*IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*
*e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 11, Issue 4, Ver. III (Jul.-Aug .2016), PP 17-27*
*www.iosrjournals.org*

# Designing of Avionics Full Duplex Switch on Netfpga

Ch.Sai Sreenidhi[1], S.V.R.Manimala[2],
[1]*Electronics and communication engineering, Master of Engineering,MVSR engineering college, mandal, Nadergul,saroor nagar mandal,Rangareddy,Hyderabad, An institute affiliated to Osmania University, Telangana,india,pincode-*
[2]*Associate Professor, Electronics and communication engineering, MVSR engineering college,nadergul,saroor nagar Rangareddy,Hyderabad. An institute affiliated toOsmania University Telangana,india,pincode-*

***Abstract***: *Avionics Full-Duplex Switched Ethernet Network(AFDX), Originally The Word "Avionics" is the combination of the Aviation and electronics, which is defined as electronics of aircrafts, artificial satellites and spacecrafts. MIL-1553 as its communication protocol for Avionics projects. But MIL-1553 communication is half duplex, asynchronous and works at 1 Mbps which may not be able to cater to the future communication performance requirements of bandwidth and maintainability. So a need exists to improve bandwidth, reliability and maintainability while reducing the physical dimensions of size, weight and number of connectors. A new network infrastructure is needed to provide more flexibility in avionics system design and cope with the increasing number of interconnections between systems. In this work an 8 -port AFDX switch has been designed on NET FPGA to form a deterministic network. This network can connect all the AFDX End systems through AFDX switches via Virtual links. Finally the link utilizes the data speed of 100Mbps full-duplex switched Ethernet according to ARINC-664P7. Main advantage of this work is redundancy.It can be achieved at output of end-system and complexity of the network can be reduced. The tools are used - for synthesis Xilinx Vivado 2014.4, xilinx 14.6 ise simulator for simulating and the packet flow at end-system can be captured by Wire-shark software.*

***Keyword:*** *Arinc 664p7, Mil-1553,Net Fpga,Afdx*

## I. Introduction

The Word "Avionics" is the combination of the Aviation and electronics, which could be defined as electronics of aircrafts, artificial satellites and spacecrafts. The Scientist Mr.Collinson mentions that the term avionics "was first used in the United State of America (USA) in the early 1950s and has since gained wide scale usage and acceptance" simply it is Avinoics Full-Duplex Etherenet network protocol. The paper is organized as follows. AFDX protocol is given in section II. Section III provides AFDX overview, while Section IV describes simulation results and end system test setup. Section V concludes the paper and provides guidelines for future scope.

## II. Afdx Protocol

Aircraft Data Networks (ADN) primarily utilizes the ARINC 429 standard. This standard, developed over thirty years ago and still widely used today, has proven to be highly reliable in safety critical applications. ARINC 429 networks, which can be found on a variety of aircraft from both Boeing and Airbus, utilize a unidirectional bus with a single transmitter and up to twenty receivers. A data word consists of 32 bits communicated over a twisted pair cable. There are two speeds of transmission: high speed operates at 100 Kbit/s and low speed operates at 12.5 Kbit/s.

Another standard, ARINC 629, introduced by Boeing for the 777 aircraft provides increased data speeds of up to 2 Mbit/s and allowing a maximum of 120 data terminals. ARINC 629 network operates without the use of a bus controller, thereby increasing the reliability of the network architecture. One of the primary draw-backs of this network type is that it is very specific to civil aircraft applications, requiring custom hardware which can add significant cost and development time to the aircraft.

ARINC 664 Part 7 [1] is defined as the next generation aircraft data network (AFDX). It is based on IEEE 802.3 Ethernet, enabling greater potential use of Commercial Off-The-Shelf (COTS) hardware, thereby reducing aircraft cost and development time. AFDX was developed by Airbus Industries for the A380, has since been accepted by Boeing and used on the Boeing 787 Dreamliner, and is being used or considered today for other applications.

## III. Afdx Overview

In this section covers about overview of AFDX and switch description and end system specification, and virtual link concept. Finally the brief explanation of Jitter and bag concepts.

Mainly AFDX network consists of three parts
1) AFDX Switch
2) AFDX End system
3) AFDX Virtual links

An aircraft data network has been described elsewhere in this standard as a profiled version of an IEEE 802.3 Ethernet utilizing IP addressing and related transport protocols. Part 7 describes a subset of this network, where quality of service including timely delivery is paramount. The AFDX network is a special case of a profiled network. A deterministic network may communicate with a wider profiled network and by inference, with a compliant network through routers or gateways. Fig.1. depicts this network hierarchy.

