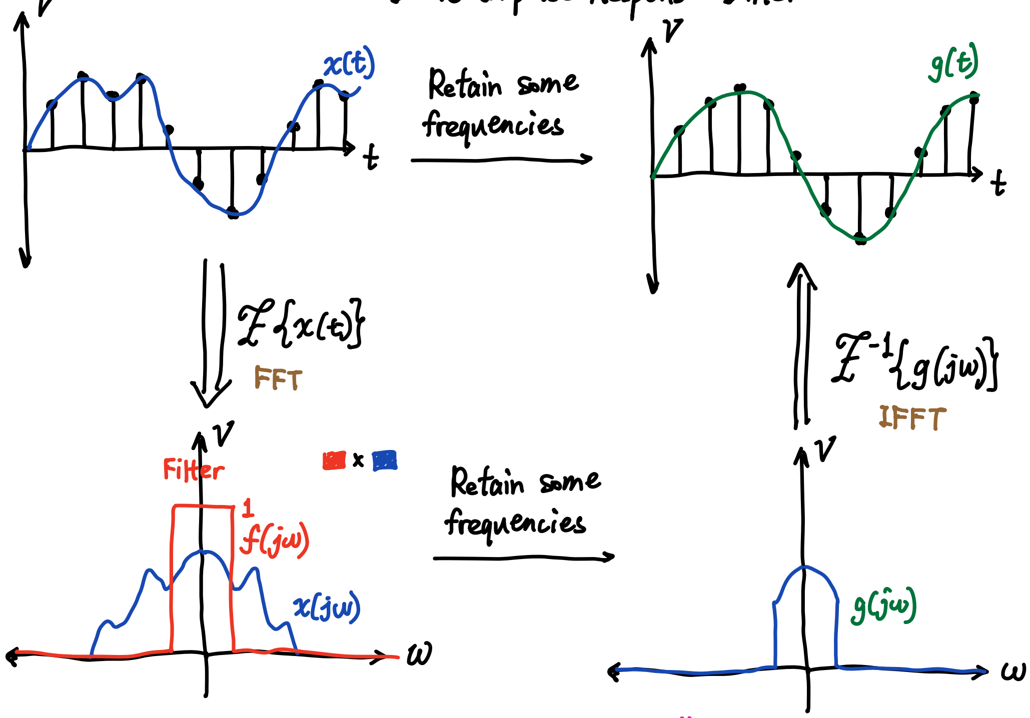
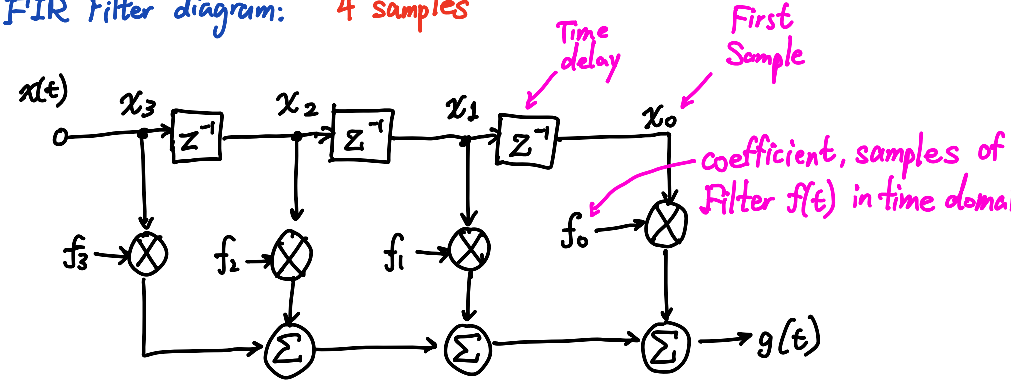
Finite Impulse Response Filter



$x(j\omega)\cdot f(j\omega) = \boxed{x(t) * f(t)}$ FIR filter does this.

FIR Filter diagram: 4 samples



Convolution: Multiplication of one signal by another and the integrating that product.

