(Detailed solutions to problem that is not necessary to be covered during tutorial)

3.3 Generating Time Delays using Software

Fig. 3.3 shows a subroutine for generating a short time delay.

- (1) Calculate the delay produced by the subroutine in terms of number of execution cycles.
- (2) Given that the VIP processor system clock is 10MHz, what is the duration of the time delay produced by the routine? You may assume that each memory access cycle uses one cycle of the system clock.

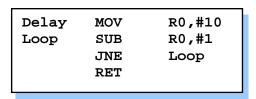


Fig 3.3 – Delay subroutine

(3) How can a time delay of 1ms (millisecond) be achieved?

(Solutions)

Note: This question shows how short and precise time delays can be generated using appropriate loop structures. It should be highlighted that in practical microprocessor systems, it is more likely that hardware timers with appropriate interrupts are used to generate precise time delays as such software methods can be very power hungry and prevents the processor from doing anything else that is useful during the generation of the time delay.

Delay Loop	MOV SUB	R0,#10 R0,#1
	JNE	Loop
	RET	

No. of cycles	No. of times executed
2	1
2	10
1	10
2	1

Fig. 3.3 - Delay subroutine

(1) Calculate the delay produced by the subroutine in terms of number of execution states.

Ans: Tallying up the total number of cycles will give $(2 + 2 \times 10 + 1 \times 10 + 2) = 34$ cycles.

(2) Given that the processor clock is 10MHz, what is the duration of the time delay produced by the routine?

Ans: At 10MHz, each cycle is $0.1\mu s$, therefore total delay = $34*0.1 = 3.4\mu s$

(3) How can a time delay of 1ms be achieved?

Ans: Delay can be increased by putting a larger value into R0 to increase the number of loops. A delay of 1ms will be generated by $(1 \times 10^{-3} * 10 \times 10^{6}) = 10000$ clock cycles if the clock is 10 MHz. This value can be obtained by solving the equation: $2 + 2 + (3 \times N) = 10000$. Solving for N gives 3332. So we change MOV R0,#10 to MOV R0,#3332.