

Block Design Guide

SoC Design

Block Design Guide

- In Lab1 and Lab2, you may know how to create a block design and connect each IP.
- In this document, I'll introduce how to build block design by TCL script

Create Project

```
set _xil_proj_name "vvd_caravel_fpga"  
create_project ${_xil_proj_name} ./${_xil_proj_name} -part xc7z020clg400-1  
set proj_dir [get_property directory [current_project]]  
set origin_dir "."  
set obj [get_filesets sources_1]
```

Add files

```
set files [list \  
[file normalize "${origin_dir}/vitis_prj/verilog_spiflash/spiflash.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/vip/RAM256.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/vip/RAM128.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/rtl/soc/VexRiscv_MinDebugCache.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/rtl/soc/chip_io.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/rtl/soc/gpio_control_block.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/rtl/soc/gpio_defaults_block.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/rtl/soc/housekeeping.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/rtl/soc/housekeeping_spi.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/rtl/soc/mgmt_core.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/rtl/soc/mgmt_core_wrapper.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/rtl/soc/mprj_io.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/rtl/soc/caravel.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/rtl/user/${user_design_file}"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/rtl/user/bram.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/rtl/user/user_project_wrapper.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/rtl/header/user_defines.v"] \  
[file normalize "${origin_dir}/vvd_srcs/caravel_soc/rtl/header/defines.v"] \  
]  
add_files -norecurse -fileset $obj $files
```

Add Define and Pre-built IP

```
set file "$origin_dir/vvd_srcs/caravel_soc/rtl/header/user_defines.v"
set file [file normalize $file]
set file_obj [get_files -of_objects [get_filesets sources_1] [list "$file"]]
set_property -name "file_type" -value "Verilog Header" -objects $file_obj
set_property -name "is_global_include" -value "1" -objects $file_obj

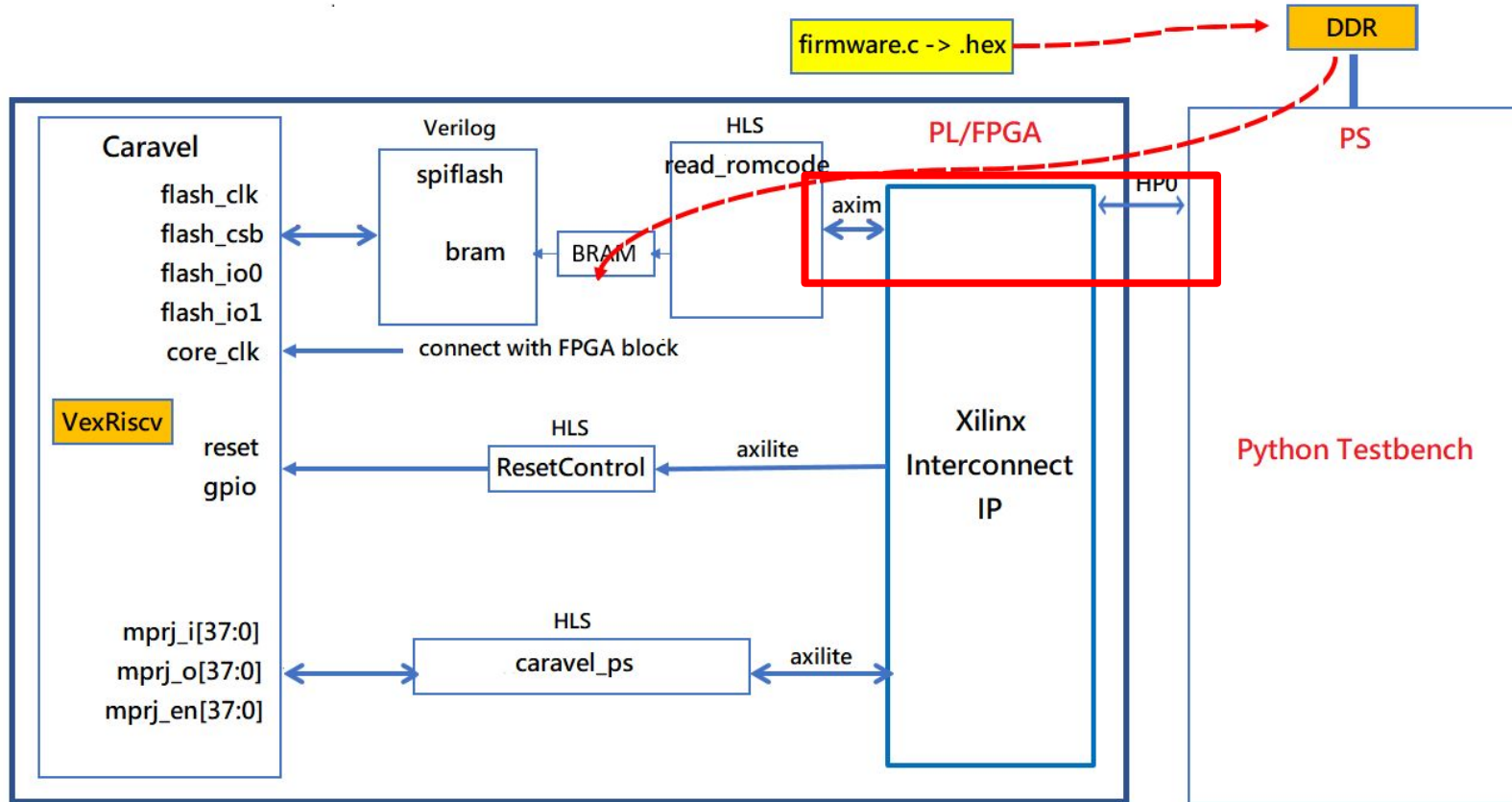
set file "$origin_dir/vvd_srcs/caravel_soc/rtl/header/defines.v"
set file [file normalize $file]
set file_obj [get_files -of_objects [get_filesets sources_1] [list "$file"]]
set_property -name "file_type" -value "Verilog Header" -objects $file_obj
set_property -name "is_global_include" -value "1" -objects $file_obj

set obj [get_filesets sources_1]
set_property "ip_repo_paths" "[file normalize "$origin_dir/vitis_prj/hls_caravel_ps"]
                                [file normalize "$origin_dir/vitis_prj/hls_output_pin"]
                                [file normalize "$origin_dir/vitis_prj/hls_read_romcode"]" $obj
update_ip_catalog -rebuild
```

Create Block Design

```
set design_name design_1  
create_bd_design $design_name
```

Create AXI Interconnect for readmem



Create AXI Interconnect for readmem- (1)

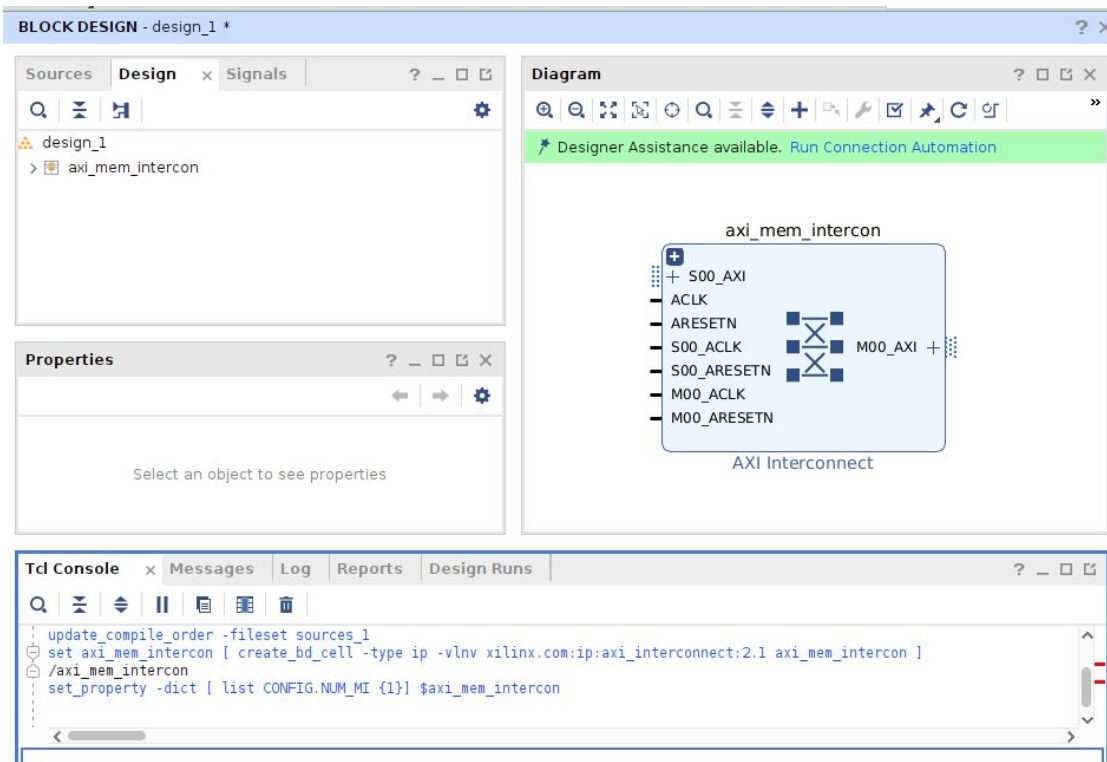
```
set axi_mem_intercon [ create_bd_cell -type ip -vlnv xilinx.com:ip:axi_interconnect:2.1  
axi_mem_intercon ]
```

The screenshot displays the Xilinx Block Design IDE interface for a project named 'design_1'. The 'Sources' pane on the left shows the project hierarchy with 'design_1' and its sub-component 'axi_mem_intercon'. The 'Diagram' pane on the right shows a block diagram of the 'axi_mem_intercon' component, which is an 'AXI Interconnect'. The diagram shows the interconnect's ports: S00_AXI, ACLK, ARESETN, S00_ACLK, S00_ARESETN, M00_AXI, M00_ACLK, M00_ARESETN, M01_ACLK, and M01_ARESETN. The interconnect is represented by a blue box with a cross symbol inside. The 'Properties' pane at the bottom left is empty, showing a message 'Select an object to see properties'. The 'Tcl Console' at the bottom shows the command used to create the interconnect: `set axi_mem_intercon [create_bd_cell -type ip -vlnv xilinx.com:ip:axi_interconnect:2.1 axi_mem_intercon]`. The console also shows the command `/axi_mem_intercon` and a prompt 'Type a Tcl command here'.

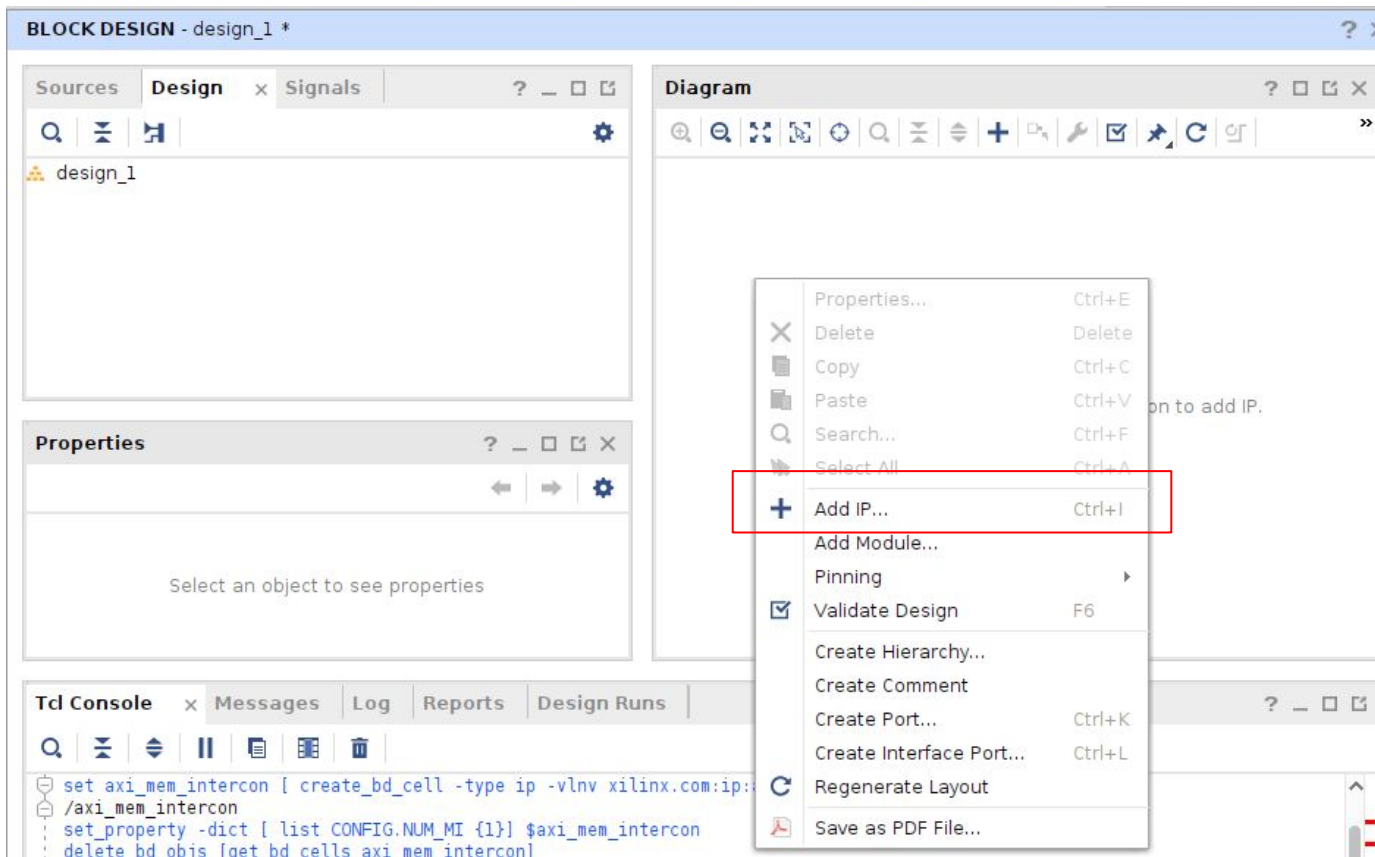
Create AXI Interconnect for readmem- (2)

```
set_property -dict [ list CONFIG.NUM_MI {1}] $axi_mem_intercon
```

