UNIVERSITY OF PATRAS
SCHOOL OF ENGINEERING

FACULTY OF ELECTRICAL
AND COMPUTER ENGINEERING
DIVISION OF ELECTRONICS
AND COMPUTERS

DIPLOMA THESIS

OF

PAPADOPOULOS CHARALABOS

S.N. 5731

Digital Receiver Subsystem Design for Ultra Wideband System

SUPERVISING PROFESSOR: KALIVAS GRIGORIOS

Diploma Thesis Number:

PATRAS 2009

CERTIFICATION

It is certified that the diploma thesis with subject:

Digital Receiver Subsystem Design for Ultra Wideband System

of the student of the Electrical and Computer Engineering Department

Papadopoulos Charalabos

(S.N. 5731)

Was publicly presented and examined at the

Electrical and Computer Engineering Department on

19/6/2009

The Supervisor	The Director of the Division
Kalivas Grigorios	Goutis Constantinos

SUMMARY

The purpose of this diploma thesis is the design and implementation of a RAKE digital receiver for UWB wireless transmission. The circuit was designed in VHDL code and it was modeled at register-transfer level.

Initially, basic information is discussed about the UWB technology (chronology, advantages, applications, modulation techniques, spectrum broadening techniques, etc.). Then the UWB channel is studied and conclusions are drawn about the necessary architecture of the Rake receiver. Afterwards, the operation of the Rake receiver is analyzed mathematically and the channel estimation algorithm used is explained. Finally, our system architecture is presented, along with the results of the simulations from the corresponding software simulations suite. All of the VHDL code is provided in the DVD that accompanies this diploma thesis.

FOREWORD

This diploma thesis constitutes part of relevant research taking place at the Applied Electronics Laboratory supervised by the Associate Professor Mr. G. Kalivas. The contribution of post-graduate student Christos Thomou was of profound importance, as we worked closely together on an almost daily basis.

During the development of this diploma thesis I acquired valuable knowledge and experience in the field of Electrical Engineering, both on a theoretical as well as on a practical level. For this reason I feel the need to thank the Associate Professor κ Mr. G. Kalivas and the post-graduate student C. Thomou, with whom I collaborated ideally over the course of this diploma thesis.

Patras 2009,
Charalabos Papadopoulos

TABLE OF CONTENTS

Chapter 1 : Theoretical Approach

1.1. Introduction

Wireless communication systems have evolved significantly over the last two decades. The explosive growth of the wireless communications market is expected to continue over the following years, due to the continually increasing demand for wireless services of every form. The new generations of mobile wireless systems are required to provide flexible data transfer rates (high, moderate, low) for a wide range of applications (such as video, data transfer, position detection, etc.) to as many users as possible. That task however must be accomplished by taking into account the limitations with regard to the available resources, such as energy and spectrum. Since more and more devices transmit wirelessly, the future technologies will be faced with spectrum congestion and the coexistence of various wireless devices will pose a serious issue.

The UWB (Ultra Wide Band) wireless transmission technology promises to provide a solution to the above problem and has drawn the attention of the scientific community over the recent years since it has the potential for very high data transfer rates at a relatively low price. The UWB systems coexist with other wireless systems. The emitting power of the UWB devices is controlled by the authorities in charge , such as the Federal Communications Commission – FCC in the United States, so that the limited bandwidth systems do not get significantly affected by the UWB systems.

1.2. Chronology

In reality, wireless transmission started out with the form of wideband transmission – that is to say with the form of UWB transmission according to the present definition. The transmitters of that era were devices that transmitted either a pulse (corresponding to one) or a gap (corresponding to zero). The receivers of that era which received that simple, full of noise signal, were simple amplitude detectors that could not collect the wideband signal energy in an effective manner. The SNR performance of these systems was rudimentary and consequently there was a demand for great amounts of transmission energy in order for the desired communication over the distances in question to be achieved. The high transmission energies combined with the great range

of the transmission spectrum naturally did not make allowance for simultaneous transmission by many users. As a result, wireless technology was forced to turn to transmission of increasingly narrower spectrum per signal. Ideally, a signal was as narrow as the information bandwidth. The regulations of 1912 imposed the narrowest possible transmission ranges and determined the segregation of the wireless services based on the frequency differentiation.

In 1933 however, the advantages of the deliberate and controllable broadening of the signal spectrum at values multiple of the information bandwidth were realized, in the form of the FM radio. Based on that approach, the bandwidth was able to be replaced with noise immunity, an especially desirable attribute for the transmission of FM signals. The “one channel per user” technique had been used since 1912, and the transmission was accomplished with the smallest possible bandwidth. Afterwards however, the spread spectrum technology was to surface. In 1985, the Federal Communications Commission (FCC) allowed the commercial use of the spread spectrum technique, according to which the users are separated with different Direct sequence Spread-Spectrum (DSSS) codes instead of using different frequencies. Later, in 1995, the CDMA (Code Division Multiple Access) technology was applied commercially in mobile telephony. In 1999, the International Telecommunications Union adopted an industrial standard for third generation wireless systems which is able to provide data transfer at high transfer rates, along with other new services. This 3G standard is based on three operating modes based on the CDMA technology. Consequently, the spectrum distribution policy broke free from the “frequency separation” mentality. Still, according to this standard, multiple users transmit in a spectrum that has been predefined for that purpose.

During the last fifty years, a fair amount of research has been conducted with regard to the wideband impulse radars, which are the predecessors of the UWB. Independently, during the decades of 1980 and 1990, commercial experiments, inventions and patents lead the FCC to publish a relevant standard in 2002, which allows the use of low power UWB for commercial applications. This fact establishes a new order in the management policy of the electromagnetic spectrum. According to the new regulations, multiple users, often without permission, are able to use the spectrum that was previously allocated to other users, including users with permission, with no occurrence of

interference. Consequently, the UWB technology is an exciting new technology since it allows unprecedented access to a very wide part of the spectrum, and without permission. At this moment, commercial standards are created and the technology is at a market infiltration phase.

1.3. Description of the UWB technology

At Ultra Wide Band wireless communication, pulses of very short duration (namely at the nanosecond range) are used, which have great spectral range. Due to the very narrow pulses that are used, this type of communication is called *Impulse radio* (IR).

Using conventional wireless technology, information transmission is accomplished by modulating sinusoidal waves, where the bandwidth increase must be accompanied by carrier signal frequency increase. UWB can be considered as an outstretched spectrum technique which uses a very wide spectrum even without modulation. The signal energy is thus very outstretched and the UWB spectrum bears great resemblance to the noise spectrum.

The UWB signals can be defined as signals with bandwidth greater than 25% of their main frequency, or signals with total bandwidth greater than 500 MHz. UWB devices often transmit at frequencies ranging between 1,5 and 4 GHz.
The Fractional bandwidth B_f is defined as :

$$B_f = 2\frac{(f_h - f_L)}{(f_h + f_L)}$$

where f_h and f_L the highest and lowest frequency observed in the system.

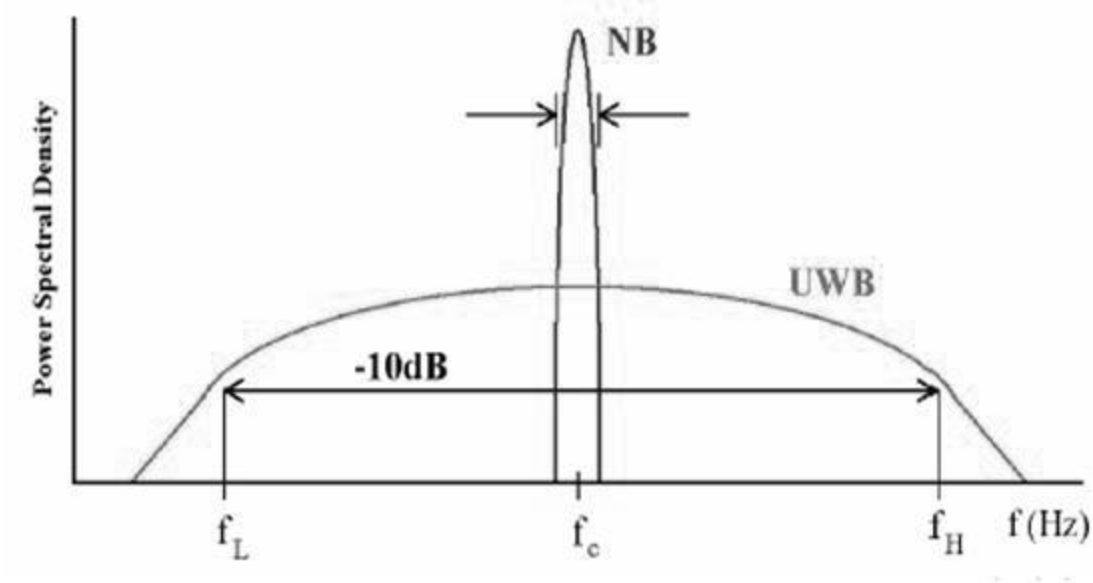


Figure 1.1 Comparison of UWB spectrum with that of conventional wireless communications

1.4. UWB Advantages

There are certain characteristics of UWB that constitute its use more beneficial compared to the other wireless communication techniques.

- It offers high data transfer rate (Bit Rate)

The demands of the market for wideband services are continually increasing and it is expected that the demands for greater spectral range will increase exponentially as most of the new services require high data transfer rates. For the transmission of high bit rates, greater bandwidth is a necessity, which leads to the use of high carrier frequency, which is at a conflict with the fact that greater carrier frequencies sustain significant absorption in the upper layers of the atmosphere and with the fact that the parts of a transmitter or a receiver (antennas, amplifiers) operate at relatively low frequencies. This is because by increasing the carrier frequency, the price of the processing systems (e.g. amplifiers) increases exponentially. As a result, processing is preferable to take place at lower frequencies.

- Great bandwidth – high resolution

High resolution is of great importance in radar applications. From signal theory it is known that a narrow pulse at the time domain has an outstretched spectrum at the frequency domain. The inverse of the spectral range $\frac{1}{\Delta f}$ is proportional to the achievable resolution.

- Great bandwidth–high resistance at multipath transmission

Since the UWB signal has as great a bandwidth, its channel is sufficiently selective to the frequencies and the received signal has a significant number of multipath components. As a result, the UWB system displays sufficient capability in separating its signals from those that are received from different paths, and from noise. In this way, UWB operates well enough in applications such as home networking, wireless internet access in interior environments as it is necessary that a system displays high tolerance to multipath phenomena, since the differences between the signals' times of arrival in a closed space are a few nanoseconds.

- Low detection probability and low obstruction probability

UWB spreads its energy over a quite large spectrum and this results in low detection probability and low obstruction probability. UWB is thus especially suitable for military applications such as secret communication inside a hostile environment. Apart from that, it is observed that UWB is not sensitive to intentional interference.

- Possesses free space in its spectrum

UWB's low power density means that it could be used without permission from the FCC (Federal Communications Commission) since the maximum limit of radiation power that is allowed to be emitted by electronic devices without the need of a special permission has been determined by FCC. The transmitted signal is similar in shape to noise and as a result can be used without interfering with the wireless systems in existence.

- Low implementation cost

Since UWB is based on communication without carrier, it is obvious that the transceivers can be produced at very low cost using the CMOS technology, instead of the expensive GaAs MMIC (Monolithic Microwave Integrated Circuit). Consequently, UWB is very promising for a number of low-cost applications. Collision avoidance systems, airbag sensors and fluid flow level sensors are mere examples of such applications. Bluetooth is a technology already in existence that is competent to UWB. In order to show the difference between the 2 technologies, with regard to cost, a simple example is presented. For the development of a Bluetooth circuit, the cost is at 5$ while it is estimated that for the development of that circuit using UWB, less than 1$ is required.

1.4.1 UWB Spectral Supremacy

By comparing the spectral efficiency of UWB with that of the other competitive technologies, it is possible to showcase its supremacy in that aspect.

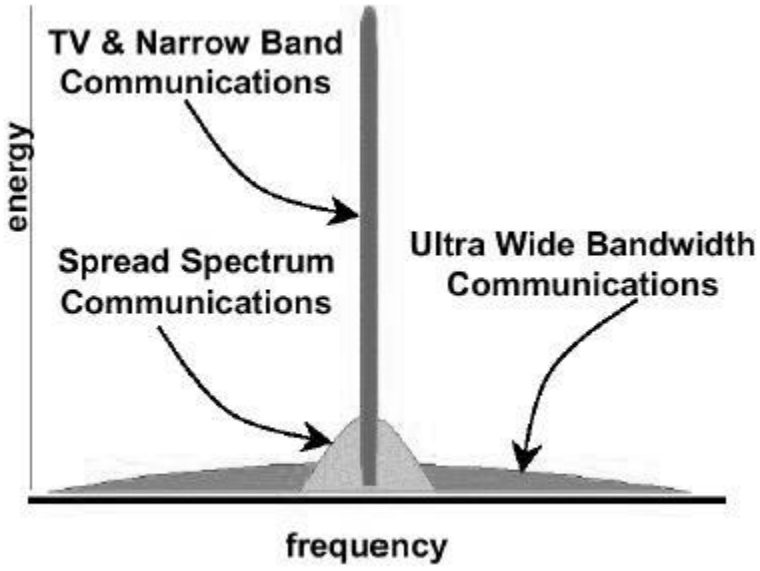


Figure 1.2 Comparison of UWB spectrum with that of the Spread Spectrum και Narrowband systems

IEEE 802.11b : This technology has a range of 100 meters in free space. Inside a circle of 100m radius, 3 IEEE 802.11b systems can operate simultaneously providing a maximum speed of 11Mbps each. The total speed is thus 33 Mbps, which divided by the surface of the circle yields a spatial capacity of 1Kbps/m^2. ($\frac{33Mbps}{\pi \cdot (100m)^2} = 1.05Kbps/m^2$)

Bluetooth : Bluetooth, using relatively low power, has a range of 10 meters in free space. Inside a circle of 10m radius, 10 Bluetooth piconets can operate simultaneously, with 5 of them providing a total speed of 10Mbps. Dividing the speed by the surface of the circle yields a spatial capacity of 30Kbps/m^2.

$$\left(\frac{10Mbps}{\pi \cdot (10m)^2} = 31.8Kbps/m^2\right)$$

IEEE 802.11a : This technology has a range of 50 meters in free space. Inside a circle of 50m radius, 10 IEEE802.11a systems can operate simultaneously, providing a speed of 54Mbps each. The total speed is thus 648 Mbps which divided by the surface of the circle yields a spatial capacity of about 83Kbps/m^2.

$$\left(\frac{648Mbps}{\pi \cdot (50m)^2} = 82.51Kbps/m^2\right)$$

UWB : This technology has a range of 10 meters in free space. Inside a circle of 10m radius, 6 UWB systems can operate simultaneously, providing a speed of 5Mbps each. The total speed is thus 300 Mbps, which divided by the surface of the circle yields a spatial capacity of about 1000Kbps/m^2.

$$\left(\frac{300Mbps}{\pi \cdot (10m)^2} = 1000Kbps/m^2\right)$$

UWB's spectral supremacy becomes apparent from the above, and it can also be observed graphically in the below figure 1.3.

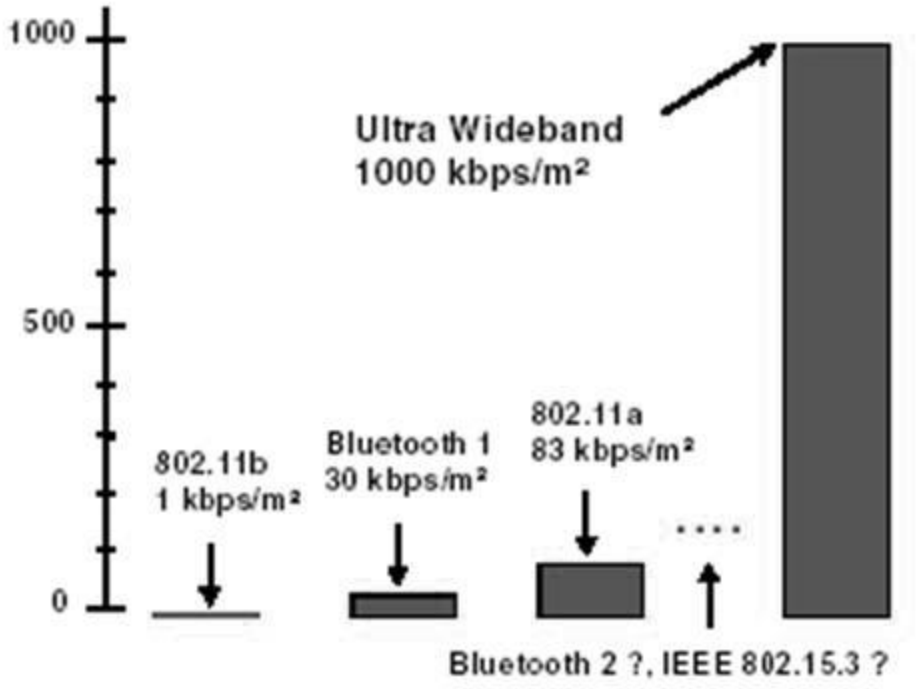


Figure 1.3 Maximum transfer rates of different technologies

1.5. Issues regarding regulations

According to the Federal Communications Commission (FCC) , every type of wireless transmission is defined as UWB when it is characterized by a) fractional spectrum range W/fc ≥ 20 %, where W is the transmission bandwidth and fc is the central spectrum frequency or b) at least 500 MHz of bandwidth. FCC approved the operation of UWB systems in the band range 3.1 – 10.6 GHz under the restrictions of a modified version of the regulations 15.209. According to the commission, the power spectral density (PSD) of the transmitted signal must satisfy the relevant transmission masks. The transmission mask for indoor applications, as it is defined by FCC, is provided below.

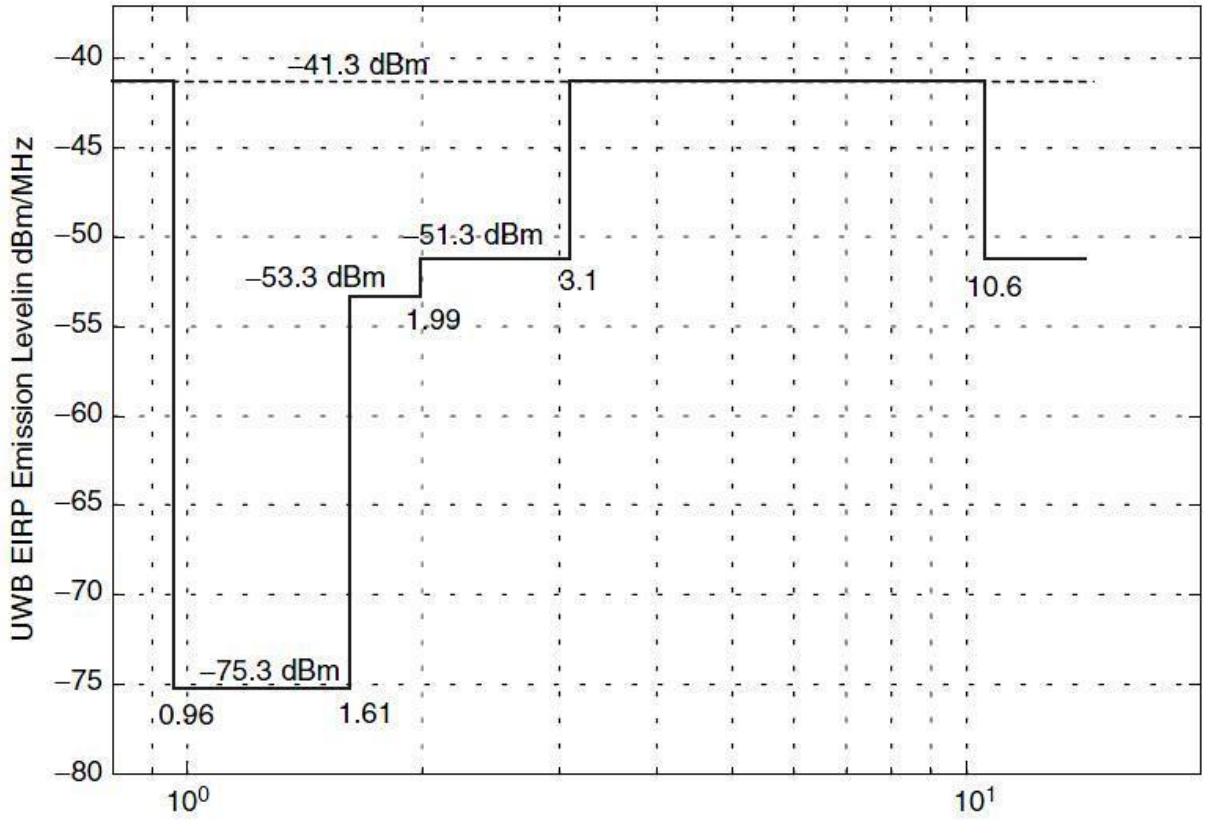


Figure 1.4 FCC spectral mask for indoor applications

1.6. Applications

Due to the especially low allowable transmission power, UWB systems are suitable mainly for indoor and short distance applications. However, due to the short duration of the UWB pulses, it is easier to achieve very high transmission rates. In addition, it is possible to easily exchange the transmission rate with the transmission range by simply varying the pulse energy per data bit using either spread spectrum techniques or coding. Furthermore, UWB's high transmission rates allow the generation of wireless screens, the efficient data transfer from digital cameras, wireless printing, swift file transmission between mobile phones. In other words, UWB promises to be the wireless alternative to the USB cable.

Another application of UWB is position detection. The astonishing accuracy that UWB promises, combined with the extremely low power levels, mean that it is ideal for wireless services in specific environments, such as a hospital. Another advantage of UWB is the short transmission time, which allows the setup of transmitter tags that surpass those attainable using the conventional technologies by several orders of magnitude. UWB is also used in radars that can see through walls, in position detection applications with high precision by measuring the distances between the transceivers, along with position detection applications where the precise measurement of the signal's time of arrival technique is used.

In addition, UWB was suggested as the most suitable technology for Personal Area Networks – PAN, and appeared in the 802.15.3a PAN standard of IEEE. However, after many years of stagnation, the 802.15.3a group of IEEE was disbanded in 2006. The slow progress in the development of UWB standards, the high cost of the initial implementations and the considerable lower performance compared with the one initially expected are some of the reasons that lead to the reduced success of the UWB technology in commercial products, which lead quite a few companies to discontinue their relevant pursuits during 2008 and 2009.

1.7. UWB transmission modes

UWB appears in two basic modes: with carrier and without carrier. Initially, UWB systems were based on the transmission of exceptionally short pulses (the so-called IR – impulse radio). Then, however, the regulation of 2002 was published by FCC, which defines as UWB every signal with fractional bandwidth greater than 0.2 or signal bandwidth greater than 500 MHZ. In this way, thanks to the new regulation there is a variety of already established wireless transmission technologies that fall in the general UWB category. An example is the multiband UWB, where the total bandwidth is split into multiple smaller bands, each of which fulfills the FCC criterion. Another example is the OFDM multicarrier transmission system, which can be applied in UWB systems. In general, systems with carrier possibly take better advantage of the available spectrum, but IR systems require lower implementation cost. **In this thesis we will only focus on IR UWB.**

1.8. Pulse Shaping

Here the transmitted signal consists of Gaussian pulses. It is interesting that Gaussian pulses are practically more easily realized, compared to sinusoidal signals. The transmission of this type of pulses using the conventional CMOS technology was made possible after the invention of the UWB LCR (Large Current Radiator) antennas by Harmuth in 1990. When a pulse current flows through these antennas, they generate a pulse that is transmitted through the air.

The most easily generated pulse is of Gaussian type. Its shape is described mathematically as follows:

$$p(t) = \pm \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{t^2}{2\sigma^2}\right)} = \pm \frac{\sqrt{2}}{\alpha} e^{-\frac{2\pi t^2}{\alpha^2}}$$

where $\alpha^2 = 4\pi\sigma^2$ is the “shape coefficient” and σ^2 the variance.

