IFSI CNR	**Herschel** **SPIRE DPU Hardware User Manual**	**Ref.:** SPIRE-IFS-PRJ-001390 **Issue:** Issue 1 **Date:** 7/10/2002

HERSCHEL

SPIRE DPU Hardware User Manual

Document Ref: IFSPIRE-IFS-PRJ-001390

Issue 1

Distribution List :

K. King		
B. Swinyard		
R. Cerulli		
S. Molinari		
G. Liu'		
A. Longoni		

Prepared by: Renato Orfei

Date: 7 October 2002

1 INTRODUCTION 5

1.1 Scope of the document 5

1.2 Acronyms 5

1.3 References 8

1.3.1 Applicable Documents 8

1.3.2 Reference Documents 8

2 Design Description 10

2.1 DPU Switch-ON precautions 10

2.2 DPU connectors 10

2.3 Connectors Pin Functions 14

2.4 Analogue Channels 18

3 CPU board characteristics 19

4 Installation 21

5 Jumpers configuration 24

6 Using the ez-ICE emulator 26

7 Programmer’s guide 26

7.1 Data Memory Map 27

7.1.1 Wait States 27

7.2 Program Memory Map 28

7.2.1 Wait States 28

7.3 Interrupt Table 29

7.4 EEPROM 30

7.5 Interrupt Manager 31

7.6 Watchdog & Reset 34

7.7 Interval Timer 35

7.8 32-bit Bus Interface 35

7.9 Data Memory 36

7.10 Program Memory 36

7.11 Bootloader 36

8 Payload Interface board characteristics 38

9 Installation 40

10 Jumpers configuration 40

11 Programmer's guide 41

11.1 Spacecraft interface 41

11.1.1 Software interface 41

11.1.2 Hardware interface 44

11.1.3 Interface to MIL-STD-1553 bus 45

11.2 Fast science data link interfaces 46

11.2.1 Interrupt generation 48

11.2.2 FIFO flags 48

11.2.3 Software interface 49

11.3 Low speed link interface 50

11.3.1 Interface protocol 51

11.3.2 Software interface 55

11.4 32 bit bus interface 57

11.5 Analogue inputs 58

11.5.1 Analogue input interfaces 58

11.5.2 Software interface 59

11.6 Control logic 60

11.6.1 Board register 60

11.7 Board reset 62

Document Status Sheet:

Document Title: Herschel SPIRE DPU Hardware User Manual			
Issue	**Revision**	**Date**	**Reason for Change**
Issue 1.0		7 October 2002	First Issue

Document Change Record :

Document Title: Herschel SPIRE DPU Hardware User Manual	
DOCUMENT REFERENCE NUMBER: SPIRE-IFS-PRJ-001390	
Document Issue/Revision Number: Issue 1	
Section	**Reason For Change**
All	Issue 1

1 INTRODUCTION

1.1 Scope of the document

The purpose of this document is to describe the DPU hardware so that the potential user can find all relevant hardware information.

1.2 Acronyms

AD	Architectural Design
ATP	Acceptance Test Plan
AVM	Avionic Model
CIDL	Configuration Item Data List
CSL	Configuration Status List
CNR	Consiglio Nazionale delle Ricerche
CPP	Common Parts Procurement
CPU	Control Processing Unit
CDMS	Central Data Management System
CDMU	Central Data Management Unit
CQM	Cryogenic Qualification Model
DDD	Detailed Design Document
DPU	Digital Processing Unit
EEPROM	Electrically Erasable Programmable Read Only Memory
EIDP	End Item Data Package
EMC	Electro Magnetic Compatibility
EMI	Electro Magnetic Interference

ESA	European Space Agency
FCU	Focal plane Control Unit
FIRST	Far InfraRed and Submillimeter Telescope
FSDL	Fast Serial Data Link
HK	HouseKeeping
HRS	High Resolution Spectrometer
HW	HardWare
IBDR	Instrument Baseline Design Review
ICD	Interface Control Document
ICDR	Instrument Critical Design Review
DPU	Instrument Control Unit
IHDR	Instrument Hardware Design Review
IFSI	Istituto di Fisica dello Spazio Interplanetario
ILT	Instrument Level Test
ISVR	Instrument Science Verification Review
LCU	Local oscillator Control Unit
LSU	Local oscillator Source Unit
NA	Not Applicable
OBS	On-Board Software
PA	Product Assurance
PDU	Power Distribution Unit
PROM	Programmable Read Only Memory
S/C	SpaceCraft
SCC	SpaceCraft Components
SEU	Single Event Upset
SPIRE	Spectral and Photometric Imaging Receiver
S/S	SubSystem
SVM	Service Module

SW	Software
TBC	To Be Confirmed
TBD	To Be Defined
TBW	To Be Written
TV	Thermal Vacuum
WBS	Work Breakdown Structure
WBS	Wide Band Spectrometer

1.3 References

1.3.1 Applicable Documents

Document Reference	Name
AD1	SPIRE Instrument Specification
AD2	Herschel/Planck Instrument Interface Document Part A
AD3	Herschel/Planck Instrument Interface Document Part B Instrument “SPIRE”
AD4	DPU Subsystem Description
AD5	DPU-ICU P.A. Plan
AD6	DPU/ICU On Board Software Product Assurance Plan
AD7	SPIRE ICD

1.3.2 Reference Documents

Document Reference	Name	Reference
REF1	CPU Board Specification	DPU-SP-CGS-001 Issue 1
REF2	I/F Board Specification	DPU-SP-CGS-002 Issue 1
REF3	DC/DC Board Specification	DPU-SP-CGS-004 Issue 1
REF4	Mother Board Specification	DPU-SP-CGS-005 Issue 1
REF5	CPU Board Design	Herschel CPU Main EM1.DSN
REF6	CPU Board Piggy-Back Design	Herschel CPU DM PIGGY EM1.DSN
REF7	Interface Design	PSI-E000_0.00.DSN
REF8	DC/DC Converter Design	DC-DC FIRST DESIGN1 EM.DSN
REF9	Motherboard Design	MOTHERBOARD_2_D.DSN
REF10	DPU-BSW SOFTWARE REQUIREMENTS DOCUMENT	DPU-SQ-CGS-001 Issue 2
REF11	DPU-BSW SOFTWARE A.D. DOCUMENT	DPU-AD-CGS-001 Issue 1

REF12	Herschel SPIRE DPU OBS SSD	IFSI/OBS/SP/2002-001 Draft 1
REF13	HSO DPU/DPU OBS Product Assurance Plan	IFSI/OBS/PL/2000-001 Issue 1.2
REF14	HSO DPU/DPU OBS User Requirements Document	IFSI/OBS/SP/2000-001 Issue 1.3
REF15	CPU Board User Manual	DPU-MA-CGS-001
REF16	PL IF BOARD USER MANUAL	DPU-MA-CGS-002
REF17	Space Level MIL-STD-1553 BC/RT/MT Advanced Communication Engine (ACE)	Data Book ILC Data Device Corporation 1995
REF18	ACE/Mini-ACE Series BC/RT/MT Advanced Communication Engine Integrated 1553 Terminal…	User's Guide ILC Data Device Corporation, June 1999, REV J-2
REF19	CPU Board S/N 05 Test Report	DPU-RP-CGS-026
REF20	I/F Board S/N 05 Test Report	DPU-RP-CGS-032
REF21	DC/DC Board S/N 03 Test Report	DPU-RP-CGS-031
REF22	Mother Board S/N 05Test Report	DPU-RP-CGS-011
REF23	DRCU ICD	Sap-SPIRE-Cca-25-00

2 Design Description

The AVM unit is electrically, mechanically and thermally representative of the flight unit, but without redundancy. It makes use of standard commercial components. The DPU specifications as far as the hardware is concerned are described in REF1 to REF4, while the hardware design is described in REF5 to REF9.

The AVM software is decribed in REF10 to REF14.

2.1 DPU Switch-ON precautions

The DC/DC converter is designed so as to accommodate large variations of the DPU load, nevertheless it is recommended that all cables be connected prior to switch-on of the DPU.

Once the DPU is connected to the subsystems the following values apply to the power consumption (+- 10%):

V	I (A)	P (W)
28.0	0.45	12.6
26.0	0.46	12
29.0	0.44	12.8

2.2 DPU connectors

In the following figures are shown:

- the DPU box interface control drawing
- the top view of the motherboard/box showing how the three boards are connected to the motherboard itself.

The locations and the functions of the various connectors are shown in the DPU box interface control drawing . The connectors are clearly labelled on the box.

In detail:

J01 DEMA-9P: Primary Power to be connected to a power supply 28 V, >= 1A

J03 DEMA-9S: Prime Bus A Prime to be connected to the relevant “A” CDMS channel

J04 DEMA-9S: Prime Bus B Prime to be connected to the relevant “B” CDMS channel

J07 DBMA-25P: DPU Prime to DCU Prime (CMD&DATA: Fast 0, Slow 0)

J08 DBMA-25P: DPU Prime to MCU Prime (CMD&DATA: Fast 1, Slow 1)

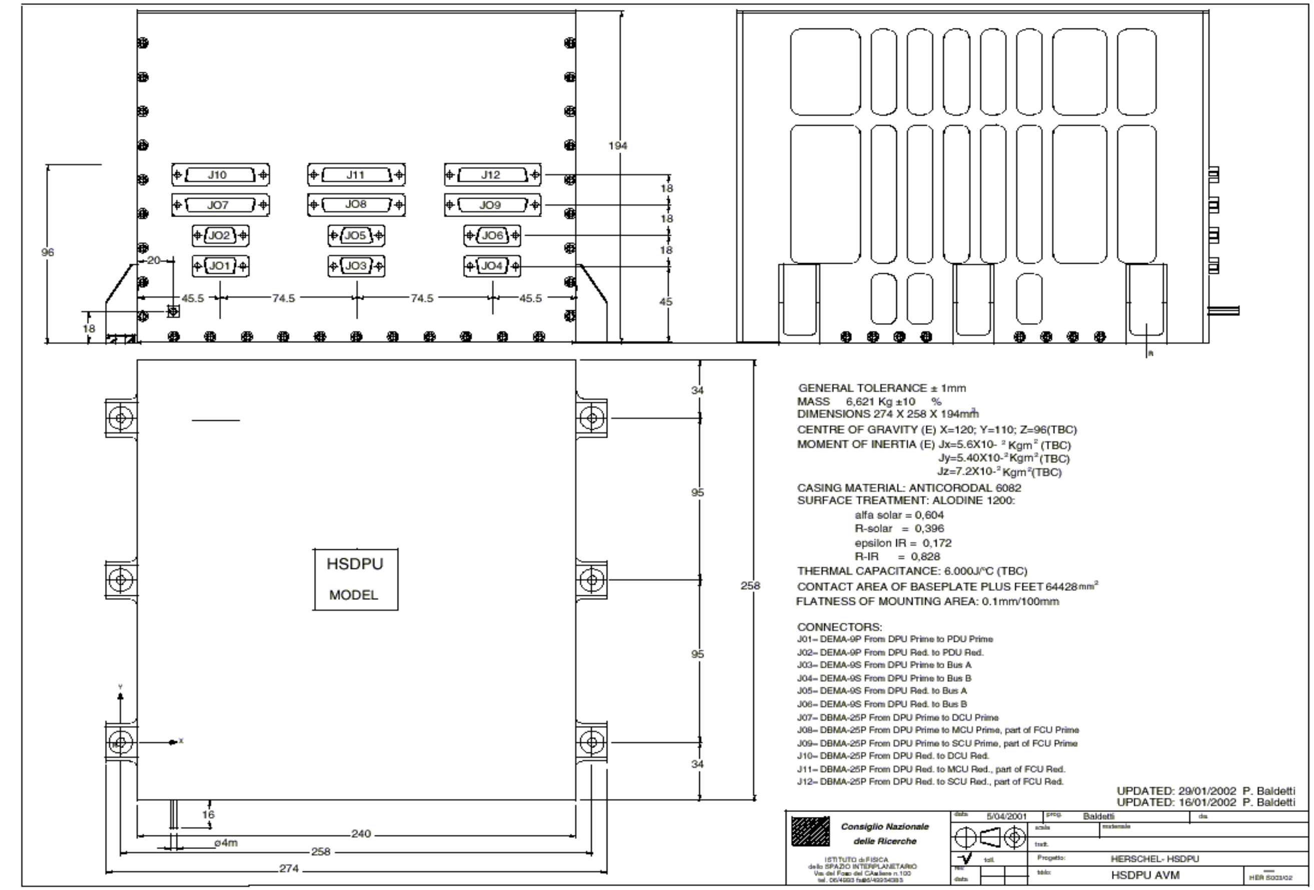
J09 DBMA-25P: DPU Prime to SCU Prime (CMD&DATA: Fast 2, Slow 2)

J10 DBMA-25P: DPU Redundant to DCU Redundant (CMD&DATA: Fast 0, Slow 0)

J11 DBMA-25P: DPU Redundant to MCU Redundant (CMD&DATA: Fast 1, Slow 1)

J12 DBMA-25P: DPU Redundant to SCU Redundant (CMD&DATA: Fast 2, Slow 2)

NOTE: for the AVM the redundant connectors are not mounted, the connectors receptacles are covered with a glued Al foil.



