PWE-CES DEMO README FOR WDDS 4.3.2

MARCH 2013

ISSUE 1

Legal Information

Copyright

Disclaimer

None of the information contained in this document constitutes an express or implied warranty by PMC-Sierra, Inc. as to the sufficiency, fitness or suitability for a particular purpose of any such information or the fitness, or suitability for a particular purpose, merchantability, performance, compatibility with other parts or systems, of any of the products of PMC-Sierra, Inc., or any portion thereof, referred to in this document. PMC-Sierra, Inc. expressly disclaims all representations and warranties of any kind regarding the contents or use of the information, including, but not limited to, express and implied warranties of accuracy, completeness, merchantability, fitness for a particular use, or non-infringement.

In no event will PMC-Sierra, Inc. be liable for any direct, indirect, special, incidental or consequential damages, including, but not limited to, lost profits, lost business or lost data resulting from any use of or reliance upon the information, whether or not PMC-Sierra, Inc. has been advised of the possibility of such damage.

Trademarks

For a complete list of PMC-Sierra's trademarks and registered trademarks, visit: http://www.pmc-sierra.com/legal/

Other product and company names mentioned herein may be the trademarks of their respective owners.

Revision History

Issue	Date	Details of Change
1	March 2013	This document applies to the 4.3.2 release.

Contacting PMC-Sierra

PMC-Sierra
8555 Baxter Place
Burnaby, BC
Canada V5A 4V7
Tel: +1 (604) 415-6000
Fax: +1 (604) 415-6200

PMC-Sierra, WIN Div.
6850 Austin Center Blvd.
Suite 215
Austin, TX 78731
USA
Tel: +1 (512) 345-3808

PMC-Sierra Israel Ltd.
7 Arie Shenkar St.
Gav-Yam Bldg 2
Herzliya 46120
Israel
Tel: +972 (9) 743-9998

PMC-Sierra, WIN Div.
International House, Stanley Blvd.
Hamilton Int. Technology Park
Blantyre, Glasgow. G72 OBN.
United Kingdom
Tel: +44 (0) 1698-404889

Document Information:	document@pmc-sierra.com
Corporate Information:	info@pmc-sierra.com
Technical Support:	apps@pmc-sierra.com
Web Site:	http://www.pmc-sierra.com

Before You Start Read This First

Important Safety Instructions

Electrical Safety Hazards

- The power cord must be disconnected before servicing equipment.
- It is best to connect the DC power connection to the board first, and then connect the power supply to the power source.

Laser Safety Hazards

- Avoid direct eye exposure

WARNING: Because invisible radiation may be emitted from the aperture of the port when no fiber optic cable is connected, avoid exposure to radiation and do not stare into open apertures

- Never assume laser power is turned off or that the fiber is disconnected at the other end.
- The fiber optics ports must never be exposed during operation. Place a protective cap over any radiating optical fiber connector.

Electrostatic Discharge (ESD) Precautions

- Exercise proper Electrostatic Discharge precautions when handling boards.
- Use an ESD wrist strap connected to the ground when handling the WDS board. Failure to do so may result in electrostatic discharge (ESD) damage to electronic components.
- When not in use, SFPs should be stored in protective packaging.

Symbols Used in this Manual

WARNING: Indicates information on how to avoid personal injury.

LASER WARNING: Indicates information on how to avoid personal injury. All personnel handling equipment must be aware that laser radiation is invisible. Therefore, although protective devices generally prevent direct exposure to the beam, personnel must strictly observe the applicable safety precautions and, in particular, must avoid staring into optical connectors, either directly or using optical instruments.

CAUTION: Indicates information on how to avoid damage to the equipment or to avoid possible service disruption.

ESD: Indicates information on how to avoid discharge of static electricity and subsequent damage to the unit.

NOTE: Indicates useful information.

Programming **suggestions** are denoted with a “light bulb” in the left margin.

ContentsImportant Safety Instructions 3
Electrical Safety Hazards 3
Laser Safety Hazards 3
Electrostatic Discharge (ESD) Precautions 3
Symbols Used in this Manual 4
Acronyms and Terms 7

Chapter 1. Introduction 8
1.1. PWE3-CES Demo Overview 8
1.2. Basic Physical Setups 9
1.3. Master – Slave Setup 10
1.4. Test Setup 13

Chapter 2. Hardware Setup 14
2.1. Required Hardware 14
2.1.1. Evaluation Boards 15
2.1.1.1 Pins Configuration and Clock Connectivity 18
2.1.1.2 Clock Connectivity – WP2 18
2.1.1.3 Clock Connectivity – WP3 19
2.1.2. UFE3 Card 20
2.1.3. UFE4 Cards 21
2.1.3.1 UFE412 21
2.1.3.2 UFE448 23
2.1.4. E1/T1 COMET Card 24
2.1.5. PC 25
2.1.6. Network Emulator 26

Chapter 3. Software Setup 27
3.1. Required Software 27
3.2. Software Setup 27
3.2.1. TFTPD32 27
3.3. Demo Files 28
3.4. Demo Configuration 30
3.5. Demo Compilation 31
3.6. Demo Menu Usage 33
3.7. Demo 33
3.7.1. Load Demo 33
3.7.2. UFE Events 34
3.7.3. UFE3 Junction Temperature 35
3.7.4. Statistics 36
3.7.4.1 Global Statistics 36
3.7.4.2 Detailed Data Path Statistics (UFE) 38
3.7.4.3 Detailed Clock Recovery Statistics 44
3.7.5. Clock Recovery and General System Displays 49
3.7.5.1 Clock Recovery 49
3.7.5.2 System Display 51
3.7.6. UFE4 OC3/12 –Adaptive / Differential CR Mode 56
3.7.6.1 WDS3 – UFE412 4xOC3/12 –Adaptive / Differential CR Setup. 57
3.7.6.2 WDS3 – UFE448 4xOC12 –Adaptive / Differential CR Setup. 58
3.7.7. UFE3 OC3/12 –SBI Adaptive or Regular Differential CR Mode .. 59
3.7.7.1 WDS3 - OC3/12 –SBI Adaptive or Regular Dif. CR Setup. 61
3.7.7.2 WDS2 - OC3/12 –SBI Adaptive or Regular Dif. CR Setup. 61

3.7.8. UFE3 OC3/12 –SBI Extended Differential CR Mode 63
3.7.9. UFE3 OC3/12 –SBI Adaptive Dif. and Extended Dif.CR Mode ... 63
3.7.10. UFE3 E1/T1 –CAD Adaptive/Dif.or Extended Dif.CR Modes 64
3.7.10.1 UFE3 E1/T1 –CAD CR Setup 64
3.7.11. TDI Adaptive/Differential CR Modes 64
3.7.12. Two Boards Setup 65
3.8. Appendix A – Example Setting 66
3.9. Appendix B – Troubleshooting 76

Acronyms and Terms

PWE3	Pseudo-wire Emulation Edge-to-Edge
UFE	Universal Front End
CR	Clock Recovery
DCO	Digital Control Oscillator
CAD	Clock and Data
TDI	Time Division Multiplexing Interface
SBI	Scalable Bandwidth Interconnect
CD	Clock Domain
ACR	Adaptive CR
DCR	Differential CR
CRS	Clock Recovery System

Chapter 1. Introduction

1.1. PWE3-CES Demo Overview

The PWE3 CES over Packet demo demonstrates the Winpath implementation of the CES over Packet and SAT over Packet protocols using the Winpath2/3 Development System (WDS2/3).This file includes various demo menus that used in order to initialize the CES over Packet system, initialize Winpath ports, devices and Pseudo Wires. The file provides the command line interface that uses different utility functions presented in the wti_cesopsn_util.c file. The demo can be compiled to run over UFE interface or over TDI interface. The WTI_CESOP_TDI define within this file will specify the port interface for the demo. The demo includes many predefined scenarios which can be selected from the main Menu.

The demo can be compiled to run over Winmon operating system or over Linux. The compile command line will specify which operating system being used (see section 3.5 Demo Compilation).

The PWE3 CES demo also includes the PWE3-CES Clock Recovery (CR) implementation. CR can run over TDI, UFE3-SDH (SBI), and UFE3-PDH (CAD). The CR includes adaptive/regular differential CR methods and extended differential method (also known as 336CD).

Part of the predefined test scenarios can run with at least one of the CR modes.

The CR tests are divided in to 3 groups:

1. UFE3-SONET/SDH (SBI) - includes support for adaptive/differential and extended differential CR and starts with the 1xx_SBI prefix.
2. UFE3-PDH (CAD) - includes support for adaptive/differential and extended differential CR and starts with the 2xx_CAD prefix.
3. UFE4-SONET/SDH - includes support for adaptive/differential CR and starts with the CLI_F_CR_1xx prefix.
4. TDI - includes support for adaptive/differential CR and starts with the 3xx_TDI prefix.

Note: few of the xxx tests above can run without clock recovery and can be used for data path testing.

This application example does NOT support the combined UFE+TDI functionality in which both TDM and UFE interfaces work in parallel though this functionality is supported by the WDDI.

1.2. Basic Physical Setups

Basic setup includes one WDS board; this WDS can perform both TDM2PSN and PSN2TDM directions. The PSN is replaced by loopback cable. A PC is used to load the program; once loaded it can run freely without any control.

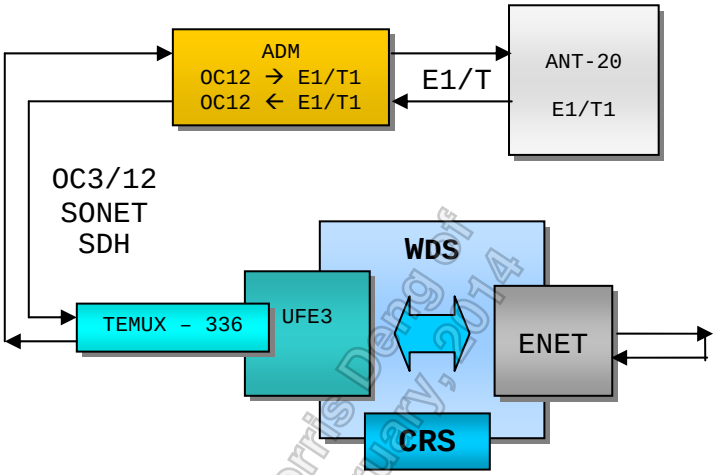


Figure 1: UFE3 OC3/12 - Basic Setup Diagram

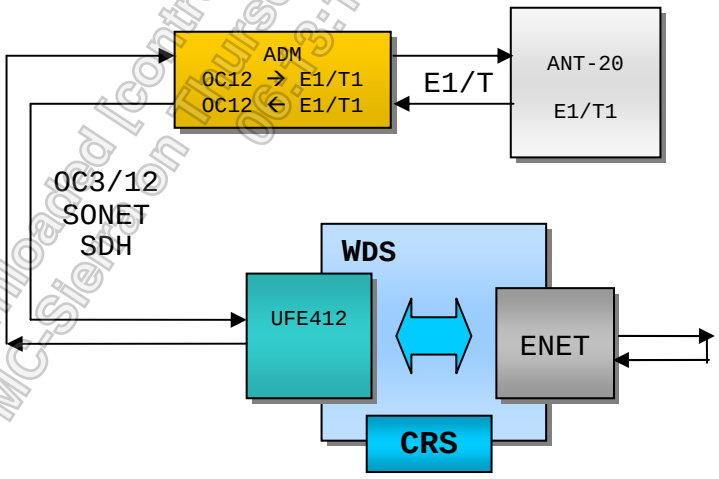


Figure 2: UFE4 OC3/12 - Basic Setup Diagram

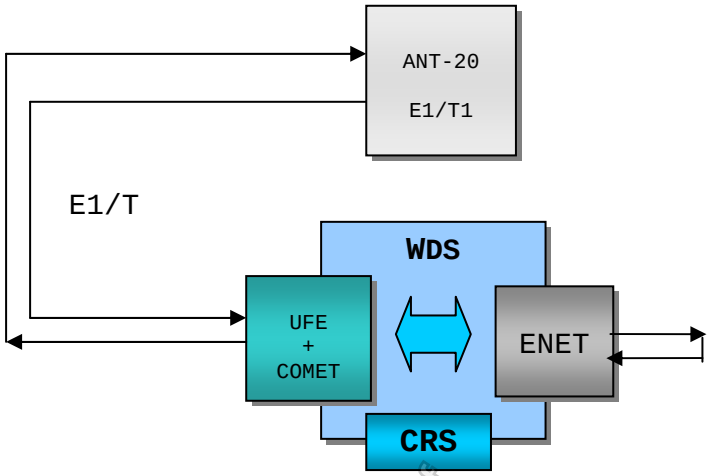


Figure 3: UFE E1/T1 CAD - Basic Setup Diagram

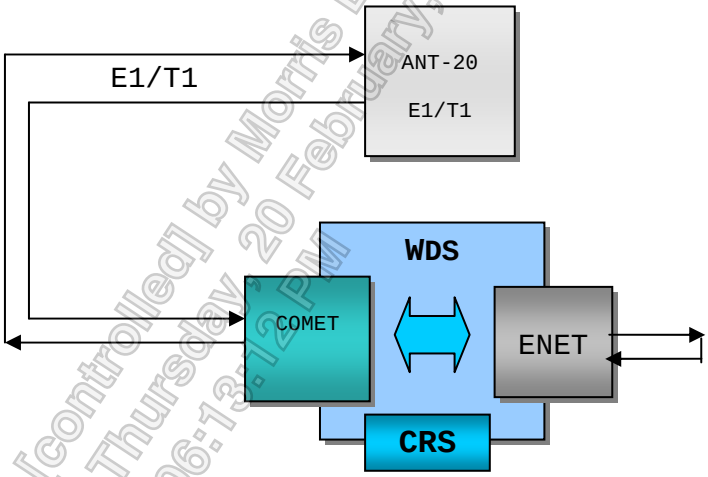


Figure 4: TDI E1/T1 - Basic Setup Diagram

1.3. Master – Slave Setup

This setup includes two WDS board; the first WDS perform the TDM2PSN (master) and second WDS perform thePSN2TDM (slave) direction. A PC is used to load the program; once loaded it can run freely without any control.

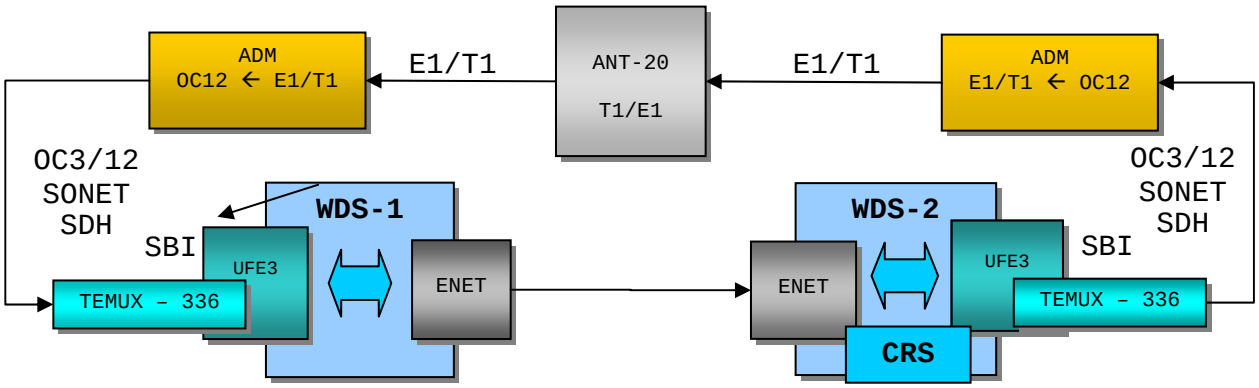


Figure 5: : UFE3 OC3/12 - Master -Slave Setup Diagram

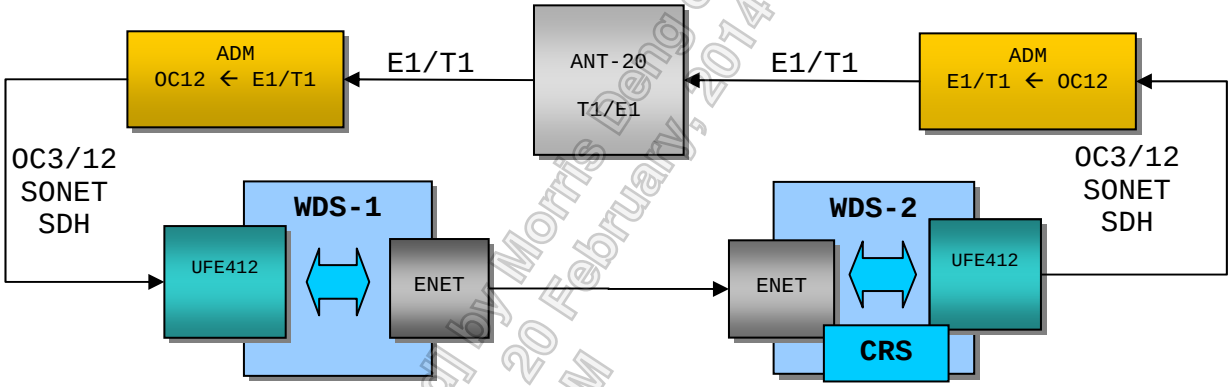


Figure 6: : UFE4 OC3/12 - Master -Slave Setup Diagram

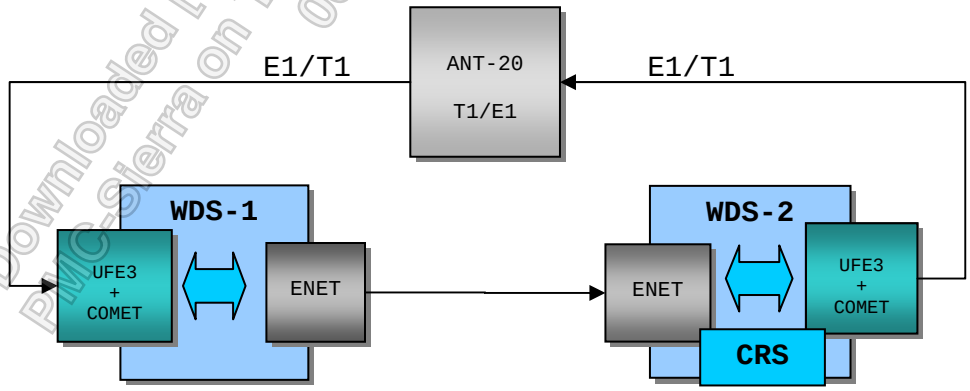


Figure 7: UFE3 E1/T1 CAD - Master -Slave Setup Diagram

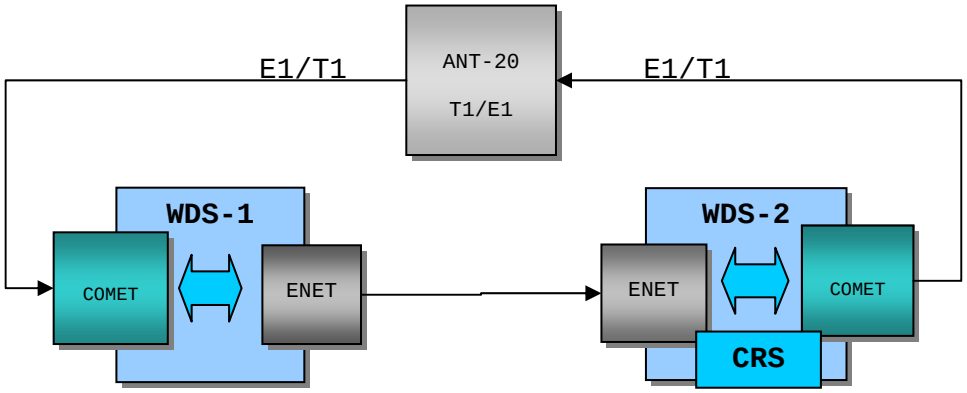


Figure 8: TDI E1/T1 - Master -Slave Setup Diagram

1.4. Test Setup

The test setup may include performance measurement and analyzer tools for the data path and clock recovery performance testing such as ANT-20.

In order to test under conditions similar to those in the real world such us G.8261, a network emulator can be connected between the two boards.

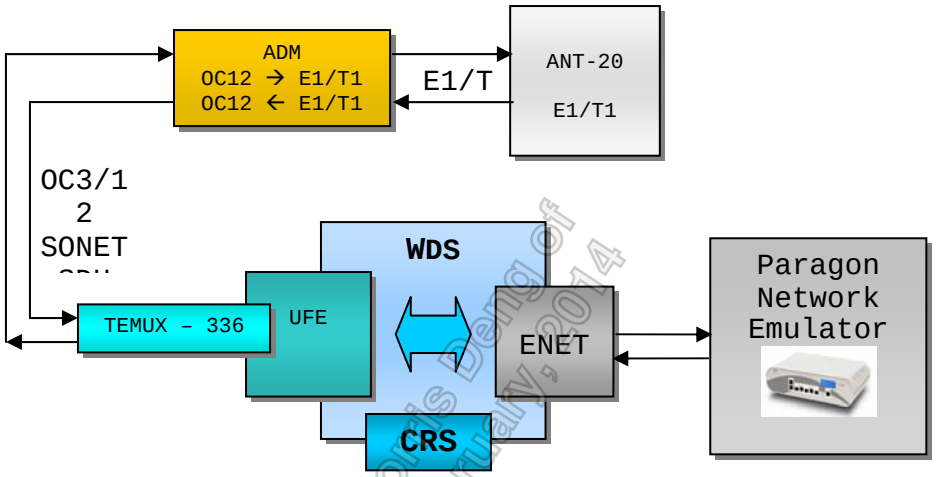


Figure 9: Test Setup Example Diagram – One WDS

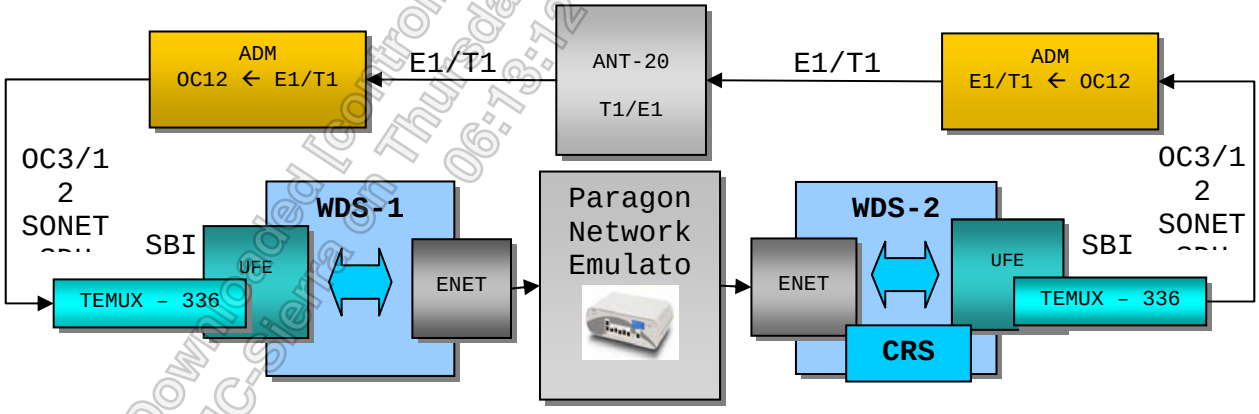


Figure 10: Test Setup Example Diagram – two WDS

Similar test setup can be used in UFE4, UFE3-CAD and TDI mode.

Chapter 2. Hardware Setup

2.1. Required Hardware

The PWE3-CES application demo requires:

- One/two WDS boards
- COMET /UFE cards (according to the desired configuration).
- a PC
- cables (depending on the setup, different cables may be needed)
  - Ethernet Cables (optical or copper)
  - Com cables
  - SMA connectors
- Scope (optional)
- Network emulator (optional, to check performance under PDV)
- PDH/SDH Generator/Analyzer – ANT-20 (optional)

2.1.1. Evaluation Boards

In a simple setup, one WDS evaluation board can be used as master and slave. In a more complex setup two WDS evaluation boards can be used. One will serve as master and the second as slave.

The control options for these boards are:

- Each WDS board can be controlled by connecting it directly to a PC via its RS232 (COM) port. In such cases, two RS232 cables will be needed.

A power supply is needed for each board.

There is an FPGA on the WDS3 board which provides the clocks routing functionality (among other functionalities).

More information about the FPGA programming and functionalities can be found in the WDS2/3 user guide.

Please note that in order to use the CR functionality on the WDS3 board, the FPGA version must be 6.a and above.