The shape of the above pulse for different values of a is shown in the figure below:

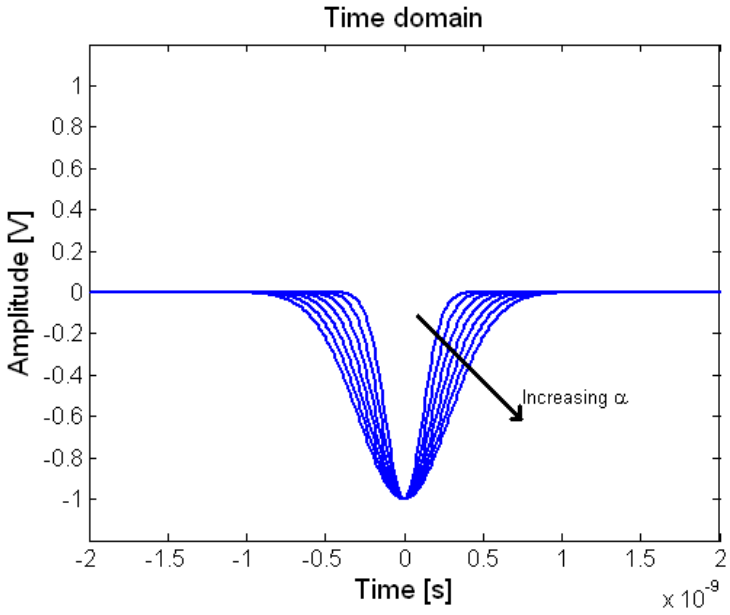


Figure 1.5 Shape of the Gaussian pulse

The power spectral density diagram of the above pulse family is displayed below:

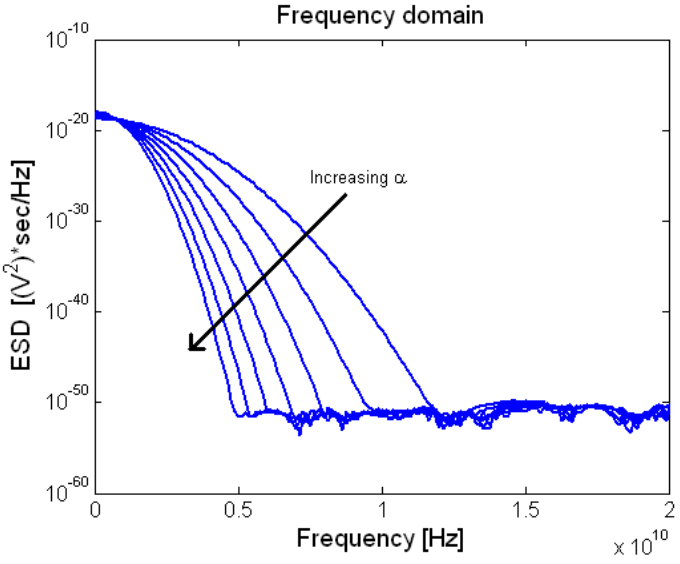


Figure 1.6 Spectral content of the Gaussian pulse

In order for the UWB pulse to be transmitted efficiently however, the pulse must necessarily be characterized by a zero direct current (dc) component. In other words the above diagram must be characterized by zero value at the zero frequency. Several waveforms that satisfy the above property have been devised, the most popular of which is modeled by the second derivative of the Gaussian function, which is described by the following relationship:

$$\frac{d^2 p(t)}{dt^2} = \left(1 - 4\pi \frac{t^2}{a^2}\right) e^{-\frac{2\pi t^2}{a^2}}$$

The second derivative of the Gaussian pulse is often mentioned as “the pulse at the receiver”, that is to say after it travels through the antenna of the transmitter and the antenna of the receiver. Ideally, a pulse of that shape appears after the transmitter's antenna when the transmitter's antenna is supplied with a current pulse that has the form of the first derivative of the Gaussian pulse (and consequently has zero dc component). In other words, the transmission antenna can be modeled as a differentiator. In contrast, the frequency response of the receiving antenna is almost flat.

Pulses of other different shapes have been suggested, for example Laplacian pulses and greater order derivatives of the Gaussian pulse. The Gaussian pulse, along with its first 15 derivatives, is depicted in the diagram below.

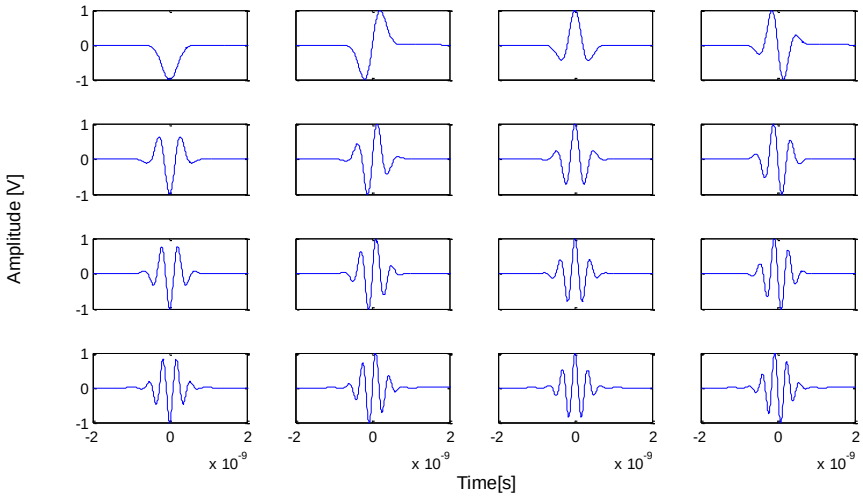


Figure 1.7 Gaussian pulse and its 15 first derivatives

1.9. Modulation techniques

The exceptionally large frequency bandwidth displayed by UWB creates new challenges with regard to the search of suitable modulation techniques. There has been extensive research on that particular subject over the last years. There are quite a few available options, every one of which has different advantages depending on the criterion in question, e.g. the application nature, the limitations set by the specifications, the range, the quality of the supplied service, the satisfaction of the regulations, the transmission rate, the channel reliability, the capacity, the performance depending on the noise levels in the channel. The most studied UWB modulation techniques are BPSK, QPSK, PAM, OOK, PPM, PIM and PSM. BPSK, which is used in this diploma thesis, is probably the most popular because it has a smooth frequency spectrum and low BER. However, the precise phase estimation in BPSK requires good channel estimation. This is a disadvantage of BPSK compared to OOK and PPM where the signal either exists or it does not and consequently no channel estimation is required.

Some of the most popular modulation techniques are displayed below:

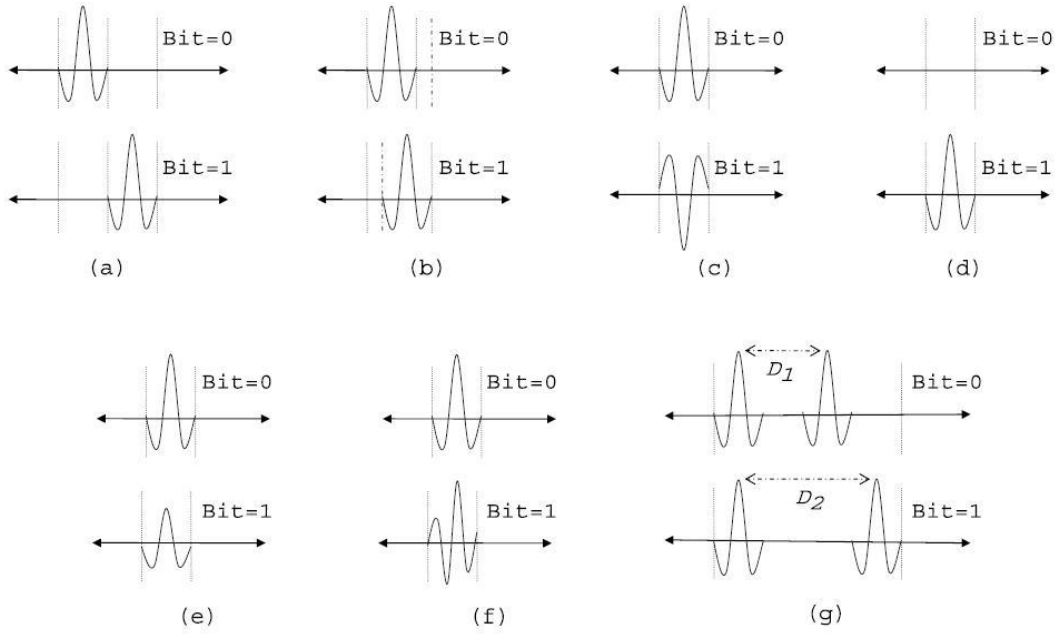


Figure 1.8 (a)Orthogonal PPM (b) Optimal PPM (c) BPSK (d) OOK (e) PAM (f)PSM (g) PIM

In this diploma thesis we use **BPSK modulation.**

After one of the above modulation techniques has been selected, so that multiple users are able to transmit simultaneously, either the Time Hopping technique (TH) or the Direct Sequence (DS) is applied. Both of these are analyzed below.

1.10. TH-UWB-IR

Here, every symbol period is divided into N time slots (in the figure N=5). At every period, the transmitter transmits in a different time slot, based on a random code.

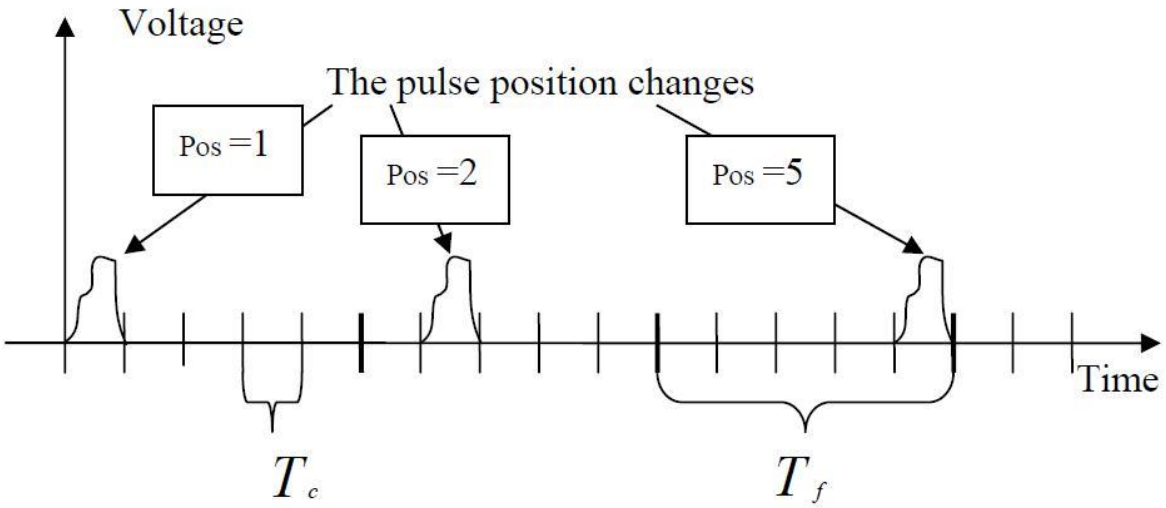


Figure 1.9 Time Hopping (TH) technique

1.11. DS-UWB-IR

Here, for every information bit, the transmitter transmits N pulses (in the figure N=8), based on a randomly generated code (PN code), which is though designed based on some rules so as to showcase the best possible properties. In this way, we achieve better tolerance to interference, simultaneous channel use by multiple users, satisfying system operation even under very low SNR values (often the power of the signal is lower than that of noise). However, a disadvantage of this method is that it sacrifices bitrate in order to achieve the aforementioned advantages.

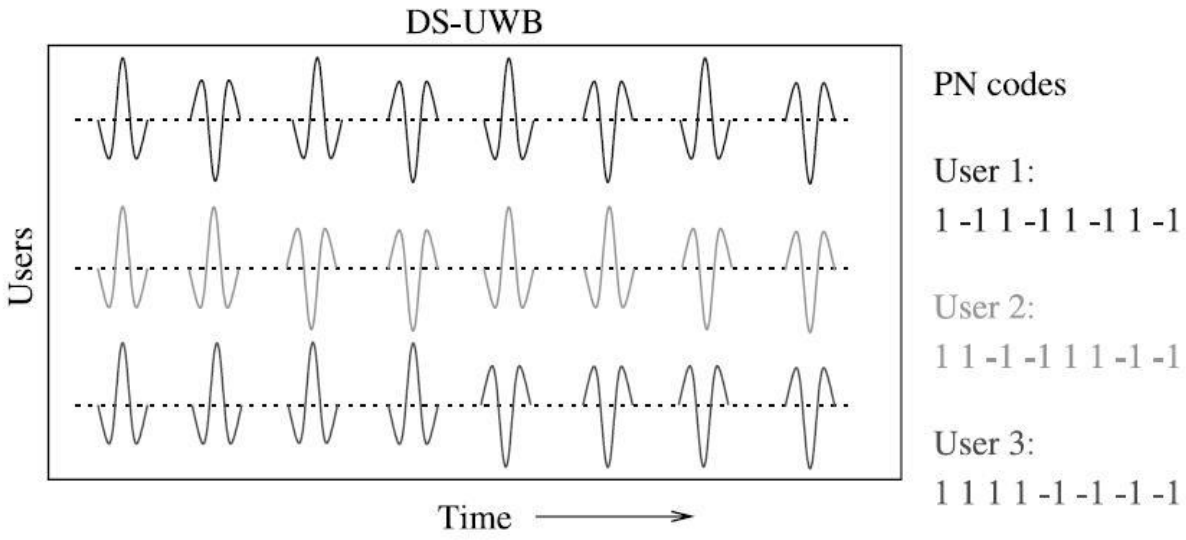


Figure 1.10 Direct Sequence – DS technique with BPSK modulation

This is the technique used in this diploma thesis, this is why we will further elaborate on it below.

1.11.1 Generation of DS-UWB pulses

The DS-UWB (Direct Sequence-Ultra Wideband) Transmission transmitter can be modeled as follows:

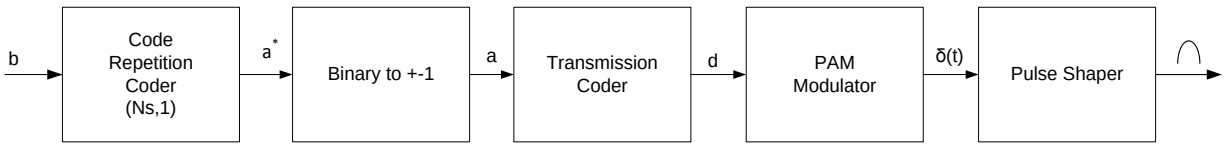


Figure 1.11 Generation of DS-UWB pulses

Let the input signal (the sequence to be transmitted) be $b= (b_0, b_1, b_2, \ldots, b_k, b_{k+1}, \ldots)$ which is generated at a rate of $R_b=1/T_b$ bits/s. The first block repeats every bit N_s times, thus creating the binary sequence $a^* = (b_0, b_0, \ldots, b_0, b_1, b_1, \ldots, b_1, \ldots, b_k, b_k, \ldots, b_k, \ldots)$, which is generated at a rate of $R_{cb} = N_s/T_b = 1/T_s$ bits/s .

The second block transforms the above bit sequence (0 or 1) into a sequence consisting of +-1. The new sequence is $a=(\alpha_0, \alpha_1, \ldots, \alpha_j, \alpha_{j+1}, \ldots)$.

The transmission encoder multiplies the above sequence with the PN code, which consists of +-1 and has a period equal to N_s. The new sequence $d = a \cdot c$ is thus generated, at a rate of $R_c = 1/T_s$ bits/s.

The d sequence enters the PAM modulator, which converts the +-1 into Dirac pulses at a rate of $R_P = N_s/T_b = 1/T_s$ pulses/s.

The output of the modulator enters the pulse shaper filter which is characterized by an impulse response of p(t). The impulse response p(t) is shaped like a Gaussian pulse with width considerably smaller than T_{chip}.

The output signal of the above system can be expressed as:

$$s(t) = \sum_{j=-\infty}^{+\infty} d_j p(t - jT_s) \quad ,$$

where d_j = +-1 and T_s the chip time (see below).

A typical example of a DS-UWB waveform transmitted by the transmitter is illustrated below:

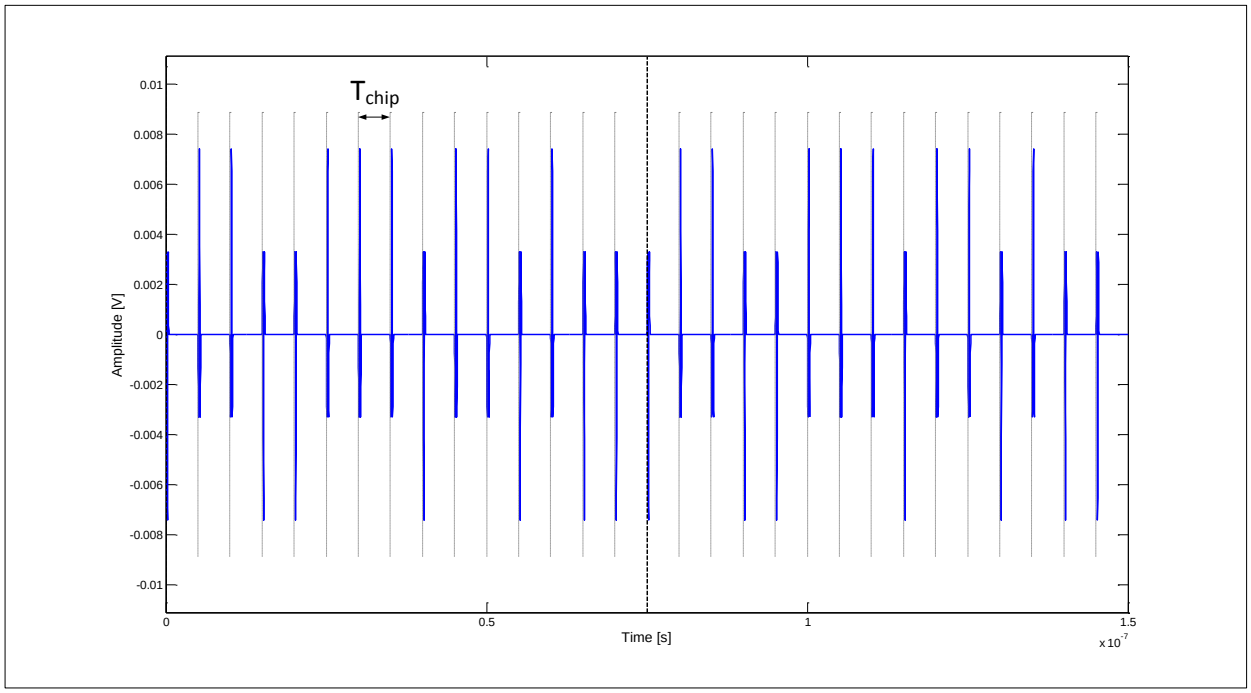


Figure 1.12 : DS-UWB waveform

The above figure corresponds to a DS-UWB signal with:

- T_{chip} = 5ns
- number of transmitted bits equal to two
- PN sequence = [-1 1 1 -1 -1 1 1 1 -1 1 1 -1 1 -1 -1]
- Pulse width of 0.5ns
- N_s = 15

1.11.2 Broadening codes (Pseudo Noise - PN codes)

As seen above, in Direct Sequence Spread Spectrum systems, for every information bit the transmitter transmits N_s pulses, based on a pseudorandom sequence c. The selection of this sequence is crucial, since it affects the system performance along with the maximum allowed number of users in the system. For this reason the generation of this kind of sequences constitutes an independent research area. In order for a sequence to be suitable for use in a Spread Spectrum system, it must possess certain basic attributes, which are mentioned below:

- **Balance.** The number of 1s in the sequence must differ from the number of -1s by one at most. In this way, the carrier compression is limited during its modulation by the PN code, since the carrier compression is significantly affected by the symmetry of the system to be modulated.

- **Autocorrelation.** The Autocorrelation function of a periodic sequence is defined as:

$$R_a(t) = \int_{-N_s \cdot T_{chip/2}}^{N_s \cdot T_{chip/2}} PN(t) \cdot PN(t+\tau) dt$$

or else as

$$R_a(k) = \sum_{n=0}^{N_s - 1} PN(n) \cdot PN(n+k)$$

It is desirable for the autocorrelation of the PN sequences to approximate as much as possible the autocorrelation of white noise, which is depicted in the figure below:

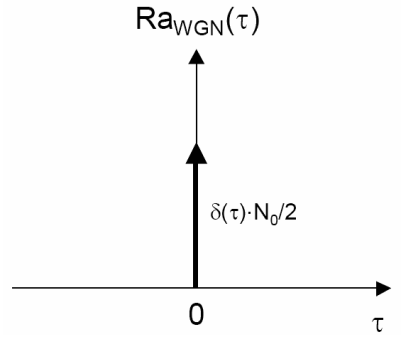


Figure 1.13 Autocorrelation of white noise

Below the autocorrelation of the PN sequence used in our system is shown for various τ:

<u>τ=0</u>

PN(0)	1	-1	1	1	-1	-1	1	-1	-1	-1	1	1	1	1	-1
PN(0)	1	-1	1	1	-1	-1	1	-1	-1	-1	1	1	1	1	-1
Product	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$R_a(\tau=0) = 15$

<u>τ=1</u>

PN(0)	1	-1	1	1	-1	-1	1	-1	-1	-1	1	1	1	1	-1
PN(1)	-1	1	-1	1	1	-1	-1	1	-1	-1	-1	1	1	1	1
Product	-1	-1	-1	1	-1	1	-1	-1	1	1	-1	1	1	1	-1

$R_a(\tau=1) = -1$

<u>τ=2</u>

PN(0)	1	-1	1	1	-1	-1	1	-1	-1	-1	1	1	1	1	-1
PN(2)	1	-1	1	-1	1	1	-1	-1	1	-1	-1	-1	1	1	1
Product	1	1	1	-1	-1	-1	-1	1	-1	1	-1	-1	1	1	-1

$R_a(\tau=2) = -1$

It stands that for every τ unequal to zero, where $-15<\tau<15$: $R_a(\tau)=-1$.
Graphically:

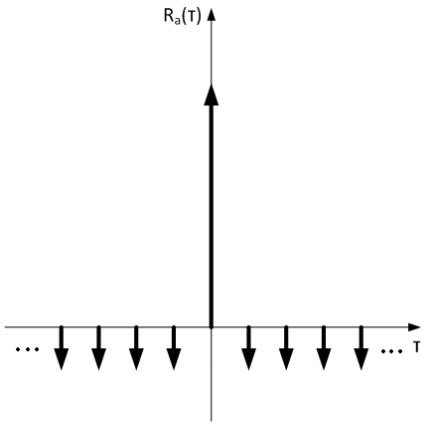


Figure 1.14 Autocorrelation of a real PN sequence

It is observed that the autocorrelation displays a high maximum point, which is also the case in white noise. The synchronization of the receiver is based on this exact property. Similarly, the channel estimation algorithm used is also based on this property.

- **Cross-correlation.** Cross-correlation is given by:

$$R_c(k) = \sum_{n=0}^{N_c-1} PN_i(n) \cdot PN_j(n+k)$$

In Spread Spectrum systems, user multiplexing is achieved through the use of a different PN code by each user. In order for more than one user in the channel to be able to transmit and receive simultaneously, the different PN codes are required to be orthogonal, unequal to zero. However, this requirement is in practice in conflict with the requirement for high autocorrelation, as was analyzed above. A trade-off thus takes place, so that the system can satisfy the needs of its users.

1.11.3 Generation of PN codes

Up to now, several techniques have been devised in order for suitable PN codes to be generated, which can serve the needs for wireless communication of many users in the same channel. The m-sequence codes, the gold codes, the Hadamard-Walsh codes and the Orthogonal Variable Spreading Factor codes are some indicative ones.

1.12. The UWB channel

1.12.1 Channel modeling

The transmission of a signal inside a closed environment (such as a room for example), is mainly accompanied by the appearance of multiple delayed versions of the transmitted pulse, due to the multipaths of the system inside the closed space. The transmitted signal can be expressed as:

$$r(t) = \sum_{n=1}^{N(t)} a_n(t) \cdot p(t - \tau_n(t)) + n(t)$$

where $a_n(t)$ and $\tau_n(t)$ are the channel gain and the channel delay measured at time t for the n^{th} path. N(t) is the number of multipath components at time t and n(t) is the noise at the receiver.

