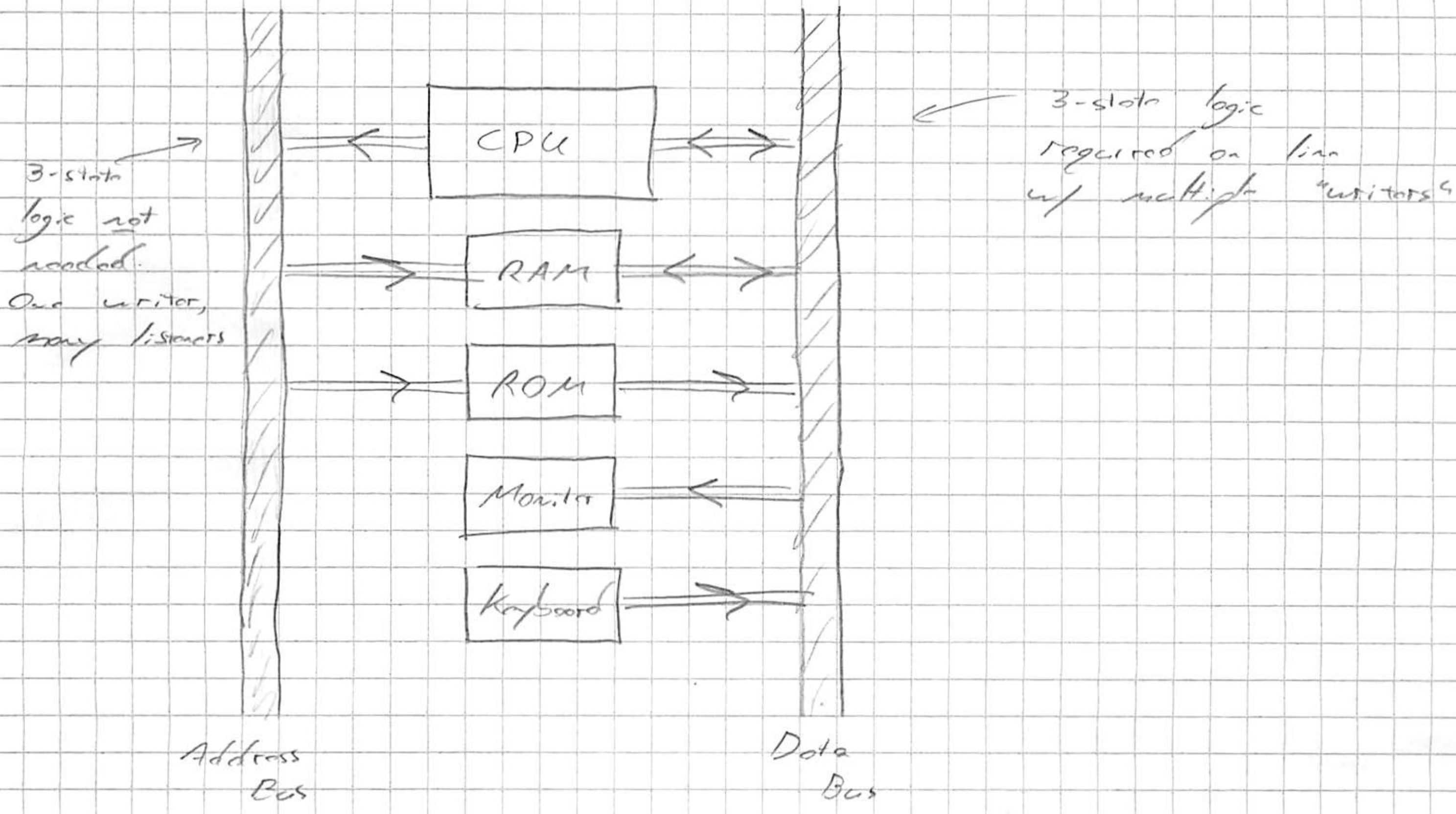
| Class # 11 |

Digital III & Digital meets Analog

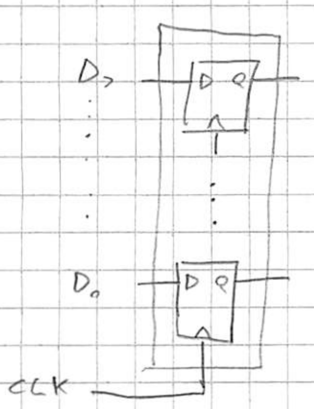
Computer Organization:



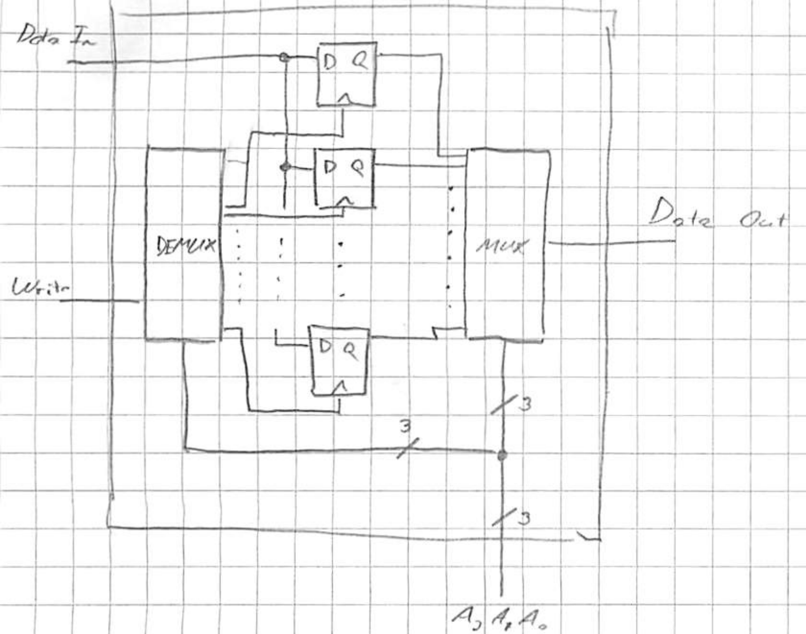
Memory

Array of flip-flops (or other device which can hold bits of information).

8-bit memory : register (collection of flip-flops)



	Pins
data in	8
data out	8
clock/write	1
power/gnd	2
	19



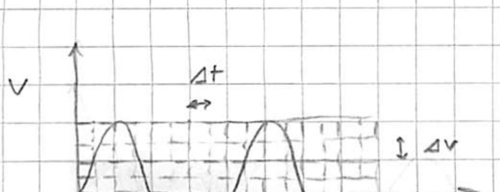
	Pins
data in	1
data out	1
clock/write	1
power/gnd	2
address	3
	8

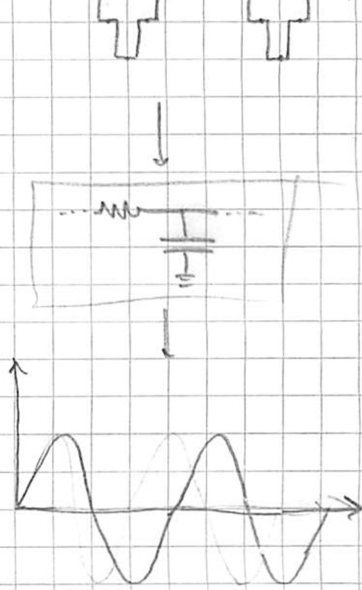
Fewer pins ... as memory becomes larger & larger the 2-d architecture becomes more and more favorable.

D-to-A & A-to-D

DAC, D/A   ADC, A/D

D-to-A (Digital to Analog Conversion)





ΔV → voltage slice

V ⟷ binary #

8-bit encoding : $2^8 = 256$ slices

$$\Delta V = \frac{V_{Full scale}}{256}$$

$\frac{\Delta V}{2}$ = voltage resolution

How many bits required for 0.01 % resolution?

$0.01\% = \frac{1}{10000}$ → For 5 V full scale,

→ 0.5 mV

4.9115 → 4.912 V

4.9105 → 4.911 V ↕ $\Delta V = 1\,mV$ ⇝ corresponds to

4.910 V

5000 slices ≈ 2^{12}

∴ 12-bit encoding

Δt → sampling period

$1/\Delta t$ → sampling rate } Just over 2 samples per period are required to describe a sine wave ← Nyquist Theorem

f_s

⇒ If $x(t)$ contains no frequencies higher than B, it can be completely determined by measuring its value a a series of points spaced by $\Delta t = \frac{1}{2B}$.

