Synaptic Labs'

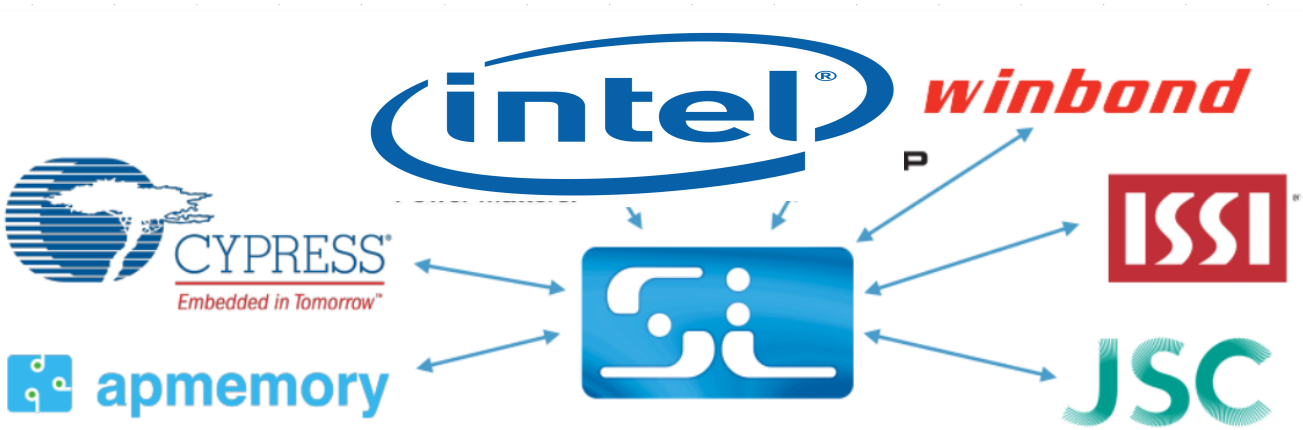
Multi-Bus Memory Controller Design Guidelines

Table of Contents

Introduction

SLL Multiple Bus Memory Controller(MBMC) supports the following Octal DDR PseudoRAMs and Flash from the following memory vendors :

- Cypress/ISSI (HyperRAM and HyperFlash)
- AP Memory (OctaRAM and Xccela RAM)
- JSC – ISSI (OctaRAM)



The pseudoRAM / Flash memories operates in Double Data Rate (DDR) mode. They have an 8-bit Serial Interface and are used for automotive semiconductor market, Wearable device market and IoT market. They can support a maximum operating frequency of 200 MHz Speed.

In general, the Pseudo DDR memory requires only 11 external bus signals. Example documentation can be found at

- https://www.cypress.com/products/PSRAM/Flash-memory
- http://www.jeju-semi.com/Products/OctaRAM
- http://www.apmemory.com/html/prod_pseudo.php

Synaptic Labs' Multiple Bus Memory Controller IP has an Avalon-MM slave interface. It supports burst mode access (up to 128 words). It can be configured in single access or burst mode access. Further more, burst mode access can be configured with either Avalon burst wrap support or Avalon burst-on-burst-boundaries support.

All external I/O pads to the memory device are generated from within Synaptic Labs' Multiple Bus Memory Controller IP. The user does not need to manually instantiate the I/O pads in the design.

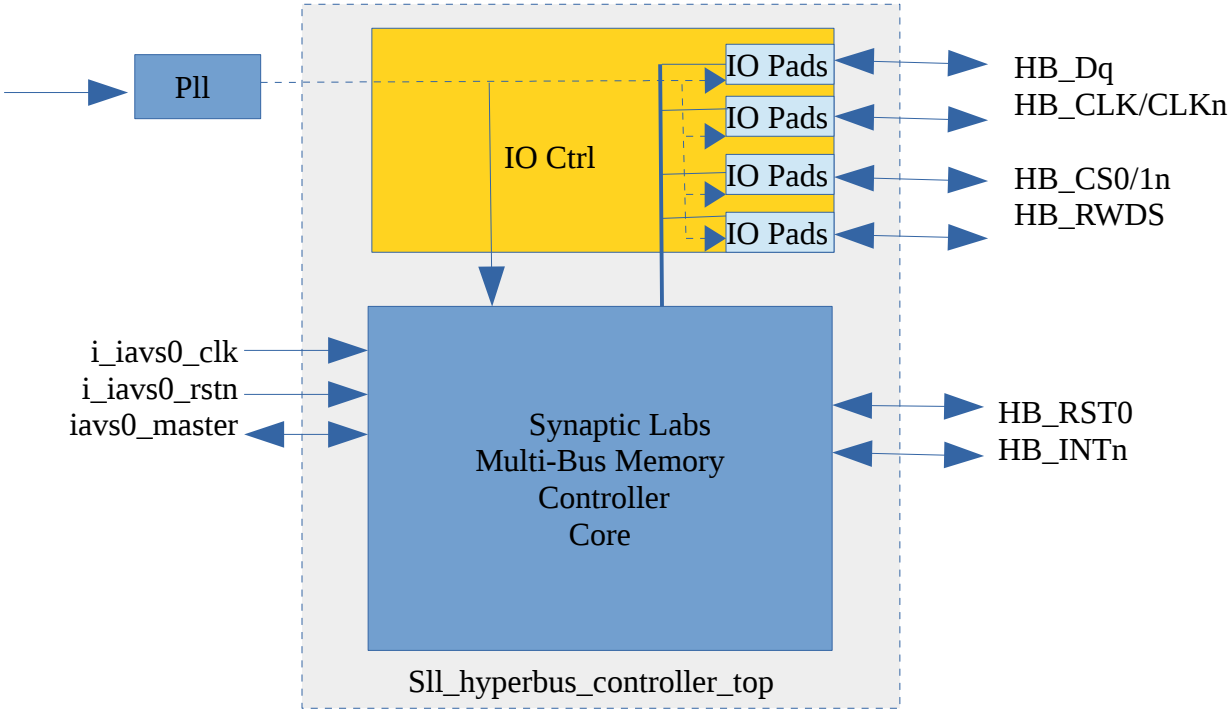
An external pll is used to generate all the necessary clocks. .

Note: Synaptic Labs' Multi-Bus memory Controller IP does NOT support DCARS functionality (Hyper-RAM PSC mode).

