SALESMEN'S SEMINAR

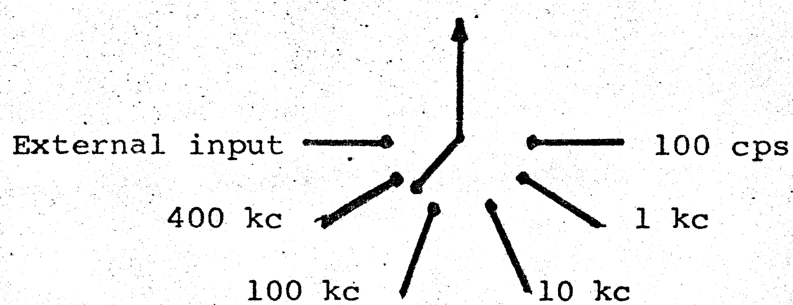
KW12 REAL TIME CLOCK

Introduction

The KW12 Clock is an exceedingly flexible tool. The following discussion is designed to present a functional "big-picture" description of the Clock and its features.

Pulse Generator - Counting Rate

KW12 has a Variable Time Base. The counting rate is selected by program control.



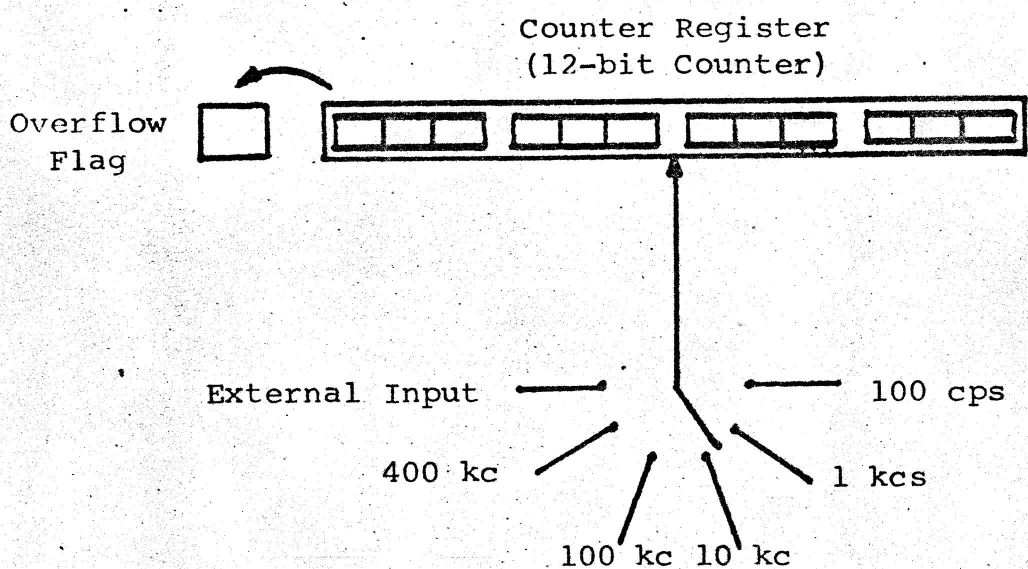
The external input line enables the user to utilize external events (such as line frequency) as a time base or to simply count external events.

The crystal-controlled pulse generator has an accuracy of .01% at 400 kc.

Counter Register

Each generated pulse causes the counter to be incremented by one.

The counter is a 12-bit register and counts up to 7777_8 and then "overflows" on the next pulse. The overflow flag can be detected by an interrupt, skip instruction, and/or read status instruction.



<u>Note:</u> Each time the counter reaches $7777_8 + 1$, the program-detectable overflow flag is set.

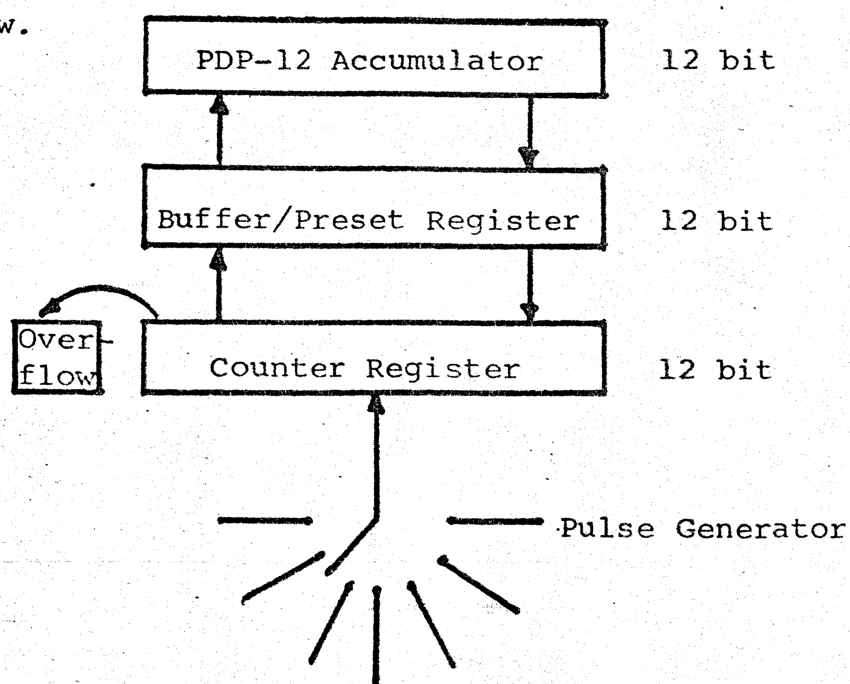
<u>Buffer/Preset Register</u>

The buffer/preset register ties the Clock to the PDP-12 accumulator. It has two functions:

1. To read the current contents of the counter into the AC (to see what "time" it is).

2. To set the counter to a desired starting value and allow it to count up from that number to overflow.

This means that the user may select not only the pulse generation or counting rate, but may also determine how many counts will cause a detectable overflow.



The user program can read the counter by loading its contents into the AC through the buffer/preset register.

