PITCH 32 WAY DOUBLE SIDED

PEST. PRECISION ECHO-SOUNDER TRACKER

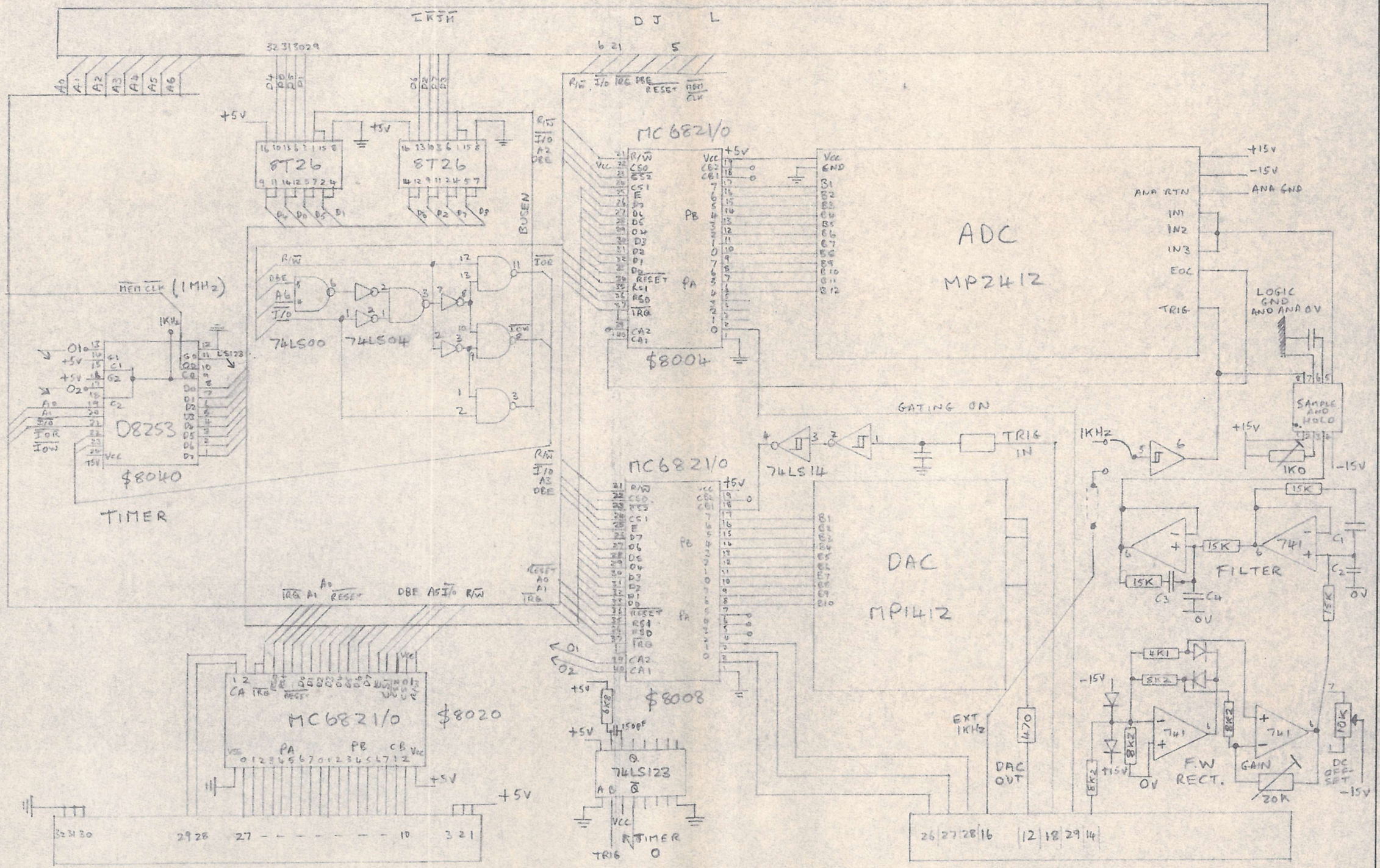
CPU BOARD

INSTITUTE OF OCEANOGRAPHIC SCIENCES.

ISSUED TO	DRAWING No. I.S.O./	
REVISED	DRAWN BY E Llewellyn DATE 27/9/82	SHEET No.
	ISSUED FOR ORDER No.	FOR UNITS

Woking Dyeline

A3 PS2935



PEST PRECISION ECHO-SOUNDER TRACKER
I/O BOARD

INSTITUTE OF OCEANOGRAPHIC SCIENCES.

ISSUED TO	DRAWING No. I.S.O./
REVISED	DRAWN BY C. O'Rawellon DATE 27/9/82 SHEET No.
	ISSUED FOR ORDER No. FOR UNITS

A3 PS2935

Woking Dyeline