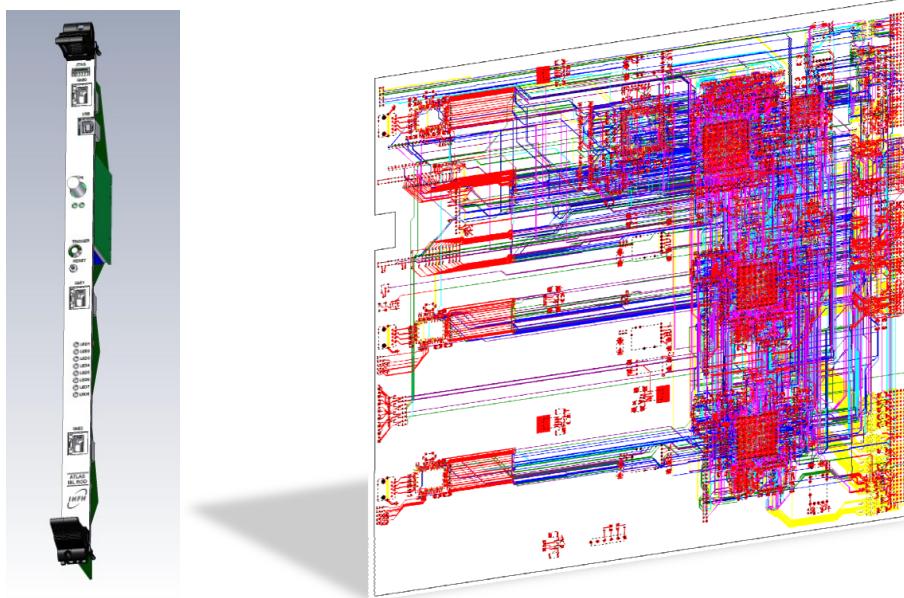




IBL ROD board revD reference manual



Gabriele Balbi, **Davide Falchieri**, Alessandro Gabrielli,
Luca Lama, Riccardo Travaglini, Samuele Zannoli

Department of Physics and INFN Bologna

November 2012

Version management

Number	Date	Changes	Author
1	22/11/2012	Initial version	Davide Falchieri
2	06/06/2013	Added script to create STAPL files from BIT	Riccardo Travaglini
3	04/02/2014	Extended booting embedded processors paragraph and added “How to reset the ROD” paragraph	Davide Falchieri
4	22/07/2014	Adding paragraph “Programming FPGAs via ACE files”	Davide Falchieri

Contents

IBL ROD : what is it ?	4
IBL ROD main features	6
Main modifications from revB to revC	8
Main modifications from revC to revD	10
Known HW bugs and limitations	11
PCB stackup	12
FPGA JTAG programming	23
FPGA VME programming	25
Booting embedded processors	29
How to reset the ROD	30
MDSP JTAG programming	31
Clock distribution	32
Power supply distribution	36
BOC-ROD interface signals	40
ROD bus	43
PRM pin assignment (UCF file)	45
S6A-S6B pin assignment (UCF file)	52
Virtex5 pin assignment (UCF file)	63
Main components in the BOM	75
References	77

IBL ROD : what is it ?

IBL ROD board is meant to be the upgrade of the ATLAS Silicon Read Out Driver (SiROD) [1], that is currently used in the ATLAS Off-Detector electronics sub-system in order to interface with Silicon Tracker (SCT) and Pixel Front End Detector modules.

The goal of the IBL ROD board is to readout and process data coming from the Insertable B-Layer, a new layer of pixels that is going to be inserted as the innermost pixel layer of the ATLAS detector during the 2013 long shutdown.

The IBL ROD board is going to work together with a new BOC board [2], that is the upgrade of the BOC board working with the SiROD.

The whole IBL readout requires 14 BOC-ROD pairs hosted in a single VME crate, where a TIM [3] board distributes trigger information to all the boards.

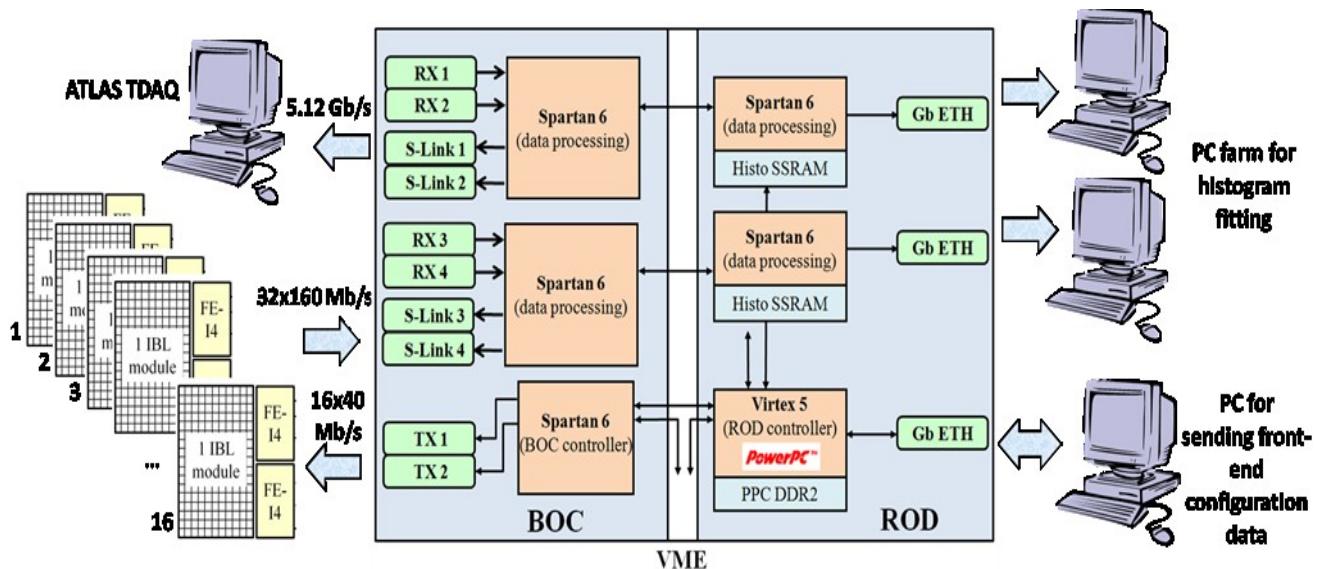


Figure 1: BOC-ROD toplevel diagram

Each BOC-ROD pair is able to readout data coming from 16 IBL modules (= 32 FE-I4 front-end chips) for a total input bandwidth of $32 \times 160 \text{ Mb/s} = 5.12 \text{ Gb/s}$.

IBL modules data are received from the BOC board via the RX optical modules, then 8B/10B decoding is performed before passing data to the ROD, where data processing happens:

- data gathering and event fragment building during physics runs,
- histogramming during calibration runs

During physics runs events to be sent to the ATLAS TDAQ are sent back to the BOC, where 4 S-Link modules are implemented for a total output bandwidth of 5.12 Gb/s.

16 40 Mb/s links allow to send configuration data and triggers to the IBL modules.

The ROD board hosts 3 Gb Ethernet interfaces:

- 1 of them (connected to the PowerPC) allows to receive front-end configuration data from an external PC.

- 2 of them allow to send calibration histograms toward an external PC farm.

IBL ROD main features

- IBL ROD is a 14-layer 9U x 400 mm VME64x board.
- It contains:
 - 1 Digital Signal Processor (MDSP):
 - 1 **Texas Instruments TMS320C6201-GJC200**: it is the same DSP device as in the SiROD board.
 - **32 MByte SDRAM DDR**
 - **4 Mbit FLASH SST39VF040-70-4C-NH**
 - 3 types of FPGA devices:
 - 1 **Xilinx Spartan6 XC6SLX45-FGG484**: it is the new PRM device interfacing with the VME bus, the MDSP and the ROD Controller FPGA.
 - 1 **Xilinx Virtex5 XC5VFX70T-FF1136**: it is the new ROD Controller FPGA. It also hosts as an internal HW core a PowerPC device, whose performances will be compared to the MDSP.
 - **2 GByte DDR2 SODIMM**
 - **64 Mbit FLASH Atmel AT45DB642D**
 - 2 **Xilinx Spartan6 XC6SLX150-FGG900**: they implement the ROD data path (data gathering, event fragment building, histogramming). Each of these 2 Spartan6 devices features:
 - **one 2-Gbit DDR2 chip (Micor MT47H128M16RT-25E)**,
 - **two 512Kx36 SRAM chips (Cypress CY7C1370D-250AXC-ND)**.
 - 3 Gbit Ethernet interfaces with PHY DP83865.
- The ROD front panel features:
 - 1 TTCrq mezzanine
 - 1 USB mezzanine (UM232R USB)
 - 1 JTAG connector (J9) for FPGA programming
 - 1 LEMO trigger input (J4)
 - 1 reset push button (SW1)
 - 3 Gbit Ethernet interfaces
 - 8 status LEDs



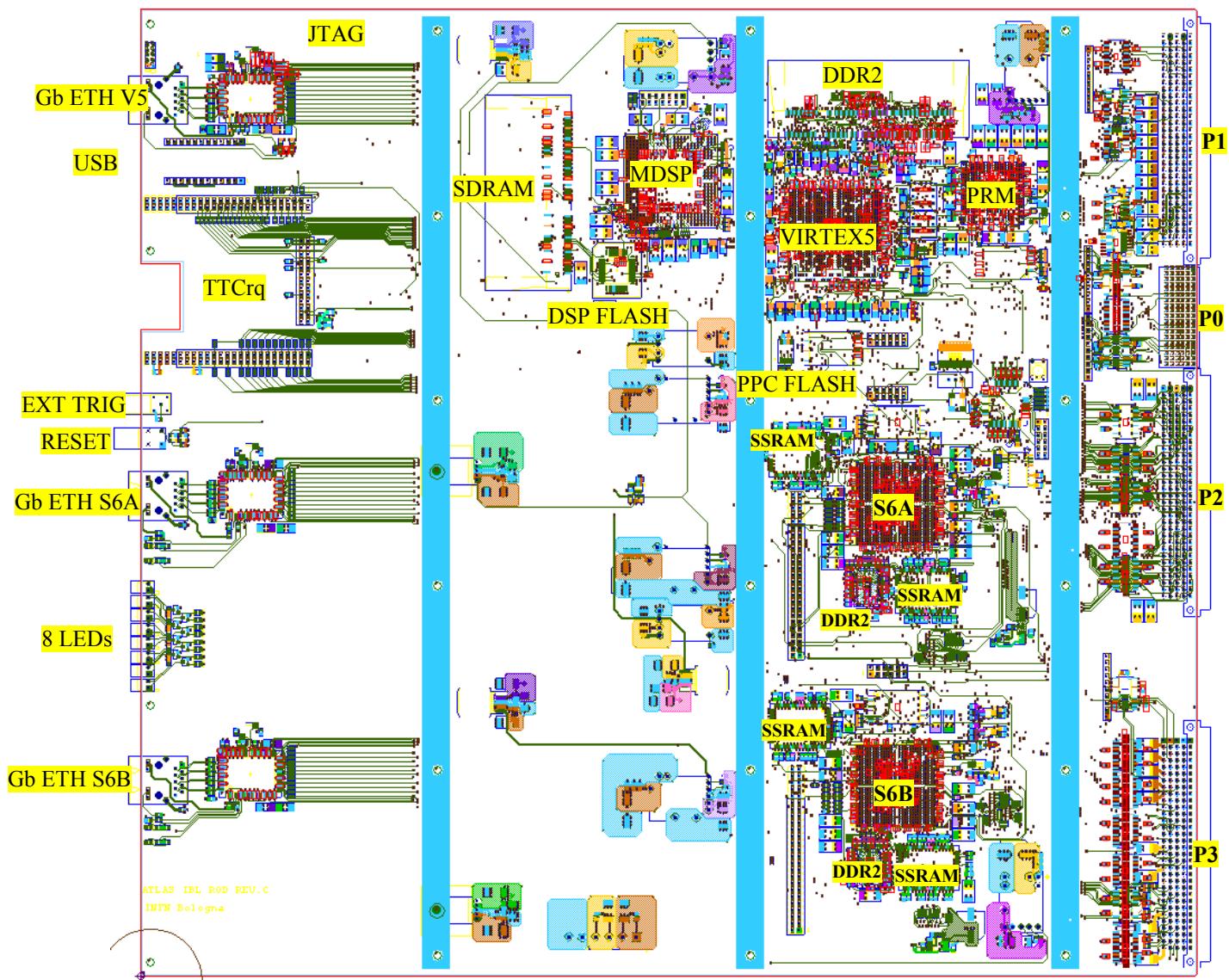
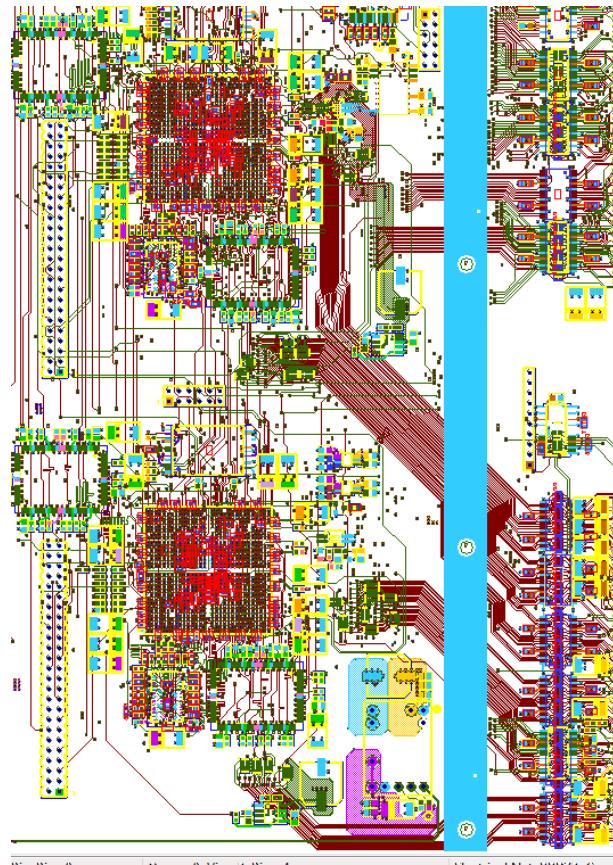


Figure 2: IBL ROD revD main devices and external interfaces

Main modifications from revB to revC

Here is a list of the main modifications that have been introduced on IBL ROD revC in order to correct some problems affecting the revB boards:

- **revB:** no termination resistors on SSTL3 lines from BOC to ROD (RD[95:0]) are present.
- **revC:** 50 Ohm termination resistors to VTT=1.5V on SSTL3 lines from BOC to ROD (RD[95:0]) have been added.



- **revB:** RJ45 Ethernet connectors P2, P3 and P4 each have 2 LEDS which never blink.
- **revC:** Proper connection on pins 15, 16, 17 and 18 on RJ45 Ethernet connectors P2, P3 and P4, in order to properly connect the LEDS: one is on when the LINK1000 (1Gb) connection is established, the other blinks when there is some activity on the line.
- **revB:** The following pairs of signals are shorted with each other: DDR2_CKE0-DDR2_CKE1, DDR2_CS0-DDR2_CS1, DDR2_ODT0-DDR2_ODT1. This implies that only single ranks SODIMM modules can be used or, in case of dual ranks, only half of the memory can be actually used.

- **revC:** The pairs of signals DDR2_CKE0-DDR2_CKE1, DDR2_CS0-DDR2_CS1, DDR2_ODT0-DDR2_ODT1 are divided and routed to different pins. DDR2_CKE1 is connected to pin AD30 of U74, DDR2_CS1 is connected to pin J29 of U74 and DDR2_ODT1 is connected to pin E31 of U74 between the Virtex5 and the SODIMM. 5 new 0603 resistors are added: R807, R808, R809, R810 e R811.
- **revB:** the system monitor on the Virtex5 is not working.
- **revC:** in order to have a working system monitor, AVDD (pin T18 of U74) has been connected to 2V5_S6A (it was connected to 3.3V), while VREFP (pin V18 of U74) has been connected to 2V5_S6A (it was connected to GND).
- **revB:** the ROD_BUS_BOE_BUSY signal from BOE to ROD is generated as a LVCMOS33 signal on the BOE BCF, while it is received on a 2.5V bank on the ROD Virtex5.
- **revC:** A 3-resistor voltage divider has been used in order to decrease the voltage from 3.3V to 2.5V. The signal ROD_BUS_BOE_BUSY25 coming out of the voltage divider (3 resistors: R816-R817-R818) goes to pin AB10 of U74.
- **revB:** the JTAG connector for the Lattice chip (J8) is not accessible when the board is inserted in a VME crate.
- **revC:** The JTAG connector J8 is expanded from 1x6 to 2x6 pins. Four of them drive the Lattice JTAG port as before, other four pins are connected to a 3.3V bank on the PRM FPGA U60: AB14 (lattice_tdi), AB15 (lattice_tms), AB17 (lattice_tck) e AA16 (lattice_tdo). The other two pins are connected to GND and 3.3V. This allows to access the Lattice JTAG port from the front panel JTAG port or from the VME, if a proper firmware is loaded on the PRM.
- **revB:** The main JTAG connector (J9) is difficult to access, due to its position.
- **revC:** The main JTAG connector (J9) has been moved closer to the edge of the board, in order to be accessible on the front panel. **Be careful to the pins polarity, since the J9 signals have been swapped left-right from revB to revC !**
- **revB:** The MDSP power supply filter FL1 is mounted on the bottom side, but is mounted in vertical position (5 mm) and it is easily removed/hurled when handling the board..
- **revC:** The FL1 filter has been positioned on the TOP side close to the MDSP.

IMPORTANT NOTE:

The UCF files for revB and revC boards are almost 100% compatible. This means that a revB UCF file can be safely used on a revC board.

The other way round could not be true due to a couple of very tiny differences:

- One difference is that revC board features an extra pin (D8, active low) which can be used to shutdown the VTT=1.5V used to pullup the SSTL3 signals RD[95:0] coming from the BOC, if needed.
- The other difference is that revC board features three extra pins (DDR2_CKE1 connected to pin AD30 of U74, DDR2_CS1 is connected to pin J29 of U74 and DDR2_ODT1 is connected to pin E31 of U74) which can be used for the correct behavior of dual-rank SODIMM modules.

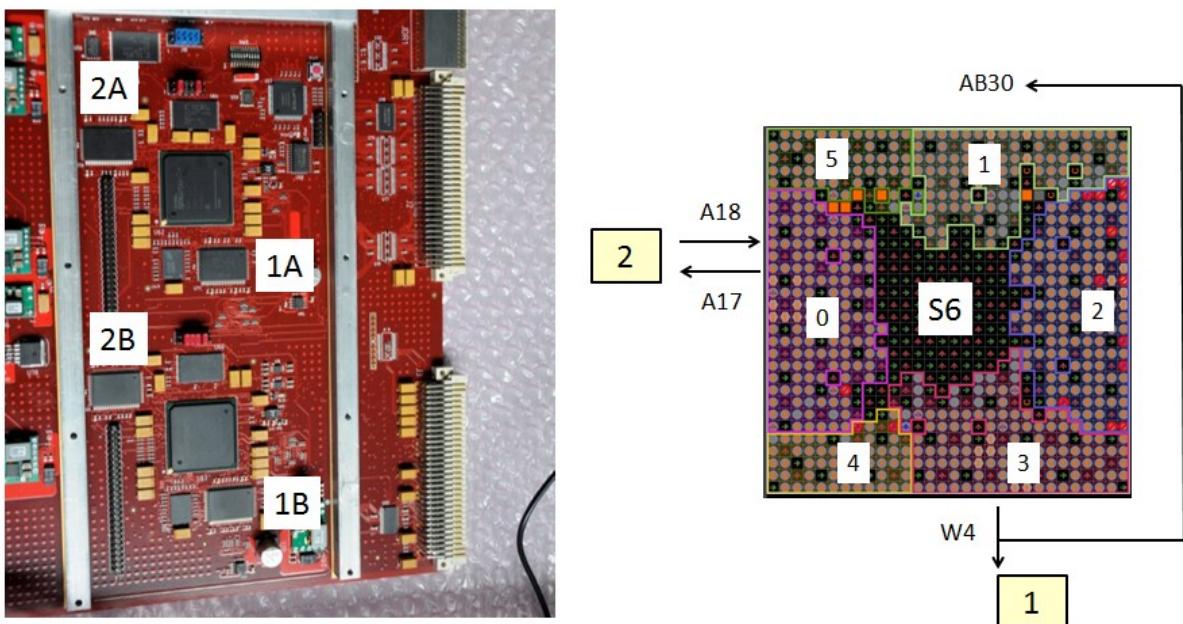
Main modifications from revC to revD

The only difference from revC and revD RODs is that revD boards mount Spartan6 with faster -3 speed grade (revC boards mount devices with -2 speed grade).

Known HW bugs and limitations

Here is a list of known bugs and limitations of IBL ROD revD that might be removed / corrected in the next board revisions:

- front end emulator connector:
The connector has been designed for single ended signal and not for LVDS signals, as it should have been. In this way LVDS pairs are placed in locations far apart to each other on the connector, leading to a more difficult connection to the FeI4 boards.
- The feedback clock SSR1B_CLK path on the board can be improved, since now the path from W4 to the SSRAM and then back to AB30 is not correctly balanced.



- Take in consideration the multi-boot option for the PRM by using a XCF32P PROM, instead of the XCF16P PROM being used in revA, revB and revC ?

PCB stackup

The PCB stackup is built on 14 layers, as shown in Fig. 3.

1. TOP
- 2. GND**
3. SIGNAL ($\uparrow\downarrow$)
4. SIGNAL (\leftrightarrow)
5. VCC
6. SIGNAL ($\uparrow\downarrow$)
- 7. GND**
8. VCC
9. SIGNAL (\leftrightarrow)
10. VCC
11. SIGNAL ($\uparrow\downarrow$)
12. SIGNAL (\leftrightarrow)
- 13. GND**
14. BOTTOM

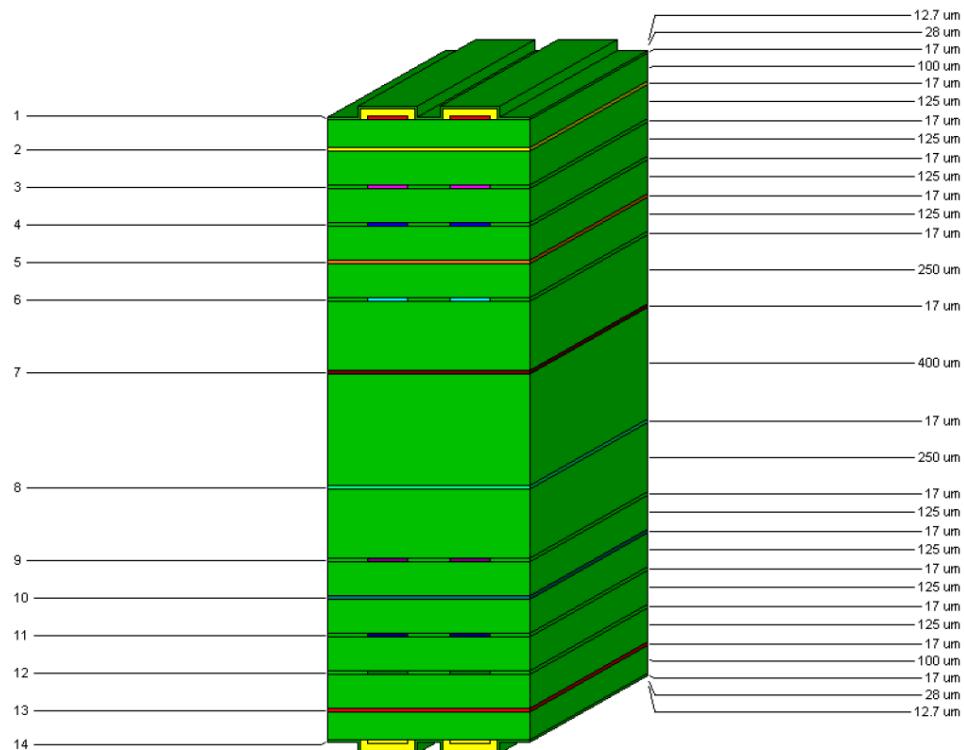


Figure 3: ROD PCB stackup

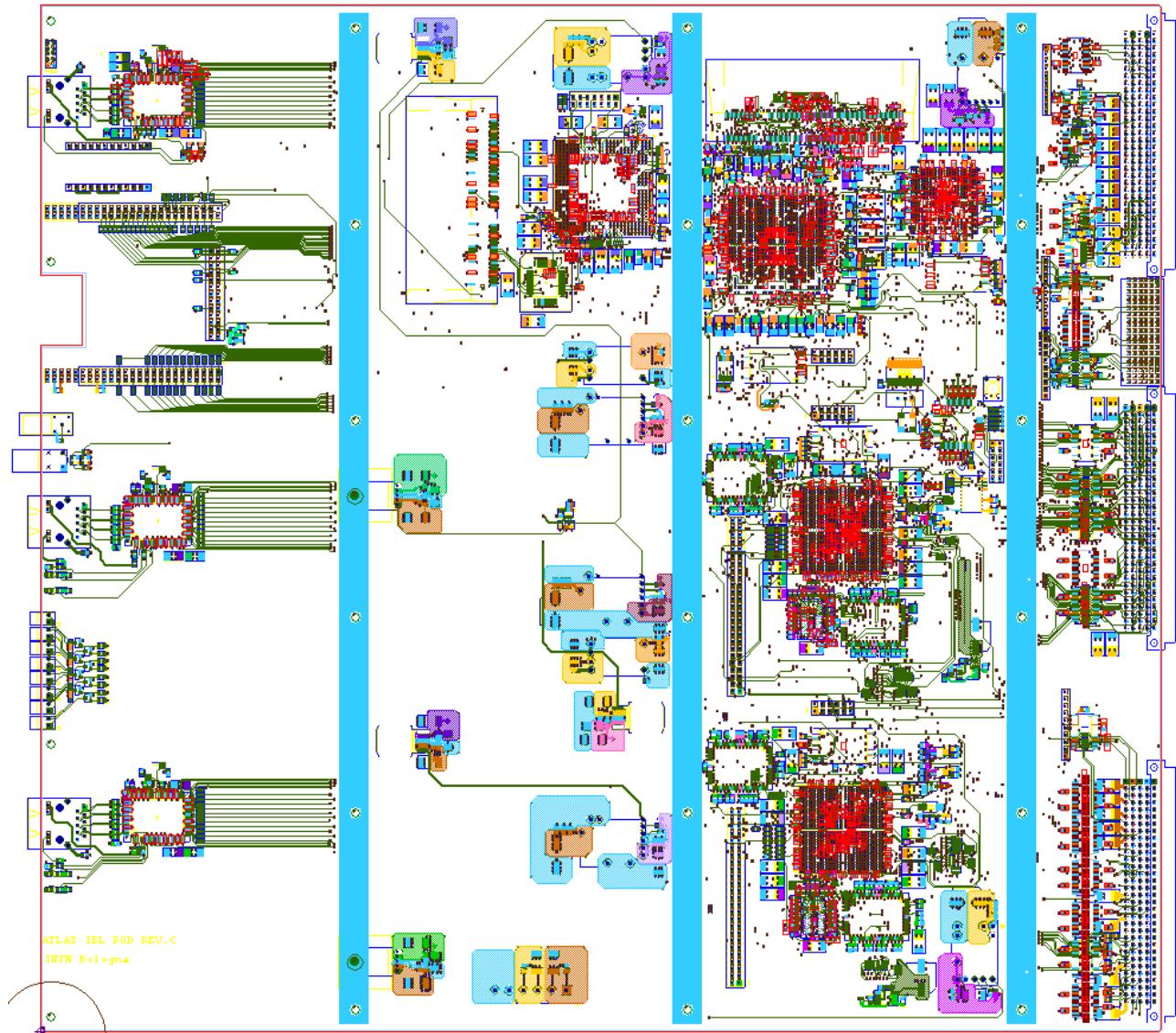


Figure 4: IBL ROD final layout (layer 1 - top layer)

<input checked="" type="checkbox"/>	0V9_VTTREF_CHIPS
<input checked="" type="checkbox"/>	0V9_VTTREF_SODIMM
<input checked="" type="checkbox"/>	0V9_VTT_CHIPS
<input checked="" type="checkbox"/>	0V9_VTT_SODIMM
<input checked="" type="checkbox"/>	1V0
<input checked="" type="checkbox"/>	1V2_PRM
<input checked="" type="checkbox"/>	1V2_S6A
<input checked="" type="checkbox"/>	1V2_S6B
<input checked="" type="checkbox"/>	1V5_SSTL3_VREF_S6A
<input checked="" type="checkbox"/>	1V5_SSTL3_VREF_S6B
<input checked="" type="checkbox"/>	1V8_DDR2_CHIPS
<input checked="" type="checkbox"/>	1V8_ETH_S6
<input checked="" type="checkbox"/>	1V8_MDSP_ETH_V5
<input checked="" type="checkbox"/>	1V8_SODIMM
<input checked="" type="checkbox"/>	2.5V_QPLL
<input checked="" type="checkbox"/>	2V5_S6A
<input checked="" type="checkbox"/>	2V5_S6B
<input checked="" type="checkbox"/>	3.3VME
<input checked="" type="checkbox"/>	3V3_A
<input checked="" type="checkbox"/>	3V3_B
<input checked="" type="checkbox"/>	3V3_VDD_SSRAM
<input checked="" type="checkbox"/>	5.0VME