The FPGA version is displayed when the WDS3 is reset (refer to the picture below)

More information is available in the following documentation:

1. WDS2/3 user guide.

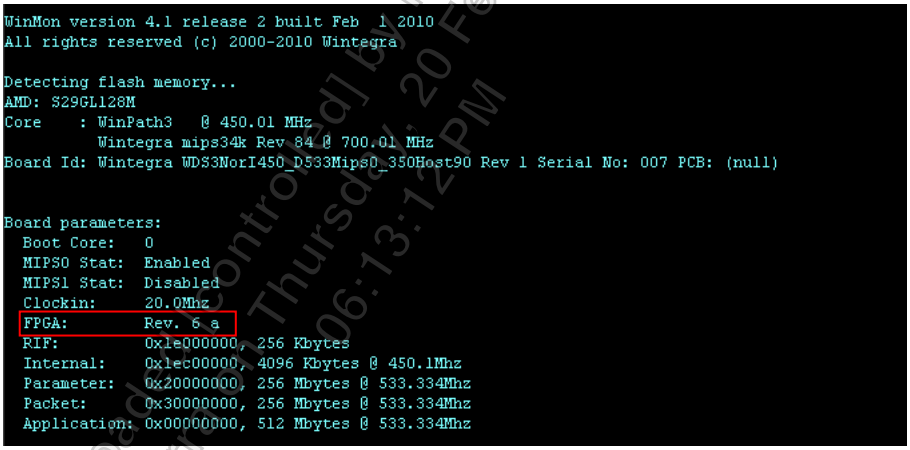


Figure 11: WDS3 FPGA Version Display

Figure 12: WDS2

Figure 13: WDS3 (version 2)

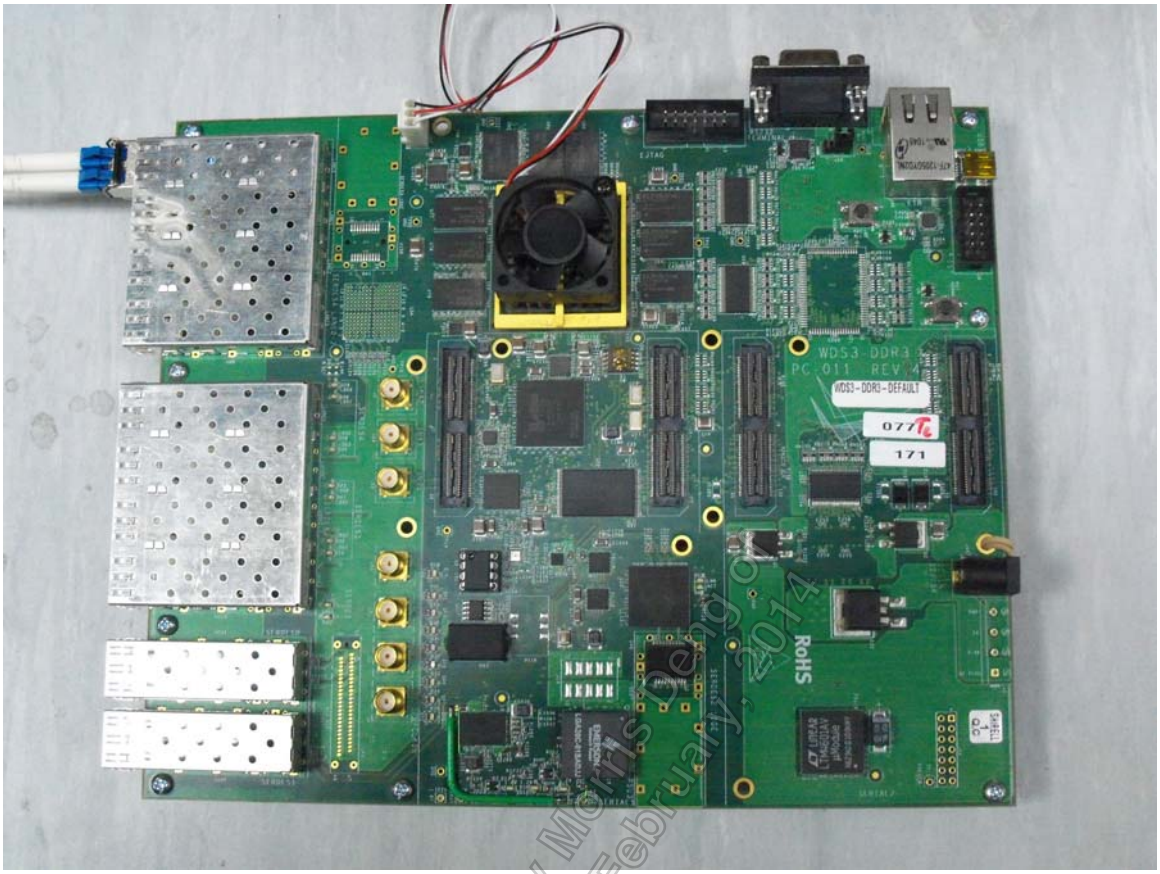


Figure 14: WP3- revB (version 4)

2.1.1.1 Pins Configuration and Clock Connectivity

2.1.1.2 Clock Connectivity – WP2

In TDI mode the DCO reference clock is taken from the WP2 internal clock (in case the WP2 internal clock is driven form TCXO) or from external input via BRGIN3 connected to on board SMA.

In case of TDI DCR setup the TS logic is implemented on the external card (on the UFE FPGA), the differential clock should be connected to the UFE differential reference clock input.

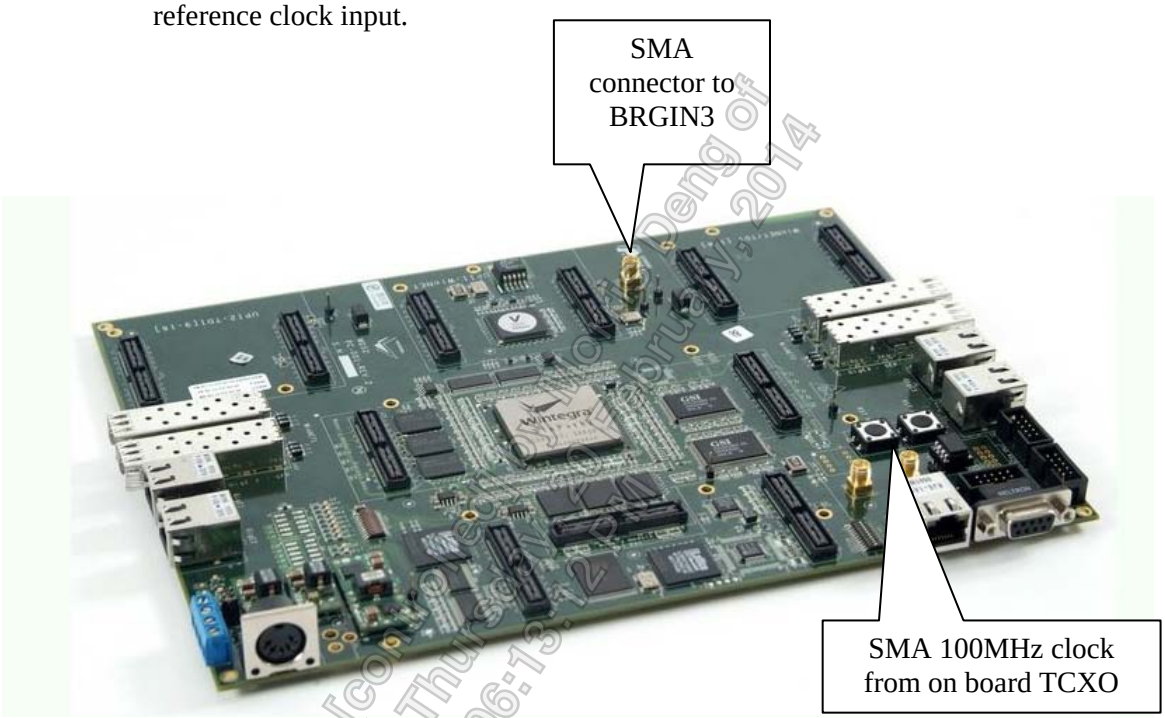


Figure 15: WDS2 Clocks Connectivity

In UFE3 setup over WDS2 the UFE3 adaptive reference clock should be connected to the 100MHz SMA. And the differential reference clock should be the DCO reference clock output from the SMA or from external source.

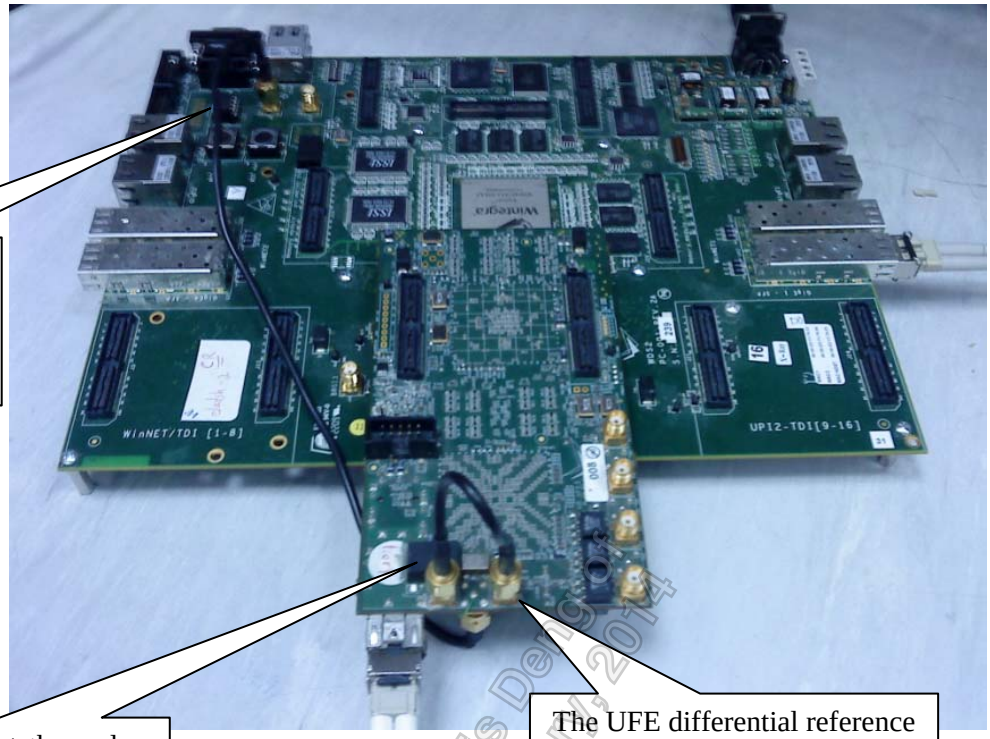


Figure 16: WDS2 Clocks Connectivity - UFE

2.1.1.3 Clock Connectivity – WP3

The WDS3 clocks can support different clocks connectivity required in ACR/DCR (and other setups such as PTP-1588 and SyncE) by using the WDS3-FPGA along with on board Zarlink PLL. The PLL APLL0_CLK is connected to the FPGA and to the Winpath BRGI1 input.

The PLL output can be configured to free running mode, in this mode the PLL can generate an accurate clock (125MHz) derived from the 20MHz TCXO on board.

This PLL output clock can be used as the DCO reference clock, timestamp clock and differential reference clock. Other configuration may be used in other setups.

More information is available in the following documents:

1. WDS3_UG.pdf
2. zl30143-datasheet.pdf

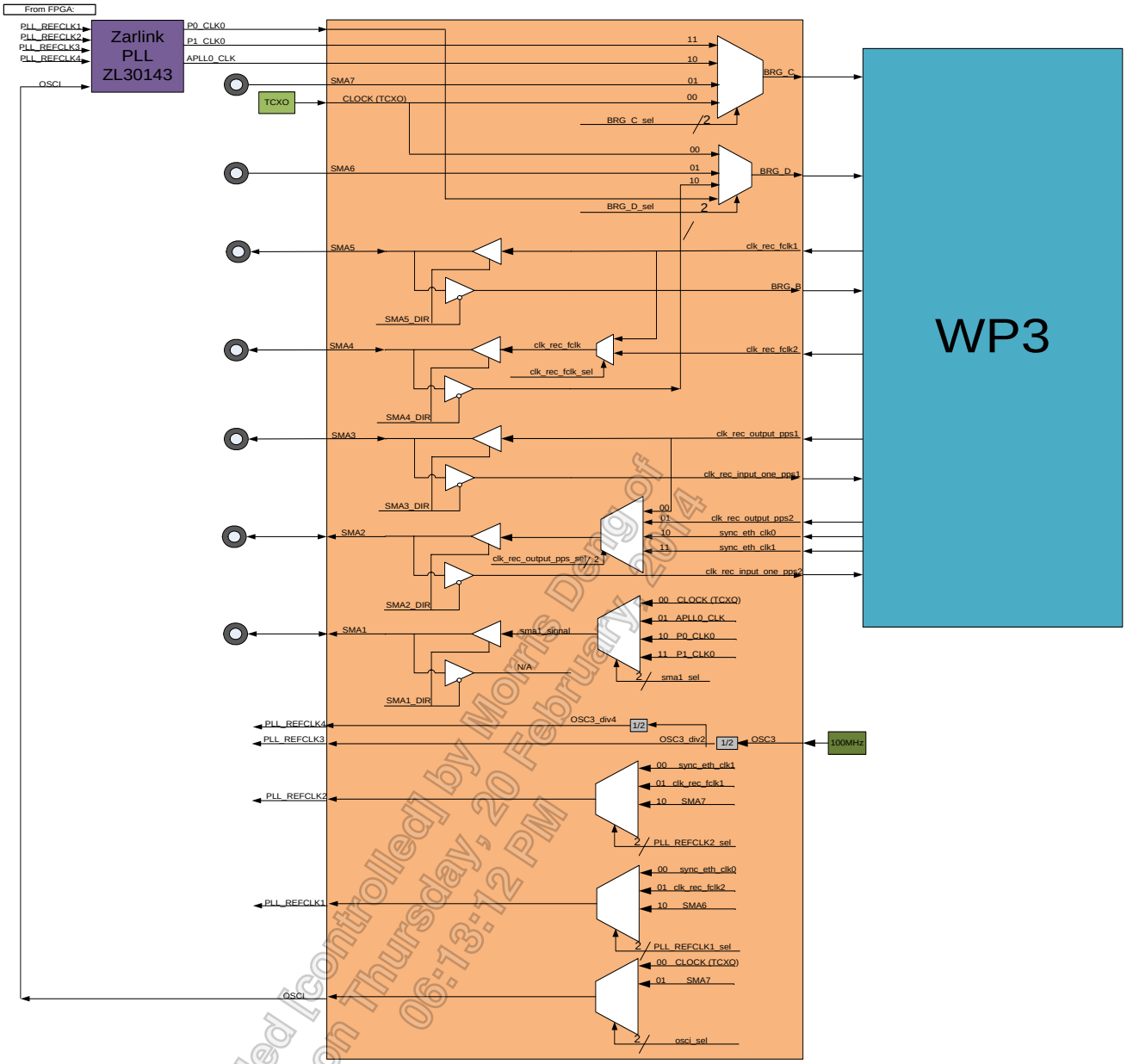


Figure 17: WDS3 FPGA clocking connectivity (10a)

2.1.2. UFE3 Card

The UFE3 card is needed in case OC3/12 or UFE E1/T1 is used. The UFE required an FPGA synthesis according to the desire configuration (WP2/WP3, UFE OC3/12 and UFE CAD).

More information is available in the following documentation:

1. UFE3 datasheet - UFE3 hardware spec.
2. UFE manual - UFE driver manual

NOTE:

1. **External fan is needed in order to cool the UFE3 to the required temperature.**
2. **The UFE3 card for WDS2 and the UFE3 card for WDS3 are NOT the same (they look very similar) – one must make sure the appropriate UFE and FPGA synthesis is used according to the Winpath version used.**

Figure 18: UFE3 Card

2.1.3. UFE4 Cards

2.1.3.1 UFE412

The UFE412 card is needed in case OC3/12 or UFE E1/T1 is used to carry the E1/T1 lines.

More information is available in the following documentation:

1. UFE412 datasheet – UFE412 Hardware Manual.
2. UFE4 programmers guide.
3. PWE3 Clock Recovery User Guide for UFE4.

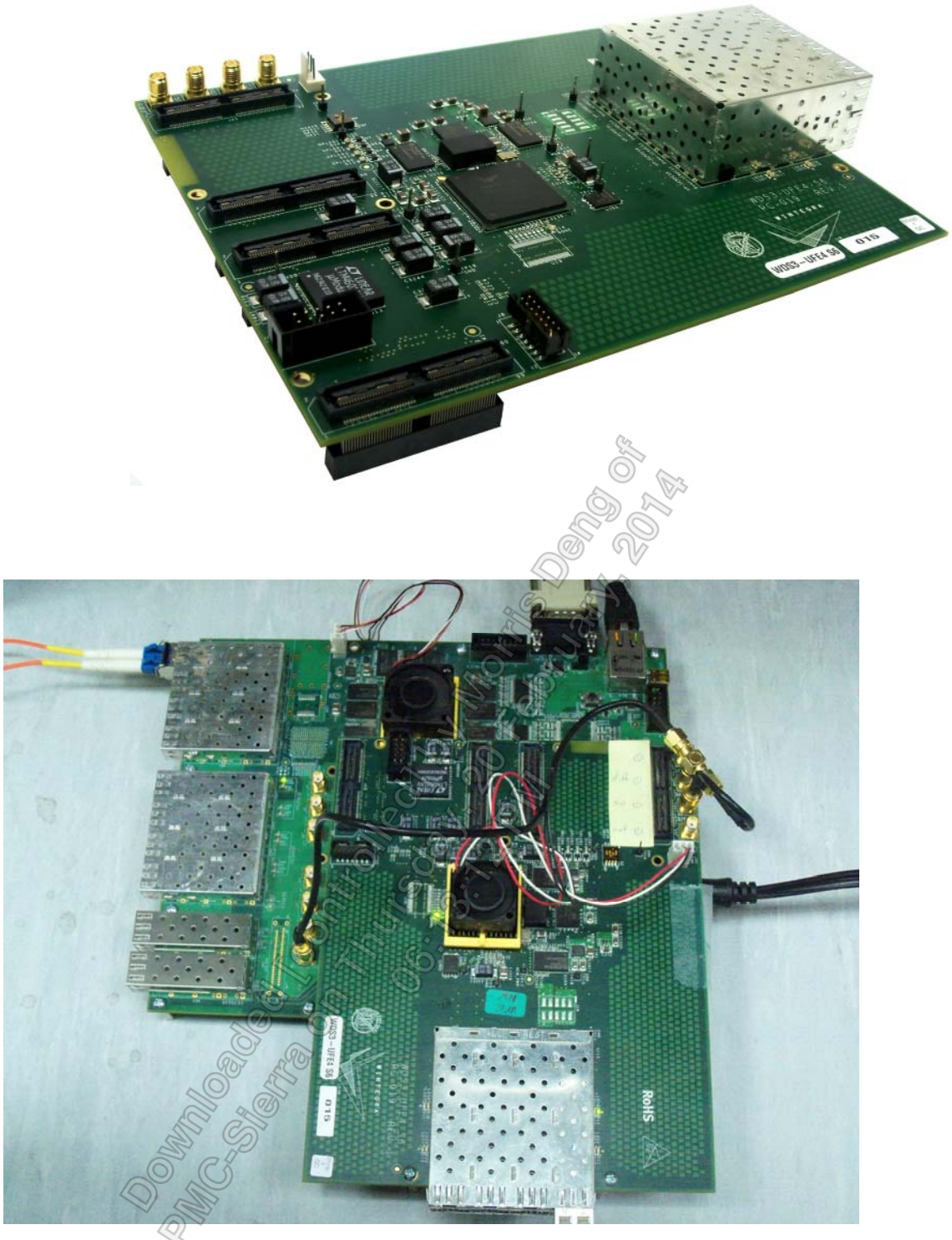


Figure 19: UFE412 over WP3

Note: these are the mapping of the physical line ports of the UFE412

Port 5	Port 4	Port 1	Port 0
Port 7	Port 6	Port 3	Port 2

2.1.3.2 UFE448

The UFE448 card is needed in case OC48 or UFE E1/T1 is used.

More information is available in the following documentation:

1. UFE448 Hardware Manual
2. UFE4 manual - UFE driver manual
3. PWE3 Clock Recovery User Guide for UFE4.

Note: these are the mapping of the physical line ports of the UFE448

Port 3	Port 2	Port 1	Port 0

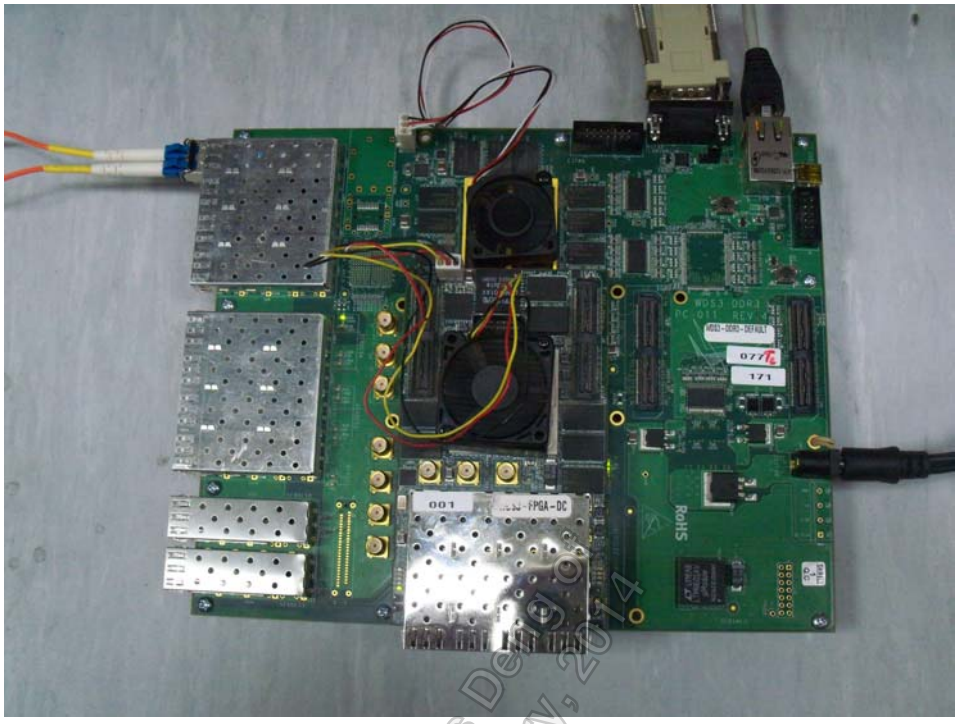


Figure 20: UFE**448 over WP3**

Note: an Ethernet loopback should be plugged in the Ethernet port SFP in order to transfer the data back to the ufe. The default Ethernet ports are ports 13 and 14

2.1.4. E1 / T1 COMET Card

The COMET card is needed in case TDI setup is used. The COMET required an COMET synthesis according to the desire configuration (WP2/WP3).

Each COMET card can support up 8 E1/T1 ports therefore 2 COMET cards may be needed in case 16 E1/T1 setup is used.

More information is available in the following documentation:

1. WDS2 CometOctalUsersGuide.pdf
2. PMC_4358_sw_regs.pdf

NOTE:

1. The COMET card for WDS2 and the COMET card for WDS3 are NOT the same (they look very similar) – one must make sure the appropriate COMET card and CPLD is used according to the Winpath version used. In COMET for WDS2 the RJ45 connectors below the PCB where as in COMET for WDS2 the RJ45 connectors are above the PCB.

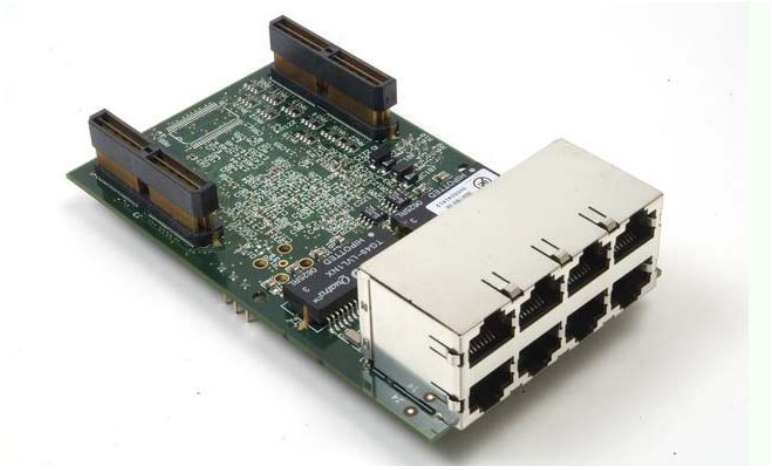


Figure 21: WP2 COMET Card

Figure 22: WP3 COMET Card

2.1.5. PC

A PC is used to run all the software that controls the hardware devices.

The COM port is used for direct control over the board. It uses a direct connection from the PC's COM port to the WDS's COM port.

The Enet port is used for a loading of the application to the WDS. It uses either a direct cable connection between the WDS Enet port and an Ethernet port on the PC, or through a network connection.

2.1.6. Network Emulator

In order to test the system under different network conditions, a network emulator can be connected between the master's and slave's Enet ports.

A direct crossed cable or fiber can also be used to connect the two. This case would provide a very low (almost zero) network delay.

One option is to use the Paragon from Calnex as a network emulator. See the Paragon manual for further information.

If a paragon is used:

- Connect one Enet cable from the appropriate WDS Enet port (copper or optical depending upon configuration) of the master to the Paragon's "Ethernet port 1".
- Connect one Enet cable from the appropriate WDS Enet port (copper or optical depending up on configuration) of the slave to the Paragon's "Ethernet port 2".
- Connect the Paragon according to the Paragon manual..

