Available online at https://asric.africa/engineering-sciences
ASRIC Journal on Engineering Sciences Vol.4(1)(2023) 15-21

An Architect of a Layer 2 Contact Plane for Concurrent Dual Mode of Satellite Constellation Networking

Amr Ahmed Azzam[1], Mehaseb Ahmed[2], Alemam El Said[3] & Mohmaed Medhat Mokhtar[4]

Egyptian Space Agency, amr.azzam@egsa.gov.eg
Misr International University, mehasb.ahmed@miuegypt.edu.eg
Misr International University, alemam.said@miuegypt.edu.eg High Institute of engineering and technology at Obour, medhat.mokhtar@oi.edu.eg

Abstract

The main challenge in satellite constellation networking is the process burden on the satellites as network nodes and Host terminals, where each satellite performs both forwarding traffic task, and downloading its data. The paper introduces the multi-Processor system on chip (MPSoc) FPGA envisioned technologies to perform satellite networking tasks, and presents a proposed model of a Layer 2 Contact Plane (CP) in charge of traffic management. The design of a satellite CP refers to IEEE 802.11 technology in terms of configurations and protocols. It is successfully implemented in a framework of a project aims a development of satellite constellation network subsystem using the development kit ULTRA96-v2.

***Keywords- Satellite constellation,* MPSoc, DTN, ISL, contact plane, contact plan, networking**

I. Introduction

In [1], a concept of contact plan is introduced to optimize Delay/Disruption Tolerant Networking (DTN) routing solutions. It presents the first prototype of Contact Plan Designer; a tool to generate an accurate contact plan. Regardless of the design approach, Practical implementation requires a dedicated Layer 2 entity to manage this plan. This leads to a concept of Contact Plane (CP) for On-Board management of Inter Satellite Links (ISLs), Contact Plan (Contact Table CT) for satellites subjected to intermitted communication, as well as normal Forwarding Plan (Forwarding Table FT). For the dynamic topology of satellite constellation, many topological dynamics shielding methods have been proposed and can be classified into two types: Virtual Topology (VT) [2, 3], and Virtual Node (VN) methods [4]. Routing primarily includes the routing algorithm based on the virtual topology, and routing strategy based on the virtual node.

In this paper, the CP is presented to support the principle of the VT method, where the constellation period is divided n time slices (snapshots). In the time interval of each snapshot, the dynamic topology of the satellite network can be modeled as a static virtual topology. Therefore, the computation of satellite network routing is transformed into the routing optimization of n static virtual topologies. This optimization process is developed in two identified states of ISLs. The first concerns the normal ISL where the FT is used for data forwarding using P2P protocol with the adjacent satellite. The second is the case of intermitted/disrupted ISL where the CT is called and DTN protocol with the surround group of satellites is applied. This gives rise to the need of dual modes of operation; P2P and Group Owner (GO).

Obviously, this method requires a significant storage space to store routing information for n time slices, in addition to the processing burden on the satellites as network nodes and Host terminals, where each satellite performs both forwarding traffic task, and downloading its data as a Host. The satellite as a Host is in charge of numerous tasks in two main categories; Network's Management and Control Tasks (NMCTs), and

Application's Management and Control Tasks (AMCTs). The NMCTs are mainly the Logical Link Control (LLC) sublayer of the Data Link Layer (DLL) tasks, in addition to Contact Plane (CP) services for establishment ISLs with neighbors, and forwarding process according to FTs or CTs. The AMCTs includes adaption of satellite subsystems data format, configuration and update both the FTs and CTs, and assessment of ISL QoS.[

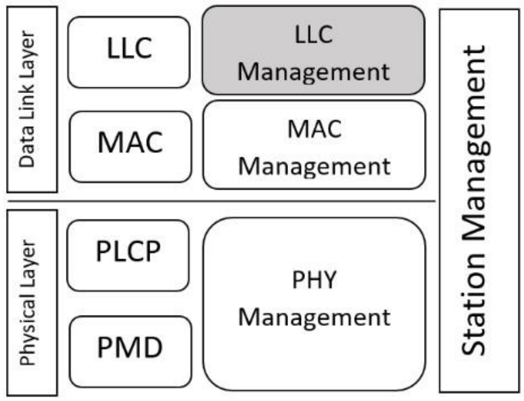

Figure 1: IEEE standard 802.11 reference model with a proposed LLME.

