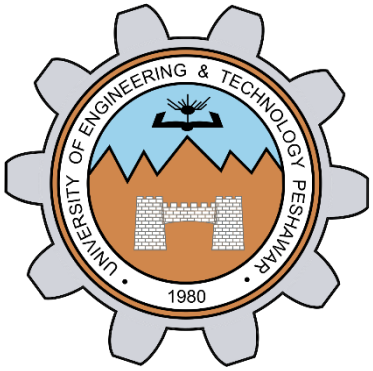
RFSoC Based Digitizing Receiver for Parametric Estimation of High Frequency Signals



Submitted By

Group No 08	
Name of Student	**Registration Numbers**
Alishbah Shafique	19abelt0835
Imran Ali	19abelt0844

Supervisor

Engr Mehmoona Gul

Lecturer

DEPARTMENT OF ELECTRONIC ENGINEERING

UNIVERSITY OF ENGINEERING AND TECHNOLOGY

ABBOTTABAD

September,2023

RFSoC Based Digitizing Receiver for Parametric Estimation of High Frequency Signals

Submitted By

Name of Student	Registration Numbers
Alishbah Shafique	19abelt0835
Imran Ali	19abelt0844

A report submitted in partial fulfillment of the requirements for the degree of B.Sc. Electronic Engineering

Thesis Supervisor:

Engr Mehmoona Gul

Lecturer, Electronics Department

Head of Department Signature: _____

External Examiner Signature: _____

Thesis Supervisor Signature: _____

DEPARTMENT OF ELECTRONIC ENGINEERING

UNIVERSITY OF ENGINEERING AND TECHNOLOGY

ABBOTTABAD CAMPUS

September, 2023

DEDICATION

I dedicate my work to my parent who supports me in throughout my educational journey, without their supports this achievement was not possible. I am extending my dedication to my final year project head Dr. Anees Ullah Assistant Professor, University of Engineering and technology Peshawar, Abbottabad Campus and project Supervisor Engr Mehmoona Gul . I am very thankful to my respected supervisor who motivates and guide me in my final year project and with the help of my respected supervisor I have complete my final year project and achieved the required millstones.

ACKNOWLEDGEMENTS

ALISHBAH SHAFIQUE

First and foremost, I want to thank Allah for giving us the skills and knowledge needed to finish this project . I would like extend my gratitude to my parent who support me throughout my journey, without their support this millstones was not possible. I would like to extend my gratitude to all my teachers who help and support me throughout my journey. Special Thanks to my supervisors, Dr. Anees Ullah and Engr Mehmoona Gul for providing guidance and feedback throughout this project. Without his support this project was impossible.

IMRAN ALI

I would like to thank my supervisors, Professor Dr. Anees Ullah and Engr Mehmoona Gul for providing the inspiration for this project and for the innumerable suggestions he has given me through the course of the project. Also, I would like to thank Dr. Anees Ullah and Engr Mehmoona Gul for their countless advices and hours spent on this project. I owe special gratitude to my family for teaching me not to avoid challenges. Without their support I would have not been able to pursue my dreams and to be the person I am today. Last, but not least I want to thank to my friends for their unwavering support and encouragement during my years at UET Peshawar Abbottabad Campus .

ABSTRACT

Radar, communication, and biological imaging are just a few of the many industries and applications that use high frequency signals. Due to their wide bandwidth, high resolution, and low power needs, processing and interpreting these signals might be difficult. The goal of this project is to create a brand-new gadget that can digitize high frequency signals and estimate their parameters on a single chip. The system's design is based on RFSOC technology, which integrates analog and digital parts into a single chip to minimize system size, cost, and power consumption. The gadget is connected to the ZCU111 RFSOC board, which offers a versatile and programmable testing and evaluation platform. To design and improve the device's performance and accuracy, the project will make use of a variety of techniques and approaches, including signal processing, machine learning, and optimization. A high-quality digitizing receiver with a high dynamic range, low noise, and quick processing speed is what the project hopes to achieve. The project will also illustrate the device's possible uses and advantages in various contexts and situations.

Radio astronomy can benefit greatly from the implementation of a whole next-generation signal processing system on a single board using radio frequency system on chip (RFSOC). On the ZCU111 board, we created a pulsar digital backend system. In the Xilinx ZCU111 chip, the system implements digitization, channelization, correlation, and high-speed data transmission using RFSOC technology. With a maximum sampling rate of 4.096 GSPS, we assessed the performance of the eight 16-bit RF-ADCs built into the RFSOC. In our system's design, the RF-ADC sampling frequency, channel bandwidth, and time resolution may be dynamically changed. We installed the RFSOC board on the Nanshan 26 m radio telescope and examined the pulsar signal with a frequency resolution of 1MHz to test the system's performance.

KEY WORDS

Rfsoc, Zcu11board, Socblockset, Transmit, Dac, Adc, Dsp, Luts, Corrtex, Matlab, Vivado, Xilinx, Processor Logic, Cpu, Control Unit, Simulink, Electronic Warfare, Radar System, Digitizig Receiver, Parameters: Pulse Width, Frequency, Time Of Arrival, Time Of Departure, Fft, Amplitude, Architecture, Direct Rf, Antenna, Sine Wave, Pulse Train, Fpga, Gsps, Msps.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Meaning
RFSOC	Radio Frequency System On Chip
ZCU111	Zync Ultra Scale
ADC	Analog To Digital Converter
DAC	Digital To Analog Converter
FPGA	Field Programmable Gate Array
DSP	Digital Signal Processing
ASIC	Application Specific Integrated Circuit
ARM	Advance RISC Mahine
AFE	Analog Front End
LUT	Look Up Table
IP	Intellectual Property
RFDC	Radio Frequency Data Converter
LNA	Low Noise Amplifier
CPU	Central Processing Unit
DDR4	Double Data Rate
AXi-4	Advance Extensible Interfaces

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION AND RELATED LITERATURE

1.1 Motivation

A radio telescope system's essential digital backend, which is primarily in charge of digitization, signal processing, and high-speed data transfer. The original analog signal systems have been replaced by digital signal systems in many facilities after years of development, and pulsar backends have been built in line with the trend of increased integration, high-speed digitization, and general-purpose devices. ADCs and ASICs were frequently used in astronomical digital backend system designs decades ago. A Field Programmable Gate Array (FPGA), which offers strong processing capability in signal processing, has essentially superseded ASIC in contemporary systems. Since the FPGA and CPU cores are now integrated into one system on chip (SoC), software control of the programmable logic may be realized much more readily than in earlier FPGA system development where an isolated CPU was needed to manage the FPGA. High-speed ADCs are still separate gadgets, though. With the quick advancement of communication technology, Xilinx first introduces radio frequency (RF) SoC by combining high-speed ADC, DAC, CPU cores, and FPGA on the same chip. The Radio Frequency System on Chip (RFSoC) combines RF data converters with programmable logic, a microcontroller, and essential device tasks such real-time signal processing, digitalization, high-speed interface, and software control. The RFSoC is highly integrated, uses little power, and can substantially simplify the design and implementation of an astronomical backend system while also lowering the expenses associated with hardware development.

1.2 RFSOC Overview:

As the hardware used in this project, the Xilinx ZCU111 RFSOC development board's architecture and capabilities should be discussed. Fig. 6 depicts the RFSoC device architecture from a high level. The gadget includes high-speed RF data converters, FPGAs, ARM CPUs, and high throughput SD-FEC blocks. The Xilinx ZCU111 RFSoC's architecture is described in the following general terms:

1.2.1 Processing System (PS):

A quad-core ARM Cortex-A53 application processor and a dual-core ARM Cortex-R5 real-time processor are both found on the ZCU111 RFSoC. While the real-time processor offers predictable and low-latency processing for applications like control and communication, the application processor offers general-purpose processing capabilities.

1.2.2 Programmable Logic (PL):

A field-programmable gate array (FPGA) fabric that can be programmed to build unique logic designs is also included in the ZCU111 RFSoC. A variety of different digital signal processing operations can be implemented using the programmable logic, digital signal processing (DSP), and memory blocks found in the FPGA fabric.

1.2.3 RF Data Converter:

A dual-channel 12-bit analog-to-digital converter (ADC) and a dual-channel 14-bit digital-to-analog converter (DAC) are both included in the ZCU111 RFSoC. The RFSoC can interact with RF signals in the frequency range of DC to 6 GHz thanks to these converters.

1.2.4 RF Data Transceiver:

The RF data transceiver, which has a number of signal conditioning and processing blocks to allow the RFSoC to be tuned for a wide band width signal (order of GHz), is combined with the RF data converter.

1.2.5 Ethernet and PCIe:

Two 10 Gigabit Ethernet ports and a PCIe Gen3 x16 interface are also included in the ZCU111 RFSoC. High-speed data transport to and from the RFSoC is made possible by these connections.

1.2.6 Memory:

The ZCU111 RFSoC has a variety of memory resources, including QSPI flash memory for system booting and configuration data storage and DDR4 memory for the processing system and programmable logic.

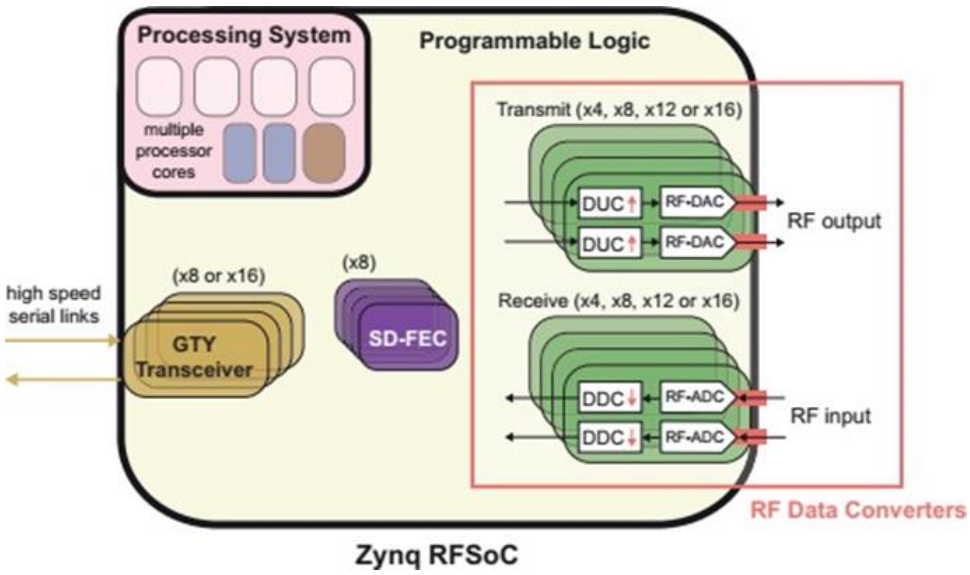


Figure 1.2:ZYNQ RFSOC Architecture.

With its mix of processing power, FPGA programmability, and RF signal interface capabilities, the Xilinx ZCU111 RFSoC offers a strong foundation for creating and testing high-performance RF applications.

1.3 RFSOC-Based Backend Design:

1.3.1 Hardware Platform:

The Xilinx ZCU111 board is used to construct a pulsar digital backend system. The motherboard's primary processing unit is the ZU28DR processor, a first-generation RFSoC device from Xilinx. The ZU28DR contains eight 12 bit ADCs (4.096 GSPS), eight 14 bit DACs (6.554 GSPS), and eight SD-FECs. For the production of signal PS, ZCU111 provides exceptionally powerful computer resources. The XCZU28DR2FFVG1517E RFSoC, four 4G DDR4 components, four SFP/SFP28 modules, high speed network connectors, PMOD connectors, and FMC RFDC connectors are the essential components.The RFSoC device has four RF-ADC tiles, each of which has two RF-ADCs. The RFDC can be adjusted independently and each 12 bit can be changed depending on the input signal ,data stream is aligned with the most significant bits of 16 bit samples in the RF-ADC core. We have implemented a real-time monitoring program, which can display the signal acquisition status of eight RFADC channels on the ZCU111 board, and developed a user friendly graphical user interface (GUI). We enter the bit file name produced by the firmware design

along with the RFSoC board IP address.Then, when clicked, the corresponding button for the RF-ADC channel needs to be monitored, and it displays the root mean square (rms), ADC histogram, and the time and frequency domain of the digitized signal in real-time. Figure 8 is an example of a monitor display plot for RF-ADC channel 01, with tone frequency set to 100 MHz Before the signal entered the RF-ADCs, a tone was produced by the signal generator (SMA110B) and noise was added by the noise source (NF-1000).There is a power difference between low and high frequencies, and the output bandwidth of noise source is not flat. A hardware filter provides a bandwidth.

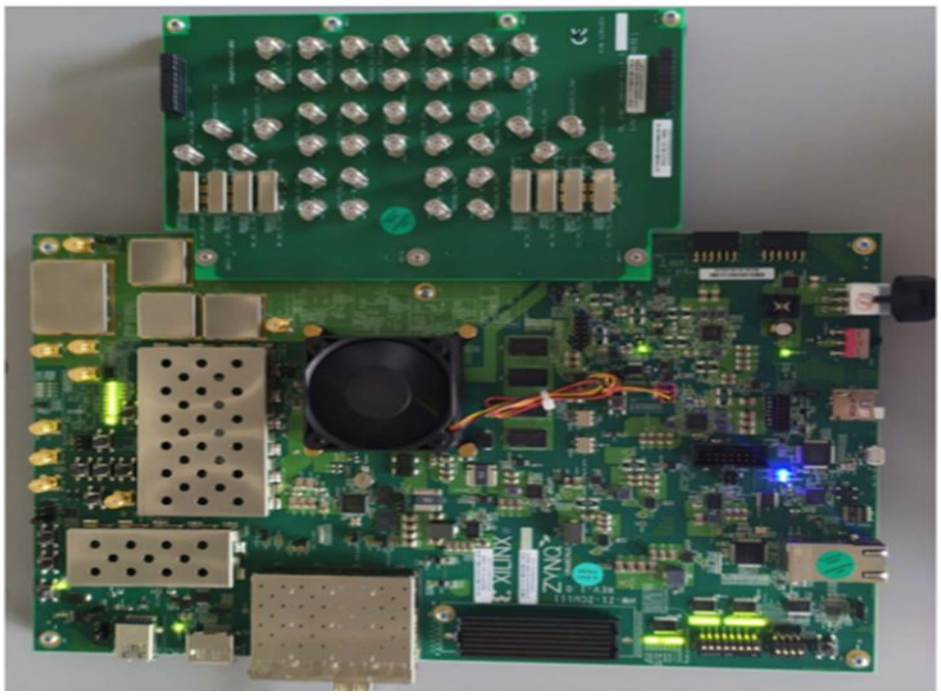


Figure1.3 :Hardware of ZCU111 Board.

RELATED CONCEPT AND LITERATURE REVIEW

1.4 Background reading:

1.4.1 Digitizing Receiver:

The style of present and future RF sensing has evolved greatly from the conventional single role big, static,non-dynamic radar systems. Digital signal processing has gained popularity in a variety of industries in recent years, including communication, radar, and medical imaging. One of the main components in thesesystems is the digitizing receiver, which is

responsible for extracting pulse parameter from incoming signals. Pulse parameters are useful because they can provide information about the characteristics of the signal, such as its frequency, amplitude, and phase. The digitizing receiver is particularly important in applications that require high-speed data processing, as it allows for the rapid extraction of pulse parameters. This is essential in many fields, such as communication and radar, where signals need to be processed quickly in order to extract useful information.

1.4.2 Main Blocks Of Digitizing Receiver:

The digitizing receiver consists of three main blocks: an analog front-end (AFE), a digital signal processor (DSP), and a control unit. The AFE is responsible for down converting and digitizing the incoming RF signal. The DSP performs digital signal processing on the digitized signal to extract various pulse parameters such as pulse width, pulse amplitude, and pulse rep etition interval (PRI). The control unit is responsible for controlling the operation of the AFE and DSP and provides a user interface for the receiver.

1.4.2.1 Analog Front-End (AFE):

The analog front-end (AFE) is the first block in the digitizing receiver. Its primary function is to receive the analog signal and convert it into a digital signal for further processing. The AFE typically includes an antenna, a preamplifier, a bandpass filter, and an analog-to-digital converter (ADC). The antenna collects the analog signal, which is then amplified by the preamplifier to increase its strength. The bandpass filter is used to remove unwanted signals that are outside the frequency range of interest. Finally, the ADC converts the analog signal into a digital signal, which can be further processed by the DSP.

1.4.2.2 Digital Signal Processor (DSP):

The digital signal processor (DSP) is the second block in the digitizing receiver. Its primary function is to perform digital signal processing on the digitized signal received from the AFE. The DSP typically includes a digital filter, a demodulator, a decoder, and a data processor. The digital filter removes unwanted noise and interference from the signal. The demodulator extracts the modulation information from the signal, and the decoder decodes the information into its original form. Finally, the data processor performs any necessary post-processing on the decoded information.

1.4.2.3 Control Unit (CU):

The control unit is the third block in the digitizing receiver. Its primary function is to control and coordinate the operation of the AFE and DSP blocks. The control unit typically includes a microprocessor, memory, and interface circuits. The microprocessor executes the control software that manages the operation of the digitizing receiver. The memory stores the control software and any other data required by the receiver. The interface circuits allow the receiver to communicate with other devices or systems, such asa computer or a network.

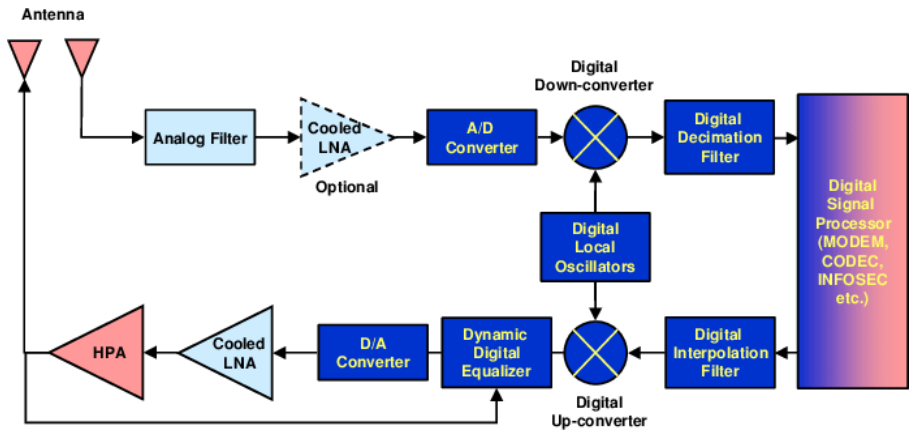


Figure1.4.2 2: Architecture of Digitizing Receiver

A commercially available device that has revolutionized RF sensing capabilities is the Xilinx Radio Frequency System on a Chip (RFSoC)development board. This device combines a powerful processing system, a Field Programmable Gate Array (FPGA), and high-speed ADCs and DACs. By integrating these key components with high-quality specifications, the RFSoC has become a hardware tool capable of supporting a wide range of applications. The development board provides a complete hardware platform for implementing the receiver, including the AFE and DSP blocks. The AFE is implemented using the onboard LNA, mixer, and low-pass filter, while the DSP is implemented using the onboard FPGA and processor.