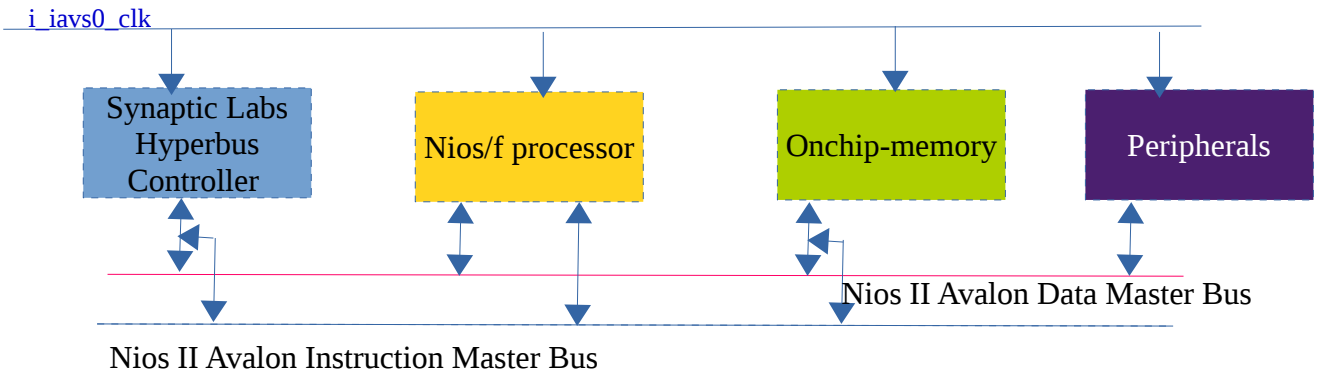
- *DCARS is a very specific capability requested by a very specific customer / chipset partner and is only supported by Cypress devices.*
- *ISSI devices do not have DCARS support.*
- *Cypress do not recommend the implementation of DCARS functionality .*
- *DCARS has a maximum frequency of 133 MHz.*

1.0 Synaptic Labs' MBMC Controller IP Qsys Component

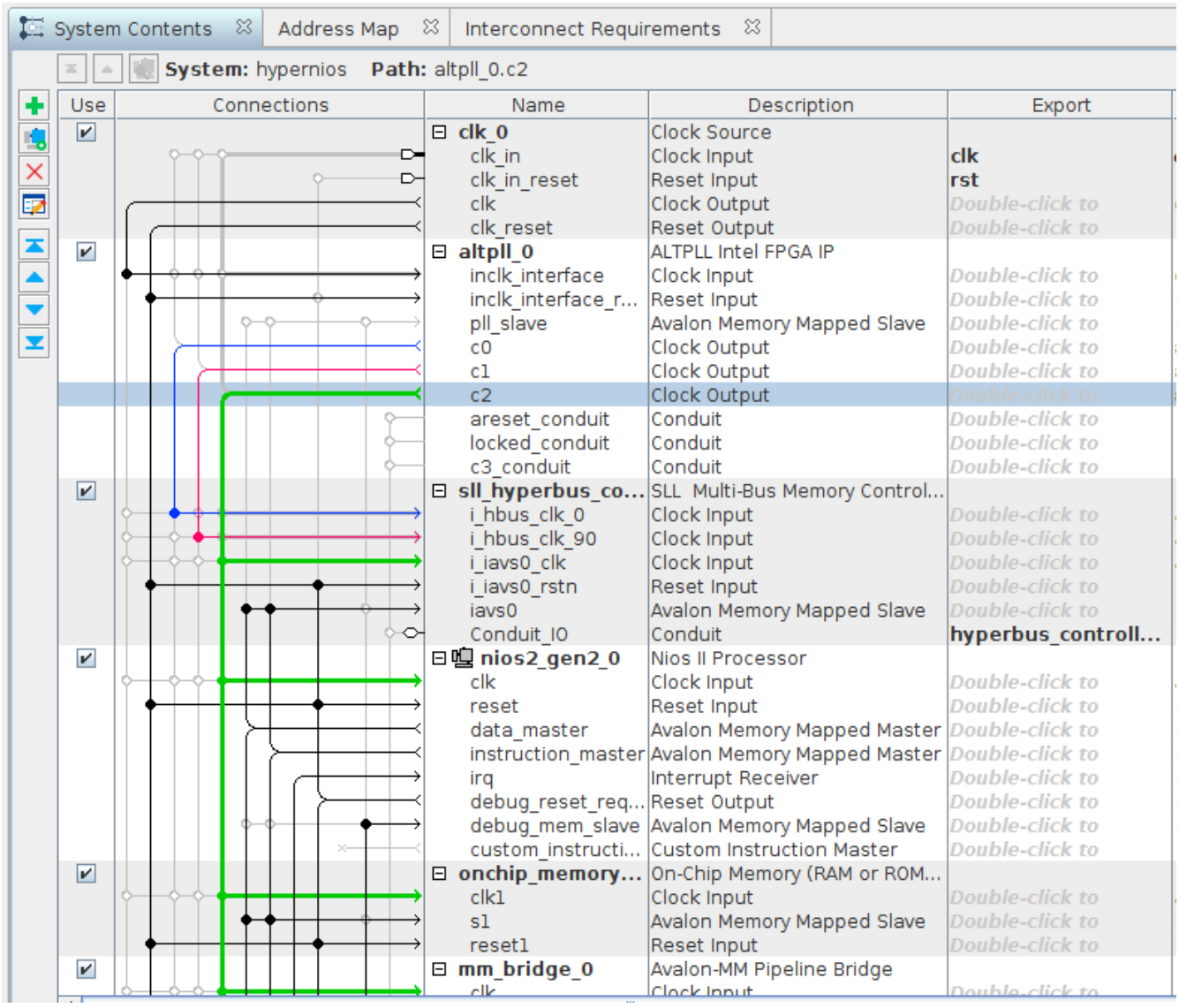
The figure below show the top level block diagram of Synaptic Labs' Multi-Bus memory controller IP.



An External clock generates the necessary signals for S/Labs' MBMC IP.



2.0 Typical SLL MBMC connection in Qsys



Please note how system component with an Avalon interface (such as NIOS II/f/e processor, onchip-memory and peripheral bridges etc), are connected to same clock. This clock is generated by the external PLL.

3.0 Typical top level signal connections (verilog/vhdl)

The figure below shows a typical top level connection for a PSRAM only design. Note that some signals are left unconnected since they are not used in the design. **For the PSRAM/HyperFlash signal names, it is very important to use the signal naming convention as suggested in section 4.1**

```
-------------------------------------
-- Instantiation of main QSys system
-------------------------------------
u0_main : component lab2
   port map (
      clk_clk       => c10_clk50m,
      reset_reset_n => c10_resetn,

      sll_hyperbus_HB_RSTn  => hbus_rstn,   --  .HB_RSTn
      sll_hyperbus_HB_CLK0  => hbus_clk0p,  --  .HB_CLK0
      sll_hyperbus_HB_CLK0n => hbus_clk0n,  --  .HB_CLK0n
      sll_hyperbus_HB_CLK1  => open,        --  .HB_CLK1
      sll_hyperbus_HB_CLK1n => open,        --  .HB_CLK1n
      sll_hyperbus_HB_CS0n  => open,        --  .HB_CS0n
      sll_hyperbus_HB_CS1n  => hbus_cs2n,   --  .HB_CS1n
      sll_hyperbus_HB_WPn   => open,        --  .HB_WPn
      sll_hyperbus_HB_RWDS  => hbus_rwds,   --  .HB_RWDS
      sll_hyperbus_HB_dq    => hbus_dq,     --  .HB_dq
      sll_hyperbus_HB_RSTOn => '1',         --  .HB_RSTOn
      sll_hyperbus_HB_INTn  => '1'          --  .HB_INTn

      pio_led_export => LEDs_debug,
   );
```

