STAR-Dundee

HPPDSP

High Processing Power Digital Signal Processor

Steve Parkes,

Bruce Yu, Stuart Mills and Andy Whyte

Space Tech Centre, Univ. of Dundee

STAR-Dundee Ltd

STAR-Dundee

Outline

- Background
- System Architecture
- HPPDSP Board Design
- Some Basic Tests
- Software/API Architecture
- Conclusions

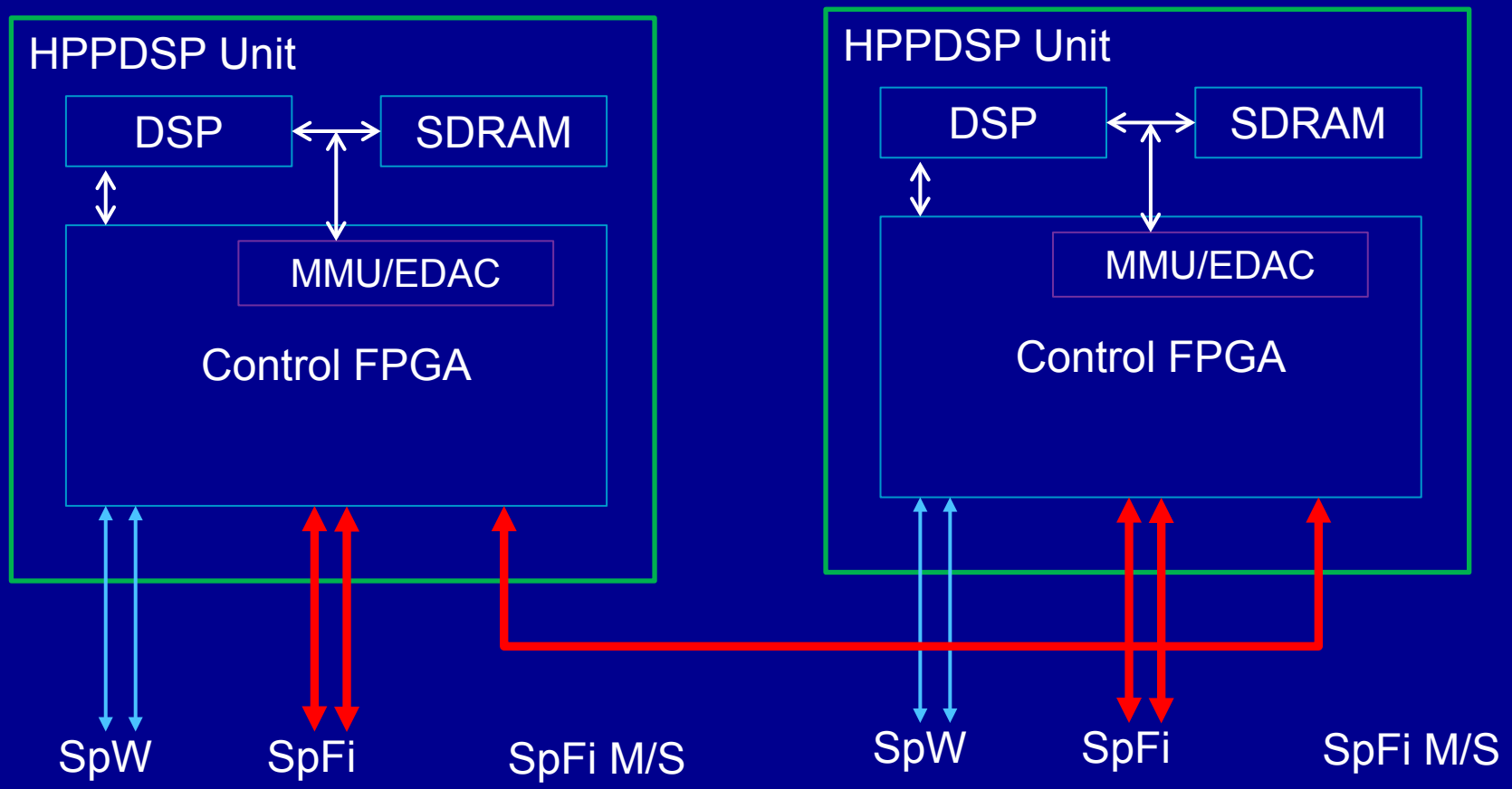
Background

- High data rate payload, requiring DSP with
  - High processing power
  - High throughput
- DSP processor
  - TI TMS320C6727B - 250MHz/300MHz (Floating-Point)
    - 2400 MIPS/1800 MFLOPS at 300MHz
  - External Memory Interface (EMIF)
    - A single bank of SDRAM – 133MHz
    - Plus a single bank of asynchronous memory space
  - Universal Host-Port Interface (UHPI)
    - Parallel interface – 32-bit data
    - Enable a host to access DSP memory space
- High throughput using SpW and SpFi interfaces

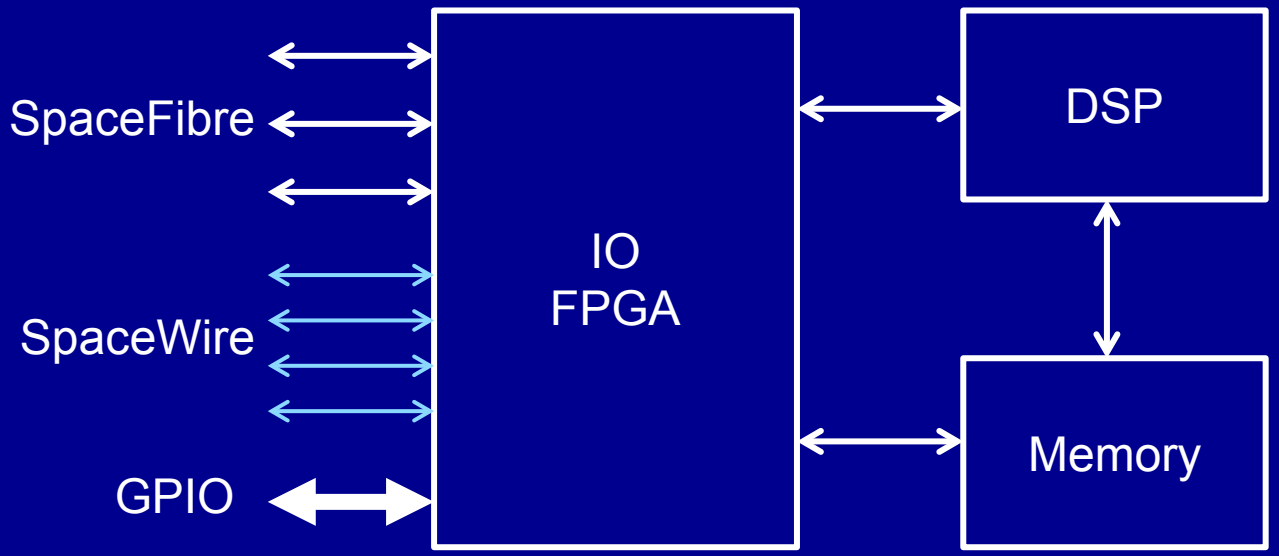
Background

- Rad-hardened DSP Processor
  - TI SMV320C6727B-SP, 250-MHz, Floating-Point
    - Total Ionizing Dose tolerance (TID 100Krad)
    - Single Event Latch-up immune (SEL 117 MeV cm^2/mg)
- No protection for external SDRAM
  - Need to add
    - EDAC design for SEUs
    - Memory management unit (MMU) for illegal writes

