



**HAL**  
open science

# Design of a Radio Frequency Front-End Receiver dedicated to Software-Radio for Mobile Terminals

Francois Rivet

► **To cite this version:**

Francois Rivet. Design of a Radio Frequency Front-End Receiver dedicated to Software-Radio for Mobile Terminals. Micro and nanotechnologies/Microelectronics. Université Sciences et Technologies - Bordeaux I, 2009. English. NNT: . tel-00412242

**HAL Id: tel-00412242**

**<https://theses.hal.science/tel-00412242>**

Submitted on 1 Sep 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : 3811

# THÈSE

présentée à

## L'UNIVERSITÉ BORDEAUX 1

Ecole doctorale des Sciences Physiques et de l'Ingénieur

par **François RIVET**

POUR OBTENIR LE GRADE DE

## DOCTEUR

### SPÉCIALITÉ : ÉLECTRONIQUE

---

CONTRIBUTION À L'ÉTUDE ET À LA RÉALISATION D'UN FRONTAL RADIOFRÉQUENCE  
ANALOGIQUE EN TEMPS DISCRETS POUR LA RADIO-LOGICIELLE INTÉGRALE

---

Soutenue le : 19 Juin 2009

Après avis de :

<b>M. Edgar SÁNCHEZ-SINENCIO</b>	Professeur	Texas A&M University	Rapporteur
<b>Patrice GAMAND</b>	HDR	NXP Semiconductors	Rapporteur

Devant la commission d'examen formée de :

<b>M. Jean-Baptiste BÉGUERET</b>	Professeur	Université Bordeaux 1	Co-directeur de thèse
<b>Didier BELOT</b>	Expert	ST Microelectronics	Industriel
<b>Philippe CATHELIN</b>	Ingénieur	ST Microelectronics	Industriel
<b>Dominique DALLET</b>	Professeur	ENSEIRB Bordeaux	Président
<b>Yann DEVAL</b>	Professeur	ENSEIRB Bordeaux	Directeur de thèse
<b>Patrice GAMAND</b>	HDR	NXP Semiconductors	Rapporteur
<b>Edgar SÁNCHEZ-SINENCIO</b>	Professeur	Texas A&M University	Rapporteur



*"Dans le futur, vous serez trop occupés à regarder le téléphone pour répondre à la télévision."*

# Remerciements

*A mes parents,  
à ma famille,  
à mes amis.*

*A Yann, Jean-Baptiste et Dominique,  
à l'équipe Conception de Circuits,  
à l'équipe Circuits et Systèmes Hyperfréquences.*

*Aux membres du laboratoire IMS.*

# Contents

<b>List of Abbreviations</b>	<b>17</b>
<b>List of Notations</b>	<b>19</b>
<b>Introduction</b>	<b>21</b>
<b>1 The Software Radio Concept</b>	<b>23</b>
1.1 Wireless communication systems . . . . .	24
1.1.1 Wireless communication architectures . . . . .	25
1.1.2 Wireless communication standards . . . . .	27
1.1.3 Wireless communication market and trends . . . . .	28
1.1.4 Conclusion . . . . .	30
1.2 Software Radio Background . . . . .	30
1.2.1 Definition . . . . .	30
1.2.2 History . . . . .	31
1.2.3 Software Radio Characteristics . . . . .	33
1.2.4 Technological Bottlenecks . . . . .	35
1.2.5 Software Defined Radio Architectures . . . . .	40
1.3 Analog Signal Processing . . . . .	44
1.4 Conclusion . . . . .	49
<b>2 Sampled Analog Signal Processor</b>	<b>51</b>
2.1 Principle . . . . .	52
2.1.1 Analog Signal Processor Principle . . . . .	52
2.1.2 Frequency Translation . . . . .	53
2.1.3 A Fourier Transform . . . . .	56
2.2 A Fast Fourier Transform . . . . .	57
2.2.1 The Cooley-Tukey algorithm . . . . .	57
2.2.2 A pipelined DFT . . . . .	60

2.3	Architecture . . . . .	67
2.3.1	Signal pre-processing . . . . .	67
2.3.2	DFT implementation . . . . .	73
2.3.3	Post-signal processing . . . . .	76
2.4	A Software Radio System . . . . .	78
2.4.1	Concurrent reception . . . . .	78
2.4.2	Frequency demodulation . . . . .	79
2.5	Conclusion . . . . .	82
<b>3</b>	<b>Schematics and Modeling results</b>	<b>83</b>
3.1	Discrete Analog Operations . . . . .	84
3.1.1	Accumulation Delay Line . . . . .	88
3.1.2	Matrix Unit . . . . .	95
3.1.3	Weighting Unit . . . . .	98
3.2	Digital Instructions . . . . .	104
3.2.1	A base-4 algorithm clock generation . . . . .	104
3.2.2	A hardware-implemented algorithm . . . . .	108
3.3	Design - <i>SASPEPA</i> and <i>LUCATESTA</i> . . . . .	112
3.3.1	Peripheral building blocks . . . . .	112
3.3.2	Layout considerations . . . . .	115
3.3.3	A building block library . . . . .	115
3.3.4	Post-Layout Simulations . . . . .	118
3.4	Conclusion . . . . .	125
<b>4</b>	<b>Measurements and Perspectives</b>	<b>127</b>
4.1	Test Setup and Experimental Results . . . . .	128
4.1.1	Test setup . . . . .	128
4.1.2	SASP validation measurements . . . . .	131
4.1.3	SASP applications measurements . . . . .	135
4.1.4	SASPEPA Characteristics . . . . .	140
4.2	An open window to RF applications - Achievement of SASP65K . . . . .	142
4.2.1	Schematic perspectives . . . . .	142
4.2.2	Technology issues . . . . .	145
4.2.3	Signal processing accuracy . . . . .	145
4.2.4	Real-Time error correction . . . . .	145
4.3	Conclusion . . . . .	146

<i>Contents</i>	7
<b>Conclusion</b>	<b>147</b>
<b>Publications</b>	<b>149</b>
<b>Bibliography</b>	<b>152</b>





# List of Figures

1.1	Transceiver Architecture . . . . .	25
1.2	Emitter Architecture . . . . .	26
1.3	Receiver Architecture . . . . .	26
1.4	Evolution of the GSM standard to LTE . . . . .	28
1.5	Global Subscribers by Technology . . . . .	29
1.6	Components parameters evolution in Mobile Phones . . . . .	29
1.7	Ideal Software Radio receiver architecture . . . . .	31
1.8	Realistic Software-Defined Radio receiver architecture . . . . .	33
1.9	Software-Defined to Software Radio Classification . . . . .	34
1.10	ADC issues . . . . .	35
1.11	ADC power consumption . . . . .	36
1.12	ADC limitations . . . . .	37
1.13	ADC Figure of merit . . . . .	38
1.14	Power consumption per MIPS for DSP the last 20 years . . . . .	39
1.15	Software Defined Radio by Baseband Conversion Architecture . . . . .	41
1.16	RF signal to baseband translation by Baseband Conversion Architecture . . . . .	41
1.17	RF signal to IF translation by IF Conversion Architecture . . . . .	41
1.18	Software Defined Radio by sub-sampling . . . . .	42
1.19	Software Defined to Software Radio State of the Art . . . . .	43
1.20	Proposed SR architecture . . . . .	44
1.21	Charge Coupled circuit in 3 working phases . . . . .	45
1.22	Block Diagram of a Transversal Filter . . . . .	46
1.23	Block Diagram of a Correlator . . . . .	47
1.24	Block Diagram of a Chirp Transform . . . . .	48
2.1	Proposed SR architecture . . . . .	53
2.2	Principle of the frequency translation . . . . .	54

2.3	Envelope selection and digitization . . . . .	55
2.4	Sinewave FFT . . . . .	55
2.5	Frequency Translation of a modulated signal . . . . .	56
2.6	$W_N^{nk}$ properties . . . . .	58
2.7	$N = 4$ Pipelined DFT . . . . .	60
2.8	Module of radix-2 DFT . . . . .	60
2.9	Basic cell of a pipelined DFT . . . . .	61
2.10	Step 1 . . . . .	62
2.11	Step 2 . . . . .	62
2.12	Step 3 . . . . .	63
2.13	Step 4 . . . . .	63
2.14	Step 5 . . . . .	64
2.15	Step 6 . . . . .	64
2.16	Step 7 . . . . .	65
2.17	Step 8 . . . . .	65
2.18	Step 9 . . . . .	66
2.19	Step 10 . . . . .	66
2.20	SASP Architecture . . . . .	67
2.21	Aliasing Matters without Anti Aliasing Filter . . . . .	68
2.22	Aliasing Matters with Anti Aliasing Filter . . . . .	68
2.23	Aperture Time Error . . . . .	69
2.24	Aperture Uncertainty Error . . . . .	70
2.25	Differents kind of windows . . . . .	72
2.26	Window characteristics . . . . .	72
2.27	Diagram flow of a radix-4 FFT with $N = 16$ , i.e. 2 stages . . . . .	73
2.28	Basic radix-4 FFT module . . . . .	74
2.29	Stage architecture . . . . .	74
2.30	Accumulation Delay Line . . . . .	75
2.31	Samples selection . . . . .	76
2.32	Envelope voltage samples selection . . . . .	77
2.33	Concurrent reception . . . . .	79
2.34	Theoretical BPSK signal processing . . . . .	80
2.35	A synchronized BPSK signal processed by 4096-point SASP . . . . .	80
2.36	A non-synchronized BPSK signal processed by 4096-point SASP . . . . .	81
2.37	A OFDM signal processed by the SASP . . . . .	81

3.1	Processor Architecture . . . . .	84
3.2	(a) Architecture of SASP64, (b) Close-up of a stage . . . . .	85
3.3	Diagram flow of a radix-4 FFT with $N = 64$ . . . . .	86
3.4	Design Flow . . . . .	87
3.5	Processing phases . . . . .	88
3.6	Delay Line system view . . . . .	89
3.7	Simplified schematic of a delay cell . . . . .	90
3.8	(a) Buffer characterization, (b) Output derivation . . . . .	91
3.9	Delay line lowest working frequency . . . . .	92
3.10	Charge Transfer. Load (a) and Display (b) of a voltage sample . . . . .	92
3.11	Simulation of a delay line . . . . .	93
3.12	Delay Cell layout . . . . .	93
3.13	Delay Line with 64 samples layout . . . . .	94
3.14	Matrix design based on adders implementation . . . . .	95
3.15	Simplified schematic of a 4-voltage sample adder . . . . .	96
3.16	Simulation of a 4-voltage sample adder . . . . .	97
3.17	Simulation of a 4-voltage sample adder, a signal and its inverse added . . . . .	97
3.18	Matrix layout . . . . .	98
3.19	Comparaison between non-optimized and optimized WU position . . . . .	99
3.20	Weighting Unit architecture . . . . .	99
3.21	Simplified schematic of the Weighting Unit . . . . .	101
3.22	A 100mV voltage sample weighted by 4 coefficients . . . . .	102
3.23	Weighting Unit layout . . . . .	103
3.24	Design startegy of the digital circuitry . . . . .	105
3.25	Generation of 4 pulses . . . . .	105
3.26	Simulation of pulses generations . . . . .	106
3.27	Simulation of 64 pulses . . . . .	107
3.28	Simulation of the maximal $f_{\text{sampling}}$ . . . . .	107
3.29	Example of a logic circuit to address WU . . . . .	108
3.30	Simulation result of a state combination, $M_3$ . . . . .	109
3.31	Architecture of the Sample Selector . . . . .	109
3.32	Generation of the selection of the 7 <sup>th</sup> voltage sample . . . . .	111
3.33	Digital part layout . . . . .	111
3.34	Sampler Architecture . . . . .	113
3.35	Sampler simulation . . . . .	113

3.36	Windowing circuit (simplified schematic) . . . . .	114
3.37	Simulation result of a Hamming Window . . . . .	114
3.38	Design strategy . . . . .	115
3.39	Stage 1, Stage 2 and Stage 3 layouts . . . . .	116
3.40	SASPEPA Layout . . . . .	117
3.41	PLS of clock signal generation . . . . .	118
3.42	PLS of windowing operation . . . . .	119
3.43	PLS of output spectrum . . . . .	120
3.44	Output spectrum in order . . . . .	121
3.45	PLS of the sample selection . . . . .	121
3.46	PLS of output spectrum . . . . .	122
3.47	Output spectrum in order . . . . .	122
3.48	PLS of output spectrum . . . . .	123
3.49	Output spectrum in order . . . . .	123
3.50	A non-entire frequency sinewave processed by 64-point SASP . . . . .	124
4.1	Floorplan of SASPEPA . . . . .	129
4.2	Floorplan of LUCATESTA . . . . .	129
4.3	Test board of SASEPA . . . . .	130
4.4	Coplanar waveguide . . . . .	130
4.5	Photo of instruments configurations . . . . .	131
4.6	OUTCLK signal generation . . . . .	132
4.7	OUTCLK measurements . . . . .	132
4.8	Measured hamming window . . . . .	133
4.9	Measured error on hamming window . . . . .	134
4.10	Retro-simulation of a windowed zero-signal . . . . .	134
4.11	SASP output saturation . . . . .	135
4.12	Principle of frequency Shifting . . . . .	136
4.13	Measure of frequency Shifting . . . . .	136
4.14	Measure of a non-entire frequency sinewave processed by 64-point SASP . . . . .	137
4.15	Frequency shift of FM signal . . . . .	137
4.16	BPSK modulation . . . . .	138
4.17	FSK modulation . . . . .	139
4.18	ASK modulation . . . . .	139
4.19	Output amplitude vs $f_{\text{sampling}}$ . . . . .	140
4.20	Output amplitude vs $n_{\text{sample}}$ . . . . .	141

---

4.21	First proposed delay line architecture . . . . .	143
4.22	Second proposed delay line architecture . . . . .	143
4.23	SASP system auto-calibration . . . . .	146



# List of Tables

1.1	Wireless Communication Systems Characteristics . . . . .	27
1.2	Figure of Merit . . . . .	38
1.3	Charge Coupled Devices Sum up . . . . .	48
2.1	Windows and Figure of Merit . . . . .	72
2.2	Operating states and switches configurations . . . . .	75
2.3	Comparison of number of samples at a given sampling frequency of 4GHz . . . . .	78
2.4	RF standards addressed by 65536-point SASP . . . . .	78
3.1	Voltage loss vs $f_{\text{sampling}}$ . . . . .	92
3.2	Simulated delay lines power consumption . . . . .	94
3.3	WU Coefficients . . . . .	100
3.4	Weighting Unit Coefficients Application . . . . .	101
3.5	Weighting Unit Coefficients Simulation . . . . .	103
3.6	Simulated WU power consumption . . . . .	103
3.7	Binary code . . . . .	110
3.8	Metals . . . . .	117
3.9	Power Consumption . . . . .	125
4.1	Different power supplies . . . . .	128
4.2	Digital part validation . . . . .	132
4.3	Power Consumption under 1.4V . . . . .	141
4.4	SASPEPA Characteristics . . . . .	142
4.5	Weighting Unit Coefficients extension . . . . .	144
4.6	Estimated Power Consumption . . . . .	145





# List of Abbreviations

<b>A/D</b>	Analog-to-Digital
<b>AAF</b>	Anti-Aliasing Filter
<b>AC</b>	Alternating Current
<b>ADC</b>	Analog-to-Digital Converter
<b>AM</b>	Amplitude Modulation
<b>ASIC</b>	Application-Specific Integrated Circuit
<b>BIST</b>	Built-In Self-Test
<b>BPSK</b>	Binary Phase Shift Keying
<b>CCD</b>	Charge-Coupled Device
<b>CDMA</b>	Code Division Multiple Access
<b>CMOS</b>	Complementary MOS
<b>DAC</b>	Digital-to-analog converter
<b>DARPA</b>	Defense Advanced Research Projects Agency
<b>DC</b>	Direct Current
<b>DCS</b>	Defense Communications System
<b>DFT</b>	Discrete Fourier Transform
<b>DSP</b>	Digital Signal Processor
<b>DUT</b>	Device Under Test
<b>EDGE</b>	Enhanced Data for GSM Evolution
<b>ENOB</b>	Effective Number of Bits
<b>FFT</b>	Fast Fourier Transform
<b>FM</b>	Frequency Modulation
<b>FSK</b>	Frequency Shift Keying
<b>GMSK</b>	Gaussian Minimum Shift Keying
<b>GPRS</b>	General Packet Radio Service
<b>GPS</b>	Global Positioning System
<b>GSM</b>	Global System for Mobile Communications
<b>HSDPA</b>	High-Speed Downlink Packet Access

---

<b>IC</b>	Integrated Circuit
<b>IF</b>	Intermediate Frequency
<b>IFFT</b>	Inverse FFT
<b>ISSCC</b>	International Solid-State Circuits Conference
<b>JTO</b>	Joint Program Office
<b>JTRS</b>	Joint Tactical Radio System
<b>LNA</b>	Low Noise Amplifier
<b>LTE</b>	Long Term Evolution
<b>MEMS</b>	Micro Electro Mechanical Systems
<b>MIPS</b>	Million Instructions Per Second
<b>MNOS</b>	Metal Nitride-Oxide Semiconductor
<b>MOS</b>	Metal Oxide Semiconductor
<b>MU</b>	Matrix Unit
<b>OFDM</b>	Orthogonal Frequency Division Multiplexing
<b>PA</b>	Power Amplifier
<b>PCB</b>	Printed Circuit Board
<b>PCS</b>	Personal Communications System
<b>PLS</b>	Post Layout Simulation
<b>PSK</b>	Phase Shift Keying
<b>QPSK</b>	Quadrature PSK
<b>QAM</b>	Quadrature AM
<b>RF</b>	Radio-Frequency
<b>SASP</b>	Sampled Analog Signal Processor
<b>SDR</b>	Software-Defined Radio
<b>SR</b>	Software Radio
<b>SNR</b>	Signal-to-Noise ratio
<b>T/H</b>	Track-and-Hold
<b>UMTS</b>	Universal Mobile Telecommunications System
<b>UWB</b>	Ultra-Wideband
<b>VHDL</b>	Very high-speed integrated circuits Hardware Description Language
<b>VHDL-AMS</b>	VHDL-Analog and Mixed-Signal
<b>WCDMA</b>	Wideband CDMA
<b>WU</b>	Weighting Unit

# List of Notations

$f_{\text{sampling}}$	SASP Sampling frequency
$k$	$n^{\text{th}}$ root of unity index
$N$	Number of voltage samples handled by the SASP
$n_{\text{envelope}}$	Number of samples in a RF envelope
$n_{\text{sample}}$	Frequency sample number in a DFT processing sequence
$T_p = N.T_{\text{sampling}}$	Period of a processing sequence
$T_{\text{sampling}}$	Sampling Period of the SASP
$r_{\text{stage}}$	Stage number in the pipelined DFT $[1, \log_4(N)]$
$W_N^k$	Twiddle factor



# Introduction

The recent increase in the demand for wireless devices has led to the emergence of various standards. Cellular systems are more and more required to accept different kinds of applications such as audio, graphic or video data. Mobile terminals are the place for a real multimedia convergence. For this reason multifunctional wireless devices are required. They are able to accommodate different wireless standards with different carrier frequencies, channel bandwidths, modulation schemes or data rates.

Multifunctional circuits and systems are part of the solution. They can integrate the concept of Software Radio (SR) in just one chip. A SR circuit can be tuned to any frequency band, select any reasonable channel bandwidth, and detect any known modulation. This thesis presents the design of a Radio Frequency Front-End receiver dedicated to SR for mobile terminals. This receiver is based on a Sampled Analog Signal Processor (SASP). The latter is placed between the antenna and the ADC in a receiver chain.

Chapter 1 presents the Software Radio concept. It emphasizes the technological bottleneck of a full Software Radio receiver chain. The receiver is composed of an antenna, an ADC and a DSP. But, to accept any RF standard, ADC and DSP have to deal with high signal resolution at RF frequencies. It leads to a very high power consumption unsuited to mobile terminals. The idea is to perform part of digital signal processing in analog. A state of the art of analog signal processing components is proposed.

Chapter 2 exposes the principle of the SASP. It does basic analog operations on discrete time voltage samples. The purpose is to reduce the RF signal data rate before digital conversion. Analog operations give the opportunity to work directly at RF frequencies at an acceptable power consumption and to display a low frequency output signal. The SASP aims at processing the RF input signal spectrum. Only the spectral envelope of the desired RF signal is analogically selected and sent toward an ADC. To carry out the operation, the SASP implemented an analog Discrete Fourier Transform (DFT). Two parameters inherited from the DFT equation mastered

the SR requirements such as reprogrammability and flexibility: the sampling frequency  $f_{sampling}$  and the number of voltage samples  $N$ . Two applications are proposed to enhance SASP principle: concurrent reception enables to receive and shift to baseband several RF signals, frequency demodulation performs part of the digital signal processing in analog and participates to reduce the ADC input frequency.

Chapter 3 exhibits the SASP design. The goal is to validate the feasibility of the SASP with a demonstrator using 65nm CMOS technology of STMicroelectronics. This demonstrator handles 64 voltage samples. Each part of the system is detailed. The implementation is discussed and characteristics are given through behavioral simulations.

Chapter 4 concludes on measurements of two chips. They both confirm the physical feasibility of the SASP and identify technical points to be improved for an industrial product. A technological roadmap is paved for a Software Radio chip. Perspectives are given concerning the architecture, the design and the targeted characteristics.

---

# The Software Radio Concept

## Contents

---

<b>1.1</b>	<b>Wireless communication systems</b> . . . . .	<b>24</b>
1.1.1	Wireless communication architectures . . . . .	25
1.1.2	Wireless communication standards . . . . .	27
1.1.3	Wireless communication market and trends . . . . .	28
1.1.4	Conclusion . . . . .	30
<b>1.2</b>	<b>Software Radio Background</b> . . . . .	<b>30</b>
1.2.1	Definition . . . . .	30
1.2.2	History . . . . .	31
1.2.3	Software Radio Characteristics . . . . .	33
1.2.4	Technological Bottlenecks . . . . .	35
1.2.5	Software Defined Radio Architectures . . . . .	40
<b>1.3</b>	<b>Analog Signal Processing</b> . . . . .	<b>44</b>
<b>1.4</b>	<b>Conclusion</b> . . . . .	<b>49</b>

---



*Chapter 1 presents the Software Radio concept. The first part reminds RF transceivers architectures, RF standards and wireless devices market. The second part shows how telecommunication industry is faced with new challenges. It tends to integrate more and more functionalities in mobile terminals whereas technological bottlenecks prevent from designing low cost solutions. The Software Radio concept proposes new ways to success in a full multimedia convergence at the lowest price fulfilling constraints imposed by mobile terminals.*

**Key words:** RF architectures, RF standards, multimedia convergence, software radio, analog signal processing

Wireless communication systems are faced with the emergence of various standards dedicated to voice transmission, data transfer and localization. The past decade has seen a fast evolution regarding the communication standards: their data rates have increased, their carrier frequencies are higher and their modulations are more complex. While integrating all these changes, mobile terminals tend to address several communication standards in just one handset. But, conventional architectures cannot challenge this multimedia convergence satisfying technological matters imposed by handsets at a lowest price. Thus, new architectures need to be studied in order to answer to mobile terminals constraints. This chapter presents the classical architectures of transceivers in the case of mobile terminals and an overview of the new solutions proposed by the Radio Frequency community to overcome technological issues.

## 1.1 Wireless communication systems

Transceivers architectures dedicated to mobile terminals have emerged in the 90's to answer wireless communications such as GSM. A continuous evolution of wireless systems has been led by an increasing demand of the market to design products which integrate more and more applications at the lowest price.

### 1.1.1 Wireless communication architectures

The aim of a wireless communication system is to receive and transmit information at high frequencies. A basic architecture can be summarized up as follow (Fig. 1.1) with two paths:

- A transmitting path: to send information (voice, data, localization), a digital signal encoding the information by a data stream is processed through a modulation scheme. It is then carried at RF frequencies (carrier frequency) and amplified in order to be transmitted.
- A receiving path: to receive information, RF signals are amplified, filtered, translated into baseband by a Front End. The signal is processed digitally to recover the information contained in the data stream.

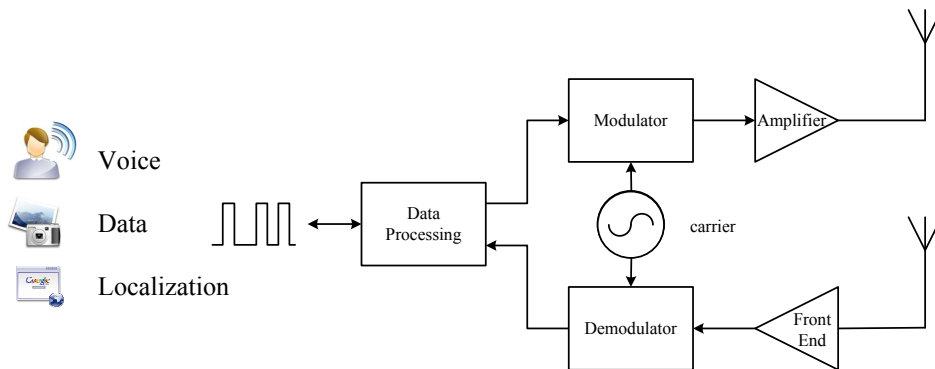


Figure 1.1: Transceiver Architecture

#### 1.1.1.1 Emitter chain

The emitter chain is composed by (Fig. 1.2):

- A Digital to Analog Converter (DAC).
- A Mixer to up convert the signal.
- A Power Amplifier (PA).

Digital information is encoded and modulated to be up converted at RF frequencies. Then, a PA amplifies the signal meeting the requirements given by international standards. This step is one of the most important, as amplification is a trade-off between linearity (signal quality) and power (communication range and power consumption). Designers are faced to issues like improving PA performances without increasing power consumption (limited in the case of a handset) and decreasing the PA yield.

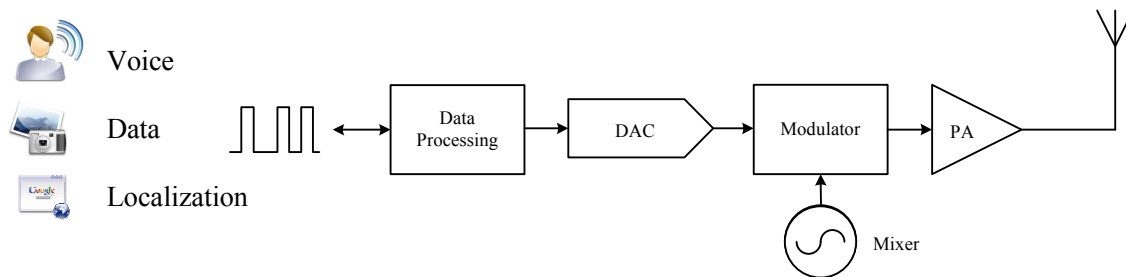


Figure 1.2: Emitter Architecture

### 1.1.1.2 Receiver chain

The receiver chain aims at receiving signals from the base station. The chain is composed by (Fig. 1.3):

- A Low Noise Amplifier (LNA).
- A Mixer to down convert the signal.
- An Analog to Digital Converter (ADC).

The LNA amplifies the received signal to be processed fitting the requirements. In fact, the signal received by the antenna is very weak and cannot be treated ad-hoc properly. LNA features (noise, gain) are crucial to guarantee a good functionality. The mixer down converts the RF frequency signal to intermediate frequencies or baseband. It is then converted into digital by an ADC. A DSP finally processes the signal to decode the information and transmit recovered data to the user through an analog interface (earphone, screen).

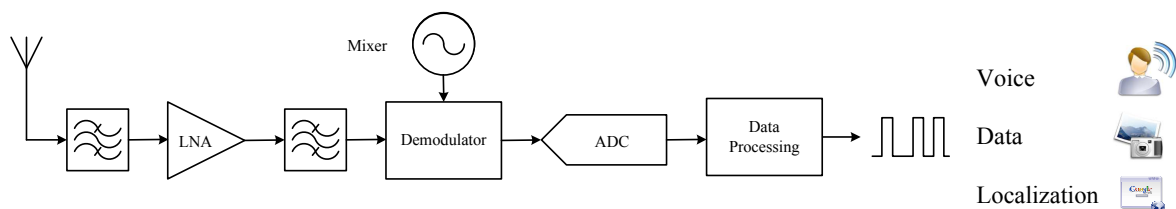


Figure 1.3: Receiver Architecture

The principle exposed here is a general one. Several kinds of architectures are used to perform the reception. It is determined by a trade-off between parameters to maximize the reception. For instance, it can be cited: heterodyne, super-heterodyne, image rejection, homodyne (direct conversion or zero-Intermediate Frequency), low Intermediate Frequency, polyphase [1].

The super-heterodyne structure is the most used. But, the trend to integrate circuits on a single chip at a low power represents a real drawback of this architecture. At the same time, multi-standard systems are claimed. It leads to search for more adapted topologies such as homodyne, low-IF or polyphase [2].

### 1.1.2 Wireless communication standards

The wireless communication standards world is a world of diversity [3]. Historically, the GSM (2G) is the first standard to have dominated the cell phone market in 1992. In June 2008, it numbered 3 billion of users in the world. The native version operated at 900MHz and was extended to two other frequency bands at 1800MHz and 1900MHz. It is mainly used to transmit voice and data (SMS). The GPRS (2.5G) and EDGE (2.75G) standards are evolutions of the GSM. The same frequencies are used but the data rate is higher thanks to new modulation schemes (8-PSK). Multimedia services are thus proposed to users.

The successor of GSM is UMTS (3G). It has a high data rate and enables video transmission. While this new standard is appearing and bringing multimedia closer and closer to the final user, other standards provide better wireless communications between multimedia devices (Bluetooth and WiFi standards). Table 1.1 offers an overview of the standards nowadays used. It exhibits clearly the fast growth of wireless device market and the diversity of the technological constraints.

Table 1.1: Wireless Communication Systems Characteristics

Standards	GSM900	UMTS	Bluetooth	WiFi (802.11b)
RX Band (MHz)	925-960	2110-2170	2448-2482	2412-2472
TX Band (MHz)	880-915	1920-1980	2448-2482	2412-2472
Channel Bandwidth	200kHz	5MHz	1MHz	20MHz
Modulation	GMSK	QPSK	GFSK	OFDM / QPSK
Resolution (bits)	14	9	12	7
Data Rate	14.4-115kbps	384kbps to 2Mbps	723kbps	11Mbps to 54Mbps
Application	Voice	Data/Voice	Data	Data

### 1.1.3 Wireless communication market and trends

#### A growing diversity

With all the sophistication that characterizes today's mobile phones, it is easy to forget that the handset, at core, is a radio set! Traditionally, radios have been implemented entirely in hardware thanks to analog circuits, with new waveforms added by integrating new hardware. However, last generation of handsets needs to support all of the following wireless standards: GSM, GPRS, EDGE, WCDMA, HSDPA, LTE, GPS, mobile TV, WiFi, Bluetooth and UWB (Fig. 1.4). Besides, every new standard added on the market is faced with faster and faster introduction (Fig. 1.5) whereas multi-mode handsets must be able to operate across GSM and CDMA networks. The number of waveforms to be supported is consequently considerable, various and becomes hard to handle. The trend can be summed up by a multimedia convergence on a lonely handset.

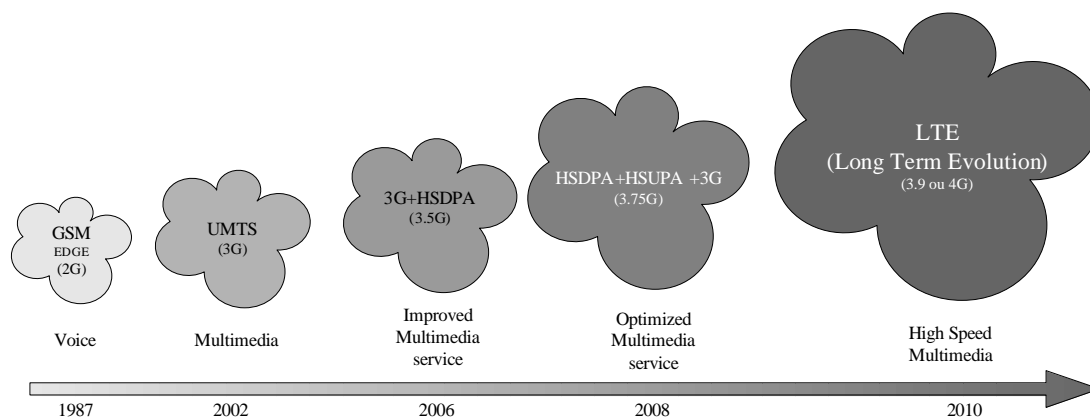


Figure 1.4: Evolution of the GSM standard to LTE

#### Telecommunication industry challenges

Telecommunication industry dilemma is to maintain a constant (or lowest) production cost and handset size while integrating more and more functions with a maximum reactivity. Figure 1.6 depicts the reduction of Integrated Circuit (IC) footprint in cell phones during the first years of a product. GSM is here compared to a dual mode chip (GSM and UMTS). The industry is answering to new technological matters faster when new solutions appear. For instance UMTS came to maturity more than two times faster than the GSM. As said before, it can be seen than size and complexity of discrete components are tried to be maintained as stable as possible. Thus, the trend is to integrate more features for free thanks to miniaturization based upon multiple chip package, high density package, or novel RF architectures.

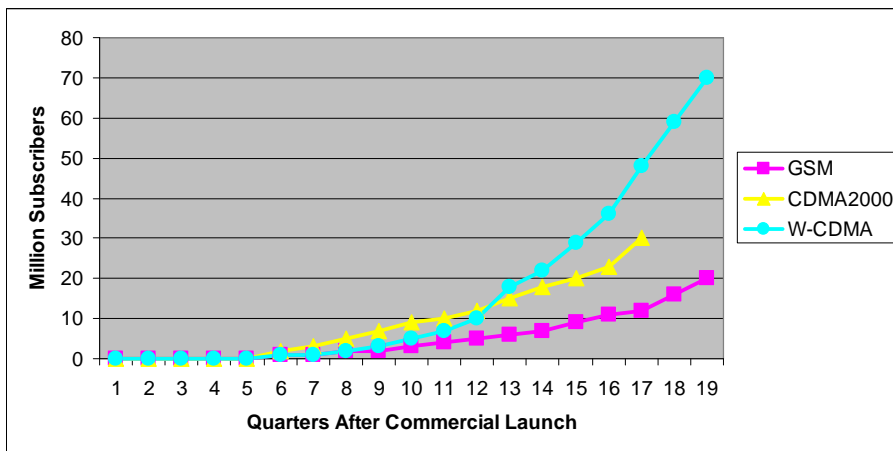


Figure 1.5: Global Subscribers by Technology

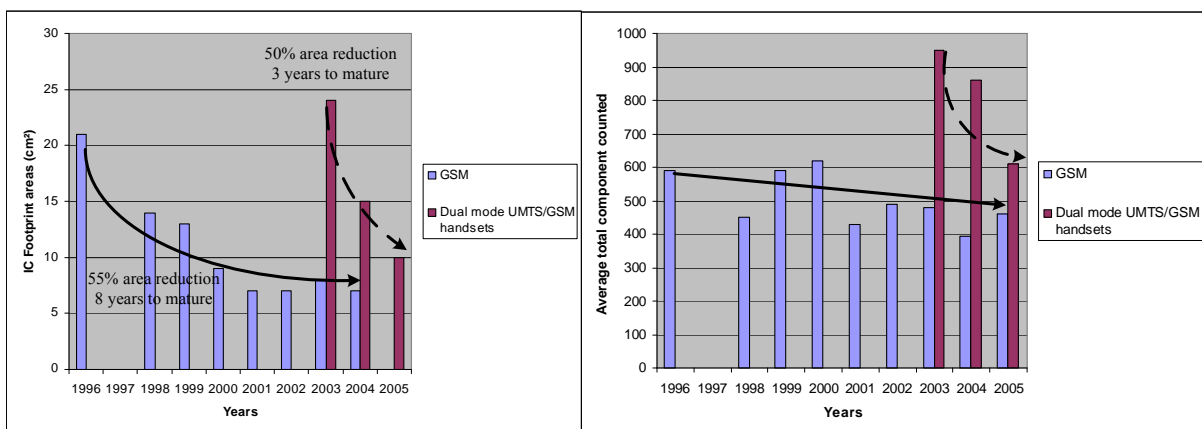


Figure 1.6: Components parameters evolution in Mobile Phones

For this reason, integrating additional radio hardware is impractical beyond a point because it increases the handset size, complexity and price. As mentioned before, novel RF architectures are to be created. Software-Defined Radio (SDR) and Software Radio (SR) are concepts aiming at integrating a RF architecture able to handle any kind of RF standards in only one chip. The attraction of SDR/SR is its ability to support multiple waveforms by re-using the same hardware while changing its parameters by software. This has enormous benefits for handset size, cost, development cycle, upgrade and interoperability. Whereas the demand of SDR/SR product is very strong, technical challenges are to be solved by telecommunication industry. The next section exposes the most important obstacles to be solved to design a handset based on SDR or SR concepts.

### 1.1.4 Conclusion

This overview of wireless domain is concluded by two main observations that exhibit the need of rupturs in RF architectures design.

**User habits :** Handset needs to have evolved from the mere phone call to multimedia services.

A multimedia convergence is required by the omnipresence of communication device in our every day life. Telecommunication industry and consumers claim for a one-product solution.

**Network diversity :** As the wireless market has exploded, various telecommunication standards can be found. This diversity implies a technological challenge to merge any standard in a one-product solution.

The classical way of building Radio architectures is over. To answer to the new constraints exposed previously, new structures are to be found.

## 1.2 Software Radio Background

The wireless industry is faced with the multimedia convergence in mobile terminals, looking for new RF architectures. Joseph Mitola exposed for the very first time the concept of Software Radio (SR) in 1995 [4]. The concept of SR aims at designing a reconfigurable radio architecture accepting all cellular and non-cellular standards working in a 0 to 5 GHz frequency range. Technical challenges need to be solved to address this concept. This section presents the concept of SR, its technological bottlenecks and an overview of existing SDR RF architectures dedicated to mobile terminals.

### 1.2.1 Definition

#### Principle

A definition of Software Radio is a "Radio in which the entire physical layer functions are software defined". A particular case of that principle is the concept of Software Defined Radio (SDR) defined as a "Radio in which some or all of the physical layer functions are software defined", commonly adopted by the SDR industry association [5].

A Software-Radio receiver architecture is depicted in figure 1.7. The concept is to bring as close as possible the Analog to Digital Conversion to the antenna. Thus, the ideal system is composed by an antenna, an ADC and a DSP. The DSP is reconfigurable by software and can address

any standard. This architecture can adapt itself to any kind of radio context and treat any RF signal. But, nowadays technological bottlenecks prevent from realizing such a utopian system. The Analog to Digital Conversion, if done directly after the antenna requires a high resolution at high sampling frequencies. It would consume a lot of power which is not compatible with mobile terminal battery life. Intermediate solutions are studied to achieve the Software Radio concept.

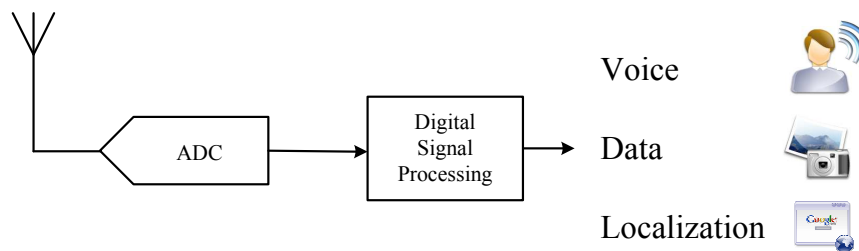


Figure 1.7: Ideal Software Radio receiver architecture

### From Software-Defined to Software Radio

Joseph Mitola defined SDR as follow: *A Software-Defined Radio (SDR) is a radio that can accommodate a significant range of RF bands and air interface modes through software. For the ideal software radio, that range includes all the bands and modes required by the user/host platform* [6]. As explained before, researches tend to bring closer and closer the ADC to the antenna in RF transceivers (Fig. 1.1). SDR is consequently a step toward SR. SDR is characterized by:

- A narrow band analog signal processing (under 20MHz).
- Baseband digital signal processing.

SR is characterized by:

- A wide-band analog signal processing (over 2GHz).
- Full RF spectrum covered.

### 1.2.2 History

The US Army is the first to study SDR projects. The purpose was to secure radio communications between operational units on a hostile battlefield. Radio communications had to be reconfigured rapidly in order to prevent from being spied enemies. Defense Advanced Research Projects Agency (DARPA) financed researches. Project "Speakeasy" gave the first result at the



beginning of the 90's. The evolution was first to handle several standards on a 2MHz - 2GHz frequency band and reconfigure "on the fly" the communication device with one known standard. It went on the implementation of new standards through known standards in order to switch rapidly on a new way of communication. This necessity found obviously all its meaning in the military market.

### **A military market**

The military market is more and more interested in that technology thanks to the differences existing between allied armies or any coalitions. At the end of the 90's, US army established the JPO (Joint Program Office) to launch JTRS (Joint Tactical Radio System) [7]. Today, JTRS finances projects on Software Radio to be operated in the US army at a term of 10 years. Companies start working on Software Radio systems at the demand of many countries. Several armies in the world are to be equipped in Software Radios, following the US army strategy.

### **A commercial market**

Software Radio is not only a military subject. The wireless industry gave rapidly an interest to SR. An international organization, called SDR Forum promotes researches, development and use of SR technologies concerning communication systems. It brings together companies, universities and governmental organizations to give direction to researches and exchange ideas. The final aspiration is to provide worldwide, flexible and low cost handsets.

### **A civil market**

An other market targeted by SR systems is the civil and emergency one. It took all its importance since the 9/11 attacks. Police and Fire departments, emergency units are able to inter-communicate thanks to a unique way of communication as coordination is an essential key in case of emergency. The fight against terrorism increased the demand of technological solutions to share information securely and as fast as possible.

### **First researches**

Researches on SDR were brought by reconfigurable architectures (Fig. 1.8). Each part of the architecture is reconfigurable and thus can be changed while required. Filters scale on the best bandwidth, mixers down convert at the chosen frequency in order to optimize the A/D conversion. The DSP manages the configuration of each element.

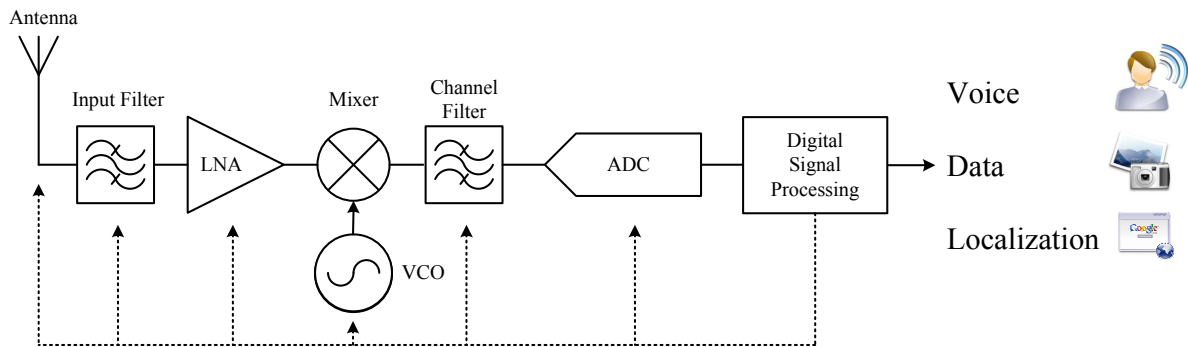


Figure 1.8: Realistic Software-Defined Radio receiver architecture

Despite the concept of Software-Defined Radio is approached and may appear easy to be explained and designed, the system developed consumes a lot of power and cannot be adapted to a handset and at any kind of standards. Technological bottlenecks are clearly identified and pave the way to develop a Software Radio mobile terminal. An effort has to be done to work on new architectures. That is why each market is financing and pulling up researches for their interests.

### 1.2.3 Software Radio Characteristics

Our researches are focused on commercial handsets. Software Radio brings flexibility and adaptability. Many gains are expected by telecommunication industry:

**Gain of compatibility :** a common system can address any kind of standards and thus can be used wherever in the world. A mass production leads to a cost reduction.

**Gain of production time :** research and development time is optimized between the apparition of a new standard and its use. As a basic architecture is to be designed, only updates (design, software) are required to accept new standards.

**Gain of performance :** a SR system is able to reconfigure itself depending on the context (geographical, data rate, etc). It can adapt the data rate and the bandwidth using the most efficient standard.

Performances of SR are not only technological. Industrial efficiency and easiness are thus proven.

A graphical classification is proposed to point out the degree of flexibility of an architecture (Fig. 1.9). It is determined by the access frequency and the addressed bandwidth performed

into analog domain as digital part is the technological bottleneck. It can determine if a system is either a Software-defined or a Software Radio. The ideal Software Radio receives and handles a wide-band of RF signals. The architecture to be designed in this thesis is a Software Radio architecture.

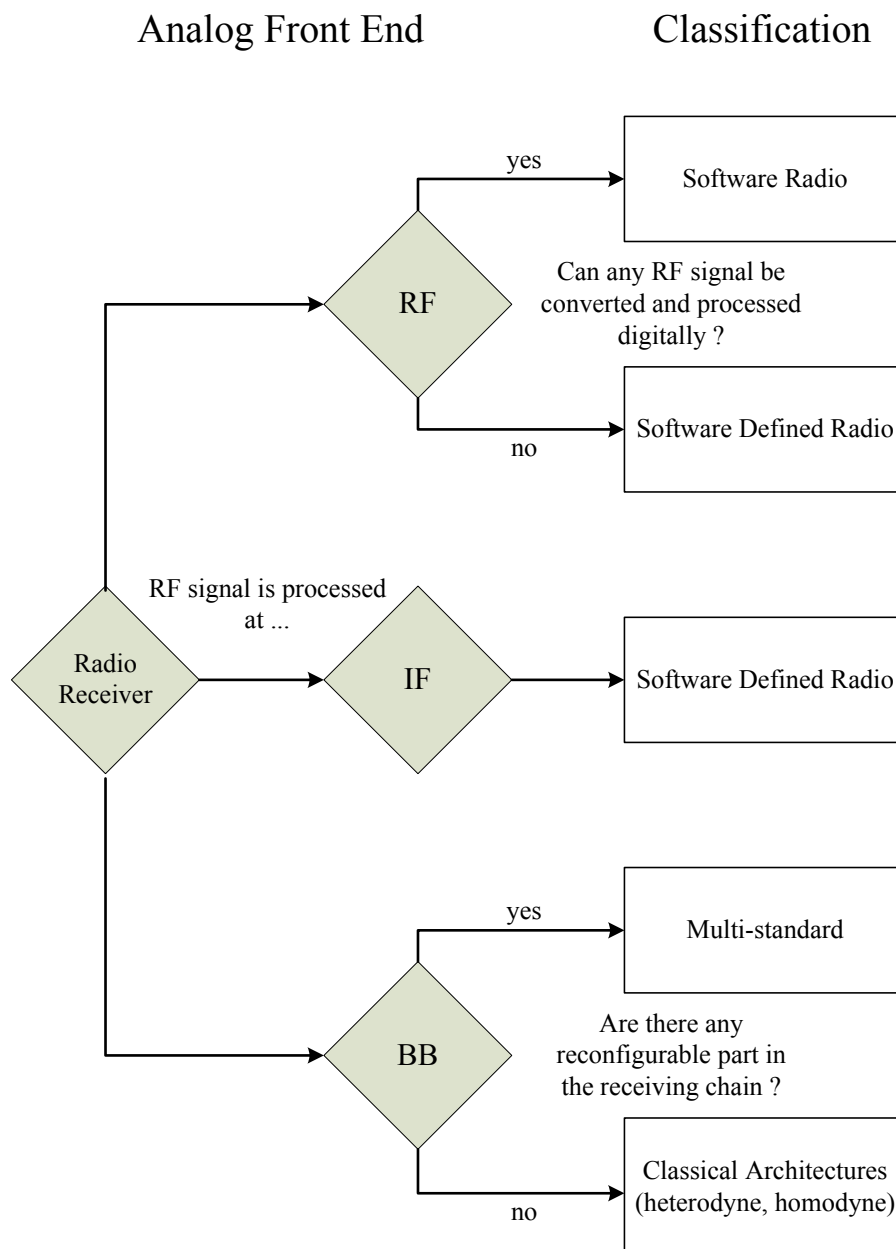


Figure 1.9: Software-Defined to Software Radio Classification

### 1.2.4 Technological Bottlenecks

To achieve the development of a SR RF device, 3 technological bottlenecks are described, from the most difficult to the less difficult to overcome.

