Chapter 3.
Digital Front End (DFE)

Chapter 3. DFE - Outline

- General Aspects
- DFE Tx
- DFE Rx

General Aspects - Purpose

- Several Tx/Rx signal processing operations are performed at this stage
- These operations are not related to a particular standard and depend on the equipment manufacturer
- Examples:
  - Filtering
  - Sampling frequency conversion
  - Amplifier linearization
  - Signal level conversion
  - DC offset compensation
  - Carrier recovery
  - Automatic gain control

General Aspects - Implementation

- Can be implemented in a dedicated chipset
  - Analog devices (AD9857, AD9777)
  - Texas Instruments
- Can be implemented in the baseband processing DSP
- Can be implemented in a dedicated processor, together with the RF part

Chapter 2. DFE - Outline

- General Aspects
- **DFE Tx**
- DFE Rx

General diagram – DFE Tx

- INT – Interpolation block
  - Increaseas the sampling frequency (Oversampling)

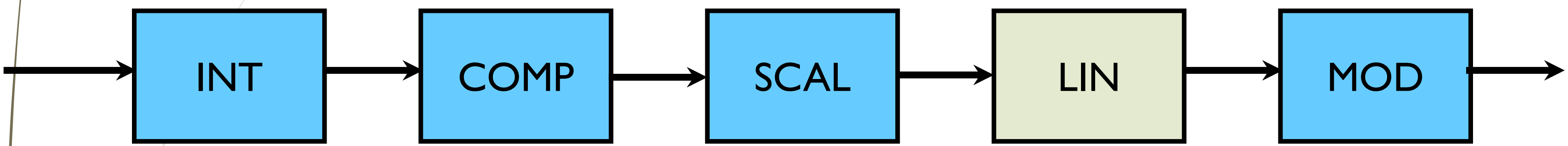
General diagram – DFE Tx

- COMP – Compensation block
  - Compensates in frequency domain the possible imperfections from the DFE

General diagram – DFE Tx

- SCAL – Scaling block
  - Adjusts the power level of the output signal

General diagram – DFE Tx



- LIN – Amplifier Characteristic Linearization
  - Decreases the peak power of the signal and compensates the non-linearities that the power amplifier **will** introduce

General diagram – DFE Tx

- MOD – Digital mixing block
  - Performs a frequency translation of the obtained signal

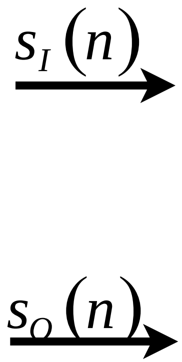
General diagram – DFE Tx

- Many of the blocks are optional, and the corresponding implemented functions are varied and don't comply with a particular standard

DFE Tx - Architectures

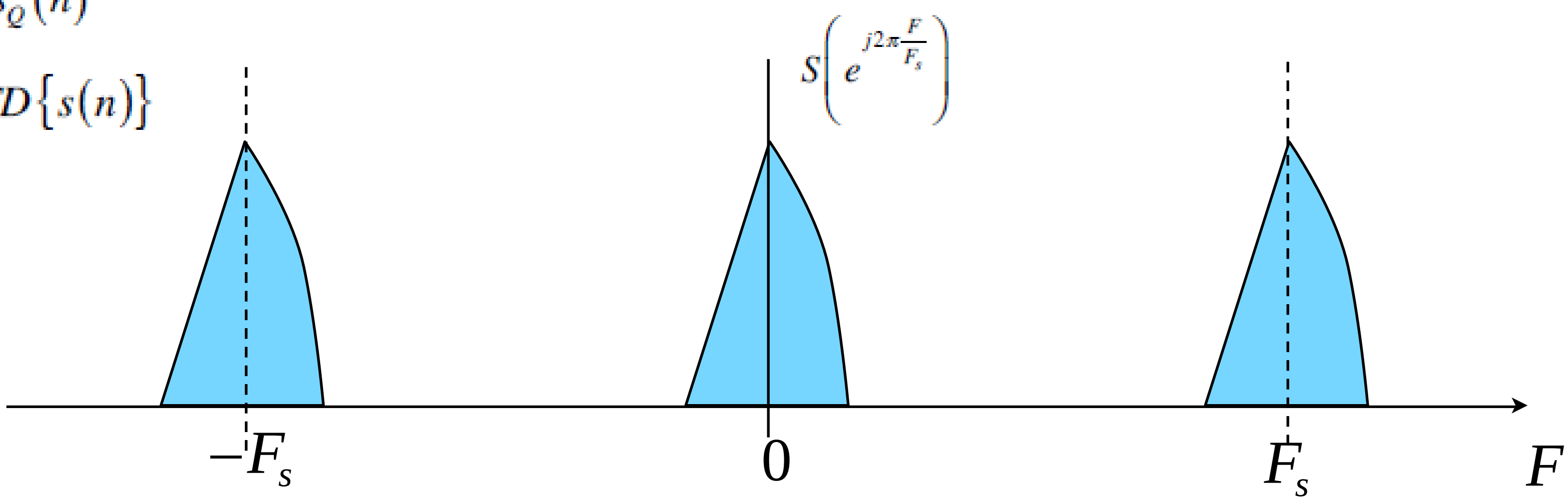
- Direct conversion architecture
  - The BB output is double: I/Q
  - Better suited for integration
- Digital intermediate frequency architecture
  - The BB output is simple
  - The modulation on first IF is made in the digital domain

Digital IF architecture

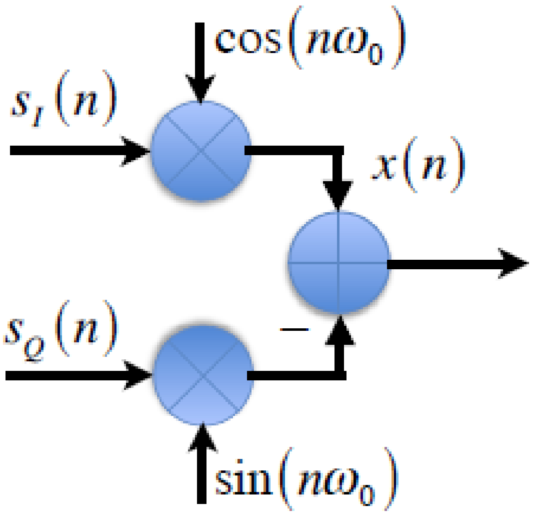


$$s(n) = s_I(n) + js_Q(n)$$

$$S\left(e^{j2\pi\frac{F}{F_s}}\right) = TFTD\{s(n)\}$$



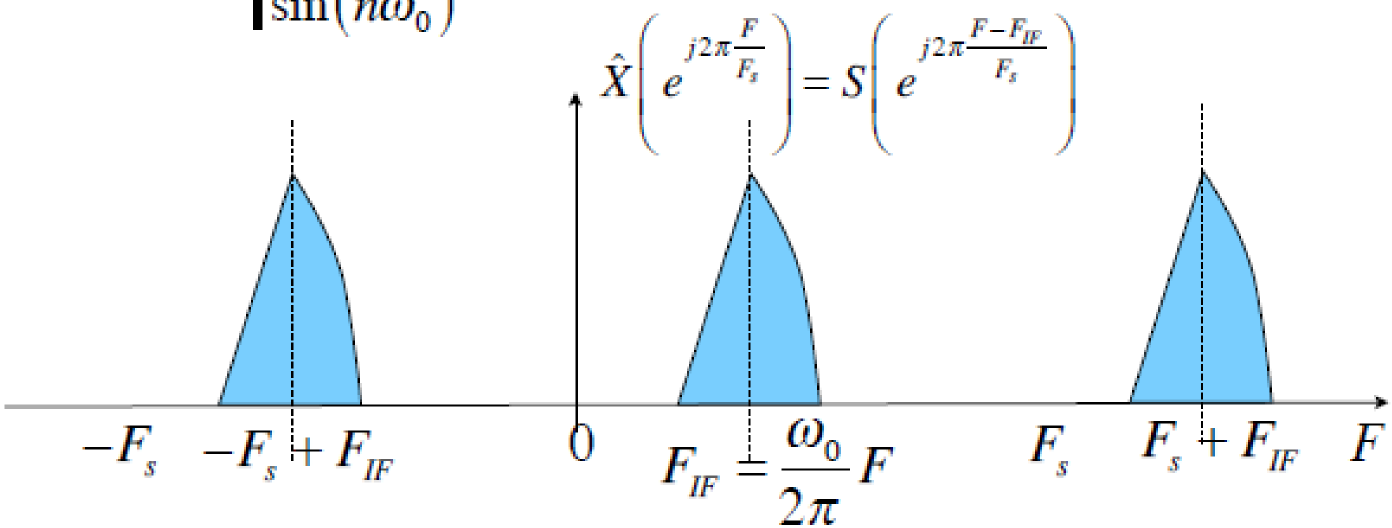
Digital IF architecture (2)