Chapter 3. Software Setup

3.1. Required Software

The Demo can be run with two operating systems:

- Linux
- Winmon

Each piece of hardware needs dedicated software:

- WDS
  - DEMO executable
- PC
  - TeraTerm
  - TFTPD32 (if running with Winmon)
- Network Emulator (optional for performance)
  - Calnex's Paragon Remote Client

3.2. Software Setup

3.2.1. TFTPD32

TFTPD32 is a TFTP program that allows the WinMon, which is the Operating System running on the WDS upon startup, to load the demo executable from a specified location. The Load is performed through the network connecting the boards and the PC.

After opening the program:

1. Set the PC's IP in the "server interfaces" field (192.w.x.y).
2. Browse to the location of the demo executable.
3. in the WDS : eld 192.w.x.y demo_xxx.exe;go

3.3. Demo Files

The demo contains various test cases. Usually, each case test verifies specific scenario but some of the cases test full and complex systems (see the examples section).

The demo includes the following files:

1. pwe3_cesop_demo.c

   Main demo file. It includes various demo menus that used in order to initialize the CES over Packet system with or without CR.

2. wti_cesopsn_predef_cfg.c

   This file holds different predefined demo configuration that can be run over TDI and UFE3. Each function within this file is called from the pwe3_cesop_demo.c demo menu. In order to use these predefined configuration you should chose option “10” from the demo when it runs.

3. wti_cesopsn_predef_ufe412.c

   This file holds different predefined demo configuration that can be run over UFE412. Each function within this file is called from the pwe3_cesop_demo.c demo menu. In order to use these predefined configuration you should chose option “10” from the demo when it runs.

4. wti_cesopsn_predef_ufe448.c

   This file holds different predefined demo configuration that can be run over UFE448. Each function within this file is called from the pwe3_cesop_demo.c demo menu. In order to use these predefined configuration you should chose option “10” from the demo when it runs.

5. wti_cesopsn_util.c

   This is the main utility file within this demo. This file includes the CES over Packet utility functions that SHOWS how to use WDDI API for CES over Packet application. The user must use the WDDI CESoP API as it is shown in this file.

6. wti_cesopsn_util.h

   This file includes various definitions for CESoP utility functions.

7. wti_cesopsn_defines.h

   This file includes various definitions for demo functions.

8. wti_ufe_util.c

   This file includes the utility functions that show usage of the UFE2/UFE3 API. The UFE drivers, together with PMC Temux84/336 drivers, are required in order to compile and run this demo.

9. wti_ufe_utility.c

   This file includes the utility functions that show usage of the UFE4 API. The UFE drivers, together with UFE4 internal framer (flexmux) drivers, are required in order to compile and run this demo.

10. wti_ufe_util.h

This file includes various definitions for UFE3 utility functions.

11. wti_ufe_utility.h

    This file includes various definitions for UFE4 utility functions.

12. wti_tun_driver_util.c

    This file includes linux TUN driver utility functions.

13. wti_cesopsn_statistics.c

    This file includes various statistics usage.

14. wti_tmx84_util.c and wti_tmx336_util.c

    Those files include the utility functions for UFE3 external framer (TEMUX) tmx336.

15. wti_flexmux_util..c

    This files include the utility functions for UFE4 internal framer (flexmux).

16. wti_flexmux_util..h

    This file includes various definitions for UFE4 internal framer (flexmux) utility functions.

3.4. Demo Configuration

The following variables need to be set inside pwe3_satop_cesop_demo.c

WTI_CESOP_TDI	Should be set for TDI interface and cleared for UFE interface
WTI_CESOP_CLOCK_RECOVERY_ENABLE	Should be set when running with clock recovery
WTI_CESOP_RX_TIMING_ENABLE	Should be set when using RTP timestamp
WTI_CESOP_REGRESSION_TEST	Control the clock source when working in SBI mode. When transmitting using SBI, the SBI extracts the clock from the line, when no clock appears on the optic line (like in SBI external loopback) the clock must be revealed internally. The later case called REGRESSION TEST. Set bit to '0' when TDM /OC3/12 line is connected to test equipment, and '1' when using external loopback on the OC3/12 line.
WTI_XGI_MODE	Should be set to '1' when working with 10 Giga Ethernet. In that case, WTI_ENET_PORT and WTI_ENET_TYPE should be defined properly. This mode is not valid for WP2.
WTI_CESOP_MPLS_IW	Should be set when using MPLS over HSPI/L2PI. In this case the WTI_DFC_CLASSIFIER must be cleared. In UFE4 this bit MUST be set (UFE4 predefined scenarios are using MPLS cascading).
WTI_CESOP_MEF8_ENCAP	Specifies whether MEF8 encapsulation is used.
WTI_MPLS_LOOPBACK	Should be set when using HSPI/L2PI port in loopback mode, and cleared when using normal mode with OC12 card
WTI_DFC_CLASSIFIER	Should be set when DFC is used. In this case the WTI_CESOP_MPLS_IW must be cleared.
WTI_ENET_MODE	Should be set to WP_ENET_LOOPBACK for loopback mode and to WP_ENET_NORMAL otherwise. This mode is active only when DFC is used.
TWO_BOARDS_SIMULATION	Set this bit to '1' when running two boards application.
WTI_BOARD_1	Define that parameter for master board, undefined it for slave board.
WTI_2_ENET_DEVICES	Set this bit to '1' when transmit from one Enet to second Enet on the same board. In this case, WTI_SECOND_ENET_MODE, WTI_SECOND_ENET_PORT, WTI_SECOND_ENET_TYPE and WTI_SECOND_GMII_TYPE should be defined properly.
WTI_8K_CH_SETUP	This mode is required in order to configure 8064 PW's on the UFE side.
WTI_CAD_0_3_SETUP	This mode is required in order to configure CAD over UFE3 - requires suitable CAD synthesis on the UFE3. The physical configuration is UFE3 card on UPI2 and Comet card on TDI1-8.
WTI_EMPHY_PORT	Set the UPI port used. In case UFE412 is used - WP_PORT_UPI1 must be use,
WTI_UFE_4_1_2	This bit must be set incase UFE412 IS USED and clear otherwise.
WTI_CLOCK_REC_MODE	Set this bit to '1' when working in regular differential CR mode. Set this bit to '0' when working in adaptive CR mode.
WTI_EXT_DIFF_MODE_ACTIVE	Set this bit to '1' when working in extended differential CR mode. In this case, WTI_CLOCK_REC_MODE must be

	set to ‘0’.
WTI_CES_SHAPING_ENABLED	When working in adaptive CR mode with UFE4 , set this bit to "1" to enable TDM2PSN packets shaping (required for PSN2TDM clock recovery). Note: For now CES Shaping is not supported with MPLS Cascading. WTI_CESOP_TO_MPLS_FA_CASCADING should be set to "0".
WTI_CES_SHAPING_MAX_CHANNELS	Defines the max number of Rx channels that can be used in CES Shaping mode (ACR).
WTI_ADM_MODE	Set to "1" when working with PMC Add-drop Multiplexer.
WTI_ACTIVE_CR_FOR_ASYNC_MODE	Defines the number of active CRS in the system (for SBI and CAD).
WTI_ACTIVE_CR_FOR_EXT_DIFF_MODE	Defines the number of active CRS in extended differential mode. In order to active this mode, set WTI_EXT_DIFF_MODE_ACTIVE to '1' and select the number of extended differential CD by setting this define. In addition, set WTI_CLOCK_REC_MODE to '0' (release 4.0 does not support combined modes of regular and extended differential CD).
WTI_CLOCK_REC_TDI_LINES_4_7	Defines on which TDI lines to do CR (0-3 or 4-7). Set bit to '1' if CR is done on lines 4-7. Valid in WP2 only.
WTI_CLOCK_REC_ADA_BW_COM_MODE	In CR Adaptive CAD mode, the TS can be taken in DPS instead of UFE (from UFE synthesis 2.4). Only the DCO is in the UFE FPGA. In normal operation mode, set this bit to '0'.
WTI_CESOP_DIFF_DOCSIS_MODE	When working in DOCSIS mode, set this bit to '1'.
WTI_TDI_DCO_CLOCK_SOURCE	Defines the source clock of the TDI’s DCO.
WTI_TDI_TS_CLOCK_SOURCE	Defines the source clock of the TDI TS. Not in used in WP2
WTI_WG_TS2_CLOCK_SOURCE	Defines the source clock of the WinGine TS. Not in used in WP2

3.5. Demo Compilation

WDDI sources, UFE driver ,PMC Temux336 and UFE4 framer drivers must be compiled prior the demo compilation. The demo is compiled with the following compiler:

For Linux:

mips-linux-gnu-gcc Target: mips-linux-gnu gcc version 4.2.3 (Sourcery G++ Lite 4.2-177) compiler.

For Winmon:

mips-sde-elf-gcc Target: mips-sde-elf gcc version 4.4.1 (Sourcery G++ Lite 4.4-87) compiler.

And can be run using WinMon/Linux.

The compilation command should be as presented in the example below:

make WDDS_DIR=/xxx/WDDS \

WDDI_TARGET_NAME=mips_winmon \ mips_linux

WDDI_BOARD_NAME=wds \

WDDI_HOST=linux \

WDDI_HW_DEVICE=winpathN \ (N= 2 or 3)

UFE_DIR=/xxx/wddi/phy/wpx_ufe \

UFE=ufe3/ufe412/ufe412dual/ufe448 \

PMC=tmx336 (for ufe3 only) \

FRAMER=flexmux (ufe4 only)

Note that there are two versions of UFE4:

The UFE448 which supports OC48, and the UFE412 which supports OC12. Both versions are compiled with the same command. (With the suitable flags)

It is important when compiling the test to notice with each version you are working. In the file pwe3_cesop_demo.c there is macro
#define WTI_UFE_4_1_2
When working with the UFE448 this macro should be off and when working with UFE412 this macro should be on.

When compiling UFE412 make sure to set the WTI_INITIALIZE_FIRMWARE in wti_cesopsn_util.c. This flag causes the UFE412 firmware to be programmed from the host when initializing the UFE. (for UFE448 this flag should be clear)

In order to compile to dual EMPHY there is a need to compile with UFE=ufe412dual, it will define the macros: WUFE_DUAL_EMPHY and __WT_UFE4_DUAL_EMPHY__.

Note: in order to work with dual EMPHY the UFE driver needs to be compiled with the flag UFE= ufe412dual as well.

The 'pwe3_cesop_demo' executable is located at WDDS_DIR/objects/WDDI_HOST/WDDI_TARGET_NAME/WDDI_BOARD_NAME/bin.

3.6. Demo Menu Usage

3.7. Demo

This section includes several examples on how to configure CR tests including board setup and on-board clocks connections.

3.7.1. Load Demo

After all needed hardware and software are set, the demo can be loaded and run as shown below:

```
WinMon>@ld
eld 192.168.151.77 37900b.exe;go

ethx: Connecting to host 192.168.151.77
Host connected. Downloading file 37900b.exe
Hit Cntl-C to exit
ELF Read: copying 0x0098bf60 bytes to address 0x80400000-0x80d8bf5f
Entry point: 0x80400000
************************************************************
            SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu
  Name: Main Menu
************************************************************
1.    -> System setup
2.    -> Ufe setup
3.    -> PW add
4.    -> PW delete
5.    -> PW enable
6.    -> PW disable
7.    -> PW modify
8.    -> PW display
9.    -> System display
10.   -> Predefined configurations
11.   -> Statistics
12.   -> Host interface
13.   -> Debug utilities
14.   -> Memory Display
15.   -> Quit without reset
16.   -> Quit
```


Figure 23: Load Example

Once the demo is up one of the predefined configuration can be chosen.

Choosing ‘10’ option will lead to all the scenarios implemented in the demo.

Ufe412

```
*****************************************************************
          SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu
  Name: System configurations: UFE4 Scenarios
*****************************************************************
1.  -> Initialize UFE framer (for Flexmux debug) [build_personality (7-1P1, 8-4P4, 15-2P2))]
2.  -> E1, Sonet/Sdh/CAD, Unframed, Without Cas. [0- SDH, 1- SONET, 2- CAD] [physical conf: 0-oc12,1-4oc3,2-2oc3][in SDH:0-AU3
,1-AU4][NUM_OF_LINES]
3.  -> E1, Sonet/Sdh/CAD, Framed, Without Cas. [0- SDH, 1- SONET, 2- CAD][physical conf: 0-oc12,1-4oc3,2-2oc3][in SDH:0-AU3,1-
AU4] [NUM_OF_LINES]
4.  -> E1, Sonet/Sdh/CAD, Framed, With Cas. [0- SDH, 1- SONET, 2- CAD] [NUM_OF_LINES]
5.  -> T1, Sonet/Sdh/CAD, Unframed, Without Cas. [0 - SDH, 1- SONET, 2- CAD][physical conf: 0-oc12,1-4oc3,2-2oc3][in SDH:0-AU3
,1-AU4] [NUM_OF_LINES]
6.  -> T1, Sonet/Sdh/CAD, Framed, Without Cas. [0 - SDH, 1- SONET, 2- CAD][physical conf: 0-oc12,1-4oc3,2-2oc3][in SDH:0-AU3,1
-AU4] [NUM_OF_LINES]
7.  -> T1, Sonet/Sdh/CAD, Framed, With Cas. [0 - SDH, 1- SONET, 2- CAD] [NUM_OF_LINES]
8.  -> E1, Sonet/Sdh, Framed, Without Cas. 4032 8xds0 PWs. [0- SDH, 1- SONET] [NUM_OF_LINES]
9.  -> T1, Sonet/Sdh, Framed, Without Cas. 4032 6xds0 PWs. [0- SDH, 1- SONET] [NUM_OF_LINES]
10.  -> E1 & T1 combined, Sonet/Sdh, Unframed, Without Cas. [0- SDH, 1- SONET] [NUM_OF_LINES]
11.  -> E1, Sonet/Sdh, System Reconfigure. [0- SDH, 1- SONET]
12.  -> E1, Sonet/Sdh, Line Reconfigure. [0- SDH, 1- SONET]
13.  -> E1, Sonet/Sdh, Phy Reconfigure. [0- SDH, 1- SONET]
14.  -> T1, Sonet/Sdh, Phy Create/Delete. [0- SDH, 1- SONET]
15.  -> E1, Sonet/Sdh, Phy Enable/Disable. [0- SDH, 1- SONET][0-load lines][1-disable][2-enable][3-enable & disable in loop]
16.  -> T1, Sonet/Sdh, Line Enable/Disable. [0- SDH, 1- SONET, 2- CAD]
17.  -> E1, Sonet/Sdh, Line Enable/Disable. [0- SDH, 1- SONET, 2- CAD]
18.  -> E1, Sonet/Sdh, error_4_e1_on_1_tug2. [0- SDH, 1- SONET]
19.  -> E1, Sonet/Sdh, error_disable_before_enable. [0- SDH, 1- SONET]
20.  -> E1, Sonet/Sdh, error_ReEnable_line. [0- SDH, 1- SONET]
21.  -> E1, Sonet/Sdh, Error_Enable_Disable_Disable. [0- SDH, 1- SONET]
22.  -> E1, Sonet/Sdh, Error_Enable_Before_Create. [0- SDH, 1- SONET]
23.  -> CR 101, E1, Sonet/Sdh/CAD, Unframed, Without Cas. [NUM_OF_LINES][0- SDH, 1- SONET, 2- CAD]
24.  -> CR 101, T1, Sonet/Sdh/CAD, Unframed, Without Cas. [NUM_OF_LINES][0- SDH, 1- SONET, 2- CAD]
25.  -> CR 101, E1, Sonet/Sdh/CAD, Framed, Without Cas. [NUM_OF_LINES][0- SDH, 1- SONET, 2- CAD]
26.  -> CR 101, T1, Sonet/Sdh/CAD, Framed, Without Cas. [NUM_OF_LINES][0- SDH, 1- SONET, 2- CAD]
27.  -> CR 101, E1, Sonet/Sdh/CAD, Framed, 1xDS0, slot #30, Without Cas. [NUM_OF_LINES][0- SDH, 1- SONET, 2- CAD]
28.  -> CR 101, T1, Sonet/Sdh/CAD, Framed, 1xDS0, slot #14, Without Cas.  [NUM_OF_LINES][0- SDH, 1- SONET, 2- CAD]
29.  -> CR 102, E1, Framed, With Cas. [NUM_OF_LINES][0-SDH, 1-SONET, 2-CAD][0- Unfragmented, 1-Fragmented]
30.  -> CR 102, T1, Framed, With Cas. [NUM_OF_LINES][0-SDH, 1-SONET, 2-CAD][0- Unfragmented, 1-Fragmented]
31.  -> CR 103, E1, Framed, No Cas. Multiple PWs per line [NUM_OF_LINES][0-SDH, 1-SONET, 2-CAD][#PWs/line][recovered PW]
32.  -> CR 103, T1, Framed, No Cas. Multiple PWs per line [NUM_OF_LINES][0-SDH, 1-SONET, 2-CAD][#PWs/line][recovered PW]
33.  -> CR 104, E1, Unframed, No Cas. Remove/recreate all PW/lines. [NUM_OF_LINES][0-SDH, 1-SONET, 2-CAD][0-remove PW, 1-remov
e PW+Lines]
34.  -> CR 104, T1, Unframed, No Cas. Remove/recreate all PW/lines. [NUM_OF_LINES][0-SDH, 1-SONET, 2-CAD][0-remove PW, 1-remov
e PW+Lines]
```

In ufe4 options 2-3,5-6 implement the cases of E1/T1 lines. both framed and unframed.
For T1 lines the user can configure 336 lines and for E1 – 252 lines

For example we can choose to configure one E1 line in SDH mode with OC12
So we have to choose option number two as you can see below:

```
# Enter your command:2 0 0 0 1
```

3.7.2. UFE Events

The demo contains UFE events display statistics which is accessible from the main menu. Select option ‘2’ (Ufe setup) then ‘2’ (Display UFE events):

```
*****************************************************************
            SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu
  Name: UFE Setup
*****************************************************************
1.  -> Show info
2.  -> Display UFE events
3.  -> Check UFE Temperature
4.  -> UPI port [0-UPI1, 1-UPI2]
5.  -> FPGA mode [1-OCTAL, 2-SBI]
6.  -> SPE transfer mode [spe_id (0-2) spe_transfer_mode (0-IDLE, 1-E1, 2-T1, 3-DS3, 4-E3)]
7.  -> Transfer mode [line_index (octal (0-7), SBI (0-83)) transfer_mode (1-T1_FRM, 2-T1_UNFRM 3-E1_FRM 4-E1_UNFRM), clo
ck_rec_mode 0-disable 1-tx 2-rx 3-duplex]
8.  -> CAS mode [line_index (octal (0-7), SBI (0-83)) cas_mode (0-DISABLE, 1-ENABLE)]
9.  -> Set clock recovery method (0-adaptive, 1-differential)]
10.  -> Clock translator ID [line_index(0-83), translator ID (0-31)]
11.  -> Set UFE line Tx clock mode [line_index(0-83))]
12.  -> SBI loopback mode [0-NORMAL, 1-LOOPBACK, 2-SYSTEM LOOPBACK]
13.  -> TDM loopback mode [0-NO LPBK,1-SYSTEM LPBK,2-LINE LPBK][line_index]
14.  -> Clock mode [0-LOOP TIMING, 1-EXTERNAL]
15.  -> Initialize external chip and create UFE lines
16.  -> Temux Register dump
17. PW add
# Enter your command:2
core 0 event register 0x0000
emphy event register 0x0000
core 1 event register 0x0000
emphy event register 0x0000
# Enter your command:
```

Figure 24: UFE Events Display

The display includes the events of the two cores and in normal operation should be all cleared after the second read (the first read clears the events) .

3.7.3. UFE3 Junction Temperature

The demo contains UFE3 temperature display statistics which is accessible from the main menu. Select option ‘2’ (Ufe setup) then ‘3’ (Check UFE Temperature):

```
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu
  Name: UFE Setup
*****************************************************************
1.  -> Show info
2.  -> Display UFE events
3.  -> Check UFE Temperature
4.  -> UPI port [0-UPI1, 1-UPI2]
5.  -> FPGA mode [1-OCTAL, 2-SBI]
6.  -> SPE transfer mode [spe_id (0-2) spe_transfer_mode (0-IDLE, 1-E1, 2-T1, 3-DS3, 4-E3)]
7.  -> Transfer mode [line_index (octal (0-7), SBI (0-83)) transfer_mode (1-T1_FRM, 2-T1_UNFRM 3-E1_FRM 4-E1_UNFRM), clo
ck_rec_mode 0-disable 1-tx 2-rx 3-duplex]
8.  -> CAS mode [line_index (octal (0-7), SBI (0-83)) cas_mode (0-DISABLE, 1-ENABLE)]
9.  -> Set clock recovery method (0-adaptive, 1-differential)]
10.  -> Clock translator ID [line_index(0-83), translator ID (0-31)]
11.  -> Set UFE line Tx clock mode [line_index(0-83))]
12.  -> SBI loopback mode [0-NORMAL, 1-LOOPBACK, 2-SYSTEM LOOPBACK]
13.  -> TDM loopback mode [0-NO LPBK,1-SYSTEM LPBK,2-LINE LPBK][line_index]
14.  -> Clock mode [0-LOOP TIMING, 1-EXTERNAL]
15.  -> Initialize external chip and create UFE lines
16.  -> Temux Register dump
17. PW add
# Enter your command:3
Temperature Sensor Standby Mode Was Set!
Remote Temperature:87.625, Max:87.625
Remote Temperature:87.625, Max:87.625
Remote Temperature:87.750, Max:87.750
Remote Temperature:87.625, Max:87.750
Remote Temperature:87.625, Max:87.750
Remote Temperature:87.625, Max:87.750
Remote Temperature:87.625, Max:87.750
Remote Temperature:87.625, Max:87.750
Remote Temperature:87.625, Max:87.750
Remote Temperature:87.625, Max:87.750
Remote Temperature:87.625, Max:87.750
/
```

Figure 25: UFE Temperature Display

The maximal UFE temperature can be 85 C, in case a higher temperature is measured an external fan and/or heat sink must be used.

3.7.4. Statistics

3.7.4.1 Global Statistics

The demo contains data path statistics and CRS statistics both are accessible from the main menu. Selecting option ‘11’ (statistics) for data path statistics and ‘13’ (Clock recovery utils) for CR menu.

Data path statistics can be verified over all lines by clearing all listed statistics and then reading them using options ‘7’ and ‘8’ in the statistics menu respectively.

