CORE | Vision
NEXT LEVEL | EMBEDDED DEVELOPMENT

Designing a dual-fuel motor management system with Zynq

Frank de Bont
Trainer / Consultant

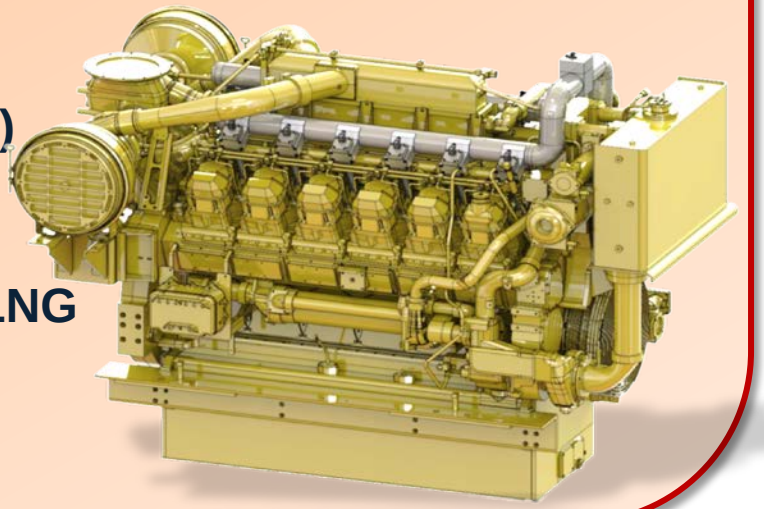
Cereslaan 10b
5384 VT Heesch
✆ +31 (0)412 660088
✉ info@core-vision.nl
www.core-vision.nl

Agenda

- Customer Requirements
- Engine Management System
- LNG Supply
- Sensors / Actuators
- AR Electronic Control Unit
- Zynq Architecture
- Design Flow and Tools
- High-Speed ADC IP-block
- Software Partitioning and Communication Channels
- Why Linux with Xenomai
- eMMC Configuration
- Who is Core|Vision

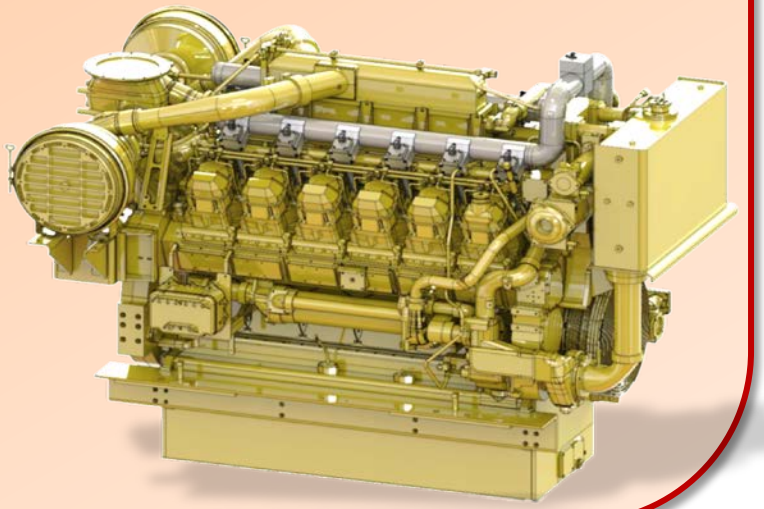
Customer Requirements

- **Design a dual-fuel motor management system with real-time control**
  - **Hardware Platform Caterpillar 3500**
    - **Diesel**
    - **LNG (Liquefied Natural Gas)**
- **High fuel efficiency**
  - **Start up on Diesel and then switch over to a mixture with LNG**
- **Lowest possible emissions**
  - **NO_x , SO_x , CO_2 and CH_4 (methaan)**
- **Easy to retrofit**
  - **Suitable for different qualities of LNG**
  - **Still 100% Diesel as a fall back**

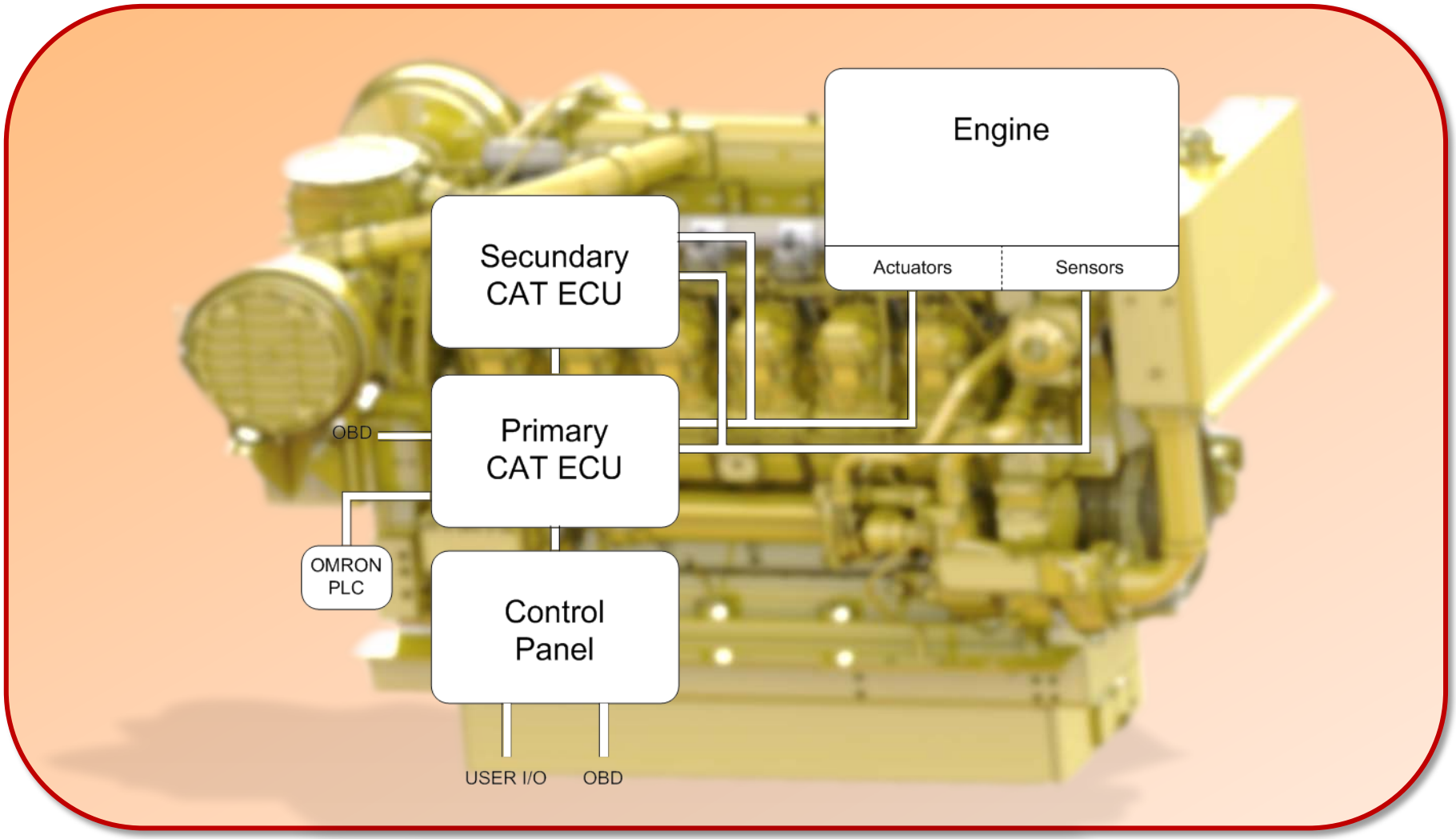


Customer Requirements cont

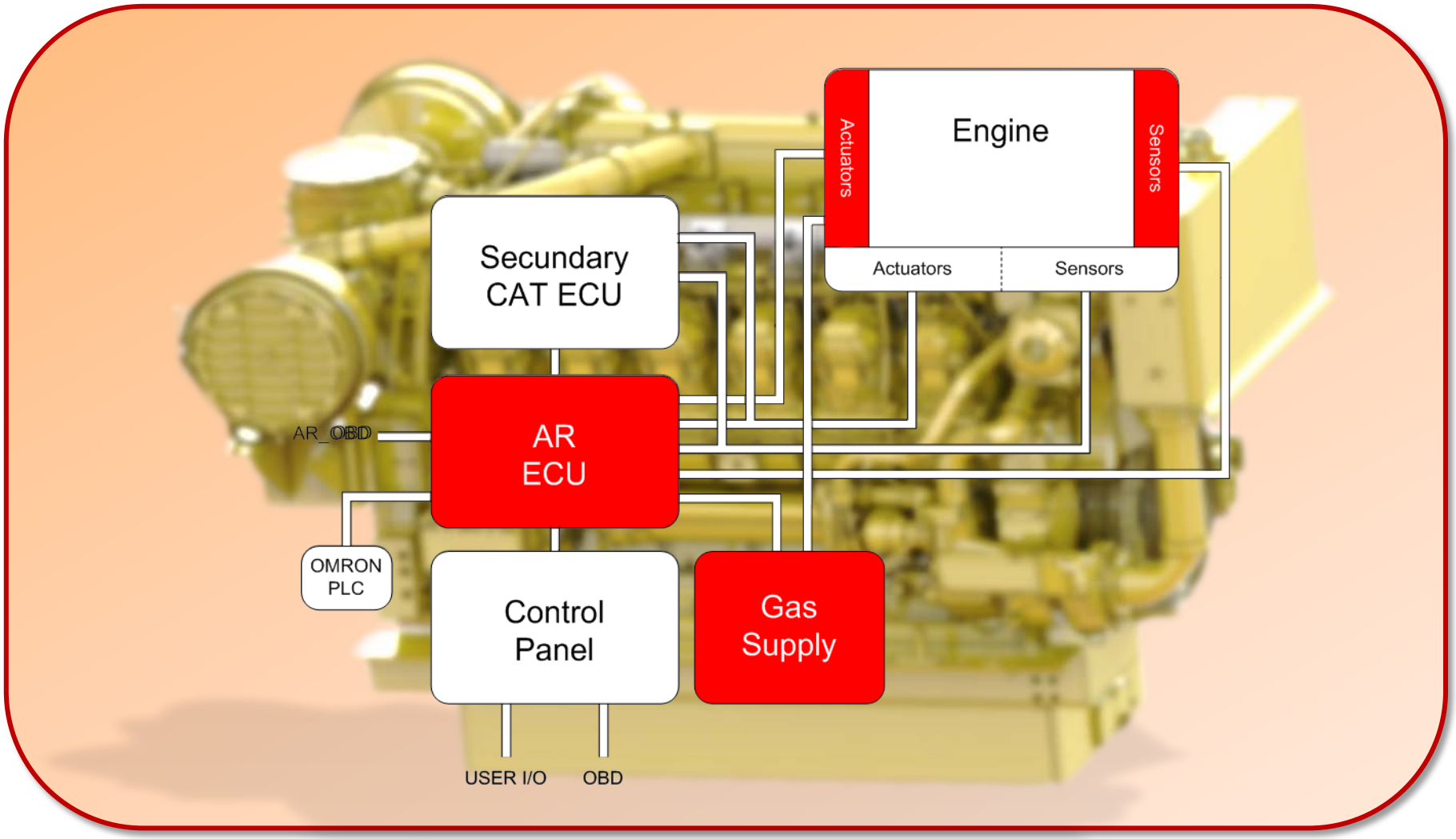
- **Focus on motor management control also called *ArenaRed* Electronic Control Unit**
  - **Hardware Platform Caterpillar 3500**
    - **Diesel**
    - **LNG (Liquefied Natural Gas)**



