**Elop Condor MS-C10A SBC Board**

**Software Requirements Specifications (SRS)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name and Title** | **Signature** | **Date** |
| Issued by: | Lior Yosef, R&D Engineer |  | First Issue |
| Michael Khomyakov, Firmware Engineer |  | 10/06/2024 |
| Approved by: | Bar Shpilman, R&D Team Leader |  |  |

**Effective Date:** 01/10/2024 **Last Updated By:** Michael K.

Date 15-04-2024

# Contents

[1. Contents 1](#_Toc177032128)

[2. General Description 3](#_Toc177032129)

[2.1. System Architecture Diagram 3](#_Toc177032130)

[2.2. Module Design and Requirements Description 4](#_Toc177032131)

[2.2.1. Sensing Modules 4](#_Toc177032132)

[2.2.2. DC/DC Boards 12](#_Toc177032133)

[2.2.3. Relay Module 21](#_Toc177032134)

[2.2.4. Fans Control Module 22](#_Toc177032135)

[2.2.5. Power Status Indication Module 24](#_Toc177032136)

[2.2.6. External Interfaces 25](#_Toc177032137)

[2.2.7. MIU Interface 27](#_Toc177032138)

[3. Runtime Operation Specifications 29](#_Toc177032139)

[3.1. Start-Up Sequence 29](#_Toc177032140)

[3.1.1. Overview 29](#_Toc177032141)

[3.1.2. System Power ON Sequence Requirements 29](#_Toc177032142)

[3.2. System Reset 31](#_Toc177032143)

[3.2.1. Specific Requirements 31](#_Toc177032144)

[3.3. Shutdown Sequences 32](#_Toc177032145)

[3.3.1. Overview 32](#_Toc177032146)

[3.3.2. Standard Shutdown Specific Requirements 33](#_Toc177032147)

[3.3.3. Emergency Shutdown Specific Requirements 34](#_Toc177032148)

[3.4. Protection Specifications 35](#_Toc177032149)

[3.4.1. Over Voltage Protection (OVP) 35](#_Toc177032150)

[3.4.2. Under Voltage Protection (UVP) 36](#_Toc177032151)

[3.4.3. Over Temperature Protection (OTP) 38](#_Toc177032152)

[4. Communication Interface 39](#_Toc177032153)

[5. Log File functionality 40](#_Toc177032154)

[5.1. General description: 40](#_Toc177032155)

[5.2. PSU Logfile handling - general rules 41](#_Toc177032156)

[6. PSU Status 64-bit Register 43](#_Toc177032157)

[6.1. Overall Description 43](#_Toc177032158)

[6.2. Detailed Description 43](#_Toc177032159)

# General Description

## System Architecture Diagram

The following diagram provides a complete description of Architectural design of the SBC board:

User LED [MIO47]

Program Flash Memory

DDR3 DRAM

ON/OFF

QSPI

DDR3

**Zynq 7000**

**System on Chip**

DC/DC Boards Interface

ON/OFF Logic

eMMC

Module

SD Card Interface

Power Good Logic

Power In Status

RS485 (RX, TX, DE)

Power Status

Module

Power Out Status

Fan Control Modules

FAN Control PWM

Aircraft Interface

Speed Sense PWM

Lamp Status

ON/OFF Logic

Pod Status

AC and DC

Current Sense

Reset

Full Duplex SPI

MIU Interface Module

Shutdown

Voltage Sense

Emergency

Spare Logic

Full Duplex SPI

XADC PS

12V FPGA Power Input Sense

MDC 100MbE

Ethernet PHY Module

Sensing Module

ON/OFF Logic

TCU Module

QSPI

Time & S/N Storage Memory

External Interfaces

ON/OFF Logic

Figure 1 - SBC Block diagram with specified interfaces

Relay Module

## Module Design and Requirements Description

### Sensing Modules

#### Module A

The module is based on ADS7951, 8-Channel 12-bit A/D converter, while communication with the module is performed by Full-Duplex SPI interface.

The SPI interface is implemented according to the following pinout diagram on the FPGA:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Package Pin | SPI Protocol Name | FPGA Direction |
| I\_sns\_ADC\_CS\_fpga | V7 | NSS (Active Low) | OUT |
| I\_sns\_ADC\_SCLK\_fpga | T9 | SCK | OUT |
| I\_sns\_ADC\_SDI\_fpga | U10 | MOSI | OUT |
| I\_sns\_ADC\_SDO\_fpga | Y7 | MISO | IN |

The following table describes the A/D module analog inputs alignment:

|  |  |  |
| --- | --- | --- |
| Name | A/D Channel | Description |
| DC\_PWR\_I\_sns | Channel 0 | Output DC Power Current |
| PH1\_I\_sns | Channel 1 | Phase 1 AC Current |
| PH2\_I\_sns | Channel 2 | Phase 2 AC Current |
| PH3\_I\_sns | Channel 3 | Phase 3 AC Current |
| 28V\_IN\_sns | Channel 4 | 28V Input Voltage |
| *Not Used* | Channel 5 | *Not Used* |
| *Not Used* | Channel 6 | *Not Used* |
| *Not Used* | Channel 7 | *Not Used* |

The ADS7951 is a 12-bit 8-channel Analog to Digital converter module, operated in a full-duplex SPI mode with a periodic sampling feature. The following is the sequence for ADS7951 operation:

1. **FPGA Initialization:**
   1. Rising Edge SCK for valid MOSI (CPOL= 0).
   2. Falling Edge SCK for valid MISO (CPHA = 0).
   3. Active LOW chip select (NSS).
   4. During the transactions: NSS = LOW.
   5. No transactions: NSS = HIGH (Deselected).
2. **ADS7950 Initialization:**
   1. Transmit Frame #1: 0x8000.
   2. Transmit Frame #2: 0xFFFF.
   3. Transmit Frame #3: 0x2800.
3. After the initialization frames are transmitted to the A/D module, periodic sampling can be performed as follows:
   1. Transmit Frame #3: 0x2800 and receive data on the same frame (16-bit frame).
   2. Decode the received 16-bit frame as follows: <Channel><Data>.
      1. <Channel> Represents the number of sampled channel [0...7].
      2. <Data> Represents the 12-bit sampled value [0...4095] from a specified channel.

**Example for data Reception frame-by-frame:**

1. Transmit = 0x2800 (MOSI), Receive = 0x1AAA
   1. Sampled Channel[0…7]: Channel 1 (0x10) – PH1\_I\_sns.
   2. Sampled Data[0:0xFFF]: AAA.
2. Transmit 0x2800 (MOSI), Receive = 0x2BBB
   1. Sampled Channel[0…7]: Channel 2 (0x20) – PH2\_I\_sns.
   2. Sampled Data[0:0xFFF]: 0xBBB.

#### Module B

The module is based on ADS7951, 8-Channel 12-bit A/D converter, while communication with the module is performed by Full-Duplex SPI interface. The SPI interface is implemented according to the following pinout diagram on the FPGA:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Package Pin | SPI Protocol Name | FPGA Direction |
| HV\_ADC\_CS\_fpga | M15 | NSS (Active Low) | OUT |
| HV\_ADC\_SCLK\_fpga | L14 | SCK | OUT |
| HV\_ADC\_SDI\_fpga | L15 | MOSI | OUT |
| HV\_ADC\_SDO\_fpga | M14 | MISO | IN |

The following table describes the A/D module analog inputs alignment:

|  |  |  |
| --- | --- | --- |
| Name | A/D Channel | Description |
| OUT4\_sns | Channel 0 | Single-Phase Output Voltage |
| Vsns\_PH1 | Channel 1 | Phase 1 Input Voltage |
| Vsns\_PH2 | Channel 2 | Phase 2 Input Voltage |
| Vsns\_PH3 | Channel 3 | Phase 3 Input Voltage |
| OUT4\_Isns | Channel 4 | Single-Phase Output Current |
| Vsns\_PH\_C\_RLY | Channel 5 | Phase 3 Relay Output Voltage |
| Vsns\_PH\_B\_RLY | Channel 6 | Phase 2 Relay Output Voltage |
| Vsns\_PH\_A\_RLY | Channel 7 | Phase 1 Relay Output Voltage |

The ADS7951 is a 12-bit, 8-channel Analog to Digital converter module, operated in a full-duplex SPI mode with a periodic sampling feature. The following is the sequence for ADS7951 operation:

1. **FPGA Initialization:**
   1. Rising Edge SCK: MOSI (CPOL= 0).
   2. Falling Edge SCK: MISO (CPHA = 0).
   3. Active LOW chip select (NSS).
   4. During the transactions: NSS = LOW.
   5. No transactions: NSS = HIGH (Deselected).
   6. Each frame is 2 bytes (16-bit).
2. **ADS7951 Initialization:**
   1. Transmit Frame #1: 0x8000.
   2. Transmit Frame #2: 0xFFFF.
   3. Transmit Frame #3: 0x2800.
3. After the initialization frames are transmitted to the A/D module, periodic sampling can be performed as follows:
   1. Transmit Frame #3: 0x2800 and receive data on the same frame (16-bit frame).
   2. Decode the received 16-bit frame as follows: <Channel><Data>.
      1. <Channel> Represents the number of sampled channel [0...7].
      2. <Data> Represents the 12-bit sampled value [0...4095] from a specified channel.

