

ELD Lab 7

Exploring ARM Processor

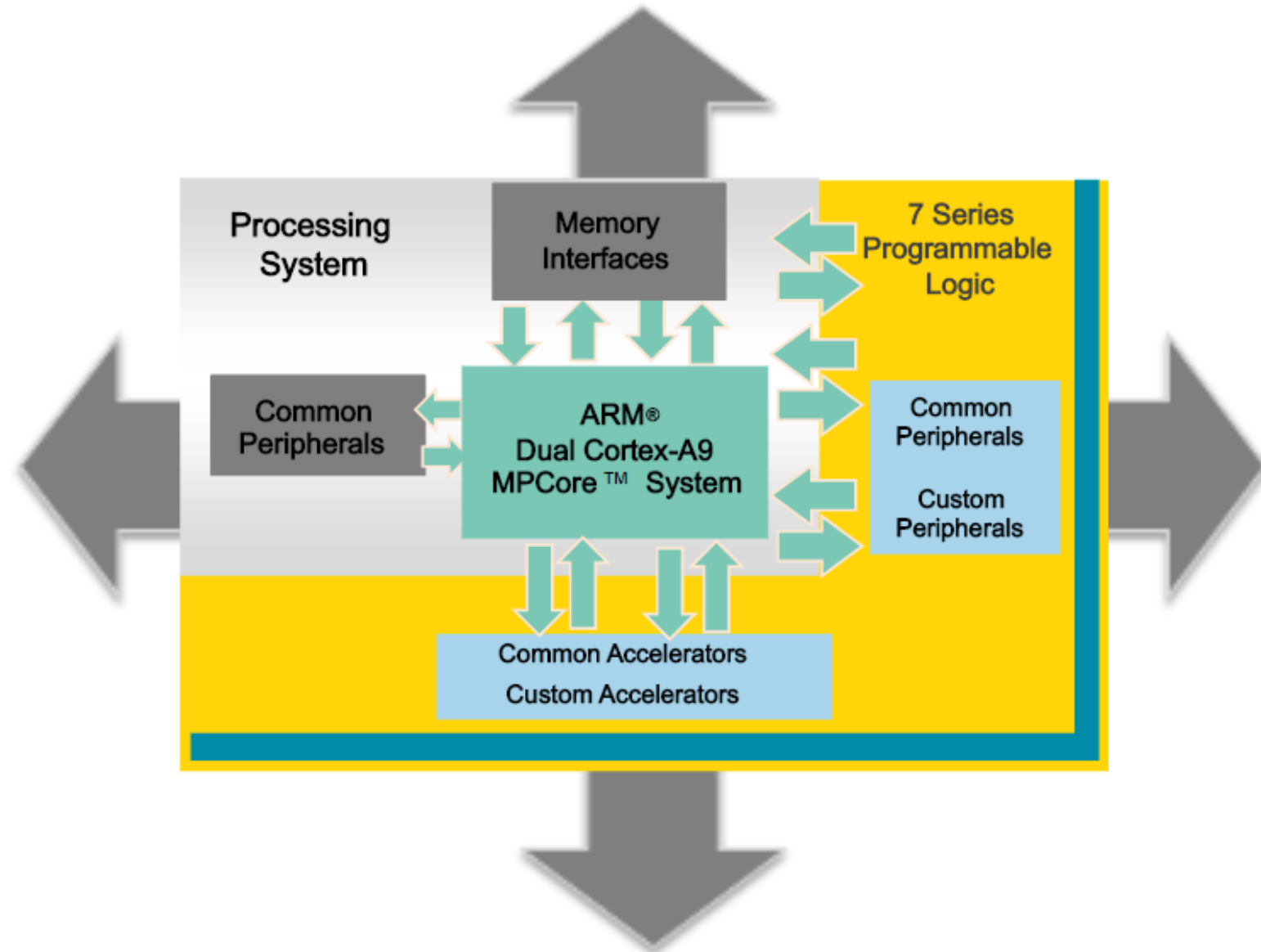
Objective

- Run simple programs on ARM Cortex A9 processor of Zynq SoC
- Homework: Implement 2x2 matrix multiplication on ARM Processor