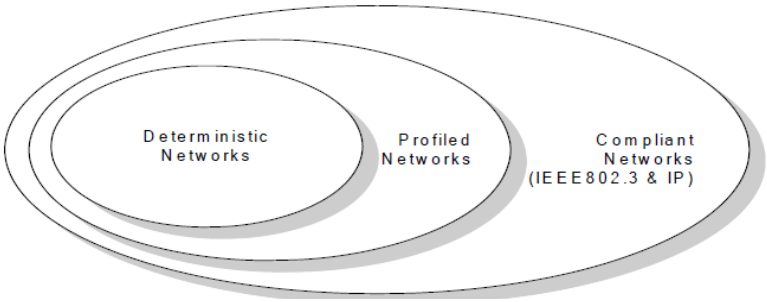


**Fig. 1** AFDX Network Hierarchy

### *A . AFDX Switch*

The AFDX switch consists of five functional blocks that interact with each other, as shown in figure 2

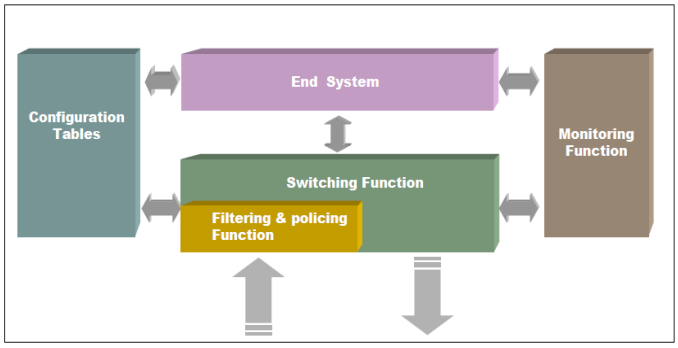


**Fig. 2** Functional elements of AFDX Switch

All frames arrive at the switch in the Filtering & Policing function stage where they are filtered in various steps. That apply rules about frame integrity, frame length, traffic budget and acceptable destination(s). The core of the switching activity is performed by the switching function. Frames filtered by the filtering and policing function are forwarded to the appropriate physical output ports where they leave the switch again.

The above all functions are controlled by configuration data contained in static configuration tables.The end system stage provides the means to communicate with the switch (receives frames dedicated to the switch and allows the switch to send frames). This is used for data loading and monitoring functions. All operations are monitored by monitoring functions that logs events such as the arrival of a frame or a failed CRC check and additionally creates statistics about the internal situation. Since the switch is a part of a network, it communicates with the network management function for operational information and for health related information. The frame size value shown in figure3

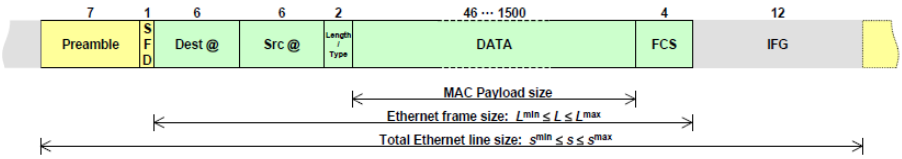


**Fig. 3** Frame size values

The actual time a frame occupies the Ethernet line (s). Therefore, all fields have to be taken into account: IFG (12 octets) + Preamble (7 octets) + SFD (1 octet) + MAC Frame size (L = 64 to 1518 octets).

***1) Filtering function***

The filtering causes the switch to distribute only valid frames to selected destinations. Upon arrival in a switch, each frame is examined and the contents of certain fields of the frame header (e.g. destination address field, frame check sequence field, etc.) and the construction of the frame itself are monitored:

• The frame size: whether the frame is either longer or shorter than the envelope allows
• The frame integrity: whether the FCS embedded in the frame matches the calculation upon reception
• The frame path: whether the destination requested by the content of the Destination Address field (which in case of the AFDX is the Virtual Link Identifier) of the arriving frame is permitted or not

If the frame properties do not comply with the configuration parameters, the frame is filtered i.e. discarded, and one or more MIB entries updated. The definition of the MIB entries as well as the update conditions and processes are specified in the relevant specification document.

The following aspects of the frame are tested as part of the filtering function:

• Destination address validity (Ethernet Address corresponds to a valid VL, including constant field)
• This VL is valid to be received on that destination port (according to the switch configuration table)
• Frame Check Sequence validity
• Ethernet frame size (L) is an integral number of octets (alignment)
• Ethernet frame size (L) in the range [64 octets, 1518 octets]
• Ethernet frame size (L) less than or equal to Lmax
• Ethernet frame size (S) greater than or equal to Smin, only in case where byte-based traffic policing is used

***2) Traffic policing***

This section describes a model of an algorithm that performs traffic based policing on the Destination Address basis. The Destination Address field contains the information that is used to identify a Virtual Link in an AFDX environment. A VirtualLink defines a traffic flow that has certain properties such as a group of recipients, or a minimum allowed gap between two frames. This traffic flow has to be maintained in a segregated way to guarantee its associated properties. In order to use in the context of this specification the same terminology as the one used in commercial documents the Destination Address is used synonymously with the term Virtual Link or "VL." The jitter phenomenon is dependent upon Virtual Link and switch characteristics, as it is a function of the traffic of the total of all Virtual Links arriving at a particular switch. If a Virtual Link spans several switches, the actual jitter may be different on each of the intermediate switch. However, maximum values for latency and jitter for any switch provide an upper bound.The traffic-policing model is described from an End-System point of view because End-Systems are the main traffic generators into the switch, shown in Figure 4. Model based description of the switch provides a switch centric view of the properties expected from a proper implementation of the switch.The switch may implement one of the two algorithm, either Byte-based or Frame based, or both of them. The choice of the algorithm will have an impact on the method used to prove the schedulability of the network.