GENERAL TOLERANCE ± 1mm
MASS 6,621 Kg ±10 %
DIMENSIONS 274 X 258 X 194mm

CENTRE OF GRAVITY (E) X=120; Y=110; Z=96(TBC)
MOMENT OF INERTIA (E) Jx=5.6X10-^{2} Kgm^{2} (TBC)
Jy=5.40X10-^{2} Kgm^{2} (TBC)
Jz=7.2X10-^{2} Kgm^{2}(TBC)

CASING MATERIAL: ANTICORODAL 6082
SURFACE TREATMENT: ALODINE 1200:
alfa solar = 0,604
R-solar = 0,396
epsilon IR = 0,172
R-IR = 0,828
THERMAL CAPACITANCE: 6.000J/°C (TBC)
CONTACT AREA OF BASEPLATE PLUS FEET 64428 mm^{2}
FLATNESS OF MOUNTING AREA: 0.1mm/100mm

CONNECTORS:
J01– DEMA-9P From DPU Prime to PDU Prime
J02– DEMA-9P From DPU Red. to PDU Red.
J03– DEMA-9S From DPU Prime to Bus A
J04– DEMA-9S From DPU Prime to Bus B
J05– DEMA-9S From DPU Red. to Bus A
J06– DEMA-9S From DPU Red. to Bus B
J07– DBMA-25P From DPU Prime to DCU Prime
J08– DBMA-25P From DPU Prime to MCU Prime, part of FCU Prime
J09– DBMA-25P From DPU Prime to SCU Prime, part of FCU Prime
J10– DBMA-25P From DPU Red. to DCU Red.
J11– DBMA-25P From DPU Red. to MCU Red., part of FCU Red.
J12– DBMA-25P From DPU Red. to SCU Red., part of FCU Red.

UPDATED: 29/01/2002 P. Baldetti
UPDATED: 16/01/2002 P. Baldetti

Consiglio Nazionale delle Ricerche ISTITUTO di FISICA dello SPAZIO INTERPLANETARIO Via del Foso del CAvaliere n.100 tel. 06/4993 fax06/49934383	data 5/04/2001	prog. Baldetti		da	
		scala	materiale		
		tratt.			
	✓ toll.	Progetto:	HERSCHEL- HSDPU		
	rev. data	titolo	HSDPU AVM		HER 5003/02

Fig.1 Box I/F Control Drawing

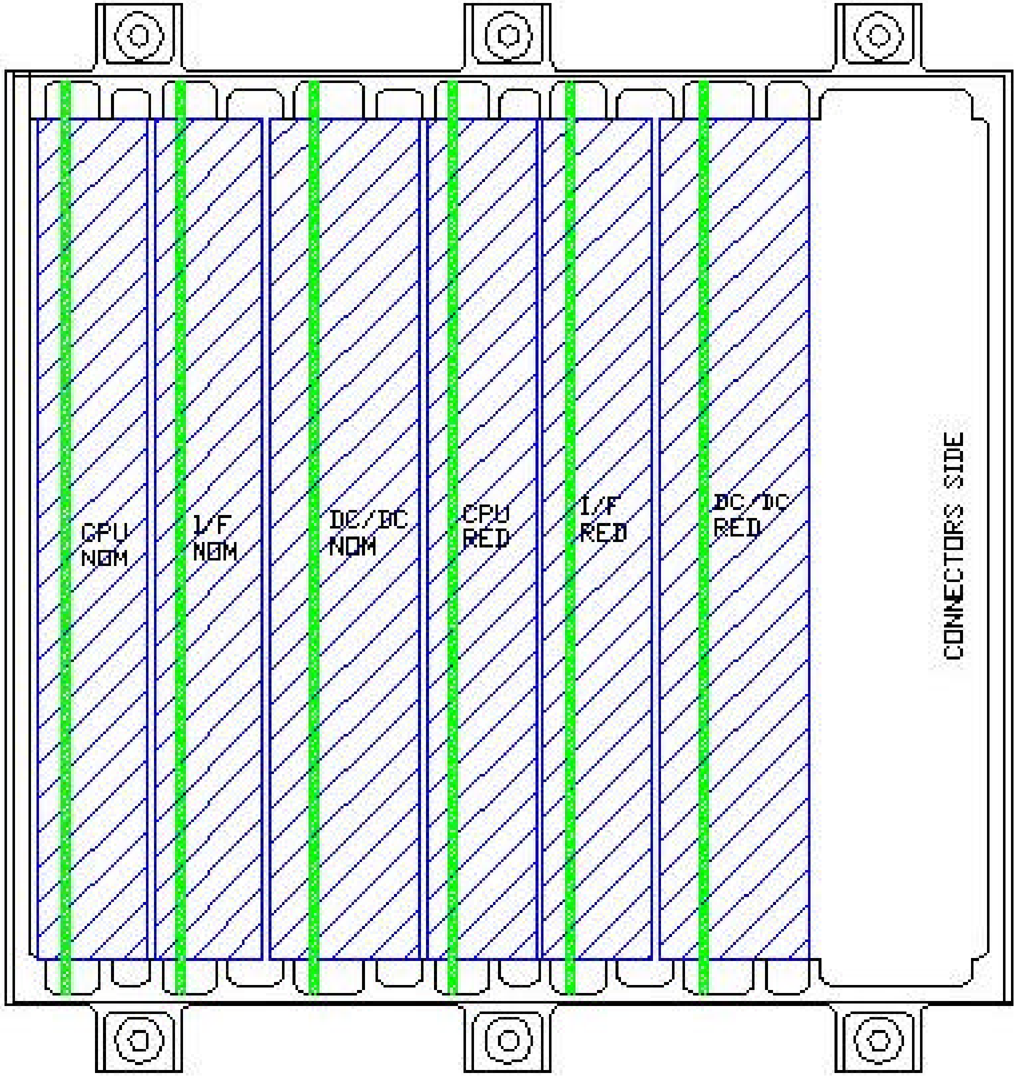
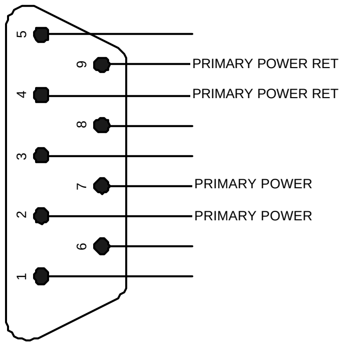


Figure 2 Electronics cards top assembly view

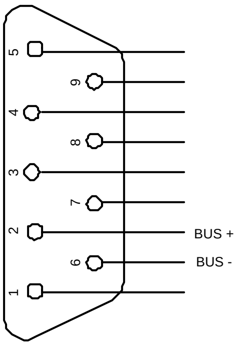
2.3 Connectors Pin Functions

The DPU connectors pin functions are shown in the pertinent sections of in the following:

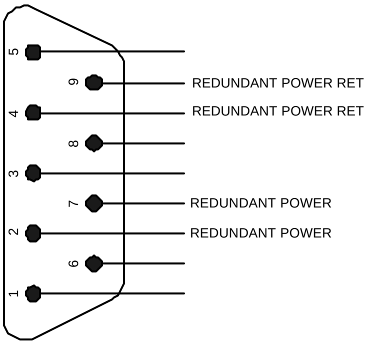
ID: J01
TYPE: DEMA-9P
FUNCTION: PRIMARY POWER



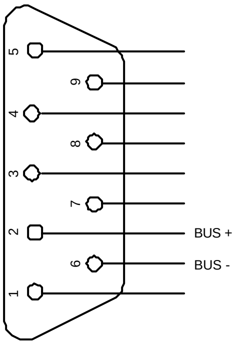
ID: J03
TYPE: DEMA-9S
FUNCTION: PRIME BUS A PRIME



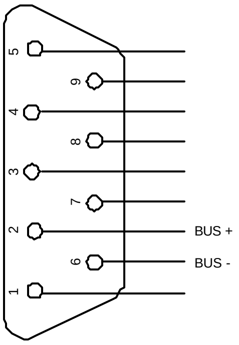
ID: J02
TYPE: DEMA-9P
FUNCTION: REDUNDANT POWER



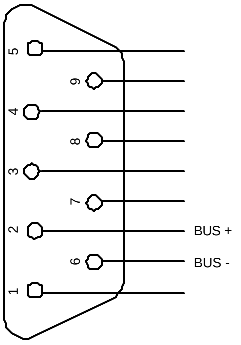
ID: J04
TYPE: DEMA-9S
FUNCTION: PRIME BUS B PRIME



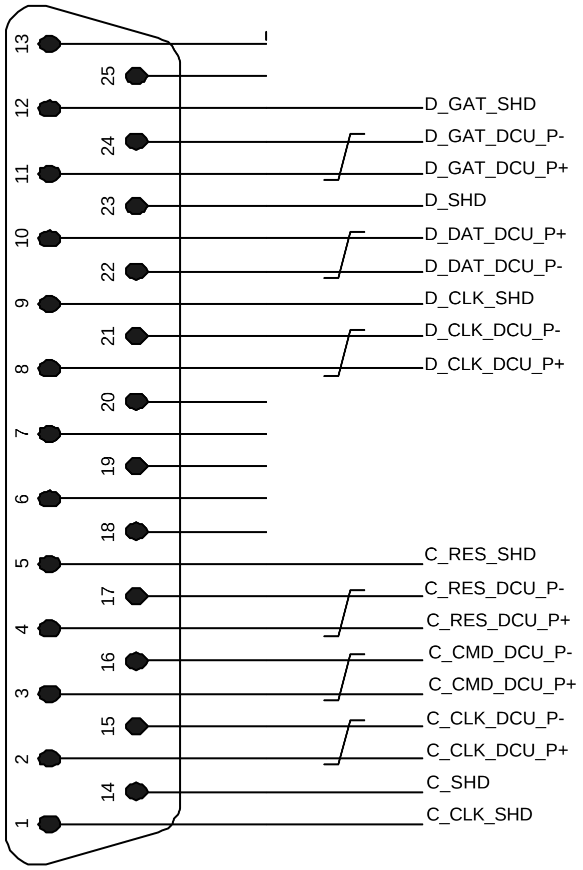
ID: J05
TYPE: DEMA-9S
FUNCTION: RED BUS A RED



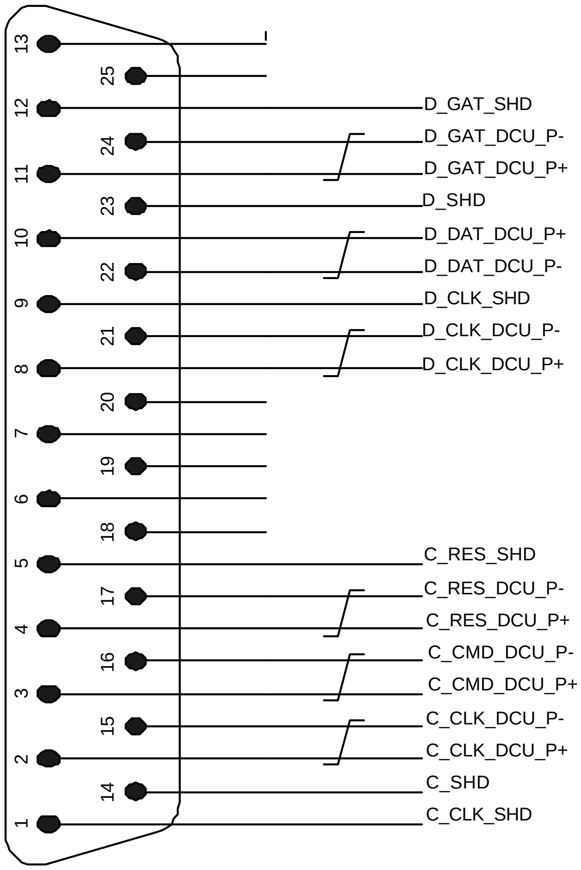
ID: J06
TYPE: DEMA-9S
FUNCTION: RED BUS B RED



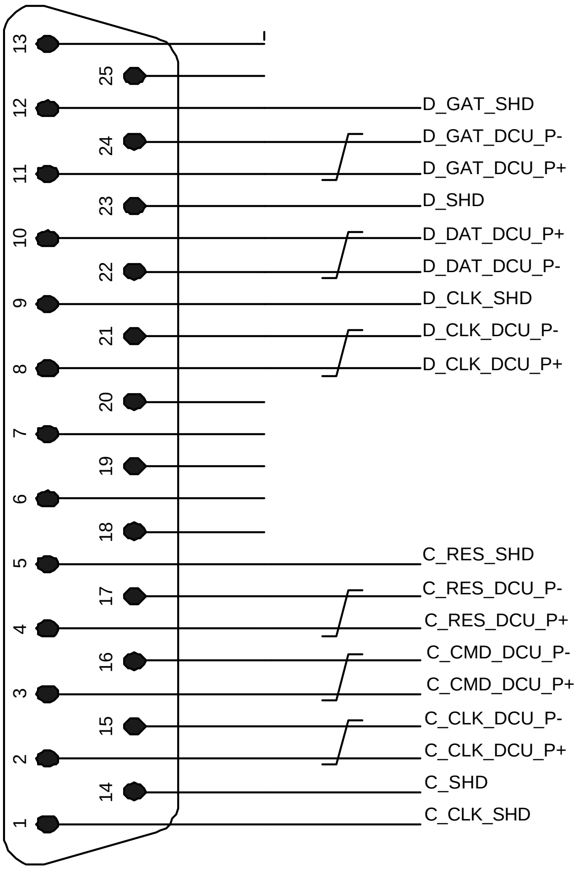
ID: J07
TYPE: DBMA-25P
FUNCTION: DPU PRIME TO DCU PRIME
CMD&DATA: FAST 0, SLOW 0)



