

Memo

Q 1 a):

```

                                mov    #$00,DDRB      ; port B inputs
                                mov    #$FF,DDRD      ; port D outputs
                                mov    #$08,count     ; starting count
                                lda     count
                                swapa                    ; swap nybbles
                                sta     PTD           ; output count on LEDs
putleds:                        jsr     getbuttons
                                lda     count
                                brz     freeze        ; if zero, freeze
                                cmp     #$0F          ; compare with 1111
                                brz     freeze        ; if zero (same), freeze
                                swapa                    ; swap nybbles
                                sta     PTD           ; output count on LEDs
                                jmp     putleds        ; and do again

freeze:                          wait                ; wait till reset
```

,*****

; subroutines

,*****

```

getbuttons:                     mov    #$00,count
                                brset   1,PTB,next
                                dec     count
                                jsr     delay          ;0.1s delay for debounce
next:                           brset   0,PTB,fin
                                inc     count
fin:                             rts
```

Q2:

a)

```

                                mov    #$FF,DDRB      ; port B outputs
                                mov    #$00,DDRD      ; port D inputs
                                bset    7,DDRD        ; except D7

or                               mov    #$80,DDRD

DAC_out:                        bset    7,PTD         ; raise R/W
                                sta     PTB          ; output data byte
```

```

        jsr    delay                ; delay for say 10 us
        bclr   7,PTD                ; lower R/W
waitEoC: brclr   6,PTD,waitEoC      ; wait for EoC
        bset   7,PTD                ; raise R/W
        rts

```

b) ramp adc:

```

next:    mov    #$00,Timeout1      ; starting level
        lda     Timeout1           ;
        jsr     DAC_out            ; output it
        brset   5,PTD,finish       ; DAC larger than vin
        inc     Timeout1
        jmp     next

finish:   dec     Timeout1
        wait

```

c) SA adc:

```

        mov     #$00,Timeout1      ; starting approx
        mov     #$80,Count         ; starting mask
        mov     #$08,Loop          ; loop counter

next:    lda     Timeout1
        add     Count              ; add mask bit to present approx
        jsr     DAC_out            ; output new approx
        brclr   chmask            ; if less, keep mask bit
        sub     Count              ; else discard
chmask:  sta     Timeout1          ; best approx stored
        ror     Count              ; move mask bit to right
        dbnz    Loop,next

finish:   wait.

```

Q 3:

- a) \$F61E
- b) \$F621
- c) \$39
- d) \$3980
- e) \$3B82F7
- f) 0
- g) \$78

h) 56

i)

two_seconds:

mov #c1,count3

delay: mov #\$3f,count1

outside: mov #\$3f,count2

inside:

dbnz count2,inside

dbnz count1,outside

dbnz count3,delay ;7 933 824 cycles

;correction loops

mov #\$c8,count1

outside2: mov #\$3f,count2

inside2:

dbnz count2,inside2

dbnz count1,outside2 ;65962 cycles

mov #\$C6,count2

inside3:

dbnz count2,inside3 ;209 cycles

nop ;1 cycle

rts ;4 cycles

time = count3*(count1*((count2)*5+9)+4)+4

Try: FC, F6 +3 nops, 16

Q4:

a) 0.0195m

b) non-inverting amplifier with gain of 5/3 (draw it)

c) 0.0117m

d) 0.000305m