**Example for data Reception frame-by-frame:**

1. Transmit = 0x2800 (MOSI), Receive = 0x2AAA
   1. Sampled Channel[0…7]: 2 – Vsns\_PH\_C\_RLY.
   2. Sampled Data[0:0xFFF]: AAA.
2. Transmit 0x2800 (MOSI), Receive = 0x3BBB
   1. Sampled Channel[0…7]: 3 – OUT4\_Isns.
   2. Sampled Data[0:0xFFF]: 0xBBB.

#### Sampled Data Conversion

Since all the sampled values represent the raw 12-bit A/D measurements, the values shall be converted to the valid data in accordance with the specified channel and value implemented in the design.

1. **Voltage and Current Measurement**

To convert the values from unsigned A/D raw 12-bit data to 16-bit signed value, the following equation is used:

Since the IRS document provided by customer provides a specific set of requirements for the scale factor and range, the following table shall be implemented:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | Type | Name in Log File | Name in FPGA | Units | ADC Module | | Parameter A | | Parameter B | Type of Measurement |
| 3 | S2 | VAC\_IN\_PH\_A | Vsns\_PH1 | 100mV | Module B Channel 7 | | 1.0889 | | 2047 | RMS |
| 4 | S2 | VAC\_IN\_PH\_B | Vsns\_PH2 | 100mV | Module B Channel 6 | |
| 5 | S2 | VAC\_IN\_PH\_C | Vsns\_PH3 | 100mV | Module B Channel 5 | |
| 12 | S2 | V\_OUT\_3\_ph1 | Vsns\_PH\_A\_RLY | 100mV | Module B Channel 1 | |
| 13 | S2 | V\_OUT\_3\_ph2 | Vsns\_PH\_B\_RLY | 100mV | Module B Channel 2 | |
| 14 | S2 | V\_OUT\_3\_ph3 | Vsns\_PH\_C\_RLY | 100mV | Module B Channel 3 | |
| 15 | S2 | V\_OUT\_4 | OUT4\_sns | 100mV | Module B Channel 0 | |
| 7 | S2 | I\_AC\_IN\_PH\_A | PH1\_I\_sns | 100mA | Module A Channel 1 | | 0.14652 | | 2252 |
| 8 | S2 | I\_AC \_IN\_PH\_B | PH2\_I\_sns | 100mA | Module A Channel 2 | |
| 9 | S2 | I\_AC \_IN\_PH\_C | PH3\_I\_sns | 100mA | Module A Channel 3 | |
| 24 | S2 | I\_OUT\_3\_ph1 | PH1\_I\_sns | 100mA | *Defined in the next section* | | | | |  |
| 25 | S2 | I\_OUT\_3\_ph2 | PH2\_I\_sns | 100mA |  |
| 26 | S2 | I\_OUT\_3\_ph3 | PH3\_I\_sns | 100mA |  |
| 27 | S2 | I\_OUT\_4 | OUT4\_Isns | 100mA | Module B Channel 4 | | 0.092 | | 3412 | RMS |
| 34 | S2 | AC\_Power | N/A | 1W | See specific Requirements section **2.2.1.3.B** | | | | | |
| 2 | S2 | VDC\_IN | 28V\_IN\_sns | 50mV | Module A Channel 4 | 0.362637 | | 0 | | Instantaneous |
| 6 | S2 | I\_DC\_IN | DC\_PWR\_I\_sns | 50mA | Module A Channel 0 | 0.32 | | 2047 | |

1. **Power Calculation:**

The calculated power is a sum of all the three (3) phases of current multiplied by voltage. The following is an example for power calculation:

|  |  |  |  |
| --- | --- | --- | --- |
| Time | Instantaneous Voltage  [100mV] | Instantaneous Current [50mA] | Instantaneous Power |
| 100us |  | 20 |  |
| 200us | 200 | 40 |  |
| 300us | 300 | 60 |  |
| 400us | 400 | 80 |  |
| 500us | 500 | 100 |  |
| 600us | 600 | 120 |  |
| 700us | 700 | 140 |  |
| 800us | 800 | 160 |  |
| 900us | 900 | 180 |  |
| 1ms | 1000 | 200 |  |

#### 12V FPGA Power Input Sense

The system implements an integrated A/D used to continuously monitor the input voltage supplied to the FPGA. This module is used to mitigate the safety concerns regarding the overall Power Supply operation and unpredictive hardware behavior. The following is the A/D based interface:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Package Pin | Description | FPGA Direction |
| VP\_0 | K9 | Positive terminal of XADC Input | IN |
| VN\_0 | L10 | Negative terminal of XADC input | IN |

The 12V power supply is sensed through the VP\_0 line, while the VN\_0 is connected to the 0V rail (Ground). The sensed 12V voltage is calculated as follows:

Hence the A/D measurement performed by XADC with reference voltage of 1.8[V], is calculated as follows:

#### 12V FPGA Power Input Sense Specific Requirements

1. If drops below 10V, hence the measurement is below **1300d**, the FPGA should immediately halt the eMMC writing and close the log file.
2. The A/D sampling rate shall be 100KSamples/second for only one single channel.

#### Sensing Module Specific Requirements

1. The firmware shall sample the values with a sample rate of at least **10KSamples/sec.** (25 Points on the complete 400[Hz] AC cycle).
2. The firmware shall calculate the RMS values in accordance with the calculated zero crossings to determine the cycles timing (**See appendix A: Zero Crossing Detection**):
   1. **Voltage Calculation:** Three (3) consecutive cycles of AC voltage.

RMS value that is being transferred through Running average.

* 1. **Current Calculation:** Three (3) consecutive cycles of AC current.

RMS value that is being transferred through Running average.

* 1. **Average Power Calculation:** Total power calculation for 3-Phase input. The power shall be calculated by taking an average of 10 instantaneous Current and voltage points per 1[millisecond] for **each phase**: The following is the equation to use in the implementation:

1. The firmware shall provide the main application manager (Runtime) access to all the input data received from the Sensing Module with a sampling rate of at least **1KSamples/sec**. for all the measurements.
2. The Sensing Module shall perform periodic self-test and provide an output to the Runtime application for the following events:
   1. The voltage sensing module does not communicate through SPI correctly.
   2. The current sensing module does not communicate through SPI correctly.

**Sequence Example:**

The number of points for the sampling frequency was set to 25 points per 2.5mS cycle: Hence the number of points for the calculation (N) shall be implemented as a function of input frequency as described in the following:

1. Determine the signal period (Refer to Appendix A: Zero Crossing Detection):
2. Determine the number of points:
3. Calculate the , and values.

### DC/DC Boards

#### Interface Description

|  |  |  |
| --- | --- | --- |
| Name | Description | FPGA Direction |
| RS485\_RxD\_i | UART Transmit Data | IN |
| RS485\_TxD\_i | UART Receive Data | OUT |
| RS485\_DE\_i | UART Data Direction | OUT |
| EN\_PSU\_i | Board Enable (Active High) | OUT |
| PG\_PSU\_i | Board Power Good (Active High) | IN |

The system contains 9 operating board units with implementation of the logical control and RS485 communication features. The following is the RS485-based interface and logical control for each board:

The “i” character in the line alias name represents the index of the DC/DC board, whereas by implemented design.

#### Logical Communication

1. **Logical output EN\_PSU\_i:** The Board Enable signal is active HIGH logical output from the FPGA, that provides Turn ON / Turn OFF commands to the DC/DC or Main Board outputs from the System.
2. **Logical input PG\_PSU\_i:** The Board Power Good signal is active HIGH logical input to the FPGA. This input is an indication for a FPGA, that the corresponding board status is OK (Driven HIGH) or FAIL (Driven LOW).

#### RS485 Communication

1. RS485\_DE\_i: The Data Enable signal is a RS485 data direction selection FPGA output used in communication handling.
2. RS485\_RxD\_i: The signal is the UART data receive line input to the FPGA.
3. RS485\_TxD\_i: The signal is the UART data transmit line output from the FPGA.

|  |  |  |
| --- | --- | --- |
| Data | State | FPGA Direction |
| RS485\_DE\_i = HIGH | Data can be transmitted by UART TxD, Reception on UART RxD is disabled | FPGA to Board |
| RS485\_DE\_i = LOW | Data can be received by UART RxD, Transmission on UART TxD is disabled | Board To FPGA |

The following is the description of data flow selection implemented in the hardware design:

#### Complete Pinout and Board Arrangement Description

The following describes the complete description for the DC/DC Boards & Main Board Interface:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FPGA Line** | **Index in FPGA** | **Package Pin** | **Direction** | **Actual Board** |
| RS485\_RxD\_1 | 0 | R19 | IN | DCDC\_1 |
| RS485\_TxD\_1 | T11 | OUT |
| RS485\_DE\_1 | T10 | OUT |
| EN\_PSU\_1\_FB | Y19 | OUT |
| PG\_PSU\_1\_FB | Y18 | IN |
| RS485\_RxD\_2 | 1 | T12 | IN | DCDC\_2 |
| RS485\_TxD\_2 | U12 | OUT |
| RS485\_DE\_2 | N20 | OUT |
| EN\_PSU\_2\_FB | W16 | OUT |
| PG\_PSU\_2\_FB | V16 | IN |
| RS485\_RxD\_5 | 2 | R14 | IN | DCDC\_5 |
| RS485\_TxD\_5 | Y16 | OUT |
| RS485\_DE\_5 | Y17 | OUT |
| EN\_PSU\_5\_FB | R17 | OUT |
| PG\_PSU\_5\_FB | R16 | IN |
| RS485\_RxD\_6 | 3 | W14 | IN | DCDC\_6 |
| RS485\_TxD\_6 | Y14 | OUT |
| RS485\_DE\_6 | T16 | OUT |
| EN\_PSU\_6\_FB | R18 | OUT |
| PG\_PSU\_6\_FB | T17 | IN |
| RS485\_RxD\_7 | 4 | U17 | IN | DCDC\_7 |
| RS485\_TxD\_7 | V15 | OUT |
| RS485\_DE\_7 | W15 | OUT |
| EN\_PSU\_7\_FB | V18 | OUT |
| PG\_PSU\_7\_FB | V17 | IN |
| RS485\_RxD\_8 | 5 | U14 | IN | DCDC\_8 |
| RS485\_TxD\_8 | U15 | OUT |
| RS485\_DE\_8 | U18 | OUT |
| EN\_PSU\_8\_FB | W19 | OUT |
| PG\_PSU\_8\_FB | W18 | IN |
| RS485\_RxD\_9 | 6 | U19 | IN | DCDC\_9 |
| RS485\_TxD\_9 | N18 | OUT |
| RS485\_DE\_9 | P19 | OUT |
| EN\_PSU\_9\_FB | P18 | OUT |
| PG\_PSU\_9\_FB | N17 | IN |
| RS485\_RxD\_10 | 7 | T14 | IN | DCDC\_10 |
| RS485\_TxD\_10 | T15 | OUT |
| RS485\_DE\_10 | P14 | OUT |
| EN\_PSU\_10\_FB | P16 | OUT |
| PG\_PSU\_10\_FB | P15 | IN |
| RS485\_RxD\_Buck | 8 | V13 | IN | Main Board |
| RS485\_TxD\_ Buck | V12 | OUT |
| RS485\_DE\_ Buck | W13 | OUT |
| EN\_PFC\_FB | U20 | OUT |
| PG\_BUCK\_FB | T20 | IN |

#### Communication Protocol: DC/DC Boards and Main Board

The communication between the FPGA and the DC/DC boards is implemented as a three (3) step process:

1. FPGA sets the Data Enable (DE) High.
2. FPGA Transmits the data request frame (0xAA).
3. FPGA sets the Data Enable (DE) Low.
4. FPGA Receives the 9-byte data response from the DC/DC board.

The data request frame from the FPGA to the DC/DC board shall be implemented as follows:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data Request Frame from Master to Slave (Table 1) | | | | | | | | |
| Bytes: 1 | Bit 7  (MSB) | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 (LSB) |
| Request Frame | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |

#### Received Frame Description: DC/DC Boards

The data response from the **DC/DC board** to the FPGA is implemented as follows:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data Frames from Slave to Master (Table 2) | | | | | | | | |
| Bytes:  Total: 9 | Bit 7 (MSB) | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 (LSB) |
| Byte 1: | Temp.  bit 7 | Temp.  bit 6 | Temp.  bit 5 | Temp.  bit 4 | Temp.  bit 3 | Temp.  bit 2 | Temp.  bit 1 | Temp.  bit 0 (LSB) |
| Byte 2: | Vin  bit 11 (MSB) | Vin  bit 10 | Vin  bit 9 | Vin  bit 8 | 0  0 | 0  0 | 0  0 | 0  0 |
| Byte 3: | Vin  bit 7 | Vin  bit 6 | Vin  bit 5 | Vin  bit 4 | Vin  bit 3 | Vin  bit 2 | Vin  bit 1 | Vin  bit 0 (LSB) |
| Byte 4: | Vout bit 7 | Vout bit 6 | Vout bit 5 | Vout bit 4 | Vout bit 3 | Vout bit 2 | Vout  bit 1 | Vout  bit 0 (LSB) |
| Byte 5: | Iin  bit 11 (MSB) | Iin  bit 10 | Iin  bit 9 | Iin  bit 8 | Vout  bit 11 (MSB) | Vout  bit 10 | Vout  bit 9 | Vout  bit 8 |
| Byte 6: | Iin  bit 7 | Iin  bit 6 | Iin  bit 5 | Iin  bit 4 | Iin  bit 3 | Iin  bit 2 | Iin  bit 1 | Iin  bit 0 (LSB) |
| Byte 7: | Iout bit 7 | Iout bit 6 | Iout bit 5 | Iout bit 4 | Iout bit 3 | Iout bit 2 | Iout  bit 1 | Iout  bit 0 (LSB) |
| Byte 8: | OVP | OCP | OTP | VinP | Iout  bit 11 (MSB) | Iout  bit 10 | Iout  bit 9 | Iout  bit 8 |
| Byte 9: | crc8 bit 7 (MSB) | crc8 bit 6 | crc8 bit 5 | crc8 bit 4 | crc8 bit 3 | crc8 bit 2 | crc8 bit 1 (LSB) | crc8 bit 0 |

#### Data Description – DC/DC Boards

1. Temp[7:0]: The raw 8-bit [0…255] DC/DC board temperature sample.
2. Vin[12:0]: The raw 12-bit [0…4095] DC/DC board input voltage sample.
3. Vout[12:0]: The raw 12-bit [0…4095] DC/DC board output voltage sample.
4. Iin[12:0]: The raw 12-bit [0…4095] DC/DC board input current sample.
5. Iout[12:0]: The raw 12-bit [0…4095] DC/DC board output current sample.
6. res: Reserved bit.
7. crc8[8:0]: The 8-bit CRC of the eight (8) consecutive bytes (Not including the CRC byte).
8. OVP, OCP, OTP, VinP: The 1-bit status data for each one of the corresponding events in the selected DC/DC board.

Since all the sampled values represent the raw 12-bit A/D measurements, the values shall be converted to the 16-bit signed data in accordance with the specified channel and value implemented in the design. The following table is the values of parameters required for conversion for each DC/DC board (“i” represents DC/DC board number):

|  |  |  |
| --- | --- | --- |
| Name | Value Type | Expression |
| Vin[i] | Voltage | Vin[i] = 0.56 \* Sample |
| Vout[i] | Voltage | Vout[i] = 0.28 \* Sample |
| Iin[i] | Current | Iin[i] = 0.32234 \* Sample |
| Iout[i] | Current | Iout[i] = 0.32234 \* Sample |
| Temp[i] | Temperature | Temp[i] = Sample |

#### Received Frame Description: Main board

The data response from the **Main board** to the FPGA is implemented as follows:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data Frames from Slave (Buck micro) to Master (SBC FPGA) | | | | | | | | |
| Bytes:  Total: 9 | Bit 7  (MSB) | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 (LSB) |
| Byte 1: | Temp.  bit 7 | Temp.  bit 6 | Temp.  bit 5 | Temp.  bit 4 | Temp.  bit 3 | Temp.  bit 2 | Temp.  bit 1 | Temp.  bit 0 (LSB) |
| Byte 2: | IphA  bit 11 (MSB) | IphA  bit 10 | IphA  bit 9 | IphA  bit 8 | 0 | 0 | 0 | 0 |
| Byte 3: | IphA  bit 7 | IphA  bit 6 | IphA  bit 5 | IphA  bit 4 | IphA  bit 3 | IphA  bit 2 | IphA  bit 1 | IphA  bit 0 (LSB) |
| Byte 4: | IphB bit 7 | IphB bit 6 | IphB bit 5 | IphB bit 4 | IphB bit 3 | IphB bit 2 | IphB  bit 1 | IphB  bit 0 (LSB) |
| Byte 5: | IphC  bit 11 (MSB) | IphC  bit 10 | IphC  bit 9 | IphC  bit 8 | IphB  bit 11 (MSB) | IphB  bit 10 | IphB  bit 9 | IphB  bit 8 |
| Byte 6: | IphC  bit 7 | IphC  bit 6 | IphC  bit 5 | IphC  bit 4 | IphC  bit 3 | IphC  bit 2 | IphC  bit 1 | IphC  bit 0 (LSB) |
| Byte 7: | OVP | OCP | OTP | VinP | res | res | res | Cap\_EOL |
| Byte 8: | crc8 bit 7 (MSB) | crc8 bit 6 | crc8 bit 5 | crc8 bit 4 | crc8 bit 3 | crc8 bit 2 | crc8 bit 1 (LSB) | crc8 bit 0 |
| Byte 9: | 0  bit 7 (MSB) | 0  bit 6 | 0  bit 5 | 0  bit 4 | 0  bit 3 | 0  bit 2 | 0  bit 1 (LSB) | 0  bit 0 |

#### Data Description – Main Board

1. Temp[7:0]: The converted 8-bit [0…255] DC/DC board temperature sample.
2. IphA[12:0], IphB[12:0], and IphC[12:0]: The raw 12-bit [0…4095] RMS values of phase voltage input current samples on the Main Board.
3. crc8[8:0]: The 8-bit CRC of the eight (8) consecutive bytes (Not including the CRC byte).
4. OVP, OCP, OTP, VinP: The 1-bit status data for each one of the corresponding events in the selected DC/DC board.
5. Cap\_EOL: the 1-bit status data for The Capacitor end-of-life.

