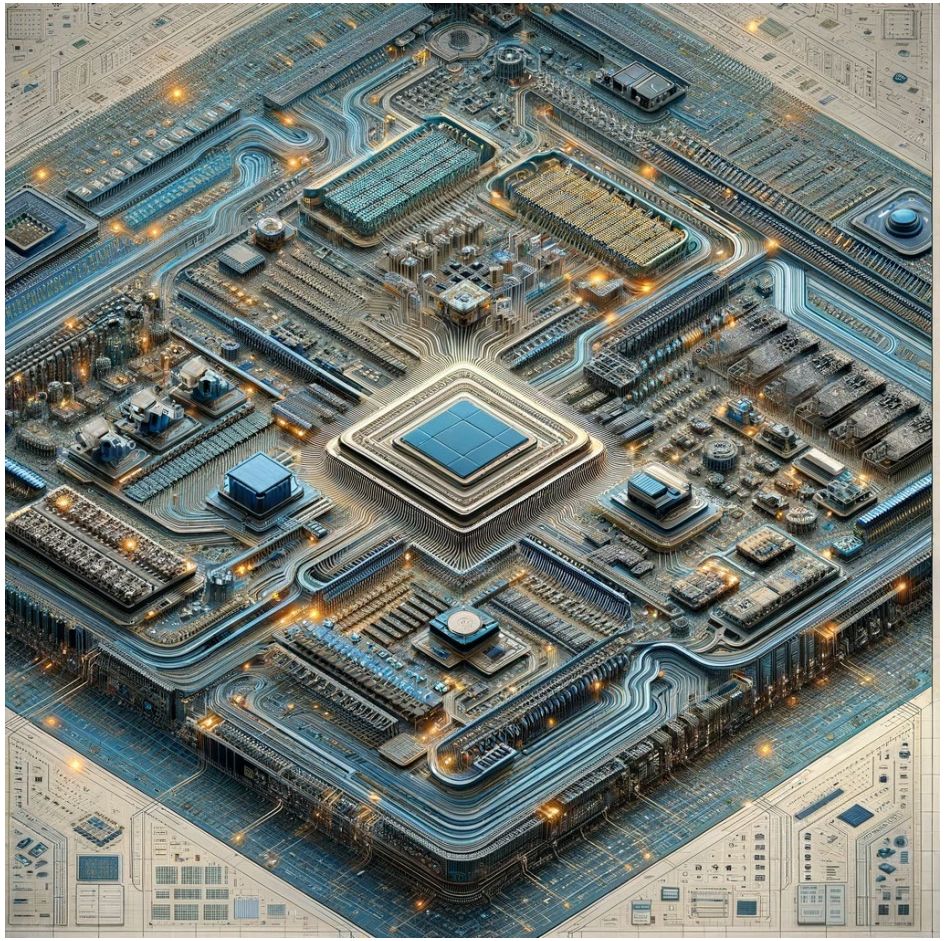
ADIUVO

ENGINEERING AND TRAINING, LTD.

