USB3 DRD Kick-off Slides

USB3 controller AE: Dehuan Meng

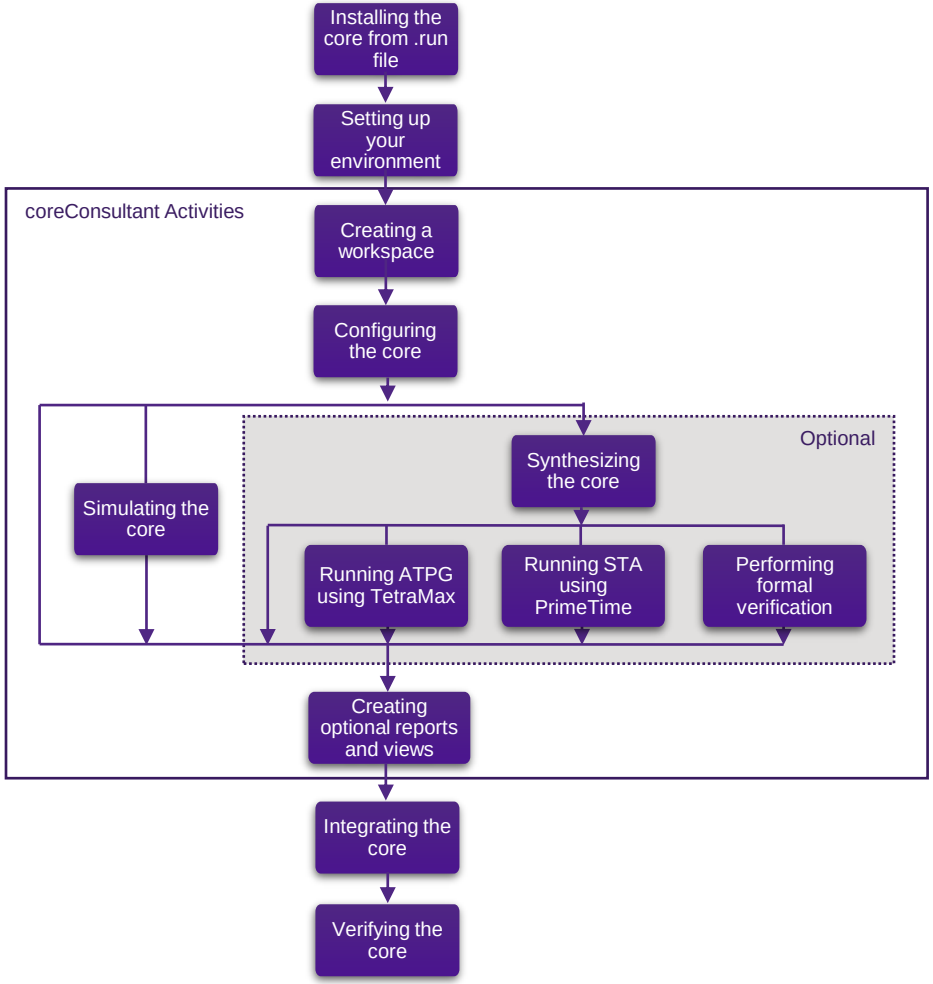
2021/09/29

Agenda

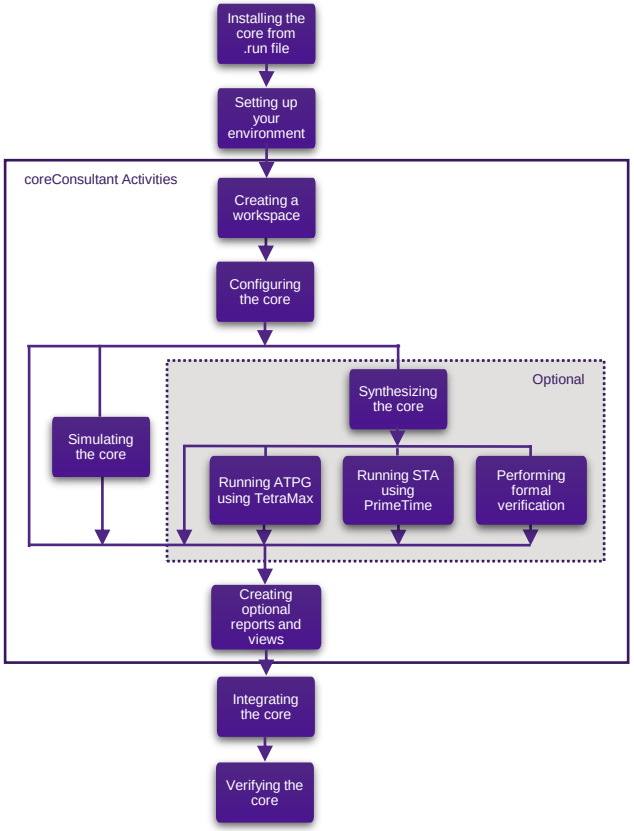
- Design flow with coreConsultant
- Important Configuration Parameters
- Built-in simulation, synthesis and spyglass in cC
- Questions

Design flow with coreConsultant

Design Flow

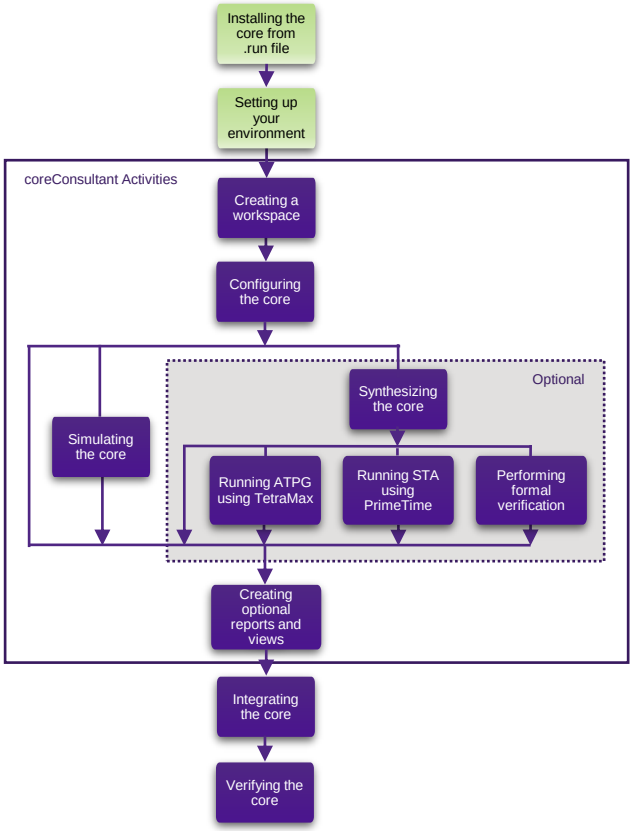


coreConsultant Tool



- GUI guides you through design flow activities
- Allows you to
  - Configure
  - Simulate
  - Synthesize
  - Export DWC_usb3 core to your design flow

Installing DWC_usb3 core



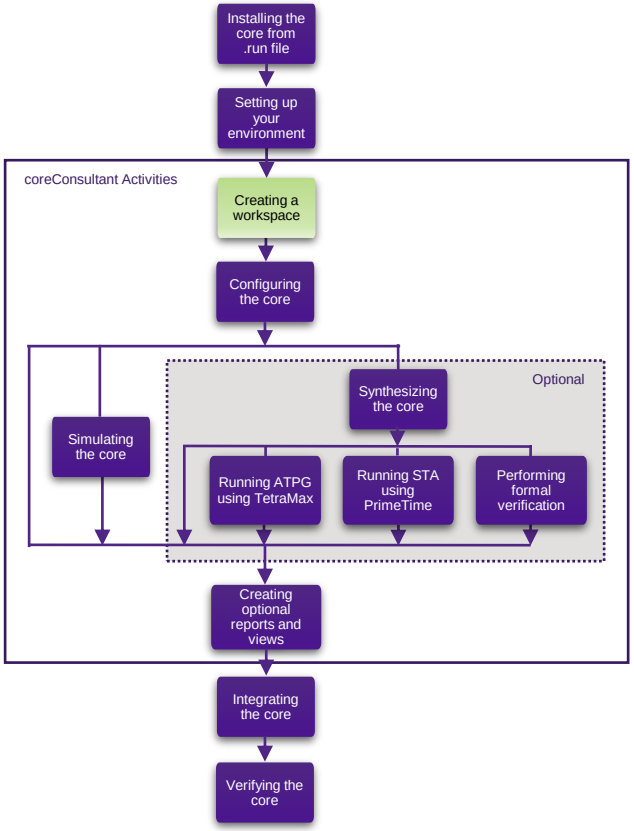
Step 1: Install

- DWC_usb3 core
- coreConsultant tool
- Setting up your environment

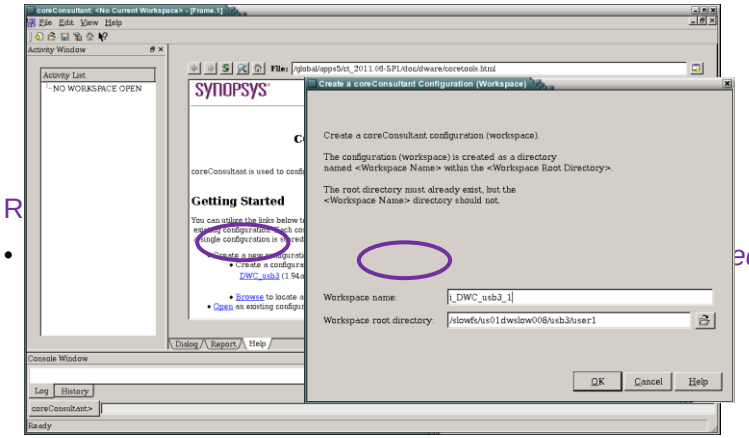
Reference:

- *DWC SuperSpeed USB 3.0 Controller Install Guide*

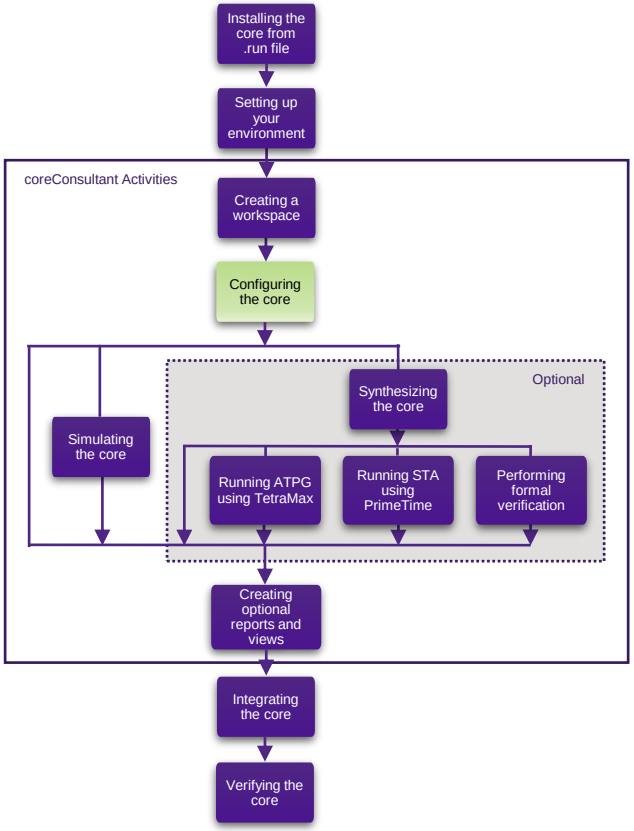
Creating a Workspace



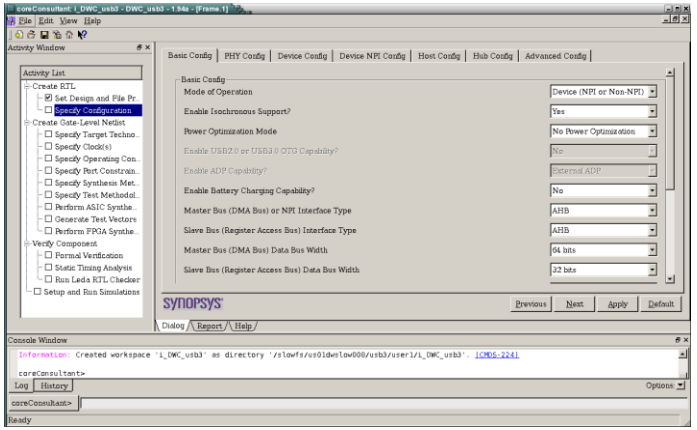
Step 2: Open coreConsultant, and create a new workspace