The user program can initialize the counter to a starting quantity by loading the counter from the AC through the buffer/preset register.

It can now be seen that under program control:

1. Counting rate is selected.

2. Counter overflow is detected.

3. Contents of counter can be transferred to AC.

4. Counter can be reset to a known value from AC.

The question now is how does the program know when it should read and/or reset the clock counter. This depends upon the status of the real-world events with which the program is dealing. External events (pulses) are detected on three input channels.

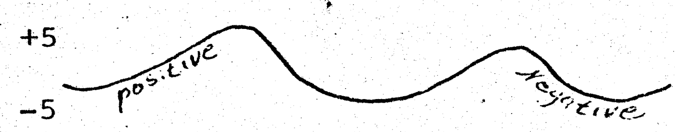
Input Synchronizers

Each of the three input channels incorporates a Schmitt trigger with pulse generator and associated logic.

The clock control panel allows selection of

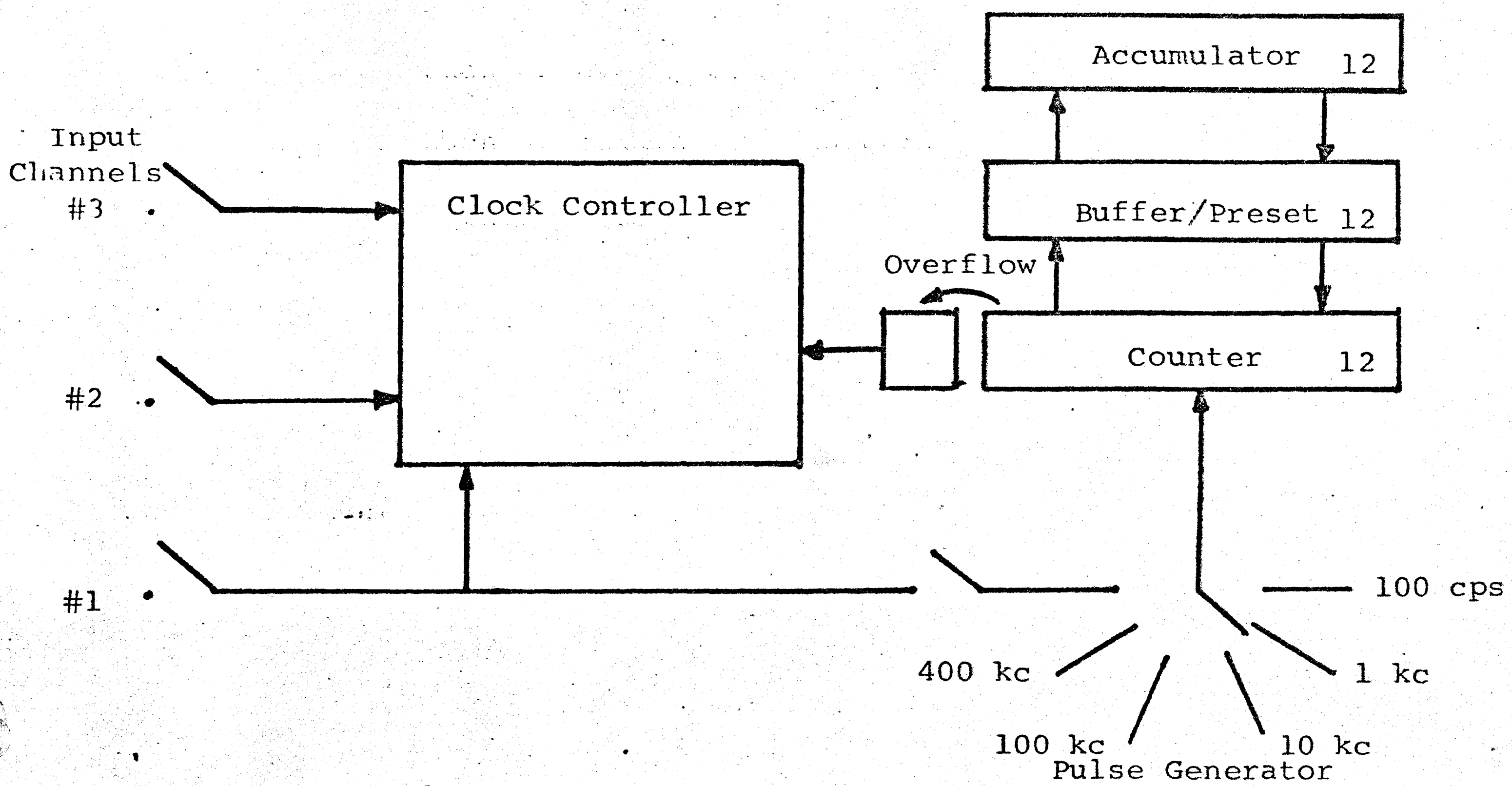
1) Input signal voltage threshold (±5 volt range).

2) Either positive- or negative-going slope.



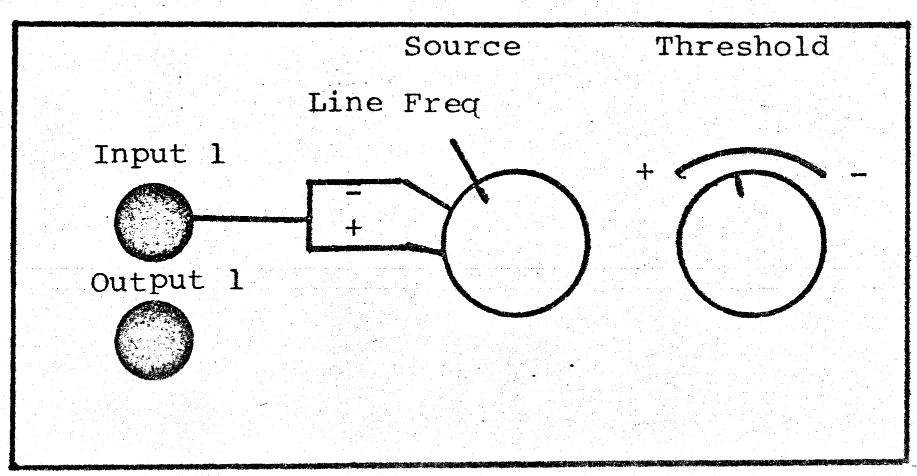
When these preselected conditions are met, the input signal is converted to a program-detectable event.

