ARL-TR-9373 ● JAN 2022

DEVCOM ARMY RESEARCH LABORATORY

Xilinx RF System-on-Chip (RFSoC) 100 Gigabit Ethernet Loop-back Demonstration

by William Diehl and Edward Viveiros

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-9373 ● JAN 2022

Xilinx RF System-on-Chip (RFSoC) 100 Gigabit Ethernet Loop-back Demonstration

William Diehl and Edward Viveiros
Sensors and Electron Devices Directorate, DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
January 2022	Technical Report	June–November 2021

4. TITLE AND SUBTITLE
Xilinx RF System-on-Chip (RFSoC) 100 Gigabit Ethernet Loop-back Demonstration

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
William Diehl and Edward Viveiros

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
DEVCOM Army Research Laboratory
ATTN: FCDD-RLS-EW
2800 Powder Mill Rd
Adelphi MD 20783

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-9373

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID IDs: Diehl, 0000-0002-6293-5018; Viveiros, 0000-0002-5116-1010

14. ABSTRACT
The ability to move high quantities of instantaneous RF bandwidth data to or from transmitter or receiver processing hardware is paramount to realizing the value of modern RF system-on-chips (RFSoCs), which are staples of many software-defined radio applications. However, practically establishing, initiating, and verifying a working 100 Gigabit Ethernet (100GigE) construction in a high-speed RF construct is not trivial. This report describes practical procedures to implement and observe a 100GigE loop-back demonstration on the Xilinx Generation 1 RFSoC. The authors first discuss the proper initialization of 100GigE hardware and its interconnected RF transmit and receive architecture. Then demonstrated are several working 100GigE loop-back configurations, including single and dual analog-to-Ethernet-to-digital transmissions. Output artifacts are observed on external oscilloscopes, and transmission latency attributed to 100GigE is measured.

15. SUBJECT TERMS
software-defined radio, SDR, system-on-chip, RFSoC, Ethernet, 100GigE, high-speed communication, SFP28, CAUI-4, CMAC, Phased Locked Loop, PLL

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			William Diehl
Unclassified	Unclassified	Unclassified	UU	38	19b. TELEPHONE NUMBER (Include area code) (301) 394-0931

Contents

List of Figures v

Acknowledgments vi

1. Introduction 1

2. Background 2

2.1 Required Throughput for High-Speed Communications 2

2.2 ZCU111 Physical Accommodations for High-Speed Communications 3

2.3 100 Gigabit Ethernet (100GigE) Standard 4

2.4 Xilinx CMAC UltraScale+ Intellectual Property Module 4

2.5 SFP28 Loop-back Connections 5

3. CMAC Clocking Considerations 6

3.1 Gigabit Transceiver Reference Clock Domain 7

3.2 System Clock Source Domain 7

3.3 Initiation Clock Domain 8

3.4 Receiver User Clock Domain 8

3.5 Transmitter User Clock Domain 8

3.6 Gigabit Transceiver Reference Clock (Out) Domain 8

3.7 AXI Control and Status Clock Domain 9

4. Verification of Correct CMAC Initiation 9

5. Software Initialization of CMAC 10

6. Radio Frequency Digital Converter 11

7. 100GigE Loop-back and Latency Demonstration 11

7.1 Linear Frequency Modulation Generator (LFMGEN) 12

7.2 100GigE Transmission Chain 12

7.3 Observations 13

8. Single-Channel ADC-to-Ethernet-to-DAC Loop-back Demonstration 14

8.1 ADC to DAC Channel 15

8.2 Observations 15

9. Two-Channel ADC to Ethernet to DAC Loop-back Demonstration 16

9.1 Dual ADC to DAC Channels 16

9.2 Observations 17

10. Conclusions and Future Work 18

11. References 19

Appendix A. Block Designs 21

Appendix B. Vivado Power Report (abridged) 24

List of Symbols, Abbreviations, and Acronyms 28

Distribution List 30

List of Figures

Fig. 1 Two RFSoCs connected by a 100GigE link 1
Fig. 2 Xilinx RFSoC with Quad SFP28 interface highlighted 3
Fig. 3 Xilinx CMAC IP usage between transmit and receive user domains ... 5
Fig. 4 SFP28 Passive Loop-back Module 6
Fig. 5 Four SFP28 Passive Loop-back Modules inserted into Quad SFP28 Cage 6
Fig. 6 CMAC IP clock configuration 6
Fig. 7 Connection of CMAC to AXI Timers 10
Fig. 8 Basic code sequences to initialize CMAC in a software construct 11
Fig. 9 100GigE loop-back and latency demonstration 12
Fig. 10 Oscilloscope output of 100GigE loop-back demonstration and latency test 14
Fig. 11 Single-channel ADC-to-Ethernet-to-DAC loop-back demonstration. 14
Fig. 12 Output of single-channel ADC-to-Ethernet-to-DAC loop-back demonstration 15
Fig. 13 Two-channel ADC-to-Ethernet-to-DAC loop-back demonstration 16
Fig. 14 Output of two-channel ADC-to-Ethernet-to-DAC loop-back demonstration 17
Fig. 15 RF split routing for ZCU111 Ethernet loop-back testing 17
Fig. A-1 100GigE loop-back demonstration block design 22
Fig. A-2 Two-channel ADC-to-Ethernet-to-DAC loop-back demonstration 23

Acknowledgments

The authors would like to acknowledge B Kirk of DEVCOM ARL, D Galanos of Huntington Ingalls Industries (DEVCOM ARL contractor), and M Franklin of Massachusetts Institute of Technology Lincoln Labs for their helpful suggestions.

1. Introduction

The radio frequency (RF) system-on-chip, or RFSoC, is an emerging paradigm in RF engineering. Specifically, it combines the flexibility of embedded processing power with tightly coupled RF analog-to-digital (ADC) and digital-to-analog (DAC) converters on a single chip. This greatly reduces the design complexity of adding so-called "RF front ends" and generally reduces size, weight, and power (SWaP) requirements. RFSoCs come in many forms and with varying degrees of sophistication; some are designed for initial laboratory demonstrations and prototypes, and some come in reduced form-factor and are designed for deployable or production applications. In general, the RFSoC has greatly accelerated the market for software-defined radios (SDR); i.e., RF-capable computational devices that can be reprogrammed and reconfigured at various layers of abstraction, from factory, to laboratory, to production floor, and even end-user.

While much of the attention on RFSoCs is devoted to either RF conversion (ADC and DAC) or processing power (processors, field-programmable gate arrays [FPGAs], etc.), high-speed digital communication, either between boards or from board-to-computer or computer-to-board, is a critical requirement for RFSoC use in high-performance SDR applications. One popular method of high-speed communication is 100 Gigabit Ethernet (100GigE), which allows for approximately 100 gigabits per second (Gbps) of data communication, depending on the configuration. Figure 1 illustrates a two-board scenario, where one RFSoC (RFSoC 1) digitizes RF data with its ADCs and performs a transformation on the data $h_1(t)$. It then forwards the streaming data to a second board (RFSoC 2) over a 100GigE connection, where it is subjected to a second transformation $h_2(t)$ before being retransmitted using the DACs on RFSoC 2.

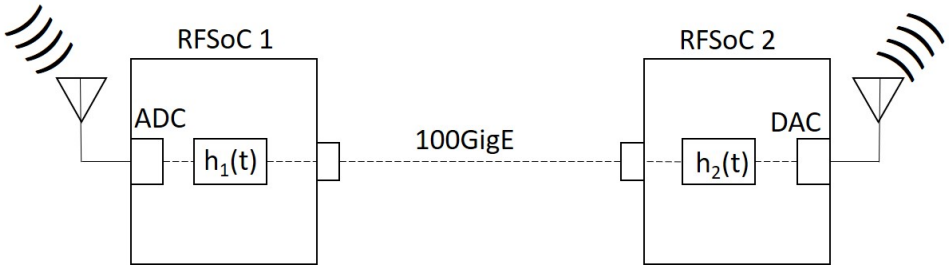

Fig. 1 Two RFSoCs connected by a 100GigE link

However, setting up a 100GigE link is challenging, even for those with significant electronics or digital engineering experience. There are many considerations to address at myriad levels of abstraction, including physical, register-transfer, block design, software, and protocol levels. While some examples of 100GigE

demonstrations can be found online or in publications, creating a working implementation for a designer's exact tailored application can still be difficult.

This report illustrates one particular 100GigE application: a 100GigE loop-back demonstration on an RFSoC. Specifically, we chose a Xilinx Generation 1 RFSoC in the ZCU111 Evaluation Board form factor and used SFP28 loop-back connectors to establish all lanes of a CAUI-4 (chip-to-module 100 Gbps four-lane Attachment Unit Interface) 100GigE link. We then instantiated, initialized, and verified the Xilinx CMAC UltraScale+ intellectual property (IP) module as part of several digital designs, which also incorporate elements of the Xilinx RF Digital Converter (RFDC) IP. Digital designs, assembled as block designs using Vivado IP Integrator, were compiled to bitstreams, exported to Vitis platform projects, and integrated through a thin hardware abstraction layer (HAL) into a Baremetal applications project with a single-core ARM Cortex A53 Processor, and controlled through a serial command line interface (CLI). Results of various demonstrations confirm the successful operation of the 100GigE loop-back mechanism and facilitate the measurement and observation of design factors (e.g., latencies) that can improve RF digital processing designs.