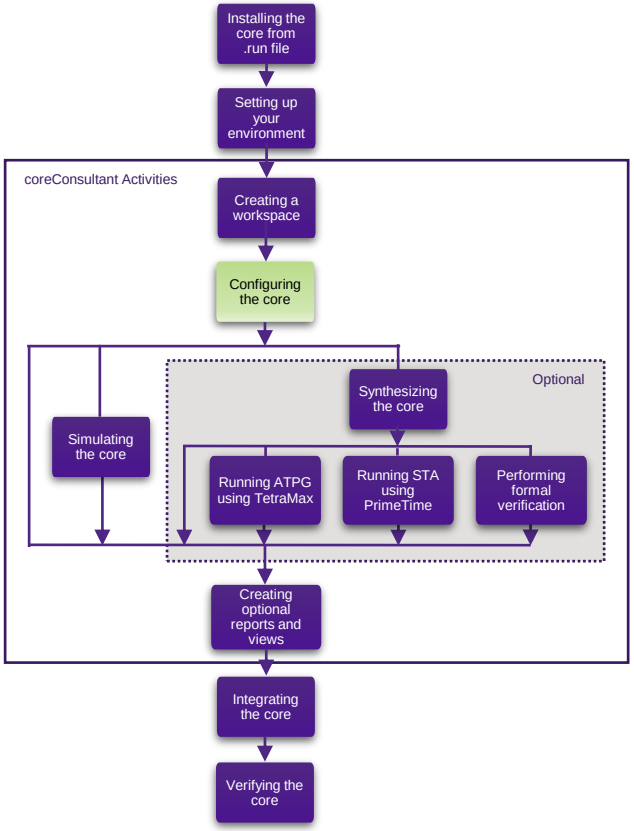
Configuring the DWC_usb3 core



Step 3: Configure the core, and generate the RTL



Configuring the DWC_usb3 core

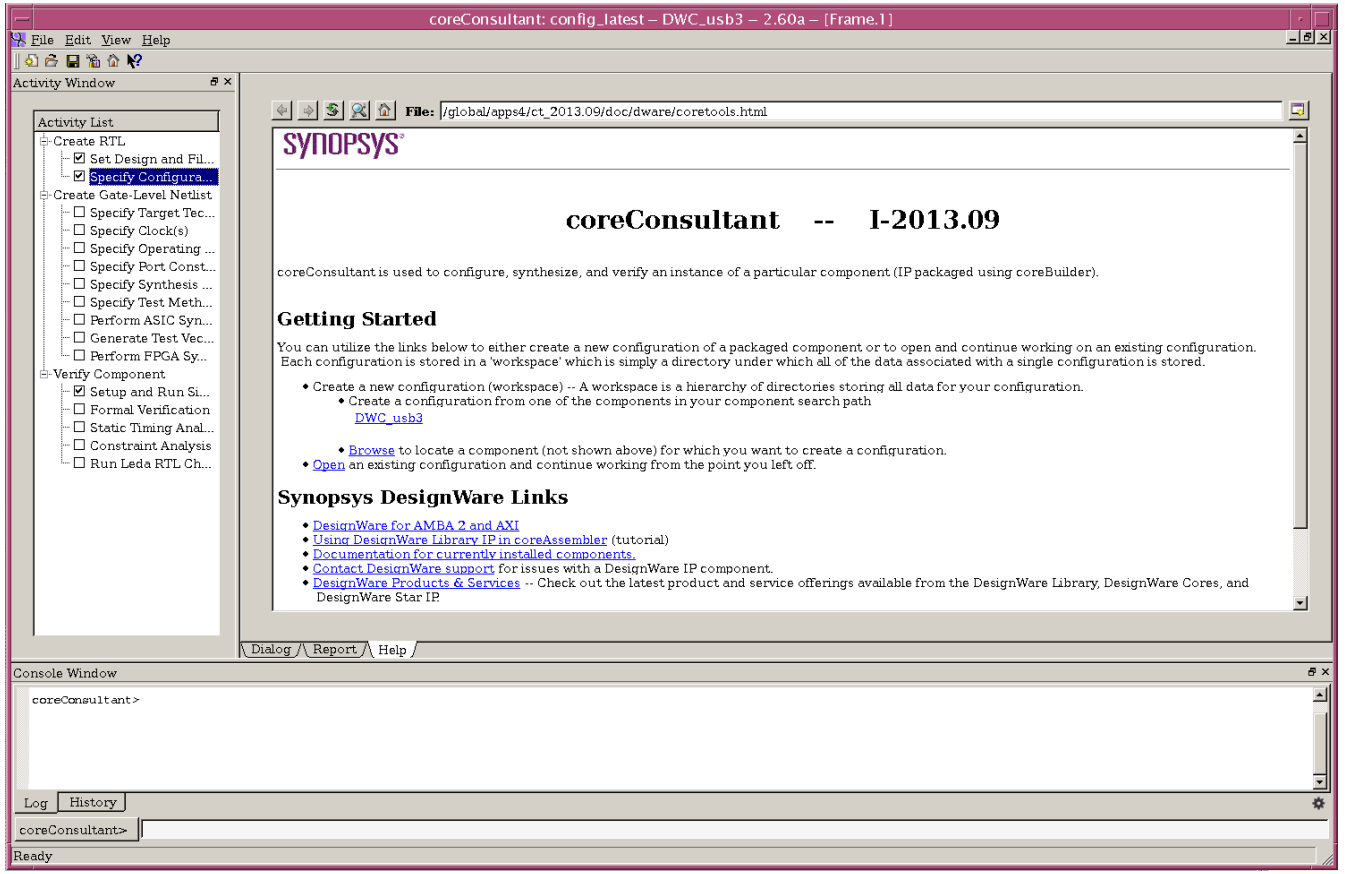


Default configure

- Helps you to get more familiarized with the design flow
- Understand a basic example configuration and it's parameters
- Understand a basic VTB test

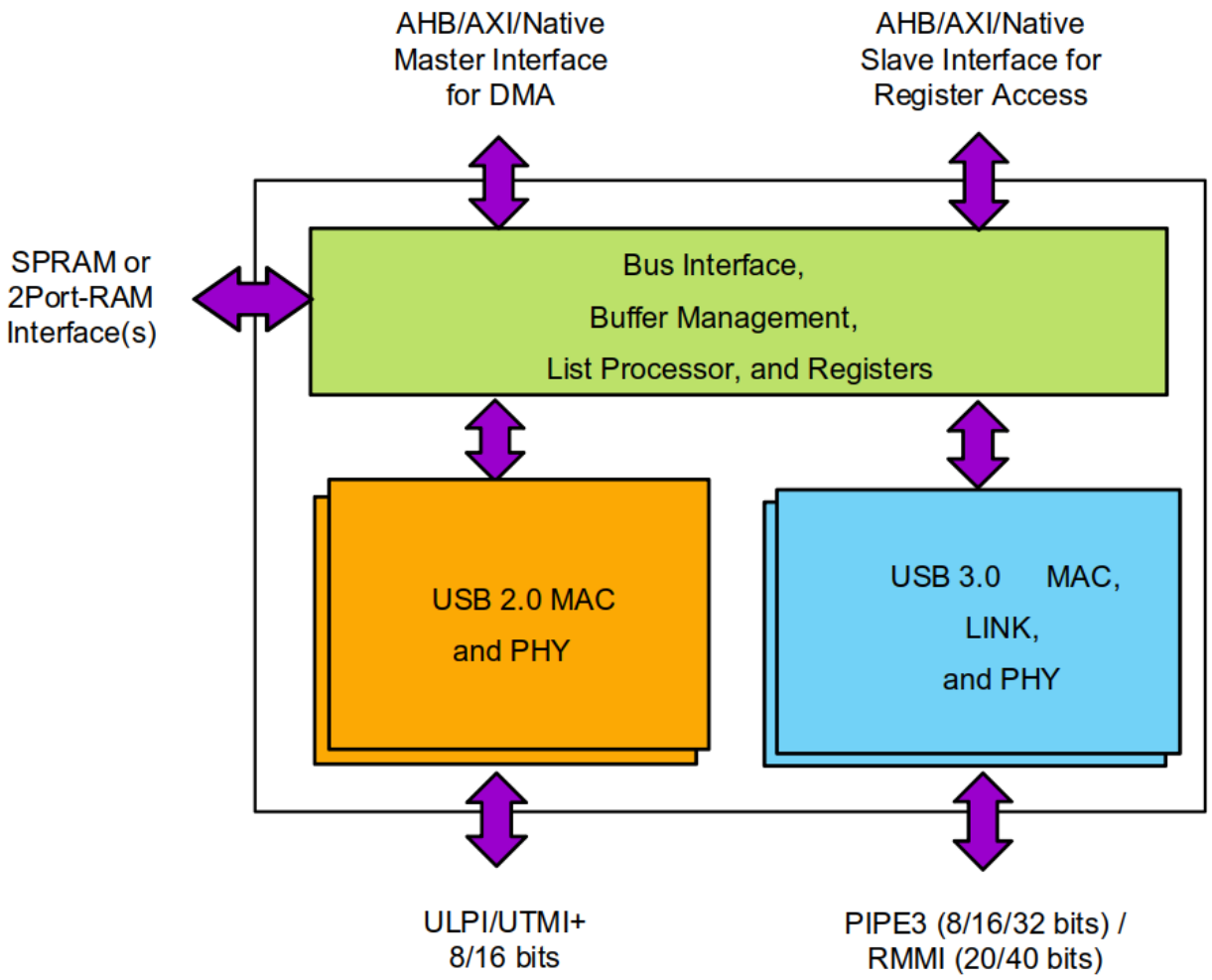
- Steps
  - Generate the core with default parameters
  - Run simulations from cC
  - Run simulations from Unix
  - View waveforms
  - Run synthesis