The following table is the values of parameters required for conversion for Main Board:

|  |  |  |
| --- | --- | --- |
| Name | Value Type | Expression |
| MAIN\_I\_PH1 | Current | MAIN\_I\_PH1 = Sample |
| MAIN\_I\_PH2 | Current | MAIN\_I\_PH2 = Sample |
| MAIN\_I\_PH3 | Current | MAIN\_I\_PH3 = Sample |
| Capacitor EOL | Logic | EOL = 1 or 0 |
| Temp | Temperature | Temp = Sample |

#### Data Collection Example – DC/DC Boards and Main Board

The following is an example flow of the data collection from all the DC/DC boards:

1. Set RS\_485\_DE\_x = HIGH.
2. Send “Data Request Character” 0xAA to the DC/DC board or Main board.
3. Set RS\_485\_DE\_x = LOW.
4. Receive and store the acquired packets (9-byte sequence).
5. Convert the values for all the boards in accordance with value conversion tables.
6. Make all the values accessible to Runtime.

#### DC/DC Boards Measurements and Log File compatibility

The following table provides summary for the calculations and final log file appearance, including the units used. To convert the values from DC/DC boards received by UART to 16-bit signed value, the following equation is used:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Type | Name in Log File | Units | DC/DC Board | Parameter A |
| 10 | S2 | V\_OUT\_1 | 50mV | DCDC1 | 0.28 |
| 11 | S2 | V\_OUT\_2 | DCDC2 |
| 16 | S2 | V\_OUT\_5 | DCDC5 |
| 17 | S2 | V\_OUT\_6 | DCDC6 |
| 18 | S2 | V\_OUT\_7 | DCDC7 |
| 19 | S2 | V\_OUT\_8 | DCDC8 |
| 20 | S2 | V\_OUT\_9 | DCDC9 |
| 21 | S2 | V\_OUT\_10 | DCDC10 |
| 22 | S2 | I\_OUT\_1 | 50mA | DCDC1 | 0.32234 |
| 23 | S2 | I\_OUT\_2 | DCDC2 |
| 28 | S2 | I\_OUT\_5 | DCDC5 |
| 29 | S2 | I\_OUT\_6 | DCDC6 |
| 30 | S2 | I\_OUT\_7 | DCDC7 |
| 31 | S2 | I\_OUT\_8 | DCDC8 |
| 32 | S2 | I\_OUT\_9 | DCDC9 |
| 33 | S2 | I\_OUT\_10 | DCDC10 |
| 40 | S1 | T1 | 1C | DCDC1 | 1 |
| 41 | S1 | T2 | DCDC2 |
| 42 | S1 | T3 | DCDC10 |
| 43 | S1 | T4 | MAIN\_BOARD |
| 44 | S1 | T5 | DCDC5 |
| 45 | S1 | T6 | DCDC6 |
| 46 | S1 | T7 | DCDC7 |
| 47 | S1 | T8 | DCDC8 |
| 48 | S1 | T9 | DCDC9 |
| 24 | S2 | I\_OUT\_3\_ph1 | 100mA | I\_OUT\_3\_ph1 = (I\_AC \_IN\_PH\_A – MAIN\_I\_PH1) | N/A |
| 25 | S2 | I\_OUT\_3\_ph2 | I\_OUT\_3\_ph2 = (I\_AC \_IN\_PH\_B – MAIN\_I\_PH2) |
| 26 | S2 | I\_OUT\_3\_ph3 | I\_OUT\_3\_ph3 = (I\_AC \_IN\_PH\_C – MAIN\_I\_PH3) |

#### DC/DC and Main Board Specific Requirements

1. The firmware shall request the data from the DC/DC boards and Main Board the values with a sample rate of at least **1KSamples/sec for each board**:
   1. The baud rate for the communication shall be 115200 Baud/sec.
   2. Number of stop bits shall be 1.
   3. Parity shall be NONE.
2. Each data transfer shall be done in accordance with RS485 Hardware flow (RS485\_DE\_x) terminal shall be driven HIGH on data request and driven LOW on data reception for each board.
3. The firmware shall provide the main application manager (Runtime) access to all the input data received from the Sensing Module with a sampling rate of at least **1KSamples/sec**. for each channel.
4. The boards in the firmware shall be mapped in accordance with their RS485 connection in the schematic diagram implemented in hardware (refer to mapping table between the FPGA UART interface and the DC/DC boards).
5. The logical lines (PG and EN) shall be the **same** as the actual board number (Only the communication is remapped).
6. The calculation constants shall be implemented in an upper application layer, so they can be modified outside the module:
   1. Each board that is can be used in firmware. For Example:
      1. DCDC\_1\_USED = ‘1’, DCDC\_2\_USED = ‘0’, etc.
   2. Explanation: Since the integration process is evolutionary, the number of boards used with RS485 communication support will rise sequentially.

### Relay Module

#### General Description

The Relay Module is responsible for enabling the AC output to be used by the customer. The module is controlled by a single logical signal as follows:

3 Phase Input

3-Phase Output

Relay Module

1-Phase Output

Relay Enable: Package Pin **K16**

#### Relay Module Specific Requirements

1. The relay module shall be turned ON or OFF by driving the line Relay\_3ph\_fpga HIGH or LOW.

### **Fans Control Module**

#### General description:

The module contains three (3) fans controlled by integrated circuits (ICs), while the control and sensing are provided according to the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Package Pin | Fan Number | Description | Direction |
| FAN\_CTRL1\_fpga | K17 | 1 | Fan 1 Control Signal | OUT |
| FAN\_HALL1\_fpga | L16 | Fan 1 Speed Frequency Sense | IN |
| FAN\_CTRL2\_fpga | E18 | 2 | Fan 2 Control PWM | OUT |
| FAN\_HALL2\_fpga | D19 | Fan 2 Speed Frequency Sense | IN |
| FAN\_CTRL3\_fpga | E17 | 3 | Fan 3 Control PWM | OUT |
| FAN\_HALL3\_fpga | B19 | Fan 3 Speed Frequency Sense | IN |

#### Control Description

All the three (3) fans can be controlled separately by providing the PWM signal to the control lines. The fan speed is controlled by generating a PWM signal on FAN\_CTRLx\_fpga in accordance with the following equation:

The typical curve for the implemented fan implicates that in the frequency region of around 900[Hz] the controlled fan speed has a linear behavior, as it can be seen in the following figure:



#### Speed Measurement

The fan module is rated for **22000 RPM** for its full speed (100%) value. The sensing mechanism is based on frequency measurement on the FAN\_HALLx\_fpga line. The following expression provides the measured fan speed derived from the implemented hardware:

#### Closed loop description

The following is the procedure of the closed loop for the fan speed control, including the implementation of the measurements:

* 1. Define a setpoint for the speed of fan (in range from 0 to 100%).
  2. Set Duty Cycle to 50% at 900[Hz] frequency at FAN\_CTRLx\_fpga line.
  3. Measure the frequency of FAN\_HALLx\_fpga.
  4. Convert the frequency to the RPM using the 2.2.4.3 equation.
  5. Increase or decrease the duty cycle of the FAN\_CTRLx\_fpga based on the measured RPM.

**Example:**

1. Defining the setpoint for the speed of each fan for **80%** of its maximum speed:

1. Duty cycle was set to 50% at 900[Hz] frequency at FAN\_CTRLx\_fpga line.
2. The measured frequency of FAN\_HALLx\_fpga was 295[Hz].
3. Converting the frequency to the RPM:
4. The actual speed of the fan is 100 RPM higher than the setpoint – The duty cycle needs to be slowly decreased.
5. If the actual speed was lower than the setpoint, the duty cycle must be slowly increased.
6. Repeat the steps iii-vi.

#### Fans Control Module Specific Requirements

1. Each fan shall be able to receive an independent PWM signal through FAN\_CTRLx\_fpga lines with a fixed frequency of 900[Hz], while the pulse width of each signal defines the fan speed.
2. Each fan speed output FAN\_HALLx\_fpga shall be treated as frequency output.

### Power Status Indication Module

#### General description:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Package Pin | Direction | Description |
| P\_in\_status\_fpga | G18 | OUT | PSU input voltage indication line. |
| P\_out\_status\_fpga | J20 | OUT | PSU output voltage indication line. |

The module provides 2 status lines to the customer LED panel. The following is the description of the lines implemented:

#### Specific Requirements

* 1. P\_in\_status\_fpga shall be HIGH if the following conditions are met:
     1. 108[V] < Vsns\_PH1 < 118[V].
     2. 108[V] < Vsns\_PH2 < 118[V].
     3. 108[V] < Vsns\_PH3 < 118[V].
     4. 22[V] < 28V\_IN\_sns < 30[V].
  2. If one or more conditions are not met – the Pin\_status\_fpga shall be LOW.
  3. P\_out\_status\_fpga shall be HIGH if the following conditions are met:
     1. 108[V] < Vsns\_PH\_A\_RLY < 118[V].
     2. 108[V] < Vsns\_PH\_B\_RLY < 118[V].
     3. 108[V] < Vsns\_PH\_C\_RLY < 118[V].
     4. 108[V] < OUT4\_sns < 118[V].
     5. DC/DC boards output range is defined in accordance with the following table:

|  |  |  |
| --- | --- | --- |
| DC/DC Board | FPGA Reference | Range |
| 1 | VOUT\_1 | 34.92V-37.08V |
| 2 | VOUT\_2 | 30.0425V-30.9575V |
| 5 | VOUT\_5 | 27.16V-28.84V |
| 6 | VOUT\_6 | 27.16V-28.84V |
| 7 | VOUT\_7 | 27.16V-28.84V |
| 8 | VOUT\_8 | 27.16V-28.84V |
| 9 | VOUT\_9 | 27.16V-28.84V |
| 10 | VOUT\_10 | 27.16V-28.84V |

* 1. If one or more conditions are not met – the P\_out\_status\_fpga shall be LOW.