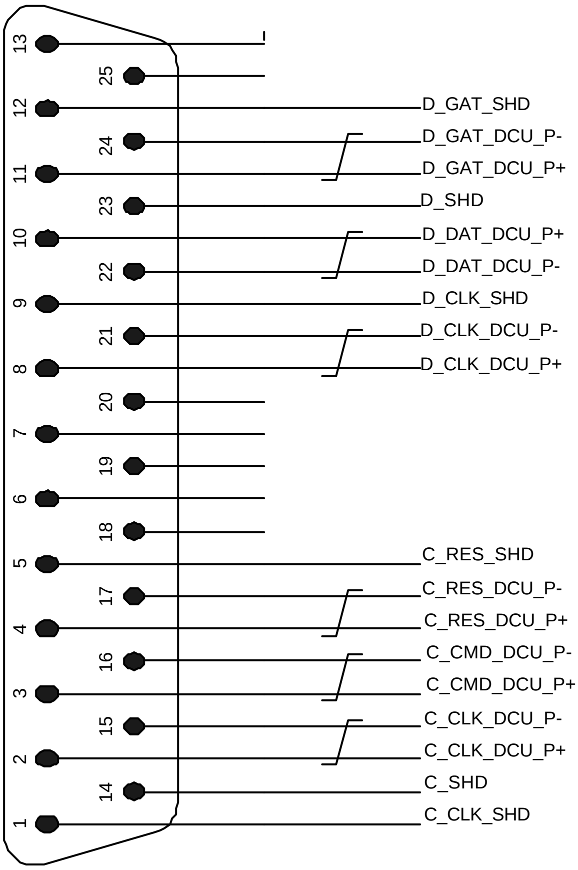
ID: J10
TYPE: DBMA-25P
FUNCTION: DPU RED TO DCU RED
(CMD&DATA: FAST 0, SLOW 0)



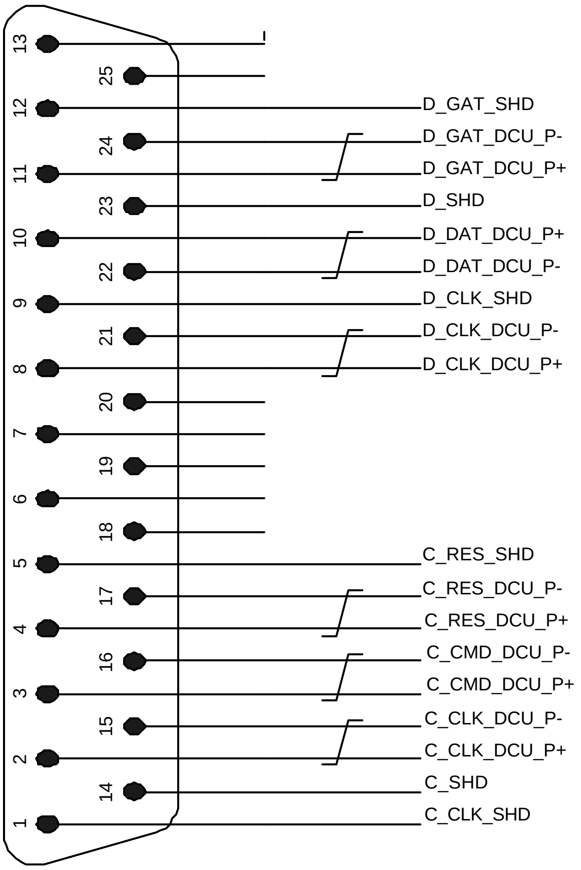
ID: J08
TYPE: DBMA-25P
FUNCTION: DPU PRIME TO MCU PRIME CMD&DATA: FAST 1, SLOW 1)



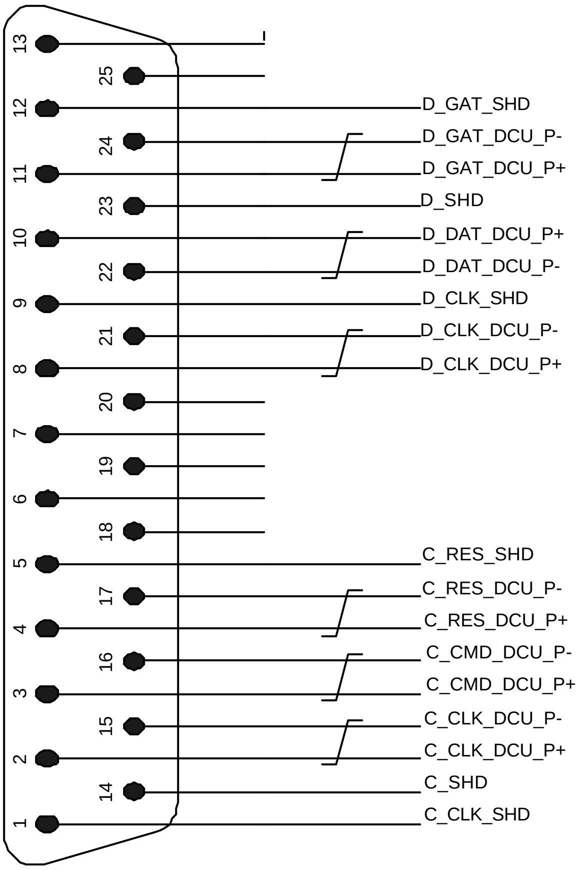
ID: J11
TYPE: DBMA-25P
FUNCTION: DPU RED TO MCU RED (CMD&DATA: FAST 1, SLOW 1)



ID: J09
TYPE: DBMA-25P
FUNCTION: DPU PRIME TO SCU PRIME
CMD&DATA: FAST 2, SLOW 2)



ID: J12
TYPE: DBMA-25P
FUNCTION: DPU RED TO SCU RED
(CMD&DATA: FAST 2, SLOW 2)



2.4 Analogue Channels

The following table shows the relevant information of the DPU analogue channels:

Channel #	Function	Accuracy	Scale
0	SPARE		
1	+5V Monitor	+- 5%	0V => 0 counts 6V => 4095 Counts
2	+15V Monitor	+- 5%	0V => 0 counts 18V => 4095 Counts
3	-15V Monitor	+- 5%	0V => 0 counts -18V => 4095 Counts
4	Temp. Monitor	+- 5%	-50 C° => 0 counts +80 C° => 4095 Counts
5	SPARE		
6	SPARE		
7	+2.5V Reference	+- 5%	0V => 0 counts 5V => 4095 Counts

3 CPU board characteristics

The EM CPU board is a double-eurocard processing unit based on the Analog Devices 21020 DSP. The main characteristics of the board are the following (Figure 3):

- ADSP21020 clocked @20MHz, 50ns instruction cycle time
- 20 MIPS 66MFlops (peak) of instruction rate
- 512K x 48bit (3MByte) program memory (0 Wait States operation)
- 512K x 32bit (2MByte) data memory (0 Wait States operation)
- 256K x 32bit (1MByte) EEPROM memory
- 32K x 8bit EPROM based boot-loader that automatically stores the boot program in program memory after switch-on
- Internal watchdog for monitoring of software execution (disabled via jumper)
- Custom parallel data bus with 24 bit address and 32 bit data
- Interrupt manager, totally configurable, with 6 available interrupt lines
- 32bit programmable interval timer (1µs of resolution)
- Three IEEE1355 interfaces up to 100MHz, (50% of the time machine for the transferring) with 8K x 32bit DPR buffer and LVDS standard outputs
- JTAG plug for the on board software debugging
- AMP connector for test purpose

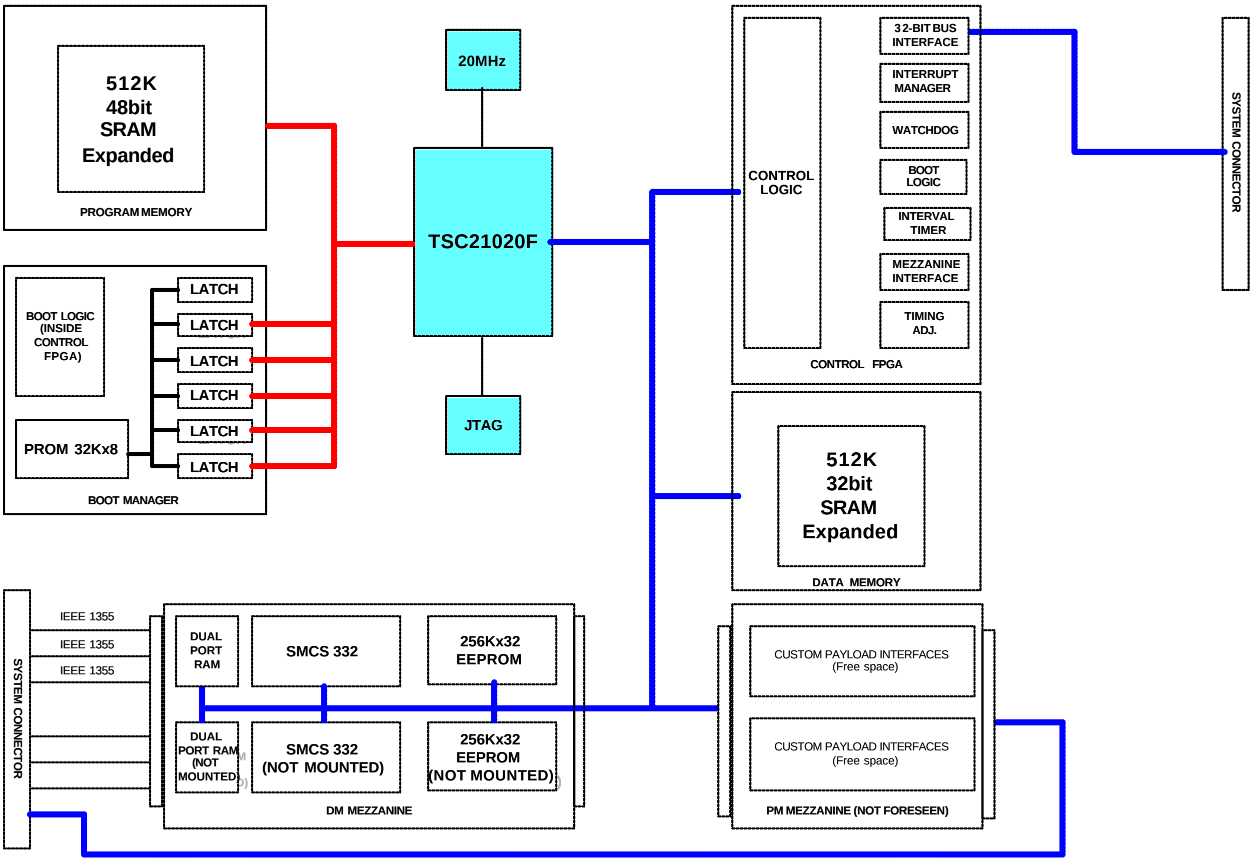


Figure 3 CPU Board block diagram

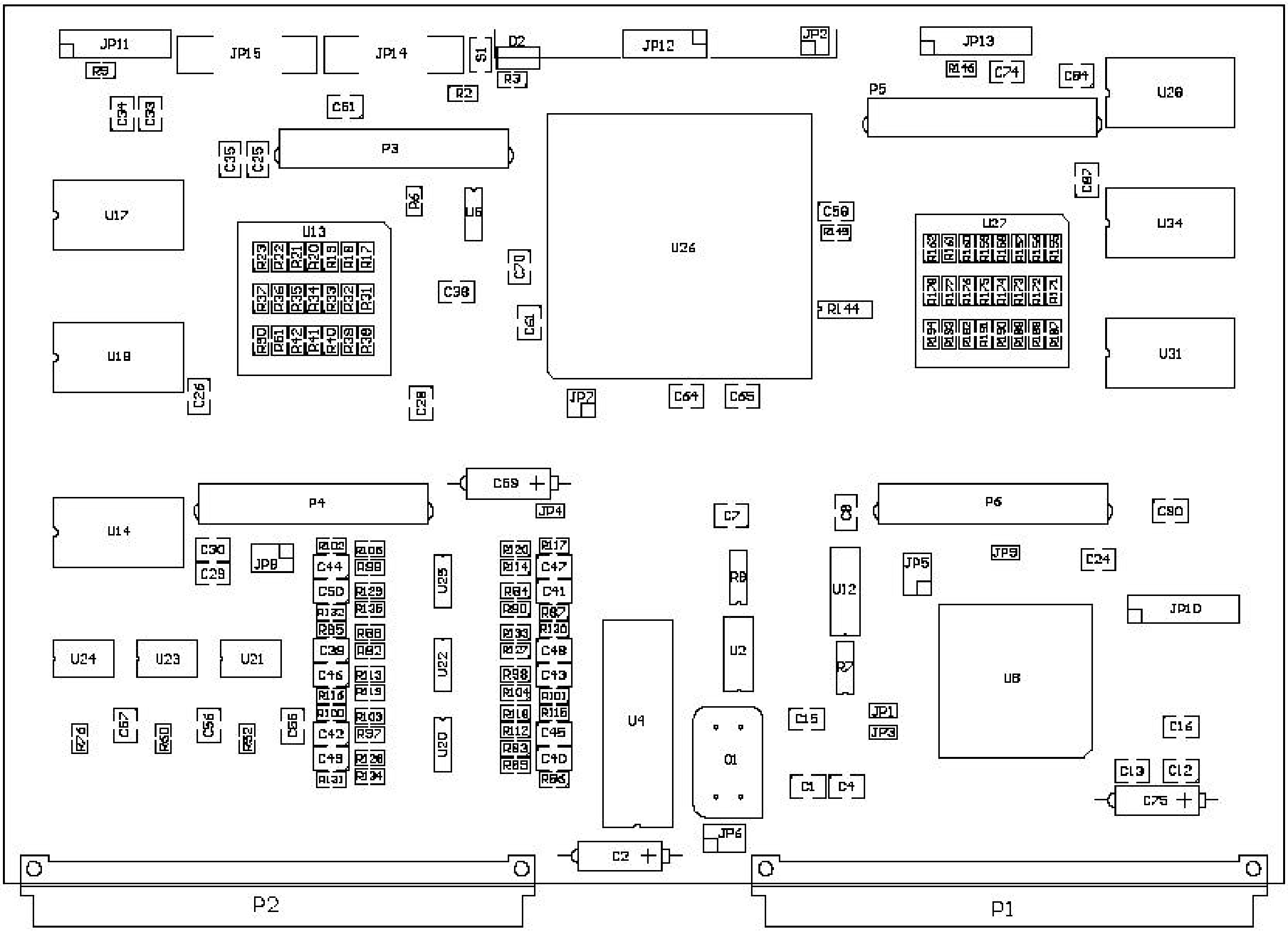
4 Installation

The CPU board is plugged in the dedicated slot on the DPU motherboard. The board has some jumpers to configure all the programmable features (see next paragraph). Three LEDs placed on the board (solder side) are available for test purposes. Their meaning is the following:

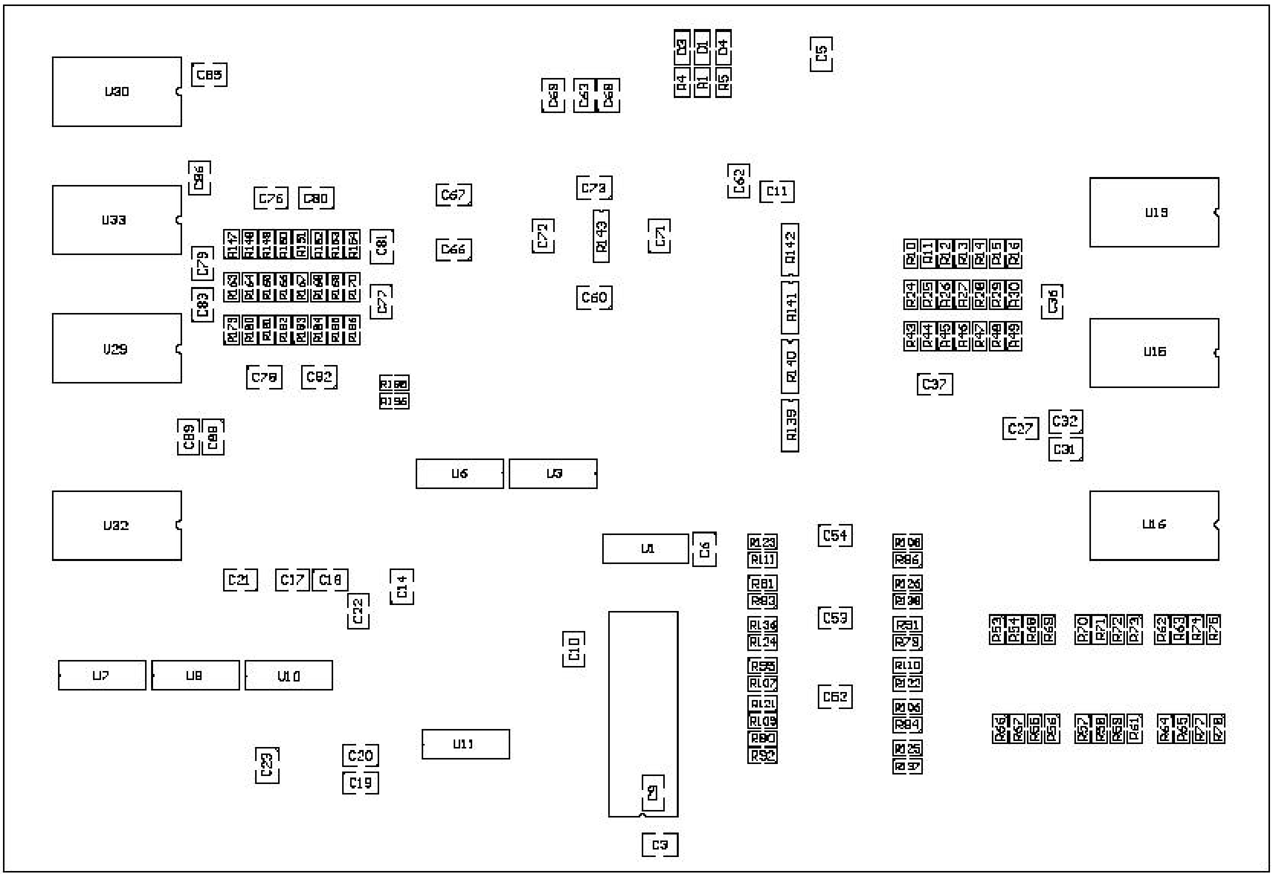
Color	Meaning
Green LED	Power on
Yellow LED	Software active indicator
Red LED	Reset in progress

Table 1

A reset pushbutton (S1) is available near the front side of the board for debugging software.
The layout of the board is shown in Figure 4.


JP11
JP15
JP14
S1
D2
JP12
JP2
JP13
R9
R2
R3
R46
C74
C84
C34
C33
C61
P5
U28
C35
C25
P3
U17
R6
U6
U26
C87
U13
C58
R149
U27
U34
C38
C71
R144
C91
U18
U31
C26
C28
JP2
C64
C65
C69
P4
JP4
U14
C7
C8
P6
C80
C30
C29
JP8
C44
C50
C39
C46
C42
C49
U25
U22
U20
C47
C41
C48
C43
C45
C40
R8
U12
JP5
JP9
C24
JP10
U24
U23
U21
U2
U4
R7
U8
Q1
C15
JP1
JP3
C16
R74
C57
R60
C56
R52
C55
C1
C4
C13
C12
C75
JP6
C2
P2
P1

a]


U30
C85
C86
U33
C76
C80
C67
C73
C62
C11
C5
D3
D1
D4
R4
R1
R5
C79
C83
C81
C77
C66
C72
R143
C71
R142
R141
R140
R139
C60
U19
R10
R11
R12
R13
R14
R15
R16
R24
R25
R26
R27
R28
R29
R30
C36
R43
R44
R45
R46
R47
R48
R49
U16
C37
C27
C32
C31
U29
C78
C82
R195
C89
C88
U6
U9
U32
U1
C6
C54
C53
C52
R123
R111
R81
R83
R134
R124
R107
R109
R98
R86
R126
R91
R79
R110
R106
R84
R125
R97
U16
C21
C17
C18
C14
C22
C10
U7
U8
U10
U11
C23
C20
C19
C9
R53
R54
R70
R71
R72
R73
R62
R63
R74
R76
R57
R61
R64
R65
R77
R78

b]

Figure 4 Layout (a] component side, b] solder side)

5 Jumpers configuration

There are some jumpers that must be set before the board power on. The board is set with a standard default configuration; the programmer can choose any other configuration to satisfy his own requirement. It has to be pointed out that the various boards are serialised and delivered for a defined Herschel instrument so that the jumpers are set accordingly. The following list describes the function of each jumper.

CPU MAIN BOARD

1. **JP1**: it allows to enable/disable the watchdog reset. **It is open during the test** and open for SPIRE.

2. **JP2**: it allows the connection between \IRQ_DEDAC and \IRQ3 of the DSP (jumper position 1-2) or between \IRQ_SFPGA and \IRQ3 (jumper position 3-4). **Normally jumped 3-4**.

3. **JP3**: it allows to enable writing on the lower 128KW of the EEPROM 1 during the programming. **It is jumped during the usual working (writing disabled) and open at closed box or when the user wishes to store the programme in EEPROM**.

4. **JP4**: it allows the connection between \SMCS_RESET_OUT (SMCS reset) and \SMCS_RESET to FPGA (used for PACS). **Normally open**.

5. **JP5**: it allows to configure the PROM width according to the following scheme

JUMPER POSITION	***PROM WIDTH***
3-5 ; 4-6 jumped	4K
3-5 ; 2-4 jumped	8K
1-3 ; 4-6 jumped	16K
1-3 ; 2-4 jumped	32K

6. **JP6**: it allows to change the path of two signals because of the different pin function of PROM (used in FM version) and EPROM (used in EM version).

PROM	JUMP 3-5 ; 2-4
EPROM	JUMP 1-3 ; 4-6

7. **JP7**: it allows the connection between \IRQ_SSMCS and \IRQ2 of the DSP (jumper position 1-2) or between \BUS_IRQM7 and \IRQ2 (jumper position 3-4) (used for PACS). **Normally jumped 3-4**.

8. **JP8**: it allows to let the SMCS332 to drive the LVDS Drivers (jumper position 1-2, 3-4, 5-6) or let the Drivers always on (no jumpers) (used for PACS). **Normally open**.

9. **JP9**: it allows the connection between analog ground (AGND) and digital ground (DGND). **Normally open**.

<u>CPU PIGGY BACK (DM MEZZANINE)</u>

1. **JP1** : it allows to set the SMCS ID lines: offers possibility to use sixteen SMCS within one chip select (used for PACS). **Normally jumped 1-2, 3-4, 5-6, 7-8**.

2. **JP2** : it allows to set the control of the SMCS by host (jumper position 5-6) or by link (jumper position 1-2) (used for PACS). **Normally jumped 5-6**.

3. **JP3** : it allows to select Big/Little endian mode of the host control interface (HOCI) of the SMCS. By connecting HOSTBIGE to either Vcc (jumper position 1-2) or GND (jumper position 5-6) the HOCI is in big endian mode or little endian mode respectively (used for PACS). **Normally jumped 1-2**.

6 Using the ez-ICE emulator

In order to test and debug the software directly on the board a JTAG plug is available for the connection of an Analog Devices EZ ICE emulator as shown in the Figure 5. The plug has a reference pin so that it is impossible to insert the EZ ICE probe in a wrong way.

Be careful when resetting the board with the S1 pushbutton when the emulator is running: the operation may cause a loss in synchronisation between the emulator and the board. This could cause the emulator to crash.

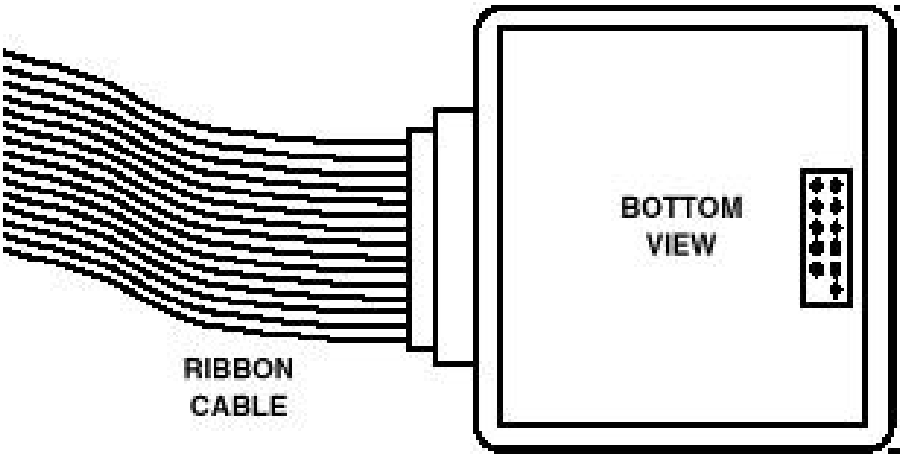


Figure 5 EZ ICE Test Probe

7 Programmer’s guide

In the following paragraphs the programming of each block of the board will be described in detail.

7.1 Data Memory Map

In the data memory map all the peripherals connected to the CPU are mapped, in agreement with the following table 2:

Range	Bank	Peripheral
0x00000000-0x0007FFFF	0	Data Memory (512KWord 32-bit wide)
0x20000000-0x3FFFFFFF	1	PM Mezzanine (not used)
0x40000000-0x40001FFF	2	IEEE1355 interface (8Kx32-bit wide)
0x80000000-0x8003FFFF	3	EEPROM (256Kx32-bit wide)
0x81000000-0x81FFFFFF	3	Interval timer
0x82000000-0x82FFFFFF	3	Watchdog
0x83000000-0x83FFFFFF	3	Interrupt manager
0x84000000-0x84FFFFFF	3	SMCS332 configuration registers
0x88000000-0x8FFFFFFF	3	32-bit Bus Interface

Table 2

The two most significant digits are not decoded by hardware. Their value is set in the bank registers. The table reflects the default configuration:

- DMBANK1 = 0x20000000
- DMBANK2 = 0x40000000
- DMBANK3 = 0x80000000

Changing the values of these registers will be reflected in the memory map. Consequently if there isn't any reason to have different bank sizes these registers should be left as they are at power-up.