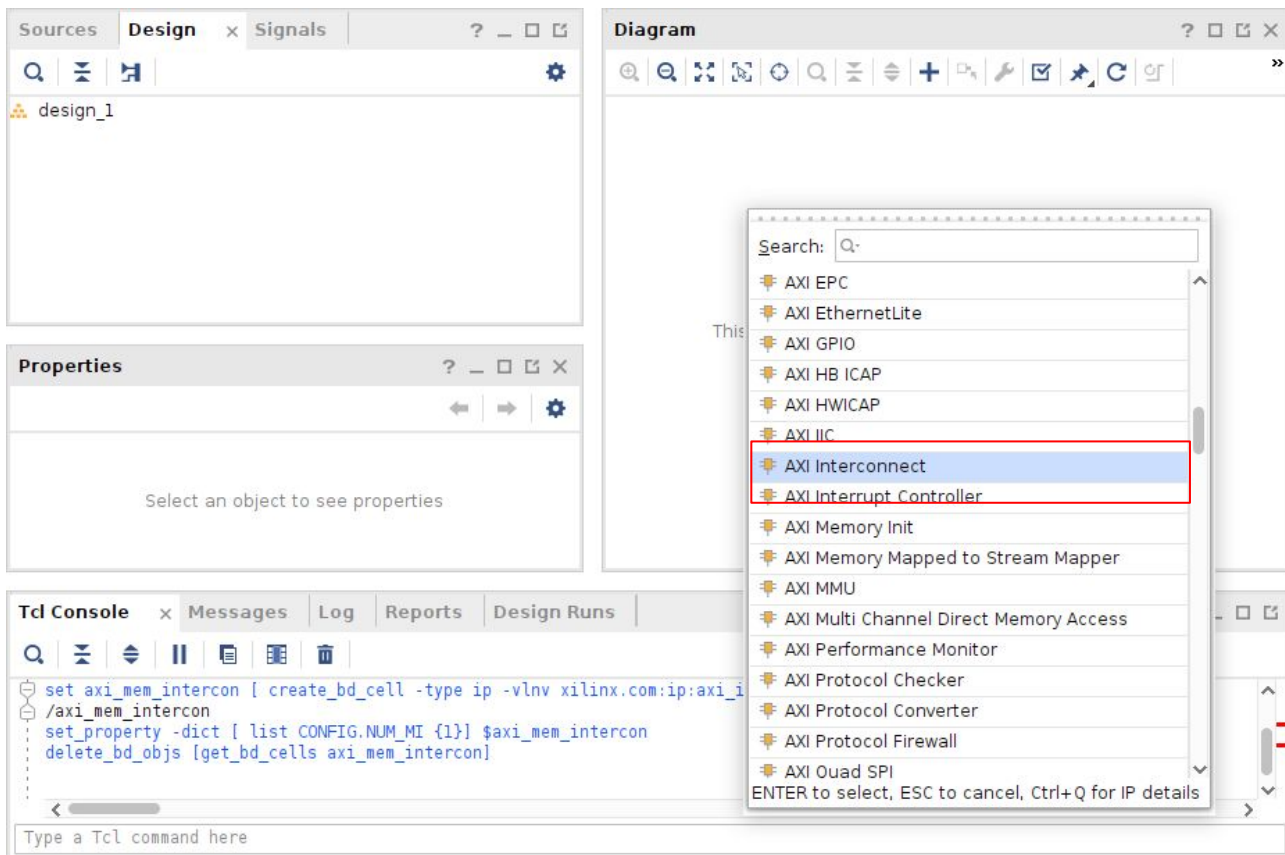
- Set one master channel



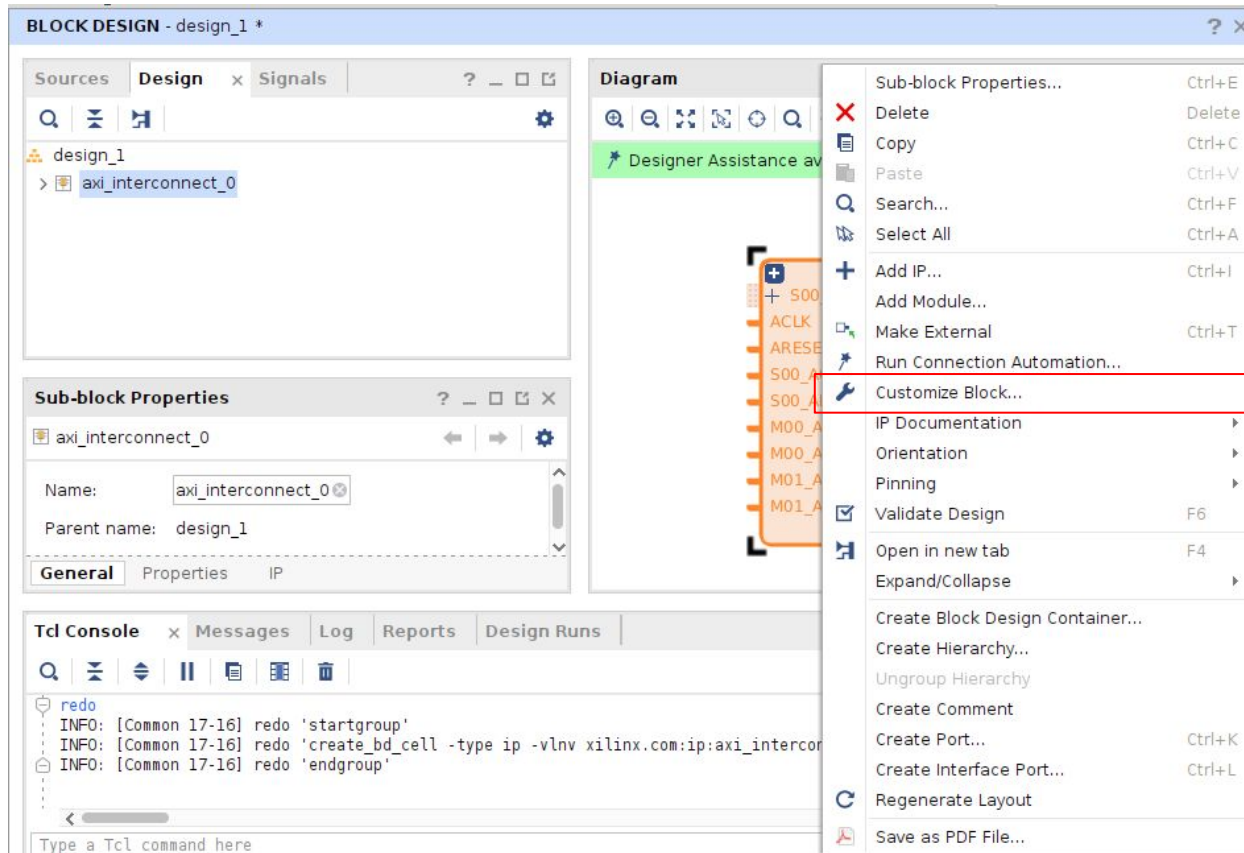
Create AXI Interconnect without TCL - (1)



Create AXI Interconnect without TCL - (2)



Create AXI Interconnect without TCL - (3)



Create AXI Interconnect without TCL - (4)

Re-customize IP

AXI Interconnect (2.1)

Documentation IP Location

Component Name

Top Level Settings Slave Interfaces

Number of Slave Interfaces

Number of Master Interfaces

Interconnect Optimization Strategy

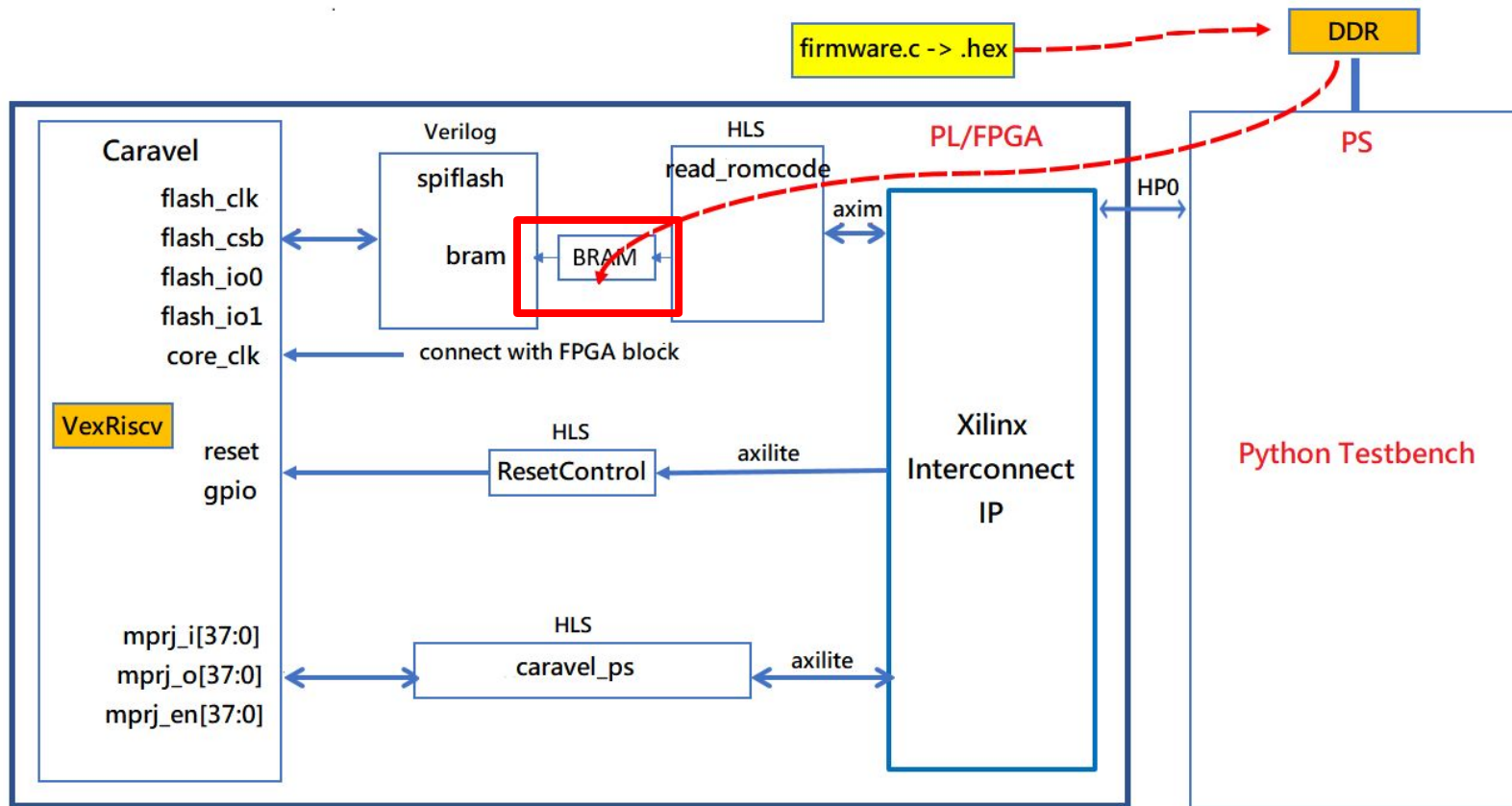
AXI Interconnect includes IP Integrator automatic configuration. When the endpoint IPs attached to the interconnect differ in width, clock or protocol, a converter IP is inserted inside the interconnect. If a converter IP is inserted, IP Integrator automatically configures the converter to match the endpoints. To see which conversion IPs have been inserted, click the 'expand hierarchy' buttons to explore in the IP Integrator address editor.

NOTE: Addressing information for AXI Interconnect is available in the IP Integrator address editor.