### External Interfaces

#### Overview

A screenshot of a computer

Description automatically generatedThe following diagram provides an overall view of the PSU external interfaces, while the blue interface connection lines are referred to as FPGA connectivity:

#### Aircraft Interface

The following table illustrates the FPGA interface with the controls from the Aircraft Interface:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Package Pin | Direction | Description |
| PowerON\_fpga | V5 | IN | Dry contact switch from the cockpit (PSU ON/OFF Switch) |
| pod\_status\_fpga | G17 | IN | The pod status is the logic to the cockpit lamp. The lamp status is measuring the actual voltage at the cockpit lamp to make sure the lamp and lamp driver are not shorted. |
| lamp\_status\_fpga | J15 | OUT | The lamp indication on the control panel in the cockpit |
| ECTCU\_INH\_fpga | H20 | OUT | TCU External Inhibit Line 1 |
| CCTCU\_INH\_fpga | J19 | OUT | TCU External Inhibit Line 2 |

#### Aircraft Interface Specific Requirements

1. The Firmware shall include an internal checking mechanism for the AC and the DC inputs, while the following is the definition:
   1. VIN\_DC\_28V\_STATUS = HIGH:
      1. 18[V] < 28V\_IN < 32[V].
      2. Otherwise, LOW.
   2. VIN\_AC\_115V\_STATUS = HIGH:
      1. 95[V] < Vsns\_PH1 < 125[V].
      2. 95[V] < Vsns\_PH2 < 125[V].
      3. 95[V] < Vsns\_PH3 < 125[V].
      4. Otherwise, LOW.
2. The PSU shall provide a discrete indication to a lamp on the control panel in the cockpit, the following table summarizes the required behavior of the output discrete to the LED according to the required inputs:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Inputs | | | | Output |
| VIN\_DC\_28V\_STATUS | VIN\_AC\_115V\_STATUS | PowerON\_fpga | MIU Ethernet Communication | lamp\_status\_fpga |
| LOW | LOW | N/A | N/A | LOW |
| HIGH | LOW | N/A | N/A | Square Wave 1[Hz] |
| LOW | HIGH | N/A | N/A | LOW |
| HIGH | HIGH | LOW | N/A | HIGH |
| HIGH | HIGH | HIGH | No communication | Square Wave 4[Hz] |
| HIGH | HIGH | HIGH | Communication established | LOW |

### MIU Interface

#### General Description

This module integrates a critical component of the system. The MIU interface is consists of the following:

MIU

(Customer)

Reset\_Out\_fpga

MIU Interface Module

Shutdown\_Out\_fpga

EShutdown\_Out\_fpga

Spare\_Out\_fpga

Ethernet PHY Module

MDC 100MbE

The module consists of 4 logical lines, that can be driven HIGH or LOW and an 100Mbit/s Ethernet interface (MDC). The following is the description of the module lines:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Package Pin | Direction | Description |
| Reset\_Out\_fpga | H15 | OUT | Reset request output to MIU. The terminal provides reset request signals to MIU according to the Runtime Specification. |
| Shutdown\_out\_fpga | G15 | OUT | Standard shutdown indication to MIU. The terminal provides an indication of a **standard shutdown** procedure to the MIU, according to Runtime Specification. |
| EShutdown\_Out\_fpga | K14 | OUT | Emergency shutdown indication to MIU. The terminal provides an indication of an **emergency shutdown** procedure to the MIU, according to Runtime Specification. |
| Spare\_Out\_fpga | J14 | OUT | Spare logical output to the MIU, that was requested by customer and its operation can be implemented in the future. |
| Ethernet PHY 100MbE | PS | I/O | Standard 100Mbit/s Ethernet communication interface with the MIU. The Ethernet communication enables the control over the data stream of the PSU, the requests from the PSU and the operation definitions according to the Runtime specification. |

#### Ethernet Interface

The PSU implements an Ethernet communication interface with the MIU interface, while the PSU shall be able to maintain continuous UDP connections with responsiveness to the MIU interface commands. The PSU IP address shall be 192.168.1.30 and the specific requirements are described in the Customer IRS document: “**IRS MIU-PSU-V1**”.

# Runtime Operation Specifications

## Overview

The Runtime defines the system behavior and consistency of the inputs, outputs, events and overall reaction of the system. Each section of state defined for Runtime consists of the following.

* 1. Conditions for state transitions.
  2. Procedure to be performed.

## System Runtime State Diagram missing an arrow to connect IDLE state to Power on condition.

## This is not a state diagram this is a flow chart. Please change it to state diagram (flow chart is SW oriented and state diagram is HW oriented. In flow chart you (the CPU) can actually be in the question and not in the state but in HW you can only be in the state you ask the questions inside the state. You write the transition condition on the arrow itself)

Idle State

Power ON condition was met?

No

Yes

## 

Yes

System Power ON Sequence

No

Is POWERON\_FPGA LOW?

No

Successful sequence?

Emergency Shutdown State

Yes

System is ON

No

Is Reset condition met?

Emergency Shutdown Procedure

Yes

Reset Sequence

No

Is shutdown condition met?

Yes

Shutdown Sequence

Yes

No

Is emergency shutdown condition met?

No

## Start-Up Sequence

### Overview

The FPGA shall be able to turn the entire PSU system ON or OFF in response to:

1. Dry contact switch in the cockpit (Aircraft Interface) position.
2. Emergency Shutdown Event.

The following is the states defined as a source for initialization of the PSU system

Short = ON (POWERON\_FPGA = HIGH).

Open = OFF (POWERON\_FPGA = LOW).

Missing here the IDLE state transitions and IDLE state outputs values

### Conditions for System Power ON Sequence

* + - 1. The previous system state is **Idle** state.no need for previous state just next state
      2. POWER\_ON\_FPGA = HIGH.

### System Power ON Sequence

* 1. 50ms Switch debounce mechanism. In the following section b-j you are missing what happens if power on switch (after debounce goes down what to do? I used to ignore it in the old state machine.)
  2. You also need to specify in b-j what happens if inputs (A2D and UARTs) go out of range. I assume you would like to ignore it and check it only if system state is on, but it used to be that I check it. But if I check it there should be a list of what to check in each phase as they turn on in sequence.
  3. The following lines shall be as follows:
     + 1. ECTCU\_INH\_fpga = LOW.
       2. CCTCU\_INH\_fpga = LOW.
       3. reset\_out\_fpga = HIGH.
       4. shutdown\_out\_fpga = HIGH.
       5. Eshutdown\_out\_fpga = HIGH.
       6. There used to be here a requirement for the temperature to be in range I commented this requirement for now.
       7. There used to be here a requirement for the V/A values to be in range but it doesn’t make sense so I commented it out.
       8. There should be here a requirement for the sampling A2D and Uarts to be active. Or do you want the power on to turn on on all the modules before checking this ????
  4. Set the Main Board enable: EN\_PFC\_FB = HIGH and start an elapsed time measurement.
  5. Turn on the Fans module by providing a PWM signal to all the three (3) fans with 55% duty cycle.
  6. Start measuring the fan speed.
  7. Start an elapsed time measurement. This was done in (section c.) above
  8. Measure the line PG\_BUCK\_FB.
  9. If more than 20 seconds elapsed and PG\_BUCK\_FB was driven LOW – missing here “or fans are not OK” then: the system shall enter emergency shutdown state.
  10. If fans are at the correct speed and PG\_BUCK\_FB is driven HIGH, the following lines shall be set as follows:
      1. EN\_PSU\_1\_FB = HIGH
      2. EN\_PSU\_2\_FB = HIGH
      3. EN\_PSU\_5\_FB = HIGH
      4. EN\_PSU\_6\_FB = HIGH
      5. EN\_PSU\_7\_FB = HIGH
      6. EN\_PSU\_8\_FB = HIGH
      7. EN\_PSU\_9\_FB = HIGH
      8. EN\_PSU\_10\_FB = HIGH
      9. Relay\_3ph\_fpga = HIGH
  11. The System Power ON Sequence is treated as finished, if the following conditions are met:
      1. The power good (PG) logical lines are driven HIGH by all the used DC/DC boards and Main Board in accordance with the following:
         1. PG\_PSU\_1\_FB = HIGH
         2. PG\_PSU\_2\_FB = HIGH
         3. PG\_PSU\_5\_FB = HIGH
         4. PG\_PSU\_6\_FB = HIGH
         5. PG\_PSU\_7\_FB = HIGH
         6. PG\_PSU\_8\_FB = HIGH
         7. PG\_PSU\_9\_FB = HIGH
         8. PG\_PSU\_10\_FB = HIGH
      2. The measured fan speed module (3 units) is more than 1000RPM for each fan. (in previous section it already says fans should be in correct value. the correct place it to put the speed value in the fans module and here only to say speed is correct) should say fans speed is still OK or remove

if you repeat this requirement you should also repeat the need for buck pg.

