A Wave Pipeline-Based WCDMA Multipath Searcher for a High Speed Operation

Pierre-Andre Laporte

A Thesis

In

The Department

Of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Applied Science in Electrical Engineering

Concordia University

Montreal, Quebec, Canada

November 2007

Your file Votre référence
ISBN: 978-0-494-40887-2
Our file Notre référence
ISBN: 978-0-494-40887-2

ABSTRACT

A Wave Pipeline-Based WCDMA Multipath Searcher for a High Speed Operation

Pierre-Andre Laporte

The multiplexing technique of the Wideband-Code Division Multiple Access (WCDMA) is widely applied in the third generation (3G) of cellular systems. The WCDMA uses scrambling codes to differentiate the mobile terminals. In a channel, multipaths may occur when the transmitted signal is reflected from objects in the receiver's environment, so that multiple copies of the signal arrive at the antenna at different moments. Thus, a wideband signal may suffer frequency selective fading due to the multipath propagations. A Rake receiver is often used to combine the energies received on different paths, and a multipath searcher is needed to identify the multipath components and their associated delays. Correlating some shifted versions of the scrambling code with an incoming signal results in energy peaks at the multipath locations, when the locally generated scrambling sequence is aligned with the scrambling sequence of the incoming signal. A path acquisition in such a process requires a speed of millions of Multiply-Accumulate (MAC) cycles per second. The performances of the multipath searcher are mainly determined by the resolution and the acquisition time, which are often limited by the operation speed of the hardware resources.

This thesis presents the design of a multipath searcher with a high resolution and a short acquisition time. The design consists of two aspects. The first aspect is of the searching algorithm. It is based on a double-dwell algorithm and a verification stage is introduced to lower the rate of false alarms. The second aspect in the design is the circuit of the searcher. This circuit is expected to operate at the chip rate of 3.84 MHz and the search period is chosen to be equal to the time interval of 5 slots, which requires a high operation speed of the computation units employed in the circuit. Moreover, in order to

reduce the circuit complexity, only one Complex Multiplier-Accumulator (CMAC), instead of several ones in many existing searcher circuits, is employed to perform all the computation tasks without extending the search period, which make the computation time in the circuit more critical. Aiming at this challenge of the high speed requirement, a structure of the CMAC cell is designed with the technique of the wave pipeline, which permits the signal propagation through the circuit stages without constraints of clocks. For a good use of this technique, the circuit blocks are made to have equalized delay, by means of pass transistor logic cells, and by keeping such a delay short, the total computation time of the CMAC can be made within the required time limit of the searching.

A complete circuit of the CMAC has been developed. It has two versions, with the Normal Process Complementary Pass Logic (NPCPL) and the Complementary Pass-Logic Transmission-Gates (CPL-TG), respectively. The structures of the arithmetic units have been chosen carefully so that the fan-in/fan-out constraints of the NPCPL and the CPL-TG logics are taken into consideration. The results of the simulation with a 0.18 μm models have shown that this wave pipelined CMAC can process four inputs of 8 bits at a rate of 830 Mb/s. In order to evaluate the effectiveness of the searching algorithm, a Matlab simulation of the searcher circuit has been conducted. It has been observed that the proposed multipath searcher can lead to low probabilities of misdetection and false alarm for the test cases recommended by the 3rd Generation Partnership Project (3GPP) standard. A test chip of the CMAC circuit has been fabricated in a CMOS 0.18 μm technology process. The circuit is currently under test.

Acknoledgments

This project is part of the collaboration between Axiocom inc., Concordia University and Université du Québec à Trois-Rivières. It is a continuation of the work initialized by Frédéric Morin and Martin Vidal from Université du Québec à Trois-Rivières in 1999.

I would like to express my sincere gratitude to my supervisors, Dr. Chunyan Wang and Dr. Daniel Massicotte (UQTR) for their guidance and support thoughout the course of this work.

A special thanks to Ted Obuchowicz for his patience and help with the CAD tools. I would like to convey my sincere appreciation to Messaoud Ahmed-Ouameur (Axiocom) for his time and helpfull advices.

I would like to thank the Fond Quebecois de Recherche sur la Nature et les Technologies (FQRNT) and the Microelectronics Strategic Alliance of Quebec (ReSMIQ) for their financial supports, as well as CMC Microsystems for the fabrication services.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

3G	3rd Generation of cellular networks.
3GPP	3rd Generation Partnership Project.
4G	4th Generation of cellular networks.
ALU	Arithmetic Logic Unit.
ASIC	Application Specific Intregrated Circuit.
AWGN	Additive White Gaussian Noise.
CDMA	Code Division Multiple Access.
CLB	Combinational Logic Block.
CMAC	Complex Multiplier-Accumulator.
CMOS	Complementary Metal-Oxide Semiconductor.
CPICH	Common Pilot Channel; used for channel estimation.
CPL-TG	Complementary Pass-Logic Transmission-Gate.
CSA	Carry-Save Adder.
DRC	Design Rule Checking.
DRSCMOS	Dual-Rail Static CMOS.
DSP	Digital Signal Processor.
DUT	Device Under Test.
FDMA	Frequency Division Multiple Access.
FPGA	Field Programmable Gate Array.
IC	Integrated Circuit.

LTE	Long Term Evolution of the third generation.
LVS	Layout Versus Schematic.
MAC	Multipliy-Accumulate.
MAT	Mean Acquisition Time of the multipath searcher.
MIMO	Multiple-inputs Multiple-outputs.
MOS	Metal–Oxide Semiconductor.
MOSFET	Metal–Oxide Semiconductor Field-Effect Transistor.
MSB	Most Significant Bit .
NMOS	N-channel MOSFET.
NPCPL	Normal Process Complementary Pass Transistor Logic.
OVSF	Orthogonal Variable Spreading Factor.
PE	Processing Element.
PMOS	P-channel MOSFET.
PN	Pseudo-Noise.
PTL	Pass Transmission Logic.
QoS	Quality of Service.
QPSK	Quadrature Phase-Shift Keying.
RAM	Random-Access Memory.
RCA	Ripple-Carry Adder.
SoC	System on Chip.
SNR	Signal-to-Noise ratio.
TDMA	Time Division Multiple Access.
VLSI	Very Large Scale Integration.

WCDMA	Wideband Code Division Multiple Access.
WTGL	Wave pipelined Transmission Gate Logic.

List of Symbols

$\alpha_l(t)$	Time varying complex number attenuation of the l^{th} path.
β_f	Scaling factor for temperatures above nominal.
β_S	Scaling factor for temperatures below nominal.
c	Speed of light.
$C_d[n]$	Vector of the candidate delays.
$c_{(k)}[\ell;n]$	ℓ^{th} element of the spreading sequence for the n^{th} symbol of the k^{th} user.
$\overline{c}[\ell;n]$	ℓ^{th} element of the spreading sequence for the n^{th} pilot symbol.
$\overline{\mathbf{c}}[n]$	Sampled pilot scrambling code matrix.
$\overline{d}(t;n)$	Spreading waveform of the n^{th} pilot symbol $s_{(k)}[n]$.
$D_d[n']$	Vector of the detected delays.
$d_{(k)}(t;n)$	Spreading waveform of the n^{th} symbol for the k^{th} user.
$D_{MAX,i}$	Largest propagation delay of the actual wave.
$D_{MIN,i+1}$	Shortest propagation delay of the next wave.
D_R	Propagation delay of a register.
Δ_{Skew}	Controlled clock skew in a wave pipeline circuit.
ΔT	Uncertainty area or search period of the multipath searcher (in μsec).
E_n	Noise energy.

$\mathbf{E}[n]$	Matrix of energies resulting from squaring and summing the I-Q components of the observation matrix $\mathbf{R}'[n]$.
$\varphi(t)$	Chip waveform of unit energy, also called pulse shaping filter. Used to minimize the inter-symbol interferences caused by the channel.
f_c	Carrier frequency.
f_d	Doppler frequency.
f_{max}	Maximal frequency of a circuit.
F_s	Sampling frequency of the analog-to-digital converter in a CDMA receiver.
G_k	Gain of the k^{th} user in the downlink transmission.
G_p	Gain of the pilot in the downlink transmission.
γ	Element in the interval [0; χ-1].
$h(t,\tau)$	Time varying impulse response of the channel.
I	In-phase component of a complex baseband signal.
$\mathbf{i}[n]$	Matrix of interference-plus-noise.
K	Number of mobile stations covered by one base station in a CDMA system.
L	Starting point of the correlation window. Corresponds to a phase shift of the scrambling code.
M	Cyclic period of the scrambling code, equal to 38 400 chips in downlink transmissions.
N'	Length of the correlation.
N_c	Spreading factor.

N_p	Number of paths in a fading channel.
N_{unc}	Number of uncertainties (or offsets) to be tested by the multipath searcher.
$n(t)$	Complex valued additive white gaussian noise.
N_w	Number of waves in a wave pipelined circuit.
Ω	Number of CMACs in the first dwell of the multipath searcher.
Ω_1	Number of CMACs in the second dwell of the multipath searcher.
P_D	Probability of detection of the valid paths.
P_{fa}	Probability of false alarm, i.e. declaring a valid path where none exists.
P_{miss}	Probability of misdetection of a valid path.
$p(t)$	Time varying pilot baseband transmission.
χ	Oversampling factor of the analog-to-digital converter in a CDMA receiver.
q	Element in the interval [0 N'-1].
Q	Quadrature-phase component of a complex baseband signal.
$\mathbf{r}[n]$	Sample vector sequence of $r(t)$ after the analog-to-digital converter.
$\mathbf{R}[n]$	Matrix of correlation.
$\mathbf{R}'[n]$	Matrix of observation.
$r(t)$	Received signal at the mobile terminal.
$s_{(k)}[n]$	n^{th} symbol of the k^{th} user, i.e. a binary data.
$\bar{s}[n]$	n^{th} pilot symbol.
T	Symbol interval.
T_c	Chip interval; equal to $\frac{1}{3.84\ MHz} \approx 260$ nsec.

T_{CLK}	Clock period of a digital circuit.
T_H	Hold time of a register.
TH1	First threshold of a double dwell algorithm.
TH2	Second threshold of a double dwell algorithm.
T_L	Sampling period in a wave pipeline circuit.
$\tau_l(t)$	Time varying temporal delay of the l^{th} path
$\tau_l'(t)$	Estimated time varying temporal delay of the l^{th} path
T_{res}	Time resolution of a multipath searcher.
T_S	Setup time of a register.
v	Speed of a mobile terminal.
W	Correlation window size, it corresponds to the search period in chips.
$x_k(t)$	Time varying baseband transmission of the k^{th} user.
$y(t)$	Transmitted signal from the base station in a CDMA system.

1 Introduction

Wideband Code Division Multiple Access (WCDMA) is a core technology of the third generation (3G) of cellular networks. In a CDMA system, every mobile station (referred to as a *user* in communication literature) is transmitting/receiving data at all time and its signal is spread over the entire bandwidth. Figure 1.1 highlights the differences between the various multiplexing techniques used in the different cellular generations. In the figure, the time and frequency resources allocated to one user are represented by a color stripe. The CDMA allows for a more efficient utilization of the spectrum in comparison with the Frequency Division Multiple Access (FDMA) used in the first generation and the Time Division Multiple Access (TDMA) used in the second generation of cellular systems. Moreover, it has a capacity of supporting some high-speed data transmissions such as video, e-mail and web browsing. The third generation systems are standardized by the 3rd Generation Partnership Project (3GPP) in an effort to develop a globally applicable communication protocol.

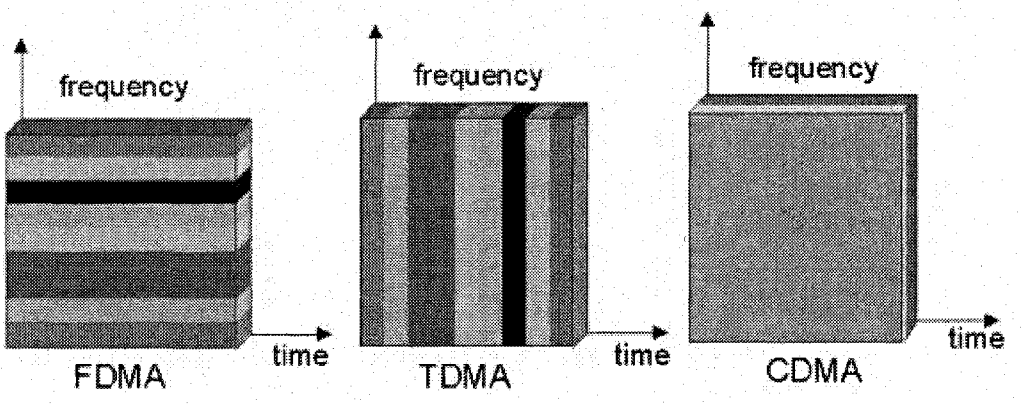


Figure 1.1 Multiplexing techniques used in the different cellular generations.

The differentiation between users in a CDMA system is done by coding the data with pseudo-noise (PN) sequences specific to one user. The PN sequences, also called *signatures* or *scrambling codes*, have low cross-correlation and auto-correlation properties to improve the resistance against interferences [1].

In the channel, the wideband signal experiences multipath propagations due to reflections of the signal on different obstacles as shown in Figure 1.2. A multipath propagation with N_p paths can be represented by the time-varying impulse response $h(t,\tau)$ defined as,

$$h(t,\tau)=\sum_{l=1}^{N_p}\alpha_l(t)\delta(t-\tau_l(t)) \qquad (1.1)$$

where $\alpha_l(t)$ represents the time varying complex number attenuation, i.e. the *fading coefficient*, and $\tau_l(t)$ is the time varying delay of the l^{th} path. Multipath propagations can lead to interferences because different replicas of the same signal may be added destructively at the receiver. However, in a CDMA system, the performances are improved by combining the energies received on different paths by means of a *Rake* receiver [2]. The general scheme of a Rake is illustrated in Figure 1.3. In this type of receiver, some *fingers* consisting of correlators are used to detect the signal received on different paths. Thereafter, the contributions of the different fingers are combined using a weighted summation with respect to the fading coefficients.

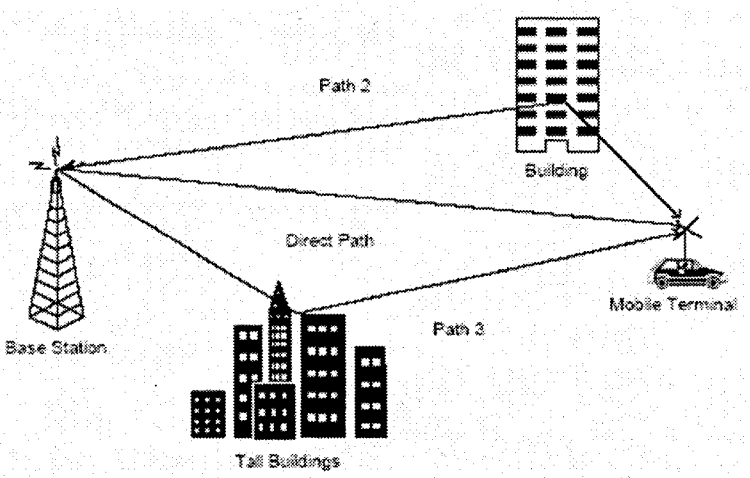


Figure 1.2 Multipath occurrences.

In general, one Rake finger is assigned to every path. *Channel estimation* is the operation of identifying the delays of the multipath components as well as their fading coefficients in the channel. It is a critical issue that cannot be overlooked because it has a great influence on the performance of the receiver. This estimation is usually composed of two steps. In the first step, the multipath searcher identifies the potential paths and their associated delays. Once the searcher has found the delays of the paths, it initializes the offsets of the scrambling code generators in the Rake as shown in figure 1.3 to time-align the locally generated PN sequences with each identified path. In the second step, the channel estimator evaluates the complex attenuations of the selected paths.

Some very accurate delay estimates are required to guaranty the Quality of Service (QoS) at the reception. Missing potential paths or assigning incorrect paths to the Rake fingers will result in an increase of the interference level after combining the Rake fingers. Moreover, any path misalignment can introduce a lot of noise due to the low auto-correlation property of the scrambling codes. The consequences of wrong path detection can be in many aspects, such as information loss and the need for retransmission, higher transmission power requirements and decreased system capacity.

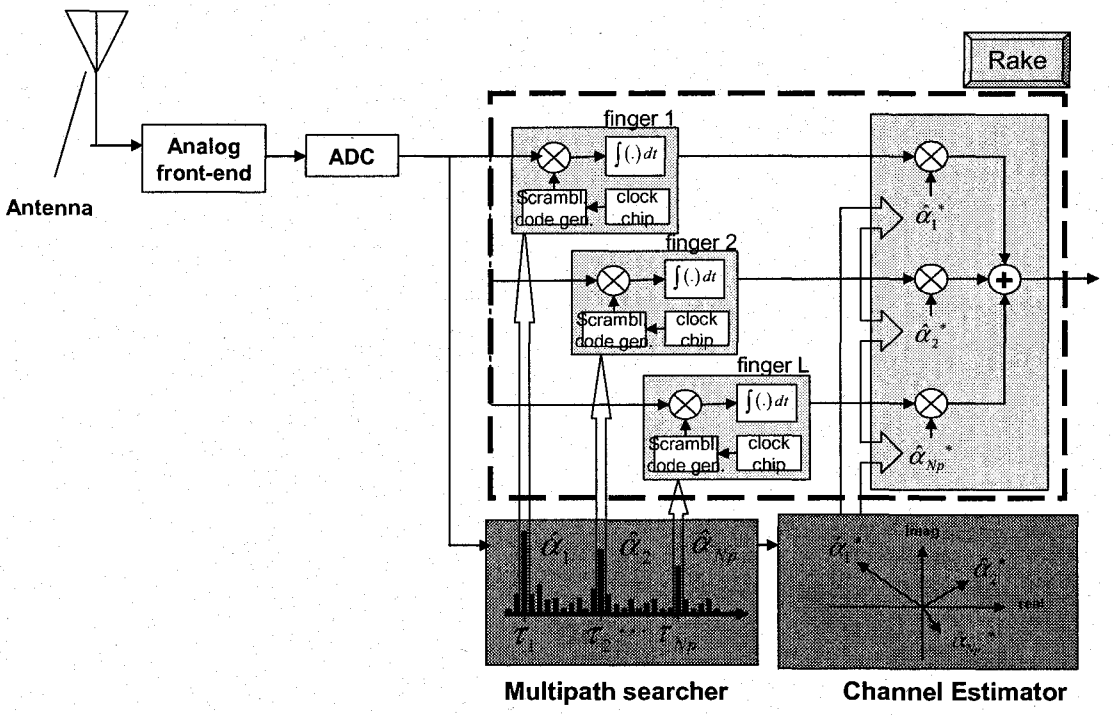


Figure 1.3 Schematic of a Rake receiver [2].

The function of multipath searcher is done by correlating some shifted versions of the scrambling code with the incoming signal, which results in energy peaks at the multipath locations, i.e. when the locally generated PN sequence is aligned with the PN sequence of the incoming signal. The basic operations are in fact the multiplication and the addition, performed in most cases by Multiply-Accumulate (MAC) cells. The speed of the MAC cells employed in a path searcher is one of the critical element in the performance metric of the system.

1.1 Objectives and Motivations

A high performance multipath searcher is needed in 3G mobile receivers for the following reasons: (i) the performance of the Rake receiver highly depends on the quality of the channel estimation and (ii) the path searcher is the first block of the channel estimation chain, influencing all of its aspects.

Basically, the performance metrics of a multipath searcher can be summarized as follows:

1) The time resolution of the searcher must be small to ensure a good alignment of the Rake fingers with each multipath component.

2) The acquisition time must be short so that the delay profile can be updated quickly for good performances, even over fast fading channels, i.e. when the mobile terminal is moving fast.

3) Maximize the probability of detection of a valid multipath (P_D) and minimize the probability of declaring a valid multipath where none exist, i.e. false alarm (P_{fa}).

4) Offer good system performance and low hardware complexity.

The performances of the multipath searcher are mainly determined by the resolution and the acquisition time, which are often limited by the operation speed of the hardware resources. This thesis aims at providing a multipath searcher with a high resolution and a short acquisition time. The implementation of the multipath searcher presented in this thesis can result in increased speed requirements compared to conventional implementations [3]-[4]. In order to keep the hardware architecture simple, an implementation based on high-speed processing is proposed.

The target of this work is to develop a high-speed CMAC that performs the computations of the multipath searcher. For decades, pipelining has been used in the industry to improve the throughput of digital circuits. However, the synchronous pipeline technique (referred to as *conventional pipeline* in this work) has some drawbacks, such as the propagation delays of the clock, i.e. clock skew, and the increased power consumption caused by the clock tree and its associated registers.

In this project, pipelining is needed to optimize the speed of the arithmetic units. In a wave pipeline circuit, many different inputs, also called *waves*, are processed simultaneously in the Combinational Logic Block (CLB) without interfering with each other. In other words, subsequent inputs can be introduced into the circuit at time intervals that are shorter than the propagation delay of the circuit [5]. This is achieved by equalizing the path delays in the circuit. The wave pipeline method allows for synchronizing digital circuits without introducing a complex clock distribution tree. Therefore, the area and power consumption are improved in comparison with the conventional pipeline method.

To achieve the main objective of improving the performance of the multipath searcher without increasing the hardware complexity, the work of this thesis aims at the following measures:

1) The hardware implementation of the multipath searcher based on high-speed processing to improve the resolution and the acquisition time,

2) The modification of a multipath searcher algorithm in order to minimize the false alarm rate,

3) The design of a high-speed CMAC using the wave pipeline method,

4) The investigation of low-level issues related to the design of standard cells and their impact on the performances of wave pipelined circuits.

1.2 Scope and Organization of the Thesis

This work deals with different aspects of the structure in CDMA systems. Firstly, different algorithms of path searcher need to be investigated. An effective method should be developped to have better detections with a low acquisition time and a high resolution. In the aspects of the circuit design, the work of this thesis is related to different levels. It is important to determine the architecture of the computation circuit. Pipeline structures are considered. Moreover, aiming at a high speed operation, the work has also an emphasis on the cell-level design, including that of physical structure in silicon.

The thesis is organized as follow. In Chapter 2, some background about the functionality of multipath searchers and some of the widely used schemes for their design and implementation are presented. Then, existing approaches for the design of wave pipeline circuits will be described along with a discussion on their advantages and drawbacks.

Chapter 3 will give details about the design steps for different levels of abstraction. In a first step, the multipath searcher algorithm will be presented and its speed requirements will be determined. Then, the design of a high speed CMAC using the wave pipeline method will be described. The choice of the logic family for wave pipelining will be justified and the architectures of the adder, multiplier and CMAC will be chosen to comply with the precision requirements of the system model while meeting the fan-in/fan-out constraints of the standard cells.

Chapter 4 will present the performance evaluation of the different levels of abstraction. The developed logic gates will be compared in terms of their delay profile and power consumption. Then, the wave pipelined CMAC will be compared with existing designs in term of the functionality, area, frequency, supply voltage and implementation technology. In a last step, the performances of the proposed multipath searcher will be analyzed against existing implementations in terms of hardware cost and accuracy.

Finally, Chapter 5 will conclude this work by summarizing the design approaches and the performance evaluation. Some suggestions for future investigations will also be proposed.

2 Background

In this chapter, the mathematical model of the CDMA system will be first introduced along with a presentation of the existing work on the multipath searcher. Then, existing hardware implementations of high performance arithmetic units will be reviewed. Finally, some background material about the wave pipeline technique will be presented.

2.1 System Model

The Third Generation Partnership Project is widely accepted as a standard for the third generation of cellular systems and it uses WCDMA as a multiplexing technique. As opposed to other multiple access methods such as frequency division multiplex or time division multiplex, the differentiation between users in a WCDMA system is done by means of scrambling codes.

2.1.1 WCDMA Downlink Transmission Model

In WCDMA systems, the scrambling operation is achieved by multiplying the binary data with a PN sequence of a faster rate. The user's binary data must be over-sampled N_c times in order to match the rate of the PN sequence and the new sampling period corresponds to the fraction of a *symbol*, i.e. binary data, commonly called a *chip* [6]. The over-sampling factor N_c, known as the *spreading factor*, can take any value as specified by the equation

$$N_c = 2^i, i = 2,3,...,9. \tag{2.1}$$

The variable length of the spreading factor allows for many data rates due to the variety of the services offered (voice, video, Internet access, etc) in the third generation of cellular systems. High speed transmissions such as video are assigned a short code

length (eg. $N_c = 4$), while low bit-rate communications such as voice or text messaging have a larger spreading factor.

The downlink scrambling codes of the 3rd generation are made from Gold sequences of a cyclic period $M = 38\,400$ chips, i.e. the outputs of some shift registers are combined with XOR gates and feedback loops to create some PN sequences as illustrated in Figure 2.1. These sequences have low auto-correlation and cross-correlation properties to facilitate the data extraction of one particular user. As shown in the figure, the PN sequences have some in-phase and quadrature-phase (I-Q) components because of the quadrature phase shift keying (QPSK) modulation. The role of the multipath searcher is to provide the initial states (offsets) of these generators on the receiver side.

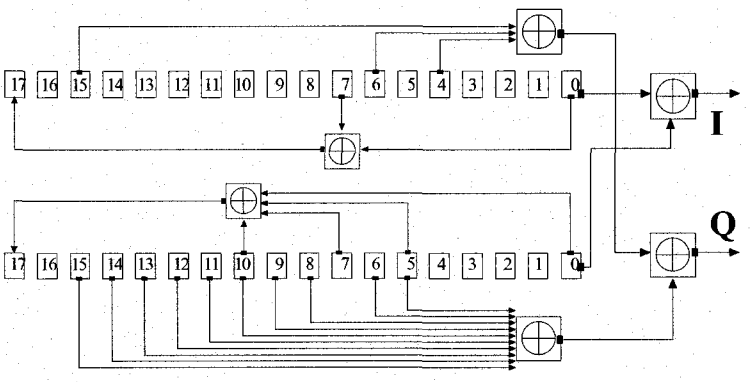


Figure 2.1 Downlink scrambling code generator [6]. Every shift register is represented by a numbered square. The data flow of the shift registers is from 0 to 17. The XOR gates are represented by summation signs.

Consider a forward link from a base station to K mobile stations. Figure 2.2 illustrates the scrambling process of the data of the k^{th} user. In the figure, $s_{(k)}[n] \in S$ ($s_{(k)}[n]$ is a symbol in the alphabet $S = \{\pm 1/\sqrt{2} \pm j/\sqrt{2}\}$ because of QPSK modulation) is the n^{th} symbol, $d_{(k)}(t,n)$ is the spreading waveform of the n^{th} symbol and $x_k(t)$ is the time varying baseband transmission (stream) of the k^{th} user.

Mathematically, the spreading waveform $d_{(k)}(t,n)$ and the baseband signal $x_k(t)$ of the k^{th} user can be expressed as:

$$d_{(k)}(t,n) = \sum_{\ell=0}^{N_c-1} c_{(k)}[\ell,n]\varphi(t-\ell T_c)\,, \tag{2.2}$$

$$x_{(k)}(t) = \sum_{n=-\infty}^{\infty} s_{(k)}[n] d_{(k)}(t-nT,n), \tag{2.3}$$

where T_c is the chip interval ($T_c = \dfrac{1}{3.84\ \text{MHz}} \approx 260$ nsec); T represents the symbol interval; $c_{(k)}[\ell,n]$ is the ℓ^{th} element of the spreading sequence for the n^{th} symbol of the k^{th} user; and $\varphi(t)$ is the chip waveform of unit energy, also called pulse shaping. Pulse shaping consists in modifying the waveform of the transmitted pulses from a rectangular shape to a sin(x)/x shape by means of a raised-cosine filter [8]. By doing so, the bandwidth of the transmitted signal is reduced and the inter-symbol interferences caused by the channel are minimized.

