A Mixed-Domain Modeling Method for RF Systems

Zhimiao Chen

Outline

- Motivation
- Baseband modeling method
  - Traditional baseband method
  - Extended baseband method
- Frequency Response for baseband
  - shifted transfer function
  - based on sinc base function
  - based on FFT

Motivation: Need for Speed

- Analog circuitry: 2% transistors, 20% area, but 40% design effort, 50% re-spins [1]
- Low efficient SPICE-like simulator to analyze analog circuits including RF circuits

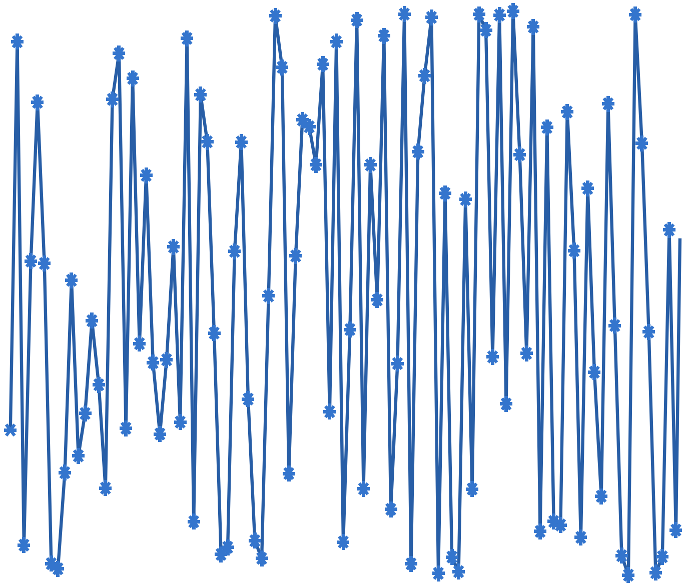
	transistors	max. freq.	tran. sim.	time cost
LNA	*~500*	*800MHz*	*10us*	*480 s*
DSM-ADC	*~2000*	*96MHz*	*10us*	*360 s*
RF front-end	*~5000*	*2.4GHz*	*10us*	*~2 hr*
PLL	*~3000*	*3.2GHz*	*10us*	*~35 hr*

- SoC simulation requires high simulation speed

[1] Cadence Design Systems, "The rise of digital/mixed-signal semiconductors and system-on-a-chip", white paper, oct. 17. 2002

Motivation: Need for Speed

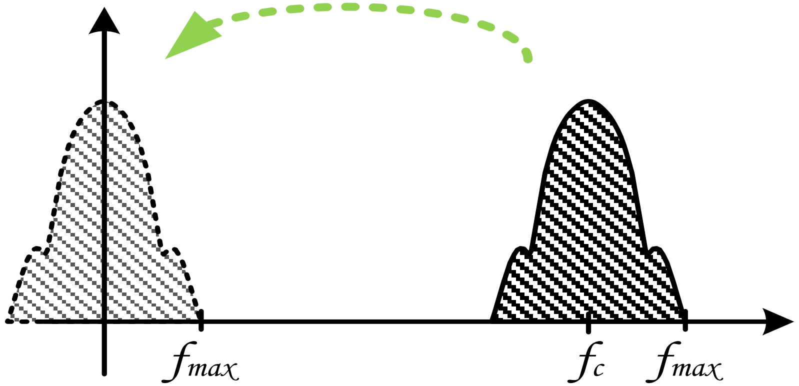
- Transient analysis for RF circuits
  - Small time step to capture high frequent signals
  - Huge size of ODE matrix
  - Convergent problem
- Other RF simulation methods
  - Fast SPICE
  - PSS, harmonic balance
  - HDL behavioral modeling



$Fs >> 2f_{max}$

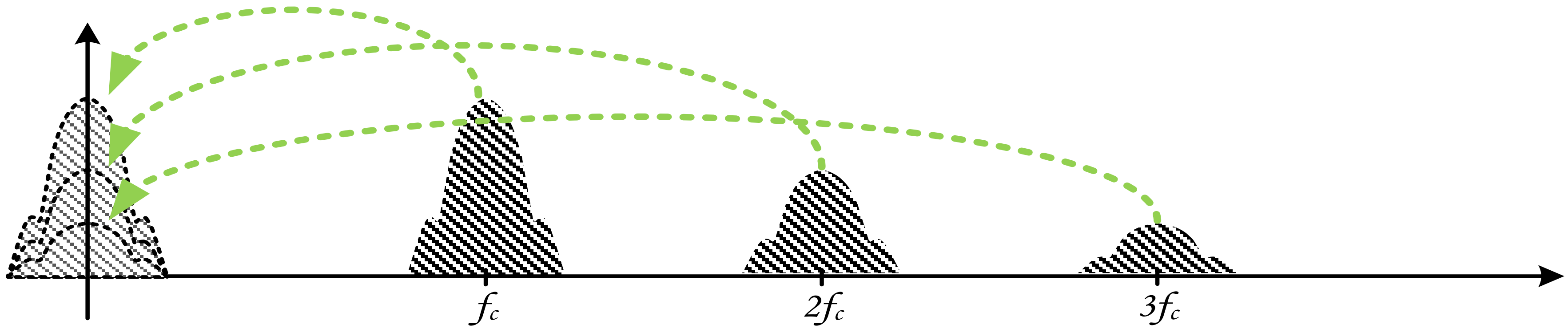
Traditional Baseband Modeling Method

- $s(t) = I(t)\cos(2\pi f_c t) - Q(t)\sin(2\pi f_c t)$;
- Benefits
  - Fast simulation: relative low frequent I(t), Q(t)
  - Easy processing for frequency conversion behavior
- Limitations
  - Ignore carrier distortion
  - limit bandwidth of I(t)/Q(t)
  - Increase modeling effort
  - Pin compatible issue