Engine Management System

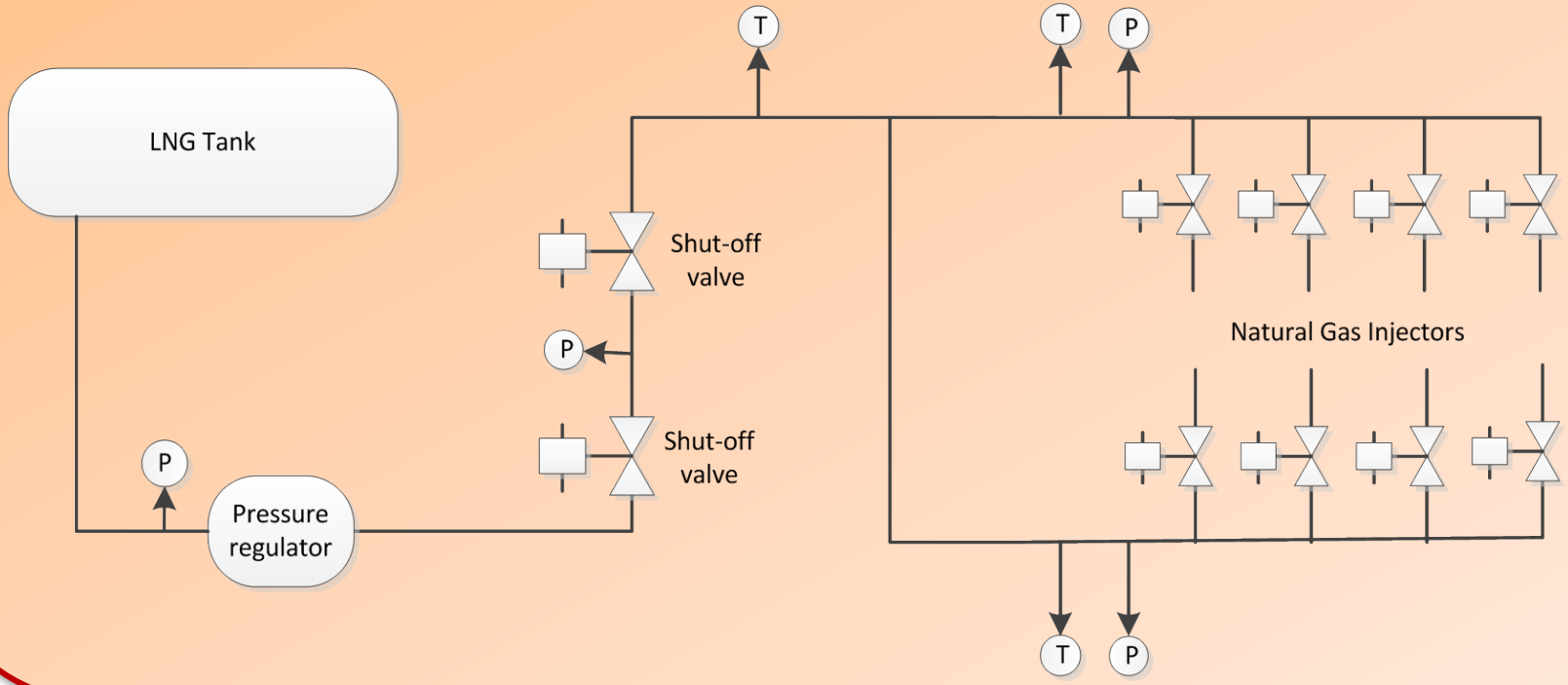


Engine Management System



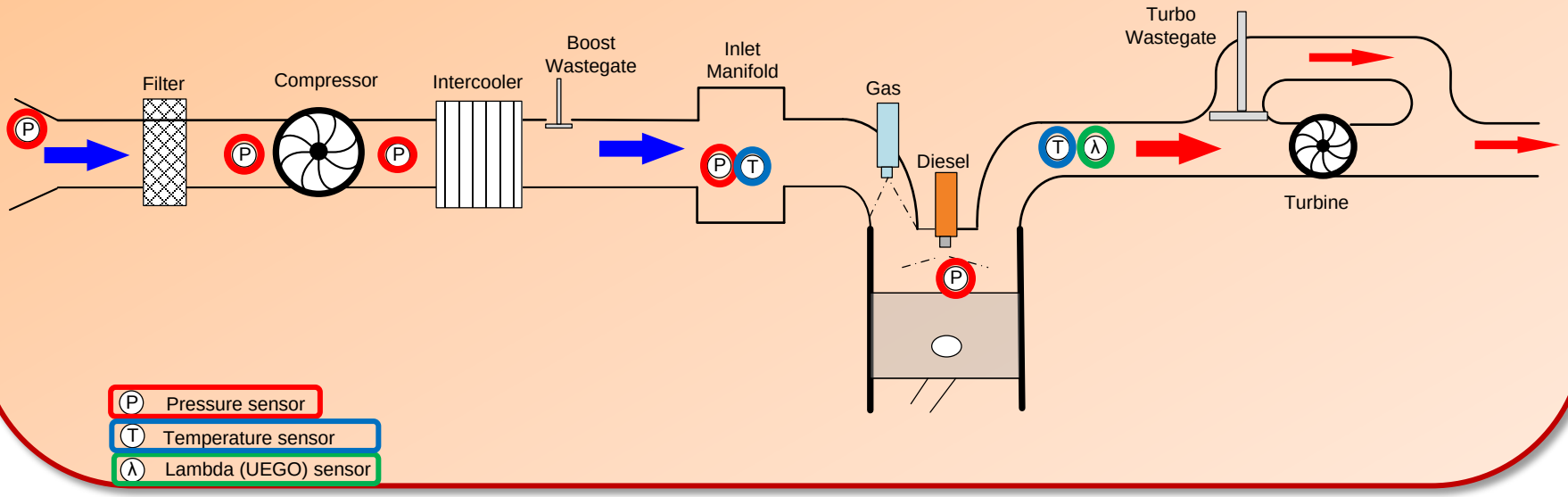
LNG Supply

- **Full controlled by the *AR*-ECU**
  - **A full fail-safe system**
  - **On-board diagnostics**
  - **Power and Temperature sensors**

