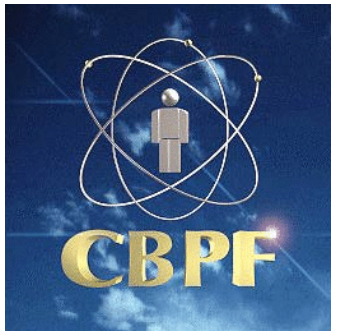


A Generic Firmware Core to Drive the Front-End GBT-SCAs

Federico Alessio, Cairo Caplan

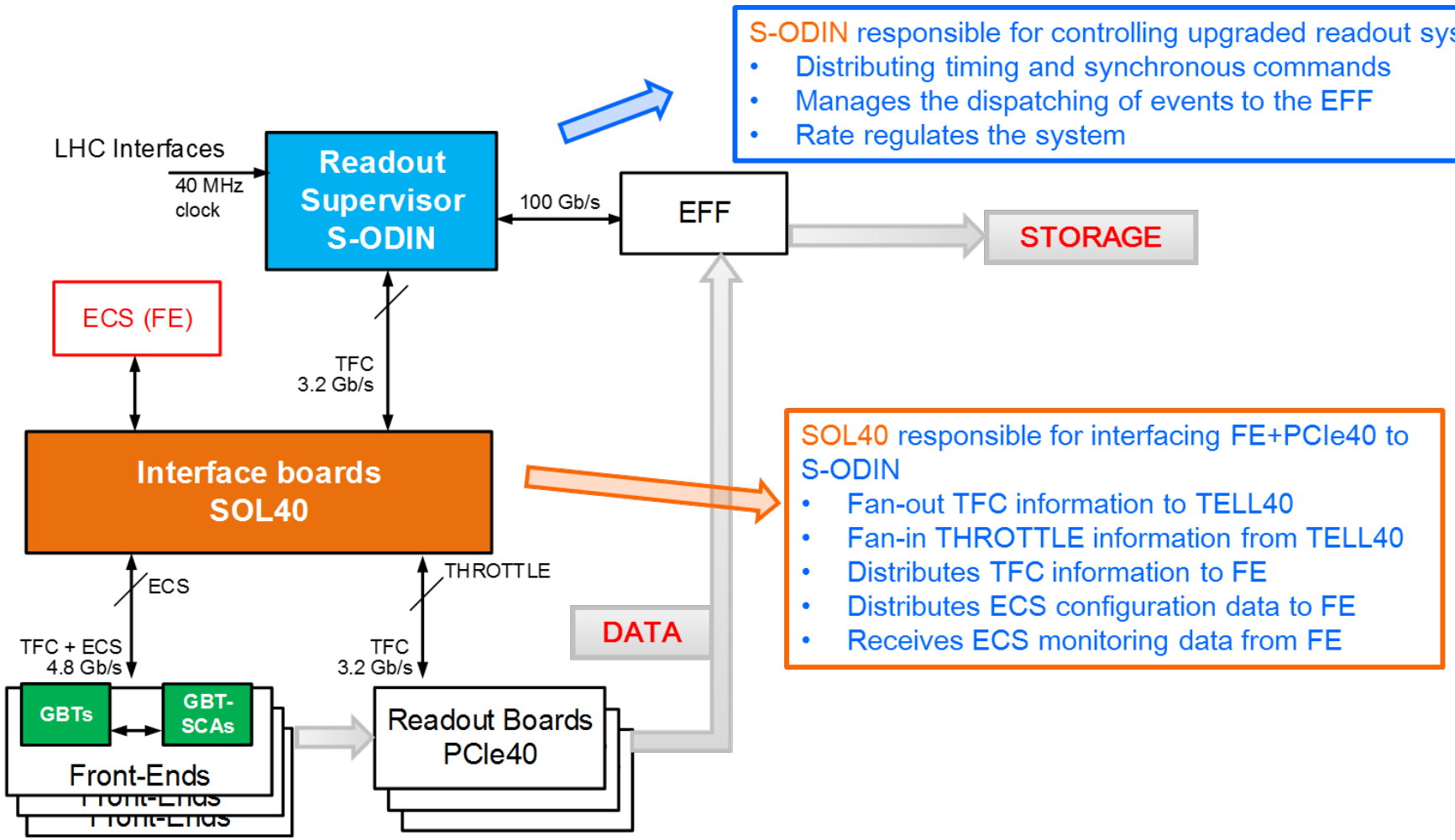
With acknowledgments to S. Bonacini, A. Caratelli, C. Gaspar, R. Jacobsson, K. Kloukinas, K. Wyllie