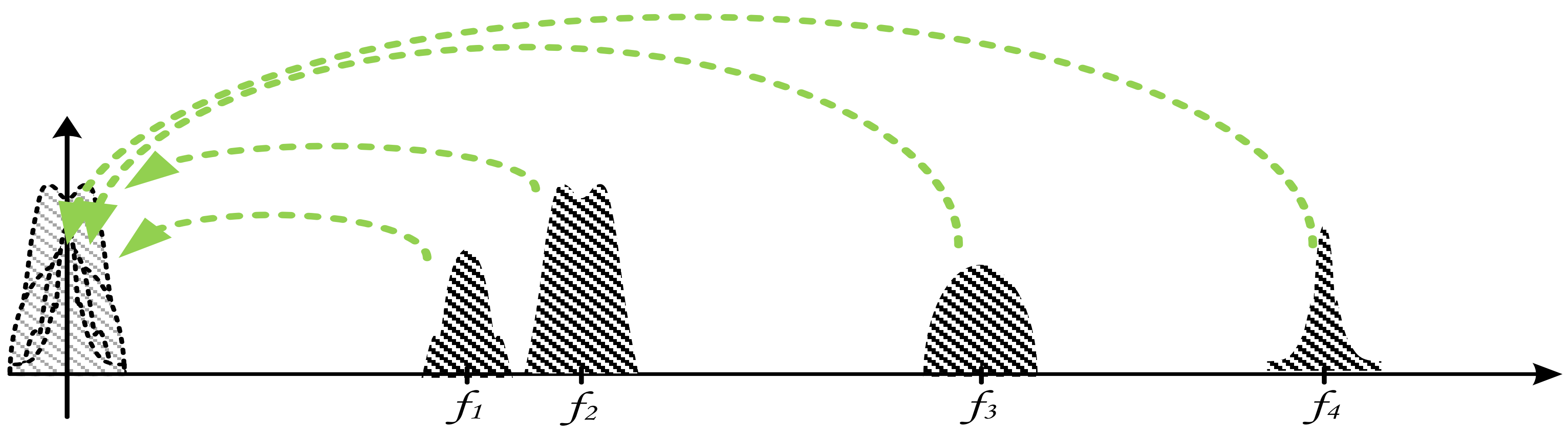


Extended baseband modeling method

- harmonic balance

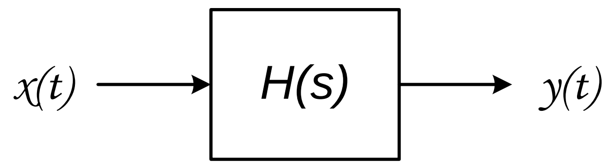


- arbitrary carrier frequency



Frequency Response

- Laplace transfer function



- time-domain response, event-driven[2]

$$H(s)=\sum_{l=1}^{L}\frac{b_l}{s+a_l} \qquad \frac{b_l}{s+a_l}\Leftrightarrow\begin{cases}a_l=0, & b_l\cdot u(t)\\ a_l\neq 0, & b_l e^{-a_l t}u(t)\end{cases}$$

$$\Rightarrow\begin{cases}y(t)=b_l\int_0^t x(\tau)d\tau\\ y(t)=b_l\int_0^t e^{-a_l(t-\tau)}x(\tau)d\tau\end{cases}$$

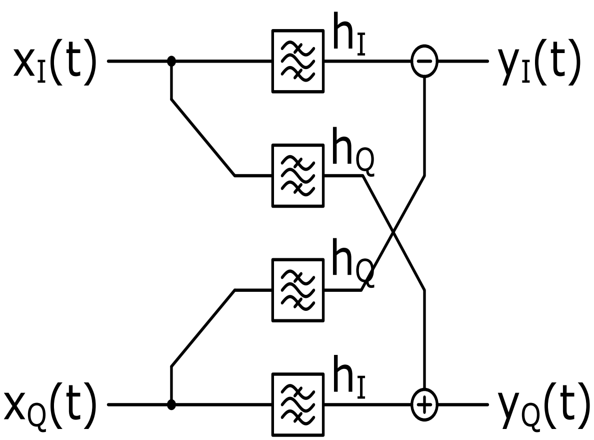
$$\Rightarrow\begin{cases}y(t_k)=y(t_{k-1})+b_l x(t_{k-1})(t_k-t_{k-1})\\ y(t_k)=e^{-a_l(t_k-t_{k-1})}y(t_{k-1})+\dfrac{b_l}{a_l}(1-e^{-a_l(t_k-t_{k-1})})x(t_{k-1})\end{cases}$$

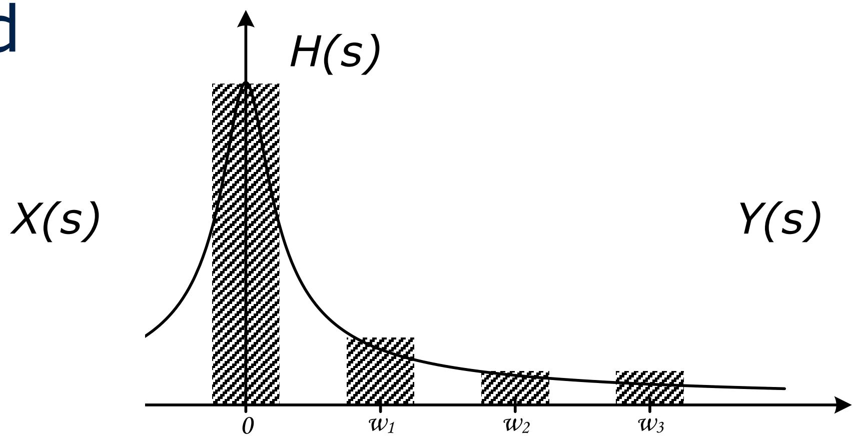
[2] J.-E. Jang and et al, “True event-driven simulation of analog/mixed-signal behaviors in SystemVerilog: A decision-feedback equalizing (DFE) receiver example,” in Custom Integrated Circuits Conference (CICC), 2012 IEEE, 2012, pp. 1–4.