Power supply distribution
color scheme

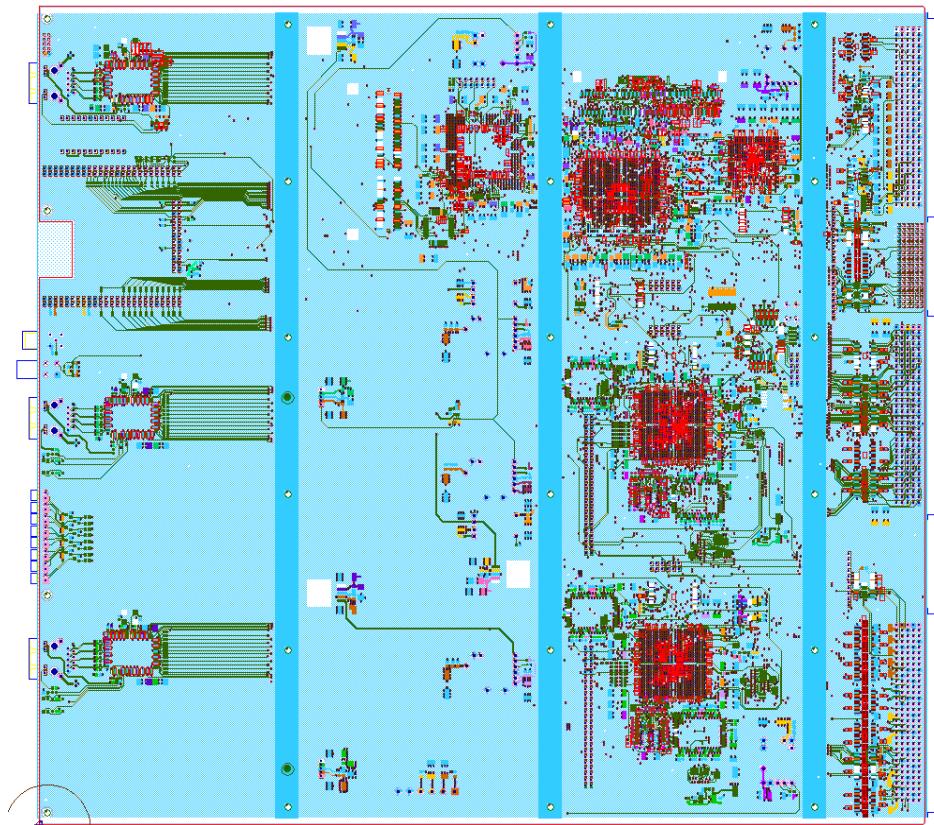


Figure 5: layer 2 (GND)

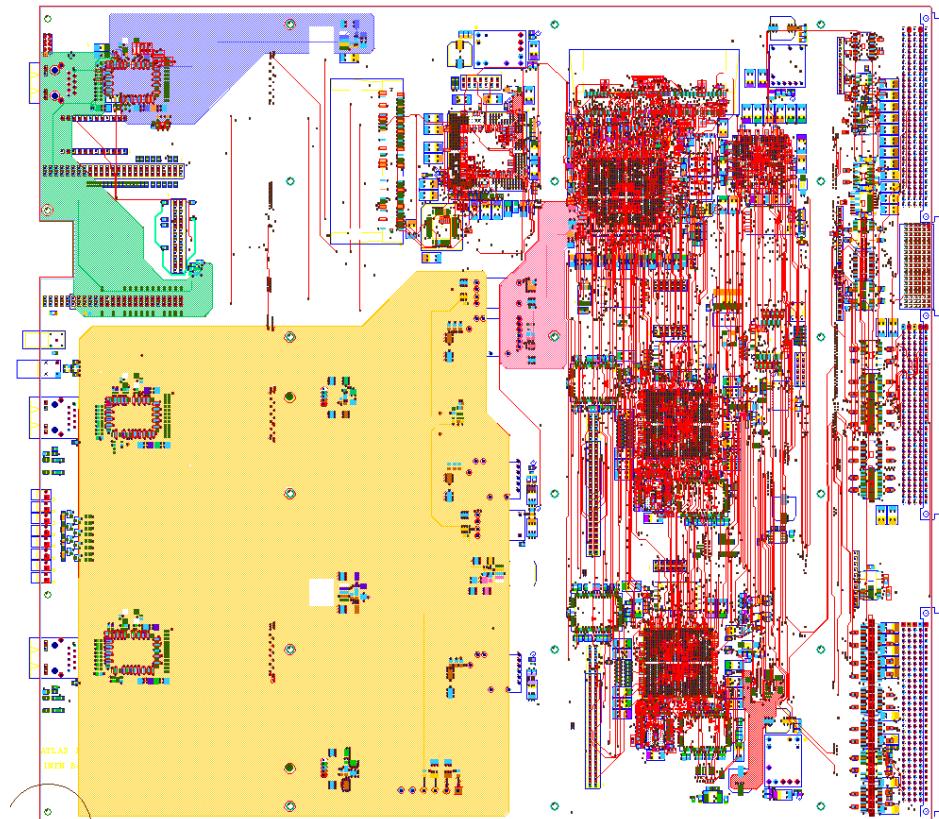


Figure 6: layer 3

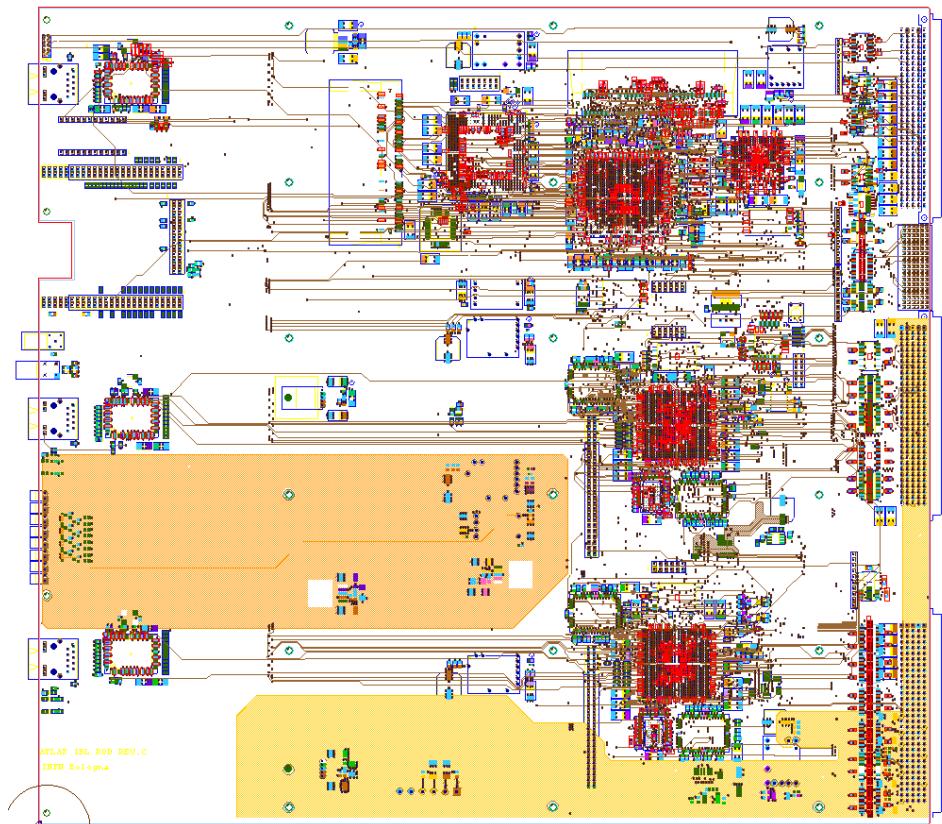


Figure 7: layer 4

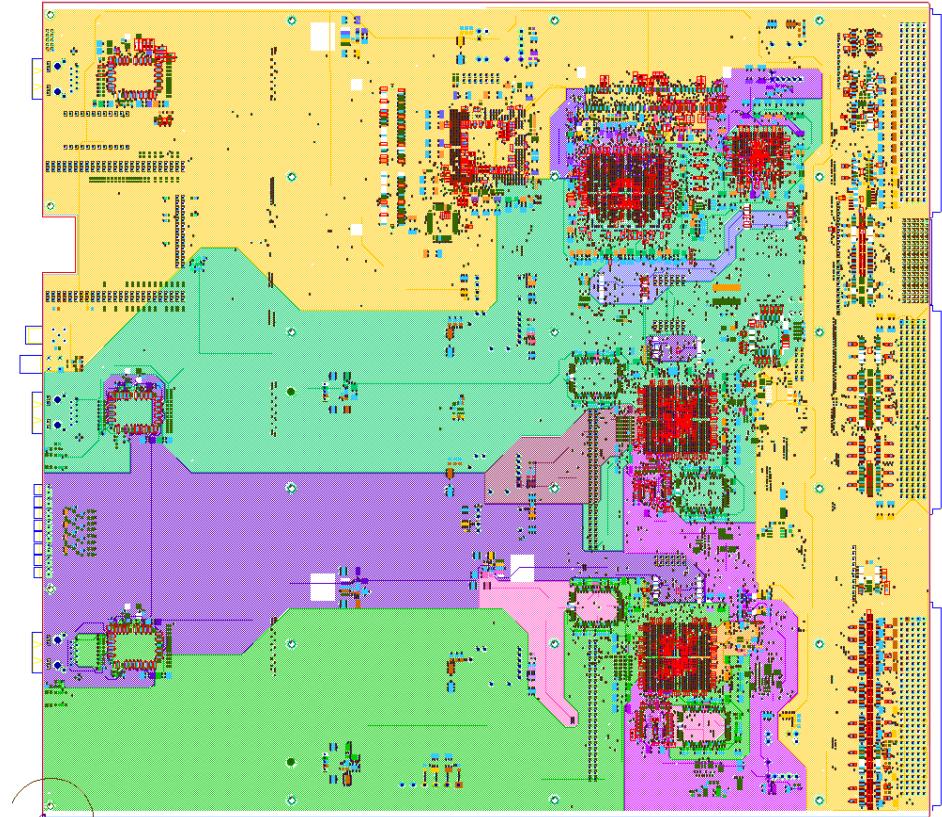


Figure 8: layer 5

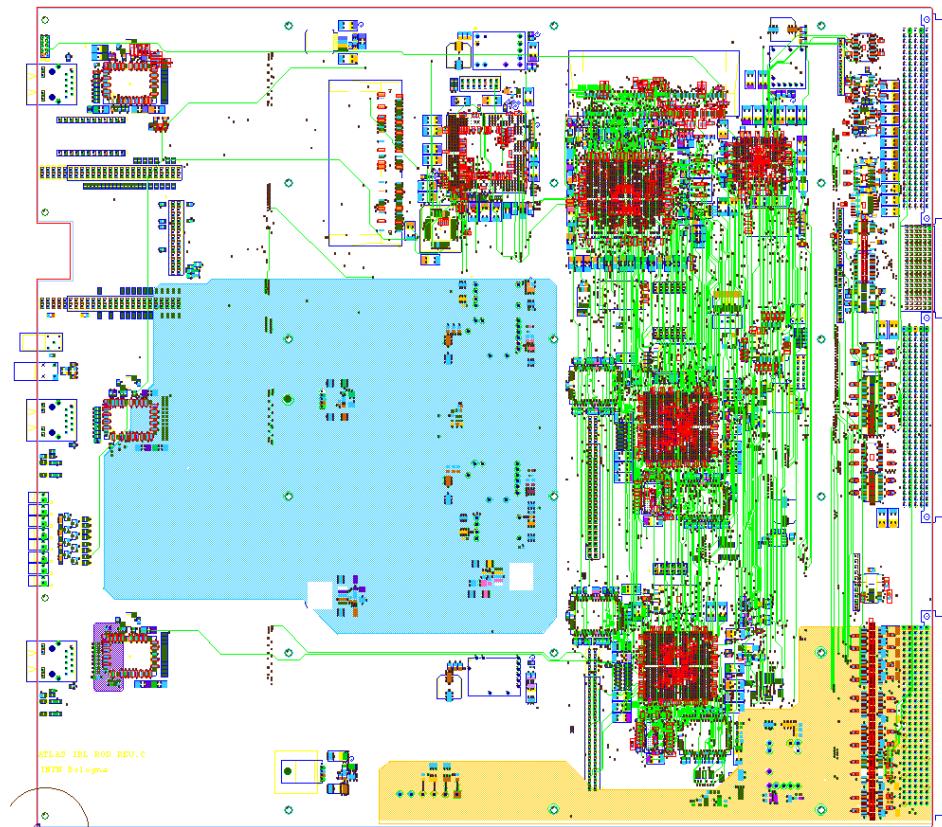


Figure 9: layer 6

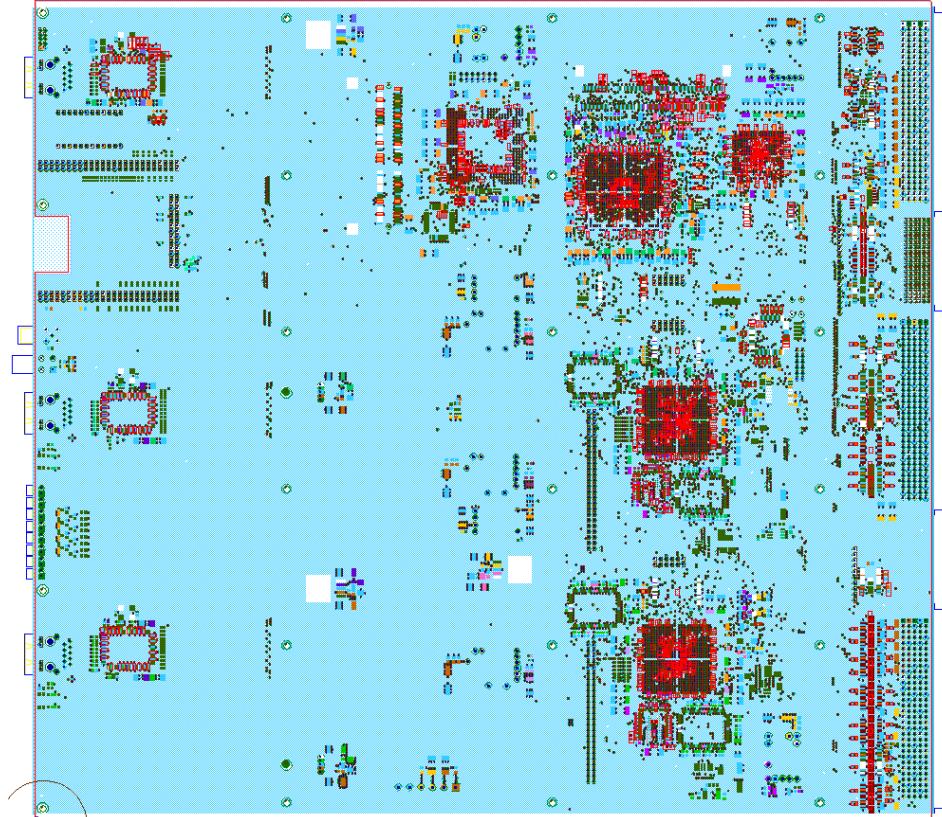


Figure 10: layer 7 (GND)

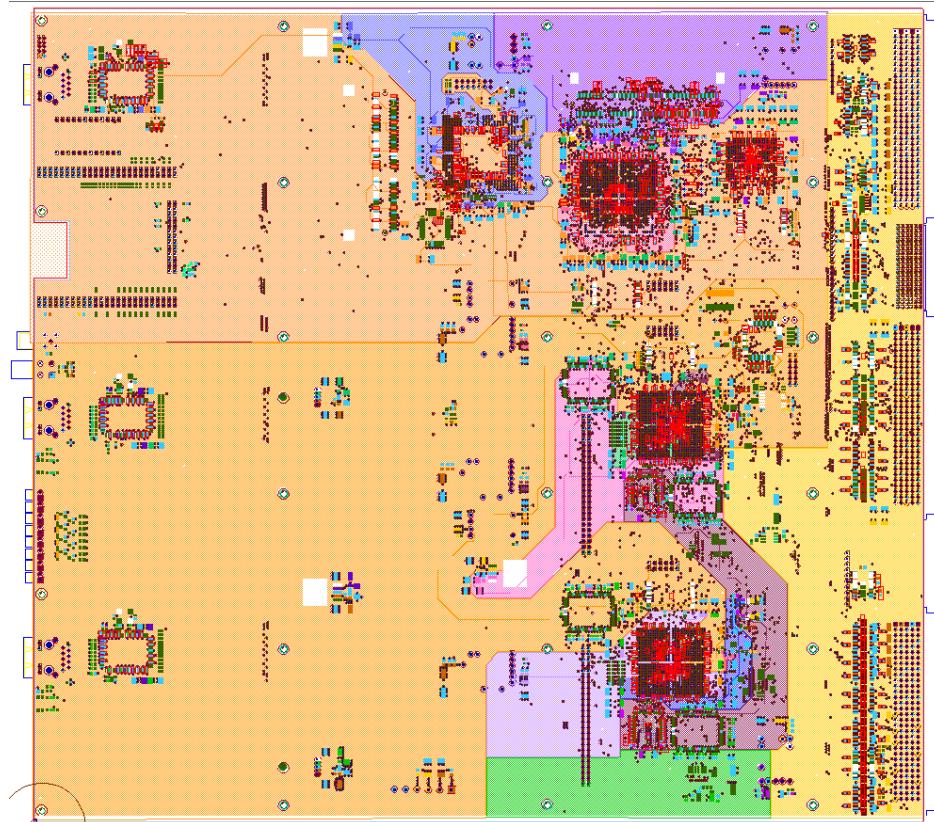


Figure 11: layer 8

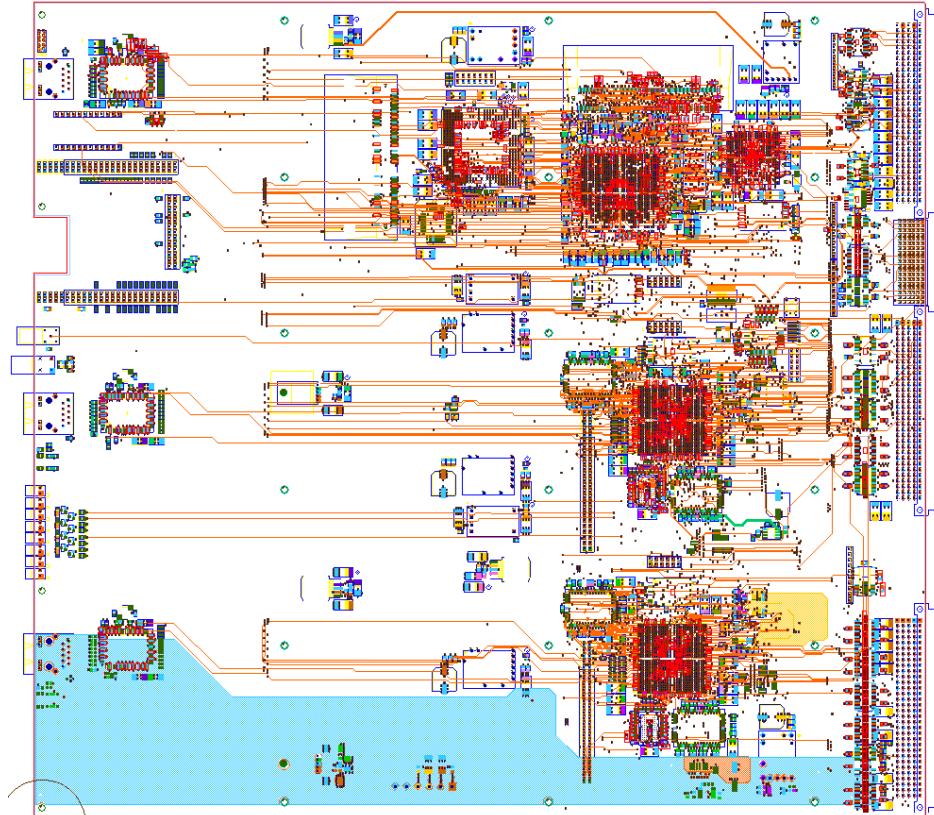


Figure 12: layer 9

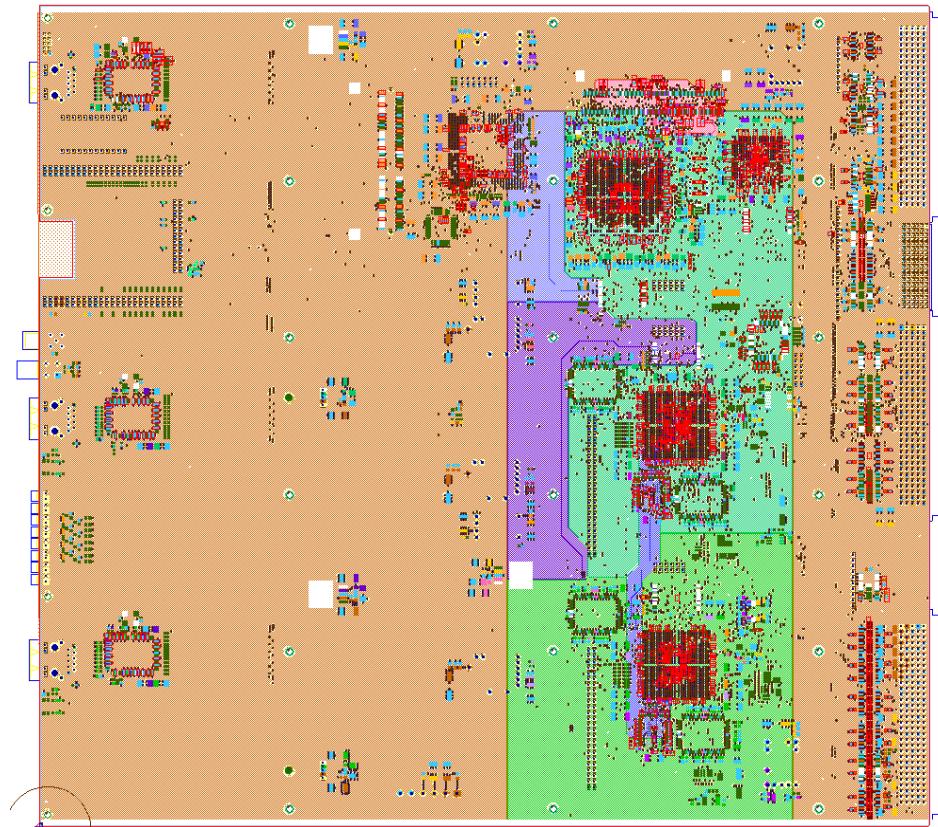


Figure 13: layer 10

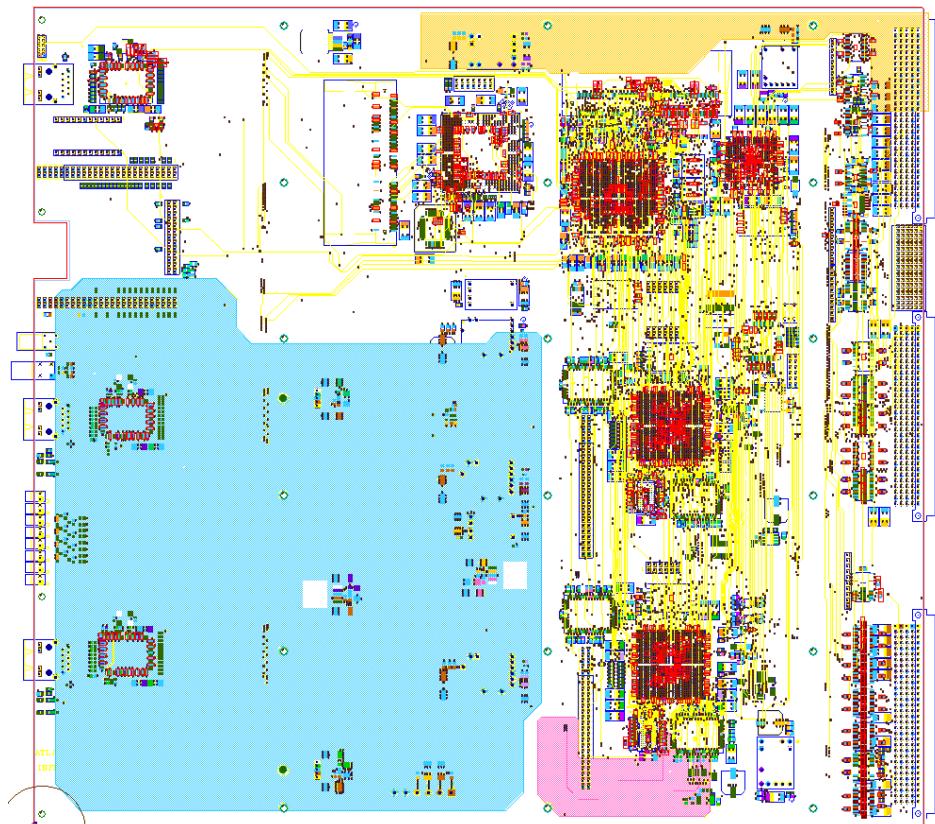


Figure 14: layer 11

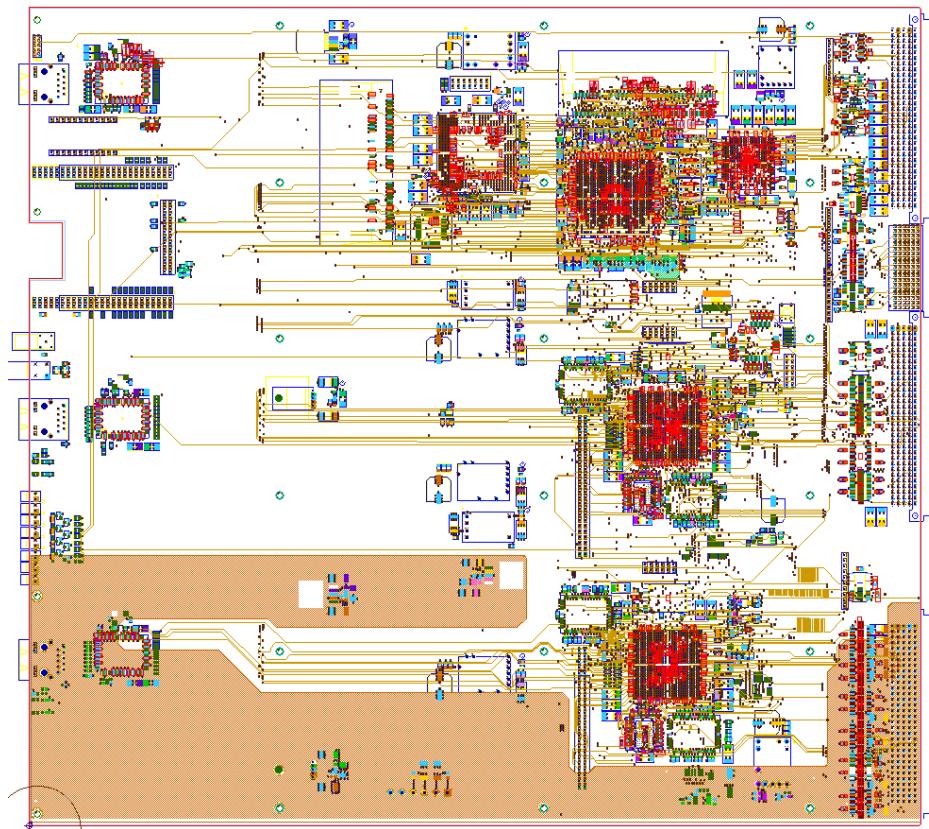


Figure 15: layer 12

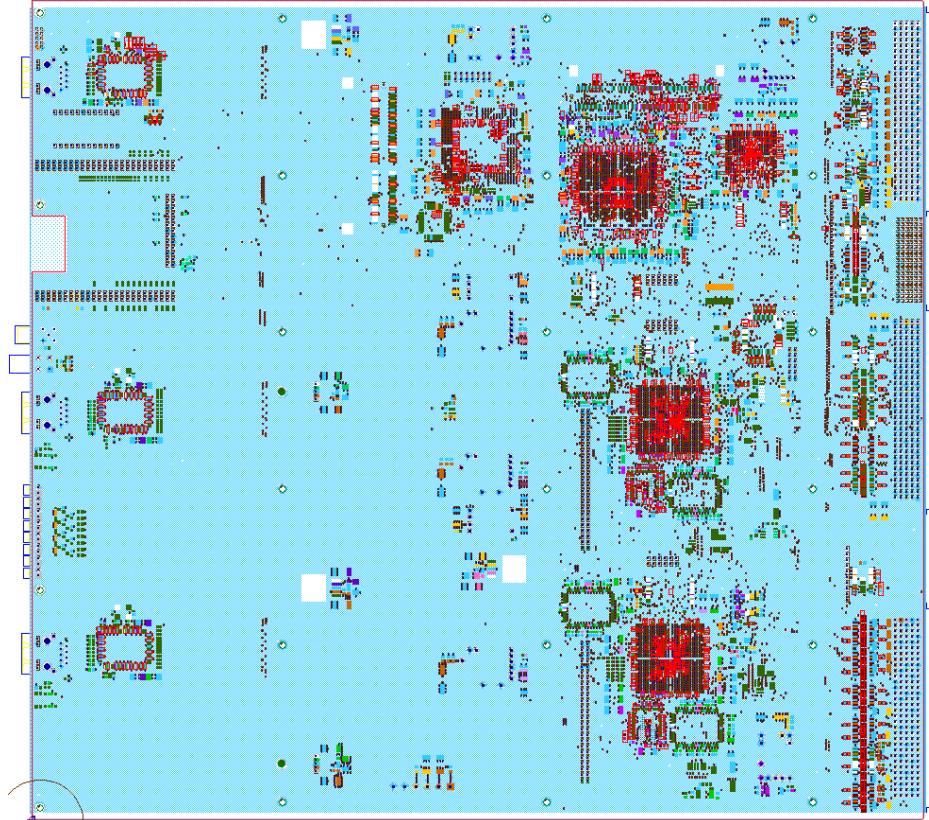


Figure 16: layer 13 (GND)