This demonstration is intended for the designer with a background in digital design and ideally working knowledge of the Xilinx RFSoC. All demonstrations were completed using Xilinx Vivado and Vitis 2020.1.

2. Background

These demonstrations are based on the Xilinx Gen 1 RFSoC, device nomenclature XZCU28DR, as hosted on the ZCU111 evaluation board. Notable features of the board include integrated RF-DAC and RF-ADC functionality, FPGA programmable logic (PL) fabric and quad core ARM Cortex-A53 processing system (PS), and DDR4 memory. While many of the features of the Gen 1 RFSoC are beyond the scope of this report, the ability to efficiently move data onto or off of the board at required throughput is paramount to realizing the full potential of this device. Therefore, this demonstration focuses on simple 100GigE applications.

2.1 Required Throughput for High-Speed Communications

RFSoCs are capable of processing high instantaneous bandwidth (IBW) by SDR standards; an often-used benchmark for high-performance IBW is 1 GHz. According to the Shannon-Nyquist theorem, an analog waveform must be sampled at a minimum of 2 Gsps to reproduce all elements of the signal (not including guard bands). Given the RFSoC's 12-bit ADCs and 14-bit DACs, all internal samples are represented in 16-bit signed complex numbers, meaning 16 bits of I and 16 bits of

Q data (where "I" and "Q" represent "in-phase" and "quadrature", respectively), for a total of 32 bits per sample. Thus, a minimum of 64 Gbps of throughput are required to transport even 1 GHz IBW. When we add requirements for oversampling, error correction, and Ethernet frame overhead, we rapidly approach our 100-Gbps limit. Further, 64 Gsps of throughput are sufficient for only one RF channel; there are eight such channels on the Gen 1 RFSoC, meaning that current RFSoCs are *communications-constrained*. If all data will be processed on a single RFSoC board, high-speed communication is less critical. However, if we intend to transport RF-quality data between RFSoCs, or from RFSoC to computer or vice versa, the importance of high-speed communication is evident.

2.2 ZCU111 Physical Accommodations for High-Speed Communications

The XZCU28DR RFSoC provides 16 GTY transceivers, each capable of 32.75 Gbps.[1] These are intended for communication between the PL and the outside world (the PS can also communicate at high speed via GTRs). A full description of GTY transceivers is beyond the scope of this report; more details can be found in the *UltraScale Architecture GTH Transceivers User Guide UG576*.[2]

The GTY transceivers are grouped into four (quad) banks: 128, 129, 130, and 131 on the ZCU111 evaluation board. Our demonstration concentrates on Bank 128, which is hardwired to the SFP28 interface. SFP stands for "small form factor pluggable"; the "28" refers to the supported throughput of 28 Gbps per physical connection (depending on implemented standard). The SFP physical standard is defined in *INF-8074i Specification for SFP (Small Formfactor Pluggable) Transceiver*.[3] The ZCU111 board, with the housing for the quad SFP28 connectors hardwired to GTY Bank 128 highlighted, is shown in Fig. 2.

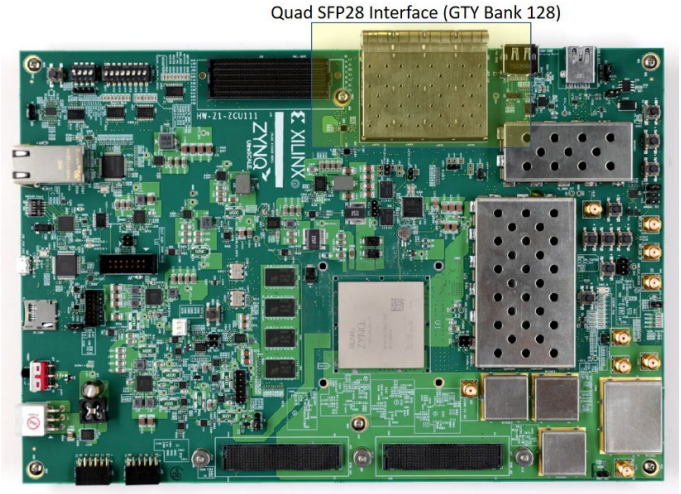

Fig. 2 Xilinx RFSoC with Quad SFP28 interface highlighted[4]

2.3 100 Gigabit Ethernet (100GigE) Standard

Ethernet is ubiquitous in computer communications, being found today in almost all computer-to-computer links, from Internet of Things and home convenience to the largest server farms. Ethernet wraps information payloads into frames, surrounds the frames with protocol and metadata designed to allow for control, status, accounting, and error correction, and sends the frames from device to device (often further encapsulated into higher protocol according to the Open Systems Interconnect, or OSI Model). Very high-speed Ethernet, in the form of 40 Gbps and 100 Gbps, has only emerged in the last decade and achieves one of the highest commercially available information throughputs. While standard Ethernet frame structures are maintained, innovations in physical transport media, such as fiber optic cables and coding techniques, are able to increase throughput. Point-to-point communication solutions based on Aurora or Interlaken can also provide very high throughput but are not considered in this report.[5,6] To implement 100 GigE on the ZCU111, we use the CMAC UltraScale+ Intellectual Property Module (described subsequently). The CMAC IP implements the IEEE 802.3-2012 standard for 100GigE.[7*]

There are many physical transport modes of 100GigE (e.g., RF, fiber optic), and several attachment interfaces (e.g., CAUI-4, CAUI-10, 100GAUI-4, 100GAUI-2, etc.). This demonstration uses CAUI-4, which is a 100-Gbps, four-lane electrical interface defined in 802.3bm Annex 83E with a nominal signaling rate for each lane of 25.78125 Gbaud using non-return-to-zero modulation.[9]

2.4 Xilinx CMAC UltraScale+ Intellectual Property Module

Xilinx provides the CMAC UltraScale+ IP module as an efficient way of implementing 100GigE in PL applications. Some advantages of the CMAC IP include access to several physical implementations of 100GigE (including CAUI-4 and CAUI-10), several interface standards between Ethernet and user applications (including local bus [LBUS] and Advanced Extensible Interface – Streaming [AXIS]), a robust control and status interface, and built-in Reed-Solomon Forward Error Correction (RS-FEC). The bit error rate (BER) over commercial optical transceivers can be unacceptably high, which is why easy access to RS-FEC provided by the CMAC IP is paramount. Datalink transport of Ethernet frames between devices is entirely encapsulated inside the CMAC and is generally transparent to higher-level (including software) applications. A thin set of drivers

[*] The IEEE 802.3-2012 standard has been superseded by the IEEE 802.3-2015 standard.[8]

is required to initialize (and optionally control and monitor) the CMAC; one such implementation is subsequently described.

In this research we implement the CAUI-4 physical interface using the Advanced Extensible Interface (AXI) Stream interface to user applications. This means that user applications (i.e., anything that manipulates RF data inside the RFSoC) will forward data to be transported via the 512-bit `axis_tx` interface. The data is then transmitted via GTY interfaces to SFP28 external connections. Internal contents of CMAC data are discussed further in the *UltraScale+™ Devices Integrated 100G Ethernet Subsystem v3.1 Product Guide*.[10] CMAC can support either simplex (one-way) or duplex (two-way) modes; we select duplex, meaning data is coming back to CMAC in the receive interface. Once frames are received, data is present at the 512-bit `axis_rx` interface for use in downstream applications. This basic arrangement is illustrated in Fig. 3. AXI Streaming protocol applies and is discussed in the *AXI Reference Guide UG761*.[11] The required clocking arrangements for the CMAC IP are complex and are subsequently described.

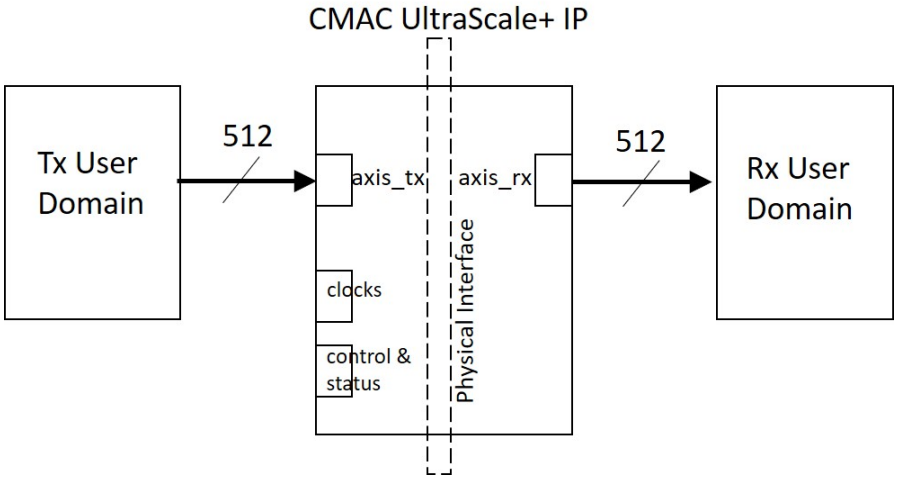

Fig. 3 Xilinx CMAC IP usage between transmit and receive user domains

2.5 SFP28 Loop-back Connections

Data departing the board on physical media can connect to SFP28 connections on other boards. However, during initial flow verification, it is often helpful to tie the transmit side of GTY transceivers directly to the receive side and allow data to "loop back" to the RFSoC for continued processing. This requires the installation of loop-back connectors. We use four Multilane SFP28 Passive Loop-back Modules (ML4026-28),[12] as shown in Fig. 4. The connectors are inserted into the Quad SFP28 assembly (four slots) on the ZCU111, as shown in Fig. 5.

