ISSN 2039-7941
Anno 2013_Numero 257

rapporti tecnici INGV

Double pulse envelope radar

257

Istituto Nazionale di
Geofisica e Vulcanologia

ISSN 2039-7941
Anno 2013_Numero 257

rapporti tecnici INGV

DOUBLE PULSE ENVELOPE RADAR

James Arokiasamy Baskaradas, Achille Zirizzotti, Giuseppe Tutone

INGV (Istituto Nazionale di Geofisica e Vulcanologia, Sezione Geomagnetismo, Aeronomia e Geofisica Ambientale)

257

Table of contents

Introduction

Radio Echo Sounding (RES) technique is used in glaciology, to perform the study of the properties of glaciers and ice [Bogorodsky, 1985]. It is based on the use of radar techniques, mainly to obtain information about the thickness of glaciers and polar ice caps. The RES instrumentation also allows the study of interglacial stratification and identification of areas of inhomogeneity of the ice, the exploration of subglacial lakes and physical characterization of the ice - bedrock interface [Cafarella, 2006; Frezzotti, 2000; Forieri, 2003; Mancini, 2003; Tabacco, 2000; Tabacco, 2003; Zirizzotti, 2010].

Simplifying RES instrumentation, it is a radar system mainly constituted by two main components: a radio frequency transmitter and a receiver connected to an antenna [Plewes, 2001]. A short electromagnetic pulse is transmitted through an antenna and the receiver detects the transmitted pulse and all the pulses reflected from the ice. The reflecting surfaces are surfaces of discontinuity in the electromagnetic characteristics of the medium. In the case of the ice we have: air-ice surface, the layers of ice or "internal layers" due to annual ice stratifications, the rocky "bedrock" (ice-rock interfaces) and subglacial lakes (liquid water present at the bedrock interface) all those are thus seen as electromagnetic discontinuity (figure 1).

From measurements of the delay times of the received pulses, knowing the speed of electromagnetic waves propagation in the ice, it is possible to determine the distance to these surfaces. The radio waves have different speeds depending on the media in which they propagate: in the air it is the light speed (300 m / µs), while in the ice it is about 168 m /µs [Glen, 1975].

Moreover, from measurements of the intensity (amplitude) of the detected echoes, it is possible to obtain information on the electromagnetic characteristics of the reflecting interface, allowing us to determine, for example, the physical state of the ice (wet or dry) and water presence at the bottom.

Subglacial lakes however are easily detected by the characteristic flat shape of the reflector surface and by the strong amplitude of the echo [Zirizzotti, 2011].

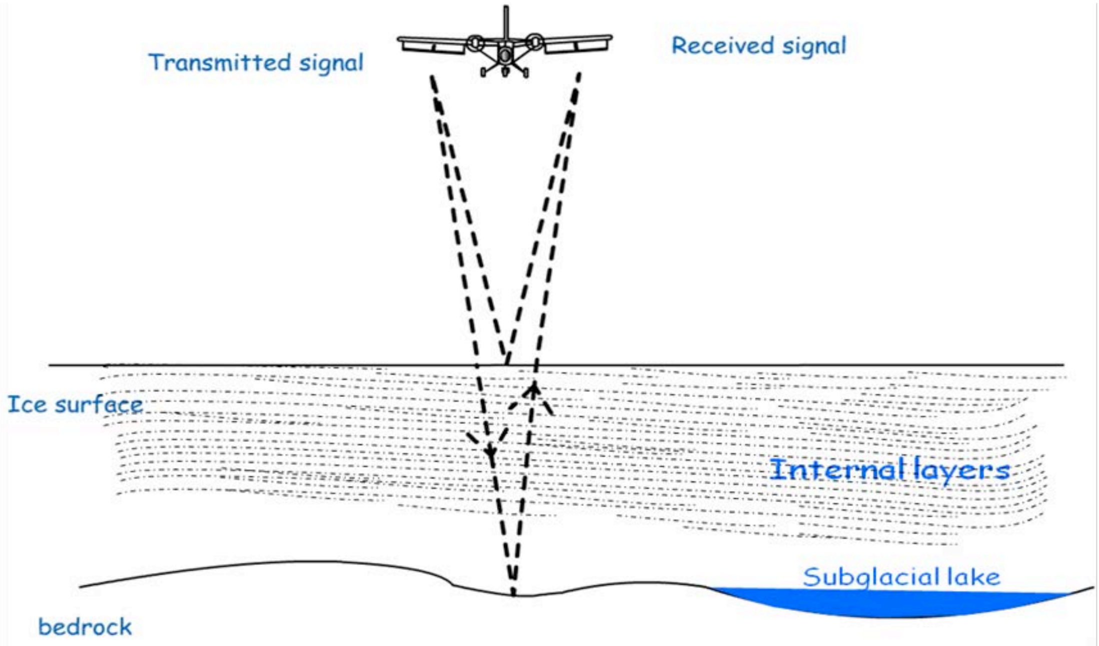

Figure 1. Radio Echo Sounding investigation.

RES measurements on the ice can be done using snow mobile vehicles, or by airplane. By mounting the radar system in a snowmobile vehicle and the antenna on a sledge it is possible to make local ice measurements on glaciers. While to analyze the vast areas of the ice caps in Greenland or in Antarctica a small twin engine aircraft DHC-6 Twin Otter is used. In this case the instrumentation is mounted inside the fuselage and the transmitting and receiving antennas are mounted below the wings of the plane.

A sketch of the radio paths between the plane and the glacier is illustrated in Figure 1. The transmitting antennas emit an electromagnetic pulse that is directly revealed by the receiving antennas (direct coupling between transmitter and receiver).

Continuing the journey the pulse is partially reflected by the ice surface and in part by the discontinuities of the medium, and then going through the whole ice thickness reflected by the surface of the rocky bottom (bedrock). All these echoes are detected at different times (delay) at the receiving antenna.

The radar pulses are generated continuously with a repetition rate PRR (Pulse Repetition Rate), which is chosen depending on the speed of the used vehicle to move on the ice (aircraft, snowmobile) and the required horizontal resolution of the measurements. All these acquired traces are plotted basically in two ways. Refer to Figure 2, which represents a radar trace of the Drygalsky ice tongue in Antarctica, measured with the RES system mounted on a plane [Tabacco, 2000; Bianchi, 2001]. In "colour mode" (left), the tracks are represented side by side with points of different colours depending on the amplitude of the echoes, while in the "oscope mode" (right), the track is plotted over time as it is received by the instrument. On this trace you can see the first transmitted pulse, the reflection in the air of the ice surface and on the bottom surface of the glacier (the sea-ice interface of the glacier). Instead the representation in colour allows to display directly the trend of the bottom of the tongue of the glacier, which jumps up and down from the sea (left part) side going deeper close to the ground line (right part).

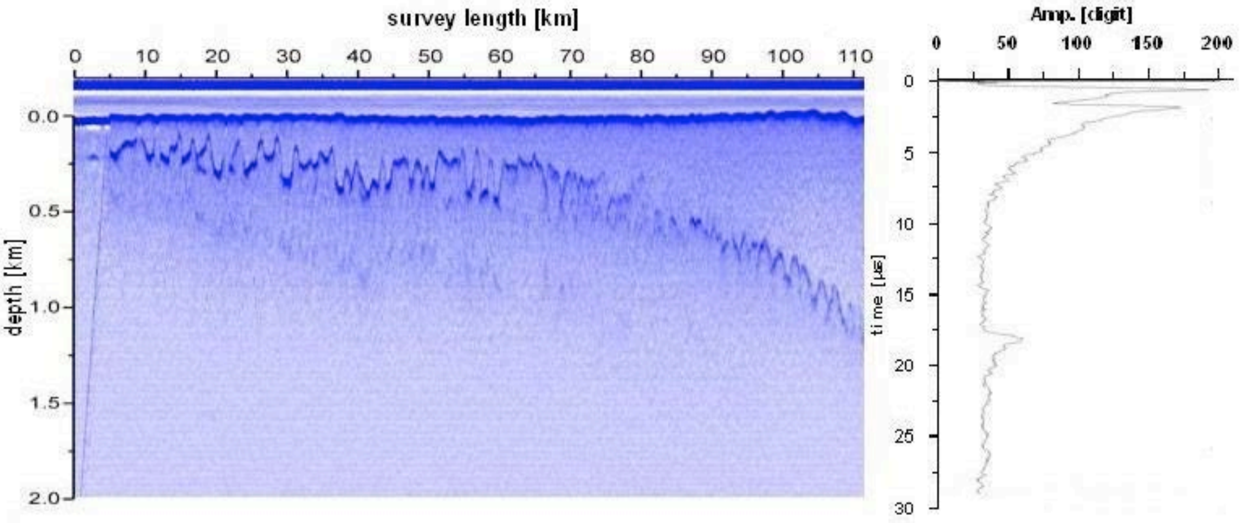

Figure 2. Airplane RES measurement on the Drygalsky ice tongue.