When such an event occurs, the program can take appropriate action, such as reading or resetting the counter, incrementing an event tally, etc.



Note that external input channels are selected under program control. The voltage threshold and signal slope are selected via the control panel.

Input Control Panel

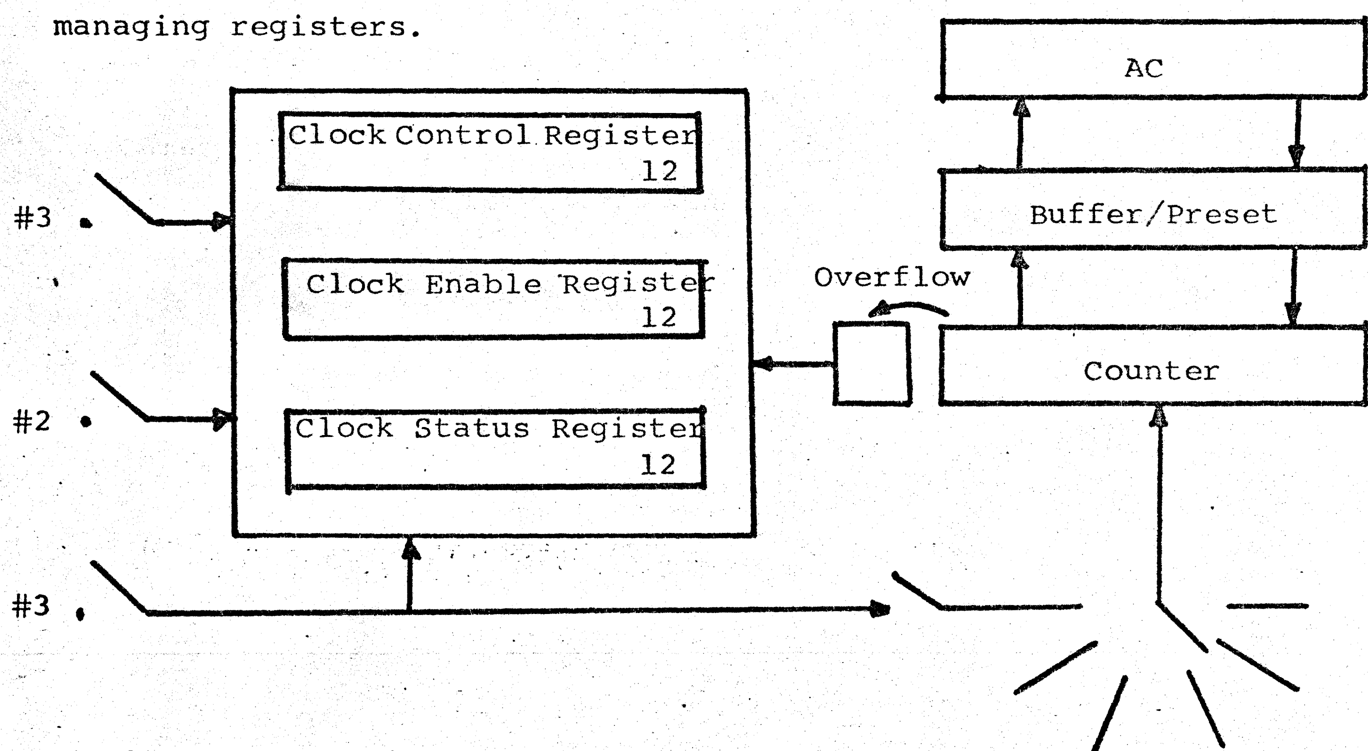


Jacks - Input 1 and Output 1 are phone jacks. The output is available for daisy-chaining to another input, A/D converter, etc.

Source - Input signal slope selection. Also line voltage selection for 60-cycle/50-cycle timing or counting.

Threshold - Coarse ± 5 volt selection.

Program control of the clock includes the use of three managing registers.



Clock Control Register

This register selects the counting rate and mode of clock operation. It is initialized through 8-mode IOT instruction 6132 - CLLR (Load clock control register from AC).

AC	Ø 1 2	3 4 5	6 7 8 9 10 11
	Counting Rate Select	Mode Select	Input Simulation (For Diagnostic Programming)

Counting Rate

$AC_{Ø-2}$	
ØØØ	STOP
ØØ1	400 kc
Ø1Ø	100 kc
Ø11	10 kc
1ØØ	1 kc
1Ø1	100 cps
11Ø	Count from channel 1
111	STOP

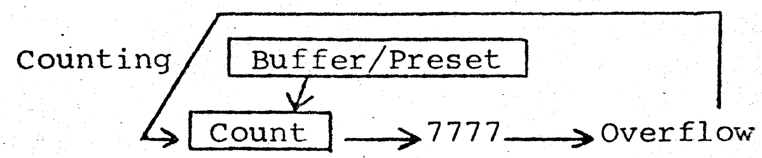
Modes of Clock Operation

AC_{3-5}

ØØØ – Free Running Mode - Counter runs at selected rate, and overflow occurs every 4096_{10} counts. This mode is probably not too useful except for producing overflow "ticks" every (counting rate x 4096) second. It is easy to understand

Counting → ØØØØ → 7777 → Overflow

ØØ1 - Free Running Mode with Preset - Counter runs a selected rate, and when overflow occurs, the contents of the buffer/preset register are transferred to the counter and counting continues.