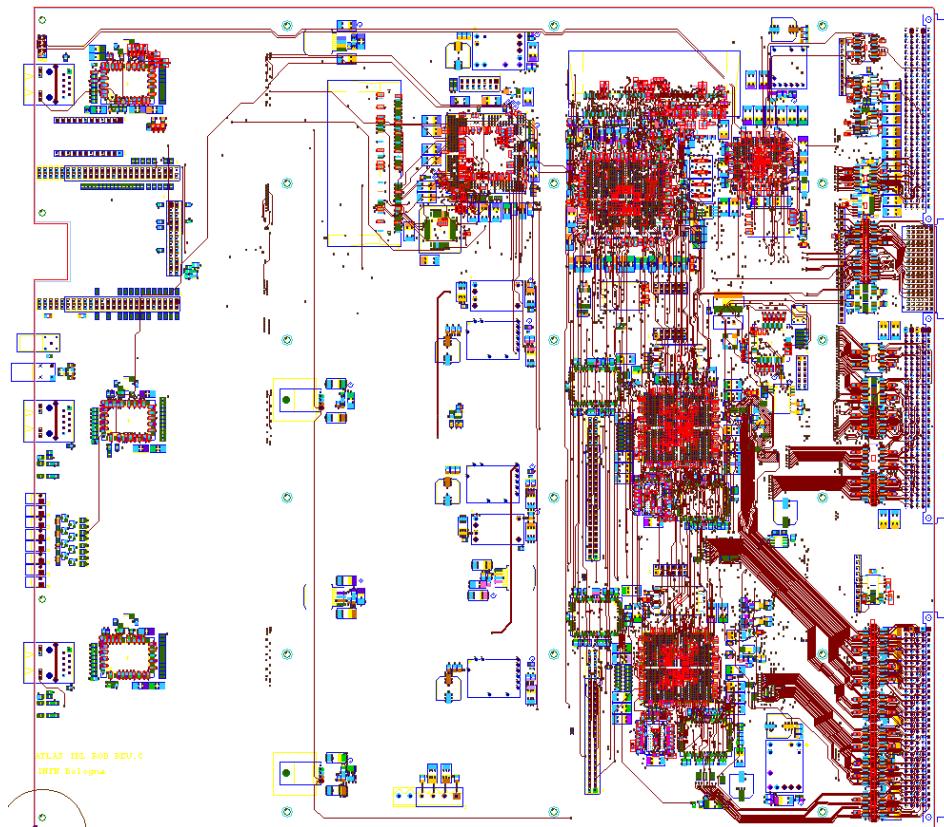


Figure 17: layer 14 (bottom layer)

FPGA JTAG programming

How to program ROD FPGAs firmware:

- Each FPGA has its own JTAG connector on which the Xilinx programmer can be plugged:
 - Virtex5: **J18**
 - S6A: **J11**
 - S6B: **J12**
 - PRM: **J9**
- The Xilinx programmer flying leads shall be plugged on the odd number pins:
 - *tdi*: pin 9
 - *tdo*: pin 7
 - *tck*: pin 5
 - *tms*: pin 3

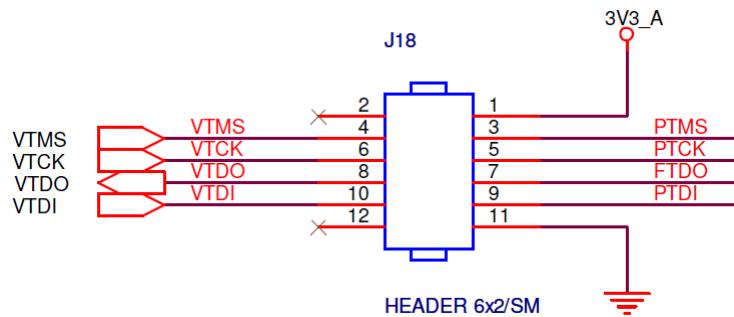


Figure 18: Virtex5 JTAG connector J18 (the same holds for J11 and J12)

- In order to program the PRM, the J9 connector has to be used, (note that the ROD comes with an ad-hoc cable Fig. 20):

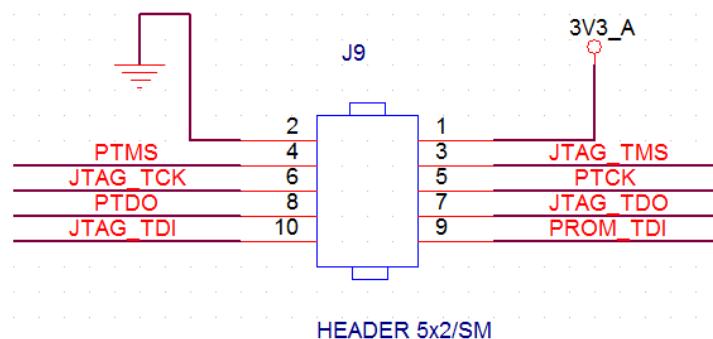


Figure 19: PRM JTAG connector J9

- *tdi*: pin 9
- *tdo*: pin 8
- *tck*: pin 5
- *tms*: pin 4

- It can be useful to have a single JTAG chain with all the main FPGAs (S6A, S6B, Virtex5). This can be arranged by programming the PRM with a special firmware and connecting the Xilinx programmer flying leads on the following pins of J9:
 - *tdi*: pin 10
 - *tdo*: pin 7
 - *tck*: pin 6
 - *tms*: pin 3
- You are invited to use our jtag connector that can be plugged on the Xilinx programmer. It has two different pin-outs, the upper one, used in Fig. 20, connects to the prm jtag chain; the other one connects to the V5-S6B-S6A chain.

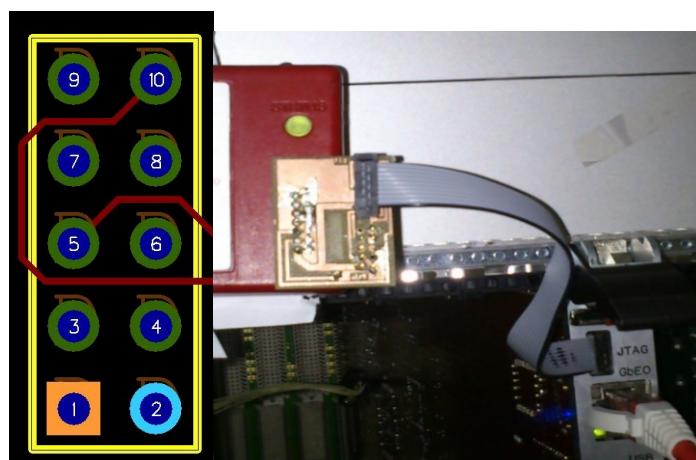


Figure 20: PRM JTAG connector J9 as viewed from the front panel

In this case jumpers have to be placed on the following pins on connectors J11, J12 and J18:

- pin 3 → pin 4
- pin 5 → pin 6
- pin 7 → pin 8
- pin 9 → pin 10

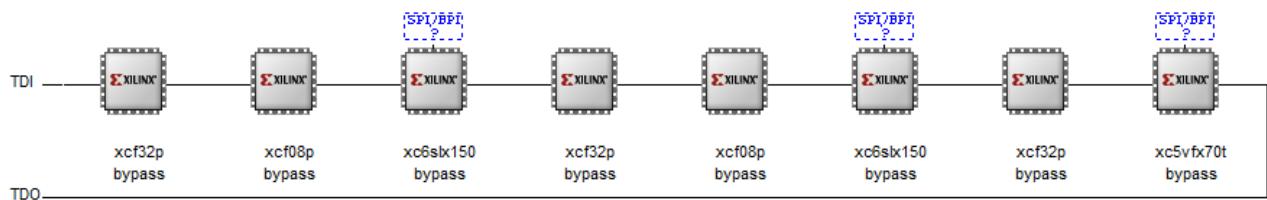


Figure 21: Complete JTAG chain (the order is S6A-S6B-V5)

FPGA VME programming

There is an alternative way for programming the IBL ROD FPGAs: via VME. This procedure can be done using **STAPL** files (slower) or **ACE** files (faster), which can be used to configure the FPGAs through the SBC.

Programming FPGAs via STAPL files

How to generate a STAPL file:

- open Impact,
- manually generate a JTAG chain like the one in Fig. 21 (alternatively an IPF file can be asked to the author)
- select the BIT or MCS file for the device to be configured
- select Ouput → STAPL File → Create STAPL File
- IMPORTANT: in case you are generating the STAPL file for programming a PRM (XCF device) be sure to run at this point the command: Erase device
- Program the device
- select Ouput → STAPL File → Stop writing to STAPL File
- The STAPL file is ready to be transferred to the SBC

The **bit2stapl** program in IblUtils allows to convert a bit file into a stapl one.

Sample script for generating a STAPL file from a BIT file (under Linux):

```
#!/bin/bash
#
# create .cmd file for impact and execute it
# convert .bit (input argument) to stapl
#
E_ERR_ARG=65

if [ -z "$1" ]
then
    echo "Utilizzo: `basename $0` nomefile"
    exit $E_ERR_ARG
fi

if [ -e "$1" ]
then
    FILE=$1
    FILENAME=${FILE%\.*}
    EXTENSION=${FILE##*.}

    if [ ${EXTENSION} != "bit" ]
    then
        echo "Error: invalid file type (.${EXTENSION}) - .bit is required"
        exit $E_ERR_ARG
    else
        FILEOUT=${FILENAME}.stapl
```

```

echo "Converting $1 to $FILEOUT"
fi

else
    echo "Error: file $1 doesn't exist!"
    echo "Exiting script"
    exit $E_ERR_ARG
fi

FILECMD=doStaplTmp.cmd

if [ -e "${FILECMD}" ]
then
rm -f $FILECMD
fi

echo "setMode -bscan" >> $FILECMD
echo "addDevice -p 1 -part xcf32p" >> $FILECMD
echo "addDevice -p 2 -part xcf08p" >> $FILECMD
echo "addDevice -p 3 -part xc6slx150" >> $FILECMD
echo "addDevice -p 4 -part xcf32p" >> $FILECMD
echo "addDevice -p 5 -part xcf08p" >> $FILECMD
echo "addDevice -p 6 -part xc6slx150" >> $FILECMD
echo "addDevice -p 7 -part xcf32p" >> $FILECMD
echo "addDevice -p 8 -part xc5vfx70t" >> $FILECMD
echo "assignFile -p 8 -file $FILE" >> $FILECMD
echo "setCable -p stapl -file $FILEOUT" >> $FILECMD
echo "program -p 8" >> $FILECMD
echo "closeCable" >> $FILECMD
echo "quit" >> $FILECMD

impact -batch doStaplTmp.cmd

if [ -e "${FILEOUT}" ]
then
echo "$FILEOUT created!"
else
echo "Error $FILEOUT tapple file creatednot created!"
fi

exit 0

```

How to program the device (FPGA/PROM):

The **ProgramIBLROD** tool in IblUtils allows to configure the ROD FPGAs by choosing the *stapl* file and the slot number.

This tool makes use of the Altera SW JAM Player in the following way:

- *./jp_25 -X<slot_number> -v -aRUN_XILINX_PROC <stapl_file>*

As an example you can run the following command:

```
./jp_25 -X6 -v -aRUN_XILINX_PROC /home/davide/data_flow_eth_s6a.stapl
```

in order to program the FPGA S6A on the IBL ROD placed in slot 6.

Programming a S6 or V5 device requires about 1 minute and 30 seconds.
Programming a XCF32P PROM device requires about 15 minutes.

Programming FPGAs via ACE files

In the git branch feature/Ace_player in the folder IblUtils/Vme/scripts you can find the following tools:

- **bit2mcs** (interactive, no arguments needed)

If you ned to program the PROM, you need to convert the BIT file into the MCS format.

Warning: 2 MCS files are generated for the Spartan6 devices.

- **bit2svf**

It allows to convert one BIT or MCS files into a SVF (Serial Vector Format) file. This is an intermediate format before converting to ACE.

Examples on how to use it for BIT files:

- bit2svf -8 rodMaster.bit -o <master.svf>
(8 is the position of the device in the JTAG chain starting from 1)
- bit2svf -3 rodSlave0.bit -o <slave0.svf>
- bit2svf -6 rodSlave1.bit -o <slave1.svf>

Examples on how to use it for MCS files:

- bit2svf -7 xcf32p.mcs -o <master_prom.svf>
- bit2svf -1 xcf32p.mcs -2 xcf08p.mcs -o <slave0_prom.svf>
- bit2svf -4 xcf32p.mcs -5 xcf08p.mcs -o <slave1_prom.svf>

- **svf2ace** (interactive, no arguments needed)

It allows to convert the SVF file into an ACE file.

Then in the folder IblUtils/Vme/bin/i686-slc5-gcc43-opt, you can find the tool:

- **AceProgrammer**

which actually programs the FPGAs or PROMs. It can ber un in this way:

./AceProg <slot_number> <ace_file.ace>

Programming a S6 or V5 device requires about 15 seconds.

Programming a XCF32P PROM device requires about 4 minutes.

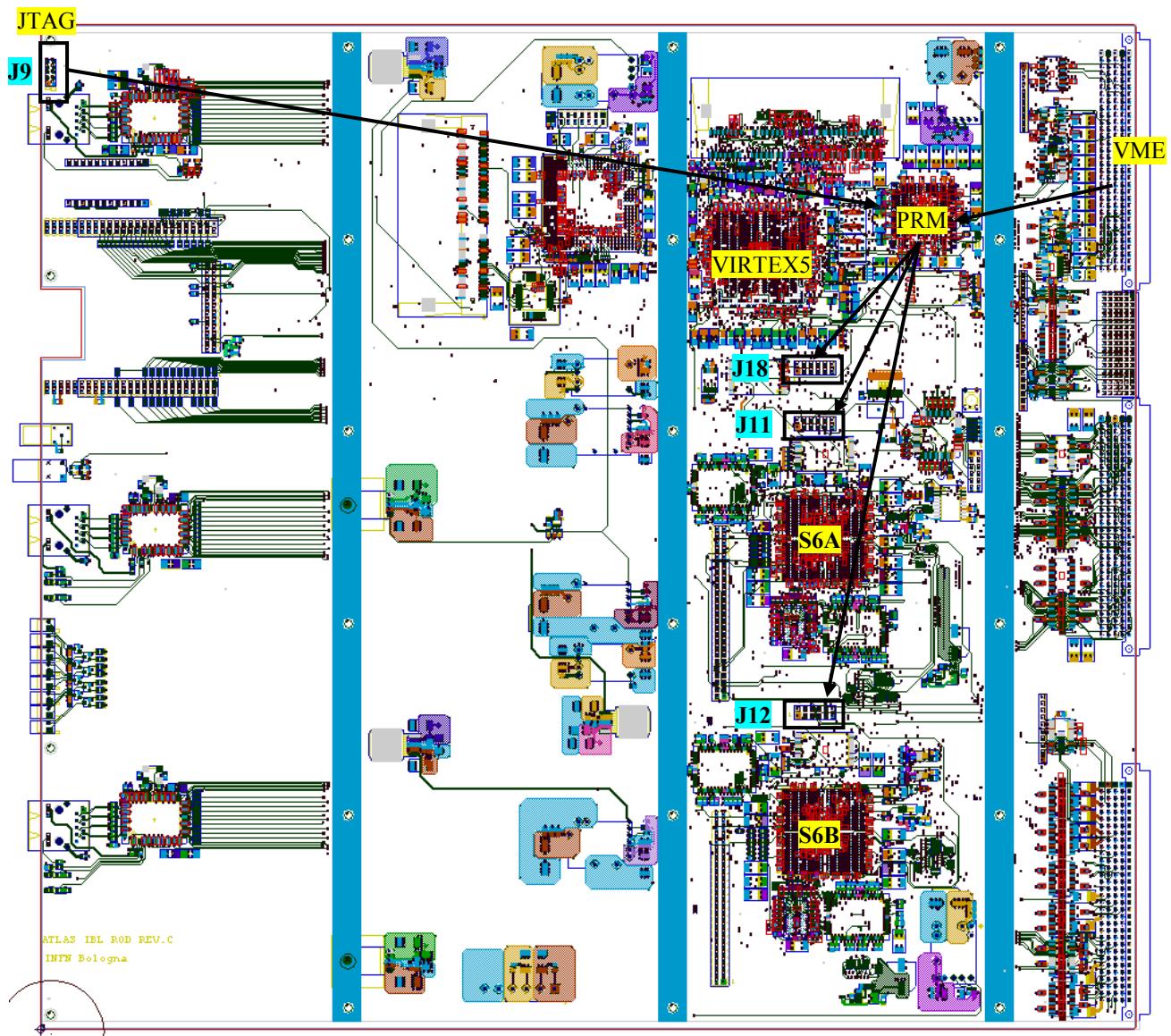


Figure 22: JTAG or VME programming the FPGAs

Booting embedded processors

The IBL ROD can be completely configured and operated without using a JTAG programmer. For instance via VME it is possible to load the FPGAs firmware and embedded processor software.

How to load the PPC program file (via VME)

- 1) create a bitfile (**rodmaster_flashloader.bit**) adding to the rodMaster ISE project an embedded dedicated program file (**flashloader.elf**) which enables HPI access via VME (Warning: a dedicated executable is needed because HPI control over VME is disabled by default. Alternative schemes could be envisaged but this has been chosen for safety: if you want to act as a master of the PPC bus you are required to load a dedicated fw).

To do this, from the ISE GUI select the PPC → Xmp/Elf file Associations → flashloader.elf and then recompile the whole project.

- 2) create a STAPL file from the BIT one → **rodmaster_flashloader.stapl**

- 3) program Virtex5 via VME using JAM player, for instance:

```
~/jp_25/source/jp_25 -X<slot_number> -v -aRUN_XILINX_PROC  
~/stapl/master_slave/rodmaster_flashloader.stapl
```

- 4) convert the main PPC software from ELF to SREC format:

```
powerpc-eabi-objcopy -S -O srec ./iblDsp_ppc.elf ./iblDsp_ppc.srec
```

You can find the executable powerpc-eabi-objcopy in the following Xilinx folder:
Xilinx/14.6/ISE_DS/EDK/gnu/powerpc-eabi/nt/bin

- 5) program the Virtex5 FLASH via VME using the SREC file using a tool from IblUtils:

```
~/daq/RodDaq/IblUtils/bin/ProgramVirtexFlash <slot_number>  
~/stapl/master_slave/iblDsp_ppc.srec
```

(it takes about one minute)

- 6) create a new bitfile (**rodmaster_bootloader.bit**) with the rodMaster ISE project and an embedded dedicated elf file which allows reading the file for FLASH and executing the SREC file, called **bootloader.elf**.

- 7) create the STAPL file from the BIT one → **rodmaster_bootloader.stapl**

- 8) program Virtex5 via VME , for instance:

```
~/jp_25/source/jp_25 -X<slot_number> -v -aRUN_XILINX_PROC  
~/stapl/master_slave/rodmaster_bootloader.stapl
```

At the end the PPC boots up by reading the SREC file from FLASH in a few seconds.

The FLASH stores its contents permanently, so its contents have to be updated onl when a new ELF program has to be used. If no modifications are needed, if the Virtex5 PROM contains the **rodmaster_bootloader.mcs** file (equivalent of the rodmaster_bootloader.bit file for the PROM) and the FLASH contains the required SREC program, when switching on the board the PPC boots automatically.

In order to speed up the procedure, instead of the STAPL files, the ACE ones can be used (in conjunction with AceProg). In order to program the FLASH the usual ProgramVirtexFlash will be used.

How to load the MB program file (via Ethernet)

Once the PPC is programmed, a primitive can be downloaded and started (loadSlaves) from an external PC, in order to load a BIN file (ELF equivalent) into the slaves DDR2 and make them boot.

How to reset the ROD

You can reset the ROD in one of the following ways:

- by pressing the push button on the front panel;
- by asserting the *vme_sysreset* signal on the VME backplane (all the RODs in the crate are reset when asserting this signal)
- by using the program **reset_ROD_FPGAs** (in IblUtils/Jp_25/source) which allows to selectively reset one FPGA among the master and the two slaves. The menu of the program is shown when run without any options.
- by selectively forcing the FPGAs to be reconfigured from the related PROM by using the option in the **ProgramIBLROD** (from the FPGAs point of view, this is equivalent to switching off and on the board).

MDSP JTAG programming

How to program ROD MDSP via JTAG:

- The MDSP has its own JTAG connector on which the Texas Instruments programmer (**XDS510 TI DSP JTAG Emulator Programmer TMS320**) can be plugged: **J5**.
- The Texas Instruments programmer flying leads shall be plugged on odd number pins:
 - *tdi*: pin 3
 - *tdo*: pin 7
 - *tck*: pin 11
 - *tms*: pin 1
 - *trst*: pin 2

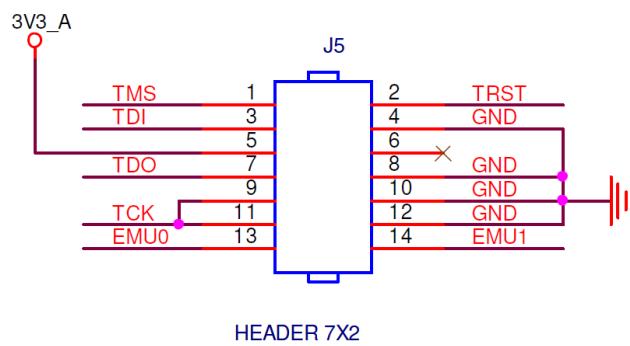


Figure 23: MDSP JTAG connector J5

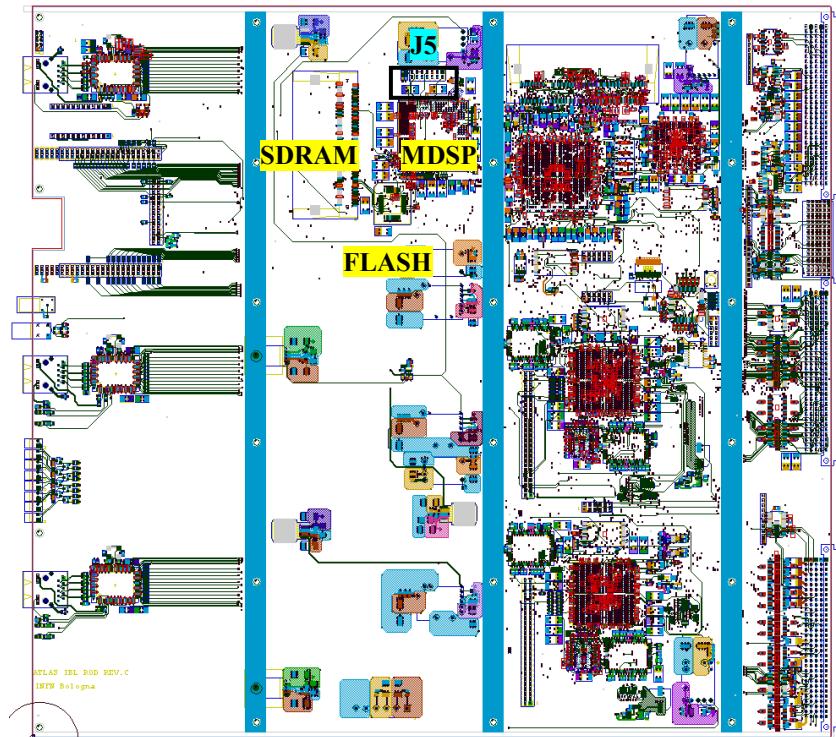


Figure 24: MDSP JTAG connector on the ROD

Clock distribution

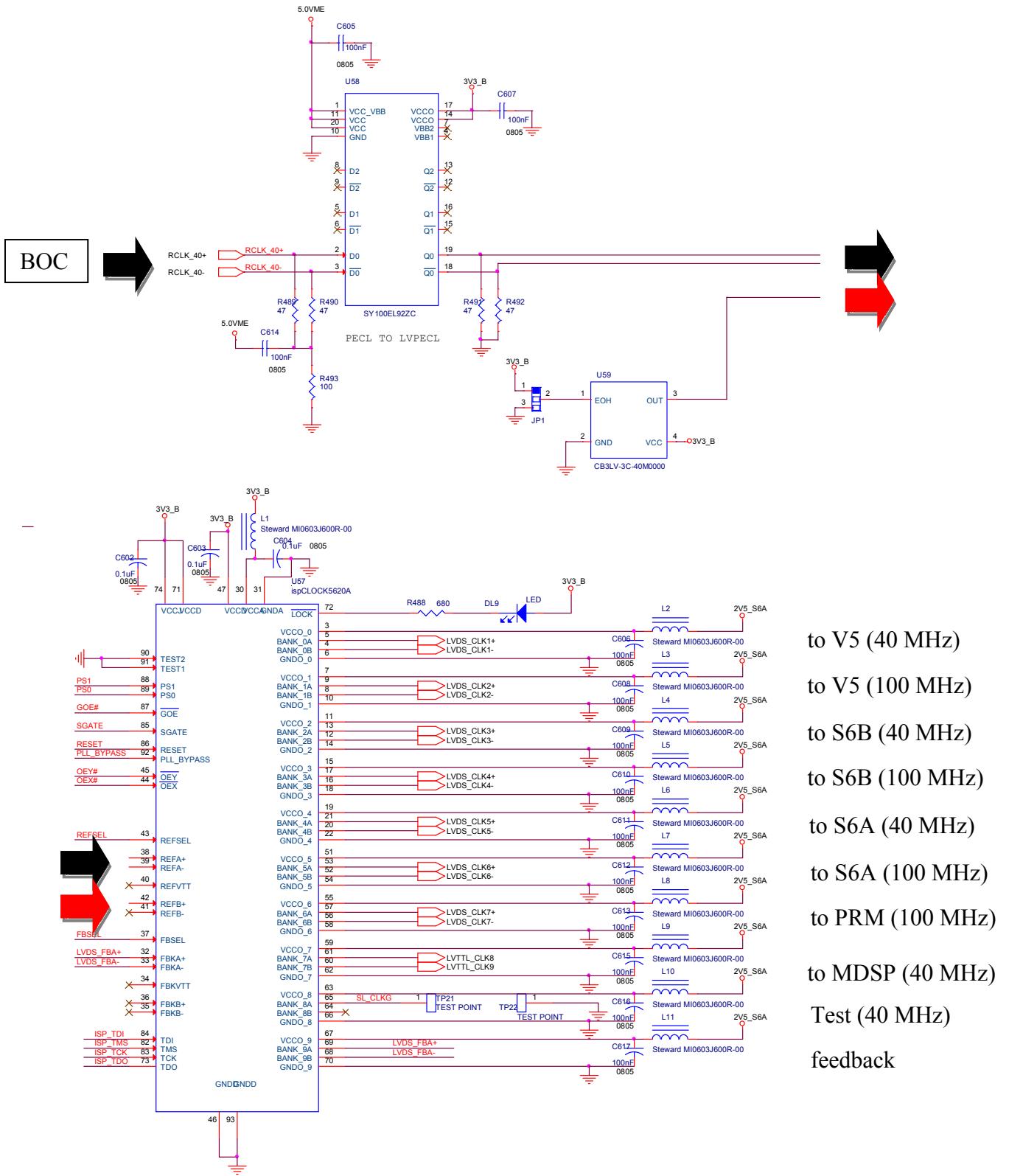


Figure 25: IBL ROD clock distribution

The ROD receives a PECL clock signal from the BOC through the P3 backplane connector. If this clock is not available, a 40 MHz on-board oscillator clock can be used. After PECL to LVPECL conversion (SY100EL92ZC), the clock enters an in-system-programmable high fanout enhanced zero delay clock generator (Lattice ispCLOCK5620A), which provides up to 20 single-ended or 10 differential clock outputs.

Each large FPGA (Virtex5, S6A and S6B) receive 2 LVDS clock pairs (namely 40 MHz and 100 MHz), while the PRM receives one LVDS clock signal at 100 MHz. The MDSP receives 2 LVTTL clock signals at 40 MHz.

The Lattice ispCLOCK5620A chip is programmable using the Lattice programmer ispDOWNLOAD cable (RS components.it code: 703-1317) and related software (PAC-Designer), by connecting the cable flying leads to connector J8.