In this paper, the CP is designed as a shell to a proposed LLC Layer Management Entities (LLME) that has an interface to the MAC Layer Management Entities (MLME) [1]. This approach is normally proposed and applied in the OSI standards; e.g., in IEEE802.11x as shown in Fig. 1. The paper is organized as follows. In Section II, the essential requirements for concurrent dual mode of operation for satellite constellation networking are described. It is followed in Section III by a model proposal of Layer 2 CP to manage the constellation traffics. The design and implementation of the CP is presented in Section IV, and finally is the conclusion.

II. Concurrent Dual Modes of Satellite Constellation Nodes

The mode/s of operation must be appropriate to the nature of the mega constellations; typically, dynamic attitude in deterministic orbit subjected to intermittent and high failure rate of ISLs. The proposed model of a satellite constellation node refers to IEEE 802.11 technology in terms of configurations and protocols shown in Fig.1. The term 802.11 technology means topology, modes of operation, functions and protocols. As described early, each node in the network performs switching functions for forwarding traffic data according to the associated FT and CT, and download its data as a Host. It seems similar to mesh architecture based on P2PDirect, where each node can act as a L2 switch to forward the data [5,6].

The essential requirements imposed on mesh architecture based on P2P-Direct are high stability and low mobility. If a node failure occurs due to any hardware problem or any others, the neighbor node will perform rerouting with the help of routing protocols. It worth notifying that it is applied in LEO Iridium constellation with 66 satellites [10]. The Iridium is considered a low mobility network since it comprises a few numbers of satellites with long separation between each. On the other side, the mega constellation of thousands satellite with small separation between each is a high mobility one. Therefore, this mesh approach of design is not appropriate for LEO mega constellation

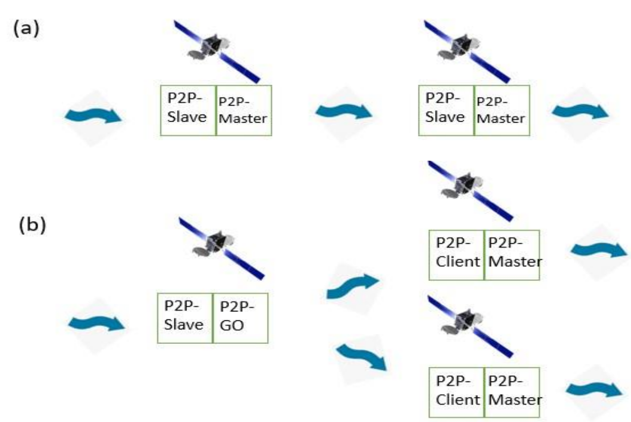

Figure 2: ISL in Two Different Modes of Operation; network.

(a) P2P Master-Slave, and (b) P2P Go-Client. More precisely, it is more visible to consider the ISLs as a P2P-Master/Slave connection with a neighbor in the FT as shown in Fig.2 (a). In an event of ISL failure, the Host invokes the associated CT, and manage a DTN protocol to establish a GO/Client connection with the recommended new neighbor/s as shown in Fig.2 (b). It is considered an appropriate approach of self-configurable and self-healable of mega constellation.

III. The contact plane; description and Architecture

As mentioned early, the CP represents a shell to the routing management functions in the LLME that has an interface to the MLME, with an Architecture shown in Fig.3 where the satellite starts the planned master peer session and search for slave for an allowed period and if the salve is found. It will immediately connect to and return success. If it cannot find a peer slave , it will start planned peer group owner session assuming itself as a group owner and starts a GO/Client connection with the recommended new neighbor/s . if the formation complete and connected , it will update the contact table and report. And returns with success. But In case of formation failure , it will report and returns with failed to connect.