System Architecture Design

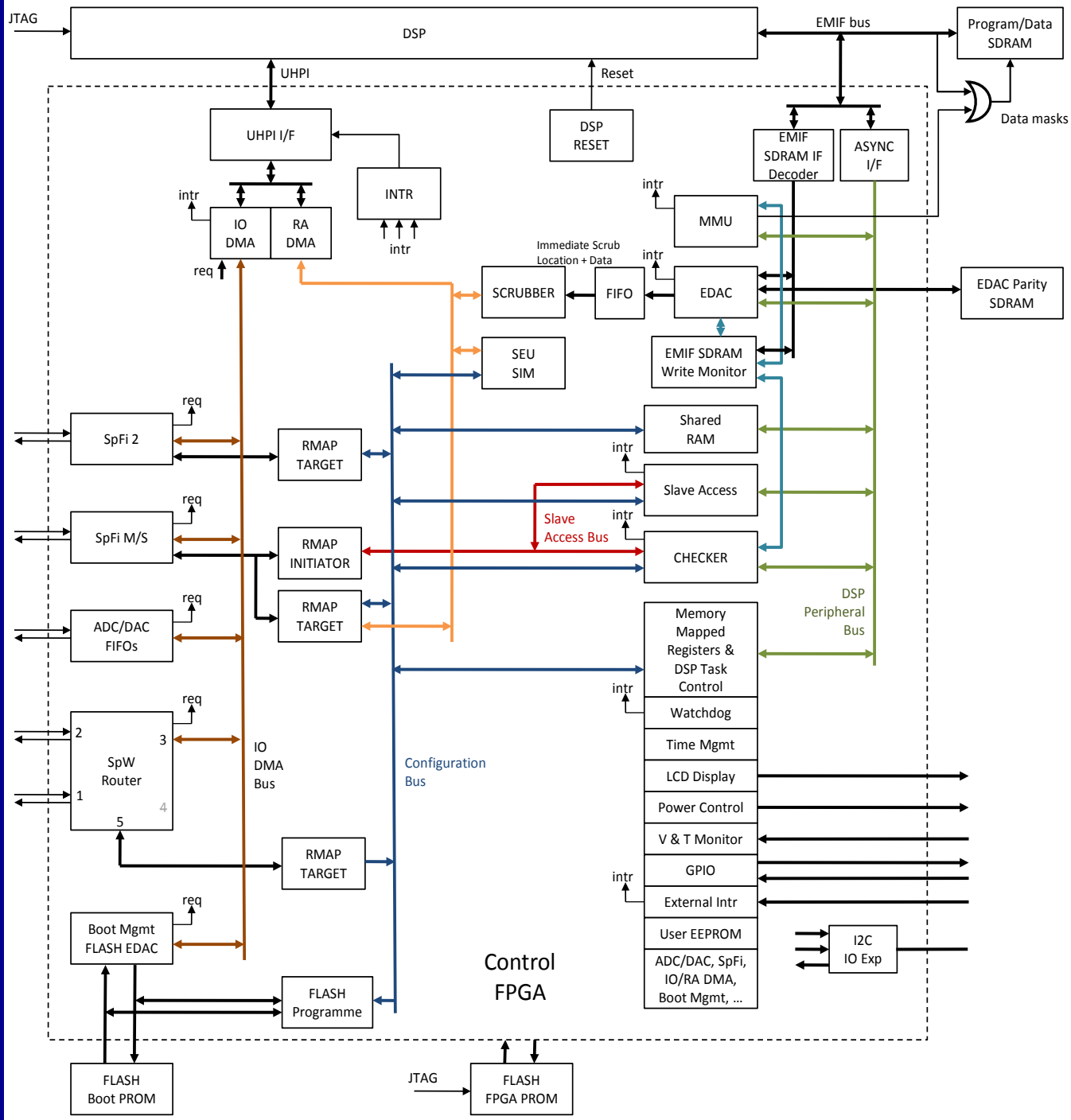


High Processing Power DSP



STAR-Dundee

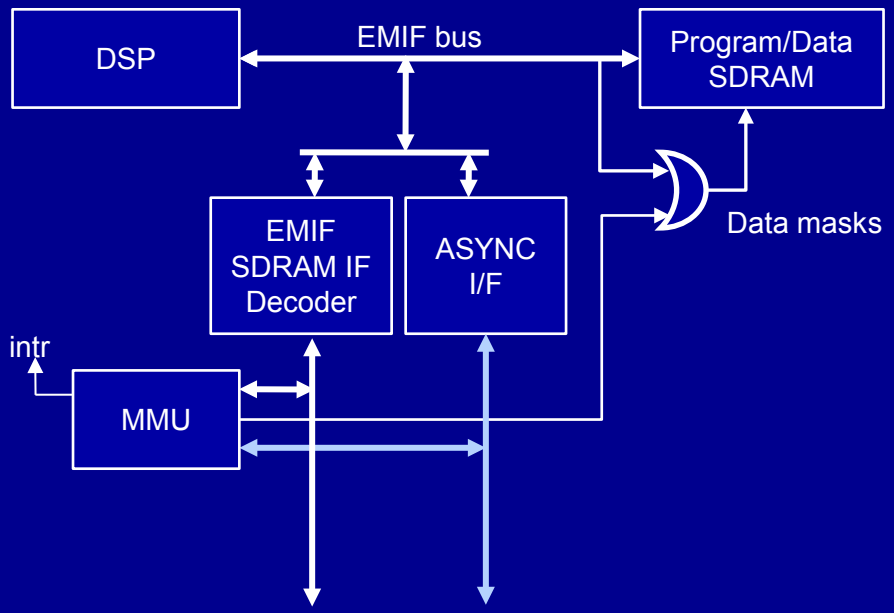
Block Diagram of the FPGA Design



EMIF bus

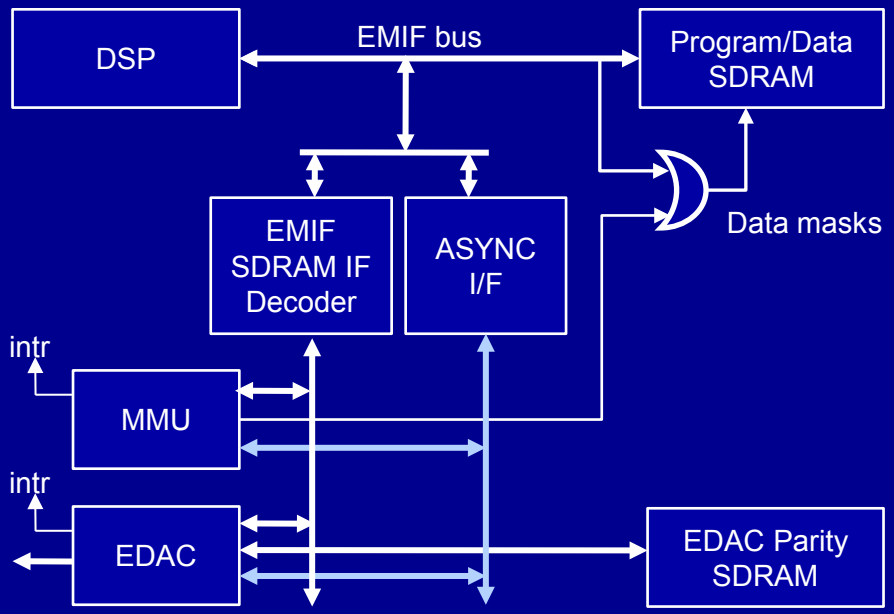
- External Memory Interface bus
- 32-bits wide
- Used to connect external SDRAM
- No MMU
- No EDAC
- No BusREQ / BusGNT signals
  - Only the DSP can be EMIF bus master!

Memory Management Unit (MMU)



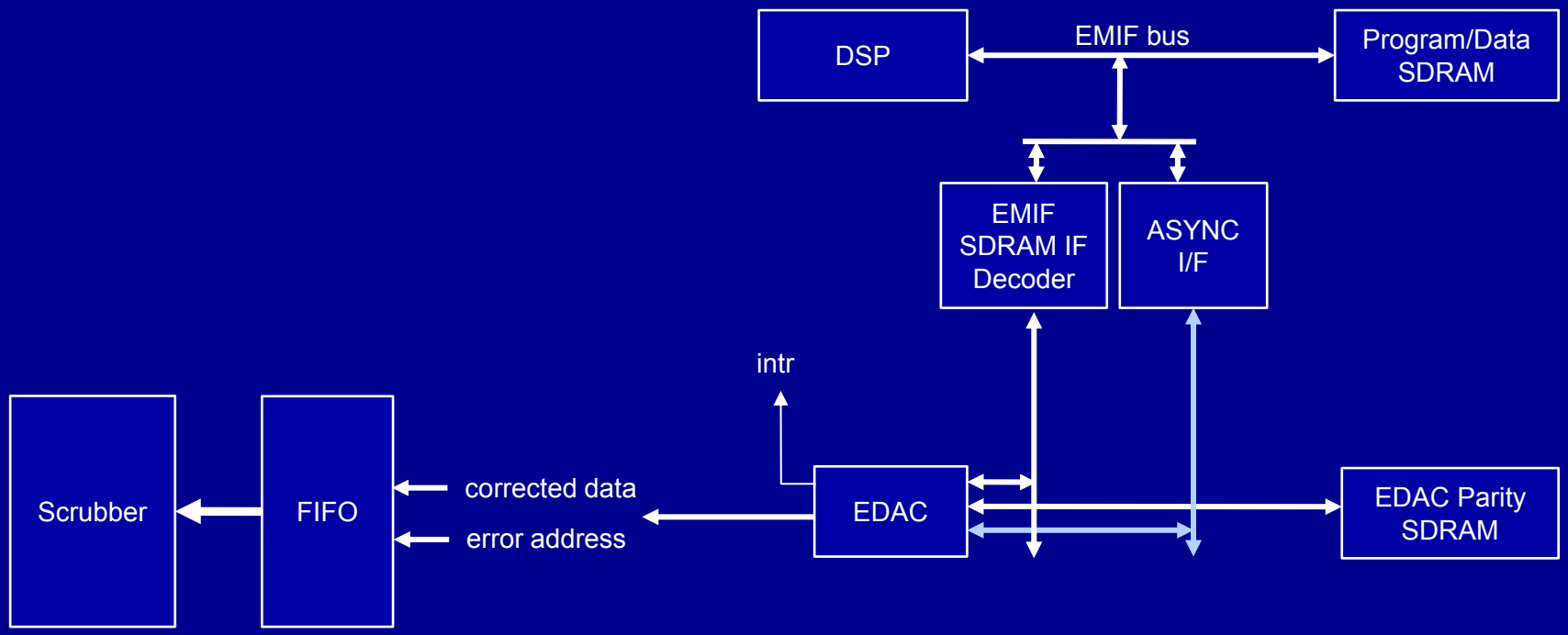
- Prevents inadvertent write to SDRAM, by turning on data mask
- External OR gates protect Program Data SDRAM
- Internal OR gates protect Parity SDRAM
- Area protected defined by registers
- When an MMU error occurs
  - Interrupt to DSP processor
  - Blocked state is cleared by writing to a control register