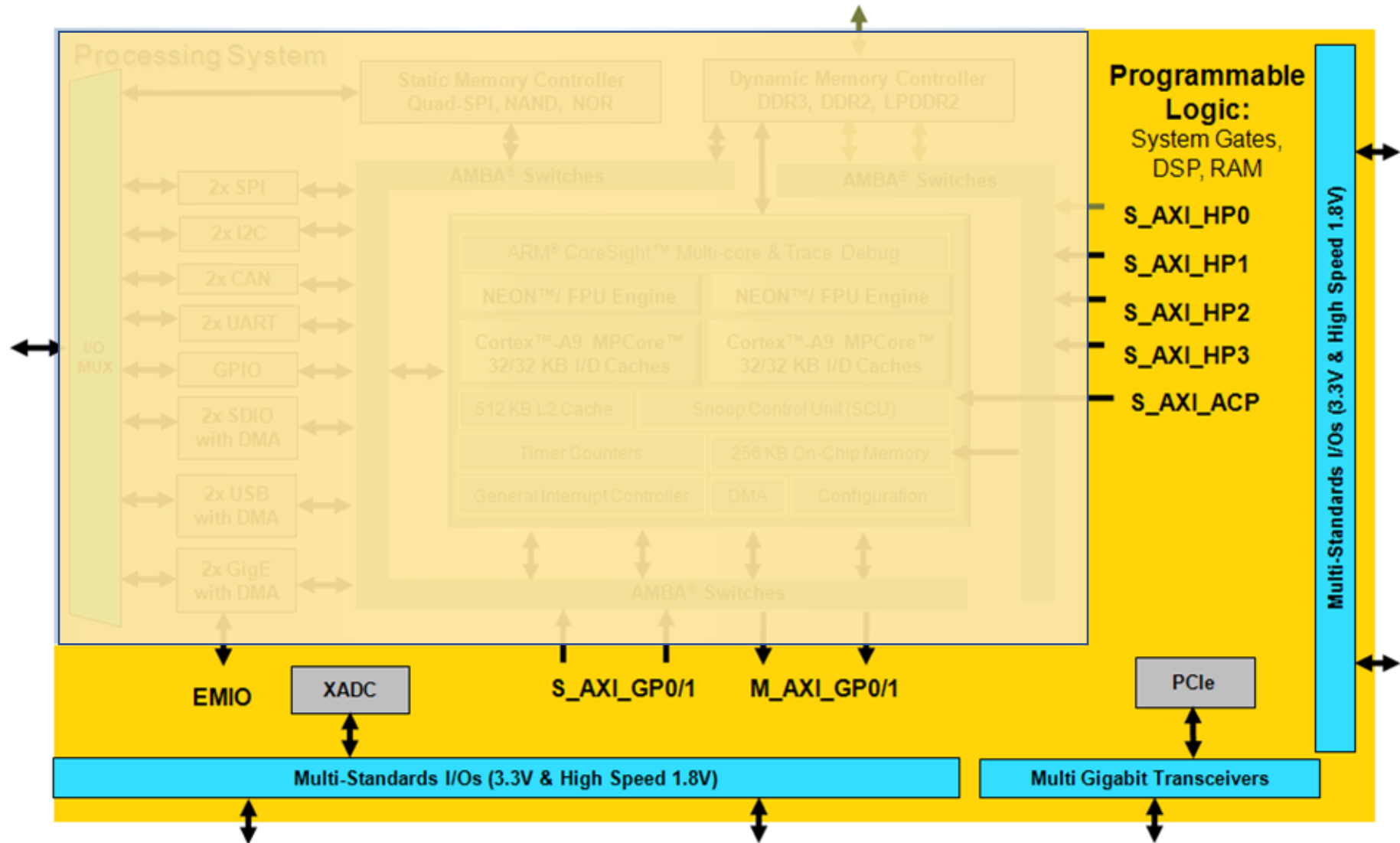
Theory

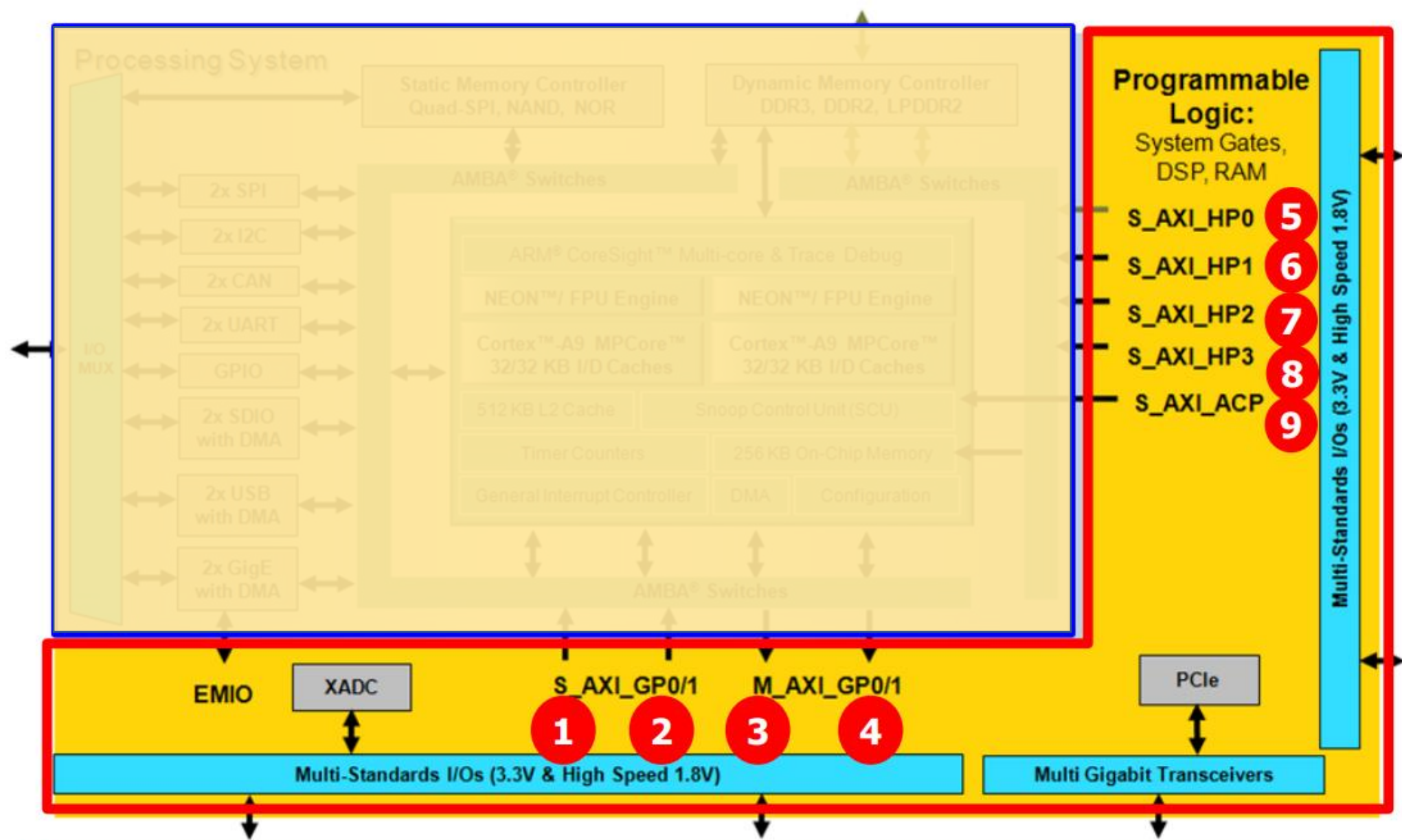
Zynq SoC

Zynq Architecture: PS and PL



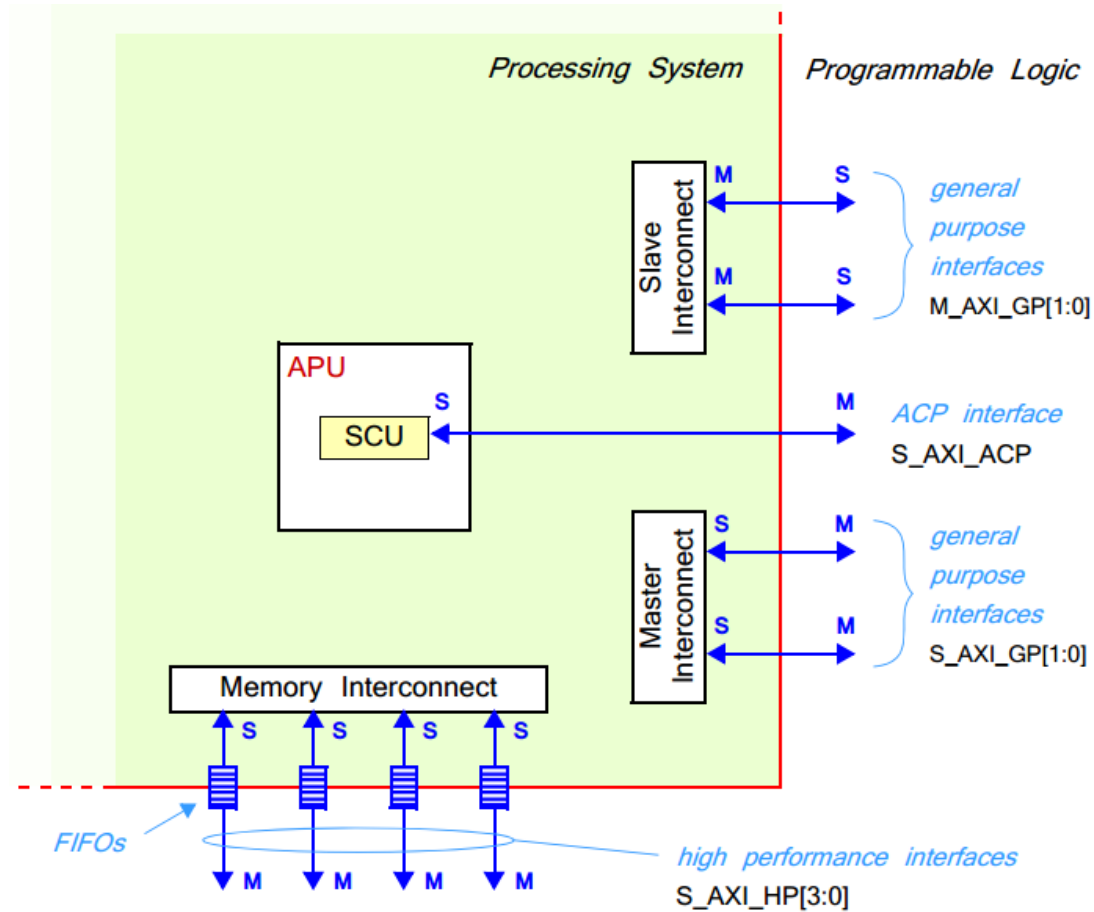
Zynq Architecture: PS and PL



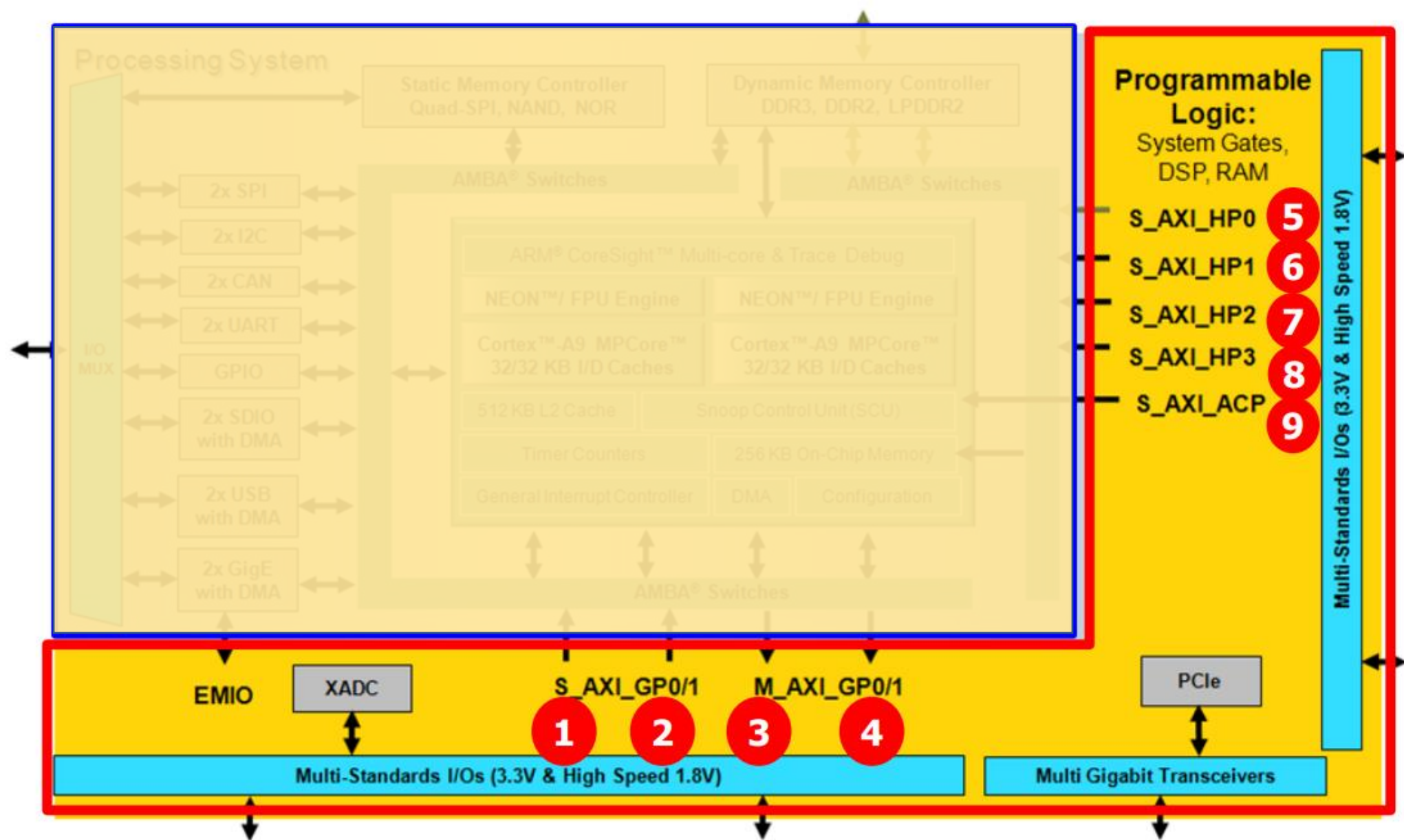


9 Independent PS-to-PL Interface ~100Gbps of Bandwidth

Zynq PS-PL Interface

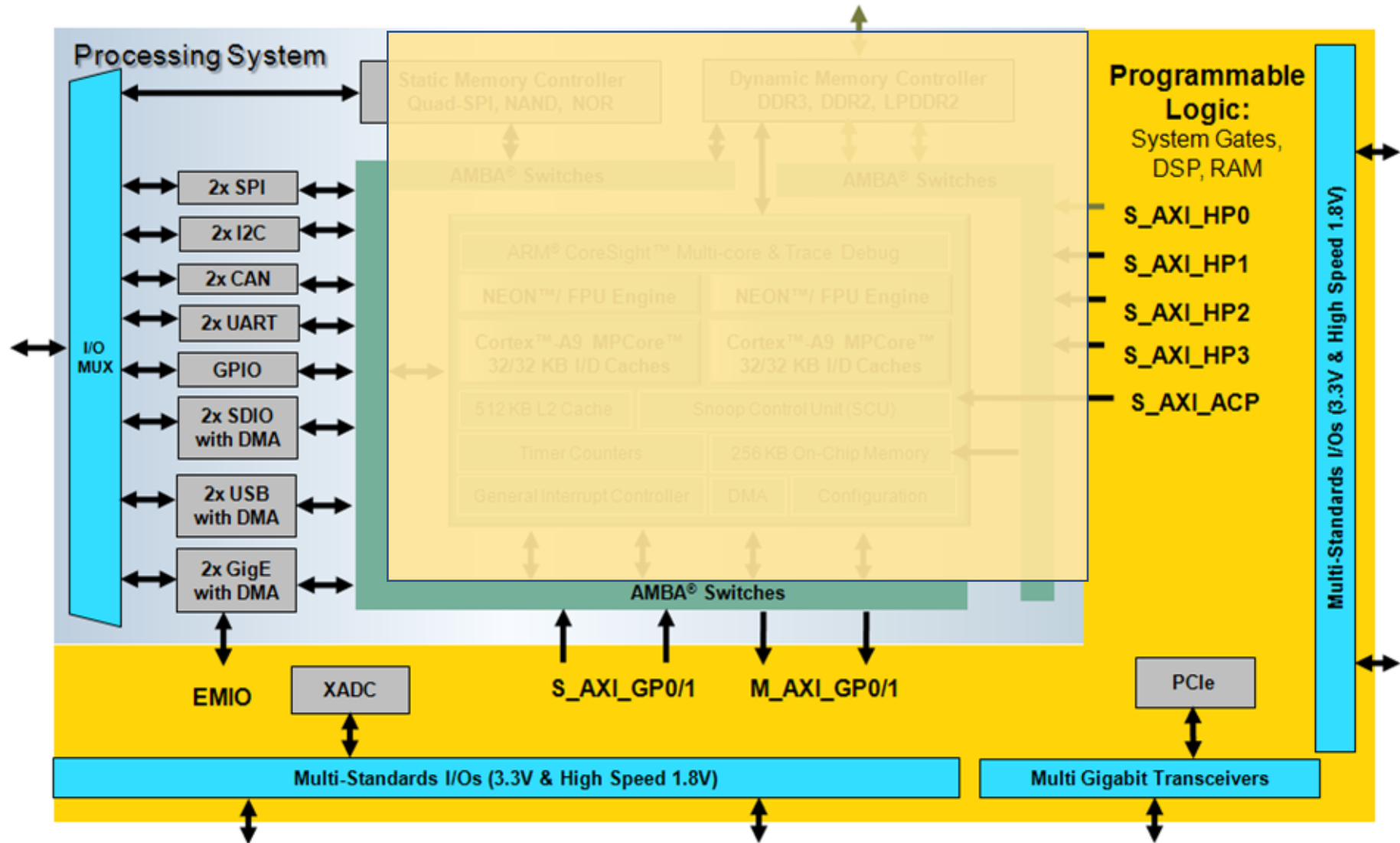


Interface Name	Interface Description	Master	Slave
M_AXI_GP0	General Purpose (AXI_GP)	PS	PL
M_AXI_GP1		PS	PL
S_AXI_GP0	General Purpose (AXI_GP)	PL	PS
S_AXI_GP1		PL	PS
S_AXI_ACP	Accelerator Coherency Port, cache-coherent transaction (ACP)	PL	PS
S_AXI_HP0	High Performance ports (AXI_HP) with read/write FIFOs and two dedicated memory ports on DDR controller and a path to the OCM. The AXI_HP interfaces are known also as AFI.	PL	PS
S_AXI_HP1		PL	PS
S_AXI_HP2		PL	PS
S_AXI_HP3		PL	PS

