UNIVERSITÀ DEGLI STUDI DI FIRENZE
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE (DINFO)
CORSO DI DOTTORATO IN INGEGNERIA DELL'INFORMAZIONE

CURRICULUM: ELETTRONICA ED ELETTROMAGNETISMO
SSD: ING-INF/01

ADVANCED ELECTRONIC SYSTEMS FOR DOPPLER ULTRASOUND APPLICATIONS

Candidate
Dario Russo

Supervisor
Prof. Stefano Ricci

PhD Coordinator
Prof. Fabio Schoen

PhD CYCLE XXXIII, 2017-2020

Università degli Studi di Firenze, Dipartimento di Ingegneria dell'Informazione (DINFO).

October 2020

Acknowledgments

I would like to sincerely thank my supervisor, Prof. Stefano Ricci, for his time, motivation and knowledgeable advices. I would also thank Prof. Piero Tortoli and my lab mates for the good times, their help and support and all the fun and “schiacciata” we have in these years. A big thanks also to my girlfriend who shares with me the ups and downs of my studies over these years. A special thank to my family, my parents Salvatore e Maria, and my brother Luca. They given up many things for me over the years, supported me, encouraged me to follow my way and gave a lot of advices and I will always be grateful to them.

Contents

Introduction

Objective

Doppler ultrasound techniques are nowadays widely employed in biomedical and industrial applications due to their non-invasive and non-destructive features. Their application spreads from industrial systems finalized, for example, to the characterization of industrial suspensions, to complex biomedical apparatuses like echographs. In the last decades, ultrasound techniques have been continuously growing in both these fields, proposing novel methods and electronic systems. The experimentation of a novel Doppler method and the development of a new electronic system require several tests, which are typically carried out by ultrasound Doppler phantoms and flow-rigs. These consist in hydraulic systems where a pump pushes a scattering fluid through a structure that mimics a morphological tissue or an industrial part. Although modern phantoms have high quality and performances, they are still affected by several problems. For instance, the choice and the preparation of the materials for the phantom realization is not trivial; the mimicked vessel wall can introduce distortion on the ultrasound beam and affect the flow dynamics; the scattering fluid preparation is not easy and requires a long time. However, the most significant flaw is probably the lack of an accurate ground-truth for the velocity distribution of the flow present in the phantom, which limits the evaluation of the accuracy of the method/system under tests in velocity measurements.

This PhD work was dedicated to the realization of an accurate testing system for the evaluation of Doppler methods and electronics in both industrial and biomedical fields.

The work is divided in two main parts. In the first a special synchronization circuit, implemented in a Field Programmable Gate Array (FPGA), is designed. This circuit is propaedeutic to the realization of the electronic flow rig. In fact, Doppler analysis is based on the detection of the phase difference among echoes acquired in subsequent time, any error (jitter) on this phase can possibly destroy the Doppler information. For this reason, a severe synchronization, in the order of 100 ps, is required between the electronic phantom and the Doppler system under test. Sharing a common clock can be the solution, but often the clock signal is not accessible in the system under test. The proposed synchronization circuit is capable of generating a clock signal phased to an external aperiodic pulse with the required accuracy, thus solving the problem.

The second part is dedicated to the realization of the “Flow Emulator”. It is a flexible electronic Doppler phantom that is able to generate in real-time the radio-frequency echo signals of a real-like and programmable flow configuration. The Flow Emulator synchronizes to a pulse signal generated by the device under test thanks to the circuit previously developed. In this way, the Flow Emulator can replace a flow-rig in Doppler tests by injecting the echo signal directly in the receiving channel of the Doppler system under test. Alternatively, a transducer can be used for the acoustical coupling with a multi-channel system. The model used for the signal generation is based on the summation of single scatterers contributions, which is implemented in a last-generation Field Programmable Gate Array (FPGA). Unlike other electronic Doppler phantoms, the Flow Emulator allows to emulate the transit time effect, the limited sample volume and disturbances like clutter, background noise and in-depth attenuation. The emulator is a single channel system and it is ideal to be coupled to single channel Doppler systems like most industrial sensors and some specific biomedical devices. However, its employment with multi-channel echographs is still effective to evaluate methods based on the reception of a single Doppler line.

The manuscript is organized as follows:

- Chapter 1: the fundamental of the ultrasonic wave propagation, the characteristic parameters of the propagation media, the structure of the ultrasound transducers and pulsed wave system are briefly described. Basics on the Ultrasound Velocity Profiling (UVP) method is summarized.

- Chapter 2: a brief introduction on the Rheology basics and Newtonian and non-Newtonian models and pipe flow is reported. Industrial and biomedical UVP applications are summarized and two example systems are briefly described.

- Chapter 3: the novel clock synchronization method and circuit developed is described in details. Two versions of this circuit were implemented in different FPGAs. The reported experiments show the synchronization capability of this circuit for both the implementations. Moreover, experiments on the effect of the clock synchronization on the Doppler analysis are reported.

- Chapter 4: the designed Flow Emulator system is described in detail. Two versions of this system are reported, based on different hardware and FPGAs. The first version is the simplest since it only allows the off-line signal generation. It was used mainly for industrial rheological tests. The second version allows both off-line and real-time generations, providing more flexibility. The measurement setups and results for both versions are reported.

Contributions

Journal papers

- **Russo, Dario**, Ricci Stefano. «Electronic Flow Emulator for Ultrasound Doppler Investigations». *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2020. (Submitted)
- **Russo, Dario**, Ricci Stefano. «FPGA Implementation of a Synchronization Circuit for Arbitrary Trigger Sequences». *IEEE Transactions on Instrumentation and Measurement*, 2019.

Conference proceedings

- **Russo D.**, Ricci S., «FPGA-based Trigger-Synchronizer for low Frame-Jitter Signal Generation». In *IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2019.
- **Russo D.**, Ricci S., «Industrial Fluids Electronic Emulator for Rheological Doppler Tests». In *IEEE International Ultrasonics Symposium (IUS)* 2019.
- **Russo D.**, Ricci S., «Low-Jitter Systems Synchronization for Doppler Measurements». In *IEEE International Ultrasonics Symposium (IUS)*, 2019.
- **Russo, Dario**, Ricci Stefano. «FPGA-based Clock Phase Alignment Circuit for Frame Jitter Reduction». In *Applications in Electronics Pervading Industry, Environment and Society*, 2019.
- **Russo, D.**, V. Meacci, and Ricci S. «Profile Generator for Ultrasound Doppler Systems». In 2018 *New Generation of CAS (NGCAS)*, 33–36, 2018.
- **Russo, Dario**, Valentino Meacci, and Stefano Ricci. «Electronics System for Velocity Profile Emulation». In *Applications in Electronics Pervading Industry, Environment and Society*, pp 101–107. Springer International Publishing, 2019.
- Meacci, Valentino, Enrico Boni, Alessandro Dallai, Alessandro Ramalli, Monica Scaringella, Francesco Guidi, **Dario Russo**, e Stefano Ricci. «FPGA-Based Multi Cycle Parallel Architecture for Real-Time Processing in Ultrasound Applications». In *Applications in Electronics Pervading Industry, Environment and Society*, pp 295–301. Springer International Publishing, 2019.

Chapter 1. Ultrasounds Basics

This chapter briefly describes the Ultrasound basic propagation principles, the structure of ultrasound transducers, the Ultrasound Pulsed Wave systems and the Ultrasound Velocity Profiling (UVP) technique.

1.1 Ultrasound waves

Ultrasounds consists in mechanical waves that propagate from a vibrating source through a medium constituted by solids, fluids or gases. These waves generate a perturbation of the medium particles, which moves around the equilibrium position, transmitting the perturbation to the adjacent particles. Ultrasounds are characterized by sound waves at the frequency above the range of the human hearing, i.e. frequencies higher than 20 kHz [1].

1.1.1 Ultrasound propagation

Ultrasound waves can propagate in three modes (Fig. 1), depending on the medium they are propagating in and the way the particles move:

- Longitudinal waves
- Shear waves
- Surface waves

In longitudinal waves, the oscillation of the medium particles occurs in the direction of the wave propagation (blue arrow in Fig. 1). Since compressional and dilatational forces are active in these waves, they are also called pressure or compressional waves. These waves propagate in liquid, as well as solid or gas

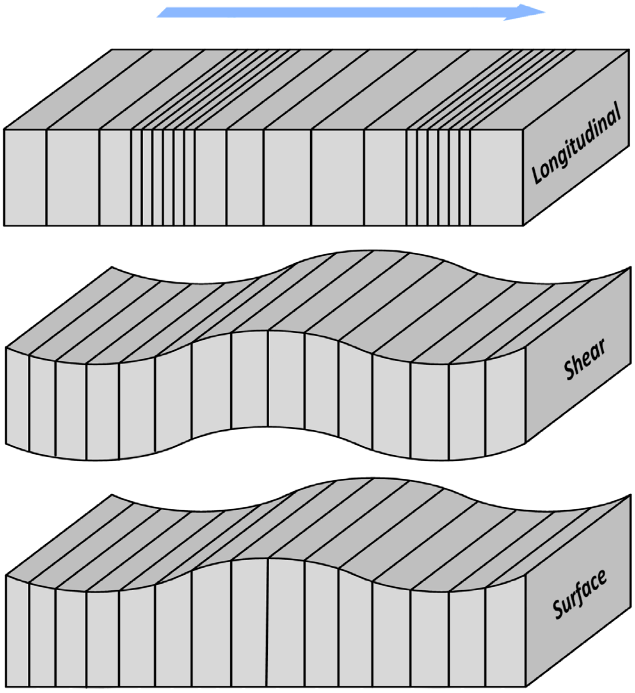


Fig. 1: Propagation modes of the Ultrasound Waves.

because the energy travels in the medium by a series of compression and expansion movements.

In shear or transverse waves, the medium particles oscillate transverse to the direction of the wave propagation. These waves propagate only in solids, since in liquids and gases they are very attenuated, given that in these media the tangential stress develops only thanks to the viscosity.

The surface waves, also called Rayleigh waves, propagate only on the surface of a solid penetrating to a depth of one wavelength. The waves propagation depends on the elastic properties of the medium as well as its mass density.

1.1.2 Linear and Non-Linear Propagation

In linear propagation, the ultrasound wave travel inside the medium without changes in the shape of the wave. The ultrasound wave propagation along the z direction is described by a second order partial differential equation:

$$\frac{1}{c^2}\frac{\partial^2 p}{\partial t^2}=\frac{\partial^2 p}{\partial z^2} \qquad (1)$$

where p is the acoustic pressure and c the sound speed in the medium. The solution of (1) is the well-known plane wave function:

$$p(z,t)=p_0 e^{j(kz\pm 2\pi f t)} \qquad (2)$$

where p_0 is the amplitude of the wave at $z=0$ and k the wave number (equal to $2\pi/\lambda$ with λ the wave length).

The propagation speed of the acoustic wave c is strictly dependent on the elastic properties and the density of the medium and it can be expressed as:

$$c=\sqrt{\beta/\rho} \qquad (3)$$

where ρ is the volumetric medium density and β is the elastic constant that describes how the density changes in relation with the pressure. For example, in a perfectly elastic medium in steady pressure and temperature conditions, the speed of sound can be considered constant since ρ and β are so.

Another important medium property is the acoustic impedance Z, also used for the characterization of the medium itself, that can be expressed as:

$$Z=\rho c \qquad (4)$$

Actually, the propagation of the ultrasound is non-linear, it means that the shape of the ultrasound wave changes during its propagation, losing proportionality to the shape of the excitation. In this case, the speed of sound isn't constant but changes in relation with the pressure. Indeed, in a compressible medium, an increase in pressure causes an increase in temperature and consequently a higher propagation speed. Therefore, a wave travels faster at higher pressure and slower at lower pressure. The most important effect of the non-linearity is the harmonic generation: the speed variations due to pressure changing during the wave propagation modifies the spectral content of the propagating signal, generating harmonics. For example, if the original signal is a sine wave at frequency f, the energy will be spread in multiples of this frequency, nf called harmonics (with n positive integer).

The non-linear pressure-density relation can be found by expanding the linear relation $p = c^2\rho$ as Taylor series:

$$\begin{cases} p - p_0 = A\left(\frac{\rho - \rho_0}{\rho_0}\right) + \frac{B}{2!}\left(\frac{\rho - \rho_0}{\rho_0}\right)^2 +.. \\ A = \rho_0\left(\frac{\partial p}{\partial \rho}\right) = \rho_0 {c_0}^2 \\ B = {\rho_0}^2\left(\frac{\partial^2 p}{\partial \rho^2}\right) \end{cases} \tag{5}$$

where c_0 is the sound speed at the density ρ_0.

The speed of sound in the non-linear propagation can be expressed as:

$$c = c_0 + \left(1 + \frac{B}{2A}\right) v_p \tag{6}$$

where v_p is the particle velocity. Then, the non-linear coefficient of the medium is defined as:

$$\beta = 1 + \frac{B}{2A} \tag{7}$$

1.1.3 Waves Reflection and Transmission

When a wave encounters a boundary between two media with different acoustic impedance, Z_1 and Z_2, part of its energy is reflected and part transmitted in the second medium. If I is the intensity of the incident wave, the reflected and transmitted intensities are expressed by the following relations:

$$I = \frac{p^2}{2Z_1} \tag{8}$$

$$I_r = \Gamma \cdot I \tag{9}$$

$$I_t = \mathrm{T} \cdot I \tag{10}$$

where p is the acoustic pressure of the incident wave, Γ and T are the reflection and transmission coefficient respectively, given by:

$$\Gamma = \left(\frac{Z_1 - Z_2}{Z_1 + Z_2}\right)^2 \tag{11}$$

$$\mathrm{T} = \frac{4 \cdot Z_1 \cdot Z_2}{(Z_1 + Z_2)^2} \tag{12}$$

Obviously, for the energy balance the following equation must be verified:

$$\Gamma + \mathrm{T} = 1 \tag{13}$$

1.1.4 Wave Refraction

When a wave travels from one medium to another with an incidence angle other than 0, the phenomenon of refraction of the transmitted angle occurs as shown in Fig. 2. The Snell law, reported below, lets to determine the transmitted (or refracted) angle in the second medium:

$$\frac{sin\ (\theta_i)}{c_1} = \frac{sin\ (\theta_t)}{c_2} \tag{14}$$

where θ_i is the incidence angle, θ_t the transmission angle, c_1 and c_2 the propagation speed in the medium 1 and 2, respectively. The refraction phenomena occurs when θ_i is below the "critical angle" θ_c, that is the incidence angle which

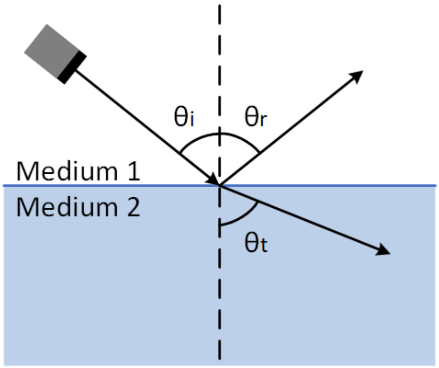


Fig. 2: Reflection and Refraction phenomena with incident angle θ_i.

corresponds to a refracted angle of 90 degrees, that can be found from the Snell law as:

$$\theta_c = \sin^{-1}\left(\frac{c_1}{c_2}\right) \tag{15}$$

Otherwise ($\theta_i > \theta_c$), the incident wave it totally reflected.

1.1.5 Scattering, Attenuation and Absorption

Ultrasound wave in a homogeneous medium (i.e. characterized by physical and chemical properties independent from space and time) propagates along straight lines but, if it meets an interface (boundary between two regions with different acoustic impedance) smaller or compatible to its wavelength, part of its energy is transmitted through the interface and part is spread isotropically in all directions. This phenomenon is called "scattering" and it is quantified by the scattering cross section σ, defined as:

$$\sigma = \frac{S}{I} \tag{16}$$

where S is the total spread power and I the intensity of the incident wave. Instead, when a wave meets an interface whose roughness is larger than the wavelength of the ultrasound wave, reflection and refraction phenomena occur.

During the propagation of an ultrasound wave through a medium, part of its energy is lost as heat (absorption) and part in reflection, scattering, etc. (attenuation). For a plane wave that propagates in a medium with non-zero absorption, an additional exponential-decay factor must be added to the wave equation, as following shown:

$$p(z,t) = p_0 e^{j(kz \pm 2\pi f t)} e^{-j\alpha z} \tag{17}$$

where α is the medium absorption coefficeint.

1.2 Transducers

Ultrasound transducers are capable of converting electrical signals into mechanical vibration and vice versa. The transducers used in the tests reported in this thesis are piezoelectric transducers which are the most commonly used in biomedical and industrial applications.

1.2.1 Piezoelectric Effect

Piezoelectricity is a natural characteristic of some crystals to produce an electric field when subjected to a mechanical stress. The electric field is generated by the deformation of the crystal lattice that consequently will be no more neutral. This effect is called "Direct Piezoelectric Effect" (see Fig. 3-top). Vice versa, applying an alternate electric field to the crystal, it starts to oscillate and produces mechanical waves (Inverse Piezoelectric Effect, Fig. 3-bottom). The more the frequency of the applied electric field is close to the natural frequency of the crystal, the higher is the amplitude of the mechanical wave.

1.2.2 Piezoelectric transducer structure

The most common materials used to realize piezoelectric transducer are piezoelectric ceramics, like the lead zirconate titanate (PZT), or polymers, like polyvinylidene fluoride (PVDF), given their piezoelectric propriety and acoustic impedance similar to the fluids or tissues to be investigates in most cases.

The structure of an ultrasound transducer is shown in Fig. 4. The piezoelectric crystal, placed near the external surface of the transducer, is $\lambda/2$ thick, where λ is the wavelength evaluated at the nominal frequency of the transducer. Both sides of the crystal are laminated and works as electrodes, linked to the cable that carries the transmission and the echo signals. A matching layer ($\lambda/4$ thick) is placed on the crystal to match its acoustic impedance to the one of the tissue or fluid to optimize the energy transfer. An acoustic lens, placed on the top of the transducer, focalizes the ultrasound beam in the desired point. The lens and the matching layer

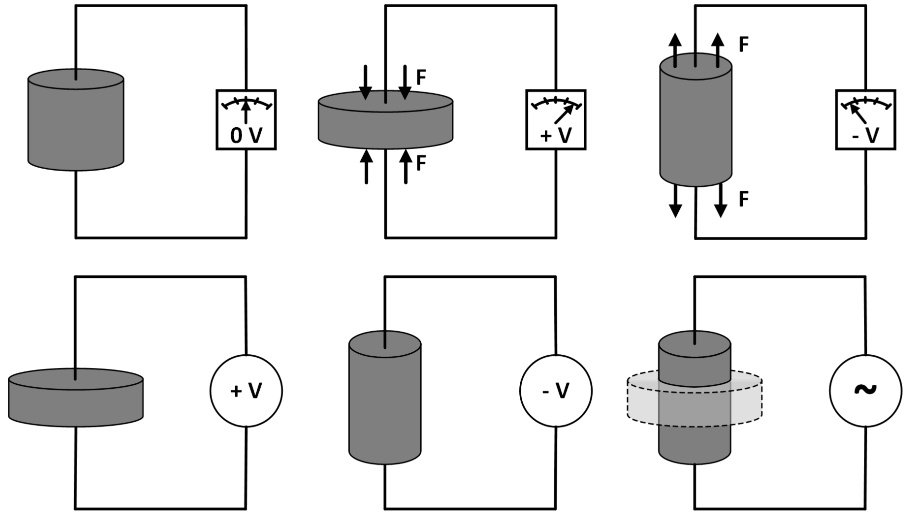


Fig. 3: Direct (top) and Inverse (bottom) Piezoelectric Effect.

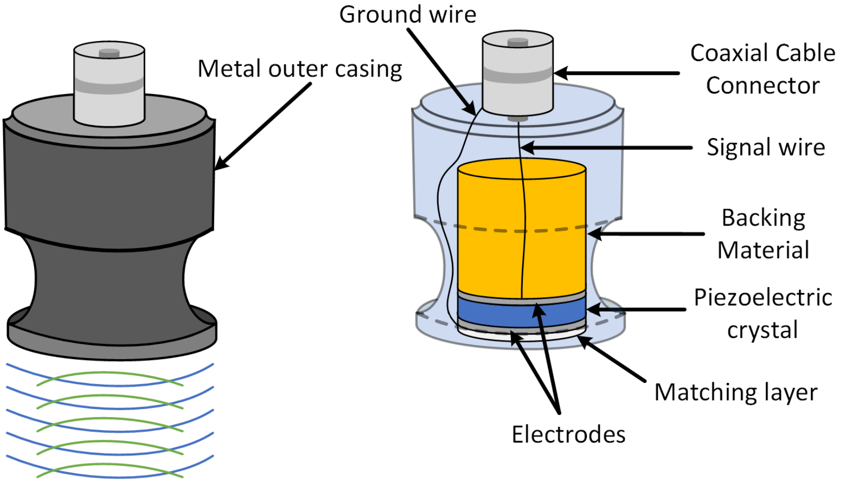


Fig. 4: Piezoelectric transducer structure.

are often designed to “weigh” the infinitesimal contributions produced by each infinitesimal section of the crystal surface (apodization), thus it is possible to reduce some side effect, like side lobes.

The resonance frequency of the piezoelectric crystal is expressed as:

$$f_{res} = \frac{c_{piezo}}{2A} \tag{18}$$

where c_{piezo} is the sound speed inside the piezoelectric crystal and A is its thickness. Even if the sensitivity of the crystal is very high, its bandwidth is very narrow around f_{res}, limiting the use of short burst signals and producing very long oscillation that reduces the resolution of the ultrasound system. For this reason, a “backing layer” is placed on the back side of the crystal to absorb the reflected waves coming from the transducer-fluid interface and fade the crystal oscillation. Despite the backing layer reduces the efficiency of the conversion, the transducer bandwidth increases. The bandwidth is typically expressed as fractional bandwidth, i.e. the bandwidth normalized to the center frequency:

$$\Delta B_{\%} = 100 \cdot \frac{f_2 - f_1}{f_{res}} \tag{19}$$

where f_1 and f_2 are the lower and upper frequencies at which the amplitude response is decreased by 3dB with respect to f_{res}.

Finally, the transducer is enclosed in a metal case grounded to minimize the sensitivity to external electromagnetic disturbances.

1.2.3 Acoustic Beam

The acoustic field generated by a transducer in the surrounding space depends on its geometry. For instance, the acoustic field generated by a cylindrical transducer is reported in Fig. 5. There are two main zones: the first, called “near field” or “Fresnel zone”, in which the field is approximatively cylindrical; the second, referred to as “far field” or “Fraunhofer zone”, in which the beam diverges. The limit between these zones is placed at a distance from the transducer equal to:

$$z = \frac{r^2}{\lambda} \tag{20}$$

where r is the transducer radius and λ is the wavelength of the transmitted signal.

The acoustic beam consists of a main lobe (highest intensity) and side lobes of lower intensity due to the constructive and destructive interferences of the waves generated from each point of the transducer. The origin of these lobes can be demonstrated by the diffraction theory, that states the diffracted beam in the far field zone has the same shape of the Fourier transform of the beam on the aperture, i.e. the transducer surface that generates the beam. Thus, the side lobes are generated by the lobes of the sinc-shape function related to the Fourier transform of finite apertures.

Usually, the sensitivity of an ultrasound transducer is increased by using an acoustic lens. The latter is made of a specific material with an ultrasound propagation velocity different to that of the fluid or tissue to be investigated. By properly designing the lens, the beam pressure can be maximized at a specific point, i.e. the beam is focused in that point. The lens also reduces the divergence of the beam in the far field zone. Thus, the sensitivity to objects close to the focus is increased respect to other objects.

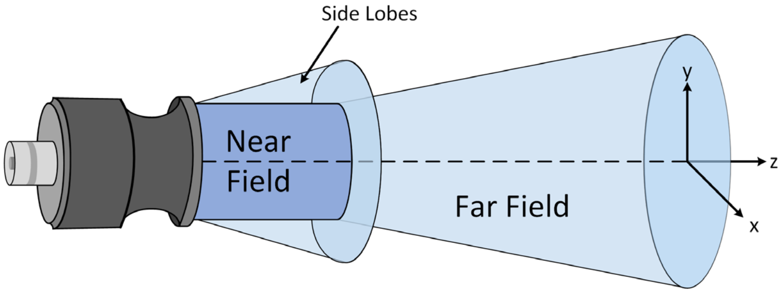


Fig. 5: Acoustic field generated by a cylindrical transducer.

1.2.4 Axial and Lateral Resolutions

Axial (or longitudinal) resolution indicates the minimum distance that can be differentiated between two objects in the direction parallel to the ultrasound beam. It is equal to half of the spatial pulse length or:

$$res_{ax} = \frac{N\lambda}{2} \tag{21}$$

where N is the number of the cycles in the transmitted burst and λ is the wavelength of the transmitted signal. Consequently, the shorter the spatial pulse length, the higher the axial resolution. Thus, increasing the frequency of the transmitted pulse or reducing the number of cycles improves the axial resolution.

By contrast, lateral resolution is the ability to distinguish two objects in the direction perpendicular to the one of the ultrasound beam. It can be approximated with the width of the acoustic beam. As previously stated, the beam width changes moving away from the transducer: thus, the lateral resolution is maximum in the focus and decreases at higher depth.

1.2.5 Array transducers

An array transducer is composed of several little radiating elements placed one close to the other, that can be excited individually. Fig. 6 reports the basic structure of a linear array, whose elements of width *w* are placed with a periodicity p, called "pitch". As for the single element transducer, an acoustic lens is used to focalizes the beam, defining a focal distance on the elevation plane yz. The beam generated by such a narrow element (order of hundreds of µm) will diverge very rapidly, resulting in a poor lateral resolution. For this reason, adjacent elements are used simultaneously to achieve a wider aperture and a more useful beam shape. Indeed, the main advantage of the array transducers is the high flexibility due to the individual control of each element. Exciting each element with a properly delayed signal, it is possible to focus the beam in different depths and steers it at different directions ("electronic focusing", "beamforming"). Moreover, it is possible to change the beam shape by changing the amplitude, width and shape of the apodization, i.e. the weight applied to each element. So, the array transducers let to electronically change the acoustic beam characteristics, as focus, position and direction. This control is also possible in reception, elaborating the echo signal received by each element of the array. The "reception beamforming" consists in properly delaying the echo signal received by each element and summing it to the other. It lets to dynamically focus the reception along the entire axis of interest, unlike the normal transducers that have a focus fixed in a predefined point.

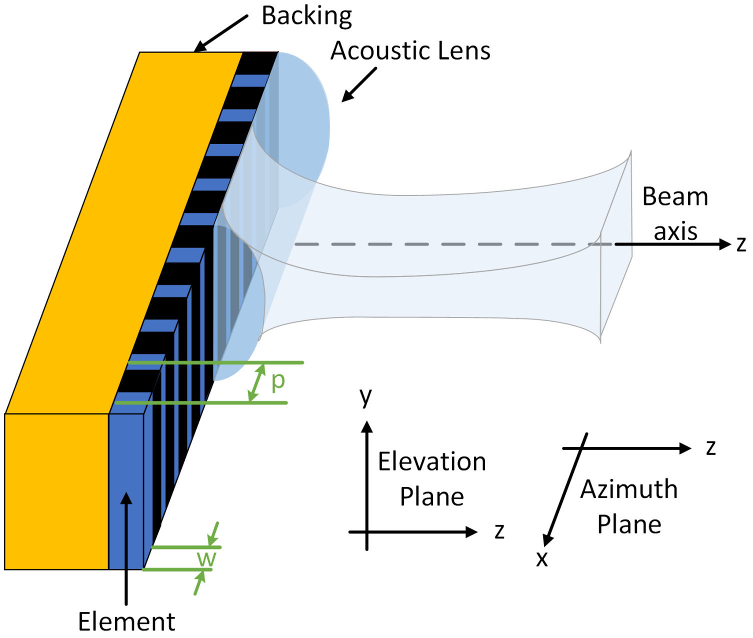


Fig. 6: Array transducer structure.

1.3 Pulsed Wave Systems

Pulsed Wave Systems (PWS) use a single transducer both in transmission and reception, unlike Continuous Wave Systems (CWS) that need at least two transducers, one to transmit continuously the investigation signal and the other to receive the echoes. PWS periodically sends short duration pulses at Pulse Repetition Frequency (PRF) rate, called “burst”, composed by a programmable number of sinusoidal cycles or by a single short pulse. In the temporal interval between two successive bursts, it listens to the echoes. For this reason, PWS can uses a single transducer. The burst transmitted by the transducer propagates in the medium and produces echoes when it encounters particles (scatterers) or impedance discontinuities. The echoes generated at a depth d propagate back to the transducer after a temporal interval Δt from the transmission of the burst, that can be expressed as:

$$\Delta t = \frac{2d}{c} \tag{22}$$

where c is the speed of ultrasound in the medium of interest.

Depending on the number of listening windows (called “gates”) between two successive bursts, a PWS can be a “single” or “multi-gate” system, as shown in

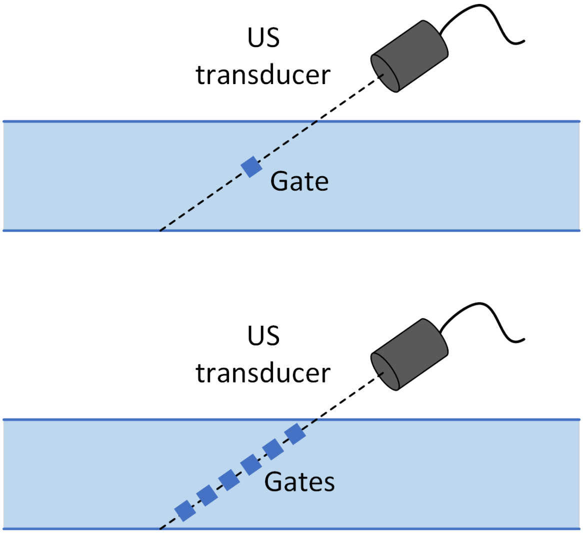


Fig. 7: Single (top) and multi-gate (bottom) systems.

Fig. 7. A single-gate system can isolate the echo coming from only one depth while a multi-gate system can discriminate different depths at the same time. In both cases, the range of depth that can be investigated depends on the duration D of the burst and the PRF used. In fact, the system, after the transmission, has to switch in reception mode to avoid the overlap between the transmitted and received waves. Therefore, the system can’t detect echoes from distances less than:

$$d_{min} = \frac{c \cdot D}{2} \tag{23}$$

Moreover, it will not be possible to receive an echo that takes longer than $\frac{1}{PRF}$ to reach the receiver, thus the maximum depth that can be investigated is:

$$d_{max} = \frac{c}{2 \cdot PRF} \tag{24}$$

The generic architecture of a PWS is shown in Fig. 8. There are two main sections: an “Analog Front-End” (AFE) and a “Digital Control Unit” (DCU). The latter is the core of the system, controls all the operations and the processing. It embeds programmable digital devices like Field Programmable Gate Array (FPGA) or Digital Signal Processing (DSP), memories and interfaces needed to configure the system and download the processed data. In a system that manages few transducers, a single programmable device is enough to process the data and handle the system operations while in more complex systems, like echographs, the architecture is composed by more programmable devices with a dedicated fast

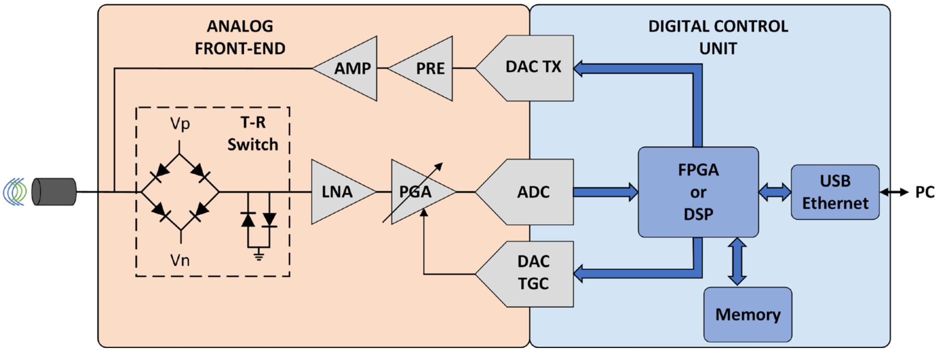


Fig. 8: Pulsed Wave System architecture.

memory to store data. The AFE, in contrast, embeds the analog devices that handle the transmission and reception signals of the transducer, properly amplifying them.

During the transmission, the DCU generates the samples of the excitation signal that are analog-converted through a Digital-to-Analog Converter (DAC). The AFE amplifies this signal through the TX chain, composed by a preamplifier and a high voltage amplifier, used to increase the amplitude of the signal up to ±100V. Then, the signal pass through a "T-R switch" to prevent the high voltage signals coming from the TX chain damaging the RX chain that typically works with signal of ±100mV amplitude. The T-R switch is basically a bridge of biased diodes used to clamp the signal with amplitude over the diode threshold. So, the high amplitude transmission signals can't reach the RX chain, while the low amplitude echoes pass through the T-R switch and go to the first device of the RX chain, that is a Low Noise Amplifier (LNA). The latter is the most important device in reception because its characteristics affect the noise performance of the entire system. In particular, it should have a high noise performance in the system bandwidth to introduce a very low amount of noise and distortion in order to guarantee an high Signal-to-Noise Ratio (SNR). The following device is a Programmable Gain Amplifier (PGA), that acts as Time Gain Amplifier (TGC). In fact, during the propagation in a medium, the ultrasound signal attenuates, and the echoes strength decrease with the time (the more the signal travels, the more it fades). So, the TGC compensates the echo signal attenuation due to the signal roundtrip. The gain of the TGC, controlled by the DCU, is typically a ramp as function of time to adapt the echo signal to the dynamics of the Analog-to-Digital Converter (ADC). Then the ADC digitalizes the amplified echoes and the DCU section process them to obtain results which, finally, will be downloaded to a PC.