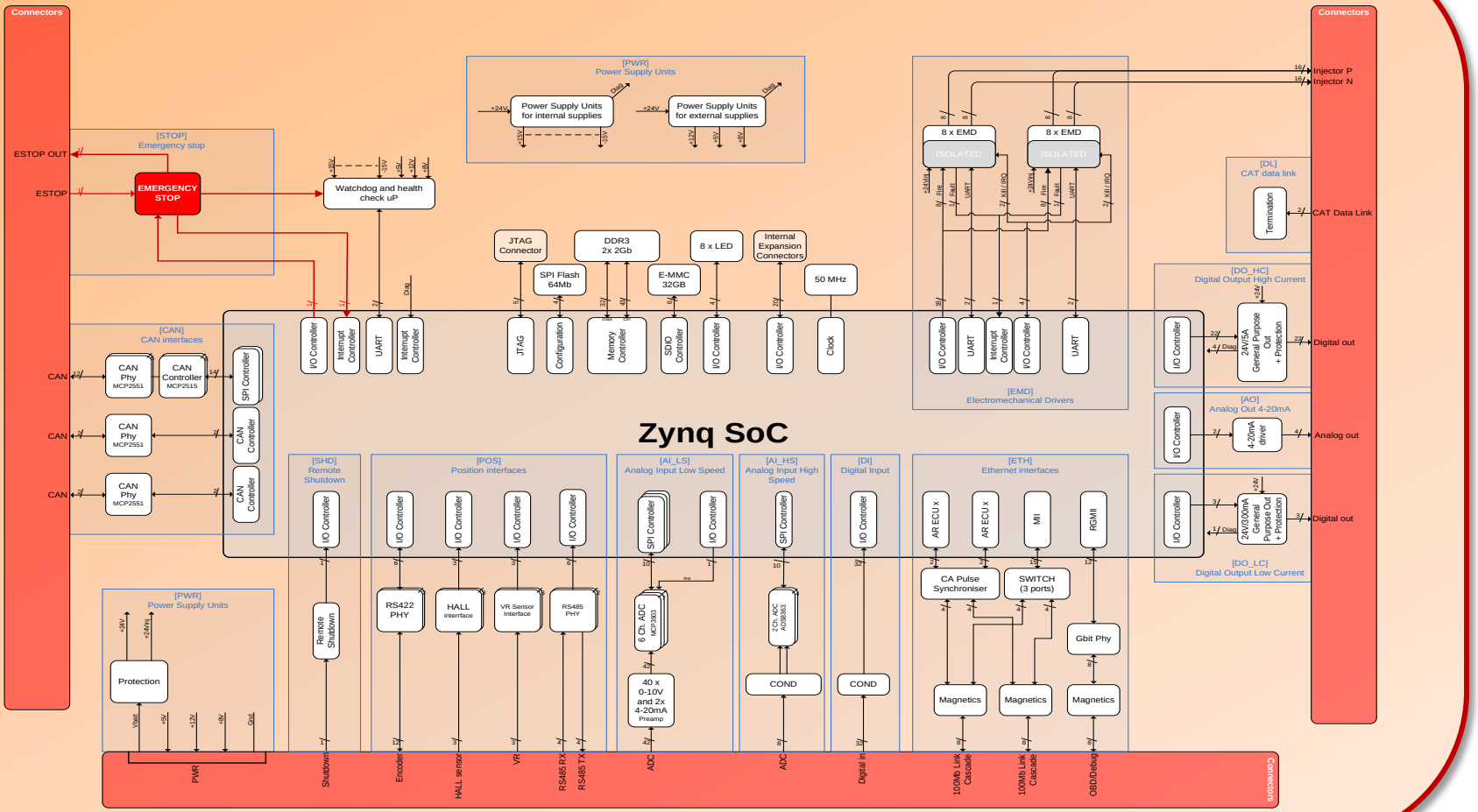


Sensors / Actuators

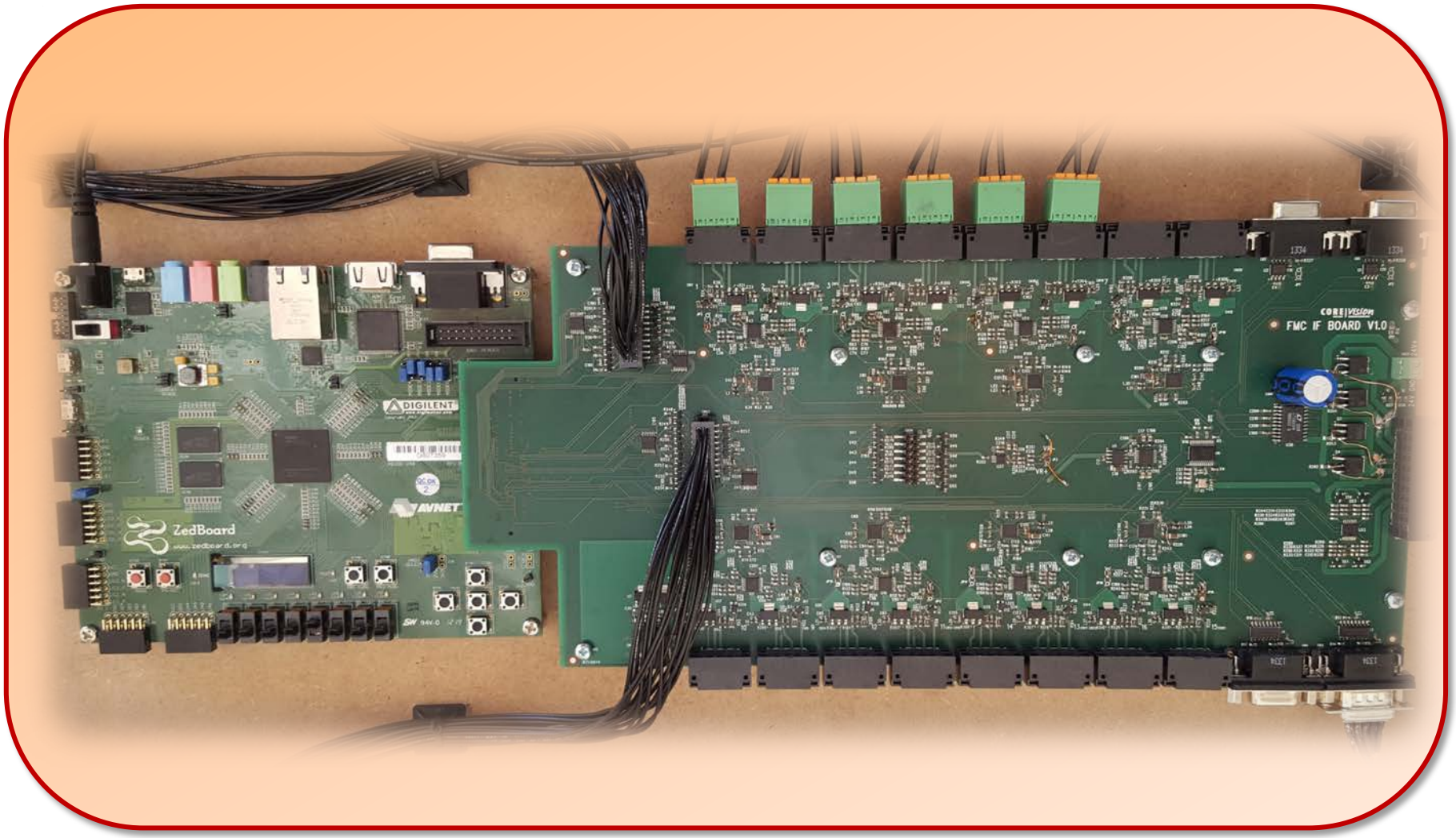
- Combustion in each cylinder is monitored by a set of cylinder pressure sensors
- These allows the *AR*-ECU to optimize the combustion timing and gas mixture
- In the exhaust are UEGO λ-sensors and temperature sensors to monitor the results of each combustion cycle



AR Electronic Control Unit

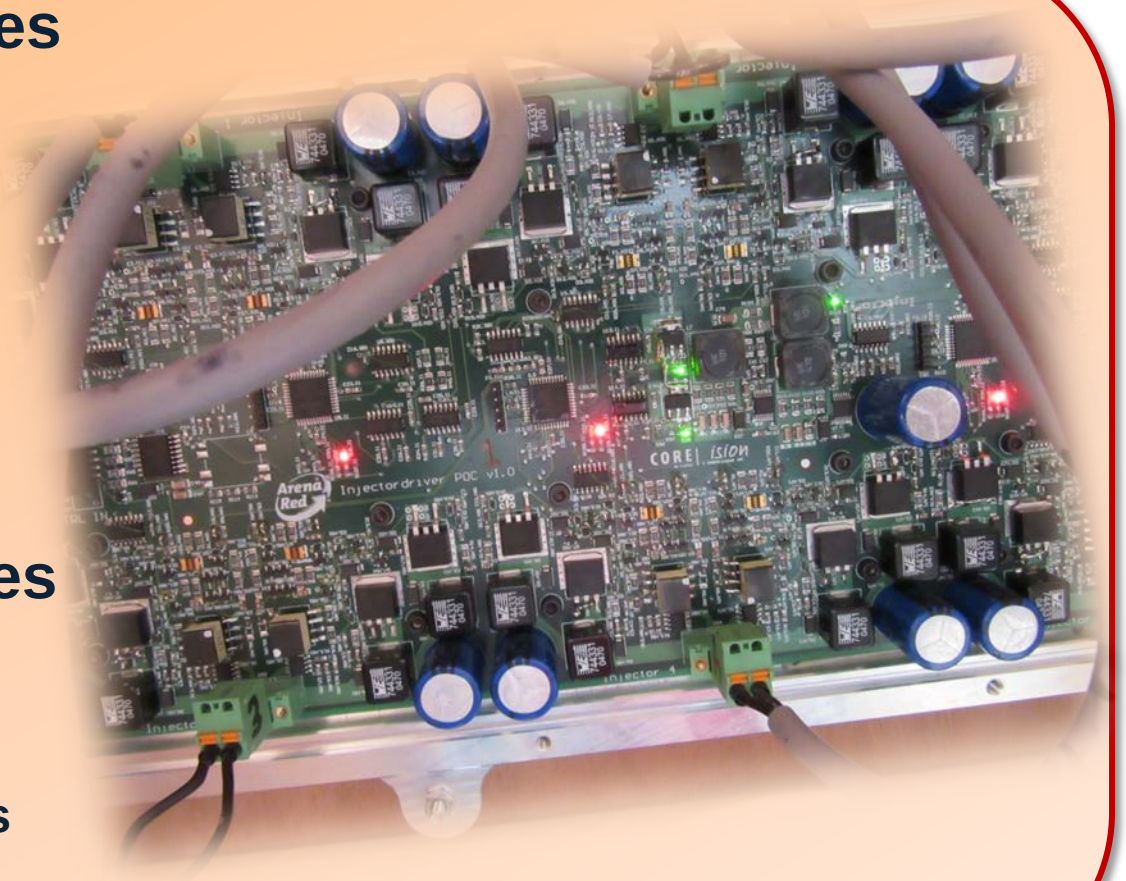


AR Electronic Control Unit



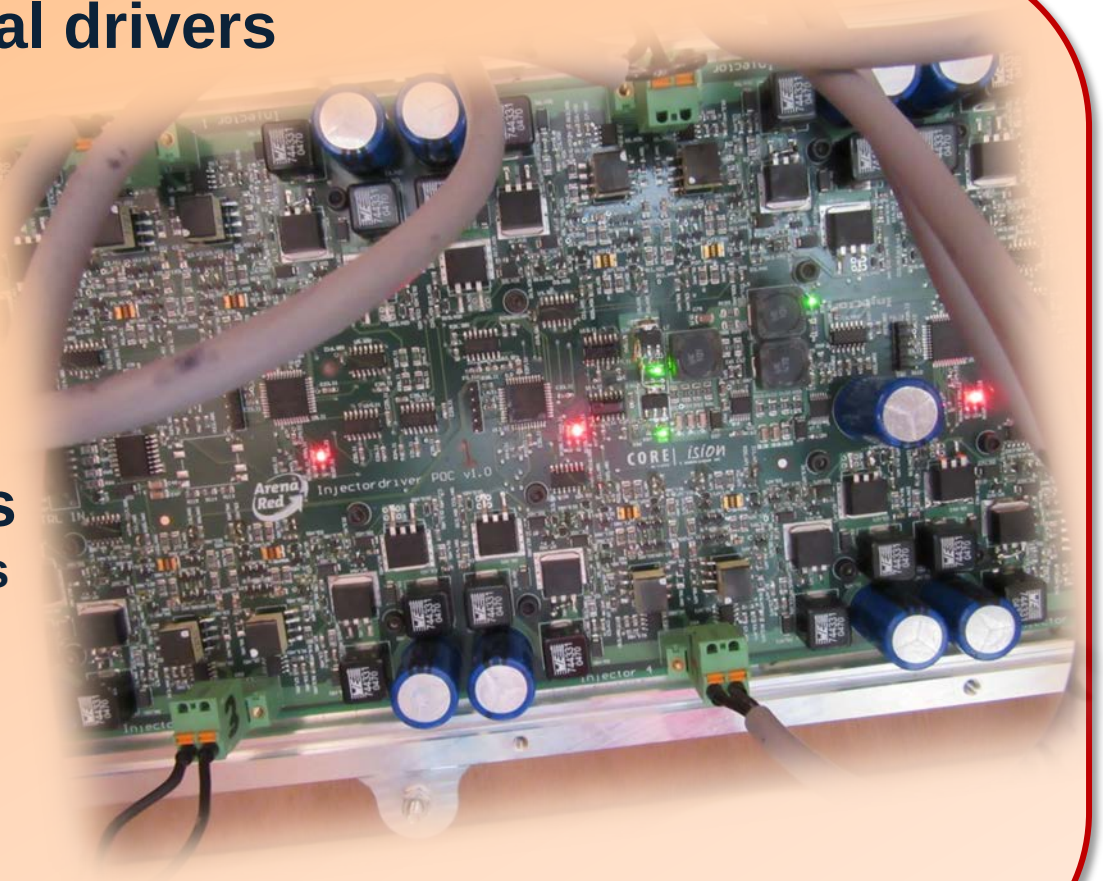
AR Electronic Control Unit cont

- **General Interfaces**
  - DDR3 memory
  - eMMC
  - SPI Flash
  - JTAG
  - RS485
  - CAN
  - Ethernet
  - …
- **Monitor Interfaces**
  - 16x 12 bits ADC
  - 2x UARTs
  - 1x CAN
  - 10x PWM outputs

