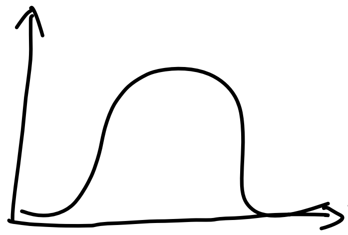
$$y[n] = \sum_{i=0}^{N} b_i \cdot x[n-1]$$

$b \circledast x \to 2^{16}$

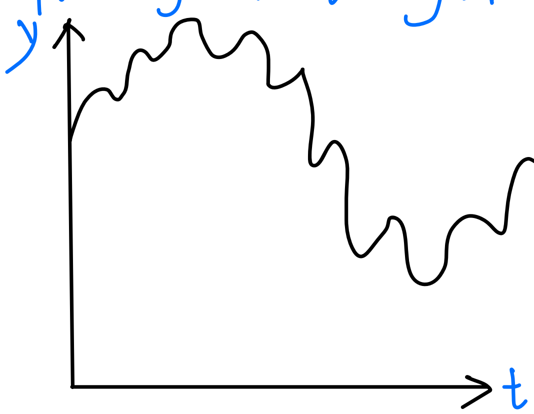
64 → $b(s)$

16bit

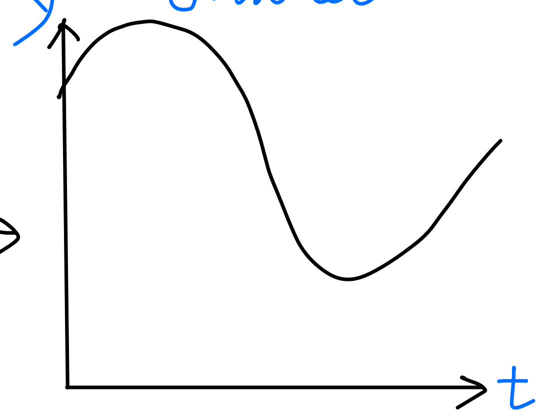


filter.

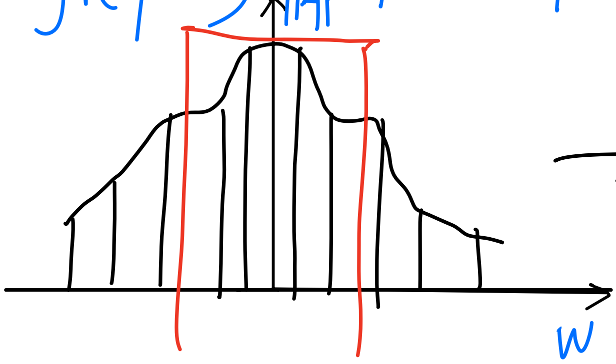
Input signal with high noise



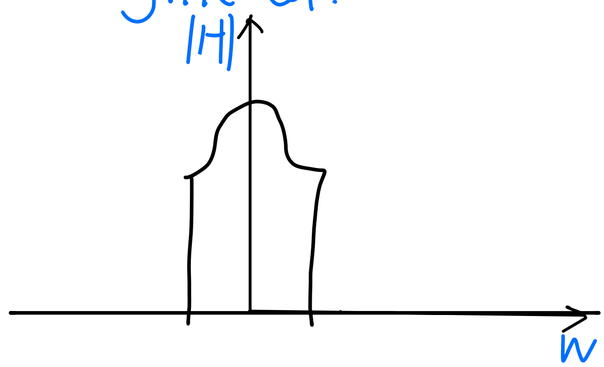
filtered



frequency Response input



filtered.



$$y[n] = \sum_{i=0}^{N} b_i\, x[n-i] \overset{\text{z-Laplace}}{\Longleftrightarrow} Y[z] = H[z]\cdot X[z].$$

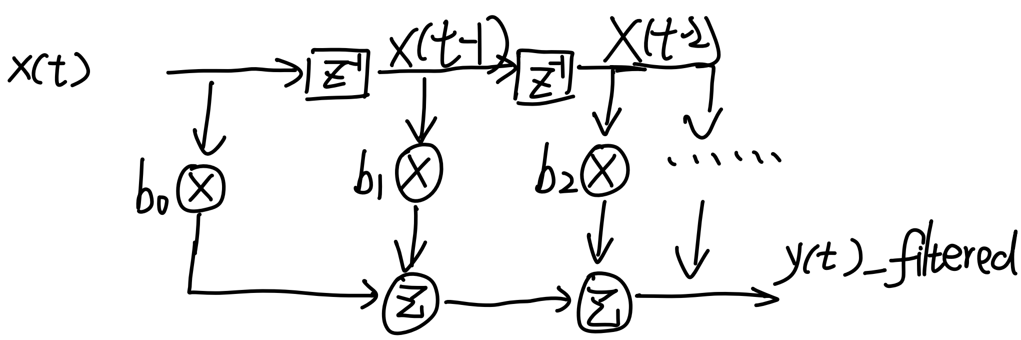
$$b \circledast x$$

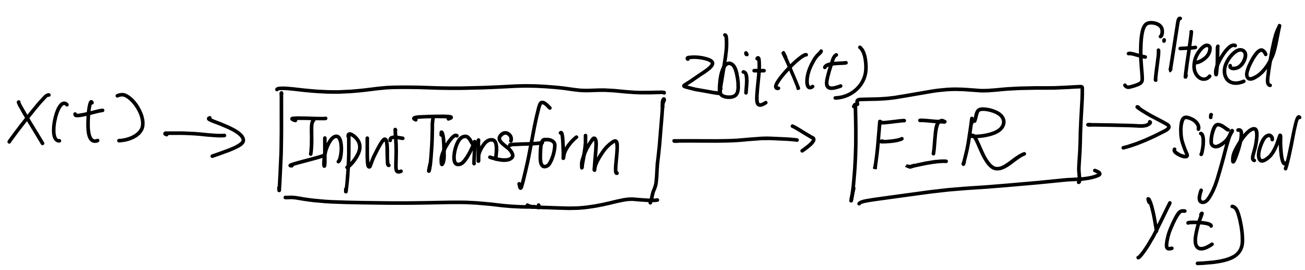
As we know: $x[n-n_0] \Leftrightarrow z^{-n_0}\cdot X(z) \longrightarrow X(z) = \sum_{-\infty}^{+\infty} x[n]\cdot z^{-n}$