coreConsultant GUI



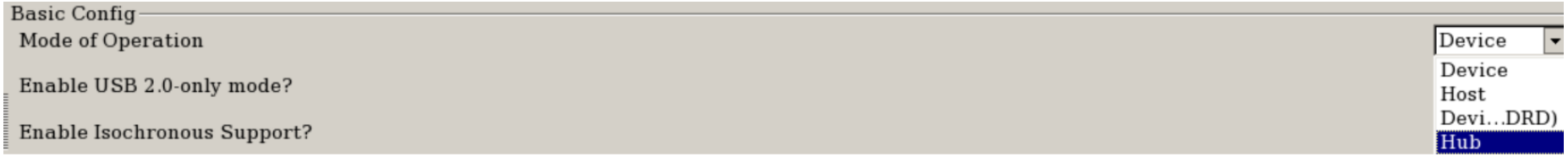
Basic Configuration Parameters

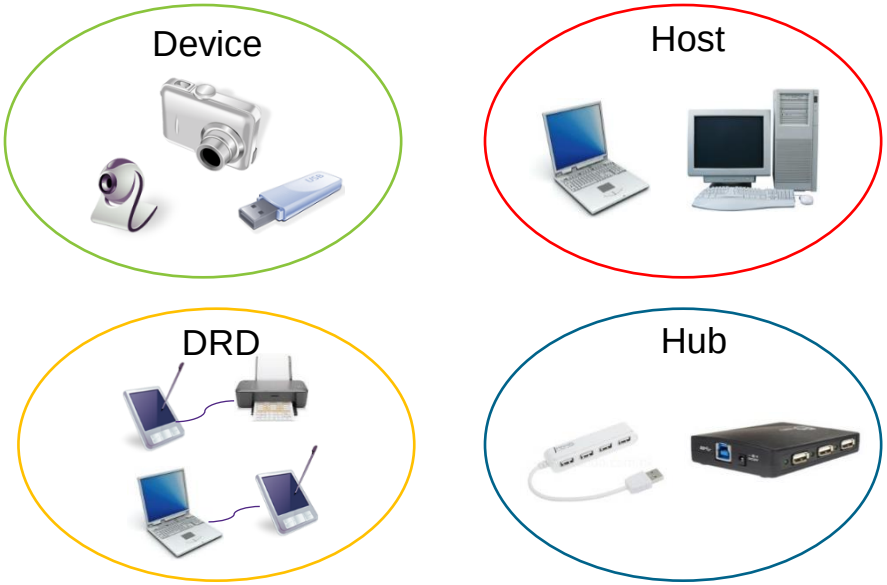
Block diagram



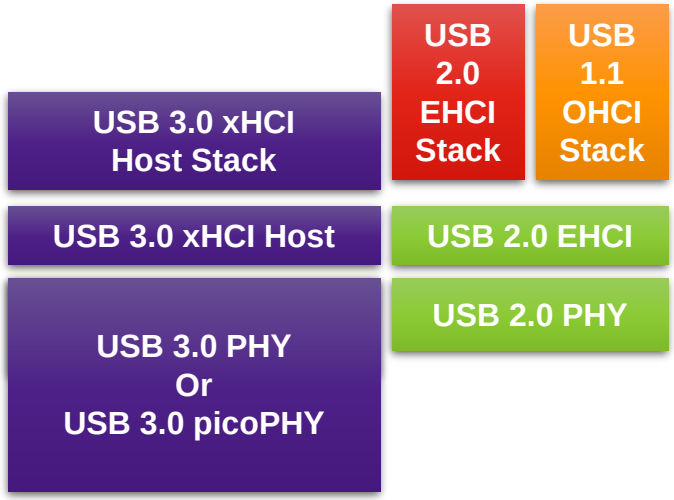
Mode of Operation

- Introduction



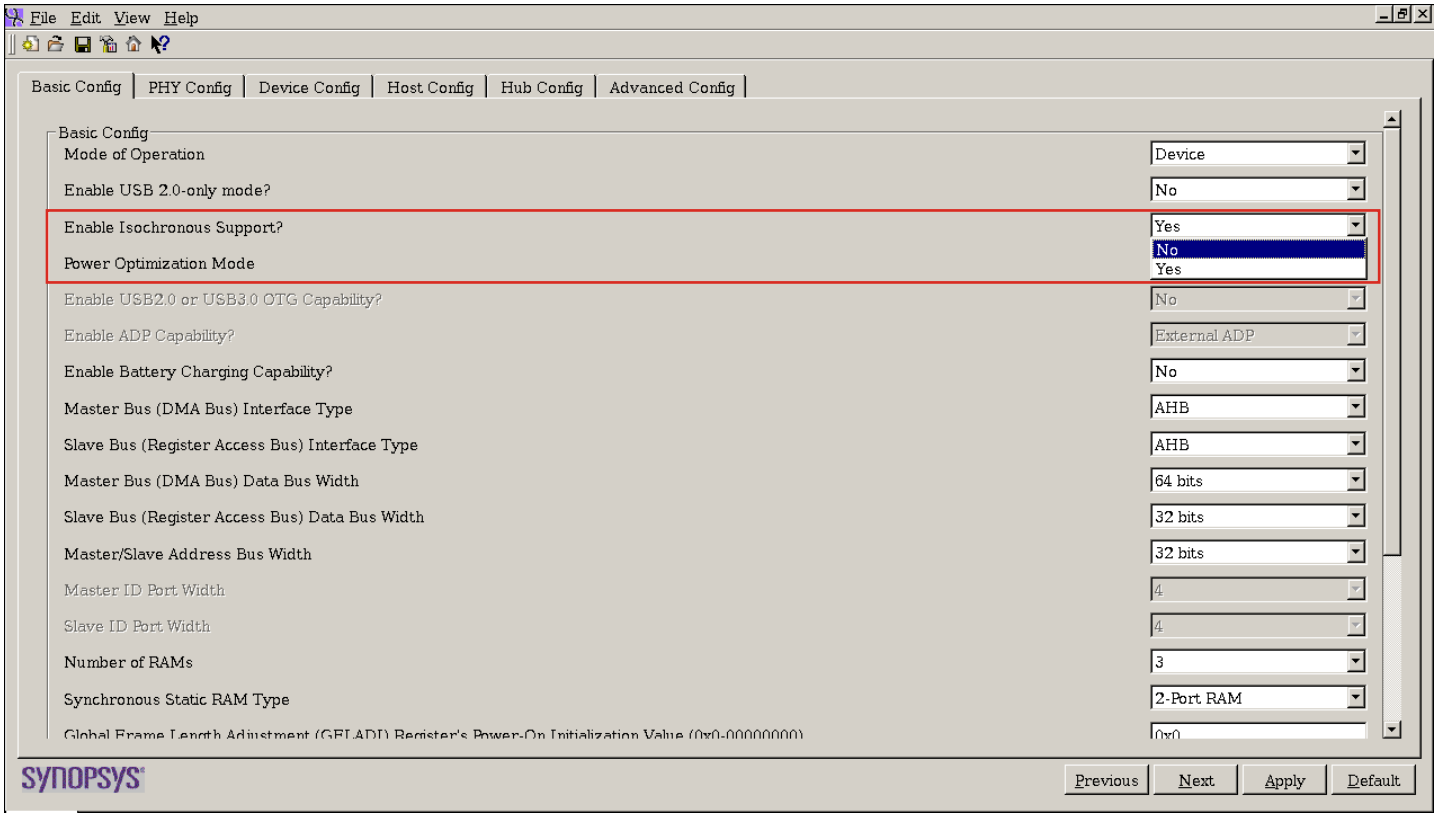


USB 2.0-only mode

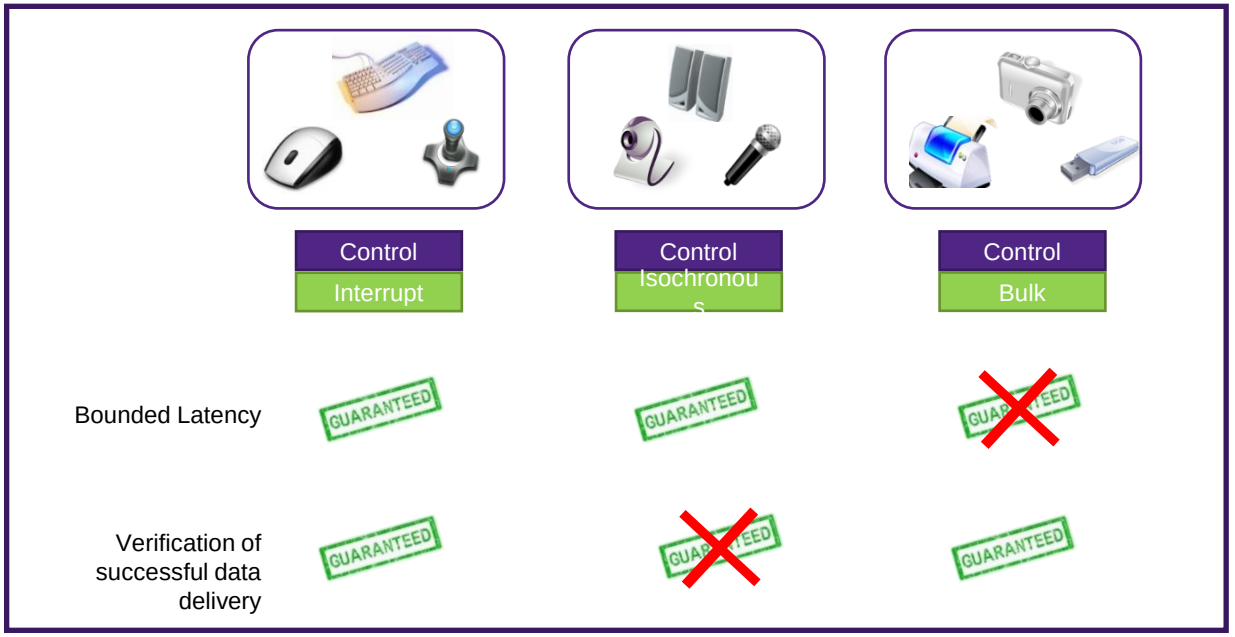


- Enabling this parameter removes SuperSpeed gates and reduces RAM requirement.
- Benefit
  - Standardize on one driver stack for all USB 3.0 and USB 2.0 projectsion
    - For Host
      - xHCI for USB 3.0 & USB 2.0,
      - Replaces USB 2.0 EHCI & USB 1.1 OHCI Driver Stacks
  - One Driver Stack for interoperability testing
  - Saves Area
    - USB 2.0 picoPHY or nanoPHY is ½ the area of USB 3.0 PHY
- Proven in Silicon