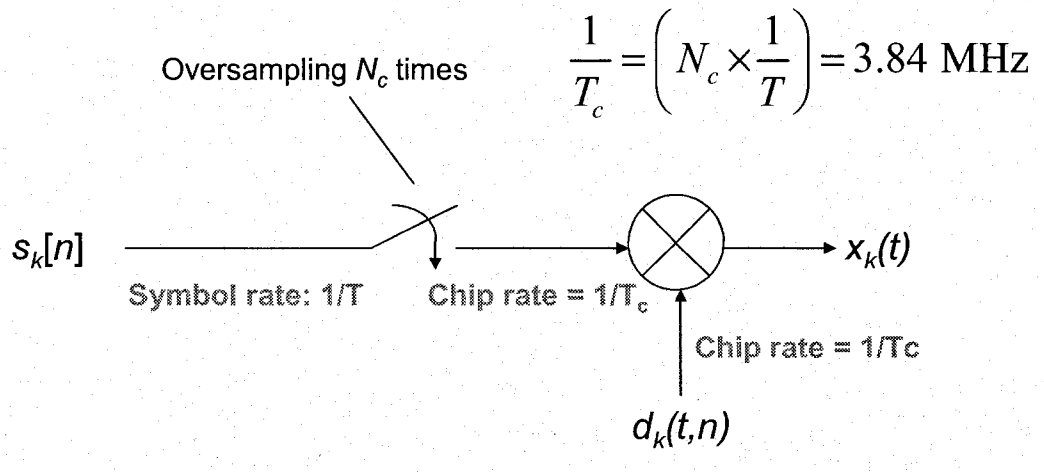
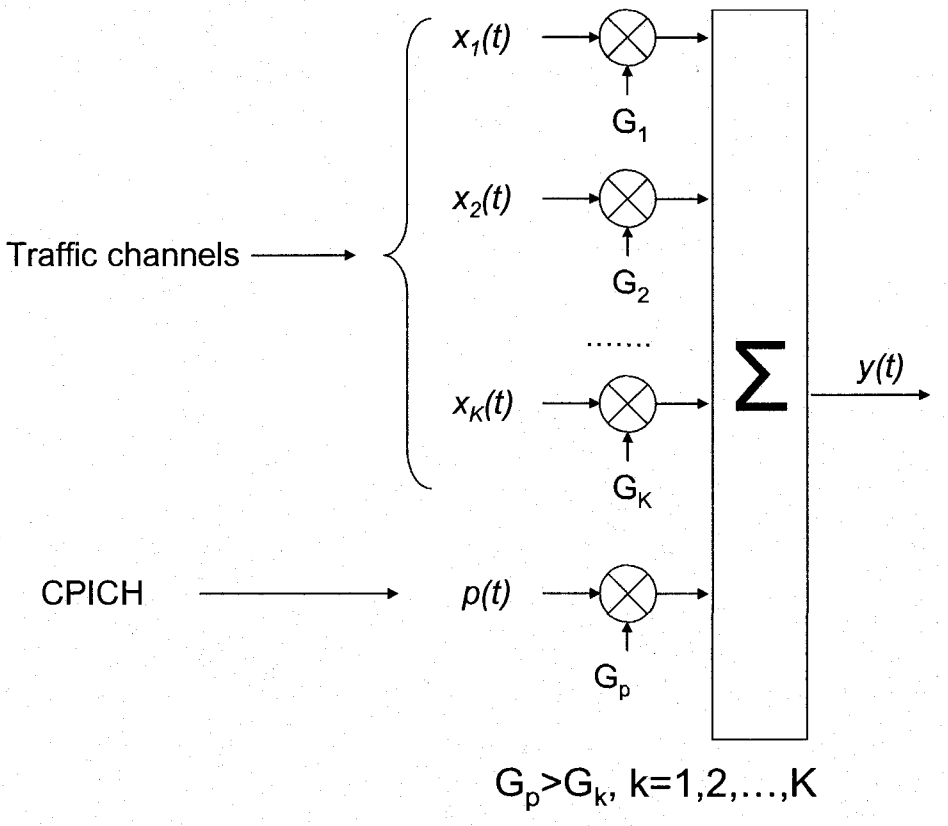


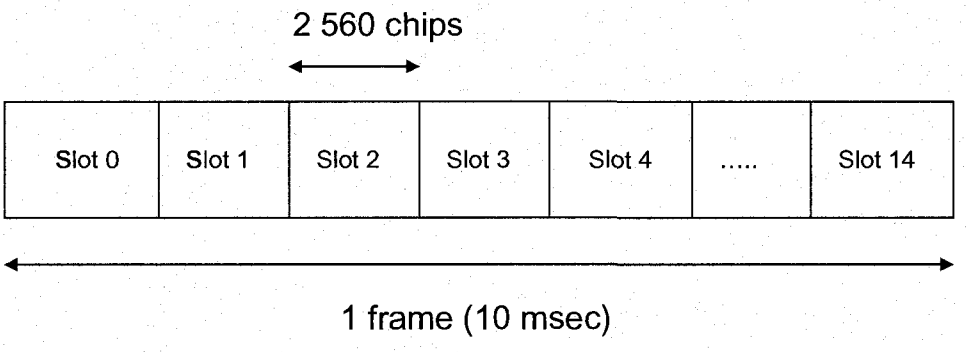
Figure 2.2 Scrambling operation.

Figure 2.3 (a) shows how the baseband streams $x_k(t)$ of the K users are orthogonally multiplexed in the downlink transmission. The different gains, illustrated as G_ks in the figure, are used for power control. For a channel estimation purpose, a Common Pilot Channel (CPICH) is transmitted with higher power than the traffic channels, i.e. the users. The principle of the CPICH is to send a stream of data known by the receiver, such as a

stream of 1's, so that the convolution between the channel's impulse response and the pilot signal can be easily resolved. Figure 2.3 (b) illustrates the baseband frame structure of the different streams $x_k(t)$ as well as $p(t)$. In the figure, it is observed that 1 frame corresponds to a 10 msec transmission and it is composed of 15 slots of 2 560 chips [7].



(a)



(b)

Figure 2.3 a) Downlink multiplexing [6], b) Baseband frame structure [7].

The pilot transmission $p(t)$ can be written as

$$p(t) = \sum_{n=-\infty}^{\infty} \bar{s}[n]\bar{d}(t-nT, n), \tag{2.4}$$

where $\bar{s}[n]$ is the pilot symbol; and $\bar{d}(t,n)$ the spreading waveform of the n^{th} pilot transmission. For simplicity, the pilot symbol are assumed to be fixed, i.e., $\bar{s}[n] = \bar{s}$, and the spreading waveform is $\bar{d}(t,n) = \sum_{\ell=0}^{N_c-1} \bar{c}[\ell,n]\varphi(t-\ell T_c)$, where $\bar{c}[\ell,n]$ is the spreading sequence for the n^{th} pilot transmission. In WCDMA downlink transmissions, the spreading factor N_c of the CPICH is of 256 chips [7].

Therefore, the transmitted signal $y(t)$ can be represented by the following equation

$$y(t) = G_p p(t) + \sum_{k=1}^{K} G_k x_k(t). \tag{2.5}$$

In the channel, the transmitted signal $y(t)$ undergoes a multipath fading channel with N_p paths defined by the time varying impulse response $h(t,\tau)$ defined as,

$$h(t,\tau) = \sum_{l=1}^{N_p} \alpha_l(t)\delta(t-\tau_l(t)), \tag{2.6}$$

where $\alpha_l(t)$ represents the time varying complex number attenuation and $\tau_l(t)$ is the time varying temporal delay of the l^{th} path. In general, $\alpha_l(t)$ can be considered as a random variable. For Rayleigh fading channels $\alpha_l(t)$ is a complex Gaussian random variable.

The rate at which the channel's coefficients vary in time, i.e. *fade*, can be expressed by the Doppler frequency f_d,

$$f_d = v \times f_c / c, \tag{2.7}$$

where v is the speed of the mobile terminal, f_c is the carrier frequency and c is the speed of light. As highlighted in (2.7), the Doppler frequency is directly proportional to the speed of the terminal. In general, we observe a sliding channel where the exact values of the delays, phases and amplitudes change approximately at the Doppler frequency over time. Note that we can also observe some birth and death conditions where the energy of the signal suddenly disappears at some delays and reappears at other delays with a totally different complex attenuation.

The signal vector received at the mobile terminal *r(t)* results from the convolution between the transmitted signal *y(t)* and the channel impulse response *h(t,τ)*,

$$r(t) = \sum_{l=1}^{N_p} \alpha_l(t) y(t) \delta(t - \tau_l(t)) + n(t), \qquad (2.8)$$

where *n(t)* is the complex valued Additive White Gaussian Noise (AWGN) of mean zero and variance σ^2. The noise energy E_n can be written as a function of time for all instant *d*.

$$E_n\left[n(t)n^*(t)\right] = \sigma^2 \delta(t-d). \qquad (2.9)$$

Let $\mathbf{r}[n]$ denote the sample vector sequence of $r(t)$ from analog-to-digital converters following a matched filter for the chip waveform $\varphi(t)$. The sampled matched filter output can be written as $r[\ell] = \int_{-\infty}^{+\infty} r(t-\tau)\varphi(\tau)d\tau\Big|_{t=\ell T_c}$.

2.1.2 Multipath Searcher

Basically, the role of the multipath searcher is to identify the temporal delay $\tau_{(l)}$ of all the paths. Correlating some shifted versions of the sampled pilot scrambling code expressed by the matrix $\bar{\mathbf{c}}[n]$ of size $W \times N'$, where W represents the correlation window size, N' represents the length of the correlation, $q \in [0 \;\; N'-1]$ and L is the starting point of the correlation window,

$$\overline{\mathbf{c}}[n]=\begin{bmatrix} \overline{s}[n]\overline{c}[L;n] & \cdots & \overline{s}[n]\overline{c}[q;n] & \cdots & \overline{s}[n]\overline{c}\left[L+N'-1;n\right] \\ \overline{s}[n]\overline{c}[L+1;n] & \cdots & \overline{s}[n]\overline{c}[q+1;n] & \cdots & \overline{s}[n]\overline{c}\left[L+N';n\right] \\ \vdots & & \vdots & & \vdots \\ \overline{s}[n]\overline{c}[L+W-1;n] & \cdots & \overline{s}[n]\overline{c}[L+q+W-1;n] & \cdots & \overline{s}[n]\overline{c}\left[L+W+N'-2;n\right] \end{bmatrix},$$

with the incoming sampled sequence vector $\mathbf{r}[n]=\left[\mathbf{r}[n+0] \;\cdots\; \mathbf{r}[n+q] \;\cdots\; \mathbf{r}\left[n+N'-1\right]\right]^T$ results in energy peaks at the multipath location. In case where $\ell > M$ ($M = 38\,400$ chips; the cyclic period of the scrambling code sequence) in the term $\overline{c}[\ell,n]$, the index ℓ is wrapped-around by means of modulo function with respect to M.

In WCDMA systems, the analog-to-digital converter samples the signal at a frequency F_s corresponding to χ samples per chip. In high performance receivers, $\chi = 4$, i.e. $F_s = 15.36$ MHz, so that the multipath components can be resolved within a quarter of a chip duration. Therefore, the $N'\times\chi$ matrix of the incoming sampled sequence $\mathbf{r}[n]$ is written as

$$\mathbf{r}[n]=\begin{bmatrix} r[n+0] & \cdots & r[n+q*\chi] & \cdots & r\left[n+(N'-1)*\chi\right] \\ \vdots & & \vdots & & \vdots \\ r[n+\gamma] & \cdots & r[n+\gamma+q*\chi] & \cdots & r\left[n+\gamma+(N'-1)*\chi\right] \\ \vdots & & \vdots & & \vdots \\ r[n+(\chi-1)] & \cdots & r[n+(\chi-1)+q*\chi] & \cdots & r\left[n+(\chi-1)+(N'-1)*\chi\right] \end{bmatrix}^T,$$

where $\gamma \in [0;\ \chi-1]$. The matrix product between sampled sequence $\mathbf{r}[n]$ and the sampled pilot scrambling code $\overline{\mathbf{c}}[n]$ is expressed as

$$\mathbf{R}'[n]=\mathbf{r}[n]\overline{\mathbf{c}}[n], \tag{2.10}$$

where the $W\times\chi$ observation matrix $\mathbf{R}'[n]$ results from the summation of the correlation matrix $\mathbf{R}[n]$ and the interference plus noise matrix $\mathbf{i}[n]$

$$\mathbf{R}'[n]=\mathbf{R}[n]+\mathbf{i}[n]. \tag{2.11}$$

$\mathbf{R}'[n]$ is expressed as follow

$$\mathbf{R}'[n]=\begin{bmatrix} R'[n+0] & \cdots & R'[n+\gamma] & \cdots & R'[n+(\chi-1)] \\ \vdots & & \vdots & & \vdots \\ R'[n+q] & \cdots & R'[n+q+\gamma] & \cdots & R'[n+q+(\chi-1)] \\ \vdots & & \vdots & & \vdots \\ R'[n+(W-1)] & \cdots & R'[n+(W-1)+\gamma] & \cdots & R'[n+(W-1)+(\chi-1)] \end{bmatrix}$$

Since the signal has been converted to a digital baseband, the correlation is usually performed by means of a Complex Multiplier-Accumulator (CMAC) in digital hardware. However, the CMAC only produces the in-phase and quadrature-phase (I-Q) components of the signal. The signal energy is equal to the square of the amplitude and the I-Q components must be squared and summed (see Figure 2.4) in order to have the energy estimates.

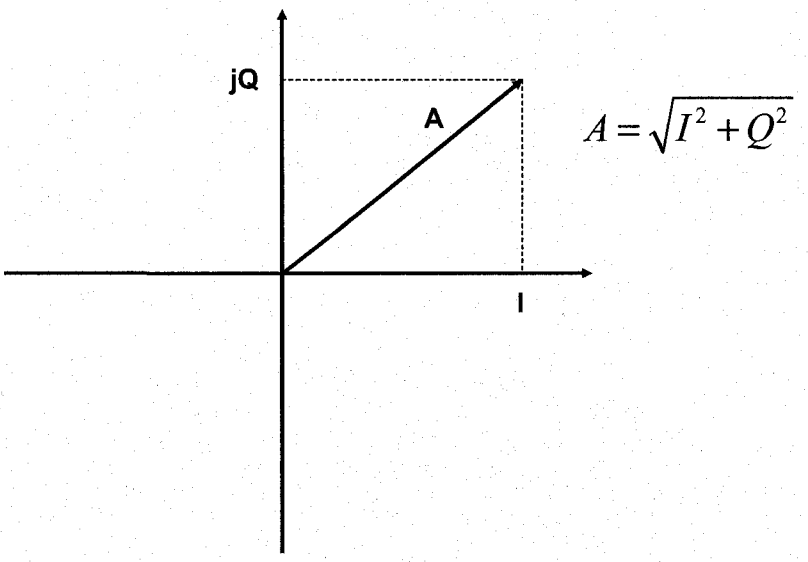


Figure 2.4 Amplitude of a phasor.

Let $\mathbf{E}[n]$ be the $W\times\chi$ matrix of energies that results from squaring and summing the I-Q components of each correlation value in the observation matrix $\mathbf{R}'[n]$

$$\mathbf{E}[n]=\begin{bmatrix} E[n+0] & \cdots & E[n+\gamma] & \cdots & E[n+(\chi-1)] \\ \vdots & & \vdots & & \vdots \\ E[n+q] & \cdots & E[n+q+\gamma] & \cdots & E[n+q+(\chi-1)] \\ \vdots & & \vdots & & \vdots \\ E[n+(W-1)] & \cdots & E[n+(W-1)+\gamma] & \cdots & E[n+(W-1)+(\chi-1)] \end{bmatrix}.$$

The *finger management algorithm* consists in some control logic responsible for identifying the paths with the greatest energies in the matrix $\mathbf{E}[n]$. Once the multipath components have been identified, their estimated delays $\tau'_{(l)}$ can be expressed in seconds by means of the equation

$$\tau'_{(l)} = q_{(l)}T_c + \frac{\gamma_{(l)}}{\chi}T_c. \tag{2.12}$$

Figure 2.5 summarizes the different steps for delay acquisition. In the figure, W CMACs operate in parallel to compute the γth column of the matrix $\mathbf{R}'[n]$. Then, the I-Q components of the correlation results are squared and summed (represented by $|.|^2$ in the figure), and the energy values are stored in the matrix $\mathbf{E}[n]$ by the finger management algorithm. This procedure must be repeated χ times to fill all the columns of the matrix $\mathbf{E}[n]$ and at every time, the starting point γ of the vector $\mathbf{r}[n+\gamma]$ is increased by one sample.

To compensate for some noise, interferences and sudden variations of the channel represented by the matrix $\mathbf{i}[n]$ in (2.11), the correlation window is applied at many different points in time and the energy values are averaged before the strongest paths are identified. This strategy results in more accurate delay estimates.

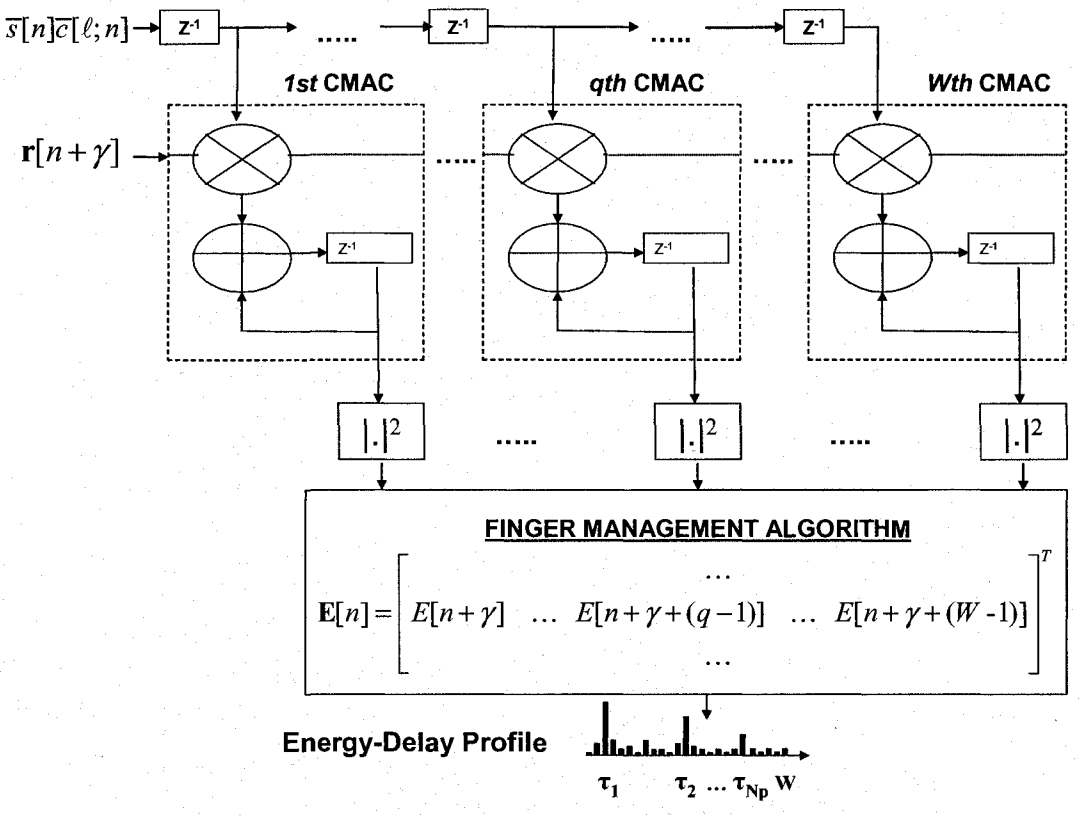


Figure 2.5 Schematic of the coarse acquisition.

The procedure that we have defined in this section is referred to as coarse acquisition and it is the focus of this thesis. In actual receivers, the delay estimates from the coarse acquisition are not accurate enough for data detection because they have been averaged over many correlation windows. Therefore, a second step called delay tracking is performed in order to have some more accurate delay estimates before they are assigned to the Rake fingers. Note that delay tracking will not be discussed any further in this work.

2.1.3 Existing Coarse Acquisition Algorithms

The coarse acquisition phase of the multipath searcher has received a lot of attention in recent years ([4], [9] – [20]). This task must optimize the following parameters:

1) The time resolution of the searcher must be small to ensure a good alignment of the Rake fingers with each multipath component.

2) The Mean Acquisition Time (MAT) must be short so that the delay profile can be updated quickly for good performances, even over fast fading channels (high Doppler frequency).

3) Maximize the probability of detection of a valid multipath (P_D) and minimize the probability of misdetection ($P_{miss} = 1 - P_D$).

4) Minimize the probability of declaring a valid multipath where none exist, i.e. false alarm (P_{fa}).

5) Offer good system performance and low hardware complexity.

It has been shown that path acquisition is a hypothesis testing problem in [8]. The total number of offsets (or *uncertainties*) that the receiver needs to test depends on the temporal uncertainty range of phase delay (ΔT) and the resolution of the process (T_{res}). The expression for the number of uncertainties to test N_{unc} is given by

$$N_{unc} = \Delta T / T_{res} \tag{2.13}$$

For WCDMA systems, the uncertainty area ΔT is of approximately ±20 µsec; the largest delay that can be considered as a path in the 3GPP standard [44]. Considering an oversampling factor $\chi = 4$, the resolution is of ¼ of a chip and $T_{res} = T_c/4$ ($\approx$ 65 nsec). Therefore, the number of uncertainties N_{unc} to be tested is in the range of a few hundreds. It is important to mention that the term N_{unc} is also equal to $W \times \chi$.

In the literature, several search methods for testing the N_{unc} uncertainties are presented:

- The parallel search is treated in [9]-[11],
- The serial search is covered by [12]-[17],

- The hybrid search method is presented in [4] and [18].

In the parallel search, each possible offset N_{unc} is assigned a correlator. The acquisition time of a completely parallel search method is the shortest; the matrix product $\mathbf{r}[n]\overline{\mathbf{c}}[n]$ is computed in N' clock cycles. For a WCDMA receiver with an over-sampling factor $\chi = 4$, the number of uncertainties N_{unc} is approximately of 615 (2.13). Therefore, a complex hardware architecture comprising 615 CMACs would be required to implement a parallel search. From a practical point of view, the MAT of a completely parallel structure is much shorter than what is needed.

In the serial search, a single correlator scans all possible uncertainties resulting in a reduced hardware complexity. However, this strategy has a longer acquisition time compared to the parallel method. The matrix product $\mathbf{r}[n]\overline{\mathbf{c}}[n]$ is completed in $N' \times N_{unc}$ clock cycles.

Finally, the hybrid search combines the features of the serial and parallel searches by using a few correlators in parallel, but not as much as the number of offsets to be tested, to reduce the acquisition time. In the case where Ω CMACs are combined in parallel, the matrix product $\mathbf{r}[n]\overline{\mathbf{c}}[n]$ is performed in $\frac{N' \times N_{unc}}{\Omega}$ clock cycles.

More advanced techniques such as those presented in [19]-[20] have two stages. They are commonly known as *double-dwell* acquisition techniques. In these schemes, if the correlation energy is greater than a first threshold *TH1* (also called the first dwell), a second test with more strict parameters is performed (threshold *TH*2) as illustrated on Figure 2.6 (a). The different thresholds can be determined in function of the signal properties [15], such as the signal-to-noise ratio (SNR). The schematic of a double-dwell algorithm is presented in Figure 2.6 (b). As illustrated on the figure, Ω and Ω_1 CMACs are used in the first and the second dwell respectively. Both stages can be made of a serial, hybrid or parallel search and the values of Ω and Ω_1 are determined as follow:

- $\Omega = 1$ for the serial search,

- $\Omega \in [2; N_{unc} - 1]$ for the hybrid search,
- $\Omega = N_{unc}$ for the parallel search.

At any time n, all the delays with correlation energy greater than the first threshold *TH1* are stored in the vector of the potential candidate $C_d[n]$. Thereafter, the potential candidates are re-correlated at a time n' where (n' modulo M) = (n modulo M) and $n' \neq n$ to take advantage of the random nature of the interfering effects. In case where the energy of the second correlation is greater than the threshold *TH2*, the delay is declared as a valid path and it is stored in the vector of the detected delays $D_d[n']$ for further processing, i.e. delay tracking. In general, high threshold values will result in low probabilities of false alarms and detections; this will in turn increase the MAT. Low threshold values will have high probabilities of detections and of false alarms, and they will also have a large MAT. An optimal threshold value will result in the minimum MAT. In general, double-dwell methods are more complex to implement than single dwell ones. However, they help in reducing the probability of a false alarm considerably.

In [4], a new acquisition strategy combining three pipelined stages is proposed and implemented. The different stages consist of:

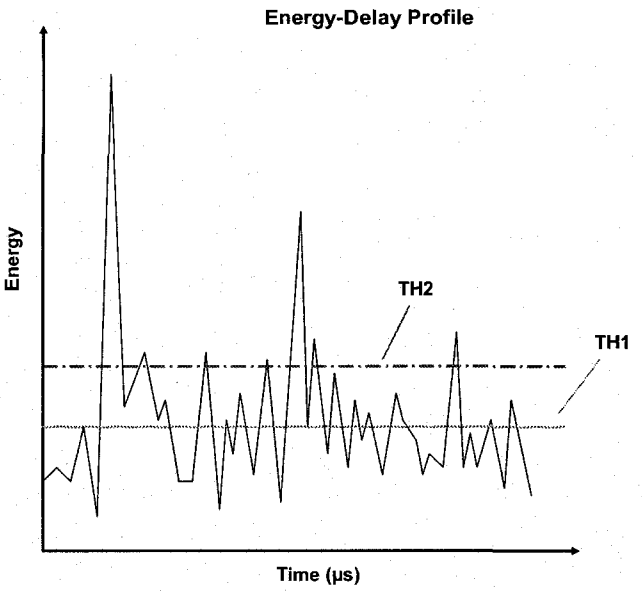
1- A first dwell (hybrid search)

2- A second dwell (hybrid verification)

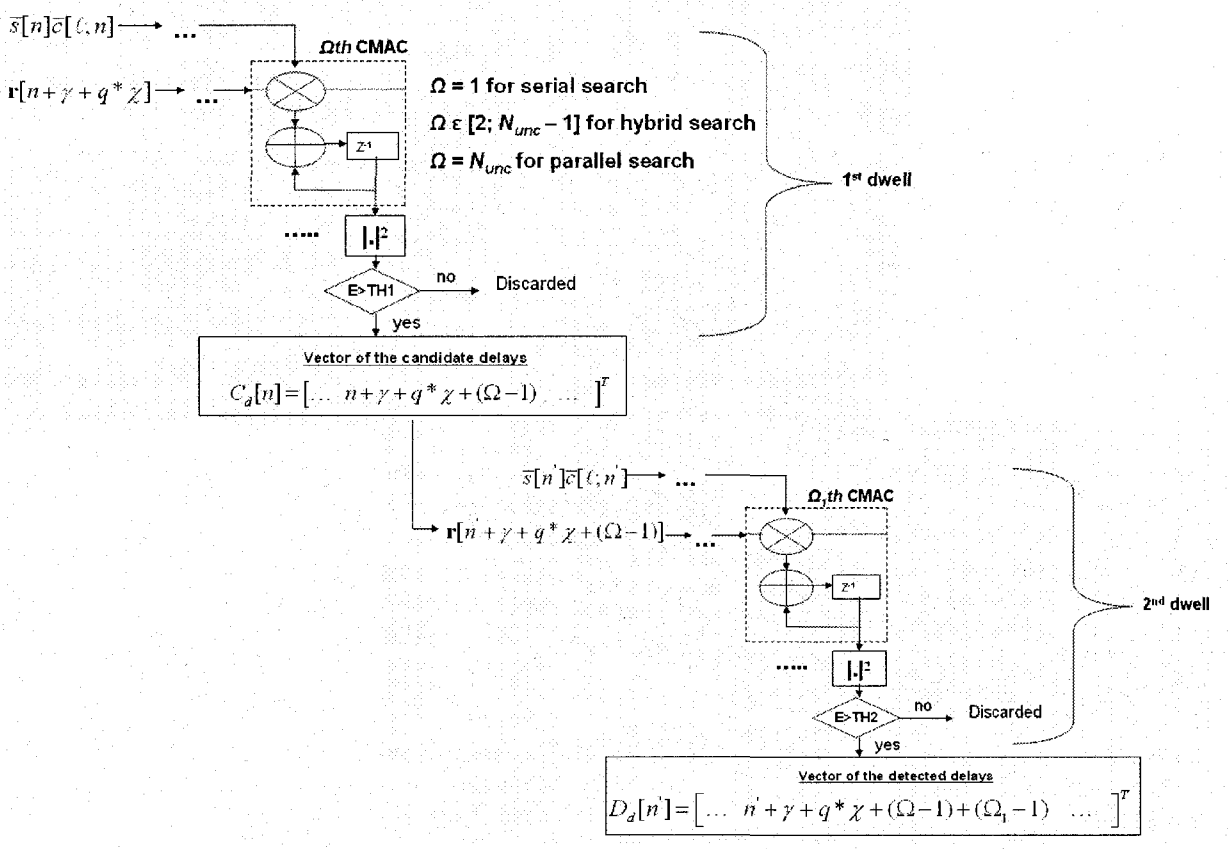
3- A detection stage

The authors also make the following design choices:

- The correlation window W is of 204 chips,
- The over-sampling factor χ is of 2,
- The number of uncertainties N_{unc} is equal to 408, i.e. $W \times \chi$,
- The length N' of the correlation is 512 chips for both the first and the second dwell,
- A hybrid search with $\Omega = \Omega_1 = 6$ CMACs is used in both the first and the second dwell.



(a)



(b)

Figure 2.6 Two-stages delay acquisition a) Threshold values, b) Schematic.

Figure 2.7 shows how the different stages interact in [4]. In the first dwell, 6 CMACs are used in parallel to compute the matrix product $\mathbf{r}[n]\overline{\mathbf{c}}[n]$. Because of the chip rate processing, a time interval of 1 frame is needed to perform the 408 correlations of length $N' = 512$ chips. Thereafter, the delays of the candidates with energy greater than a first threshold *TH1* are stored in a vector (up to 36 candidates). As shown in the figure, the potential candidates are only sent to the verification stage once the frame has been completely scanned.