FPGA Architect & Model Based Engineering

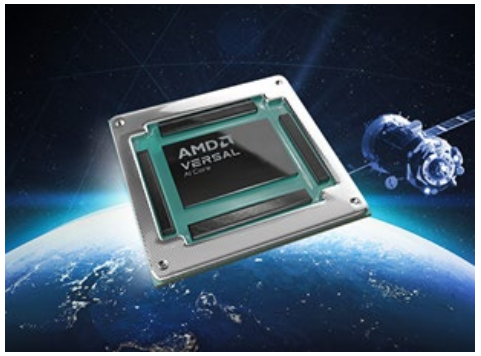
Adam@AdiuvoEngineering.com

Devices are getting large

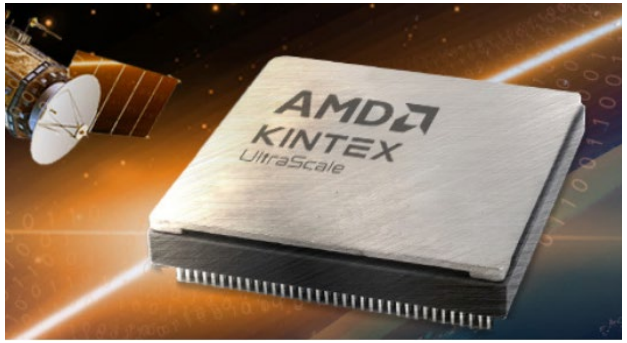


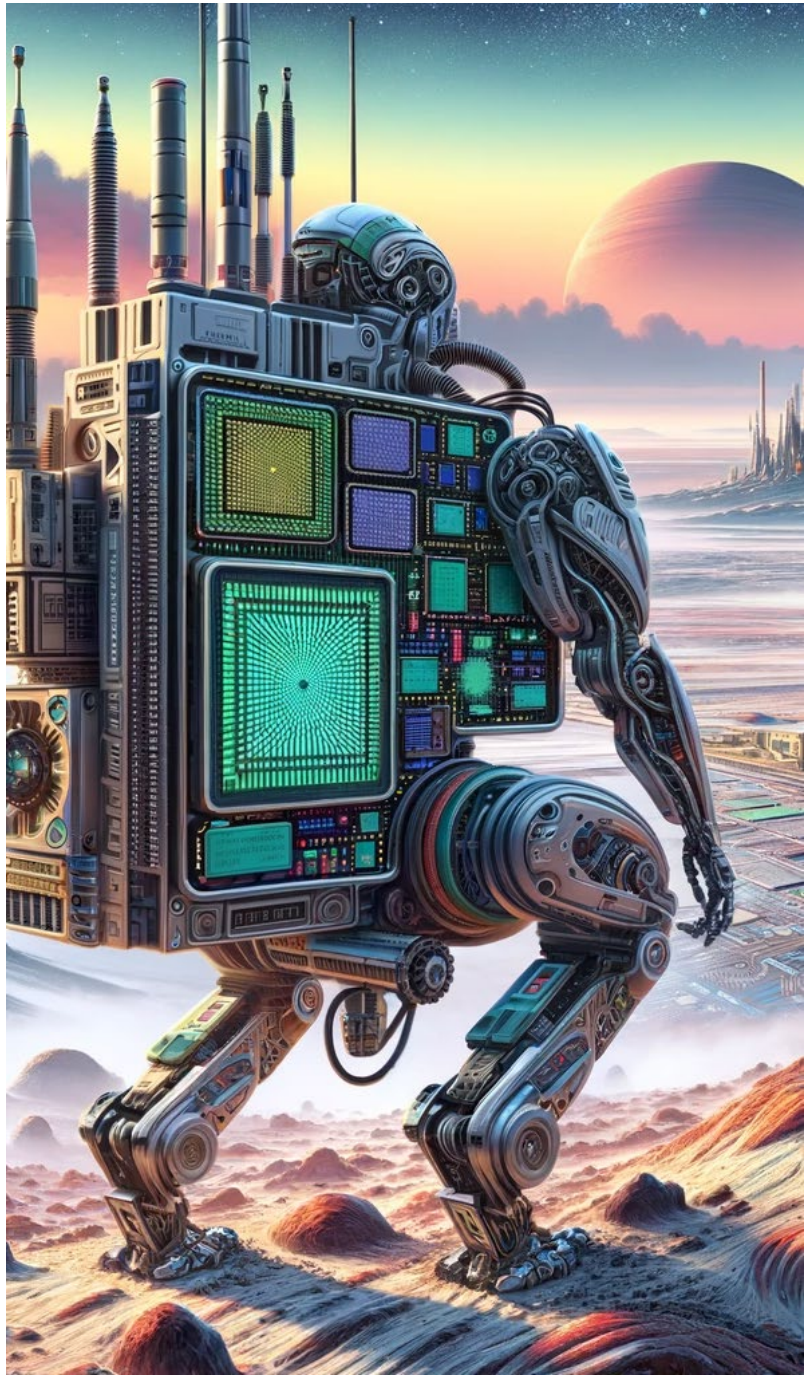
481,000 LUTs

899,840 LUTs



331,000 LUTs



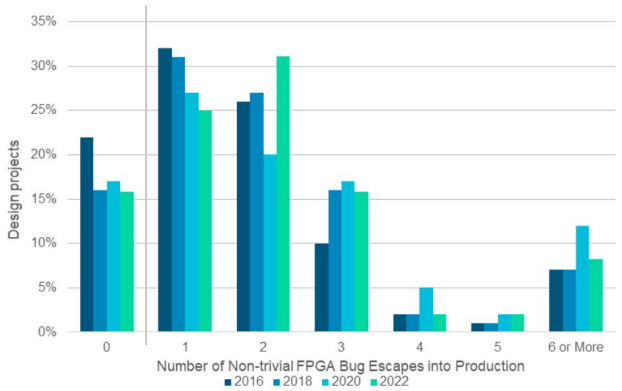


We are not getting better at designing

FPGA projects are performing no better than ASIC

84%

FPGA design projects have non-trivial bug escapes into production



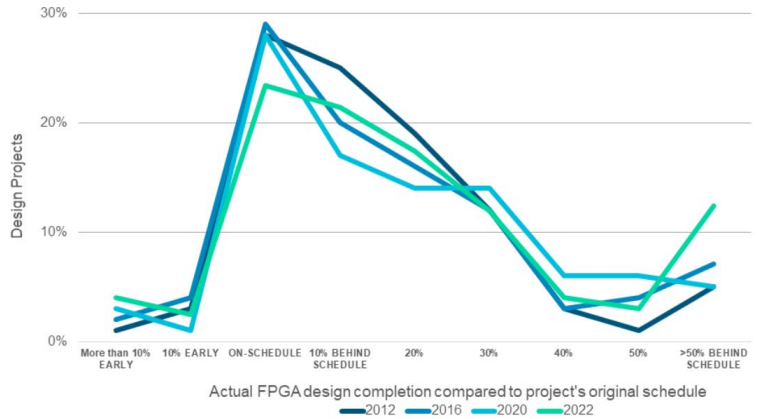
Most FPGA projects miss schedule

70%

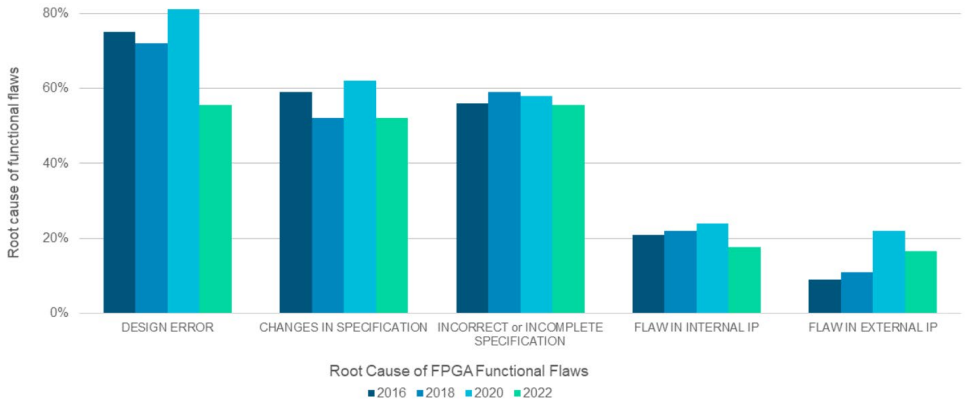
Projects behind schedule

12%

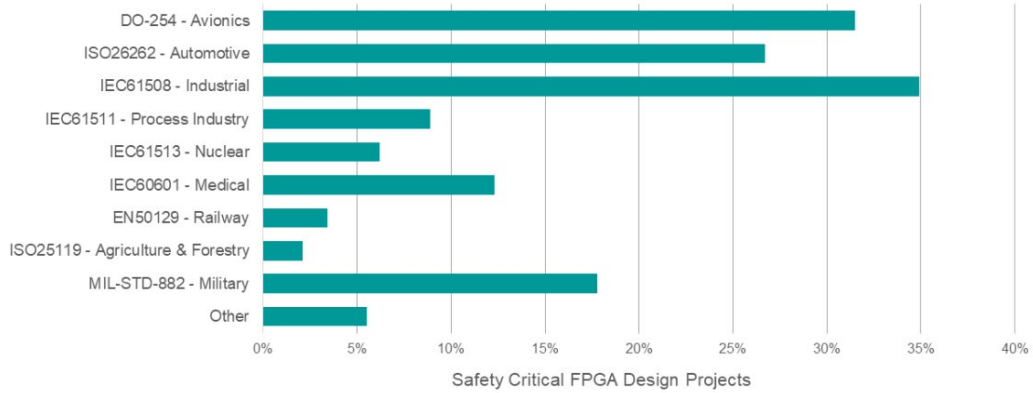
Projects behind schedule more than 50%



Root cause of FPGA functional flaws



FPGA project adoption of various functional safety standards



We cannot keep designing as we have been.

"Insanity is doing the same thing over and over again and expecting different results."

ADIUVO
ENGINEERING AND TRAINING, LTD.

Adiuvo Model Based Flow

Adiuvo has completed two projects recently for Space Industry using model-based techniques. What we have seen is on time development and fewer bugs.

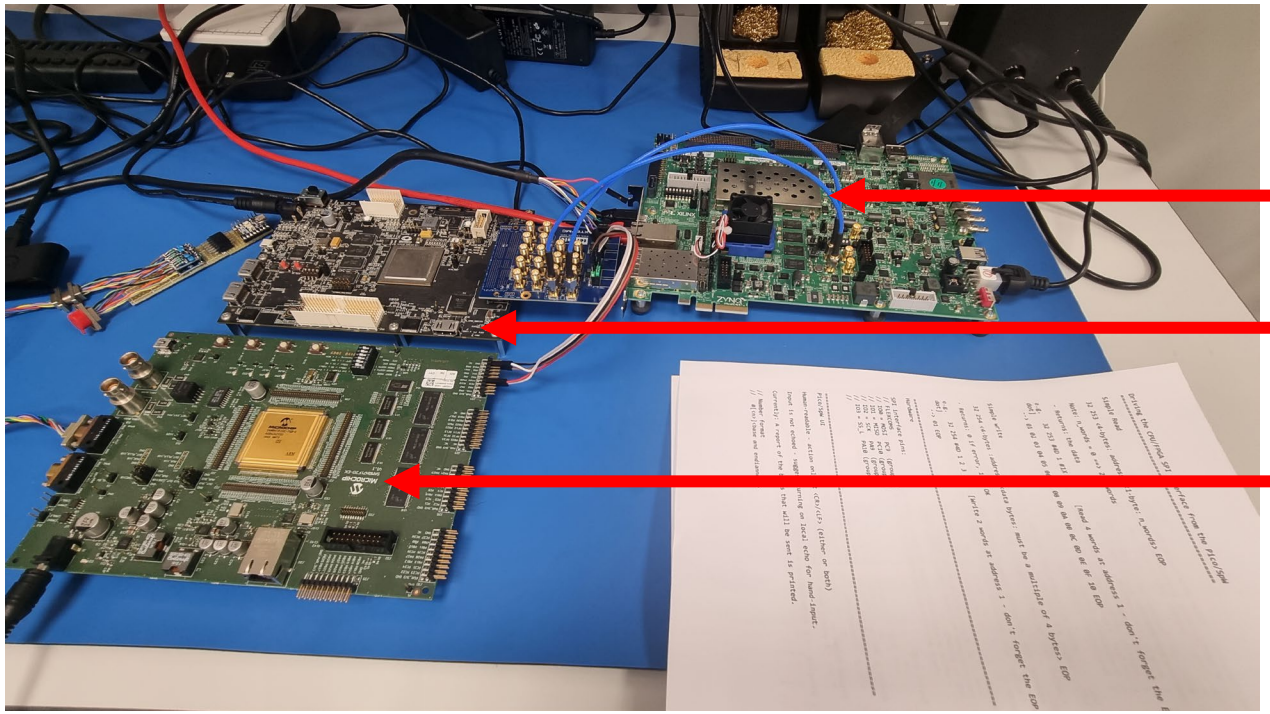
Applications

1. Space Imager based around PolarFire and SAM71RH– Complex image sensor interface, DDR4, Binning, Frame Subtraction, Output over Serdes – Adiuvo developed FPGA and SW applications.

2. Terrestrial / Space Laser Communications based around Xilinx MPSoC – Camera Interfacing, Image Processing, Spot detection, Histograms etc. Adiuvo Developed FPGA and SW.

