Lecture 6

- fixed-pt implementation
- zero-placing FIR filter design
- Linear phase filters
- FIR implementation on the DSK

Fixed-point

ex Assume $0 \le |h[k]| \le 1$ $k = 0, 1, \ldots, N$

4 bits of resolution

If $h[k] = .703125\, \delta[k]$ (one filter coeff)

(decimal)		(binary)
0.703125 $\left(= \frac{1}{2} + \frac{1}{8} + \frac{1}{16} + \frac{1}{64}\right)$	$\longleftrightarrow$	0.101101
↓		↓
multiply by $2^3 (=8)$	$\longleftrightarrow$	bit shift 3 bits left
↓		↓
5.625 $\left(= 4 + 1 + \frac{1}{2} + \frac{1}{8}\right)$		0101.101 (sign bit)
↓		↓
round to nearest integer	$\longleftrightarrow$	round to nearest integer
↓		↓
6		0110

NB: round(5.625)
= floor(5.625 + 0.5)
= floor(6.125)
= 6

round(0101.101)
= floor(" + 0.1)
= floor(0110.001)
= 0110

fixed-pt. implementation

$$x[n] \longrightarrow \boxed{2^3 H(z)} \longrightarrow 2^3 y[n] \xrightarrow[\text{down scale}]{\times 2^{-3}} y[n]$$

$x[n]$	$2^3 H(z)$	$2^3 y[n]$	$y[n]$
5	6	30	$30/8 = 3.75$
0101 (short)	0110 (short)	0001 1 1 1 0 (int)	0011.11 (short)

$\underset{\text{sign}}{0}\,\underset{64}{0}\,\underset{32}{0}\,\underset{16}{1}\,\underset{8}{1}\,\underset{4}{1}\,\underset{2}{1}\,\underset{1}{0}$

bit shift right 3 bits

At the output, $y[n]$ is cast as a short, so the output to the fixed-pt codec is

$$y[n] = 3$$

<u>Compared to floating-point</u>

$$x[n] \longrightarrow \boxed{H(z)} \longrightarrow y[n] \xrightarrow[\text{as short}]{\text{typecast}} \text{output to codec}$$

$x[n]$	$H(z)$	$y[n]$	output to codec
5	.703125	3.515625	3
(short)	(float)	(float)	

<u>On the DSK</u>

(fixed)
- 16-bits of resolution (one sign bit) (short)
- Scale filter coeff. $h[K]$ by 2^{15} → · fixed-pt arithmetic
- filtered output stored in 32-bit int.
- output sent to codec scaled by 2^{-15} (bit shift 15 places to the right or (output >> 15) in C) and typecasted as a 16-bit short

(floating-pt)
- 32-bit IEEE floating-pt (~ 6 decimal places)
- floating-pt arithmetic
- filter output store in 32 bit float
- output sent to codec typecasted as a 16-bit short

Zero-placing FIR design

$$h[k] \text{ real} \longleftrightarrow H(e^{j\omega t_0}) = \overline{H(e^{-j\omega t_0})}$$

or

$$H(z) = \overline{H(z^{-1})}$$

Implies that the poles and zeros of $H(z)$ must appear in complex conjugate pairs, or they must be real. (i.e. $\omega = 0, \pm\frac{\pi}{t_0}$).

To place a zero at $\omega = \omega_1$,

$$H_1(z) = \left(1 - e^{j\omega_1 t_0} z^{-1}\right)\left(1 - e^{-j\omega_1 t_0} z^{-1}\right)$$

$$= 1 - e^{j\omega_1 t_0} z^{-1} - e^{-j\omega_1 t_0} z^{-1} + z^{-2}$$

$$= 1 - 2\underbrace{\frac{\left(e^{j\omega_1 t_0} + e^{-j\omega_1 t_0}\right)}{2}} z^{-1} + z^{-2}$$

