Software Defined Radio
PhD Program on Electrical Engineering

Multirate Systems and CIC Filters

José Vieira

Summary

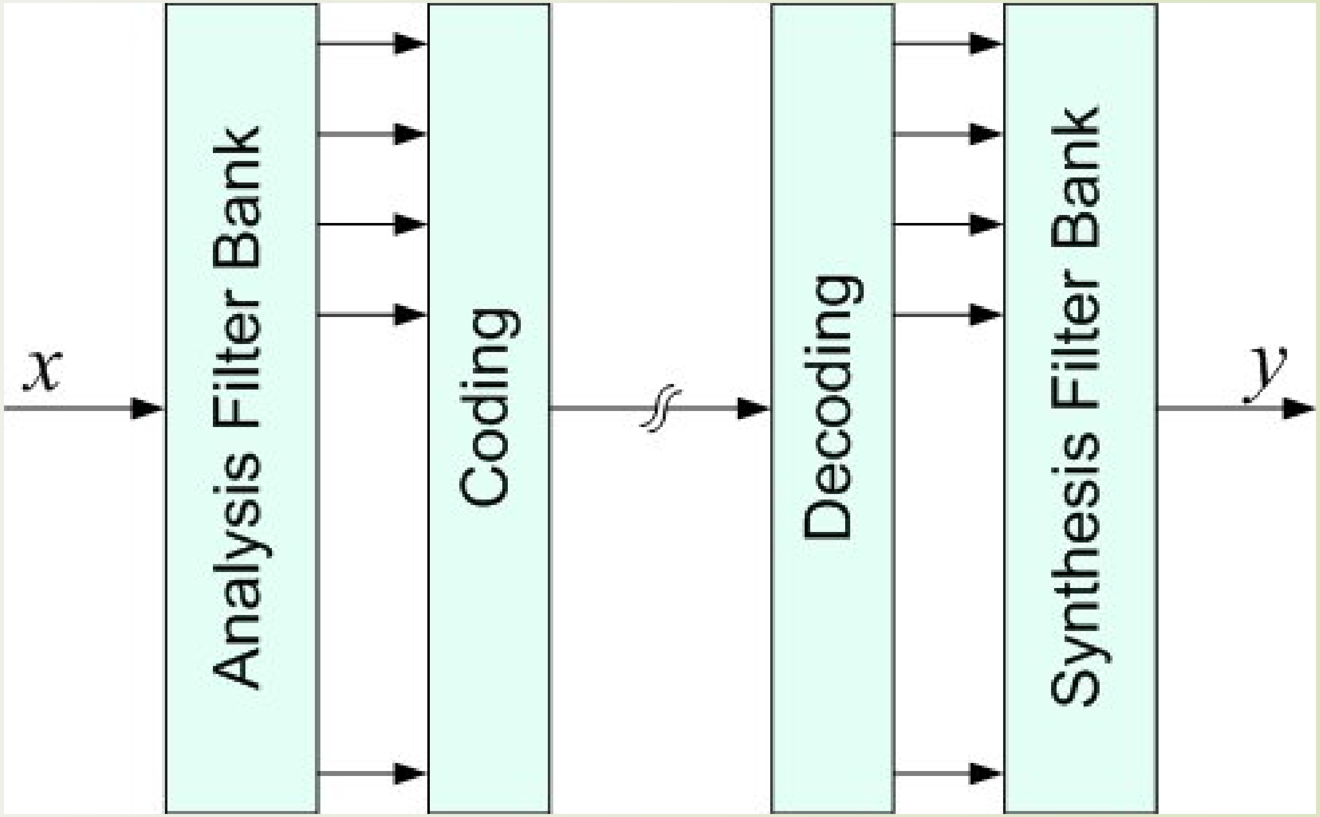
- Interpolation
- Decimation
- Time and Frequency Multiplexing
- Changing the Sampling Rate by a Rational Factor
- The Noble Identities
- The Polyphase Decomposition
- Filter banks
  - The two channels case
  - Alias cancelation
  - Perfect reconstruction
- Tree filter banks
- Time frequency decomposition

Bibliography

- Sebenta de Processamento Digital de Sinal do Prof. Tomás Oliveira e Silva
- P. P. Vaidyanathan, “Multirate Systems and Filter Banks”, Prentice Hall, 1993.

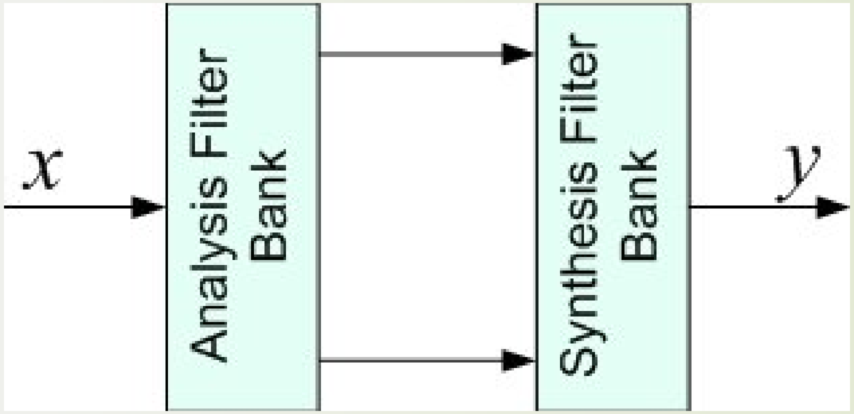
- Recommended reading from this book
  - Chapter 4
  - Sections 5.0, 5.1, 5.2, 5.3, 5.3.1, 5.3.6, 5.4, 5.5, 5.8, 5.9
- Optional reading
  - Sections 5.6 and 5.7

Filter Banks



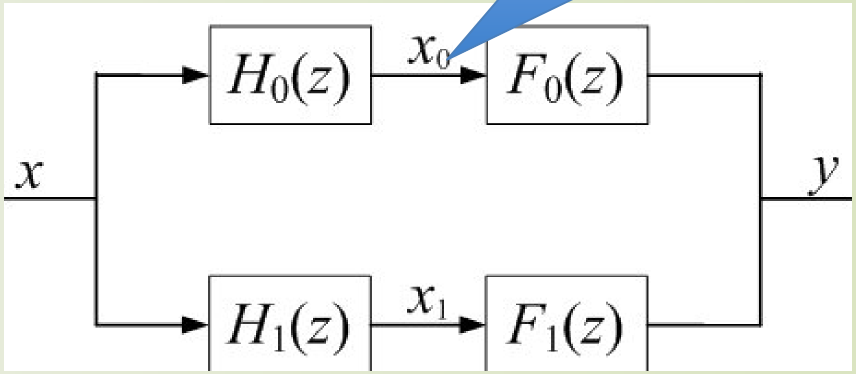
The analysis filter bank splits the input signal into several frequency bands
The synthesis filter bank combines the bands into one signal
The coding or other processing is performed on each band

The Two Channel Case



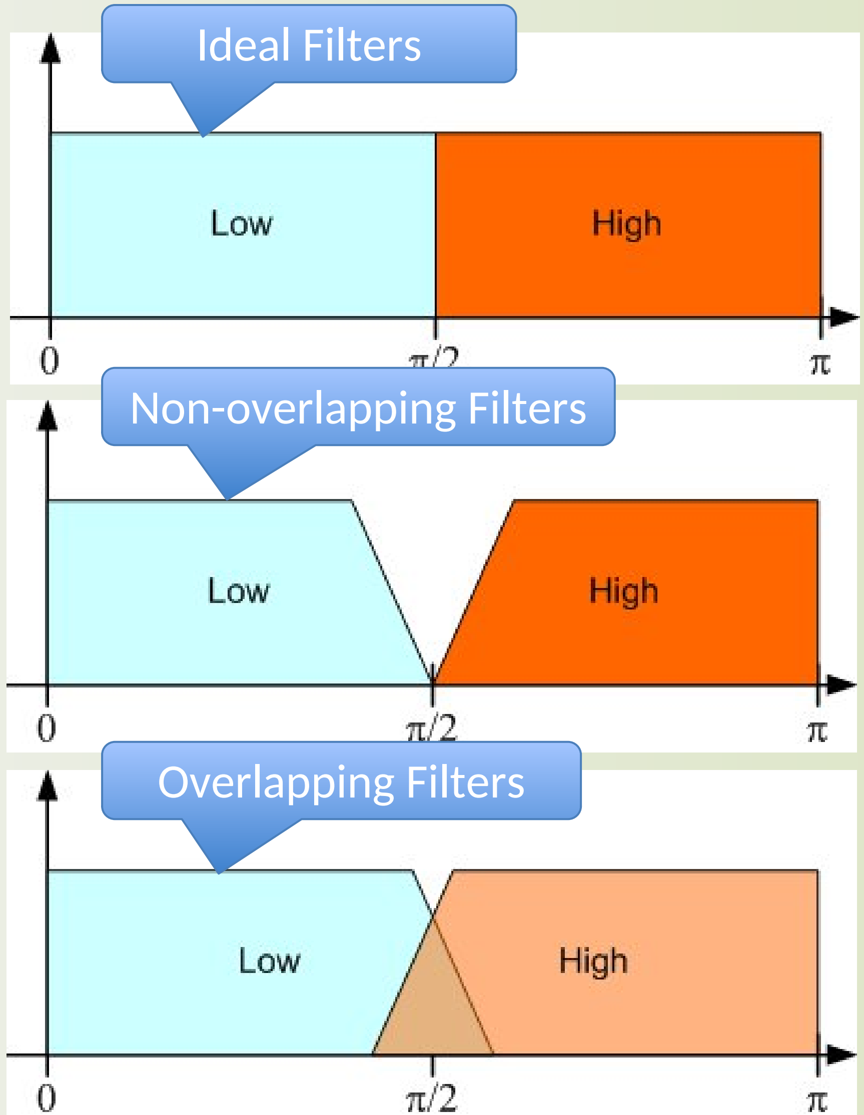
The input signal is split into two frequency components, low and high.

The signals x_0 and x_1 only have half the bandwidth of the input signal *x*.

