



Synaptic Labs'

xSPI-MC Controller Upgrade Notes

Introduction

This short document outlines the basic steps for upgrading any Quartus project in order to use SLL xSPI Multi-Bus Memory Controller (xSPI-MC) instead of SLL MBMC IP.

Step 1: Remove SLL MBMC Qsys Component from the project IP Folder

- Remove any copy of SLL MBMC IP bundle from your project ip installation folder.

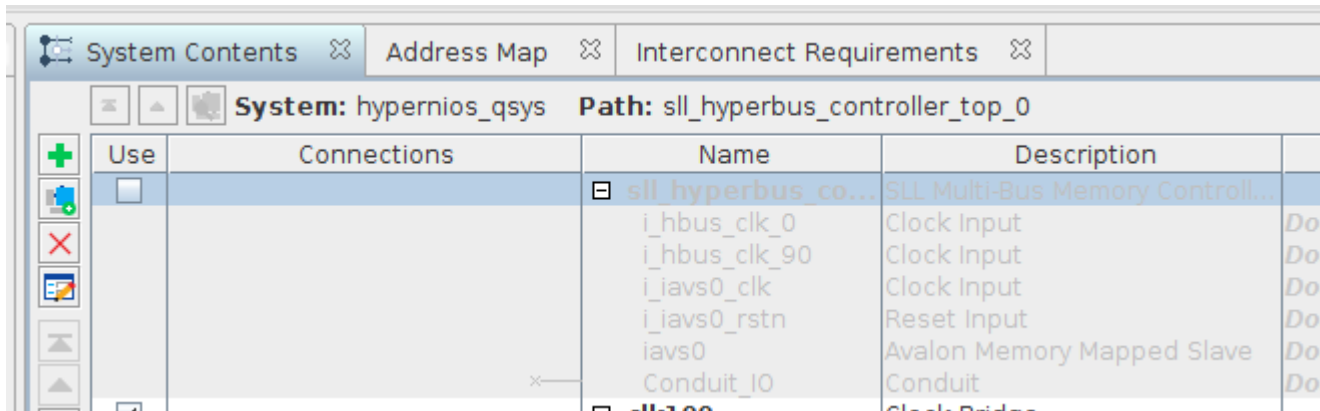
Step 2: License Setup

- Next you need to apply for Synaptic Labs' xSPI-MC license. You can skip this step if you already installed the license at some earlier stage. Contact Synaptic Labs' on info@synaptic-labs.com
- Install xSPI-MC license in Quartus License Setup.

Step 3: Install xSPI-MC Qsys Component into the project IP Folder

1. In this tutorial we assume that SLL xSPI Multi-Bus Memory Controller (xSPI-MC) will be located in the Project directory.
2. Contact Synaptic Labs' for the latest version of Synaptic Labs' xSPI-MC IP
3. Copy SLL XSPI-MC IP into the **project/ip** folder or IP installation folder.