Recently shared with a Adiuvo Client to help with their PolarFire Project

Camera Development Project

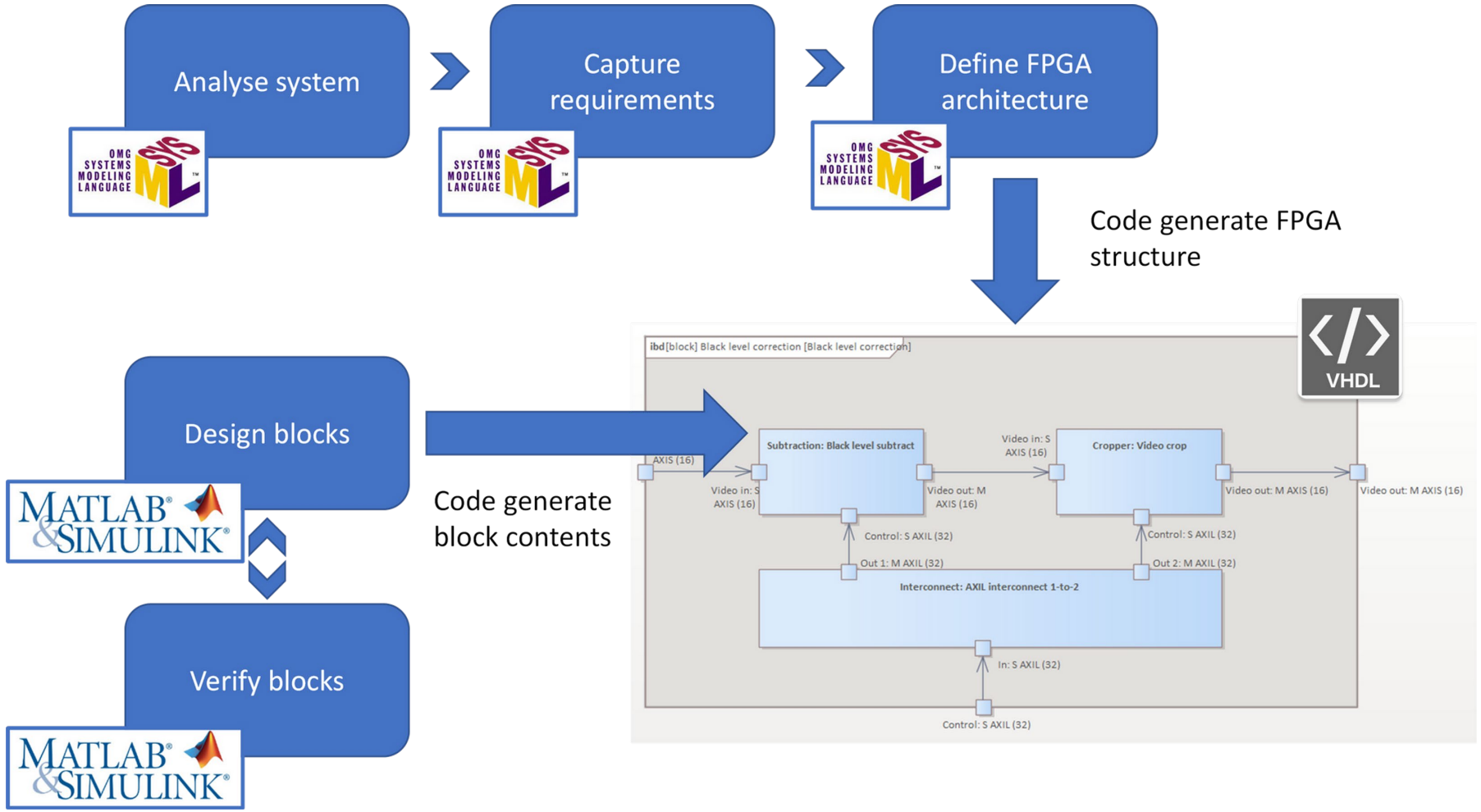


Zynq MPSoC Frame Grabber

PolarFire – Imager FPGA Test Bed

SAM71RH

Adiuvo Model Based Flow



Adiuvo Model Based Flow

Basis of the model-based flow is an internally based architecting tool

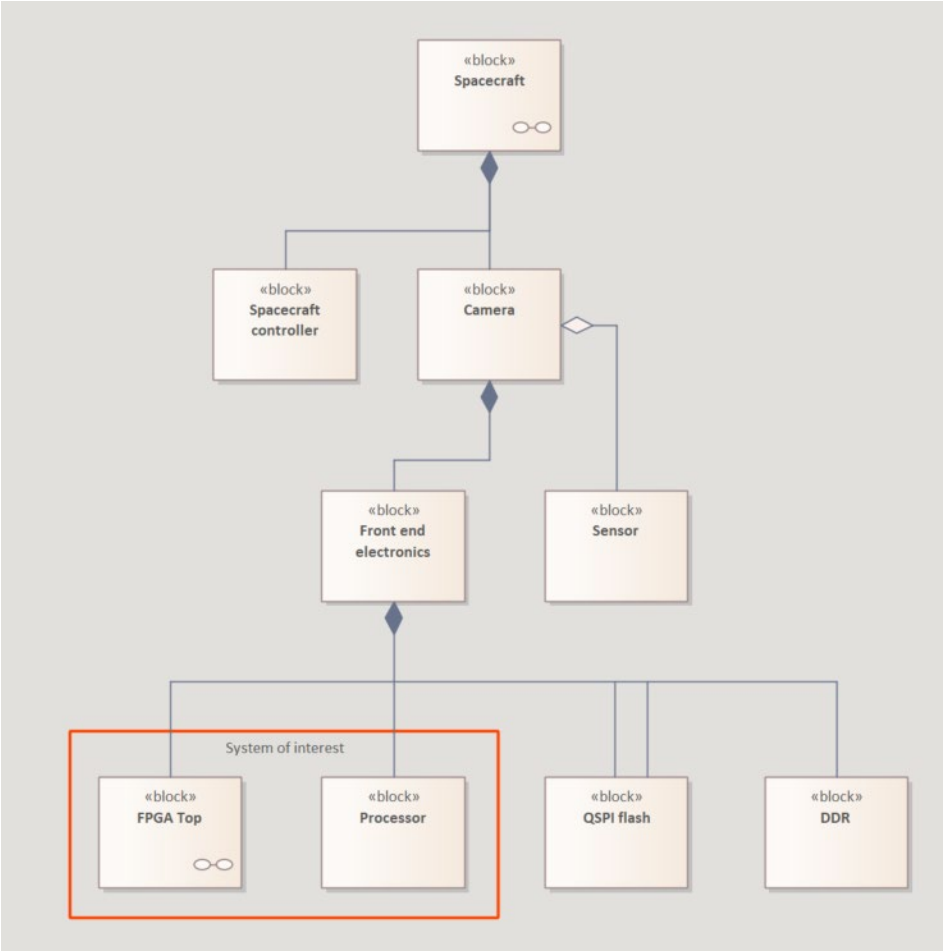
Our workflow typically starts with us developing a ‘system level’ model that allows us to explore issues at the system level and to help us understand the context in which the FPGA is operating.

We can use that model for design reviews e.g. SRR / PDR and get customer buy in to the solution

Robust design that traces directly back to our requirements and also to the higher-level system design