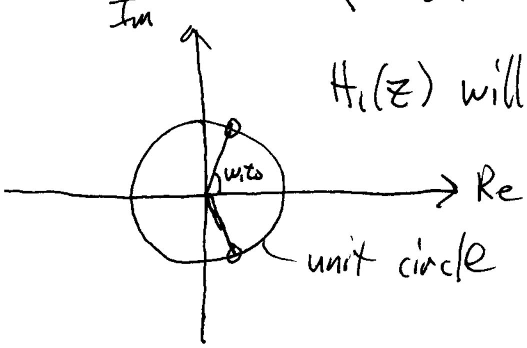
$$= 1 - 2\cos(\omega_1 t_0)\, z^{-1} + 1 z^{-2}$$

$$= h_1[0] + h[1] z^{-1} + h[2] z^{-2}$$

$$\left(\text{Recall } H(z) = \sum_{k=0}^{N} h[k] z^{-k} = h[0] + h[1] z^{-1} + \ldots + h[N] z^{-N}\right)$$

(assuming causality and finite)

(Notch Filter)



$H_1(z)$ will zero out inputs of the form $A\cos(\omega_1 \check{t} + \phi)$

Transfer Function zeros.

Let $H_i(z) = \left(1 - r_i e^{j\omega_i t_o} z^{-1}\right)\left(1 - r_i e^{-j\omega_i t_o} z^{-1}\right)$

$= 1 - 2 r_i \cos(\omega_i t_o) z^{-1} + r_i^2 z^{-2}$

$r_i < 1$ minimum-phase zero

$r_i = 1$ true zero (notch)

$r_i > 1$ non-minimum phase zero

To place more than one zero in a transfer function create

$$H(z) = \underbrace{H_1(z) H_2(z) H_3(z)}_{\text{multiply out polynomials}} \cdot \ldots$$

<u>Design</u> <u>Coded in MATLAB</u>

$H_1(z) = 1 - 2 r_1 \cos(\omega_1 t_o) z^{-1} + r_1^2 z^{-2} \longleftrightarrow h1 = [1 \;\; -2r_1\cos(\omega_1 t_o) \;\; r_1^2]$

$H_2(z) = 1 - 2 r_2 \cos(\omega_2 t_o) z^{-1} + r_2^2 z \longleftrightarrow h2 = [1 \;\; -2r_2\cos(\omega_2 t_o) \;\; r_2^2]$

Then, $H(z) = H_1(z) H_2(z) \longleftrightarrow$ h = conv(h1, h2);

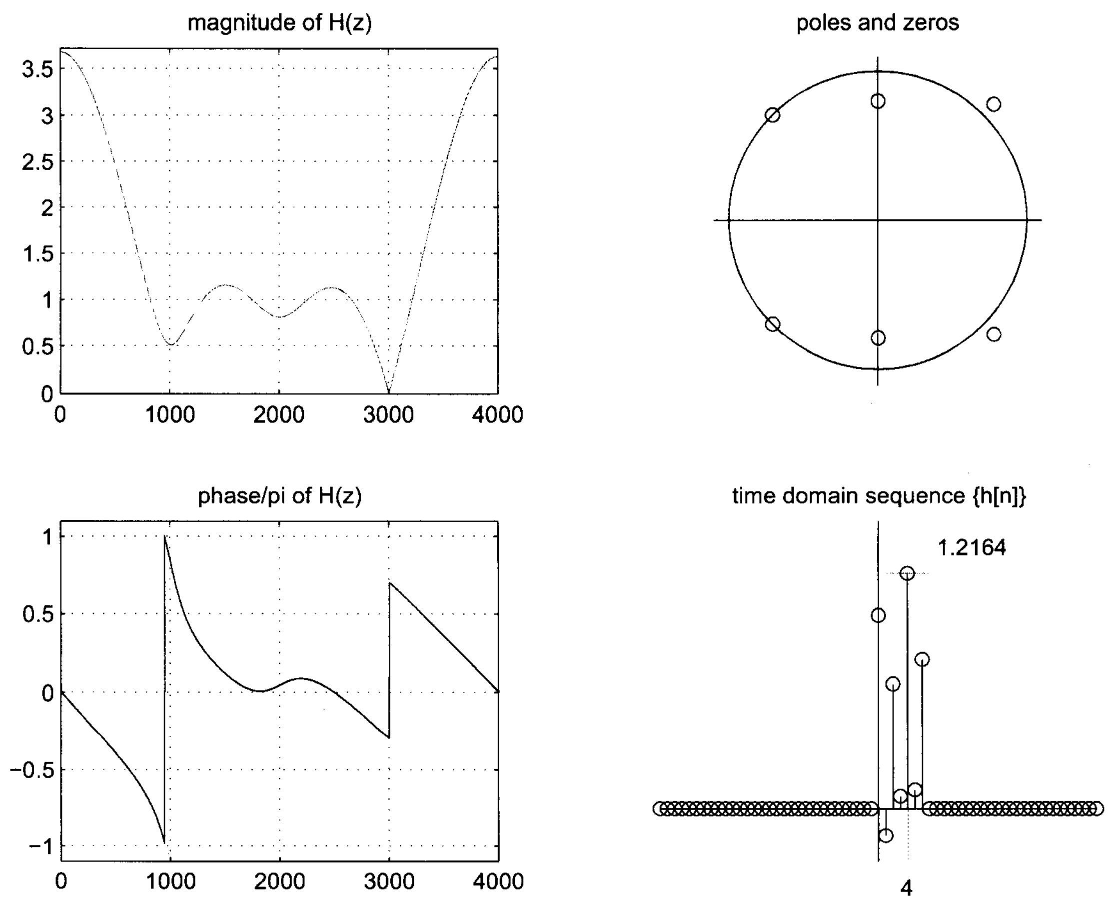
filter coefficients stored in the row vector h