#### 1.2.4.1 A/D conversion

SR receiving chain has to digitize as close as possible to the antenna any RF signal. A 0 to 5GHz RF band is the widest band to target in order to cover the entire RF spectrum. So, the SR ADC requirements are given by:

- At least a 10GHz sampling frequency ( $f_{\text{sampling}}$ ).
- A 16 Effective Number Of Bit ( $N_b$ ) is required to accept any dynamic range among all defined RF standards.
- The power consumption ( $P(W)$ ) is directly linked to the sampling frequency. The higher the frequency, the higher the power consumption. In a context of mobility, the battery life is the major parameter to take into account.
- Silicon area ( $A(mm^2)$ ) is important as it determines the component cost.

Given the figure 1.10, such an ADC at low power consumption(Fig. 1.11) [8], at high frequencies with an acceptable accuracy is nowadays not feasible.

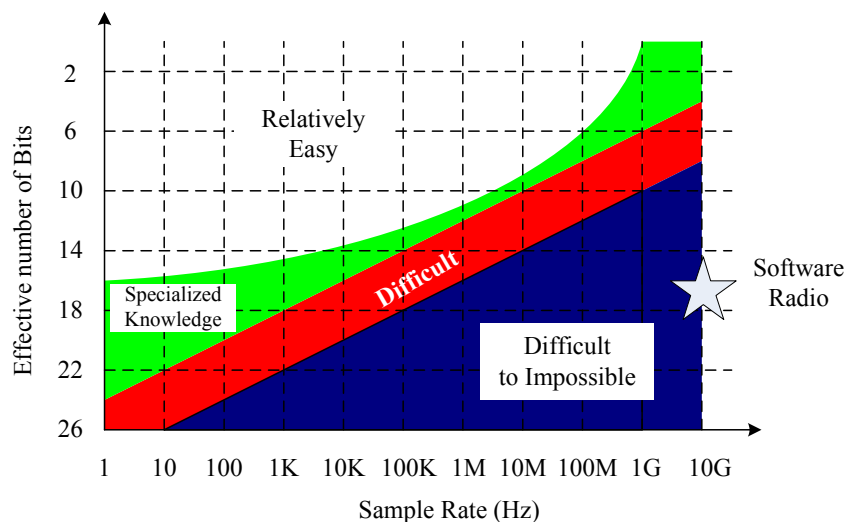


Figure 1.10: ADC issues

Extrapolating current A/D converter characteristics, the A/D converter for SR would consume about 1 kW (Fig. 1.11). This is far too much for handsets. The progress in A/D converters at the same power level (at the same sample frequency) is about 1.5 bit in 8 years [9]. As the power consumption issue depends on frequency and resolution, the next paragraph gives clues to understand the technological bottleneck [10].

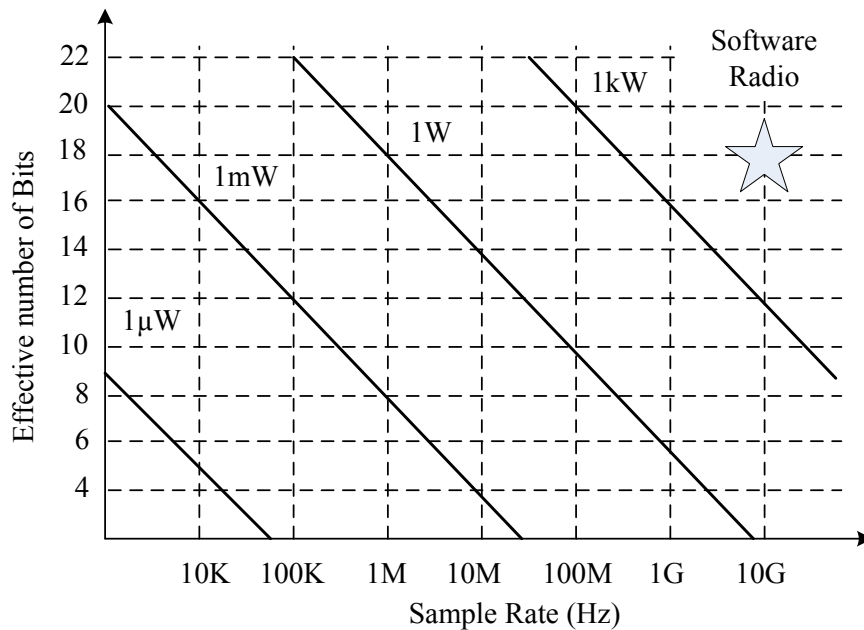


Figure 1.11: ADC power consumption

### Limiting factors

Two main limiting factors are thus exhibited: frequency and resolution. Power consumption is considered as depending on these two factors. A/D converters performances are mainly limited by three physical phenomena: thermal noise, jitter and quantization (minimal resolution) (Fig. 1.12). Currently, ADCs found on the market have specifications approximately from 500kHz with a 24-bit resolution to 2GHz with a 8-bit resolution. These specifications are under the requirements of an ideal Software Radio system (Fig. 1.10). It can be estimated that at least 15 years of works are required to achieve a low power ADC answering to SR constraints, if one day feasible !

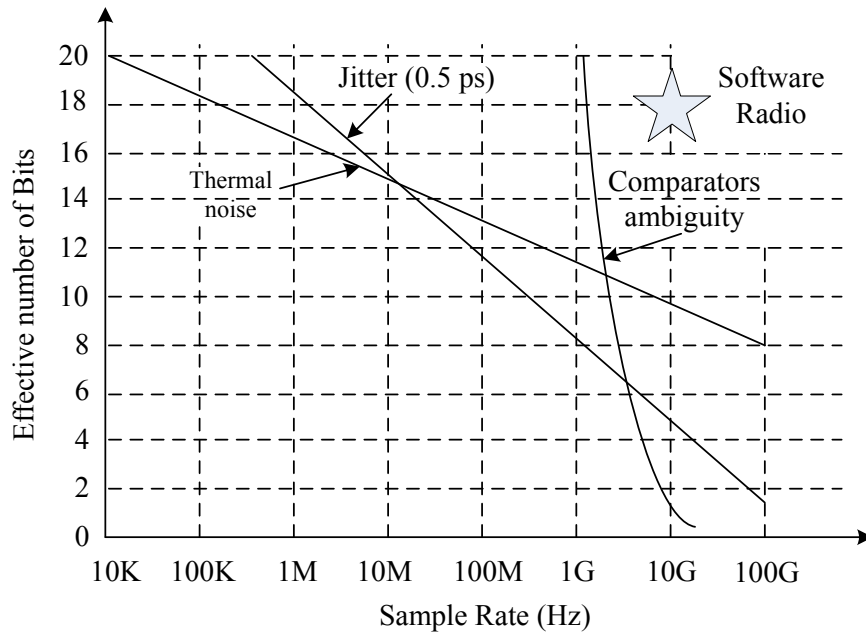


Figure 1.12: ADC limitations

Two parameters are commonly defined as a figure of merit to evaluate converters:

- $F_m$  which take into account all the variables of an ADC.
- $F_t$  which is limited to technological matters.

They are given by:

$$F_m = \frac{2^{N_b} \cdot f_{\text{sampling}} (\text{MHz})}{P (\text{mW}) \cdot A (\text{mm}^2)} \quad (1.1)$$

$$F_t = \frac{P (W)}{2^{N_b} \cdot f_{\text{sampling}}} \quad (1.2)$$

These two parameters reflect the "cost" of an ADC in the case of SR. For instance,  $F_m$  exhibits the silicon area ( $A (\text{mm}^2)$ ) impact on the circuit, considering technological cost (the bigger, the more noisy the circuit is) and considering the price (the smallest, the cheapest, or at least the same price). If  $A$  decreases,  $F_m$  increases.

Figure 1.13 and Table 1.2 show the evolution of ADCs found in literature of ISSCC. It depicts the evolution of converters figure of merit in the past decade and thus draws directions for the next decade. It exhibits the hard trade-offs between resolution,  $f_{\text{sampling}}$  and power consumption (e.g. [11] [12]). Two observations are made:

- The progression of  $F_m$  is constant, it is done whereas  $N_b$  remained low at 10.
- $F_m$  is slowing down while  $N_b$  is increased.

This observation enables to draw a perspective of 15 years at least required to target the goal of  $F_m = 660000$  with  $N_b = 16$ .

Table 1.2: Figure of Merit

Parameters	Minimum	Typical	Maximal	ISSCC'04 [13]	ISSCC'06 [11]	ISSCC'08 [12]
$N_b$	12	14	16	10	9.5	11
$f_{\text{sampling}}(\text{MHz})$	20	40	10 000	160	32	200
P(mW)	20	10	100	122	22	180
$A(\text{mm}^2)$	5	2	10	1.7	0.3	1.1
$F_m$	820	32 768	655 360	787	3500	2068
$F_t(fJ)$	244	15	0.15	747	949	630

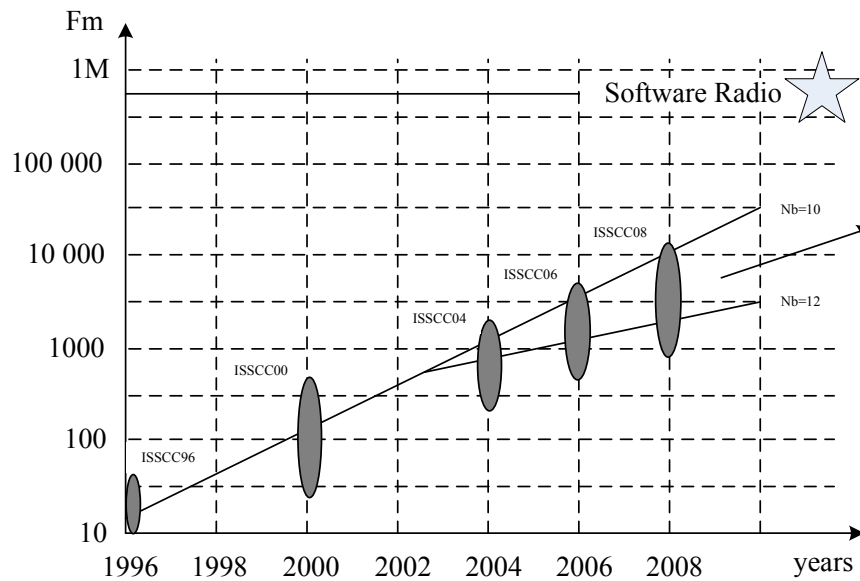


Figure 1.13: ADC Figure of merit

## Conclusion

Technical informations given in this part show how critical is the A/D conversion. ADCs suiting strong SR requirements are not expected to be achieved before 15 years or more. Designers must find new architectures which relax ADC requirements to achieve a SR system [10].

### 1.2.4.2 Digital Processing

Considering an A/D conversion feasible at RF frequencies ( $>10\text{GHz}$ ), a DSP should handle 16-bit words at  $10\text{GHz}$ . This is equivalent to more than 600Gops which implies a minimal power consumption of  $200\text{W}$ . This is not compatible with a handset limited by its battery life. Considering defined digital functions in the case of an ASIC, Moore's law is providing good hope to lower the huge power consumption coming from millions of instructions per seconds (MIPS). But, this progression is limited like ADCs and a 15-year perspective to obtain such chip is a very optimistic vision. In general, the performance of DSP will keep on increasing as chip sizes are reduced and the number of gates increased. New structures are also expected to improve DSP performance. Some technics such as decimation could lower the working frequency despite the fact that a part of the calculation would remain at RF frequencies. Figure 1.14 presents the power consumption per MIPS for DSP on the market thanks to a survey done in [14]. The barrier of the  $50\text{mW}$  at  $10\,000$  MIPS is said to be a step toward SR. It also exhibits how long the technical roadmap to a DSP adapted to SR architectures is.

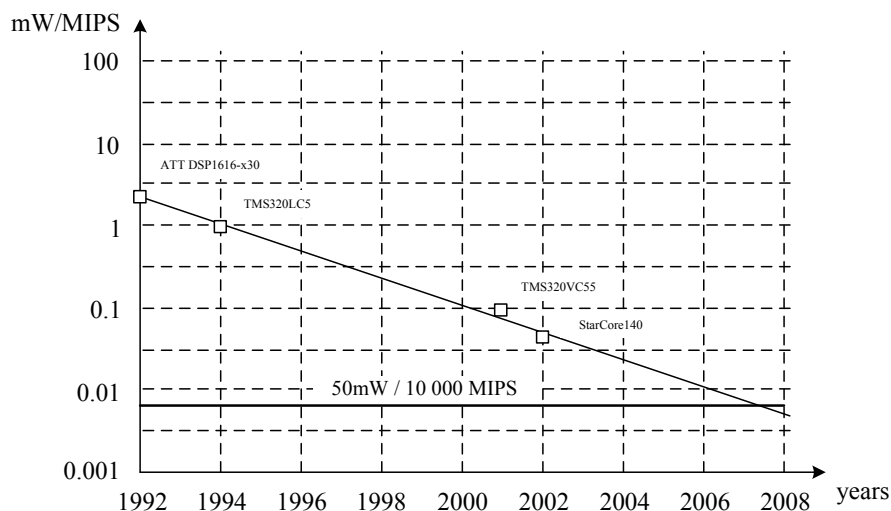


Figure 1.14: Power consumption per MIPS for DSP the last 20 years

### 1.2.4.3 RF Front End

RF Front End is also part of the technological bottleneck. It is composed by pure analog functions: LNA, Power Amplifier, Filter and Antenna. SR concept imposes these functions to be as wide-band as possible (0 to  $5\text{GHz}$  typically). This section presents an overview of the technical difficulties to design such functions in the case of a SR receiver.



## Antennas

A very wide-band antenna is required to cover the entire RF spectrum. Studies are done on "meta-materials" or MEMs (micro-switches) to target the unique SR antenna. But, a SR antenna is nowadays quite impossible to be realized and a panel of antennas, specifically designed for optimized performances are preferable.

## Amplifiers

LNA and PA topologies are based on few active (transistors) and passive components (inductors, capacitors). Technological reduction leads to decrease parasitic elements but, in the meantime, it lowers inductors performances. This enables to maintain good performances on dedicated devices which are narrow band and only used for specific standards. In the case of SR, LNA must be very wide-band (0 to 5GHz). Capacitors and inductors parasites must be as low as possible to guarantee a maximal linearity bandwidth. That is why new topologies are proposed to overcome these technological issues [15, 16].

## Conclusion and Technological strategies

This section has exhibited the technological bottlenecks constrained by mobile context. The power consumption is the main one. It is driven by the A/D conversion and the digital signal processing. Consequently, researches are to be focused on exploring new structures in rupture with the traditional ones. As the digital domain is an obstacle to a SR system realization, analog signal processing becomes a key. The strategy consequently adopted in this work to develop a SR chip is to design an Analog Signal Processor. The next part offers a state of the art of SDR architectures to highlight difficulties encountered by designers. It will draw a technical strategy for an analog signal processor conception.

### 1.2.5 Software Defined Radio Architectures

SDR and SR architectures only appeared the past few years. Researches are wide-spread and it is hardly possible to summarize SDR and SR works. This part gives clues to understand directions taken by designers to overcome technological bottlenecks. Solutions are proposed to bring closer and closer the digital part to the antenna. They are presented from the less flexible to the most advanced toward a SR system.

#### 1.2.5.1 Baseband Conversion

The first step toward a SR system is an analog and continuous translation into baseband of RF signals [17]. A direct conversion architecture processes analogically the frequency translation

thanks to mixers (Fig. 1.15). Once in baseband, the signal can be converted into digital after filtering (Fig. 1.16). The major advantage is relaxed ADC requirements (low frequency) but at the cost of narrowband SDR systems, limited by mixers and filters characteristics. This architecture is considered as multi-standard.

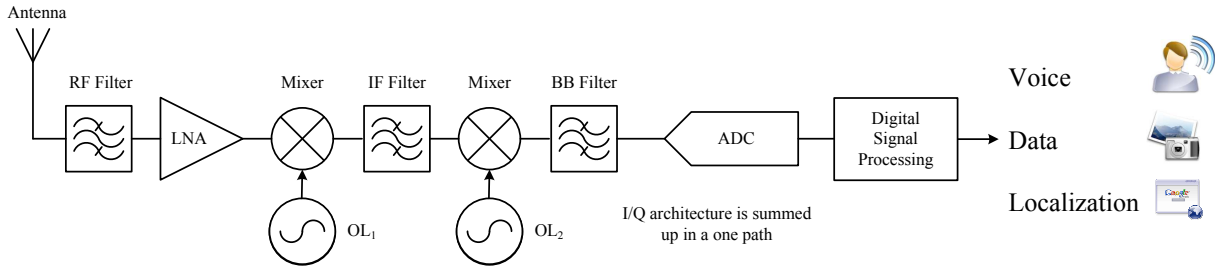


Figure 1.15: Software Defined Radio by Baseband Conversion Architecture

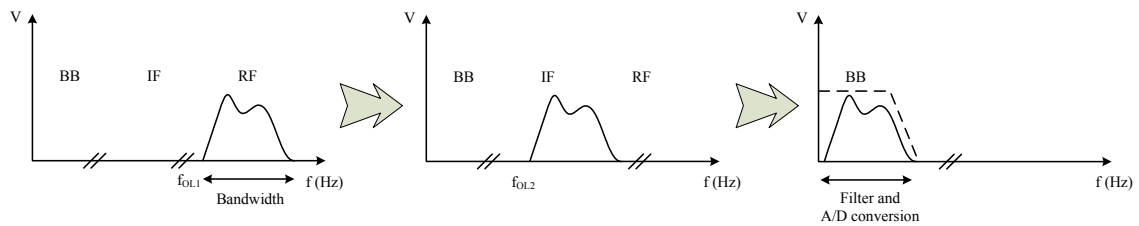


Figure 1.16: RF signal to baseband translation by Baseband Conversion Architecture

### 1.2.5.2 IF Conversion

The idea presented before is extended to IF frequencies. The A/D conversion is processed at IF. The baseband translation is thus done digitally and only one mixer is required to down convert RF signal to IF. The number of reconfigurable parameters is consequently reduced to the only mixer but stronger requirements are imposed to the ADC such as a wider access band [18].

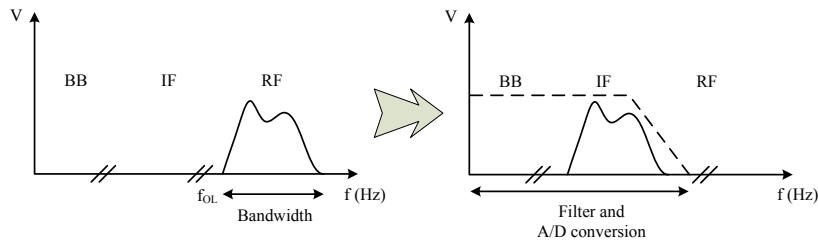


Figure 1.17: RF signal to IF translation by IF Conversion Architecture

### 1.2.5.3 Sub-sampling

The sub-sampling principle is to keep the idea of an ideal SR system (Fig. 1.7). The signal is sub-sampled to be converted into digital. It implies an aliasing effect. Thus, the RF signal is translated into baseband thanks to aliasing (Fig. 1.18) [19]. Once in baseband, the signal is filtered and converted into digital.

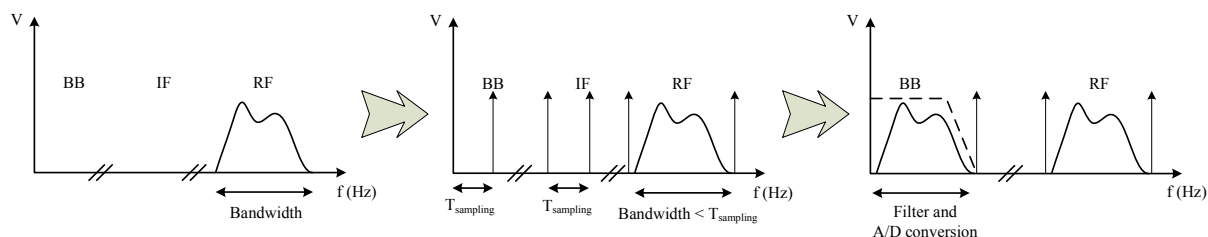


Figure 1.18: Software Defined Radio by sub-sampling

Despite the easiness of this principle, three main drawbacks are mentioned:

**Filter :** As described in figure 1.18, once in baseband the signal is filtered. A very selective filter has to be chosen to avoid any interference with neighbor channels.

**ADC bandwidth :** The maximal bandwidth of the ADCs must address any channel bandwidth (while guaranteeing other requirements: resolution, ...). Hopefully, ADC requirements are relaxed thanks to the baseband translation, and that drawback does not appear as unfeasible.

**Dynamic loss :** The Signal to Noise Ratio (SNR) is decreased. The dynamic range is reduced. This is the major drawback of such a system.

The sub-sampling solution is an interesting solution for SDR but technological issues are still strong to be solved. Besides, this system is narrowband and can concern only one standard at a time. Among the three solutions listed here, it is the closest to a SR architecture. The technical idea of sampling the signal directly at RF frequencies in order to keep any degree of flexibility is retained to design a wide-band SR system.

### 1.2.5.4 State of the Art

This section presents a state of the art of SDR architectures and technological ideas to provide reconfigurable structures. Five papers are chosen to give an overview of the recent researches in the domain.

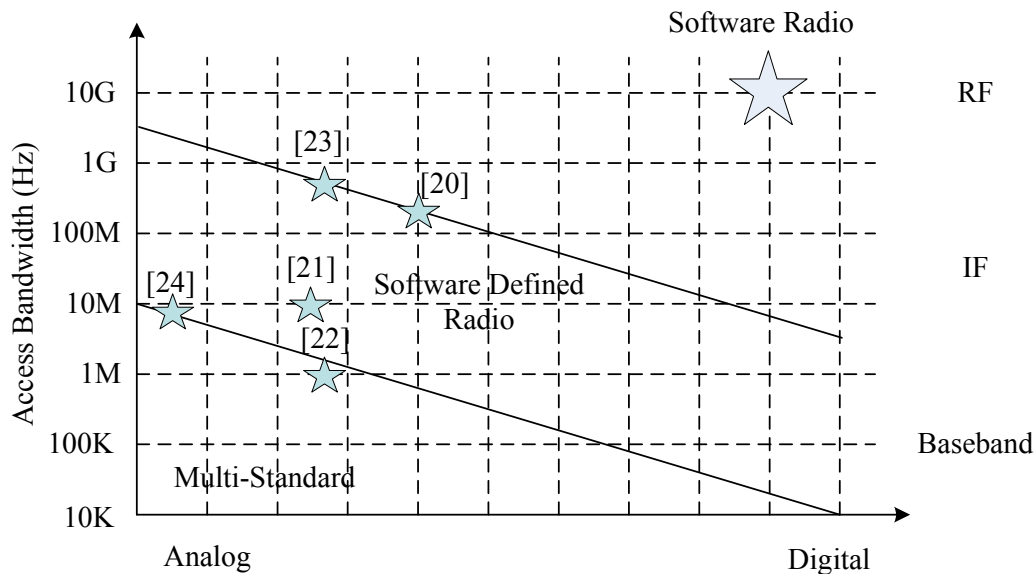


Figure 1.19: Software Defined to Software Radio State of the Art

- [20] proposes a receiver which makes use of a windowed integrator to perform charge sampling, which provides inherent anti-aliasing. The approach is multi standard. Programmable decimation filters attenuate unwanted channels and scale on the desired signal. It addresses both narrow and wide-band standards.
- [21] presents a demonstrator that receives both Bluetooth and HiperLan. Despite only two standards are targeted by the chip, the paper exposed the problems encountered for a SDR design. The two selected standards are different in terms of frequency band, signal bandwidth and modulation types. Challenges implied by the wide range of frequencies covered are exhibited. Main requirements are explained: ADC, LNA, mixer, antenna. A multi-standard architecture is shown.
- [22] depicts a discrete-time receiver. It is dedicated to Bluetooth standard. The down-conversion is processed analogically thanks to a discrete-time analog signal processing. As proven the feasibility of this architecture, it is proposed to extend it to meet requirements of several standards.
- [23] describes a full SDR receiver. It plays on the aliasing effect, such as [20], by a windowed integration. The receiver acts as a signal conditioner for the ADC by emphasizing only the wanted channel. The ADC requirements are relaxed thanks to analog signal processing. One step is done toward SR, overcoming the A/D technological bottleneck. The chip designed is able to handle GSM and 802.11g modes.

- [24] offers a discrete time analog signal processor as an alternative to the DSP. It means that a part of the processing realized normally into digital is moved to the analog part. The proposed system is a sampled-data system. It exposes the advantages, such as the suitability to real time applications, the trade-off between speed and accuracy. Switched capacitor array is proposed to perform analog operations.

This selection draws the trend in the design of SDR architectures: researches are not only focused on traditional structures with reconfigurable elements but are also oriented toward new structures. Multi-standard era is arrived at its maturity and it is common to find chips dealing with several standards [21]. As a great part of the job is still to overcome the A/D technological bottleneck, solutions must be found to relax ADC requirements. Discrete-time analog signal processing is proposed to be one of the solutions. The next part offers an overview of architectures tending to bring closer the digital part to the antenna and a state of the art of analog signal processors as a technical solution to new RF structures.

### 1.3 Analog Signal Processing

This thesis presents the design of a mixed signal architecture to overcome the A/D-conversion bottleneck. The RF signal is pre-conditioned analogically by a Sampled Analog Signal Processor (SASP) located between the LNA and the ADC (Fig. 1.20). The SASP does basic analog operations on discrete time voltage samples. The purpose is to reduce the RF signal data rate before digital conversion. The signal frequency has to be lowered. Analog operations give the opportunity to work directly at RF frequencies at acceptable power consumption and to display a low frequency output signal.

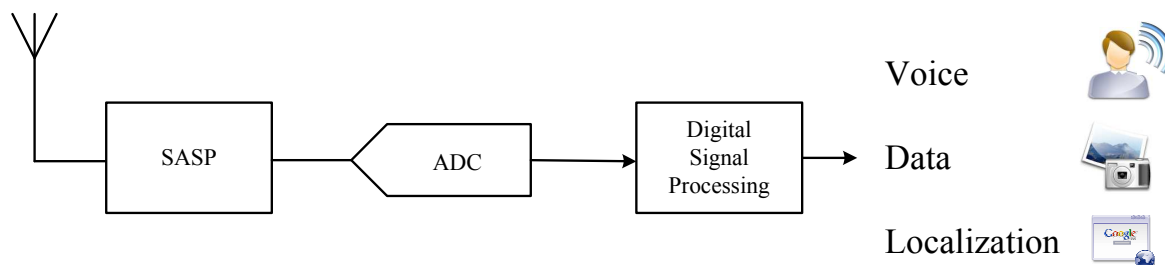


Figure 1.20: Proposed SR architecture

### Analog Applications

Sampled analog circuits are designed since the 70's. They were very used before the domination of digital circuit and gave many applications. A study on Charge Coupled Device (CCD) is a starting point of exploration on analog signal processing architectures. CCD was first introduced by Boyle and Smith [25]. It consists on storing charge in potential wells created at the surface of a semiconductor and moving the charge over the surface (Fig.1.21). Clock timing exhibits the moving charge in three operating phases. The number of carrier in a charge represents the value of the sampled signal. This is a continuous value. CCD enables integrating on a single chip capacitors, MOS switches and MOS sensors. A smaller silicon area is thus used and a higher yield is performed. One of the main interest in CCD is the low dissipated power because of the big part of capacitive device.

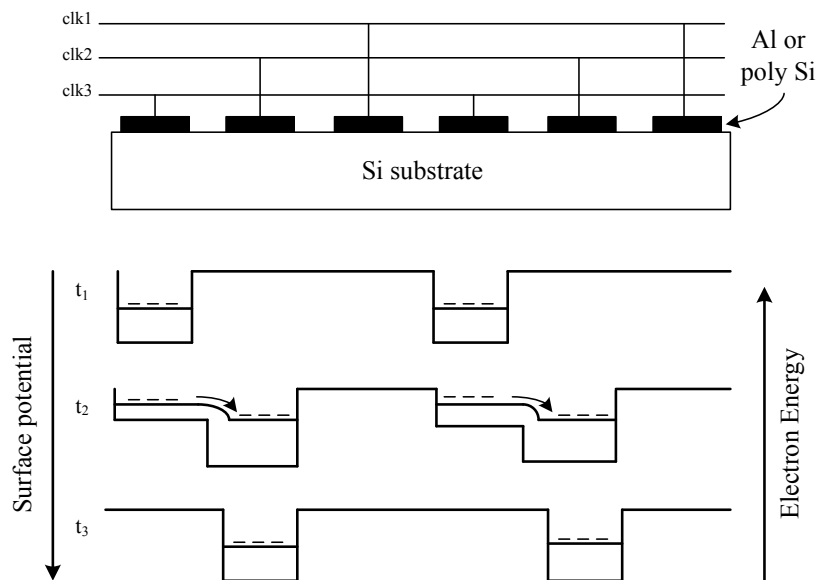


Figure 1.21: Charge Coupled circuit in 3 working phases

Many functions are designed thanks to this technology during the 70's. The easiest analog signal processing circuit designed with CCD is an input and an output, i.e. a delay line. Considered as a basic function, a delay line can give birth to more complex applications:

**Delay lines :** Delay lines are the first circuit designed thanks to CCD. The delay is provided thanks to the propagation time of the charges. It was mainly used for television applications as image correction circuits to delete ghost images [26].

**Filters :** One of the most successful application is discrete filtering. For instance, in [27], techniques are presented for making transversal filters using CCD. In a CCD transversal filter,

the delayed signals are sampled by measuring the current flowing in the clock lines during transfer, and the sampled signals are weighted by a split electrode technique (Fig. 1.22). Examples are given of CCD filters that are "matched" to particular signaling waveforms, and the limitations of charge-transfer devices (CTD) in matched filtering applications are discussed. Finally, the application of CTD transversal filters to other signal processing functions is debated.

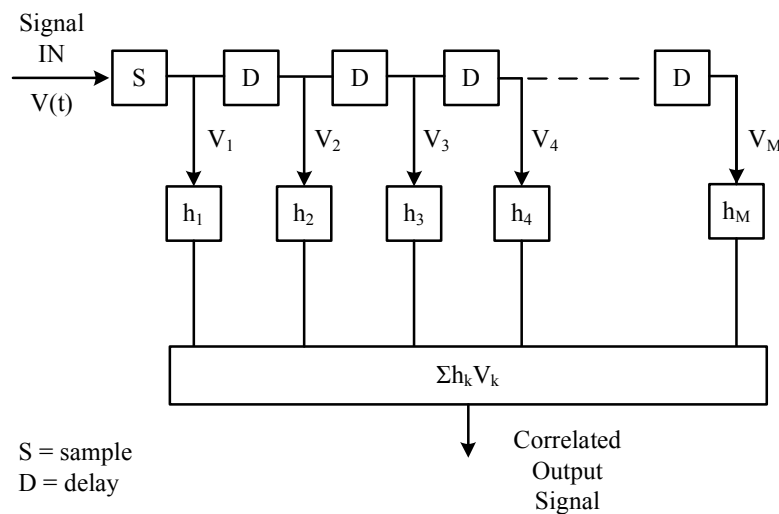


Figure 1.22: Block Diagram of a Transversal Filter

[28] displays an other approach. It shows a programmable analog transversal filter disclosed for processing analog signals and for receiving a series of discrete analog signals to be delayed by increasing periods and applied to the outputs of a CCD, and a plurality of MNOS memory devices. CCD and MNOS devices are coupled to the taps of the CCD and programmed so that the output of a CCD tap is weighted by a particular factor (dedicated to filtering in that case). This particular factor is, as it appears today, the first step toward a discrete analog signal processor. In fact, the factor (if it can be changed) can lead to different application not only as filtering but also as Fourier Transform, correlators and adaptive filters.

**Resonators :** Resonators are applied to filtering applications with a high quality factor. [29] exhibits a discrete analog Chebyshev filter. [30] shows a passive CCD resonator as a recursive CCD building block with a high-Q band pass filters. Advantages of this approach are an extremely low sensitivity of the center frequency which is determined by an external

clock frequency, a relative bandwidth which does not depend on the center frequency but is controlled by a capacitance ratio.

**Correlators :** An analog correlator performs the convolution of two analog functions. It can lead to a hardware economy in complex signal processing circuit. A basic structure can be found in [31] and depicted in figure 1.23. A 20-stage analog CCD correlator has been constructed in [31]. Four quadrant analog multipliers have been used to combine the voltages from CCD taps to produce a correlation output. Multipliers achieve 1% linearity. Correlators are a new step toward complex analog signal processor because of their flexibility given by an operation between two variables.

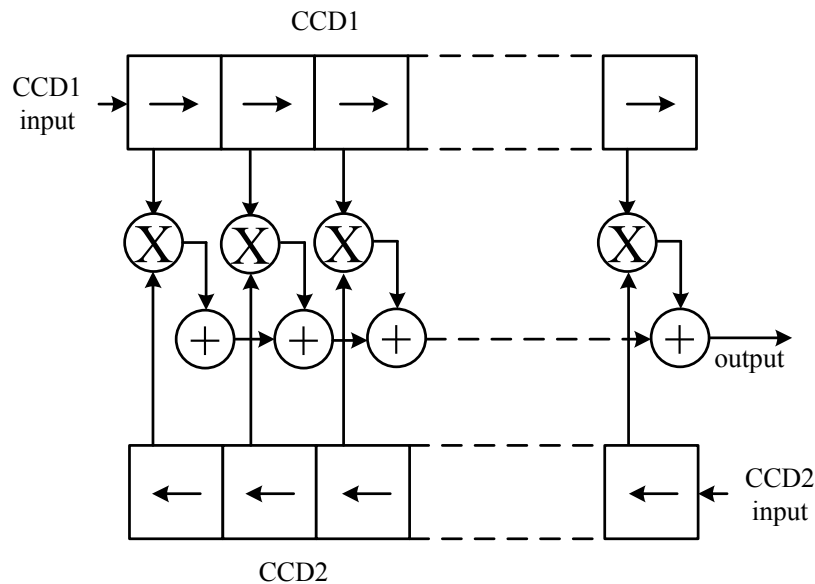


Figure 1.23: Block Diagram of a Correlator

**Chirp Transform :** It is a signal processing algorithm which enables to process z-transform of a sampled signal. As described in [32], the Chirp Transform is based on CCD Correlators and CCD Delay Lines (Fig. 1.24). The design of complex applications thanks to basic ones is demonstrated. The Chirp Transform validates the transformation of a temporal signal into another domain (here z-domain). New applications can be envisaged in optimized domain, for instance in the spectral domain [33, 34, 35, 36]. That point is crucial in the research of disruptive architectures as it allows dreaming to new structures offering best requirements.



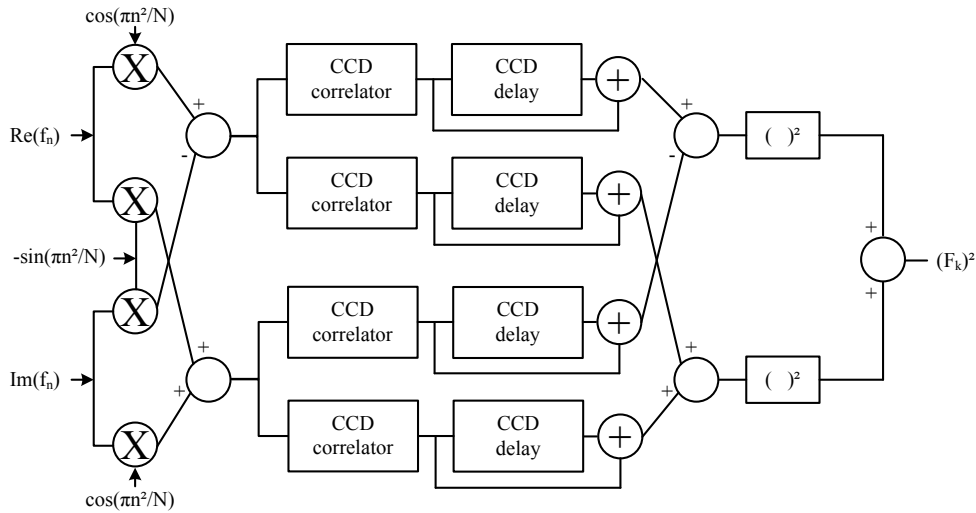


Figure 1.24: Block Diagram of a Chirp Transform

Table 1.3: Charge Coupled Devices Sum up

References	[26]	[27]	[29]	[30]	[31]	[36]	[34]
Application	Delay Lines	Filter	Resonator	Resonator	Correlator	FFT	FFT
Year	1979	1973	1981	1981	1976	1978	1975
$f_{max}$	-	20MHz	4.75MHz	100kHz	5MHz	1MHz	10MHz

Researches in the 70's on CCD in the case of analog signal processing enable to offer a wide range of applications. Table 1.3 gives an overview of these applications and their maximal frequency of operation. Thus, discrete time analog signal processing can be a solution to find new architectures dedicated to SDR or SR to overcome the A/D technological bottleneck. If several megahertz was the maximal frequency and thanks to technology improvements, circuits designed 25 years ago may be scaled in the most recent technology and find an application at gigahertz frequencies. It allows thinking that RF functions can be processed directly in the analog domain. All constraints on the ADC and DSP are relaxed because major part of the signal processing is shifted in analog domain. A Sampled Analog Signal Processor (SASP) is technically feasible and its goal to relax ADC and DSP requirements can be considered to meet the requirements of Software Radio concept.

## 1.4 Conclusion

Telecommunication industry claims for new RF architectures. It is faced with the diversity and the **multimedia convergence** in mobile terminals. A concept called **Software Radio** brings the flexibility required by the market constraints. But, in the case of handsets, **technological bottlenecks** prevent from realizing this concept. The A/D conversion is the crucial part but it is limited by the current technology and the high power consumption. Hence, new analog architectures are explored. This thesis is focused on a **discrete time, analog** solution. State of the art has demonstrated the feasibility of discrete time analog signal processors. A Software Radio chip called **Sampled Analog Signal Processor (SASP)** is to be designed in the most recent technology from STMicroelectronics. The SASP processes RF signals and displays information at low frequency to an ADC. This chip is able to handle any RF signal and can reconfigure itself to accept any RF standard.

Technical options remain to implement a hardware **algorithm** thanks to **discrete time analog voltage samples** processing. Chapter 2 presents the chip architecture and the algorithm used. Chapter 3 depicts behavioral simulations and schematics design. Chapter 4 describes the chip layout, the technical issues and measurements.



---

# Sampled Analog Signal Processor

---

## Contents

---

<b>2.1</b>	<b>Principle . . . . .</b>	<b>52</b>
2.1.1	Analog Signal Processor Principle . . . . .	52
2.1.2	Frequency Translation . . . . .	53
2.1.3	A Fourier Transform . . . . .	56
<b>2.2</b>	<b>A Fast Fourier Transform . . . . .</b>	<b>57</b>
2.2.1	The Cooley-Tukey algorithm . . . . .	57
2.2.2	A pipelined DFT . . . . .	60
<b>2.3</b>	<b>Architecture . . . . .</b>	<b>67</b>
2.3.1	Signal pre-processing . . . . .	67
2.3.2	DFT implementation . . . . .	73
2.3.3	Post-signal processing . . . . .	76
<b>2.4</b>	<b>A Software Radio System . . . . .</b>	<b>78</b>
2.4.1	Concurrent reception . . . . .	78
2.4.2	Frequency demodulation . . . . .	79
<b>2.5</b>	<b>Conclusion . . . . .</b>	<b>82</b>

---

*Chapter 2 presents the principle of the Sampled Analog Signal Processor (SASP). A state of the art has exposed the challenges of a Software Radio system. As many technological bottlenecks need to be overcome, an idea is to design a discrete analog signal processor. The main bottleneck of the A/D conversion is thus avoided and a lower processing speed can be envisaged concerning the digital part of the architecture. The SASP aims at selecting a spectral envelope of a RF signal among all RF signals. To reach this target, the SASP processes analogically the RF input signal spectrum thanks to an analog Discrete time Fourier Transform (DFT). Once the spectrum is processed, voltage samples representing the spectral signal envelope to be treated are converted into digital. The selection of few voltage samples among thousands replaces the classical mixing and filtering operations. It reduces the A/D conversion frequency from GHz frequencies to MHz ones. Algorithm and design strategy are presented. Finally, applications are proposed.*

**Key words:** Sampled Analog Signal Processor (SASP), Discrete Fourier Transform, frequency translation, concurrent reception, frequency demodulation

## 2.1 Principle

The Sampled Analog Signal Processor (SASP) principle will be presented. This design is challenged by two ways:

- Discrete time analog voltage samples are used to analogically process RF signal.
- A Fourier Transform is the implemented algorithm to address the Software Radio concept.

This section presents the principle of the SASP.

### 2.1.1 Analog Signal Processor Principle

A mixed signal architecture is proposed to overcome the A/D-conversion bottleneck. The RF signal is pre-conditioned analogically by the SASP located between the antenna and the ADC (Fig. 2.1). The SASP does basic analog operations on discrete-time voltage samples. The

purpose is to reduce the RF signal data rate before digital conversion. The signal frequency has to be lowered. Analog operations give the opportunity to operate directly at RF frequencies at an acceptable power consumption given by mobile terminals constraints and to display a low frequency output signal.

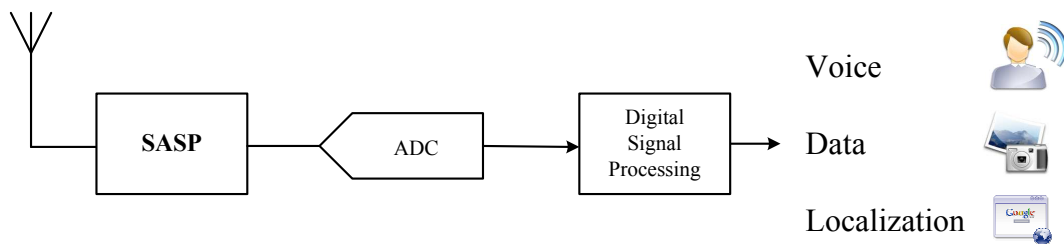


Figure 2.1: Proposed SR architecture

In order to lower the output data rate, the idea of frequency translation was chosen.

### 2.1.2 Frequency Translation

The SASP principle is based on the frequency translation (Fig. 2.2). The idea is to work in frequency domain instead of working in time domain. The constatation is that a RF signal envelope varies slowly compared to its carrier frequency. Signal processing on the RF signal envelope is focused. Given the criterium of reconfigurability constrained by SR, the SASP aims at processing any RF signal envelope in a 0 to 5GHz frequency range. Consequently, it is decided to work on all the RF frequency spectrum.

The principle of the SASP is to receive any RF signal. It processes analogically a Fast Fourier Transform (FFT) with discrete time voltage samples. Voltage sample carry out analog signal processing. They are the way to transform a temporal discrete signal in a frequencial signal. The frequencial signal is displayed in the time domain. A frequency image of all the RF spectrum is thus given. The information of RF input signal is still contained in discrete voltage samples.

Figure 2.2 exposes the purpose which is to extract the desired input signal spectrum envelope by recovering its spectrum. The RF signal is sampled and transformed. Among thousands of voltage samples, only the ones representing the RF signal envelope are sent out towards an A/D converter (Fig. 2.3). The selection provides a frequency translation as once in digital domain, the signal is considered to be baseband. An Inverse Fast Fourier Transform (IFFT) or an optimized signal processing is done to demodulate the RF signal.

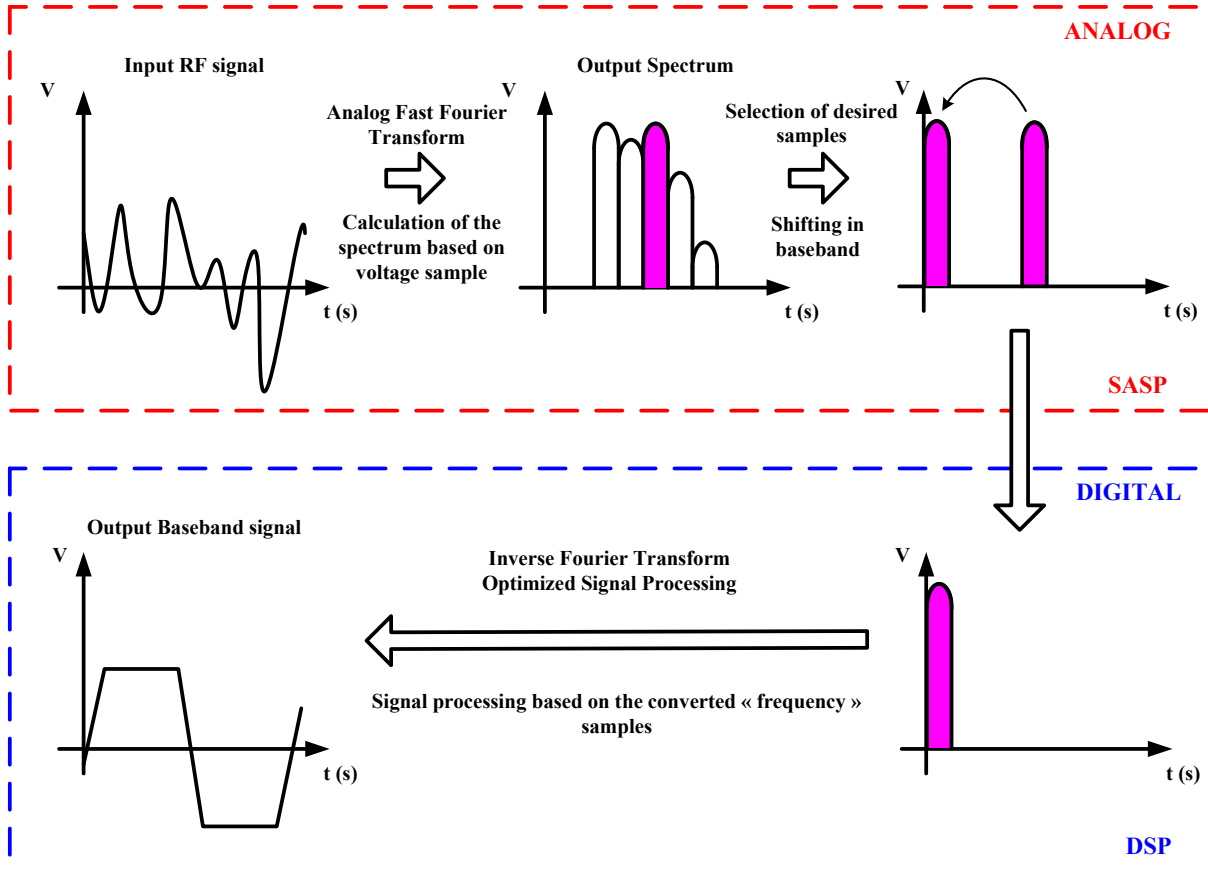


Figure 2.2: Principle of the frequency translation

The goals of SR are fulfilled thanks to 2 major choices:

- The FFT enables to switch from time to frequency domain. All the RF spectrum is considered and any RF signal can be thus processed. The system is consequently wideband.
- Discrete time voltage samples guarantee the signal resolution. Their selection before the A/D conversion reduces the data rate and requires a low frequency conversion and a low working frequency of the DSP.