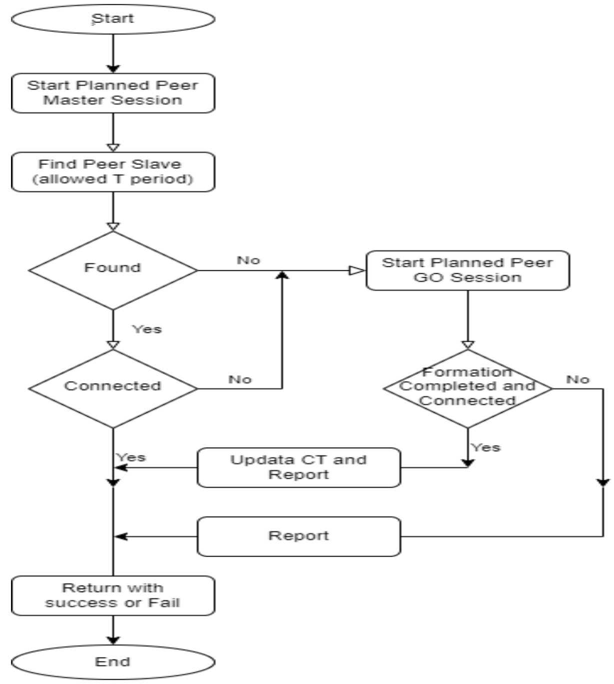

Figure 3: The Architecture of P2P-Master/Go to P2P-Slave/ Client Switching Mode of Operation and Connection Establishment

In the default Master mode of operation, the node transmits beacon frames to an assigned Slave in the FT. It shall wait for receiving Request frame from the assigned Slave for a specified interval of time. Upon arrival of the Request frames from the Client, the Invitation Response frame is transmitted. In this case, the Slave find operation is successful, and return for completing the CP operation

If it doesn't receive Request frame in the specified time interval, it returns with failed (Slave is not founded). It starts a planned peer-Master/Slave session by updating process of both FT and CT, and setting the modes of a concurrent operation. The node, as a master, starts the Find operation of the Slave in the FT through the MAC address stored for this private network. The Find operation is described by the Chart of Master Application Architecture in Fig.4. If the Slave is successfully founded and connected, the Slave MAC address is stored in the driver interface and the connection is terminated for the time being. If it fails in finding the Slave, it starts Peer-GO/Client operation, shown in Fig.5. If it successfully formed a Group, the MAC addresses of the Clients are stored.

For this particular application, the P2P-device starts autonomous group formation; i.e., it announces itself as GO without any GO negotiation phase and starts sending beacons as a conventional Wi-Fi AP. After completion, it starts the DTN protocol for forwarding task.

A. Master Application Architecture.

In the default Master mode of operation, the node transmits beacon frames to an assigned Slave in the FT. It shall wait for receiving Request frame from the assigned Slave for a specified interval of time. Upon arrival of the Request frames from the Client, the Invitation Response

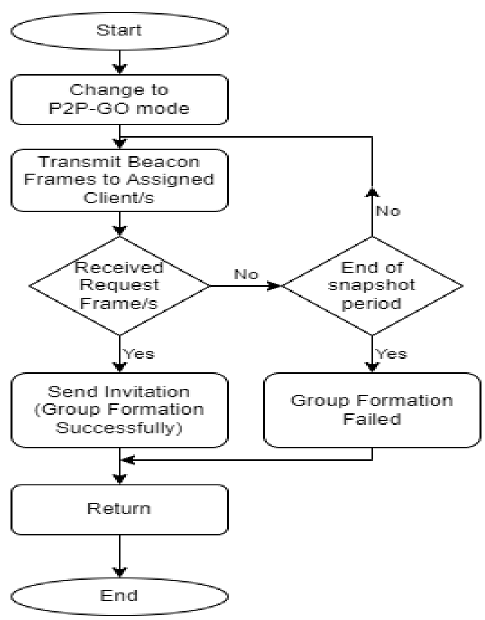

Figure 4: The Architecture of Find Slave Operation.

frame is transmitted. In this case, the Slave find operation is successful, and return for completing the CP operation. If it doesn't receive Request frame in the specified time interval, it returns with failed (Slave is not founded).