Since 1997, the Istituto Nazionale di Geofisica e Vulcanologia (INGV), as part of the Programma Nazionale di Ricerca in Antartide (PNRA) [Tabacco 1999], is involved in the development of an airborne RES radar system called "glacio RADAR" [Zirizzotti 2008]. The instrument has been constantly updated and improved through several exploration missions [Bianchi 2003].

This radar has been used in several Antarctic expeditions of the Italian National Research Program in particular in the measurement campaigns of 1995, 1997, 1999, 2001, 2003, 2009, 2011 and 2012. All data were collected in the database IRES (Italian Radio Echo Sounding database). This database has been integrated with a WebGIS interface where a photographic view (map) of the locations of the acquired traces and their geographical location are shown. The interface can be reached at http://labtel2.rm.ingv.it/antarctica/.

Moreover in 2010 began the research for the design of a new radar in the PAPRIKA PROJECT SHARE-CNR to perform the measurements of the thickness of the Baltoro Glacier in Pakistan. This glacier is about 60 km long and located in the Karakorum mountain range and is one of the largest valley glaciers in the world. This new project gives us the opportunity to develop instruments that can be used also on the Alps glaciers. At our latitude alpine glaciers (44° - 47° North) and Himalayan (28° - 34° N) have two key differences to Antarctic glaciers: thinner and lower temperature of the ice, they are called temperate glaciers. In Antarctica, the average annual temperature in the interior is very low, for example at Dome C is -54.5 ° C (www.polarnet.cnr.it). The point of maximum thickness of the ice ever measured is 4755 m [Cafarella 2006]. On the Baltoro glacier, a temperate glacier, average surface temperatures are higher than expected, and the thickness is less than 1000m. These differences indicate different measurement modes, in fact the temperatures of the ice leads to higher absorption of electromagnetic waves at high frequencies, while the reduced thickness to be analyzed limits the length of the pulse to be transmitted and therefore its average power. These two conditions make it more difficult paradoxically the radar measurements on temperate glaciers than on the polar ice.

As explained the hard working conditions make RES measurements very difficult. Moreover the choice of correct radar parameters, like pulse length or receiver attenuation factor, are very important to acquire correctly radar traces, have a strong signal from the bedrock and a detailed internal layering. For this reason a new transmitter has been developed with the possibility to transmit two pulses with different length. Short and long pulse can be combined to have in one leg both a high resolution and a high energy response as shown later.

1. Radar waveform

The capabilities for the target detections and the radar measurement characteristics depend on the waveform transmitted and received by the radar. The conventional waveforms of radars are continuous wave (CW) or pulsed signals. CWs radar are suitable for target radial velocity detection while pulsed radar are for precise target ranging. The velocity can be detected from the Doppler effect frequency shift while the range measurements are obtained from the echo time delay. The most common radar transmits a short pulse and listens for target echoes. This technique has the advantage of not having interferences between transmitted and received signals, allowing the use of a single antenna for transmitting and receiving, but precision of the range measurement varies inversely with the pulse length. However, the pulse energy depends on the duration of the transmitted signal, so for a long range radar a high power very short pulses are necessary to obtain a suitable received signal level and precise range detection. On the other hand it is hard to generate such a pulse and a compromise between pulse length and output power level is always needed. This conflict could be solved using the pulse compression, that allows on a large pulse a higher range precision. Several techniques are used for the pulse compression: a pulse can be generated with a linear frequency shift or a numerical code modulated (frequency, amplitude or phase) on the carrier. But demodulation or numerical correlation process, necessary for the peak detection, can cause problems in the use of pulse compressed waveform, in fact extra peaks (sidelobes) appear, degrading the detection capability.

As mentioned the range of a target is calculated using the delay time of the echo:

$$d = \frac{1}{2}\frac{c}{\sqrt{\varepsilon_r}}t \tag{1}$$

where c is the light speed ε_r is the medium dielectric permittivity ($\varepsilon_r = 3.22$ for the ice) and t is the delay time. The range resolution can be evaluated from the range using:

$$\Delta d = \frac{1}{2}\frac{c}{\sqrt{\varepsilon_r}}\tau \tag{2}$$

where τ is the time resolution equal to pulse length in a pulsed radar.

The total radiated energy (on a sphere) of the electromagnetic field is constant and independent from the source distance in a lossless medium [Bo Thide]. The transmitted power calculated on a sphere surface of ray R is also constant and as consequence the power density reduces as $1/R^2$ (spherical divergence). While in the reflection process between two media (interface) with different electromagnetic characteristics (conductivity and dielectric permittivity), the power loss is due to many factors, transmission and reflection losses and electromagnetic absorption of the media.

The power density P_D [W/m^2] at a distance R away from a radar using a directive antenna of gain *G* and a radiated power P_t [W] is then given by:

$$P_D = \frac{P_t G}{4\pi R^2} \tag{3}$$

When the radar radiated energy impinges on a target, the induced surface currents on that target radiate electromagnetic energy in all directions. The amount of the radiated energy is proportional to the target size, orientation, physical shape, and material, which are all lumped together in one target-specific parameter called the Radar Cross Section (RCS).The RCS σ [m^2] is defined as the ratio of the power P_r [W] reflected back to the radar to the power density incident on the target:

$$\sigma = \frac{P_r}{P_D} \tag{4}$$

Thus, the total power delivered to the radar signal processor by the antenna is:

$$P_r = P_D\sigma = \left(P_t G\right)\frac{\sigma}{4\pi R^2}\frac{A_e}{4\pi R^2} \tag{5}$$

where the term *(P_tG)* represents the emitted power in the direction of the target, $\frac{\sigma}{4\pi R^2}$ the fraction of the power backscattered by a target of cross section σ and $\frac{A_e}{4\pi R^2}$ is the fraction of scattered power captured by the receiving antenna of aperture A_e (effective area). It is important to note that the receiving power decreases as a fourth power law thus the difficulty of target detection increases with the range. In order to double the radar range, one must increase the peak transmitted power sixteen times to receive the same power signal at the antenna. Directional antennas are usually characterized by the antenna gain G and the antenna effective aperture A_e. They are related by:

$$A_e = \frac{G\lambda^2}{4\pi} \tag{6}$$

Where λ is the wavelength.

As shown in (5) the received power is not related to the pulse length. For this reason the reduction of the pulse length will increase the range resolution (2) and will not affect the received signal amplitude. This is not completely true in the particular case of multiple reflectors. When the distances between the reflectors are bigger than the space covered by the pulse (short pulse) multiple single thin reflections will arise from the illuminated target. While if the distances between adjacent reflectors are shorter than the pulse the reflections will integrate coherently producing a stronger amplitude of the echo (see the next paragraph).

Therefore, in this scenario it is clear the advantage of using short pulses to increase the range resolution, and of using long pulses to increase the received amplitude. In the case of RES system the received signal power is very important, in fact the bedrock in Antarctica is at a maximum range of 6 kilometers with received signal attenuation greater than 80 dB (mainly due to the electromagnetic ice attenuation). On the other hand pulse compression techniques, used to gain the received signal without reducing the range resolution, can be replaced in Antartica (ground measurements) by short envelope pulses with a large numbers of averaged traces, avoiding sidelobes problems.

For these reasons a new radar has been developed for Glaciological radar measurements. The new RES system exploits the transmission of a double pulse, alternatively large and short. In this way both benefits will be gathered.

2. Double Pulse radar

The realization of the double pulse radar involves the design of a new version of the transmitter with the possibility of double pulse generation and a new acquisition program of the received signal to store the different pulse width data on separate channels for further elaboration.

2.1 Double pulse transmitter

Double pulse transmitter is an upgraded version of the previous single pulse transmitter. The old version is capable of generating Tx pulses at 150 MHz and 300 MHz (on two separate outputs lines) with a complete set of trigger signals for both the Tx signals. In Figure 3 the double pulse and trigger signals timings are shown. In the figure the two pulses at a constant frequency with different pulse width are visible, the pulses are selectable by the user and generated alternatively. The triggers are: AMP trigger to switch on the RF power amplifier, in advance (pre opening time) with respect to the generated pulses, OSC trigger synchronized with the pulses useful for starting the trace acquisition process and oscilloscope visualization of the traces (for radar monitoring) [Bagiacchi, 2012].