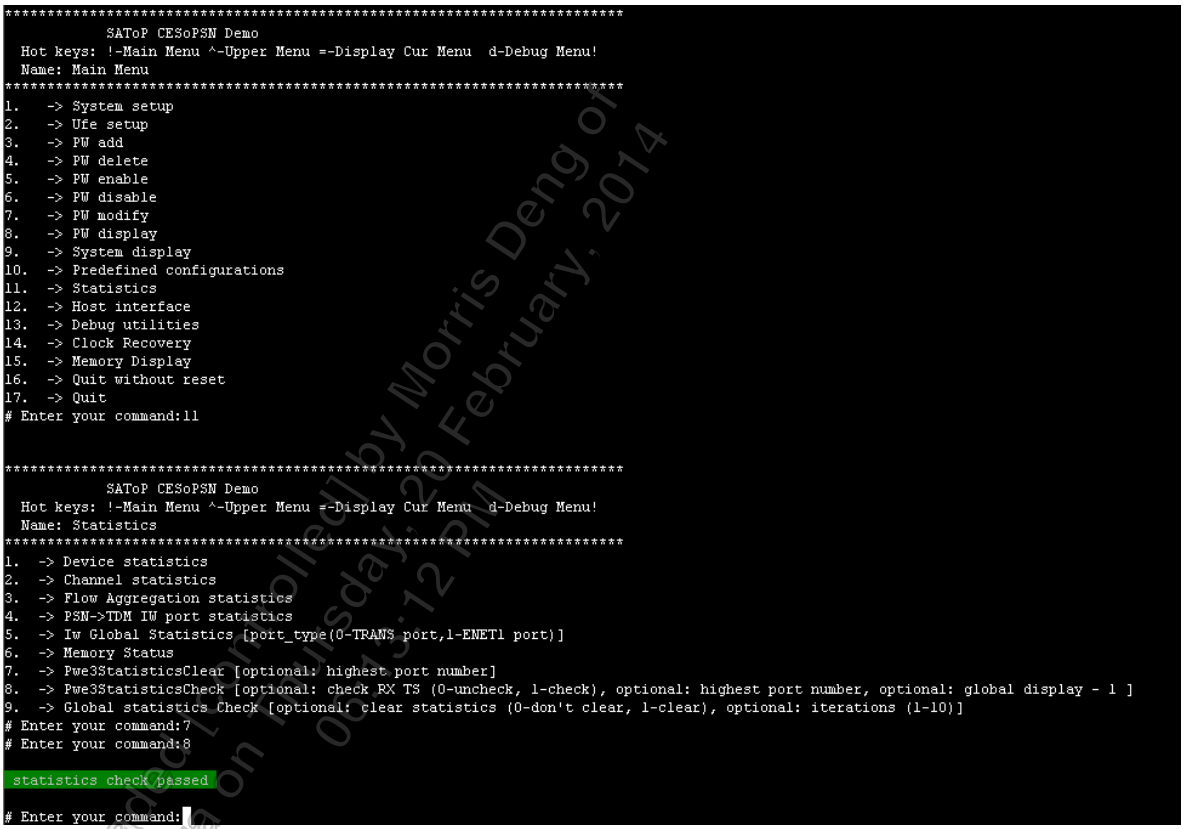


Figure 26: Data path Statistics Check

The statistics check menu item (8) can also display information about the different ports, and it is possible to check the Rx TS if needed. This is done by using the parameters
listed.

```
**************************************************************************
          SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Statistics
**************************************************************************
1.  -> Device statistics
2.  -> Channel statistics
3.  -> Flow Aggregation statistics
4.  -> PSN->TDM IW port statistics
5.  -> Iw Global Statistics [port_type(0-TRANS port,1-ENET1 port)]
6.  -> Memory Status
7.  -> Pwe3StatisticsClear [optional: highest port number]
8.  -> Pwe3StatisticsCheck [optional: check RX TS (0-uncheck, 1-check), optional: highest port number, optional: global display - 1 ]
9.  -> Global statistics Check [optional: clear statistics (0-don't clear, 1-clear), optional: iterations (1-10)]
# Enter your command:8 0 15 1
PW Index:   0 PASS. PW Index:   1 PASS. PW Index:   2 PASS. PW Index:   3 PASS. PW Index:   4 PASS. PW Index:   5 PASS. PW Index:   6 PASS. PW Index
  7 PASS. PW Index:   8 PASS. PW Index:   9 PASS. PW Index:  10 PASS. PW Index:  11 PASS. PW Index:  12 PASS. PW Index:  13 PASS. PW Index:  14 PAS
.

# Enter your command:8 1 15 1
PW Index:   0 PASS. PW Index:   1 PASS. PW Index:   2 PASS. PW Index:   3 PASS. PW Index:   4 PASS. PW Index:   5 PASS. PW Index:   6 PASS. PW Index
  7 PASS. PW Index:   8 PASS. PW Index:   9 PASS. PW Index:  10 PASS. PW Index:  11 PASS. PW Index:  12 PASS. PW Index:  13 PASS. PW Index:  14 PAS
.

# Enter your command:
```

Figure 27: Data path Statistics Check

Using menu item 9 provides a way to check that the entire system is up and running correctly several times. The check includes an option to clear the current statistics and iterate several times.

```
**************************************************************************
1.  -> Device statistics
2.  -> Channel statistics
3.  -> Flow Aggregation statistics
4.  -> PSN->TDM IW port statistics
5.  -> Iw Global Statistics [port_type(0-TRANS port,1-ENET1 port)]
6.  -> Memory Status
7.  -> Pwe3StatisticsClear [optional: highest port number]
8.  -> Pwe3StatisticsCheck [optional: check RX TS (0-uncheck, 1-check), optional: highest port number, optional: global display - 1 ]
9.  -> Global statistics Check [optional: clear statistics (0-don't clear, 1-clear), optional: iterations (1-10)]
# Enter your command:9 1 2

 PWE-3 Clear Statistics: Passed

Reset Statistics from CRS 0 to CRS 335

 Clock Recovery Reset System Statistics: Passed


 statistics check passed


 statistics check passed


 PWE-3 Check Statistics: Passed

Emphy event register for core id 0: 0x0000
TDM event register 0x0000
FM event register 0x0000
Machine event Rx register 0x0000
Machine event Tx register 0x0000

Clock Recovery event register 0x0000

Emphy event register for core id 0: 0x0000
TDM event register 0x0000
FM event register 0x0000
Machine event Rx register 0x0000
Machine event Tx register 0x0000

Clock Recovery event register 0x0000

Emphy event register for core id 0: 0x0000
TDM event register 0x0000
FM event register 0x0000
Machine event Rx register 0x0000
Machine event Tx register 0x0000

Clock Recovery event register 0x0000

There are no UFE Events

Remote Temperature:0.000, Max:0.000
Remote Temperature:0.000, Max:0.000

Statistics of all CRS are zero - PASS

 Clock Recovery System Statistics Status: Passed

CLI_F_GlobalClockRecoveryShowState Passed, All Clocks On NORMAL State


Clock Recovery State: Passed
```

Figure 28: Global Statistics Check

The CR statistics can be verified by clearing all listed statistics and then reading them using options ‘11’ and ‘10’ in the CR utils menu. Selecting option ‘4’ will list the CRS state over all CR systems. Option ‘8’, along with a specific CR index, will print the CR statistics for that CR.

```
# Enter your command:13
*****************************************************************
            SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu
  Name: Clock recovery utils
*****************************************************************
1.  -> view number of packets in queue[pw_index]
2.  -> Force holdover state [pw_index]
3.  -> Clear holdover state [pw_index]
4.  -> Show clock recovery systems states
5.  -> Show clock recovery state [pw_index]
6.  -> Enable clock recovery interface [pw_index]
7.  -> Disable clock recovery interface [pw_index]
8.  -> Clock Recovery statistics [cr table index]
9.  -> Clock Recovery statistics reset [cr table index]
10.  -> Clock Recovery Systems statistics Status
11.  -> Clock Recovery Systems statistics Reset
12.  -> Clock Recovery Command [pw_index, command (0-reset, 1-force freerun)]
13.  -> Clock Recovery Indirect Memory Display
14.  -> Clock Recovery Calculate DCO parameters [service clock, diff_ref clock, ppm_offset]
15.  -> Configure WDS3 board pins in FPGA [config (1-default, 2-ptp), board_number (1,2)]
16.  -> Clock Recovery DCO HW Registers Display [DCO id]
# Enter your command:11
Reset Statistics from CRS 0 to CRS 7
# Enter your command:10
Statistics of all CRS are zero - PASS
# Enter your command:10
Statistics of all CRS are zero - PASS
# Enter your command:10
Statistics of all CRS are zero - PASS
# Enter your command:10
Statistics of all CRS are zero - PASS
# Enter your command:4

CRS state (PW   0): NORMAL_STATE      CRS state (PW   1): NORMAL_STATE      CRS state (PW   2): NORMAL_STATE
CRS state (PW   3): NORMAL_STATE      CRS state (PW   4): NORMAL_STATE      CRS state (PW   5): NORMAL_STATE
CRS state (PW   6): NORMAL_STATE      CRS state (PW   7): NORMAL_STATE
```

Figure 29: CRS Statistics Check

3.7.4.2 Detailed Data Path Statistics (UFE)

General overview of a UFE+WP:

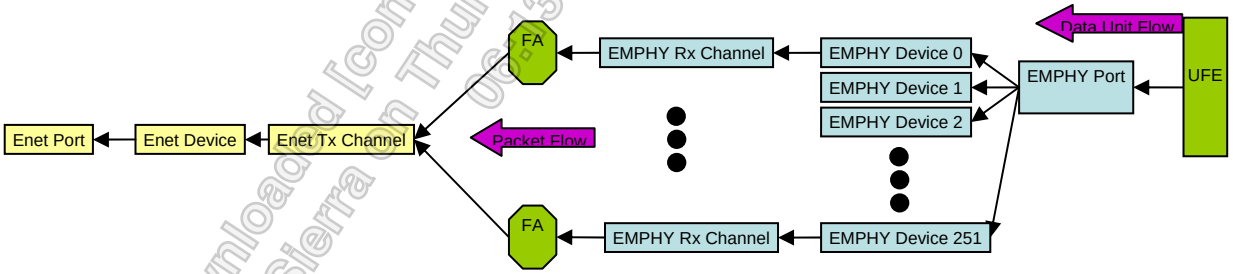


Figure 30: TDM to PSN UFE data path illustration

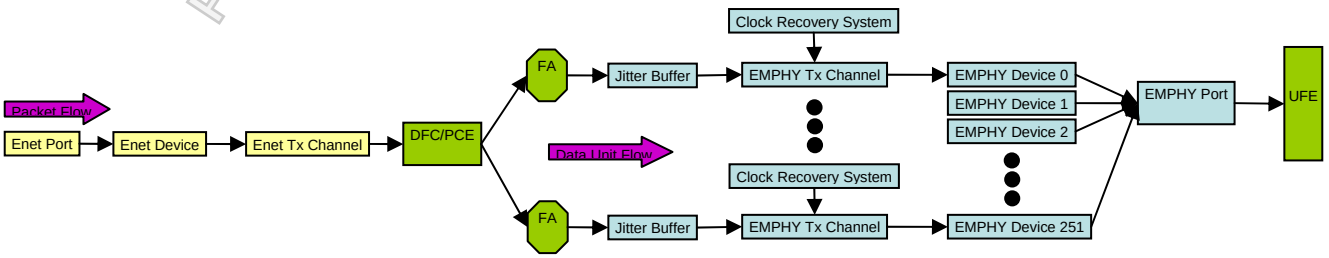
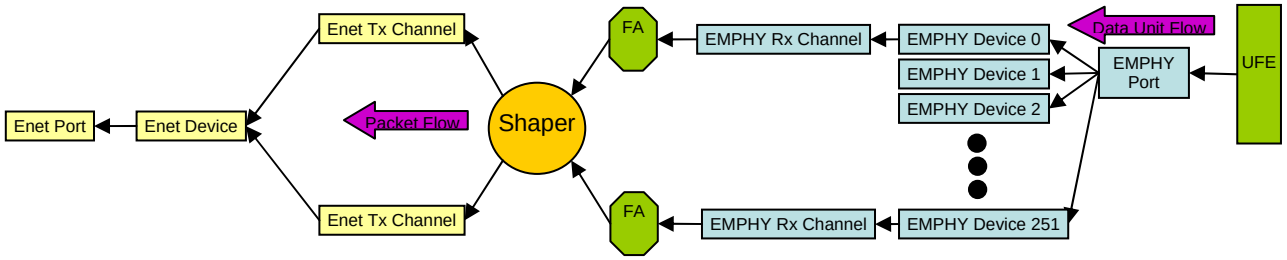


Figure 31: PSN to TDM UFE data path illustration

Figure 32: TDM to PSN UFE CES Shaping data path illustration




```
**************************************************************************
           SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Statistics
**************************************************************************
1.  -> Device statistics
2.  -> Channel statistics
3.  -> Flow Aggregation statistics
4.  -> PSN->TDM IW port statistics
5.  -> Iw Global Statistics [port_type(0-TRANS port,1-ENET1 port)]
6.  -> Memory Status
7.  -> Pwe3StatisticsClear [optional: highest port number]
8.  -> Pwe3StatisticsCheck [optional: check RX TS (0-uncheck, 1-check), optional: highest port number, optional: global display - 1 ]
9.  -> Global statistics Check [optional: clear statistics (0-don't clear, 1-clear), optional: iterations (1-10)]
# Enter your command:1


**************************************************************************
           SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Device Statistics
**************************************************************************
1.  -> Enet Device Statistics
2.  -> EMPHY Port Statistics
3.  -> EMPHY Transparent Device Statistics [device_num]
# Enter your command:2
```

Figure 33: EMPHY port statistics menu

```
EMPHY Port STATISTICS:

Port HANDLE c100000:
----------------------------------------

pos_err:              0000000000000000      rx_err_parity:              0000000000000000
rx_unsupported_device:0000000000000000      tx_unsupported_device:      00000000000000e8
rx_err_sop_eop:       0000000000000000
```

Figure 34: EMPHY port statistics

```
**************************************************************************
           SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Device Statistics
**************************************************************************
1.  -> Enet Device Statistics
2.  -> EMPHY Port Statistics
3.  -> EMPHY Transparent Device Statistics [device_num]
# Enter your command:3 5
```

Figure 35: EMPHY device statistics menu

```
DEVICE STATISTICS:

EMPHY TRANSPARENT DEVICE 5 ( DEVICE HANDLE 11900005 ):
-------------------------------------------------

phy_buffer_overrun:    0000000000000000          phy_out_of_sync:          0000000000000000
```

Figure 36: EMPHY device statistics

```
**************************************************************************
           SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Statistics
**************************************************************************
1.  -> Device statistics
2.  -> Channel statistics
3.  -> Flow Aggregation statistics
4.  -> PSN->TDM IW port statistics
5.  -> Iw Global Statistics [port_type(0-TRANS port,1-ENET1 port)]
6.  -> Memory Status
7.  -> Pwe3StatisticsClear [optional: highest port number]
8.  -> Pwe3StatisticsCheck [optional: check RX TS (0-uncheck, 1-check), optional: highest port number, optional: global display - 1 ]
9.  -> Global statistics Check [optional: clear statistics (0-don't clear, 1-clear), optional: iterations (1-10)]
# Enter your command:2


**************************************************************************
           SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Channel Statistics
**************************************************************************
1.  -> PWE3 Rx Channel statistics [pw_index]
2.  -> PWE3 Tx Channel statistics [pw_index]
3.  -> Clear PWE3 Rx Channel statistics [pw_index]
4.  -> Clear PWE3 Tx Channel statistics [pw_index]
# Enter your command:1 60
```

Figure 37: EMPHY channel statistics menu

On proper operation the 'RX TS sync lost' field (marked in red) should remain constant.

```
CHANNEL STATISTICS:

TRANSPARENT PWE3 RX CHANNEL 60 ( CHANNEL HANDLE 5e00047 ):
---------------------------------------------------

channel_out_of_sync:   0000000000000000          empty fbp:          0000000000000000
valid_packets:         0000000000253aba          RX TS sync lost:    0000000000000000

Free Buffer Pool size: 0x12c6 (min 0x12c6,   max 0x12cf)
```

Figure 38: EMPHY channel statistics

```
**********************************************************************
           SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Statistics
**********************************************************************
1.  -> Device statistics
2.  -> Channel statistics
3.  -> Flow Aggregation statistics
4.  -> PSN->TDM IW port statistics
5.  -> Iw Global Statistics [port_type(0-TRANS port,1-ENET1 port)]
6.  -> Memory Status
7.  -> Pwe3StatisticsClear [optional: highest port number]
8.  -> Pwe3StatisticsCheck [optional: check RX TS (0-uncheck, 1-check), optional: highest port number, optional: global display - 1 ]
9.  -> Global statistics Check [optional: clear statistics (0-don't clear, 1-clear), optional: iterations (1-10)]
# Enter your command:3


**********************************************************************
           SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Flow Aggregation Statistics
**********************************************************************
1.  -> TDM to PSN Flow Aggregation statistics [flowagg_num, type(0-DATA,1-CAS)]
2.  -> PSN to TDM Flow Aggregation statistics [flowagg_num]
3.  -> L2 ROUTER MPLS Flow Aggregation statistics [flowagg_num, dir(0-tdm2psn, 1-psn2tdm)]
4.  -> Clear L2 ROUTER MPLS Flow Aggregation statistics [flowagg_num, dir(0-tdm2psn, 1-psn2tdm)]
# Enter your command:1 125 0
```

Figure 39: TDM to PSN flow aggregation statistics menu

```
FLOW AGGREGATION STATISTICS:

TDM TO PSN FLOW AGGREGATION 125 ( FLOW AGGREGATION HANDLE 34c001le ):
-------------------------------------------------------------------

forwarded_packets:     0000000000057733          fbp_drop_packets:     0000000000000000
mtu_drop_packets:      0000000000000000          tx_queue_drop_packets:0000000000000000
```

Figure 40: TDM to PSN flow aggregation statistics

On proper operation all fields (excluding good packets) should be "0" except on start up.

```
 CES shaping stats:
-------------------------------------------------------------
underrun (FTD packets = 0):  0000000000000000    l2pi fifo put error:     0000000000000000
overrun  (FTD packets > 1):  0000000000000000    l2pi fifo get error:     0000000000000000
invalid delta ts:            0000000000000000    good packets (fifo put): 000000000000773e
```

Figure 41: TDM to PSN CES shaping flow aggregation statistics

```
**********************************************************************
           SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Statistics
**********************************************************************
1.  -> Device statistics
2.  -> Channel statistics
3.  -> Flow Aggregation statistics
4.  -> PSN->TDM IW port statistics
5.  -> Iw Global Statistics [port_type(0-TRANS port,1-ENET1 port)]
6.  -> Memory Status
7.  -> Pwe3StatisticsClear [optional: highest port number]
8.  -> Pwe3StatisticsCheck [optional: check RX TS (0-uncheck, 1-check), optional: highest port number, optional: global display - 1 ]
9.  -> Global statistics Check [optional: clear statistics (0-don't clear, 1-clear), optional: iterations (1-10)]
# Enter your command:1


**********************************************************************
           SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Device Statistics
**********************************************************************
1.  -> Enet Device Statistics
2.  -> EMPHY Port Statistics
3.  -> EMPHY Transparent Device Statistics [device_num]
# Enter your command:1
```

Figure 42: Enet device and port statistics menu

```
-------------------------------------------------
HS ENET Port Statistics (HW)
TxRx Frame  64:              0000000000000000
Tx Bytes:                    000000006f464f50
TxRx Frame 127:              0000000000000000
Tx Packets:                  000000000123fea8
TxRx Frame 255:              0000000000000000
Tx Multicast:                0000000000000000
TxRx Frame 1023:             0000000000000000
Tx Broadcast:                0000000000000000
TxRx Frame 1518:             0000000000000000
Tx Mac Pause:                0000000000000000
TxRx Frame 1522:             0000000000000000
Tx Defer:                    0000000000000000
Rx Bytes:                    000000006f464e0e
Tx Exess Defer:              0000000000000000
Rx Packets:                  000000000123fea7
Tx Single Collision:         0000000000000000
Rx Error FCS:                0000000000000000
Tx Multi Collision:          0000000000000000
Rx Multicast:                0000000000000000
Tx Late Collision:           0000000000000000
Rx Broadcast:                0000000000000000
Tx Excess Collision:         0000000000000000
Rx Mac Control:              0000000000000000
Tx No Collision:             0000000000000000
Rx Mac Pause:                0000000000000000
Tx Mac Pause Honored:        0000000000000000
Rx Mac Unknown:              0000000000000000
Tx Dropped:                  0000000000000000
Rx Error Alignment:          0000000000000000
Tx Jabber:                   0000000000000000
Rx Error LEN:                0000000000000000
Tx Errors FCS:               0000000000000000
Rx Error Code:               0000000000000000
Tx Control:                  0000000000000000
Rx False Carrier:            0000000000000001
Tx Oversize:                 0000000000000000
Rx Undersize:                0000000000000000
Tx Undersize:                0000000000000000
Rx Oversize:                 0000000000000000
Tx Fragments:                0000000000000000
Rx Fragments:                0000000000000000
Rx Jabber:                   0000000000000000
Rx Dropped:                  0000000000000000
HS ENET Device Statistics (DPS)
Rx Host Frames:              0000000000000000
Rx Iw Frames:                000000004123fea6
Rx Error Host Full:          0000000000000000
Rx Error Fbp Underrun:       0000000000000000
Rx Error Nonvalid Mac:       0000000000000000
Rx Error Mru:                0000000000000000
Rx Error Sdu:                0000000000000000
Tx Error Underrun:           0000000000000000
Rx Error Overrun:            0000000000000000
Tx Frames:                   000000004123fea7
-------------------------------------------------
# Enter your command:
```


Figure 43: Enet device and port statistics

```
**********************************************************************
          SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Statistics
**********************************************************************
1.  -> Device statistics
2.  -> Channel statistics
3.  -> Flow Aggregation statistics
4.  -> PSN->TDM IW port statistics
5.  -> Iw Global Statistics [port_type(0-TRANS port,1-ENET1 port)]
6.  -> Memory Status
7.  -> Pwe3StatisticsClear [optional: highest port number]
8.  -> Pwe3StatisticsCheck [optional: check RX TS (0-uncheck, 1-check), optional: highest port number, optional: global display - 1 ]
9.  -> Global statistics Check [optional: clear statistics (0-don't clear, 1-clear), optional: iterations (1-10)]
# Enter your command:3


**********************************************************************
          SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Flow Aggregation Statistics
**********************************************************************
1.  -> TDM to PSN Flow Aggregation statistics [flowagg_num, type(0-DATA,1-CAS)]
2.  -> PSN to TDM Flow Aggregation statistics [flowagg_num]
3.  -> L2 ROUTER MPLS Flow Aggregation statistics [flowagg_num, dir(0-tdm2psn, 1-psn2tdm)]
4.  -> Clear L2 ROUTER MPLS Flow Aggregation statistics [flowagg_num, dir(0-tdm2psn, 1-psn2tdm)]
```

Figure 44: PSN to TDM flow aggregation statistics menu

```
FLOW AGGREGATION STATISTICS:

PSN TO TDM FLOW AGGREGATION 250 ( FLOW AGGREGATION HANDLE 34c00219 ):
-------------------------------------------------------------

forwarded_packets:        0000000000007c068 fbp_drop_packets:              0000000000000000
out_of_window_packets:    0000000000000000 buffer_overun_dropped_packets: 0000000000000000
window_switchover:        0000000000000000 buffer_overun_events:          00000000000008a7
stray_packets:            0000000000000000 malformed_packets:             0000000000000000
cw_ais_drop_packets:      0000000000000000 multiple_packets:              0000000000000000
mpls_drop_packets:        0000000000000000 denied_packets:                0000000000000000
out_of_sequence_packets: 0000000000000000 out_of_band_cas_packets:       0000000000000000
rai_packets:              0000000000000000 rdi_dropped_packets:           0000000000000000
```

Figure 45: PSN to TDM flow aggregation statistics

```
**********************************************************************
          SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Statistics
**********************************************************************
1.  -> Device statistics
2.  -> Channel statistics
3.  -> Flow Aggregation statistics
4.  -> PSN->TDM IW port statistics
5.  -> Iw Global Statistics [port_type(0-TRANS port,1-ENET1 port)]
6.  -> Memory Status
7.  -> Pwe3StatisticsClear [optional: highest port number]
8.  -> Pwe3StatisticsCheck [optional: check RX TS (0-uncheck, 1-check), optional: highest port number, optional: global display - 1 ]
9.  -> Global statistics Check [optional: clear statistics (0-don't clear, 1-clear), optional: iterations (1-10)]
# Enter your command:2


**********************************************************************
          SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Channel Statistics
**********************************************************************
1.  -> PWE3 Rx Channel statistics [pw_index]
2.  -> PWE3 Tx Channel statistics [pw_index]
3.  -> Clear PWE3 Rx Channel statistics [pw_index]
4.  -> Clear PWE3 Tx Channel statistics [pw_index]
# Enter your command:2 4
```

Figure 46: PSN to TDM EMPHY channel menu

```
CHANNEL STATISTICS:

TRANSPARENT PWE3 TX CHANNEL 4 ( CHANNEL HANDLE 9e00010 ):
-----------------------------------------------------

dummy_packet:          0000000000000000            underrun_dataunit:       0000000000000042
buffer_underrun:       0000000000000001            valid_packets:           0000000000091c26
Number of packets in queue: 0x0e (min 0x0e,    max 0x0f)
```

Figure 47: PSN to TDM EMPHY channel statistics

3.7.4.3 Detailed Clock Recovery Statistics

```
*************************************************************************
            SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Clock recovery utils
*************************************************************************
1.  -> view number of packets in queue[pw_index]
2.  -> Force holdover state [pw_index]
3.  -> Clear holdover state [pw_index]
4.  -> Show clock recovery systems states
5.  -> Show clock recovery state [pw_index]
6.  -> Enable clock recovery interface [pw_index]
7.  -> Disable clock recovery interface [pw_index]
8.  -> Clock Recovery statistics [cr table index]
9.  -> Clock Recovery statistics reset [cr table index]
10.  -> Clock Recovery Systems statistics Status
11.  -> Clock Recovery Systems statistics Reset
12.  -> Clock Recovery Command [pw_index, command (0-reset, 1-force freerun)]
13.  -> Clock Recovery Indirect Memory Display
14.  -> Clock Recovery Calculate DCO parameters [service clock, diff_ref clock, ppm_offset]
15.  -> Clock Recovery RX Phy reconfigure [pw_index, enable CR '0' (0-disable, 1-enable)]
16.  -> Configure WDS3 board pins in FPGA [config (1-default, 2-ptp, 3-SyncE), board_number (1,2)]
17.  -> Clock Recovery DCO HW Registers Display [DCO id]
18.  -> Global Clock Recovery State [optional: iteration (1-10)]
19.  -> Clock Recovery PPB display [optional: ppb_offset ppb_margins display_all]
20.  -> Set timestamp stub (dedicated synthesis required) [1-enable 0-disable]
# Enter your command:4
```