1.5 Evolution of RF digital Systems

1.5.1 High Level Architecture of RF Systems:

1.5.1.1 First Generation RF System:

The Earlier a Digital RF systems, all of the modulation to RF, filtering and RF amplification was carried out using discrete analogue RF components. Fig.1.5.1.1 shows the generic High-level architecture of first-generation RF system. The analog section down covert signals from RF carrier using analog Local Oscillators. The baseband signal is then sampled and digitized using an ADC. The DSP operation are used to perform final processing stages to recover the transmitted information. The main limitation of this generation RF system was the limited speed of A/D converter which was operated at 100 KSs with 16-bit resolution.

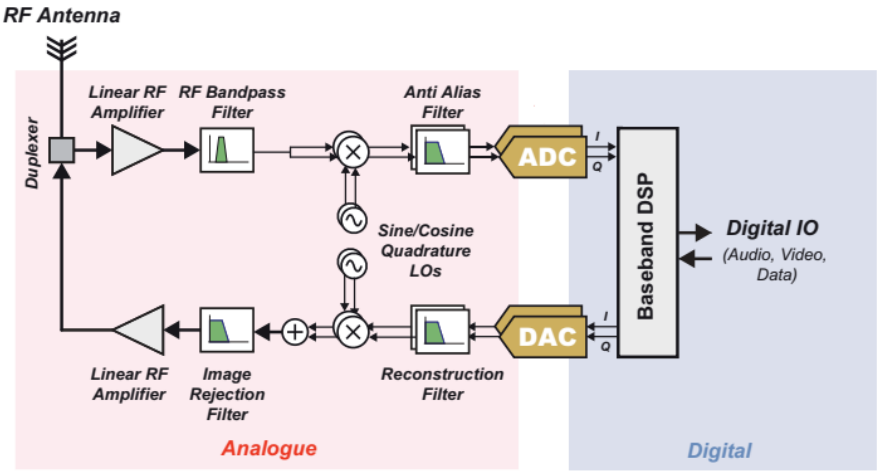


Figure1.5.1.1: High Level of Architecture of First-Generation RF System

1.5.1.2 Digital Sampling of RF System:

As the sampling speed of A/D interfaces increased to MPs range, the sampling and digitization were performed at and IF, rather than baseband as shown in Fig.1.5.1.2. In this architecture, IF signals are shifted to baseband signals using Direct Digital Downconverters. After shifting, DSP techniques was then performed on signal for final processing. This architecture provides greater flexibility for functional implantation in digital domain.

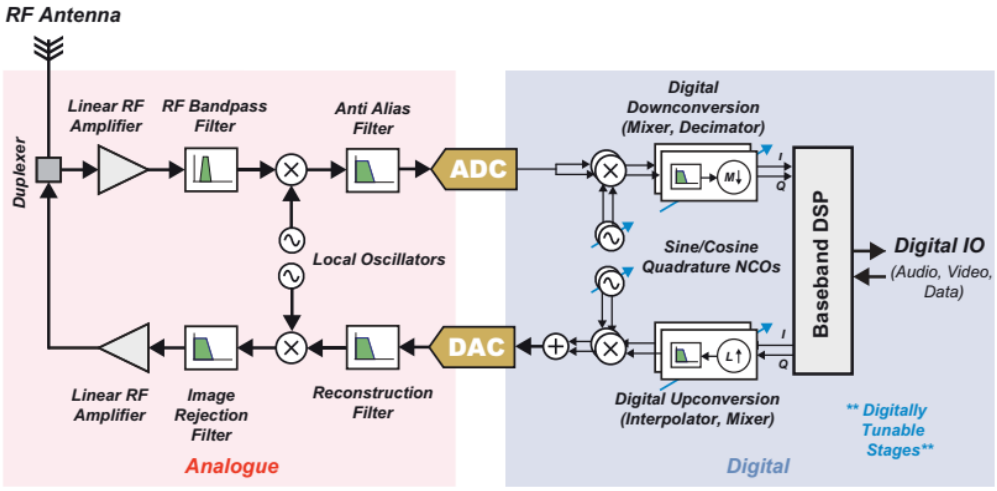


Figure1.5.1.2: High Level of Architecture Of Digital IF Sampling Of RF Systems

1.5.1.3 Tunable RF System:

With the increasing demand of the RF application, IC manufacturer developed single chip RF system that combined some of analog and digital stages together. These systems were enable with the software/digital tenability to analogue oscillator, filtering and amplifiers stages as shown in Fig. 1.5.1.3. These changes made A/D converter capable to operate at sample rate of 100s of MSps. By 2010, the higher CPU speed and larger memories made general purpose computers more capable of performing the final DSP operation using software such as MATLAB. FPGA plays key role as interconnected hub between computer interfaces and RF system front end chip.

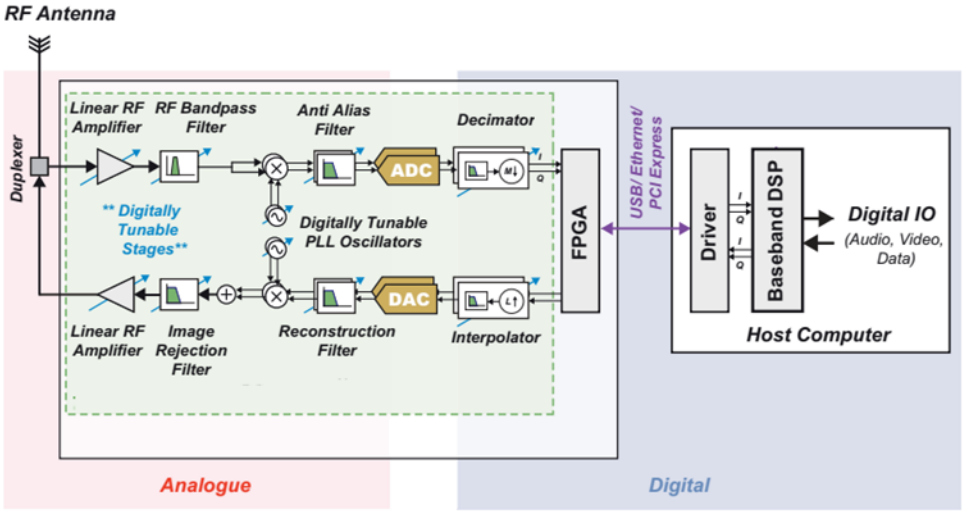


Figure1.5.1.3: High Level Of Architecture Of Tunable RF Systems

1.5.1.4 Direct RF System:

In recent years, a commercially available device that has revolutionized RF sensing capabilities is the Xilinx Radio Frequency System on a Chip (RFSoC) development board.These systems are capable to run at multi GSps, making them suitable for wide range of applications. In these systems (as shown in Fig. 1.5.1.4), RF frequencies are directly down converted to baseband signals in a single stage using DSP. These systems require very little analogue processing, mostly at the front end RF filters and RF amplifiers. This device combines a powerful processing system, a Field Programmable Gate Array (FPGA), and high-speed ADCs and DACs. By integrating these key components with high-quality specifications, the RFSoC has become a hardware tool capable of supporting a wide range of applications. The development board provides a complete hardware platform for implementing the receiver, including the AFE and DSP blocks. The AFE is implemented using the onboard LNA, mixer, and low-pass filter, while the DSP is implemented using the onboard FPGA and processor.

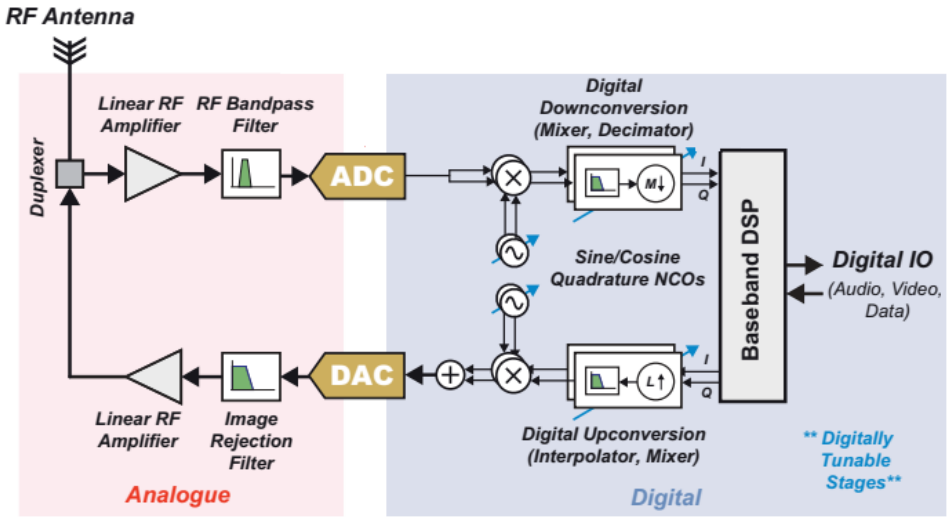


Figure 1.5.1.4 2: High Level Of Architecture Of Direct RF Systems

1.6 Related Work:

The implementation of radar has been a topic of interest for researchers. Early radar system were implemented using analog circuits. However, with the advancement in modern digital circuits, Direct Digital synthesis have been realized and widely researched. Early RF system depends on the partial use of analog waveform generator components which lack reconfigurability or wideband performance. The work of [1], realize linear frequency modulation synthesizer for a limited band of 500 MHz using the analog device to realize direct digital synthesizer. In [2], the authors developed a programmable Digital Waveform Generator for 200MHz Bandwidth. Similarly, the advancement in FPGA technology has enabled implementation of DDS. For example, the authors in [3] used ZYNQ FPGAs to implement LUT based Direct Digital Synthesizer for 2 MHz bandwidth. These systems, however shows a limited capability for a digital synthesizers for high bandwidth. Recently an effort has been made by authors of [4] to design and implement a high speed multichannel software defined radio with CPU for a bandwidth across WIFI signals.

As discussed earlier, the development in FPGA technology has made it possible to implement wide band digital signal processor on FPGA. Several studies have been conducted to implement radar system using FPGA. For example, the authors in [5] discussed the modern real-time wideband digital signal processing algorithms for test and evaluation of present-day radars. In [6], authors discussed the design and implementation of digital pulse compression for radars using FPGA IP core to simply the design. In [7] work, FPGA technology was used to implement a 320 MHz digital linear modulated signal generator. Similarly in [8] the author used virtual array method and time division technique to implement Multiple Input Multiple Output (MIMO) radar imaging system, where the signal processing part is implemented on FPGA and DSP. For usage with a shared RF aperture in simultaneous operation, the authors of [9] developed a system combining a coherent linear frequency modulated radar transmitter and receiver with a Digital Radio Frequency Memory (DRFM) jammer utilizing a reasonably priced Field Programmable Gate Array (FPGA).

The implementation of radar system using RFSOC is new phenomenon and the work on this topic is limited. However, some work provide information on design and implementation of radar systems using RFSOC FPGAs. In [10], the authors design a Software Define Radio

(SDR) on Xilinx RFSOCs. The authors of [11] used a pseudo random-number generator to construct a real-time digital signal generator for a noise radar. The authors of [12] compare numerous variants of digital passive radars, including one based on the RFSoC. The authors claim that RFSoC can give area, weight, and power advantages when many reception channels are needed. Similarly, the authors in [13] presented a wideband Multiple Input Multiple Output Frequency Modulated Continuous Wave (MIMO-FMCW) radar design based on a commercial RF system-on-chip (RFSoC). Furthermore, it describes the digital and RF hardware system that enables effective amplitude digital pre-distortion, resulting in improved quality of sending chirped and receiving de-chirped signals.

CHAPTER 2

INSTALLATION OF RELATED SOFTWARES:

2.1 MATLAB 2022a Installation Methodology

2.1.1 Procedure:

- **Step01:**

First copy the folder you have given with a name Matlab_2022a in any local disk of your PC.After this open the Matlab_2022a folder you will get two more folder .One is MATLAB_2022a windows_setup folder and another is MATLAB_2022a_installation folder.

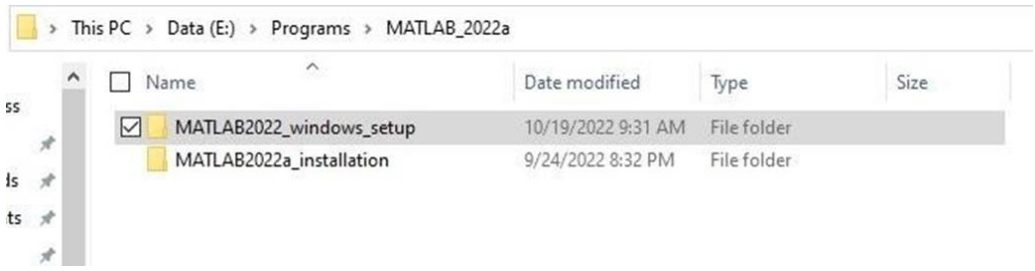


Figure2.1.1a : Open The Matlab Installation Folder

- **Step02:**

Open the folder MATLAB_2022a _windows_setup you will get various zip files. Now Open the folder MATLAB_2022a_installation you will get various zip files.Now extract all the files that are present in MATLAB_2022a _windows_setup and MATLAB_2022a_installation folder using WinRAR.

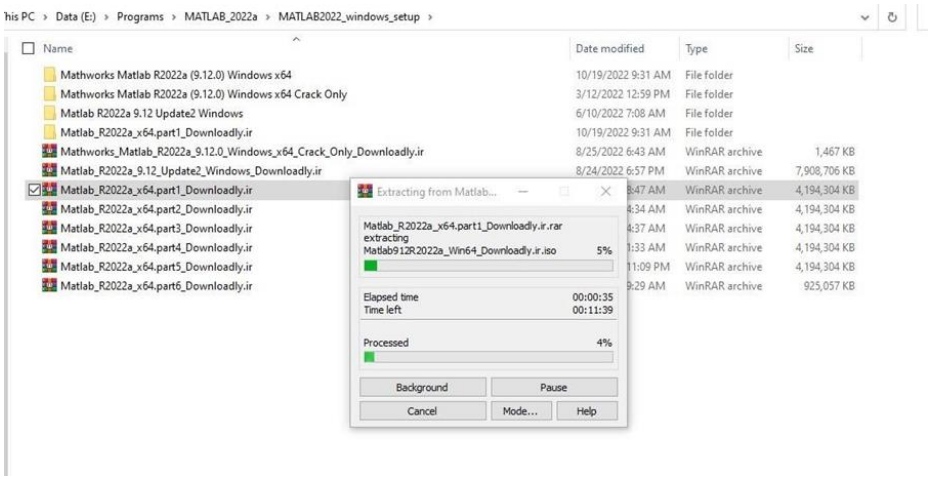


Figure2.1.1b : Extract the Matlab _Installation Files

- **Step03:**
  First open the setup and paste here the installation key from Read Me file. And then click next.

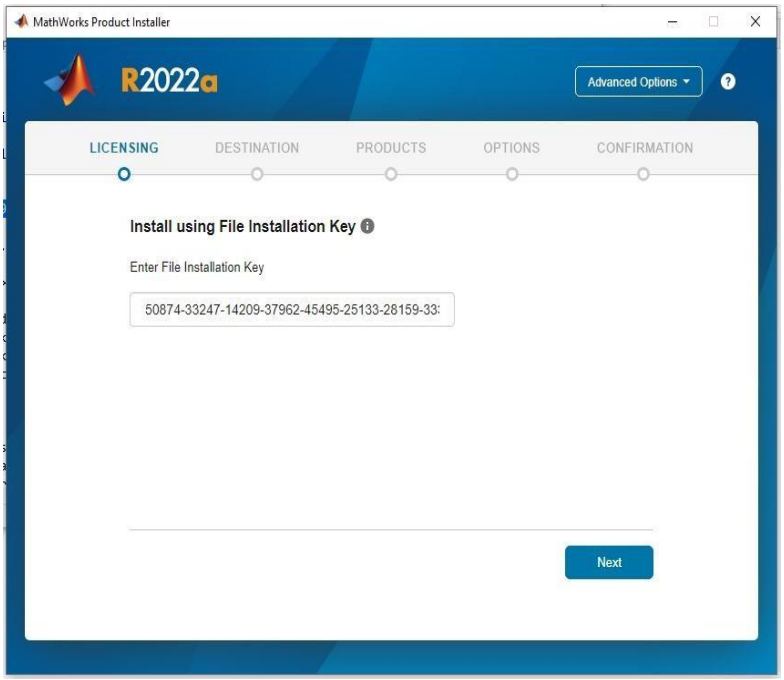


Figure 2.1.1c : Copy And Paste The Installation Key From Read.Me file

- **Step04:**

Now browse the location of the licence file from MATLAB_2022a _windows_setup folder and then click next.

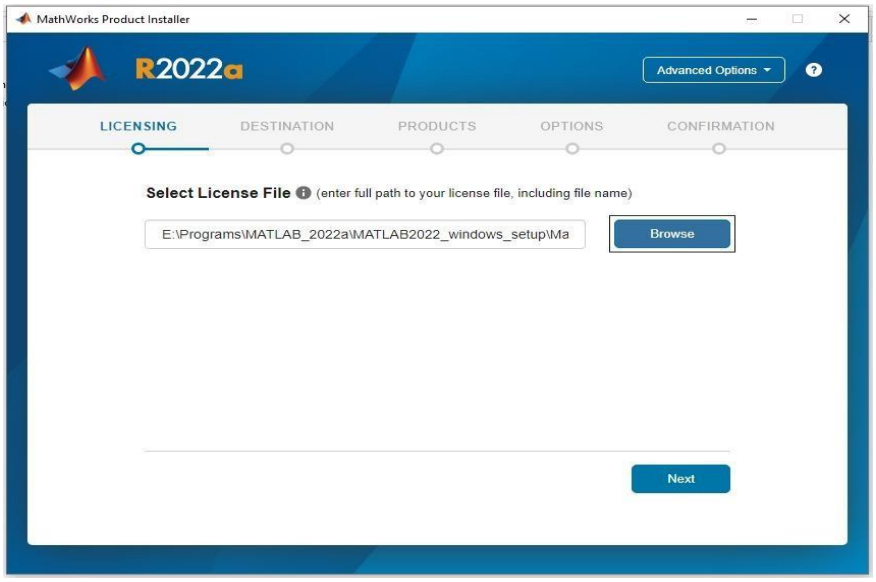


Figure2.1.1d : Paste the licence File Located at Matlab Folder

- **Step05:**
  Now select the folder where you want to install the Matlab and then click next. Select all the products and click next.