☐ Enable Advanced Configuration Options

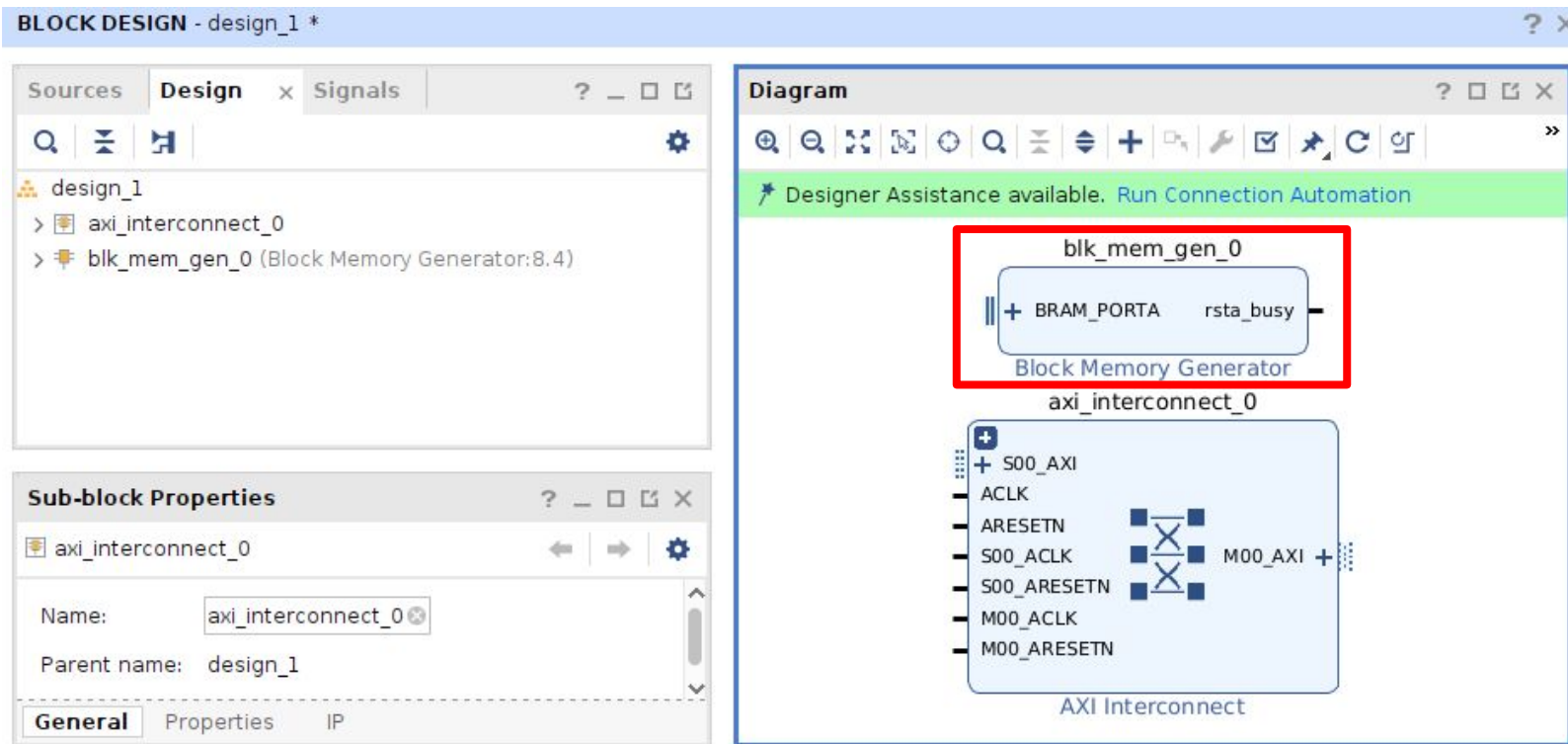
OK Cancel

Create BRAM



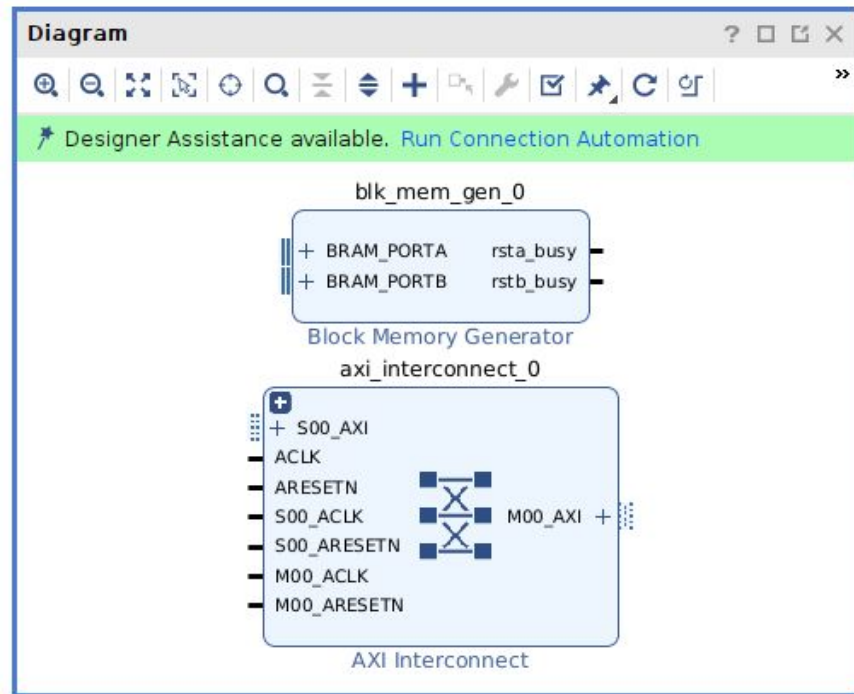
Create BRAM – (1)

```
set blk_mem_gen_0 [ create_bd_cell -type ip -vlnv xilinx.com:ip:blk_mem_gen:8.4 blk_mem_gen_0 ]
```



Create BRAM – (2)

```
set_property -dict [ list \  
    CONFIG.Enable_B {Use_ENB_Pin} \  
    CONFIG.Memory_Type {True_Dual_Port_RAM} \  
    CONFIG.Port_B_Clock {100} \  
    CONFIG.Port_B_Enable_Rate {100} \  
    CONFIG.Port_B_Write_Rate {50} \  
    CONFIG.Use_RSTB_Pin {true} \  
] $blk_mem_gen_0
```



Create BRAM without TCL – (1)

The screenshot displays the Vivado IDE interface with the following components:

- Sources**: Shows a design named `design_1` containing a block named `axi_interconnect_0`.
- Sub-block Properties**: Shows the properties of the `axi_interconnect_0` block. The Name is `axi_interconnect_0` and the Parent name is `design_1`.
- Diagram**: Shows a search for `Q- block` with two matches: `2D Graphics Accelerator Bit Block Transfer` and `Block Memory Generator`. The `Block Memory Generator` is highlighted by a red box.
- Tcl Console**: Shows the TCL commands used to create the BRAM:

```
CONFIG.Port_B_Write_Rate {50} \  
CONFIG.Use_RSTB_Pin {true} \  
] $blk_mem_gen_0  
delete_bd_objs [get_bd_cells blk_mem_gen_0]
```

ENTER to select, ESC to cancel, Ctrl+Q for IP details

Create BRAM without TCL – (2)

- Customize Block
- Set memory type to dual port RAM
- Other attributes are set by default

```
set_property -dict [ list \  
    CONFIG.Enable_B {Use_ENB_Pin} \  
    CONFIG.Memory_Type {True_Dual_Port_RAM} \  
    CONFIG.Port_B_Clock {100} \  
    CONFIG.Port_B_Enable_Rate {100} \  
    CONFIG.Port_B_Write_Rate {50} \  
    CONFIG.Use_RSTB_Pin {true} \  
] $blk_mem_gen_0
```

Block Memory Generator (8.4)

Documentation IP Location

Component Name blk_mem_gen_0

Basic Port A Options Port B Options Other Options Summary

Mode BRAM Controller ☒ Generate address interface with 32 bits

Memory Type True Dual Port RAM ☐ Common Clock

ECC Options

ECC Type No ECC

☐ Error Injection Pins Single Bit Error Injection

Write Enable

☒ Byte Write Enable

Byte Size (bits) 8

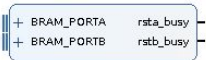
Algorithm Options

Defines the algorithm used to concatenate the block RAM primitives. Refer datasheet for more information.