The figure below shows a typical top level connection for a HyperRAM and HyperFlash design.

```
-------------------------------------
-- Instantiation of main QSys system
-- copied from Qsys generated file
-------------------------------------
u0 : component lab2
   port map (
      clk_clk            => c10_clk50m,
      reset_reset_n      => c10_resetn,

      sll_hyperbus_HB_RSTn       => hbus_rstn,  --    .HB_RSTn
      sll_hyperbus_HB_CLK0       => hbus_clk0p, --    .HB_CLK0
      sll_hyperbus_HB_CLK0n      => hbus_clk0n, --    .HB_CLK0n
      sll_hyperbus_HB_CLK1       => hbus_clk0p, --    .HB_CLK1
      sll_hyperbus_HB_CLK1n      => hbus_clk0n, --    .HB_CLK1n
      sll_hyperbus_HB_CS0n       => hbus_cs1n,  --    .HB_CS0n
      sll_hyperbus_HB_CS1n       => hbus_cs2n,  --    .HB_CS1n
      sll_hyperbus_HB_WPn        => hbus_wpn,   --    .HB_WPn
      sll_hyperbus_HB_RWDS       => hbus_rwds,  --    .HB_RWDS
      sll_hyperbus_HB_dq         => hbus_dq,    --    .HB_dq
      sll_hyperbus_HB_RSTOn      => hbus_rston, --    .HB_RSTOn
      sll_hyperbus_HB_INTn       => hbus_intn   --    .HB_INTn--
```

4.0 Current Pin assignment

All devices connected to a memory channel must use the same Voltage standard.

In Quartus Prime Assignment editor, all pins related to the PSRAM/Flash Memories are set to 1.8 V. (This is device dependent and should be changed accordingly if a 3V PSRAM/Flash memory device is used in the design).

4.1 External MBMC Signal port names

Use the same MBMC port names as used in SLL reference designs. (This ensures that sll_ca_hbc_t001_top.sdc will automatically set the correct timing constraints). **This naming convention is mandatory.** The following signal names are recommended :

Suggested top level Port Names	**Connect to SLL MBMC IP signal**	**Description**
c10_clk50m	in_clk	Input Clock to SLL MBMC internal PLL
HB_CLK0 or hbus_clk0p	HB_CLK0	Differential clock pair 0
HB_CLK0n or hbus_clk0n	HB_CLK0n	Differential clock pair 0
HB_CLK1 or hbus_clk1p	HB_CLK1	Differential clock pair 1
HB_CLK1n or hbus_clk1n	HB_CLK1n	Differential clock pair 1
HB_CS...n or hbus_cs..n	HB_CS0n/HB_CS1n	Device 0/1 chip select
HB_dq or hbus_dq	HB_dq	PSRAM/Flash data bus
HB_RWDS or hbus_rwds	HB_RWDS	PSRAM/Flash read strobe/write mask
HB_RSTn or hbus_rstn	HB_RSTn	Reset to the PSRAM/Flash device
HB_RSTOn or hbus_rston	HB_RSTOn	Reset from the HyperFlash device
HB_WPn or hbus_wpn	HB_WPn	HyperFlash Write protect (not used)
HB_INTn or hbus_intn	HB_INTn	HyperFlash Interrupt (not used)

5.0 SLL MBMC with external PLL Configuration

SLL MBMC IP always needs an external PLL clock. SLL MBMC IP requires 2 or 3 clocks, depending on the configuration:

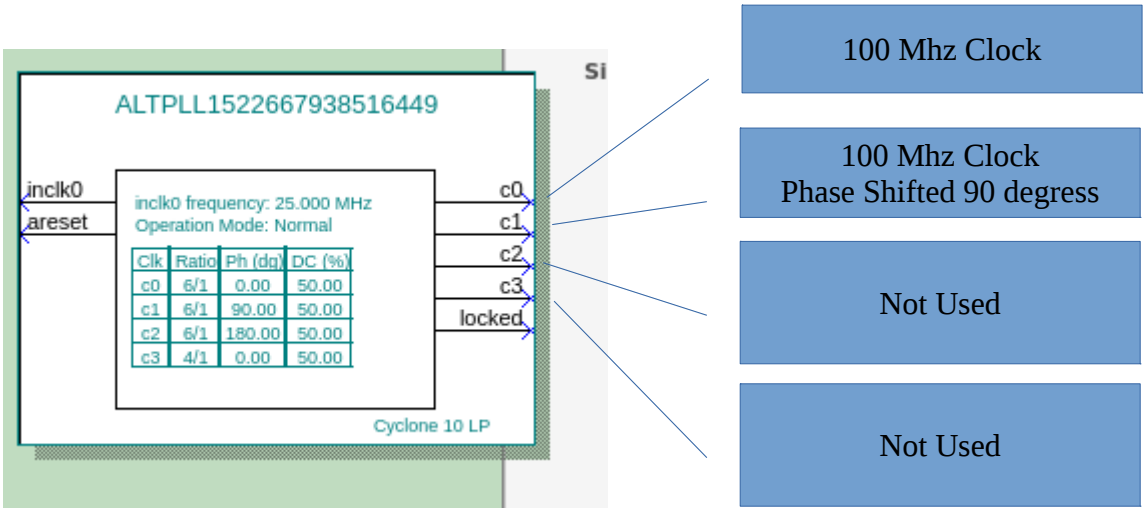
- i_hbus_clk_0 : clock driving the PSRAM/Flash controller
- i_hbus_clk_90 : clock for driving some PSRAM/Flash I/O Signals. It operates at the same frequency as i_hbus_clk_0 but is phase shifted 90 degrees
- i_iavs0_clk : clock driving the Avalon-MM interface. When SLL MBMC IP is configured to run at a single clock speed, this clock is connected to i_hbus_clk_0 clock.

5.1 Option A – Same Clock for the memory channel and Avalon-MM channel.

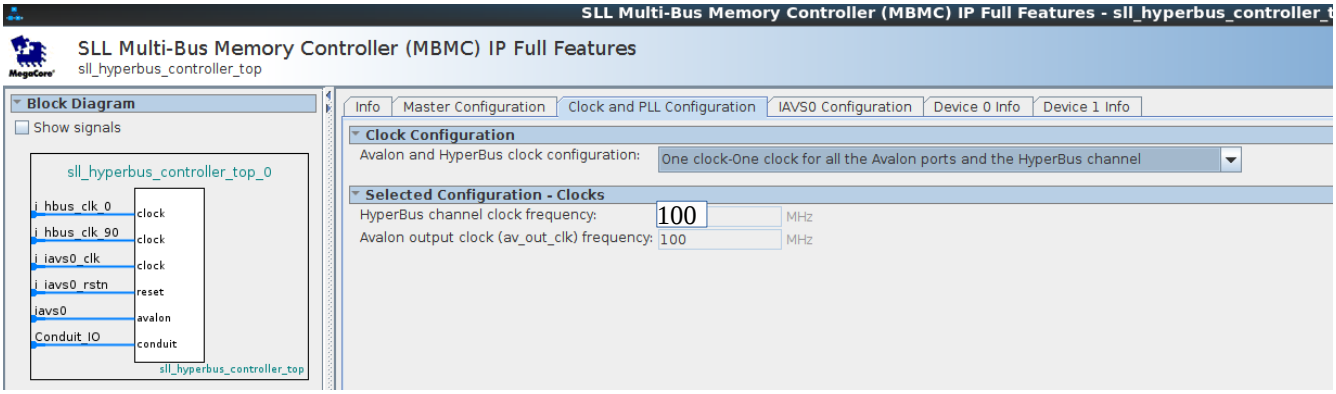
This configuration shows how to connect SLL MBMC IP so that the PSRAM/Flash memory channel operates at the same frequency as the Avalon-MM bus interface. The advantage of this configuration is lower circuit area.