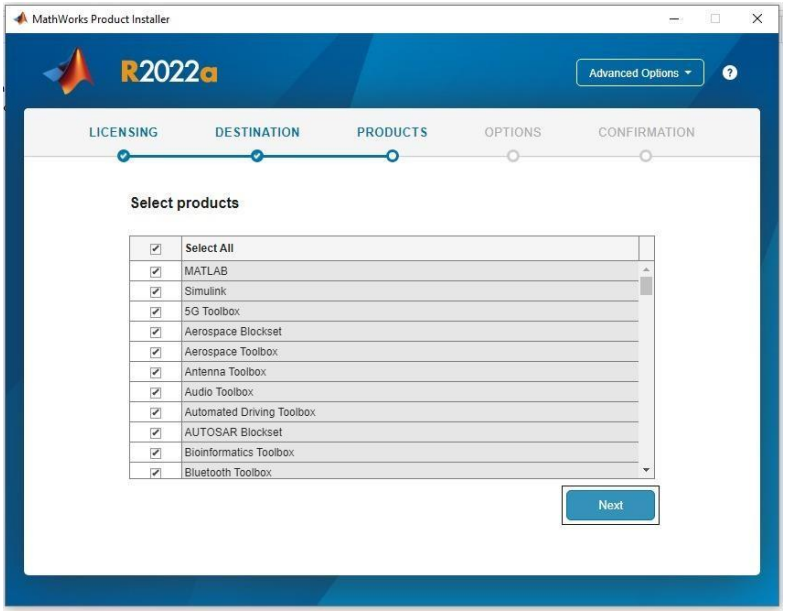


Figure2.1.1e : Install The Desire Products

- **Step06:**

Now select the option to create a shortcut on desktop. Click next. Click begin installation.it takes several minutes to complete.

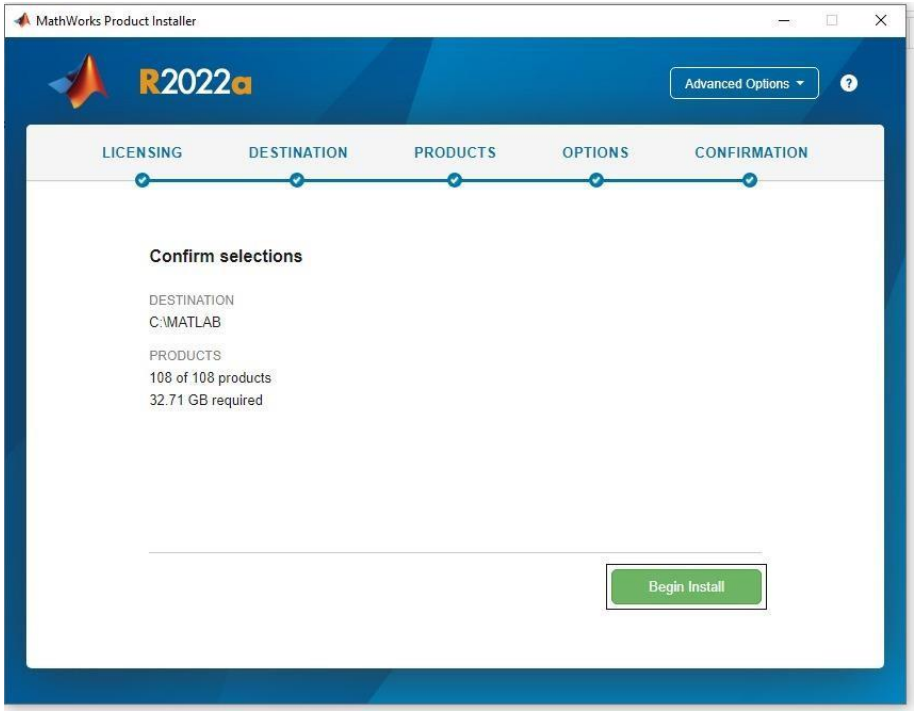


Figure 2.1.1f : Create A Desktop Shortcut And Begin Installations

- **Step07:**

After several minutes installation setup will completes. Then select the button. Now open the folder where you have the installation setup .Open the crack folder in it and copy following files to the directory where you have installed the Matlab. (libmwlmgrimpl.dll)

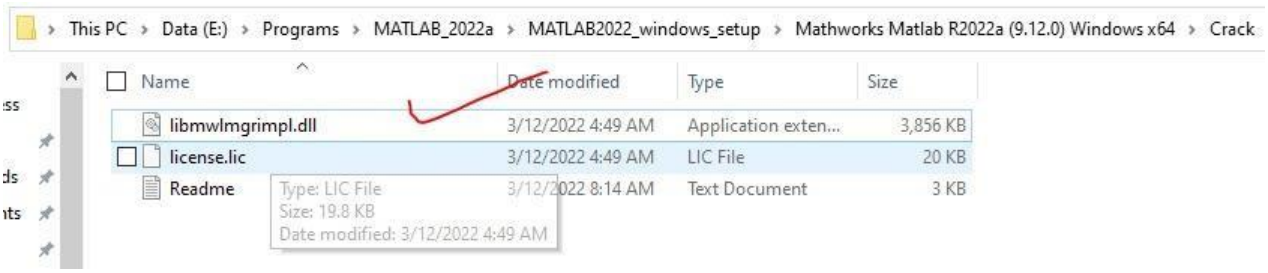


Figure 2.1.1g : Copy The Directories to Installed Matlab Folder

- **Step08:**

Now copy the above selected file and paste it into the setup destination folder directory.The following is mine directory you may use your own.

C:\MATLAB\bin\win64\matlab_startup_plugins\lmgrimpl .

Thus your MATLAB_2022a is installed successfully.

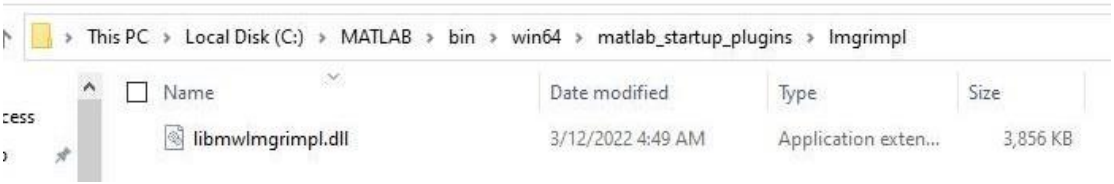


Figure 2.1.1h : Paste the Directory To Destination Folder

2.2 Matlab_2022a Updates Installation:

2.2.1 Procedure:

- **Step01:**
  Make sure your computer is not connected with any type of internet sources when you are installing updates to your matlab. Make sure your Support Package folder is on the same hard drive as your MATLAB installation. If not, move it. Open windows powershell.

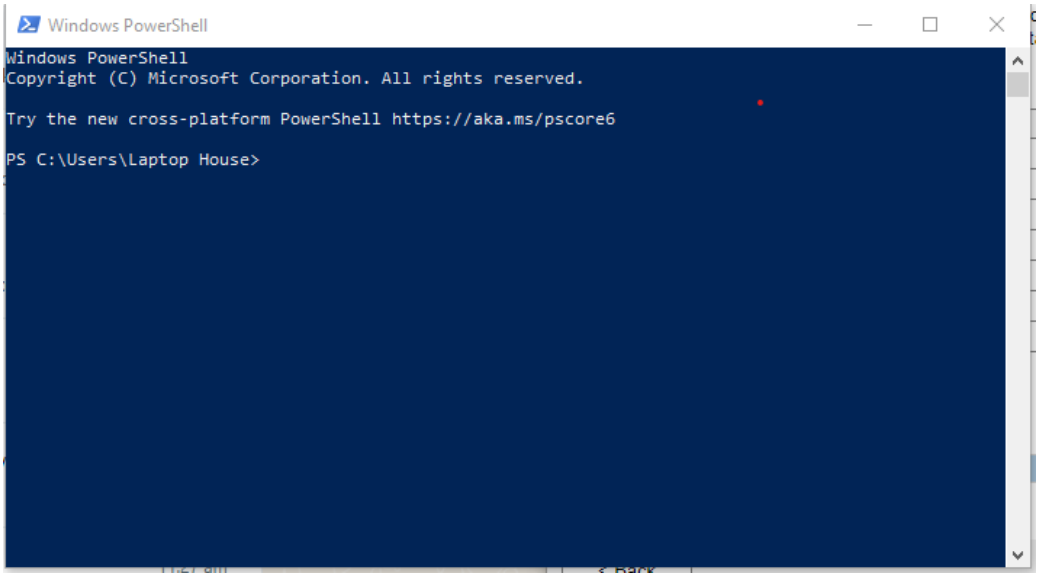


Figure 2.2.1a:Open Power shell From Windows

- **Step02:**

➢ **(In windows powershell) Make sure your path is on the same hard drive as your MATLAB installation drive. If not, change it by typing:**

➢ **D: (to change it to D drive, for instance)**

➢ **(In windows powershell) Change directory to your MATLAB folder, e.g. cd "C:\Program Files\MATLAB\R2017b\bin\win64"**

➢ **PSC:\MATLAB\bin\win64>.\install_supportsoftware.exe–archives C:\MATLAB_packages\ -matlab root C:\MATLAB**

➢ **NOTICE: the path should end in the directory in which there exists an "archives" folder.**

➢ **A new MATLAB window and an installation window should open.**

➢ **Select all the packages or just select the soc blockset package by searching the soc. And then select next.**

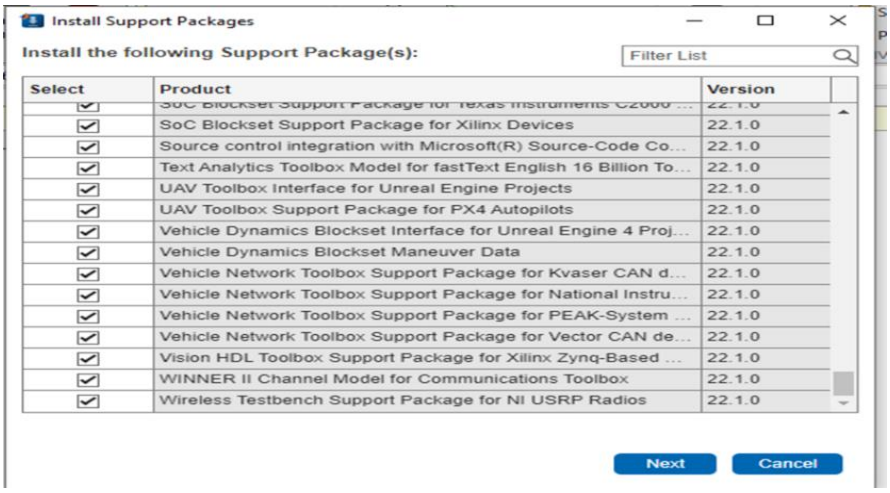


Figure 2.2.1b : Select The Require Support Packages

- **Step03:**

Now accept all the licenses. You will begin with a installation window. It will take a several minutes to complete its installation. After completion .Click next.

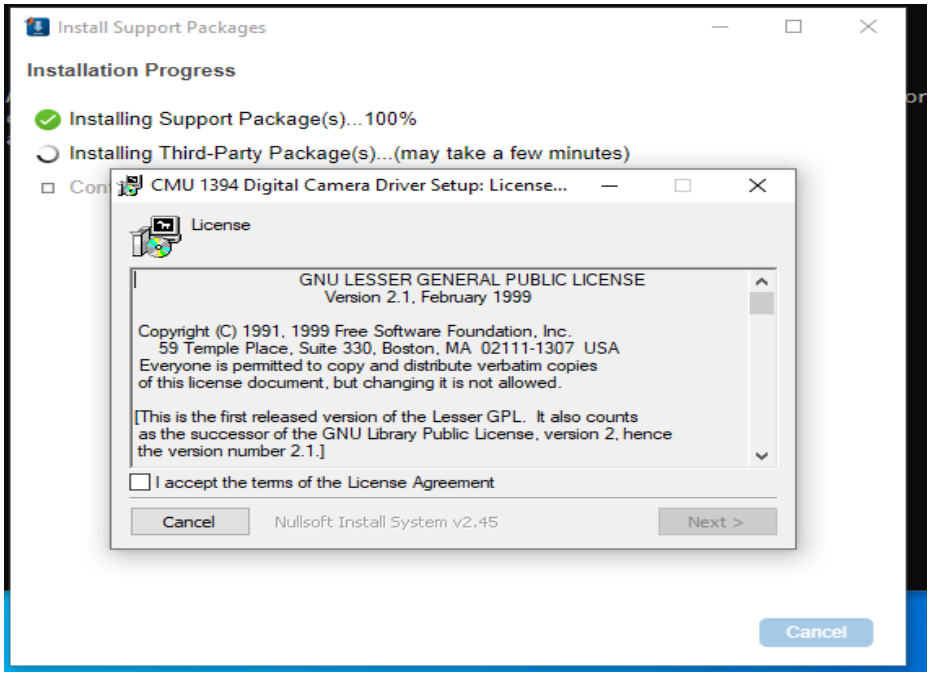


Figure 2.2.1c : Installation Process Started

- **Step04:**
  After successful installation of the packages open matlab and click on Ads-Ons in Matlab Window. Open and install the third party tools for soc blockset for Xilinx.

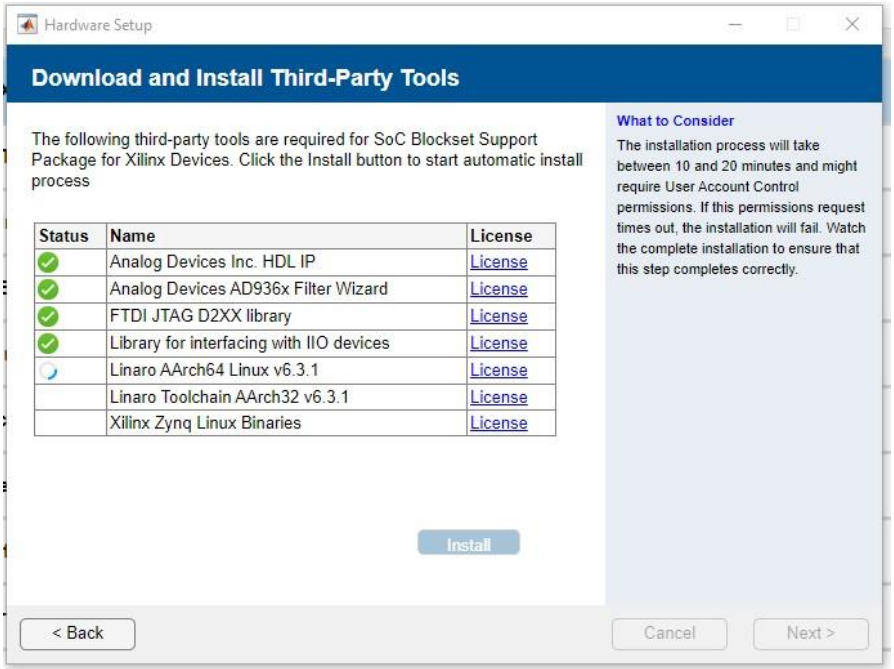


Figure 2.2.1d : Installation Of Third Party Tools

- **Step05:**

After successful installation of all the licences Restart the matlab And now your matlab is ready.After all if your matlab is not working and gives some errors then change your matlab installation folder to any other drive from C if it is in C drive

2.3 Vivado_2020.2 Installation Process

2.3.1 Procedure :

- **Step01:**

Open the folder of Vivado_2020.2 where you will find the files .Extract all the files. After extraction Now open the following folder.

Figure 2.3.1a: Open Vivado and Extract Files

- **Step02:**

Now double click on setup. The following window will and in it click on next.

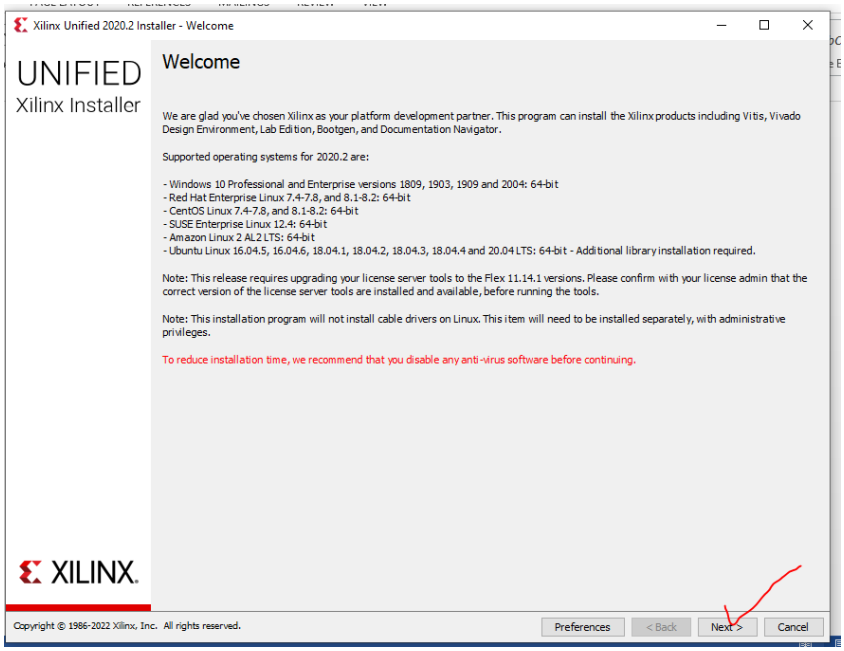


Figure 2.3.1b : Follows Setup Instruction

- **Step03:**

Select the Vitis and click on next. Just click on next. Next you will Click on all the agree options and click on next. Choose your destination for the vivado and click next and then start the installation

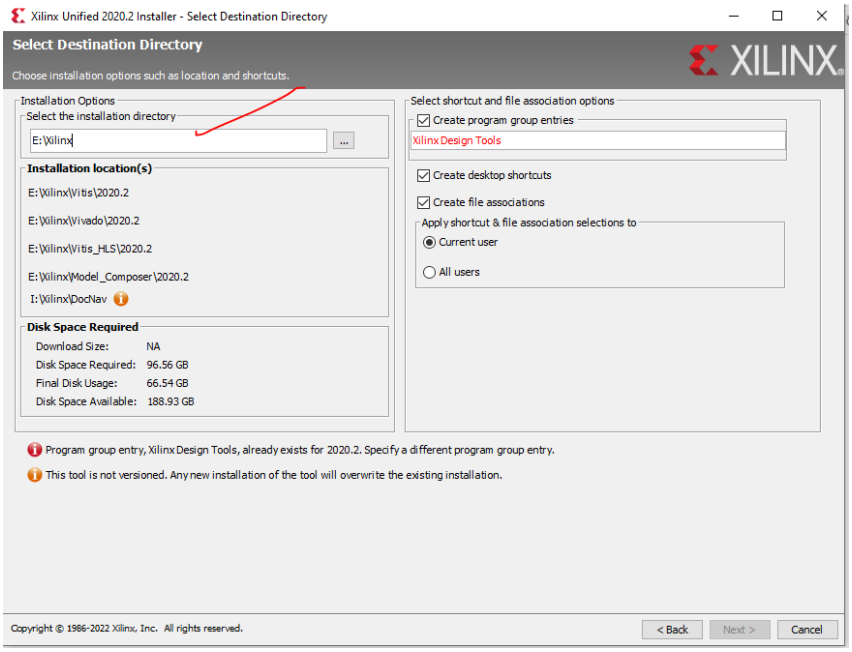


Figure 2.3.1c : Select Destination Directory

- **Step04:**

  After several minutes your installation will completes. Then click on close. Now open the vivado and click on help . Now click on manage licenses you will get the following window**.**

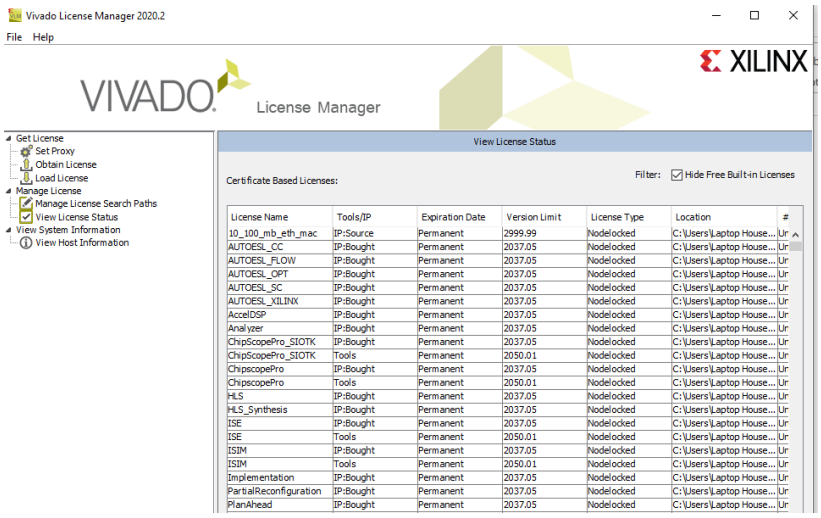


Figure 2.3.1d : Open License Window In Vivado

- **Step05:**

After following these steps now add the following two licences to the vivado by clicking on load license. Now your Vivado_2020.2 installation is successfully completed.

