Lattice USB Type-C Solution

April 2015 **Reference Design RD1210**

Introduction

USB Type-C is the reversible nature of the connector and plug. Lattice Type-C solution provides feature as a Cable Detection (CD). This detects the orientation of Type-C connector. Lattice Power Delivery (PD) controls switching action of SS (Super Speed Switch) and the HS (High Speed Switch) used for the data transfer.

This design document describes USB Type-C Power Delivery (PD) and Cable Detect (CD) solutions as:

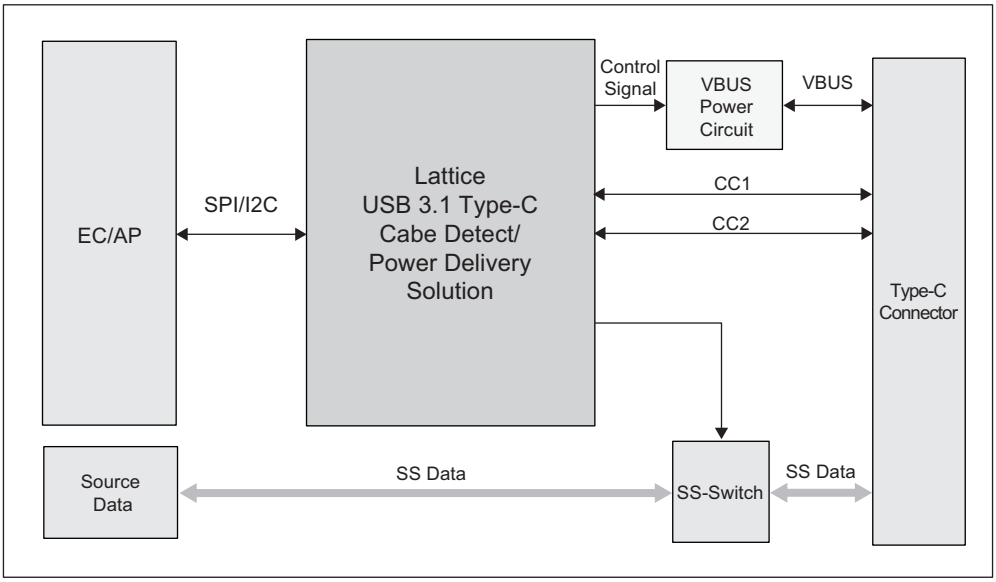
- CD/PD for charger
- CD/PD PHY for hosts/device
- CD/PD for hosts/device

Refer to DS1052, Lattice USB Type-C Solution Data Sheet and RD1209, Lattice USB Type-C Solution as a reference design.

System Block Diagram

Figure 1 shows the system block diagram of the USB Type-C Solution. Application Processor (AP) or Embedded Controller communicates Lattice USB 3.1 solution via I2C or SPI serial link. Lattice Type-C solution designed to provide configuration channel (CC1/CC2) signals to Type-C connector. CC1/CC2 lines are bidirectional. Lattice USB 3.1 solution provides control signal for VBUS power circuit and superfast switch selection signals. SS data transfer happens between external source/sink devices via SS switch. Lattice Type-C solution is capable of addressing all USB 3.1 Type-C connector requirements.