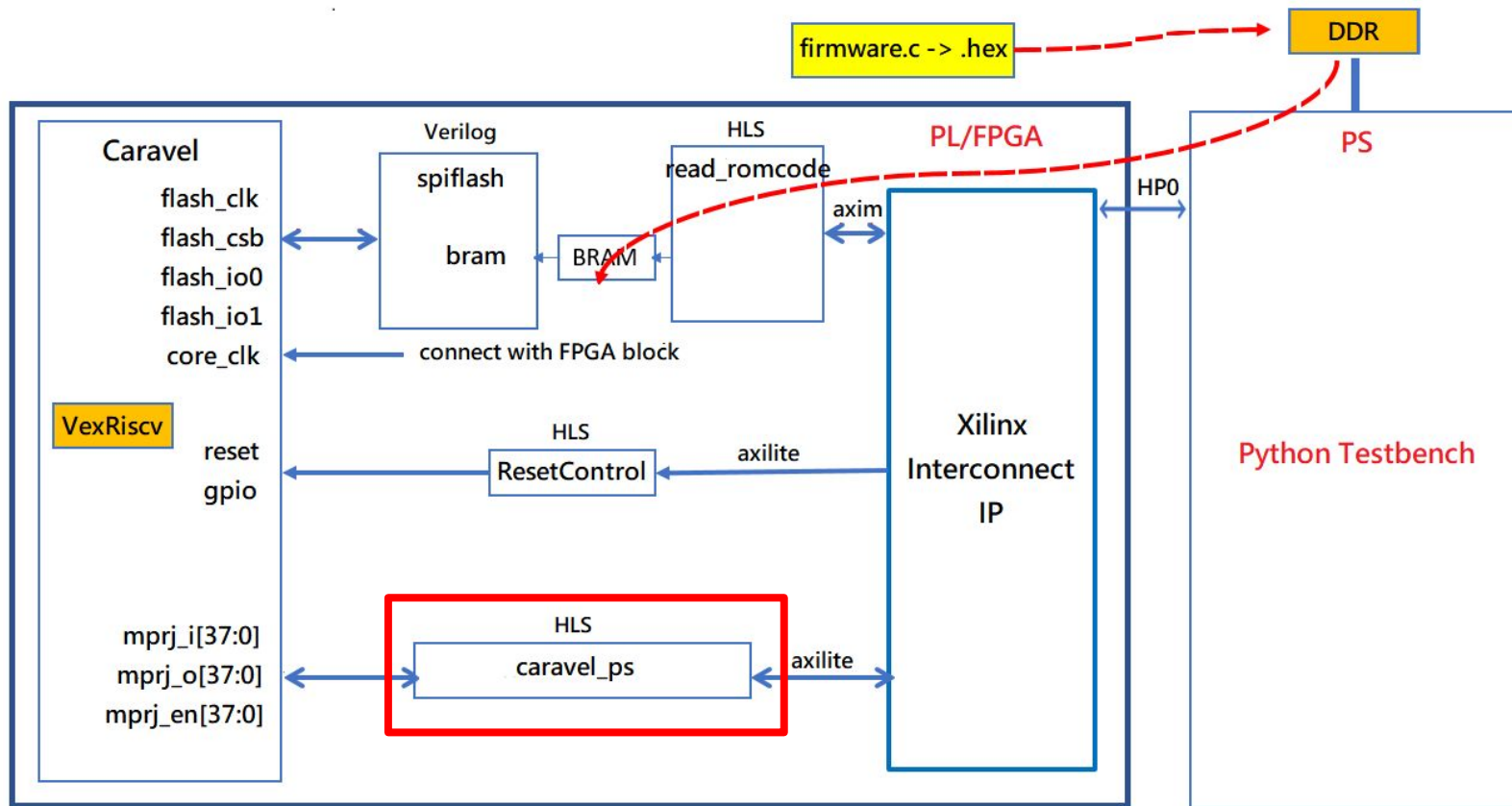
Algorithm Minimum Area

Primitive 8kx2

OK Cancel

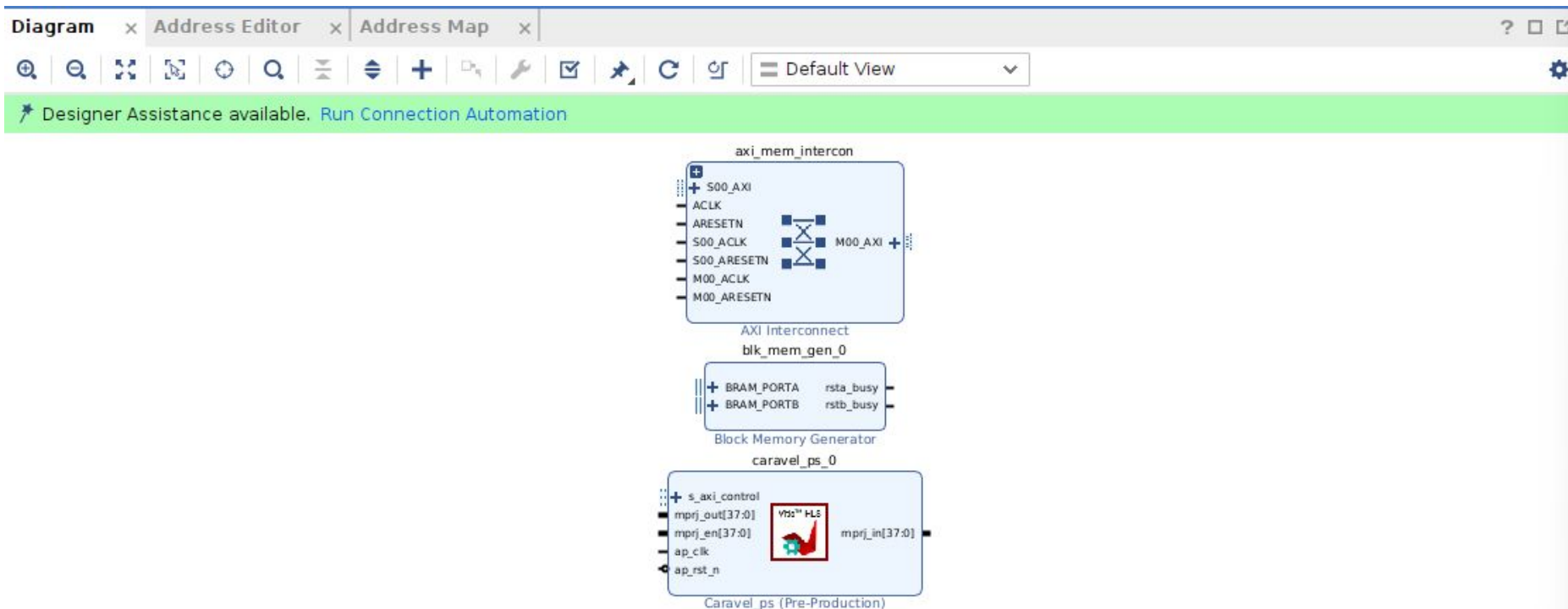


Create Caravel PS



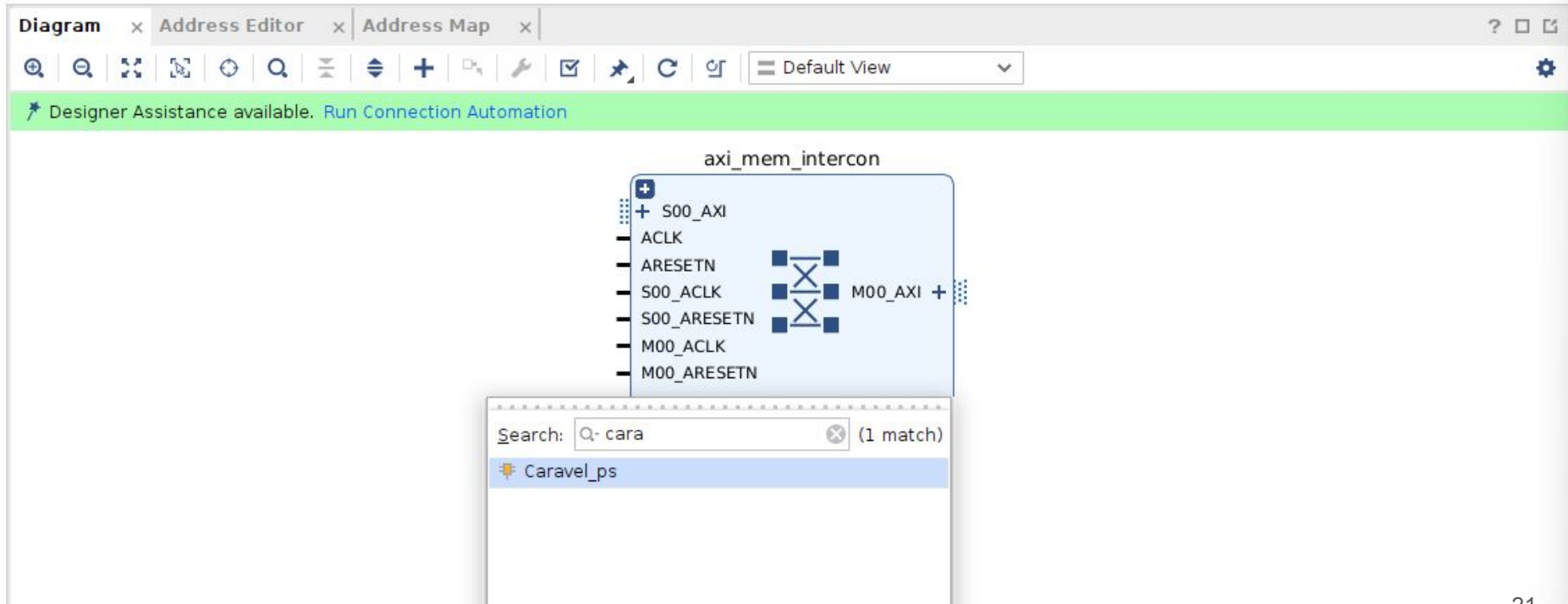
Create Caravel PS

```
set caravel_ps_0 [ create_bd_cell -type ip -vlnv xilinx.com:hls:caravel_ps:0.0 caravel_ps_0 ]
```

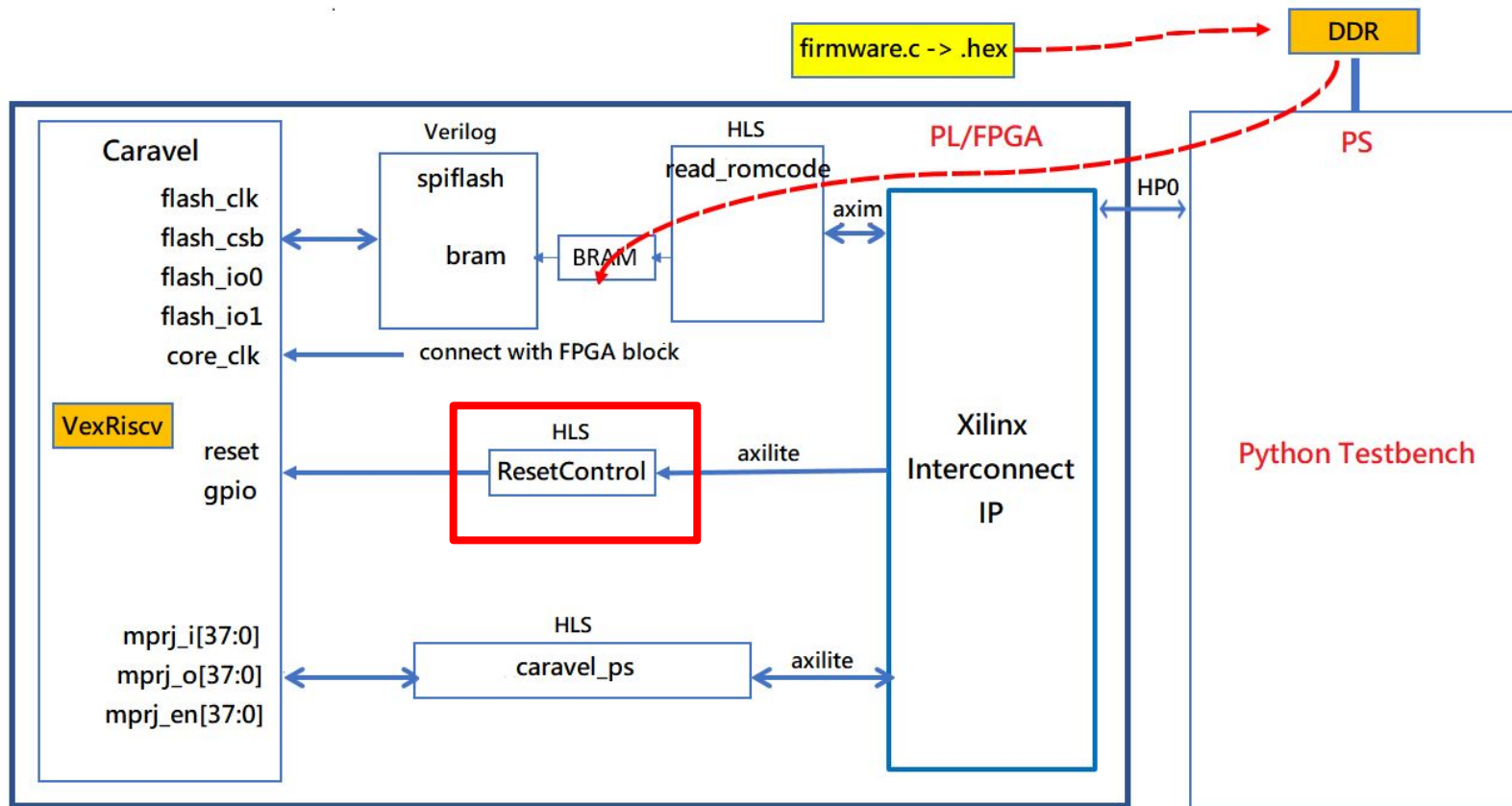


Create Caravel PS without TCL

- Add IP



Create Output Pin



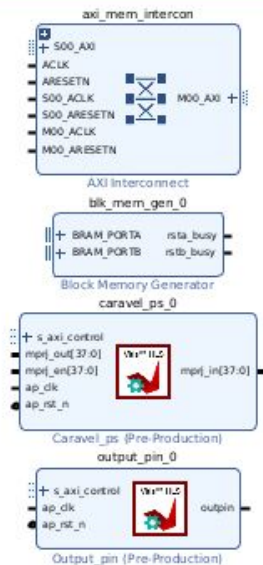
Create Output Pin

```
set output_pin_0 [ create_bd_cell -type ip -vlnv xilinx.com:hls:output_pin:0.0 output_pin_0 ]
```

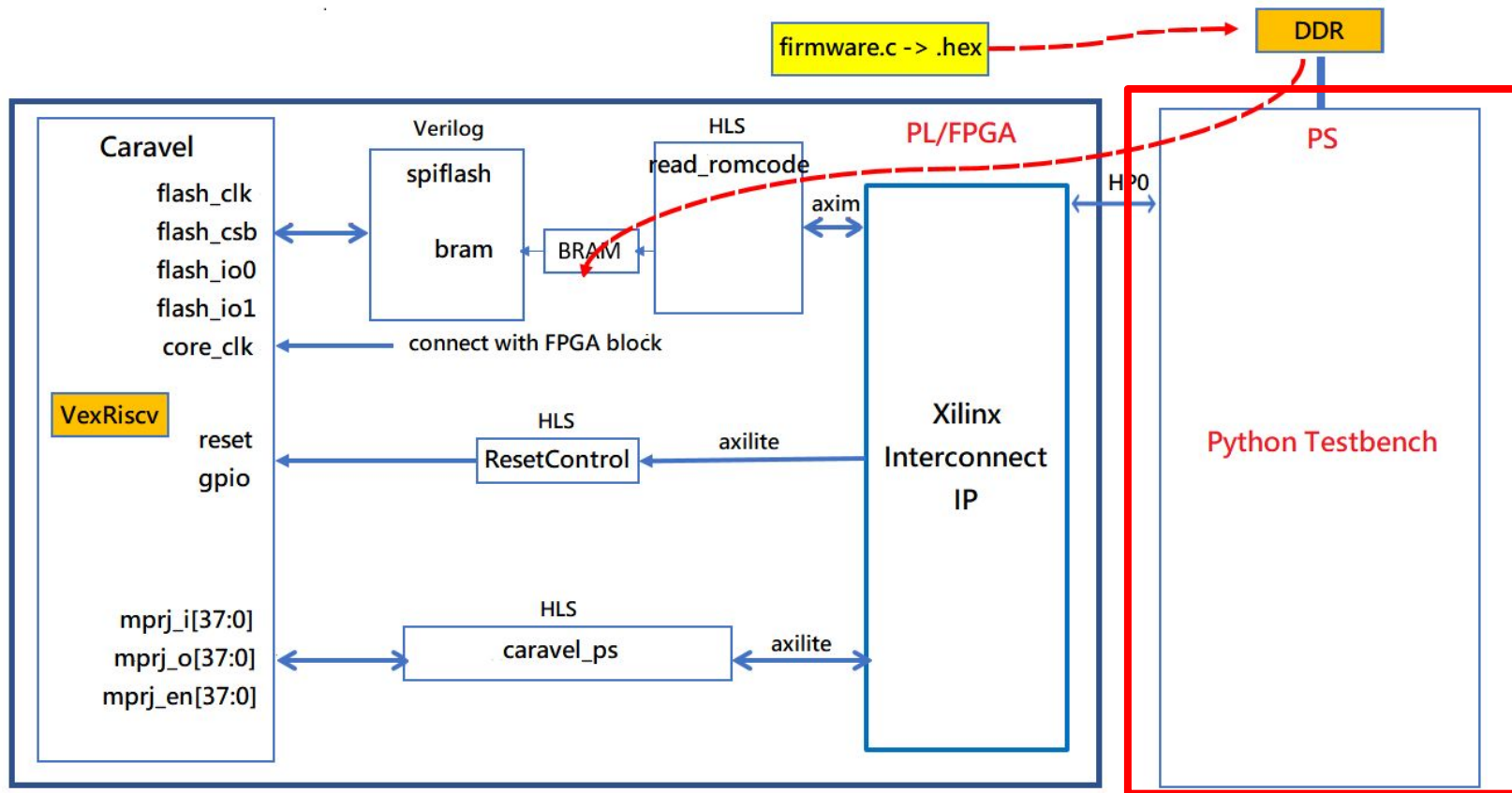
Diagram x Address Editor x Address Map x