1.4 Ultrasound Velocity Profiling (UVP)

1.4.1 UVP Method

The Ultrasound Velocity Profiling (UVP) is a method widely used in biomedical applications [2] to assesses velocity of blood which flows in a vessel or artery and in industries to evaluate the velocity profile of suspensions that flow in industrial pipes [3]. In this technique, a burst $S_{TX}(t)$ is transmitted every Pulse Repetition Interval (PRI) into the medium, that is the fluid of interest (blood, industrial suspension, etc.). When the burst encounter moving particles of the fluid, the burst produces an echo affected by a frequency shift related to the particle axial velocity v_z (see Fig. 9):

$$f_D = \frac{2 f_{TX} v_z}{c} \tag{25}$$

$$v_z = |v| \cos \vartheta \tag{26}$$

where f_D is the Doppler shift frequency, f_{TX} is the transmission frequency, c the sound velocity in the medium, v the particle velocity and θ the angle between the ultrasound beam and the flow direction.

The spatial velocity distribution along the axis of an emitted pulse can be obtained by measuring the Doppler shift at several hundreds of depths aligned along the beam. Let consider a single scatterer, i.e. a single particle of the fluid, that is moving with an axial velocity v_z. The echo generated when the ultrasound beam meets the scatterer generates a signal whose analytical description is $S_{RX}(t)$:

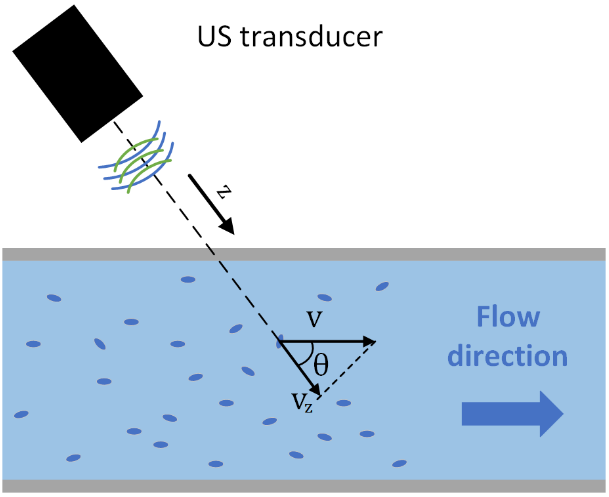


Fig. 9: Transducer-pipe configuration.

$$S_{RX}(t) = x(t)e^{j[2\pi(f_{TX}-f_D)t+\varphi_0]} \tag{27}$$

where φ_0 is the initial phase and $x(t)$ depends on several factors as the system's impulse response, the attenuation of the medium and the transmitted signal $S_{TX}(t)$.

In each PRI, a Phase-Quadrature (IQ) demodulation is applied to the echo signal $S_{RX}(t)$ to remove the transmission frequency f_{TX} and obtain the complex base band signal $S_{BB}(t)$:

$$S_{BB}(t) = S_{RX}(t)e^{-j2\pi f_{TX}} = x(t)e^{-j(2\pi f_D t+\varphi_0)} \tag{28}$$

The in-phase and quadrature components of $S_{BB}(t)$ are stored along the column of a matrix. When enough PRIs are stored, the data are read out by row in an operation known as "corner turning" (see Fig. 10). Each row represents the Doppler samples collected at the same depth. Then the power spectrum $f_{PS}(f)$ is evaluated by summing the squares of the spectra of the weighted read out data. A power spectral matrix like that shown in Fig. 11-left can be obtained by plotting the power spectra of multiple depths. A weighted mean is typically performed on every spectrum to obtain a single Doppler frequency from each depth:

$$f_{PS_m} = \frac{\int f_{PS}(f) \cdot f \cdot df}{\int f_{PS}(f) \cdot df} \tag{29}$$

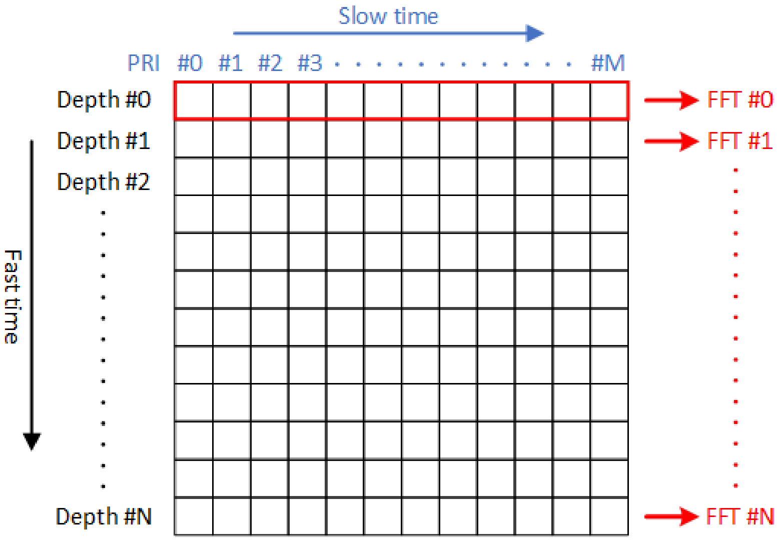


Fig. 10: Corner turning operation.

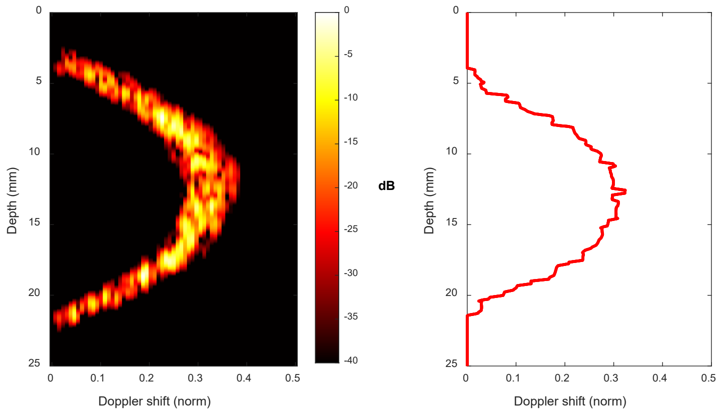


Fig. 11: Power spectral matrix (left) and velocity profile (right) of a suspension obtained by applying the UVP method.

Finally, the velocity can be obtained by applying the (25) to the measured f_{PS_m}. Repeating this process for several values f_{PS_m}, it is possible to find the flow velocity profile developed by the fluid of interest, like in Fig. 11-right.

1.4.2 Spectral Broadening

Based on the Doppler equation (25), it can be thought that the Doppler spectrum of a moving scatterer is composed by only one frequency component, i.e. a single line. However, a real Doppler spectrum has a finite bandwidth. The “broadening” of the Doppler spectrum is related to various phenomena. Following, some of these phenomena are briefly described

The “velocity broadening” is the first cause of broadening of the Doppler spectrum and it is related to the sample volume dimension. Analyzing a flow inside a pipe or a vessel, there are a lot of particles (or red blood cells) which move with different velocities according to their position in the pipe (or vessel). Consequently, there are particles with multiple velocities within the sample volume (see Fig. 12) and thus the received signal is given by several velocity contributions. This results in multiple frequency components in the Doppler signal (instead of a single tone) depending on the range of velocities within the sample volume.

Another element that contributes to the spectral broadening is the “transit time” that is the time needed to the scatterer to cross the sample volume (SV lateral dimension of Fig. 12). This time coincides with the duration of the echo signal associated to that scatterer in the slow time domain of the UVP analysis and, consequently, affects the Doppler bandwidth. The transit time was studied for the first time by Vermon L. Newhouse [4] who observed an amplitude modulation of

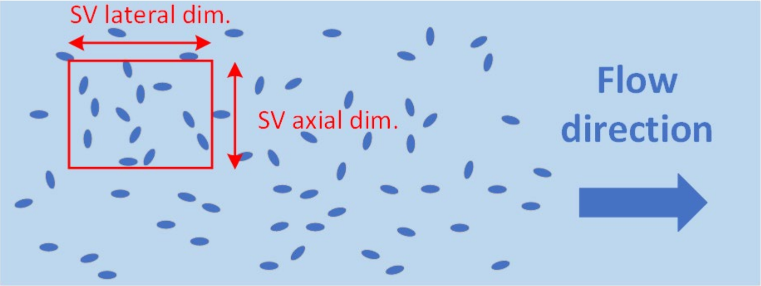


Fig. 12: Sample Volume (SV) lateral and axial dimensions.

the echo signals due to the non-uniformity of focused beams which additionally enlarges the Doppler bandwidth. Therefore, the duration of the path crossed by a scatter and the non-uniformity of the beams it encounters determine an increase in the Doppler bandwidth.

The broadening of the Doppler spectrum was subsequently interpreted as "Geometrical broadening" still by Newhouse [5] and later the equivalence between geometrical and transit time broadening was demonstrated [6]. In focused beams, the Doppler angle between the ultrasound and the target direction is not unique, but is given by a range of angles over which ultrasound is backscattered to the transducer. Therefore, the focused beam can be seen as composed by multiple rays which propagate with different Doppler angles and thus different Doppler shift, causing the widening of the bandwidth.

In literature the enlargement of the Doppler spectrum due to transit time and geometrical effects is referred as "Intrinsic Spectral Broadening" (ISB) [7] because it is associated to the intrinsic way to deliver and receive the acoustic energy.

Another source of broadening is the presence of acceleration within the sample volume (non-stationary movement) which is present, for example, in systolic acceleration of the blood flow.

Chapter 2. Applications of the UVP method

In this chapter both industrial and biomedical applications of the UVP method are briefly presented. The first part describes the rheology basics, some of the main models used to describe Newtonian and non-Newtonian fluids, and the pipe flow behavior. As described in the next chapter, these models are fundamental for the velocity profile emulation. The second part focuses on the applications of UVP method, presenting an industrial system for rheological parameter assessment and an echographic system which performs Doppler analysis to evaluate the blood velocity peak and the velocity profile features. Both the industrial and medical systems were designed in the MSDLab laboratory and were used to perform some of the tests described in the next chapters.

2.1 Rheology Basics

The rheological characterization of fluids is of paramount importance in several industries, like food, chemical, pharmaceutical, building industries and many others [8]-[10]. The assessment of the fluids proprieties makes it possible to optimize and monitor the production process and guarantee the final product quality. In medicine, the "hemorheology", i.e. the rheology of the blood, studies the flow properties of the blood which mostly depends on red blood cells, plasma and hematocrit. An alteration of the rheological properties of the blood can indicate blood or vascular diseases. Moreover, arterial wall shear stress studies lets to evaluate arterial endothelial dysfunctions related to artery constriction or thrombosis [11].

The "Rheology" is the science of deformation and flow of matter [12]. It studies the viscous behavior of fluids, that is shown in a plot called "rheogram". Before seeing a rheogram, it is necessary to introduce some concepts.

First of all, there are three different flow regimes, namely laminar, turbulent and transitional flow. In laminar flow, the motion of the particles is very orderly, and they move parallel to the pipe walls. By contrast, in turbulent flow, the particles move in random directions, both parallel and transverse to the direction of the main flow. When particles flow changes from laminar to turbulent, a transitional flow develops very fast in the region that separates the laminar and turbulent regions (called "transition zone"). The "Reynolds number" lets to know which flow regime will be developed by a fluid, and is expressed as:

$$Re = \frac{\rho \cdot v \cdot L}{\eta} \tag{30}$$

where ρ is the fluid density, v the flow velocity, L the pipe length and η the viscosity. A laminar flow occurs when the Reynolds number is low (< 2000), while a turbulent flow occurs with high numbers (> 4000) [12]. In other words, laminar flow will develop below a particular velocity v_1, while turbulent flow above a velocity v_2. Between these velocities, a transitional flow develops in the unstable transition zone.

To understand the behavior of a fluid and the parameters used in rheology for its description, the "Two-plates model" can be used (Fig. 13). There are two parallel plates and a thin layer of fluid between them. The lower plate is fixed. The flow is laminar, so no vortices occur. Flow can be depicted as layers of fluid that slide one over other. Applying a force F to the upper plate, a stress τ, called "shear stress" is applied to the fluid, parallel to the plate surface, and be expressed as:

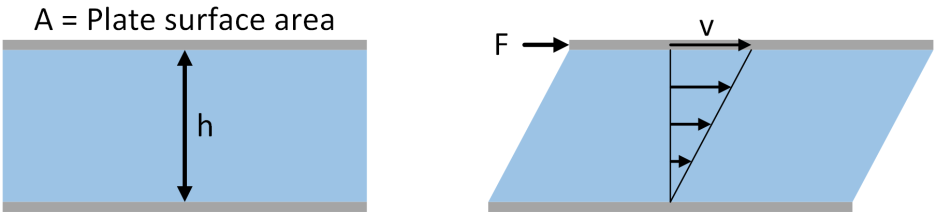


Fig. 13: Two-plates model.

$$\tau = \frac{F}{A} \tag{31}$$

where A is the plate surface area.

Therefore, the layer of fluid closest to the upper plate moves faster than the others beneath it, and the velocity of the layers decreases moving towards the stationary plate. The velocity gradient in the direction perpendicular to that of the shear stress is called "shear rate" $\dot{\gamma}$, and it can be expressed as ratio between the velocity v of the upper plate and the distance h between the plates:

$$\dot{\gamma} = \frac{v}{h} \tag{32}$$

The relationship between shear stress τ and shear rate $\dot{\gamma}$ lets to find some characteristic of the suspension of interest and is reported in the rheogram. In particular, for Newtonian fluids, like water, milk, honey, etc., the relationship between shear stress and shear rate is linear:

$$\tau = \eta \cdot \dot{\gamma} \tag{33}$$

where η is the viscosity, an index of the internal resistance of a fluid to being deformed.

The rheogram of a Newtonian fluid, as shown in Fig. 14-left, is a straight line of slope η_N that pass through the origin of the axes. Thus, the viscosity depends only on the material, temperature and pressure, and it is constant as the shear rate changes, as shown in the plot η-$\dot{\gamma}$ in Fig. 14-right. It is important to keep in mind the dependence of the viscosity on temperature and pressure, that typically increases with increasing pressure and decreasing temperature.

However, almost 95% of all fluids employed in industries [13] and also the blood are non-Newtonian. In these fluids, the relationship between shear stress and shear rate is non-linear. Therefore, the viscosity is not constant but depends on the shear

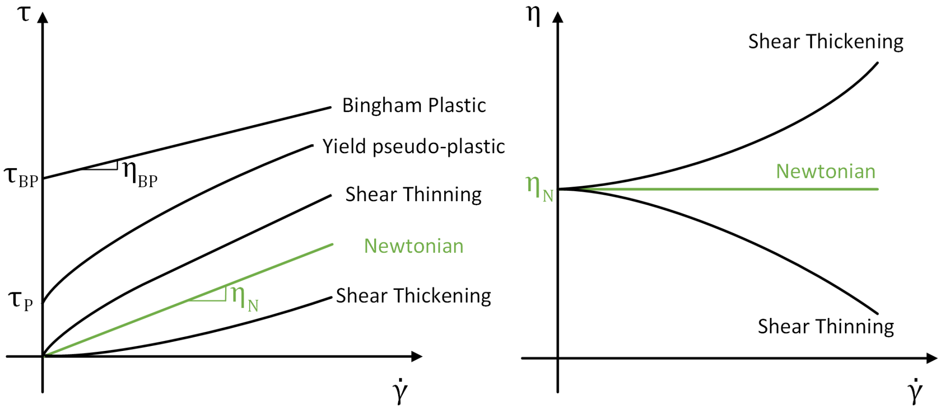


Fig. 14: Rheogram (left) and viscosity varying of shear rate (right).

rate and it can also depend on time. The rheogram for these fluids is a curve that does not necessarily pass through the origin of the axes (black curves in Fig. 14).

The most common non-Newtonian fluids are "shear thinning" (or "pseudo-plastic"). They are characterized by a viscosity which decreases with increasing shear rate. At very low shear rates, shear thinning fluids shows a constant viscosity, like a Newtonian fluid, that is called "zero shear viscosity" η_0. At a critical shear rate value, the viscosity suddenly drops and the shear thinning region starts. Then, if high values of shear rate are reached, the viscosity will be constant again and equal to η_∞, called "infinite shear viscosity", that can be several orders of magnitude lower than η_0. This trend is clearly depicted in a double logarithmic plot η-$\dot{\gamma}$, shown in Fig. 15. Some highly shear thinning fluids, called "Yield pseudo-plastic" in Fig. 14, are fluids with a "yield stress". The yield stress is the finite stress that should be applied before fluids begin to flow. Below this value,

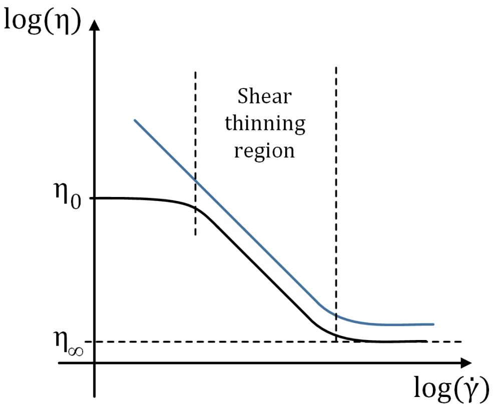


Fig. 15: Shear thinning fluids with (blue line) and without (black line) yield stress.

the viscosity becomes very high and therefore the fluid behaves as a solid (see Fig. 15). For instance, the shear stress τ must be greater than τ_P or τ_{BP} (Fig. 14) so that Yield pseudo-plastic or Bingham Plastic type fluids can flow.

The blood is also a shear-thinning fluid: in fact, during the systolic peaks, the increasing flow leads to high shear rates and the blood becomes less viscous. Conversely, the viscosity increases with low flow and large blood vessel, like aorta, where the non-Newtonian behavior is not very significant and the blood can be assumed as Newtonian fluid.

Some materials show the opposite behavior to shear thinning, that means an increase of viscosity with increasing shear rate. They are called “shear thickening” fluids (or “dilatant” fluids) and are less common than shear thinning ones. However, the characteristic of becoming thicker following the application of a force is useful in the realization of shock absorber or impact protection equipment.

The viscosity of the fluids analyzed above is “time independent”, that means the recovery time needed by the viscosity to reach the original value (when the shearing force is removed) is negligible. If that doesn’t happen, the viscosity is “time dependent”. A shear thinning fluid with a time dependent viscosity is called “Thixotropic”. A typical example is the paint, that is thick (high viscosity) in the can when stored for a long time and thin (low viscosity) when we steer it, but it takes time to became thick again. In contrast, a “Rheopectic” fluid is a time dependent shear thickening fluid. Some examples of rheopectic fluids are Gypsum paste and some lubricants that became thick (or solidify) when shaken.

In this thesis, only time independent fluids are considered. Therefore, hereafter it is implicitly assumed that the fluids are time independent.

2.1.1 Models for non-Newtonian fluids

In literature there are several models for describing the behavior of a non-Newtonian fluid [12], [14]-[16]. The applicability of each model basically depends on the required range of shear rates.

The simplest model that approximates the behavior of a non-Newtonian fluid is the “Power-Law” model. The latter mathematically describes the relationship between shear stress and shear rate as follow:

$$\tau = K \cdot \dot{\gamma}^n \tag{34}$$

where K and n are respectively the “Power-Law consistency index” and “Power-Law exponent”, two empirical curve-fitting parameters.

Starting from this relation, the viscosity can be expressed as:

$$\eta = K \cdot \dot{\gamma}^{n-1} \qquad (35)$$

Depending on the value of n, the Power-Law model can describe the two types of non-Newtonian fluid behaviors seen in the previous paragraph:

- Shear-thinning (or pseudo-plastic) $n < 1$
- Newtonian $n = 1$
- Shear-thickening (or dilatant) $n > 1$

The limitation of this model is that it is valid only in a limited range of shear rates, therefore the values of K and n are dependent on the range of shear rates taken into account [17]. However, this model is very simple and easy to use and particularly effective for modelling shear thinning fluids, which are the most common type of fluids. Therefore, despite its limitations, it is widely used in process engineering applications that involve low-viscosity fluids like weak gels, resins, motor oils, some food (e.g. ketchup, orange juice concentrate, whipped cream, etc.) and for modelling blood behavior. Given the wide use of the Power-Law model, its simplicity and the type of fluids that can be modeled, this model was used in this thesis to emulate shear thinning fluids, as described in the Chapter 4.

The Power-Law model describes fluids without yield stress, i.e. fluids whose rheogram passes through the origin of the axes (Fig. 14). The simplest model that describes the flow behavior of a non-Newtonian fluid with a yield stress is the "Bingham Plastic" model. The relationship between shear stress and shear rate stated by the Bingham Plastic model is the follow:

$$\tau = \tau_Y + \eta_{BP} \cdot \dot{\gamma} \qquad (36)$$

where τ_Y is the yield stress and η_{BP} the plastic viscosity, that is the slope of the rheogram line (Fig. 14). Thus, the Bingham Plastic model is a linear equation and the yield stress represents the intercept of the rheogram with the shear stress axis.

Further generalizing, that means a non-linear rheogram with a yield stress, the "Herschel-Bulkley" model is obtained:

$$\tau = \tau_Y + K \cdot \dot{\gamma}^{n} \qquad (37)$$

This model is basically a Power-Law model with yield stress, used to characterize pseudo-plastic material that have a yield stress rather than 0.

The models just described are very simple and depend only by two parameters. The benefit of this approach is that it is possible to describe the rheogram with relatively few fitting parameters and predict the behavior at unmeasured shear rates. There are more complex models that uses more parameters, like the "Cross model" [15], that covers the entire shear rate range, but they are outside the scope of this thesis.

2.1.2 Newtonian and non-Newtonian pipe flows

Generally, the shape of the velocity profile developed in a pipe provides important information on the characteristics of the fluid, since it is strongly influenced by the rheological properties of the fluid. In this paragraph, the fundamentals of Newtonian and non-Newtonian pipe flow are presented [12][14].

Consider a Newtonian fluid that flows in a cylindrical pipe of radius R in laminar and steady flow conditions, as shown in Fig. 16. The force balance on a fluid element situated at distance r from the center of the pipe can be expressed as:

$$p \cdot (\pi r^2) - (p + \Delta p) \cdot (\pi r^2) = 2\pi r L \cdot \tau \tag{38}$$

Therefore the shear stress τ is:

$$\tau(r) = -\frac{\Delta p \cdot r}{2L} \tag{39}$$

Note that no assumptions have been made on the type of fluid, thus the above equations are applicable to any fluid.

Analyzing the equation (39), the shear rate is zero in the pipe center and maximum at the pipe wall τ_w (see Fig. 17):

$$\tau_w = \tau(R) = \frac{\Delta p \cdot R}{2L} = \frac{\Delta p \cdot D}{4L} \tag{40}$$

where D is the pipe diameter.

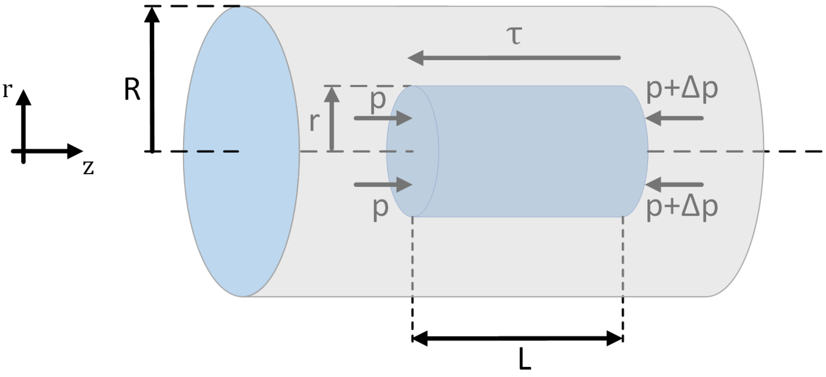


Fig. 16: Flow through a cylindrical pipe.

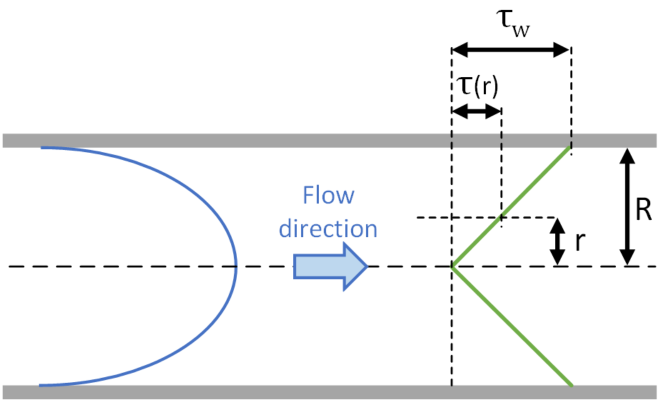


Fig. 17: Velocity (blue curve) and shear stress (green curve) distributions of a Newtonian fluid that flows in a pipe of radius R.

On the other hand, the velocity is maximum at the pipe center and minimum at the pipe wall, and has a parabolic shape.

Consider now a non-Newtonian fluid flowing in a pipe of radius R. Applying the Power-Law model seen in the previous paragraph, the equations (39) and (40) and assuming zero velocity at the pipe wall, it is possible to obtain the radial velocity, shear rate and the viscosity profiles:

$$v(r) = \frac{nR}{1+n}\left(\frac{R \cdot \Delta p}{2LK}\right)^{1/n}\left(1-\left(\frac{r}{R}\right)^{1+\frac{1}{n}}\right) \tag{41}$$

$$\dot{\gamma}(r) = \left(\frac{r \cdot \Delta p}{2LK}\right)^{1/n} \tag{42}$$

$$\eta(r) = \frac{\tau(r)}{\dot{\gamma}(r)} = K\left(\frac{r \cdot \Delta p}{2LK}\right)^{1-\frac{1}{n}} \tag{43}$$

where n and K are the Power-Law indices. Moreover, the maximum value of shear rate is that obtained at the pipe walls and is:

$$\dot{\gamma}_w = \dot{\gamma}(R) = \left(\frac{R \cdot \Delta p}{2LK}\right)^{1/n} \tag{44}$$

The equation (41) lets to find also the volumetric flow rate Q as:

$$Q = 2\pi\int_0^R vrdr = \frac{\pi n R^3}{3n+1}\left(\frac{R \cdot \Delta p}{2LK}\right)^{1/n} \tag{45}$$

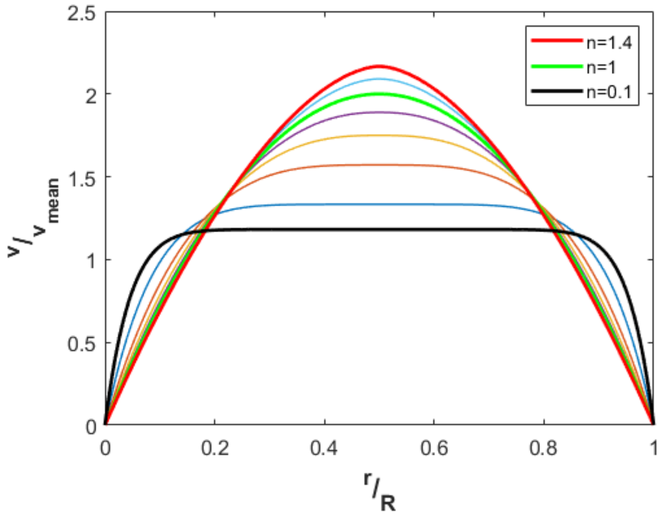


Fig. 18: Laminar velocity profiles of Power-Law fluids as n changes.

Fig. 18 shows laminar velocity profiles of Power-Law fluids with different values of n. For shear thinning fluids, i.e. $n < 1$, the lower the value of n, the flatter the velocity profile. By contrast for shear thickening fluids, i.e. $n > 1$, the profiles become sharper as n increases. Finally, as anticipated, the profile for Newtonian fluids ($n = 1$) has a parabolic shape.

As previously stated, some shear thinning fluids with yield stress behave like a solid at very low shear rates. This kind of fluids develop a very flattened profile, resulting in a “plug flow” like the one shown in Fig. 19. The yield stress value τ_Y is linked to a parameter called “plug radius” R^*, measured from the velocity profile, by the following equation:

$$R^* = \frac{2L \cdot \tau_Y}{\Delta p} \qquad (46)$$

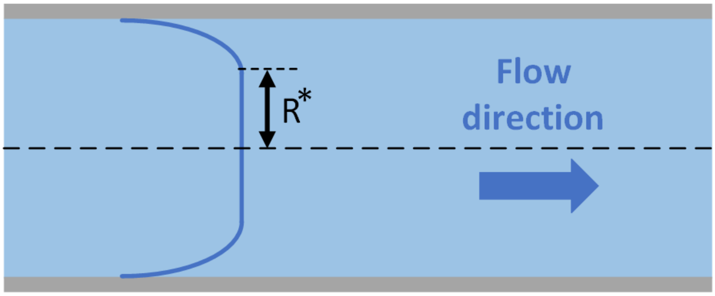


Fig. 19: Example of a plug flow and corresponding plug radius R^*.

2.2 Industrial rheological parameters assessment

The classical methods for the assessment of rheological parameters of a fluid require collecting fluid specimen at different points of the production chain and move them to laboratory where they are analyzed by viscometers or rheometers [12][17][18]. However, collecting and moving samples to laboratory is not always possible. Indeed, in some cases, the rheological parameters of interest changes quickly or depend on the configuration setup of the industrial plant. In other cases, like for example in food or pharmaceutical industries, it is not possible to collect samples for hygiene reasons and for avoiding the contamination of the product. Moreover, the classical approaches require several tests to properly characterize a fluid. Therefore, the classical methods are very time consuming and impractical for industrial process control that requires a continuous monitoring of the parameters of interest. An efficient method used for automatic, non-invasive and in-line assessment of the rheological properties of fluids is the Pulsed Ultrasound Velocimetry and Pressure Drop (PUV-PD) method [20]-[25].

The PUV-PD method is a multi-gate spectral Doppler technique that lets to characterize a wide range of non-Newtonian and opaque fluids measuring the velocity profile of the fluid under test through the UVP method (1.4.1). By combining the velocity information with the pressure drop measurement, it is possible to assess the rheological proprieties like viscosity. As previously seen, the velocity distribution across the pipe diameter depends on the rheological properties of the fluid. In Fig. 20, the three most common profiles are shown: a parabolic profile, typical of viscosity independent shear rate fluids (i.e. Newtonian fluids); a flattened profile of a shear thinning fluid and a plug flow, typical of fluids with yield stress. The flow profiles obtained with the PUV method are used to evaluate the shear rate $\dot{\gamma}$ along the radius r of the pipe, that is the derivative of the flow velocity v:

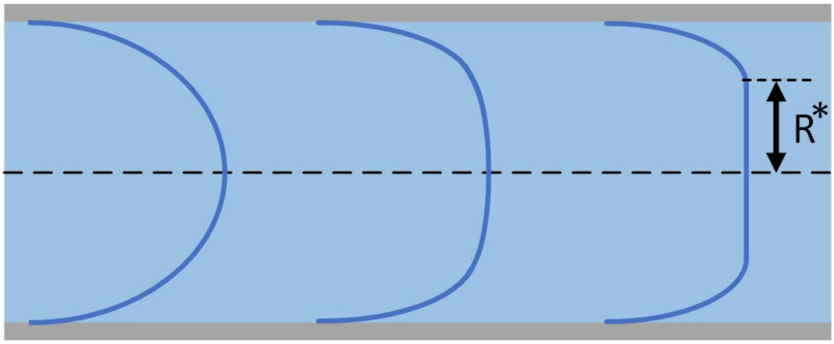


Fig. 20: Three common flow velocity profiles in industries: a parabolic flow of a Newtonian fluid (left), a flattened flow of a shear thinning fluid (middle) and a plug flow of a yield fluid (right).

$$\dot{\gamma}(\mathrm{r}) = -\frac{dv}{dr} \tag{47}$$

Instead, the pressure drop measurement permits to find the shear stress $\tau(r)$ through the equation (39). By combining the flow velocity profile, obtained through ultrasound, and the pressure drop measurement (PUV-PD method), it is possible to calculate the shear dependent viscosity η in real-time as:

$$\eta(r) = \frac{\tau(r)}{\dot{\gamma}(r)} \tag{48}$$

Note that the classical methods (viscometers and rheometers) determine the viscosity at a single shear rate. This is fine for Newtonian fluid, whose viscosity is independent of the shear rate, but for non-Newtonian fluids it is necessary to evaluate the viscosity in several shear rate values. The PUV-PD method, instead, lets to derive the shear dependent viscosity very quickly in real-time and lets to continuous monitoring of the fluid of interest, that is not possible with classical approaches.

2.2.1 V3 system

The V3 system [26]-[28] is an embedded system for in-line fluids characterization, specifically designed for the in-line velocity profile measurement and rheological assessment of opaque, non-Newtonian industrial fluids. The system consists of an operator's panel, a custom sensor unit [29][30] and a software composed by a GUI software interface for setting parameters, controlling the operations of the system and signal-processing. The electronics that implements the UVP method was designed in the MSDLab laboratory of the University of Florence (Italy), and it is briefly described below.