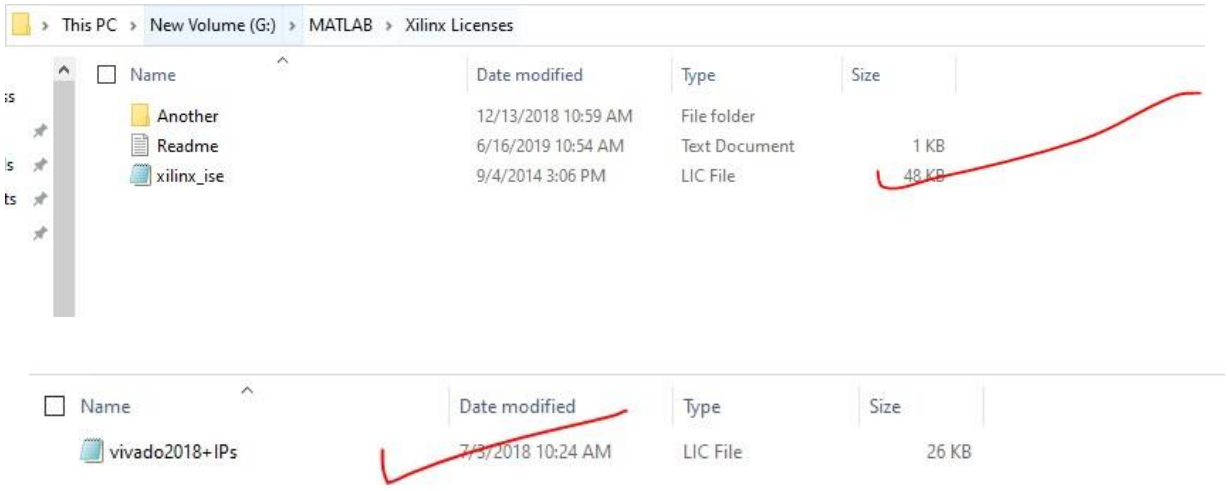


Figure2.3.1d : Paste License Files In Vivado Folder

CHAPTER 3

SOC WORKFLOW OVERVIEW

3.1 Transmit and Receive Tone Using Xilinx RFSoC Device

3.1.1 Introduction:

This example demonstrates how to use SoC Blockset to construct a data channel for a Xilinx RFSoC device. Using the RF Data Converter (RFDC) block, you will construct and simulate a system that transmits a sinusoidal tone across a number of RF channels from an FPGA. The system will then use the RFDC block to receive the data back into the FPGA and an embedded CPU to show the received tone for a single channel at a time.

3.1.2 Design Task and System Specification:

Consider a wireless application that calls for the Xilinx RFSoC to access multiple RF channels in duplex mode at a gigasample-per-second (GSPS) data rate. On the ZCU111 evaluation kit, the design job in this example is to configure the RFDC block, generate a sinusoid tone from the FPGA, and receive the data back into the FPGA.

Table 3.1.2: System Specifications for ZCU111 Evaluation Kit

System Blocks	Sampling Rate	No Of Channels	Digital Data Interface
ADC	2048MSPS	8	Real
DAC	2048MSPS	8	Real

3.1.3 Design and implementation of the above system presents the following challenges:

1) It is challenging to configure the RFDC block's many parameter values to satisfy the system need.
2) Because of the high data rate, it is more complicated to create an algorithm in an FPGA to handle several samples at once.
3) Given the asynchronous nature of FPGA and processor, it is challenging to design the data connection between them to satisfy a certain system requirement.

3.1.4 Design Using SoC Blockset:

You can simulate the design before it is put into practice by modeling the system using the RF Data Converter block and external memory blocks offered by SoC Blockset. Prior to implementing the design on the hardware, you can find and address Simulink's bugs by simulating the design.

Three elite models make up this design illustration.

1) soc_rfsoc_datacapture — Capture real data on the ZCU111 evaluation kit with 8 channels.
2) soc_rfsoc_IQ_datacapture_top — ZCU216 assessment kit can capture sophisticated IQ data with 16 channels.
3) soc_IQ_datacapture_top — Gather sophisticated intelligence data using the ZCU208 assessment kit.

3.1.5 SOC Top Model:

Create the Xilinx Zynq UltraScale+ RFSoC ZCU111 Evaluation Kit as the Hardware Board and select the SoC model soc_rfsoc_datacapture as the top model. The processor model soc_rfsoc_datacapture_proc and the FPGA model soc_rfsoc_datacapture_fpga are included in this model as model references. The top variant also has an AXI4-Stream to Software block that allows the processor and FPGA to share external memory.

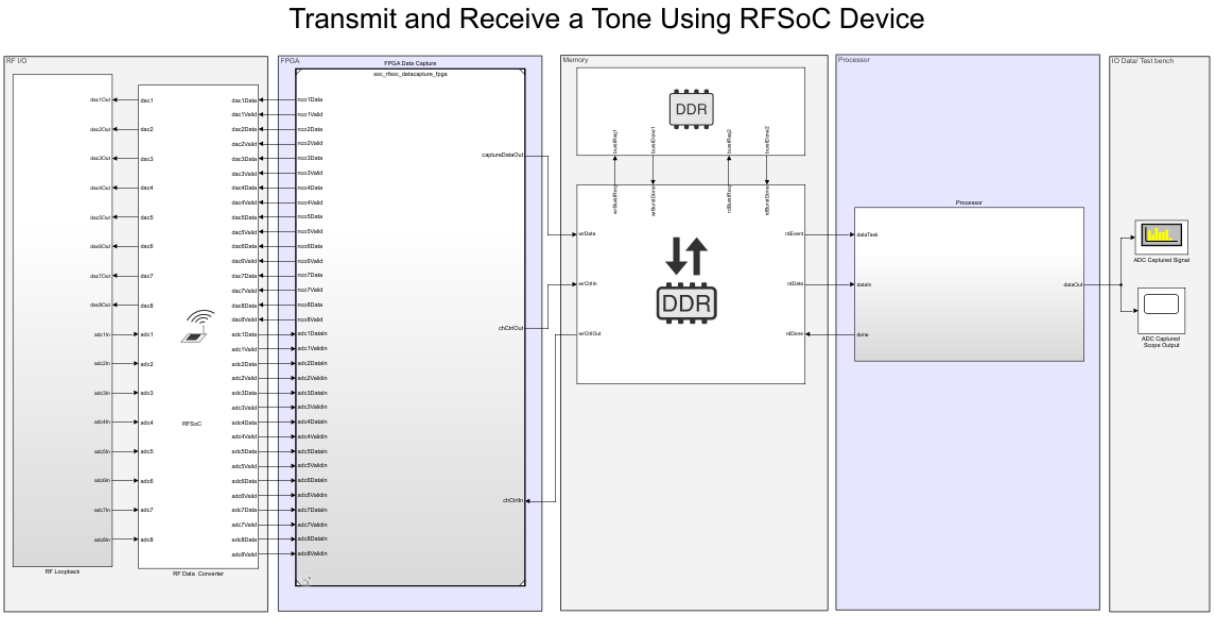


Figure 3.1.5 :Transmit and Receive Single Tone Using RFSOC Device

3.1.6 RF Data Converter Configuration:

The RF Data Converter block offers the wireless algorithm represented by the hardware logic an RF data channel interface. This block can be used to connect the data coming from and going to the hardware logic as well as to model and set up the ADCs and DACs. The block gives Simulink users access to the Xilinx RF Data Converter IP for modeling wireless systems that will be implemented on Xilinx RFSoC hardware.

You must set the Interpolation mode, Decimation mode, and Samples per clock cycle settings such that the effective clock cycle (sample rate) for the wireless algorithm FPGA is within the desired range in order to satisfy the system requirement of 2048 MSPS as the data rate for DACs and ADCs.

Set the Decimation mode and Samples per clock cycle parameter values to 4 in the ADC tab and the Interpolation mode and Samples per clock cycle parameter values to 4 in the DAC tab for this example. This suggests that the Stream clock frequency is 128 MHz, or 2048/(4*4).

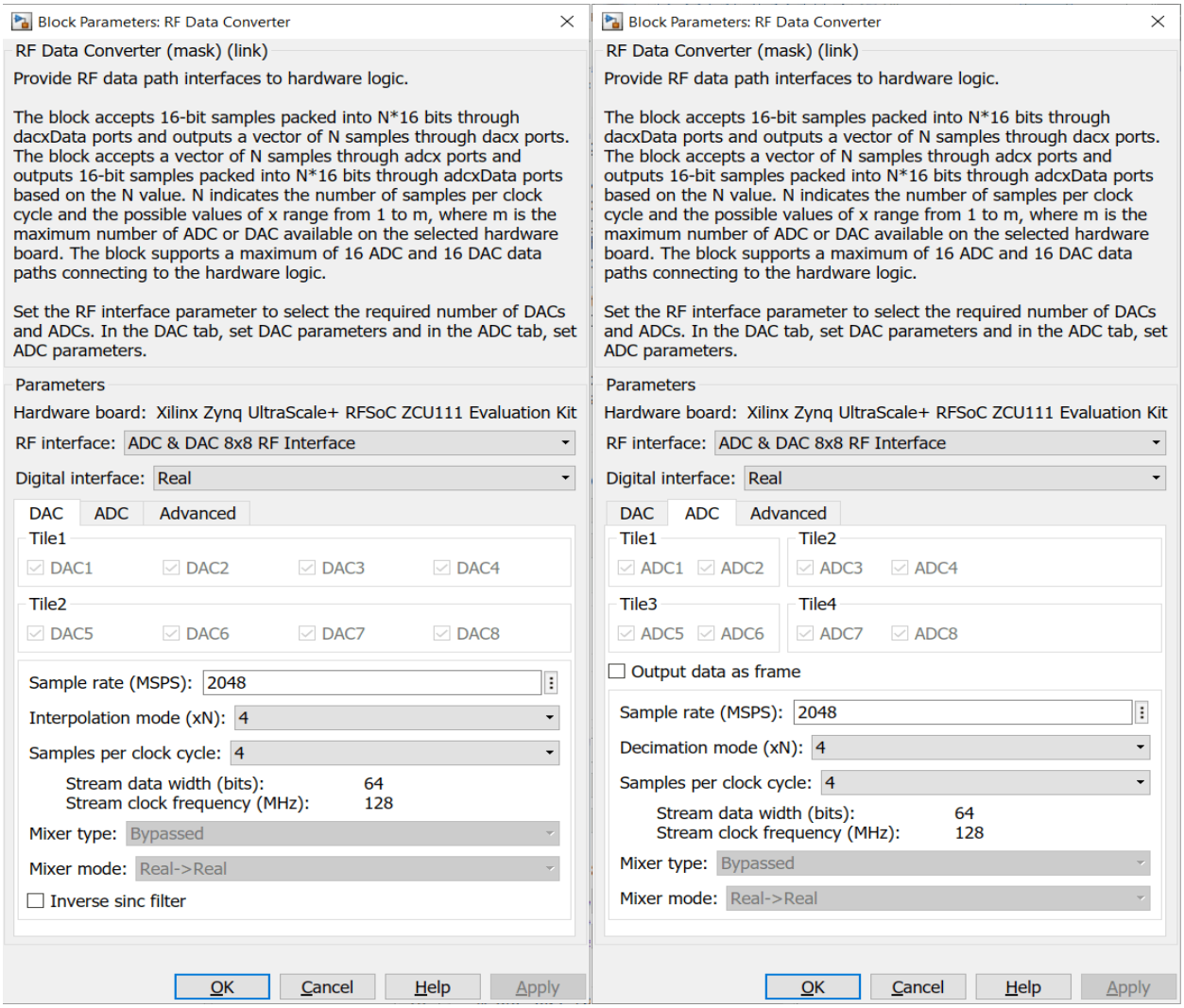


Figure 3.1.6 :Block Parameters of RF Data Converter

On the left side of the RF Data Converter block are input ports for ADC (adc1, adc2, and so on up to adc8) and output ports for DAC (dac1, dac2, and so on up to dac8). The right-side port signals that are connected to the FPGA are vectorized and used in the left-side port signals. The right-side DAC and ADC signals in this example are each 64 bits wide (uint64) and range from dac1Data to dac8Data and adc1Data to adc8Data, respectively. The left-side signals are a 4x1 int16 vector

version of the corresponding right-side signal. These ports can be used to represent the data flow for a wireless channel after the ADC and DAC, if necessary.

3.1.7 Hardware Logic Design:

The transmit route of the hardware logic must send four samples every cycle of a 128 MHz clock in order to correspond to the DAC digital interface of the RF Data Converter block. One of the eight DAC data ports for the RF Data Converter block is depicted in this image. The sinusoid tone production sampling rate is 512 MSPS (128 x 4)

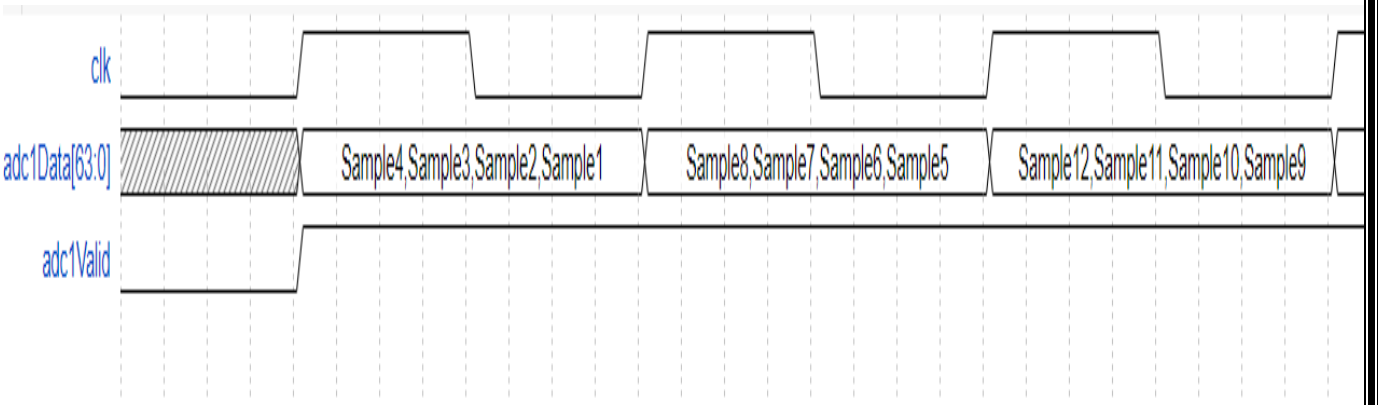


Figure 3.1.7 : Data Interfacing and Sampling Rate

3.1.8 FPGA Model Blocks:

The DAC Tone Generation subsystem is connected to the DAC section of the RFDC block, while the ADC Capture subsystem is attached to the ADC portion, in the FPGA model soc_rfsoc_datacapture_fpga. Using four HDL Optimized NCO blocks and a variable offset for each block, four consecutive samples of the sinusoid waveform are produced in parallel. Each of the four samples is compressed into 64 bits of AXI-Stream data. Furthermore, the same waveform is broadcast through each of the eight channels. Each channel is scaled using a unique scaling factor that the CPU sets in a register.

One of the eight channels in the ADC Capture subsystem is chosen using a register that the processor sets. The information from the chosen channel is down sampled by 128 .

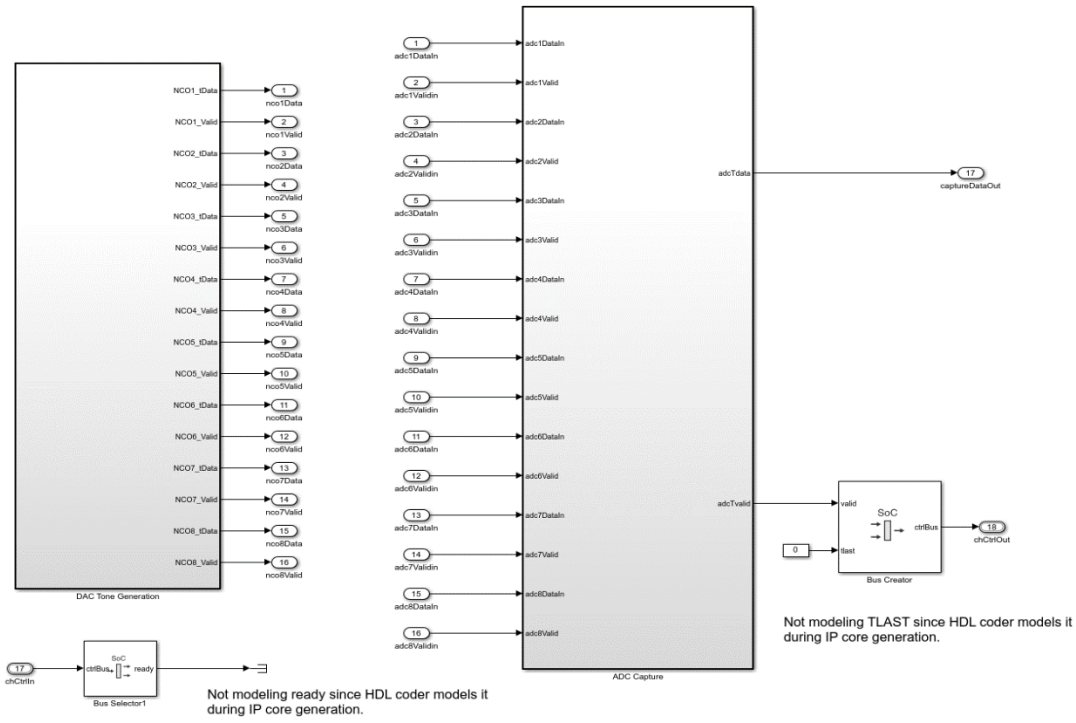


Figure 3.1.8 : FPGA Model Blocks

3.1.9 Processor Logic Design:

The logic of the processor includes an event-based task that is activated by the arrival of data from the FPGA via DDR memory. The Task Manager block designates the processor algorithm task as dataTask and specifies that it is event-driven. By using a buffer ready event, or rdEvent in memory, the Task Manager block can asynchronously schedule data. This event signifies the arrival of a frame of data from the FPGA. The actual method is modeled in the processor model soc_rfsoc_datacapture_proc within the Processor method Wrapper subsystem and is linked to the Task Manager block at the top level. It is necessary to unpack the data and then restore its signedness before performing any operations on the data that was sent to you as a frame of four samples packed as uint64.