Fig. 4 SFP28 Passive Loop-back Module[4]

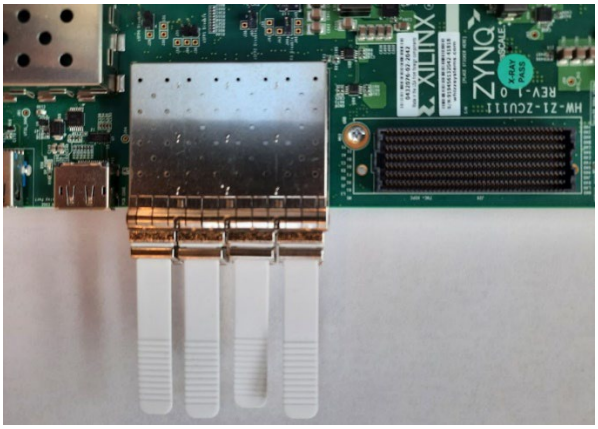

Fig. 5 Four SFP28 Passive Loop-back Modules inserted into Quad SFP28 Cage

3. CMAC Clocking Considerations

Establishment of clock domains required for use of the CMAC UltraScale+ IP is described in Xilinx's 100G Ethernet Subsystem Product Guide.[10] However, additional explanation is necessary to establish and initiate adequate clock functions to realize 100GigE. There are seven required CMAC clock domains as described in the following section and shown in Fig. 6.

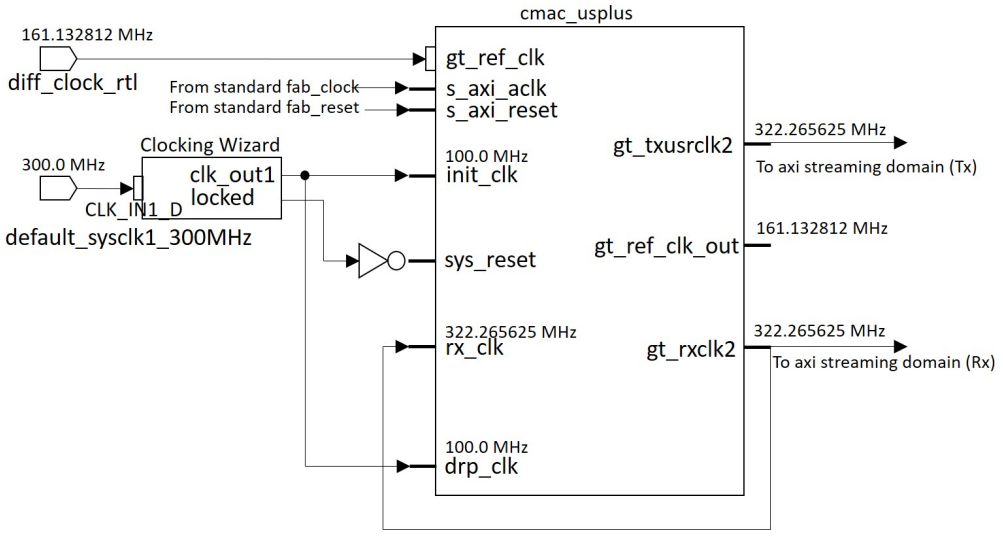

Fig. 6 CMAC IP clock configuration

3.1 Gigabit Transceiver Reference Clock Domain

This clock domain is the primary clock source for all 100GigE functions. The clock frequency is 161.132812 MHz (additional requirements are outlined in Xilinx's 100G Ethernet Subsystem Product Guide[10]). It arrives via an external port called `diff_clock_rtl`, shown in Fig. 6. This external port must be a differential clock and must connect to `gt_ref_clk` in the CMAC IP. The "run block automation" feature of Vivado IP Integrator will create the correct external port and make the connection. However, this clock must be sourced from an external suitable clock (e.g., the Silicon Labs USER_MGT_SI570 on the ZCU111). USER_MGT_SI570 must be annotated in a Xilinx Constraints File (`.xdc`), connecting package pins V31 to <clock name>`clk_p` and V32 to <clock name>`clk_n`, respectively,[1] as shown in the following annotation:

```
create_clock -period 6.206060625 [get_ports diff_clock_rtl_clk_p]
set_property PACKAGE_PIN V31 [get_ports diff_clock_rtl_clk_p]
set_property PACKAGE_PIN V32 [get_ports diff_clock_rtl_clk_n]
```

This clock defaults to 156.25 MHz on board power-up and must be set to 161.132812 using the ZCU111 System Controller[4] prior to application start. A software-controlled (e.g., from a Vitis application) initiation of the SI570 was not accomplished in this demonstration but is a target of future research.

3.2 System Clock Source Domain

This clock domain sources the clock wizard for the CMAC initiation clock (`init_clk`) and the dynamic reconfiguration port clock (`drp_clock`). The clock frequency is 300.0 MHz. It arrives via an external port called `default_sysclk1_300MHz` in Fig. 6. This external port must be a differential clock and must connect to a clock wizard with a differential clock input, such as `CLK_IN1_D` in the Clocking Wizard (Fig. 6). The "run block automation" feature of Vivado IP Integrator will create the correct external port and make the connection. However, this clock must be sourced from an external suitable clock (e.g., the Silicon Labs USER_SI570 on the ZCU111. USER_SI570 must be annotated in a Xilinx Constraints File (`.xdc`), connecting package pins J19 to <clock name>`clk_p`, and J18 to <clock name>`clk_n`, respectively,[1] as shown in the following annotation:

```
create_clock          -period          3.333333333          [get_ports
default_sysclk1_300mhz_clk_p]
set_property PACKAGE_PIN J19 [get_ports default_sysclk1_300mhz_clk_p]
set_property PACKAGE_PIN J18 [get_ports default_sysclk1_300mhz_clk_n]
```

This clock defaults to 300.0 MHz on board power-up; therefore, no external adjustment is required. (It is not clear from Xilinx's 100G Ethernet Subsystem Product Guide[10] that 300.0 MHz is the required frequency for sourcing this net; however, other frequencies were not attempted.)

3.3 Initiation Clock Domain

This clock domain is sourced by the output of the Clocking Wizard (Fig. 6) as a single-ended clock and connects to the CMAC initiation clock (`init_clk`) and the dynamic reconfiguration port clock (`drp_clock`). The clock frequency is 100.0 MHz. Note that the CMAC system reset (`sys_reset`) must be held high until the initiation clock is stable. This is performed by the locked strobe of the Clocking Wizard and logic as shown in Fig. 6. The "run block automation" feature of Vivado IP Integrator will make the correct connections. However, the run block automation feature does not automatically connect this clock to `drp_clk`. While dynamic reconfiguration was not used in this demonstration, failure to connect the `drp_clk` will cause the block validation check to fail. Manually connecting `drp_clk` to the same 100-MHz clock as `init_clk` produced successful results.

3.4 Receiver User Clock Domain

This clock domain is sourced by the output of `gt_rxclk2` at 322.265625 MHz. This frequency is internally generated by the CMAC IP and is based on `gt_ref_clk`. The run block automation feature does not automatically connect this output to `rx_clk`; this must be performed manually. This clock will additionally source any AXI streaming components directly connecting to the CMAC IP receive port (`axis_rx` in Fig. 3).

3.5 Transmitter User Clock Domain

This clock domain is sourced by the output of `gt_txclk2` at 322.265625 MHz. This frequency is internally generated by the CMAC IP and is based on `gt_ref_clk`. No loop-back connections to CMAC IP are required. This clock will additionally source any AXI streaming components directly connecting to the CMAC IP transmit port (`axis_tx` in Fig. 3).

3.6 Gigabit Transceiver Reference Clock (Out) Domain

This clock domain is sourced by the output of `gt_ref_clk_out` at 161.132812 MHz. This frequency is internally generated by the CMAC IP and is based on `gt_ref_clk`. No loop-back connections to CMAC IP are required. This clock domain is not used in the demonstration but is used as a diagnostic, as described subsequently.

3.7 AXI Control and Status Clock Domain

This clock domain is sourced by the processor system (Zynq UltraScale+ MPSOC) at the default `pl_clk0`, at the default frequency of 100.0 MHz (small variations are possible). No loop-back connections to CMAC IP are required. Vivado "run connection automation" will properly connect this clock and its associated reset.