In the second dwell, each potential candidate from the first dwell is correlated 10 times. The energies of the 10 correlations are averaged and compared against a second threshold *TH2* for the verification of the potential candidates. Six CMACs and a time interval of 1 frame are also necessary to implement this second stage.

The procedure described above is repeated for three subsequent frames. Once the 2nd dwell of the 3rd frame is completed, the detection stage compares the energy-delay profiles of the two dwells for the three frames. The valid paths are declared based on the observation of the 6 energy-delay profiles. Therefore, the path detection is only done at the 5th frame as shown in Figure 2.7.

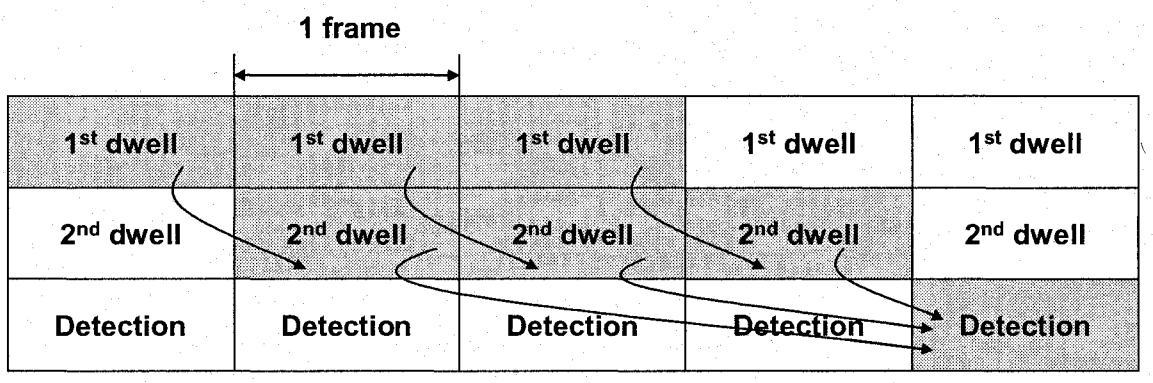


Figure 2.7 Timing diagram of a pipelined multipath searcher [4].

One can observe that the pipeline approach as a fixed acquisition time, i.e. it does not depend on the values of the thresholds anymore, which is of 5 frames in the present

reference. Thus, there is no need to refer to the MAT and we can simply use the term *acquisition time*. It is important to mention that the probabilities of a false alarm and misdetection of the pipeline approach are still dependant on the values of the thresholds through the different dwells.

Even though the method presented in [4] has interesting features for practical implementation, the data are processed at the chip rate and 12 CMACs are required for real-time processing. Moreover, extra hardware is needed for the computation of the signal energy ($I^2 + Q^2$) as well as for averaging the energies over many correlation periods.

Because of the limited hardware resources and the low rate processing, the multipath searcher can only screen 408 offsets per frame during the initial search. Moreover, a time interval of 1 frame is necessary for the verification of the 36 candidates in the verification stage. The combination of these constraints results in a long acquisition time and a low resolution. By developing a high performance complex multiplier-accumulator, we could reduce the acquisition process and even improve the resolution of the system by testing more offsets in a shorter period of time. Moreover, the following calculations could be performed by a single CMAC:

- The correlation between the complex scrambling code $\overline{\mathbf{c}}[n]$ and the I-Q components of the incoming signal $\mathbf{r}[n]$.
- The calculation of the correlation energies $\mathbf{E}[n]$ ($I^2 + Q^2$).
- The averaging of the energies over many correlation windows.

The most important benefit of such a method would be the reduction of the hardware complexity.

2.2 Existing Designs of Fast Arithmetic Units

In order to provide a powerful motor capable of handling all of the tasks identified in the previous section, we must design fast and polyvalent complex multiplier-accumulator. Although many designs of rapid multipliers and MACs have been reported in the literature [21] - [24], actual realization of CMACs on ASIC with sufficient details about their implementation remains an uncovered topic. Only a few references describe circuits with such functionality [25] - [27].

In [28], some conceptual schemes for the multiplication of complex numbers in a radix-r format are presented but no implementation result is reported. In [25], a 60 MHz CMAC is proposed as a DSP engine and is implemented in ASIC. Unfortunately, the implementation result focuses on the overall DSP architecture and no detail about the CMAC structure and transistor count is given. The only available implementation results of a complex multiplier and CMAC are provided by [26] – [27], and they are realized on Xilinx FPGAs.

Some wave pipelined and conventional pipelined ASIC designs of multiplier and MAC are reported in [21] – [24]. All of theses circuits are implemented in various technologies, using different power supplies. Moreover, the method for measuring the throughput varies depending on the authors; some results are from schematic simulations [22] while others are from post layout simulations [21]-[23] or even hardware testing [24]. It is at our knowledge, the first time that a complex multiplier-accumulator is implemented in ASIC using the wave pipeline method. Therefore, it is difficult to find a reference that will be a fair comparison. The reference [21] will be used only to compare the design approaches while the MAC presented in [23] will be used to compare the speed and the area. The choice of reference [23] for the performance comparison is justified by the fact that the author presents the detail of how the maximal frequency has been determined; information that is not available in the other references.

2.3 Wave Pipeline Technique

The wave pipeline method is an optimization technique of the data flow inside a combinational logic block. In the conventional pipeline, the synchronization is secured by inserting sequential elements such as registers, at the cost of increased power dissipation and circuit area. The throughput of a wave pipeline circuit depends on the difference between the longest and the shortest paths [5]. The synchronization is achieved by equalizing all the propagation delays in the CLB. By doing so, subsequent inputs can be introduced into the circuit at time intervals that are shorter than the propagation delay of the circuit. Therefore, many different inputs, also called *waves*, are processed simultaneously in the CLB without interfering with each other as shown in Fig. 2.8.

The conventional pipeline method benefits from optimization tools, design flows and libraries of standard cells. Nevertheless, it has some drawbacks, such as the propagation delays of the clock (clock skew) and the increased power consumption caused by the clock tree and the associated registers. The wave pipeline technique allows one to increase the speed of a circuit without introducing these disadvantages [21], [22]. As a drawback, custom design is required and tuning must be done by hand at different levels of abstraction (architectures, layout, standard cells), making the design cycle time consuming. In this section, fundamental concepts about wave pipelining will be presented as well as existing design approaches.

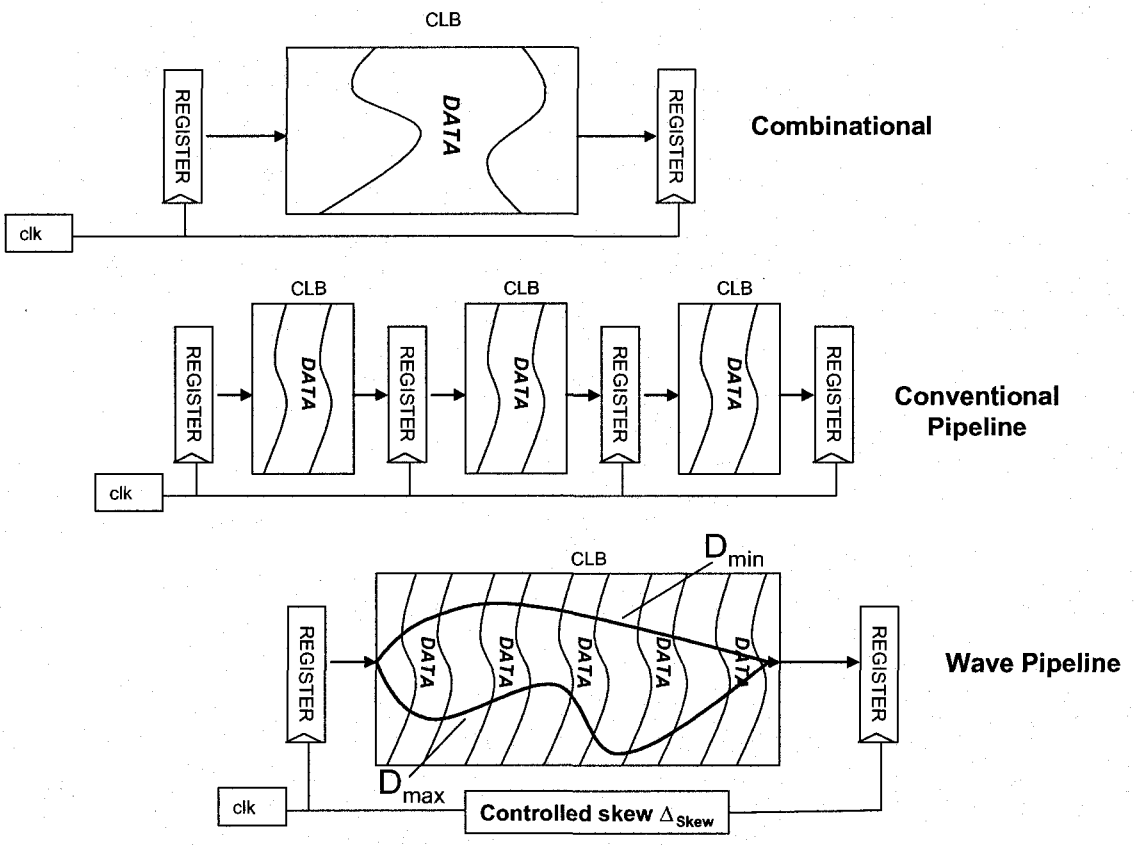


Figure 2.8 Concept of the wave pipeline [21].

2.3.1 Wave Pipelining Timing Constraints

A new data can only be transmitted into a wave pipeline circuit when it is guaranteed that its fastest path will not arrive at the output register before the slowest path of the previous data. Thus, we can determine the register clocking constraints that will ensure the respect of this condition with regards to the different path lengths. As a general rule, every wave must be captured at the output register at every time T_L expressed as [5],

$$T_L = N_w \times T_{CLK} + \Delta_{Skew}, \tag{2.14}$$

where N_w represents the number of waves, i.e. samples, introduced into the circuit since the beginning of the process, T_{CLK} is the clock period and Δ_{Skew} is the controlled clock skew between the input and output registers (see Fig. 2.8).

Two different cases must be considered to determine the minimum clock period. The first one is the sampling of the slowest path of the actual wave, the i[th] wave ($D_{MAX,i}$). It is equivalent to the critical path timing constraint of the conventional pipeline and it is expressed as:

$$T_L > D_R + D_{MAX,i} + T_S + \Delta_{CLK}\,, \tag{2.15}$$

where D_R and T_S are the propagation delay and the setup time of the registers, and Δ_{CLK} is the uncontrolled clock skew. The second timing constraint is related to the sampling of the fastest path of the next wave, the (i+1)[th] wave ($D_{MIN,i+1}$) and it is expressed as:

$$T_L < T_{CLK} + D_R + D_{MIN,i+1} - (\Delta_{CLK} + T_H)\,, \tag{2.16}$$

where T_H is the hold time of the registers. Figure 2.9 illustrates the three precedent equations, i.e. (2.14), (2.15) and (2.16).

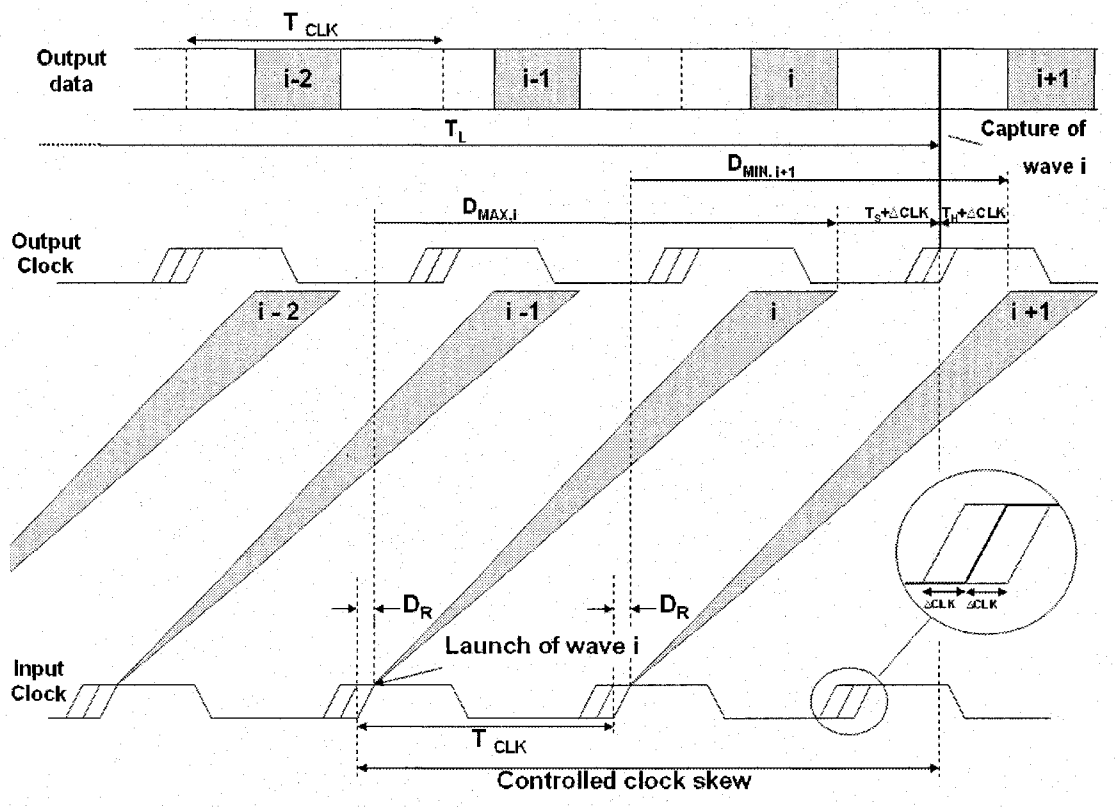


Figure 2.9 Wave pipeline timing constraints [5].

By combining equations (2.15) and (2.16), we obtain the requirement of the minimal clock period,

$$T_{CLK} > (D_{MAX} - D_{MIN}) + T_S + T_H + 2 \times \Delta_{CLK} . \quad (2.17)$$

As outlined in (2.17), the maximal frequency depends on the difference between the maximum and the minimum propagation delays plus the overhead due to the sample and hold times of the registers as well as the uncertainty on the clock signal.

2.3.2 Source of Delay Variations

The performance of a wave pipelined circuit highly depends on the difference between the propagation delays of the fastest and the slowest paths. However, the propagation delays of a VLSI circuit can be affected by many different factors so that the delay equalization is a challenging task. Here is a list of items that can degrade the performance of a wave pipeline circuit [5]:

1) Gate delay variations: Different input combinations will create different current paths to charge/discharge the output node of the logic gate so that the transition time will not always be the same. This phenomenon is also called input pattern sensitivity.
2) Temperature variations: The driving capacity of MOS transistors is sensitive to temperature variations. This factor must be considered if an IC is intended to operate in a broad range of temperature.
3) Coupling capacitances: This is a local phenomenon in the layout due to coupling between wire interconnects. As the technology is scaled down, they become more important and they limit the layout performances.
4) Process variations: The foundry's chemical process is not perfect and there might be some variations in the composition of the transistors. Consequently, some delay variations will be induced by these mismatches.
5) Power supply noise and variations.

A mathematical expression that represents the influence of the environmental condition variations can be developed starting from equations (2.15) and (2.16). If we introduce two new variables T_{MIN} and T_{MAX} expressed as

$$T_{MAX} = D_R + D_{MAX} + T_S + \Delta_{CLK}, \tag{2.18}$$

$$T_{MIN} = D_R + D_{MIN} - (T_H + \Delta_{CLK}). \tag{2.19}$$

Equation 2.17 can be rewritten as

$$T_{CLK} > T_{MAX} - T_{MIN}, \tag{2.20}$$

where T_{max} and T_{min} are specified at nominal temperature. Considering β_f >1, a scaling factor for higher temperatures, the clock period becomes

$$T_{CLK} > \beta_f (T_{MAX} - T_{MIN}). \tag{2.21}$$

For a temperatures below the nominal, the scaling factor would be β_s <1

$$T_{CLK} > \beta_S (T_{MAX} - T_{MIN}). \tag{2.22}$$

By combining (2.21) and (2.22), we obtain the worst case clock period requirement over a range of temperature

$$T_{CLK} > \beta_f T_{MAX} - \beta_S T_{MIN}. \tag{2.23}$$

In the conventional pipeline, only the worst case delay limits the frequency, while the operating frequency of a wave pipeline circuit also depends on the best case. Thus, two constraints can be influenced by external parameters, and this type of circuit is more sensitive to process, temperature and voltage variations.

2.3.3 Delay Difference Minimization

The principal task in the design of a wave pipeline circuit is the delay equalization of all the logic paths in order to minimize the term $D_{max} - D_{min}$ in equation (2.17). This equalization is done at two different levels of tuning: logic depth (rough tuning) and cell delay (fine tuning).

A combinational circuit is presented in Figure 2.10 (a). As one can observe, the different signals go through a different number of logic gates. In figure 2.10 (b), some buffers have been added to the paths of B and C so that the variables all have the same propagation delay. In the design of a wave pipeline circuit, this step is referred to as coarse tuning. Special care must also be taken during the layout generation so that parallel signals go through similar wire lengths.

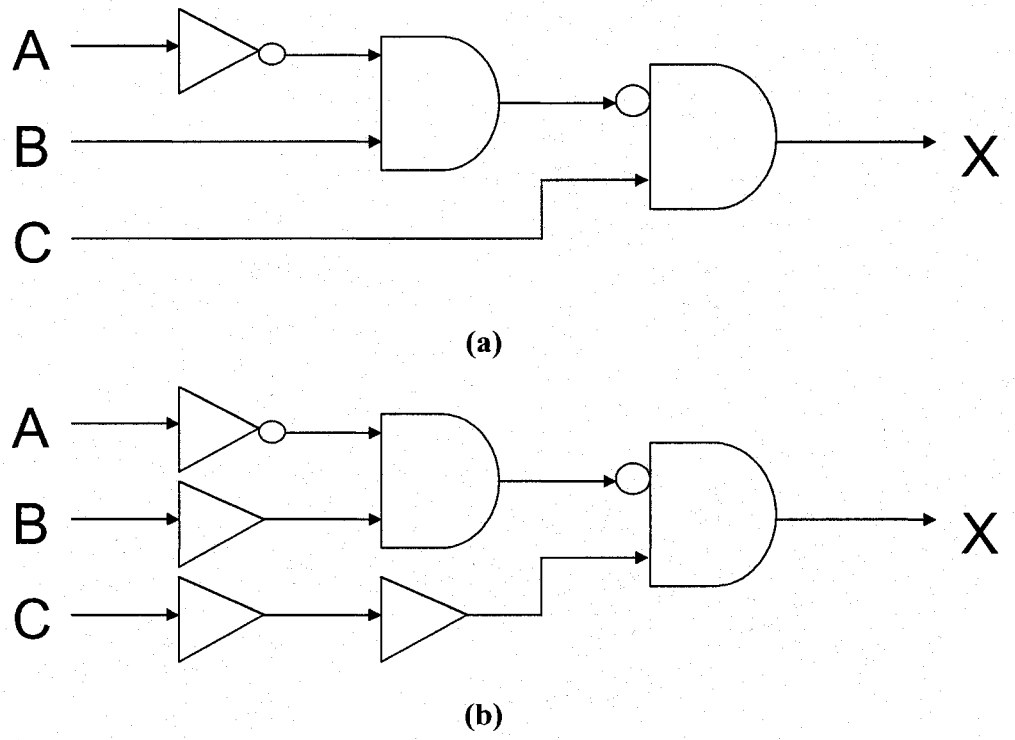


Figure 2.10 a) Original combinational circuit, b) Combinational circuit with coarse tuning.

The second step of tuning, gate delay equalization, is not straightforward to accomplish using conventional CMOS gates because (i) their propagation delays depend on the input combinations and also, (ii) their configuration at the transistor level vary depending on the function, so that different logic gates have different delays. For this purpose, different logic families have been used in the literature and a review of the various approaches is presented in the next section.

2.3.4 Existing Approaches of Fine Tuning

Among the first authors to report a successful implementation of a wave pipelined multiplier, Ghosh and Nandy [29] – [30] have introduced the Normal Process Complementary Pass Transistor Logic (NPCPL) and they have been followed by others ([21], [31]). However, as the supply voltage is scaled down, the NMOS pass transistors have more difficulties in transferring logic ones introducing some significant delay variations. The Complementary Pass-Logic Transmission-Gate (CPL-TG) has been used to implement some full-adder blocks in [32]. Basically, this cell can implement the same logic functions than the NPCPL family. The main difference is that some CMOS transmission gates are replacing the NMOS pass transistors, allowing for a better delay equalization. Pass logic gates such as NPCPL and CPL-TG mostly implement two-input logic functions, resulting in a fan-in constraint.

In [33] – [34], another approach based on Wave pipeline Transmission Gate Logic (WTGL) is presented. Other types of logic such as the pseudo-NMOS [35] and the dual-rail static CMOS (DRSCMOS) [36] have also been used in the literature. However, the logic families presented in [33]-[36] all suffer from unequal current paths.

Some authors have implemented wave pipelined arithmetic units on FPGAs in [37] – [39]. However, the maximal speed obtained on an FPGA is 185 MHz for a 2×6 bit MAC in [38]; which is not impressive compared to the throughputs obtained with wave pipelined circuits in ASIC.

More recently, a hybrid technique named HyPipe [40]-[42] has combined the features of the conventional pipeline and the wave pipeline methods. Even though impressive throughput can be obtained with this method, the area cost increases significantly as well as the switching activity. The authors of these references do not report the transistor count of their designs, and they do not make comparisons of the power consumption with similar wave pipelined circuits.

In this thesis, the NPCPL and the CPL-TG logics will be used to quantify the impact of the logic family on the performances of the wave pipeline CMAC.

2.4 Summary

Many acquisition techniques are presented in the literature. The serial, the parallel and the hybrid searches can be used to implement one verification stage of the multipath searcher. More advanced techniques combine a few verification stages, called dwells, to improve the quality of the detection. The pipelining of the verification stages in multiple-dwell schemes has the advantage of a fixed acquisition time. In conventional implementation of the multipath searcher, a processing rate of 3.84 MHz (chip rate) is used and hardware duplication is needed for real-time processing.

Certain parameters of the multipath searcher such as the acquisition time and the resolution highly depend on the hardware ressources. Improving one of these parameters normally requires proportional increase in either hardware or processing time. In this work, high-speed processing will be used to avoid these two drawbacks.

The wave pipeline method will allow for the development of a high performance CMAC. In a wave pipeline circuit, many different inputs (also called *waves*) are processed simultaneously in the combinational logic block without interfering with each other. This is achieved by equalizing all the propagation delays in the circuit. Moreover, the wave pipeline technique does not suffer from clock skew because all of the data and control signals go through the same logic depth. Since additional registers are not

introduced, the clock tree is kept simple and the switching activity is reduced compared to the conventional pipeline method. Considering that the clock tree can dissipate up to 40 % of the power in an Integrated Circuit (IC) ([3], [43]), the overall power savings can be significant.

The design of a wave pipeline circuit in ASIC requires development of logic gates that present specific characteristics:

(i) A given logic gate must have constant propagation delays for different input transition patterns.

(ii) Different logic gates must have similar propagation delays.

The NPCPL logic has been widely used in the literature but it presents some delay variations has the supply voltage is scaled down; because the NMOS pass transistors have more difficulties in transferring the logic ones. The CPL-TG will allow for a better fine tuning at the cost of an increased number of transistors and power consumption.

In the next chapter, the drawbacks of the existing approaches for the design of the multipath searcher will be addressed as a contribution of this work. Further investigations will target the realization of wave pipelined arithmetic units. Finally, the NPCPL and the CPL-TG logics will be used to quantify the impact of the logic family on the performances of the wave pipeline CMAC.

3 Design of a Multipath Searcher

In the previous chapter, some existing algorithms of multipath searcher have been presented. It is shown that multiple-dwell techniques combined with hybrid searches are among the most widely used schemes at present time. For practical implementations, the pipelining of the search process has the advantage of a fixed acquisition time.

This thesis presents a specific algorithm based on a multiple-dwell scheme in which different detection stages are pipelined. The algorithm and its implementation aim at providing a better quality of path detection.

In the work of this thesis, high-speed processing is expected to improve the multipath searcher quality without increasing the hardware complexity. A single high-speed CMAC could perform more calculations in a shorter time interval. The implementation of the algorithm based on a processing rate of 3.84 MHz, i.e. the chip rate, would normally require tens of CMACs resulting in complex hardware architecture.

In this section, a multipath searcher algorithm and its implementation will be presented. Then, the speed requirement of the implementation will be determined. Finally, a CMAC will be developed using the wave pipeline method and low level issues regarding the timing properties of different types of logic will be addressed.

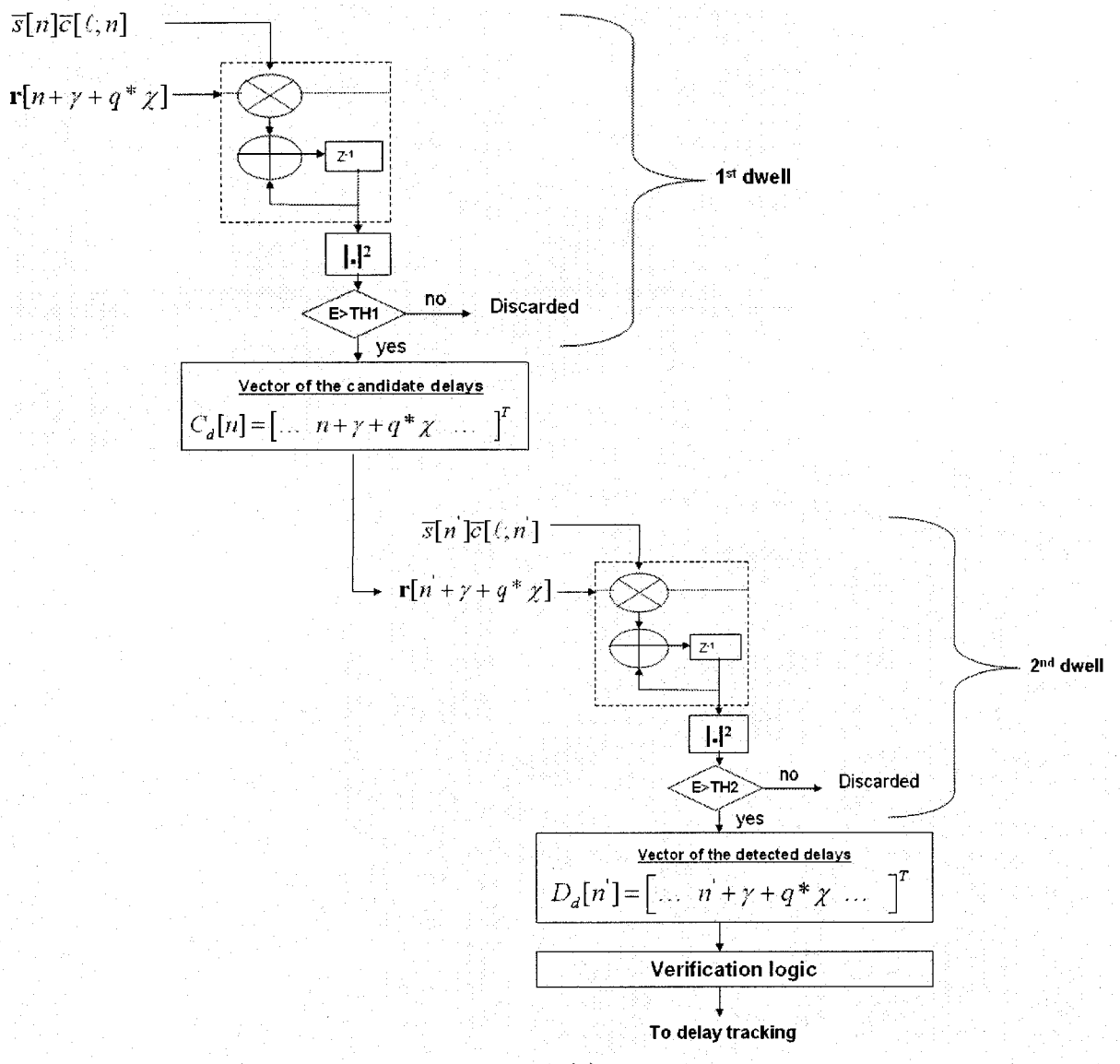
3.1 Coarse Acquisition Algorithm

The main objective of this section is to determine a multipath searcher with the ability to update the delay profile at short time intervals and with a high accuracy. The double-dwell algorithm [19]-[20] is considered as one of the most effective searching algorithms. Its top level scheme is shown in Figure 2.6 (b). The algorithm specified in this work is based on it. However, issues concerning the computation process and its implementation are addressed as follows.