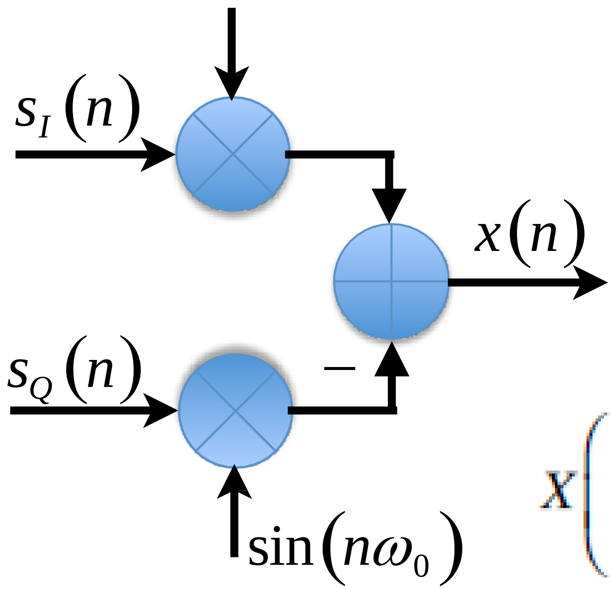
$$x(n) = s_I(n)\cos(n\omega_0) - s_Q(n)\sin(n\omega_0)$$

$$x(n) = \mathrm{Re}\{\hat{x}(n)\}$$

$$\hat{x}(n) = s(n)e^{jn\omega_0}$$

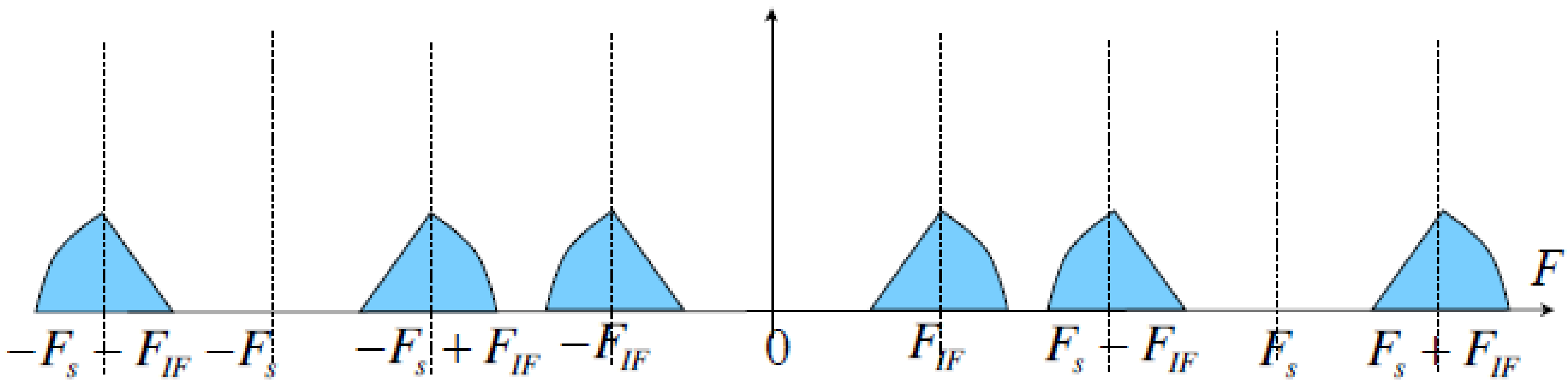


Digital IF architecture (3)

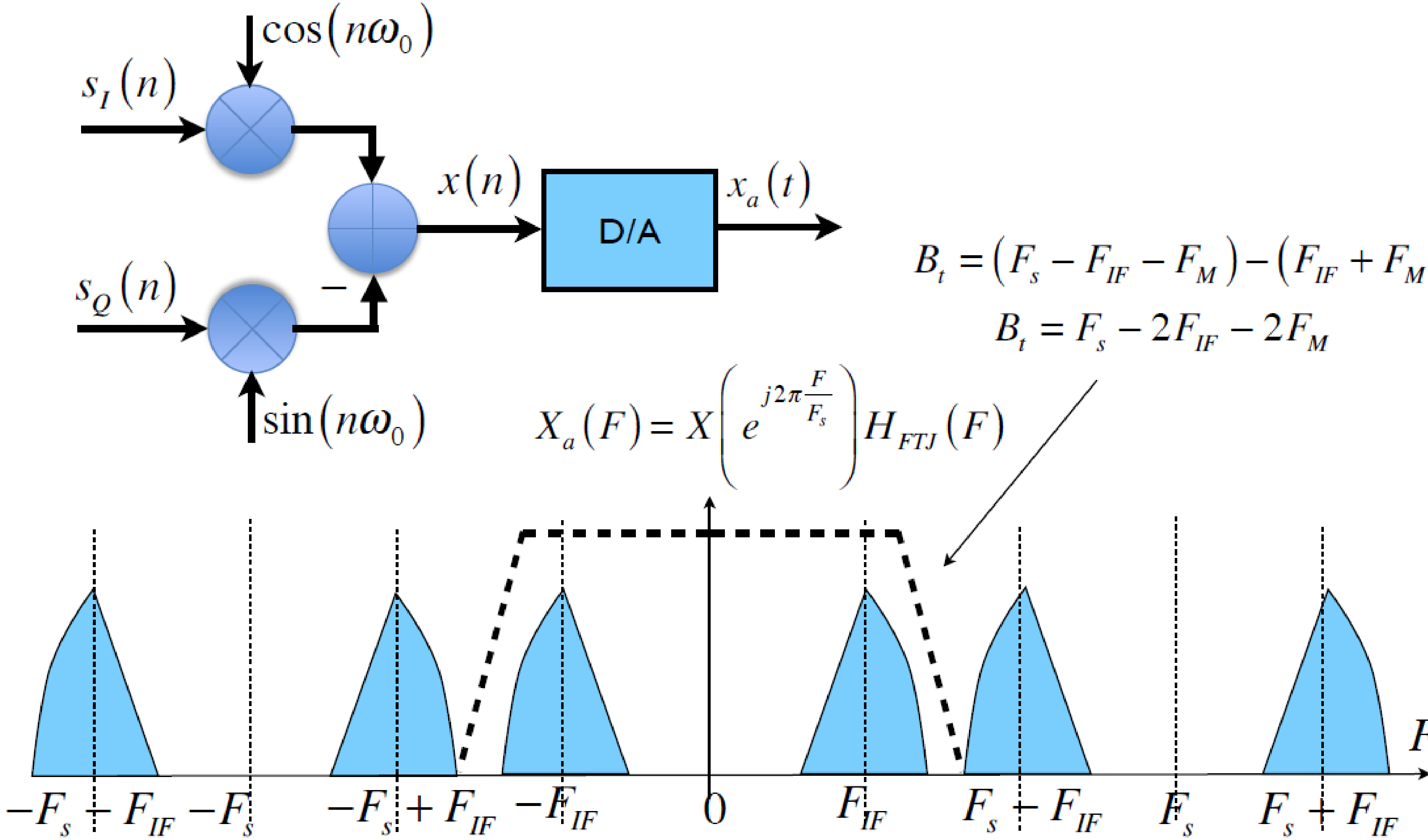


$$x(n) = \mathrm{Re}\{\hat{x}(n)\}$$

$$X\left(e^{j2\pi\frac{F}{F_s}}\right) = \frac{1}{2}\left(\hat{X}\left(e^{j2\pi\frac{F}{F_s}}\right) + \hat{X}^*\left(e^{-j2\pi\frac{F}{F_s}}\right)\right)$$



Digital IF architecture (4)

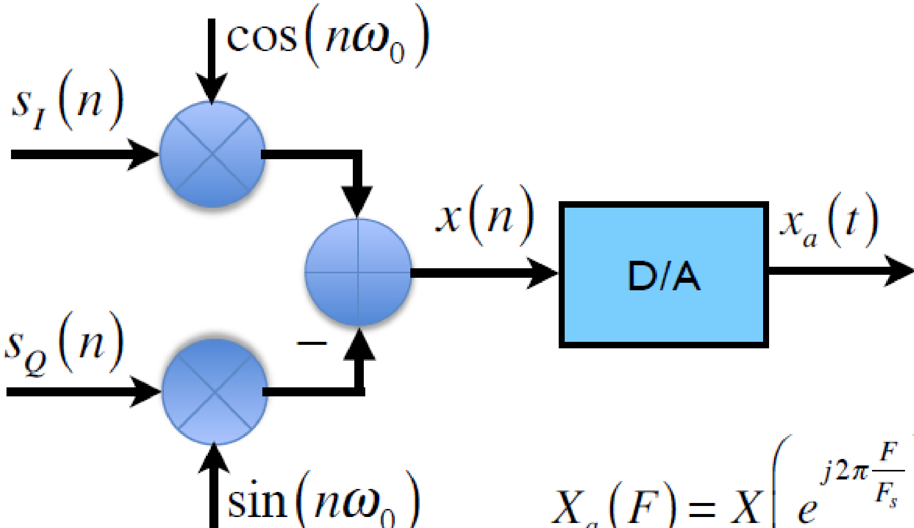


$$B_t = (F_s - F_{IF} - F_M) - (F_{IF} + F_M)$$

$$B_t = F_s - 2F_{IF} - 2F_M$$

$$X_a(F) = X\left(e^{j2\pi\frac{F}{F_s}}\right)H_{FTJ}(F)$$

Digital IF architecture (4)