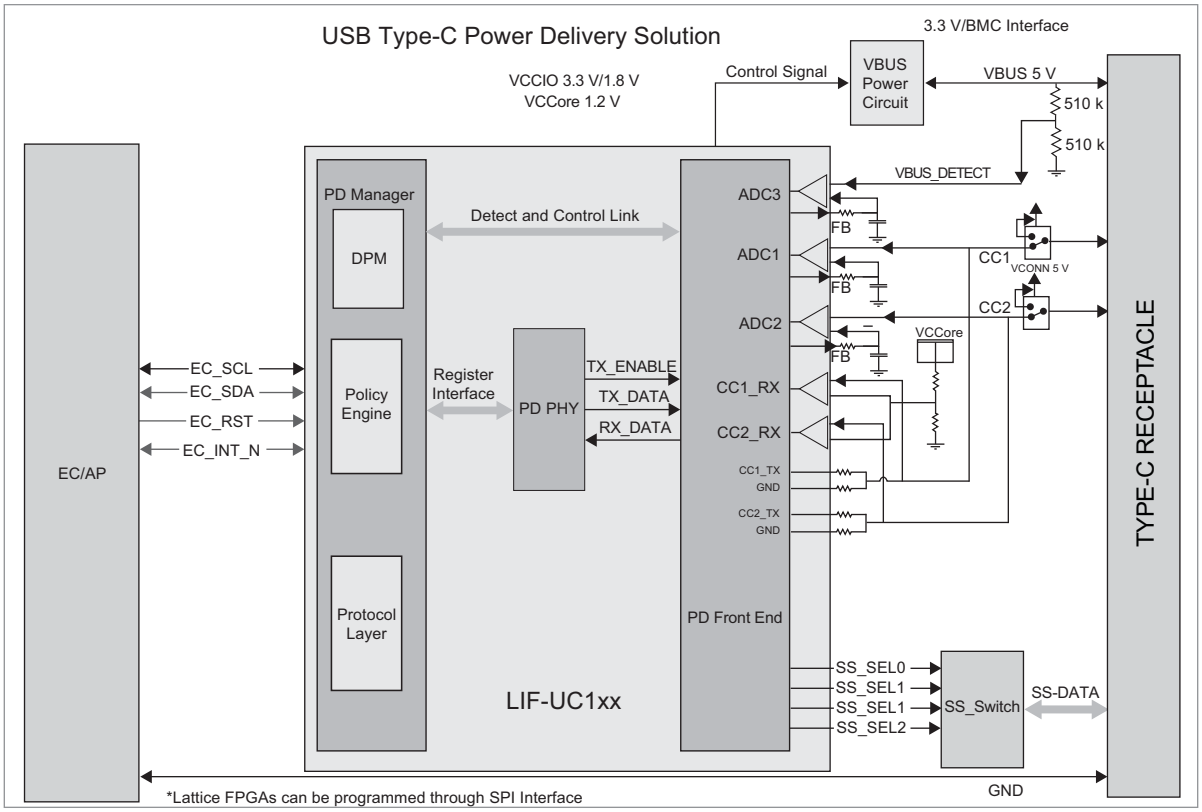
Figure 1. System Level Block Diagram



System Level Functional Block Diagram

Figure 2 shows the system level functional block diagram. It gives internal modules of Lattice USB Type-C solution as PD manager, PD PHY, and PD Frontend.

Figure 2. System Level Functional Block Diagram



Module Level Diagram of Lattice USB 3.1 Type-C Solution

Figure 3 shows detailed module level diagram of Lattice USB 3.1 Type-C solution. It consists of 3 modules.

- Device Policy Manager (DPM)
- Policy Engine
- Protocol Layer

Figure 3. Module Level Diagram

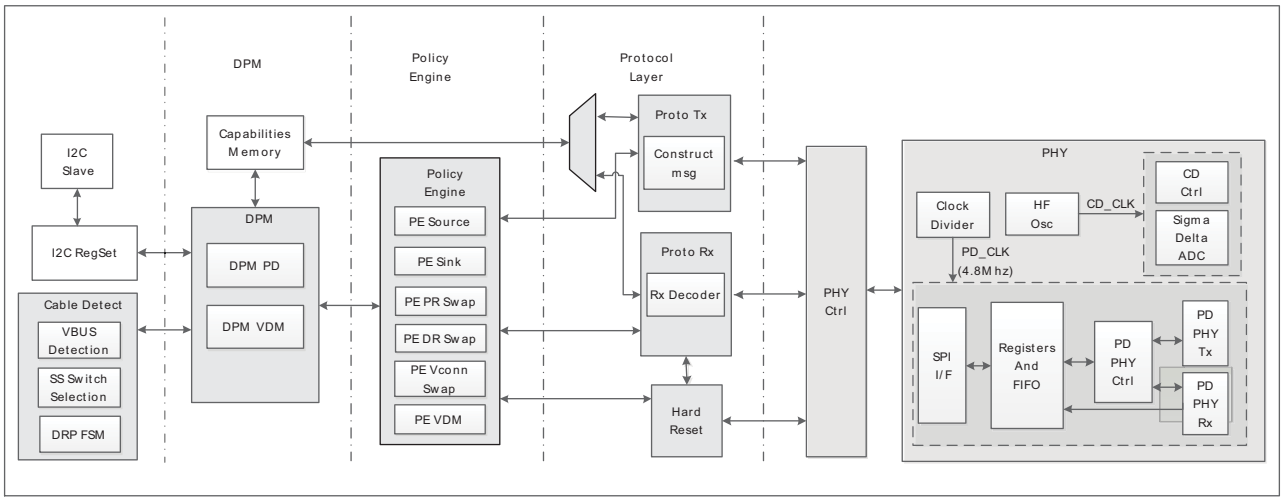
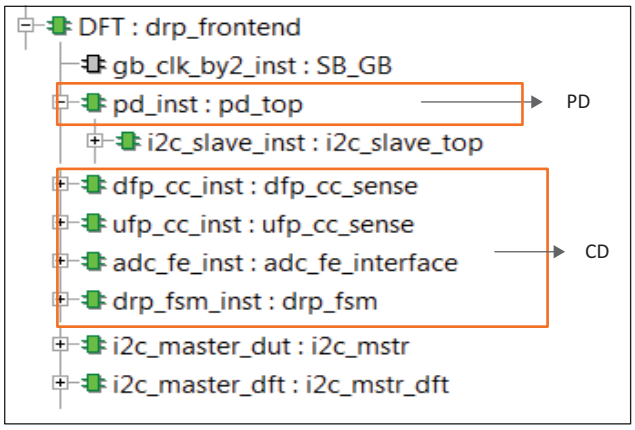