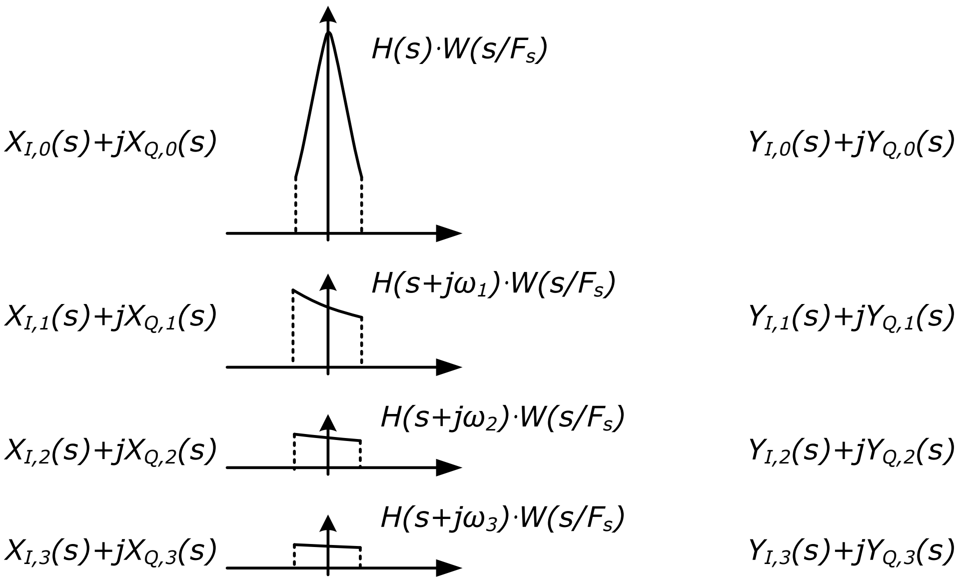
Frequency Response for baseband

- Complex filter for baseband

$$
\begin{aligned}
Y_I + jY_Q &= (H_I + jH_Q)\cdot(X_I + jX_Q) \\
&= (X_I \cdot H_I - X_Q \cdot H_Q) \\
&\quad + j(X_I \cdot H_Q - X_Q \cdot H_I)
\end{aligned}
$$







Frequency Response for baseband

- Event-driven realization

$$H(s+j\omega_c)\cdot W(\frac{s}{F_s}) = \sum_{l=1}^{L} \frac{b_l}{s+j\omega_c+a_l}\cdot W(\frac{s}{F_s})$$

$$\frac{b_l}{s+j\omega_c+a_l}\cdot W(\frac{s}{F_s}) \Leftrightarrow b_l e^{-(a_l+j\omega_c)t}u(t) * \frac{\sin(F_s t/2)}{F_s t/2}$$

$$y(t) = b_l e^{-(a_l+j\omega_c)t}u(t) * \frac{\sin(F_s t/2)}{F_s t/2} * x(t)$$

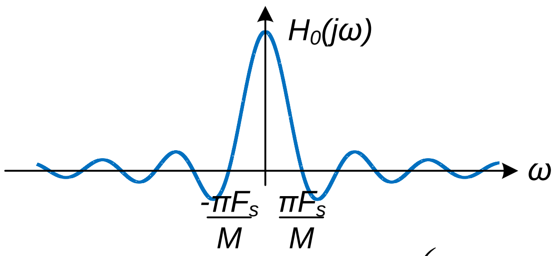
$$\Rightarrow y(t) = b_l e^{-(a_l+j\omega_c)t}\int_0^t e^{(a_l+j\omega_c)\lambda}\left(\int_0^{t-\lambda} e^{(a_l+j\omega_c)\tau}\frac{\sin(F_s\tau/2)}{F_s\tau/2}d\tau\right)x(\lambda)d\lambda$$

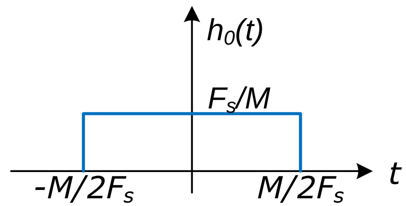
$$\Rightarrow y(t_k) = y(t_{k-1})e^{-(a_l+j\omega_c)(t_k-t_{k-1})}+\ldots$$

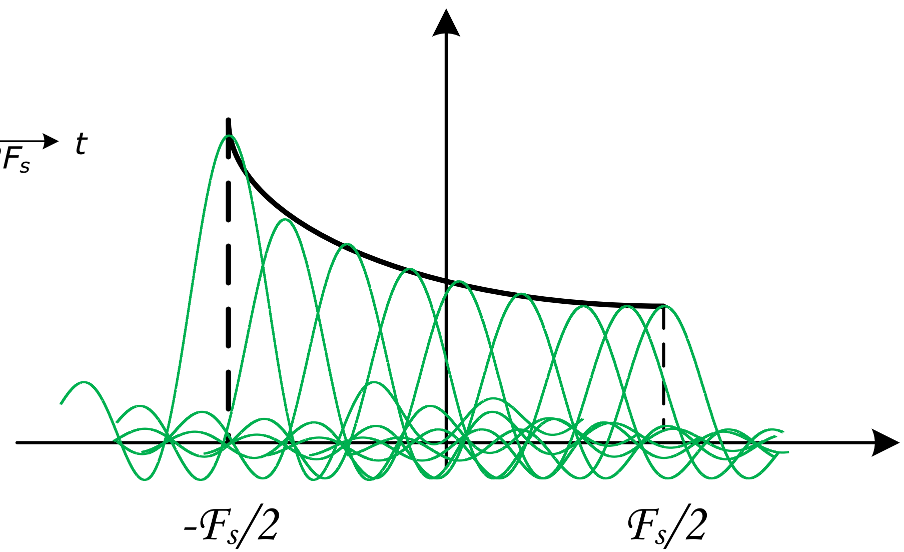
- too complex derivation
- potentially unstable with extra poles