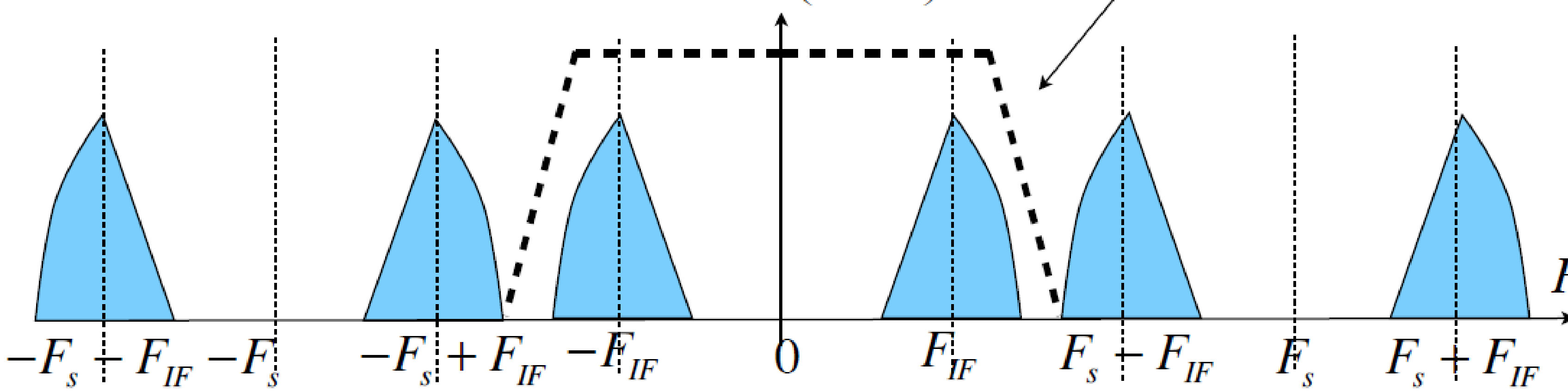


Transition band

$$B_t = (F_s - F_{IF} - F_M) - (F_{IF} + F_M)$$

$$B_t = F_s - 2F_{IF} - 2F_M$$

$$X_a(F) = X\left(e^{j2\pi\frac{F}{F_s}}\right)H_{FTJ}(F)$$



Comments regarding the mixing process

- After the mixing, F_S has to comply with the Nyquist condition for the digital mixed signal:

$$F_s \geq 2\left(F_{IF} + F_M\right)$$

- Usually, for ease: $F_{IF} = \dfrac{F_s}{4}$
  - It is not compulsory

$$F_M \leq \frac{F_s}{4}$$

- The signal has to be oversampled
  - The pulse-shaping filter will be used

Comments regarding the mixing process (2)

- The transition band of the filter that is responsible for suppressing the image components is

$$B_t = F_s - 2F_{IF} - 2F_M = \frac{F_s}{2} - 2F_M$$

- It cannot be chosen $F_M = \frac{F_s}{4} \quad (B_t = 0)$
  - An oversampling with at least 2 has to be done $(F_{Nyquist} = \frac{F_s}{2})$
- The oversampling factor *L* can be defined as

$$L = \frac{F_s}{2F_M} \qquad B_t = 2F_M(\frac{L}{2} - 1)$$

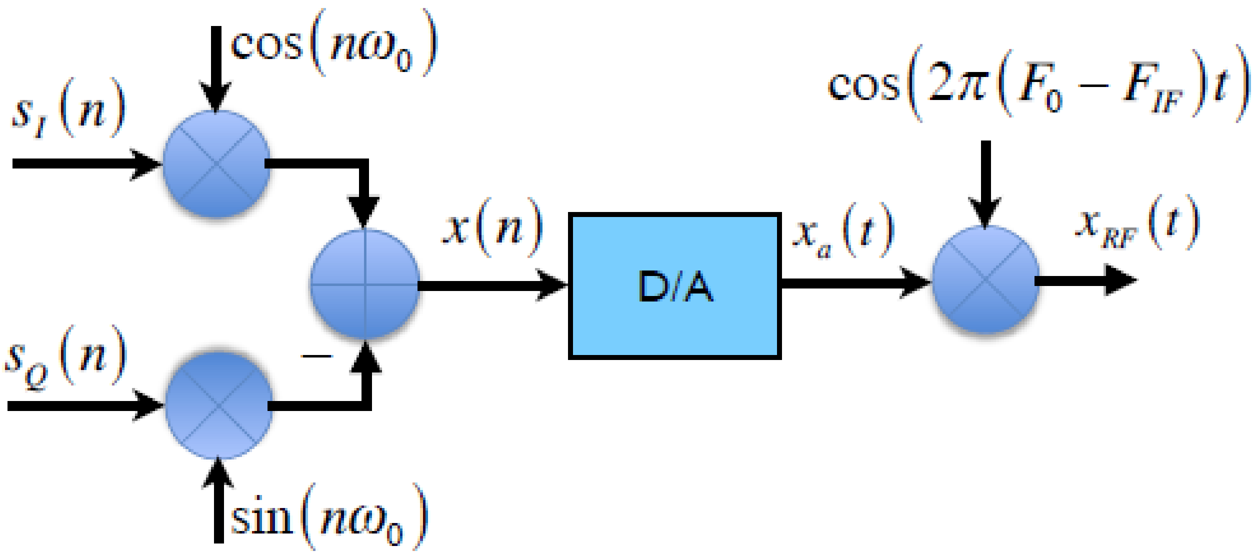
- If $F_{IF} \neq \frac{F_s}{4}$

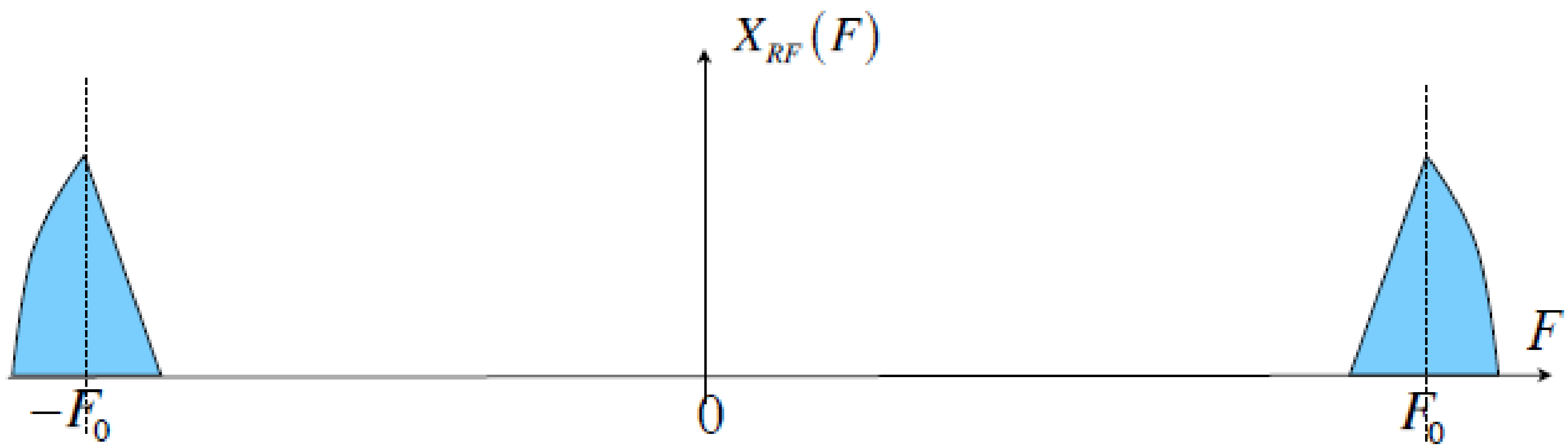
$$B_t = 2F_M(L - \frac{F_{IF}}{F_M} - 1)$$

Comments regarding the mixing process (3)

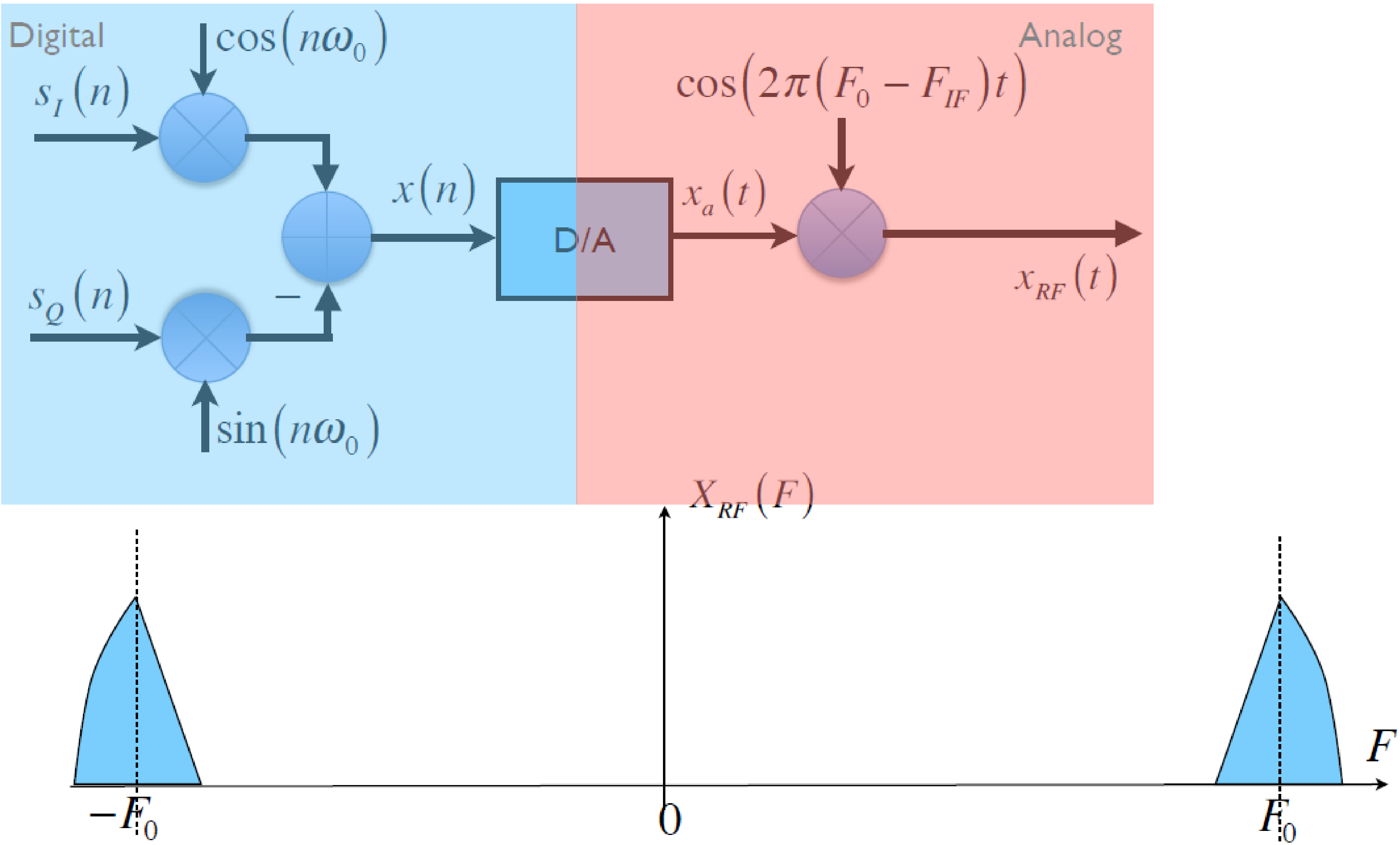
- The higher the oversampling in the digital domain, the higher the transition band of the analog filter
  - The analog filtering is easier to be done
- The oversampling can be achieved
  - In the pulse-shaping filter
    - Seldom, here an interpolation order as small as possible is usually chosen
  - In the DFE
    - The interpolation order can be in the range 32…64