Adiuvo has invested heavily in the ability to code generate the SysML model

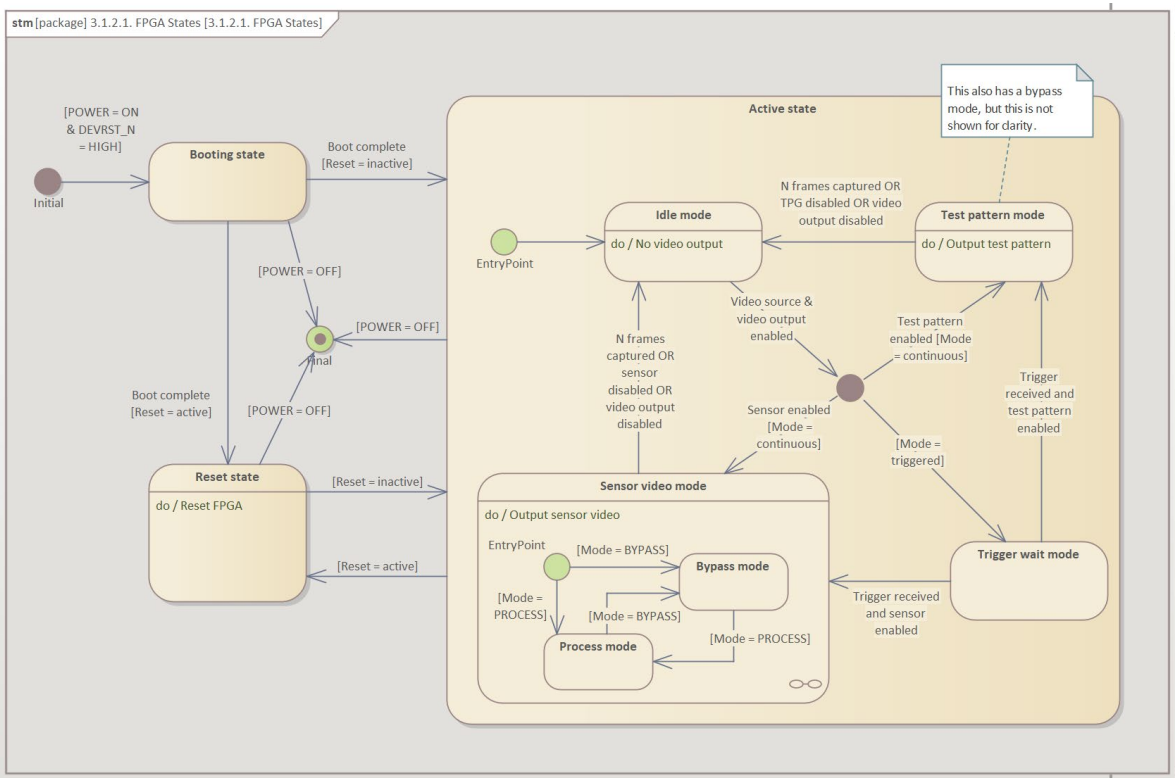
Context Diagram

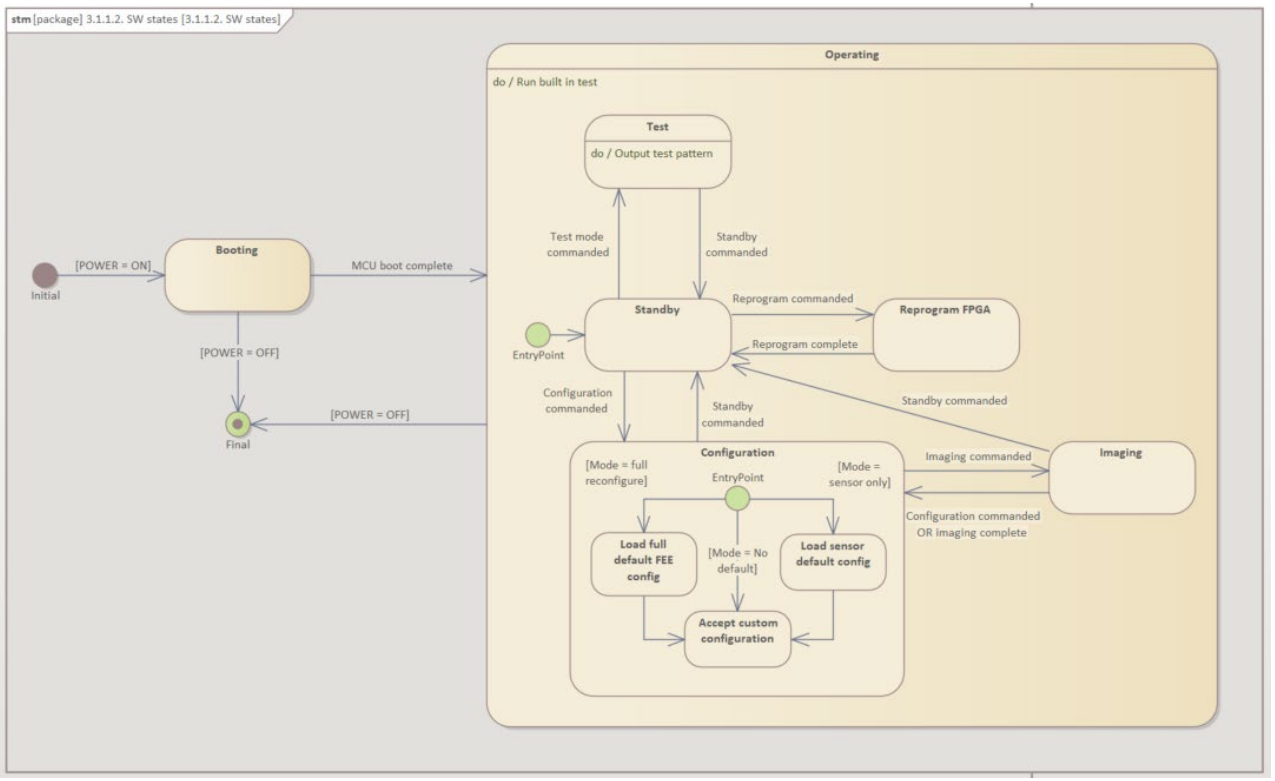


Enter Requirements

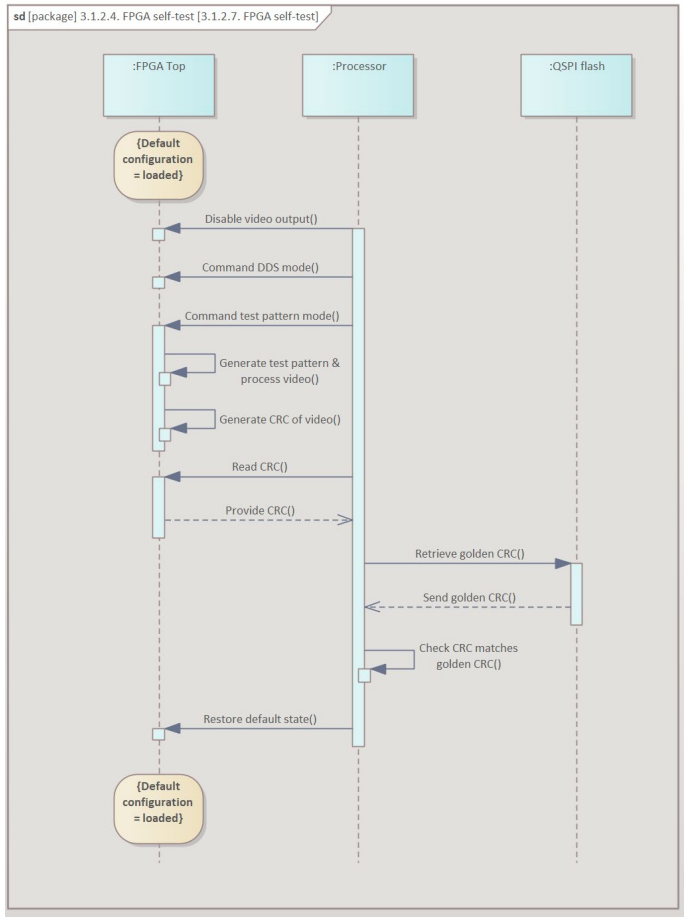
☑ **External video clock** The FPGA when the **[Video output timing]** register is set to 'external clock', **shall** clock the **[High speed serial video output]** at the same rate as the **[External serial reference clock.REFCLK]**.	Functional	Proposed	Medium	Medium
☑ **Internal video clock** The FPGA when the **[Video output timing]** register is set to 'internal clock', **shall** clock the **[High speed serial video output]** at 2.5 Gbps.	Functional	Proposed	Medium	Medium
☑ **Boot complete indicator** The FPGA **shall** indicate when the boot process has completed by asserting the signal **[FPGA boot flag.FPGA_INIT]** once boot has completed.	Functional	Proposed	Medium	Medium
☑ **Move to reset state** The FPGA, when not in the boot state, **shall** transition to the reset state when the signal **[System reset.FPGA_RESETN]** is asserted for more than 10 primary system clock cycles.	Functional	Proposed	Medium	Medium
☑ **Output states** The FPGA, when in the reset state, **shall** ensure that all outputs from the device are held in their inactive state.	Functional	Proposed	Medium	Medium
☑ **Register resets** The FPGA, when in the reset state, **shall** ensure that the following software programmable registers are in the following states:	Functional	Proposed	Medium	Medium