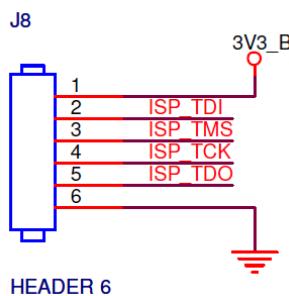


Figure 26: ISP JTAG connector

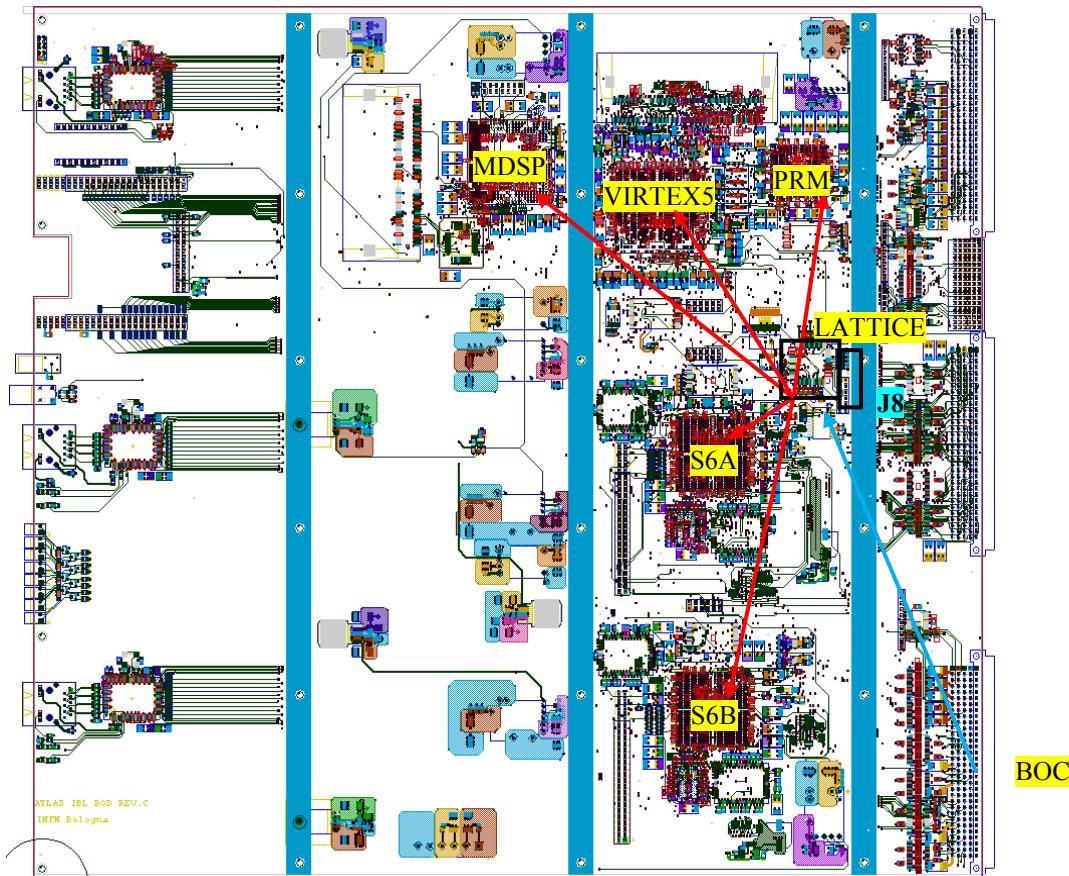


Figure 27: ROD clock distribution

In order to program the ispCLOCK5620A chip, the software tool **PAC Designer** from Lattice has to be used (see Fig. 28).

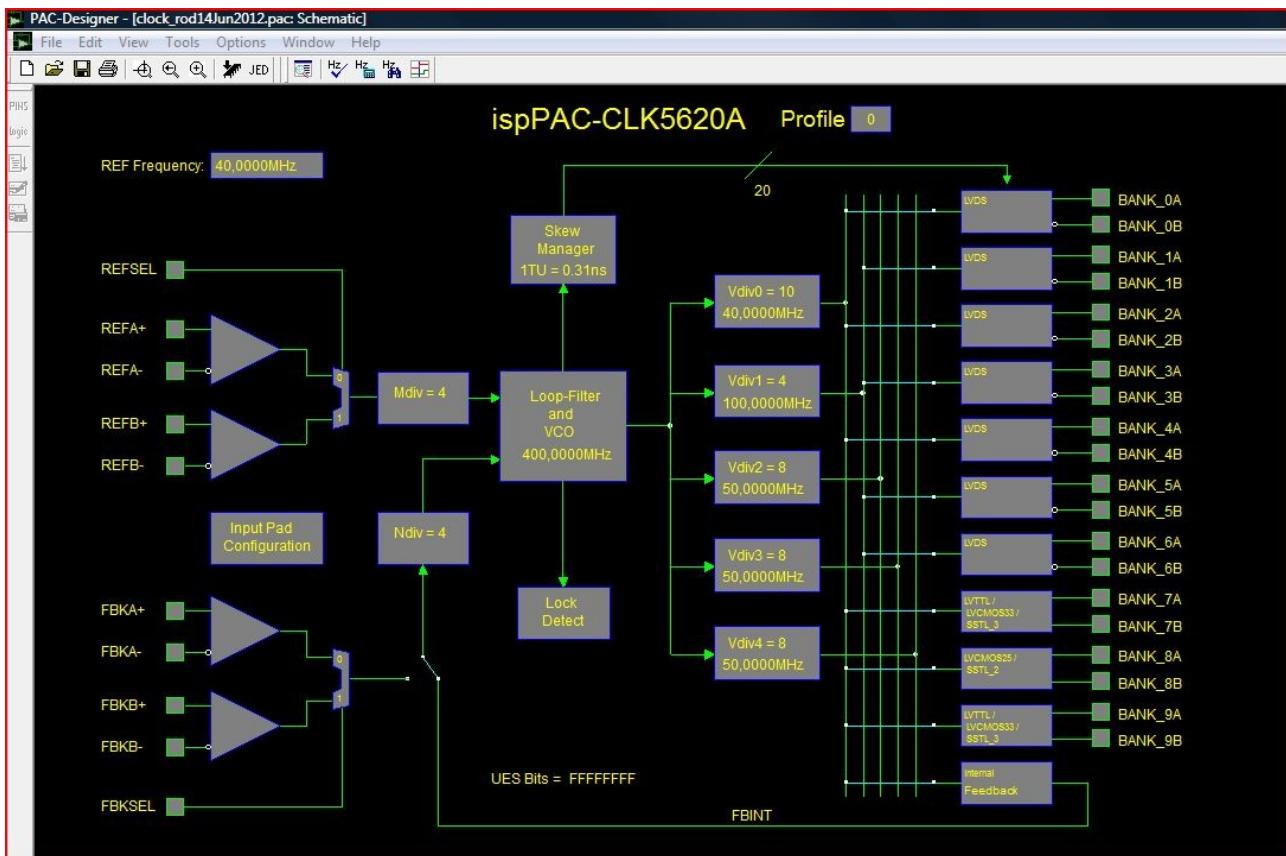


Figura 28: PAC Designer project

Once the project is complete, a JEDEC output file can be produced (see icon on the menu bar “Export JEDEC file”): this file is a bitstream that can be used to program the chip with the desired functionality. Do not use PAC Designer to directly program the chip, an other tool has to be used.

The tool used to actually program the chip is **Diamond Programmer** (see Fig. 29): the program searches for a Lattice chip on the JTAG chain and then downloads the JEDEC file in a few seconds, after pressing the “Program” button. It is necessary to press the reset button in order to have the chip working with the file just being downloaded.

Lattice Diamond Programmer is available as a tool belonging to the complete Lattice Diamond development kit.

The tool release being used up to now are:

- Lattice Diamond 1.3
- Lattice PAC-Designer 6.1

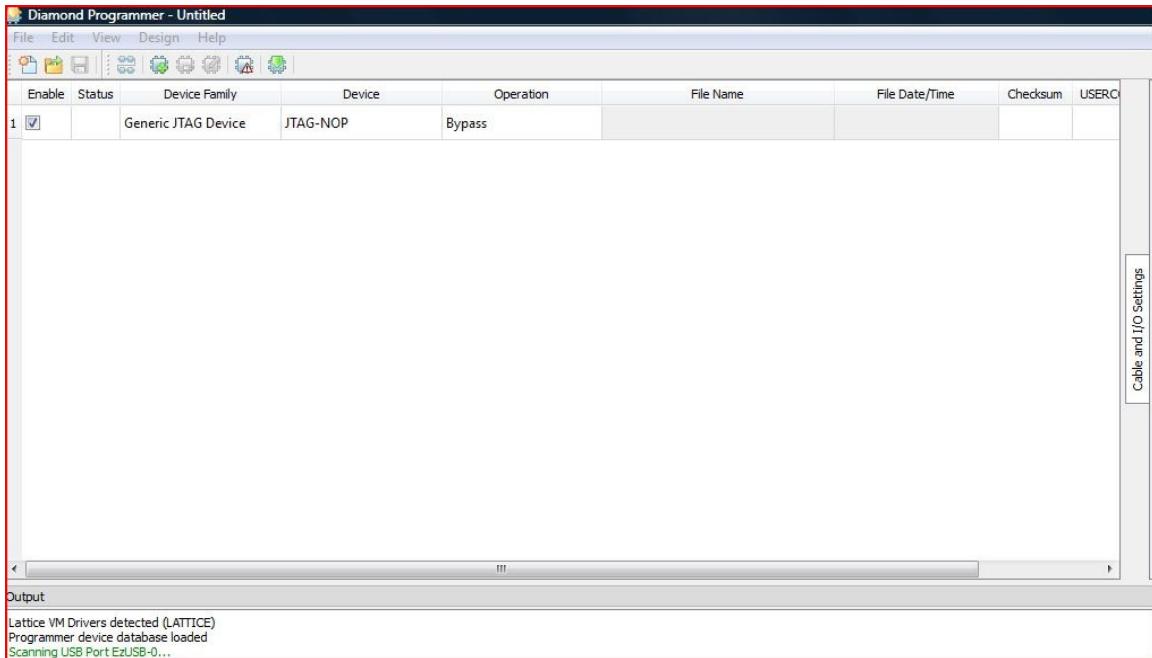


Figura 29: Diamond Programmer graphical user interface

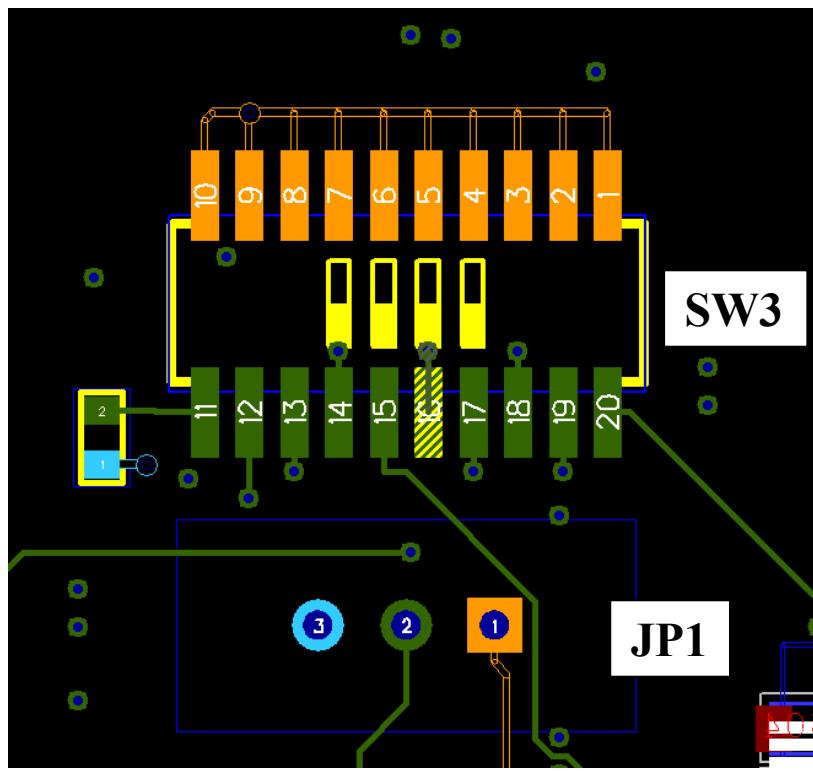


Figura 30: SW3 with REFSEL highlighted

SW3 bit 6 (highlighted in Fig. 30) allows you to choose between BOC clock (OFF position) and local oscillator ROD clock (ON position), all the other bits shall be in OFF position. If you plan to use the ROD local clock, be sure that JP1 switch is pushed on the left position.

Power supply distribution

The ROD power supply can be provided in either of two ways:

- In lab it can be provided by means of a dedicated connector (**M1**) placed in the middle bottom part of the board.

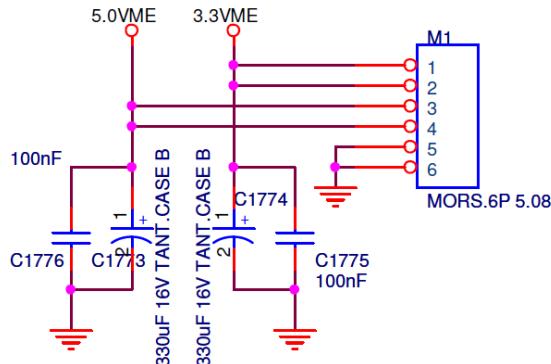


Figure 31: M1 power supply connector

- It can be supplied by a standard ATLAS VME crate power supply.

Fig. 32 and Fig. 33 show the ROD internal power distribution used to derive all the required voltage levels and the expected current consumption:

- 5.0 V
- 3.3 V
- 2.5 V
- 1.8 V
- 1.5 V
- 1.2 V
- 1.0 V
- 0.9 V

The ROD features both switching and linear power modules:

- 8 switching power modules;
- 7 linear LDO voltage regulators;
- 2 voltage reference REF3330 to provide 1.5V voltage for SSTL3-I.

These regulators are placed next to the metallic vertical bars, which help in dissipating heat and in maintaining mechanical rigidity on such a large board.

The TTCrq mezzanine and the VME buffers directly receive 5V and 3.3V from the external power supply.

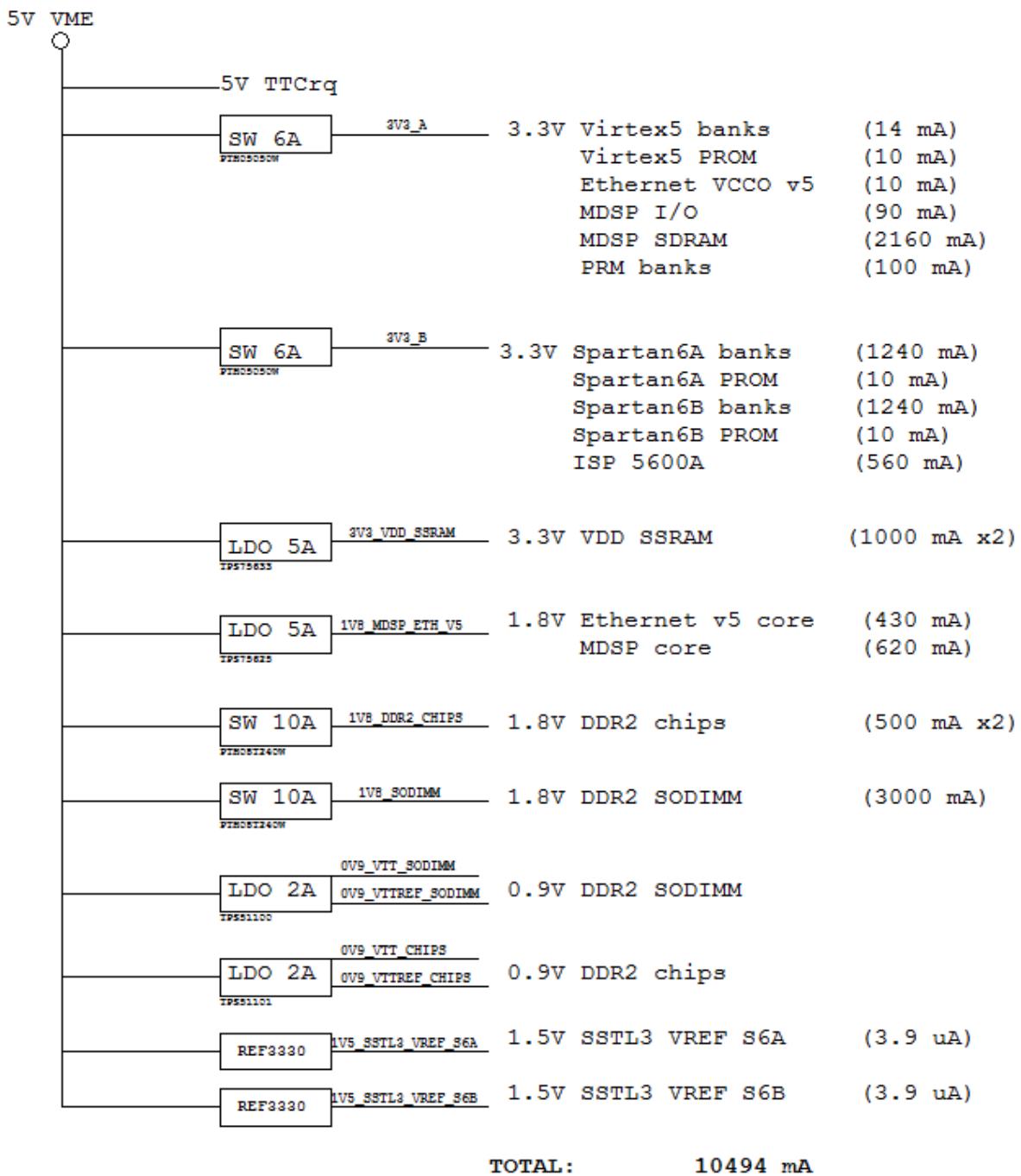


Figure 32: 5V power supply distribution

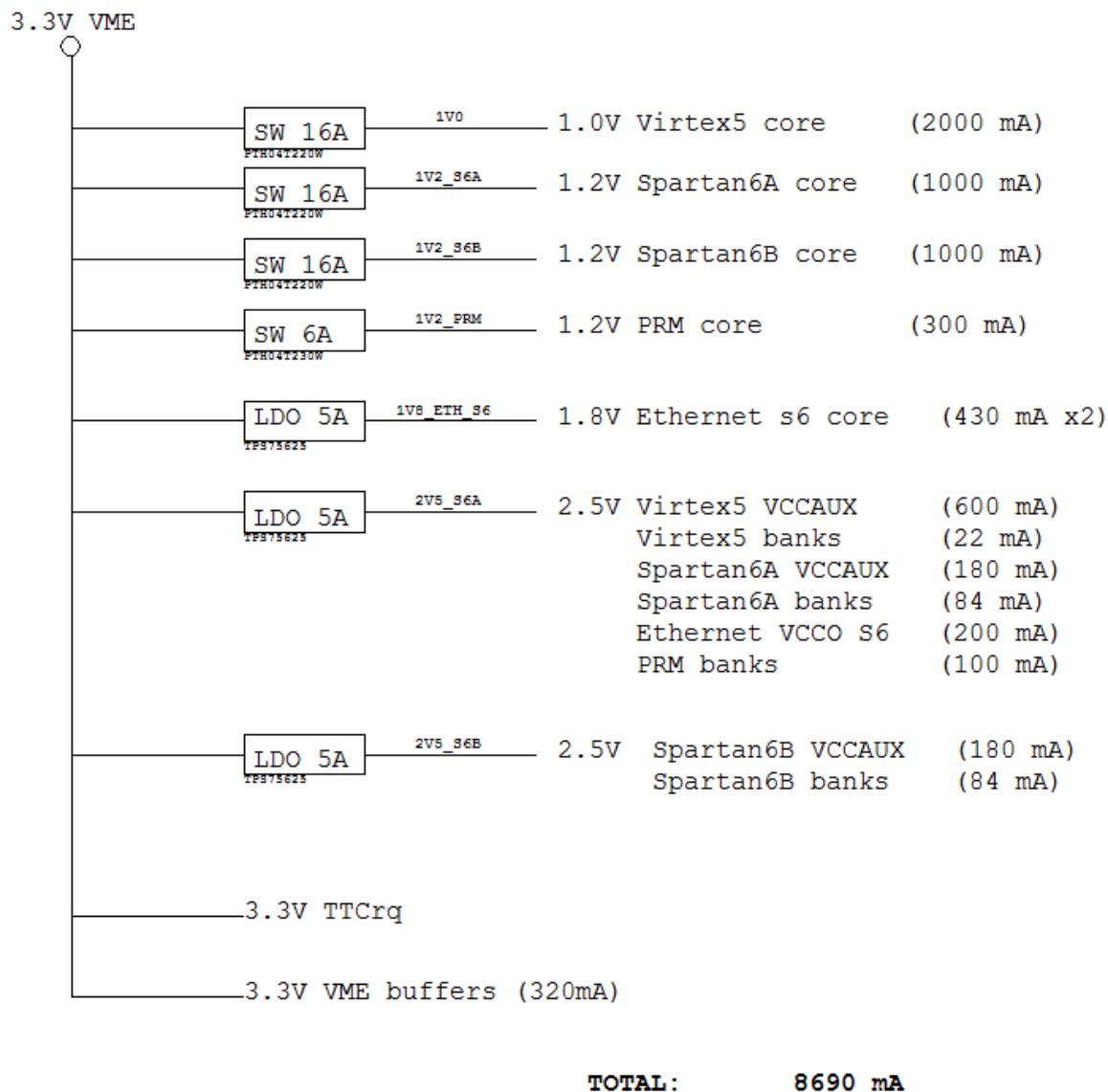


Figure 33: 3.3V power supply distribution

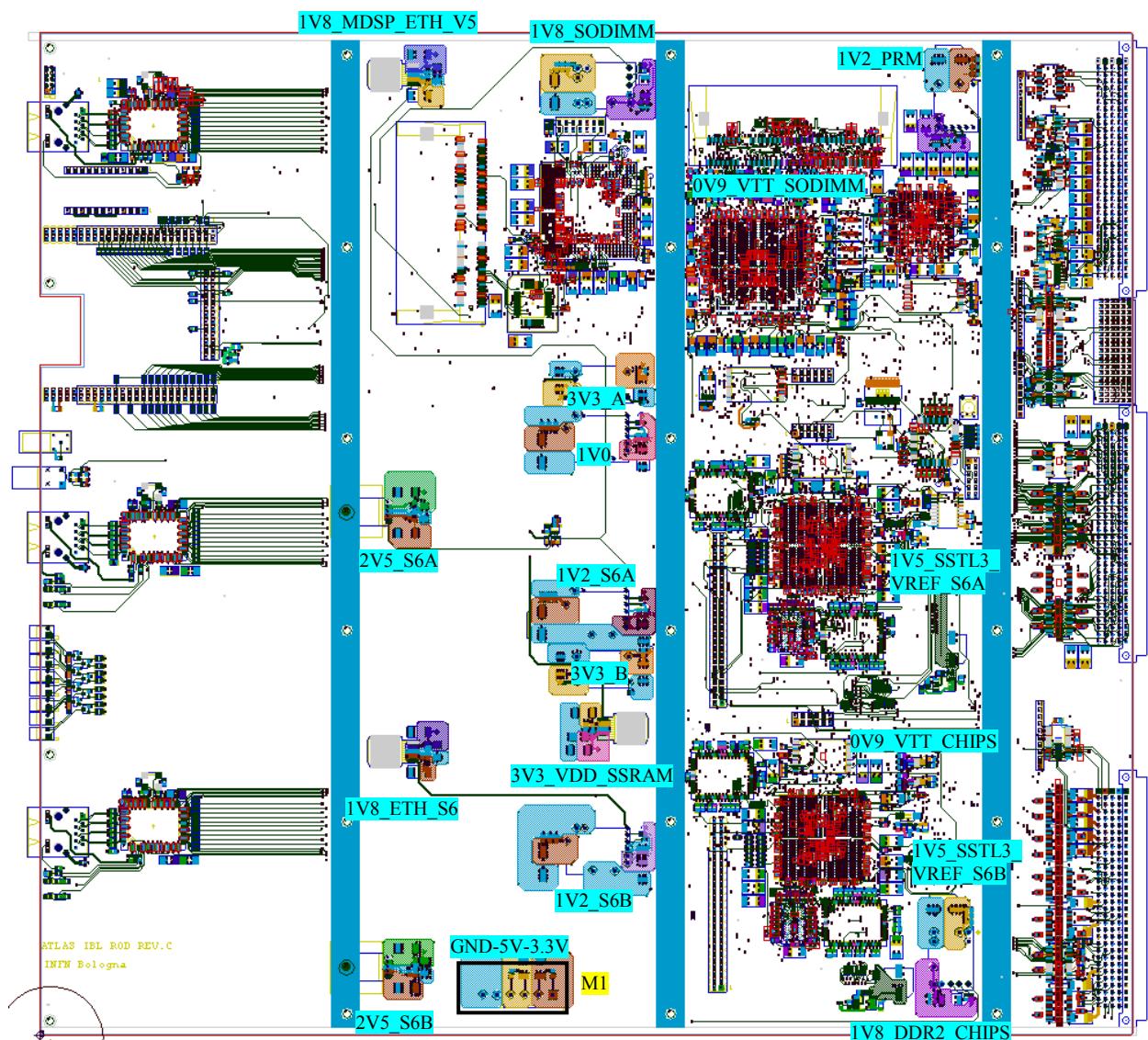


Figure 34: ROD power supply regulators distribution

BOC-ROD interface signals

The interface signals between BOC and ROD can be logically subdivided in 4 categories (the signal names used are the same as in the ROD schematics):

1. RX bus: $RD[95:0]$ @ 80 MHz (8 12-bit buses each serving 4 FE-I4s)
2. CMD bus: $XC[15:0]$ @ 40 MHz (1 signal out of 2 Fe-I4s)
3. S-Link bus: $L_UD[31:0], L_UD1-[31:0]$ @ 80 MHz plus 24 control signals:
 $L_URESET\#$,
 $L_UTEST\#$,
 $L_LDOWN\#$,
 $L_UWEN\#$,
 $L_UCTRL\#$,
 $L_LFF\#$
(# ranging from 0 to 3)
4. setup bus: $SA_BOC_D[7:0], SA_BOC_AD[15:0]$ plus 3 control signals:
 $SBUSY, SSTBN$ and $SWRN$

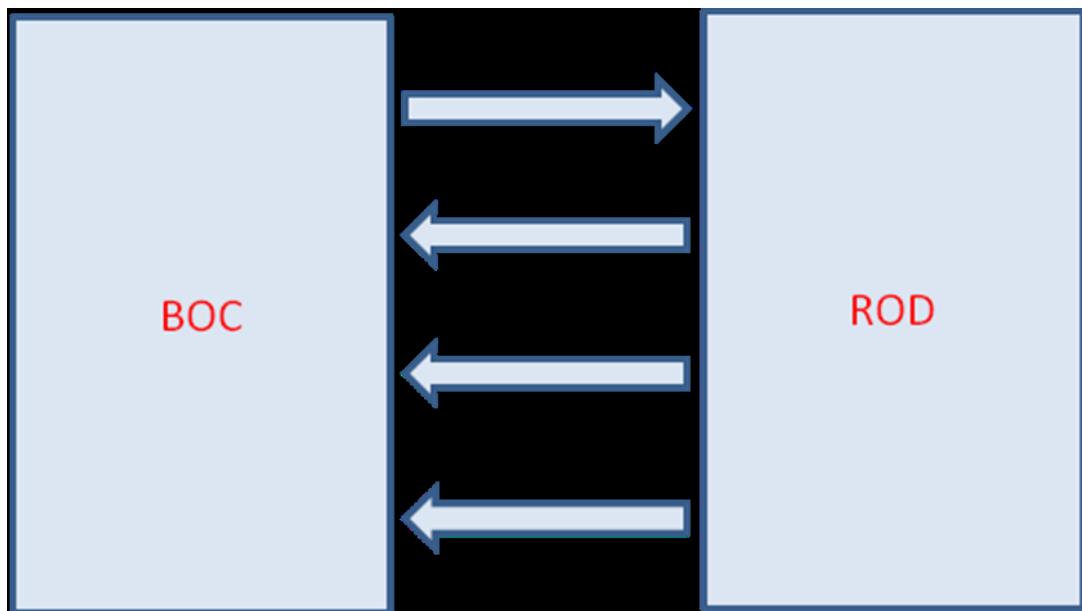


Figure 35: BOC-ROD interface signals

Fig. 36, 37, 38 and 39 show the backplane connector signals between BOC and ROD. Signals belonging to RX bus, CMD bus and S-Link bus can be configured in either of the following ways:

- LVTTL
- SSTL3_I

The setup bus can only be configured as LVTTL.