5.1.1 Clocking (PLL) Wizard Configuration

The figure below shows a typical example of configuring the Clocking wizard . In this case, the clocks for the PSRAM/Flash channel and Avalon-MM interface channels are all set to 100 Mhz.



5.1.2 SLL' MBMC Configuration

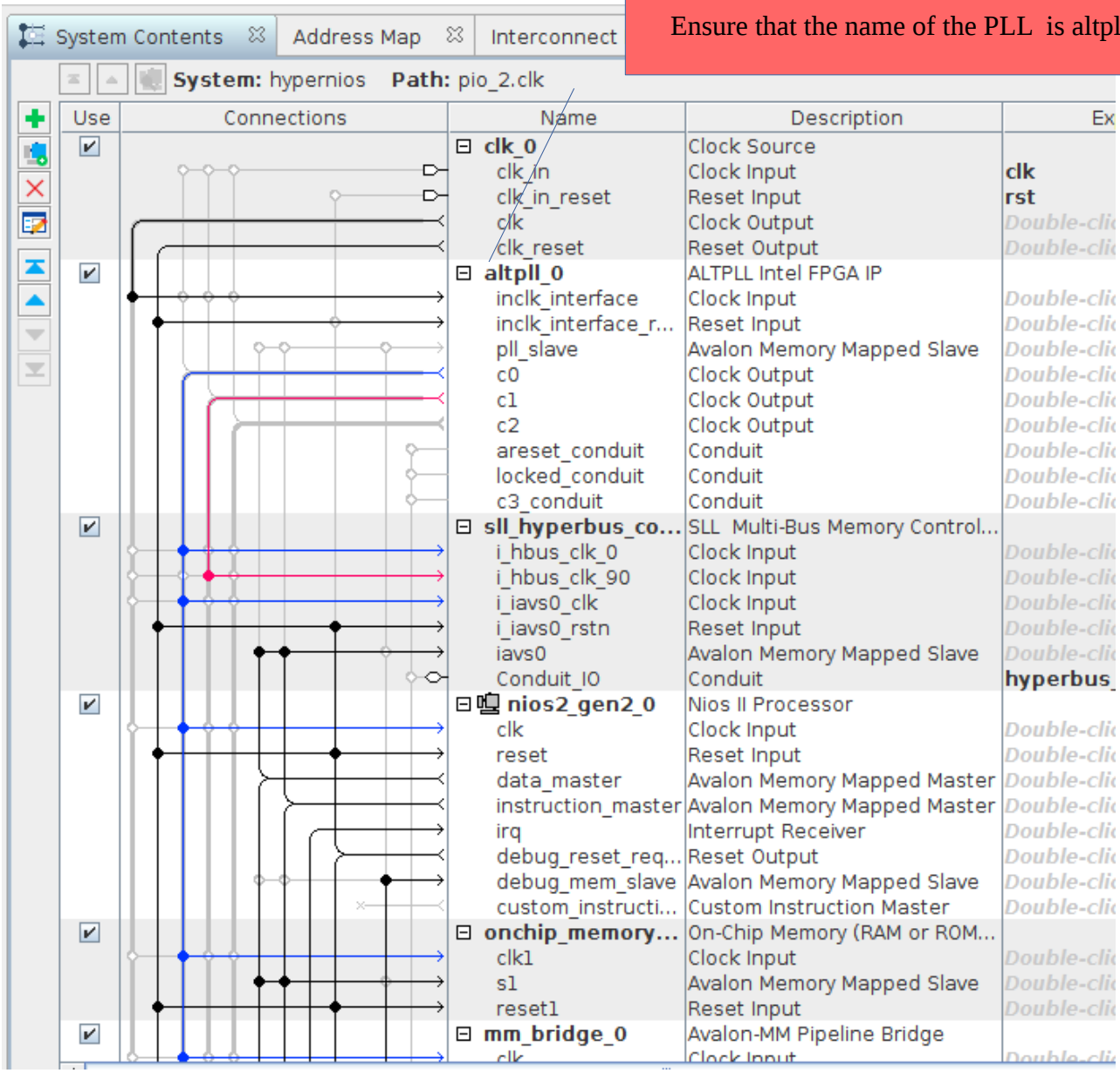


In this example, SLL MBMC IP is configured with :

- **Avalon and Hyperbus Clock Dependency** : **One clock**

5.1.3 SLL MBMC wiring

Ensure that the name of the PLL is altpll_0



Altera PLL output clock 0 (c0)

- Connect to i_hbus_clk_0 and i_iavs0_clk on SLL MBMC IP
- Connect to other Avalon-MM slaves and masters clock sinks

Altera PLL output clock 1 (c1)

- Connect to i_hbus_clk_90 on SLL MBMC IP

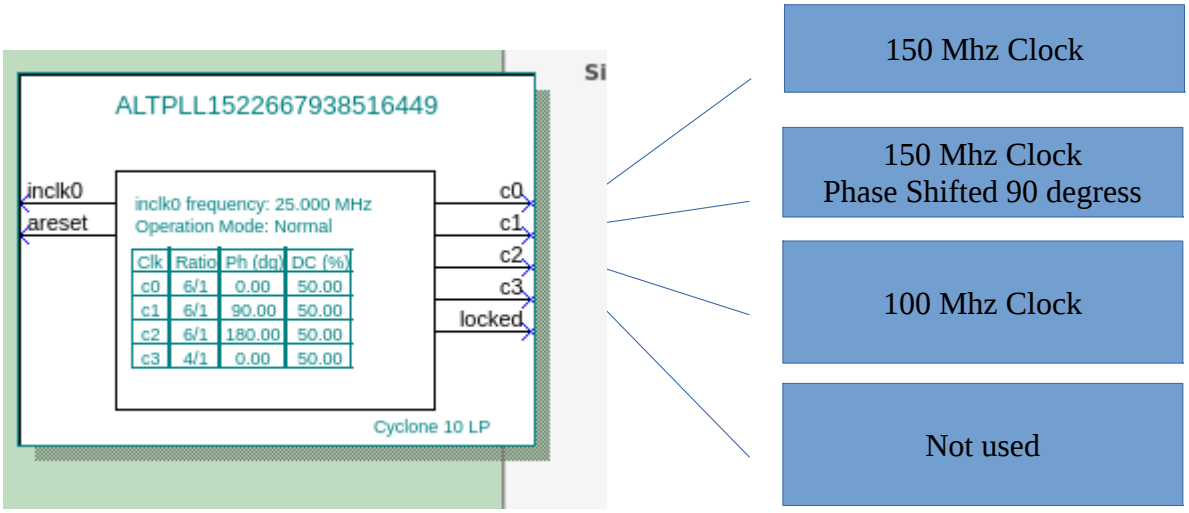
Note how i_hbus_clk_0 and i_iavs0_clk are connected to the same clock.