4. Verification of Correct CMAC Initiation

As verifying and troubleshooting initial digital designs can be frustrating, some deterministic method of verification is helpful. Many different clocks source the CMAC IP; the initiation procedure is also paramount. One way to verify proper clock function is to use the Vivado Integrated Logic Analyzer (ILA).[13] This is a powerful utility where the designer installs an ILA IP using Vivado IP Integrator. The ILA IP is basically a virtual oscilloscope placed into the design. The ILA IP must be sourced with proper triggers that (ideally) trigger at only finite instances. For example, a clock signal cannot be used as a trigger because it would create an event on every clock cycle and rapidly fill any available output buffers (and all computer memory). Likely, an AXI command signal such as an `axis_tvalid` could be used to command an ILA event and start a capture, which can be viewed in the Vivado ILA application. This method requires finesse and experience on the part of the designer and can take time to establish correctly.

Another possible method is to use AXI Timers.[14] AXI Timers can be conveniently inserted into the block design. Each AXI Timer is connected to a target clock domain and its associated reset. When an associated Vitis application is run, the clocks can be properly initialized in C code, and a constant "for" loop (e.g., from 1 to 1,000,000) can be run. Each AXI Timer instance can then be queried to see how far it counted during the "for" loop, which was run at PS speed (which, though not specified, is likely to be significantly different than any clock in the PL). The relative ratios of the results should ideally match the ratios of the target clocks (e.g., 322:322:161:100). Figure 7 shows the connection of target clocks to AXI Timers.

Upon completion of this verification step, the timers can be removed or retained in future upgrades. We chose to retain the AXI Timer verification step as a useful diagnostic to run at the beginning of all future test events.

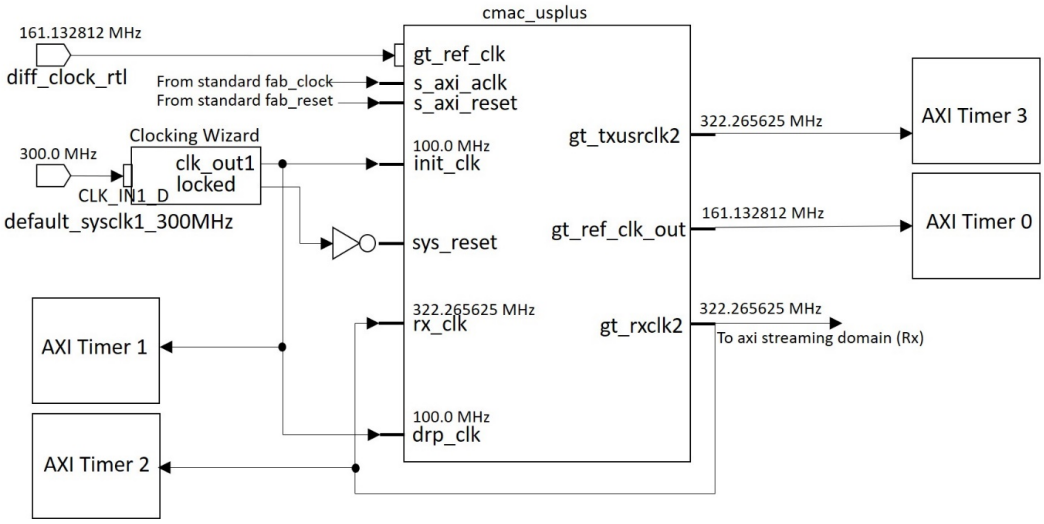

Fig. 7 Connection of CMAC to AXI Timers

5. Software Initialization of CMAC

As stated, our designs are implemented in a Baremetal Vitis Applications Project running on a single ARM Cortex A53 processor. Applications are implemented in a thin HAL with a CLI accessed through a serial connection to a host computer at 115,200 baud using the MobaXterm open-source terminal application.

The software initialization sequence for the CMAC IP can be inferred from Xilinx's 100G Ethernet Subsystem Product Guide[10] but is not explicitly stated, nor is C code provided. The following basic code sequences (Fig. 8) were used to initialize and start CMAC. Note that `Xil_In32` and `Xil_Out32` are nominally defined in `xil_io.h`, which is defined in the default board support package (BSP) generated in the associated Vitis Platform Project; CMAC_BASEADDR is the starting memory-mapped address of the CMAC IP defined in `xparameters.h` (from the BSP), and all other constants (e.g., memory addresses, registers, and bit masks) are defined in the Product Guide.[10] Importantly, this code segment must be executed before attempting to run the AXI Timer verification step described previously; otherwise, execution will deadlock because clocks will not have been properly initialized.

Note that flow control is not instantiated in this demonstration. If flow control is used, additional CMAC initialization code must be added and is inferred in the Product Guide.[10]

```
int IsRxAligned(){
            if ((Xil_In32(CMAC_BASEADDR + STAT_RX_STATUS_REG) & STAT_RX_ALIGNED) == 0)
                        return 0;
            else
                        return 1;
}

void cmac_init(){
            Xil_Out32(CMAC_BASEADDR + CONFIGURATION_RX_REG1, 0x00000001);
            Xil_Out32(CMAC_BASEADDR + CONFIGURATION_TX_REG1, 0x00000010);
            while (IsRxAligned()==0) {
                        xil_printf("CMAC not ready \n\r");
            }
            Xil_Out32(CMAC_BASEADDR + CONFIGURATION_TX_REG1, 0x00000001);
}


int main(void){

      cmac_init();

      /* Initialize and test AXI Timers to verify clock domains */

      /* Rest of Code */

      …

      return 0;

}
```

Fig. 8 Basic code sequences to initialize CMAC in a software construct

6. Radio Frequency Digital Converter

The Xilinx CMAC Ultrascale+ IP module is one of the key IPs in this design; the other is the Xilinx RFDC. The RFDC on the Xilinx Gen 1 RFSoC (ZCU111) contains four dual ADCs and two quad DACs. The maximum sample rate of the ADCs is 4.096 Gbps, and the maximum sample rate of the DACs is 6.554 Gbps. A complete discussion of the instantiation and use of the RFDC is beyond the scope of this report; the RFDC is described in the *Zynq UltraScale+ RFSoC RF Data Converter v2.3 LogiCORE IP Product Guide*.[15]

7. 100GigE Loop-back and Latency Demonstration

The first loop-back demonstration is shown in Fig. 9 and subsequently described. The concept is to generate a waveform onboard the RFSoC and send it down two paths: one path goes directly to the DAC, where it is immediately transmitted, and the second path must transit the 100GigE circuit, including transmit, external loop-back connectors, and receive sides, and is then transmitted to a second DAC. The ideal result is to observe the same output signal at both DACs; however, the second transmitted signal should lag behind the first, where latency is caused by the 100GigE transmission chain. Major elements of the demonstration, including observations, are described in the following section.

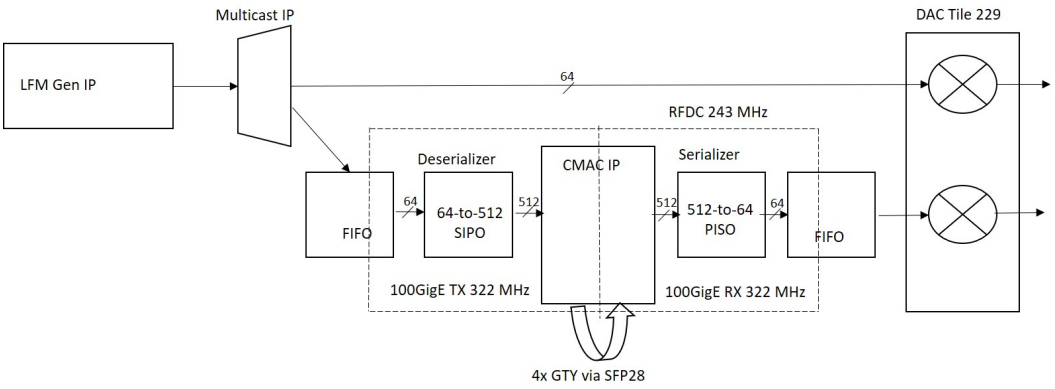

Fig. 9 100GigE loop-back and latency demonstration

7.1 Linear Frequency Modulation Generator (LFMGEN)

The Linear Frequency Modulation Generator (LFMGEN) IP is a custom IP, developed at the US Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory (ARL), designed to provide parameterized LFM pulses at variable frequency excursion, pulse duration, and pulse repetition intervals. In this demo, the LFMGEN is used as a convenient waveform generator. In this instance, the LFMGEN output consists of 16-bit signed I&Q data with two samples per clock cycle, and interleaved in accordance with Xilinx's LogiCORE IP Product Guide[15] as Q1:I1:Q0:I0. We split the output into two paths using the Multicast IP, which is another custom IP developed at DEVCOM ARL (however, default Vivado splitter IPs should also suffice). The LFMGEN operates natively on the RFDC transmission clock domain at 243 MHz. Note that RFDC transmit and receive clock domains are entirely distinct from CMAC clock domains described previously; it is the designer's responsibility to ensure proper clock boundary crossing. One path of the split goes immediately to its corresponding DAC, DAC 04 on RFDC Tile 229. The other path goes to the 100GigE transmission chain.

The choice of the LFMGEN IP as a waveform generation source is arbitrary; any waveform generator or direct digital synthesizer can be used to generate applicable RF for observation.