The example of a pure sinewave is given. Figure 2.4 depicts the FFT of such a signal. The FFT is a Dirac at the frequency of the signal. Figure 2.5 exhibits an Amplitude Modulation (AM). It is characterised by a carrier ( $f_{\text{carrier}}$ ) and a modulated signal ( $f_{\text{signal}}$ ). The FFT is given by a Dirac at the frequency of  $f_{\text{carrier}} + f_{\text{signal}}$ . The SASP enables to select the Dirac and translate it into baseband in the frequency domain. Once done, it is just a matter to process an IFFT to recover the modulated signal. Consequently, the principle of the SASP is to remove any RF

carrier of RF signals. Figure 2.5 represents both time domain and frequency domain whereas the SASP only works in time domain. In the case of the SASP, frequency domain is projected in time domain.

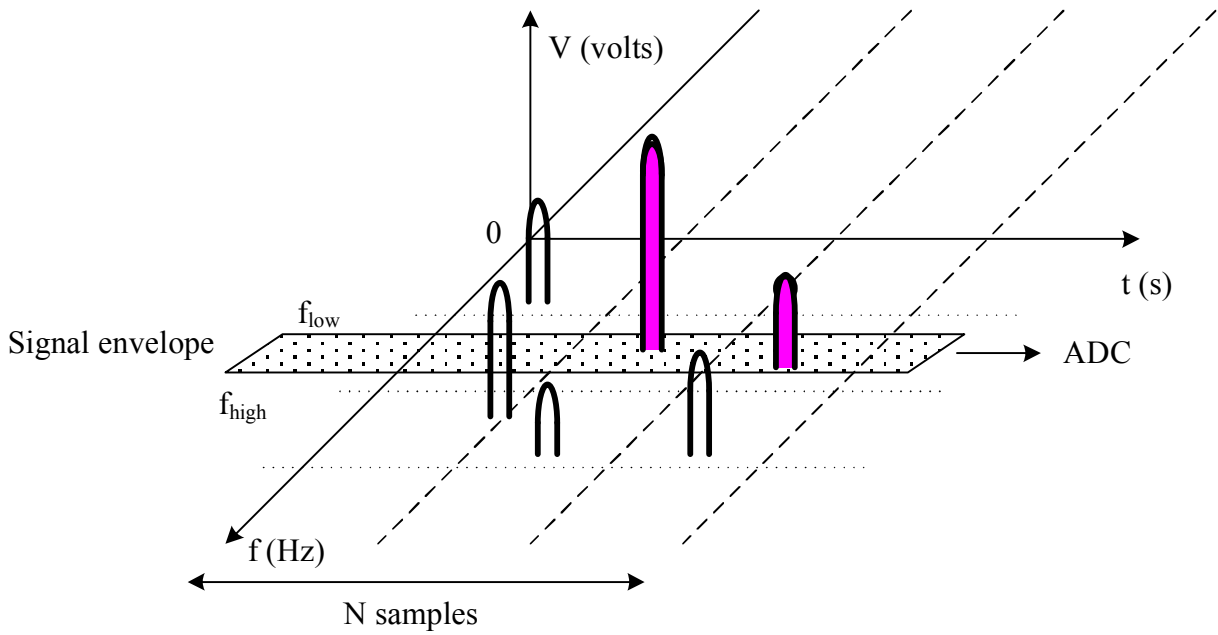


Figure 2.3: Envelope selection and digitization

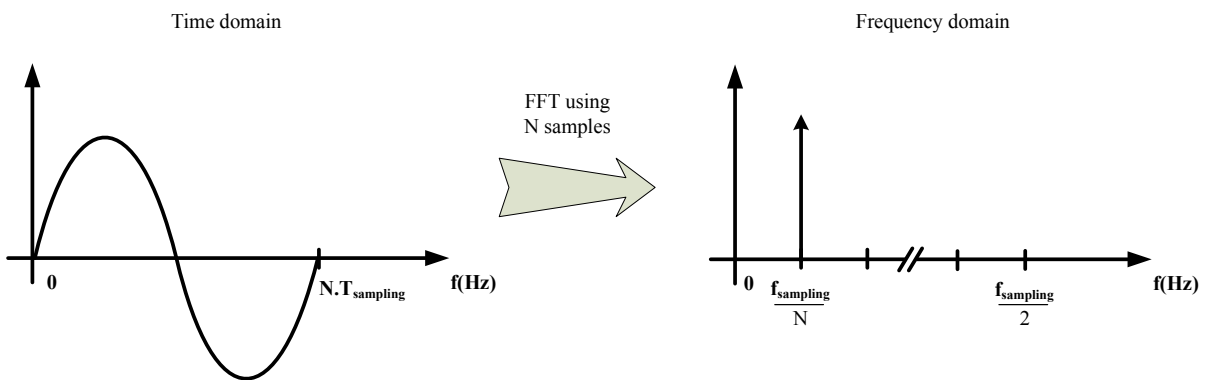


Figure 2.4: Sinewave FFT



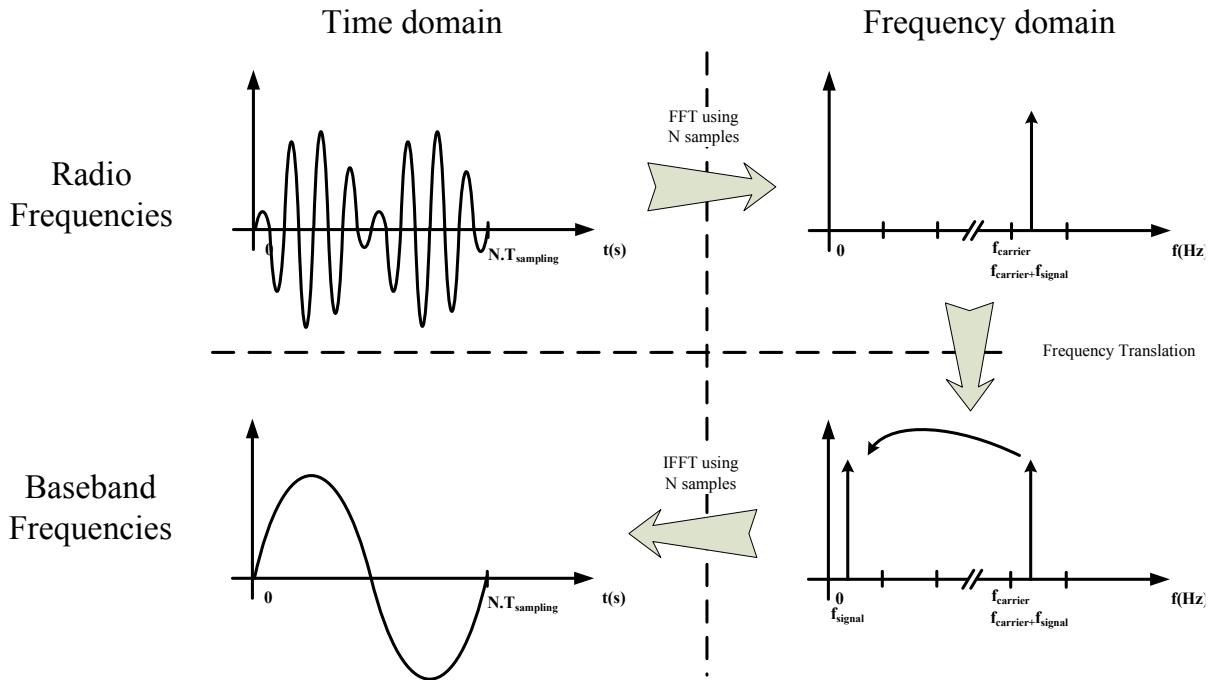


Figure 2.5: Frequency Translation of a modulated signal

### 2.1.3 A Fourier Transform

The SASP aims at processing the RF input signal spectrum. The spectral envelope of the target RF signal is analogically selected and sent to the ADC (Fig. 2.2). To carry out the operation, the SASP implements an analog Discrete Fourier Transform (DFT). Parameters inherited from the DFT equation (Eq. 2.2) master the SR requirements such as reprogrammability and flexibility: the sampling frequency  $f_{\text{sampling}}$  and the number of voltage samples  $N$  are taken into account to determine the spectral accuracy.

The Fourier Transform equation in continuous time domain is given by:

$$X(f) = \int_{-\infty}^{+\infty} x(t) \exp(-j2\pi ft) dt \quad (2.1)$$

The SASP is based on a discretization of the input signal  $x(t)$ .  $x(t)$  is sampled at a period of  $T_{\text{sampling}}$  in an infinite range of time. The Discrete time Fourier Transform is thus given by:

$$\begin{aligned}
x(k) &= x(t)_{t=kT_{\text{sampling}}} \\
X(\nu) &= \sum_{k=-\infty}^{+\infty} x(k) \exp(-j2\pi\nu kT_{\text{sampling}}), \quad -\infty < \nu < +\infty
\end{aligned} \tag{2.2}$$

The Discrete time Fourier Transform calculation is limited to a finite number of points  $N$ . As  $N$  samples are selected in the time domain,  $N$  frequencies are processed thanks to a Discrete Fourier Transform (DFT).  $n_{\text{sample}}$  is the frequency sample number in a DFT processing sequence. It gives the equation:

$$X(n_{\text{sample}}) = \sum_{k=0}^{N-1} x(k) \exp\left(\frac{-j2\pi n_{\text{sample}}k}{N}\right), \quad n_{\text{sample}} = 0, 1, \dots, N-1 \tag{2.3}$$

Equation 2.3 requires:

- $N$  equations to represent each frequency.
- For each equation,  $N$  complex multiplications and  $N-1$  complex additions.

In terms of operations, equation 2.3 needs  $N^2$  complex multiplications and  $N(N-1)$  complex additions.  $N$  is an integer that can be very high. Consequently, the signal processing asked by a DFT can cost a lot of processing complexity and consequently in die area and power consumption. An other solution has to be found: a FFT.

In order to simplify the SASP architecture, the FFT algorithm [37] in its pipeline form was used [38, 39]. It reduces the number of operations from  $N^2$  to  $N \cdot \log_2(N)$ . [38, 39] efficiently computes a DFT.

## 2.2 A Fast Fourier Transform

### 2.2.1 The Cooley-Tukey algorithm

The starting point of a Fast Fourier Transform (FFT) is given by the Cooley-Tukey algorithm [37]. Let us take an example to illustrate the way to switch from DFT to FFT defined by [37]. For a matter of simplicity without loss of generality,  $N = 4$ .

It is chosen to define  $W_N = \exp\left(\frac{-j2\pi}{N}\right)$ . It can be noticed that  $(W_N)^k = W_N^k$  where  $W_N^k$  is defined as a twiddle factor and  $k$  as a  $n^{\text{th}}$  root of unity index.  $x(n)$  becomes  $x_i(n)$  where  $i$  is an indice to indicate a processing phase to be used latter.  $n$  is a local variable to point out the voltage sample from 0 to  $N-1$  i.e. 0 to 3. Equation 2.3 can be developed in 4 equations:

$$\begin{aligned}
X(0) &= x_0(0)W_4^0 + x_0(1)W_4^0 + x_0(2)W_4^0 + x_0(3)W_4^0 \\
X(1) &= x_0(0)W_4^0 + x_0(1)W_4^1 + x_0(2)W_4^2 + x_0(3)W_4^3 \\
X(2) &= x_0(0)W_4^0 + x_0(1)W_4^2 + x_0(2)W_4^4 + x_0(3)W_4^6 \\
X(3) &= x_0(0)W_4^0 + x_0(1)W_4^3 + x_0(2)W_4^6 + x_0(3)W_4^9
\end{aligned} \tag{2.4}$$

Equation 2.4 is equivalent to a matrix product given by:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} \cdot \begin{bmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{bmatrix} \tag{2.5}$$

$$\mathbf{X}(n) = \mathbf{W}_N^{nk} \mathbf{x}_0(k) \tag{2.6}$$

Some simplifications can be performed on the matrix  $\mathbf{W}_N^{nk}$  for  $nk = 0, 1, \dots, 9$ . In the case of  $N = 4$ , let us observe the trigonometric circle in figure 2.6.

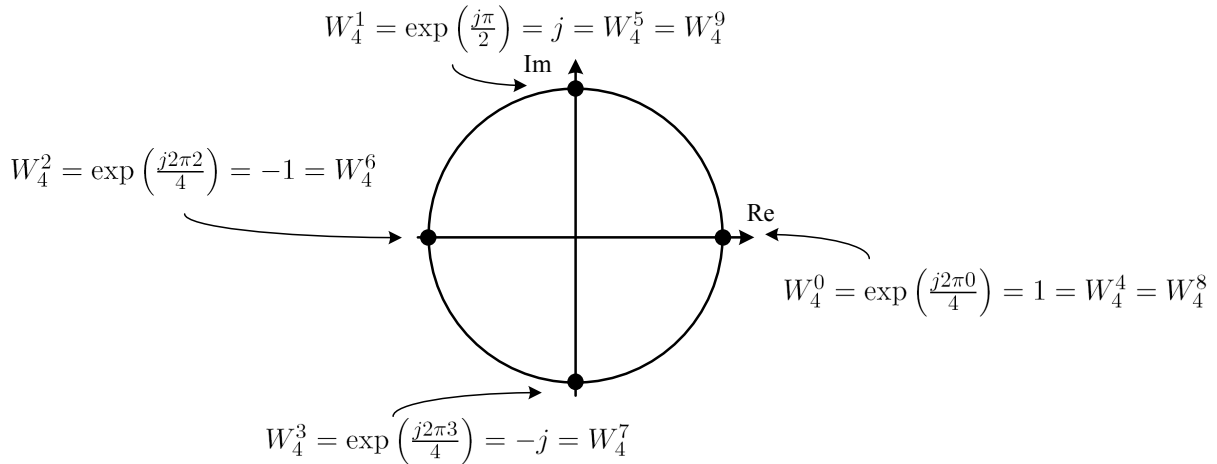


Figure 2.6:  $W_N^{nk}$  properties

$W_N^{nk}$  is periodic. The period is  $N$  because  $W_N^{nk} = W_N^{nk \bmod N}$ . As  $W_N^0 = W_4^0 = 1$ , equation 2.5 can be simplified in:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & W_4^3 \\ 1 & W_4^2 & W_4^0 & W_4^2 \\ 1 & W_4^3 & W_4^2 & W_4^1 \end{bmatrix} \cdot \begin{bmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{bmatrix} \tag{2.7}$$

The number of twiddle factors  $W_N^{nk}$  has been decreased from  $(N-1)^2 + 1$   $W_N^{nk}$  to  $N$   $W_N^{nk \bmod N}$ . The matrix  $\mathbf{W}_N^{nk}$  can be factorized in order to let appear the redundancies. It is decided to write  $k$  and  $n$  under their binary form:  $k = 0, 1, 2, 3 \Rightarrow k = (k_1, k_0) = 00, 01, 10, 11$ ,  $n = 0, 1, 2, 3 \Rightarrow n = (n_1, n_0) = 00, 01, 10, 11$ , which can be written  $k = 2k_1 + k_0$  and  $n = 2n_1 + n_0$ . In the example  $N = 4$ , the DFT is:

$$X(n) = \sum_{k_0=0}^1 \sum_{k_1=0}^1 x_0(k_1, k_0) W_4^{(2n_1+n_0)(2k_1+k_0)}, \quad n_{1,0} = 0, 1 \quad (2.8)$$

A factorization can be done:

$$\begin{aligned} W_4^{(2n_1+n_0)(2k_1+k_0)} &= W_4^{2k_1(2n_1+n_0)} W_4^{k_0(2n_1+n_0)} \\ &= W_4^{4n_1k_1} W_4^{2n_0k_1} W_4^{k_0(2n_1+n_0)} \end{aligned} \quad (2.9)$$

but, it can be noticed that  $W_4^{4n_1k_1} = [W_4^4]^{n_1k_1} = [1]^{n_1k_1} = 1$ . Hence,

$$X(n_1, n_0) = \sum_{k_0=0}^1 \left[ \sum_{k_1=0}^1 x_0(k_1, k_0) W_4^{2n_0k_1} \right] W_4^{k_0(2n_1+n_0)}, \quad n_{1,0} = 0, 1 \quad (2.10)$$

The algorithm of the DFT appears from equation 2.10.  $\left[ \sum_{k_1=0}^1 x_0(k_1, k_0) W_4^{2n_0k_1} \right]$  depends only on the variable  $k_0$  and thus a 2-step pipeline calculation can be described. Let us write  $x_1(n_0, k_0) = \left[ \sum_{k_1=0}^1 x_0(k_1, k_0) W_4^{2n_0k_1} \right]$ . The matrix form is given by:

$$x_1(n_0, k_0) = \begin{bmatrix} x_1(0, 0) \\ x_1(0, 1) \\ x_1(1, 0) \\ x_1(1, 1) \end{bmatrix} = \begin{bmatrix} x_0(0, 0) + x_0(1, 0) W_4^0 \\ x_0(0, 1) + x_0(1, 1) W_4^0 \\ x_0(0, 0) + x_0(1, 0) W_4^2 \\ x_0(0, 1) + x_0(1, 0) W_4^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & W_4^0 & 0 \\ 0 & 1 & 0 & W_4^0 \\ 1 & 0 & W_4^2 & 0 \\ 0 & 1 & 0 & W_4^2 \end{bmatrix} \cdot \begin{bmatrix} x_0(0, 0) \\ x_0(0, 1) \\ x_0(1, 0) \\ x_0(1, 1) \end{bmatrix} \quad (2.11)$$

It gives:

$$\begin{aligned} X(n_1, n_0) &= \sum_{k_0=0}^1 x_1(n_0, k_0) W_4^{k_0(2n_1+n_0)}, \quad n_{1,0} = 0, 1 \\ &= x_2(n_0, n_1) \end{aligned} \quad (2.12)$$

Under its matrix form:

$$\begin{bmatrix} x_2(0, 0) \\ x_2(0, 1) \\ x_2(1, 0) \\ x_2(1, 1) \end{bmatrix} = \begin{bmatrix} 1 & W_4^0 & 0 & 0 \\ 1 & W_4^2 & 0 & 0 \\ 0 & 0 & 1 & W_4^1 \\ 0 & 0 & 1 & W_4^3 \end{bmatrix} \cdot \begin{bmatrix} x_1(0, 0) \\ x_1(0, 1) \\ x_1(1, 0) \\ x_1(1, 1) \end{bmatrix} \quad (2.13)$$

Data processed by this algorithm are output in an order defined by  $X(n_1, n_0) = x_2(n_0, n_1)$  (Eq. 2.12). This is a reverse binary order. The result matrix is finally given by:

$$\begin{bmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & W_4^0 & 0 & 0 \\ 1 & W_4^2 & 0 & 0 \\ 0 & 0 & 1 & W_4^1 \\ 0 & 0 & 1 & W_4^3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & W_4^0 & 0 \\ 0 & 1 & 0 & W_4^0 \\ 1 & 0 & W_4^2 & 0 \\ 0 & 1 & 0 & W_4^2 \end{bmatrix} = \begin{bmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{bmatrix} \quad (2.14)$$

This section has described the Cooley-Tukey algorithm with an example for a matter of simplicity. The next part presents the implementation of this algorithm as a pipelined DFT.

### 2.2.2 A pipelined DFT

Let us generalize this algorithm at any order  $N$ . The Cooley-Tukey algorithm [37] is a pipelined algorithm. In the case on  $N = 4$ , the algorithm is depicted in figure 2.7. In this figure, it can be noticed that a module is used several times to perform the calculation (Fig. 2.8). Relations between inputs and outputs are given in equation 2.15.

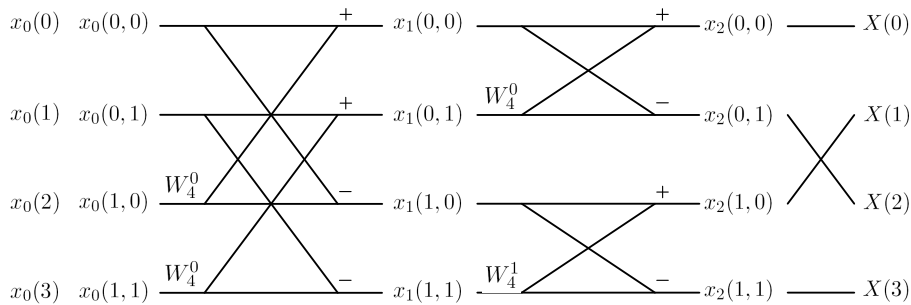


Figure 2.7:  $N = 4$  Pipelined DFT

An implementation of this module is proposed in [38, 39]. The processing is said to be recursive as the second term is calculated thanks to the first one.

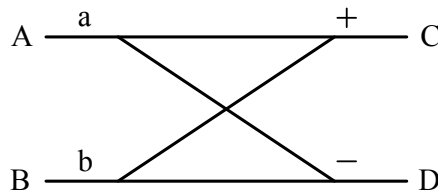


Figure 2.8: Module of radix-2 DFT

$$\begin{aligned} C &= aA + bB \\ D &= aA - bB \end{aligned} \tag{2.15}$$

Data are a constant flow of information. Each data is represented by a voltage sample. The algorithm has to be described as one-input one-output unit. It receives serial data and outputs serial data. Let us think about the implementation of the previous pipeline DFT. The best way to exhibit the algorithm is a graphical explanation. A basic cell is designed to embed every pipelined modules of the DFT (Fig. 2.9). It is composed by:

- A delay line. The length of the delay line is equal to  $2^{\frac{N}{r_{\text{stage}}}}$  where  $r_{\text{stage}}$  is the stage number in the pipelined DFT.
- A processing unit. It has 2 inputs and 2 outputs. It is the implementation of the module described previously (Fig. 2.8). It is ruled by the equations:

$$\begin{aligned} x_{out} &= x_{in} + y_{in}W_N^z \\ y_{out} &= x_{in} - y_{in}W_N^z \end{aligned} \tag{2.16}$$

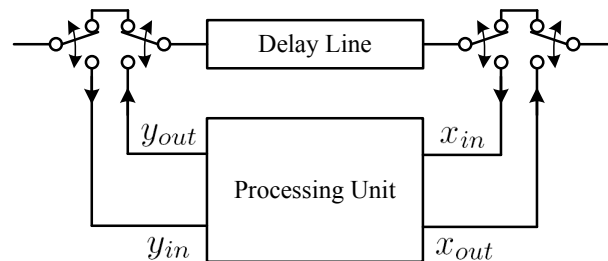


Figure 2.9: Basic cell of a pipelined DFT

Let us describe the signal processing done in the case of  $N = 4$ . The number of cell is 2.

The first 2 data  $x_0(0, 0)$  and  $x_0(0, 1)$  are stored in the delay line (length=2) (Fig. 2.10).

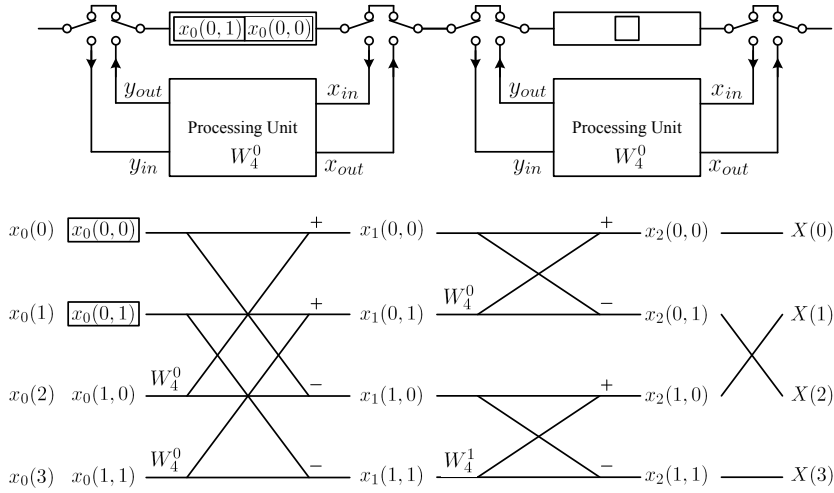


Figure 2.10: Step 1

$x_0(0, 0)$  is output from the delay line. The third data  $x_0(1, 0)$  is injected directly by the input. They are both directed toward the processing unit (Fig. 2.11).

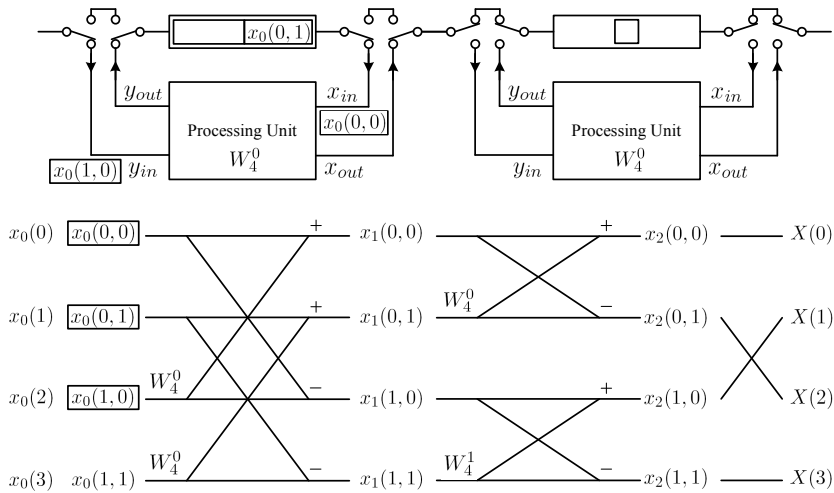


Figure 2.11: Step 2

The first processed data  $x_{out} = x_1(0, 0)$  is output from the processing unit and directed to the second stage. At the same time,  $y_{out} = x_1(1, 0)$  is directed to the delay line and takes the place of  $x_0(0, 0)$  (Fig. 2.12).

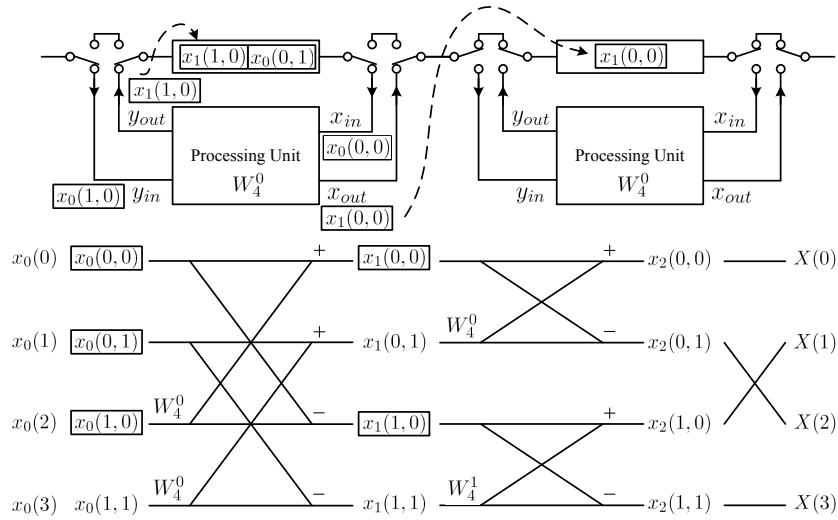


Figure 2.12: Step 3

$x_0(0, 1)$  is output from the delay line. The fourth data  $x_0(1, 1)$  is injected directly by the input. They are both directed toward the processing unit (Fig. 2.13).

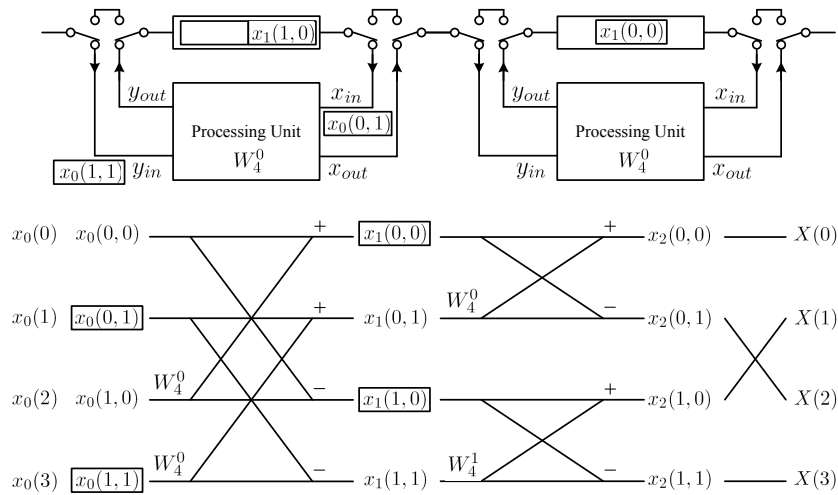


Figure 2.13: Step 4



The processed data  $x_{out} = x_1(0, 1)$  is output from the processing unit and directed to the second stage. At the same time,  $y_{out} = x_1(1, 1)$  is directed to the delay line and takes the place of  $x_0(0, 0)$  (Fig. 2.14). On the second stage,  $x_1(0, 0)$  is output from the delay line and directed toward the processing unit while  $x_1(0, 1)$  goes directly from the first stage toward the second stage processing unit.

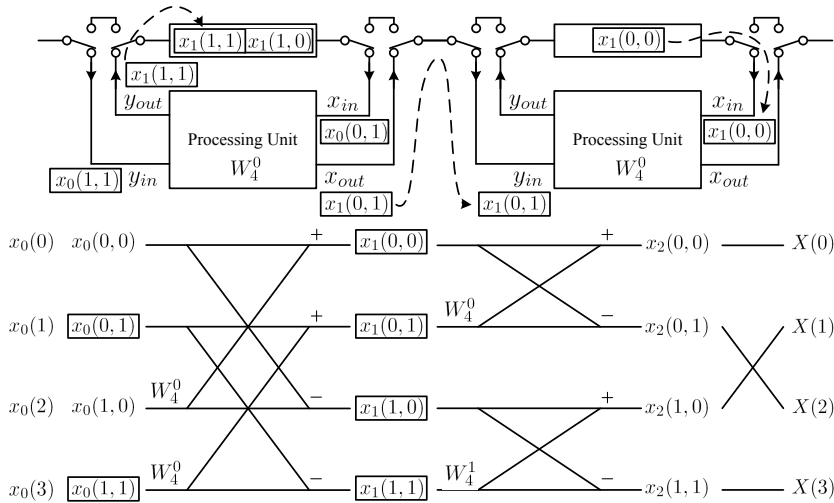


Figure 2.14: Step 5

$x_1(1, 1)$  is output from the first stage delay line and directed toward the second stage delay line. At the same time,  $x_2(0, 0)$  and  $x_2(0, 1)$  are processed (Fig. 2.15).

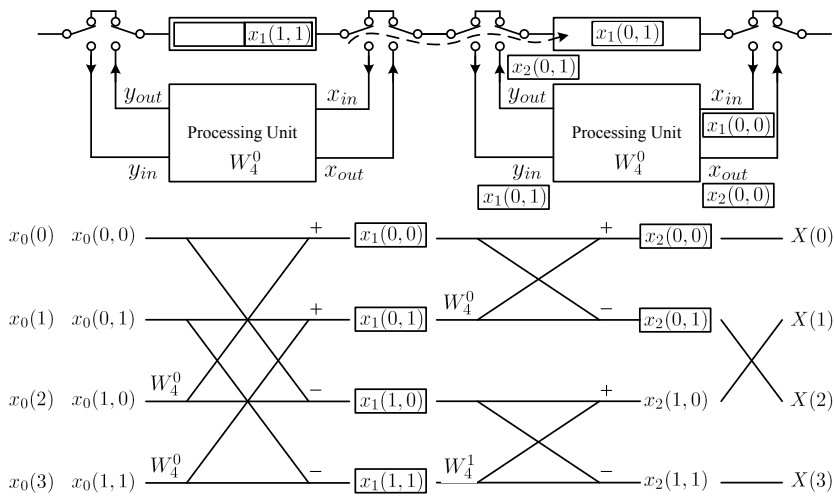


Figure 2.15: Step 6

$x_2(0,0)$  is output and  $x_2(0,1)$  is stored in the delay line to be output on the next step as a serialized data (Fig. 2.16).

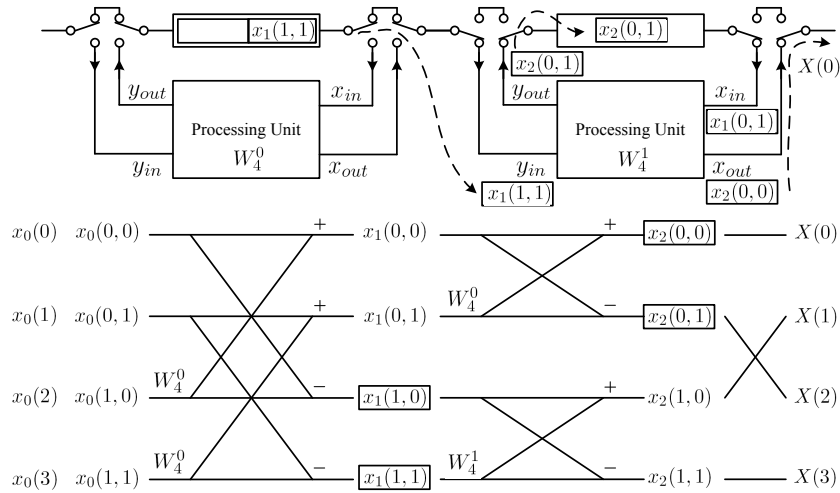


Figure 2.16: Step 7

At the same time,  $x_1(1,0)$  and  $x_1(1,1)$  are processed to give the data  $x_2(1,0)$  and  $x_2(1,1)$ . These data are output on the same way the first were: one is output directly, one is stored in the delay line to be output in series (Fig. 2.17, 2.18).

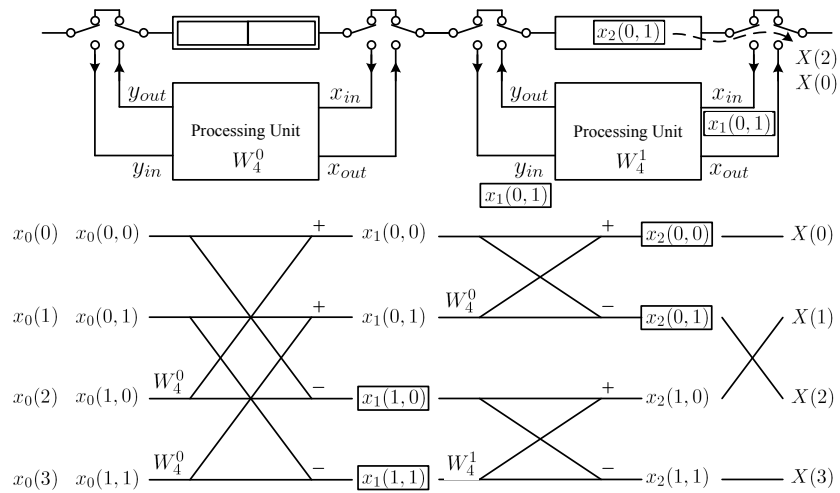


Figure 2.17: Step 8

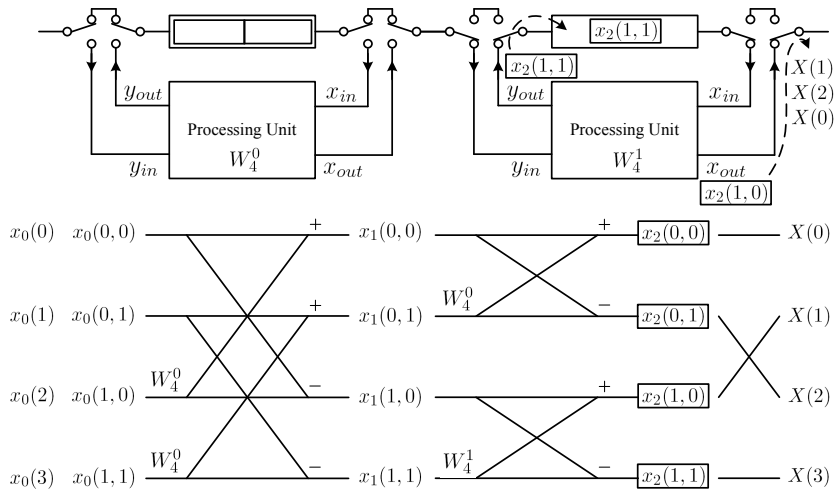


Figure 2.18: Step 9

All the data are output as serial data  $X(0)$ ,  $X(2)$ ,  $X(1)$ ,  $X(3)$  in binary-reverse order (Fig. 2.19).

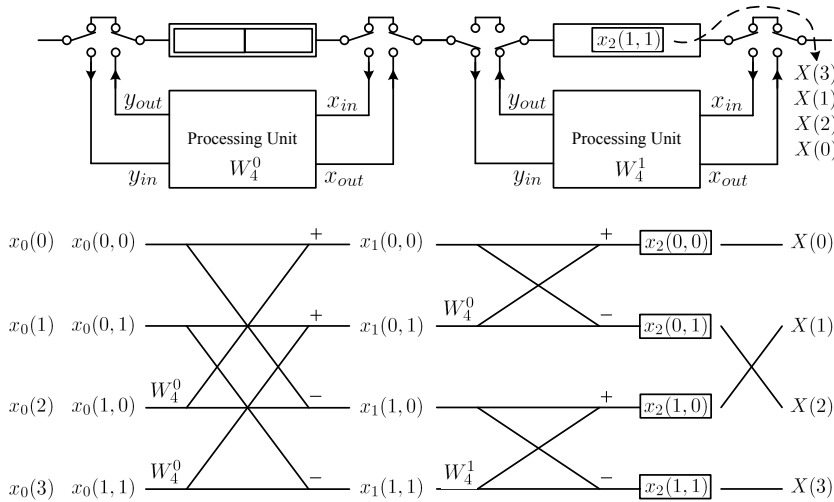


Figure 2.19: Step 10

Once the first stage has processed all the data, the second had already started its processing work, using the same procedure. The difference is the depth of the delay line (length=1) and the coefficient  $W_N^z$  applied by the processing unit.

This section has presented the principle of a pipelined DFT based on the Cooley-Tukey algorithm. It enables an easy hardware implementation of a FFT. The next part exhibits the architecture of a Sampled Analog Signal Processor using this implementation.

## 2.3 Architecture

A Sampled Analog Signal Processor (SASP) is to be designed. An architecture was thus proposed to implement the DFT algorithm in order to process the frequency translation principle. A full SR system is given. The study is focused on the DFT part and technical perspectives are given on the RF-surrounding building blocks. Figure 2.20 presents the SASP architecture. 3 parts compose the SASP:

1. An analog signal pre-processing
2. An analog discrete-time signal processing
3. A digital signal processing

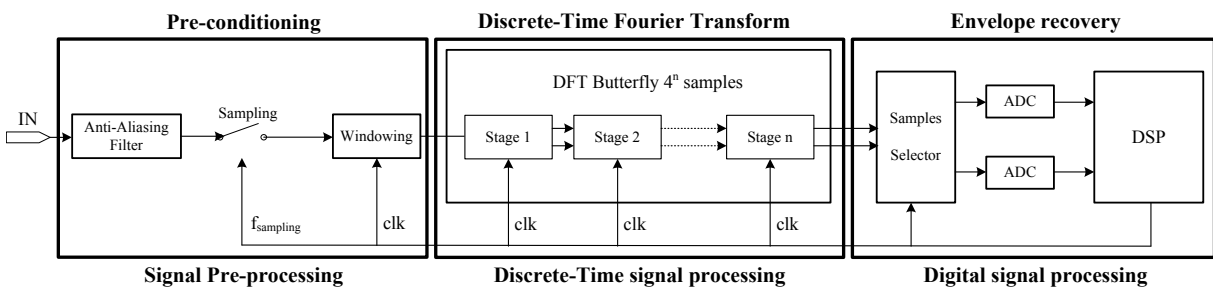


Figure 2.20: SASP Architecture

### 2.3.1 Signal pre-processing

The signal is pre-conditioned before its discrete-time processing. This includes filtering, sampling and windowing operations.

### 2.3.1.1 The Anti-Aliasing Filter

In the RF reception chain, the signal is amplified by the LNA. It must be a wideband amplification in order to cover all the spectrum from 0 to 5GHz. LNA architectures researches are currently done to display such characteristics [15, 16]. Once amplified, the signal has to be filtered to avoid any aliasing effect (Fig. 2.21). The anti-aliasing filter (AAF) has to be a low pass filter from 0 to  $\frac{f_{\text{sampling}}}{2}$ . This frequency range is imposed by the Shannon's theorem and selects the widest band. This step is crucial because the AAF enables two kinds of interference to be suppressed (Fig. 2.21):

- Frequencies from  $\frac{f_{\text{sampling}}}{2}$  to  $f_{\text{sampling}}$  could not be baseband mirrored (Fig. 2.22).
- Frequencies from  $f_{\text{sampling}}$  and higher are also suppressed and do not disturb the desired signal (highest order mirror effect).

As this is not the scope of this thesis, the feasibility of the AAF is not here discussed.

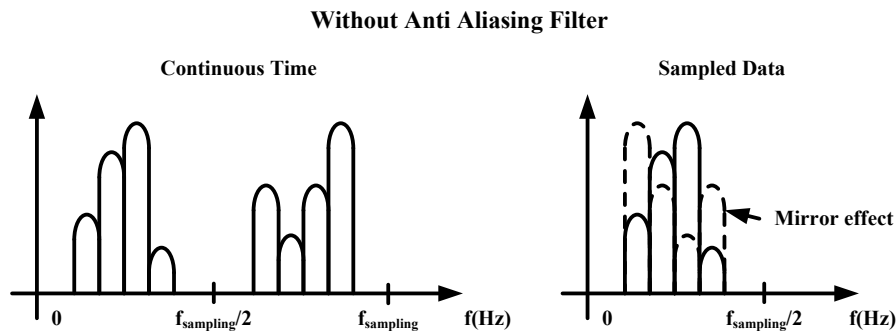


Figure 2.21: Aliasing Matters without Anti Aliasing Filter

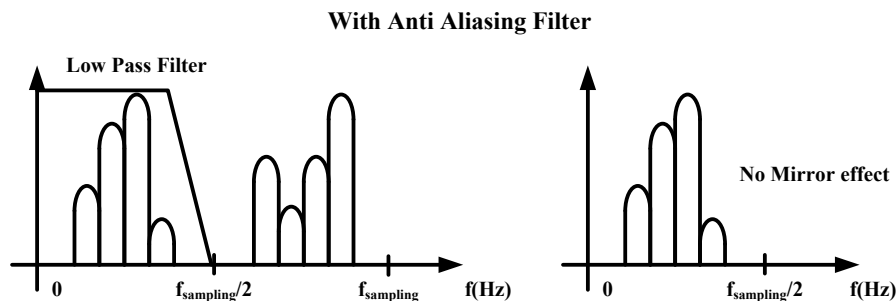


Figure 2.22: Aliasing Matters with Anti Aliasing Filter

### 2.3.1.2 Sampling

Once filtered, the RF signal is sampled at at least twice the RF signal frequency to respect the Shannon's theorem. A Track and Hold (T/H) sampler pre-discretizes the signal and displays the voltage samples to the FFT circuit. Sampling is the most important part of the system because the resolution of the calculation depends on its accuracy. The sampling frequency  $f_{\text{sampling}}$  determines the FFT timing ( $N \cdot f_{\text{sampling}}$ ), the spectrum range (from 0 Hz to  $\frac{f_{\text{sampling}}}{2}$ ) and the spectrum resolution ( $\frac{f_{\text{sampling}}}{N}$ ).

The major technological issue is the aperture error. It is composed of 3 types of errors: aperture jitter, aperture uncertainty and aperture delay [40]. They play on the robustness of a discrete-time system and are to be known to avoid sampling errors. Further investigations can lighten those issues.

#### Aperture Jitter

Sampling is governed by a sampling clock. The rising clock edge decides of the sampling instant. It is the source of the major aperture error: aperture jitter. It is due to random variations on sampling instants, caused by the circuit noise (thermal, power-supply, clock). A variation of time  $\Delta T$  implies an error of  $\Delta V$  on the voltage sample (Fig. 2.23). Sampling instants are to be spaced uniformly to perform the best Fourier Transform but as the error is a gaussian error due to the noise, it is hardly feasible to reduce the aperture jitter.

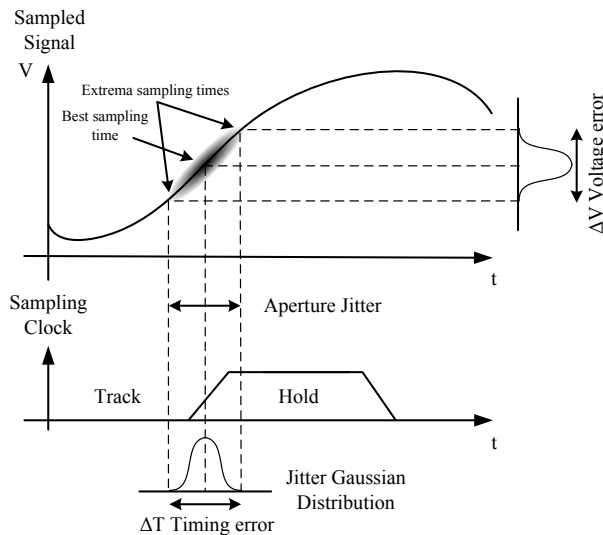


Figure 2.23: Aperture Time Error

### Aperture Uncertainty

Aperture uncertainty is due to a variation of the commutation threshold (Fig. 2.24). It can be reduced by increasing the slope of the sampling clock. The ideal case is an infinite slope but not realistic. On the designer point of view, it is possible to add circuitry to increase the slope, but at the cost of a power consumption enhancement. A trade-off has to be done between aperture uncertainty and power consumption.