(show design example)

NB: In $\cos(\omega_i t_o)$ $\omega_i t_o = 2\pi f_i t_o = 2\pi \frac{f_i}{f_o}$

```
% Zero-placing example coded in MATLAB
r1 = 1.1;  % non-minimum phase
r2 = .8;   % minimum phase
r3 = 1;    % notch

f1 = 1000;
f2 = 2000;
f3 = 3000;

f0=8000;

h1=[1 -2*r1*cos(2*pi*f1/f0) r1^2];
h2=[1 -2*r2*cos(2*pi*f2/f0) r2^2];
h3=[1 -2*r3*cos(2*pi*f3/f0) r3^2];

h = conv(h1,h2);
h = conv(h,h3);

plotZTP_FIR(h,8000)
```

- non-minimum phase zero at 1000Hz
- minimum phase zero at 2000Hz
- zero at 3000Hz
- $H(z) = (1 - 1.1e^{j\omega_1 t_0} z^{-1})(1 - 1.1e^{-j\omega_1 t_0} z^{-1})(1 - .8e^{j\omega_2 t_0} z^{-1})(1 - .8e^{-j\omega_2 t_0} z^{-1})(1 - e^{j\omega_3 t_0} z^{-1})(1 - e^{-j\omega_3 t_0} z^{-1})$
- $\omega_1 = 2\pi * 1000\text{rad/sec}$, $\omega_2 = 2\pi * 2000\text{rad/sec}$, $\omega_3 = 2\pi * 3000\text{rad/sec}$, $t_0 = 1/f_0$
- Sample Rate of Filter: $f_0 = 8\text{kHz}$



Linear Phase FIR Filters

$$\cos[\omega n t_0] \longrightarrow \boxed{H(e^{j\omega t_0})} \longrightarrow |H(e^{j\omega t_0})| \cos[\omega n t_0 + \arg\{H(e^{j\omega t_0})\}]$$

time delay of τ seconds

$$\cos[\omega(n t_0 - \tau)] = \cos[\omega n t_0 + \theta]$$

$$\Rightarrow \theta = -\omega\tau$$

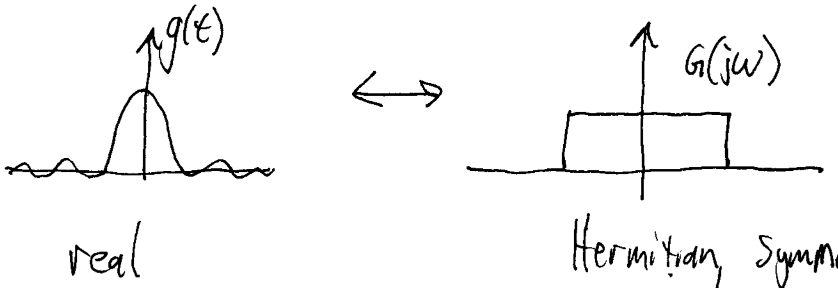
(The phasing of a cosine by θ will delay the signal $\tau = \frac{-\theta}{\omega}$ seconds in time)