B. Go Application Architecture

The operation starts by setting of autonomous GO mode, Group ID, P2P Group BSSID, Channel List, Operating Channel, and Configuration Timeout. The group is limited to client members in the CT. So, the illustrated architecture shall not go through Discovery Procedure (Device and Services), or include a P2P Client Info Descriptor in the P2P Group Info attribute for each P2P Client that is connected to the P2P Group [6].

The start of P2P-GO session is Transmitting beacon frames. It shall wait for receiving Request frame from the group members (Clients). Upon arrival of Invitation Request frames, it transmits the
Invitation Response frame.

This phase may be subjected to an assessment to pick the clients according to the applied DTN protocol. In this case, the Group formation process is successful, return for completing the CP operation. If it fails in group formation during the time interval of a snapshot, it returns with failed group formation.

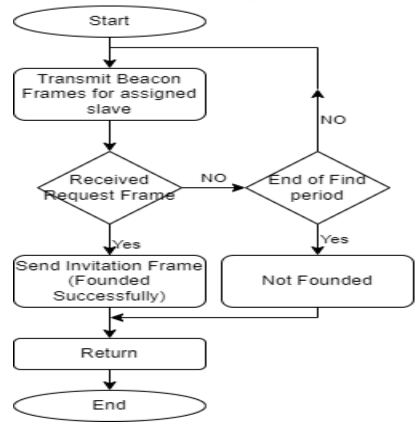

Figure 5: The Architecture of Start P2P-GO Session operation.

IV. Design and Implementation

This design is implemented on Vitis platform that combine an integrated design environment of Xilinx's Vivado for creating hardware designs [9]. It is implemented in a project framework that deploys three APUs Cortex-A53 of the Quad-Cores of FPGA Zynq UltraScale MPSoc ZU3EG SBVA484 [8]. It also deploys the Generic Interrupt Controller (GIC) of the FPGA to orchestrate the data flow between different modules running on the APUs [7]. The first Core, denoted by CPU0, will carry the configuration tasks of GIC-400 unit, Orchestrates the data follow by enabling/disabling the interrupts to/from the other cores; CPU1 and CPU2. The CPU1 and CPU2 are allocated to perform the NMCTs and AMCTs; respectively.

The Orchestrion is a coordination of interrupt events in and between NMC and AMC modules, and its tasks include a setup process (configuration and initialization) of an Interrupt Engine. One of these events is triggered from a built-in Timer to start the process depicted by the Architecture in Fig.3 at due time of a new snapshot.

A. Interrupt Engine

It is designed based on the GIC-400 of FPGA ZU3EG. It manages interrupts from two parts; the first is developed on the Programmable Logic (PL) and denoted PL Interrupt (PLI), while the second is generated from Shared Peripherals (SP) to the MIO and denoted SP Interrupt (SPI). These two types are processed by the orchestration module that control the GIC-400 operation.

The schematic diagram of the PLI is presented in Fig.6, where the first processing core of the available Quad cores, denoted by CPU0, is connected to the PL blocks using AXI interfaces. The Processing System (PS) of CPU0 drives control and setting outputs from two M_AXI output ports to NMC_AMC_Unit and Timer_unit through AXI Interconnect module, as shown in Fig.6. These two units generate a set of interrupts, which is concatenated in the xlconcat_0 unit and feeds to the PS through PL_PS_IRQ[4:0].