Digital IF architecture (5)





Digital IF architecture (6)



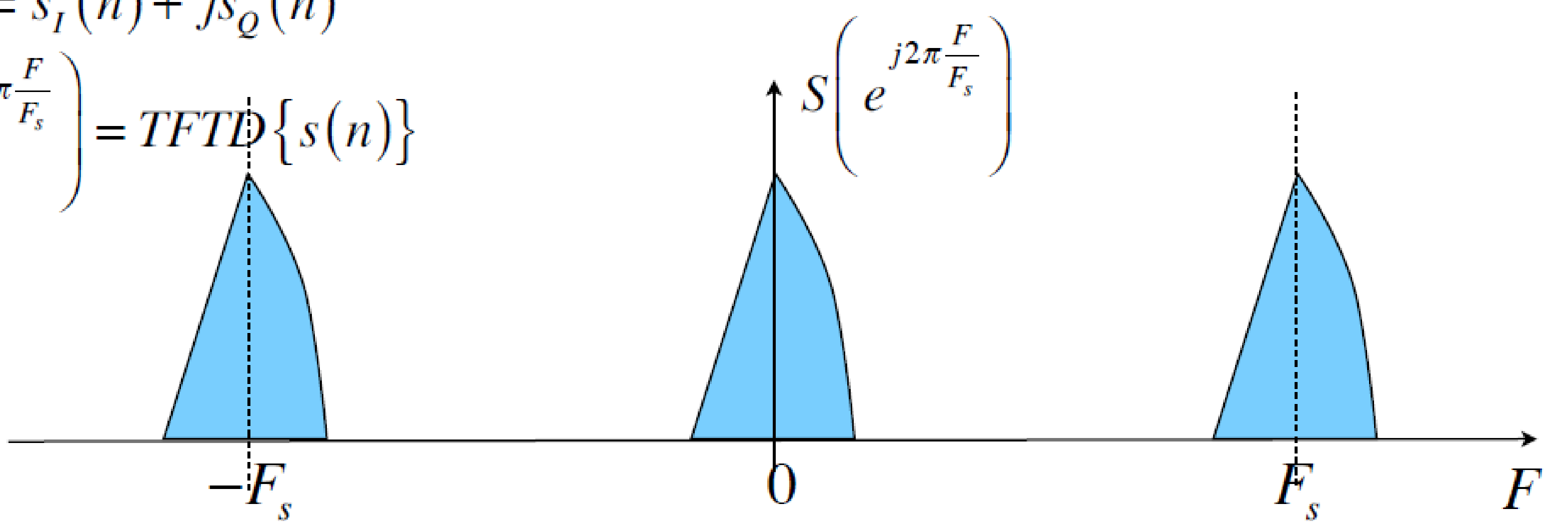
Direct conversion architecture

$s_I(n)$

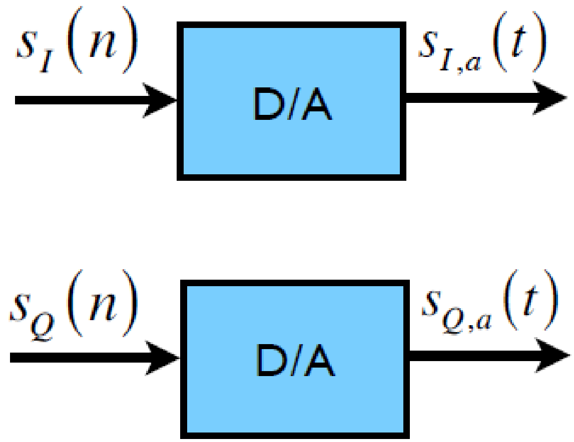
$s_Q(n)$

$$s(n) = s_I(n) + js_Q(n)$$

$$S\left(e^{j2\pi\frac{F}{F_s}}\right) = TFTD\{s(n)\}$$

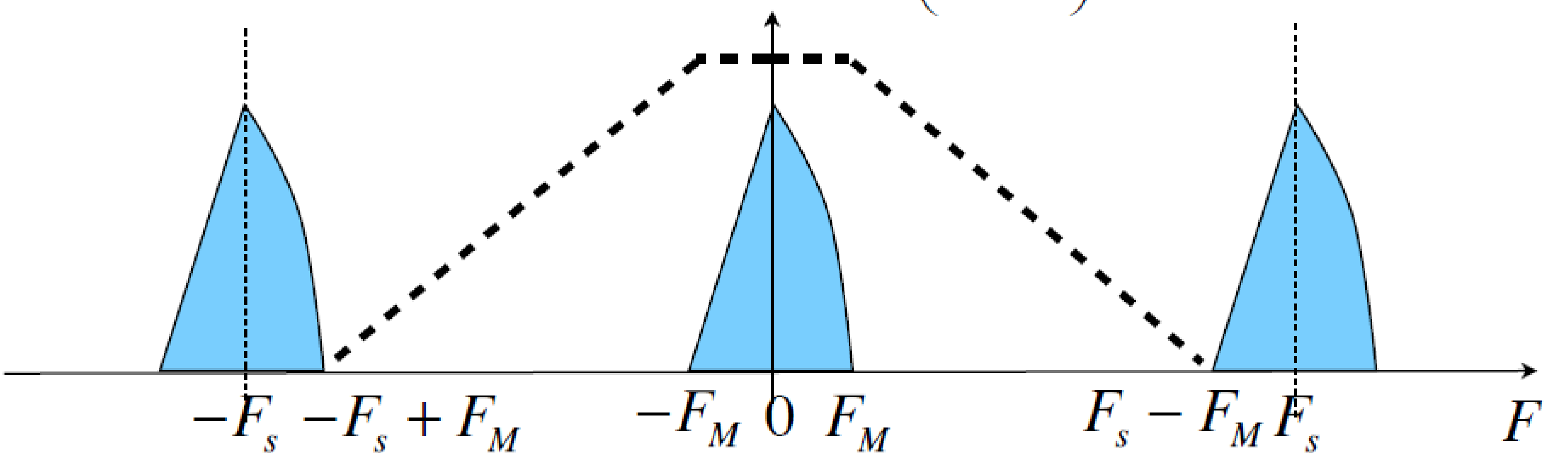


Direct conversion architecture (2)



