|  |
| --- |
| Qantum-PCI Time card  Программа тестирования и испытаний |

Содержание

Сводка тестовых элементов и результатов 8

Введение 10

Проверка короткого замыкания источника питания 11

Проверка выходного напряжения источника питания 12

Проверка последовательности включения 13

5.1 Требования к последовательности включения. 13

5.2 Результат тестирования последовательности включения. 13

Измерение уровня напряжения и пульсаций на шинах питания устройств 16

6.1 Измерение напряжения на шинах питания устройств. 16

6.1.1 Измерение напряжения на шинах питания FPGA. 16

6.1.2 Измерение напряжения на шинах питания модуля MAC. 16

6.1.3 Измерение напряжения на шинах питания модуля GPS. 17

6.2 Измерение пульсаций напряжения на шинах питания устройств 17

6.2.1 Измерение пульсаций напряжения на шинах питания FPGA 17

6.2.2 Измерение пульсаций напряжения на шинах питания модуля MAC 19

6.2.3 Измерение пульсаций напряжения на шинах питания модуля GPS 20

Тестирование выходных сигналов генератора тактовых импульсов 22

7.1 Тестирование выходных сигналов генератора тактовых импульсов 125МГц LVDS 22

7.2 Тестирование выходных сигналов генератора тактовых импульсов 200МГц LVDS 23

Предварительное тестирование 25

8.1 Модуль GPS 25

8.1.1 Интерфейс UART модуля GPS 25

8.1.2 Выходные сигналы 1PPS модуля GPS 25

8.2 Модуль MAC 26

8.2.1 Интерфейс UART модуля MAC 26

8.2.2 Выходной сигнал 1PPS модуля MAC 27

8.2.3 Выходной сигнал RF (10МГц) модуля MAC. 28

8.2.4 USB интерфейсы модуля MAC. 29

8.3 Датчики 30

8.3.1 I2C интерфейс для всех датчиков. 30

8.4 Устройства EEPROM 33

8.4.1 I2C интерфейс для устройства EEPROM. 33

8.4.2 I2C интерфейс для второго устройства EEPROM [DVT]. 34

8.5 Драйвер RGB светодиодов и RGB светодиод [DVT] 35

8.5.1 I2C интерфейс для драйвера RGB светодиодов [DVT]. 35

8.5.2 Тестирование RGB светодиодов [DVT]. 35

8.6 FPGA 36

8.6.1 Загрузка образа FPGA в SPI флеш-память через JTAG 36

8.6.2 Тестирование PCIe интерфейса FPGA 37

8.7 Результат измерения Tr, Tf для тактовых сигналов и данных I2C шины. 38

8.8 Просадка напряжения на выводе VCC буферного микрочипа для сигналов FPGA<>SMA I/O. 41

8.9 FT4232 USB to UART/MPSSE [DVT]. 44

8.9.1 Тестирование интерфейса USB2.0 для FT4232 [DVT]. 44

8.9.2 FT4232: Тестирование двигателя MPSSE канала A для JTAG интерфейса [DVT]. 45

8.9.3 FT4232: Тестирование двигателя MPSSE канала A для SPI интерфейса [DVT]. 45

8.9.4 FT4232: Тестирование двигателя MPSSE канала B для I2C интерфейса [DVT]. 46

8.9.5 FT4232: Тестирование UART интерфейса канала C [DVT]. 47

8.9.6 FT4232: Тестирование UART интерфейса канала D [DVT]. 48

8.10 Функция программируемой задержки времени U34 [DVT] 49

8.11 RC-завершение для уменьшения отражения сигнала. [DVT] 50

8.11.1 R-завершение для 10МГц от модуля MAC к FPGA [DVT]. 50

8.11.2 RC-завершение для 1PPS от FPGA к буферному микрочипу [DVT] 52

Функциональное тестирование 57

9.1 Тестирование PCIe интерфейсов FPGA 57

9.2 Тестирование интерфейса I2C-мастер FPGA <> I2C-устройства 58

9.2.1 Тестирование интерфейса I2C-мастер FPGA <> I2C-датчики 58

9.2.2 Тестирование интерфейса I2C-мастер FPGA <> I2C-устройства EEPROM 60

9.3 Тестирование драйвера светодиодов FPGA для RGB\_LED 61

9.4 Интерфейс ядра UART FPGA с модулями GPS и MAC 63

9.4.1 Тестирование интерфейсов UART FPGA<> GPS 63

9.4.2 Тестирование интерфейсов UART FPGA<> MAC 64

9.5 Тестирование PTP с одним клиентом с помощью команды phc2sys 65

9.6 Тестирование PTP с одним клиентом с помощью команды phc2sys [DVT] 66

9.7 Тестирование PTP с одним клиентом с помощью команды ts2phc 68

9.8 Тестирование PTP с одним клиентом с помощью команды ts2phc [DVT] 69

9.9 Тестирование 24-часового удержания 1PPS 71

9.10 Тестирование 24-часового удержания 1PPS [DVT] 72

9.11 Тестирование загрузки нескольких клиентов PTP. 73

9.12 Тестирование задержки одного PCIE. 73

Список таблиц:

Таблица 1: История версий 7

Таблица 2: Справочная документация 7

Таблица 3: Сводка тестовых элементов и результатов 8

Таблица 4: Результаты проверки короткого замыкания источника питания. 11

Таблица 5: Результаты проверки выходного напряжения источника питания. 12

Таблица 6: Результаты проверки последовательности включения. 13

Таблица 7: Измеренное напряжение на шинах питания FPGA. 16

Таблица 8: Измеренное напряжение на шинах питания модуля MAC. 17

Таблица 9: Измеренное напряжение на шинах питания модуля GPS. 17

Таблица 10: Измеренные пульсации напряжения на шинах питания FPGA. 19

Таблица 11: Измеренные пульсации напряжения на шинах питания модуля MAC. 20

Таблица 12: Измеренные пульсации напряжения на шинах питания модуля GPS. 21

Таблица 13: Тестирование выходных сигналов генератора тактовых импульсов 125МГц LVDS. 23

Таблица 14: Тестирование выходных сигналов генератора тактовых импульсов 200МГц LVDS. 24

Список рисунков:

Рисунок 1: Последовательность включения. 13

Рисунок 2: Спецификация напряжения на шинах питания FPGA. 16

Рисунок 3: Характеристики выхода LVDS. 22  
Рисунок 4: Схема подключения I2C для датчиков. 30  
Рисунок 5: Схема подключения I2C для устройства EEPROM. 34  
Рисунок 6: Спецификация шины I2C. 39  
Рисунок 7: Интерфейс I2C-мастер FPGA <> I2C-датчики. 58  
Рисунок 8: Интерфейс I2C-мастер FPGA <> I2C-устройства EEPROM. 60

**История версий платы**

**Таблица 1: История версий**

| **Версия** | **Дата** | **Автор** | **Проверил** |
| --- | --- | --- | --- |
| 0.1 | 21/Oct/2021 |  |  |
| 1.1 | 18/Mar/2023 |  |  |
|  |  |  |  |
|  |  |  |  |

**Ссылки на документы**

| **Item** | Document ID | Version | **Description** |
| --- | --- | --- | --- |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
|  |  |  |  |