$\to f_s \geq 2B$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)\, e^{i\omega t}\, d\omega$$

No freqs higher than B:

$$f(t) = \frac{1}{2\pi} \int_{-2\pi B}^{2\pi B} F(\omega)\, e^{i\omega t}\, d\omega$$

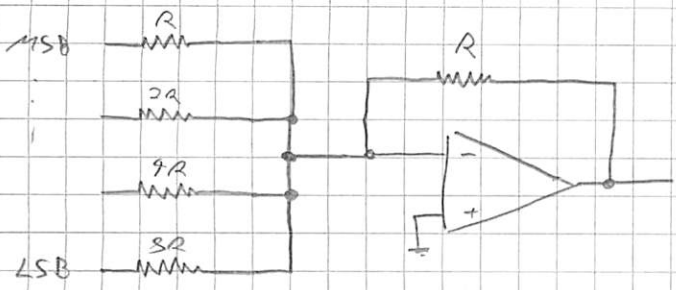
$$F(\omega) = \sum_{-\infty}^{\infty} c_n e^{in\omega}$$

$$c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} F(\omega)\, e^{in\omega} d\omega$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)\, e^{i\omega t}\, d\omega$$

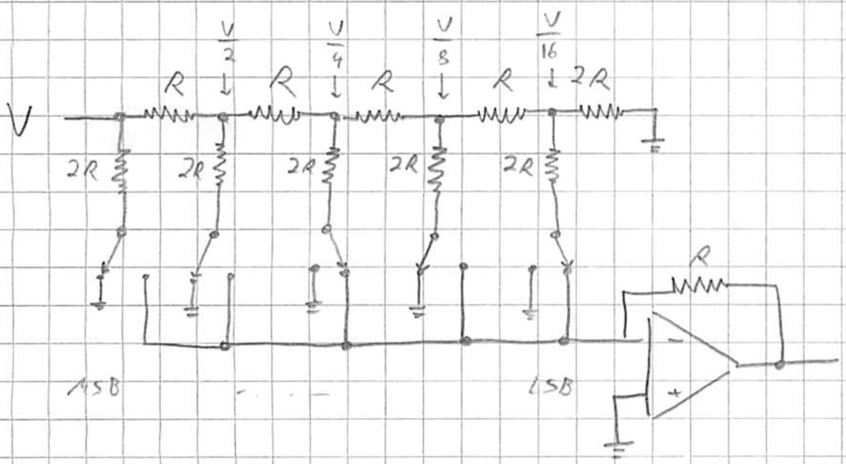
$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \sum_{-\infty}^{\infty} c_n e^{in\omega}\, e^{i\omega t}\, d\omega$$

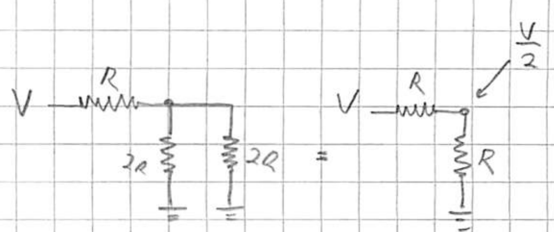
D-to-A



This method becomes intractable for large bit encoding... Resistors w/ ratios of 2000:1 for a 12-bit converter...