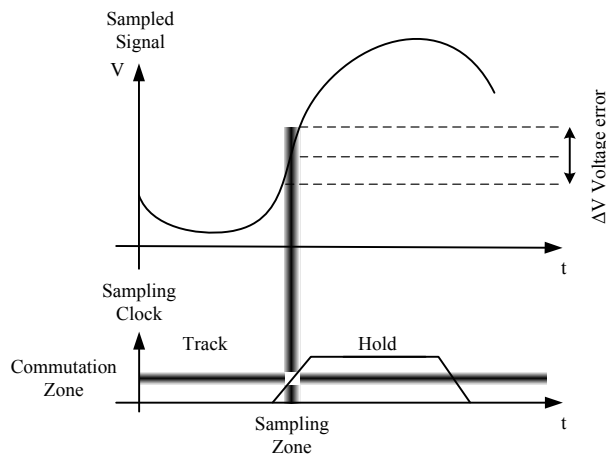


Figure 2.24: Aperture Uncertainty Error

### Aperture Delay

The aperture delay is characterised by the delay from the instant when the system receives the sampling command and the actual time when it occurs. The delay is a propagation delay in the circuitry and does not affect the sampler of the SASP as this error is transparent. The sampling time is not important but the sampling period must be constant.

### Conclusion

Sampling is the most critical phase in a discrete-time system. This section has exposed the technological issues and opens questions to be answered for a full robust SR system at RF frequencies [41]. As the topic of the thesis is not focused on that part, a classical sampler architecture will be considered for a matter of simplicity.

### 2.3.1.3 Windowing matters

Once sampled, the signal has to be windowed. In fact, the range of the data processing has to be limited to the number of stored samples  $N$ . The boundaries of the processed signal are abrupt. Turning data abruptly on and off has an undesired effect on the spectrum. This effect is reduced thanks to a weighting function called window. It weights the data in order to turn them 'on' and 'off' slowly and gently. In our case, the window period considered is equal to  $T_p = N.T_{\text{sampling}}$  and is synchronized on the FFT processing period. Different kinds of windows can be applied on the sampled signal [42]. A choice has to be made to maximize the FFT accuracy as the window is hard-implemented in the circuit and cannot be modified. Time and spectral response of common window functions are exhibited in figure 2.25. Rectangular, Hamming and Hanning are proposed. A choice is done depending on the targeted application.

[42] explains that a window with a very narrow main lobe will have a high spectral resolvability and a lower uncertainty in measuring the frequency of a spectral component. In most cases, a narrow main lobe implies high side lobes causing low detectability of weak spectral components. In addition, a narrow main lobe will cause a commensurate uncertainty in the measurement of the spectral component amplitudes as a result of high scallop loss (Fig. 2.26).

[42] cites the example of the Hanning window which is the window of choice in applications requiring high resolvability. As a result of its narrow main lobe, the frequency resolution is maximized and the frequency measurement uncertainty is minimized. However, because of the higher side lobes, the detectability of low-level nearby spectral terms is reduced in comparison with other windows [43]. The uncertainty of the amplitude measurement is greater than for windows designed for low scallop loss, such as the Harris Flat Top window (Fig. 2.26). As a result of the wide flat spectral response, the Flat Top has small scallop loss and hence exhibits small amplitude errors, but the same flatness leads to significant frequency uncertainty, which must be resolved by one of the spectral interpolation options.

A trade-off between dynamic range and frequency range is to be done. The Hamming window is the best compromise in term of bandwidth and loss as it provides a moderate frequency resolution and a moderate scallop loss (Fig. 2.26). Its equation is given in equation 2.17. It will be hard-implemented to window received RF signal in the SASP.

$$W(t) = 0.54 + 0.46 \cos\left(2\pi \frac{t}{T_p}\right) \quad (2.17)$$



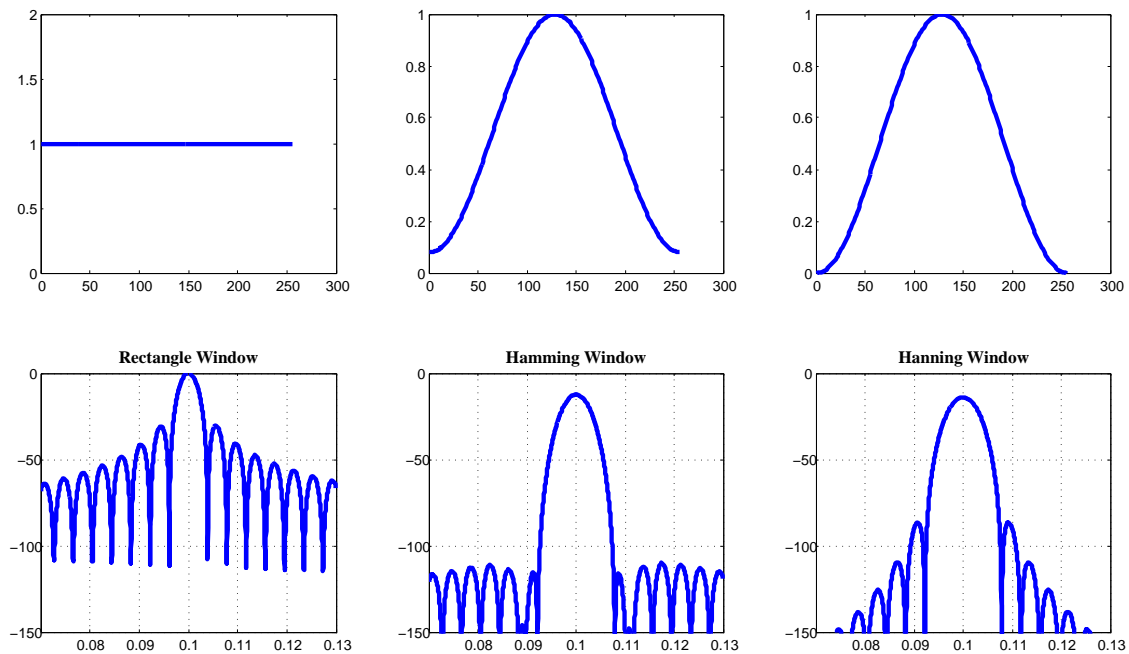


Figure 2.25: Different kind of windows

Table 2.1: Windows and Figure of Merit

Window	Highest side lobe (dB)	Scallop Loss (dB)
Rectangular	-13	3.92
Hamming	-43	1.78
Hanning	-23	2.10
Kaiser-Bessel	-46	1.46
Blackman	-58	1.10

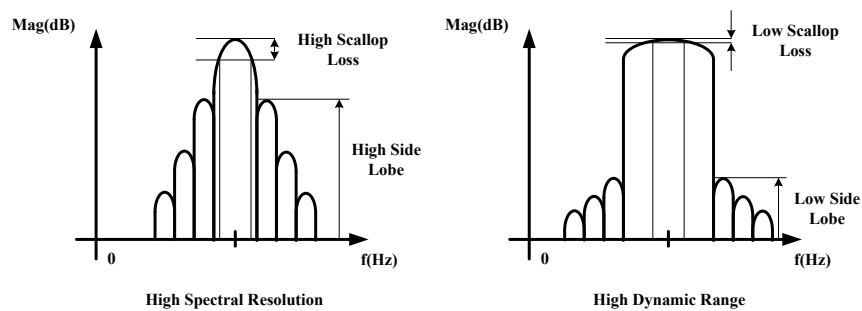


Figure 2.26: Window characteristics

### 2.3.2 DFT implementation

Previous references exhibit a radix-2 pipeline FFT implying  $\log_2(N)$  stages in the Butterfly scheme [37]. In order to improve the speed efficiency, a radix-4 FFT using  $\log_4(N)$  stages was chosen (Fig. 2.27) [44, 45].

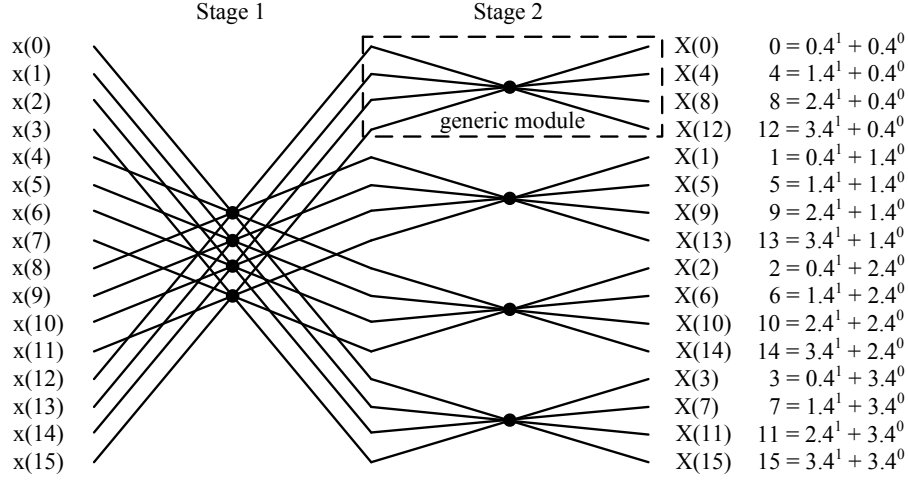


Figure 2.27: Diagram flow of a radix-4 FFT with  $N = 16$ , i.e. 2 stages

All stages of a radix-4 FFT use a basic module with 4 weighted inputs by twiddle factors  $W_N^k$  and 4 outputs. This module (Fig. 2.28) can be decomposed and expressed in its matrix form using the simplifications and factorizations described in [46, 47]:

$$\begin{bmatrix} X(k) \\ X(k + \frac{N}{4}) \\ X(k + \frac{N}{2}) \\ X(k + \frac{3N}{4}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \cdot \begin{bmatrix} W_N^0 F_0(k) \\ W_N^k F_1(k) \\ W_N^{2k} F_2(k) \\ W_N^{3k} F_3(k) \end{bmatrix} \quad (2.18)$$

where  $F_{0,1,2,3}(k)$  is the input-sample vector from the previous stage defined as:

$$F_{n_1}(k) = \sum_{n_2=0}^{N/4-1} x(n_1 + 4n_2) W_{N/4}^{n_2 k} \quad (2.19)$$

for  $n_1 = 0, 1, 2, 3$  and  $k = 0, 1, 2, \dots, \frac{N}{4} - 1$

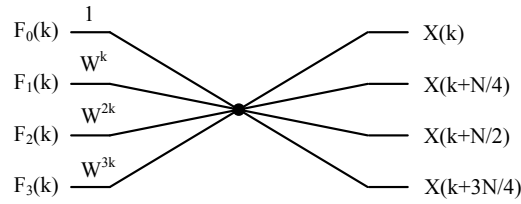


Figure 2.28: Basic radix-4 FFT module

A pipeline implementation consists in using one basic module per stage which runs with two processing phases [44, 45]:

1. Summation/substraction and weighting factor [48].
2. Feedback storage.

This is implemented by a stage composed by (Fig. 2.29):

1. An Input Delay Line. The length of the delay line of a given stage  $r_{\text{stage}}$  is equal to  $3 \cdot 4^{\log_4(N) - r_{\text{stage}}}$  samples.
2. A Processing Unit (Weighting Unit and Matrix Unit).
3. A Feedback Delay Line.

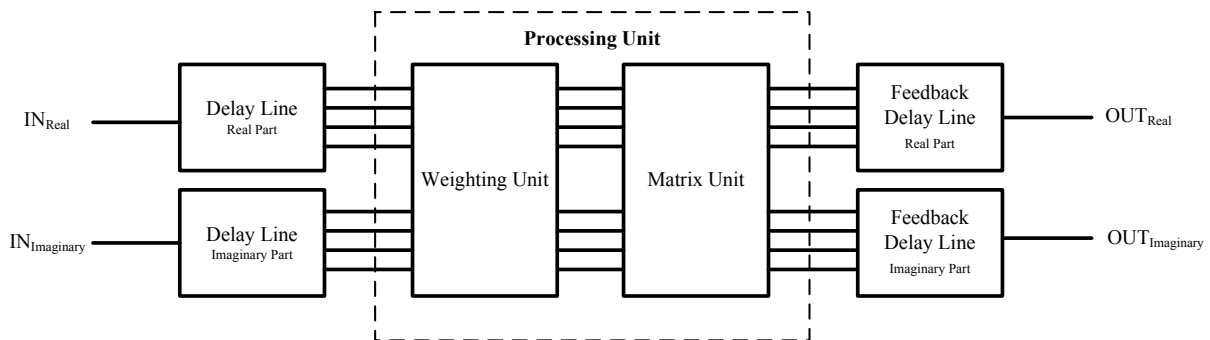


Figure 2.29: Stage architecture

**Delay Line:** The delay line processes a delay equivalent to  $z^{-1}$  as a  $z$ -transformation. It was carried out by an accumulation delay line that stores the voltage samples during a given time in a capacitor (Fig. 2.30). It enables a non-destructive readout of the voltage samples so that they

could be processed several times in further operations when required. Three operating states of the delay line are performed. For example, at a given time, Table 2.2 describes the switches configurations corresponding to each operating states.

Table 2.2: Operating states and switches configurations

Operating state	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$
Sample loaded in $C_1$	closed	open	open	open	open	open
Sample stored in $C_2$	open	open	open	open	open	open
Sample output from $C_3$	open	open	open	open	closed	open

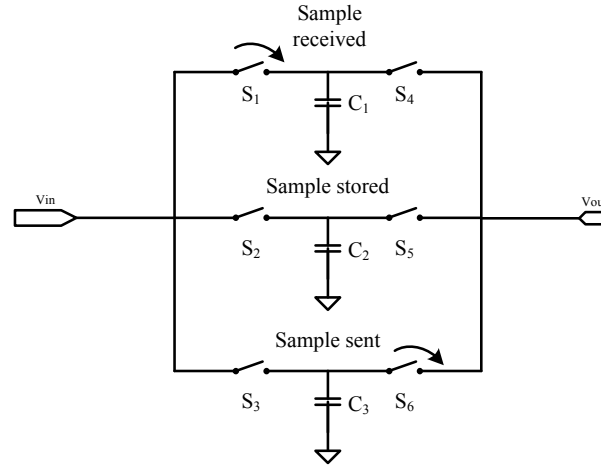


Figure 2.30: Accumulation Delay Line

The delay is created by the fact that the sample is stored during the expected delay time in a capacitor. This principle improves the accuracy of charge transfers by opposition to a regular delay line. The whole structure counts  $4^{\log_4 N} + 4^{\log_4 N - 1} - 2$  capacitors.

**Processing Unit:** The processing unit weights the delay voltage samples by a coefficient from 0 to 1 according to  $W_N^k = \exp\left(\frac{-j2k\pi}{N}\right)$  (Eq. 2.3). The coefficients are applied as described in Figure 2.28.  $k$  is increased by a power of 4 from one stage to the next one and displayed in base-4-reverse order. Once weighted, the processing unit adds and subtracts the voltage samples as described in the summation/substraction matrix (Eq. 2.18).

All these discrete analog operations give the calculation of the FFT. Their realization and synchronization are the goal of this thesis.

### 2.3.3 Post-signal processing

#### Samples selector

Voltage samples are output and display the spectrum. It is not necessary to convert into digital all the voltage samples. Only the voltage samples representing the desired RF envelope are converted. Figure 2.31 depicts the case of the selection of one voltage sample among  $N$  in the output spectrum. The case of several samples is explained latter. The A/D conversion is done at  $\frac{f_{\text{sampling}}}{N}$ . Consequently, the technological bottleneck coming from the ADC working frequency is relaxed.

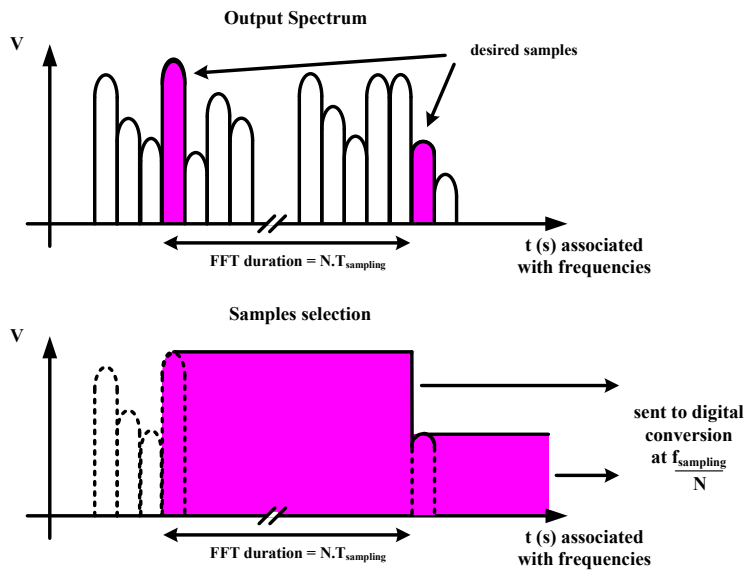


Figure 2.31: Samples selection

The FFT algorithm implemented in the architecture (Fig. 2.20) displayed the output spectrum sample in a base-4-reversed order. This order implies that two neighbor frequency samples in the spectrum are several samples apart. If the frequency band to be considered after the FFT calculation is " $n_{\text{envelope}}$  samples" wide, then the space between each sample to be sent to the ADC is  $4^{N-\log_4(n_{\text{envelope}})}$  samples (Fig. 2.32). The aim of the samples selector was to capture the required output voltage samples knowing their expected output timing. Once the output order is exposed, it is easy to keep the right voltage sample and convert it into digital form at the right instant. The A/D conversion is done at  $n_{\text{envelope}} \cdot \frac{f_{\text{sampling}}}{N}$ . This operation of selection/conversion still led to work at a dramatically reduced frequency. For instance, in the case of a 4096-point FFT, the space between 2 neighbor frequency samples is 1024 samples. If an envelope counts 4 voltage samples, the A/D conversion is done at  $4 \cdot \frac{f_{\text{sampling}}}{4096} = \frac{f_{\text{sampling}}}{1024}$ .

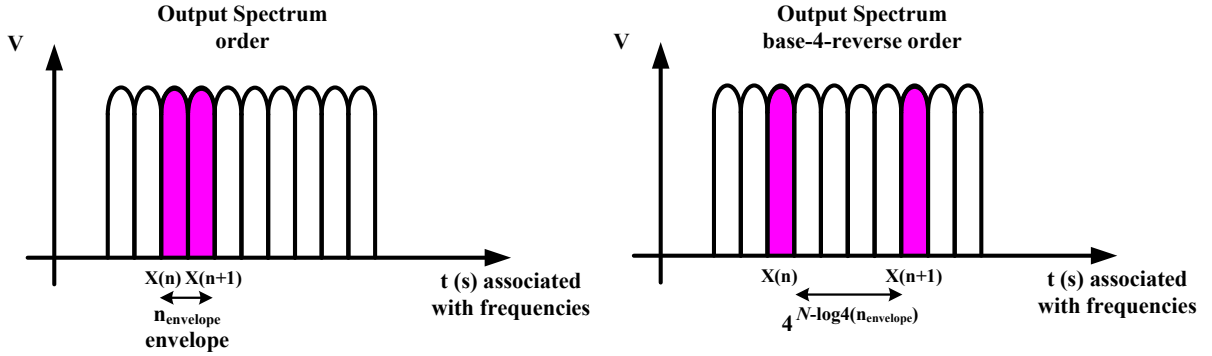


Figure 2.32: Envelope voltage samples selection

### A/D conversion and digital signal processing

The decimation in frequency reduces the output data rate. The ADC bandwidth is equal to  $n_{\text{envelope}} \cdot \frac{f_{\text{sampling}}}{N}$ . Table 2.3 shows how the data rate is reduced for three different configurations of SASP. In this example, the sampling frequency is 4GHz in order to address all the cellular standard from 0 to 2GHz, including GSM, DCS and PCS. For instance, the GSM bit rate is equal to 271kbits in a channel bandwidth of 200kHz. Considering a 65536-element SASP, 4 samples could recover the RF signal easily. The output frequency is 244kHz which is very easy for an ADC to convert and for a DSP to handle.

Table 2.4 presents the 65536-sample SASP configuration to address common standards. An entire number of samples is chosen to represent the channel bandwidth. This number is chosen as follow:

- $f_{\text{sampling}}$  is at least twice higher than  $f_{\text{carrier}}$ .
- $\frac{f_{\text{sampling}}}{f_{\text{bandwidth}}}$  must be an integer.  $f_{\text{sampling}}$  is given by  $f_{\text{sampling}} = f_{\text{bandwidth}} \cdot \frac{N}{n_{\text{envelope}}}$ . As  $f_{\text{bandwidth}}$  and  $N$  are given, we increase  $n_{\text{envelope}}$  till  $f_{\text{sampling}} \geq 2 \cdot f_{\text{carrier}}$ . The first integer satisfying this condition is  $n_{\text{envelope}}$ .
- $f_{\text{sampling}}$  can be calculated with  $f_{\text{bandwidth}}$ ,  $N$  and  $n_{\text{envelope}}$ .

For instance, concerning the GSM standard,  $\Delta f_{\text{bandwidth}} = n_{\text{envelope}} \cdot \frac{f_{\text{sampling}}}{N}$  i.e.  $200kHz = 6 \cdot \frac{2.184GHz}{65536}$ .

The choice of configuration emphasizes SASP calculation accuracy. The higher  $N$  is, the more accurate the FFT is. But, the higher  $N$  is, the higher is the power consumption and the die area are. That is why  $n_{\text{envelope}}$  has to be as low as possible regardless the criteria before exposed.

Table 2.3: Comparison of number of samples at a given sampling frequency of 4GHz

N-sample SASP sampling frequency at 4GHz			
Number of samples $N$	4096	16384	65536
Frequency resolution	977kHz	244kHz	61kHz
Output Data Rate of a $n_{\text{envelope}}$ -sample wide envelope after decimation			
$k=4$ samples	3.9MHz	977kHz	244kHz
$k=16$ samples	15.6MHz	3.9MHz	977kHz

Table 2.4: RF standards addressed by 65536-point SASP

System	$f_{\text{carrier}}$	$f_{\text{bandwidth}}$	Modulation	$n_{\text{envelope}}$	$f_{\text{sampling}}$
GSM	925-960MHz	200kHz	GMSK	6	2.184GHz
DCS	1805-1880MHz	200kHz	GMSK	3	4.368GHz
UMTS	2110-2170MHz	5MHz	QPSK,HPSK	65	5.041GHz
Bluetooth	2402-2480MHz	1MHz	GFSK	12	5.461GHz
802.11g	2412-2472MHz	20MHz	OFDM	250	5.243GHz

## 2.4 A Software Radio System

A VHDL-AMS model of the SASP was designed. It aims at validating the algorithm and the applications of the SASP. Architecture presented in figure 2.20 was simulated and approved. Applications presented in this section are simulated with this model.

### 2.4.1 Concurrent reception

The envelope selection is not limited to the selection of only one RF signal envelope. The output samples representing several signal envelopes could be buffered to be converted at a lower rate. This is the concept of concurrent reception. Figure 2.33 depicts the capture of samples representing two signal envelopes among  $N$  samples output by the SASP. It is just a matter of selecting the samples of both envelopes.

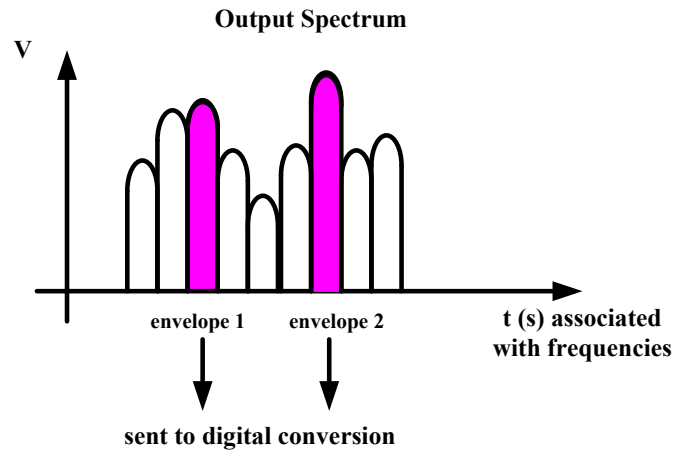


Figure 2.33: Concurrent reception

## 2.4.2 Frequency demodulation

Once the spectrum processed, an IFFT can be performed digitally to recover a baseband transient signal (Fig. 2.5). A temporal demodulation is done. This concept is not optimized in terms of performance and new applications can be explored by the use of the spectrum composed by the voltage samples. In fact, amplitude and phase information are directly carried by the spectrum. Modulations such as PSK, QAM, FSK and by extension, OFDM can be processed by optimized algorithms in the frequency domain. Example of a BPSK modulation is here considered. It is the simplest form of PSK.

Figure 2.34 depicts the example of a BPSK modulation. The input bits are encoded through a phase shifting of  $180^\circ$ . The RF signal amplitude remains the same but as the phase changes, the real and imaginary output spectrum of a '0' is reversed compared to a '1' (Fig. 2.34).

A BPSK demodulation could be optimized with the SASP by a relevant interpretation of the output spectrum. A simulation is done with a carrier frequency of 500MHz for simplicity. The BPSK modulated signal received is first windowed. Its length was sized to be the timing of a modulated bit (here  $2.048\mu s$ ). The sampling frequency is  $f_{\text{sampling}} = 2GHz$ . The spectral accuracy is thus  $f_{\text{bandwidth}} = 488kHz$ . Figure 2.35 depicts the signal processing. It can be noticed that bit signature can be well recognized in the output spectrum of the SASP: the main voltage sample during a FFT processing is reversed depending on a '1' or a '0' is encoded. This main voltage sample is here encircled. The processing is also proven whereas the calculation was not synchronised on the bit rate (Fig. 2.36). The same modulated signal but delayed of half



a bit timing is depicted in figure 2.36. The principle remained an adapted interpretation of the output spectrum. A comparator is just sufficient to recover every encoded bits. The output data rate is 488kHz. In this case, the working frequency is thus divided by more than 1000 and part of the demodulation was processed by the SASP.

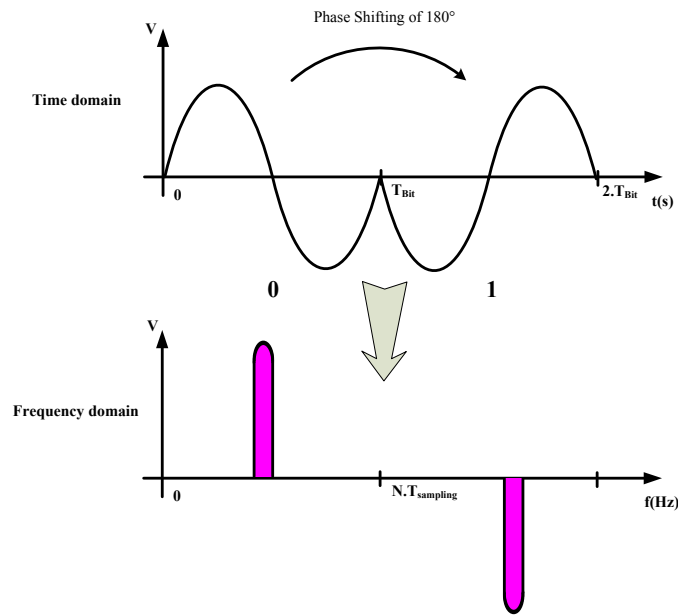


Figure 2.34: Theoretical BPSK signal processing

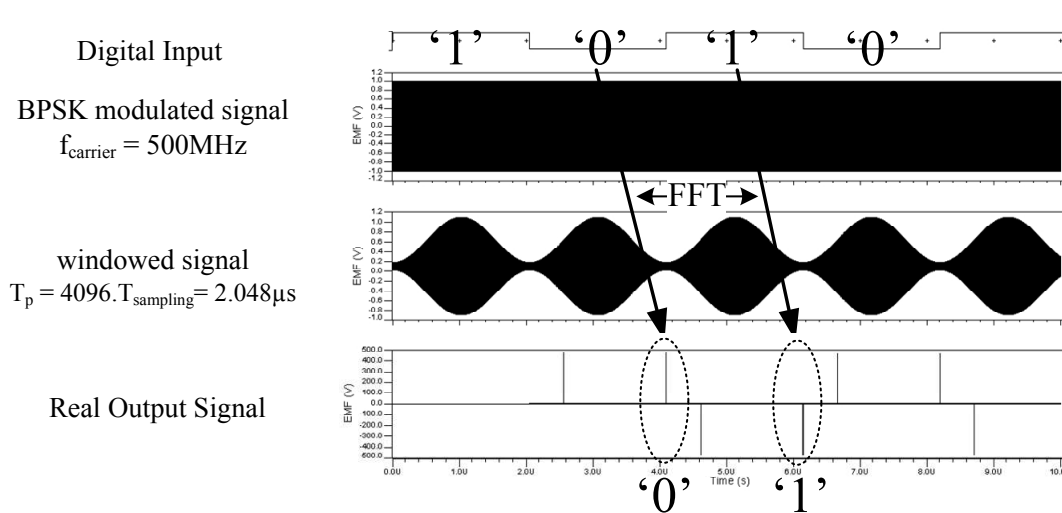


Figure 2.35: A synchronized BPSK signal processed by 4096-point SASP

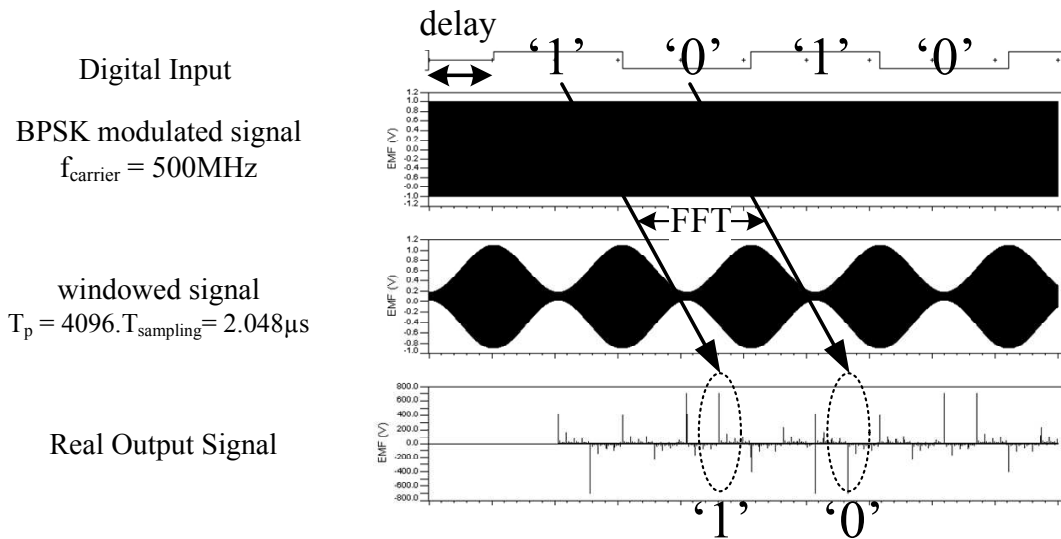


Figure 2.36: A non-synchronized BPSK signal processed by 4096-point SASP

This concept can be enlarged to OFDM modulation (Fig. 2.37). In fact, as the SASP performs a FFT, all the sub-carriers are directly demodulated and can be processed separately into digital. The A/D conversion and the digital processing speed are performed at a lower rate which implies a reduction of power consumption. Other applications can be envisaged such as Frequency Hopping modulation types, easily demodulated by the mean of the spectrum.

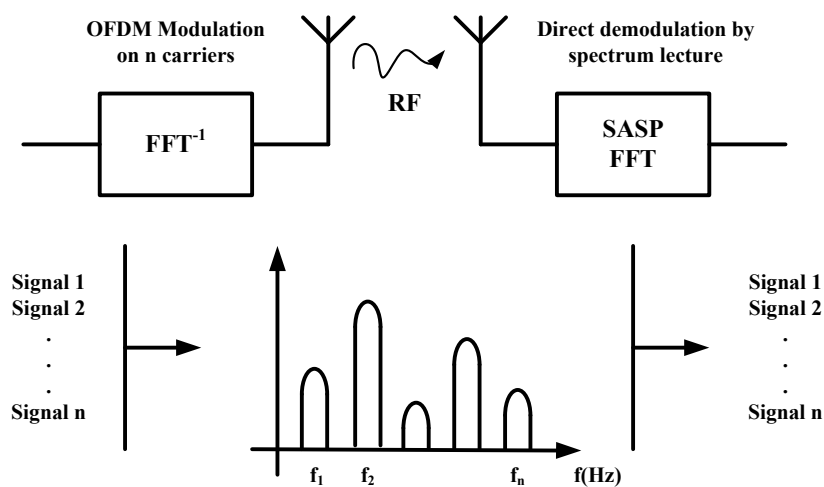


Figure 2.37: A OFDM signal processed by the SASP

## 2.5 Conclusion

The principle of the SASP has been exposed. It performs a frequency translation by the mean of discrete analog voltage samples. Its strength lies in the implemented algorithm. A FFT is processed using the Cooley-Tukey algorithm. Its low complexity is easy to be implemented in analog. FFT aims at selecting the desired RF envelope to be demodulated. Then, it is displayed in baseband to ADC to be treated in digital by a DSP. Applications of the SASP were proposed. Several RF envelopes can be selected and demodulated at the same time. This is called the concurrent reception. Part of the demodulation can be done directly in frequency domain by the mean of FFT. This is called frequency demodulation.

In order to validate the feasibility of the system and its applications, a SASP prototype is designed. The characteristics retained are:

- The FFT is processed with 64 samples.
- The technology used is 65nm CMOS from STMicroelectronics.
- Maximal sampling frequency is 500MHz.
- Maximal input dynamic range is 200mV.

Chapter 3 exposes the design flow, the schematic simulations, the layout design and the Post Layout Simulations.

---

# Schematics and Modeling results

---

## Contents

---

<b>3.1</b>	<b>Discrete Analog Operations</b> . . . . .	<b>84</b>
3.1.1	Accumulation Delay Line . . . . .	88
3.1.2	Matrix Unit . . . . .	95
3.1.3	Weighting Unit . . . . .	98
<b>3.2</b>	<b>Digital Instructions</b> . . . . .	<b>104</b>
3.2.1	A base-4 algorithm clock generation . . . . .	104
3.2.2	A hardware-implemented algorithm . . . . .	108
<b>3.3</b>	<b>Design - <i>SASPEPA</i> and <i>LUCATESTA</i></b> . . . . .	<b>112</b>
3.3.1	Peripheral building blocks . . . . .	112
3.3.2	Layout considerations . . . . .	115
3.3.3	A building block library . . . . .	115
3.3.4	Post-Layout Simulations . . . . .	118
<b>3.4</b>	<b>Conclusion</b> . . . . .	<b>125</b>

---

Chapter 3 describes the design of the SASP. First, each building block is presented. Analog and digital parts are exposed, their schematics explained and their simulation results depicted. The principle of analog signal processing is explored in details. Specifications are extracted from schematic simulations, such as power consumption, minimum and maximum working frequencies. Finally, full simulations of the SASP are proposed. It validates the concurrent reception and the frequency demodulation principles. A layout is laid out and Post-Layout Simulations (PLS) are performed.

**Key words:** discrete analog operations, schematics, design, simulations, specifications.

### 3.1 Discrete Analog Operations

Discrete analog operations are synchronised to process the voltage samples. A state machine is developed to display the right operation at the right time. It is composed by a digital part synchronised on the sampling frequency and an analog part to process the FFT. This architecture is similar to a processor architecture as it is the crossing of instructions and datas (Fig. 3.1).

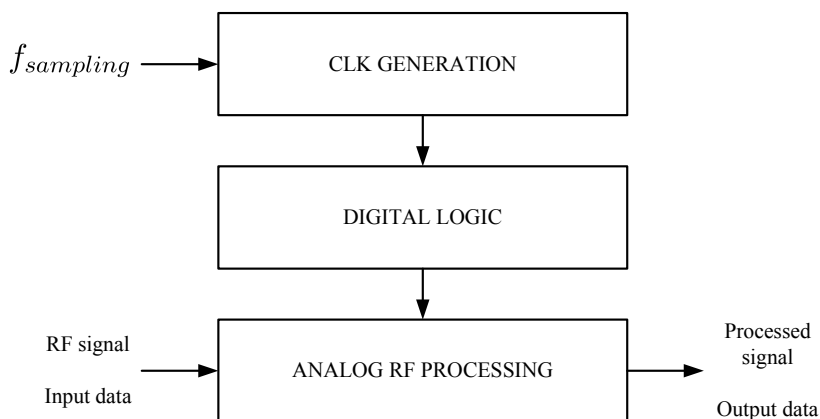


Figure 3.1: Processor Architecture

The design tasks are consequently divided into 2 parts: the design of the analog part and the design of the digital one. The analog part is composed by the pipelined stages. The designed SASP processes **64 samples**. It is called **SASP64**. The clock frequency is given by  $f_{\text{sampling}} = \frac{1}{T_{\text{sampling}}}$ . The whole signal processing takes  $64 \cdot T_{\text{sampling}}$ .

The SASP64 requires 3 stages to process an analog FFT. Its architecture is depicted in figure 3.2. Figure 3.3 summarizes the FFT algorithm to process 64 samples. The first stage processes 64 samples, the second one 16 samples and the last 4 samples. Each stage input is weighted by a coefficient equal to  $W_N^{nk}$ , where only  $k$  is given by figure 3.3. It exhibits also the base-4 reverse output order which is explained thanks to its decomposition into base 4.

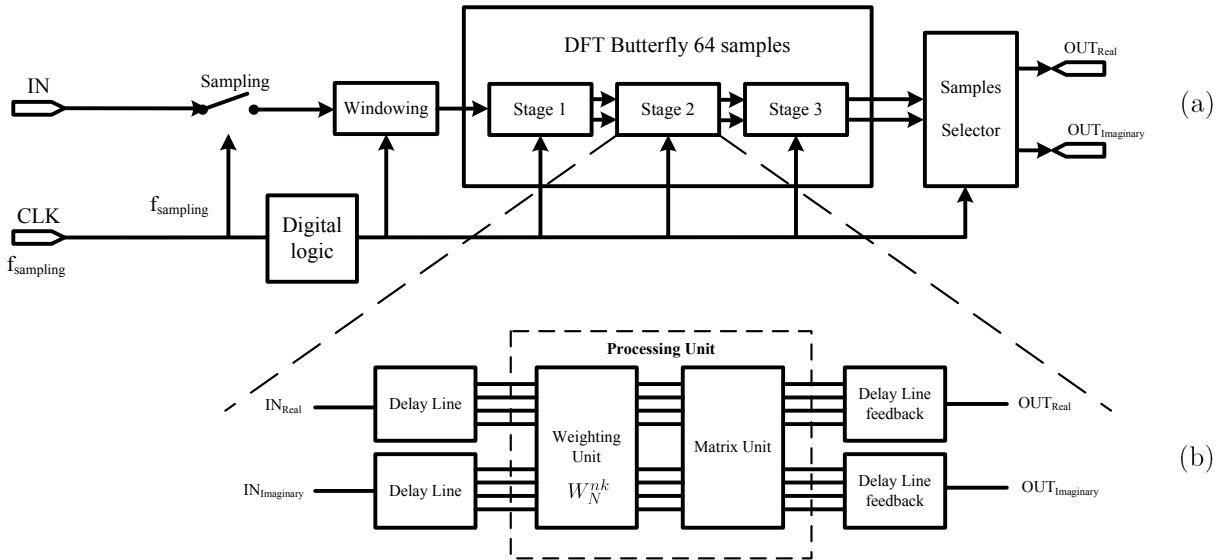


Figure 3.2: (a) Architecture of SASP64, (b) Close-up of a stage

The following sections present the design of one stage. Each part of the stage is explained. Their design respects the design flow depicted in figure 3.4. First, analog operations are extracted from the algorithm. Then, it is implemented into VHDL-AMS language to validate the system behavior and check the synchronization of all the system. Once done, each part of the circuit is designed with 65nm CMOS technology Design Kit from STMicroelectronics. Schematics are simulated to fit as much as possible the VHDL-AMS simulations considering all the parasitic elements that can occur and disturbate the design of the processor. Finally, a layout is proposed and post layout simulations are performed before sending the circuit to foundry.

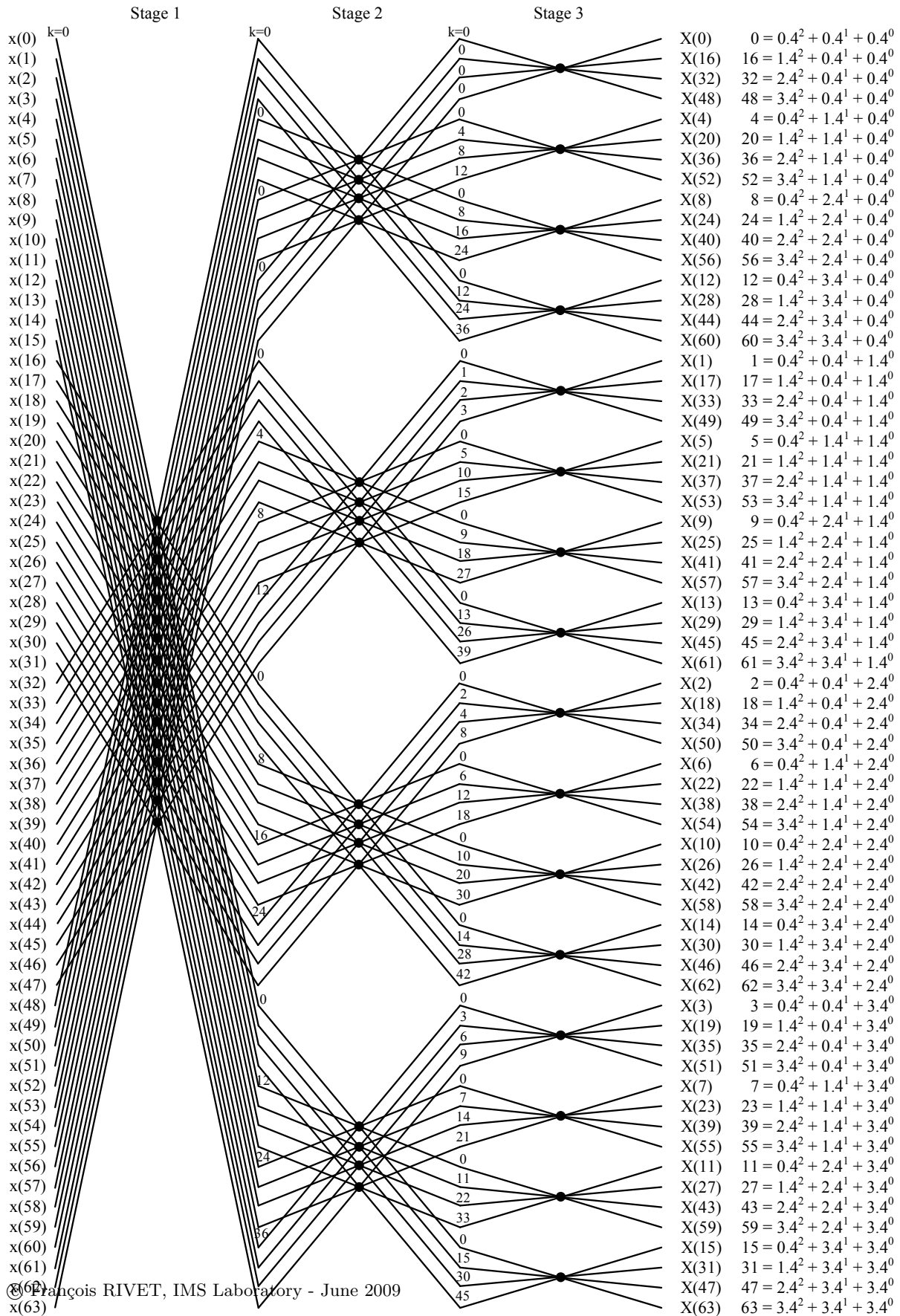


Figure 3.3: Diagram flow of a radix-4 FFT with  $N = 64$

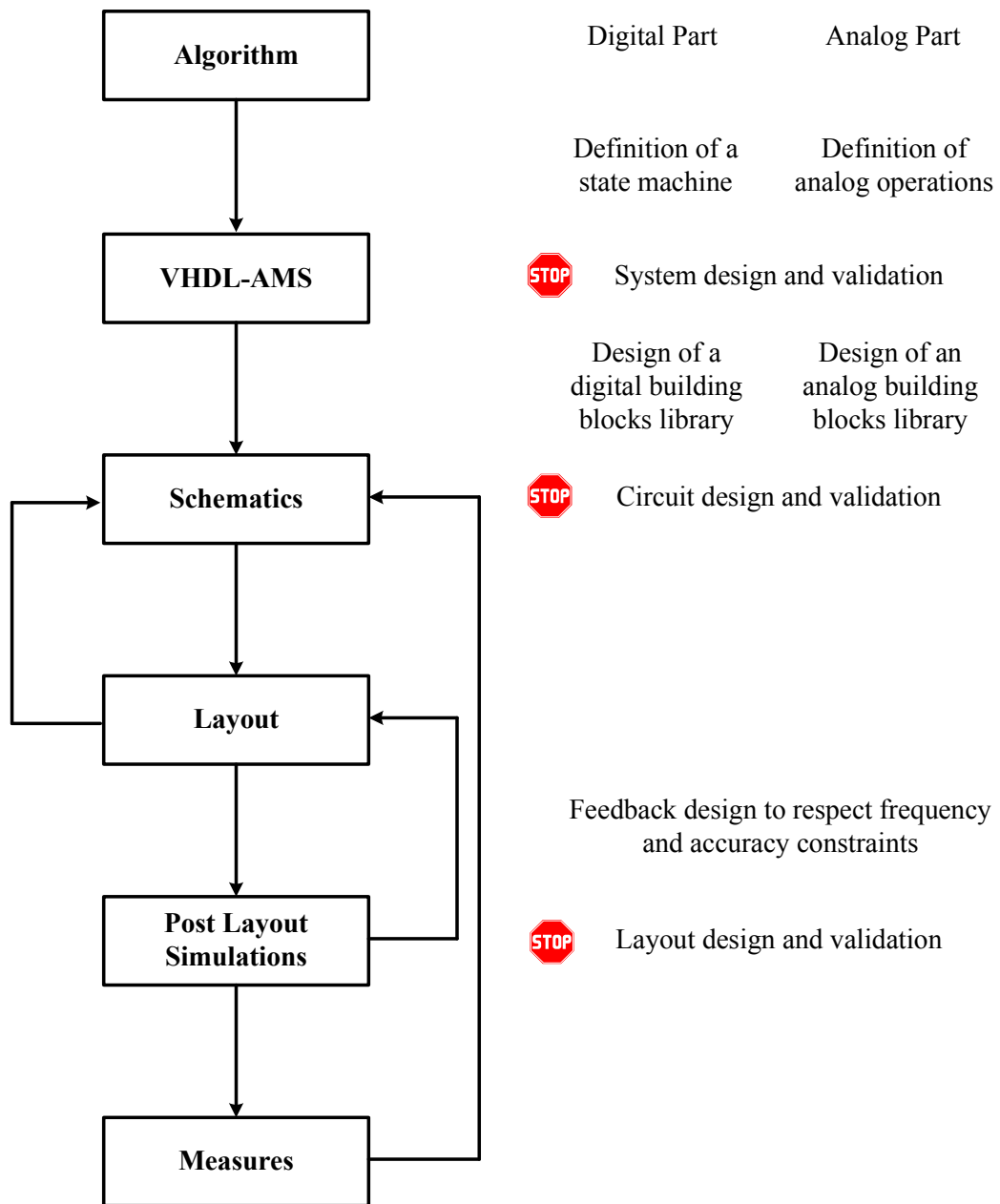


Figure 3.4: Design Flow



### 3.1.1 Accumulation Delay Line

The algorithm is divided into 2 processing phases:

- A storage phase.
- A processing phase.