Figure 48: Clock Recovery system states menu

```
Clock Recovery Systems states: <PW> <State>
PW   0:NORMAL   PW   1:NORMAL   PW   2:NORMAL   PW   3:NORMAL   PW   4:NORMAL   PW   5:NORMAL   PW   6:NORMAL
PW   7:NORMAL   PW   8:NORMAL   PW   9:NORMAL   PW  10:NORMAL   PW  11:NORMAL   PW  12:NORMAL   PW  13:NORMAL
PW  14:NORMAL   PW  15:NORMAL   PW  16:NORMAL   PW  17:NORMAL   PW  18:NORMAL   PW  19:NORMAL   PW  20:NORMAL
PW  21:NORMAL   PW  22:NORMAL   PW  23:NORMAL   PW  24:NORMAL   PW  25:NORMAL   PW  26:NORMAL   PW  27:NORMAL
PW  28:NORMAL   PW  29:NORMAL   PW  30:NORMAL   PW  31:NORMAL   PW  32:NORMAL   PW  33:NORMAL   PW  34:NORMAL
PW  35:HOLDOVER PW  36:HOLDOVER PW  37:HOLDOVER PW  38:NORMAL   PW  39:HOLDOVER PW  40:NORMAL   PW  41:NORMAL
PW  42:HOLDOVER PW  43:NORMAL   PW  44:HOLDOVER PW  45:HOLDOVER PW  46:HOLDOVER PW  47:HOLDOVER PW  48:NORMAL
PW  49:HOLDOVER PW  50:HOLDOVER PW  51:HOLDOVER PW  52:HOLDOVER PW  53:HOLDOVER PW  54:HOLDOVER PW  55:HOLDOVER
PW  56:HOLDOVER PW  57:HOLDOVER PW  58:HOLDOVER PW  59:HOLDOVER PW  60:HOLDOVER PW  61:HOLDOVER PW  62:HOLDOVER
PW  63:HOLDOVER PW  64:HOLDOVER PW  65:HOLDOVER PW  66:HOLDOVER PW  67:HOLDOVER PW  68:HOLDOVER PW  69:HOLDOVER
PW  70:NORMAL   PW  71:HOLDOVER PW  72:HOLDOVER PW  73:NORMAL   PW  74:HOLDOVER PW  75:NORMAL   PW  76:HOLDOVER
PW  77:NORMAL   PW  78:NORMAL   PW  79:NORMAL   PW  80:NORMAL   PW  81:NORMAL   PW  82:NORMAL   PW  83:NORMAL
PW  84:NORMAL   PW  85:NORMAL   PW  86:NORMAL   PW  87:NORMAL   PW  88:NORMAL   PW  89:NORMAL   PW  90:NORMAL
PW  91:NORMAL   PW  92:NORMAL   PW  93:NORMAL   PW  94:NORMAL   PW  95:NORMAL   PW  96:NORMAL   PW  97:NORMAL
PW  98:NORMAL   PW  99:NORMAL   PW 100:NORMAL   PW 101:NORMAL   PW 102:NORMAL   PW 103:NORMAL   PW 104:NORMAL
PW 105:NORMAL   PW 106:NORMAL   PW 107:NORMAL   PW 108:NORMAL   PW 109:NORMAL   PW 110:NORMAL   PW 111:NORMAL
PW 112:NORMAL   PW 113:NORMAL   PW 114:NORMAL   PW 115:NORMAL   PW 116:NORMAL   PW 117:NORMAL   PW 118:NORMAL
PW 119:NORMAL   PW 120:NORMAL   PW 121:NORMAL   PW 122:NORMAL   PW 123:NORMAL   PW 124:NORMAL   PW 125:NORMAL
PW 126:NORMAL   PW 127:NORMAL   PW 128:NORMAL   PW 129:NORMAL   PW 130:NORMAL   PW 131:NORMAL   PW 132:NORMAL
PW 133:NORMAL   PW 134:NORMAL   PW 135:NORMAL   PW 136:NORMAL   PW 137:NORMAL   PW 138:NORMAL   PW 139:NORMAL
PW 140:NORMAL   PW 141:NORMAL   PW 142:NORMAL   PW 143:NORMAL   PW 144:NORMAL   PW 145:NORMAL   PW 146:NORMAL
PW 147:NORMAL   PW 148:NORMAL   PW 149:NORMAL   PW 150:NORMAL   PW 151:NORMAL   PW 152:NORMAL   PW 153:NORMAL
PW 154:NORMAL   PW 155:NORMAL   PW 156:NORMAL   PW 157:NORMAL   PW 158:NORMAL   PW 159:NORMAL   PW 160:NORMAL
PW 161:NORMAL   PW 162:NORMAL   PW 163:NORMAL   PW 164:NORMAL   PW 165:NORMAL   PW 166:NORMAL   PW 167:NORMAL
PW 168:NORMAL   PW 169:NORMAL   PW 170:NORMAL   PW 171:NORMAL   PW 172:NORMAL   PW 173:NORMAL   PW 174:NORMAL
PW 175:NORMAL   PW 176:NORMAL   PW 177:NORMAL   PW 178:NORMAL   PW 179:HOLDOVER PW 180:NORMAL   PW 181:NORMAL
PW 182:HOLDOVER PW 183:NORMAL   PW 184:NORMAL   PW 185:NORMAL   PW 186:NORMAL   PW 187:NORMAL   PW 188:NORMAL
PW 189:HOLDOVER PW 190:NORMAL   PW 191:HOLDOVER PW 192:HOLDOVER PW 193:HOLDOVER PW 194:HOLDOVER PW 195:HOLDOVER
PW 196:HOLDOVER PW 197:HOLDOVER PW 198:HOLDOVER PW 199:HOLDOVER PW 200:HOLDOVER PW 201:HOLDOVER PW 202:HOLDOVER
PW 203:HOLDOVER PW 204:HOLDOVER PW 205:HOLDOVER PW 206:HOLDOVER PW 207:HOLDOVER PW 208:HOLDOVER PW 209:HOLDOVER
PW 210:HOLDOVER PW 211:HOLDOVER PW 212:NORMAL   PW 213:HOLDOVER PW 214:HOLDOVER PW 215:HOLDOVER PW 216:HOLDOVER
PW 217:HOLDOVER PW 218:HOLDOVER PW 219:HOLDOVER PW 220:NORMAL   PW 221:HOLDOVER PW 222:HOLDOVER PW 223:NORMAL
PW 224:NORMAL   PW 225:NORMAL   PW 226:NORMAL   PW 227:NORMAL   PW 228:NORMAL   PW 229:NORMAL   PW 230:NORMAL
PW 231:NORMAL   PW 232:NORMAL   PW 233:NORMAL   PW 234:NORMAL   PW 235:NORMAL   PW 236:NORMAL   PW 237:NORMAL
PW 238:NORMAL   PW 239:NORMAL   PW 240:NORMAL   PW 241:NORMAL   PW 242:NORMAL   PW 243:NORMAL   PW 244:NORMAL
PW 245:NORMAL   PW 246:NORMAL   PW 247:NORMAL   PW 248:NORMAL   PW 249:NORMAL   PW 250:NORMAL   PW 251:NORMAL
```

Figure 49: Clock Recovery system states

```
************************************************************************
            SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Clock recovery utils
************************************************************************
1.   -> view number of packets in queue[pw_index]
2.   -> Force holdover state [pw_index]
3.   -> Clear holdover state [pw_index]
4.   -> Show clock recovery systems states
5.   -> Show clock recovery state [pw_index]
6.   -> Enable clock recovery interface [pw_index]
7.   -> Disable clock recovery interface [pw_index]
8.   -> Clock Recovery statistics [cr table index]
9.   -> Clock Recovery statistics reset [cr table index]
10.   -> Clock Recovery Systems statistics Status
11.   -> Clock Recovery Systems statistics Reset
12.   -> Clock Recovery Command [pw_index, command (0-reset, 1-force freerun)]
13.   -> Clock Recovery Indirect Memory Display
14.   -> Clock Recovery Calculate DCO parameters [service clock, diff_ref clock, ppm_offset]
15.   -> Clock Recovery RX Phy reconfigure [pw_index, enable CR '0' (0-disable, 1-enable)]
16.   -> Configure WDS3 board pins in FPGA [config (1-default, 2-ptp, 3-SyncE), board_number (1,2)]
17.   -> Clock Recovery DCO HW Registers Display [DCO id]
18.   -> Global Clock Recovery State [optional: iteration (1-10)]
19.   -> Clock Recovery PPB display [optional: ppb_offset ppb_margins display_all]
20.   -> Set timestamp stub (dedicated synthesis required) [1-enable 0-disable]
# Enter your command:8 5
```

Figure 50: Clock Recovery line statistics menu

```
---- CLOCK RECOVERY STATISTICS FOR TABLE 5 ----
-------------------------------------------------

reset events:                    0000000000000248
holdover period expired events:  0000000000000000
invalid input timestamp:         000000000000003d
invalid output timestamp:        0000000000000000
invalid phase gap:               0000000000000000
service to nominal clock ppb:                 639
socket to diff clock ppb:                     640
socket to XO clock ppb:                       640
```

Figure 51: Clock Recovery line statistics

```
**************************************************************************
            SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Clock recovery utils
**************************************************************************
1.  -> view number of packets in queue[pw_index]
2.  -> Force holdover state [pw_index]
3.  -> Clear holdover state [pw_index]
4.  -> Show clock recovery systems states
5.  -> Show clock recovery state [pw_index]
6.  -> Enable clock recovery interface [pw_index]
7.  -> Disable clock recovery interface [pw_index]
8.  -> Clock Recovery statistics [cr table index]
9.  -> Clock Recovery statistics reset [cr table index]
10.  -> Clock Recovery Systems statistics Status
11.  -> Clock Recovery Systems statistics Reset
12.  -> Clock Recovery Command [pw_index, command (0-reset, 1-force freerun)]
13.  -> Clock Recovery Indirect Memory Display
14.  -> Clock Recovery Calculate DCO parameters [service clock, diff_ref clock, ppm_offset]
15.  -> Clock Recovery RX Phy reconfigure [pw_index, enable CR '0' (0-disable, 1-enable)]
16.  -> Configure WDS3 board pins in FPGA [config (1-default, 2-ptp, 3-SyncE), board_number (1,2)]
17.  -> Clock Recovery DCO HW Registers Display [DCO id]
18.  -> Global Clock Recovery State [optional: iteration (1-10)]
19.  -> Clock Recovery PPB display [optional: ppb_offset ppb_margins display_all]
20.  -> Set timestamp stub (dedicated synthesis required) [1-enable 0-disable]
# Enter your command:10
Statistics of all CRS are zero - PASS
# Enter your command:
```

Figure 52: Clock Recovery system statistics status

```
************************************************************************
            SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Clock recovery utils
************************************************************************
1.  -> view number of packets in queue[pw_index]
2.  -> Force holdover state [pw_index]
3.  -> Clear holdover state [pw_index]
4.  -> Show clock recovery systems states
5.  -> Show clock recovery state [pw_index]
6.  -> Enable clock recovery interface [pw_index]
7.  -> Disable clock recovery interface [pw_index]
8.  -> Clock Recovery statistics [cr table index]
9.  -> Clock Recovery statistics reset [cr table index]
10.  -> Clock Recovery Systems statistics Status
11.  -> Clock Recovery Systems statistics Reset
12.  -> Clock Recovery Command [pw_index, command (0-reset, 1-force freerun)]
13.  -> Clock Recovery Indirect Memory Display
14.  -> Clock Recovery Calculate DCO parameters [service clock, diff_ref clock, ppm_offset]
15.  -> Clock Recovery RX Phy reconfigure [pw_index, enable CR '0' (0-disable, 1-enable)]
16.  -> Configure WDS3 board pins in FPGA [config (1-default, 2-ptp, 3-SyncE), board_number (1,2)]
17.  -> Clock Recovery DCO HW Registers Display [DCO id]
18.  -> Global Clock Recovery State [optional: iteration (1-10)]
19.  -> Clock Recovery PPB display [optional: ppb_offset ppb_margins display_all]
20.  -> Set timestamp stub (dedicated synthesis required) [1-enable 0-disable]
# Enter your command:11
Reset Statistics from CRS 0 to CRS 335
# Enter your command:
```

Figure 53: Clock Recovery system statistics status reset

```
************************************************************************
            SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Clock recovery utils
************************************************************************
1.  -> view number of packets in queue[pw_index]
2.  -> Force holdover state [pw_index]
3.  -> Clear holdover state [pw_index]
4.  -> Show clock recovery systems states
5.  -> Show clock recovery state [pw_index]
6.  -> Enable clock recovery interface [pw_index]
7.  -> Disable clock recovery interface [pw_index]
8.  -> Clock Recovery statistics [cr table index]
9.  -> Clock Recovery statistics reset [cr table index]
10.  -> Clock Recovery Systems statistics Status
11.  -> Clock Recovery Systems statistics Reset
12.  -> Clock Recovery Command [pw_index, command (0-reset, 1-force freerun)]
13.  -> Clock Recovery Indirect Memory Display
14.  -> Clock Recovery Calculate DCO parameters [service clock, diff_ref clock, ppm_offset]
15.  -> Clock Recovery RX Phy reconfigure [pw_index, enable CR '0' (0-disable, 1-enable)]
16.  -> Configure WDS3 board pins in FPGA [config (1-default, 2-ptp, 3-SyncE), board_number (1,2)]
17.  -> Clock Recovery DCO HW Registers Display [DCO id]
18.  -> Global Clock Recovery State [optional: iteration (1-10)]
19.  -> Clock Recovery PPB display [optional: ppb_offset ppb_margins display_all]
20.  -> Set timestamp stub (dedicated synthesis required) [1-enable 0-disable]
# Enter your command:19
```

Figure 54: Clock Recovery PPB Display menu

```
------------------------------------------------
---- CLOCK RECOVERY STATISTICS PPB ----
------------------------------------------------

socket to diff clock ppb:              624
socket to XO clock ppb:                624
------------------------------------------------
count/line/cr:service to nominal clock ppb:
000/000/000:   633  001/001/001:   633  002/002/002:   634  003/003/004:   634  004/004/005:   633  005/005/006:   632  006/006/008:   634
007/007/009:   633  008/008/010:   634  009/009/012:   634  010/010/013:   633  011/011/014:   634  012/012/016:   633  013/013/017:   632
014/014/018:   633  015/015/020:   633  016/016/021:   634  017/017/022:   634  018/018/024:   632  019/019/025:   634  020/020/026:   633
021/028/028:   634  022/029/029:   633  023/030/030:   632  024/031/032:   632  025/032/033:   634  026/033/034:   633  027/034/036:   633
028/035/037:   632  029/036/038:   634  030/037/040:   633  031/038/041:   633  032/039/042:   634  033/040/044:   633  034/041/045:   634
035/042/046:   635  036/043/048:   634  037/044/049:   633  038/045/050:   633  039/046/052:   634  040/047/053:   633  041/048/054:   633
042/056/056:   633  043/057/057:   633  044/058/058:   634  045/059/060:   633  046/060/061:   633  047/061/062:   634  048/062/064:   634
049/063/065:   633  050/064/066:   633  051/065/068:   634  052/066/069:   633  053/067/070:   635  054/068/072:   632  055/069/073:   633
056/070/074:   634  057/071/076:   633  058/072/077:   634  059/073/078:   634  060/074/080:   633  061/075/081:   633  062/076/082:   633
063/084/084:   633  064/085/085:   631  065/086/086:   633  066/087/088:   634  067/088/089:   633  068/089/090:   633  069/090/092:   634
070/091/093:   633  071/092/094:   633  072/093/096:   633  073/094/097:   633  074/095/098:   634  075/096/100:   632  076/097/101:   633
077/098/102:   633  078/099/104:   632  079/100/105:   633  080/101/106:   633  081/102/108:   632  082/103/109:   634  083/104/110:   632
084/112/112:   635  085/113/113:   632  086/114/114:   633  087/115/116:   632  088/116/117:   633  089/117/118:   635  090/118/120:   634
091/119/121:   632  092/120/122:   634  093/121/124:   634  094/122/125:   634  095/123/126:   633  096/124/128:   633  097/125/129:   633
098/126/130:   633  099/127/132:   635  100/128/133:   633  101/129/134:   632  102/130/136:   633  103/131/137:   633  104/132/138:   633
105/140/140:   633  106/141/141:   633  107/142/142:   633  108/143/144:   633  109/144/145:   633  110/145/146:   632  111/146/148:   631
112/147/149:   632  113/148/150:   632  114/149/152:   634  115/150/153:   632  116/151/154:   633  117/152/156:   633  118/153/157:   633
119/154/158:   633  120/155/160:   632  121/156/161:   634  122/157/162:   632  123/158/164:   634  124/159/165:   632  125/160/166:   634
126/168/168:   634  127/169/169:   631  128/170/170:   634  129/171/172:   632  130/172/173:   633  131/173/174:   633  132/174/176:   632
133/175/177:   633  134/176/178:   634  135/177/180:   633  136/178/181:   632  137/179/182:   633  138/180/184:   633  139/181/185:   633
140/182/186:   632  141/183/188:   632  142/184/189:   634  143/185/190:   633  144/186/192:   634  145/187/193:   633  146/188/194:   633
147/196/196:   632  148/197/197:   633  149/198/198:   634  150/199/200:   633  151/200/201:   632  152/201/202:   633  153/202/204:   632
154/203/205:   633  155/204/206:   634  156/205/208:   633  157/206/209:   633  158/207/210:   633  159/208/212:   632  160/209/213:   633
161/210/214:   634  162/211/216:   633  163/212/217:   633  164/213/218:   634  165/214/220:   632  166/215/221:   634  167/216/222:   634
168/224/224:   633  169/225/225:   632  170/226/226:   633  171/227/228:   632  172/228/229:   632  173/229/230:   633  174/230/232:   634
175/231/233:   632  176/232/234:   634  177/233/236:   632  178/234/237:   633  179/235/238:   633  180/236/240:   631  181/237/241:   633
182/238/242:   633  183/239/244:   633  184/240/245:   634  185/241/246:   633  186/242/248:   632  187/243/249:   633  188/244/250:   632
189/252/252:   632  190/253/253:   631  191/254/254:   632  192/255/256:   633  193/256/257:   633  194/257/258:   633  195/258/260:   632
196/259/261:   633  197/260/262:   635  198/261/264:   633  199/262/265:   633  200/263/266:   634  201/264/268:   635  202/265/269:   633
203/266/270:   633  204/267/272:   634  205/268/273:   635  206/269/274:   634  207/270/276:   634  208/271/277:   634  209/272/278:   634
210/280/280:   632  211/281/281:   633  212/282/282:   632  213/283/284:   633  214/284/285:   632  215/285/286:   633  216/286/288:   633
217/287/289:   633  218/288/290:   629  219/289/292:   632  220/290/293:   633  221/291/294:   634  222/292/296:   633  223/293/297:   633
224/294/298:   634  225/295/300:   633  226/296/301:   631  227/297/302:   631  228/298/304:   629  229/299/305:   632  230/300/306:   633
231/308/308:   633  232/309/309:   633  233/310/310:   630  234/311/312:   633  235/312/313:   633  236/313/314:   632  237/314/316:   633
238/315/317:   633  239/316/318:   633  240/317/320:   633  241/318/321:   633  242/319/322:   633  243/320/324:   633  244/321/325:   633
245/322/326:   633  246/323/328:   633  247/324/329:   635  248/325/330:   633  249/326/332:   633  250/327/333:   633  251/328/334:   633
```

Figure 55: Clock Recovery PPB Display menu

3.7.5. Clock Recovery and General System Displays

Following are several examples of the different displays the PWE3 CES Demo includes. There are many different displays that can be used during a debug process, all of which may help solve many of the problems that can arise.

3.7.5.1 Clock Recovery

```
********************************************************************************
            SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Main Menu
********************************************************************************
1.    -> System setup
2.    -> Ufe setup
3.    -> PW add
4.    -> PW delete
5.    -> PW enable
6.    -> PW disable
7.    -> PW modify
8.    -> PW display
9.    -> System display
10.   -> Predefined configurations
11.   -> Statistics
12.   -> Host interface
13.   -> Debug utilities
14.   -> Clock Recovery
15.   -> Memory Display
16.   -> Quit without reset
17.   -> Quit
# Enter your command:9


********************************************************************************
            SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: System display
********************************************************************************
1.   -> System Display: Slots assignment table [line_index (0-84)]
2.   -> System Display: TDM to PSN IW system
3.   -> System Display: PSN to TDM IW system
4.   -> System Display: PSN to TDM Rx binding
5.   -> System Display: CFU display
6.   -> System Display: Clock Recovery (0-simple, 1-full)
# Enter your command:6
```


Figure 56: System Display: Clock Recovery menu

```
CESOP PORT INTERFACE = UFE:
CESOP CLOCK RECOVERY REGISTRY:
    logic_base: 0x1f212000
    clock_rec_portsize: 0x00000001
    clock_rec_bustype: 0x00000003
    clock_rec_interfaces: 0x00000540
    max_clock_rec_ext_diff_interfaces: 0x00000000
    DPS clock_rec_base: 0x1ec67500
    DPS host_clock_rec_base: 0xbec67500
    DPS ext_clock_rec_base: 0x1ec91500
    DPS host_ext_clock_rec_base: 0xbec91500
    DPS stats_base: 0x1c1bd000
    DPS host_stats_base: 0xbc1bd000
    DPS cr_rtpt_base: 0x1ec9bd00
    DPS host_cr_rtpt_base: 0xbec9bd00
    DPS cr_ts_buffer_base: 0x1eca1100
    DPS host_cr_ts_buffer_base: 0xbeca1100
    input_devices_num: 0x00000001
    sync_delay: 0x00000000
    dco_clock_selection: 0x00000000
    sync_mode: 0x00000000
    sync_active_mode: 0x00000000
    diff_ref_clock: 0x07735940
    ext_dco_clock: 0x00000000
    reg_dco_clock: 0x07735940

TX TIMESTAMP GLOBAL TABLE:
    Control Register 1: 0xf977d93f
        Time Stamp Ratio Mode: 0x00000003
        Port Size Factor: 0x00000003
        Parameter Bus Type: 0x00000003
    Time Stamp Counter Base: 0x1fdacc81

RATIO TIMESTAMP TABLE:
    Differential-Socket Delta Timestamp      : 0x03b9ac7b
    Differential-Socket Previous Timestamp   : 0x7ef853f7
    Oscilator-Socket Delta Timestamp         : 0x03b9ac7b
    Oscilator-Socket Previous Timestamp      : 0x7ef8547a
    Invalid PHY number Counter: 0x00000000

# Enter your command:
```


Figure 57: System Display: Clock Recovery – simple

3.7.5.2 System Display

```
*******************************************************************************
            SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Main Menu
*******************************************************************************
1.    -> System setup
2.    -> Ufe setup
3.    -> PW add
4.    -> PW delete
5.    -> PW enable
6.    -> PW disable
7.    -> PW modify
8.    -> PW display
9.    -> System display
10.   -> Predefined configurations
11.   -> Statistics
12.   -> Host interface
13.   -> Debug utilities
14.   -> Clock Recovery
15.   -> Memory Display
16.   -> Quit without reset
17.   -> Quit
# Enter your command:9


*******************************************************************************
            SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: System display
*******************************************************************************
1.   -> System Display: Slots assignment table [line_index (0-84)]
2.   -> System Display: TDM to PSN IW system
3.   -> System Display: PSN to TDM IW system
4.   -> System Display: PSN to TDM Rx binding
5.   -> System Display: CFU display
6.   -> System Display: Clock Recovery (0-simple, 1-full)
# Enter your command:2
```

Figure 58: System Display: TDM to PSN menu

```
TDM TO PSN IW SYSTEM INFO:

TDM TO PSN CESOP IW SYSTEM ( SYSTEM HANDLE 40c00000 ):
-------------------------------------------------------

SYSTEM PARAMETERS:
buffer_gap: 64
 System handle: 0x40c00000
     Type: WP_IW_CESOP_MODE
         CESOP port interface = UFE:
         Buffer gap: 64
# Enter your command:
```

Figure 59: System Display: TDM to PSN

```
PSN TO TDM IW SYSTEM INFO:

PSN TO TDM MPLS ROUTING IW SYSTEM ( SYSTEM HANDLE 40300000 ):
-----------------------------------------------------------

SYSTEM PARAMETERS:
max_flows: 1349
buffer_gap: 64
edge_out_system: 0
 System handle: 0x40300000
     Type: WP_IW_MPLS_ROUTING_MODE
         Max flows: 1349
         Buffer gap: 64
# Enter your command:
```

Figure 60: System Display: PSN to DTM

```
*****************************************************************************
            SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Main Menu
*****************************************************************************
1.    -> System setup
2.    -> Ufe setup
3.    -> PW add
4.    -> PW delete
5.    -> PW enable
6.    -> PW disable
7.    -> PW modify
8.    -> PW display
9.    -> System display
10.   -> Predefined configurations
11.   -> Statistics
12.   -> Host interface
13.   -> Debug utilities
14.   -> Clock Recovery
15.   -> Memory Display
16.   -> Quit without reset
17.   -> Quit
# Enter your command:8


*****************************************************************************
            SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: PW display
*****************************************************************************
1.   -> PW Display: EMPHY transparent device [line_index][pw_index]
2.   -> PW Display: PWE3 Transparent channel [type(0-Rx,1-Tx),pw_index]
3.   -> PW Display: TDM to PSN flow aggregation [pw_index]
4.   -> PW Display: PSN to TDM flow aggregation [pw_index]
5.   -> PW Display: TDM to PSN Rx binding [pw_index]
6.   -> PW Display: PSN to TDM Tx binding [pw_index]
# Enter your command:1 0 0
```


Figure 61: PW Display: EMPHY Device menu

```
*************************************************************************
            SAToP CESoPSN Demo
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: PW display
*************************************************************************
1.   -> PW Display: EMPHY transparent device [line_index][pw_index]
2.   -> PW Display: PWE3 Transparent channel [type(0-Rx,1-Tx),pw_index]
3.   -> PW Display: TDM to PSN flow aggregation [pw_index]
4.   -> PW Display: PSN to TDM flow aggregation [pw_index]
5.   -> PW Display: TDM to PSN Rx binding [pw_index]
6.   -> PW Display: PSN to TDM Tx binding [pw_index]
# Enter your command:2 0 0
```

Figure 62: PW Display: PWE3 Transparent Channel (Rx) menu

```
PWE3 TRANSPARENT CHANNEL INFO:

PWE3 TRANSPARENT RX CHANNEL 0 ( CHANNEL HANDLE 5e0000b ):
-------------------------------------------------------

CHANNEL PARAMETERS:

num_slots: 32
rx_buffersize: 256

 Channel handle: 0x05e0000b
     Mode: WP_CH_RX
     Tag: 0
     Protocol: WP_TRANSPARENT_PWE3
 wpi_res_channel_trans_pwe3: invalid pointer value is 0x833cce30
 DPS Emphy Transparent PWE3 Channel RCPT:
     IW Buffer size: 0x00000300
     IWBRS: 0x00000200
     BD ptr: 0x201d1600
     Data Buffer ptr: 0x345aa900
     Control Bits 1: 0x00000184
     RRP: 0x00400000
     LAST CAS Pointer: 0x345aa82c
     TS config information: 0x00007800
         Timing Index: 0x00000000
         TS Buffer Entry Integer: 0x00000007
         Timing Mode Indication: 0x00000000
         LeapSize Remainder: 0x00000000
         TS Buffer Entry Remainder: 0x00000000
     Control Bits 2: 0x24b80024
     IW Gap, Fbp, Buffer Length: 0x40900100
     SyErr: 0
     FRG: 0
     Fetch BD: 0
     CAS Size: 0
     IWM: 1
     CAS: 0
     DBT: 1
     STE: 1
     IW Buffer Gap: 64
     FBP Num: 9
     LeapSize Integer: 0
     Buffer Length: 256
# Enter your command:
```


Figure 63: PW Display: PWE3 Transparent Channel (Rx)

The following example shows a partial view of the PSN to TDM Flow Aggregation display. During proper operation the 'Invalid RX TS Counter' (marked in yellow) should not be increasing as the delta between time stamps remains roughly constant. Furthermore, following this field is a table displaying the time stamps of the entering packets (marked in red). The values in the table should have an equal difference for every two consecutive values.