7.1.1 Wait States

The wait states for the Data Memory **must be set by software when the program starts execution**. The DMWAIT register must be set as in table 3:

	Number of WS	Bits to set	WS Mode	Bits to set
Bank0	0x00	4-2	Internal SW (0x01)	1-0
Bank1	0x02	9-7	Both int & ext (0x02)	6-5
Bank2	0x00	14-12	Internal SW (0x01)	11-10
Bank3	0x01	19-17	Both int & ext (0x02)	16-15
	DMWAIT=0x00030541			

Table 3

7.2 Program Memory Map

In the program memory only the SRAM is present, mapped according to the following table 4:

Range	Bank	Peripheral
0x000000-0x07FFFF	0	Program Memory (512KWord 48-bit wide)

Table 4

The same considerations made for the data memory are valid for the program memory. The default configuration is:

- PMBANK1 = 0x800000

7.2.1 Wait States

The wait states for the Program Memory **must be set by software when the program starts execution**. The PMWAIT register must be set as shown in table 5:

	Number of WS	Bits to set	WS Mode	Bits to set
Bank0	0x0	4-2	Internal only (0x01)	1-0
Bank1	0x0	9-7	Internal only (0x01)	6-5
	PMWAIT=0x00000021			

Table 5

7.3 Interrupt Table

DSP IRQ #	Peripheral
0	Interrupt manager (see paragraph 7.5)
1	IEEE1355 dual port ram
2	Not used
3	Not used (highest priority)

Table 6

All the interrupts are active low. The programmer must enable the interrupts by setting at 1 the bit 12 in the register MODE1. The four interrupts can be set as edge or level mode by setting the MODE2 register as in table 7:

IRQ #	Bit
0	0
1	1
2	2
3	3
0=level 1=edge	

Table 7

The interrupt 0 must be configured as level in any case. The four interrupts can be read in the IRPTL register (1=asserted) in the following positions as shown in table 8:

IRQ #	Bit
0	8
1	7
2	6

3	5

Table 8

Each interrupt can be masked writing a 0 in the corresponding bit in the IMASK register. The bits corresponding to each interrupt are the same than the IRPTL register.

When an interrupt is asserted the program sequencer jumps to the interrupt service routine. The ISR are 8-instruction wide and are mapped in memory agree with the following table 9:

IRQ #	PM Address
0	0x000040
1	0x000038
2	0x000030
3	0x000028

Table 9

In order to avoid that high level interrupts are not detected during the execution of an ISR related to a lower level IRQ, the interrupt-testing mode of the DSP should be activated setting the bit 11 of the MODE1 register.

7.4 EEPROM

The EEPROM block contains the program to be executed. It is formed by a single 256Kx32 bit (32-bit parallelism) EEPROM. The chip is allocated on DM Mezzanine and the 32 bits are mapped in the bits D8-D39 of the DSP data bus. An additional EEPROM can be mounted on the mezzanine to extend the memory of the block.

The lower half (128K x 32 bit) of the memory is writing-protected; an hardware jumper (JP3, see section 5) allows writing during the programming stage. The device is capable of in-system electrical Byte and Page programmability. It has a 512-kilobyte Page Programming function to make its erase and write operations faster.

The transfer of the program from this module to the program memory and the reconstruction of the 48-bit words from the 8-bit code, shall be performed by software.

7.5 Interrupt Manager

The interrupt manager expands the overall number of interrupts adding 8 more interrupt lines to the 4 native of the DSP.

This peripheral is mapped as a set of registers starting at the base address specified in the DM map and is connected to the interrupt 0 of the DSP.

The interrupts are connected to this peripheral as indicated in figure 6 and in Table 10.

Please note that the FSDL #3 is SPARE.

IRQM#	Peripheral
0	PM Mezzanine (not used, kept low)
1	PM Mezzanine (not used, kept low)
2	Interval timer
3	Bus interface (FSDL 0)
4	Bus interface (FSDL 1)
5	Bus interface (FSDL 2)
6	SPARE
7	Bus interface (MIL-STD 1553)

Table 10

The programmer must set the interface at start-up so that each interrupt is configured with the correct polarity.

The meaning of the registers is explained in the following table 11 along with the corresponding location in memory (*= default).

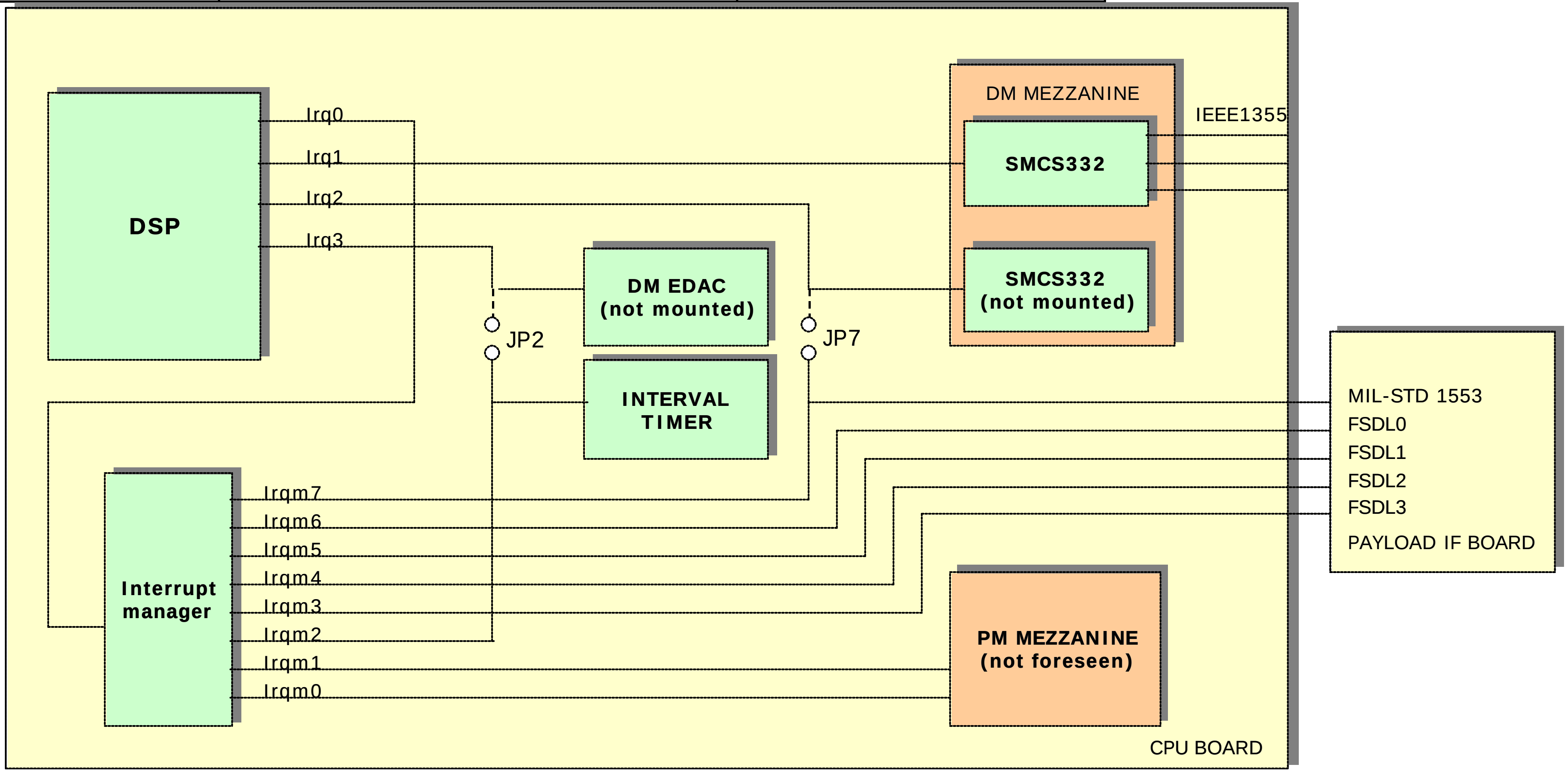


Figure 6 Interrupts distribution (FSDL #3 SPARE)

DM Address	Register	Type
BASE+0x00	IRQ level register (0=high*, 1=low)	R/W
BASE+0x01	IRQ type register (0=edge*, 1=level)	R/W
BASE+0x02	IRQ mask register (0=masked*, 1=active)	R/W
BASE+0x03	IRQ acknowledge register (0=ignored, 1=acknowledge)	W
BASE+0x03	IRQ vector register (0=no interrupt, 1=interrupt pending)	R

Table 11

The interrupt service routine related to DSP IRQ0 must read the vector register to know which peripheral requested an interrupt, then it must perform the operations related to the interrupt and has to write the acknowledge register by setting at 1 the bits corresponding to the interrupts to acknowledge.

The registers are 8-bit wide, mapped on the bits 15-8 of the data memory. For each register the link between the interrupt number and the bit number is shown in the following table 12:

IRQM#	Bit
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

Table 12

All the bits of the registers are set to 0 at power up.

7.6 Watchdog & Reset

N.A.

7.7 Interval Timer

A programmable 32-bit binary interval timer with 1 µs of timing resolution is implemented in the FPGA and it is visible in the data memory space as a set of three registers.

The meaning of the registers is explained in the following table 13 along with the corresponding location in memory.

DM Address	Register	Type
BASE+0x00	Control register	R/W
BASE+0x01	Loading register	W
BASE+0x02	Read-Back register	R

Table 13

The registers are 32-bit wide, mapped on the bits 39-8 of the data memory. The control register allows to manage the counter by programming the first 3 bit according to the scheme hereafter:

D0 : 0 Start counting / 1 Stop counting

D1 : 0 Disable interrupt / 1 Enable interrupt

D2 : 0 Reset timer / 1 Active timer

D3÷D31 : Not used

The loading register allows to set the programmable value; an edge-mode interrupt is generated whenever the programmed time is elapsed. If a new value is set in the loading register while the timer is working, the actual cycle won't be affected; only from the next counting the new interval will be taken into account.

The read-back register is an only reading-register and permits the counting monitoring.

7.8 32-bit Bus Interface

The internal bus is directly derived from the DSP data bus. On this bus are mapped all the external peripherals. The control logic block decodes eight chip select, according to the following table 15:

\CS	Address range
0	0x88000000-0x88FFFFFF
1	0x89000000-0x89FFFFFF
2	0x8A000000-0x8AFFFFFF
3	0x8B000000-0x8BFFFFFF
4	0x8C000000-0x8CFFFFFF
5	0x8D000000-0x8DFFFFFF
6	0x8E000000-0x8EFFFFFF
7	0x8F000000-0x8FFFFFFF

Table 15

The timing of the bus is compliant to the 21020 read and write timing with 1 W/S for the devices addressed by CS0÷CS6; the last one has 6 hardware W/S to address slow peripherals. The 32 bits shall be mapped in the bits D8-D39 of the DSP data bus.

Five interrupts are available for the peripherals mapped on the bus as explained in paragraph 7.5.

7.9 Data Memory

The data memory is 512 KWord, 32-bit wide. It can be accessed in 0 W/S mode.

7.10 Program Memory

The program memory is 512 KWord, 48-bit wide. It can be accessed in 0 W/S mode.

7.11 Bootloader

The bootloader program is automatically loaded by hardware in program memory after a hardware reset, starting from location 0x000000. Its aim is to copy the main program, stored in the EEPROM block, in the program memory, after internal memory checks.

The bootloader is stored in a 32Kx8 PROM (EPROM in EM version). A jumper (JP5) makes it possible to insert a differently sized PROM; if a 32Kx8 chip is chosen the boot-loader program cannot exceed the limit of (32x1024)/6=5460 instructions.

The programmer must also keep in mind that the DSP starts execution from the location 0x000008.

8 Payload Interface board characteristics

The PL IF board is based on the Actel 16000 gates RT54SX32S (A54SX32A for the EM version). The main characteristics of the board are the following (see the block diagram in figure 7):

- MIL-STD-1553 RT-mode serial interface up to 400KHz (burst mode) for the connection to the spacecraft
- 16 MHz square wave oscillator onto the board (for the 1553 terminal unit)
- Four (3 for SPIRE) Fast Science Data Link (FSDL) ports at 1 MHz for the subsystems interfacing.
- Low Speed Link serial bus at 312.5 KHz towards the 3 sub-systems, for commands sending to all sub systems and HK reception.
- Eight, not conditioned, 0-5V, single ended, analogue inputs
- 32-bit parallel system bus
- Software interface implemented by registers mapped on the parallel bus address space
- AMP connector for test purpose (EM version only)