Figure 3.5 exhibits the 2 processing phases of the first stage of the SASP64. 48 samples are stored to be delayed during  $\frac{3}{4}.64.T_{\text{sampling}}$  which corresponds to phases 1, 2 and 3. Voltage samples  $x(0)$ ,  $x(16)$ ,  $x(32)$  are sent toward the processing unit at the same time to be processed with  $x(48)$  during phase 4. It gives 4 new voltage samples that are sent out toward the next stage (Fig. 3.5). This is the first node shown in figure 3.3.

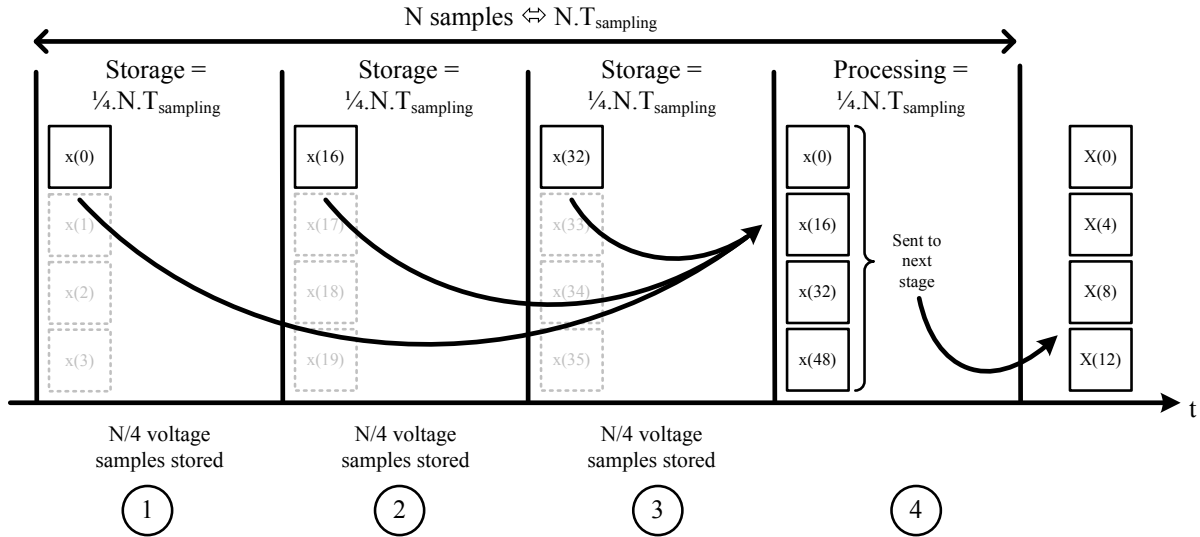


Figure 3.5: Processing phases

A delay cell aims at storing a voltage sample. Consequently, a delay line is composed by  $3.4^{\log_4(N)-r_{\text{stage}}}$  delay cells i.e.  $r_{\text{stage}} \in [1; 3]$  (see section 2.3.2). A delay line stores  $3.4^{\log_4(N)-k-1}$  voltage samples to be delayed during the storage phase. To fulfil the FFT algorithm process, the delay line receives  $4^{\log_4(N)-k}$  serial voltage samples and displays them into 4 parallel outputs (Fig. 3.5). This is done thanks to a repartition of the delay cells into  $4^{\log_4(N)-k-1}$  columns and 4 rows (Fig. 3.6). It can be seen as an array. First  $3.4^{\log_4(N)-k-1}$  voltage samples are delayed during phases 1, 2 and 3 to be output at the same time of the last  $4^{\log_4(N)-k-1}$  voltage samples during phase 4.

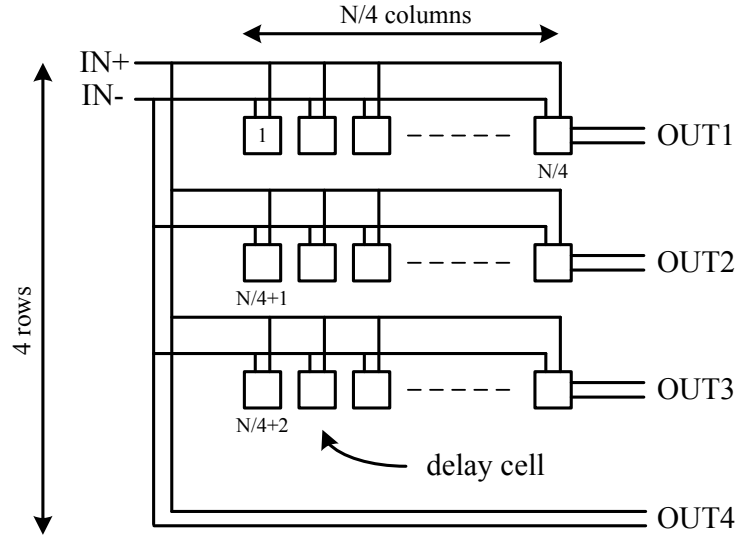


Figure 3.6: Delay Line system view

### 3.1.1.1 Schematics

An accumulation delay line is preferred contrary to a regular one that delays elements thanks to a propagation time [49]. Accumulation enables a non-destructive readout of the voltage samples and limits charge transfer errors due to the circuit non-idealities. A delay line is composed by parallel delay cells (Fig. 3.6). A delay cell is composed by:

- an input switch to load the voltage sample,
- a capacitor to store the voltage sample,
- an output switch to display the voltage sample.

Because the circuit runs mainly with charge transfers, DC offset errors known as pedestral errors are the main drawbacks induced (Eq. 3.1). Thus, a pseudo-differential structure is chosen with both positive and negative signals centered on a DC voltage of 800mV with a linearity range of 200mV around the DC voltage. Only the transistors non linearities and the parasitic capacitors matters remain. **These specifications rule the all specifications of the circuit.**

$$V_{out} = V_{in} \underbrace{\left(1 + \frac{W.L.C_{ox}}{C_{hold}}\right)}_{\text{non unity gain}} - \underbrace{\frac{W.L.C_{ox}}{C_{hold}} (V_{DD} - V_{TH})}_{\text{pedestal error}} \quad (3.1)$$

Each delay cell has a positive and a negative part to store the samples coming from the positive and the negative signals. A capacitor  $C_{hold}$  stores the voltage sample. An inside buffer protects the voltage sample against the output voltage variations and improves the charge transfer. It is implemented with a single transistor  $M_{1,2}$  and a resistor  $R_{1,2}$  (Fig. 3.7).  $M_{1,2}$  are scaled as  $W/L = 20$ . It has a gain slightly superior to 1 to guarantee a gain of 2 along all the charge transfers to compensate the attenuation provided by the FFT calculation on each stage.

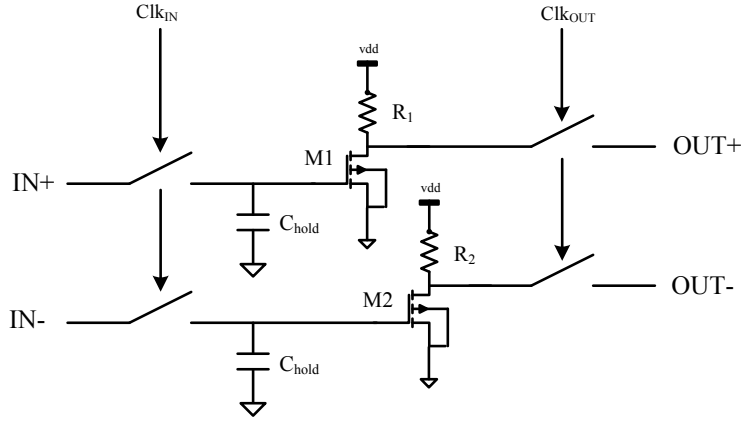


Figure 3.7: Simplified schematic of a delay cell

A trade-off is done on the capacitor value  $C_{hold}$ . It has to enable a fast load of the voltage sample without losing charge due to leakage during the delay time. It defines a low and a high frequency limit of the working range of the delay line. The highest limit is given by the fastest load time of such a circuit. The lowest is given by the permitted leakage to guarantee an acceptable error on the charge transfer error. As the basic structure of a delay cell is based on capacitors, the die area occupied by the capacitor in the circuit is also a factor to take into account. The whole SASP counts  $4^{\log_4 N} + 4^{\log_4 N - 1} - 2$  capacitors [45]. Thus, the trade-off implies to choose a capacitor value as low as possible. The storage capacitor  $C_{hold}$  value is 50fF (Fig. 3.7).

### 3.1.1.2 Simulation results

The first element to be simulated is the buffer made by  $M_{1,2}$  and  $R_{1,2}$ . Figure 3.8 gives its characterization in the chosen technology. In figure 3.8(a), the input signal is a ramp from 0V to  $V_{dd} = 1.2V$ . An acceptable voltage region of linearity is determined. The derivative of the output signal of the buffer is shown in figure 3.8(b). A 1% error is allowed as an acceptable non-linearity. A voltage linearity region of 100mV centered on 800mV is simulated. Thus, the

linearity region of the differential structure is 200mV, which is the chosen dynamic range. The gain of the buffer is 1.15.

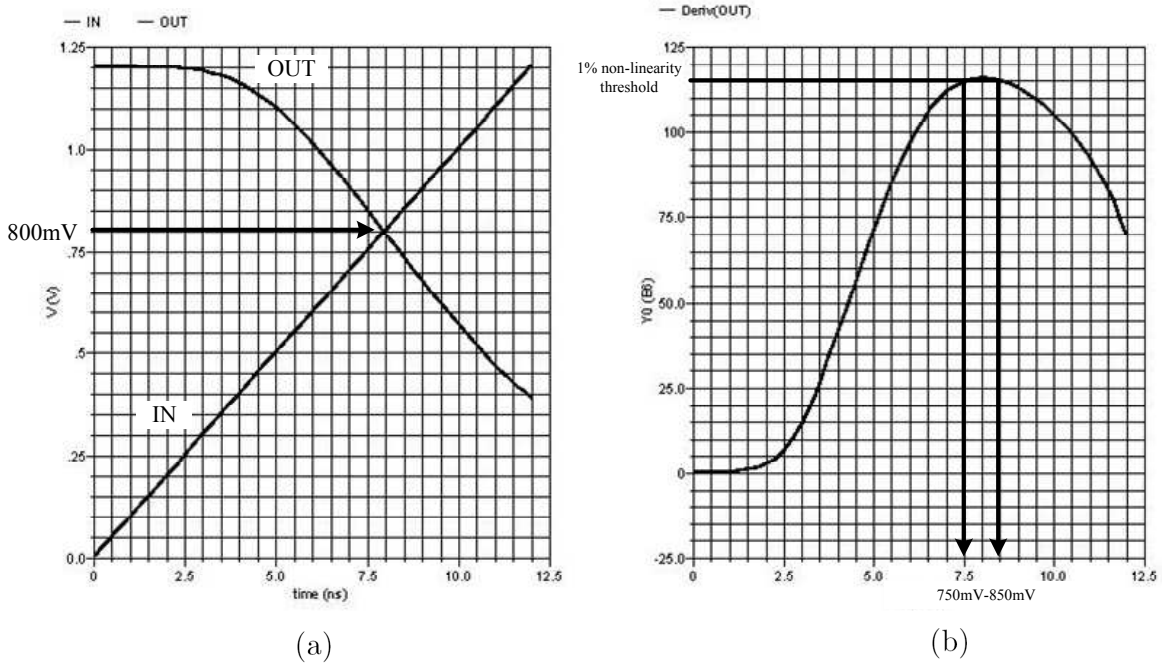


Figure 3.8: (a) Buffer characterization, (b) Output derivation

The delay line lowest working frequency is given by the maximal voltage loss allowed. This is determined by figure 3.9. An input signal of 100mV is stored. Then, the voltage sample loss is simulated. If it exceeds more than 1%, the FFT calculation is no more guaranteed. Simulations with  $f_{\text{sampling}} = 10\text{MHz}$  and  $f_{\text{sampling}} = 100\text{MHz}$  are presented. It can be seen that at 10MHz,  $f_{\text{sampling}}$  is too low to keep the voltage sample stored in  $C_{\text{hold}}$  whereas at 100MHz, the threshold of 1% is almost respected. Table 3.1 presents error rates. It is easy to conclude that the highest  $f_{\text{sampling}}$  is, the lowest the error rate is.

Figure 3.10 presents the simulation of a charge transfer with  $f_{\text{sampling}} = 640\text{MHz}$ . The delay line has a length of 4 voltage samples. First, a charge is loaded at the first period and stored during 3 periods (Fig. 3.10(a)). Then, it is replaced by a new charge. The charge is buffered continuously to be displayed at the delay cell output (Fig. 3.10(b)). Finally, the charge is output during a period. The amplification of 1.15 carried out by the buffer can be observed. The power consumption of a delay cell is  $676\mu\text{A}$  under a 1.2V voltage supply.

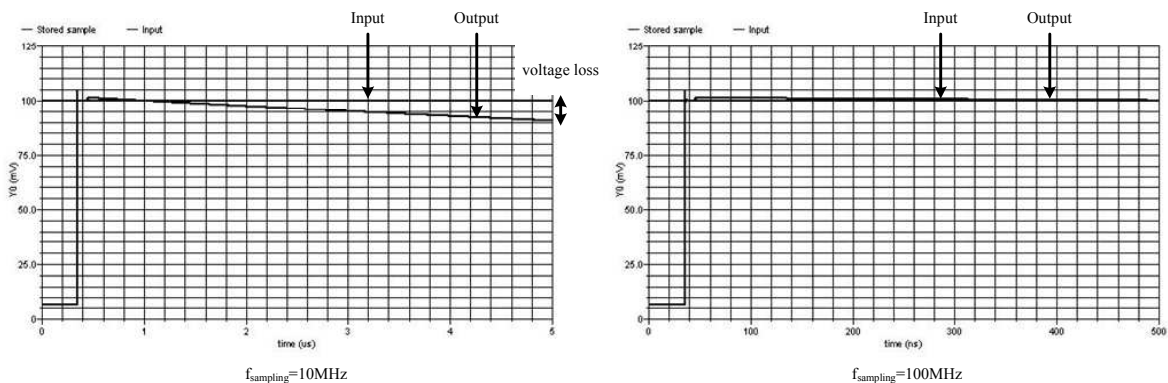


Figure 3.9: Delay line lowest working frequency

Table 3.1: Voltage loss vs  $f_{\text{sampling}}$

$f_{\text{sampling}}$	Total loss during delay time	Error rate
10MHz	11.57mV	11.57%
100MHz	1.30mV	1.3%
640MHz	0.19mV	0.19%

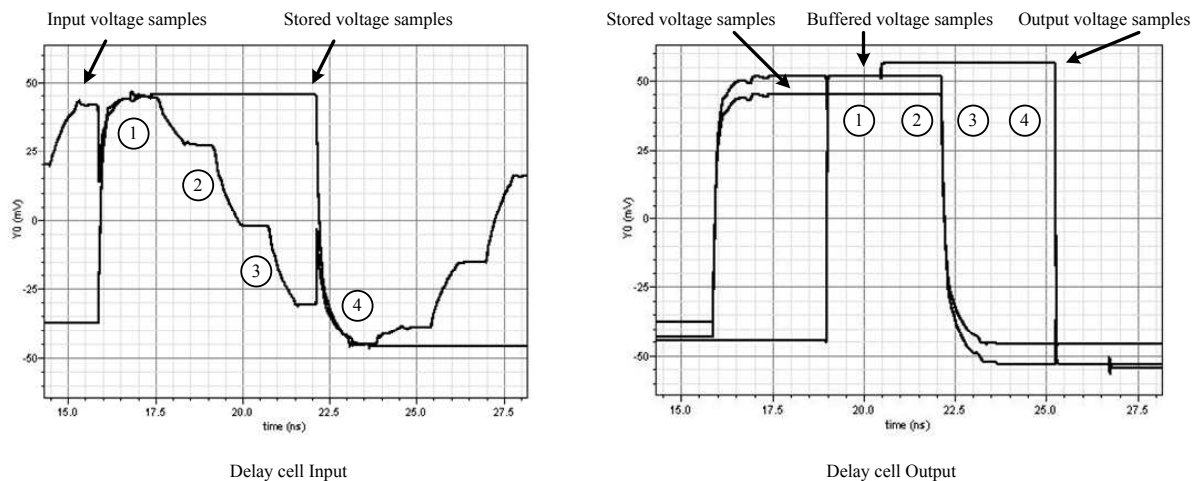


Figure 3.10: Charge Transfer. Load (a) and Display (b) of a voltage sample

Figure 3.11 depicts the 3 kinds of delays provided by a delay line. For instance, it is chosen the first stage delay line. Samples are delayed by 48, 32 and 16 periods. It enables the three first phases presented in figure 3.5.

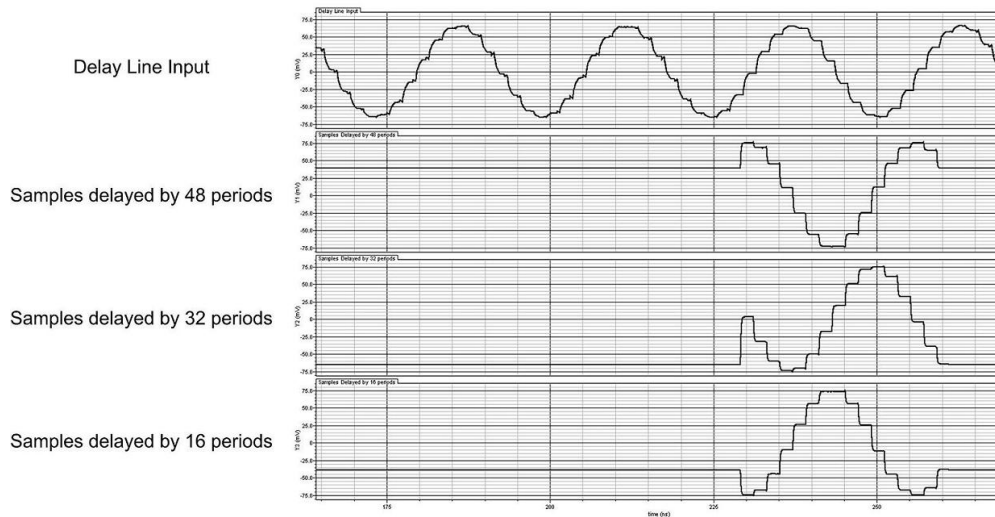


Figure 3.11: Simulation of a delay line

Figure 3.12 proposes an implementation of a delay cell. Input and output RF lines can be seen on left and right parts of the layout. The structure is designed to be duplicated as many times as required. It is organized to be as square as possible. It is just a matter of juxtaposing every cells to build a delay line. The layout die area is  $8\mu\text{m} \times 11\mu\text{m}$ .

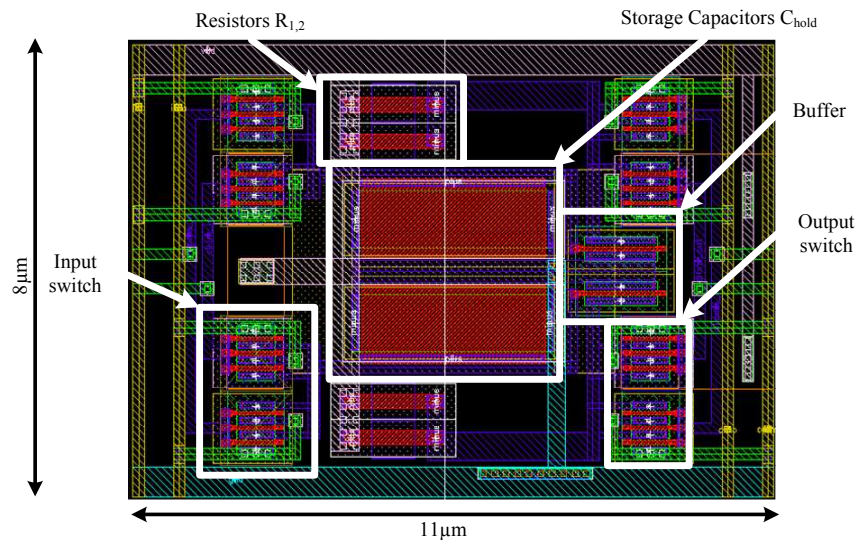


Figure 3.12: Delay Cell layout

Figure 3.13 proposes an implementation of a delay line to handle 64 samples. RF lines are distributed on each side and their length are tried to be as equal as possible to access any delay cell. This is done to maximize the maximal working frequency of the delay line. The layout die area is  $95\mu\text{m} \times 118\mu\text{m}$ . Table 3.2 presents different delay lines power consumption.

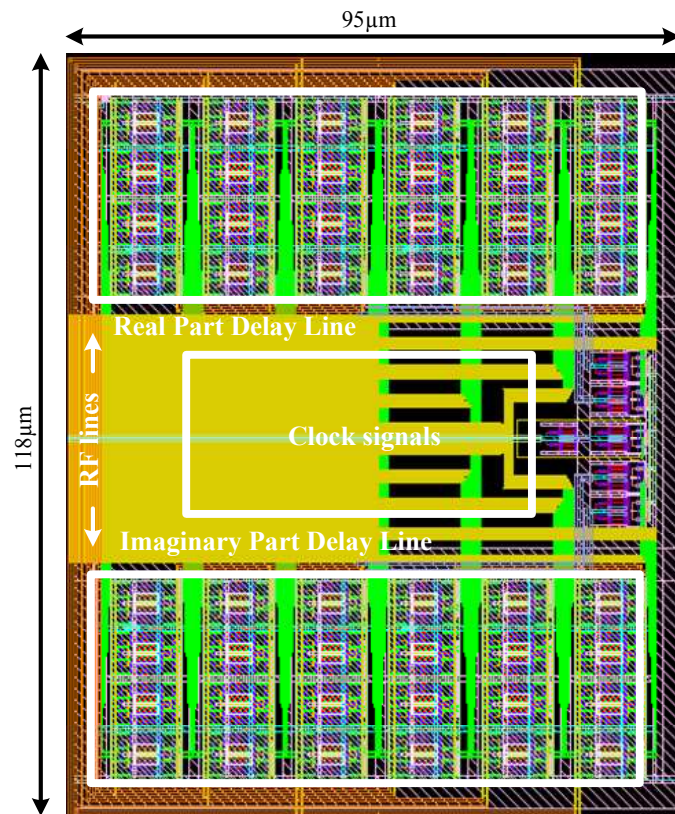


Figure 3.13: Delay Line with 64 samples layout

Table 3.2: Simulated delay lines power consumption

Delay Lines	Power consumption
Stage 1 (48 samples)	33.86mA
Stage 2 (12 samples)	9.6mA
Stage 3 (3 samples)	2.7mA

### 3.1.2 Matrix Unit

The Matrix Unit is part of the Processing Unit. Matrix Unit (MU) adds and subtracts voltage samples 4 by 4 as described in the algorithm (see section 2.3.2). It is composed by basic adders. Figure 3.14 describes how an adder is also a subtracter: by inverting an input, the calculation can be modified. As the matrix performs additions and subtractions on 4 different combinations of voltage samples, it is easy to implement into hardware the matrix described in section 2.3.2. A multiplication by a complex number  $j$  is done by an inversion between real and imaginary part of the system.

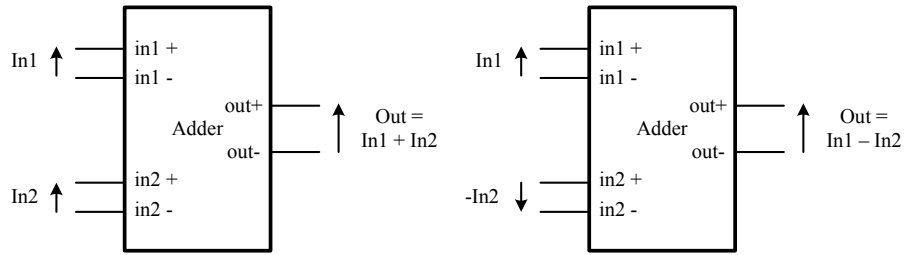


Figure 3.14: Matrix design based on adders implementation

Adders add 4 voltage samples. They have 4 inputs and 4 outputs. The MU has 8 inputs and 8 outputs (4 for real parts, 4 for imaginary parts). Each adder input is connected either  $\Re(IN_x)$  or  $\Im(IN_x)$ ,  $x \in [0; 4]$ . The configuration is given by equation 3.2 and equation 3.3.

$$\begin{bmatrix} OUT_1 \\ OUT_2 \\ OUT_3 \\ OUT_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \cdot \begin{bmatrix} IN_1 \\ IN_2 \\ IN_3 \\ IN_4 \end{bmatrix} \quad (3.2)$$

$$\begin{aligned} \Re(OUT_1) &= \Re(IN_1) + \Re(IN_2) + \Re(IN_3) + \Re(IN_4) \\ \Re(OUT_2) &= \Re(IN_1) - \Im(IN_2) - \Re(IN_3) + \Im(IN_4) \\ \Re(OUT_3) &= \Re(IN_1) - \Re(IN_2) + \Re(IN_3) - \Re(IN_4) \\ \Re(OUT_4) &= \Re(IN_1) + \Im(IN_2) - \Re(IN_3) - \Im(IN_4) \\ \Im(OUT_1) &= \Im(IN_1) + \Im(IN_2) + \Im(IN_3) + \Im(IN_4) \\ \Im(OUT_2) &= \Im(IN_1) - \Re(IN_2) - \Im(IN_3) + \Re(IN_4) \\ \Im(OUT_3) &= \Im(IN_1) - \Im(IN_2) + \Im(IN_3) - \Im(IN_4) \\ \Im(OUT_4) &= \Im(IN_1) + \Re(IN_2) - \Im(IN_3) - \Re(IN_4) \end{aligned} \quad (3.3)$$



### 3.1.2.1 Schematics

Each adder is designed with 4 transistors connected to a common resistor (Fig. 3.15). Each transistor has a size of  $W/L = 20$  and follows the same characteristics as the buffer chosen for the delay cell (dynamic range of 100mV in single mode centered on a DC voltage of 800mV). The current crossing each transistor is proportionnal to the input voltage equal to  $V_{gs}$ . The current crossing the resistor is the sum of the 4 currents coming from each transistors. Thus, the voltage seen at the drain of the transistors is proportionnal to the sum of the input voltages. The structure substracts voltage samples by inverting some of the positive signal input with the negative ones. In figure 3.15, IN3 et IN4 are inverted. Voltage samples on IN3 and IN4 are substracted to those on IN1 and IN2. The MU contains adders et mixed adders/substracters to carry out the FFT calculation as exposed before. The MU is always the same from one stage to an other. Once designed, this part of the circuit has just to be duplicated in each stage.

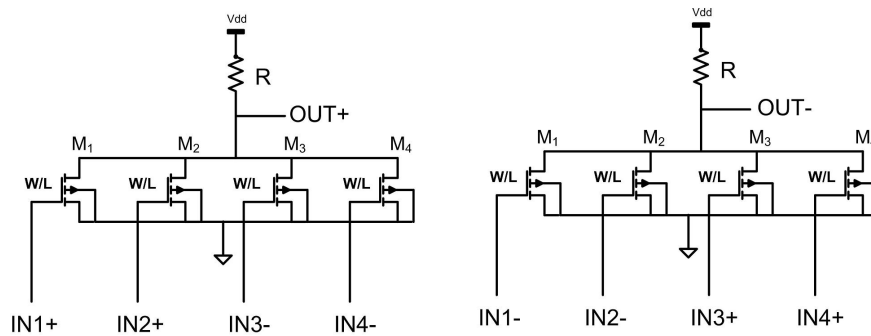


Figure 3.15: Simplified schematic of a 4-voltage sample adder

### 3.1.2.2 Simulation results

A simulation of an adder is done to exhibit its characteristics (Fig. 3.16). Here is given the example of a one signal added to zero signals ( $IN_1 = 50mV$  and  $IN_2 = IN_3 = IN_4 = 0V$ ) (Fig. 3.16(a)). The result is obviously the signal itself but divided by 4. The input signal has an amplitude of 50mV, consequently the output should be 12.5mV but, as the adder has the characteristics similar to the buffer in the delay line, it is amplified. The output amplitude is 15.4mV. The gain is 1.232. A simulation is done with 2 inputs at 50mV ( $IN_1 = IN_2 = 50mV$ ) and others at zero ( $IN_3 = IN_4 = 0V$ ) (Fig. 3.16(b)). The output amplitude is 30.8mV. A simulation is done with the 4 inputs at 50mV ( $IN_1 = IN_2 = IN_3 = IN_4 = 50mV$ ) (Fig. 3.16(c)). The output amplitude is 61.6mV. It is confident with the amplification of 1.232 ( $50mV * 1.232 = 61.6mV$ ). An other simulation is done with a signal added to its inverse. The result is a zero signal (Fig. 3.17). The power consumption of an adder is 2.85mA. The whole power consumption of MU is 22.8mA.

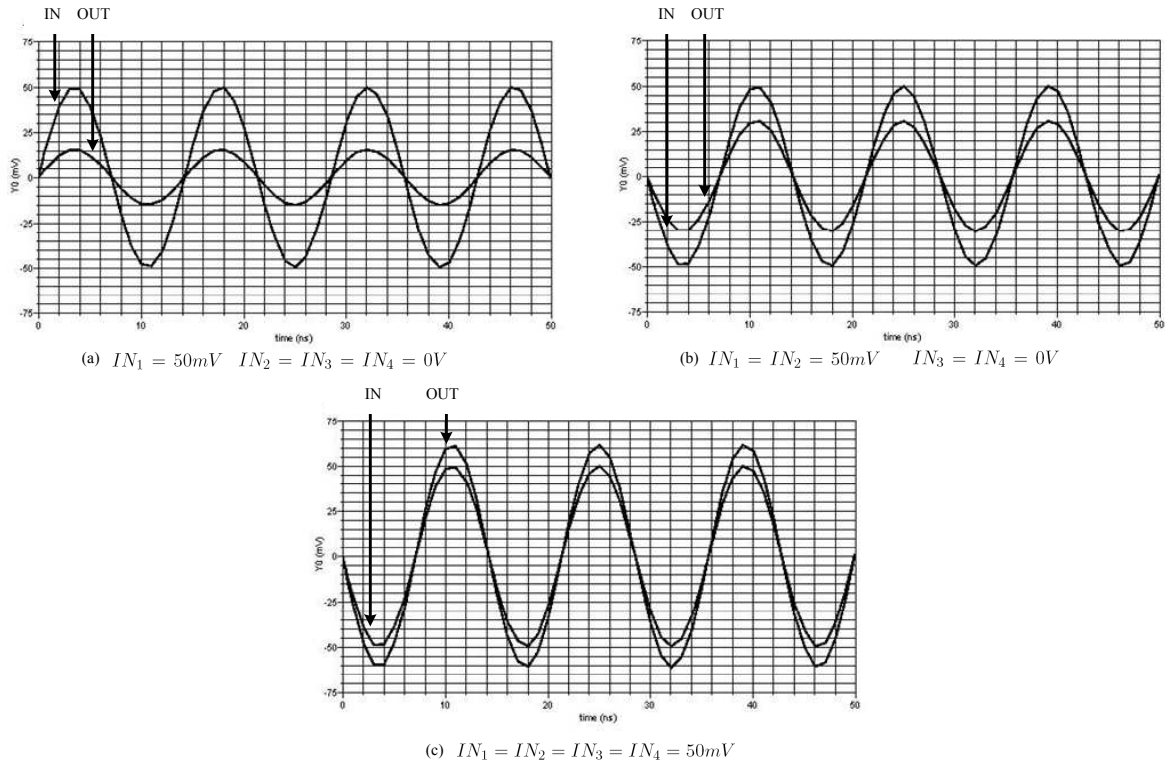


Figure 3.16: Simulation of a 4-voltage sample adder

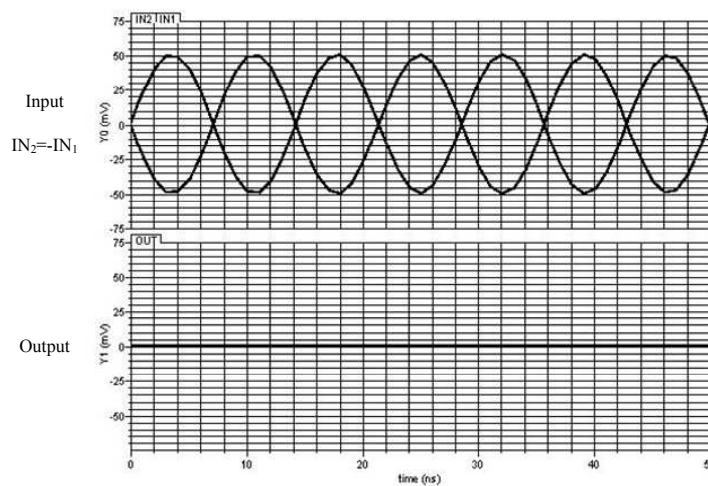


Figure 3.17: Simulation of a 4-voltage sample adder, a signal and its inverse added

Figure 3.18 proposes an implementation of MU. RF lines are at the center of the layout to display at every adder the desired voltage sample. Real part is at the top. Imaginary part at the bottom. The layout die area is  $23\mu m \times 36\mu m$ .

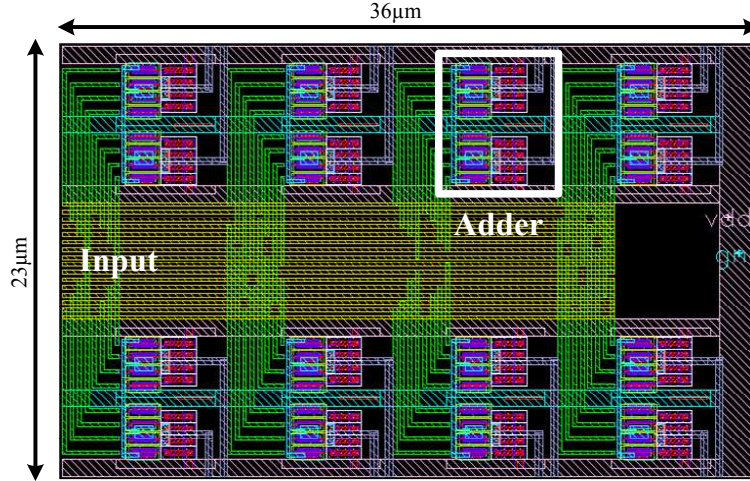


Figure 3.18: Matrix layout

### 3.1.3 Weighting Unit

The Weighting Unit (WU) is part of the Processing Unit. WU weights each sample with a coefficient  $W_N^k = \cos(\frac{2\pi.nk}{N}) + i.\sin(\frac{2\pi.nk}{N})$  (Eq. 2.3). The real part and the imaginary part by of the signal are weighted by  $\cos(\frac{2\pi.nk}{N})$  and  $\sin(\frac{2\pi.nk}{N})$  (Fig. 3.20). Every discrete sample is consequently weighted by a factor within the interval  $[0, 1]$ . As the algorithm is hard-implemented, both real and imaginary part are processed separately. The weighting operation takes into account this specificity by separating the calculation. The real part and the imaginary part is defined by equation 3.4.  $In_{\Re}$  and  $In_{\Im}$  represents real and imaginary part of an input of the WU.  $Out_{\Re}$  and  $Out_{\Im}$  represents real and imaginary part of the output of the WU.

$$\begin{aligned} Out_{\Re} &= In_{\Re}.\cos(\frac{2\pi.nk}{N}) - In_{\Im}.\sin(\frac{2\pi.nk}{N}) \\ Out_{\Im} &= In_{\Re}.\sin(\frac{2\pi.nk}{N}) + In_{\Im}.\cos(\frac{2\pi.nk}{N}) \end{aligned} \quad (3.4)$$

The weighting operation takes place during the processing phase. 4 voltage samples are weighted at the same time. It requires a WU to handle 4 voltage samples. In order to optimize the power consumption, it is chosen to weight the voltage samples before the delay line to divide by 4 the circuit size and maximize the yield of the circuit (Fig. 3.19). Hence, the WU is used 100% of the

time and has a power consumption divided at least by 4. Then, the weighted voltage samples are sent to the delay line. This modification is fully transparent for the algorithm.

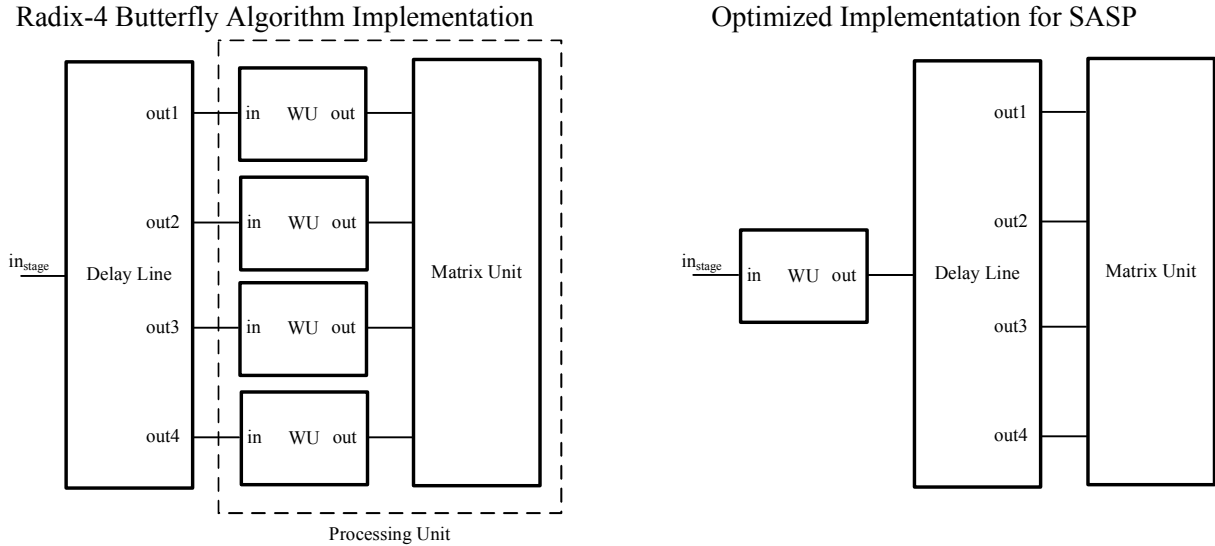


Figure 3.19: Comparison between non-optimized and optimized WU position

The WU is composed into 2 parts. First part is the weighting of Real and Imaginary part and the second one is the addition of the weighted inputs (Fig. 3.20). It enables performing the calculation given by the equation 3.4. WU is synchronized by a logic circuit to display the right weight at the right time. The logic circuit will be discussed later.

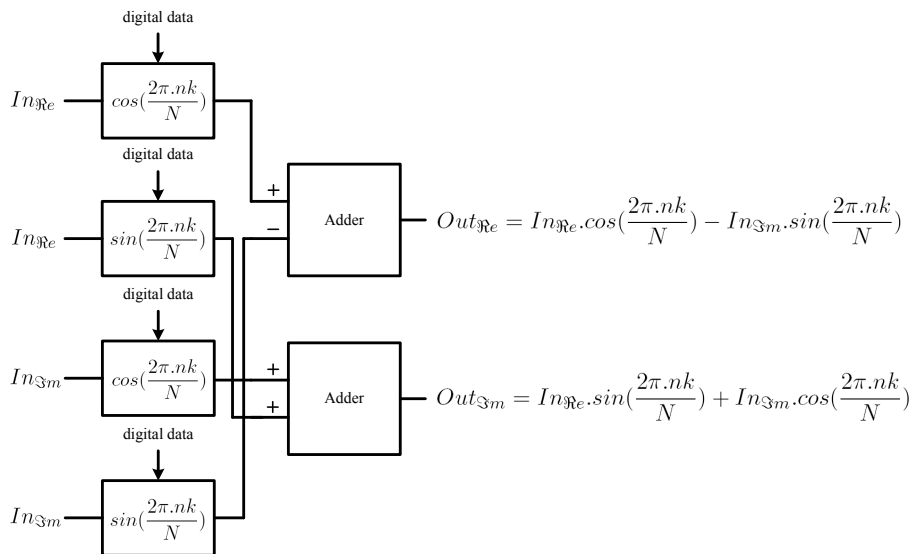


Figure 3.20: Weigthing Unit architecture

Table 3.3 exhibits the coefficients to be applied by each of the 3 stages of the SASP in their base-4 reverse order. This order is the one used to apply the coefficients in serial order before their load by the delay line. It can be noticed that their number is increased by a power of 4 from one stage to another. The number of coefficients used by the first stage is 2 ( $=1+1$ ). The number of coefficients used by the second stage is 5 ( $=4+1$ ). The number of coefficients used by the third stage is 17 ( $=16+1$ ). Negative coefficients are generated by the positive ones by inverting positive and negative inputs. The next part explains how these coefficients are generated.

Table 3.3: WU Coefficients

Used by stage	$k$	$\cos(\frac{2\pi \cdot nk}{N})$				$\sin(\frac{2\pi \cdot nk}{N})$			
		$n = 0$	$n = 1$	$n = 2$	$n = 3$	$n = 0$	$n = 1$	$n = 2$	$n = 3$
1/2/3	base-4 reverse								
1, 2 and 3	0	1	1	1	1	0	0	0	0
2 and 3	4	1	0.924	0.707	0.383	0	0.383	0.707	0.924
2 and 3	8	1	0.707	0.000	-0.707	0	0.707	1.000	0.707
2 and 3	12	1	0.383	-0.707	-0.924	0	0.924	0.707	-0.383
3	1	1	0.995	0.981	0.957	0	0.098	0.195	0.290
3	5	1	0.882	0.556	0.098	0	0.471	0.831	0.995
3	9	1	0.634	-0.195	-0.882	0	0.773	0.981	0.471
3	13	1	0.290	-0.831	-0.773	0	0.957	0.556	-0.634
3	2	1	0.981	0.924	0.831	0	0.195	0.383	0.556
3	6	1	0.831	0.383	-0.195	0	0.556	0.924	0.981
3	10	1	0.556	-0.383	-0.981	0	0.831	0.924	0.195
3	14	1	0.195	-0.924	-0.556	0	0.981	0.383	-0.831
3	3	1	0.957	0.831	0.634	0	0.290	0.556	0.773
3	7	1	0.773	0.195	-0.471	0	0.634	0.981	0.882
3	11	1	0.471	-0.556	-0.995	0	0.882	0.831	-0.098
3	15	1	0.098	-0.981	-0.290	0	0.995	0.195	-0.957

3.1.3.1 Schematics

The principle of the voltage/current/voltage conversion coming from the Matrix Unit is used to carry out this analog operation. A switch network ( $S_x$ ) selects the input voltage of each transistor gate (Fig. 3.21). The input voltage can be either a voltage sample to be weighted or the DC reference voltage. Table 3.4 depicts the possible configuration of the switch network in the case of 4 coefficients. Every transistor ( $M_x$ ) has a different size. It implies that the current crossing each transistor is no more proportionnal to the input voltage but to the width of the transistor.

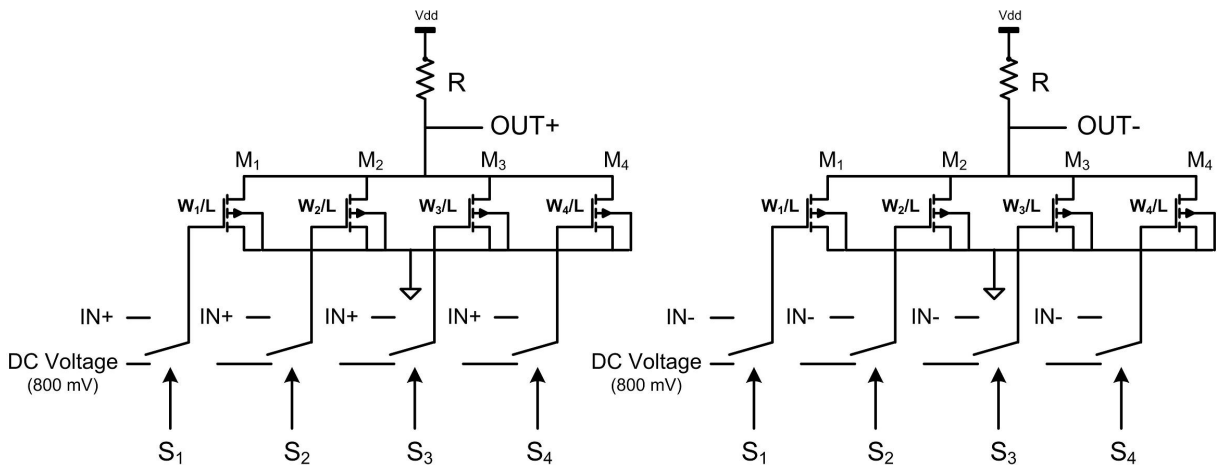


Figure 3.21: Simplified schematic of the Weighting Unit

Table 3.4: Weighting Unit Coefficients Application

Switch Network Configuration				Coefficients
$S_1$	$S_2$	$S_3$	$S_4$	applied
$\frac{W_1}{L} = 2.1$	$\frac{W_2}{L} = 5.3$	$\frac{W_3}{L} = 7.8$	$\frac{W_4}{L} = 9.3$	$\cos(\frac{2\pi \cdot nk}{N})$
DC	DC	DC	DC	0
DC	DC	DC	IN	0.383
DC	DC	IN	IN	0.707
DC	IN	IN	IN	0.924
IN	IN	IN	IN	1

### 3.1.3.2 Simulation results

Figure 3.22 exhibits the weighting of a 100mV voltage sample by 4 coefficients as shown in table 3.5. The sequence is chosen to foresee the delay line behaviour.  $M_x$  states are displayed. If  $M_x$  is at *gnd*, the transistor is switched to the DC reference. If  $M_x$  is at  $V_{dd}$ , the transistor is switched to the input voltage sample. This WU is implemented in the stage 2 (Fig. 3.2). For instance, if  $M_3$  and  $M_4$  are switched to the input and  $M_1$  and  $M_2$  to DC voltage reference, the voltage sample is weighted by 0.707. The output voltage is 84.5mV. It implies an amplification of 1.232 such as in the delay line and in the MU. Power consumptions of WUs are presented in table 3.6.