Figure 4. RTL Design Hierarchy

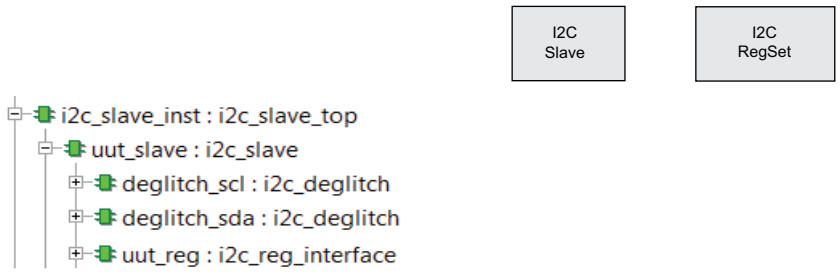


I2C Slave Controller

I2C Slave controller is the communication channel between Application Processor (AP) and Lattice USB 3.1 Type-C Power Delivery (PD) solution. Power Delivery process is interrupt driven. Any major event in the Power Delivery process generates an interrupt to the AP and AP responds with necessary actions. AP can access the following set of Registers defined to control and know the status of Power Delivery process. AP updates its own Power Capabilities by updating its own Power Capabilities Object memory.

- Control Register
- Status Register
- Interrupt Register
- Capabilities Memory

Figure 5. Design Hierarchy




```
../source/i2c/i2c_slave_top.v
      ->/i2c_reg_interface.v
      ->/i2c_slave_ctrl.v
      ->/i2c_deglitch.v
```

Cable Detect (CD)

Cable detect is an Electrical/Analog frontend. It's function as:

- To detect which CC line is connected by the cable with the port partner.
- To control the SS switch select signals.

The Type-C cable detection can be further divided into two parts:

- DFP/Source side CC vRd detection for 'Attach' and vOpen detection for 'Detach'
- UFP/Sink side presence of VBUS for 'Attach' and absence for 'Detach'

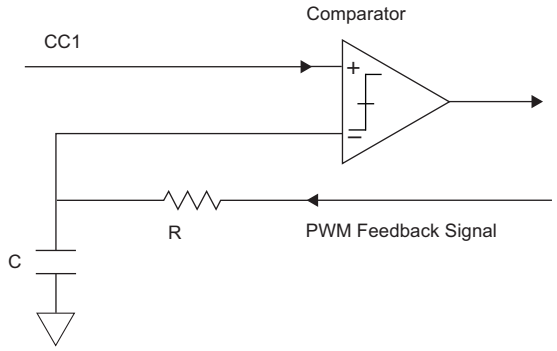
Table 1. DFP and UFP Behaviors by State

State	DFP Behavior	UFP Behavior
Nothing attached	– Sense CC pins for attach – Do not apply V_{BUS} or V_{CONN}	– Sense V_{BUS} for attach
UFP attached	– Sense CC for orientation – Sense CC for detach – Apply V_{BUS} or V_{CONN}	– Sense CC pins for orientation – Sense loss of V_{BUS} for detach
Powered cable/No UFP attached	– Sense CC for attach – Do not apply V_{BUS} or V_{CONN}	– Sense V_{BUS} for attach
Powered cable/UFP attached	– Sense CC for orientation – Sense CC for detach – Apply V_{BUS} or V_{CONN}	– Sense CC for orientation – Sense loss of V_{BUS} for detach
Debug Accessory Mode attached	– Sense CC for detach – Reconfigure for debug	– N/A
Audio Adapter Accessory Mode attached	– Sense CC for detach – Reconfigure for analog audio	– N/A

Table 2. USB Type-C DFP Connection States

CC1	CC2	State	
Open	Open	vOpen Nothing attached	N/A
Rd	Open	vRd UFP attached	①
Open	Rd		②
Open	Ra	vRa Powered cable / No UFP attached	①
Ra	Open		②
Rd	Ra	vRd + vRa Powered cable / UFP attached	①
Ra	Rd		②
Rd	Rd	vRd CC1 and CC2 Debug accessory mode attached (Appendix B)	N/A
Ra	Ra	vRa CC1 and CC2 Audio Adapter Accessory Mode attached (Appendix A)	N/A

Figure 6. Direct Topology Used for vRd and V_{BUS} Detection



The USB Type-C 1.0 specification have a wide range of voltage detection requirements based on three different types of current advertisements from the Source/DFP. The voltage detection on DFP/Source side and the UFP/Sink side are defined in the specification and for a quick reference the tables below show us the values for current advertisement of 3A by the DFP.