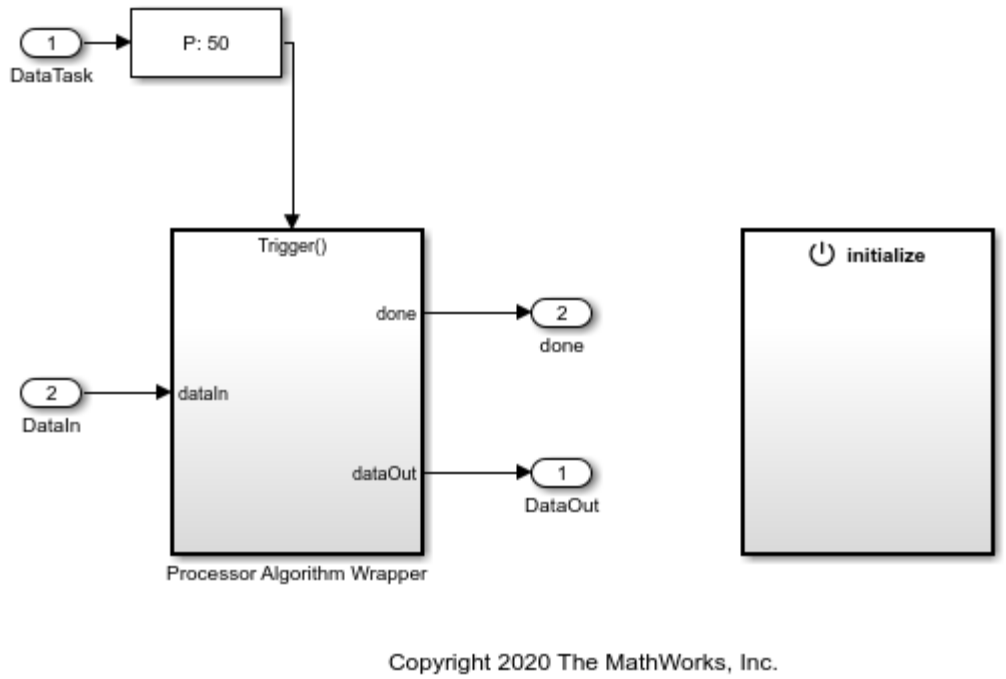


Figure 3.1.9 :Processor Event-Based Task Driven Block

3.2 Simultion Results:

3.2.1 ADC Capture Signal Waveform:

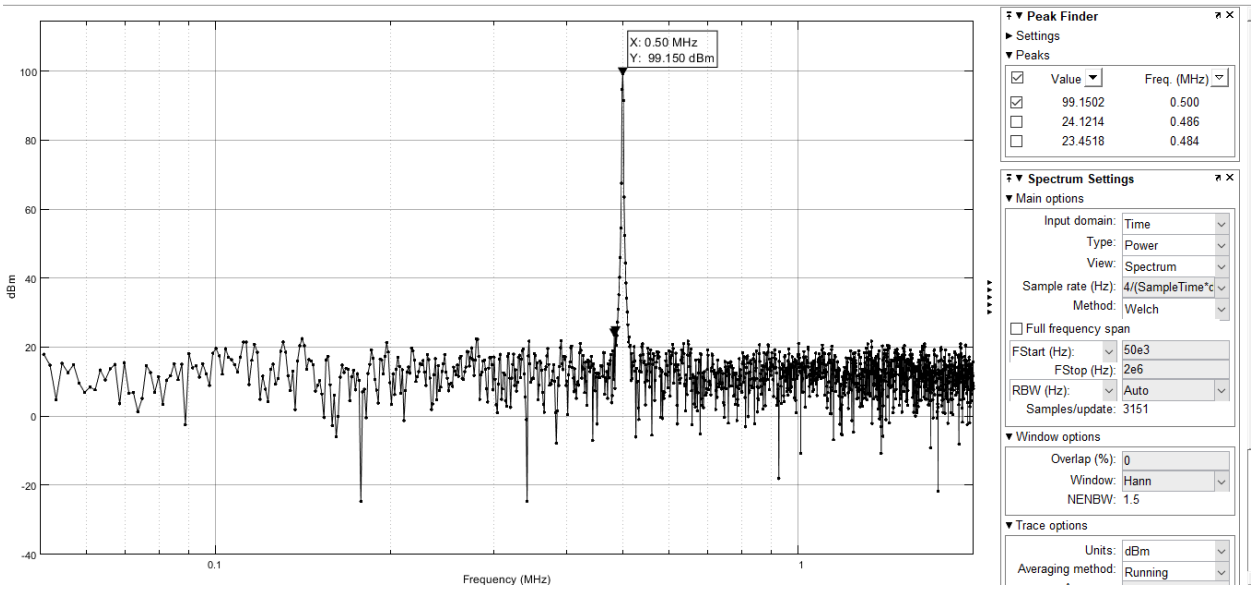


Figure 3.2.1 : ADC Capture Block Output

3.3 Vivado Architecture:

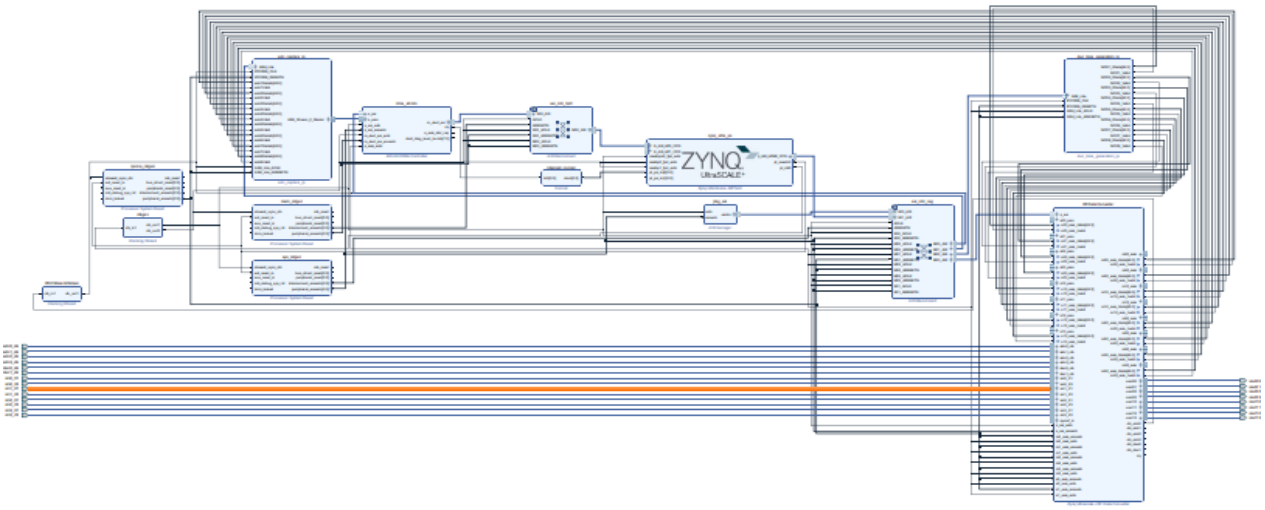


Figure 3.3 : Vivado IP's generation for Single tone signal Generation For RFSOC Device

3.4 Transmit and Receive Tone Using Fixed Reference Design Workflow on RFSoC Device

3.4.1 Introduction:

This example demonstrates how to use the RFSoC Support for a fixed reference design workflow to design and build a hardware algorithm on FPGA fabric that sends and receives a tone signal.In this illustration, the algorithm's HDL code is generated as an IP core and integrated into a reference design to create a system. After that, you install the system on hardware and use Simulink to display the signal that was received.These hardware systems are supported by this example.

"RFSoC ZCU111 test kit and XM500 balun card from Xilinx."

There are four models in this example. All of the models use the NCO HDL Optimized block to generate a sinusoid signal from the FPGA, which is then sent over the RFSoC device's DAC channels. The signal is then returned from the FPGA's analog-to-digital converter (ADC) channels. One of the received ADC channels is chosen by the receive-side FPGA logic, together with the logic for data storage. The host I/O models are then used in Simulink to show the data that was read from memory. These four models are somewhat dissimilar to one another.

1) actual data capture with eight channels in the internal BRAM FIFO with the soc_datacapture_8x8real_zcu111_top command.
2) Complex in-phase/quadrature (I/Q) data can be captured with four channels using the internal BRAM FIFO by using the soc_datacapture_4x4IQ_zcu111_top function.
3) sc_datacapture_8x8IQMTS_zcu111_top — Eight channels in the internal BRAM FIFO and various channels with multi-tile sync enabled allow you to capture sophisticated I/Q data.
4) the file soc_datacapture_4x4realDDR4_zcu111_top — Utilize the external DDR4 memory's four channels to capture actual data.

3.4.2 Designing Hardware Algorithm

3.4.2.1 SoC Model Creator:

Utilizing the SoC Model Creator tool, perform the following steps to generate the model.

1) Choose the Xilinx Zynq UltraScale+ RFSoC ZCU111 Evaluation Kit as the reference design board.
2) Real ADC/DAC Interface is the reference design name to use.
3) Set soc_datacapture_8x8real_zcu111 as the top model name.
4) Choose the FPGA only model type.
5) In the Reference Design Parameters pane, enter the necessary reference design parameters.
6) In the Internal Interfaces window, choose the necessary interfaces.
7) In the External IO Interfaces window, choose the necessary interfaces.
8) In the AXI Registers pane, add the necessary registers.
9) Press Create.

The soc_top model and soc_datacapture_8x8real_zcu111_fpga FPGA model are generated by the tool. Ports for each subsystem are already set up in the FPGA model to correspond to the reference design you've selected. The Transmit and Receive Tone subsystem is where you can enter your algorithm, and the top model is where you can add stimuli and scopes to simulate it. The hardware generation model that adds the transmit and receive method is soc_datacapture_8x8real_zcu111_fpga.

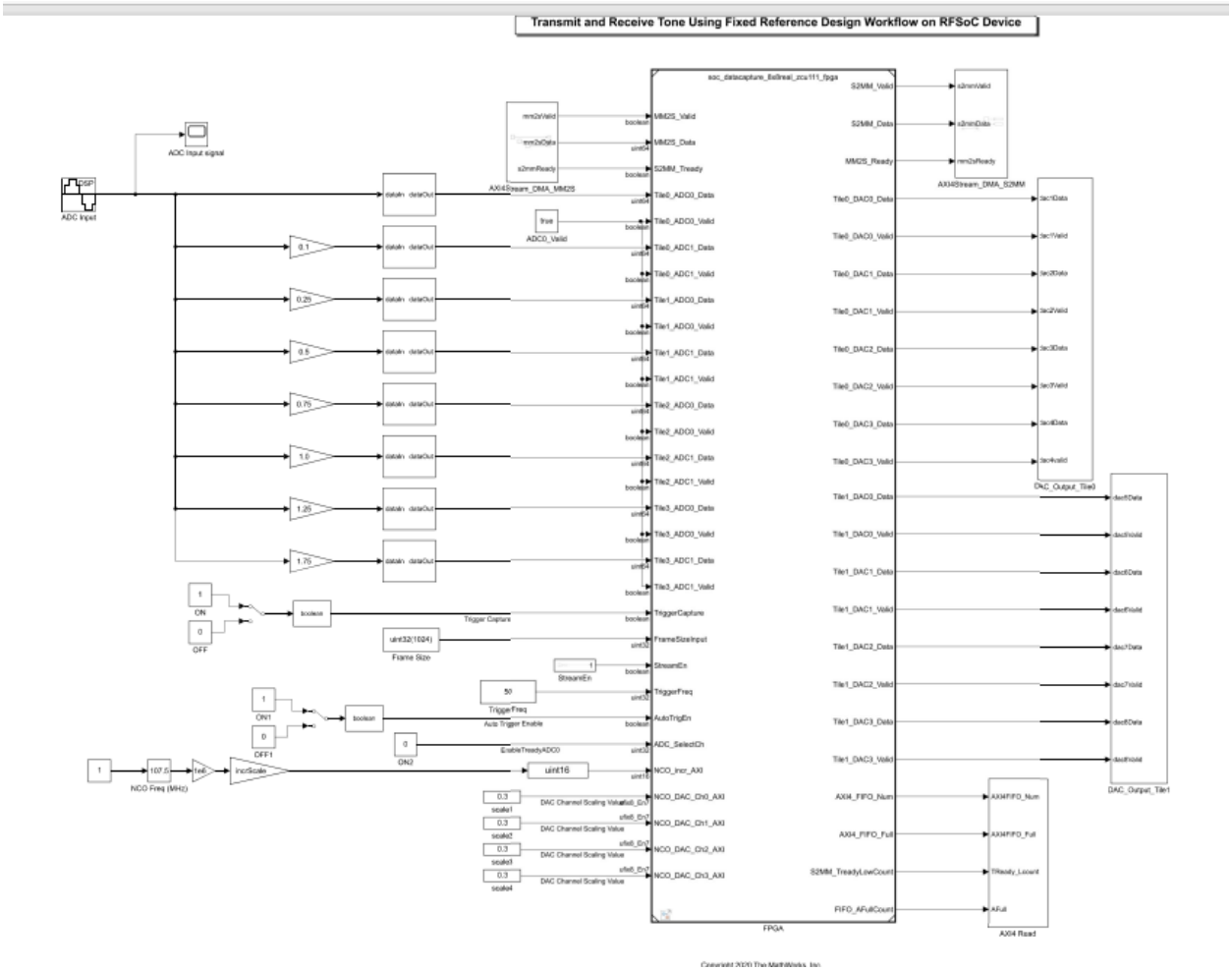


Figure 3.4.2.1 : Top Model Of Reference Design

3.5 Simulation Results:

3.5.1 ADC Capture Output:

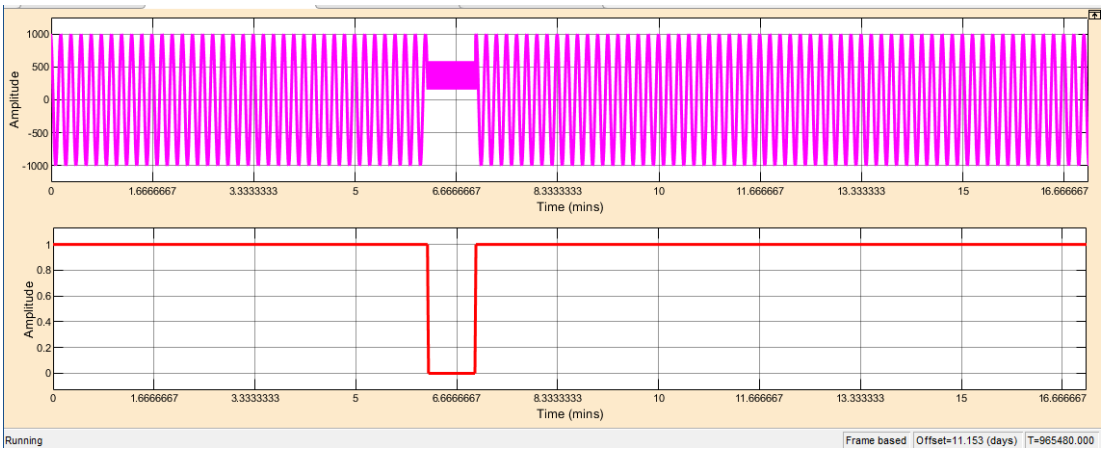


Figure 3.5.1: Reference Signal Generation

3.5.2 DAC Output Spectrum:

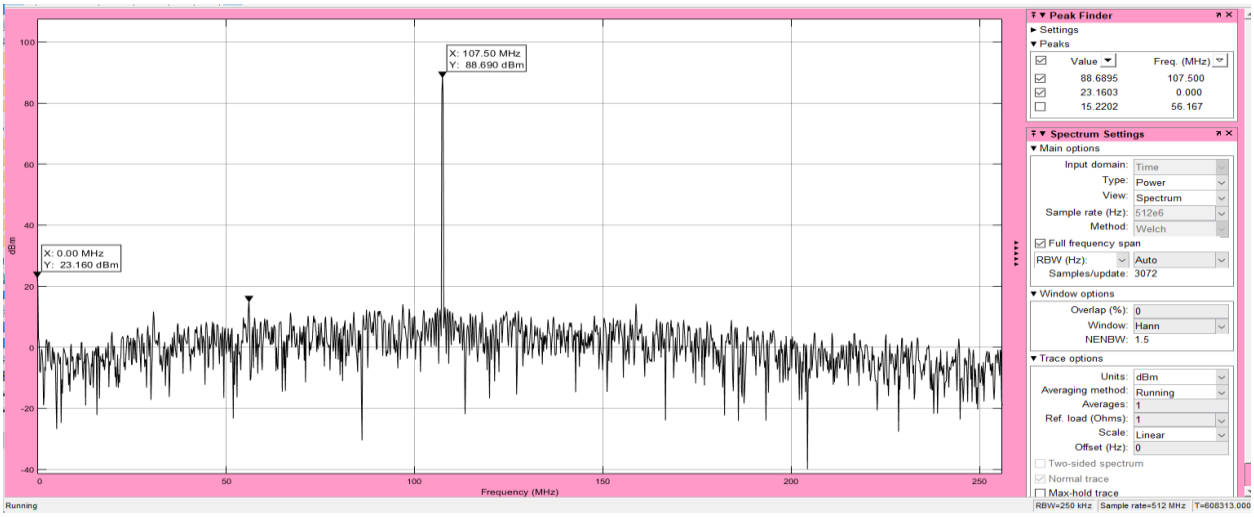


Figure 3.5.2 : Frequency Spectrum Of Reference Signal

3.6 Implement and Run on Hardware Specifications:

By selecting Block Parameters (Subsystem) and Treat as atomic unit from the context menu when you right-click the top-level DUT design subsystem Transmit and Receive Tone, you can make the DUT subsystem of the FPGA model an atomic subsystem. Use this command to configure the Vivado® tool version (which presumes Xilinx Vivado is installed at C:XilinxVivado2020.2bin).

"ToolName," "Xilinx Vivado," "ToolPath," "C:Xilinx Vivado 2020.2binvivado.bat,"

Use the SoC Builder tool to implement the model on a compatible SoC board. Make sure that the Xilinx Zynq UltraScale+ RFSoC ZCU111 Evaluation Kit is selected as the Hardware Board on the Simulink toolstrip's System on Chip tab before using this tool.