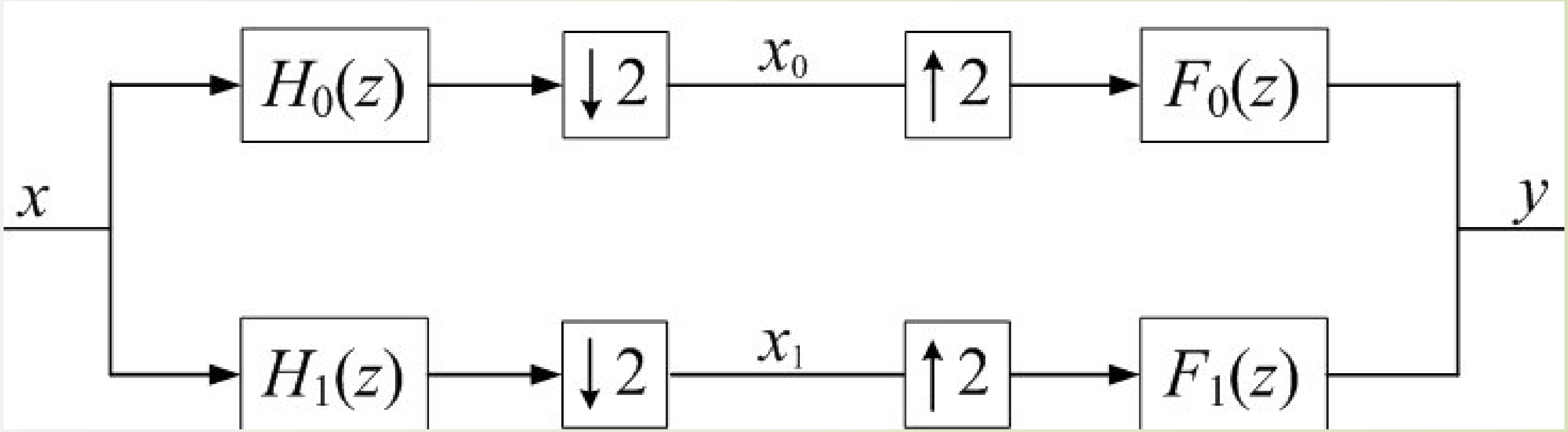


The analysis has a low pass-filter $H_0(z)$ and a high-pass filter $H_1(z)$. That way $F_0(z)$ is also low-pass and $F_1(z)$ high-pass.

Designing the Filters



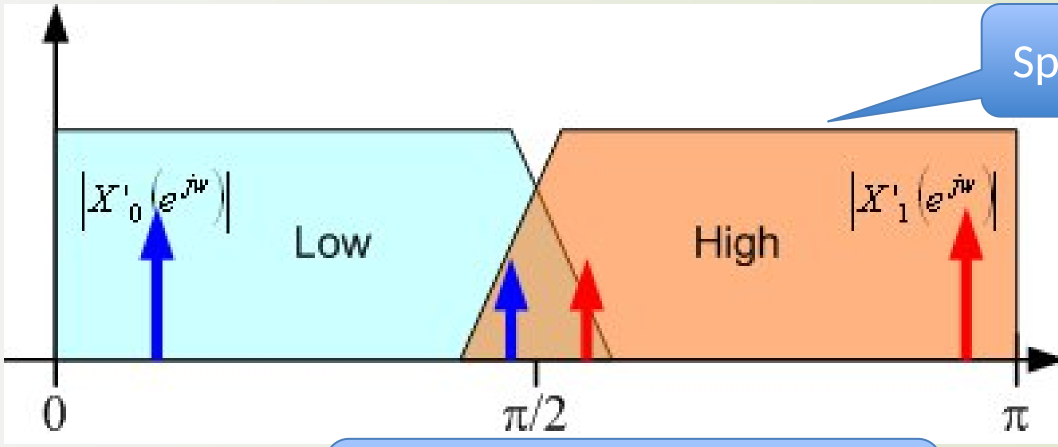
Decimated Two-Channel Filter Bank



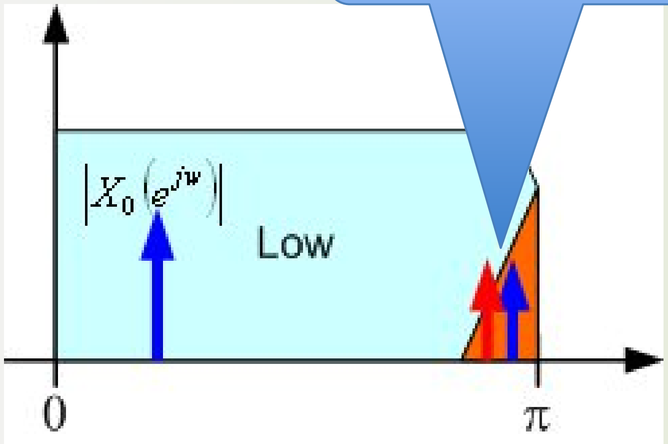
- The sampling rate of the signals x_0 and x_1 are reduced by a factor of two
- The bandwidth of the signals x_0 and x_1 is also reduced by a factor of two.
- Due to this, it is not possible to avoid aliasing on the the signals x_0 and x_1

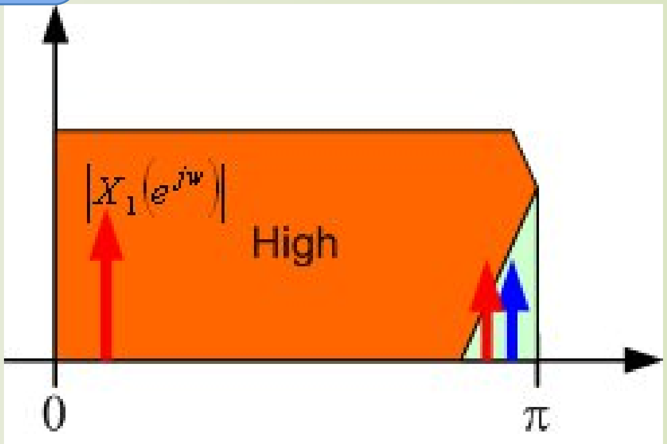
Aliasing in a Two-Channel Filter Bank

Spectrum before the decimators



Aliasing component after the decimators





Canceling the Aliasing

- The Z transform of the output signal can be written as a function of $X(z)$ and $X(-z)$

$$Y(z) = \frac{1}{2}\left[H_0(z)F_0(z) + H_1(z)F_1(z)\right]X(z) + \frac{1}{2}\left[H_0(-z)F_0(z) + H_1(-z)F_1(z)\right]X(-z)$$

- To cancel the aliasing component we should have

$$H_0(-z)F_0(z) + H_1(-z)F_1(z) = 0$$

- This can be obtained by doing

$$F_0(z) = H_1(-z) \quad \text{and} \quad F_1(z) = -H_0(-z)$$

The Transfer Function and Perfect Reconstruction

- When the alias is canceled, the filter bank has a transfer function given by

$$\frac{Y(z)}{X(z)} = \frac{1}{2}\left[H_0(z)H_1(-z) - H_1(z)H_0(-z)\right] = \frac{1}{2}\left[P(Z) - P(-z)\right]$$

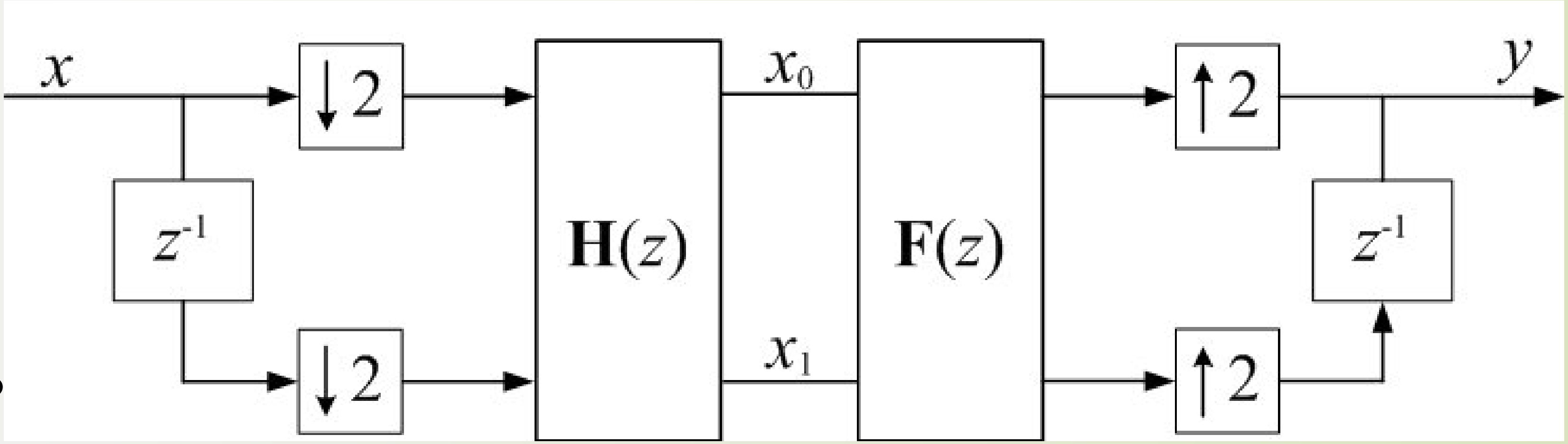
- If this transfer function can be put in the form

