Condor FPGA PS-PL

requirements

# Scope

This document details the requirements for the Condor FPGA SW and FW.

It is based on the requirements detailed in Elbit’s 5495-6000-00PS\_PIDS-PSU\_VER18 (1).docx spec.

Document and breaks down the into FPGA tasks.

Out of the Elbit spec we detail here only the tasks for the SW and FW.

# PS and PL pin list

This section lists each of the operational pins on the FPGA (both PS and PL) and adds a short description of the purpose of the pin. It also links to the section that explains what the algorithm needs to be done for the pin.

# Tasks

This chapter lists all tasks done by the FPGA. We may reference the Elbit spec for applicable tasks.

# Power On – Off

The following is the sequence requirements:

# Lamp indication

The PSU shall provide a discrete indication to a lamp on the control panel in the cockpit. The lamp indication shall be determined by the following inputs:

(1) 28VDC input to PSU is OK/NOK.

(2) 115VAC input to PSU is OK/NOK.

(3) MIU communication established.

Refer to [3.1.6 lamp indication](spec%20v18.pdf)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pin name | Location | PS/PL | Dir I/O/IO | FW/SW control | what to do? |
| POD\_STATUS\_FPGA | G17 | PL | out | FW | after all is good set this on. |
| lamp\_status\_fpga | J15 | PL | in |  | in indication of problem with lamps |

# Uarts

There are 9 modules of UART communicate with the external PS.

Refer to document [Protocol\_RS485\_DC-DC\_Control\_SEC.docx](file:///D:\EDM_Workspace\Elop_CondorMS_C10A\Elop_CondorMS_C10A_SBC_Board\R01A\Doc\Protocol_RS485_DC-DC_Control_SEC.docx)

Each module use:

EN - Data-enable PS output pin XX active high

PG - feedback from PS input pin XX active high = Power good

Tx - output pin XX data out to PS

Rx - input pin XX data from PS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Pin name | Location | PS/PL | Dir I/O/IO | FW/SW control | what to do? |
| RS485\_1 | EN\_PSU\_1\_FB | Y19 | PL | out | FW | Enable power supply 1, part of power on sequence (POWERON\_FPGA) |
| PG\_PSU\_1\_FB | Y18 | PL | in | FW | FB from power supply 1, part of power on sequence (POWERON\_FPGA) |
| RS485\_DE\_1 | T10 | PL | out | SW | Active-high driver data out enable, SW will ask for info from microcontroller and get answers to log them |
| RS485\_TXD\_1 | T11 | PL | out | SW | Mosi, SW will ask for info from microcontroller and get answers to log them |
| RS485\_RXD\_1 | R19 | PL | in | SW | Miso, SW will ask for info from microcontroller and get answers to log them |
| RS485\_2 | EN\_PSU\_2\_FB | W16 | PL | out | FW | Enable power supply 2, part of power on sequence (POWERON\_FPGA) |
| PG\_PSU\_2\_FB | V16 | PL | in | FW | FB from power supply 2, part of power on sequence (POWERON\_FPGA) |
| RS485\_DE\_2 | U13 | PL | out | SW | Active-high driver data out enable, SW will ask for info from microcontroller and get answers to log them |
| RS485\_TXD\_2 | U12 | PL | out | SW | Mosi, SW will ask for info from microcontroller and get answers to log them |
| RS485\_RXD\_2 | T12 | PL | in | SW | Miso, SW will ask for info from microcontroller and get answers to log them |
| RS485\_3 |  |  |  |  |  | uart 3 operates the buck without external enable |
| PG\_BUCK\_FB | T20 | PL | in | FW | power good of BUCK supply , part of power on sequence (POWERON\_FPGA) |
| RS485\_DE\_3 | W13 | PL | out | SW | Active-high driver data out enable, SW will ask for info from microcontroller and get answers to log them |
| RS485\_TXD\_3 | V12 | PL | out | SW | Mosi, SW will ask for info from microcontroller and get answers to log them |
| RS485\_RXD\_3 | V13 | PL | in | SW | Miso, SW will ask for info from microcontroller and get answers to log them |
| RS485\_4 | EN\_PSU\_10\_FB | P16 | PL | out | FW | Enable power supply 10, part of power on sequence (POWERON\_FPGA) |
| PG\_PSU\_10\_FB | P15 | PL | in | FW | FB from power supply 10, part of power on sequence (POWERON\_FPGA) |
| RS485\_DE\_4 | P14 | PL | out | SW | Active-high driver data out enable, SW will ask for info from microcontroller and get answers to log them |
| RS485\_TXD\_4 | T15 | PL | out | SW | Mosi, SW will ask for info from microcontroller and get answers to log them |
| RS485\_RXD\_4 | T14 | PL | in | SW | Miso, SW will ask for info from microcontroller and get answers to log them |
| RS485\_5 | EN\_PSU\_5\_FB | R17 | PL | out | FW | Enable power supply 5, part of power on sequence (POWERON\_FPGA) |
| PG\_PSU\_5\_FB | R16 | PL | in | FW | FB from power supply 5, part of power on sequence (POWERON\_FPGA) |
| RS485\_DE\_5 | Y17 | PL | out | SW | Active-high driver data out enable, SW will ask for info from microcontroller and get answers to log them |
| RS485\_TXD\_5 | Y16 | PL | out | SW | Mosi, SW will ask for info from microcontroller and get answers to log them |
| RS485\_RXD\_5 | R14 | PL | in | SW | Miso, SW will ask for info from microcontroller and get answers to log them |
| RS485\_6 | EN\_PSU\_6\_FB | R18 | PL | out | FW | Enable power supply 6, part of power on sequence (POWERON\_FPGA) |
| PG\_PSU\_6\_FB | T17 | PL | in | FW | FB from power supply 6, part of power on sequence (POWERON\_FPGA) |
| RS485\_DE\_6 | T16 | PL | out | SW | Active-high driver data out enable, SW will ask for info from microcontroller and get answers to log them |
| RS485\_TXD\_6 | Y14 | PL | out | SW | Mosi, SW will ask for info from microcontroller and get answers to log them |
| RS485\_RXD\_6 | W14 | PL | in | SW | Miso, SW will ask for info from microcontroller and get answers to log them |
| RS485\_7 | EN\_PSU\_7\_FB | V18 | PL | out | FW | Enable power supply 7, part of power on sequence (POWERON\_FPGA) |
| PG\_PSU\_7\_FB | V17 | PL | in | FW | FB from power supply 7, part of power on sequence (POWERON\_FPGA) |
| RS485\_DE\_7 | W15 | PL | out | SW | Active-high driver data out enable, SW will ask for info from microcontroller and get answers to log them |
| RS485\_TXD\_7 | V15 | PL | out | SW | Mosi, SW will ask for info from microcontroller and get answers to log them |
| RS485\_RXD\_7 | U17 | PL | in | SW | Miso, SW will ask for info from microcontroller and get answers to log them |
| RS485\_8 | EN\_PSU\_8\_FB | W19 | PL | out | FW | Enable power supply 8, part of power on sequence (POWERON\_FPGA) |
| PG\_PSU\_8\_FB | W18 | PL | in | FW | FB from power supply 8, part of power on sequence (POWERON\_FPGA) |
| RS485\_DE\_8 | U18 | PL | out | SW | Active-high driver data out enable, SW will ask for info from microcontroller and get answers to log them |
| RS485\_TXD\_8 | U15 | PL | out | SW | Mosi, SW will ask for info from microcontroller and get answers to log them |
| RS485\_RXD\_8 | U14 | PL | in | SW | Miso, SW will ask for info from microcontroller and get answers to log them |
| RS485\_9 | EN\_PSU\_9\_FB | P18 | PL | out | FW | Enable power supply 9, part of power on sequence (POWERON\_FPGA) |
| PG\_PSU\_9\_FB | N17 | PL | in | FW | FB from power supply 9, part of power on sequence (POWERON\_FPGA) |
| RS485\_DE\_9 | P19 | PL | out | SW | Active-high driver data out enable, SW will ask for info from microcontroller and get answers to log them |
| RS485\_TXD\_9 | N18 | PL | out | SW | Mosi, SW will ask for info from microcontroller and get answers to log them |
| RS485\_RXD\_9 | U19 | PL | in | SW | Miso, SW will ask for info from microcontroller and get answers to log them |

# SPIs

Hv spi convert 8 channels using spi protocol according to [ADS7951SBRGET](https://www.ti.com/lit/ds/symlink/ads7950.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1704647963193&ref_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%252Fsuppproductinfo.tsp%253FdistId%253D10%2526gotoUrl%253Dhttps%253A%252F%252Fwww.ti.com%252Flit%252Fgpn%252Fads7950).

SPI lines:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pin name | Location | PS/PL | Dir I/O/IO | FW/SW control | what to do? |
| HV\_ADC\_CS\_FPGA | M15 | PL | out | SW | spi for current sense of in supply, SW will do some calculation based on this in |
| HV\_ADC\_SCLK\_FPGA | L14 | PL | out | SW | spi for current sense of in supply, SW will do some calculation based on this in |
| HV\_ADC\_SDI\_FPGA | L15 | PL | out | SW | out of A/D (Mosi), spi for current sense of in supply, SW will do some calculation based on this in |
| HV\_ADC\_SDO\_FPGA | M14 | PL | in | SW | input of A/D (Miso), spi for current sense of in supply, SW will do some calculation based on this in |

Channel address:

|  |  |  |  |
| --- | --- | --- | --- |
| Channel |  | Dir I/O/IO |  |
| Channel 0 | OUT4\_sns | in | AC voltage from the 1-phase relay |
| Channel 1 | Vsns\_PH1 | in | input voltage sense |
| Channel 2 | Vsns\_PH2 | in | input voltage sense |
| Channel 3 | Vsns\_PH3 | in | input voltage sense |
| Channel 4 | OUT4\_Isns | in | AC current from the 1-phase relay |
| Channel 5 | Vsns\_PH\_C\_RLY | in | AC voltage sense of the 3-phase relay |
| Channel 6 | Vsns\_PH\_B\_RLY | in | AC voltage sense of the 3-phase relay |
| Channel 7 | Vsns\_PH\_A\_RLY | in | AC voltage sense of the 3-phase relay |