★ Designer Assistance available. [Run Connection Automation](#)

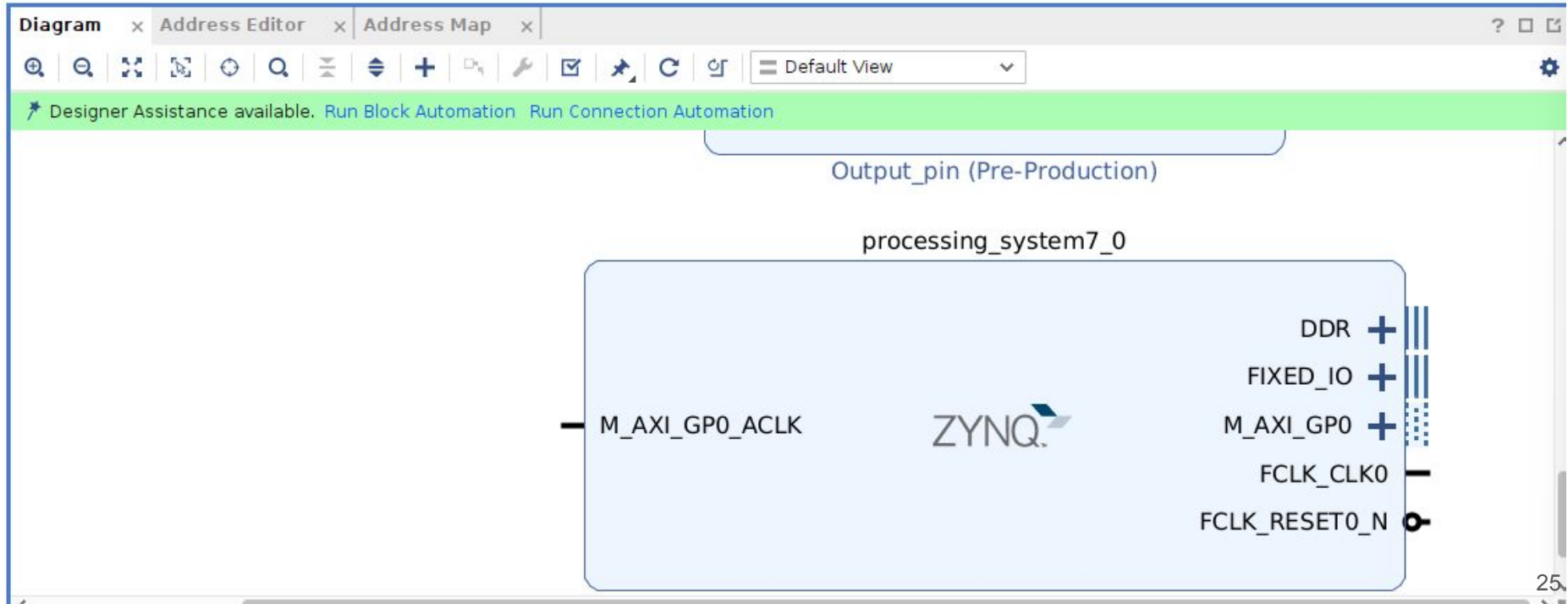


Create PS



Create PS - (1)

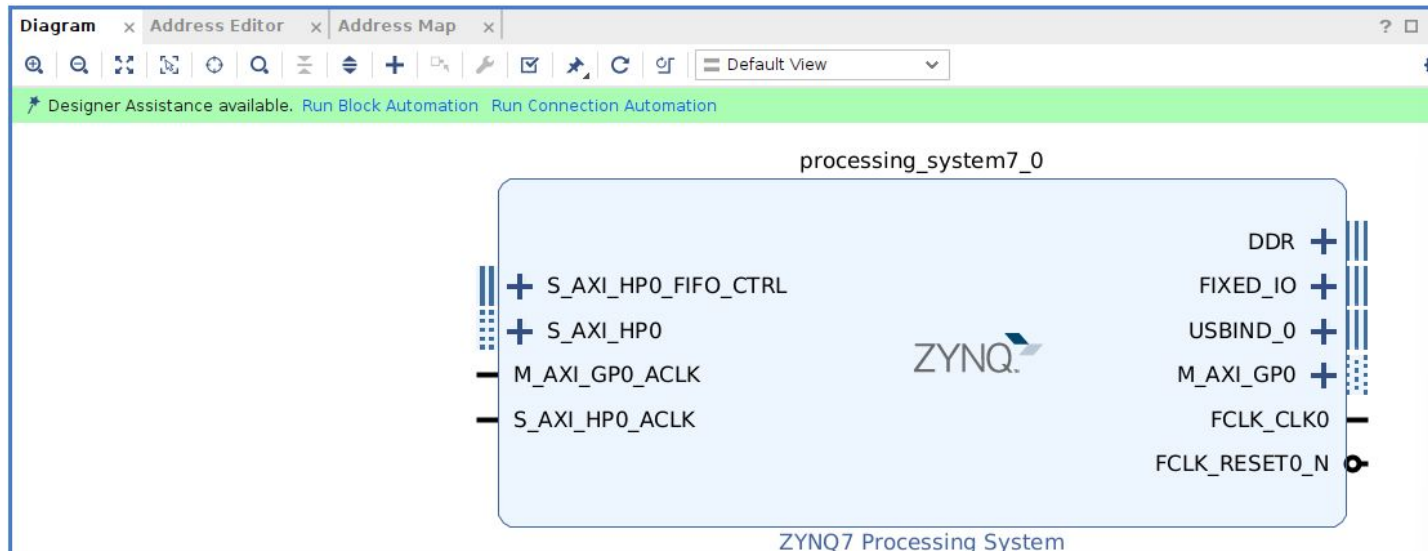
```
set processing_system7_0 [ create_bd_cell -type ip -vlnv xilinx.com:ip:processing_system7:5.5  
processing_system7_0 ]
```



Create PS - (2)

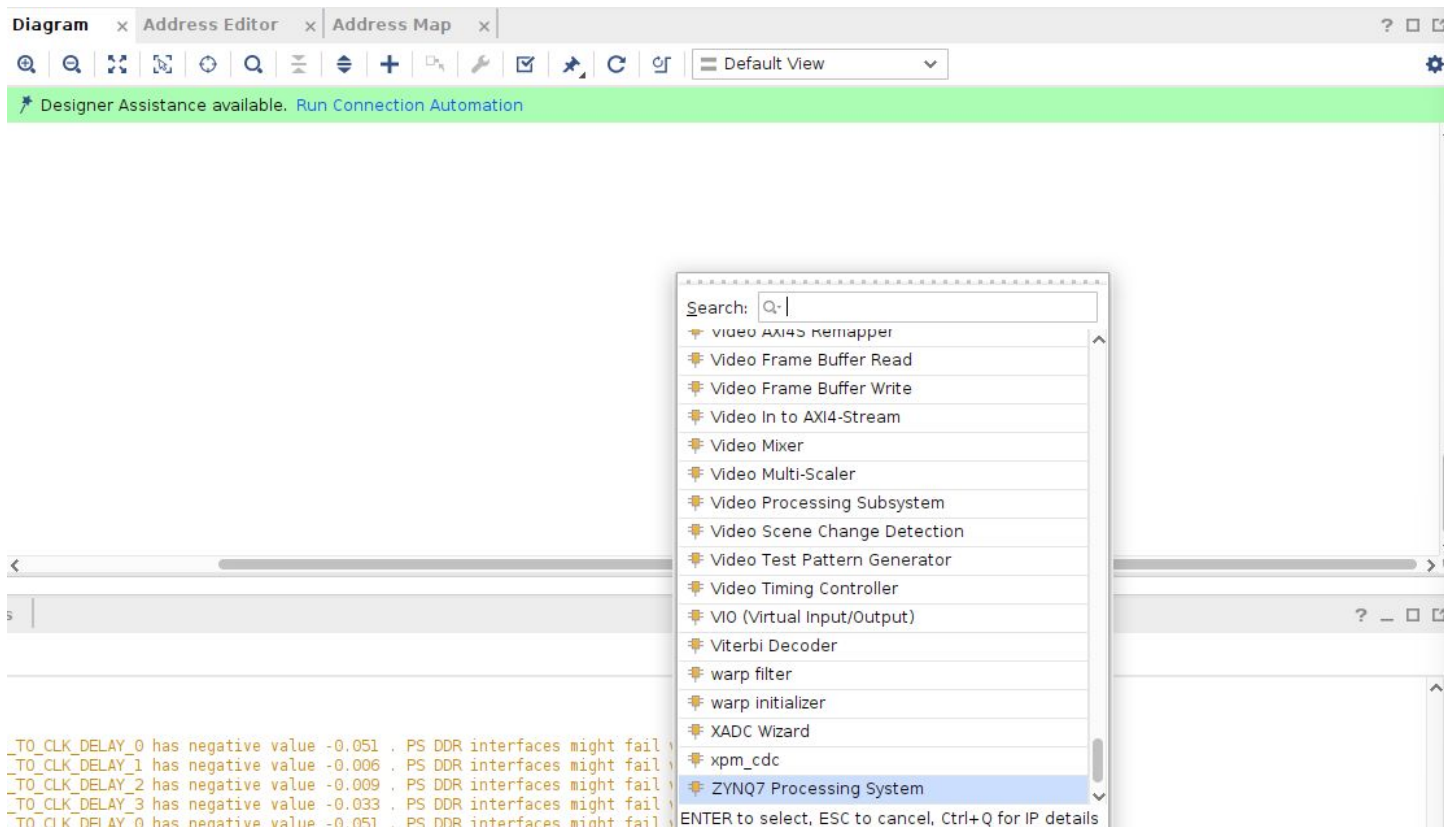
- It has to configure lots of attributes
- Please refer to

https://github.com/bol-edu/caravel-soc_fpga-lab/blob/main/lab-exmem_fir/vivado/vvd_caravel_fpga_10mhz.tcl#L408-L1184



Create PS without TCL - (1)

- Add IP



Create PS without TCL - (2)

- One HP port
- Refer to

https://github.com/bol-edu/caravel-soc_fpga-lab/blob/main/lab-exmem_fir/vivado/vvd_caravel_fpga_10mhz.tcl#L1174

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 [Summary Report](#)

Search: Q-

Name	Select	Description
> General		
> AXI Non Secure Enablement	0	Enable AXI Non Secure Transaction
> GP Slave AXI Interface		
> S AXI HP0 interface	<input checked="" type="checkbox"/>	Enables AXI high performance slave interface 0
> S AXI HP1 interface	<input type="checkbox"/>	Enables AXI high performance slave interface 1
> S AXI HP2 interface	<input type="checkbox"/>	Enables AXI high performance slave interface 2
> S AXI HP3 interface	<input type="checkbox"/>	Enables AXI high performance slave interface 3
> ACP Slave AXI Interface		
> DMA Controller		
> PS-PL Cross Trigger interface	<input type="checkbox"/>	Enables PL cross trigger signals to PS and vice-versa

Create PS without TCL - (3)

- Set Peripheral IO
- SPI flash
- Ethernet
- USB
- SD
- UART
- GPIO

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

Peripheral I/O Pins

Summary Report

Search:

Bank 0 LVCMOS 3.3V Bank 1 LVCMOS 1.8V

Peripherals

- ☒ Quad SPI Flash
- ☐ SRAM/NOR Flash
- ☐ NAND Flash
- ☒ Ethernet 0
- ☐ Ethernet 1
- ☒ USB 0
- ☐ USB 1
- ☒ SD 0
- ☐ SD 1
- ☐ SPI 0
- ☐ SPI 1
- ☒ UART 0
- ☐ UART 1

