MULTIPURPOSE MICROPROCESSOR BASED FREQUENCY COUNTER

* *GEORGE THANGADURAI,* ** *S BIRLASEKARAN and* * *PM RAMANATHAN*

* Alagappa Chettiar College of Engg & Tech, Karaikudi - 623 006 ** Central Electrochemical Research Institute, Karaikudi - 623 006

For real time computer based studies in coulometry, mercury drop time measurement, thickness measurement in plating, speed and frequency measurement in electrochemical process control equipments, the requirement of a microprocessor based industrial counter is felt. With little number of external hardware, a microprocessor based frequency counter is designed. Software controlled facilities are: (a) Debounce time facility (b) frequency range upto a maximum of 100 kHz (c) six digit display (d) selection of time interval (e) storage and sequence controlled measurement. The details on developed software on MPF-1 - microprocessor kit are presented.

Key words: Frequency counter, microprocessor, Z-80

INTRODUCTION

The basic function of a frequency counter is to count pulses during a precisely defined time interval. In industrial environment, the counting signal may contain some distortion in the earlier part of the wave due to poor switching, or due to the poor response of transducer etc. That described portion is called normally, contact rebounce. By delaying the proper time before testing the signal again, this bounce may be disregarded in accumulating counts. Software has been prepared for a variable debounce delay time which is normally about one millisecond. Another important feature of this multipurpose frequency counter is that it can be used for specialised applications involving different time base intervals e.g: 1 sec, 10 secs and 1 min etc. viz: in slow stirring rate used in electrochemical processes. Also here we test for one level state but also zero level state before incrementing the counter. It is a portable unit and it can be operated with battery. The unit may cost around Rs.3000/.. with other peripherals.

CIRCUIT LAYOUT

The complete hardware and monitor routines are available in Reference 1. For simplification, block diagram representation is given in Fig: 1

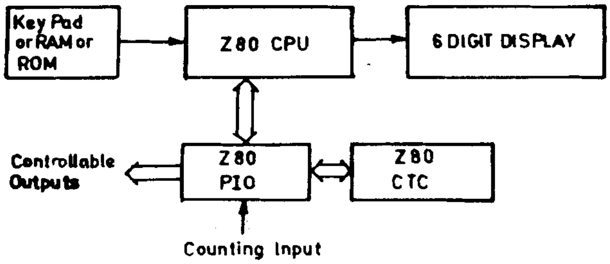

Fig.1: Block diagram layout of uP based frequency counter

The complete architecture is centered around Z80 CPU with Z80 BTC as a programmable clock timer unit having four independent channels for counting and timing applications and with Z80 PIO as a programmable input/output interface having 14 I/O lines [1]. Using keypad, the required parameters like time base interval, debounce time interval are keyed in. On executing depending on the software selection, counting sequence is started.

OPERATION OF THE SYSTEM

It involves with clock and counting operation.

Counting operation

Counting operation requires the feeding of the signal whose frequency is to be measured. Here the Z80 PIO is used to feed the pulses into the microprocessor. Using the PIO control word we select any one of the eight bits either in Port A or Port B as the input gate. For example, for this project we have chosen the bit A7 (7) of the PIO as the input gate.

The actual counting procedure consists in asking the microprocessor to wait till the pulse goes to the 'One' state. As a loop sequence the microprocessor kit keeps on testing for the 'One' state. As soon as the pulse returns to the 'Zero' state the microprocessor after making sure it has gone out of the 'One' state, then goes on to test for the 'Zero' state. Again as mentioned above the microprocessor keeps on testing for the 'Zero' state till the pulse again goes high.

As soon as the microprocessor senses that the pulse has gone to the high state, it increments the count by 'One' and jumps on to the initial location where it again tests for the 'One' state and the whole process is gone through again and again. Since this process takes only about 40 instructions cycle for system clockwise rate of 4 MHz, this counting sequence can count nearly upto 100 kHz without any appreciable error. This demands a minimum pulse width of 5 microseconds [2].

The whole operation is done for a single period. Here the HL register is used as a counter wherein for every pulse occurring, it increments by one. After certain time base interval which is fixed by the clock the total count in the HL register is transferred to some memory locations. In this case it is stored in 1AOOH and 1AO1H.

Clock operation

Fixing the time base is important in any counter operation. Though usually the time base is one second for all counting operations, for some specialised applications like tachometer, interruption counter, pulse monitor and chemical process control where either due to the low number of counts or due to some specialised reason (tachometer usually reads in rpm) it demands a different timebase, we should have the option of choosing different time bases at free will with very slight variations in software.

Z80 counter timer is a programmable component with four independent channels that provide counting and timing functions. Dach channel has a down counter which when loaded from initial value in the Time constant register decrements the initial value until it reaches zero. When the zero count occurs, the channel sends a pulse thro' the ZC/TO (O count/time out) pin of that particular channel. This is how the channel operates in the timer mode. To operate the channel in the counter mode requires the clock trigger for every decrement in the Down counter. Using these principles, one, two or more channels can be combined whereby the first channel is asked to operate in the Timer mode and the others are made to operate in the counter mode.

On reaching the zero count the first channel sends out a zero count pulse which may be used to trigger the next channel for decrement operation. By using different initial values in the time constant register of each channel we are thus able to get different time base intervals. When the final channel decrements the zero it is made to generate an interrupt pulse.