5.2 Option B – Different Clocks for the memory channel and AXI channel .

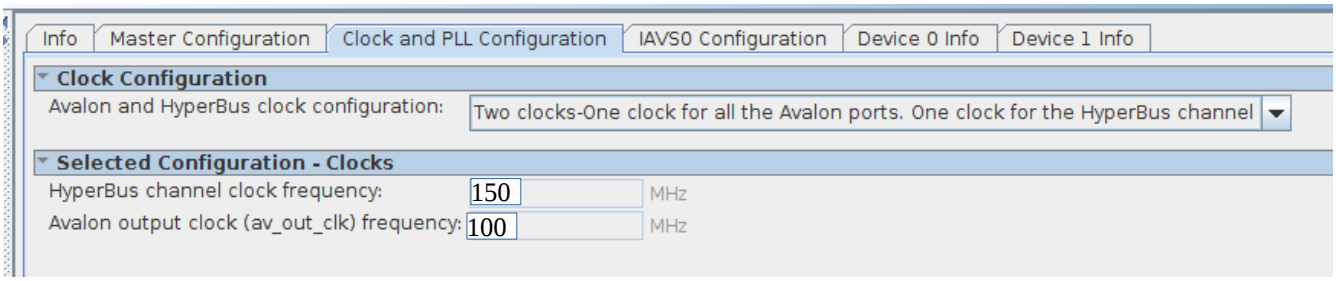
This configuration shows how to connect SLL MBMC IP so the PSRAM/Flash memory channel operates at a different clock frequency then the Avalon-MM bus interface.

5.2.1 Clocking Wizard Configuration

The figure below shows a typical example of configuring the Clocking wizard . In this case, the clocks for the PSRAM/Flash channel are all set to 150 Mhz, while the clock for the Avalon-MM interface channel is set to 100 Mhz.



5.2.2 SLL MBMC Configuration

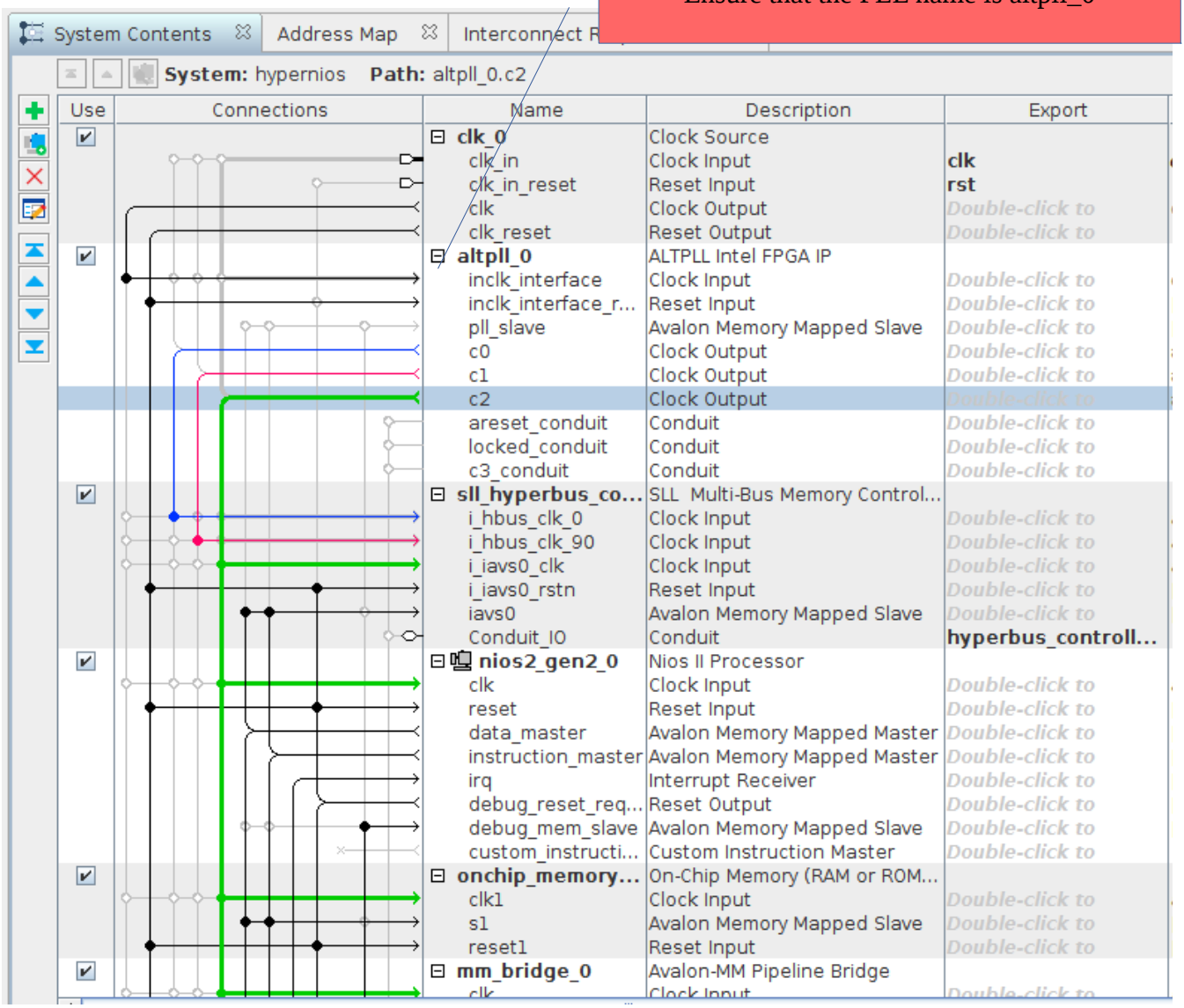


In this example, SLL MBMC IP is configured with :

- **Avalon and Hyperbus Clock Dependency** : **Two clocks**

5.2.3 SLL MBMC wiring

Ensure that the PLL name is altpll_0



Altera PLL output clock 0 (c0)

- Connect to i_hbus_clk_0 on SLL MBMC IP

Altera PLL output clock 1 (c1)

- Connect to i_hbus_clk_90 on SLL MBMC IP

Altera PLL output clock 2 (c2)

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

Note how i_hbus_clk_0 and i_iavs0_clk are connected to a different clock.