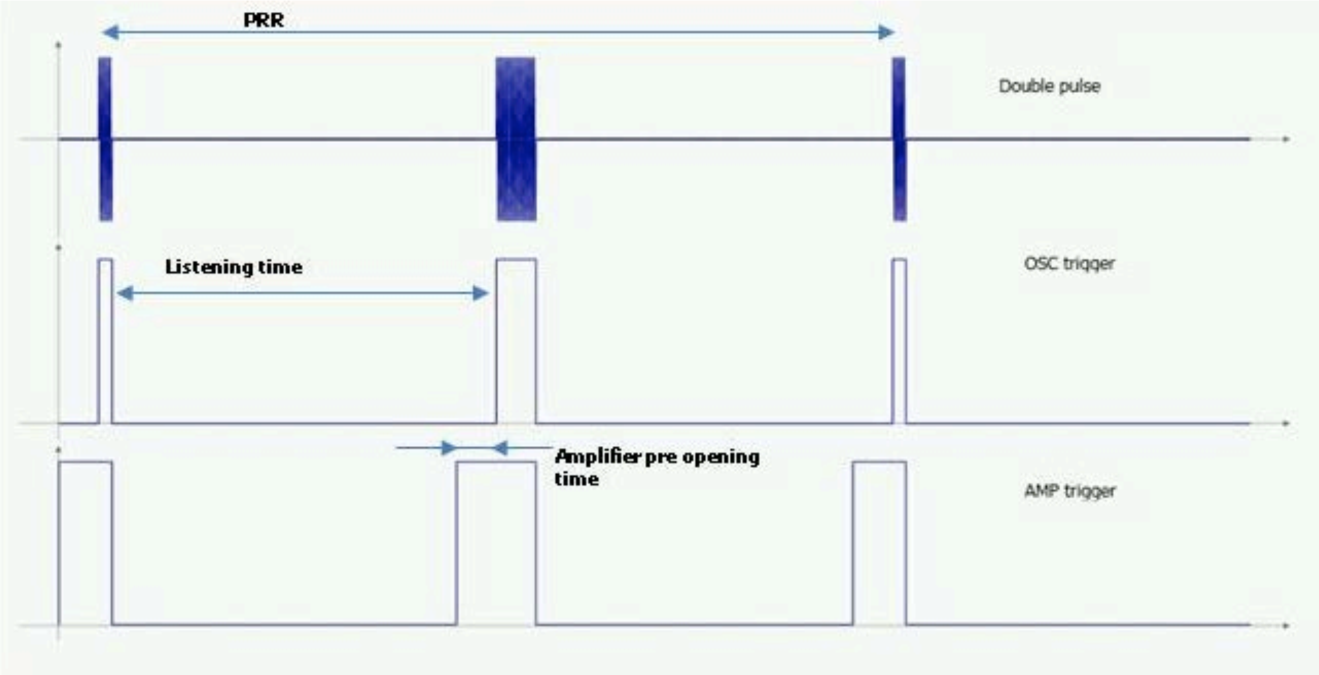

Figure 3. Double pulse and trigger signals.

Amplifier pre opening time is necessary to make sure the amplifier is settled before transmitting the pulse, it is variable and depends on the RF power amplifier characteristics (bandwidth) while the PRR (1 kHz in our radar) depends on the acquisition time of the traces, the speed of vehicle on the ice field and the listening time on the maximum range [Zirizzotti, 2012].

The single pulse version of the transmitter has been developed in the past with an easy procedure. A digital square pulse has been filtered with a band pass filter tuned with the pulse frequency to isolate the fundamental frequency and transform the square digital pulse in a sine wave pulse. This method has various advantages: a) a digital pulse can be created easily using digital circuits; b) the pulse length can be varied from 1 cycle to infinity; c) there are reduced signal to noise ratio problems. Moreover using this method phase changes of the square wave also are maintained after the filtering process, allowing generating phase modulated pulse useful for phase coded radar waveform.

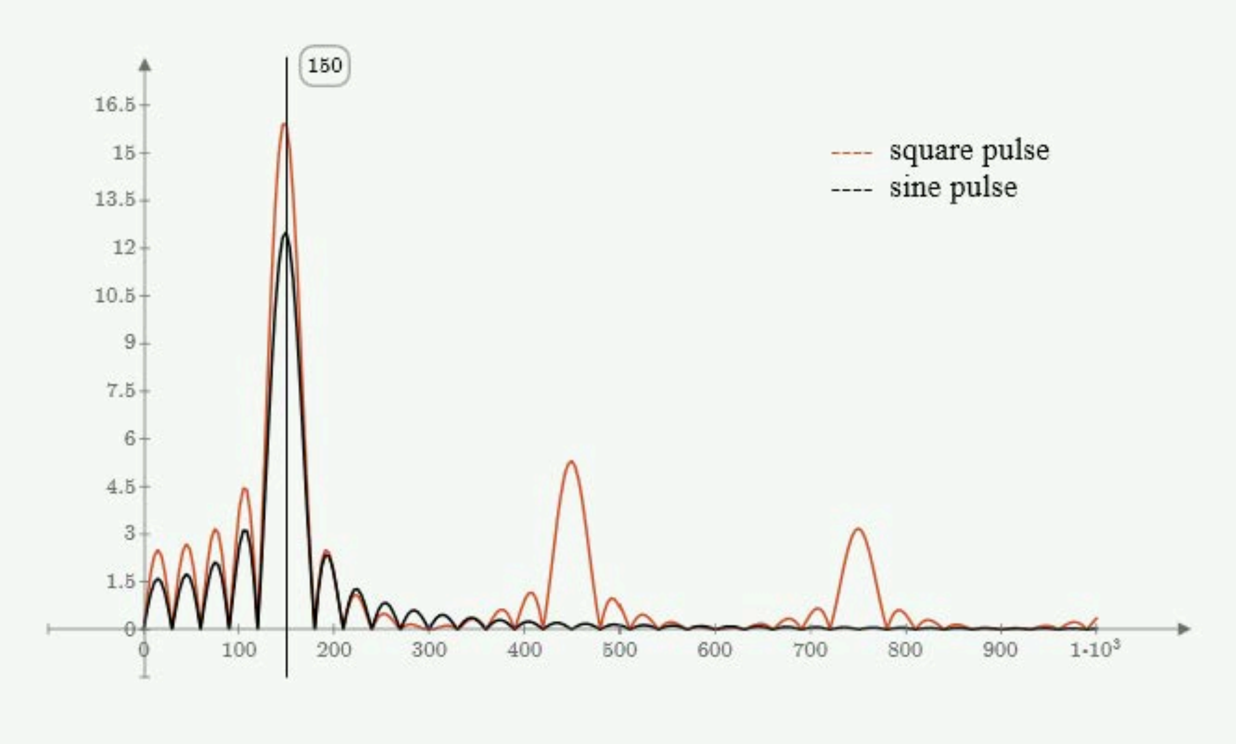

Figure 4. Fourier transform of sine pulse and square pulse.

In fig. 4 the Fourier transforms of a sine pulse and a square pulse of f = 150 MHz and pulse length of about 33 ns are shown. The two spectrums are quite similar until 250MHz (only odd harmonics are present). Filtering the square pulse with a band pass filter (B= 250 MHz) the output will be quite identical to a filtered sine pulse signal (filtering is always necessary to limit the noises).

Using this method double pulse can be generated creating alternate square pulse of different length and filtering the output. Other used signals are the clock reference that is obtained directly from the digital clock and the OSC. Trigger and AMP trigger are generated at the start of both the digital pulses of different length and a new synchronization signal for the large (or short) pulse to distinguish between the pulse length cases (see schematic part).

2.2 Double pulse receiver & acquisition program

The envelope receiver [Zirizzotti 2012] used in the new system has not been changed with respect to the old system (that used a single pulse).The receiver was designed with a bandwidth suitable for the shorter transmitted pulse (50 ns) and it is good also for larger pulses or double pulse transmission method.

Changes have been done in the acquisition program of the received radar trace coded using Visual Studio C++. The program acquires from a Sygnatec PDA 14 acquisition card (14 bits) mounted in the acquisition computer of the radar. The received traces are digitalized with 2048 samples at an acquisition frequency selectable from 25 to 100 MHz (for different radar ranges and resolution). The average of several traces is stored in a binary file for post processing analysis. The process that handles the acquisition was modified to acquire separately the odd and even traces, at the “OSC Trigger” event and averaging and storing them into the file.

3. Transmitter schematic

The design of the digital electronic part of the transmitter has been done using two Complex Programmable Logic Devices (CPLD) from Xilinx. A programmable logic device is a component used to build reconfigurable digital circuits. Programmable logic devices (PLDs) are standard, ready to use parts that offer to the electronic engineer a wide range of logic capacity, features, speed, and voltage characteristics - and these devices can be reconfigured at any time to perform different functions. On the other hand the circuits in a fixed logic device are permanent, they perform one function or set of functions, so once manufactured they cannot be changed. During the prototype definition of new instrumentation or during upgrade of already tested instruments fixed logic can cost the designer in time and resources of silicon foundry, including engineering resources, expensive software design tools, expensive photolithography mask sets for manufacturing the various metal layers of the chip. All these costs can be drastically reduced by the use of programmable logic devices, especially in the design of innovative scientific instrumentation where results and behaviours are not well known. With programmable logic devices, designers use inexpensive software tools to quickly develop, simulate, and test their designs. Then, a design can be quickly programmed into a device, and immediately tested in a live circuit. The CPLD that is used for this prototyping is exactly the same CPLD that will be used in the production of a piece of final equipment. Complex programmable logic devices are evolution of PLD with a higher amount of logic density, features and performances. This type of devices allows us to build instrumentations that can be easily upgraded allowing to redefine new logic functions with increased complexity by replacing only the CPLD program.

Xilinx (in the same way of other factories) offers free software for CPLD programming, the ISE-Web Pack (http://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.htm) is free for downloading at http://www.xilinx.com/support/download/index.htm. Using this software it is possible also to simulate the CPLD functionality allowing evaluating the number of macrocells necessary for the correct operations and the propagation time of the signals through the devices for the precise timing and unwanted conditions valuations.