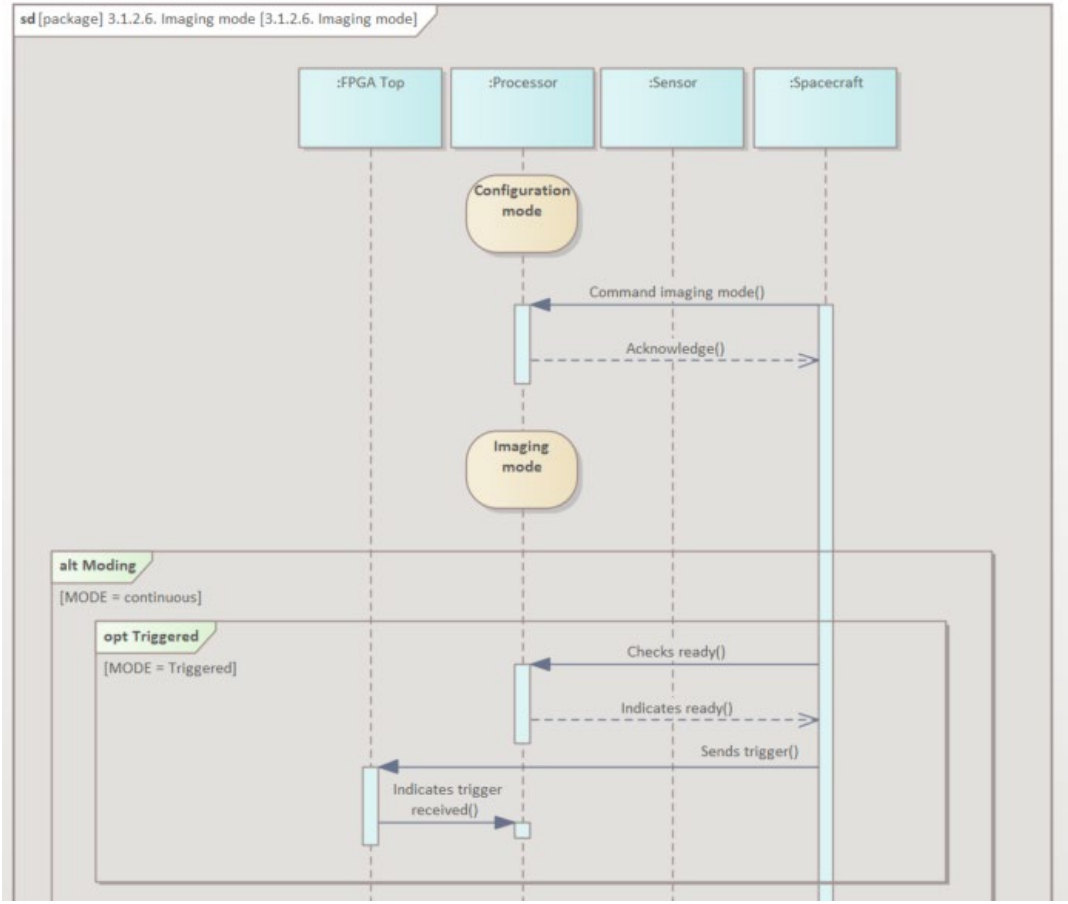
System Model - Behavior – FPGA & SW





System Model – Sequences



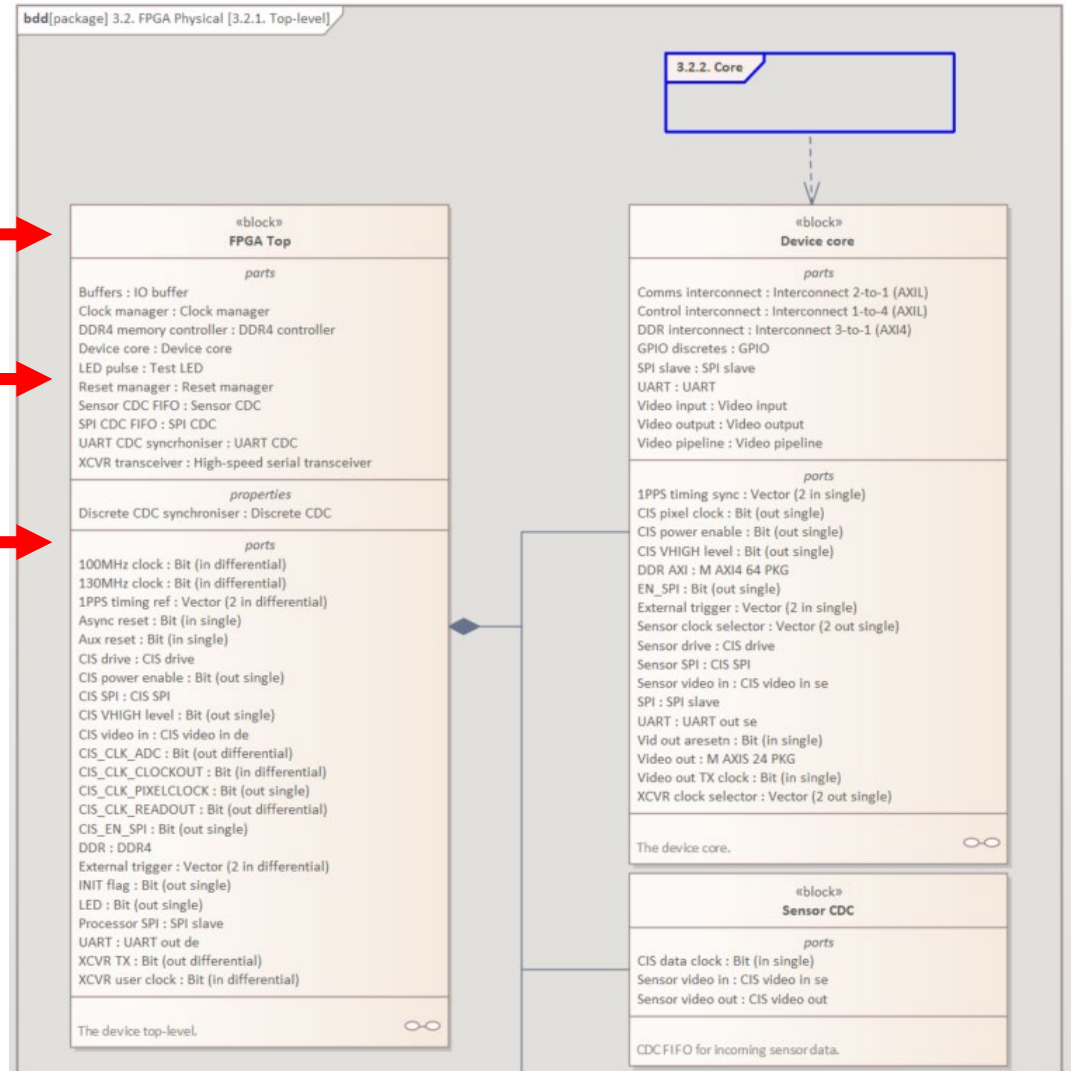


FPGA Context

FPGA Top Level

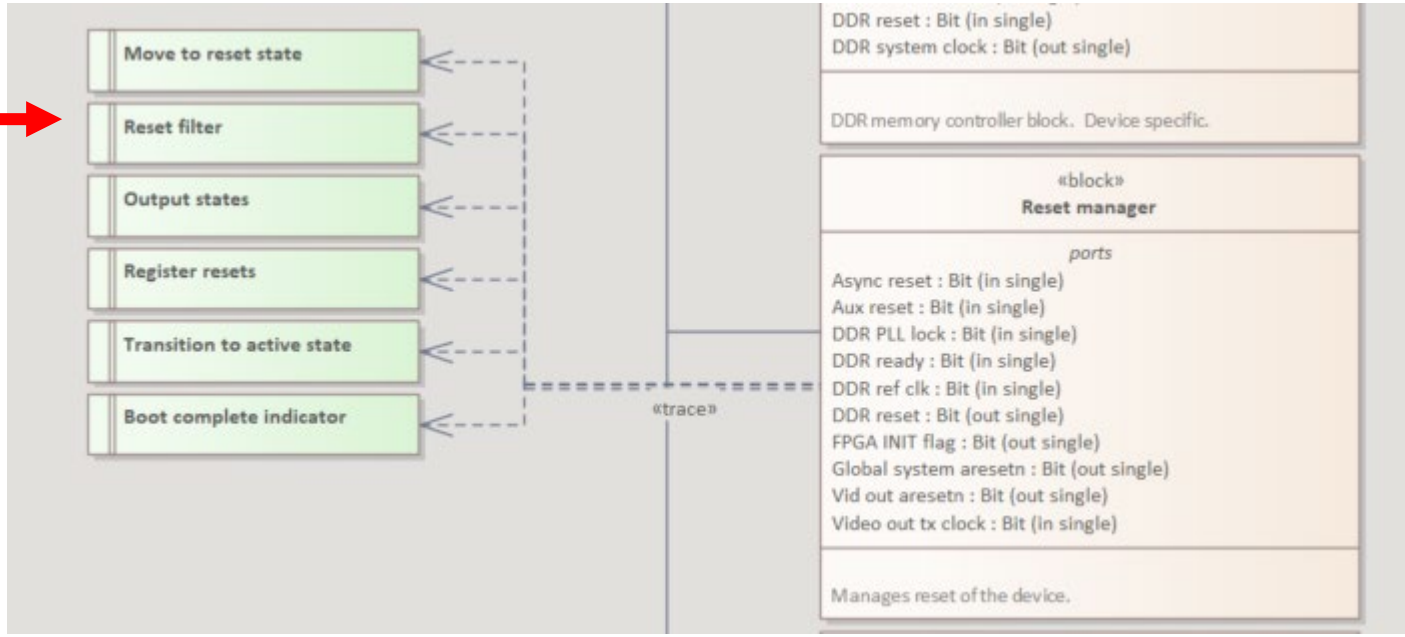
Constituent Block

FPGA Ports



FPGA Context

Trace FPGA Blocks to Requirements

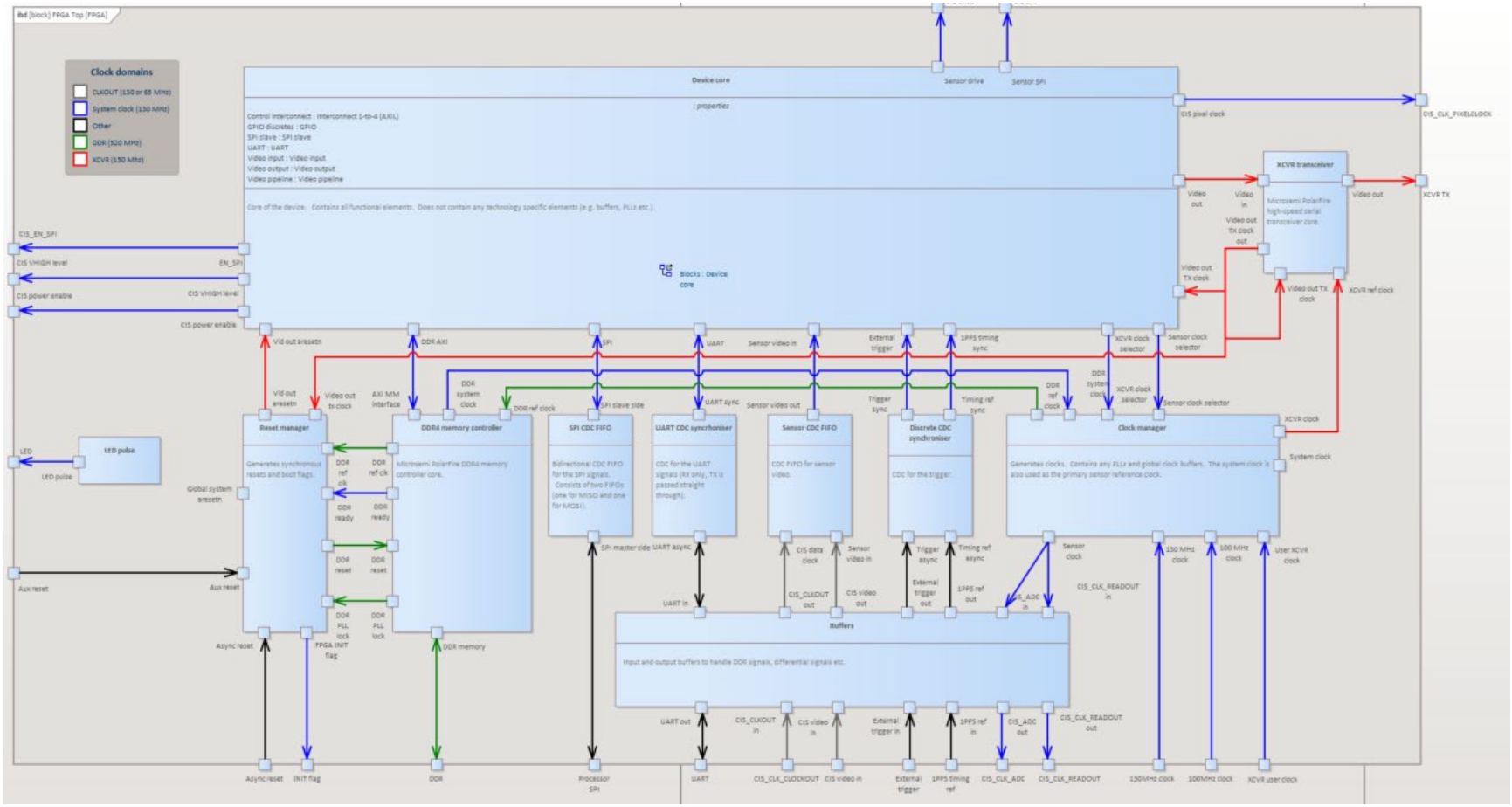


FPGA Architecture - Project One

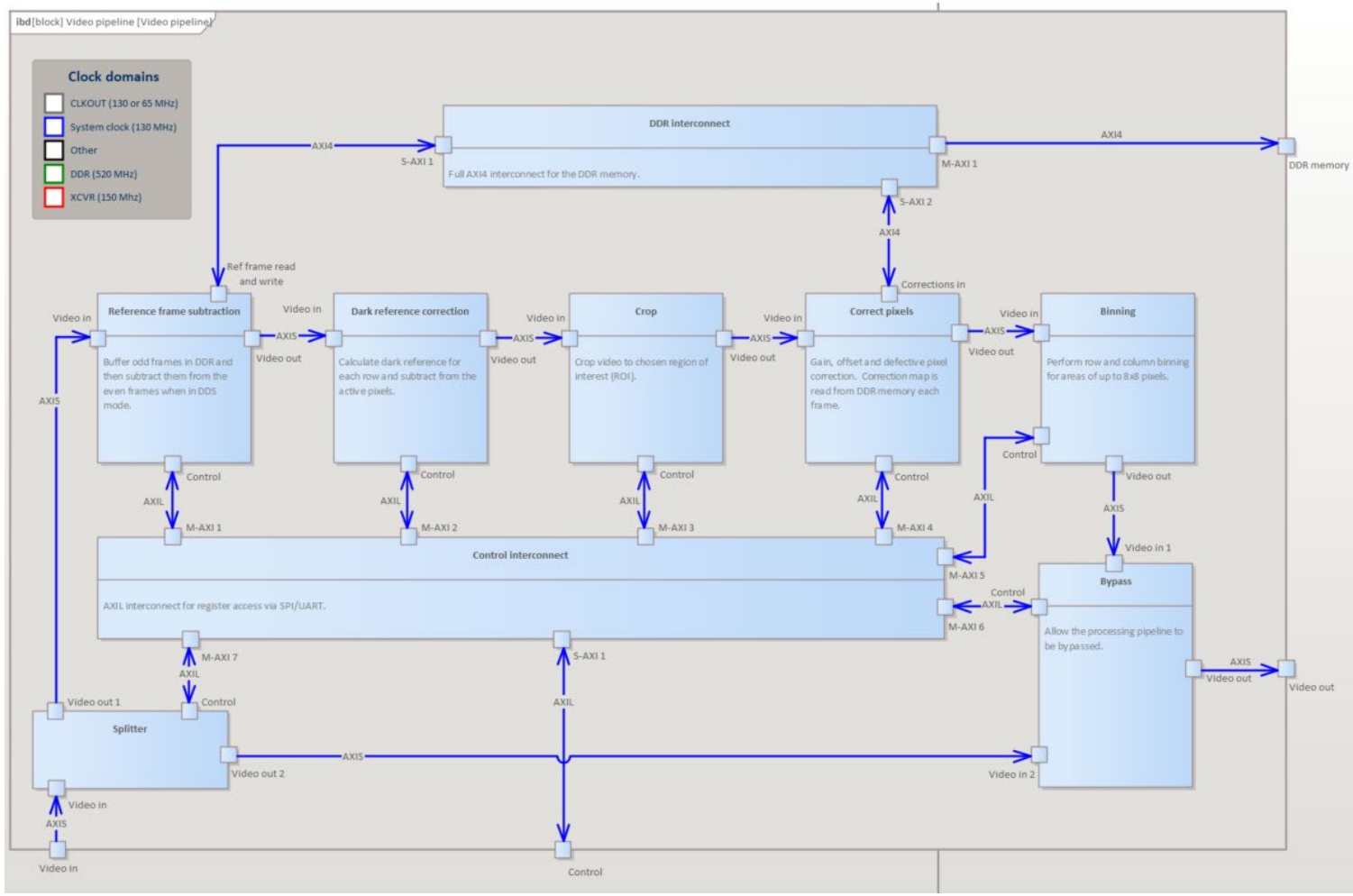