The V3 system consists of three boards: an analog front-end (AFE), a digital board and a commercial ethernet network connection board used to control the system and download the data to the PC. The AFE embeds all the electronics necessary for conditioning the ultrasound signals of the two channels, it includes power amplifiers for the transmission, low noise and programmable gain amplifiers for the reception and a T-R switch. The AFE can generate bursts with an amplitude up to 80 Vpp with a bandwidth between 0.8 and 7 MHz and an overall gain from 7 to 55 dB. The digital board is based on an FPGA of Altera-Intel that generates the samples of the transmission burst through a Direct Digital Synthesizer (DDS) implemented directly on the FPGA, handles the digital-to-analog (DA) and analog-to-digital (AD) converters and elaborates the acquired data. The elaboration performed on the acquired data is that described in

1.4.1: the samples are coherently demodulated, filtered and stored in an on-board SDRAM. When enough data are stored, the spectral analysis starts by the corner turning operation and FFTs (Fast Fourier Transform) and then the power spectrum is obtained by summing the square of the FFT output. Moreover, it is possible to average a programmable number of spectral matrices for improving the signal-to-noise ratio before the frequency profile is extracted. Finally, a weighted mean is performed on the power spectrum and the frequency profile is moved to the PC where it is converted to a velocity profile and the rheological parameters are extracted.

In Fig. 21 the spectral matrix is reported together with the velocity profile obtained by the V3 system when analyzing a non-Newtonian industrial fluid with a particle concentration of 26%, that was flowing in a stainless steel pipe with an inner diameter of 48.6 mm and flow rate of 360 L/min [28]. The corresponding rheogram is shown in Fig. 22, where the shear rate and shear stress values were obtained directly from the measured data. As shown in the rheogram, the tested fluid was a yield fluid and the yield stress, obtained directly from the measured plug radius, was 8 Pa. This measure is in excellent agreement with the yield stress obtained with a viscometer, equal to 8.1 Pa. Generally, the rheogram obtained by the V3 are very close to the one obtained by rotational rheometers, within 5% range [31].

Therefore, the system lets to accurately extract the rheogram of fluids of interest directly in-line and in a non-invasive way.

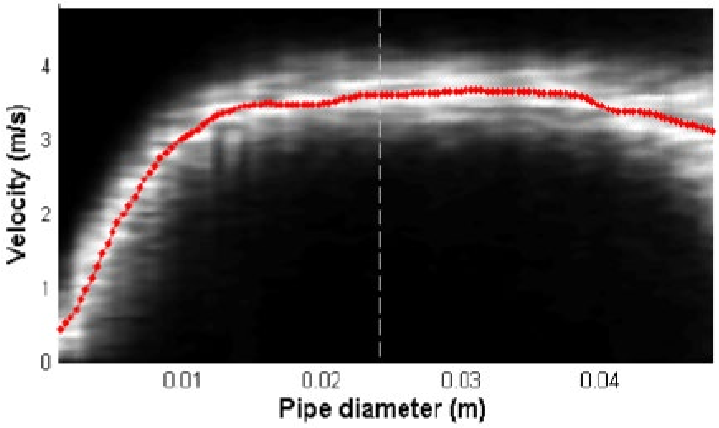


Fig. 21: Spectral matrix and velocity profile (overlapped in red) obtained from an industrial fluid, measured at a volumetric flow rate of 360 L/min in a pipe with an inner diameter of 48.6 mm.

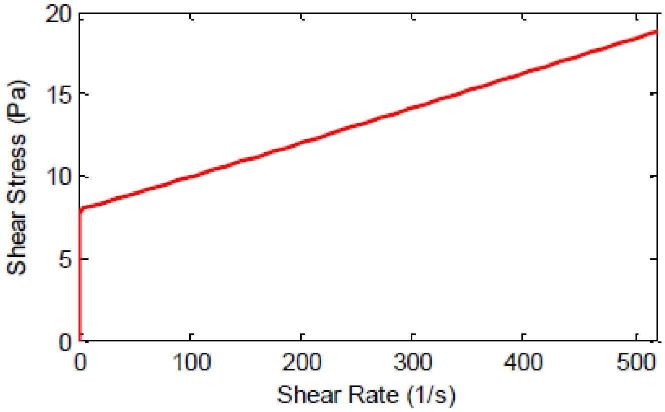


Fig. 22: Rheogram measured in-line by the V3 system. The yield stress obtained from the measured plug radius was equal to 8 Pa.

2.3 Biomedical applications

The analysis of the velocity profile properties is of paramount importance also in biomedical applications. The peak blood velocity, for instance, permits measurements of several indices of high clinical interest, including the investigation of carotid stenosis and thrombus that determines the reduction of the vessel lumen. The reduction of the carotid lumen reduces the blood flow toward the brain and increases the risk of ictus. Therefore, a prompt diagnosis is important to reduce the risk of complication that could be lethal.

The UVP method (1.4.1) represents the basic technique for several medical application. For example, it lets to estimate the blood velocity peak with high accuracy, as in [2], or the blood velocity distribution that is also used for the assessment of the wall shear rate, a parameter related to the plaque formation and vasodilatation/vasoconstriction diseases. Moreover, combining the velocity distribution information and the measurement of the vessel diameter, it is possible to perform blood volume flow measurements [32], which are used in various medical techniques as anesthesia, hemodialysis, etc.

2.3.1 ULA-OP

The ULtrasound Advanced-Open Platform (ULA-OP) is a research echograph completely developed at the MSDLab of the University of Florence, Italy. It is a system specifically designed for scientific research for the development of novel investigation methods and the study of new efficient signal generation modalities. For these reasons, the ULA-OP was designed as re-programmable system with a modular structure that lets to expand the system capability by changing properly designed boards. Two versions of ULA-OP have been developed over the years: a 64 channels (ULA-OP [33]) and then a 256 channels (ULA-OP 256 [34]) system. In this work, the 64 channel version was used in some tests, and thus it is briefly described in the following.

The ULA-OP is composed by two main boards: an analog board with the front-end and a digital board for the excitation signals generation and the real-time processing. A back-plane board connects the main boards together, delivers the power and hosts the probe connector.

The system manages 64 independent TX-RX channels, which are connected to a 192-element probe connector through a programmable switch matrix. The transmission signals are generated by 64 independent arbitrary waveform generators while the echo signals are amplified by a low noise amplifier and sampled at 50 Msps, 12 bit.

The transmission and receiving are performed by four Front-End Boards (see Fig. 23), each of them controls 16 channels generating the excitation signals during the transmission phase and acquiring and dynamically beamforming the echo signals during the receiving phase. Another FPGA ("Master FPGA" in Fig. 23) partially processes the beamformed data which are mainly elaborated by a Digital-Signal-Processor (DSP). The latter implements real-time processing algorithms depending on the specific applications and also manages the system operation.

The communication with the host PC is performed through a USB 2.0 controller, connected to the Master FPGA. The PC runs a custom software with a Graphical

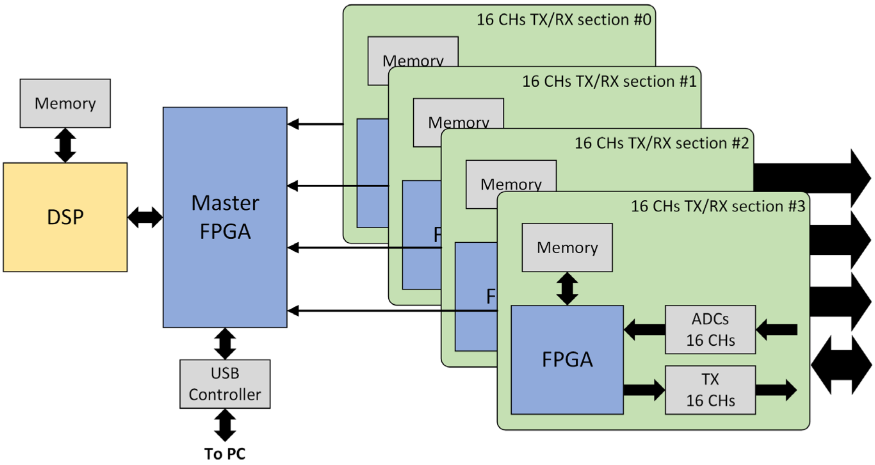


Fig. 23: ULA-OP digital board architecture.

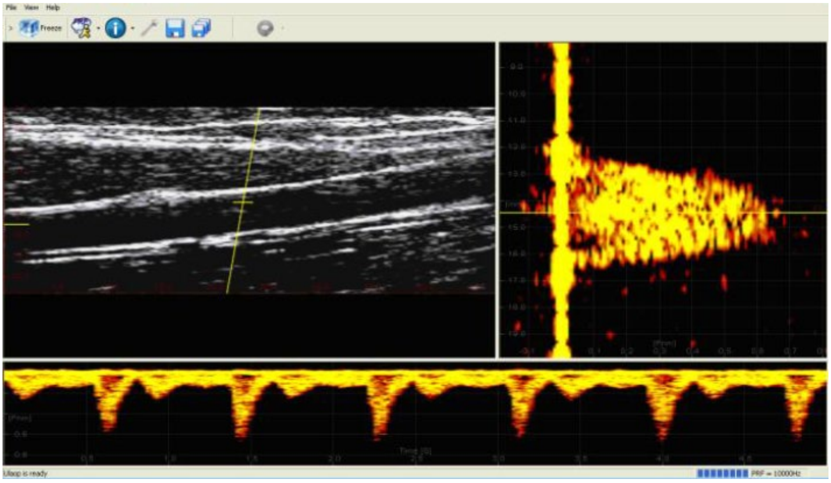


Fig. 24: Example of ULA-OP GUI in MSD mode. The top panels reports the B-Mode image (left) and the spectral profile (right), while the bottom panel shows the sonogram.

User Interface (GUI) that displays the real-time processing and the operating parameters in different panels.

Among the various operating modes, the Multi-gate Spectral Doppler (MSD) mode sets ULA-OP to perform a spectral analysis through the UVP method. As shown in Fig. 24, the GUI reports three panels: the top-left panel shows the B-Mode image, where the yellow line indicates the direction selected by the operator for the spectral analysis, shown in the top-right panel. The bottom panel reports the sonogram related to the depth selected by the yellow line in the spectral profile panel, showing how the velocity (at that depth) changes over time. Typically, the sonogram is evaluated at the center of the vessel and the ULA-OP software locates the systolic cycles and the heart rate.

Chapter 3. Clock Synchronization Circuit

In this chapter a full digital clock synchronization method is presented, that is suitable for FPGA implementation. The synchronization circuit is able to re-phase an internal FPGA clock to every occurrence of an external asynchronous trigger. The re-phased clock can also be employed to feed an analog-to-digital (AD) or digital-to-analog (DA) conversions, reducing the frame jitter with respect to the external synchronism. Two versions of the circuit are presented, that were implemented in an Altera-Intel Cyclone III and Cyclone V SoC FPGAs. Finally, experiments are reported that show how the proposed circuit reduces the frame jitter to less than 100 ps rms. The effect of the clock re-phasing on the Doppler analysis is also shown.

3.1 Introduction

In several applications, the generation or acquisition of data sequences is triggered by an external signal. This is the case, for example, of radar interferometry [36] or Pulsed Ultrasound Velocimetry (PUV) [37], where the transmission and the receiving can be handled by separate apparatus or instrumentation. In these applications, relative target displacements and/or velocities are measured by comparing and correlating the phase of echoes received from subsequent frames. Consequently, these applications are strongly affected by a kind of jitter called "frame jitter", i.e. a random temporal variation between subsequent frames [38], that differs from the typical data jitter that applies to each single frame [39]. Indeed, uncertainty in the timing of the trigger signal leads to successive frames being randomly shifted relative to each other.

The frame jitter is produced for example when a digital system synchronizes to an external trigger by sampling the trigger with its own clock at frequency f_s. An uncertainty linearly distributed in the temporal interval $1/f_s$ is produced, which corresponds to a jitter with standard deviation of [38]:

$$\sigma_f = \frac{1}{f_s\sqrt{12}} \tag{49}$$

For example, a clock at $f_s = 100$ MHz results in $\sigma_f \approx$ 2.9 ns. This jitter is unacceptable for radar interferometry, PUV and other sensible applications. The obvious solution is distributing a single master clock in all the circuits of the system. This approach is easily applied in compact apparatuses, but it is less practical in complex systems composed by physically separated parts. This is the case, for example of electronic apparatuses employed in nuclear physics experiments [40], where the master clock is distributed through complex optical links [41]. A similar approach is used in ultrasound research [42], where 4 independent 256-channel echographs work together to manage a 1024-transducer probe. The echographs are synchronized by a single master clock distributed through HDMI cables.

Synchronization clock integrated devices are available from several semiconductor companies. However, these devices are typically based on Digital Phase Locked Loop (DPLL) techniques that cannot lock to arbitrary trigger sequences like those present in some radar or ultrasound applications [43].

In this chapter a full digital synchronization circuit, implemented in two different FPGAs, is presented that synchronizes an internal clock to an external asynchronous trigger. The synchronization occurs on every pulse issued on the trigger channel, independently from the trigger sequence. Few µs after the trigger is received, the internal clock phase is adjusted so that the frame jitter between the trigger edge and the regenerated clock is around 100 ps rms, which is suitable, for example, for PUV applications.

3.2 Synchronization Method Basics

The proposed method is based on 2 main operations, reported in the schematic of Fig. 25: a) phase measurement between the external trigger and the internal clock; b) phase shift of the clock. CLK_{Ref} is the stable local clock reference, unrelated to the clock used by an external system to produce the trigger signal *Sync*. For every pulse of the "Sync" input, the "Phase Measurement" block measures the phase difference between the Sync active edge and the next active edge of CLK_{Ref}. The detected difference, Δ_{Ph}, is used by in the "Phase Shifter" to generate CLK_{Sync}, that is a clock at the same frequency of CLK_{Ref} but phased by $-\Delta_{Ph}$. As global result, the circuit produces CLK_{Sync} that is a "copy" of CLK_{Ref} with the phase adjusted with respect to the external signal Sync. The "Control Unit" block manages the operation sequence so that the re-phasing procedure occurs on-the-fly every time the circuit receives a Sync pulse. Since several applications (e.g. PUV [44]) work with Sync pulses at several kHz rate, the time needed for each re-phasing event should be as low as possible, and, in general, represents an important quality parameter.

Following paragraphs detail the two main blocks of the proposed Synchronization Circuit.

3.2.1 Phase Measurement

The measurement of the phase difference between the Sync pulse and the internal reference clock CLK_{Ref} is performed through a Tapped-Delay-Line (TDL), which is a circuit typically employed in Time-to-Digital Converters (TDCs) [45][46] and phase measurements applications [47]. A TDL is basically a

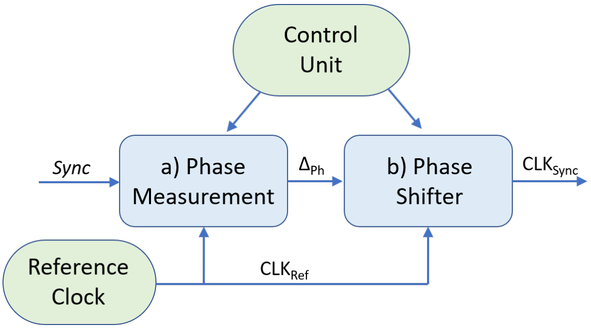


Fig. 25: Basic operations of the proposed method. The phase Δ_{Ph} between Sync edge and CLK_{Ref} is measured and used to correct CLK_{Sync}.

chain of delay elements. Its most common architecture is shown in Fig. 26: it consists of the N cells $C_0..C_N$-1, each composed by a delay element followed by the corresponding register $R_0..R_N$. The delay chain is fed by the *Sync* signal whose edge propagates inside the chain. The propagation along the delay chain is much slower than the propagation of the clock signal towards the registers, so it is assumed that every cell's register samples simultaneously at "CLK_{TDL}" rate. At each rising edge of CLK_{TDL} the registers freeze the TDL status, i.e. the updated position of the pulse edge in the chain. The position is represented by a thermometric code. Assuming that each cell features a delay t_{cell}, the phase delay measured between the Sync and the CLK_{TDL} edge is:

$$\Delta_{\mathrm{Ph}} = M \cdot t_{cell} \tag{50}$$

where M is the number of delay cells crossed as frozen in the status register. According to (50), the delay t_{cell} represents the temporal resolution of the TDL. This value depends on the implementation details. For FPGA implementation the resolution between few ps to a hundred ps are reported in literature [48][49].

Since Δ_{Ph} ranges from 0 to the t_{TDL}, i.e. the period of CLK_{TDL}, the minimum number of cells required to cover the clock period is:

$$N = \frac{t_{TDL}}{t_{cell}} \tag{51}$$

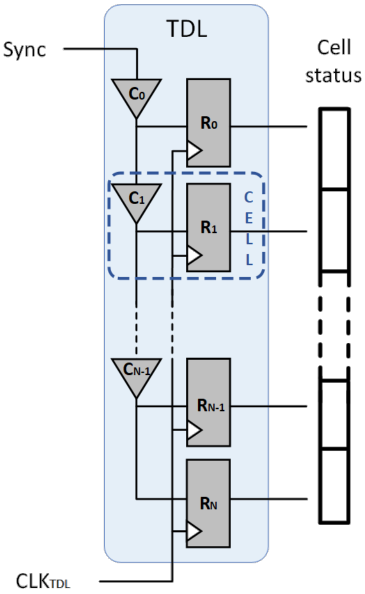


Fig. 26: Basic architecture of a TDL. The Sync 0-1 edge travels along the C_0 C_{N-1} delay cells. Its position at the CLK_{TDL} clock edge is frozen in the R_0 R_N registers.

For example, in case of CLK_{TDL} = 100 MHz and t_{cell}=50 ps, more than 200 delay cells are necessary, which makes the TDL a not-trivial structure in FPGA.

3.2.2 TDL calibration process

Several non-idealities should be taken into account in a TDL practical realization. The thermometric output code should ideally feature a single “0-1” transition; however, like for flash Analog-to-Digital converters, multiple transitions (typically named “bubbles”) can be present. Bubbles originate from the mismatch in the internal routing delays, or metastability events. To face this problem, thermometric encoders include Bubble-Error-Correction (BEC) strategies suitable to recover the right code [50] in most cases.

Non-identical delays among cells is another well-known non ideality. Moreover, cell delays depend on variable parameters like temperature, power voltage, aging, etc. For these reasons, a calibration process is often used to measure the delays of each cell at run-time [51].

One of the most effective calibration procedures is the “statistical approach” [52] based on a “Code Density Test” (CDT). In CDT the TDL input is fed by an asynchronous signal and thousands of phase measurements are saved in a memory. Since the phase difference is evenly spread in the time interval t_{TDL}, the number of measurements detected in each delay-bin, M_i, is proportional to the cell temporal delay, t_{ci}. In other words, the higher the delay of a cell, the higher the probability to freeze the transition in that cell. We have:

$$t_{ci} = \frac{M_i}{M_{tot}} t_{TDL} \tag{52}$$

where M_i is the number of hits detected for the i-th delay cell and M_{tot} the total number of occurrences. The mean delay of the TDL element can be easily obtained from (52) by:

$$t_{cell} = \frac{1}{N} \sum_{j=0}^{N-1} t_{cj} \tag{53}$$

The performance of the TDL is typically evaluated by the standard deviation of the t_{ci}, which gives a quantification of the non-uniformity of the delay chain. Other common quality indexes are the Differential Non-Linearity (DNL);

$$\text{DNL}(i) = \frac{t_{ci} - t_{ci-1}}{t_{cell}} - 1 \tag{54}$$

and the Integral Non Linearity (INL):

$$\mathrm{INL}(i) = \sum_{k=0}^{k=i-1} t_{ck} - i \cdot t_{cell} \qquad (55)$$

3.2.3 Phase Shifter

The "Phase shifter" shifts the phase of a clock signal according to the Δ_{Ph} measured by the TDL. The resolution of the shifting steps should be comparable to the TDL resolution. This is done by the means of the Phase Locked Loops (PLLs) hardware blocks available in FPGAs (see Fig. 27). They feature a Voltage Controlled Oscillator (VCO) whose Ring Oscillator [53] has multiple taps. The VCO high-frequency is thus available with different phases. A set of multiplexers followed by programmable divisors produce multiple clocks starting from a selected phase of the VCO frequency. A digital interface, accessible at run-time from the FPGA fabric, allows to dynamically modify the phase of the clocks by controlling the multiplexers and the divisors. The VCO frequency is untouched, thus the PLL lock is never lost. The Altera-Intel PLL VCO [54], for example, features 8 outputs whose phases are rotated at 45° steps. The phase resolution is:

$$t_{step} = \frac{1}{8 \cdot f_{VCO}} \qquad (56)$$

The Altera-Intel PLL block features a "Dynamic Phase Shifting" Interface (DPSI) [54], which consists of a command (increment or decrement) line and the $\mathrm{CLK}_{\mathrm{DPSI}}$ clock. It accepts a command every 5 cycles of $\mathrm{CLK}_{\mathrm{DPSI}}$, and each command moves the phase of a t_{step}. For instance, if f_{VCO} is 900 MHz, the clock phase is shifted in steps of 140 ps each command. The PLL takes 5 DPSI clock cycles to perform each shift step, corresponding to 50 ns in case of $\mathrm{CLK}_{\mathrm{DPSI}}$ = 100 MHz.

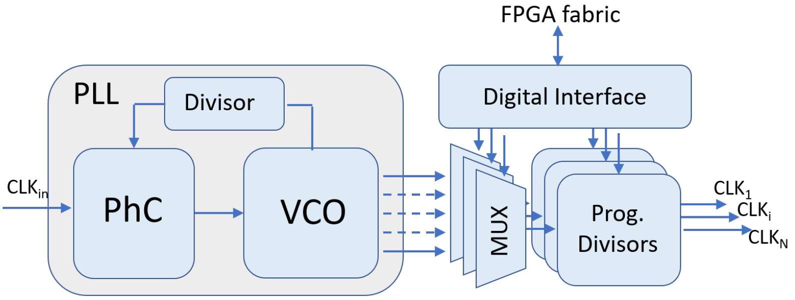


Fig. 27: The typical PLL block in FPGA includes a VCO with multiple outputs with different phase. A MUX selects one of the phases that is divided to produce a clock signal CLK_i. The clock phase can be changed through an interface that modifies the MUX/divisor settings.

The maximum number of shift steps occurs when it is necessary to shift the CLK$_{\mathrm{Sync}}$ clock of a full period. They are expressed by:

$$St_M = \frac{8 \cdot f_{VCO}}{\mathrm{CLK_{Sync}}} \tag{57}$$

3.3 FPGA Implementations

The proposed circuit, whose general architecture was shown in Fig. 25, was implemented in two different FPGAs of Altera-Intel (Santa Clara, CA USA), namely an EP3C25F256I7 Cyclone III FPGA and a 5CSXFC6C6U23C7 Cyclone V System-on-Chip (SoC) FPGA.

The architecture of the proposed Synchronization Circuit, implemented in the Cyclone III FPGA, is shown in Fig. 28. This is practically the same for the Cyclone V SoC FPGA but with some differences in the TDL, Encoder and Supervisor blocks, as described in the following paragraphs. In both cases, a single PLL block was employed to generate the CLK$_{\mathrm{TDL}}$ (corresponding to CLK$_{\mathrm{Ref}}$ of Fig. 25), and the CLK$_{\mathrm{Sync}}$. CLK$_{\mathrm{TDL}}$ is the main clock that feeds all the blocks included the TDL while CLK$_{\mathrm{Sync}}$ is the re-phased clock available at the output of the Synchronization Circuit.

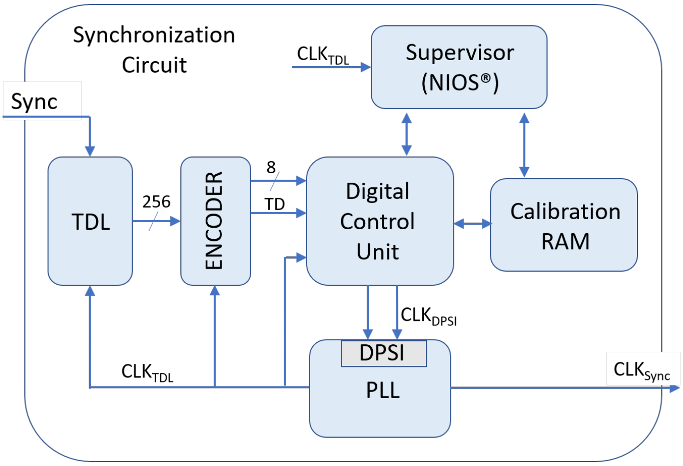


Fig. 28: Architecture of the proposed Synchronization Circuit as implemented in a Altera-Intel Cyclone III FPGA.

3.3.1 Cyclone III

Table I lists the main parameters used in the implementation, while the following paragraphs provide more details about each block.

Table I: Implementation parameters in the Cyclone III FPGA.

Parameter	Symbol	Value
	TDL	
Delay Elements	N	256
Mean Delay	t_{cell}	44 ps
Total TDL delay		11.26 ns
Calibration Meas.	M_{tot}	4096
	PLL	
VCO Frequency	f_{VCO}	600 MHz, 900 MHz
Phase resolution	t_{step}	210 ps, 140 ps
TDL clock	$\mathrm{CLK}_{\mathrm{TDL}}$	100 MHz
Re-phased clock	$\mathrm{CLK}_{\mathrm{Sync}}$	100 MHz
	Encoder	
Bubble Error Correction		First order bubble
Missing edge control		Yes
	DPSI	
Max Step number	St_M	48, 71
DPSI clock	$\mathrm{Clk}_{\mathrm{DPSI}}$	100 MHz
Time per Step		50 ns

3.3.1.1 TDL Implementation

A 256-element TDL was implemented in the Cyclone III FPGA. The basic unit of the Cyclone III is a Logic-Element (LE), composed by a 4-input Look-Up-Table (LUT) and a register (FF) as shown in Fig. 29-top. Unfortunately, logic operations (e.g. NOT, AND) implemented in the LE cells produce irregular and excessive delays that are unusable for TDL. However, when the LE is used to realize adders, it is configured in "Arithmetic Mode" (Fig. 29-bottom). In this modality each LE realizes a full adder and, most important, a fast carry-chain propagates along a dedicated path among the adjacent LEs of the device [54]. The delays of the carry-chain paths are in the order of tens of ps, and are reasonably uniform, as required by the TDL.

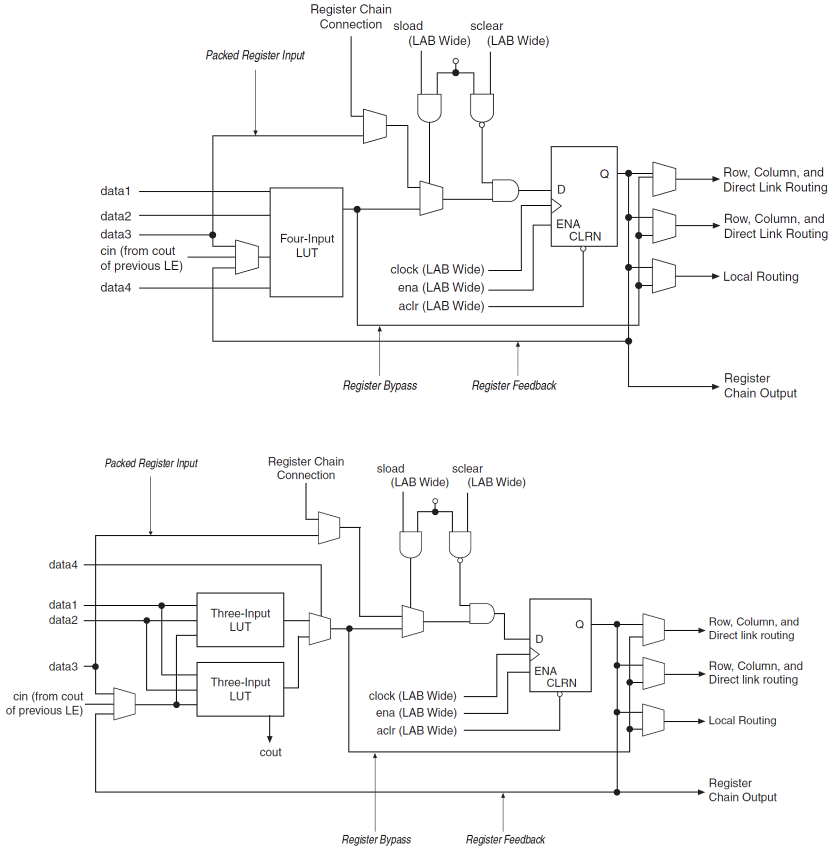
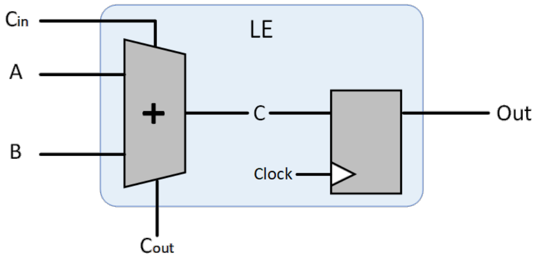


Fig. 29: Cyclone III device family LEs in Normal Mode (top) and Arithmetic Mode (bottom).

The carry-chains are typically implemented during the “Fitter” stage of the Quartus [55] software when realizing adder, counters, etc. Therefore, it is



Cin	A	B	C	Cout
0	0	1	1	0
1	0	1	0	1

Fig. 30: A 2-bit adder and register implemented in a single LE to realize a TDL cell.

necessary to force the compilation software to use these chains to take advantage of their low propagation delay. Consider a 2-bit adder with the input fixed at "0" and "1", as shown in Fig. 30. In steady-state with the carry-in input at "0", the output will be "1" and the carry-out "0". A carry-in transition 0-1 leads to an output transition 1-0 and the carry-out goes high. The 2-bit adder is physically implemented in the "Three-Input LUTs" of a single LE (see Fig. 29), together with the register that samples the output of the adder, realizing a single cell of the TDL.

Following this strategy, the 256-element TDL was implemented through a 256-bit adder with the two input words fixed at "00…00" and "11…11" respectively (see Fig. 31), which is equivalent to use 256 2-bit adders linked together through the carry signals. The Sync signal feeds the carry chain of the first adder. When Sync is at "0", all the adder outputs are at "1" as shown in Fig. 32. When Sync commutes to "1", the 0-1 edge propagates through the carry chain and changes the adder outputs from "1" to "0" in sequence. The adder output configuration is latched in the registers at the edge of CLK$_{TDL}$.

For a successful realization of the TDL and for achieving a reliable and reproducible design, it is mandatory to force the placement of the logic elements and registers exactly in the desired physical position of the device: the full-adders must be sequentially placed along the device columns, and the adder register couple must belong to the same LE to minimize the paths and improve their matching. LEs are grouped in Logic Array Block (LAB), 16 LEs for LAB. Thus,

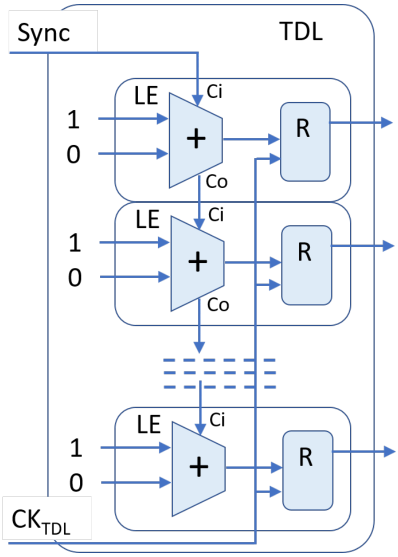


Fig. 31: TDL mapped on Altera-Intel Cyclone III Logic Elements set in Arithmetic Mode. The Sync signal feeds the carry chain that crosses all the adders.

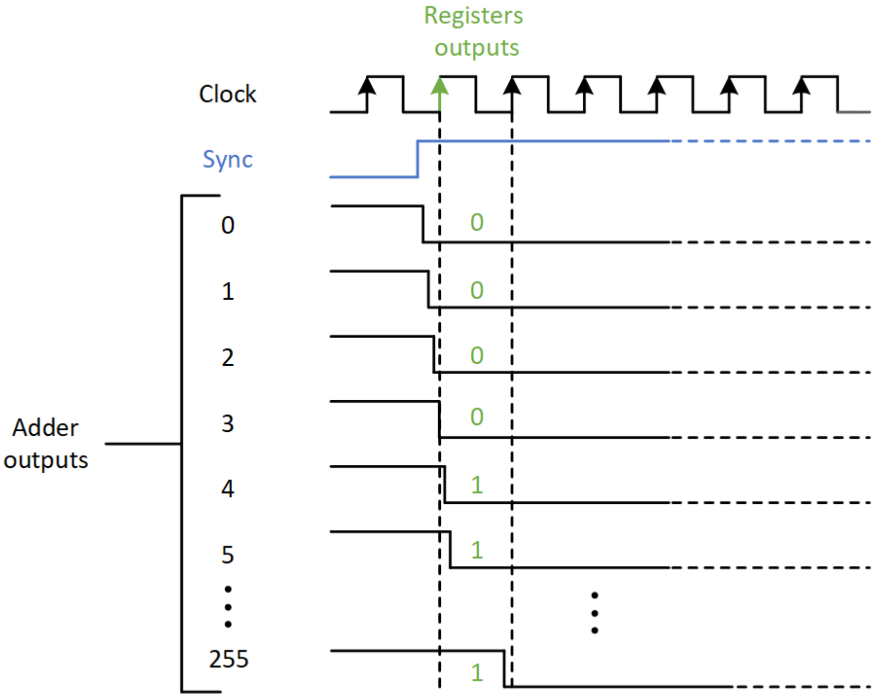


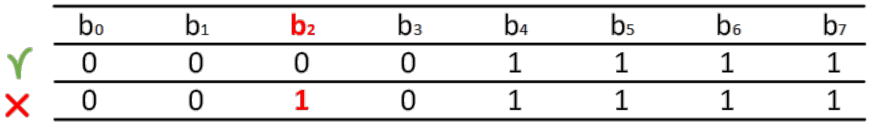
Fig. 32: Propagation of the carry signal in the adder and sampling of the registers.

for a 256 cells TDL, 16 LABs should be concatenated. This was achieved by locking LEs in column-wise structure with proper location constraints for the place and route tool of the FPGA software package [55]. This procedure grants that the Sync crosses serially the delay elements, while the clock, distributed by the global clock network, is delivered to registers with minimal skew. Possible residual skew is measured as cell delay variation in the CDT test, and compensated in the calibration process.