The schematics of the transmitter is shown in figure 5, it uses two Xilinx CPLDs to generate at the same time two frequencies 150 and 300 MHz. In fact the first CPLD is a XC2C32A a faster CPLD capable of an operating frequency of 323 MHz but with only 32 macrocells. The other used CPLD is the XC95288XL with an operating frequency of 208 MHz but with a greater number of macrocells (288). The primary digital clock signal from a very stable T1185 temperature controlled XTAL (300 MHz) is connected to the GCK0 clock input (pin 6) of U2, while an easy VHDL program generates the 150 MHz secondary clock for the second CPLD at GCK1 input (pin 30) on U1 using a divider by 2 procedure (annex 2). The second CPLD generates all the trigger signals for the radar controls and also for the first CPLD using several counters synchronized with the primary clock. The VHDL program is in annex 3. AMP trigger and OSC trigger, from the 3.3 CPLD logic level are shifted to standard 5 V TTL level using U6.

The digital pulsed signals at 150 and 300 MHz together with their respective reference clocks and pulses are sent to four band pass filters as shown in the analog part of the schematics. Two filters RBP 160 and RBP 400 (Mini Circuits) respectively for the 150 MHz and 300 MHz have been used for this scope in the analog sections. The RBP 160 has a central frequency of 160 MHz and a bandwidth of 90 MHz, while the RBP-400 has a central frequency of 280 MHz with a 50 MHz bandwidth. To increase the signal to noise ratio of the sine pulse, the filter output is sent to a RF switch of the National Instrument the ADG901 controlled by two control signals (CTRL1 and CTRL2) from the second CPLD (pins 74 and 75). To select the pulse width, two thumb wheel switches are connected to the CPLDs. A large number of pins of the CPLD have been connected for future use.

The assembly of the transmitter is shown in figure 6. The power supply together with the transmitter board has been mounted on a "1U" box that can be mounted on a rack for electronic instrumentation. In the front panel the two thumb wheel switches for the pulses width selections are placed, while on the rear panel there is the long row of connectors for different input and output signals.

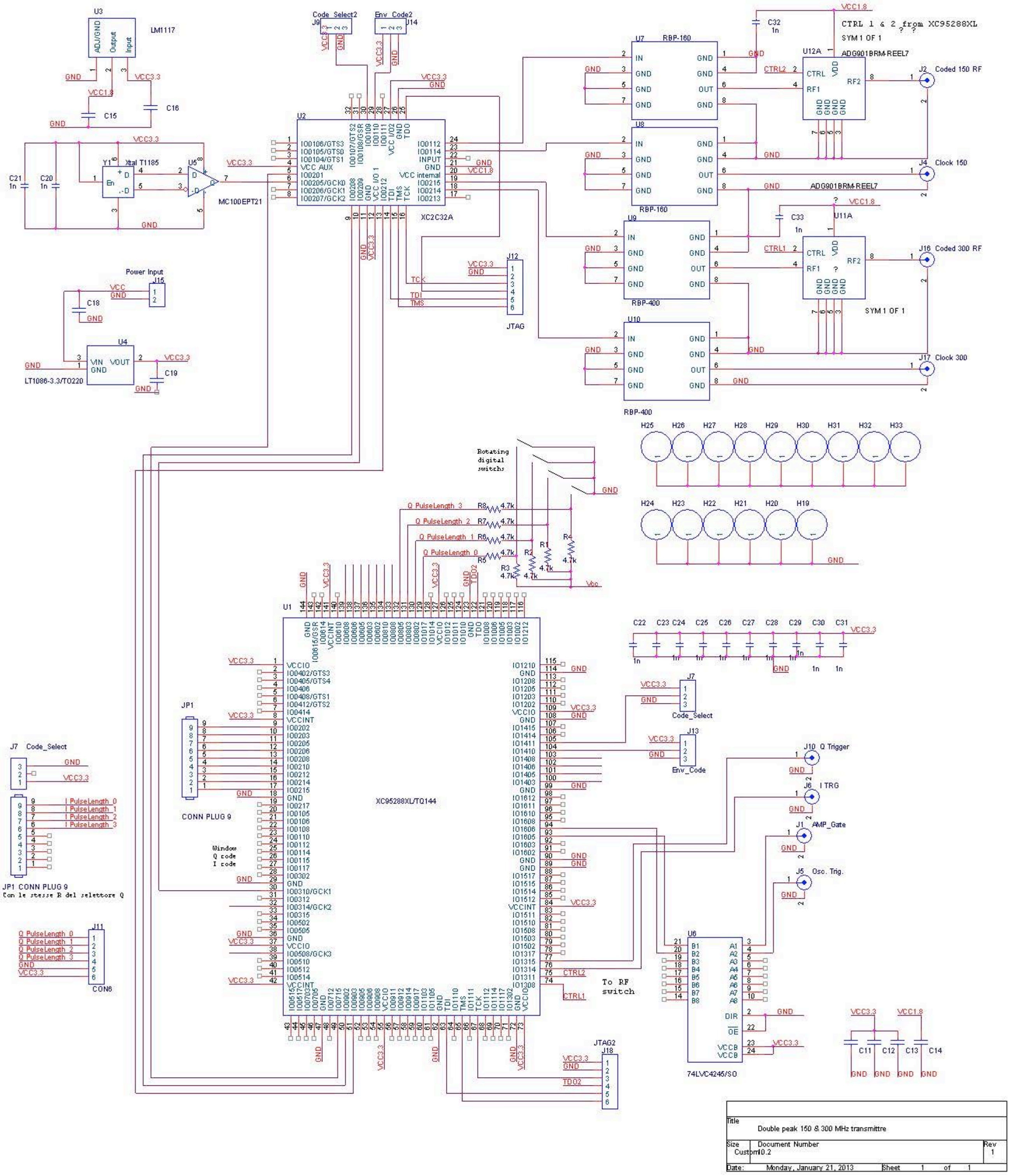

Figure 5. Transmitter schematic.

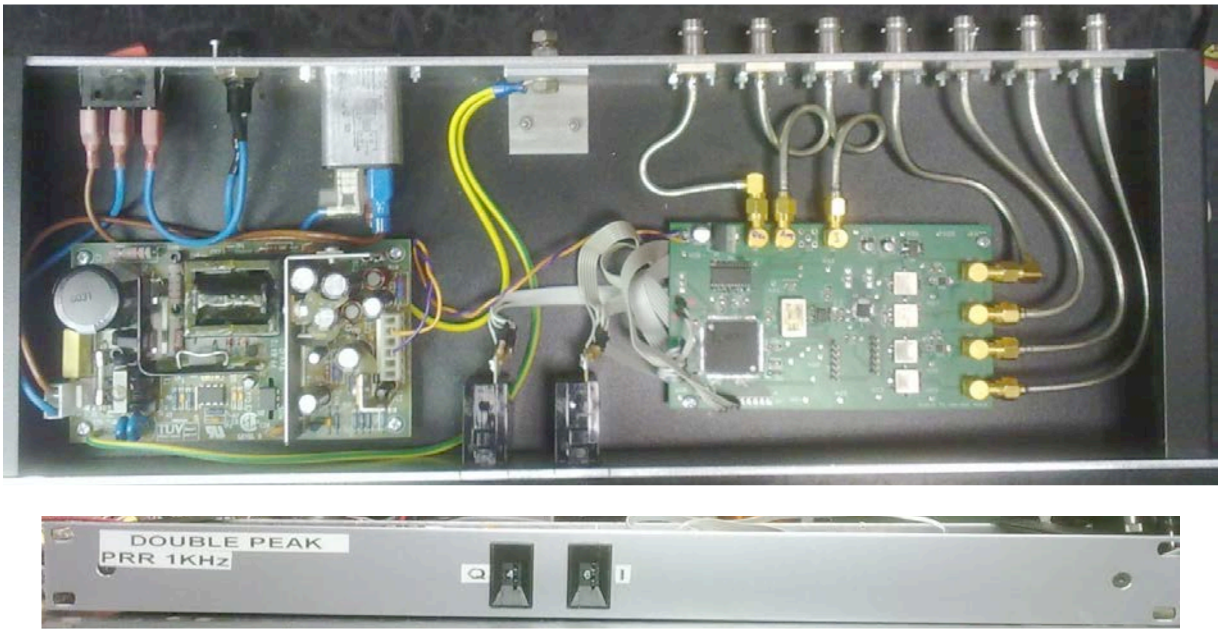

Figure 6. Transmitter assembly on a rack mounting box.

4. CPLD VHDL programs

The description of the transmitter function and the logic relation between digital input and output has been written in the VHDL language [VHDL]. This is not necessarily the only method to describe circuits, easier and graphical methods can translate electronic circuits into VHDL code lines (Visual IPD designer [VIPD]). The CPLDs programs are the most important part of the transmitter (annex 2 and 3). Fundamentally a VHDL program can be divided into two processes: one is of type combinatory, which calculates the next state and the other which serves as a memory element for what will be considered the present state. The process responsible for the calculation of the next state is based on constructs such as “Case – When” while the transition from one state and the next is based on the construct “If - then – else”. The program takes precautions to avoid improper operation. In fact, it must describe the behaviour of the machine for all possible states the synthesizer could produce; moreover, it must highlight the existence of states from which it is not possible to exit (metastability). Both situations have a defined state at power-up.

