Synaptic Labs'

Multi Bus Memory Controller *(MBMC-T001)* User Manual

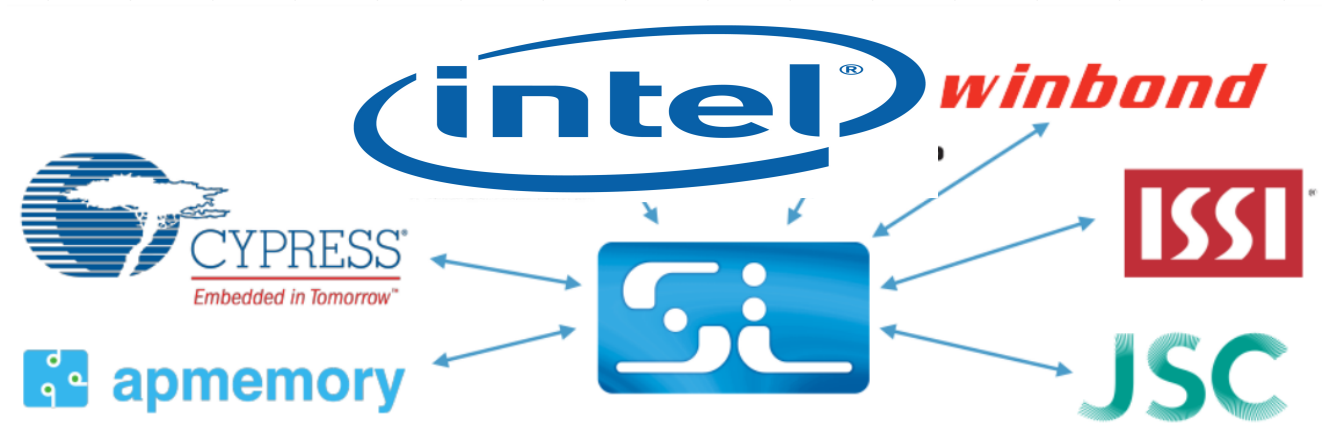
Table of Contents

1.0 Introduction

This user manual will guide you through the basic steps of employing SLL Multiple Bus Memory Controller (MBMC) IP unit in the Qsys environment.

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 (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.*

2.0 Finding SLL Multiple Bus Memory Controller IP in the Qsys IP Catalog

Within Qsys, SLL Multi-Bus Memory Controller IP can be found in the Qsys IP Catalog panel under:

+ Synaptic Labs
 + Memory
  + Hyperbus
   + SLL Multi Bus Memory Controller

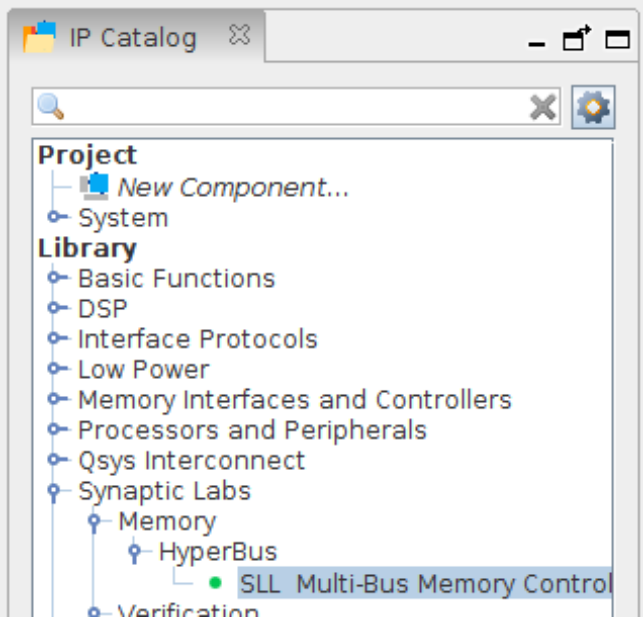
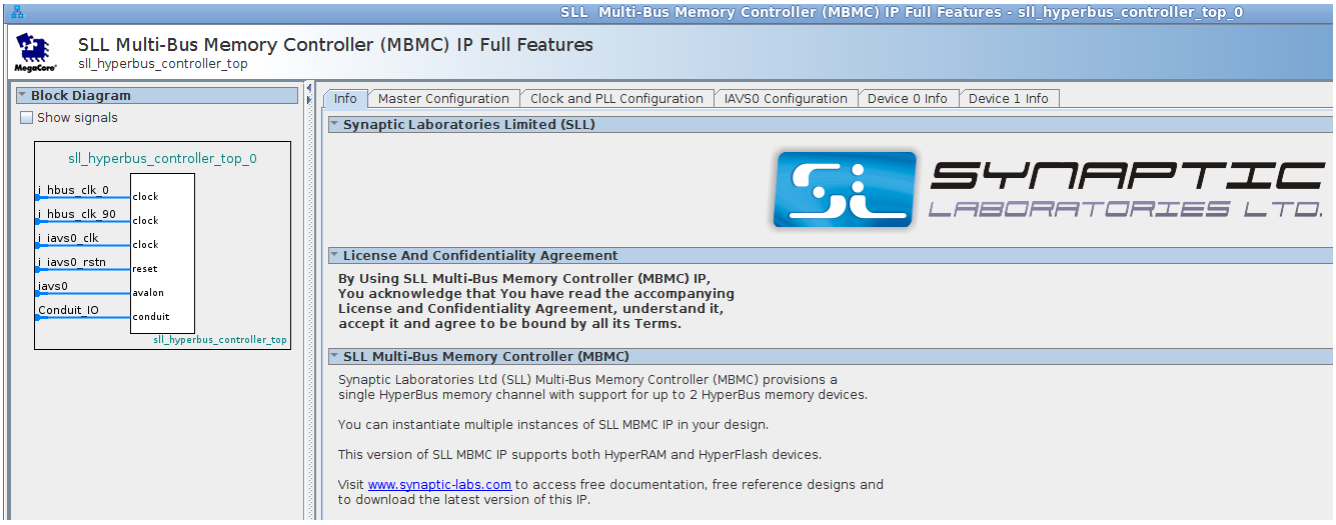


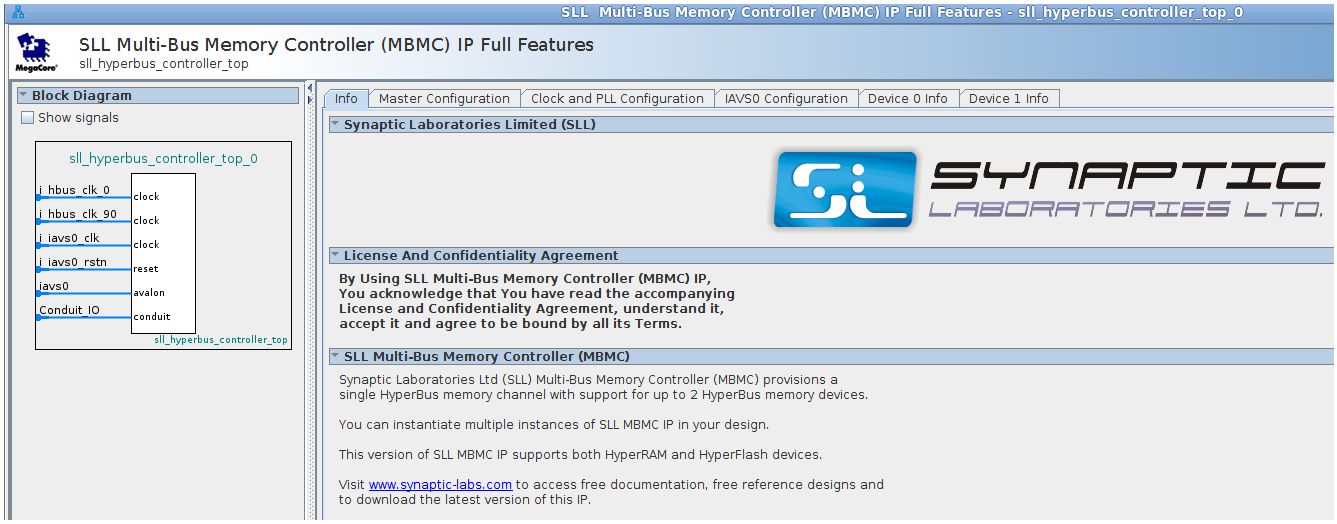
Figure 1: Finding SLL Multi Bus Memory Controller IP in the Qsys IP Catalog



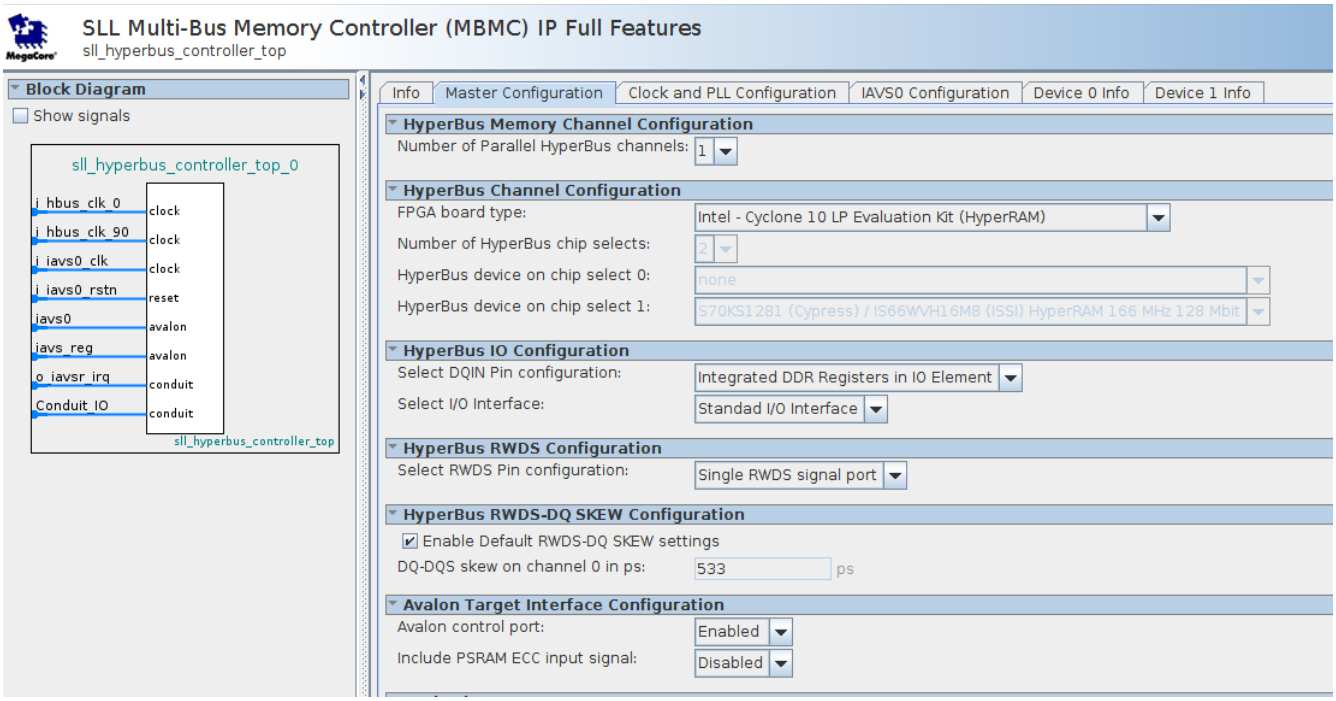
2.1 Instantiating SLL Multiple Bus Memory Controller IP

Double click on Synaptic-Labs Multi Bus Memory Controller IP to instantiate an instance of this IP in your Qsys project.