4 channels using spi protocol according to [ADS7951SBRGET](https://www.ti.com/lit/ds/symlink/ads7950.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1704647963193&ref_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%252Fsuppproductinfo.tsp%253FdistId%253D10%2526gotoUrl%253Dhttps%253A%252F%252Fwww.ti.com%252Flit%252Fgpn%252Fads7950).

SPI lines:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pin name | Location | PS/PL | Dir I/O/IO | FW/SW control | what to do? |
| I\_sns\_ADC\_CS\_fpga | V7 | PL | out | SW | SPI for current sense, SW will do some calculation based on this in |
| I\_sns\_ADC\_SCLK\_fpga | T9 | PL | out | SW | SPI for current sense, SW will do some calculation based on this in |
| I\_sns\_ADC\_SDI\_fpga | U10 | PL | out | SW | SPI for current sense, SW will do some calculation based on this in |
| I\_sns\_ADC\_SDO\_fpga | Y7 | PL | in | SW | SPI for current sense, SW will do some calculation based on this in |

Channel address:

|  |  |  |  |
| --- | --- | --- | --- |
| Channel |  | Dir I/O/IO |  |
| Channel 0 | DC\_PWR\_I\_sns | in |  |
| Channel 1 | PH1\_I\_sns | in | input current sense |
| Channel 2 | PH2\_I\_sns | in | input current sense |
| Channel 3 | PH3\_I\_sns | in | input current sense |

4 channels using spi protocol according to [ADS7951SBRGET](https://www.ti.com/lit/ds/symlink/ads7950.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1704647963193&ref_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%252Fsuppproductinfo.tsp%253FdistId%253D10%2526gotoUrl%253Dhttps%253A%252F%252Fwww.ti.com%252Flit%252Fgpn%252Fads7950).

SPI lines:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pin name | Location | PS/PL | Dir I/O/IO | FW/SW control | what to do? |
| ZCR\_sns\_ADC\_CS\_fpga | Y9 | PL | out | SW | SPI for reference current sense, SW will do some calculation based on this in |
| ZCR\_sns\_ADC\_SCLK\_fpga | Y8 | PL | out | SW | SPI for reference current sense, SW will do some calculation based on this in |
| ZCR\_sns\_ADC\_SDI\_fpga | V8 | PL | out | SW | SPI for reference current sense, SW will do some calculation based on this in |
| ZCR\_sns\_ADC\_SDO\_fpga | W8 | PL | in | SW | SPI for reference current sense, SW will do some calculation based on this in |

Channel address:

|  |  |  |  |
| --- | --- | --- | --- |
| Channel |  | Dir I/O/IO |  |
| Channel 0 | DC\_PWR\_ZCR\_sns | in |  |
| Channel 1 | PH1\_ZCR\_sns | in | input current sense |
| Channel 2 | PH2\_ZCR\_sns | in | input current sense |
| Channel 3 | PH3\_ZCR\_sns | in | input current sense |

# QSPI

QSPI pins for boot memory driving S25FL128SAGBHM200

|  |  |  |  |
| --- | --- | --- | --- |
| Pin name | Location | PS/PL | what to do? |
| QSPI\_1\_CS# | A7 | PS | program flash |
| QSPI\_DQ0 | B8 | PS | program flash |
| QSPI\_DQ1 | D6 | PS | program flash |
| QSPI\_DQ2 | B7 | PS | program flash |
| QSPI\_DQ3 | A6 | PS | program flash |
| QSPI\_1\_CLK | A5 | PS | program flash |

# Input on command

Active high Operate PFC enable pin xx

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pin name | Location | PS/PL | Dir I/O/IO | FW/SW control | what to do? |
| POWERON\_FPGA | V5 | PL | in | FW | in from system to FPGA to turn on the power supplies |