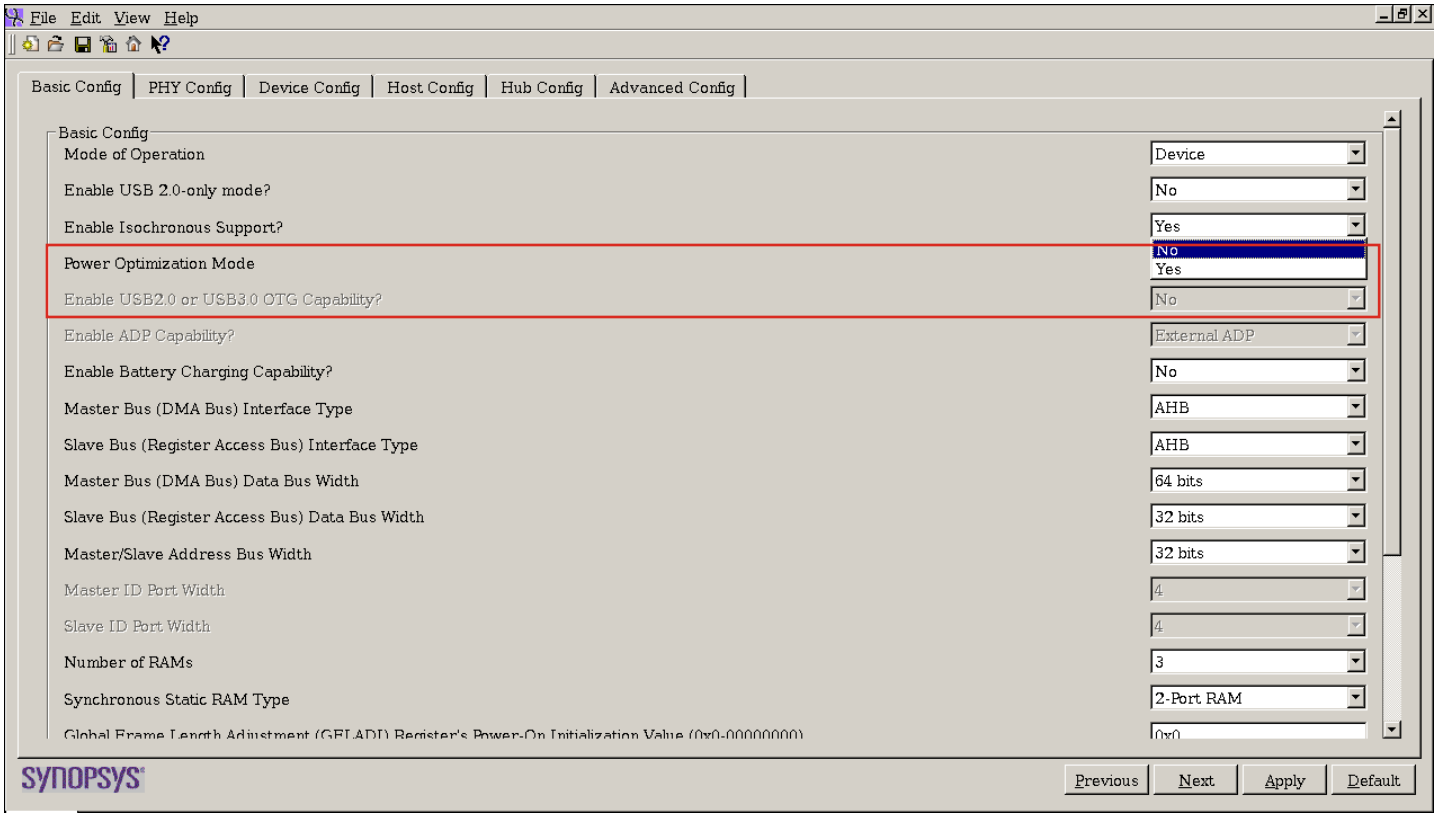
Isochronous Support



Isochronous Support



Power Optimization Mode



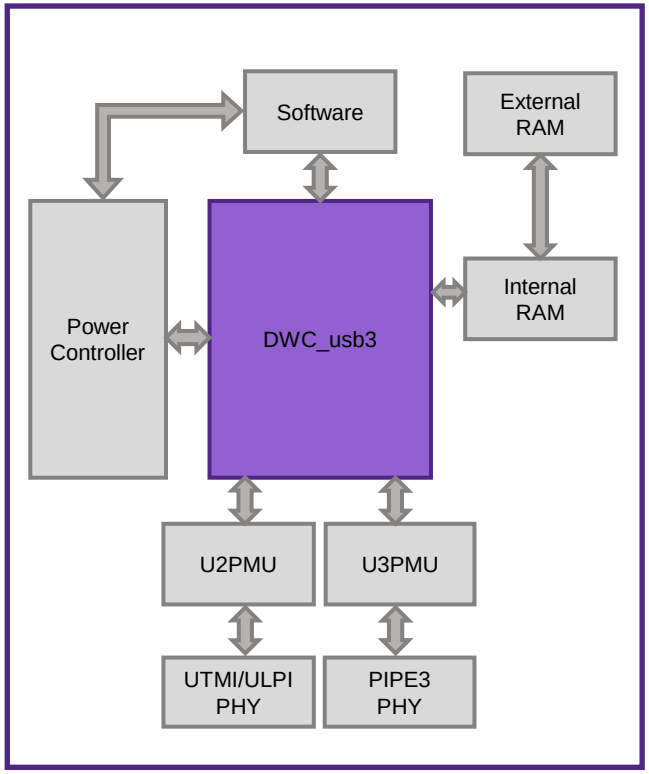
Power Optimization Mode

- Clock Gating
  - Controller turns off clocks to its internal modules when idle and
    - SS: U1, U2, or U3
    - USB 2.0: LPM-L1 or suspend
  - Controller turns off
    - RAM clock
    - BUS clock to all modules except BUS_GS (detects wakeup from software or slave interface)
    - MAC3 and PIPE3 clocks during U3
    - MAC2 and USB2 PHY clocks during suspend and LPM deep suspend

Power Optimization Mode

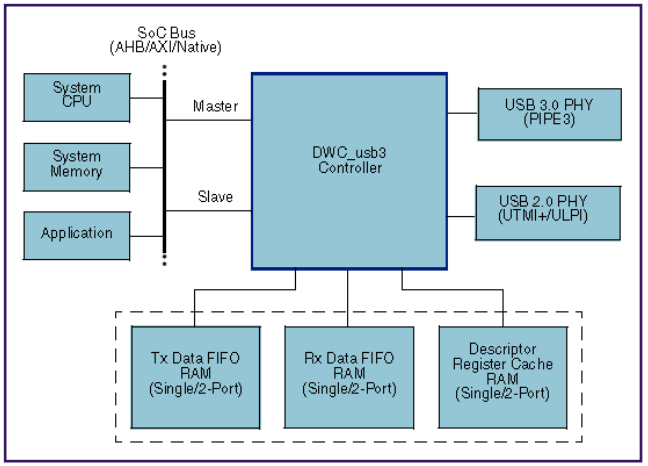
Hibernation Design Considerations

- On a wakeup request from the PMU,
  - the customer-provided power controller hardware must:
    - restore power within 100 us
    - provide a stable bus_clk within 5ms
  - the software driver must
    - re-initialize and finish the restoration process within 6ms



Bus Details

Master Bus (DMA Bus) or NPI Interface Type	AHB
Slave Bus (Register Access Bus) Interface Type	AHB
Master Bus (DMA Bus) Data Bus Width	64 bits
Slave Bus (Register Access Bus) Data Bus Width	32 bits
Master/Slave Address Bus Width	32 bits



- Master Bus (AHB, AXI, or Native)
  - DMA
- Slave Bus (AHB, AXI, or Native)
  - CSR access
- Data Bus Width:
  - Master 32, 33(only for hub), 64, 128 bits
  - Slave 32, 64, 128 bits
- Address Bus Width: 32, or 64 bits
- Depends on your SoC specification/requirements

RAM

- Number of RAMs (for registers and descriptor cache, RX buffering, and TX prefetch)
  - –2 or 3 for SS operation
  - –1 for HS-only operation
- For lower minimum ram_clk frequency, select the 3-RAM configuration
  - –The 2-RAM configuration requires 15 MHz of additional RAM bandwidth to accommodate internal Cache RAM accesses
- The minimum ram_clk frequency requirement should follow what list in databook chapter A.

RAM

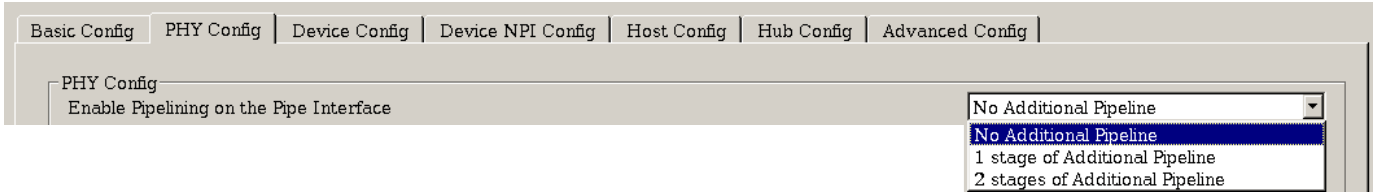
- Synchronous Static RAM Type
  - –2-Port RAM
    - Performance-sensitive applications
    - System bus frequency is low (for example, < 66 MHz)
    - Comparatively larger
  - –Single-port RAM (SPRAM)
    - Area-sensitive applications
    - Comparatively smaller
- Because USB traffic occurs concurrently with DMA traffic, it is recommended to select the 2-Port RAM configuration
- For SS Device, three 2-Port RAMs provide the best compromise between performance, lower clock frequency, and area

Other Basic Parameters

- This parameter is used to recommend allocation of 3-packet or 5-packet FIFO for burst support
  - If Yes, then 5-packet FIFO is allocated for burst support
  - If No, then 3-packet FIFO is allocated for burst support

PHY Configuration Parameters

Pipelining on the PIPE Interface



- Insert additional pipelines in the Rx and Tx of the PIPE interface
- Additional pipelines are normally not required
- FPGA: Depends on your I/O timing

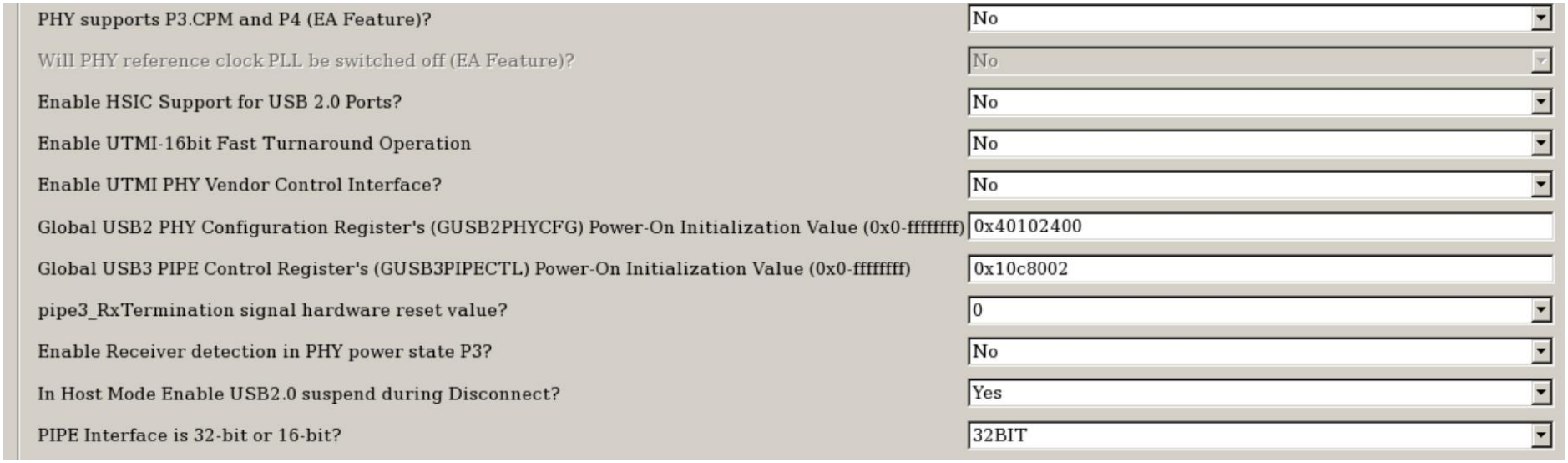
- ASIC: Needed only when,
  - The SS PHY is placed far from the controller
    - Normally, even if your PHY is placed far from your controller, there is no need to use more than one pipeline
  - There is problem in timing closure with the PIPE interface
    - Since most on-chip PHYs' PIPE interface runs at only 125 MHz@32-bits, timing closure (with no additional pipeline) should not be an issue

High Speed PHY

- Selecting ULPI
  - adds about 1.5k gates for the ULPI -> UTMI+ conversion logic because the MAC connects to the UTMI+ interface
- Selecting UTMI+ and ULPI
  - to allow for the subsequent selection of either interface as the on-chip PHY in the product
  - with an on-chip UTMI+ PHY to expose the ULPI interface as a backup against an on-chip PHY failure
  - Software can select either interface
  - the clocks for both UTMI+ and ULPI must be running

Other PHY Parameters

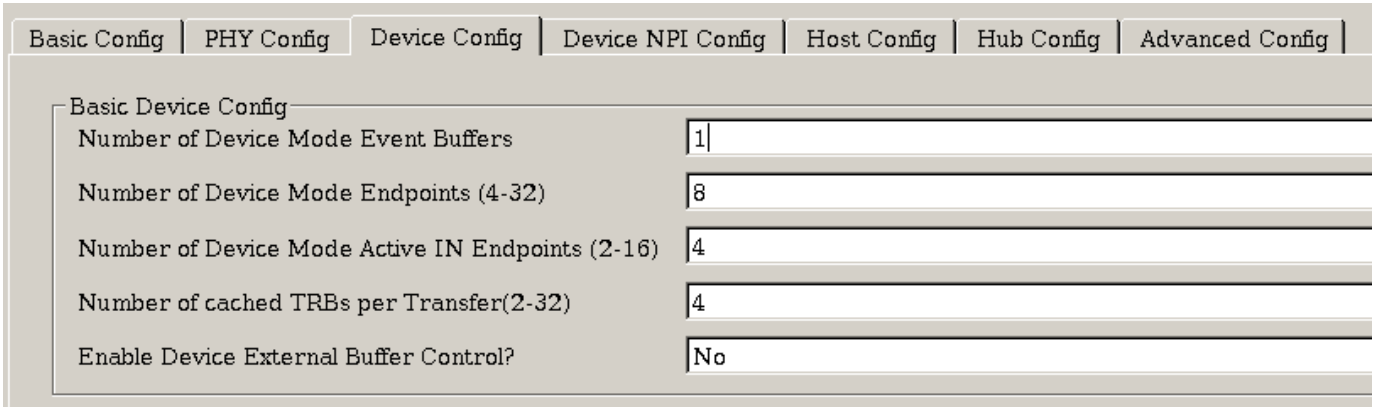
- The default values are mostly for Synopsys PHY.
- If you are using a Third-Party PHY, make sure that you understand these default values and match them for your PHY. Reference:
  - Chapter 4, “Parameters” in the DWC SuperSpeed USB 3.0 Controller Databook