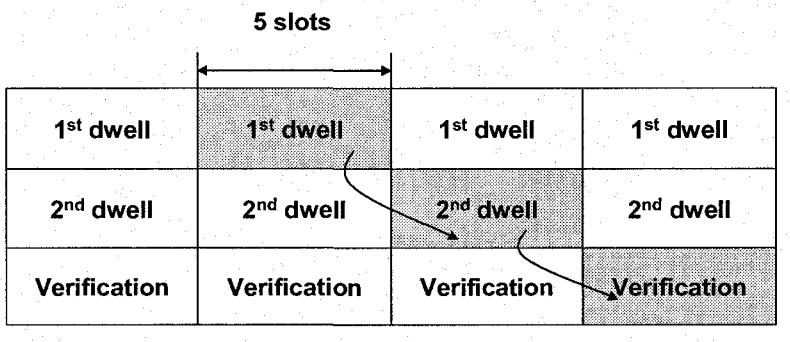
1- After two stages of dwell, a verification stage is added to reduce the rate of false alarm.
2- In view of the algorithm implementation, a short search period is targeted so that an acquisition can be made during a short time interval, to improve the quality of detection. Thus, the parameters of the algorithm need to be determined according to this target.

Figure 3.1 (a) presents the top level scheme of the multipath searcher algorithm of this work. As one can observe, the scheme consists of a double-dwell, each of which is made of a serial search, and a stage of verification logic.

Figure 3.1 (b) presents the timing diagram of the target implementation of the algorithm. In order to reduce the acquisition time, the search period, which can be thought as the "clock period" of the process, is defined to be 5 slots. Thus, a total of 15 slots are necessary to perform the path detection. As illustrated in the figure, the potential candidates are passed from one stage to another at the beginning of a search period. A reduction of the search period results in increased requirements of computation speed, which makes a critical challenge to the hardware implementation of the algorithm.



(a)



(b)

Figure 3.1 a) Top level scheme and b) Timing diagram of the proposed multipath searcher.

Considering the objective of better detection quality, the algorithm and its implementation must meet the following specifications:

1 - The acquisition time must be inferior to 2 frames,

2 – The precision of the estimated delays must be within one half of a chip period, i.e. $T_c/2$,

3- The false alarm and misdetection probabilities must be of 10% or less ($P_{fa} \leq 0.1$, $P_{miss} \leq 0.1$), for certain channel conditions defined by the 3GPP standard. The target of 10 % corresponds approximately to the results obtained in an existing implementation [4] in order to meet the standard requirements.

In order to meet the above objectives, the following measures are taken:

- The acquisition time is improved by using a short search period in comparison with an existing multipath searcher [4]. As a result, an increased processing speed is required in hardware.

- To meet the resolution requirement, the over-sampling factor χ is set to 4 so that the multipath components can be resolved within a quarter of chip duration. Increasing the resolution normally requires proportional increase in either hardware or processing time. In this work, high-speed processing is used to avoid these two drawbacks.

- Four different measures are taken to improve the false alarm and misdetection probabilities:

  - A third stage, called the verification logic, is added to make sure that the identified paths comply with a set of pre-established rules.

  - The length of the correlation is doubled in the second dwell in order to have better energy estimates [8].

  - The thresholds of the first and the second dwell (*TH1* and *TH2*) are proportional to the noise floor. The different scaling factors depend on the Signal-to-Noise Ratio (SNR) [15].

  - Offset interleaving is used in both the first and the second dwell to take advantage of the time diversity [4], [20].

3.1.1 Stage of the 1st Dwell

This section presents the details related to the first stage of the algorithm. In a first step, we must define the different parameters such as the correlation window size W, the over-sampling factor χ and the length of correlations N'. It has been mentioned at section 2.1.1 that the spreading factor N_c of the pilot signal corresponds to 256 chips in WCDMA downlink transmissions. In order to have better energy estimates, the correlation will be performed over 2 pilot symbols and therefore, the correlation length N' is fixed to 512 chips for the first dwell. The precision of the estimated delays must be within one half of a chip period, i.e. $T_c/2$, as per our specifications. To meet this requirement, the over-sampling factor χ is set to 4, allowing for an error margin of ¼ of a chip.

Accordingly to the 3GPP specifications [44], the maximal delay one can expect is of approximately 20 µsec (81 chips). For a proper operation, we will consider a safety margin of 20 % so that the multipath searcher will verify the delays in the range of ±96 chips. Therefore, considering an over-sampling factor $\chi = 4$, the number of uncertainties N_{unc} to be tested by the first stage is defined by:

$$N_{unc} = 2 \times 96 \times 4 = 768 \text{ offsets.}$$

Offset interleaving is a robust strategy that has been introduced in [20] to take advantage of the random nature of the interfering effects. This method helps in resolving closely spaced multipath by testing subsequent samples at different points in time so that the interfering noise is uncorrelated. In the proposed searcher, subsequent offsets separated by 0.25 chip will be tested at one slot (i.e. 10 pilot symbols) interval and offsets separated by 1 chip (4 samples) will be tested at 1 pilot symbol interval (256 chips). Moreover, every offset will be correlated 2 times, at one slot interval, and the values will be averaged before the candidate delays $C_d[n]$ are identified.

Figure 3.2 illustrates the offset interleaving strategy used in this work. The search window W is divided into two sections. The first section consists in the tests 1 to 12 and it is intended to scan the positive offsets in the interval [1; 96] chip. The second section corresponds to the tests 13 to 24 and it is scanning the negative offsets in the interval [-96

; -1] chip. As shown on figure 3.2, there is an interval of 8 chips (32 samples) between subsequent correlations and a total of 24 offsets are tested per pilot symbol.

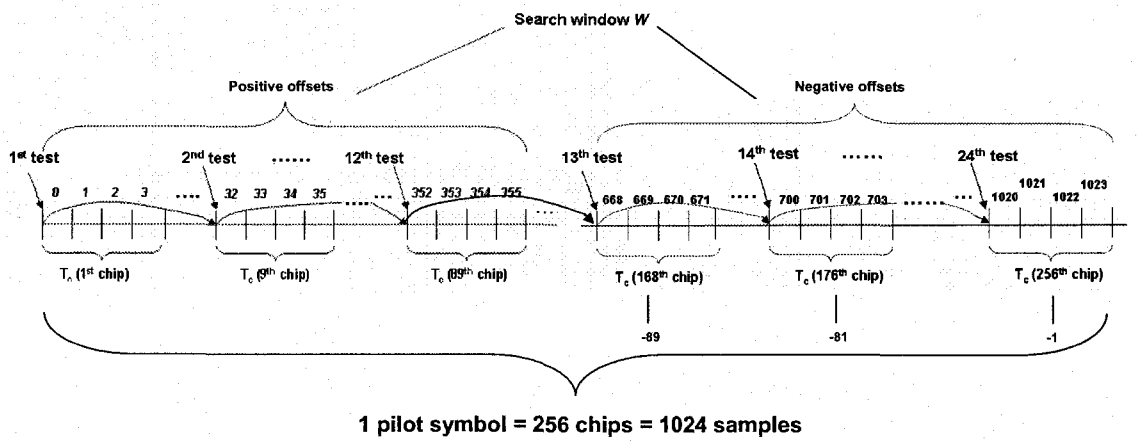


Figure 3.2 Offset interleaving (1st pilot symbol, 1st slot).

The timing diagram of the second pilot symbol of the first slot is shown in Figure 3.3. On the figure, one can observe that the positive offset window is now in the range of [2; 90] chips. Similarly, the negative offset window goes from -90 to -2 chips. In comparison with the first pilot symbol, the starting point of the positive offsets has been increased by 1 chip (4 samples) and the starting point of the negative offsets has been decreased by 1 chip.

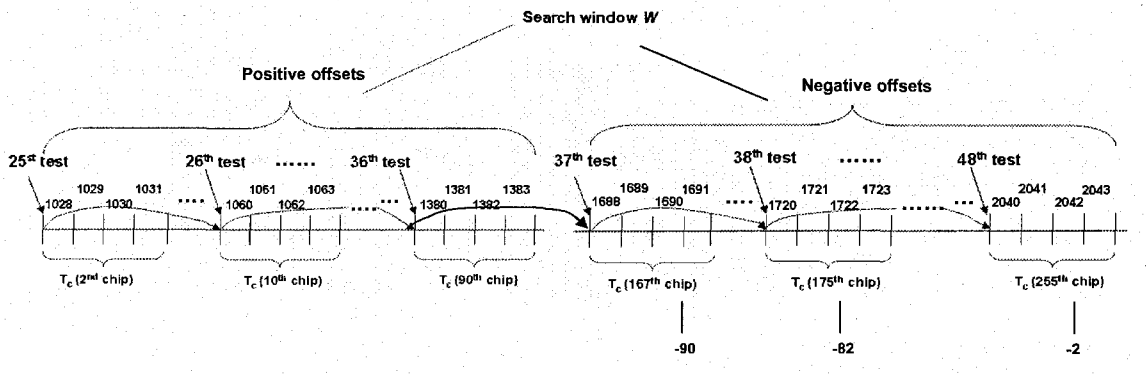


Figure 3.3 Offset interleaving for the 2nd pilot symbol of the 1st slot.

There are 10 pilot symbols per slot in WCDMA downlink transmissions. Since the correlations are performed over two pilot symbols (N'= 512), no offset is tested at

symbols 9 and 10 (shadowed in Figure 3.4) so that the correlations do not overflow into next slot.

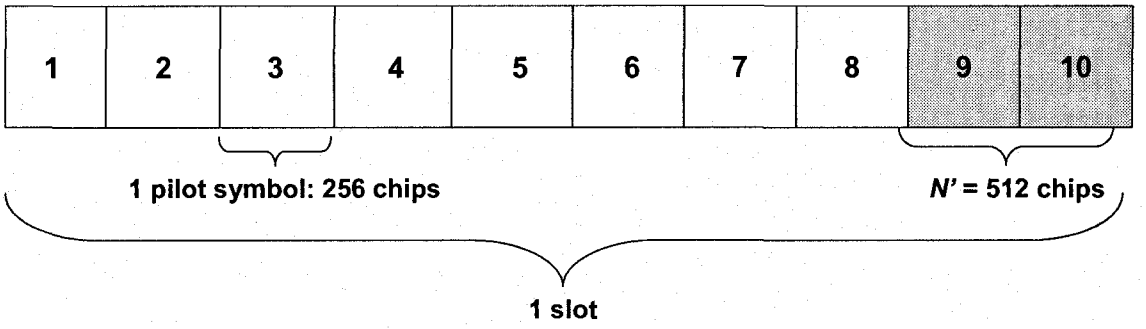


Figure 3.4 Time guard between subsequent slots.

There are 192 offsets tested per slot (8 symbols/slot × 24 test/symbol) and 4 slots are necessary to scan the 768 uncertainties. Table 3.1 summarizes the offset interleaving strategy used in this work. In the table, the values between the brackets represent the chip indexes that are tested for a given pilot symbol, and the fraction stands for either the 1st, 2nd, 3rd or 4th sample of the chip.

Pilot symbol	1st slot	2nd slot	3rd slot	4th slot
1	±{1,9,17,25,33,41,49,53 ,65,73,81,89}×1/4	±{1,9,17,25,33,41,49,57 ,65,73,81,89}×2/4	±{1,9,17,25,33,41,49,57 ,65,73,81,89}×3/4	±{1,9,17,25,33,41,49,57 ,65,73,81,89}×4/4
2	±{2,10,18,26,34,42, 50,58,66,74,82,90}×1/4	±{2,10,18,26,34,42, 50,58,66,74,82,9}×2/4	±{2,10,18,26,34,42, 50,58,66,74,82,90}×3/4	±{2,10,18,26,34,42, 50,58,66,74,82,90}×4/4
3	±{3,11,19,27,35,43,51, 59,67,75,83,91}×1/4	±{3,11,19,27,35,43,51, 59,67,75,83,91}×2/4	±{3,11,19,27,35,43,51, 59,67,75,83,91}×3/4	±{3,11,19,27,35,43,51,59, 67,75,83,91}×4/4
4	±{4,12,20,28,36,44,52, 60,68,76,84,92}×1/4	±{4,12,20,28,36,44,52, 60,68,76,84,92}×2/4	±{4,12,20,28,36,44,52, 60,68,76,84,92}×3/4	±{4,12,20,28,36,44,52,60, 68,76,84,92}×4/4
5	±{5,13,21,29,37,45,53, 61,69,77,85,93}×1/4	±{5,13,21,29,37,45,53, 61,69,77,85,93}×2/4	±{5,13,21,29,37,45,53, 61,69,77,85,93}×3/4	±{5,13,21,29,37,45,53,61, 69,77,85,93}×4/4
6	±{6,14,22,30,38,46,54, 62,70,78,86,94}×1/4	±{6,14,22,30,38,46,54, 62,70,78,86,94}×2/4	±{6,14,22,30,38,46,54, 62,70,78,86,94}×3/4	±{6,14,22,30,38,46,54,62, 70,78,86,94}×4/4
7	±{7,15,23,31,39,47,55, 63,71,79,87,95}×1/4	±{7,15,23,31,39,47,55, 63,71,79,87,95}×2/4	±{7,15,23,31,39,47,55, 63,71,79,87,95}×3/4	±{7,15,23,31,39,47,55,63, 71,79,87,95}×4/4
8	±{8,16,24,32,40,48,56, 64,72,80,88,96}×1/4	±{8,16,24,32,40,48,56, 64,72,80,88,96}×2/4	±{8,16,24,32,40,48,56, 64,72,80,88,96}×3/4	±{8,16,24,32,40,48,56,64, 72,80,88,96}×4/4

Table 3.1 Offset interleaving strategy.

As shown in Figure 3.5 (a), the procedure presented above is repeated two times, at one slot interval. The first search window is performed in slot 1 to 4 and it results in the observation matrix $\mathbf{R}'[n]$. The second search window is performed in slots 2 to 5 and the associated observation matrix is $\mathbf{R}'[n']$, where $n' = n + 2560 * 4$ (in samples). Figure 3.5 (b) presents the schematic of the first dwell. As illustrated, the correlation values in the matrices $\mathbf{R}'[n]$ and $\mathbf{R}'[n']$ are squared and summed to produce the matrix of the energies $\mathbf{E}[n]$ that can be written as

$$\mathbf{E}[n] = \begin{bmatrix} E[n-96*4] & \cdots & E[n-96*4+\gamma] & \cdots & E[n-96*4+3] \\ \vdots & & \vdots & & \vdots \\ E[n-q*4] & \cdots & E[n-q*4+\gamma] & \cdots & E[n+q*4+3] \\ \vdots & & \vdots & & \vdots \\ E[n+0] & \cdots & E[n+\gamma] & \cdots & E[n+3] \\ \vdots & & \vdots & & \vdots \\ E[n+q*4] & \cdots & E[n+q*4+\gamma] & \cdots & E[n+q*4+3] \\ \vdots & & \vdots & & \vdots \\ E[n+95*4] & \cdots & E[n+95*4+\gamma] & \cdots & E[n+95*4+3] \end{bmatrix}.$$

The starting point of the PN sequence $\overline{s}[n]\overline{c}[\ell;n]$ must be increased by 256 chips for every new pilot symbol as illustrated on Figure 3.5 (c). The scrambling code sequence is constant for the negative offset window of a given pilot symbol as well as for the positive offset window of the next pilot symbol, where it is aligned with offset 0.

It has been mentioned previously (Section 2.1.3) that the threshold *TH1* is set by scaling the noise floor. In a first step, the noise floor is approximated by computing the mean of the matrix of energy $\mathbf{E}[n]$. Then, the threshold *TH1* is set as follows:

- *TH1* = 1.5 × Noise Floor for SNR ≤ 4 dB
- *TH1* = 1.75 × Noise Floor for SNR > 4 dB.

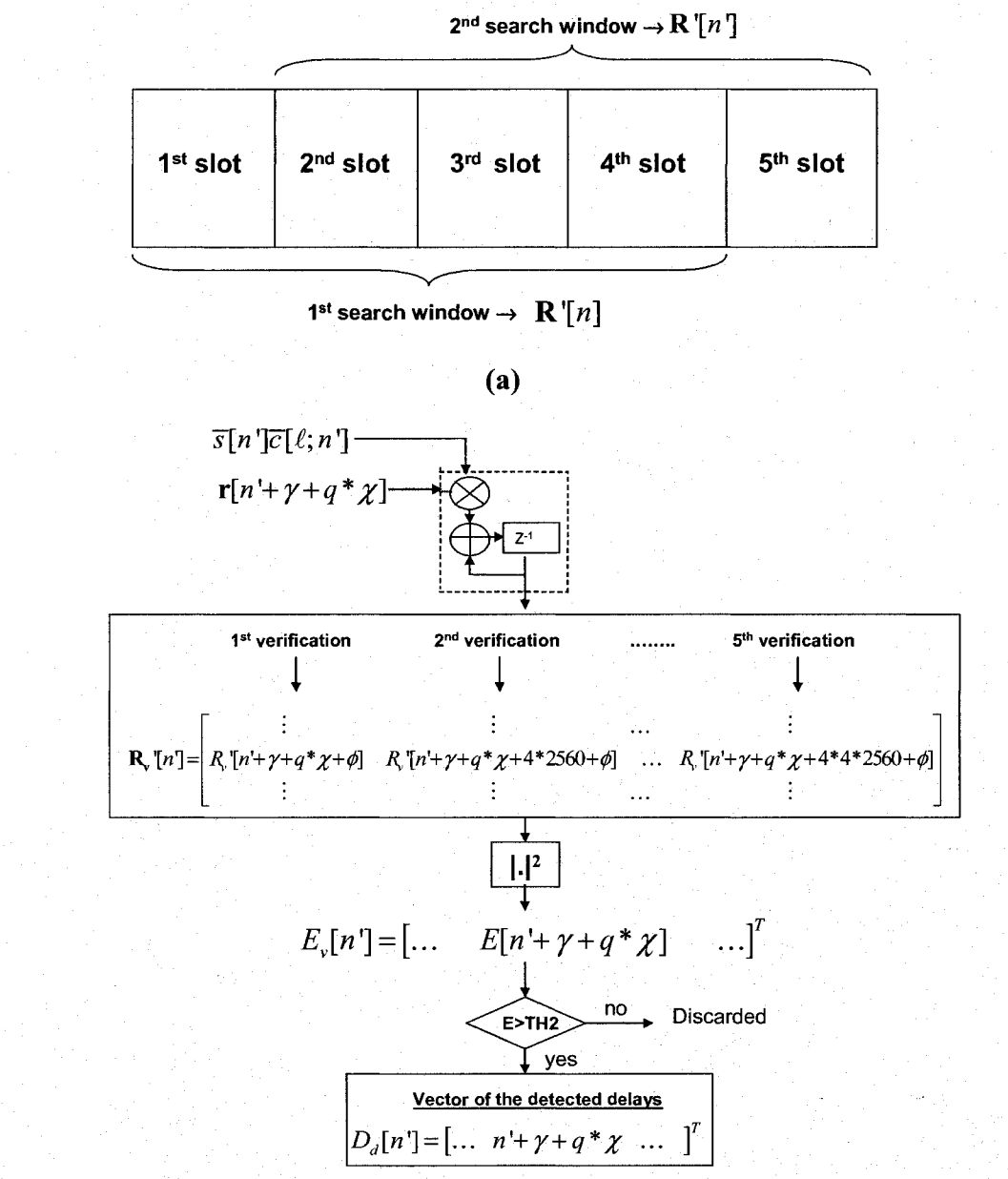


Figure 3.5 **a) Observation periods, b) Schematic of the first dwell, c) Scrambling code sequences.**

The different scaling factors are due to the fact that the noise floor has a higher value at low SNR and therefore, a smaller scaling factor must be used to improve the probability of misdetection. The SNR is calculated by an external module based on the data detection results. Since the SNR estimates are only available after a few frames, a value of 4 dB is assumed at the beginning of the process.

Once the threshold has been determined, every value in the matrix $\mathbf{E}[n]$ is compared with *THI*. The delays with greater energy are stored in the vector of the potential candidate $C_d[n]$ under the form $\{d_i, E_i\}$ where d and E stand for the delay and the energy values of the ith candidate. The vector $C_d[n]$ can accept up to 36 candidates and control logic is necessary in case where more than 36 offsets have energy greater than the threshold. The control logic can be summarized as follow:

1- If a 37^{th} candidate has sufficient energy, place the candidates in the vector $C_d[n]$ in order of decreasing energy.

2- Compare the energy of the 37^{th} candidate with every value in the vector $C_d[n]$, starting with the one with the most energy (the left most candidate).

3- As soon as the 37^{th} candidate has energy greater than a candidate in the vector $C_d[n]$, replace the candidate by the 37^{th} value and move right the remaining of the vector.

4- The right most candidate (i.e. the one with the least energy) is discarded.

At the end of a search period, the $1\times\psi$ vector of the potential candidates $C_d[n]$, where ψ represents the number of candidate, is send to the second dwell for verification.

3.1.2 Stage of the 2nd Dwell

This section presents the details related to the second stage. The role of the second dwell is to verify that potential candidates identified by the first dwell have sufficient energy to be declared as valid paths. There is a latency of 5 slots before this operation can start because it is waiting for the vector $C_d[n]$ from the first stage.

In order to produce some more accurate energy estimates, the length of correlation N' is increased to 1024 chips, which is equivalent to 4 pilot symbols. The use of such a correlation length will help in removing energy peaks dued to noise and interferences. Moreover, every candidate will be correlated 5 times and the different correlation energies will be averaged before they are compared to the second threshold *TH2*.

No candidate is tested at symbols 7, 8, 9 and 10 (shadowed in Figure 3.6) so that the correlations do not overflow into next slot. Therefore, up to 36 candidates have to be correlated during the 6 first symbols.

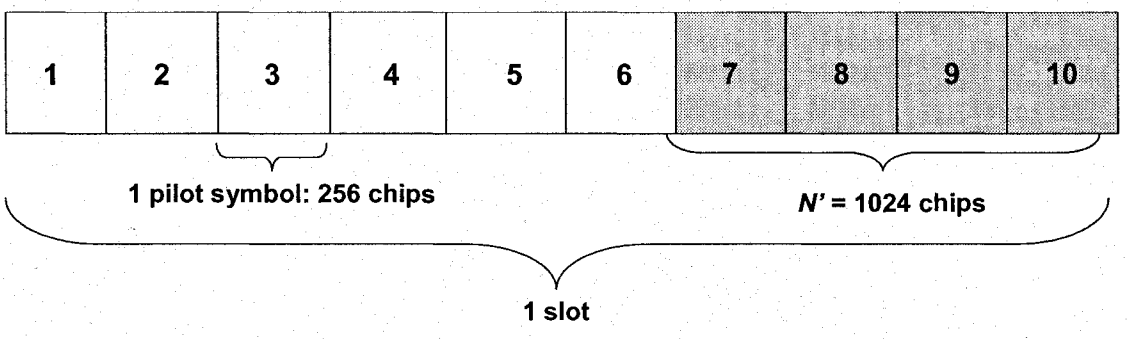


Figure 3.6 Time guard between subsequent slots (2nd dwell).

In most cases, the delays of adjacent candidates in the vector $C_d[n]$ are separated by only 1 sample because of the combined effects of the pulse shaping filter and the oversampling of the analog-to-digital converter. To take advantage of the time diversity, the candidates are tested using an interleaving strategy.

Figure 3.7 illustrates the offset interleaving strategy in the 2nd dwell. As a starting point, it is known that all the candidates must be tested within 6 slots. Therefore, the vector $C_d[n]$ will be divided into 6 groups and the delays belonging to 1 group are correlated within 1 pilot symbol. With this method, the slot is divided into 6 uniformly distributed averaging periods. The number of candidate in each group is defined as:

$$\#\ of\ candidate/group = \psi/6$$

In case where the above operation produces a number with a fractional part, the quotient is simply rounded to the largest integer. Then, an interleaving step is defined using the following equation:

$$Step = \psi\ /\ (\#\ of\ candidate/group).$$

The step value is also rounded to the largest integer. In most cases, the step is equal to 6 candidates. At the beginning of the process, the starting point of the interleaving scheme corresponds to the left most candidate of the vector $C_d[n]$, as illustrated on figure 3.7. Assuming a step of 6 and a total of 36 candidates, the potential delays would be tested in the following order: 1, 7, 13, 19, 25 and 31. Since a next step would bring us to the index 37, which does not exist in the vector $C_d[n]$, the starting point is increased by 1 and the process is restarted. Thus, the next delays to be tested would be those at indexes 2, 8, 14, 20, 26 and 32. This process continues until all the potential delays in the vector $C_d[n]$ have been tested.

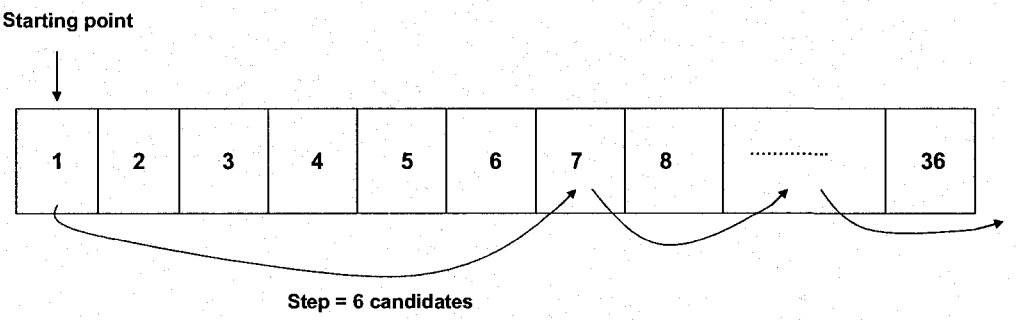


Figure 3.7 Offset interleaving strategy for the 2nd dwell.

The procedure presented above is repeated 5 times, at one slot interval and the correlation results are stored in the $\psi \times 5$ verification matrix $\mathbf{R}_v'[n']$. As shown in Figure 3.8, every column of the matrix $\mathbf{R}_v'[n']$ is used to store the correlation results of 1 verification period. In the figure, the variable ϕ can take any value in the range [0; 6×4×256-1] samples. The symbol ϕ is only used to illustrate that a given offset can be tested anywhere within the 6 first pilot symbols of a given slot.

As in the first dwell, the stream $\mathbf{r}[n]$ is shifted for different offsets while the scrambling code sequence $\overline{\mathbf{c}}[n]$ is constant for a given pilot symbol. Whenever the number of correlation per group is reached (6 in the example), the starting point L of the PN sequence $\overline{\mathbf{c}}[n]=\left[\overline{s}[n]\overline{c}[L;n] \ \cdots \ \overline{s}[n]\overline{c}[\ell;n] \ \cdots \ \overline{s}[n]\overline{c}[L+N-1;n]\right]$ is increased by 256 chips. Similarly, the starting point n of the incoming signal $\mathbf{r}[n]$ is increased by 4×256 samples to consider the next pilot symbol.

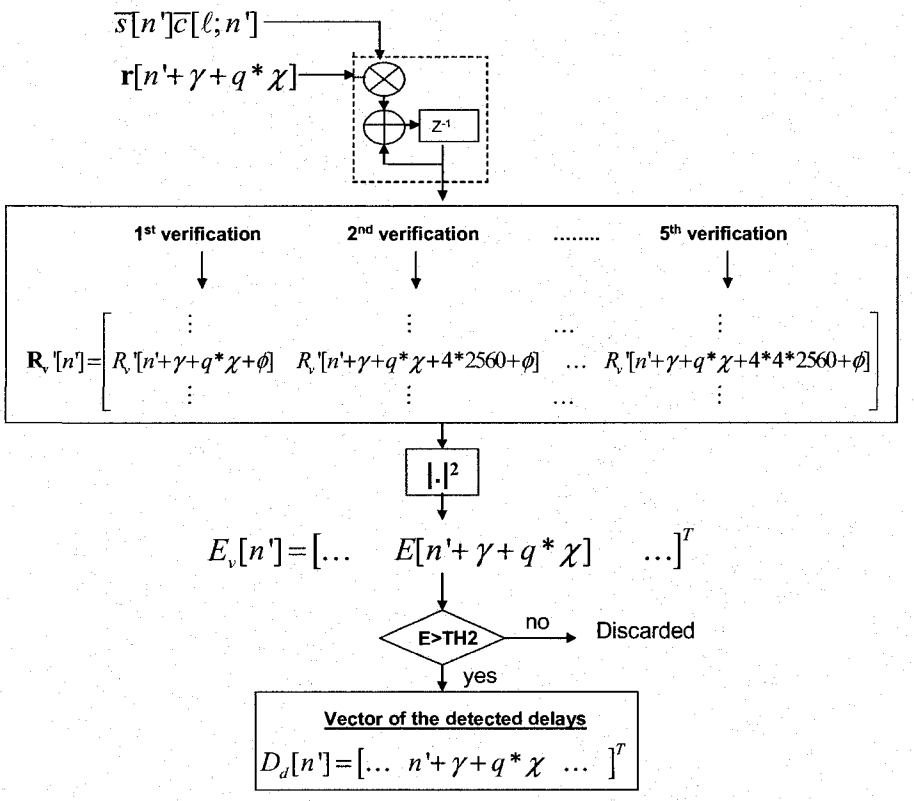


Figure 3.8 Schematic of the 2nd dwell.