Interface aware

AXI4
AXILite
Serdes
Custom – Logic

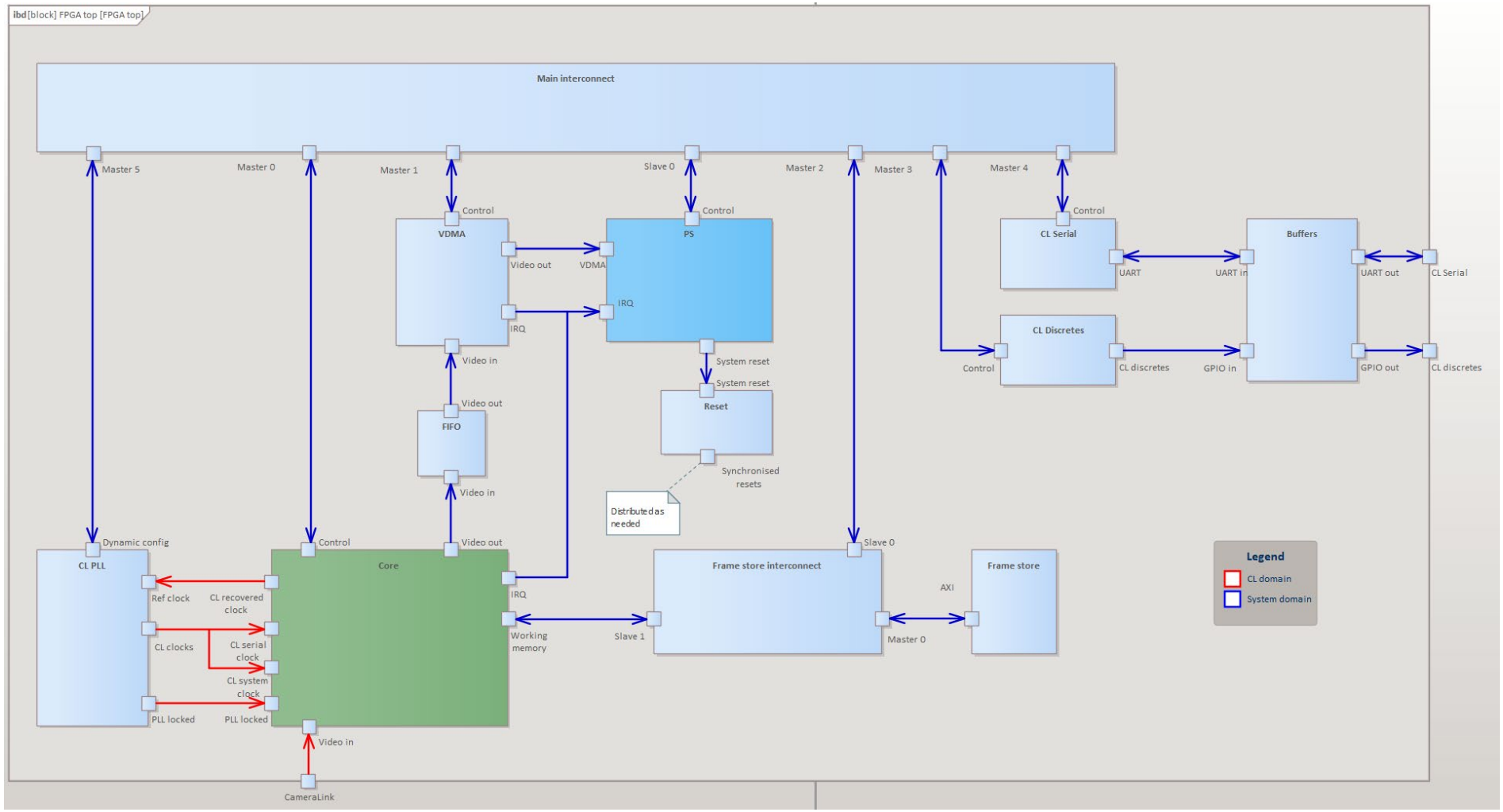
Define in interface diagram



FPGA Architecture – Project One



FPGA Architecture – Project Two



Code Generation

SysML code generation - XML from EA then Python

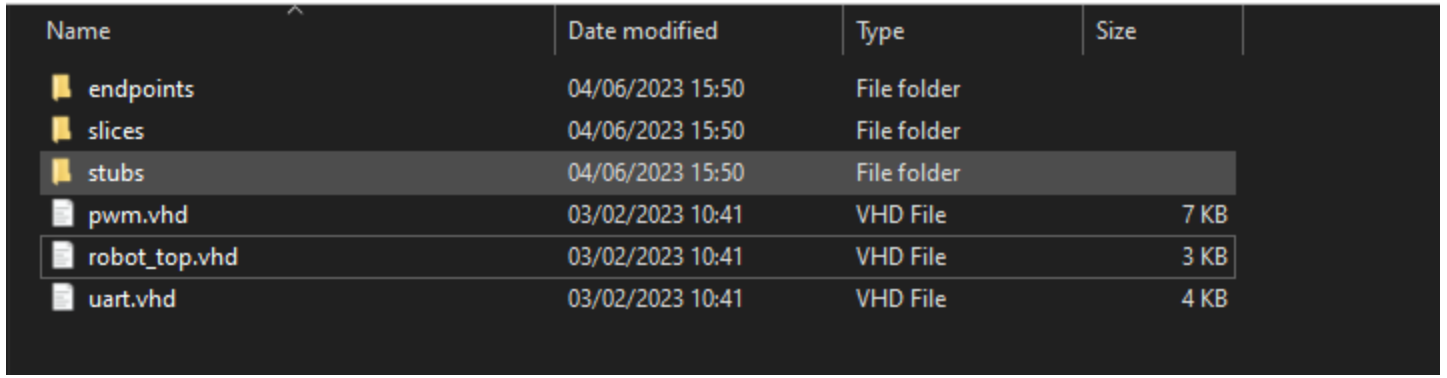
- Generates VHDL
  - All interfaces are created as per definition