Annexes 2 and 3 report the two VHDL programs for the two CPLDs used in the transmitter circuit. In the first program after a definition of variables and initial states an “if” statement is used as divider by two to obtain the 150 MHz clock of the second CPLD (clock150). This clock in the second CPLD generates all the trigger signals, in particular the “window”, I and Q signals (here and after I and Q are references not related to “In phase” or “Quadrature” signals) are fed back to the generation of the digital pulses (simple or phase coded). Reference clock and pulses at 150 and 300 MHz are buffered and sent to output analog section.

In the second program, in the same way, after a definition of variables and initial states, two “while case” statements check for the position of the two thumb wheel switches which define the two pulses lengths. Then follows a series of counters synchronized with the secondary clock to generate all the triggers signals with the required timing. At the end, the triggers are buffered and sent out.

5. Radar test

After a complete test in the laboratory of the INGV, a field test of the double pulse radar has been conducted in Antarctica at Concordia station during the last campaign of glaciological radar measurement in 2012. After the positive test the instrument has been used for a campaign of measurement of the bedrock topography in the area of the EPICA drilling site. The results of the test measurements are shown in figure 7. Several received radar traces are plotted as couple for clarity. The first set of plots (red) shows received samples from pulse length of 50 ns (position 1 on the transmitter selector) to the 700 ns (pos. 13). The blue line in each plot corresponds to a radar trace with a longer transmitted pulse with respect to the red line. It is important to note: first plots have a high resolution with a high number of internal layers reflections (sharp peaks), blue lines are always over the red line and this means the larger pulses generate higher power level in the received trace. The received power level depends on the electromagnetic characteristics of the media that form the interfaces. As explained in eq. 5, the received power does not depend on the pulse length, unless a filter or integrator in the receiver is working on the trace signal with a time constant bigger than that of the used pulse length. In this case, integration time constant over different pulses can generate different power levels but this is not the case of our receiver. Otherwise, as explained in annex 1, the receiver bandwidth will not affect the peak power level because the receiver has been designed to work with the shorter pulse.

This can be explained considering the internal layer reflectors as multiple reflectors in a very little range shorter than the space occupied by pulse (range resolution defined by eq. 2). In fact, in this case multiple reflections could rise and will sum coherently together showing a power higher for large pulses. Moreover this explanation can be partially confirmed by the fact that the bedrock reflections (the latest little peaks at the end of the traces) have a quite constant power level of -90dBm, except for the first very short four pulses.

The result of campaign measurement will be soon published.

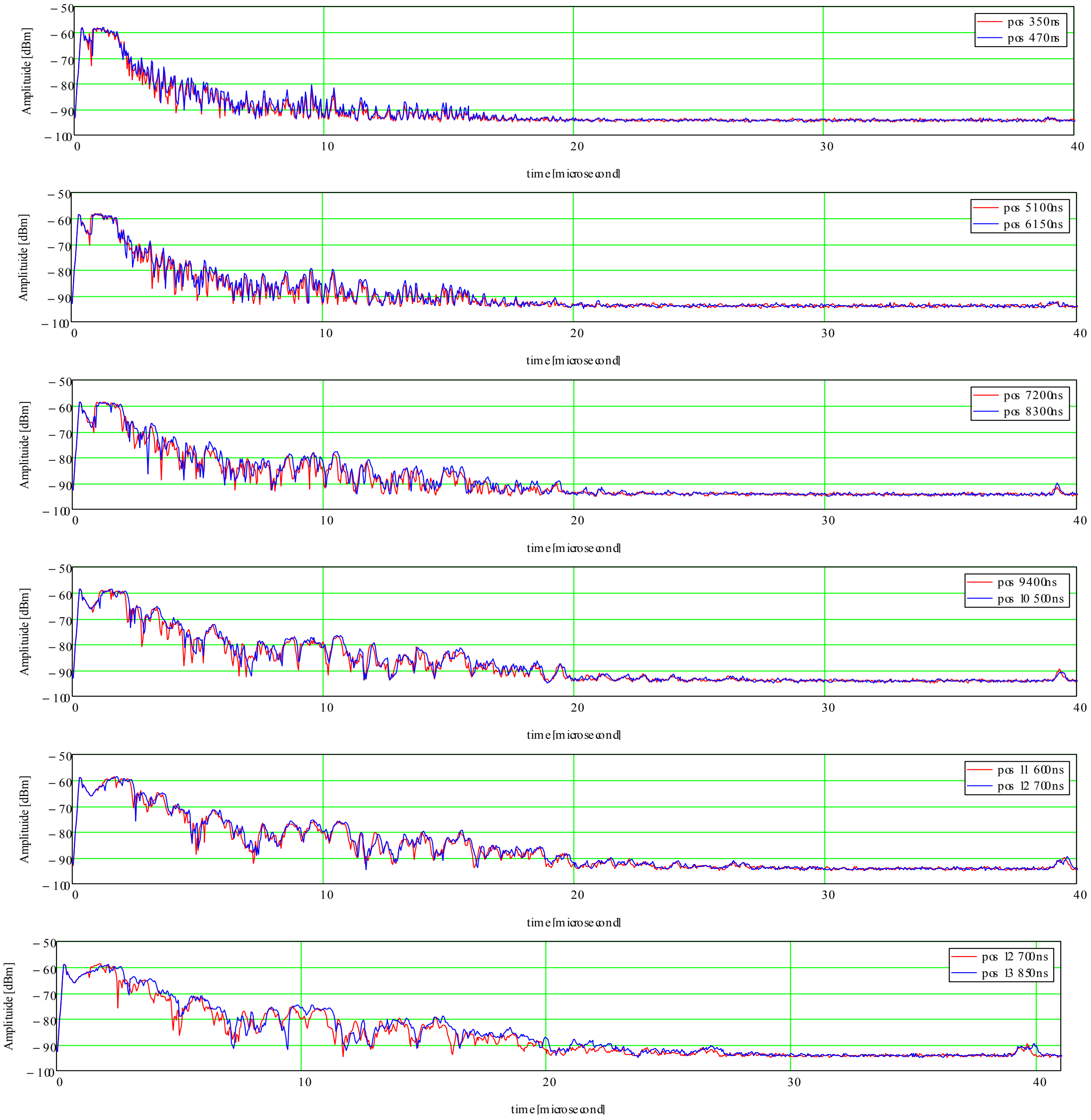

Figure 7. Acquired radar traces for different pulse lengths at the same test position at Concordia Base in Antarctica.

Bibliography

Bagiacchi P., Zirizzotti A., (2012). *Generatore di segnali di sincronizzazione per misure RES*. Rapporti Tecnici INGV n.217.

Bianchi C., Chiappini M., Tabacco I.E., Passerini A., Zirizzotti A., Zuccheretti E., (2001). *Morphology of bottom surfaces of glaciers ice tongues in the East Antarctic region*. Annali di Geofisica Vol. 44 N. 1 February

Bianchi C., Sciacca U., Zirizzotti A., Zuccheretti E., Baskaradas J. A., (2003). *Signal Processing techniques for phase-coded HF-VHF radars*. Annals of Geophysics, 46, 4, 697-705.

Bogorodsky V.V., Bentley C.R., and Gudmandsen P.E., (1985) *Radioglaciology*, pp.272, Reidel Publishing Company, Dordrecht, Holland,.

Cafarella L., Urbini S., Bianchi C., Zirizzotti A., Tabacco I.E., Forieri A., (2006). *Five subglacial lakes and one of Anatrctica's thickest ice covers newly determined by Radio Echo Sounding over the Vostock-Dome C region*. Polar Research Letter, 25, 1, 69-73.

Glen J.W. and Paren J.G., (1975). *The electrical properties of snow ice*. Journal of Glaciology, 151, 73, 15-37.

Frezzotti M., Tabacco I.E. and Zirizzotti A., (2000). *Ice discharge of eastern Dome C drainage area, Antarctica, detemined from airbone radar survey and satellite image analysis*. Journal of Glaciology, vol 46, No 153,.

Forieri A., Tabacco I.E., Della Vedova A., Zirizzotti A., Bianchi C., De Michelis P., Passerini A., (2003). *A new bedrock map of Dome Concordia area*. Terra Antarctica Report, 8, 169-174.

Mancini M., Frezzotti M., Smiraglia C., Gragnani R., Proposito M., Zirizzotti A., Motta M., Pavan M., (2003). *Mass balance of Campbell Glacier (Northern Victoria Land, Antarctica)*. Terra Antarctica Report, 8, 11-14.

Plewes L.A., and Hubbard B., (2001). *A review of the use of radio-echo sounding in glaciology*. Progress in Physical Geography, 25, 2, 203-236.

Tabacco I.E., Bianchi C., Chiappini M., Passerini A., Zirizzotti A., Zuccheretti E., (1999). *Latest improvements for the echo sounding system of the Italian radar glaciological group and measurements in Antartica*. Annali di Geofisica, 42, 2, 271-276.