Important :

Please note that SLL MBMC contains a script that sets timing constraints for the PSRAM/Flash IO signals. For the external PLL configuration, this script assumes that in Qsys, Altera PLL's instance name is altpll_0.

6.0 Signal Description

6.1 : PSRAM/Flash Signal Interface

PSRAM - Flash Signal	Description	Comments	C10LP board consideration (HyperRAmM only).
HB_CLK0	Differential clock pair 0	To be connected to pin CK on PSRAM/Flash Memory device 0.	Connect to HyperRam CK signal
HB_CLK0n	Differential clock pair 0 *(used with 1.8V Memories)*	To be connected to pin CK# on PSRAM/Flash Memory device 0.	Connect to HyperRam CK# signal
HB_CLK1	Differential clock pair 1	To be connected to pin CK on PSRAM/Flash Memory device 1.	Not connected
HB_CLK1n	Differential clock pair 1 *(used with 1.8V Memories)*	To be connected to pin CK# on PSRAM/Flash Memory device 1.	Not connected
HB_CS0n	Chip select device 0	To be connected to pin CS# on PSRAM/Flash Memory device 0.	Connect to HyperFlash CS# signal
HB_CS1n	Chip select device 1	To be connected to pin CS# on PSRAM/Flash Memory device 1.	Connect to HyperRam CS# signal
HB_Wpn	Disables writes to Flash memory *(Write protect)*	To be connected to pin WP# on any HyperFlash device.	Connect to HyperFlash WP# if needed
HB_RWDS	Read strobe/Write mask signal	To be connected to pin RWDS on all PSRAM/Flash memory devices.	Connect to HyperRam/HyperFLASH RWDS signal
HB_Dq	Data bus (8-bit)		Connect to HyperRam/HyperFLASH Dq signals
HB_INTn	PSRAM/Flash interrupt from HyperFlash to FPGA.	To be connected to pin INT# on all HyperFlash device.	Connect to HyperFlash INT# signal
HB_RST0	PSRAM/Flash reset from HyperFlash to FPGA	To be connected to pin RSTO on all HyperFlash devices.	Connect to HyperFlash RSTO signal
HB_RSTn	Reset to PSRAM/Flash Memory	To be connected to pin RESET# on all PSRAM/Flash memory devices.	To be connected to HyperRam/HyperFLASH RESET# pin.

Optional Pullup Resistor *(not used for the HyperRam Only Configuraton)*

HB_INTn and **HB_RSTO** signals of the PSRAM/Flash memories are open drain output without a pull-up resistor. Therefore, when these signals are used in the design, it is suggested that a pull-up resistor is inserted either by setting the appropriate constraint on the input FPGA pad or adding a resistor on the board itself.

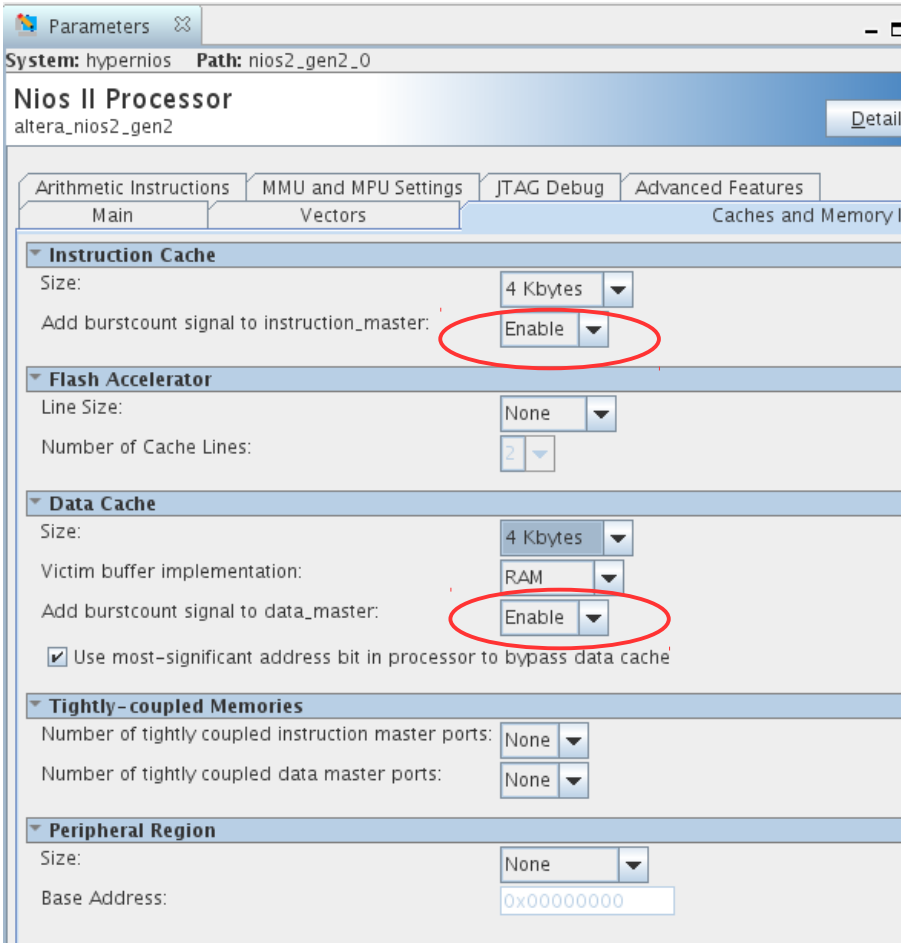
6.2 : Clock Signal Interface

Pll Signal	Description	Comments
i_hbus_clk_0	Clock driving the PSRAM/Flash core and memory I/O Pads	
i_hbus_clk_90	Clock used for driving the PSRAM/Flash Clock	
i_iavs0_clk	clock driving the Avalon-MM interface	