The following IP configuration options will appear on your screen:

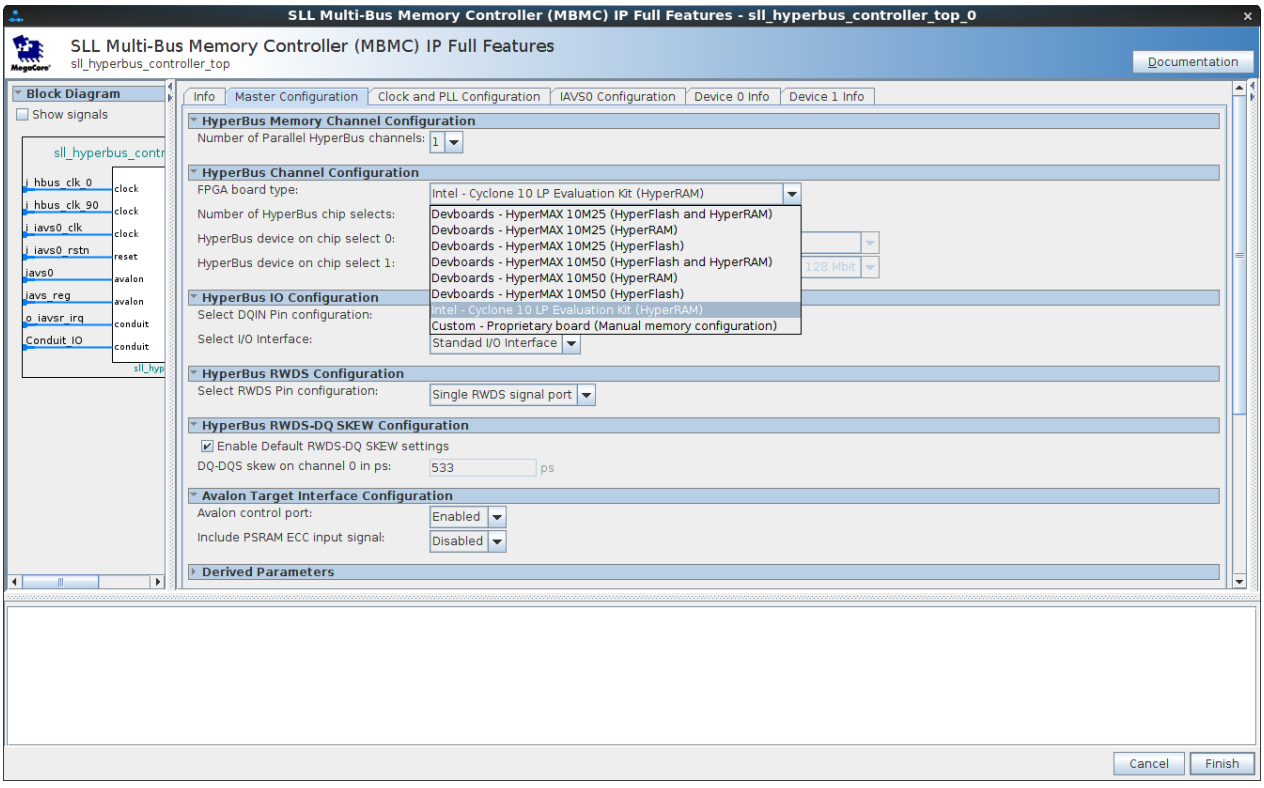


2.2 Configuring SLL MBMC IP in the Master User Interface Tab

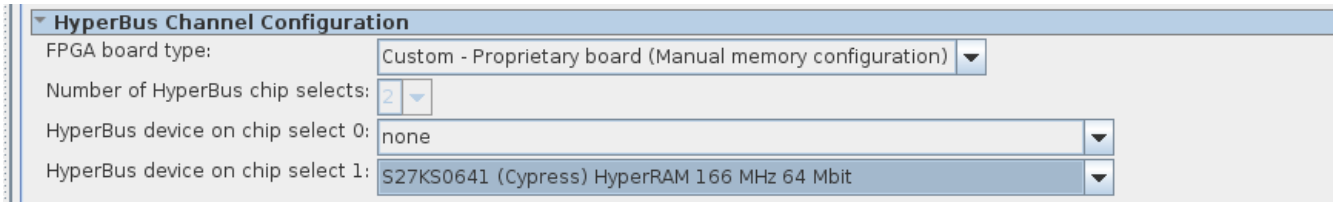


2.2.1 PSRAM/Flash Channel Configuration Section

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



For Custom boards, select Custom-Proprietary Board (Manual Memory Configuration)



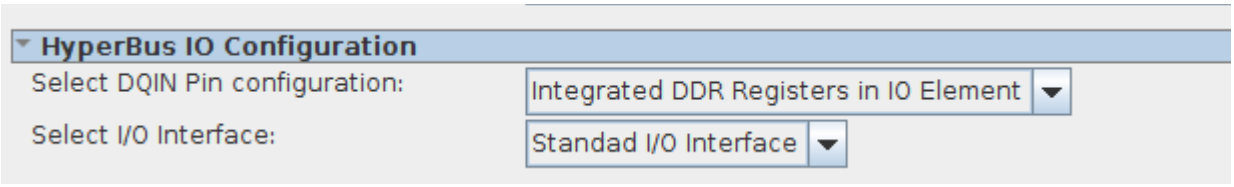
The image above shows an example for a custom board with a 64 Mbit HyperRAM connected to SLL MBMC chip select 1. The listed devices can be used for both the 100 Mhz (3.3V) and 166 Mhz (1.8V) memory devices.

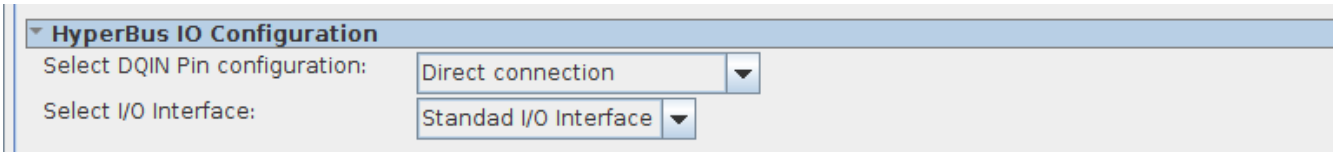
2.2.2 Hyperbus IO Configuration

The user can select can either select

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

The Integrated DDR Register in the IO Element option allows for higher clock speeds to be achieved. This feature is supported by all FPGA families.

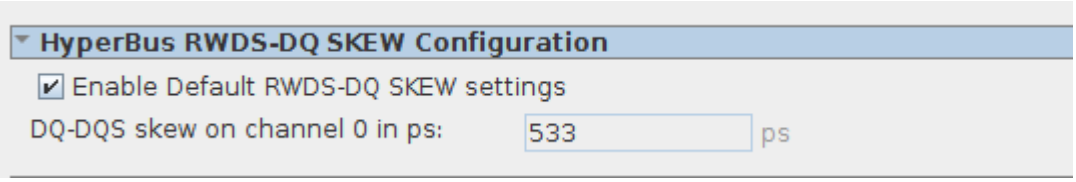




2.2.3 PSRAM/Flash RWDS Configuration

Currently only the **Single RWDS signal port** configuration is supported.

2.2.4 RWDS-DQ Skew Configuration



When the **[x] Enable Default RWDS-SKEW** is checked, the RWDS-DQ skew will be automatically calculated.

When the **[ ] Enable Default RWDS-SKEW** option is left unchecked, the user can enter the RWDS-DQ skew manually.

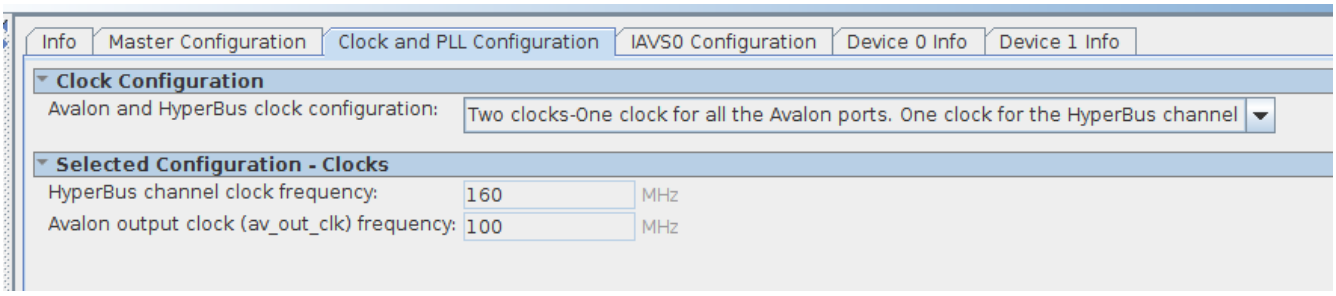
The value will be used to set the correct timing parameters in the timing constraint script automatically generated by SLL MBMC IP.

2.2.5 Avalon Target interface Configuration

When enabled, a secondary Avalon-MM port is created.
This can be used for programming HyperFlash or accessing pseudoRAM registers.

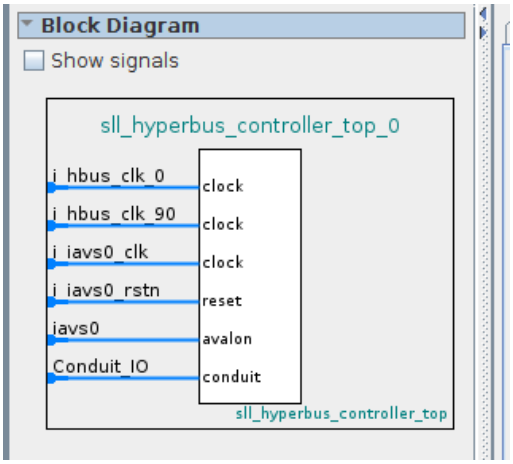
2.3 Configuring SLL MBMC IP in the Clock and PLL Configuration Tab

SLL MBMC IP is configured in external PLL mode. Internal PLL mode is no longer supported. The option is no longer visible in the configuration settings.



2.3.1 External PLL mode

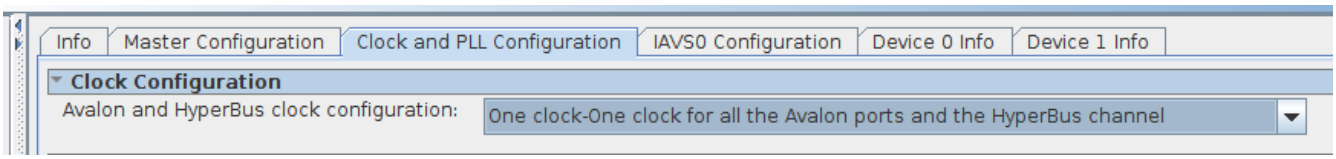
The different Avalon-MM master/slave clocks and PSRAM/Flash Clocks in the design are generated by an external PLL.