Because number of f_i is finite, filter can't be ideal. (Trade passband ripple vs. roll off slope)

$$y[n] = \sum_{k=0}^{N} b_k\, x[n-k] \qquad H(z) = \sum_{k=0}^{N} b_k z^{-k} = b_0 \cdot \prod_{k=1}^{N} (1 - c_k z^{-1})$$

Only pole at $z = 0$.

$$h[n] = \begin{cases} b_n, & 0 \le n \le N \\ 0, & \text{otherwise} \end{cases}$$

Optimize RMSE or Maximum Error : $H(e^{j\omega}) \Leftrightarrow H_d(e^{j\omega})$

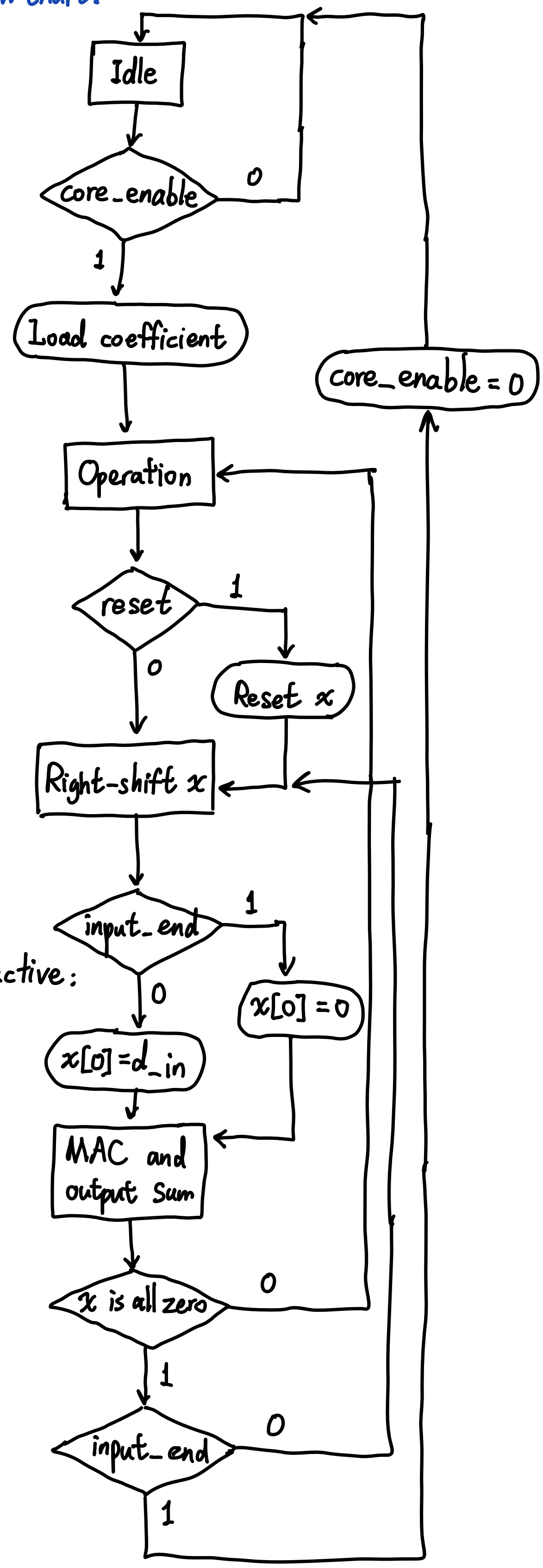
This class (RMSE)   desired (H_d)

FIR can have arbitrary magnitude/phase response, can obtain linear phase.