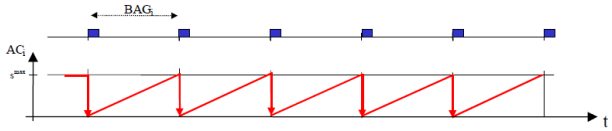


**Fig.4** Example of Traffic Without Jitter

Description of the policing algorithm: Initially, the ACcount for VLi (also called ACi, expressed in bytes) is set to

$$s_i^{max} \cdot \left(1 + \frac{J_{i,switch}}{BAG_i}\right)$$

• ACi is checked every time a frame of VLi arrives in the switch.
• Let s be the total Ethernet line size of the received frame (S = Ethernet frame size (L) + 20 octets; the 20 octets correspond to IFG+Preamble+SFD).

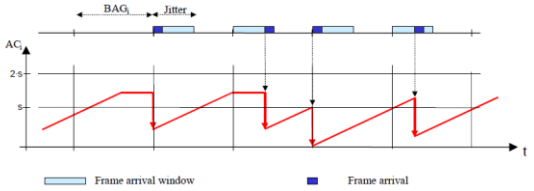


**Fig. 5** Example of Traffic with Jitter = BAG/2

A traffic policing mechanism should be implemented on the switch in order to ensure fault containment function of the network. Since a failed End System must not disturb the network, any frame belonging to a traffic flow that is not compliant with the network configuration should be discarded. The Configuration table has a relationship between MAC destination address, ACi,
BAG, Jitter, Smax, and eventually Smin in the case where byte-based traffic policing is used.

Traffic policing should be based on parameters: BAG, Jitter, Smax, and eventually Smin in the case where byte-based traffic policing is used. It should mechanize the algorithm described in paragraph "filtering and policing function introduction." For each VL or group of VLs sharing the same account, the Traffic Policing function should authorize one BAG value according to the configuration table. The Traffic Policing function of the Switch should at least be configurable for BAG values in the range 1ms to 128 ms.

For each VL or group of VLs sharing the same account, the Traffic Policing function should authorize one maximum Jitter value, according to the configuration table. The traffic policing function should at a minimum be configurable for maximum allowed Jitter values in the range 0 to 10 milliseconds.

**This implies that ACi has a maximum size of at least:**

$$s_i^{\max} \cdot \left(1 + \frac{10}{\mathrm{BAG}_i}\right)$$

The traffic policing function should be able to handle at least Ethernet frames sizes (L) in the range [64-1518] octets.

### *B . End System*

The main function of the End System (ES) is to provide services, which guarantee a secure and reliable data exchange to the partition software. Quality of Service (QoS) provides a method for categorizing traffic and for ensuring that particular categories of traffic will always flow across the network at the service level to which they are entitled, regardless of competing demands. A description of the End System communication stack is shown in Figure 6:

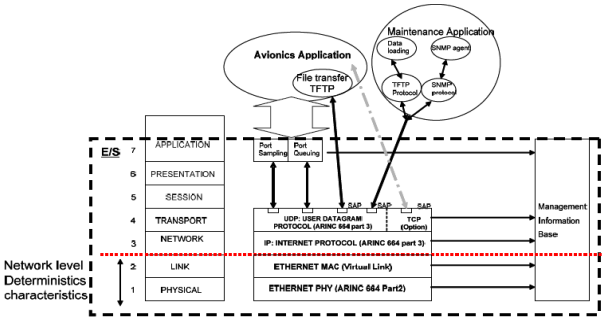


**Fig.6** End system protocol layers

### A. *Virtual Link*

Description of the "Virtual Link" concept is presented in Figure 7, since it is widely used in the paper. An end system may be designed to only receive VLs and not transmit VLs, or the contrary; therefore, an ES can originate or receive zero VLs. End-systems exchange Ethernet frames through VL. Only one End System within the Avionics network should be the source of any one VL.A Virtual Link is a conceptual communication object, which has the following properties:

• A Virtual Link defines a logical unidirectional connection from one source end-system to one or more destination end-systems, shown in Figure 7.

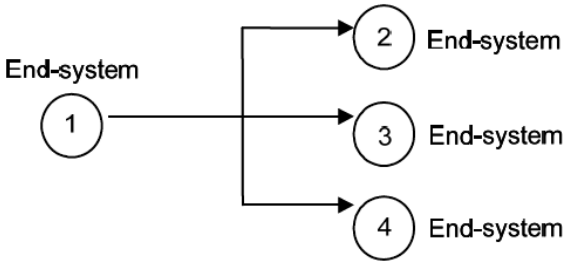


**Fig.7** virtual link path

• Each Virtual Link has a dedicated maximum bandwidth. This bandwidth is allocated by the System Integrator. The ES should provide logical isolation with respect to available bandwidth among the Virtual Link(s) it supports. Regardless of the attempted utilization of a VL by one partition, the available Bandwidth on any other VL is unaffected.

For each Virtual Link, the End System should maintain the ordering of data as delivered by a partition, for both transmission and reception (ordinal integrity).