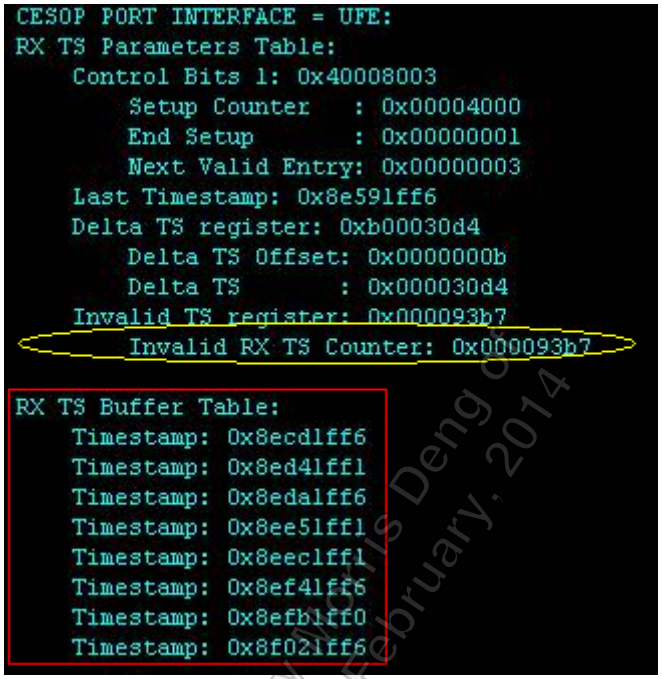


Figure 64: PW Display: TDM to PSN Flow Aggregation

As a part of the overall information available on the PSN to TDM Flow Aggregation there is the clock recovery table. This is a partial image of the entire display showing the clock recovery table (first image) and extension table (second image), both are very useful for debug information.

```
CESOP CLOCK RECOVERY TABLE:
    Control Bits 1: 0x10c063a9
        Output Ratio: 0x00000004
        Holdover Period Expired: 0x00000000
        Force Holdover: 0x00000000
        Statistics Enable: 0x00000001
        Interrupt mode: 0x00000001
        Interrupt Queue: 0x00000001
        Debug Enable: 0x00000000
        ALPHA: 0x00000006
        GAMA: 0x0000000e
        Average Divisor Low Factor: 0x0000000a
    Holdover Interrupt Register: 0x80000000
        Holdover Interrupt Counter: 0x00000000
        Holdover Interrupt Limit: 0x00000010
    Tx Counter Register: 0x08000045
        State: 0x00000004
        Tx Counter: 0x00000002
        Tx Counter Limit: 0x00000005
    Dco Used Bit Factor: 0x00000001
    Current Input Time Stamp: 0x56684c81
    Current Output Time Stamp: 0x65a289bb
    Last Divisor Low: 0x08fd9c0f
    Average Divisor Low: 0x08fd9b7c
    Accumulated Phase Error High: 0x00000000
    Accumulated Phase Error Middle: 0x00000000
    Accumulated Phase Error Low: 0x00000000
    Dummy Counter Register: 0x00000000
        Dummy Counter: 0x00000000
        Dummy Event Limit: 0x00000000
    Cai High: 0x00000000
    Cai Middle: 0x00000002
    Cai Low: 0x9472502e
    Control Bits 2: 0x00000207
        In Convergence: 0x00000000
        First Phase Error Flag: 0x00000000
        Freq Sign: 0x00000000
        Last Freq Sign: 0x00000000
        Holdover Expired Interrupt Assert bit: 0x00000000
        Clock Recovery Method: Differential
        Clock Recovery loop mode: With Feedback
        Setup Counter: 0x00000000
        End Setup: 0x00000001
        In Holdover: 0x00000000
        Port Size Factor: 0x00000001
        DCO Bus Type: 0x00000003
    Phase Gap: 0xfffe3778
    UPE phy number: 0x00000000
    HO update counter: 0x00000006
    Last Div High: 0x0000003d
    Last Div Low: 0x08fd9c0f
    Output TS High: 0x65b07203
    Output TS Low: 0xd2377800
    Updates Factor Numerator: 0x000000f3
    Updates Factor Denominator: 0x00006400
    Service Per TS: 0x00000800
    N Socket: 0x000025f8
```


Figure 65: PW Display: PSN to TDM Flow Aggregation – clock recovery table

```
CESOP CLOCK RECOVERY EXTENSION TABLE:
    Nominal Divisor Integer: 0x0000003d
    Nominal Divisor Remainder: 0x09000000
    Clock Recovery Base Address: 0x1f212000
    Divisor Reminder Limit: 0x01900a3d
    Num samples factor: 0x00008637
    Service clocks per packet: 0x00001000
    Delta timestamp limits and Differential Reference Clock Ratio Register: 0x1b211000
        Delta timestamp min value: 0x0000001b
        Delta timestamp max value: 0x00000021
        DRC ratio Valid Bit: 0x00000000
        DRC ratio int value: 0x00000001
        DRC ratio fractional high value: 0x00000000
        DRC ratio fractional low value: 0x00000000

# Enter your command:
```

Figure 66: PW Display: PSN to TDM Flow Aggregation – extension table

3.7.6. UFE4 OC3/12 –Adaptive / Differential CR Mode

In order to run a UFE4 adaptive or regular differential CR test, set the following defines:

#define WTI_CESOP_TDI 0

#define WTI_CESOP_CLOCK_RECOVERY_ENABLE 1

#define WTI_CESOP_RX_TIMING_ENABLE 1

#define WTI_CESOP_REGRESSION_TEST 0

#define WTI_CLOCK_REC_MODE 1 (for differential or '0' for adaptive)

If working in WP3, select the DCO clock source by setting the following define:

Available tests in this case are all CLI_F_CR_1xx tests

Example of UFE4 board configuration for WP3 shown in the following figures:

3.7.6.1 WDS3 – UFE412 4xOC3/12 –Adaptive / Differential CR Setup.

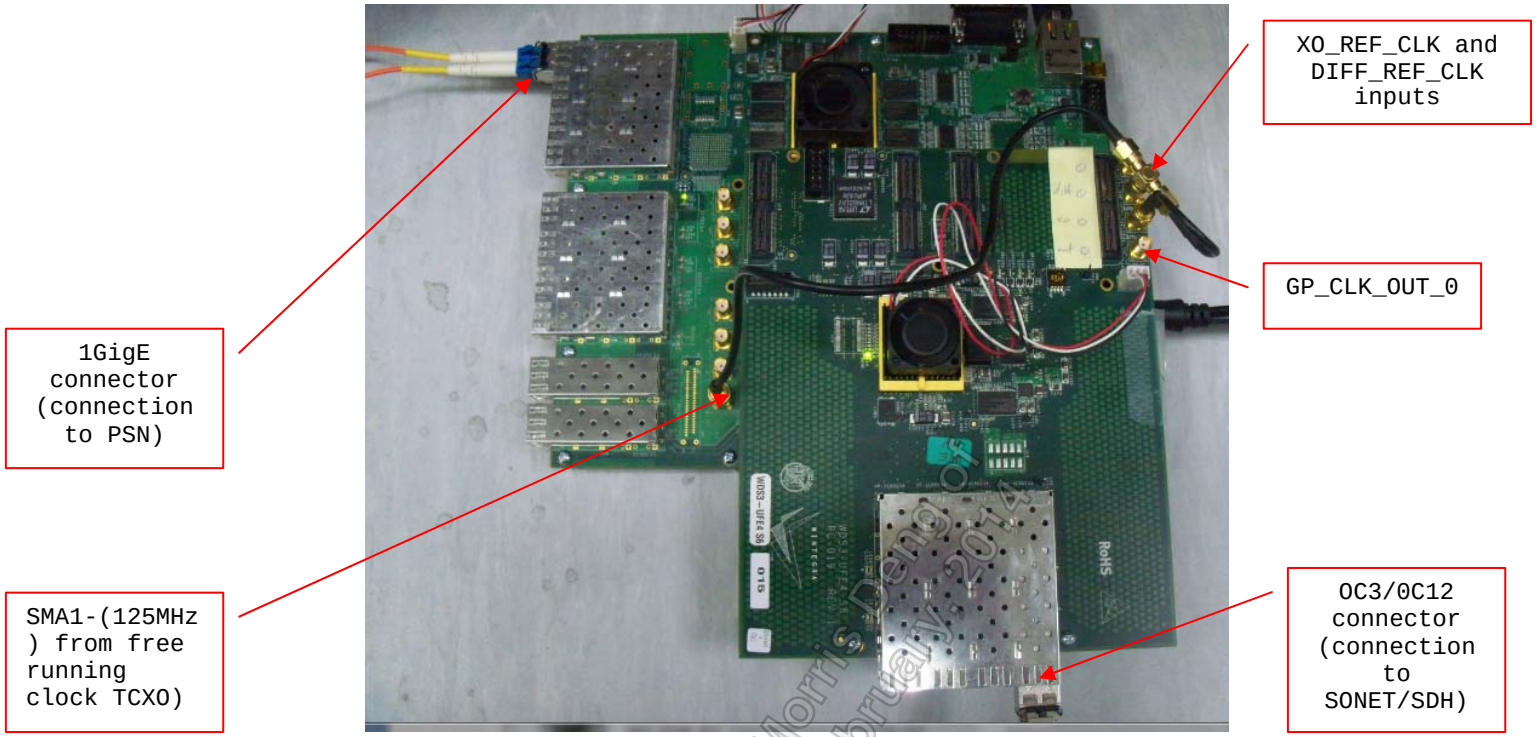


Figure 67: WDS3 and UFE412 setting

By default the 125MHz clock from WDS3-SMA1 is connected to both XO_REF_CLK and DIFF_REF_CLK inputs using a T connector.

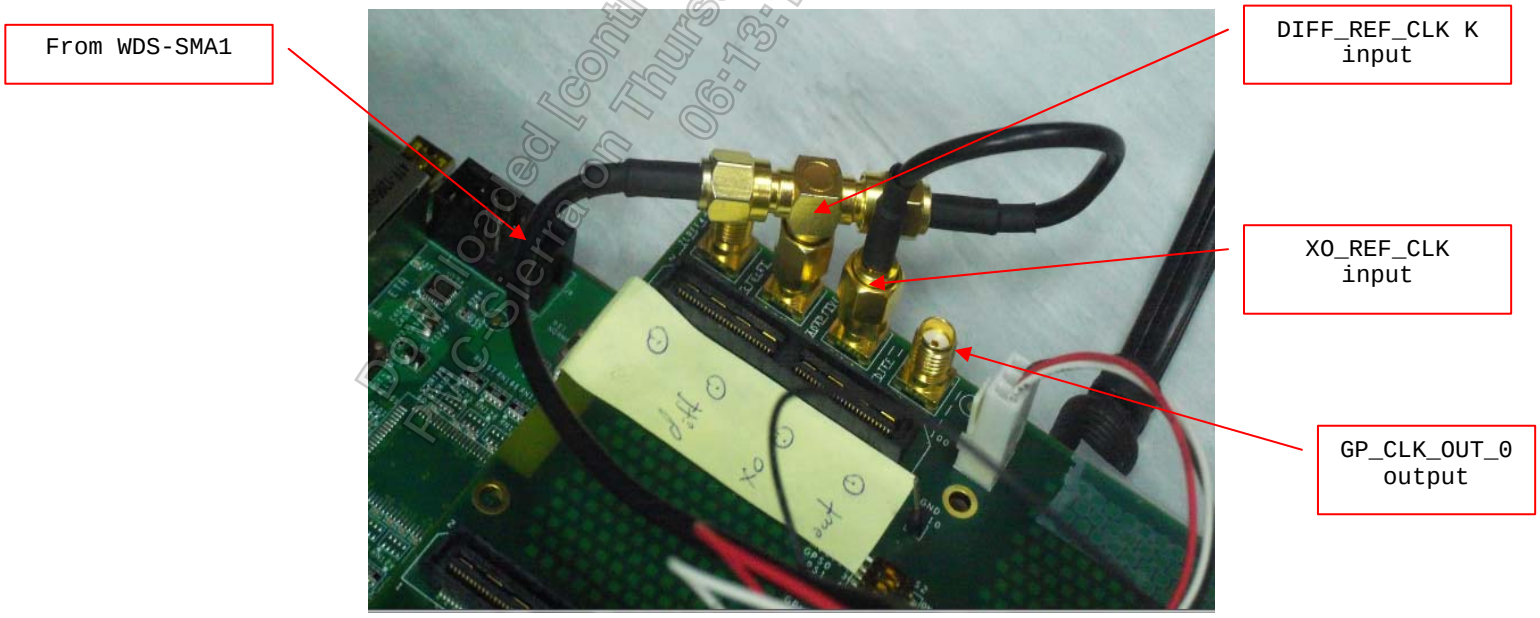


Figure 68: Clocks connectivity for UFE412-CR testing

3.7.6.2 WDS3 – UFE448 4xOC12 –Adaptive / Differential CR Setup.

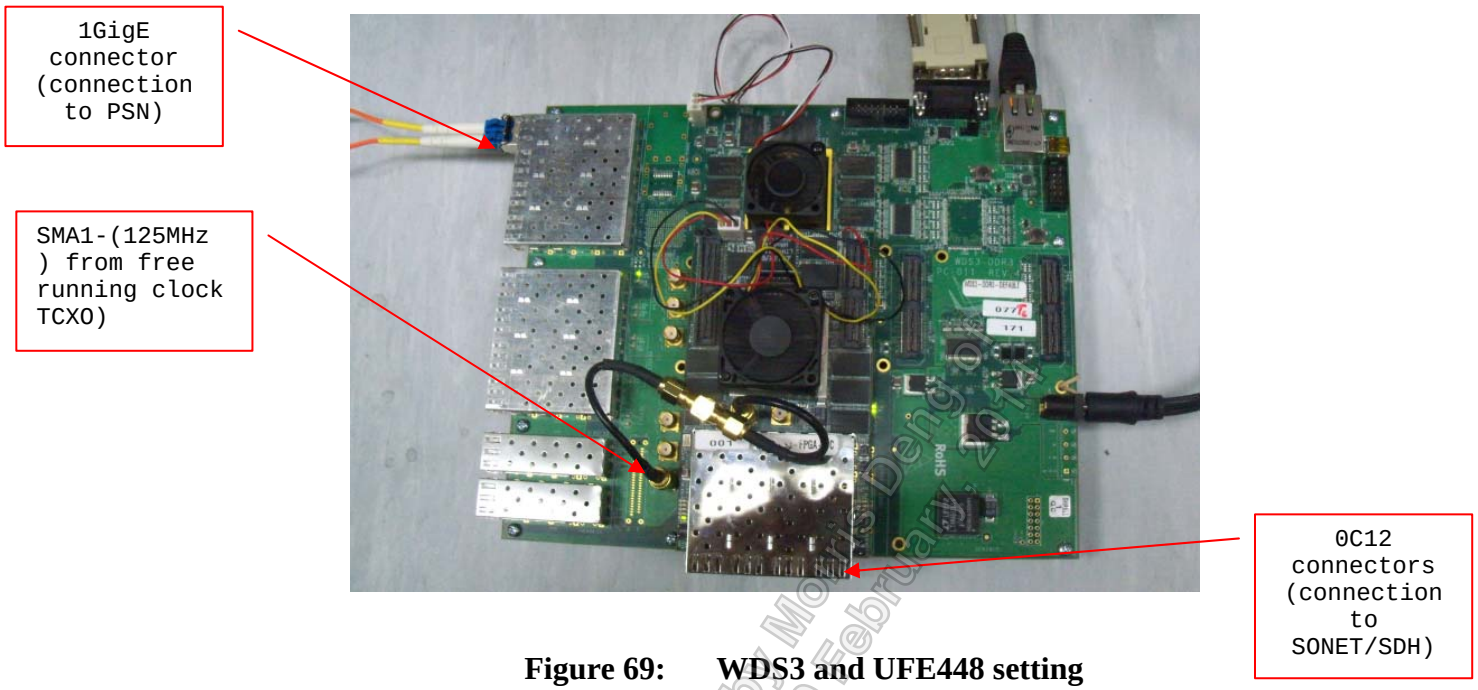
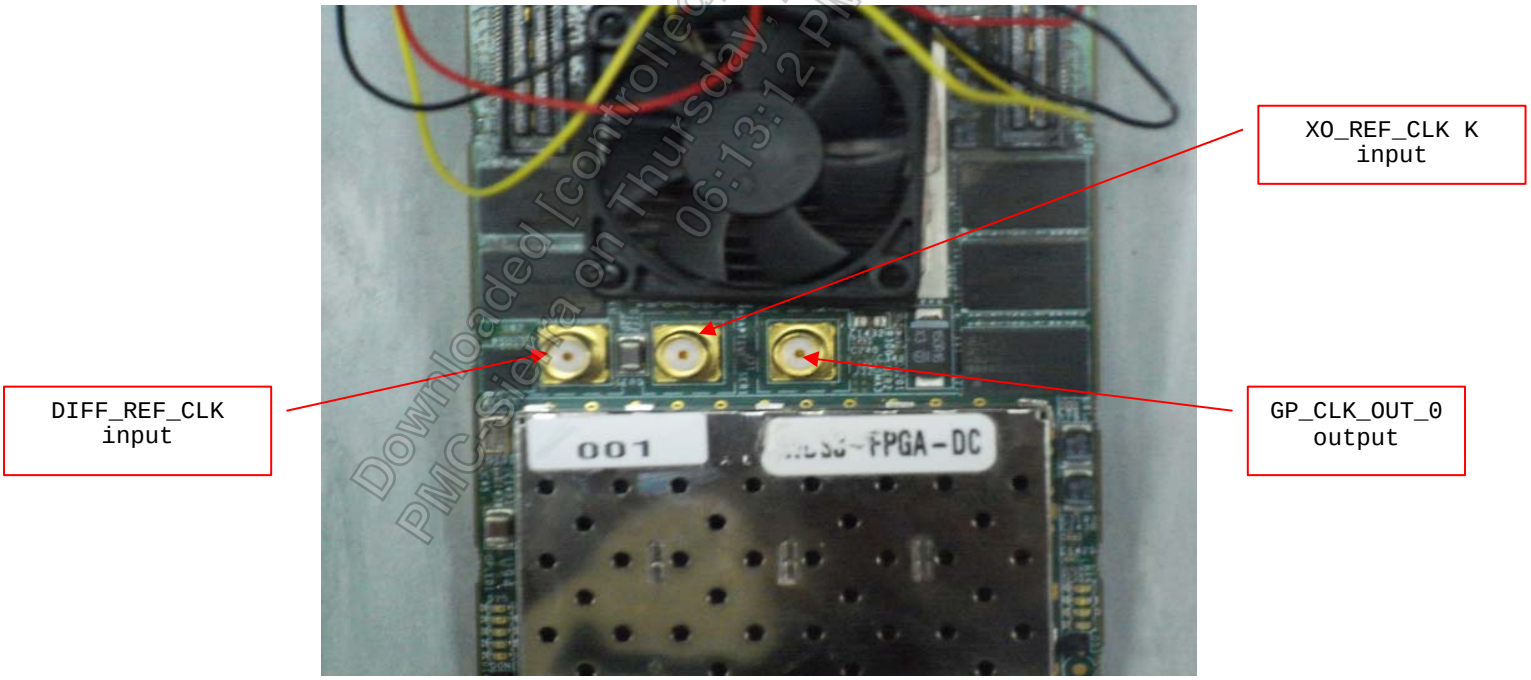


Figure 69: WDS3 and UFE448 setting



By default the 125MHz clock from WDS3-SMA1 is connected to both XO_REF_CLK and DIFF_REF_CLK inputs using a T connector.

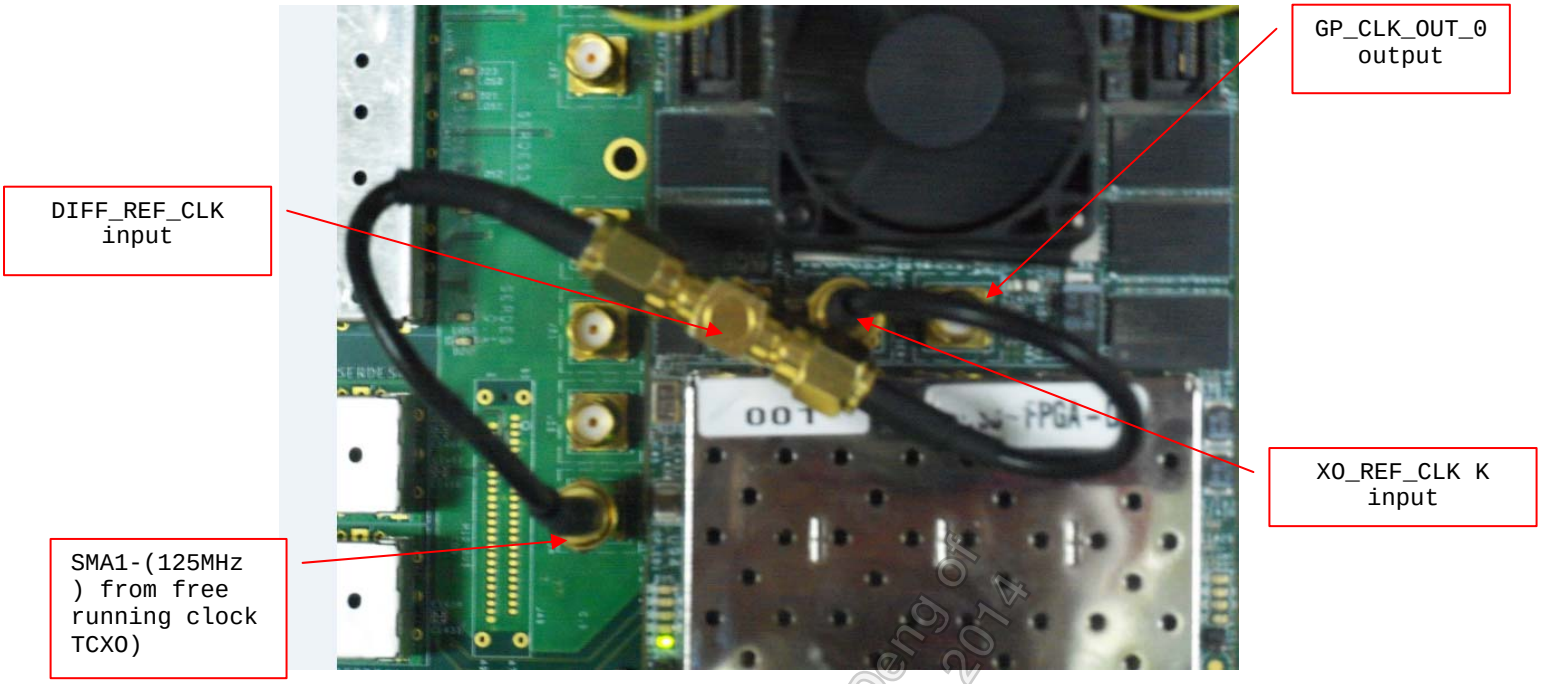


Figure 70: Clocks connectivity for UFE448-CR testing

3.7.7. UFE3 OC3/12 –SBI Adaptive or Regular Differential CR Mode

In order to run a SBI adaptive or regular differential CR test, set the following defines:

#define WTI_CESOP_TDI 0

#define WTI_CESOP_CLOCK_RECOVERY_ENABLE 1

#define WTI_CESOP_RX_TIMING_ENABLE 1

#define WTI_CESOP_REGRESSION_TEST 0

#define WTI_CLOCK_REC_MODE 1 (for differential or ‘0’ for adaptive)

#define WTI_EXT_DIFF_MODE_ACTIVE 0

#define WTI_ACTIVE_CR_FOR_ASYNC_MODE 32

If working in WP3, select the DCO clock source by setting the following define:

#define WTI_TDI_DCO_CLOCK_SOURCE

WP_CLOCK_REC_DCO_CLOCK_BRGI1

Burn suitable synthesis on the UFE and run the compiled test.