# Fan

There are 3 of them:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pin name | Location | PS/PL | Dir I/O/IO | FW/SW control | what to do? |
| FAN\_HALL1\_fpga | L16 | PL | in | FW | in square wave, measure the frequency. Change the FAN\_CTRL to bring to correct speed. |
| FAN\_PG1\_fpga | L17 | PL | in | SW | power good. Log |
| FAN\_CTRL1\_fpga | K17 | PL | out | FW | in square wave, measure the frequency. Change the FAN\_CTRL to bring to correct speed. |
| FAN\_EN1\_fpga | K18 | PL | out | FW | in square wave, measure the frequency. Change the FAN\_CTRL to bring to correct speed. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pin name | Location | PS/PL | Dir I/O/IO | FW/SW control | what to do? |
| FAN\_HALL2\_fpga | D19 | PL | in | FW | in square wave, measure the frequency. Change the FAN\_CTRL to bring to correct speed. |
| FAN\_PG2\_fpga | D20 | PL | in | FW | power good. Log |
| FAN\_CTRL2\_fpga | E18 | PL | out | FW | in square wave, measure the frequency. Change the FAN\_CTRL to bring to correct speed. |
| FAN\_EN2\_fpga | E19 | PL | out | FW | in square wave, measure the frequency. Change the FAN\_CTRL to bring to correct speed. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pin name | Location | PS/PL | Dir I/O/IO | FW/SW control | what to do? |
| FAN\_HALL3\_fpga | B19 | PL | in | FW | in square wave, measure the frequency. Change the FAN\_CTRL to bring to correct speed. |
| FAN\_PG3\_fpga | A20 | PL | in | FW | power good. Log |
| FAN\_CTRL3\_fpga | E17 | PL | out | FW | in square wave, measure the frequency. Change the FAN\_CTRL to bring to correct speed. |
| FAN\_EN3\_fpga | D18 | PL | out | FW | in square wave, measure the frequency. Change the FAN\_CTRL to bring to correct speed. |

Hall is square wave   
enable active high output.

Ctrl fan speed control output square wave start at fix frequency of (500Hz) - tbd.

With 50% PWM - the PWM is increase decrease at a rate of 2sec according to the hall input.

# Ethernet communication

|  |  |  |
| --- | --- | --- |
| Pin name | Location | PS/PL |
| ETH\_TX\_CLK | A19 | PS |
| ETH\_TXD0 | E14 | PS |
| ETH\_TXD1 | B18 | PS |
| ETH\_TXD2 | D10 | PS |
| ETH\_TXD3 | A17 | PS |
| ETH\_TX\_CTRL | F14 | PS |
| ETH\_RX\_CLK | B17 | PS |
| ETH\_RXD0 | D11 | PS |
| ETH\_RXD1 | A16 | PS |
| ETH\_RXD2 | F15 | PS |
| ETH\_RXD3 | A15 | PS |
| ETH\_RX\_CTRL | D13 | PS |
| ETH\_MDC | C10 | PS |
| ETH\_MDIO | C11 | PS |

# Elapsed time FLASH memory communication

Each of the PSU shall provide a digital Elapsed Time Indication with resolution of hours. This information shall be provided along with the serial number indication. The indication shall be accessible by MIU via communication and recorded in the PSU internal log file.

The time indication can only be reset separately as part of the PSU manufacturing process (can be in final stages, during check-out). The time reset shall not be accessible when the PSU box is closed, only when it is opened and connected via specific port to test equipment at developer facility.

The ETI shall be accessible / data shall be restored in case of most types of PSU malfunctions.

Refer to [3.8.4 elapsed time indication (ETI)](spec%20v18.pdf)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pin name | Location | PS/PL | Dir I/O/IO | what to do? |
| SPI\_CLK | C16 | PS | out | elapsed time FLASH MIO28 |
| SPI\_MISO | C13 | PS | in | elapsed time FLASH MIO29 |
| SPI\_CSN | C15 | PS | out | elapsed time FLASH MIO30 |
| SPI\_MOSI | D15 | PS | out | elapsed time FLASH MIO33 |

# Software/firmware upload

In case of the operational software/firmware is damaged (i.e. – due to unplanned shutdown. power loss, etc…) and the PSU can’t power up, the PSU shall have a basic backup software/firmware which shall allow minimal communication to re-upload the updated software/firmware versions.

Uploading updated software/firmware to the PSU shall be possible via PSU connectors without the need of removal of the PSU cover. This can be performed via dedicated pins or connectors.

# Log File functionality

Refer to [3.1.4 Log File Functionality](spec%20v18.pdf).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pin name | Location | PS/PL | Dir I/O/IO | what to do? |
| SD\_CLK | D9 | PS | out | eMMC |
| SD\_CMD | C6 | PS | out | eMMC |
| SD\_D0 | E9 | PS | inout | eMMC |
| SD\_D1 | E8 | PS | inout | eMMC |
| SD\_D2 | C5 | PS | inout | eMMC |
| SD\_D3 | C8 | PS | inout | eMMC |

Log file will be manage on EMMC memory using

# Debug UART

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pin name | Location | PS/PL | Dir I/O/IO | FW/SW control | what to do? |
| UART\_RXD\_PL | N15 | PL | in | FW | FPGA debug UART(MISO) |
| UART\_TXD\_PL | N16 | PL | out | FW | FPGA debug UART(MOSI) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pin name | Location | PS/PL | Dir I/O/IO | FW/SW control | what to do? |
| UART\_RXD | M19 | PS | in | FW | debug UART for PS(MISO) |
| UART\_TXD | M20 | PS | out | FW | debug UART for PS(MOSI) |