3.3.1.2 Encoder

The “Encoder” takes in input the 256-element thermometric code generated by the TDL and reports the position of the 1-0 transition in 8-bit binary code. A 9th bit (TD in Fig. 28) is generated to signal if the transition is detected (TD = 1) or not (TD = 0).

As briefly described in 3.2.2, there are several non-idealities in the actual realization of the TDL like “bubbles”. The latter originate mostly from metastability events and mismatch in the internal routing delays. The TDL has an intrinsically metastable structure: indeed the setup time of some registers that sample the 1-0 transitions at the outputs of the adder aren’t respected (like for



	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7
✓	0	0	0	0	1	1	1	1
✗	0	0	1	0	1	1	1	1

Fig. 33: Example of first order bubble.

example the adder's output number 3 of Fig. 32). Consequently, it is reasonable to expect errors in the registers output near the 0-1 transition (see "Registers output" in Fig. 32).

The simplest error in the thermometric code is the first order bubble, like the one shown in Fig. 33 where it is assumed that the error is located in the third bit due to metastability or routing delay mismatch, thus "1" wrongly replaces "0" at b_2. As shown, the pattern is no more thermometric and a highest "0" (or lower "1") detection wrongly recognizes b_1-b_2 as 0-1 transition instead of b_3-b_4. The implemented encoder includes a first order Bubble-Error-Correction (BEC) strategy [50][56] to solve this kind of error by locating the first occurrence of the 3-bit pattern '011', starting from the bit nearest to the Sync. Obviously, this strategy can't recognize all types of one bit error, like for example an error in the lower "1" in the thermometric code replaced by a "0" (which is impossible to identify).

In a similar way, there are bubbles of second order (and more) where a couple of bit (or more) are wrongly replaced, but this kind of error hardly occurs in a properly constrained FPGA design.

First order bubbles in the thermometric code are relatively rare in this FPGA implementation, as seen during the experiments described in the next paragraphs. However, since BEC requires few additional FPGA resources, its use seems convenient to avoid wrong phase corrections of the output clock that would have produced outliers (see Fig. 48).

3.3.1.3 Digital Control Unit

The "Digital Control Unit" block monitors the TD line produced by the encoder. When a transition is detected, it reads the 8-bit code reporting the position of the transition. This code, used as address in the calibration RAM, selects the number of phase shift steps to be applied to Clk_{TDL}. This value is applied to the PLL through the "Dynamic Phase Shifting" Interface.

Since the TDL temporal length is designed to be greater than the TDL clock period, the encoder should always recognize a transition, i.e. TD equal to 1. However, if the transition is not detected, an error bit is activated in a dedicated status register to inform the user of this situation and the phase correction is performed with a fixed and predefined value. Fortunately, this error was never detected while using the Synchronization Circuit in the presented experiments.

3.3.1.4 NIOS soft processor

The NIOS soft processor (Intellectual Property from Altera-Intel) is used as supervisor for all of the system operations, and in particular it manages the calibration procedure of the TDL. An asynchronous signal is internally routed to the TDL input and M_{tot} = 4096 phase measurements are stored in the calibration RAM. The soft processor calculates the delay of each TDL cells, t_{ci}, from (52), and the cumulative delay corresponding to the thermometric readings from the TDL:

$$t_{Th}(i) = \frac{t_{ci}}{2} + \sum_{j=0}^{i-1} t_{cj} \tag{58}$$

Then the processor fills a look-up table that, for each reading from the TDL, reports the number of phase shift steps to be applied to the PLL block:

$$PH_{St}(i) = \text{round}\left(\frac{t_{Th}(i)}{t_{step}}\right) \tag{59}$$

where round(·) is the approximation to the nearest integer. This table is written back in the calibration RAM and is used during the real-time operations.

3.3.1.5 FPGA Resources

The proposed circuit required the resources listed in Table II when implemented in an EP3C25F256I7 FPGA Altera-Intel device. The required Logic Cells are about 20% of the device total resources, however most of them are employed by the soft processor, which can be used for additional tasks as well. The time-closure was achieved for CLK_{TDL} and CLK_{Sync} at 100 MHz, with the TDL properly constrained.

Table II: Cyclone III FPGA resources utilization.

Section	**Logic cell**	**Memory bits**
TDL	256	-
Encoder	1038	-
Digital Control Unit	211	128
NIOS	3907	8192
Calibration RAM	77	2048
PLL	17	-

3.3.2 Cyclone V SoC

The 5CSXFC6C6U23C7 Cyclone V SoC device embeds an FPGA logic fabric and a dual-core ARM Cortex A9 32-bit processor. The latter is used to manage the TDL calibration process, all the Synchronization Circuit operations (replacing the NIOS II soft-processor of the Cyclone III implementation) and the communication through the ethernet interface used, for example, to download the calibration hits, curve, etc. Compared to the Cyclone III device seen before, both the structure of the basic unit of the FPGA logic, and the technology changed (28 nm [57] with respect to 65 nm [54] of the Cyclone III). For this reason, a different TDL and encoder structure are required.

Table III: Implementation parameters in the Cyclone V SoC FPGA.

Parameter	**Symbol**	**Value**
	TDL	
Delay Elements	N	400
Mean Delay	t_{cell}	24.5 ps
Total TDL delay		10.7 ns
Calibration Meas.	M_{tot}	4096
	PLL	
VCO Frequency	f_{VCO}	525 MHz
Phase resolution	t_{step}	238 ps
TDL clock	CLK_{TDL}	105 MHz
Re-phased clock	CLK_{Sync}	105 MHz
	Encoder	
Type		Zero-counter encoder
Missing edge control		Yes
	DPSI	
Max Step number	St_M	40
DPSI clock	Clk_{DPSI}	100 MHz
Time per Step		50 ns

3.3.2.1 TDL Implementation

The basic building block of the FPGA logic fabric is the "Adaptive Logic Module" (ALM) [30]. Each ALM (see Fig. 34) hosts a 8-input fracturable LUT, two adders and four registers which help to improve timing closure in register-rich designs [57]. The ALM can operate in four different modes, ensuring wide versatility. Indeed, two functions (up to 4 inputs each) or a single function (up to 8 inputs for combinational logic) can be implemented in the same ALM. As in the Cyclone III device, the Arithmetic Mode allows to use the fast carry chains that have been used as delay elements for the TDL. However, in the Cyclone V device there are physical adders and the ultra-scaled technology ensures very low carry chain delays (average value lower than 10 ps). Each LAB contains 10 ALMs and to avoid routing congestion, the LAB can support carry chains that use only the top half or bottom half of the LAB, leaving the other half of the ALM available for implementing other functions. Carry chains longer than 10 ALMs are implemented by linking together LABs vertically up to the end of the device column.

The abundance of resources in each ALM allows to implement the TDL in several ways. For instance, the "dual sampling clock method" [58] exploits the 4 registers of each ALM and 2 clocks to halve the average cell delay and consequently increase the time resolution. Instead, the "interleaved sampling method" subdivides the ALM delay using only 2 registers in each ALM exploiting the different propagation delays from the output of the adders to the corresponding

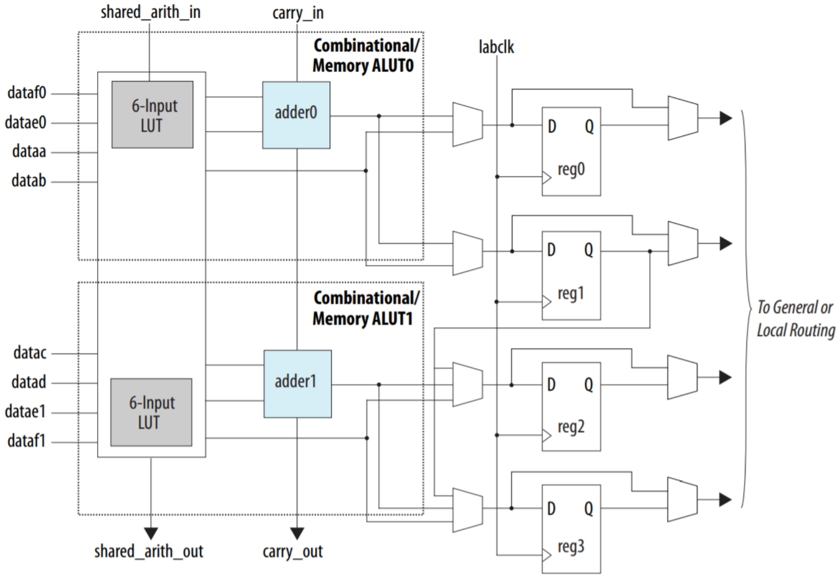


Fig. 34: Cyclone V Adaptive Logic Module (ALM) structure.

registers [59]. Moreover it is possible to combine these techniques to further subdivide the cell delays or using more TDLs in a parallel structure [60], but the implementation complexity and the resources usage consequently increase.

The implemented TDL is based on the interleaved approach and it is very similar to the one implemented in the Cyclone III device. Every cell has been implemented through a physical adder (instead of LUT adder implementation) with the input fixed at "0" and "1" and a register. Therefore, each ALM realizes two TDL cells and consequently each LAB implements 20 TDL cells.

The interleaved approach uses only two of the four register banks available in a ALM, for example reg0 and reg2 of Fig. 34, and a single clock that feeds all the registers. The uneven propagation delay from adder0 to reg0 and from adder1 to reg2 causes the registers outputs to be interleaved, reducing the average ALM delay. For this reason, the method is called "interleaved", but it is the same used in the Cyclone III implementation (note that an ALM contains two TDL cell, i.e. adder plus register, while a LE of Cyclone III FPGA implements a single TDL cell). However, the delay made by a single TDL cell (half ALM) is much smaller than the one implemented in a LE, because of the ultra-scale technology and the physical adders. The average delay measured during preliminary test is about 6.7 ps rms, that agrees with results present in literature, like in [59]. Therefore, it is necessary a chain of about 1450 cells (725 ALMs) to implement a TDL that works with a clock of 105 MHz. Actually, the implemented TDL cover an entire device column, i.e. 80 LABs. In this way, the overall delay realized by the TDL is about 10.7 ns, greater than the TDL clock period (9.5 ns).

The TDL realized in an advanced FPGA, like the Cyclone V device, can reach very high resolution, also below 4 ps rms [60][61]. These resolutions are typically

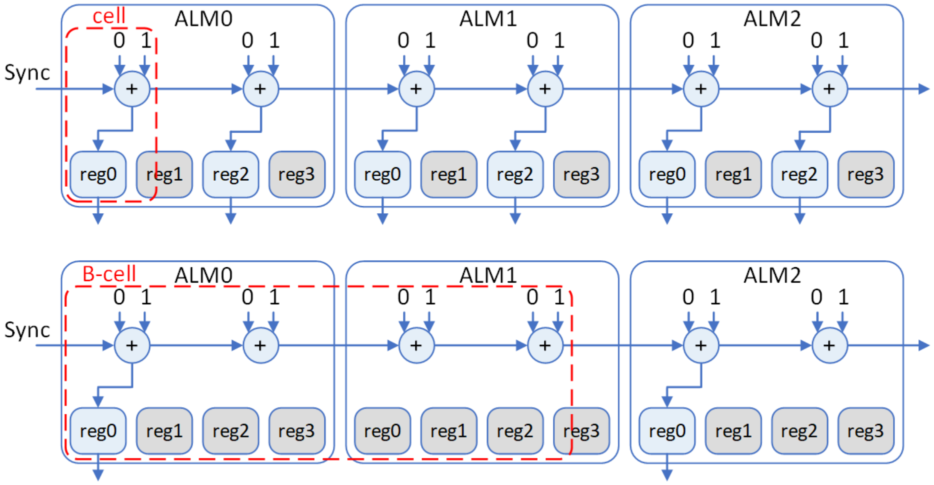


Fig. 35: TDL cell (top) and B-cell (bottom) in Cyclone V FPGA.

required in nuclear or physics experiments, but for the proposed Synchronization Circuit such high time resolution is unnecessary, and it would lead to a waste of resources. For this reason, the delay cell of the implemented TDL sums the delay of 2 ALMs, realizing a bigger TDL cell (B-cell) as shown in Fig. 35. The B-cell is equivalent to 4 TDL cell, i.e. a delay of about 4·6.7 ps = 26.8 ps rms. Therefore, the entire TDL composed by 1600 cell is “reduced” in 400 B-cells. As described in the next paragraph, this approach is useful to reduce the encoder complexity and resources.

3.3.2.2 Encoder

In ultra-scaled FPGA, the carry chain delays became smaller and the routing delays and clock skew play an increasingly important role in the delay of a cell. For these devices, the status of the Sync signal that propagates in the delay line is tapped out by the registers differently in relation to their physical location. The intuitive idea of the propagation of the Sync signal is shown in Fig. 32, where the 1-0 transitions in the adders outputs are consistent with the adder positions, i.e. for example the third output changes after the second. In this way, the code at the output of the TDL is completely thermometric, allowing an immediate decoding of the temporal information. However, this situation is difficult to obtain in devices such as Cyclone III, where few bubbles can occur. In devices realized by more advanced technology, this problem is more serious and the tap sequence at the output of the TDL is influenced by both clock skew and routing delays, making the tap sequence not consistent with the expected time order, as shown in Fig. 36.

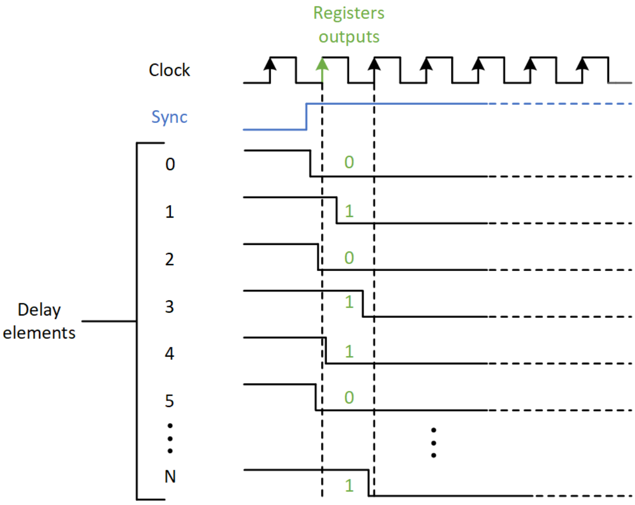


Fig. 36: Propagation of the carry signal and register sampling in an advance FPGA, where the tap sequence is not consistent with the physical position.

Moreover, the average delay of the cells becomes comparable to the clock skew, thus two adjacent cells appear as one causing a “zero width” cell [59].

The tap disorder is typically detected by the “bin realignment” procedure [58][59][61], which is only briefly described below. The goal of the bin realignment is to change the order of the TDL taps before they are sent to the thermometric-to-binary encoder. As in the code density test, a random hit signal generated with an independent clock, feeds the TDL. Every hit, the TDL output code is read and the 1s are sequentially moved to the bottom and the 0s on the top, changing the tap order accordingly. This procedure is repeated until the final tap order is consistent with the real time delay. This process requires at least 6x10^4 runs until there are almost no bubbles [58]. Moreover, it is necessary to use incremental compilation techniques to maintain the optimal tap sequence [59]. Therefore, the bin realignment is an extensive statistical test which takes time before the correct tap order is found, but it should only be performed once or after each TDL position change.

Another approach to face the tap disorder in advanced FPGAs is to use a “one-counter encoder” [60]. The operating principle of the one counter encoder can be explained by the examples of Fig. 37, where are displayed 8 delay elements and their status (registers outputs or taps) when the tap sequence is consistent with the real delay time (left figure) and not (right figure). As previously stated, a tap sequence consistent with the real time delay is easily identified by a thermometric-to-binary encoder while the register status of Fig. 37-right can’t be recognized by that encoder. However, the number of 1s (and 0s) in both registers outputs of Fig. 37 is the same. Indeed, a tap at “1” means that the Sync signal that is propagating inside the delay line is bigger than the delay of that tap.

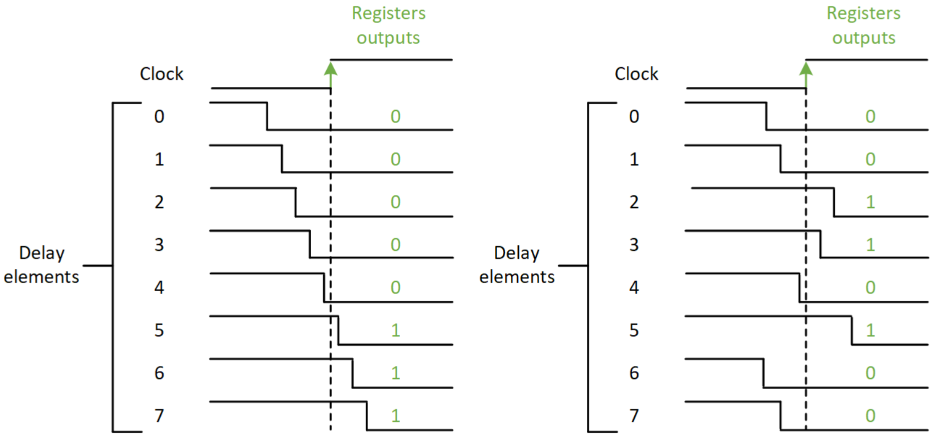


Fig. 37: Propagation of the Sync signal in 8 delay elements when the tap sequence is consistent with the real time order (left) and not (right).

Consequently, the number of taps at “1” represents the number of taps covered by the Sync signal and this is another way to identify the number of delay elements crossed by the Sync signal, independently by the tap sequence order.

The one-counter encoder is easier to implements with respect to the bin realignment and doesn’t require specific physical placement constraints and an intermediate layer between the TDL and the encoder. Obviously, this encoder is completely different from the thermometric-to-binary encoder. The main operating difference is the encoding time: the thermometric-to-binary encoder gives immediately the results (1 clock cycle), while the one-counter encoder has a pipeline structure (typically less than 10 clock cycles). Generally, this is not a limitation in TDC applications, nor in the Synchronization Circuit described in this chapter.

The one-counter encoder designed for the Cyclone V implementation of the Synchronization Circuit actually counts the 0s instead of 1s, so it is a “Zero-Counter Encoder”. The reason is simple and easier to understand in a perfect thermometric code, as in Fig. 32. In steady state, all the taps are “1” and turn serially to “0” while the Sync signal propagates in the delay line. Indeed, the thermometric-to-binary encoder identifies the 0-1 transition in the registers outputs, that is equivalent to counts the number of “0”.

The functional diagram of the implemented Zero-Counter Encoder is sketched in Fig. 38. It is a pipeline structure, whose first stage is composed by the “zero-cnt block” that encodes the 4 inputs (i.e. the outputs of the TDL) by simply counting the 0s. The following stages are full adders that sum the outputs of the previous stages. The final stage has a dynamic of 10 bits, but only the least

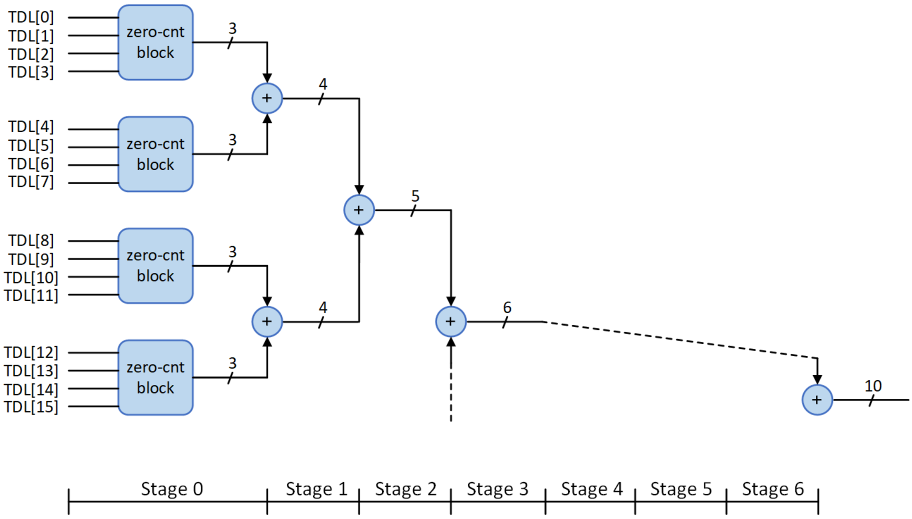


Fig. 38: Functional diagram of the Zero-Counter Encoder.

significant 9 bits are effectively used because the implemented TDL is composed by 400 B-cells. All the stages work at the TDL clock rate (105 MHz) and the latency of the encoder is 7 clock cycles, i.e. 67 ns.

3.3.2.3 Digital Control Unit and Supervisor

The Digital Control Unit and the Supervisor work similarly to the Cyclone III implementation. The main difference is that in Cyclone V device there is an ARM processor (called HPS i.e. Hard Processor System) embedded together with the FPGA, thus it is not necessary to use a soft-processor like NIOS. The HPS communicates with the FPGA side through dedicated bridges, that are used in this case to drive the Digital Control Unit, manage the calibration process and fill the calibration RAM. Moreover, the HPS handles an ethernet interface that is used to send commands (like for example to start the calibration process) and download data like the calibration hits or the content of the calibration RAM.

3.3.2.4 FPGA Resources

Table IV reports the resources required to implement the proposed Synchronization Circuit in the Altera-Intel 5CSXFC6C6U23C7 Cyclone V SoC device. The required resources are about 3% of the total ALMs available and 0.06% of the overall memory bits.

The heart of the Supervisor is actually implemented in the HPS, thus no FPGA resources are required except a bridge for properly interfacing the HPS with the FPGA blocks. In particular, this bridge requires the resources listed in Table IV as “Supervisor (FPGA side)”, where the memory bits are used for FIFOs between different clock domains. The HPS code that manages the calibration process uses part of the HPS memory, but this memory is dynamically allocated and freed after the calibration RAM is written.

Table IV: Cyclone V FPGA resources utilization.

Section	ALMs	Memory bits
TDL	800	-
Zero-counter encoder	360	
Digital Control Unit	59	170
Supervisor (FPGA side)	37	296
Calibration RAM	-	3200
PLL	20	-

3.4 TDL performance evaluation

3.4.1 Cyclone III

The TDL performance was evaluated before calibration according to the metric reported in 3.2.2. For the CDT test an asynchronous signal was internally routed to the TDL input. The code running in the NIOS soft-processor performed M_{tot}= 4096 measurements. Data were moved in Matlab (The Mathworks, Natick, MA) for elaboration. The number of hits for each delay element, reported in Fig. 39-A, ranges between 60 and 114. The delay of each cell t_{ci} was estimated with (52), and the histogram of the delays distribution is shown in Fig. 39-B. The average cell delay is t_{cell} = 44 ps, while the standard deviation is 7 ps. Data were further processed for Differential and Integral non-Linearity (DNL and INL), obtaining the curves shown in Fig. 40, and the numerical results listed in Table V. The trend of INL observed in Fig. 40 shows a correlation among subsequent delay elements that can be explained with the typical spatial variations of the planar fabrication process of the silicon die. The delay chain elements cross, in sequence,

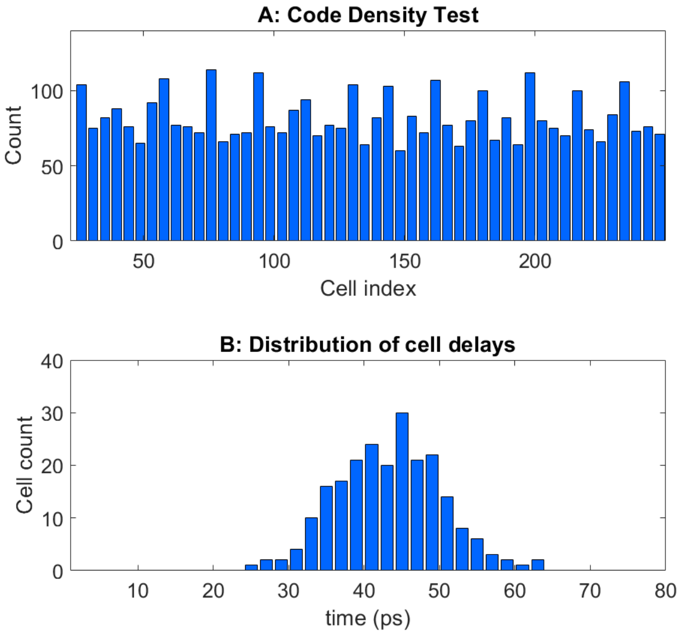


Fig. 39: Cyclone III TDL performance before calibration evaluated by Code Density Test (CDT) with 4096 measurements. A. Number of hits (Cell count) for each delay Cell (Cell index); B. Histogram of the measured delays.

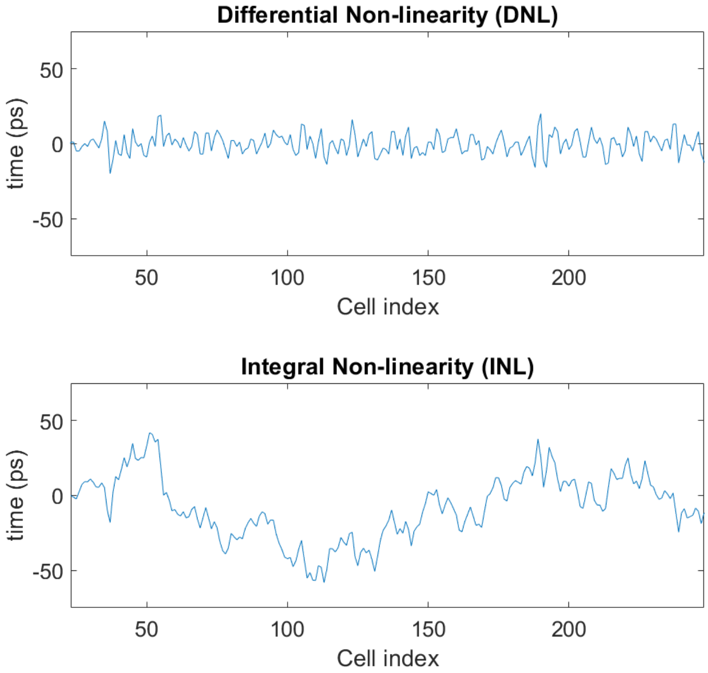


Fig. 40: Differential (top) and Integral (bottom) non Linearity for the delay cell sequence (Cell index).

different physical regions which feature slightly different delays. On the other hand, the DNL shows no correlation among elements.

The NIOS calculated the calibration table with the procedure described in 3.2.2 for compensating the mismatch in the measured delays. The calibration curve for f_{VCO}= 600 MHz is shown in Fig. 41 with the cell index in horizontal axis and corresponding shift steps in vertical axis. Note that the maximum shift steps to be applied, according to (57), is 48.

Table V: Cyclone III TDL performance.

Parameter	**Value**
Min Cell Delay	24 ps
Mean Cell Delay	44 ps
Max Cell Delay	64 ps
Standard deviation	7 ps
DNL	[-20; 20] ps
INL	[-58; 42] ps

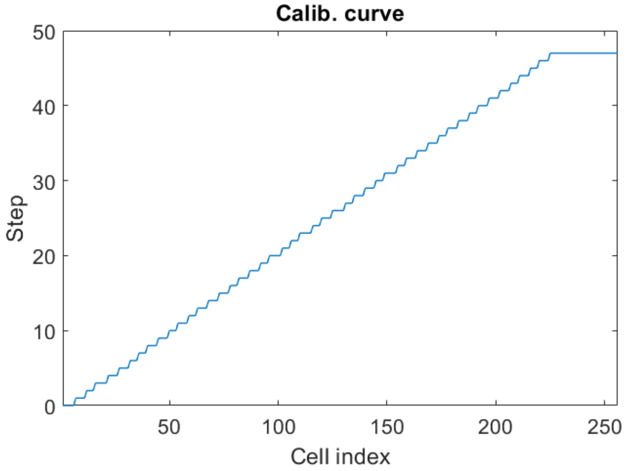


Fig. 41: Example of calibration curve obtained for f_{VCO} = 600 MHz calculated at run-time by NIOS processor and stored in the on-chip calibration RAM.

3.4.2 Cyclone V SoC

The performances of the TDL implemented in the Cyclone V FPGA were evaluated similarly to the Cyclone III device as shown in the previous paragraph.

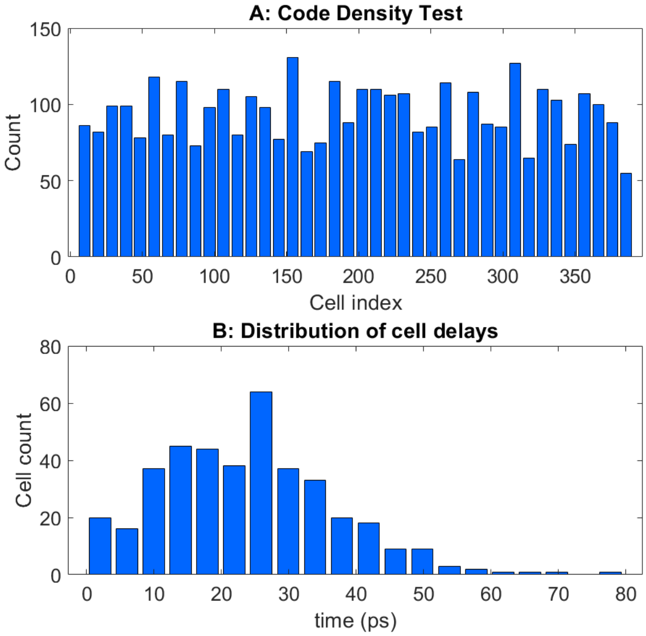


Fig. 42: Cyclone V TDL performance. A. Code Density Test (CDT) performed with 4096 measurements. B. Distribution of the cell delays.

As described in 3.3.2.1, the “base” delay cell of the implemented TDL sums the delays of 4 physical adders, obtaining an overall delay in order of about 26-27 ps. Hereafter, the groups of 4 adders, i.e. a B-cell, is referred simply as cell.

According to 3.2.2, the CDT was performed acquiring 4096 hits generated by feeding the TDL with an asynchronous signal internally routed to the TDL. The calibration process is here performed by the HPS that communicates directly with the FPGA through dedicated bridges. The number of hits detected for each delay cells is shown in Fig. 42-A, where the count range between 55 and 131. The distribution of the cell delays is reported in Fig. 42-B. The average cell delay is 24.5 ps and the standard deviation of 13 ps. Moreover, the Differential and Integral Non-linearity were evaluated obtaining the results reported in Fig. 43 and Table VI.

The HPS elaborates the acquired calibration hits to fill the calibration RAM embedded in FPGA side. Fig. 44 reports the calibration curve calculated by the HPS for the implemented 400-cells TDL and a f_{VCO} of 525 MHz, where the maximum number of PLL shift steps is 40 as expected from (57).

The performances of the Cyclone V TDL implementation are quite different from the Cyclone III case. The distribution of the cell delays has no more a

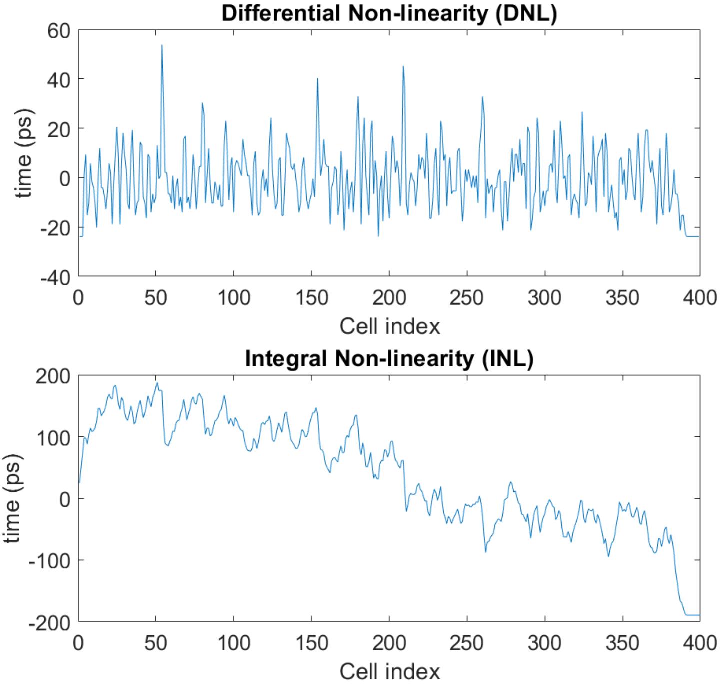


Fig. 43: DNL (top) and INL (bottom) for the TDL implemented in the Cyclone V device.