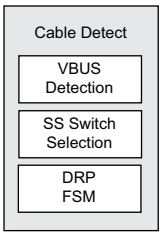
Table 3. CC Voltages on DFP Side - 3.0 A @ 5 V

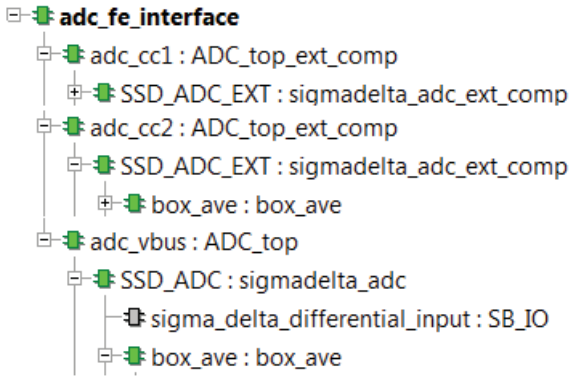
	Minimum Voltage	Maximum Voltage	Threshold
Powered cable / adapter (vRa)	0.00 V	0.75 V	0.80 V
UFP (vRd)	0.85 V	2.45 V	2.60 V
No connection (vOpen)	2.75 V		

Table 4. Voltage on UFP CC pins (Multiple DFP Current Advertisements)

Detection	Minimum Voltage	Maximum Voltage	2Threshold
vRa	–0.25 V	0.15 V	0.2 V
vRd-Connect	0.25 V	2.04 V	
vRd-USB	0.25 V	0.61 V	0.66 V
vRd-1.5	0.70 V	1.16 V	1.23 V
vRd-3.0	1.31 V	2.04 V	

Figure 7. Design Hierarchy






```
../source/adc/adc_top.v
      ->adc_fe_intfc.v
      ->box_ave.v
      ->sigmadelta_adc_ice40.v
      ->sigmadelta_adc_ext_comp.v
      ->adc_top_ext_comp.v
```

VBUS Detection

At the UFP/Sink side, the first event after cable connection is the presence of V_{BUS} on the Type-C receptacle. Thus it is mandatory for a Device/Sink/UFP to detect this voltage and enter the 'Attached UFP' state as mentioned in the DRP FSM of the USB Type-C1.0 specification.

Since the V_{BUS} detection does not have a very wide range unlike vRd hence the ADC used will have the RC network based on the 'Direct' topology. The most important consideration is the capability to handle all three voltage ratings specified by the USB PD revision 2.0. To achieve the 5 V, 12 V and 20 V capability, a resistor divider network is used on the V_{BUS} input coming from the receptacle. This is done to ensure that there is no damage to the Lattice device IO since device IO are 3.3 V compatible. The value of the two resistors used should be carefully ratioed to get a proper range from 0 V to 20 V or as per requirement.

Figure 8. Design Hierarchy


```
../source/dfp_cc_sense.v
```

SS Switch Selection

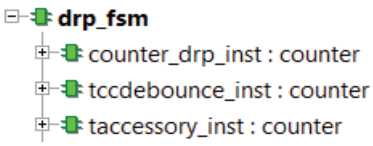
One of the most exciting feature of the USB Type-C is the reversible nature of the plug. But this new feature calls for the need of detecting the orientation which is successfully done by Configuration Channel (CC1 and CC2) vRd detection and as described in the previous section. But once this orientation is detected, the device should be able to re-arrange its Data Channels to proper alignment in order to send the data across to the Port-Partner. This is done by means of external Super-Speed Switches (SS Switch) for the SSTX and the SSRX data lanes of USB3.1.

This design provides flexible number of IOs that can be used to perform the SS switch lane selection by controlling the select lines available on the switches. The design can provide up to four IOs for the select lines.

Note:

- *Not all USB Type-C devices will need the SS switches. This requirement is only for the USB communications capable devices.*
- *The devices such as USB Chargers and the devices with Captive Cables may not need these cross bar switches since the orientation is either fixed due to captive nature or not required due to no possible USB data communication.*

Figure 9. Design Hierarchy



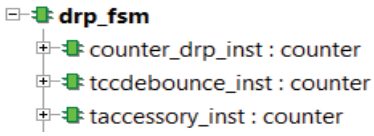
DRP FSM as Per USB Type-C 1.0

This design is implemented in order to achieve the full functionality defined in the USB Type-C1.0. Based on the specification, there are three type of ports, UFP (Device type), DFP (Host type) and the DRP (can be both UFP and a DFP).

The Lattice CD/PD controller implements a Dual Role Port such that for any other implementation like UFP only or DFP only, will just need removal of extra features but the required feature will already be present. This makes this design a superset of Type-C implementation, allowing easy customization for any subset implementation. For example, a Charger is just a DFP without any SS switch selection can be easily created from the solution by removing the Sink/UFP related modules from both the CD and the PD logic.

Further details of the selected DRP implementation can be found on page 119, Figure 4-16 of the USB Type-C 1.0 specification.

Figure 10. Design Hierarchy



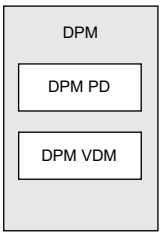

```
../source/drp_fsm.v
      ->counter_generic.v
```

Device Policy Manager (DPM)

The Device Policy Manager consists of two modules:

- DPM for Power Delivery
- DPM for Vendor defined messages

Figure 11. DPM



Device Policy Manager (DPM) for Power Delivery (PD)

The Device Policy Manager is responsible for applying local policies based on the fixed I^2C Register Set or inputs from the AP, to the Power Delivery negotiation process.