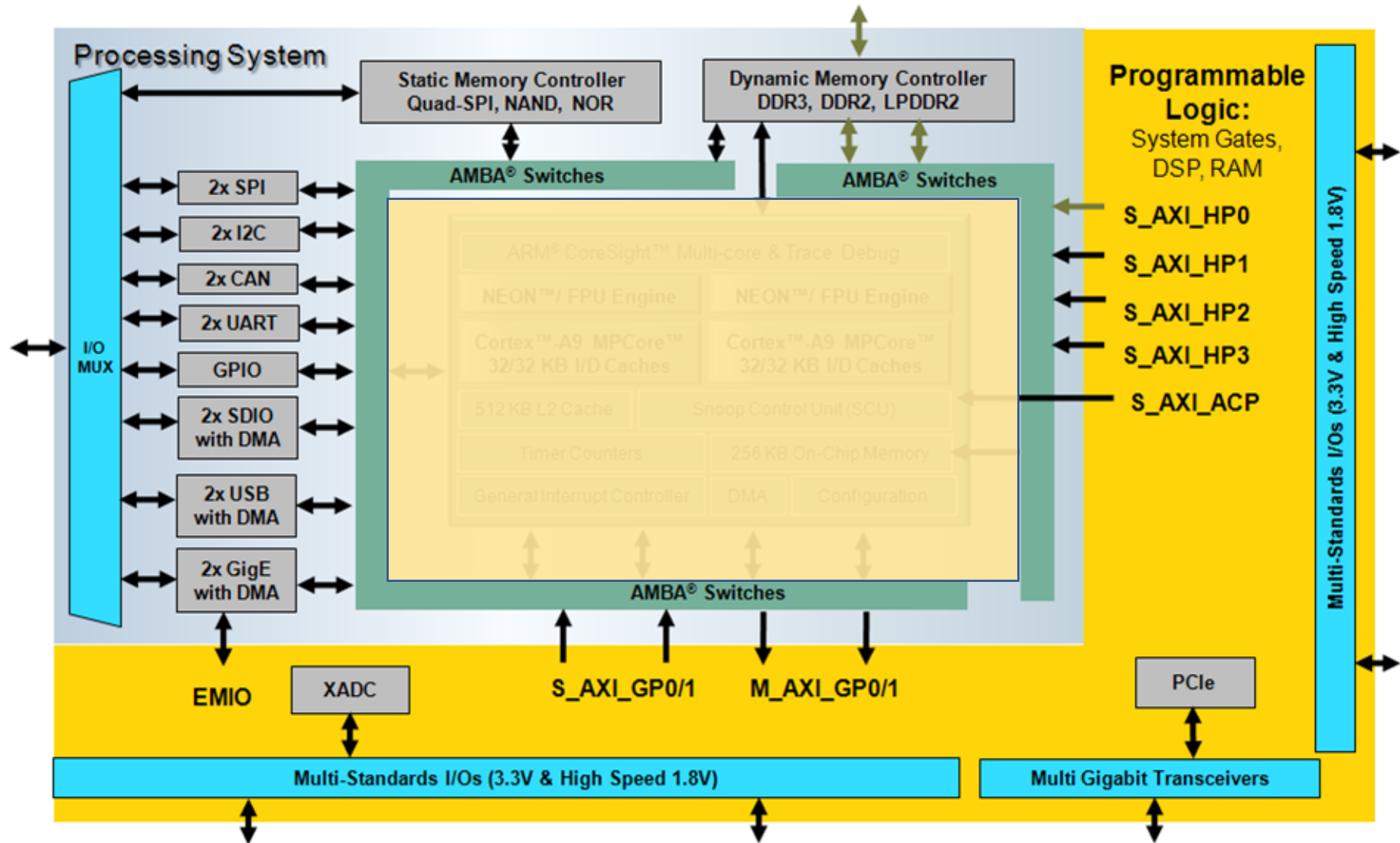


9 Independent PS-to-PL Interface ~100Gbps of Bandwidth

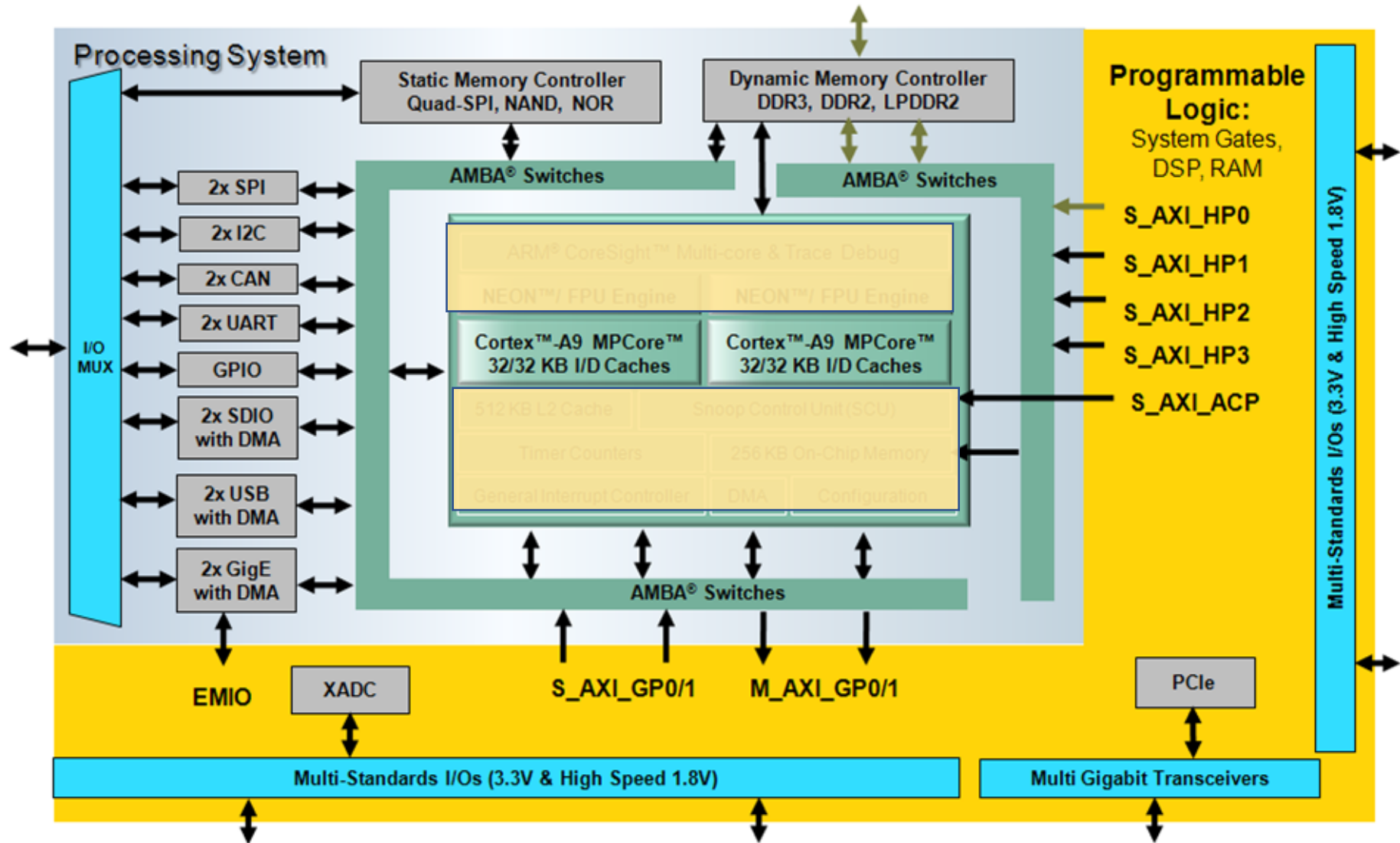
Zynq Architecture: PS and PL



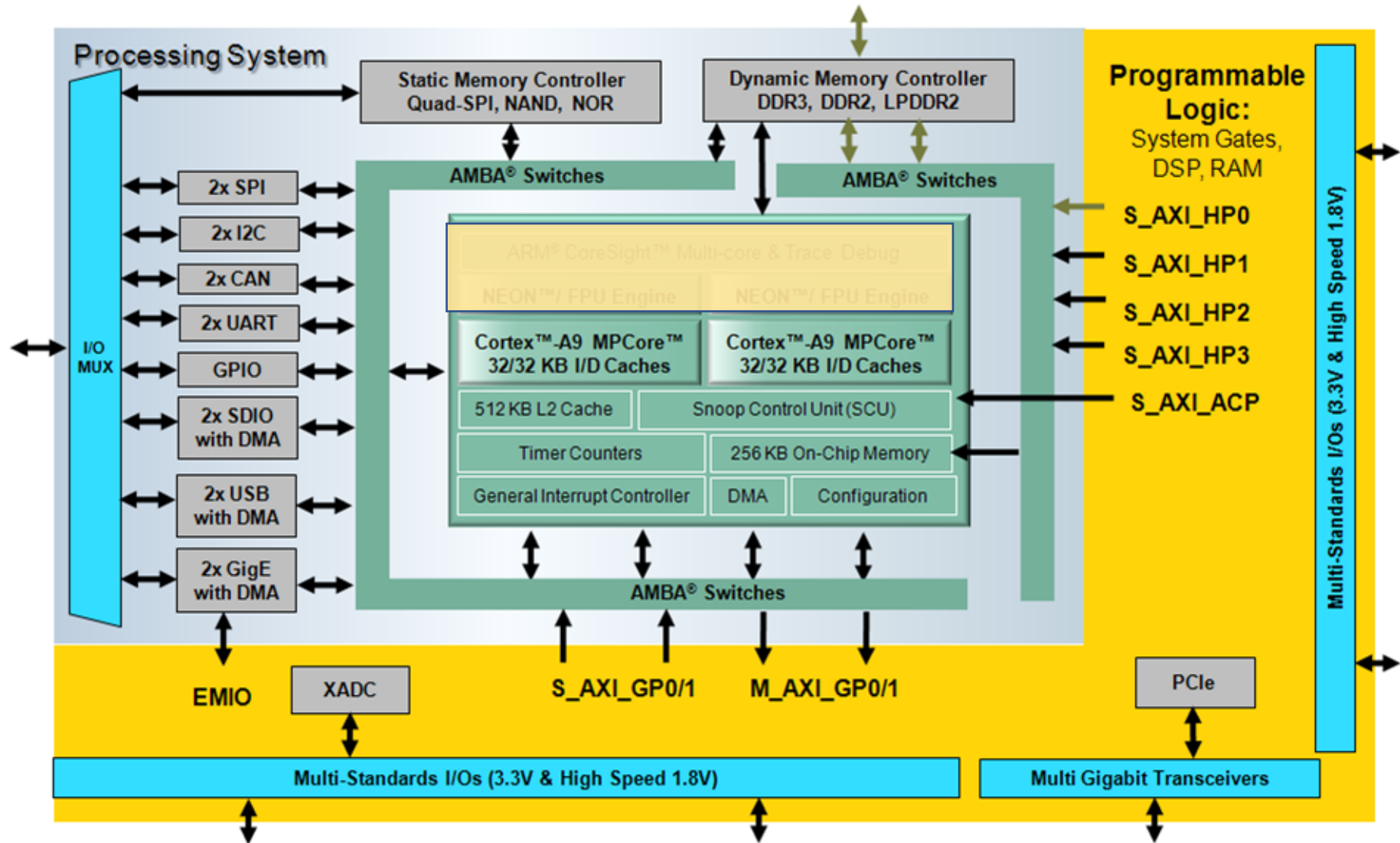
Zynq Architecture: PS and PL



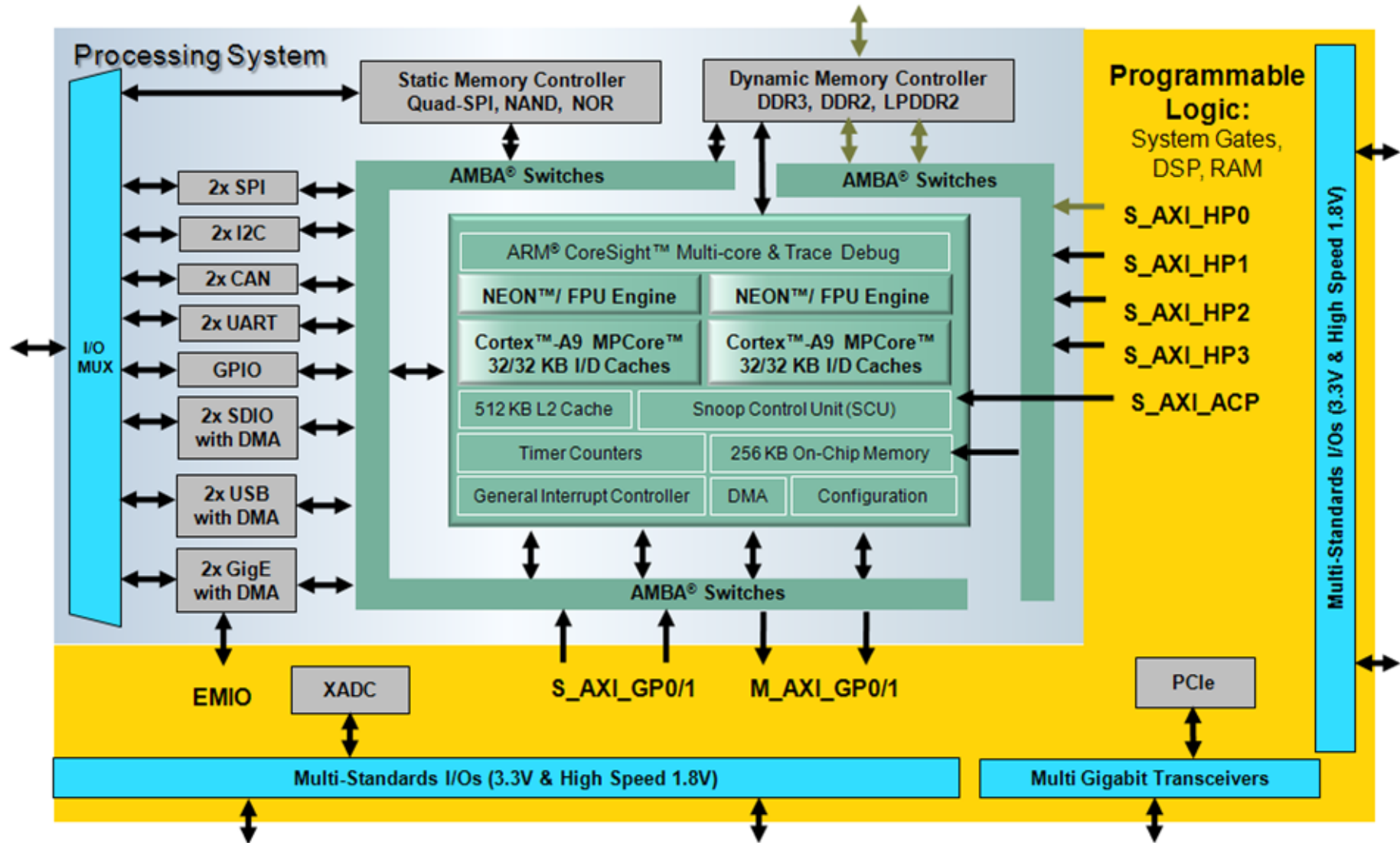
Zynq Architecture: PS and PL



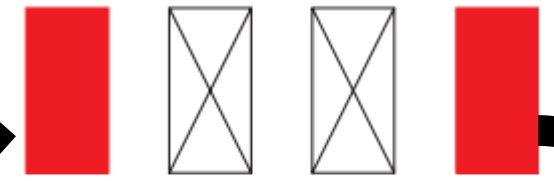
Zynq Architecture: PS and PL



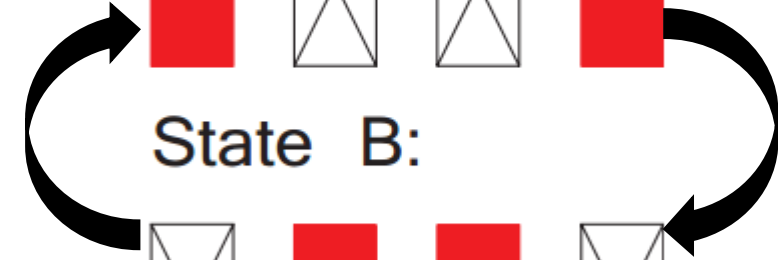