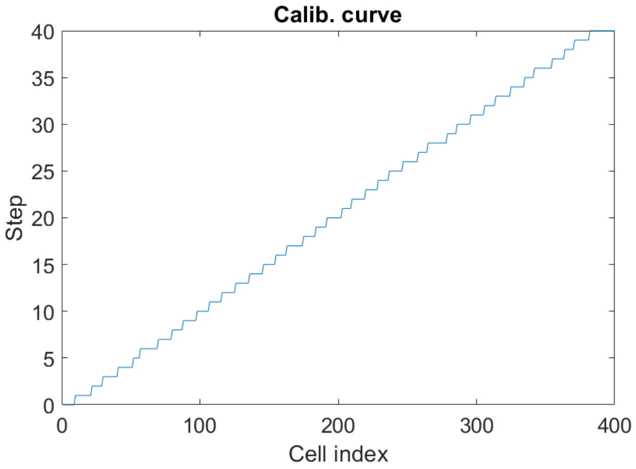


Fig. 44: Calibration curve calculated at run-time by the HPS of the Cyclone V SoC device and stored in the FPGA calibration RAM.

bell-shape, but it is asymmetric with a long tale on the right. The average cell delay drops but the standard deviation increases. This is because the real cell delay is not only the delay of the physical adder but also depends on the skew of the clock used to register the output of the adder. In an ultra-scaled device, the clock skew becomes more important because the average delay cell is very low, and impacts directly on the delay cell uniformity and performances. The DNL and INL of Fig. 43 show again this aspect. Nevertheless, it is worth underlining that the performance analysis reported is referred to a TDL where a single cell is effectively composed by 4 adders. In literature it is easy to find TDL implemented in a Cyclone V device that reaches low average delay cell (about 6.6 ps rms) and a good uniformity, like [59]. However, the DNL and INL trends of Fig. 43 are quite similar to the ones found in literature, that is the range of DNL and INL covers a time interval equal to some cell delays. As mentioned in the previous paragraphs, the TDL performance required by the Synchronization Circuit is not

Table VI: Cyclone V TDL performances.

Parameter	**Value**
Min Cell Delay	3 ps
Mean Cell Delay	24.5 ps
Max Cell Delay	79 ps
Standard deviation	13 ps
DNL	[-24; 54] ps
INL	[-189; 188] ps

so stringent. According to (56), the resolution of the PLL phase step is about 238 ps (f_{VCO} = 525 MHz), thus the performance of the implemented TDL are satisfactory for the Synchronization Circuit.

3.5 Experiments and Results

In this section, three experiments are reported for both implementation of the Synchronization Circuit. In the first two experiments, the Synchronization Circuit was used to resynchronize the internal FPGA clock in order to generate square pulses or sinusoidal bursts, while in the last experiment it was employed to re-phase the echo signals generated by the “Flow Emulator” (described in the next chapter).

3.5.1 Cyclone III

3.5.1.1 Re-phasing of a square pulse

In the first test the proposed circuit was used to produce a simple pulse synchronous with the CLK_{Sync} clock every time an event on the TDL input was detected. The set-up is sketched in Fig. 45. The function generator 33612A (Keysight Technologies Inc. Santa Rosa, CA, USA) was set for generating a pulse every 200 µs. This pulse was used as TDL input (*Sync*) and as trigger input to the oscilloscope TDS5104 (Tektronix, Inc. Beaverton. OR, USA). The output jitter of the function generator is lower than 1 ps, and the jitter of the scope with respect to trigger is 8 ps rms, both can be neglected in this experiment. The scope input was connected to the pulse generated by the proposed system, and its display set for high persistence. The preliminary on-line calibration was performed like

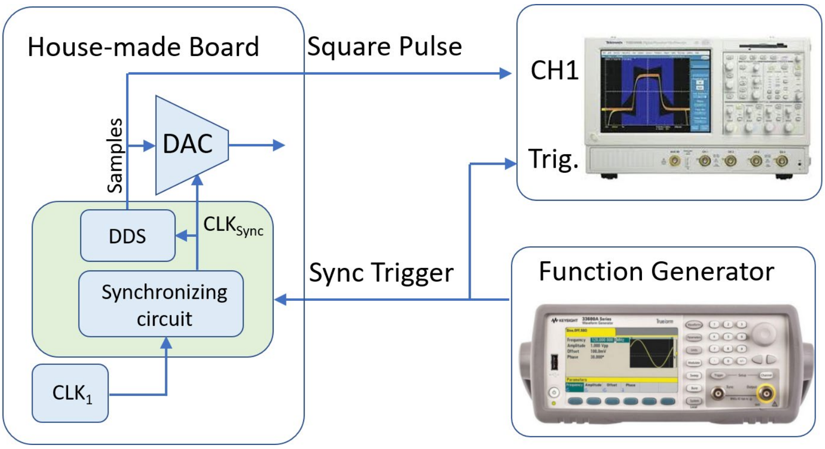


Fig. 45: Experimental setup for the “Re-phasing of a square pulse” test.

described in 3.2.2 before data acquisition. Fig. 46-A shows in a 4 ns/div scale a scope screenshot taken with the proposed circuit not active (it was forced $CLK_{Sync} = CLK_{TDL}$). In this case the display shows, as expected, a jitter that spans in a range of 10 ns, i.e. the CLK_{TDL} period. Then the proposed circuit was activated, and the experiment was repeated. In this condition the jitter is noticeably reduced, as qualitatively visible in Fig. 46-B.

Starting from this condition, the temporal interval among *Sync* pulses was progressively reduced down to the limit when the synchronization circuits began to fail. Fig. 46-C shows this condition, that was reached for an interval of 5 μs.

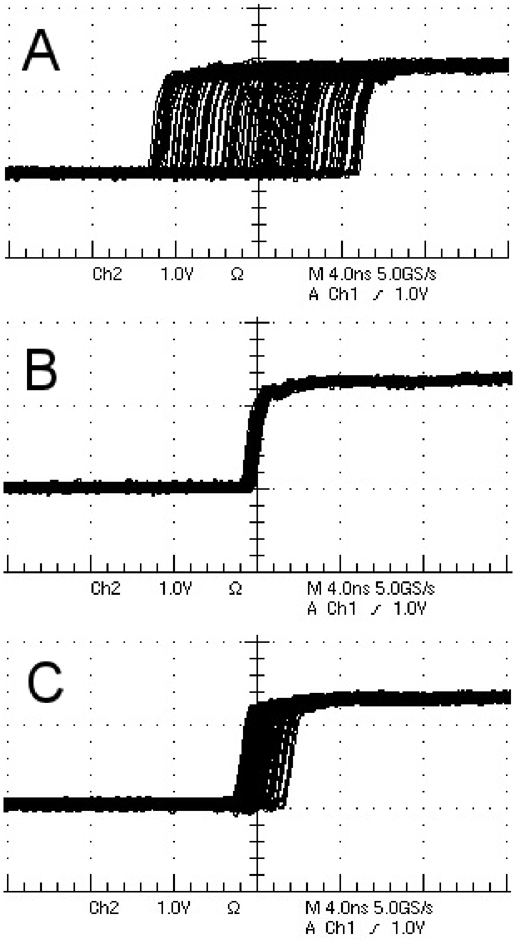


Fig. 46: Pulse sequence synchronized to CLK_{Sync} visualized by TDS5104 scope in high persistence display. A: the proposed re-synchronization circuit is not used and the jitter spans for 10 ns; B: The re-synchronization circuit is activated, and a visible jitter reduction is obtained; C: The temporal interval among Sync pulses is at 5 μs and the circuit starts to fail.

3.5.1.2 Re-phasing of a random sequence of sinusoidal bursts

In the second experiment the house-made system [62], which included the proposed circuit, was coupled to the ADC-SoC developing board from Terasic Inc. (Hsinchu City, Taiwan) as shown in Fig. 47. ADC-SoC includes a SoC FPGA and a 2-channel, 14-bit, 150 Msps Analog-to-Digital (AD) converter. In this experiment ADC-SoC is used for generating a random Sync sequence and for acquiring the sinusoidal bursts produced by the house-made system [62]. The FPGA included a pseudo-random number generator (PRNG) based on a linear-feedback shift register (polynomial 24,23,22,17,0), whose values where used to change the temporal interval among Sync pulses in the range 100µs-10ms. The Sync pulse was synthetized in a state-machine clocked by an on-board low-jitter clock generator (FPGA PLL was not used). FPGA output buffer added further jitter to the Sync pulse, however these contributions can be considered negligible [54] in the experiment.

The house-made system [62] received the Sync pulse and, through the proposed procedure, synchronized the CLK_{Sync} clock. CLK_{Sync} fed a Digital Direct Synthetize (DDS) implemented in [62] that, at every Sync pulse, produced a sinusoidal burst composed by 7 cycles at 3MHz with a Hamming window applied. Samples were Digital-to-Analog (DA) converted at 100 Msps and transferred to the ADC-SoC, where they were acquired by the on-board AD.

In the experiment the acquisition and Sync generation are performed in the ADC-SoC board with the same common clock, CLK_2 in Fig. 47, while the sinusoidal burst generation is re-synchronized to the Sync pulse in the house-made

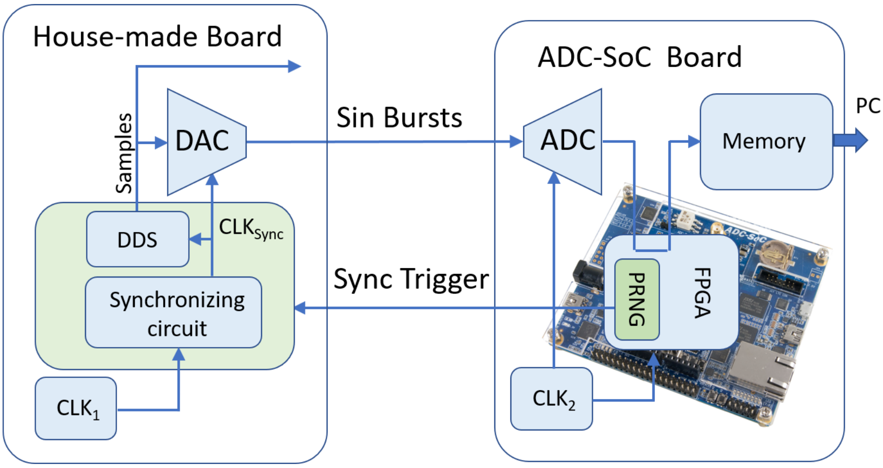


Fig. 47: Experimental setup for the “Re-phasing of a random sequence of sinusoidal burst” test.

board [62]. The tests run by generating 4096 bursts with the resynchronization circuit inactive. Then the resynchronization was activated, and the test was repeated with f_{VCO}= 600 and 900 MHz. The experiement with f_{VCO}= 600 MHz was also repeated with the Bubble-Error-Correction circuit non-active.

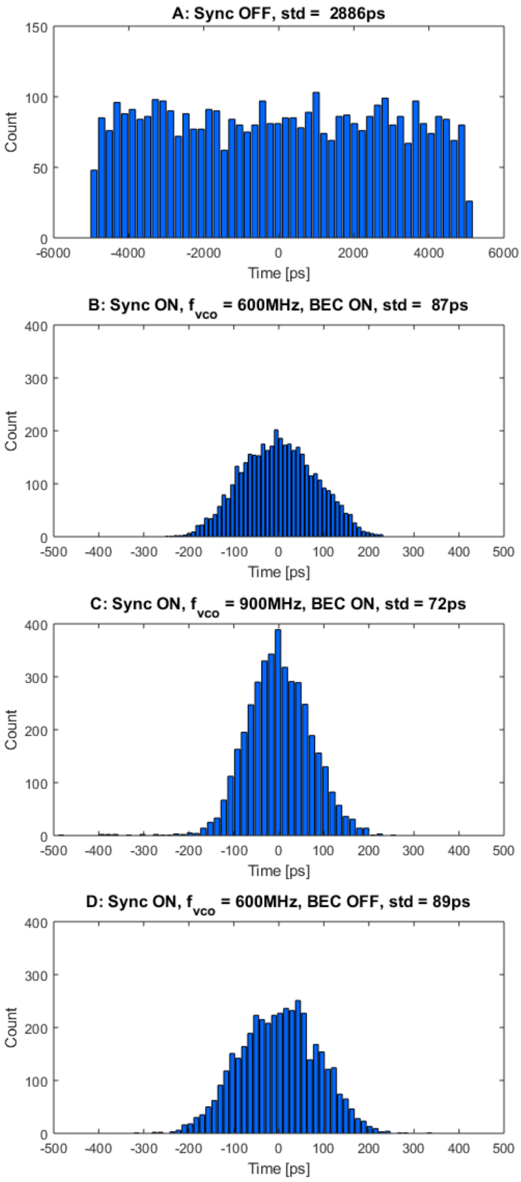


Fig. 48: Histograms of the frame jitter distribution measured in the sinusoidal bursts sequence. A: synchronization off; B: synchronization on with f_{VCO}= 600 MHz; C: synchronization on with f_{VCO}= 900 MHz; D similar to B but with BEC off.

The sinusoidal bursts acquired in the 4 cases were moved from the ADC-SoC memory to PC where they were analyzed in Matlab. The relative phase difference among each group of 4096 bursts was evaluated in the frequency domain [63] after applying the Discrete Fourier Transform (DFT). Fig. 48-A reports the histogram of the relative phases measured at every burst without the synchronization. The mean value was removed. As expected, the jitter features a constant distribution in a ±5 ns range, i.e. the period of the CLK_{TDL} clock used to sample the *Sync* signal. The jitter rms value was 2.88 ns, in accordance to 2.9 ns calculated by (49). Fig. 48-B shows the jitter distribution with resynchronization active and f_{VCO}= 600 MHz. The jitter features a bell distribution with 87 ps rms value. When the VCO is reprogrammed for f_{VCO}= 900 MHz the jitter slightly reduces to a 72 ps rms, like shown in Fig. 48-C. Fig. 48-D shows the frame jitter distribution with BEC circuit disabled.

3.5.1.3 Re-phasing of echo signals generated by the Flow Emulator

The last experiment shows how the Synchronization Circuit is able to rephase the clock of the Flow Emulator used to generate the echo signals of a desired velocity profile. As mentioned in 3.1, the frame jitter is particularly critical for PUV systems. In fact, Doppler analysis is sensitive to that jitter which produces a strong background noise on the spectral matrices, reducing the maximum Signal-to-Noise Ratio (SNR) achievable.

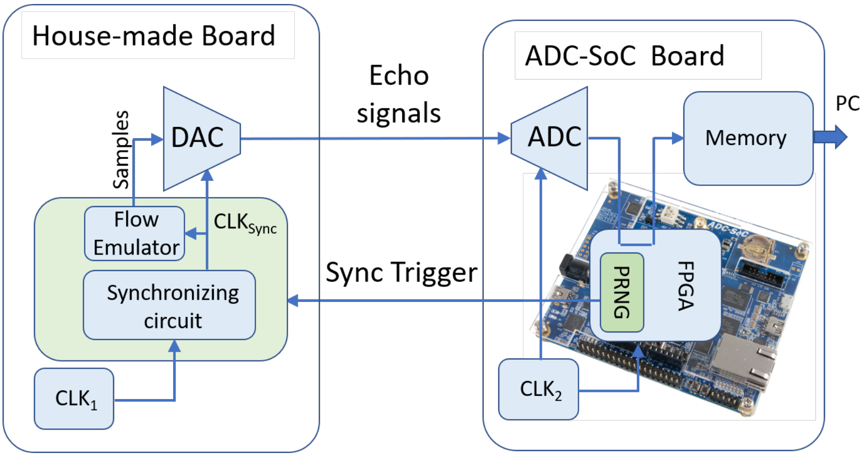


Fig. 49: Experimental setup for the "Re-phasing of echo signals generated by the Flow Emulator" test.

The experimental setup is similar to the one of the previous experiment and is reported in Fig. 49. The House-made board hosts both Flow Emulator and Synchronization Circuit, while the ADC-SoC board is used to acquire the echoes generated. Then the Doppler analysis is performed on the acquired data in Matlab. Note that the ADC-SoC board can be replaced with a PUV system that directly elaborates the echoes, like [62].

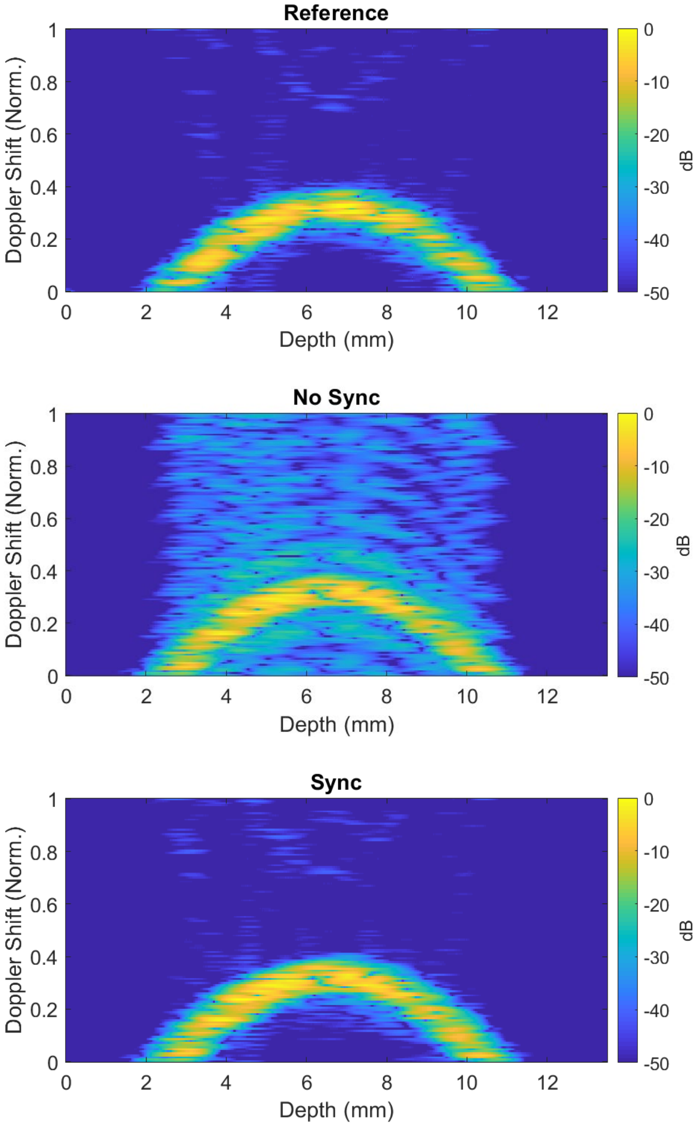


Fig. 50: Power spectral density matrices for the Doppler signal. Depth and frequency normalized to 1/PRI are reported in horizontal and vertical axes, respectively. Power is color code in a 50dB dynamics. Top: reference matrix measured with no jitter. Middle: matrix in presence of jitter and Synchronization circuit not active. Bottom: in presence of jitter and Synchronization Circuit active.

Fig. 50 reports three power spectral matrices obtained by mimicking a Newtonian fluid that flows in a 8 mm pipe. The "Reference" matrix is obtained by sharing the system clock between the House-made board and the ADC-SoC board through a dedicated cable, thus no synchronization problem is present. Therefore, the reference matrix represents the desired velocity profile with the best SNR for that configuration, i.e. 39.7 dB for the reference profile of Fig. 50. The other two cases of Fig. 50 were obtained by no longer sharing the clock between the boards. The "No Sync" spectral matrix shows the case with the Synchronization Circuit not activated. As previously stated, in this case a 10 ns peak-to-peak noise (i.e. TDL clock period) was present which produces artefact clearly visible in the image. The reduction of the image quality is confirmed by the decrease of the SNR, now reduced to 20.1 dB. The last case of Fig. 50 was obtained after activating the Synchronization Circuit and calibrating the TDL. The artefacts are no more visible and the quality of the image is significantly improved becoming very similar to the reference one. This result is confirmed by the SNR of 39.3 dB, comparable to the 39.7 dB of the reference image.

3.5.2 Cyclone V SoC

The experiments made to evaluate the performance of the Synchronization Circuit implemented in the Cyclone V FPGA are the same of the Cyclone III case. The experimental setups are identical to Fig. 45, Fig. 47 and Fig. 49 except for the House-made board that is replaced by one that hosts the Cyclone V SoC FPGA.

3.5.2.1 Re-phasing of a square pulse

Similarly to 3.5.1.1, the Keysight 33612A function generator was set for generating Sync (see Fig. 45) pulses every 100 µs, which feed both the TDL and a channel of the oscilloscope used as trigger. A pulse is generated by the proposed system with the re-phased clock CLK_{Sync} and sent to another channel of the oscilloscope. Fig. 51-A shows the high persistence display of the oscilloscope for the square pulses generated by the system when the Synchronization Circuit was not enabled. As expected, the range of variation of the square pulses was about 9.5 ns, i.e. the period of the sampling clock (105 MHz). Enabling the Synchronization circuit and calibrating the TDL as described in 3.2.2, the range of variation of the square pulses reduces significantly, as shown in Fig. 51-B.

The temporal interval among *Sync* pulses was progressively reduced, as in 3.5.1.1, to find the limit where the Synchronization Circuit starts to fail (see Fig. 51-C). In both FPGA implementations the Dynamic Shifting Interface of the PLL works with a 100 MHz clock, thus the shifting time per step is the same. However, the PLL step resolution of the Cyclone V implementation is lower than in the

Cyclone III case, thus few PLL shift steps are required. In fact, the limit for the temporal interval among subsequent Sync pulses is now reduced to 4 µs.

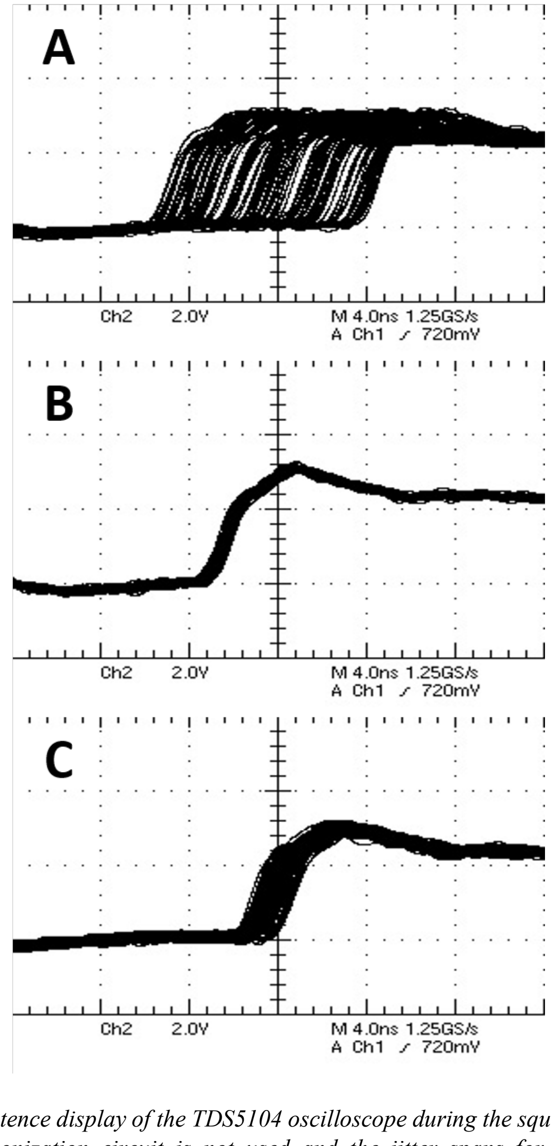


Fig. 51: High persistence display of the TDS5104 oscilloscope during the square pulse test. A: the proposed re-synchronization circuit is not used and the jitter spans for 9.5 ns; B: The re-synchronization circuit is activated, and a visible jitter reduction is obtained; C: The temporal interval among Sync pulses is at 4 µs and the circuit starts to fail.

3.5.2.2 Re-phasing of a random sequence of sinusoidal bursts

In the second experiment, the ADC SoC board is used to generate random Sync sequences like in 3.5.1.2. The Synchronization Circuit, implemented in the proposed system, uses the Sync signal (see Fig. 47) to realign the internal clock CLK_{Sync}, used also for the digital-to-analog (DA) conversion. Every Sync pulse, the proposed system generates a sinusoidal burst composed by 7 cycles at 3 MHz (Hamming windowed). Then the sinusoidal burst is acquired by the ADC SoC board with the same clock used for the Sync generation. The raw data are moved in Matlab to further elaboration. In particular, the phase of each burst is evaluated to find the distribution of the residual frame jitter.

First, the test runs by generating 4096 sinusoidal bursts with the resynchronization circuit disabled. As expected, the results of Fig. 52-A shows an uniform distribution of the frame jitter that range in temporal interval of about 9.5 ns, i.e. the TDL clock period. Repeating the test with the Synchronization Circuit activated, the range of variation of the frame jitter reduces significantly up to 107 ps rms, as reported in Fig. 52-B.

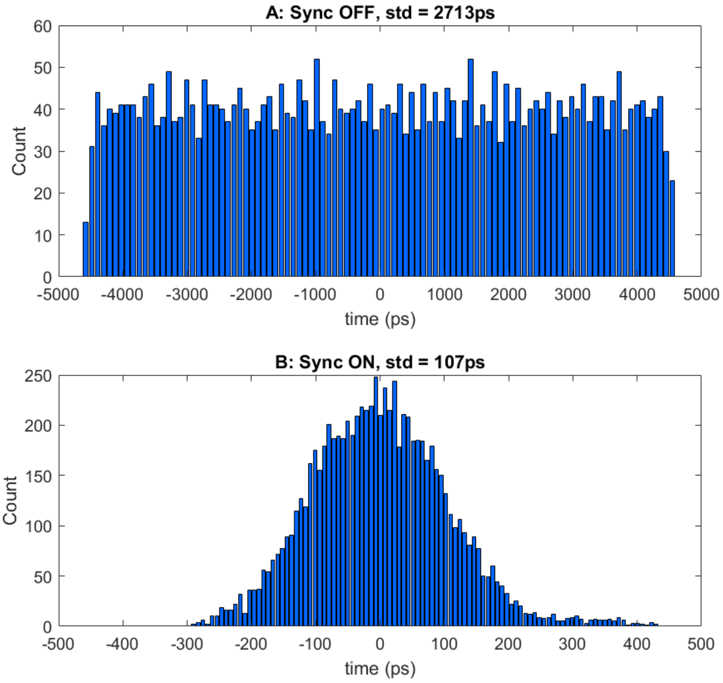


Fig. 52: Histograms of the frame jitter distribution measured in the sinusoidal bursts sequence with the Synchronization Circuit deactivated (A) and enabled (B).

3.5.2.3 Re-phasing of echo signals generated by the Flow Emulator

This experiment lets to evaluate how the Synchronization Circuit helps to reduce the frame jitter in the echoes generated by the Flow Emulator. The latter was set to emulate a flat profile of a fluid that flows in a 10 mm pipe with a velocity peak

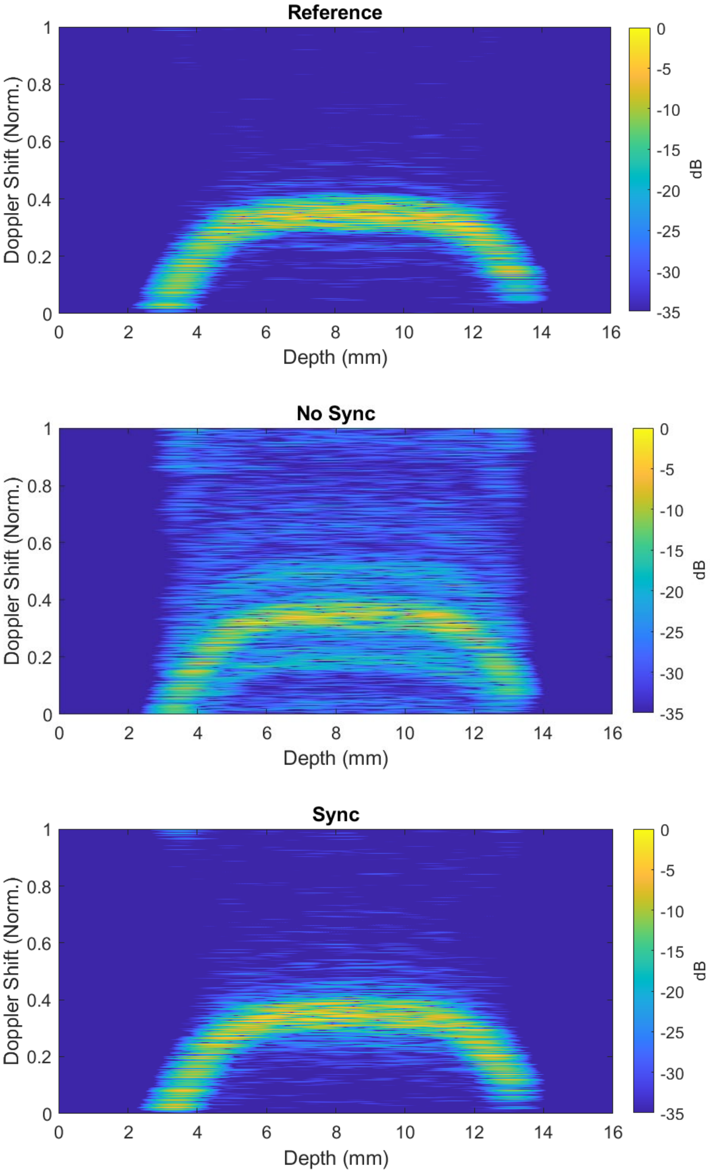


Fig. 53: Power spectral density matrices for the Doppler signal color-coded in a 35 dB dynamics. Top: reference matrix measured with no jitter. Middle: matrix in presence of jitter Synchronization and circuit not active. Bottom: in presence of jitter and Synchronization Circuit active.

of 30 cm/s (see Fig. 53). As in 3.5.1.3, the “Reference” profile represents the best configuration in terms of SNR and it was obtained by sharing the system clock between the Flow Emulator and the ADC SoC boards. The SNR measured for this case is of 31.2 dB. When the system clock was not shared between the boards and the Synchronization Circuit was off, the result of the middle of Fig. 53 was obtained, where the SNR drops to 16.7 dB. Enabling the Synchronization Circuit, the “Sync” case of Fig. 53 was obtained, where the background noise is no more present. The SNR measured in this case is about 30.5 dB, very close to the “Reference”.

3.5.3 Discussion and conclusion

The Synchronization Circuit presented in this chapter is able to re-phase a clock to every occurrence of the Sync signal. The proposed method is based on a full digital approach, suitable for FPGA implementation. However, the realization of the TDL in FPGA needs a careful low-level programming to correctly build the delay chain. Moreover, the TDL design changes with the target device, as shown in the Altera-Intel Cyclone III and Cyclone V SoC FPGA implementations. Once the desired result is achieved, the TDL should be locked in the hardware to grant the needed reproducibility to the project. In spite of these difficulties, the delay elements feature very low values (also below 10 ps), but not so small standard deviations. Moreover, the TDL calibration at run-time compensates for the cells mismatch and reduces the effects of the unavoidable variations due to aging, temperature, voltage.

Commercial integrated synchronization devices feature better jitter performance, also in the order of 100 fs [64], but unfortunately cannot work with arbitrary trigger sequences, which are often present, for example, in PUV. Moreover, such low values are not required for application of interest like Doppler Ultrasound techniques. Additionally, these devices require an hardware upgrade, while the proposed circuit can be added to a system that embeds a FPGA (like almost all ultrasound systems) by simply changing the firmware.

The proposed circuit was implemented and tested in a Cyclone III and a Cyclone V SoC FPGAs, nevertheless it can be retargeted to different devices relatively easily. The most challenging section is the TDL, since its dependability on the actual hardware structure and signal routing, but literature reports several examples of TDL implemented in different FPGAs: Xilinx Virtex-5 XC5VLX110T [49], EP4CE55F23C8 (Cyclone IV, Altera) [48], EP2C8T144C6 (Cyclone II, Altera) [51], and others.

The variety FPGA hardware structure and technology can also require different encoders. For example, in the Cyclone III device it was implemented a Bubble

Error Encoder (see 3.3.1.2) while in the Cyclone V a Zero-Counter Encoder (see 3.3.2.2). This because in recent ultra-scaled FPGAs (like the Cyclone V) the internal clock skew weighs much more than in older devices (e.g. Cyclone III). Therefore, the bubble problem in the thermometric code at the output of the TDL became more serious and the tap order of the TDL is no more consistent with the real delay. This drove the need to find other ways to read the temporal information at the output of the TDL in advanced FPGA. As shown in 3.3.2.2, the most common approaches are the bin realignment and the one-counter encoder.

The impact of the clock skew in recent FPGA is also visible in DNL and INL trends. Fig. 40 and Fig. 43 show a greater non-uniformity in the Cyclone V device. This because the average delay value is much smaller than that of Cyclone III implementation (6.7 ps respect to 44 ps). Moreover, the detected delay is not only related to the delay element (LUT adder or physical adder) but also depends on the skew of the clock used to register the output of the adder. Actually it should be noted that in Fig. 43 the DNL and INL describes the linearity of the TDL implemented with B-cell, i.e. 4 physical adders jointed together.

The PLL reprogramming acts on the selection of the ring-oscillator high-frequency output and on the reset instant of the following digital divisors. The VCO frequency and phase detector is untouched, thus the PLL never lose the lock condition, which is essential to grant continuity and reliability to the output clock.