LHCb Electronics Upgrade Meeting, 09/10/2014

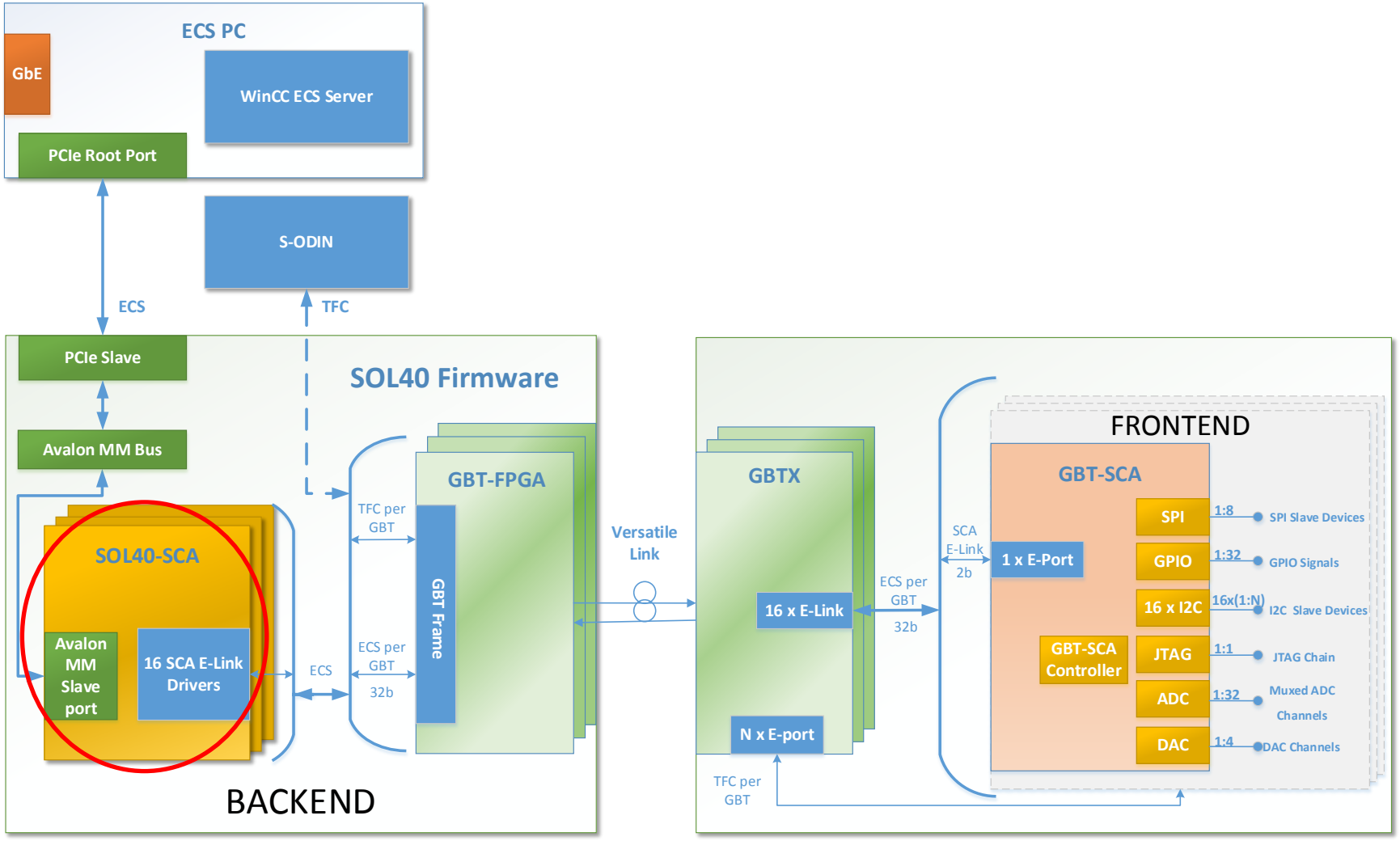
Description

- This project aims to provides a generic interface between the various slow control devices at the Front-End Electronics of the sub-detectors and the Control PCs through optical link and GBT chips
- It is achieved by developing the needed firmware layer between the LHCb ECS and the GBT-SCAs chips at the Front-Ends to be put in the SOL40 boards

The logical TFC system in the LHCb Upgrade



ECS to FE datapath



Off-Detector Electronics

On-Detector Electronics

Requirements

- Provide a generic hardware interface (FPGA) to the ECS configuration packets and ECS monitoring packets to the FE
- Build and encode/decode GBT-SCA compliant packets
- Serialize and deserialize command packets in the TFC+ECS command word according to GBT-SCA specs
- Support for all GBT-SCA protocols (SPI, I2C, JTAG, GPIO and ADC+DAC)
- Support for all GBT-SCA commands and channels in a modular fashion
- Support for many GBT-SCAs per GBT link and many GBT links per FPGA
- Robustness, reliability, programmability, flexibility