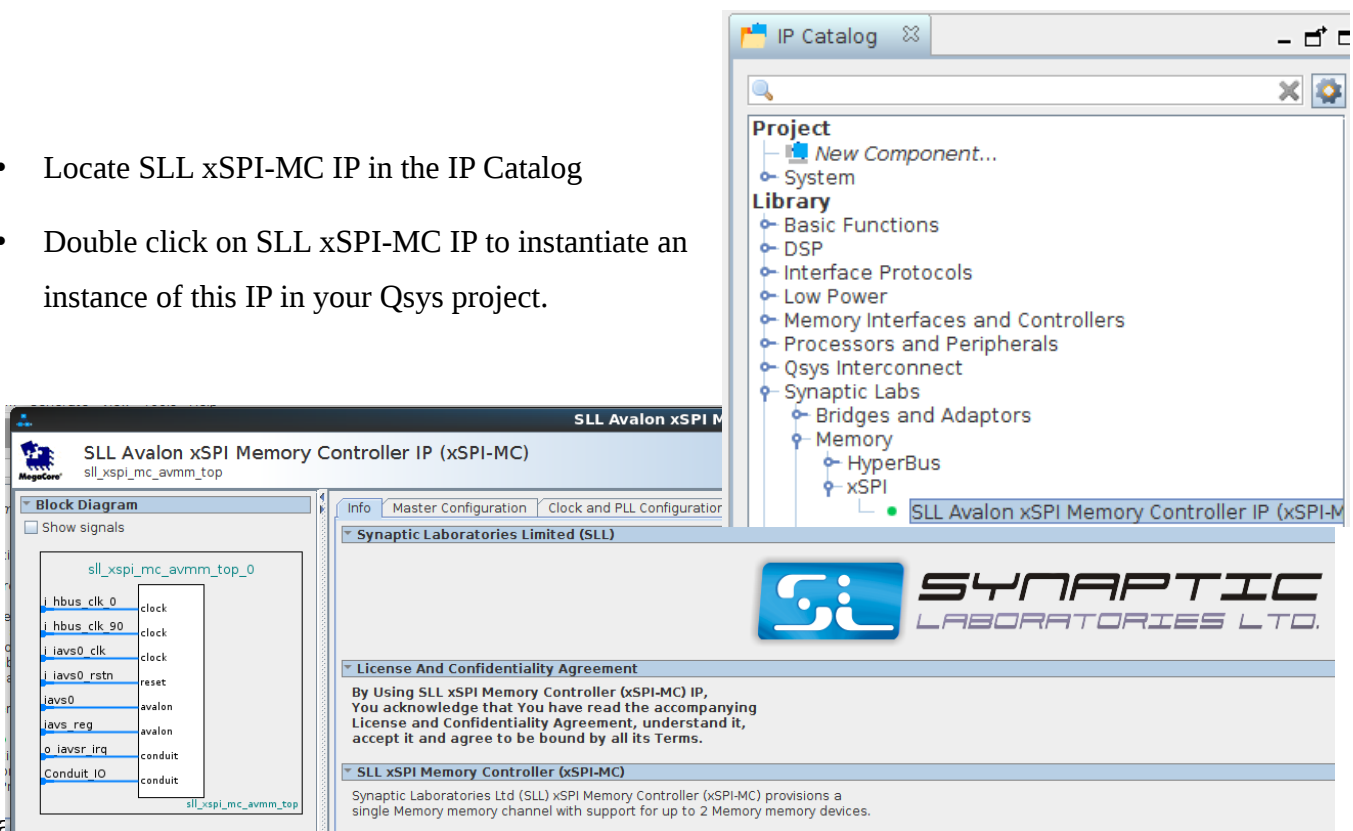
Step 4: Open Qsys and delete SLL MBMC IP



- Initially you can Un-tick the [] USE button in the System Contents Window.
- Ignore any errors at this point

Step 5: Add SLL xSPI-MC IP to your design

- Locate SLL xSPI-MC IP in the IP Catalog
- Double click on SLL xSPI-MC IP to instantiate an instance of this IP in your Qsys project.



Step 6: Configure SLL xSPI-MC IP in the Master User Interface Tab

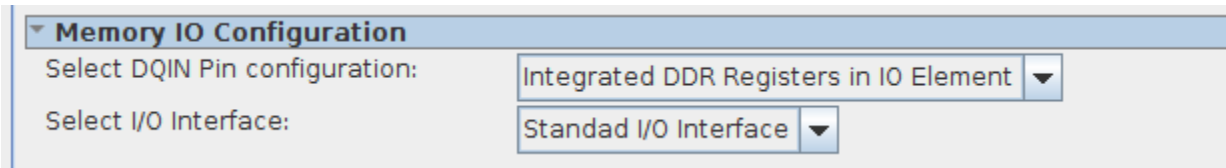
- In this section, the user selects the memory subsystem based on the development board.

The screenshot displays the 'Master Configuration' tab of a configuration tool. The 'Memory Channel Configuration' section is expanded, showing settings for two memory channels. Channel 0 is configured with 'HyperFlash' protocol and 'S26KS512S (Cypress) 1.8V HyperFlash 166 MHz 512 Mbit' device. Channel 1 is configured with 'XccelaRAM' protocol and 'APS12808L OB (AP) 1.8V XccelaRAM 200 Mhz 128 Mbit' device. Other sections include 'Memory IO Configuration' (DDR registers, standard I/O interface), 'Memory Powerup Configuration' (150 us delay), 'Memory RWDS-DQ SKEW Configuration' (810 ps skew), and 'Avalon Target Interface Configuration' (disabled ports).