The phase of the generated clock should be re-aligned to the Sync input at every Sync pulse. The time needed for the operation is critical and impacts on the real-time performance of the proposed method. Phase measurement on TDL and calibration through look-up RAM take few clock cycles. The most time-consuming operation is the PLL reprogramming through the DPSI serial interface. In the proposed implementations, the $CLKD_{DPSI}$ is 100 MHz for both cases, while f_{VCO} is 600 MHz and 525 MHz for Cyclone III and Cyclone V implementations respectively. Therefore, 50 or 40 shift steps are needed respectively in the worst cases to align back CLK_{Sync} to CLK_{TDL} and other 50 or 40 steps to achieve the alignment to the actual Sync edge. This corresponds to $(50 + 50)\cdot 50$ ns = 5.0 µs for the Cyclone III and $(40 + 40)\cdot 50$ ns = 4.0 µs for the Cyclone V, as confirmed by the experiments (see Fig. 46-C and Fig. 51-C). However, this value is widely compatible with the timings of most applications of interest [65].

3.5.4 Contributions

The work exposed in this chapter was published in the follows papers:

Journal paper

- **Russo, Dario**, Stefano Ricci. «FPGA Implementation of a Synchronization Circuit for Arbitrary Trigger Sequences». *IEEE Transactions on Instrumentation and Measurement*, 2019.

Conference proceedings

- **Russo, Dario**, Stefano Ricci. «FPGA-based Clock Phase Alignment Circuit for Frame Jitter Reduction». In *Applications in Electronics Pervading Industry, Environment and Society*, 2019.
- **Russo D.**, Ricci S., «Low-Jitter Systems Synchronization for Doppler Measurements». In *IEEE International Ultrasonics Symposium (IUS)*, 2019.
- **Russo D.**, Ricci S., «FPGA-based Trigger-Synchronizer for low Frame-Jitter Signal Generation». In *IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2019.

Chapter 4. Flow Emulator

In this chapter a flexible electronic system, called Flow Emulator, is presented, that can be used to test industrial and biomedical Doppler systems. Classical tests require hydraulic circuits like flow-rig and phantoms to reproduce a known flow configuration. However, they are affected by several issues, the main being the lack of an accurate reference of the velocity distribution developed by the fluid. The proposed Flow Emulator is an Electronic Doppler Phantom (EDP) that generates the radio-frequency echo signals of a real-like and programmable flow and pipe configurations. Two versions of the Flow Emulator has been developed, both based on an FPGA: the first version is the simplest and it allows to emulate echo signals previously generated in an ultrasound simulation software; the second version, implemented in a last-generation FPGA, adds the real-time signal generation based on the summation of the contributions of random scatterers.

4.1 Introduction

Ultrasound Doppler techniques are nowadays widely employed in both biomedical and industrial applications. They are implemented in clinical and research echographs like ULA-OP (2.3.1), and in electronic systems for industrial fluids characterization, like the V3 system (2.2.1). Industrial and academic research is highly active in the field, and improved methods and novel dedicated electronics systems are continuously proposed. The experimentation of a new Doppler method, and its deployment in an electronics system, requires several tests [30][66], which are typically carried out through ultrasound Doppler phantoms and flow-rigs [67]-[70]. These represent hydraulic circuits where a pump moves a scattering fluid through a structure that mimics a morphological tissue or an industrial part. The testing fluid has known properties and it flows in the pipe circuit under controlled conditions, like flow rate, temperature, pressure, etc. In this way, the Doppler System Under Test (DSUT) works similarly to its final application and the test results will be compared to the know fluid properties and configuration (e.g. velocity peak). Unfortunately, flow-rigs and phantoms are affected by several problems: the choice and preparation of the material for phantoms fabrication is not trivial [71]; the fluid employed requires a long preparation [72]; flow-rig are very cumbersome and its set-up is very time-consuming, limiting the number of tests that are typically performed. Moreover, flow-rigs are not always present in the laboratories where the Doppler method or systems are developed. Finally, the main drawback of flow-rigs is that the exact flow velocity profile present in the pipe is only partially known because of uncertainties in the fluid features and flow conditions. This is an important limit in the evaluation of the accuracy of the method or DSUT performance.

Electronics Doppler Phantoms (EDPs) represent an interesting alternative to hydraulic phantoms. They consist in electronics boards capable of injecting at the input of the DSUT a signal that mimics a known and programmable flow. The typical EDP [73]-[78] is electrically or acoustically coupled to the transducer of the DSUT. It synthetizes the desired Doppler shift and then modulates it over the signal transmitted by the DSUT. The result is injected back in the DSUT receiver. In spite of the potential capacities of this approach, no new EDP has been developed for long; thus the available devices, based on old electronics technology, result quite basics. In fact, they generate a simple frequency tone, or a spectrum obtained by shaping a white noise; emulate a single sample volume only; and manage a very limited set of parameters. These constraints make these EDPs inappropriate for the testing of the complex Doppler methods and systems nowadays employed in research laboratories and industrial production sites [66][79].

In this chapter, the design of a compact and flexible EDP for pulsed wave applications is presented. It overcomes most of the aforementioned limitations. Two versions of the "Flow Emulator" system are reported: the first version is able to reproduce echo signals previously generated off-line while the second version, implemented in a newer FPGA, synthesizes the echo signals in real-time according to the pipe and flow configuration programmed by the user.

4.2 Flow Emulator v1

The first version of the Flow Emulator is basically a signal synthesizer that produces, for every PRI (Pulse Repetition Interval), the complex ultrasound echo signal generated by the fluid scatterers, like the one shown in Fig. 54. The samples of the echo signals to be emulated are previously generated in Matlab® (The Mathworks, Natick, MA) through an ultrasound simulation software. This signal simulates a flow where features like velocity profile shape, velocity peak, Signal-to-Noise Ratio (SNR) and Clutter-to-Signal Ratio (CSR) are known. The DSUT, connected to the emulator as in Fig. 55, will process the output signal of the emulator that should results in the desired spectral matrix and velocity profile. Although the tests presented for this version of the emulator are related to industrial applications, the profile generation process also applies to medical fields, where it is equally important to assessing the blood flow configuration in the vessels.

The synchronization between the emulator and the DSUT is a critical aspect. Indeed, sub-ns random temporal variation between the DSUT synchronism (that

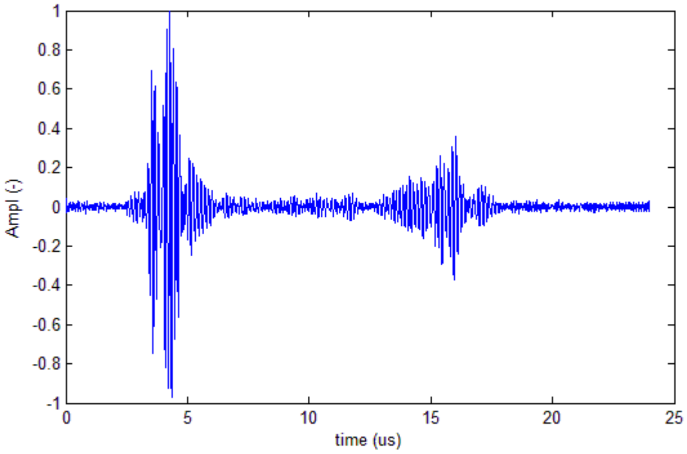


Fig. 54: Example of RF signal from a fluid in a 8mm pipe investigated at 7MHz.

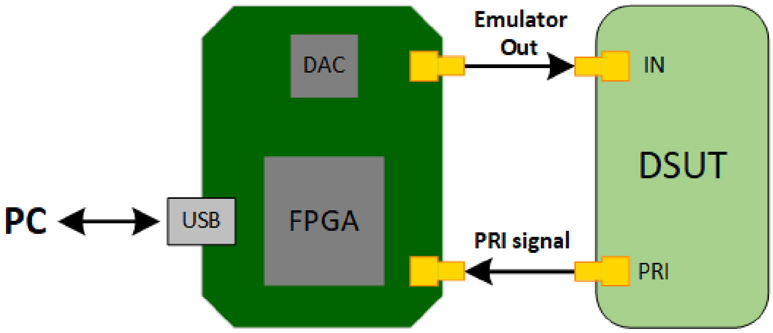


Fig. 55: Connection between the Flow Emulator and the DSUT.

is the PRI signal) and the effective start of the echo generation produce an unbearable phase noise in the Doppler signal. For this reason, the emulator includes the custom re-synchronization circuit, analyzed in the previous chapter, that reduces the jitter below 100 ps rms, suitable for the application.

4.2.1 Hardware architecture

The main features of the Flow Emulator are listed in Table VII. The system is based on a custom electronic board that includes the EP3C25F256 Field Programmable Gate Array (FPGA) from Cyclone family of Altera-Intel (San Jose, CA, USA). The FPGA is connected to an EPCS flash memory (Altera-Intel) and a 64 MB SDRAM of Micron Technology (Boise, USA), used as memory buffer by the emulator, and an AD9707 (Analog Devices, Norwood, MA, USA) 14 bit digital-to-analog (DA) converter. The 14 bit of the DA accommodates the large dynamics produced by the strong signal from the wall and the weak echoes from

Table VII: Main features of the Flow Emulator v1.

Parameter	Value
Channel	1
Output voltage	Up to 400 mVpp
Output frequency range	0.1 ÷ 10 MHz
Output burst	Arbitrary waveform
PRI range	0.1 ÷ 10 ms
Sampling Freq.	50 Msps
Resolution	14 bit
SDRAM size	64 MB
Flash size	128 MB
Pipe diameter	Up to 15 cm

the fluid, respectively. An analog section follows the DA convert that amplifies the signal up to 400 mVpp over a 0.1-10 MHz bandwidth. A Universal Serial Bus (USB) interface lets to communicate with the emulator and transfer the data, while two SMA connectors are used for the emulator output and the PRI signal input.

4.2.2 FPGA Architecture

The FPGA architecture of the Flow Emulator is sketched in Fig. 56. The “Sync Circuit” is the custom block that synchronize the internal FPGA clock, called “Clk sync”, with every occurrence of the input trigger, i.e. the “PRI signal”. The Clk sync must feed the internal FIFO memory and it is also used as clock for the external DA converter after a buffer stage.

The FPGA includes a NIOS II® soft-processor that manages all the board operations and the communication to the host PC through the “USB cntr” block. During the initialization phase, the soft-processor loads from the PC the echo signal samples, previously generated off-line in Matlab® (see 4.2.4), and moves them in the SDRAM and in the Flash memory for non-volatile storing. The echoes samples are reproduced by the DA converter at 50 Msps, 14 bit. Accordingly, the memories of the system can store several thousands of PRIs. For example, a 10mm diameter pipe is emulated with an echo burst of 13 µs temporal length (sound velocity 1500 m/s), corresponding to about 650 samples per PRI and 1200 bytes. In this example the SDRAM and the Flash memory can accommodate 53k and 106k PRIs respectively. The Flash memory is quite slow compared to the SDRAM and the PRI values required by the application, thus the stored data are moved to SDRAM when in use.

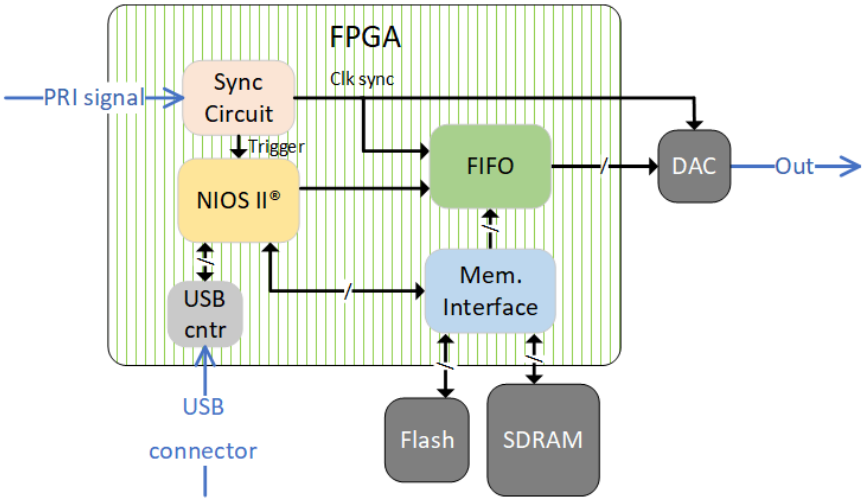


Fig. 56: FPGA architecture of the Flow Emulator v1.

After the board initialization, the first PRI is moved from the SDRAM to the FIFO memory in the FPGA, ready to be produced. Now the emulator waits for the PRI trigger from the DSUT to start the echoes generation. At trigger, the FPGA waits a programmable time that accounts for the depth of the pipe, then starts the signal production. From now on, every next PRI trigger, the soft-processor starts the data transferring from the FIFO to the DA converter, and simultaneously, prepares the next PRI samples in the FIFO memory. These operations are repeated until the whole available signal is reproduced or the DSUT stops to send the PRI triggers.

4.2.3 FPGA Resource Usage

The resources used to implement the Flow Emulator in the Altera-Intel EP3C25F256 Cyclone III device are listed in Table VIII. Logic cells and memory bits usage are reported for the blocks in the architecture of Fig. 56, while no Digital Signal Processor (DSP) units are used. The number of total elements used and the percentage with respect to the overall device capability are reported on the bottom of Table VIII. The project reached the time closure with a 100 MHz clock.

Table VIII: Cyclone III FPGA resources.

Section	***Logic cells***	***Memory bits***
Mem. Interface	565	-
FIFO	-	28672
NIOS II	3973	72640
USB cntr	51	-
Sync Circuit	1599	2176
TOT	6215 25.2%	111680 18.4%

4.2.4 Echoes Signal Synthesis

The signal that the Flow Emulator produces is generated off-line through a specialized ultrasound simulation software called “Field II” [80][81], freely available at http://field-ii.dk. Field II is a well-established ultrasound simulator that is widely used in the biomedical research for ultrasound imaging simulations. An example is reported in [82], where Field II is used together with other external specific CADs for modelling the behavior of complex non-Newtonian fluids or pipe geometries.

Field II works as extension of Matlab and it is capable of calculating the emitted and received ultrasound fields for several different types of transducers. In particular, given the geometrical and electrical features of the transducer, the samples of the transmission signal, the static configuration of the scatterers present in the field of view of the transducer, the desired Signal-to-Noise Ratio (SNR), etc., Field II generates an accurate simulation of the RF signal received from the mimicked configuration. Moreover, it is possible to generate a Doppler simulation of a flow by updating the scatterers configuration between successive PRIs according to the desired flow velocity profile.

The simulations performed for generating the Flow Emulator signals refer to cylindrical transducers and pipes. As previously stated in Chapter 2, in most industrial processes the production involves fluid and/or suspensions that have non-Newtonian behavior. For this kind of fluids, the viscosity is not constant but depends on the shear rate by a non-linear relation. One of the simplest and most effective models that describes the non-Newtonian behavior is the Power-Law model which states the viscosity-shear rate relationship as (35). This model uses only two indices K and n (Power-Law consistency index and Power-Law exponent respectively), the latter of which mainly affects the shape of the velocity profiles (see Fig. 18). Using the formulas of the paragraph 2.1.2, it is possible to describe the velocity profile developed by a non-Newtonian fluid that is flowing in a pipe of radius R as function of the flow-rate Q, R and n:

$$v(r) = \frac{Q}{\pi R^2}\frac{3n+1}{n+1}\left(1-\left(\frac{r}{R}\right)^{1+\frac{1}{n}}\right) \qquad (60)$$

where $v(r)$ is the velocity in the parallel direction to pipe axis at distance r from the pipe center.

To complete the model, the Power-Law consistency index K can be related to the flow-rate Q and the pressure drop ΔP over the distance L (between the pressure sensors) measured in pipe axis direction:

$$K = \frac{\Delta P}{2L}\left(\frac{3Q}{\pi}R^{\frac{-3n-1}{n}}\right)^{n} \tag{61}$$

The user selects in the Matlab interface of the Flow Emulator the parameters needed by the (60) to obtain the velocity profile. The latter is used in Field II for generating the RF signal of the echoes, together with other parameters, always set in the Matlab interface, used by the simulation software Field II like the excitation frequency, transducer features, PRI length, and so on. Finally, the Flow Emulator interface calculates the pressure drop ΔP over the distance L by inverting the equation (61). Indeed, this value is required by the DSUT for the assessment of the rheological properties, in addition to the velocity profile obtained through ultrasound. Actually, the emulator displays $\Delta P/L$ for convenience.

Since a complete Doppler simulation in a typical PC can last a relative long time, a set of pre-calculated signals are stored in a signal library that can be used at any moment. Library emulates fluids or suspensions with different rheological features and/or different flow conditions (pipe diameter, volume flow rates, etc.), or even acquisition condition, like signal-to-noise (SNR) ratios or other disturbs.

4.2.5 Experiments and Results

In this paragraph, some experiments to test the Flow Emulator are described. First, the capability of the emulator to generate profiles with different SNR (SNR test) and shape (Profile Shape test) is shown and, finally, two fluids with different rheological features are emulated (Emulation test). The experimental set-up for all tests is shown in Fig. 55. Here, the DSUT is the V3 system described in 2.2.1.

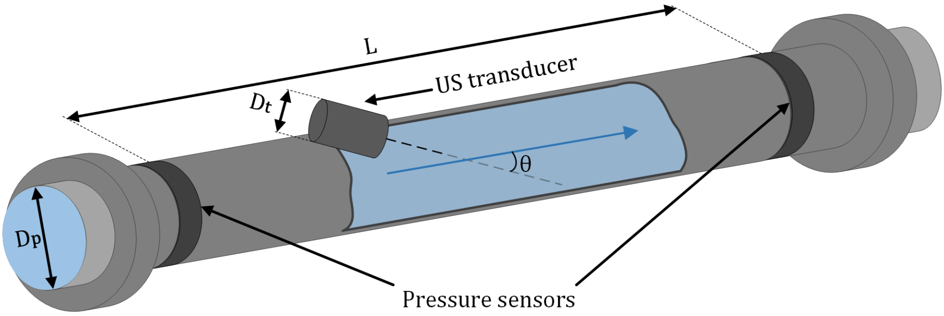


Fig. 57: Transducer-pipe set-up mimicked by the Flow Emulator in the reported experiments.

4.2.5.1 SNR Test

In this test, a simple parabolic profile, typical of Newtonian fluid like water, is generated. The parameters used for the experiment are reported in Table IX. In particular, the Flow Emulator mimics the experimental setup of Fig. 57 that involves a 7mm diameter cylindrical transducer excited by sinusoidal bursts at 5 MHz. The Doppler angle θ, i.e. the angle between the transducer axis and the flow direction, is equal to 60°. The parabolic velocity profile has a velocity peak of 0.5 m/s that correspond to a Doppler frequency of 1689 Hz or 0.34 when normalized with respect to 1/PRI. A white noise was added to the signal for achieving a SNR of 10, -15 and -20 dB. For each profile, 1024 PRIs were stored in the emulator memory. The DSUT was programmed to produce a power spectral matrix every 64 PRIs. Fig. 58 reports, in a 60 dB dynamics, the power spectral matrices and also the velocity profiles measured by the DSUT for the three value of SNR. The case of SNR equal to 10 dB (Fig. 58-top) represents the condition of a good industrial set-up in a typical application, and the profile is clearly detectable. In case of -15 dB (Fig. 58-middle), the background noise is visible, but the profile is still detectable. When the SNR is lower than -20 dB (Fig. 58-bottom) part of the profile is confused with the background noise and the velocity profile detection fails.

Table IX: Parameters used in the SNR and Profile Shape tests.

Parameter	Value
General	
PRI per exp.	1024
PRI length	0.2 ms
Sample/PRI	2048
Transducer and Transmission	
Sensor Diameter	7 mm
Bandwidth	3 ÷ 7 MHz
Burst	Sinusoidal
Frequency	5 MHz
Cycles	5
Apodization	Hanning
Pipe and Profile	
Diameter	16 mm
Velocity Peak	0.5 m/s
Doppler angle	60°
Profile shapes	Parabolic, Smashed, M-shape
SNR	+10,-15,-20 dB
Peak Doppler shift	1689 Hz ; 0.34/PRI

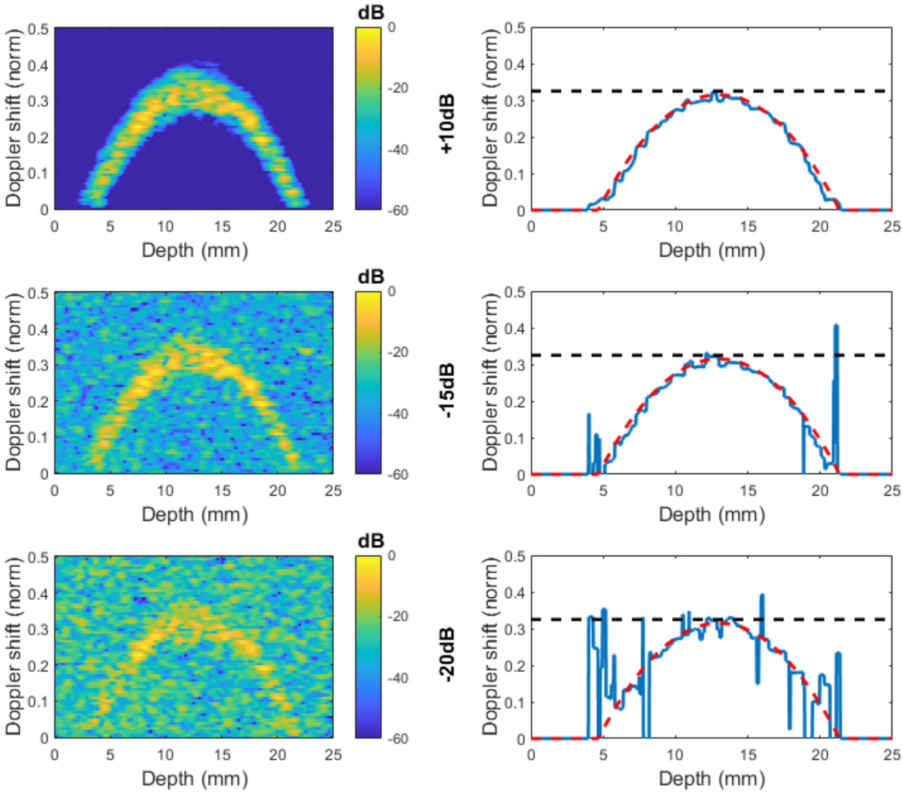


Fig. 58: Power spectral matrices (left) and velocity profiles (right – continuous blue curves) measured by the DSUT when the Flow Emulator mimics a parabolic profile in a 16mm diameter pipe with SNR of 10 dB (top), -15 dB (middle) and -20 dB (bottom). The red and black dotted lines on the right represent the desired velocity profile and the velocity peak, respectively.

4.2.5.2 Profile Shape Test

In the Profile Shape test, the same parameters used for the previous test, reported in Table IX, are used. In addition to the parabolic velocity profile, a Smashed and M-shape profiles are emulated [83]. The first profile, as previously seen, is typical of several non-Newtonian fluids often used in industrial applications. The second profile is a particularly complex flow profile that can be found in non-straight pipe configurations and non-steady flow like, for example, after pipe curvatures. In both cases a SNR of 10 dB was used and, as in the previous experiment, 1024 PRIs of the signal were stored in the emulator memory for each profile. Again, the DSUT was programmed to produce a power spectral matrix every 64 PRIs (128-point FFT with 50% overlap). Thus 8 frames per profile were produced and averaged, obtaining the power spectral matrices and the velocity profile shown in

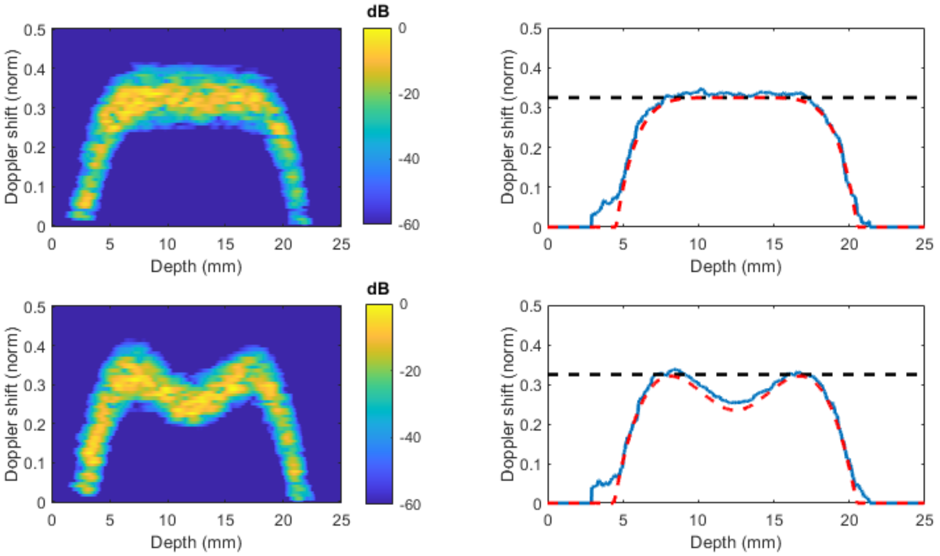


Fig. 59: Power spectral matrices (left) and velocity profiles (right – continuous blue curves) measured by the DSUT during the Profile Shape test. A typical smashed profile (top) of a non-Newtonian fluid and a M-shape profile (bottom) are reported. In both cases, the emulated pipe diameter was 16 mm and the SNR 10 dB. The red and black dotted lines on the right represent the desired velocity profile and the velocity peak, respectively.

Fig. 59. The continuous blue curves represent the measured velocity profiles and the black dotted lines the velocity peaks which are, as expected, at 0.34 (normalized peak Doppler shift). Instead, the red dotted lines represent the reference profiles used for the echoes signal generation, that were compared to the measured profile and their agreement was quantified by evaluating the root square mean error (RMSE) between the curves. The RMSE for the parabolic (top case of Fig. 58), smashed and M-shape profiles are listed in Table X. These measurements confirm the almost perfect correspondence between the measured and reference velocity profiles of Fig. 58-top and Fig. 59. A slight difference is visible at the profile borders, near the 0-frequency. However, this is the typical artefact due to the clutter and the finite dimension of the transmitted US packet [2]. This is not a flaw of the Flow Emulator, rather a confirmation of its correct reproduction of a real-like RF signal.

Table X: Root Mean Square Error (RMSE) results.

Profile type	RMSE
Parabolic	2.9 %
Smashed	4.7 %
M-Shape	4.7 %

4.2.5.3 Emulsion Test

In this test, two examples of fluids (labelled F_1 and F_2), whose rheological data are inspired to cosmetic emulsions, are reported in Fig. 60. They have different rheological features, like reported in Table XI. The top panel of Fig. 60 shows the viscosity/shear rate trend, according to the Power-Law model. Both emulsions are shear-thinning type, i.e. the viscosity decrease with increasing shear-rate, as expected for the Power-Law exponent $n < 1$. The corresponding velocity profiles for a 16 mm diameter pipe and a flow-rate Q = 2.5 ml/s are plotted in the bottom of Fig. 60. As expected, they are flattened (or smashed) profiles .

The emulation parameters are reported in Table XII. In particular, as in the previous tests, a 7 mm cylindrical transducer, excited by sinusoidal bursts composed by 5 cycles at 5 MHz was mimicked. It investigated a 16 mm pipe where the emulsions flowed with a flow rate of 2.5 ml/s and a Doppler angle of 60°. 1024 PRIs were simulated and downloaded on the emulator, that generated

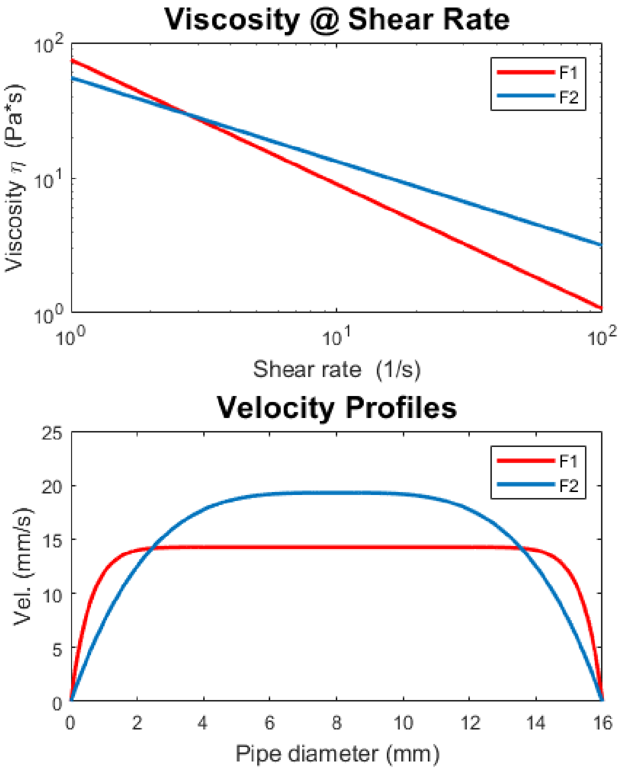


Fig. 60: Relation between shear rate and viscosity (top), and velocity profile (bottom) of 2 fluids (F1 in red and F2 in blue) with the different rheological characteristics reported in Table XI.

Table XI: Rheological and flow parameters.

Parameter	Symbol	Value	
		F1	**F2**
Flow rate	Q	2.5 ml/s	2.5 ml/s
Exponent	n	0.08	0.038
Consistency index	K	74.61	54.99
Velocity peaks	v_p	14 mm/s	19 mm/s
Shear rates	$\dot{\gamma}$	24.09 1/s	8.75 1/s
Viscosity	η	4 $Pa \cdot s$	14.3 $Pa \cdot s$

them with a PRI length of 0.7 ms. The DSUT processed the data as in the previous tests, generating 8 frames per profile which were then averaged. Moreover, a wall filter with a cut-off frequency of 70 Hz was used. The velocity profiles obtained by the DSUT, reported as continuous red and blue curves in Fig. 61, were then compared in Matlab to the reference profiles (black dotted lines) and the RMSE was evaluated. The relative RMSE between the curves was lower than 3%.

Table XII: Emulation parameters for the Emulsion Test.

Parameter	Value
Acquisition	
Number of PRI	1024
PRI length	0.7 ms
Samples per PRI	1500
SNR	30 dB
Transmission	
Transducer	Piston 7 mm
Frequency	5 MHz
Number of cycles	5
Apodization	Hanning
Pipe	
Diameter	16 mm
Doppler angle	60°

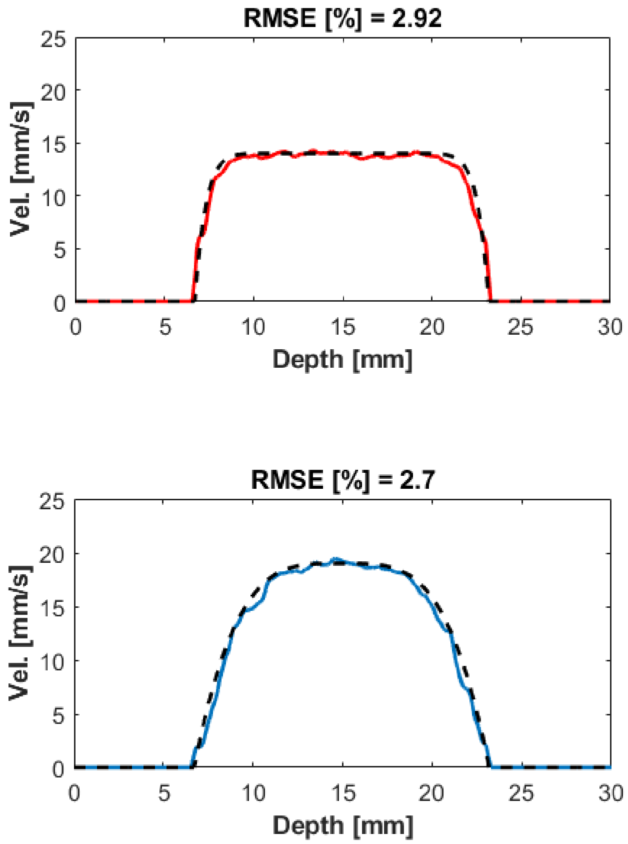


Fig. 61: Velocity profiles measured by the DSUT from the data produced by the emulator for the fluid F1 (red, top) and F2 (blue, bottom). Measured profiles are compared to the reference profiles (black dotted curves); the error is 2.9% and 2.7% for F1 and F2, respectively.

Moreover, the viscosity of the emulated emulsions was back calculated from the measured profiles through the Power-Law model. An estimation of the error on the viscosity measurement is obtained by comparing this value with the desired viscosity, obtaining an error of about 4%.

4.3 Flow Emulator v2

In the second version of the Flow Emulator, the echo signals can be synthetized both off-line, as in the first version, and in real-time. The real-time generation is made possible thanks to the calculation power of a last-generation FPGA. The latter integrates a signal model based on the summation of the contributions of random scatterers, able to reproduce a real-like Doppler signal. The user sets the desired configuration through a graphical interface in Matlab, selecting the geometry of the vessel/pipe, the features of the ultrasound beam, the flow profile and, if desired, it is possible to add source of disturbances like white noise, clutter and the in-depth attenuation.

As in the previous version, the synchronization between the clock of the Flow Emulator and DSUT is mandatory. However, the FPGA structure is changed and this requires some modifications in the synchronization circuit, as seen in the previous chapter.