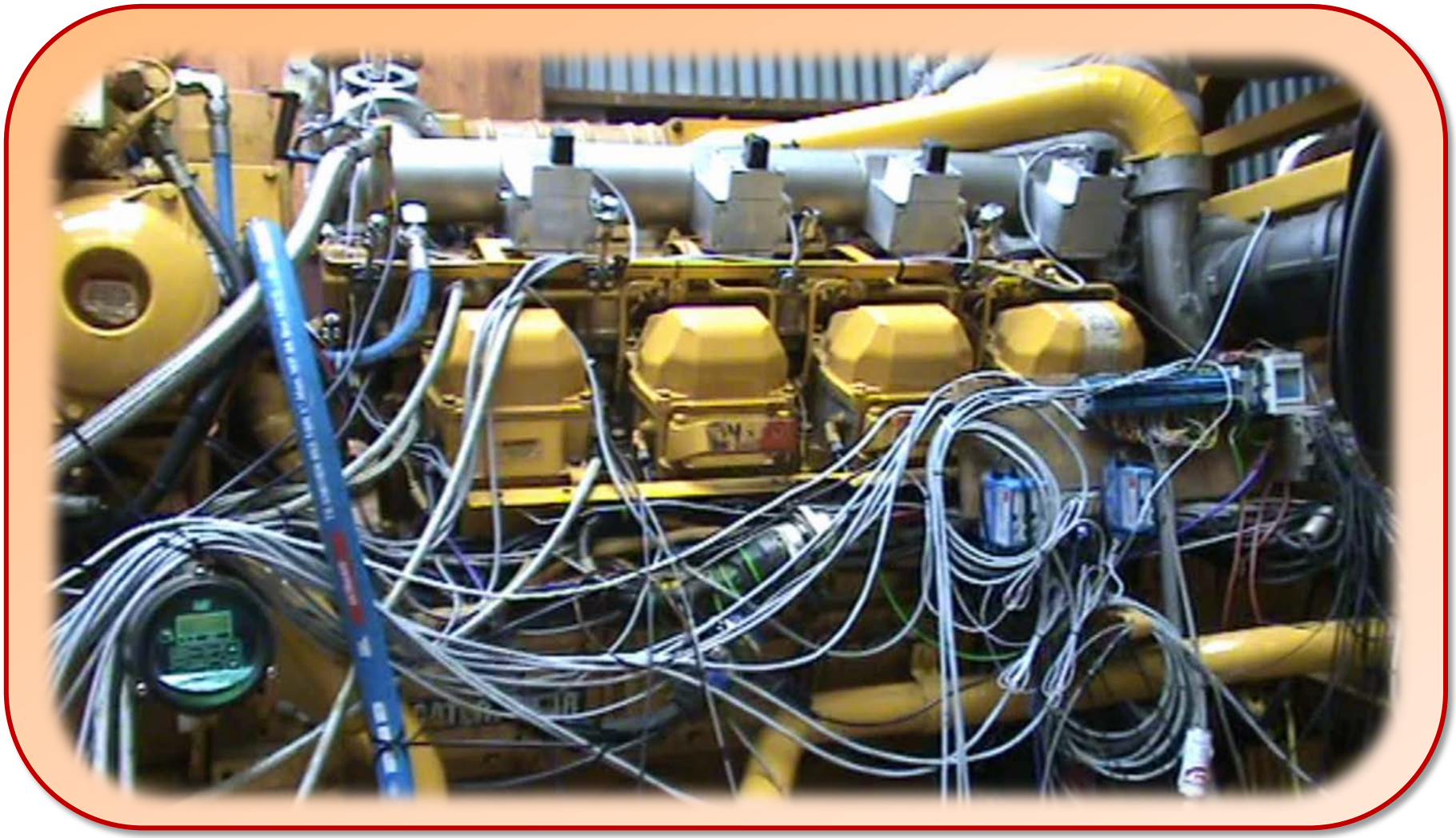


AR Electronic Control Unit cont

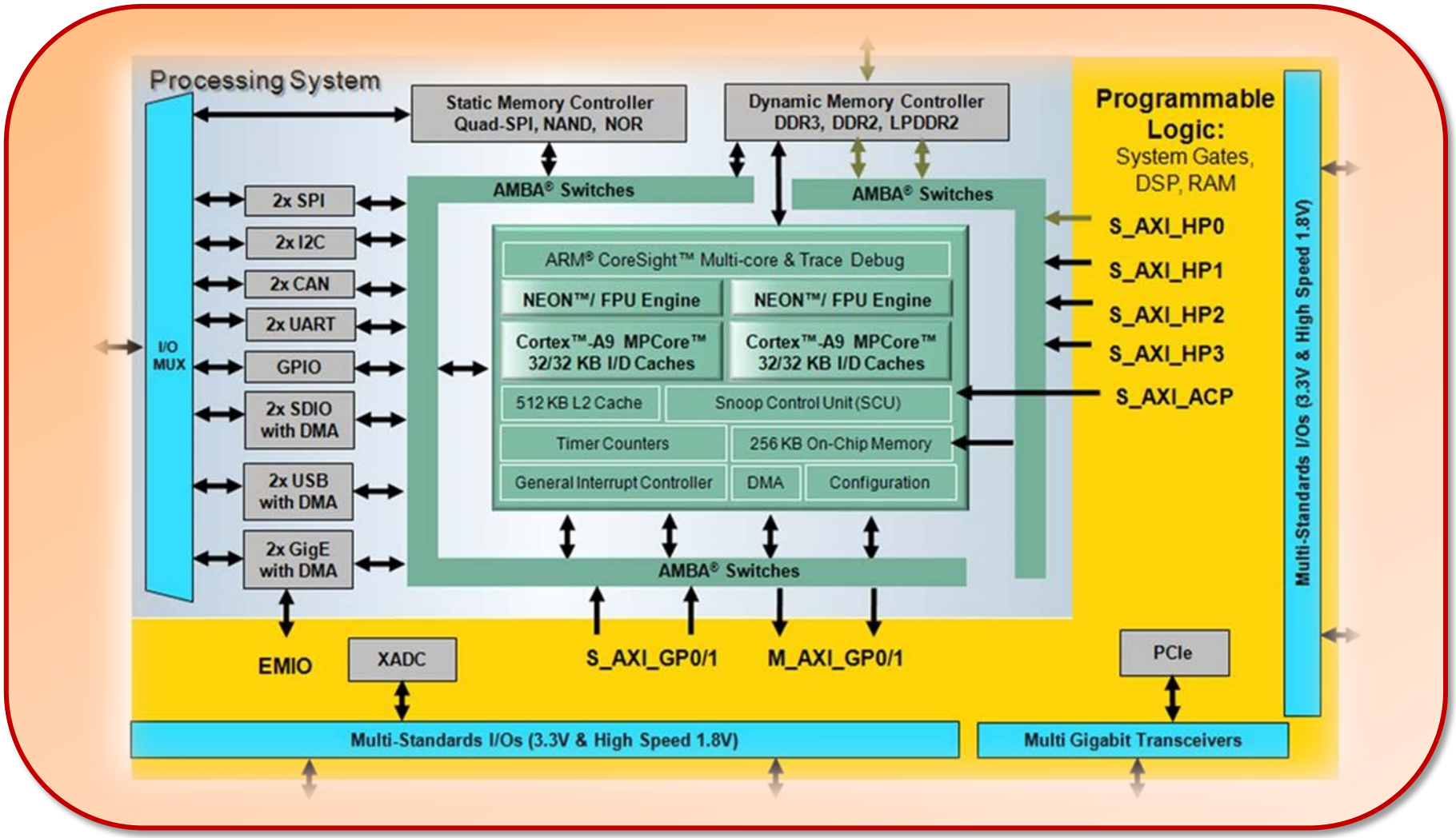
- **Electromechanical drivers**
  - High Voltage IO
  - High Current IO
- **Digital Interfaces**
  - High Current IO
  - Low Current IO
  - Sensor Inputs
- **Analog Interfaces**
  - High Speed Inputs
  - Low Speed Inputs
  - Analog Outputs
- **Misc Interfaces**
  - LED bank
  - Position Interface
  - Expansion Connector