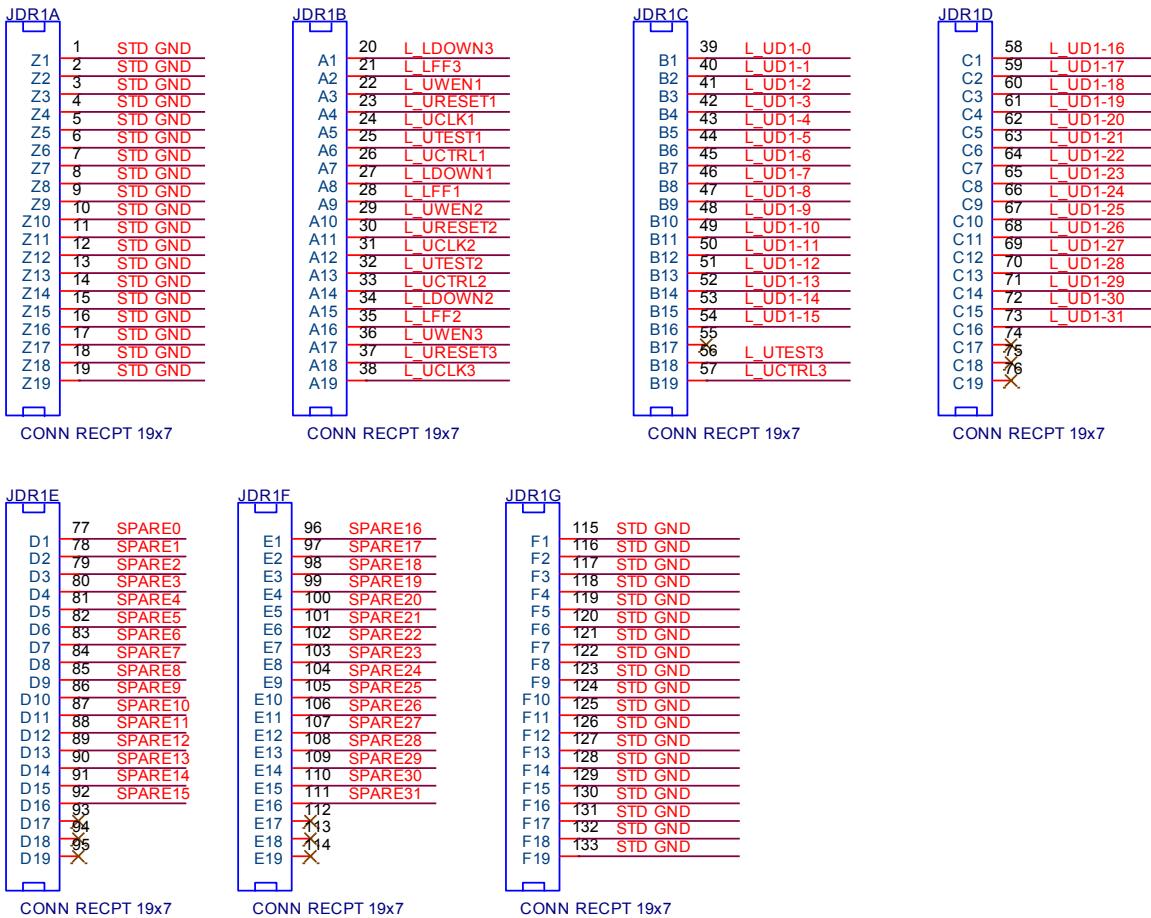


Figure 36: P0 connector wire diagram

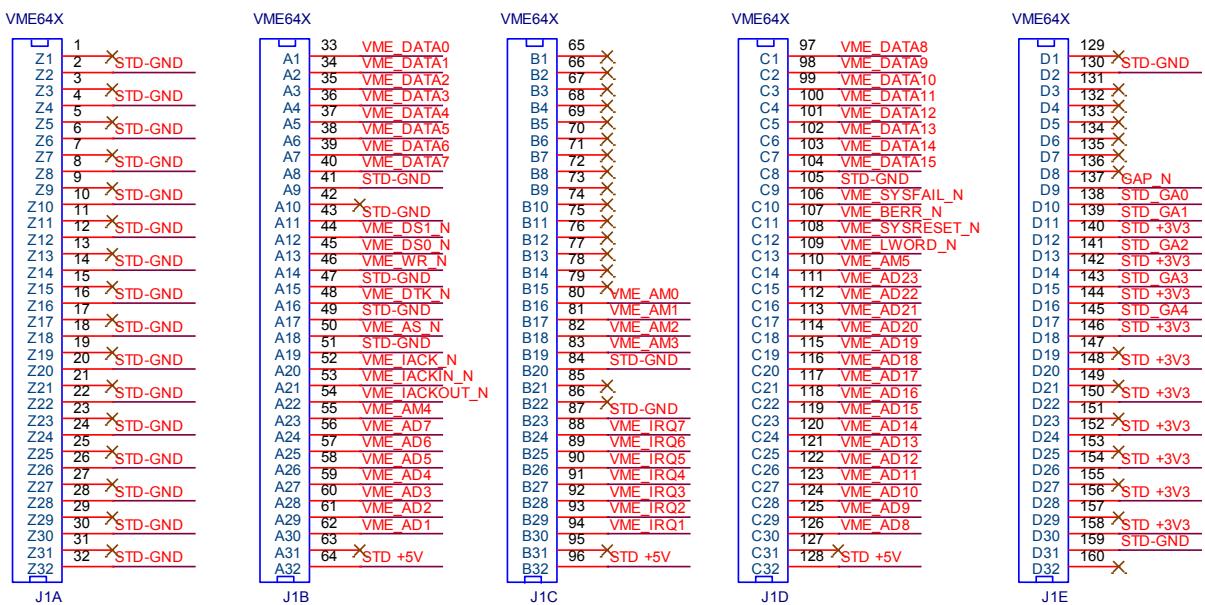


Figure 37: P1 connector wire diagram

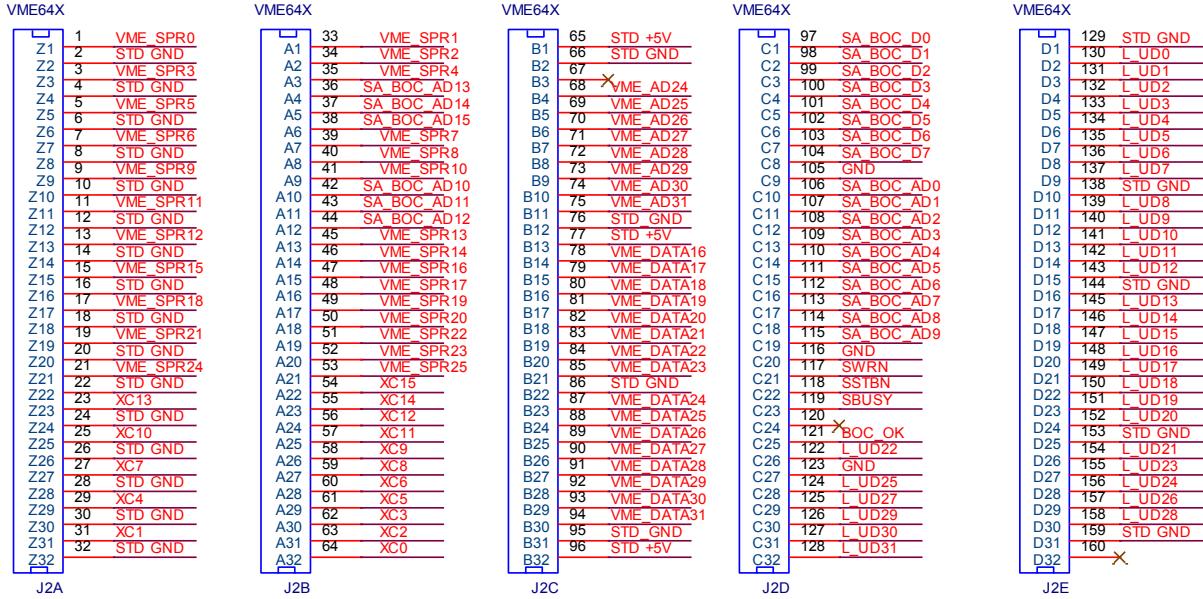
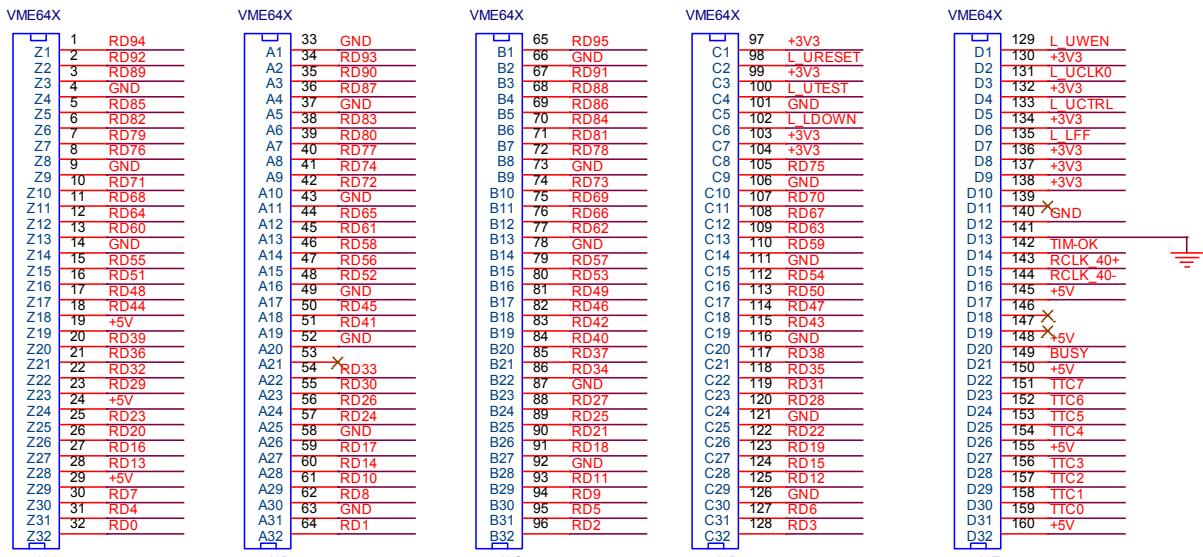


Figure 38: P2 connector wire diagram



ROD bus

The IBL ROD bus is an asynchronous bus used to access in read/write mode all ROD FPGAs internal registers and to drive the BOC bus. The ROD bus is driven by the Virtex5 FPGA which acts as the bus master and reaches the 2 Spartan6 devices and the BOC, as shown in Fig. 40.

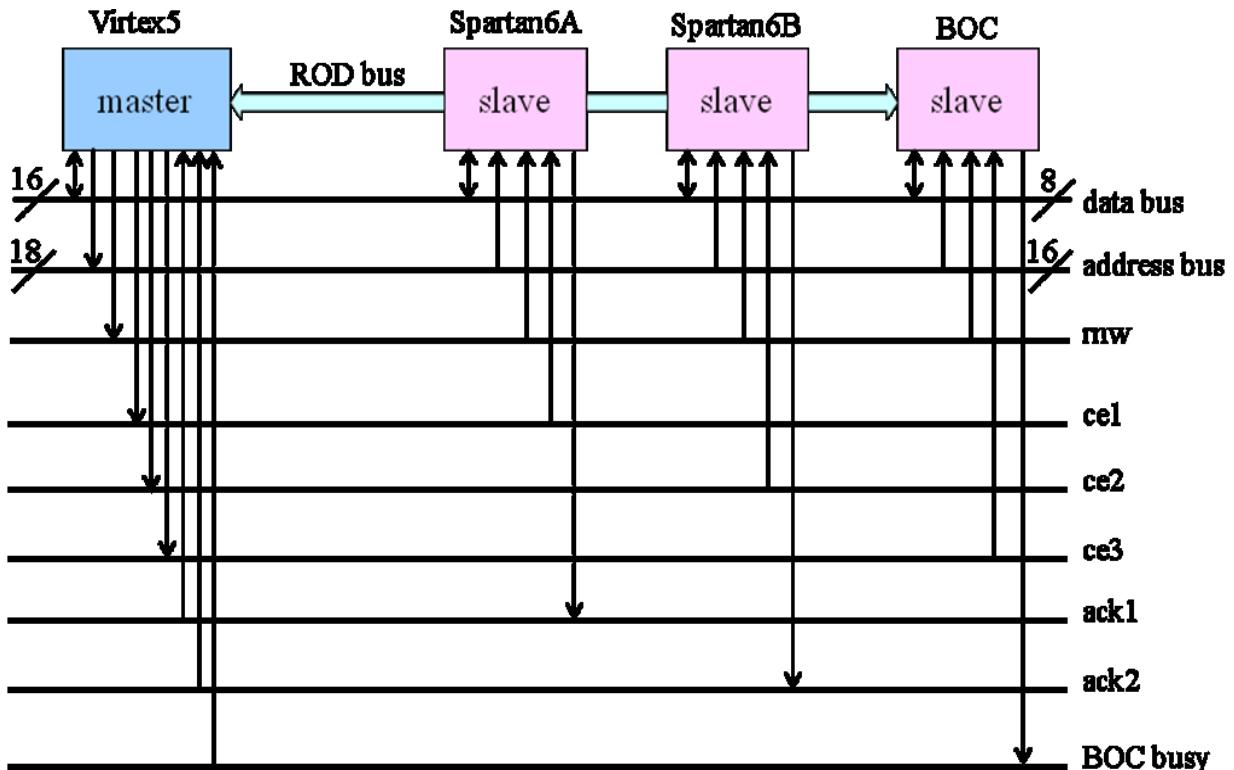


Figure 40: IBL ROD bus

The ROD data bus consists of:

- 18-bit address bus (*ROD_BUS_ADDR[17:0]*)
- 16-bit data bus (*ROD_BUS_DATA[15:0]*)
- 1 RNW control signal (*ROD_BUS_RNW*)
- 3 CE control signals (*ROD_BUS_CE1*, *ROD_BUS_CE2*, *ROD_BUS_CE_BO*)
- 3 ACK feedback signals (*ROD_BUS_ACK1*, *ROD_BUS_ACK2*, *ROD_BO_C_BUSY*)

The BOC board deals with just a subset of the data bus (8-bit bus) and address bus (16-bit bus).

Fig. 40 shows the microprocessors actually driving the ROD bus:

- **the MDSP:** it is controlled by the VME CPU through the HPI port (*MDSP_HD[15:0]*) and drives the ROD bus using the EMIF bus (*MDSP_EA[21:2]*, *MDSP_ED[31:0]*).
- **the PowerPC:** it is controlled by the VME CPU through a 32 bit port (*PRM_DATA_IN[31:0]*) and drives the ROD bus using the PLB bus and a specific peripheral.

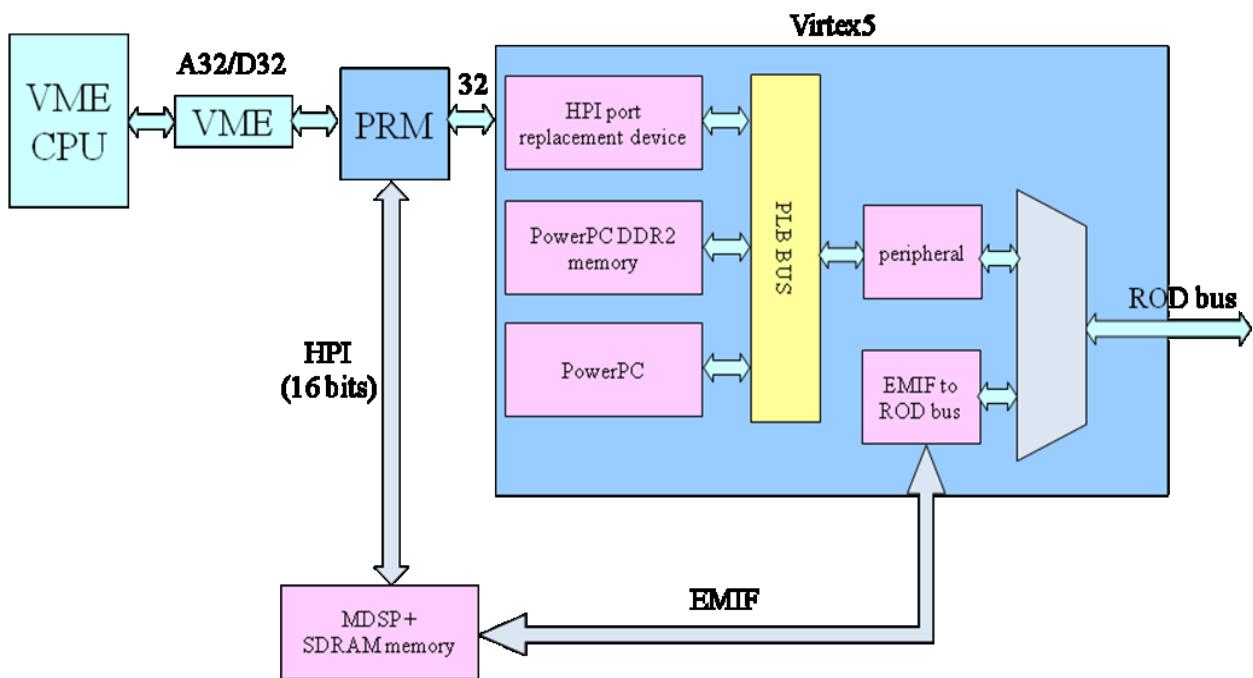


Figure 41: IBL ROD bus

PRM pin assignment (UCF file)

```

CONFIG PROHIBIT = A3;
CONFIG PROHIBIT = T20;
CONFIG PROHIBIT = Y21;
CONFIG PROHIBIT = AA22;
CONFIG PROHIBIT = AA20;
CONFIG PROHIBIT = AB20;
CONFIG PROHIBIT = U15;
CONFIG PROHIBIT = T6;
CONFIG PROHIBIT = T5;
CONFIG PROHIBIT = K3;
CONFIG PROHIBIT = J4;
CONFIG PROHIBIT = J22;
CONFIG PROHIBIT = J20;
CONFIG PROHIBIT = H21;
CONFIG PROHIBIT = H22;
NET "board_address[0]" LOC = Y3;
NET "board_address[1]" LOC = AB3;
NET "board_address[2]" LOC = Y4;
NET "board_address[3]" LOC = AA4;
NET "board_address[4]" LOC = AB4;
NET "board_address[5]" LOC = Y5;
NET "board_address[6]" LOC = AB5;
NET "board_address[7]" LOC = Y6;
NET "fpga_A_tck" LOC = K2;
NET "fpga_A_tdi" LOC = H3;
NET "fpga_A_tdo" LOC = L1;
NET "fpga_A_tms" LOC = K1;
NET "fpga_B_tck" LOC = M2;
NET "fpga_B_tdi" LOC = L3;
NET "fpga_B_tdo" LOC = M3;
NET "fpga_B_tms" LOC = M1;
NET "fpga_C_tck" LOC = N3;
NET "fpga_C_tdi" LOC = N1;
NET "fpga_C_tdo" LOC = P2;
NET "fpga_C_tms" LOC = P1;
NET "jtag_tck" LOC = H1;
NET "jtag_tdi" LOC = G1;
NET "jtag_tdo" LOC = H2;
NET "jtag_tms" LOC = G3;
NET "mdsp_bootmode[0]" LOC = R19;
NET "mdsp_bootmode[1]" LOC = P18;
NET "mdsp_bootmode[2]" LOC = N16;
NET "mdsp_bootmode[3]" LOC = AA21;
NET "mdsp_bootmode[4]" LOC = AB21;
NET "mdsp_hcntl0" LOC = Y19;
NET "mdsp_hcntl1" LOC = AB18;
NET "mdsp_hcs_n" LOC = W17;
NET "mdsp_hd[0]" LOC = R20;
NET "mdsp_hd[1]" LOC = P22;
NET "mdsp_hd[2]" LOC = P21;
NET "mdsp_hd[3]" LOC = N22;
NET "mdsp_hd[4]" LOC = N20;
NET "mdsp_hd[5]" LOC = M22;
NET "mdsp_hd[6]" LOC = L22;
NET "mdsp_hd[7]" LOC = K22;
NET "mdsp_hd[8]" LOC = R22;
NET "mdsp_hd[9]" LOC = T21;
NET "mdsp_hd[10]" LOC = T22;
NET "mdsp_hd[11]" LOC = U20;
NET "mdsp_hd[12]" LOC = U22;
NET "mdsp_hd[13]" LOC = V21;
NET "mdsp_hd[14]" LOC = V22;
NET "mdsp_hd[15]" LOC = N19;
NET "mdsp_hds1_n" LOC = W18;
NET "mdsp_hhwil" LOC = Y17;
NET "mdsp_hrdy_n" LOC = Y18;
NET "mdsp_hrnw" LOC = AA18;
NET "mdsp_reset" LOC = AB19;
NET "prm_ctrl_in[0]" LOC = H8;
NET "prm_ctrl_in[1]" LOC = J7;
NET "prm_ctrl_in[2]" LOC = B1;
NET "prm_ctrl_in[3]" LOC = B2;
NET "prm_ctrl_in[4]" LOC = F7;
NET "prm_ctrl_in[5]" LOC = G7;
NET "prm_ctrl_in[6]" LOC = C4;
NET "prm_ctrl_in[7]" LOC = D3;
NET "prm_ctrl_in[8]" LOC = E6;
NET "prm_ctrl_in[9]" LOC = E5;
NET "prm_ctrl_in[10]" LOC = B3;
NET "prm_ctrl_in[11]" LOC = A2;
NET "prm_ctrl_in[12]" LOC = M6;
NET "prm_ctrl_in[13]" LOC = L6;
NET "prm_ctrl_in[14]" LOC = R4;
NET "prm_ctrl_in[15]" LOC = P4;
NET "prm_ctrl_in[16]" LOC = M7;
NET "prm_ctrl_in[17]" LOC = M8;
NET "prm_ctrl_in[18]" LOC = N6;
NET "prm_ctrl_in[19]" LOC = N7;
NET "prm_ctrl_in[20]" LOC = U4;
NET "prm_ctrl_in[21]" LOC = V3;
NET "prm_ctrl_in[22]" LOC = T4;
NET "prm_data[0]" LOC = V1;
NET "prm_data[1]" LOC = V2;
NET "prm_data[2]" LOC = U1;
NET "prm_data[3]" LOC = U3;
NET "prm_data[4]" LOC = T1;
NET "prm_data[5]" LOC = T2;
NET "prm_data[6]" LOC = R1;
NET "prm_data[7]" LOC = R3;
NET "prm_data[8]" LOC = L4;
NET "prm_data[9]" LOC = K4;
NET "prm_data[10]" LOC = K5;
NET "prm_data[11]" LOC = J6;
NET "prm_data[12]" LOC = K6;
NET "prm_data[13]" LOC = H4;
NET "prm_data[14]" LOC = H5;

```

```
NET "prm_data[15]" LOC = H6;
NET "prm_data[16]" LOC = F1;
NET "prm_data[17]" LOC = F2;
NET "prm_data[18]" LOC = F3;
NET "prm_data[19]" LOC = G4;
NET "prm_data[20]" LOC = E1;
NET "prm_data[21]" LOC = E3;
NET "prm_data[22]" LOC = D1;
NET "prm_data[23]" LOC = D2;
NET "prm_data[24]" LOC = C1;
NET "prm_data[25]" LOC = C3;
NET "prm_data[26]" LOC = F5;
NET "prm_data[27]" LOC = G6;
NET "prm_data[28]" LOC = K8;
NET "prm_data[29]" LOC = K7;
NET "prm_data[30]" LOC = E4;
NET "prm_data[31]" LOC = D5;
NET "viackout" LOC = K18;
NET "vme_addr[0]" LOC = C10;
NET "vme_addr[1]" LOC = A11;
NET "vme_addr[2]" LOC = C5;
NET "vme_addr[3]" LOC = B6;
NET "vme_addr[4]" LOC = B8;
NET "vme_addr[5]" LOC = C9;
NET "vme_addr[6]" LOC = D10;
NET "vme_addr[7]" LOC = C11;
NET "vme_addr[8]" LOC = E16;
NET "vme_addr[9]" LOC = A4;
NET "vme_addr[10]" LOC = C6;
NET "vme_addr[11]" LOC = A7;
NET "vme_addr[12]" LOC = C8;
NET "vme_addr[13]" LOC = D8;
NET "vme_addr[14]" LOC = A10;
NET "vme_addr[15]" LOC = D6;
NET "vme_addr[16]" LOC = C7;
NET "vme_addr[17]" LOC = D9;
NET "vme_addr[18]" LOC = D7;
NET "vme_addr[19]" LOC = B10;
NET "vme_addr[20]" LOC = D11;
NET "vme_addr[21]" LOC = A9;
NET "vme_addr[22]" LOC = A8;
NET "vme_addr[23]" LOC = A6;
NET "vme_addr[24]" LOC = A5;
NET "vme_addr[25]" LOC = F18;
NET "vme_addr[26]" LOC = F19;
NET "vme_addr[27]" LOC = E20;
NET "vme_addr[28]" LOC = F20;
NET "vme_addr[29]" LOC = F21;
NET "vme_addr[30]" LOC = E22;
NET "vme_addr[31]" LOC = F22;
NET "vme_am[0]" LOC = F16;
NET "vme_am[1]" LOC = G16;
NET "vme_am[2]" LOC = H16;
NET "vme_am[3]" LOC = J16;
NET "vme_am[4]" LOC = K16;
NET "vme_am[5]" LOC = M16;
NET "vme_as" LOC = M18;
NET "vme_berr" LOC = G19;
NET "vme_data[0]" LOC = A12;
NET "vme_data[1]" LOC = B12;
NET "vme_data[2]" LOC = C12;
NET "vme_data[3]" LOC = A13;
NET "vme_data[4]" LOC = C13;
NET "vme_data[5]" LOC = A14;
NET "vme_data[6]" LOC = B14;
NET "vme_data[7]" LOC = C14;
NET "vme_data[8]" LOC = D14;
NET "vme_data[9]" LOC = A15;
NET "vme_data[10]" LOC = C15;
NET "vme_data[11]" LOC = D15;
NET "vme_data[12]" LOC = A16;
NET "vme_data[13]" LOC = B16;
NET "vme_data[14]" LOC = C16;
NET "vme_data[15]" LOC = A17;
NET "vme_data[16]" LOC = C17;
NET "vme_data[17]" LOC = D17;
NET "vme_data[18]" LOC = A18;
NET "vme_data[19]" LOC = B18;
NET "vme_data[20]" LOC = C19;
NET "vme_data[21]" LOC = D19;
NET "vme_data[22]" LOC = A20;
NET "vme_data[23]" LOC = B20;
NET "vme_data[24]" LOC = C20;
NET "vme_data[25]" LOC = D20;
NET "vme_data[26]" LOC = A21;
NET "vme_data[27]" LOC = B21;
NET "vme_data[28]" LOC = D21;
NET "vme_data[29]" LOC = B22;
NET "vme_data[30]" LOC = C22;
NET "vme_data[31]" LOC = D22;
NET "vme_dir" LOC = H19;
NET "vme_ds0" LOC = J19;
NET "vme_ds1" LOC = K19;
NET "vme_dtack" LOC = L19;
NET "vme_ga_n[0]" LOC = F17;
NET "vme_ga_n[1]" LOC = G17;
NET "vme_ga_n[2]" LOC = H17;
NET "vme_ga_n[3]" LOC = J17;
NET "vme_ga_n[4]" LOC = K17;
NET "vme_gap_n" LOC = M19;
NET "vme_iack_n" LOC = G20;
NET "vme_iackin" LOC = H20;
NET "vme_irq[0]" LOC = L17;
NET "vme_irq[1]" LOC = M17;
NET "vme_irq[2]" LOC = H18;
NET "vme_irq_n" LOC = K20;
NET "vme_lword" LOC = L20;
NET "vme_oe" LOC = M20;
NET "vme_reset" LOC = K21;
NET "vme_sysfail" LOC = M21;
NET "vme_wr" LOC = G22;
NET "prm_ctrl_out[0]" LOC = W10;
NET "prm_ctrl_out[1]" LOC = V11;
NET "prm_ctrl_out[2]" LOC = R11;
```

```

NET "prm_ctrl_out[3]" LOC = AA12;
NET "prm_ctrl_out[4]" LOC = W12;
NET "prm_ctrl_out[5]" LOC = T12;
NET "prm_ctrl_out[6]" LOC = W14;
NET "prm_ctrl_out[7]" LOC = V13;
NET "prm_ctrl_out[8]" LOC = Y16;
NET "prm_ctrl_out[9]" LOC = V17;
NET "prm_ctrl_out[10]" LOC = V19;
NET "prm_ctrl_out[11]" LOC = T16;
NET "prm_ctrl_out[12]" LOC = V5;
NET "prm_ctrl_out[13]" LOC = R7;
NET "prm_ctrl_out[14]" LOC = U10;
NET "prm_ctrl_out[15]" LOC = V9;
NET "prm_strobe[0]" LOC = T3;
NET "prm_strobe[1]" LOC = P3;
NET "reset_cmds_in[0]" LOC = AB7;
NET "reset_cmds_in[1]" LOC = V7;
NET "reset_cmds_out[0]" LOC = AB8;
NET "reset_cmds_out[1]" LOC = AB9;
NET "ck40_N" LOC = J1;
NET "ck40_P" LOC = J3;
NET "rod_busy" LOC = AB6;
NET "busy_from_v5" LOC = Y15;
NET "master_reset" LOC = AA10;
NET "ck40_P" IOSTANDARD = LVDS_25;
NET "ck40_N" IOSTANDARD = LVDS_25;
NET "rod_busy" IOSTANDARD = LVCMOS33;
NET "rod_busy" DRIVE = 12;
NET "rod_busy" SLEW = SLOW;
NET "busy_from_v5" IOSTANDARD = LVCMOS33;
NET "master_reset" IOSTANDARD = LVCMOS33;
NET "done_virtex" IOSTANDARD = LVCMOS33;
NET "done_s6a" IOSTANDARD = LVCMOS33;
NET "done_s6b" IOSTANDARD = LVCMOS33;
NET "prm_ctrl_out[0]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[0]" DRIVE = 12;
NET "prm_ctrl_out[0]" SLEW = SLOW;
NET "prm_ctrl_out[1]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[1]" DRIVE = 12;
NET "prm_ctrl_out[1]" SLEW = SLOW;
NET "prm_ctrl_out[2]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[2]" DRIVE = 12;
NET "prm_ctrl_out[2]" SLEW = SLOW;
NET "prm_ctrl_out[3]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[3]" DRIVE = 12;
NET "prm_ctrl_out[3]" SLEW = SLOW;
NET "prm_ctrl_out[4]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[4]" DRIVE = 12;
NET "prm_ctrl_out[4]" SLEW = SLOW;
NET "prm_ctrl_out[5]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[5]" DRIVE = 12;
NET "prm_ctrl_out[5]" SLEW = SLOW;
NET "prm_ctrl_out[6]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[6]" DRIVE = 12;
NET "prm_ctrl_out[6]" SLEW = SLOW;
NET "prm_ctrl_out[7]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[7]" DRIVE = 12;
NET "prm_ctrl_out[7]" SLEW = SLOW;
NET "prm_ctrl_out[8]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[8]" DRIVE = 12;
NET "prm_ctrl_out[8]" SLEW = SLOW;
NET "prm_ctrl_out[9]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[9]" DRIVE = 12;
NET "prm_ctrl_out[9]" SLEW = SLOW;
NET "prm_ctrl_out[10]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[10]" DRIVE = 12;
NET "prm_ctrl_out[10]" SLEW = SLOW;
NET "prm_ctrl_out[11]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[11]" DRIVE = 12;
NET "prm_ctrl_out[11]" SLEW = SLOW;
NET "prm_ctrl_out[12]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[12]" DRIVE = 12;
NET "prm_ctrl_out[12]" SLEW = SLOW;
NET "prm_ctrl_out[13]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[13]" DRIVE = 12;
NET "prm_ctrl_out[13]" SLEW = SLOW;
NET "prm_ctrl_out[14]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[14]" DRIVE = 12;
NET "prm_ctrl_out[14]" SLEW = SLOW;
NET "prm_ctrl_out[15]" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[15]" DRIVE = 12;
NET "prm_ctrl_out[15]" SLEW = SLOW;
NET "reset_cmds_in[0]" IOSTANDARD = LVTTL;
NET "reset_cmds_in[1]" IOSTANDARD = LVTTL;
NET "reset_cmds_out[0]" IOSTANDARD = LVTTL;
NET "reset_cmds_out[0]" DRIVE = 12;
NET "reset_cmds_out[0]" SLEW = SLOW;
NET "reset_cmds_out[1]" IOSTANDARD = LVTTL;
NET "reset_cmds_out[1]" DRIVE = 12;
NET "reset_cmds_out[1]" SLEW = SLOW;
NET "reset_cmds_out[2]" IOSTANDARD = LVTTL;
NET "reset_cmds_out[2]" DRIVE = 12;
NET "reset_cmds_out[2]" SLEW = SLOW;
NET "reset_cmds_out[3]" IOSTANDARD = LVTTL;
NET "reset_cmds_out[3]" DRIVE = 12;
NET "reset_cmds_out[3]" SLEW = SLOW;
NET "reset_cmds_out[4]" IOSTANDARD = LVTTL;
NET "reset_cmds_out[4]" DRIVE = 12;
NET "reset_cmds_out[4]" SLEW = SLOW;
NET "reset_cmds_out[5]" IOSTANDARD = LVTTL;
NET "reset_cmds_out[5]" DRIVE = 12;
NET "reset_cmds_out[5]" SLEW = SLOW;
NET "reset_cmds_out[6]" IOSTANDARD = LVTTL;
NET "reset_cmds_out[6]" DRIVE = 12;
NET "reset_cmds_out[6]" SLEW = SLOW;
NET "reset_cmds_out[7]" IOSTANDARD = LVTTL;
NET "reset_cmds_out[7]" DRIVE = 12;
NET "reset_cmds_out[7]" SLEW = SLOW;
NET "board_address[0]" IOSTANDARD = LVTTL;
NET "board_address[1]" IOSTANDARD = LVTTL;
NET "board_address[2]" IOSTANDARD = LVTTL;
NET "board_address[3]" IOSTANDARD = LVTTL;
NET "board_address[4]" IOSTANDARD = LVTTL;
NET "board_address[5]" IOSTANDARD = LVTTL;
NET "board_address[6]" IOSTANDARD = LVTTL;
NET "board_address[7]" IOSTANDARD = LVTTL;
NET "jtag_tdi" IOSTANDARD = LVCMOS25;
NET "jtag_tms" IOSTANDARD = LVCMOS25;
NET "jtag_tck" IOSTANDARD = LVCMOS25;
NET "jtag_tdo" IOSTANDARD = LVCMOS25;
NET "jtag_tdo" DRIVE = 12;
NET "jtag_tdo" SLEW = SLOW;
NET "fpga_A_tdi" IOSTANDARD = LVCMOS25;
NET "fpga_A_tdi" DRIVE = 12;
NET "fpga_A_tdi" SLEW = SLOW;
NET "fpga_A_tms" IOSTANDARD = LVCMOS25;
NET "fpga_A_tms" DRIVE = 12;
NET "fpga_A_tms" SLEW = SLOW;
NET "fpga_A_tck" IOSTANDARD = LVCMOS25;
NET "fpga_A_tck" DRIVE = 12;
NET "fpga_A_tck" SLEW = SLOW;
NET "fpga_A_tdo" IOSTANDARD = LVCMOS25;

