NPTEL
Summer Workshop on Microcontrollers
WEL, IIT Bombay

Quiz-1 Date: June 9, 2022

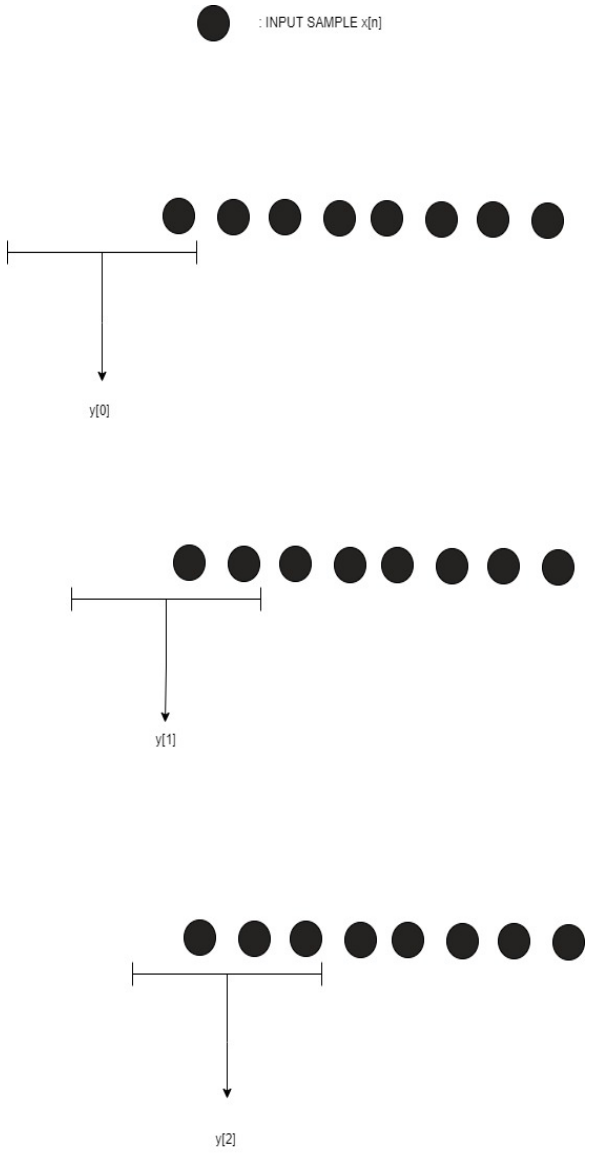
Moving Average Filter

Moving Average Filter is used in Signal Processing and Data Analysis to smooth out high frequency noise and short-term fluctuations. Write an assembly language program to implement a 4-point Moving Average Filter on an input signal consisting of 8 samples.

The formula for output signal is provided-

$$y[n] = \frac{x[n] + x[n-1] + x[n-2] + x[n-3]}{4}$$

Assume that $x[-1],\ x[-2]$ and $x[-3]$ are equal to 0.



And so on...

```
// -- DO NOT CHANGE ANYTHING UNTIL THE **** LINE--//
ORG 0H
LJMP MAIN
ORG 100H
MAIN:
CALL FILT
HERE: SJMP HERE
ORG 130H
// ****************

FILT:
// ADD YOUR CODE HERE
RET
END
```

- The input signal $x[n]$ samples must be stored in memory locations `60H` to `67H`.
- The output signal $y[n]$ samples must be stored in memory locations `70H` to `77H`.
- To reduce the effort involved in adding multiple items in memory locations, you can use the command window in Keil.
  - Start a Keil debugging session.
  - Enter the following command in the Keil command window to load an array of 8 numbers represented in decimal format. The `I:60h` refers to indirect addressing of location `60H`. To inspect the memory, you should enter `I:0x60` in the Keil memory window.

```
E char I:60h = 04h,24h,38h,11h,3ah,2bh,0eh,69h
```

- You can assume that the addition result never crosses 8 bits (i.e there is no overflow and each operation can be done in a single instruction).

Checkpoints

Check the following cases:

- Input samples : `04H`, `24H`, `38H`, `11H`, `3AH`, `2BH`, `0EH`, `69H`.
  Output samples : `01H`, `0AH`, `18H`, `1CH`, `29H`, `2BH`, `21H`, `37H`.
- Input samples : `56H`, `1AH`, `33H`, `4DH`, `46H`, `02H`, `4EH`, `56H`.
  Output samples : `15H`, `1CH`, `28H`, `3CH`, `38H`, `32H`, `38H`, `3BH`.

IMPORTANT HINT : Use DIV instruction for performing division by 4. You can only use accumulator: A and register: B for performing division and NO OTHER REGISTERS. For example, if we want to divide 13 by 4

```
MOV A,#13
MOV B,#4
DIV AB
```

After the "DIV" instruction is executed, accumulator: A contains the quotient, and register: B contains the remainder.

$$a = 3$$

$$\text{b} = 1$$