$$\frac{Y(z)}{X(z)} = Kz^{-\Delta}$$

- Then we say that the filter bank has

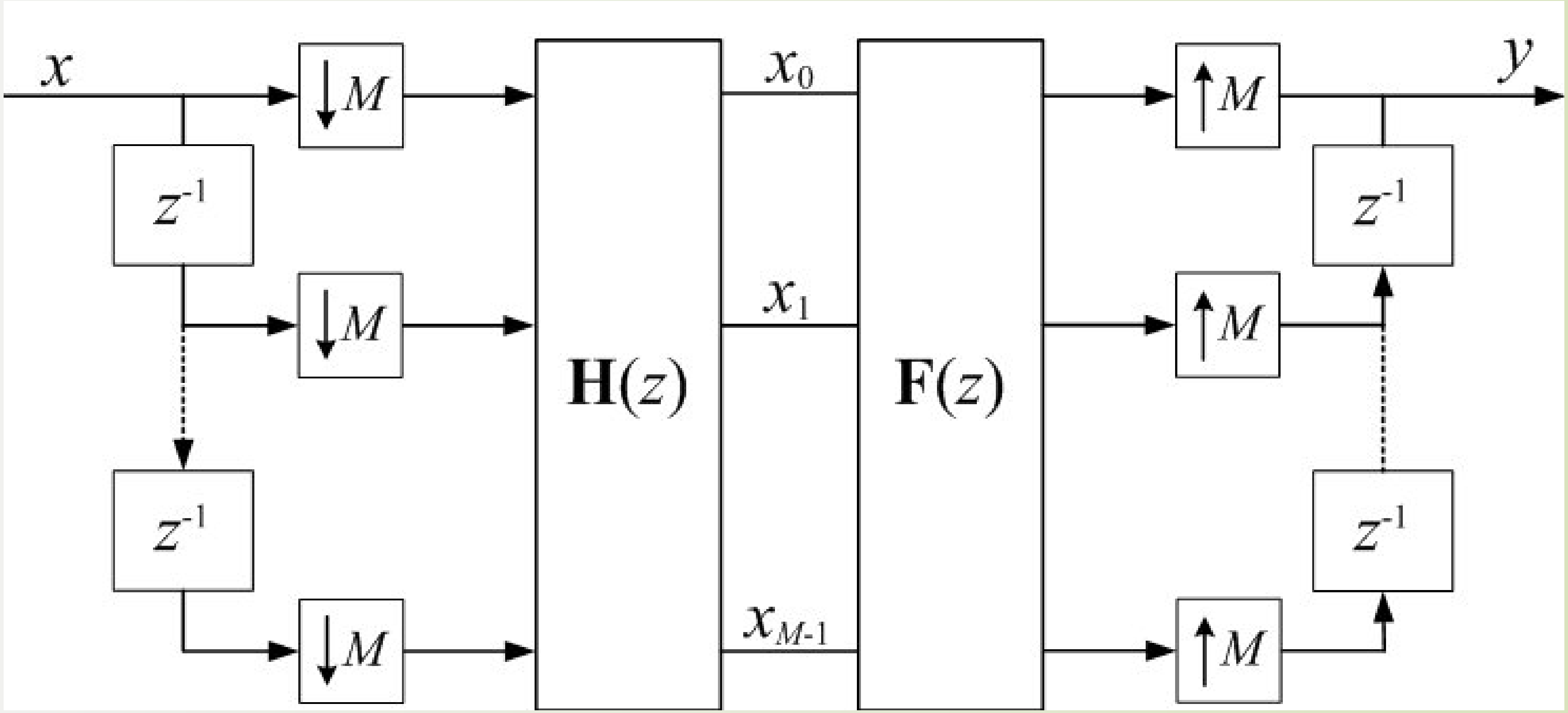
Perfect Reconstruction

Polyphase Decomposition of a Filter Bank – Two channel case

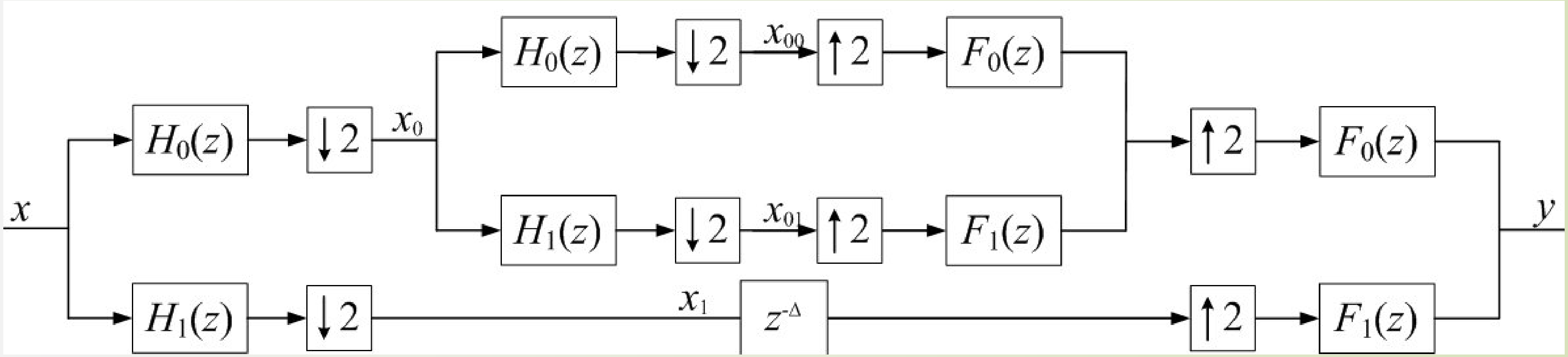
- If the filters of the two channel filter bank are represented using the polyphase decomposition we can move the decimators and expanders and get the following system
- 

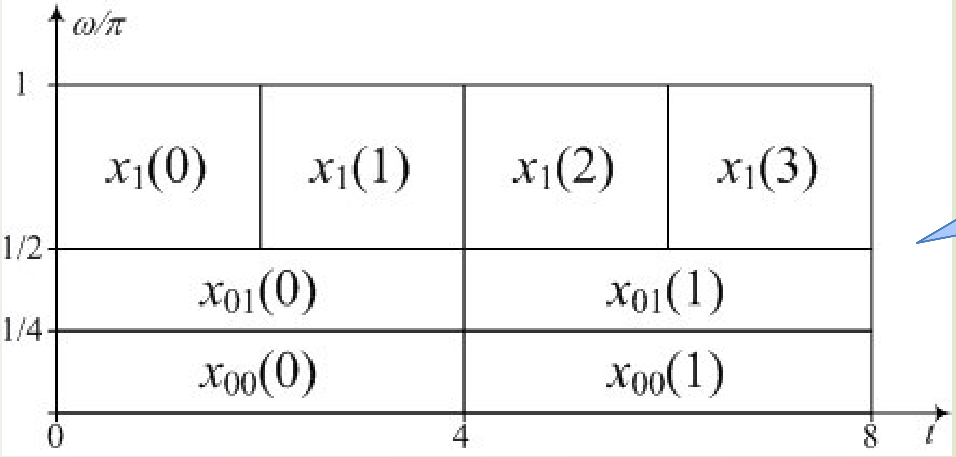
$$\mathbf{F}(z)\mathbf{H}(z) = Kz^{-\Delta}\mathbf{I}$$

Polyphase Decomposition of a Filter Bank – The General Case



Tree Filter Banks





The time frequency decomposition

Cascaded Integrator-Comb (CIC) Filters

- The CIC filters are an efficient implementation of “Moving Average Filters”.
- A “Moving Average Filter” has the following impulse response