It stands that

$$r(t) = s(t) * h(t) + n(t)$$

where h(t) is the channel impulse response. Consequently, for the above channel it stands that:

$$h(t) = \sum_{n=1}^{N(t)} a_n(t) \cdot \delta(t - \tau_n(t))$$

Since the channel is slow-varying compared to the pulse transfer rate we can derive that:

$$h(t) = \sum_{n=1}^{N} a_n \cdot \delta(t - \tau_n)$$

The above channel model is the Turin model proposed in 1956 by Turin. The above model however has the significant disadvantage that it does not account for the distortion of the shape of the pulses due to reflection or penetration in various materials, when it regards IR (Impulse Radio) transmission. Every path should be characterized by its own shape pulse. The signal at the receiver is then:

$$r(t) = \sum_{n=1}^{N} a_n \cdot p_n(t - \tau_n) + n(t)$$

In the channel model used here, however, the shape change of the pulse is not taken into account.

1.12.2 Discretization of the channel impulse response

As was suggested by Hashemi in 1993, it would be more convenient if a discrete channel impulse response were to be adopted, for the description of the wireless channels affected by the multipath phenomenon. In that model, the time axis is divided into small time segments, called bins, which either include a multipath component or nothing at all. In other words, no more than one multipath components are allowed to correspond to a bin. The bin time is equal to the largest time segment for which the receiver cannot "realize" the existence of more than one multipath components, which means that it is equal to the analysis of the devices used in channel estimation.

The introduction of the above model simplifies significantly not only the analysis but also the simulation of the multipath channel behavior. The channel impulse response is then modeled as:

$$h(t) = \sum_{n=1}^{N_{\max}} a_n \delta(t - n \cdot \Delta\tau)$$

where N_{max} is the maximum number of bins taken into account and $\Delta\tau$ is the duration of the bin.

1.12.3 The UWB channel model proposed by IEEE

After a number of measurements of different UWB channels conducted by various researchers, it was ascertained that the UWB channel displays a certain unconventionality: the multipath components appear in clusters, as is shown in the figure below:

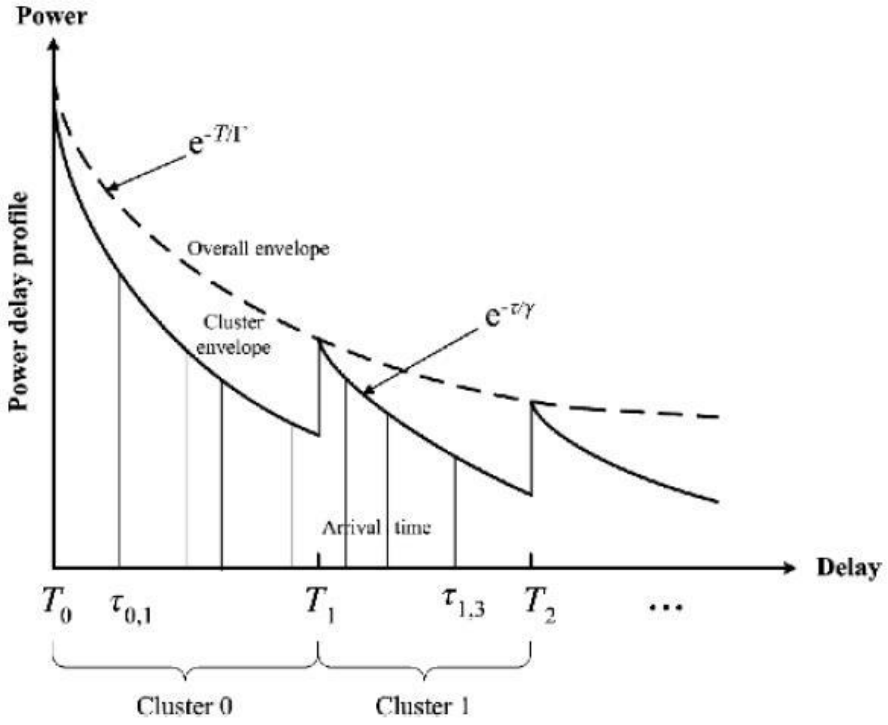


Figure 1.15 Channel with clusters of multipath components

The official model for the UWB channel (IEEE P802.15.3a UWB channel model) that was adopted by IEEE follows the Saleh-Valenzuela model with some small corrections. As was found through measurements, the magnitude distribution of the multipath components is better modeled using the Log-normal distribution than the Rayleigh distribution. A second important correction is that here the channel coefficients are real numbers. As can be observed from the below relationship, a separate attenuation coefficient is used for every cluster, but also for every radius belonging to the cluster. The channel impulse response is modeled as follows:

$$h_i(t) = X_i \sum_{l=0}^{L} \sum_{k=0}^{K} a^i{}_{k,l} \delta(t - T_l^i - \tau^i{}_{k,l})$$

Where:

$a^i{}_{k,l}$ are the magnitudes of the multipath components

T_l^i is the delay of the l[th] cluster

$\tau^i{}_{k,l}$ is the delay of the κ[th] radius compared to the arrival of the l[th] cluster

X_i represents the log-normal shadowing

i refers to the i[th] version of the channel

Through measurements, IEEE modeled 4 UWB channels based on the below conditions:

a) Existence of Line of Sight (LOS) , 0-4 meters distance

b) Non-existence of Line of Sight (NLOS) , 0-4 meters distance

c) Non-existence of Line of Sight (NLOS) , 4-10 meters distance

d) RMS delay spread equal to 25ns

The resulting channels are depicted in the figures below:

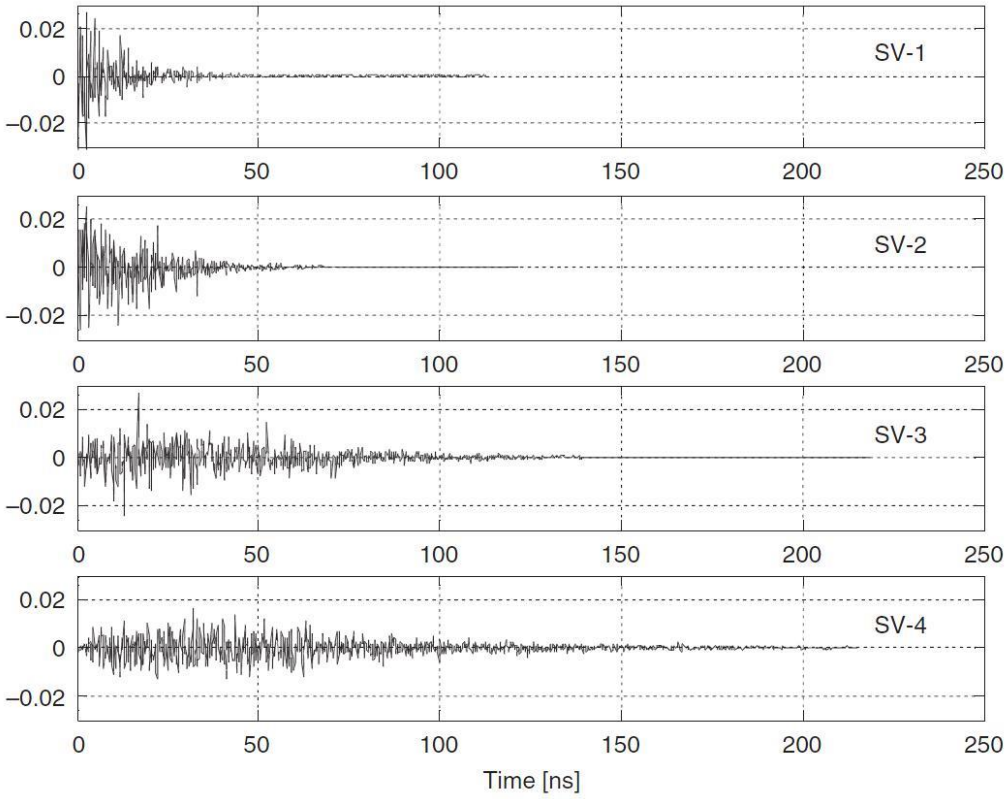


Figure 1.16 The 4 UWB channel models by IEEE

1.12.4 The discrete UWB channel model

Below we will calculate the discrete channel response corresponding to the first of the above IEEE channel models (case A). The selected bin time equals to 5 ns which is the period of the clock that the channel estimator circuit we designed operates at.

It is observed that in the relevant “continuous” channel model depicted in the above figure, the corresponding multipath components at 5ns are many more than one. However, as was stated previously, only one component must correspond to the time segment of one bin, for the discrete channel model. In order for the continuous channel to be modeled correctly using a discrete one, there are several methods. We will use the method according to which the component corresponding to each bin of the discrete

model results from calculating the average of all the components appearing during the corresponding time frame in the continuous model.

Using matlab, we obtain the following diagrams:

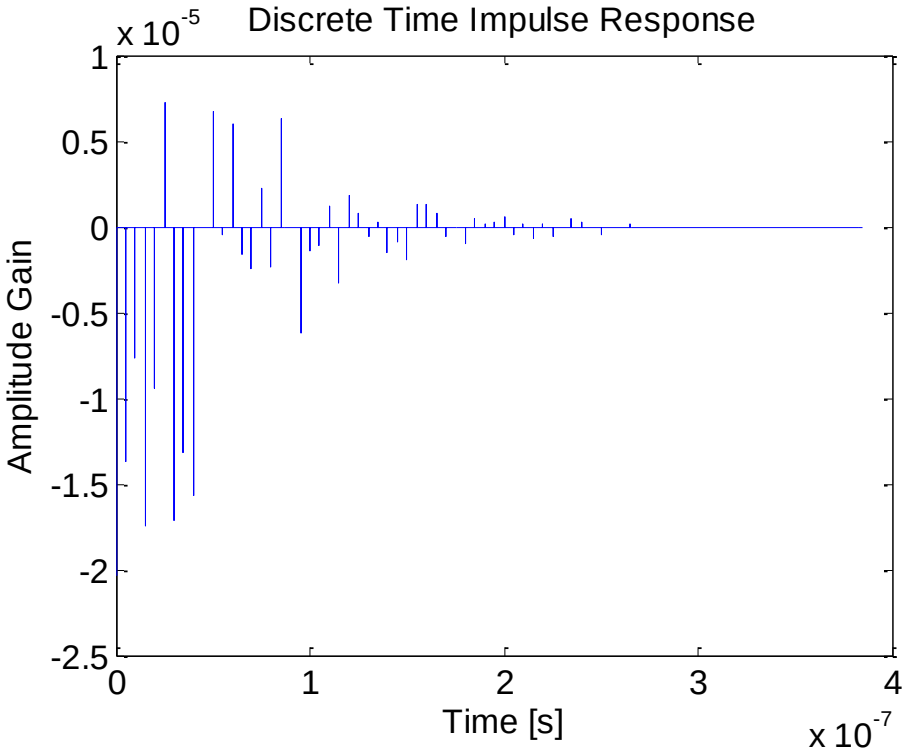


Figure 1.17 Equivalent discrete channel of the Case A continuous channel

Chapter 2 : Rake Receiver – Mathematical Analysis

2.1. RAKE Receiver

As was shown from the plot of the UWB channel impulse response (for all four models), the signal energy is split into tenths, if not hundreds, of multipath components. Consequently, a conventional receiver which would receive only one signal component would take no advantage of a considerable portion of the signal's power at the receiver. For this reason, the so-called “RAKE Receiver” is used, which was initially suggested by Price and Green in 1956. The RAKE receiver has a similar layout with a channel equalizer. Its main difference though, compared with the equalizer, is that the Rake receiver does not attempt to negate the interference caused by the multipath components, on the contrary it combines them with the desired component so that as much energy as possible can be concentrated at the receiver in order for us to have optimum BER performance with a given SNR at the transmitter.

The position of the RAKE circuit at the digital receiver is right before the decision circuit (detector). The preceding circuits are the RF circuits, the modified filter, the synchronization circuit, the analog to digital converter (ADC). In other words, the RAKE receiver processes digital samples of the input signal, which are received at time instances indicated by the synchronization circuit. The RAKE receiver implemented here is chip spaced, that is to say the ADC receives a sample every T, where T equals to the chip time of the UWB signal. In other words, we received a sample for every transmitted Gaussian pulse. The basic concept of the RAKE circuit is illustrated below:

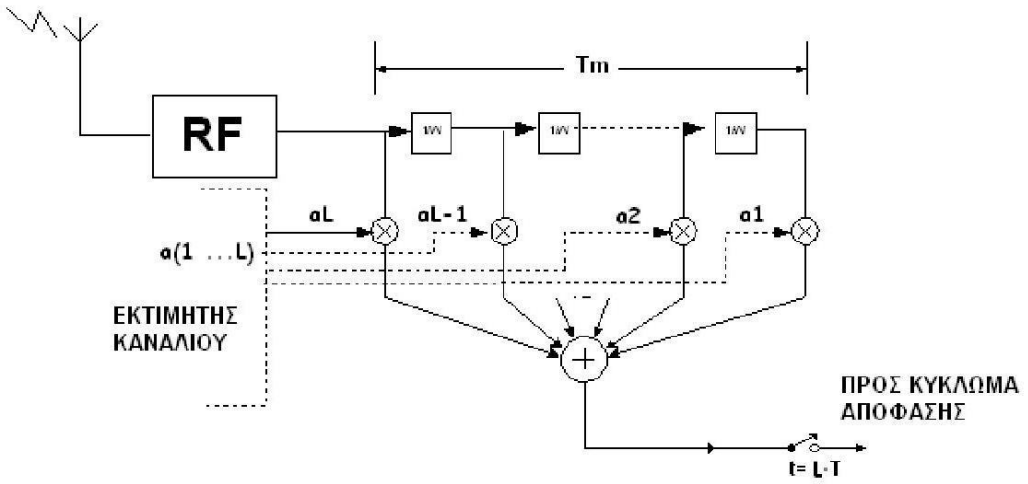


Figure 2.1 General form of the Rake Receiver

The received samples are directed to a sequence of delay circuits (similarly to an equalizer). The output of every delay circuit (tap) is multiplied with the corresponding channel coefficient calculated by the channel estimator circuit, the layout of which will be analyzed below. The products of the multiplications are summed and the resulting digital signal is directed to the decision circuit. In this way, energy collection from the signal's multipath components is achieved. The operation of the RAKE receiver is descriptively illustrated in the diagram below:

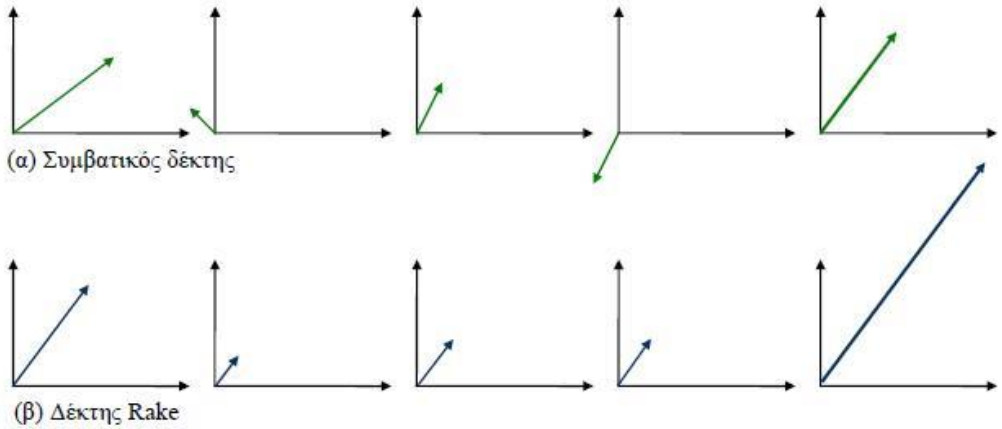


Figure 2.2 Utilization of multipath components in a simple receiver and in a Rake receiver

2.2. Comparison of Rake with Equalizer

In order for us to gain a deeper understanding of the operation of the Rake receiver, we will compare it with a similar system, the channel equalizer. The figure depicting the general form of the channel equalizer is shown below, along with a second figure depicting the most typical example of that circuit family, the MSE equalizer:

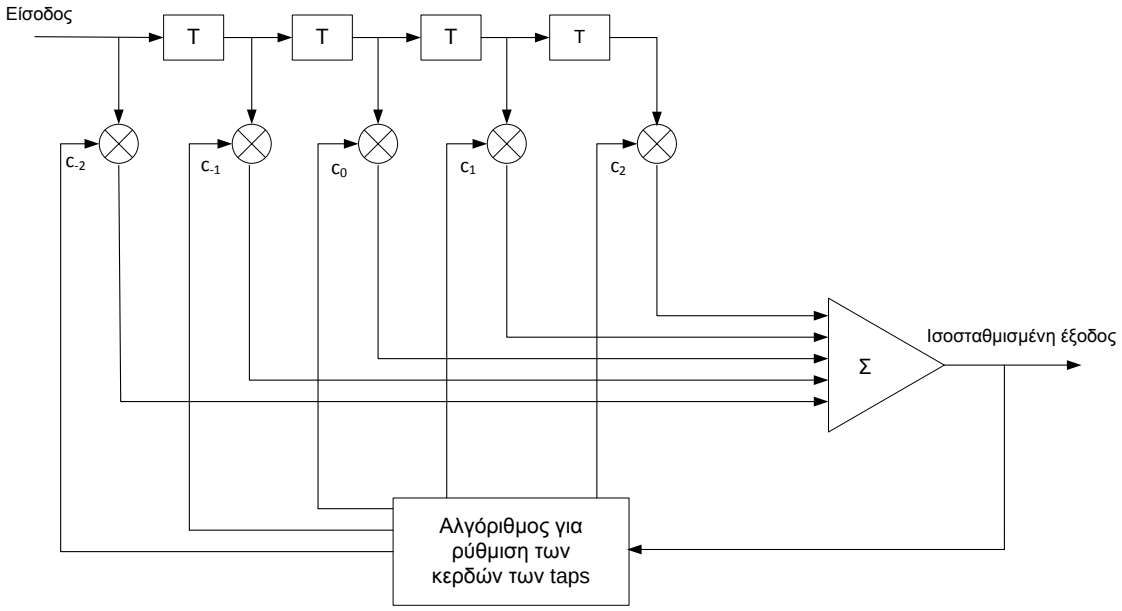


Figure 2.3 General form of the channel equalizer

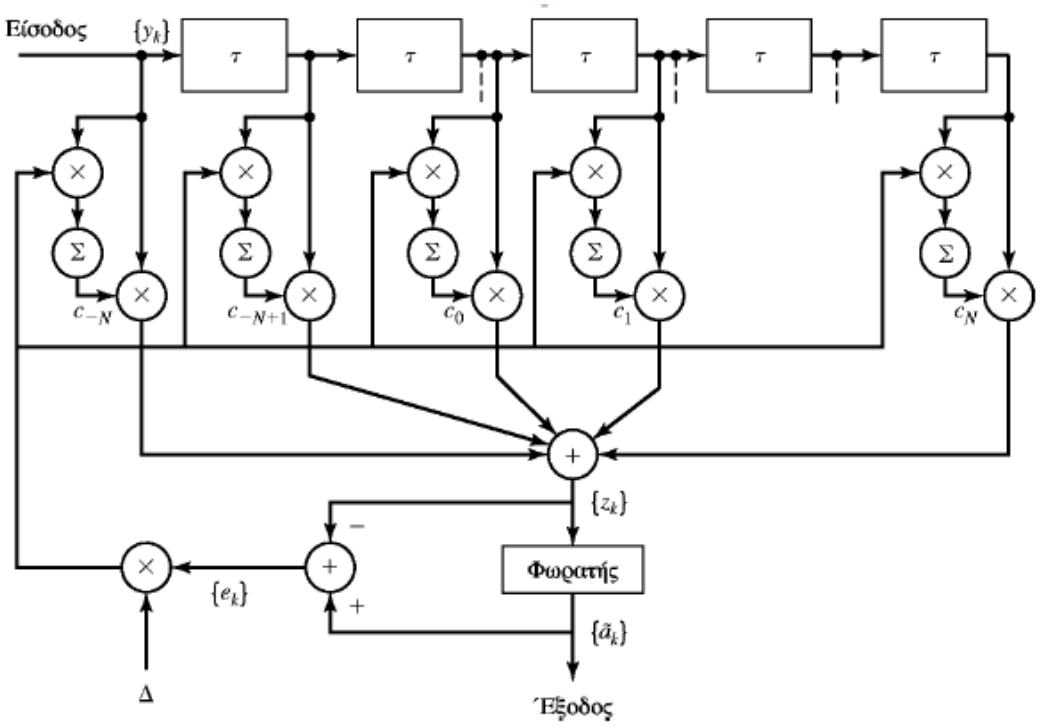


Figure 2.4 Typical example of an equalizer: The MSE equalizer

In the first figure, we observe the apparent similarities that exist between the layout of the equalizer and that of the Rake circuit. In the second figure however, we observe the difference between the two circuits: the equalizer attempts to adapt its coefficients in the best way possible so as to negate the impact of the multipath components, apart from one, the desirable one. This is achieved through the use of a feedback signal which is derived from the difference of the signal that results from the summation of the tap outputs with the symbol that finally decides that the equalizer was transmitted. In contrast, the RAKE receiver takes advantage of the multipath components instead of negating them.

2.3. RAKE Receiver for Direct sequence Spread-Spectrum (DSSS) Signals

Here, the decision circuit operates every N periods T_c, where T_c the time of one chip. It must also be taken into account that every transmitted pulse has been first multiplied with the corresponding value of the PN sequence. The RAKE receiver circuit is modified as follows:

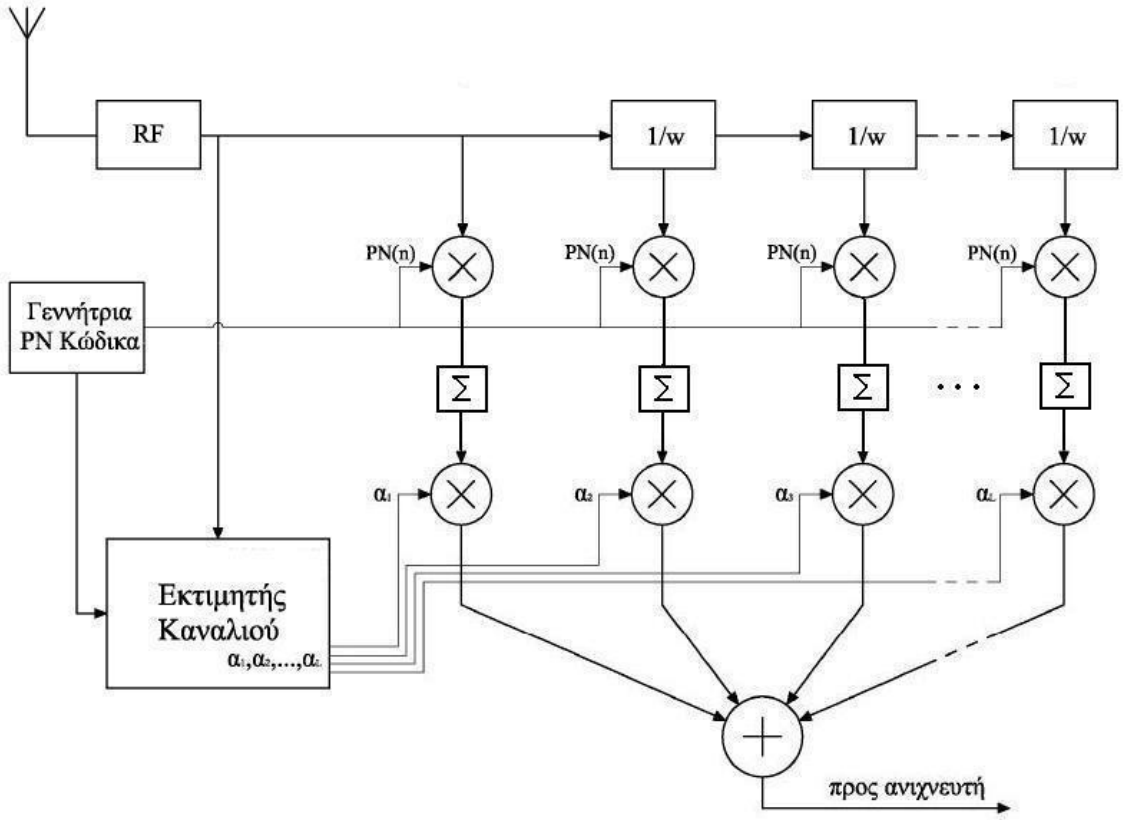


Figure 2.5 Architecture of Rake receiver for DSSS signals

It is observed that some accumulators, a PN code generator (which is in essence a circular buffer containing the PN sequence) and some multipliers have been added. Here, the accumulators add up the N products derived from the PN multipliers (for PN(n)= PN_1,PN_2, … , PN_N) and their output is the sum of the N products. Then, they set their value to zero and begin to add up again from scratch the products that correspond to the new PN sequence, and thus to a new symbol.