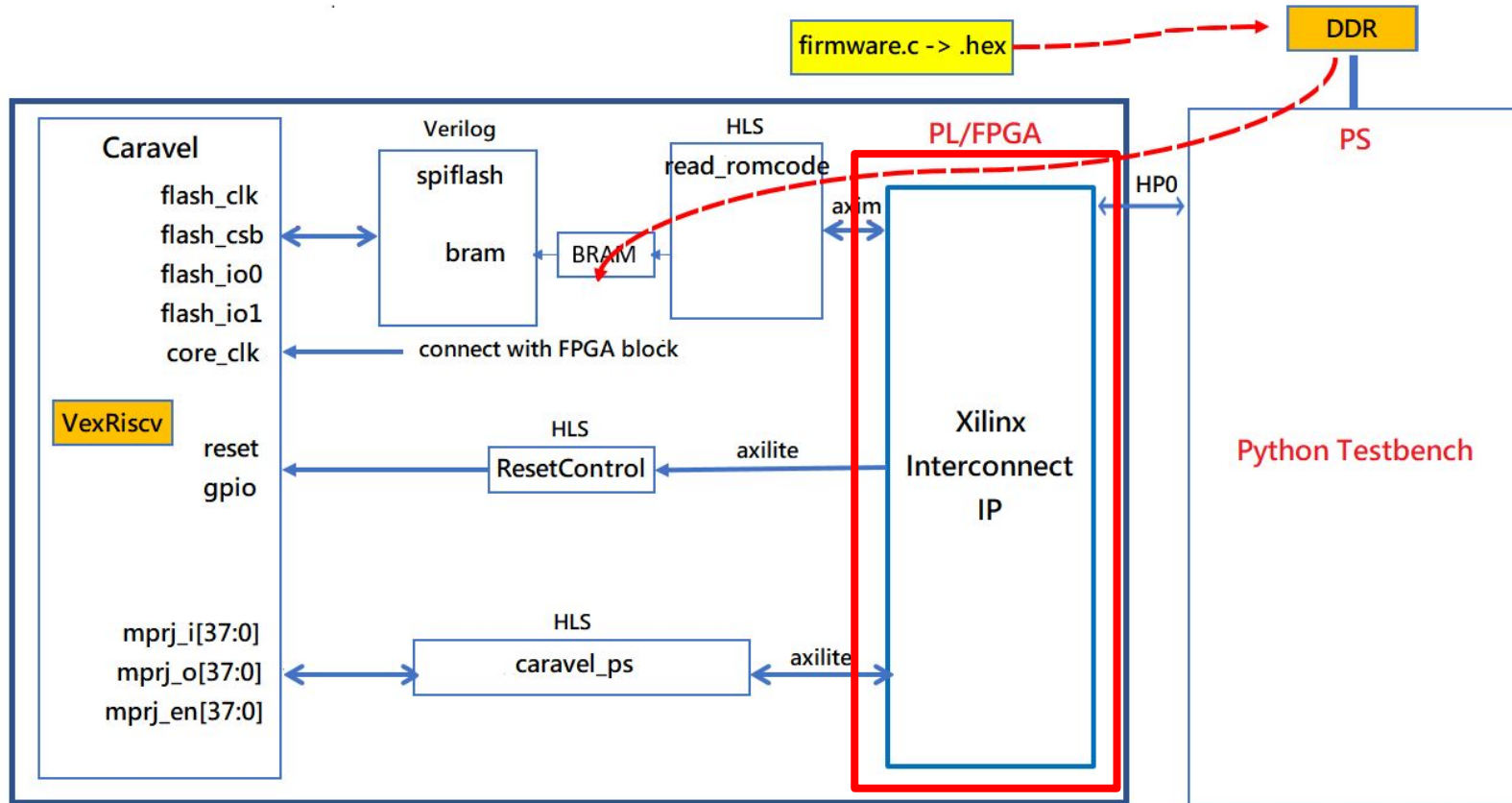
Diagram showing peripheral I/O pins mapped to Bank 0 and Bank 1. The diagram includes a grid of pins (0-30) and various peripheral blocks (Quad SPI Flash, SRAM/NOR Flash, NAND Flash, Ethernet 0, USB 0, SD 0, SD 1, SPI 0, SPI 1, UART 0, UART 1) connected to specific pins.

OK Cancel

Create PS without TCL - (4)

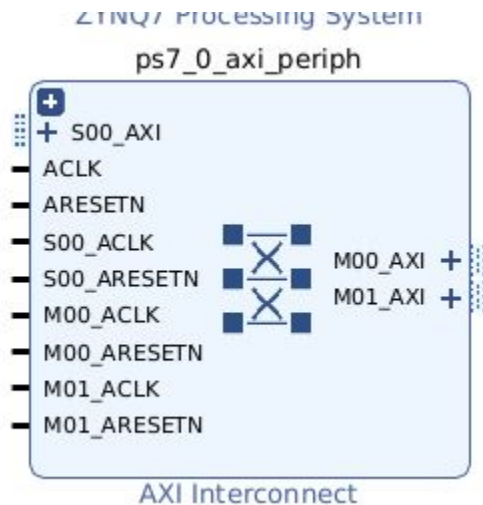
- DDR Configuration
- Memory part
 - https://github.com/bol-edu/caravel-soc_fpga-lab/blob/main/lab-exmem_fir/vivado/vvd_caravel_fpga_10mhz.tcl#L1123
- ~~DQS to Clock Delay~~
 - https://github.com/bol-edu/caravel-soc_fpga-lab/blob/main/lab-exmem_fir/vivado/vvd_caravel_fpga_10mhz.tcl#L1101-L1104
- ~~Board Delay~~
 - https://github.com/bol-edu/caravel-soc_fpga-lab/blob/main/lab-exmem_fir/vivado/vvd_caravel_fpga_10mhz.tcl#L1067-L1070

Create AXI for peripheral



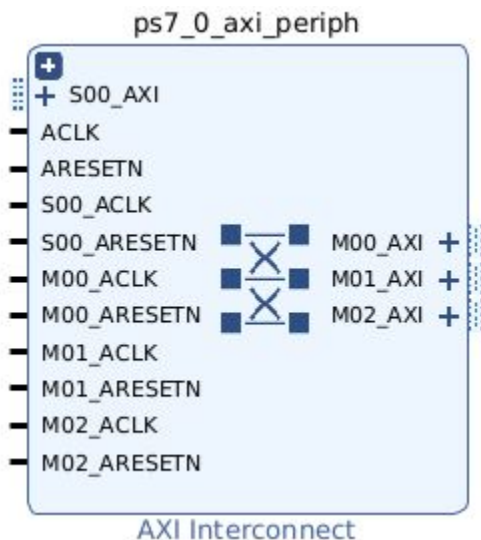
Create AXI for peripheral - (1)

```
set ps7_0_axi_periph [ create_bd_cell -type ip -vlnv xilinx.com:ip:axi_interconnect:2.1  
ps7_0_axi_periph ]
```



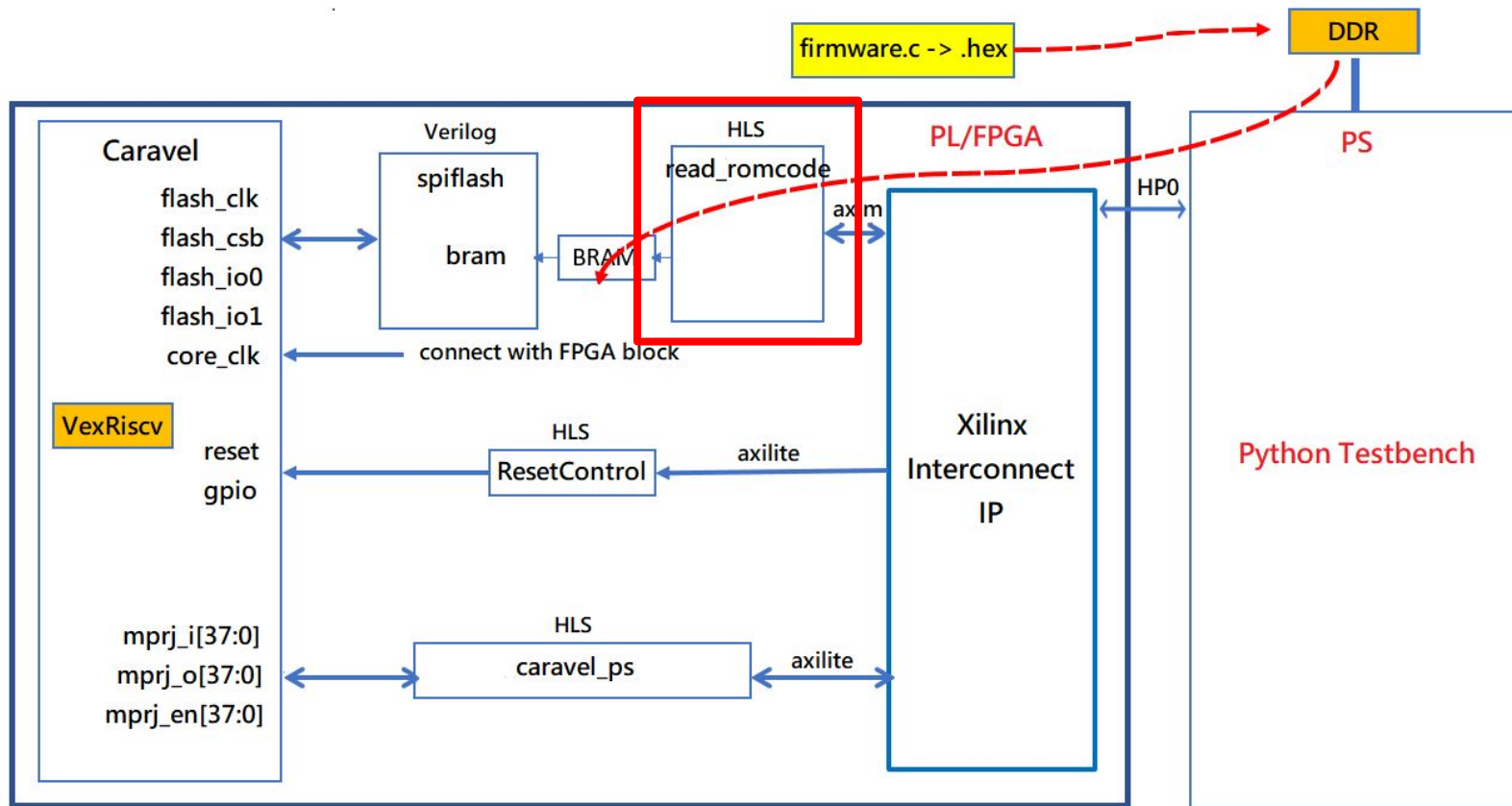
Create AXI for peripheral - (2)

```
set_property -dict [ list CONFIG.NUM_MI {3}] $ps7_0_axi_periph
```



- Without TCL, you can do the same way as page 10~13

Create Read ROM



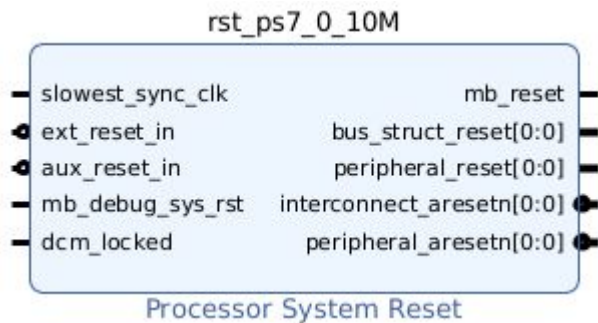
Create Read ROM

```
set read_romcode_0 [ create_bd_cell -type ip -vlnv  
xilinx.com:hls:read_romcode:0.0 read_romcode_0 ]
```

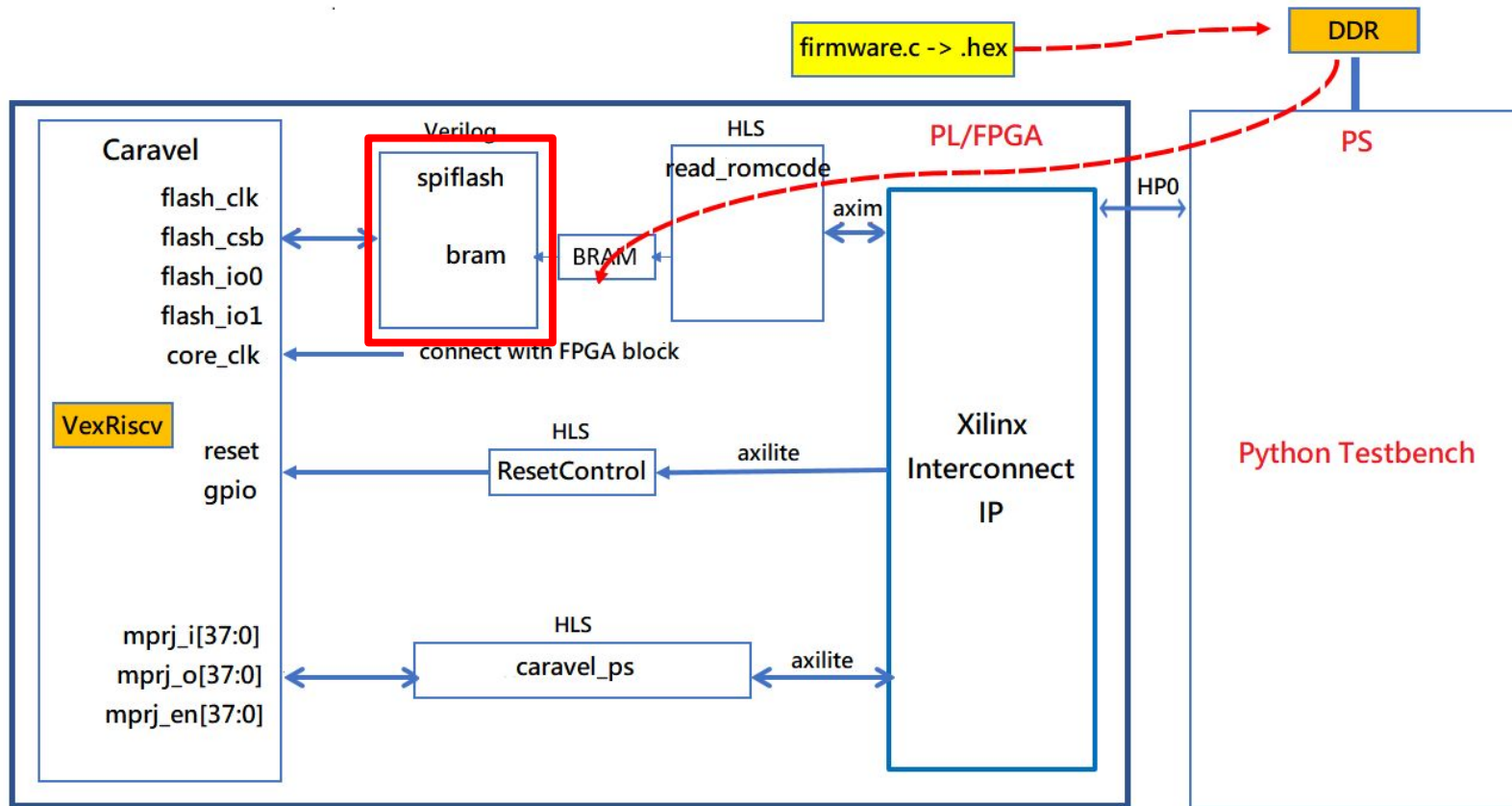


Create Reset

```
set rst_ps7_0_10M [ create_bd_cell -type ip -vlnv xilinx.com:ip:proc_sys_reset:5.0  
rst_ps7_0_10M ]
```