4.3.1 Doppler Signal Model

In a Doppler analysis a scatterer moving at velocity v is investigated by transmitting ultrasound bursts at frequency F_t at Pulse Repetition Interval (PRI) T_{pri}. The echo produced by the scatterer is acquired at each PRI in the receiver and sampled at $F_c = 1/T_c$ rate. If k is the sample index along the depth (typically referred as "fast time"), and l is the sample index along the PRI sequence (typically referred as "slow time"), the received echo is represented by a 2D matrix of indexes (k, l):

$$S_n(k, l) = A_n \cdot W_n(k, l) \cdot \sin\left[2\pi F_t\left(kT_c + \frac{l v_n T_{pri}}{c}\right)\right] \tag{62}$$

where c is the sound velocity. The coefficient A_n accounts for the backscattering property and is generated randomly with a uniform distribution in the interval 0.5,1. $W_n(k, l)$ represents a 2D tapering window, it is 0 at the external regions of the matrix ($|l| > L_n$, $|k| > K_n$) and reaches its maximum at the center, where $W_n(0,0)$=1. Thus, the matrix $S_n(k, l)$ has non-zero samples placed in $2K_n + 1$ rows, i.e. the depths with $-K_n < k < K_n$ and $2L + 1$ columns, where $-L_n < l < L_n$. In this study we employed the 50% central section of the 1D Blackman window WB_{50}, to compose the 2D window:

$$W_n(k, l) = WB_{50}(k) \cdot WB_{50}(l) \tag{63}$$

The K_n parameter accounts for the extension of the scatterer echo in the fast-time direction, which is directly related to the axial Doppler sample volume [84]. If SV_n is the -6 dB sample volume extension along depths, it can be stated that:

$$2K_n + 1 = \frac{SV_n}{cT_c}\frac{1}{\alpha} \tag{64}$$

The coefficient α represents the window relative -6 dB extension. For example, for the Blackman-derived window (63) α is 0.8. Similarly, the parameter L_n accounts for the echo extension in the slow-time direction. This parameter affects the transit time BW_n/v_n, which is one of the sources of the spectral Doppler broadening (see 1.4.2). The parameter L_n is regulated on the desired -6 dB beam extension BW_n:

$$2L_n + 1 = \frac{BW_n}{v_n \cdot T_{pri}}\frac{1}{\alpha} \tag{65}$$

The coefficient α is the same described above. An example of echo generated for a single scatterer is shown in Fig. 62.

The final signal matrix $M(k,l)$ is composed by summing the contributions of N scatterers:

$$M(k,l) = \sum_{n=0}^{N-1} S_n(k - k_n, l - l_n) \tag{66}$$

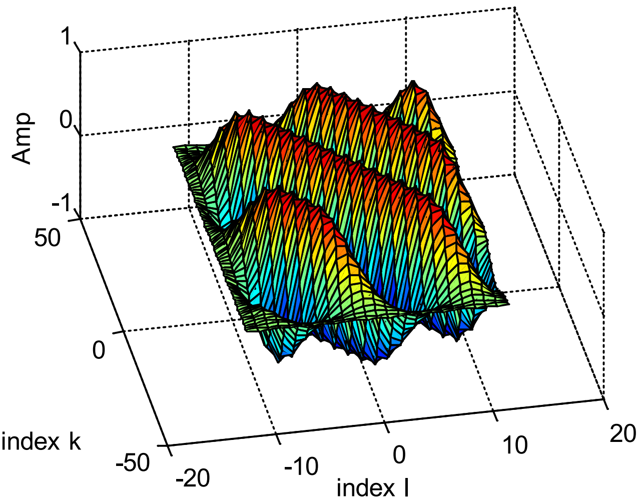


Fig. 62: Echo of a scatterer emulated by (62) with v_n=0.3 m/s, T_c=10 ns, $1/T_{pri}$=6 kHz, F_t=3 MHz, BW_n= SV_n= 1 mm.

Table XIII: Signal model parameters.

Session Parameters	***Description***	***Unit***
$\boldsymbol{c}$	Sound velocity	m/s
$\boldsymbol{T_c}$	Fast-time sampling period	s
$\boldsymbol{T_{pri}}$	Slow-time sampling period	s
$\boldsymbol{F_t}$	Transmission frequency	Hz
Scatterer Parameters		
$\boldsymbol{A_n}$	Backscatterer coefficient	-
$\boldsymbol{v_n}$	Scatterer velocity	m/s
$\boldsymbol{BW_n}$	Beam Width	m
$\boldsymbol{SV_n}$	Sample Volume fast-time extension	m
$\boldsymbol{2K_n+1}$	Fast-time scatterer extension	-
$\boldsymbol{2L_n+1}$	Slow-time scatterer extension	-

The scatterers are located in the random depth-time positions (k_n, l_n), with $0 < n < N-1$. An average of about 10 scatterers in the Doppler sample volume are typically enough to generate a suitable statistics. The desired flow, beam, and noise configuration is emulated by tuning the parameters of each of the scatterers added in (66) in function of their depth and time. For example, a known flow profile along depths is obtained by imposing the suitable v_n to the scatterers that belongs to specific depths.

The parameters of the model are summarized in Table XIII. They are split in “session parameters” that are common to all of the N scatterers generated in the experiments; and “scatterers parameters”, which can differ for every single scatterer, and for this reason have the “n” subscript. The parameters listed in Table XIII are independent, with the only exception of K_n and L_n that originate from other parameters like described by (64) and (65).

4.3.2 Hardware architecture

The architecture of the Flow Emulator and its main features are reported in Fig. 63 and Table XIV respectively. The Flow Emulator is based on a custom electronic board that encases the MitySOM-5CSX-H6-42A-RC commercial System-On-Module (SOM) produced by Critical Link, LLC (Syracuse, NY). The system is connected through the Ethernet link to a host PC where a custom interface, developed in Matlab, runs. The SOM includes most of the high velocity digital electronics required by the project, like, among others, a System on Chip (SoC) FPGA of the Cyclone V family (Intel-Altera, Santa Clara, CA, USA), an

Table XIV: Main features of the Flow Emulator v2.

Parameter	Value
Channel	1
Output voltage	Up to 500 mVpp
Output frequency range	0.1 ÷ 15 MHz
Output burst	Arbitrary waveform
Input TX att.	Up to 100 Vpp
PRI range	0.07 ÷ 10 ms
Sampling Freq.	100 Msps
Resolution	14 bit
FPGA DDR3 size	256 MB
ARM DDR3 size	1 GB

SD card; two DDR SDRAM buffers; a 1 Gb Ethernet controller; power management; several input/output connections. The SoC FPGA integrates a 800 MHz dual-core ARM processor, which directly interfaces to the 1 GB RAM and the SD card for boot. The custom baseboard also includes the devices required by the specific application, like, for example, the 100 MSPS, 14 bit DA converter, followed by the required analog conditioning circuits, and an attenuator which removes the high voltage transmission pulse generated by the DSUT.

4.3.3 FPGA Firmware

The FPGA logic fabric includes several blocks as sketched in Fig. 63. A software memory controller ("DDR Ctrl") connects to the FPGA 256 MB DDR3 bank. Custom hardware blocks as "Scatter generator", "Adder", "DP mem" and "DMA" allow the real-time synthesis as described in 4.3.1. The 256 MB buffer located in the FPGA DDR memory stores the $M(k,l)$ matrix (66). The buffer is managed dynamically: if l_i is the index of the column that holds the data of the next PRI, the buffer stores the PRIs from l_i to l_{i+A}, where A is wide enough to include several scatterers, i.e. $A \gg (2L_n + 1)$ (see Fig. 64). Every PRI, the column l_i is removed from the buffer and sent to the FIFO (Fig. 64-right) while a new empty column l_{i+M+1} is queued in the buffer (Fig. 64-left). The scatter generator accumulates continuously new scatterers to the $M(k,l)$ buffer. For adding a scatterer at position (k_n, l_n), the data block $M(k,l)$ with $k_n - K_n < k < k_n + K_n$; $l_n - L_n < l < l_n + L_n$ is moved from the DDR buffer to the internal dual port memory ("DP mem" in Fig. 64). Here data are read, summed to the $S(k,l)$ produced by the scatter generator, and saved back in the

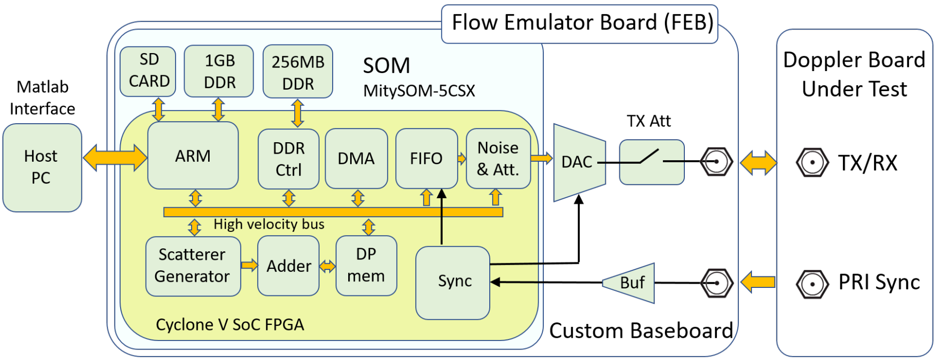


Fig. 63: General architecture of the Flow Emulator v2 board. It is based on a System on Chip (SOC) FPGA that interfaces to a custom ultrasound front-end. The FE is connected with the host PC (left) and the DSUT board (right).

DP memory. Finally, the data are moved in the original position of the $M(k,l)$ buffer in DDR memory. Data moving is performed by custom DMAs (Direct Memory Access). The Scatterer Generator calculates (62) in a 7-stage pipeline and produces a sample per clock cycle at 16 bit resolution. For example, a typical scatterer echo with $L_n = K_n = 16$, composed by 33x33 = 1089 samples, is calculated and summed in the DP memory in 7.26 µs. By including the data block moving, the FE refills the matrix $M(k,l)$ with up to 49M echo samples per second, corresponding to about 45k scatterers/s.

The FIFO memory interfaces a Digital-to-Analog (DA) converter through the ‘Noise & Att.’ block. The latter adds, on the fly, a programmable background

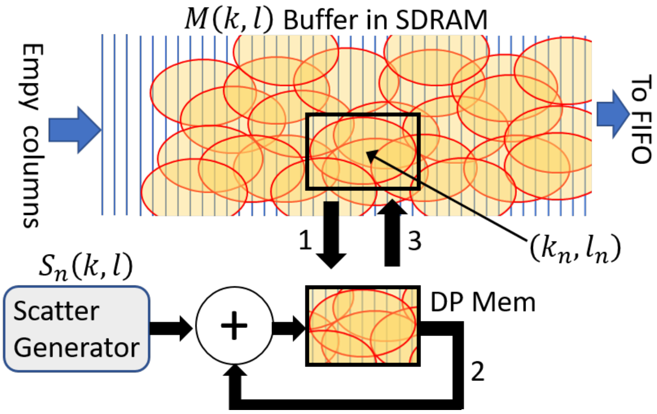


Fig. 64: Synthesis of the Doppler signal by the real-time summation of the contribution of random scatterers.

noise (white noise) and manages the in-depth signal attenuation, which are programmable by the user.

All the FPGA blocks, and the ARM as well, communicate through an internal high velocity bus. DMA processors are employed to quickly move data among peripherals, both to signal generation and data moving from ARM DDR to the FPGA DDR (for example, when the user loads the echo signal samples instead of real-time generation). Finally, the custom "Sync" block, described in Chapter 3, generates the on-board timings and manages the synchronization with the DSUT.

4.3.4 FPGA Resource Usage

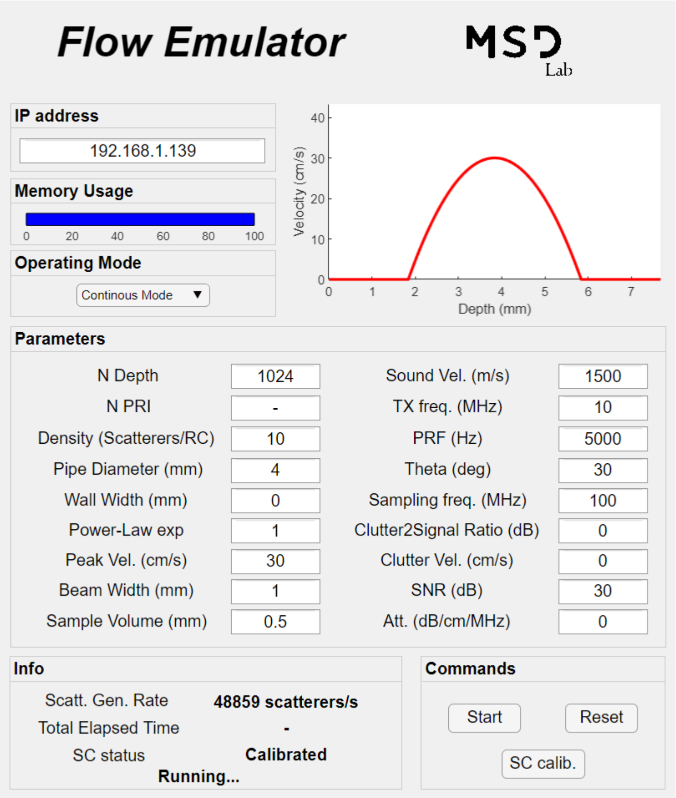
Table XV summarizes the FPGA resources in terms of "Adaptive Logic Modules" (ALMs), memory bits and "Digital Signal Processors" (DSPs) required for the project. Resources are detailed for the main blocks shown in the architecture of the Fig. 63. The bottom of Table XV reports the total (second-last row) and the percentage with respect to the capacity of the employed FPGA. The project reached the time closure with a 150 MHz clock.

Table XV: Cyclone V SoC FPGA resources.

Section	*ALMs*	*Memory bits*	*DSPs*
Scatter Generator	77	15872	8
Adder	113	-	-
DP Mem.	-	65536	
DDR Controller	3609	181264	-
DMA	424	-	-
Noise & Attenuation	28	65536	1
FIFO	-	65536	-
Sync	1276	3666	-
TOT	5527 32.8%	397410 8.1%	9 9%

4.3.5 ARM processor and Matlab GUI

The simple graphical user interface of the Flow Emulator is shown in Fig. 65. The GUI has several panels, where the user can: set the IP address of the FEB; select the operating mode and the emulation parameters; visualize the velocity profile set and the board info ("Memory Usage" and "Info" panels); send commands to the FE. There are three operating modes: Uploading Mode (UM), where the profile is externally generated by the user and uploaded in the FE, similarly to the first version of the emulator; Normal Mode (NM), where the FE generates in real-time a limited number of PRI according to the parameter set by the user; Continuous Mode (CM), where the FE generates the profiles in real-time until the "Reset" command is received. The screenshot of the GUI reported in Fig. 65 shows the FE in CM, where the "NPRI" parameter, the "Memory Usage" and panel and the "Total Elapsed Time" info are not employed, but they are used when the FE is set in NM. The "SC calib." command starts the calibration process of



Flow Emulator

IP address

192.168.1.139

Memory Usage

Operating Mode

Continous Mode ▼

Parameters

N Depth	1024	Sound Vel. (m/s)	1500
N PRI	-	TX freq. (MHz)	10
Density (Scatterers/RC)	10	PRF (Hz)	5000
Pipe Diameter (mm)	4	Theta (deg)	30
Wall Width (mm)	0	Sampling freq. (MHz)	100
Power-Law exp	1	Clutter2Signal Ratio (dB)	0
Peak Vel. (cm/s)	30	Clutter Vel. (cm/s)	0
Beam Width (mm)	1	SNR (dB)	30
Sample Volume (mm)	0.5	Att. (dB/cm/MHz)	0

Info

Scatt. Gen. Rate 48859 scatterers/s

Total Elapsed Time -

SC status Calibrated

Running...

Commands

Fig. 65: Screenshot of the Flow Emulator GUI in Matlab. The user sets the desired scatterer and flow parameters and then starts the real-time signal generations.

the synchronization circuit (“Sync” block in Fig. 63), that is typically performed before each test.

In the on-board signal generation modalities (NM and CM), the user selects the features reported in Table XVI, like, for example, the geometry of the vessel/pipe and the shape of the flow profile $v(r)$. This is selected by choosing the “n” exponent of Power-Law formula (see 2.1.1):

$$v(r) = 1 - \left|\frac{r}{R}\right|^{\left(1+\frac{1}{n}\right)} \tag{67}$$

where R and r are the radius and the distance from the center. The user sets the sample volume geometry (lateral beam width and axial width) and adds, if desired, the emulation of noise and source of disturbances, like in-depth signal attenuation, background white noise, and clutter. Finally, velocity of sound and scatterers density are set. When ready, user commands the start of the signal emulations (“Start” button in the GUI “Commands” panel), which begins immediately.

The ARM processor integrated in the FPGA runs the Linux® operative system, and a custom code, written in C++. This code manages the high-level communication with the GUI that runs on the host PC, and interfaces to the low-level logics integrated in the FPGA fabric. The ARM processor, according to the data and parameters received from the Matlab GUI, directs the real-time generations of the scatterers. It is supported by the “Scatterer Generator” block (Fig. 63), which works as an efficient coprocessor. The ARM processor generates the random scatterer positions, then tunes the parameters that depend on position (like, e.g., velocity, in-depth attenuation - see Table XIII), and commands the coprocessor to generate the samples. For example, for emulating the clutter, it

Table XVI: GUI settings.

Programmable Features	***Description***
Flow Profile	Power law or user-defined
Vessel/pipe geometry	Depth position, Diameter
Beam	Width (*BW*) axial sample volume (*SV*) Transmission frequency (F_t)
Noise and disturbances	In-depth signal attenuation Level of background white noise Amplitude and bandwidth of clutter
General	Sound velocity in medium (*c*) Scatter density (*D*)

places scatterers around the desired wall position. These scatterers will have a coefficient A_n tuned on the desired clutter/signal amplitude ratio (“Clutter2Signal Ratio” in the GUI), and a velocity will be distributed between 0 and a desired high limit value (“Clutter vel.” GUI parameter).

4.3.6 System Performance Evaluation

4.3.6.1 Mathematical Accuracy

The scatterer generator performs the calculations in 16 and 32 bit fixed-point mathematical representation. Mathematical noise is typically produced by the limited dynamics of this format. In this section the mathematical noise produced by the scatter generator was quantified. The calculation chain implemented in the scatterer generator was reproduced in Matlab. A total of 40 matrices, corresponding to 40000 scatterers, were generated both in the FE, $M_C(k,l)$, and in the Matlab chain, $M_R(k,l)$, by employing the same parameters. The matrices were then compared and the Signal-to-Noise (SNR) ratio was calculated according to the metrics:

$$\mathrm{SNR} = 10\mathrm{Log}_{10}\left(\frac{\sum_k \sum_l \left(M_C(k,l)\right)^2}{\sum_k \sum_l \left(M_C(k,l) - M_R(k,l)\right)^2}\right) \tag{68}$$

The SNR averaged over the 40 matrices was 61.4 dB, with a standard deviation of ± 0.8 dB.

4.3.6.2 Real-time Throughput

Like detailed in the description of the signal model, the dimensions of the resolution cell in “pixels”, i.e. points of the matrix, is $\alpha(2K_n + 1) \cdot \alpha(2L_n + 1)$, where α represents the window relative -6 dB extension. If D is the desired density of scatterers per resolution cell, in this area we should locate, on average, the centers of D scatterers. In other words, each of the pixels in the $\alpha(2K_n + 1) \cdot \alpha(2L_n + 1)$ area should be the summation of D contributions in (66). On the other hand, each of the contribution in (66) is composed by $(2K_n + 1) \cdot (2L_n + 1)$ pixels. Thus, for each pixel of the $M(k,l)$ matrix the number of samples that must be summed is:

$$N = D \cdot \frac{(2K_n + 1) \cdot (2L_n + 1)}{\alpha(2K_n + 1) \cdot \alpha(2L_n + 1)} = D \cdot \frac{1}{\alpha^2} \tag{69}$$

This value depends on D and on the tapering window only, and it is independent on any of the configuration parameters.

The scatter generator is able to produce up to 49M sample/s, thus the number of pixels per second, N_p/s, that the system produces in real-time, obtained by (69) is:

$$N_p/s = 49\text{Ms/s} \cdot \frac{\alpha^2}{D} \tag{70}$$

For example, for D = 10 and α = 0.8, we have N_p/s = 3136k. In this condition the system emulates a flow at 512 depths up to a PRF = 1/PRI of 6.125 kHz (512·6125 = 3136k), or 256 depths and PRF up to 12.25 kHz, etc.

4.3.7 Experiments and Results

The following experiments are carried out by connecting the FE to the industrial system (2.2.1) and ULA-OP (2.3.1) research scanner, that were here employed in place of the DSUT. The industrial sensor was connected electrically: the ultrasound transducer was removed, and its TX/RX channel was joined to the FE. The ULA-OP scanner was coupled acoustically: the FE injected the signal in a 7 MHz, 60% bandwidth cylindrical transducer, which was placed transversally in front of the LA533 (Esaote s.p.a., Genoa, Italy) linear array probe, connected to the scanner. Acoustic gel was interposed between the transducer and the probe to grant a suitable coupling. A second connection was used for the PRI synchronism generated by the industrial system and the ULA-OP, like shown in Fig. 63.

Table XVII summarizes the parameters employed in the experiments described in the following sections. The first row reports the paragraph where the experiment is described. A similar configuration is employed in most of the presented experiments, but specific parameters are varied in each experiment to highlight particular FE features. The typical settings employed with the industrial sensor emulates a depth range of 10 mm (768 samples @ 100 Msps) with inside a pipe with an 8 mm diameter. The flow profile was parabolic or smashed with n = 0.3 in (6) and a peak velocity of 0.3 m/s. The transmission frequency was F_t = 5 MHz and the PRF was 6 kHz. The flow was investigated by a 1 mm beam width and SV = 0.5 mm.

The FE, when connected to ULA-OP, was programmed to emulate a 6 mm range (384 samples at 100 MHz) with a 4 mm diameter vessel inside. ULA-OP was set in Doppler mode, with an unsteered Doppler line, dynamic focus with $F\#$ = 1, PRF = 6 kHz and F_t = 7 MHz.

During the experiments, the industrial sensor and the echograph acquired the signal and processed it in real-time through complex demodulation [85], filtering, 128-point FFT (packet size 128) to obtain the spectral profiles [86]. These were

Table XVII: Parameters employed in the experiments.

Experiment	***SNR*** **4.3.7.1**	***Variability*** **4.3.7.2**	***Clutter*** **4.3.7.3**	***Beam Width*** **4.3.7.4**	***Axial SV*** **4.3.7.5**	***In-depth att.*** **4.3.7.6**	***ULA-OP*** **4.3.7.7**
c **(m/s)**	1500	1500	1500	1500	1500	1500	1500
T_c **(ns)**	10	10	10	10	10	10	10
$1/T_{pri}$ **(kHz)**	6	6	1	6	6	6	6
SNR (dB)	10,20 ,30	-	-	-	-	-	-
In-dept Att. (dB/cm)	-	-	-	-	-	25	-
Depths	768	768	768	768	384	768	384
Power Law exp.	1	1	0.1	0.3	0.3	1	1, 0.3
Pipe diam (mm)	8	8	8	8	4	8	4
Clutter bandwidth (Hz)	-	-	45,90 ,180	-	-	-	-
A_n	0.5-1	0.5-1	0.5-1	0.5-1	0.5-1	0.5-1	0.5-1
v_P **(m/s)**	0.3	0.3	0.026	0.3	0.3	0.3	0.24
BW **(mm)**	1	1	1	1,2,3	1	1	1
SV_n (mm)	0.5	0.5	0.5	0.5	0.4, 1.1, 1.8	0.5	0.5
F_t **(MHz)**	5	5	5	5	5	5	7
D **(scatters /resolution cell)**	10	10	10	10	10	10	10

further processed on-board through a modified centroid estimator [87] to achieve the flow velocity. Spectral profiles and flow velocity were downloaded on the host PC and further analyzed in Matlab. Signals with different SNR, flow profiles, clutter features, in-depth attenuation, beam widths, and axial sample volumes, were tested like detailed in the following paragraphs.

4.3.7.1 SNR Test

The FE was connected to the industrial sensor like described in the previous section, and programmed with the parameters reported in Table XVII, second column from left. In this experiment the FE was employed to emulate a parabolic flow profiles with SNR of 10, 20 and 30 dB. Neither the clutter, nor the in-depth attenuation were added. Profiles and velocity data elaborated by the DSUT were

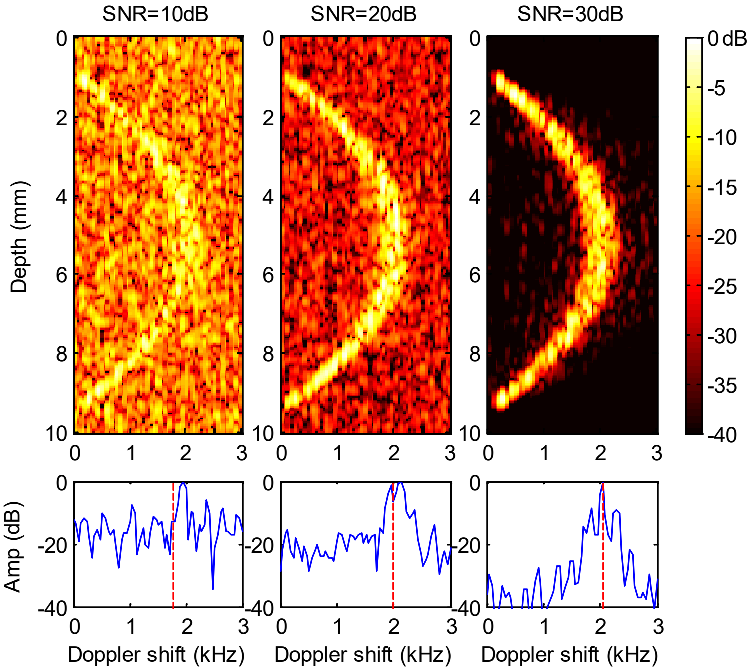


Fig. 66: Spectral profiles (top) and Doppler spectra (bottom) at vessel center (depth =5 mm), detected by the industrial system when the FE emulated a parabolic flow in an 8 mm diameter vessel/pipe with SNR of 10, 20, 30 dB, respectively. Nor temporal neither spatial averaging was applied. The dashed vertical lines on spectra show the detected spectral centroid. Data are represented in a 40 dB dynamics.

downloaded and further processed in Matlab. Fig. 66 shows the profiles measured by the DSUT (top), and the Doppler spectra detected at the vessel center (bottom). Vertical dashed lines on the spectra indicate the centroids. To better highlight the noise signature, neither spatial nor temporal averaging was applied. The velocity detected by the DSUT was 26.3 ± 1.5, 28.9 ± 0.9 and 30.3 ± 0.5 cm/s in the 3 conditions tested. As expected, the accuracy and precision of the velocity measured by the DSUT improved with higher SNR. In this example the FE allowed an accurate quantification of the DSUT performance and its susceptibility to white noise.

4.3.7.2 Doppler Signal Variability

This experiment shows how the FE is able to reproduce the typical variability present in any real ultrasound signal. The same set-up described in the previous experiment was employed, but without adding noise. The flow configuration used in the experiment was reproduced in Field II; and the Doppler spectra where

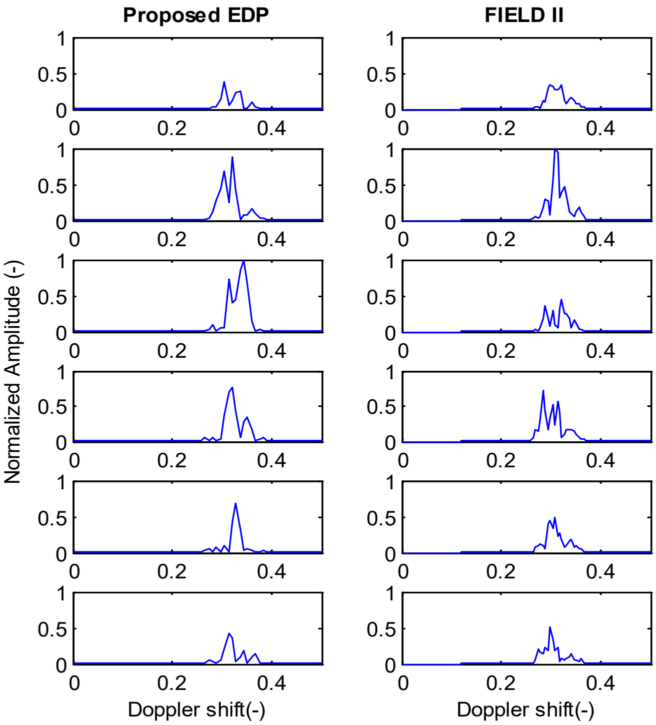


Fig. 67: Comparison between Doppler spectra calculated from the signal generated by the FE (left column) and those calculated in Field II (right column), by simulating the same flow configuration employed in the FE.

calculated from the simulated signal with the same operations applied by the industrial system. Each spectrum was obtained from 128 subsequent PRIs, selected with no overlap. Neither spatial nor temporal averaging was applied. Fig. 67 compares the first 6 spectra generated by the FE (left column) to those obtained by Field II (right column). Although the spectra from the 2 sources are different (they originate from different configurations of random scatterers), the variability generated by the FE is quite similar to that present in a Field II simulations.

4.3.7.3 Emulation of Clutter

The FE was set with the parameters listed in Table XVII, 4th column, and connected to the industrial sensor. A 2.6 cm/s flat flow was emulated, and a clutter signal with an amplitude of 16 dB higher with respect to the flow signal was added. The PRF was reduced to 1 kHz to better highlight the low velocity range. The experiment was repeated with clutter bandwidths that extended from 0 Hz to 45, 90, and 180 Hz. The spectra measured by the sensor, averaged in time, are

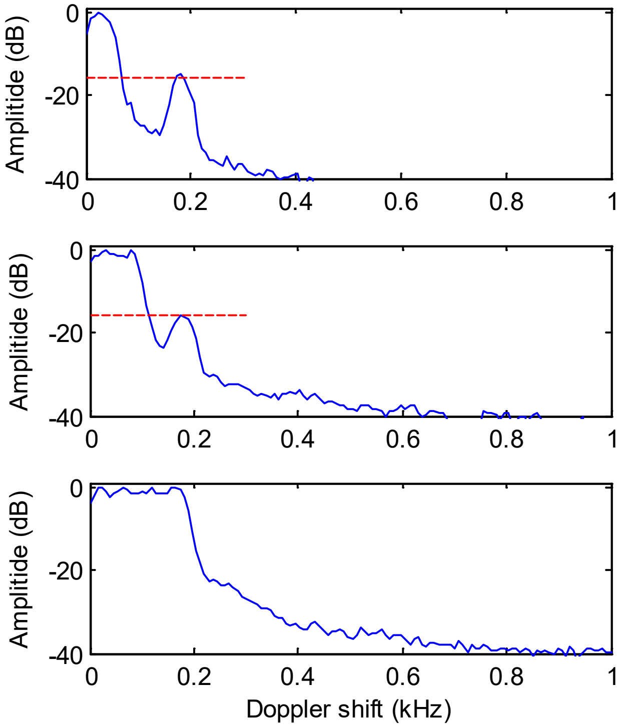


Fig. 68: Examples of clutter signals emulated by the FE and superimposed to a 2.6 cm/s flow. The clutter-to-signal ratio is 16 dB. Top to bottom panels show clutter generated with a bandwidth of 45, 90, and 180 Hz, respectively.

reported in Fig. 68. The flow signal is located at about 170 Hz, and in the first 2 experiments (top and central panels) it is clearly detectable from clutter. In last experiment, clutter hides completely the flow signal.

4.3.7.4 Emulation of Beam Width Extension

The beam width directly affects the spectral broadening, since it modifies the time the scatterers employ to cross the beam [5]. This experiment aims at verifying how the FE emulates this phenomenon.

The FE was connected to the industrial sensor like in previous tests, and set with the scatterer parameters of Table XVII, 5th column. The flow configuration was set as smashed, obtained by the Power-Law (67) with $n = 0.3$. In this experiment the beam width was changed among 1, 2 and 3 mm. Data acquired from the industrial sensor were saved and further analyzed in Matlab. Fig. 69 shows, on top, the spectra profiles obtained for the 3 tested beam widths. Profiles were

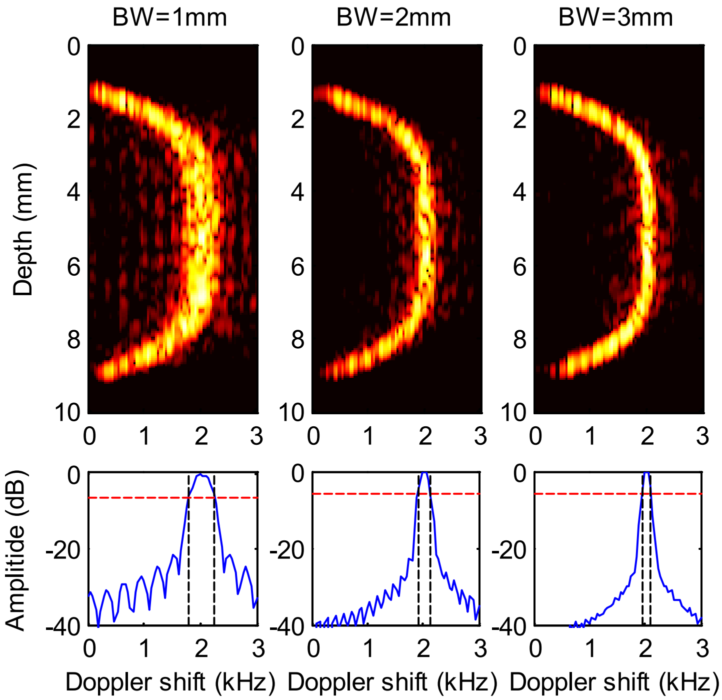


Fig. 69: Doppler spectral profiles (top) and spectra taken at the vessel center at depth = 5mm (bottom) generated by the FE when emulating a beam width (BW) of 1, 2 and 3 mm. Vertical dashed lines in spectra show the -6 dB widths, which are 426, 203 and 128 Hz for beams from 1 to 3 mm. Data are shown over a 40 dB dynamics, flow profiles are generated by power law with n = 0.3.

averaged neither in depth nor in time. As expected, the spectral broadening reduces as long as a wider beam is emulated. Spectra measured in vessel center (depth = 5 mm), averaged in time to reduce the variability, where used to quantify the broadening (see Fig. 69 bottom row). The -6 dB spectral broadening (see dashed vertical lines on spectra) was 426, 203, 128 Hz, respectively.