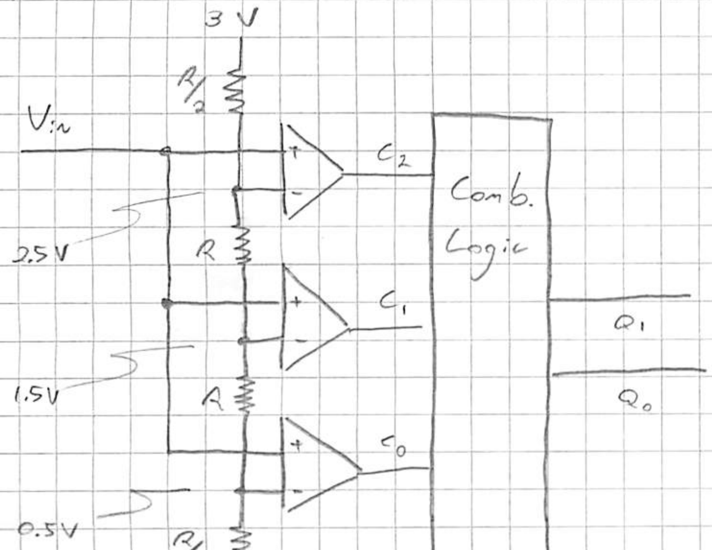
Solution: R-2R Ladder





A-to-D

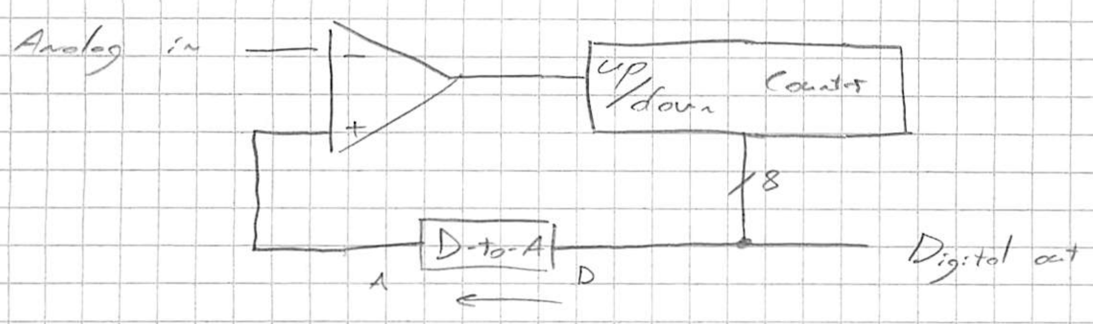
Parallel Encoder (Flash encoder)

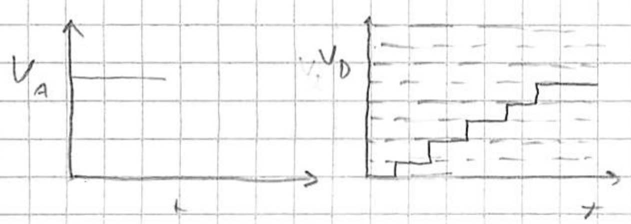


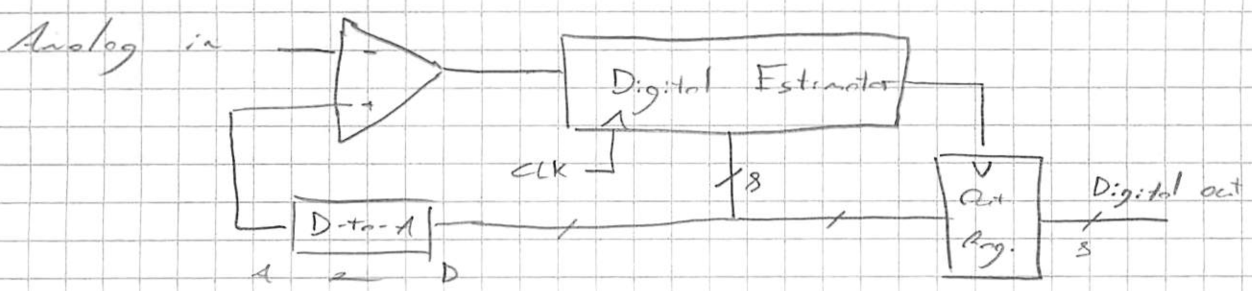
Thresholds	Vin	Comp. Output C_2	C_1	C_0	Code Q_1	Q_0
3V						
	2.8V	1	1	1	1	1
2.5V						
	2.0V	0	1	1	1	0
1.5V						
	1.0V	0	0	1	0	1
0.5V						
	0.25V	0	0	0	0	0
0V						

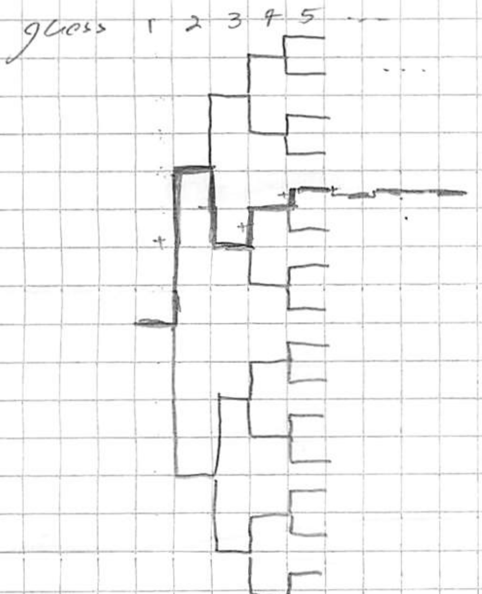
$2^2 - 1$ $\quad$ $2^9 - 1 = 511$

Closed-loop : Successive-approximation & tracking









Guess #	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
1	0	1	1	1	1	1	1	1	+
2	1	0	1	1	1	1	1	1	−
3	1	0	0	1	1	1	1	1	+
4	1	0	1	0	1	1	1	1	+
5	1	0	1	1	0	1	1	1	−
6	1	0	1	1	0	0	1	1	+
7	1	0	1	1	0	1	0	1	−
8	1	0	1	1	0	1	0	0	