7.2 100GigE Transmission Chain

One split path of the Multicast IP output goes into the 100GigE transmission chain. The AXI streaming bus width at this point is 64 bits and on the RFDC clock domain at 243 MHz. This must be upconverted to 512 bits, which is the specific AXI streaming bus width required for the CMAC IP,[10] and the clock frequency must be upconverted to 322 MHz on the CMAC Transmitter User Clock Domain.

Xilinx first-in, first-out (FIFO) generator is used for clock domain crossing.[16] FIFO IPs are constructed using Block Random Access Memory (Block RAM), are 64 bits wide, and are instantiated with a depth of 16,384 words; this high word count might be excessively consuming FPGA resources. Future designers are encouraged to experiment with optimally sized FIFOs. However, it is important not to overflow the FIFO, as there is significant latency introduced by the 100GigE transmission on the order of hundreds of clock cycles.

The 100GigE transmission path also requires up-conversion from 64 to 512 bits, or a change in aspect ratio. Xilinx FIFO Generator supports asymmetric aspect ratio FIFOs only using native interfaces, not AXI Stream.[16] Therefore, custom buffers are designed and tested for this application. The custom deserializer, or single-in, parallel-out (SIPO) buffers arrival of eight, 64-bit words and outputs a single 512-bit word on the same clock cycle as the last arriving word. An interesting observation is that the order of the FIFO and SIPO cannot be reversed; if we try to first up-convert from 64 to 512 bits, then use a 512-bit FIFO for clock domain crossing from 243 to 322 MHz, Xilinx Place and Route (P&R) tools cannot meet timing requirements. Further research on the limits of FIFO sizes is recommended.

The CMAC IP abstracts away from any further implementation details of the 100GigE transmission. From the block designer's point of view, 512 bits go into CMAC `axis_tx` (Fig. 3), go through the GTY transceivers and through the SFP28 loop-back connectors, and come back into the block via `axis_rx` (Fig. 3). The provenance of data on the receive side is similar to that described previously—namely, a down-conversion from 512 to 64 bits using a serializer, or custom parallel-in, serial-out (PISO) module, then clock crossing from 322 to 243 MHz using a FIFO. Finally, the 64-bit stream is transmitted from DAC 05 on RFDC Tile 229. One observation is that data flow from RFDC to aspect converters, FIFOs, and clock domain crossings is almost certainly discontinuous; it is incumbent on the designer to ensure that discrepancies in local throughputs do not cause overflows or starvations.

The complete Vivado IP Integrator block design is shown in Appendix A, though at a greatly reduced scale.

7.3 Observations

The RFDC Tile 229 is driven using its internal phase locked loop (PLL) at 3.89975 Gsps. The internal PLL is sourced using the ZCU111's Texas Instruments TI LMX2594 PLL, which is initialized using a software script as described in Refs 10 and 17. Internal I&Q data is mixed at an output carrier frequency of 3.2 GHz in the 2nd Nyquist Zone. The LFM pulse frequency excursion itself is 2.0 MHz. DACs

04 and 05 on Tile 229 are directly connected to the High-Frequency Balance-Unbalance (baluns) in the XM500 RFDC balun board and therefore suffer minimal attenuation at target output frequency. The output is observed on the Keysight DSOX6004A Digital Storage Oscilloscope. The LFM generator itself sources an external trigger that fires at the beginning of each pulse and is used to synchronize the oscilloscope output.

The resulting output is shown in Fig. 10. On the top left (in green) is the output of DAC 04, and on the bottom left (blue) is the output of DAC 05, which went through the 100GigE transmission chain. At expanded resolution on the right, we see that the DAC 05 pulse train lags the DAC 04 pulse train by 370 ns, which is equivalent to additional latency caused by the 100GigE transmission chain. Some of the latency is caused by our own conversions, including FIFOs (one or two clock cycles) and the SIPO and PISO (eight clock cycles each). Assuming 18 clock cycles at 322 MHz (3.106-ns period), about 55 ns are added to the actual 100GigE transmission; the CMAC *plus* any physical cabling latency consumes approximately 315 ns (where physical cabling latency, of course, can be large).

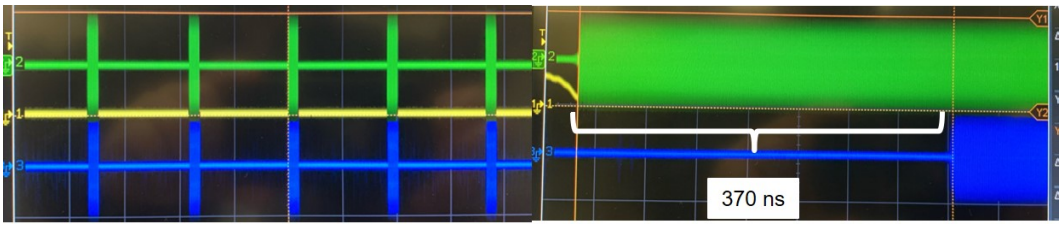

Fig. 10 Oscilloscope output of 100GigE loop-back demonstration and latency test

8. Single-Channel ADC-to-Ethernet-to-DAC Loop-back Demonstration

The next loop-back demonstration is shown in Fig. 11 and subsequently described.

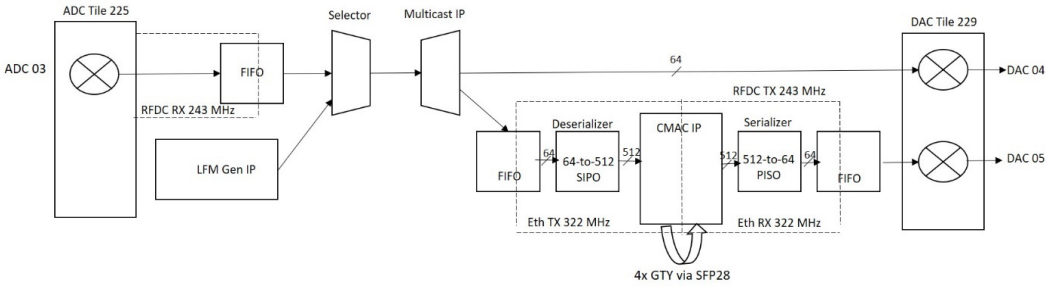

Fig. 11 Single-channel ADC-to-Ethernet-to-DAC loop-back demonstration

8.1 ADC to DAC Channel

The design shown in Fig. 11 is altered to include a receive element across a single ADC (Tile 225 ADC 03). The sampling frequency is 3.89975 Gsps as described previously with the same TI LMX2594 PLL sourcing. Note, however, that the RFDC receiver (Rx) clock domain at 243 MHz is not guaranteed to be phase-aligned with the RFDC transmitter (Tx) clock domain, even at the same frequency because they are sourced from different internal PLLs with no automatic cross-PLL lock. Therefore, we treat them as separate clock domains, and the ADC has its own clock domain crossing FIFO.

The I&Q data sequencing for dual ADCs is different from quad-DACs (and quad ADCs) in that I and Q samples arrive on separate 32-bit buses and must be manually interleaved into the Q1:I1:Q0:I0 format (not shown). An AXI-Lite controlled selector IP allows the user to select from software which device will be directed to output, either the signal arriving on the ADC, or the LFM waveform. Aside from these differences, the 100GigE transmission path is essentially as just described.

8.2 Observations

The resulting output to the oscilloscope is shown in Fig. 12. The externally generated reference square wave is shown at the center. The direct transmission path from ADC 03 to DAC 04 is shown at the top, and the transmission from ADC 03 to DAC 05 via the 100GigE transmission chain is shown at the bottom. The results verify the functionality of the ADC-Ethernet-DAC loop-back. However, the ADC 03 to DAC 05 transmission (bottom) is noisy; the noise was possibly due to RF cabling or distortions from the use of the HF baluns on the XM500 and was not observed in future builds.

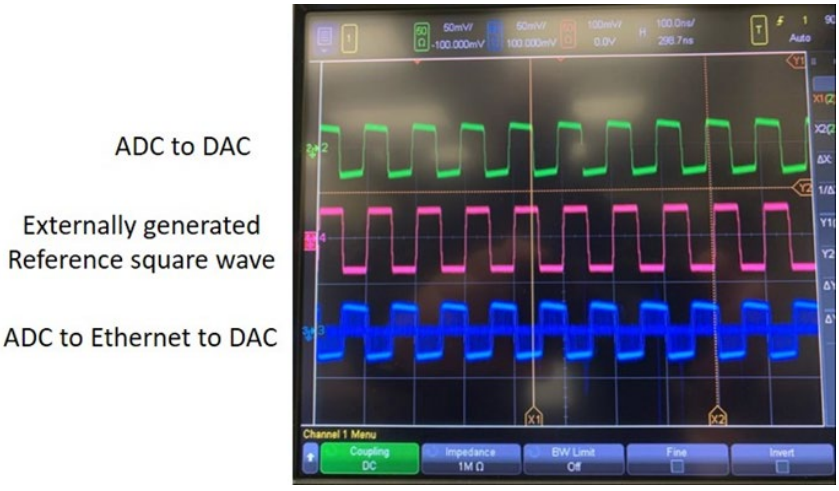

Fig. 12 Output of single-channel ADC-to-Ethernet-to-DAC loop-back demonstration

9. Two-Channel ADC to Ethernet to DAC Loop-back Demonstration

The final loop-back demonstration is shown in Fig. 13 and subsequently described.