3.6.1 Procedural Steps:

1) Simply click Configure, Build & Deploy to launch SoC Builder. Take these actions after the SoC Builder tool has opened.
2) Choose Build using fixed reference design from the Setup screen. Choose Next.
3) Choose Build, load, and run from the Select Build Action screen. Choose Next.
4) Enter the project folder on the Select Project Folder page. Choose Next.
5) View the memory map on the Review Memory Map screen by selecting View/Edit. Choose Next.
6) By clicking Validate on the Validate Model screen, you can determine whether the model can be used for implementation. Choose Next
7) Build the model by selecting Build on the Build Model screen. As soon as FPGA synthesis starts, an external shell opens. Choose Next.
8) Click Test Connection on the Connect Hardware page to check the host computer's connectivity with the SoC board. Click Next to go to the Load Bitstream screen.

The Build stage creates the corresponding embedded system with bitstream, a host interface library, and a host interface model in addition to integrating the newly generated IP core into the RFSoC IP core reference design.

Table3.6.1: Hardware Specification For ZCU111 Board

Channel	Source	destination	Connection Cable Type
Channel 5	DAC228_TO_CHO (J26(P))	ADC226_T2_Ch0 (133(P))	SMA cable with DC block
	DAC228_TO_CH0 (J27(N))	ADC226_T2_Ch0 (J32(N))	SMA cable with DC block
Channel 6	DAC228_TO_Ch1 (J20(P))	ADC226_T2_Ch1 (J34(P))	SMA cable with DC block

	DAC228_TO_Ch1 (J21(N))	ADC226_T2_Ch1 (J35(N))	SMA cable with DC block
Channel 7	DAC228_TO_Ch2 (J22(P))	ADC227_T3_Ch0 (J37(P))	SMA cable with DC block
	DAC228_TO_Ch2 (J23(N))	ADC227 T3 Ch0 (J36(N))	SMA cable with DC block
Channel 8	DAC228_TO_Ch3 (124(P))	ADC227 T3 Ch1 (J39(P))	SMA cable with DC block
	DAC228_TO_Ch3 (J25(N))	ADC227 T3 Ch1 (J40(N))	SMA cable with DC block
Channel 3	DAC229_T1_Ch0 (J7)	ADC225_T1_Ch0 (J2)	SMA cable
Channel 4	DAC229_T1_Ch1 (J8)	ADC225 T1 Ch1 (J1)	SMA cable
Channel 1	DAC229_T1_Ch2 (J5)	ADC224_TO_Ch0 (J4)	SMA cable
Channel 2	DAC229_T1_Ch3 (J6)	ADC224_TO_Ch1 (J3)	SMA cable

3.6 Vivado Architectue Model :

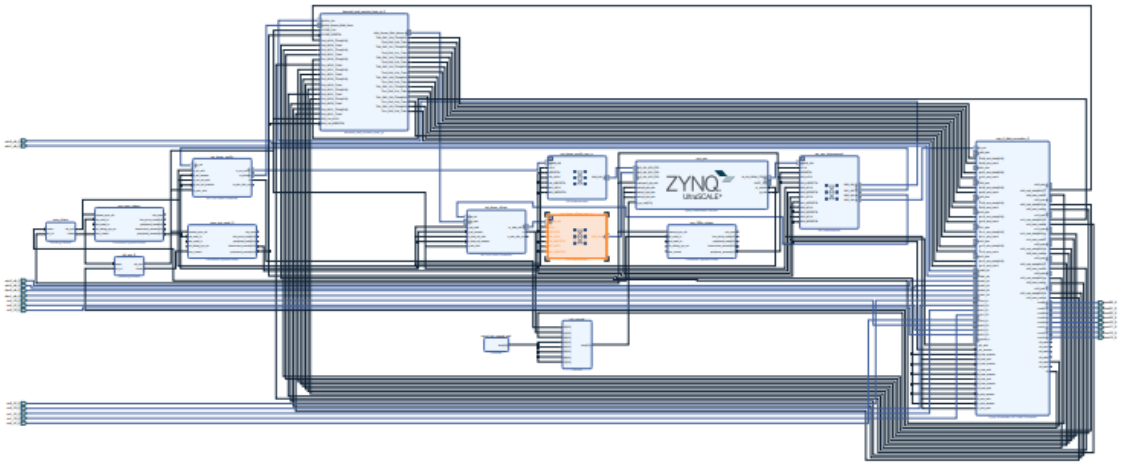


Figure 3.7 : Vivado IP's Generation For Reference Model

CHAPTER 4

SIGNAL GENERATON AND PULSE PARAMETER EXTRACION

4.1 Selection of design parameters for RFSoC

The system frequency band for this project is 0.5 GHz to 2GHz.The bandwidth of the signal is1.5GHz thus having the center frequency Fc=1.25Ghz. The ADC sampling frequency of 4 Gsps is enough to digitize the input signal as it falls in the first Nyquist zone (i.e., 0 to 2 fs/2 pulse width). Therefore, the direct RF architecture given below figure can be used. This architecture eliminates the use of intermediate frequency amplifier and filter often termed as RF to IF architecture.

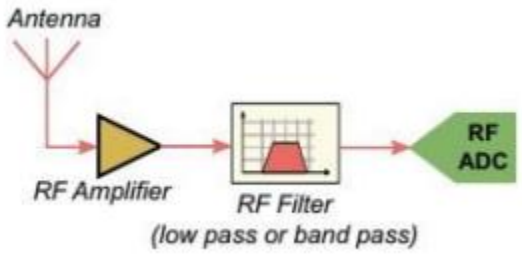


Figure 4.1a: Direct RF Receiver Architecture

The block diagram of the direct RF receiver is shown in figure . In block diagram both RF amplifier and anti-aliasing RF lowpass filter of the analog signal conditioning is external to RFSoC device. The spectrum of the signal after analogue signal conditioning is shown in Figure. Here, the lowpass filter pass signals in the Nyquist zone 1 (0 to 2GHz) and blocks all signal in the Nyquist zone 2. The output of the RF lowpass filter is input to RF-ADC sampling at fs = 4GHz. At this stage the output of RFADC is a real signal with a two-sided complex spectrum cantered at 0 Hz. The spectrum after RFADC stage is shown in Figure. The complex digital mixture at-1250 MHz is used to shift down the positive band of the spectrum to baseband and centered at 0 Hz as shown in Figure. Using the fine mixer NCO in the RFSoC and selecting a frequency of , the pair of spectra are shifted leftwards in the frequency spectrum plot by 1250 MHz. As a result, the complex mixed version of the signal are generated at-750 MHz to 750 Mhz and-3250 Mhz to 1750 Mhz. The result of the demodulation is shown in Figure. To extract the1500 MHz bandwidth signal from -750 MHz to 750 MHz, the lowpass decimation filter from -1GHz to 1GHz (bandwidth of the filter is 2 GHz) issued. To down sample the sampling frequency to 1GHz, the decimation factor of 4 is chosen. The spectrum after demodulation and decimation is shown in Figure Following this Direct RF receiver, the separate I and Q signals at the rate of F dsp equals to 1GHz are then passed to the PL for the next stage of processing.

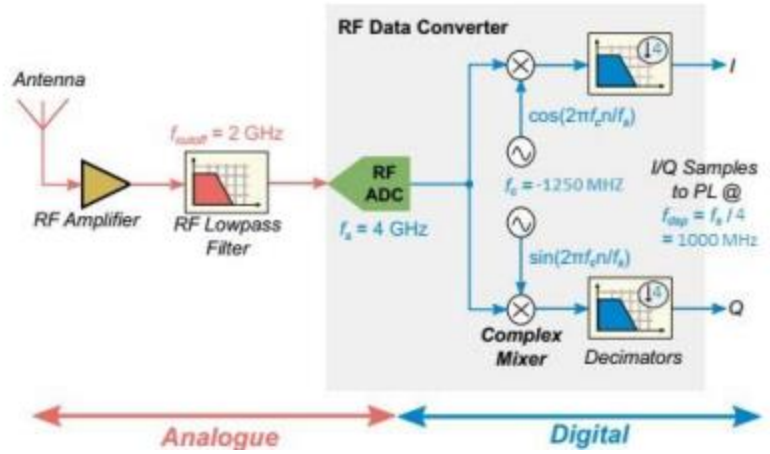


Figure 4.1b: Direct RF Quadrature Mixture Receiver

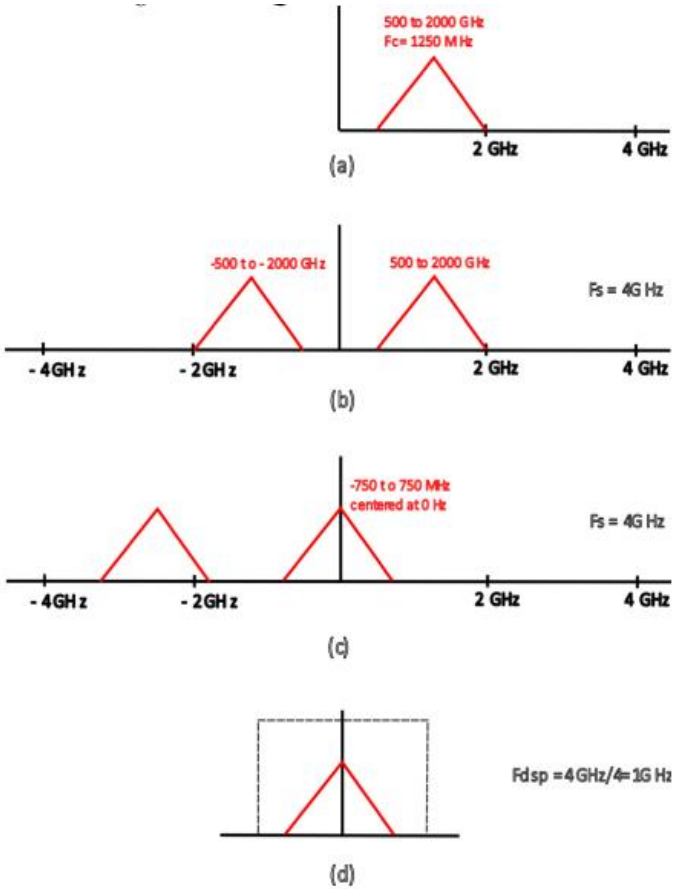


Figure 4.1c:(a) input to RF-ADC after anti-aliasing filter

(b)Two Sided Symmetric Spectrum Of Real Digitized Output From RF ADC

(c) Signal After Mixing Stage.(d) Final Demodulated And Decimated Signal.

4.2 Pulse parameter extraction:

The receiver receives the signal using directional antennas and processes these signals to get the required information about the target. The correct measurement of intra-pulse and inter-pulse parameters of received signals is critical for establishing their properties and countermeasures. Analog receivers have limitations in sensitivity, accuracy, and resolving time coincidence signals, while digital receivers use FFT processing to extract parameters in the frequency domain, overcoming these limitations. The high sensitivity of the digital receiver is achieved through FFT processing gain, and accuracy is improved by processing signals at higher point FFT. Using a digital receiver facilitates radar signal detection and provides a range advantage. Pulse radar waveform can be defined by: 1) pulse width 2) carrier frequency 4) pulse amplitude and 5) Time of Arrival.

4.2.1 Frequency:

To find the frequency of the input signal using Fast Fourier Transform (FFT), we need to know the sampling frequency (fs), local oscillator frequency (fLO), and frame length (F). in case of direct sampling, the sampling frequency is the rate at which the signal is sampled, and it should be more than twice the signal's maximum frequency according to the Nyquist sampling theorem [14]. The signal is down-converted to an intermediate frequency using a mixer, and this process is known as local oscillator down-conversion [15]. The frame length is the number of samples used in the FFT calculation [16]. The frequency resolution of the FFT is given by the formula:

$$\Delta f = fs/F$$

where Δf is the frequency resolution, fs is the sampling frequency, and F is the frame length [17]. The frequency of the input signal can be estimated by finding the peak frequency in the FFT output. The peak frequency corresponds to the frequency component with the highest amplitude in the frequency domain [16]. The frequency of the input signal can be calculated using the following formula:

$$f = fLO \pm k\Delta f$$

where f is the frequency of the input signal, fLO is the local oscillator frequency, k is the index of the peak frequency in the FFT output, and Δf is the frequency resolution [15]

4.2.2 Amplitude:

The amplitude of the received radar signal is calculated using the FFT spectrum. The ADC's 14-bit data input is fed to the FPGA for N-point FFT. The FFT output will include both real and imaginary data. The peak amplitude is obtained by squaring the sum of the peak values of the real and imaginary FFT outputs. The PA parameter is calculated using the peak value of the processed

data in a specific pulse. Accurate pulse amplitude measurement can help generate accurate range estimation for usage in a variety of applications.

4.2.3 Pulse width:

The pulse width of the radar signal can be obtained by calculating the time difference between the leading and trailing edges of the pulse. The leading and trailing edges can be detected by thresholding the signal and finding the first and last points that cross the threshold. The pulse width can then be calculated as the time difference between the two edges [10]. The equation for calculating the pulse width of a radar signal using FFT is:

$$Pulse\ Width = (Trailing\ Edge - Leading\ Edge) * ts$$

Where ts the sampling is the sampling time of the radar signal,

4.2.4 Pulse Train Signal:

A pulse wave or pulse train is a type of non-sinusoidal waveform that includes square waves (duty cycle of 50%) and similarly periodic but asymmetrical waves (duty cycles other than 50%).

A digital signal in digital electronics is a pulse train (a pulse amplitude modulated signal), which is a series of electrical or light pulses with fixed-width square waves, each occupying one of two distinct levels of amplitude. Due to their quick on-off electronic switching characteristic, metal-oxide-semiconductor field-effect transistor (MOSFET) devices are often used to produce these electronic pulse trains as opposed to BJT transistors, which slowly produce signals more akin to sine waves.