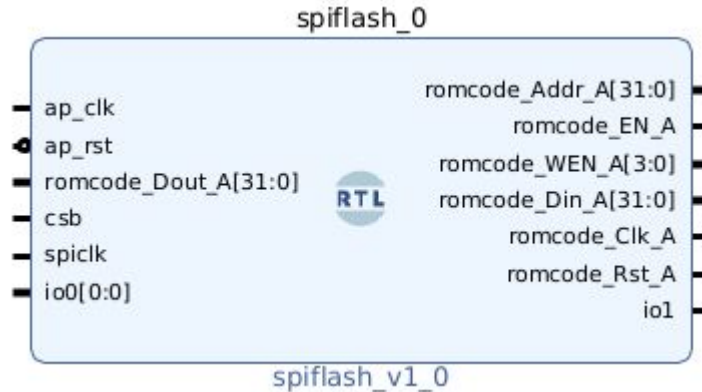


Create SPI Flash

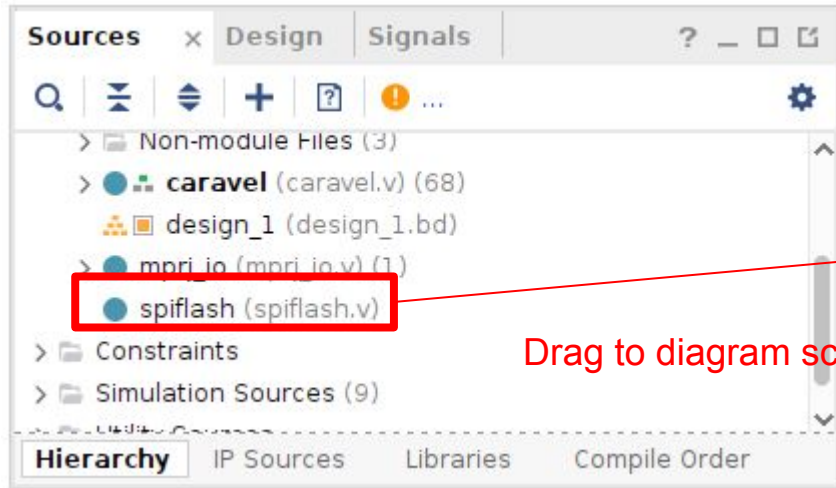


Create SPI Flash

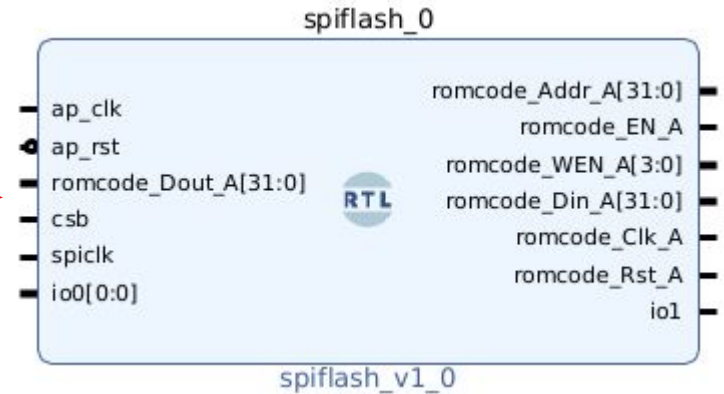
```
set block_name spiflash
set block_cell_name spiflash_0
set spiflash_0 [create_bd_cell -type module -reference $block_name $block_cell_name]
```



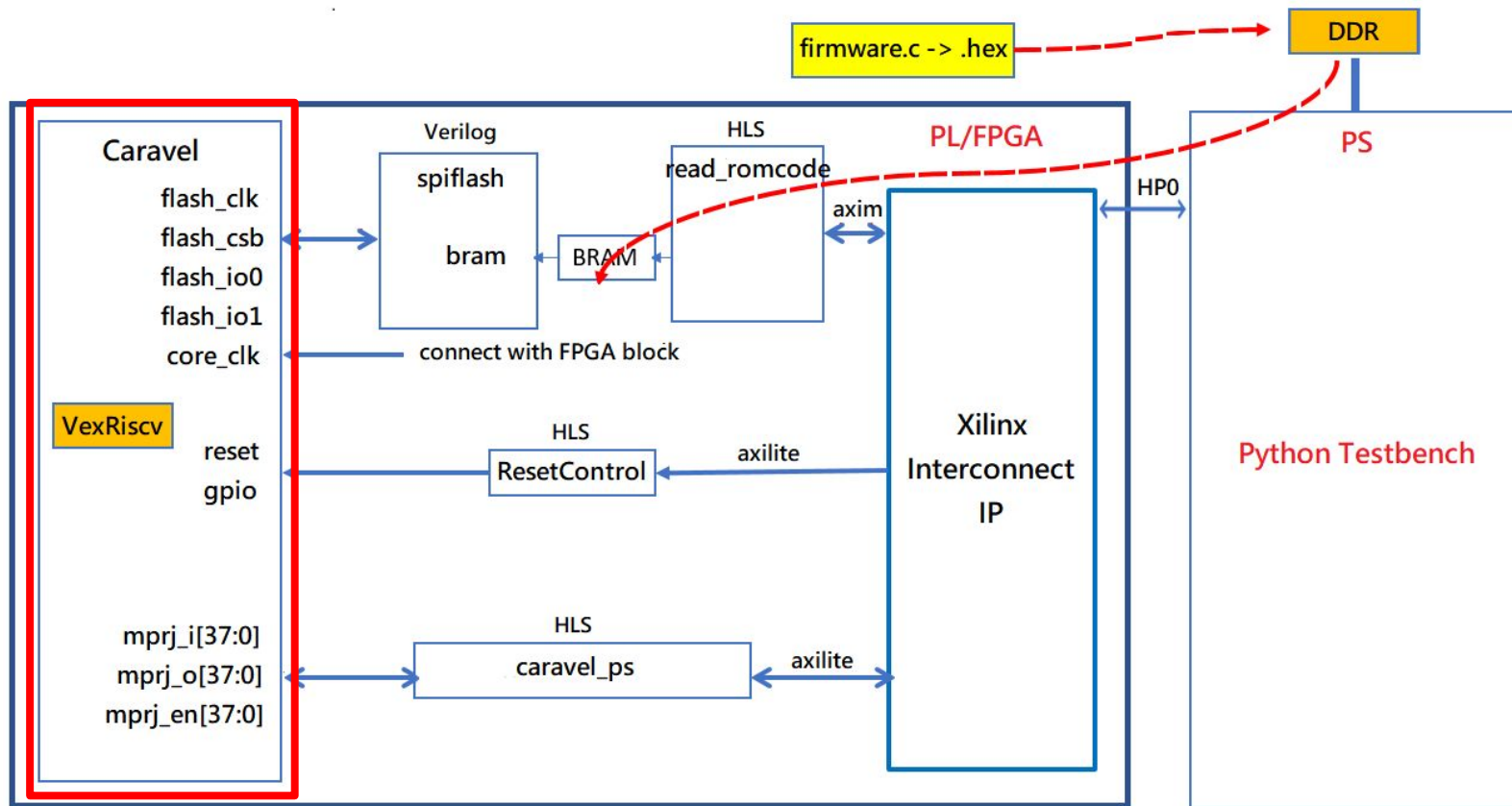
Create SPI Flash without TCL



Drag to diagram scope

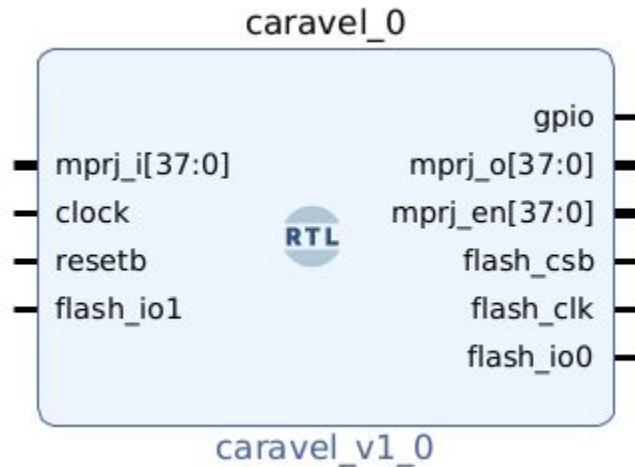


Create Caravel



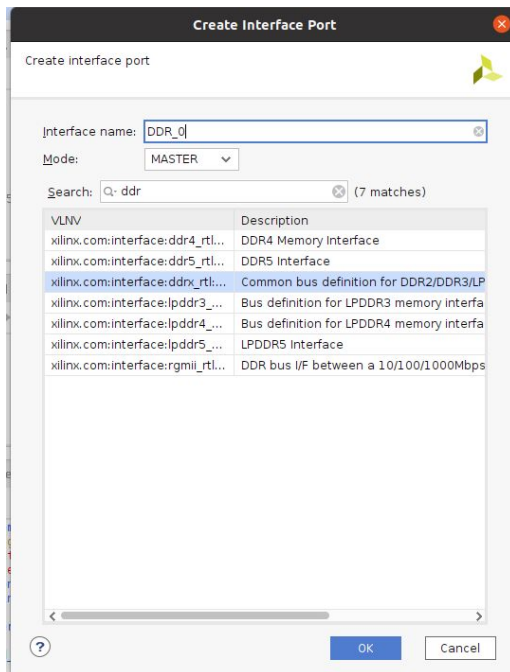
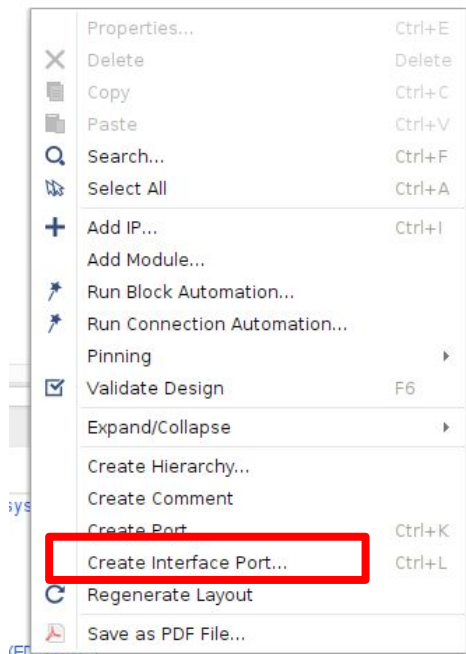
Create Caravel

```
set block_name caravel
set block_cell_name caravel_0
set caravel_0 [create_bd_cell -type module -reference $block_name $block_cell_name]
```