Frequency Response for baseband

- Based on sinc base function







$$H(j\omega) = \sum_{m=-M}^{M} A_m \cdot \sin c\left(\frac{\frac{M}{F_s}(\omega - m\frac{\pi F_s}{M})}{2} \right)$$

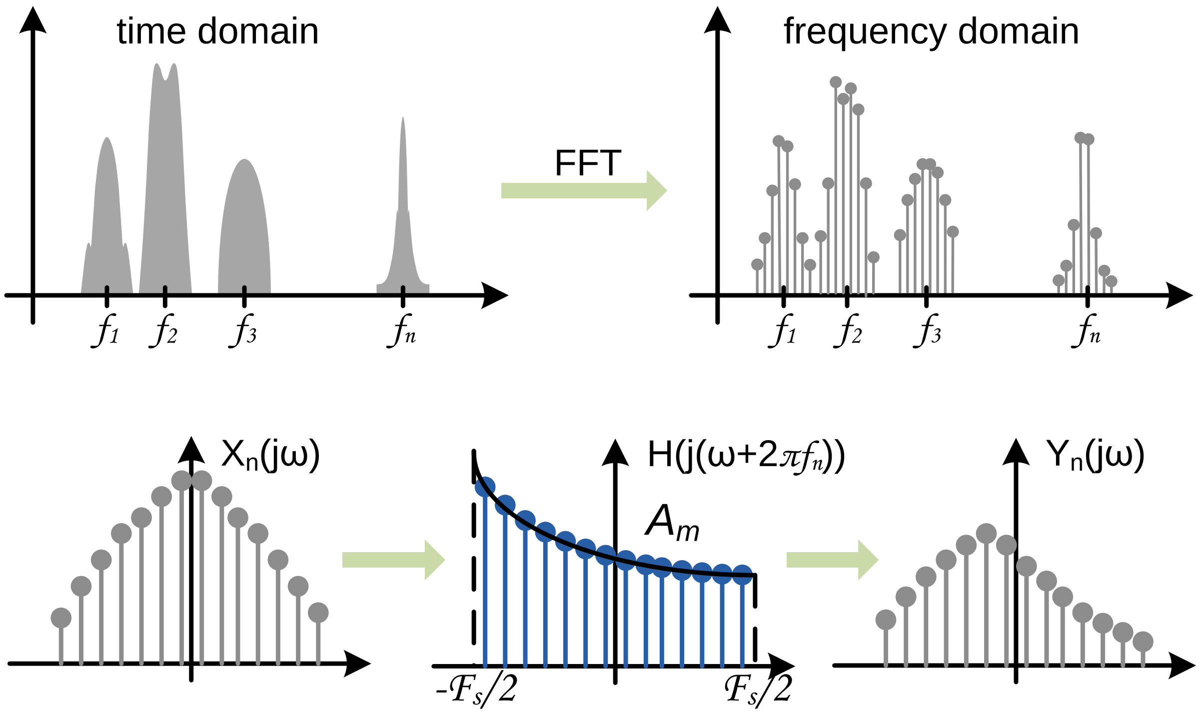
$$h(t) = \sum_{m=-M}^{M} A_m h_0(t) e^{j\frac{m\pi}{M}F_S t}$$

$$y(t) = h(t) * x(t) = \sum_{m=-M}^{M} A_m \cdot \frac{F_s}{M} \cdot e^{j\frac{m\pi}{M}F_S t} \cdot \int_{t-MF_S/2}^{t+MF_S/2} e^{-j\frac{m\pi}{M}F_S \tau} x(\tau)d\tau$$

$$y(kT_s) = \sum_{m=-M}^{M} \frac{A_m}{M} \cdot e^{-j\frac{m\pi}{2M}} \frac{\sin(\frac{m\pi}{2M})}{\frac{m\pi}{2M}} \cdot \sum_{l=k-M/2}^{k+M/2-1} x(lT_s) e^{j\frac{m\pi}{M}(k-l)}$$

