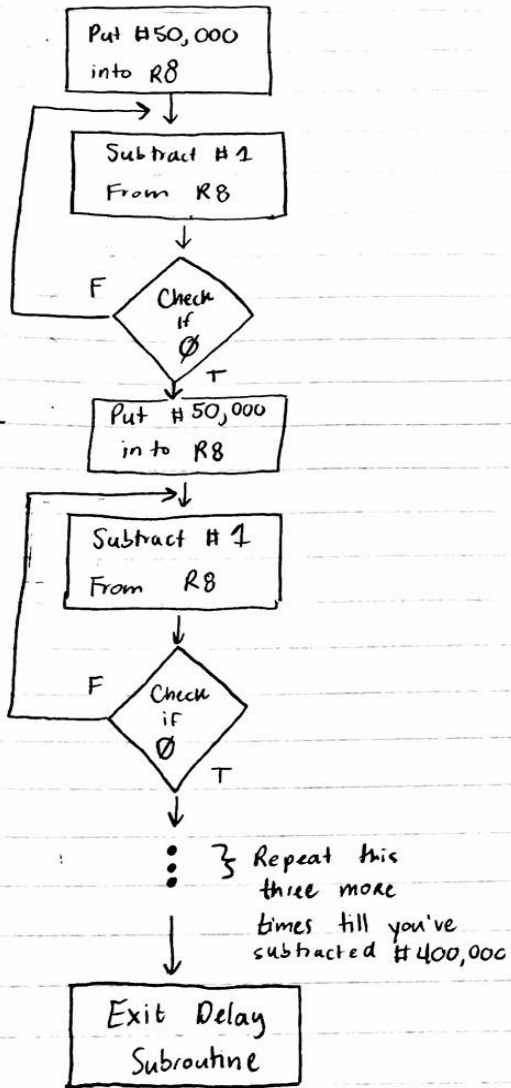


Part A (Delay 80-120ms)



Part A Pseudocode

```

R8 ← # 50,000
wait 1  R8 ← R8 - 1
        BNE wait 1
        R8 ← # 50,000
wait 2  R8 ← R8 - 1
        BNE wait 2
        R8 ← # 50,000
wait 3  R8 ← R8 - 1
        BNE wait 3
        R8 ← # 50,000
wait 4  R8 ← R8 - 1
        BNE wait 4
        R8 ← # 50,000
wait 5  R8 ← R8 - 1
        BNE wait 5
        R8 ← # 50,000
wait 6  R8 ← R8 - 1
        BNE wait 6
        R8 ← # 50,000
wait 7  R8 ← R8 - 1
        BNE wait 7
        R8 ← # 50,000
wait 8  R8 ← R8 - 1
        BNE wait 8
        BX    LR

```

④ Why #400,000? , Note $100\text{ms} = 1e8\text{ns}$

$$\text{Delay time} = \text{CLKspd} \times \text{Value of R8} \times \text{Cycles per instruction}$$

$$= 62.5 \text{ ns} \times R8 \times 4 \text{ cycles} \Rightarrow \text{subs} = 1$$

$$1e8ns = 250 \times R8$$

250 \rightarrow $R8 = \$400,000$

↑ but that's above limit of bits
so we used 50,000

```
;***** main.s *****  
  
; Program written by: Megan Cooper  
; Date Created: 2/12/2016  
; Last Modified: 2/12/2016  
; Section Wednesday 4-5  
; Instructor: Ramesh Yerraballi  
; Lab number: 2  
; Brief description of the program  
; The overall objective of this system an interactive  
alarm  
  
; Hardware connections  
  
; PF4 is switch input (1 means SW1 is not pressed, 0  
means SW1 is pressed)  
  
; PF3 is LED output (1 activates green LED)  
; The specific operation of this system  
; 1) Make PF3 an output and make PF4 an input  
(enable PUR for PF4).  
; 2) The system starts with the LED OFF (make PF3  
=0).  
; 3) Delay for about 100 ms  
; 4) If the switch is pressed (PF4 is 0), then  
toggle the LED once, else turn the LED OFF.  
; 5) Repeat steps 3 and 4 over and over
```

```
GPIO_PORTF_DATA_R      EQU    0x400253FC  
GPIO_PORTF_DIR_R       EQU    0x40025400  
GPIO_PORTF_AFSEL_R     EQU    0x40025420  
GPIO_PORTF_PUR_R       EQU    0x40025510  
GPIO_PORTF_DEN_R       EQU    0x4002551C  
GPIO_PORTF_AMSEL_R     EQU    0x40025528  
GPIO_PORTF_PCTL_R      EQU    0x4002552C  
SYSCTL_RCGCGPIO_R      EQU    0x400FE608  
PF4                    EQU    0x40025040  
PF3                    EQU    0x40025020
```

```
AREA    |.text|, CODE, READONLY, ALIGN=2  
THUMB  
  
EXPORT  Start  
  
Start  LDR R0,= SYSCTL_RCGCGPIO_R  
        LDR R1,[R0]  
        ORR R1,#0x20  
        STR R1,[R0]  
  
        NOP  
        NOP  
  
        LDR R0,= GPIO_PORTF_DIR_R  
        LDR R1,[R0]  
        BIC R1,#0x10  
        ORR R1,#0x08  
        STR R1,[R0]
```

```
LDR R0,= GPIO_PORTF_AFSEL_R
LDR R1,[R0]
BIC R1,#0x18
STR R1,[R0]
```

```
LDR R0,= GPIO_PORTF_DEN_R
LDR R1,[R0]
ORR R1,#0x18
STR R1,[R0]
```

```
LDR R0,= GPIO_PORTF_PUR_R
LDR R1,[R0]
ORR R1,#0x10
STR R1,[R0]
```

```
LDR R0,= PF3
LDR R1,[R0]
MOV R1,#0x0
STR R1,[R0]
```

```
loop    BL Delay
```

```
LDR R0,= PF4
LDR R2,= PF3
```

```
LDR R1,[R0]
CMP R1,#0
BEQ Toggle
BNE Clear
```

```
Toggle LDR R1,[R2]
EOR R1, R1, #0xFF
STR R1,[R2]
B loop
```

```
Clear LDR R1,[R2]
AND R1, R1, #0x0
STR R1,[R2]
```

```
B loop
```

```
Delay MOV R8, #50000
wait1 SUBS R8, #1
BNE wait1
MOV R8, #50000
wait2 SUBS R8, #1
BNE wait2
```

```

MOV R8, #50000
wait3 SUBS R8, #1
      BNE wait3
      MOV R8, #50000
wait4 SUBS R8, #1
      BNE wait4
      MOV R8, #50000
wait5 SUBS R8, #1
      BNE wait5
      MOV R8, #50000
wait6 SUBS R8, #1
      BNE wait6
      MOV R8, #50000
wait7 SUBS R8, #1
      BNE wait7
      MOV R8, #50000
wait8 SUBS R8, #1
      BNE wait8
      BX LR

ALIGN      ; make sure the end of this section
END        ; end of file

```

| MC Simulated Time | Real Time |
|-------------------|---------------|
| 3.562 sec | 10 sec |