B. ***Scheduling***

In a transmitting end system with multiple vls, the scheduler [2] multiplexes the Different flows coming from the regulators, as illustrated in figure 8

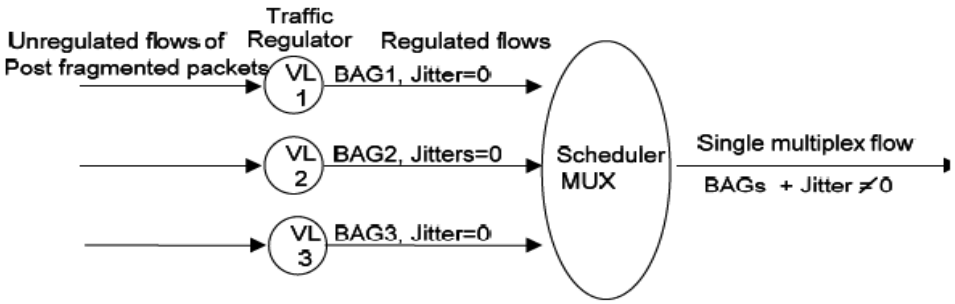


**Fig.8** Model of the Scheduled Flow Control Mechanism

At the output of the scheduler, for a given Virtual Link, frames can appear in abounded time interval. This interval is defined as the maximum admissible jitter. This jitter is introduced by the scheduler.

C. ***Jitter***

Jitter is variation of packet arrival time after bag. In transmission, the maximum allowed jitter on each VL at the output of the end system should comply with both of the following formula.

$$\begin{cases} max\_jitter \le 40\mu s + \dfrac{\sum_{i \in \{setof\ VLs\}} \left(20\ bytes + L^{max}{}_{bytes}\right) \times 8\ Bits/bytes}{Nbw_{\ bits/s}} \\ max\_jitter \le 500\mu s \end{cases}$$

D. ***MAC Addressing***

***1) MAC Destination Address***

A Virtual Link should only be identified by the MAC destination address as illustrated in Figure 9, and the MAC source address of AFDX frames should be the MAC unicast address used to identify the physical Ethernet interface. A MAC destination address in the AFDX frame should be a Group and Locally Administered address and should be compliant with the following format.

| 48 bits | |
|---|---|
| Constant field<br>32 bits | Virtual Link Identifier<br>16 bits |
| xxxx xx11 xxxx xxxx xxxx xxxx xxxx xxxx | |

**Fig.9** MAC Multicast Addressing Format

Each ES should get "constant field" and "Virtual Link Identifier" values from the system integrator. The values are not specified in ARINC Specification 664. The constant field should be the same for each ES in any given AFDX network. The least significant bit of the first byte indicates the group address (always = 1). In order to use the standard Ethernet frame, MAC group addresses should be used to send frames from End System to End System(s). The second to least significant bit of the first byte indicates the locally administered address (always = 1).

***2) MAC Destination Address***

The MAC Source address should be an Individual and Locally Administered address compliant with IEEE 802.3. The structure of the address is specified in the following format as shown in figure10

| Ethernet MAC Controller Identification (48-bits) | | | |
|---|---|---|---|
| Constant field: 24-bits | User_Defined_ID 16-bits | Interface_ID 3-bits | Constant field: 5-bits |
| "0000 0010 0000 0000 0000 0000 " | "nnnn nnnn nnnn nnnn" | "mmm" | "0 0000" |

**Fig.10** MAC Source Addressing Format

***E) Redundancy Management***

The Redundancy Management (RM) assumes that the network is working properly and, in particular, the deterministic properties are verified. As shown in figure11

Definitions:

• Redundant VL means that the same frames are sent through both network, A and B.

• Non-redundant VL means that (possibly different) frames are sent through either network A or B

On a per VL basis, the ES should be able to receive:

• A redundant VL and deliver to the partition one of the redundant data (RM active).

• A redundant VL and deliver to the partition both redundant data (RM not active).

• A non redundant VL on either interface and submit data from it to the partition (in this case, RM can be active or not)

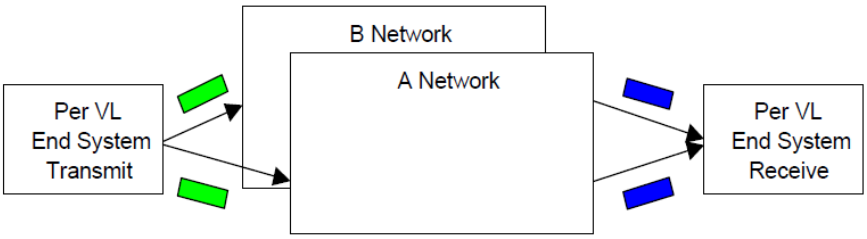


**Fig. 11** Network Redundancy Concept

## IV. Previous Techniquies

MIL-1553 as its communication protocol for its projects. But MIL-1553 communication is half duplex, asynchronous and works at 1 Mbps which may not be able to cater to the future communication performance requirements of bandwidth and maintainability and delay also more.So a need exists to improve bandwidth, reliability and maintainability while at the same time reducing the physical dimensions of size, weight and number of connectors. A new network infrastructure is needed to provide more flexibility in avionics system design and cope with the increasing number of interconnections between systems.