# Relay

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pin name | Location | PS/PL | Dir I/O/IO | FW/SW control | what to do? |
| Relay\_3ph\_fpga | K16 | PL | out | FW | RELAY 3PH OPERATE, part of power on sequence (POWERON\_FPGA) |
| Relay\_1ph\_fpga | J16 | PL | out | FW | RELAY 1PH OPERATE, part of power on sequence (POWERON\_FPGA) |

Condor Internal Software Protocol

**Inputs and Outputs**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **INPUT** | **output**  **#** | **Output**  **Voltage** | **OUTPUT POWER** | | | **Hold-Up Time (ms)** | **Target ELOP System** |
| Typical Continuous | Maximum Continuous | Transient/Peak |
| **28VDC** | 1 | 36VDC | 150W | 220W | 250W (<5s) | 0 | FAN CCA |
| **115VAC**  **3Ø 400Hz** | 2 | 30.5VDC | 20W | 50W | 150W (<50ms) | 10 | MWIR  COOLER |
| 3 | 115VAC  3Ø | 850VA | 1000VA | 1.5kVA | -- | ADLS |
| 4 | 115VAC 1Ø | 12W | 15W | NA | -- | EDU |
| 5 | 28VDC | 80W | 100W | 120W (<50ms) | 50 | VCC |
| 6 | 28VDC | 80W | 100W | 120W | 10 | SMCC |
| 7 | 28VDC | 100W | 150W | 300W (<100ms) | 1ms @12A, then 20ms @1A | MCC |
| 8 | 28VDC | 900W | 1200W | 1300W | 10 | MIU |
| 9 | 28VDC | 100W | 200W | 500W (<100ms) | 0 | Roll/FMC  Motors |
| 10 | 28VDC | 200W | 300W | 350W (<100ms) | 10 | INS/EDU/  SPARE |

**System components and controllers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | PCB | Output | Description | Controllers | communication with the FPGA |
| SBC | SBC | N/A | System controller | FPGA |  |
| PFC | Main | Internal | Converts 3-phase 115VAC to 440VDC bus providing aux power to system and power to HV Buck | PFC micro |  |
| HV\_Buck | Main | Internal | Converts 440VDC to 85VDC bus providing power to several DCDCs | Buck micro |  |
| 3-phase relay | Filter | 3 | 3-phase 115VAC pass-through | N/A | enable lines |
| 1-phase relay | Filter | 4 | 1-phase 115VAC pass-through | N/A | enable lines |
| 50W DCDC | 50W DCDC | 2 | Converts 85VDC to 30.5V Output 2 | Primary-side secondary-side | RS485 |
| DCDC1 | DCDC | 1 | Converts 28VDC to 36VDC | Primary-side secondary-side | RS485 |
| DCDC4 | DCDC | 4 | Converts 85VDC to 28VDC | Primary-side secondary-side | RS485 |
| DCDC5 | DCDC | 5 | Converts 85VDC to 28VDC | Primary-side secondary-side | RS485 |
| DCDC6 | DCDC | 6 | Converts 85VDC to 28VDC | Primary-side secondary-side | RS485 |
| DCDC7 | DCDC | 7 | Converts 85VDC to 28VDC | Primary-side secondary-side | RS485 |
| LLC | 1k2W LLC | 8 | Converts 85VDC to 28VDC | Primary-side secondary-side | RS485 |
| DCDC9 | DCDC | 9 | Converts 85VDC to 28VDC | Primary-side secondary-side | RS485 |
| DCDC10 | DCDC | 10 | Converts 85VDC to 28VDC | Primary-side secondary-side | RS485 |

1. All secondary-side micro controllers communicate with the primary-side micro controllers through UART as well as dedicated logic lines.
2. The buck micro communicates with the PFC micro through UART as well as dedicated logic lines.
3. The FPGA has dedicated EN (enable) lines to PFC, 50WDCDC, DCDC1, DCDC4, DCDC5, DCDC6, DCDC7, LLC, DCDC9, DCDC10 and dedicated PG (Power Good) lines from HV\_Buck, 50WDCDC, DCDC1, DCDC4, DCDC5, DCDC6, DCDC7, LLC, DCDC9, DCDC10.