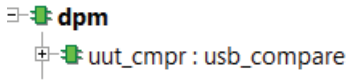
Role of DPM in sink operation:

- Compare the received PDOs from the port partner and make a valid request.
- Handle Power Role SWAP, Data Role SWAP and VConn SWAP requests.
- Inform the AP about different Power transitions.
- Decide on which SOP* to use for the current communication.

Role of DPM in source operation:

- Evaluate the request received from the port partner and make a decision on whether to send an Accept message or a Reject message.
- Handle Power Role SWAP, Data Role SWAP and VConn SWAP requests.
- Inform the AP about different Power transitions.
- Decide on which SOP* to use for the current communication.

Figure 12. Design Hierarchy



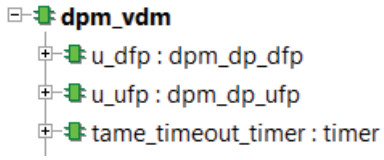

```
../source/pd_manager/dpm.v
      -> dpm_usb_compare.v
      ->vdm_reg_ctrl.v
      ->vdm_reg_set.v
      ->ram_256x32_vdm.v
```

Device Policy Manager (DPM) for Vendor Defined Message (VDM)

Device Policy Manager for Vendor Defined Messages (VDM) is responsible for implementing the VDM message flow as defined by Display port Alternate Mode. The message sequence identifies the attached UFP device and if the attached device supports Display Port Alternate Mode, then necessary actions to Enter Display Mode and configures the SS Switch accordingly, else AP gets information to enter Billboard Mode.

For VDM Message Flows for Display Port, refer yo section 5.3 of the VESA Proposed Display Port Alternate Mode on USB Type-C Standard specification revision 1.0.

Figure 13. Design Hierarchy

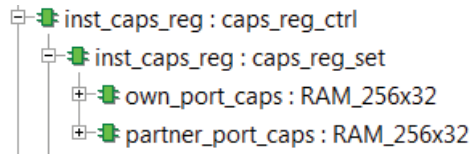



```
../source/pd_manager/ dpm_vdm.v
      ->dpm_dp_dfp.v
      ->dpm_dp_ufp.v
      ->pd_to_hpd.v
```

Capabilities Memory

This is a pre-initialized memory which has the default Port Source and Sink Power capabilities (PDO) defined. In addition, it has space to hold the received port partners Source and Sink Power capabilities (PDO) for the AP to access.

Figure 14. Design Hierarchy



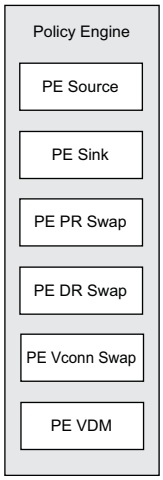

```
../source/pd_manager/caps_reg_ctrl.v
      ->caps_reg_set.v
```

Policy Engine (PE)

The Policy Engine interprets the Device Policy Manager's input in order to implement Policy for a given Port and directs the Protocol Layer to send appropriate messages. The Policy Engine consists of six modules,

- PE Source
- PE Sink
- PE PR Swap
- PE DR Swap
- PE VConn Swap
- PE VDM

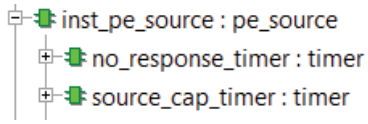
Figure 15. Policy Engine



PE Source

In a Dual Role Port (DRP), if the negotiated role by the Cable Detection module is Source, then this module will be active, else will be disabled.

Figure 16. Design Hierarchy




```
../source/pd_manager/pe_source.v
      -> timer.v
```

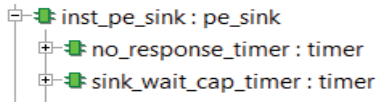
Functions:

- To handle the flow of different Power Delivery messages that has to be transmitted and received during the negotiation process.
- To handle the Soft Reset and Hard Reset messages to reset the protocol layer and power levels respectively.

For implementation details, refer to Figure 8-39 in page 387 and Figure 8-41 in page 396 of the USB PD specification revision 2.0.

PE Sink
In a Dual Role Port (DRP), if the negotiated role by the Cable Detection module is Sink, then this module will be active, else will be disabled.

Figure 17. Design Hierarchy




```
../source/pd_manager/pe_sink.v
      -> timer.v
```

Functions:

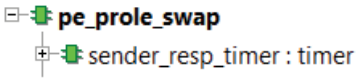
- To handle the flow of different Power Delivery messages that has to be transmitted and received during the negotiation process.
- To handle the Soft Reset and Hard Reset messages to reset the protocol layer and power levels respectively.

For implementation details, refer to Figure 8-40 in page 392 and Figure 8-42 in page 398 of the USB PD specification revision 2.0.

PE Power Role (PR) Swap
In a Dual Role Port (DRP), this module is responsible for handling Port Role Swap requests and maintains the new and original Port Roles.