$$y(n) = \frac{1}{D}\left[x(n) + x(n-1) + \ldots + x(n-D+1)\right]$$

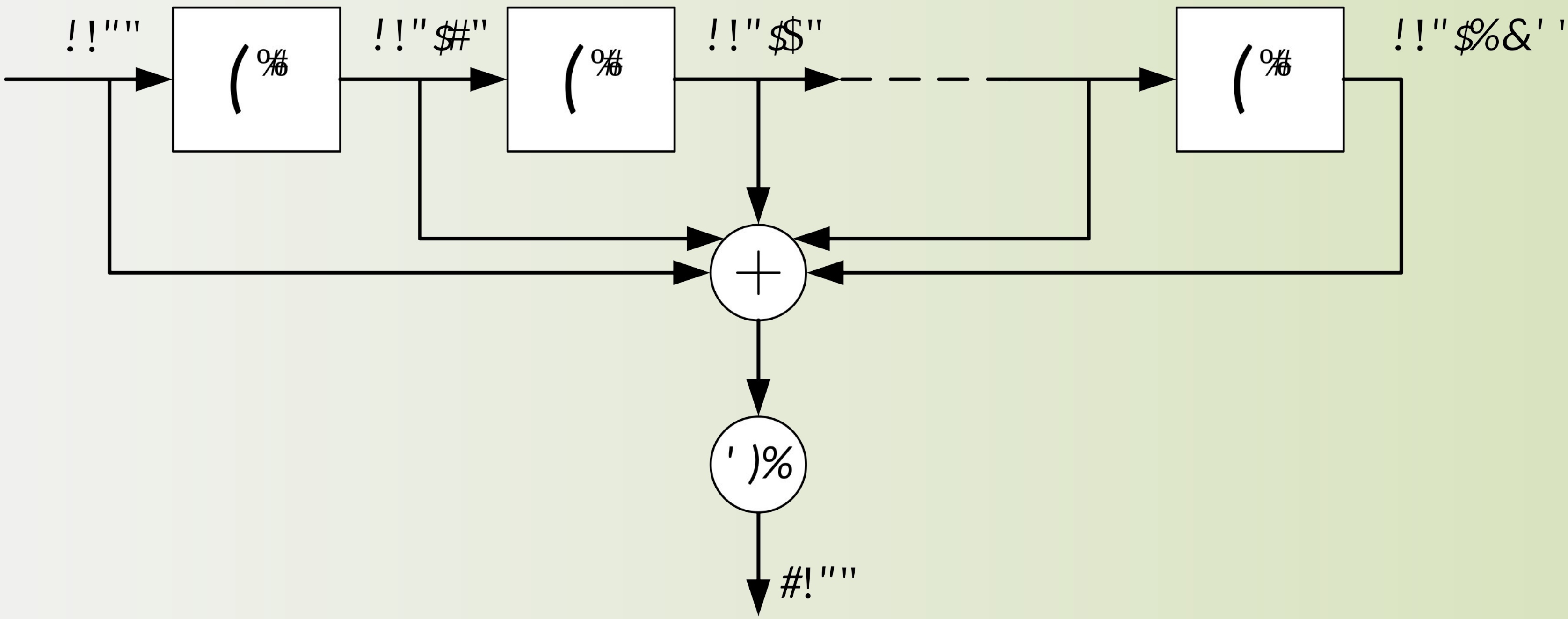
- and the *Z* transform

$$Y(z) = \frac{1}{D}\left[X(z) + z^{-1}X(z) + \ldots + z^{-D+1}X(z)\right]$$

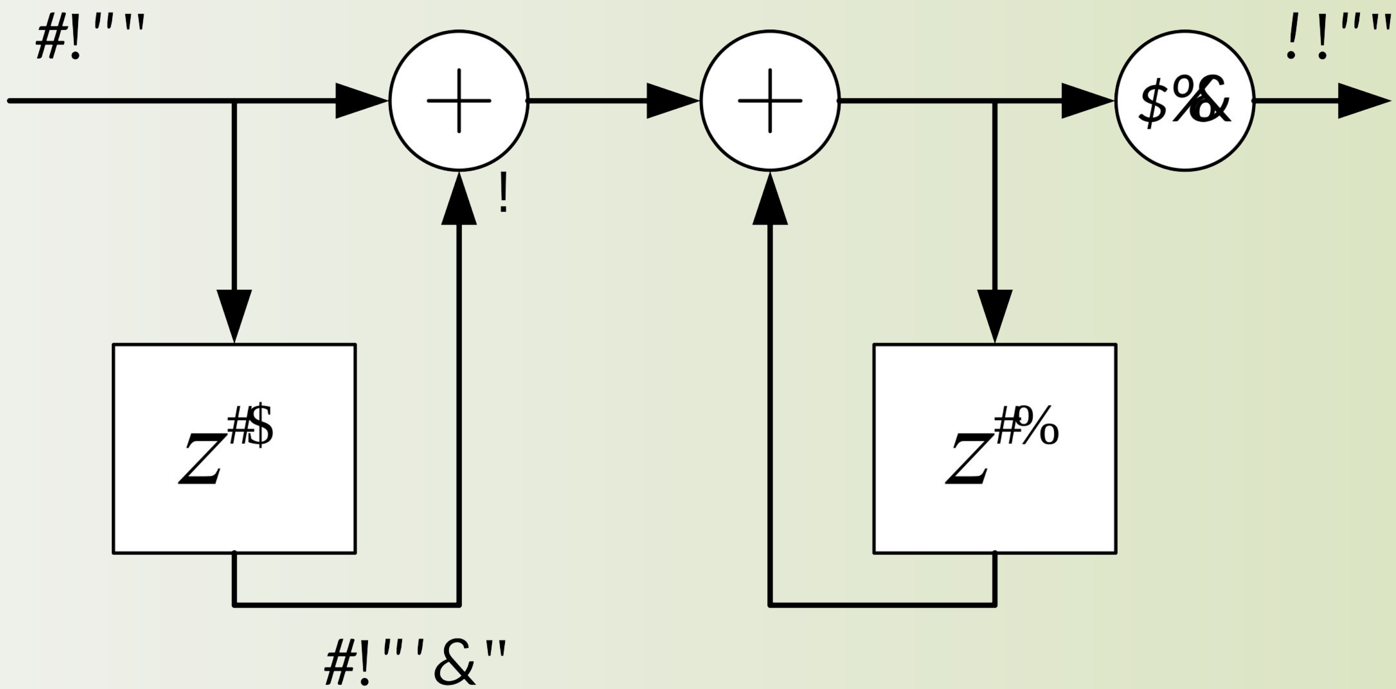
- the transfer function would be given by

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{D}\sum_{k=0}^{D-1} z^{-k} = \frac{1}{D}\,\frac{1-z^{-D}}{1-z^{-1}}$$

Moving Average Filter



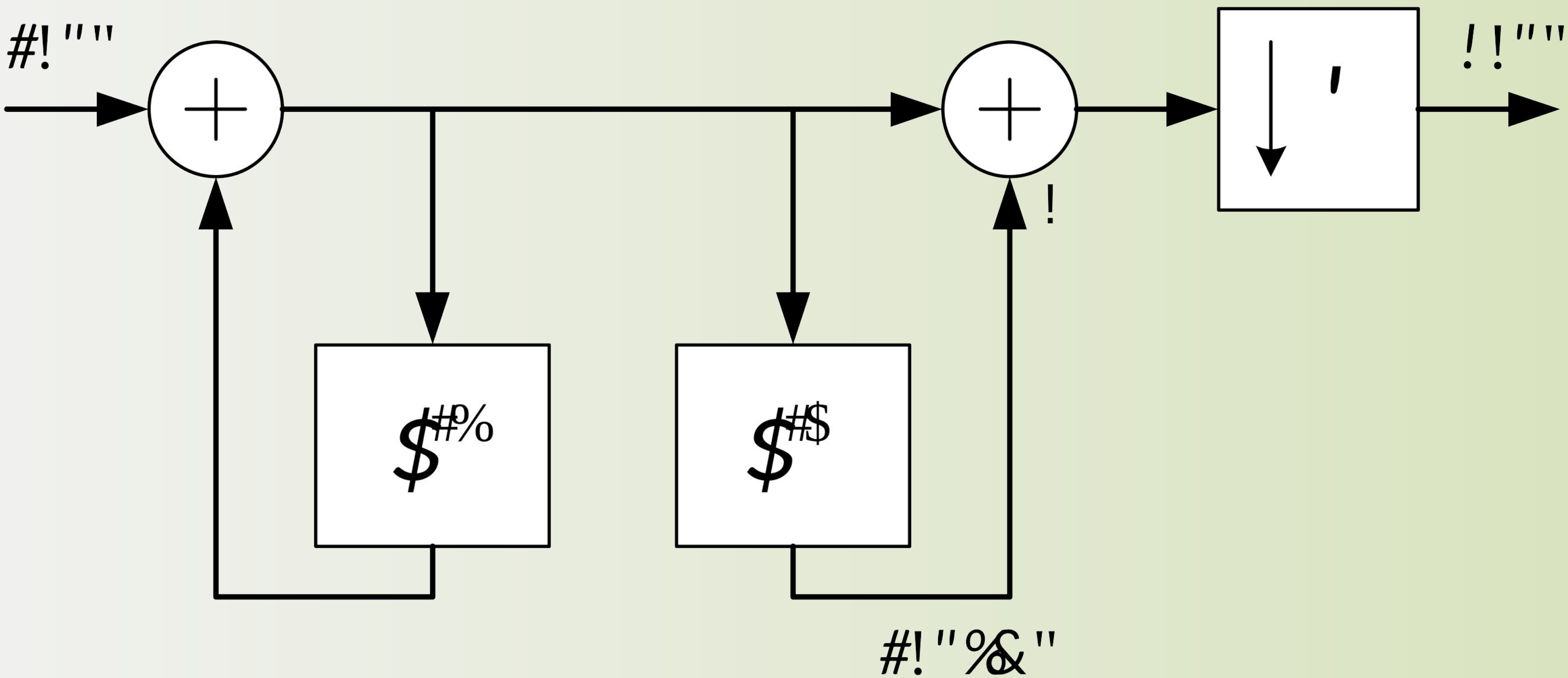
Moving Average Filter – CIC Version
Cascaded Integrator-Comb Filters



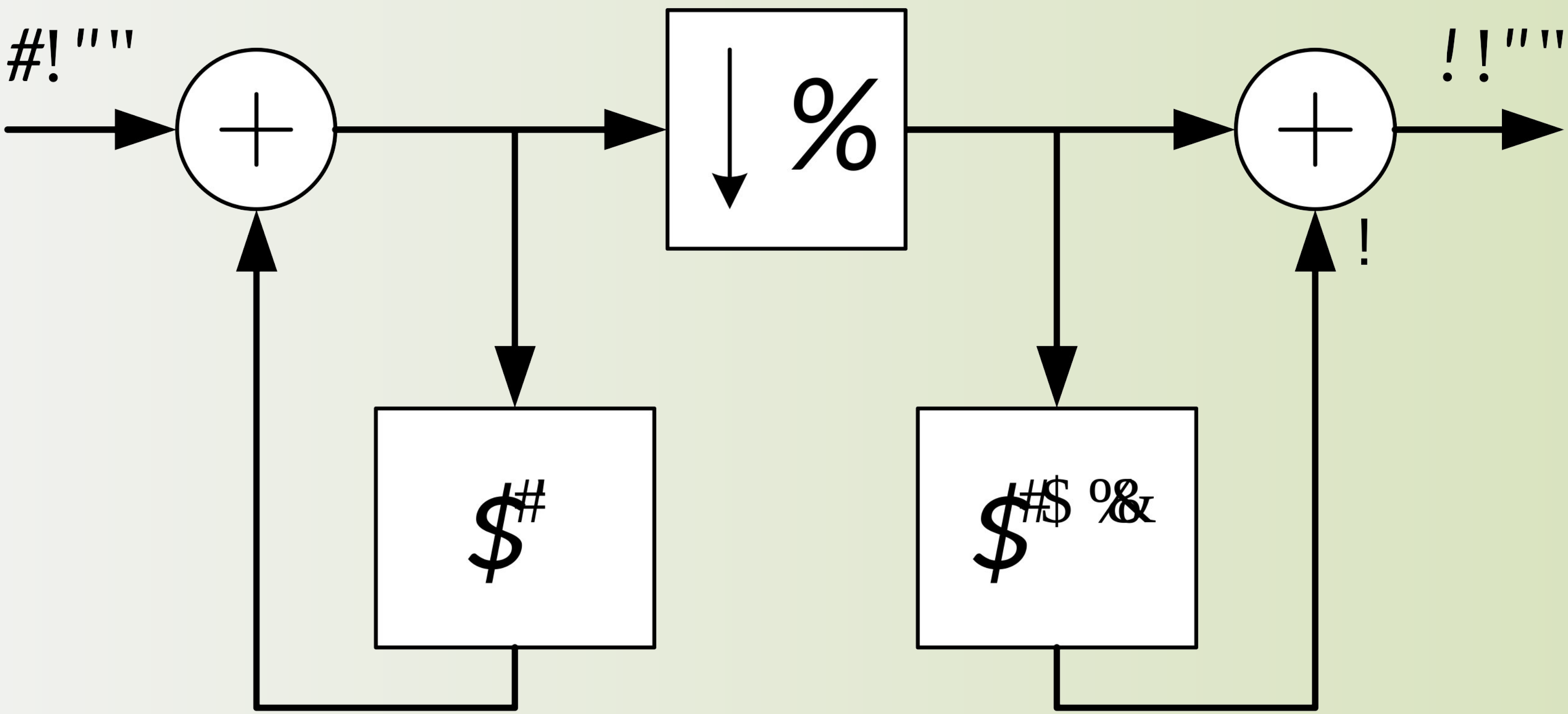
Only two additions per output sample regardless the value of *D*

$$\left| H\left(e^{j\omega}\right) \right| = \left| \frac{\sin(\omega D/2}{\sin(\omega/2)} \right|$$