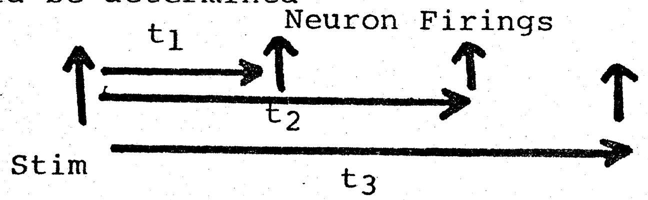
This mode allows for an overflow event to occur at predetermined time intervals.

The buffer/preset register is usually initialized to the negative value (two's compliment) of the number of counts desired before overflow.

Note that the overflow flag set in the above two modes remains set until cleared by IOT instruction 6135.

Ø1Ø - Event Timing Mode with no resetting. Counter runs at selected rate, and on the occurence of an event on a selected input channel, the contents of the counter are transferred to the buffer/preset register and the counter continues to count. This mode is useful for determining the total elapsed time from one initial event to subsequent events.

A poststimulus histogram (PST) could be generated using this mode. A stimulus could be issued to a subject and the time to succeeding neuron firings could be determined



Ø11 - Event Timing with Resetting to Zero - Counter runs at selected rate. Events on channels 1 and 2 can cause the transfer of the counter to the buffer preset register and the counter will continue to count. An event on channel 3, however, will cause the counter to be cleared after its contens have been transferred to the buffer/preset register.

This mode is useful in "restarting" the counter on one external event, but not on the others. For the above example, another stimulus on channel 3 could cause clearing of counter for timing following neuron firing with respect to the second stimulus. It can also be used to record the time between events.

Clock Initiated A/D Conversion

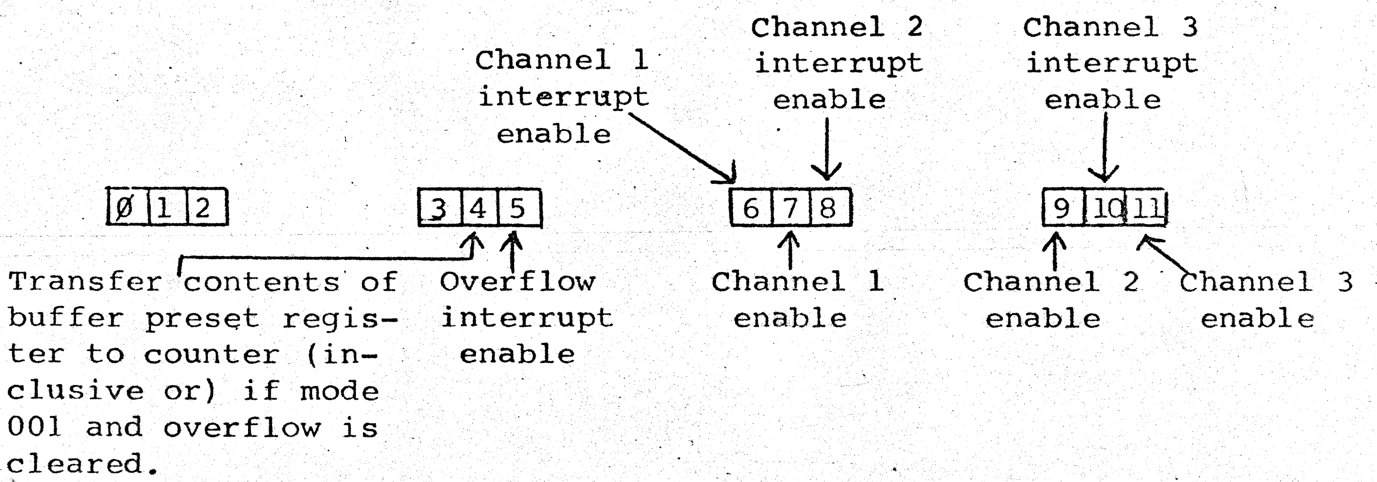
AC_{3-5}

100, 101, 110, 111 are identical to 000, 001, 010, 011 except that the occurrence of overflow will trigger the A/D converter if fast-sample function is set.

Clock Enable Register

The "who will do it" register

1. Selects the input channels which are to be active during the program run.
2. Controls interrupt enabling for external events and overflow flag.
3. Allows initializing of counter from buffer/preset register.

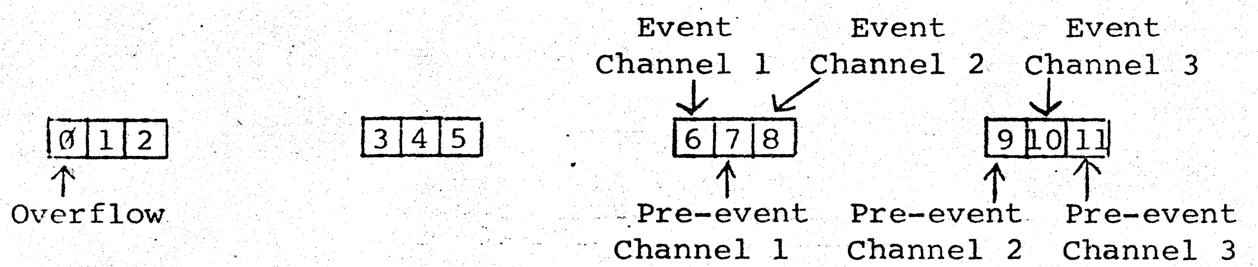


This register is loaded from AC by IOT 6134 CLEN (Load Clock Enable register).

Clock Status Register

This is the "who done it" register. When an external event or overflow occurs, a bit is set in this register and the entire register is available for program interrogation.

If more than one event occurs before this register is read, another bit is set indicating an event overlap condition. If the overlap event is on the same channel as the first event, a "pre-event" bit is set.



The status register is transferred to the AC with IOT 6135 CLSA (Clock Status to AC) inclusive or reading the status register clears all event indicators including the overflow flag.

Clock Handling Instructions (8 MODE)

6137 CLCA Clock Counter to AC

AC cleared first, then the counter contents is transferred to the buffer/preset register then to AC.

6136 CLBA Buffer Preset to AC

AC cleared first.