INTERRUPT OPERATION

The Z80 CPU can suspend the current program execution by external interrupt request. The CPU then starts executing the interrupt service routine. Once the service routine is completed, the CPU returns to the main program from which it was interrupted. In the interrupt routine we have the binary to BCD conversion and the display subroutines. As soon as the required time base say (one second or one minute) is reached, the Z80 CTC generates an interrupt wherein the CPU stops counting.

BINARY TO BCD CONVERSION

The number of counts stored is in binary form, hence it becomes necessary to convert them to BCD for easy read out from the display. The method is described below.

Two and three memory locations are assigned to store binary and BCD data respectively. The memory addresses for BCD data are initially cleared to zero. The process of shifting and checking data is repeated until all binary data bits are shifted left completely. The flow chart will be:

(1) Store binary data in RAM with a starting address of 1AOOH.
(2) Clear RAM section (starting address) 1AO2H for the BCD data.
(3) Shift the binary data stored in RAM left one bit. The left most bit is automatically transferred to the carry flag.
(4) Add carry to the BCD data (1AO2H) and then double the BCD data.
(5) Check if all the bits of binary data have been shifted out of the original memory section. If not repeat the step three. If yes, it is the end of the programme.

DISPLAY ROUTINE

Having got the total number of counts within a specified time base interval (the memory locations 1AO2H- 1AO4H) it becomes necessary to bring them into display for easy readout. We now make use of the monitor subroutine HEX7SG with address O678H. This converts any two hexa decimal numbers into seven segments display format. Thus the six hexa decimal number in locations 1AO2 H - 1AO4 H are brought into seven segment display format and displayed.

The complete assembly language programme is shown in listing. This has the time base interval of one minute. It can measure upto 100 kHz for 4 MHz system clock rate. The same programme is used to function as a tachometer and the flow chart of the programme is given in Fig:2.

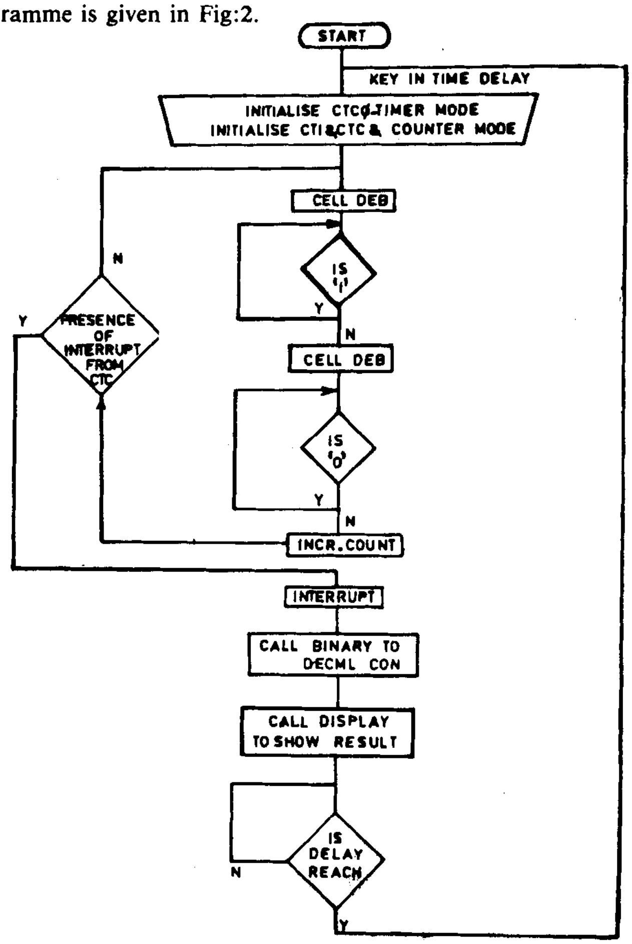
START
KEY IN TIME DELAY
INITIALISE CTCØ-TIMER MODE
INITIALISE CTI&CTC& COUNTER MODE
CELL DEB
IS '1'
Y
N
CELL DEB
IS '0'
Y
N
INCR.COUNT
PRESENCE OF INTERRUPT FROM CTC
N
Y
INTERRUPT
CALL BINARY TO DECML CON
CALL DISPLAY TO SHOW RESULT
IS DELAY REACH
N
Y

Fig.2: Flow chart of the developed programme

A debounce time delay of one millisecond is also provided for the above tachometer. Repetitive periodic counting can be incorporated by enlarging the software.

A slight modification in the operation of the Frequency counter gives rise to the Industrial counter time base interval and displayed the count for a certain time, it is made to go back to the original routine where it is made to count again and then display. This is gone through again and again. Using the return interrupt instruction the microprocessor is made to come out of the Interrupt routine. Consequently a compare statement makes the microprocessor go to the initial start location and the process is made to repeat again and again. Thus we are able to get the frequency display at frequent intervals and this helps us to monitor the fluctutations in frequency, rpm, rate of occurrence of events etc.

CONCLUSION

The unit has been tested successfully for the speed measurement in the range of 1 to 10 rpm especially used in slow but precise stirring arrangement. Also it has been checked for the measurement of 50 Hz mains input frequency. The unit is found to work satisfactorily and it will be interfaced for real time process control.

REFERENCES

1. Micro-processor MPF-I User's Manual & Monitor Program Source Listing
2. Lance A Leventhal, Z80 Assembly language programming (A book), Osborne / McGraw Hill Publishers, 1979