2.4. Channel estimator

2.4.1. Theory

As shown by the RAKE receiver block diagram presented above, the design of a “channel estimator” is required, that is to say a circuit capable of supplying the RAKE receiver with the suitable coefficients, which it must multiply the delayed samples of the received signal with, so as to achieve the optimal symbol detection. It can be proven that the optimal coefficients, provided that the noise is white (AWGN), are equal to the coefficients of the channel impulse response. In this case, we have the Maximum Ratio Combining – MRC method, which according to the literature (Proakis 1995) yields the optimal results. This is reasonable, since in this way more weight is attributed to signals originating from the optimal multipaths which showcase better SNR, since the noise is white.

It is deducted from the above that the channel estimator must be capable of detecting the impulse response of the channel. However, there are two problems that arise in this case. Firstly, the channel is not constant and secondly, as was illustrated by the UWB channel model, the impulse response of the channel contains hundreds of components. The solution of the first problem is simple, if the channel is relatively slow-varying: we repeat the channel estimation over regular time intervals. The solution of the second problem is the use of a partial or selective rake receiver instead of the all-rake receiver. The basic concepts regarding the operation of these all-rake variants are illustrated in the figures below.

All-rake receiver

Here, the rake receiver has one tap for every component of the channel's impulse response (impossible)

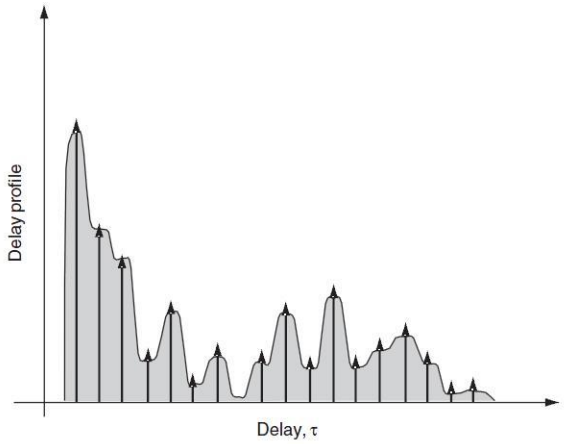


Figure 2.6 All-rake receiver

Selective-rake receiver

Here, the rake receiver uses the N_{sel} best multipath components of the channel (in the figure $N_{sel} = 7$)

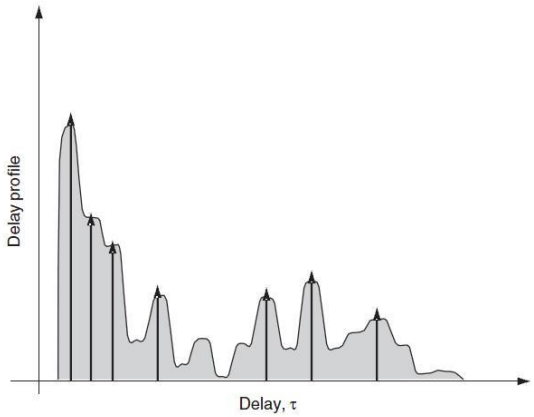


Figure 2.7 Selective-rake receiver

Partial-rake receiver

Here, only the first N_{par} multipath components of the channel's impulse response are used, because it has been observed that the first components of the impulse response are often the best.

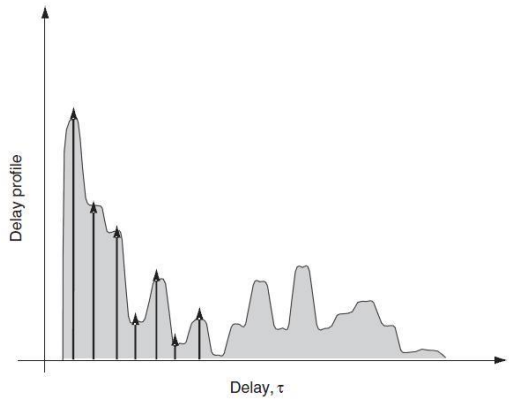


Figure 2.8 Partial-rake receiver

In this diploma thesis we will implement a simple rake and a selective rake receiver.

2.4.2. The channel estimation algorithm

The channel estimation algorithm that is implemented in this diploma thesis is based on the properties of the PN sequences (sequences from 1 or -1). The properties of the PN sequences that are of use here are the following:

- Balance. The number of 1s and -1s contained in a sequence is either the same or differs by one bit at most. Here, we assume that the number of 1s is higher by one, compared to the number of -1s, hence $\sum_{n=1}^{T} p(n) = 1$.
- The product PN(n)*PN(n) is always equal to 1.
- PN(n+T)=PN(n), where T is the length of the PN sequence.
- The PN sequence has very good autocorrelation properties

The autocorrelations of an ideal and of a real PN sequence are illustrated below:

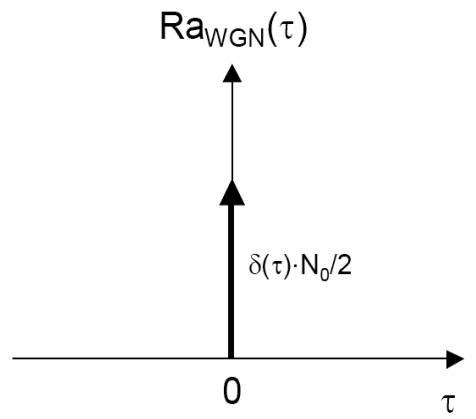


Figure 2.9 Autocorrelation of an ideal PN sequence

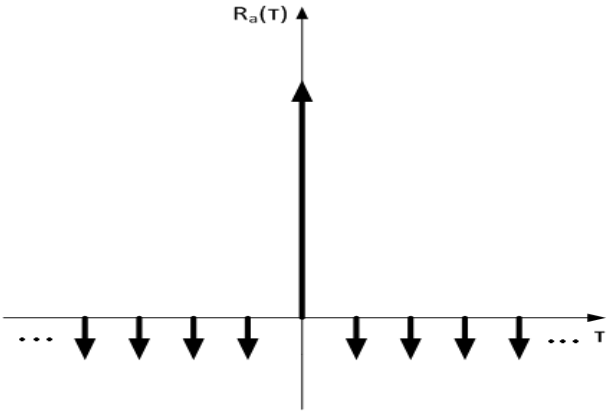


Figure 2.10 Autocorrelation of a real PN sequence

We will initially present the block diagram of the channel estimator that calculates the channel coefficients correctly, on condition that the PN sequence is ideal (has the same autocorrelation as white noise). We will later examine the required modifications in architecture in order for the channel coefficients to be correctly calculated for a non-ideal PN sequence as well. The block diagram for an ideal PN sequence is shown below:

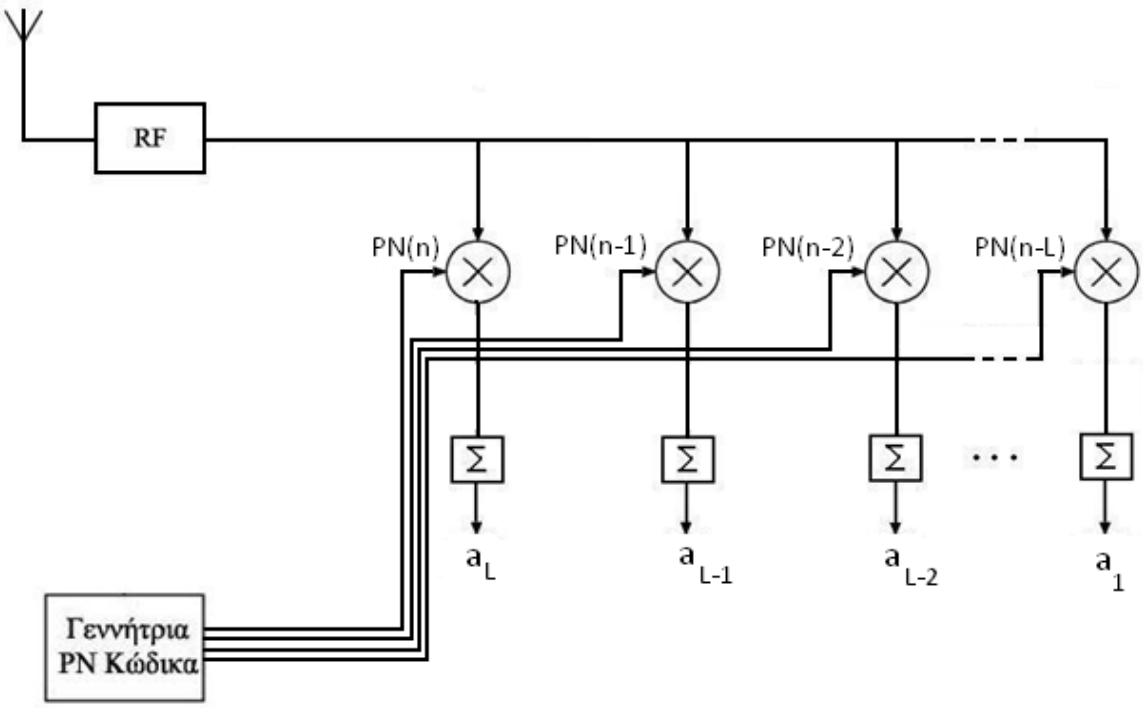


Figure 2.11 Architecture of a channel estimator for an ideal PN sequence

The input of the circuit y(t) is the signal that has been distorted by the channel, since it has been processed by the RF part of the receiver , while the outputs of the circuit are the calculated coefficients of the channel $\alpha_1 - \alpha_L$.

In order for the channel estimation to occur, the signal transmitted by the transmitter must be known to the receiver, at least until the calculation of the coefficients has been completed. As a "pilot" signal, the transmitter transmits the PN sequence twice, the period of which is $N \cdot T_c$, where T_c is the chip period and N is the number of chips contained in a sequence. Let h(t) be the channel impulse response , with a length of $L \cdot T_h$, where L are its components and T_h is the duration of each component. In order for the receiver to operate we must have $T_h \geq T_c$. Here, we also assume that $T_h = T_c = T$ and the channel impulse response becomes h(nT)=h(n).

The transmitter transmits the following signal:

$$x(n) = PN(n)$$

The signal received by the receiver is the convolution:

$$y(n) = x(n) * h(n) = \sum_{i=0}^{L-1} h(i) \cdot x(n-i) = \sum_{i=0}^{L-1} h(i) \cdot PN(n-i)$$

As shown by the block diagram of the estimator, the signal y(n) is multiplied in each tap with a different PN(n) and, after the accumulators, it is derived:

$$a_k = \sum_{n=0}^{N-1} y(n) \cdot PN(n-k) \qquad (2.1)$$

We claim that the resulting a_k from that equation are the coefficients of the channel's impulse response. This can be proven as follows:

$$a_k = \sum_{n=0}^{N-1} \sum_{i=0}^{L-1} h(i) \cdot PN(n-k) \cdot PN(n-i) \qquad (2.2)$$

From the properties of the (ideal) PN sequences we have that

$$PN(n-N) = PN(n)$$
$$PN(n) \cdot PN(n) = 1$$
$$\sum_{n=1}^{N} PN(n) = 1$$
$$PN(t) * PN(t-\tau) = \omega$$

where ω not equal to zero only for τ=0,T,2T,…

It is deducted from the above that, in order for the equation (2.2) to be non-zero, $k-i$ must be equal to a multiple of N , that is to say $i = l \cdot N + k$, $l = 0,1,2,3,...$

Since the values of both i and k fall inside [0,L-1] this equation can be modified to

$$i = k$$

As a result, (2.2) becomes:

$$a_k = \sum_{n=0}^{N-1} \sum_{i=0}^{L-1} h(i) \cdot PN(n-k) \cdot PN(n-i) \Leftrightarrow$$
$$a_k = \sum_{n=0}^{N-1} \sum_{i=0}^{L-1} h(k) \cdot PN(n-k) \cdot PN(n-k) \Leftrightarrow$$
$$a_k = \sum_{n=0}^{N-1} \sum_{i=0}^{L-1} h(k) \cdot 1 \Leftrightarrow$$
$$a_k = h(k)$$

Consequently, we have indeed proven that the outputs of the channel estimator correspond to the components of the channel's impulse response. In the application examined below, the modified layout of the channel estimator will also be presented.

2.4.3. Application

In order for us to gain a deeper understanding of the channel estimation process, we will provide a hypothetical example. As mentioned earlier, channel estimation takes place in a periodic fashion, and in order for it to be feasible, the transmitter must transmit two known PN sequences every time, as a pilot signal. It must also be mentioned here that not only in this specific example, but in the following simulations conducted as well, we have assumed that the noise in the channel has a zero value. If this is not the case, the estimation of the channel coefficients will not be precise, but it still will not differ from the actual values considerably. Let us define the following:

N number of chips in the sequence
Pni elements of the broadening code, i = 1, 2,…, N
L number of components of the channel impulse response
c_k coefficients of the channel impulse response, k = 1, 2,…, L
R_i distorted by the channel signal that arrives at the receiver. The i numbering starts from the moment the PN_1 of the second PN arrives at the receiver

For this hypothetical example, we assume that N=L. In reality though, that is not true since the UWB channel contains hundreds of components. However, even if L > N, we would calculate the coefficients of the N first channel impulses.

As soon as we transmit the first bit of the PN sequence for the second time, we will receive at the receiver:

$$R_1 = PN_1 \cdot c_1 + PN_N \cdot c_2 + PN_{N-1} \cdot c_3 + \ldots + PN_2 \cdot c_N$$

We will then receive the following R signals:

$$R_2 = PN_2 \cdot c_1 + PN_1 \cdot c_2 + PN_N \cdot c_3 + \ldots + PN_3 \cdot c_N$$
$$R_3 = PN_3 \cdot c_1 + PN_2 \cdot c_2 + PN_1 \cdot c_3 + \ldots + PN_4 \cdot c_N$$
$$\vdots$$
$$R_N = PN_N \cdot c_1 + PN_{N-1} \cdot c_2 + PN_{N-2} \cdot c_3 + \ldots + PN_1 \cdot c_N$$

When R_1 is received at the receiver, the first tap will multiply it with PN_1, the second one with PN_2, the third one with PN_3, the N^{th} tap with PN_N. R_2 will arrive at the following time instance (after one T_{chip}). The first tap will multiply it with PN_2, the second one with PN_3, the third one with PN_4, the $N\text{-}1^{th}$ tap with PN_N and the N^{th} tap with PN_1. Consequently, the following sums will appear at the accumulators' outputs:

$$S_1 = R_1 \cdot PN_1 + R_2 \cdot PN_2 + \ldots + R_{N-1} \cdot PN_{N-1} + R_N \cdot PN_N$$
$$S_2 = R_1 \cdot PN_2 + R_2 \cdot PN_3 + \ldots + R_{N-1} \cdot PN_N + R_N \cdot PN_1$$
$$\vdots$$
$$S_N = R_1 \cdot PN_N + R_2 \cdot PN_1 + \ldots + R_{N-1} \cdot PN_{N-2} + R_N \cdot PN_{N-1}$$

The sum S_j results from each tap. In order for us to find the coefficients α_k we apply the equation (2.1). We will use a numerical example with a known PN sequence, the PN = 101100100011110.

The signals R_j are shown in the table below:

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
R_1=	1	-1	1	1	1	1	-1	-1	-1	1	-1	-1	1	1	-1
R_2=	-1	1	-1	1	1	1	1	-1	-1	-1	1	-1	-1	1	1
R_3=	1	-1	1	-1	1	1	1	1	-1	-1	-1	1	-1	-1	1
R_4=	1	1	-1	1	-1	1	1	1	1	-1	-1	-1	1	-1	-1
R_5=	-1	1	1	-1	1	-1	1	1	1	1	-1	-1	-1	1	-1
R_6=	-1	-1	1	1	-1	1	-1	1	1	1	1	-1	-1	-1	1
R_7=	1	-1	-1	1	1	-1	1	-1	1	1	1	1	-1	-1	-1
R_8=	-1	1	-1	-1	1	1	-1	1	-1	1	1	1	1	-1	-1

R_9=	-1	-1	1	-1	-1	1	1	-1	1	-1	1	1	1	1	-1
R_{10}=	-1	-1	-1	1	-1	-1	1	1	-1	1	-1	1	1	1	1
R_{11}=	1	-1	-1	-1	1	-1	-1	1	1	-1	1	-1	1	1	1
R_{12}=	1	1	-1	-1	-1	1	-1	-1	1	1	-1	1	-1	1	1
R_{13}=	1	1	1	-1	-1	-1	1	-1	-1	1	1	-1	1	-1	1
R_{14}=	1	1	1	1	-1	-1	-1	1	-1	-1	1	1	-1	1	-1
R_{15}=	-1	1	1	1	1	-1	-1	-1	1	-1	-1	1	1	-1	1
SUM	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The sum Rj is calculated in the last line. We observe that we have:

$$\sum_{j=1}^{N} R_j = \sum_{m=1}^{N} c_m$$

This was to be expected since, as was mentioned earlier, we have $\sum_{i=1}^{N} PN(i) = 1$.

Now that we have calculated the R_j elements, we will apply the equation (2.1) for κ=0:

$$\alpha_0 = \sum_{n=0}^{N-1} y(n) \cdot PN(n-k) = \sum_{n=0}^{14} y(n) \cdot PN(n) = \sum_{n=0}^{14} R_{n+1} \cdot PN(n)$$

The last equation is valid because we have represented the signal y(n) at the receiver as R_{n+1} (the numbering of the R elements starts from one instead of zero). Consequently, according to the above equation, we have:

$$\alpha_0 = \sum_{n=0}^{14} R_{n+1} \cdot PN(n) = S_1 =$$

$$= R_1 - R_2 + R_3 + R_4 - R_5 - R_6 + R_7 - R_8 - R_9 - R_{10} + R_{11} + R_{12} + R_{13} + R_{14} - R_{15}$$

We perform the corresponding calculations in the table presented above, which yields:

	· C_1	· C_2	· C_3	· C_4	· C_5	· C_6	· C_7	· C_8	· C_9	· C_{10}	· C_{11}	· C_{12}	· C_{13}	· C_{14}	· C_{15}
$R_{1=}$	1	-1	1	1	1	1	-1	-1	-1	1	-1	-1	1	1	-1
$R_{2=}$	-1	1	-1	1	1	1	1	-1	-1	-1	1	-1	-1	1	1
$R_{3=}$	1	-1	1	-1	1	1	1	1	-1	-1	-1	1	-1	-1	1
$R_{4=}$	1	1	-1	1	-1	1	1	1	1	-1	-1	-1	1	-1	-1
$R_{5=}$	-1	1	1	-1	1	-1	1	1	1	1	-1	-1	-1	1	-1
$R_{6=}$	-1	-1	1	1	-1	1	-1	1	1	1	1	-1	-1	-1	1
$R_{7=}$	1	-1	-1	1	1	-1	1	-1	1	1	1	1	-1	-1	-1
$R_{8=}$	-1	1	-1	-1	1	1	-1	1	-1	1	1	1	1	-1	-1
$R_{9=}$	-1	-1	1	-1	-1	1	1	-1	1	-1	1	1	1	1	-1
$R_{10=}$	-1	-1	-1	1	-1	-1	1	1	-1	1	-1	1	1	1	1
$R_{11=}$	1	-1	-1	-1	1	-1	-1	1	1	-1	1	-1	1	1	1
$R_{12=}$	1	1	-1	-1	-1	1	-1	-1	1	1	-1	1	-1	1	1
$R_{13=}$	1	1	1	-1	-1	-1	1	-1	-1	1	1	-1	1	-1	1
$R_{14=}$	1	1	1	1	-1	-1	-1	1	-1	-1	1	1	-1	1	-1
$R_{15=}$	-1	1	1	1	1	-1	-1	-1	1	-1	-1	1	1	-1	1
S_1	15	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

We observe that the equation (2.1) yields

$$\alpha_0 = 16 \cdot c_1 - (c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7 + c_8 + c_9 + c_{10} + c_{11} + c_{12} + c_{13} + c_{14} + c_{15}) =$$

$$= (N+1) \cdot c_1 - \sum_{m=1}^{N} C_m$$

Similar results can be obtained if we also calculate the rest of the α_k coefficients. Therefore, in order for us to calculate the precise c_k we merely need to apply the equation:

$$a_k = \frac{S_k + \sum_{m=1}^{N} C_m}{N+1}$$

However, we do not know the value of $\sum_{m=1}^{N} c_m$ at the receiver. It is though valid that $\sum_{m=1}^{N} c_m = \sum_{j=1}^{N} R_j$ so we calculate the channel coefficients as follows:

$$a_k = \frac{S_k + \sum_{m=1}^{N} c_m}{N+1} = \frac{S_k + \sum_{j=1}^{N} R_j}{N+1} = c_k$$

In order to confirm that the above equation indeed yields the corresponding channel coefficient we calculate α_1. Let us assume that PN = 101100100011110.

We have

$$\sum_{j=1}^{N} R_j + S_1 = \sum_{j=1}^{15} R_j + R_1 - R_2 + R_3 + R_4 - R_5 - R_6 + R_7 - R_8 - R_9 - R_{10} + R_{11} + R_{12} + R_{13} + R_{14} - R_{15}$$

$$= 2 \cdot (R_1 + R_3 + R_4 + R_7 + R_{11} + R_{12} + R_{13} + R_{14})$$

The above calculation is shown in the table below:

	· C_1	· C_2	· C_3	· C_4	· C_5	· C_6	· C_7	· C_8	· C_9	· C_{10}	· C_{11}	· C_{12}	· C_{13}	· C_{14}	· C_{15}
R_1=	1	-1	1	1	1	1	-1	-1	-1	1	-1	-1	1	1	-1
R_3=	1	-1	1	-1	1	1	1	1	-1	-1	-1	1	-1	-1	1
R_4=	1	1	-1	1	-1	1	1	1	1	-1	-1	-1	1	-1	-1
R_7=	1	-1	-1	1	1	-1	1	-1	1	1	1	1	-1	-1	-1
R_{11}=	1	-1	-1	-1	1	-1	-1	1	1	-1	1	-1	1	1	1
R_{12}=	1	1	-1	-1	-1	1	-1	-1	1	1	-1	1	-1	1	1
R_{13}=	1	1	1	-1	-1	-1	1	-1	-1	1	1	-1	1	-1	1
R_{14}=	1	1	1	1	-1	-1	-1	1	-1	-1	1	1	-1	1	-1
SUM=	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0

We then have that:

$$S_1 + \sum_{j=1}^{N} R_j = 2 \cdot (8 \cdot c_1) = 16 \cdot c_1$$

and

$$a_1 = \frac{S_1 + \sum_{j=1}^{N} R_j}{N+1} = \frac{16 \cdot c_1}{16} = c_1$$

Consequently, the calculated values are indeed equal to the channel coefficients. Below we can observe how the new algorithm is translated in hardware terms:

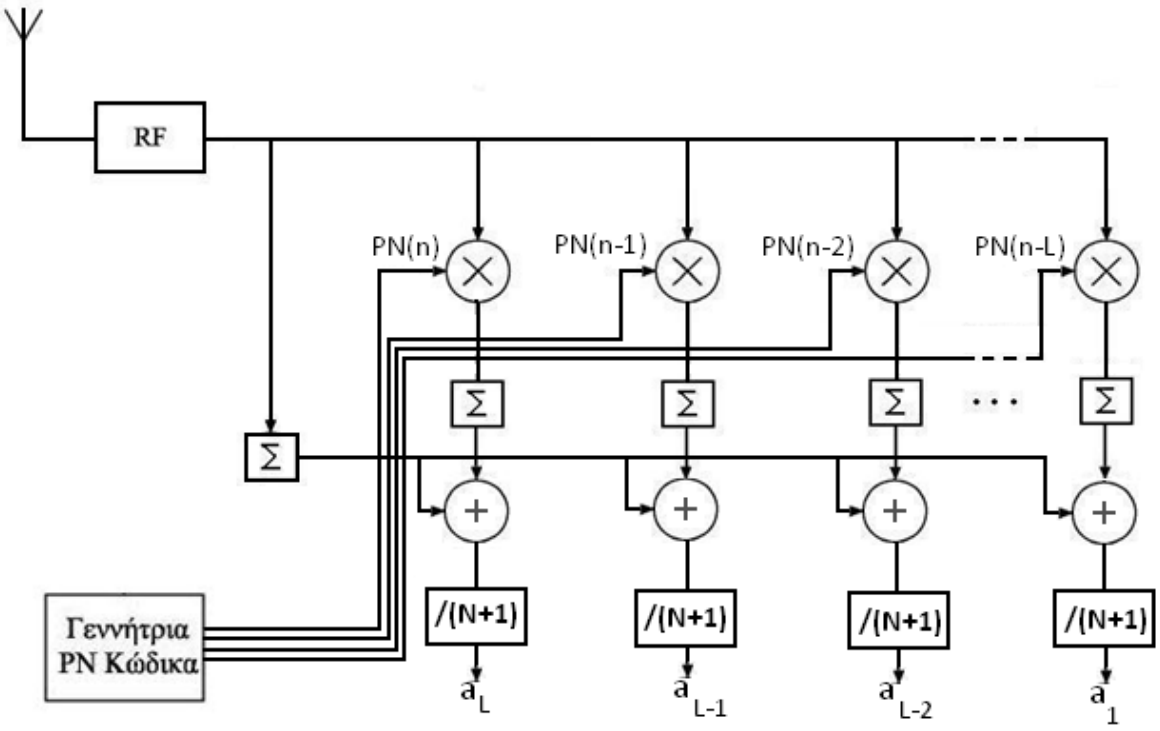


Figure 2.12 Architecture of a channel estimator for a real PN sequence

2.5. RAKE Receiver

2.5.1. Theory

Once channel estimation is complete, the transmitter starts sending the useful data to the receiver. Before sending the useful information signal, the receiver first broadens the information signal using the PN code.