Tabacco I.E., Bianchi C., Chiappini M., Zirizzotti A., (2000). *Analysis of bottom morphology of the David Glacier - Drygalski Ice Tongue, East Antartica.* Annals of Glaciology, 30, 47-51.

Tabacco I.E., Forieri A., Della Vedova A., Zirizzotti A., Bianchi C., De Michelis P., Passerini A., (2003). *Evidence of 14 new subglacial lakes in Dome C-Vostok area*. Terra Antarctica Report, 8, 175-179.

Thidé B., (2012). *Electromagnetic Field Theory* http://www.plasma.uu.se/CED/Book/EMFT_Book.pdf

Zirizzotti A., Baskaradas J.A., Bianchi C., Sciacca, U., Tabacco I.E., Zuccheretti E., (2008). *Glacio RADAR system and results*. In Proc. RADAR '08, IEEE, 26-30 May 2008 Roma, pp.1 -3.

Zirizzotti A., Cafarella L.,. Baskaradas J. A, Tabacco I. E., Urbini S., Mangialetti M., and Bianchi C., (2010). *Dry-wet bedrock interface detection by radio echo sounding measurements*. IEEE Transactions on Geoscience and Remote Sensing, 48, 2343-2348, DOI:10.1109/TGRS.2009.2038900.

Zirizzotti A., Cafarella L., Urbini S., (2011). *Ice and bedrock characteristics underneath Dome C (Antarctica) from Radio Echo Sounding data analysis*. IEEE Transactions on Geoscience and Remote Sensing, Vol. 99 pp.1-7 DOI:10.1109/TGRS.2011.2160551

Zirizzotti A. Baskaradas J. A., Tutone G., (2012). *Ricevitore a inviluppo per misure RES.* Rapporti Tecnici INGV n.227

VHDL reference book http://tams-www.informatik.uni-hamburg.de/vhdl/doc/cookbook/VHDL-Cookbook.pdf

VIPD ref. http://www.visualipd.com/

Annex 1

Theoretically the received reflected power at a fixed range is independent of the pulse length (with a fixed receiver bandwidth) but it depends only on the transmitted peak power and the electromagnetic characteristics of the reflecting interface.

The receiver response also is independent of the pulse length (for pulse lengths greater than the rising time of the receiver). In fact we can easily sketch the receiver as a CR integrator with a pulse at the input. Fig.8 (integration time τ=1 s, B=0.16 Hz, t_s=2.2 s) shows the integrator response to a pulse of different length. For pulse length greater than the slope time t_S, that depends on receiver bandwidth from:

$$t_S = \frac{0.35}{B}$$

the integrator (receiver) output is constant and equal to the pulse amplitude.

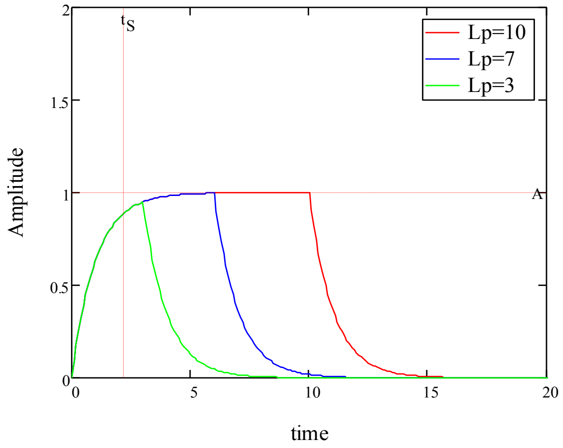

Figure 8. integrator responses on several peaks of different lengths.

While in the case of a very short bandwidth (a large integration time) the response depends on pulse length as shown in figure 9 (τ=2 s, B=0.08 Hz, t_s=4.2 s).

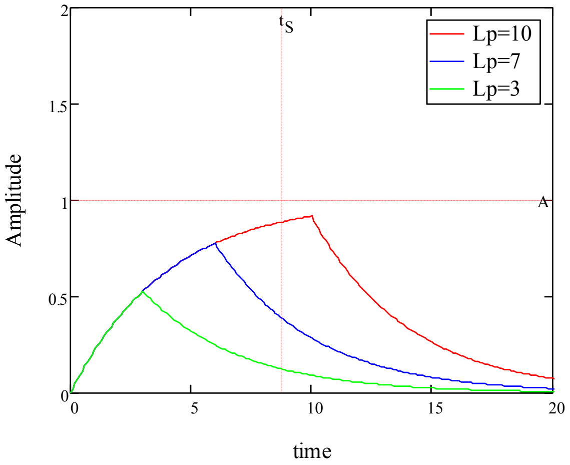

Figure 9. integrator responses on several peaks of different lengths.

In our real case with a receiver bandwidth of 10 MHz, t_s = 35 ns, so for pulse length greater than 50 ns the receiver output will be constant.

Annex 2

VHDL code first CPLD

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity glacio is
    Port ( clock : in std_logic;
         clock300 : out std_logic;
         clock150 : out std_logic;
        code300 : out std_logic;
        code150 : out std_logic;
        clokout : out std_logic;
        Window : in std_logic;
        I : in std_logic;
        Q : in std_logic
                );
end glacio;

architecture Behavioral of glacio is
signal clockbuf: STD_LOGIC := '0';
signal code300buf: STD_LOGIC := '0' ;
signal code150buf: STD_LOGIC := '0' ;

begin
process (clock,Window,I,Q)
begin
        -       Divider by two
if clock='1' and clock'event then
        if clockbuf='1' then
                clockbuf <= '0';
        else
                clockbuf <= '1';
        end if;
end if;--clock event

if window='1' then
                 if I='1' and Q='0' then  --coded
                                code150buf<= clockbuf;
                                code300buf<=   clock ;
                end if;

                if Q='1' and I='0' then                 --coded
                                code150buf<= (not clockbuf);
                                code300buf<= (not clock);
                end if;
                if I='1' and Q='1' then    --Envelope
                                code150buf<=  clockbuf;
                                code300buf<=   clock;
                end if;
                if I='0' and Q='0' then
                        code150buf<= '0';
                        code300buf<='0';
                end if;
end if;
if window='0' then
                        code150buf<= '0';
                        code300buf<='0';
end if;
clock150<=clockbuf;
clock300<= clock;
clokout<=clockbuf;    --clock 150 MHz II° CPLD 95288XL
code150<=not (not code150buf);
code300<=(not (not code300buf));