Zynq Architecture: PS and PL

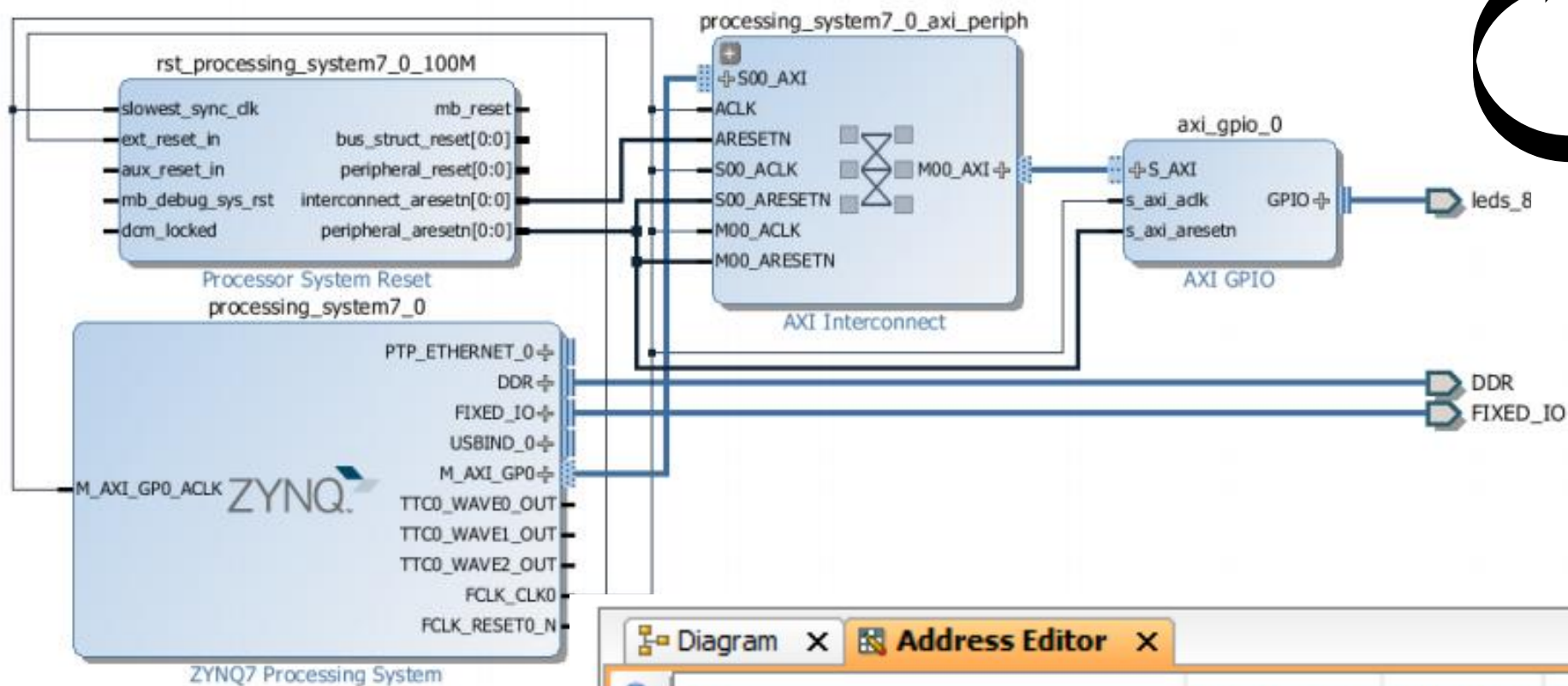


State A:

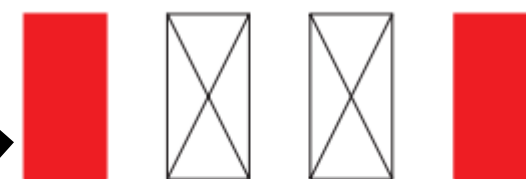


State B:





State A:



State B:



Diagram X Address Editor X

Cell	Interface Pin	Base Name	Offset Address	Range	High Address
processing_system7_0					
Data (32 address bits : 4G)					
axi_gpio_0	S_AXI	Reg	0x41200000	64K	0x4120FFFF

Peripherals

Application Processing Unit (APU)