EDAC Design



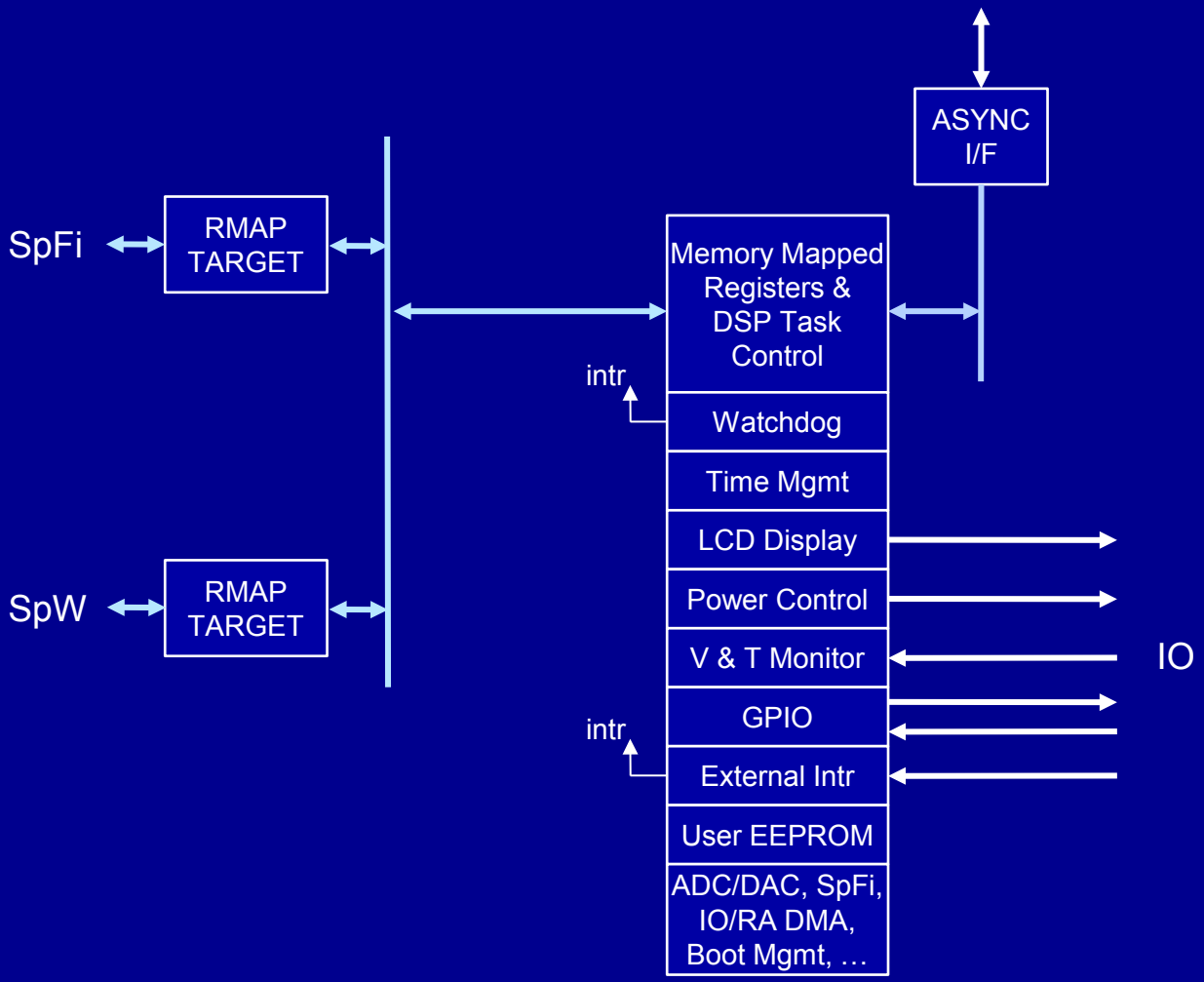
- Hamming code (39, 32)

EDAC Design



- When an error detected
  - Interrupt generated
  - Address of error and corrected data put in FIFO
  - Scrubber reads FIFO and updates memory to correct error

IO on the EMIF Async Bus

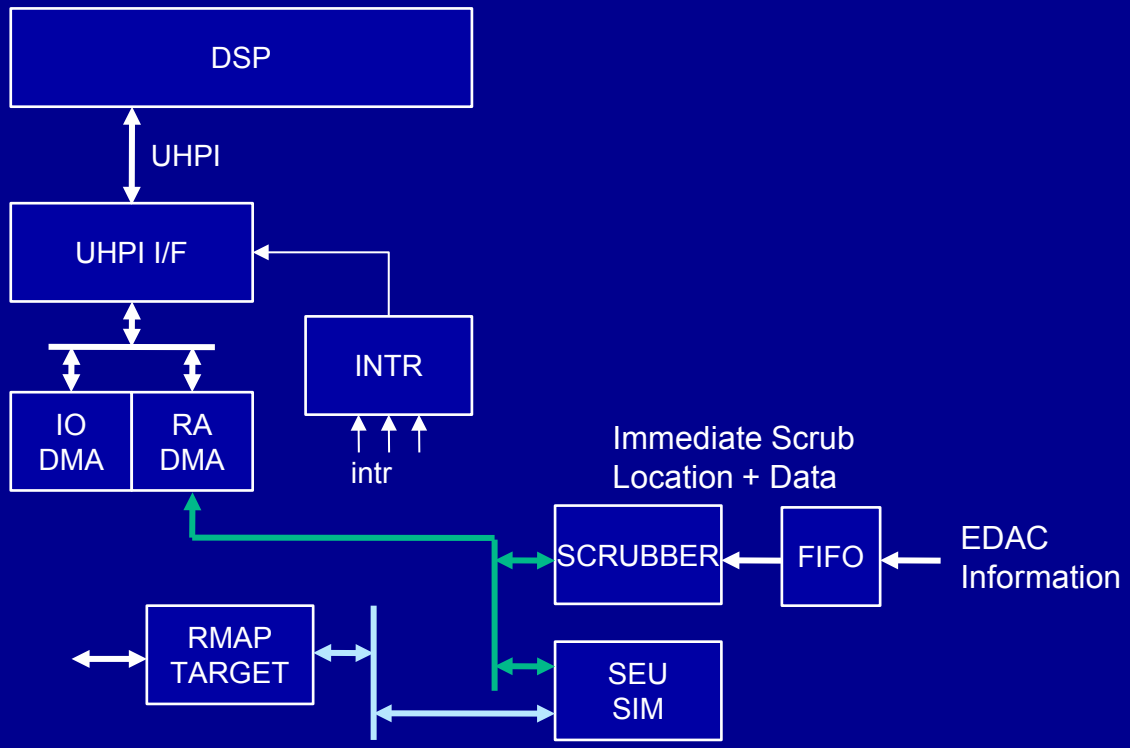


- Registers and IO can be accessed by EMIF
- Or SpW/SpFi via RMAP Targets

UHPI Interface

- Universal Host Peripheral Interface
- FPGA acts as a host to the DSP processor
- Enables access to DSP memory
- DSP boots from UHPI interface, boot-up settings configured
- UHPI interface clock is 125 MHz
  - Data throughput of around 500 Mbits/s max
  - Due to architecture of UHPI interface in DSP chip
- UHPI
  - Random access to memory
    - RA DMA
  - IO access to memory
    - IO DMA

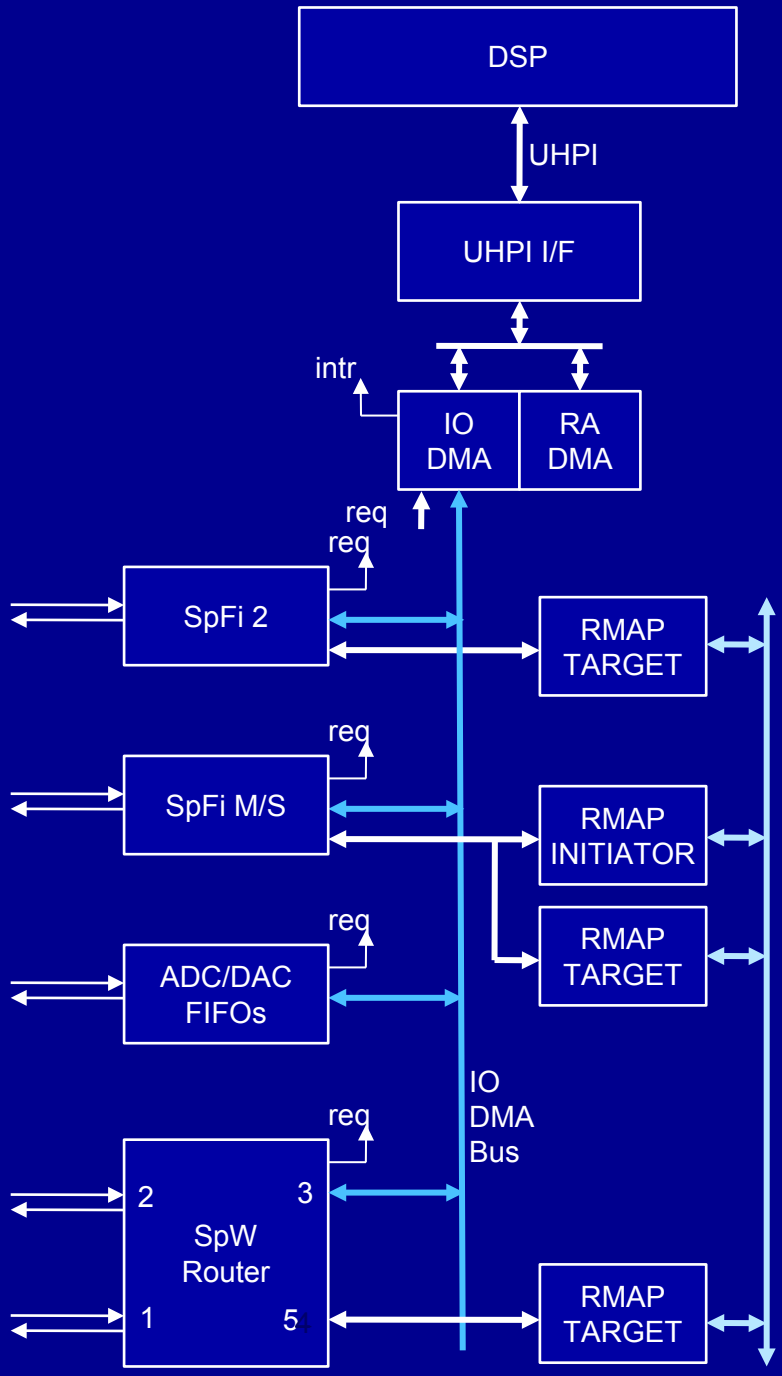
UHPI Interface and Random Access DMA



- Random Access DMA
  - Scrubber
    - Correcting memory
  - SEU simulator
    - Corrupting memory