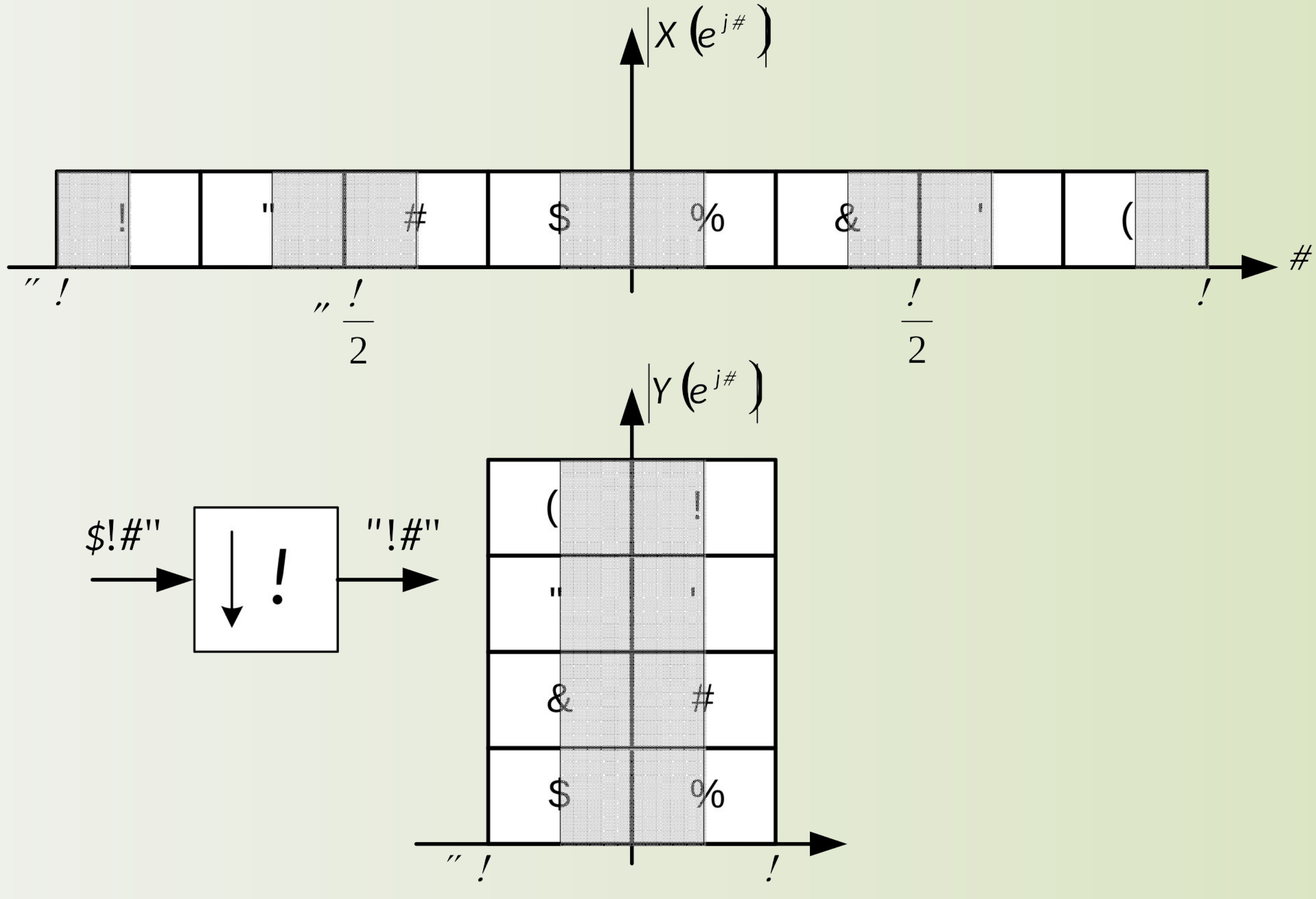
CIC Filters for Decimation



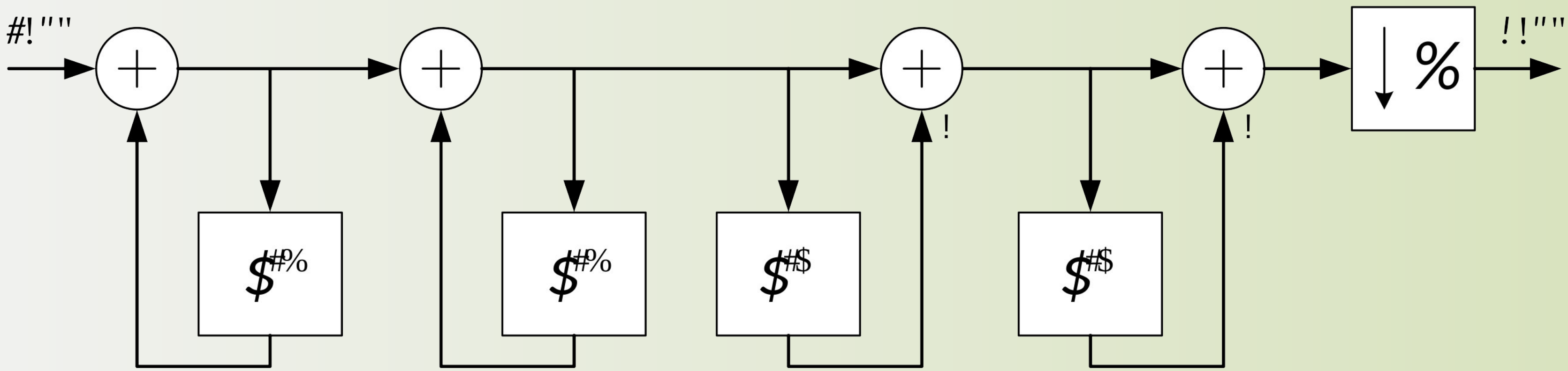
CIC Filters for Decimation – Efficient Version



Aliasing in Decimation with CIC Filters



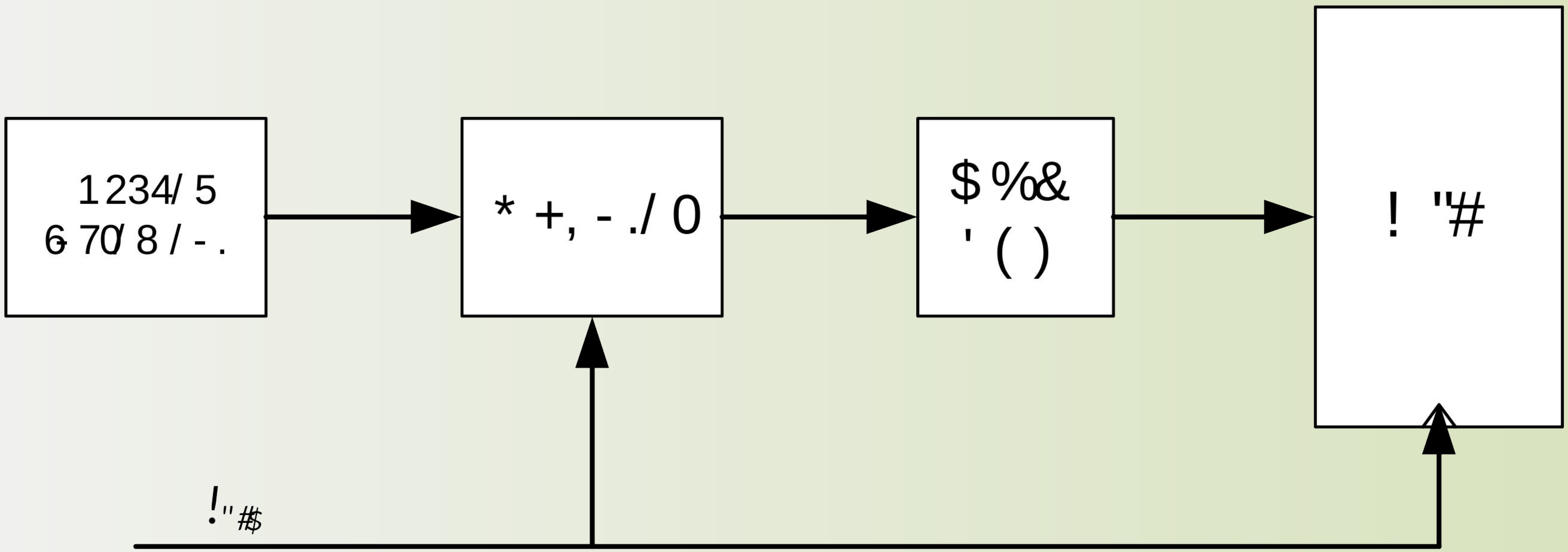
M^{th} Order CIC FIlters



$$\left|H\left(e^{j\omega}\right)\right| = \left|\frac{\sin(\omega D/2}{\sin(\omega/2)}\right|^2$$

Direct Digital Synthesis – DDS

Direct Digital Synthesis – DDS



ROM LUT - ROM Look Up Table

Direct Digital Synthesis

- Advantages over the analog generation of signals
  - Precision on the generated frequency
  - Flexibility. The parameters of the generated signal can be easily changed
  - Controlled phase on frequency changing
  - Smaller size
  - Generation of high quality signals with small distortion

Periodic Arbitrary Waveform

- It is possible to generate any waveform by storing one period on the ROM LUT
- If the period is too long we need a large memory
- However, sinusoid generation is the most common case

Sinusoid Generation by DDS

- It is possible to generate a sinusoid using DDS with a phase accumulator
- Consider a continuous sine wave

$$\sin(2\pi f_0 t + \phi) = \sin(2\pi f_0 t + 2\pi f_0 T_0 + \phi) = \sin(2\pi f_0 (t + T_0) + \phi)$$

- The sine function is periodic because
- If we consider the discrete case we will have

$$2\pi f_0 T_0 = 2\pi$$

$$\sin(2\pi f_0 n T_s + \phi) = \sin(2\pi \frac{f_0}{f_s} n + 2\pi \frac{f_0}{f_s} N + \phi) = \sin(2\pi \frac{f_0}{f_s}(n + N) + \phi)$$

Sinusoid Generation by DDS

- To have a periodic sine wave the value at the sample *n* must be equal to the value at the sample *n*+*N*.
- This implies that

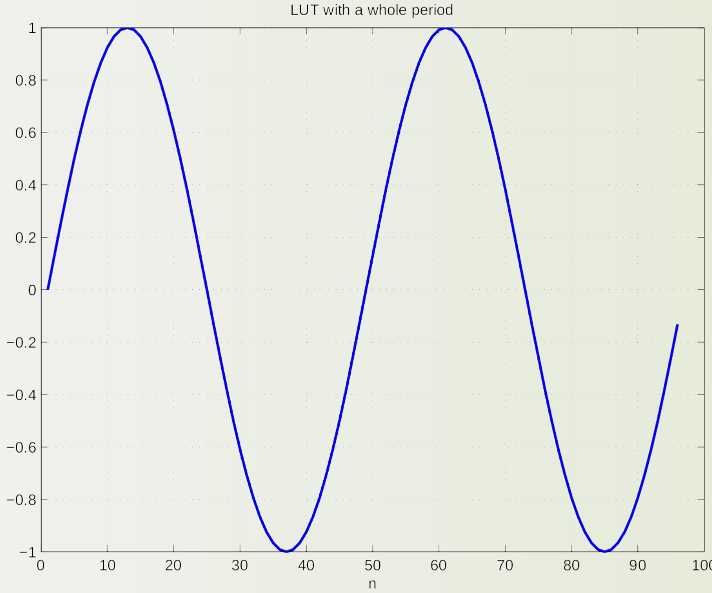
$$2\pi \frac{f_0}{f_s} N = 2\pi k$$

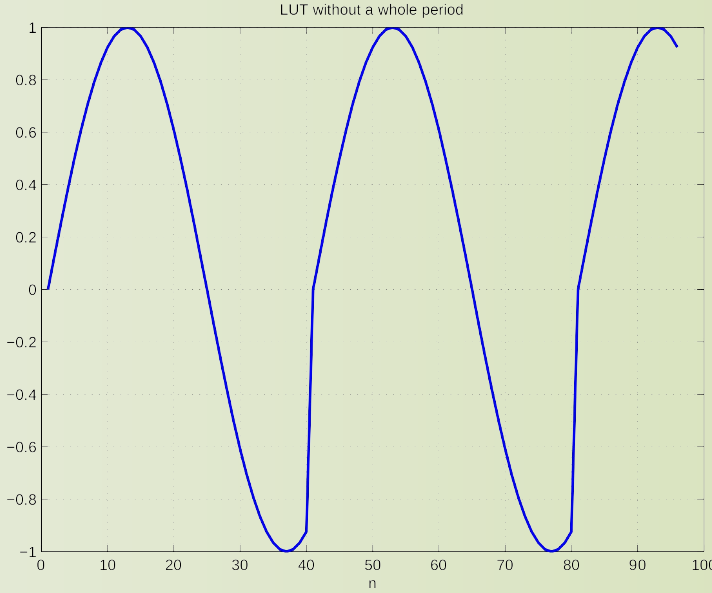
- where *k* is an integer. Rearranging this equation we have

$$\frac{N}{k} = \frac{f_s}{f_0}$$

- As *N* and *k* are integers f_s/f_0 must be a rational number in order the discrete sine to be periodic