Processors + RAM

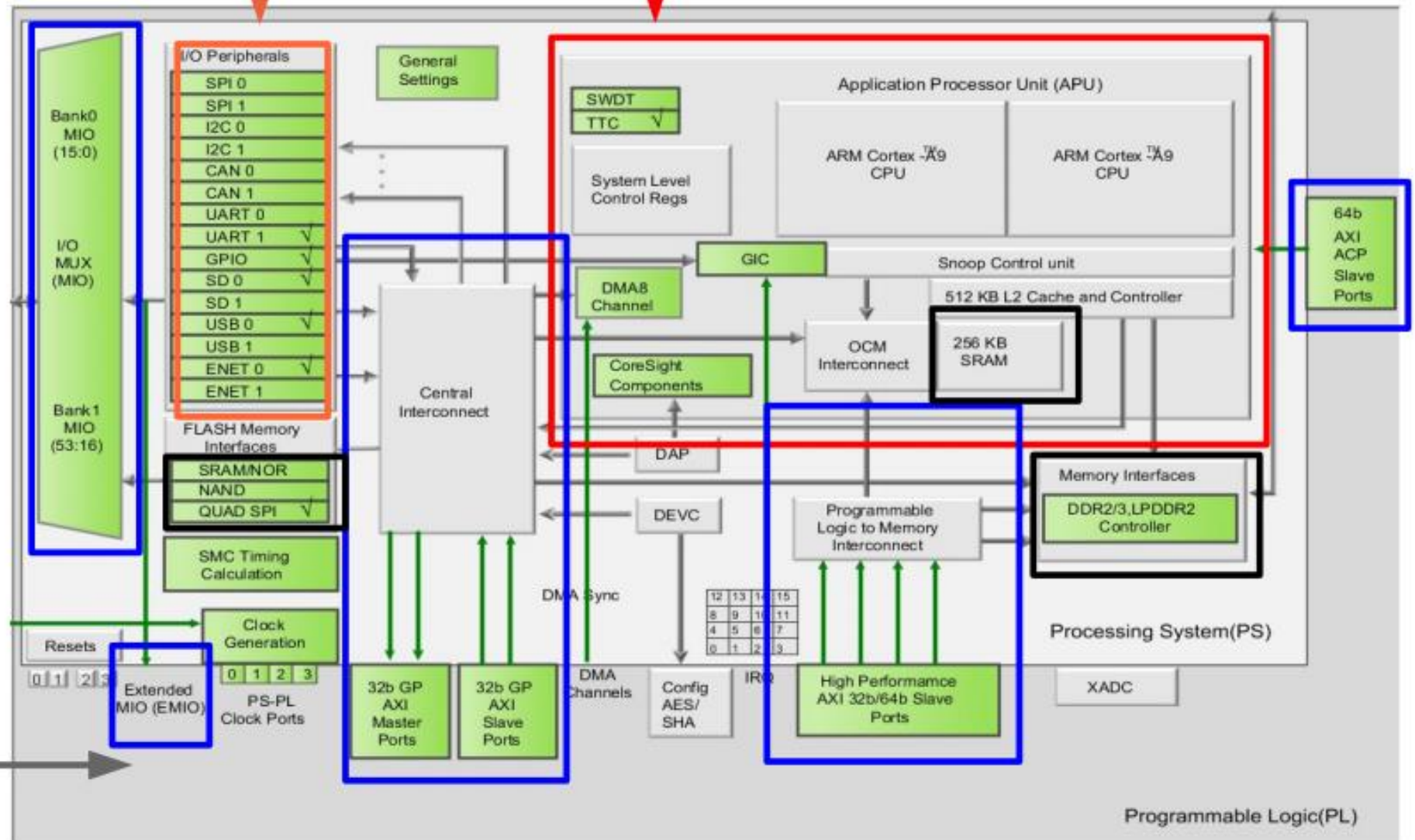
Interconnections

AXI
Peripherals
Memory
User defined

Memory

On-chip
DDR
Flash

Programmable Logic



Lab

Vivado: Empty Project

- Use either Zedboard or Zybo
- Create block diagram and add Zynq IP

PROJECT MANAGER

Settings

Add Sources

Language Templates

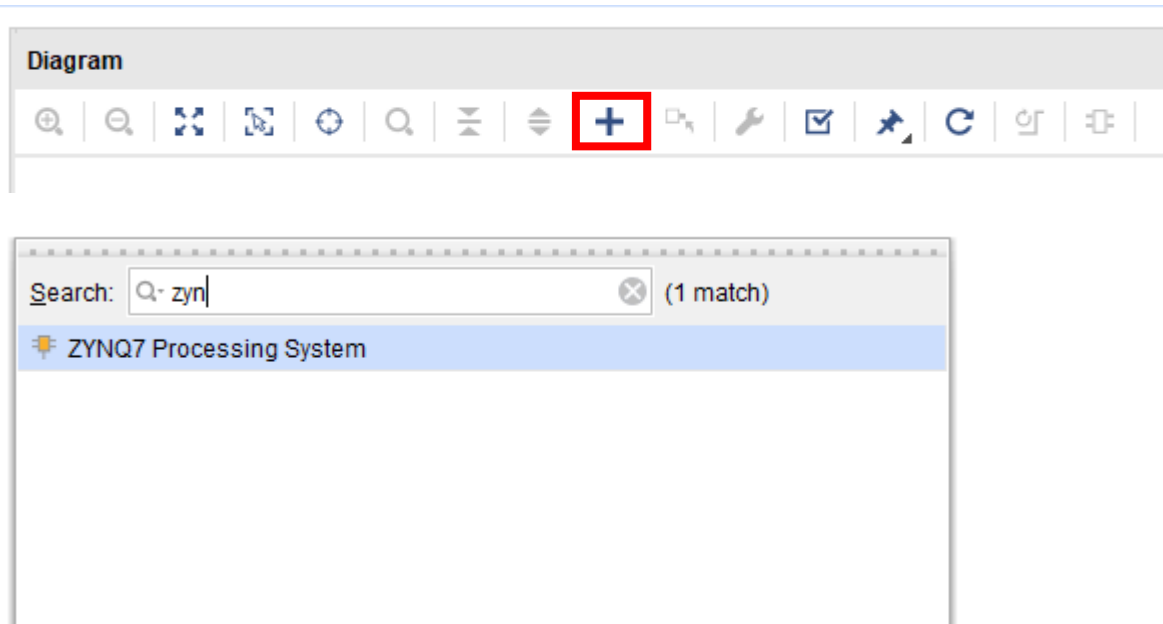
IP Catalog

IP INTEGRATOR

Create Block Design

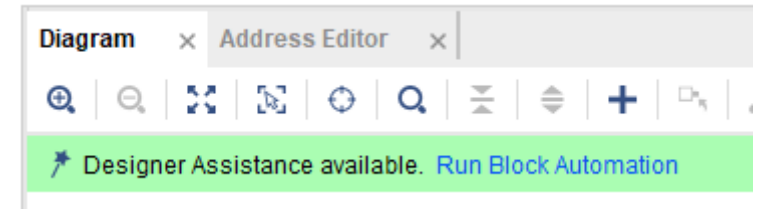
Open Block Design

Generate Block Design



Zynq IP Configuration

- Click on Run Block Automation and Select OK
- Remove following ports by configuring the Zynq IP: GP Master Interface



ZYNQ7 Processing System (5.5)

[Documentation](#) [Presets](#) [IP Location](#) [Import XPS Settings](#)

Page Navigator —

- Zynq Block Design
- PS-PL Configuration**
- Peripheral I/O Pins
- MIO Configuration
- Clock Configuration
- DDR Configuration

PS-PL Configuration [Sumn](#)

Search:

Name	Select	Description
> General		
▼ AXI Non Secure Enablement	0	Enable AXI Non Secure Transaction
▼ GP Master AXI Interface		
> M AXI GP0 interface	<input type="checkbox"/>	Enables General purpose AXI master interface 0
> M AXI GP1 interface	<input type="checkbox"/>	Enables General purpose AXI master interface 1

Zynq IP Configuration

- Remove following ports by configuring the Zynq IP: Ethernet0, USB0, SD0, TTC0, GPIO MIO

The screenshot shows the 'Peripheral I/O Pins' configuration window for a ZYNQ7 Processing System (5.5). The window is divided into a 'Page Navigator' on the left and a main configuration area on the right. The 'Page Navigator' lists various configuration sections: Zynq Block Design, PS-PL Configuration, Peripheral I/O Pins (selected), MIO Configuration, Clock Configuration, DDR Configuration, SMC Timing Calculation, and Interrupts. The 'Peripheral I/O Pins' section is further divided into 'Peripherals' and 'GPIO MIO'. Under 'Peripherals', several options are listed with checkboxes: Ethernet 0, Ethernet 1, USB 0, USB 1, SD 0, SD 1, SPI 0, SPI 1, UART 0, **UART 1** (checked), I2C 0, I2C 1, CAN 0, CAN 1, TTC 0, TTC 1, SWDT, PJTAG, TPIU, **GPIO MIO** (checked), and **GPIO EMIO** (checked). The main configuration area displays a grid of pins for Bank 0 (LVCMOS 3.3V) and Bank 1 (LVCMOS 1.8V). The grid shows the physical location of various peripherals. For example, Ethernet 0 is located at pins 16-24 of Bank 1. SD0 is at pins 16-17 of Bank 1. SD1 is at pins 10-11 of Bank 0 and pins 23-24 of Bank 1. SPI0 is at pins 17-18 of Bank 1. SPI1 is at pins 11-12 of Bank 0 and pins 21-22 of Bank 1. UART0 is at pins 10-11 of Bank 0 and pins 23-24 of Bank 1. UART1 is at pins 8-9 of Bank 0 and pins 21-22 of Bank 1. I2C0 is at pins 10-11 of Bank 0 and pins 23-24 of Bank 1. I2C1 is at pins 12-13 of Bank 0 and pins 21-22 of Bank 1. CAN0 is at pins 10-11 of Bank 0 and pins 23-24 of Bank 1. CAN1 is at pins 12-13 of Bank 0 and pins 21-22 of Bank 1. TTC0 is at pins 10-11 of Bank 0 and pins 23-24 of Bank 1. TTC1 is at pins 12-13 of Bank 0 and pins 21-22 of Bank 1. SWDT is at pins 10-11 of Bank 0 and pins 23-24 of Bank 1. PJTAG is at pins 10-11 of Bank 0 and pins 23-24 of Bank 1. TPIU is at pins 10-11 of Bank 0 and pins 23-24 of Bank 1. GPIO MIO is at pins 10-11 of Bank 0 and pins 23-24 of Bank 1. GPIO EMIO is at pins 10-11 of Bank 0 and pins 23-24 of Bank 1. The grid also shows the location of various other peripherals like Ethernet 1, USB 0, USB 1, SD 0, SD 1, SPI 0, SPI 1, UART 0, UART 1, I2C 0, I2C 1, CAN 0, CAN 1, TTC 0, TTC 1, SWDT, PJTAG, TPIU, and GPIO MIO. At the bottom right, there are 'OK' and 'Cancel' buttons.

Zynq IP Configuration

- Remove following ports by configuring the Zynq IP: Fabric Clock and reset

ZYNQ7 Processing System (5.5)

Documentation Presets IP Location Import XPS Settings

Page Navigator

- Zynq Block Design
- PS-PL Configuration
- Peripheral I/O Pins
- MIO Configuration
- Clock Configuration**
- DDR Configuration
- SMC Timing Calculation
- Interrupts

Clock Configuration

Basic Clocking Advanced Clocking

Input Frequency (MHz) 33.333333 CPU C

Search: Q-

Component	Clock Source	F
> Processor/Memory Clocks		
> IO Peripheral Clocks		
> PL Fabric Clocks		
<input checked="" type="checkbox"/> FCLK_CLK0	IO PLL	1
<input type="checkbox"/> FCLK_CLK1	IO PLL	5
<input type="checkbox"/> FCLK_CLK2	IO PLL	5
<input type="checkbox"/> FCLK_CLK3	IO PLL	5
> System Debug Clocks		
> Timers		

ZYNQ7 Processing System (5.5)

Documentation Presets IP Location Import XPS Settings

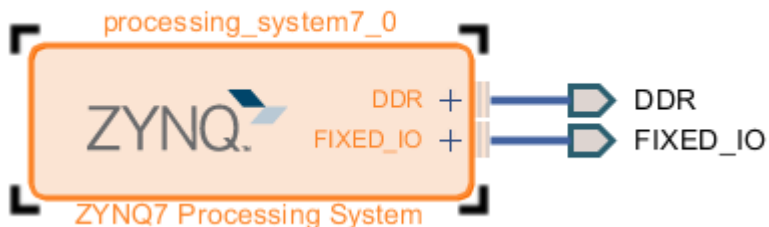
Page Navigator

- Zynq Block Design
- PS-PL Configuration**
- Peripheral I/O Pins
- MIO Configuration
- Clock Configuration
- DDR Configuration
- SMC Timing Calculation
- Interrupts