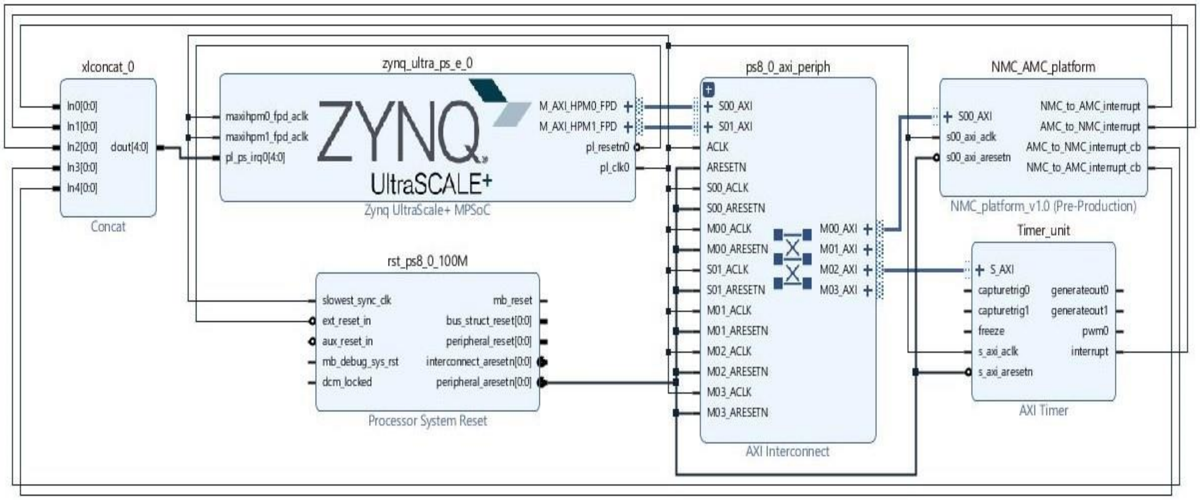

Figure 6: The Schematic Diagram of the PL Interrupt (PLI).

B. NMC_AMC_Platform

The unit is a customized AXI4 peripheral, generated by Verilog code for the AXI4-Lite slave. It is developed to set (enable/disable/pend) and clear the interrupt from the CPU1 and CPU2 using the first bit of five dedicated registers. Each register is assigned to an Interrupt Service Routine (ISR) managed by the orchestrion module. The first bit of each register is connected to an input of a five inputs port for set or clear the assigned interrupt, and connected to an output of a five outputs port for interrupting the corresponding processor.

C. Timer_unit

The Timer is selected from IP repository of Vivado, and used to switch between the timetagged FTs or CTs of different snapshots. The interrupt output of the Timer is concatenated at input in0[0:0] and trigger IRQ[4] of CPU0. The interrupt is managed by the Orchestrion module (enabled/disabled) using the GIC, and distributed to CPU1 to perform the corresponding ISR. At the end of the routine, the timer is "set to rested" to enable the trigger at the due time of the next snapshot. The SPI in the CP design is initiated from the network controller of the Ultra96 kit upon an event in the MAC layer. It is used to restart the CP Architecture in Fig.3 upon ISL disconnection.

V. Implementation Results

The results of implementation process are presented on the platform of both Vivado and Vitis. In Vivado, the implementation process is the final step after top design wrapping and synthesis of the Schematic Diagram of the PL Interrupt (PLI) in Fig.6. it is built on state-of-art partitioning, placement, and routing algorithms guided by Machine Learning-based predictors. It is followed by a generate bit stream file that contains the configuration information of the design on the PL. The successful outcomes of the both implementation and bitstream generations are shown in Figs, 7 and 8; respectively.