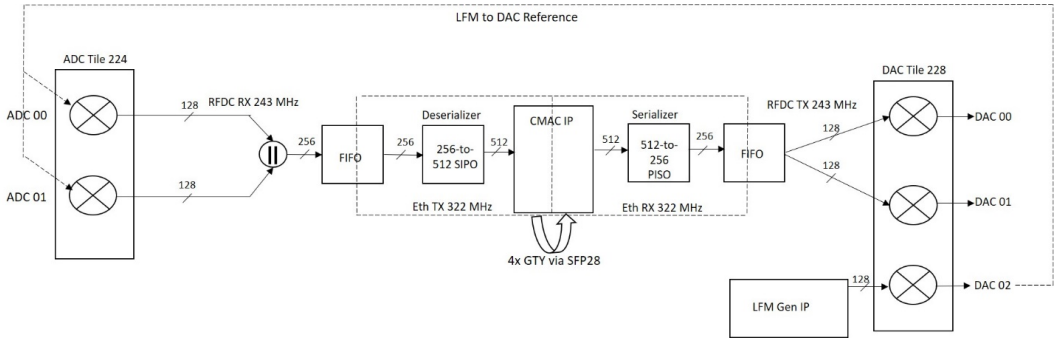

Fig. 13 Two-channel ADC-to-Ethernet-to-DAC loop-back demonstration

9.1 Dual ADC to DAC Channels

The main upgrade in this demonstration is the introduction of two channels (i.e., two ADC inputs mapped to two DAC outputs). Additionally, the digital bandwidth of each channel is upgraded to four samples at 128 bits using the interleaving scheme Q3:I3:Q2:I2:Q1:I1:Q0:I0. Both channels are concatenated prior to entering the 256-to-512 bit conversion (deserializer) and are split to two 128-bit channels after the 512-to-256 bit conversion (serializer). This has the net effect of reducing 12 clock cycles of latency in the 100GigE transmission chain, since the new aspect ratio is 2:1.

Another change was to realign ADCs and DACs to differential (non-balun) inputs and outputs in the XM500 balun board. This was done to reduce any distortion or attenuation effects of the high- or low-band baluns since baseband testing is desired. Inputs of channels 1 and 2 are aligned to ADC 00 and ADC 01 on Tile 224, respectively, and outputs of channels 1 and 2 are aligned to DAC 00 and 01 on Tile 228, respectively.

Finally, the LFM generator is no longer part of the transmission chain; it is routed to its own independent output on DAC 02 Tile 228. The LFM output can optionally be routed back to inputs ADC 00 and/or 01, which was performed in this demo.

The complete Vivado IP Integrator block design is shown in Appendix A. The Vivado post P&R power report is available in Appendix B.

9.2 Observations

The LFM generator was used to generate LFM pulses with 2.0-MHz frequency excursion. The ADCs and DACs were software-adjusted to a mixing frequency of 0 (i.e., the LFM output is at baseband).

In Fig. 14, the LFM waveform generated from DAC 02 is displayed at the top. The RF LFM output is then split and routed to both channels (Ch 1 and Ch 2). The RF then correctly propagates through both ADC-to-Ethernet-to-DAC channels and is displayed at the center and bottom of Fig. 14. The RF output is generated using an RF-splitter from the LFM waveform from DAC 02 to ADC Ch 1 and Ch 2, per Fig. 15. There is an obvious delay between LFM to DAC output and the two channels; the latency in this demo is indeterminate, as RF cabling was not calibrated to determine an accurate latency. This will be an area for future research.

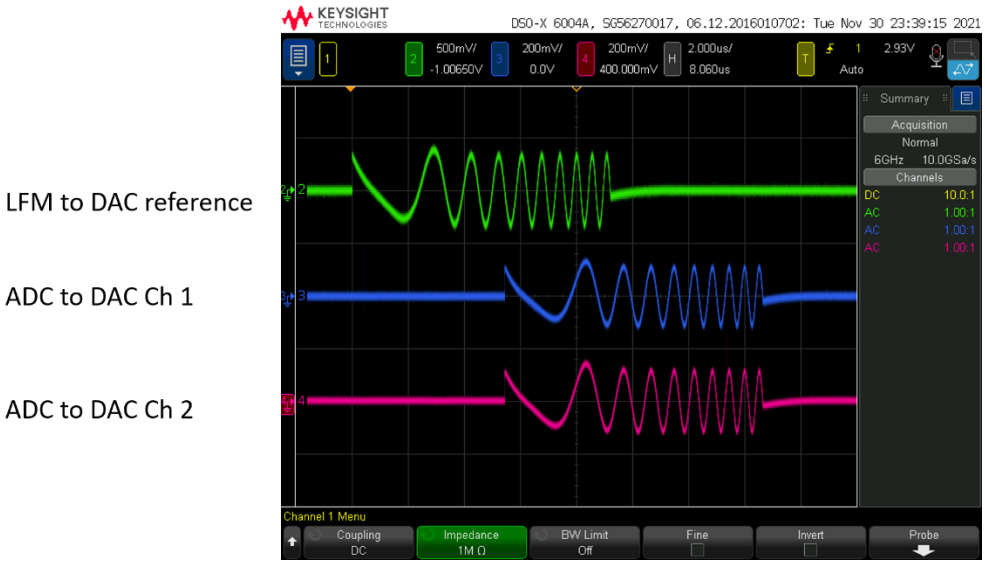

Fig. 14 Output of two-channel ADC-to-Ethernet-to-DAC loop-back demonstration

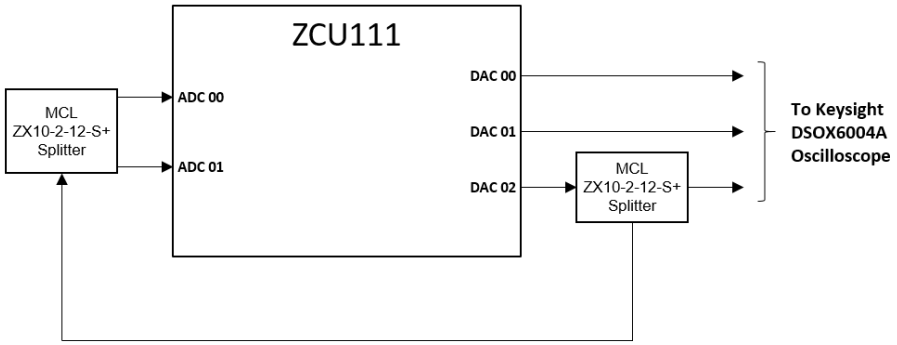

Fig. 15 RF split routing for ZCU111 Ethernet loop-back testing

10. Conclusions and Future Work

In this research we investigated the procedures for building, initializing, verifying, and observing the performance of 100GigE high-speed communication links using the Xilinx Generation 1 RFSoC. We reviewed the physical and datalink layer requirements for the CAUI-4 and Xilinx CMAC UltraScale+ IP Ethernet implementations, respectively. After a comprehensive review of CMAC clocking domains, we described three demonstrations of 100GigE loop-back capability, including a board-synthesized waveform-to-Ethernet-to-DAC demonstration, and single- and dual-channel ADC-to-Ethernet-to-DAC demonstrations. We observed correct functionality of 100GigE in all three configurations and that the 100GigE layer itself introduces about 315–370 ns of data latency. Remaining topics to be investigated in future work include verifying ADC-to-Ethernet-to-DAC latency using precisely calibrated external RF cabling, characterizing errors (including BER) and related behaviors, initializing software of the Si570 clocks, and investigating the ability to close timing on wide FIFOs operating at 100GigE clock frequency.

11. References

1. Xilinx. ZCU111 evaluation board user guide UG1271 (v1.2); 2018 Oct 2.
2. Xilinx. UltraScale architecture GTH transceivers user guide UG576 (v1.7.1); 2021 Aug 18.
3. SFF Committee. INF-8074i specification for SFP (small form factor pluggable) transceiver rev. 1.0; 2001 May 12.
4. Xilinx. ZCU111 software install and board setup; 2019 May.
5. Xilinx. Aurora 64B/66B protocol specification SP011 (v1.3); 2014 Oct 1.
6. Interlaken protocol definition, a joint specification of Cortina systems and Cisco systems, revision 1.2; 2008 Oct 7.
7. IEEE 802.3-2012. IEEE standard for Ethernet. IEEE Standards Association; 2012 Aug 30 [accessed 2021 Dec 14]. https://standards.ieee.org/standard/802_3-2012.html
8. IEEE 802.3-2015. IEEE Standard for Ethernet. IEEE Standards Association; 2015 Sep 3 [accessed 2021 Dec 14]. https://standards.ieee.org/standard/802_3-2015.html
9. Draft amendment to IEEE Std 802.3-2012 IEEE draft P802.3bm/D3.1, IEEE P802.3bm 40 Gb/s and 100 Gb/s Fiber Optic Task Force; 2014 Jul 22 [accessed 2021 Dec 14]. https://www.ieee802.org/3/bm/public/cuadhoc/meetings/aug1_14/8023-anx83ECMP.pdf
10. Xilinx. UltraScale+™ devices integrated 100G Ethernet subsystem v3.1 product guide, Vivado design suite, PG203; 2020 Jun 24.
11. Xilinx. AXI reference guide UG761 (v13.1); 2011 Mar 7.
12. multiLane, Inc. SFP28 passive loopback module ML4026-28; n.d. [accessed 2021 Dec 14]. https://multilaneinc.com/wp-content/uploads/2020/03/ML4026-28-MDS-1.pdf
13. Xilinx. Integrated logic analyzer v6.2 LogiCORE IP product guide Vivado design suite PG172; 2016 Oct 5.
14. Xilinx. AXI timer v2.0 LogiCORE IP product guide Vivado design suite PG079; 2016 Oct 5.
15. Xilinx. Zynq UltraScale+ RFSoC RF data converter v2.3 LogiCORE IP product guide, Vivado design suite, PG269 (v2.3); 2020 Jun 3.