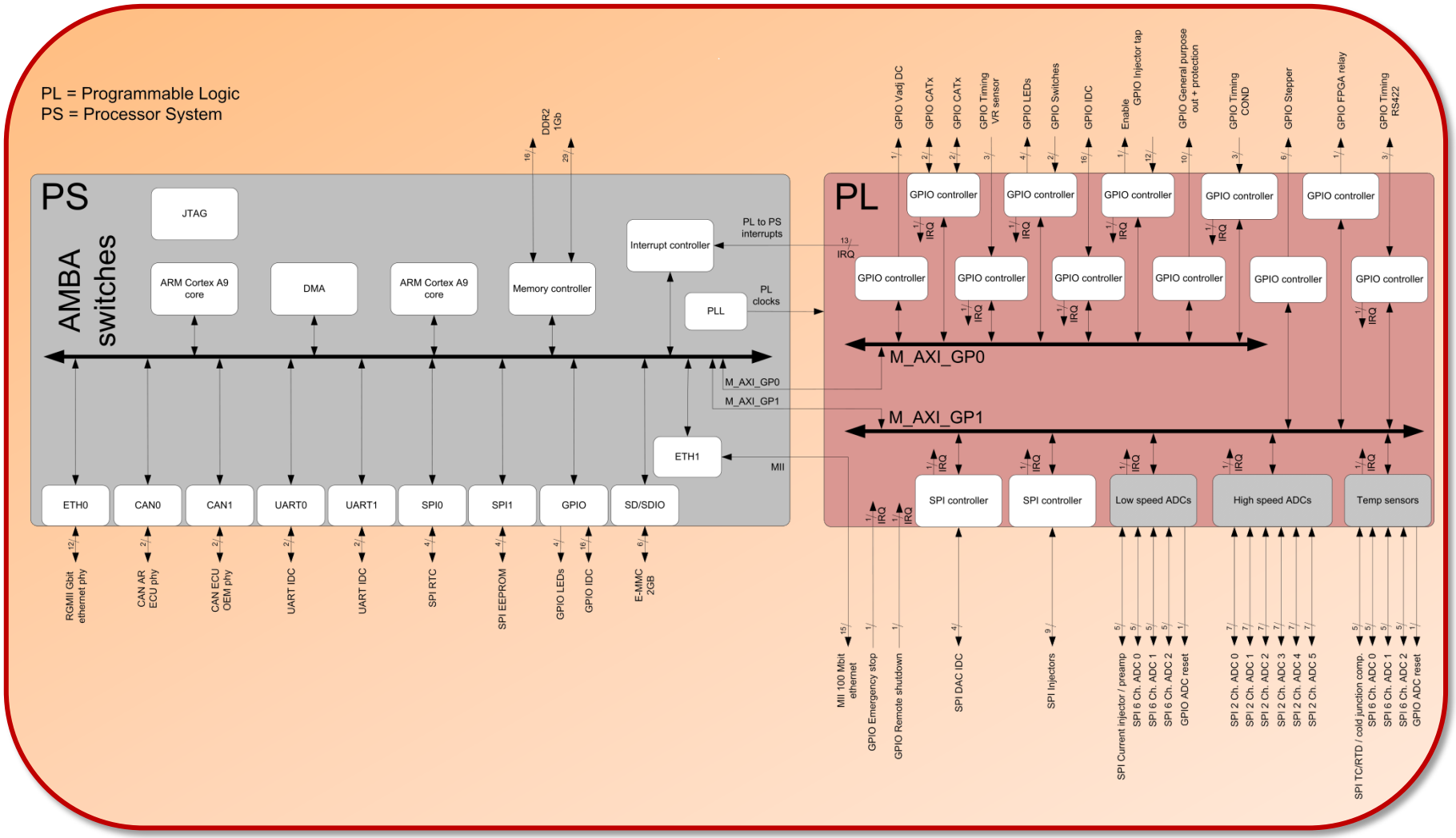
Catepillar 3500 & *AR*-ECU



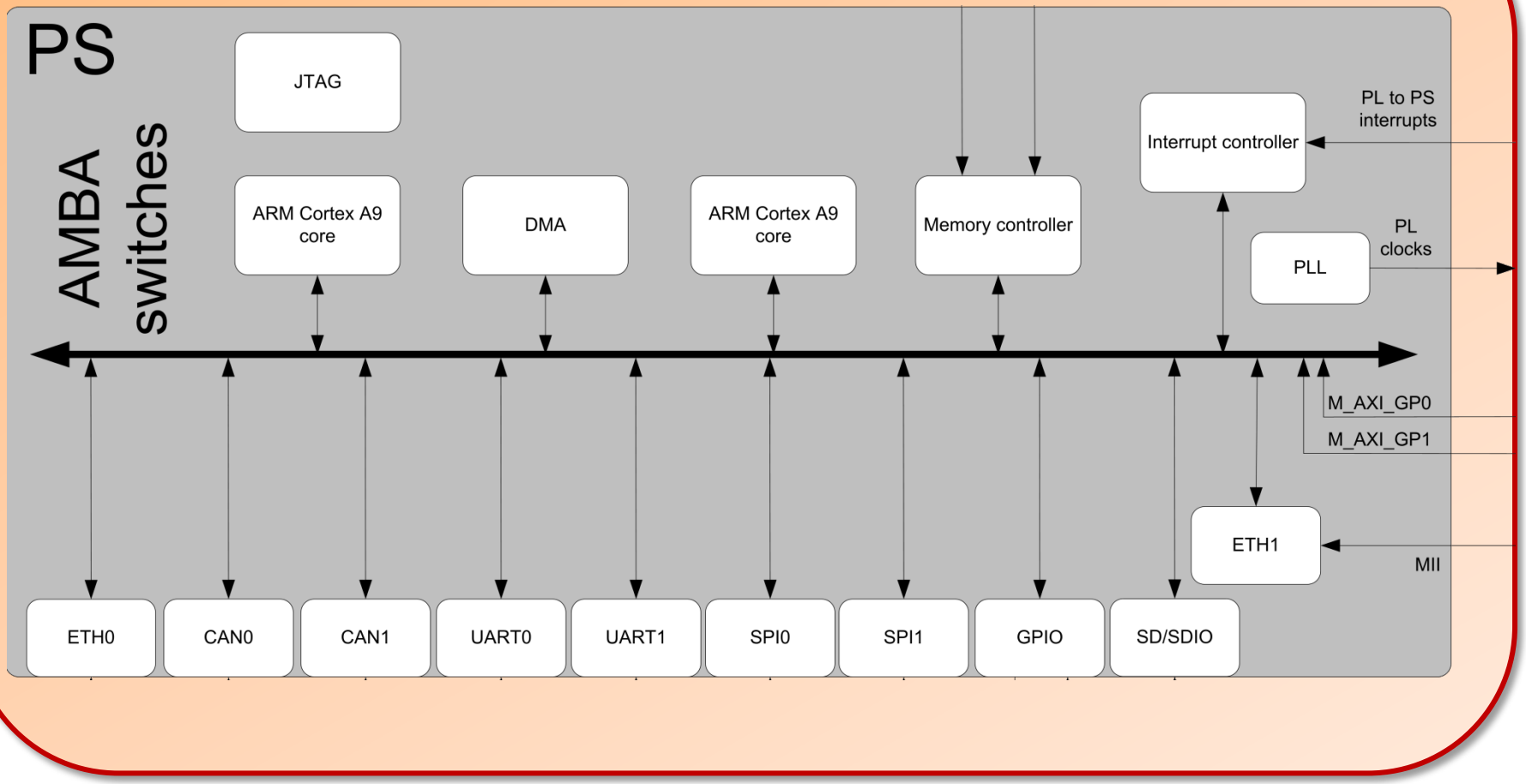
Zynq Architecture



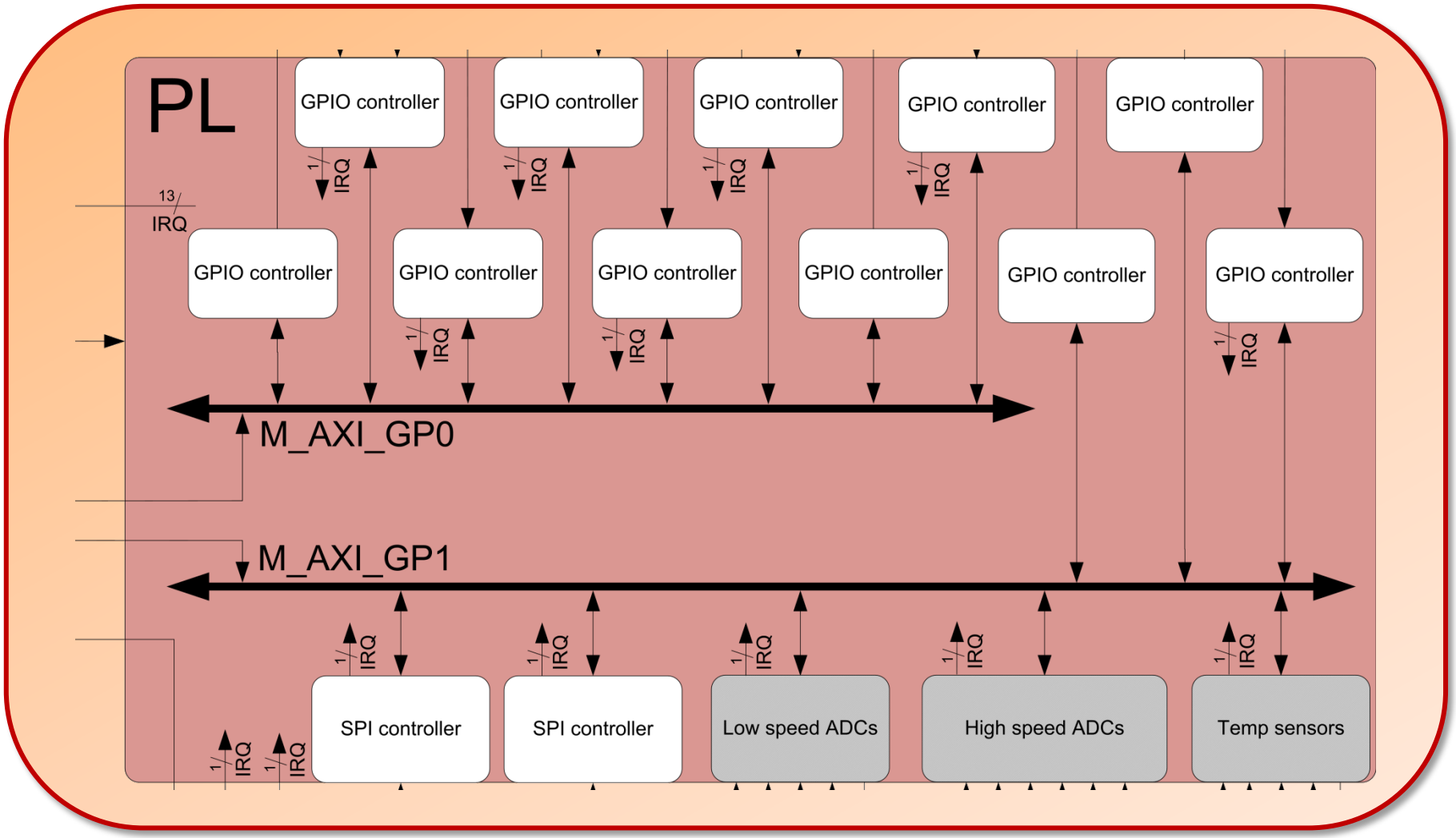
Zynq Architecture cont



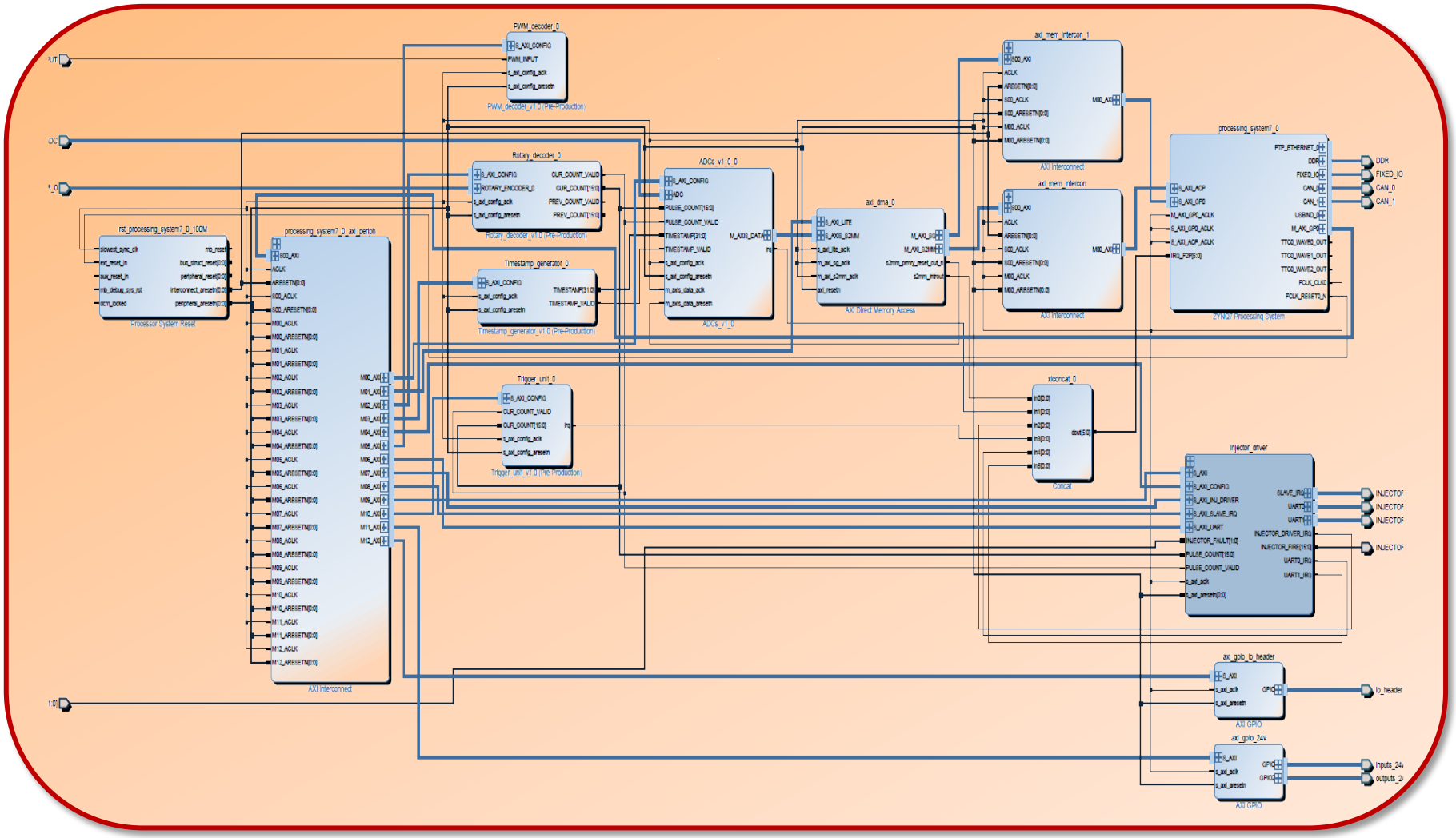
Zynq Architecture cont



Zynq Architecture cont

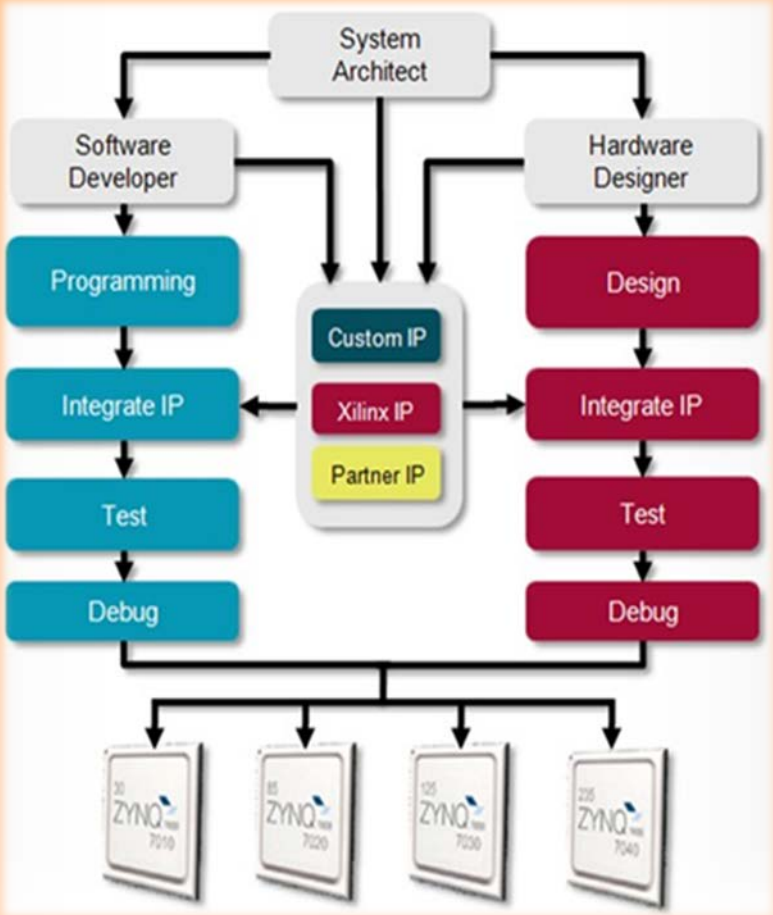


Zynq Architecture cont



Design Flow and Tools

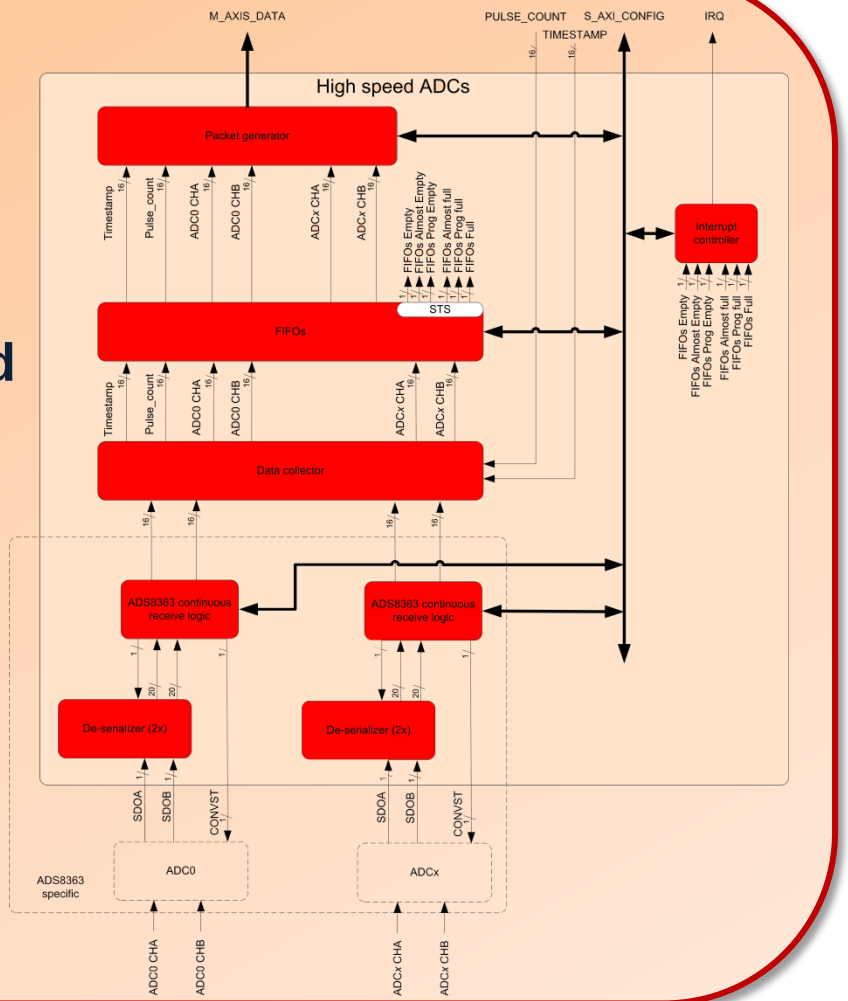
- **Software**
  - Real-time OS
  - Drivers
  - APIs
  - …
- **Tools**
  - Vivado SDK
