

7. add this in, the _v1_0.v file, if you are unsure about this step, refer to the github where its uploaded:

a. At line 7:

parameter WIDTH=32, SIZE=512, //WIDTH is bits per word(shouldn't be
changed), SIZE is # of WORDS

parameter NUM_COL = 4,

parameter COL_WIDTH = 8,

localparam integer LOGSIZE=\$clog2(SIZE),

b. at line 49 add: wire ps_reset;

c. add this inside the instantiation .ps_reset(ps_reset)

d. output ps_reset, -> add at line 18

 assign ps_reset = slv_reg0; -> add this to line 117

Add Interface

Use the tabs and fields below to modify the Bus Interface on your IP.

GeneralPort MappingParameters

Interface Definition:

bram_rtl

Name:

BRAM_CTRL

Mode:

slave

Display name:

Description:

Interface presence:

☒ Mandatory☐ Optional

Example: \$my_num_var > 0

more info

OK

Cancel

Mapped Ports Summary

Logical Port	Physical Port
ADDR	bram_addr
WE	bram_we
DIN	bram_wrdata
DOUT	bram_rddata

Use the tabs and fields below to modify the Bus Interface on your IP.

Parameters

Choose Parameters to Override

Search:

- > Auto-calculated
- Requires User Setting
- Optional

Name	Description	Value
Overridden		
MASTER_TYPE		BRAM_CTRL
READ_WRITE_MODE		READ_WRITE
User Set		
Optional		

Package IP - integrated_MMM_pipeline

Ports and Interfaces

Name	Interface Mode	Enablement	Is Declaration	Access Handle	Access Type	Direction	Driver Value	Size Left	Size Right	Size Left Dependency	Size Dep
s00_axi	slave										
bram_data	slave				ref	in		31	0	(WIDTH-1)	
bram_we					ref	out		31	0	(WIDTH-1)	
bram_addr					ref	in		3	0	(ADDR_COL-1)	

IP Port Properties

Port: bram_addr

size left: 11

size left dependency: (ADDR_COL-1)

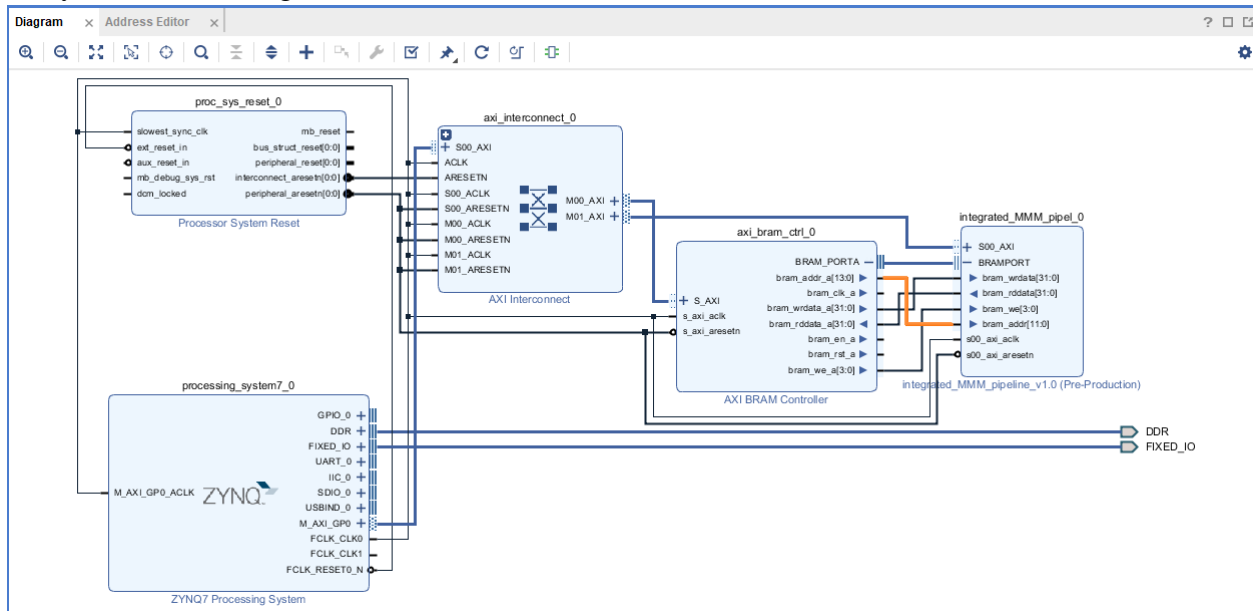
Tcl Console

```
ERROR: [IP_Flow 19-3428] Failed to create Customization object integrated_MMM_pipeline_0
CRITICAL WARNING: [IP_Flow 19-973] Failed to create IP instance 'integrated_MMM_pipeline_0'. Error during customization.
ERROR: [Common 17-39] 'ipgenopen_ipgen_file' failed due to earlier errors.
export IC_ALL="es_VS_VTF-0"
ambiguous command name "export": export_as_example_design export_bd_synth export_ip_user_files export_simulation
set_property driver_value 12 [ipgen_ports bram_addr -of_objects [ipgen_objects]]
```

Edit bram_addr in ip port properties for size left and make it 11.

8. merge changes in the packaging steps tabs and package IP.

9. By now, Block diagram should look like this:



Validate design to make sure there is no errors.

Lastly, create HDL wrapper

SDK STUFF

File -> export hardware -> launch sdk

In the SDK:

Change linker in the application project under the source (src) folder, open lscript.ld (LINKER SCRIPT).

Under, section to memory region mapping, Change all region memory fields from ps7_DDR_0 to PS7_RAM_0

In case there are weird errors such as an assembly loop or when trying to running the program (AXI error):

1. Create new Application project called "FSBL", choose zynq FSBL project as the template. Hit Finish.
2. Select the FSBL in project explorer on the left side, in top menu toolbar -> Xilinx-> create boot image-> on the bottom (boot image partitions) -> click add -> browse file path -> navigate to original application project
(./Project.sdk/application_project_name/debug/application_name.elf) -> open elf and hit OK

3. Create image
4. With FSBL still highlighted, program and run. Then run application project and proceed as normal.

For UART and SDK testing

RUN THE PYTHON AND THEN THE C FILE IN THE SDK