4.3.7.5 Emulation of Axial Extension of the Sample Volume

This experiment tests how FE emulates the variation of the axial resolution in response to the change of the axial dimension of the of sample volume [84]. The FE was connected like in previous tests, and programmed like reported in Table XVII, 6th column. A relatively small pipe/vessel of 4 mm diameter was chosen to enhance the impact of the axial resolution. The low pass filter applied after the demodulator [85] was set for a cut-off frequency of 1.5 MHz, in order to minimize its effect on the axial resolution. The sample volume extension was set at 0.4, 1.1,

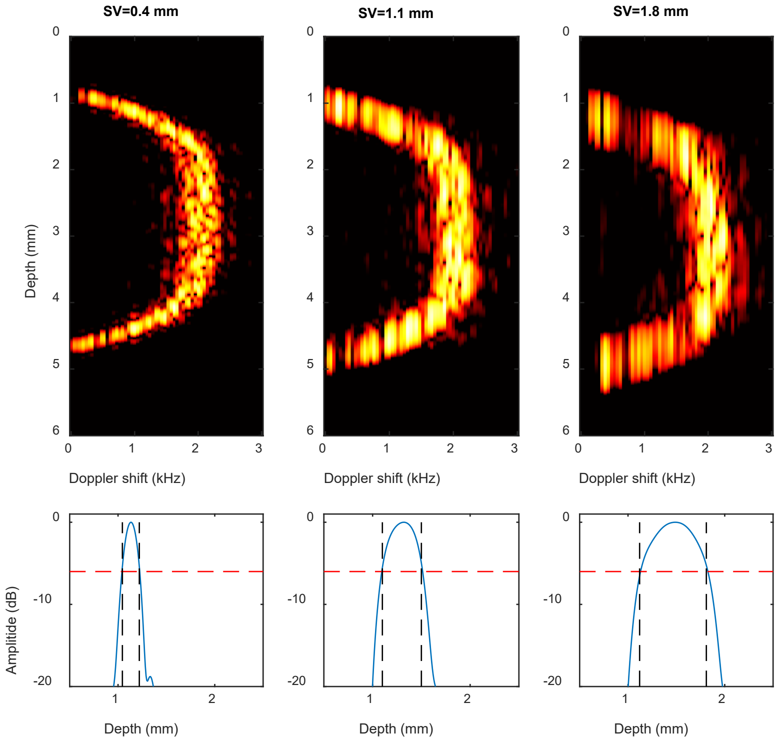


Fig. 70: Spectral profile (top) and axial widths (bottom) measured by the DSUT when the FE emulated a smashed flow in a 4 mm diameter pipe. The axial width refers to Doppler frequency 1kHz.

and 1.8 mm. Top of Fig. 70, from left to right, reports the measured spectral profiles for increasing SV dimensions. The widening of the figure "grain" towards the depth direction is clearly visible. The bottom of the figure shows the signal taken from the profile at frequency f =1 kHz. Horizontal dashed segments mark the -6 dB threshold, vertical segments quantify the lobe extensions in 0.17, 0.40, 0.69 mm.

4.3.7.6 Emulation of In-Depth Attenuation

In this experiment the FE was connected again to the industrial system, and programmed with the scatterer parameters of Table XVII, 7th column. It was set for a parabolic flow profile of 8 mm diameter and 0.3 m/s peak velocity. An in-depth attenuation of 25 dB/cm was imposed to the signal. A relatively high attenuation was used in order to highlight its effect along the 8 mm pipe diameter. The attenuation expected along the diameter was of 25/10*8 = 20 dB. The spectral

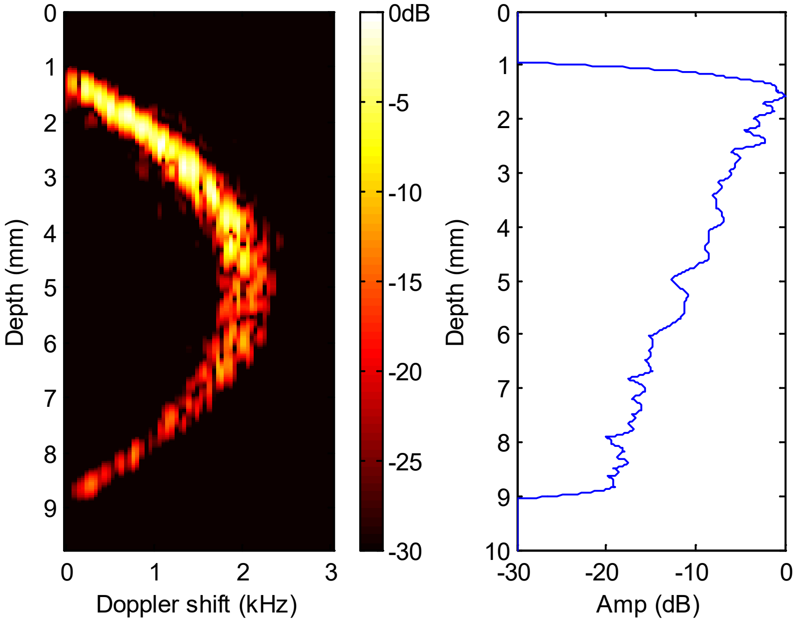


Fig. 71: Spectral profile (left) and power along dept (right) measured by the DSUT when the FE emulated a parabolic flow in an 8 mm diameter pipe with a 25 dB/cm attenuation along depths.

profiles calculated by the DSUT were saved and analyzed in Matlab. Fig. 71 reports an example of the acquired spectral profiles. The color degradation from yellow to red along the increasing depths confirms the signal attenuation. The signal power along depth is reported on the right of Fig. 71. The figure shows a linear (in dB) attenuation of 20 dB along the profile depth-axis, like expected.

4.3.7.7 ULA-OP test

In this experiment the FE was acoustically coupled to the ULA-OP scanner as detailed in the paragraph 4.3.7. The FE was programmed to emulate a parabolic and a smashed flow profile with 0.24 m/s peak velocity and 4 mm pipe diameter. Fig. 72 shows 2 screenshots taken from the real-time display of the scanner, while presenting the parabolic (left) and smashed (right) profiles produced by FE. No averaging was applied in the displayed profiles. The velocity measured by ULA-OP at vessel center was 0.243±0.009 m/s and 0.241±0.009 m/s for parabolic and smashed profiles with an error of about 1.5% and 0.5%, respectively.

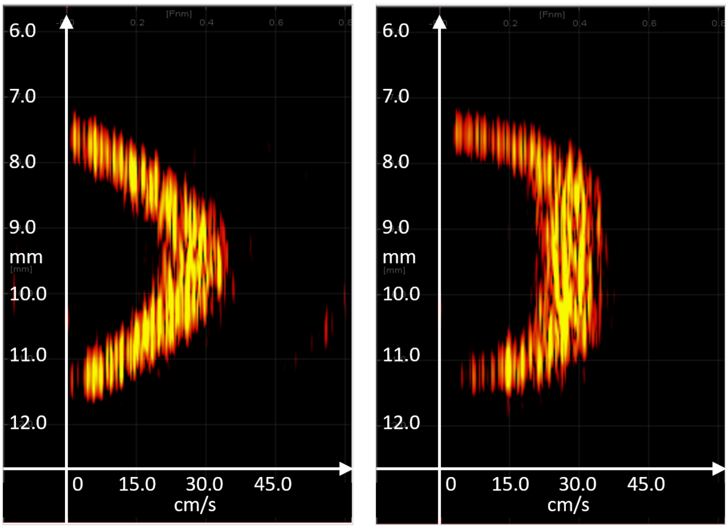


Fig. 72: Screenshot of the display of the ULA-OP scanner. ULA-OP is acoustically coupled to the FE that emulated a parabolic (left) and smashed (right) profiles with 0.24 m/s peak velocity in a 4 mm pipe.

4.3.8 Discussion and Conclusions

The proposed FE represents a tool for the accurate tests of Doppler methods and Doppler electronics systems. It can be exploited during the implementation of novel methods, for the quality monitoring during industrial production of Doppler apparatuses, or for the periodic maintenance and calibration of Doppler instruments and sensors [30]. Its employment is easy and immediate. However, the FE is not intended of substituting computer simulations (like, for example Field II) and hydraulic phantoms in all of their applications, but rather, FE features complementary characteristics that make FE useful where other tools are weak. For example, a morphological hydraulic phantom of the carotid bifurcation is preferable for a qualitative test of a flow imaging method; but FE is better for a quantitative evaluation of the accuracy in velocity measurement of a Doppler method/instrument (see, e.g. experiments 4.3.7.1 or 4.3.7.7).

The proposed FE generates a real-like Doppler signal based on the summation of single scatterers contributions. This approach requires a high calculation power. Nevertheless, the FE generates the signal in real-time thanks to its FPGA accelerator. The emulated signal follows immediately the commands that the user applies in the interface, making the FE an ideal test-bench instrument.

Other EDP boards are described in literature [73]-[78], however the proposed system is by far the most complete and flexible. It produces a real-like signal and arbitrary flow profiles, it emulates the transit time effect [5] and the limited sample volume, disturbances like clutter and in-depth attenuation can be added.

The proposed FE is basically a single channel system. It is ideal to be coupled to single channel DSUTs, like most industrial sensors and some specific biomedical devices are. However, its employment to multi-channel echographs is still effective (see experiment 4.3.7.7), and methods based on the reception of a single Doppler line can be easily tested. The method employed in the FE is scalable. The real time performance depends from the scatter generator, which employs relatively few FPGA resources (see Table XV). Paralleling more generators opens the possibility to implement a multi-channel version of the FE, capable of testing Doppler methods based on multiple lines or plane waves [88][89].

In Doppler analysis the phase coherence among pulses from subsequent PRIs is mandatory, and the residual jitter must be maintained below 100ps rms. In the FE we employed a special resynchronization circuit (see Chapter 3) fed by the PRI sync pulse generates by DSUT, which grants the required performance.

In conclusion, the proposed EDP is not able to completely reproduce the phenomena that occur in a morphological phantom, neither it can compete in several aspects with computer simulations. However, it presents important complementary features that makes it an alternative tool in the hands of scientists and industries to foster the development and dissemination of more accurate Doppler methods [90] and more efficient electronics Doppler systems.

4.3.9 Contributions

The work exposed in this chapter has contributed to the following papers:

Journal paper

- **Russo, Dario**, Stefano Ricci. «Electronic Flow Emulator for Ultrasound Doppler Investigations». *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2020. (Submitted)

Conference proceedings

- **Russo D.**, Ricci S., «Industrial Fluids Electronic Emulator for Rheological Doppler Tests». In *IEEE International Ultrasonics Symposium (IUS),* 2019.
- **Russo, D.**, V. Meacci, and S. Ricci. «Profile Generator for Ultrasound Doppler Systems». In 2018 *New Generation of CAS (NGCAS)*, 33–36, 2018.
- **Russo, Dario**, Valentino Meacci, and Stefano Ricci. «Electronics System for Velocity Profile Emulation». In *Applications in Electronics Pervading Industry, Environment and Society*, pp 101–107. Springer International Publishing, 2019.

Conclusions

This chapter summarizes the contribution of the thesis and discusses possible directions for the future research.

Summary of Contributions

This PhD project introduces an innovative and flexible system for testing both industrial and biomedical Doppler ultrasound methods/systems and a novel clock synchronization method.

The Flow Emulator is able to generate in real-time real-like echo signal of a fluid flowing in a pipe or vessel with a configuration programmed by the user. Unlike classical Doppler tests which are typically carried out by using phantoms and flow-rigs, the Flow Emulator allows to test a system simply connecting the Flow Emulator board to the system under test. The Flow Emulator can be connected acoustically or electronically, i.e. without the transducer, injecting the echo signal in the RX channel of the system under test. As shown in the reported experiments, the Flow Emulator is very flexible and allows to rapidly perform tests by varying parameters such as velocity profile shape and peak, transmission frequency, PRF, ultrasound beam features (beam width and axial dimension of the sample volume), pipe geometry (pipe and wall widths) and, if desired, it is also possible to add sources of noise like clutter, background white noise and in-depth attenuation.

The Flow Emulator and the Doppler system are connected together only by two cables, that are a PRF trigger, and a cable for echo signal. However, the Flow Emulator and the Doppler system are separated systems which work with independent clock. This means that the PRF trigger of the Doppler system is affected by a random noise (frame jitter) when sampled by the Flow Emulator clock, reducing the quality of the Doppler analysis as seen in the reported experiments. The proposed synchronization circuit allows to re-phase the Flow Emulator clock starting from a phase measurement on the PRF trigger without requiring a clock connection between the systems. The phase measurement is performed in the FPGA through a tapped-delay-line, i.e. a delay line followed by registers that freezes the delay between the PRF trigger and the Flow Emulator clock. This phase measurement is then used to dynamically tune the phase of the output clock of an internal FPGA PLL. However, the implementation of the synchronization circuit is not trivial, especially the tapped-delay-line that requires physically placements constraints to guarantee a reliable and reproducible delay line structure. The results show a very good re-phasing capability, reducing the frame jitter from 3 ns rms, unbearable for Doppler analysis, to about 100 ps rms, where the effect of the frame jitter is no longer visible.

Direction of Future Works

Future work will first focus on the implementation of a dual-channels version of the Flow Emulator that will be able to test vector Doppler methods based on dual-line investigations. Then, the capability of the emulator will be extended to 32 or 64 channels, for directly interfacing with a multi-channel echograph. This will require a new baseboard, probably with tens of ΣΔ DA converters and a switch matrix, and a probe connector with a cable to connect the Flow Emulator to the ecographic system.

Bibliography

[1] T. Szabo, “Diagnostic Ultrasound Imaging: Inside Out”, 2nd Edition. Elsevier.

[2] Ricci, S., Matera, R., Tortoli, P. , “An improved Doppler model for obtaining accurate maximum blood velocities”, *Ultrasonics*, 54(7), pp2006-2014, 2014.

[3] M. Mueller, P. O. Brunn, and T. Wunderlich, “New rheometric technique: the gradient-ultrasound pulse Doppler method,” *Appl. Rheol.*, vol. 7, no. 5, pp. 204–210, 1997.

[4] V. L. Newhouse, P. J. Bendick, and W. Varner, “Analysis of Transit Time Effects on Doppler Flow Measurement,” *IEEE Trans. Biomed. Eng.*, vol. BME-23, no. 5, pp. 381–387, Sep. 1976.

[5] V. L. Newhouse, L. W. Varner, and P. J. Bendick, “Geometrical spectrum broadening in ultrasonic Doppler systems,” *IEEE Trans. Biomed. Eng.*, vol. 24, no. 5, pp. 478–480, 1977.

[6] R. V. Edwards, J. C. Angus, M. J. French, and J. W. Dunning, “Spectral Analysis of the Signal from the Laser Doppler Flowmeter: Time-Independent Systems,” *J. Appl. Phys.*, vol. 42, no. 2, pp. 837–850, Feb. 1971.

[7] G. Guidi, C. Licciardello, and S. Falteri, “Intrinsic spectral broadening (ISB) in ultrasound Doppler as a combination of transit time and local geometrical broadening,” *Ultrasound Med. Biol.*, vol. 26, no. 5, pp. 853–862, Jun. 2000.

[8] J. M. Dealy and K. F. Wissbrun, “Melt Rheology and its Role in Plastics Processing: Theory and Applications”. New York, NY: Van Nostrand, 1990.

[9] J. Salazar, J.M. Alava, S.S. Sahi, A. Turo, J.A. Chavez, M.J. Garcia, “Ultrasound measurements for determining rheological properties of flour-water systems”, *IEEE Ultrasonics Symposium* 2002 Proceedings, 2002.

[10] C Létang et al., “Characterization of Wheat-Flour-Water Doughs: A New Method Using Ultrasound” *Ultrasonics* 39 (March 1, 2001): 133–42.

[11] Papaioannou, Theodoros G., and Christodoulos Stefanadis. “Vascular Wall Shear Stress: Basic Principles and Methods.” *Hellenic Journal of Cardiology*: HJC = Hellenike Kardiologike Epitheorese 46, no. 1 (February 2005): 9–15.

[12] R. Kotzé, R. Haldenwang, and P. Slatter, “Rheological characterization of highly concentrated mineral suspensions using ultrasound velocity profiling with combined pressure difference method”, Jan 2008.

[13] I. Roberts, “In-line and online rheology measurement”, Kress-Rogers & Brimelow (eds), *Instrumentation and sensors for the food industry*. 2° ed. Woodhead Publishing Limited, Abington Hall, Cambridge (2001).

[14] R.P. Chhabra, J.F. Richardson, "Non-Newtonian Flow in the Process Industries", 1999.

[15] Malvern Instruments Limited company note, "A Basic Introduction to Rheology", online available https://cdn.technologynetworks.com/TN/Resources/PDF/WP160620BasicIntroRheology.pdf

[16] H.A. Barnes, J.F. Hutton, K. Walters, “An introduction to Rheology”, 1993.

[17] Gan, Yong X. “Continuum Mechanics - Progress in Fundamentals and Engineering Applications”, 2012.

[18] Christopher W. Macosko, “Rheology: Principles, Measurements, and Applications”. 1994.

[19] “Measurement, Instrumentation, and Sensors Handbook, Second Edition: Spatial, Mechanical, Thermal, and Radiation Measurement,” CRC Press, 29-Jan-2014

[20] Y. Takeda, “Development of an ultrasound velocity profile monitor,” Nucl. Eng. Des., vol. 126, no. 2, pp. 277–284, Apr. 1991.

[21] T. Wunderlich and P. O. Brunn, “Ultrasound pulse Doppler method as a viscometer for process monitoring,” *Flow Meas. Instrum.*, vol. 10, no. 4, pp. 201–205, Dec. 1999.

[22] J. Wiklund, I. Shahram, and M. Stading, “Methodology for in-line rheology by ultrasound Doppler velocity profiling and pressure difference techniques,” *Chem. Eng. Sci.*, vol. 62, no. 16, pp. 4277–4293, Aug. 2007.

[23] J. Wiklund, R. Kotzé, R. Haldenwang, and M. Stading, “Development of an industrial UVP+PD based rheometer - optimisation of UVP system and transducer technology,” presented at the *8th International Symposium on Ultrasonic Doppler Methods for Fluid Mechanics and Fluid Engineering*, 2012, pp. 49–52.

[24] J. Wiklund et al., “In-Line Ultrasound based Rheometry of industrial and model suspensions flowing through pipes,” in ResearchGate, 2002.

[25] B. Ouriev and E. J. Windhab, “Rheological study of concentrated suspensions in pressure-driven shear flow using a novel in-line ultrasound Doppler method,” *Exp. Fluids*, vol. 32, no. 2, pp. 204–211.

[26] J. Wiklund et al., “Flow-VizTM–A fully integrated and commercial in-line fluid characterization system for industrial applications,” in Proceedings of the *9th International Symposium on Ultrasonic Doppler Methods for Fluid Mechanics and Fluid Engineering*, 2014, p. 105.

[27] J. Wiklund and M. Stading, “Application of in-line ultrasound Doppler-based UVP-PD rheometry method to concentrated model and industrial suspensions,” *Flow Meas. Instrum.*, vol. 19, no. 3–4, pp. 171–179, 2008.

[28] Meacci, V., Ricci, S., Wiklund, J., Birkhofer, B., Kotze, R.: Flow-Viz - An integrated digital in-line fluid characterization system for industrial applications. *2016 IEEE Sensors Applications Symposium (SAS)* Proceedings, pp. 1–6 (2016).

[29] R. Kotzé, J. Wiklund, and R. Haldenwang, "Optimisation of Pulsed Ultrasonic Velocimetry system and transducer technology for industrial applications," *Ultrasonics*, vol. 53, no. 2, pp. 459–469, Feb. 2013.

[30] R. Kotzé, S. Ricci, B. Birkhofer, and J. Wiklund, "Performance tests of a new non-invasive sensor unit and ultrasound electronics," *Flow Meas. Instrum.*, vol. 48, pp. 104–111, Apr. 2016.

[31] B. Birkhofer, A. Debacker, S. Russo, S. Ricci, and D. Lootens, "In-line rheometry based on ultrasonic velocity profiles: comparison of data processing methods," vol. 22, no. 4, 2012.

[32] S. Ricci, M. Cinthio, M. Lenge, R. Matera, J. Albinsson, P. Tortoli, "Volume Flow Assessment through Simultaneous B-Mode and Multigate Doppler", *Ultrasonic Symposium (IUS)*, 2012 IEEE International, 2012, pp. 1588-1591.

[33] P. Tortoli, L. Bassi, E. Boni, A. Dallai, F. Guidi, and S. Ricci, "ULA-OP: an advanced open platform for ultrasound research," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 56, no. 10, pp. 2207–2216, Oct. 2009.

[34] E. Boni *et al.*, "ULA-OP 256: A 256-Channel Open Scanner for Development and Real-Time Implementation of New Ultrasound Methods," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 63, no. 10, pp. 1488–1495, Oct. 2016.

[35] Cyclone V Device Handbook, CV-5V2, Altera-Intel 2020. [Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-v/cv_5v2.pdf

[36] M. Pieraccini, L. Miccinesi, "Ground-based radar interferometry: A bibliographic review", *Remote Sens.*, 11(9): 1029, 2019.

[37] D.H. Evans, W.N. McDicken. Doppler ultrasound Physics, instrumentation and signal processing. Chichester, UK: Wiley, 2000, ISBN: 978-0471970019.

[38] A.N. Kalashnikov, R.E. Challis, M.E. Unwin, A.K. Holmes, "Effects of frame jitter in data acquisition systems", *IEEE Trans. Instrum. Meas.*, 54(6): 2177 – 2183, 2005.

[39] Galton, I., and C. Weltin-Wu. "Understanding Phase Error and Jitter: Definitions, Implications, Simulations, and Measurement." *IEEE Trans. Circuits Syst*. I-Regul. Pap., 66(1): 1–19, 2019.

[40] F. Alessio, R. Jacobsson, "Timing and Fast Control for the Upgraded Readout Architecture of the LHCb Experiment at CERN", *IEEE Trans. Nucl. Sci.*, 60(5): 3438 – 3445, 2013.

[41] M. Rizzi, M. Lipinski, P. Ferrari, S. Rinaldi, A. Flammini, "White Rabbit Clock Synchronization: Ultimate Limits on Close-In Phase Noise and Short-Term Stability Due to FPGA Implementation", *IEEE Trans. Ultrason., Ferroelect., Freq. Contr*, 65(9): 1726 - 1737, 2018.

[42] L. Petrusca, F. Varray, R. Souchon, A. Bernard, J.Y. Chapelon, H. Liebgott, W. A. N'Djin, M. Viallon, "Fast Volumetric Ultrasound B-Mode and Doppler Imaging with a New High-Channels Density Platform for Advanced 4D Cardiac Imaging/Therapy", *Appl. Sci.*, 8(2): 200, 2018.

[43] D. Posada, J. Poree, A. Pellissier, B. Chayer, F. Tournoux, G. Cloutier, D. Garcia, "Staggered Multiple-PRF Ultrafast Color Doppler", *IEEE Trans. Med. Imaging.*, 35(6):1510–1521, 2016.

[44] S. Ricci, D. Vilkomerson, R. Matera, P. Tortoli, "Accurate Blood Peak Velocity Estimation Using Spectral Models and Vector Doppler", *IEEE Trans. Ultrason., Ferroelect., Freq. Contr*, 62(4):686-696, 2015.

[45] G.W. Roberts, M. Ali-Bakhshian, "A Brief Introduction to Time-to-Digital and Digital-to-Time Converters", *IEEE Trans. Circuits Syst.* II-Express Briefs, 57(3): 153 - 157 , 2010.

[46] B. Van Bockel, J. Prinzie, P. Leroux, "Radiation Assessment of a 15.6ps Single-Shot Time-to-Digital Converter in Terms of TID", *Electronics*, 8(5), 558, 2019.

[47] M. Zhang, H. Wang, H. Qin, W. Zhao, Y. Liu, "Phase Difference Measurement Method Based on Progressive Phase Shift", *Electronics*, 7(6):86, 2018.

[48] F. Dadouche, T. Turko, W. Uhring, I. Malass, N. Dumas, J.P. Le Normand, "New Design-methodology of High-performance TDC on a Low Cost FPGA Targets", *Sensors & Transducers Journal*, 193(10);123-134, 2015.

[49] M. Zhang, H. Wang, Y. Liu, "A 7.4 ps FPGA-Based TDC with a 1024-Unit Measurement Matrix", *Sensors*, 17(4), 865, 2017.

[50] S. Kumar, M. Suman, K. Baishnab, "A novel approach to thermometer-to-binary encoder of flash adcs-bubble error correction circuit", *2nd International Conference on Devices Circuits and Systems (ICDCS)*, 2014, pp. 1-6, 2014.

[51] J. Wu, "Several Key Issues on Implementing Delay Line Based TDCs Using FPGAs", *IEEE Trans. Nucl. Sci.*, 57(3):1543-1548, 2010.

[52] L. Zhao, X. Hu, S.Liu, J. Wang, Q. Shen, H. Fan, Q. An, "The Design of a 16-Channel 15 ps TDC Implemented in a 65 nm FPGA", *IEEE Trans. Nucl. Sci.*, 60(5): 3532 - 3536, 2013.

[53] F. Pepe, P. Andreani, "An Accurate Analysis of Phase Noise in CMOS Ring Oscillators", *IEEE Trans. Circuits Syst.* II-Express Briefs, in print, 2018.

[54] Cyclone III Device Handbook, CIII 5V1-4.2, Altera Corp, 2012. [Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyc3/cyclone3_handbook.pdf

[55] Quartus II Handbook Version 13.1, [Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/qts/archives/quartusii_handbook_archive_131.pdf

[56] V. H. Bui, S. Beak, S. Choi, J. Seon, T. T. Jeong, “Thermometer-to-binary Encoder with Bubble Error Correction (BEC) Circuit for Flash Analog-to-Digital Converter (FADC)”, *IEEE International Conference on Consumer Electronics*, 2010.

[57] Cyclone V Device Overview, [Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-v/cv_51001.pdf

[58] Wang, Yonggang, e Chong Liu. «A 4.2 ps Time-Interval RMS Resolution Time-to-Digital Converter Using a Bin Decimation Method in an UltraScale FPGA». *IEEE Transactions on Nuclear Science* 63, n. 5 (October 2016): 2632–38.

[59] Xia, Haojie, Guiping Cao, and Ning Dong. “A 6.6 Ps RMS Resolution Time-to-Digital Converter Using Interleaved Sampling Method in a 28 Nm FPGA.” *Review of Scientific Instruments* 90 (April 1, 2019): 044706.

[60] Wang, Yonggang, Jie Kuang, Chong Liu, and Qiang Cao. “A 3.9-Ps RMS Precision Time-to-Digital Converter Using Ones-Counter Encoding Scheme in a Kintex-7 FPGA.” *IEEE Transactions on Nuclear Science* 64, no. 10 (October 2017): 2713–18.

[61] Wang, Yonggang, and Chong Liu. “A 3.9 Ps Time-Interval RMS Precision Time-to-Digital Converter Using a Dual-Sampling Method in an UltraScale FPGA.” *IEEE Transactions on Nuclear Science* 63, no. 5 (October 2016): 2617–21.

[62] S. Ricci, M. Meacci. B. Birkhofer, J. Wiklund. “FPGA-based System for In-Line Measurement of Velocity Profiles of Fluids in Industrial Pipe Flow”, *IEEE Trans. Ind. Electron.*, 64(5):3997 - 4005, 2017.

[63] Tu, T. Shen, H. Zhang, M Li, “Two New Sliding DTFT Algorithms for Phase Difference Measurement Based on a New Kind of Windows”, *Meas. Sci. Rev.*, 14(6):350:356, 2014.

[64] S. Tancock, E. Arabul, N. Dahnoun,” A Review of New Time-to-Digital Conversion Techniques”, *IEEE Trans. Instrum. Meas.*, 68(10): 3406 - 3417, 2019.

[65] D. Russo, V. Meacci, S. Ricci, “Profile Generator for Ultrasound Doppler Systems*”, Proc. of New Generation of Circuits and Systems Conference*, Malta, pp. 33-36, November 2018.

[66] J. E. Browne, “A review of Doppler ultrasound quality assurance protocols and test devices”, *Phys. Medica*, 30(7):742-751, 2014.

[67] INTERNATIONAL STANDARD IEC 61685, “Ultrasonics—Flow measurement systems—Flow Test Object”, Geneva, Switzerland, 2001.

[68] C. K. Ho, A. J. Y. Chee, B. Y. S. Yiu, A. C. O. Tsang, K. W. Chow and A. C. H. Yu, "Wall-Less Flow Phantoms With Tortuous Vascular Geometries:

Design Principles and a Patient-Specific Model Fabrication Example," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 64(1): 25-38, 2017.

[69] T.L. Poepping, H.N. Nikolov, R.N. Rankin, M. Lee, D.W. Holdsworth, “An in vitro system for Doppler ultrasound flow studies in the stenosed carotid artery bifurcation”, *Ultrasound Med. Biol.,* 28(4):495-506, 2002.

[70] M. Zauli, C. Corsi and L. De Marchi, "Design and Prototype Development of a Low-Cost Blood Flow Simulator for Vascular Phantoms," *Computing in Cardiology (CinC)*, Singapore, 2019.

[71] X. Zhou, D. A. Kenwright, S. Wang, J. A. Hossack and P. R. Hoskins, "Fabrication of Two Flow Phantoms for Doppler Ultrasound Imaging," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*., 64(1):53-65, 2017.

[72] K. V. Ramnarine, D. K. Nassiri, P. R. Hoskins, J. Lubbers. “Validation of a new blood- mimicking fluid for use in Doppler flow test objects.” *Ultrasound Med. Biol.,*24:451–459, 1998.

[73] J A Evans, R Price, F Luhana, “A novel testing device for Doppler ultrasound equipmentnt”, *Phys. Med. Biol.*, 34, 1701, 1989.

[74] C. A. Bastos, P. J. Fish, “A Doppler signal simulator”, Clinical physics and physiological measurement, 12(2):177-183, 1991.

[75] M. J. Lunt, R. Anderson, “Measurement of Doppler gate length using signal reinjection”, *Phys. Med. Biol.*, 38(11):1631-1636, 1993.

[76] A. P.G. Hoeks, M. A. P. Boulanger, P. J. Brands, “Test signal injection for Doppler systems”, *European Journal of Ultrasound* 6(3) 203–212, 1997.

[77] S. F. Li, P. R. Hoskins, T. Anderson, W. N. McDicken, “An acoustic injection test object for colour flow imaging systems”, *Ultrasound Med. Biol.*, 24(1):161-1644, 1998.

[78] J. Gittins, K. Martin, “The Leicester Doppler phantom - a digital electronic phantom for ultrasound pulsed Doppler system testing”, *Ultrasound Med. Biol.*, 36(4):647–655, 2010.

[79] P. R. Hoskins, “Simulation and Validation of Arterial Ultrasound Imaging and Blood Flow”, *Ultrasound Med. Biol.*, 34(5):693-717, 2008.

[80] J. A. Jensen and N. B. Svendsen, “Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers,” *IEEE Trans. Ultrason. Ferroelect.. Freq. Contr*., 39(2):262–267, 1992.

[81] J.A. Jensen: “Field: A Program for Simulating Ultrasound Systems”, *Med. & Biol. Engin. & Comp.*, 34(1):351-353, 1996.

[82] S. Ricci, A. Swillens, A. Ramalli, P. Segers, P. Tortoli, “Wall shear rate measurement: Validation of a new method through Multiphysics simulations”, *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, 64(1), pp. 66-77, 2017.

[83] T.A. Kowalewski: “Velocity profiles of suspension flowing through a tube”. Archiv. Mech., 32(6), pp. 857-865, 1980.

[84] D. H. Evans, “Doppler Ultrasound: Physics, Instrumentation, and Clinical Applications”, New York: John Wiley & Sons, 2007.

[85] S. Ricci, V. Meacci, “Data-Adaptive Coherent Demodulator for High Dynamics Pulse-Wave Ultrasound Applications”, *Electronics*, 7(12), 434; 2018.

[86] P. Tortoli, F. Guidi, G. Guidi, C. Atzeni, “Spectral velocity profiles for detailed ultrasound flow analysis”, *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 43(4):654-659, 1996.

[87] S. Ricci, V. Meacci, “FPGA-Based Doppler Frequency Estimator for Real-Time Velocimetry”, *Electronics,* 9(3), 456, 2020.

[88] S. Ricci, A. Ramalli, L. Bassi, E. Boni, P. Tortoli, “Real-Time Blood Velocity Vector Measurement over a 2D Region”, *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 65(2):201-209, 2018.

[89] J.A. Jensen, S.I. Nikolov, A.C.H. Yu, D. Garcia, “Ultrasound Vector Flow Imaging-Part II: Parallel Systems”, *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 63(11): 722-1732, 2016.

[90] Walker, A.R., Uejima, T., Prinz, C., Voigt, J.-U., Fraser, A.G., “Inaccuracies in Measuring Velocities and Timing of Flow and Tissue Motion Using High-End Ultrasound Systems”, *Ultrasound Med. Biol.*, 45(6):1446-1454, 2019.