The receiver receives the signal, which is distorted by the channel, and tries to take advantage in the best possible way of the different delayed signal components from the various multipaths, by using the RAKE circuit. The RAKE circuit has the following layout:

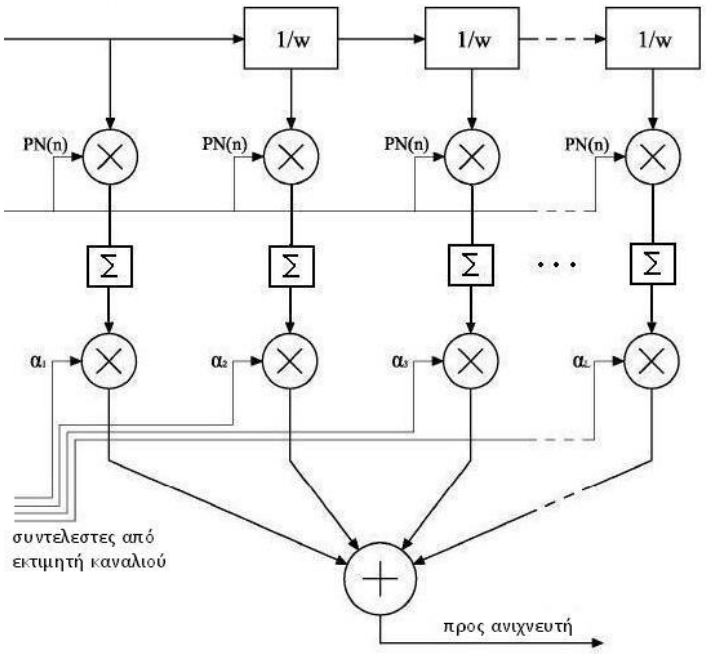


Figure 2.13 Architecture of the Rake receiver

The adders sum the multiplication results of the signal with the PN sequence and they clear their contents every N chip periods, thus starting the summing from scratch. Before clearing their contents, they send the sum that has been calculated up to that point to the multipliers, where the signal is multiplied with the channel weighted-coefficients that originate from the channel estimator. Then, the resulting products are

summed and sent to the detector. It is reminded that in this way we achieve the best possible detection since we have maximum ratio combining (MRC) (assuming that the channel coefficients have been precisely estimated).

2.5.2. Mathematical Analysis

The following is assumed:

- T_c is the chip period
- N is the number of bits of the PN sequence
- PN sequence with a length of $N \cdot T_c$
- Channel response h(t) with a length of $L \cdot T_h$, L is the number of components
- $T_h = T_c = T$
- Channel impulse response h(n)
- Information signal (before broadening) $d_m(t)$

According to the above, the signal transmitted by the transmitter is:

$$x(n) = d_m(n) \cdot PN(n)$$

The signal received by the receiver is:

$$y(n) = h(n) * x(n) = \sum_{i=0}^{L-1} h(i) \cdot x(n-i) = \sum_{i=0}^{L-1} h(i) \cdot d_m(n-i) \cdot PN(n-i)$$

At the k^{th} tap of the rake circuit, the following signal results after the accumulator:

$$A_k = \sum_{n=0}^{N-1} PN(n) \cdot y(n-k) \Rightarrow$$
$$A_k = \sum_{n=0}^{N-1} PN(n) \cdot \sum_{i=0}^{L-1} h(i) \cdot d_m(n-k-i) \cdot PN(n-k-i) \Rightarrow$$
$$A_k = \sum_{n=0}^{N-1} \sum_{i=0}^{L-1} h(i) \cdot PN(n) \cdot d_m(n-k-i) \cdot PN(n-k-i) \quad (2.3)$$

From the properties of the PN sequences we have that

$$PN(n-N)=PN(n)$$
$$PN(n)\cdot PN(n)=1$$
$$PN(t)*PN(t-\tau)=\omega$$

where ω is not equal to zero only for τ=0,T,2T,…

As a result, in order for (2.3) not to yield zero values, we must have

$$i=l\cdot N-k,$$
$$l=0,1,2,\ldots$$

and since $0\le i\le L-1$ and $0\le k\le L-1$ we will have

$$i=N-k$$

Consequently, (2.3) becomes

$$A_k=\sum_{n=0}^{N-1}\sum_{i=0}^{L-1}h(i)\cdot PN(n)\cdot d_m(n-k-i)\cdot PN(n-k-i)\Rightarrow$$
$$A_k=\sum_{n=0}^{N-1}\sum_{i=0}^{L-1}h(N-k)\cdot d_m(n-N)\cdot PN(n)\cdot PN(n-N)\Rightarrow$$
$$A_k=\sum_{n=0}^{N-1}\sum_{i=0}^{L-1}h(N-k)\cdot d_m(n-N)\cdot PN(n)\cdot PN(n)\Rightarrow$$
$$A_k=h(N-k)\cdot\sum_{n=0}^{N-1}d_m(n-N)$$

The $A_κ$ coefficients are multiplied with the corresponding channel coefficients $α_κ=h(N-κ)$ and after the final adder it is derived:

$$A_{final}=\sum_{k=0}^{L-1}\left(h^2(N-k)\cdot\sum_{n=0}^{N-1}d_m(n-N)\right)\Rightarrow$$
$$A_{final}=\sum_{k=0}^{L-1}h^2(N-k)\cdot\sum_{n=0}^{N-1}d_m(n-N)$$

This signal is then driven to the detector.

2.5.3. Application

As was shown by the above diagram where the layout of the RAKE receiver is presented, the RAKE circuit taps are supplied by a moving chain of delay circuits. As was also stated earlier, during the receiving process of an X symbol, the summers add N values and are then set to zero. A problem that can be observed here is the following: The accumulation is initiated once all the signals corresponding to the X symbol are inside the moving chain. From that point onwards though, during the next N clock cycles where accumulation continues, there are also signals corresponding to the symbol transmitted after X inside the chain. Therefore, the final resulting sum will also include terms that are not related to the X symbol. The impact of this problem is examined below.

Let us assume that the transmitter sends the following symbol sequence:

A,B,Γ,A,B,Γ,A,B,Γ,...

After the broadening, we will have:

$A_1,A_2,A_3,\ldots, A_N, B_1,B_2,B_3, ,\ldots,B_N, \Gamma_1,\Gamma_2,\Gamma_3,\ldots, \Gamma_N, A_1,A_2,A_3,\ldots,A_N,\ldots$

We are going to study what happens when we wish to detect the B symbol at the receiver. The following signals arrive at the receiver:

$$R_{B_1} = B_1 \cdot c_1 + A_N \cdot c_2 + A_{N-1} \cdot c_3 + \ldots + A_2 \cdot c_N$$
$$R_{B_2} = B_2 \cdot c_1 + B_1 \cdot c_2 + A_N \cdot c_3 + \ldots + A_3 \cdot c_N$$
$$\vdots$$
$$R_{B_N} = B_N \cdot c_1 + B_{N-1} \cdot c_2 + B_{N-2} \cdot c_3 + \ldots + B_1 \cdot c_N$$
$$R_{\Gamma_1} = \Gamma_1 \cdot c_1 + B_N \cdot c_2 + B_{N-1} \cdot c_3 + \ldots + B_2 \cdot c_N$$
$$\vdots$$
$$R_{\Gamma_N} = \Gamma_N \cdot c_1 + \Gamma_{N-1} \cdot c_2 + \Gamma_{N-2} \cdot c_3 + \ldots + \Gamma_1 \cdot c_N$$

We choose to initiate the accumulation with the aim of detecting the B symbol, at the time instance when the signal R_{B_1} enters the last tap. The accumulation ceases as soon as the signal R_{B_N} enters the first tap.

Let S_k be the output of each tap, after the multiplication with the corresponding channel coefficient. We have that:

$$S_1 = c_N \cdot (R_{B_N} \cdot PN_1 + R_{\Gamma_1} \cdot PN_2 + R_{\Gamma_2} \cdot PN_3 + ... + R_{\Gamma_{N-1}} \cdot PN_N)$$
$$S_2 = c_{N-1} \cdot (R_{B_{N-1}} \cdot PN_1 + R_{B_N} \cdot PN_2 + R_{\Gamma_1} \cdot PN_3 + ... + R_{\Gamma_{N-2}} \cdot PN_N)$$
$$S_3 = c_{N-2} \cdot (R_{B_{N-2}} \cdot PN_1 + R_{B_{N-1}} \cdot PN_2 + R_{B_N} \cdot PN_3 + ... + R_{\Gamma_{N-3}} \cdot PN_N)$$

.

.

.

$$S_N = c_N \cdot (R_{B_1} \cdot PN_1 + R_{B_2} \cdot PN_2 + R_{B_3} \cdot PN_3 + ... + R_{B_N} \cdot PN_N)$$

Finally, the RAKE circuit will add up all the above S_j terms and will send this sum to the detector. It is observed that the final sum contains not only terms of the desirable B symbol, but also terms of the Γ and A symbols (via R_B). Therefore, according to this model, (which was analyzed in the beginning of this example) we always have undesirable terms at the receiver, which are related to the following and previous symbol transmitted through the channel. Below we calculate the precise effect that each symbol has on the final sum.

We will provide an example for N=7, because for N=15 the required calculations are particularly extensive. Let us assume the PN sequence PN = 1110100. The final sum of the S_j terms is equal to:

$SUM= S_1 + S_2 + S_3 + S_4 + S_5 + S_6 + S_7 =$
$C_7\cdot(-BC_1-2BC_2+7BC_7-BC_3+BC_5+\Gamma C_2-2\Gamma C_5-\Gamma C_4-\Gamma C_6) + C_6\cdot(-2BC_1+7BC_6-AC_7-BC_2+BC_4+\Gamma C_1-2\Gamma C_4-\Gamma C_3-\Gamma C_5)+$
$C_5\cdot(7BC_5-AC_6-2AC_7-BC_1+BC_3+BC_7-2\Gamma C_3-\Gamma C_2-\Gamma C_4) + C_4\cdot(7BC_4-AC_5-2AC_6+BC_2-AC_7+BC_6-2\Gamma C_2-\Gamma C_1-\Gamma C_3)+$
$C_3\cdot(7BC_3-AC_4-2AC_5+BC_1-AC_6+BC_5-2\Gamma C_1-GC_2-BC_7) + C_2\cdot(7BC_2-AC_3-2AC_4-AC_5+AC_7+BC_4-2BC_7-\Gamma C_1-BC_6)+$
$C_1\cdot(7BC_1-AC_2-2AC_3-AC_4+AC_6+BC_3-2BC_6-BC_5-BC_7)$

After calculations, it is derived:

$SUM= B\cdot[7\cdot(c_1^2 + c_2^2 + c_3^2 + c_4^2 + c_5^2 + c_6^2 + c_7^2) + \Omega_B] + A\cdot\Omega_A + \Gamma\cdot\Omega_\Gamma$,

where Ω_B , Ω_A , Ω_Γ are polynomials with terms of the ac_ic_j form.

In addition, it is deducted by simulations that the coefficient that the B symbol is multiplied with is considerably greater than the coefficients Ω_A and Ω_Γ .

For instance, for $c_1=c_2=\ldots=c_7=1$ we have:

$SUM = 43\cdot B - 18\cdot\Gamma - 18\cdot A$

while for $c_1=7$, $c_2=6$, $c_3=5$, $c_4=4$, $c_5=3$, $c_6= 2$, $c_7=1$

$SUM = 980\cdot B - 322\cdot\Gamma - 322\cdot A$

It is observed that $\Omega_A = \Omega_\Gamma$.

Chapter 3 : Implementation

3.1. Introduction

In this chapter, we will thoroughly describe the architecture of the three circuits that we designed. The first one is a simple RAKE receiver, which implements the function that was briefly described in the previous chapter. The second one is a selective RAKE, which was based on the simple RAKE design. The third circuit is a simple RAKE receiver, which was based on the design of the first circuit, but has been suitably modified in order for its operating frequency to be increased significantly. The design was made in VHDL, using the software tool Xilinx ISE 9.2i. It must be explicitly stated that all three circuits are working with BPSK signals and as a result they manage real numbers, instead of complex ones.

3.2. Architecture of the simple RAKE receiver

The designed circuit consists of three main sub circuits:

- The channel estimator
- The RAKE circuit
- The control circuit

All three circuits are digital. The control circuit is the circuit that provides the necessary clock signals. We have selected N=15, which means that the PN sequence has 15 chips. In addition, the RAKE circuit and the channel estimator have 15 taps each. This means that the simple RAKE receiver will operate properly for a channel impulse response with a number of components less than or equal to 15. For a greater number of components, the receiver will probably be less effective.

Initially, the receiver receives the two non-modulated PN sequences that were sent by the transmitter. During that time period, only the channel estimator is working, while the RAKE circuit is disabled. Once the mission of this pilot signal is complete, the channel estimator produces the calculated channel coefficients and then becomes disabled. Now

is the turn of the RAKE circuit, which uses the coefficients calculated by the channel estimator and produces the signal that is going to be sent to the detector every N clock pulses. During the operation time frame of the RAKE circuit, the channel is assumed to remain unchanged and the channel coefficients are not refreshed. Following the arrival of a certain number of symbols, the RAKE circuit interrupts its operation and the channel estimator takes it from there. That number of symbols was not studied in this diploma thesis and constitutes an open subject. In order to calculate it, a suitable simulation must be conducted in the Simulink environment using a relevant varying UWB channel model.

The RF and synchronization circuits, the PN sequence generator circuit, as well as the detector circuit are assumed given and will not be dealt with in this diploma thesis.

3.2.1. Complete System

The complete system has the following inputs and outputs:

Inputs

- **PN_code**: the 15bit PN sequence that is externally defined
- **clk**: the system clock (period of 5ns)
- **rst**: Reset signal
- **en**: Enable signal
- **signal_real**: Input signal at the RAKE circuit

Outputs

- **estimated_signal**: The final signal driven to the detector (every 75ns)

The block diagram of the complete system is illustrated below:

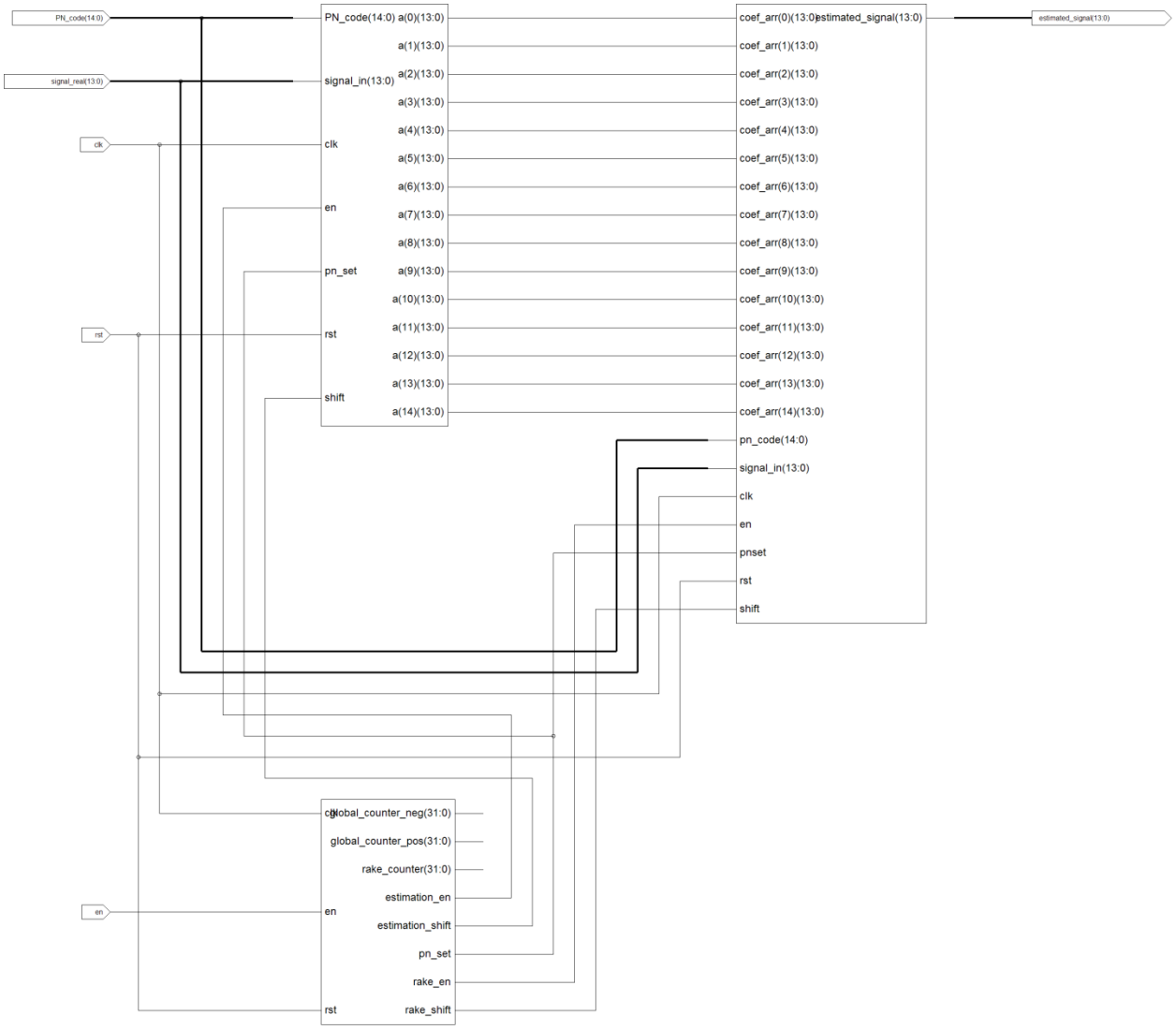


Figure 3.1 Architecture of the simple RAKE receiver

On the top left lies the channel estimator, on the bottom the control circuit, and on the top right lies the RAKE circuit. The receiver has been designed using higher level structural VHDL, while the basic components (summers, registers, multipliers, etc.) are implemented using behavioral VHDL.

3.2.2. Control Circuit

The control circuit is the circuit that regulates the clock cycles of the channel estimator and of the RAKE circuit. It also regulates when the registers containing the PN sequence shall get loaded. The PN sequence is introduced to the system by external means. The control circuit uses the following signals:

Inputs

- **clk**: The system clock (period of T=5 nsec)
- **rst**: The system reset signal
- **en**: The system enable signal

Outputs

- **rake_en**: This signal enables the RAKE circuit
- **estimation_en**: This signal enables the channel estimator
- **rake_shift**: Signal that defines when some flip-flops of the RAKE circuit shall reset
- **estimation_shift**: Signal that defines when some flip-flops of the channel estimator shall reset
- **pn_set**: The signal defines when the PN sequence shall get loaded onto the corresponding registers

The control circuit uses three internal counters so as to produce the control signals at the correct time instances. These counters are as follows:

- global_counter_pos: Counts the positive clock edges.
- global_counter_neg: Counts the negative clock edges.
- rake_counter: Becomes enabled when the RAKE receiver enters its normal mode, once the channel estimation is complete and the circular sequence of delay circuits of the RAKE circuit has been “filled” for the first time.

When en=1 and once reset=0, the global_counter_pos and global_counter_neg counters start counting, the first one the positive clock edges, while the second one the negative ones. The rake_en and estimation_en signals are controlled by the

global_counter_pos counter, while the rake_shift and estimation_shift signals by the global_counter_neg counter. We have therefore defined that the shift signals be enabled at the negative clock edges, while the enable signals at the positive ones.

The reason for choosing the shift signals to be enabled at the negative clock edges is the fact that the shift signal must arrive “a little later” than the positive clock edge so that the calculations in the accumulators of the channel estimator and rake circuit can be completed correctly. Since we wish to avoid the use of a clock with frequency multiple of that of the complete system clock, we use the negative clock edges in order to achieve this aforementioned “a little later”. In this way we provide the accumulator with time equal to T/2=2.5 nsec so that it is able to complete its last accumulation, before sending the accumulation result to the next stage of the circuit and before clearing its contents. The realization that the shift signals must follow the positive clock edges was one of the most challenging aspects of this diploma thesis.

Initially, once en=1 and reset=0, the channel estimator becomes enabled (estimation=1). The channel estimator requires two non-modulated PN sequences in order to calculate the channel coefficients. Therefore, N clock cycles after the initiation of its operation, it receives a signal from the control circuit and clears its accumulators. During the next N cycles, it calculates the channel coefficients and immediately afterwards receives yet again an estimation_shift signal. Then it sends the channel coefficients to its output and becomes disabled.

It is then time for the RAKE circuit to become enabled (rake_en=1). During the first N clock cycles, the sequence of delay circuits gets filled with the signal values from the channel. At the N^{th} clock cycle, rake_shift=1 is set instantaneously so that the accumulators can be cleared. During the next N clock cycles the real symbol is received and at the $2N^{th}$ clock cycle rake_shift=1 is set yet again, in this way we receive the signal to be detected at the rake circuit output.

3.2.3. Channel Estimator Circuit

The channel estimator circuit applies the algorithm that was analyzed earlier. It consists of an accumulator (adding the R_j), a PN sequence shift register and 15 taps. In each tap, the same y(n) is multiplied with a different PN_i, as dictated by the algorithm. The inputs and outputs of the algorithm are shown below:

Inputs

- **clk**: The circuit clock signal (5 nsec = 200MHz)
- **rst**: The complete system reset signal
- **pn_set**: Signal originating from the control circuit that defines when the PN sequence shift register shall be loaded
- **en**: The estimaton_en signal originating from the control circuit
- **shift**: The estimation_shift signal originating from the control circuit
- **signal_in**: The input signal

Outputs

- **a**: The calculated channel coefficients (type Array 15 signal, 10-bit coefficients)

The channel estimator circuit is illustrated as follows:

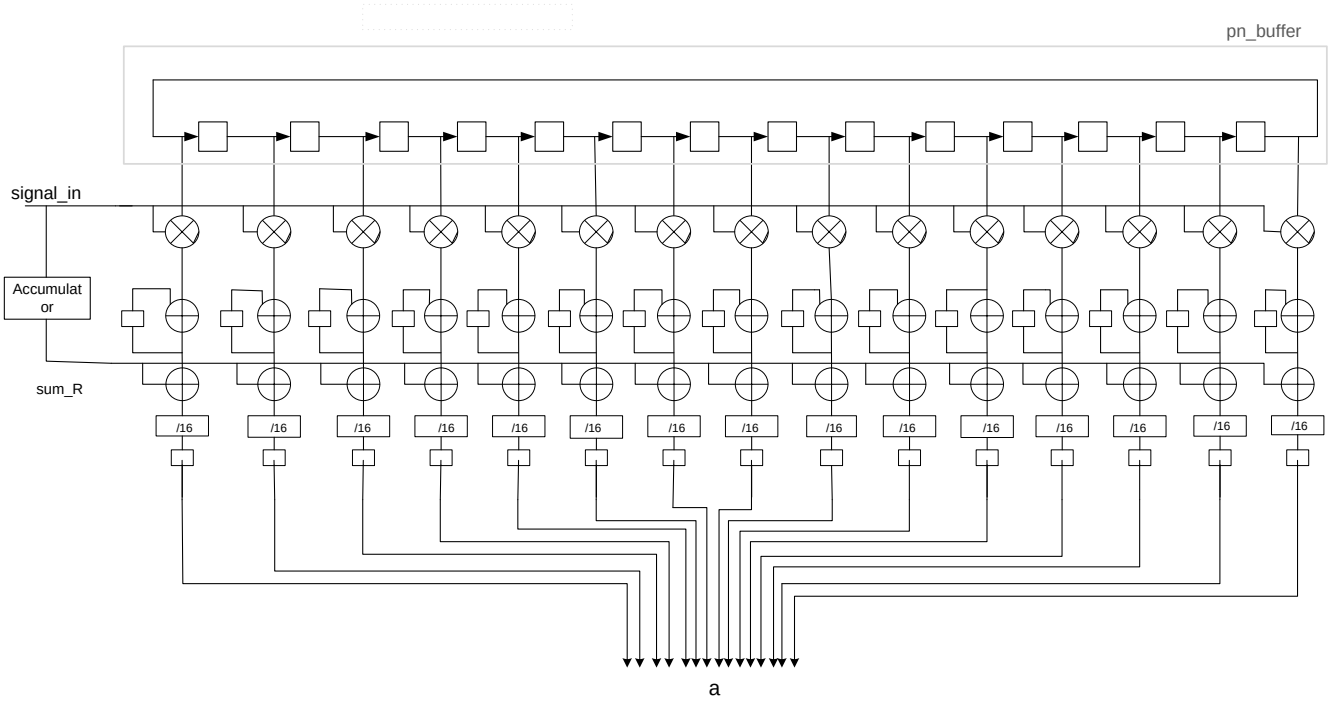


Figure 3.2 Architecture of the channel estimator

Below we will analyze the operation of each sub circuit of the channel estimator separately.