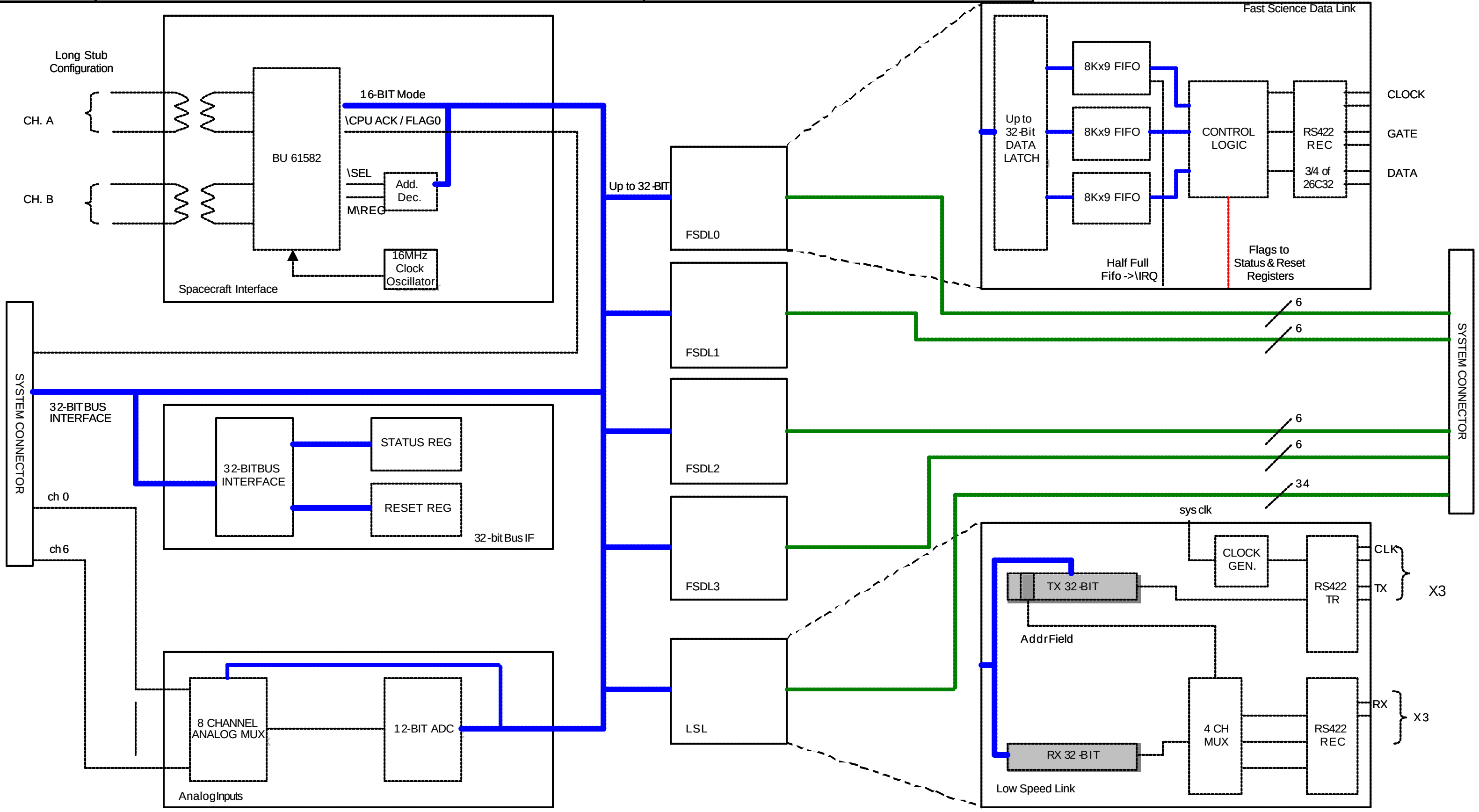


Fig. 7 PL IF Board block diagram

9 Installation

The PL IF board is a double-eurocard and must be plugged in the dedicated slot on the motherboard (see figure 2. The board has some jumpers to configure all the programmable features (see next paragraph). In figure 8 their position on PCB is shown.

10 Jumpers configuration

There are some jumpers that must be set before the board power on. The user has to choose the right configuration according to the requirements. It has to be pointed out that the various boards are serialised and delivered for a defined Herschel instrument so that the jumpers are set accordingly. The following list describes the function of each jumper:

1. **JP1**: it allows to set the FPGA's management of serial links (FSDL and LSL) according to the different PAYLOAD: **SPIRE jumped.**

2. **JP2**: it allows to select between the nonzero wait mode (jumper position 1-2) and the zero wait mode (jumper position 2-3) of BU61580 (U2). **Normally jumped 1-2.**

3. **JP3**: it allows to set the External Time Tag Clock input of BU61580 (U2). If timer is used jump 2-3, if not jump 1-2. **Normally jumped 1-2.**

4. **JP4**: it allows to send \READYD output signal of BU61580 (U2) to FPGA (U30) (jumper position 2-3) or to FLAG1 of DSP (jumper position 1-2). **Normally jumped 2-3.**

5. **JP5**: it allows to determine whether the FPGA is in dedicated or flexible mode. **Normally open.**

6. **JP6**: Active probe for FPGA debugging.

7. **JP7÷JP11**: they allow to set the Remote Terminal Address. **Defined by the user.**

11 Programmer's guide

In the following paragraphs the programming of each block will be described in detail.

11.1 Spacecraft interface

The spacecraft interface is the MIL-STD-1553B standard serial link at 100Kbit/s up to 400Kbit/s (burst mode). This interface is implemented with a dedicated hybrid circuit, made by DDC: BU61580 (U2, see Figure 8 .). It integrates dual transceivers, protocol, memory management and processors interface logic, 4Kx16 bit of Dual Port RAM and a Time Tag Register used to allow the software to manage the "on board time". The timer rollover generates an interrupt to be used to increment the software time registers. The clock of the Time Tag Register implies the time resolution. The programmable resolution is 2, 4, 8, 16, 32 or 64µs per LSB.

The terminal unit is used in 16-bit buffered mode [REF18].

11.1.1 Software interface

The SC interface registers and the interface memory space are mapped in the same system bus I/O space (/CS 7). The selection between the configuration registers and the memory space (16 Kwords) is implemented by using bit A14 of the CPU address bus:

A14	BU61580 Internal Peripheral
0	Memory
1	Configuration Registers

Table 16 BU61580 internal peripheral selection

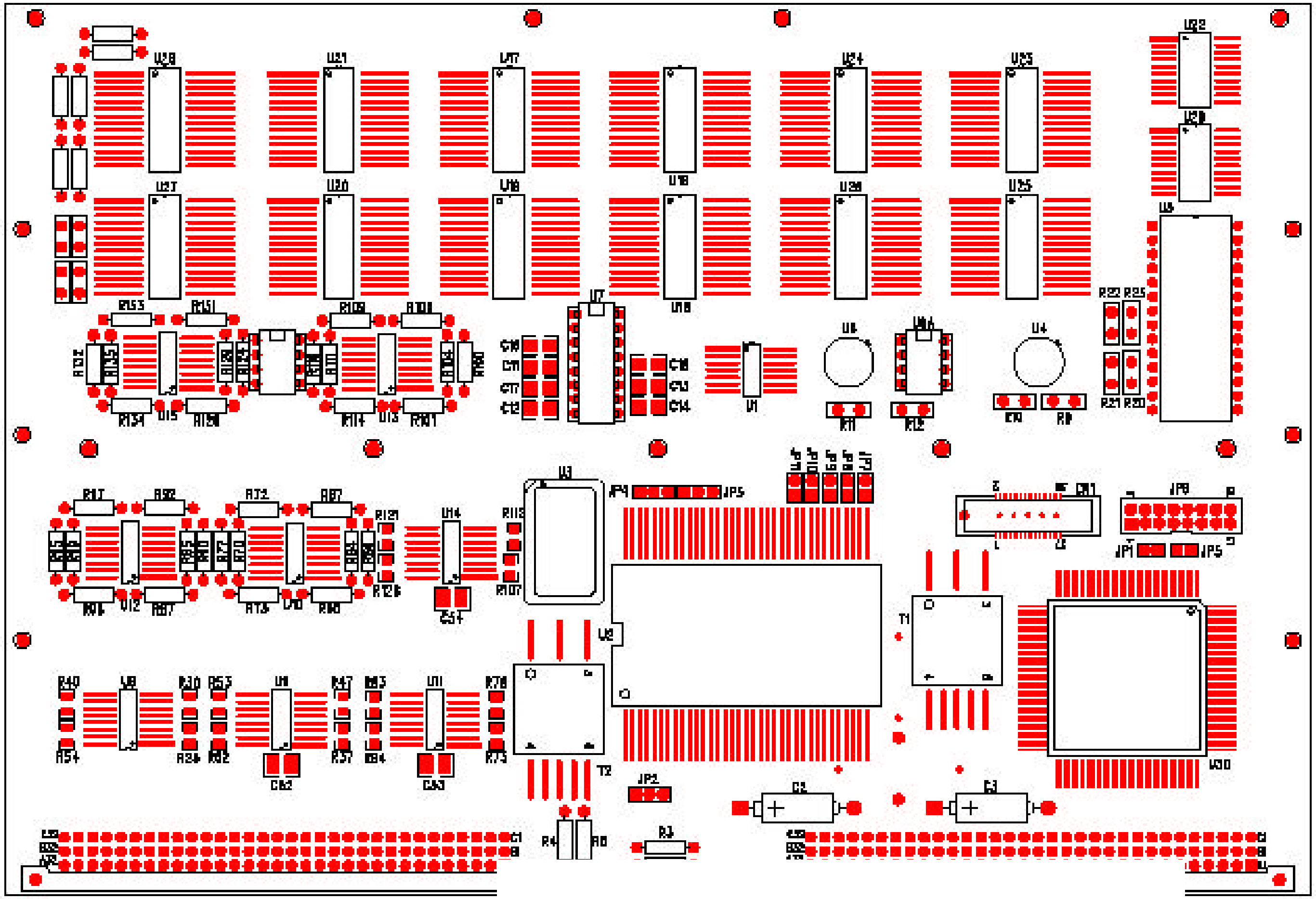


Figure 8 .Board layout (component side)

The internal configuration registers are reported in table 17.

Address	Register	Size [bit]	Direction
0x4000	Interrupt mask register	16	read/write
0x4001	Configuration register #1	16	read/write
0x4002	Configuration register #2	16	read/write
0x4003	Start/Reset register	16	write
0x4003	BC/RT Command Stack Pointer register	16	read
0x4004	BC Control Word/RT Sub-address Control Word register	16	read/write
0x4005	Time Tag register	16	read/write
0x4006	Interrupt Status register	16	read
0x4007	Configuration register #3	16	read/write
0x4008	Configuration register #4	16	read/write
0x4009	Configuration register #5	16	read/write
0x400A	Data Stack Address register	16	read/write
0x400B	BC Frame Time Remaining register	16	read/write
0x400C	BC Time Remaining to Next Message register	16	read/write
0x400D	BC Frame Time/RT Last Command/MT Trigger Word register	16	read/write
0x400E	RT Status Word register	16	read
0x400F	RT BIT Word register	16	read
0x4010	Test Mode register 0	16	
0x4011	Test Mode register 1	16	
0x4012	Test Mode register 2	16	
0x4013	Test Mode register 3	16	
0x4014	Test Mode register 4	16	
0x4015	Test Mode register 5	16	
0x4016	Test Mode register 6	16	
0x4017	Test Mode register 7	16	
0x4018 to 0x401F	Reserved	16	

Table 17 MIL-STD-1553 registers

All registers are described in detail in [REF17].

For the data exchange, a 4Kx16 bit (16Kx16 bit for FM version) internal Dual Port RAM is used and it is mapped according to table 18:

Address	Peripheral	Size [bit]	Direction
0x0000÷0x3FFF	Buffer RAM	16	read/write

Table 18 SC interface buffer memory

The \INCMD hardware flag is mapped on the board status register (see table 29) whereas the \SSFLAG pin is not connected.

11.1.2 Hardware interface

11.1.2.1 HW reset

A hardware reset, acting on the MSTCLR pin of the terminal, is mapped in the reset register of the board.

11.1.2.2 Handshake

This terminal unit is mapped as slow peripheral (\CS7) of the system bus; in order to adjust the timing, an acknowledge management mode is foreseen [REF18].

A hardware acknowledge generation is implemented in the FPGA to add wait states if needed. The functional circuit is shown in the following figure 9.

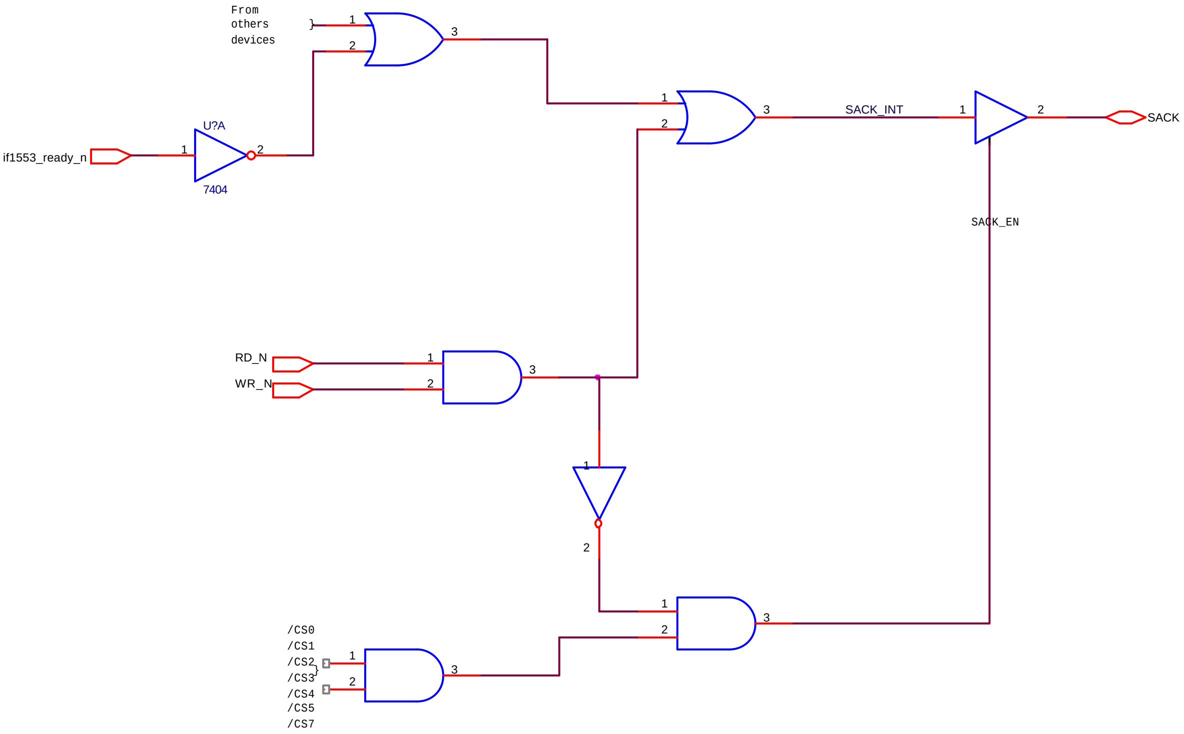


Fig. 9 Sack generation

11.1.3 Interface to MIL-STD-1553 bus

The interface to MIL-STD-1553 bus is the transformer coupled configuration (long stub). Two isolation transformers (one per section) are present on the board. Since the terminal unit is powered at ±15V, the transformers have a turns ratio of 2:1.