$$\therefore \arg\{H(e^{j\omega t_0})\} = -\omega\tau$$

in order to ensure that all of the frequency content in the input is shifted by the same amount in time (i.e. there is no phase distortion in the filter)

How can we guarentee a linear phase?

Recall from last time

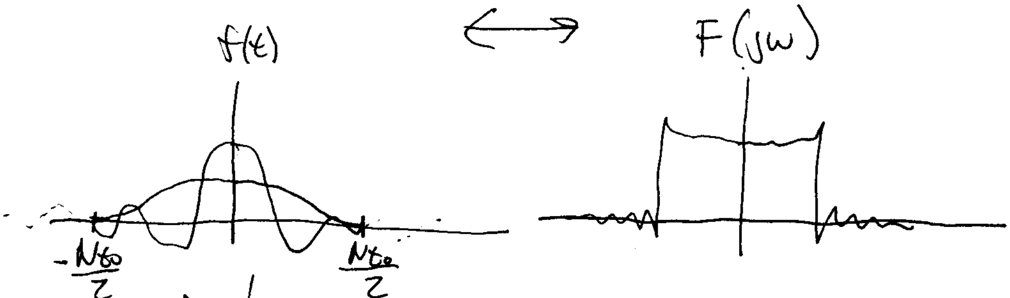


$\longleftrightarrow$

phase $= 0$

real & even

Hermitian Symmetric & real

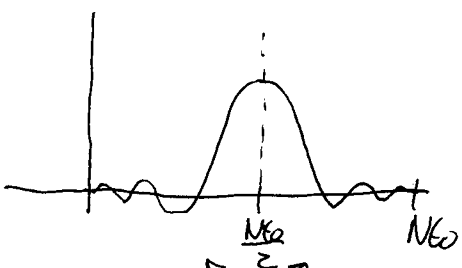


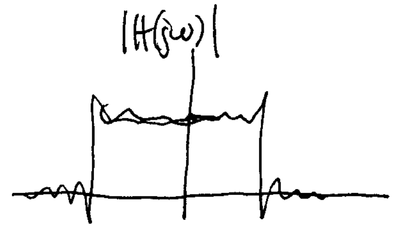
real & even

even & real

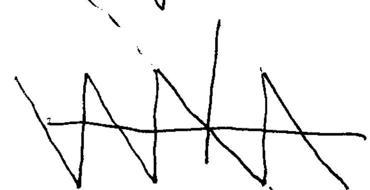
(zero-phase)

$$h(t) = f\left(t - \tfrac{N}{2}t_0\right) \longleftrightarrow H(j\omega) = e^{-j\frac{N}{2}t_0\omega}F(j\omega)$$