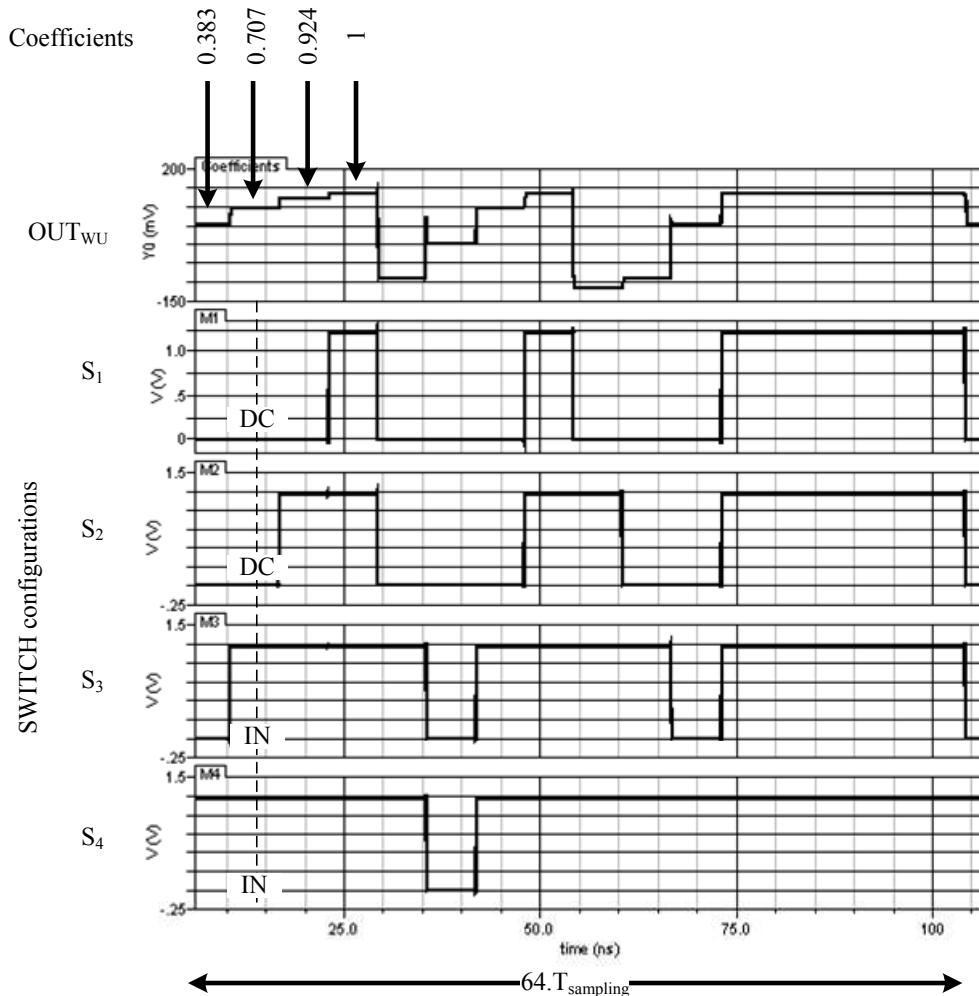


Figure 3.22: A 100mV voltage sample weighted by 4 coefficients

Table 3.5: Weighting Unit Coefficients Simulation

Coefficients			
For a 100mV input voltage sample simulation			
$\cos(\frac{2\pi \cdot nk}{N})$	Output	Effective Coefficient	Error
0	0mV	0	0%
0.383	45.9mV	0.372	2.87%
0.707	84.5mV	0.686	2.97 %
0.924	111mV	0.901	2.48%
1	123.1mV	1	0%

Table 3.6: Simulated WU power consumption

Weighting Unit	Power consumption
Stage 1	N/A
Stage 2 (4 coefficients)	6.44mA
Stage 3 (16 coefficients)	56.5mA

Figure 3.23 proposes an implementation of a WU with 4 coefficients. Transistors banks are duplicated to generate every weighing factor on both inputs. Weighting factor are synchronized by clocks signals displayed at the center of the layout to harmonize delays. Once weighted, an addition is performed following equation 3.4. The layout die area is  $41\mu m \times 48\mu m$ .

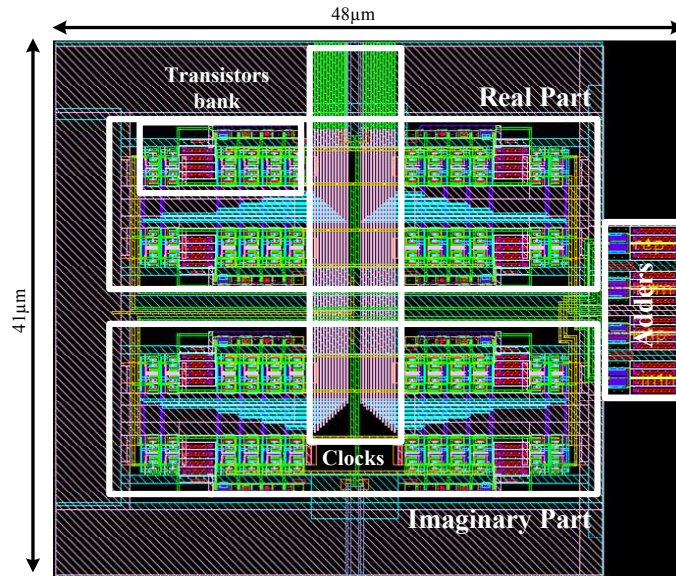


Figure 3.23: Weighting Unit layout



## 3.2 Digital Instructions

A digital circuitry is designed to synchronize the analog circuitry. It is controlled by an input clock signal. This input clock signal is the parameter of reconfigurability of the SASP. Its frequency is  $f_{\text{sampling}}$ . This part describes the architecture of the digital circuitry and its application to the analog part.

### 3.2.1 A base-4 algorithm clock generation

The butterfly algorithm is a base-4 algorithm (see section 2.3.2). A classical logic circuitry is a base-2 circuit, based on a 2-state clock. A base-4 circuit is developed to monitor the algorithm. This circuit is based on 4 pulses representing the 4 states of the base 4. Then, it declines all the states required by the butterfly algorithm. The clock requirements are given for one stage (Fig. 3.2):

- The delay line requires a pulse for each delay cell to upload and download each voltage sample at the right time.
- The WU requires a specific logic to switch the transistor network in the desired configuration to process the chosen coefficient.
- The MU does not require any synchronisation as it is a continuous process element.

The strategy to generate all the desired logic states is to display to the circuit all the pulses of the base 4 and build all the needed logic based on that generated pulses. Figure 3.24 depicts the example of 64 pulses. They are generated thanks to the input clock. Each pulse has a width of  $T_{\text{sampling}}$ . Their duty cycle is  $\frac{1}{64} * 100 = 1.56\%$ . All the logic to be applied to the rest of the circuit is given by a combination of the 64 pulses. The logic is achieved thanks to logic gates such as AND, OR, NAND, NOR. Every logic variable is driven by a dedicated path till its addressed transistor.

#### 3.2.1.1 Schematics

The first step is to generate every 64 pulses. 2 flip-flops are put into series to divide by 4 the input clock (Fig. 3.25). A reset function guarantee the right synchronization of both flip-flops. Once divided by 4, they are 4 clock signals delayed by an input clock period from each other. It is just a matter of recombination between them to create each of the 4 pulses. For instance, let us take the example of the second pulse. It is the logic combination between the clock divided by 4 shifted from 0 AND  $\pi/2$ .

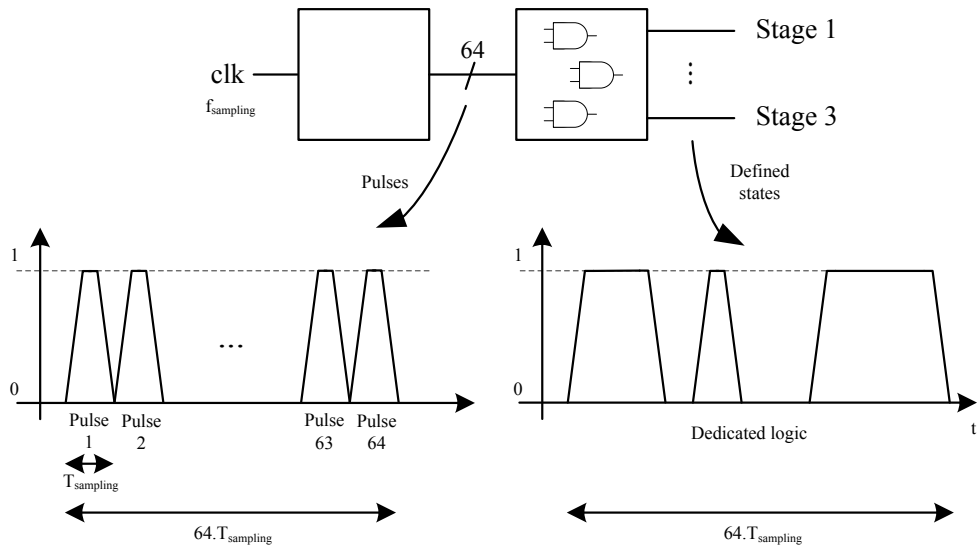


Figure 3.24: Design strategy of the digital circuitry

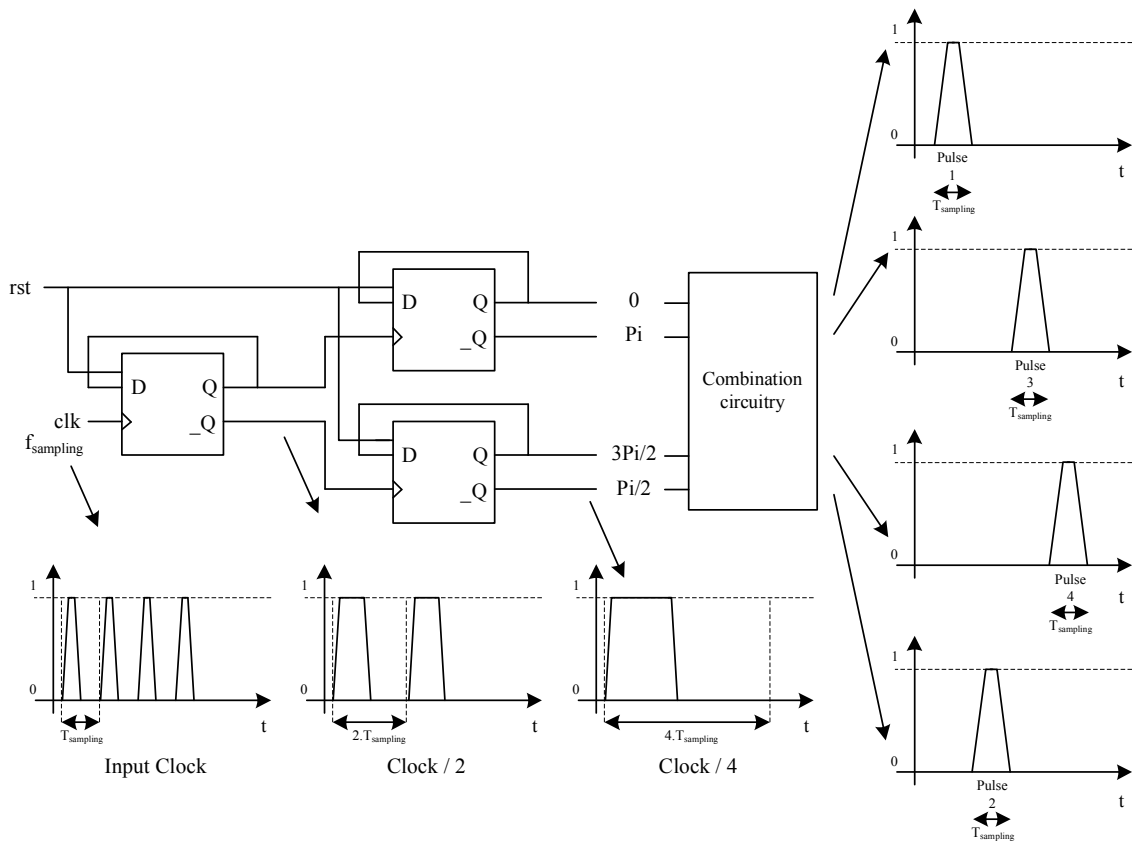


Figure 3.25: Generation of 4 pulses

This way of design is repeated twice to generate every 64 pulses. It is required to have 3 division by 4 to reach the target of a clock divided by 64. Each stage recombines with the previous one its generated pulses. It leads that each stage  $r_{\text{stage}}$  generates  $4^{r_{\text{stage}}}$  pulses with the  $4^{r_{\text{stage}}-1}$  pulses from the previous stage. The digital circuitry has consequently produced every required pulse to govern the analog circuitry. Any logic state will be given by a combination of pulses.

### 3.2.1.2 Simulation results

The design of the flip-flops are based on a library of basic gates. Figure 3.26 presents a simulation of the generation of the first pulse among the base of 4 pulses. It is the combination of the input clock divided by 4, shifted from 0 and  $\pi/2$  as presented in figure 3.25. Figure 3.27 presents a simulation of the 64 pulses. All simulations are set with  $f_{\text{sampling}} = 640\text{MHz}$ . The average power consumption of the digital circuitry is 0.87mA.

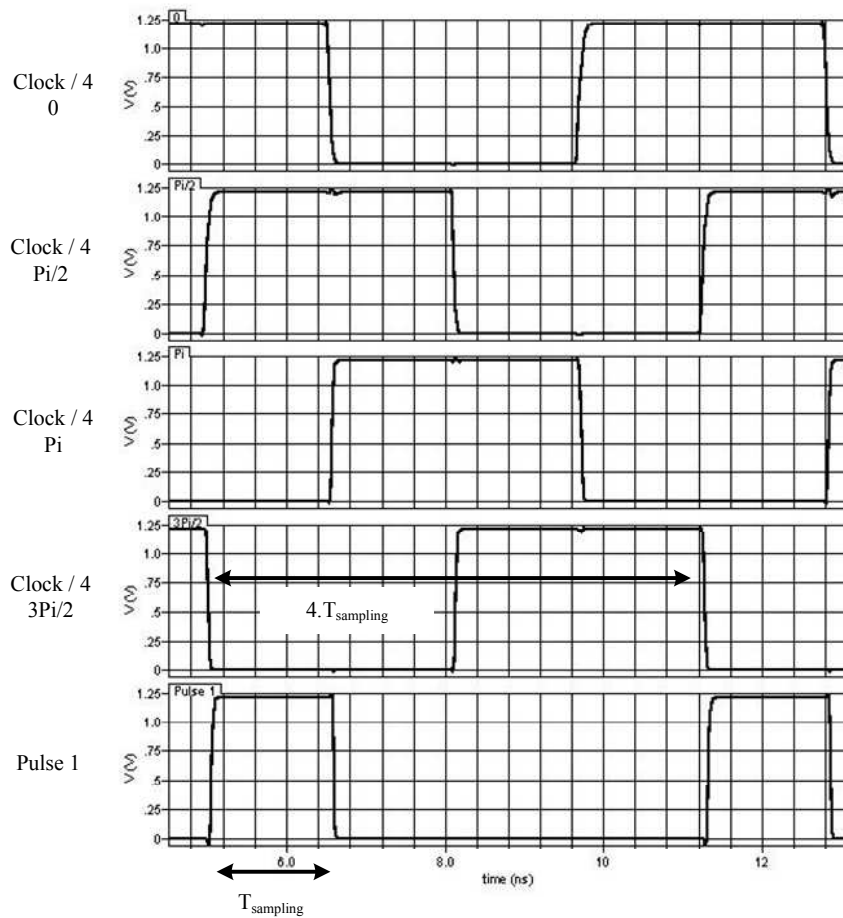


Figure 3.26: Simulation of pulses generations

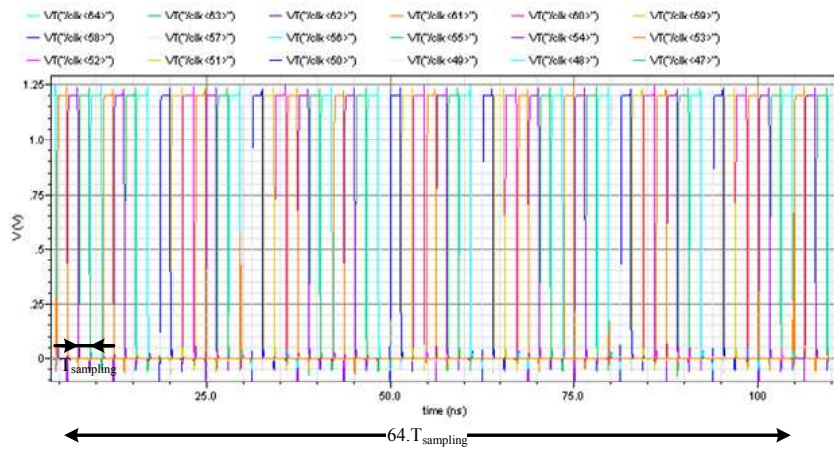


Figure 3.27: Simulation of 64 pulses

Simulations have been done at different frequencies to determine the maximal frequency of the digital part. It is exhibited in figure 3.28. The first pulse generated among 64 is taken as a witness. The maximal frequency in simulation is 2.51GHz. Pulses cannot be generated beyond this theoretical frequency because of the slew rate limitation given by parasitic elements.

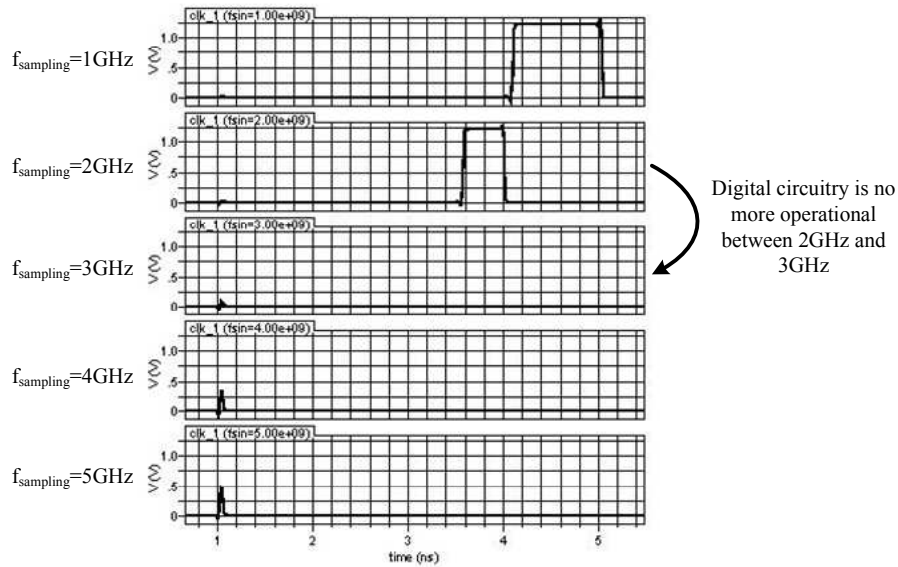


Figure 3.28: Simulation of the maximal  $f_{\text{sampling}}$

## 3.2.2 A hardware-implemented algorithm

### 3.2.2.1 WU logic circuitry

Any logic state which governs the analog circuitry such as the pre-processing unit, the weighting unit and the delay lines are displayed by a combination of the 64 generated pulses (Fig. 3.24). The combination circuitry is hand designed thanks to a library of basic elements. Let us take the example of the weighting unit of stage 2. Table 3.5 exhibits the coefficients to be applied and figure 3.22 its simulation result. The switch network given by table 3.4 imposes the rules of the combination circuitry shown in figure 3.29.

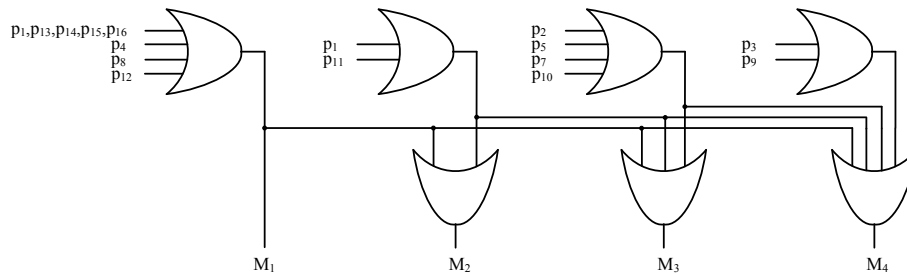


Figure 3.29: Example of a logic circuit to address WU

16 different pulses  $p_x$  are displayed during  $64 \cdot T_{\text{sampling}}$ . Each pulse has a length of  $4 \cdot T_{\text{sampling}}$ . For instance, coefficient 0.707 is used 4 times. In figure 3.22,  $M_1$  and  $M_2$  are at  $gnd$ ,  $M_3$  and  $M_4$  are at  $V_{dd}$ . Consequently, this occurs every time except when  $p_3$ ,  $p_6$  and  $p_9$  are at high state. Figure 3.30 depicts a simulation of this combination. It can be noticed that  $p_3$ ,  $p_6$  and  $p_9$  are displayed in reverse order. This is because the WU is placed before the delay line. The delay line reverses the order of the voltage samples and that is why a "double-reverse" is required to be transparent.

### 3.2.2.2 Sample Selector logic circuitry

One voltage sample among 64 has to be selected at the out of SASP64. This is done by switches on the PCB. Only a binary combination can encode the voltage sample to be selected. It is given by table 3.7. 6 bits encode the binary word. The binary word is written  $\overline{16B_116B_24B_14B_21B_11B_2}^2$ . A logic circuitry to convert the binary word to a base-4 combination is designed (Fig. 3.31). A first part is to selected the right combination of different pulses (their length is  $T_{\text{sampling}}$ ,  $4 \cdot T_{\text{sampling}}$ ,  $16 \cdot T_{\text{sampling}}$ ) to display only one pulse among the 64. This pulse has a width of  $T_{\text{sampling}}$ .

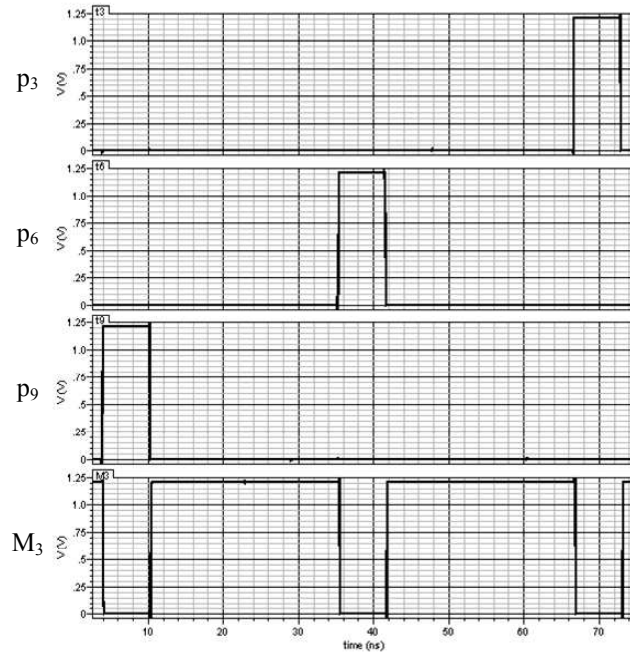


Figure 3.30: Simulation result of a state combination,  $M_3$

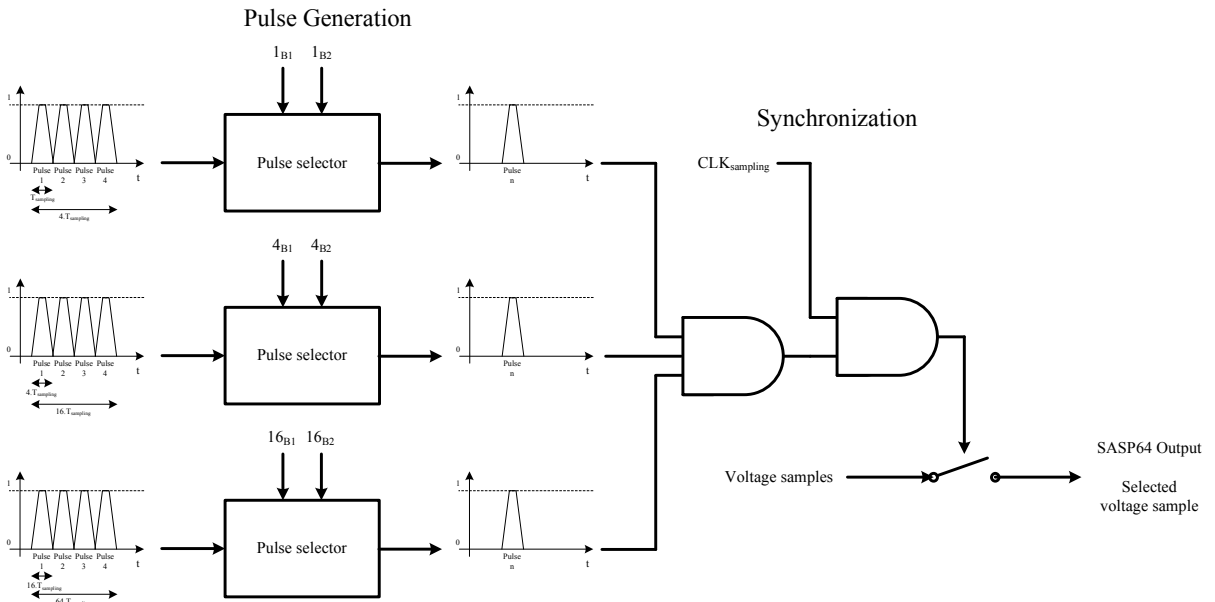


Figure 3.31: Architecture of the Sample Selector

Table 3.7: Binary code

Voltage sample	$16_{B1}$	$16_{B2}$	$4_{B1}$	$4_{B2}$	$1_{B1}$	$1_{B2}$
0	1	1	1	1	1	1
1	0	0	1	1	1	1
2	0	1	1	1	1	1
3	1	0	1	1	1	1
4	0	0	0	0	1	1
5	0	1	0	0	1	1
6	1	0	0	0	1	1
7	1	1	0	0	1	1
8	0	0	0	1	1	1
9	0	1	0	1	1	1
10	1	0	0	1	1	1
11	1	1	0	1	1	1
12	0	0	1	0	1	1
13	0	1	1	0	1	1
14	1	0	1	0	1	1
15	1	1	1	0	1	1
16	0	0	0	0	0	0
17	0	1	0	0	0	0
18	1	0	0	0	0	0
19	1	1	0	0	0	0
20	0	0	0	1	0	0
21	0	1	0	1	0	0
22	1	0	0	1	0	0
23	1	1	0	1	0	0
24	0	0	1	0	0	0
25	0	1	1	0	0	0
26	1	0	1	0	0	0
27	1	1	1	0	0	0
28	0	0	1	1	0	0
29	0	1	1	1	0	0
30	1	0	1	1	0	0
31	1	1	1	1	0	0
32	0	0	0	0	0	1

Then, it is re-synchronized with the input clock. Finally, this synchronized pulse controls the output switch to display or not the voltage sample to a capacitor to be stored during  $64.T_{\text{sampling}}$ . A frequency decimation is done. The output frequency of the SASP64 is  $f_{\text{sampling}}/64$ . A simulation result is depicted in figure 3.32. The configuration chosen is  $\overline{110011}^2$  which corresponds to the 7<sup>th</sup> voltage sample. Although voltage samples are output in base-4 reverse order, the selection pulse is also generated in base-4 reverse order.

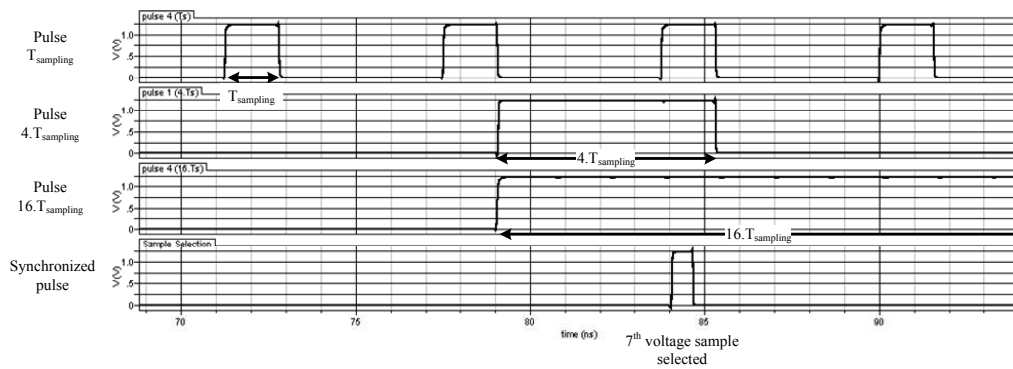


Figure 3.32: Generation of the selection of the 7<sup>th</sup> voltage sample

Figure 3.33 proposes an implementation of the digital part. Every digital element are placed in the same area and surrounded by a guard ring to isolate it from any interference coming from other part of the circuit and to avoid any noise pertubating to analog parts. The layout die area is  $174\mu\text{m} \times 223\mu\text{m}$ .

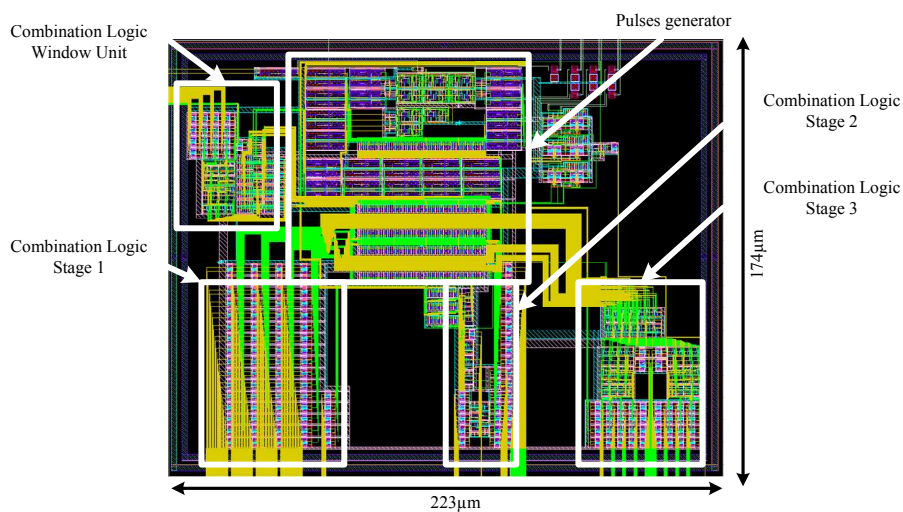


Figure 3.33: Digital part layout



### 3.3 Design - *SASPEPA* and *LUCATESTA*

The designed chip sent to foundry is called SASPEP-A. A test chip is sent at the same time. It is called LUCATEST-A. It aims at validating separated building blocks. Their designs are presented in this section.

#### 3.3.1 Peripheral building blocks

##### 3.3.1.1 Input Adapter Amplifier

The input adapter amplifier is the first building block. It adapts and amplifies the analog input signal before sampling. A resistor of  $50\Omega$  placed in parallel with the input enables to adapt at  $50\Omega$ . Capacitors allow filtering only the AC signals and chose a DC voltage at  $800\text{mV}$ .

##### 3.3.1.2 Sampling Stage

Once the signal is well pre-conditionned, it is sampled. Sampling is the most important part of the system because the resolution of the calculation depends on its accuracy. The sampling frequency  $f_{\text{sampling}}$  determines the FFT timing ( $64 * f_{\text{sampling}}$ ), the spectrum range (from 0 Hz to  $\frac{N * f_{\text{sampling}}}{2} = 32 * f_{\text{sampling}}$ ) and the spectrum resolution ( $\frac{f_{\text{sampling}}}{64}$ ). A differential structure was thus chosen to maximize the accuracy of each voltage sample.

A Track and Hold (T/H) sampler was used to pre-discretize the signal and display the voltage samples to the others stages (Fig. 3.34). CMOS dummy switches are used to avoid accumulated charge in the hold capacitor  $C_{\text{hold}}$ . Equation 3.5 describes how charge can be accumulated in  $C_{\text{hold}}$ . It depends on carriers mobility  $\mu_n$ , the gate capacitor  $C_{\text{ox}}$ , the drain and source voltage  $V_D$ ,  $V_S$ , their associated capacitors  $C_{D\text{tot}}$ ,  $C_{S\text{tot}}$  and the aperture time of the switch  $t_c$ . In fact, when the switch goes off, charge stored in the channel is evacuated in the hold capacitor. It adds unwanted charge to the output sampled signal which is then degraded (Eq. 3.5). The dummy switch architecture compensates this charge injection if design rules are respected, e.g. the  $M_1$  width gate is twice as large as  $M_2$  (Fig. 3.34). Charge injected by  $M_1$  are collected by  $M_2$  which is in shortcircuit configuration. A simulation result is proposed in figure 3.35. The input signal is first sampled, then buffered. The two phases of Track and Hold are presented.

$$(\delta Q) \approx \mu_n C_{\text{ox}} W (V_D - V_S - V_{\text{th}})^2 \left( \frac{1}{C_{D\text{tot}}} - \frac{1}{C_{S\text{tot}}} \right) \frac{t_c}{6} \quad (3.5)$$

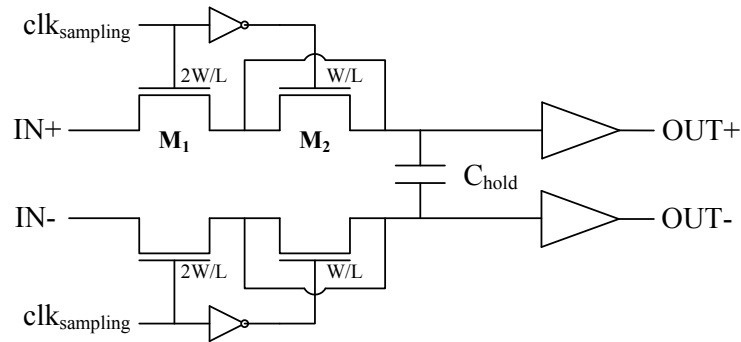


Figure 3.34: Sampler Architecture

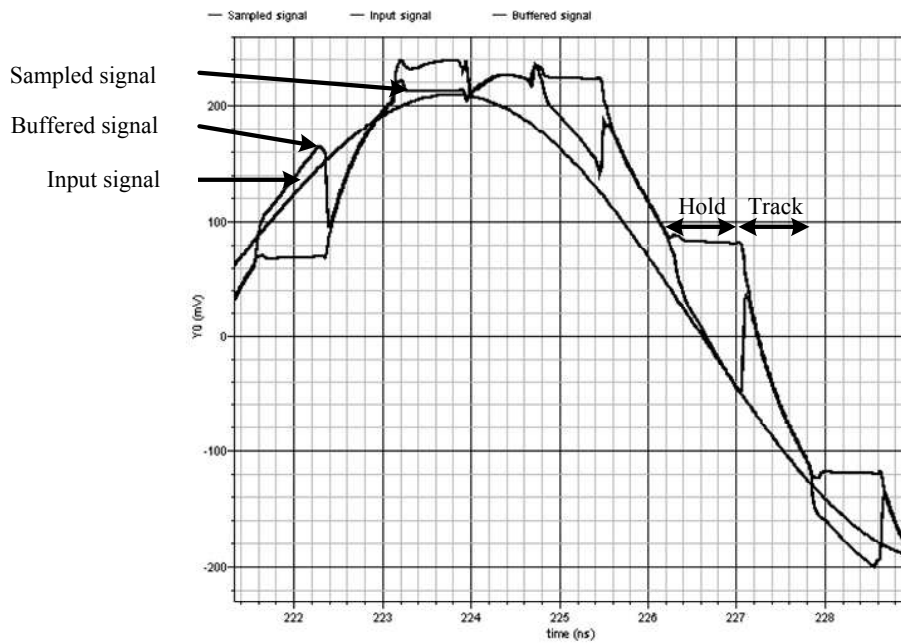


Figure 3.35: Sampler simulation

### 3.3.1.3 Windowing

Hamming window equation (Eq. 2.17) is decomposed into two parts. Weighting operation is performed by applying a factor within the interval  $[0, 1]$  which corresponds to  $0.46 \cdot \cos(2\pi \frac{t}{64 \cdot T_{\text{sampling}}})$  and adding the weighted input by a factor of 0.54 (Fig. 3.36). The principle of the voltage/current/voltage conversion presented before is used to carry out this analog operation.

An extended switch network ( $S_x$ ) connected to 16 transistors is proposed. It selects the input voltage of each transistor ( $M_x$ ) gate which has a different ratio ( $W_x/L$ ). The input voltage can be either a voltage sample to be weighted or the DC reference voltage (800mV). Consequently, the current crossing each transistor is proportional to the width of the selected transistors. This configuration carries out every 16 coefficients. Behavioural simulation is exhibited in figure 3.37 to illustrate this signal processing. The input signal is a constant voltage of 100mV. The output is the expected Hamming window itself. This part will be measured in LUCATESTA chip.

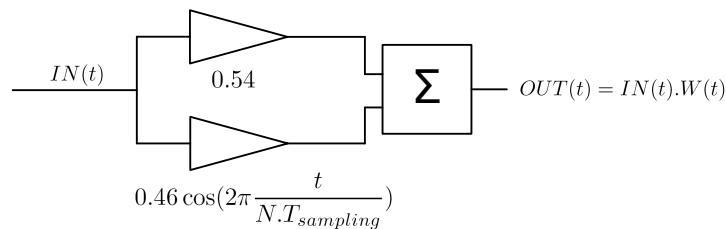


Figure 3.36: Windowing circuit (simplified schematic)

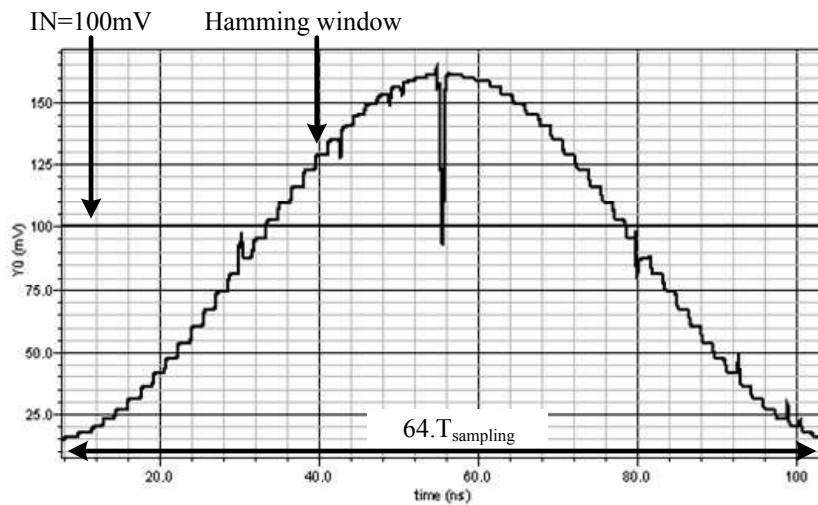


Figure 3.37: Simulation result of a Hamming Window

### 3.3.1.4 Output buffers

High impedance buffers are chosen to output the selected voltage sample. The maximal amplitude of the voltage sample before amplification is set at 200mV. It is amplified with a gain of 3. The maximal output voltage is consequently 600mV to maximize accuracy for measurements.

### 3.3.2 Layout considerations

SASPEPA is essentially a RF analog circuit. It also has digital blocks such as the clock generator. To achieve a good accuracy and linearity, a great deal of caution should be taken in the layout design to reduce the effects of parasitics, especially for RF lines and noise coupling from digital blocks to analog blocks.

The chip is very complex. A design methodology is created. To ensure the feasibility of the circuit, every power supply of each part of the circuit are separated. To improve isolation between analog and digital blocks, some design techniques such as shielding and guard ring are used. Decoupling capacitors are placed in every free area and as close as possible to the more critical blocks.

### 3.3.3 A building block library

A building block library is designed to relax the complexity of the design. A bottom-up methodology can be applied. First, basic blocks are designed. They are linked between them to form bigger parts of the circuit. If at a given instant, one of them does not respect the specifications, the inner blocks are redesigned till an acceptable simulation is obtained. Figure 3.38 depicts the design order of each block.

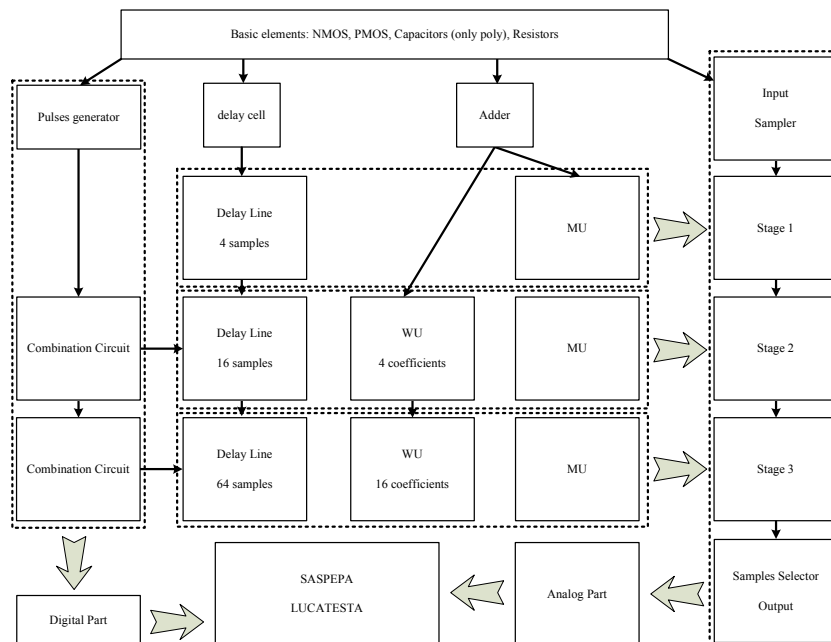


Figure 3.38: Design strategy

Each stage, from the first to the third are co-designed with their digital part. The three stages are then connected to input and output part to form the analog part and can be simulated with the whole digital part. Post-Layout Simulations are performed to extract parasitic elements coming from connections between each part. This parasitic elements can prevent from achieving good performances, such as parasitic capacitors which reduce voltage sample loading time.

### 3.3.3.1 Parasitic considerations

The choice of metal was done depending on signals conveyed. Table 3.8 depicts the selected metals for each signal. Figure 3.39 presents each stage layout. Each part of each stage are named (Delay Line, MU, WU). A layout of SASPEPA is presented in figure 3.40.

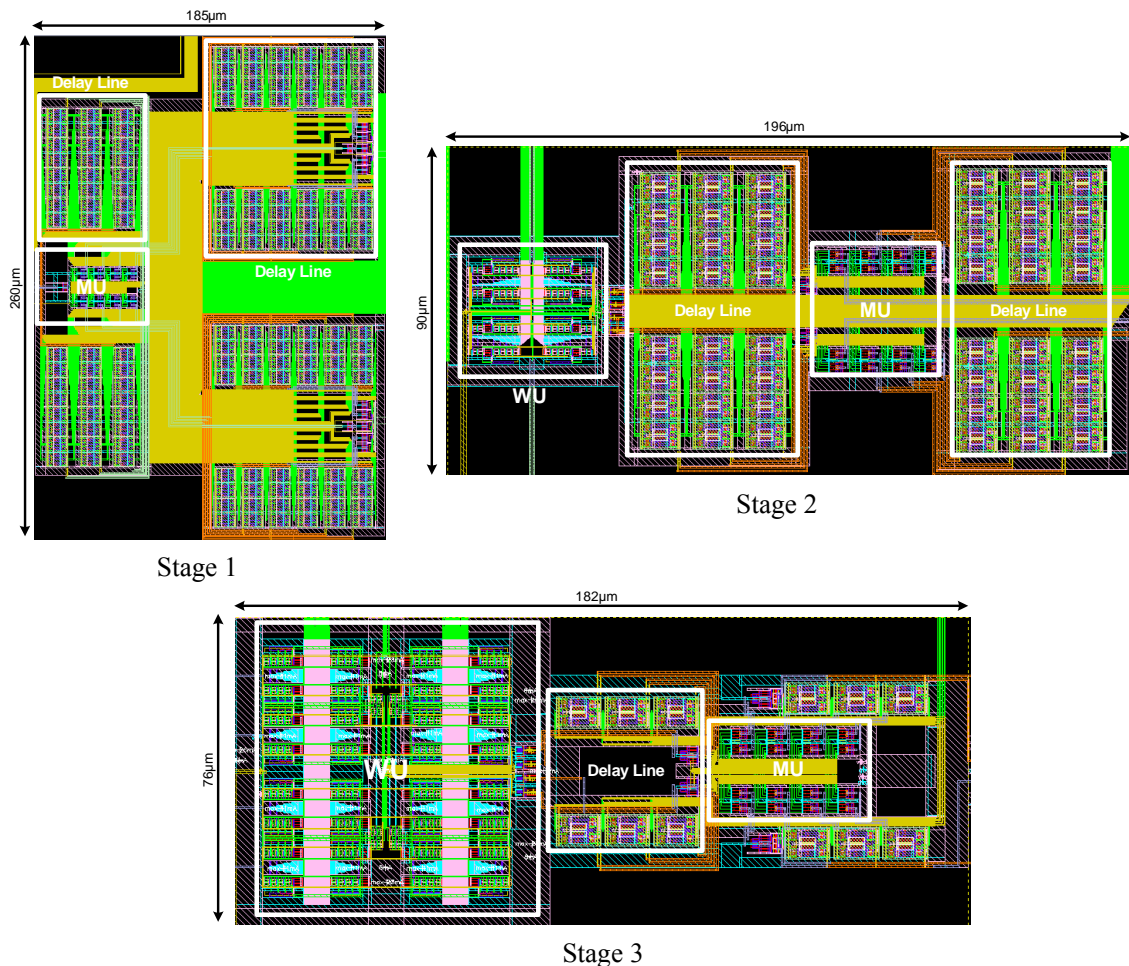


Figure 3.39: Stage 1, Stage 2 and Stage 3 layouts

Table 3.8: Metals

Metal	Signal
2	GND
3	VDD
4, 5	Clock signals
6, 7	RF signals

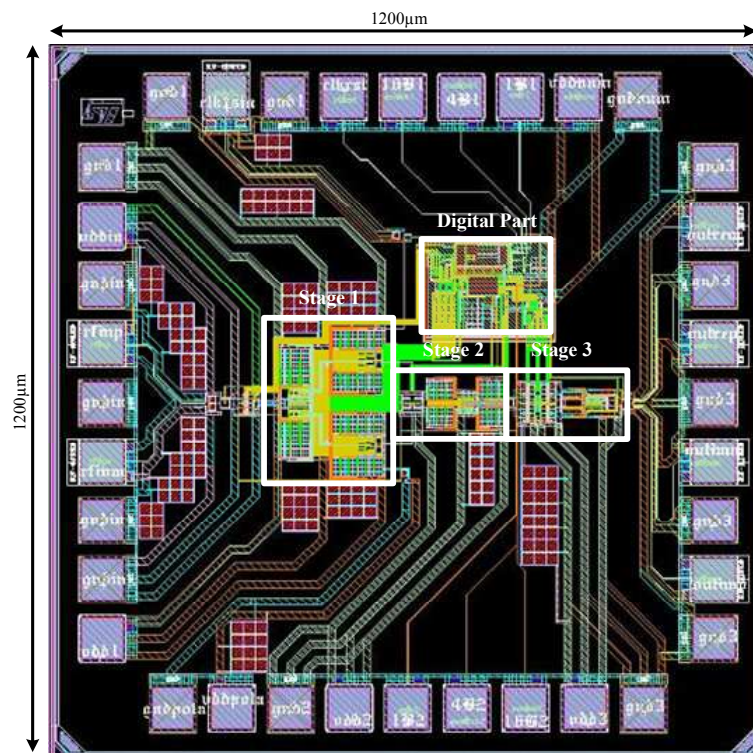


Figure 3.40: SASPEPA Layout

### 3.3.4 Post-Layout Simulations

Post-Layout Simulations (PLS) are performed. The whole SASPEPA is simulated with capacitor parasitic elements.