  - Registers inserted where defined in model
  - AXI Networks implemented using Adiuvo Interconnects (Technology Ind)
  - Pulls in Adiuvo IP blocks

- Generates ModelSim Compile scripts

- Generate ICD – C, Python, VHDL, SystemVerilog and CSV

Code Generation - HDL

Register End Point IP
AXI(S) Slice IP
Holes to Fill in Simulink
Top Level

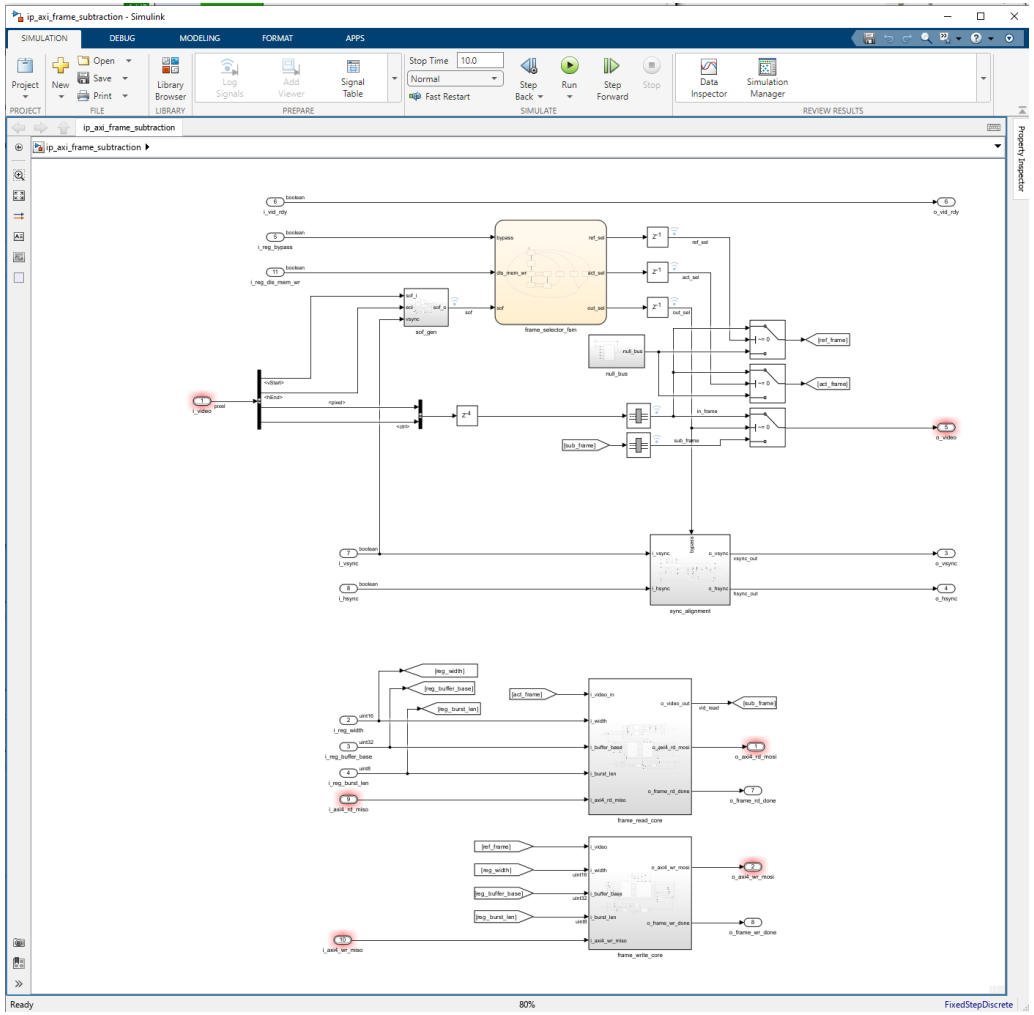


What does the Code Look Like?