  - Matlab
  - …



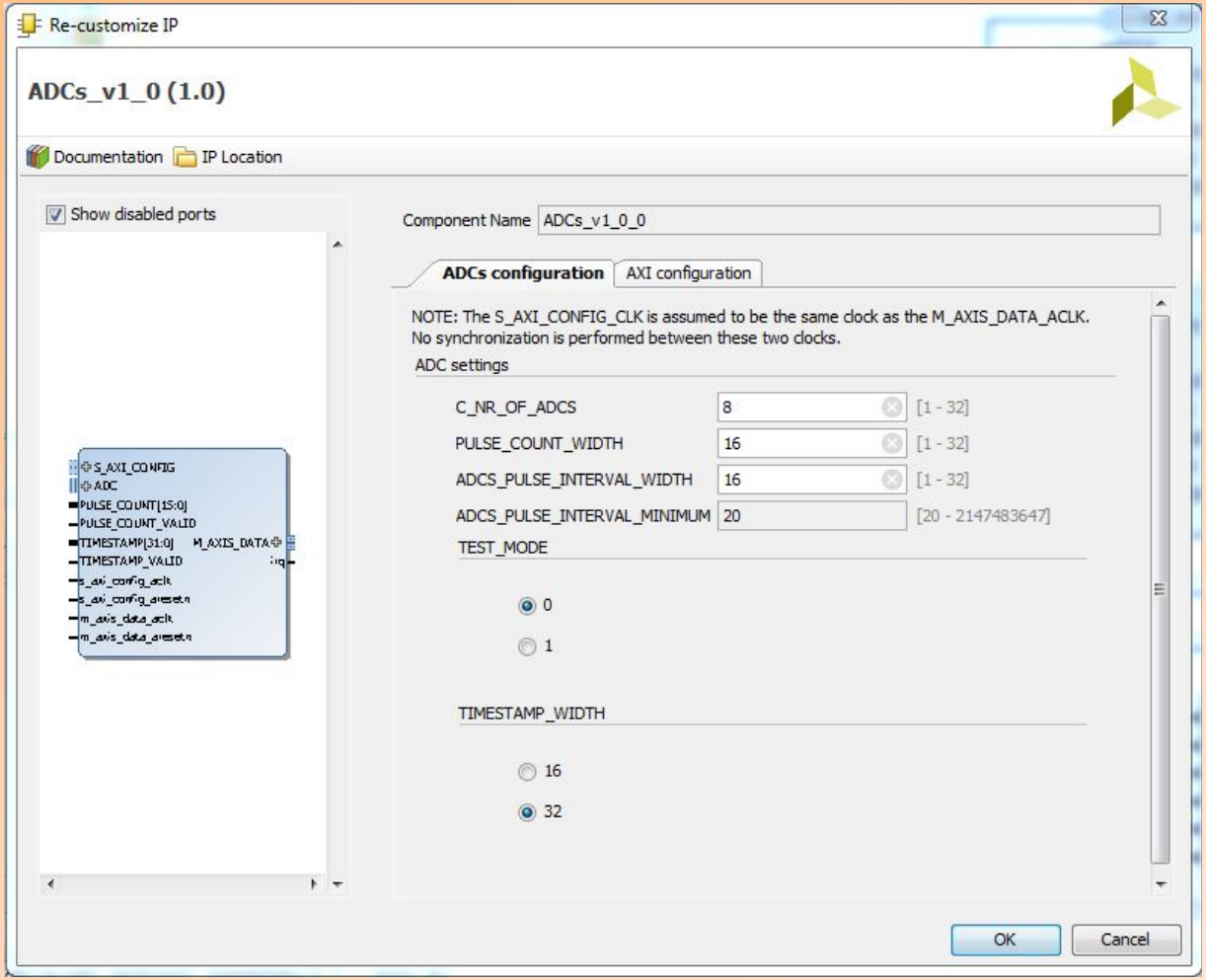
- **FPGA hardware**
  - ADC IP Interface
  - eMMC interface
  - DDR3 memory
  - IO
  - JTAG
  - …
- **Tools**
  - Vivado
  - Matlab
  - ModelSim
  - Altium
  - …

High-Speed ADC IP-block

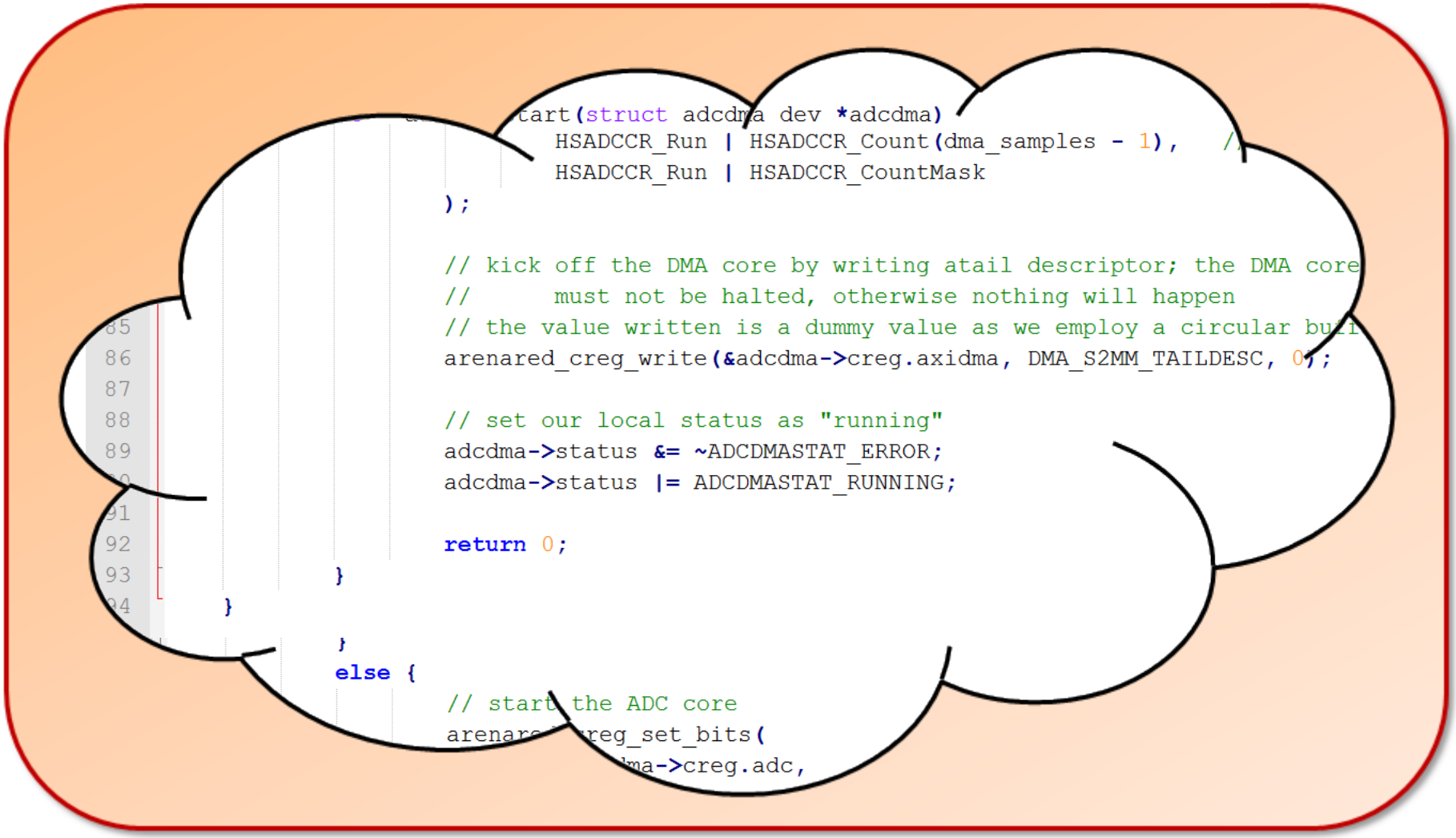
- High-Speed ADC to read pressure sensors
- AXI streaming interface for transmitting ADC data completed with timestamp and crank angle
- AXI slave interface for configuration
- Interface to 2 ADC channels of the ADS8363