IO DMA

- DMA data between DSP memory and:
  - SpaceFibre
  - SpaceWire
  - ADC/DAC
- Different operations for Normal, Master, and Slave mode
- When a IO DMA is completed, an interrupt is generated
- IO DMA and RA DMA compete for access to UHPI
- Boot management also goes through IO DMA

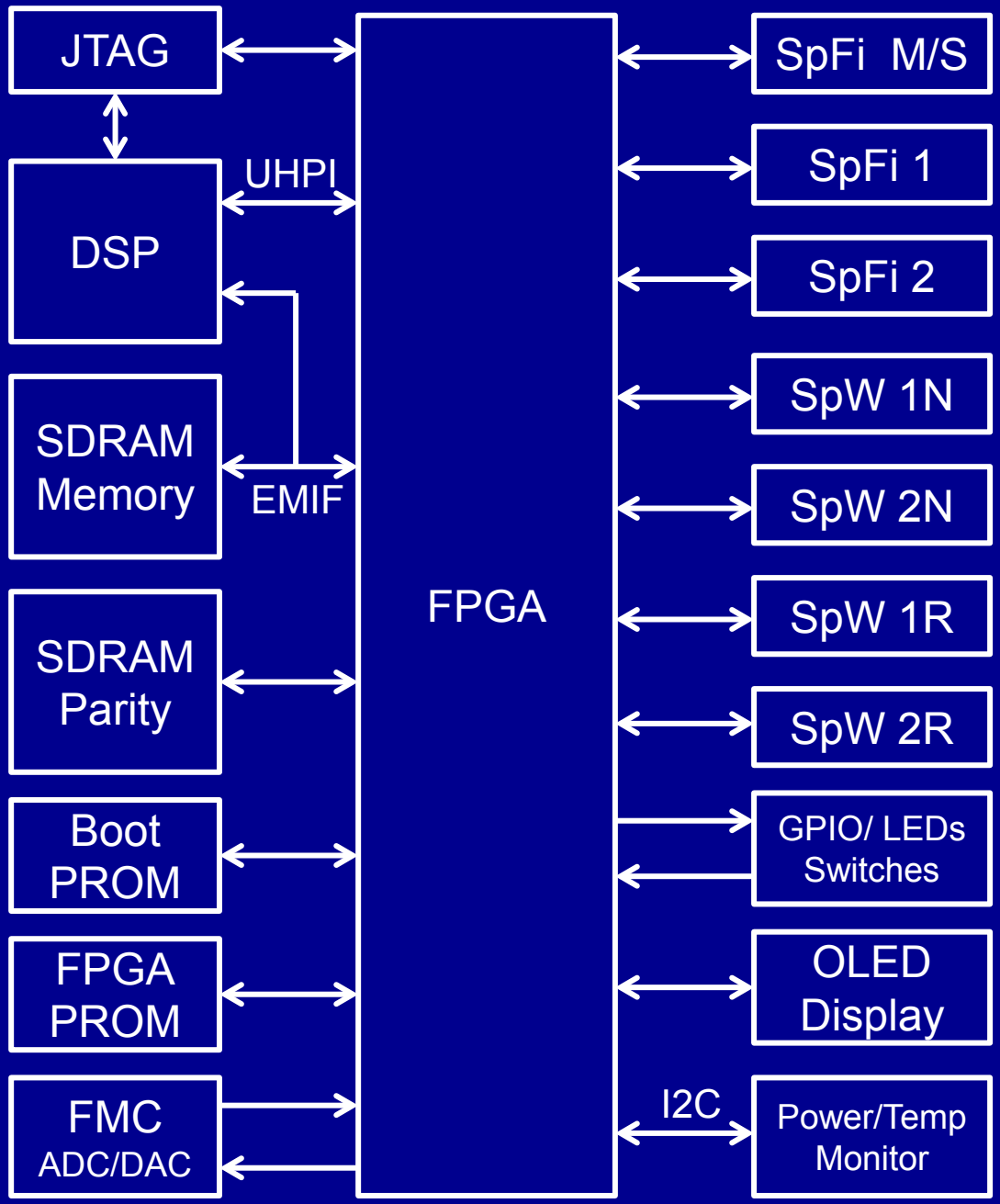


SpaceWire and SpaceFibre IO

- SpaceWire
  - Internal SpaceWire Router
    - 2 ports to external LVDS drivers/receivers
    - 2 ports to IO DMA bus
    - 1 port to RMAP Target
- SpaceFibre
  - Two interfaces each at 2.5 Gbits/s
  - Two different SerDes (Xilinx MGT and TI TLK2711)
  - Master/Slave
    - For connecting to second HPPDSP
    - 4 virtual channels
  - SpaceFibre 2
    - For high speed IO
    - 2 virtual channels