```

```
NET "fpga_B_tdi" IOSTANDARD = LVCMOS25;
NET "fpga_B_tdi" DRIVE = 12;
NET "fpga_B_tdi" SLEW = SLOW;
NET "fpga_B_tms" IOSTANDARD = LVCMOS25;
NET "fpga_B_tms" DRIVE = 12;
NET "fpga_B_tms" SLEW = SLOW;
NET "fpga_B_tck" IOSTANDARD = LVCMOS25;
NET "fpga_B_tck" DRIVE = 12;
NET "fpga_B_tck" SLEW = SLOW;
NET "fpga_B_tdo" IOSTANDARD = LVCMOS25;
NET "fpga_C_tdi" IOSTANDARD = LVCMOS25;
NET "fpga_C_tdi" DRIVE = 12;
NET "fpga_C_tdi" SLEW = SLOW;
NET "fpga_C_tck" IOSTANDARD = LVCMOS25;
NET "fpga_C_tck" DRIVE = 12;
NET "fpga_C_tck" SLEW = SLOW;
NET "fpga_C_tms" IOSTANDARD = LVCMOS25;
NET "fpga_C_tms" DRIVE = 12;
NET "fpga_C_tms" SLEW = SLOW;
NET "fpga_C_tdo" IOSTANDARD = LVCMOS25;
NET "prm_data[0]" IOSTANDARD = LVCMOS25;
NET "prm_data[0]" DRIVE = 12;
NET "prm_data[0]" SLEW = SLOW;
NET "prm_data[1]" IOSTANDARD = LVCMOS25;
NET "prm_data[1]" DRIVE = 12;
NET "prm_data[1]" SLEW = SLOW;
NET "prm_data[2]" IOSTANDARD = LVCMOS25;
NET "prm_data[2]" DRIVE = 12;
NET "prm_data[2]" SLEW = SLOW;
NET "prm_data[3]" IOSTANDARD = LVCMOS25;
NET "prm_data[3]" DRIVE = 12;
NET "prm_data[3]" SLEW = SLOW;
NET "prm_data[4]" IOSTANDARD = LVCMOS25;
NET "prm_data[4]" DRIVE = 12;
NET "prm_data[4]" SLEW = SLOW;
NET "prm_data[5]" IOSTANDARD = LVCMOS25;
NET "prm_data[5]" DRIVE = 12;
NET "prm_data[5]" SLEW = SLOW;
NET "prm_data[6]" IOSTANDARD = LVCMOS25;
NET "prm_data[6]" DRIVE = 12;
NET "prm_data[6]" SLEW = SLOW;
NET "prm_data[7]" IOSTANDARD = LVCMOS25;
NET "prm_data[7]" DRIVE = 12;
NET "prm_data[7]" SLEW = SLOW;
NET "prm_data[8]" IOSTANDARD = LVCMOS25;
NET "prm_data[8]" DRIVE = 12;
NET "prm_data[8]" SLEW = SLOW;
NET "prm_data[9]" IOSTANDARD = LVCMOS25;
NET "prm_data[9]" DRIVE = 12;
NET "prm_data[9]" SLEW = SLOW;
NET "prm_data[10]" IOSTANDARD = LVCMOS25;
NET "prm_data[10]" DRIVE = 12;
NET "prm_data[10]" SLEW = SLOW;
NET "prm_data[11]" IOSTANDARD = LVCMOS25;
NET "prm_data[11]" DRIVE = 12;
NET "prm_data[11]" SLEW = SLOW;
NET "prm_data[12]" IOSTANDARD = LVCMOS25;
NET "prm_data[12]" DRIVE = 12;
NET "prm_data[12]" SLEW = SLOW;
NET "prm_data[13]" IOSTANDARD = LVCMOS25;
NET "prm_data[13]" DRIVE = 12;
NET "prm_data[13]" SLEW = SLOW;
NET "prm_data[14]" IOSTANDARD = LVCMOS25;
NET "prm_data[14]" DRIVE = 12;
NET "prm_data[14]" SLEW = SLOW;
NET "prm_data[15]" IOSTANDARD = LVCMOS25;
NET "prm_data[15]" DRIVE = 12;
NET "prm_data[15]" SLEW = SLOW;
NET "prm_data[16]" IOSTANDARD = LVCMOS25;
NET "prm_data[16]" DRIVE = 12;
NET "prm_data[16]" SLEW = SLOW;
NET "prm_data[17]" IOSTANDARD = LVCMOS25;
NET "prm_data[17]" DRIVE = 12;
NET "prm_data[17]" SLEW = SLOW;
NET "prm_data[18]" IOSTANDARD = LVCMOS25;
NET "prm_data[18]" DRIVE = 12;
NET "prm_data[18]" SLEW = SLOW;
NET "prm_data[19]" IOSTANDARD = LVCMOS25;
NET "prm_data[19]" DRIVE = 12;
NET "prm_data[19]" SLEW = SLOW;
NET "prm_data[20]" IOSTANDARD = LVCMOS25;
NET "prm_data[20]" DRIVE = 12;
NET "prm_data[20]" SLEW = SLOW;
NET "prm_data[21]" IOSTANDARD = LVCMOS25;
NET "prm_data[21]" DRIVE = 12;
NET "prm_data[21]" SLEW = SLOW;
NET "prm_data[22]" IOSTANDARD = LVCMOS25;
NET "prm_data[22]" DRIVE = 12;
NET "prm_data[22]" SLEW = SLOW;
NET "prm_data[23]" IOSTANDARD = LVCMOS25;
NET "prm_data[23]" DRIVE = 12;
NET "prm_data[23]" SLEW = SLOW;
NET "prm_data[24]" IOSTANDARD = LVCMOS25;
NET "prm_data[24]" DRIVE = 12;
NET "prm_data[24]" SLEW = SLOW;
NET "prm_data[25]" IOSTANDARD = LVCMOS25;
NET "prm_data[25]" DRIVE = 12;
NET "prm_data[25]" SLEW = SLOW;
NET "prm_data[26]" IOSTANDARD = LVCMOS25;
NET "prm_data[26]" DRIVE = 12;
NET "prm_data[26]" SLEW = SLOW;
NET "prm_data[27]" IOSTANDARD = LVCMOS25;
NET "prm_data[27]" DRIVE = 12;
NET "prm_data[27]" SLEW = SLOW;
NET "prm_data[28]" IOSTANDARD = LVCMOS25;
NET "prm_data[28]" DRIVE = 12;
NET "prm_data[28]" SLEW = SLOW;
NET "prm_data[29]" IOSTANDARD = LVCMOS25;
NET "prm_data[29]" DRIVE = 12;
NET "prm_data[29]" SLEW = SLOW;
NET "prm_data[30]" IOSTANDARD = LVCMOS25;
NET "prm_data[30]" DRIVE = 12;
NET "prm_data[30]" SLEW = SLOW;
NET "prm_data[31]" IOSTANDARD = LVCMOS25;
```

```
NET "prm_data[31]" DRIVE = 12;
NET "prm_data[31]" SLEW = SLOW;
NET "prm_ctrl_in[0]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[1]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[2]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[3]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[4]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[5]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[6]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[7]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[8]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[9]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[10]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[11]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[12]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[13]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[14]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[15]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[16]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[17]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[18]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[19]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[20]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[21]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[22]" IOSTANDARD = LVCMOS25;
NET "prm_strobe[0]" IOSTANDARD = LVCMOS25;
NET "prm_strobe[0]" DRIVE = 12;
NET "prm_strobe[0]" SLEW = SLOW;
NET "prm_strobe[1]" IOSTANDARD = LVCMOS25;
NET "prm_strobe[1]" DRIVE = 12;
NET "prm_strobe[1]" SLEW = SLOW;
NET "mdsp_hd[0]" IOSTANDARD = LVTTL;
NET "mdsp_hd[0]" DRIVE = 12;
NET "mdsp_hd[0]" SLEW = SLOW;
NET "mdsp_hd[1]" IOSTANDARD = LVTTL;
NET "mdsp_hd[1]" DRIVE = 12;
NET "mdsp_hd[1]" SLEW = SLOW;
NET "mdsp_hd[2]" IOSTANDARD = LVTTL;
NET "mdsp_hd[2]" DRIVE = 12;
NET "mdsp_hd[2]" SLEW = SLOW;
NET "mdsp_hd[3]" IOSTANDARD = LVTTL;
NET "mdsp_hd[3]" DRIVE = 12;
NET "mdsp_hd[3]" SLEW = SLOW;
NET "mdsp_hd[4]" IOSTANDARD = LVTTL;
NET "mdsp_hd[4]" DRIVE = 12;
NET "mdsp_hd[4]" SLEW = SLOW;
NET "mdsp_hd[5]" IOSTANDARD = LVTTL;
NET "mdsp_hd[5]" DRIVE = 12;
NET "mdsp_hd[5]" SLEW = SLOW;
NET "mdsp_hd[6]" IOSTANDARD = LVTTL;
NET "mdsp_hd[6]" DRIVE = 12;
NET "mdsp_hd[6]" SLEW = SLOW;
NET "mdsp_hd[7]" IOSTANDARD = LVTTL;
NET "mdsp_hd[7]" DRIVE = 12;
NET "mdsp_hd[7]" SLEW = SLOW;
NET "mdsp_hd[8]" IOSTANDARD = LVTTL;
NET "mdsp_hd[8]" DRIVE = 12;
NET "mdsp_hd[8]" SLEW = SLOW;
NET "mdsp_hd[9]" IOSTANDARD = LVTTL;
NET "mdsp_hd[9]" DRIVE = 12;
NET "mdsp_hd[9]" SLEW = SLOW;
NET "mdsp_hd[10]" IOSTANDARD = LVTTL;
NET "mdsp_hd[10]" DRIVE = 12;
NET "mdsp_hd[10]" SLEW = SLOW;
NET "mdsp_hd[11]" IOSTANDARD = LVTTL;
NET "mdsp_hd[11]" DRIVE = 12;
NET "mdsp_hd[11]" SLEW = SLOW;
NET "mdsp_hd[12]" IOSTANDARD = LVTTL;
NET "mdsp_hd[12]" DRIVE = 12;
NET "mdsp_hd[12]" SLEW = SLOW;
NET "mdsp_hd[13]" IOSTANDARD = LVTTL;
NET "mdsp_hd[13]" DRIVE = 12;
NET "mdsp_hd[13]" SLEW = SLOW;
NET "mdsp_hd[14]" IOSTANDARD = LVTTL;
NET "mdsp_hd[14]" DRIVE = 12;
NET "mdsp_hd[14]" SLEW = SLOW;
NET "mdsp_hd[15]" IOSTANDARD = LVTTL;
NET "mdsp_hd[15]" DRIVE = 12;
NET "mdsp_hd[15]" SLEW = SLOW;
NET "mdsp_bootmode[0]" IOSTANDARD = LVTTL;
NET "mdsp_bootmode[0]" DRIVE = 12;
NET "mdsp_bootmode[0]" SLEW = SLOW;
NET "mdsp_bootmode[1]" IOSTANDARD = LVTTL;
NET "mdsp_bootmode[1]" DRIVE = 12;
NET "mdsp_bootmode[1]" SLEW = SLOW;
NET "mdsp_bootmode[2]" IOSTANDARD = LVTTL;
NET "mdsp_bootmode[2]" DRIVE = 12;
NET "mdsp_bootmode[2]" SLEW = SLOW;
NET "mdsp_bootmode[3]" IOSTANDARD = LVTTL;
NET "mdsp_bootmode[3]" DRIVE = 12;
NET "mdsp_bootmode[3]" SLEW = SLOW;
NET "mdsp_bootmode[4]" IOSTANDARD = LVTTL;
NET "mdsp_bootmode[4]" DRIVE = 12;
NET "mdsp_bootmode[4]" SLEW = SLOW;
NET "mdsp_hcntl0" IOSTANDARD = LVTTL;
NET "mdsp_hcntl0" DRIVE = 12;
NET "mdsp_hcntl0" SLEW = SLOW;
NET "mdsp_hcntl1" IOSTANDARD = LVTTL;
NET "mdsp_hcntl1" DRIVE = 12;
NET "mdsp_hcntl1" SLEW = SLOW;
NET "mdsp_hcs_n" IOSTANDARD = LVTTL;
NET "mdsp_hcs_n" DRIVE = 12;
NET "mdsp_hcs_n" SLEW = SLOW;
NET "mdsp_hds1_n" IOSTANDARD = LVTTL;
NET "mdsp_hds1_n" DRIVE = 12;
NET "mdsp_hds1_n" SLEW = SLOW;
NET "mdsp_hhwil" IOSTANDARD = LVTTL;
NET "mdsp_hhwil" DRIVE = 12;
NET "mdsp_hhwil" SLEW = SLOW;
NET "mdsp_hrdy_n" IOSTANDARD = LVTTL;
NET "mdsp_hrnw" IOSTANDARD = LVTTL;
NET "mdsp_hrnw" DRIVE = 12;
NET "mdsp_hrnw" SLEW = SLOW;
NET "mdsp_reset" IOSTANDARD = LVTTL;
```

```
NET "mdsp_reset" DRIVE = 12;
NET "mdsp_reset" SLEW = SLOW;
NET "vme_data[0]" IOSTANDARD = LVTTL;
NET "vme_data[0]" DRIVE = 12;
NET "vme_data[0]" SLEW = SLOW;
NET "vme_data[1]" IOSTANDARD = LVTTL;
NET "vme_data[1]" DRIVE = 12;
NET "vme_data[1]" SLEW = SLOW;
NET "vme_data[2]" IOSTANDARD = LVTTL;
NET "vme_data[2]" DRIVE = 12;
NET "vme_data[2]" SLEW = SLOW;
NET "vme_data[3]" IOSTANDARD = LVTTL;
NET "vme_data[3]" DRIVE = 12;
NET "vme_data[3]" SLEW = SLOW;
NET "vme_data[4]" IOSTANDARD = LVTTL;
NET "vme_data[4]" DRIVE = 12;
NET "vme_data[4]" SLEW = SLOW;
NET "vme_data[5]" IOSTANDARD = LVTTL;
NET "vme_data[5]" DRIVE = 12;
NET "vme_data[5]" SLEW = SLOW;
NET "vme_data[6]" IOSTANDARD = LVTTL;
NET "vme_data[6]" DRIVE = 12;
NET "vme_data[6]" SLEW = SLOW;
NET "vme_data[7]" IOSTANDARD = LVTTL;
NET "vme_data[7]" DRIVE = 12;
NET "vme_data[7]" SLEW = SLOW;
NET "vme_data[8]" IOSTANDARD = LVTTL;
NET "vme_data[8]" DRIVE = 12;
NET "vme_data[8]" SLEW = SLOW;
NET "vme_data[9]" IOSTANDARD = LVTTL;
NET "vme_data[9]" DRIVE = 12;
NET "vme_data[9]" SLEW = SLOW;
NET "vme_data[10]" IOSTANDARD = LVTTL;
NET "vme_data[10]" DRIVE = 12;
NET "vme_data[10]" SLEW = SLOW;
NET "vme_data[11]" IOSTANDARD = LVTTL;
NET "vme_data[11]" DRIVE = 12;
NET "vme_data[11]" SLEW = SLOW;
NET "vme_data[12]" IOSTANDARD = LVTTL;
NET "vme_data[12]" DRIVE = 12;
NET "vme_data[12]" SLEW = SLOW;
NET "vme_data[13]" IOSTANDARD = LVTTL;
NET "vme_data[13]" DRIVE = 12;
NET "vme_data[13]" SLEW = SLOW;
NET "vme_data[14]" IOSTANDARD = LVTTL;
NET "vme_data[14]" DRIVE = 12;
NET "vme_data[14]" SLEW = SLOW;
NET "vme_data[15]" IOSTANDARD = LVTTL;
NET "vme_data[15]" DRIVE = 12;
NET "vme_data[15]" SLEW = SLOW;
NET "vme_data[16]" IOSTANDARD = LVTTL;
NET "vme_data[16]" DRIVE = 12;
NET "vme_data[16]" SLEW = SLOW;
NET "vme_data[17]" IOSTANDARD = LVTTL;
NET "vme_data[17]" DRIVE = 12;
NET "vme_data[17]" SLEW = SLOW;
NET "vme_data[18]" IOSTANDARD = LVTTL;
NET "vme_data[18]" DRIVE = 12;
NET "vme_data[18]" SLEW = SLOW;
NET "vme_data[19]" IOSTANDARD = LVTTL;
NET "vme_data[19]" DRIVE = 12;
NET "vme_data[19]" SLEW = SLOW;
NET "vme_data[20]" IOSTANDARD = LVTTL;
NET "vme_data[20]" DRIVE = 12;
NET "vme_data[20]" SLEW = SLOW;
NET "vme_data[21]" IOSTANDARD = LVTTL;
NET "vme_data[21]" DRIVE = 12;
NET "vme_data[21]" SLEW = SLOW;
NET "vme_data[22]" IOSTANDARD = LVTTL;
NET "vme_data[22]" DRIVE = 12;
NET "vme_data[22]" SLEW = SLOW;
NET "vme_data[23]" IOSTANDARD = LVTTL;
NET "vme_data[23]" DRIVE = 12;
NET "vme_data[23]" SLEW = SLOW;
NET "vme_data[24]" IOSTANDARD = LVTTL;
NET "vme_data[24]" DRIVE = 12;
NET "vme_data[24]" SLEW = SLOW;
NET "vme_data[25]" IOSTANDARD = LVTTL;
NET "vme_data[25]" DRIVE = 12;
NET "vme_data[25]" SLEW = SLOW;
NET "vme_data[26]" IOSTANDARD = LVTTL;
NET "vme_data[26]" DRIVE = 12;
NET "vme_data[26]" SLEW = SLOW;
NET "vme_data[27]" IOSTANDARD = LVTTL;
NET "vme_data[27]" DRIVE = 12;
NET "vme_data[27]" SLEW = SLOW;
NET "vme_data[28]" IOSTANDARD = LVTTL;
NET "vme_data[28]" DRIVE = 12;
NET "vme_data[28]" SLEW = SLOW;
NET "vme_data[29]" IOSTANDARD = LVTTL;
NET "vme_data[29]" DRIVE = 12;
NET "vme_data[29]" SLEW = SLOW;
NET "vme_data[30]" IOSTANDARD = LVTTL;
NET "vme_data[30]" DRIVE = 12;
NET "vme_data[30]" SLEW = SLOW;
NET "vme_data[31]" IOSTANDARD = LVTTL;
NET "vme_data[31]" DRIVE = 12;
NET "vme_data[31]" SLEW = SLOW;
NET "vme_addr[0]" IOSTANDARD = LVTTL;
NET "vme_addr[1]" IOSTANDARD = LVTTL;
NET "vme_addr[2]" IOSTANDARD = LVTTL;
NET "vme_addr[3]" IOSTANDARD = LVTTL;
NET "vme_addr[4]" IOSTANDARD = LVTTL;
NET "vme_addr[5]" IOSTANDARD = LVTTL;
NET "vme_addr[6]" IOSTANDARD = LVTTL;
NET "vme_addr[7]" IOSTANDARD = LVTTL;
NET "vme_addr[8]" IOSTANDARD = LVTTL;
NET "vme_addr[9]" IOSTANDARD = LVTTL;
NET "vme_addr[10]" IOSTANDARD = LVTTL;
NET "vme_addr[11]" IOSTANDARD = LVTTL;
NET "vme_addr[12]" IOSTANDARD = LVTTL;
NET "vme_addr[13]" IOSTANDARD = LVTTL;
NET "vme_addr[14]" IOSTANDARD = LVTTL;
NET "vme_addr[15]" IOSTANDARD = LVTTL;
```

```
NET "vme_addr[16]" IOSTANDARD = LVTTL;
NET "vme_addr[17]" IOSTANDARD = LVTTL;
NET "vme_addr[18]" IOSTANDARD = LVTTL;
NET "vme_addr[19]" IOSTANDARD = LVTTL;
NET "vme_addr[20]" IOSTANDARD = LVTTL;
NET "vme_addr[21]" IOSTANDARD = LVTTL;
NET "vme_addr[22]" IOSTANDARD = LVTTL;
NET "vme_addr[23]" IOSTANDARD = LVTTL;
NET "vme_addr[24]" IOSTANDARD = LVTTL;
NET "vme_addr[25]" IOSTANDARD = LVTTL;
NET "vme_addr[26]" IOSTANDARD = LVTTL;
NET "vme_addr[27]" IOSTANDARD = LVTTL;
NET "vme_addr[28]" IOSTANDARD = LVTTL;
NET "vme_addr[29]" IOSTANDARD = LVTTL;
NET "vme_addr[30]" IOSTANDARD = LVTTL;
NET "vme_addr[31]" IOSTANDARD = LVTTL;
NET "vme_ga_n[0]" IOSTANDARD = LVTTL;
NET "vme_ga_n[1]" IOSTANDARD = LVTTL;
NET "vme_ga_n[2]" IOSTANDARD = LVTTL;
NET "vme_ga_n[3]" IOSTANDARD = LVTTL;
NET "vme_ga_n[4]" IOSTANDARD = LVTTL;
NET "vme_am[0]" IOSTANDARD = LVTTL;
NET "vme_am[1]" IOSTANDARD = LVTTL;
NET "vme_am[2]" IOSTANDARD = LVTTL;
NET "vme_am[3]" IOSTANDARD = LVTTL;
NET "vme_am[4]" IOSTANDARD = LVTTL;
NET "vme_am[5]" IOSTANDARD = LVTTL;
NET "vme_irq[0]" IOSTANDARD = LVTTL;
NET "vme_irq[0]" DRIVE = 12;
NET "vme_irq[0]" SLEW = SLOW;
NET "vme_irq[1]" IOSTANDARD = LVTTL;
NET "vme_irq[1]" DRIVE = 12;
NET "vme_irq[1]" SLEW = SLOW;
NET "vme_irq[2]" IOSTANDARD = LVTTL;
NET "vme_irq[2]" DRIVE = 12;
NET "vme_irq[2]" SLEW = SLOW;

NET "vme_as" IOSTANDARD = LVTTL;
NET "vme_berr" IOSTANDARD = LVTTL;
NET "vme_berr" DRIVE = 12;
NET "vme_berr" SLEW = SLOW;
NET "vme_ds0" IOSTANDARD = LVTTL;
NET "vme_ds1" IOSTANDARD = LVTTL;
NET "vme_dtack" IOSTANDARD = LVTTL;
NET "vme_dtack" DRIVE = 12;
NET "vme_dtack" SLEW = SLOW;
NET "vme_gap_n" IOSTANDARD = LVTTL;
NET "vme_iack_n" IOSTANDARD = LVTTL;
NET "vme_wr" IOSTANDARD = LVTTL;
NET "vme_reset" IOSTANDARD = LVTTL;
NET "vme_sysfail" IOSTANDARD = LVTTL;
NET "vme_iackin" IOSTANDARD = LVTTL;
NET "viackout" IOSTANDARD = LVTTL;
NET "viackout" DRIVE = 12;
NET "viackout" SLEW = SLOW;
NET "vme_dir" IOSTANDARD = LVTTL;
NET "vme_dir" DRIVE = 12;
NET "vme_dir" SLEW = SLOW;
NET "vme_oe" IOSTANDARD = LVTTL;
NET "vme_oe" DRIVE = 12;
NET "vme_oe" SLEW = SLOW;
NET "vme_lword" IOSTANDARD = LVTTL;
NET "vme_irq_n" IOSTANDARD = LVTTL;
NET "vme_irq_n" DRIVE = 12;
NET "vme_irq_n" SLEW = SLOW;
NET "done_s6a" LOC = W20;
NET "done_s6b" LOC = P20;
NET "done_virtex" LOC = W22;
NET "lattice_tdi" LOC = AB14;
NET "lattice_tms" LOC = AB15;
NET "lattice_tdo" LOC = AA16;
NET "lattice_tck" LOC = AB17
NET "lattice_**" IOSTANDARD = LVCMOS33;
```