1. Сводка тестовых элементов и результатов

**Таблица 3**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Тестовый элемент** | | | | | **Описание теста** | | **Результат (Успешно/**  **Неуспешно)** |
| 3 | Проверка короткого замыкания источника питания | | | | | |  |
| 4 | Проверка выходного напряжения источника питания | | | | | |  |
| 5 | Проверка последовательности включения | | | | | |  |
|  | 5.1 | Требования к последовательности включения. | | | | | - |
|  | 5.2 | Результаты теста последовательности включения. | | | | |  |
| **6** | **Измерение уровня напряжения и пульсаций на шинах питания устройств** | | | | | | - |
|  | 6.1 | Измерение напряжения на шинах питания устройств. | | | | | - |
|  |  | 6.1.1 | | | Измерение напряжения на шинах питания FPGA. | |  |
|  |  | 6.1.2 | | | Измерение напряжения на шинах питания модуля MAC. | |  |
|  |  | 6.1.3 | | | Измерение напряжения на шинах питания модуля GPS. | |  |
|  | 6.2 | Измерение пульсаций напряжения на шинах питания устройств. | | | | |  |
|  |  | 6.2.1 | | | Измерение пульсаций напряжения на шинах питания FPGA. | |  |
|  |  | 6.2.2 | | | Измерение пульсаций напряжения на шинах питания модуля MAC. | |  |
|  |  | 6.2.3 | | | Измерение пульсаций напряжения на шинах питания модуля GPS. | |  |
| 7 | Тестирование выходных сигналов генератора тактовых импульсов | | | | | |  |
|  | 7.1 | Тестирование выходных сигналов генератора тактовых импульсов 125МГц LVDS. | | | | |  |
|  | 7.2 | Тестирование выходных сигналов генератора тактовых импульсов 200МГц LVDS. | | | | |  |
| 8 | **Предварительное тестирование** | | | | | |  |
|  | 8.1 | GPS модуль | | | | |  |
|  |  | 8.1.1 | | | Интерфейс UART модуля GPS. | |  |
|  |  | 8.1.2 | | | Выходные сигналы 1PPS модуля GPS. | |  |
|  | 8.2 | MAC модуль | | | | |  |
|  |  | 8.2.1 | | | Интерфейс UART модуля MAC. | |  |
|  |  | 8.2.2 | | | Выходной сигнал 1PPS модуля MAC. | |  |
|  |  | 8.2.3 | | | Выходной сигнал RF модуля MAC. | |  |
|  |  | 8.2.4 | | | USB интерфейсы модуля MAC. | |  |
|  | 8.3 | Датчики | | | | |  |
|  |  | 8.3.1 | | | I2C интерфейс для всех датчиков. | |  |
|  | 8.4 | EEPROM | | | | |  |
|  |  | 8.4.1 | | | I2C интерфейс для устройств EEPROM. | |  |
|  | 8.5 | Драйвер RGB светодиодов и RGB светодиод | | | | |  |
|  |  | 8.5.1 | | | | I2C интерфейс для драйвера RGB светодиодов [DVT]. |  |
|  |  | 8.5.2 | | | | Тестирование RGB светодиодов [DVT]. |  |
|  | 8.6 | FPGA | | | | |  |
|  |  | 8.5.1 | | | Загрузка образа FPGA в SPI флеш-память через JTAG. | |  |
|  |  | 8.5.2 | | | Тестирование PCIe интерфейса FPGA (Требуется образ FPGA для поддержки теста PCIe). | |  |
|  | 8.7 | Результаты измерения Tr/Tf для тактовых сигналов и данных I2C шины. | | | | |  |
|  | 8.8 | Просадка напряжения на выводе VCC буферного микрочипа для сигналов FPGA<>SMA I/O. | | | | |  |
|  | 8.9 | FT4232 USB to UART/MPSSE [DVT] | | | | |  |
|  |  | 8.9.1 | | Тестирование интерфейса USB2.0 для FT4232 [DVT] | | |  |
|  |  | 8.9.2 | | FT4232: Тестирование двигателя MPSSE канала A для JTAG интерфейса [DVT] | | |  |
|  |  | 8.9.3 | | FT4232: Тестирование двигателя MPSSE канала A для SPI интерфейса [DVT] | | |  |
|  |  | 8.9.4 | | FT4232: Тестирование двигателя MPSSE канала B для I2C интерфейса [DVT] | | |  |
|  |  | 8.9.5 | | FT4232: Тестирование UART интерфейса канала C [DVT] | | |  |
|  |  | 8.9.6 | | FT4232: Тестирование UART интерфейса канала D [DVT] | | |  |
|  | 8.10 | Функция программируемой задержки времени U34 [DVT] | | | | |  |
|  | 8.11 | RC-завершение для уменьшения отражения сигнала. [DVT] | | | | |  |
|  |  | 8.11.1 | R-завершение для 10МГц от модуля MAC к FPGA [DVT] | | | |  |
|  |  | 8.11.2 | RC-завершение для 1PPS от FPGA к буферному микрочипу [DVT] | | | |  |
| 9 | Function test | | | | | |  |
|  | 9.1 | Тестирование PCIe интерфейсов FPGA. | | | | |  |
|  | 9.2 | Тестирование интерфейса I2C-мастер FPGA <> I2C-устройства. | | | | |  |
|  |  | 9.2.1 | | | Тестирование интерфейса I2C-мастер FPGA <> I2C-датчики. | |  |
|  |  | 9.2.2 | | | Тестирование интерфейса I2C-мастер FPGA <> I2C-устройства EEPROM. | |  |
|  | 9.3 | Тестирование драйвера светодиодов FPGA для USR\_LED и RGB\_LED. | | | | |  |
|  | 9.4 | Интерфейс ядра UART FPGA с модулями GPS и MAC. | | | | |  |
|  |  | 9.4.1 | | | Тестирование интерфейсов UART FPGA<> GPS. | |  |
|  |  | 9.4.2 | | | Тестирование интерфейсов UART FPGA<> MAC. | |  |
|  | 9.5 | Тестирование PTP с одним клиентом в течение 24 часов с помощью команды phc2sys. | | | | |  |
|  | 9.6 | Тестирование PTP с одним клиентом с помощью команды phc2sys [DVT]. | | | | |  |
|  | 9.7 | Тестирование PTP с одним клиентом в течение 24 часов с помощью команды ts2phc. | | | | |  |
|  | 9.8 | Тестирование PTP с одним клиентом с помощью команды ts2phc [DVT]. | | | | |  |
|  | 9.9 | Тестирование 24-часового удержания 1PPS. | | | | |  |
|  | 9.10 | Тестирование 24-часового удержания 1PPS [DVT]. | | | | |  |
|  | 9.11 | Тестирование загрузки нескольких клиентов PTP. | | | | |  |
|  | 9.12 | Тестирование задержки одного PCIE. | | | | |  |

1. Введение

Этот документ описывает этапы запуска, аппаратного тестирования и функционального тестирования проекта Qantum-PCI. Документ включает в себя этапы запуска, предварительного тестирования и функционального тестирования.

Предварительное тестирование включает проверку короткого замыкания источника питания, уровня напряжения, пульсаций и последовательности напряжения на шинах питания. Также проводится предварительное тестирование выходных сигналов генератора тактовых импульсов. Предварительные сигнальные тесты для интерфейсов PCIe, SPI, I2C и GPIO для каждого устройства на плате проводятся, чтобы убедиться и подтвердить, что все устройства и функции работают нормально.

1. Power supply short check

Use the DMM to measure the resistance between test point and SG. All the power supplies shouldn’t be short to SG.

**Table 4: Power supply short check test result.**

| **Test Item** | **Test Point +** | **Test Point -** | **Result (ohm）** | **Pass / Fail** |
| --- | --- | --- | --- | --- |
| XP12R0V | TP52 | SG | >1M | Pass |
| XP3R3V | TP53 | SG | 1.86k | Pass |
| XP1R0V\_FPGA | TP54 | SG | 1.18k | Pass |
| XP1R2V\_FPGA | TP55 | SG | 3.36k | Pass |
| XP1R8V\_FPGA | TP56 | SG | 1.28k | Pass |
| XP2R5V\_FPGA | TP57 | SG | 3.08k | Pass |
| XP3R3V\_FPGA | TP58 | SG | 5.06k | Pass |
| XP5R0V | TP60 | SG | 25.32k | Pass |

3 Проверка выходного напряжения источника питания

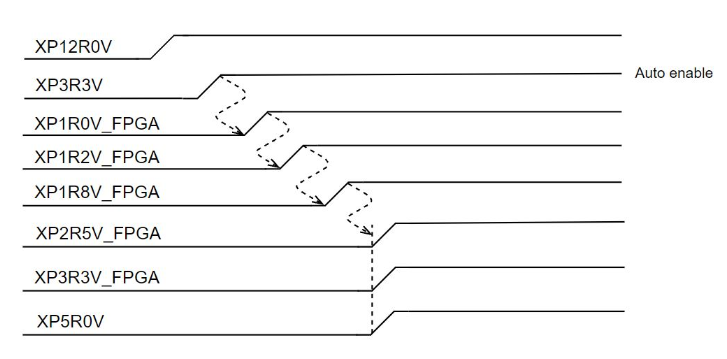
Используйте цифровой измерительный прибор для измерения напряжения на каждой шине питания платы.

Таблица 5: Результаты проверки выходного напряжения источника питания.

| **Тестовый элемент** | **Контрольная точка +** | **Контрольная точка -** | **Результат измерения (V)** | **Критерий** | **Положительно/отрицательно** |
| --- | --- | --- | --- | --- | --- |
| XP12R0V | TP52 | SG | 11.96 | 11.4V ~ 12.6V | Pass |
| XP3R3V | TP53 | SG | 3.325 | 3.135V ~ 3.465V | Pass |
| XP1R0V\_FPGA | TP54 | SG | 1.009 | 0.97V ~ 1.03V | Pass |
| XP1R2V\_FPGA | TP55 | SG | 1.206 | 1.14 ~ 1.26V | Pass |
| XP1R8V\_FPGA | TP56 | SG | 1.796 | 1.71V ~ 1.89V | Pass |
| XP2R5V\_FPGA | TP57 | SG | 2.515 | 2.375V ~ 2.625V | Pass |
| XP3R3V\_FPGA | TP58 | SG | 3.324 | 3.135V ~ 3.465V | Pass |
| XP5R0V | TP60 | SG | 5.051 | 4.75V ~ 5.25V | Pass |

4 Проверка последовательности включения питания

4.1 Требования к последовательности включения питания.



**Рисунок 1: Последовательность включения питания.**

* 1. Power-up sequence test result.

