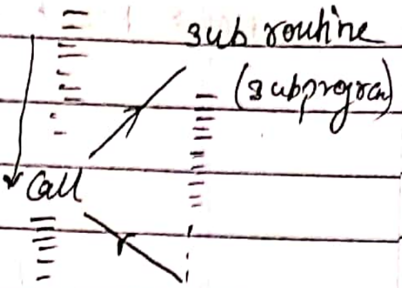
Call instructions

ACALL (Absolute call)
LCALL (Long call)



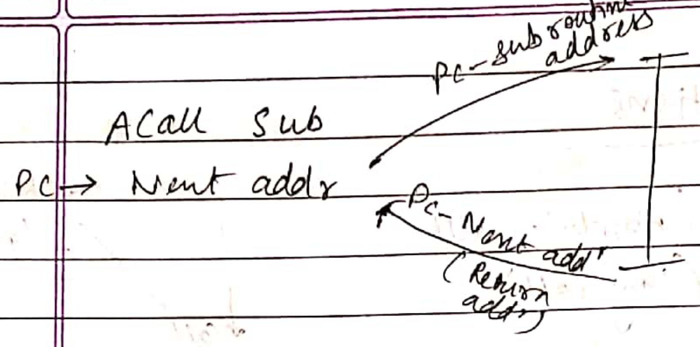
DJNZ R0, back
↳ main 001 – FF
main 255 counts
to count 266 start from 00h ← R0
00h → – FF ← 00
∴ total 256 couns.

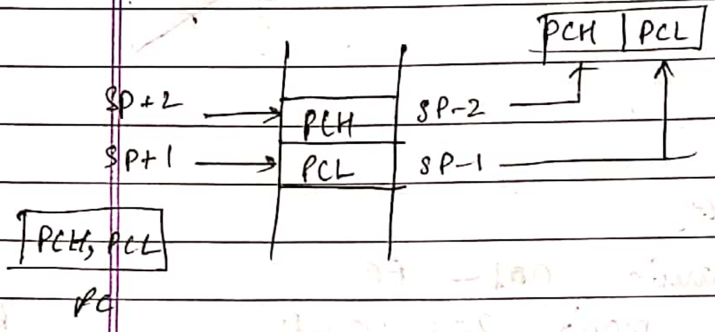
* when call is then PC contain is ~~there~~ copied to stack and then sub routine is executed and the return statement ~~st~~ load the PC with stack value .

LIFO Fation is helpfull in case of nested loop.

* All ISR are subrouting but all subroutine can't be ISR

0101
1010





Q) Write a prograggm to toggel all the bits of poot 1 by sending the values 55 H and AAH continuesly put a time delay b/w each values send to point.

```
MOV  P1, 55

HERE: XRL  P1, #FFh
      ~~SJMP~~  ~~HERE~~
back: MOV  R0, #20h
      DJNZ R0, back
      DJN2 HERE


MOV  P1, #55h
MOV  a, #FFh
XRL  P1, a
```

Date
Page

```
ORG 00H
back : mow P1, #55h
       LCALL DELAY
       MOV P1, #0AAh
       LCALL DELAY
       SJMP BACK

Delay : MOV R1, #255h
Here : DJNZ R1, Here
       RET
       END
```

Assembler directive → for the assembler to understand

EQU directive

```
Count EQU 25
MOV R0, # count        R0 = 25
```

ORG 500H → from where to start decoding

```
500H  Data1 : DB 28d
501H  Data 2 : DB 11011101 B
502H  Data 0 : DB 20h
503H  Data 4 : DB "3132"
504H  Data 5 : DB 'J'
```

→ not 16 bit

31 → 1   ASCII

32 → 2

∴ 12 8 bits.

```
505H  Data 6 : DB "All the best".
```

506H 507H 508H space (509H)

8051 ALP Factorial of a no (upto 5)

```
MOV r0, #25h

        MOV a, r0
        Dec r0
back:   MOV b, r0
~~back:~~ MUL ab
        ~~MOV @ r1, a~~
        ~~DEC r0~~
        DJNZ r0, back
        MOV 26h, a
```

Fibonacci series till 12

```
        MOV 12H, #1
        MOV 13H, #1
        MOV r1, #0AH
        MOV a, 12H
back:   ADD a, 13h  → MOV ~~13h~~ r0, 13h
        MOV 13h, a
        MOV a, r0
        DJNZ r0, back
```

```
        MOV Ro, #20H
        MOV R1, #10
Back:   MOV A, @Ro
        INC Ro
        ADD A, @Ro   → DAA
        INC Ro
        MOV @Ro, A
        DEC Ro
        DJNZ R1, Back
```

* ADD two 32 bit number

12, 13, 14, 15 | 16, 17, 18, 19 | 1A, 1B, 1C, 1D |

```
       MOV r0,04h
       MOV r1, #12 h
       MOV r2, #16 h
       MOV r3, # 1Ah
       CLR C
back:  MOV a, @r1
       ADDC a, @r2
       MOV @r3, a
       INC r1
       INC r2
       INC r3
       DJNZ r0, back
```