The $\psi \times 1$ vector of the energies $E_v[n']$ is obtained by squaring the I-Q components of every correlation result and by computing the mean of the verification matrix $\mathbf{R}_v'[n']$ along the second dimension. Every value of the vector $E_v[n']$ is compared against the second threshold *TH2* which is determined accordingly to the following rules:

- *TH2* = 1 × Noise Floor for SNR < 4 dB
- *TH2* = 1.5 × Noise Floor for SNR ε [4; 7.99] dB
- *TH2* = 1.75 × Noise Floor for SNR ε [8; 11.99] dB
- *TH2* = 2.25 × Noise Floor for SNR ≥ 12 dB

It is important to mention that the noise floor calculated in the first dwell is re-used in this second step. In general, this approximation yields in accurate threshold values because the SNR does not change dramatically over a period of 5 slots. The delays with energy greater than the threshold *TH2* are stored in the vector of the detected delays $D_d[n']$, which is sent to the verification logic at the end of a search period.

3.1.3 Stage of the Proposed Verification Logic

As mentioned in section 2.1.1, the received signal *r(t)* is first filtered by means of a matched filter, i.e. the pulse shaping filter, before being sampled at an oversampling rate and converted to a digital stream. The filtering process combined with the sampling results in spilling the energy of the multipath components over a few samples, producing some energy lobes that typically spread over 4 to 7 samples when χ is equal to 4. A sample in the energy lobe be detected as a path that is in fact not existing lead to a false alarm.

Figure 3.9 presents an example of a channel with closely spaced multipath components separated by only one chip interval at offsets 1 and 5 (highlighted in bold). In the example, pulse shaping and oversampling are used, and some energy lobes appear around the multipath locations. The contributions of the two energy lobes are added

between the two paths, and a peak with energy greater than the threshold appears at offset 3. In this work, a third stage called the verification logic has been added. This is a new feature in comparison with existing double-dwell algorithms [19]-[20]. Its role consists in verifying that the combined effects of filtering and sampling do not create some valid paths where none exist.

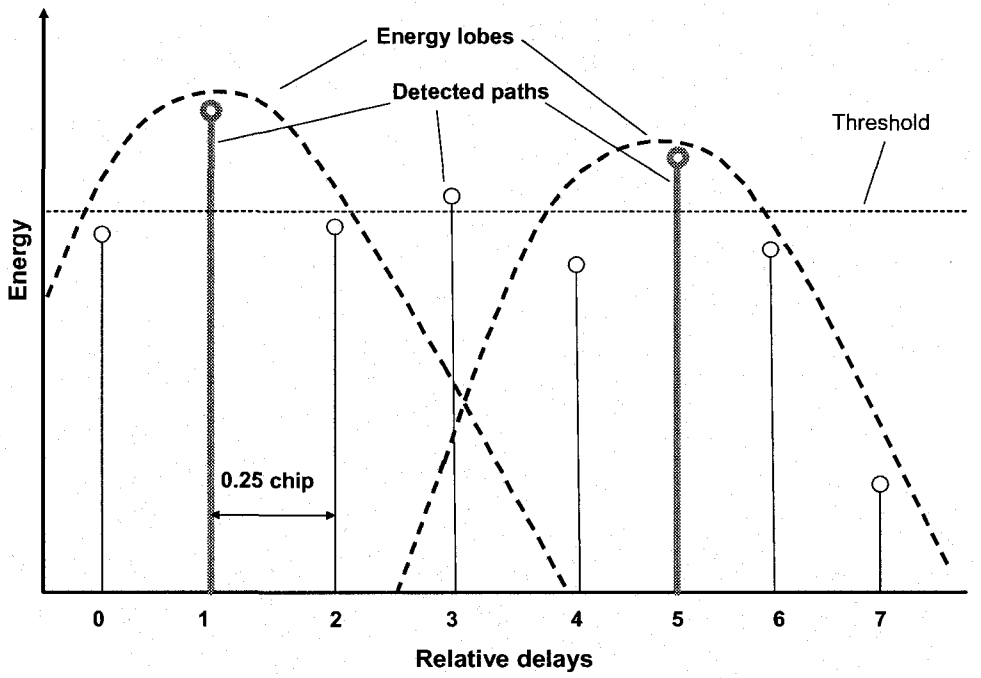


Figure 3.9 Energy lobes caused by the filtering and sampling.

3.1.3.1 Principle of the Verification Logic

The verification logic is based on three principles:

1- Remove all the weak energy peaks located within ±2 samples of a local maximum.

2- If the interval between 2 local maximums is of more than ±8 samples: remove also the energy peaks of smaller value located at ±3 samples of a local maximum.

3- A post processing verification is needed to remove any path located at ±2 samples of each other; the weaker peaks are removed.

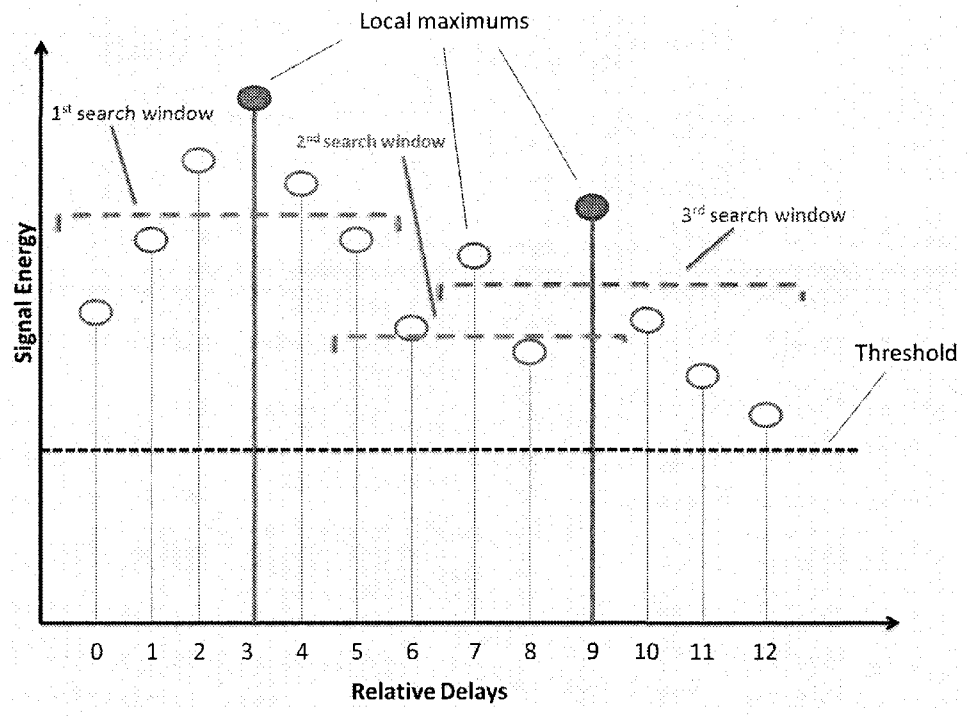
In order to be considered as a local maximum, a given delay must have greater energy than its two immediate neighbours.

The following example illustrates the principle of the proposed verification logic. A typical energy-delay profile for a channel with closely spaced multipath components at relative delays 3 and 9 is illustrated in Figure 3.10 (a). In the figure, the actual paths are highlighted in bold. Some correlation peaks caused by filtering and sampling have energy greater than the threshold and are identified as valid paths at offsets 0, 1, 2, 4, 5, 6, 7, 8, 10, 11 and 12.

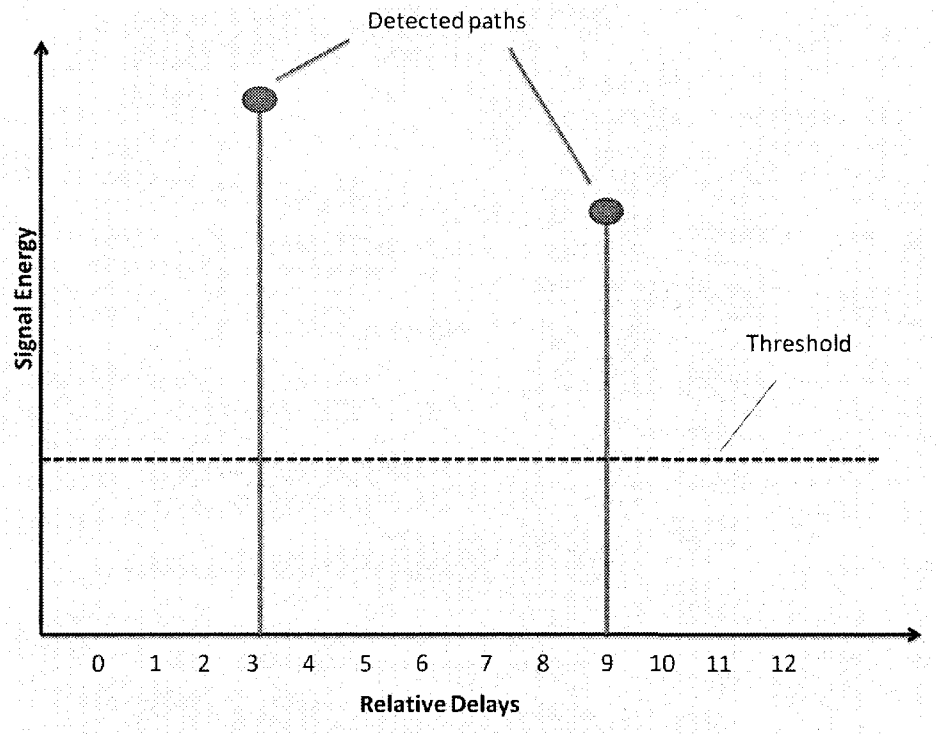
In Figure 3.10 (a), three local maximums appear at offsets 3, 7 and 9; only those located at offsets 3 and 9 are actual paths (highlighted in bold). Therefore, the verification stage would identify 3 search windows represented by brackets in the figure. Since the interval in between the local maximums is of less than 8 samples, the different search windows have the following values:

- 1^{st} search window: [-3; +2] samples,
- 2^{nd} search window: [-2; +2] samples,
- 3^{rd} search window: [-2; +3] samples.

After applying the verification rules, many correlation peaks are removed and the energy-delay profile of Figure 3.10 (b) is obtained. In this particular example, the local maximum at offset 7 is removed because it is located at 2 samples of a stronger energy peak, i.e. the one of offset 9. Even though the proposed verification logic cannot eliminate completely the occurrence of false alarms, it produces accurate results for most channel conditions.



(a)



(b)

Figure 3.10 Energy-delay profile a) Before and b) After the verification stage.

3.1.3.2 Verification Logic Algorithm

Before translating the above principles into a flowchart, the main steps of the algorithm have to be identified. Following is a list of the principal steps of the verification logic:

1- Verify all the candidates of the vector D_d and identify the local maximums. The four cases shown in figure 3.11 result in the identification of a local maximum. These four situations can be summarized as follows: identify the candidate under verification as a local maximum if it has greater energy than its two immediate neighbours (Case 1), or if it has only one immediate neighbour with smaller energy (Cases 2 and 3), or if it has no immediate neighbour at all (Case 4). In turn, this can be expressed under the form of a flowchart as in Figure 3.12, where $D_d[x]$ and $E_d[x]$ are the delay and the energy of the x^{th} element in the vector.

2- Verify all energy peaks located in the interval of [-3; +3] samples from a local maximum. The flowchart of figure 3.13 illustrates this procedure. In case where the interval between 2 local maximums is of less than ±8 samples, do not verify the peaks located at ±3 samples. Identify all peaks with smaller energy and write their indexes in the vector *Delete_value*. Finally, remove in the vector D_d the candidates whose indexes figure in the vector *Delete_value* and go to the post processing verification.

3- A last step, called the post processing verification, is needed to verify that there is at least an interval of 3 samples between subsequent delays in the vector D_d. The flowchart of figure 3.14 illustrates this procedure. In the case where the 3 samples interval is not respected, the weaker peaks are removed. It is important to mention that a candidate that has already been marked as a value to delete cannot remove other paths. Finally, the candidates whose indexes figure in the vector *Delete_value* are removed in the vector D_d.

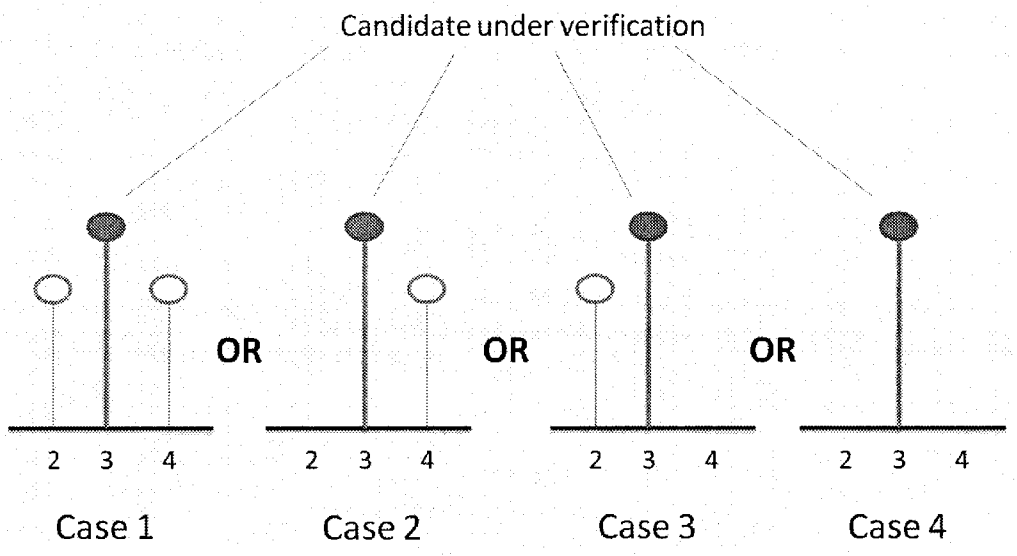


Figure 3.11 Identification of a local maximum.

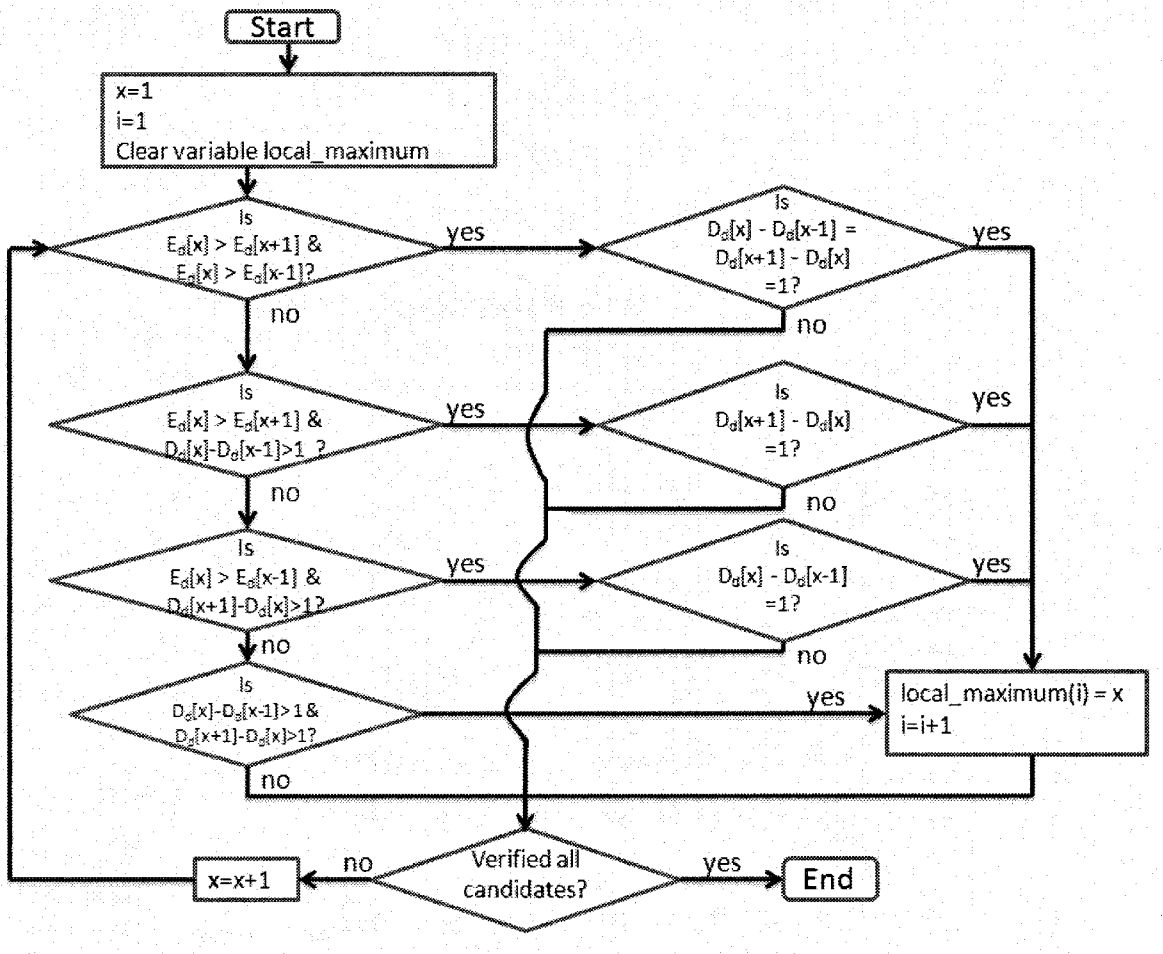


Figure 3.12 Flowchart of the identification of a local maximum.

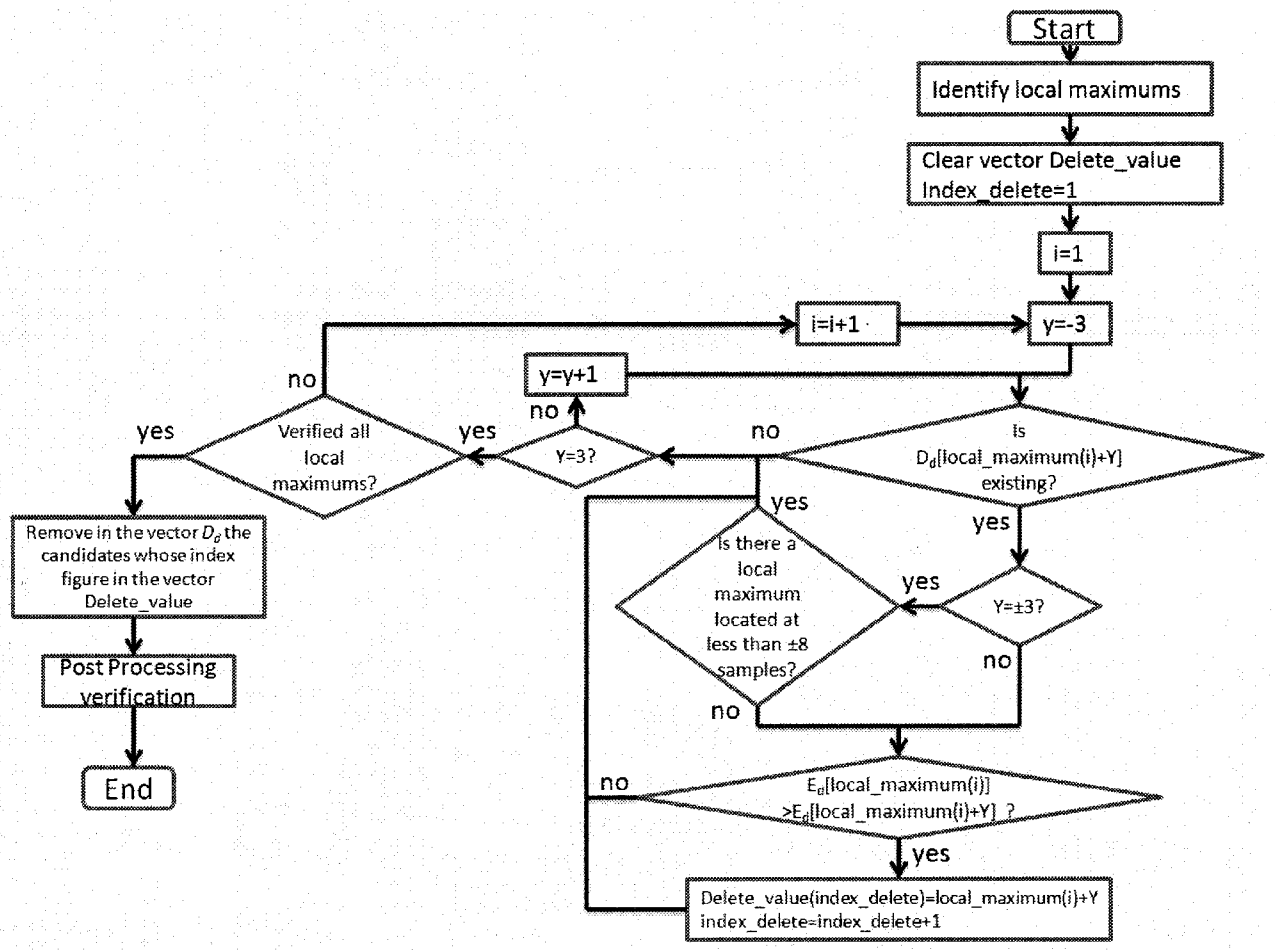


Figure 3.13 Flowchart of the verification logic.

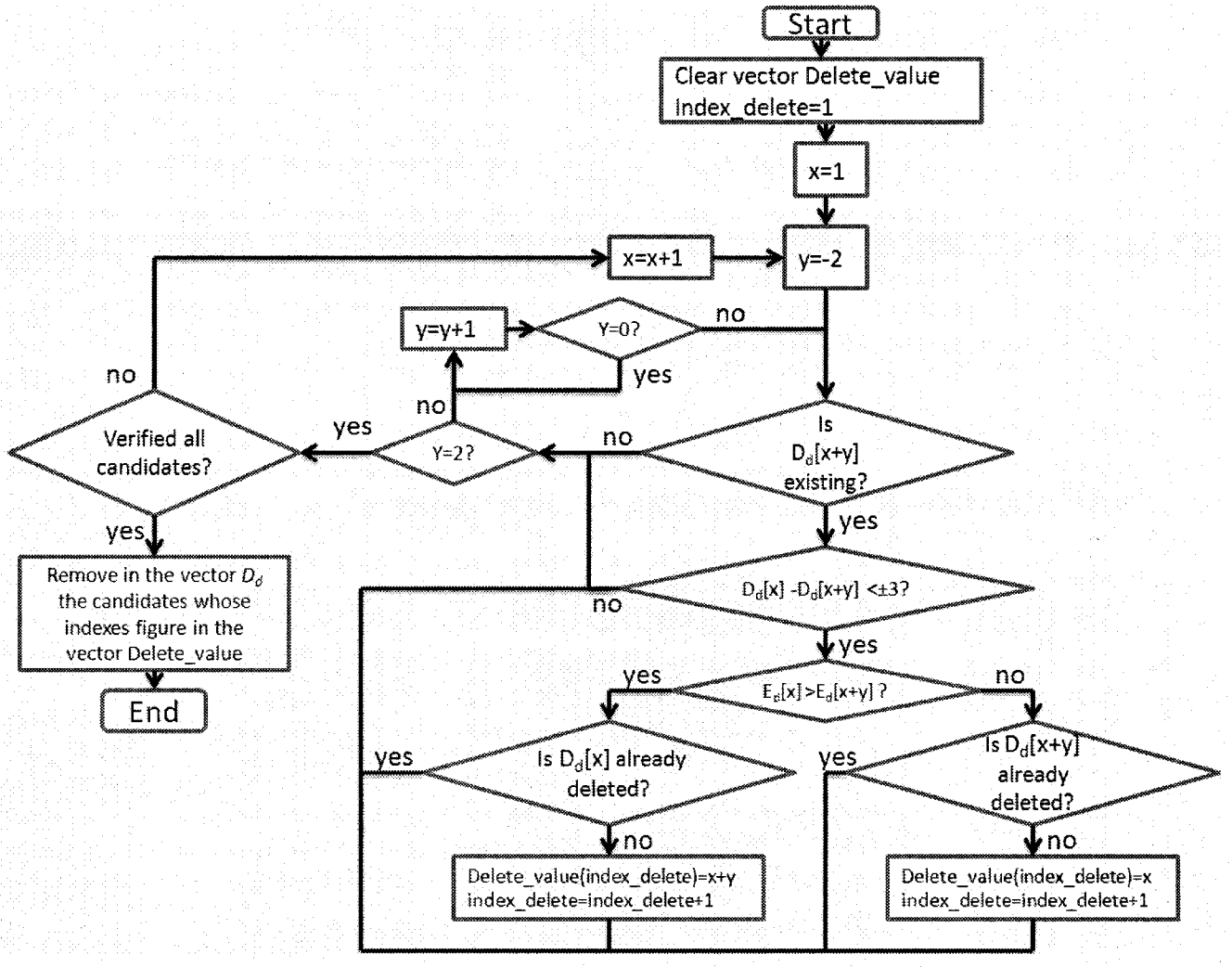


Figure 3.14 Flowchart of the post processing verification.

3.2 Computational Requirements of the Proposed Searcher

This sub-section illustrates how a single complex multiplier-accumulator can be used to perform the calculations related to different dwells. Then, the computational requirements of the proposed multipath searcher will be determined.

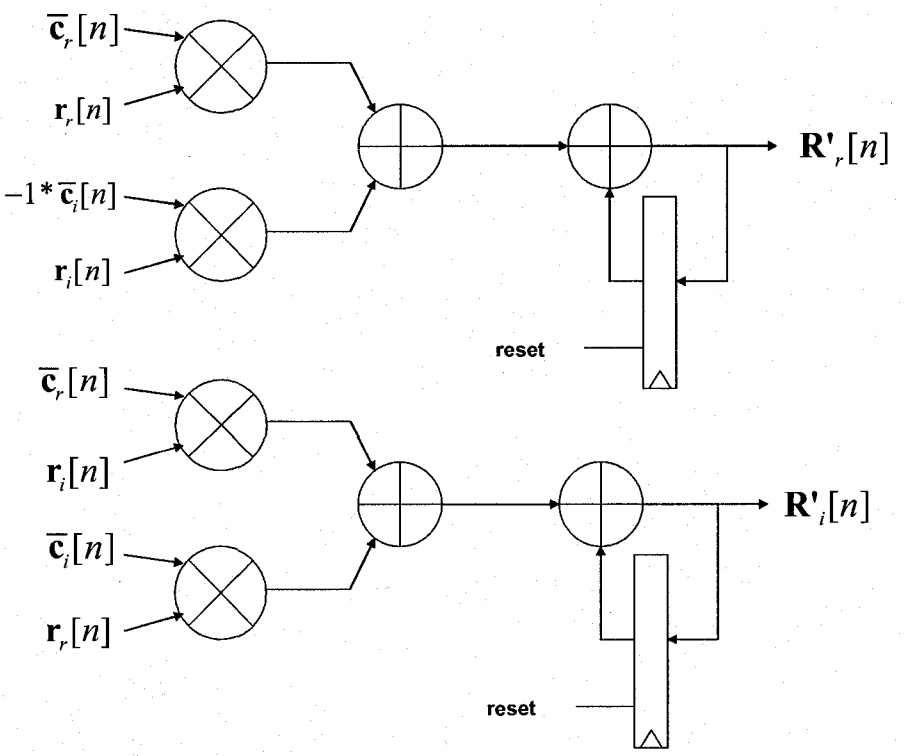
Before we go any further, an overview of the CMAC architecture is presented. Basically, the complex mutiplier-accumulator can perform two arithmetic operations:

1- Correlation between the complex scrambling code $\overline{\mathbf{c}}[n]$ and the I-Q components of the incoming signal $\mathbf{r}[n]$,

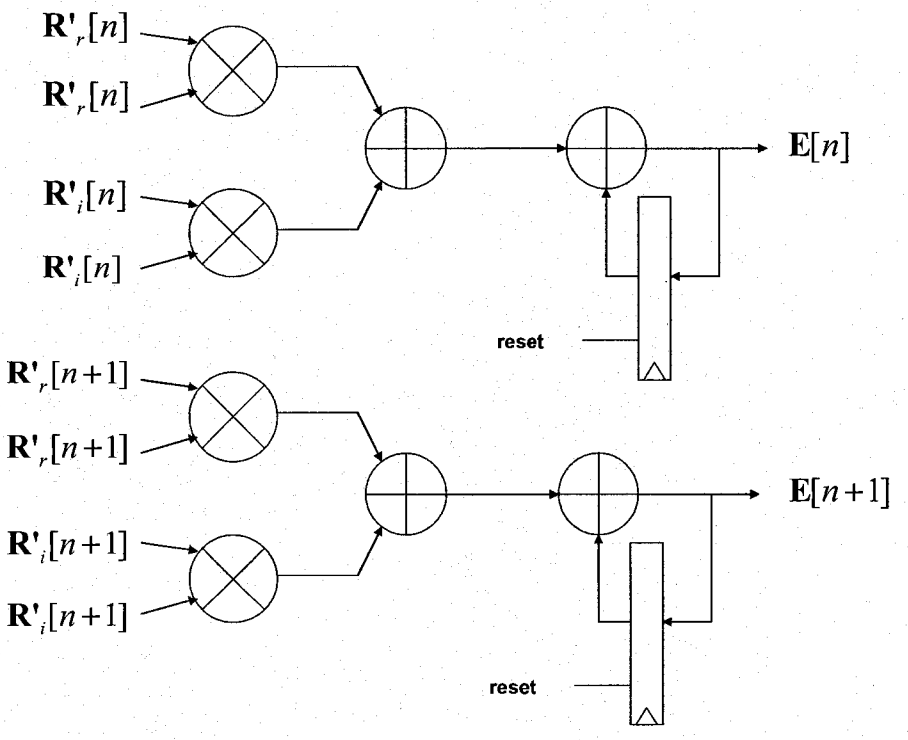
2- Computation and averaging of the correlation energies ($I^2 + Q^2$).