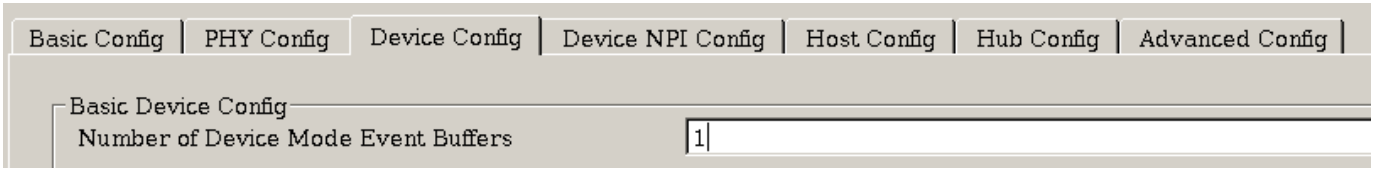
Please note there is a bug in PIPE width selection in GUI for 3.30a/b

Device Configuration Parameters

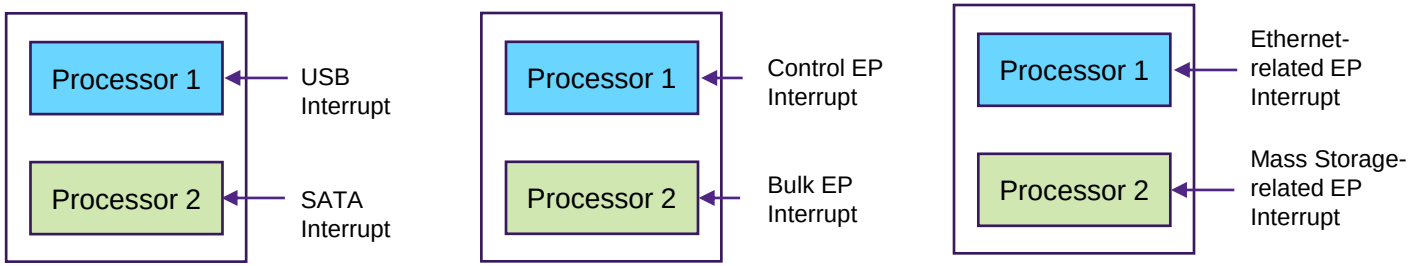
Basic Device Configuration Parameters



Number of Event Buffers



- For multiple processor or multi-core applications,
  - Configure this value accordingly
  - Valid range is 1 to 32
- Example:
  - Different peripheral interrupts can be mapped to different processors
  - Different endpoint interrupts can be mapped to different processors.



Number of Endpoints

- Specifies the number of uni-directional endpoints
- Depends on your application
- For *n* bi-directional endpoints, choose *2n* uni-directional endpoints
- Choose the maximum possible number of endpoints required in all configurations and alternate settings
- Area of one OUT/IN endpoint is 2.5/3.5Kgates plus transfer resource cache

- Additional endpoints only increases area, and does not improve performance
- The default value 8 reflects the UASP mass-storage class plus an ISOC application requirement
  - Control-OUT, Control-IN
  - Bulk- Data-OUT(S), Bulk-Data-IN(S), Bulk-Command-OUT, Bulk-Status-IN(S)
  - ISOC-OUT, ISOC-IN

Reference: For more examples, refer to 5.2 Example Device Endpoint Mapping in Different Applications in the *DWC SuperSpeed USB 3.0 Controller Databook*

Number of Active IN Endpoints

- Specifies the maximum number of Device mode IN endpoints active at any time, including control endpoint 0, which is always present.
- Determines the number of
  - –TxFIFOs
  - –TxRAMs
- The default value 4 reflects the UASP mass-storage class plus an ISOC application requirement:
  - –Control-IN
  - –Bulk-Data-IN, Bulk-Status-IN
  - –ISOC-IN

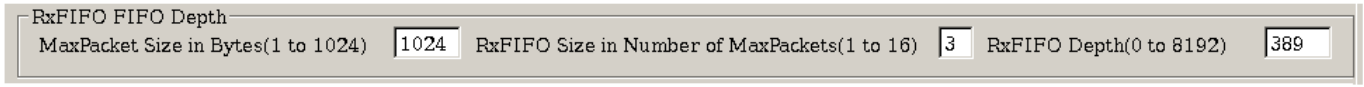
Number of Cached TRBs per Transfer

- Selects the number of Transfer Request Blocks (TRBs) per transfer that can be cached within the core
- Recommended value is one burst amount of TRBs
- If your application buffer size is less than the packet size, then the TRB cache must also have at least one packet amount of TRBs for correct operation
- The cost of TRB cache = 16 * Number of cached TRBs bytes
- Value range: 2 -32
- TRB is 16bytes size, section 3.1.2 of the programming guide

External Buffer Control

- Provides an application hardware-level control to regulate the DMA read/write of data at the packet boundaries without involving software control.
- Allows transfers to be setup in external FIFOs, the application must implement an external buffer to utilize the feature
- Allows the application layer to have a better control of how the DWC_usb3 controller will access the application's TRB buffer space, so that it can more effectively manage the data associated with the buffer space.
- The bus transfers occur in maximum-packet size lengths, and are initiated by the DWC_usb3 controller only when the application indicates that there is either a packet-sized space available to be written to, or a packet-sized amount of data is available to be read

RxFIFO Depth



- MaxPacketSize in Bytes
  - All Device OUT Endpoints share a single RxFIFO
  - Select the largest packet size for the supported endpoints
  - 1 to 1024 (Refer to USB specification for the maximum packet size requirements for each endpoint type)
- RxFIFO Size in Number of MaxPackets
  - A minimum of 3 MaxPacket FIFO is needed to support burst.
  - A 5 MaxPacket FIFO is recommended if the master DMA bus latency plus access time for an 1KB packet is larger than 2.1 Micro seconds
  - 1 to 16

IN EP0 FIFO Depth

- Selects TxFIFO-0 depth in MDWIDTH-bit words
- The recommended value is:

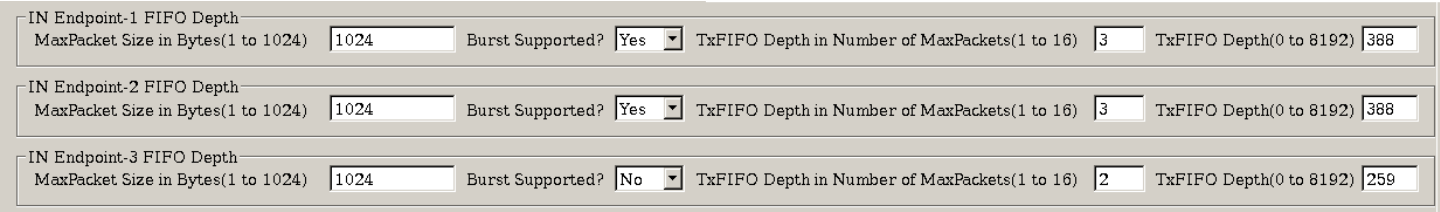
  (512 + 2*MDWIDTH-Bytes)/MDWIDTH-Bytes

- Value Range:

  (1 * (512 / DWC_USB3_MBYTES + DWC_USB3_NPI_N) + DWC_USB3_NPI_N) to 8192
- Default Value:

  (1 * (512 / DWC_USB3_MBYTES + DWC_USB3_NPI_N) + DWC_USB3_NPI_N)

IN Endpointn FIFO Depth



- TxFIFOn Maximum Packet Size in Bytes
  - 1 to 1024
- Burst Supported?

  #EP0 to buffer 512-byte packet, and all other IN endpoints are 1024 byte packet burst-capable
- TxFIFO Depth in Number of MaxPackets
  - 1 to 16

  #For a burstable SuperSpeed IN endpoint, if the system latency is more than 2.1 µsec, then a 5 to 6-packet FIFO is recommended instead 3.
- TxFIFO Depth
  - 0 to 8192 -> (3* (1024+ MDWIDTH-Bytes) + MDWIDTH-Bytes) /MDWIDTH-Bytes

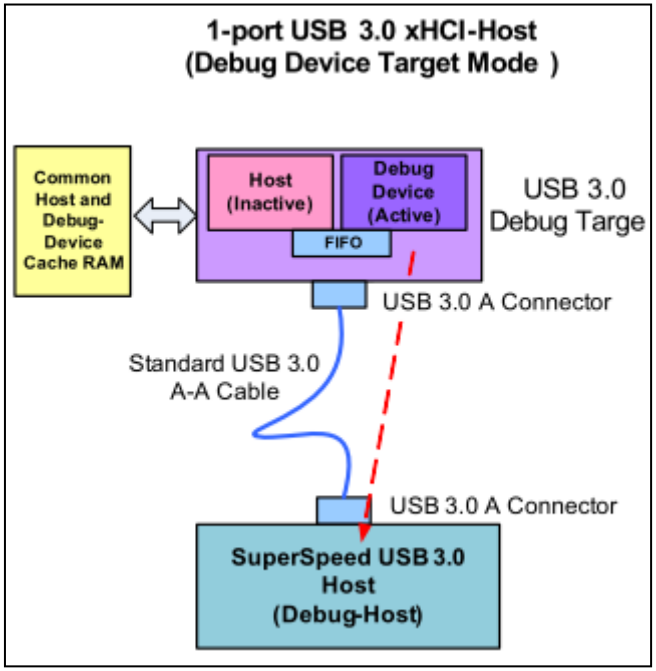
  #In addition to data, each Tx-FIFO also stores additional information and hence MDWIDTH-Bytes needed for each packet plus MDWIDTH-Bytes for each FIFO.

Host Configuration Parameters

xHCI Debug Capability

- The xHCI Debug Capability (DbC) allows one of the host ports to negotiate as an upstream (device) port. When this happens, the debug port is called the “Debug Target,” and the connected host is called the “Debug Host.”

- When the Debug Capability is enabled, additional internal RAM is needed due to:
  - Extra TxFIFOs and RxFIFOs
  - Extra space for the DbC LSP Cache
  - DbC RAM-based CSRs



Number of Interrupters

- Specifies the number of interrupters supported. The value you choose affects the gate count and memory requirements of DWC_usb3.
- If you have one processor in your SoC, then configuring multiple interrupts is not very useful. This is used when you have multi-core or multiple processors in your SoC and you plan to do load balancing.

Number of Root Hub Ports

- Specifies the number of USB2 Root Hub ports.
  - –The area of a USB 2.0 port is about 12K gates.
- Specifies the number of USB 3.0 Root Hub ports.
  - –The area of a USB 3.0 port is about 52K gates.

Number of Device / Periodic Endpoints supported

- Specifies the number of devices supported.
  - The value you choose does not increase the DWC_usb3's gate count, but the memory requirements increase to store context information.

- Specifies the number of periodic endpoints for all USB Bus instances.
  - Twenty bytes of internal RAM is allocated for each periodic endpoint.
  - For example, the Gold Tree test setup (used at USB-IF Host compliance) has nine hubs, two webcams, one printer, two keyboards, three mass storage devices, one mouse, and one headset. This configuration has 20 periodic endpoints.

Free running PHY clock on USB2 PHY

- Specifies whether your USB 2.0 PHY provides a free-running PHY clock, which is active when the clock control input is active.