end process;
end Behavioral;
```

Annex 3

VHDL code second CPLD :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity GlacioII is
   Port ( clock : in std_logic;
              Pulse_length_SEL: in  STD_LOGIC_VECTOR (3 downto 0);
              Pulse_length_SEL2: in  STD_LOGIC_VECTOR (3 downto 0);
              Amp_Gate : out std_logic;
              Trig_Gate : out std_logic;
              I : out std_logic;
              Q : out std_logic;
              window : out std_logic;
              Switch150 : out std_logic;
              Switch300 : out std_logic;
              Iout: out std_logic;
              Qout: out std_logic;
              Trig_Gate3_3: out std_logic
              );
end GlacioII;

architecture Behavioral of GlacioII is
  signal START2: STD_LOGIC := '1';
  signal LOAD3: STD_LOGIC := '1';
  signal START4: STD_LOGIC := '0';
  signal START5: STD_LOGIC := '1';
  signal REP: STD_LOGIC := '0';
  signal REP2: STD_LOGIC := '0';
  signal REP4: STD_LOGIC := '0';
  signal REP5: STD_LOGIC := '0';
  signal Amp_buf: STD_LOGIC := '1';
  signal  Trig_Buf: STD_LOGIC := '0';
  signal Switch : STD_LOGIC := '1';
  signal Switch_buf : STD_LOGIC := '0';
  signal window_buf: STD_LOGIC := '0';
  --------------------
 -- contatore  PRR      --
 --------------------
  signal COUNT: STD_LOGIC_VECTOR (20 downto 0) := CONV_STD_LOGIC_VECTOR (150000, 21);    --PRR 1000 HZ @ 150MHz
  constant DIN: STD_LOGIC_VECTOR (20 downto 0):=   CONV_STD_LOGIC_VECTOR (150000, 21);          --PRR 1000 HZ @ 150MHz
 ----------------------------------------
 --contatore 1 lunghezza Amp gate=Apertura anticipata amplif. +  impulso totale            --
 ----------------------------------------
 subpulse of 66.6 ns     per 16 subpulse
  signal DIN1: STD_LOGIC_VECTOR (8 downto 0)   := "100000110";  -- "11111111" ;--Pulse lenght256-3      16   cicli   in   1   subpulse   of   66.6   ns
              per 16 subpulse
 ----------------------------------------
 --contatore 2 lunghezza Amp gate=Apertura anticipata amplif. +  impulso totale            --
 ----------------------------------------
  signal COUNT2: STD_LOGIC_VECTOR (8 downto 0) := "100000110";--"11111111" ;             --Pulse lenght           256-3     16 cicli in 1 subpulse
       of 66.6 ns           per 16 subpulse
  signal DIN2: STD_LOGIC_VECTOR (8 downto 0)   := "100000110";  -- "11111111" ;             --Pulse lenght256-3       16  cicli  in  1  subpulse  of
       66.6 ns              per 16 subpulse
 ---------------------------------
 -- contatore divisore a 0 MHz  --
 ---------------------------------
  signal COUNT3: STD_LOGIC_VECTOR (2 downto 0) := "011";--/"011" ;
  constant DIN3: STD_LOGIC_VECTOR (2 downto 0) := "011";--/"011";
 -------------------------------
 -- contatore doppio codice          --
 -------------------------------
  signal COUNT4: STD_LOGIC_VECTOR (15 downto 0) := CONV_STD_LOGIC_VECTOR (15000, 16);  --double pulse separation
  constant DIN4: STD_LOGIC_VECTOR (15 downto 0) := CONV_STD_LOGIC_VECTOR (15000, 16);           --double pulse separation
------------------------------------
------------------------------------
--contatore apertura anticipata    --
------------------------------------
  signal COUNT5: STD_LOGIC_VECTOR (7 downto 0) := "00111100";--60*6.666 ns = 400 ns     --contatore apertura anticipata amplificatore
  constant DIN5: STD_LOGIC_VECTOR (7 downto 0) := "00111100";  -- "00111100" ;--contatore apertura anticipata amplificatore
-----------------------------------------------------------------------------------------------------
begin
process (clock,Amp_buf, Trig_Buf, Switch)
      variable Barker: STD_LOGIC_VECTOR (12 downto 0) := "0101001100000";
```

```
     variable Codec2: STD_LOGIC_VECTOR (15 downto 0) := "0010111001111011";
     variable Codec: STD_LOGIC_VECTOR (15 downto 0)  := "1101000101111011";
     variable Codecindex: INTEGER range 0 to 16;
     variable index: INTEGER range 0 to 16;
      variable Env_Code: BOOLEAN  ;
      variable Bar_2Code: BOOLEAN  ;
      variable QPSK: BOOLEAN  ;
begin
if clock='1' and clock'event then
     case Pulse_length_SEL is
      when "1111"  =>                                                                        --0
          Barker:= "1010110011111";                          --          13 subpuls   va scritto al contrario
                    DIN2  <= CONV_STD_LOGIC_VECTOR (255, 9);
                    COUNT2<= CONV_STD_LOGIC_VECTOR (255, 9);
                    Env_Code:= FALSE;
                    Bar_2Code:=TRUE   ;
                    QPSK:=FALSE        ;
      when "1110" =>                                                                         --1
           Codec2:= "0010111001111011";                      -- 16 subpuls
      Codec := "1101000101111011";
          DIN2  <= CONV_STD_LOGIC_VECTOR (300, 9);
                    COUNT2<= CONV_STD_LOGIC_VECTOR (300, 9);
                    Env_Code:=FALSE ;
                    Bar_2Code:=FALSE  ;
                    QPSK:=FALSE        ;
      when "1101" =>                                                                         --2                    QPSK
           Codec2:= "0010111001111011";                      -- 16 subpuls
      Codec := "1101000101111011";
          DIN2  <= CONV_STD_LOGIC_VECTOR (300, 9);
                    COUNT2<= CONV_STD_LOGIC_VECTOR (300, 9);
                    Env_Code:=FALSE ;
                    Bar_2Code:=FALSE  ;
                    QPSK:=TRUE         ;
      when "1100"  =>                                                                        --3           50 ns
          QPSK:=FALSE        ;
          Env_Code:=TRUE ;
          Bar_2Code:=FALSE  ;                          --      60*6.666 ns + pulse length
          DIN2<= CONV_STD_LOGIC_VECTOR (68, 9);              -- 75*6.6666 ns=500 ns = COUNT5 + PULSE LENGTH
      when "1011"  =>                                                                        --4                    100 ns
       QPSK:=FALSE             ;
          Env_Code:=TRUE ;
          Bar_2Code:=FALSE  ;
          DIN2<= CONV_STD_LOGIC_VECTOR (71, 9);
      when "1010"  =>                                                                        --5                    150 ns
      QPSK:=FALSE              ;
          Env_Code:=TRUE ;
          Bar_2Code:=FALSE  ;
          DIN2<= CONV_STD_LOGIC_VECTOR (75, 9);
      when "1001"  =>                                                                        --6                    200 ns
       QPSK:=FALSE             ;
          Env_Code:=TRUE ;
          Bar_2Code:=FALSE  ;
          DIN2<= CONV_STD_LOGIC_VECTOR (82, 9);
      when "1000"  =>                                                                        --7           250 ns
       QPSK:=FALSE             ;
          Env_Code:=TRUE ;
          Bar_2Code:=FALSE  ;
          DIN2<=CONV_STD_LOGIC_VECTOR (90, 9);
      when "0111"  =>                                                               --8                    300 ns
       QPSK:=FALSE             ;
          Env_Code:=TRUE ;
          Bar_2Code:=FALSE  ;
          DIN2<= CONV_STD_LOGIC_VECTOR (105, 9);
      when "0110"  =>                                                      --9--                           500 us
       QPSK:=FALSE             ;
          Env_Code:=TRUE ;
          Bar_2Code:=FALSE  ;
          DIN2<= CONV_STD_LOGIC_VECTOR (120, 9);
      when "0101"  =>                                               -- 10 --  600 us
       QPSK:=FALSE             ;
          Env_Code:=TRUE ;
          Bar_2Code:=FALSE  ;
          DIN2<= CONV_STD_LOGIC_VECTOR (135, 9);
      when "0100"  =>                                                               --8                    700 ns
       QPSK:=FALSE             ;
          Env_Code:=TRUE ;
          Bar_2Code:=FALSE  ;
          DIN2<= CONV_STD_LOGIC_VECTOR (150, 9);
      when "0011"  =>
```

```
  QPSK:=FALSE              ;
     Env_Code:=TRUE ;
     Bar_2Code:=FALSE  ;
     DIN2<= CONV_STD_LOGIC_VECTOR (165, 9);
 when "0010"  =>                                                                                   --9--
      1 us
  QPSK:=FALSE              ;
     Env_Code:=TRUE ;
     Bar_2Code:=FALSE  ;
     DIN2<= CONV_STD_LOGIC_VECTOR (185, 9);
 when "0001"  =>                                                                                   --9--
      1.? us
  QPSK:=FALSE              ;
     Env_Code:=TRUE ;
     Bar_2Code:=FALSE  ;
     DIN2<= CONV_STD_LOGIC_VECTOR (210, 9);
 when "0000"  =>                                                                                   --9--
      2 us
  QPSK:=FALSE              ;
     Env_Code:=TRUE ;
     Bar_2Code:=FALSE  ;
     DIN2<= CONV_STD_LOGIC_VECTOR (400, 9);
 when others =>
  QPSK:=FALSE              ;
  Env_Code:=TRUE ;
     Bar_2Code:=FALSE  ;
     DIN2<= CONV_STD_LOGIC_VECTOR (400, 9);
end case;

case Pulse_length_SEL2 is
 when "1111"  =>                                                                            --0
     Barker:= "1010110011111";                           --          13 subpuls   va scritto al contrario
               DIN1  <= CONV_STD_LOGIC_VECTOR (255, 9);
               --COUNT1<= CONV_STD_LOGIC_VECTOR (255, 9);
               --Env_Code:= FALSE;
               --Bar_2Code:=TRUE ;
               --QPSK:=FALSE       ;
 when "1110" =>                                                                             --1
      Codec2:= "0010111001111011";                       -- 16 subpuls
 Codec := "1101000101111011";
     DIN1  <= CONV_STD_LOGIC_VECTOR (300, 9);
               --COUNT1<= CONV_STD_LOGIC_VECTOR (300, 9);
               --Env_Code:=FALSE ;
               --Bar_2Code:=FALSE               ;
               --QPSK:=FALSE       ;
 when "1101" =>                                                                             --2                QPSK
      Codec2:= "0010111001111011";                       -- 16 subpuls
 Codec := "1101000101111011";
     DIN1  <= CONV_STD_LOGIC_VECTOR (300, 9);
               --COUNT1<= CONV_STD_LOGIC_VECTOR (300, 9);
               --Env_Code:=FALSE ;
               --Bar_2Code:=FALSE               ;
               --QPSK:=TRUE        ;
 when "1100"  =>                                                                            --3          50 ns
     QPSK:=FALSE        ;
     Env_Code:=TRUE ;
     Bar_2Code:=FALSE  ;                          --      60*6.666 ns + pulse length
     DIN1<= CONV_STD_LOGIC_VECTOR (68, 9);              -- 75*6.6666 ns=500 ns = COUNT5 + PULSE LENGTH