Figure 3.15 (a) illustrates the schematic of the CMAC in the correlation mode. In the figure, the subscripts *r* and *i* stand for the real and imaginary component of the different matrices. The *RESET* signal is only asserted after *N'* clock cycles, once the correlation is completed. Figure 3.15 (b) presents the schematic of the CMAC in the 2^{nd} mode of operation. Since only one half of the CMAC is needed to perform the operation $I^2 + Q^2$, two calculations can be done simultaneously. In Figure 3.15 (b), two rows of the matrix $\mathbf{E}[n]$ are calculated in parallel, reducing by a factor of two the number of clock cycles needed to conclude the operation. The time interval between subsequent *RESET* assertions in the 2^{nd} mode depends on the calculation to be performed.

In order be able to process the data at a faster rate than the chip rate of 3.84 MHz, the multipath searcher must store in memory the data related to one search period. Thus, there is an extra latency of 5 slots at the beginning of the process before the CMAC can start the calculations.



(a)



(b)

Figure 3.15 a) Mode 1 - correlation, b) Mode 2 – energy calculation.

In the case where the clock frequency of the CMAC is faster than the one of the control logic, some circular First-In First-Out (FIFO) queues act as buffer between the two clock domains. The FIFOs are read at the clock speed of the CMAC while they are written at the clock frequency of the controlling logic. The utilization of the CMAC must be optimized to avoid hardware duplication and the pauses between different accumulations must be minimized. In general, these pauses are dued to the time that is needed to write the FIFOs. It is important to make sure that the writing capacity of the control logic matches the reading rate of the CMAC.

It is observed that a total of 5 different vectors are needed in the 1st mode of operation ($\overline{\mathbf{c}}_r[n], \overline{\mathbf{c}}_i[n], \mathbf{r}_r[n], \mathbf{r}_i[n], -1*\overline{\mathbf{c}}_i[n]$) while 4 different vectors are necessary in the 2nd mode ($\mathbf{R'}_r[n], \mathbf{R'}_i[n], \mathbf{R'}_r[n+1], \mathbf{R'}_i[n+1]$). Therefore, 5 FIFOs are used and a demultiplexer is responsible for routing the vectors to the right input as illustrated in Figure 3.16. Depending on the mode of operation, the demultiplexer makes the connection accordingly to the truth table presented in Table 3.2.

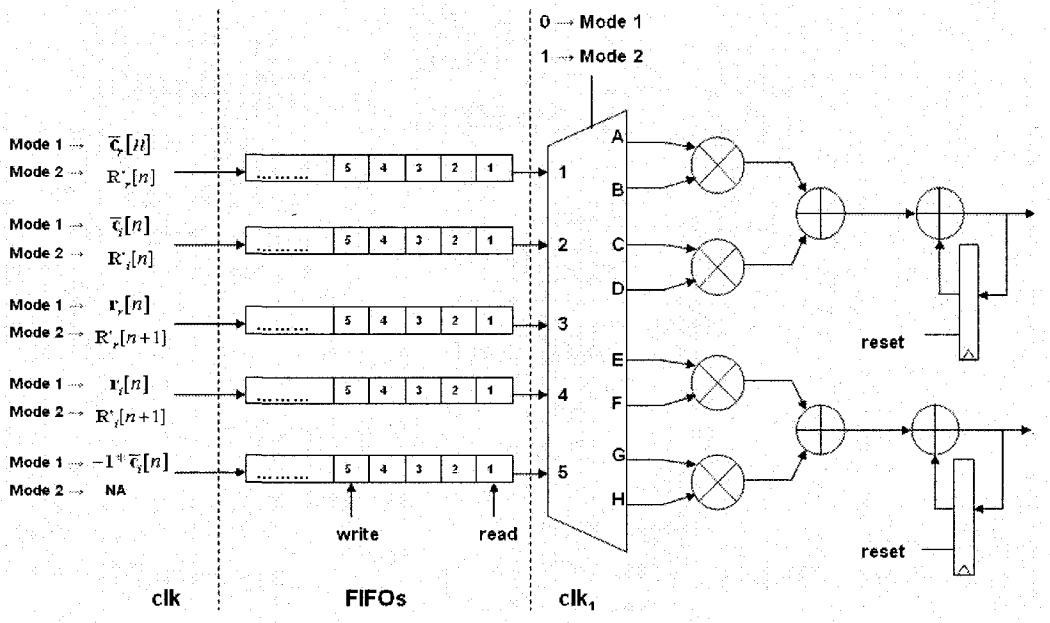


Figure 3.16 Inputs of the CMAC.

Output name	Mode of operation	Output value
A	1	1
	2	1
B	1	3
	2	1
C	1	5
	2	2
D	1	4
	2	2
E	1	1
	2	3
F	1	4
	2	3
G	1	2
	2	4
H	1	3
	2	4

Table 3.2 Truth table of the demultiplexer.

The computational requirements of the proposed implementation are summarized in Tables 3.3 and 3.4. In the tables, the mode of operation of the CMAC appears in the third column. It is important to mention that only the two first stages of the algorithm require the use of the CMAC.

Operation	# of CMAC cycles/5 slots	Mode of operation
Calculation of matrix $\mathbf{R}'[n]$	768×512	1
Calculation of matrix $\mathbf{R}'[n']$	768×512	1
Calculation of the noise floor	2×768	2
Calculation of matrix $\mathbf{E}[n]$	768	2
Total	788 736	

Table 3.3 Computational requirement of the 1st dwell.

Operation	# of CMAC cycles/5 slots	Mode of operation
Calculation of matrix $\mathbf{R}_v'[n']$	36×1024×5	1
Calculation of vector $E_v[n']$	5×18	2
Total	184 410	

Table 3.4 Computational requirement of the 2nd dwell.

The computational requirements of the 1st and the 2nd dwell result in a total of 973 146 CMAC cycles per 5 slots. Therefore, a total of 2 919 438 CMAC cycles are required per frame. Considering a rate of 100 frames per second in WCDMA systems, a speed of 291 943 800 CMAC cycles/sec is needed to implement the proposed multipath searcher. The implementation of the searcher based on a processing rate of 3.84 MHz, i.e. the chip rate, would normally require 77 CMACs resulting in complex hardware architecture.

3.3 Design of a High-Speed CMAC Using the Wave Pipeline Method

The computation of the algorithm shown in Figure 3.1 is executed mostly by means of CMAC cells. As mentioned previously, the target of the algorithm implementation is to make the computation to be completed within a short period of 5 slots, and thus the circuit cells need to be able to operate with short delay. Also, it is expected to deal with high resolution data, requiring a high speed process. In order to have a simple structure, this design is to use only one CMAC cell to perform the double-dwell computation without doubling the processing time. Thus, the speed of the CMAC cell to be developed needs to double.

Figure 3.17 recalls the general scheme of a multipath searcher where Ω CMACs operate in parallel for real-time processing. Moreover, some extra hardware represented by the symbol $|\bullet|^2$ in the figure, is necessary to compute the correlation energies, i.e. $(I^2 + Q^2)$.

In the design presented in this thesis, a clock rate of the searching system of 3.84 MHz would require the use of 77 CMACs, resulting in complex hardware architecture. If instead of 77 CMAC cells, a single high-speed CMAC is used to perform the same amount of calculations for the circuit structure to be simple; this CMAC cell has to be made to operate at a speed of at least 292 MHz.

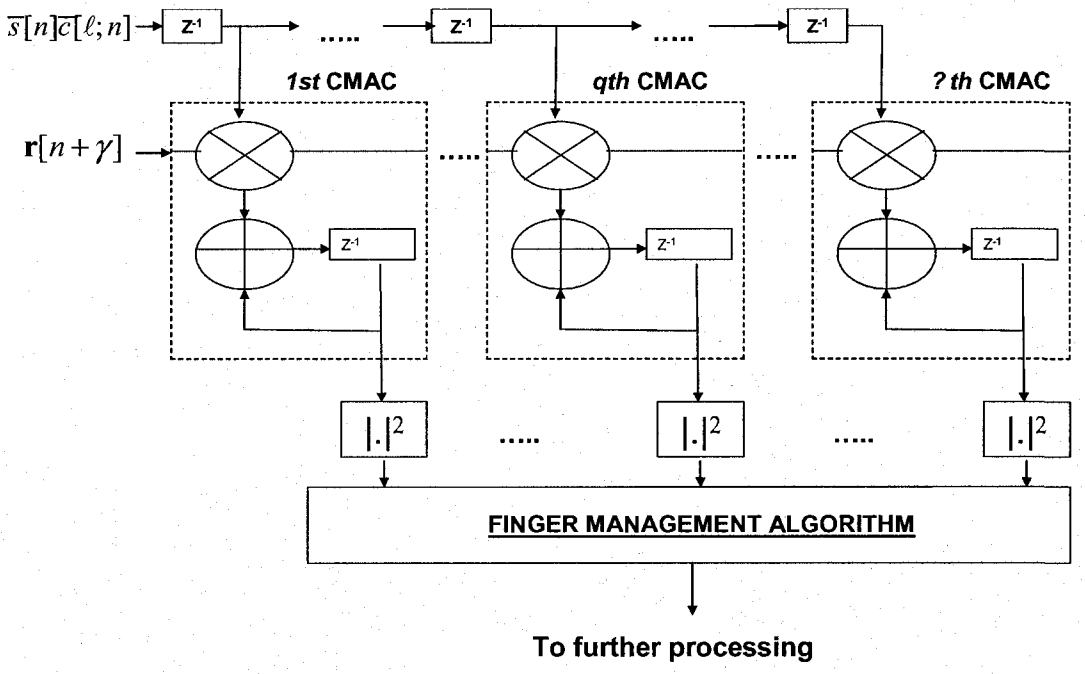


Figure 3.17 General scheme of the coarse acquisition.

The two first steps for synchronizing a digital circuit are (i) the determination of the hardware architecture and (ii) the choice of a logic family. In this section, the different design steps of the CMAC will be presented following a top-down approach, but it is important to mention that low level design choices also have repercussions on higher levels of abstraction.

The wave pipeline technique has been used for the development of a high performance complex multiplier-accumulator. A wave pipeline circuit does not need a clock for synchronization. The operations in the circuit are synchronized by means of equalized path delays. Thus, its main advantage beneficial for this work is that it does not

suffer from clock skew because all of the data and control signals go through the same depth. If all the circuit units are designed to have short delay, whitout clock skew, the total of the circuit can be minimized easily. Also, the wave pipeline can help to reduce the switching activity and the circuit area because there is no clock tree.

It should be mentioned that to apply the wave pipeline technique, the circuit should be regular-structured and its basic blocks be grouped with equalized delay paths easily. A CMAC, of which the diagram is shown in Figure 3.18, has a structure of regular arranged building blocks. Thus, it is suitable to the wave pipeline method.

To complete the multiplication of two complex numbers, four multipliers are required as shown in the figure. The products are then added two by two to generate the real and imaginary components. Therefore, the multiplier is the most important block and it is the topic of the next section.

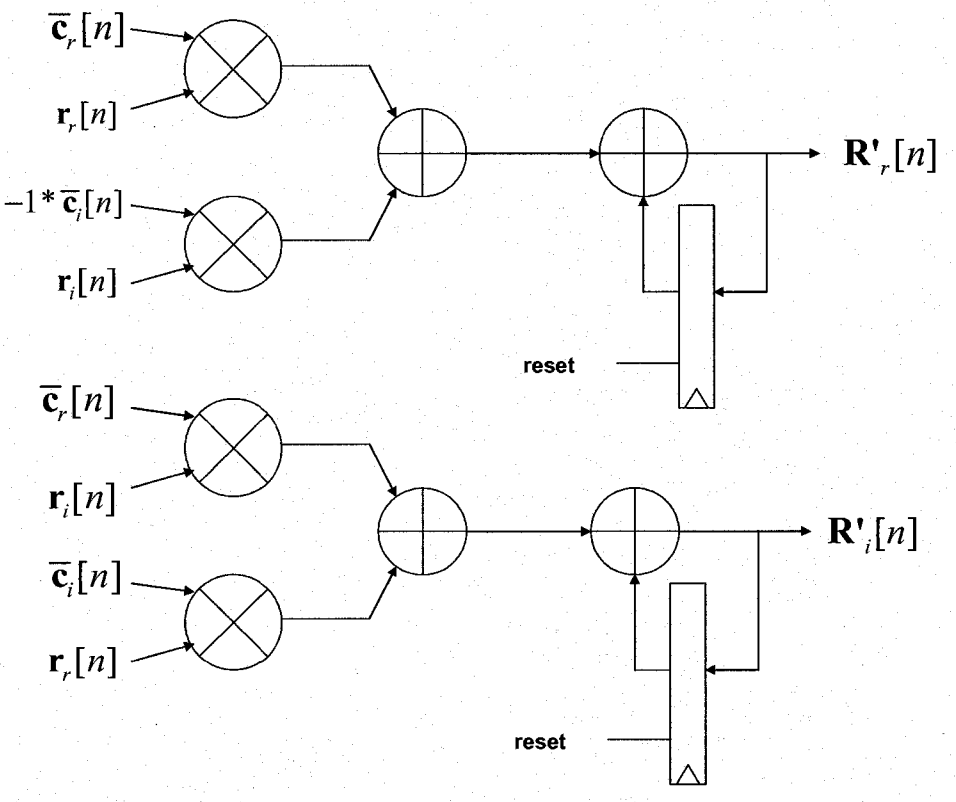


Figure 3.18 General scheme of a CMAC.

3.3.1 Wave Pipelined Multiplier

The choice of the multiplier architecture is determined by the fan-in constraints of two logic gates imposed by the CPL-TG and the NPCPL families (section 2.3.4). We estimate that a Booth multiplier would require a combinational area of $17\times(word\ length)^2$ two-input gates as opposed to $9.5\times(word\ length)^2$ two-input gates for an array multiplier.

In order to reduce the area and the logic depth, an unsigned array multiplier has been implemented using the wave pipeline method. The inputs of the multiplier are quantized on 8 bits in a sign-modulus format and the CSA array only processes the modulus. The sign is generated separately using a XOR gate. This results in a reduction of the hardware required because we now consider two inputs of 7 bits. The CSA tree will perform the multiplication accordingly to the principle shown in Figure 3.19, where X represents the multiplier, Y is the multiplicand, U represents the partial sums and V the partial carries.

$$
\begin{array}{rl}
 & X_0(0\ \ Y_6\ \ Y_5\ \ Y_4\ \ Y_3\ \ Y_2\ \ Y_1\ \ Y_0) \\
 & +\ X_1(Y_6\ \ Y_5\ \ Y_4\ \ Y_3\ \ Y_2\ \ Y_1\ \ Y_0) \\
\hline
 & U_{07}\ \ U_{06}\ U_{05}\ \ U_{04}\ U_{03}\ U_{02}\ U_{01}\ \mathbf{U_{00}} \\
+ & \ \ V_{07}\ V_{06}\ V_{05}\ \ V_{04}\ V_{03}\ V_{02}\ V_{01}\ V_{00} \\
+ & X_2\ (0\ \ Y_6\ \ Y_5\ \ Y_4\ \ Y_3\ \ Y_2\ \ Y_1\ \ Y_0) \\
\hline
 & U_{17}\ \ U_{16}\ U_{15}\ \ U_{14}\ U_{13}\ U_{12}\ U_{11}\ \mathbf{U_{10}} \\
+ & \ \ V_{17}\ V_{16}\ V_{15}\ \ V_{14}\ V_{13}\ V_{12}\ V_{11}\ V_{10} \\
+ & X_3(0\ \ Y_6\ \ Y_5\ \ Y_4\ \ Y_3\ \ Y_2\ \ Y_1\ \ Y_0) \\
\hline
 & \cdots\cdots\cdots\cdots
\end{array}
$$

Figure 3.19 CSA array principle.

The processing element of the CSA tree is basically a full adder to which an AND gate has been added. The pins *X* and *Y* (Figure 3.20) take as inputs some bits from the multiplier and the multiplicand. Some sum and carry (*U, V)* generated by a previous stage are applied to the pins labelled *A* and *B*. The AND gate performs the bit-wise multiplication of the multiplier and the multiplicand. In the actual circuit, three buffers have been added [21]. They do not interfere with the functionality of the cell; their role is only to add some delay in certain logic paths to synchronize the different signals. They alter the propagation time of the multiplicand *Y* through the different stages. The input signal *A* is also buffered so it reaches the CARRY block at the same time as the output signal from the XOR gate.

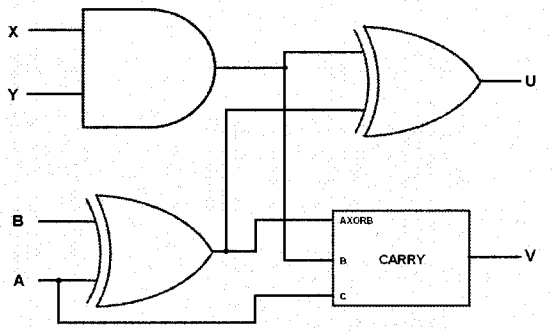


Figure 3.20 Processing element of the CSA tree.

In order to be able to accumulate the different results with a simple structure in the CMAC, the output of the multiplier has to be in two's complement. Since we use the sign-modulus format, the first step to make the conversion is to generate the complement of the bits if the result is negative. This is achieved by the 1's complement units in Figure 3.21.

Two adders are used in the architecture as presented in Figure 3.21. The first adder performs the final summation. The role of the second adder is to convert the data format of the result, i.e., the 1's complement, into the two's complement by adding the sign to it. The LSBs (highlighted characters in Figure 3.19) are not discarded for accuracy

of the result; they are sent via buffers to the second adder. Some ripple-carry adders (RCAs) have been chosen because of the nature of the data arrival; the LSBs are generated at different logic depths in the architecture, so that they already have some time skews.

Some buffers, illustrated by Δt_2, Δt_3, Δt_4 and Δt_5 on Figure 3.21, are used to synchronize the propagation of the sign so that the multiplier can operate at any frequency up to f_{max}. Finally, a first sign extension is performed and the output is represented on 16 bits, instead of 15. This is needed to avoid overflow when the products are summed in the CMAC.

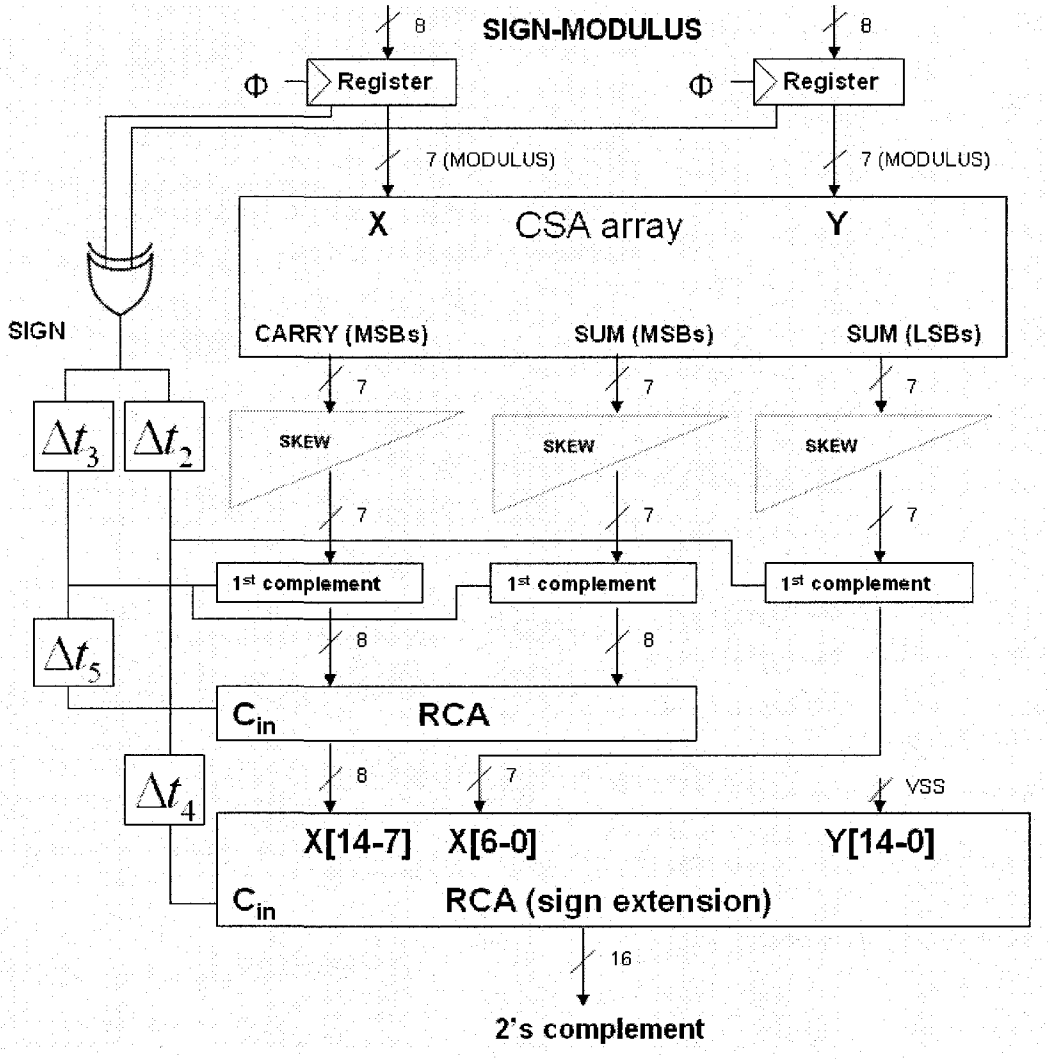


Figure 3.21 Functional schematic of one multiplier.

3.3.2 Wave Pipelined CMAC

The architecture of the wave pipelined CMAC is presented in Figure 3.22. In the figure, the four multipliers correspond to the wave pipelined multiplier described in the previous section. The sign extension performed at the output of the multiplier was intended to avoid overflow when summing the products.

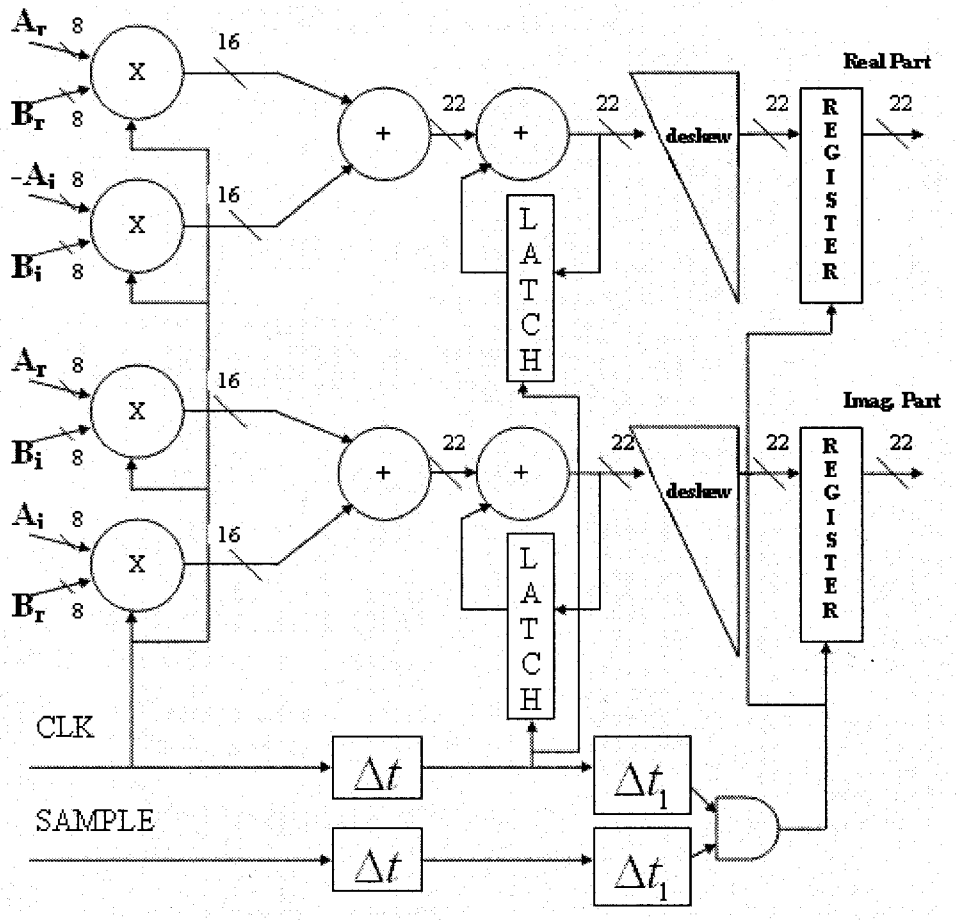


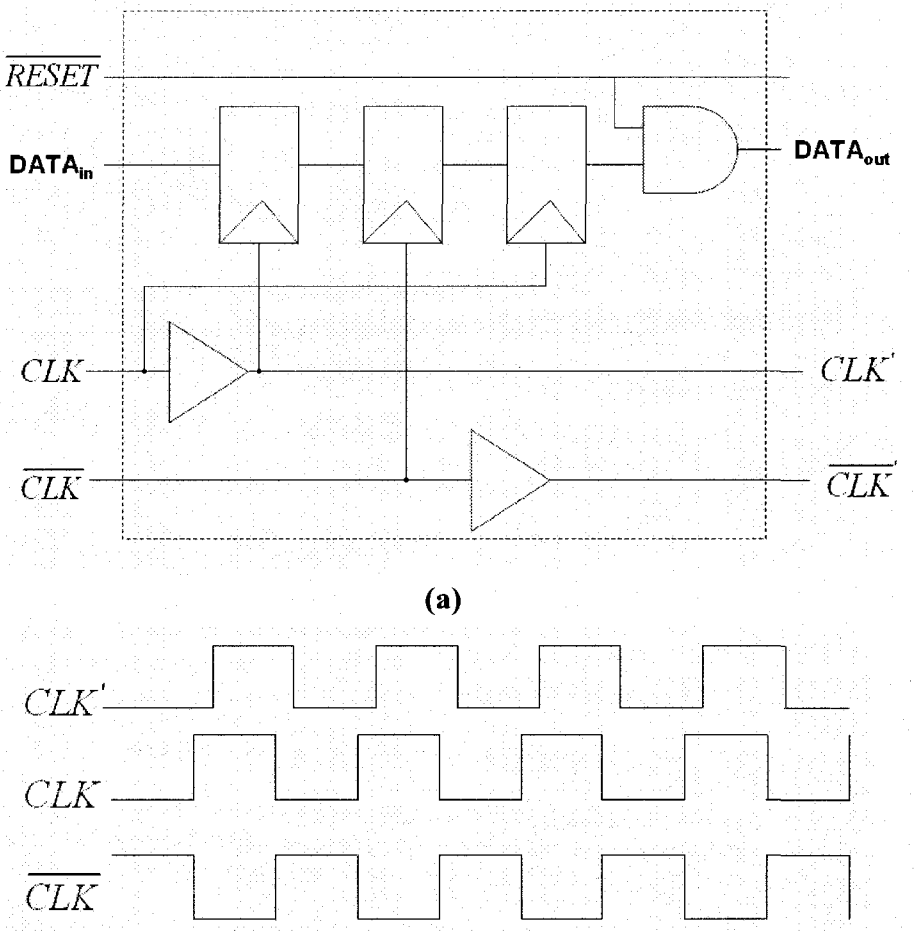
Figure 3.22 Architecture of the wave pipelined CMAC.

A model of the CMAC at the gate level has been developed using Matlab® and incorporated into the multipath searcher model, with which we could observe the effect of the proposed arithmetic units in simulations of the overall communication system.

From these simulations, it is observed that a sign extension of six bits is sufficient to make the accumulation without facing any wrap around phenomenons. This second sign extension is performed before the accumulation, at the output of the first row of adders in Fig. 3.22. Some RCAs are also used in the CMAC because the different bits have already got some skew.

Some buffers, represented by *deskew* triangles in Fig. 3.22, are used to uniform the skews in order to re-synchronize the data at the output of the structure. The *SAMPLE* signal allows for different accumulation lengths. The two control signals (*CLK* and *SAMPLE*) get additional skew, represented by Δt and Δt_1 in the figure, so that they are in phase with the flow of the data and the CMAC works properly at any frequency up to $f_{max.}$

The accumulation is executed bit by bit because of the delay of the data due to RCA adders. Therefore, the latch of Figure 3.22 must have a special architecture for a proper synchronization. Figure 3.23 (a) shows the building block of the accumulation latch. The principle is that the output of the adder ($DATA_{in}$ in the figure) can only be re-introduced into the adder ($DATA_{out}$) at the next clock cycle. Therefore, three D flip-flops (DFFs) are used with the three phases clocking scheme presented in Figure 3.23 (b). The first flip-flop is clocked with the late version of *CLK* to make sure that the signal does not propagate through the latch within less than one clock period. The second flip-flop samples the signal during the second half of the clock period while the third DFF re-introduces the signal into the adder at the next clock cycle. The delayed version of *CLK* and $\overline{CLK}$ are propagated to the next bit. An asynchronous *RESET* is used to control the different accumulations lengths.