Section	Parameter	Value
Memory Channel Configuration	Number of Parallel Memory channels:	1
	Memory Channel 0 Configuration	
Memory Channel 0 Configuration	Protocol 0:	HyperFlash
	Memory device on chip select 0:	S26KS512S (Cypress) 1.8V HyperFlash 166 MHz 512 Mbit
Memory Channel 1 Configuration	Protocol 1:	XccelaRAM
	Memory device on chip select 1:	APS12808L OB (AP) 1.8V XccelaRAM 200 Mhz 128 Mbit
Memory IO Configuration	Select DQIN Pin configuration:	Integrated DDR Registers in IO Element
	Select I/O Interface:	Standad I/O Interface
Memory Powerup Configuration	PowerUp Delay Timer:	150 us
Memory RWDS-DQ SKEW Configuration	Enable Default RWDS-DQ SKEW settings	<input checked="" type="checkbox"/>
	DQ-DQS skew on channel 0 in ps:	810 ps
Avalon Target Interface Configuration	Avalon control port:	Disabled
	Include PSRAM ECC input signal:	Disabled

- Memory Device 0 Configuration** : Selects the Flash memory device present on the FPGA board. Currently, SLL xSPI-MC supports the following protocols :HyperFlash 1.0, SemperFlash, OctaFlash and xSPI Profile 1.0 Flash. If no device is present select **None**. Kindly contact Synaptic Labs' (info@synaptic-labs.com) in case the Flash device present on your FPGA board is not available in the list.
- Memory Device 1 Configuration** : Selects the PSRAM memory device present on the FPGA board. Currently, SLL xSPI-MC supports the following protocols :HyperRAM 1.0, HyperRAM 2.0, OctaRAM, XccelaRAM and xSPI Profile 2.0 PSRAM. If no device is present select **None**. Kindly contact Synaptic Labs' (info@synaptic-labs.com) in case the PSRAM device present on your FPGA board is not available in the list.

Memory IO Configuration



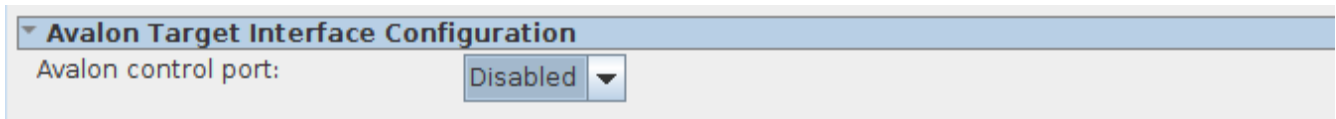
The screenshot shows a configuration panel titled "Memory IO Configuration". It contains two dropdown menus. The first is labeled "Select DQIN Pin configuration:" and is set to "Integrated DDR Registers in IO Element". The second is labeled "Select I/O Interface:" and is set to "Standad I/O Interface".

The user can select can either select

- Direct Connection or
- Integrated DDR Register in the IO Element (preferred option)

The Integrated DDR feature can be used for all device families (not just Cyclone V). The Integrated DDR feature instantiates the PHY in a way that enables the DDR FF inside the I/O Elements. The Integrated DDR Register in the IO Element option allows for potential higher clock speeds to be achieved.

Avalon Target interface Configuration



The screenshot shows a configuration panel titled "Avalon Target Interface Configuration". It contains a dropdown menu labeled "Avalon control port:" which is set to "Disabled".

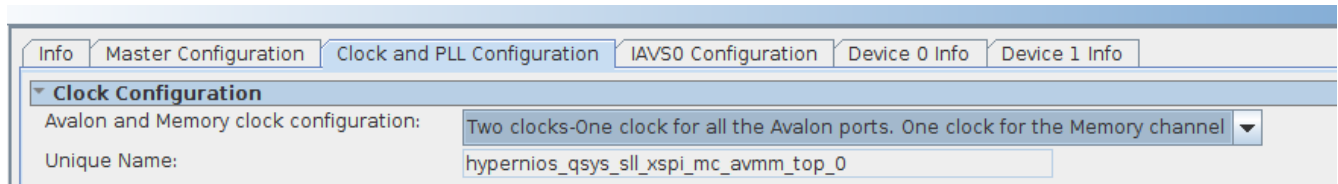
When enabled, a secondary Avalon-MM port is created.