16. Xilinx. FIFO generator v13.1 LogiCORE IP product guide, Vivado design suite, PG057; 2017 Apr 5.
17. Texas Instruments. LMX2594 15 GHz wideband PLLatinum™ RF synthesizer with phase synchronization and JESD204B support, SNAS696; 2017 Mar.

Appendix A. Block Designs

Block designs for the 100GigE loop-back demonstration and two-channel ADC-to-Ethernet-to-DAC loop-back demonstration are shown in Figs. A-1 and A-2, respectively. While it is not possible to glean much detail at this level of resolution, the relative component separation, complexity, and flow of these designs can be observed.

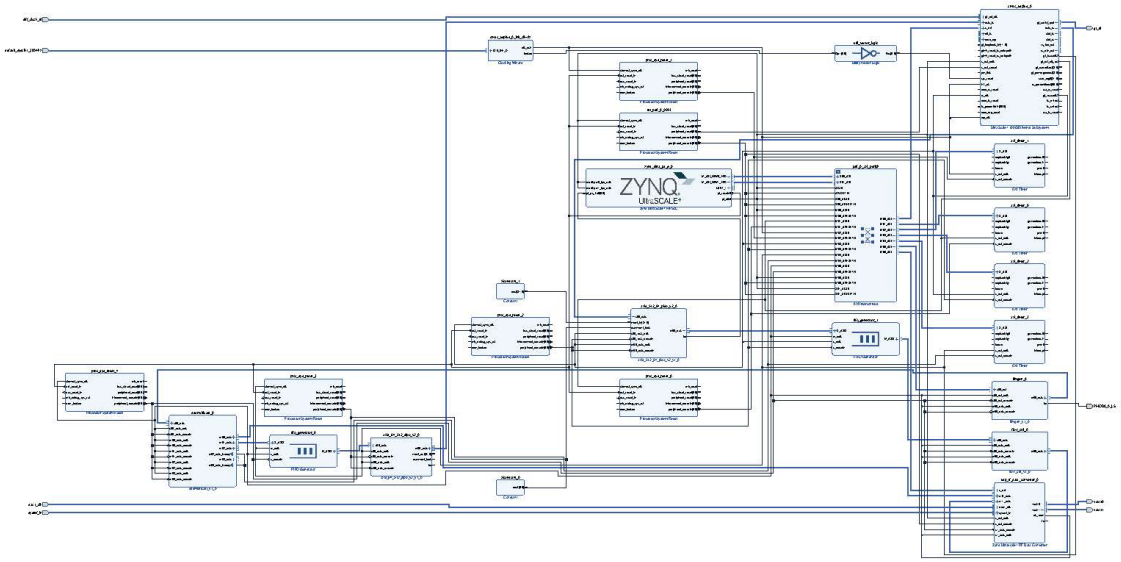

Fig. A-1 100GigE loop-back demonstration block design

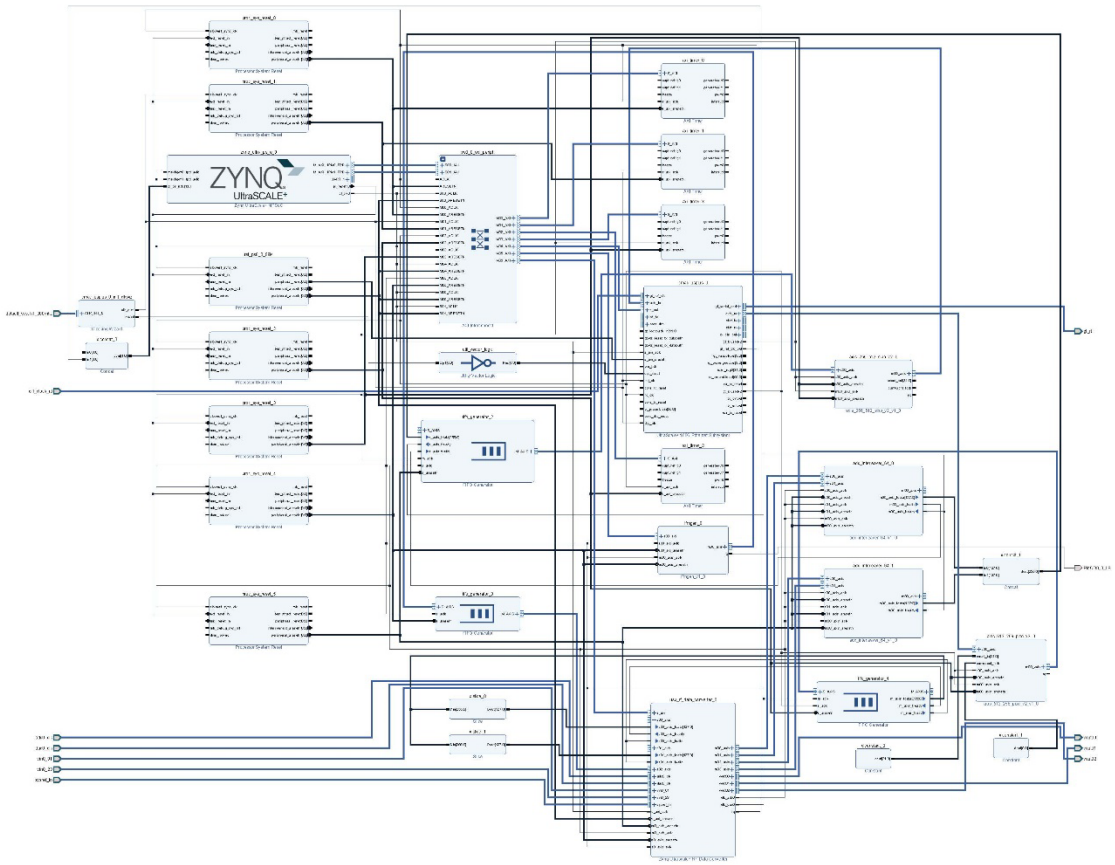

Fig. A-2 Two-channel ADC-to-Ethernet-to-DAC loop-back demonstration

Appendix B. Vivado Power Report (abridged)

This appendix contains the abridged Vivado Power Report for the two-channel ADC-to-Ethernet-to-DAC demonstration.

```
Copyright 1986-2020 Xilinx, Inc. All Rights Reserved.
----------------------------------------------------------------------------------------
| Tool Version     : Vivado v.2020.1 (win64) Build 2902540 Wed May 27 19:54:49 MDT 2020
| Date             : Fri Oct 29 16:26:49 2021
| Host             : DESKTOP-G8U1IK1 running 64-bit major release  (build 9200)
| Command          : report_power -file design_1_wrapper_power_routed.rpt -pb
design_1_wrapper_power_summary_routed.pb -rpx design_1_wrapper_power_routed.rpx
| Design           : design_1_wrapper
| Device           : xczu28dr-ffvg1517-2-e
| Design State     : routed
| Grade            : extended
| Process          : typical
| Characterization : Production
----------------------------------------------------------------------------------------

Power Report

Table of Contents
-----------------
1. Summary
1.1 On-Chip Components
1.2 Power Supply Summary
1.3 Confidence Level
2. Settings
2.1 Environment
2.2 Clock Constraints
3. Detailed Reports
3.1 By Hierarchy

1. Summary
----------

+--------------------------+--------------+
| Total On-Chip Power (W)  | 10.042       |
| Design Power Budget (W)  | Unspecified* |
| Power Budget Margin (W)  | NA           |
| Dynamic (W)              | 8.695        |
| Device Static (W)        | 1.347        |
| Effective TJA (C/W)      | 0.8          |
| Max Ambient (C)          | 91.5         |
| Junction Temperature (C) | 33.5         |
| Confidence Level         | Medium       |
| Setting File             | ---          |
| Simulation Activity File | ---          |
| Design Nets Matched      | NA           |
+--------------------------+--------------+
* Specify Design Power Budget using, set_operating_conditions -design_power_budget <value in Watts>


1.1 On-Chip Components
----------------------

+--------------------------+-----------+----------+-----------+-----------------+
| On-Chip                  | Power (W) | Used     | Available | Utilization (%) |
+--------------------------+-----------+----------+-----------+-----------------+
| Clocks                   |     0.371 |       42 |       --- |             --- |
| CLB Logic                |     0.360 |   100290 |       --- |             --- |
|   LUT as Logic           |     0.299 |    38338 |    425280 |            9.01 |
|   LUT as Distributed RAM |     0.045 |      992 |    213600 |            0.46 |
|   Register               |     0.010 |    41017 |    850560 |            4.82 |
|   CARRY8                 |     0.004 |     1266 |     53160 |            2.38 |
|   LUT as Shift Register  |    <0.001 |      101 |    213600 |            0.05 |
|   Others                 |     0.000 |     2192 |       --- |             --- |
|   F7/F8 Muxes            |     0.000 |    10815 |    425280 |            2.54 |
| Signals                  |     0.346 |    70850 |       --- |             --- |
| Block RAM                |     0.029 |       15 |      1080 |            1.39 |
| RFAMS                    |     2.837 |        6 |       --- |             --- |
|   RFADC                  |     1.485 |        4 |         4 |          100.00 |
|   RFDAC                  |     1.352 |        2 |         4 |           50.00 |
| MMCM                     |     0.098 |        0 |       --- |             --- |
| I/O                      |     0.009 |        3 |       347 |            0.86 |
| GTY                      |     2.094 |        4 |        16 |           25.00 |
| PS8                      |     2.299 |        1 |       --- |             --- |
| Hard IPs                 |     0.251 |        1 |       --- |             --- |
|   CMAC                   |     0.251 |        1 |       --- |             --- |
| Static Power             |     1.347 |          |           |                 |
|   PS Static              |     0.103 |          |           |                 |
|   PL Static              |     1.244 |          |           |                 |
| Total                    |    10.042 |          |           |                 |
+--------------------------+-----------+----------+-----------+-----------------+
```