PS-PL Configuration

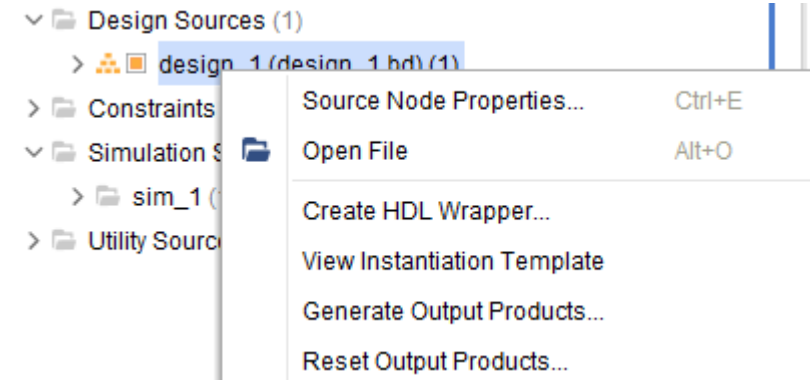
Search: Q-

Name	Select	Description
UART0 Baud Rate	115200	Baud rate is generated with internally fixed UART Ref Clock Freq=100MHz
UART1 Baud Rate	115200	Baud rate is generated with internally fixed UART Ref Clock Freq=100MHz
PL AXI idle Port	<input type="checkbox"/>	Enables idle AXI signal to the PS used to indicate that there are no outstanding transactions
DDR ARB bypass Port	<input type="checkbox"/>	Enables DDR urgent/arb signal used to signal a critical memory starvation
PS-PL Debug interface	<input type="checkbox"/>	Enables PL debug signals to PS and vice-versa
FTM Trace data interface	<input type="checkbox"/>	Enables FTM Trace AXI stream interface used to capture data from PL to PS
FTM Trace buffer	0	Generates a FIFO to hold trace data
FTM Data edge detector	0	Stores trace data in the FIFO when the data changes as marked by edge
FTM Trace buffer FIFO size	128	FTM Trace buffer FIFO size
FTM Trace buffer clock delay	12	Number of clock cycles interval for a trace data output from FIFO being processed
Include ACP transaction checker	<input type="checkbox"/>	Enables ACP transaction checker.
Trace data/control signal pipeline width	8	Enables configurable number of pipeline stages on the TRACE DATA/CONTROL
Power-on reset(POR) 4k timer	<input type="checkbox"/>	Enables power-on reset(POR) 4k timer. By default, 64k timer is used.
Processor event interface	<input type="checkbox"/>	Enables event bus which provides a low-latency and direct mechanism for
> Address Editor		
> Enable Clock Triggers		
> Enable Clock Resets		
<input checked="" type="checkbox"/> FCLK_RESET0_N	<input type="checkbox"/>	Enables general purpose reset signal 0 for PL logic
<input type="checkbox"/> FCLK_RESET1_N	<input type="checkbox"/>	Enables general purpose reset signal 1 for PL logic



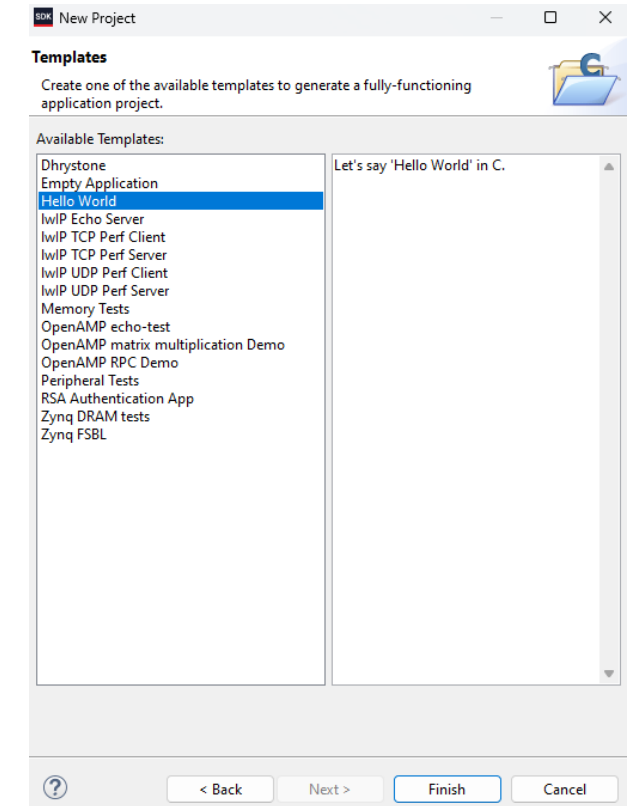
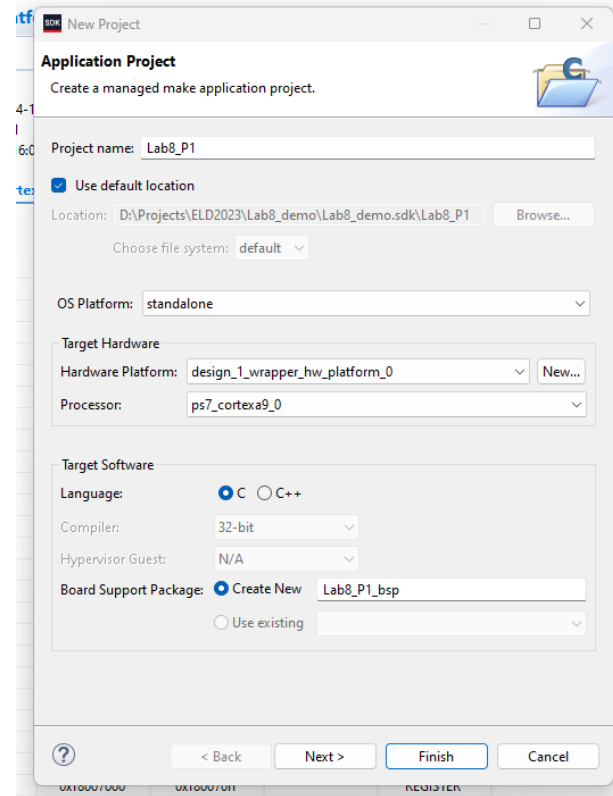
Export the Data and Launch SDK

- Create HDL Wrapper and Generate Output Products
- File -> Export Hardware
- File -> Launch SDK



SDK

- Create New Application Project and select Hello World



Application Code

- Update the code

```
#include <stdio.h>
#include "platform.h"
#include "xil_printf.h"

int main()
{
    init_platform();

    print("Hello World\n\r");

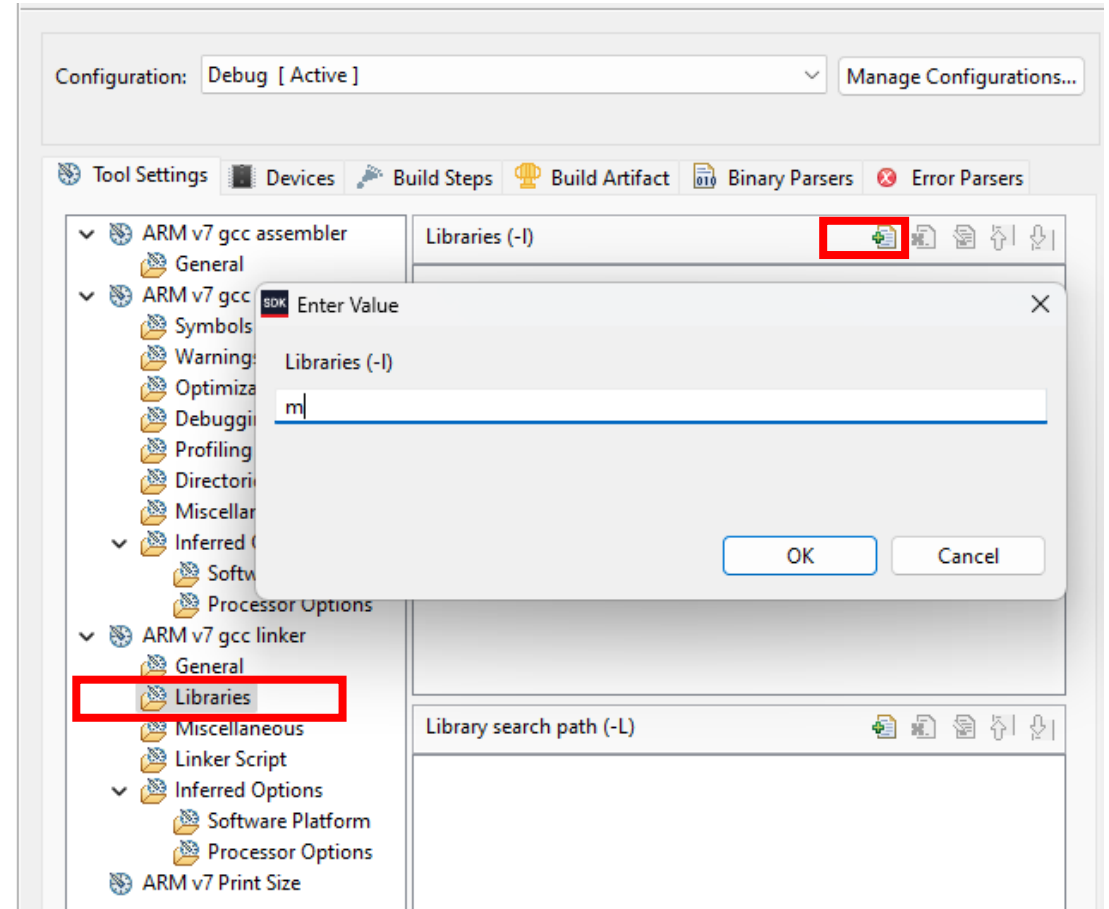
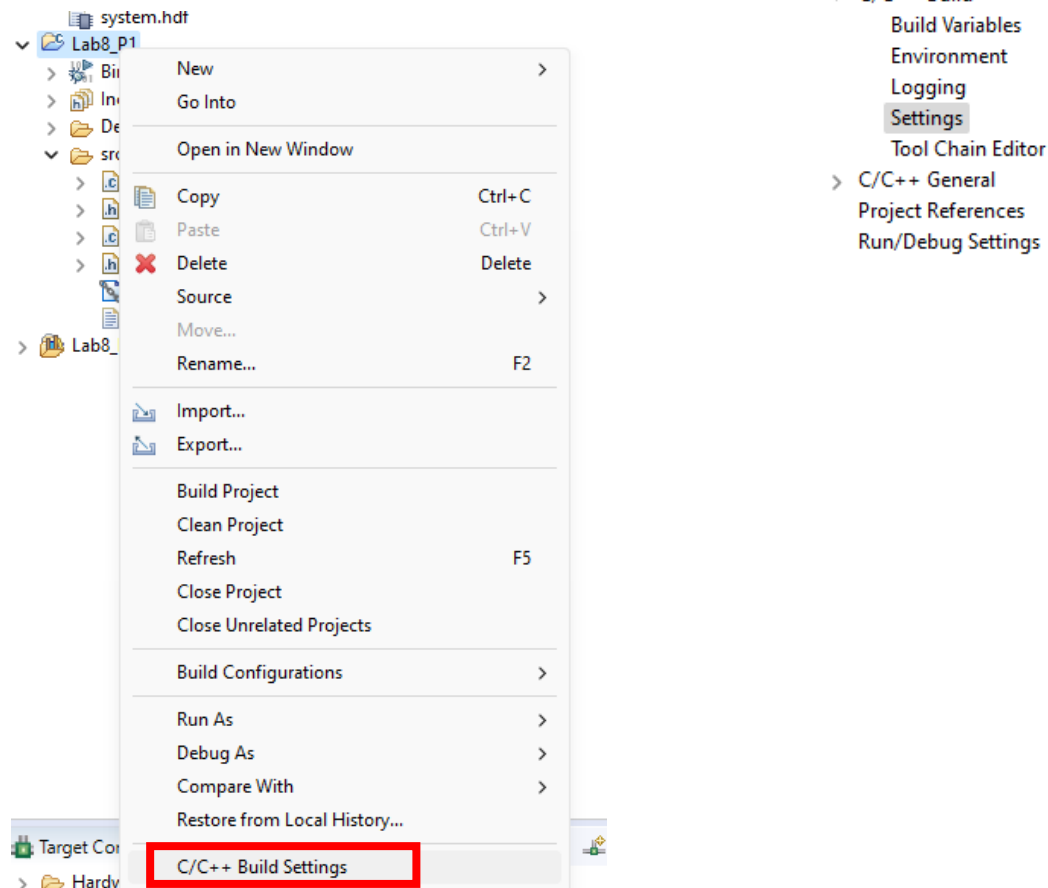
    int x=4;
    int n=200;
    int t=2;
    float res1=x/t;
    float res2=(2*log(n))/t;
    float res3=sqrt(res2);
    float result=res1+res3;

    printf("The expression x/t + sqrt(2*ln(n))/t is being calculated\n");
    printf("x/t resulted in %f\n",res1);
    printf("2*ln(n)/t resulted in %f\n",res2);
    printf("sqrt(2*ln(n))/t resulted in %f\n",res3);
    printf("The final result of x/t + sqrt(2*ln(n))/t is %f\n",result);

    cleanup_platform();
    return 0;
}
```