Sine Generation with DDS





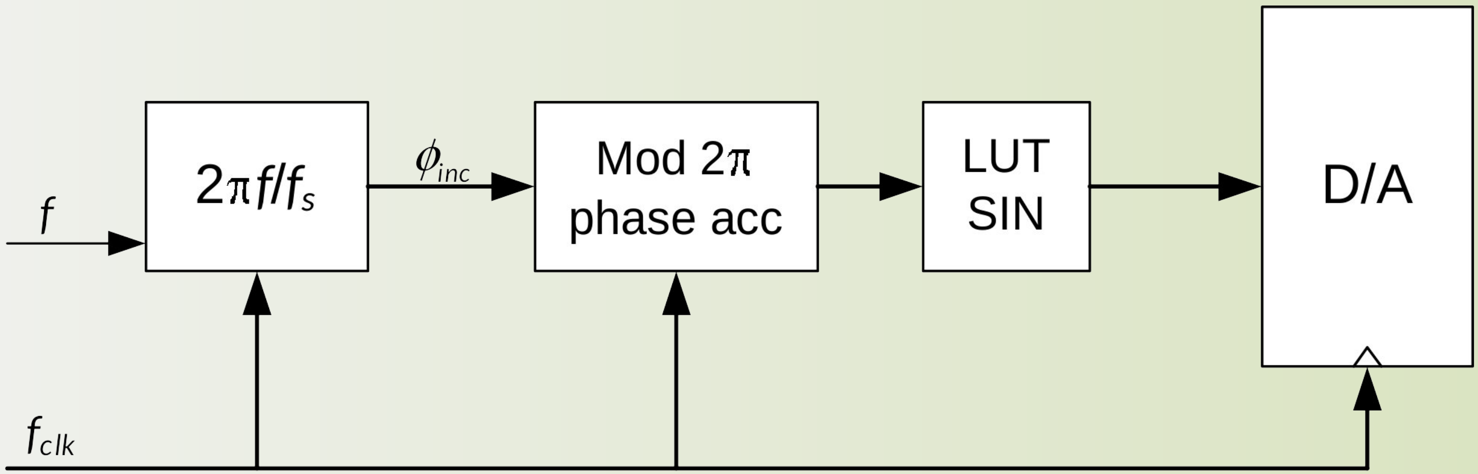
The LUT should have a complete sine period

Sine Generation with a Phase Accumulator

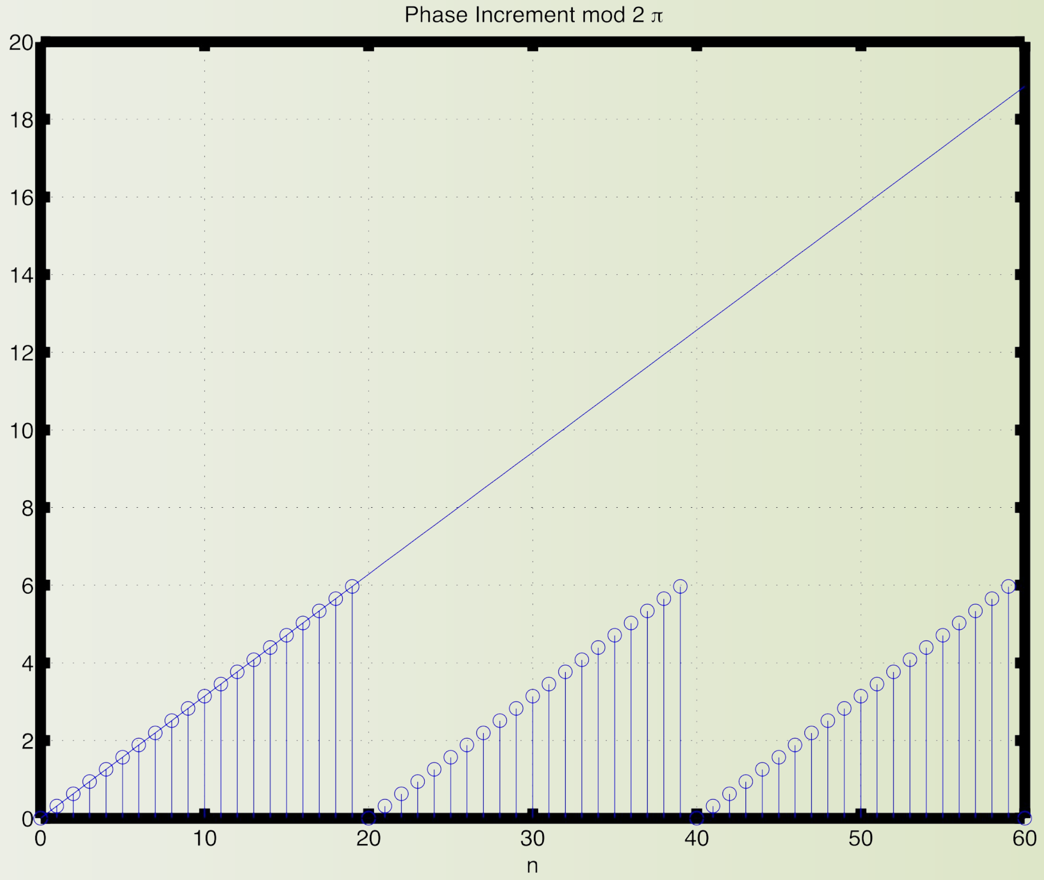
- Returning to the expression of the sampled sinusoid

$$\sin(2\pi f_0 n T_s) = \sin(2\pi \frac{f_0}{f_s} n) = \sin(\phi_{inc} n)$$

- The phase increment ϕ_{inc} is also known as the normalized frequency

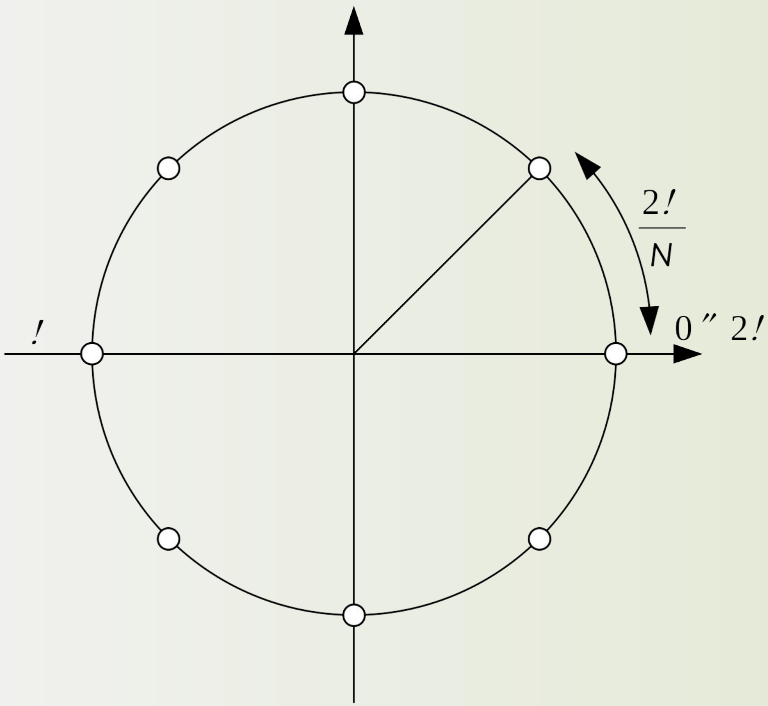


Sine Generation with a Phase Accumulator



Sine Generation with a Phase Accumulator

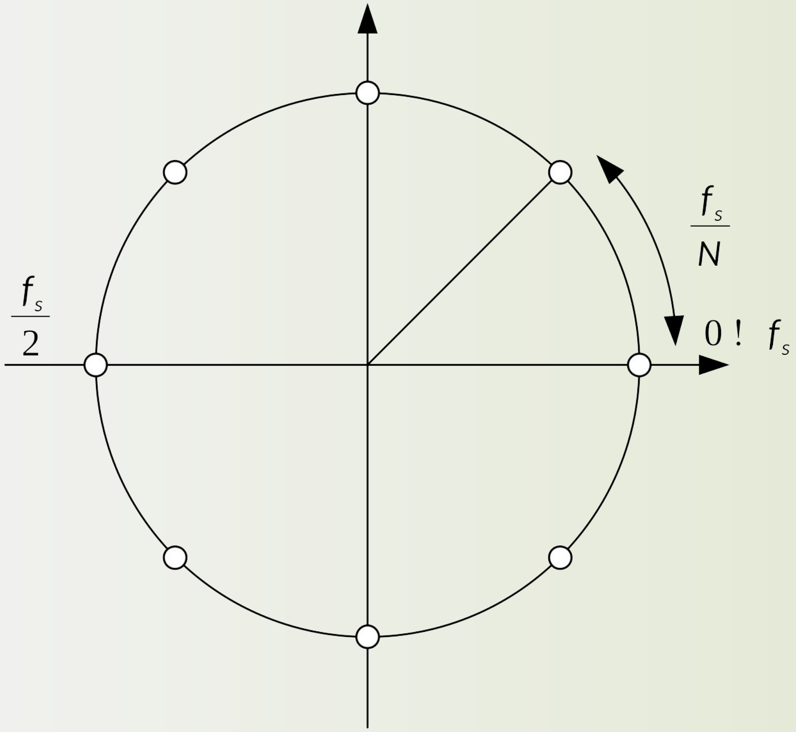
Sample points on the unit circle in radians