* WAP to find the address of 1st ~~two~~ internal ram location b/w 20h to 60h which contains ~~cony~~ cosecutive number.

```
         MOV r1, #20h
         MOV r0, #41h
back :   MOV a, @r1
         Inc r1
         SUBB a, r1         →  MOV r0, a
         ~~INC~~ ~~LAST~~          CJNE r0, #00, last
last :   DJNZ R0, back
~~LAST~~ : MOV a, r1 .
         ~~DEC~~ MOV r2, a
         Dec r1
         MOV a, r1
         MOV r3, a
```

3)18(
18

xch a,r1

6 5 → a b

⇐ LCM. of two no.

```
ORG 00h
mov r0, 20h
mov r1, 21h
mov r2, #01h
mov a, r0
mov b, r1
subb a, r1
jc next
back: mov a, r0
mov b, r2
mul ab
mov r3, a
mov b, r1
div ab
inc r2
xch a,b
cjne a,#00h, back
sjmp last
next: mov a, r0
xch a, r1
mov r0, a
sjmp back
last: mov 22h, r3
sjmp $
END
```

Q) To check nibble wise pallindra

```
    MOV  R2, #00h
    MOV  R3, #00h
    MOV  a, 25h
    MOV  r0, a
    SWAP a
    MOV  r1, a
    CLR
    MOV  a, r0
    SUBB a, r1
    CJNE a, #00, NP
    INC  R2
    SJMP NEXT
NP: INC  R3
NEXT: SJMP $
      END
```

Q) Cube of no'

```
mov  a, #0FFh                      { a = FF }
mov  b, #0FFh                      { b = FF }
mov  r0, #0FFh                       r = FF
MUL  AB                                          255
                                                ×255
MOV  R1, B                        R1 = HB
MOV  B, R0                        B = FF
MUL  AB
MOV  50H, A
MOV  R5, 50h                         FF × FF
MOV  R2, B
```

$\underbrace{FE}_{B}\underbrace{01}_{A}$

```
MOV  A, R1
MOV  B, R0
MUL AB
ADD  A, R2
JNC  Skip
MOV  R3, #00H
INC R3
SKIP: MOV 51H, A
MOV  R6, 51H
MOV  A, B
ADD  A, R3
MOV  52H, A
MOV  R7, 52H
```

FF
FF

15) 255 (

16) 255

1F

FF
FF

Q) To check bitwise pallindrom

MOV A, #1Fh

1	0	0	0	0	0	0	1

(4) Cube of a number using push and pom

```
MOV  R0, #20h
MOV  A, @R0
MOV  R1, A
MOV  B, A
INC  R0
MOV  A, @R0
MUL  AB                 MOV SP, #30h
PUSH 0F0H
MOV  B, A  // B = 01h
MOV  A, R1 // A = FFh
MUL  AB
INC  R0
MOV  @R0, A
MOV  A, B
MOV  R2, A
POP  0F0H
MOV  A, R1
MUL  AB
ADD  A, R2
INC  R0
MOV  @R0, A
MOV  A, B
ADDC A, #00h
INC  R0
MOV  @R0, A
MOV  A, B
ADDC A, #00h
INC  R0
MOV  @R0, A
```

Q) To implement DA A operation without using DA·A .

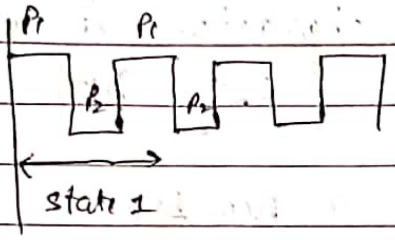
Q) To multiply two 16 bits number.

Q) To check given character is a character wise pallindrom. (using DB)

Q) To find HCF of a number.

delay
Software ~~delay~~ program

12 clock pulses are required to execute 1 instruction



$$T = \frac{\text{Osc freq}}{12}$$

Q) Find the time period required to execute 1 machine cycle in each case.

① 11.0592 MHz ⇒ $\left(\frac{11.0592}{12}\right)^{-1} = (0.9216)^{-1}$

$= \underline{1.085\ \mu s}$

② 16 MHz ⇒ 0.75 μs

③ 20 MHz ⇒ 0.6 μs

④ 22 MHz ⇒ 0.54 μs

Q) Find the size of the delay in the following program if crystal freq is 11.0592 MHz

```
       MOV   A,#55h
Again: MOV   P1 , A
       ACALL Delay
```

```
        CPL A
        SJMP  AGAIN
Delay : MOV   R3, #200    ——  1          }
Here :  DJNZ  R3, Here    ——  2 × 200    } × 1.085 µs
        RET               ——  2          }
```

= 437 µs

loop delay

Q) Find the delay program for the following sub program (not given f = 11.0592)

```
        MOV  R3, #250   ——  1          }
        NOP             ——  1  × 250   }
        NOP             ——  1  × 250   }
        NOP             ——  1  × 250   }  × 1.085 µs
        NOP             ——  1  × 250   }
        ~~NOP           ——  1  × 250~~ }  ≃ 1630.75 µs
Here :  DJNZ R3, Here   ——  2  × 250   }
        RET             ——  2
```