In the external PLL mode, S/Labs' includes 3 clock signals

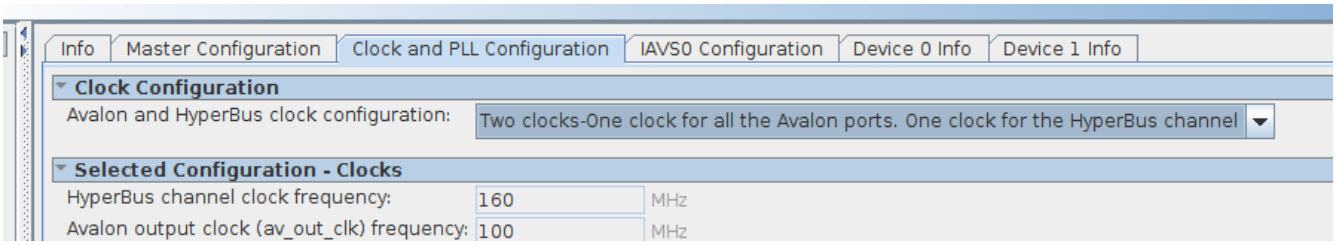
i_hbus_clk_0 : input clock driving the Hyperbus controller
i_hbus_clk_90 : input clock driving the Hyperbus controller
Operates at same frequency as i_hbus_clk_0
Phase shifted 90 degrees
i_iavs_clk : input clock driving the Avalon-MM interface

2.3.1.1 For Single Clock Operation



Select this option when the PSRAM/Flash memory channel is operating at the same frequency as the Avalon Memory slave interface.

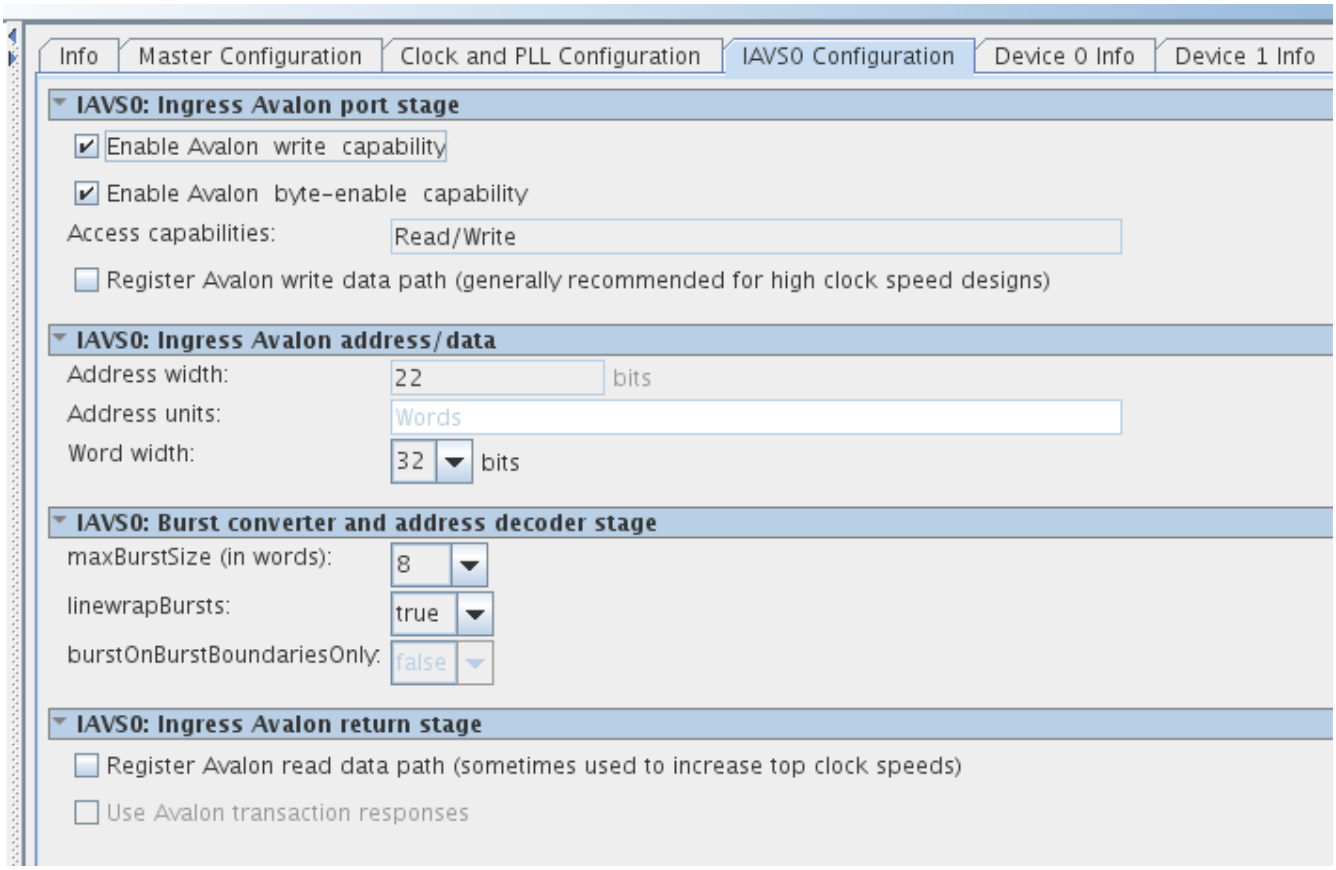
2.3.1.2 For Dual Clock Operation



Select this option when the PSRAM/Flash memory channel is operating at a different frequency than the Avalon Memory slave interface.

2.4 Configuring SLL MBMC IP in the IAVS0 Configuration Tab

The Avalon port slave has its own port configuration tab. This offers great flexibility and enables the interface to be configured under different parameters.



2.4.1 Configuring Ingress Avalon port slave parameters

The "**Enable Avalon write capability**" feature is enabled by default and adds the Avalon write signal conduits on this slave interface. If the Avalon slave interface is a read only connection, such as the Nios II instruction master, the write signals can be disabled by un-ticking this box.

The " **Enable Avalon byte-enable capability**" feature is enabled by default and adds the Avalon byeenable signal conduit on this slave interface. For bus masters that do NOT employ the byte-enable signals, this feature can be turned off.

The "**Register Avalon Write data**" feature allows the registration of the write data signal on the Avalon slave to potentially increase clock speeds and ease place-and-route.

2.4.2 Configuring the burst count on the Ingress Avalon port slave

The " **maxBurstSize (in words)**" GUI label select the maximum burst length of read and/or write burst memory transfer requests on this port in words. The value of **maxBurstSize** must be set to the value of the largest **maxBurstSize** of all Avalon bus masters interfaces that are connected to this slave interface.

When a burst size greater than 1 is selected, additional parameters become available to configure the burst behaviour.

If one or more of the Altera Avalon bus master interfaces connected to this slave port employs **linewrapBursts=true**, then set the **linewrapBursts** of this slave port to true.

An additional parameter **burstOnBurstBoundariesOnly** becomes available.

- Set the value of **burstOnBurstBoundariesOnly** to true for the slave port when all bus master interfaces connected to this port have their **burstOnBurstBoundariesOnly** parameter set to true.
- Otherwise, set the value of **burstOnBurstBoundariesOnly** to false for the slave port.

Example : if the Nios II processor is configured with

- Instruction cache enabled and the "Add burstcount signal to instruction master " enabled. (This will configure the Nios II instruction master with a burst size of 8 and linewrapBursts=true).
- Data cache enabled and the "Add burstcount signal to data master " enabled. (This will configure the Nios II data master with a burst size of 8, linewrapBursts=false and burstOnBurstBoundariesOnly=true).
- and we are connecting both the Instruction Master and the Data master to SLL Multiple Bus Memory Controller Avalon-MM slave

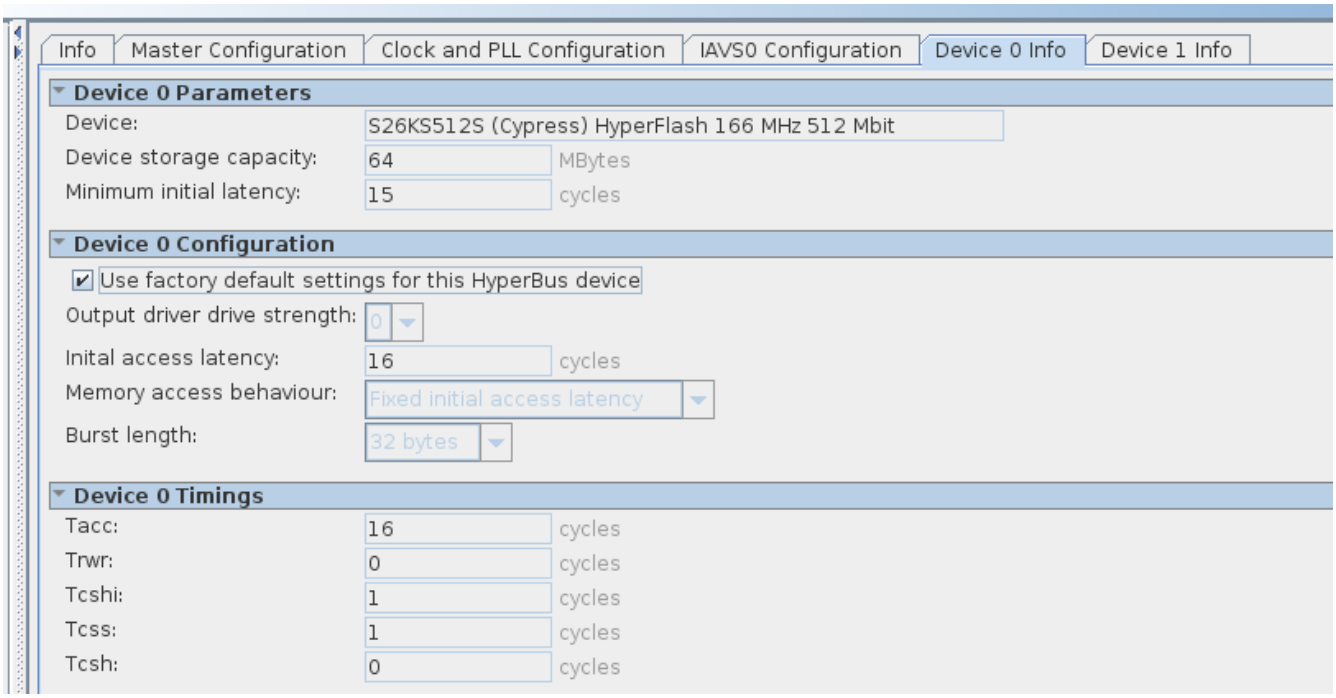
then we set **maxBurstSize (in words) to 8**, and **linewrapBursts=true.**

2.4.3 Configuring the Ingress Avalon port slave return path

The “**Register Avalon Read data**” feature allows the registration of the read data signal on the Avalon slave to potentially increase clock speeds and eases place-and-route.

The “**Use Avalon Transaction Response**” feature adds the Avalon transaction response conduit.

2.5 Configuring SLL MBMC IP in the Device 0/1 Tab



The **Device** Info field shows information about the Hyerbus memory.

The **Device Size** field shows the size (in megabytes) of the the PSRAM/Flash memory.

The "**Minimum Initial Latency for the current frequency"** field shows the optimal latency for the current memory device based on the Multi Bus Memory Controller frequency selected earlier.