- Each point on the unit circle represents a sample of the sin function to store on the table
- The minimum phase increment is $2\pi/N$
- This is also the frequency resolution of the generator

Sine Generation with a Phase Accumulator

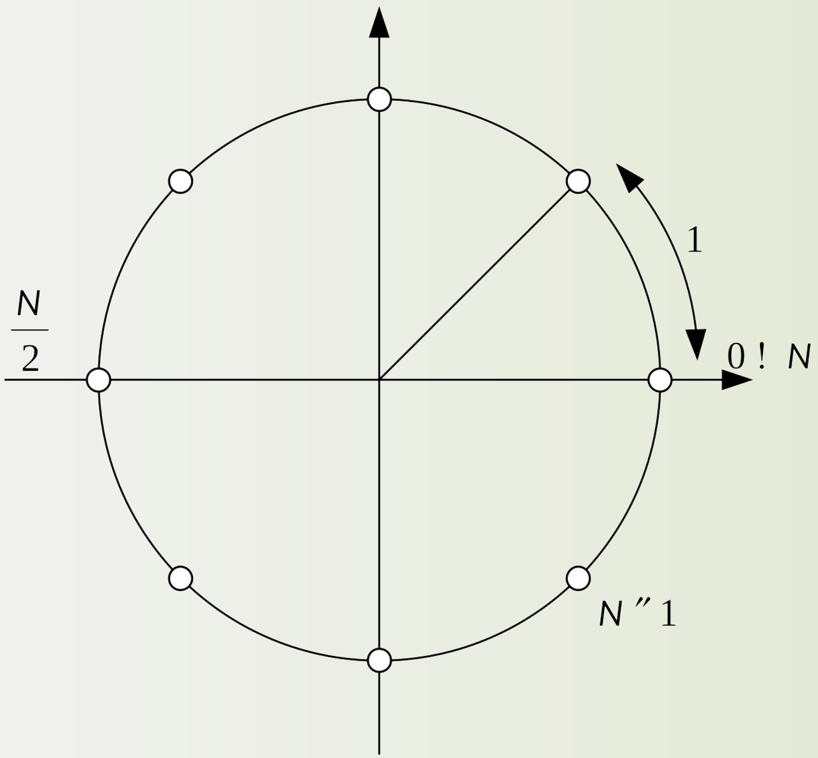
Sample points on the unit circle in Hertz



- If we normalize to Hertz we can read directly the generated frequency
- The resolution is f_s/N

Sine Generation with a Phase Accumulator

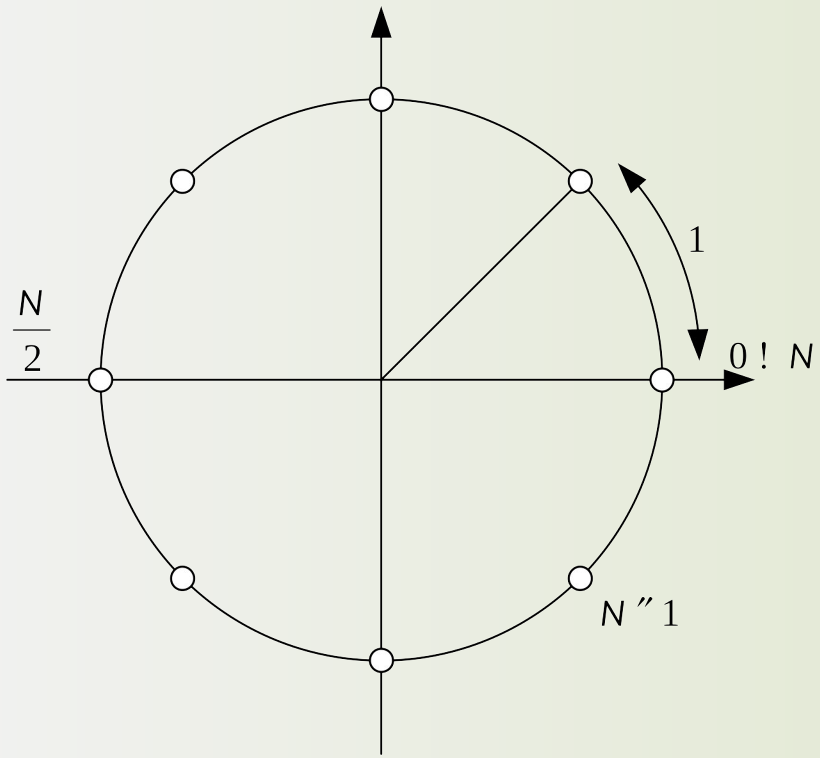
Sample points on the unit circle normalized to *N*



- Instead of dealing with arithmetic mod 2π we can normalize to *N* and work with the more convenient arithmetic mod *N*
- The minimum frequency is achieved with a step of 1

Sine Generation with a Phase Accumulator

Sample points on the unit circle normalized to *N*



Sin table with *N* samples

Frequency Resolution and LUT Size

- If the phase increment is represented by an integer then the minimum value is 1
- As the LUT has a whole sine period, if it has *N* samples the maximum value for the generated sine period is

$$T = NT_s$$

- Then the minimum frequency is

$$f_0 = \frac{f_s}{N}$$

- This f_0 corresponds to a phase increment of 1 and it is the frequency resolution of the generator

Frequency Resolution and LUT Size

- To guarantee the frequency resolution we should have

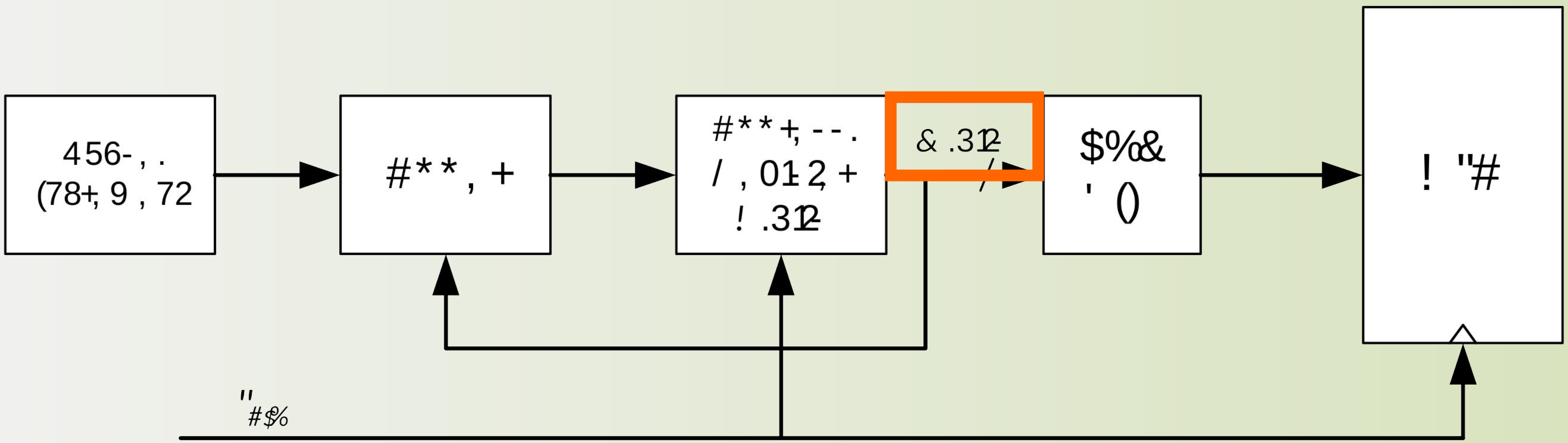
$$N > \frac{f_s}{f_0}$$

- If we have f_s=100MHz and f_0=10Hz then N must be larger than 1e7. We will need a counter with 24 bits and a table with 16MB.
- This table size is not reasonable. Fortunately, several techniques can be used to reduce the table size.

Techniques to Reduce the LUT Size

1. Use a fractional counter where only the integer part is used to address the LUT
2. Store only one quadrant of the sine period
3. Interpolation using Taylor’s Series Expansion
4. Interpolation using trigonometric identities