$$s_a(t) = s_{I,a}(t) + j s_{Q,a}(t)$$

$$S_a(F) = S\left(e^{j2\pi \frac{F}{F_s}}\right) H_{FTJ}(F)$$



Comments regarding the filtering

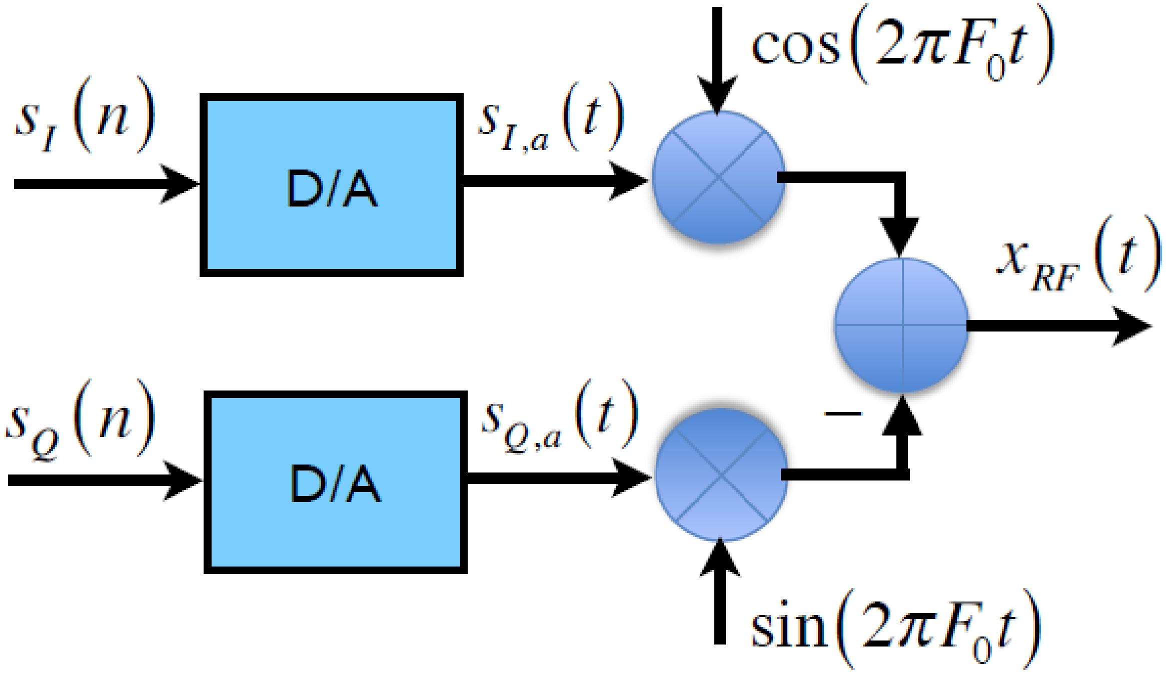
- The filter has the transition band

$$B_t = (F_s - F_M) - F_M = F_s - 2F_M$$

$$B_t = 2F_M(L-1)$$

- The oversampling is needed in this case too
- The filtering is not ideal and is usually not performed in the D/A converter
  - An analog filter is needed after the D/A converter in order to remove the unwanted components from the spectrum

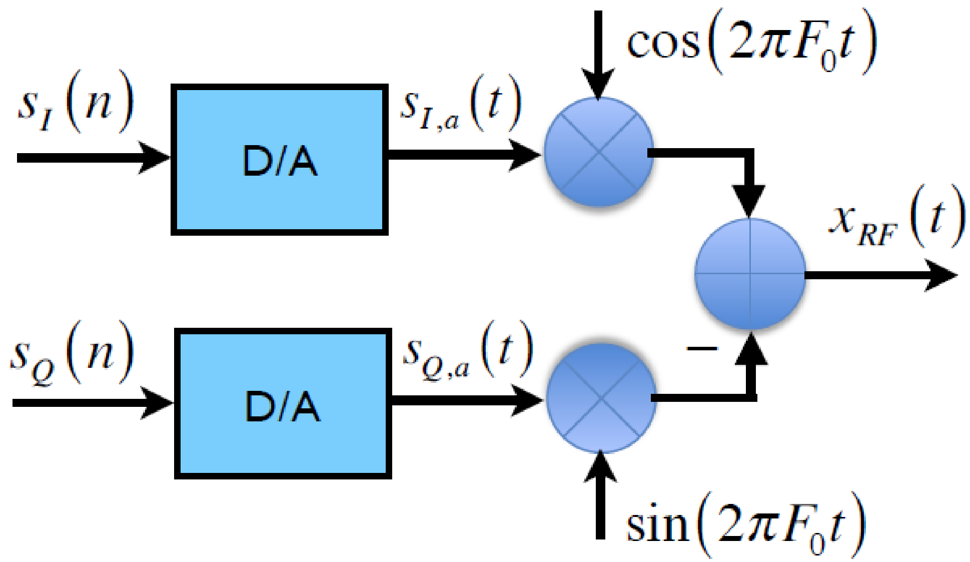
Direct conversion architecture (3)

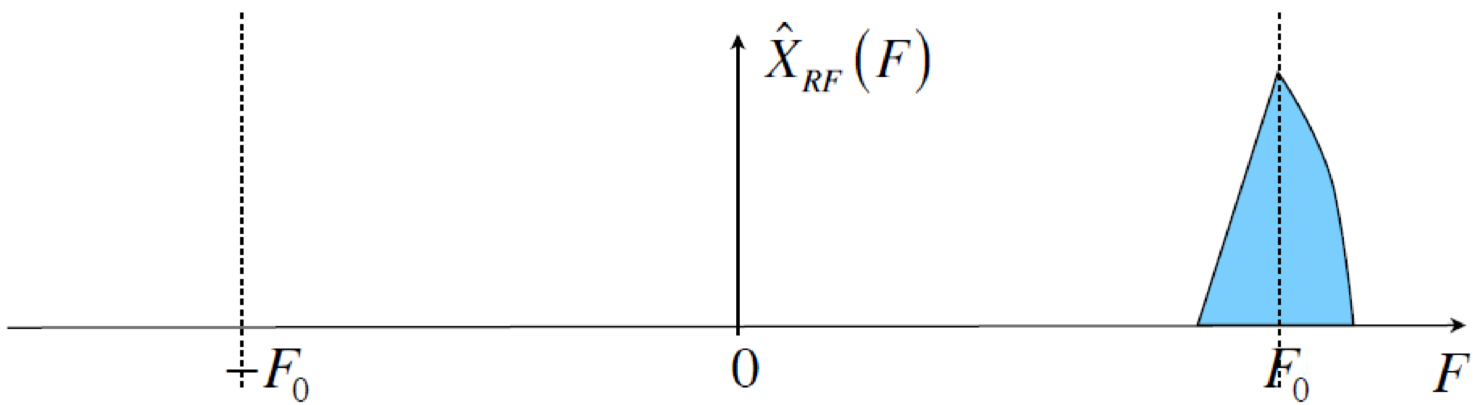


$$x_{RF}(t) = s_{I,a}(t)\cos(2\pi F_0 t) - s_{Q,a}(t)\sin(2\pi F_0 t)$$

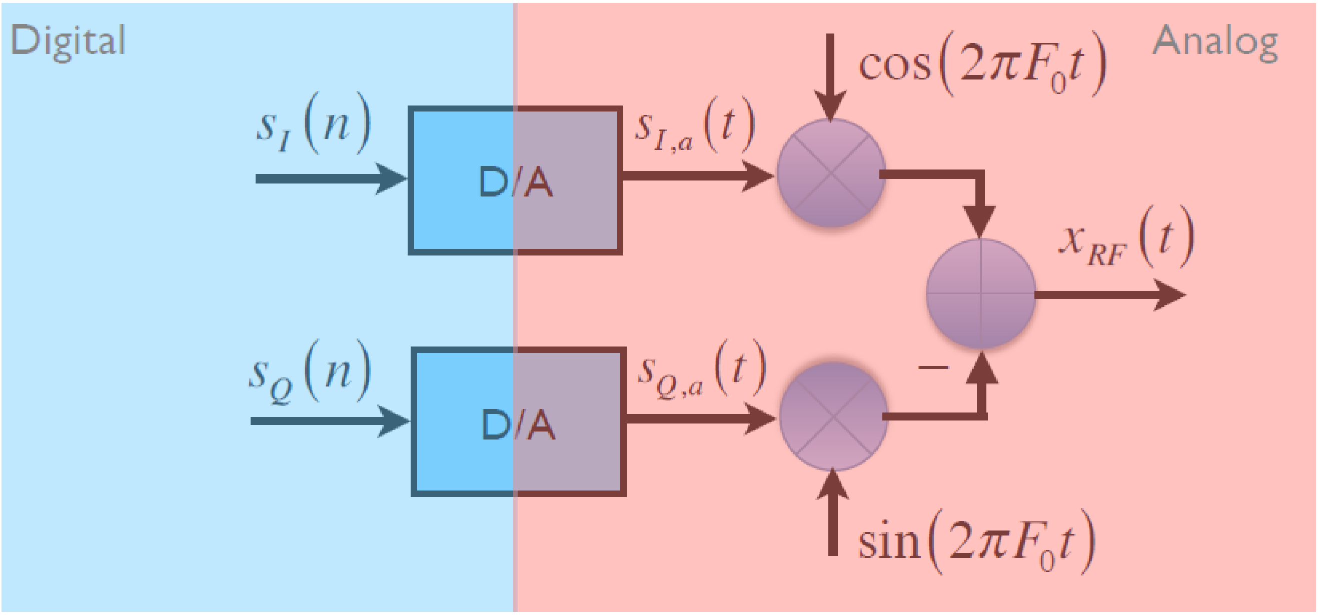
$$x_{RF}(t) = \mathrm{Re}\{\hat{x}_{RF}(t)\} \qquad \hat{x}_{RF}(t) = s_a(t)e^{j2\pi F_0 t}$$