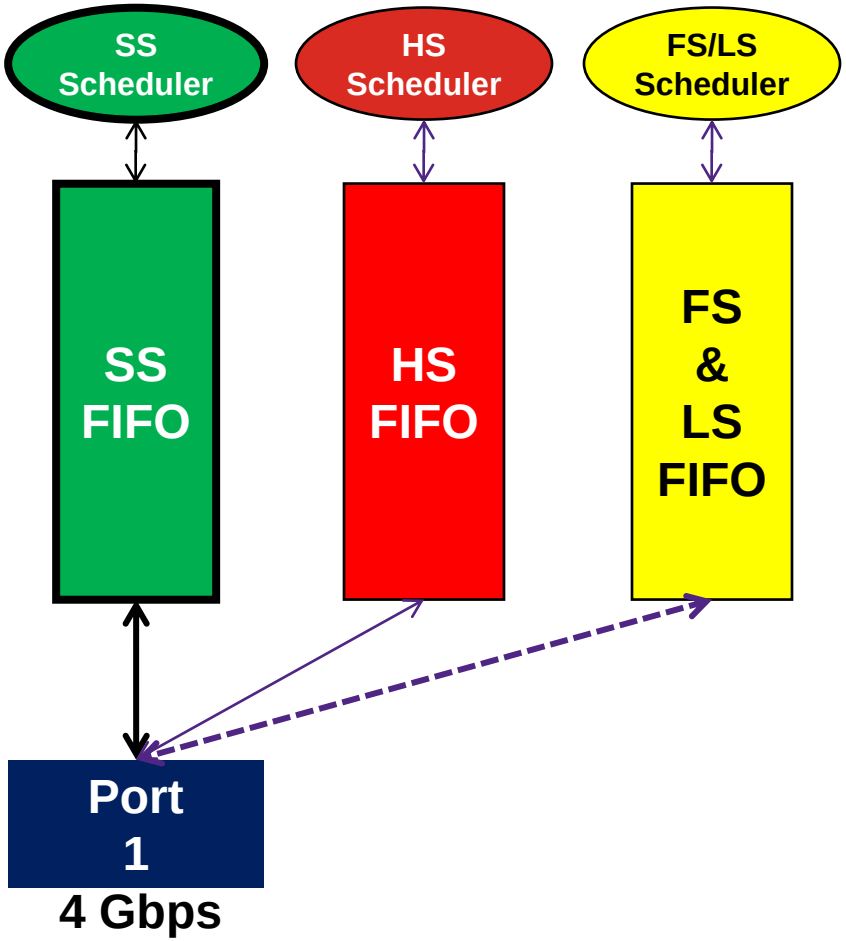
- If your USB 2.0 PHY provides a free-running PHY clock, it should be connected to the utmi_clk[0] input. The remaining utmi_clk[n] should be connected to the respective port clocks. Port-0 clock is used by the core for generating the internal mac2 clock.

- If your USB 2.0 PHY provides a free-running PHY clock, it must be connected to the utmi_clk[0] input.

- Please refer to section 5.3.2 of the userguide about integration with USB2 PHY

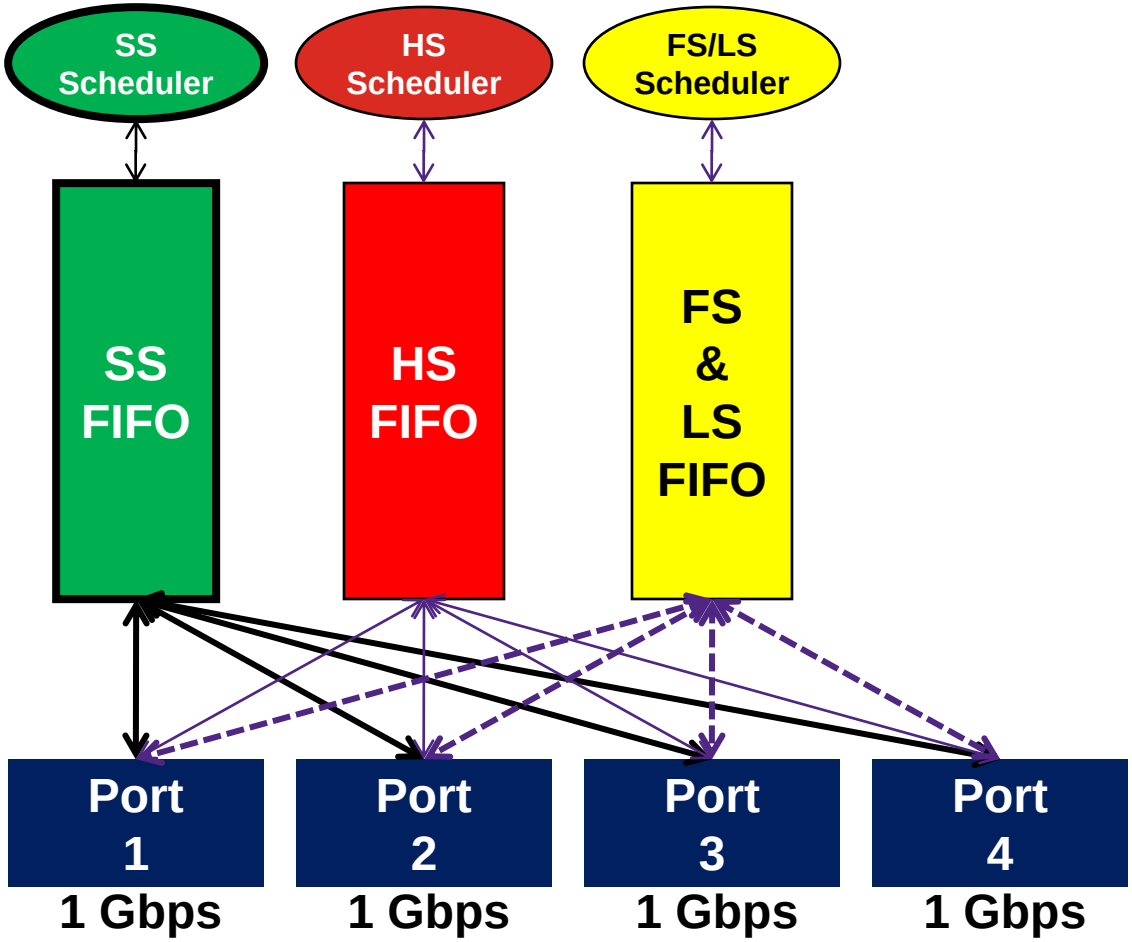
Number of USB Bus-Instances

Number of SuperSpeed USB Bus-Instances (1-4)	1
Number of High-Speed USB Bus-Instances (1-4)	1



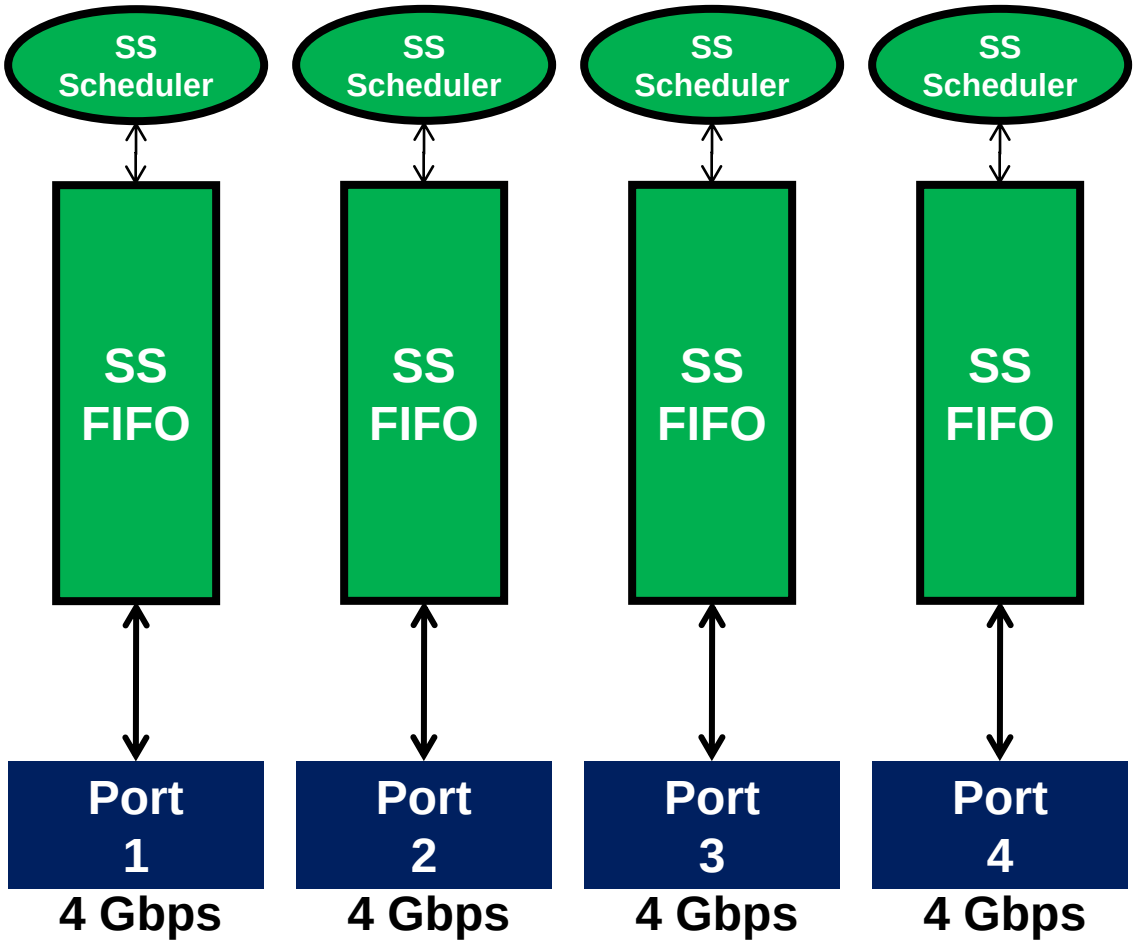
Number of USB Bus-Instances

Number of SuperSpeed USB Bus-Instances (1-4)	1
Number of High-Speed USB Bus-Instances (1-4)	1

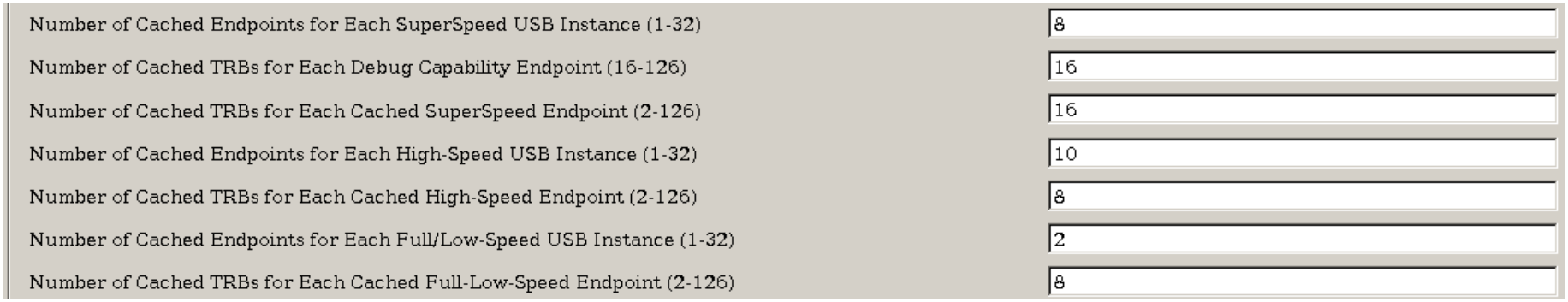


Number of USB Bus-Instances

Number of SuperSpeed USB Bus-Instances (1-4)	1
Number of High-Speed USB Bus-Instances (1-4)	1



Number of Cached Endpoints / TRBs



- Specifies the number of Endpoints / TRBs that can be cached in the core for each cached USB Bus instance / endpoint.
- Internal RAM are allocated for each cached USB Bus instance / endpoint.