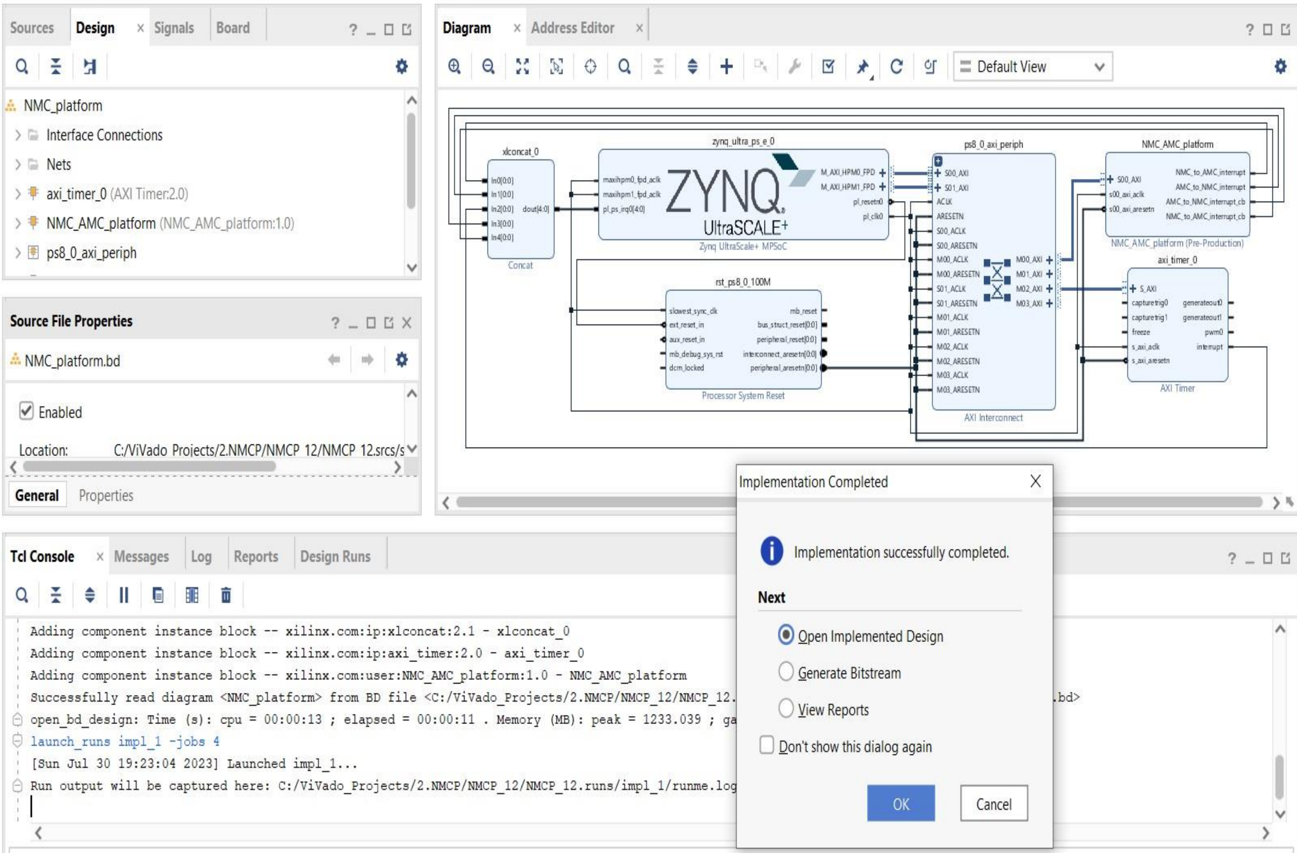

Figure 7: The outcomes of the implantation process.