If the **"Use default parameters"** button is

- enabled, the PSRAM/Flash memory will operate with its default settings. It might not necessary be running at optimal settings for the selected frequency.
- disabled, the PSRAM/Flash memory will operate with the setting configured in the **Device Configuration** Tab. On startup or after reset, SLL Multi Bus Memory Controller will automatically program the selected PSRAM/Flash Memory control register with the appropriate settings.

If the **"Use default parameters"** is disabled, the user can set the settings for the PSRAM/Flash Memory control registers.

The fields on the **Device Configuration** tab are slightly different for the HyperRAM and HyperFlash Memory as shown in the figures below.

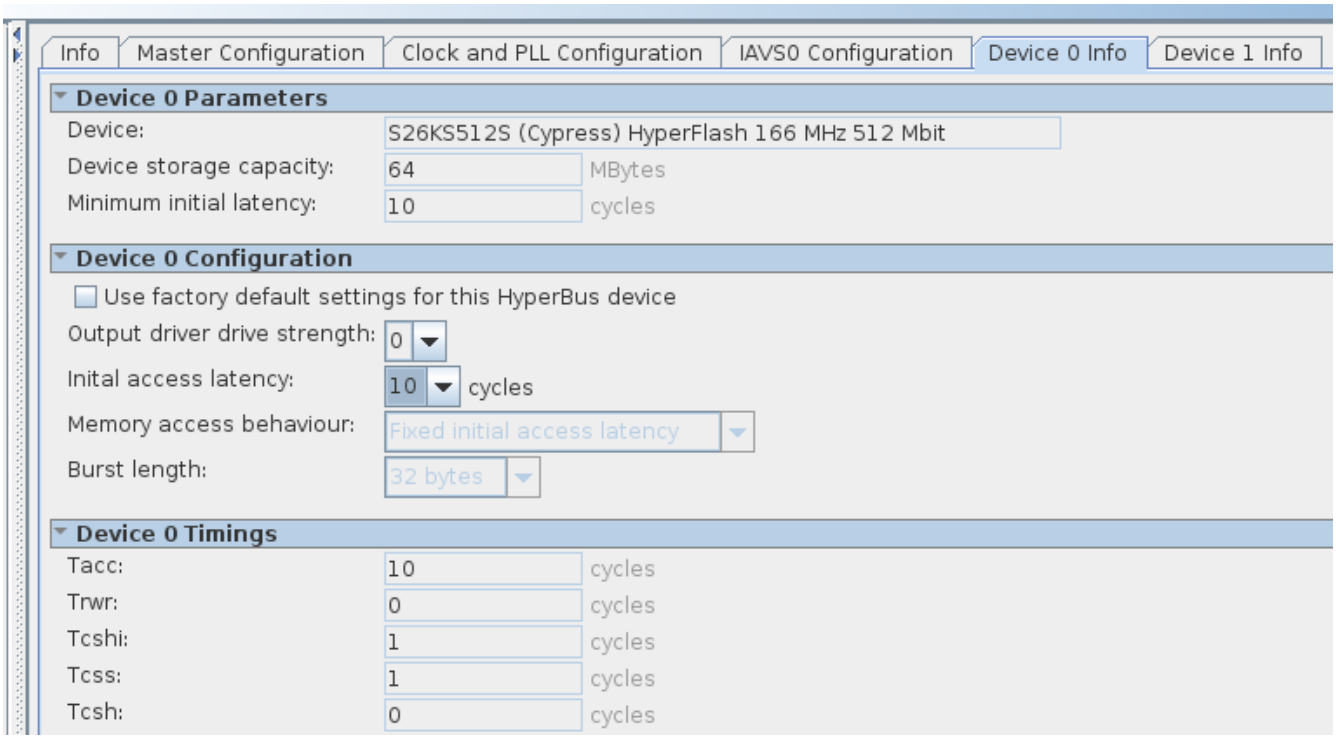
The **Drive Strength** feature provide a means to adjust the DQ[7:0] signal output impedance to customize the DQ signal impedance to the system to minimize high speed signal behaviors such as overshoot, undershoot, and ringing. The default POR or reset configuration value is 000b to select the mid point of the available output impedance options. It allows customization of the DQ signal line loading, length, and impedance. Check the PSRAM/Flash memory device datasheet.

The **Initial Latency** field selects the number of clocks for initial latency. The number of latency clocks needed to satisfy tACC depends on the PSRAM/Flash frequency and can vary from 3 to 16 clocks. The number of initial latency options implemented and the POR or reset default initial value is device dependent. The initial latency value must be equal to or greater than the **Minimum Initial Latency for the current frequency"** field described above.

When the **Fixed Latency** field is set, all read or write transactions have the same initial latency. (This option is not available for the HyperFlash Device).

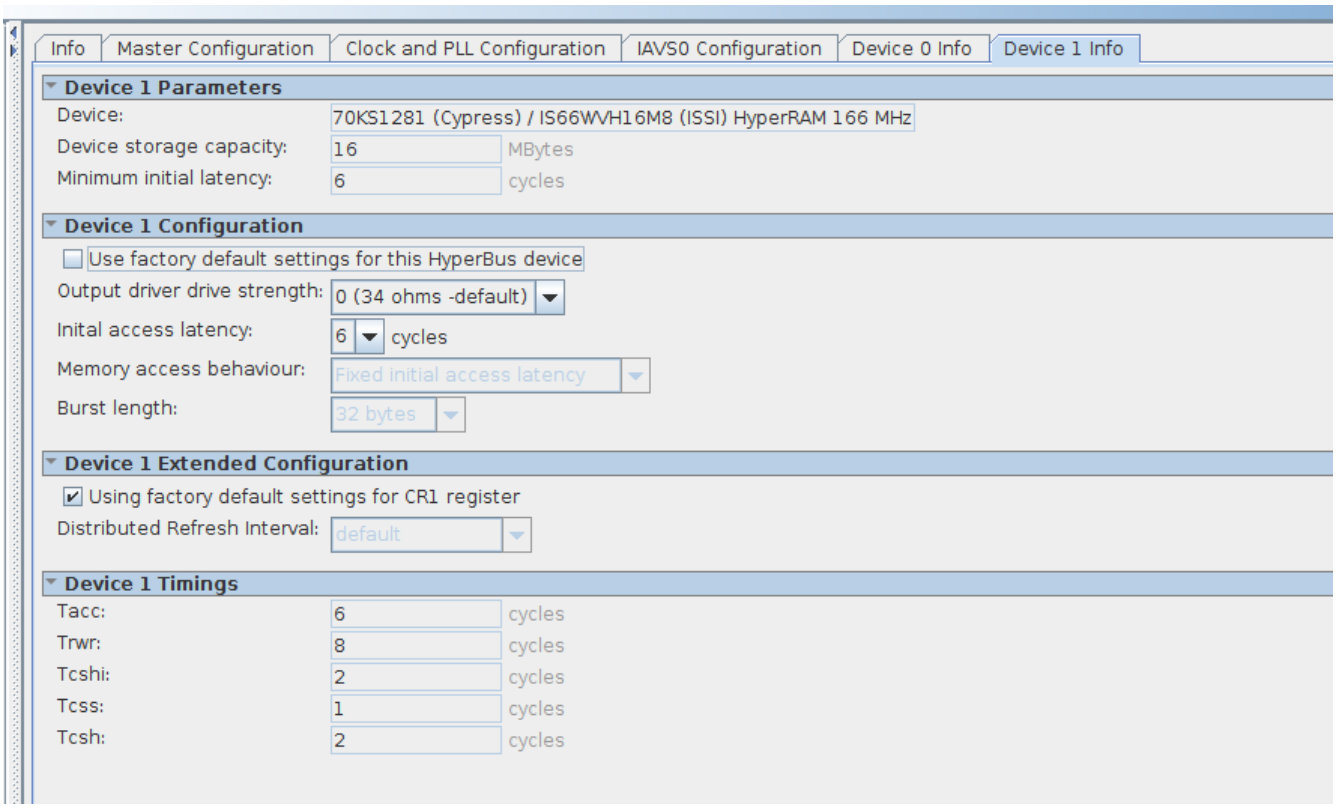
When the **Variable Latency** field is set, the RWDS input pin is used to detect whether 1 or 2 initial latency counts are required. All read or write transactions have different initial latencies based on the value of RWDS during the Command/Address phase. **(This option is currently not available)**

2.5.1 HyperFlash Configuration example



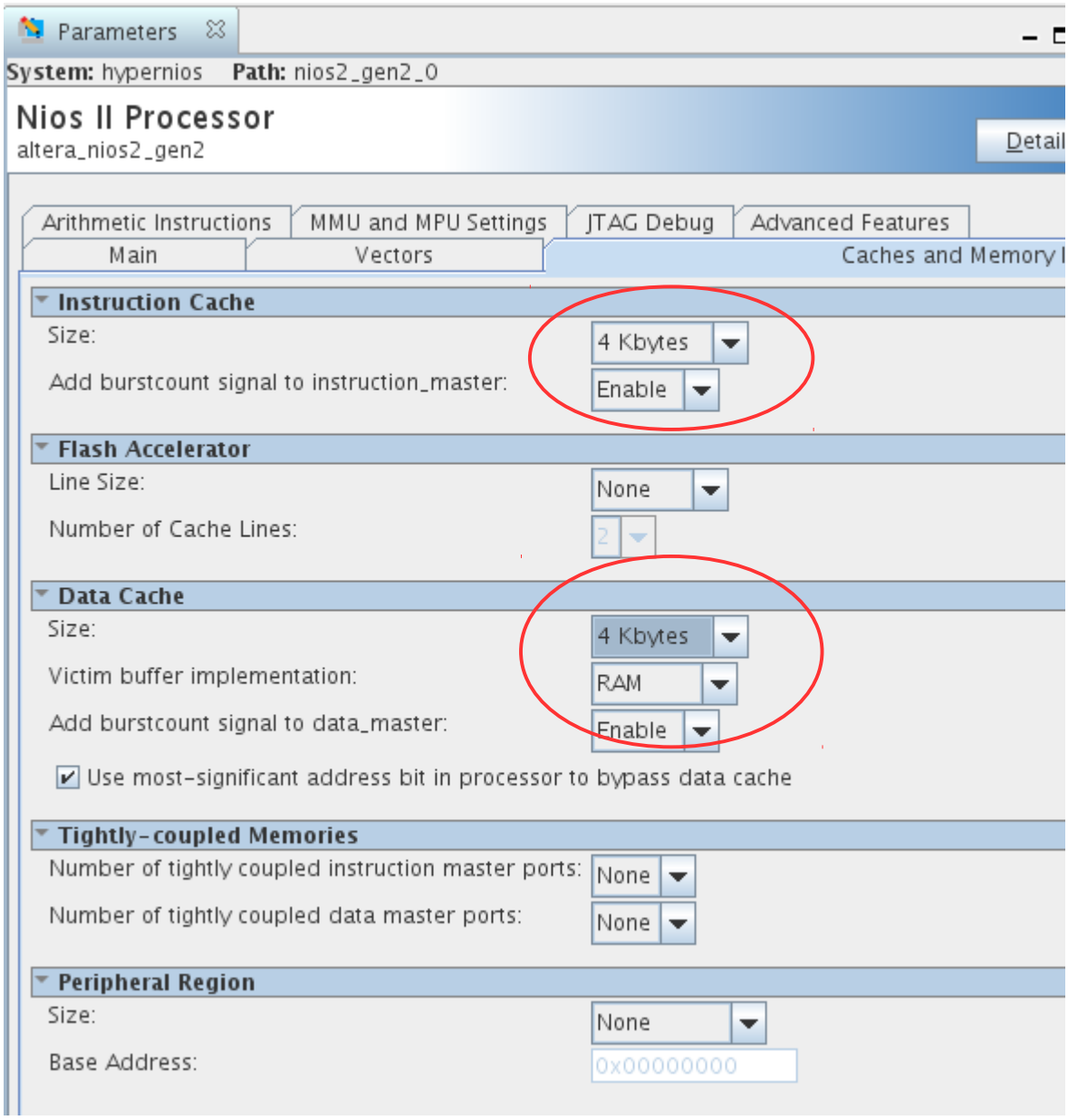
The example above show a typical configuration when the HyperFlash is programmed in non default factory settings. The PSRAM/Flash memory is running at 100 Mhz. On startup or after reset, SLL Multi Bus Memory Controller will automatically program the selected PSRAM/Flash Memory control register with the chosen settings.