$\arg\{H(j\omega)\} = \frac{-Nt_0}{2}\omega$

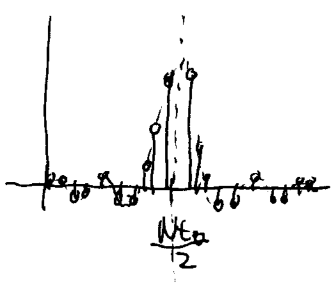


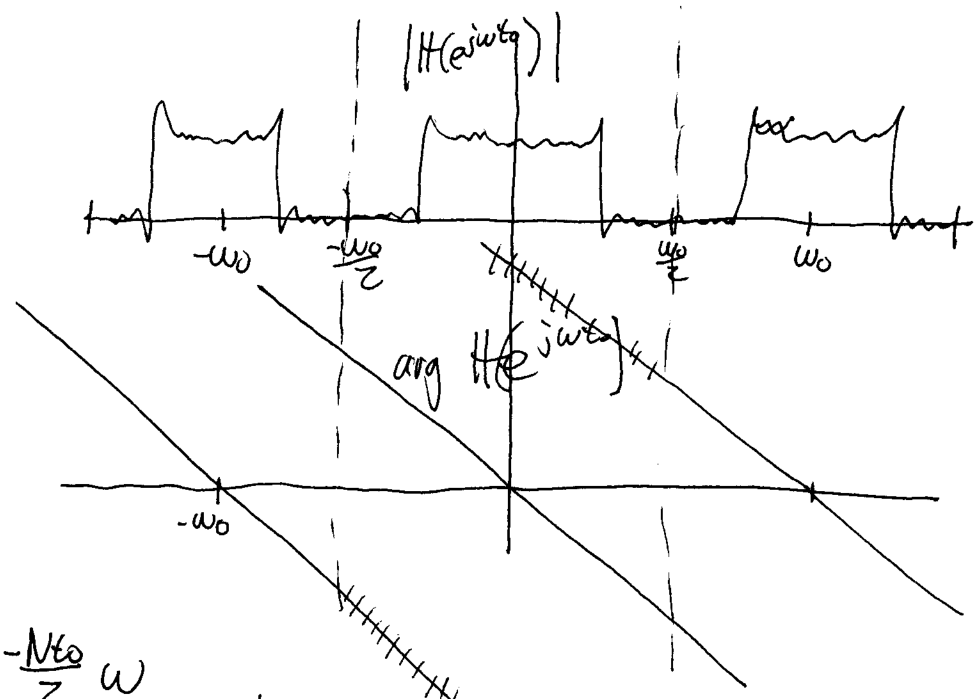
even symmetry here

linear phase here

$$h[n] = t_0 h(nt_0)$$

$$H(e^{j\omega t_0}) = \sum_{k=-\infty}^{\infty} H(j(\omega - k\omega_0))$$





$\therefore \arg\{H(e^{j\omega t_0})\} = \frac{-Nt_0}{2}\omega$

or delay $\tau = \frac{Nt_0}{2}$ seconds

- linear phase results from initial zero-phase constraint and time invariance

- All causal N-order FIR filters that have even symmetry about the sample point $N/2$ will have a (strict) linear phase.

← <u>Differentiators and Hilbert Transformers</u> have the phase constraint $\arg\{G(j\omega)\} = \pm \pi/2$

$$\underset{\substack{\text{(purely imaginary)} \\ e^{\pm j\pi/2} = \pm j}}{G(j\omega)} \longleftrightarrow \underset{\text{(odd symmetry)}}{g(t)}$$

- Same design process (with even windows) will preserve the phase constraint.

(i.e. $\arg\{H(e^{j\omega t_0})\} = \underbrace{\pm \frac{\pi}{2}}_{\substack{\text{phase} \\ \text{constraint}}} - \underbrace{\frac{N}{2} t_0 \omega}_{\substack{\text{linear phase} \\ \text{(required for causal implementation)}}}$

<u>Remark</u>: When $\arg\{H(e^{j\omega t_0})\} = -\alpha\omega + \beta$, the filter $H(e^{j\omega t_0})$ is said to have a generalized linear phase where

β : phase constraint $\left(\begin{array}{ll} \text{not a function of } \omega & \\ \beta = \beta & \omega > 0 \\ \beta = -\beta & \omega < 0 \end{array}\right)$ — conjugate symmetry required in $H(e^{j\omega t_0})$

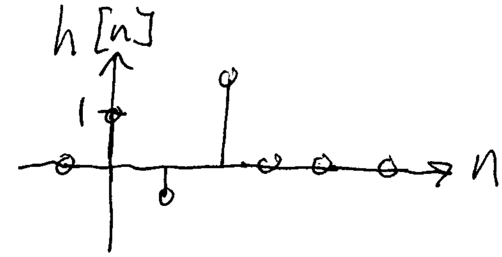
α : time delay associated with making it causal