Available tests in this case are all 10x_SBI_xxx tests

Example of SBI board configuration for WP3 and WP2 shown in the following figures.

3.7.7.1 WDS3 - OC3/12 –SBI Adaptive or Regular Differential CR Setup.

The UFE can be located on the first or the second UPI (according to WTI_UFE_UPI_PORT). In the following figure the second UPI port is used.

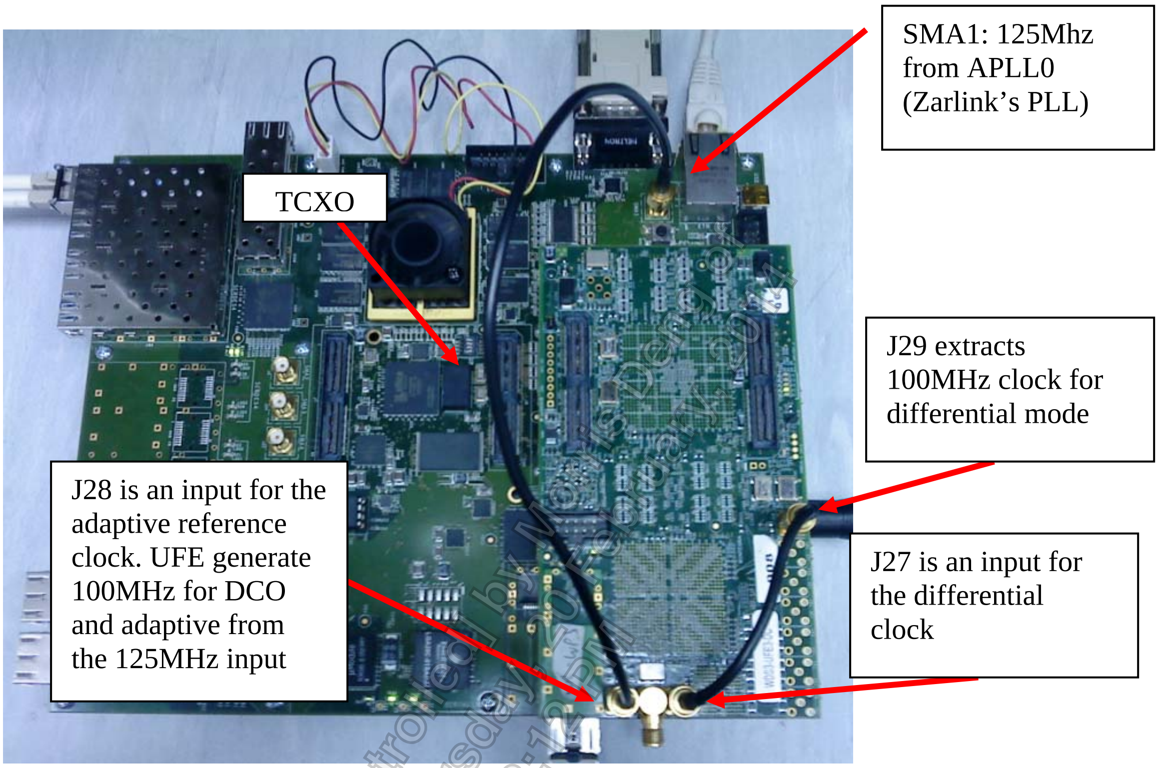


Figure 71: General view of WDS3 with UFE-SBI over second UPI port

3.7.7.2 WDS2 - OC3/12 –SBI Adaptive or Regular Differential CR Setup.

The UFE can be located on the first or the second UPI (according to WTI_UFE_UPI_PORT). In the following figure the second UPI port is used.

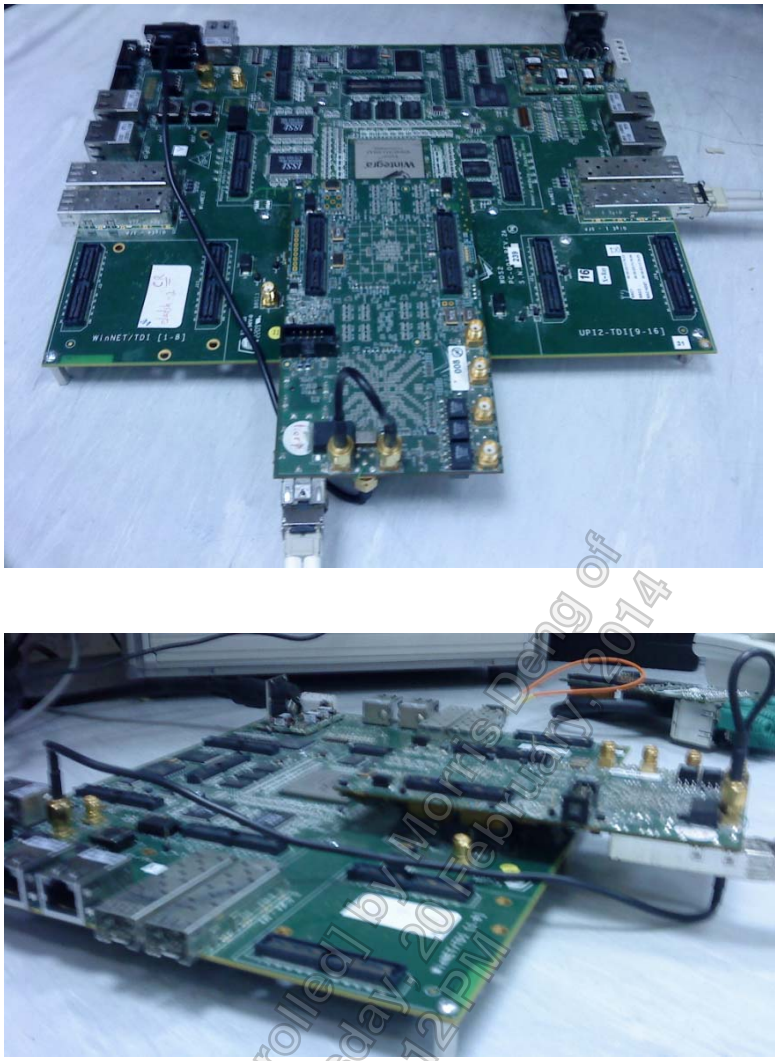


Figure 72: General view of WDS3 with UFE-SBI over first UPI port

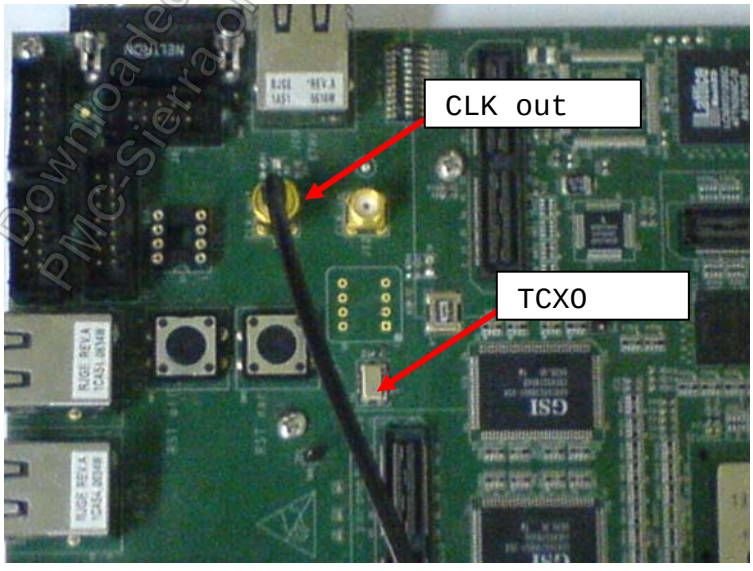


Figure 73: WDS2's J9 extracts 100MHz clock

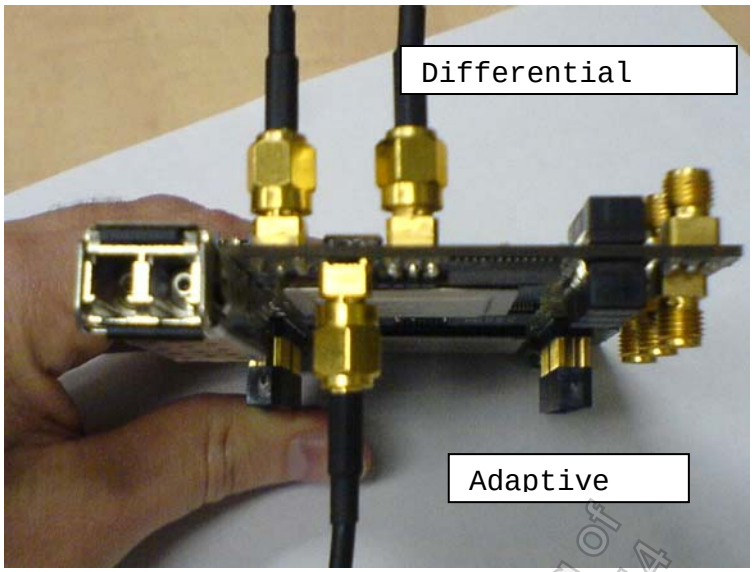


Figure 74: UFE3 Connection for WP2

3.7.8. UFE3 OC3/12 –SBI Extended Differential CR Mode

To run an extended differential mode, set defines as shown in previous section but replace the follow:

#define WTI_CLOCK_REC_MODE 0

#define WTI_EXT_DIFF_MODE_ACTIVE 1

#define WTI_ACTIVE_CR_FOR_ASYNC_MODE 0

#define WTI_ACTIVE_CR_FOR_EXT_DIFF_MODE 336

Burn suitable synthesis on the UFE and run the compiled test.

Available tests in this case are all 13x_SBI_xxx tests.

3.7.9. UFE3 OC3/12 –SBI Adaptive Differential and Extended Differential CR Mode

To run combined mode of adaptive and extended differential, change the following:

#define WTI_ACTIVE_CR_FOR_ASYNC_MODE 32

#define WTI_ACTIVE_CR_FOR_EXT_DIFF_MODE 336

Change 32/336 to fit the number of available adaptive units and extended differential units existing in the synthesis.

Burn suitable synthesis on the UFE and run the compiled test.

Available tests in this case are all 14x_SBI_xxx tests.

3.7.10. UFE3 E1/T1 –CAD Adaptive/Differential or Extended Differential CR Modes

In CAD mode the UFE3 is processing E1/T1 PDH lines and therefore connected to E1/T1 framer (PMC COMET).

To run an UFE3 CAD mode, set defines as shown in 6.1 or 6.2 and add the follow:

#define WTI_CAD_0_3_SETUP 1

#define N_ACTIVE_UFE_OCT_LINES 4

Burn suitable synthesis on the UFE -CAD and run the compiled test.

Available tests in this case are all 2xx_CAD tests.

3.7.10.1 UFE3 E1/T1 –CAD CR Setup

In WDS3 the UFE must located on the first UPI (UPI1) and COMET used by the UFE3 must be located on UPI3 (below the WDS3).in this mode the Winpath fan should be relocated.

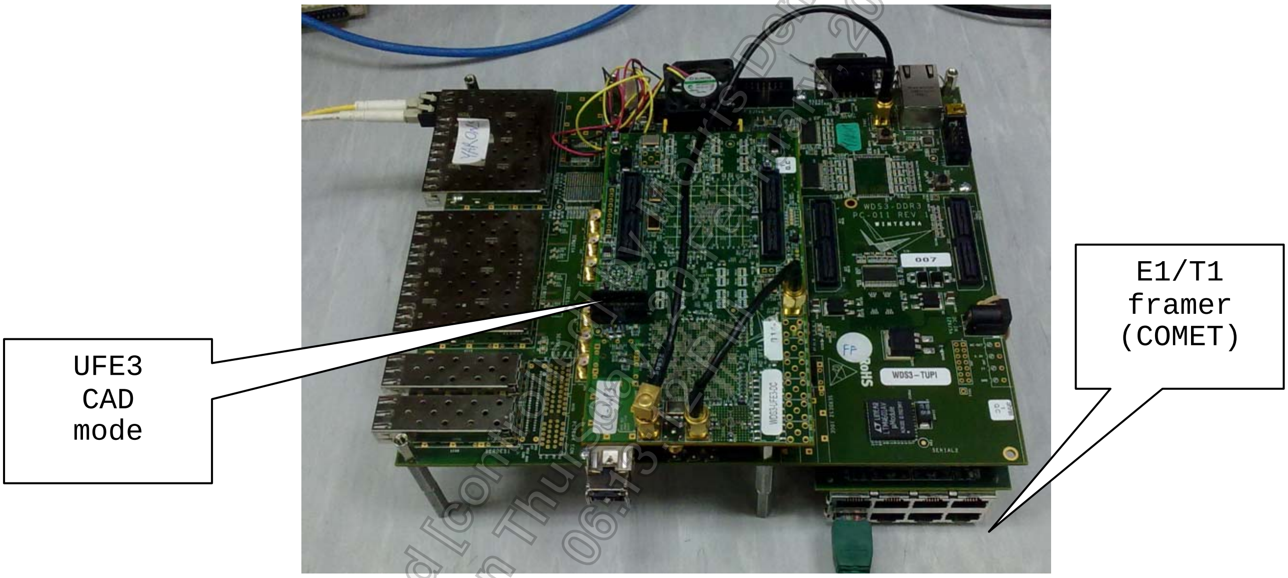


Figure 75: WDS3 and UFE3 E1/T1 CAD CR setup

3.7.11. TDI Adaptive/Differential CR Modes

In order to run a TDI adaptive or regular differential CR test, set the following defines:

#define WTI_CESOP_TDI 1

#define WTI_CESOP_CLOCK_RECOVERY_ENABLE 1

#define WTI_CESOP_RX_TIMING_ENABLE 1

#define WTI_CESOP_REGRESSION_TEST 0

#define WTI_CLOCK_REC_MODE 1 (for differential or '0' for adaptive)

#define WTI_EXT_DIFF_MODE_ACTIVE 0

#define WTI_ACTIVE_CR_FOR_ASYNC_MODE 32

If working in WP3, select the DCO clock and TS clock sources by setting the following defines:

#define WTI_TDI_DCO_CLOCK_SOURCE

WP_CLOCK_REC_DCO_CLOCK_BRGI1

#define WTI_TDI_TS_CLOCK_SOURCE WP_CLK_SRC_BRGI1

#define WTI_WG_TS2_CLOCK_SOURCE WP_CLK_SRC_BRGI1

Run the compiled test.

Available tests in this case are all 30x_TDI_xxx tests.

Example of WDS3 board for TDI configuration shown in the following figure:

Figure 76: WDS3 16 TDI E1/T1 CR setup

3.7.12. Two Boards Setup

When WDS-to-WDS test is required, verify the following defines:

#define WTI_ENET_MODE WP_ENET_NORMAL

#define TWO_BOARDS_SIMULATION 1

#define WTI_BOARD_1 or undef WTI_BOARD_1, depends on the board's number (board #1 VS. board #2)

3.7.13. Alarms & PM modes

In order to run alarms and performance monitoring test, set the following defines inside wti_flexmux_util.h:

#define WTI_FLEXMUX_ENABLE_ALARMS 1

#define WTI_FLEXMUX_ENABLE_PERFORMANCE_MONITORING 1

The Demo application alarms & performance monitoring related functions located in the following files:

wti_flexmux_util.h

wti_flexmux_util.c

wti_flexmux_alarms_and_pm.c

The framer can service the framer mailbox in two ways:

Poll for work or wait for interrupts in order to work.

In order to Poll for work set inside wti_flexmux_util.h:

#define USE_POLLING 1

Otherwise set:

#define USE_POLLING 0

Interrupts are used to control the threads which service the framer (that is, the mailbox thread and the thread to service alarms and/or PM reports) for performance reasons. Otherwise these subsidiary threads must poll for work, which consumes unproductive CPU cycles.

The model implemented uses two software locks:

(1) the interrupt handler should release the mailbox thread, if a framer interrupt is received

(2) the mailbox thread should release the service thread, if it finds alarms and/or PM messages requiring additional processing

(3) the mailbox thread waits on its lock until released by the interrupt handler then processes the mailbox

(4) the service thread waits on its lock until released by the mailbox thread

Any implementation using interrupts/locks should use software locks. Hardware locks are an extremely scarce resource.

NOTE: WPID == 0 in any code snippets.

In order to use interrupts with the WDS3 + UFE412 on which UFE412 interrupts are routed to EINT3, you must

1) Create an interrupt handler for the UFE412 interrupts
    a. The interrupt handler prototype is:

        Void handler (WP_U32 wpid, WP_U32 signal_info)

    b. The handler must call WPX_UFE_FRAMER_ISR for your device to dismiss the framer interrupts

    c. The handler must call WUFE_UfeReadEvents to dismiss UFE interrupts

2) register the handler to service EINT3 interrupts, by calling WPL_IntConnect and WPL_IntEnable.

   If your interrupt handler is named Eint3InterruptHandler,

   WPL_IntConnect(WP_WINPATH(0), WPL_Eint3Ufe4, 0, Eint3InterruptHandler);

   WPL_IntEnable(WP_WINPATH(0), WPL_Eint3Ufe4);

3) If you are running on WinMon (but not Linux), unmask EINT3 in the interrupt controller by calling

   WPL_ InterruptConfigureEint3(WP_WINPATH(0));

4) Enable the framer and UFE interrupts, by calling

   WUFE_UfeFramerHwInterruptEnable(ufeid);

   WUFE_ SystemInterruptEnable(ufeid);

   Other interrupt sources are possible, but it is recommended that UFE core interrupts NOT be enabled.

5) Select EINT3 for the UFE412 interrupts in the CPLD by calling

   WPX_Ufe412CpldInterruptSelect(0, WPX_INTERRUPT_EINT3);

6) Unmask framer interrupts in the CPLD by calling

   WPX_Ufe412CpldInterruptMaskSet

   giving it the appropriate value based on the serial on which the UFE412 resides:

   for UPI1, use ~WPX_FPGA_INTR_SERIAL_1_CONNECTOR

   for UPI3, use ~WPX_FPGA_INTR_SERIAL_3_CONNECTOR

   for the dual, ~(WPX_FPGA_INTR_SERIAL_1_CONNECTOR |
   WPX_FPGA_INTR_SERIAL_3_CONNECTOR)

7) Register the application's function to "wait on its lock" with the mailbox thread by calling

   WPX_FRMR_RegisterFramerLockCallback(void *fnc);

   Where the arguments is of type:

   WPX_UFE_FRAMER_LOCK_CALLBACK_TYPE

   Which is defined as:

   Void (*WPX_UFE_FRAMER_LOCK_CALLBACK_TYPE)(void)

8) Initialize the software locks with WPL_LockKeyInit

9) If running on Linux, the CPLD mask at the time of the interrupt is available to the interrupt handler in the signal_info argument. It is contained in the lower 16 bits. Linux will mask all interrupts at the CPLD before the interrupt handler is called. You must unmask the framer interrupts at the CPLD before exiting interrupt handler as described in #6 above.

In the sample implementation in the CES demo, taking a lock (that is, blocking until released):

```
void WTI_flexmux_take_lock(void)
{
  int rc;
  while (1)
  {
    rc = wpv_swsem_acquire((int *)(&lock), NO_WAIT);
    if (rc == WPL_OK)
      /* Got the lock */
      return;
    /* Force a noncached reference */
    WPX_Ufe412CpldInterruptMaskGet(0);
  }
}
```

and releasing the lock:

```
void WTI_flexmux_release_lock(void)
{
  wpv_swsem_release((int *)(&lock));
}
```

In the demo, we have 2 function pairs, one for the "framer lock" and one for the "server lock".