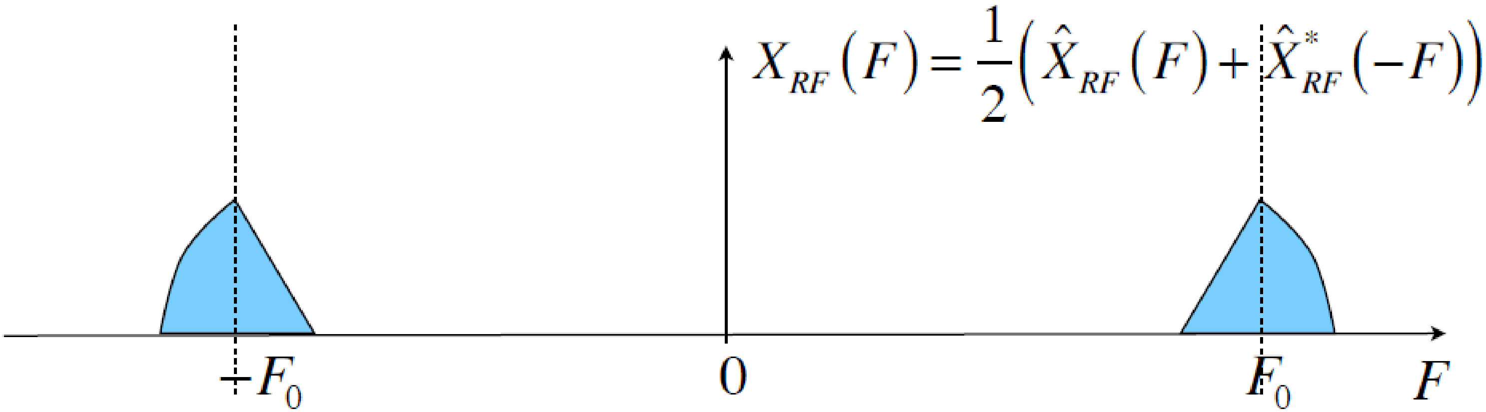
Direct conversion architecture (4)





Direct conversion architecture (5)





Oversampling

- In the pulse-shape filtering stage, an oversampling with N_{sym}
- The typical values of roll-off factor for RC filters are: $\alpha \in [0.25, 0.5]$

$$F_M = \frac{1+\alpha}{T_{sym}} \qquad F_s = \frac{N_{sym}}{T_{sym}}$$

$$F_M = \frac{N_{sym}}{1+\alpha} F_M$$

Oversampling (2)

- Example:

$$\alpha = 0.25 \quad N_{sym} = 8 \quad F_M = 5MHz \quad F_s = 32MHz$$

- In case of the direct conversion architecture:

$$B_t = F_s - 2F_M = 12MHz$$

- The analog filter can be easily implemented
  - For N_{sym}=4 it results B_t=6MHz

Oversampling (3)

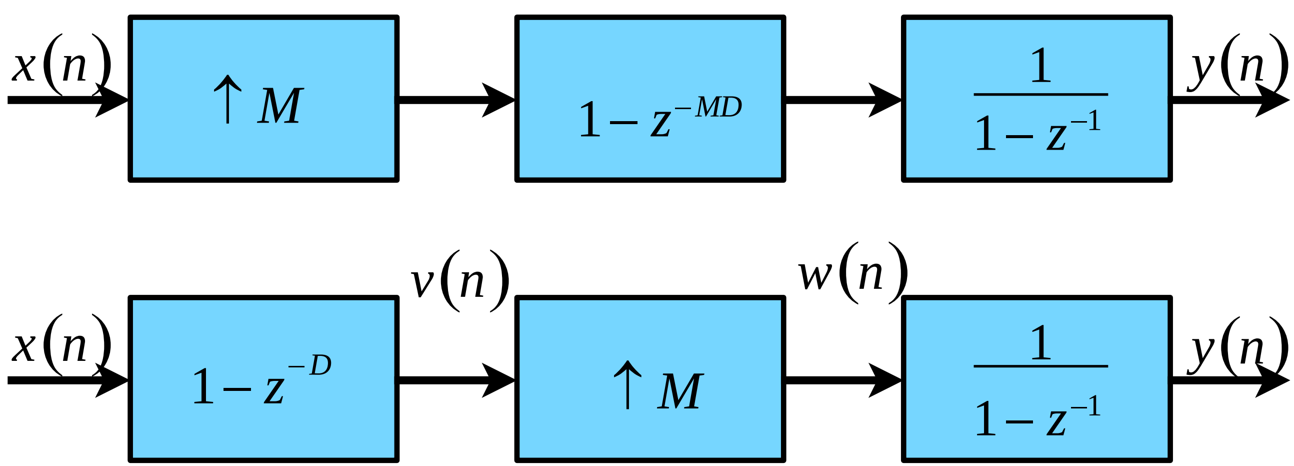
- The RC interpolation filter should be implemented in an efficient way
- Fact: we cannot use only the RC filter for performing the interpolation
- The RC filter implements a minimum necessary interpolation in order to avoid aliasing (ex. N_{sym}=4)
- The interpolation has to be continued in the DFE:

$$L = N_{sym} N_{DFE}$$

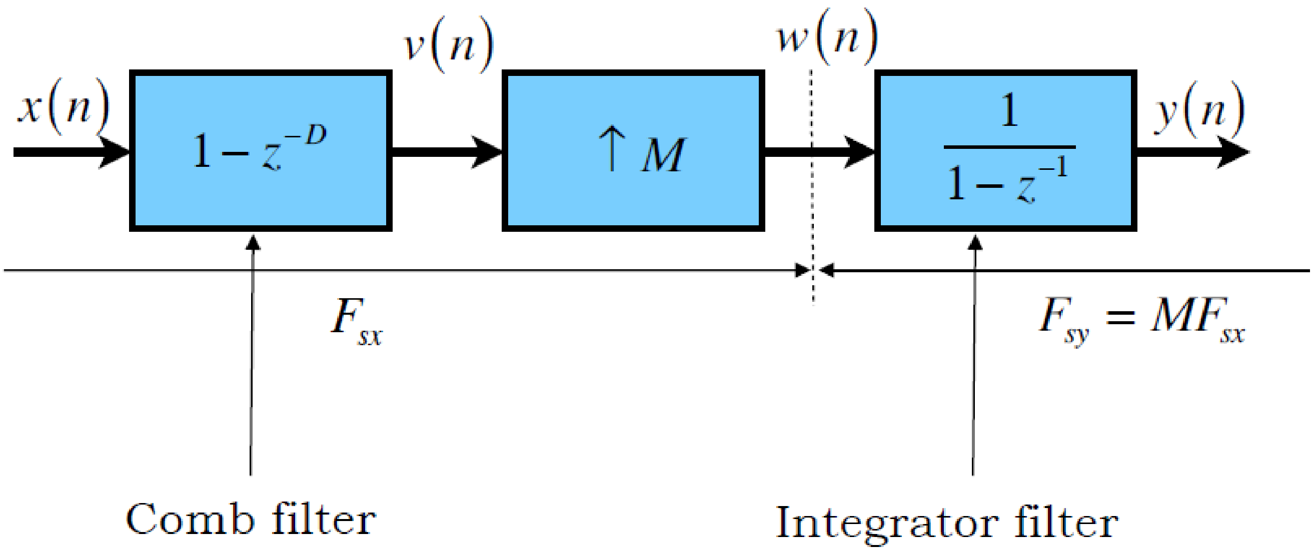
CIC Filters

- CIC = Cascaded Integrator Comb
- Very fast interpolation filters
- Ideal for FPGA implementation

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



CIC Filters (2)



CIC (Cascaded Integrator Comb)

$$v(n) = x(n) - x(n-D)$$

$$w(n) = \begin{cases} v\left(\frac{n}{M}\right), n = pM \\ 0, n \neq pM \end{cases}$$

$$y(n) = y(n-1) + w(n)$$

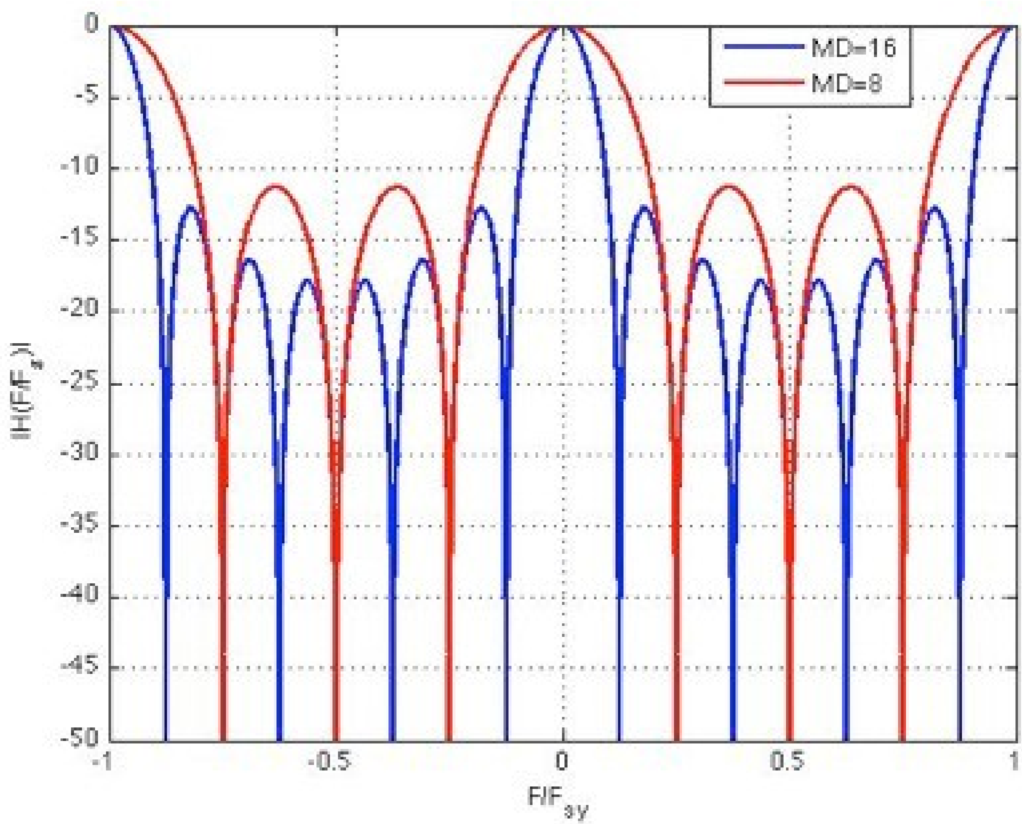
$$y(n) = \begin{cases} y(n-1) + v\left(\frac{n}{M}\right), n = pM \\ y(n-1), n \neq pM \end{cases}$$