Application Code

- Add Math library in SDK



Application Code

- Update the system.mss stdin and stdout

Board Support Package Settings

Control various settings of your Board Support Package.

- Overview
 - standalone
 - drivers
 - ps7_cortexa9_0

Configuration for OS: standalone

Name	Value	Default	Type	Description
hypervisor_guest	false	false	boolean	Enable hypervisor guest support
lockstep_mode_debug	false	false	boolean	Enable debug logic in non-JTAG
sleep_timer	none	none	peripheral	This parameter is used to select s
stdin	ps7_coresight_comp_0	none	peripheral	stdin peripheral
stdout	ps7_coresight_comp_0	none	peripheral	stdout peripheral
ttc_select_cntr	2	2	enum	Selects the counter to be used in
zynqmp_fsbl_bsp	false	false	boolean	Disable or Enable Optimization f
> microblaze_exceptions	false	false	boolean	Enable MicroBlaze Exceptions
> enable_sw_intrusive_profiling	false	false	boolean	Enable S/W Intrusive Profiling or

- design_1_wrapper_hw_platform_0
 - ps7_init_gpl.c
 - ps7_init_gpl.h
 - ps7_init.c
 - ps7_init.h
 - ps7_init.html
 - ps7_init.tcl
 - system.hdf
- Lab8_P1
 - Binaries
 - Includes
 - Debug
 - src
 - helloworld.c
 - platform_config.h
 - platform.c
 - platform.h
 - lscript.ld
 - Xilinx.spec
- Lab8_P1_bsp
 - BSP Documentation
 - ps7_cortexa9_0
 - Makefile
 - system.mss

Remote Access

The screenshot displays the Xilinx IDE interface. On the left, a project tree shows the structure of the 'design_1_wrapper_hw_platform_0' project, including files like 'ps7_init_gpl.c', 'ps7_init_gpl.h', 'ps7_init.c', 'ps7_init.h', 'ps7_init.html', 'ps7_init.tcl', 'system.hdf', and the 'Lab8_P1' directory. The 'Lab8_P1' directory contains subfolders for 'Binaries', 'Includes', 'Debug', and 'src', as well as files like 'helloworld.c', 'platform_config.h', 'platform.c', 'platform.h', 'lscript.ld', 'Xilinx.spec', 'BSP Documentation', 'ps7_cortexa9_0', 'Makefile', and 'system.mss'.

The main window shows the 'Lab8_P1_bsp Board Support Package' configuration. A 'Target Connection Details' dialog box is open, allowing the user to edit the target connection. The dialog has tabs for 'Target Info' and 'Operating'. The 'Target Info' tab is active, showing the 'Target Name' as 'remotebd' and a checkbox for 'Set as default target'. The 'Operating' tab is also visible, showing the 'Specify the connection type and properties' section with a dropdown for 'Type' set to 'Hardware Server', a 'Host' field with '192.168.226.142', a 'Port' field with '65105', and a checked checkbox for 'Use Symbol Server'. A note states: 'Note: Use Symbol server for source level debugging on remote machine.' There is an 'Advanced >>' button and a 'Test Connection' button.

At the bottom, the 'Target Connections' panel shows a list of connections: 'Hardware Server', 'Local [default]', 'remotebd', 'Linux TCF Agent', and 'QEMU TcfGdbClient'. The 'remotebd' connection is highlighted. The 'Problems' panel shows 0 items. The 'Console' panel shows the output of the 'targets -set -nocase -fil con' command, indicating the end of the session. The 'SDK Log' panel shows the launch and channel information.

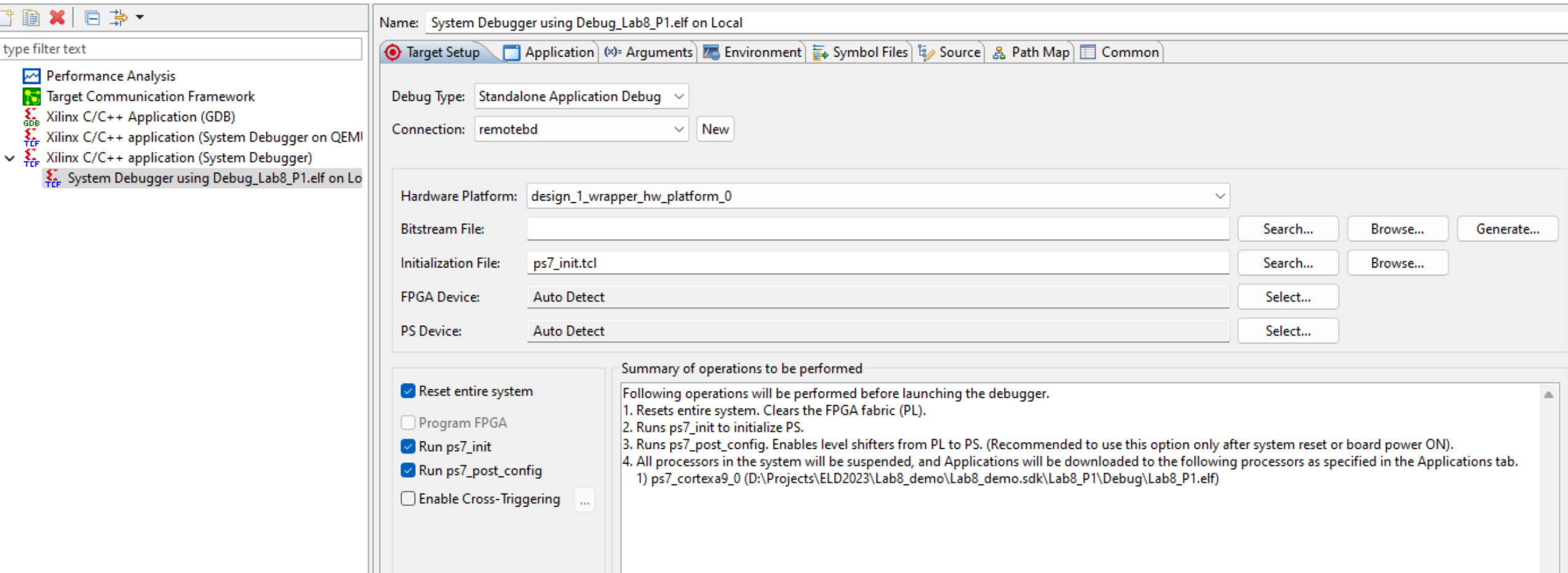
Running Application on Board

The screenshot displays an IDE interface with three main components:

- Project Tree (Left):** Shows a project structure with folders like `design_1_wrapper_hw_platform_0`, `ps7_init_gpl.c`, `ps7`, `sys`, `Lab8_P1`, `Bin`, `Inc`, `Del`, `src`, `Lab8_P1`, `BSF`, `ps7`, `Ma`, `sys`, `Target Con`, `Hardw`, `Loc`, `ren`, `Linux T`, and `QEMU`.
- Context Menu (Center):** A right-click menu is open over the `Lab8_P1` folder. It includes options like `New`, `Go Into`, `Open in New Window`, `Copy` (Ctrl+C), `Paste` (Ctrl+V), `Delete`, `Source`, `Move...`, `Rename...` (F2), `Import...`, `Export...`, `Build Project`, `Clean Project`, `Refresh` (F5), `Close Project`, `Close Unrelated Projects`, `Build Configurations`, `Run As`, `Debug As`, `Compare With`, `Restore from Local History...`, `C/C++ Build Settings`, `Generate Linker Script`, and `Change Referenced BSP`.
- Lab8_P1_bsp Board Support Package (Right):** A panel showing configuration details for the `Lab8_P1_bsp`. It includes buttons for `Modify this BSP's Settings` and `Re-generate BSP Sources`. The `Target Information` section states: "This Board Support Package is compiled to run on the following Hardware Specification: D:\Projects\ELD2023\Lab8_demo\Lab8_P1_bsp Target Processor: ps7_cortexa9_0". The `Operating System` section lists: "Board Support Package OS. Name: standalone Version: 7.0 Description: Standalone is a simple, low-level software library that provides the basic features of a hosted environment. Documentation: [standalone v7.0](#)". The `Peripheral Drivers` section lists: "Drivers present in the Board Support Package. ps7_afi_0 generic ps7_afi_1 generic ps7_afi_2 generic ps7_afi_3 generic ps7_coresight_comp_0 coresightps_dcc Documentation ps7_ddr_0 ddrps Documentation". A sub-menu is open for `Debug As`, showing options: `1 Launch on Hardware (System Debugger)`, `2 Start Performance Analysis`, `3 Launch on Hardware (System Debugger on QEMU)`, `4 Launch on Hardware (GDB)`, `5 Local C/C++ Application`, and `Debug Configurations...` (highlighted with a red box).