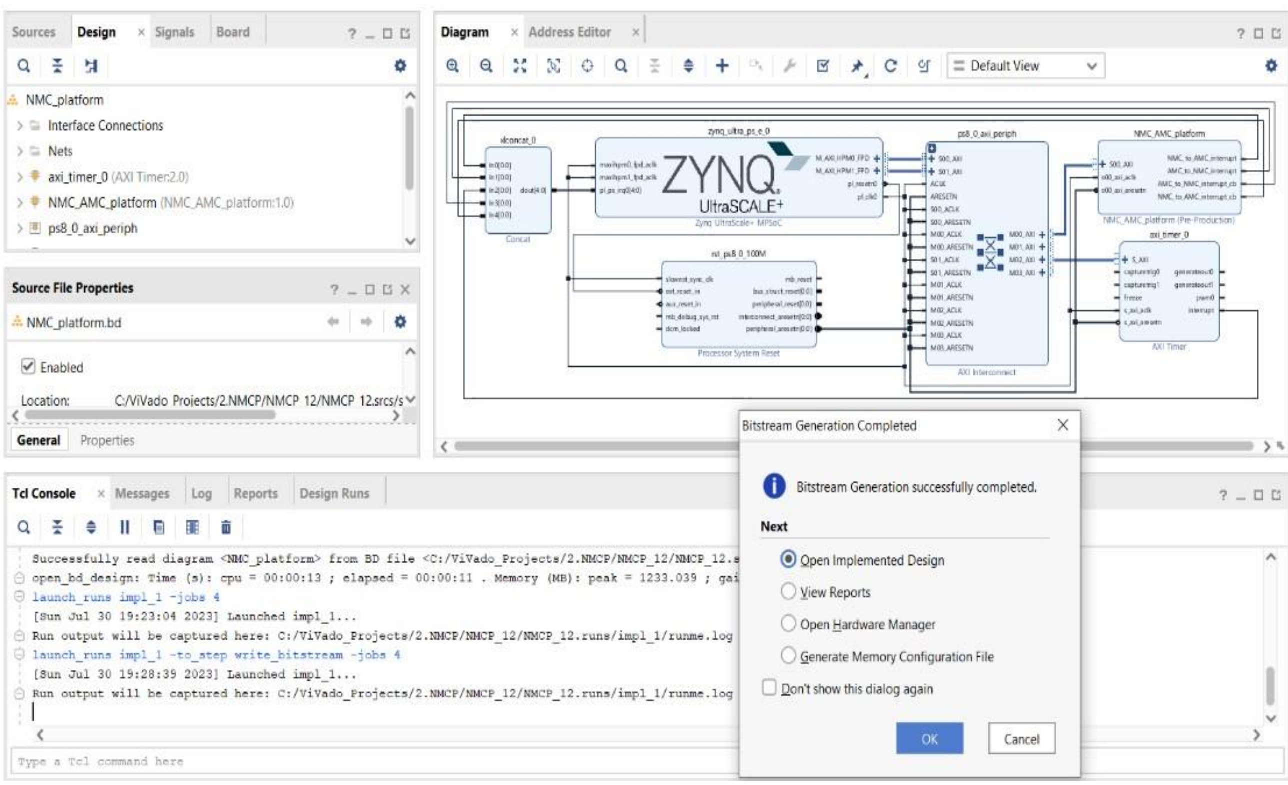

Figure 8: A successful completion of bitstream file generation.

The design in Vivado is exported to Vitis to add three domains for the three APU Cortex-A53_0,_1, and _2. In Fig.9, these domains are listed in the wrapper file.