**Таблица 6: Результаты проверки последовательности включения питания.**

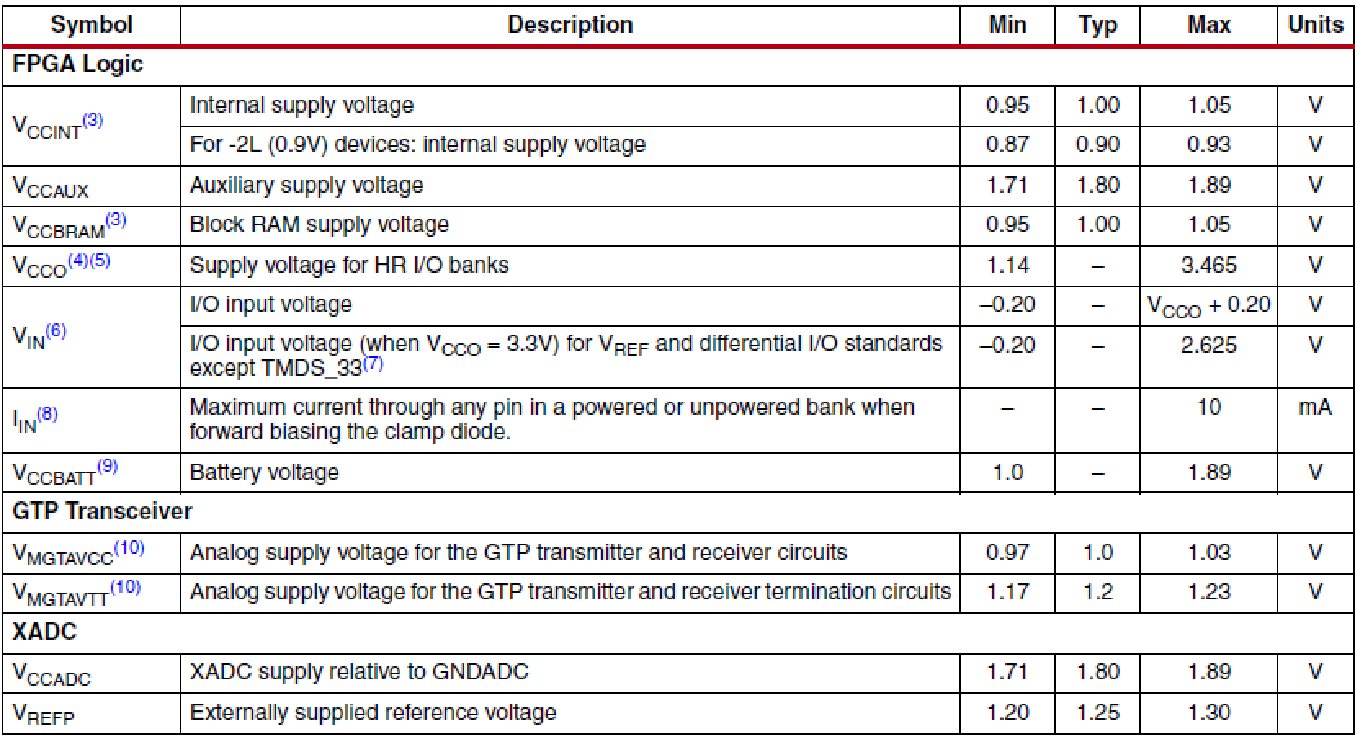
|  |  |  |
| --- | --- | --- |
| **Силовая шина** | **Форма волны** | **Положительно/отрицательно** |
| XP12R0V > XP3R3V  CH1= XP12R0V  CH2= XP3R3V |  | Pass. |
| XP3R3V > XP1R0V \_FPGA  CH1= XP3R3V  CH2= XP1R0V \_FPGA |  | Pass. |
| XP1R0V \_FPGA > XP1R2V\_FPGA  CH1= XP1R0V \_FPGA  CH2= XP1R2V\_FPGA |  | Pass. |
| XP1R2V \_FPGA > XP1R8V\_FPGA  CH1= XP1R2V \_FPGA  CH2= XP1R8V\_FPGA |  | Pass. |
| XP1R8V\_FPGA, > XP2R5V\_FPGA, XP3R3V\_FPGA, XP5R0V  CH1= XP1R8V\_FPGA  CH2= XP2R5V\_FPGA  CH3= XP3R3V\_FPGA  CH4= XP5R0V |  | Pass. |

5 Измерение уровня напряжения на шинах питания и пульсаций напряжения на приборах

Напряжения должны соответствовать требованиям к питанию каждого чипа или модуля.

4.3 Измерение напряжения на шинах питания устройств.

4.3.1 Измерение напряжения на шинах питания ПЛИС.



**Рисунок 2: Спецификация напряжения на силовой шине ПЛИС.**

Таблица 7: Напряжение на шине питания, измеренное на ПЛИС.

| **Voltage** | **Test Point** | **Requirement** | **Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- |
| XP1R0V\_FPGA\_VCCINT | Under ASIC | 0.97V ~ 1.03V | 1.009 | Pass. |
| XP1R0V\_FPGA\_MGTAVCC | Under ASIC | 0.97V ~ 1.03V | 1.009 | Pass. |
| FPGA\_MGTAVTT | Under ASIC | 1.17 ~ 1.23V | 1.206 | Pass. |
| XP1R8V\_FPGA\_VCCAUX | Under ASIC | 1.71V ~ 1.89V | 1.796 | Pass. |
| XP2R5V\_FPGA | Under ASIC | 2.375V ~ 2.625V | 2.514 | Pass. |
| XP3R3V\_FPGA | Under ASIC | 3.135V ~ 3.465V | 3.324 | Pass. |

* + 1. Напряжение на силовой шине, измеренное на модуле MAC.

Таблица 8: Напряжение на шине питания, измеренное на модуле MAC.

| **Voltage** | **Test Point** | **Requirement** | **Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- |
| XP5R0V\_MAC\_USB | P1: pin6 | 4.75V ~ 5.25V | 5.038 | Pass. |
| XP5R0V\_MAC | MAC\_PIN5 | 4.5V ~ 32V | 5.03V | Pass. |

* + 1. Напряжение на силовой шине, измеренное на модуле GPS

Таблица 9: Напряжение на силовой шине, измеренное на модуле GPS.

| **Voltage** | **Test Point** | **Requirement** | **Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- |
| XP5R0V\_VCC\_ANT | P1: pin6 | 4.5V ~ 5.5V | 5.05V | Pass. |
| XP3R3V | P2: pin2 | 2.7V ~ 3.6V | 3.321V | Pass. |

4.3 Измерение пульсаций напряжения на силовой шине на устройствах

|  |  |  |
| --- | --- | --- |
| **Предмет** | **Форма сигнала** | **Положит/отриц** |
| XP1R8V\_FPGA\_VCCAUX  Read from DMM = 1.796V  **Criteria:** V(ripple): < 100mVp-p |  | Pass. |
| XP1R0V\_FPGA\_VCCINT  Read from DMM = 1.009V  **Criteria:** V(ripple) < 100mVp-p |  | Pass. |
| XP3R3V\_FPGA  Read from DMM = 3.324  **Criteria:** V(ripple) < 100mVp-p |  | Pass. |
| XP1R0V\_FPGA\_MGTAVCC  Read from DMM = 1.009  **Criteria:** V(ripple) < 100mVp-p |  | Pass. |
| FPGA\_MGTAVTT  Read from DMM = 1.206V  **Criteria:** V(ripple) < 60mVp-p |  | Pass. |
|  |  |  |

**Table 10: Power rail voltage ripple measured at the FPGA.**

* + 1. Power rail voltage ripple measured at the MAC module

Таблица 11: Пульсации напряжения на силовой шине, измеренные на модуле MAC

|  |  |  |
| --- | --- | --- |
| **Item** | **Waveform** | **Pass/Fail** |
| XP5R0V\_MAC\_USB  Read from DMM = 5.038V  **Criteria:** V(ripple): < 100mVp-p |  | Pass. |
| XP5R0V\_MAC  Read from DMM = 5.03V  **Criteria:** V(ripple) < 100mVp-p |  | Pass. |
|  |  |  |

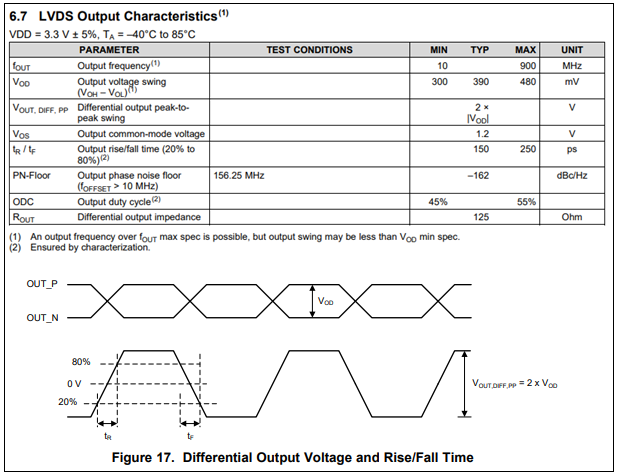
**.**

* + 1. Пульсации напряжения на силовой шине, измеренные на модуле GPS

Таблица 12: Пульсации напряжения на силовой шине, измеренные на модуле GPS.

|  |  |  |
| --- | --- | --- |
| **Item** | **Waveform** | **Pass/Fail** |
| XP5R0V\_VCC\_ANT  Read from DMM = 5.05V  **Criteria:** V(ripple): < 100mVp-p |  | Pass. |
| XP3R3V (P2: pin2)  Read from DMM = 3.321V  **Criteria:** V(ripple) < 100mVp-p |  | Pass. |

5 Проверка выходной мощности тактового генератора



**Рисунок 3: Характеристики выхода LVDS.**

* 1. Проверка выходного сигнала тактового генератора LVDS на частоте 125 МГц

**\* Это измерение выполняется с помощью несимметричного датчика. Для этого измерения мы просто проверяем выход тактовой частоты. Остальные параметры приведены в отчете SI.**

Таблица 13: Проверка выходного сигнала LVDS тактового генератора на частоте 125 МГц.

|  |  |  |
| --- | --- | --- |
| **Signal** | **Waveform** | **Pass/Fail** |
| MHZ125CLKP\_CLKIN  MHZ125CLKN\_CLKIN | 125CLK\_P    //===================================  125CLK\_N    //=================================== |  |

4.2 Проверка выходного сигнала генератора тактовых импульсов 200 МГц LVDS

Для этого измерения мы просто проверяем выходной сигнал генератора тактовых импульсов. Остальные параметры приведены в отчете SI.