* + 1. The Relay Module measurements meet the requirements as follows:
       1. Vsns\_PH\_A\_RLY - Vsns\_PH1 < 20V
       2. Vsns\_PH\_B\_RLY - Vsns\_PH2 < 20V
       3. Vsns\_PH\_C\_RLY - Vsns\_PH3 < 20V
       4. Should you also wait for all the inputs to be in range? I assume not, but it used to be in the state machine.

you are missing here what to do if 20 sec has passed and not all requirements are met. I am leaving here the old behavior that the eshoutdown is initiated after 20 SEC.

* 1. If System Power ON Sequence is finished and successful:
     1. Wait for 1 millisecond.
     2. Set the logical values for interface:
        1. ECTCU\_INH\_fpga = HIGH
        2. CCTCU\_INH\_fpga = HIGH
        3. reset\_out\_fpga = LOW. These 3 used to go low before 1 MSEC so i changed it to be after 1 MSEC.
        4. shutdown\_out\_fpga = LOW.
        5. Eshutdown\_out\_fpga = LOW.
        6. To prevent a dead lock I added that before we enter the system is on state we wait for all inputs to be in range, so if they are not in range yet (still settling) the system will wait. But if 20 sec has passed it will go into eshutdown enyhow.
        7. If we don’t do it, the system will go emedially to power on state and if something is out of range it will go to eshutdown. It may cause the system to never power on.
     3. Set the state to **System is ON.**
  2. If System Power ON Sequence fails, initiate the Emergency Shutdown procedure. This is too general and has no actual influence on the statemachine.

## System Reset

### Conditions for System Reset Sequence

The following sequence describes the system reset performed by the Aircraft Interface switch:

Note that all requirements for poweron\_fpga are after 50 ms debounce

1. System is currently in **system is ON** state.
2. The POWERON\_FPGA = LOW detected.
3. Start elapsed time measurement.
4. POWERON\_FPGA = HIGH detected.
5. If the POWERON\_FPGA = HIGH was detected within the time frame of

**50[milliseconds] < Elapsed Time < 6[seconds]:**

**Add the conditions to go to shutdown/eshutdown in the middle of the reset sequence**

**I left it as it was, that an eshutdown event can interrupt the reset sequence.**

### System Reset Sequence

* + 1. The PSU shall not disable the output voltages.
    2. The PSU shall generate a reset sequence by driving the MIU interface line reset\_out\_fpga line HIGH for 5ms ± 10%.
    3. System shall return to **system is ON** state.

## **Shutdown Sequences**

### Overview

System shutdown implementation consists of two different sequences: standard shutdown and emergency shutdown.

#### Standard Shutdown Conditions

* 1. The system is in **System is ON** state.
  2. If the POWERON\_FPGA = LOW was detected for more than 6 seconds, the PSU shall enter a Power OFF sequence.

### Standard Shutdown Procedure

in current code if eshutdown happens during waiting 6 sec we go into eshutdown. I am leaving the code as is.

Note that all requirements for poweron\_fpga are after 50 ms debounce

In the old code during power down I used to monitor the out of range(limits) but now I removed it, so during power down it will not go into eshutdown on limit violation. (if you need elsewere it should be declared how and what to check for the limit violations)

it is not clear what to do with reset and shutdown. So since we arive here from power on state where these signals are not active I leave them not active. Once the shutdown finishes, it will go into idle state were these signals will go to active state again.

1. If the System Shutdown procedure was initiated, the POWERON\_FPGA line shall be ignored for 10 seconds after the PSU finishes the procedure (effectively 26 seconds after the switch has been turned off since the first 6 seconds are ignored and then an additional 10 seconds are needed to power down).
2. Standard shutdown procedure shall be performed when POWERON\_FPGA was driven LOW for a time above 6[seconds] or after an OTP event.
3. shutdown\_out\_fpga shall be driven HIGH for 10[seconds].
4. After the step (C), the shutdown\_out\_fpga shall be driven LOW.
5. All the DC/DC boards shall be disabled by driving the following lines LOW:
   1. EN\_PSU\_1\_FB = LOW
   2. EN\_PSU\_2\_FB = LOW
   3. EN\_PSU\_5\_FB = LOW
   4. EN\_PSU\_6\_FB = LOW
   5. EN\_PSU\_7\_FB = LOW
   6. EN\_PSU\_8\_FB = LOW
   7. EN\_PSU\_9\_FB = LOW
   8. EN\_PSU\_10\_FB = LOW
6. The relay module shall be disabled: Relay\_3ph\_fpga = LOW
7. Main Board shall be disabled: EN\_PFC\_FB = LOW.
8. CCTCU = LOW and ECTCU = LOW.
9. 1 minute after step (H), the fans control module shall be disabled. This used to be 10 minutes, I changed the code according to the new requirement
10. System shall enter the **Idle** state. This is not after stage (I) above rather after 26 sec from power\_on\_fpga low or 20 sec after otp.

#### Emergency Shutdown **Conditions**

Emergency shutdown shall be initiated, if one or more of the following conditions are met:

Please add what system states can trigger eshutdown if these events happen. For now I implemented that during turning on power that takes more than 20 sec. During power on state or detection of pweron\_fpga low (waiting to see if there is reset or power down I am still able to receive eshutdown)

Not during power down or idle.

1. Power Good signal (PG) from one or more DC/DC boards is driven LOW.
2. Power Good signal (PG) from the Main board is driven LOW.
3. Relay module is turned ON and the measurements of Relay Module are not valid (See section 2.2.3).
4. Sensing module is not communicating correctly. This wasn’t in the old spec, I will add it.
5. DC/DC boards or Main board do not communicate correctly. This wasn’t in the old spec, I will add it.
6. Fan speed is below 1000RPM.
7. Input OVP/UVP condition.
8. Output OVP condition.
9. Missing: power on sequence takes more than 20 sec should also initiate eshutdown

### Emergency Shutdown Procedure

1. If the Emergency Shutdown Procedure was initiated, the **EShutdown\_Out\_fpga line shall be driven HIGH for 51[milliseconds].**
2. **Missing instruction what to do with reset and powerdown. So the reset and powerdown are driven as their state before entering eshutdown until idle state. Once in idle state they will go to active.**
3. The PSU Shall disable the power supplies according to the following hold-up sequence:
   1. Immediately:
      1. EN\_PSU\_1\_FB = LOW.
      2. EN\_PSU\_9\_FB = LOW.
      3. Relay\_3ph\_fpga = LOW.
   2. 10[milliseconds] after EShutdown\_Out\_fpga = HIGH:
      1. EN\_PSU\_2\_FB = LOW.
      2. EN\_PSU\_6\_FB = LOW.
      3. EN\_PSU\_8\_FB = LOW.
      4. EN\_PSU\_10\_FB = LOW.
   3. 20[milliseconds] after EShutdown\_Out\_fpga = HIGH:
      1. EN\_PSU\_7\_FB = LOW.
   4. 50[milliseconds] after EShutdown\_Out\_fpga = HIGH:
      1. EN\_PSU\_5\_FB = LOW.
      2. EN\_PFC\_FB = LOW.
      3. ECTCU\_INH\_fpga = LOW. This used to be after 51 mili, I changed it to 50
      4. CCTCU\_INH\_fpga = LOW. This used to be after 51 mili, I changed it to 50
   5. System shall enter the **Emergency Shutdown** state.
   6. If the POWERON\_FPGA is LOW, the system shall enter **Idle** state.
   7. Fans are turned off after 1min after initiating eshutdown. This follows old spec and I leave it as it was. (it used to be 10 min but I set it as it is in the shutdown mode so the 2 modes have the same fans behavior)
   8. In the old spec there was a difference between what is the reason for entering eshutdown and how to get out of this state.
   9. If out of range (limits) error than wait for that to be OK then wait for 950 miliseconds and then go to idle. If other reason (power on sequence fail, pg not good, fans out of range,) than wait for poweron\_fpga (after debounce) to go low then wait for poweron\_fpga (after debounce) to be high for 6 seconds and then go to idle mode
   10. I could remove all this logic but I am not sure if this is a mistake or on purpose so it is easier to leave it in the code waiting for your decision.
   11. The condition for uart ans spi communication to be good, I put with the limits option regarding how to exit the eshutdown

General regarding the power on state:

In the old spec if poweron\_fpga was low for 26 seconds the complete state machine was overridden and the state would go to IDLE state.

Since it’s not in the spec I put it in comment, (I am really not in favor of such override to the state machine) so if you really need it let me know and I will return it into the code.

## **Protection Specifications**

### Over Voltage Protection (OVP)

#### Overview

Please reevaluate how this is related to the main power on /off state and statemachine

|  |  |  |
| --- | --- | --- |
| Voltage | O.V. Trip Point | Detection Response Time |
| 115VAC | 125VAC | 100msec |
| 28VDC | 35VDC | 2ms |
| 30.5VDC | 31.5VDC | 2ms |
| 36VDC | 45 VDC | 2ms |

The following is the Over Voltage Protection trip point specifications from the customer SPEC, generalizing the input and output voltages from the PSU, that are continuously measured:

#### Over Voltage Protection Specific Requirements

1. The following is the Over Voltage Protection trip point primary specifications, that shall be implemented:

|  |  |  |
| --- | --- | --- |
| Measured Lines | OVP Tripping Point | Tripping Time  [milliseconds] |
| * + - 1. Vsns\_PH\_A\_RLY       2. Vsns\_PH\_B\_RLY       3. Vsns\_PH\_C\_RLY       4. Vsns\_PH1       5. Vsns\_PH2       6. Vsns\_PH3       7. OUT4\_sns | V > 125[V] | 100 |
| * + - 1. Vsns\_28V\_IN | V > 32[V] | 5000 |

1. If the OVP event occurs, the system shall initiate the Standard Shutdown sequence. Should be eshutdown
2. If the system Input voltages are within the specified range for at least 1 second, the system shall initiate the Power On sequence.
3. The following is the Over Voltage Protection **Output** trip point specifications, that shall be implemented:

|  |  |  |
| --- | --- | --- |
| Measured Lines | OVP Tripping Point | Tripping Time  [milliseconds] |
| * + - 1. VOUT\_1 | V > 45[V] | 2 |
| * + - 1. VOUT\_2 | V > 31.5[V] |
| * + - 1. VOUT\_5       2. VOUT\_6       3. VOUT\_7       4. VOUT\_8       5. VOUT\_9       6. VOUT\_10 | V > 35[V] |

1. If the output OVP event occurs, the system shall initiate the Emergency Shutdown procedure.
2. If the dry-contact switch (PowerOn\_fpga) line was driven LOW and driven HIGH back again, the system shall initiate the Power On sequence.