3.8. Appendix A – Example Setting

This section contains all main defines that includes in the pwe3_cesop_demo. If not mention otherwise, use this setting for running the application:

```
#define WTI_CESOP_TDI                    0          /* use of TDM I/F */
#define WTI_CESOP_CLOCK_RECOVERY_ENABLE  1          /* enable clock recovery module */
#define WTI_CESOP_RX_TIMING_ENABLE       1          /* enable Rx timing direction */
#define WTI_CESOP_REGRESSION_TEST        0          /* when '1' running in regression mode*/

#define WTI_8K_CH_SETUP                  0          /* Should be set only for 8k ch test (UFE3)*/
#define WTI_CAD_0_3_SETUP                0          /* Should be set only for CAD 0-3 test with Winpath2 and
UFE3
                                                    Note that the physical configuration requires UFE3 on UPI2 and Comet on
TDI 1-8 */

#ifdef WP_HW_WINPATH3
#define WTI_XGI_MODE                     0          /* Select if XGI mode active (10-Giga ENET) - for WP3 only
*/
#else
#define WTI_XGI_MODE                     0          /* Always '0' - Not valid for WP2 */
#endif /* WP_HW_WINPATH3 */

#define WTI_WINHDP2_CONFIGURATION        0          /* This compilation macro indicates the use of
WinHDP2 (enet ports are 3 and 4 instead of 1 and 2) */
#define WTI_CESOP_MPLS_IW                0          /* POS MPLS on UPI2 */
#define WTI_CESOP_MEF8_ENCAP             0          /* MEF8 encapsulation */
#if WTI_CESOP_MPLS_IW
#define WTI_CESOP_MPLS_OVER_ENET         0
#define WTI_CESOP_TO_MPLS_FA_CASCADING   0          /* MPLS FA cascading enable - in this mode
the TDM to PSN flow aggregation points at a layer2 router
                                                    MPLS flow aggregation which adds the PSN tunnel MPLS label */
#define WTI_CESOP_NATIVE_MPLS_IW         0          /* NATIVE POS MPLS on UPI2 (PPP header is not
transmitted) */
#define WTI_MPLS_LOOPBACK                0          /* Use loop back on UPI MPLS port (no need for OC card
*/
#endif /* WTI_CESOP_MPLS_IW */
#define WTI_DEBUG_LEVEL                  0          /* DEBUG LEVEL (0-lowest, 2-highest) */
#if WTI_8K_CH_SETUP
#define WTI_MAX_PW                       8064       /* maximum number of PW's in PWE3 system */
#else
#define WTI_MAX_PW                       336        /* maximum number of PW's in PWE3 system */
#endif

#if (!WTI_CESOP_TDI)
#define WTI_MAX_NUM_EMPHY_PHYS           WTI_MAX_PW /* maximum number of emphy phys,
relevant only if working with UFE */
#endif /* (!WTI_CESOP_TDI) */

#if WTI_CESOP_TO_MPLS_FA_CASCADING
#define WTI_MAX_NUM_OF_MPLS_FA           4          /* maximum number of layer2 router MPLS flow
aggregations */
#define WTI_TDM2PSN_L2_MPLS_ROUTER_TAG   0xf0f00000 /* unique tag for each layer2 router
MPLS flow aggregation (| FA INDEX) */
#define WTI_PSN2TDM_L2_MPLS_ROUTER_TAG   0xe0e00000 /* unique tag for each layer2 router
MPLS flow aggregation (| FA INDEX) */
#endif /* WTI_CESOP_TO_MPLS_FA_CASCADING */
#define WTI_INFO_STRUCTURES_DEFINED 1
#if WTI_8K_CH_SETUP
#define WTI_2_ENET_DEVICES               1          /* two enet devices are created, pwe3 packets are received
at one port and transmitted from the second port */
#else
```

```
#define WTI_2_ENET_DEVICES                  0          /* two enet devices are created, pwe3 packets are received
at one port and transmitted from the second port */
#endif
#if WTI_8K_CH_SETUP
#define WTI_FIRST_ENET_MAX_PW               (WTI_MAX_PW / 2) /* defines how many PW's point to the
first ENET port (in 8K configuration 4k goes to first Enet and 4k goes to second Enet*/
#else
#define WTI_FIRST_ENET_MAX_PW               (WTI_MAX_PW+1) /* defines how many PW's point to the
first ENET port (all PWs go to the first Enet port) */
#endif

#define WTI_WPID                    0          /* Winpath ID */

#define WTI_EMPHY_PORT                  WP_PORT_UPI2  /* EMPHY port */

#if WTI_CAD_0_3_SETUP
#undef WTI_EMPHY_PORT
#define WTI_EMPHY_PORT                  WP_PORT_UPI2  /* EMPHY port */
#endif
#define WTI_UFE_UPI_PORT                WTI_EMPHY_PORT

#define WTI_ENET_MODE                   WP_ENET_NORMAL /* defines loopback or normal operation on
ENET port */

#ifdef WP_HW_WINPATH2
#define WTI_ENET_PORT                 WP_PORT_ENET1     /* ENET port */
#define  WTI_ENET_TYPE                      WP_ENET_RGMII_1000 /* (WP_ENET_RGMII_10_100)
or(WP_ENET_RGMII_1000) */
#define WTI_GMII_TYPE                 WTI_GMII_OV_FIBER /* GMII over fiber or over copper*/
#endif  /* WP_HW_WINPATH2 */
#ifdef WP_HW_WINPATH3
#if WTI_XGI_MODE
#define  WTI_ENET_PORT                      WP_PORT_XGI1        /* XGI port: WP_PORT_XGI1 or
WP_PORT_XGI2 */
#define WTI_ENET_TYPE                 WP_ENET_XAUI     /* defines XAUI modes for XGI */
#else
#define WTI_ENET_PORT                 WP_PORT_ENET13     /* ENET port */
#define  WTI_ENET_TYPE                      WP_ENET_SGMII_1000 /* (WP_ENET_RGMII_10_100)
or(WP_ENET_RGMII_1000) */
#define WTI_GMII_TYPE                 WTI_GMII_OV_FIBER
#endif /* WTI_XGI_MODE */
#endif  /* WP_HW_WINPATH3 */

#define WTI_ENET_IEEE_802           0          /* defines working in IEEE 802.1Q ethernet header */

#if (WTI_2_ENET_DEVICES)
#define WTI_SECOND_ENET_MODE                WP_ENET_NORMAL /* defines loopback or normal operation
on ENET port */
#ifdef WP_HW_WINPATH2
/* second ENET port (relevant only in working in simulate network mode or in 2 ENET devices mode) */
#define WTI_SECOND_ENET_PORT                WP_PORT_ENET2
#define WTI_SECOND_ENET_TYPE                    WP_ENET_RGMII_1000 /* (WP_ENET_RGMII_10_100)
or(WP_ENET_RGMII_1000) */
#define WTI_SECOND_GMII_TYPE                WTI_GMII_OV_FIBER  /* GMII over fiber or over copper for
second enet*/
#endif /* WP_HW_WINPATH2 */
#ifdef WP_HW_WINPATH3
/* second ENET port (relevant only in working in simulate network mode or in 2 ENET devices mode) */
#if WTI_XGI_MODE
#define WTI_SECOND_ENET_PORT                WP_PORT_XGI2      /* Second XGI port: WP_PORT_XGI1 or
WP_PORT_XGI2 */
#define WTI_SECOND_ENET_TYPE                WP_ENET_XAUI      /* defines XAUI modes for XGI */
#else /* WTI_XGI_MODE */
#define WTI_SECOND_ENET_PORT                WP_PORT_ENET14
```

```
#define WTI_SECOND_ENET_TYPE                    WP_ENET_SGMII_1000 /* (WP_ENET_RGMII_10_100)
or(WP_ENET_RGMII_1000) */
#define WTI_SECOND_GMII_TYPE                WTI_GMII_OV_FIBER
#endif /* WTI_XGI_MODE */
#endif /* WP_HW_WINPATH3 */
#endif /* WTI_2_ENET_DEVICES */


#if WTI_WINHDP2_CONFIGURATION
#undef WTI_GMII_TYPE
#define WTI_GMII_TYPE WTI_GMII_OV_COPPER
#define WTI_FIRST_ENET_PORT             WP_PORT_ENET4       /* ENET port */
#define WTI_FIRST_ENET_MODE                 WP_ENET_NORMAL      /* defines loopback or normal
operation on ENET port */
#define WTI_FIRST_ENET_TYPE                 WP_ENET_RGMII_1000   /* defines 10/100 (WP_ENET_MII) or
GIGA (WP_ENET_GMII)
                                        modes for ENET */
#define WTI_SECOND_ENET_PORT            WP_PORT_ENET3       /* ENET port */
#define WTI_SECOND_ENET_MODE                WP_ENET_NORMAL      /* defines loopback or normal
operation on ENET port */
#define WTI_SECOND_ENET_TYPE                WP_ENET_RGMII_1000   /* defines 10/100 (WP_ENET_MII)
or GIGA (WP_ENET_GMII)                                  modes for ENET */
#endif /* WTI_WINHDP2_CONFIGURATION */

#define N_ENET_TX_CHANNELS              8
#define N_ENET_RX_CHANNELS              8

#define N_ENET_PQ_BLOCK_SIZE            0
#define N_ENET_PQ_BLOCKS            0
#define N_IW_FLOW_AGGREGATIONS              (WTI_MAX_PW * 2)
#define N_IW_ROUTING_SYSTEMS            1

#if WTI_CESOP_TO_MPLS_FA_CASCADING
#define N_ROUTING_MAX_FLOWS             ((WTI_MAX_PW + 1) + 4)
#else /* !WTI_CESOP_TO_MPLS_FA_CASCADING */
#define N_ROUTING_MAX_FLOWS             (WTI_MAX_PW + 1)
#endif /* WTI_CESOP_TO_MPLS_FA_CASCADING */

#define N_MAX_OUT_OF_BAND_CAS_FLOWS     2
#define N_POOLS                 10

#define N_ACTIVE_UFE_SBI_SPE            12
#if WTI_CAD_0_3_SETUP
#define N_ACTIVE_UFE_OCT_LINES              4               /* UFE :number of active OCTAL lines, must be
defined if working in UFE OCTAL mode */
#else
#define N_ACTIVE_UFE_OCT_LINES              8               /* UFE :number of active OCTAL lines, must be
defined if working in UFE OCTAL mode */
#endif /* N_ACTIVE_UFE_OCT_LINES */

#define N_HOST_RX_CHANNELS              1
#define N_HOST_TX_CHANNELS              1
#define N_HOST_TX_GROUPS                N_HOST_TX_CHANNELS

#if WTI_CESOP_MPLS_IW
#define WTI_MPLSoPPP                0xff030281
#define WTI_MPLS_LABEL                  0x100001ff
#if ( WTI_CESOP_MPLS_OVER_ENET || WTI_CESOP_MEF8_ENCAP)
#define WTI_ETHERNET_HEADER_SIZE            14
#else
#define WTI_PPP_HEADER_SIZE             4
#endif
#endif /* WTI_CESOP_MPLS_IW */

#define WTI_NUM_OF_VLAN_TAGS                0
```

```
#if WTI_CESOP_MEF8_ENCAP
#ifdef WTI_ETHERNET_HEADER_SIZE
#undef WTI_ETHERNET_HEADER_SIZE
#endif
#define WTI_ETHERNET_HEADER_SIZE            14
#define WTI_ECID_HEADER_SIZE                4
#define WTI_VLAN_TAG_SIZE                   4
#endif /*WTI_CESOP_MEF8_ENCAP*/
#if WTI_CESOP_TO_MPLS_FA_CASCADING
#if WTI_CESOP_MPLS_OVER_ENET
#undef WTI_ETHERNET_HEADER_SIZE
#define WTI_ETHERNET_HEADER_SIZE            0
#else
#undef WTI_PPP_HEADER_SIZE
#define WTI_PPP_HEADER_SIZE                 0
#endif
#endif/* WTI_CESOP_TO_MPLS_FA_CASCADING */

#define WTI_MPLS_LABEL_SIZE                 4

#if WTI_CESOP_MEF8_ENCAP
#define WTI_ECID_LABEL_SIZE                 4
#endif /*WTI_CESOP_MEF8_ENCAP*/

/* TDI configuration */
#if WTI_CESOP_TDI
#define WTI_ACTIVE_TDI_PORTS                16    /* number of active TDI ports, must be defined if working in
TDI mode */
#define WTI_TDM_MODE                    WP_TDM_NORMAL
#define WTI_TDM_INT_MODE                WP_TDM_INT_ENABLE
#define WTI_TX_PARAM_BUS_TYPE             WP_BUS_HOST
#define WTI_TX_PARAM_BUS_BANK             APP_BANK_HOST
#define WTI_TX_DATA_BUS_TYPE            WP_BUS_PACKET
#define WTI_TX_DATA_BUS_BANK            APP_BANK_PACKET
#define WTI_TX_INTERRUPT_QUEUE            WP_IRQT1
#define WTI_RX_PARAM_BUS_TYPE             WP_BUS_PARAM
#define WTI_RX_PARAM_BUS_BANK             APP_BANK_PARAM
#define WTI_RX_DATA_BUS_TYPE            WP_BUS_PACKET
#define WTI_RX_INTERRUPT_QUEUE            WP_IRQT1


#if WTI_CESOP_CLOCK_RECOVERY_ENABLE

/* WTI_TDI_DCO_CLOCK_SOURCE - select the soruce clock for TDI DCO block */
#ifdef WP_HW_WINPATH3
/*
  WP3 WTI_TDI_DCO_CLOCK_SOURCE:
  WP_CLOCK_REC_DCO_CLOCK_BRGI1
  WP_CLOCK_REC_DCO_CLOCK_BRGI2
  WP_CLOCK_REC_DCO_CLOCK_BRGI3
  WP_CLOCK_REC_DCO_CLOCK_BRGI4
  WP_CLOCK_REC_DCO_CLOCK_ENET
  WP_CLOCK_REC_DCO_CLOCK_INTERNAL
*/
#define WTI_TDI_DCO_CLOCK_SOURCE            WP_CLOCK_REC_DCO_CLOCK_BRGI1

#else /* WP_HW_WINPATH2 */
/*  WP2 WTI_TDI_DCO_CLOCK_SOURCE:
   WP_CLOCK_REC_DCO_CLOCK_BRGI3
   WP_CLOCK_REC_DCO_CLOCK_INTERNAL
*/
#define WTI_TDI_DCO_CLOCK_SOURCE            WP_CLOCK_REC_DCO_CLOCK_INTERNAL
#endif /* WP_HW_WINPATH2 */
#endif /* WTI_CESOP_CLOCK_RECOVERY_ENABLE */
```

```
/* WP33 :WTI_TDI_TS_CLOCK_SOURCE - select the soruce clock for TDI timestamp block
   WP_CLK_SRC_BRGI1          clock source: BRGin #1
   WP_CLK_SRC_BRGI2,         clock source: BRGin #2
   WP_CLK_SRC_BRGI3,         clock source: BRGin #3
   WP_CLK_SRC_BRGI4,         clock source: BRGin #4
   WP_CLK_SRC_ENET,          clock source: Enet clock
   WP_CLK_SRC_FCLK1,         clock source: Global DCO #1
   WP_CLK_SRC_FCLK2,         clock source: Global DCO #1
   WP_CLK_SRC_INT_SYS        clock source: Internal sys clk */

#define WTI_TDI_TS_CLOCK_SOURCE                WP_CLK_SRC_BRGI1 /* WP3 tdi timestamp source     */
#endif /* WTI_CESOP_TDI */
#define WTI_WG_TS2_CLOCK_SOURCE                WP_CLK_SRC_BRGI1 /* WP3 wingine timestamp source */
#define WTI_NONE_ZERO                          1

#ifdef WP_HW_WINPATH3
/* Classifier config */
#define WTI_PCE_CLASSIFIER                     0          /* PCE classifier is in use instead of the DFC */
#endif
#if !WTI_PCE_CLASSIFIER
/* Classifier config */
#define NUM_OF_ROUTE_FLOWS                     WTI_MAX_PW
#define MAX_NUM_OF_CLASS_FILTERS               32
#define MAX_FLOWS_PER_FILTER                   100
#endif

/* Qnodes configuration */
#define N_HOST_QUEUE_BUFFERS                   1000
#define N_HOST_QUEUE_BUFFER_SIZE               800 /* regular tests need only 256. 800 is for 30/24
                                               slots tests in CAS mode */
#define N_HOST_QUEUE_RINGS                     100
#define N_HOST_QUEUE_RING_LENGTH               4

#define N_IW_PSN2TDM_QUEUE_ADJUNCT_POOL_BUFFERS    20
#define N_IW_PSN2TDM_QUEUE_BUFFERS             5000
#define N_IW_PSN2TDM_QUEUE_BUFFER_SIZE         800 /* regular tests need only 512. 800 is for 30/24
                                               slots tests in CAS mode */

#if WTI_8K_CH_SETUP
#define N_IW_TDM2PSN_QUEUE_ADJUNCT_POOL_BUFFERS    10
#define N_IW_TDM2PSN_QUEUE_BUFFERS             10000
#define N_IW_TDM2PSN_QUEUE_BUFFER_SIZE         800 /* regular tests need only 512. 800 is for 30/24
                                               slots tests in CAS mode */

#define N_TRANS_TX_QUEUE_RINGS                 (2*WTI_MAX_PW)
#define N_TRANS_TX_QUEUE_RING_LENGTH           4
#define  N_TRANS_TX_QUEUE_BUFFERS                                  (N_TRANS_TX_QUEUE_RINGS  *
N_TRANS_TX_QUEUE_RING_LENGTH)

#else
#define N_IW_TDM2PSN_QUEUE_ADJUNCT_POOL_BUFFERS    10
#define N_IW_TDM2PSN_QUEUE_BUFFERS             5000
#define N_IW_TDM2PSN_QUEUE_BUFFER_SIZE         800 /* regular tests need only 512. 800 is for 30/24
                                               slots tests in CAS mode */
#if (WTI_CESOP_CLOCK_RECOVERY_ENABLE || WTI_CESOP_RX_TIMING_ENABLE)
#define N_TRANS_TX_QUEUE_RINGS                 (WTI_MAX_PW)
#define N_TRANS_TX_QUEUE_RING_LENGTH           64
#define  N_TRANS_TX_QUEUE_BUFFERS                                  (N_TRANS_TX_QUEUE_RINGS  *
N_TRANS_TX_QUEUE_RING_LENGTH)
#else /* CR modes */
#define N_TRANS_TX_QUEUE_RINGS                 (WTI_MAX_PW)
#define N_TRANS_TX_QUEUE_RING_LENGTH           256
#define  N_TRANS_TX_QUEUE_BUFFERS                                  (N_TRANS_TX_QUEUE_RINGS  *
N_TRANS_TX_QUEUE_RING_LENGTH)
```

```
#endif /* CR modes */

#endif /* WTI_8K_CH_SETUP */

/* TDM --> PSN */
#define WTI_TDM2PSN_RTP_MODE                    WP_ENABLE
#define WTI_TDM2PSN_INTERRUPT_MODE              WP_DISABLE
#define WTI_TDM2PSN_INTERRUPT_QUEUE             WP_IRQT1
#define WTI_TDM2PSN_OOBC_INTERRUPT_MODE         WP_DISABLE
#define WTI_TDM2PSN_OOBC_INTERRUPT_QUEUE        WP_IRQT1
#define WTI_TDM2PSN_STAT_MODE                   WP_ENABLE
#define WTI_TDM2PSN_OV_POOL_MODE                WP_DISABLE
#define WTI_TDM2PSN_L2_LENGTH_UPDATE_MODE       WP_DISABLE
#define WTI_TDM2PSN_L4_CHECKSUM_RECALC_MODE     WP_DISABLE
#if WTI_CESOP_MPLS_IW
#define WTI_L2_HEADER_LEN                       4
#else
#define WTI_L2_HEADER_LEN                       sizeof(WTI_enet_header)
#endif
#if WTI_CESOP_MEF8_ENCAP
#define WTI_L3_HEADER_LEN                       0
#else
#define WTI_L3_HEADER_LEN                       sizeof(WTI_ip_header)
#endif
#define WTI_SRC_IP_OFFSET                       12
#define WTI_DEST_IP_OFFSET                      16
#define WTI_L4_HEADER_LEN                       sizeof(WTI_udp_header)
#define WTI_L4_LENGTH_OFFSET                    4
#define WTI_RTP_HEADER_LEN                      sizeof(WTI_rtp_header)
#define WTI_CONTROL_WORD_LEN                    4
#define WTI_TDM2PSN_TIMESTAMP_MODE              WP_IW_TIME_STAMP_DISABLE
#define WTI_TDM2PSN_FBP_DROP_THRESHOLD          0xf
#define WTI_TDM2PSN_MTU                         1536
#define WTI_TDM2PSN_L4_LENGTH_UPDATE_MODE       WP_DISABLE
#define WTI_TDM2PSN_INITIAL_SEQ_NUMBER          1
#define WTI_TDM2PSN_OOBC_CAS_INITIAL_SEQ_NUMBER 1000
#define WTI_CAS_LOCATION                        WP_CAS_NIBBLE_LOW

/* PSN --> TDM */
#define WTI_PSN2TDM_INTERRUPT_MODE              WP_DISABLE
#define WTI_PSN2TDM_INTERRUPT_QUEUE             WP_IRQT1
#define WTI_PSN2TDM_STAT_MODE                   WP_ENABLE
#define WTI_PSN2TDM_PAYLOAD_TYPE_CHECK          WP_DISABLE
#define WTI_PSN2TDM_PAYLOAD_TYPE                0x00
#define WTI_PSN2TDM_OOBC_PAYLOAD_TYPE           0x00
#define WTI_PSN2TDM_SSRC_TYPE_CHECK             WP_DISABLE
#define WTI_PSN2TDM_SSRC_TYPE                   0x00000000
#define WTI_PSN2TDM_OOBC_SSRC_TYPE              0x00000000
#define WTI_PSN2TDM_CONTROL_WORD_MODE           WP_ENABLE
#define WTI_PSN2TDM_TIMESTAMP_MODE              WP_IW_TIME_STAMP_DISABLE
#define WTI_PSN2TDM_OV_POOL_MODE                WP_DISABLE
#define WTI_PSN2TDM_FBP_DROP_THRESHOLD          0
#define WTI_PSN2TDM_MTU                         1536

#define WTI_E1_SLOTS                    32
#define WTI_T1_SLOTS                    24
#define WTI_DS3_SLOTS                   1
#define WTI_E3_SLOTS                    WTI_DS3_SLOTS
#define WTI_NUMBER_OF_SPE_DS3_E3        12 /* Number of SPE used for transport of DS3 E3 tests*/

#define WTI_T1_LINES_PER_SPE            28 /* TMX84_T1_LINK_MAX 28 */
#define WTI_E1_LINES_PER_SPE            21 /* TMX84_E1_LINK_MAX 21 */
#define WTI_TDM_SLOTS_PER_SPE           672
#define WTI_MIN_DATAUNIT_SIZE           2
#define WTI_MAX_DATAUNIT_SIZE           64
```

```
#define WTI_MAX_STRING_SIZE                 128
#define WTI_MIN_SLOTS_PER_PHY               1
#define WTI_UNUSED_SLOT                     0xffffffff
#define WTI_MAX_SLOTS                       8064 /* was 2016 in wp1 */
#define WTI_MAX_SPE_PER_UFE                 WUFE_MAX_N_SPE

/* boards configuration */
#define TWO_BOARDS_SIMULATION               0
/*#define WTI_BOARD_1*/

#define CLOCK_REC_TABLE_SIZE 64                     /* CR table size in the dps */

#define CR_CHIP_SELEC_VAL                   0xff807f24 /* chip select value */
#define CR_CHIP_SELECT_ADD                  0x1d03080c /* chip select address for WP2 */

#if WTI_CESOP_TDI
/* CR registers in TDI */
#define CR_HW_MODE_REG_ADD                  0x1f134000 /* clock recovery mode reg' address*/
#else
/* CR registers in UFE */
#define CR_HW_MODE_REG_ADD                  0x1da42100 /* clock recovery mode reg' address*/
#endif /* WTI_CESOP_TDI */

#if (WTI_CESOP_CLOCK_RECOVERY_ENABLE || WTI_CESOP_RX_TIMING_ENABLE)
#define WTI_CLOCK_REC_MODE                  1       /* 1 - differential, 0 - adaptive */
#define WTI_EXT_DIFF_MODE_ACTIVE            0       /* 1 - ext diff, 0 - regular diff */

#if(defined__WT_UFE4__)&&(!WTI_CLOCK_REC_MODE) && (WTI_CESOP_CLOCK_RECOVERY_ENABLE)
#define WTI_CES_SHAPING_ENABLED             1
#define WTI_CES_SHAPING_MAX_CHANNELS   336   /* keep at most WTI_MAX_PW and multiplication of 8  */
#define WTI_CES_SHAPING_FAST_8              0
#endif  /*  WTI_CESOP_TO_MPLS_FA_CASCADING */
#endif   /*  __WT_UFE4__  WTI_CESOP_CLOCK_RECOVERY_ENABLE WTI_CLOCK_REC_MODE*/

#if (WTI_CESOP_CLOCK_RECOVERY_ENABLE && !WTI_CESOP_REGRESSION_TEST)
#define WTI_ADM_MODE                        1   /* on CR usually use ADM */
#else
#define WTI_ADM_MODE                        0
#endif

#if !(WTI_CESOP_TDI)
#define WTI_ACTIVE_CR_FOR_ASYNC_MODE        32      /* number of active clock recovery for
                                                       A-Synchronous mode */
#define WTI_ACTIVE_CR_FOR_EXT_DIFF_MODE     336
#else /* In TDI mode each line is different CRS */
#define WTI_ACTIVE_CR_FOR_ASYNC_MODE        WTI_ACTIVE_TDI_PORTS
#define WTI_ACTIVE_CR_FOR_EXT_DIFF_MODE     0
#ifdef WP_HW_WINPATH2
#if WTI_CLOCK_REC_MODE
#define WTI_CLOCK_REC_TDI_LINES_4_7         0   /* In WP2, TDI differential CR mode,
                                                   choose between lines 0-3 (0) or lines 4-7 (1)
                                                   When 0-3 are used the FPGA (UFE) should be placed on UPI1
                                                   and the correct tdi-cr synthesis should be used

                                                   When 4-7 are used the FPGA (UFE) should be placed on UPI2
                                                   and the correct tdi-cr synthesis should be used
                                                   WTI_ACTIVE_TDI_PORTS <= 8 */
#endif /*WTI_CLOCK_REC_MODE*/
#endif /*WP_HW_WINPATH2*/
#endif

/*------------------- CR Adaptive CAD Backward Compatibly Mode --------------------------------*/
/* In CR Adaptive CAD mode the TS can be taken in DPS instead of UFE (from UFE  synthesis 2.4) */
/* Only the DCO is in the UFE FPGA */
```

```
#if
(!(WTI_CESOP_TDI)&&(!(WTI_CLOCK_REC_MODE))&&((WTI_CESOP_CLOCK_RECOVERY_ENABLE)))
#define WTI_CLOCK_REC_ADA_BW_COM_MODE 0
#else
/*In all other mode Backward Compatibly is not supported */
#define WTI_CLOCK_REC_ADA_BW_COM_MODE 0
#endif
/*-----------------------------------------------------------------------------------------*/


/*------------------- CR Differetial TDI PDCR DOCSIS Mode ---------------------------------*/
/* This mode is supported by the TDI differential mode only and required dedicated FPGA synthesis*/
#if(WTI_CESOP_TDI                    &&                    WTI_CESOP_RX_TIMING_ENABLE                    &&
WTI_CESOP_CLOCK_RECOVERY_ENABLE && WTI_CLOCK_REC_MODE)
#define WTI_CESOP_DIFF_DOCSIS_MODE              0       /* DOCSIS differential mode (in tdi only */
#else
#define WTI_CESOP_DIFF_DOCSIS_MODE              0
#endif
/*-----------------------------------------------------------------------------------------*/

#define WTI_PW_PER_LINE                    1        /* defines the PW location of the second CR
                                        master. Used in the two CR master test.
                                        Have to give integer after devided 24 (T1)
                                        or 32 (E1) with it. */
#else /*No clock Rx or TX Clock recovery */
#define WTI_ACTIVE_CR_FOR_ASYNC_MODE            0       /* No clock recovery */
#define WTI_ACTIVE_CR_FOR_EXT_DIFF_MODE         0
#define WTI_EXT_DIFF_MODE_ACTIVE                0       /* 1 - ext diff, 0 - regular diff */
#define WTI_CLOCK_REC_MODE                      0       /* 1 - differential, 0 - adaptive */
#define WTI_PW_PER_LINE                    1
#endif


#define WTI_MAX_NUM_CLOCK_TRANSLATORS      0       /* define the number of Clock Translators
                                        in the synthesis when CR is not defined.
                                        If CR is defined, the number of CT in
                                        the synthesis is read from the synthesis.
                                        Usualy when CR is not used, the UFE is
                                        TX clock slave and therefore no CT are
                                        used.*/
/*------------------- TDM2PSN RTP Timestamp Absolute mode ---------------------------------*/
                #define WTI_ACTIVE_ABSOLUTE_MODE
```


3.9. Appendix B – Troubleshooting

This section is meant to assist in the debugging process. This section is constantly "in the making". It is currently located on a different file:
PWE3_CES_Demo_Readme_Troubleshooting_v0.1.doc