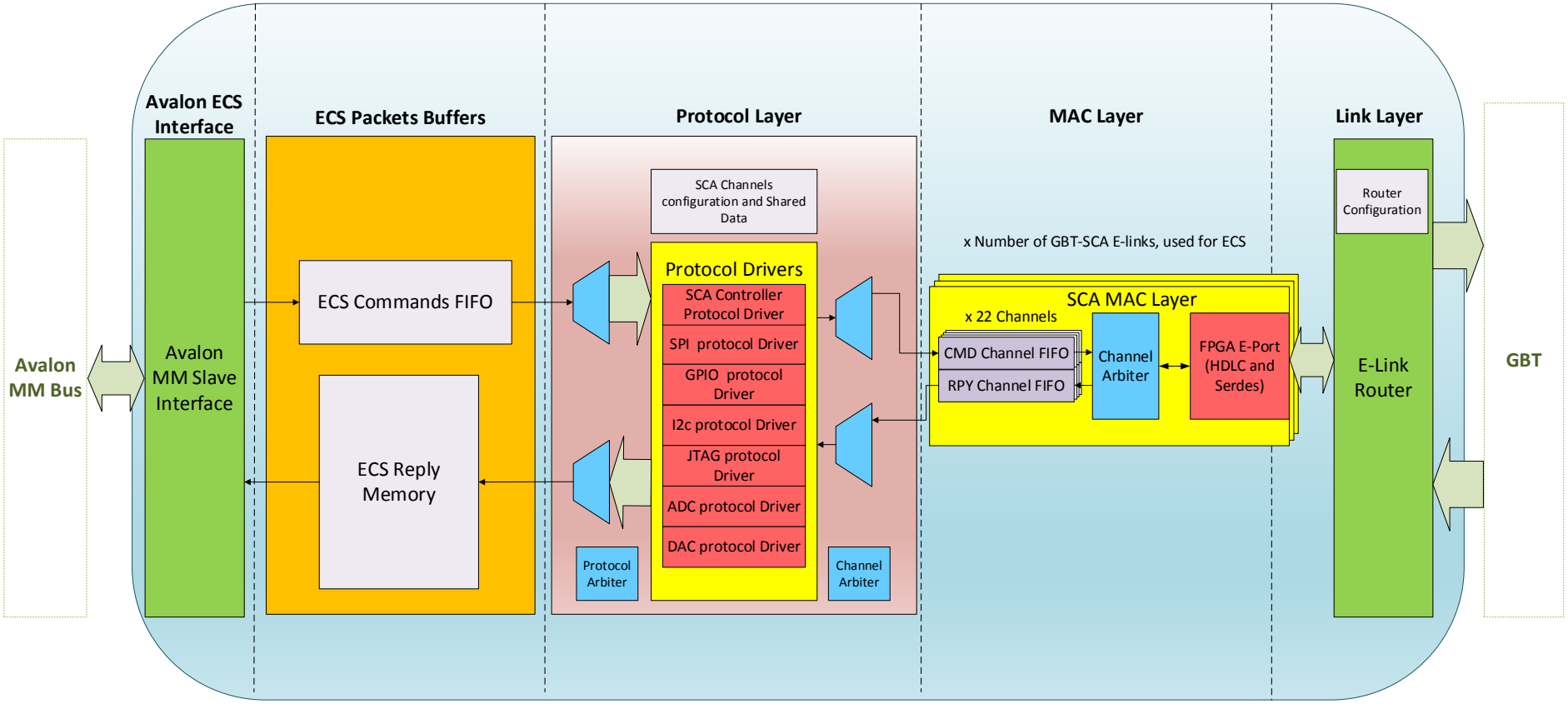
Targeted numbers

- GBT links / FPGA (SOL40): 36
- GBT-SCAs / GBT link: 16+1(EC field)
- Number of device channels supported by each GBT-SCA (GBT-SCA specs):
  - 16 I2C, 32 GPIO, 1 JTAG chain, 1 SPI,
  - 4 ADC and 1 DAC
- Bandwidth per GBT-SCA: 80 Mbps

→One SOL40 covers:

36*17 = 612 SCAs (=9792 I2C devices)