High-Speed ADC IP-block cont



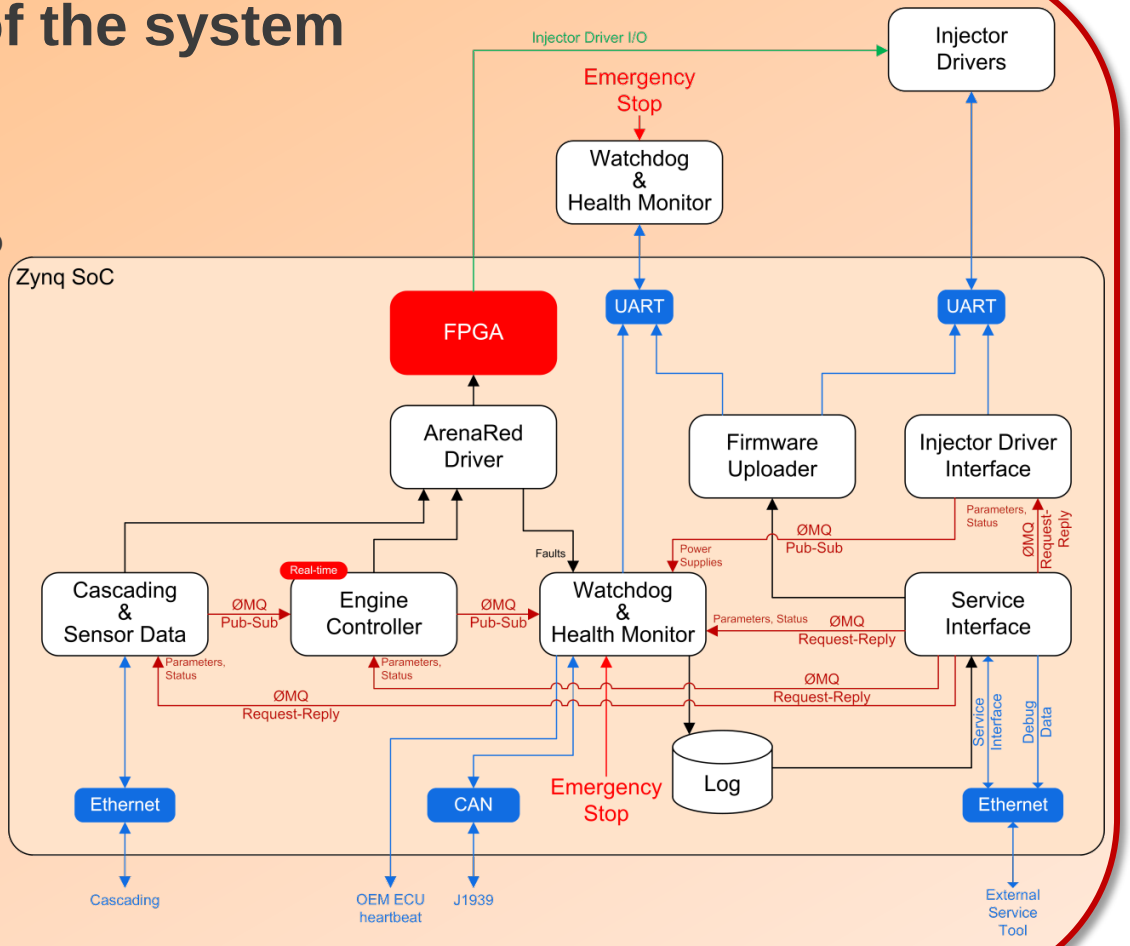
High-Speed ADC IP-block cont



```
tart(struct adcdma dev *adcdma)
        HSADCCR_Run | HSADCCR_Count(dma_samples - 1),    /
        HSADCCR_Run | HSADCCR_CountMask
    );

    // kick off the DMA core by writing atail descriptor; the DMA core
    //      must not be halted, otherwise nothing will happen
    // the value written is a dummy value as we employ a circular buf
    arenared_creg_write(&adcdma->creg.axidma, DMA_S2MM_TAILDESC, 0);

    // set our local status as "running"
    adcdma->status &= ~ADCDMASTAT_ERROR;
    adcdma->status |= ADCDMASTAT_RUNNING;

    return 0;
}
}
}
else {
    // start the ADC core
    arenar reg_set_bits(
        ma->creg.adc,
```

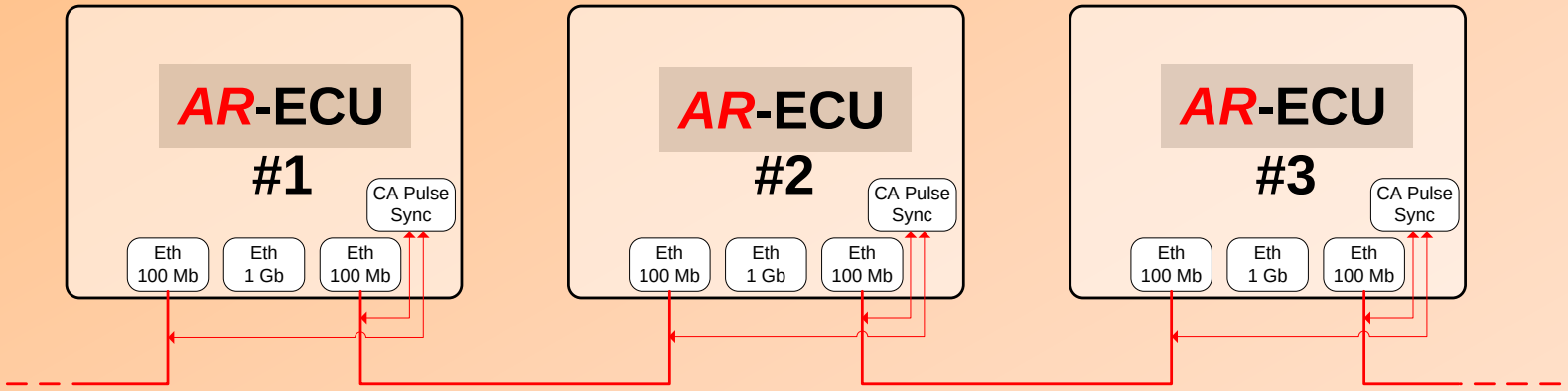
Software Partitioning and Communication Channels

- Zynq SoC heart of the system
  - Linux with Realtime extension Xenomai
  - Licensed & custom IP in FPGA PL
  - Microprocessor for Watchdog & Health Monitor
  - Software controlled EMD Injector Drivers
  - Communication cascading channel



Cascading Channels

- **Multiple *AR*-ECUs can be cascaded so the system can control more than eight cyclinders.**



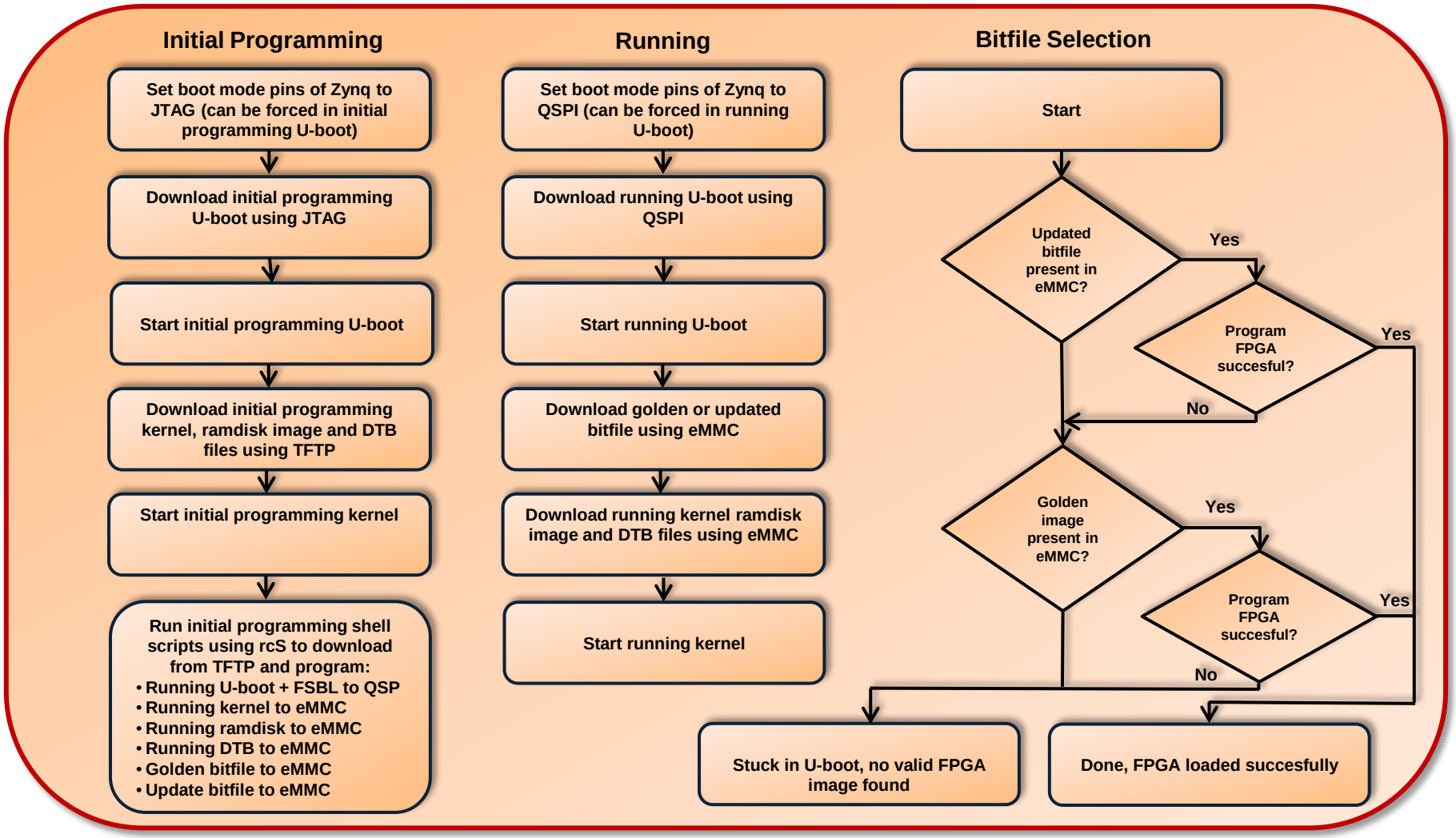
  - **Since Ethernet is non real-time by nature, crucial real-time signals will be routed over the four dedicated wires.**