**Table I:** Comparision Of Avionics Protocols

| S.No | Attribute | AFDX | FC | CAN | TTP | MIL1553B |
|---|---|---|---|---|---|---|
| 1 | Max Frame Length | 1518 bytes | 2112 bytes | 8 bytes | 240 bytes | 20 bits |
| 2 | Frame Types | Normal | Normal | data, remote, error, overload | initialization, normal, x-frames | Normal |
| 3 | Max bit rate of current implementations | 10/100 Mbit/s | 1 Gbit/s | 1 Mbit/s | 25 Mbit/s | 1Mbps |
| 4 | Media Access | Direct | Direct | CSMA/CA | TDMA | Direct/Transformer coupled |
| 5 | Max bus length | <100 m | 50 m | 40 m recommended | typically < 100 m (not limited by protocol) | 20 ft (TC); 1ft(Direct) |

## V. Proposed Project

The proposed project is implementation of 8 -port AFDX switch on ZYNQ FPGA and also to form a deterministic network means that all the AFDX End systems connected through AFDX switches via Virtual links,finaly the utilizes the data speed is 100Mbps full-duplex switched Ethernet according to ARINC-664P7 protocol. Main advantages of this project is redundancy can be achieved i.e. there is no repetition of frames at out put of end-system and another one is complexity of the network can be reduced. The tools are used - for synthesis Xilinx Vivado 2014.4 and xilinx 14.6 ISE simulator and the packet flow at end-system can be captured by Wire-shark software. The proposed architecture of 8X8 AFDX switch as shown below figure 11

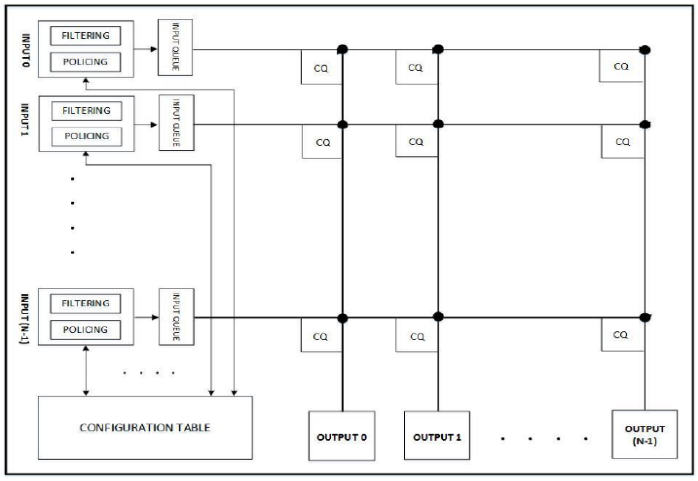


**Fig.11** proposed architecture of 8X8 AFDX switch

## VI. Results

This section discusses the proposed hardware architecture to achieve high throughput and low latency of multicast AFDX frames through an 8X8 crossbar fabric. Input frames are first filtered and policed and checked for validity before being transmitted across the fabric. The proposed architecture will be implemented on ZYNQ board and verified for functionality. It should be noted that the architecture has been designed keeping in mind the resource constraints of the FPGA as well as the requirements of the protocol as outlined in ARINC 664-P7.

### *A) Major Requirements*

The major requirements need for this project are queuing scheme, configuration table , input memory ,cross point buffers and AFDX switch test setup means switches are interconnect with end systems through virtual links. A Configuration Table containing 16 entries was created with the following parameters.

**Table II:** Configuration Table

| Address | VLID | Input Port | Output Ports | BAG | Priority | Jitter | Lmax | Lmin |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 3,4,5,6 | 16 | 0 | 10 | 1420 | 64 |
| 1 | 5 | 1 | 1,2,3 | 1 | 1 | 100 | 1380 | 64 |
| 2 | 7 | 2 | 5,6 | 64 | 0 | 40 | 1200 | 64 |
| 3 | 10 | 3 | 6,7 | 8 | 1 | 70 | 1400 | 64 |
| 4 | 13 | 4 | 0,1,7 | 32 | 1 | 200 | 720 | 64 |
| 5 | 15 | 5 | 1,3,5 | 128 | 0 | 30 | 716 | 64 |
| 6 | 16 | 6 | 2,4 | 64 | 0 | 80 | 324 | 64 |
| 7 | 18 | 7 | 0,1,3,4 | 4 | 1 | 40 | 623 | 64 |
| 8 | 22 | 0 | 2,3,6 | 8 | 1 | 90 | 389 | 64 |
| 9 | 25 | 1 | 0,4,7 | 4 | 0 | 500 | 1200 | 64 |
| 10 | 28 | 2 | 0,5,6,7 | 1 | 1 | 10 | 720 | 64 |
| 11 | 29 | 3 | 0,1,2,7 | 8 | 1 | 150 | 78 | 64 |
| 12 | 31 | 4 | 2,4,6 | 16 | 0 | 50 | 100 | 64 |
| 13 | 35 | 5 | 3,5,7 | 4 | 0 | 100 | 1300 | 64 |
| 14 | 40 | 6 | 0,2,3,4,6 | 32 | 1 | 20 | 1232 | 64 |
| 15 | 42 | 7 | 1,2,3 | 128 | 0 | 400 | 1000 | 64 |