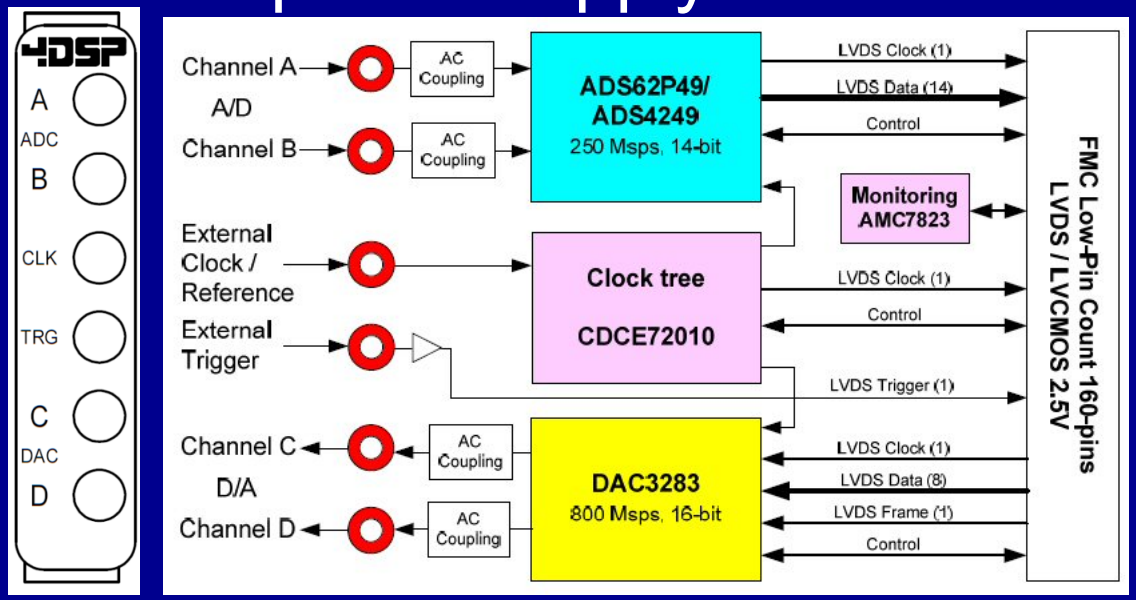
STAR-Dundee

HPPDSP Board

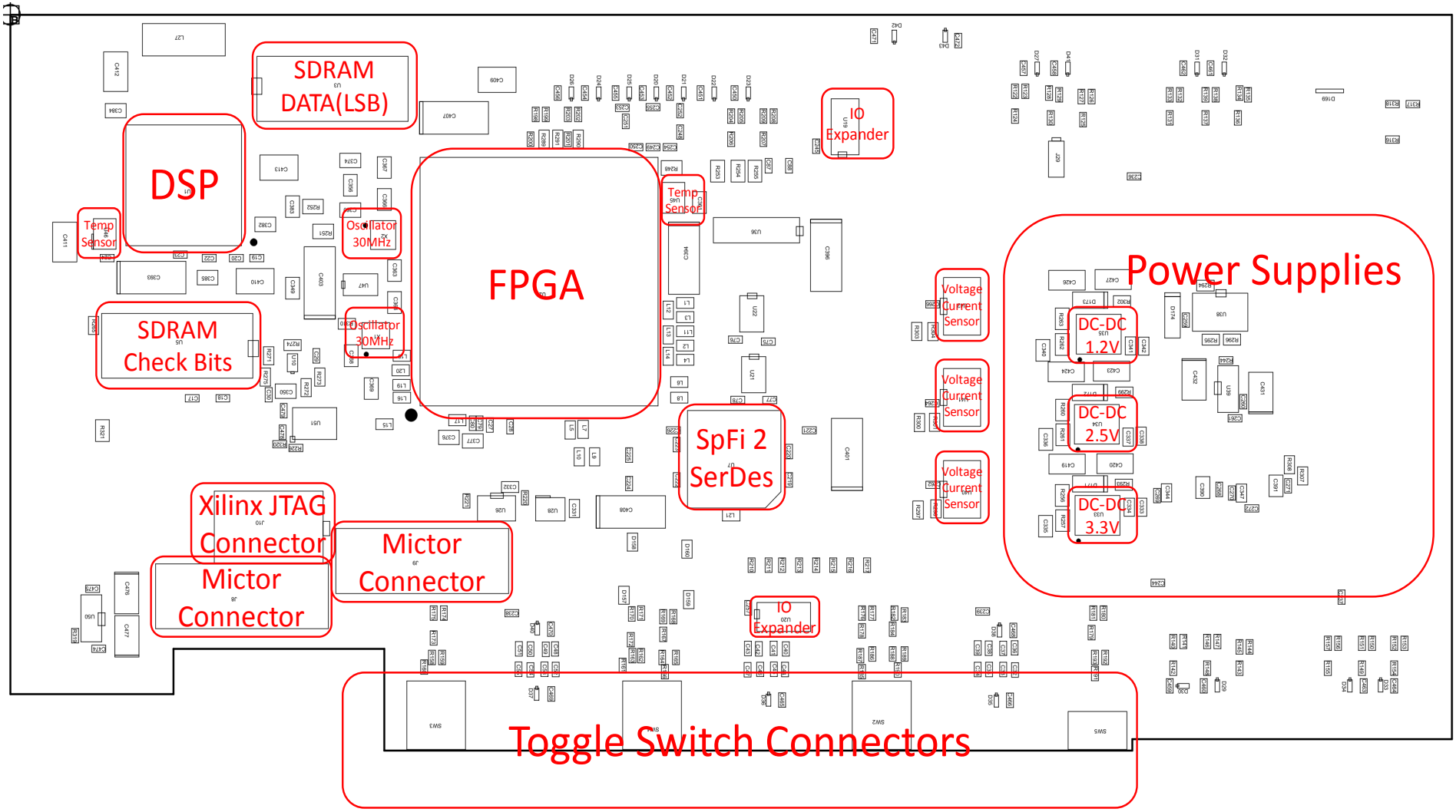


ADC/DAC Mezzanine Board

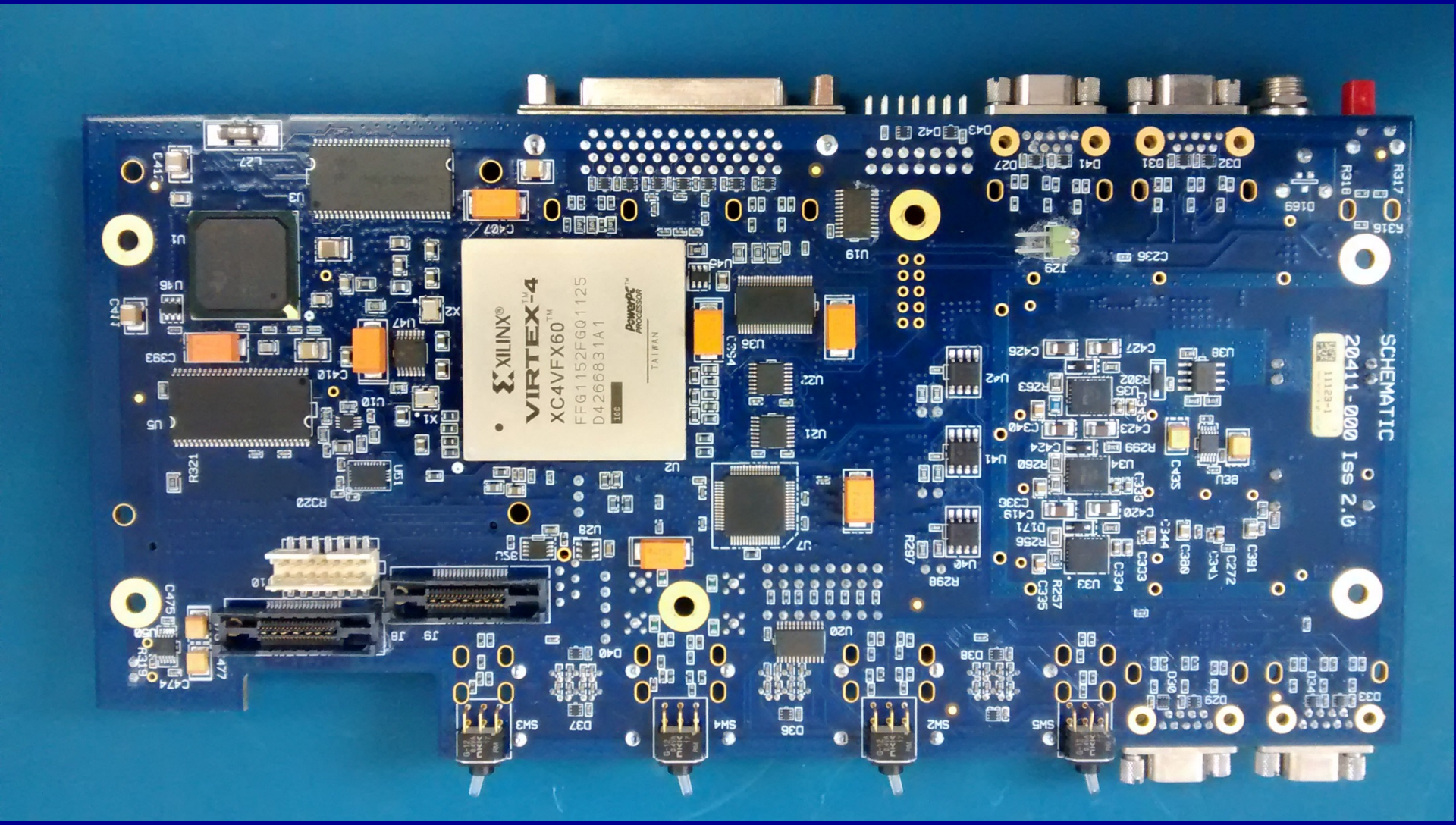
- FMC150 from 4DSP
  - 14-bit ADC and 16-bit DAC
    - 163.84Msps ADC, and 491.52Msps DAC with 4x interpolation
  - Internal clock, or optional external clock source
  - Software trigger, or optional external trigger
  - FMC (FPGA Mezzanine Card) interface to FPGA
- Need additional power supply – 12V