Why Linux with Xenomai

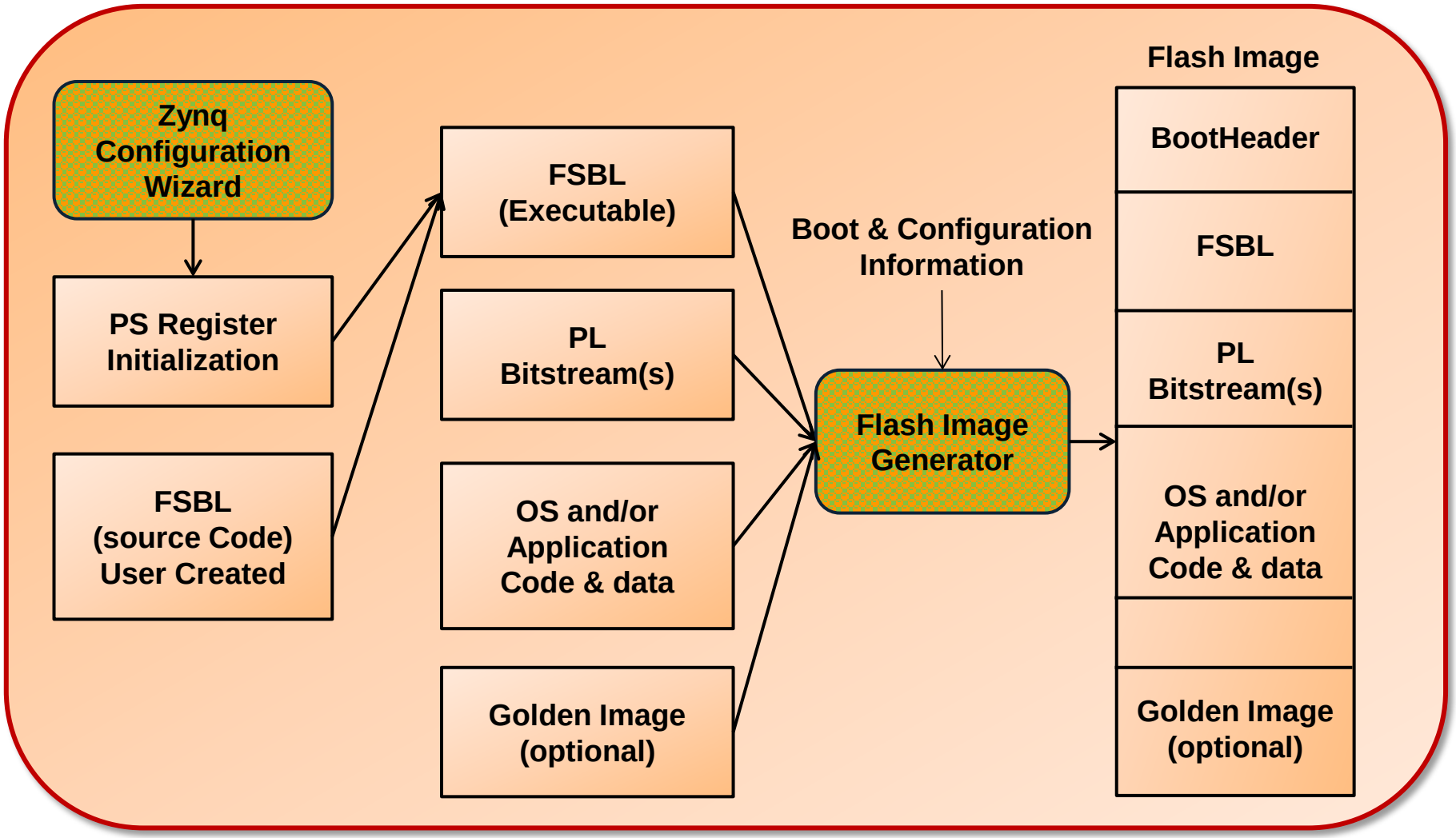
- Zynq All Programmable SoC dual core Cortex-A9
- Xilinx offers a kernel build and Linux kernel distribution by 3rd parties

- Support Linux 3.14.17 with patches for real-time extension Xenomai 2.6.4 within SDK, version choice is limited
- Xenomai offers greater flexibility and consistency because the real-time tasks can share the same drivers, synchronization primitives and memory as other non real-time tasks

eMMC Configuration



Flash Image Generation



Core|Vision

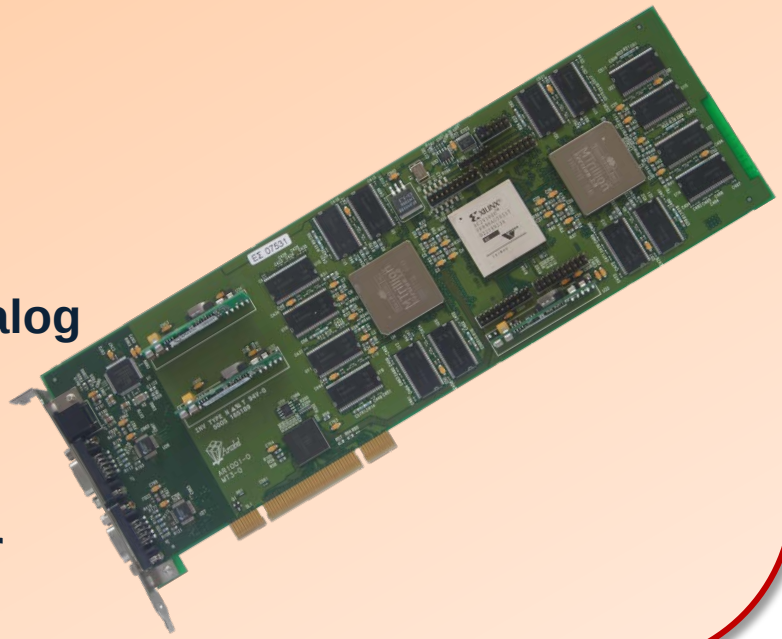
4 NOV ← BRABANTHALLEN DEN BOSCH D&E event 2015

XILINX ALL PROGRAMMABLE™

Our competences

Core|Vision has more than 125 man years of design experience in hard- and software development. Our competence areas are:

- System Design
- FPGA Design
- Consultancy / Training
- Digital Signal Processing
- Embedded Real-time Software
- App development, IOS Android
- Data Acquisition, digital and analog
- Modeling & Simulation
- ASIC Conversion & Prototyping
- PCB design & Layout
- Doulos & Xilinx Training Partner



4 NOV ← BRABANTHALLEN DEN BOSCH D&E event 2015

CORE | Vision
NEXT LEVEL | EMBEDDED DEVELOPMENT

Cereslaan 10b
5384 VT Heesch
✆ +31 (0)412 660088

www.*core-vision*.nl

Email : info@*core-vision*.nl

4 NOV ← BRABANTHALLEN DEN BOSCH D&E event 2015

CORE | Vision

NEXT LEVEL | EMBEDDED DEVELOPMENT

- FPGA DESIGN
- SYSTEM DEVELOPMENT
- DEDICATED ELECTRONICS
- EMBEDDED SOFTWARE
- DESIGN SERVICES
- MODELING AND SIMULATION

CORE Vision NEXT LEVEL EMBEDDED DEVELOPMENT

CORE Vision NEXT LEVEL EMBEDDED DEVELOPMENT

Visit our booth 28

BRINGING YOU THE NEXT LEVEL IN EMBEDDED DEVELOPMENT

Training Program

4 NOV ← BRABANTHALLEN DEN BOSCH D&E event 2015

XILINX ALL PROGRAMMABLE™

Essentials of FPGA Design	*1 day*
Designing for Performance	*2 days*
Advanced FPGA Implementation	*2 days*
Design Techniques for Lower Cost	*1 day*
Designing with Spartan-6 and Virtex-6 Family	*3 days*
Essential Design with the PlanAhead Analysis Tool	*1 day*
Advanced Design with the PlanAhead Analysis Tool	*2 days*
Xilinx Partial Reconfiguration Tools and Techniques	*2 days*
Designing with the 7 Series Families	*2 days*

Training Program

4 NOV ← BRABANTHALLEN DEN BOSCH D&E event 2015

XILINX ALL PROGRAMMABLE™

Vivado Essentials of FPGA Design	*2 days*
Vivado Design Suite Tool Flow	*1 day*
Vivado Design Suite for ISE Users	*1 day*
Vivado Avanced XDC and STA for ISE Users	*2 days*
Vivado Advanced Tools & Techniques	*2 days*
Vivado Static Timing Analysis and XDC	*2 days*
Debugging Techniques Using Vivado Logic Analyzer	*1 day*
Essential Tcl Scripting for Vivado Design Suite	*1 day*
Vivado FPGA Design Methodology	*1 day*
Designing with the UltraScale Architecture	*2 days*

Training Program

4 NOV ← BRABANTHALLEN DEN BOSCH D&E event 2015

XILINX ALL PROGRAMMABLE™

➢ Designing with Multi Gigabit Serial IO	*3 days*
➢ High Level Synthesis with Vivado	*2 days*
➢ C-Based HLS Coding for Hardware Designers	*1 day*
➢ C-Based HLS Coding for Software Designers	*1 day*
➢ DSP Design Using System Generator	*2 days*
➢ Essential DSP Implementation Techniques for Xilinx FPGAs	*2 days*

Training Program

XILINX
ALL PROGRAMMABLE™

Embedded Systems Design	*2 days*
Embedded Systems Software Design	*2 days*
Advanced Features and Techniques of SDK	*2 days*
Advanced Features and Techniques of EDK	*2 days*
Zynq All Programmable SoC Systems Archicture	*2 days*
Zynq All Programmable SoC Accelerators	*1 day*
C Language Programming with SDK	*2 days*
Embedded Design with PetaLinux SDK	*2 days*
Embedded C/C++ SDSoC Development Environment and Methodology	*1 day*

Training Program

- VHDL Design for FPGA — *3 days*
- Advanced VDHL — *2 days*
- Comprehensive VHDL — *5 days*
- Expert VHDL Verification — *3 days*
- Expert VDHL Design — *2 days*
- Expert VHDL — *5 days*
- Essential Digital Design Techniques — *2 days*