$z^{-n} = e^{-jn\omega T} \rightarrow$ delay $x(t)$ with nT

$h(z) = \sum_{k=0}^{N} b_k \cdot z^{-k}$, $N = 63$ in this case.

We could draw a signal diagram for FIR filter,





d-in

loop, t<1e^4

$\underbrace{t=1e^4}_{x(t)} \rightarrow \underbrace{1e^4\ 16'b\ 9}_{x[n]} \rightarrow y[n] = \text{filter} \circledast x[n]$

$y[n]$

$$y[n] = \sum_{0}^{63} b_i \cdot x[n-i]$$

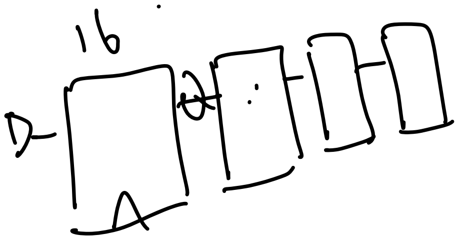
$y[0] \leftarrow x[0]$

$x[1] \rightarrow y[1] \rightarrow 1e^4\ y[n]$

0

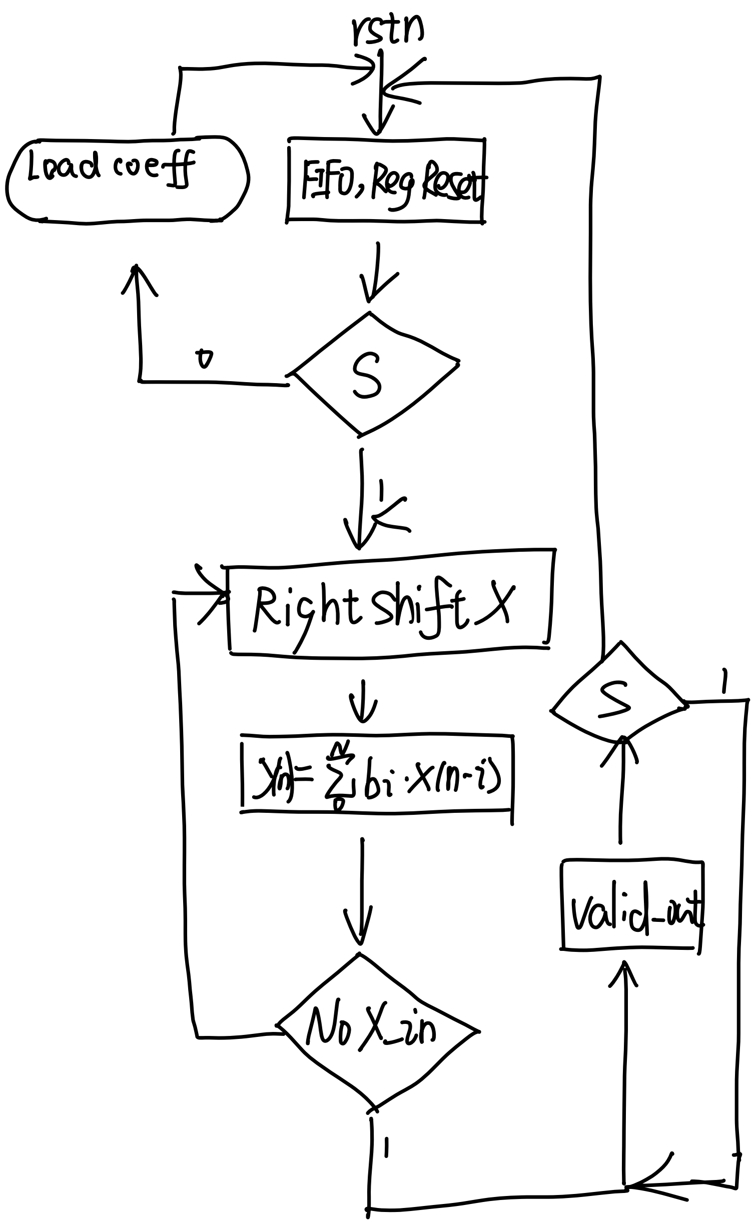
0 1 2 3 → $1e^4$

X lef-shift.



64x 16 bit wire

64 - 16bit wire



FIR pseudo code

```
module FIR #(Parameter NBits = 64)(
  output [NBits-1:0] dout,
  output reg valid-out
  input clk1, clk2
  input rstn,
  input din
  input valid-in)
   reg ([5:0]) X [0:NBits-1]
    always @(posedge clk)
     begin
     If(!rstn) begin
        Reset dout, shifter
     end else begin.
        if(S)
          RightshiftX
         for i=0, i≤NBits-1, i++
           dout = dout + bi Xi
         end for
       if(No more X-in)
         valid-out = 1.
       end.
```