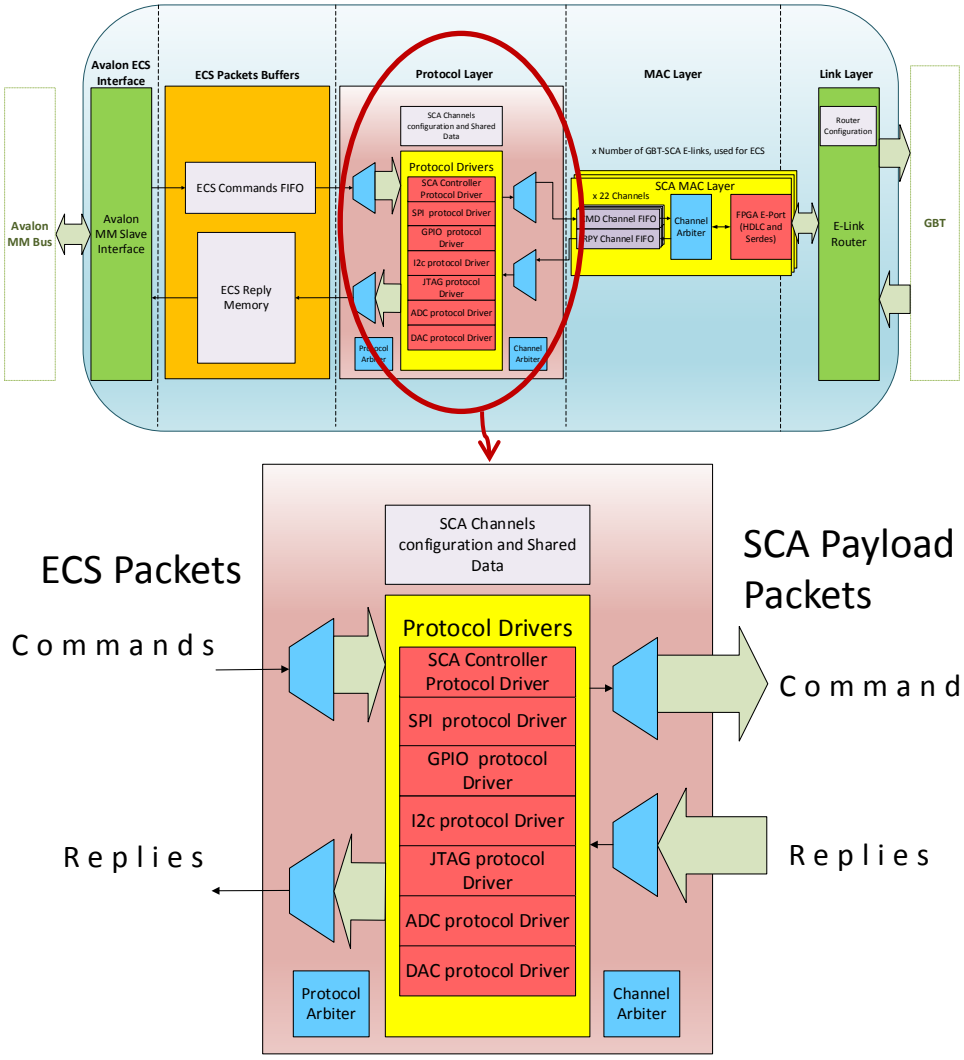
Architecture of the Core



- 1 core per GBT link: 36 cores per SOL40, instantiated in a programmable way
- each core supports 16(+1) GBT-SCAs, instantiated in a programmable way

Architecture of the Core

Protocol Layer

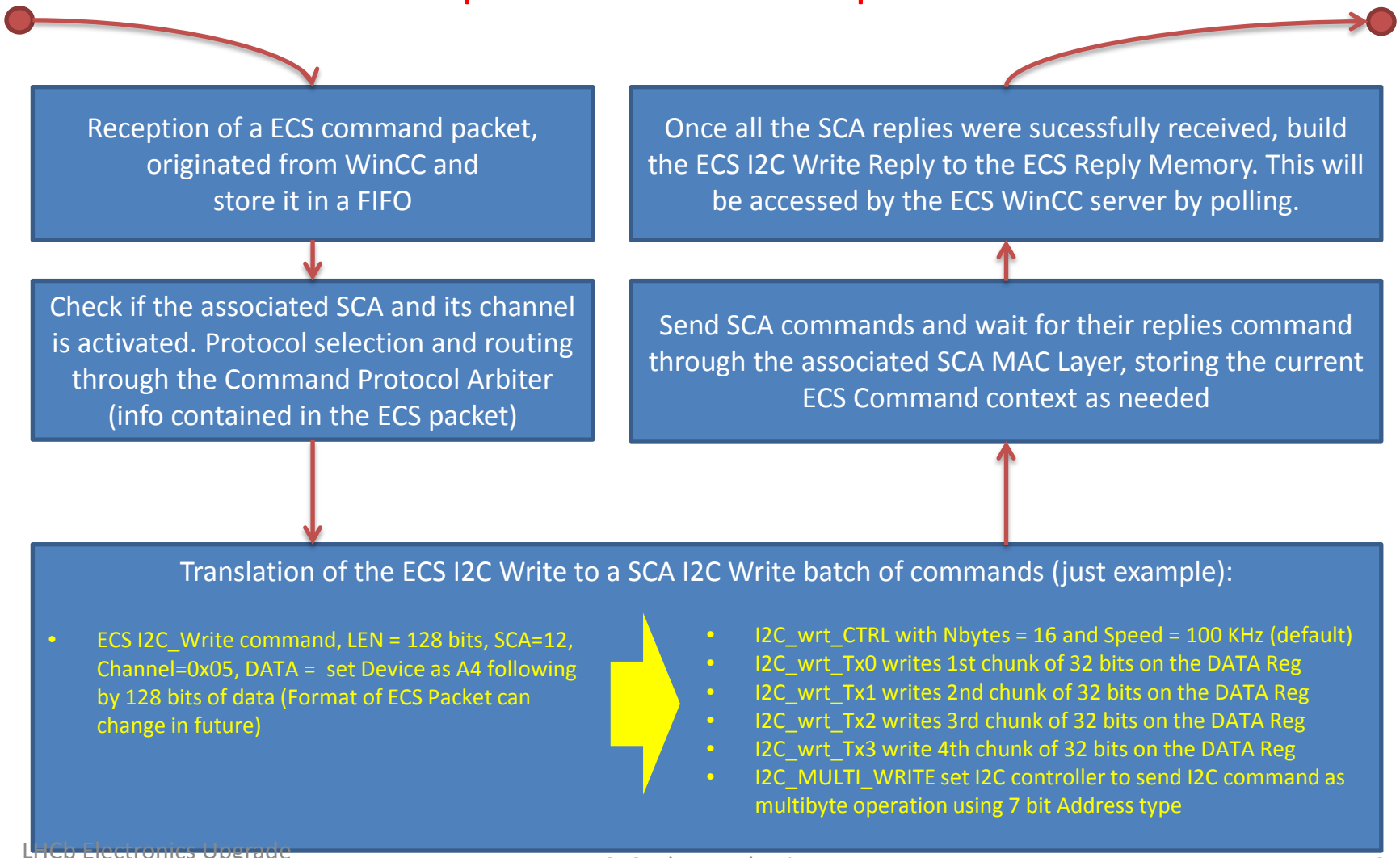


- Responsible to do the conversion between the ECS command packet, generated by software and build the correspondent GBT-SCA payload data packets
- One protocol driver for each of the SCA protocols
- Responsible for keeping information regarding the state of all the channels of each of the GBT-SCA driven by the block. Such as:
  - Activated SCAs, activated channels and timeout for replies.