Size of the RxFIFO / TxFIFO

Size of the SuperSpeed RxFIFO in Number of 1024-byte Packets - Per SS Bus-Instance (1-32)	3
Size of the High-Speed RxFIFO in Number of 1024-byte Packets - Per HS Bus-Instance (1-16)	2
Size of the Full-Speed/Low-Speed RxFIFO in Number of 1024-byte Packets - Per FS/LS Bus-Instance (1-4)	1
Size of the SuperSpeed TxFIFO in Number of 1024-byte Packets - Per SS Bus-Instance (1-32)	4
Size of the High-Speed TxFIFO in Number of 1024-byte Packets - Per HS Bus-Instance (1-16)	2
Size of the Full-Speed/Low-Speed TxFIFO in Number of 1024-byte Packets - Per FS/LS Bus-Instance (1-4)	1

- Specifies the size of the each link speed RxFIFO / TxFIFO in number of 1024-byte packets. Each link speed USB Bus instance requires one RxFIFO / TxFIFO.
- Note: It is recommended not to reduce the size from the default value.

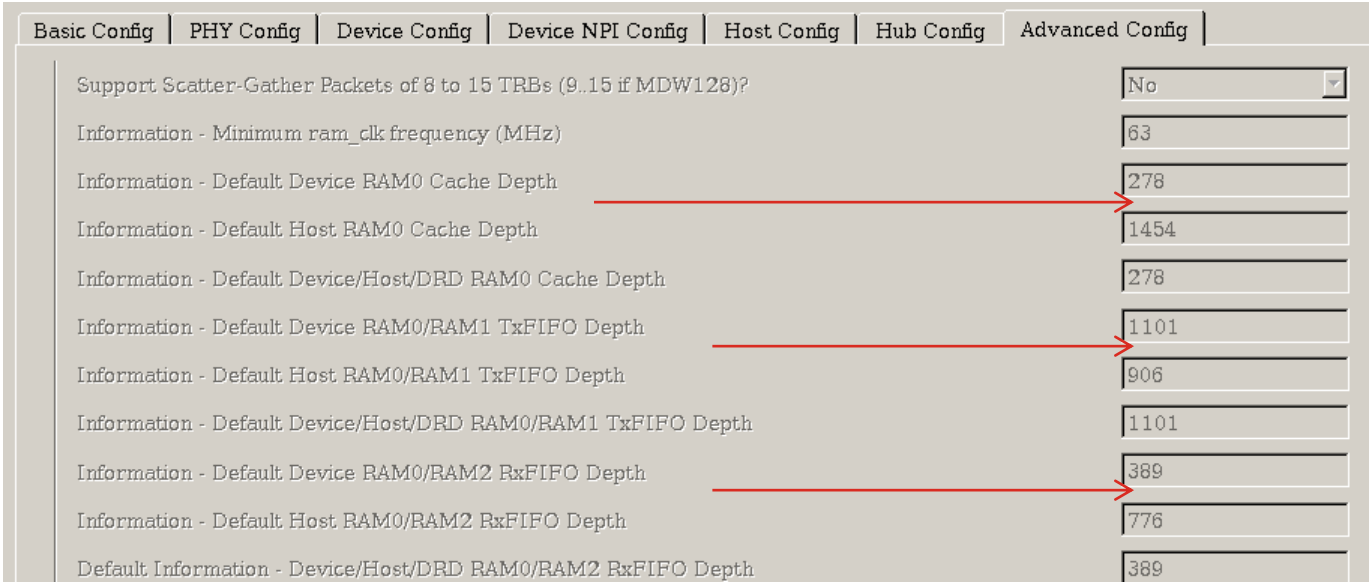
Advanced Configuration Parameters

RAM Calculations

- Cache
  - Internal Queues
  - RAM Registers
  - Device TRB Cache
- TxFIFOs (One per EP)
- RxFIFO (Shared among all EPs)

RAM	3-RAM Configuration	2-RAM Configuration	1-RAM Configuration
RAM0	Descriptor Cache	Descriptor Cache RxFIFOs	Descriptor Cache RxFIFOs TxFIFOs
RAM1	TxFIFOs	TxFIFOs	-
RAM2	RxFIFOs	-	-

RAM Calculations



- The minimum values for Cache, TxFIFO and RxFIFO depth is displayed in CCT based on the previous parameters selected.
  - It is recommended that you do not reduce the RAM depth.
  - Depending up on your system bus latency, you can only increase it.

RAM Calculations

- Total TxFIFO Depth
  - Recommended: EP0 to buffer 512-byte packet, and all other IN endpoints are 1024 byte packet burst-capable (each requires 3 packet buffer)
  - For example, if you have two Bulk IN endpoints in addition to Control endpoint, the RAM1 Depth requirement is:
    - ((512+2*MDWIDTH-Bytes) + (2*(3*(1024+MDWIDTH-Bytes))+ MDWIDTH-Bytes)) /MDWIDTH-Bytes
  - For a burstable SuperSpeed IN endpoint, if the system latency is more than 2.1 µsec, then a 5 to 6-packet FIFO is recommended instead 3.
  - In addition to data, each Tx-FIFO also stores additional information and hence MDWIDTH-Bytes needed for each packet plus MDWIDTH-Bytes for each FIFO.
- Value: 32 to 65536

RAM Calculations

- RxFIFO Depth
  - (3 * 1024 + 40 to 64)/MDWIDTH-Bytes
    - 3 * 1024
      - For a burstable SuperSpeed IN endpoint, if the system latency is more than 2.1 µsec, then a 5 to 6-packet FIFO is recommended instead 3.
    - 40 to 64
      - These bytes are to store up to three back-to-back SETUP packets; when the Master Bus width is 128 bits, 48 bytes are needed; else 24 bytes are needed.
      - 16 bytes are for clock domain crossing tolerance
- Value 32 to 65536

Enable FPGA Implementation?

- Hardware validation or driver development with an FPGA platform

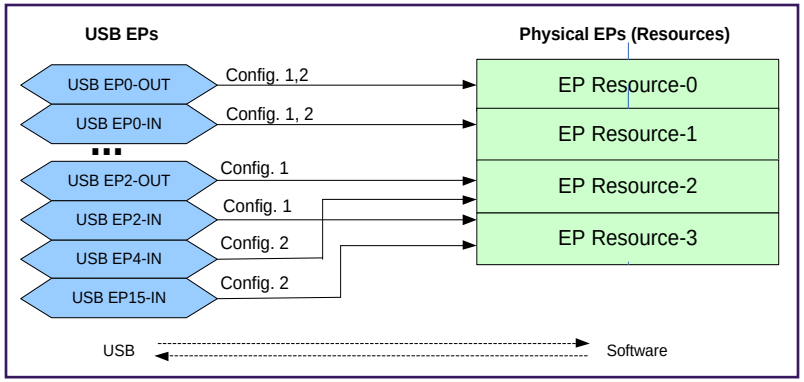
Enable Logical to Physical EP Mapping?

- Selects whether the device core supports a flexible or fixed logical to physical endpoint mapping.
- Fixed mapping
  - improves timing
  - reduces gate count (5K gate reduction)
- Flexible mapping
  - Area-optimized for multiple applications/USB set-configuration modes
  - During coreConsultant configuration, you can configure *n* number of physical endpoints (register and FIFO resources)
  - Post-silicon, the software can map these physical endpoints to USB endpoints, even if the USB endpoints are not contiguous

Enable Logical to Physical EP Mapping?

For example:

Configuration 1	Configuration 2
USB EP0-OUT	USB EP0-OUT
USB EP0-IN	USB EP0-IN
USB EP2-OUT	USB EP4-IN
USB EP2-IN	USB EP15-IN



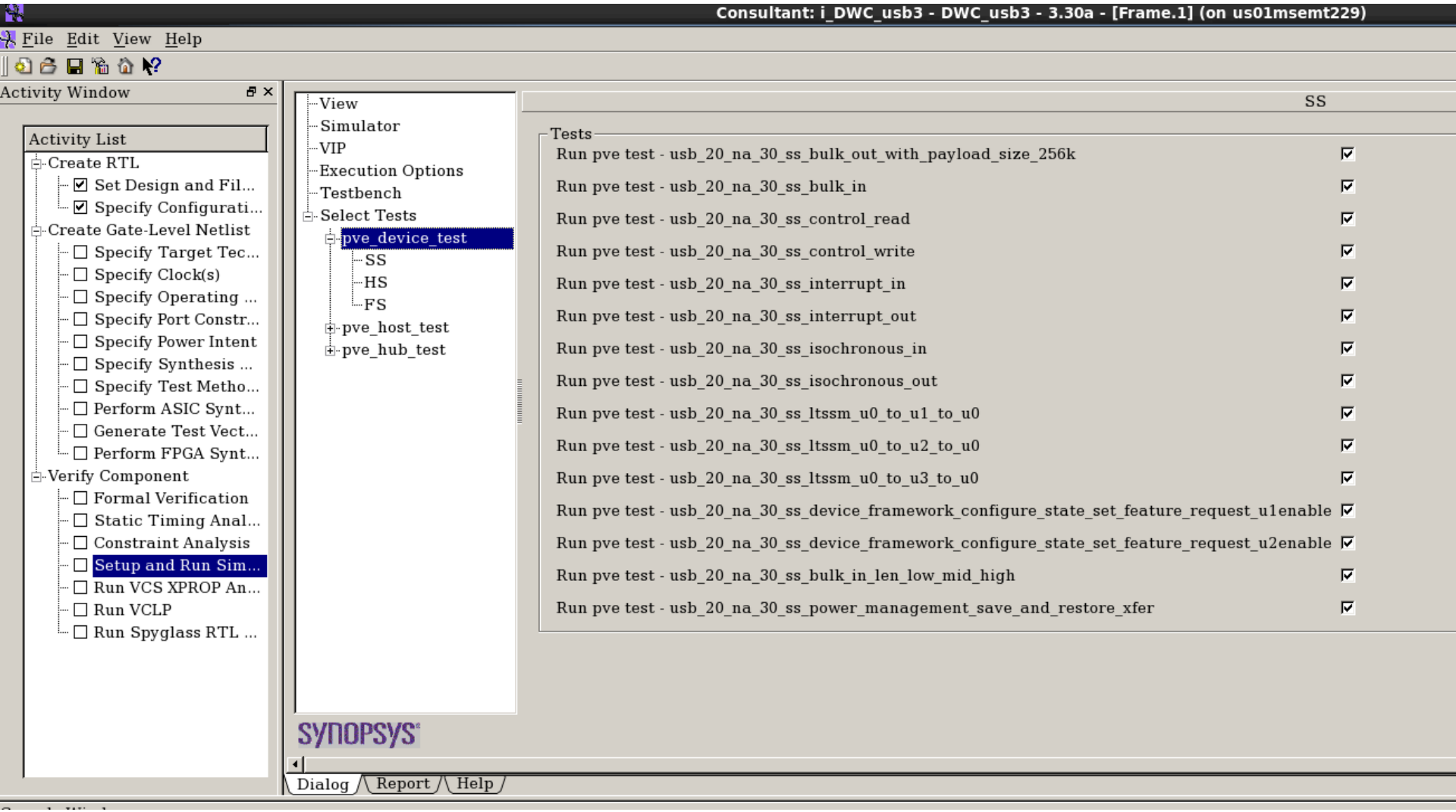
With flexible endpoint mapping hardware needs only 4 endpoints instead of 32