This can be used for programming Flash or accessing pseudoRAM registers.

Configuring SLL xSPI-MC IP in the Clock and PLL Configuration Tab

The user should select the same configuration settings as used with SLL MBMC IP.

The figure below shows an example when the PSRAM/Flash memory channel is operating at a different frequency than the Avalon Memory slave interface.,



Configuring SLL xSPI-MC IP in the IAVS Configuration Tab

The user should select the same configuration settings as used with SLL MBMC IP.

The figure below shows an example when the the Avalon Memory slave interface is configured with line wrap Burst Enabled and MAX Burst Size setting of 8.

The screenshot displays the 'IAVS0 Configuration' tab within a software interface. The tab is selected, and the configuration is organized into several sections:

- IAVS0: Ingress Avalon port stage**
 - ☒ Enable Avalon write capability
 - ☒ Enable Avalon byte-enable capability
 - Access capabilities: Read/Write
 - ☐ Register Avalon write data path (generally recommended for high clock speed designs)
- IAVS0: Ingress Avalon address/data**
 - Address width: 25 bits
 - Address units: Words
 - Word width: 32 bits
- IAVS0: Burst converter and address decoder stage**
 - maxBurstSize (in words): 8
 - linewrapBursts: true
 - burstOnBurstBoundariesOnly: false
- IAVS0: Ingress Avalon return stage**
 - ☐ Register Avalon read data path (sometimes used to increase top clock speeds)
 - ☐ Use Avalon transaction responses

Configuring SLL xSPI-MC IP in the Device 0 / Device 1 Info Tabs

Device 0 and Device 1 Info tabs are for information purpose only.

The screenshot shows the 'Device 0 Info' tab selected in a configuration tool. The 'Device 0 Parameters' section includes fields for Device (s26ks512s), Memory Type (Hyperflash), Device storage capacity (64 MBytes), Device Speed Grade (166 MHz), Device DQ data width (8), and Device Wrap Support (1). A checkbox for 'Non-default configuration settings are programmed after Power-Up' is unchecked. The 'Device 0 Timings' section includes fields for Tacc (16 cycles), Tcshi - Trwr (7 cycles), Tcss (2 cycles), and Tcsh (1 cycles).

Device 0 Parameters		
Device:	s26ks512s	
Memory Type:	Hyperflash	
Device storage capacity:	64	MBytes
Device Speed Grade:	166	MHz
Device DQ data width:	8	
Device Wrap Support:	1	
<input type="checkbox"/> Non-default configuration settings are programmed after Power-Up		

Device 0 Timings		
Tacc:	16	cycles
Tcshi - Trwr:	7	cycles
Tcss:	2	cycles
Tcsh:	1	cycles

