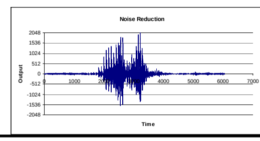
11. Use FFT to Design DF

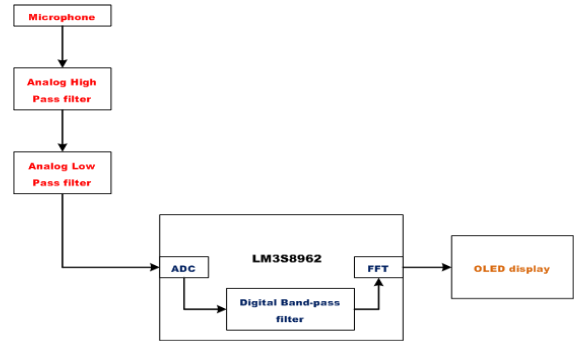
Lab 4 Application of RTOS

- **Input sound, analog filter**
- **Digital filter, FFT**
- **Display amplitude versus freq on the oLED**

Reference EE345M book, chapter 5



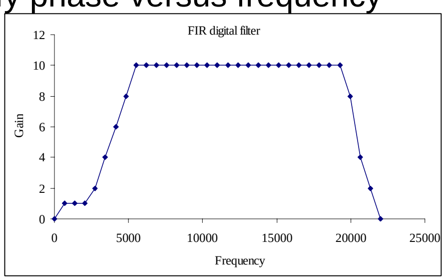
Lab4 Spectrum Analyzer



Design a linear FIR digital filter

- Specify gain versus frequency
- Specify phase versus frequency

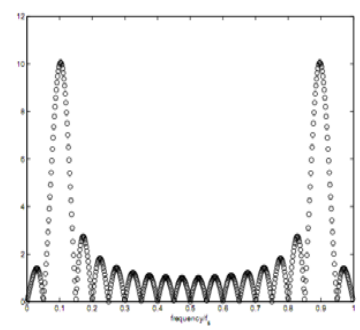
x(n)



y(n)

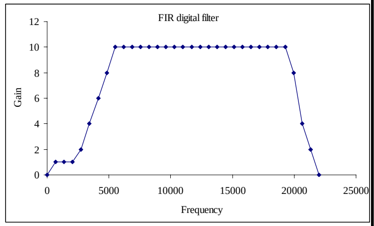
Fast Fourier Transform (FFT)

- FFT is a faster version of the Discrete Fourier Transform (DFT)
- FFT spectrum of a cosine, which has a frequency of 0.1 Fs
  - Fs = 10000 Hz
  - Cosine freq = 1000 Hz
- Region interested in from 0 to Fs/2 (symmetric about Fs/2)



FIR digital filter

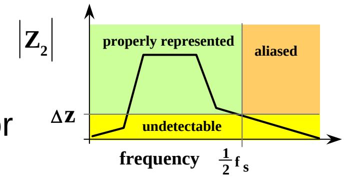
- $Y(z) = H(z) X(z)$
- $h(n) = IFFT \{H(z)\}$
- Convolution
  - $y(n) = h(n)*x(n)$



- Constants h_0, h_1, ... h_{N-1}
- $y(n) = h_0 \cdot x(n)+h_1 \cdot x(n-1)+ \ldots +h_{N-1} \cdot x(n-(N-1))$
- N multiplies, N-1 additions per sample

How to choose sampling rate

- Nyquist Rate
- Limitation of display
- Limitation of processor
- Limitation of RAM
- Limitation of human eyes and ears
- Limitation of communication channel



How to choose number of samples

- Frequency resolution = f_s/N
- Increase in N results in better frequency resolution
- However, increase in N leads to a bigger MACQ buffer and more multiplies and additions
- Does not need to be a power of 2
  - DFT calculated once, off line

Design process (1)