3.2.3.1. PN sequence Shift Register

The PN sequence shift register consists of a circular chain of 15 delay circuits (1 bit each). This register is loaded when pn_set=1, while when pn_set=0 it shifts by one at each clock pulse. This register supplies the taps with the suitable PN sequence bits, as was described in the channel estimation algorithm chapter. The inputs and outputs of this circuit are shown below:

Inputs

- **PN_CODE**: The PN sequence (15 bits) is initially loaded through this input.
- **clk**: The system clock.
- **rst**: The complete system reset signal.
- **pnset**: The control circuit pn_set signal.
- **en**: The complete system enable signal.

Outputs

- **PN**: The shifted PN sequence (15 cables of 1 bit each – every cable is directed to a different tap).

3.2.3.2. Channel estimator tap

As was also mentioned earlier, the channel estimator consists of 15 similar structures, the so-called “taps”. Every tap has the following inputs and outputs:

Inputs

- **in_num:** The input signal
- **sum_R:** The sum of the R_j terms originating from the accumulator
- **pn:** The PN sequence corresponding bit
- **shift:** The estimation_shift signal originating from the control circuit
- **rst:** The system reset signal
- **clk:** The system clock
- **en:** The system enable signal

Outputs

- **out_num:** The tap output

The channel estimator tap is illustrated in the figure below. The adder combined with the two flip flops constitute an accumulator. The top flip flop receives the system clock as its clock and the rake_sh signal as its reset. The bottom flip flop receives the rake_sh signal as its clock and the system reset signal as its reset. In this way, the top flip-flop provides the accumulator with the immediate previous accumulation value, while the bottom flip flop regulates when the accumulation result shall be sent (after it has been added to sum_R and divided by 16) to the channel estimator output.

The structure of each channel estimator tap is shown below:

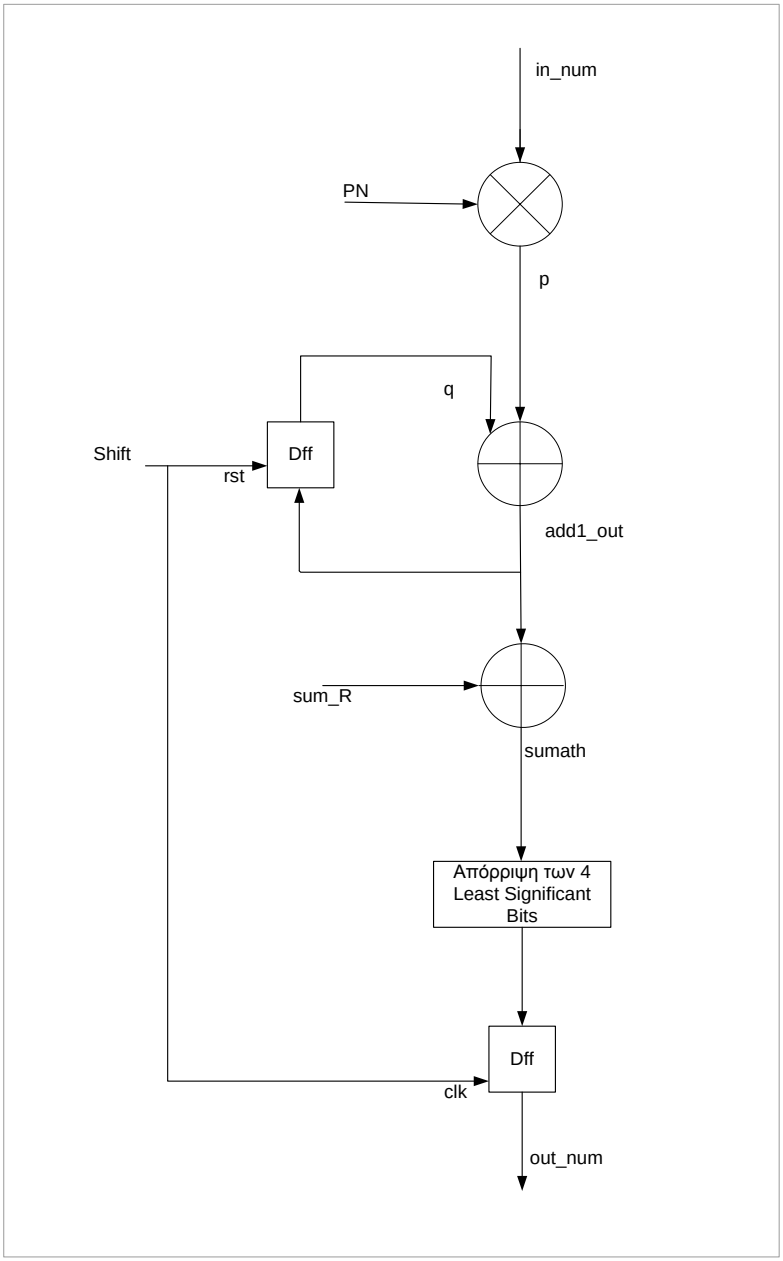


Figure 3.3 The channel estimator tap

The dismissal of the 4 Least Significant Bits is equivalent to a division by 16 (=N+1).

The multiplication circuit of the input signal with the PN sequence bit is shown below, as it is produced by the RTL schematic of the ISE software tool:

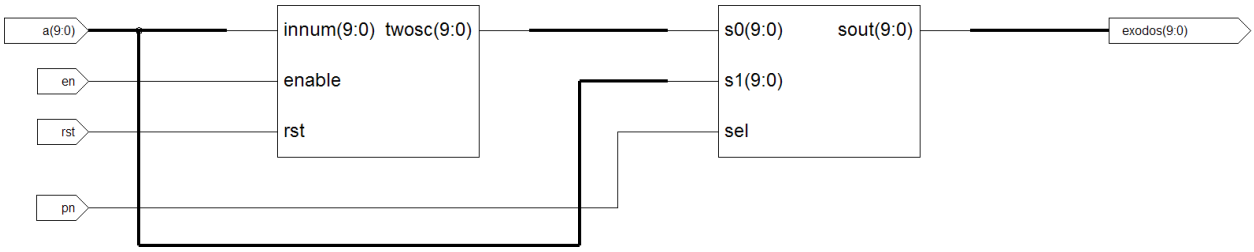


Figure 3.4 The PN multiplier

The circuit on the left produces the two's complement of the input signal. The circuit on the right is a multiplexer that selects as its output either the initial signal or its complement, depending on the PN bit value.

3.2.3.3. The Accumulator

This circuit adds the input signal values in pairs of 15. As was proven above, this sum is equal to the sum of the channel coefficients, which is required in order for the channel estimation algorithm to be executed. This circuit has the following inputs and outputs:

Inputs

- **signal_in:** The system input signal
- **shift:** The estimation_shift signal originating from the control circuit
- **clk:** The system clock
- **rst:** The system reset signal
- **en:** The estimation_en signal originating from the control circuit

Outputs

- **sum_R:** The sum of input signals in pairs of 15

3.2.4 RAKE Circuit

The RAKE circuit is similar in structure as that of the channel estimator. It also consists of 15 taps and one shift register. However, here the shift register contains the old values of the input signal, instead of the PN sequence.

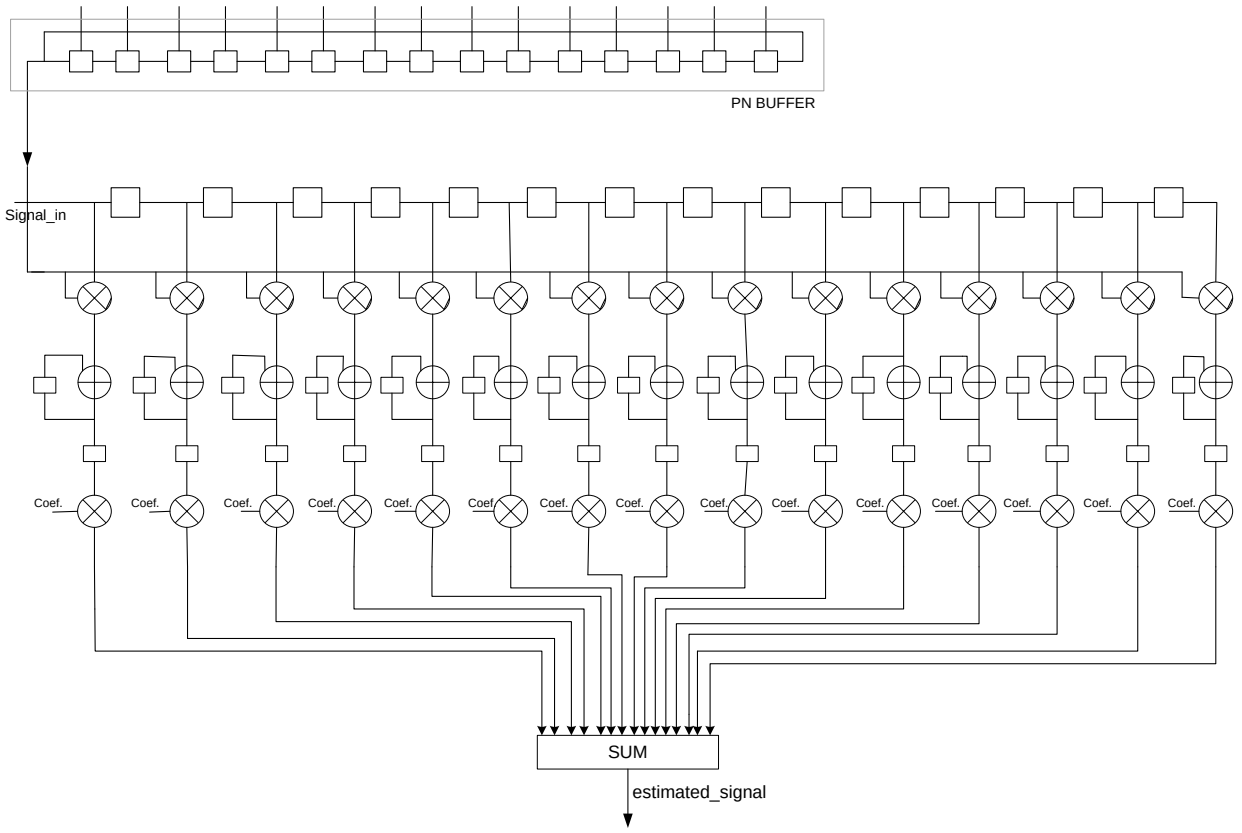


Figure 3.5 Architecture of the Rake circuit

Every tap uses a different channel coefficient originating from the channel estimator, which remains constant until the channel is estimated anew. The inputs and outputs of this circuit are described below:

Inputs

- **signal_in:** The system input signal
- **coef_arr:** The 15 coefficients originating from the channel estimator
- **pn_code:** The PN code
- **shift:** The rake_shift signal originating from the control circuit
- **clk:** The system clock
- **rst:** The system reset signal
- **pnset:** The pn_set signal originating from the control circuit
- **en:** The rake_en signal originating from the control circuit

Outputs

- **estimated_signal:** The circuit output

3.2.4.1. PN sequence Register

A main difference of this circuit, compared to the channel estimator, is that here all the taps are using the same PN sequence bit. The PN sequence register is described more extensively below:

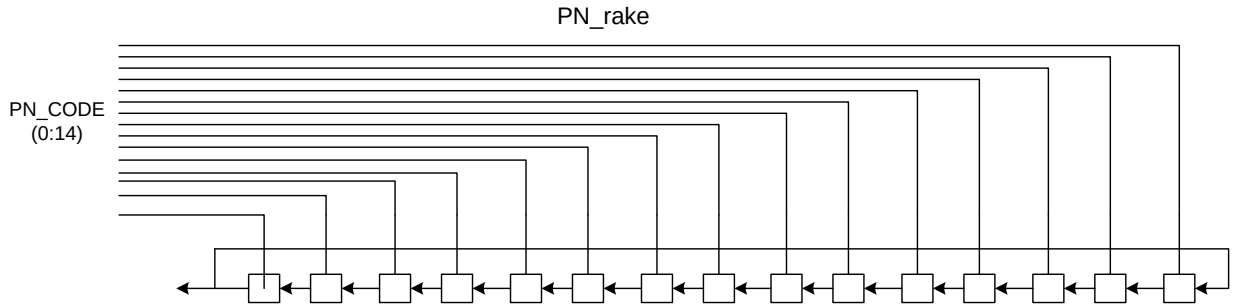


Figure 3.6 Architecture of the PN sequence register

The sequence is loaded onto the register when pn_set=1. The register is shifted by one on the left at every positive clock edge.

The inputs and outputs of this circuit are described below:

Inputs

- **PN_CODE:** The PN code
- **clk:** The system clock
- **rst:** The system reset signal
- **pnset:** The pn_set signal originating from the control circuit
- **en:** The rake_en signal originating from the control circuit

Outputs

- **PN_t:** The PN sequence bit which will be used by all the taps

3.2.4.2. Signal Register

The circuit of the signal register is shown below:

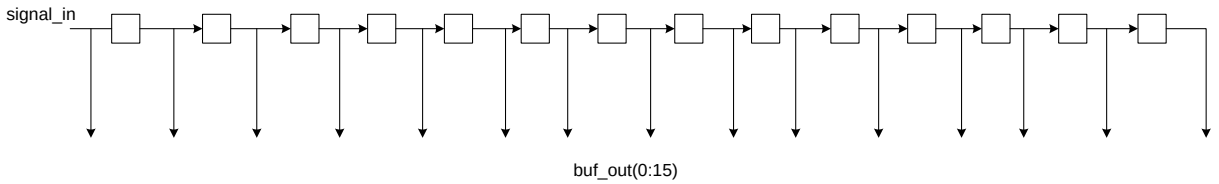


Figure 3.7 The signal buffer

This register consists of 14 delay circuits. The signal that will be sent to the first tap of the rake circuit does not get delayed at all, and this is why we use 14 delay circuits instead of 15. A right shifting takes place at every positive clock edge. The above circuit has the following inputs and outputs:

Inputs

- **signal_in:** The system input signal
- **clk:** The system clock
- **rst:** The system reset signal
- **en:** The rake_en signal originating from the control circuit

Outputs

- **buf_out:** The 15 y(n-k) values , κ=0,1,…14 , which are supplied to the taps of the rake circuit.

3.2.4.3. The RAKE Calculations Circuit

This circuit, which is called tap for short, is the main component of the complete system, since most calculations are performed here and the most area-demanding sub circuits also appear here. The structure of this circuit is illustrated below:

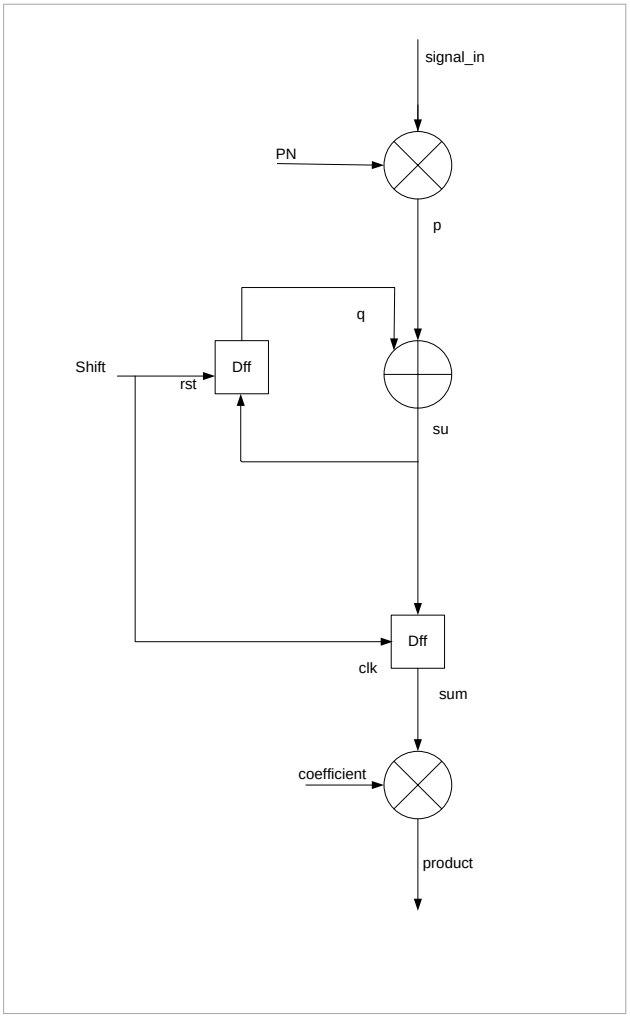


Figure 3.8 The Rake circuit Tap

The same PN multiplier used in the channel estimator also performs the PN multiplication. The adder combined with the two flip flops constitutes an accumulator. The top flip flop receives the system clock as its clock and the rake_sh signal as its reset signal. The bottom flip flop receives the rake_sh system signal as its clock and the system reset signal as its reset signal. Therefore, the top flip-flop provides the

accumulator with the immediate previous accumulation value, while the bottom flip flop regulates when the accumulation result shall be sent to the multiplier.

The inputs and outputs of the circuit are as follows:

Inputs

- **signal_in:** The system input signal
- **coefficient:** The 15 coefficients originating from the channel estimator
- **pn_code:** The PN code
- **shift:** The rake_shift signal originating from the control circuit
- **clk:** The system clock
- **rst:** The system reset signal
- **pnset:** The pn_set signal originating from the control circuit
- **en:** The rake_en signal originating from the control circuit

Outputs

- **estimated_signal:** The circuit output

3.2.4.4. Adder (adder_15)

Below we can observe the RTL schematic of the final adder of the RAKE circuit:

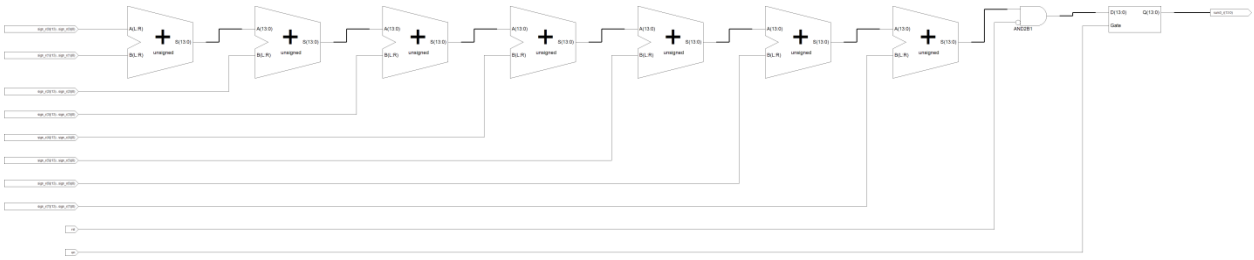


Figure 3.9 The RTL schematic of the adder circuit

Evidently, this adder is not optimal because of its very large critical path. However, we must not forget that the diagram illustrated above is derived from the RTL schematic. Upon selecting the technology schematic option in the ISE program, which represents

the real circuit to be constructed in the FPGA, we obtain the following diagram for the above adder:

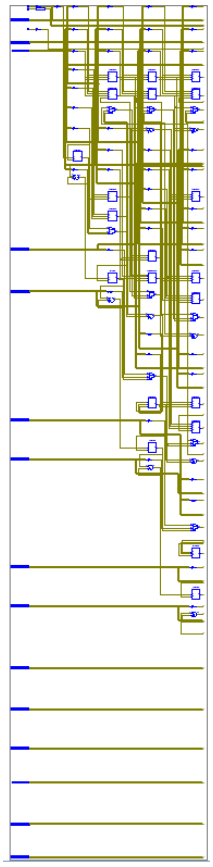


Figure 3.10 Technology schematic of the adder circuit

Inputs

- **rst:** The system reset signal
- **en:** The rake_en signal originating from the control circuit
- **taps_outputs:** The tap outputs

Outputs

- **sum_15taps:** The final sum

3.3. The second circuit: Selective Rake Receiver

3.3.1. Introduction

The design of the selective rake circuit was based on the design of the above simple RAKE receiver. We chose to preserve the 8 best out of 15 system coefficients. In general, the following was done:

- The channel estimator remained unchanged. However, a circuit which selects the 8 best out of 15 coefficients was added. The code for this circuit was written in behavioral form. Indeed, this behavioral code corresponds to the VHDL version of the bubble sort algorithm, with one significant modification, which will be analyzed below. The ISE tool translates this algorithm into hardware terms in the most optimal way.

- The control circuit remained unchanged.

- The rake circuit now has only 8 taps. In addition, the signal buffer of the rake circuit selects the y(n-k) that correspond to the best channel coefficients α_k and sends only these y(n-k) to the taps.

3.3.2. Architecture of the selective rake receiver

The general structure of the complete selective rake system that was designed is illustrated in the page below. In that figure, we observe on the top left the circuit that selects the 8 best out of 15 coefficients. The outputs of this circuit are the positions of the 8 best coefficients. These positions are 4-bit numbers ranging from 0 to 15. On the very bottom of that figure we observe the control circuit, which supplies the rest of the circuits with the control signals. On the right we observe the RAKE circuit, which receives the coefficients from the channel estimator (middle circuit) along with the positions of the best coefficients from the top left circuit.
The above complete circuit has the same inputs and outputs as the simple RAKE receiver.