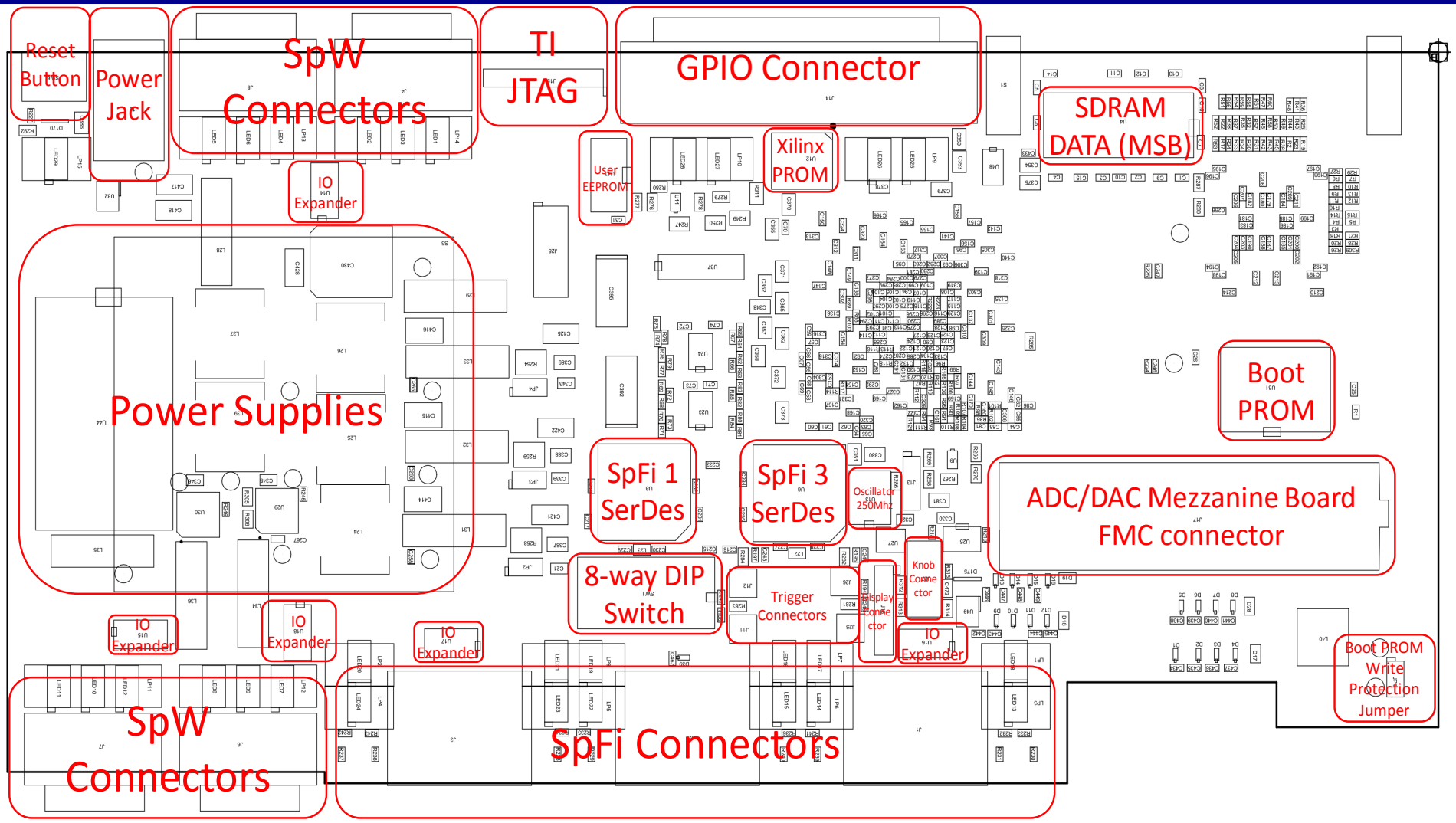
PCB Layout – Bottom Side



HPPDSP – Bottom Side

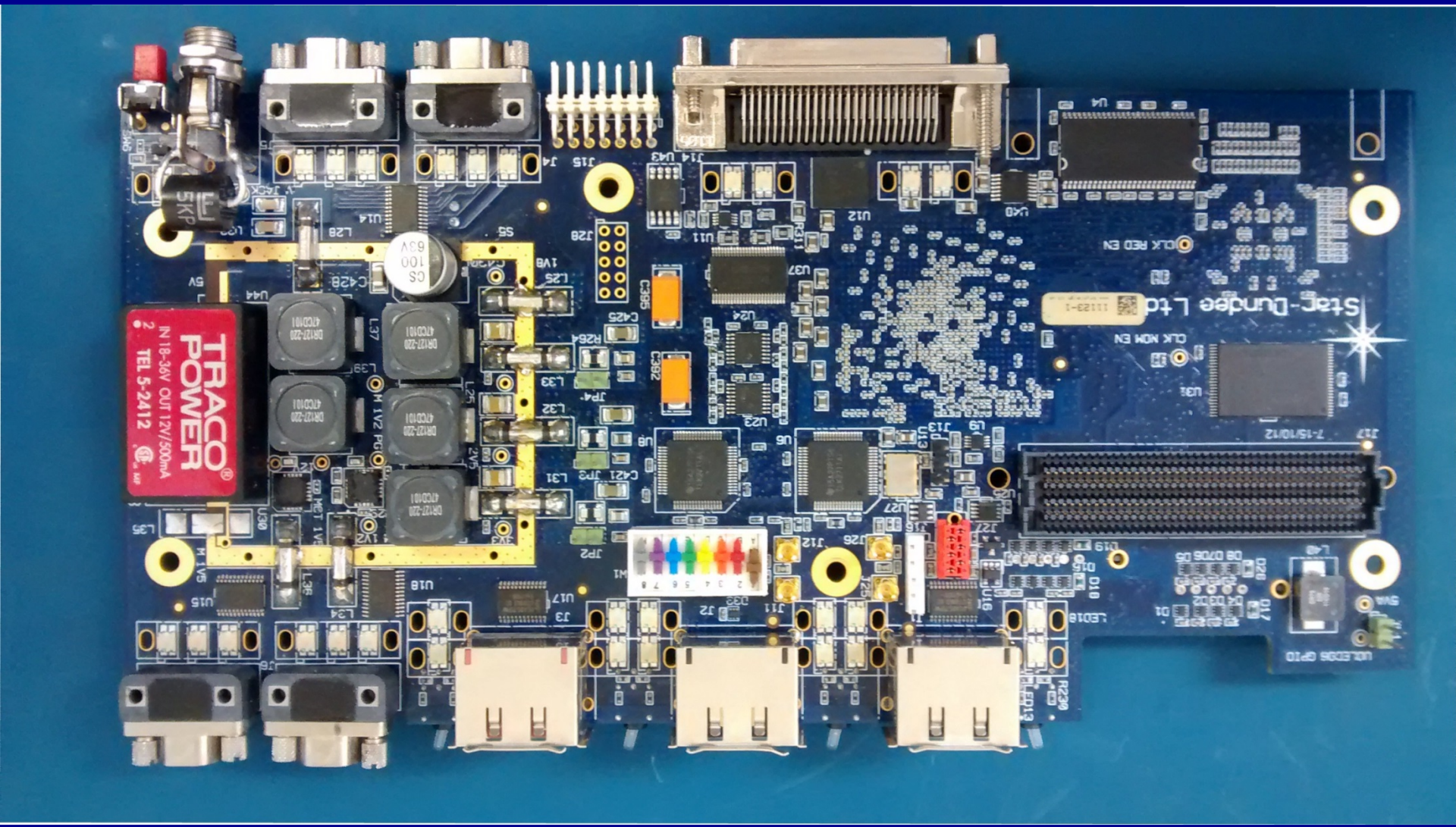


PCB Layout – Top Side

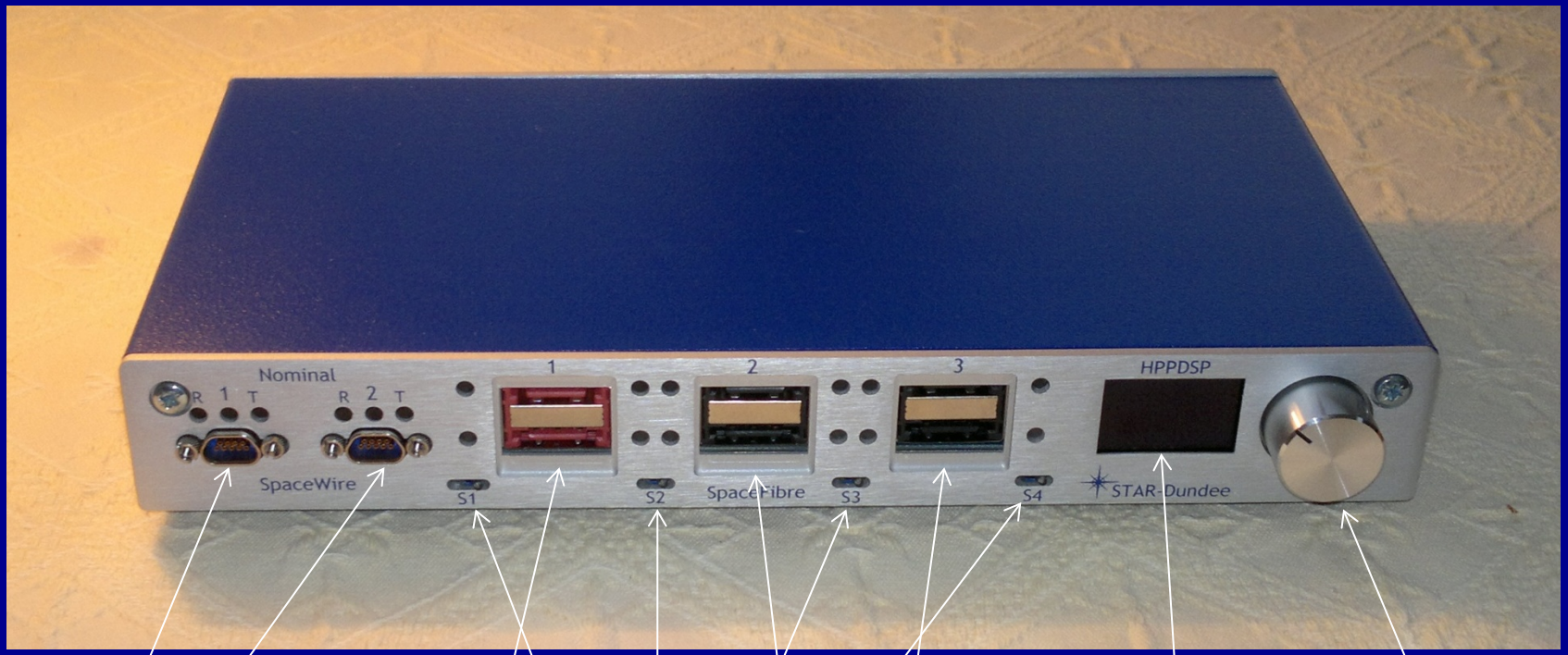


STAR-Dundee

HPPDSP – Top Side



HPPDSP Front Panel



SpaceWire Nominal Ports 1 & 2

SpaceFibre M/S

Switches

SpaceFibre

Display

Push Button/ Knob

HPPDSP Rear Panel

ADC/DAC
FMC Board

Triggers

GPIO

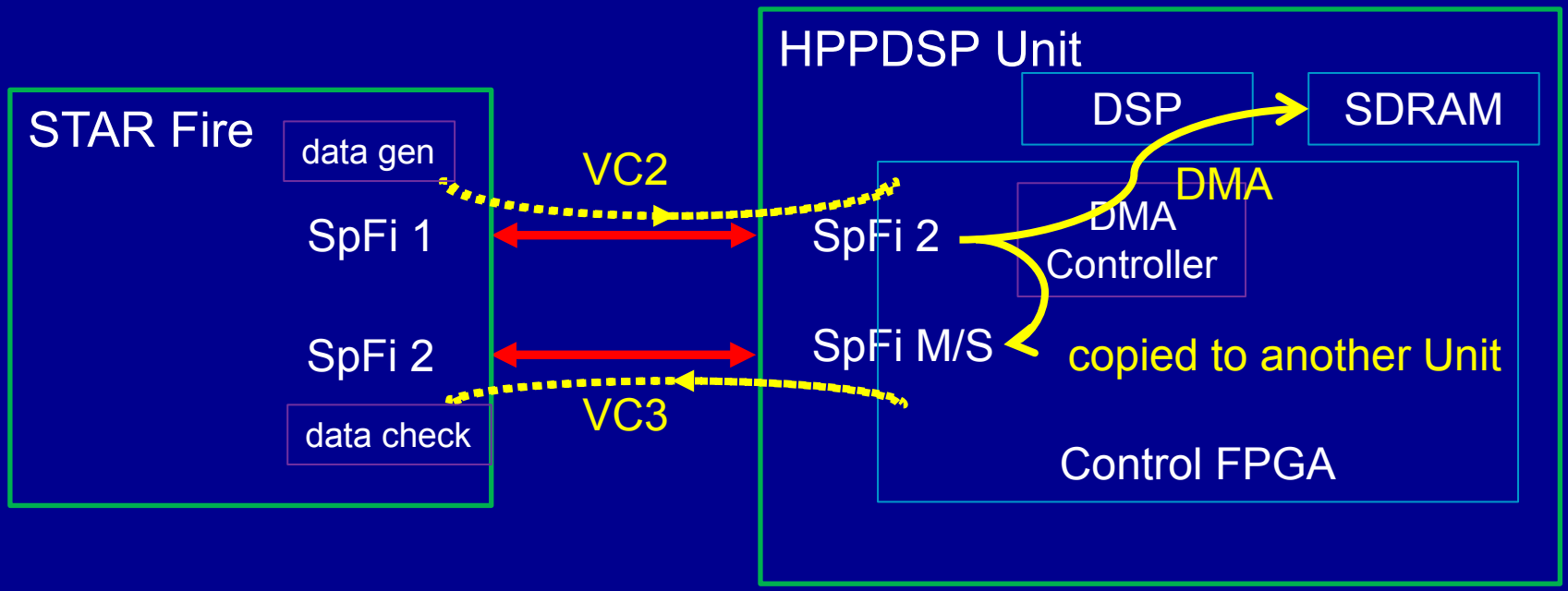
DSP JTAG

SpaceWire
Redundant
Ports 1 & 2

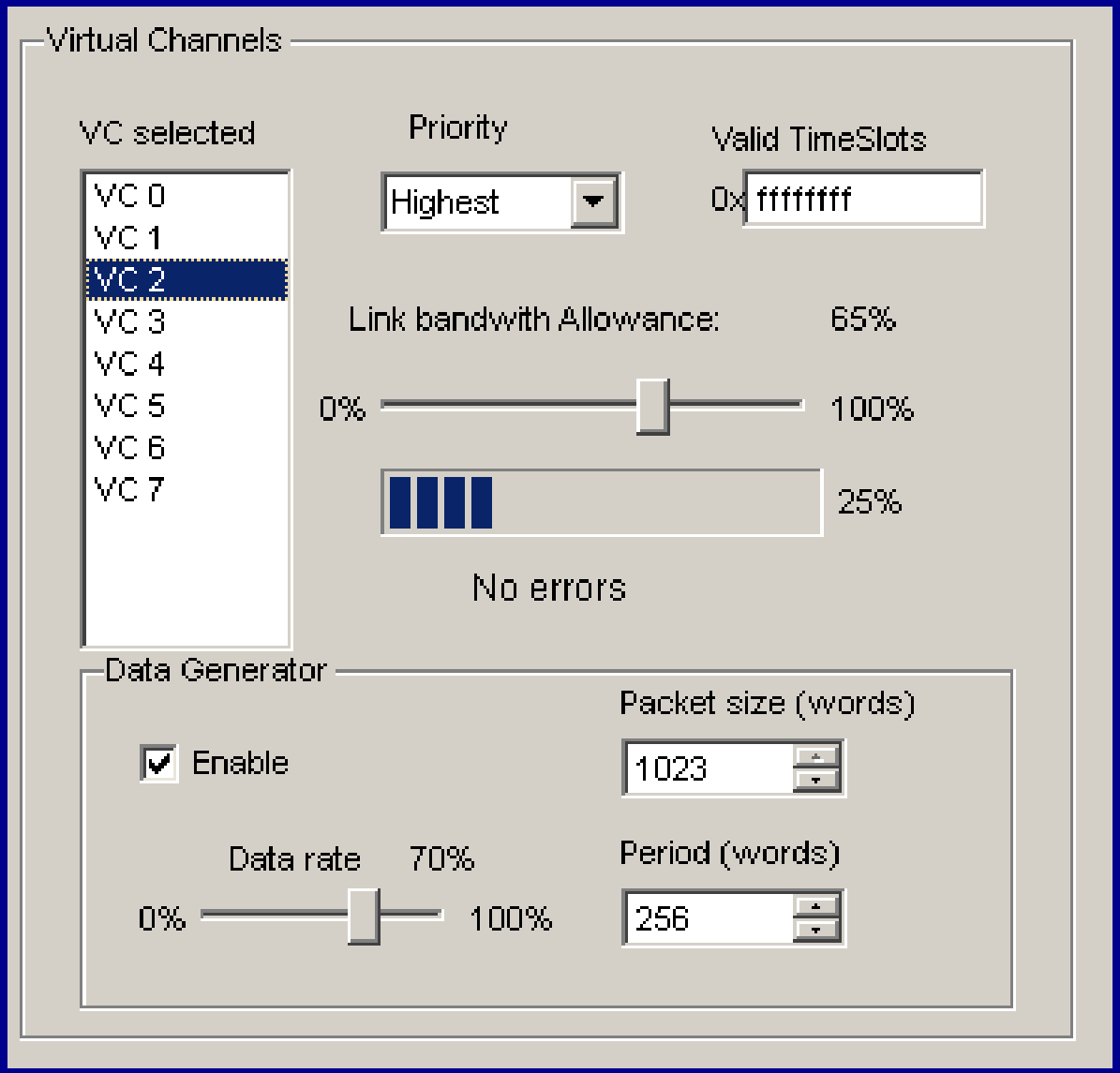
Power

Reset

SpaceFibre Test

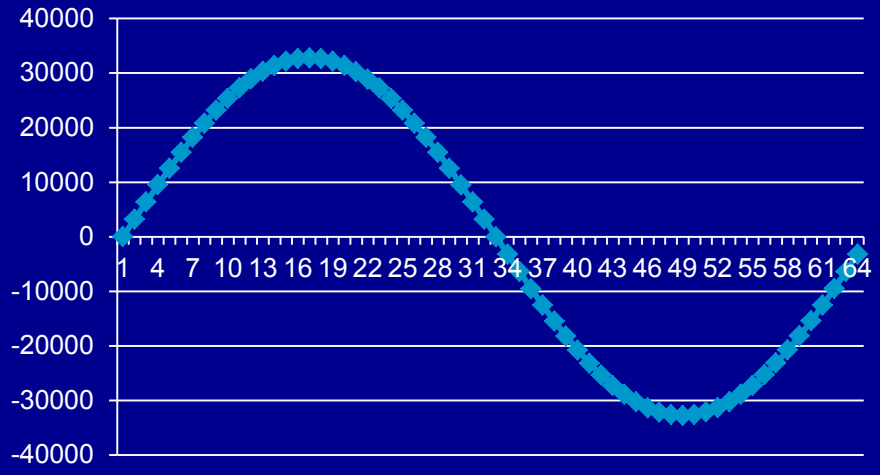