Architecture of the Core

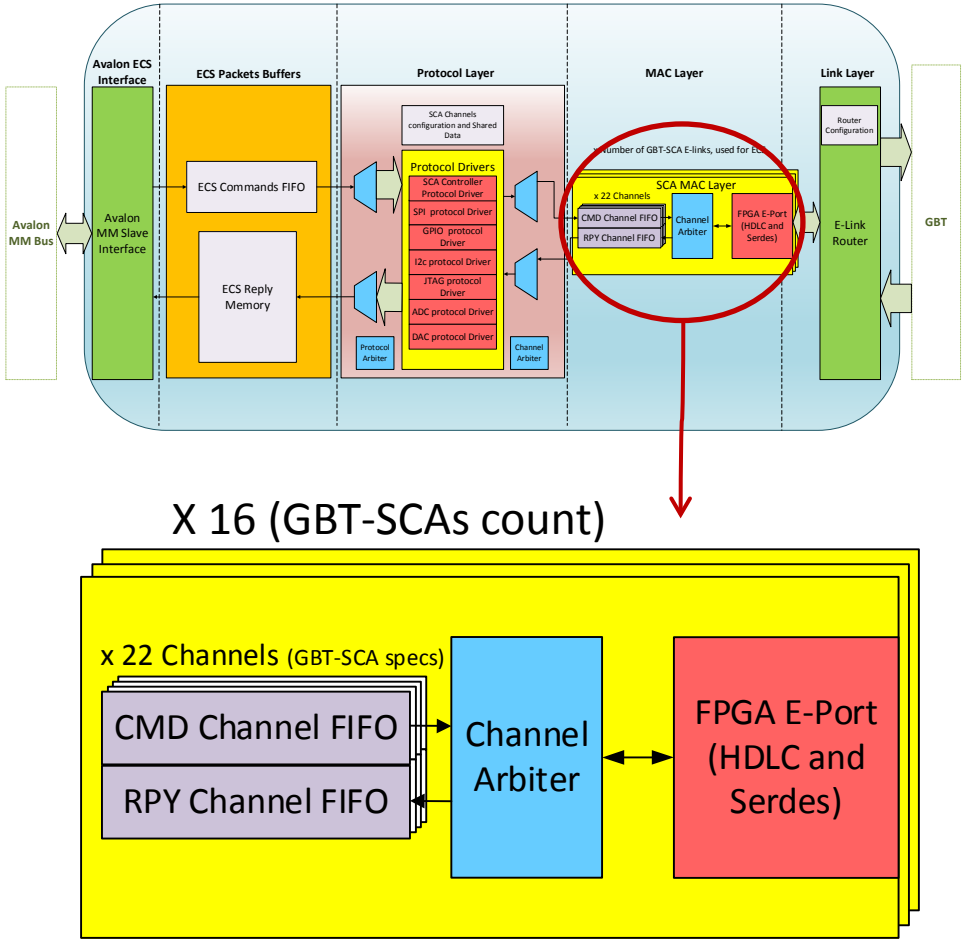
Protocol Layer

Example of a I2C write operation



Architecture of the Core

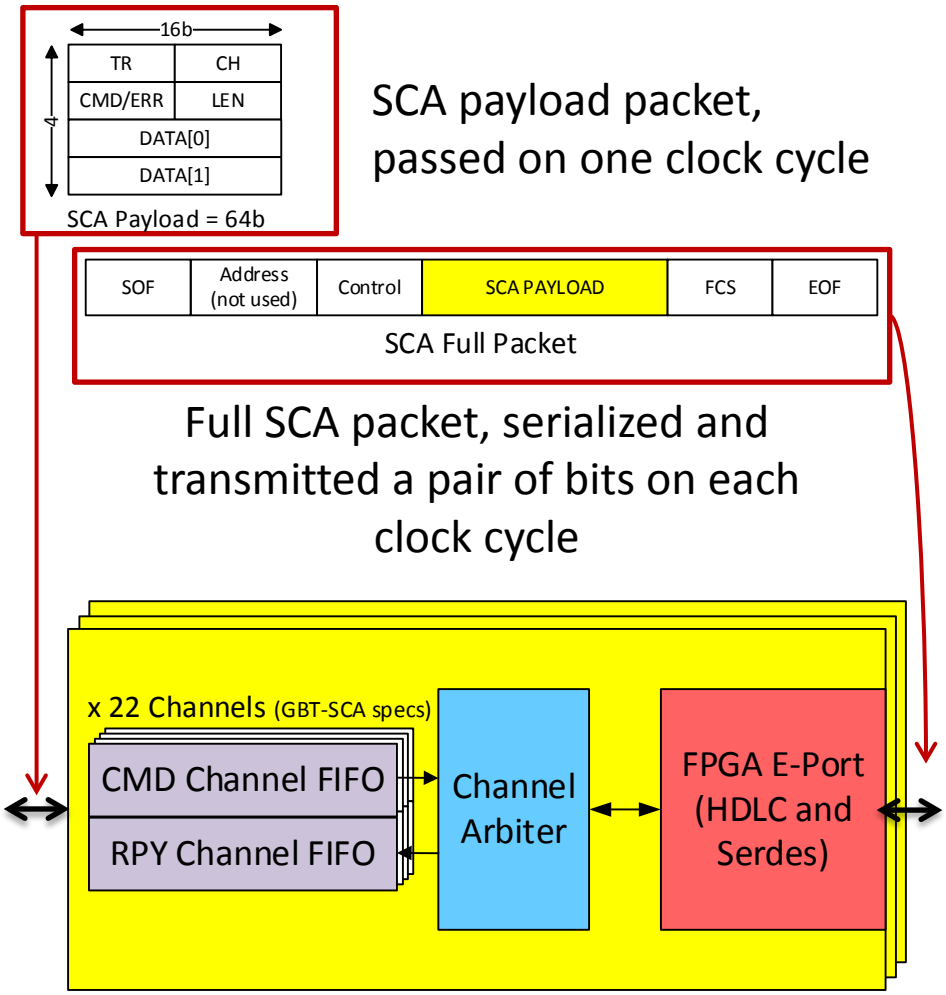
MAC Layer



- Responsible for providing access and data transmission to a GBT-SCA, through a pair of bits on the full TFC+ECS frame to a GBT
- This unit is replicated for each GBT-SCA available on same GBT link
- Currently contains a FIFO for each channel of the GBT-SCA, an arbiter for controlling the commands transmission to the link.
- The HDLC protocol encapsulation and the serialization to the pair of bits, through an e-Link, is based on the e-Port source code project by Sandro Bonacini.

Architecture of the Core

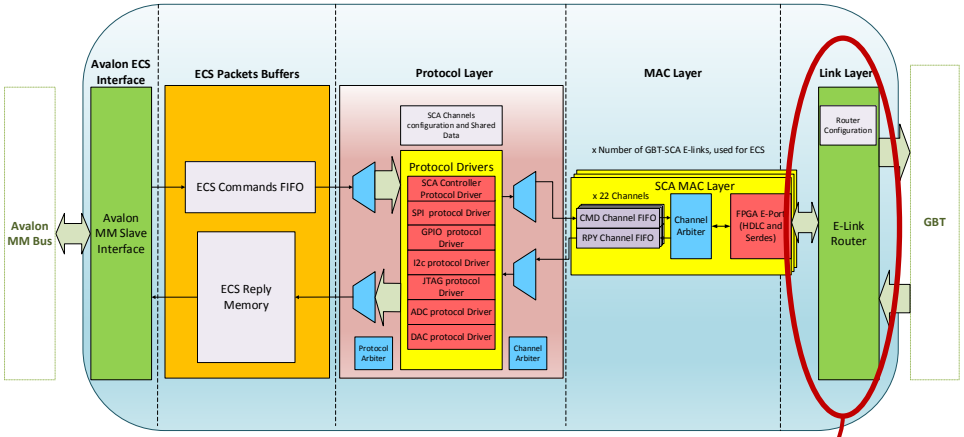
MAC Layer



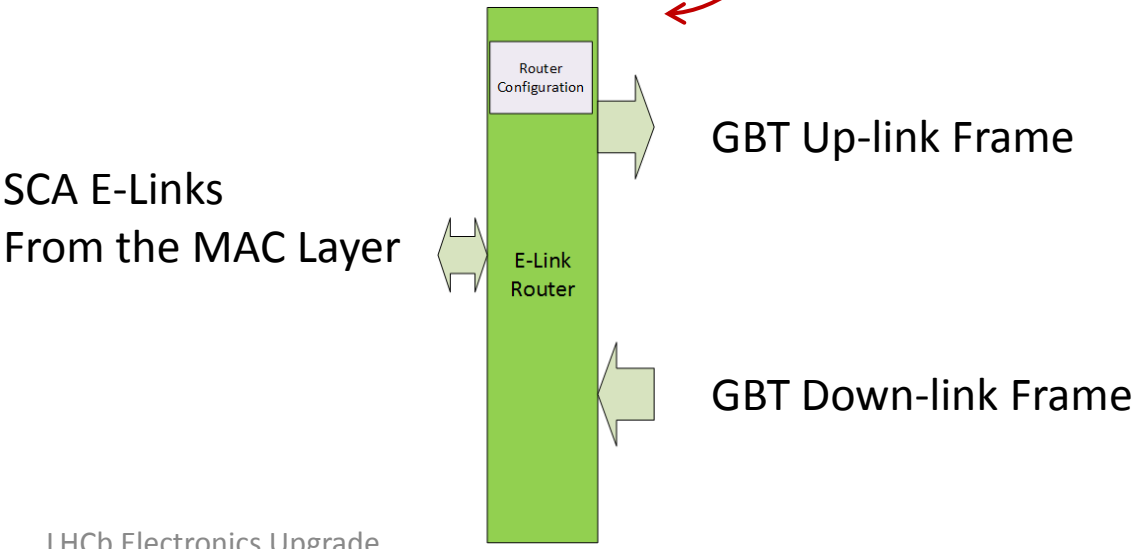
- The HDLC protocol provides connection type transmission to each link, frame error check and link specific control commands

Architecture of the Core

Link Layer



- This layers is responsible for routing the ECS bits to each of the e-Links – route the information to the right GBT-SCA and viceversa
- It makes possible the routing of GBT-SCA connections at runtime (simply by reconfiguring the matrix)



Features

- ✓ Scalable support number of SCAs, currently 16(+1)
- ✓ Throughput of 80 Mb/s for transmission and reception by each SCA e-Link, for a total of 1.28 Gb/s
- ✓ Support for all buses available at the GBT-SCA
- ✓ Vendor independent VHDL firmware code
- ✓ Protocol commands and number of activated channels configurable at runtime – simplification at the software level
- ✓ Parallel and modular control of all channels of each GBT-SCA
- ✓ Serialization and error detection with the HDLC encapsulation
- ✓ Support for packet retransmission