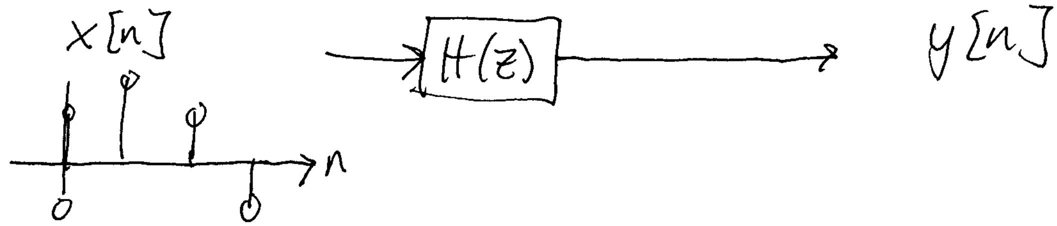
NB: When $\beta = 0$, $H(e^{j\omega t_0})$ is said to have a strict linear phase.

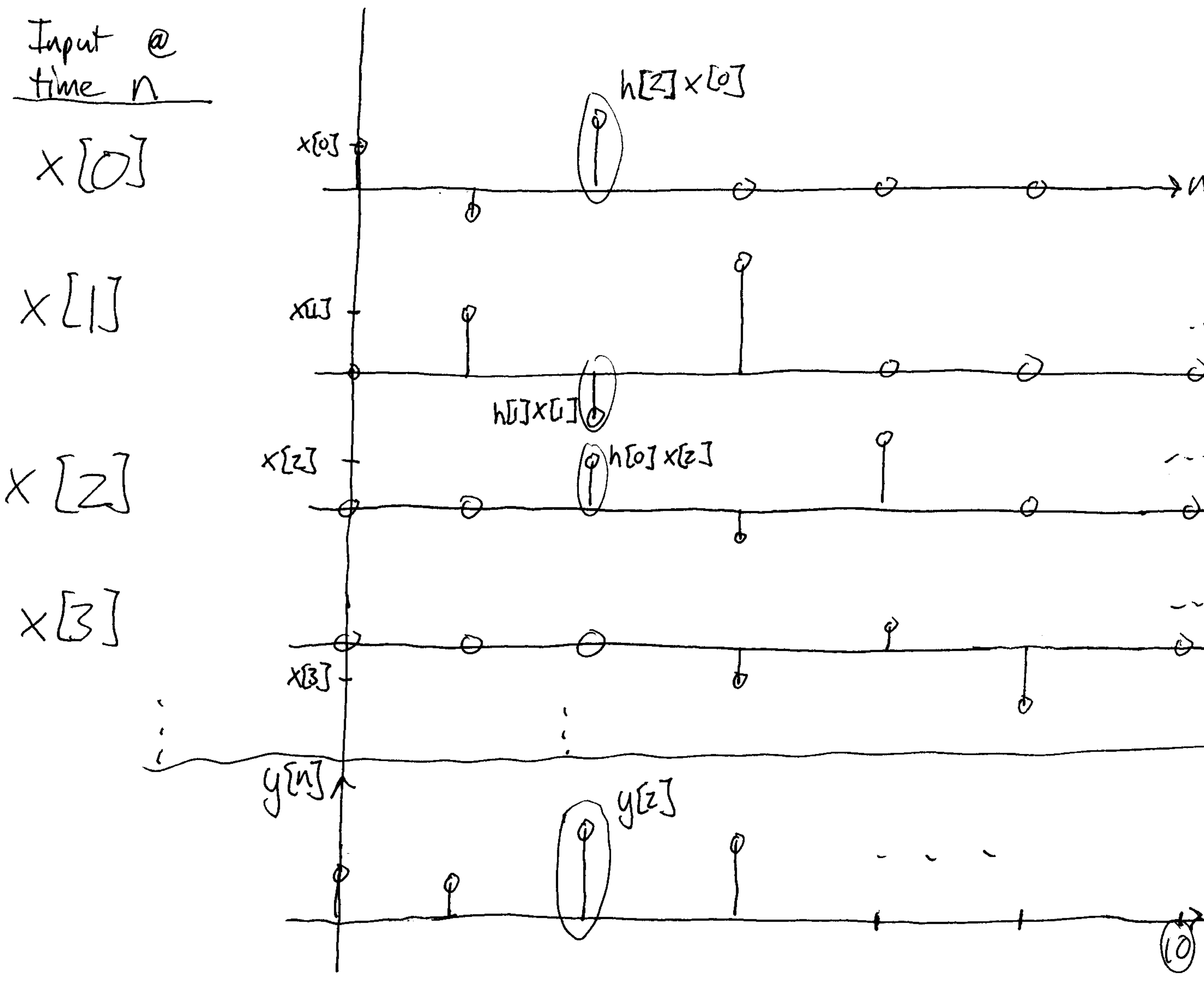
FIR Filter Implementation on the DSK

Action of a filter :

Let





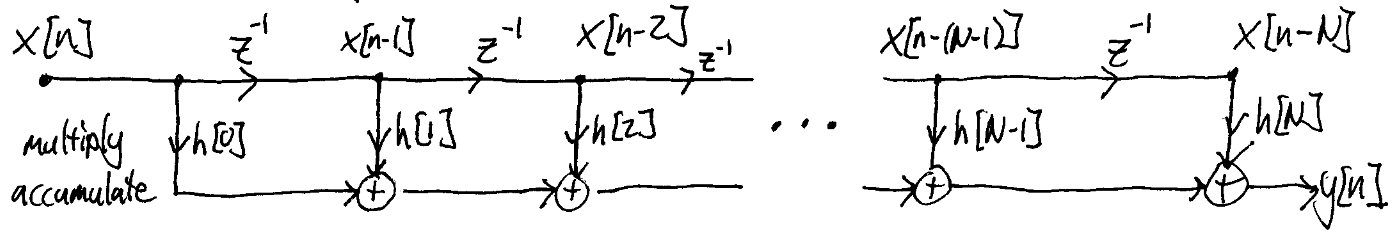


$$\therefore\quad y[2] = h[0]\,x[2] + h[1]\,x[1] + \underset{\substack{\uparrow \\ \text{Here } N=2}}{h[2]}\,x[0]$$