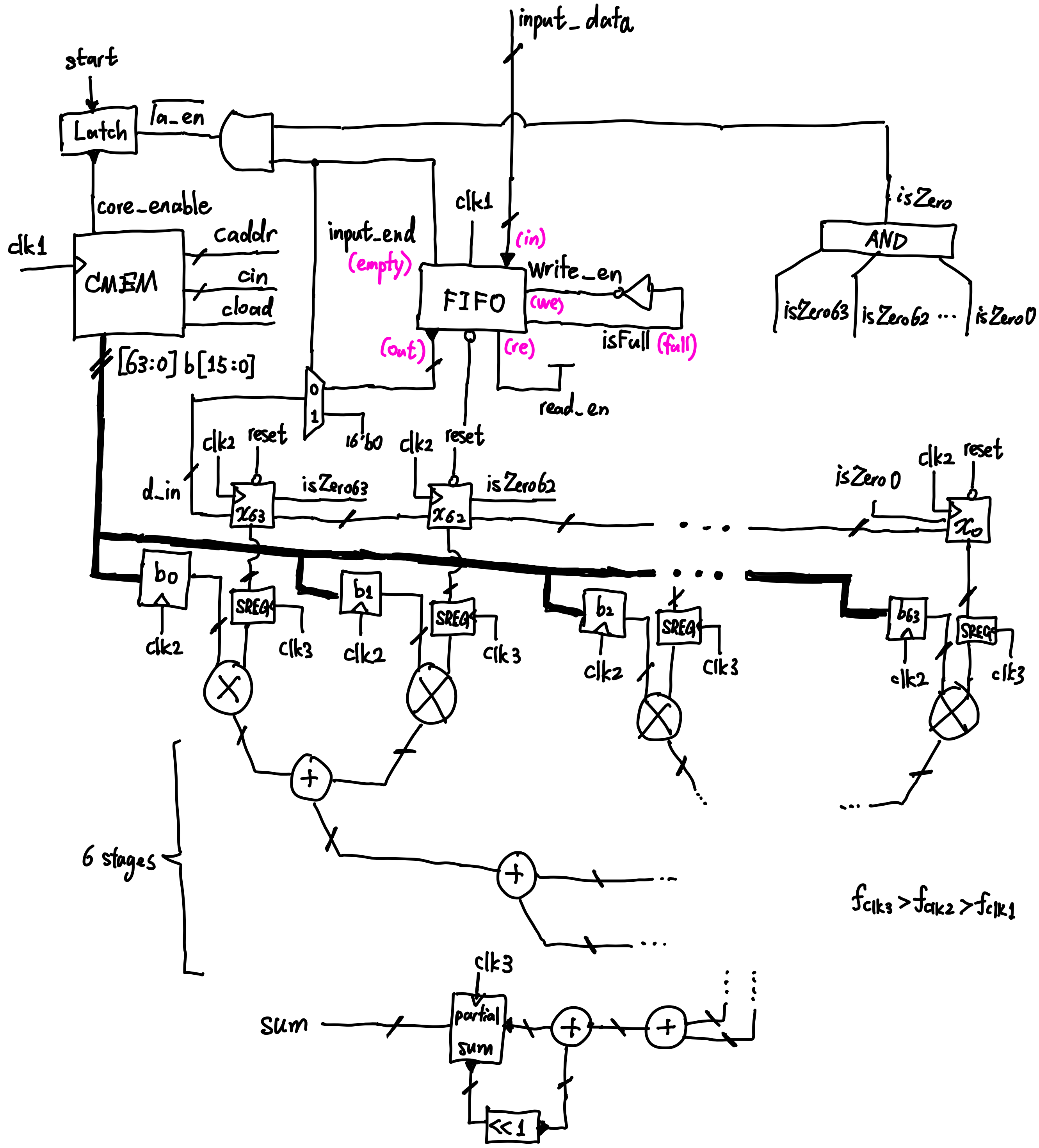
PIR pseudocode:

```
while core_enable is True:
    Load coefficient
    If reset is active:
        reset x
    End if
    for i=63, i>0, i=i-1:
        x[i] = x[i-1]
    end for
    If input_end is not active:
        x[0] = d_in
    End if
    Else:
        x[0] = 0
    for i=63, i≥0, i=i-1:
        sum += b[63-i]·x[i]
    end for
    output sum
    If x is empty AND input_end is active:
        core_enable = False
    End if
End while
```

ASM chart:



Datapath:



input_data
start
Latch
la_en
isZero
core_enable
clk1
caddr
cin
cload
CMEM
input_end
(empty)
clk1
(in)
write_en
(we)
FIFO
isFull (full)
(out)
(re)
read_en
AND
isZero63
isZero62 ···
isZero0
[63:0] b[15:0]
0
1
16'b0
d_in
clk2
reset
isZero63
x63
clk2
reset
isZero62
x62
isZero0
clk2
reset
x0
b0
b1
b2
b63
SREG
clk2
clk3
6 stages
sum
clk3
partial sum
<<1
f_clk3 > f_clk2 > f_clk1