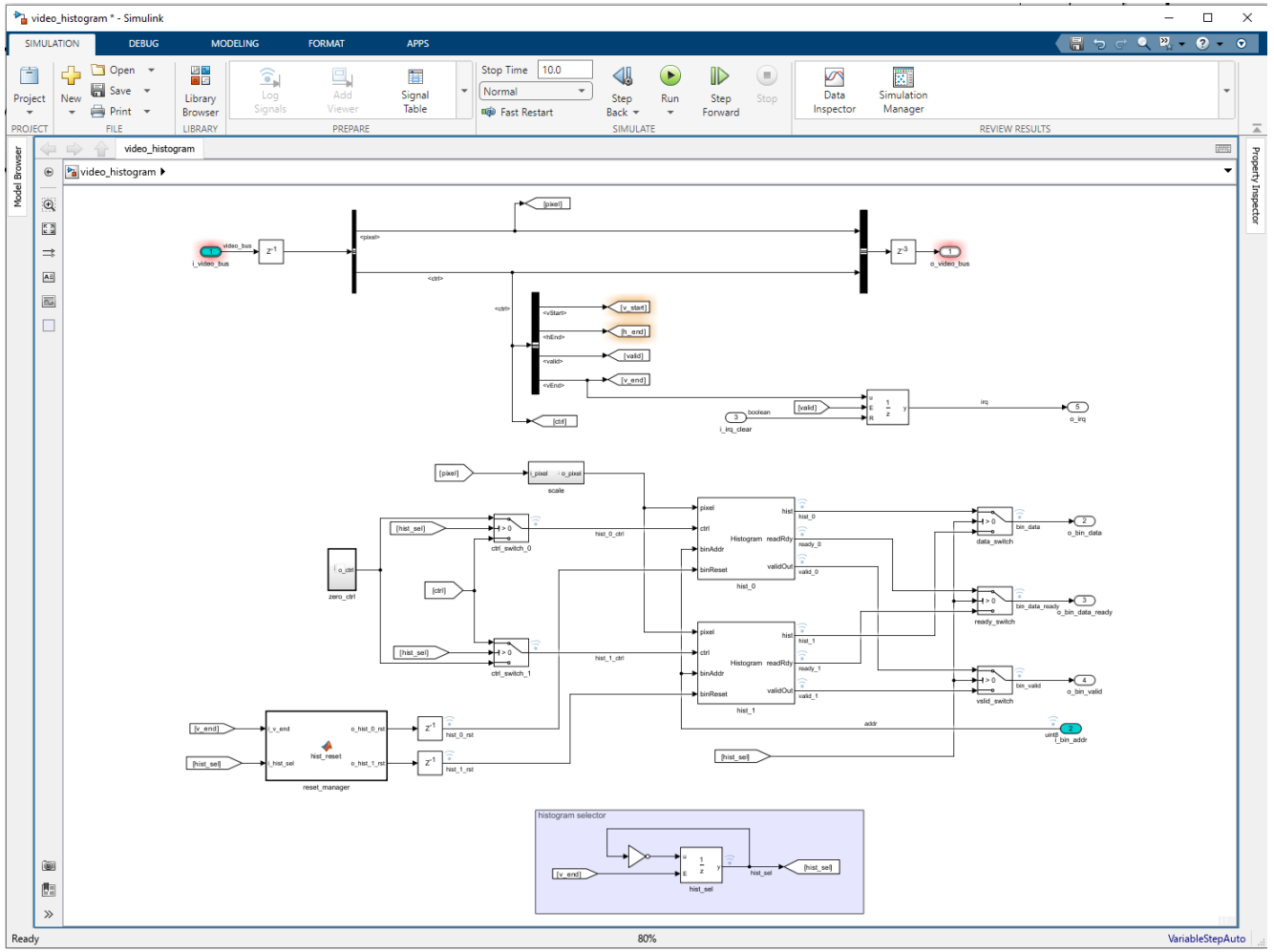
```
architecture rtl of robot_top is

    ------------------------------------------------------------------------
    --Internal signals
    signal s_sys_clk_this : std_ulogic;
    signal s_sys_aresetn_this : std_ulogic;
    signal s_pwm_pwm : std_ulogic_vector(6 downto 0);
    signal s_axil_32_pkg_mosi_maxil_uart : work.axil_32_pkg.r_axil_mosi;
    signal s_axil_32_pkg_miso_maxil_uart : work.axil_32_pkg.r_axil_miso;

    signal s_rx_uart_this : std_ulogic;
    signal s_tx_uart_this : std_ulogic;

begin

    ------------------------------------------------------------------------
    --Internal signal assignments
    o_pwm0 <= s_pwm_pwm;
    o_tx_uart <= s_tx_uart_this;

    s_sys_clk_this <= i_sys_clk;
    s_sys_aresetn_this <= i_sys_aresetn;
    s_rx_uart_this <= i_rx_uart;

    ------------------------------------------------------------------------
    --pwm instantiation
    u_pwm : entity work.pwm
    port map(
        i_sys_clk => s_sys_clk_this,
        i_sys_aresetn => s_sys_aresetn_this,
        o_pwm => s_pwm_pwm,

        i_axil_32_pkg_mosi_saxil => s_axil_32_pkg_mosi_maxil_uart,
        o_axil_32_pkg_miso_saxil => s_axil_32_pkg_miso_maxil_uart
    );

    ------------------------------------------------------------------------
    --uart instantiation
    u_uart : entity work.uart
    port map(
        i_sys_clk => s_sys_clk_this,
        i_sys_aresetn => s_sys_aresetn_this,

        o_axil_32_pkg_mosi_maxil => s_axil_32_pkg_mosi_maxil_uart,
        i_axil_32_pkg_miso_maxil => s_axil_32_pkg_miso_maxil_uart,

        i_rx_uart => s_rx_uart_this,
        o_tx_uart => s_tx_uart_this
    );

end rtl;
```

```
------------------------------------------------------------------------------
--Entity declaration
entity uart_core is
port (
    i_sys_clk : in std_ulogic;
    i_sys_aresetn : in std_ulogic;
    o_axil_32_pkg_mosi_maxil : out work.axil_32_pkg.r_axil_mosi;
    i_axil_32_pkg_miso_maxil : in work.axil_32_pkg.r_axil_miso;
    i_rx_uart : in std_ulogic;
    o_tx_uart : out std_ulogic
);
end uart_core;

architecture rtl of uart_core is

begin

    ------------------------------------------------------------------------------
    --Dummy signal assignments to outputs - delete as needed
    o_axil_32_pkg_mosi_maxil.s_axi_awaddr    <= (others => '0');
    o_axil_32_pkg_mosi_maxil.s_axi_awprot    <= (others => '0');
    o_axil_32_pkg_mosi_maxil.s_axi_awvalid   <= '0';
    o_axil_32_pkg_mosi_maxil.s_axi_wdata     <= (others => '0');
    o_axil_32_pkg_mosi_maxil.s_axi_wstrb     <= (others => '0');
    o_axil_32_pkg_mosi_maxil.s_axi_wvalid    <= '0';
    o_axil_32_pkg_mosi_maxil.s_axi_bready    <= '0';
    o_axil_32_pkg_mosi_maxil.s_axi_araddr    <= (others => '0');
    o_axil_32_pkg_mosi_maxil.s_axi_arprot    <= (others => '0');
    o_axil_32_pkg_mosi_maxil.s_axi_arvalid   <= '0';
    o_axil_32_pkg_mosi_maxil.s_axi_rready    <= '0';

    o_tx_uart <= '0';

end rtl;
```

Simulink HDL Coder – Frame Subtraction



Simulink HDL Coder – Histogram



What does HDL Code OP look like?

```
c_2_process : PROCESS (i_clk, i_aresetn)
BEGIN
  IF i_aresetn = '0' THEN
    alpha_reg_2 <= (OTHERS => '0');
  ELSIF i_clk'EVENT AND i_clk = '1' THEN
    alpha_reg_2(0) <= ctrl_vStart;
    alpha_reg_2(1) <= alpha_reg_2(0);
  END IF;
END PROCESS c_2_process;

ctrl_vStart_1 <= alpha_reg_2(1);

c_3_process : PROCESS (i_clk, i_aresetn)
BEGIN
  IF i_aresetn = '0' THEN
    alpha_reg_3 <= (OTHERS => '0');
  ELSIF i_clk'EVENT AND i_clk = '1' THEN
    alpha_reg_3(0) <= ctrl_vEnd;
    alpha_reg_3(1) <= alpha_reg_3(0);
  END IF;
END PROCESS c_3_process;

ctrl_vEnd_1 <= alpha_reg_3(1);

c_4_process : PROCESS (i_clk, i_aresetn)
BEGIN
  IF i_aresetn = '0' THEN
    alpha_reg_4 <= (OTHERS => '0');
  ELSIF i_clk'EVENT AND i_clk = '1' THEN
    alpha_reg_4(0) <= ctrl_valid;
    alpha_reg_4(1) <= alpha_reg_4(0);
  END IF;
END PROCESS c_4_process;
```

```
CASE is_frame_selector_fsm IS
  WHEN IN_active_frame =>
    sf_internal_predicateOutput := hdlcoder_to_stdlogic((sof AND bypass) = '1');
    IF sf_internal_predicateOutput = '1' THEN
      is_frame_selector_fsm_next <= IN_bypass_ref;
      ref_sel <= '1';
      act_sel <= '0';
      out_sel <= '1';
    ELSE
      b_sf_internal_predicateOutput := hdlcoder_to_stdlogic((sof AND ( NOT bypass)) = '1');
      IF b_sf_internal_predicateOutput = '1' THEN
        is_frame_selector_fsm_next <= IN_ref_frame;
        ref_sel <= '1';
        act_sel <= '0';
        out_sel <= '0';
      ELSE
        ref_sel <= '0';
        act_sel <= '1';
        out_sel <= '0';
      END IF;
    END IF;
  WHEN IN_bypass_act =>
    sf_internal_predicateoutput_0 := hdlcoder_to_stdlogic((sof AND bypass) = '1');
    IF sf_internal_predicateoutput_0 = '1' THEN
      is_frame_selector_fsm_next <= IN_bypass_ref;
      ref_sel <= '1';
      act_sel <= '0';
      out_sel <= '1';
    ELSE
      b_sf_internal_predicateoutput_0 := hdlcoder_to_stdlogic((sof AND ( NOT bypass)) = '1');
      IF b_sf_internal_predicateoutput_0 = '1' THEN
        is_frame_selector_fsm_next <= IN_ref_frame;
        ref_sel <= '1';
        act_sel <= '0';
        out_sel <= '0';
      ELSE
        ref_sel <= '1';
        act_sel <= '0';
        out_sel <= '1';
      END IF;
    END IF;
```

ADIUVO
ENGINEERING AND TRAINING, LTD.

Conclusion & Wrap Up

- Model Based approach is helping on time delivery & quality of design
    - Major benefit is functionality designed and verified in single environment of Simulink.
- Code is portable frees us from Vendor – Which HLS would not do.
- Need to decide on what we do with Adiuvo FPGA Architect tool
    - Develop, Open Source, Keep as internal advantage!
- Question?

ADIUVO

ENGINEERING AND TRAINING, LTD.

www.adiuvoengineering.com

info@adiuvoengineering.com