2.5.2 HyperRAM Configuration example



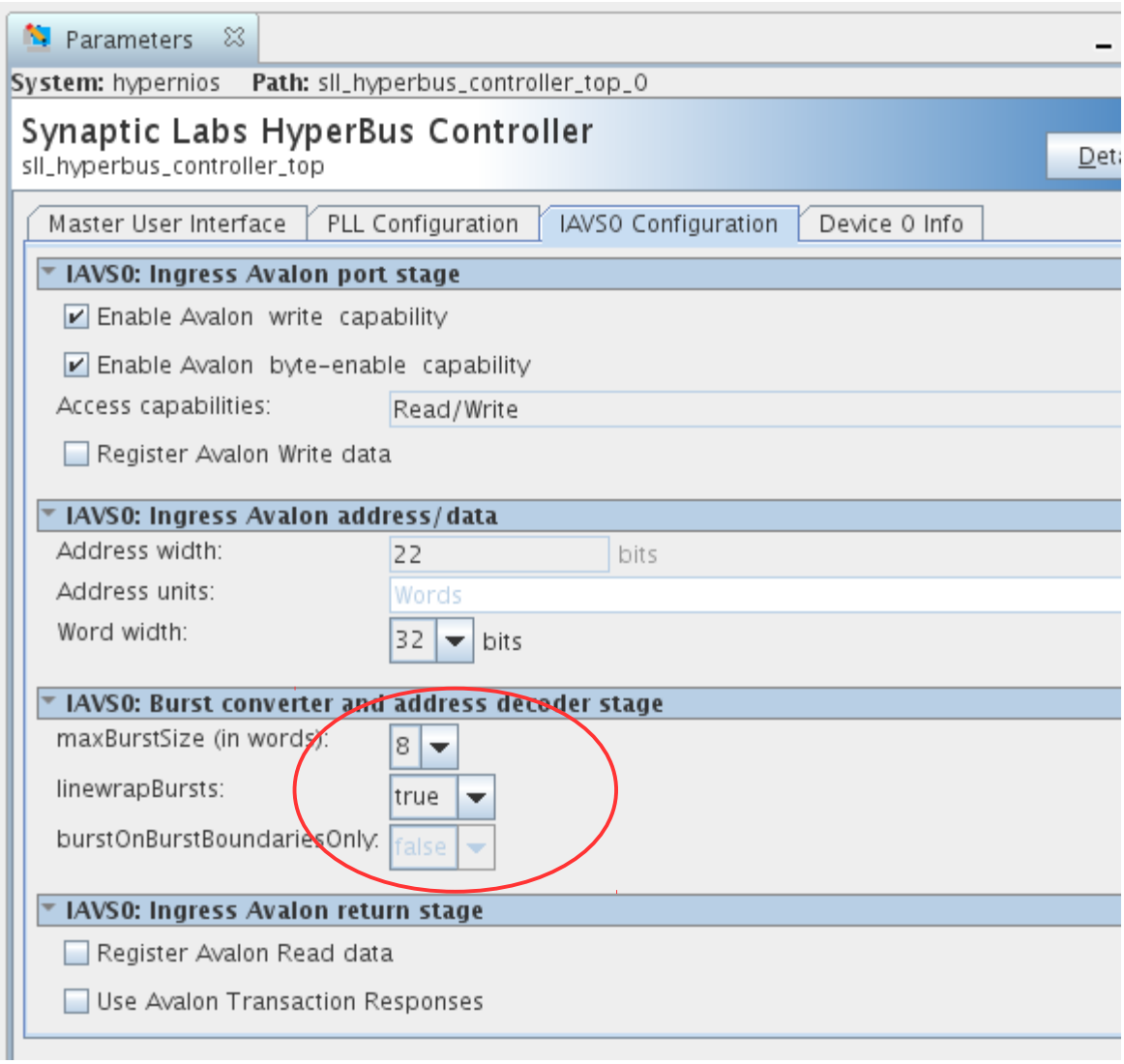
The example above show a typical configuration when the HyperRAM is programmed in non default factory settings. The PSRAM/Flash memory is running at 160 Mhz. On startup or after reset, SLL Multi Bus Memory Controller will automatically program the selected PSRAM/Flash Memory control register with the chosen settings.

3.0 Connecting SLL Multi Bus Memory Controller to the NIOS II/f embedded processor



Configure the Nios II/f processor with Instruction/Data caches enabled and **burstcount** signals enabled.

Configure the SLL Multi Bus Memory Controller Avalon slave (IAVS0 configuration tab) with a **burstsize** of 8 and **linewrap burst** enabled.



3.1 SLL MBMC with external PLL Configuration

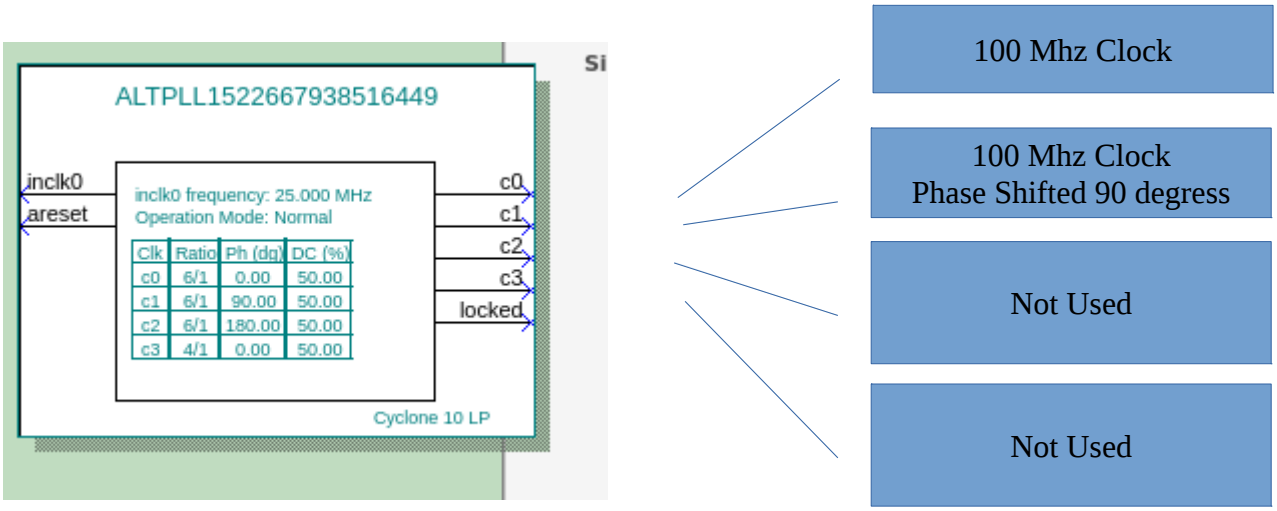
With the external PLL configuration, S/Labs MBMC IP requires 4 clocks :

- **i_hbus_clk_0** : clock driving the Multiple Bus Memory 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 S/Labs MBMC IP is configured to run at a single clock speed, this clock is connected to **i_hbus_clk_0** clock.

3.1.1 Option A – Same Clock for the PSRAM/Flash memory channel an Avalon-MM channel.

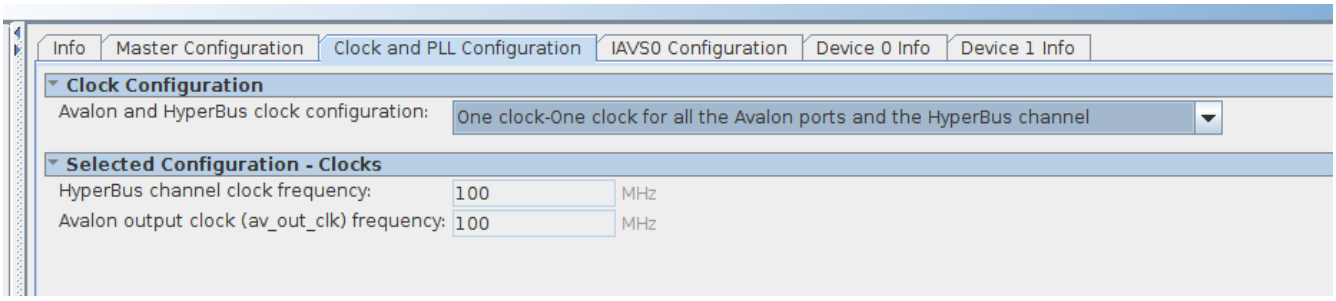
This configuration shows how to connect SLL MBMC IP so that the PSRAM/Flash memory channel operates at the same frequency as the Avalon-MM bus interface. The advantage of this configuration is lower circuit area.

3.1.2 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.