Reducing the LUT Size



Spurious Frequencies

- The main frequency of a DDS is given by

$$\Delta_r / \left(N / T_{clk} \right)$$

- When we have the relation

$$N / \gcd\left(\Delta_r, N\right)$$

- Example 1: N=16 and Δ_r=2

$$16 / \gcd(2,16) = 8$$

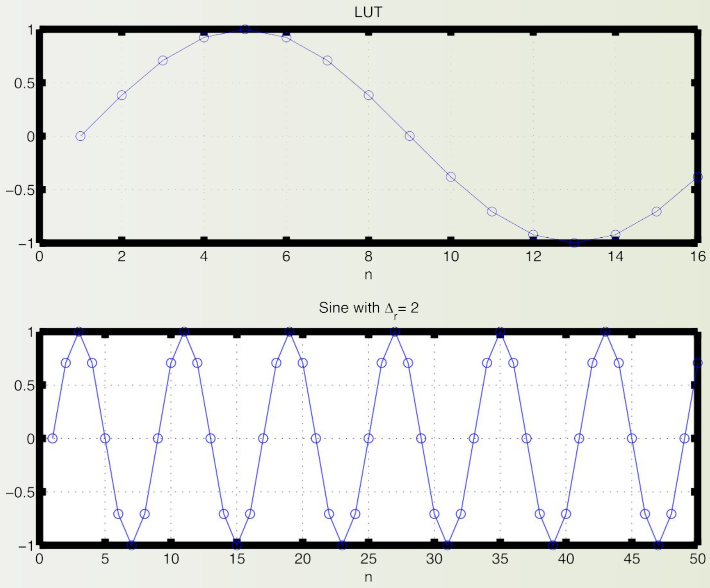
- Example 2: N=16 and Δ_r=6

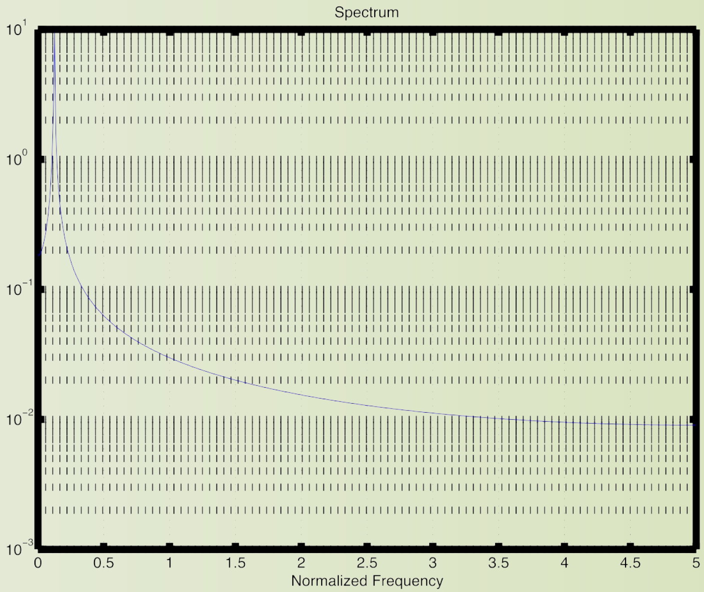
$$16 / \gcd(6,16) = 8$$

We have a secondary period with 8 samples

Spurious Frequencies

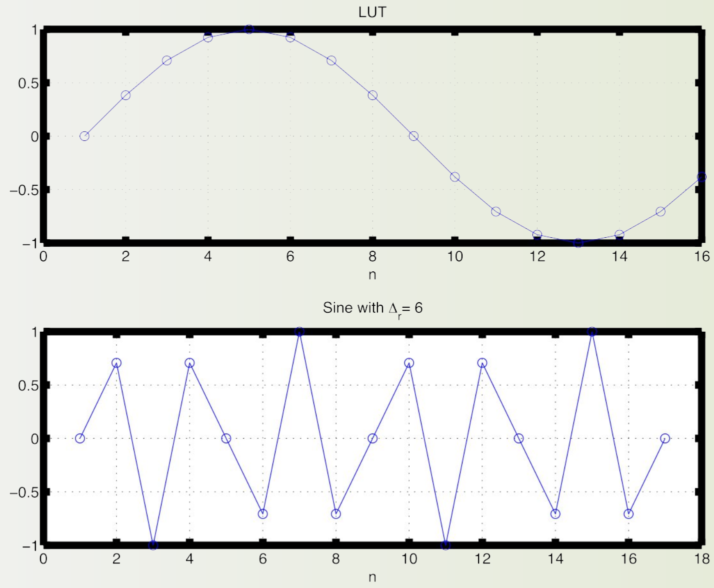
Example 1

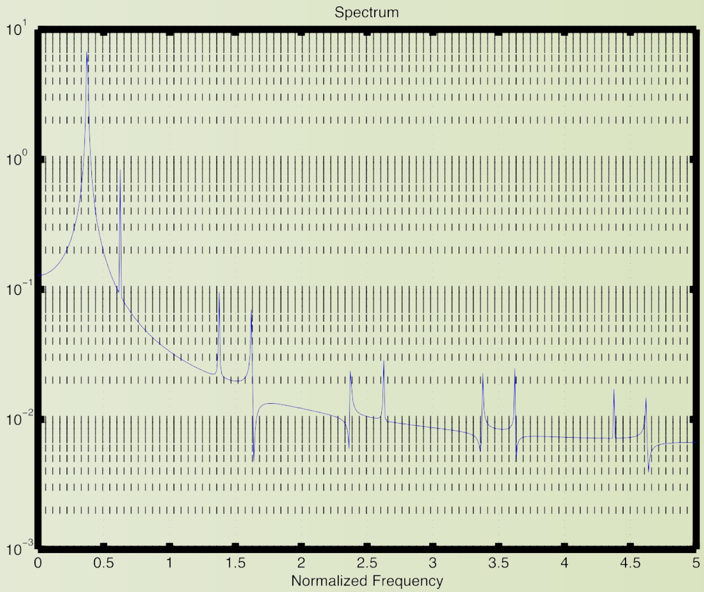




Spurious Frequencies

Example 2





Interpolation Using Trigonometric Identity

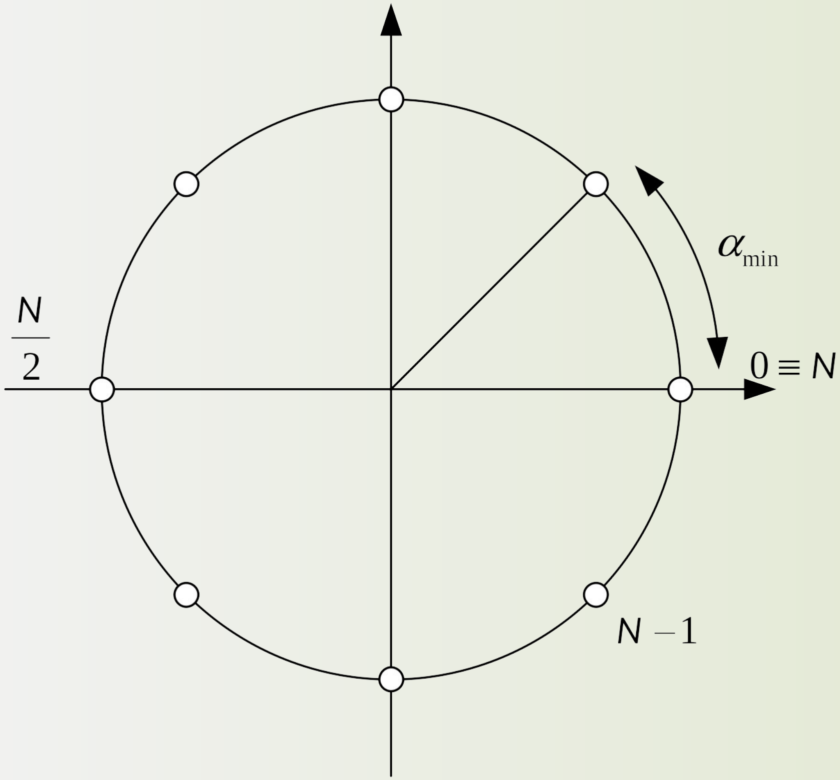
- To increase the resolution of the sine function without increasing the size of the LUT we can use the following trigonometric identity

$$\sin(\alpha+\beta)=\sin(\alpha)\cos(\beta)+\cos(\alpha)\sin(\beta)$$

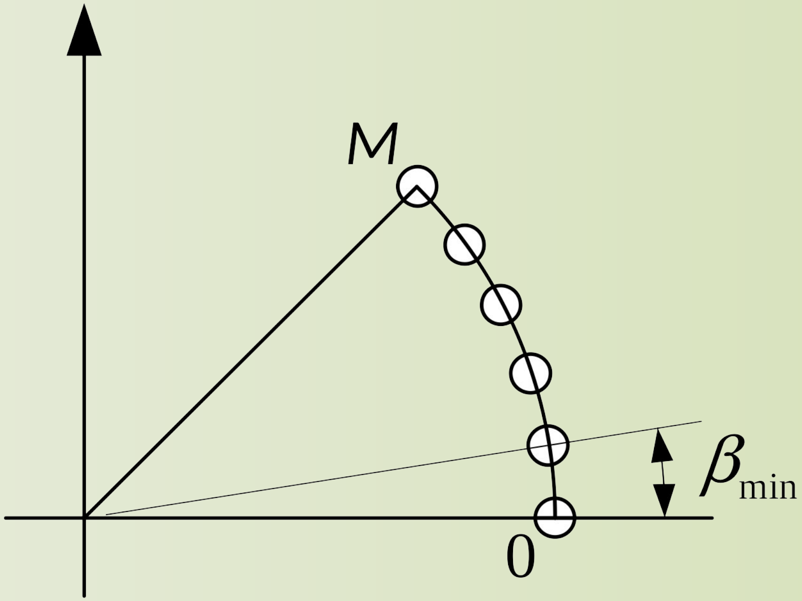
- The angle α is a coarse table having the sine values for angles around the circle.
- The angle as a fine resolution from zero to the first value of α.

Interpolation Using Trigonometric Identity

Coarse angle α



Fine angle β



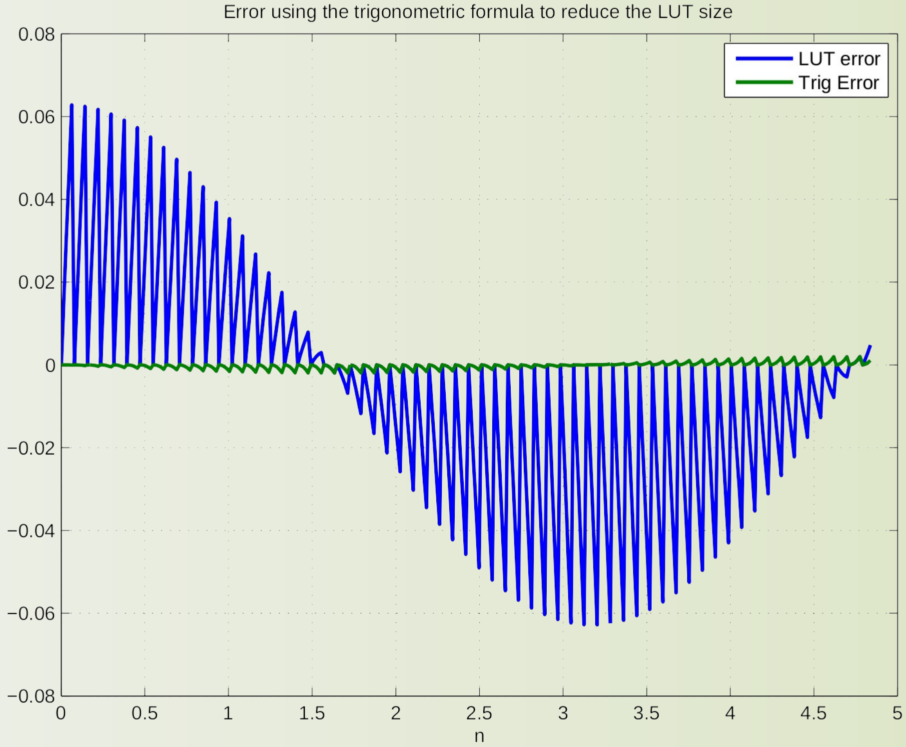
Interpolation Using Trigonometric Identity

- So we can use two tables, one for α and another one for β.
- However, if α_{min} is small enough, then the angle β is always close to zero, and we can simplify the previous equation

$$\text{if} \quad \beta \approx 0$$
$$\cos(\beta) \approx 1$$
$$\sin(\beta) \approx \beta$$
$$\sin(\alpha + \beta) \approx \sin(\alpha) + \cos(\alpha)\beta$$

Interpolation Using Trigonometric Identity

- With this technique is possible to evaluate the sine function with a great resolution using a small LUT and a few more calculations and two LUT reads.



Example for a LUT with 80 samples

Main Source of Errors on DDS Systems

- Amplitude truncation – lack of bits on the LUT
- Phase truncation – lack of bits to address the LUT
- DAC resolution – the number of bits of the DAC