Create DDR Port and IO Port for PS

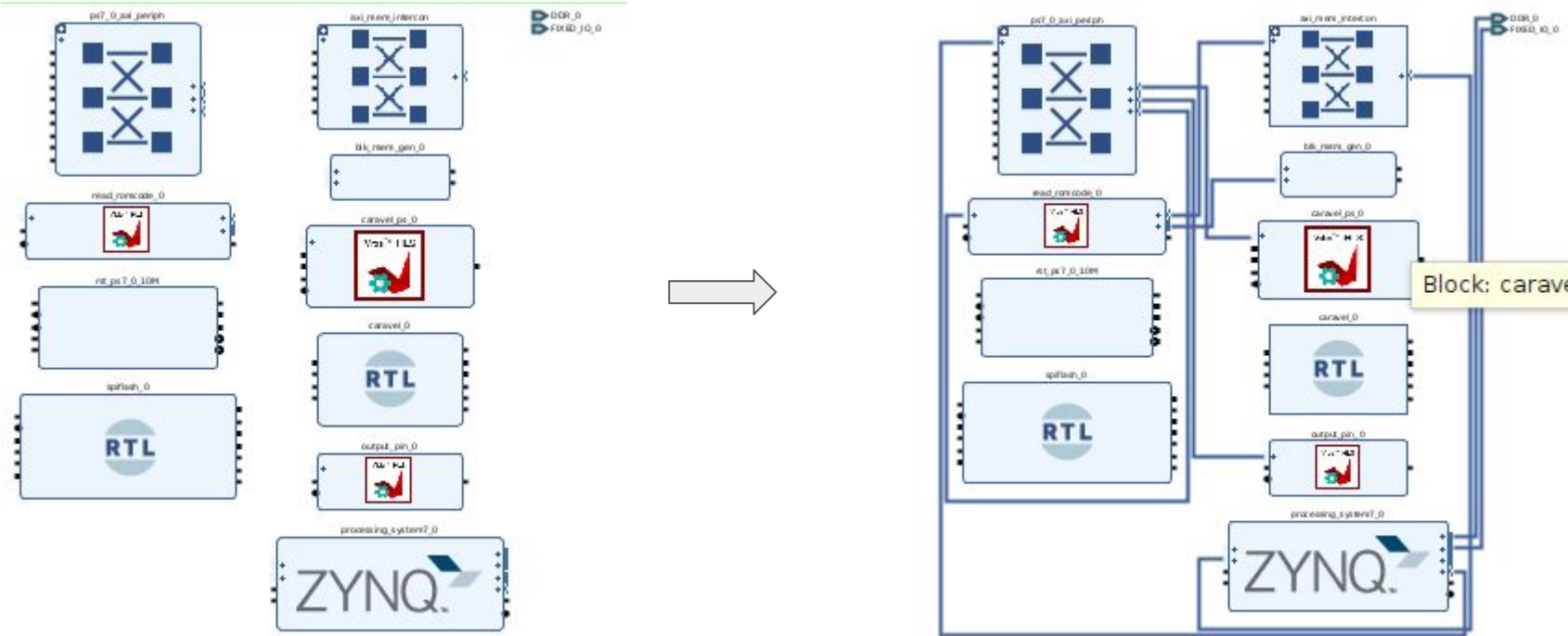
```
set DDR_0 [ create_bd_intf_port -mode Master -vlnv xilinx.com:interface:ddrx_rtl:1.0 DDR_0 ]  
set FIXED_IO_0 [ create_bd_intf_port -mode Master -vlnv  
xilinx.com:display_processing_system7:fixedio_rtl:1.0 FIXED_IO_0 ]
```



Create Interface Connection

```
1 connect_bd_intf_net -intf_net axi_mem_intercon_M00_AXI [get_bd_intf_pins axi_mem_intercon/M00_AXI]
[get_bd_intf_pins processing_system7_0/S_AXI_HP0]
2 connect_bd_intf_net -intf_net processing_system7_0_DDR [get_bd_intf_ports DDR_0] [get_bd_intf_pins
processing_system7_0/DDR]
3 connect_bd_intf_net -intf_net processing_system7_0_FIXED_IO [get_bd_intf_ports FIXED_IO_0]
[get_bd_intf_pins processing_system7_0/FIXED_IO]
4 connect_bd_intf_net -intf_net processing_system7_0_M_AXI_GP0 [get_bd_intf_pins
processing_system7_0/M_AXI_GP0] [get_bd_intf_pins ps7_0_axi_periph/S00_AXI]
5 connect_bd_intf_net -intf_net ps7_0_axi_periph_M00_AXI [get_bd_intf_pins caravel_ps_0/s_axi_control]
[get_bd_intf_pins ps7_0_axi_periph/M00_AXI]
6 connect_bd_intf_net -intf_net ps7_0_axi_periph_M01_AXI [get_bd_intf_pins output_pin_0/s_axi_control]
[get_bd_intf_pins ps7_0_axi_periph/M01_AXI]
7 connect_bd_intf_net -intf_net ps7_0_axi_periph_M02_AXI [get_bd_intf_pins ps7_0_axi_periph/M02_AXI]
[get_bd_intf_pins read_romcode_0/s_axi_control]
8 connect_bd_intf_net -intf_net read_romcode_0_internal_bram_PORTA [get_bd_intf_pins
blk_mem_gen_0/BRAM_PORTB] [get_bd_intf_pins read_romcode_0/internal_bram_PORTA]
9 connect_bd_intf_net -intf_net read_romcode_0_m_axi_BUS0 [get_bd_intf_pins axi_mem_intercon/S00_AXI]
[get_bd_intf_pins read_romcode_0/m_axi_BUS0]
```

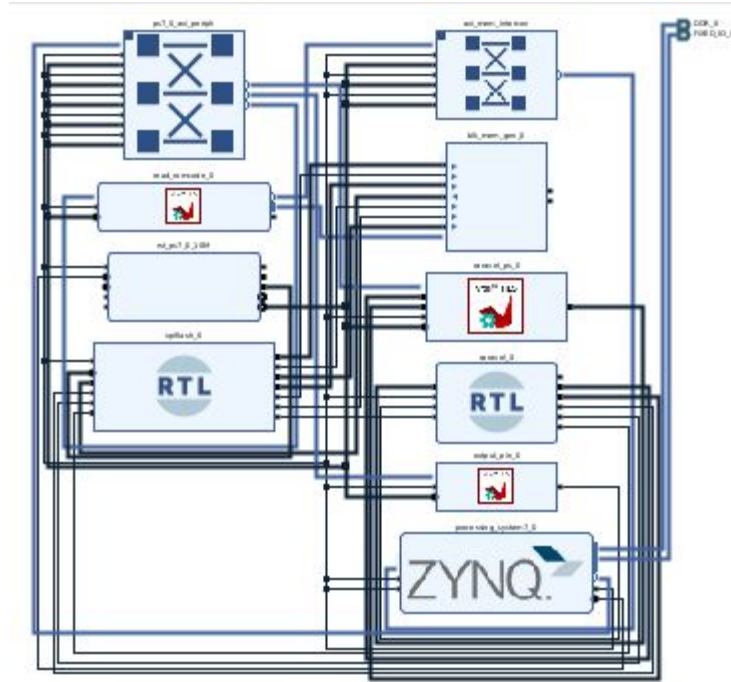
Create Interface Connection



Create Port Connection

```
1 connect_bd_net -net blk_mem_gen_0_douta [get_bd_pins blk_mem_gen_0/douta] [get_bd_pins spiflash_0/romcode_Dout_A]
2 connect_bd_net -net caravel_0_flash_clk [get_bd_pins caravel_0/flash_clk] [get_bd_pins spiflash_0/spiclk]
3 connect_bd_net -net caravel_0_flash_csb [get_bd_pins caravel_0/flash_csb] [get_bd_pins spiflash_0/csb]
4 connect_bd_net -net caravel_0_flash_io0 [get_bd_pins caravel_0/flash_io0] [get_bd_pins spiflash_0/io0]
5 connect_bd_net -net caravel_0_mprj_en [get_bd_pins caravel_0/mprj_en] [get_bd_pins caravel_ps_0/mprj_en]
6 connect_bd_net -net caravel_0_mprj_o [get_bd_pins caravel_0/mprj_o] [get_bd_pins caravel_ps_0/mprj_out]
7 connect_bd_net -net caravel_ps_0_mprj_in [get_bd_pins caravel_0/mprj_i] [get_bd_pins caravel_ps_0/mprj_in]
8 connect_bd_net -net output_pin_0_outpin [get_bd_pins caravel_0/resetb] [get_bd_pins output_pin_0/outpin]
9 connect_bd_net -net processing_system7_0_FCLK_CLK1 [get_bd_pins axi_mem_intercon/ACLK] [get_bd_pins axi_mem_intercon/M00_ACLK] [get_bd_pins
axi_mem_intercon/S00_ACLK] [get_bd_pins caravel_0/clock] [get_bd_pins caravel_ps_0/ap_clk] [get_bd_pins output_pin_0/ap_clk] [get_bd_pins
processing_system7_0/FCLK_CLK0] [get_bd_pins processing_system7_0/M_AXI_GP0_ACLK] [get_bd_pins processing_system7_0/S_AXI_HP0_ACLK]
[get_bd_pins ps7_0_axi_periph/ACLK] [get_bd_pins ps7_0_axi_periph/M00_ACLK] [get_bd_pins ps7_0_axi_periph/M01_ACLK] [get_bd_pins
ps7_0_axi_periph/M02_ACLK] [get_bd_pins ps7_0_axi_periph/S00_ACLK] [get_bd_pins read_romcode_0/ap_clk] [get_bd_pins
rst_ps7_0_10M/slowest_sync_clk] [get_bd_pins spiflash_0/ap_clk]
10 connect_bd_net -net processing_system7_0_FCLK_RESET0_N [get_bd_pins processing_system7_0/FCLK_RESET0_N] [get_bd_pins
rst_ps7_0_10M/ext_reset_in]
11 connect_bd_net -net rst_ps7_0_100M_peripheral_aresetn [get_bd_pins axi_mem_intercon/ARESETN] [get_bd_pins axi_mem_intercon/M00_ARESETN]
[get_bd_pins axi_mem_intercon/S00_ARESETN] [get_bd_pins caravel_ps_0/ap_rst_n] [get_bd_pins output_pin_0/ap_rst_n] [get_bd_pins
ps7_0_axi_periph/ARESETN] [get_bd_pins ps7_0_axi_periph/M00_ARESETN] [get_bd_pins ps7_0_axi_periph/M01_ARESETN] [get_bd_pins
ps7_0_axi_periph/M02_ARESETN] [get_bd_pins ps7_0_axi_periph/S00_ARESETN] [get_bd_pins read_romcode_0/ap_rst_n] [get_bd_pins
rst_ps7_0_10M/peripheral_aresetn]
12 connect_bd_net -net rst_ps7_0_50M_peripheral_reset [get_bd_pins rst_ps7_0_10M/peripheral_reset] [get_bd_pins spiflash_0/ap_rst]
13 connect_bd_net -net spiflash_0_io1 [get_bd_pins caravel_0/flash_io1] [get_bd_pins spiflash_0/io1]
14 connect_bd_net -net spiflash_0_romcode_Addr_A [get_bd_pins blk_mem_gen_0/addr_a] [get_bd_pins spiflash_0/romcode_Addr_A]
15 connect_bd_net -net spiflash_0_romcode_Clk_A [get_bd_pins blk_mem_gen_0/clka] [get_bd_pins spiflash_0/romcode_Clk_A]
16 connect_bd_net -net spiflash_0_romcode_Din_A [get_bd_pins blk_mem_gen_0/dina] [get_bd_pins spiflash_0/romcode_Din_A]
17 connect_bd_net -net spiflash_0_romcode_EN_A [get_bd_pins blk_mem_gen_0/ena] [get_bd_pins spiflash_0/romcode_EN_A]
18 connect_bd_net -net spiflash_0_romcode_Rst_A [get_bd_pins blk_mem_gen_0/rsta] [get_bd_pins spiflash_0/romcode_Rst_A]
19 connect_bd_net -net spiflash_0_romcode_WEN_A [get_bd_pins blk_mem_gen_0/wea] [get_bd_pins spiflash_0/romcode_WEN_A]
```

Create Port Connection



Create Address Segments

```
1 assign_bd_address -offset 0x40000000 -range 0x00010000 -target_address_space  
[get_bd_addr_spaces processing_system7_0/Data] [get_bd_addr_segs  
caravel_ps_0/s_axi_control/Reg] -force  
2 assign_bd_address -offset 0x40010000 -range 0x00010000 -target_address_space  
[get_bd_addr_spaces processing_system7_0/Data] [get_bd_addr_segs  
output_pin_0/s_axi_control/Reg] -force  
3 assign_bd_address -offset 0x40020000 -range 0x00010000 -target_address_space  
[get_bd_addr_spaces processing_system7_0/Data] [get_bd_addr_segs  
read_romcode_0/s_axi_control/Reg] -force  
4 assign_bd_address -offset 0x00000000 -range 0x20000000 -target_address_space  
[get_bd_addr_spaces read_romcode_0/Data_m_axi_BUS0] [get_bd_addr_segs  
processing_system7_0/S_AXI_HP0/HP0_DDR_LOWOCM] -force
```

Create Address Segments

- Open address editor

Network 0							
/processing_system7_0							
/processing_system7_0/Data (32 address bits : 0x40000000 [1G])							
/caravel_ps_0/s_axi_control	s_axi_control	Reg	0x4000_0000		64k		0x4000_FFFF
/output_pin_0/s_axi_control	s_axi_control	Reg	0x4001_0000		64k		0x4001_FFFF
/read_romcode_0/s_axi_control	s_axi_control	Reg	0x4002_0000		64k		0x4002_FFFF
Network 1							
/read_romcode_0							
/read_romcode_0/Data_m_axi_BUS0 (64 address bits : 16E)							
/processing_system7_0/S_AXI_HP0	S_AXI_HP0	HP0_DDR_LOWOCM	0x0000_0000_0000_0000		512		0x0000_0000_1FFF_FFF

Validate and Save Block Design

```
1 validate_bd_design  
2 save_bd_design  
3 close_bd_design $design_name
```

Make Wrapper

```
set_property REGISTERED_WITH_MANAGER "1" [get_files design_1.bd ]
set_property SYNTH_CHECKPOINT_MODE "Hierarchical" [get_files design_1.bd ]

#call make_wrapper to create wrapper files
if { [get_property IS_LOCKED [ get_files -norecurse design_1.bd] ] == 1 } {
    import_files -fileset sources_1 [file normalize
"${origin_dir}/vvd_caravel_fpga/vvd_caravel_fpga.gen/sources_1/bd/design_1/hdl/design_1_wrapper.v" ]
} else {
    set wrapper_path [make_wrapper -fileset sources_1 -files [ get_files -norecurse design_1.bd] -top]
    add_files -norecurse -fileset sources_1 $wrapper_path
}
```