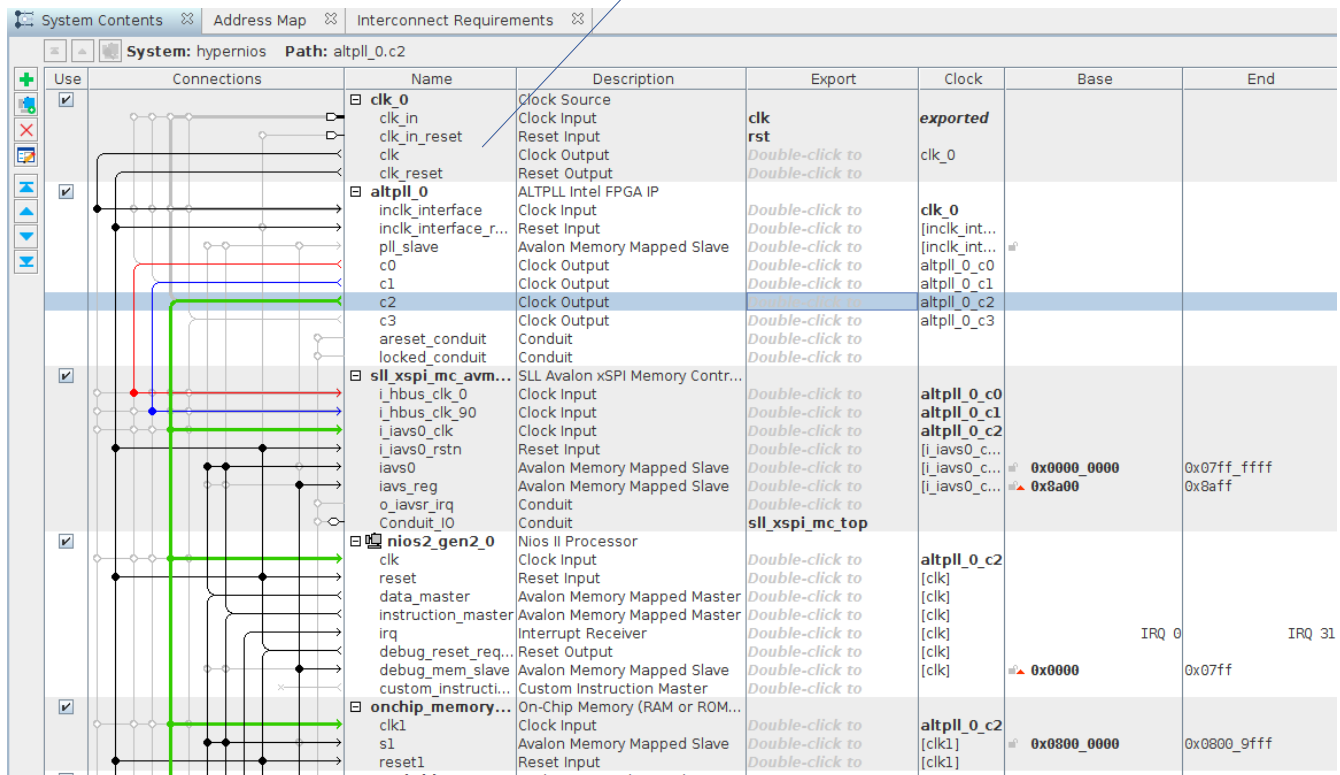
The screenshot shows the 'Device 1 Info' tab selected in a configuration tool. The 'Device 1 Parameters' section includes fields for Device (ap12808l), Memory Type (Xccelaram), Device storage capacity (16 MBytes), Device Speed Grade (200 MHz), Device DQ data width (8), and Device Wrap Support (1). A checkbox for 'Non-default configuration settings are programmed after Power-Up' is checked. The 'Device 1 Timings' section includes fields for Tacc (14 cycles), Tcshi - Trwr (8 cycles), Tcss (2 cycles), and Tcsh (2 cycles).

Device 1 Parameters		
Device:	ap12808l	
Memory Type:	Xccelaram	
Device storage capacity:	16	MBytes
Device Speed Grade:	200	MHz
Device DQ data width:	8	
Device Wrap Support:	1	
<input checked="" type="checkbox"/> Non-default configuration settings are programmed after Power-Up		

Device 1 Timings		
Tacc:	14	cycles
Tcshi - Trwr:	8	cycles
Tcss:	2	cycles
Tcsh:	2	cycles

Step 7 SLL xSPI-MC wiring

Ensure that the PLL name is altpll_0



Altera PLL output clock 0 (c0)

- Connect to i_hbus_clk_0 on SLL xSPI-MC IP

Altera PLL output clock 1 (c1)

- Connect to i_hbus_clk_90 on SLL xSPI-MC IP

Altera PLL output clock 2 (c2)