4.3 Signal Generation:

4.3.1 Simulink Model:

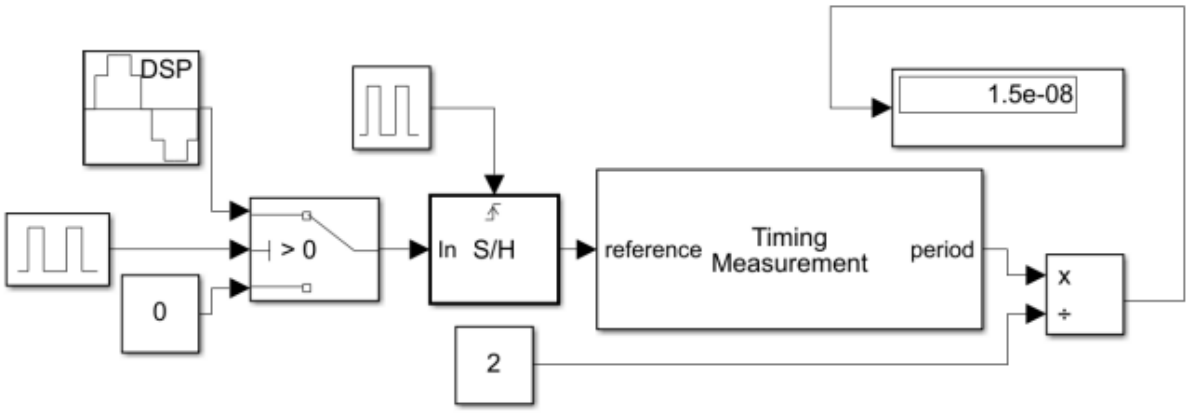


Figure 4.2.2 : signal Generation Block Diagram

4.3.2 Blocks Specifications:

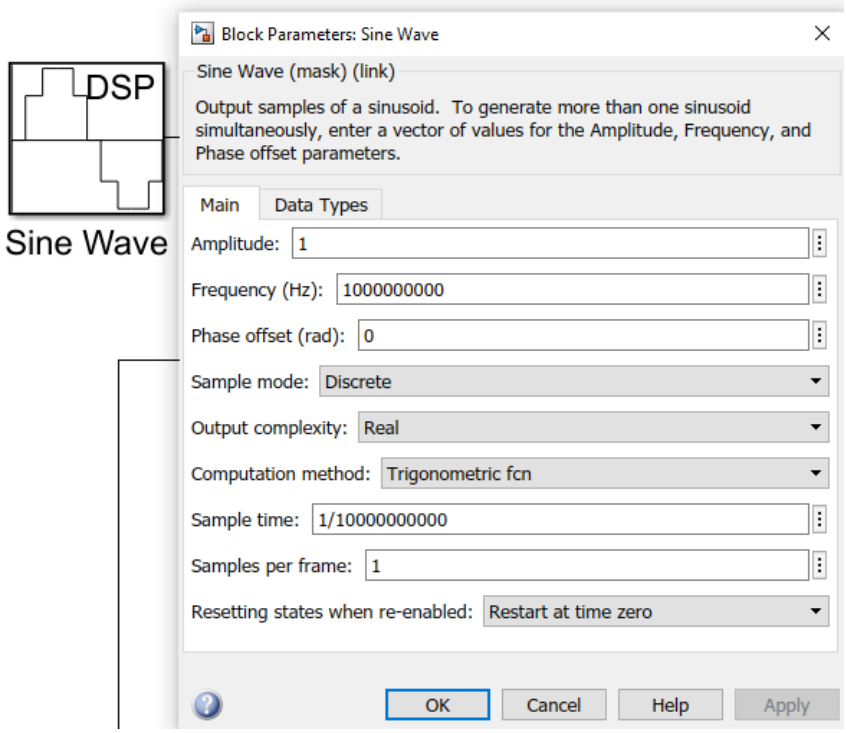


Figure 4.3.2a: Sine Wave Block Parameters

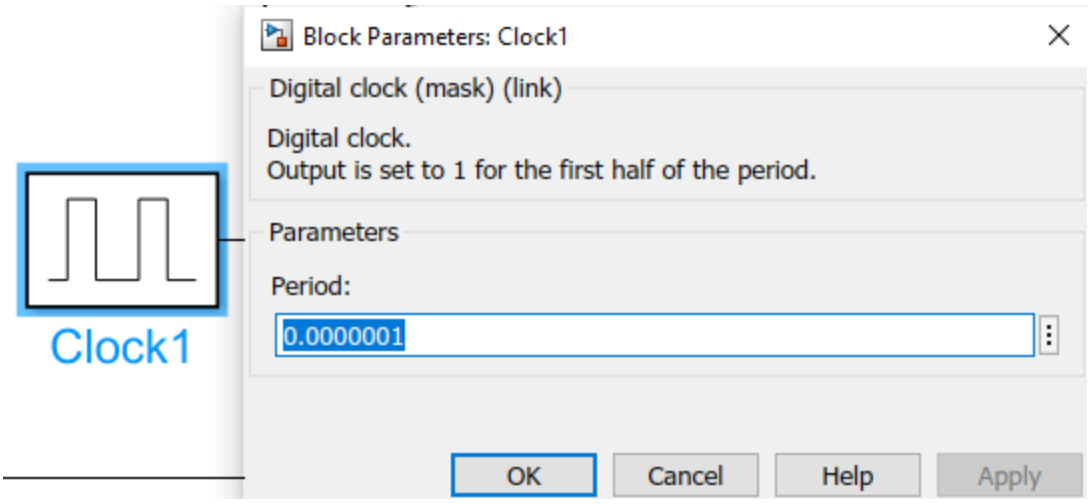


Figure 4.3.2b: Clock Parameters Selection

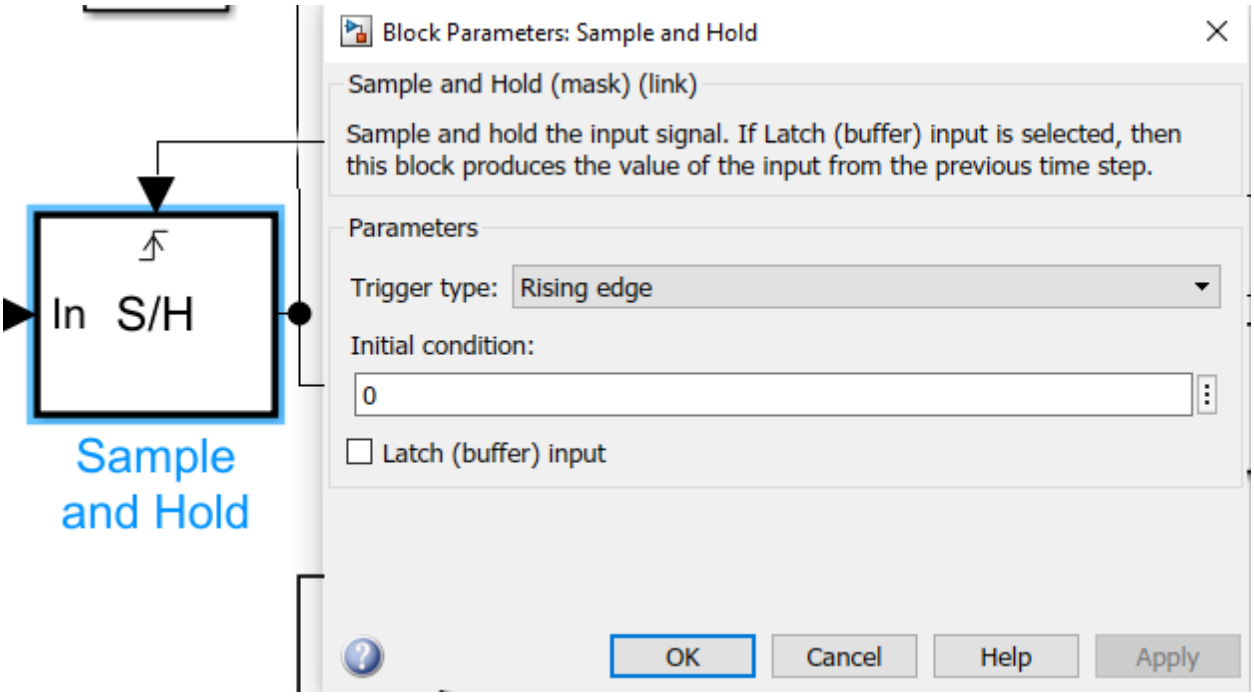


Figure 4.3.2c: S/H Block Parameter Selection

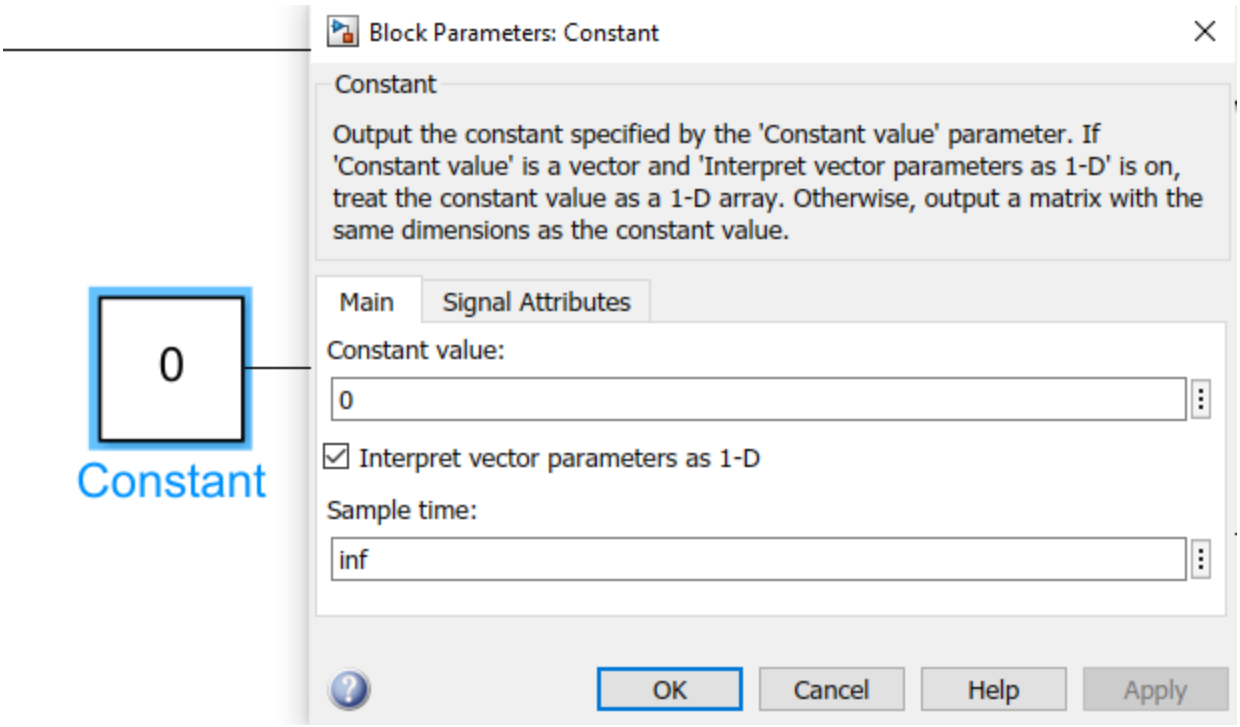


Figure 4.3.2d: Constant Block Parameters

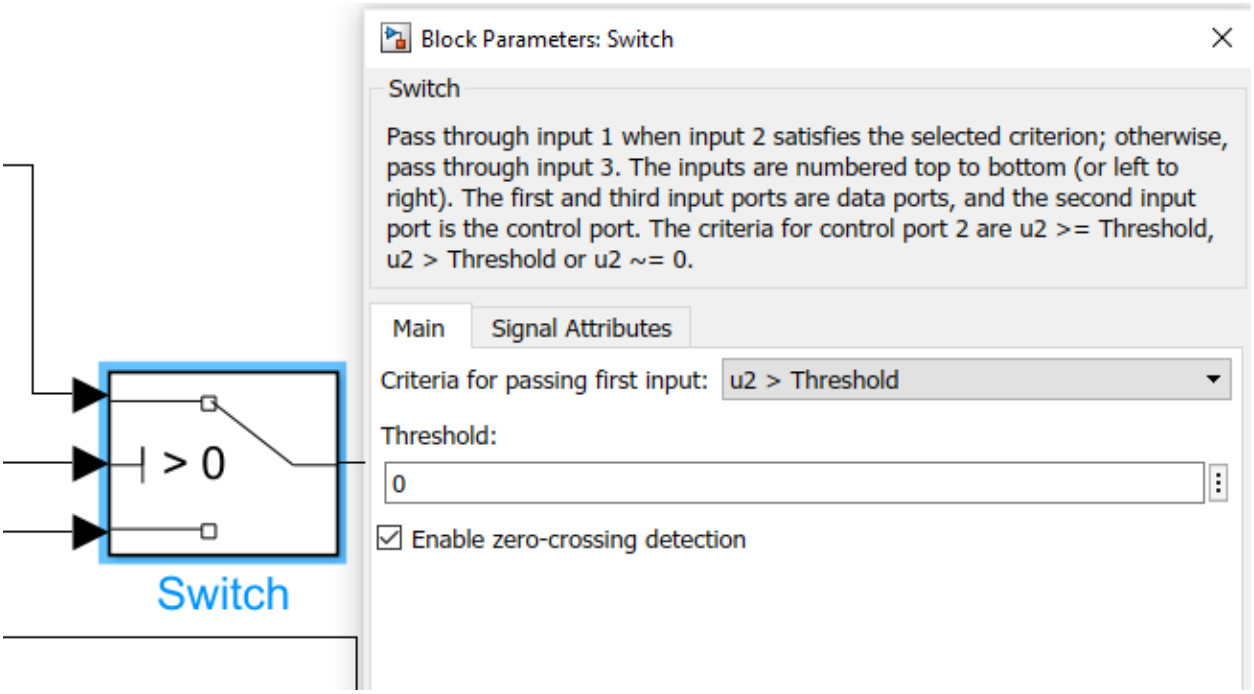


Figure 4.3.2e: Switch Block Parameters

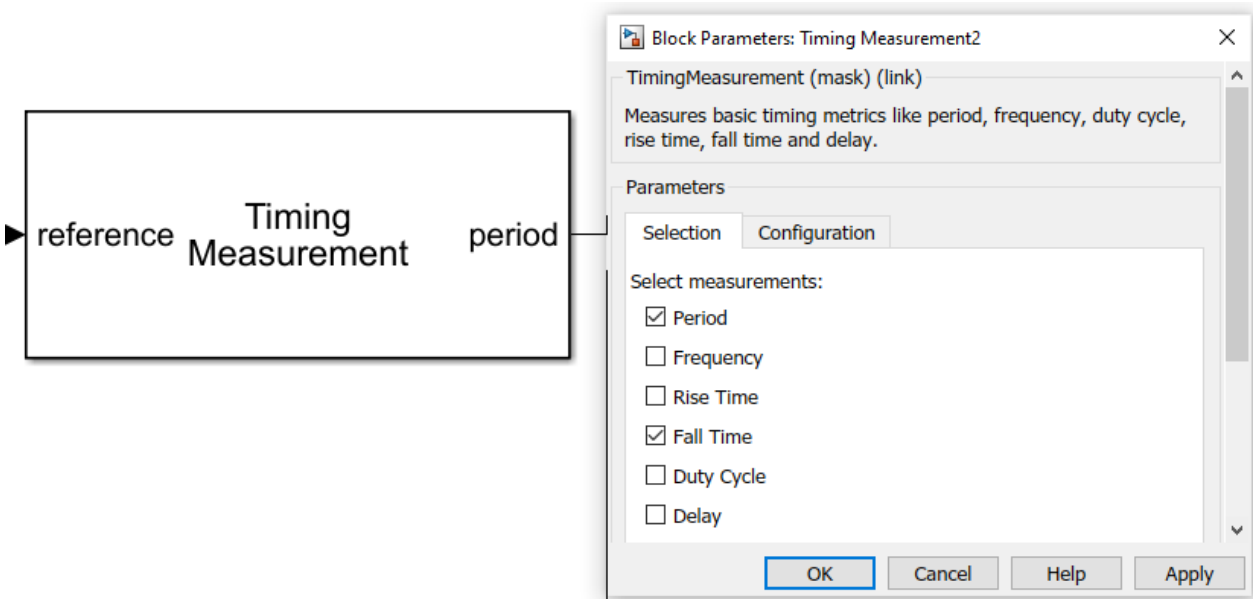


Figure 4.3.2f: Time Measurement Block Parameters

4.4 Simulation Results:

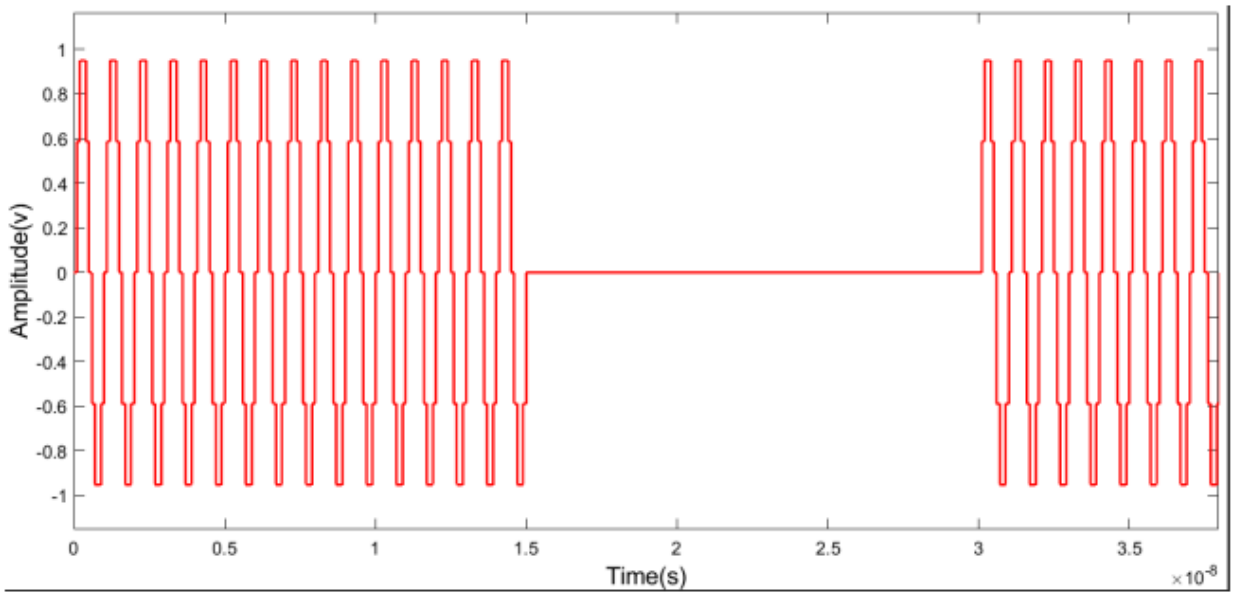


Figure 4.4: Pulse Train Signal Generation

4.4.1 Display Ports:

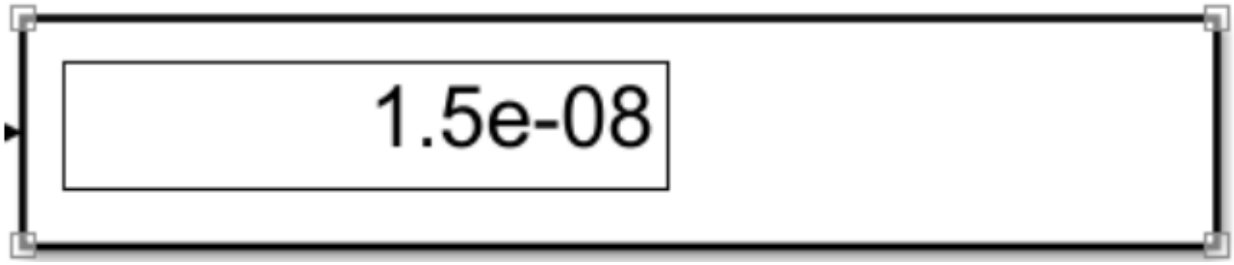


Figure 4.4.1a: Pulse Width Parameter Display

TOD=15ns

TOA=0.1ns

Figure 4.4.1b: TOA and TOD Parameter Display

4.5 FFT Algorithm:

4.5.1 PYNQ :

An open-source initiative from AMD® called PYNQ simplifies the use of platforms for adaptive computing.Designers may take advantage of programmable logic and microprocessors to create more powerful and interesting electronic devices by utilising the Python programming language and libraries.High performance applications can be created using PYNQ with Zynq, Zynq UltraScale+, Zynq RFSoC, Alveo accelerator boards, and AWS-F1 by using the following:

1) concurrent hardware operation
2) high frame rate hardware-accelerated techniques for video processing
3) processing signals in real time
4) high bandwidth control with low latency input

4.6 Purpose Of PYNQ:

A broad spectrum of designers and developers are expected to adopt PYNQ, including:

Developers of software that wish to leverage the features of Adaptive Computing platforms without needing to construct hardware with ASIC-style design tools.System architects who wish to quickly prototype and build their Zynq, Alveo, and AWS-F1 designs using an intuitive software interface and architecture.designers of hardware whose goal is to have as many people as possible use their designs.

4.7 PYNQ Technoologies:

4.7.1 Jupiter NoteBook:

An interactive computing environment based on a browser is called Jupyter Notebook. It is possible to generate Jupyter notebook papers with live code, interactive widgets, graphs, explanation text, equations, pictures, and videos.

4.7.2 Python NoteBook:

Python can be readily programmed in a Jupyter Notebook to control a PYNQ capable board.Developers can leverage hardware overlays and libraries on the programmable logic with Python. Software operating on a Zynq or Alveo board can be accelerated via hardware overlays, or libraries, which can also be used to modify the hardware platform and interfaces.

4.8 PYNQ framework:

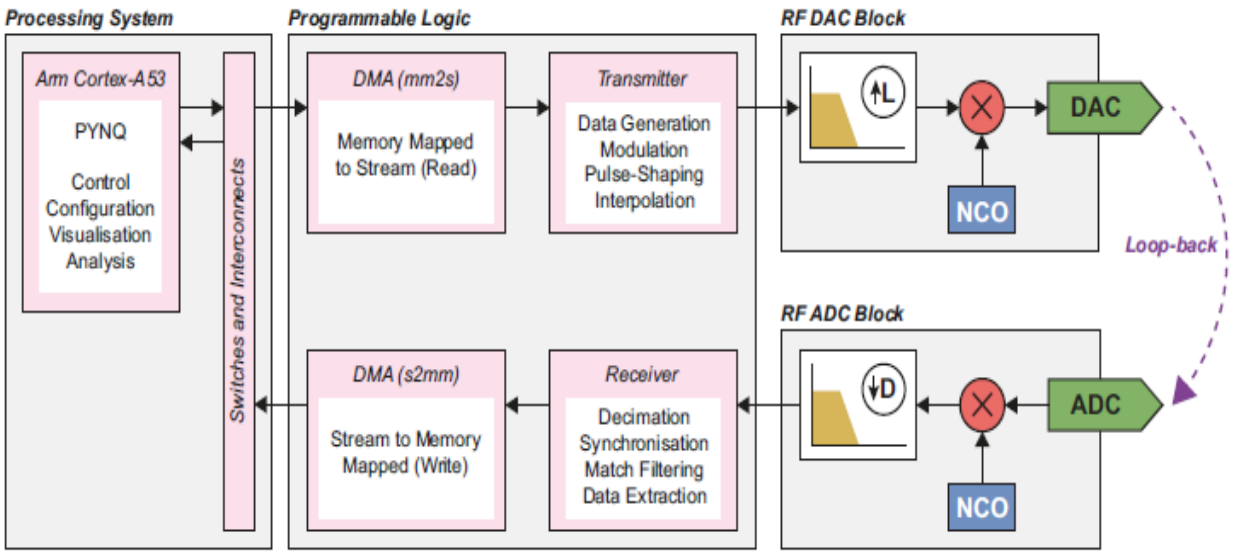


Figure 4.8: Architecture Of PYNQ

CHAPTER 05

ZCU111+RFSOC BOARD IMPLEMENTATION

5.1 Introduction:

5.1.1 Overview:

One of the devices on the ZCU111 evaluation board is the Zynq UltraScale+TM RFSoC ZCU28DR. With the help of this board, it is possible to assess the integrated RF-DAC and RF-ADC capability, soft decision forward error correction (SDFEC), FPGA fabric, and RFSoC features, including the dual-core Arm Cortex-R5 real-time processor and the quad-core Arm CortexTM-A53 processing system (PS). Many of the standard board-level capabilities required for design development are present on the ZCU111 evaluation board, including DDR4 memory, networking interfaces, an FMC+ expansion connection, and access to the recently introduced RF-FMC interface.