**FPGA Measurements**

|  |  |  |
| --- | --- | --- |
| Measurement | Source | Details |
| VDC\_IN | DCDC1 RS485 | DCDC1 primary micro measures VDC\_IN and reports to DCDC1 secondary micro. Secondary micro reports VDC\_IN to FPGA through RS485. |
| VAC\_IN\_PH\_A | A/D on SBC | SBC A/D U48 CH1 represents instantaneous VAC\_IN\_PH\_A. Must calculate RMS. |
| VAC\_IN\_PH\_B | A/D on SBC | SBC A/D U48 CH2 represents instantaneous VAC\_IN\_PH\_B. Must calculate RMS. |
| VAC\_IN\_PH\_C | A/D on SBC | SBC A/D U48 CH3 represents instantaneous VAC\_IN\_PH\_C. Must calculate RMS. |
| I\_DC\_IN | A/D on SBC | SBC A/D U54 CH0 represents I\_DC\_IN |
| I\_AC\_IN\_PH\_A | A/D on SBC | SBC A/D U54 CH1 represents instantaneous phase A input current. Must calculate RMS. |
| I\_AC\_IN\_PH\_B | A/D on SBC | SBC A/D U54 CH2 represents instantaneous phase B input current. Must calculate RMS. |
| I\_AC\_IN\_PH\_C | A/D on SBC | SBC A/D U54 CH3 represents instantaneous phase C input current. Must calculate RMS. |
| V\_OUT\_2 | 50WDCDC RS485 | 50WDCDC secondary micro measures V\_OUT\_2 and reports to FPGA through RS485. |
| V\_OUT\_3\_ph1 | A/D on SBC | SBC A/D U48 CH7 represents instantaneous V\_OUT\_3\_ph1. Must calculate RMS. |
| V\_OUT\_3\_ph2 | A/D on SBC | SBC A/D U48 CH6 represents instantaneous V\_OUT\_3\_ph2. Must calculate RMS. |
| V\_OUT\_3\_ph3 | A/D on SBC | SBC A/D U48 CH5 represents instantaneous V\_OUT\_3\_ph3. Must calculate RMS. |
| V\_OUT\_4 | A/D on SBC | SBC A/D U48 CH0 represents instantaneous V\_OUT\_4. Must calculate RMS. |
| V\_OUT\_x  x = 1,5-7,9,10 | DCDCx RS485 | DCDCx secondary micro measures V\_OUT\_x and reports to FPGA through RS485. |
| V\_OUT\_8 | LLC RS485 | LLC secondary micro measures output voltage and reports to FPGA through RS485. |
| I\_OUT\_2 | 50WDCDC RS485 | 50WDCDC secondary micro measures I\_OUT\_2 and reports to FPGA through RS485. |
| I\_OUT\_3\_ph1 | Calculation | See note 1. Subtract instantaneous current I\_OUT\_4 and I\_PFC\_PH\_A from I\_AC\_IN\_PH\_A and calculate rms value. |
| I\_OUT\_3\_ph2 | Calculation | See note 1. Subtract instantaneous current I\_PFC\_PH\_B from I\_AC\_IN\_PH\_B and calculate rms value. |
| I\_OUT\_3\_ph3 | Calculation | See note 1. Subtract instantaneous current I\_PFC\_PH\_C from I\_AC\_IN\_PH\_C and calculate rms value. |
| I\_OUT\_4 | A/D on SBC | SBC A/D U48 CH4 represents instantaneous current through Output 4. Must calculate RMS. |
| I\_OUT\_x  x = 1,5-7,9,10 | DCDCx RS485 | DCDCx secondary micro measures I\_OUT\_x and reports to FPGA through RS485. |
| I\_OUT\_8 | LLC RS485 | LLC secondary micro measures I\_OUT\_x and reports to FPGA through RS485. |
| AC\_Power | Calculation | Running average of VAC\_IN\_PH\_A\* I\_AC\_IN\_PH\_A + VAC\_IN\_PH\_B\* I\_AC\_IN\_PH\_B + VAC\_IN\_PH\_C\* I\_AC\_IN\_PH\_C |
| Fan\_Speed\_1 | Hall sensor from Fan1 | Frequency of hall sensor3 same as fan speed. Change Hz to RPM. |
| Fan\_Speed\_2 | Hall sensor from Fan2 | Frequency of hall sensor2 same as fan2 speed. Change Hz to RPM. |
| Fan\_Speed\_3 | Hall sensor from Fan3 | Frequency of hall sensor3 same as fan3 speed. Change Hz to RPM. |
| T\_x  x = 1,5-7,9,10 | DCDCx RS485 | Temperature of DCDCx reported from DCDCx secondary through RS485. See note 2. |
| T\_2 | 50WDCDC RS485 | Temperature of 50WDCDC reported from 50WDCDC secondary through RS485. See note 2. |
| T\_8 | LLC RS485 | The temperature of LLC reported from LLC secondary through RS485. See note 2 |
| T\_main | HV\_BUCK RS485 | Temperature of Main Board reported from Buck through RS485. See note 2 |
| DC\_IN\_Status | Calculation | If 28VDC from Aircraft is within correct range, then logical 1; otherwise, logical 0. See note 3. |
| AC\_IN\_Status | Calculation | If RMS values of VAC\_IN\_PH\_A, B, and C are in correct range, then logical 1; otherwise, logical 0. See note 4. |
| Power\_Out\_Status | Calculation | If all modules produce PG=1, then 1, otherwise 0. |
| OUTx\_OC  x = 1,5-7,9,10 | DCDCx RS485 | If PG goes low on any power supply unit, then the secondary of that unit must send status information to the FPGA via RS485 indicating if the shutdown occurred due to output overcurrent, output overvoltage, overtemperature, input voltage out-of-range, or another reason. |
| OUT2\_OC | DCDC2 RS485 |
| OUT8\_OC | LLC RS485 |
| DC\_IN\_OV | DCDC1 RS485 |
| OUTx OV  x = 1,5-7,9,10 | DCDCx RS485 |
| OUT2 OV | DCDC2 RS485 |
| OUT8 OV | LLC RS485 |
| DC\_IN\_UV | DCDC1 RS485 |
| AC\_IN\_UV | Calculation | See note 4. |
| PHx\_Status  x = 1, 2, 3 | Calculation | See note 4. Report according to voltage on each phase. |
| Neutral\_Status | Calculation | Normally a logical 1. Average of instantaneous sum of VAC\_IN\_PH\_A, VAC\_IN\_PH\_B, and VAC\_IN\_PH\_C should always be close to 0. If departing more than some threshold amount from 0, that indicates that the neutral fuse opened in which case the logical status will change to 0. |
| OVER\_TEMP\_Status | RS485 all PSUs | Logical 1 unless a PSU reports an overtemperature through the RS485 bus. |
| Capacitor\_end\_of\_life | Buck RS485 | See note 5. |
|  |  |  |