1.2 Power Supply Summary

Source	Voltage (V)	Total (A)	Dynamic (A)	Static (A)	Powerup (A)	Budget (A)	Margin (A)
Vccint	0.850	2.319	1.945	0.374	NA	Unspecified	NA
Vccint_io	0.850	0.074	0.001	0.073	NA	Unspecified	NA
Vccbram	0.850	0.008	0.002	0.005	NA	Unspecified	NA
Vccaux	1.800	0.330	0.054	0.276	NA	Unspecified	NA
Vccaux_io	1.800	0.062	0.004	0.058	NA	Unspecified	NA
Vcco33	3.300	0.000	0.000	0.000	NA	Unspecified	NA
Vcco25	2.500	0.000	0.000	0.000	NA	Unspecified	NA
Vcco18	1.800	0.000	0.000	0.000	NA	Unspecified	NA
Vcco15	1.500	0.000	0.000	0.000	NA	Unspecified	NA
Vcco135	1.350	0.000	0.000	0.000	NA	Unspecified	NA
Vcco12	1.200	0.000	0.000	0.000	NA	Unspecified	NA
Vcco10	1.000	0.000	0.000	0.000	NA	Unspecified	NA
Vccadc	1.800	0.008	0.000	0.008	NA	Unspecified	NA
VCC_PSINTFP	0.850	0.633	0.595	0.039	NA	Unspecified	NA
VCC_PSINTLP	0.850	0.250	0.242	0.008	NA	Unspecified	NA
VPS_MGTRAVCC	0.850	0.139	0.138	0.001	NA	Unspecified	NA
VCC_PSINTFP_DDR	0.850	0.731	0.726	0.005	NA	Unspecified	NA
VCC_PSPLL	1.200	0.071	0.069	0.002	NA	Unspecified	NA
VPS_MGTRAVTT	1.800	0.034	0.033	0.001	NA	Unspecified	NA
VCCO_PSDDR_504	1.200	0.626	0.592	0.034	NA	Unspecified	NA
VCC_PSAUX	1.800	0.002	0.000	0.002	NA	Unspecified	NA
VCC_PSBATT	1.200	0.000	0.000	0.000	NA	Unspecified	NA
VCC_PSDDR_PLL	1.800	0.001	0.000	0.001	NA	Unspecified	NA
VCCO_PSIO0_500	1.800	0.001	0.000	0.001	NA	Unspecified	NA
VCCO_PSIO1_501	1.800	0.001	0.000	0.001	NA	Unspecified	NA
VCCO_PSIO2_502	1.800	0.001	0.000	0.001	NA	Unspecified	NA
VCCO_PSIO3_503	3.300	0.001	0.000	0.001	NA	Unspecified	NA
VCC_PSADC	1.800	0.002	0.000	0.002	NA	Unspecified	NA
VCCINT_AMS	0.850	1.591	1.580	0.011	NA	Unspecified	NA
DACAVCC	0.925	0.302	0.294	0.009	NA	Unspecified	NA
DACAVCCAUX	1.800	0.067	0.067	0.000	NA	Unspecified	NA
DACAVTT	2.500	0.080	0.075	0.005	NA	Unspecified	NA
ADCAVCC	0.925	0.390	0.376	0.014	NA	Unspecified	NA
ADCAVCCAUX	1.800	0.363	0.315	0.048	NA	Unspecified	NA
VCCSDFEC	0.850	0.036	0.000	0.036	NA	Unspecified	NA
MGTYAVcc	0.900	0.493	0.438	0.055	NA	Unspecified	NA
MGTYAVtt	1.200	1.115	1.091	0.024	NA	Unspecified	NA
MGTYVccaux	1.800	0.054	0.051	0.003	NA	Unspecified	NA

1.3 Confidence Level

User Input Data	Confidence	Details	Action
Design implementation state	High	Design is routed	
Clock nodes activity	High	User specified more than 95% of clocks	
I/O nodes activity	High	User specified more than 95% of inputs	
Internal nodes activity	Medium	User specified less than 25% of internal nodes	Provide missing internal nodes activity with simulation results or by editing the "By Resource Type" views
Device models	High	Device models are Production	
Overall confidence level	Medium		

2. Settings

2.1 Environment

Ambient Temp (C)	25.0
ThetaJA (C/W)	0.8
Airflow (LFM)	250
Heat Sink	medium (Medium Profile)
ThetaSA (C/W)	1.2
Board Selection	medium (10"x10")
# of Board Layers	12to15 (12 to 15 Layers)
Board Temperature (C)	25.0

3. Detailed Reports

3.1 By Hierarchy

Name	Power (W)
design_1_wrapper	8.695
design_1_i	8.695
axi_timer_0	0.004
U0	0.004
axi_timer_1	0.004
U0	0.004
axi_timer_2	0.008
U0	0.008
axi_timer_3	0.009
U0	0.009
axis_256_512_sipo_v2_0	0.009
U0	0.009
axis_512_256_piso_v2_0	0.012
U0	0.012
cmac_usplus_0	2.755
inst	2.755
cmac_usplus_0_init_clkwiz	0.101
inst	0.101
fifo_generator_2	0.043
U0	0.043
fifo_generator_3	0.009
U0	0.009
fifo_generator_4	0.021
U0	0.021
lfmgen_0	0.460
U0	0.460
proc_sys_reset_4	0.001
U0	0.001
proc_sys_reset_5	0.002
U0	0.002
ps8_0_axi_periph	0.079
m01_couplers	0.002
m02_couplers	0.003
m03_couplers	0.004
m04_couplers	0.003
m05_couplers	0.004
m06_couplers	0.003
s00_couplers	0.029
s01_couplers	0.028
xbar	0.003
usp_rf_data_converter_0	2.874
inst	2.874
zynq_ultra_ps_e_0	2.300
inst	2.300

List of Symbols, Abbreviations, and Acronyms

100GigE	100 Gigabit Ethernet
ADC	analog-to-digital converter
ARL	Army Research Laboratory
AXI	Advanced Extensible Interface
AXIS	Advanced Extensible Interface – Streaming
balun	balanced-unbalanced
BER	bit error rate
BSP	board support package
CAUI-4	chip-to-module 100 Gb/s four-lane attachment unit interface
CLI	command line interface
DAC	digital-to-analog converter
DDR4	Double Data Rate 4
DEVCOM	US Army Combat Capabilities Development Command
DRP	dynamic reconfiguration port
FIFO	first-in, first-out
FPGA	field-programmable gate array
Gbps	gigabits per second
GHz	gigahertz
Gsps	giga samples per second
HAL	hardware abstraction layer
HF	high frequency
IBW	instantaneous bandwidth
I	in-phase
IEEE	Institute of Electronics and Electrical Engineers
ILA	Integrated Logic Analyzer
IP	intellectual property
LBUS	local bus
LFM	Linear Frequency Modulation

LFMGEN	Linear Frequency Modulation Generator
MHz	megahertz
MPSOC	multi-processor system-on-chip
ns	nanoseconds
OSI	Open Systems Interconnect
P&R	Place and Route
PISO	parallel-in, serial-out
PL	programmable logic
PLL	phase locked loop
PS	processor system
Q	quadrature
RAM	random access memory
RF	radio frequency
RFDC	Radio Frequency Digital Converter
RFSoC	radio frequency system-on-chip
RS-FEC	Reed–Solomon Forward Error Correction
Rx	receive
SDR	software-defined radio
SFP	small form factor pluggable
SIPO	serial-in, parallel-out
SWaP	size, weight, and power
TI	Texas Instruments
Tx	transmit

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLD DCI TECH LIB
2 (PDF)	DEVCOM ARL FCDD RLS EW W DIEHL E VIVEIROS