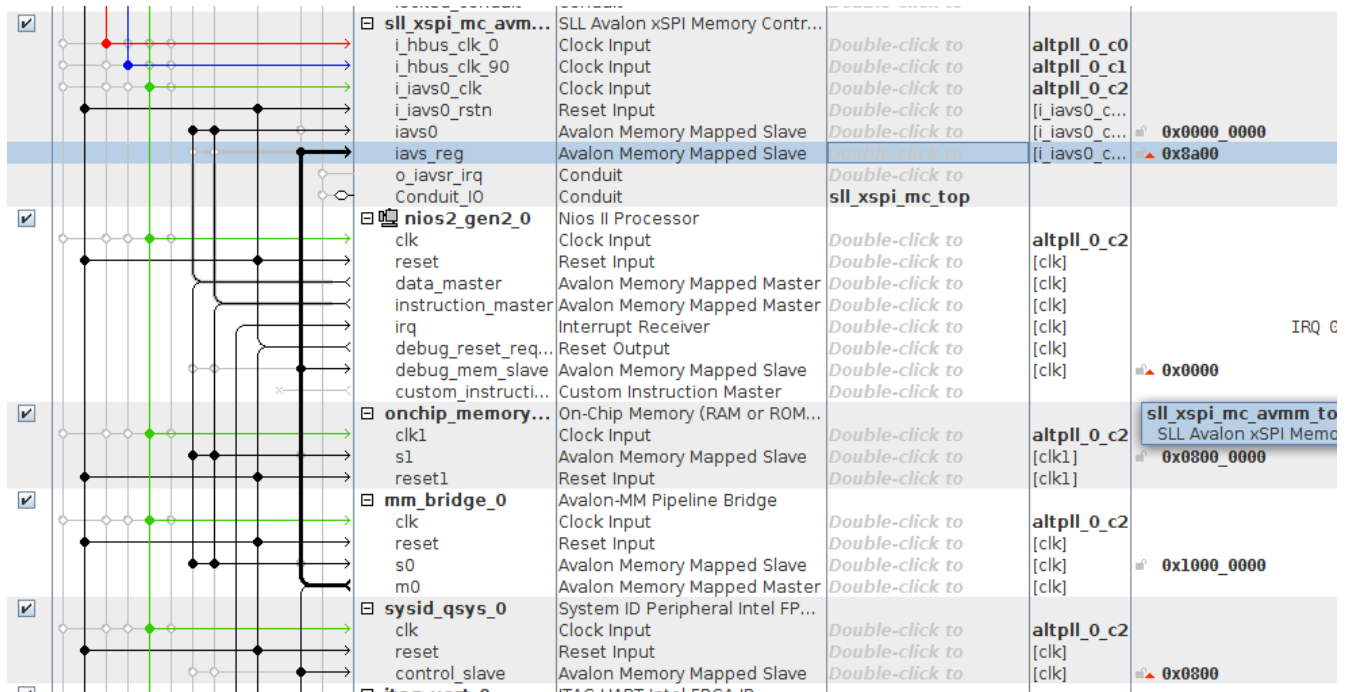
- Connect to i_iavs0_clk on SLL xSPI-MC IP
- Connect to other Avalon-MM slaves and masters clock sinks

Avalon Master/Slave Data Port Connection

- In this example, the Nios 2 Instruction and Data Avalon masters are connected on SLL xSPI-MC IP Avalon Data port (iavs). Please connect SLL xSPI-MC IP Avalon Data Port accordingly.

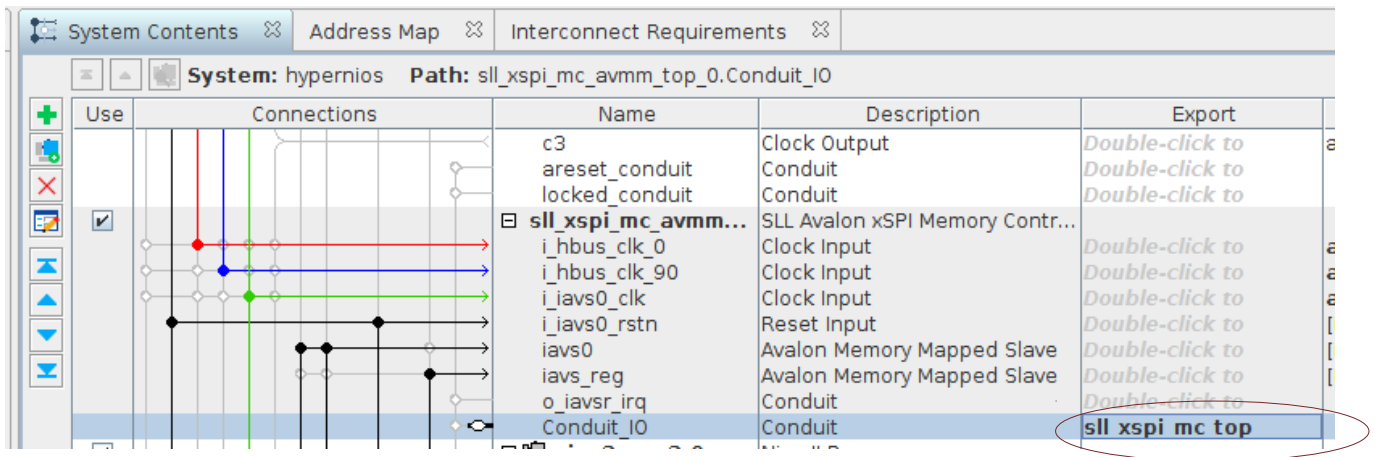
- **Avalon Master/Slave Control Port Connection**