11.2 Fast science data link interfaces

The high speed data link (science data link) is made by three monodirectional fast synchronous serial input interfaces, each of which with 8 KW 16 bits FIFO. The FIFOs half full signal generates an interrupt on the DSP. Three independent interfaces are required since simultaneous data transfer can occur. The high speed I/F will transfer data from ICU/DRU sub-units to DPU as 16 bit words using a clock up to 2.5 MHz (baseline 1 MHz). Being the interface unidirectional, all signals are generated by ICU/DRU. The DPU side of the 3 high speed I/F is shown in the following figure 10:

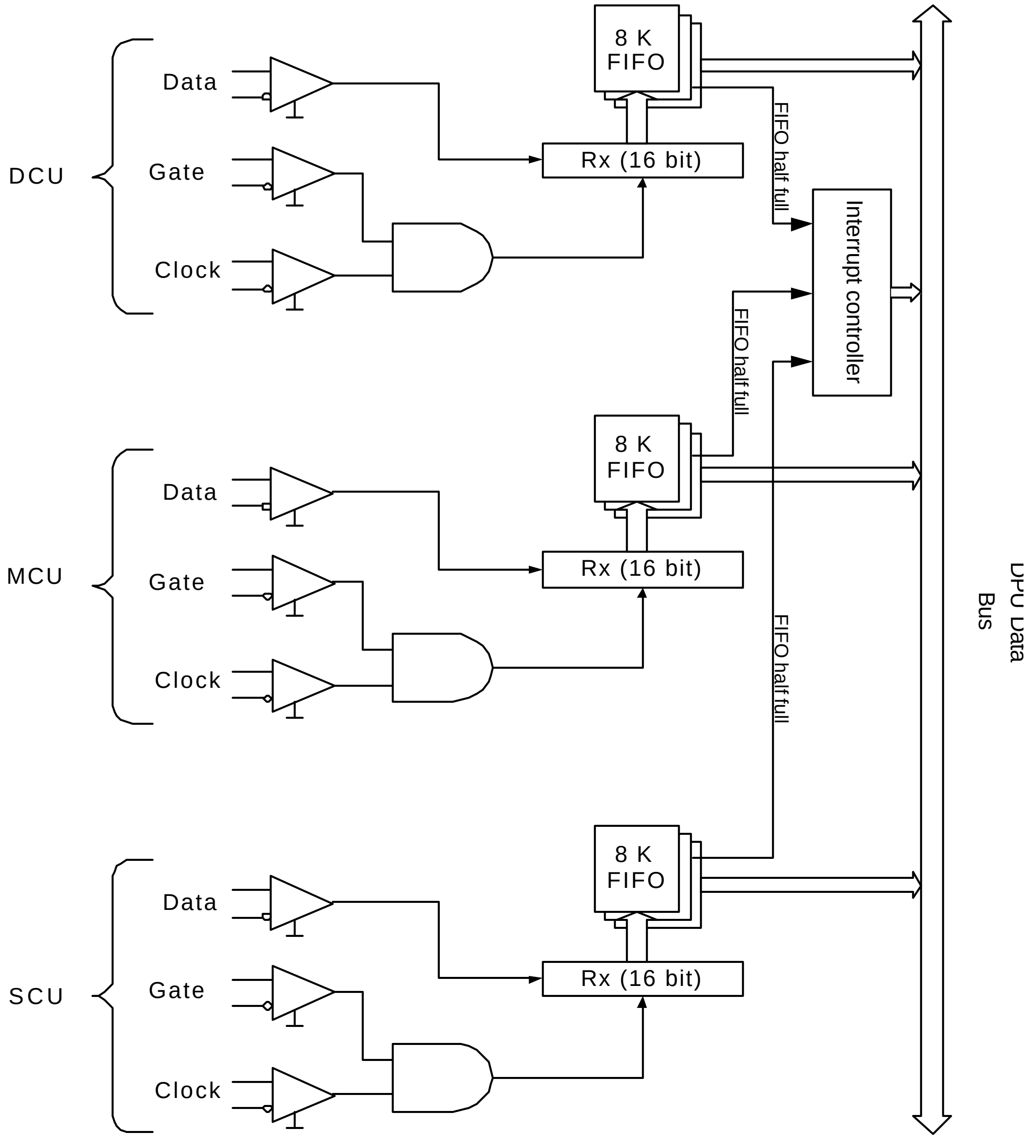


Figure 10 Detailed monodirectional high speed interface

The clock, gate and data signals coming from the subsystems are as in figure 11.

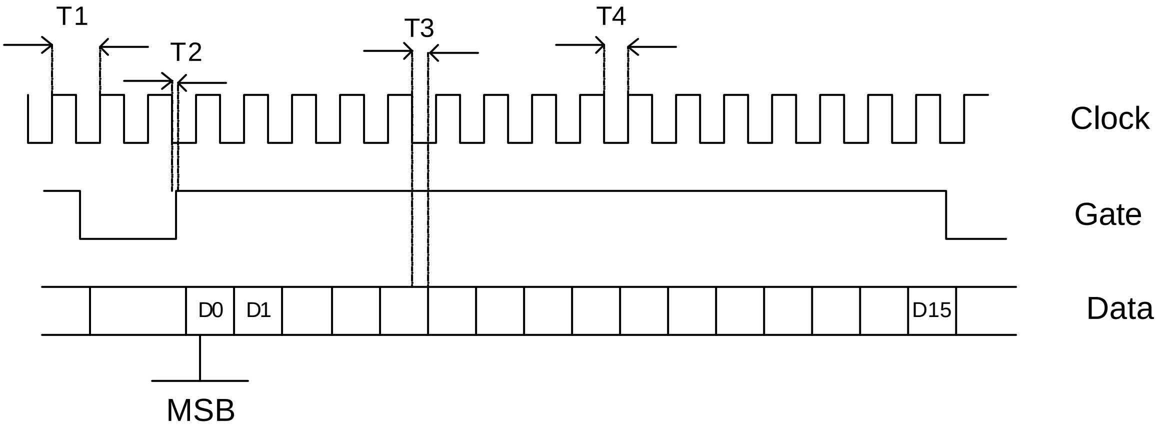


Figure 11 High speed interface protocol

Signal lines are defined as in figure and the clock frequency is 1 MHz. The tolerances on the indicated time parameters are:

Parameter	Min. Value (TBC)	Max. Value (TBC)	Unit
T1	990	1010	ns
T2	0	100	ns
T3	0	300	ns
T4	450	550	ns

11.2.1 Interrupt generation

When the FIFO reaches the half quantity of data, an edge mode (150 nS) interrupt is generated; each FSDL provides a dedicated interrupt line as shown in table 19

IRQM#	Peripheral
4	Bus interface (FSDL 0)
5	Bus interface (FSDL 1)
6	Bus interface (FSDL 2)
7	Bus interface (FSDL 3)

Tab. 19 FSDL interrupts

11.2.2 FIFO flags

In the status register (see table 29) the empty FIFO (EF), half full FIFO (HF) and full FIFO (FF) of each link are mapped; an echo of the gate signals (ACQ) are present in the status register to inform the system about the receiving data from the payload. The status register is implemented in the FPGA. An echo of the empty and full FIFO flags is directly in the FRX registers of each FSDL interface (see figure 12).

Four FIFO error flags (one per link) are mapped in the status register to inform that a misalignment is occurred. The logical implementation is shown in figure 12. The error flag is generated by empty FIFO flags and it is masked during the FIFO writing. The "inhibit" signal is the gate signal of the RS-422 receiver.

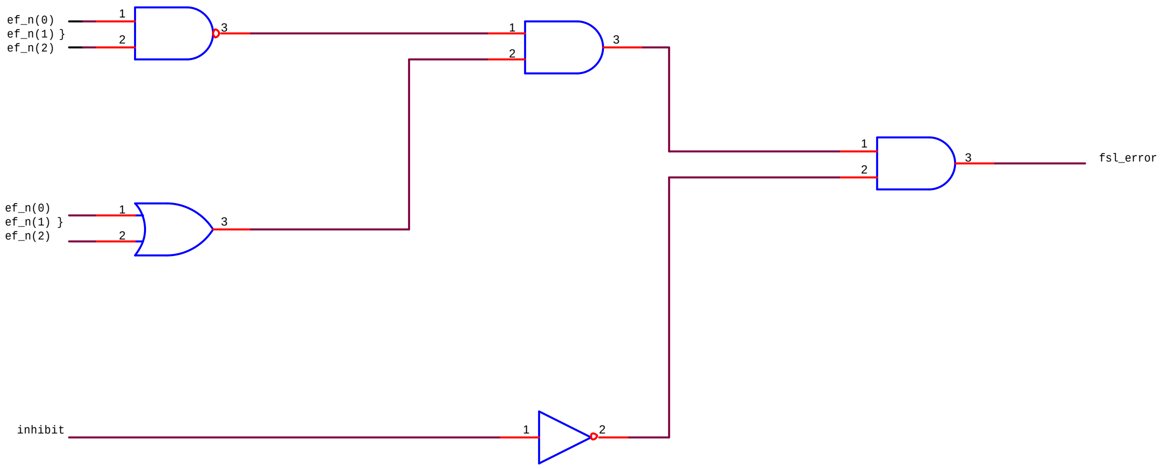


Fig. 12 FIFO error flag generation

In the reset register (see table 28) three FIFO reset are mapped to clear the memories.

11.2.3 Software interface

Address	Register name	Size [bit]	Direction
0x0	FRX_register	32	read

Tab. 20 FSDL registers

The 32 bit (whose 16 bit are for SPIRE) are mapped on the lowest space of the system data bus.

If the software accesses the registers when the FIFO are empty (EF flag asserted to 0 on the status register), the loaded value is not significant.

11.3 Low speed link interface

This command/housekeeping interface is applicable to DCU, MCU and SCU sub-units.

The signals constituting the links are: clock line (LCK), transmitter line (LTX) and receiver line (LRX). LCK and LTX have to be generated by the DPU Payload I/F board and LRX arrives from the sub-units. The LTX and LRX signals change state on the falling edge of the clock signal.

The clock is fixed at 312.5 kHz and it is generated dividing the system clock by 64 (in the FPGA with a divider block).

Two LRX lines have dedicates RS422 receivers and are selected, for the acquisition, by a MUX.

A schematic view of the DPU Payload I/F board-side is the following figure 13.

By means of this interface it is possible to send commands or housekeeping requests. Only one FPGA project allow the implementation of the two configurations (HIFI and SPIRE); JP1 has to be set according to the payload that the board has to interface (see Chapter 10).

All data formats are performed by software. In particular, for what concerns the HK request, the programmer must add 16 high bits LSB to normalise the word in 32 bit format.

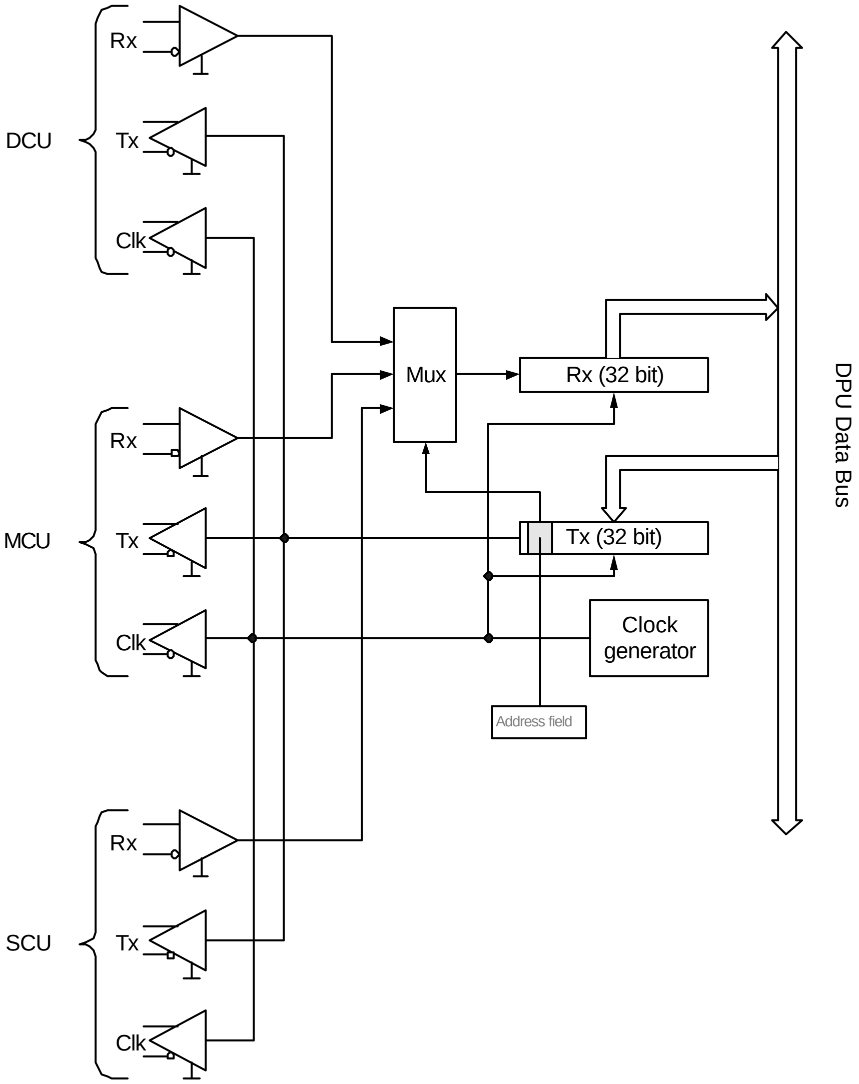


Fig. 13 Low Speed Link Interface

11.3.1 Interface protocol

The command and housekeeping request protocol and timing diagram is shown in figure 14.