 when "1011"  =>                                                                            --4                 100 ns
  QPSK:=FALSE              ;
     Env_Code:=TRUE ;
     Bar_2Code:=FALSE  ;
     DIN1<= CONV_STD_LOGIC_VECTOR (71, 9);
 when "1010"  =>                                                                            --5                 150 ns
 QPSK:=FALSE               ;
     Env_Code:=TRUE ;
     Bar_2Code:=FALSE  ;
     DIN1<= CONV_STD_LOGIC_VECTOR (75, 9);
 when "1001"  =>                                                                            --6                 200 ns
  QPSK:=FALSE              ;
     Env_Code:=TRUE ;
     Bar_2Code:=FALSE  ;
     DIN1<= CONV_STD_LOGIC_VECTOR (82, 9);
 when "1000"  =>                                                                            --7          250 ns
  QPSK:=FALSE              ;
     Env_Code:=TRUE ;
     Bar_2Code:=FALSE  ;
     DIN1<=CONV_STD_LOGIC_VECTOR (90, 9);
 when "0111"  =>                                                                     --8                300 ns
```

```
        QPSK:=FALSE             ;
            Env_Code:=TRUE ;
            Bar_2Code:=FALSE  ;
            DIN1<= CONV_STD_LOGIC_VECTOR (105, 9);
       when "0110"  =>                                                         --9--                          500 us
        QPSK:=FALSE             ;
            Env_Code:=TRUE ;
            Bar_2Code:=FALSE  ;
            DIN1<= CONV_STD_LOGIC_VECTOR (120, 9);
       when "0101"  =>                                                  -- 10 --  600 us
        QPSK:=FALSE             ;
            Env_Code:=TRUE ;
            Bar_2Code:=FALSE  ;
            DIN1<= CONV_STD_LOGIC_VECTOR (135, 9);
       when "0100"  =>                                                                   --8                       700 ns
        QPSK:=FALSE             ;
            Env_Code:=TRUE ;
            Bar_2Code:=FALSE  ;
            DIN1<= CONV_STD_LOGIC_VECTOR (150, 9);
       when "0011"  =>
        QPSK:=FALSE             ;
            Env_Code:=TRUE ;
            Bar_2Code:=FALSE  ;
            DIN1<= CONV_STD_LOGIC_VECTOR (165, 9);
       when "0010"  =>                                                                                        --9--
            1 us
        QPSK:=FALSE             ;
            Env_Code:=TRUE ;
            Bar_2Code:=FALSE  ;
            DIN1<= CONV_STD_LOGIC_VECTOR (185, 9);
       when "0001"  =>                                                                                        --9--
            1.? us
        QPSK:=FALSE             ;
            Env_Code:=TRUE ;
            Bar_2Code:=FALSE  ;
            DIN1<= CONV_STD_LOGIC_VECTOR (210, 9);
       when "0000"  =>                                                                                        --9--
            2 us
        QPSK:=FALSE             ;
            Env_Code:=TRUE ;
            Bar_2Code:=FALSE  ;
            DIN1<= CONV_STD_LOGIC_VECTOR (400, 9);
       when others =>
        QPSK:=FALSE             ;
        Env_Code:=TRUE ;
            Bar_2Code:=FALSE  ;
            DIN1<= CONV_STD_LOGIC_VECTOR (400, 9);
      end case;
      COUNT <= COUNT - 1;     --listening time
      REP <= (not COUNT(0)) and (not COUNT(1)) and ( not COUNT(2)) and (not COUNT(3))
                  and  (not   COUNT(4)) and (not COUNT(5)) and ( not COUNT(6)) and (not COUNT(7))
                  and (not    COUNT(8)) and (not COUNT(9)) and ( not COUNT(10)) and (not COUNT(11))
                  and (not   COUNT(12)) and (not COUNT(13)) and ( not COUNT(14)) and (not COUNT(15))
                  and(not   COUNT(16)) and (not COUNT(17)) and ( not COUNT(18)) and (not COUNT(19))
                  and (not   COUNT(20));
      if REP='1' then
                  COUNT <= DIN;
                          --Switch <='1';
                          START2<='1';
                          START5<='1';-- AGGIUNTO
                          Amp_buf <= '1';
      end if;
--parte secondo contatore per la lunghezza dell'impulso
       if START2='1' then
                  COUNT2 <= COUNT2 - 1;           --tx pulse
        end if;
            REP2 <= (not COUNT2(0)) and (not COUNT2(1)) and ( not COUNT2(2)) and (not COUNT2(3)) and
                                   (not COUNT2(4)) and (not COUNT2(5)) and ( not COUNT2(6)) and (not COUNT2(7))and (not COUNT2(8))  ;
        if REP2='1' then   --
                  START2<='0';
                  Amp_buf <= '0';
                  if Env_Code=FALSE then
                          COUNT2 <= DIN2;
                          if Bar_2Code=TRUE then           --Barker Code
                                  Switch <='0';
                          else                                                                                    --Compl. 2 Code
                                  if Switch ='1' then
                                          START4<='1';
                                          Switch <='0';
```

```
                                else
                                        START4<='0';
                                end if;
                        end if;
                else                                                                    -- Envelope
                        if Switch ='1' then
                                        COUNT2 <= DIN2;
                                        Switch <='0';
                        else
                                        COUNT2 <= DIN1;
                                        Switch<='1';
                        end if;
                end if;
    end if;
--parte quarto contatore per il doppio impulso
        if START4='1'  then
                COUNT4 <= COUNT4 - 1;           --twin pulse
        end if;
        REP4 <= (not COUNT4(0)) and (not COUNT4(1)) and ( not COUNT4(2)) and (not COUNT4(3)) and
                                 (not COUNT4(4)) and (not COUNT4(5)) and ( not COUNT4(6)) and (not COUNT4(7)) and
                                 (not       COUNT4(8)) and (not COUNT4(9)) and ( not COUNT4(10)) and (not COUNT4(11)) and
                                 (not   COUNT4(12)) and (not COUNT4(13)) and ( not COUNT4(14)) and (not COUNT4(15))       ;

      if REP4='1' then   --
                COUNT4 <= DIN4;
                Amp_buf <= '1';
                START2<='1';
                START4<='0';
                Switch <='0';
      end if;
      if START5='1' then
                COUNT5 <= COUNT5 - 1;
    end if;
    REP5 <= (not COUNT5(0)) and (not COUNT5(1)) and ( not COUNT5(2)) and (not COUNT5(3)) and
                                 (not COUNT5(4)) and (not COUNT5(5)) and ( not COUNT5(6)) and (not COUNT5(7))  ;
     if REP5='1' then
                START5<='0';
                Trig_Buf<='1';
    end if;
    if Trig_buf='1' then
                index := index + 1;
                if Bar_2Code=TRUE then                  --Barker code
                        if index= 15 then               --13                                    old value=10
                                index:=0;
                                Codecindex := Codecindex + 1;
                                if Codecindex = 13 then
                                        Codecindex := 0;
                                end if;
                        end if;
                else
                        if index= 15 then               --16        complementary code                      old value=10
                                index:=0;
                                Codecindex := Codecindex + 1;
                                if Codecindex = 16 then
                                        Codecindex := 0;
                                end if;
                        end if;
                end if;
    end if;

end if;                                                         --End clock
if Amp_buf = '1' then
    if Trig_buf='1' then
        if Env_Code=FALSE then
                if Bar_2Code=TRUE then                  --BARKER code
                        if Switch='1' then
                                window_buf<='1';
                                I<=Barker(Codecindex);
                                Q<=not Barker(Codecindex);
                        end if;
                else                                    -- Complementary code 2 shot
                        if QPSK=FALSE then
                                if Switch='1' then              -- First shot
                                        window_buf<='1';
                                        I<=Codec(Codecindex);
                                        Q<=not Codec(Codecindex);
                                else                            --second shot
                                        window_buf<='1';
                                        I<=Codec2(Codecindex);
                                        Q<=not Codec2(Codecindex);
                                end if;
                        else                   -- single shot complementary code
                                                        window_buf<='1';
                                                        I<='1';
                                                        Q<='1';
                                                        Iout<=Codec(Codecindex);
                                                        Qout<=Codec2(Codecindex);
                        end if;
                end if;
        else                                            -- envelope radar
                window_buf<='1';
                I<='1';
                Q<='1';
                if Switch='1' then
                        Iout<='1';
                else
                        Iout<='0';
                end if;
        end if;
    else
         window_buf<='0';
      I<='0';
         Q<='0';
         Switch_buf<='0';
    end if;
else
    window_buf<='0';
    I<='0';
    Q<='0';
    Iout<='0';
    Qout<='0';
    Codecindex := 0;
    index:=0;
    --START5<='1';
    Trig_Buf<='0';
    COUNT5 <= DIN5;
    Switch_buf<='0';
end if;


window<=window_buf;
Switch300<=Trig_Buf ;
Trig_Gate<=Trig_Buf;
Switch150 <= Trig_Buf   ;
Amp_Gate <= Amp_buf     ;
Trig_Gate3_3<=window_buf;

end process;


end Behavioral;
```

Coordinamento editoriale e impaginazione
Centro Editoriale Nazionale | INGV

Progetto grafico e redazionale
Daniela Riposati | Laboratorio Grafica e Immagini | INGV

Via di Vigna Murata, 605
00143 Roma
Tel. +39 06518601 Fax +39 065041181

http://www.ingv.it

Istituto Nazionale di Geofisica e Vulcanologia