**Table III:** To test the functionality, inputs were given as follows table 3

| Input Port | VLID | Timestamp | Frame Length |
|---|---|---|---|
| 0 | 1 | 2500 | 80 bytes |
| 1 | 5 | 2000 | 116 bytes |
| 2 | 7 | 500 | 72 bytes |
| 3 | 10 | 1000 | 96 bytes |
| 5 | 15 | 1500 | 88 bytes |
| 7 | 18 | 2000 | 104 bytes |
| 0 | 1 | 1602000 | 100 bytes |

### *B) Simulation Results*

The RTL was designed using Verilog HDL in Vivado Platform. Simulation was done using Vivado Simulator and the Simulation waveforms are shown below. Figure 12 shows the Configuration requests (*rqst_in*) from all inputs. It is shown in hexadecimal notation. Therefore, 04 means that the vector is **8'b00000100**, indicating that input 2 has sent a request. There is a signal named *match_found* which indicates the end of binary search, and a signal *final_addr*_o which indicates the addres at which the match was found, after which we see the various Configuration parameters. The vector *conf_data_vld_o* indicates the input port for which these values are intended. Figures 13 and 14 show a detailed picture of the same.

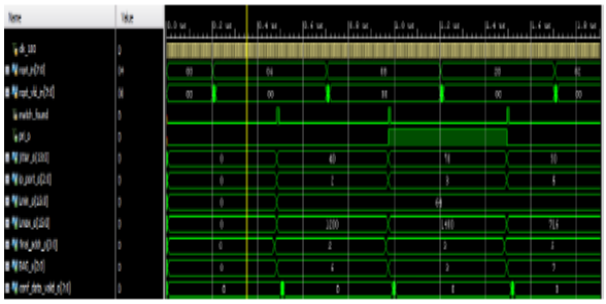

**Fig. 13** Configuration Table access requests and grants

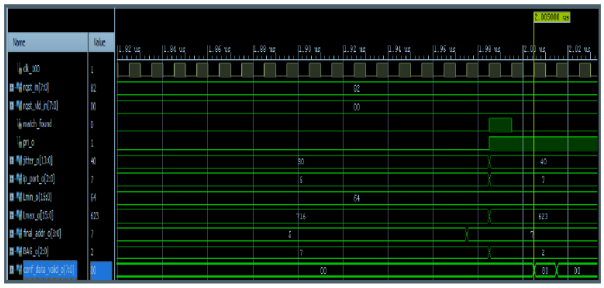

**Fig. 14** Configuration Table Interface

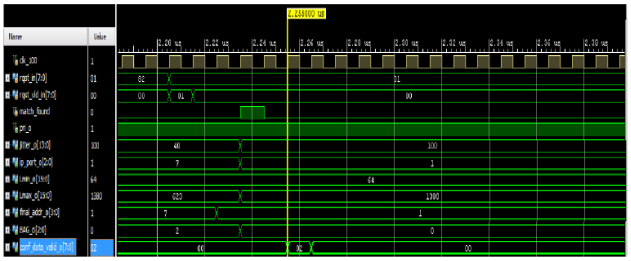

**Fig.15** Configuration Table Interface

In figures 14 and 15 we can see that the request input is **8'h82** which means that both inputs 7 and 1 are requesting at the same time. Here, the arbiter first performs the search for input 7, indicated by the value of *conf_data_vld_o* which is **8'h80.** Figures 16 and 17 show Filtering signals for two of the input ports (input 0 and input 2 respectively). They are enabled as soon as the configuration data valid corresponds to that input. The signals *cf_vld, frlen_vld, ip_port_vld, mult_8* refer to constant field validity, frame length validity, input port validity and integral number of octets respectively. Once all are asserted, *filt_complete* and *filt_valid* outputs are asserted.