SpaceFibre Test

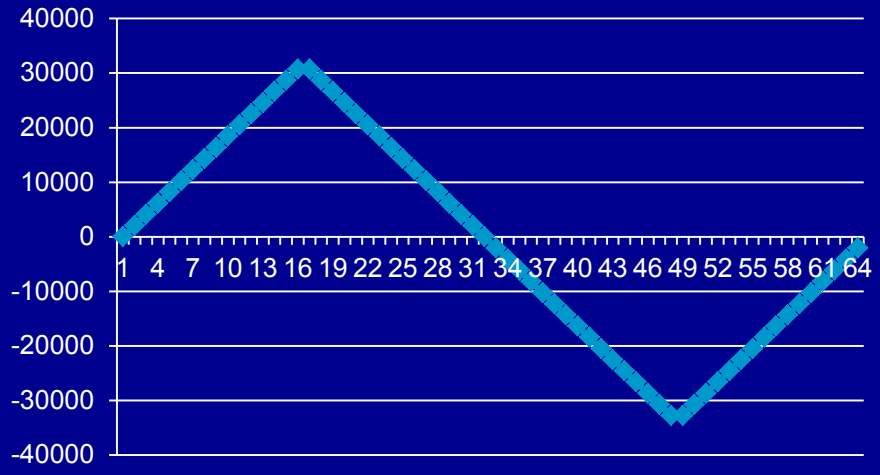


ADC/DAC Test

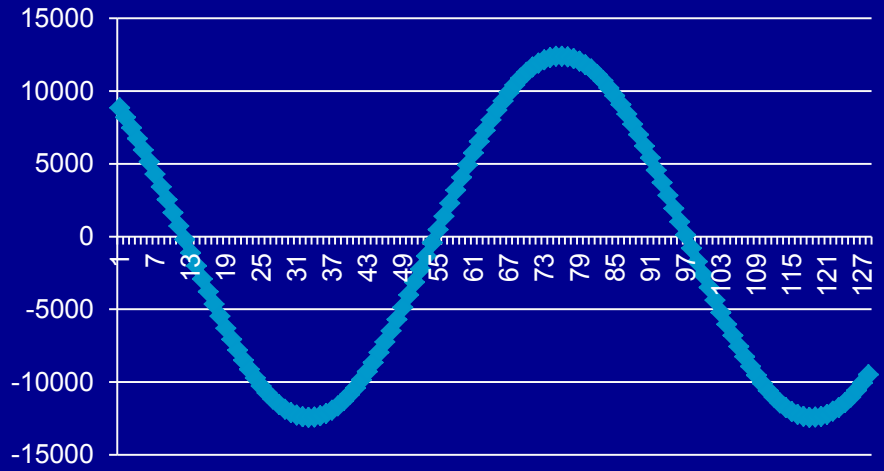
DAC Output Waveform



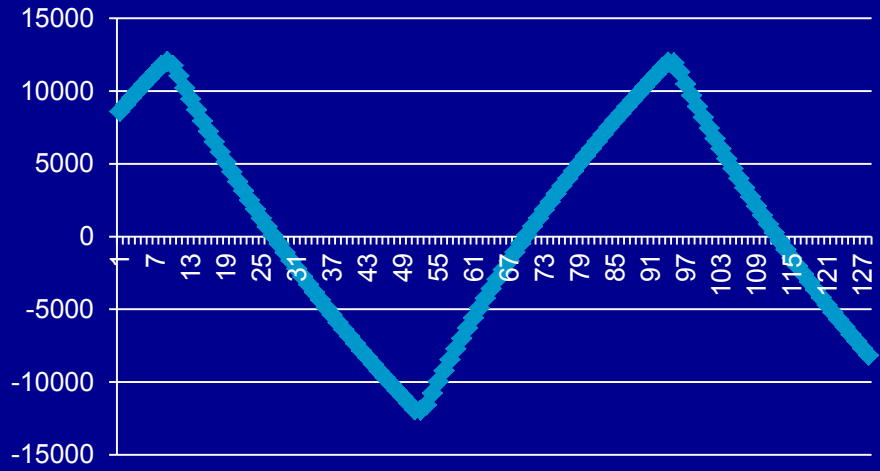
DAC Output Waveform



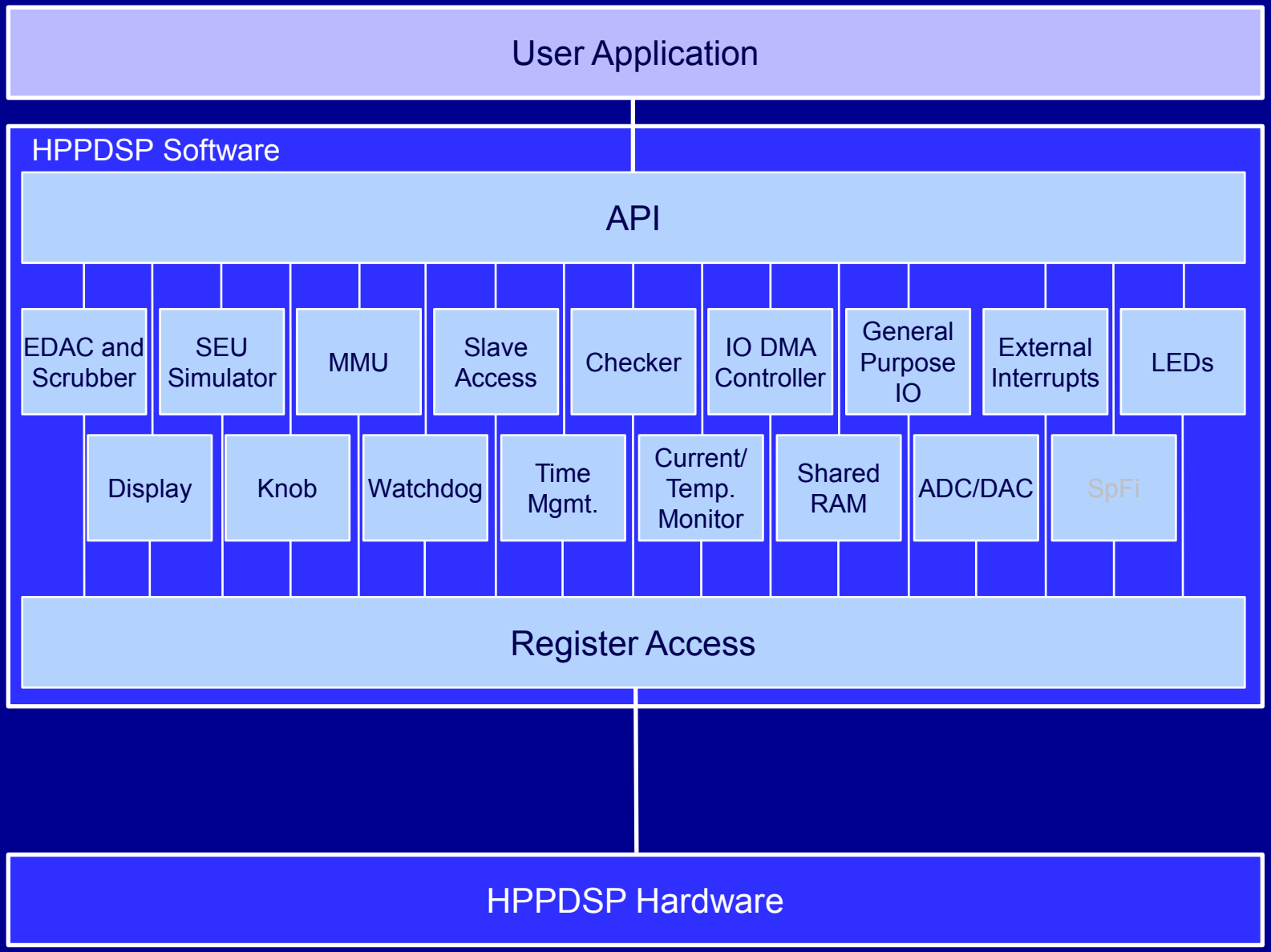
ADC Data Plot

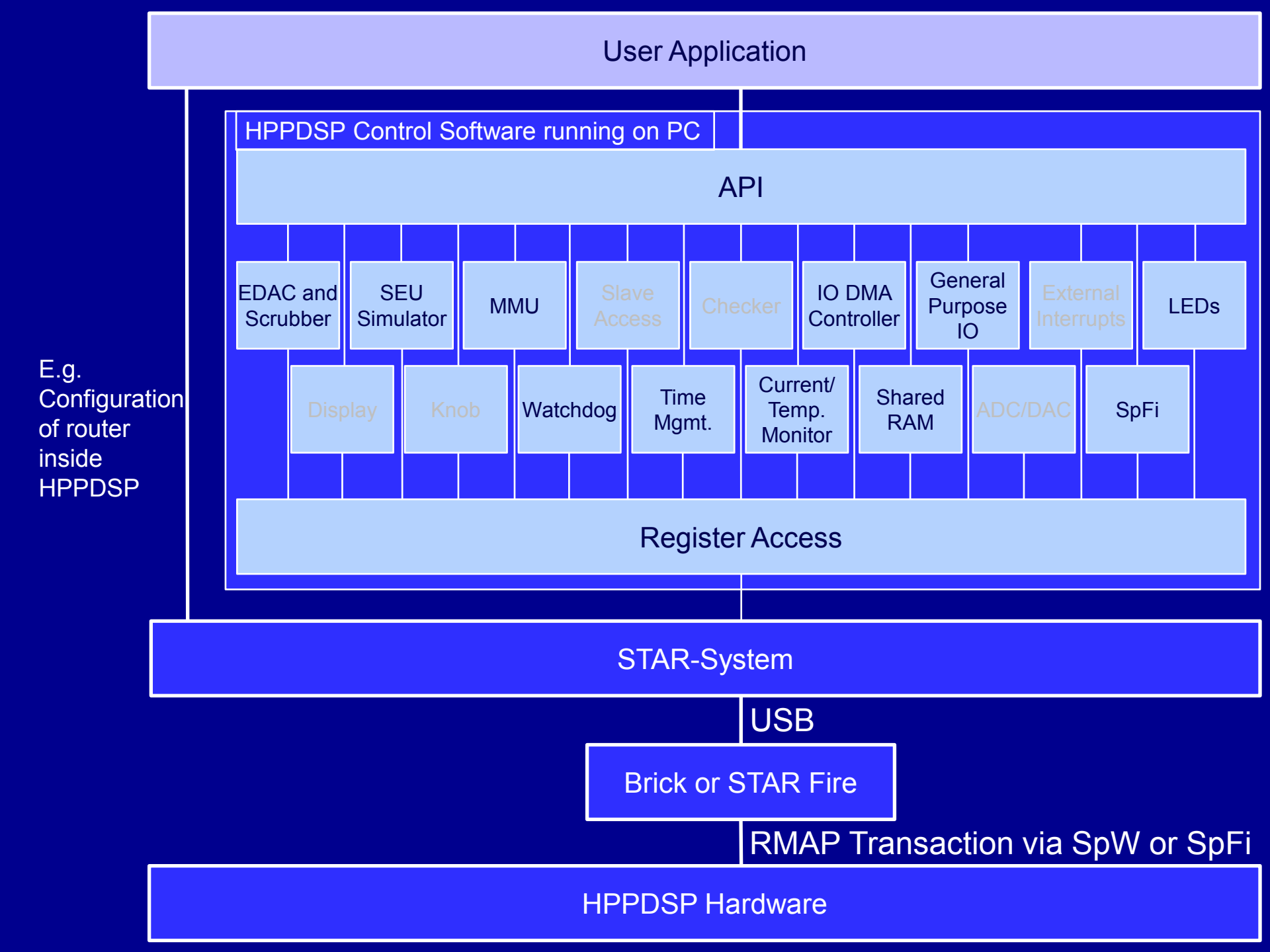


ADC Data Plot



Basic Software




User Application
HPPDSP Control Software running on PC
API
EDAC and Scrubber
SEU Simulator
MMU
Slave Access
Checker
IO DMA Controller
General Purpose IO
External Interrupts
LEDs
Display
Knob
Watchdog
Time Mgmt.
Current/ Temp. Monitor
Shared RAM
ADC/DAC
SpFi
Register Access
E.g. Configuration of router inside HPPDSP
STAR-System
USB
Brick or STAR Fire
RMAP Transaction via SpW or SpFi
HPPDSP Hardware

Conclusion

- Status
  - Board/Box designed and tested
  - FPGA designed and tested
  - Software Architecture designed
  - Software API being coded and tested
- Lessons learnt
  - BusREQ/BusGNT would be very useful
    - Unless the DSP has EDAC etc. built in
  - Difficult to add EDAC/MMU etc.
    - We did it and it works, but it is not simple!
  - UHPI did not give high enough data rate
    - 500 Mbits/s achieved