For implementation details, refer to Figure 8-51 in page 428 and Figure 8-52 in page 431 of the USB PD specification revision 2.0.

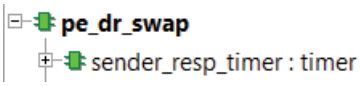
Figure 18. Design Hierarchy




```
../source/pd_manager/pe_prole_swap.v
      -> timer.v
```

PE Data Role (DR) Swap
In a Dual Role Port (DRP), this module is responsible for handling Data Role Swap requests and maintains the new and original Data Roles.

Figure 19. Design Hierarchy

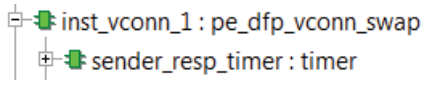



```
../source/pd_manager/pe_dr_swap.v
      -> timer.v
```

For implementation details, refer to Figure 8-51 in page 422 and Figure 8-50 in page 425 of the USB PD specification version 2.0.

PE VConn Swap
In a Dual Role Port (DRP), this module is responsible for handling VConn Swap requests.

Figure 20. Design Hierarchy

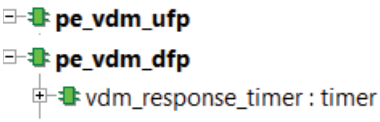



```
../source/pd_manager/e_dfp_vconn_swap.v
      -> timer.v
```

For implementation details, refer to Figure 8-57 in page 437 and Figure 8-58 in page 439 of the USB PD specification revision 2.0.

PE VDM
Policy Engine for VDM is responsible for handling VDM message transfers without interrupting PD messages. It transfers initiator (mostly DFP) VDM messages as instructed by DPM VDM and informs the response from the responder (UFP mostly) back to DPM VDM.

Figure 21. Design Hierarchy




```
../source/pd_manager/pe_vdm_dfp.v
      -> pe_vdm_ufp.v
```

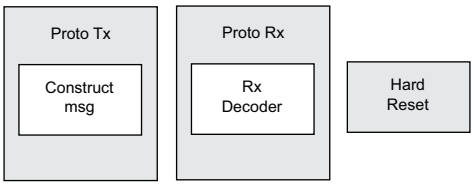
For implementation details, refer to Figure 8-61 and Figure 8-68 of the USB PD specification revision 2.0.

Protocol Layer

The Protocol Layer is divided into three modules:

- Proto Tx
- Proto Rx
- Hard reset

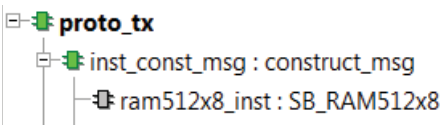
Figure 22. Protocol Layer



Proto Tx
Proto Tx module is responsible for transmission of a Power Delivery packet. It implements the necessary retry counters and message ID handlers as shown in Figure 6-19 in page 195 of USB PD specification revision 2.0

The Proto Tx module includes a construct message module which is responsible for forming the necessary Power Delivery packet based on the input from Policy Engine.

Figure 23. Design Hierarchy



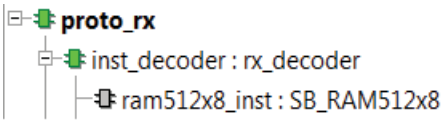

```
../source/pd_manager/proto_tx.v
      ->construct_msg.v
```

Proto Rx

Proto Rx module is responsible for reception of a Power Delivery Packet. Based on the stored message ID, the Proto Rx module decides on whether to pass the received packet or to ignore it as shown in figure 6-20 in page 198 of USB PD specification revision 2.0

The Proto Rx module includes a received packet decoder module that decodes the received packet on a successful reception (if the packet need not be ignored), and passes the received Power Delivery message type to the Policy Engine.

Figure 24. Design Hierarchy



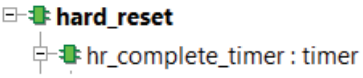

```
../source/pd_manager/proto_rx.v
        ->rx_decoder.v
```

Hard Reset

Hard Reset module handles the hard reset condition if received by the port partner or if initiated by own port. On a hard reset condition, this module makes sure that all power levels and protocol layer is reset to default before any further PD communication.

For implementation details, refer to Figure 6-21 in page 200 of USB PD specification revision 2.0.

Figure 25. Design Hierarchy

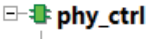



```
../source/pd_manager/hard_reset.v
      ->timer.v
```

PHY Control

The PHY control module is used as a communication interface between the Protocol Layer and the PHY.