#### 3.3.4.1 Overall Circuit Simulation Results

It is chosen a sampling frequency of  $f_{\text{sampling}} = 640\text{MHz}$  for a matter of simplicity.  $T_{\text{sampling}} = 1.5625\text{ns}$  and a full FFT duration is  $64.T_{\text{sampling}} = 100\text{ns}$ . Figure 3.41 depicts the simulation of the clock generation. The input signal of the SASPEPA is a sinewave at  $\frac{7*f_{\text{sampling}}}{64} = 70\text{MHz}$ . Its amplitude is 200mV. It is first windowed as presented in figure 3.42.

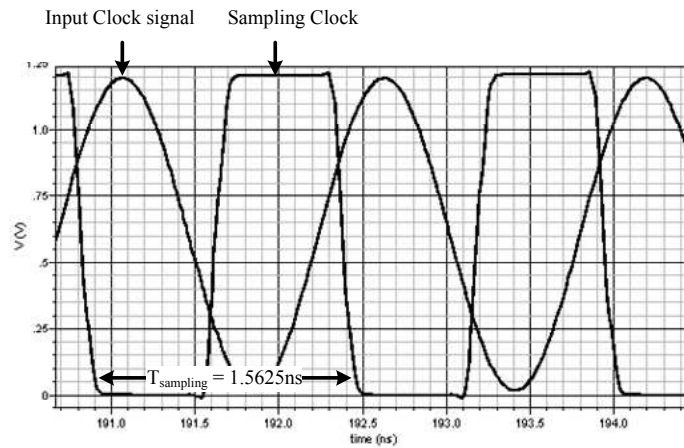


Figure 3.41: PLS of clock signal generation

The spectrum processed can be identified as pulses of the Fourier Transform of a sinewave. 6 pulses are identified (Fig. 3.43). They are recognized to be the 6<sup>th</sup>, 7<sup>th</sup>, 8<sup>th</sup> and the 56<sup>th</sup>, 57<sup>th</sup>, 58<sup>th</sup>. They are replaced in order in figure 3.44. 7<sup>th</sup> and 57<sup>th</sup> pulses correspond to the input signal spectrum. Others are due to the windowing operation and correspond to frequencies equal to  $\frac{7*f_{\text{sampling}}}{64} \pm \frac{f_{\text{sampling}}}{64} = \frac{(6;8)*f_{\text{sampling}}}{64}$ . Only the 7<sup>th</sup> is selected to be displayed at the output of the SASP64. Figure 3.45 depicts the voltage sample selection. Once selected, the sample is buffered in order to maximize its measurement. The gain is about 3.

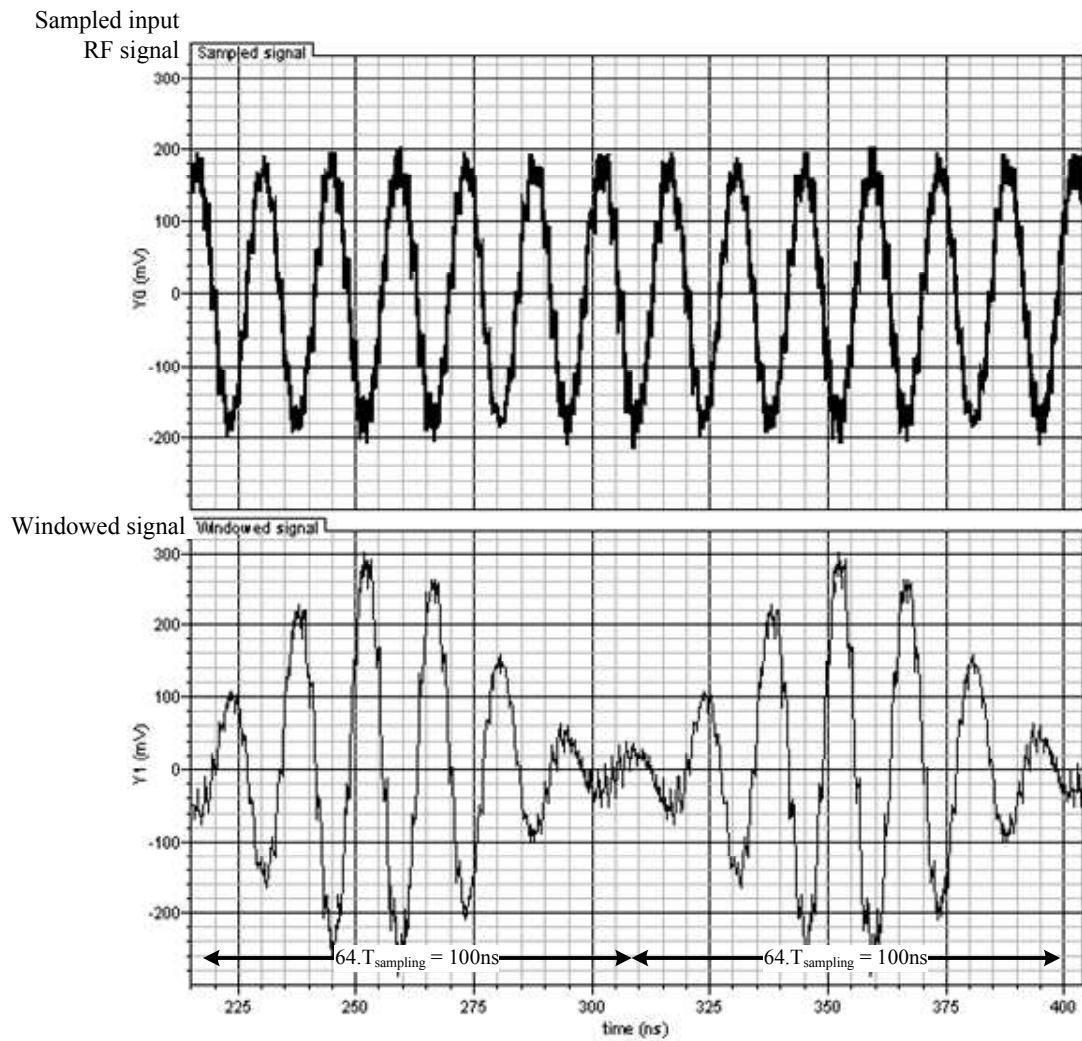


Figure 3.42: PLS of windowing operation



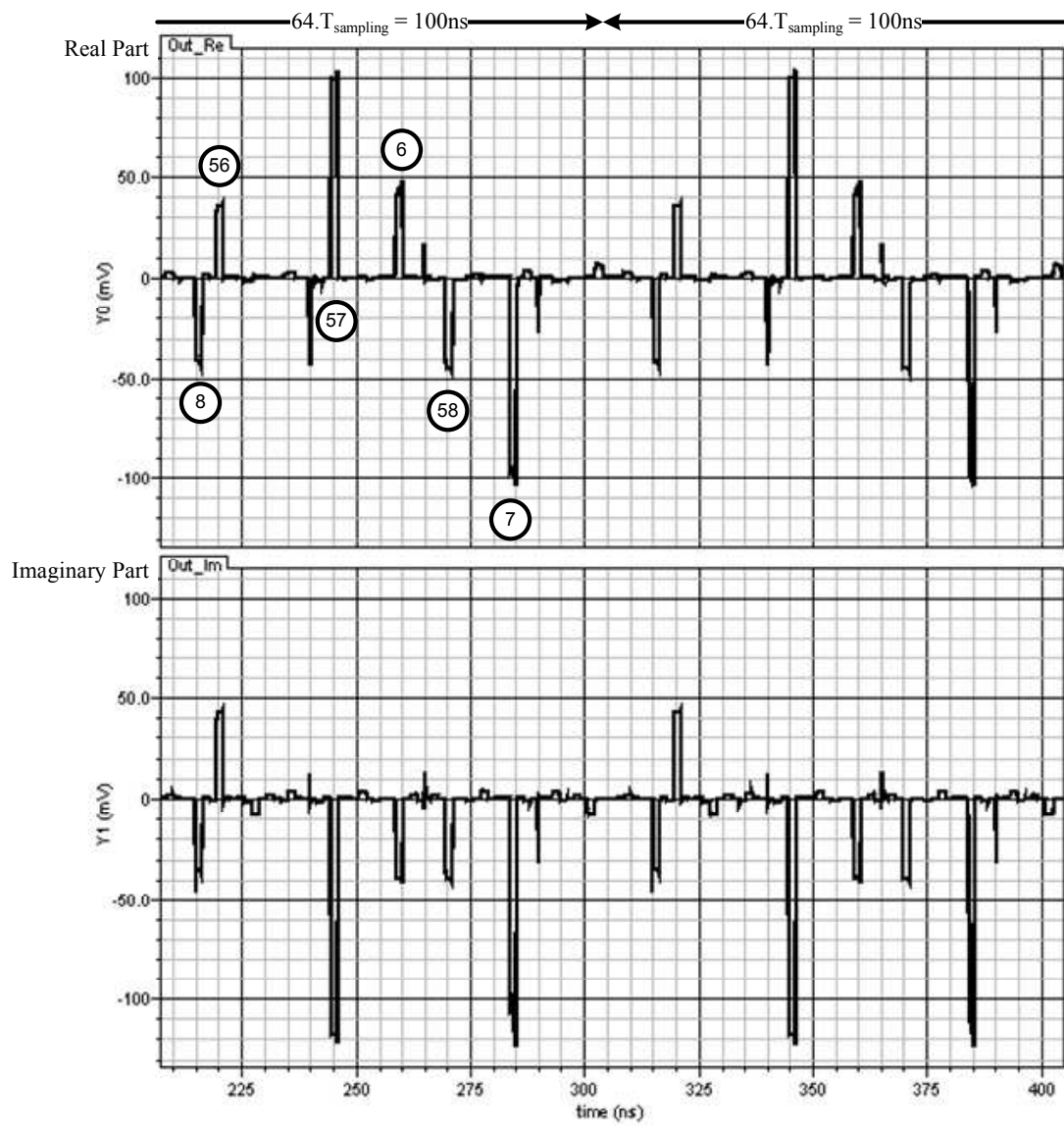


Figure 3.43: PLS of output spectrum

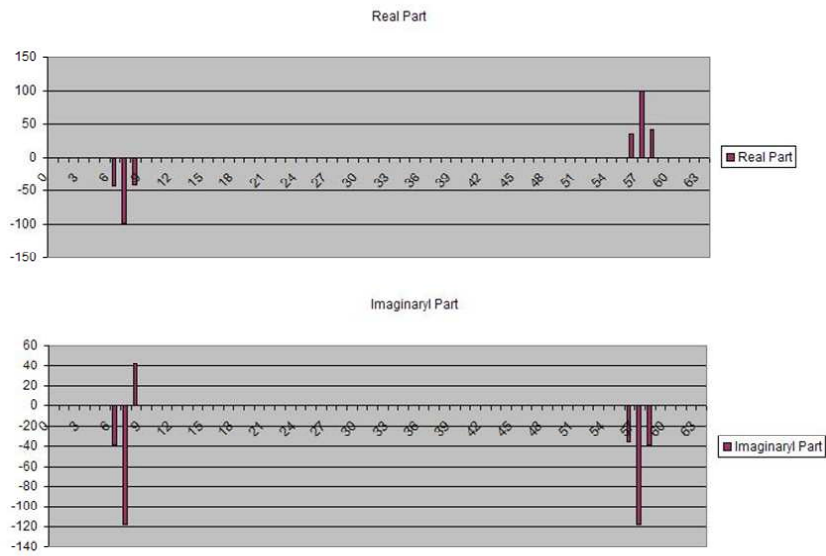


Figure 3.44: Output spectrum in order

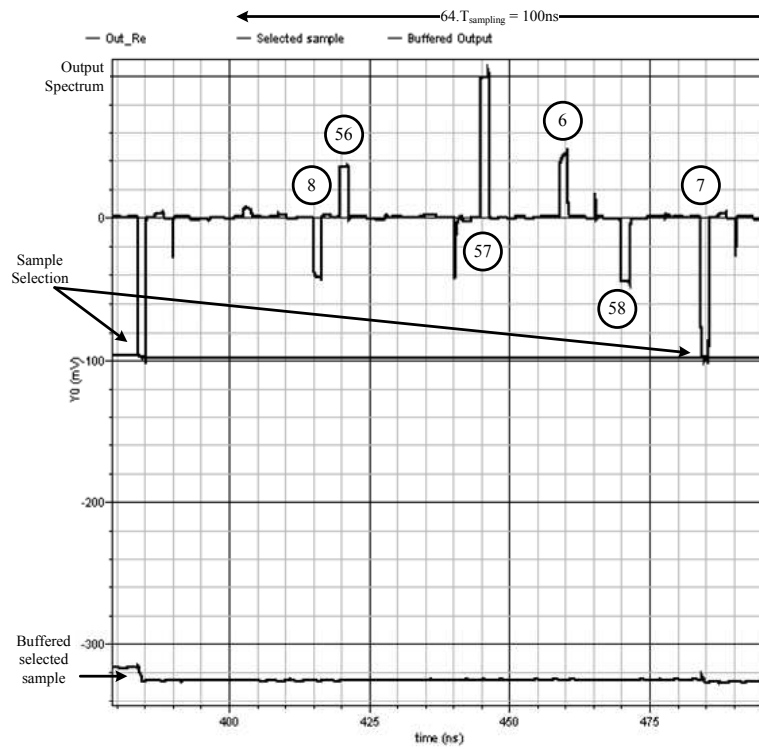


Figure 3.45: PLS of the sample selection

A multitone simulation is done to check if SASPEPA can detect several RF signals. The first one is done with two frequencies:  $f_{in1} = 70MHz$  with an amplitude of 200mV and  $f_{in2} = 110MHz$  with an amplitude of 100mV. Figure 3.46 and figure 3.47 depict the simulation result. Both frequencies can be identified in the 7<sup>th</sup> and 11<sup>th</sup> samples. This lightens the feasibility of a frequency demodulation, such as in the case of Frequency Shift Keying (FSK) modulations. The SASP is able to select only the voltage sample corresponding to frequencies encoding digital informations. An other application can be a direct channelization, for instance in the case of Orthogonal Frequency Division Multiplexing (OFDM) modulation.

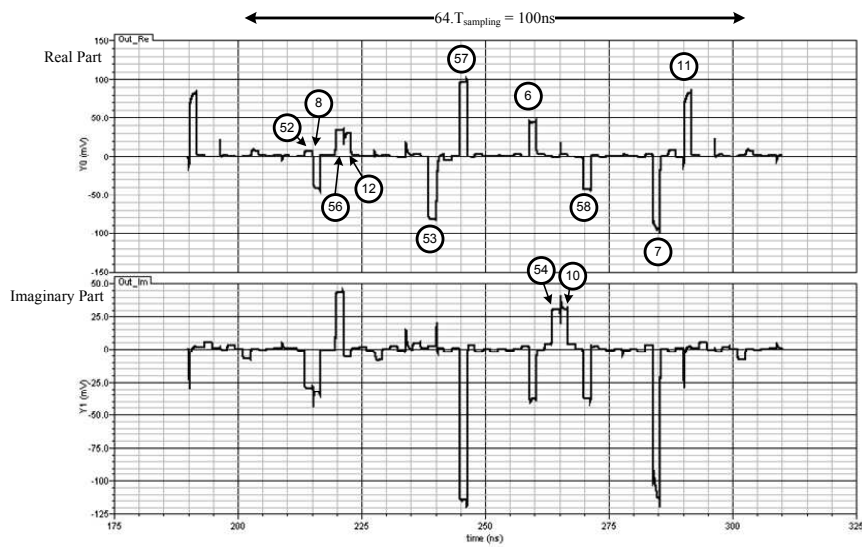


Figure 3.46: PLS of output spectrum

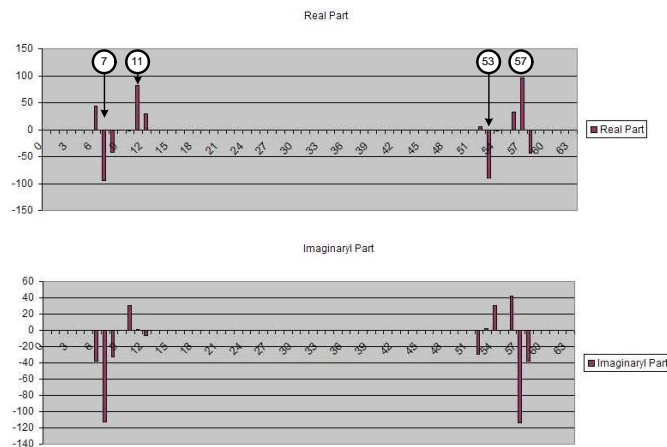


Figure 3.47: Output spectrum in order

An other simulation is done but with  $f_{in1} = 70MHz$  with an amplitude of 200mV and  $f_{in2} = 80MHz$  with an amplitude of 100mV. It can be noticed in figure 3.48 and figure 3.49 that interferences between both signals prevent from extracting any information from their spectrum.

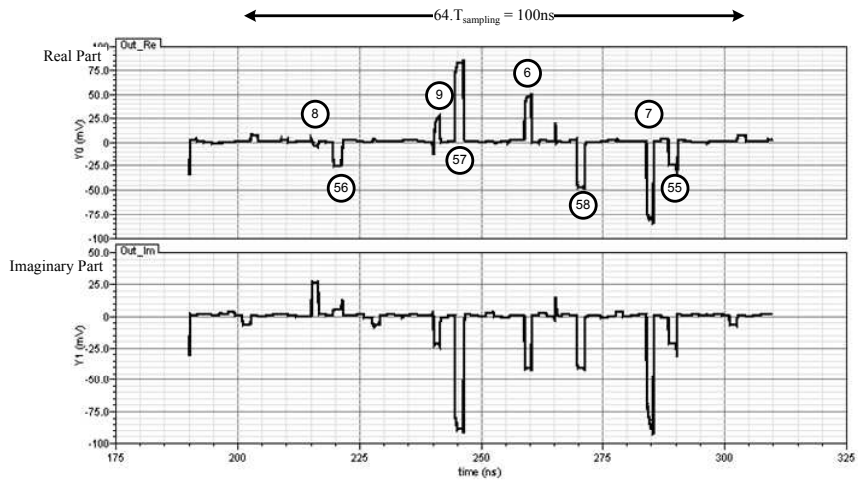


Figure 3.48: PLS of output spectrum

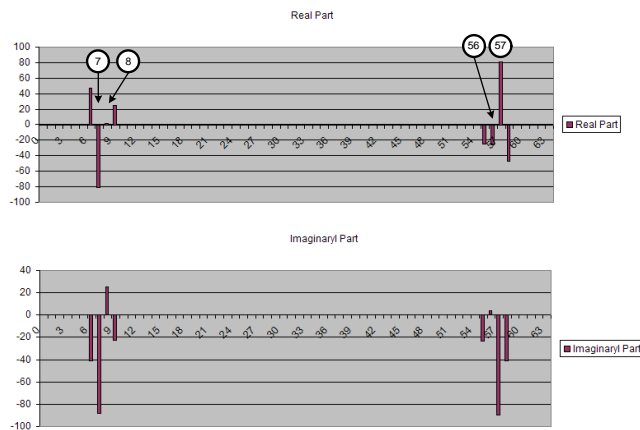


Figure 3.49: Output spectrum in order

A sinewave equal  $\frac{7.25 \cdot 640MHz}{64} = 72.5MHz$  is sent (Fig. 3.50). The sampling frequency is still the same ( $f_{sampling} = 640MHz$ ). A phase shift is observed from one FFT period to the next. The sample selected in each processed spectrum exhibits the phase variation. As the sinewave input frequency is a quarter higher than an integer part of the sampling frequency, the spectrum

is no more coherent from an FFT to an other. The phase is shifted of  $\frac{\pi}{2}$  at each processed FFT. The output signal frequency is  $\frac{7.25*640MHz}{64} - \frac{7*640MHz}{64} = 2.5MHz$ . It depicts that the sampling frequency  $f_{sampling}$  can be one of the main parameter to configure the FFT processing. Phase and amplitude recovery can be done using directly the voltage samples. This lightens the feasibility of an other frequency demodulation, such as in the case of Phase Shift Keying (PSK) modulations.

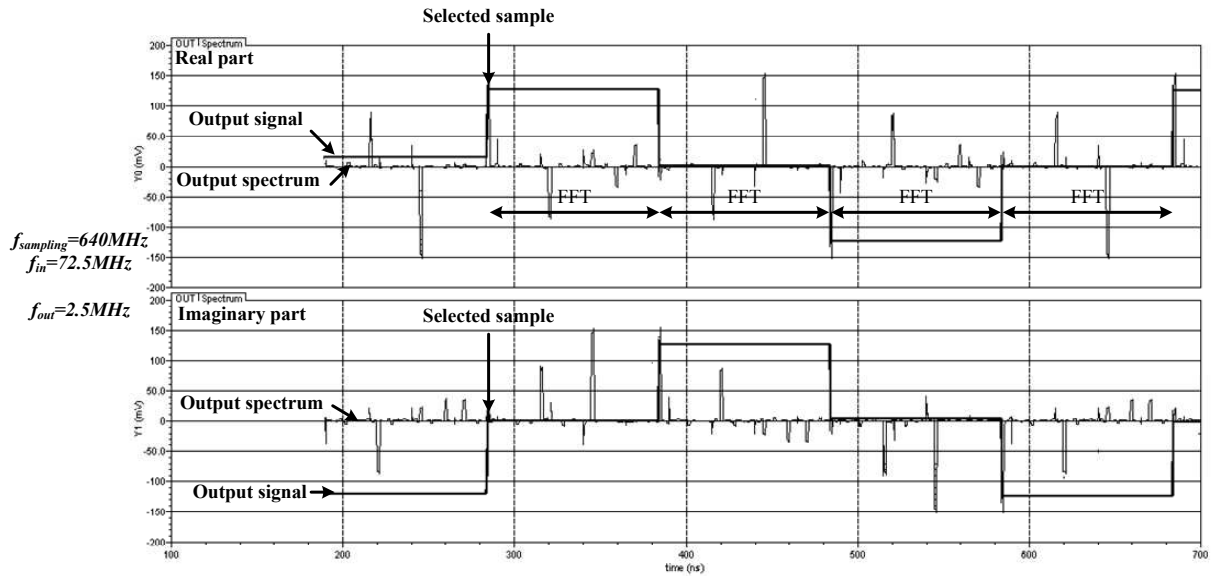


Figure 3.50: A non-entire frequency sinewave processed by 64-point SASP

### 3.3.4.2 SASPEPA Characteristics

SASPEPA was sent to foundry. It was designed using the Design Kit 65nm CMOS from STMicroelectronics. Its minimum working frequency is  $f_{sampling} = 100MHz$ . Its maximal working frequency is limited to  $f_{sampling} = 1GHz$  whereas a 10GHz is targeted for Software Radio applications. This limitation is chosen because this thesis is focused on the feasibility and not the performances. SASPEPA allows receiving any RF signal in a frequency range from 50MHz up to 500MHz. The input dynamic range is 200mV with a 50Ω impedance. The output dynamic range is 600mV with a high impedance. The circuit is supplied under 1.2V. The whole power consumption is 300.2mA (Tab. 3.9).

Table 3.9: Power Consumption

Stages	Delays Lines	Weighting Units	Matrix Unit	Total
1	101.6mA	N/A	22.8mA	124.6mA
2	38.4mA	6.44mA	22.8mA	61.2mA
3	10.8mA	56.5mA	22.8mA	90.1mA
Digital Circuit	N/A	N/A	N/A	0.8mA
Others	N/A	N/A	N/A	23.5mA
Total	150.8mA	62.94mA	68.4mA	300.2mA

### 3.4 Conclusion

SASP64 was designed using 65nm CMOS technology from STMicroelectronics. Two chips called SASPEPA and LUCATESTA have been sent to foundry to validate physical feasibility. This chapter has presented schematics of analog building blocks of SASP64. Every discrete analog operation was detailed. Architecture of a stage was discussed through its three main blocks: the delay line, the Matrix Unit, the Weighting Unit. Behavioral simulations and modeling results were exhibited.

The design methodology was exposed to explain how building blocks were designed in a complete and efficient flow. Post-Layout Simulations were performed to ensure good performances of SASPEPA and LUCATESTA. Simulation results have also presented applications of the SASP such as frequency demodulation applied to PSK, FSK and AM modulations. Chapter 4 exposes chips measurements and technological perspectives.



---

# Measurements and Perspectives

## Contents

---

<b>4.1</b>	<b>Test Setup and Experimental Results . . . . .</b>	<b>128</b>
4.1.1	Test setup . . . . .	128
4.1.2	SASP validation measurements . . . . .	131
4.1.3	SASP applications measurements . . . . .	135
4.1.4	SASPEPA Characteristics . . . . .	140
<b>4.2</b>	<b>An open window to RF applications - Achievement of SASP65K . .</b>	<b>142</b>
4.2.1	Schematic perspectives . . . . .	142
4.2.2	Technology issues . . . . .	145
4.2.3	Signal processing accuracy . . . . .	145
4.2.4	Real-Time error correction . . . . .	145
<b>4.3</b>	<b>Conclusion . . . . .</b>	<b>146</b>

---



Chapter 4 presents the measurements of the designed chips LUCATESTA and SASPEPA. Discrete-Time analog operations and SASP principles are validated through different phases to draw further technological improvements. Characteristics are given and a technological roadmap is paved to an industrial product.

**Key words:** measurements, applications, frequency demodulation, perspectives.

## 4.1 Test Setup and Experimental Results

### 4.1.1 Test setup

Chips SASPEPA and LUCATESTA were fabricated in 65nm CMOS process from STMicroelectronics and packaged in a 44-pin ceramic package (CQFP044). Within the chip SASPEPA, power supplies of different blocks were separated in order to govern each part of the circuit independently and to obtain better isolation from parasitic signals which can spread the substrate. Table 4.1 presents the different blocks and their corresponding power supplies. Figure 4.1 shows the floorplan of SASPEPA. Within the chip LUCATESTA, power supplies are connected together but inner circuit signals are displayed to measurements. It permits to validating different analog operations carried out by the circuit. Figure 4.2 shows the floorplan of LUCATESTA.

Table 4.1: Different power supplies

Power Supply	Supplied Blocks
$VDD_{IN}$	Input buffers, sampling stage
$VDD_{POLA}$	Biasing stage
$VDD_{NUM}$	Digital Part
$VDD_1$	Stage 1
$VDD_2$	Stage 2
$VDD_3$	Stage 3, output buffers

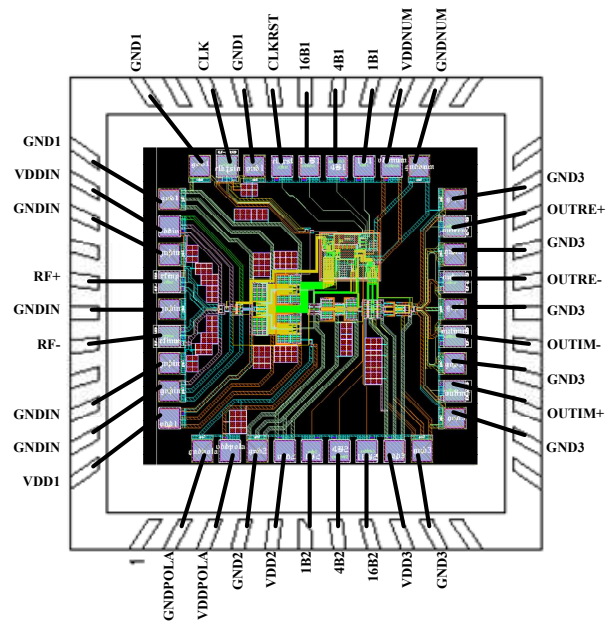


Figure 4.1: Floorplan of SASPEPA

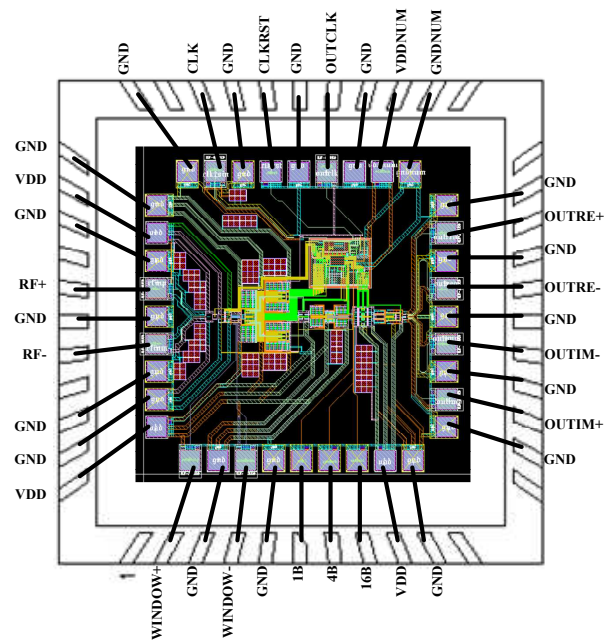


Figure 4.2: Floorplan of LUCATESTA

A two layers FR4 Printed Circuit Board (PCB) has been designed to test the prototype chip (Fig. 4.3). The ground plane is common to every blocks. The ground plane was implemented in the bottom of the PCB. Capacitors are placed as close as possible to the Device Under Test (DUT) to provide low and high-frequency decoupling. Digital inputs ( $1B_1$ ,  $1B_2$ ,  $4B_1$ ,  $4B_2$ ,  $16B_1$ ,  $16B_2$ , CLKRST) are displayed by switches connected either to VDDNUM or to GNDNUM.

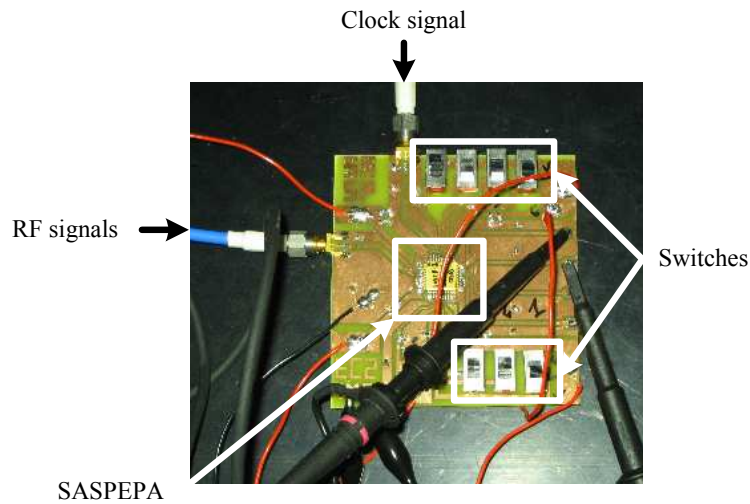


Figure 4.3: Test board of SASEPA

Input RF lines are the most critical. A coplanar configuration was chosen to perform a  $50\Omega$  adaptation in the desired range of frequency (50 to 500MHz) (Fig. 4.4). The waveguide improves the circuit isolation by surrounding the analog traces by guard traces connected to the ground plane. Vias are placed to connect ground plane to PCB bottom ground plane. It allows to have a common ground potential on all the board. Propagation delays are taken into account by designing a symmetrical PCB. RF input and output lines are symmetrical.

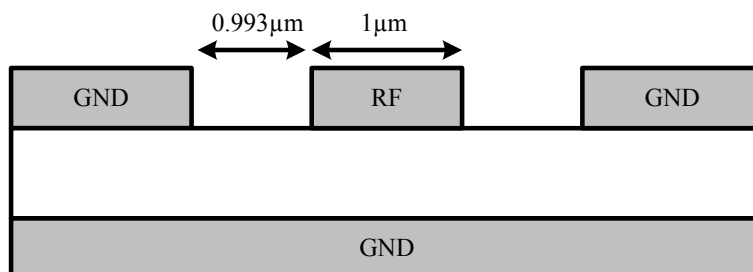


Figure 4.4: Coplanar waveguide

Figure 4.5 depicts the configuration of the instruments used to validate the prototype. The RF input is generated by HP E4433B generator. The clock signal is generated by the HP 83712B generator. A generator HP 8648B is used to synchronise the oscilloscope Lecroy WavePro960. Every RF generators are synchronized by their 10MHz-synchronization signal.

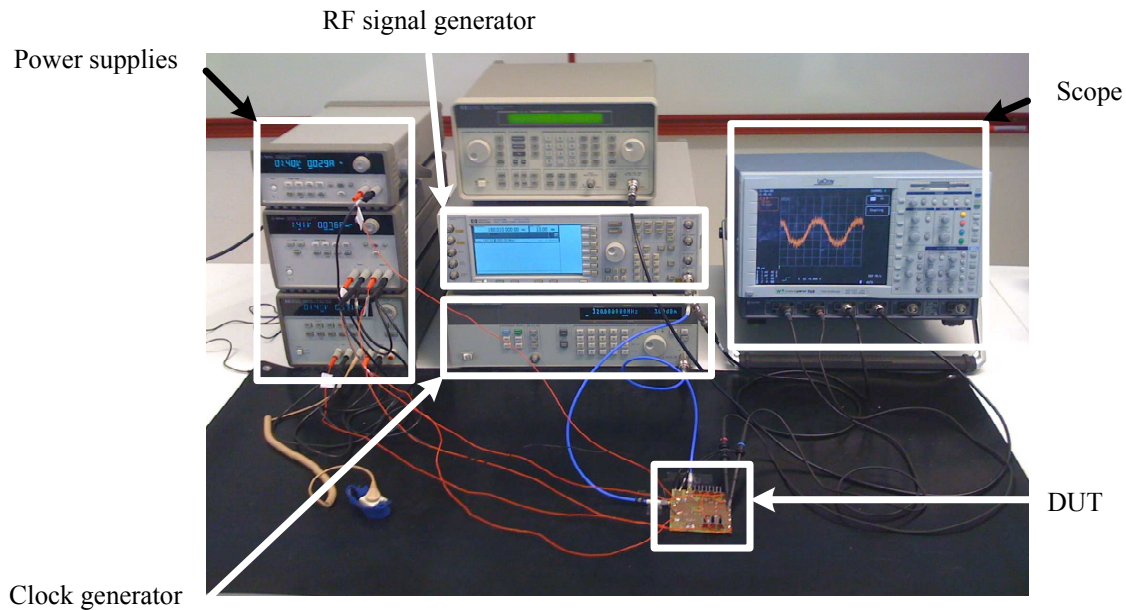


Figure 4.5: Photo of instruments configurations

#### 4.1.2 SASP validation measurements

Measurements were performed on *LUCATESTA*. It aims at validating the digital part, weighting and addition of voltage samples.

- Clock frequency is set at  $f_{\text{sampling}} = 64\text{MHz}$ . Switches are configured to cover every possible combination. OUTCLK (Fig. 4.2) gives the position of the pulse in a FFT sequence by a combination of the input clock and generated pulses (Fig. 4.6). Figure 4.7 depicts the collected information versus the switches configuration. Table 4.2 presents the interpretation of the position of each pulse. Deducted position is an integer number and rounded-off deducted positions are consistent with expected positions. Order of pulses is correct. The digital part including pulse generator and combination logic parts is validated.  $f_{\text{sampling}}$  is increased till measurements are no more consistent. The maximal sampling frequency is observed to be 1.6GHz.

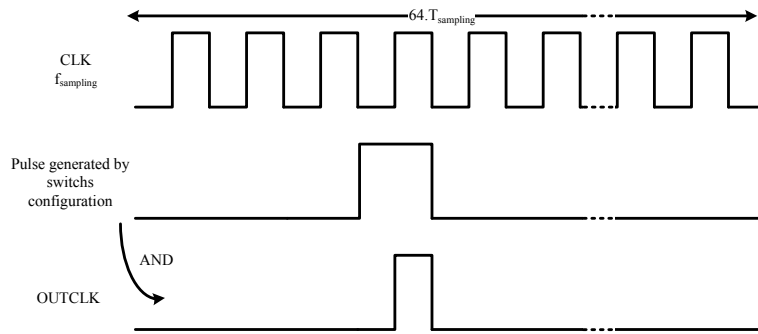


Figure 4.6: OUTCLK signal generation

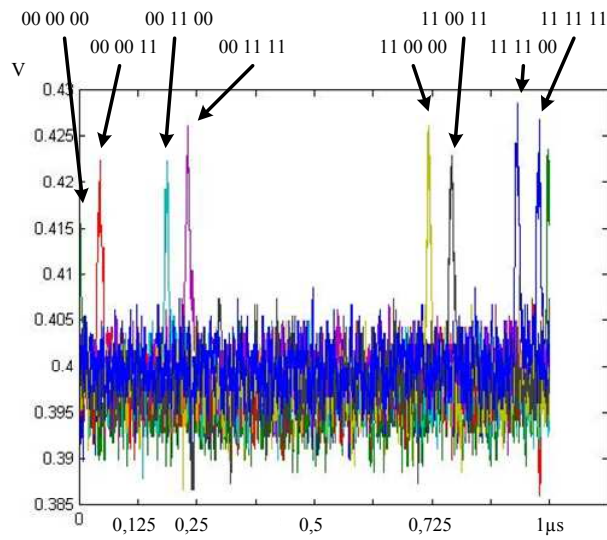


Figure 4.7: OUTCLK measurements

Table 4.2: Digital part validation

Bit	Expected Position	Measured Position	Deducted Position
00 00 00	0	0	0.0
00 00 11	3	0.04425	2.8
00 11 00	12	0.18775	12.0
00 11 11	15	0.23225	14.9
11 00 00	48	0.7425	47.6
11 00 11	51	0.793	50.8
11 11 00	60	0.9325	59.7
11 11 11	63	0.977875	62.6

- A sinewave at the frequency of  $f_{in} = 160MHz$  was windowed with a sampling frequency of  $f_{sampling} = 640MHz$  (Integer numbers were chosen for a matter of simplicity). Figure 4.8 exhibits the windowed input signal and confirms the feasibility of discrete analog operations at high frequencies.

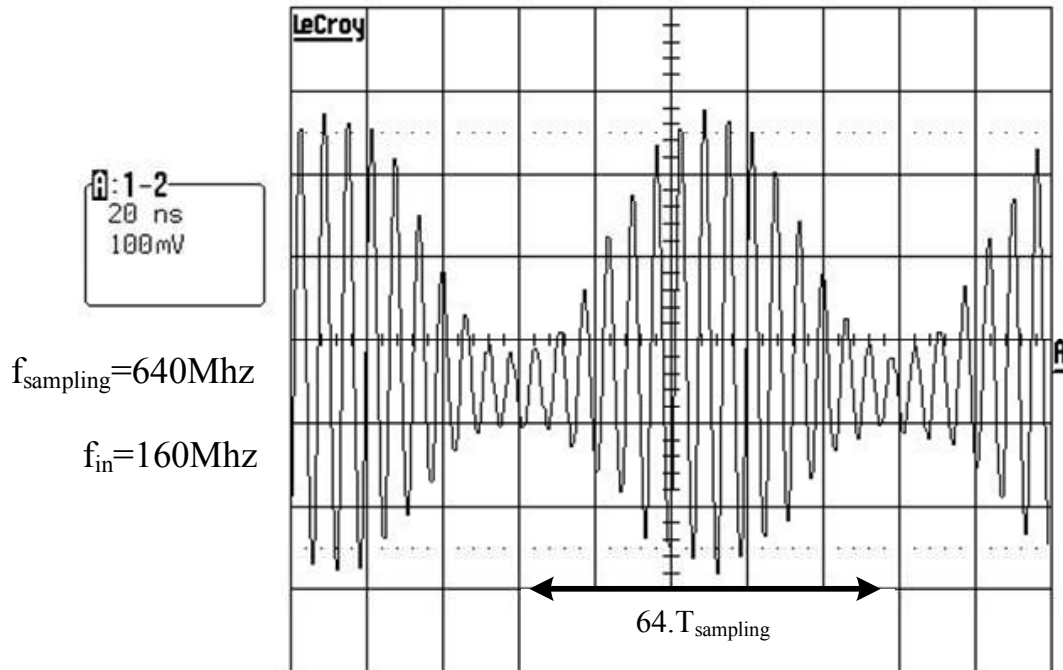


Figure 4.8: Measured hamming window

Both RF inputs are now set to ground. Figure 4.9 depicts the measurements which do not display the expected result. It should be a zero signal whereas a signal with a frequency of  $f_{error} = \frac{1}{T_p}$  is displayed. It is the frequency of the Hamming window. Inquiries lead to conclude that the differential structure was designed too weakly. The mismatching between plus and minus part of the circuit is too important and introduces a DC component. A retro-simulation is done with differences on biasing voltage between plus and minus part of the circuit. The same signal measured in figure 4.9 is obtained in simulation in figure 4.10. The design weakness is consequently pointed out. This is observed at the beginning of signal processing, and it can be expected as a major problem at the output of the SASP.

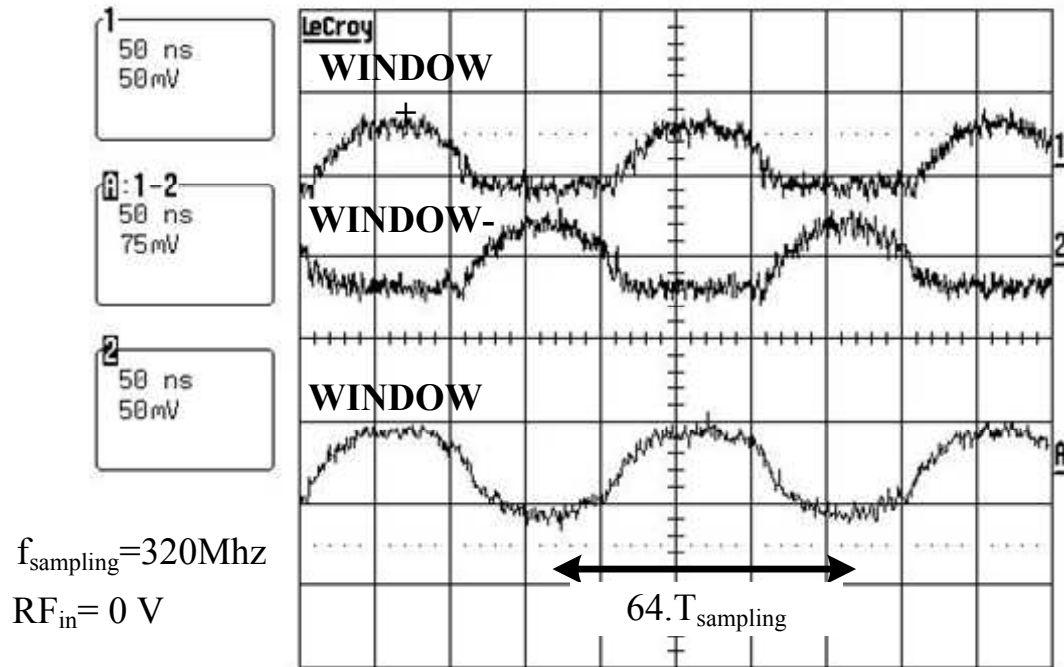


Figure 4.9: Measured error on hamming window

Hamming window

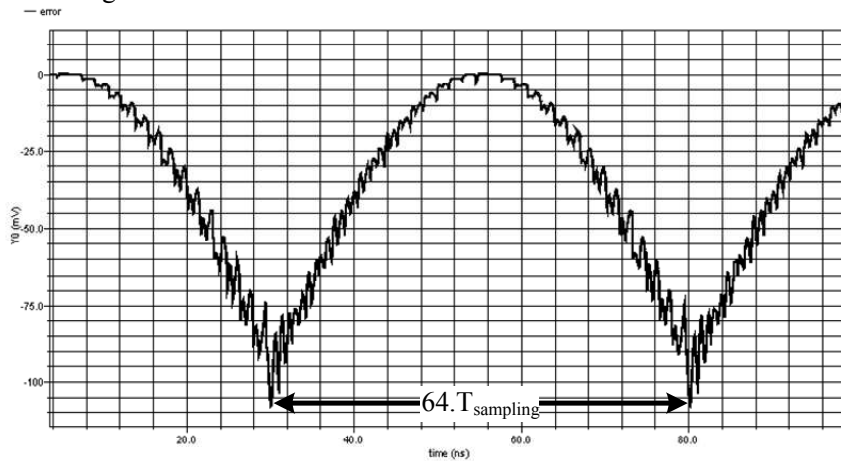


Figure 4.10: Retro-simulation of a windowed zero-signal

### 4.1.3 SASP applications measurements

Measurements were performed on *SASPEPA*. It aims at validating the whole SASP and its applications.

The sampling frequency is  $f_{\text{sampling}} = 320\text{MHz}$ . A sinewave is the input RF signal. The corresponding voltage sample must be different from 0V and others equal to 0V. But, it is noticed that output is saturated and all the voltage sample has the same value of 65mV (Fig. 4.11). It is due to the divergence of the signal processing and no more amplitude information is carried out by the FFT. Given the voltage sample corresponding to the input frequency, output must be around 400mV whereas it is 65mV. The idea is to eliminate saturation by measuring only AC signals at SASP output. Small variations are observed on the corresponding voltage sample. It is discussed on the next item.

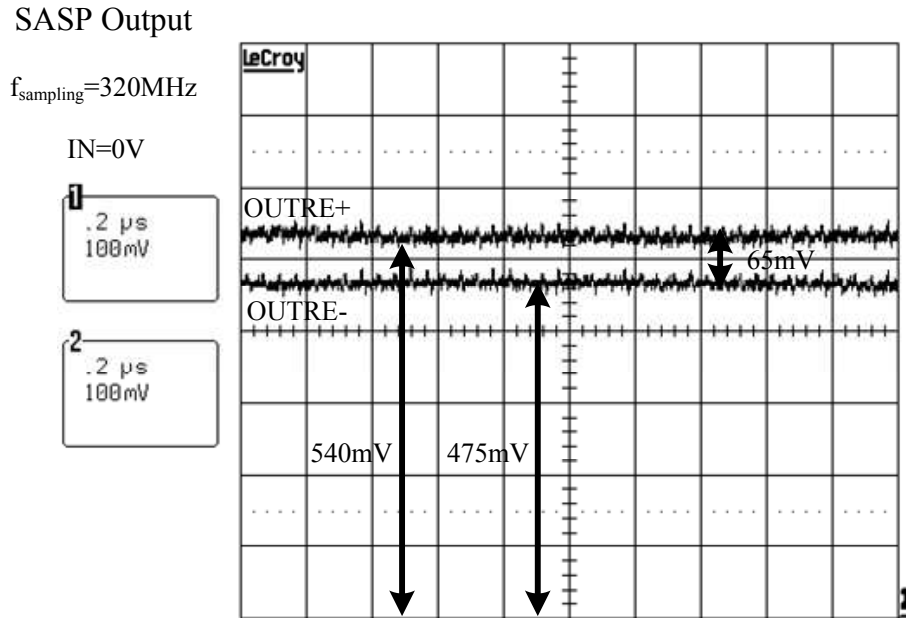


Figure 4.11: SASP output saturation

#### 4.1.3.1 Frequency shifting

The sampling frequency is  $f_{\text{sampling}} = 320\text{MHz}$ . A sinewave is the input RF signal. Its frequency is  $f_{\text{in}} = 150.01\text{MHz}$ . The only voltage sample containing the desired band is selected. It is the 15<sup>th</sup>. As the signal frequency is not an entire number of the sampling frequency, the output of the SASP displays a signal with a frequency of  $f_{\text{out}} = f_{\text{in}} - \frac{n_{\text{sample}} \cdot f_{\text{sampling}}}{64} = \Delta f = 10\text{kHz}$  where  $n_{\text{sample}} = 30$  (Fig. 4.12). Figure 4.13 depicts the shifted signal with a measured  $f_{\text{out}} = 10\text{kHz}$ .