### Under Voltage Protection (UVP)

#### Overview

Please reevaluate how this is related to the main power on /off state and statemachine

All main inputs shall be under-voltage protected. If any input voltage is in an under-voltage state, for more than the rated duration, the PSU shall handle the operation accordingly. Eshutdown

#### Under Voltage Protection AC Specific Requirements

1. All three phases of input AC voltages shall be under-voltage protected according to the following table:

|  |  |  |
| --- | --- | --- |
| Measured Lines | Tripping Point | Tripping Time  [milliseconds] |
| Vsns\_PH1  Vsns\_PH2  Vsns\_PH3 | V < 90[V] | 0 (Immediate UVP Event) |
| 90[V] < V < 95[V] | 500 |
| V > 95[V] | Normal Operation |

1. After the voltage returns to the nominal value for at least 1 second, the PSU shall initiate the Power ON Procedure in accordance with PowerOn\_fpga line status.

#### Under Voltage Protection DC Specific Requirements

1. The 28V input voltage shall be under-voltage protected according to the following table:

|  |  |  |
| --- | --- | --- |
| Measured Line | Voltage Range | Tripping Time  [milliseconds] |
| 28V\_IN\_sns | V < 17[V] | 0 (Immediate UVP Event) |
| 17[V] < V < 18[V] | 500 |
| V > 18[V] | Normal Operation |

### Over Temperature Protection (OTP)

#### Overview

The PSU shall implement a protection mechanism to protect itself from the possibility of overheating.

#### Over Temperature Protection Specific Requirements

#### The firmware shall detect the OTP event by continuous measurements of DC/DC boards and Main Board temperature. The following table provides the OTP event values for each board separately:

|  |  |  |
| --- | --- | --- |
| **Temperature value FPGA Reference** | **OTP Tripping Point** | **After OTP Event Point** |
| T4 | 100°C | 80°C |
| T1 | 110°C | 90°C |
| T2 | 110°C | 90°C |
| T5 | 110°C | 90°C |
| T6 | 110°C | 90°C |
| T7 | 110°C | 90°C |
| T8 | 110°C | 90°C |
| T9 | 110°C | 90°C |
| T3 | 110°C | 90°C |

#### If OTP event occurs, The PSU shall initiate the Emergency Shutdown (contradiction, I think you mean shut down and not emergency shut down – I am leaving it at shutdown and not emergency shut down)

#### procedure, while the order of operations is as follows: (is this here for a reason? Should there be a difference in shutdown due to OTP. I assume not so remove this section. Just say to to shout down state.

#### The PSU shall notify the MIU via communication interface.

#### Shutdown\_out\_fpga line must be driven HIGH.

#### After the Shutdown\_out\_fpga line was driven HIGH, the PSU shall wait 10 seconds, so the MIU can finish its own shutdown sequence.

#### The system shall initiate the Standard Shutdown procedure.

#### If the temperature drops back below a safe level (20[degrees] below OTP value), the PSU shall initiate a start-up sequence. Only if poweron\_fpga is high. Actually it goes into idle state and then anyhow if the poweron\_fpga is high it will turn on. This is redundant just say go to idle state and in idle state it dennds on otp to be OK to start

1. The temperatures shall be sampled from DC/DC boards and Main Board according to DC/DC Boards specifications: the resolution of the temperature is the integer value in degrees [0…255], while the temperature is calculated as follows:

# Communication Interface

The PSU shall have an Ethernet communication interface with MIU, which shall provide the following capabilities (detailed PSU-MIU communication protocol shall be defined in a separate IRS document, PSU IP address shall be 192.168.1.60):

1. Providing periodic statuses and telemetry (@100Hz) with time tag, including:
   * Input/output voltages (for each phase separately and for DC voltage)
   * Current consumptions (also for each phase & DC voltage, not including Neutral)
   * Total input (AC+DC) power consumption
   * Internal temperature of every internal card
   * Missing phase or missing neutral indication
   * PSU Fans speed for each of the 3 fans
   * Software/firmware version (format shall be “Major.Minor.Build.Hotfix”)
   * ETI (Elapsed Time Indication)
   * PSU serial number
   * Additional bit statuses such as:
     1. Reset or Shutdown indications.
     2. Input voltage OK indication.
     3. TCU ON/OFF status
     4. MIU communication OK/NOK
     5. Overvoltage & overcurrent indications
     6. Capacitor end-of-life indications

Since status changes can occur intermittently, in order not to miss the occurrence, the faulty notification shall be latched until sent to the MIU via communication channel. Only after the faulty notification was sent to MIU, it will be cleared.

1. Log file downloading from PSU (using SFTP) via dedicated command.
2. Control of TCUs ON/OFF by MIU via PSU
3. MIU shall provide the time at power-up for log file time stamp synchronization. The purpose is to correlate between events in the MIU and events in the PSU, during a flight.

# Log File functionality

## General description:

1. The PSU shall gather telemetry and status data at a high rate and store it internally. This is required to capture any transient anomalies in case they occur. Detailed logfile parameters shall be defined in the IRS. This data includes (but not limited to):

* Input voltages & currents per line (no need to record neutral line input)
* Output voltages & currents per line.
* Power consumption (VA)
* Date & time (synchronized to MIU time stamp)
* PSU Fans speed for each of the 3 fans
* Internal temperature of every internal card reading
* Logic statuses (incl. missing phase indications)
* Software/firmware version
* ETI
* Serial number
* ON/OFF signal occurrences
* Reset/power down indications.
* All indications are also sent periodically to the MIU via communication channel.

1. The log file shall be cyclic and capable of recording at a frequency of 1KHz with a minimum memory allocation of 10GBytes.

## PSU Logfile handling - general rules

* Logfile shall be recorded in the PSU at 1KHz rate.
* A Logfile is defined as a sequence of recorded entries (entry format detailed in par 6.2 below).
* Every time input minimum voltage is supplied to the PSU (see PSU PIDS paragraph 3.1.10 for minimum voltage conditions for log operation), a new log file will be started with the following name:
  + First the name of the file will be based on the starting date of 01/01/1970 and a sequential cyclic counter. E.g. **1970\_01\_01\_00\_00\_00\_PSU\_0000.log**
  + Once receiving the actual time from the MIU, the file name will be updated according to the actual date and time, using the following format: YYYY\_MM\_DD\_HH\_MM\_SS\_PSU\_XXXX.log (YYYY-Year, MM-Month, DD-Day, HH-Hour, MM-Minutes, SS-Seconds, XXXX – stays as before with the sequential counter).
  + the last 4 digits of the file name are a decimal number, but always 4 digits (for example 0000, 0001, etc.). Also, the counter is at the start of the name.
  + In case no communication with the MIU is established, then the file name shall remain with the date of 1970 and the sequential counter.
  + The suggested solution of saving in the eMMC the last number as a file. If eMMC card is replaced, then the counter is reset.
  + Another option to consider – instead of storing the counter in the eMMC, upon power up – look for the last logfile name counter and continue from there.
  + If the MIU commands the PSU to erase logfiles, then the counter should start from 0000 again.
* The log file will be recorded if the 12V redundant supply remains above 8V. When going below 8V, the log file will be closed, and recording stopped.
* Log files will be recorded if there is memory capacity for the recording. When the memory is completely full, the recording will stop.
* The logfile shall be downloaded using SFTP client.
* The log file shall be cyclic and capable of recording at a frequency of 1KHz with a memory allocation of 10GBytes.
* eMMC 64GB
* FPGA built-in controller (50MHz)
* When voltage is applied to the system, we assume that we start a new log file and do not erase any existing log files.

# PSU Status 64-bit Register

## Overall Description

The PSU status register provides consolidated bit-field information and is periodically reported to the user. The PSU status register depends on the overall data collection made by the SBC board.