(a)

(b)

Figure 3.23 a) Processing element of the accumulation module, b) Clocking scheme.

3.3.1 Standard Cells and their Impact on the Fine Tuning

In the literature, the NPCPL logic is the most widely used family for the design of wave pipeline circuits in ASIC ([21], [29], [30] and [31]). The NPCPL cell, illustrated in Figure 3.24, is a dual-rail logic block that takes as inputs all of the variables and their complement and that produces the output and its complement. Different logic functions can be implemented by modifying the input connections of the pass transistor network as shown in Table 3.5. The main advantage of the NPCPL logic is that a single block can implement most of the commonly used logic block; allowing for fine tuning as well as for capacitance equalization. However, as the voltage is scaled down, the NMOS pass

transistors have more difficulties in transferring logic '1' so that the internal nodes (see Figure 3.24) are difficult to rise to high level. Therefore, some significant variations between the rise time and the fall time of a given gate are observed. Since the NPCPL is a dual-rail logic block, it is mandatory that the two outputs Q and $\overline{Q}$ are synchronized for a proper operation of the wave pipeline circuit. The accumulation of delay uncertainties can cause some speed degradations, especially when the logic depth of the circuit increases. In this project, the impact of the logic family on the performance of the wave pipelined CMAC is investigated.

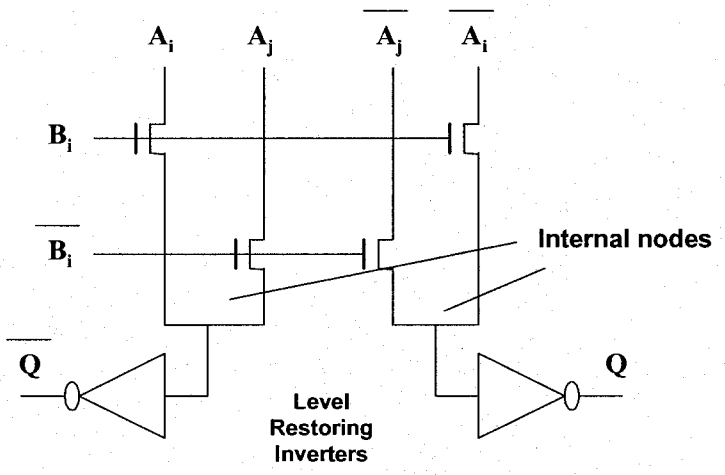


Figure 3.24 Schematic of the NPCPL cell.

Logic function	Pin label				
	A_i	A_j	B_i	Q	Q_{inv}
$AND / \overline{AND}$	A	B	B	AB	$\overline{AB}$
$OR / \overline{OR}$	A	B	$\overline{B}$	$A+B$	$\overline{A+B}$
$XOR / \overline{XOR}$	A	$\overline{A}$	$\overline{B}$	$A \oplus B$	$\overline{A \oplus B}$
$CARRY$	C	B	$A \oplus B$	$carry$	$\overline{carry}$
SUM	C	$\overline{C}$	$A \oplus B$	$A \oplus B \oplus C$	$\overline{A \oplus B \oplus C}$
$BUFFER$	$B(t)$	$B(t)$	$B(t)$	$B(t+1)$	$\overline{B(t+1)}$
$D\ flip-flop$	D	Q	CLK	Q	$\overline{Q}$

Table 3.5 Realizing different functions with the NPCPL/CPL-TG cell

In an effort to improve the fine tuning of our wave pipelined circuit, we have replaced the NMOS pass transistors by transmission gates; this cell is known in the literature as Complementary Pass-Logic Transmission-Gate (CPL-TG) [32]. Figure 3.25 presents the schematic of a CPL-TG gate. With this configuration, the internal node can benefit from the full supply voltage swing. To minimize the capacitance, the switches of the CPL-TG as well as the pass transistors of the NPCPL are sized with the ratio W/L = 0.5µm/0.18µm in a 0.18 µm technology. The level restoring inverters have been designed not to have a centered transfer characteristic at VDD/2, but to ensure the smallest delay variation between the rise time and fall time. With HSPICE™ simulations, it is found that a size of W/L = 2.5µm/0.18µm for the PMOS and a size of W/L = 2µm/0.18µm for the NMOS allow for a proper operation of the different cells. The layout views of the NPCPL and CPL-TG are presented in Figure 3.26. The schematics and layouts of the different logic gates have been designed using Virtuoso™ from Cadence®. As one can observe on Figure 3.26, the output inverters are located on each side of the cells. The four CMOS switches/NMOS pass transistors are located in the middle of the cell. The two power rails VDD and VSS figure at the top and at the bottom of the cells respectively. These gates occupy a silicon area of 7.65 µm × 9.31 µm for the NPCPL and of 8.42 µm × 9.86 µm for the CPL-TG; some values that are comparable to the size of other 0.18 µm technology standard cells. These layouts are free of backend verification errors such as Design Rule Checking (DRC) and Layout Versus Schematic (LVS).

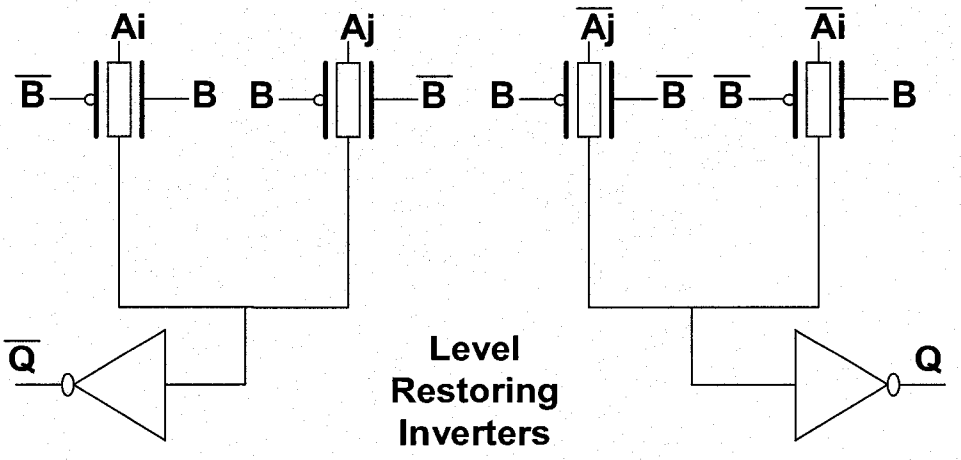
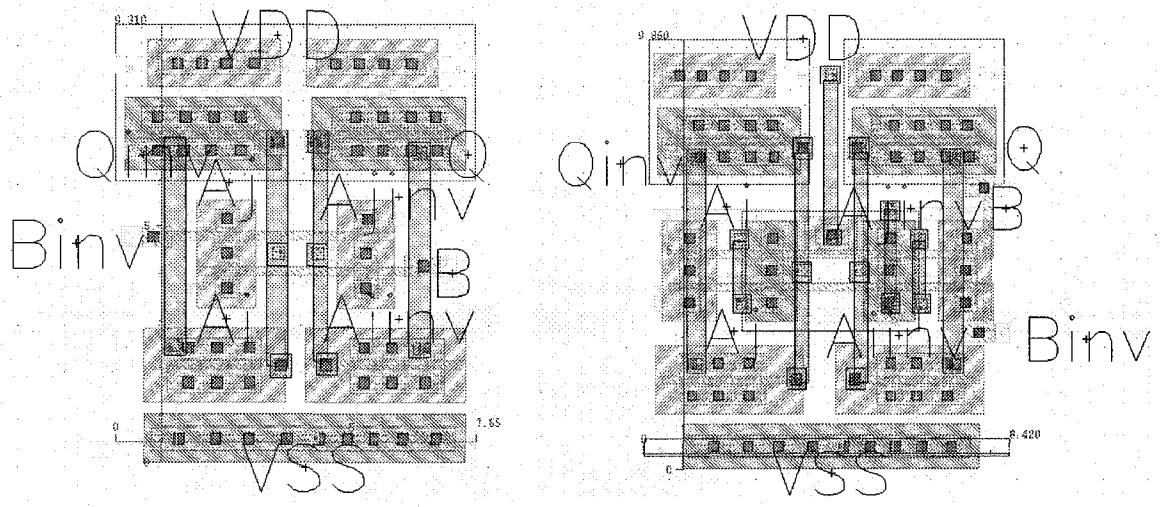


Figure 3.25 Functional schematic of the CPL-TG cell.



(a) **(b)**

Figure 3.26 Virtuoso™ layout of the a) NPCPL cell, b) CPL-TG cell.

The libraries of standard cells (NPCPL and CPL-TG) have been full-custom designed using the input connections presented in Table 3.5. As one can observe in the table, all of these gates can only accept two inputs at the exception of the CARRY function. This fan-in constraint influences the choice of the arithmetic units' architectures. Also, the level restoring inverters have small geometric sizes to minimize the capacitance. As a result, the fan-out supported by the different cells is small and every gate cannot feed more than two cells in order to ensure fast transitions.

The CPL-TG logic family also presents some similitude with the wave pipelined transmission gate logic introduced in [33]-[34], the main difference resides in the fact that a single building block is used to implement all of the logic functions in this work. It is at our knowledge the first time that the CPL-TG logic is used for wave pipelining.

3.4 Summary

The design of a multipath searcher is presented. It is focused on the improvement of the detection quality by reducing the acquisition time and by increasing the resolution of the system. This is done by testing more offsets in a shorter time interval. The probabilities of false alarm and misdetection are minimized by introducing an effective detection scheme of the potential paths, combined with the addition of a third stage called the verification logic.

In the design of the circuit, the focus is on the most important building block; the CMAC. Moreover, we use a high-speed and polyvalent CMAC, so that a single unit can perform all the calculations. In order to increase the speed of the arithmetic units, the wave pipeline technique has been used. Assuming a CMAC processing rate of 3.84 MHz, 77 CMACs would be required for the implementation of the multipath searcher.

The NPCPL and the CPL-TG logics have been used as universal building blocks for the creation of the conventional logic gates. The schematics and layouts of the different logic gates have been designed using Virtuoso™ from Cadence®. The structures of the arithmetic units have been chosen carefully to comply with the precision requirements of the system model while meeting the fan-in/fan-out constraints of the two logic families.

With the circuit structure described in this section, it is expected that the speed of the CMAC is improved significantly, which leads to a high throughput of path detection. The simulation results are presented in chapter 4.

4 Performance Evaluation

This section presents the performance evaluation. This is needed to quantify the advantages and drawbacks of the method described in this work. The circuit level simulation results are presented following a bottom-up approach, because low level design choices influence higher levels of abstraction. For every step, details about the simulation testing procedure will be given. The performances of the proposed method will be compared to those of existing works presented in the literature.

4.1 False Alarms and Misdetection Probabilities

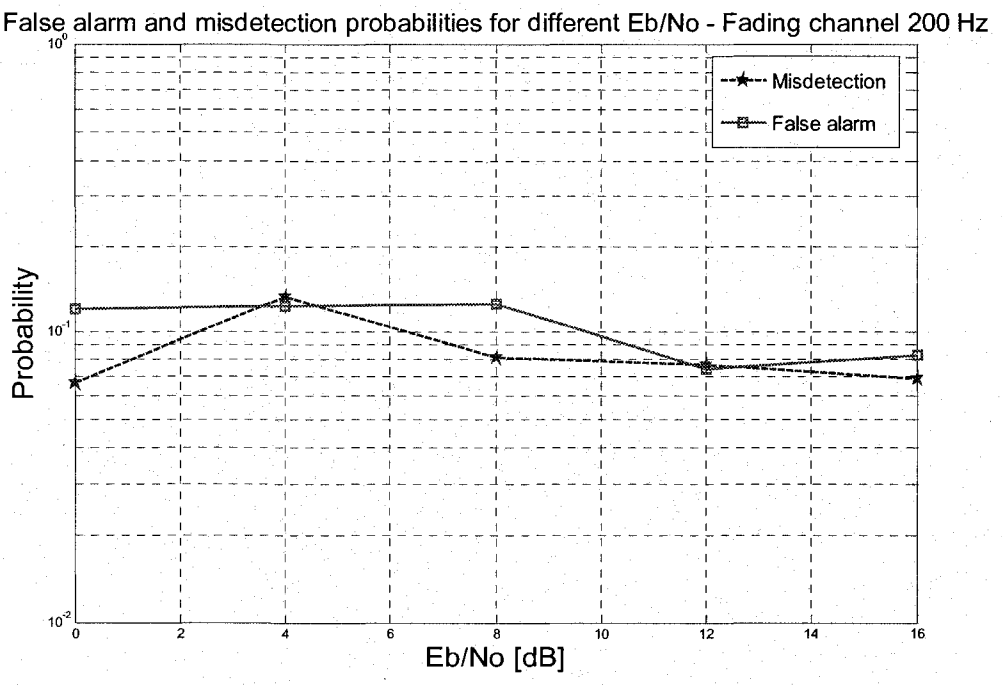
The false alarm and misdetection probabilities of the proposed multipath searcher are plotted in Figures 4.1 and 4.2 for different ratios of Energy per Bit (E_b) to the Spectral Noise Density (N_o) (E_b/N_o). These simulations were realized in Matlab® using a gate level model of the CMAC.

Figure 4.1 (a) illustrates the performance of the multipath searcher for a fading channel with a Doppler frequency f_d of 200 Hz (i.e. $v \approx 120$ Km/h), and 4 time invariant paths of respective power equal to 0, -3, -6 and -9 dB located at offsets [0, 260, 521, 781] µs. As shown in the figure, the probabilities of misdetection and false alarm are constant to approximately 10 % for different SNRs. The false alarm probability is of particular interest since it is the metric targetted by the verification logic. Figure 4.1 (b) illustrates the false alarm probability of the multipath searcher before and after the verification logic stage. The simulation results show that there is an improvement of approximately 55% after the stage of verification logic, demonstrating the effectiveness of the proposed method. The need for verification logic is justified by the high resolution of the analog-

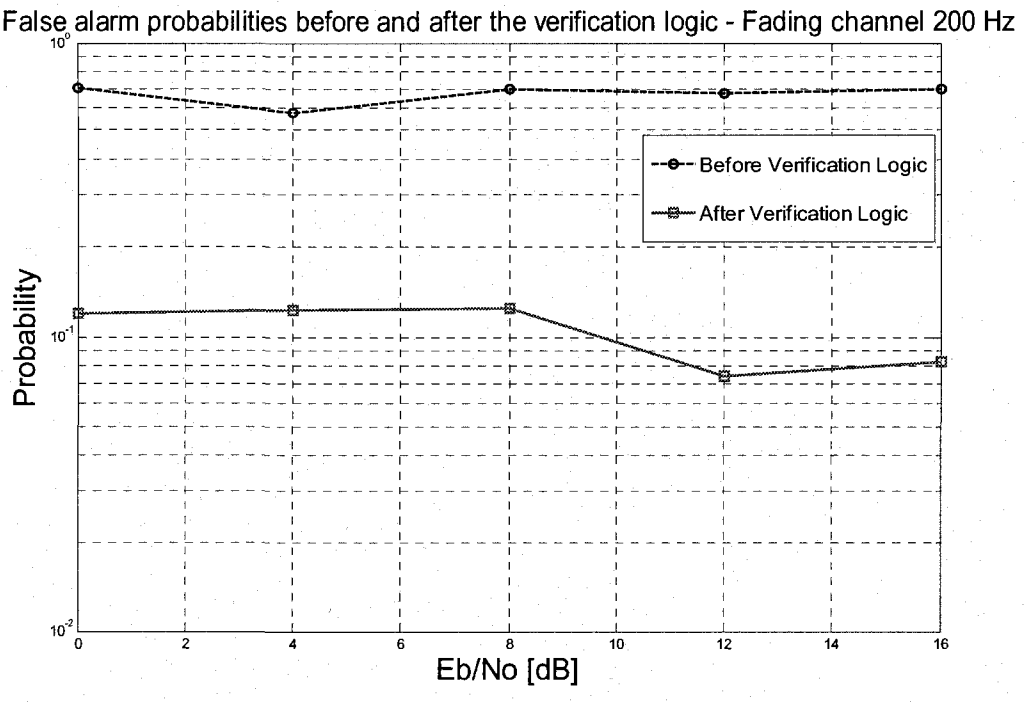
to-digital converter. In fact, the third stage of the proposed searcher is just compensating for a problem that was created when increasing the resolution.

Figure 4.2 shows the different probabilities for a channel with birth and death propagation conditions in which multipath components suddenly appear and disappear. In the test case recommended by the 3GPP standard [44], two paths of equal strength (which is set to -3 dB in the simulation) are chosen randomly from the offset distribution of [-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5] μs. For this particular case, the different probabilities are constant to 2% for all SNRs.

From the above discussion, we can state that the different probabilities are meeting the objectives of section 3.1, and thus, the receiver would be able to meet the requirements of the 3GPP standard in terms of bit-error-rate.



(a)



(b)

Figure 4.1 **(a) False alarm and misdetection probabilities for a fading channel with f_d = 200 Hz.**
(b) False alarm probabilities before and after the stage of verification logic for a fading channel with f_d = 200 Hz.

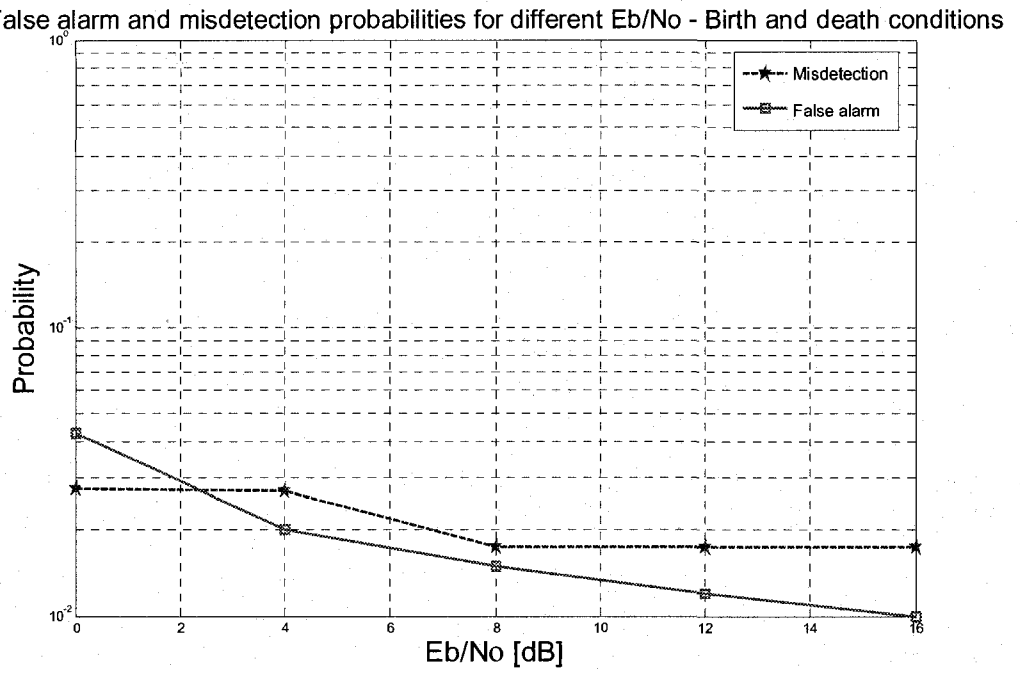


Figure 4.2 False alarm and misdetection probabilities for birth-death propagation conditions.

4.2 Logic Gate Performances

The advantages of the wave pipeline method can be realized only if the logic gates have certain timing properties. The major goal in wave pipelining is to balance the delays of the logic paths from all inputs to all outputs. Therefore, the design of the logic gates is one of the most critical steps. The design choices made at this level of abstraction will have repercussion in term of speed, power consumption and area on the overall system.

Comparison will be made between different logic families in order to have a better understanding of their influence on the performance of wave pipeline circuits. In the wave pipeline literature ([21], [29], [30], [31]), the NPCPL logic has been identified as a universal cell for the creation of conventional logic gates. The principal advantage of this type of logic is that a single building block can implement all of the logic functions. This has for effect of reducing the delay variation between different logic gates. Nevertheless, we will demonstrate that the NPCPL cell also suffers from unequal current paths when using a 0.18 µm technology and a supply voltage of 1.8 V.

In order to measure the delays of the standard cells in the NPCPL library; a proper test-bench had to be elaborated. The device under test (DUT) has been inserted between two NPCPL buffers as illustrated on Figure 4.3, in order to normalize the input drive and the output load. Similarly, the CPL-TG logic gates have been tested with CPL-TG buffers.

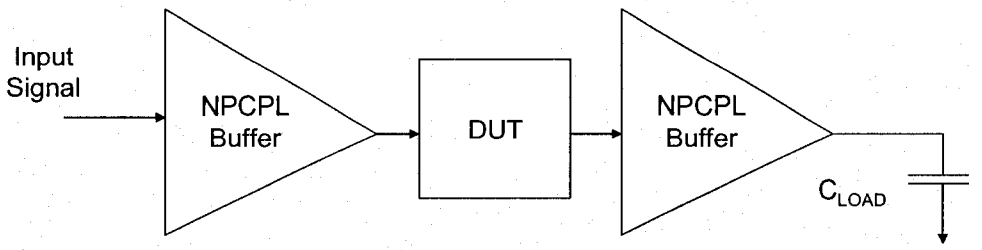


Figure 4.3 Test-bench.

All input patterns have been simulated for every logic gates using HSPICE™ from Synopsys® and for every pattern; the propagation delay has been measured. The propagation delay is defined as the time interval between the input stimulus crossing 50% of *VDD* and the output crossing 50% of *VDD* as illustrated in Figure 4.4.

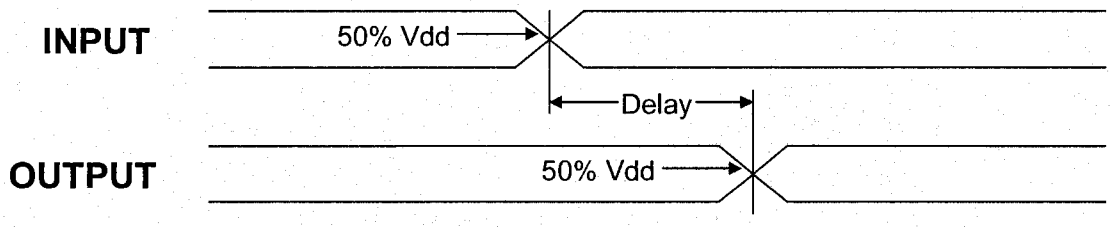
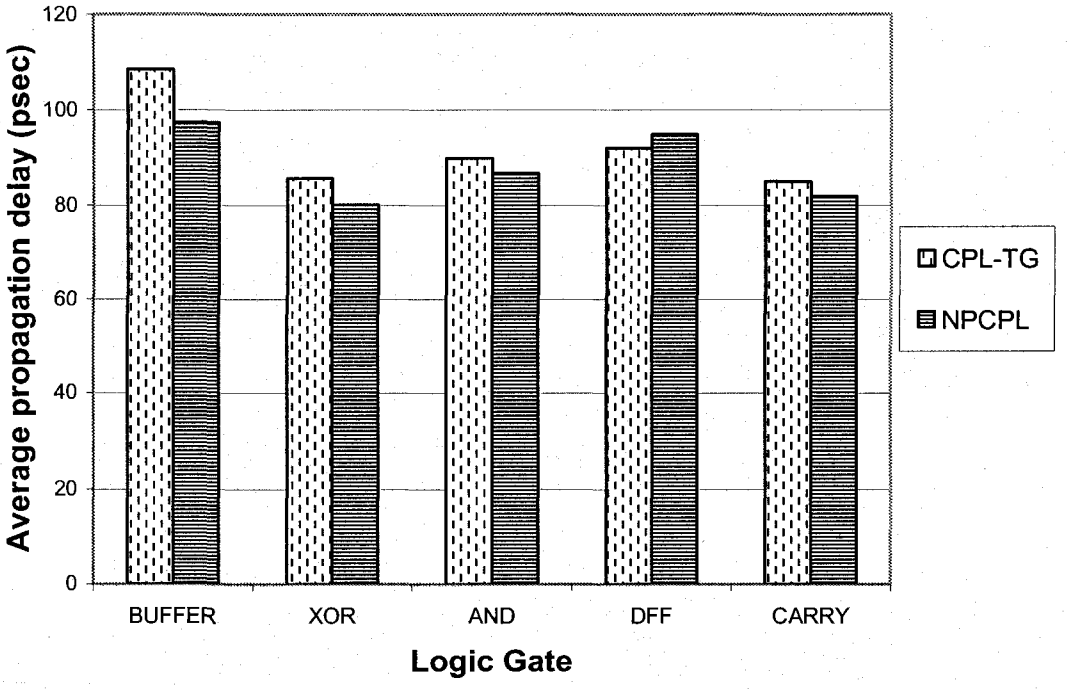
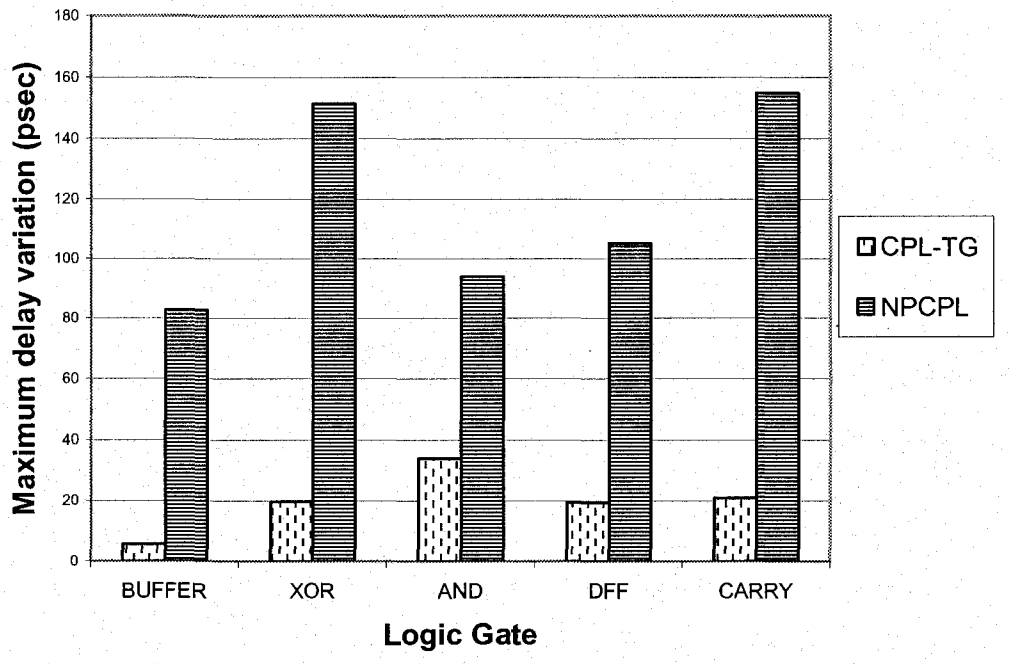


Figure 4.4 Propagation delay.

The histograms of Figure 4.5 summarize the performance of the two logic families i.e. the CPL-TG and the NPCPL. Figure 4.5 (a) illustrates the average propagation delay of a given logic gate, while Figure 4.5 (b) shows the maximum delay variation induced by different input combinations, i.e. D_{max} - D_{min}.



(a)



(b)

Figure 4.5 **a) Average propagation delays, b) Maximum delay variation for different input combinations (D_{max} - D_{min}).**

As an example, Table 4.1 presents the delays of the CPL-TG XOR gate for different input transitions. In the table, the propagation delay is of 97 psec when a = 1 and b is rising. When a is rising and b = 1, a propagation delay of 77 psec is observed. Thus, the maximum delay variation is of 20 psec.

Input transition	Delay (ps)
a ↑ | b = 0	85
a ↓ | b = 0	78
a ↑ | b = 1	77
a ↓ | b = 1	84
b ↑ | a = 0	89
b ↓ | a = 0	92
b ↑ | a = 1	97
b ↓ | a = 1	83
Average delay	**85.66**
D_{max} - D_{min}	**20**

Table 4.1 Delay of the CPL-TG XOR.

In Figure 4.5 (a), one can observe that there are only small differences in the average delays of a given logic family; a variance of 8.5 psec and of 6.9 psec are observed for the CPL-TG and the NPCPL respectively. The NPCPL library is slightly faster than the CPL-TG one; this is caused by the increased input capacitance of the buffer that acts as a load in the CPL-TG test setup.