Fourier Transform

$$f(t) = \int_{-\infty}^{\infty} \hat{f}(\nu)\, e^{2\pi i \nu t}\, d\nu$$

$$\hat{f}(\nu) = \int_{-\infty}^{\infty} f(t)\, e^{-2\pi i t \nu}\, dt$$

But we sample the continuous signal $f(t)$ at discrete time intervals : $0, T, 2T, \ldots$ w/ $T = 1/\nu_{sampling}$ and N is the number of samples. A discrete Fourier transform can then be written :

$$\underbrace{\hat{f}(\nu)}_{\text{cont. FT}} = \lim_{\substack{T\to 0\\ N\to\infty}} \underbrace{\sum_{n=0}^{N-1} f(nT)\, e^{-2\pi i n T \nu} \cdot T}_{\text{disc. FT} \to F(\nu)}$$

$$F(\nu) = T \sum_{n=0}^{N-1} f(nT)\, e^{-2\pi i n T \nu}$$

$$F(\nu_{Nyq} - \nu') = T \sum_{n=0}^{N-1} f(nT) \cdot e^{-2\pi i n T(\nu_{Nyq} - \nu')}$$

w/ $\nu_{Nyq} = \nu_{sampling}/2 = \frac{1}{2T}$

$$F(\nu_{Nyq} - \nu') = T \sum_{n=0}^{N-1} f(nT)\, \underbrace{e^{-\pi i n}\, e^{2\pi i n T \nu'}}$$

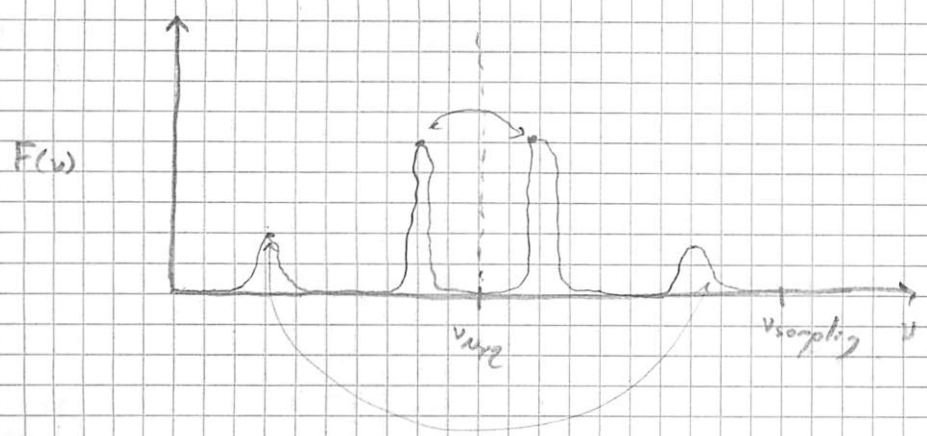
$$e^{i\pi n}\, e^{i2\pi \nu' n T} = e^{i2\pi\left(\frac{1}{2T} + \nu'\right) n T} = e^{i2\pi(\nu_{Nyq} + \nu') n T}$$

$$F(\nu_{Nyq} - \nu') = T \sum_{n=0}^{N-1} f(nT) \cdot e^{2\pi i n T(\nu_{Nyq} + \nu')}$$

$$F(\nu_{Nyq} - \nu') = \left[T \sum_{n=0}^{N-1} f(nT)\, e^{-2\pi i n T(\nu_{Nyq} + \nu')} \right]^*$$

$$\therefore \boxed{F(\nu_{Nyq} - \nu') = F(\nu_{Nyq} + \nu')}$$

The previous relation results in so-called frequency "folding."



Actually the problem is much worse!