Таблица 14: Проверка выходного сигнала LVDS тактового генератора на частоте 200 МГц.

|  |  |  |
| --- | --- | --- |
| **Signal** | **Waveform** | **Pass/Fail** |
| MHZ200CLKP\_CLKIN  MHZ200CLKN\_CLKIN | 200CLK\_P    //===================================  200CLK\_N |  |

* 1. Предварительное испытание

Предварительный тест, нам нужно запустить тест, чтобы убедиться, что все устройства, такие как GPS-модуль, MAC-модуль, ПЛИС, EEPROM и все сенсорные устройства, работают в обычном режиме.

4.5 Модуль GPS

4.5.1 Интерфейс UART модуля GPS

Требование: Протестируйте интерфейсы UART для модуля GPS, чтобы убедиться, что модуль GPS работает в обычном режиме.

Метод тестирования: (Не загружайте встроенное ПО в FPGA для этого тестового примера)

1. Используйте кабель micro USB для подключения между ПК <> J13 и платой time card.

2. Установите на J10 перемычку для подключения контактов 3 и 5, а также 6 и 8 для подключения сигналов FT232 UART к GPRS UART.

3. На главном КОМПЬЮТЕРЕ установите последовательный порт: скорость передачи данных = 115200 бод, бит данных = 8, бит контроля четности = нет, стоп-бит = 1, управление потоком данных = нет.

4. Откройте графический интерфейс U-blox для получения информации из модуля GPS.

Test result.

|  |  |  |
| --- | --- | --- |
| **Тестовый элемент** | **Результат** | **Пассивный/Отрицательный** |
| GNSS UART interface. |  | Pass |

4.4.1 Вывод сигналов GPS-модуля 1PPS

**Требование:** Проверьте сигналы 1PPS для модуля GPS, чтобы убедиться, что модуль GPS работает в обычном режиме.

**Метод проверки:** Используйте осциллограф для регистрации сигналов 1PPS. Выполните измерение на выводе TP1 модуля GNSS.

Результат проверки.

|  |  |  |
| --- | --- | --- |
| **Тестовый элемент** | **Результат** | **Пассивный/**  **отрицательный** |
| 1PPS output signal from TP1 pin of the GNSS module. |  | Pass. |

* 1. MAC модуль
     1. Интерфейс UART модуля MAC

**Требование:** Протестируйте интерфейсы UART для модуля MAC, чтобы убедиться, что модуль MAC работает в обычном режиме.

**Метод тестирования:** (Не загружайте встроенное ПО в FPGA для этого теста)

1. Используйте кабель micro USB для подключения между ПК <> J13 и платой.

2. Разрежьте картон, в точке J10 установите перемычку на контакты 3 и 4 и 7 и 8, чтобы подключить FT232 UART к MAC UART.

3. На главном компьютере установите последовательный порт: скорость передачи данных = 57600 бод, бит данных = 8, бит контроля четности = нет, стоп-бит = 1, управление потоком = нет.

4. Протестируйте с помощью интерфейса командной строки, а затем запишите журнал последовательного вывода из модуля MAC.

**Тестовый результат**

|  |  |  |
| --- | --- | --- |
| **Тестовый элемент** | **Результат** | **Пасс/Отри** |
| Test UART interface for the MAC module |  | Pass. |

* + 1. Выходной сигнал модуля MAC 1PPS

**Требование:** Проверьте сигналы 1PPS для MAC-модуля, чтобы убедиться, что модуль MAC работает в обычном режиме.

Метод проверки: Используйте осциллограф для регистрации сигналов 1PPS.

\* Это измерение выполняется с помощью несимметричного датчика. Это измерение просто подтверждает вывод тактовой частоты. Остальные параметры приведены в отчете SI.

Результат теста.

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| 1PPS output signal from MAC module | Pulse width: FPGA\_IN\_MAC\_PPS\_OUTP    Period: FPGA\_IN\_MAC\_PPS\_OUTP    //===================================  Pulse width: FPGA\_IN\_MAC\_PPS\_OUTN    Period: FPGA\_IN\_MAC\_PPS\_OUTN | Pass. |

* + 1. Радиочастотный выходной сигнал модуля MAC (10 МГц).

**Требование:** Проверьте выходной сигнал MAC-модуля на частоте 10 МГц, чтобы убедиться, что он работает в обычном режиме.

Метод проверки: С помощью осциллографа зафиксируйте выходной сигнал на частоте 10 МГц.

Результат проверки.

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| 10Mhz output signal from MAC module |  | Pass. |

4.5.4 USB-интерфейсы модуля MAC.

**Требование:** Протестируйте USB-интерфейсы для модуля MAC, чтобы убедиться, что модуль MAC работает в обычном режиме.

Test method:

1. Используйте кабель micro USB для подключения между ПК <> J12 и платой time card.

2. На главном компьютере запустите ОС Linux, а затем с помощью команды “lsusb -v” введите список USB-устройств.

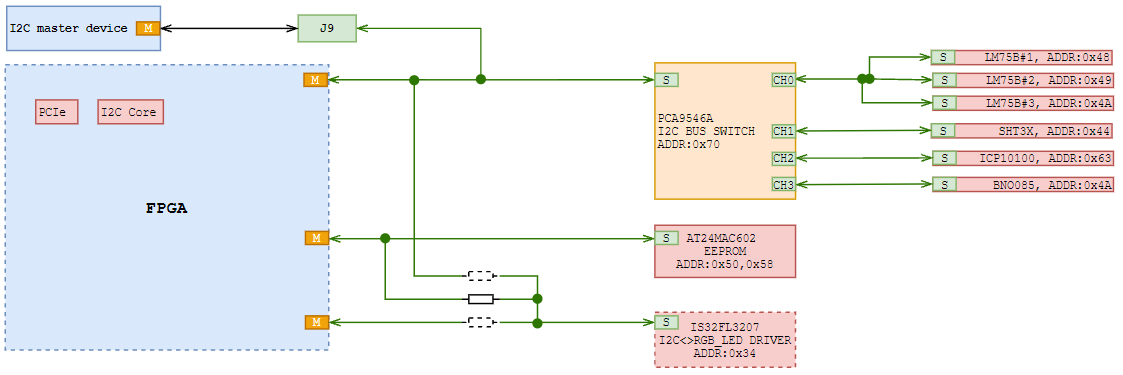
3. Проверьте выводимое сообщение, операционная система должна обнаружить USB-модуль MAC в обычном режиме.

Результат проверки.

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| USB interface test. |  | Pass. |

4.6 Сенсорные устройства.

**Требование:** Протестируйте интерфейсы I2C для всех сенсорных устройств, чтобы убедиться, что все сенсорные устройства работают в обычном режиме.

****

**Рисунок 4: Схема подключения I2C для сенсорных устройств.**

Метод тестирования: (Не загружайте встроенное ПО в FPGA для этого тестового примера)

1. Подключите главное устройство I2C к разъему J9.

2. На главном ПК запустите ОС Linux, а затем с помощью команды составьте список устройств I2C. В нем должны быть найдены все датчики, а именно: 3 датчика температуры, 1 датчик влажности, 1 датчик давления воздуха и 1 датчик вибрации.

3. Затем попробуйте прочитать/записать данные на каждый датчик, чтобы еще раз проверить, что все датчики могут работать с I2C в обычном режиме, а затем зафиксировать выходное сообщение.

Результат тестирования.

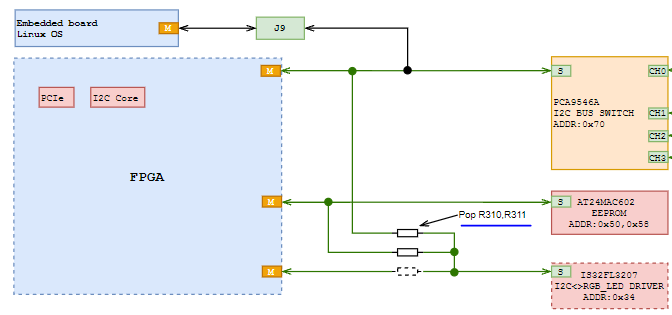
|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| Temperature sensors | I2C device list and command to test the Temperature sensors:  LM75B#1, ADDR: 0x48.  LM75B#2, ADDR: 0x49.  LM75B#3, ADDR: 0x4A.    //===================================  Try to read the register: Reg 0x02 = 0x4B00, Reg 0x03 = 0x5000.    //===================================  LM75B#1, ADDR: 0x48.    //===================================  LM75B#2, ADDR: 0x49.    //===================================  LM75B#3, ADDR: 0x4A. | Pass. |
| Humidity sensor | I2C device list and command to test the Humidity sensor:  SHT3X, ADDR: 0x44. | Pass. |
| Pressure sensor | I2C device list and command to test the pressure sensor:  ICP10100, ADDR: 0x63. | Pass. |
| Vibration sensor | I2C device list and command to test the Vibration sensor:  BNO085, ADDR: 0x4A. | Pass. |

* 1. EEPROM devices
     1. I2C interface for EEPROM device.

**Requirement:** Test I2C interfaces for the EEPROM IC to confirm that the EEPROM IC is working as normally

**Test method:** (Do not load firmware to FPGA for this test case)

1. Connect the I2C bus master to EEPROM I2C device. We can use the connector J9 connect to I2C master and pop R310, R311 to connect EEPROM toI2C bus.



**Figure 5: I2C connection diagram for the EEPROM device.**

1. At the PC host, run the Linux OS and then use the command to list the I2C device it should find EEPROM device.
2. Then, try to read/write the data to each sensor to double check the EEPROM device can R/W by I2C as normally and then capture the output message.
3. After the test is complete, remove the resistor R310, R311 out.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| EEPROM | I2C device list and command to test the EEPROM:  AT24MAC602 EEPROM ADDR: 0x50, 0x58. | Pass. |
|  |  |  |

* + 1. I2C interface for 2nd EEPROM device [DVT].