Figure 4.5 (b) illustrates the maximum delay variation for different input combinations (D_{max} - D_{min}). In average, the NPCPL gates have a delay variation of 118 psec compared to 20 psec for CPL-TG. The CPL-TG logic family obviously has better delay characteristics at the cost of more transistors. Thus, this drawback is acceptable since the performance improvement is significant. The delay variation for different input combinations is improved by six orders of magnitude with the CPL-TG logic in comparison with the NPCPL. On the other hand, the hardware cost has increased from 8 transistors per gate (NPCPL) to 12 transistors per gate (CPL-TG).

As outlined in Figure 4.5, there are still some variations in the delays of the CPL-TG gates. Depending on the function, the input signals attack a various number of gates and drains, so that the input capacitance is not always the same. Figure 4.6 highlights this concept with the case of the AND gate. In this example, the input capacitance of the *A* signal is of two drains, while the *B* signal is applied to four gates and two drains. Obviously, there will be some delay variations induced by the difference of the input capacitance. To reduce the relative variations, one can modify the inverter sizing to improve either the rise time or the fall time. Note that this strategy is not used in this project in order to equalize the driving capacity of the various gates.

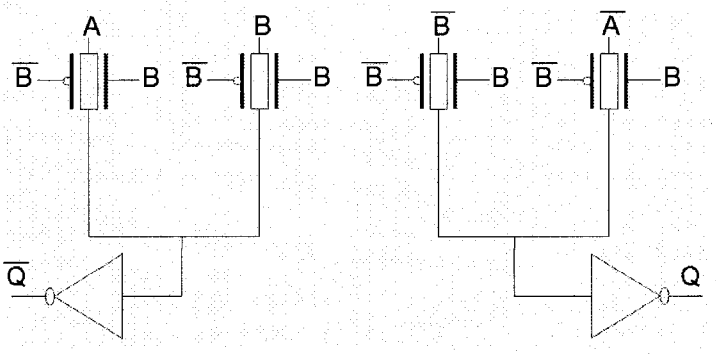


Figure 4.6 Input capacitance for different signals (AND gate).

It should be mentioned that delays of the NPCPL and CPL-TG standard cells have also been evaluated using a supply voltage of 3.3 V. It has been observed that the NPCPL block offers a better delay equalization as the supply voltage increases. However, in this work, we prefer to work with low voltages in order to reduce the power consumption. Therefore, the performance evaluation of the standard cells at 3.3 V is not presented in this thesis.

4.3 Impact of the Logic Family on the Performances of the Wave Pipeline CMAC

The influence of the logic family on the performances of the CMAC is highlighted in Table 4.2. The speeds of the wave pipeline circuits have been evaluated by means of post layout (extracted schematic) simulations with a supply voltage of 1.8 V and an ambiant temperature of 25°C. The power consumption has been measured by means of HSPICE™.

The different signals arrive at the output register of the CMAC at different moments, and some transition periods of duration $D_{max} - D_{min}$ are observed in between the waves. As the logic depth of the circuit increases, some delay uncertainties are accumulated in the various logic paths and the transition period $D_{max} - D_{min}$ becomes more important. Practically, the worst case value of the term D_{max} - D_{min} depends on the input combinations and it can only be estimated by simulations. The wave pipeline CMAC has 4 inputs of 8 bits and there is a total of 2^{32} (4.3×10^9) possible input combinations, which cannot all be simulated for obvious reasons. Therefore, the minimum clock period is estimated based on the observation of a few hundreds input combinations, and a safety margin of 5% is added to the measured time interval for a proper operation of the circuit. In the NPCPL wave pipelined CMAC, the term D_{max} - D_{min} is of approximately 840 ps, and the value of $T_s + T_h$ is equal to 200 ps. If we recall equation (2.17), the theoretical minimum clock period can be expressed as

$$T_{CLK} > (D_{MAX} - D_{MIN}) + T_S + T_H + 2\times\Delta_{CLK}.$$

By replacing the variables by their actual value, we obtain a minimum clock period of 1.1 ns + 2 × ΔCLK, resulting in a maximal frequency of approximately 900 MHz. Note that this value takes into account the 5% of uncertainty on the term D_{max} - D_{min}.

This equation has been used to determine the maximal operating frequency of the wave pipelined CMACs as presented in Table 4.2, considering a safety margin of 5%. The results were also confirmed by simulations. The number of *waves* is calculated by

counting the number of clock cycles between some given inputs and their expected outputs. It can also be estimated by the product Latency × Frequency.

Implementation method	Frequency (MHz)	Latency (ns)	D_{max} -D_{min} (ps)	Power consumption* (mW)	Number of waves	Number of transistors
NPCPL (wave pipeline)	830	4.4	840	23	4	14 800
CPL-TG (wave pipeline)	770	4.0	887	72	3	22 220

*** Power consumption measurements correspond to rough order of magnitude estimates using HSPICE™.**

Table 4.2 Performance of the wave pipelined CMACs (extracted schematic)

It is observed in simulation that the wave pipeline circuits cannot operate properly at the theoretical frequency; the circuits are slowed down by the different uncertainties, including the term $2 \times \Delta_{CLK}$ of (2.17). Surprisingly, the latency and the frequency of the NPCPL CMAC are greater than those of the CPL-TG circuit; these results seem to be in contradiction with section 4.2 (see Figure 4.5). However, they can still be explained analytically. The difference in the term $D_{max} - D_{min}$ can be explained by the broader range of input capacitances in the CPL-TG circuit. Globally, the currents in the two circuits are similar because of indentical inverter sizing. Nevertheless, at some point in the circuit, some parallel gates (at the same logic depth) sometimes have a different fanout. Depending on what input the signal is connected, the capacitance can consist of either a drain or a gate as illustrated in Figure 4.6. Thus, the impact of the fanout difference is doubled in the CPL-TG circuit, eliminating the advantage of the CMOS switches. On the other hand, the latency difference can be explained by the distribution of the NPCPL delays. Even if the average propagation delay is smaller for the NPCPL library, some transition patterns are probably not likely to occur at the different nodes inside the circuit, so that only the greatest propagation delays remain.

The CPL-TG CMAC dissipates more power. If we run the two circuits at the same frequency, the CPL-TG CMAC dissipates approximately three times of power compared to that of the NPCPL circuit.

There are three contributions to this increase of the power consumption:

i) The capacitance in the circuit is increased significantly by replacing the NMOS pass transistors by CMOS ones,

ii) The voltage swing at most of the internal node has doubled, from 0.9 V to 1.8 V.

iii) The short circuit currents may be raised because of the longer rise time and/or fall time at some circuit nodes, due to more capacitance.

The power consumption due to switching can be written as

$$P = VI = V \times \frac{Q}{\Delta T} = V \times \frac{(C \times V)}{\Delta T} = \frac{C \times V^2}{\Delta T} \qquad (4.1)$$

In comparison with the NPCPL CMAC, the capacitance in the CPL-TG circuit has doubled in 50% of the nodes, while the voltage swing has doubled in 50% of the nodes. From a power perspective, the increase of the voltage swing has more impact since it is a quadratic function. The increase of the short circuit currents occurs in 50% of the branches. It is important to mention that all three contributions should be weighted by the switching activity factor at different nodes. The combination of these three contributions is causing the increase of the power dissipation in the CPL-TG circuit. Thus, the NPCPL CMAC has been chosen for the fabrication of the prototype chip.

4.4 Comparison with Existing Designs of Arithmetic Units

It is at the author's knowledge the first time that a wave pipelined CMAC is implemented in ASIC. The wave pipelined circuit (NPCPL) is compared to another 32×16 (conventional pipeline) MAC reported in [23]. This is a fair comparison because the implementation technologies and the supply voltages are similar for the two designs. Considering that there is only one 32×16 multiplier in [23], it seems that their area is more important than ours in proportion. The wave pipeline method also has a speed advantage, considering that three stages of pipeline were necessary to implement the MAC in [23].

Design	Function	Number of Transistors	Frequency [MHz]	Supply Voltage [V]	Technology
[23]	MAC 32×16 or 24×24	41 121	300	1.5	Conventional pipeline ASIC – 0.15 µm
Proposed CMAC (NPCPL)	CMAC 4×(8×8)	29 600	830	1.8	Wave pipeline ASIC - 0.18 µm

Table 4.3 Comparison of the simulation results of existing designs of high-speed arithmetic units.

4.5 Multipath Searcher Performances

From system level considerations, the proposed implementation requires a speed of 292×10^6 CMAC cycles per second. Therefore, a single wave pipeline CMAC can handle the task. Table 4.4 summarizes the characteristics of the proposed multipath searcher in comparison with another implementation found in the literature [4]. As illustrated in the table, the high-speed processing improves the two most important metrics of the multipath searcher, the resolution and the acquisition time, while reducing the number of CMACs necessary to perform the calculations. In the table, it is observed that the acquisition time of the proposed searcher is of 20 slots instead of 15 as stated in section 3.1. This value is obtained because a period of 5 slots is necessary to store the

incoming signal $\mathbf{r}[n]$ in memory. The speed of 350 MHz allows for some time guards between the different verification steps, so that the control logic has time to process the candidates. In the table, the power consumption and the number of logic gates are flagged with a star (*) to indicate that they correspond to some rough order of magnitude estimates. It is important to mention that these two metrics are for the complete searcher unit in [4], as opposed to the CMAC only for the proposed design. The resolution and the acquisition time of the multipath searcher have been improved at the cost of more calculation (×5) in the proposed design. However, only one CMAC operating at 350 MHz is needed, instead of 12 CMACs operating at 3.84 MHz in the existing implementation. In order to meet the specifications of the proposed multipath searcher with a processing rate of 3.84 MHz, 77 CMACs would be required resulting in an increased gate count and power consumption.

Comparison criterions	Existing Implementation [4]	Proposed design
Acquisition time	50 ms (75 slots)	≈ 13.3 ms (20 slots)
Resolution of the searcher	½ chip	¼ chip
Number of CMAC	12	1
Complex MAC cycles/sec	61×10^6	292×10^6
Processing rate	3.84 MHz	350 MHz
Number of logic gates*	150 000 CMOS gates	3 700 NPCPL gates
Power consumption*	6.6 mW	12 mW
Input word length	6 bits	8 bits

*** Rough order of magnitude estimates.**

Table 4.4 Comparison of the proposed multipath searcher with an existing implementation.

The power consumption of the CMAC in Table 4.4 corresponds to a projection of what it should be at 350 MHz, considering the parasitic capacitances. Some more accurate measurements will be available after the hardware testing of the prototype chip (to be done later). Figure 4.7 presents a picture of the final layout of the CMAC made of the NPCPL standard cells.

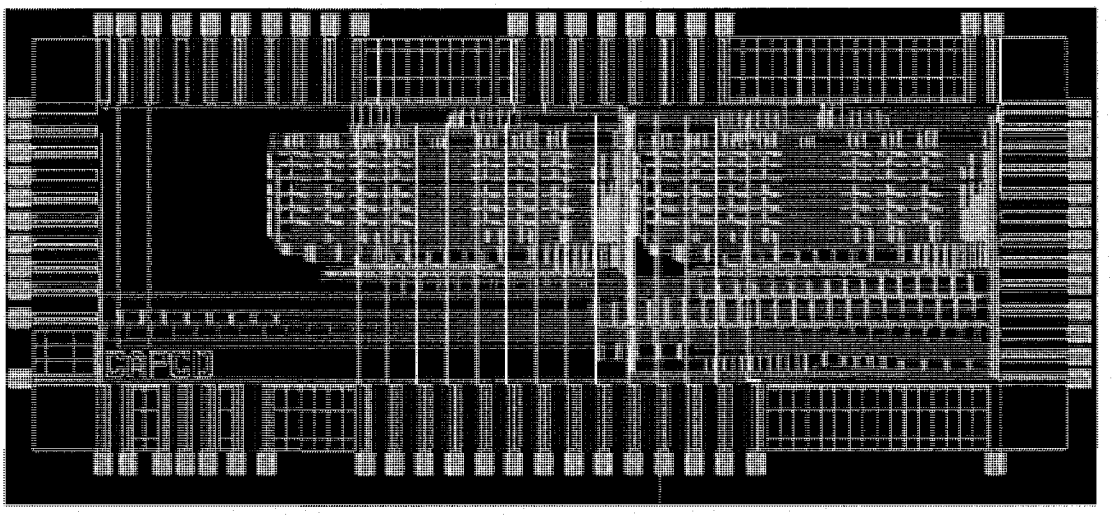


Figure 4.7 CMAC layout with I/O ring.

4.6 Summary

The performances of the multipath searcher have been presented in this section. The proposed multipath searcher has a reduced acquisition time and an improved resolution in comparison with an existing implementation. It also presents low misdetection and false alarm probabilities for the test cases recommended by the 3GPP standard, demonstrating the effectiveness of proposed detection method.

The simulation results demonstrate that the CMAC performances are degraded in terms of frequency and power when using the CPL-TG logic; which is against our expectations. As a guideline for future wave pipeline designs, it should be remembered that the equalization of the output load on the cells is more important than the individual propagation delays of the gates. For obvious reasons, the NPCPL logic was chosen for the fabrication of the prototype chip. The speed requirement of the proposed multipath searcher is of 350 MHz, and one NPCPL CMAC is fast enough to handle the calculations.

5 Conclusion

5.1 Concluding Remarks

In this thesis, the design of a multipath searcher having a short acquisition time and a high resolution has been proposed. This searcher circuit is to implement a double-dwell algorithm that includes a verification stage to reduce the rate of false alarm. The computation procedure of the algorithm is made to be performed during a short period to improve the quality of search. In order to implement the algorithm without increasing the hardware complexity, the multipath searcher has only one CMAC cell to perform the computation tasks. It is designed to operate at a high speed in order to complete the searching procedure according to the time limit of the search period. The wave pipeline technique has been used for the development of this CMAC cell.

Two libraries of standard cells have been developed using the NPCPL and the CPL-TG logics. The structures of the arithmetic units have been chosen carefully to comply with the precision requirement of the system model while meeting the fan-in/fan-out constraints of the NPCPL and the CPL-TG logics. Simulation results show that the NPCPL CMAC offers better performances in terms of speed and power consumption. The increase of the capacitance in the CPL-TG CMAC is responsible for the performances degradation of that circuit. The speed of 830 MHz offered by the NPCPL CMAC is sufficient to handle the calculations of the proposed searcher. The layout of the adders, multipliers and CMAC have been full-custom designed using the NPCPL standard cells. Globally, only one wave pipelined CMAC is needed in the multipath searcher, resulting in a low hardware complexity. A test chip has been fabricated in a CMOS 0.18 μm process. The chip test is currently in process.

5.2 Future Work

The 3GPP is also planning the Long Term Evolution (LTE) of the third generation to ensure the competitiveness of this technology over the next decades. It has been established that the fourth generation (4G) of cellular will be based on Multiple-inputs and Multiple-outputs (MIMO), where many antennas will be used for transmission and reception [45]. In MIMO systems, one multipath searcher will be required per receive antenna. Therefore, the need for correlations as well as other complex arithmetic operations will be duplicated, and the wave pipeline CMAC could be used to perform the calculations of different units. Some future work could target the implementation of a MIMO receiver based on high-speed processing.

Some other future investigations could be related to the scalability of the wave pipeline method to larger systems. In general, as the logic depth increases, the accumulation of the incertitude on the propagation delays affects the performances of the circuit. The major concern is whether wave pipelining should only be applied at the module level (e.g. multiplier, RAM) or if it can be applied to large-scale systems. In the case where only the bottlenecks are wave pipelined, conventional registers act as bondaries and the rest of the system view the wave pipelined blocks as synchronous modules. For such hybrid circuits, wave pipelining must also be applied on the clock tree in the rest of the circuit for a proper operation. The mix of both approaches is probably one of the solutions for the development of a wave pipelined System on Chip (SOC). However, the implementation of such a system would require a high level of expertise as well as a long design cycle. Scalibility can only be obtained with the automation of the design process. Even though a few optimization tools and algorithms have been presented in the literature ([46], [47], [48], [49], [50], [51]), they only target specific problems. An automated design flow that takes as input some high level specifications and that produces some performant layouts is needed. All aspects of VLSI design including logic gates delays, architectures, logic synthesis, placement, routing and parasitic effects must be incorporated into a global view of the problem. Some design methodologies could be studied as part of future investigations.

References

[1] M. Schwartz, *Mobile Wireless Communications*, New-York: Cambridge university press, 2005.

[2] H. Boujemâa, "BER Performance Analysis of the Discrete Time and the Continuous Time Rake Receivers for UMTS Downlink", *IEEE Trans. on Wireless Comm.*, Vol. 5, April 2006, pp. 900-909.

[3] A. Eltawil, B. Daneshrad, "A Low-Power DS-CDMA RAKE Receiver Utilizing Resource Allocation Techniques", *IEEE Journal of Solid-State Circuits*, Vol. 39, pp. 1321-1333, August 2004.

[4] E. Grayver, "Design and VLSI Implementation of a WCDMA Multipath Searcher", *IEEE Trans. on Vehicular Tech.,* Vol 54, May 2005, pp. 889-902.

[5] W. P. Burleson, M. Ciesielski, F. Class, W. Liu, "Wave-Pipelining: a Tutorial and Research Survey*", IEEE Trans. on VLSI Systems*, Vol. 6, Sept. 1998, pp.464-474. March 2007, pp. 518-529.

[6] Spreading and modulation (FDD), 3rd Generation Partnership Project (3GPP) TS 25.213.

[7] Physical channels and mapping of transport channel onto physical channels (FDD) , 3rd Generation Partnership Project (3GPP) TS 25.211.

[8] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication*, 1st ed. Reading, MA: Addison-Wesley, 1995.

[9] R. Rick, L. Milstein, "Parallel Acquisition in Mobile DS-CDMA Systems", *IEEE Trans. on Communications*, Vol. 45, November 1997, pp. 1466-1476.

[10] E. Sourour, S. Gupta, "Direst-Sequence Spread-Spectrum Parallel Acquisition in a Fading Mobile Channel", *IEEE Trans. on Communications*, Vol. 38, July 1990, pp. 992-998.

[11] R. Rick and L. B. Milstein, "Parallel Acquisition in Mobile DS-CDMA Systems," *IEEE Trans. Communications*, vol. 45, Nov. 1997, pp. 1466–1476.

[12] A. Polydoros and C. L. Weber, "A Unified Approach to Serial Search Spread-Spectrum Code Acquisition—Part I: General Theory," *IEEE Trans. Commun.*, vol. COM-32, May 1984, pp. 542–549.

[13] A. Polydoros and C. L. Weber, "A Unified Approach to Serial Search Spread-Spectrum Code Acquisition— Part II: A Matched-Filter Receiver," *IEEE Trans. Commun.*, vol. COM-32, May 1984, pp. 550–560.

[14] A. Polydoros and M. K. Simon, "Generalized Serial Search Code Acquisition: The Equivalent Circular State Diagram Approach," *IEEE Trans. Commun.*, vol. COM-32, Dec. 1984, pp. 1260–1268.

[15] S. Chung, "A New Serial Search Acquisition Approach with Automatic Decision Threshold Control," *in Proc. Int. Conf. IEEE Vehicular Tech.*, Chicago, USA, July 1995, pp. 530-536.

[16] S. Lee, J. Kim, "Double-Dwell Serial Search Algorithm with Multi-level Threshold Values in DS-CDMA Systems", *in Proc. Int. Conf. IEEE Trans. On Comm. Tech.*, Beijing, China, November 1998, pp. 522-525.

[17] W. Suwansantisuk, M. Win, "Multipath Aided Rapid Acquisition: Optimal Search Strategies", *IEEE Transaction on Information Theory*, Vol. 53, January 2007, pp. 174-193.

[18] A. Van Der Meer, R. Liyana-Pathirana, "Performance Analysis of a Hybrid Acquisition System for DS Spread Spectrum", *in Proc. Conf. on Convergent Technologies for Asia-Pacific Region,* Kuala-Lumpur, Malaysia, October 2003, pp. 121-125.

[19] U. Madhow and M. B. Pursley, "Mathematical Modeling and Performance Analysis for a Two-Stage Acquisition Scheme for Direct-Sequence Spread-Spectrum CDMA," *IEEE Trans. Commun.*, vol. 43, Sept. 1995, pp. 2511–2520.

[20] J. H. Kim, S. V. Sarin, M. Yasunaga, and H. Oh, "Robust Noncoherent PN-Code Acquisition for CDMA Communication Systems," *IEEE Trans. Veh. Technol.*, vol. 50, Jan. 2001, pp. 278–286.

[21] F. Morin, M. Vidal, D. Massicotte, "A High Throughput Architecture for Channel Equalization Based on a Neural Network Using a Wave Pipeline Method", in *Proc. IEEE Canadian Conf. on EECE*, Edmonton, Canada, May 1999, pp. 560-564.

[22] D. Shim, W. Kim, "The Design of 16X16 Wave Pipelined Multiplier Using Fan-in Equalization Technique", in *Proc. Midwest Symp. Circuits and Systems,* Notre-Dame, USA, May 1998, pp. 336-339.

[23] N. Itoh, Y. Tsukamoto, "A 32X24-Bit Multiplier-Accumulator with Advanced Rectangular Styled Wallace-Tree Structure", *in Proc. Int. Symp. on Circuits and Systems*, Kobe, Japan, May 2005, pp. 73-76.

[24] M. Izumikawa, H. Igura, "A 0.25-μm CMOS 0.9-V 100-MHz DSP Core", *IEEE Journal of Solid-State Circuits*, Vol. 32, Jan. 1997, pp. 52-61.

[25] C. Cheng, P. Tseng, L. Chen, "A Digital Signal Processor With Programmable Correlator Array Architecture for Third Generation Wireless Communication System", *IEEE Trans. Circuits and* Systems, Vol. 48, December 2001, pp. 1110-1120.

[26] Q. Liu, J.M.P Langlois, D. Al-Khalili, "Synthesis of a 12-Bit Complex Mixer for FPGA Implementation", *in Proc. Canadian Conference on EECE*, Winnipeg, Canada, May 2003, pp. 81-84.

[27] T. Sansaloni, J. Valls, K.K. Parhi, "FPGA-Based Digit-Serial Complex Number Multiplier-Accumulator", *in Proc. IEEE International Symposium on Circuits and Systems*, Geneva, Switzerland, May 2000, pp.585-588.

[28] G. Wang, "The Efficient Implementation of Complex Number Arithmetic", *Proc. Region 5 Conference: Annual Technical and Leadership Workshop*, Oklahoma City, USA, April 2004, pp. 113-117.

[29] Ghosh, D.; Nandy, S.K., "NPCPL: Normal Process Complementary Pass Transistor Logic for Low Latency, High Throughput Designs", *in Proc. Sixth Int. Conf. on VLSI Design*, San Diego, USA, Jan. 1993, pp. 341 – 346.

[30] Ghosh, D.; Nandy, S.K., "A 400 MHz Wave-Pipelined 8×8-bit Multiplier in CMOS Technology", *in Proc. IEEE Int. Conf. on Computer Design*, Oct. 1993, pp. 198 – 201.

[31] H. Choi, S.H. Hwang, "Design of Wave-Pipelined 900 MHz 16b Ripple-Carry Adder Using Modified NPCPL", *Proc. IEEE Int. Symp. on Circuits and Systems*, Atlanta, USA, May 1996, pp.182-184.

[32] Abu-Khater, I.S.; Bellaouar, A.; Elmasry, M.I., "Circuit Techniques for CMOS Low-Power High-Performance Multipliers", *IEEE Journal of Solid-State Circuits*, Vol. 31, Oct. 1996, pp. 1535 – 1546.

[33] Xuguang Zhang; Sridhar, R., "CMOS Wave Pipelining Using Transmission-Gate Logic", *in Proc. IEEE Int. ASIC Conf. and Exhibit*, Rochester, USA, Sept. 1994, pp. 92-95.

[34] A. Martinez-Smith, B. McGee, "Speed and Power Comparison of CMOS Wave Pipelined Systems and Low Power WTGL", *Proc. IEEE Int. Symp. on Circuits and Systems*, Atlanta, USA, May 1996, pp.156-159.

[35] Lien, W.; Burleson, W., "Wave-Domino Logic: Timing Analysis and Applications", *in Proc. IEEE Int. Symp. on Circuits and Systems*, San Diego, USA, May 1992, pp. 2949 - 2952.

[36] Enrique Fernandez, G.; Sridhar, R., "Dual Rail Static CMOS Architecture for Wave Pipelining", *in Proc. Ninth Int. Conf. on VLSI Design*, Bangalore, India, Jan. 1996, pp. 335 – 336.

[37] E.I. Boemo, S. Lopez-Buedo, J.M. Meneses, "Wave Pipelines Via Look-Up Tables", *in Proc. IEEE International Symposium on Circuits and Systems*, Atlanta, USA, May 1996, pp.185-188.

[38] G. Lakshminarayanan and B. Venkataramani, "Optimization Techniques for FPGA-Based Wave-Pipelined DSP Blocks", *IEEE Transactions on VLSI Systems*, Vol. 13, July 2005, pp. 783-793.

[39] G. Lakshminarayanan, B. Venkataramani, M. Yousuff Shariff, T. Rajavelu, and M. Ramesh, "Self Tuning Circuit for FPGA Based Wavepipelined Multipliers," *in Proc. VLSI Design Test Worksho*, Mysore, India, Aug. 2004, pp. 93–101.

[40] J.B. Sulistyo, S.H. Dong, "HyPipe: A New Approach for High Speed Circuit Design", *in Proc. IEEE International ASIC/SOC Conference*, Rochester, USA, September 2002, pp. 203-207.

[41] J. Sulistyo, D. Ha, "5 GHz Pipelined Multiplier and MAC in 0.18 /spl mu/m Complementary Static CMOS", *in Proc. IEEE International Symposium on Circuits and Systems*, Bangkok, Thailand, May 2003, pp.117-120.

[42] S.B. Tatapudi, J.G. Delgado-Frias, "A High Performance Hybrid Wave-Pipelined Multiplier", *in Proc. IEEE Computer Society Annual Symposium on VLSI*, May 2005, pp. 282-283.

[43] M. Donno, A. Ivaldi, L. Benini, "Clock-Tree Power Optimization Based on RTL Clock-Gating", *in Proc. Design Automation Conference*, Anaheim, California, June 2003, pp. 622-627.

[44] Base station (BS) radio transmission and reception (FDD) , 3rd Generation Partnership Project (3GPP) TS 25.104.

[45] J. Boyd, "Sneak Peek at Cellphone Future", *IEEE Spectrum*, Vol.43, June 2006, pp. 15.

[46] Chatterjee, M.; Banerjee, S., "Buffer Assignment Algorithms on Data Driven ASICs", *IEEE Transactions on Computers*, Vol. 49, Jan. 2000, pp. 16 – 32.

[47] Woo Jin Kim; Yong-Bin Kim, "Wave Pipelined Circuits Synthesis", *in Proc. IEEE Instrumentation and Measurement Technology Conference*, Ottawa, Canada, May 2005, pp. 32 – 36.

[48] Chen Lan; Tang Zhimin, "Design Issues of Deep Submicron Wave Pipelining Technology", *in Proc. 4th International Conference on ASIC*, Washington DC, USA, Oct. 2001, pp. 62 – 66.

[49] Lakshminarayanan, G.; Venkataramani, B., "Optimization Techniques for FPGA-Based Wave-Pipelined DSP Blocks", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 13, July 2005, pp. 783 – 793.

[50] Chang, C.-H.; Davidson, E.S.; Sakallah, K.A., "Using Constraint Geometry to Determine Maximum Rate Pipeline Clocking", *in Proc. International Conference on Computer-Aided Design*, Santa Clara, USA, Nov. 1992, pp. 142 – 148.

[51] Sun, S.-Z.; Du, D.H.C.; Chen, H.-C., "Efficient Timing Analysis for CMOS Circuit Considering Data Dependent Delays", *in Proc. IEEE International Conference on Computer Design*, San Jose, USA, Oct. 1994, pp. 156 – 159.

APPENDIX

ASIC PINOUT

Signal – pin number

A0 – 73
A1 – 44
A2 – 45
A3 – 46
A4 – 47
A5 – 71
A6 – 70
A7 – 72

B0 – 73
B1 – 22
B2 – 21
B3 – 20
B4 – 19
B5 – 18
B6 – 16
B7 – 15

C0 – 14
C1 – 12
C2 – 75
C3 – 76
C4 – 77
C5 – 11
C6 – 10
C7 – 9

D0 – 7
D1 – 8
D2 – 79
D3 – 80
D4 – 81
D5 – 82
D6 – 83
D7 – 84

Q0 – 42
Q1 – 41
Q2 – 40
Q3 – 39
Q4 – 37
Q5 – 36
Q6 – 6
Q7 – 4
Q8 – 3
Q9 – 2
Q10 –116
Q11– 117
Q12 – 105
Q13 – 104
Q14 – 102
Q15 – 101
Q16 – 100
Q17 – 99
Q18 – 97
Q19 – 96
Q20 – 119
Q21 – 120

VDD1 (Core) – 43, 85
VSS1 (Core) – 48, 88

VDD2 (Pads) – 33, 93
VSS2 (Pads) – 38, 98

RST – 59
RST_INV – 60

SAMP_OUTPUT – 86
SAMP_OUT_INV – 87

CLK – 69