6.3 : Avalon-MM Interfaces

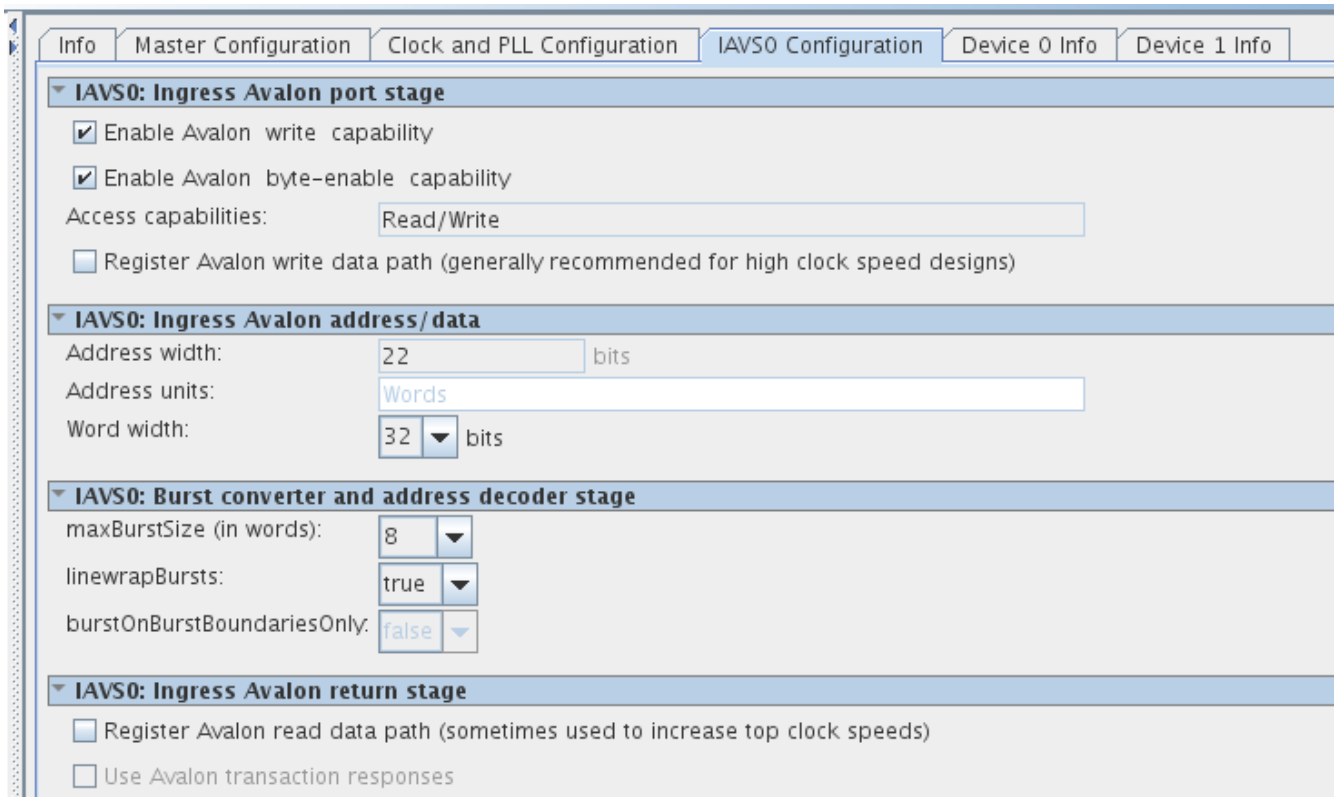
Avalon-MM signal	Description	Comments
i_iavs0_clk	Input clock related to Avalon-MM interface	
i_iavs0_rstn	Reset related to Avalon-MM interface	Resets the core and IO Pads
i_iavs0_addr	Avalon-MM address	32-bit aligned
i_iavs0_burstcount	Avalon-MM burst count	Supports increment and wrapped modes
i_iavs0_wait_request	Avalon-MM wait request	
i_iavs0_do_wr	Avalon-MM write request	
i_iavs0_byteenable	Avalon-MM byte-enable	Byte mask
i_iavs0_wdata	Avalon-MM write data	32-bit read data signal
i_iavs0_do_rd	Avalon-MM read request	
i_iavs0_rdata_valid	Avalon-MM read data valid	
i_iavs0_rdata	Avalon-MM read data	32-bit read data signal

7.0 Connecting SLL PSRAM/Flash Memory Controller to the NIOS II/f embedded processor



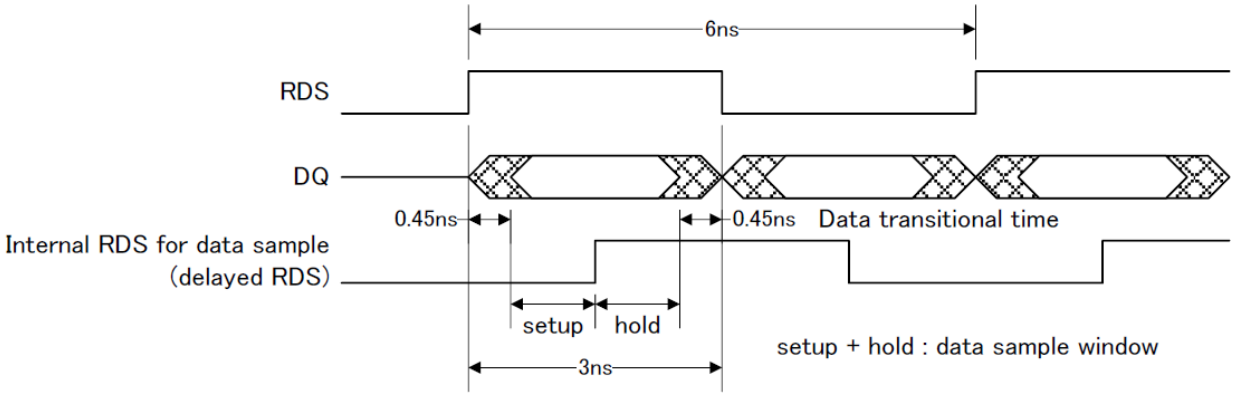
For best performance, configure the Nios II/f processor with Instruction/Data caches enabled and **burstcount** signals enabled.

Also, configure the SLL PSRAM/Flash memory controller Avalon slave (IAVS0 configuration tab) with a **burstsize** of 8 and **linewrap burst** enabled.



8.0 Static Timing Consideration

8.1 Data Input Timing Constraint



The input read strobe (HB_RWDS) signal is edge aligned to the data signal (HB_dq).

8.1 Timing Constraints

A timing constraint script (sll_ca_hbc_t001_top.sdc) will be automatically generated by Synaptic Labs' Multi-Bus memory Controller Qsys component. This script will be placed in Quartus Prime Synthesis directory and executed automatically during the Quartus Prime compilation process.

8.2 Pin Clustering

Please ensure all PSRAM/Flash channel pins are clustered physically close together in the programmable FPGA fabric. The FPGA board designer will need to balance the place-and-route requirements of the Multi-Bus memory Controller Logic against the ideal placement of pins from the board layout perspective to minimize skew across pins and to minimise pin-to-pin wire latency delay.

The signals received on the HB_DQ pins are fed as an input into a single on-chip SRAM (e.g. M9K). The parallel capture of those signals is clocked by HB_RWDS. Hence the location of the HB_DQ/HB_RWDS pins must be placed in a way to also ensure low wire latencies to that single on-chip SRAM. From the perspective of the PSRAM/Flash memory controller, try to ensure that the data being transported over all DQ signals arrive as close as possible, with as little skew, at the I/O pads”

The following output signals { HB_CLK0, HB_CLK0n, HB_CLK1, HB_CLK1n, HB_CS0n, HB_CS1n}

- use an **altera_gpio_lite/altera_ddio** output pad configured in DDR register mode.

The following bi-direction signals HB_Dq and HB_RWDS

- (Output mode) - use an **altera_gpio_lite/altera_ddio** output pad with output enable control (oe) and configured in DDR register mode.
- (Input mode) - Unregistered buffer mode (pass through)

Please ensure these DDR signals are mapped to IO Elements with DDR capabilities.