**Requirement:** Test I2C interfaces for the 2nd EEPROM IC to confirm that the 2nd EEPROM IC is working as normally.

**Test method:**

1. Power-on the UUT and then run the Linux OS. ≈
2. At the HPE server, use the I2C command to scan, read/write the 2nd EEPROM to conform that the I2C interface is working normally.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| I2C interface for 2nd EEPROM device. | 24LC64T EEPROM ADDR: 0x51 | Pass. |
|  |  |  |

* 1. RGB LED driver and RGB LED [DVT]
     1. I2C interface for RGB LED driver [DVT].

**Requirement:** Test I2C interfaces for the RGB LED driver IC to confirm that the I2C <> RGB LED driver IC is working as normally.

**Test method:**

1. Power-on the UUT and then run the Linux OS.
2. Use the command to list the I2C device it should find RGB LED driver IC.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| I2C<> RGB LED driver. |  | Pass. |
|  |  |  |

* + 1. RGB LEDs test [DVT].

**Requirement:** Test each RGB LEDs controlled by RGB LED driver IC to confirm that we can control brightness, on/off for each RGB LED by using the I2C command.

**Test method:**

1. Power-on the UUT and then run the Linux OS.
2. Use the command to list the I2C device it should find EEPROM device.
3. Uses the I2C command to control on/off, brightness each RGB LED.
4. Observe each RGB LED to make sure that all RGB LEDs are good.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| RGB LEDs test. | Observe each RGB LED when sending the command to control the RGB LED.  Set SL (Scaling Register: 0x4b ~0x5b) = 0xff.  //============ LED R =====================  Set PWM reg H\_L for all the RGB LED pin = 0x00.  Set PWM reg H\_L for the LED R pin = 0xff.  Set GCC (Global Current Control Register: 0x6e) = 0xff; and then Observe the LED. Test result = **Pass.**  Set GCC (Global Current Control Register: 0x6e) = 0x5f; and then Observe the LED. Test result = **Pass.**  Set GCC (Global Current Control Register: 0x6e) = 0x0f; and then Observe the LED. Test result = **Pass.**  //============ LED G =====================  Set PWM reg H\_L for all the RGB LED pin = 0x00.  Set PWM reg H\_L for the LED G pin = 0xff.  Set GCC (Global Current Control Register:0x6e) = 0xff; and then Observe the LED. Test result = **Pass.**  Set GCC (Global Current Control Register:0x6e) = 0x5f; and then Observe the LED. Test result = **Pass.**  Set GCC (Global Current Control Register: 0x6e) = 0x0f; and then Observe the LED. Test result = **Pass.**  //============ LED B =====================  Set PWM reg H\_L for all the RGB LED pin = 0x00.  Set PWM reg H\_L for the LED R pin = 0xff.  Set GCC (Global Current Control Register: 0x6e) = 0xff; and then Observe the LED. Test result = **Pass.**  Set GCC (Global Current Control Register: 0x6e) = 0x5f; and then Observe the LED. Test result = **Pass.**  Set GCC (Global Current Control Register: 0x6e) = 0x0f; and then Observe the LED. Test result = **Pass.** | Pass. |
|  |  |  |

* 1. FPGA
     1. Download the FPGA image to SPI flash by JTAG

**Requirement:** SPI flash of the FPGA must be programmed the firmware by using FPGA JTAG programmer as normally.

**Test method:**

1. Connect the FPGA programmer cable to the PC host and the connector J14 on the board.
2. Open “Vivado” UI for Xilinx FPGA to download the image file and then scan the FPGA device.
3. Right-click at the FPGA device that shows in the panel to add the configuration memory derive then select SPI flash part = MT25QI128 and then choose the configuration file for the FPGA and then program.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| Download the firmware to U24 FPGA by JTAG. |  | Pass. |
|  |  |  |

* + 1. PCIe interface test for the FPGA

**Requirement:** FPGA PCIe interface test, the OS can detect the FPGA device and the PCIe link status should link-up with link width = (x1 or x 2 or x4) link speed = 2.5GT/s as normally.

**Test method:**

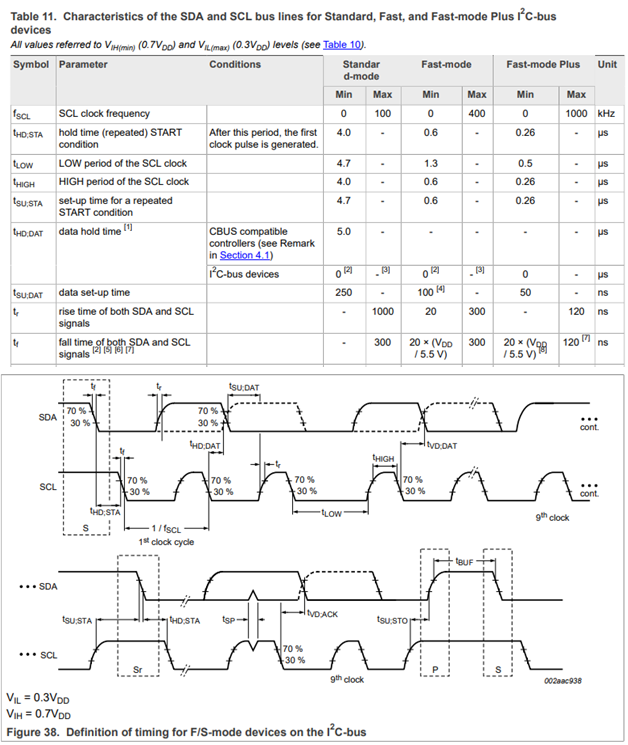
1. Download the FPGA firmware which supports the PCIe function to FPGA’s SPI flash.
2. Plug time card to main board.
3. At the PC host, run the Linux OS and then use the command “lspci -vv” to list the PCI bus and device.
4. Check the output message, we should find the FPGA device.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| Test PCIe link status for the FPGA | **Use the FPGA image version for x4 PCIe.**  **Test result:** Link speed = 2.5GT/s, Link width = x4. | Pass. |
|  |  |  |

* 1. I2C bus clock/data Tr, Tf measuring result.

**Requirement:** Test I2C bus clock/data rise time/fall time for all the I2C devices.



**Figure 6: I2C bus specification.**

**Test method: (**\* For this test case, we have connected the I2C EEPROM to the same bus with   
 PCA9546A I2C switch.)

1. Download the FPGA firmware which supports the PCIe function and FPGA I2C master function to the FPGA’s SPI flash.
2. Plug time card to main board.
3. At the PC host, run the Linux OS and then use the I2C command to access the I2C devices.
4. Measure the parameter Tr,Tf for I2C signals clock and data.

**\*This only measures Tr, Tf parameter. For the other parameters please see the SI signal measurement report.**

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| Measured at U35  P/N: LM75BDP,118  I2C ADDR: 0x49.  **Condition @100kHz:**  Tr < 1000ns.  Tf < 300ns |  | Pass. |
| Measured at U27  P/N: SHT31A  I2C ADDR: 0x44.  **Condition @100kHz:**  Tr < 1000ns.  Tf < 300ns |  | Pass. |
| Measured at U29  P/N: BNO085  I2C ADDR: 0x4A.  **Condition @100kHz:**  Tr < 1000ns.  Tf < 300ns |  | Pass. |
| Measured at U28  P/N: ICP-10100  I2C ADDR: 0x63.  **Condition @100kHz:**  Tr < 1000ns.  Tf < 300ns |  | Pass. |
| Measured at U7  P/N: AT24MAC602  I2C ADDR: 0x30, 0x50, 0x58  **Condition @100kHz:**  Tr < 1000ns.  Tf < 300ns | - We have connected the I2C eeprom to the same bus with I2C switch for this test case and change R1991,R19 Rpull-up for I2C eeprom from 4.7k to 100k.  - Set the PCA9546 I2C switch all channels = off.  - Thus, Rsum( Rpull-up I2C bus) = 4.48k. | Pass. |
|  |  |  |

* 1. Voltage droop at the VCC pin of the IC buffer for the FPGA<>SMA I/O signals.

**Requirement:** Measure the voltage droop at the VCC pin of IC buffer for the SMA I/O signals when the1PPS output is triggering.

**Test method:**

1. Connect probe1 to measure the output signal that drive from the FPGA at SMA connector.
2. Set the signal trigger for the oscilloscope by trig at the rising edge of the signal.
3. Measure the voltage at the VCC pin of the IC buffer measure point close to VCC pin.
4. Capture the signal.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| Measure the power rail voltage close to the VCC pin of U15 IC buffer. | **Oscilloscope CH-1:** 10Mhz signal which measured at J3 SMA connector.  **Oscilloscope CH-2:** Level voltage which measured at the VCC pin of the IC buffer (U15).    //======================================  Zoom in the rising edge of signal.    /======================================  Ripple voltage which measured at the VCC pin of the IC buffer U15. V(ripple) < 100mVp-p | Pass. |
| Measure the power rail voltage close to the VCC pin of U16 IC buffer. | **Oscilloscope CH-1:** CH-1: 1PPS signal which measured at J4 SMA connector.  **Oscilloscope CH-2:** Level voltage which measured at the VCC pin of the IC buffer (U16).    //======================================  Zoom in the rising edge of signal.    //======================================  Ripple voltage which measured at the VCC pin of the IC buffer U16. V(ripple) < 100mVp-p | Pass. |
|  |  |  |

* 1. FT4232 USB to UART/MPSSE [DVT].
     1. USB2.0 interface test for FT4232 [DVT].