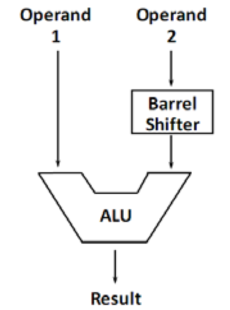
- Specify desired gain and phase, 0 to ½ f_s
  - k goes from 0 to N/2 ($f = k f_s/N$)
  - H(k) is complex
  - |H(k)| is gain
  - angle(H(k)) is phase
- For ½ f_s to f_s
  - H(N-k) is complex conjugate of H(k)
  - Poles and zeros are in complex conjugate pairs

Design process (2)

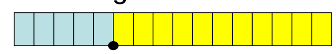
- Take IDFT of H(k) to yield h(n)
  - n goes from 0 to N-1
  - h(n) will be real, because H(k) symmetric
- The digital filter is
  - $y(n) = h_0 \cdot x(n) + h_1 \cdot x(n-1) + \ldots + h_{N-1} \cdot x(n-(N-1))$

Binary Fixed point notation

- Binary fixed-point is faster than decimal fixed-point
- Qn Number (16 bit)
  - n: specifies the resolution = 2^{-n}
  - 16-n: specifies Range
- Eg: 10.450 (unsigned number) with n=11?
- How is this number stored as an integer?



EE382N: Advanced Embedded Systems Architecture (lecture 5)



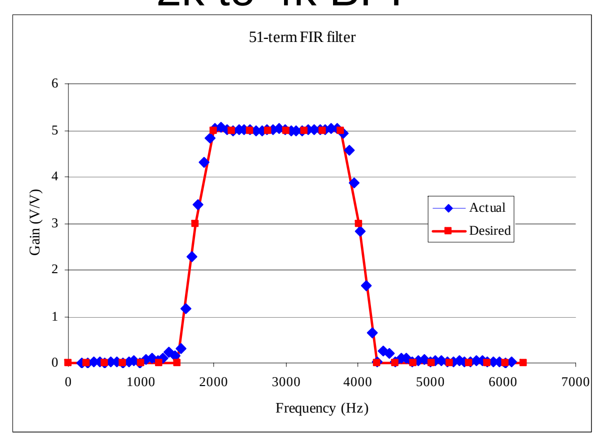
Value = Integer/2048

10.450 ≈ 01010.1110011010

Example

- Open FIRdesign51.xls
- Change sampling rate to 10,000 Hz
- Adjust red desired gain to make BPF
  - Pass 2 to 4 kHz
  - Look at sharp corner versus round corner
- Notice linear phase
- Copy 51 coefficients into software

2k to 4k BPF



FIR Filter SW design

```
const long h[51]={-3,-9,4,5,0,17,5,-20,-5,-7,-22,
   24,41,-8,2,1,-74,-31,71,20,33,125,-119,-350,67,
   462,67,-350,-119,125,33,20,71,-31,-74,1,2,-8,41,
   24,-22,-7,-5,-20,5,17,0,5,4,-9,-3};
static unsigned int n=50;   // 51,52,… 101
short Filter(short data){unsigned int k;
 static long x[102]; // this MACQ needs twice
 long y;
 n++;
 if(n==102) n=51;
 x[n] = x[n-51] = data;  // two copies of new data
 y = 0;
 for(k=0;k<51;k++){
   y = y + h[k]*x[n-k];  // convolution
 }
 y = y/256;  // fixed point
 return y;
}
```


Circular Buffering

Array Index	Filter Coefficient Array h[]	Circular Buffer Array xcirc[]
0	$h[0]$	$x[n - newest]$
1	$h[1]$	$x[n - newest + 1]$
$\vdots$	$\vdots$	$\vdots$
		$x[n-1]$
newest		$x[n]$
oldest		$x[n-N+1]$
		$x[n-N+2]$
$\vdots$	$\vdots$	$\vdots$
$N-2$	$h[N-2]$	$x[n - newest - 2]$
$N-1$	$h[N-1]$	$x[n - newest - 1]$

Source: "Communication system design using DSP algorithms" by Steven A. Tretter (chapter 3, page 73)

Optimization

- Pointer implementations of MACQ faster
- Do not try and shift the data
- Convolution x[n]*h[n] takes N multiplies, N-1 additions per sample
  - Can be optimized to N/2 multiplies
  - Coefficients are symmetric
- Assembly optimization with MLA
  - Multiply with accumulate