In this example, the SLL xSPI-MC IP Avalon Control Port (iavs_reg) is connected on an Avalon bridge. Please connect SLL xSPI-MC IP Avalon Control Port accordingly. The user might not need to use this port in the design.



Step 8: Export xSPI-MC conduit

- Locate the Conduit IO in the SLL xSPI-MC instantiation
- Double click on Export conduit section



Please note that SLL xSPI-MC IP conduit signals are different than those generated by SLL MBMC IP. The user needs to change the signals accordingly in the top level module. This is described in the next step.

Step 8: Top Level Module wiring

Top level Verilog module

The figure below shows a typical connection for SLL MBMC IO signals in the top level **verilog** module . *Please note that different designs might have different IO signal names.*

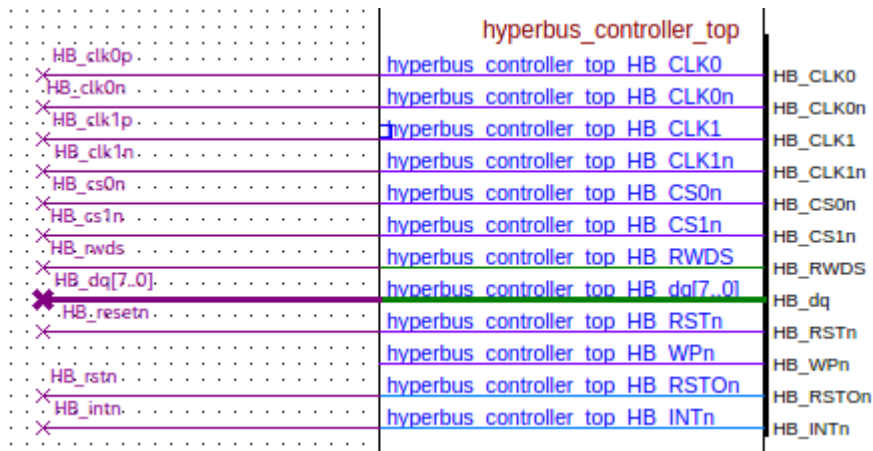
```
67
70 .hyperbus_controller_top_HB_RST0n (hbus_rst0n),
71 .hyperbus_controller_top_HB_RST1n (hbus_rst1n ),
72 .hyperbus_controller_top_HB_WPn   (          ), //not used
73 .hyperbus_controller_top_HB_INTn   (hbus_intn ),
74 .hyperbus_controller_top_HB_RWDS   (hbus_rwds ),
75 .hyperbus_controller_top_HB_dq     (hbus_dq   ),
76 .hyperbus_controller_top_HB_CLK0   (hbus_clk0p),
77 .hyperbus_controller_top_HB_CLK0n  (hbus_clk0n),
78 .hyperbus_controller_top_HB_CLK1   (          ),
79 .hyperbus_controller_top_HB_CLK1n  (          ),
80 .hyperbus_controller_top_HB_CS0n   (hbus_cs1n ),
81 .hyperbus_controller_top_HB_CS1n   (hbus_cs2n ),
82
```

SLL xSPI-MC IP has a different naming format. Please update your design to use the correct signal names as generated by the Qsys/platform Designer platform.