An FMC XM500 balun transformer add-on card is included with the ZCU111 evaluation board kit right out of the box to facilitate loopback evaluation and signal processing. On-board high-frequency and low-frequency baluns as well as SMAs for custom baluns and filters are included with this card. For additional details

5.2 Block Diagram:

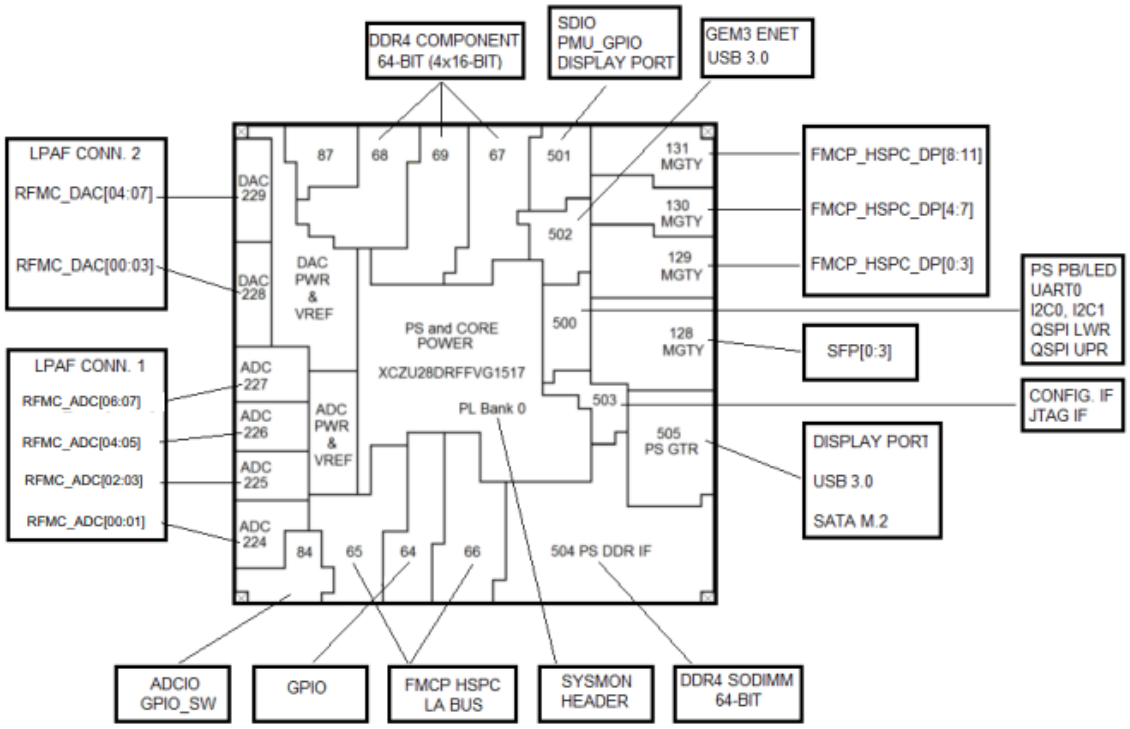


Figure 5.2 : ZCU111 Evaluation Board Block Diagram

5.3 ZCU111+RFSoc Board:

5.3.1 Overview:

The Zynq UltraScale+ RFSoC ZCU111 evaluation kit, which Xilinx recently released in 2018, garnered a lot of interest from the industry due to its utilisation of Zynq UltraScale+ RFSoC, which combines FPGA logic with multi-Gb ADC and DAC sampling functionalities. Additionally, it can offer a quick and thorough RF analogue-digital signal chain prototype platform so that customers can experience this ground-breaking technology firsthand.
The industry's first evaluation kit of its kind is the Xilinx Zynq UltraScale+ RFSoC ZCU111. Every other kind of RF-ADC or DAC has a different architecture. ADC/DAC daughtercard and FPGA evaluation card must be purchased, and connections must be made via FMC or other connectors. Distinct implementations have certain difficulties concerning both usability and design. A significant amount of power is used by the discrete ADC/DAC solution's high-speed transceiver and the serial connection, such as the JESD204 standard interface, that connects the DAC/ADC to the FPGA. In high-channel count applications (like 8x8), this will result in an additional 28W of power usage.

On the market, there are other ADC/DAC plus FPGA evaluation cards that require the use of complementary tools from two distinct chip makers. Two different manufacturers make tools: one is used to emulate ADC/DAC devices, while the other is used to modify FPGA designs. As a result, the evaluation and prototyping processes will move more slowly. Besides, you might not know who to contact for support from a manufacturer when you run into issues. Just one manufacturer, one device, and one reference design are required for the Zynq UltraScale+ RFSoC ZCU111 evaluation kit. This simplifies the process, allows for the management of FPGAs, and integrates DAC/ADC.

5.3.2 8x8 assessment feature:

The Zynq UltraScale+ RFSoC ZCU111 evaluation kit offers integration benefits such as decreased component count, low power consumption, a smaller board area, and 8x8 ADC/DAC. The cost savings are thirty to fifty percent as compared to buying an evaluation system that includes an 8x8 ADC/DAC and an FPGA function kit. To test the same configuration, different possibilities call for up to four ADC/DAC boards in addition to one FPGA board.

5.4 Board Specification:

Table 5.4 : ZCU111+RFSOC Board Blocks Specification

Board Specifications	Value
Height	11.811 inch (30.0 cm)
Width	7.874 inch (20.0 cm)
Thickness (+/-5%)	0.10799 inches (0.2743 cm)

Operating Environmental Temperature	0˚C to +45˚C
RF Data Converter	
# of 12-bit ADCs	8
Max Rate (GSPS)	4.096
# of 14-bit DACs	8
Max Rate (GSPS)	6.554
SD-FEC	8
Memory	
PS DDR4	4GB 64-bit SODIMM
SD-Card	Yes
M.2 SATA Connector	Yes
QSPI	2
Communications & Networking	
USB UART/JTAG	1
RJ45	1
SFP+	4
USB 3.0	1
Display	
DIP Switches	Yes

LEDs	Yes
Push Buttons	Yes
VESA DisplayPort 1.2	Yes
Expansion Connectors	
FMC-HPC Connector	2
PMOD	2
RFMC 1.0	2
Add-on Cards	
XM500	Yes
Control & I/O	
I2C	Yes
PMBUS	Yes
JTAG PC4 Header	Yes
Boot Options	
SD Boot	Yes
QSPI Boot	Yes
JTAG Boot	Yes
Power	
12V Wall Adapter	Yes
ATX Power Compatible	Yes

5.5 ZCU111+RFSoC Board :

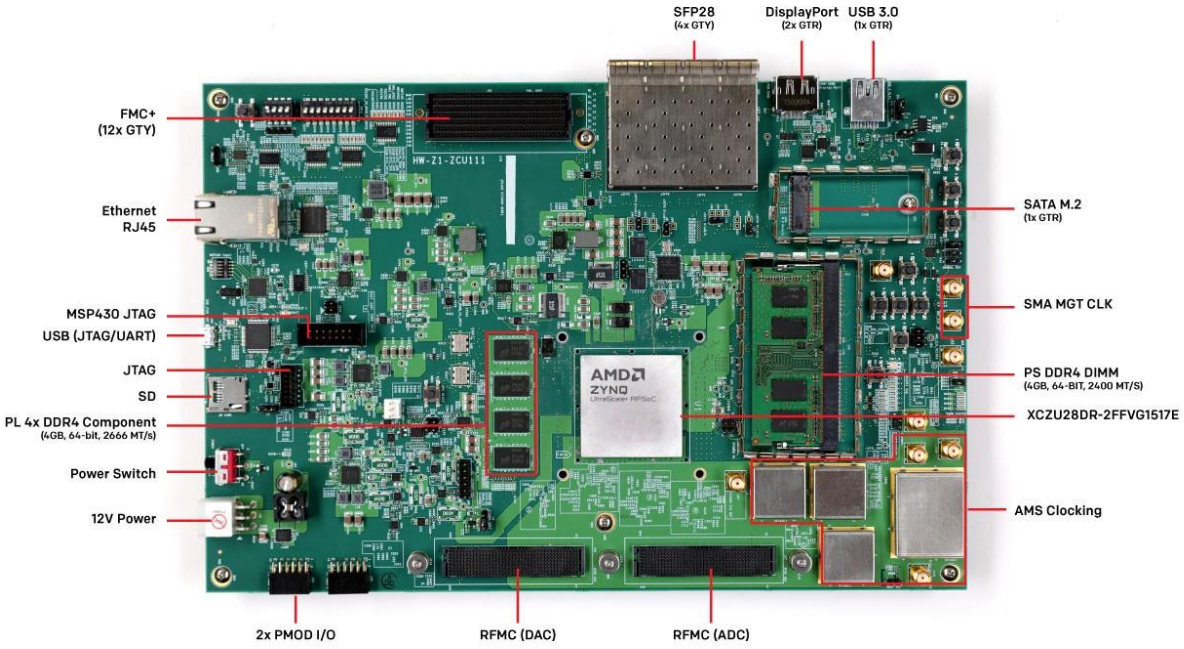


Figure 5.5: ZCU111+RFSOC Board

5.6 Board Implementation:

Figure 5.6a: ZCU111+RFSOC Board

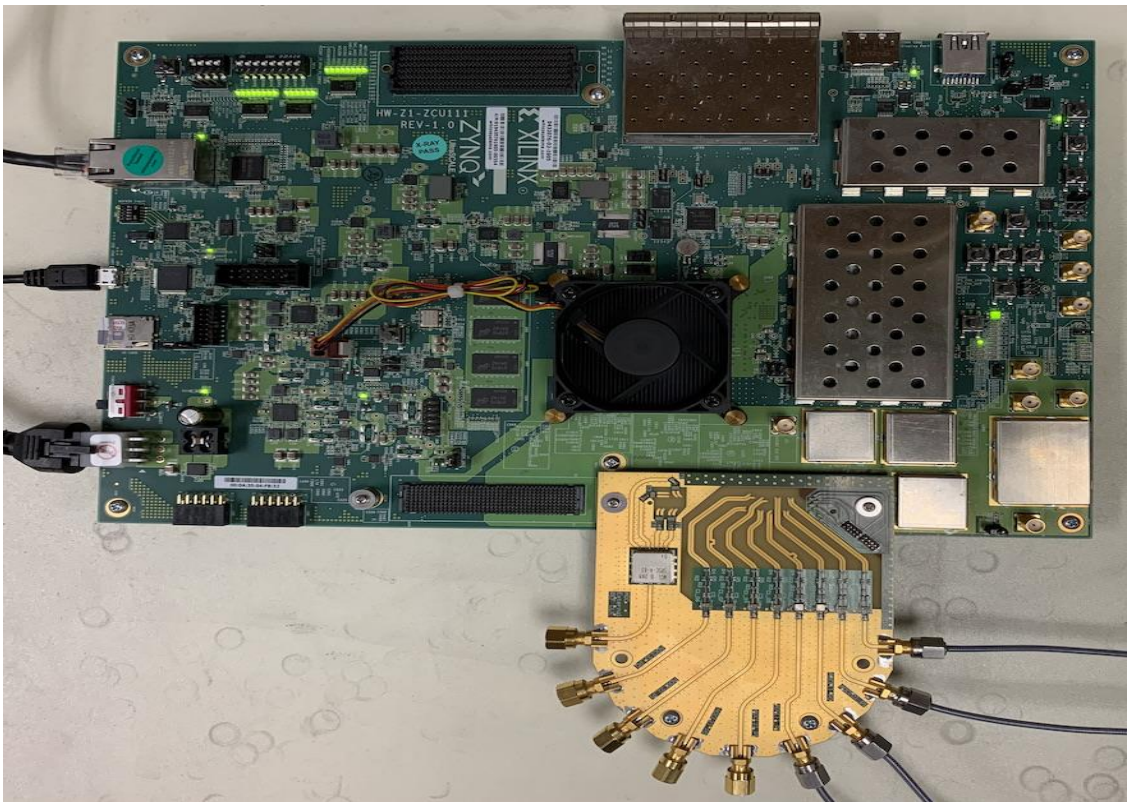


Figure 5.6b: ZCU111+RFSOC Board Loop Back System

5.7 Main performance and advantages:

1) ZCU111 test board for Zynq UltraScale+ RFSoC supporting XCZU28DR-2FFVG1517E RFSoC
2) 4GB, 64-bit, 2666 MT/s DDR4 components are coupled to programmable logic (PL)
3) 4GB, 64-bit, 2400 MT/s DDR4 SODIMM, linked to the processor subsystem (PS)
4) Up to 4 SFP/SFP+/zSFP+/SFP28 modules can be supported by a Linkage SFP28 module.
5) SATA, USB3, and DisplayPort
6) 34 user-defined differential I/O signals and 12 32.75 Gb/s GTY transceivers are included in the FPGA Mezzanine Card (FMC) interface for I/O expansion.
7) The plug-in card for the XM500 RFMC weighing converter supports For the weighing converter, 4 DACs/4 ADCs To SMA, 4 DACs/4 ADCs.

5.8 Conclusion :

For wideband high frequency transmissions, digitizing the receiver offers a solution and is crucial for reaching the requisite precision and specification. The study investigates how a digitizing receiver can be mapped to a ZCU111-based Radio Frequency System on Chip (RFSoC) from Xilinx. The Zynq UltraScale+RFSoC devices include an ARM real-time processing unit (RPU), an ARM multi-processor embedded ARM Cortex-A53 application processing unit (APU), and integrated ADC (up to 16 12-bit channels sampling at 4.0 GSPS and DAC (up to 16 14-bit channels sampling at 6.4 GSPS). Many analog signal processing operations, which usually take place close to the antenna in a digitizing receiver, can be moved into the digital domain by integrating all of these components.

Extracting the different parameters like time of arrival, time of departure, pulse width, pulse repetition frequency, pulse amplitude of the received signal for estimating their characteristics of specific desire signal is successfully achieved Implementation of these logics on board that by using advance technology is a major achievement.

5.9 Future Work:

One tool that shows promise for reading out huge arrays of microwave kinetic inductance detectors (MKIDs) is the Xilinx ZCU111 Radio Frequency System on Chip (RFSoC). The board has eight 12-bit/4.096 giga samples-per-second (GSPS) analog-to-digital converters (ADCs) and eight 14-bit/6.554 GSPS digital-to-analog converters (DACs). It also has 4,272 DSP slices and 930,000 logic cells of field-programmable gate array (FPGA) resources. Even though this data converter bandwidth is enough to read out 8,000 MKIDs at a 2 MHz channel spacing and a 1 MHz sampling rate (per channel), it doesn't match up with the ZCU111's FPGA resources. This means that more FPGA resources are needed to do the DSP needed to process this many MKIDs. This issue can be resolved using the new Xilinx RFSoC 2x2 board. With only a quarter of the data converter

resources, this board offers the same logic resources for a fifth of the cost of the ZCU111. By using multiple RFSoC 2x2 boards, it would be possible to achieve a better balance between FPGA resources and data converters, lowering the cost per pixel of the readout system from roughly €2.50 per pixel with the ZCU111 to €1 per pixel with the 2x2. This would also allow for the full utilisation of the RF bandwidth provided by the data converters.

We introduce the Quantum Instrumentation Control Kit (abbreviated QICK), which is a Xilinx RF System-on-Chip (RFSoC)-based qubit controller that makes it possible to directly create control pulses with carrier frequencies as high as 6 GHz. The QICK is capable of controlling multiple qubits or other quantum devices. Comprising an RFSoC field-programmable gate array, bespoke firmware, custom software, and an optional companion custom-designed analogue front-end board, the QICK is comprised of a digital board. We describe the system's digital latency and analogue performance, which are crucial for quantum error correction and feedback protocols. We do conventional characterizations of a transmon qubit in order to benchmark the controller. We attain a Favg of 99.93% average gate fidelity. Software, firmware, and schematics are all available as open-source projects.

REFERENCES

[1] Li Q, Yang D, Mu XH, Huo QL. Design of the L-band wideband LFM signal generator based on DDS and frequency multiplication. Proceedings of the International Conference on Microwave and Millimeter Wave Technology (ICMMT); 2012. p. 1–4. doi: 10.1109/ICMMT.2012.6230321.

[2] Pallavi N, Anjaneyulu P, Reddy PB, Mahendra V, Karthik R. Design and implementation of linear frequency modulated waveform using DDS and FPGA. Proceedings of the International Conference of Electronics, Communication and Aerospace Technology; 2017, p. 237–241. doi: 0.1109/ICECA.2017.8212806.

[3] Chekka AB, Aggala NJ. High frequency chirp signal generator using multi DDS approach on FPGA. Proceedings of the International Conference on Trends in Electronics and Informatics; 2021. p. 137–142. doi: 10.1109 /ICOEI51242.2021.9453012.

[4] Li, Tang, Vishwakarma, Woodbridge, Chetty (2022). Design of high-speed software defined radar with GPU accelerator. IET Radar Sonar & Navi, 7(16), 1083-1094. https://doi.org/10.1049/rsn2.12244

[5] Olivier, Gouws (2013). Modern wideband DRFM architecture and real-time DSP capabilities for radar test and evaluation. https://doi.org/10.1109/siecpc.2013.6551019

[6] Yu, Wang, Chen, Yang (2014). Design of Digital Pulse Compression System Based on FPGA. AMR, (1049-1050), 1718-1721.

[7] Samarah (2014). A 320 MHz Digital Linear Frequency Modulated Signal Generator for Radar

Applications Using FPGA Technology.. https://doi.org/10.1109/cicn.2014.204

[8] Wang, Liang, Wang, Yu, Liu (2015). Design and Implementation of a FPGA and DSP Based MIMO.Radar Imaging System. RADIOENGINEERING, 2(24), 518-526.https://doi.org/10.13164/re.2015.0518

[9] Davidson, Bray (2020). Understanding Digital Radio Frequency Memory Performance in

Countermeasure Design. Applied Sciences, 12(10), 4123. https://doi.org/10.3390/app10124123

[10] Fagan R, Robey FC, Miller L. Phased array radar cost reduction through the use of commercial RF systems on a chip. Proceedings of the IEEE Radar Conference (RadarConf18); 2018, p. 0935–0939. doi:10.1109/RADAR .2018.8378686.

[11] Ispir M, Yildirim A. Real-time signal generator for noise radar. IEEE Aerospace and Electronic Systems Magazine. 2020 Sep;35(9):42–49. doi: 10.1109/MAES.2020.2997415.

[12] Norheim-Naess I, Finden E. DVB-T passive radar experimental comparisons of a custom made passive radar receiver, RFSoC and a Software Defined Radio. 21st International Radar Symposium (IRS), 2021. p. 1-10. doi: 10.23919/IRS51887.2021.9466177.

[13] D. Kim, H. Kang, Y. Son, S. Kim and S. Nam, "Commercial RFSoC-Based Wideband MIMOFMCW Radar Design with Effective Pre-distortion," 2022 IEEE Radar Conference (RadarConf22), NewYork City, NY, USA, 2022, pp. 01-04, doi: 10.1109/RadarConf2248738.2022.9764323.

[14] Ramos-Cenzano, Ogueta-Gutiérrez, Pindado (2019). On the output frequency measurement within cup anemometer calibrations.Measurement, (136), 718-723. https://doi.org/10.1016/j.measurement.2019.01.015