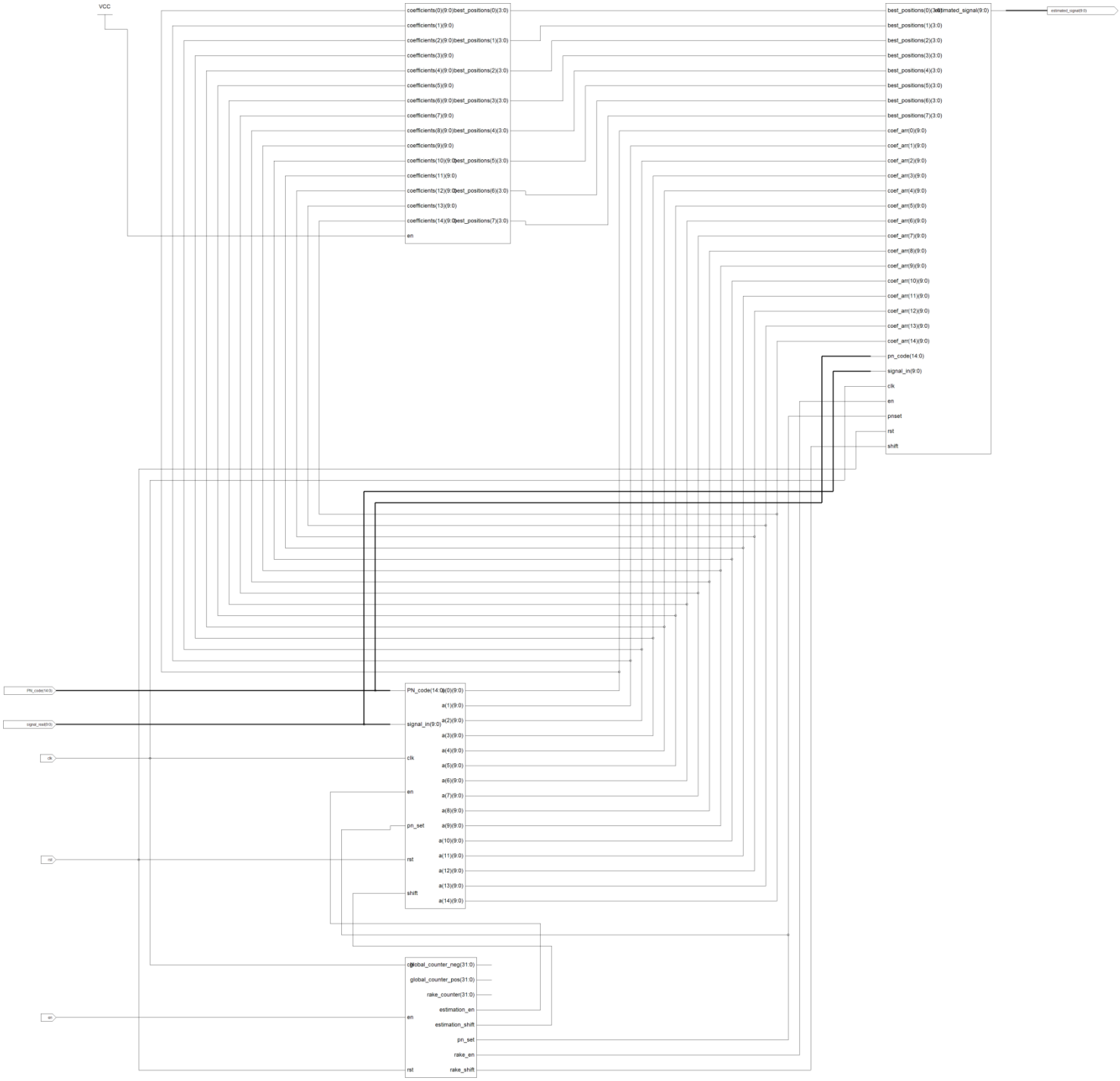


Figure 3.11 Architecture of the selective rake receiver

3.3.2.2. Components selection Circuit (selective component)

This circuit has the following inputs and outputs:

Inputs

- **coefficients:** The 15 coefficients originating from the channel estimator
- **en:** Enable signal originating from the control circuit

Outputs

- **best_positions:** The positions of the 8 best coefficients

The VHDL code of this circuit is shown below:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.STD_LOGIC_SIGNED.ALL;
use work.rake_pack.all;


entity selective_component is
   Port ( coefficients : in  arr;
                     en: in std_logic;
         best_positions : out  int_arr8);
end selective_component;

architecture Behavioral of selective_component is


 begin

 sorting : process(en,coefficients)

  variable i,j,k : integer;
              variable data_array : arr;
              variable tem_data : STD_LOGIC_VECTOR (9 downto 0);
  type int_array_15 is array (14 downto 0) of integer;
              variable pos_array : int_array_15;
```

```
            variable best_positions_var : int_arr8;
            variable tem_pos,pointer: integer;


begin
             data_array:=coefficients;

             pos_array(0):=0;
             pos_array(1):=1;
             pos_array(2):=2;
             pos_array(3):=3;
             pos_array(4):=4;
             pos_array(5):=5;
             pos_array(6):=6;
             pos_array(7):=7;
             pos_array(8):=8;
             pos_array(9):=9;
             pos_array(10):=10;
             pos_array(11):=11;
             pos_array(12):=12;
             pos_array(13):=13;
             pos_array(14):=14;



   if (en='1') then

     for i in 14 downto 7  loop     --INSTEAD OF 14 DOWNTO 0
     for j in 1 to i loop
        k := j-1;
        if ( abs(conv_integer(signed(data_array(k)))) > abs(conv_integer(signed(data_array(j))))  ) then
          tem_data := data_array(k);
          data_array(k) := data_array(j);
          data_array(j) := tem_data;

          tem_pos := pos_array(k);
          pos_array(k) := pos_array(j);
          pos_array(j) := tem_pos;

        end if;
      end loop;
    end loop;


            for m in 7 to 14 loop
               best_positions_var(m-7) := pos_array(m);
            end loop;


   best_positions<= best_positions_var;