**Requirement:** Test USB2.0 interfaces for the FT4232 USB<>MPSSE IC to confirm that the interface is working normally.

**Test method:**

1. Plug the Time card to the HPE server.
2. Power-on the HPE server.
3. Connect USB-C cable between the Time card and PC host.
4. At the PC host, run the Linux OS and then use the command “lsusb” to check that the PC host found the FT4232 device as normally.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| USB2.0 interface test for FT4232. |  | Pass. |
|  |  |  |

* + 1. FT4232: Channel-A MPSSE engine for JTAG interface test [DVT].

**Requirement:** Test the MPSSE JTAG interfaces for the FT4232 MPSSE engine to confirm that the JTAG interface is working normally.

**Test method:**

1. Plug the Time card to the HPE server.
2. Power-on the HPE server.
3. Connect USB-C cable between the Time card and PC host.
4. At the PC host, Use the API that supports JTAG scan to scan the FPGA on the Timecard board to confirm that the MPSSE engine for JTAG interface is working normally.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| MPSSE engine for JTAG interface  API: ftjrev from github. (Revise some to support FT4232 and gpiol) | Parameter: ID Code = 13631093 = xc7a100t. | Pass. |
|  |  |  |

* + 1. FT4232: Channel-A MPSSE engine for SPI interface test [DVT].

**Requirement:** Test the MPSSE SPI interfaces for the FT4232 MPSSE engine to confirm that the JTAG interface is working normally.

**Test method:**

1. Plug the Time card to the HPE server.
2. Power-on the HPE server.
3. Connect USB-C cable between the Time card and PC host.
4. At the PC host, use the API that support SPI interface between FTDI chip <> SPI flash on Timecard board to confirm that MPSSE engine for SPI interface is working normally.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| MPSSE engine for SPI interface  API: flashrom tool.  \* Since flashrom tool does not yet support the current SPI flash that we are using on the Timecard. Then, for this test we have changed SPI flash to p/n: MX25L12835F | Read    Write    Erase | Pass. |
|  |  |  |

* + 1. FT4232: Channel-B MPSSE engine for I2C interface test [DVT].

**Requirement:** Test the MPSSE I2C interfaces for the FT4232 MPSSE engine to confirm that the I2C interface is working normally.

**Test method:**

1. Power on the HPE server.
2. Connect USB-C cable between the Time card and PC host.
3. At the HPE server, run command i2cdump and then capture the result.
4. At the PC host, use the API that support I2C interface between FTDI chip <> I2C devices on the Timecard board to confirm that the MPSSE engine for I2C interface is working normally.
5. Compare the result.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| MPSSE engine for I2C interface test.  Use the "[**ftdi-i2c**](https://github.com/OriIdan/ftdi-i2c)/**i2cget.c**" C source file from github and then revise some to support FT4232 IC and add FT4232\_MUX3\_SEL signal on CH-A  to control  MUX switch for FTDI<> I2C devices. |  | Pass. |
|  |  |  |

* + 1. FT4232: Channel-C UART interface test [DVT].

**Requirement:** Test the UART interfaces for the engine to confirm that the FT4232 UART is working normally.

**Test method:** (Do not load firmware to FPGA for this test case)

1. On the Time card, connect the net name “UART\_FT4232\_TX\_FPGA\_RX” to J10 pin2, “UART\_FT4232\_TX\_FPGA\_TX” to J10 pin6. This connection is the FT4232 CH-C UART to MAC module UART.
2. Connect 12V power supply to connector P3.
3. Connect USB-C cable between the Time card and PC host.
4. At the PC host, open the UART terminal API. Set the serial terminal: baud = 57600, data bit = 8, Parity bit = none, Stop bit = 1, Flow control = none.
5. At the UART terminal API, send the character “c” to MAC module.
6. Capture the result that sends from MAC module.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| FT4232: Channel-C UART interface test. | CH-C = COM29 |  |
|  |  |  |

* + 1. FT4232: Channel-D UART interface test [DVT].

**Requirement:** Test the UART interfaces for the engine to confirm that the FT4232 UART is working normally.

**Test method:** (Do not load firmware to FPGA for this test case)

1. On the Time card, at the connector J10 put the jumper to connect pins1&2 and pins 5&6. This connection is the FT4232 CH-D UART to MAC module UART.
2. Connect 12V power supply to connector P3.
3. Connect USB-C cable between the Time card and PC host.
4. At the PC host, open the UART terminal API. Set the serial terminal: baud = 57600, data bit = 8, Parity bit = none, Stop bit = 1, Flow control = none.
5. At the UART terminal API, send the character “c” to MAC module.
6. Capture the result that is sent from the MAC module.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| FT4232: Channel-D UART interface test. | CH-D = COM30 |  |
|  |  |  |

* 1. U34 programmable delay time function [DVT]

**Requirement:** Test programmable delay time function on U34 which adjusts the delay time by R C on CT pin.

**Test method:**

1. Power-on the HPE server.
2. Use the oscilloscope to measure the FPGA\_IN\_RST\_DLY\_N to verify the delay time which programmed by U34’s CT pin (on the schematic: CT= 100nF, TD = 572ms.).

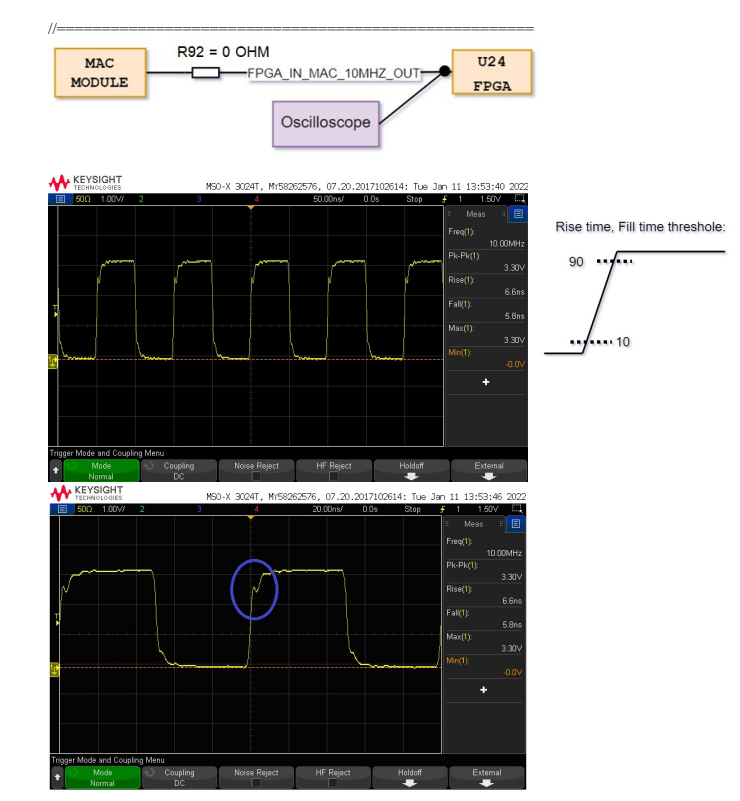
**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| U34 programmable delay time function. | TD ≈ 565ms | Pass. |
|  |  |  |

* 1. RC termination to reduce the signal reflection. [DVT]

* + 1. R termination for 10MHz from MAC module to FPGA [DVT].

**Requirement:** The EVT phase, we found the signal name FPGA\_IN\_MAC\_10MHZ\_OUT should adjust R termination (R92) to improve/reduce the signal reflection. For the DVT phase, perform measurement on the signal net name FPGA\_IN\_MAC\_10MHZ\_OUT and then check the result.



**Test method:**

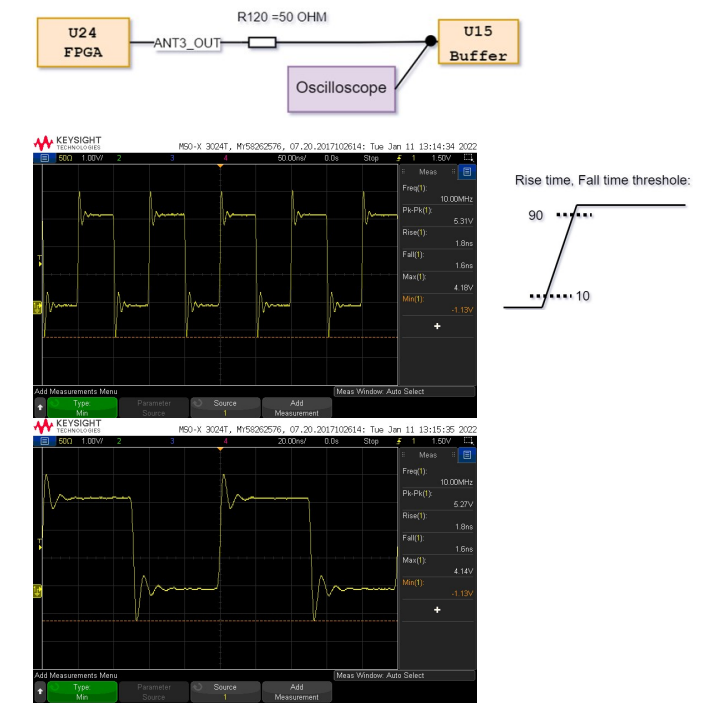
1. Power-on the HPE server.
2. Set oscilloscope for Tr,Tf threshold: Tr=30%, Tf=70%
3. Use the oscilloscope to measure the signal net name FPGA\_IN\_MAC\_10MHZ\_OUT, the measuring point close to the FPGA’s pin.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| U34 programmable delay time function. | ================EVT===================      ================DVT=================== |  |

* + 1. RC termination for 1PPS from FPGA to IC buffer [DVT]

**Requirement:** The EVT phase, we found the signals name ANT[4:1]\_OUT should adjust R series and add RC series parallel termination to improve/reduce the signal reflection. For the DVT phase, measure the signal net name ANT[4:1]\_OUT and then check the result.