Compilation Results

Fitter Summary

Fitter Status	Successful - Fri Oct 03 17:06:00 2014
Quartus II 64-Bit Version	14.0.0 Build 200 06/17/2014 SJ Full Version
Revision Name	stfc_sca_Quartus
Top-level Entity Name	stfc_sca_top
Family	Stratix V
Device	5SGXEA7N2F45C3
Timing Models	Final
Logic utilization (in ALMs)	7,959 / 234,720 (3 %)
Total registers	16451
Total pins	586 / 1,064 (55 %)
Total virtual pins	0
Total block memory bits	2,880 / 52,428,800 (< 1 %)
Total DSP Blocks	0 / 256 (0 %)
Total HSSI STD RX PCSs	0 / 48 (0 %)
Total HSSI 10G RX PCSs	0 / 48 (0 %)
Total HSSI GEN3 RX PCSs	0 / 48 (0 %)
Total HSSI PMA RX Deserializers	0 / 48 (0 %)
Total HSSI STD TX PCSs	0 / 48 (0 %)
Total HSSI 10G TX PCSs	0 / 48 (0 %)
Total HSSI GEN3 TX PCSs	0 / 48 (0 %)
Total HSSI PMA TX Serializers	0 / 48 (0 %)
Total HSSI PIPE GEN1_2s	0 / 48 (0 %)
Total HSSI GEN3s	0 / 48 (0 %)
Total PLLs	0 / 92 (0 %)
Total DLLs	0 / 4 (0 %)

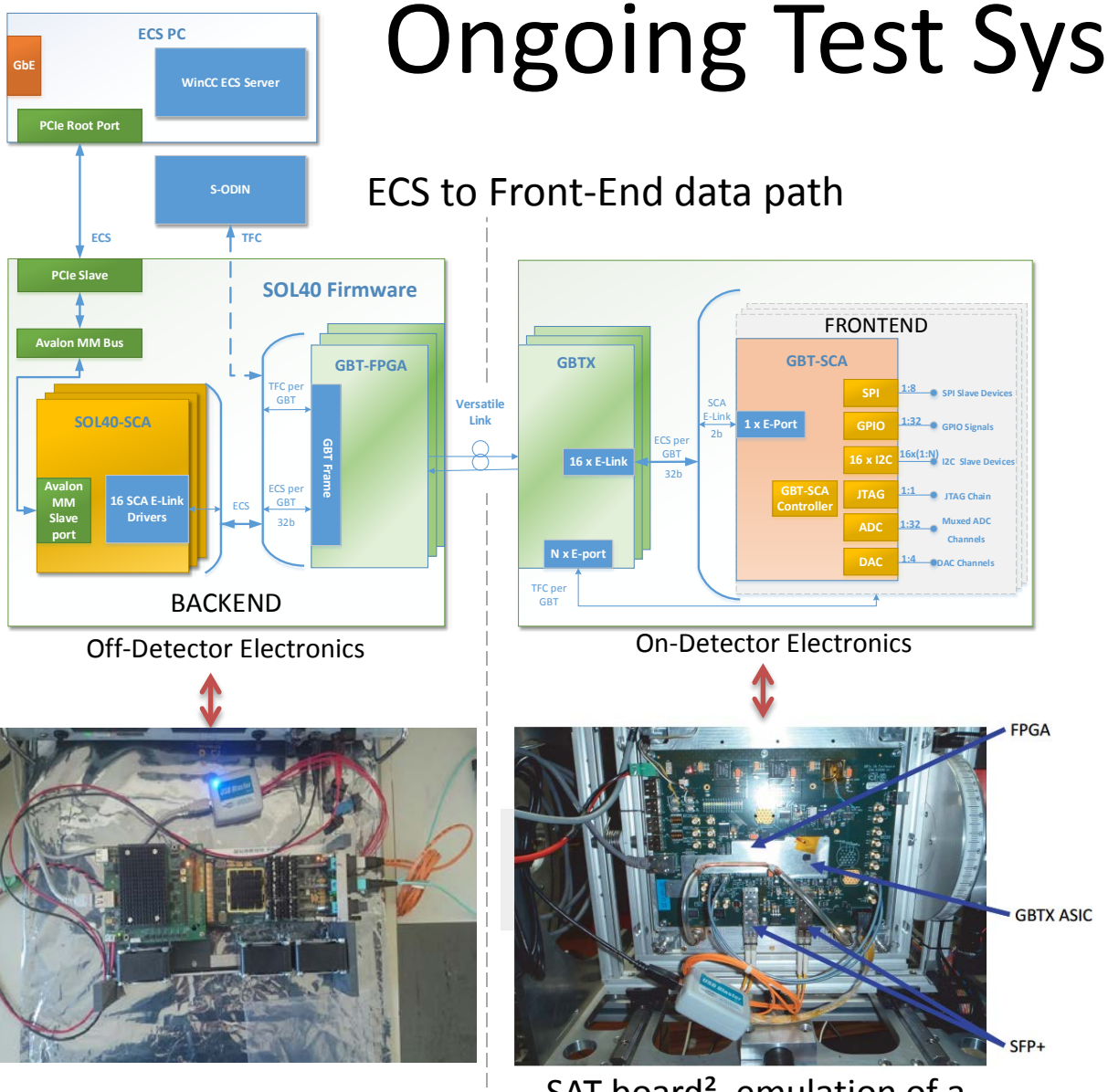
- For 4 GBT-SCA and only the SCA Controller Protocol Driver
- Protocol and MAC Layer only
- Estimation of Logic Utilization for all SCA and GBT Links in a SOL40 board:

36 * (16 SCAs *1780 + 985) = 36 * 29465 = 1060740 = 400% (!)