S6A-S6B pin assignment (UCF file)

```

CONFIG PROHIBIT = AG25;
CONFIG PROHIBIT = AK20;
CONFIG PROHIBIT = AH18;
CONFIG PROHIBIT = AK18;
CONFIG PROHIBIT = AK13;
CONFIG PROHIBIT = AK11;
CONFIG PROHIBIT = AE8;
CONFIG PROHIBIT = AJ26;
CONFIG PROHIBIT = AH25;
NET "eth_rxclk" LOC = A16;
NET "eth_txclk" LOC = C16;
NET "fe_emu_clk" LOC = C15;
CONFIG PROHIBIT = AK26;
CONFIG PROHIBIT = AK25;
CONFIG PROHIBIT = AJ22;
CONFIG PROHIBIT = AJ6;
CONFIG PROHIBIT = AK6;
NET "rd[0]" LOC = AG17;
NET "rd[1]" LOC = AG19;
NET "rd[2]" LOC = AE18;
NET "rd[3]" LOC = AE19;
NET "rd[4]" LOC = AG21;
NET "rd[5]" LOC = AH22;
NET "rd[6]" LOC = AK22;
NET "rd[7]" LOC = AK24;
NET "rd[8]" LOC = AG23;
NET "rd[9]" LOC = AE22;
NET "rd[10]" LOC = AH24;
NET "rd[11]" LOC = AE24;
NET "rd[12]" LOC = AG6;
NET "rd[13]" LOC = AD8;
NET "rd[14]" LOC = AF7;
NET "rd[15]" LOC = AF9;
NET "rd[16]" LOC = AG10;
NET "rd[17]" LOC = AJ10;
NET "rd[18]" LOC = AF11;
NET "rd[19]" LOC = AJ12;
NET "rd[20]" LOC = AD12;
NET "rd[21]" LOC = AG12;
NET "rd[22]" LOC = AF13;
NET "rd[23]" LOC = AH13;
NET "rd[24]" LOC = AJ14;
NET "rd[25]" LOC = AH15;
NET "rd[26]" LOC = AD15;
NET "rd[27]" LOC = AF15;
NET "rd[28]" LOC = AD16;
NET "rd[29]" LOC = AF17;
NET "rd[30]" LOC = AG18;
NET "rd[31]" LOC = AF19;
NET "rd[32]" LOC = AD18;
NET "rd[33]" LOC = AD19;
NET "rd[34]" LOC = AF21;
NET "rd[35]" LOC = AG22;
NET "rd[36]" LOC = AJ24;
NET "rd[37]" LOC = AF23;
NET "rd[38]" LOC = AD22;
NET "rd[39]" LOC = AG24;
NET "rd[40]" LOC = AF25;
NET "rd[41]" LOC = AD24;
NET "rd[42]" LOC = AE16;
NET "rd[43]" LOC = AG15;
NET "rd[44]" LOC = AE15;
NET "rd[45]" LOC = AK15;
NET "rd[46]" LOC = AK14;
NET "rd[47]" LOC = AG13;
NET "sl_bad_0" LOC = AG8;
NET "sl_bad_1" LOC = AD9;

NET "sl_data[0]" LOC = AH12;
NET "sl_data[1]" LOC = AE12;
NET "sl_data[2]" LOC = AK12;
NET "sl_data[3]" LOC = AG11;
NET "sl_data[4]" LOC = AK10;
NET "sl_data[5]" LOC = AH10;
NET "sl_data[6]" LOC = AG9;
NET "sl_data[7]" LOC = AG7;
NET "sl_data[8]" LOC = AH6;
NET "sl_data[9]" LOC = AD23;
NET "sl_data[10]" LOC = AC21;
NET "sl_data[11]" LOC = AD20;
NET "sl_data[12]" LOC = AH23;
NET "sl_data[13]" LOC = AH21;
NET "sl_data[14]" LOC = AJ20;
NET "sl_data[15]" LOC = AB20;
NET "sl_data[16]" LOC = AG20;
NET "sl_data[17]" LOC = AA19;
NET "sl_data[18]" LOC = AB18;
NET "sl_data[19]" LOC = AD17;
NET "sl_data[20]" LOC = AB16;
NET "sl_data[21]" LOC = AG16;
NET "sl_data[22]" LOC = AA15;
NET "sl_data[23]" LOC = AB14;
NET "sl_data[24]" LOC = AD14;
NET "sl_data[25]" LOC = AG14;
NET "sl_data[26]" LOC = AD13;
NET "sl_data[27]" LOC = AH11;
NET "sl_data[28]" LOC = AB12;
NET "sl_data[29]" LOC = AB11;
NET "sl_data[30]" LOC = AH9;
NET "sl_data[31]" LOC = AD10;
NET "sl_rst_out_0" LOC = AJ8;
NET "sl_rst_out_1" LOC = AH7;
NET "sl_test_out_0" LOC = AE23;
NET "sl_test_out_1" LOC = AD21;
NET "sl_uctrl_out_0" LOC = AE20;
NET "sl_uctrl_out_1" LOC = AK23;
NET "sl_wen_out_0" LOC = AK21;
NET "sl_wen_out_1" LOC = AC20;
NET "sl_xoff_0" LOC = AH20;
NET "sl_xoff_1" LOC = AB19;
NET "xc[0]" LOC = AC18;
NET "xc[1]" LOC = AE17;
NET "xc[2]" LOC = AC16;
NET "xc[3]" LOC = AH16;
NET "xc[4]" LOC = AB15;
NET "xc[5]" LOC = AC14;
NET "xc[6]" LOC = AE14;
NET "xc[7]" LOC = AH14;
NET "ck1_40_N" LOC = A15;
NET "ck1_40_P" LOC = B15;
NET "ck2_40_N" LOC = V27;
NET "ck2_40_P" LOC = V26;
NET "form_ds" LOC = A29;
NET "form_hwob" LOC = B30;
CONFIG PROHIBIT = H6;
CONFIG PROHIBIT = L6;
NET "efb_edm_fifo_ef" LOC = V23;
NET "efb_edm_fifo_empty_error" LOC = AD26;
NET "efb_edm_fifo_ff" LOC = V24;
NET "efb_event_dynamic_mask[0]" LOC = W27;
NET "efb_event_dynamic_mask[1]" LOC = V30;
NET "efb_event_dynamic_mask[2]" LOC = V28;
NET "efb_event_dynamic_mask[3]" LOC = U28;
NET "efb_event_dynamic_mask[4]" LOC = U27;
NET "efb_event_dynamic_mask[5]" LOC = U30;

```

```

NET "efb_event_dynamic_mask[6]" LOC = U29;
NET "efb_event_dynamic_mask[7]" LOC = T30;
NET "efb_event_dynamic_mask[8]" LOC = T28;
NET "efb_event_dynamic_mask[9]" LOC = T27;
NET "efb_event_dynamic_mask[10]" LOC = T26;
NET "efb_event_dynamic_mask[11]" LOC = R28;
NET "efb_event_dynamic_mask[12]" LOC = R27;
NET "efb_event_dynamic_mask[13]" LOC = R30;
NET "efb_event_dynamic_mask[14]" LOC = R29;
NET "efb_event_dynamic_mask[15]" LOC = P27;
NET "efb_event_dynamic_mask_fifo_rst_n" LOC = AD27;
NET "efb_event_dynamic_mask_fifo_we" LOC = W21;
NET "fe_cmd_pulse_form" LOC = AJ28;
NET "form_mb_fifo_ef" LOC = AK28;
NET "form_mb_fifo_f" LOC = U24;
NET "form_mode_bits[0]" LOC = P26;
NET "form_mode_bits[1]" LOC = P30;
NET "form_mode_bits[2]" LOC = P28;
NET "form_mode_bits[3]" LOC = N28;
NET "form_mode_bits[4]" LOC = N27;
NET "form_mode_bits[5]" LOC = N30;
NET "form_mode_bits[6]" LOC = N29;
NET "form_mode_bits[7]" LOC = M24;
NET "form_mode_bits[8]" LOC = M23;
NET "form_mode_bits[9]" LOC = W28;
NET "form_mode_bits[10]" LOC = AB28;
NET "form_mode_bits[11]" LOC = AG30;
NET "form_mode_bits_rst_n" LOC = U25;
NET "form_mode_bits_we" LOC = W22;
NET "form_trailer_info[0]" LOC = W29;
NET "form_trailer_info[1]" LOC = W30;
NET "form_trailer_info[2]" LOC = Y28;
NET "form_trailer_info[3]" LOC = Y30;
NET "form_trailer_info[4]" LOC = AA29;
NET "form_trailer_info[5]" LOC = AA30;
NET "form_trailer_info[6]" LOC = AA27;
NET "form_trailer_info[7]" LOC = AA28;
NET "form_trailer_info[8]" LOC = Y26;
NET "form_trailer_info[9]" LOC = Y27;
NET "form_trailer_info[10]" LOC = AD28;
NET "form_trailer_info[11]" LOC = AD30;
NET "header_trailer_limit" LOC = AG27;
NET "mcb3_dram_a[0]" LOC = D4;
NET "mcb3_dram_a[1]" LOC = D3;
NET "mcb3_dram_a[2]" LOC = A3;
NET "mcb3_dram_a[3]" LOC = E5;
NET "mcb3_dram_a[4]" LOC = C5;
NET "mcb3_dram_a[5]" LOC = F2;
NET "mcb3_dram_a[6]" LOC = F1;
NET "mcb3_dram_a[7]" LOC = B3;
NET "mcb3_dram_a[8]" LOC = B2;
NET "mcb3_dram_a[9]" LOC = A2;
NET "mcb3_dram_a[10]" LOC = D5;
NET "mcb3_dram_a[11]" LOC = G4;
NET "mcb3_dram_a[12]" LOC = A4;
NET "mcb3_dram_a[13]" LOC = B5;
NET "mcb3_dram_ba[0]" LOC = D2;
NET "mcb3_dram_ba[1]" LOC = D1;
NET "mcb3_dram_ba[2]" LOC = F3;
NET "mcb3_dram_cas_n" LOC = B1;
NET "mcb3_dram_ck_N" LOC = E1;
NET "mcb3_dram_ck_P" LOC = E3;
NET "mcb3_dram_cke" LOC = C4;
NET "mcb3_dram_dm" LOC = K3;
NET "mcb3_dram_dq[0]" LOC = H4;
NET "mcb3_dram_dq[1]" LOC = H3;
NET "mcb3_dram_dq[2]" LOC = J3;
NET "mcb3_dram_dq[3]" LOC = J1;
NET "mcb3_dram_dq[4]" LOC = G3;
NET "mcb3_dram_dq[5]" LOC = G1;
NET "mcb3_dram_dq[6]" LOC = H2;

NET "mcb3_dram_dq[7]" LOC = H1;
NET "mcb3_dram_dq[8]" LOC = M4;
NET "mcb3_dram_dq[9]" LOC = M3;
NET "mcb3_dram_dq[10]" LOC = L5;
NET "mcb3_dram_dq[11]" LOC = L4;
NET "mcb3_dram_dq[12]" LOC = L3;
NET "mcb3_dram_dq[13]" LOC = L1;
NET "mcb3_dram_dq[14]" LOC = M2;
NET "mcb3_dram_dq[15]" LOC = M1;
NET "mcb3_dram_dqs_N" LOC = J4;
NET "mcb3_dram_dqs_P" LOC = J5;
NET "mcb3_dram_odi" LOC = E4;
NET "mcb3_dram_ras_n" LOC = C1;
NET "mcb3_dram_udm" LOC = K4;
NET "mcb3_dram_udqs_N" LOC = K1;
NET "mcb3_dram_udqs_P" LOC = K2;
NET "mcb3_dram_we_n" LOC = F4;
NET "mcb3_rzq" LOC = L7;
NET "mcb3_zio" LOC = J6;
NET "rod_bus_ack" LOC = L25;
NET "rod_bus_addr[0]" LOC = F30;
NET "rod_bus_addr[1]" LOC = F28;
NET "rod_bus_addr[2]" LOC = G28;
NET "rod_bus_addr[3]" LOC = G27;
NET "rod_bus_addr[4]" LOC = J28;
NET "rod_bus_addr[5]" LOC = J27;
NET "rod_bus_addr[6]" LOC = K27;
NET "rod_bus_addr[7]" LOC = K26;
NET "rod_bus_addr[8]" LOC = H27;
NET "rod_bus_addr[9]" LOC = H26;
NET "rod_bus_addr[10]" LOC = E30;
NET "rod_bus_addr[11]" LOC = E29;
NET "rod_bus_addr[12]" LOC = E28;
NET "rod_bus_addr[13]" LOC = E27;
NET "rod_bus_addr[14]" LOC = D30;
NET "rod_bus_addr[15]" LOC = D28;
NET "rod_bus_addr[16]" LOC = C27;
NET "rod_bus_addr[17]" LOC = D27;
NET "rod_bus_ce" LOC = C30;
NET "rod_bus_data[0]" LOC = M30;
NET "rod_bus_data[1]" LOC = M28;
NET "rod_bus_data[2]" LOC = M27;
NET "rod_bus_data[3]" LOC = M26;
NET "rod_bus_data[4]" LOC = K30;
NET "rod_bus_data[5]" LOC = K28;
NET "rod_bus_data[6]" LOC = L30;
NET "rod_bus_data[7]" LOC = L29;
NET "rod_bus_data[8]" LOC = L28;
NET "rod_bus_data[9]" LOC = L27;
NET "rod_bus_data[10]" LOC = H30;
NET "rod_bus_data[11]" LOC = H28;
NET "rod_bus_data[12]" LOC = G30;
NET "rod_bus_data[13]" LOC = G29;
NET "rod_bus_data[14]" LOC = J30;
NET "rod_bus_data[15]" LOC = J29;
NET "rod_bus_rnw" LOC = C29;
NET "rod_busy" LOC = AG28;
NET "show_trailer_flags" LOC = AK27;
NET "spare_from_v5[0]" LOC = P24;
NET "spare_from_v5[1]" LOC = P25;
NET "spare_from_v5[2]" LOC = AH30;
NET "spare_from_v5[3]" LOC = AJ30;
NET "spare_from_v5[4]" LOC = R24;
NET "spare_from_v5[5]" LOC = R25;
NET "spare_from_v5[6]" LOC = AJ29;
NET "spare_from_v5[7]" LOC = AK29;
NET "spare_from_v5[8]" LOC = T24;
NET "spare_from_v5[9]" LOC = T25;
NET "spare_to_v5[0]" LOC = W24;
NET "spare_to_v5[1]" LOC = W25;
NET "spare_to_v5[2]" LOC = R21;

```

```

NET "spare_to_v5[3]" LOC = R22;
NET "spare_to_v5[4]" LOC = AF28;
NET "spare_to_v5[5]" LOC = AF30;
NET "spare_to_v5[6]" LOC = P22;
NET "spare_to_v5[7]" LOC = P23;
NET "spare_to_v5[8]" LOC = AG29;
NET "xc_in[0]" LOC = AC27;
NET "xc_in[1]" LOC = AC28;
NET "xc_in[2]" LOC = AC29;
NET "xc_in[3]" LOC = AC30;
NET "xc_in[4]" LOC = AE29;
NET "xc_in[5]" LOC = AE30;
NET "xc_in[6]" LOC = AE27;
NET "xc_in[7]" LOC = AE28;
NET "eth_col" LOC = A6;
NET "eth_crs" LOC = A7;
NET "eth_gtxclk" LOC = A8;
NET "eth_int" LOC = A9;
NET "eth_mdc" LOC = A10;
NET "eth_mdio" LOC = A11;
NET "eth_res" LOC = A12;
NET "eth_rxd[0]" LOC = B9;
NET "eth_rxd[1]" LOC = C9;
NET "eth_rxd[2]" LOC = D9;
NET "eth_rxd[3]" LOC = C10;
NET "eth_rxd[4]" LOC = D10;
NET "eth_rxd[5]" LOC = E10;
NET "eth_rxd[6]" LOC = C11;
NET "eth_rxd[7]" LOC = D11;
NET "eth_rxv" LOC = A13;
NET "eth_rxer" LOC = A14;
NET "eth_txd[0]" LOC = C12;
NET "eth_txd[1]" LOC = D12;
NET "eth_txd[2]" LOC = E12;
NET "eth_txd[3]" LOC = C13;
NET "eth_txd[4]" LOC = D13;
NET "eth_txd[5]" LOC = C14;
NET "eth_txd[6]" LOC = D14;
NET "eth_txd[7]" LOC = E14;
NET "eth_txen" LOC = B11;
NET "eth_txer" LOC = B13;
NET "fe_emu_in[0]" LOC = G13;
NET "fe_emu_in[1]" LOC = G15;
NET "fe_emu_in[2]" LOC = C7;
NET "fe_emu_in[3]" LOC = F9;
NET "fe_emu_in[4]" LOC = F6;
NET "fe_emu_in[5]" LOC = D6;
NET "fe_emu_in[6]" LOC = G10;
NET "fe_emu_in[7]" LOC = H11;
NET "fe_emu_in[8]" LOC = J12;
NET "fe_emu_in[9]" LOC = H13;
NET "fe_emu_in[10]" LOC = J14;
NET "fe_emu_in[11]" LOC = F14;
NET "fe_emu_in[12]" LOC = H15;
NET "fe_emu_in[13]" LOC = F16;
NET "fe_emu_in[14]" LOC = F8;
NET "fe_emu_in[15]" LOC = G6;
NET "fe_emu_in[16]" LOC = E6;
NET "fe_emu_in[17]" LOC = J10;
NET "fe_emu_in[18]" LOC = B7;
NET "fe_emu_in[19]" LOC = J11;
NET "fe_emu_in[20]" LOC = C8;
NET "fe_emu_in[21]" LOC = K12;
NET "fe_emu_in[22]" LOC = J13;
NET "fe_emu_in[23]" LOC = K14;
NET "fe_emu_in[24]" LOC = G14;
NET "fe_emu_in[25]" LOC = J15;
NET "fe_emu_in[26]" LOC = D15;
NET "fe_emu_in[27]" LOC = G16;
NET "fe_emu_in[28]" LOC = D7;
NET "fe_emu_in[29]" LOC = C6;

NET "fe_emu_in[30]" LOC = E8;
NET "fe_emu_in[31]" LOC = G8;
NET "fe_emu_in[32]" LOC = G7;
NET "fe_emu_in[33]" LOC = D16;
NET "fe_emu_in[34]" LOC = F15;
NET "fe_emu_in[35]" LOC = F13;
NET "fe_emu_in[36]" LOC = F11;
NET "fe_emu_out[0]" LOC = G11;
NET "fe_emu_out[1]" LOC = G12;
NET "s6_from_s6[0]" LOC = P3;
NET "s6_from_s6[1]" LOC = R3;
NET "s6_from_s6[2]" LOC = T3;
NET "s6_from_s6[3]" LOC = U3;
NET "s6_from_s6[4]" LOC = V3;
NET "s6_from_s6[5]" LOC = N4;
NET "s6_from_s6[6]" LOC = P4;
NET "s6_from_s6[7]" LOC = R4;
NET "s6_from_s6[8]" LOC = T4;
NET "s6_from_s6[9]" LOC = U4;
NET "s6_from_s6[10]" LOC = V4;
NET "s6_from_s6[11]" LOC = N5;
NET "s6_from_s6[12]" LOC = R5;
NET "s6_from_s6[13]" LOC = U5;
NET "s6_from_s6[14]" LOC = P6;
NET "s6_from_s6[15]" LOC = R6;
NET "s6_from_s6[16]" LOC = T6;
NET "s6_from_s6[17]" LOC = U6;
NET "s6_from_s6[18]" LOC = N7;
NET "s6_from_s6[19]" LOC = P7;
NET "s6_from_s6[20]" LOC = R7;
NET "s6_from_s6[21]" LOC = T7;
NET "s6_from_s6[22]" LOC = U7;
NET "s6_from_s6[23]" LOC = V7;
NET "s6_to_s6[0]" LOC = N1;
NET "s6_to_s6[1]" LOC = P1;
NET "s6_to_s6[2]" LOC = R1;
NET "s6_to_s6[3]" LOC = T1;
NET "s6_to_s6[4]" LOC = U1;
NET "s6_to_s6[5]" LOC = P2;
NET "s6_to_s6[6]" LOC = T2;
NET "s6_to_s6[7]" LOC = N3;
NET "ssr1_adv" LOC = Y7;
NET "ssr2_adv" LOC = D18;
NET "ssr1_bwa" LOC = Y8;
NET "ssr2_bwa" LOC = E18;
NET "ssr1_bbw" LOC = V2;
NET "ssr2_bbw" LOC = F18;
NET "ssr1_bwc" LOC = Y9;
NET "ssr2_bwc" LOC = H17;
NET "ssr1_bwd" LOC = W1;
NET "ssr2_bwd" LOC = J17;
NET "ssr1_cke" LOC = W3;
NET "ssr2_cke" LOC = G18;
NET "ssr1_clk" LOC = W4;
NET "ssr2_clk" LOC = A17;
NET "ssr1_csb" LOC = W5;
NET "ssr2_csb" LOC = B17;
NET "ssr1_lbo" LOC = W6;
NET "ssr2_lbo" LOC = C17;
NET "ssr1_oe" LOC = W7;
NET "ssr2_oe" LOC = D17;
NET "ssr1_we" LOC = W9;
NET "ssr2_we" LOC = F17;
NET "ssr1_zz" LOC = V1;
NET "ssr2_zz" LOC = G17;
NET "ssram1_ad[0]" LOC = AC5;
NET "ssram1_ad[1]" LOC = AC6;
NET "ssram1_ad[2]" LOC = AB1;
NET "ssram1_ad[3]" LOC = AB2;
NET "ssram1_ad[4]" LOC = AB3;

```

```

NET "ssram1_ad[5]" LOC = AB4;
NET "ssram1_ad[6]" LOC = AB6;
NET "ssram1_ad[7]" LOC = AB7;
NET "ssram1_ad[8]" LOC = AA1;
NET "ssram1_ad[9]" LOC = AA3;
NET "ssram1_ad[10]" LOC = AA4;
NET "ssram1_ad[11]" LOC = AA5;
NET "ssram1_ad[12]" LOC = AA6;
NET "ssram1_ad[13]" LOC = AA7;
NET "ssram1_ad[14]" LOC = AA9;
NET "ssram1_ad[15]" LOC = Y1;
NET "ssram1_ad[16]" LOC = Y2;
NET "ssram1_ad[17]" LOC = Y3;
NET "ssram1_ad[18]" LOC = Y4;
NET "ssram1_ad[19]" LOC = Y6;
NET "ssram2_ad[0]" LOC = F21;
NET "ssram2_ad[1]" LOC = G21;
NET "ssram2_ad[2]" LOC = H21;
NET "ssram2_ad[3]" LOC = J21;
NET "ssram2_ad[4]" LOC = A20;
NET "ssram2_ad[5]" LOC = C20;
NET "ssram2_ad[6]" LOC = D20;
NET "ssram2_ad[7]" LOC = E20;
NET "ssram2_ad[8]" LOC = G20;
NET "ssram2_ad[9]" LOC = J20;
NET "ssram2_ad[10]" LOC = A19;
NET "ssram2_ad[11]" LOC = B19;
NET "ssram2_ad[12]" LOC = C19;
NET "ssram2_ad[13]" LOC = D19;
NET "ssram2_ad[14]" LOC = F19;
NET "ssram2_ad[15]" LOC = G19;
NET "ssram2_ad[16]" LOC = H19;
NET "ssram2_ad[17]" LOC = J19;
NET "ssram2_ad[18]" LOC = E16;
NET "ssram2_ad[19]" LOC = C18;
NET "ssram1_dq[0]" LOC = AK2;
NET "ssram1_dq[1]" LOC = AK3;
NET "ssram1_dq[2]" LOC = AK4;
NET "ssram1_dq[3]" LOC = AK5;
NET "ssram1_dq[4]" LOC = AJ1;
NET "ssram1_dq[5]" LOC = AJ2;
NET "ssram1_dq[6]" LOC = AJ4;
NET "ssram1_dq[7]" LOC = AH1;
NET "ssram1_dq[8]" LOC = AH2;
NET "ssram1_dq[9]" LOC = AH3;
NET "ssram1_dq[10]" LOC = AH4;
NET "ssram1_dq[11]" LOC = AH5;
NET "ssram1_dq[12]" LOC = AG1;
NET "ssram1_dq[13]" LOC = AG3;
NET "ssram1_dq[14]" LOC = AG4;
NET "ssram1_dq[15]" LOC = AG5;
NET "ssram1_dq[16]" LOC = AF1;
NET "ssram1_dq[17]" LOC = AF2;
NET "ssram1_dq[18]" LOC = AF3;
NET "ssram1_dq[19]" LOC = AF4;
NET "ssram1_dq[20]" LOC = AF6;
NET "ssram1_dq[21]" LOC = AE1;
NET "ssram1_dq[22]" LOC = AE3;
NET "ssram1_dq[23]" LOC = AE4;
NET "ssram1_dq[24]" LOC = AE5;
NET "ssram1_dq[25]" LOC = AE6;
NET "ssram1_dq[26]" LOC = AE7;
NET "ssram1_dq[27]" LOC = AD1;
NET "ssram1_dq[28]" LOC = AD2;
NET "ssram1_dq[29]" LOC = AD3;
NET "ssram1_dq[30]" LOC = AD4;
NET "ssram1_dq[31]" LOC = AD6;
NET "ssram1_dq[32]" LOC = AD7;
NET "ssram1_dq[33]" LOC = AC1;
NET "ssram1_dq[34]" LOC = AC3;
NET "ssram1_dq[35]" LOC = AC4;
NET "ssram2_dq[0]" LOC = B29;
NET "ssram2_dq[1]" LOC = A28;
NET "ssram2_dq[2]" LOC = A27;
NET "ssram2_dq[3]" LOC = B27;
NET "ssram2_dq[4]" LOC = F27;
NET "ssram2_dq[5]" LOC = A26;
NET "ssram2_dq[6]" LOC = C26;
NET "ssram2_dq[7]" LOC = D26;
NET "ssram2_dq[8]" LOC = E26;
NET "ssram2_dq[9]" LOC = F26;
NET "ssram2_dq[10]" LOC = A25;
NET "ssram2_dq[11]" LOC = B25;
NET "ssram2_dq[12]" LOC = C25;
NET "ssram2_dq[13]" LOC = D25;
NET "ssram2_dq[14]" LOC = F25;
NET "ssram2_dq[15]" LOC = G25;
NET "ssram2_dq[16]" LOC = A24;
NET "ssram2_dq[17]" LOC = C24;
NET "ssram2_dq[18]" LOC = D24;
NET "ssram2_dq[19]" LOC = E24;
NET "ssram2_dq[20]" LOC = A23;
NET "ssram2_dq[21]" LOC = B23;
NET "ssram2_dq[22]" LOC = C23;
NET "ssram2_dq[23]" LOC = D23;
NET "ssram2_dq[24]" LOC = F23;
NET "ssram2_dq[25]" LOC = G23;
NET "ssram2_dq[26]" LOC = A22;
NET "ssram2_dq[27]" LOC = C22;
NET "ssram2_dq[28]" LOC = D22;
NET "ssram2_dq[29]" LOC = E22;
NET "ssram2_dq[30]" LOC = F22;
NET "ssram2_dq[31]" LOC = G22;
NET "ssram2_dq[32]" LOC = A21;
NET "ssram2_dq[33]" LOC = B21;
NET "ssram2_dq[34]" LOC = C21;
NET "ssram2_dq[35]" LOC = D21;
CONFIG PROHIBIT = G9;
NET "sl_uclk0" LOC = AK17;
NET "sl_uclk1" LOC = AH17;
NET "vme_spr[0]" LOC = AJ18;
NET "vme_spr[1]" LOC = AH19;
NET "vme_spr[2]" LOC = AJ16;
NET "vme_spr[3]" LOC = AK19;
NET "vme_spr[4]" LOC = AK16;
NET "vme_spr[5]" LOC = AE13;
NET "vme_spr[6]" LOC = AC12;
NET "vme_spr[7]" LOC = AD11;
NET "vme_spr[8]" LOC = AK9;
NET "vme_spr[9]" LOC = AE10;
NET "vme_spr[10]" LOC = AH8;
NET "vme_spr[11]" LOC = AE9;
NET "vme_spr[12]" LOC = AK8;
NET "vme_spr[13]" LOC = AK7;
NET "ck1_40_N" IOSTANDARD = LVDS_25;
NET "eth_txclk" IOSTANDARD = LVCMOS25;
NET "eth_rxclk" IOSTANDARD = LVCMOS25;
NET "eth_rxclk" IN_TERM = UNTUNED_SPLIT_25;
NET "fe_emu_clk" IOSTANDARD = LVCMOS25;
NET "fe_emu_clk" IN_TERM = UNTUNED_SPLIT_25;
NET "ck1_40_P" IOSTANDARD = LVDS_25;
NET "ck2_40_P" IOSTANDARD = LVDS_25;
NET "ck2_40_N" IOSTANDARD = LVDS_25;
NET "form_ds" IOSTANDARD = LVCMOS25;
NET "form_hwob" IOSTANDARD = LVCMOS25;
NET "sl_uclk0" IOSTANDARD = SSTL3_I;
NET "sl_uclk1" IOSTANDARD = SSTL3_I;
NET "ssr1_clkfb" IOSTANDARD = LVCMOS25;
NET "ssr2_clkfb" IOSTANDARD = LVCMOS25;
NET "ssram1_dq[0]" IOSTANDARD = LVCMOS25;
NET "ssram1_dq[0]" DRIVE = 8;
NET "ssram1_dq[0]" SLEW = SLOW;