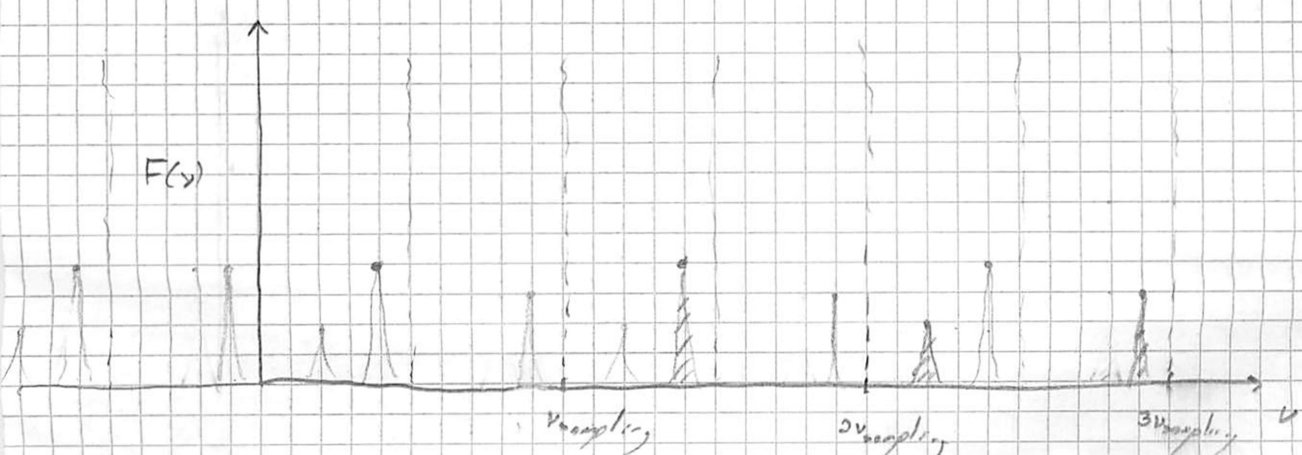
$$F(\nu) = T \sum_{n=0}^{N-1} f(nT)\, e^{-2\pi i n T \nu}$$

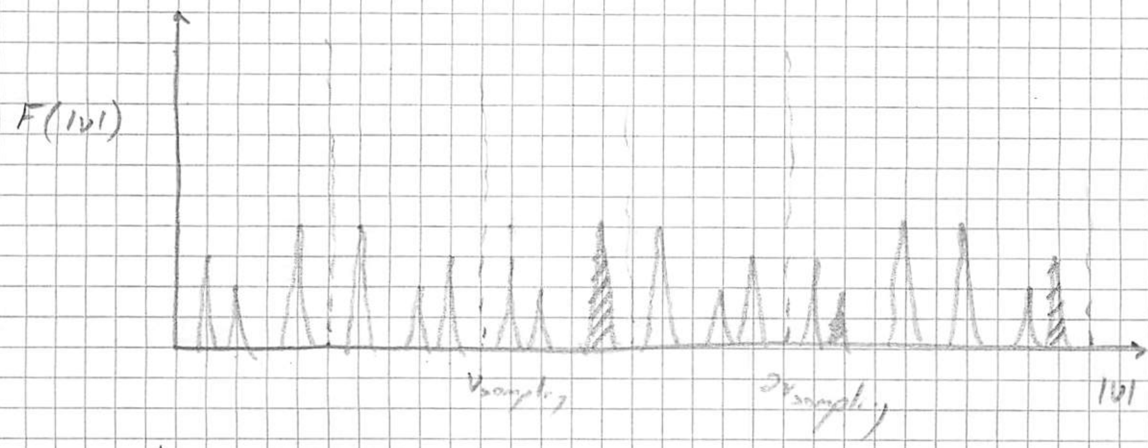
$$F(\nu + m\nu_{sampling}) = T \sum_{n=0}^{N-1} f(nT)\, \underbrace{e^{-2\pi i n T(\nu + m\nu_{sampling})}}_{e^{-2\pi i n T \nu}\, \underbrace{e^{-2\pi i n m}}_{1}}$$

w/ $m = 0, \pm 1, \pm 2, \ldots$

$$F(\nu + m\nu_{sampling}) = T \sum_{n=0}^{N-1} f(nT)\, e^{-2\pi i n T \nu}$$

$$F(\nu + m\nu_{sampling}) = F(\nu)$$





What concerns us...

Solution : Filtering for $\nu \geq \nu_{sampling}/2 = \nu_{Nyq}$