Frequency Response for baseband

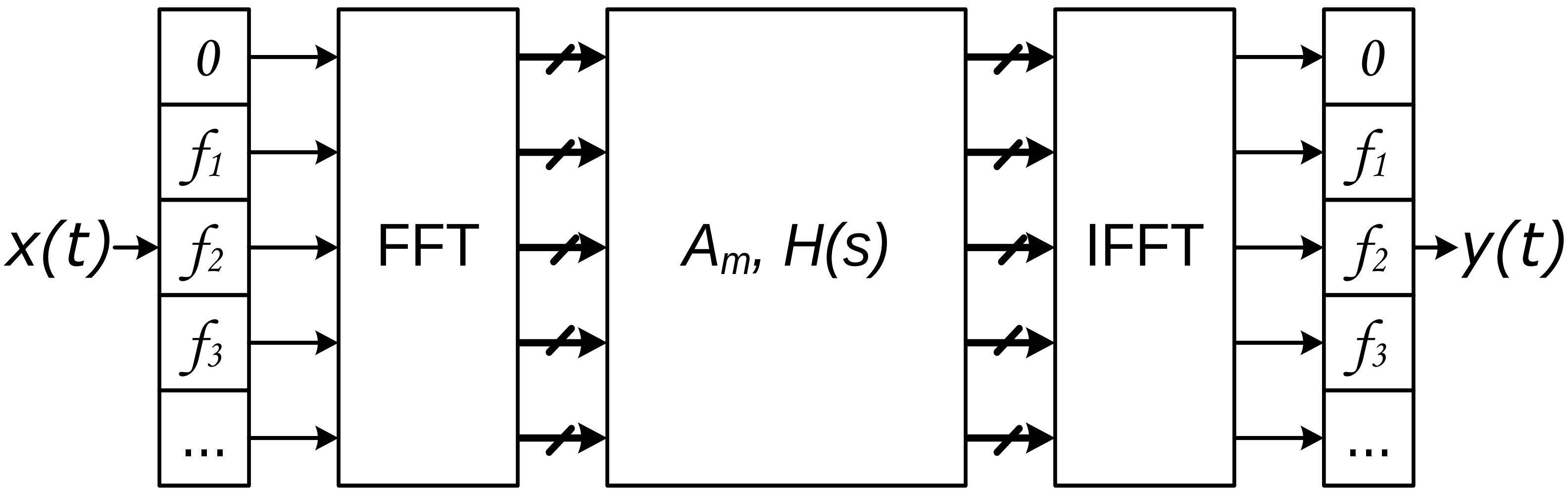
- Based on FFT



- FFT size N affects simulation accuracy/speed
- resolution bandwidth = F_s/N

Mixed-domain modeling method

- using FFT/IFFT to bridge time domain/frequency domain processing



- regarding passband as baseband signal with $f_c=0$

Implementation in SystemC

- user defined data type

TS_sig
- time_domain_valid: *bool*
- freq_domain_valid: *bool*
- bb_sig_vector: *vector<bb_sig>*
  - bb_sig: <f_c *I Q*>
- spectral_sig_vector: *vector<spectral_sig>*
  - spectral_sig: <f_c *N vector<complex>A*>

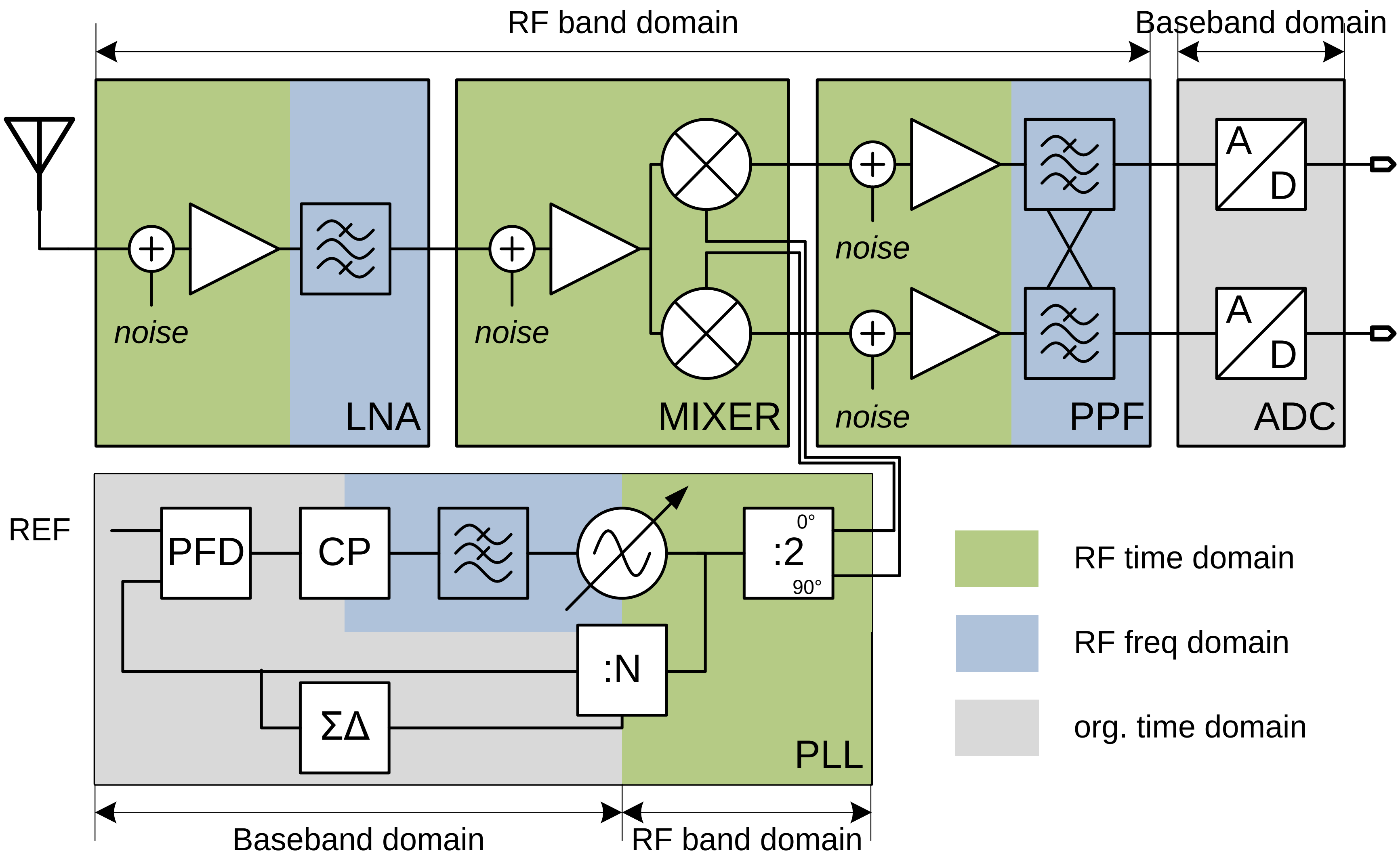
- flag signals switch time domain/frequency domain
- conversion between TS_sig and double

Implementation in SystemC

- operators overloading
  - reloading basic algorithmic & logic operators
  - redefining basic functions
  - same behavioral description for different modeling methods

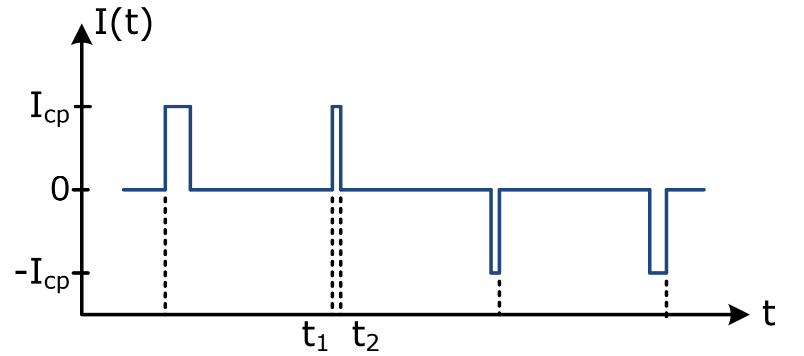
- template class for port data type
  - switch modeling methods by switching data type