Fitter Resource Utilization by Entity

	Compilation Hierarchy Node	ALMs needed [=A-B+C]	Dedicated Logic Registers	Block Memory Bits
1	|stfc_sca_top	7958.5 (3.0)	16451 (0)	2880
1	|protocol_layer:protocol_layer_inst|	985.9 (702.0)	630 (254)	0
1	|sca_controller_protocol_driver:sca_controller_protocol_driver_inst|	239.8 (239.8)	376 (376)	0
2	|stfc_sca_mac_layer:\gen_mac_layer:0:stfc_sca_mac_layer_inst|	1775.0 (0.0)	3962 (0)	720
1	|channel_fifo:\gen_FIFOs:12:u_fifo|	57.0 (57.0)	129 (129)	0
2	|channel_fifo:\gen_FIFOs:13:u_fifo|	55.0 (55.0)	129 (129)	0
3	|channel_fifo:\gen_FIFOs:7:u_fifo|	55.1 (55.1)	129 (129)	0
4	|channel_fifo:\gen_FIFOs:1:u_fifo|	56.8 (56.8)	129 (129)	0
5	|channel_fifo:\gen_FIFOs:3:u_fifo|	55.2 (55.2)	129 (129)	0
6	|channel_fifo:\gen_FIFOs:11:u_fifo|	55.7 (55.7)	129 (129)	0
7	|channel_fifo:\gen_FIFOs:15:u_fifo|	55.3 (55.3)	129 (129)	0
8	|channel_fifo:\gen_FIFOs:14:u_fifo|	55.1 (55.1)	129 (129)	0
9	|channel_fifo:\gen_FIFOs:9:u_fifo|	57.0 (57.0)	129 (129)	0
10	|channel_fifo:\gen_FIFOs:16:u_fifo|	56.8 (56.8)	129 (129)	0
11	|channel_fifo:\gen_FIFOs:4:u_fifo|	56.4 (56.4)	129 (129)	0
12	|channel_fifo:\gen_FIFOs:21:u_fifo|	54.7 (54.7)	129 (129)	0
13	|channel_fifo:\gen_FIFOs:5:u_fifo|	56.6 (56.6)	129 (129)	0
14	|channel_fifo:\gen_FIFOs:2:u_fifo|	55.3 (55.3)	129 (129)	0
15	|channel_fifo:\gen_FIFOs:8:u_fifo|	55.7 (55.7)	129 (129)	0
16	|channel_fifo:\gen_FIFOs:18:u_fifo|	56.0 (56.0)	129 (129)	0
17	|channel_fifo:\gen_FIFOs:6:u_fifo|	55.6 (55.6)	129 (129)	0
18	|channel_fifo:\gen_FIFOs:17:u_fifo|	52.7 (52.7)	129 (129)	0
19	|channel_fifo:\gen_FIFOs:10:u_fifo|	49.4 (49.4)	129 (129)	0
20	|channel_fifo:\gen_FIFOs:19:u_fifo|	51.2 (51.2)	129 (129)	0
21	|channel_fifo:\gen_FIFOs:20:u_fifo|	50.9 (50.9)	129 (129)	0
22	|channel_fifo:\gen_FIFOs:0:u_fifo|	95.2 (95.2)	238 (238)	0
23	|fpga_elink:fpga_elink_inst|	278.4 (0.0)	365 (0)	544
24	|channel_arbiter:channel_arbiter_inst|	247.8 (247.8)	650 (650)	176
3	|stfc_sca_mac_layer:\gen_mac_layer:2:stfc_sca_mac_layer_inst|	1746.9 (0.0)	3954 (0)	720
4	|stfc_sca_mac_layer:\gen_mac_layer:3:stfc_sca_mac_layer_inst|	1752.4 (0.0)	3956 (0)	720
5	|stfc_sca_mac_layer:\gen_mac_layer:1:stfc_sca_mac_layer_inst|	1695.2 (0.0)	3949 (0)	720

Ongoing Test System



MiniDAQ evaluation board

SAT board[2], emulation of a frontend

- Emulated Front-End Slow control devices through a FPGA emulation of the GBT-SCA[1].
- ECS Commands generated by software
- 1 GBT Link , 1 GBT-SCA with the available protocols

1 - **The GBT-SCA, a Radiation Tolerant ASIC for Detector Control and Monitoring Applications in HEP Experiments** poster, by A. Caratelli et al.

2- **Test bench Development for the Radiation Hard GBTX ASIC** poster, by P. Leitao et al.

Next Steps

- Evaluate the functional part of the system on the on going test, using generic data stimulus
- Develop the remaining types of protocol drivers
- Define a first version of the interface between ECS and the firmware core
- Test the Core with actual devices and integrate it with an WinCC Software Module
- Optimize the code, regarding resource utilization for example
- Instantiate it in the main SOL40 code
- Timescale: end of 2014

Thank you