Note 1: Buck secondary reports rms PH1, PH2, and PH3 currents into PFC. FPGA estimates instantaneous current into PFC by assuming that the current is proportional to VAC\_IN\_PH\_A, VAC\_IN\_PH\_B, and VAC\_IN\_PH\_C but has the rms value reported from the buck secondary on the RS485. The FPGA can use this information to calculate instantaneous values of I\_PFC\_PH\_A, I\_PFC\_PH\_B, and I\_PFC\_PH\_C.

Note 2: The secondary of each micro can be configured to report temperature either as a 12-bit A/D value or each micro can calculate temperature according to NTC curve and report as an 8-bit integer representing temperature with a known offset.

Note 3: The 28VDC from the aircraft must remain within 18-32VDC for continuous operation; however, the input voltage is allowed to go outside these limits for short duration. Spec 3.2.10.3 Below 18V there is an immediate shutdown. The PSU must work at least according to MIL-STD 704A transients. Above 90V, DCDC1 will immediately shutdown without any warning time. For 80V – 90V, an emergency shutdown will be initiated. For 32V – 80V, the PSU can work for 5s and will then shut down. We must decide how much of this functionality is determined by the FPGA and how much by DCDC1.

Note 4: Spec 3.2.1.2 and 3.2.10.3 Between 95VAC and 125VAC the input AC voltage is OK. Between 90VAC and 95VAC, the input voltage is OK for 500ms and then a shutdown must be initiated. Below 90VAC, there is an immediate shutdown. Above 125VAC, the PSU only needs to work according to MIL-STD 704A but may work longer. MIL-STD 704A allows transients up to 180VAC for 150ms decreasing to 125VAC over the next 3s. If the input voltage goes above 180VAC, the PSU shuts down immediately. We must decide how much of these functions are done within the FPGA and how much is done by the PFC micro.

Note 5: Based on the agreement during PDR, this requirement was removed. However, during development we can attempt to make this calculation and decide later whether it is accurate enough to use in production. HV\_BUCK bus capacitance value can be calculated when the PSU is shut down. Based on Buck load current and rate of fall of the bus, it is possible to calculate the bus capacitor value. When the capacitor value drops below a predetermined value, it can be assumed to be at end-of-life. This would need to be calculated and reported to the FPGA during the shutdown, and the FPGA needs to store this information. Once the PSU is turned back on, the capacitor end-of-life indication can be set high. Note that to reset the end-of-life indication would require cycling the PSU.

## Built-in Tests (BIT)

After startup, the FPGA must request a built-in test (BIT) from each PSU module at an interval of 100Hz. Each PSU module then responds with the information it is required to provide the FPGA to satisfy the customer specification requirements for a BIT. See IRS document section 5 for details.  
  
If EN for the PSU module is low at the time the BIT is received, then the BIT is assumed to be a PUBIT (power up built-in test); otherwise, the BIT is assumed to be a periodic BIT. In the case of a PUBIT, the module will not push power, but only provide whatever information is possible given that the PSU is not powered up. The Main Board (PFC and HV\_BUCK) should be given an EN signal prior to the PUBIT since the main board can and should be powered completely during the PUBIT.  
Each PSU Module should provide a status byte with the BIT output. The status byte should include bits representing status of parameters appropriate for each PSU module, for example, output OVP, output OCP, output UV, internal OCP limits, internal OTP, internal DC bus OVP, internal DC bus UV, etc. The format of the status byte is TBD.

Writing to Flash  
The FPGA has an A/D reading of the 12V\_Redundant supply that creates all of the aux power for the SBC Board. When this supply drops below 8V for more than 1ms (level and duration may be modified according to testing), then the FPGA should immediately stop writing to the eMMC to prevent the possibility of trying to write to the eMMC while the power is going down. This test should be done frequently enough that there is no possibility for the SBC power to go down in the middle of a write operation to the eMMC (e.g. every 10ms).  
Note that when the JTAG is programming the eMMC, the 12V\_Redundant supply may be as low as 5V.