6133 CLAB AC to Buffer Preset

AC is unchanged. Previous contents of buffer preset are unchanged.

6131 CLSK Skip on Clock Interrupt

KW12 REAL TIME CLOCK

SUBROUTINE SETUP PACKAGE

(The subroutine described below is contained on the Demo Tape. It is listed in the DIAL Index as KW12SUB.)

Abstract

This package of subroutines contains the necessary fixed protocol required to start the KW12. The main program need only specify the working settings of the clock control register, the clock enable register, and the clock counter. It is emphasized that this subroutine package is designed to initialize the clock. Clock interrupts must be handled using the CLSA (6135) instruction.

Requirements

Storage

This package of subroutines uses 137 (octal) or 95 (decimal) locations.

Location

The first location of this subroutine package should be assembled into the first location of any 8-mode page with the rest of the program following in successive locations. The subroutine is preset to start in location $(6000)_8$ of bank Ø unless changed by the user.

Usage

Calling Sequence

This subroutine package has five 8-mode entry points that are indirectly called as follows:

```
JMS I .+1      / Indirect jump to subroutine
KWxxxx         / whose location is contained
               / in the location immediately
               / following the JMS
```

In using one of the entry points, the subroutine will look for data contained in the third, fourth, and fifth locations following the JMS. In all cases except where the clock counter is to be read, the AC and link of the main program are preserved.

The entry point KWMEASure is used to set up the clock to measure events or intervals of time. The working contents of the clock control register, clock enable register, and the number of counts to overflow are specified in the locations following the JMS as follows:

```
.     JMS I .+1
.+1   KWMEAS        / JMS to entry point KWMEAS
.+2   C1C2C3C4      / clock control register setting

.+3   BUFF          / No. of counts to overflow
.+4   E1E2E3E4      / clock enable register setting

.+5                 / control returns here
```

Other entry points are used as follows:

```
.       JMS I .+1      / Stop the clock but preserve
.+1     KWSTOP         / software control, enable, and
                       / buffer registers
.+2                    / Control returns here with AC and
                       / link preserved


.       JMS I .+1      / Restart the clock using
.+1     KWCONTinue     / the settings contained in the
                       / software registers
.+2                    / Control returns here with the AC
                       / and link preserved


.       JMS I .+1      / Read the clock counter into the AC
.+1     KWREAD         / If overflow has occurred, set
                       / link = 1, otherwise link = Ø
.+2                    / Control returns here

                       / In the process of performing the
                       / read, the buffer preset register
                       / is restored to the contents of
                       / KWBUFF.
```

Examples

The following is a program to ring the Teletype bell 1 time per second.

```
Begin,  JMS I .+1      / JMS indirectly to
        KWMEAS         / loc KWMEAS
        51ØØ           / rate = 1ØØ Hz Mode = 1
        Ø144           / 144 octal or 100 decimal counts
                       / to overflow
        Ø1ØØ           / interrupt on clock overflow
Check,  CLSK           / (6131) skip on clock interrupt
        JMP check      / Wait
        CLSA           / Clear Clock Flag (6135)
        CLA            / Clear AC
```

```
        TAD K207        / Load 207 into AC
        TSF             / Check Teletype flag
        JMP .-1
        TLS             / Output AC to Teletype
        JMP check       / Go back and wait for next
                        / clock overflow
        K207, 0207      / Symbol definition

                        / append KW12 set up subroutine
                        / package
```

The following is a program that will ring the Teletype bell after counting 60 events on external channel 1.

```
Begin,  JMS I .+1       / JMS indirectly to
        KWMEAS          / location KWMEAS
        61ØØ            / Rate = chan 1   Mode = 1
        ØØ74            / Interrupt after (74)8 or (60)10
                        / counts

        Ø12Ø            / Enable chan 1 input
                        / interrupt on overflow
Check,  CLSK            / (6131) skip on clock interrupt
        JMPCHCK         / Wait
        CLSA            / Clear clock flag (6135)
        CLA             / Clear AC
        TAD K207        / Load AC with (207)8
        TSF             / Test Teletype flag
        JMP .-1         / Wait
        TLS             / Ring bell
        JMP CHECK       / Jump to wait loop
        K207, 0207      / Symbol definition
                        / append KW12 setup subroutine
                        / package
```

bam
11/25/69

6133 CLAB AC to Buffer Preset

AC is unchanged. Previous contents of buffer preset are unchanged.

6131 CLSK Skip on Clock Interrupt

KW12 REAL TIME CLOCK

SUBROUTINE SETUP PACKAGE

(The subroutine described below is contained on the Demo Tape. It is listed in the DIAL Index as KW12SUB.)

Abstract

This package of subroutines contains the necessary fixed protocol required to start the KW12. The main program need only specify the working settings of the clock control register, the clock enable register, and the clock counter. It is emphasized that this subroutine package is designed to initialize the clock. Clock interrupts must be handled using the CLSA (6135) instruction.

Requirements

Storage

This package of subroutines uses 137 (octal) or 95 (decimal) locations.

Location

The first location of this subroutine package should be assembled into the first location of any 8-mode page with the rest of the program following in successive locations. The subroutine is preset to start in location $(6000)_8$ of bank Ø unless changed by the user.

Usage

Calling Sequence

This subroutine package has five 8-mode entry points that are indirectly called as follows:

```
JMS I .+1       / Indirect jump to subroutine
KWxxxx          / whose location is contained

                / in the location immediately

                / following the JMS
```

In using one of the entry points, the subroutine will look for data contained in the third, fourth, and fifth locations following the JMS. In all cases except where the clock counter is to be read, the AC and link of the main program are preserved.