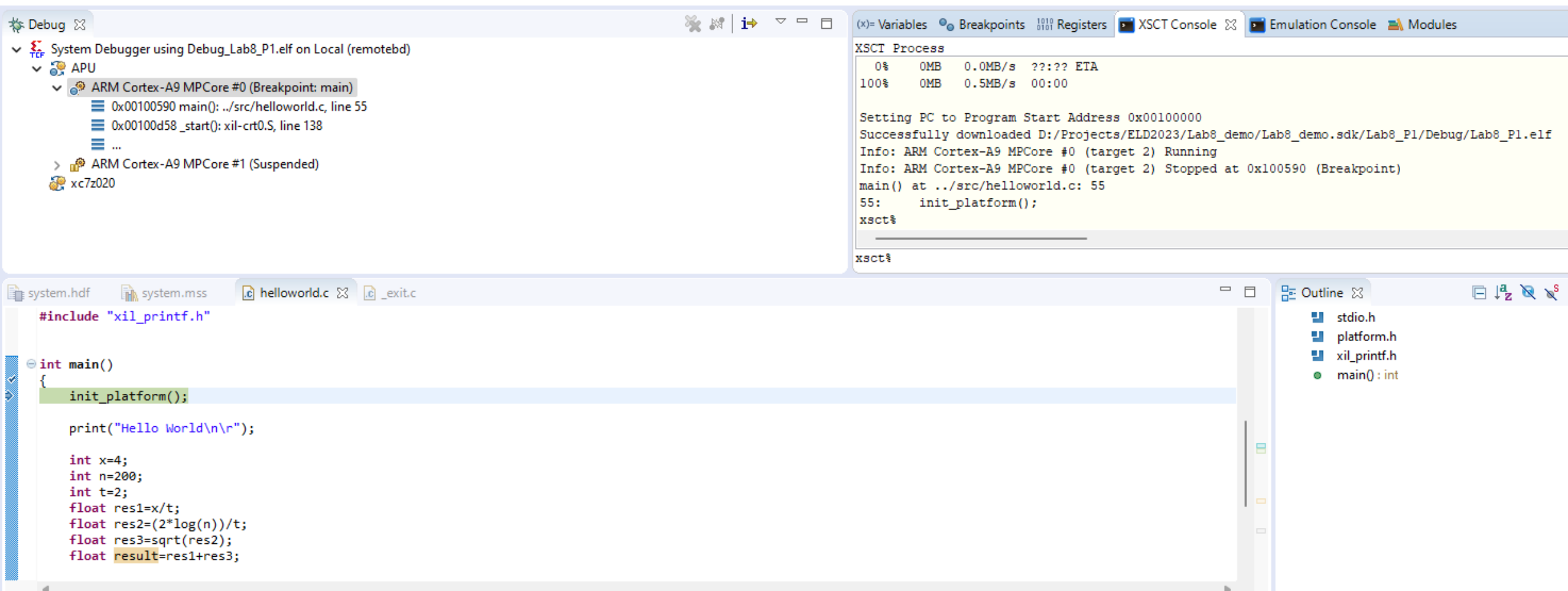
Running Application on Board

- Click on Debug



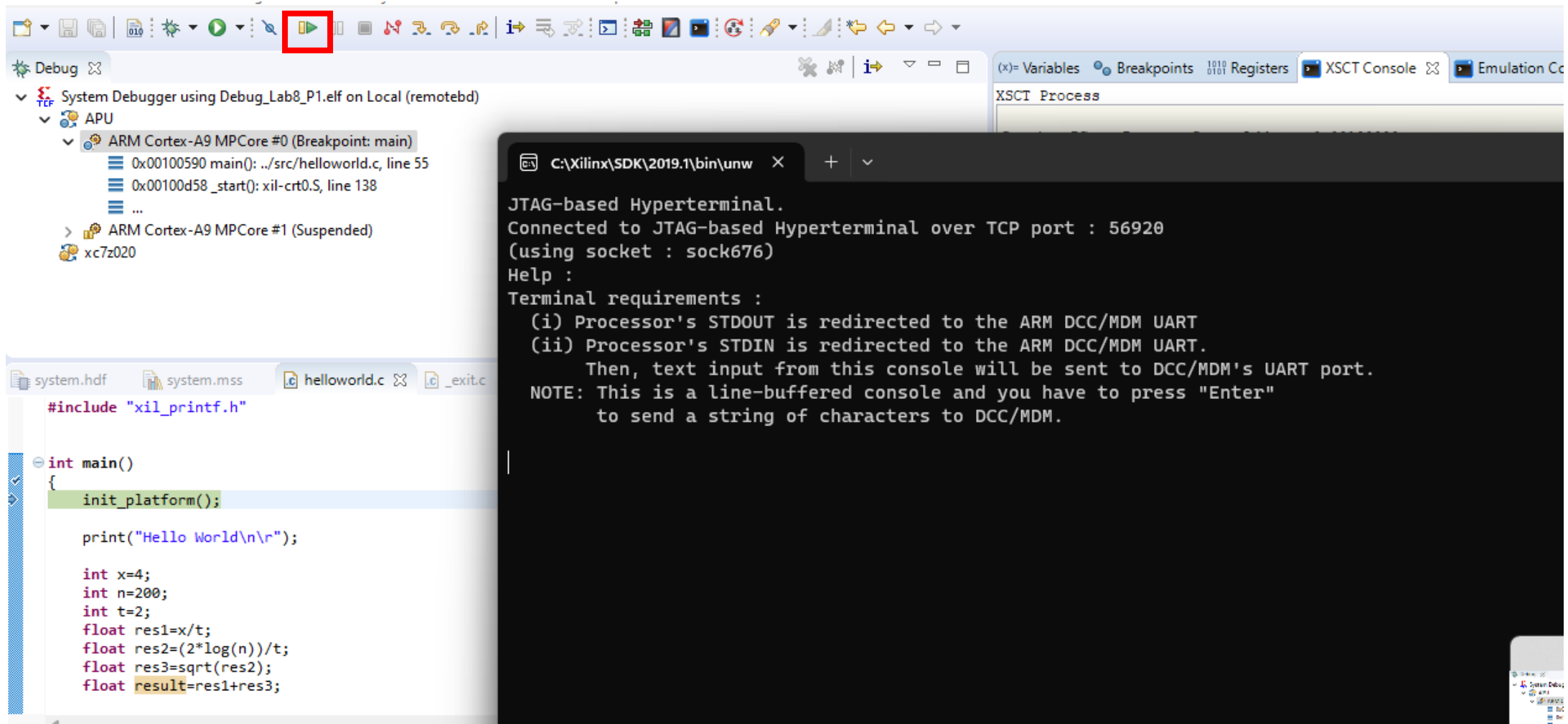
Running Application on Board

- In XSCT Console, type jtagterminal to open the terminal



Running Application on Board

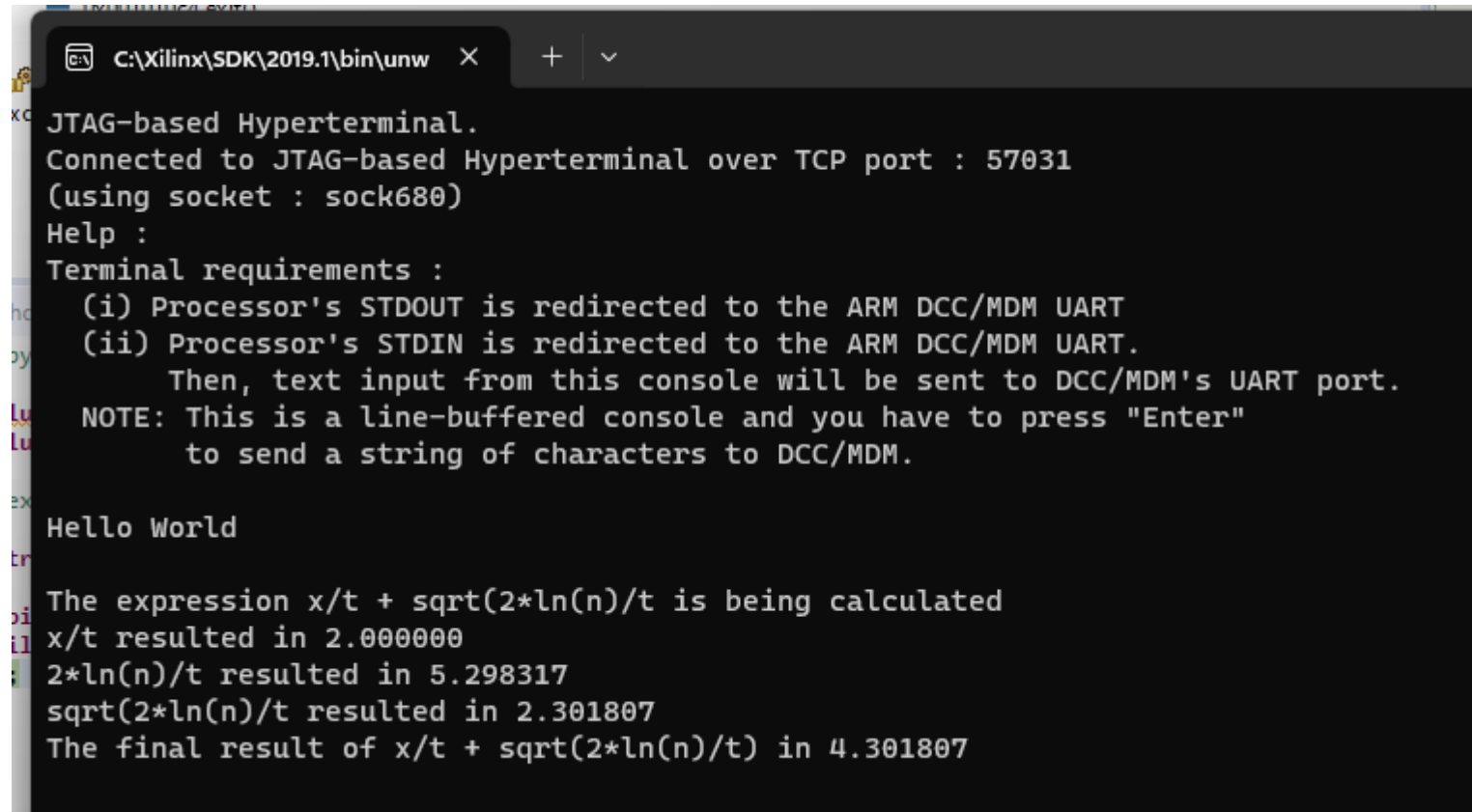
- Run the application



The screenshot displays the Xilinx IDE interface. The top toolbar features a red box around the 'Run' button (a green play icon). The left sidebar shows the 'System Debugger' window, which is connected to the 'APU' (ARM Cortex-A9 MPCore #0) and has a breakpoint set at 'main()'. The bottom-left pane shows the source code for 'helloworld.c', with the 'main()' function highlighted. The bottom-right pane shows the 'JTAG-based Hyperterminal' window, which displays the following text:

```
JTAG-based Hyperterminal.  
Connected to JTAG-based Hyperterminal over TCP port : 56920  
(using socket : sock676)  
Help :  
Terminal requirements :  
  (i) Processor's STDOUT is redirected to the ARM DCC/MDM UART  
  (ii) Processor's STDIN is redirected to the ARM DCC/MDM UART.  
Then, text input from this console will be sent to DCC/MDM's UART port.  
NOTE: This is a line-buffered console and you have to press "Enter"  
to send a string of characters to DCC/MDM.
```


Output



```
C:\Xilinx\SDK\2019.1\bin\unw X + v
JTAG-based Hyperterminal.
Connected to JTAG-based Hyperterminal over TCP port : 57031
(using socket : sock680)
Help :
Terminal requirements :
  (i) Processor's STDOUT is redirected to the ARM DCC/MDM UART
  (ii) Processor's STDIN is redirected to the ARM DCC/MDM UART.
      Then, text input from this console will be sent to DCC/MDM's UART port.
NOTE: This is a line-buffered console and you have to press "Enter"
      to send a string of characters to DCC/MDM.

Hello World

The expression  $x/t + \sqrt{2 \cdot \ln(n)/t}$  is being calculated
x/t resulted in 2.000000
2*ln(n)/t resulted in 5.298317
sqrt(2*ln(n)/t) resulted in 2.301807
The final result of  $x/t + \sqrt{2 \cdot \ln(n)/t}$  in 4.301807
```