   end if;
 end process;
end Behavioral;
```

In the line in bold lies the correction of the bubble sort algorithm which we referred to earlier. Since we do not need the complete array of the numbers, but just the 8 greater out of 15, only 8 algorithm loops are required. This modification saves a fair amount of hardware resources and makes the circuit faster. The schematic of the circuit is illustrated below, as was obtained by the ISE.

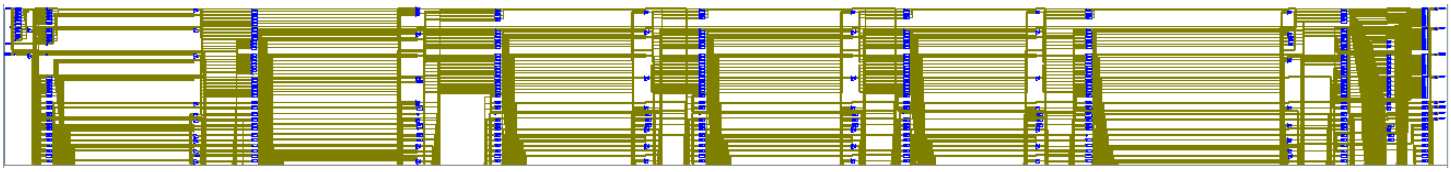


Figure 3.12 The RTL schematic of the circuit that selects the best coefficients

It is observed that ISE translates the above code into a network of comparators and multiplexers. The sorting of the numbers is obviously not done in series, as in the code, but in parallel. Besides, the above circuit does not use a clock, as it is a composite one.

3.3.2.3. The RAKE Circuit

Some significant modifications have been made to the Rake circuit. To begin with, we have only 8 taps instead of 15, as also mentioned earlier. The RTL schematic of the RAKE circuit can be seen below:

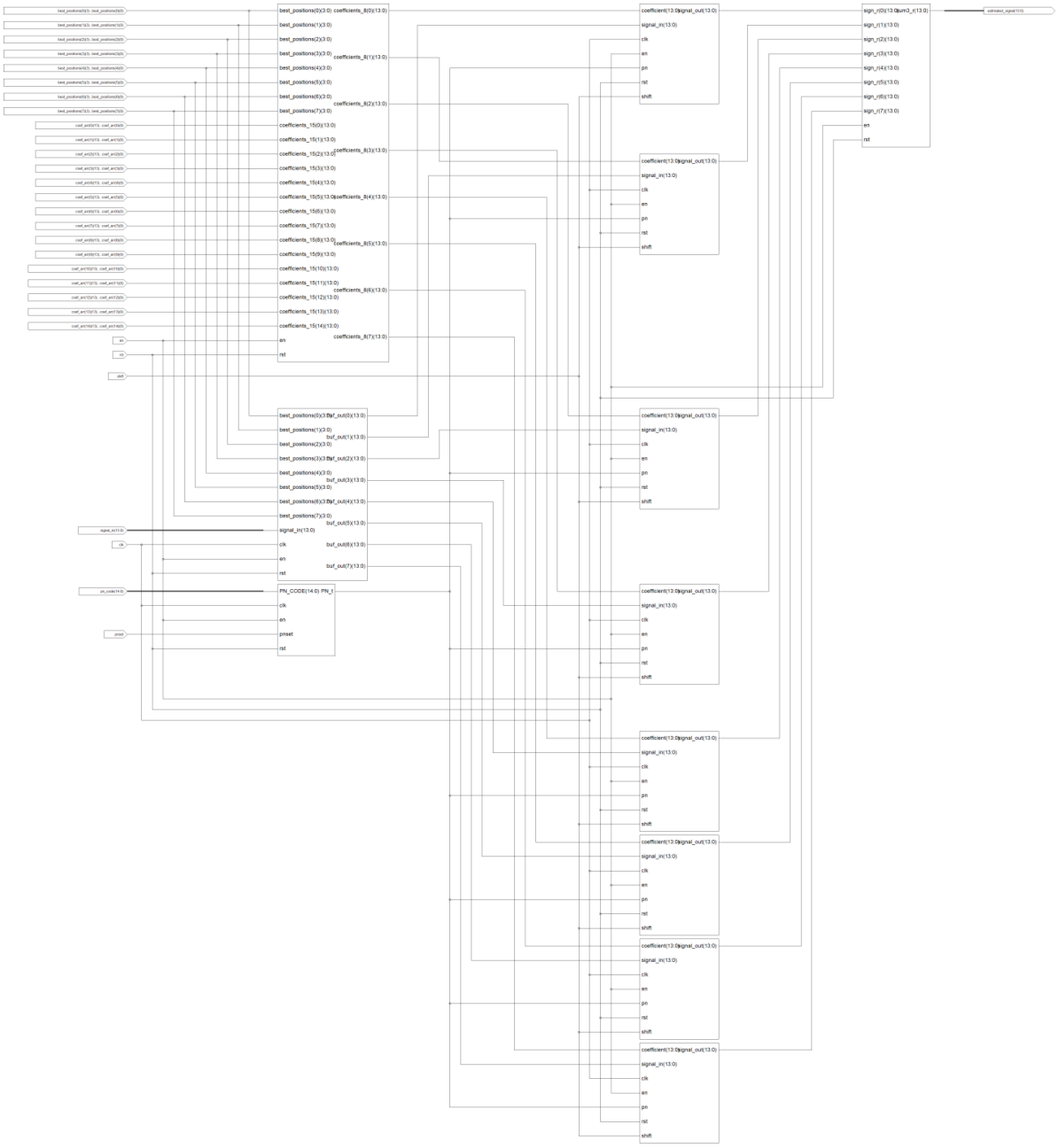


Figure 3.13 Architecture of the Rake circuit

What is seen on the above schematic is explained in the diagram below:

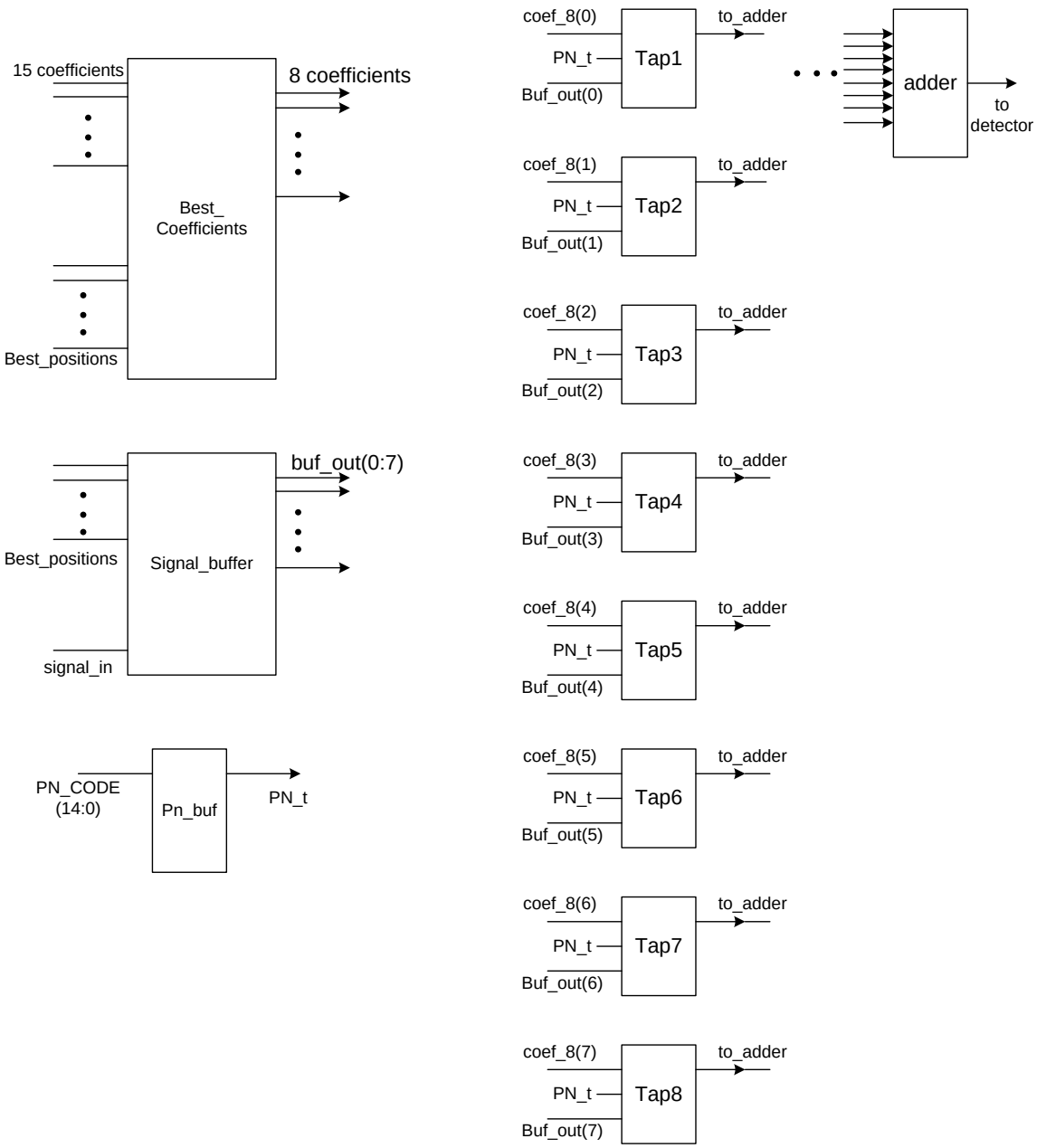


Figure 3.14 Architecture of the Rake circuit

Looking at the above sub circuits, the taps are internally the same as those of the simple rake. The same holds true for the Pn_buf and the final adder (although here it adds 8 numbers). The two circuits which we will additionally study are the best_coefficients and the Signal_buffer circuits.

The inputs and outputs of the above circuit are shown below:

Inputs

- **signal_in:** The system input signal
- **coef_arr:** The 15 coefficients of the channel impulse response
- **best_positions:** The positions of the 8 best channel components
- **pn_code:** The PN system code
- **shift:** The rake_shift signal originating from the control circuit
- **clk:** The system clock
- **rst:** The system reset signal
- **pnset:** The pn_set signal originating from the control circuit
- **en:** The rake_en signal originating from the control circuit

Outputs

- **estimated_signal:** The circuit output

3.3.2.3.1. The best_coefficients circuit

The RLT schematic of this circuit is given below. We observe that 8 15-to-1 multiplexers are used, which decide which signal will be driven to their output, depending on the best_positions(i) signal. Therefore, we eventually have the 8 best coefficients at the circuit output. Here it must be noted that the coefficients will not appear in the same order as in the channel impulse response, but in descending order instead. This however does not pose a problem, because the corresponding y(n-k) appear in the same way out of the signal_buffer, as we will see below. Thus, every α_k coefficient enters the same tap as the corresponding y(n-k).

The architecture of this circuit is illustrated below:

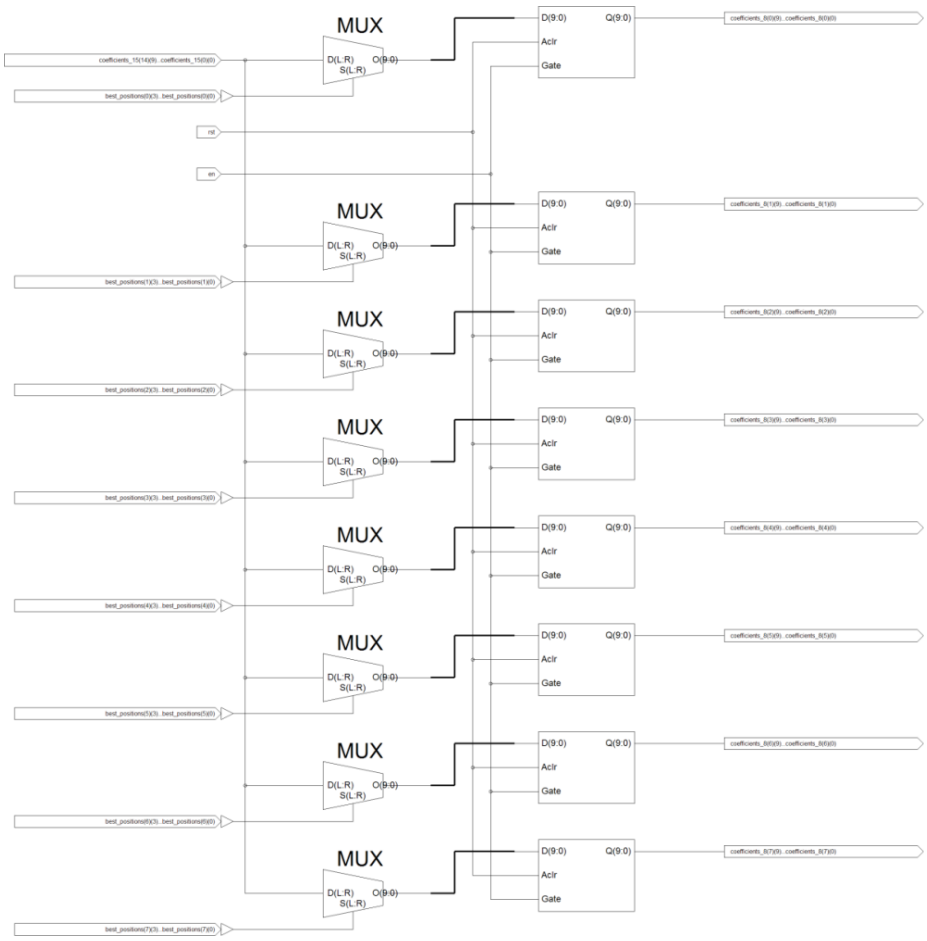


Figure 3.15 The best coefficients circuit

The inputs and outputs of the circuit are as follows:

Inputs

- **coefficients_15:** The 15 coefficients of the channel impulse response
- **best_positions:** The positions of the 8 best channel components
- **rst:** The system reset signal
- **en:** The rake_en signal originating from the control circuit

Outputs

- **coefficients_8:** The 8 best coefficients of the channel impulse response

3.2.3.2. The Signal_buffer circuit

The RLT schematic of this circuit is given below. We observe that 8 15-to-1 multiplexers are also used here, which decide which y(n-k) will be driven to their output, depending on the best_positions(i) signal. On the left, the moving chain of the 15 delay circuits is apparent.

The inputs and outputs of this circuit are shown below:

Inputs

- **signal_in:** The input signal
- **best_positions:** The positions of the 8 best channel components
- **clk:** The system clock
- **rst:** The system reset signal
- **en:** The rake_en signal originating from the control circuit

Outputs

- **buf_out:** The 8 delayed input signals corresponding to the 8 best channel components

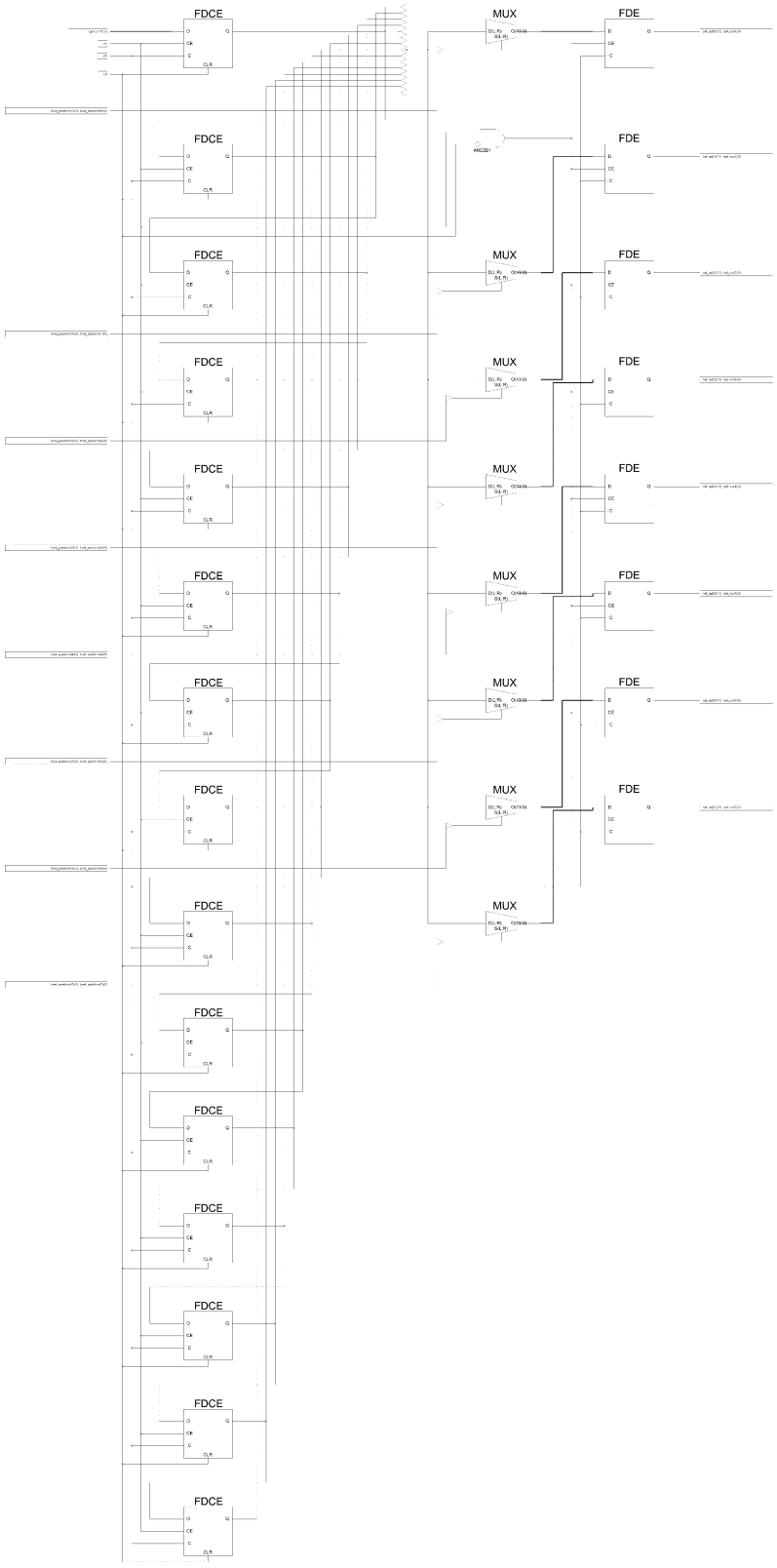


Figure 3.16 The signal buffer circuit

3.4. The third circuit - Modified simple RAKE receiver

3.4.1. Introduction

We designed a new version of the simple RAKE receiver, by applying the pipeline technique in order to increase the clock frequency. An additional innovation in this circuit is that the use of negative clock edges in the system was eliminated. In order to accomplish that, the taps of the channel estimator and of the RAKE circuit were completely redesigned. The necessary modifications were made to the clock circuit as well. Since the rest of the circuit remained unchanged, below we will only provide the new tap architecture along with the logic of the clock circuit.

3.4.2. The tap of the RAKE circuit

The new architecture of the tap of the RAKE circuit is observed below. The combination of the adder with the two flip flops was replaced by a set of two accumulators.

The reason why two accumulators are required, instead of one, is that each accumulator requires a clock cycle in order to send the result of the accumulation to its output and then clear its contents. During this cycle, the input data present at its input is lost – it does not get summed with the following data. However, we do not have this luxury in our application – all of the data must be summed in pairs of 15. So we introduced two accumulators, which we clocked appropriately. The first sums the first 15 pieces of input data and sends the result of the accumulation to its output on the 16^{th} clock cycle. The second begins its accumulation from the 16^{th} clock cycle until the 30^{th} and on the 31^{th} cycle it sends its result to its output. In this way, the data of the 16^{th} clock cycle is not lost. Similarly, the first accumulator begins yet again the accumulation on the 31^{th} clock cycle, which means that the piece of data in question is not lost. A multiplexer regulates which output of the two (accum1,accum2) will be driven to the multiplier in order for it to be multiplied with the corresponding channel coefficient.

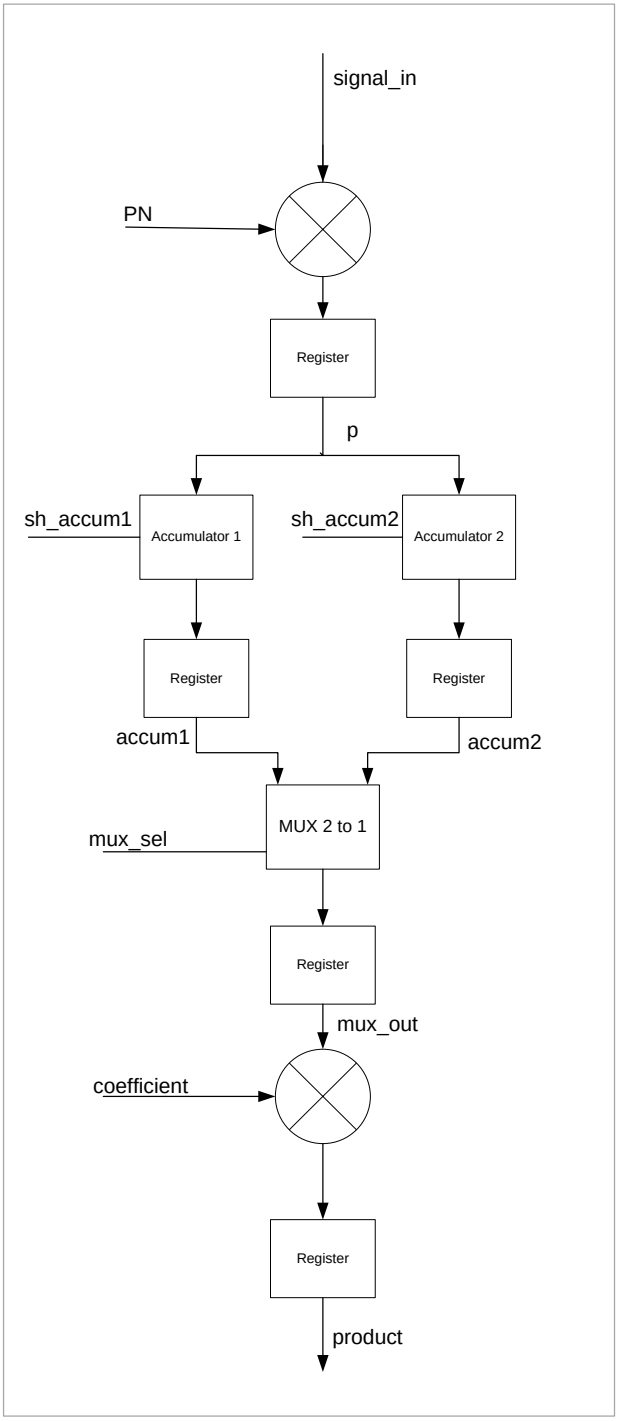


Figure 3.17 The tap of the Rake circuit

The tuning of the two accumulators is shown graphically in the table below:

Στήλη																																												
t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
a/b	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
(mux)																																														
shA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
shB	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

It is also observed in the circuit above that registers have been placed among all the tap components of the rake circuit. In this way, the simultaneous operation of all the components is achieved, that is to say we are using the pipelining technique. Therefore, the frequency of this circuit is determined by the delay of the slowest component, instead of that of the combination of all those components.

The inputs and outputs of the above circuit are as follows:

Inputs

- **signal_in:** The system input signal
- **PN:** The PN code corresponding bit
- **mux_sel:** The signal that determines which of the two outputs of the accumulators will be driven to the multiplier
- **sh_acum1, (sh_acum2):** The signal that determines when the first (second) accumulator will clear its contents, after it has sent the accumulation result to its output.
- **clk:** The system clock
- **rst:** The system reset signal
- **en:** The rake_en signal originating from the control circuit
- **coefficient:** The corresponding coefficient originating from the channel estimator

Outputs

- **estimated_signal:** The circuit output

3.4.3. The tap of the channel estimator

The tap of the channel estimator is similar to that of the RAKE circuit. The only difference here is that, instead of multiplication, summation with the signal sum_R is performed, as dictated by the channel estimation algorithm. Additionally, we have a division by 16 at the end, which is realized by dumping the 4 least significant number bits. As is evident from the diagram, the pipeline technique is also used here. The inputs and outputs of the circuit are as follows:

Inputs

- **signal_in:** The system input signal
- **PN:** The PN code corresponding bit
- **mux_sel:** The signal that determines which of the two outputs of the accumulators will be driven to the multiplier
- **sh_acum1, (sh_acum2):** The signal that determines when the first (second) accumulator will clear its contents, after it has first sent the accumulation result to its output.
- **clk:** The system clock
- **rst:** The system reset signal
- **en:** The rake_en signal originating from the control circuit
- **sum_R:** The sum of the R_j coefficients originating from the accumulator of the channel estimator.

Outputs

- **out_num:** The channel coefficient that was calculated.

The architecture of the tap of the channel estimator is shown below:

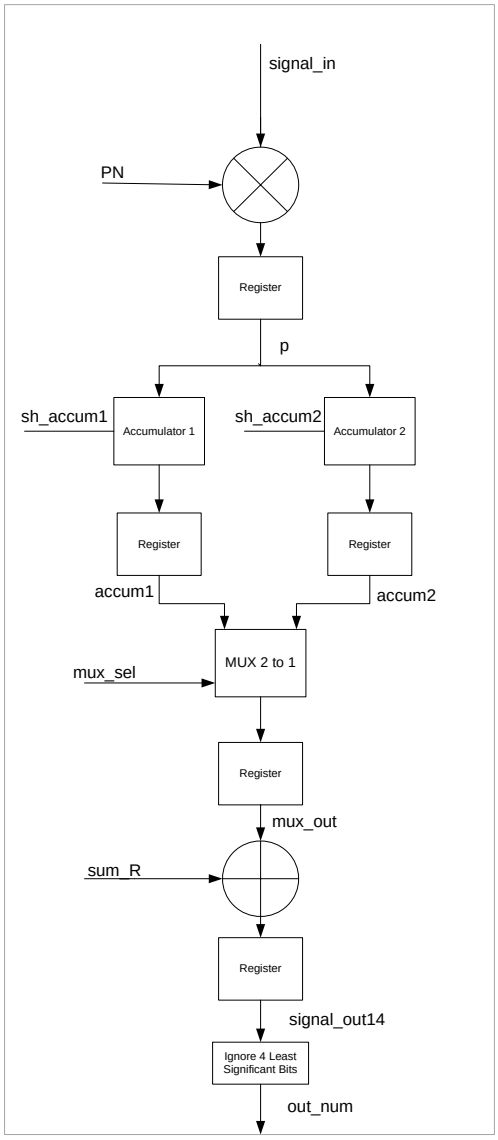


Figure 3.18 The tap of the channel estimator

Chapter 4 : Simulations

4.1. Introduction

Once we had reached a conclusion with regard to the general architecture of the three circuits that we implement, we then moved on to their design in VHDL code, which was done using the software Xilinx ISE 9.2i. All the relevant pieces of code are provided in the appendix, at the end of the diploma thesis.

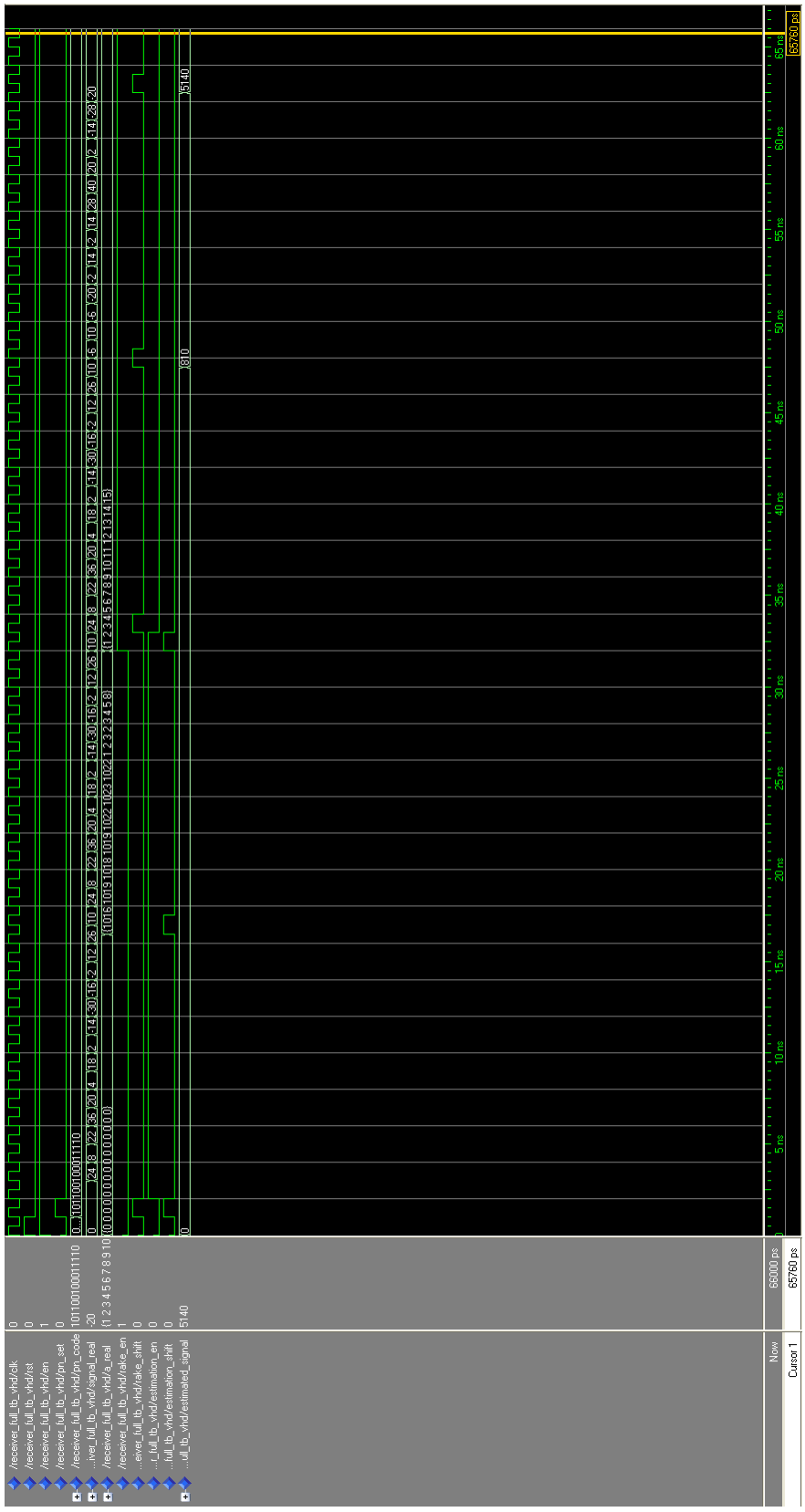
Once we had ended up with composable circuits that implement the architecture described above, we moved on to the simulation of the above circuits using the software Modelsim 6.1f, through the use of some appropriate test vectors. These test vectors correspond to the digital signal that reaches the input of the rake receiver, once it has passed through the RF part of the receiver. In order to calculate them, we developed a model in mathcad, which simulates the distortion of the data sent by the transmitter through a known channel. In this Mathcad model we also simulated the operation of the rake circuit and realized that the circuit response given by Mathcad is identical to the one given by Modelsim. Thus, we verified that the design of the circuit in VHDL was correct.

The simulations given by Modelsim for the three circuits that we designed, using the same test vector (and thus the same channel for the three circuits), are shown below.

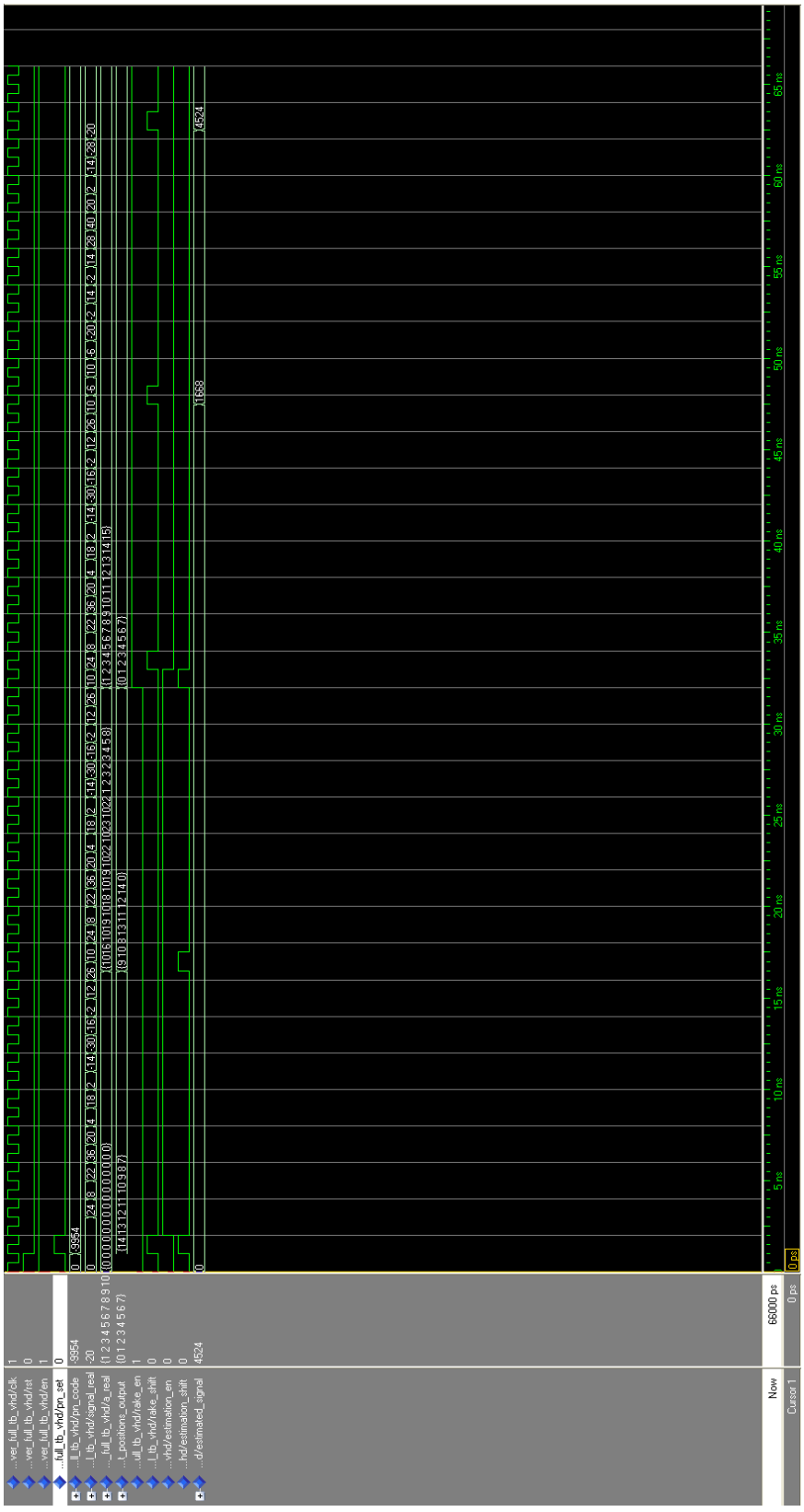
4.2. Logic simulations

The logic simulations of the three circuits that we designed are shown in the figures below, as provided by Simulink 6.1f:

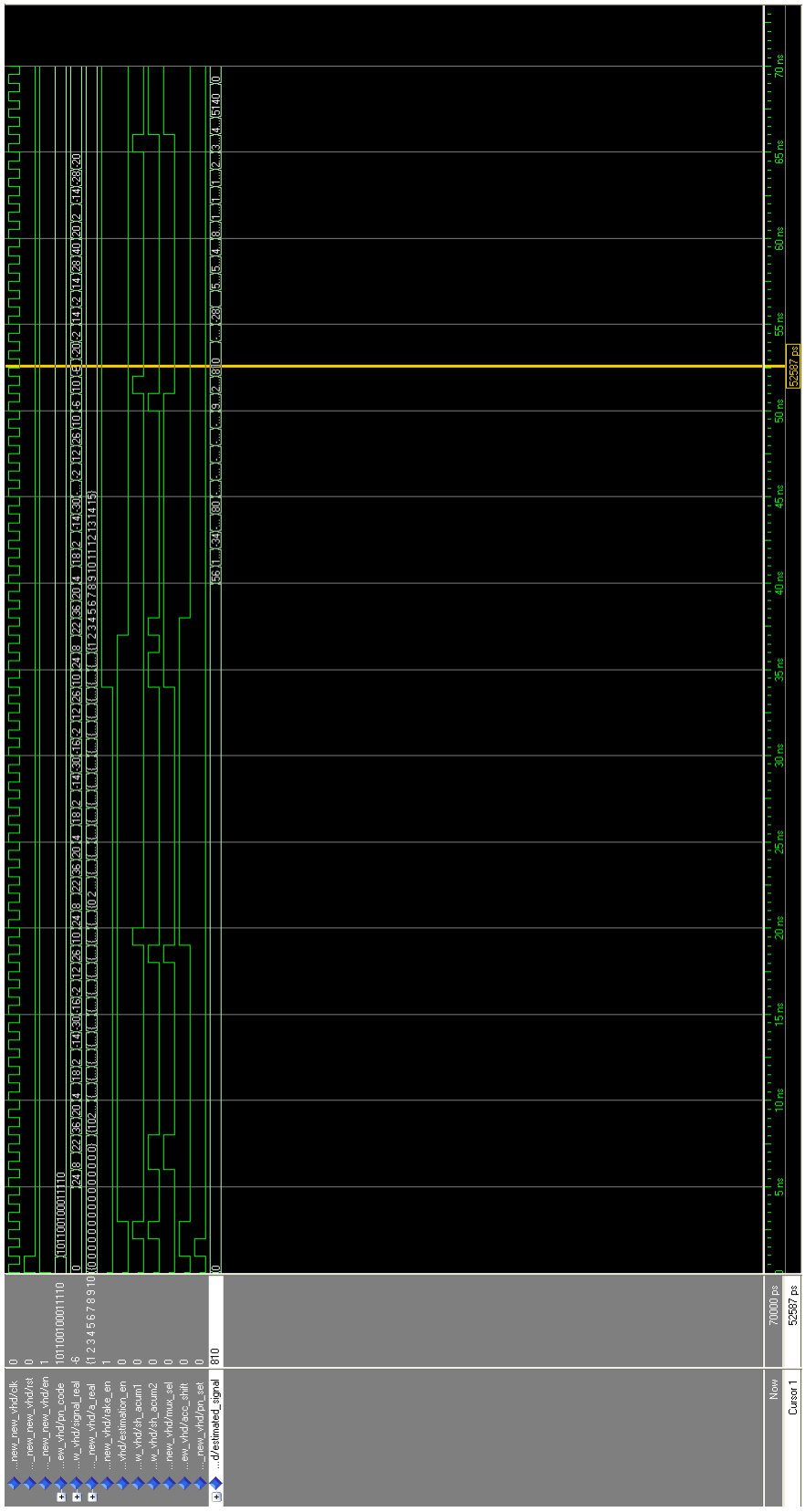
Logic simulation of the first circuit



Logic simulation of the second circuit



Logic simulation of the third circuit



4.3. Analysis of the above logic simulations

The test vector sent in all of the three simulations above corresponds to the transmission of three PN sequences and of one -1 symbol, by the transmitter. This means that the transmitter sends the PN sequence three times, then immediately sends the PN sequence once again, this time inversed. This signal passes through the following channel:

channel= [15 14 13 12 11 10 9 8 7 6 5 4 3 2 1]

The derived signal is the one seen in the field signal_in of the above waveforms.

Logic simulation of the first circuit

It is initially observed that the channel estimator has become enabled. The first PN sequence fills its taps, while the useful calculations are performed during the second one. After the passing of the two first PN sequences, the channel estimator correctly provides the 15 corresponding channel coefficients at the output of the a_real and then the RAKE circuit becomes enabled. The third PN sequence sent by the transmitter fills the rake taps, and the calculations at the RAKE circuit are performed during the next 15 clock cycles, which correspond to the -1 symbol. It is observed that a signal with a value of 5140 is obtained at the estimated_signal output of the complete system, which is the predicted basis of the simulations. Consequently, our circuit is operating correctly.

Logic simulation of the second circuit

Here, the RAKE is selective and the 8 best out of 15 components are chosen. Once the channel estimation is complete, as in the previous circuit, the selection circuit of the best coefficients provides the positions of the best channel coefficients at its best_positions_output output. It is observed that, here, the positions correspond to the 8 first channel components. This is to be expected if we are reminded of the channel used in the simulations. Then, the RAKE circuit performs the calculations, although it uses only the input signals corresponding to the 8 best channel components. Thus, we now obtain the value of 4524 at the estimated_signal output. This value is sufficiently

close to the value of 5140 that was calculated before, which means that we collected a large percentage of the energy of the multipath components.

Logic simulation of the third circuit

The new operation concept of the circuit is observed here. The clock signals differ from the previous ones, since we have two accumulators. In addition, there is no operation at negative clock edges. The final result at the estimated_signal signal is the same as that of the first circuit (5140).

4.4. Circuit Comparison with regard to the resource usage of the FPGA

The reports given by the synthesis tool of the software package Xilinx ISE 9.21i with regard to the resources of the fpga that each of the three circuits is using are shown below:

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	2134	15360	13%
Number of Slice Flip Flops	3171	30720	10%
Number of 4 input LUTs	2551	30720	8%
Number of bonded IOBs	261	448	58%
Number of GCLKs	5	32	15%
Number of DSP48s	15	192	7%

Figure 4.1 Resource usage of the first circuit

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	3766	15360	24%
Number of Slice Flip Flops	2481	30720	8%
Number of 4 input LUTs	6311	30720	20%
Number of bonded IOBs	293	448	65%
Number of GCLKs	5	32	15%
Number of DSP48s	8	192	4%

Figure 4.2 Resource usage of the second circuit

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	2591	15360	16%
Number of Slice Flip Flops	3156	30720	10%
Number of 4 input LUTs	4664	30720	15%
Number of bonded IOBs	263	448	58%
Number of GCLKs	1	32	3%
Number of DSP48s	15	192	7%

Figure 4.3 Resource usage of the third circuit

It is observed that the second circuit (selective RAKE), uses significantly more resources than the simple RAKE receiver. Therefore, we conclude that it is not profitable to implement the selective version of the RAKE receiver, at least for a ratio of selected/initial taps equal to 8/15. This is due to the large resource usage of the circuit that selects the best coefficients, and to the additional multiplexers required in the rake circuit.

In addition, it is observed that the third circuit is slightly larger compared to the first circuit. This was to be expected, as we are using two accumulators at each tap, instead of one.

4.5. Circuit Comparison with regard to the operating frequency

The operating frequencies given by the XST synthesis tool of the software package Xilinx ISE 9.2i are:

- Circuit 1: 213MHz
- Circuit 2: 178MHz
- Circuit 3: 159MHz

A surprising fact here is that the third circuit, despite being optimized using the pipeline technique, has a slower clock compared to the first circuit. This is probably due to the fact that the design of the taps of the last circuit was made in behavioral code. In order to achieve the optimal results, the design of the taps of the third circuit must be made using structural code, as was done in the first circuit. Therefore, although pipelining is applied in the third circuit, the first circuit is faster.

BIBLIOGRAPHY

[1] Benedetto, Giancola – “Understanding Ultra Wide Band Radio Fundamentals“

[2] Wiley – “Ultra Wideband Wireless Communications and Networks”

[3] Wiley – “Ultra Wideband Wireless Communications (Arslan, Benedetto)”

[4] Wiley – “Ultra-Wideband Radio Technology”

[5] Wiley – “UWB Theory and Applications”

[6] Prentice Hall – “Ultra Wideband Communications Fundamentals and Applications”

[7] Prentice Hall – “An Introduction to Ultra Wideband Communication Systems”

[8] Proakis – “Communication Systems Engineering”

[9] Douglas L. Perry – “VHDL : Programming by Example”

[10] Xilinx ISE 9.2i Tutorial

[11] Modelsim 6.1f Tutorial