Figure 26. Design Hierarchy




```
../source/pd_manager/phy_ctrl.v
```

PHY

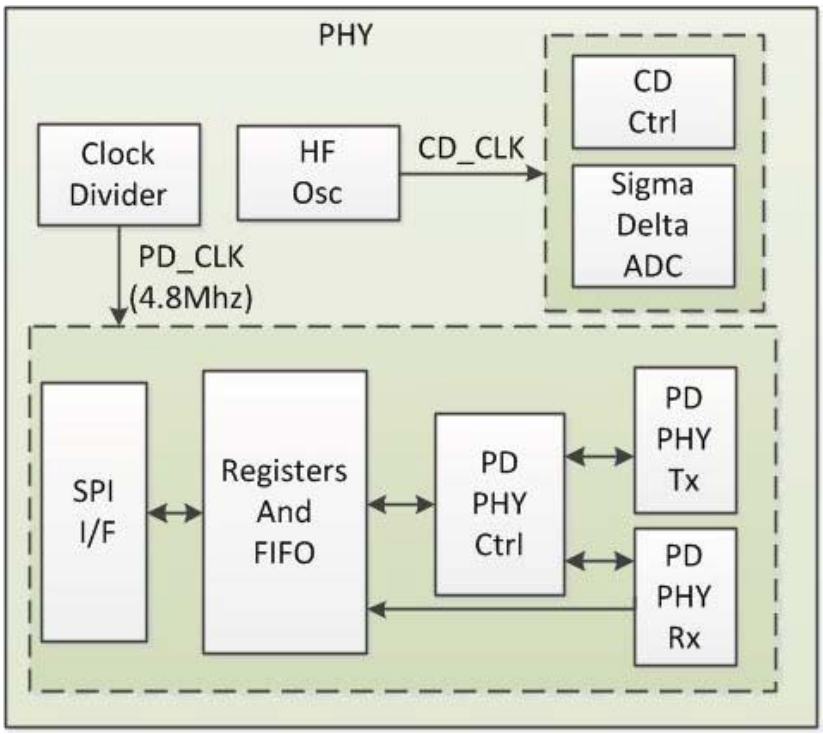
This module has three internal modules as:

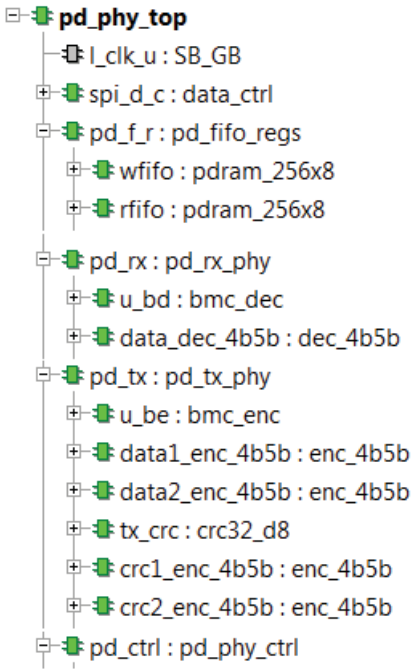
- Register Interface
- Tx with 4B5B and BMC encoding
- Rx with BMC decoding, start/stop detection and 4B5B decoder

The PD raw data size as 300K. After BMC encoding data size as 600K. By referring to the CC-line eye diagram in the PD specification, use 4.8 MHz clock to get 16 samples per nominal bit-length and 8 samples per nominal half bit length.

PHY module generates Good CRC message automatically based on the received message ID and the stored message ID from Protocol layer.

Figure 27. Design Hierarchy






```
../source/pd_manager/pd_phy_top.v
      ->data_ctrl.v
      ->pd_fifo_regs.v
      ->pd_rx_phy.v
      ->pd_tx_phy.v
      ->pd_phy_ctrl.v
      ->bmc_dec.v
      ->bmc_enc.v
      ->dec_4b5b.v
      ->enc_4b5b.v
      ->pd_crc32_d8.v
      ->pd_defines.v
      ->pdram_256x8.v
```

SPI Interface
Application processor writes and reads internal registers of module PHY via the SPI interface.

Clock Divider
Input clock to Clock Divider module has been provided by the A. It is 4.8 MHz Clock or Even Multiples of 4.8 MHz such as 9.6 MHz or 19.2 MHz etc. There is the provision of an internal oscillator if the customer does not want to use an external clock source.

Clock Divider module provides input clock signal as PD_CLK (4.8 MHz) to the modules:

- SPI I/F
- Registers and FIFO
- PD PHY Ctrl
- PD PHY Tx
- PD PHY Rx

PD_CLK can be disabled by the AP when Power Delivery communication is not being used.

HF_Osc
High Frequency Oscillator provides clock signal to Cable Detect (CD) Ctrl and Sigma delta ADC module.

CD_Ctrl and Sigma Delta ADC
Cable Detection is a power-optimized and always-on function that constantly monitors the status on both CClines and detects a cable insertion, cable orientation, cable removal and Source Port's advertisement for available current. This module consists of a state machine for Cable Detection process.

Registers and FIFO
The AP is responsible to initiate the PD message transmission by writing the associated registers via SPI bus. The PD PHY performs the PD packet transfer based on the registers set by the AP. And it provides the PD transfer results to the AP through the interrupt requests and the internal registers. Also it detects and receives the packet from the PD front end. Then it interrupts the AP with the receiving results. If the packet is error-free, the AP will read it.