## Detailed Description

The following is the bit-field list description with a complete data flow to be implemented:

|  |  |  |  |
| --- | --- | --- | --- |
| **Data bits** | **Field** | **Condition** | **Remarks** |
| 0 | DC\_IN\_Status | 22[V] < 28V\_IN\_sns < 30[V]. | DC Input Voltage Status - OK (0) or Fail (1) |
| 1 | AC\_IN\_Status | 108[V] < Vsns\_PH1 < 118[V].  108[V] < Vsns\_PH2 < 118[V].  108[V] < Vsns\_PH3 < 118[V]. | AC Input Voltage Status - OK (0) or Fail (1) |
| 2 | Power\_Out\_Status | 108[V] < Vsns\_PH\_A\_RLY < 118[V].  108[V] < Vsns\_PH\_B\_RLY < 118[V].  108[V] < Vsns\_PH\_C\_RLY < 118[V].  108[V] < OUT4\_sns < 118[V]. | Output power Status - OK (0) or Fail (1) |
| 3 | MIU\_COM\_Status | Ethernet Communication Status | MIU Communication Status - OK (0) or Fail (1) |
| 4 | OUT1\_OC | DCDC1 Message: Byte 8: bit 6 | Overcurrent Indication for Output1 |
| 5 | OUT2\_OC | DCDC2 Message: Byte 8: bit 6 | Overcurrent Indication for Output2 |
| 6 | OUT3\_OC | N/A | Overcurrent Indication for Output3 |
| 7 | OUT4\_OC | N/A | Overcurrent Indication for Output4 |
| 8 | OUT5\_OC | DCDC5 Message: Byte 8: bit 6 | Overcurrent Indication for Output5 |
| 9 | OUT6\_OC | DCDC6 Message: Byte 8: bit 6 | Overcurrent Indication for Output6 |
| 10 | OUT7\_OC | DCDC7 Message: Byte 8: bit 6 | Overcurrent Indication for Output7 |
| 11 | OUT8\_OC | DCDC8 Message: Byte 8: bit 6 | Overcurrent Indication for Output8 |
| 12 | OUT9\_OC | DCDC9 Message: Byte 8: bit 6 | Overcurrent Indication for Output9 |
| 13 | OUT10\_OC | DCDC10 Message: Byte 8: bit 6 | Overcurrent Indication for Output10 |
| 14 | DC\_IN\_OV | Vsns\_28V\_IN > 32[V] | Overvoltage indication for DC Input |
| 15 | AC\_IN\_ PH1­\_OV | Vsns\_PH1 > 125[V] | Overvoltage indication for Phase 1 Input |
| 16 | AC\_IN\_ PH2­\_OV | Vsns\_PH2 > 125[V] | Overvoltage indication for Phase 2 Input |
| 17 | AC\_IN\_ PH3­\_OV | Vsns\_PH3 > 125[V] | Overvoltage indication for Phase 3 Input |
| 18 | OUT1\_OV | DCDC1 Message: Byte 8: bit 7 | Overvoltage Indication for Output1 |
| 19 | OUT2\_OV | DCDC2 Message: Byte 8: bit 7 | Overvoltage Indication for Output2 |
| 20 | OUT3\_OV | Vsns\_PH\_A\_RLY > 125[V]  Vsns\_PH\_B\_RLY > 125[V]  Vsns\_PH\_C\_RLY > 125[V] | Overvoltage Indication for Output3 |
| 21 | OUT4\_OV | OUT4\_sns > 125[V] | Overvoltage Indication for Output4 |
| 22 | OUT5\_OV | DCDC5 Message: Byte 8: bit 7 | Overvoltage Indication for Output5 |
| 23 | OUT6\_OV | DCDC6 Message: Byte 8: bit 7 | Overvoltage Indication for Output6 |
| 24 | OUT7\_OV | DCDC7 Message: Byte 8: bit 7 | Overvoltage Indication for Output7 |
| 25 | OUT8\_OV | DCDC8 Message: Byte 8: bit 7 | Overvoltage Indication for Output8 |
| 26 | OUT9\_OV | DCDC9 Message: Byte 8: bit 7 | Overvoltage Indication for Output9 |
| 27 | OUT10\_OV | DCDC10 Message: Byte 8: bit 7 | Overvoltage Indication for Output10 |
| 28 | DC\_IN\_UV | 28V\_IN\_sns < 18[V] | Undervoltage indication for DC Input |
| 29 | AC\_IN\_ PH1­\_UV | Vsns\_PH1 < 95[V] | Undervoltage indication for Phase 1 Input |
| 30 | AC\_IN\_ PH2­\_UV | Vsns\_PH2 < 95[V] | Undervoltage indication for Phase 2 Input |
| 31 | AC\_IN\_ PH3­\_UV | Vsns\_PH3 < 95[V] | Undervoltage indication for Phase 3 Input |
| 32 | AC\_IN\_PH1\_Status | Vsns\_PH1 **No zero crossings detected** | Phase 1 OK (0) or missing indication (1) |
| 33 | AC\_IN\_PH2\_Status | Vsns\_PH2 **No zero crossings detected** | Phase 2 OK (0) or missing indication (1) |
| 34 | AC\_IN\_PH3\_Status | Vsns\_PH3 **No zero crossings detected** | Phase 3 OK (0) or missing indication (1) |
| 35 | AC\_IN\_Neutral\_Status | **TBD** | Neutral OK (0) or missing indication (1) |
| 36 | Is\_Logfile\_Running | Logfile status information bit | Is Logfile running? |
| 37 | Is\_Logfile\_Erase\_In \_Process | 0 = Logfile not in erase process 1 = Logfile in erase process | Is Logfile Erase In process? |
| 38 | Fan1\_Speed\_Status | Fan 1 Speed is in range of X +- Y [%] | Fan1 speed Status - OK (0) or Fail (1) |
| 39 | Fan2\_Speed\_Status | Fan 2 Speed is in range of X +- Y [%] | Fan2 speed Status - OK (0) or Fail (1) |
| 40 | Fan3\_Speed\_Status | Fan 3 Speed is in range of X +- Y [%] | Fan3 speed Status - OK (0) or Fail (1) |
| 41 | T1\_OVER\_TEMP\_Status | DCDC1 Message: Byte 8: bit 5 | PSU Temperature OK (0) Or Over Temperature (1) |
| 42 | T2\_OVER\_TEMP\_Status | DCDC2 Message: Byte 8: bit 5 | PSU Temperature OK (0) Or Over Temperature (1) |
| 43 | T3\_OVER\_TEMP\_Status | MAIN\_BOARD Message: Byte 7: bit 5. | PSU Temperature OK (0) Or Over Temperature (1) |
| 44 | T4\_OVER\_TEMP\_Status | DCDC10 Message: Byte 8: bit 5 | PSU Temperature OK (0) Or Over Temperature (1) |
| 45 | T5\_OVER\_TEMP\_Status | DCDC5 Message: Byte 8: bit 5 | PSU Temperature OK (0) Or Over Temperature (1) |
| 46 | T6\_OVER\_TEMP\_Status | DCDC6 Message: Byte 8: bit 5 | PSU Temperature OK (0) Or Over Temperature (1) |
| 47 | T7\_OVER\_TEMP\_Status | DCDC7 Message: Byte 8: bit 5 | PSU Temperature OK (0) Or Over Temperature (1) |
| 48 | T8\_OVER\_TEMP\_Status | DCDC8 Message: Byte 8: bit 5 | PSU Temperature OK (0) Or Over Temperature (1) |
| 49 | T9\_OVER\_TEMP\_Status | DCDC9 Message: Byte 8: bit 5 | PSU Temperature OK (0) Or Over Temperature (1) |
| 50 | CC\_TCU\_Inhibit | Logical status | CCTCU OFF (0) or ON (1) indication |
| 51 | EC\_TCU\_Inhibit | Logical status | ECTCU OFF (0) or ON (1) indication |
| 52 | Reset | 0 = Reset to MIU Not Initiated  1 = Reset to MIU Initiated | Reset to MIU Initiated indication. |
| 53 | Shutdown | 0 = Shutdown to MIU Not Initiated  1 = Shutdown to MIU Initiated | Shutdown to MIU Initiated indication. |
| 54 | Emergency\_Shutdown | 0 = Emergency Shutdown to MIU Not Initiated  1 = Emergency Shutdown to MIU Initiated | Emergency Shutdown to MIU Initiated indication. |
| 55 | System\_Off | 0 = Power Down Process Not Initiated  1 = Power Down Process Initiated | Power Down Process Initiated indication. |
| 56 | ON\_OFF\_Switch\_State | PowerON\_fpga line status | ON/OFF Switch State indication |
| 57 | Capacitor1\_end\_of\_life | TBD | Capacitor-1 end of life  indication - Optional TBD |
| 58 | Capacitor2\_end\_of\_life | TBD | Capacitor-2 end of life  indication - Optional TBD |
| 59 | Capacitor3\_end\_of\_life | TBD | Capacitor-3 end of life  indication - Optional TBD |
| 60 | Capacitor4\_end\_of\_life | TBD | Capacitor-4 end of life  indication - Optional TBD |
| 61 | Capacitor5\_end\_of\_life | TBD | Capacitor-5 end of life  indication - Optional TBD |
| 62 | Capacitor6\_end\_of\_life | TBD | Capacitor-6 end of life  indication - Optional TBD |
| 63 | Capacitor7\_end\_of\_life | TBD | Capacitor-7 end of life  indication - Optional TBD |

**Appendix A: Zero Crossing Detection**

All the RMS voltages and currents are calculated as a rolling average of exactly three 400Hz cycles and the 400Hz line frequency can vary at least up to 10%. The rms calculations need to use the correct number of samples to get close to three complete line cycles even if the line frequency varies from 400Hz. Note that the calculation of line frequency (based on Phase A voltage) can be used for all rms measurements in the system (even for phases B and C).

When the polarity changes, ignore for 10 samples (1msec) to provide a debounce.