Other Advanced Configuration Parameters

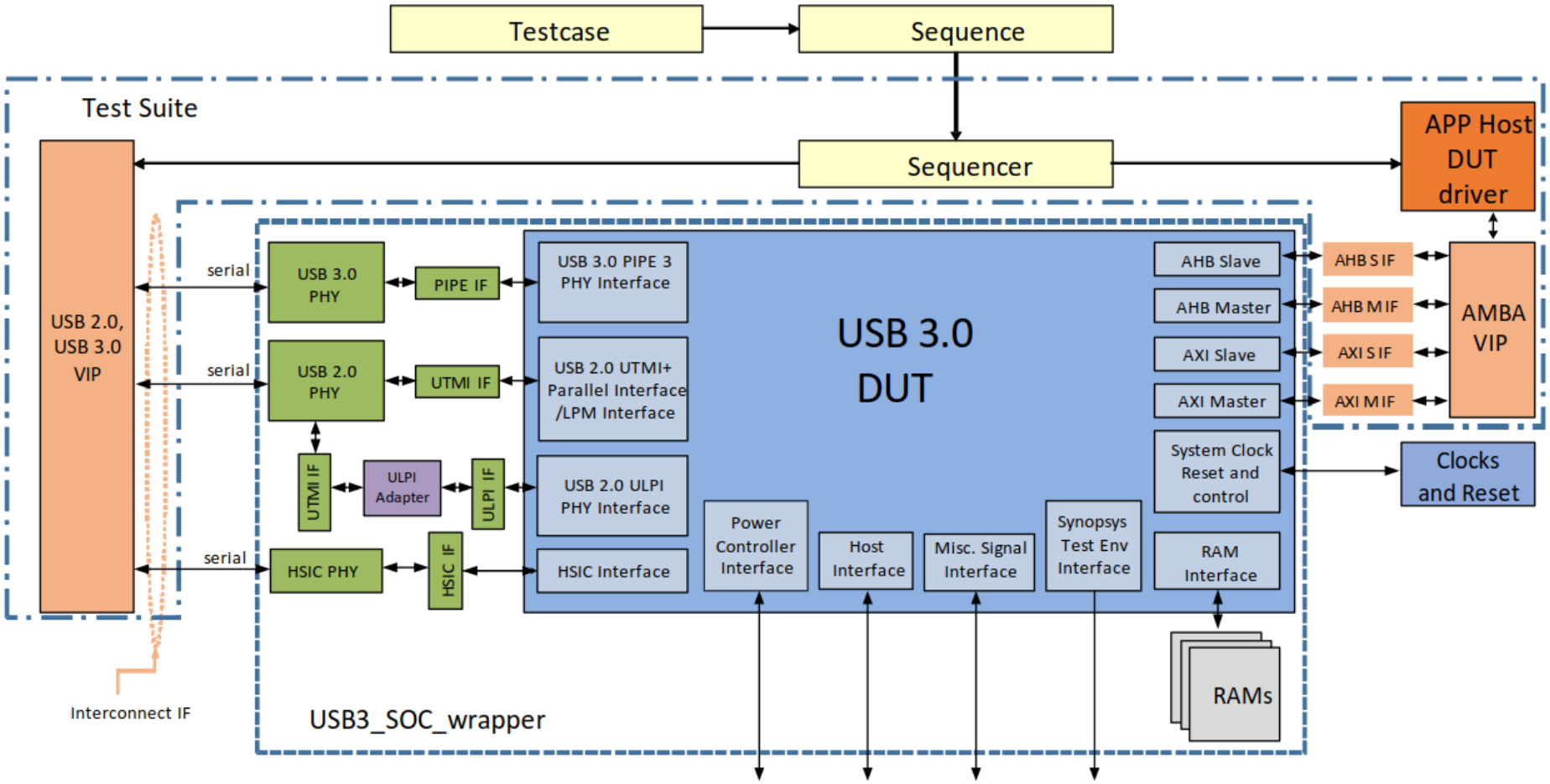
- For details on the following configuration parameters, refer:
  - Chapter 4, “Parameters” in the *DWC SuperSpeed USB 3.0 Controller Databook*

Simulation environment

Simulation in coreConsultant



UVM based PVE

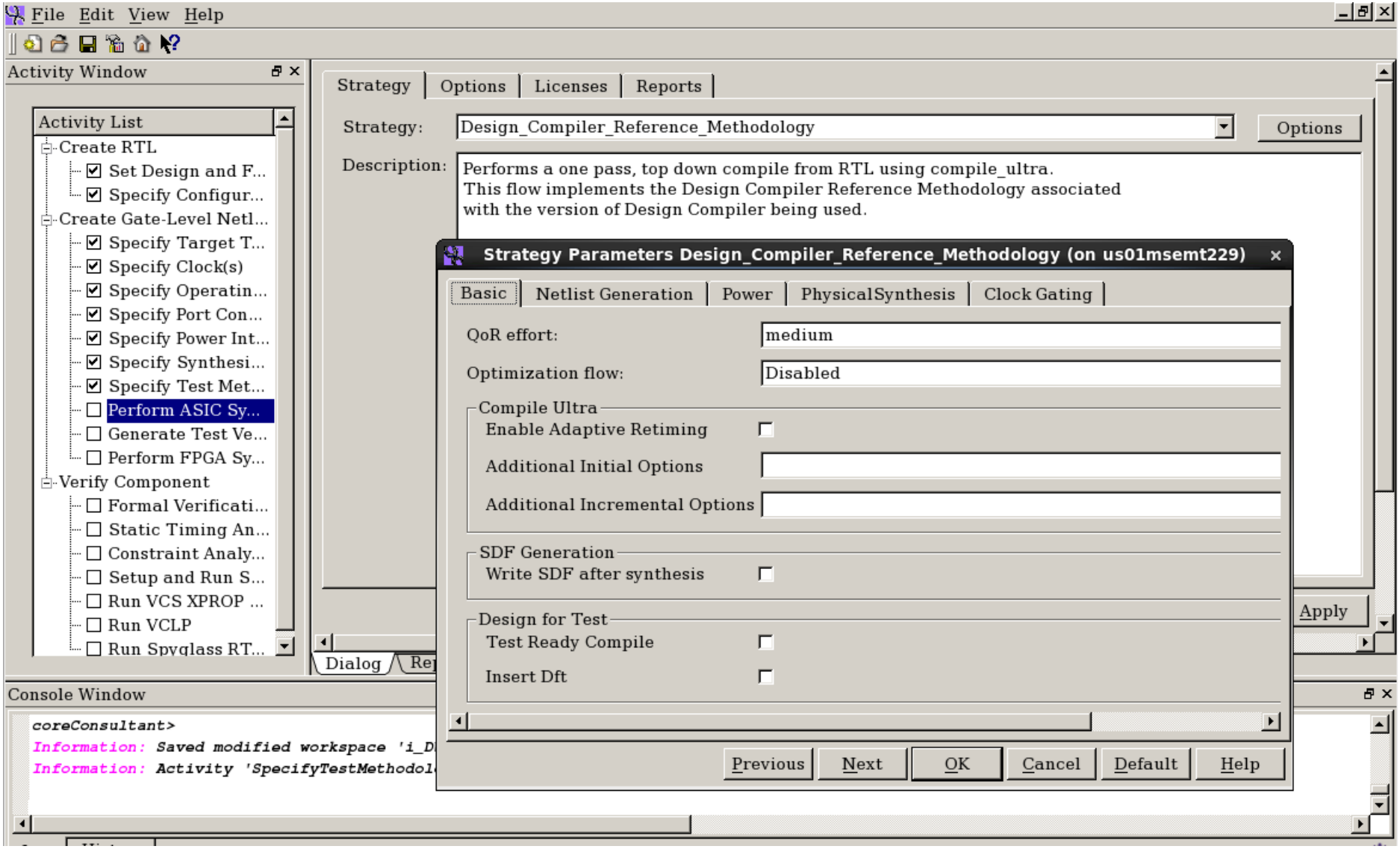


UVM based PVE

- Chapter 3 of the userguide
  - List of test cases
  - Recommendations of porting to SoC level
  - Used for basic sanity check
- PVE supports VCS simulator.
- VIPs should be downloaded first to run PVE test patterns

Synthesis

Synthesis in coreConsultant

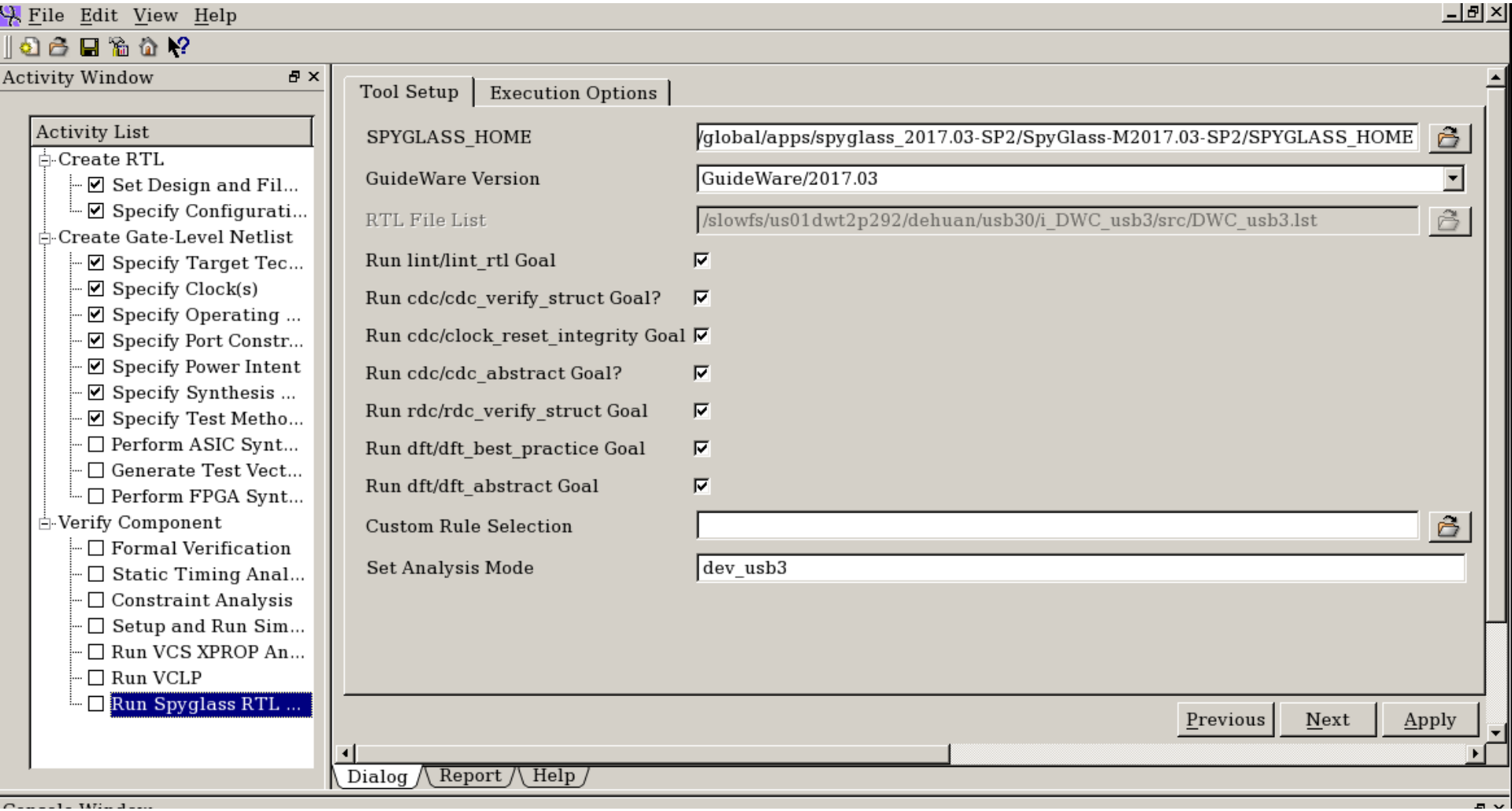


Related information

- Design Compiler 2016.12-SP4
- Section 2.2.2 of the databook
  - Clock domains information.
- Section 4.1 of the userguide
  - Synthesis steps in coreConsultant
  - Reference script in section 4.1.4
- Synthesis/CTS application note
  - https://www.synopsys.com/dw/doc.php/iip/DWC_usb3/latest/doc/DWC_usb3_Synthesis_CTS_App_Note.pdf

Spyglass

Spyglass in coreConsultant



Spyglass in coreConsultant

- Spyglass version 2017.03-SP2 is supported
- GuideWare/2017.03 is supported
  - lint_rtl
  - cdc_verify_struct
  - clock_reset_integrity
  - cdc_abstract
  - rdc_verify_struct
  - dft_best_practice
  - dft_abstract
- Analysis mode should be taken care of
- Section 2.4 of the userguide
- Integration Guide for controller+Synopsys PHY
  - https://www.synopsys.com/dw/doc.php/phy/usb3.0/appnotes/DWC_usb3_phy_cntr_integration_appnote.pdf

SYNOPSYS®
Silicon to Software™

Thank You