CIC Filters (3)

- The necessary operations are very simple
- For each new entry, at most two additions are being made
- The operations which depend on the interpolation order are:
  - Storing D samples using circular addressing
  - Counting M samples for updating the computing of *y(n)*
- Both of the above operations are **programmable**
- The same physical entity can be programmed to implement different interpolation orders

CIC Filter Frequency Response

- Example: M=2, D=8 or D=4

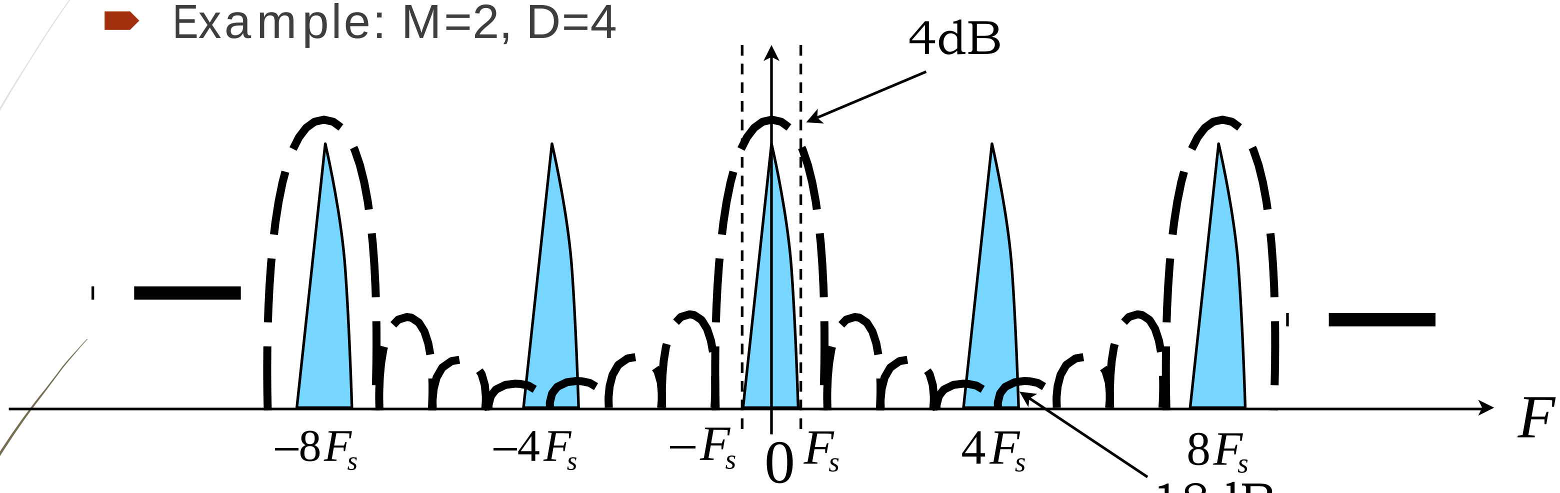


CIC Filter Frequency Response (2)

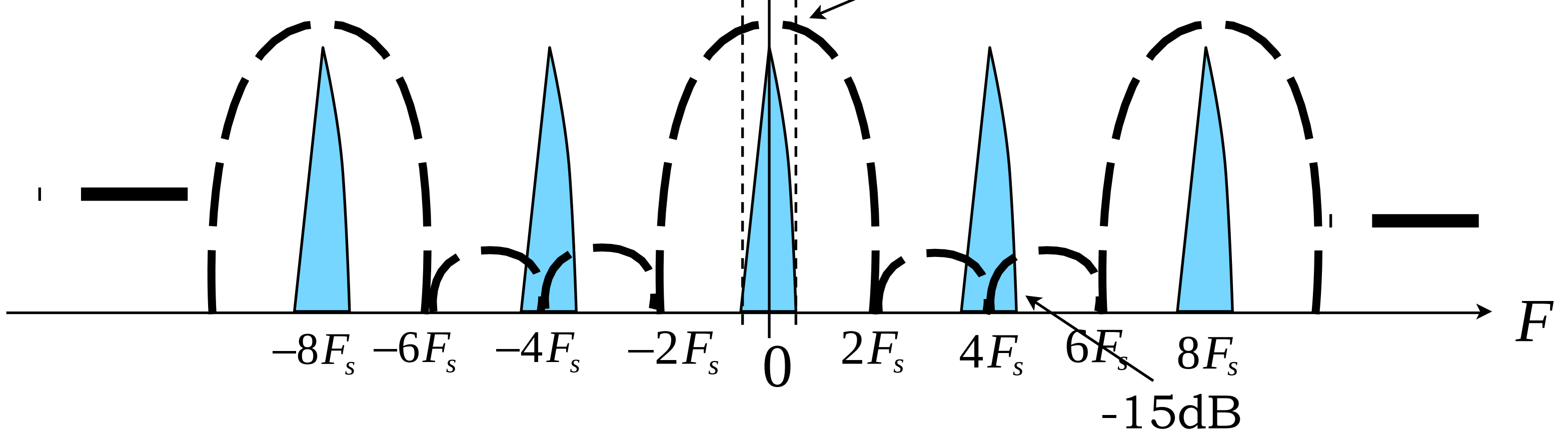
- Significant side lobes
  - The images can be insufficient attenuated
  - *D* has to take high values
- Steep fall around the maximum
  - Significant attenuation in the passband
  - The passband has to be narrow, in order for the characteristic to be as flat as possible
  - Passband distortions
  - *D* has to take small values

CIC Filter Frequency Response (3)

- Example: M=2, D=4



- Example: M=2, D=2



CIC Filter Frequency Response (4)

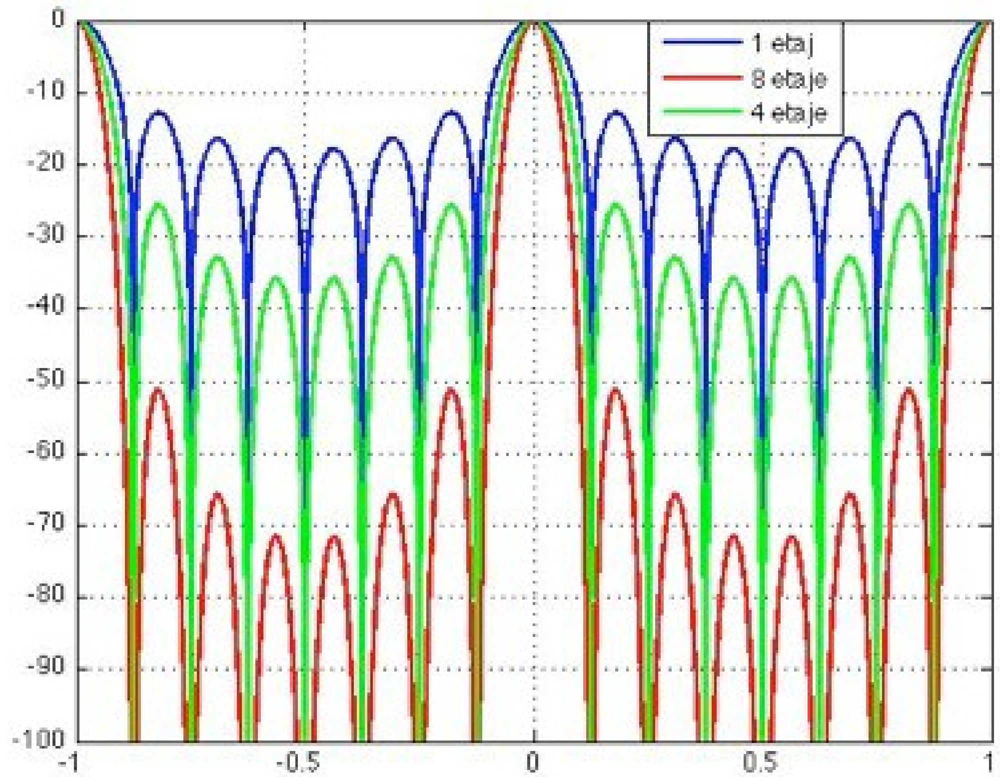
- The upper limit of the *D* parameter is the bandwidth occupied by the system

$$D \le \frac{F_{sy}}{2MKF_M}$$

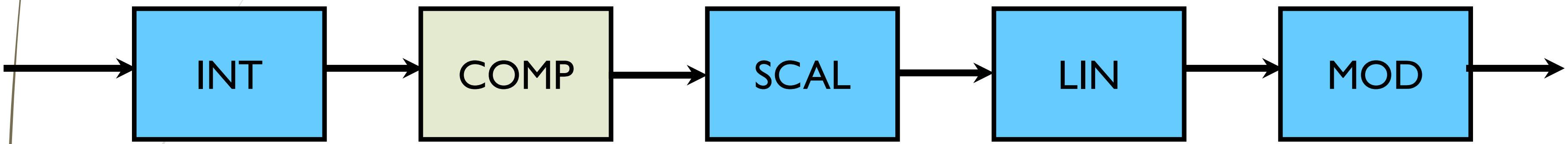
- *K* is a safety factor, typically of value 1 or 2
  - The highest the *K* value, a lower part of the passband is affected by the CIC filter characteristic
- For rejecting the images, several stages can be used