Testcase: RF Front-end with PLL



Testcase: RF Front-end with PLL

- Charge Pump
  - Fourier series expansion

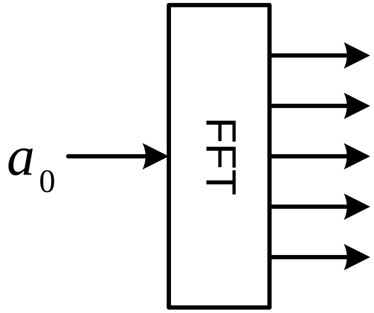


$$I(t) = a_0 + \sum_n a_n \cos(2\pi n f_c t) - b_n \sin(2\pi n f_c t)$$

$$a_0 = \frac{1}{T}\int_0^T I(t)dt = \frac{1}{T}\int_{t_1}^{t_2} I_{cp} dt = I_{cp}\frac{(t_2 - t_1)}{T}$$

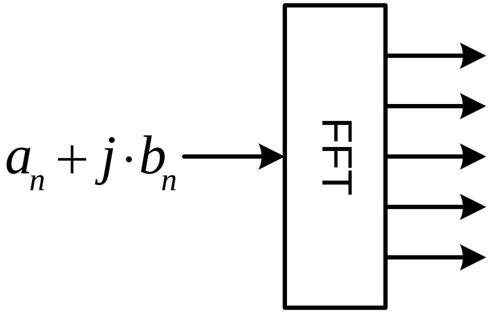
$$a_n = \frac{1}{T}\int_0^T I(t)\cdot\cos(2\pi n f_c t)dt = \frac{1}{T}\int_{t_1}^{t_2} I_{cp}\cdot\cos(2\pi n f_c t)dt$$

$$= \frac{I_{cp}\cdot(\sin(2\pi n f_c t_2) - \sin(2\pi n f_c t_1))}{2\pi n}$$



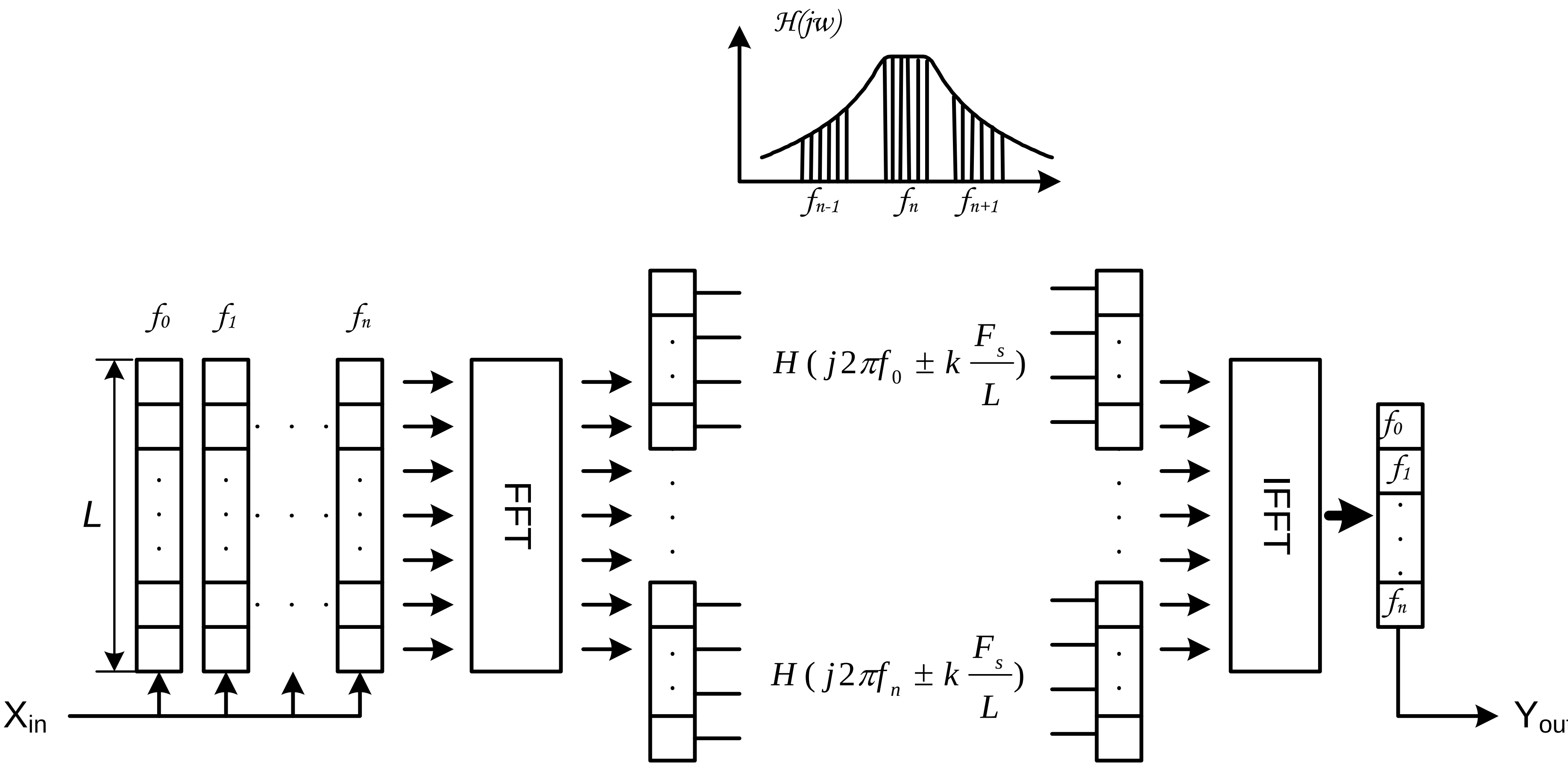
$$b_n = \frac{1}{T}\int_0^T I(t)\cdot\sin(2\pi n f_c t)dt = \frac{1}{T}\int_{t_1}^{t_2} I_{cp}\cdot\sin(2\pi n f_c t)dt$$