3.1.3 SLL MBMC Configuration

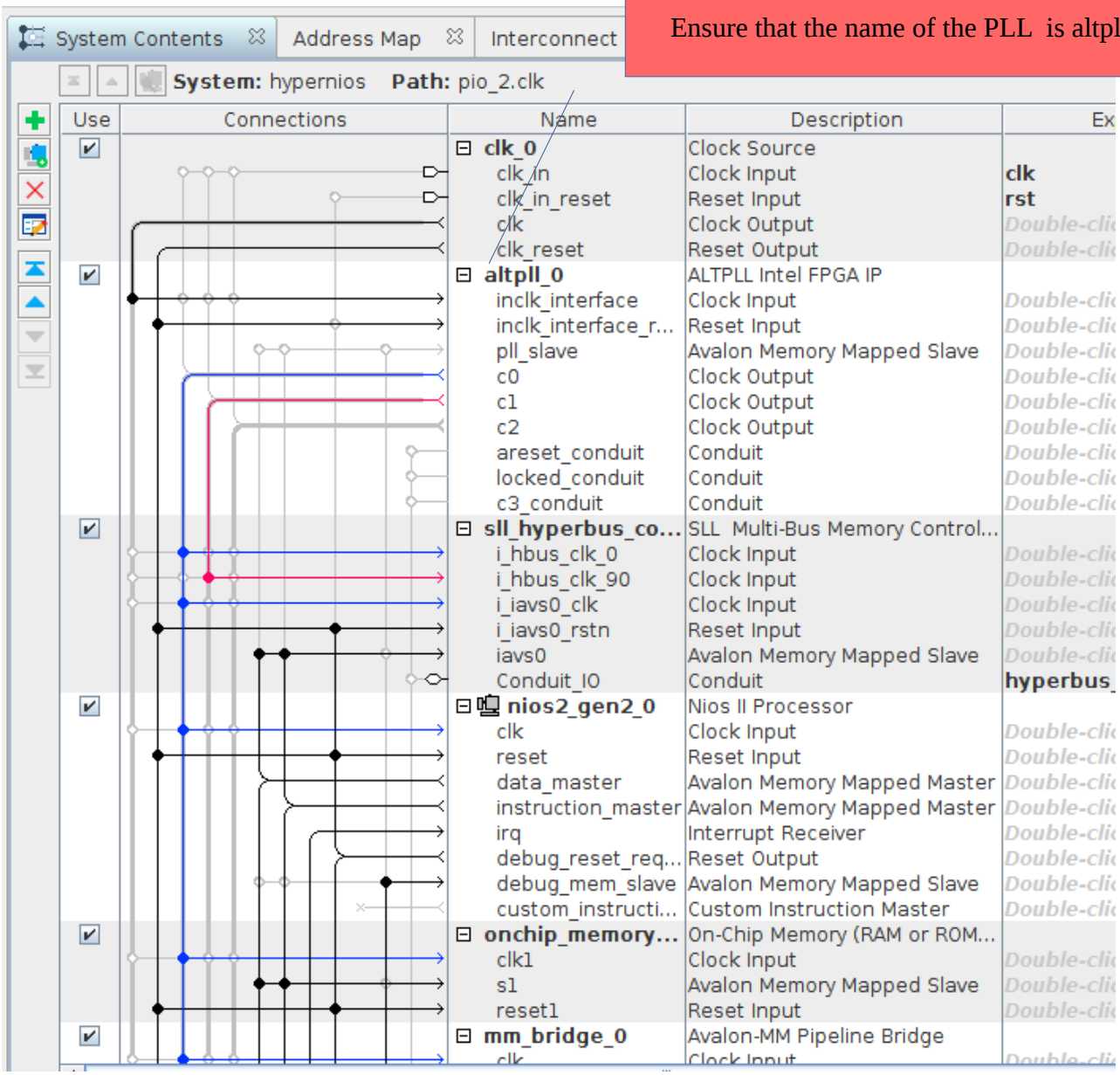


In this example, SLL MBMC IP is configured with :

- **Avalon and Hyperbus clock configuration** : **One clock**

3.1.4 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 S/LABS 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 S/LABS MBMC IP

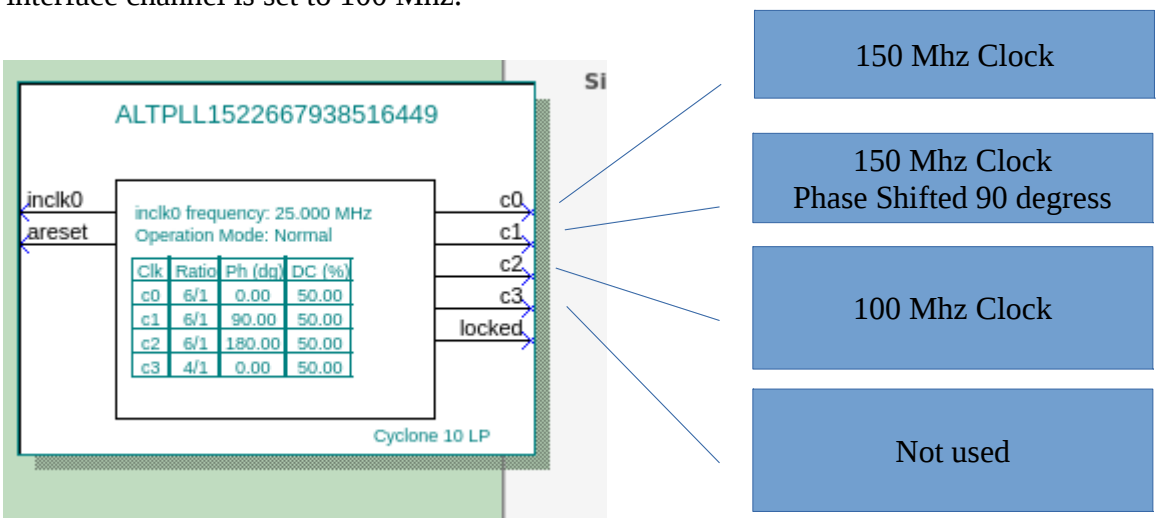
Note how i_hbus_clk_0 and i_iavs0_clk are connected to the same clock.

3.2 Option B – Different Clocks for the PSRAM/Flash memory channel and Avalon-MM channel .

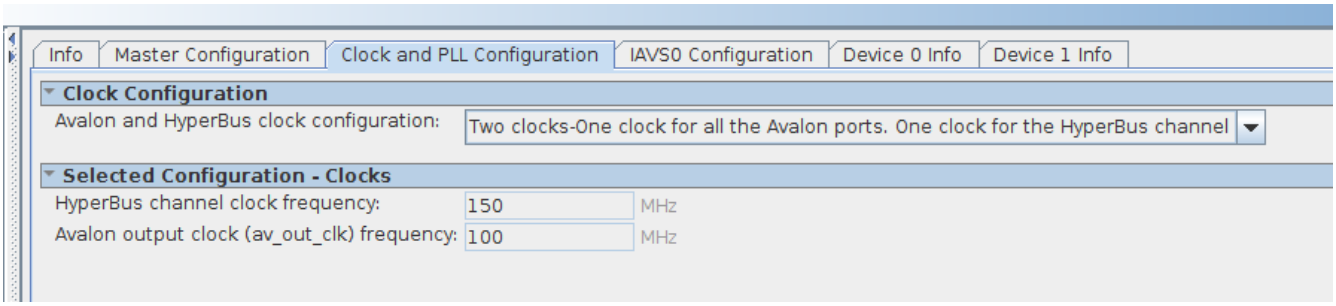
This configuration shows how to connect SLL MBMC IP so the PSRAM/Flash memory channel operates at a different clock frequency then the Avalon-MM bus interface.

3.2.1 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.



3.2.2 SLL MBMC Configuration

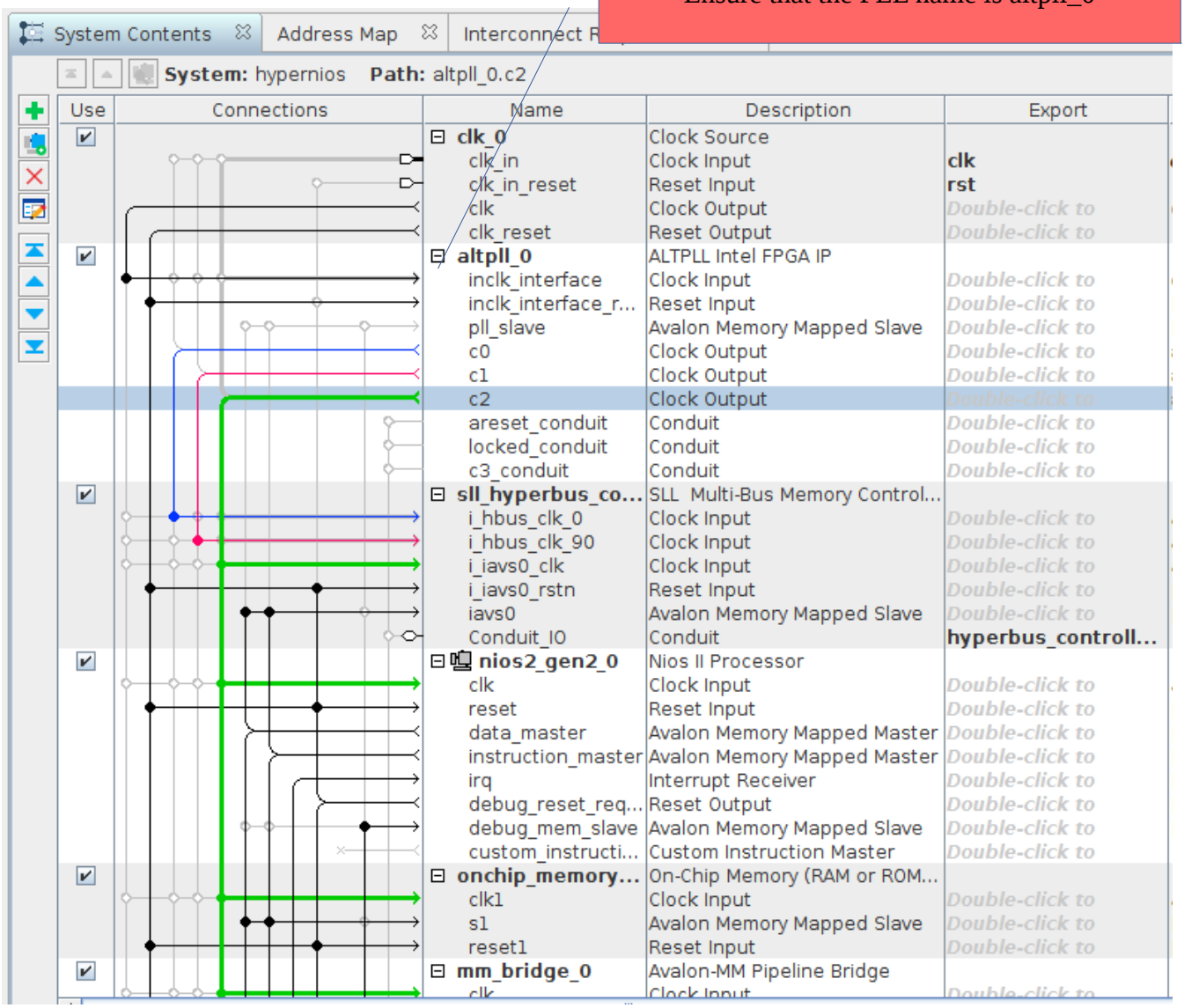


In this example, SLL MBMC IP is configured with :

- **Avalon and Hyperbus clock configuration** : **Two clocks**

3.2.3 SLL MBMC wiring

Ensure that the PLL name is altpll_0



Altera PLL output clock 0 (c0)

- Connect to i_hbus_clk_0 on S/LABS MBMC IP

Altera PLL output clock 1 (c1)

- Connect to i_hbus_clk_90 on S/LABS MBMC IP

Altera PLL output clock 2 (c2)

- Connect to i_iavs0_clk on S/LABS MBMC IP
- Connect to other Avalon-MM slaves and masters clock sinks

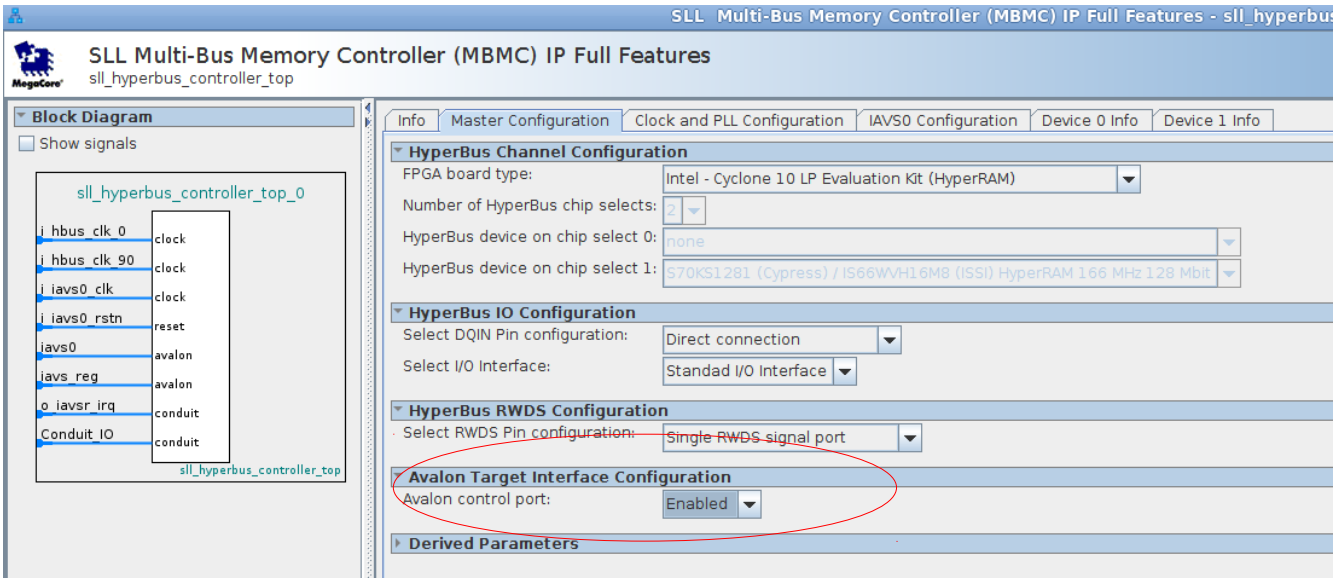
Note how i_hbus_clk_0 and i_iavs0_clk are connected to a different clock.