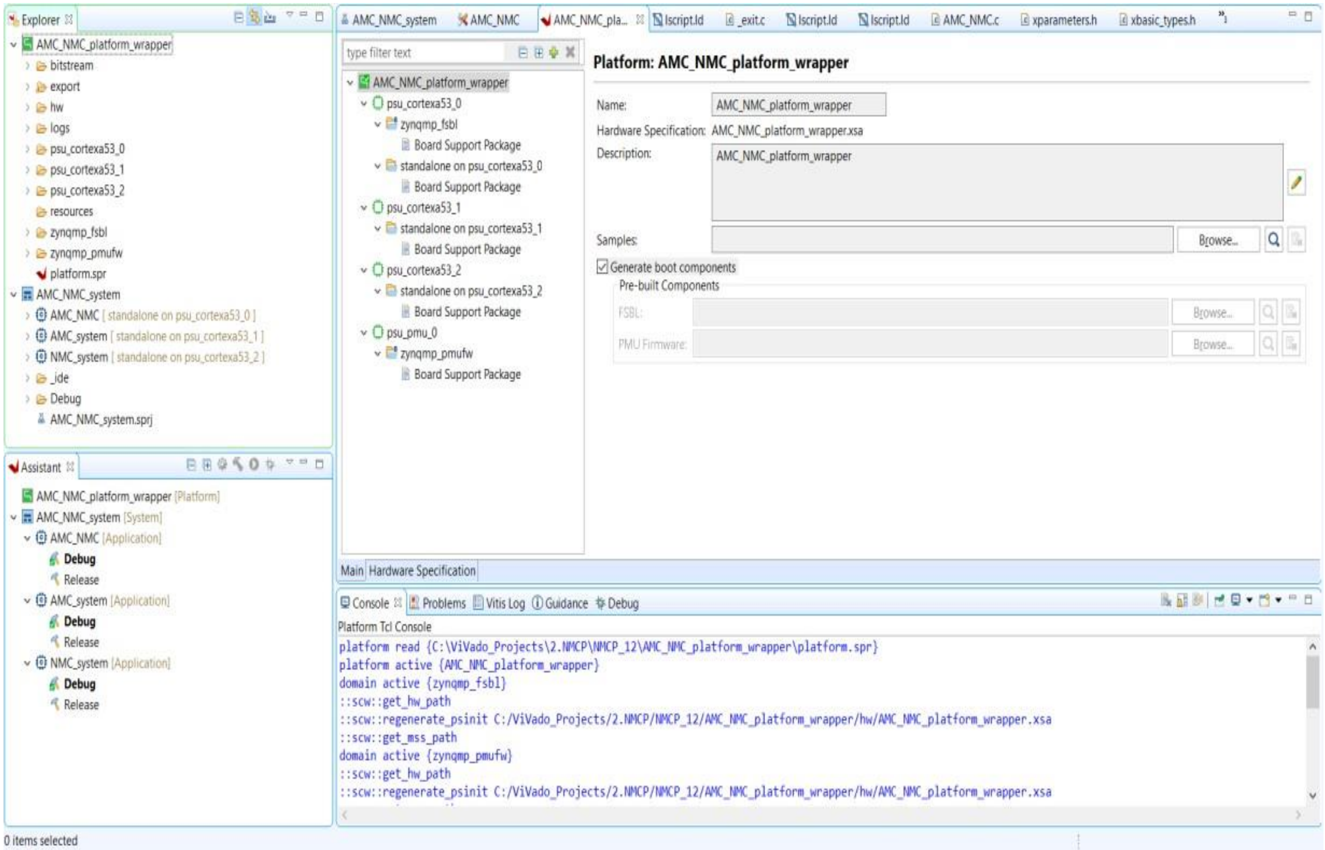

Figure 9: The AMC_NMC_platform_Wrapper after adding the dedicated domains.

The developed codes are loaded to the corresponding processor and tested individually successfully.

VI. Conclusion

The MPSoc FPGA envisioned technologies have capability of performing the satellite networking tasks in a distributed processing system. The Contact Plane (CP) is designed and successfully implemented as a shell to a proposed LLC Layer Management Entities (LLME) that has an interface to the MLME. it is in charge of configuration and initialization of the satellite constellation node to concurrent dual mode of operation, and management of the network traffics through ISL.

Acknowledgment

The authors gratefully acknowledge the financial support of the "High Institute for Engineering and Technology at Obour" and "Egyptian Space Agency" of the research in a project's frame of "Satellite Constellation Networking".

References

[1] IEEE Standard for Information Technology, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," Part 11: IEEE Standard 802.11-2020, 2020.

[2] Zhu Tang, Zhenqian Feng, Wanrong yu, and Baokang Zhao, "A Quasi-dynamic Inter-satellite Link Reassignment Method for LEO Satellite Networks," WASA 2015, LNCS 9204, pp. 497–507, 2015.

[3] Min Jia, Siyu Zhu, Linfang Wang, Qing Guo, Haitao Wang, and Zhihui Liu, “Routing Algorithm with Virtual Topology Toward to Huge Numbers of LEO Mobile Satellite Network Based on SDN,” Mobile Networks and Applications, 23, 2, pp. 285-300, 2018.

[4] Quan Chen, Jianming Guo, Lei Yang, Xianfeng Liu, and Xiaoqian Chen, “Topology Virtualization and Dynamics Shielding,” IEEE Communication Letters, 24(2), pp.433-437, 2020.

[5] Microship,“ATWILC3000A-Single-Chip-IEEE-802.11-b-g-n-Link-Controller-with-Integrated -Bluetooth,” Datasheet, Microchip Technology Inc, no.2, 2021.

[6] Wi-Fi Alliance, P2P Technical Group, “Wi-Fi Peer-to-Peer (P2P) Technical Specification,” Wi-Fi Alliance, no.3, pp..24-77, 2014.

[7] ARM, “GIC-400 Generic Interrupt Technical Reference Manual,” 2012.

[8] XLINX, “ug1085-zynq-ultrascale-trm Zynq UltraScale+ Device Technical Reference Manual,” 2020.

[9] XLINX, “ug1400-vitis-embedded Embedded Software Development,” Vitis Unified Software Platform Documentation, 2020.

[10] “Iridium satellite constellation”, Wikipedia, free encyclopedia, May.2023, en.wikipedia.org /wiki/ Iridium_ satellite_constellation.