The registers first thirteen registers are used to control the PD PHY and CD operations while the other two registers are used for the write/read buffers.

PD PHY Ctrl
The PD PHY control module provides Control path signals to the PD PHY Tx and PD PHY Rx.

PD PHY Tx
This module consists of BMC encoder and transmitter. The PD raw data is of size as 300K. After BMC encoding data size as 600K. By referring to the CC-line eye diagram in the PD specification, it has been used 4.8 MHz clock to get 16 samples per nominal bit-length and 8 samples per nominal half bit-length.

PD PHY Rx
This module consists of BMC decoder and receiver.

Module Level Diagram of Lattice USB 3.1 Type-C Solution with Mico

Figure 28 shows detailed module level diagram of Lattice USB 3.1 Type-C solution with Mico. All the modules of Figure 28 are similar to the Figure 3, except DPM module.

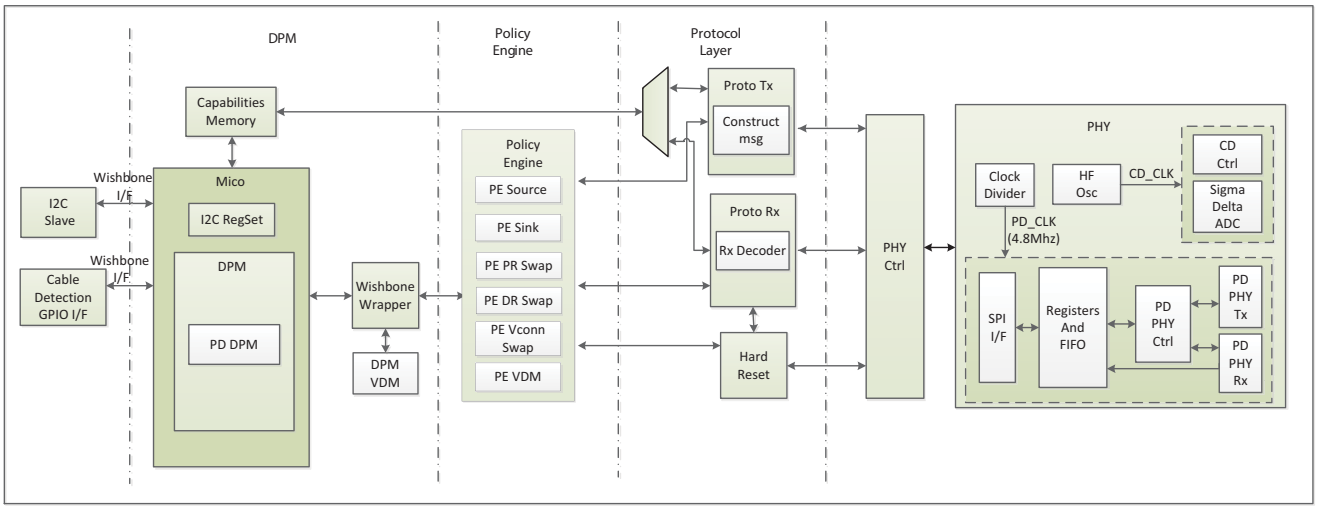
DPM module consists of following sub modules:

- Capabilities Memory
- Mico.
- Wishbone Wrapper
- DPM VDM

Mico module of Figure 6 consists of following sub modules:

- I2C Register Set
- DPM for PD which includes:
  - — Power profile handling
  - — PR SWAP, DR SWAP and VConn SWAP handling

Figure 28. Module Level Diagram of Lattice USB 3.1 Type C Solution with Mico



Mico8

The LatticeMico8 is an eight bit microcontroller.

Features:

- Innovative Open IP Core License
- Efficient Architecture and Broad Feature Set
  - — 18-bit Wide Instructions
  - — Configurable 16 or 32 General Purpose Registers
  - — Configurable Instruction Memory (PROM)
    - — Internal or external through Wishbone Interface
    - — Configurable to accommodate 256, 512, 1K, 2K or 4K instructions
  - — Scratchpad Memory
    - — Internal or external through Wishbone Interface
    - — Configurable up to 4 Gigabytes using paging (256 bytes per page)
  - — Minimum two cycles per instruction
  - — Configurable 8, 16, or 32-deep call stack
  - — Support for up to 8 external interrupts
  - — Integrated hardware loader to optionally initialize PROM and Scratch pad for an external nonvolatile memory
- Wishbone Peripheral Components:
  - — GPIO, UART, DMA Controller, SPI Flash Controller, MachXO2 EFB (I2C, SPI, Timer)
- Consumes ~800 LUTs including the necessary wishbone peripherals.

References

- USB Type-C Specification 1.0.pdf
- USB PD Specification R2.0
- DS1052, Lattice USB Type-C Solution Data Sheet
- RD1209, Lattice USB Type-C Solution

Technical Support Assistance

e-mail: techsupport@latticesemi.com

Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
April 2015	1.0	Initial release.