**Test method:**

1. Power-on the HPE server.
2. Got to path: /sys/class/timecard/ocp0 and then use the command to set all the SMA = out , by command:

echo OUT: 10Mhz > sma1

echo OUT: 10Mhz > sma2

echo OUT: 10Mhz > sma3

echo OUT: 10Mhz > sma4

1. Set oscilloscope for Tr,Tf threshold: Tr=30%, Tf=70%
2. Use the oscilloscope to measure the signal net name ANT[4:1]\_OUT, the measuring point close to pin2 of the IC buffer.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| ANT1\_OUT | ================EVT===================    ================DVT=================== | Pass. |
| ANT2\_OUT | ================EVT===================    ================DVT=================== | Pass. |
| ANT3\_OUT | ================EVT===================    ================DVT=================== | Pass. |
| ANT4\_OUT | ================DVT=================== | Pass. |
|  |  |  |

1. Function test

This section we are require the FPGA firmware and DIAG command to test all the functions like

* FPGA I2C core interface to I2C EEPROM and sensor devices.
* GPIO.
* LED and RGB LED driver.
* FPGA<>SMA I/O signals to external devices.
* FPGA UART core interface to GPS and MAC module.
* USB interfaces FPGA<>MAC module.
* etc,

* 1. FPGA PCIe interfaces test

**Requirement:** Use the DIAG command to test the FPGA PCIe interface. The OS can detect the FPGA device and the PCIe link status should link-up with link width = (x1 or x 2 or x4) link speed = 2.5GT/s as normally.

**Test method:**

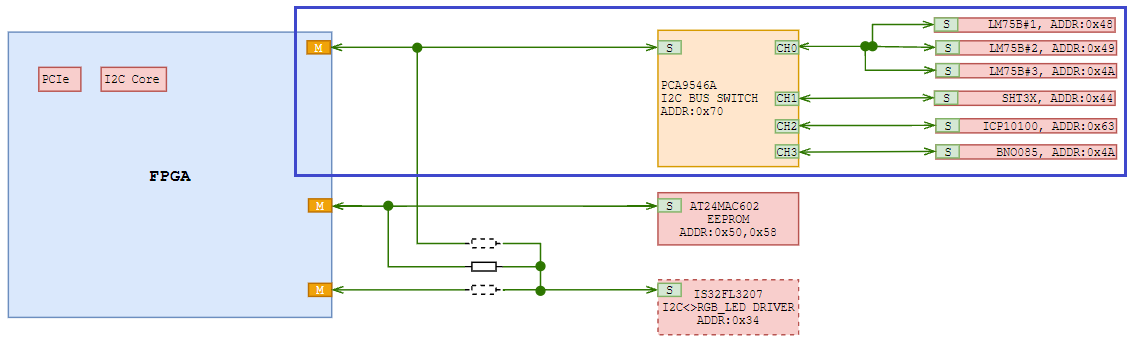
1. Download the FPGA firmware which supports the PCIe function to FPGA’s SPI flash.
2. Plug time card to main board.
3. At the PC host, run the Linux OS and then use the DIAG command to test the PCI bus and device.
4. Check the output message and then capture the output message.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| FPGA PCIe interfaces |  | PASS. |
|  |  |  |

* 1. FPGA I2C master <> I2C devices interface test
     1. FPGA I2C master <> I2C sensor devices interface test

**Requirement:** Test I2C interfaces for all the sensor devices by using FPGA as I2C master to confirm that the function FPGA I2C master<> I2C sensor device interfaces are working normally.



**Figure 7: FPGA I2C master <> I2C sensor devices interface.**

**Test method:**

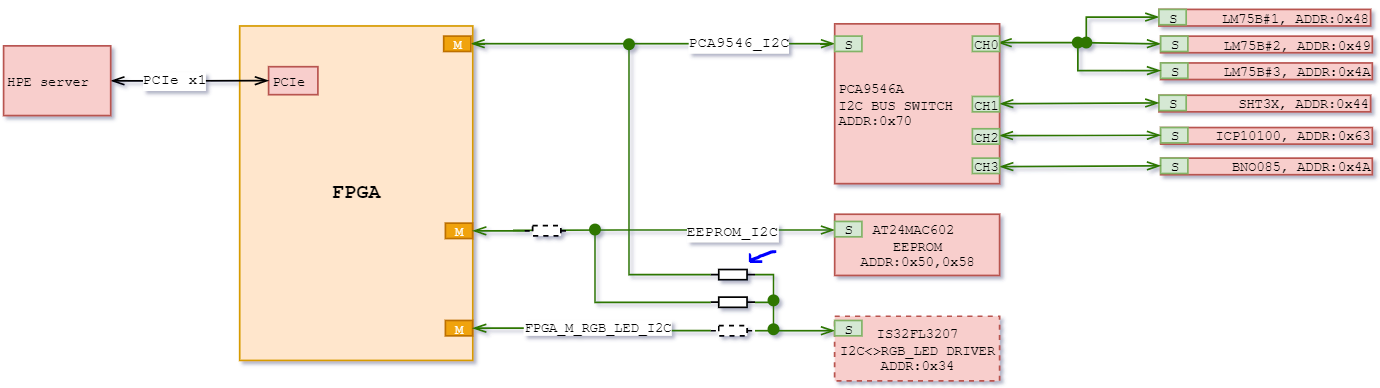
1. Plug the time card to the main board.
2. power on the then boot to Linux OS
3. At the PC host, run the Linux OS and install the package,driver that supports the FPGA I2C master<>I2C device .
4. Use the DIAG command to test each sensor to check that we can access all the sensors through the FPGA I2C master<>I2C normally and then capture the output message.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| Temperature sensors |  | PASS. |
| Humidity sensor |  | PASS. |
| Pressure sensor |  | PASS. |
| Vibration sensor |  | PASS. |
|  |  |  |

* + 1. FPGA I2C master <> I2C EEPROM devices interface test

**Requirement:** Test I2C interfaces for all the sensor devices by using FPGA as I2C master to confirm that the function FPGA I2C master<> I2C EEPROM interface is working as normally. This test, we have tested by using bus I2C-0. Then, pop R310, R311 to connect the EEPROM I2C bus to the I2C bus I2C-0.



**Figure 8: FPGA I2C master <> I2C EEPROM devices interface.**

**Test method:**

1. Plug the time card to the main board.
2. Power on the then boot to Linux OS
3. At the PC host, run the Linux OS and install the package and driver that support the FPGA I2C master<>I2C device.
4. Use the DIAG command to test the EEPROM to confirm that we can access the EEPROM by FPGA I2C master<>I2C EEPROM device as normally and then capture the output message.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| EEPROM |  | PASS. |
|  |  |  |

* 1. FPGA LED driver for RGB\_LED test

**Requirement:** Test FPGA<> LED indicator to confirm that the FPGA can control the USR\_LED, RGB\_LED as normally.

**Test method:**

1. Plug the time card to the main board.
2. Power on the then boot to Linux OS
3. At the PC host, run the Linux OS and install the package, and driver that support the FPGA LED driver function.
4. Use the DIAG command to test the RGB LED and observe the difference to confirm that FPGA<> RGB\_LED: DS14, DS15, DS16, DS17, DS18 function are working as normally.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| RGB\_LED |  | PASS. |
|  |  |  |

* 1. FPGA UART core interface to GPS and MAC module
     1. FPGA UART<> GPS module UART interfaces test

**Requirement:** Test FPGA UART<> GPS UART interfaces to confirm that the FPGA UART core can communicate with.

**Test method:**

1. Plug the time card to the main board.
2. Power on the then boot to Linux OS
3. At the PC host, run the Linux OS and install the package, driver that support the FPGA UART core function.
4. Use the DIAG command to test the FPGA UART core <> GPS UART interfaces to confirm that it is working as normally and then capture the output message.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| FPGA UART<> GPS UART interfaces |  | PASS. |
|  |  |  |

* + 1. FPGA UART<> MAC module UART interfaces test

**Requirement:** Test FPGA UART<> MAC module UART interfaces to confirm that the FPGA UART core can communicate with.

**Test method:**

1. Plug the time card to the main board.
2. Power on the then boot to Linux OS.
3. At the PC host, run the Linux OS and install the package, driver that support the FPGA UART core function.
4. Use the DIAG command to test the FPGA UART core <> MAC module UART interfaces to confirm that it is working as normally and then capture the output message.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| FPGA UART<> MAC module UART interfaces |  | PASS. |
|  |  |  |

* 1. Single Client PTP Test by phc2sys command

**Test method:**

1. Connect SFP+ cable to SFP+ port of Calnex Sentinel to the QSFP port on the Mellanox NIC (installed on the server).
2. Set-up the Calnex Sentinel equipment for PTP test.
3. At the HPE server, switch the console port to MAC module by using the command “tio –b 57600 /dev/ttyS6” and then set the parameter to:

\{set,Disciplining,1}

\{set,PpsWidth,80000000}

\{set,TauPps0,10000}

\{set,PpsOffset,-30}

\{set,DisciplineThresholdPps0,20}

\{store}

1. Warm –up the HPC server ~18 hrs. After that, at the HPE server, run ptp4l: ptp4l -i ens1f0 -f unicast-master.cfg –m and phc2sys: phc2sys -s /dev/ptp6 -c ens1f0 -O 0 -m.
2. Run test for 12 hrs.
3. Compare PTP times and verify that they are < ±2.5usec.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| 12 hrs single client PTP test by phc2sys command.  Timecard = Disciplining.  Sentinel = Disciplining (Internal Reference Disciplining Mode: Always, Disciplining source: GNSS).  \* GNSS antenna of Time card and Sentinel equipment = inside the building close to the window. |  | PASS. |
|  |  |  |

* 1. Single Client PTP Test by phc2sys command [DVT]

**Test method:**

1. Connect SFP+ cable to SFP+ port of Calnex Sentinel to the QSFP port on the Mellanox NIC (installed on the server).
2. Set-up the Calnex Sentinel equipment for PTP test.
3. At the HPE server, switch the console port to MAC module by using the command “tio –b 57600 /dev/ttyS6” and then set the parameter to:

\{set,Disciplining,1}

\{set,PpsWidth,80000000}

\{set,TauPps0,10000}

\{set,PpsOffset,-30}

\{set,DisciplineThresholdPps0,20}

\{store}

1. Warm –up the HPC server ~18 hrs. After that, at the HPE server, run run ptp4l: ptp4l -i ens1f0 -f unicast-master.cfg –m and phc2sys: phc2sys -s /dev/ptp6 -c ens1f0 -O 0 -m. Run test for 12 hrs.
2. Compare PTP times and verify that they are < ±2.5usec.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| 12 hrs single client PTP test by phc2sys command.  Timecard = Disciplining.  Sentinel = Disciplining (Internal Reference Disciplining Mode: Always, Disciplining source: GNSS).  \* GNSS antenna of Time card and Sentinel equipment = outside the building close to the window. | //====================================  //======Result for: \{set,PpsOffset,-30}=======  PTP 2WayTE mean value ≈ - 2.501us, exceeding the criteria ≈ 0.001us.      //======================================  //======Result for: \{set,PpsOffset,-2500}=======  PTP 2WayTE mean value ≈ 9.069ns. | PASS with condition by adjust the PpsOffset parameter on the MAC module. |

* 1. Single Client PTP Test by ts2phc command

**Test method:**

1. Connect SFP+ cable to SFP+ port of Calnex Sentinel to the QSFP port on the Mellanox NIC (installed on the HPE server).
2. Connect 1PPS output from Timecard to input of NIC card.
3. Set-up the Calnex Sentinel equipment for PTP test.
4. At the HPE server, switch the console port to MAC module by using the command “tio –b 57600 /dev/ttyS6” and then set the parameter to:

\{set,Disciplining,1}

\{set,PpsWidth,80000000}

\{set,TauPps0,1000}

\{set,PpsOffset, -30}

\{set,DisciplineThresholdPps0,20}

\{store}

1. Warm –up the HPC server ~18 hrs. After that, at the HPE server, run phc2sys to sync the TOD clock from Time card to system clock on HPE server by command “phc2sys -s /dev/ptp6 -c CLOCK\_REALTIME -O 0 -m”.
2. Set the 1PPS input pin for the NIC card by command: ./testptp -d /dev/ptp4 -L 0,1.
3. Stop the phc2sys command and then run ts2phc command to align the timestamp by command ts2phc -s generic -c ens1f0 -m -l 7.
4. Run ptp4l command “ptp4l -i ens1f0 -f unicast-master.cfg –m” to test PTP protocol with the Sentinel Equipment.
5. Run test for 12 hrs.
6. Compare PTP times and verify that they are < ± 500 ns.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| 12 hrs single client PTP test by ts2phc command.  Timecard = Disciplining.  Sentinel = Disciplining (Internal Reference Disciplining Mode: Always, Disciplining source: GNSS).  \* GNSS antenna of Time card and Sentinel equipment = inside the building close to the window. |  | PASS. |
|  |  |  |

* 1. Single Client PTP Test by ts2phc command [DVT]

**Test method:**

1. Connect SFP+ cable to SFP+ port of Calnex Sentinel to the QSFP port on the Mellanox NIC (installed on the HPE server).
2. Connect 1PPS output from Timecard to input of NIC card.
3. Set-up the Calnex Sentinel equipment for PTP test.
4. At the HPE server, switch the console port to MAC module by using the command “tio –b 57600 /dev/ttyS6” and then set the parameter to:

\{set,Disciplining,1}

\{set,PpsWidth,80000000}

\{set,TauPps0,1000}

\{set,PpsOffset, -30}

\{set,DisciplineThresholdPps0,20}

\{store}

1. Warm –up the HPC server ~18 hrs. After that, at the HPE server, run phc2sys to sync the TOD clock from Time card to system clock on HPE server by command “phc2sys -s /dev/ptp6 -c CLOCK\_REALTIME -O 0 -m”.
2. Set the 1PPS input pin for the NIC card by command: ./testptp -d /dev/ptp4 -L 0,1.
3. Stop the phc2sys command and then run ts2phc command to align the timestamp by command ts2phc -s generic -c ens1f0 -m -l 7.
4. Run ptp4l command “ptp4l -i ens1f0 -f unicast-master.cfg –m” to test PTP protocol with the Sentinel Equipment.
5. Run test for 12 hrs.
6. Compare PTP times and verify that they are < ± 500 ns.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| 12 hrs single client PTP test by ts2phc command.  Timecard = Disciplining.  Sentinel = Disciplining (Internal Reference Disciplining Mode: Always, Disciplining source: GNSS).  \* GNSS antenna of Time card and Sentinel equipment = outside the building close to the window. |  | PASS. |
|  |  |  |

* 1. 1PPS 24 Hour Holdover Test

**Requirement**: Verify that 1PPS from the timing card does not drift more than ±500usec over a 24 hr period when GNSS signal is lost.

**Test method:**

1. Connect the SMA cable from SMA3 (SMA3 the default = OUT PHC) of the Timecard module to input CH-C of the Calnex sentinel equipment.
2. Set-up the Calnex Sentinel equipment for 1PPS test.
3. At the HPE server, switch the console port to MAC module by using the command “tio –b 57600 /dev/ttyS6” and then set the parameter to:

\{set,Disciplining,1}

\{set,PpsWidth,80000000}

\{set,TauPps0,10000}

\{set,PpsOffset,-30}

\{set,DisciplineThresholdPps0,20}

\{store}

1. Warm-up the HPE server ~18 hrs. After that, set the parameter = \{set,Disciplining,0} for holdover mode.
2. Run test for 24 hrs.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| 1PPS 24 hrs holdover test.  Timecard = Stop disciplining.  Sentinel = Disciplining (Internal Reference Disciplining Mode: Always, Disciplining source: GNSS).  \* GNSS antenna of Time card and Sentinel equipment = inside the building close to the window. |  | PASS. |
|  |  |  |

* 1. 1PPS 24 Hour Holdover Test [DVT]

**Requirement**: Verify that 1PPS from the timing card does not drift more than ±500usec over a 24 hr period when GNSS signal is lost.

**Test method:**

1. Connect the SMA cable from SMA4 (For the DVT, the default for SMA4 = OUT PHC) of the Timecard module to input CH-B of the Calnex sentinel equipment.
2. Set-up the Calnex Sentinel equipment for 1PPS test.
3. At the HPE server, switch the console port to MAC module by using the command “tio –b 57600 /dev/ttyS6” and then set the parameter to:

\{set,Disciplining,1}

\{set,PpsWidth,80000000}

\{set,DisciplineThresholdPps0,20}

\{set,TauPps0, 10} ; Set Tau = 10 run 100sec.

\{set,TauPps0, 100} ; Set Tau = 100 run 20min.

\{set,TauPps0, 5000} ; Set Tau = 5,000 discipline time >= 6hr 56min).

1. Discipline time >= 6hr 56min. this test disciplining time ~12 hrs. After that, set the parameter = \{set,Disciplining,0} for holdover mode.
2. Run test for 24 hrs.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| 1PPS 24 hrs holdover test.  Timecard = Stop disciplining.  Sentinel = Disciplining (Internal Reference Disciplining Mode: Stop discipline during measurement, Disciplining source: GNSS).  \* GNSS antenna of Time card and Sentinel equipment = outside the building near to the window.  \*\* This test, we have monitored the digital tuning value and average value for the digital tuning. When discipline time > 5Tau, send command to stop the disciplining at the digital tuning value move close to midpoint [average line]. |  | PASS. |
|  |  |  |

* 1. Multiple PTP client load test.

**TBD**

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
|  |  |  |
|  |  |  |

* 1. Single PCIE delay test.

**Requirement**: Verify that the offset when synchronizing the timing card with the NIC over the PCIe bus is <+-5usec)

**Test method:**

1. Connect Calnex Sentinel to the PPS output of the timing card.
2. Connect the other port of Calnex sentinel to PPS output of Mellanox NIC.
3. Run the utility phc2sys (phc2sys -s /dev/ptp6 -c ens1f0 -O 0 -m).
4. Observe the difference.

**Test result.**

|  |  |  |
| --- | --- | --- |
| **Test Item** | **Result** | **Pass/Fail** |
| ens1f0 PHC offset < ±5usec | PHC offset <± 5us | Pass. |
|  |  |  |