In general, $y[n] = \sum_{k=0}^{N} h[k]\,x[n-k]$

(convolution sum)

Hardware Diagram



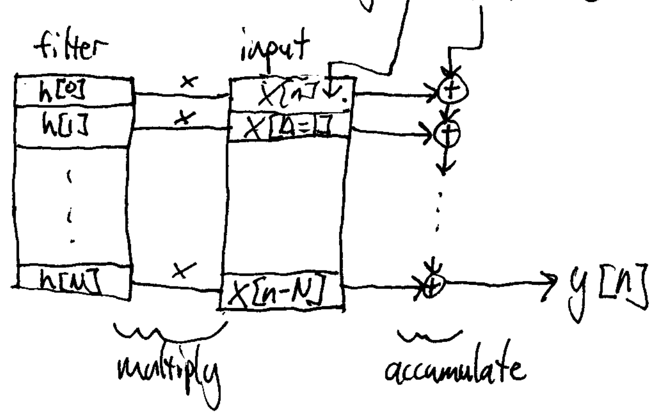
On The DSK

read in current time sample, x[n]

register on DSP Chip

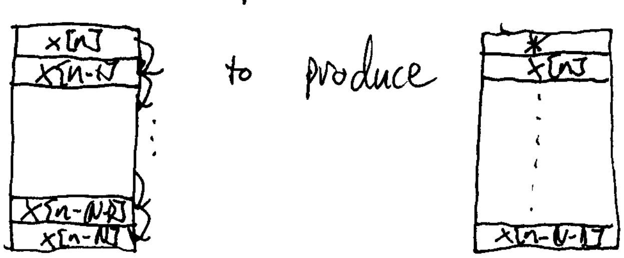
1.) Buffers in memory

Process



Update input buffer

2.)



At next time step (next interrupt), go to step 1, read in current time sample and repeat process