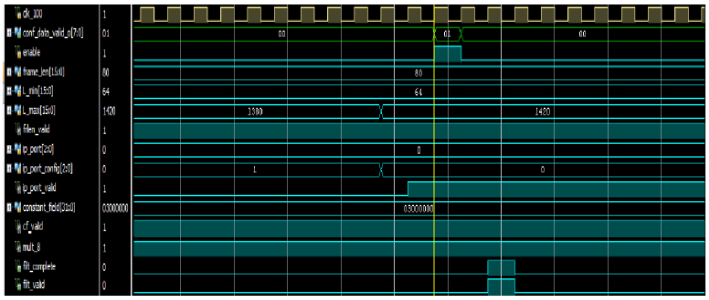

**Fig.16** Filtering at Input Port 0

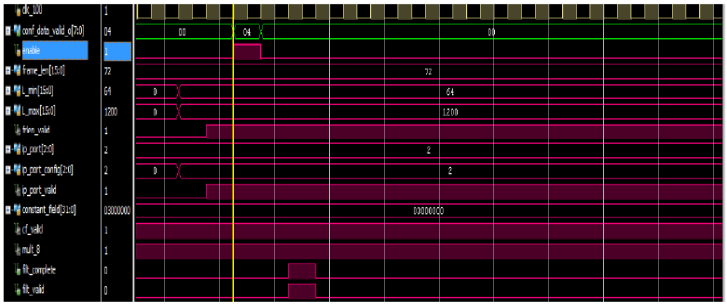

**Fig.17** Filtering at Input Port 2

Figures 18 and 19 show policing for two random input ports (input 3 and input 0 respectively). We can see that the *pol_en* signal is asserted after the configuration data is valid for that input. Following this, *ACI_vld* signal is asserted which indicates that the data from ACI Interface block is valid. The signal *first_frm* indicates that this is the first frame belonging to that VLID, in which case the frame is always valid. The ACI value to be written into ACI is (*ACIimax- Simax*). The output *ACI_new_numer_o* gives this value multiplied by 1000, and it is divided in the ACI interface block before being written into ACI memory. The validity of the numerator value is given by *ACI_new wrcmd_o*

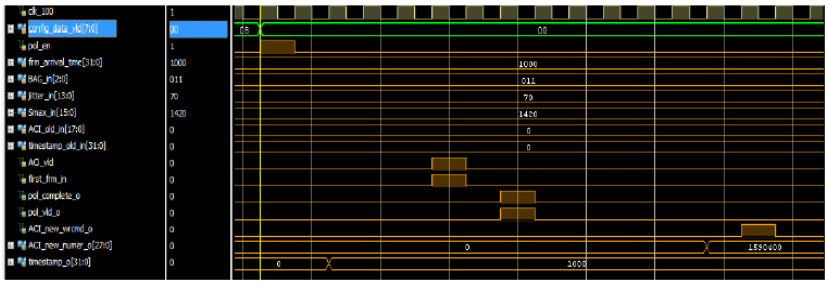

**Fig. 18** Policing at Input port 3

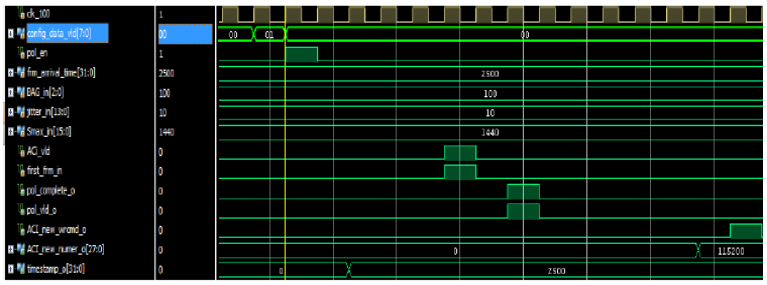

**Fig.19** Policing at Input port 0

### *C) End System Test Setup Results*

AFDX switch was inter connect with four end systems through virtual links then to form deterministic network. For end system test set-up[10] consider VL0,VL10,VL500,VL2000 then finally we observed packet flow can be captured by Wireshark packet capture tool software and data-speed maintain at 100Mbps. Figures 20, and 21,22,23 as snap-shorts of packet flow at can be captured by wire shark tool.