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.

4.0 Avalon-MM Register Control Port

When The Avalon Register Control Port is enabled, a secondary Avalon-MM slave interface becomes present (iavs_reg). This can be used for programming HyperFlash and accessing PseudoRAM registers. Please note that read and write transactions on the Avalon-MM memory port always have precedence over memory transfer requests issued by this control port.



4.1 Register Map

Address	Register Name	Direction	Comment
0x00	SLL_MBMC_CMD	Write only	Command register
0x00	SLL_MBMC_STATUS	Read only	Status register
0x04	SLL_CFG_DEV_0	Read only	Device 0 Configuration
0x08	SLL_CFG_DEV_1	Read only	Device 1 Configuration
0x0C	SLL_IRQ_MASK	Read/write	(currently not supported)
0x10	SLL_IRQ_CLR	Write Only	(currently not supported)
0x14	SLL_MBMC_CA_HI	Write Only	Upper 16-bit data CA[47:32] (command/address phase **)
0x18	SLL_MBMC_CA_LOW	Write Only	Lower 32-bit CA[31:0] (command/address phase **)
0x1C	SLL_MBMC_WDATA	Write Only	16-bit write data
0x1C	SLL_MBMC_RDATA	Read Only	16-bit return read data

4.1.1 SLL_MBMC_CMD Register

bit	*Name*	*Description*
0	RWMode	0 - write : 1- read
1	RegSpace	0 – access memory space : 1 access register space
2	CS	Device Select

When this register is written, a command is issued to the memory controller.

4.1.2 SLL_MBMC_STATUS Register

bit	*Name*	*Description*
18	CmdActive	Indicates that the Register Command is still active or pending. This bit will be cleared when SLL_MBMC_STATUS register is read
17	ReadValid	Indicates that the Return Data is Valid. This bit will be cleared when SLL_MBMC_STATUS register is read
16	PwrUp	When 1, denotes that memory controller is still powering up
15:8	FifoErr	Status of the Fifo error bits
7:4	rsv	Reserved
3:0	IrqStat	Not currently supported

4.1.3 SLL_MBMC_CA_HI and SLL_MBMC_CA_LOW Registers

When accessing the register address space, these 2 register form the 48-bit Command Address field

- SLL_MBMC_CA_HIGH should be set to CA[47:32]
- SLL_MBMC_CA_Low should be set to CA[31:0]

When accessing the memory address Space,

- SLL_MBMC_CA_HIGH should be set to 0
- SLL_MBMC_CA_Low should be set to the half-word aligned address (32-bit address >> 1).

4.1.4 SLL_MBMC_WDATA Registers

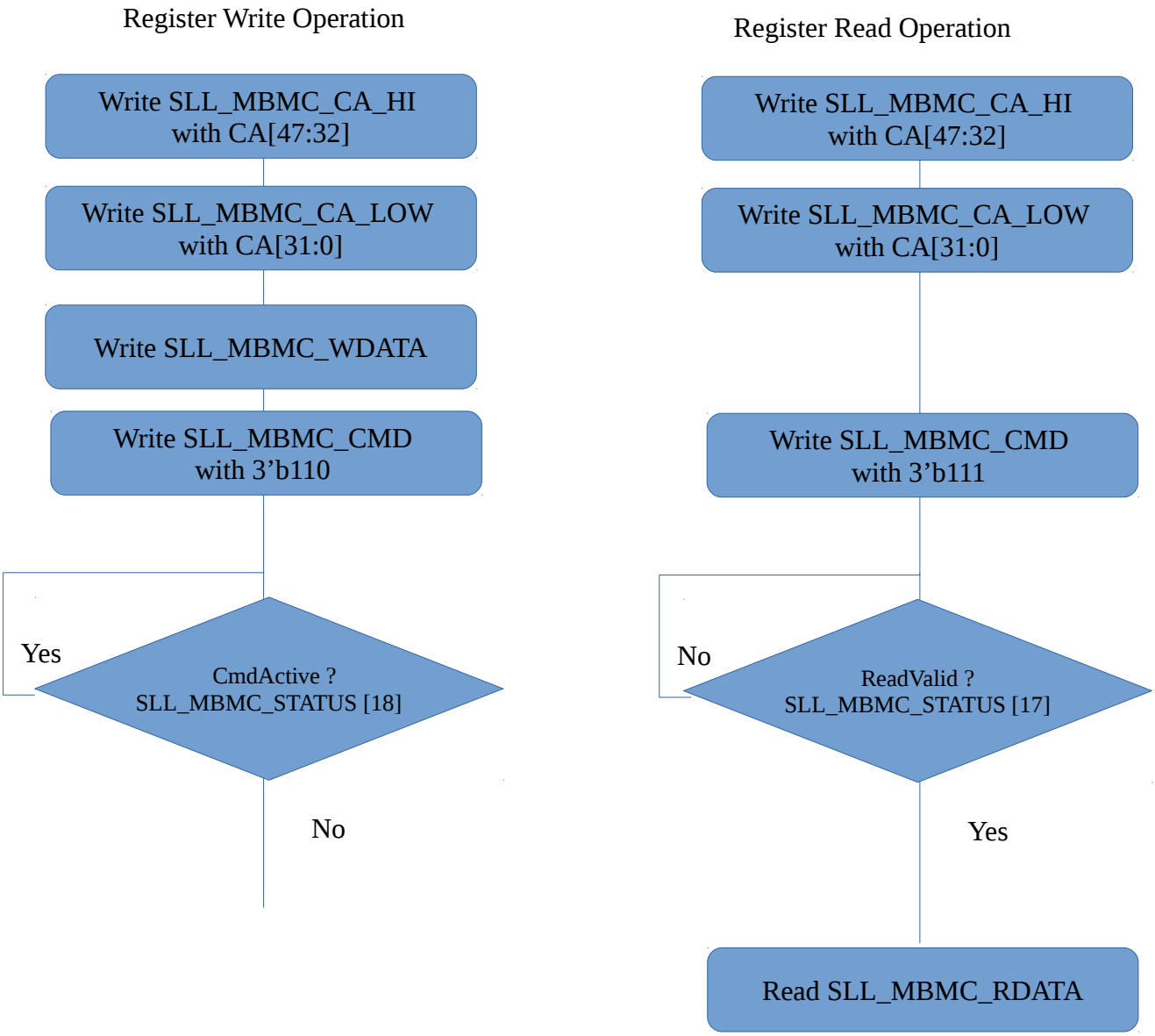
Write Data sent during write operation

4.1.5 SLL_MBMC_RDATA Registers

Return Read Data during a Read operation

4.2 Accessing PsuedoRAM Registers

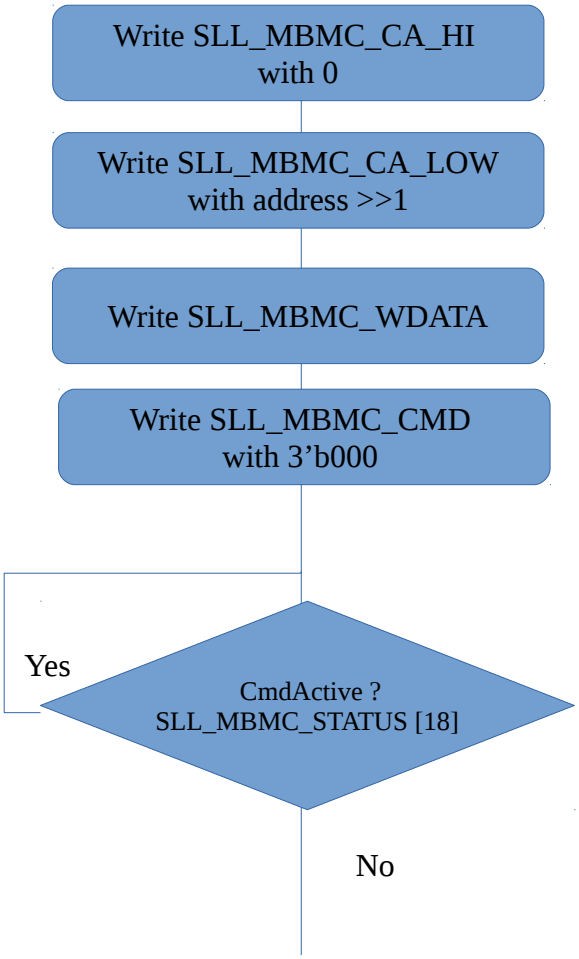
The follow flowchart shows how to use the control port interface to access the PseudoRAM registers. In this example, it is assumed that the PseudoRAM is connected to chipselect 1 (second chip select).



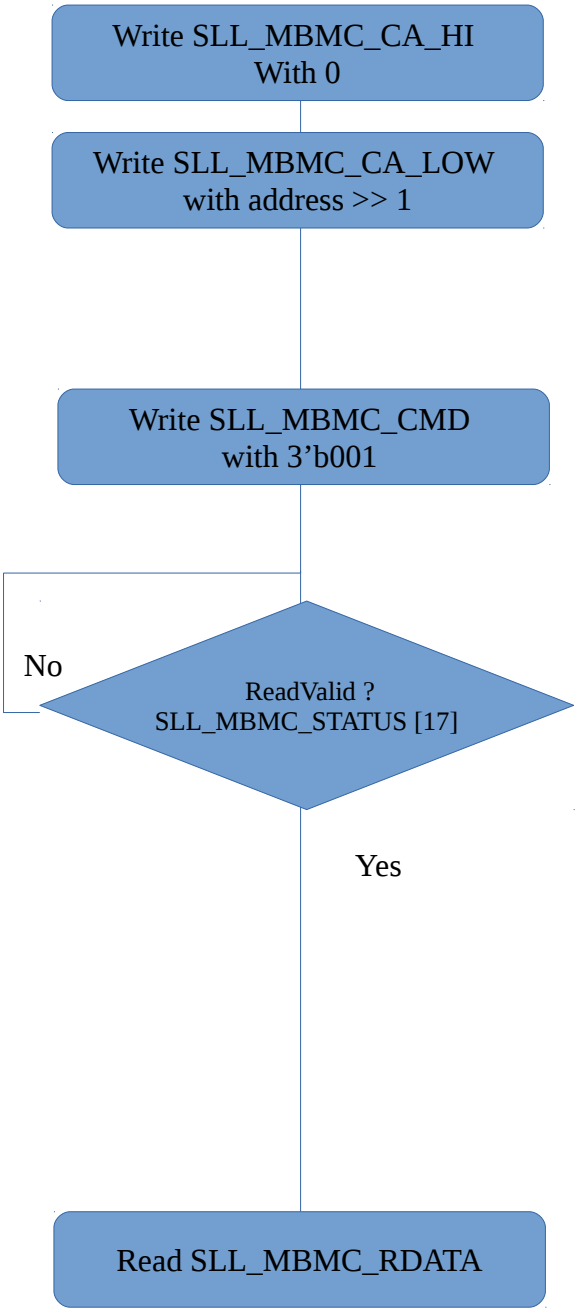
4.3 Accessing HyperFlash Memory using the Control Port

The follow flowchart shows how to use the control port interface to access the HyperFlash memory. In this example, it is assumed that the HyperFlash is connected to chipselect 0 (first chip select).