CIC Filter Frequency Response (5)

- Example: MD=16



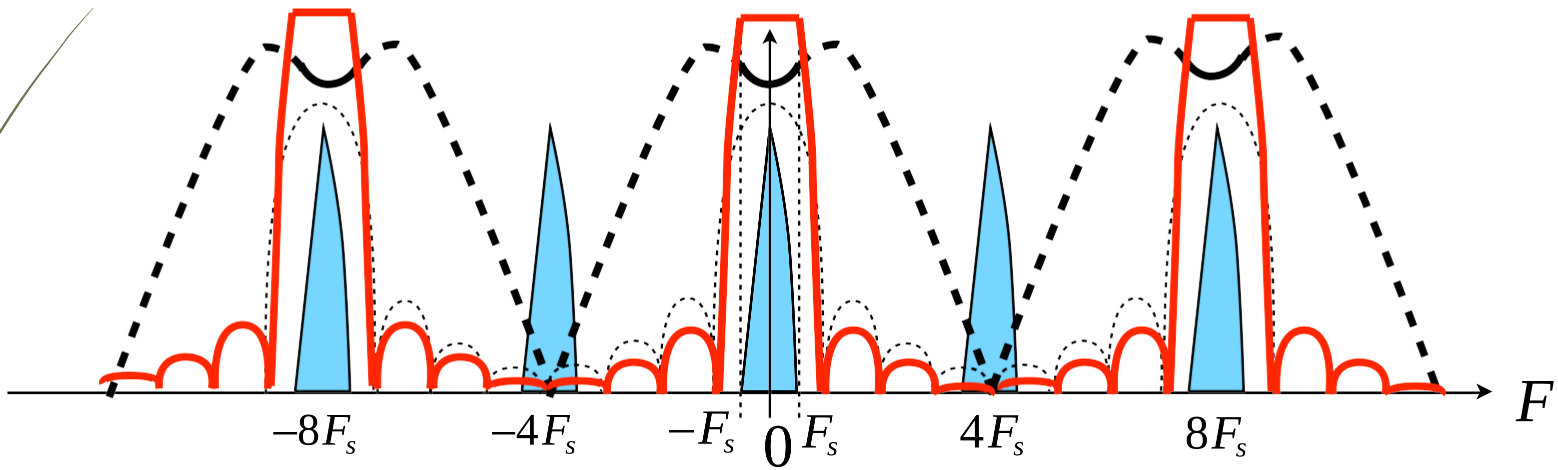
Frequency compensation



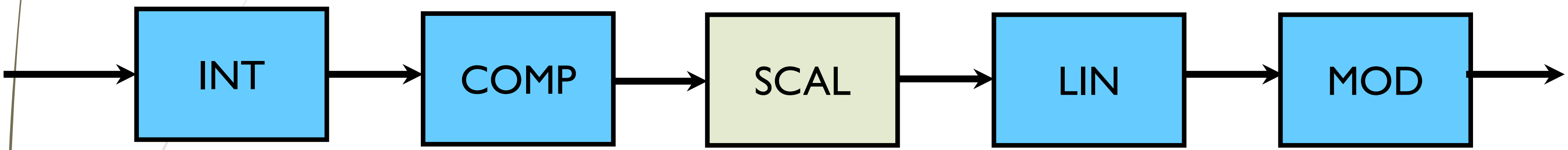
- Frequency compensation of the CIC filter characteristic
- Frequency compensation of the D/A converter characteristic
- The COMP block can be located in front of the INT block (in order to work at a lower sampling frequency)

CIC Filter Inversion

- Is done using a filter whose characteristic in the passband compensates the characteristic of the CIC filter

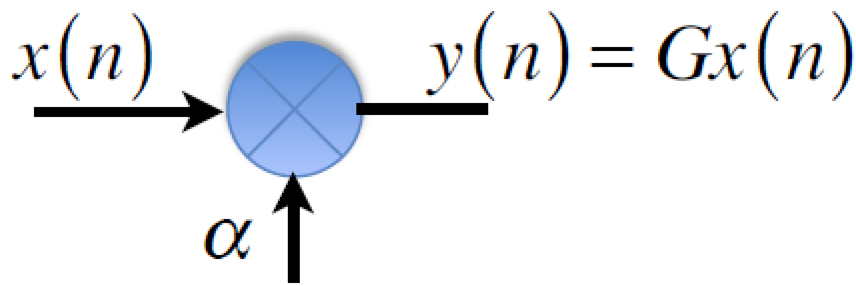


Power level scaling



- Usage of the whole dynamic range of the DAC for an efficient operation and for minimizing the quantization noise
- The maximum desired peak power shouldn’t saturate the DAC.

Power level scaling (2)



$$P_y = \alpha^2 P_x = GP_x \qquad [P_y] = [G] + [P_x]$$

- Power gain can be configured
- The variation resolution can be adjusted depending on the number of bits used for representing *G*.

Power level scaling (3)

- Simple cases:
  - Amplification/attenuation that represents a left/right shift with a certain number of bits
  - The power gain can be adjusted in 6dB steps
- Terms that are used for defining signal power:
  - Peak power
  - Average power
  - Both depend on the signal constellation

PAPR ratio

- PAPR: Peak to Average Power Ratio

$$PAPR = \frac{P_v}{P_m} \qquad [PAPR] = [P_v] - [P_m]$$

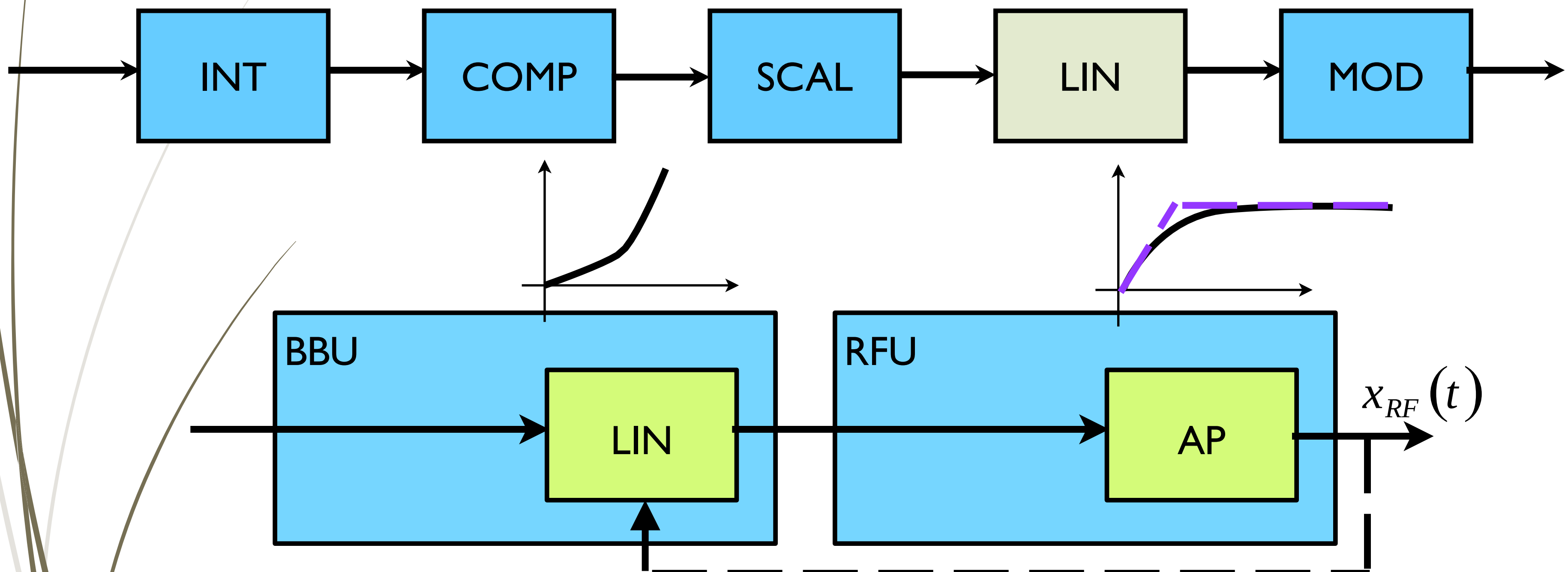
- Depends on the used digital modulation
- Adds a constraint on the power gain G
- The gain should be chosen so that the peak power doesn't saturate the DAC

$$P_m = P_y = \frac{P_v}{PAPR} \leq \frac{1}{PAPR} \qquad G \leq \frac{1}{P_x PAPR}$$

- Example: P_x=0.23W, $[PAPR]_{max}$=8dB

$$[G] = -[P_x] - [PAPR] = 6.38 - 8 = -1.62dB \qquad \alpha = 10^{\frac{G}{20}} = 0.83$$

Power Amplifier Linearization



- Extending the area where the power amplifier characteristic is linear

Digital Mixing

- Shift the digital spectrum of the signal

$$x(n) = s(n)e^{jnw_0}$$

- Appears only in case of the digital IF architecture
- The key part is to generate the function

$$f(n) = e^{jnw_0} = \cos(nw_0) + j\sin(nw_0)$$