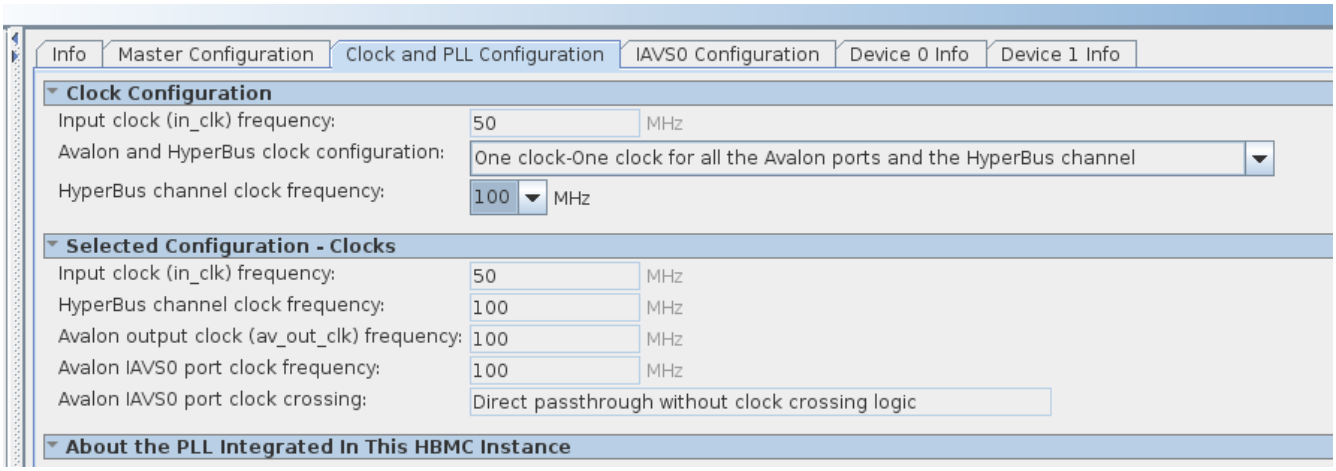
9.0 Using 3V HyperRAM and HyperFlash devices

9.1 Selecting the correct operating frequency in Qsys

The 3V PSRAM/HyperRAM and HyperFlash memories supports a lower frequency than the 1.8V memory devices. Hence the designer needs to select a frequency that is supported by the memory device.

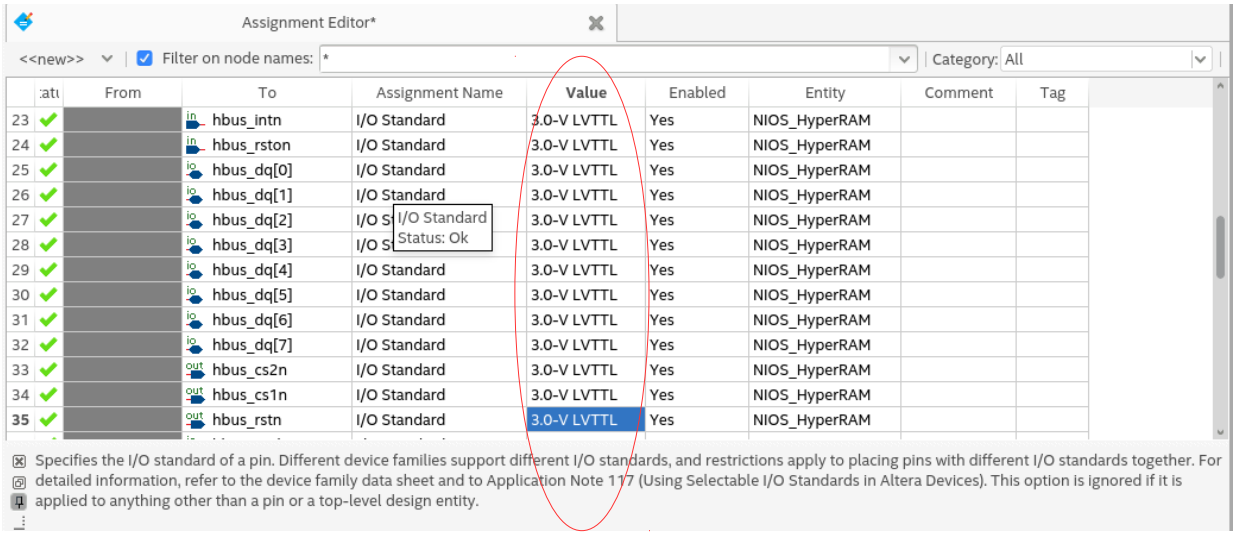
In Qsys, open SLL MBMC IP component. Locate the Clock and PLL configuration Tab. Ensure that **The PSRAM/Flash channel clock Frequency** is 100Mhz or less.

We suggest that the designer use the One Clock operation for both the Avalon Port and the PSRAM/Flash Channel Port.



9.2 Selecting the correct voltage in Quartus

The designer needs to set the PSRAM/HyperFlash voltage level signals to 3V. In Quartus, open the Assignment editor (Quartus→ Assignment → Assignment Editor). Set the PSRAM/Flash memory signals voltage level to 3V.



9.2 Connecting the PSRAM signals to the FPGA I/O

The 3V PSRAM and HyperFlash devices do NOT require a differential clock pair. HB_CLK0n is left unconnected.

The figure below shows a typical top level connection for a 3V HyperRAM only design. Note that some signals are left unconnected since they are not used in the design.

```
-------------------------------------
-- Instantiation of main QSys system
-------------------------------------
u0_main : component lab2
   port map (
      clk_clk       => c10_clk50m,
      reset_reset_n => c10_resetn,

      sll_hyperbus_HB_RSTn  => hbus_rstn,   --  .HB_RSTn
      sll_hyperbus_HB_CLK0  => hbus_clk0p,  --  .HB_CLK0
      sll_hyperbus_HB_CLK0n => open         ),  --  .HB_CLK0n
      sll_hyperbus_HB_CLK1  => open,        --  .HB_CLK1
      sll_hyperbus_HB_CLK1n => open,        --  .HB_CLK1n
      sll_hyperbus_HB_CS0n  => open,        --  .HB_CS0n
      sll_hyperbus_HB_CS1n  => hbus_cs2n,   --  .HB_CS1n
      sll_hyperbus_HB_WPn   => open,        --  .HB_WPn
      sll_hyperbus_HB_RWDS  => hbus_rwds,   --  .HB_RWDS
      sll_hyperbus_HB_dq    => hbus_dq,     --  .HB_dq
      sll_hyperbus_HB_RSTOn => '1',         --  .HB_RSTOn
      sll_hyperbus_HB_INTn  => '1'          --  .HB_INTn

      pio_led_export => LEDs_debug,
   );
```

Additional QSF constraint

Sometimes the user needs to manually add delay constraints in the top level QSF file for the Read Strobe signal (RWDS).

```
set_instance_assignment -name PAD_TO_CORE_DELAY 2 -to hbus_rwds
```

Kindly ask for a reference design in case static timing is not achieved.