The entry point KWMEASure is used to set up the clock to measure events or intervals of time. The working contents of the clock control register, clock enable register, and the number of counts to overflow are specified in the locations following the JMS as follows:

```
.      JMS I .+1
.+1    KWMEAS         / JMS to entry point KWMEAS
.+2    C1C2C3C4       / clock control register setting

.+3    BUFF           / No. of counts to overflow
.+4    E1E2E3E4       / clock enable register setting

.+5                   / control returns here
```

Other entry points are used as follows:

```
.         JMS I .+1       / Stop the clock but preserve
.+1       KWSTOP          / software control, enable, and
                          / buffer registers
.+2                       / Control returns here with AC and
                          / link preserved

.         JMS I .+1       / Restart the clock using
.+1       KWCONTinue      / the settings contained in the
                          / software registers
.+2                       / Control returns here with the AC
                          / and link preserved

.         JMS I .+1       / Read the clock counter into the AC
.+1       KWREAD          / If overflow has occurred, set
                          / link = 1, otherwise link = Ø
.+2                       / Control returns here

                          / In the process of performing the
                          / read, the buffer preset register
                          / is restored to the contents of
                          / KWBUFF.
```

Examples

The following is a program to ring the Teletype bell 1 time

per second.

```
Begin,    JMS I .+1       / JMS indirectly to
          KWMEAS          / loc KWMEAS
          51ØØ            / rate = 1ØØ Hz Mode = 1
          Ø144            / 144 octal or 100 decimal counts
                          / to overflow
          Ø1ØØ            / interrupt on clock overflow
Check,    CLSK            / (6131) skip on clock interrupt
          JMP check       / Wait
          CLSA            / Clear Clock Flag (6135)
          CLA             / Clear AC
```

KW12, REAL TIME CLOCK, SUBROUTINE SETUP PACKAGE

1.0 ABSTRACT

This package of subroutines contains the necessary fixed protocol required to start the KW12. The main program need only specify the working settings of the clock control register, the clock enable register, and the buffer register. It is emphasized that this subroutine package is designed to initialize the clock. Clock interrupts must be handled using the CLSA (6135) instruction.

2.0 REQUIREMENTS

2.1 STORAGE

137 octal or 95 decimal locations are used by the KW12 subroutine package.

2.2 Location

The first location of this subroutine package should be assembled into the first location of any 8-mode page with the rest of the program following in successive locations. The subroutine is preset to start in location 6000_8 of bank Ø unless changed by the user.

2.3 Equipment

Basic PDP-12A or B with KW12 option.

3.0 USAGE

3.1 Calling Sequence

This subroutine has five 8-mode entry points that are indirectly called, as follows:

```
        JMS  I .+1          /INDIRECT JUMP TO SUBROUTINE
                            /WHOSE LOCATION IS CONTAINED
        KWxxxx              /IN THE LOCATION IMMEDIATELY
                            /FOLLOWING THE JMS
```

In using one of the entry points, the subroutine will look for data contained in the third, fourth, and fifth locations following the JMS. In all cases, except where the clock counter is to be read, the AC and link of the main program are preserved.

4.0 DESCRIPTION

4.1 Discussion

The entry point KWMEASure is used to set up the clock to measure events or intervals of time. The working contents of the clock control register, clock enable register, and the number of counts to overflow are specified in the locations following the JMS as follows:

.	JMS I .+1	
.+1	KWMEAS	/JMS TO ENTRY POINT KWMEAS
.+2	$C_1C_2C_3C_4$	/CLOCK CONTROL REGISTER SETTING
.+3	BUFF	/NO. OF COUNTS TO OVERFLOW
.+4	$E_1E_2E_3E_4$	/CLOCK ENABLE REGISTER SETTING
.+5		/CONTROL RETURNS HERE

The first two octal digits of the control register setting control the rate and mode of the clock respectively. The last two digits are used to simulate inputs from the external channels. The interpretation of C_1 is as follows:

C_1	Rate
Ø	STOP
1	400 KHz
2	100 KHz
3	10 KHz
4	1 KHz
5	100 Hz
6	Count occurrences of events on external channel 1 if E_3 2.

The interpretation of C_2, the mode control, is as follows:

C_2	Mode
Ø	Counter runs at selected rate and overflow is set after 4096 counts. The counter continues to run and the overflow must be cleared by the IOT 6135.
1	Counter runs at selected rate. Upon occurrence of overflow caused by the most significant bit of the counter going from 1 to Ø, the contents of the buffer preset register is transferred to the counter.
2	Counter runs at selected rate. Upon occurrence of an event on an enabled channel, the contents of the counter are transferred to the buffer preset register and the counter continues to run.
3	Counter runs at selected rate. Upon occurrence of an input event on channel 3, the contents of the counter is transferred to the buffer preset register and the counter is reset to Ø from which it continues counting. Inputs on channels 1 and 2 behave as if $C_2 = 2$.

4 The counter runs at the selected rate and overflows every 4096 counts. Upon the occurrence of an overflow, an A/D conversion is initiated if the fast sample mode is in use. Overflow must be cleared by the 6135 instruction. Note that when in this mode A/D conversions are inhibited unless a clock overflow occurs.

5 This is treated as the combination of $(C_2 = 4)$ and $(C_2 = 1)$

6 $(C_2 = 4)$ and $(C_2 = 2)$

7 $(C_2 = 4)$ and $(C_2 = 3)$

BUFF is an octal number from 1 - 7777_8 and is the number of desired counts to occur before the counter overflows. The subroutine places the 2's complement of BUFF in the buffer preset register and initially in the counter.

The last 3 digits of the enable register setting are interpreted as follows:

E_2	Function
0,2,4,6	No Function
1,3,5,7	Cause clock interrupt on counter overflow.

E_3	Function
Ø	No Function
1	Enable interrupt on channel 2 if channel 2 input enabled.
2	Enable input on channel 1
3	$(E_3 = 1)$ and $(E_3 = 2)$
4	No Function
5	Same as $E_3 = 1$
6	Enable input and interrupt on channel 1
7	$(E_3 = 6)$ and $(E_3 = 1)$

E_4	Function
Ø	No Function
1	No Function
2	Enable input channel 3
3	Enable input and interrupt for channel 3
4	Enable input channel 2
5	Same as $E_4 = 4$
6	$(E_4 = 4)$ and $(E_4 = 2)$
7	$(E_4 = 4)$ and $(E_4 = 3)$

Upon entering the KW12 setup subroutine package at point KWMEASure, the subroutine retrieves the locations . 2, . 3, and . 4, considering the location (.) to be the locations of the JMS I . 1. The contents of these locations are stored in software created control (KWCONR), enable (KWENAB), and buffer registers (KWBUFF). Then the clock is set up to perform the specified tasks. Control is returned to location . 5 with the AC and link preserved and the clock running.