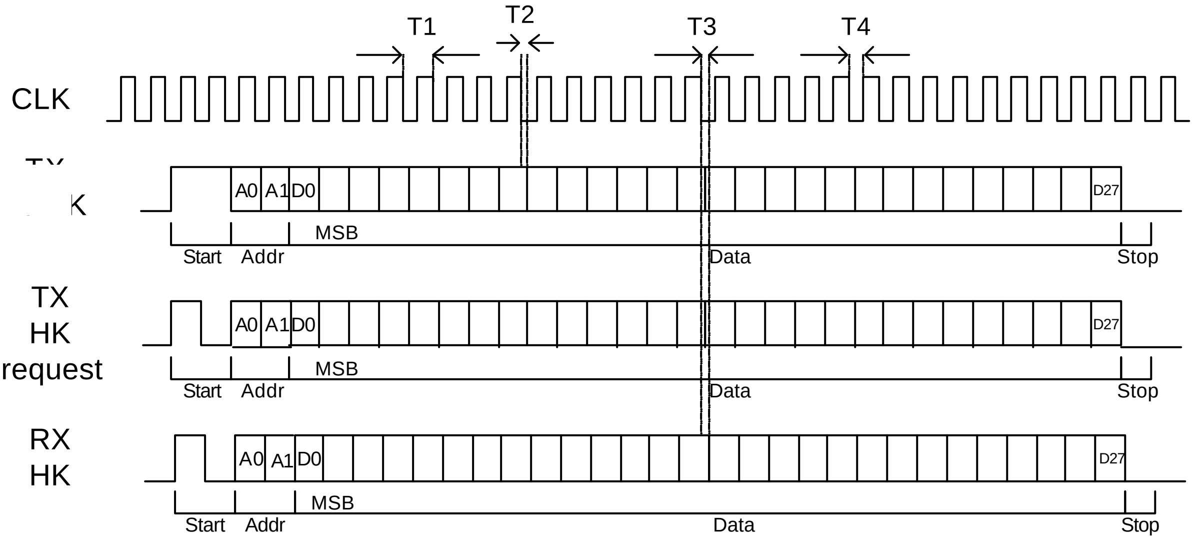


Fig. 14 Low Speed Interface protocol

With reference to the figure, the tolerances on the indicated time parameters are:

Parameter	Min. Value (TBC)	Max. Value (TBC)	Unit
T1	3.17	3.23	µs
T2	-30 nS	0.7	µs
T3	0.05	1.2	µs
T4	1.53	1.66	µs

Commands sent by the DPU via the TX lines are always received by all the subsystems, the address field of the command word selecting the relevant unit. One address is reserved as a broadcast command. The subsystem addressing is made according to the following table:

A0	A1	Subsystem
0	0	DCU
0	1	MCU
1	0	SCU
1	1	Broadcast

The DPU can send both commands and HK requests as signalled by the second start bit of the command word. When requested, the subsystems will send responses via the RX line.

A command is made of 2 start bits, 2 address bits, 28 data bits and 1 stop bit.

A HK request is issued by setting the second start bit according to the following table:

Start0	Start1	HK
1	1	No
1	0	Yes

The HK response will have the same form of the requesting command, the address field indicating the originating subsystem.

The command word data field can be subdivided in every way, the baseline is shown in the following table:

From	To	Description
D0	D7	Command identifier
D8	D27	Command parameters

As the selection of the input RX channel is done by the sub-unit address field of the last TX command sent, no command can be sent to a different sub-unit until the last HK RX corresponding to the last command sent is received. Figure 15 shows the transmission reception sequence.

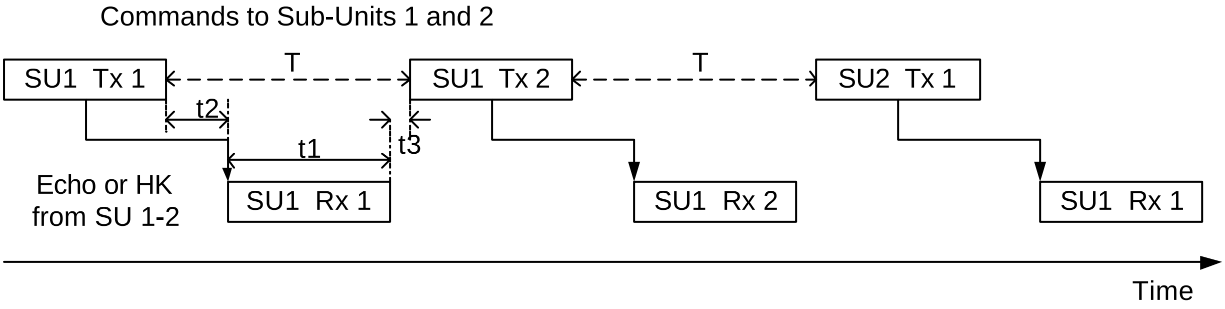


Figure 15 Low Speed Interface transmission-reception protocol

In order to avoid system lock a TBD μs timeout **TO** is defined. In any case, a new HK request cancels the previous request whether already sent or not.

The following table shows the timing requirements with and without HK request.

Parameter	Min value [us] TBC	Max value [us] TBC	Description
TO	TBD	TBD	Time-Out
t1	101	104	Command word length
t2	6.4	500 TBC [REF23]	HK response time
t3	10	NA	Time to next command
T	10	NA	No HK request
T	Minimum value between (t1+t2+t3) and TO		With HK request

A command or housekeeping request shall not start within one clock-cycle after the previous command has terminated or the housekeeping envelope has gone to the inactive state.

11.3.2 Software interface

The software interface is realised with three registers, as mentioned in Table 22 LSL register:

Address	Register name	Size [bit]	Direction
0x0	LTX_register	32	read/write
0x1	LRX_register	32	read
0x2	Lstatus_register	32	read

Table 22 LSL register

The command a/o the housekeeping request have to be written in the LTX register in 32 bit format, only when the "TX busy" flag of the Lstatus register (see table 23) is not asserted. The "TX busy" flag is asserted when the interface is engaged in a command sending.

The echo/housekeeping datum (when the mode-bit of the TX command is low) is latched in the LRX register, then the "data ready" flag is asserted. The LRX register (and the "data ready" flag) shall be HW reset after a CPU read-access of this register.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	HK aborted (HIFI only)	Timeout (HIFI only)	HK pending (HIFI only)	Data ready	TX Busy

Table 23 Lstatus register

The D3, D4 bits of the Lstatus register will be reset after a Lstatus reading cycle; in any case a new writing of the LTX register will reset the D1, D2, D3 bits of the same register.

In the reset register (see table 28) a LSL reset flag clears whole the interface aborting every command sending, parameter waiting and clearing all the Lregisters.

11.4 32 bit bus interface

The bus interface is implemented with buffer and transceiver devices, to match the bus electrical specifications and to be able to receive the following lines:

- 24 address lines
- 32 data lines
- 7 chip select lines (actives low)
- 1 read line (active low)
- 1 write line (active low)
- 1 system clock (20 MHz)
- 1 asynchronous reset line (active low)

The bus interface has to drive:

- 5 interrupt lines
- 32 data lines
- 1 acknowledge line (active high)

The timing of the bus is compliant to the 21020 read and write timing and the number of wait states, associated to each chip select, is reported in sections 7.1.1 and 7.2.1. The 32 bits are mapped in the bits D8-D39 of the DSP data bus.

For what concerns the internal peripherals, the chip select lines, coming from the bus, manage them according to the following table:

\Chip Select	Peripheral
0	FSDL 0
1	FSDL 1
2	FSDL 2
3	Not Used
4	Low Speed Link

5	Board Register
6	Not Used
7	MIL-STD-1553 Dual Port RAM (16KW)
8	MIL-STD-1553 Registers (see Table)
9	Analogue Inputs (see section 11.5)

Table 24 Map of the board into the system bus space

In case of successive accesses to the same memory area (same chip select), the \CS lines remain asserted until a different memory area is addressed; this means that is not mandatory to have chip select signal edges on a bus cycle.

11.5 Analogue inputs

This section provides 8 single-ended, 0÷5V, not conditioned analogue inputs, each one having 10K //?00pF RC filter.

11.5.1 Analogue input interfaces

The analogue input interface is implemented with an ADC chain, composed by the following stages:

- 8 analogue input channels, each one with a 10K and ?00pF RC filter
- a 8-channel analogue multiplexer HS508 (U7, see figure 8) that selects the channel
- an amplifier implemented with a OP27 (U6A, see figure 8) operational amplifier that adapts the signal to the ADC span
- an AD584 (U4, see figure 8) voltage reference that provides a high precision voltage reference to the ADC chain, allowing a reference for possible software compensation of errors (thermal drift, gain variations, ageing, offset and so on) and for calibration verification purposes

- an AD574 (U5, see Figure 8 .), 12 bit, 35µs ADC that receives the start conversion signal from the control stage of the FPGA and, after about 35µs, provides a data ready signal to the FPGA
- a 12-bit latch, implemented on the FPGA, that stores the converted value until the CPU reading

The input analogue parameters coming from the DC/DC board are specified in table 25:

Channel	Parameter
0	Spare
1	+5V Monitor
2	+15V Monitor
3	-15V Monitor
4	Temperature Monitor
5	Not Used
6	Not Used
7	Voltage Reference

Table 25 Analog Inputs

See section 2.4 for the relevant outputs scales of the analogue channels.

11.5.2 Software interface

The analogue input stage is addressed by a dedicated chip select line of the system bus. The set of registers are shown in table 26:

Address	Register name	Size [bit]	Direction
0x8000	MSEL_reg	3	read/write
0x8001	MDATA_reg	13	read

Table 26 Analog input register

The MDATA_reg contains the 12-bit converted data and a “ready” flag (MSB) that indicates when the data is ready (active high).

By writing a multiplexer selection in the MSEL_reg, the multiplexer shall select the line to be converted and the control block shall manage the conversion, after a delay of about 15µs, for the stabilization of the signal. When the conversion is finished (about 35µs conversion time), the “ready” bit of the MDATA_reg will be asserted and the converted value will be present in this register. After the reading, the MDATA_reg shall be reset by the control block of the FPGA.

The analog input #7 is reserved: a half scale voltage reference is connected to this input to provide a reference for the calibration phase.

11.6 Control logic

The control logic is implemented in the FPGA; it manages all the peripherals and realizes the interface between the system bus and the peripherals.

11.6.1 Board register

Address	Register name	Size [bit]	Direction
0x0	Reset register	32	read/write
0x1	Status register	32	read

Table 27 Board registers

The reset register allows single peripheral resets (except for the control logic) and the global software reset (see xxx). Each flag must be asserted (high) to generate a reset and it must be de-asserted (low) by another writing.

These flags are mapped according to the following table:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
SW reset	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	LSL reset	MIL1553 Reset	FSDL3 reset	FSDL2 reset	FSDL1 reset	FSDL0 reset

Table 28 Reset register

The Status register reports all the status of the peripherals; the following table shows the map of the register:

D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	LSL reset	1553 \INCMD	Ferr3	Ferr2	Ferr1	Ferr0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
ACQ3	FF3	HF3	EF3	ACQ2	FF2	HF2	EF2	ACQ1	FF1	HF1	EF1	ACQ0	FF0	HF0	EF0

Table 29 Status register (FIFO 3 not used)

The FIFO status is as indicated by the actual HW FIFOs, i.e. the status is in inverted logic. For instance, the status of the bits of the FIFO #0 have the following meaning:

D2	**D1**	**D0**	**DECIMAL VALUE**	**FIFO STATUS**
FF0	**HF0**	**EF0**		
1	1	0	6	FIFO EMPTY (0 words)
1	1	1	7	1 <= # of FIFO words < 2047
1	0	1	5	FIFO HALF FULL (2048<= # of FIFO words < 4095)
0	0	1	1	FIFO FULL (4096 words)

11.7 Board reset

Three possible actions can globally reset the board:

1. Power-on reset
2. Hardware reset with the system reset line (\SYSRESETIN) driven by the CPU
3. Software reset, asserting the global bit in the reset register

When the CPU asserts the reset line (hardware reset) the whole board is reset (also the blocks of the FPGA control logic) whereas a software reset causes the reset of all the devices except for the control logic block.