$$= \frac{I_{cp}\cdot(\cos(2\pi n f_c t_1) - \cos(2\pi n f_c t_2))}{2\pi n}$$

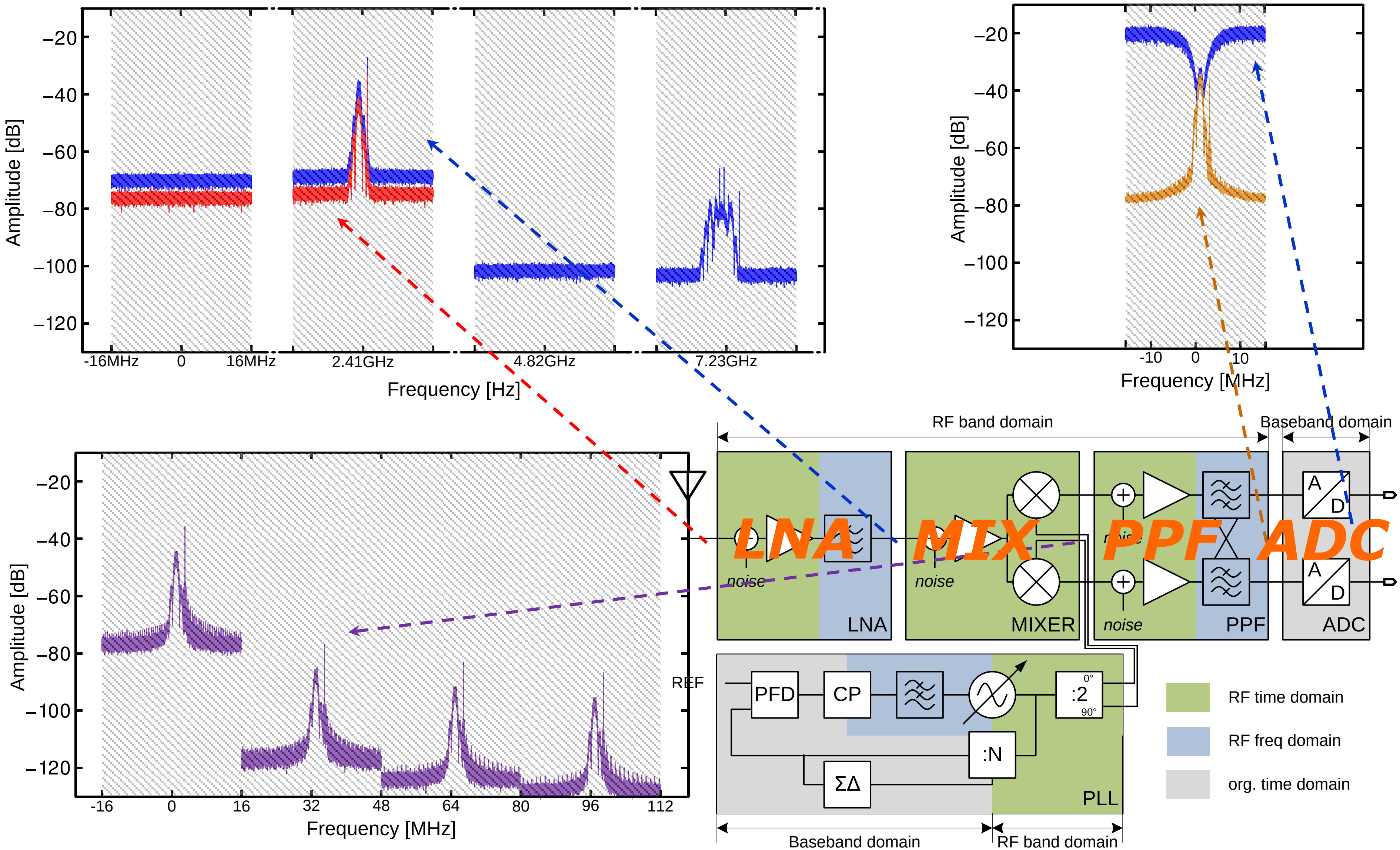


Testcase: RF Front-end with PLL

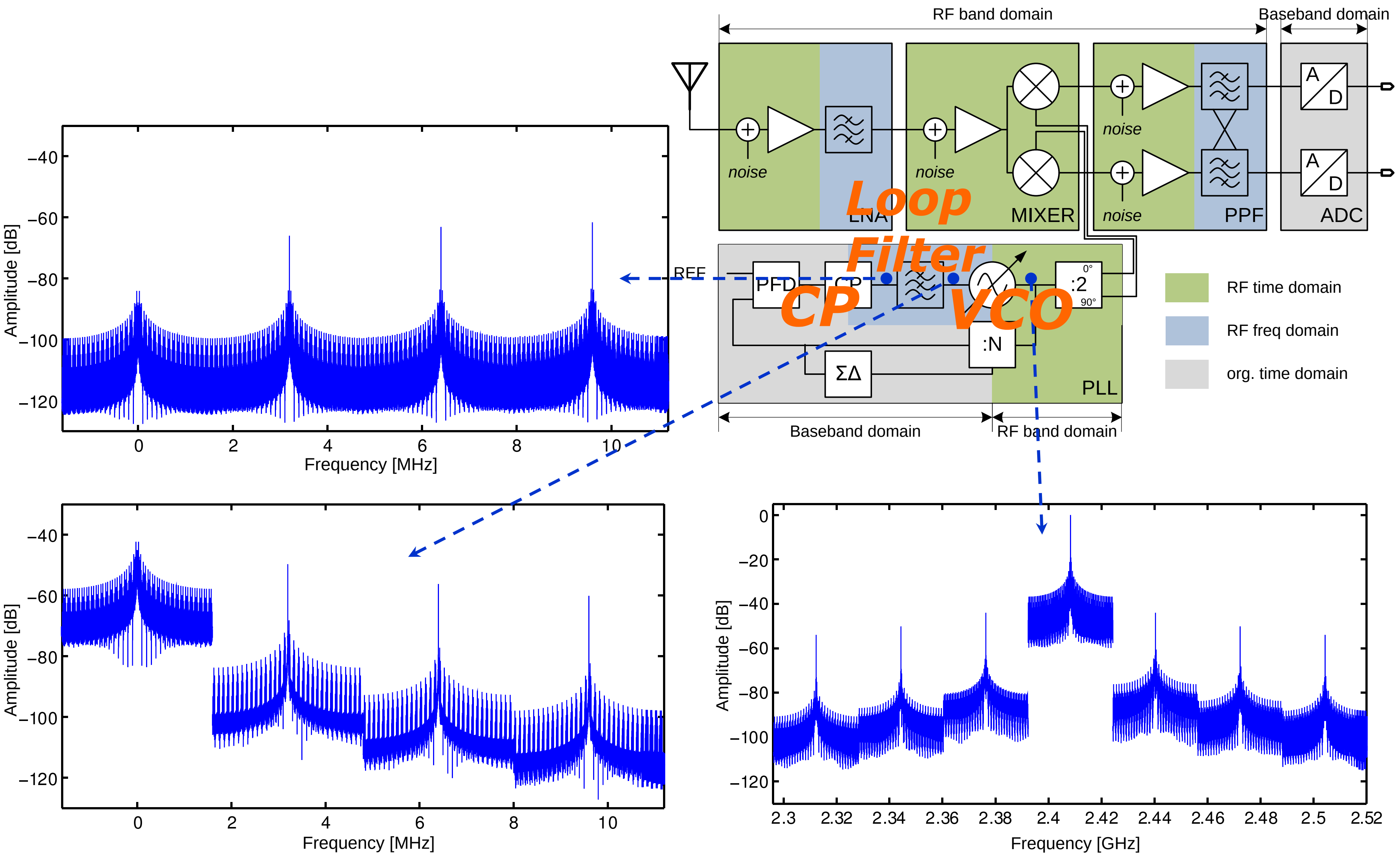
- Bandpass filter inside LNA



Simulation Results

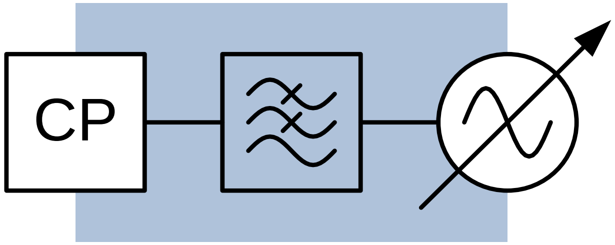


Simulation Results



Simulation Results

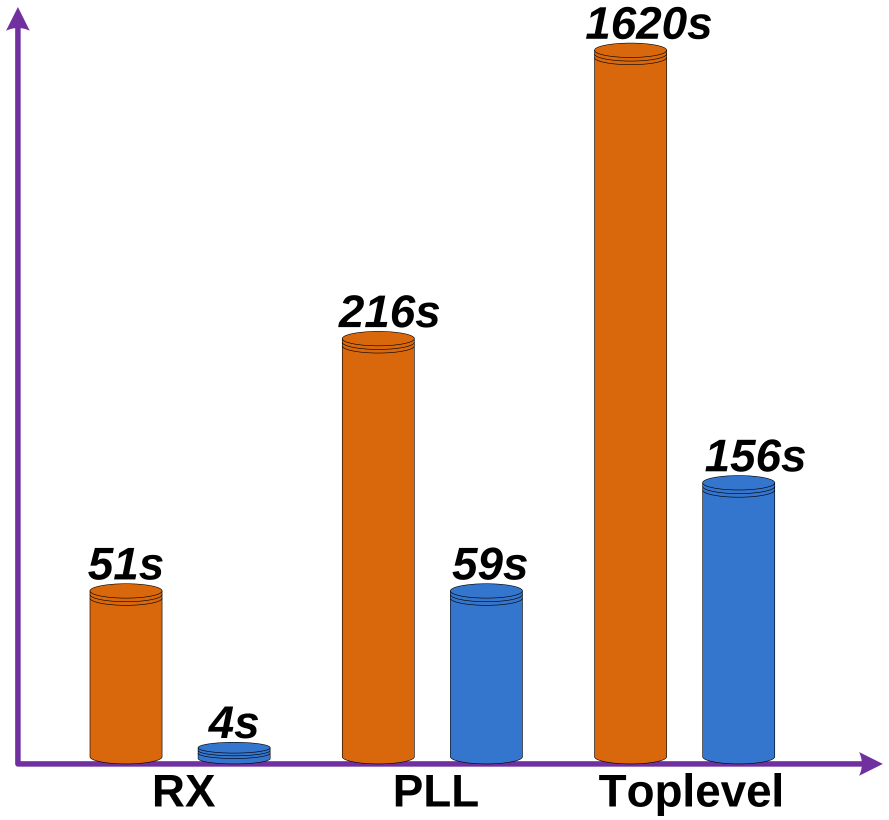
- FFT size, simulation speed, accuracy
- Loop filter inside PLL
- Run 1ms simulation to compare simulation speed



	Fs = 32MHz	Fs = 16MHz	Fs = 8MHz
N_FFT = *128*	*40.96 s*	*20.17 s*	*10.17 s*
N_FFT = *64*	*22.36 s*	*11.10 s*	*5.62 s*
N_FFT = *32*	*12.76 s*	*6.34 s*	*3.20 s*
N_FFT = *16*	*7.87 s*	*3.87 s*	*1.98 s*

Simulation Results

- simulation speed compare
- passband/mixed-domain
- around 10x speed up



Thanks for your attention!
Any Questions?