Other entry points are used as follows:

```
.       JMS I .+1       /STOP THE CLOCK BUT PRESERVE
.+1     KWSTOP          /SOFTWARE CONTROL, ENABLE, AND
                        /BUFFER REGISTERS

.+2                     /CONTROL RETURNS HERE WITH AC
                        /AND LINK PRESERVED

.       JMS I .+1       /RESTART THE CLOCK USING
.+1     KWCONTine       /THE SETTINGS CONTAINED IN
                        /THE SOFTWARE REGISTERS
.+2                     /CONTROL RETURNS HERE WITH
                        /THE AC AND LINK PRESERVED

.       JMS I .+1       /READ THE CLOCK COUNTER INTO
.+1     KWREAD          /THE AC.  IF OVERFLOW HAS
                        /OCCURRED,SET LINK = 1,
                        /OTHERWISE LINK = Ø
.+2                     /CONTROL RETURNS HERE

                        /IN THE PROCESS OF PERFORMING
                        /THE READ, THE BUFFER PRESET
                        /REGISTER IS RESTORED TO THE
                        /CONTENTS OF KWBUFF.
```

```
.        JMS I .+1          /INITIALIZES THE CLOCK AND
.+1      KWINIT             /CLEARS SOFTWARE CONTROL,
                            /ENABLE, AND BUFFER REGISTERS.
.+2                         /CONTROL RETURNS TO MAIN
                            /PROGRAM HERE WITH AC AND
                            /LINK PRESERVED.
```

4.2 <u>EXAMPLES</u>

The following is a program to ring the Teletype bell 1 time per second:

```
BEGIN,  JMS   I .+1     /JMS INDIRECTLY TO
        KWMEAS          /LOC KWMEAS
        51ØØ            /RATE = 1ØØ Hz, MODE = 1
        Ø144            /144 OCTAL or 1ØØ DECIMAL COUNTS
                        /TO OVERFLOW.
        Ø1ØØ            /INTERRUPT ON CLOCK OVERFLOW

CHECK,  CLSK            /(6131) SKIP ON CLOCK INTERRUPT
        JMP CHECK       /WAIT
        CLSA            /CLEAR CLOCK FLAG (6135)
        CLA             /CLEAR AC
        TAD  K207       /LOAD 207 INTO AC
```

```
        TSF                     /CHECK TELETYPE FLAG
        JMP  .-1                /
        TLS                     /OUTPUT AC TO TELETYPE
        JMP  CHECK              /GO BACK AND WAIT FOR NEXT
                                /CLOCK OVERFLOW
        K207,   0207            /SYMBOL DEFINITION


                                /APPEND KW12 SETUP SUBROUTINE
                                /PACKAGE
```

The following is a program that will ring the Teletype bell after counting 60 events on external channel 1:

```
BEGIN,      JMS  I   .+1        /JMS INDIRECTLY TO
            KWMEAS              /LOCATION KWMEAS
            61ØØ                /RATE = CHAN 1,   MODE = 1
            ØØ74                /INTERRUPT AFTER (74)8 or
                                /(60)10 COUNTS
            Ø12Ø                /ENABLE CHAN 1 INPUT
                                /INTERRUPT ON OVERFLOW
```

```
CHECK, CLSK                  /(6131) SKIP ON CLOCK INTERRUPT
       JMP CHECK             /WAIT
       CLSA                  /CLEAR CLOCK FLAG (6135)
       CLA                   /CLEAR AC
       TAD K207              /LOAD AC WITH (207)8
       TSF                   /TEST TELETYPE FLAG
       JMP  .-1              /WAIT
       TLS                   /RING BELL
       JMP CHECK             /JUMP TO WAIT LOOP
       K207, 0207            /SYMBOL DEFINITION

                             /APPEND KW12 SETUP SUBROUTINE
                             /PACKAGE
```

.DFRQANA or FRQANA

Original Author: A. M. Engebretson

Modified to run under the PDP-12 DEMO Monitor by: Jerre Caputo,
DEC, Washington, D. C.

~~ABSTRACT~~

FRQANA ~~This program~~ performs frequency analysis resulting in sixty-four component cosine, sine, and rms spectra. Data and spectrum displays can be scaled; ~~resynthesis of~~ the original data ~~may~~ can be resynthesized ~~also be performed~~ from the sine and cosine components to check visually the accuracy of the analysis. (See Sec. 4. for a frequency domain description. A note on the use of FRQANA to notch filter data can be found following table 3 of this writeup.)

~~USAGE~~

When the program is first started, ~~At load time,~~ and each time R is typed, ~~the user has the opportunity to~~ new input data can be retrieved from LINCtape by ~~filling in the blanks on the display from the Teletype keyboard.~~ supplying values to the message on the display. The display ~~asks~~ is:

RETRIEVE DATA

(2 BLOCKS)

STARTING AT BLOCK _ _ _

UNIT _

~~The user~~ type~~s~~ the number of the first of two consecutive LINCtape blocks to be analyzed and press the RETURN key. Leading zeros must be typed. ~~and the block number must be followed by pressing the RETURN key.~~ The user then types the tape unit number to be read, 0 - 7, followed by pressing LINEFEED. The two blocks specified are read in and displayed. Th~~is~~e input data ~~may~~ can then be scaled ~~larger~~ by typing L to increase the vertical scale by 50 percent or by typing S to decrease the vertical scale by 50 percent, ~~repeatedly in each case.~~ L or S may be typed repeatedly, if desired.

~~2.~~ To perform the frequency analysis, ~~the user~~ types F. When the analysis is complete, the display will reappear according to which display option previously has been chosen. Display options are summarized in Table 1.