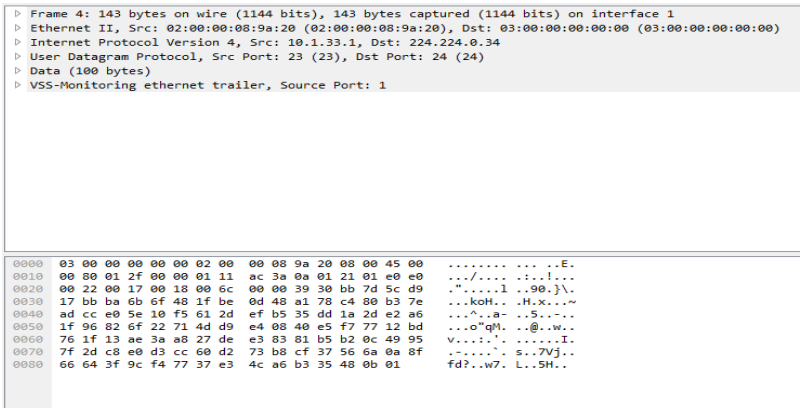


**Fig.20** Selecting VLLID 0 to transmit 100 bytes of data.

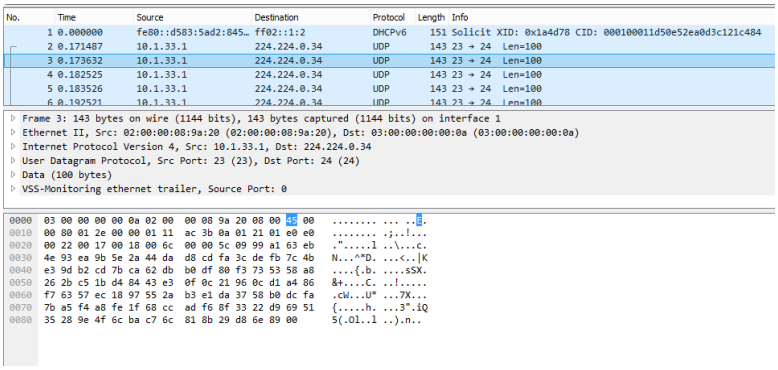


**Fig.21** Selecting VLLID 10 to transmit 100 bytes of data.

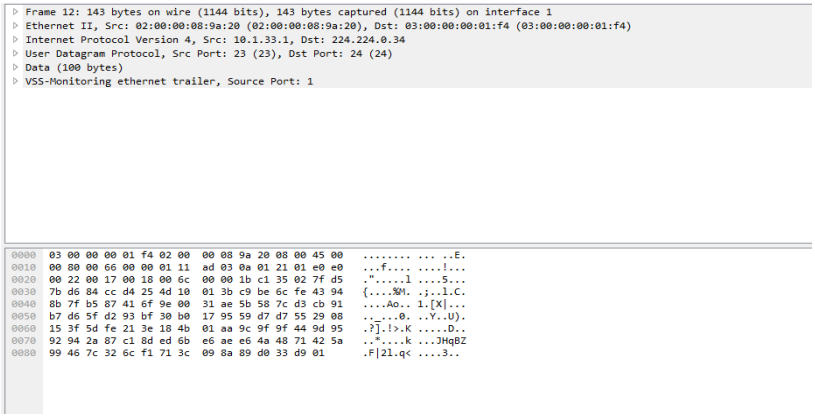


**Fig.22** Selecting VLLID 500 to transmit 100 bytes of data.

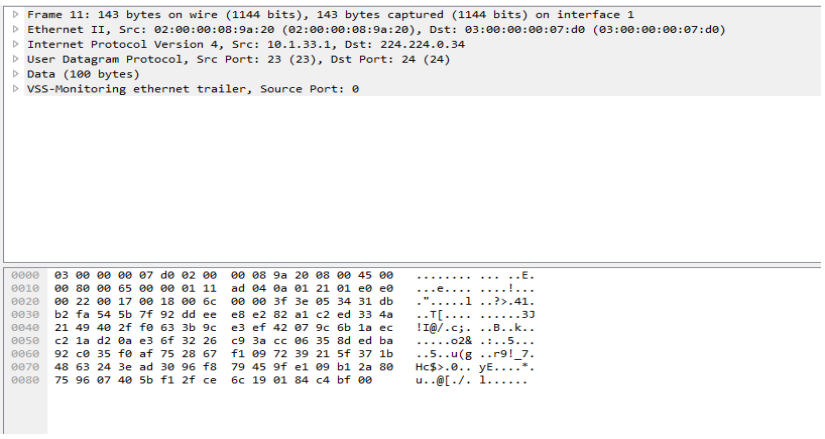


**Fig.23** Selecting VLLID 2000 to transmit 100 bytes of data

## VII. Conclusion

ARINC 664/AFDX (Avionics Full Duplex Switched Ethernet) protocol is being used as the back bone for all systems including flight controls, cockpit avionics, air-conditioning, power utilities, fuel systems, landing gear and other. Simply it mainly used in COTS, and high speed commercial Ethernet with provisions for guaranteed deterministic timing and redundancy required for avionics applications.

The RTL for the proposed 8 port AFDX switch architecture was designed and simulated with certain predefined test cases for 8-port Switch. A Configuration Table having 16 entries was given based on which the traffic was filtered and policed. The behavioral simulation was found to work as desired, thereby verifying the functionality of the design by NETFPGA, by using AFDX test setup, End systems are communicate with AFDX switch via virtual links at the data speed 100Mbps and redundancy also achieved, packet flow can be observed by Wire shark software and whole complexity of network reduced finally formed the deterministic network.

### *A) Future scope*

This work can be extended to 16 ports and 24 ports AFDX switch and it can be implement on other FPGA boards. AFDX protocol helpful to get high speed Ethernet network up to 1Gbits/sec.

## References

[1]. Air craft data network PART 7, ARNIC SPECIFICATION 664 P7-1.(Base Paper)
[2]. Chakkaphong Suthaputchakun, "Impact of End System Scheduling Policies on AFDX Performance in Avionic On-Board Data Network",pp:1-6, IEEE-Aug-2015
[3]. Nour EI-Din SafWat, Abdelhalim Zekry, Mohamed Abouelatta, "Avionics Full-duplex switched Ethernet (AFDX): Modeling and Simulation",pp:1-11, IEEE-March-2015
[4]. Meng Li, Michaël Lauer, Guchuan Zhu, "Determinism Enhancement of AFDX Networks via Frame Insertion and Sub-Virtual Link Aggregation", pp:1684-1692,IEEE-Aug-2014,
[5]. Henri Bauer, Jean-Luc Scharbarg, and Christian Fraboul," Improving the Worst-Case Delay Analysis of an AFDX Network Using an Optimized Trajectory Approach",IEEE-Nov-2010,pp:1-13.
[6]. Tasnim Hamza, Jean-Luc Scharbarg, Christian Fraboul," Priority assignment on an avionics switched Ethernet Network (QoS AFDX)", pp.1-8,IEEE-2014.
[7]. Chao Zhang and Jialuo Xiao," wireless asynchronous transfer mode based fly-by-wireless avionics network", pp.1-9,IEEE-2013.