Memory Write Operation



Memory Read Operation



5.0 CycloneV Support

Targeting Higher clock speeds on Intel's Cyclone V (and Cyclone V with HPS) FPGA family.

On the Cyclone V, there are two different ways to interface with PSRAM/Flash enabled memories:

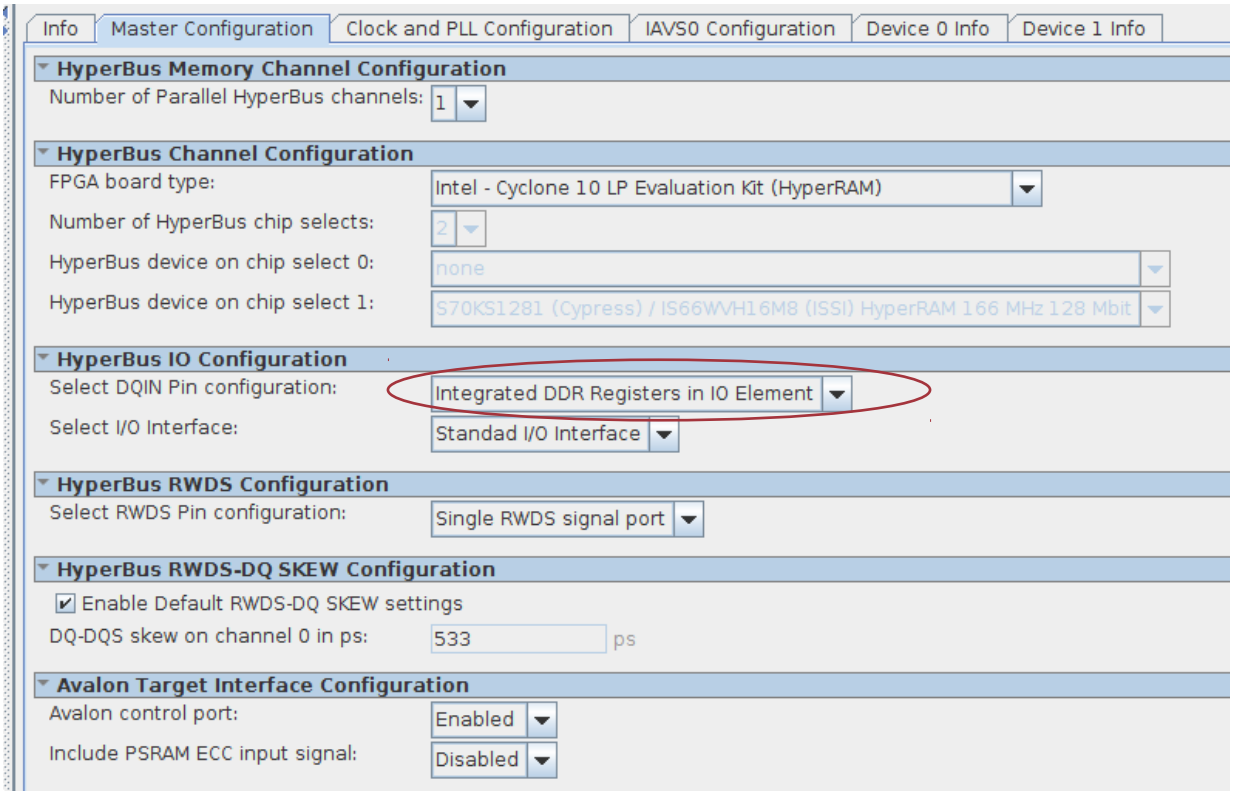
- Using a General Purpose I/O (GPIO) pin library

  This technique works on all Intel FPGA device families. However, the top clock speed achievable for accessing DDRx memories using GPIO libraries varies depending on the choice of pins, and on the FPGA device family. The GPIO library is not well suited to driving SDRAM on Cyclone V families. Maximum speed is around 50 Mhz

- Using Intel's integrated DDR/IO registers.

  These registers integrated in the I/O pads permit considerably higher clock speeds over the GPIO direct implementation on the same FPGA device. The new version of SLL MBMC IP (v3.2.x upward) now provide support this configuration.

SLL MBMC Configuration for Integrated DDIO Support



To enable Integrated DDIO support on SLL MBMC IP :

- Open SLL MBMC IP
- Locate the **Master Configuration** tab
- Locate the **Hyperbus I/O configuration** section
- Set the **Select DQIN Pin Configuration** to Integrated DDR/IO DQ input
- Set the **Select I/O interface** to **Standard I/O interface**

Additional QSF constraint

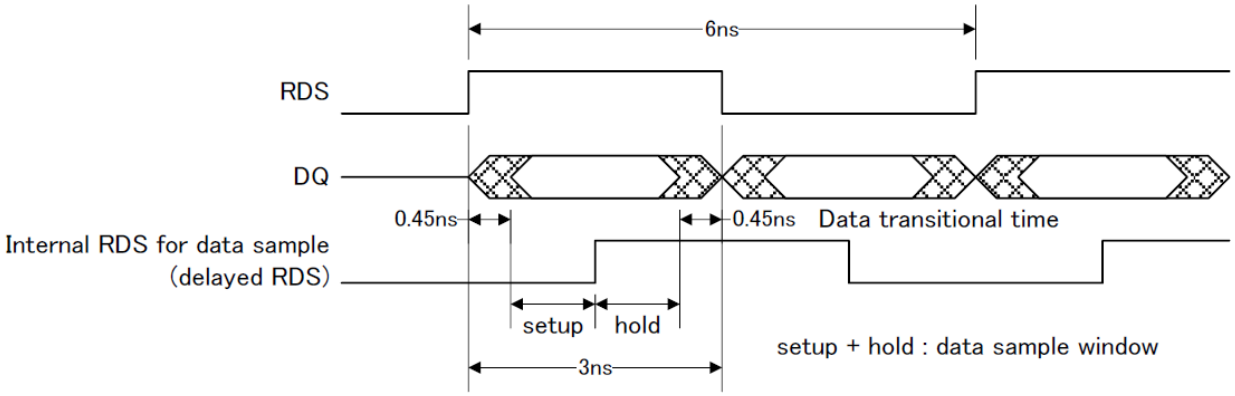
** Very important : The user needs to add the following constraint in the top level QSF file.

```
set_global_assignment -name VERILOG_MACRO "CYCLONE_V_DDIO=1"
```

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

6.0 Static Timing Consideration

6.1 Data Input Timing Constraint



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

6.1 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.

6.2 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.

7.0 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**
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)

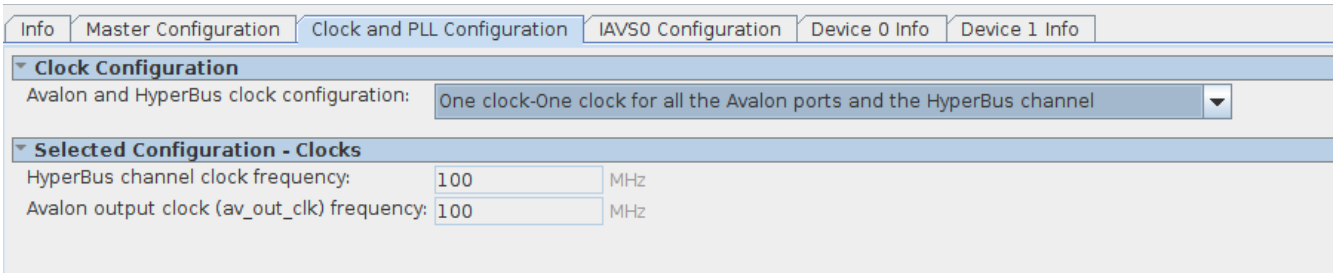
8.0 Using 3V PSRAM and HyperFlash devices

8.1 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.

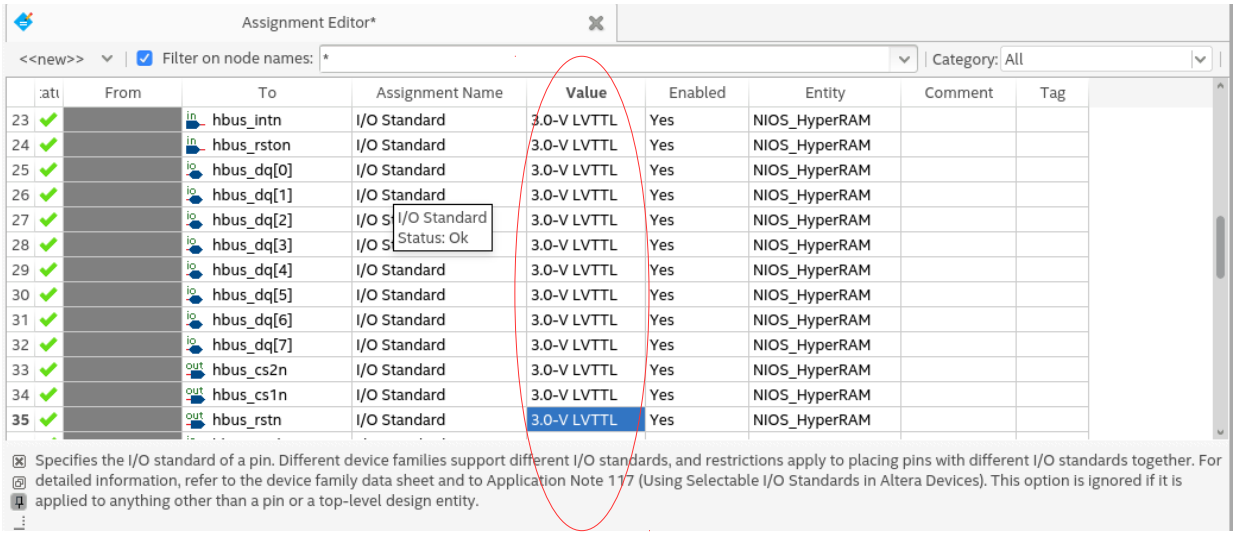
In Qsys, open SLL MBMC IP component. Locate the Clock and PLL configuration Tab. Ensure that **The Hyperbus channel clock Frequency** is 100Mhz or less.

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



8.2 Selecting the correct voltage in Quartus

The designer needs to set the HyerRAM/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.



8.3 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.