~~3.~~ To compute the resynthesized waveform from the frequency components, ~~the user~~ types T. Again, the display will reappear according to the display option that had been chosen. ~~When the~~ If the display option is I, ~~for~~ input data, the second trace ~~will be~~ is the resynthesized waveform. The input data can be compared with the resynthesized waveform by turning knob 1. Knob 1 is a trace separator, affecting only the vertical display of the data. When the knob is turned fully counterclockwise, the top trace is the original data.

~~4.~~ At any time during operation, a different display may be selected by typing the appropriate key. To select the input data and resynthesized waveform display, the user types I. This is the display that appears immediately after program start or restart.

The command causes ~~When the user types~~ C, the frequency component display to appear~~s~~ with the cosine components on the left and the sine components on the right, ~~will appear.~~ These can be scaled using knob 2.

To select an rms spectrum for display, type P. This display ~~is~~ can also be scal~~able~~ed by Knob 2.

Each bar in the graph of the components and the rms spectr~~um~~a represents a single frequency component from 1 to 64, i.e., the first bar is the component for the 1 cycle signal generator and the third bar is the component for the 3 cycle signal generator, etc., up through 64. The absolute frequency of the analyzed waveform components can only be obtained by taking the time period represented by the 512 data points sampled; this will vary depending on the sampling rate used to collect the data. If the sampling rate were 512 pts./sec. when the data was taken, then each bar of the graph ~~would~~ will represent the absolute components of the frequency domains 1 through 64. If the rate were 1024 pts./sec., then the bars ~~would~~ represents frequencies 2,4...128 cps.

~~5.~~ If, at any time, the user wishes to select new input data for analysis, he simply types R to ~~return to step 1.~~ restart the program.

FRQANA Commands

Table 1

KNOB COMMANDS

Knob 1 - Trace separator (original data and resynthesized data)

Knob 2 - Modify spectrum scale

TELETYPE COMMANDS

F - Generate Fourier Coefficients

T - Resynthesize F (t)

S - Half ~~Make~~ data ~~smaller by half~~

L - Enlarge ~~Make~~ data larger by half

P - Display power spectrum

C - Display sine and cosine components

I - Display input data (and resynthesized data if T has been typed)

R - Restore and Retrieve data from tape

PROGRAM OPERATION

FRQANA under the DEMO Monitor relies heavily on overlay capabilities and uses DEMO Monitor commands RETV (OPR 5) to obtain the starting block number of FRQANA and OKST (OPR 12) and KBD (OPR 15) to handle keyboard commands.

When FRQANA is loaded under the DEMO Monitor, an attempt is made to load blocks containing cosine tables into the second 4K of core. If there is no second 4K, the load is not completed; this will not affect program operation. The program starts at LINC location 20 and control is immediately transferred to code in LINC segment 2 to set up data retrieval procedures for later overlays.

Control is then transferred to QANDA, also in LINC segment 2, to query the user concerning data blocks to be analyzed. This code is assembled in the same segment as DEMO Summary and Help Text, and the two tape blocks containing this data (blocks 4 and 5, Table 4) are transferred directly to the help frame blocks used by the DEMO Monitor and stored as source files.

Immediately after Line Feed is pressed in the QANDA routine, LINC segments 2 and 3 of the program are overlaid with input data and cosine tables. The program then begins operation, as described in the USAGE section.

Whenever R is typed, LINC segments 2 and 3 are overlaid with QANDA and text to query the user. The commands to do this are the same as those used during program initialization.

Memory allocations at query time and at run time are described in Tables 2 and 3. To avoid confusion and illustrate the complete contents of core, both LINC segments and PDP-8 absolute addresses are given. Tape block allocation is illustrated in Table 4. The block numbers listed are relative to the starting block number on tape.

WARNING

The name of the program, .DFRQANA, is specified in the retrieve file set up in relative block number 5 at location 334 of FRQANA.

If the name under the DEMO Monitor is changed, then this tape block must be changed also.

isn't this for all of 'em?

TABLE 2

MEMORY ALLOCATION AT QUERY TIME

LINC Segment	MemBlks	PDP-8 Location	Contents
0-1	0-7	0-3777	DEMO Monitor
2	0-1½	4000-4577	Program
2	1-3	4600-5777	Unused core
2	1	6000-6377	Program summary text, help text
2	2	6400-6777	Help text, retrieve data text, code to handle subsequent overlays
3	3-4	7000-7777	QANDA and code to display text, decode answers

LINC Segment	MemBlks	PDP-8 Location	Contents
0-1	0-7	0-3777	DEMO Monitor
2	0-1½	4000-4577	Program
2	1	4600-4677[1]	Sine spectrum
2	1	4700-4777[1]	Cosine spectrum
2	2-3	5000-5777	Synthesized time function
3	0-1	6000-6777	Input data
3	2-3	7000-7777	Cosine table

[1] Note that each section is 64 words long, i.e., each word in the sine table contains the value of the sine component associated with a frequency component and likewise for the cosine component table. After F has been typed, these tables are filled with sine and cosine data representing the phase angle and amplitude of the 64 signal generators. These 64 generators, when run out mathematically as in the FOURIER series computation, will regenerate the original data almost as accurately as when it was taken. The way to produce a notch filter effect then would be to attenuate to zero whichever of the sine-cosine components representing the frequency generator are to be removed from the signal, e.g. to remove 60 cycle from the signal zero the 60th sine and cosine component of core from the console using the switch registers and deposit key. The above example assumes 512 pt/sec. sample rate at data taking time.

Relative Block No.	Contents	Residence in Core
1	Standard DIAL header block	Never gets to core
2	FRQANA	4000-4377
3	FRQANA	4400-4777
4	Summary-Help 1	6000-6377
5	Help 2 - Retrieve	6400-6777
6	QANDA	7000-7377
7	QANDA	7400-7777
10	Cosine Table	10000-10377
11	Cosine Table	10400-10777