Q) If crystal frequency is 22 MHz generate a delay of 5 ms

① $5 \times 10^{-3} \doteq 0.54 \times x \times 10^{-6}$

~~$x = \frac{0.54 \times 10^{-6}\ 10^{-3}}{5 \times 10^{-3}}$~~

~~$= 0..$~~

② $x = \frac{5 \times \cancel{10^{-3}}\ 10^{3}}{0.54 \times \cancel{10^{-6}}}$

$= 9259$ cycles

3) DJNZ - 2MC $\frac{9259}{2}$ - 4630 cycle

4) $\frac{4630}{256}$ = 18 count

$$(\underbrace{256}_{R1} \times \underbrace{18}_{R2}) = 4630$$

```
        MOV  R2,#18        — 1×            }
Here1: MOV  R1,#255       — 1×18          }
Here:  DJNZ R1, Here      — 2×256×18      }  × 0.54
       DJNZ R2, Here 1    — 2×18          }
```

This is extra

= 5.006 ms ≃ 5ms

check this way

Q) WAP to toggel all the bits of port 1 every 200 ms. Crystal frequ 11.0592

$$200 \times 10^{-3} = n \times 1.085 \times 10^{-6}$$

$$n = \frac{200}{1.085} \times 10^{3}$$

$$= 184331.7$$

$$= 184332 \text{ cycles}$$

DJNZ — 2 → $\frac{184332}{2}$ = ~~181~~ 92166 cyc

$\frac{92166}{256} = \frac{360}{\boxed{256}} = 1.4$ $\frac{360}{180} = 2$

divide by 180

256 × 360

38 256 N

~~256 × 256 × 180~~

→ 256 × 180 × 2

```
DELAY:   MOV   R3, #2           - 1 x
Here 2:  MOV   R2, #180         - 1 x 180  2
Here 1:  MOV ~~DJNZ~~ R1, #255  - 1 x 180 x 180
   Here: DJNZ  R1, Here         - 2 x 256 x 180
         DJNZ  R2, Here 1       - 2 x 180
         DJNZ  R3, Here 2       - 2 x 1 x 2
         RET  - 2
```

≃ 2001 ms

```
AGAIN: MOV   A, #55h
       MOV   P1, A
       ACALL DELAY
       CPL   A
       SJMP  AGAIN
```

Q) WAP to generate a square wave of frequency 1 KHz assume crystal frequency of 8051 is 11.092 MHz

MHz

⇒ Ton = Toff   Time = 1 ms square wave for 1 cycle

Ton = Toff = 0.5 ms

$0.5 \times 10^{-3} = x \times 1.085 \times 10^{-6}$

$x = 460.8$

$= 460$ cycles

DJNZ → $\frac{460}{2} = 230$ cycles

```
delay: MOV  R0, #230
Here : DJNZ R0, Here
       RET
```

* RESET → is non maskeble

other are maskable

5 interrupts

8 byte differ

		Vector address
$\overline{INT0}$	[ext H/w interrupts]	0003H
$\overline{INT1}$	[ext H/w interrupt]	0013H
T0	[Timer 0 interrupt] [TF0]	000BH
T1	[Timer 1 interrupt] [TF1]	001BH
Serial comn	[TI & RI flags]	0023H
RESET		0000H

TI → if 1 then 1 byte of data have been trasferred

TI – Transmission, RI – Receive.

this is also considered same time.

When we press reseat the BIOS program should run.

Reset → 0000H LJM L1 L1: BIOS Program

1 by 2 byt

total 3 byte is requird.

interrupt enable

* IE reg [bit addressable reg]

EA	—	ET2	ES	ET1	EX1	ET0	EX0

EA = 0 – disable all interrupts (all interrupt is masked)

EA = 1 – enable interrupts and are serviced if particular interrupt bit is 1

Q) So the instrn to do the following

a) enable serial interrup
b) " T0 . "
c) " Ex1 "

a) SetB IE.4

b) SetB IE.1

c) SetB IE.2

d) Clr IE.1

e) ~~Set~~ Clr ~~0~~ IE.7

d) Disable or mask Timer 0 interrupt
e) to disable all the interrupt

Mam

for 1st three (a,b,c) → MOV IE, #$\overbrace{10010110}^{96H}$ B

IP Rg (Interrupt priority Rg
bit addresable)

-	-	PT2	PS	PT1	PX1	PT0	PX0

lowest ← Heighest

dyfault

If Px1 is set 1 in IP Rg then 1st ~~serm~~ PX1 is ~~so~~ serviced after that the priority order is followed.

① Discuss what happens if interrupt INT0, TF0, ~~IE0~~, INT1 are activated ~~at~~

INT0
TF0 ↓
INT1

Q) To assign heigher priority of Timer 1 interrupt TF1. discuss what hapen if INT0 & INT1 & TF0 are activated at the same time.

a) SetB IP.3 or MOV IP, #08h

b)

TF1
INT0
TF0 ↓
INT1

a) Discuss the seqn

MOV IP, # 0000 1100 B

PX1 INT 1

PT TF1

INT 0

TF0

PS

(TI & RI)