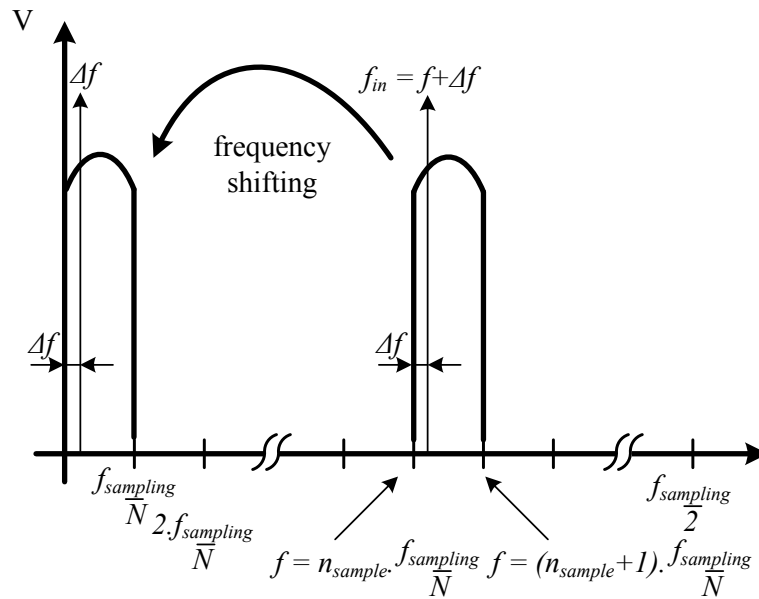


Figure 4.12: Principle of frequency Shifting

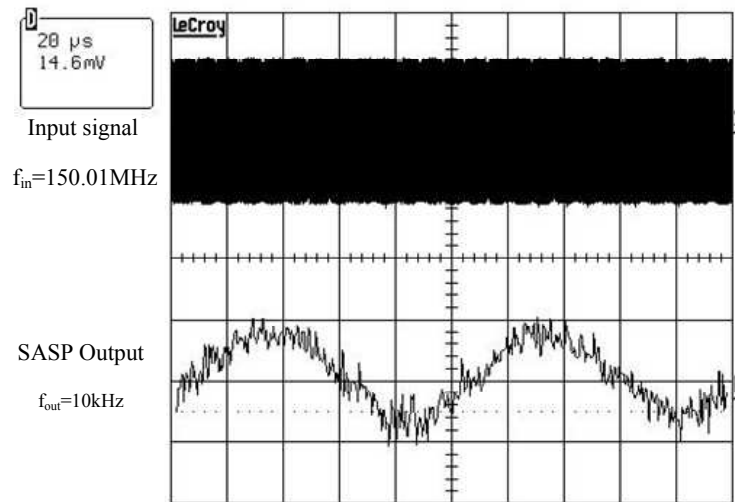


Figure 4.13: Measure of frequency Shifting

The example depicted in section 3.3.4.1 is measured and exhibited in fig. 4.14. A sinewave equal to  $f_{in} = 156.25 MHz$  is sent. In this case,  $n_{sample} = 31$  gives  $f_{out} = \Delta f = 1.25 MHz$ . This measurement shows that the feasibility of a frequency shift in baseband is achieved. The SASP here operates as a wide-band reconfigurable mixer.

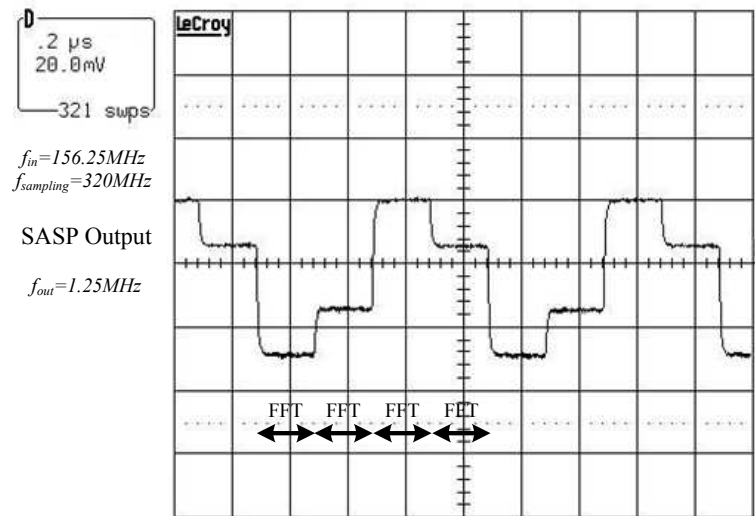


Figure 4.14: Measure of a non-entire frequency sinewave processed by 64-point SASP

#### 4.1.3.2 Frequency Modulation

Measurements are proceeded in order to validate Frequency Modulation (FM). Figure 4.15 presents a FM defined by a frequency deviation of  $f_{\text{deviation}} = 1\text{kHz}$ . The sampling frequency for the measurements is  $f_{\text{sampling}} = 320\text{MHz}$ . A frequency shift of  $160\text{MHz}$  is achieved in baseband.  $f_{\text{out}}$  varies a frequency range from 0 to  $2\text{kHz}$ . It is consistent with  $f_{\text{deviation}}$ .

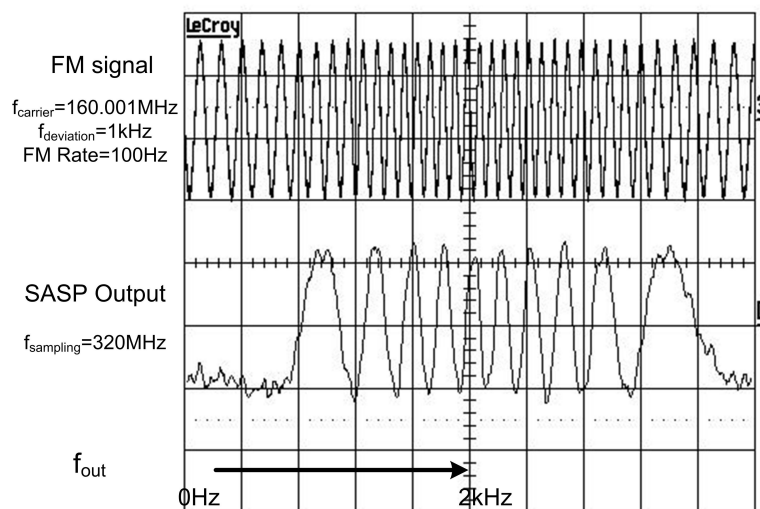


Figure 4.15: Frequency shift of FM signal

### 4.1.3.3 BPSK Modulation

Measurements went on digital modulations. They are the best example of the possibilities offered by the SASP. Figure 4.16 presents a BPSK modulated signal processed by the SASP. The sampling frequency is  $f_{\text{sampling}} = 320\text{MHz}$ . The input signal is given by  $f_{\text{in}} = 160\text{MHz}$  and a bit rate of  $1\text{Mbps}$ . FFT timing is scaled exactly on the bit rate, i.e.  $5\text{MHz}$  which is an entire number of the bit rate. 5 successive FFT are able to process one bit. The principle of frequency demodulation is proven.

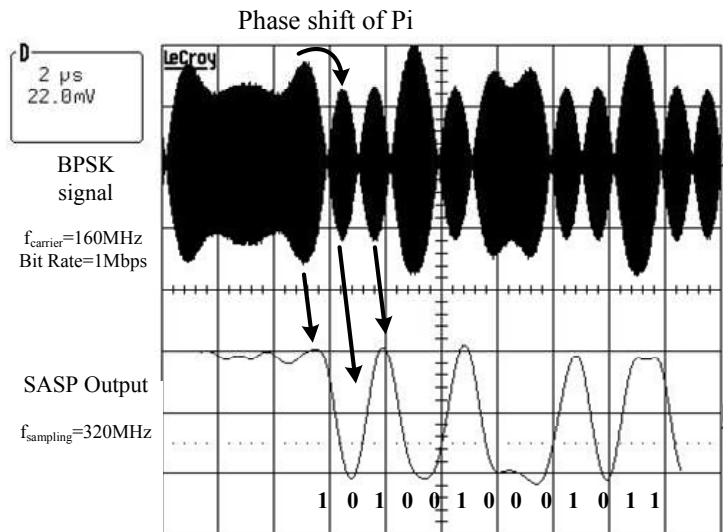


Figure 4.16: BPSK modulation

### 4.1.3.4 FSK Modulation

Assuming a 2-level Frequency Shift Keying (FSK) modulation, bits are encoded at 2 different frequencies. Here, the input signal is given by  $f_{\text{carrier}} = 160.001\text{MHz}$ ,  $f_1 = 160.002\text{MHz}$ ,  $f_2 = 160\text{MHz}$  and a bit rate of  $1\text{ksp}$ . Such characteristics are chosen for a matter of simplicity. FFT timing is scaled exactly on the bit rate, i.e.  $5\text{MHz}$  which is an entire number of the bit rate. 5000 successive FFT process one bit. The SASP shifted in baseband encoded bits. Figure 4.17 depicts how the SASP removes RF carrier and recover encoded bits. '0' is a DC signal and '1' is a  $2\text{kHz}$ -signal.

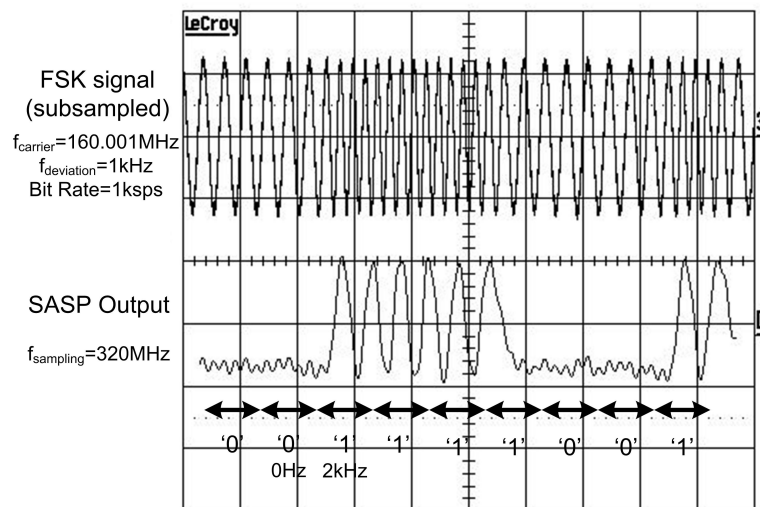


Figure 4.17: FSK modulation

#### 4.1.3.5 ASK Modulation

Assuming an Amplitude Shift Keying (ASK), either the signal is the RF signal or a zero-signal to encode bits. Fig. 4.18 depicts the example with a 160MHz-carrier 1kHz-data rate ASK signal. The SASP only displays a square signal representing the signal envelope. Hence, the output frequency of the SASP is dramatically lowered and all the constraints of ADC and DSP are fully relaxed. More generally, Amplitude Modulated (AM) signals can be demodulated. It paves the way to any demodulation such as for QAM ones.

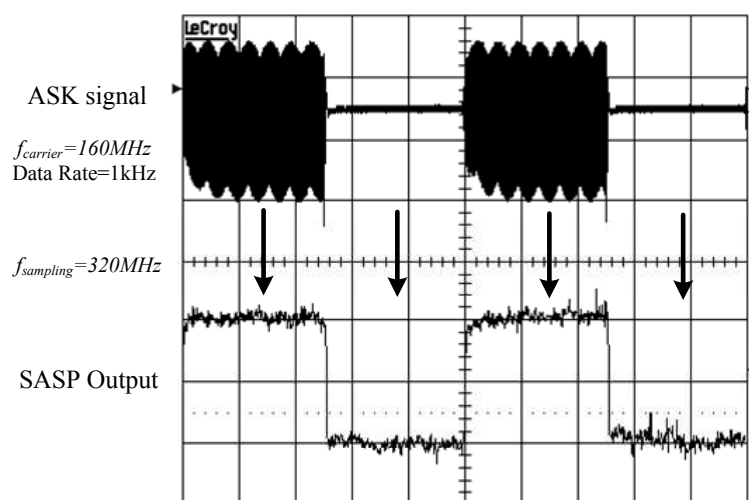


Figure 4.18: ASK modulation

#### 4.1.4 SASPEPA Characteristics

Measurements are performed to extract characteristics of SASPEPA.

- Output amplitude is measured with  $f_{in} = \frac{f_{sampling}}{2}$ .  $f_{sampling}$  is swept from 0 to 800MHz. An optimal range is observed between 100MHz and 400MHz.
- $f_{sampling} = 320MHz$  is chosen.  $f_{in}$  is swept from 0 to 160MHz. Output amplitude is measured for each corresponding  $n_{sample}$ . Only  $n_{sample} \in [8;31]$  are said to be correct whereas others have a too low output frequency.

Power consumption is measured for each stage and part of the circuit. Each power supply is separated. Current crossing power supplies are exhibited in table 4.3. The supply voltage is 1.4V to compensate the resistivity and the voltage drop of access power lines. Measurements are consistent with simulations.

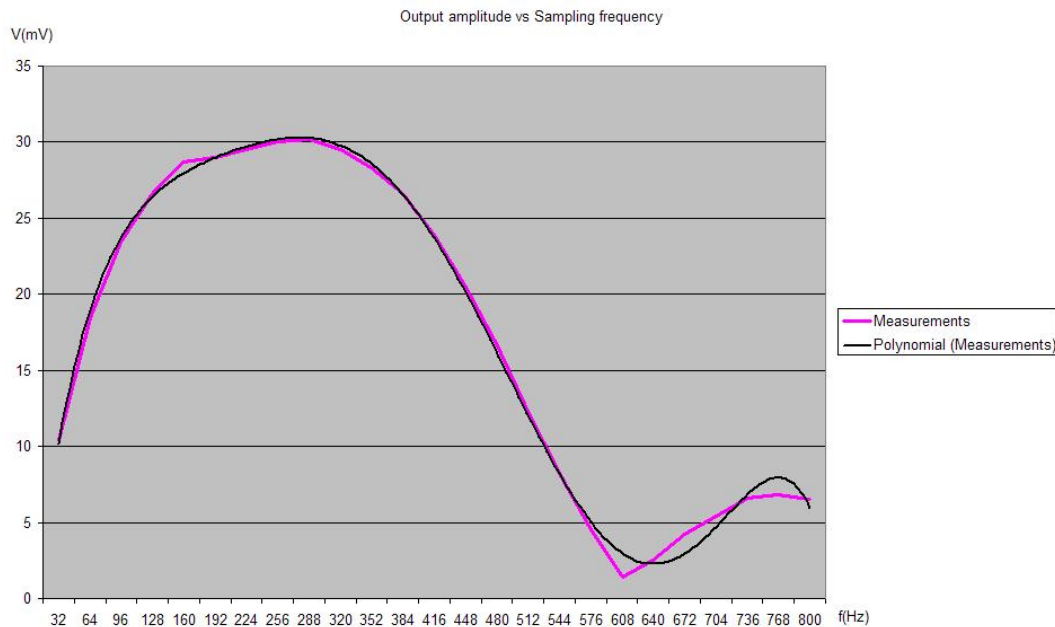


Figure 4.19: Output amplitude vs  $f_{sampling}$

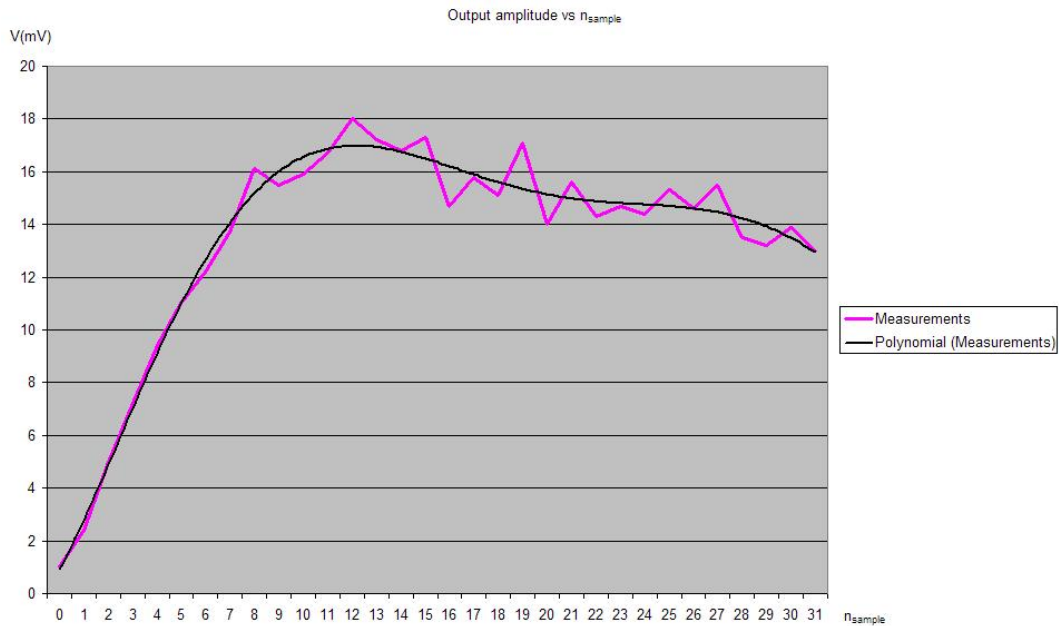
Figure 4.20: Output amplitude vs  $n_{\text{sample}}$ 

Table 4.3: Power Consumption under 1.4V

Power Supply	Measured current
$VDD_{IN}$	29mA
$VDD_{POLA}$	4mA
$VDD_{NUM}$	31mA
$VDD_1$	129mA
$VDD_2$	73mA
$VDD_3$	109mA
Total	375mA

Table 4.4 summarizes *SASPEPA* characteristics. This chip enabled to validate the SASP feasibility and to exhibit technical improvements to be carried. The following section describes the roadmap of an industrial product.

Table 4.4: SASPEPA Characteristics

Operating input frequency range	50MHz-200MHz
Operating sampling frequency	100MHz-400MHz
Power Supply	1.4V
Current Consumption	375mA
Current Consumed per sample	5.86mA per sample
Die Area	1.44mm <sup>2</sup>
Active surface	0.3mm <sup>2</sup>

## 4.2 An open window to RF applications - Achievement of SASP65K

The ultimate SASP is a 65536-sample SASP. Let us call it SASP65K. 65536 samples are required to address any telecommunication standards (Tab. 2.4). In order to remain attractive, SASP65K must be low power (under 500mW) at high frequencies (above 10GHz).

If SASP65K is implemented following SASP64 design flow and technology, it would consume 360A for a constant current consumption per sample ! System improvements are possible to dramatically lower the power consumption and remain under 500mW. Propositions are given in this section.

### 4.2.1 Schematic perspectives

SASP65K counts 8 pipelined stage to process the FFT. It can be considered that a constant power consumption of 100mA per stage is feasible, which leads to a whole current consumption of 800mA. Two main blocks are to be improved:

- Delay Line: each delay cell has buffers. Charge transfer could be done without buffer. Consequently, current consumption of delay lines are reduced. Routing matter is also important. The first stage will count delay lines with 49152 delay cells. The placement is not automatic and must be done manually or partly manually. This is done while keeping in mind that each cell is governed by digital signal generated in a different part of the circuit. Consequently, buses with 65536 digital signals must address the delay line. This is far too complex. Two architectures are proposed to overcome design issues:
  - The first one is inspired by DRAM architecture (Fig. 4.21). The delay line is seen as an array. Digital signals control delay cells by row and column addressing. If the

two digital signals *row* and *column* are both at '1', the switch is open and a voltage sample can be loaded. Every switch is controlled by a unique combination of *row* and *column* signals. Digital part complexity issue is solved.

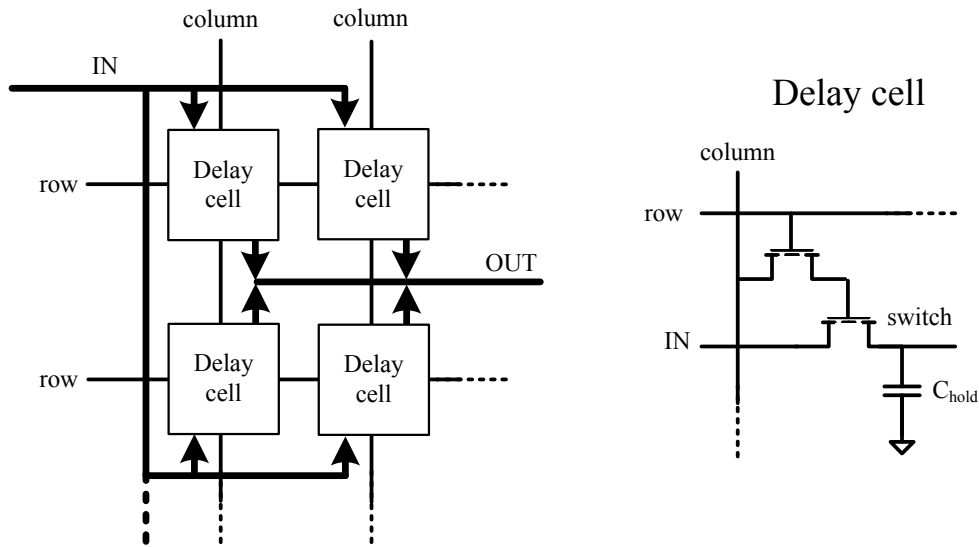


Figure 4.21: First proposed delay line architecture

- The second one is a recursive one (Fig. 4.22). Basic block is designed as small unit. It generates its own digital signal with master signals to address the delay cells of the blocks. This solution allows to build recursively a whole delay line. The example of figure 4.21 is given. The first basic block has 4 delay cells. A block with 16 delay cells is generated with 4 basic blocks. This *divide and conquer* approach facilitates the design work by focusing only on the design of basic blocks.

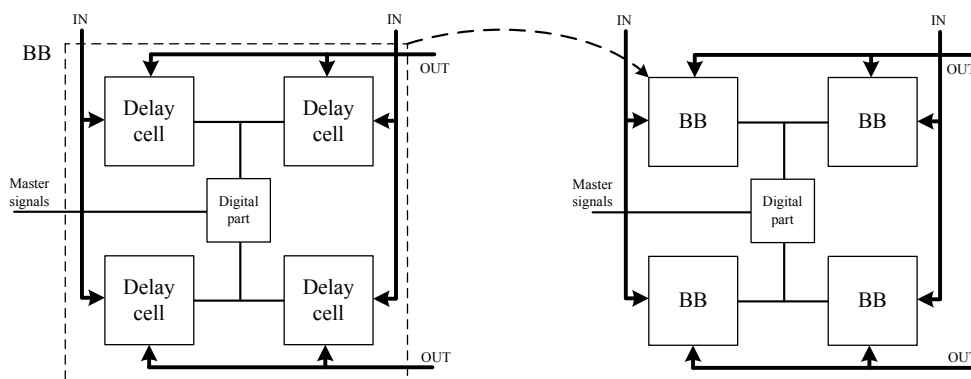


Figure 4.22: Second proposed delay line architecture



- **Weighting Unit:** the power consumption is an exponential function of the number of coefficients. SASP65K requires the generation of 16384 coefficients. Approximations can be done on coefficients, and a minimum accuracy can be tolerated [50]. Only a limited number of transistor could be used. An optimized algorithm could select a given number of transistor to weight every sample. This algorithm is based on possible combinations of transistors. Table 4.5 exposes every possible combination and the generated coefficient with 4 transistors. It is demonstrated that 4 transistors enable the generation of 16 coefficients. 16 transistors generate  $2^{16}$  coefficients. SASP65K requires 16384 coefficients but if a resolution of  $10^{-3}$  on every coefficient is chosen, only 1000 coefficients and 10 transistors are sufficient. Current consumption is estimated at 50mA.

Table 4.5: Weighting Unit Coefficients extension

Switch Network Configuration				Coefficients
$S_1$	$S_2$	$S_3$	$S_4$	applied
$\frac{W_1}{L} = 2.1$	$\frac{W_2}{L} = 5.3$	$\frac{W_3}{L} = 7.8$	$\frac{W_4}{L} = 9.3$	$\cos(\frac{2\pi \cdot nk}{N})$
DC	DC	DC	DC	<b>0.000</b>
DC	DC	DC	IN	<b>0.382</b>
DC	DC	IN	DC	0.325
DC	DC	IN	IN	<b>0.707</b>
DC	IN	DC	DC	0.217
DC	IN	DC	IN	0.599
DC	IN	IN	DC	0.541
DC	IN	IN	IN	<b>0.924</b>
IN	DC	DC	DC	0.076
IN	DC	DC	IN	0.459
IN	DC	IN	DC	0.401
IN	DC	IN	IN	0.783
IN	IN	DC	DC	0.293
IN	IN	DC	IN	0.675
IN	IN	IN	DC	0.618
IN	IN	IN	IN	<b>1.000</b>

Matrix Unit is re-used for each stage. The estimated power consumption for a SASP65K designed in 65nm CMOS technology is thus lowered to some 600mA (Tab. 4.6).

Table 4.6: Estimated Power Consumption

Stages	Delay Line	Weighting Unit	Matrix Unit	Total
Stage 1 to 8	few mA	50mA	22.8mA	75mA

### 4.2.2 Technology issues

The die area is a big issue, as voltage samples are stored in capacitors. Those capacitors occupy major part of the die area. The number of capacitors of SASP65K is 212988. Using 65nm CMOS technology, the die area of the capacitors is  $1.537mm^2$ . The whole die area is estimated at  $1.5cm^2$ . A technological roadmap let us estimate to reduce the die area in 22nm CMOS technology. Varactor capacitors can be used. Trench capacitors are also a solution. Indeed, their area is low compared to planar or stacked capacitors.

*SASPEPA* has show frequency limitations to 400MHz. It occurred because of RF lines and slew rates induced by non-optimized schematics. Migration to a more advanced technology will lower parasitic capacitors and improve density of integration.

### 4.2.3 Signal processing accuracy

As a FFT is performed, accuracy of the spectrum is directly dependent on  $f_{\text{sampling}}$ .  $f_{\text{sampling}}$  must not vary. Otherwise, it will introduce a modulation at SASP output.  $T_p$  must be constant whatever the first instant of sampling. Special care in sampler and frequency generator design has to be carried out.

### 4.2.4 Real-Time error correction

Analog operations are very dependent on process variation and environmental context. Every circuit can see its properties modified. A Buid-In Self-Test (BIST) could be helpful to determine if the circuit is in a range of operability or not. Analog compensation can be envisaged to correct errors as well as digital post-SASP corrections are feasible. For instance, a pipelined stage of the DFT can be controled by the DSP. Its biasing will be a parameter of accuracy improvement. Part of error correction can be done in digital thanks to a self-characterization of the SASP. Defined sequences can be auto-generated and a self correction code can be created and compared to its expected answer delivered by the SASP (Fig. 4.23).

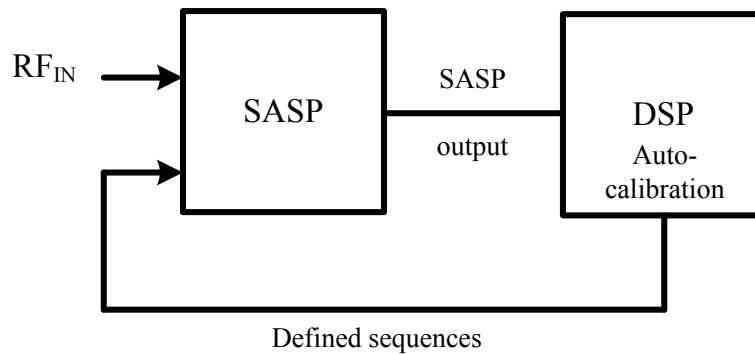


Figure 4.23: SASP system auto-calibration

### 4.3 Conclusion

Chapter 4 has presented the measurements of the chips *SASPEPA* and *LUCATESTA*. They have validated the feasibility of analog operations and confirmed the SASP ability to fulfill Software Radio constraints by removing any carrier of RF signal and relaxing any technological bottlenecks of the ADC and DSP. Part of the schematics to be corrected have been identified. An industrial implementation is envisaged by an overall circuit optimization to dramatically reduce the power consumption and reach RF frequencies. Solutions have been proposed such as a smart combination of transistors in the case of WU or a new architecture of delay lines to relax constraints on design. A technical roadmap to a SASP65K has been paved to achieve the first Software Radio chip.

# Conclusion

Mobile terminals are the place of multimedia convergence. For that reason telecommunication industry claims for a single-chip radio architecture. The concept of Software Radio proposes to shift as close as possible toward the antenna the A/D conversion. RF signals are converted in digital and a single chip would be able to handle any RF standard by software. But, the ADC and DSP bottleneck prevent from realizing such a system because of the very high required power consumption. This thesis focuses on the design of an analog device which addresses the Software Radio concept by relaxing all the constraints on the ADC and DSP.

A Sampled Analog Signal Processor (SASP) is proposed to perform a frequency translation with voltage samples. The frequency translation is done by processing a FFT at RF frequencies. As the desired RF envelope to be demodulated varies slower than its carrier frequency, it is possible by selecting only the RF envelope to shift to baseband any RF signal in a given range. The circuit to be designed is wideband and can address any standard in a 0 to 5GHz frequency range. The FFT displays a spectrum with voltage sample. Only the few voltage samples of the RF envelope among thousands are selected and converted to digital. This selection dramatically reduces the SASP output frequency and consequently the ADC input frequency. It has been demonstrated that frequencies are lowered from GHz frequencies to MHz frequencies.

Despite the principle seems easy to implement, many design issues are solved. The algorithm was implemented in order to use any resources 100% of the time and to optimize the power consumption while keeping in mind the accuracy of the analog signal processing. Behavioral simulations confirmed the feasibility of the proposed implementation. Two applications are emphasized by the SASP principle: the frequency demodulation and the concurrent reception. They highlight ADC and DSP constraints relaxation by shifting part of demodulation signal processing into the analog domain.

The circuit was designed in 65nm CMOS technology from STMicroelectronics. A demonstrator of the SASP was sent to foundry and measured. It handles 64 voltage samples to perform

a FFT on RF signals. The power consumption is 389mW with a power supply of 1.4V. The maximal frequency of operation is 400MHz. Frequency demodulation was proven with PSK, FSK, QAM modulation types. Design failures were identified and solutions are proposed to overcome further technological bottleneck. The main one is still the power consumption. This characteristic should be dramatically lowered thanks to an overall circuit optimization, which was not brought into play in this demonstrator.

Measurements have paved the way to an industrial implementation which is expected to process 65536 samples at 10GHz at low power consumption (under 500mW). Attractive SASP65K characteristics were exposed. A technological roadmap is drawn for a full realization of SASP65K in 22nm CMOS technology. This circuit would be the very first analog chip to fulfill Software Radio concept for mobile terminals.

# Publications

## Patents

- [ P1 ] F. Rivet, JB. Begueret, D. Belot, Y. Deval, H. Lapuyade and T. Taris, “Method and device for the analog processing of a radio signal for a radio frequency receiver,” in *US Patent number US20080299936*.
- [ P2 ] F. Rivet, JB. Begueret, D. Belot, Y. Deval, H. Lapuyade and T. Taris, “Procédé et dispositif de traitement analogique d’un signal radio pour récepteur radiofréquence (Analog Signal Processing),” in *French Patent number FR-0755443*.
- [ P3 ] F. Rivet, JB. Begueret, D. Belot, Y. Deval and D. Dallet, “Procédé et dispositif électronique de décalage fréquentiel d’un signal analogique, en particulier pour la téléphonie mobile (Analog Signal Frequency Translation)” in *French Patent number FR-0755441*.

## Journals

- [ R1 ] F. Rivet, Y. Deval, D. Dallet, JB. Begueret, P. Cathelin and D. Belot, “A Disruptive Receiver Architecture Dedicated To Software Defined Radio,” in *IEEE Transactions on Circuits and Systems (TCAS-II)*, vol. 55, number 4, pp. 344–348, April 2008.

## International Conferences

- [ C1 ] F. Rivet, Y. Deval, D. Dallet, JB. Begueret, P. Cathelin and D. Belot, “The first experimental demonstration of a SASP-based full Software Radio Receiver,” in *Proc. IEEE Radio Frequency Integrated Circuits Symposium (RFIC’09)*, Boston, USA, pp. 601–604, June 2009.
- [ C2 ] F. Rivet, Y. Deval, D. Dallet, JB. Begueret, P. Cathelin and D. Belot, “From Software-Defined to Software Radio: Analog Signal Processor Features,” in *Proc. IEEE Radio and Wireless Symposium (RWS’09)*, San Diego, USA, January 2009.

- [ C3 ] F. Rivet, Y. Deval, D. Dallet, JB. Begueret and D. Belot, “A 65nm CMOS RF Front End Dedicated To Software Radio,” in *Software Defined Radio Forum (SDR Forum 2008)*, Washington D.C., USA, October 26-30, 2008.
- [ C4 ] F. Rivet, Y. Deval, D. Dallet, JB. Begueret, P. Cathelin and D. Belot, “65nm CMOS Circuit Design of a Sampled Analog Signal Processor dedicated to RF Applications,” in *Proc. IEEE North East Workshop in Circuits and Systems (NEWCAS’08)*, Montreal, Quebec, pp. 233–236, June 2008.
- [ C5 ] F. Rivet, Y. Deval, D. Dallet, JB. Begueret and D. Belot, “A 65nm CMOS Analog Processor for Mobile Terminals Software Radio Front End,” in *IEEE South Symposium on Microelectronics (SIM’08)*, Bento Goncalves, Brazil, May 5-9, 2008.
- [ C6 ] F. Rivet, Y. Deval, D. Dallet, JB. Begueret and D. Belot, “A Universal RF Architecture Based on Signal Analog Sampling Dedicated to Software Defined Radio,” in *Proc. IEEE North East Workshop in Circuits and Systems (NEWCAS’07)*, Montreal, Quebec, August 5-8 2007.
- [ C7 ] F. Rivet, Y. Deval, D. Dallet, JB. Begueret and D. Belot, “A Software-Defined Radio based on Sampled Analog Signal Processing Dedicated to Digital Modulations,” in *IEEE PhD Research in MicroElectronics (PRIME’07)*, Bordeaux, France, July 2-5, 2007.
- [ C8 ] F. Rivet, Y. Deval, D. Dallet, JB. Begueret and D. Belot, “A Disruptive Software-Defined Radio Receiver Architecture Based on Sampled Analog Signal Processing,” in *Proc. IEEE Radio Frequency Integrated Circuits Symposium (RFIC’07)*, Honolulu, USA, pp. 197–200, June 2007.

### National Conferences

- [ N1 ] F. Rivet, Y. Deval, D. Dallet, JB. Begueret, P. Cathelin and D. Belot, “Vers la radio logicielle intégrale: le SASP, un processeur analogique du signal en temps discret,” in *16èmes Journées Nationales Microondes (JNM’09)*, Grenoble, France, May 27-29, 2009.
- [ N2 ] F. Rivet, Y. Deval, D. Dallet, D. Belot and JB. Begueret, “Un Processeur Analogique en technologie 65nm CMOS destiné à la Radio Logicielle pour des Terminaux Mobiles,” in *GDR SOC-SIP*, Paris, France, June 4th, 2008.
- [ N3 ] F. Rivet, Y. Deval, D. Dallet, D. Belot and JB. Begueret, “Un Processeur Analogique de Signaux Radio-Frequences destiné la Radio Logicielle pour des Terminaux Mobiles,” in

---

*Journées Nationales du Réseau des Doctorants en Microélectronique (JNRDM'08)*, Bordeaux, France, May 14-16, 2008.

- [ N4 ] F. Rivet, Y. Deval, D. Dallet, D. Belot and JB. Begueret, “Un Processeur de Traitement du Signal Analogique destiné à la Radio Logicielle,” in *15èmes Journées Nationales Microondes (JNM'07)*, Toulouse, France, May 23-25, 2007.
- [ N5 ] F. Rivet, Y. Deval, D. Dallet, D. Belot and JB. Begueret, “Architecture d’un Récepteur Radio Logicielle à Temps Discrets,” in *Journées Nationales du Réseau des Doctorants en Microélectronique (JNRDM'07)*, Lille, France, May 14-16, 2007.





# Bibliography

- [1] B. Razavi, *Rf Microelectronics*. Prentice Hall, 1997.
- [2] M. Brandolini, P. Rossi, D. Manstretta, and F. Svelto, "Toward multistandard mobile terminals fully integrated receivers requirements and architectures," *IEEE Transactions on Microwave Theory and Techniques*, vol. 53, pp. 1026–12038, March 2005.
- [3] E. McCune, "High-efficiency, multi-mode, multi-band terminal power amplifiers," *IEEE Microwave Magazine*, pp. 44–55, March 2005.
- [4] J. Mitola, "The software radio architecture," *IEEE Communications Magazine*, vol. 33, pp. 26–38, May 1995.
- [5] (2008) SDR Forum. [Online]. Available: [www.sdrforum.org](http://www.sdrforum.org)
- [6] J. Mitola, "Cognitive radio, an integrated agent architecture for software defined radio," Ph.D. dissertation, Royal Institute of Technology, 2000.
- [7] S. Grossman, "Software-defined radio poses major challenges for hardware and software developers," *www.rfdesign.com*, pp. 10–15, June 2005.
- [8] R. Schiphorst, F. Hoeksema, and C. Slump, "The front end of software-defined radio: Possibilities and challenges," in *Proc. Annual CTIT Workshop on Mobile Communications*, 2001.
- [9] R. Walden, "Performance trends for analog to digital converters," *Communications Magazine, IEEE*, vol. 37, no. 2, pp. 96–101, February 1999.
- [10] P. Seen, "Radio logicielle dans les terminaux: quels impacts technologiques ?" *Journées Nationales Microondes*, 2007.
- [11] G. Geelen, E. Paulus, D. Simanjuntak, H. Pastoor, and R. Verlinden, "A 90nm CMOS 1.2V 10b power and speed programmable pipelined ADC with 0.5pJ/conversion-step," *IEEE International Solid State Circuits Conference*, pp. 782–783, February 2006.

- [12] K.-W. Hsueh, Y.-K. Chou, Y.-H. Tu, Y.-F. Chen, Y.-L. Yang, and H.-S. Li, "A 1V 11b 200MS/s pipelined adc with digital background calibration in 65nm CMOS," *IEEE International Solid State Circuits Conference*, pp. 546–547, February 2008.
- [13] L. J. Breems, "A cascaded continuous-time sigma-delta modulator with 67dB dynamic range in 10MHz bandwidth," *IEEE International Solid State Circuits Conference*, pp. 72–73, February 2004.
- [14] R. C. Schuh, P. Eneroth, and P. Karlsson, "Multi-standard mobile terminals," 2002.
- [15] T. Taris, J. B. Begueret, and Y. Deval, "A low voltage current reuse LNA in a 130nm CMOS technology for UWB applications," *IEEE European Microwave Conference*, pp. 1105–1108, October 2007.
- [16] T. Taris, O. El-Gharniti, J. B. Begueret, and E. Kerhervé, "UWB LNAs using LC ladder and transformers for input matching networks," *IEEE International Conference on Electronics Circuits and Systems*, December 2006.
- [17] E. Colin, "Architecture reconfigurable pour la numérisation du signal radio de récepteurs mobiles multistandards," Ph.D. dissertation, Ecole Nationale Supérieure des Télécommunications, 2003.
- [18] C. Rougier, J. Begueret, H. Lapuyade, Y. Deval, and A. Malvasi, "The frequency generation unit: a complex frequency synthesizer for multi-standard smart objects," *IEEE Northeast Workshop on Circuits and Systems*, pp. 277–280, June 2004.
- [19] R. Vaughan, N. Scott, and D. Whit, "The theory of bandpass sampling," *IEEE Transactions on Signal Processing*, vol. 39, pp. 1973–1984, 1991.
- [20] P. Prakasam, M. Kulkarni, X. Chen, S. Hoyos, and B. Sadler, "Emerging technologies in software defined receivers," *IEEE Radio and Wireless Symposium*, vol. 39, pp. 719–722, January 2008.
- [21] V. J. Arkesteijn, E. A. Klumperink, and B. Nauta, "An analogue front-end architecture for software defined radio," in *Proc. IEEE ProRISC*, Veldhoven, Netherlands, November 2002.
- [22] K. Muhammad, D. Leipold, B. Staszewski, C. H. Y.-C. Ho, K. Maggio, C. Fernando, and T. Jung, "A discrete-time bluetooth receiver in a 0.13 $\mu$ m digital CMOS process," *IEEE International Solid State Circuits Conference*, pp. 268–270, 2004.
- [23] A. A. Abidi, "Software-defined radio receiver: Dream to reality," *IEEE Communications Magazine*, pp. 111–118, 2006.

- [24] A. K. Mal and A. S. Dhar, "Analog sampled data architectures for discrete cosine transform," pp. 502–505, October 2003.
- [25] W. Boyle and G. Smith, "Charge Coupled Semiconductors Devices," vol. 49, pp. 587–593, 1970.
- [26] H. Wallinga, "A general model for the frequency response of multiphase charge transfer delay lines," *IEEE Journal of Solid State Circuits*, vol. 14, pp. 653–655, 1979.
- [27] D. D. Buss, "Transversal filtering using Charge-Transfer Devices," *IEEE Journal of Solid-State Circuits*, vol. 8, pp. 138–146, April 1973.
- [28] D. Lampe, "Programmable analog transversal filter," U.S. Patent 4 034 199.
- [29] R. Schreiber, "Passive CCD resonator filters," *IEEE Journal of Solid-State Circuits*, vol. 16, pp. 125–129, June 1981.
- [30] H. Klar, "Passive CCD resonators," *IEEE Journal of Solid-State Circuits*, vol. 16, pp. 130–135, June 1981.
- [31] P. Bosshart, "An integrated analog correlator using Charge-Coupled Devices," *IEEE International Solid-State Circuits Conference*, vol. 19, pp. 198–199, February 1976.
- [32] G. Mayer, "The Chirp z-Transform - a CCD implementation," *RCA Review*, vol. 36, pp. 759–773, December 1975.
- [33] J. M. Speiser, "Discrete Fourier Transform system using the dual Chirp-z Transform," U.S. Patent 4 282 579.
- [34] R. Brodersen, H. Fu, R. Frye, and D. Buss, "A 500-point fourier transform using Charge Coupled Devices," *IEEE International Solid State Circuits Conference*, vol. 18, pp. 144–145, 1975.
- [35] R. C. Pettengill, P. W. Bosshart, M. de Wit, and C. R. Hewes, "A monolithic 512-point Chirp-z Transform processor," *IEEE International Solid State Circuits Conference*, vol. 22, pp. 68–69, 1979.
- [36] W. L. Eversole, D. J. Mayer, P. W. Bosshart, M. de Wit, C. R. Hewes, and D. D. Buss, "A completely integrated thirty-two-point Chirp-z Transform," *IEEE Journal of Solid State Circuits*, vol. 13, pp. 822–831, 1978.
- [37] J. W. Cooley and J. W. Tukey, "An algorithm for the machin calculation of complex fourier series," *Math. Comput.*, vol. 19, pp. 297–301, 1965.

- [38] H. Groginsky and G. Works, "A pipeline Fast Fourier Transform," *IEEE Trans. on Computers*, vol. 11, pp. 1015–1019, Nov. 1970.
- [39] G. Works, "Real time Fourier Transformation apparatus," in *US Patent number 3.816.729*.
- [40] A. A. Mariano, "Mixed simulations and design of a wideband continuous-time bandpass delta-sigma converter dedicated to software defined radio applications," Ph.D. dissertation, University of Bordeaux 1, 2008.
- [41] R. Darraji, R. Barrak, C. Rebai, A. Ghazel, Y. Deval, and F. Ghannouchi, "Track and hold circuit design and implementation in 65nm CMOS technology for RF subsampling receivers," *IEEE International Conference on Electronics, Circuits and Systems*, September 2008.
- [42] S. Rapuano and F. Harris, "An introduction to FFT and time domain windows," *Part 11 in a series of tutorials in instrumentation and measurement*, pp. 32–44, December 2007.
- [43] F. Harris, "On the use of windows for harmonic analysis with the Discrete Fourier Transform," *Proceedings of the IEEE*, vol. 66, January 1978.
- [44] E. Swartzlander, W. Young, and S. Joseph, "A radix-4 delay commutator for Fast Fourier Transform processor implementation," *IEEE Journal of Solid-State Circuits*, vol. 19, pp. 702–709, Oct. 1984.
- [45] A. El-Khashab, "Modular pipeline Fast Fourier Transform algorithm," Ph.D. dissertation, University of Texas at Austin, 2003.
- [46] B. Gold and T. Bially, "Parallelism in Fast Fourier Transform hardware," *IEEE Trans. on Audio and Electroacoustics*, vol. A-21, 1, pp. 5–16, Feb. 1973.
- [47] Y.-N. Chang and K. Parhi, "An efficient pipelined architecture," *IEEE Trans. on Circuits and Systems*, vol. 50, pp. 322–325, 2003.
- [48] E. Monastra and J. Huah, "Pipelined Fast Fourier Transform processor," in *US Patent number 5.038.311*.
- [49] C. Lacy, "Design of a programmable Switched-Capacitor analog FIR filter," Ph.D. dissertation, University of Toronto, 1999.
- [50] K. Boyle, P. Mercier, N. Sadeghi, V. Gaudet, C. Schlegel, C. Winstead, and M. Kashyap, "Design and implementation of an all-analog Fast-Fourier Transform processor," *IEEE Midwest Symposium on Circuits and Systems*, vol. 66, pp. 1532–1535, August 2007.



## **Design of a Radio Frequency Front-End Receiver dedicated to Software-Radio for Mobile Terminals.**

**Abstract:** Many technological bottlenecks prevent from realizing a Software Radio (SR) mobile terminal. The old way of building radio architectures is over due to the numerous communication standards a single handheld terminal have to address nowadays. This thesis exposes a disruptive SR receiver: a Sampled Analog Signal Processor (SASP) is designed and brought into play to perform downconversion and channel presort. It processes analog voltage samples in order to recover in baseband any RF signal emitted from 0 to 5GHz. An analog Fast Fourier Transform achieves both frequency shifting and filtering. A prototype using 65nm CMOS technology from STMicroelectronics is here presented and measured.

## **Contribution à l'étude et à la réalisation d'un frontal radiofréquence analogique en temps discrets pour la radio-logicielle intégrale**

**Résumé:** Le concept de Radio Logicielle propose d'intégrer en un seul circuit un émetteur / récepteur RF capable d'émettre et de recevoir n'importe quel signal RF. Cependant, ce concept doit affronter des contraintes technologiques dans le cas des terminaux mobiles. La contrainte principale est la consommation de puissance du terminal. En effet, la conversion analogique numérique qui est la clé de ce système en est aussi le principal verrou technique. Cette thèse présente une architecture de récepteur en rupture avec les architectures classiques afin de surmonter le problème de la conversion analogique numérique. Il s'agit d'un processeur analogique de traitement du signal dédié à la Radio Logicielle intégrale dans la gamme de fréquence 0 à 5GHz. Sa conception et les mesures d'un prototype sont présentées.