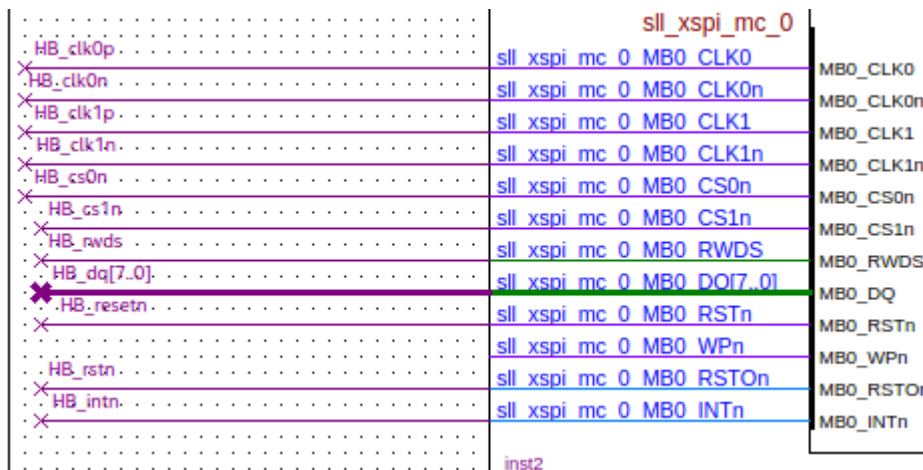
```
69
70 .sll_xspi_mc_top_MB0_RSTn   (hbus_rstn ),
71 .sll_xspi_mc_top_MB0_RWDS   (hbus_rwds ),
72 .sll_xspi_mc_top_MB0_DQ     (hbus_dq   ),
73 .sll_xspi_mc_top_MB0_CLK0   (hbus_clk0p),
74 .sll_xspi_mc_top_MB0_CLK0n  (hbus_clk0n),
75 .sll_xspi_mc_top_MB0_CLK1   (          ),
76 .sll_xspi_mc_top_MB0_CLK1n  (          ),
77 .sll_xspi_mc_top_MB0_CS0n   (hbus_cs1n ),
78 .sll_xspi_mc_top_MB0_CS1n   (hbus_cs2n ),
79 .sll_xspi_mc_top_MB0_RST0n  (hbus_rst0n ),
80 .sll_xspi_mc_top_MB0_WPn    (          ),
81 .sll_xspi_mc_top_MB0_INTn   (hbus_intn),
82
```

Top level Schematic module

The figure below shows a typical connection for SLL MBMC IO signals in the top level **schematic** module . Please note that different designs might have different IO signal names.



Please update the schematic design to use the correct signal names as generated by the Qsys/platform Designer platform.



Step 9 : Changes in NIOS II Software Built Tools for Eclipse

The user needs to re-generate the Board Support Package (BSP) for each application. This will update the **system.h** header file inside the BSP folder.

Please use the correct macro definition in your source files.

```
/*
 * sll_xspi_mc_avmm_top_0_iavs0 configuration
 */

#define ALT_MODULE_CLASS_sll_xspi_mc_avmm_top_0_iavs0 sll_xspi_mc_avmm_top
#define SLL_XSPI_MC_AVMM_TOP_0_Iavs0_BASE 0x0
#define SLL_XSPI_MC_AVMM_TOP_0_Iavs0_IRQ -1
#define SLL_XSPI_MC_AVMM_TOP_0_Iavs0_IRQ_INTERRUPT_CONTROLLER_ID -1
#define SLL_XSPI_MC_AVMM_TOP_0_Iavs0_NAME "/dev/sll_xspi_mc_avmm_top_0_iavs0"
#define SLL_XSPI_MC_AVMM_TOP_0_Iavs0_SPAN 134217728
#define SLL_XSPI_MC_AVMM_TOP_0_Iavs0_TYPE "sll_xspi_mc_avmm_top"

/*
 * sll_xspi_mc_avmm_top_0_iavs_reg configuration
 */

#define ALT_MODULE_CLASS_sll_xspi_mc_avmm_top_0_iavs_reg sll_xspi_mc_avmm_top
#define SLL_XSPI_MC_AVMM_TOP_0_Iavs_REG_BASE 0x10008a00
#define SLL_XSPI_MC_AVMM_TOP_0_Iavs_REG_IRQ -1
#define SLL_XSPI_MC_AVMM_TOP_0_Iavs_REG_IRQ_INTERRUPT_CONTROLLER_ID -1
#define SLL_XSPI_MC_AVMM_TOP_0_Iavs_REG_NAME "/dev/sll_xspi_mc_avmm_top_0_iavs_reg"
#define SLL_XSPI_MC_AVMM_TOP_0_Iavs_REG_SPAN 256
#define SLL_XSPI_MC_AVMM_TOP_0_Iavs_REG_TYPE "sll_xspi_mc_avmm_top"
```