Startup Sequence

The PFC is connected to the 115VAC regardless of whether the PSU has been started or not. When the 115VAC is connected, the PFC micro will turn on the inrush protection FET based on bus capacitor charge and time, regardless of whether the PSU is running.

When a startup signal is received by the FPGA, the FPGA should turn on the fans and set EN high to the PFC micro. When PG is received from the Buck micro, the FPGA should set EN high to all of the other modules, to the three-phase relay (Output 3), and to the single-phase relay (Output 4).

Shutdown Sequences  
E-Shutdown

In the case of an emergency shutdown, after the E-shutdown discrete signal is set high, the FPGA must shut down PSU modules (setting EN low) according to required holdup times indicated in Table I. DCDC9 as well as the three-phase and single-phase relays shut down immediately. The 50WDCDC, the LLC, DCDC6, and DCDC10 shut down after 10ms. DCDC7 shuts down after 20ms. DCDC5 shuts down after 50ms. The fans should be turned off after a cool-down period TBD (Or we can decide to turn off the fans based on temperatures measured inside the PSU).  
  
Standard shutdown

In case of an overtemperature condition, the FPGA should set the discrete shutdown high and then disable all power supplies after 10s (3.1.5.2, 3.2.10.5). CCTCU and ECTCU should be set low after disabling the internal power supplies. The fans should be turned off after a cool-down period TBD (Or we can decide to turn off the fans based on temperatures measured inside the PSU).

Reverse Voltage

A reverse voltage condition will not occur during normal operation – it would only occur when power is first applied from the aircraft. In case of a reverse voltage, the 28V from the aircraft will appear to be 0V. The fans will not run and DCDC1 will not be powered. Since DCDC1 will not be powered, it will have no communication with the FPGA. If DCDC1 has no communication with the FPGA and the fan speed is 0, then the FPGA should assume that the 28V from the aircraft is 0V (whether from not being present or a reverse voltage) and the PSU should not startup. The 28V from the aircraft will be reported as 0V.

## Input OVP

If the input AC voltage has an OVP condition (transient above 180VAC, 3.2.10.2.1), the PFC and HV\_Buck will immediately stop switching. The FPGA, which is also measuring the input AC voltage, should immediately shut down all power supplies and communication lines. If the OVP condition disappears continuously for 1s, the FPGA should restart the system.

If the aircraft 28V has an overvoltage, it will immediately shut down the primary side switching but continue to report the input voltage to the secondary-side micro. The secondary-side micro will report the overvoltage condition to the FPGA which will immediately turn off the fans and all PSUs. If the OVP condition disappears continuously for 1s, the FPGA should restart the system.

## Output OVP

Based on 3.2.10.2.2 If any output voltage exceeds the corresponding over-voltage trip point threshold for more than the rated duration, the PSU shall immediately disable the faulty output and send an emergency shutdown discrete to the MIU, and then turn off all remaining outputs after corresponding hold-up time.

After the over-voltage protection has been activated, the PSU shall turn-on again only after the ON/OFF switch is driven low and then asserted high again after a minimum period of 1s (i.e. no automatic recovery after an output over-voltage fault event).

The rated output OVP table from the customer specification is as follows:

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

## Notes on Relays (Outputs 3 and 4)

Output 3 is a mechanical relay. The timing of the coil is not critical. The rms outputs from each phase should be like the three phase rms inputs. If the rms outputs differ from the rms inputs by more than TBD percentage (let’s say 3%), then that is equivalent to Output 3 PG being low. However, due to the time it takes the relay contact to close, and the time required to compute rms voltages, the rms input and output measurements should not be compared until the output measurement has stabilized.

Output 4 is a semiconductor switch. Turn on needs to occur near the zero-crossing of phase A to reduce inrush current. Therefore, when the command to turn on Output 4 is received, the FPGA should not turn on the single-phase relay until the voltage across Phase A is close to zero – let’s say between -5V and +5V. As with Output 3, determination of PG for Output 4 also depends on comparison of Phase A input rms to Output 4 rms, except that the voltage drop for Output 4 is higher than for Output 3. Allow the rms voltages between input and output to differ by 10% before causing PG for Output 4 to be low.

## Note on Lamp Status

The lamp\_status\_fpga is a logic signal checking whether the cockpit lamp is receiving voltage. lamp\_status\_fpga should be the same logic as pod\_status\_fpga (with a short delay < 10us) if the circuit driving the cockpit lamp is working. If the logic is different, aside from the short turn-on and turn-off delay, then the lamp circuit is not working. There is currently no requirement to report this; however, the customer IRS document includes non-assigned spare status bytes, so this information might be reported in the future.

## Note on Hall Sensors

Several of the hall sensors include a zero-current reference (e.g. ZCR\_sns\_ADC\_CS\_fpga). The purpose of this measurement is to determine the voltage out of an AC hall sensor which represents zero current. The zero-current reference voltage should be subtracted from the current measurement output voltage to determine the actual ac current which is being measured.