```



```

NET "efb_event_dynamic_mask[12]" IOSTANDARD = LVCMOS25;
NET "efb_event_dynamic_mask[13]" IOSTANDARD = LVCMOS25;
NET "efb_event_dynamic_mask[14]" IOSTANDARD = LVCMOS25;
NET "efb_event_dynamic_mask[15]" IOSTANDARD = LVCMOS25;
NET "form_mode_bits[0]" IOSTANDARD = LVCMOS25;
NET "form_mode_bits[1]" IOSTANDARD = LVCMOS25;
NET "form_mode_bits[2]" IOSTANDARD = LVCMOS25;
NET "form_mode_bits[3]" IOSTANDARD = LVCMOS25;
NET "form_mode_bits[4]" IOSTANDARD = LVCMOS25;
NET "form_mode_bits[5]" IOSTANDARD = LVCMOS25;
NET "form_mode_bits[6]" IOSTANDARD = LVCMOS25;
NET "form_mode_bits[7]" IOSTANDARD = LVCMOS25;
NET "form_mode_bits[8]" IOSTANDARD = LVCMOS25;
NET "form_mode_bits[9]" IOSTANDARD = LVCMOS25;
NET "form_mode_bits[10]" IOSTANDARD = LVCMOS25;
NET "form_mode_bits[11]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[0]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[0]" DRIVE = 12;
NET "form_trailer_info[0]" SLEW = SLOW;
NET "form_trailer_info[1]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[1]" DRIVE = 12;
NET "form_trailer_info[1]" SLEW = SLOW;
NET "form_trailer_info[2]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[2]" DRIVE = 12;
NET "form_trailer_info[2]" SLEW = SLOW;
NET "form_trailer_info[3]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[3]" DRIVE = 12;
NET "form_trailer_info[3]" SLEW = SLOW;
NET "form_trailer_info[4]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[4]" DRIVE = 12;
NET "form_trailer_info[4]" SLEW = SLOW;
NET "form_trailer_info[5]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[5]" DRIVE = 12;
NET "form_trailer_info[5]" SLEW = SLOW;
NET "form_trailer_info[6]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[6]" DRIVE = 12;
NET "form_trailer_info[6]" SLEW = SLOW;
NET "form_trailer_info[7]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[7]" DRIVE = 12;
NET "form_trailer_info[7]" SLEW = SLOW;
NET "form_trailer_info[8]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[8]" DRIVE = 12;
NET "form_trailer_info[8]" SLEW = SLOW;
NET "form_trailer_info[9]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[9]" DRIVE = 12;
NET "form_trailer_info[9]" SLEW = SLOW;
NET "form_trailer_info[10]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[10]" DRIVE = 12;
NET "form_trailer_info[10]" SLEW = SLOW;
NET "form_trailer_info[11]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[11]" DRIVE = 12;
NET "form_trailer_info[11]" SLEW = SLOW;
NET "xc_in[0]" IOSTANDARD = LVCMOS25;
NET "xc_in[1]" IOSTANDARD = LVCMOS25;
NET "xc_in[2]" IOSTANDARD = LVCMOS25;
NET "xc_in[3]" IOSTANDARD = LVCMOS25;
NET "xc_in[4]" IOSTANDARD = LVCMOS25;
NET "xc_in[5]" IOSTANDARD = LVCMOS25;
NET "xc_in[6]" IOSTANDARD = LVCMOS25;
NET "xc_in[7]" IOSTANDARD = LVCMOS25;
NET "spare_to_v5[0]" IOSTANDARD = LVCMOS25;
NET "spare_to_v5[0]" DRIVE = 12;
NET "spare_to_v5[0]" SLEW = SLOW;
NET "spare_to_v5[1]" IOSTANDARD = LVCMOS25;
NET "spare_to_v5[1]" DRIVE = 12;
NET "spare_to_v5[1]" SLEW = SLOW;
NET "spare_to_v5[2]" IOSTANDARD = LVCMOS25;
NET "spare_to_v5[2]" DRIVE = 12;
NET "spare_to_v5[2]" SLEW = SLOW;
NET "spare_to_v5[3]" IOSTANDARD = LVCMOS25;
NET "spare_to_v5[3]" DRIVE = 12;
NET "spare_to_v5[3]" SLEW = SLOW;
NET "spare_to_v5[4]" IOSTANDARD = LVCMOS25;
NET "spare_to_v5[4]" DRIVE = 12;
NET "spare_to_v5[4]" SLEW = SLOW;
NET "spare_to_v5[5]" IOSTANDARD = LVCMOS25;
NET "spare_to_v5[5]" DRIVE = 12;
NET "spare_to_v5[5]" SLEW = SLOW;
NET "spare_to_v5[6]" IOSTANDARD = LVCMOS25;
NET "spare_to_v5[6]" DRIVE = 12;
NET "spare_to_v5[6]" SLEW = SLOW;
NET "spare_to_v5[7]" IOSTANDARD = LVCMOS25;
NET "spare_to_v5[7]" DRIVE = 12;
NET "spare_to_v5[7]" SLEW = SLOW;
NET "spare_to_v5[8]" IOSTANDARD = LVCMOS25;
NET "spare_to_v5[8]" DRIVE = 12;
NET "spare_to_v5[8]" SLEW = SLOW;
NET "spare_to_v5[9]" IOSTANDARD = LVCMOS25;
NET "spare_to_v5[9]" DRIVE = 12;
NET "spare_to_v5[9]" SLEW = SLOW;
NET "spare_to_v5[10]" IOSTANDARD = LVCMOS25;
NET "spare_to_v5[10]" DRIVE = 12;
NET "spare_to_v5[10]" SLEW = SLOW;
NET "spare_from_v5[0]" IOSTANDARD = LVCMOS25;
NET "spare_from_v5[1]" IOSTANDARD = LVCMOS25;
NET "spare_from_v5[2]" IOSTANDARD = LVCMOS25;
NET "spare_from_v5[3]" IOSTANDARD = LVCMOS25;
NET "spare_from_v5[4]" IOSTANDARD = LVCMOS25;
NET "spare_from_v5[5]" IOSTANDARD = LVCMOS25;
NET "spare_from_v5[6]" IOSTANDARD = LVCMOS25;
NET "spare_from_v5[7]" IOSTANDARD = LVCMOS25;
NET "spare_from_v5[8]" IOSTANDARD = LVCMOS25;
NET "spare_from_v5[9]" IOSTANDARD = LVCMOS25;
NET "fe_cmd_pulse_form" IOSTANDARD = LVCMOS25;
NET "form_mb_fifo_ef" IOSTANDARD = LVCMOS25;
NET "form_mb_fifo_ef" DRIVE = 12;
NET "form_mb_fifo_ef" SLEW = SLOW;
NET "form_mb_fifo_ff" IOSTANDARD = LVCMOS25;
NET "form_mb_fifo_ff" DRIVE = 12;
NET "form_mb_fifo_ff" SLEW = SLOW;
NET "form_mode_bits_rst_n" IOSTANDARD = LVCMOS25;
NET "header_trailer_limit" IOSTANDARD = LVCMOS25;
NET "header_trailer_limit" DRIVE = 12;
NET "header_trailer_limit" SLEW = SLOW;
NET "rod_busy" IOSTANDARD = LVCMOS25;
NET "rod_busy" DRIVE = 12;
NET "rod_busy" SLEW = SLOW;
NET "efb_edm_fifo_ef" IOSTANDARD = LVCMOS25;
NET "efb_edm_fifo_ef" DRIVE = 12;
NET "efb_edm_fifo_ef" SLEW = SLOW;
NET "efb_edm_fifo_ff" IOSTANDARD = LVCMOS25;
NET "efb_edm_fifo_ff" DRIVE = 12;
NET "efb_edm_fifo_ff" SLEW = SLOW;
NET "efb_edm_fifo_empty_error" IOSTANDARD = LVCMOS25;
NET "efb_edm_fifo_empty_error" DRIVE = 12;
NET "efb_edm_fifo_empty_error" SLEW = SLOW;
NET "efb_event_dynamic_mask_fifo_rst_n" IOSTANDARD = LVCMOS25;
NET "efb_event_dynamic_mask_fifo_we" IOSTANDARD = LVCMOS25;
NET "form_mode_bits_we" IOSTANDARD = LVCMOS25;
NET "show_trailer_flags" IOSTANDARD = LVCMOS25;
NET "ssram2_dq[0]" IOSTANDARD = LVCMOS25;
NET "ssram2_dq[0]" DRIVE = 8;
NET "ssram2_dq[0]" SLEW = SLOW;
NET "ssram2_dq[1]" IOSTANDARD = LVCMOS25;
NET "ssram2_dq[1]" DRIVE = 8;
NET "ssram2_dq[1]" SLEW = SLOW;
NET "ssram2_dq[2]" IOSTANDARD = LVCMOS25;
NET "ssram2_dq[2]" DRIVE = 8;
NET "ssram2_dq[2]" SLEW = SLOW;
NET "ssram2_dq[3]" IOSTANDARD = LVCMOS25;
NET "ssram2_dq[3]" DRIVE = 8;
NET "ssram2_dq[3]" SLEW = SLOW;
NET "ssram2_dq[4]" IOSTANDARD = LVCMOS25;

```



```

NET "ssram2_ad[15]" SLEW = SLOW;
NET "ssram2_ad[16]" IOSTANDARD = LVCMOS25;
NET "ssram2_ad[16]" DRIVE = 8;
NET "ssram2_ad[16]" SLEW = SLOW;
NET "ssram2_ad[17]" IOSTANDARD = LVCMOS25;
NET "ssram2_ad[17]" DRIVE = 8;
NET "ssram2_ad[17]" SLEW = SLOW;
NET "ssram2_ad[18]" IOSTANDARD = LVCMOS25;
NET "ssram2_ad[18]" DRIVE = 8;
NET "ssram2_ad[18]" SLEW = SLOW;
NET "ssram2_ad[19]" IOSTANDARD = LVCMOS25;
NET "ssram2_ad[19]" DRIVE = 8;
NET "ssram2_ad[19]" SLEW = SLOW;
NET "ssr2_adv" IOSTANDARD = LVCMOS25;
NET "ssr2_adv" DRIVE = 8;
NET "ssr2_adv" SLEW = SLOW;
NET "ssr2_bwa" IOSTANDARD = LVCMOS25;
NET "ssr2_bwa" DRIVE = 8;
NET "ssr2_bwa" SLEW = SLOW;
NET "ssr2_bwb" IOSTANDARD = LVCMOS25;
NET "ssr2_bwb" DRIVE = 8;
NET "ssr2_bwb" SLEW = SLOW;
NET "ssr2_cke" IOSTANDARD = LVCMOS25;
NET "ssr2_cke" DRIVE = 8;
NET "ssr2_cke" SLEW = SLOW;
NET "ssr2_clk" IOSTANDARD = LVCMOS25;
NET "ssr2_clk" DRIVE = 8;
NET "ssr2_clk" SLEW = SLOW;
NET "ssr2_csb" IOSTANDARD = LVCMOS25;
NET "ssr2_csb" DRIVE = 8;
NET "ssr2_csb" SLEW = SLOW;
NET "ssr2_lbo" IOSTANDARD = LVCMOS25;
NET "ssr2_lbo" DRIVE = 8;
NET "ssr2_lbo" SLEW = SLOW;
NET "ssr2_oe" IOSTANDARD = LVCMOS25;
NET "ssr2_oe" DRIVE = 8;
NET "ssr2_oe" SLEW = SLOW;
NET "ssr2_we" IOSTANDARD = LVCMOS25;
NET "ssr2_we" DRIVE = 8;
NET "ssr2_we" SLEW = SLOW;
NET "ssr2_zz" IOSTANDARD = LVCMOS25;
NET "ssr2_zz" DRIVE = 8;
NET "ssr2_zz" SLEW = SLOW;
NET "ssr2_bwc" IOSTANDARD = LVCMOS25;
NET "ssr2_bwc" DRIVE = 8;
NET "ssr2_bwc" SLEW = SLOW;
NET "ssr2_bwd" IOSTANDARD = LVCMOS25;
NET "ssr2_bwd" DRIVE = 8;
NET "ssr2_bwd" SLEW = SLOW;
NET "s6_to_s6[0]" IOSTANDARD = LVCMOS25;
NET "s6_to_s6[0]" DRIVE = 12;
NET "s6_to_s6[0]" SLEW = SLOW;
NET "s6_to_s6[1]" IOSTANDARD = LVCMOS25;
NET "s6_to_s6[1]" DRIVE = 12;
NET "s6_to_s6[1]" SLEW = SLOW;
NET "s6_to_s6[2]" IOSTANDARD = LVCMOS25;
NET "s6_to_s6[2]" DRIVE = 12;
NET "s6_to_s6[2]" SLEW = SLOW;
NET "s6_to_s6[3]" IOSTANDARD = LVCMOS25;
NET "s6_to_s6[3]" DRIVE = 12;
NET "s6_to_s6[3]" SLEW = SLOW;
NET "s6_to_s6[4]" IOSTANDARD = LVCMOS25;
NET "s6_to_s6[4]" DRIVE = 12;
NET "s6_to_s6[4]" SLEW = SLOW;
NET "s6_to_s6[5]" IOSTANDARD = LVCMOS25;
NET "s6_to_s6[5]" DRIVE = 12;
NET "s6_to_s6[5]" SLEW = SLOW;
NET "s6_to_s6[6]" IOSTANDARD = LVCMOS25;
NET "s6_to_s6[6]" DRIVE = 12;
NET "s6_to_s6[6]" SLEW = SLOW;
NET "s6_to_s6[7]" IOSTANDARD = LVCMOS25;
NET "s6_to_s6[7]" DRIVE = 12;
NET "s6_to_s6[7]" SLEW = SLOW;
NET "s6_from_s6[0]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[1]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[2]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[3]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[4]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[5]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[6]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[7]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[8]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[9]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[10]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[11]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[12]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[13]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[14]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[15]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[16]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[17]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[18]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[19]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[20]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[21]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[22]" IOSTANDARD = LVCMOS25;
NET "s6_from_s6[23]" IOSTANDARD = LVCMOS25;
NET "fe_emu_out[0]" IOSTANDARD = LVCMOS25;
NET "fe_emu_out[0]" DRIVE = 12;
NET "fe_emu_out[0]" SLEW = SLOW;
NET "fe_emu_out[1]" IOSTANDARD = LVCMOS25;
NET "fe_emu_out[1]" DRIVE = 12;
NET "fe_emu_out[1]" SLEW = SLOW;
NET "fe_emu_in[0]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[1]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[2]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[3]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[4]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[5]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[6]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[7]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[8]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[9]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[10]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[11]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[12]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[13]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[14]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[15]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[16]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[17]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[18]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[19]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[20]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[21]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[22]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[23]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[24]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[25]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[26]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[27]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[28]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[29]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[30]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[31]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[32]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[33]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[34]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[35]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[36]" IOSTANDARD = LVCMOS25;
NET "fe_emu_in[37]" IOSTANDARD = LVCMOS25;
NET "rd[0]" IOSTANDARD = SSTL3_I;

```



```
NET "eth_txd[7]" IOSTANDARD = LVCMOS25;
NET "eth_txd[7]" DRIVE = 12;
NET "eth_txd[7]" SLEW = SLOW;
NET "eth_col" IOSTANDARD = LVCMOS25;
NET "eth_crs" IOSTANDARD = LVCMOS25;
NET "eth_gtxclk" IOSTANDARD = LVCMOS25;
NET "eth_gtxclk" DRIVE = 12;
NET "eth_gtxclk" SLEW = SLOW;
NET "eth_int" IOSTANDARD = LVCMOS25;
NET "eth_mdc" IOSTANDARD = LVCMOS25;
NET "eth_mdc" DRIVE = 12;
NET "eth_mdc" SLEW = SLOW;
NET "eth_mdio" IOSTANDARD = LVCMOS25;
NET "eth_mdio" DRIVE = 12;
NET "eth_mdio" SLEW = SLOW;
NET "eth_res" IOSTANDARD = LVCMOS25;
NET "eth_res" DRIVE = 12;
NET "eth_res" SLEW = SLOW;
NET "eth_txer" IOSTANDARD = LVCMOS25;
NET "eth_txer" DRIVE = 12;
NET "eth_txer" SLEW = SLOW;
NET "eth_rxdv" IOSTANDARD = LVCMOS25;
NET "eth_rxdv" IN_TERM = UNTUNED_SPLIT_25;
NET "eth_rxer" IOSTANDARD = LVCMOS25;
NET "eth_rxer" IN_TERM = UNTUNED_SPLIT_25;
NET "eth_txen" IOSTANDARD = LVCMOS25;
NET "eth_txen" DRIVE = 12;
NET "eth_txen" SLEW = SLOW;
NET "mcb3_dram_a[0]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_a[10]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_a[11]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_a[12]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_a[13]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_a[1]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_a[2]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_a[3]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_a[4]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_a[5]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_a[6]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_a[7]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_a[8]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_a[9]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_ba[0]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_ba[1]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_ba[2]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[0]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[10]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[11]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[12]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[13]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[14]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[15]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[1]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[2]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[3]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[4]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[5]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[6]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[7]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[8]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_dq[9]" IOSTANDARD = SSTL18_I;
NET "mcb3_dram_cas_n" IOSTANDARD = SSTL18_II;
NET "mcb3_dram_ck_P" IOSTANDARD = DIFF_SSTL18_II;
NET "mcb3_dram_ck_N" IOSTANDARD = DIFF_SSTL18_II;
NET "mcb3_dram_cke" IOSTANDARD = SSTL18_II;
NET "mcb3_dram_dm" IOSTANDARD = SSTL18_II;
NET "mcb3_dram_dqs_P" IOSTANDARD = DIFF_SSTL18_II;
NET "mcb3_dram_dqs_N" IOSTANDARD = DIFF_SSTL18_II;
NET "mcb3_dram_odi" IOSTANDARD = SSTL18_II;
NET "mcb3_dram_ras_n" IOSTANDARD = SSTL18_II;
NET "mcb3_dram_udm" IOSTANDARD = SSTL18_II;
NET "mcb3_dram_udqs_P" IOSTANDARD = DIFF_SSTL18_II;
NET "mcb3_dram_udqs_N" IOSTANDARD = DIFF_SSTL18_II;
NET "mcb3_dram_we_n" IOSTANDARD = SSTL18_II;
NET "mcb3_rq" IOSTANDARD = SSTL18_II;
NET "mcb3_zio" IOSTANDARD = SSTL18_II;
NET "ssr1_clkfb" LOC = AB30;
NET "ssr2_clkfb" LOC = A18;
NET "disable_vtt_1V5" LOC = D8;
```

Virtex5 pin assignment (UCF file)

```

CONFIG PROHIBIT = H32;
CONFIG PROHIBIT = N34;
CONFIG PROHIBIT = W32;
CONFIG PROHIBIT = AH32;
NET "ddr2_a[0]" LOC = W24;
NET "ddr2_a[1]" LOC = V24;
NET "ddr2_a[2]" LOC = Y26;
NET "ddr2_a[3]" LOC = W26;
NET "ddr2_a[4]" LOC = V25;
NET "ddr2_a[5]" LOC = W25;
NET "ddr2_a[6]" LOC = Y27;
NET "ddr2_a[7]" LOC = W27;
NET "ddr2_a[8]" LOC = V30;
NET "ddr2_a[9]" LOC = V28;
NET "ddr2_a[10]" LOC = V27;
NET "ddr2_a[11]" LOC = W31;
NET "ddr2_a[12]" LOC = Y31;
NET "ddr2_a[13]" LOC = W29;
NET "ddr2_a[14]" LOC = V29;
NET "ddr2_ba[0]" LOC = Y29;
NET "ddr2_ba[1]" LOC = AB31;
NET "ddr2_ba[2]" LOC = AA31;
NET "ddr2_cas_n" LOC = AB30;
NET "ddr2_ras_n" LOC = AC30;
NET "ddr2_we_n" LOC = AA29;
NET "ddr2_cke[0]" LOC = P29;
NET "ddr2_cs_n[0]" LOC = T31;
NET "ddr2_odt[0]" LOC = M32;
NET "ddr2_dm[7]" LOC = U30;
NET "ddr2_dm[6]" LOC = R31;
NET "ddr2_dm[5]" LOC = H33;
NET "ddr2_dm[4]" LOC = L33;
NET "ddr2_dm[3]" LOC = P34;
NET "ddr2_dm[2]" LOC = AK34;
NET "ddr2_dm[1]" LOC = AG32;
NET "ddr2_dm[0]" LOC = AK33;
NET "ddr2_dq[63]" LOC = L29;
NET "ddr2_dq[62]" LOC = H30;
NET "ddr2_dq[61]" LOC = G31;
NET "ddr2_dq[60]" LOC = J30;
NET "ddr2_dq[59]" LOC = J31;
NET "ddr2_dq[58]" LOC = L30;
NET "ddr2_dq[57]" LOC = M30;
NET "ddr2_dq[56]" LOC = N29;
NET "ddr2_dq[55]" LOC = U27;
NET "ddr2_dq[54]" LOC = U28;
NET "ddr2_dq[53]" LOC = R26;
NET "ddr2_dq[52]" LOC = R27;
NET "ddr2_dq[51]" LOC = U26;
NET "ddr2_dq[50]" LOC = T26;
NET "ddr2_dq[49]" LOC = U25;
NET "ddr2_dq[48]" LOC = T25;
NET "ddr2_dq[47]" LOC = B32;
NET "ddr2_dq[46]" LOC = A33;
NET "ddr2_dq[45]" LOC = B33;
NET "ddr2_dq[44]" LOC = C33;
NET "ddr2_dq[43]" LOC = C32;
NET "ddr2_dq[42]" LOC = D32;
NET "ddr2_dq[41]" LOC = C34;
NET "ddr2_dq[40]" LOC = D34;
NET "ddr2_dq[39]" LOC = J32;
NET "ddr2_dq[38]" LOC = F34;
NET "ddr2_dq[37]" LOC = G33;
NET "ddr2_dq[36]" LOC = E33;
NET "ddr2_dq[35]" LOC = E32;
NET "ddr2_dq[34]" LOC = E34;
NET "ddr2_dq[33]" LOC = F33;

NET "ddr2_dq[32]" LOC = G32;
NET "ddr2_dq[31]" LOC = U31;
NET "ddr2_dq[30]" LOC = U32;
NET "ddr2_dq[29]" LOC = T34;
NET "ddr2_dq[28]" LOC = U33;
NET "ddr2_dq[27]" LOC = R32;
NET "ddr2_dq[26]" LOC = R33;
NET "ddr2_dq[25]" LOC = R34;
NET "ddr2_dq[24]" LOC = T33;
NET "ddr2_dq[23]" LOC = Y34;
NET "ddr2_dq[22]" LOC = AA34;
NET "ddr2_dq[21]" LOC = AA33;
NET "ddr2_dq[20]" LOC = Y33;
NET "ddr2_dq[19]" LOC = V34;
NET "ddr2_dq[18]" LOC = W34;
NET "ddr2_dq[17]" LOC = V33;
NET "ddr2_dq[16]" LOC = V32;
NET "ddr2_dq[15]" LOC = Y32;
NET "ddr2_dq[14]" LOC = AC34;
NET "ddr2_dq[13]" LOC = AD34;
NET "ddr2_dq[12]" LOC = AC32;
NET "ddr2_dq[11]" LOC = AB32;
NET "ddr2_dq[10]" LOC = AC33;
NET "ddr2_dq[9]" LOC = AB33;
NET "ddr2_dq[8]" LOC = AF33;
NET "ddr2_dq[7]" LOC = AL34;
NET "ddr2_dq[6]" LOC = AL33;
NET "ddr2_dq[5]" LOC = AM33;
NET "ddr2_dq[4]" LOC = AM32;
NET "ddr2_dq[3]" LOC = AN34;
NET "ddr2_dq[2]" LOC = AN33;
NET "ddr2_dq[1]" LOC = AN32;
NET "ddr2_dq[0]" LOC = AP32;
NET "sys_rst_n" LOC = AC29;
NET "phy_init_done" LOC = AF31;
NET "ddr2_ck_N[0]" LOC = T29;
NET "ddr2_ck_N[1]" LOC = L31;
NET "ddr2_ck_P[0]" LOC = T28;
NET "ddr2_ck_P[1]" LOC = K31;
NET "ddr2_dqs_N[0]" LOC = AJ34;
NET "ddr2_dqs_N[1]" LOC = AE32;
NET "ddr2_dqs_N[2]" LOC = AE34;
NET "ddr2_dqs_N[3]" LOC = K32;
NET "ddr2_dqs_N[4]" LOC = J34;
NET "ddr2_dqs_N[5]" LOC = K34;
NET "ddr2_dqs_N[6]" LOC = P30;
NET "ddr2_dqs_N[7]" LOC = N30;
NET "ddr2_dqs_P[0]" LOC = AH34;
NET "ddr2_dqs_P[1]" LOC = AD32;
NET "ddr2_dqs_P[2]" LOC = AF34;
NET "ddr2_dqs_P[3]" LOC = K33;
NET "ddr2_dqs_P[4]" LOC = H34;
NET "ddr2_dqs_P[5]" LOC = L34;
NET "ddr2_dqs_P[6]" LOC = P31;
NET "ddr2_dqs_P[7]" LOC = M31;
NET "eth_col" LOC = E19;
NET "eth_crs" LOC = C15;
NET "eth_gtxclk" LOC = C14;
NET "eth_int" LOC = A16;
NET "eth_mdc" LOC = B17;
NET "eth_mdio" LOC = A14;
NET "eth_res" LOC = A15;
NET "eth_rxd[0]" LOC = F18;
NET "eth_rxd[1]" LOC = G21;
NET "eth_rxd[2]" LOC = F21;
NET "eth_rxd[3]" LOC = E16;
NET "eth_rxd[4]" LOC = E17;

```

```
NET "eth_rxd[5]" LOC = E14;
NET "eth_rxd[6]" LOC = D14;
NET "eth_rxd[7]" LOC = F20;
NET "eth_rxv" LOC = B15;
NET "eth_rxer" LOC = B16;
NET "eth_txd[0]" LOC = G20;
NET "eth_txd[1]" LOC = D15;
NET "eth_txd[2]" LOC = D16;
NET "eth_txd[3]" LOC = D20;
NET "eth_txd[4]" LOC = E21;
NET "eth_txd[5]" LOC = D17;
NET "eth_txd[6]" LOC = C17;
NET "eth_txd[7]" LOC = F19;
NET "eth_txen" LOC = G18;
NET "eth_txer" LOC = D21;
NET "mdsp_aoe" LOC = K28;
NET "mdsp_ardy" LOC = K26;
NET "mdsp_are" LOC = L28;
NET "mdsp_ave" LOC = K27;
NET "mdsp_be_n[0]" LOC = J25;
NET "mdsp_be_n[1]" LOC = J24;
NET "mdsp_be_n[2]" LOC = L26;
NET "mdsp_be_n[3]" LOC = L25;
NET "mdsp_ce0_n" LOC = L24;
NET "mdsp_clkr0" LOC = P26;
NET "mdsp_clkr1" LOC = R24;
NET "mdsp_clkx0" LOC = P27;
NET "mdsp_clkx1" LOC = T24;
NET "mdsp_dr0" LOC = M28;
NET "mdsp_dr1" LOC = P25;
NET "mdsp_dx0" LOC = N28;
NET "mdsp_dx1" LOC = N25;
NET "mdsp_ea[2]" LOC = B31;
NET "mdsp_ea[3]" LOC = A31;
NET "mdsp_ea[4]" LOC = C30;
NET "mdsp_ea[5]" LOC = D29;
NET "mdsp_ea[6]" LOC = D31;
NET "mdsp_ea[7]" LOC = D30;
NET "mdsp_ea[8]" LOC = A30;
NET "mdsp_ea[9]" LOC = B30;
NET "mdsp_ea[10]" LOC = H27;
NET "mdsp_ea[11]" LOC = G27;
NET "mdsp_ea[12]" LOC = F26;
NET "mdsp_ea[13]" LOC = F25;
NET "mdsp_ea[14]" LOC = H24;
NET "mdsp_ea[15]" LOC = H25;
NET "mdsp_ea[16]" LOC = G26;
NET "mdsp_ea[17]" LOC = G25;
NET "mdsp_ea[18]" LOC = J26;
NET "mdsp_ea[19]" LOC = J27;
NET "mdsp_ea[20]" LOC = M26;
NET "mdsp_ea[21]" LOC = M25;
NET "mdsp_ed[0]" LOC = C27;
NET "mdsp_ed[1]" LOC = D26;
NET "mdsp_ed[2]" LOC = B28;
NET "mdsp_ed[3]" LOC = C29;
NET "mdsp_ed[4]" LOC = C25;
NET "mdsp_ed[5]" LOC = B25;
NET "mdsp_ed[6]" LOC = A26;
NET "mdsp_ed[7]" LOC = B27;
NET "mdsp_ed[8]" LOC = A25;
NET "mdsp_ed[9]" LOC = B26;
NET "mdsp_ed[10]" LOC = D25;
NET "mdsp_ed[11]" LOC = C24;
NET "mdsp_ed[12]" LOC = A24;
NET "mdsp_ed[13]" LOC = A23;
NET "mdsp_ed[14]" LOC = A20;
NET "mdsp_ed[15]" LOC = A19;
NET "mdsp_ed[16]" LOC = B23;
NET "mdsp_ed[17]" LOC = C23;
NET "mdsp_ed[18]" LOC = A18;
NET "mdsp_ed[19]" LOC = B18;
NET "mdsp_ed[20]" LOC = B22;
NET "mdsp_ed[21]" LOC = C22;
NET "mdsp_ed[22]" LOC = C18;
NET "mdsp_ed[23]" LOC = C19;
NET "mdsp_ed[24]" LOC = A21;
NET "mdsp_ed[25]" LOC = B21;
NET "mdsp_ed[26]" LOC = B20;
NET "mdsp_ed[27]" LOC = C20;
NET "mdsp_ed[28]" LOC = A29;
NET "mdsp_ed[29]" LOC = A28;
NET "mdsp_ed[30]" LOC = C28;
NET "mdsp_ed[31]" LOC = D27;
NET "mdsp_ext_int[4]" LOC = H28;
NET "mdsp_ext_int[5]" LOC = G28;
NET "mdsp_ext_int[6]" LOC = E28;
NET "mdsp_ext_int[7]" LOC = F28;
NET "mdsp_ext_int[8]" LOC = E26;
NET "mdsp_fsr0" LOC = N24;
NET "mdsp_fsr1" LOC = K21;
NET "mdsp_fsr1" LOC = K22;
NET "mdsp_fsx0" LOC = P24;
NET "mdsp_int_n" LOC = K23;
NET "mdsp_tinp[0]" LOC = F6;
NET "mdsp_tinp[1]" LOC = T10;
NET "mdsp_toutp[0]" LOC = E27;
NET "mdsp_toutp[1]" LOC = K24;
NET "prm_ctrl_in[0]" LOC = AM30;
NET "prm_ctrl_in[1]" LOC = AN30;
NET "prm_ctrl_in[2]" LOC = AL31;
NET "prm_ctrl_out[0]" LOC = AP24;
NET "prm_ctrl_out[1]" LOC = AN24;
NET "prm_ctrl_out[2]" LOC = AL24;
NET "prm_ctrl_out[3]" LOC = AL25;
NET "prm_ctrl_out[4]" LOC = AP25;
NET "prm_ctrl_out[5]" LOC = AP26;
NET "prm_ctrl_out[6]" LOC = AL26;
NET "prm_ctrl_out[7]" LOC = AM26;
NET "prm_ctrl_out[8]" LOC = AM25;
NET "prm_ctrl_out[9]" LOC = AN25;
NET "prm_ctrl_out[10]" LOC = AM28;
NET "prm_ctrl_out[11]" LOC = AN28;
NET "prm_ctrl_out[12]" LOC = AN27;
NET "prm_ctrl_out[13]" LOC = AP27;
NET "prm_ctrl_out[14]" LOC = AN29;
NET "prm_ctrl_out[15]" LOC = AP29;
NET "prm_ctrl_out[16]" LOC = AL28;
NET "prm_ctrl_out[17]" LOC = AM27;
NET "prm_ctrl_out[18]" LOC = AP31;
NET "prm_ctrl_out[19]" LOC = AP30;
NET "prm_data_in[0]" LOC = AM31;
NET "prm_data_in[1]" LOC = AL30;
NET "prm_data_in[2]" LOC = AL29;
NET "prm_data_in[3]" LOC = AM21;
NET "prm_data_in[4]" LOC = AM20;
NET "prm_data_in[5]" LOC = AN23;
NET "prm_data_in[6]" LOC = AM23;
NET "prm_data_in[7]" LOC = AN20;
NET "prm_data_in[8]" LOC = AP20;
NET "prm_data_in[9]" LOC = AN22;
NET "prm_data_in[10]" LOC = AM22;
NET "prm_data_in[11]" LOC = AN18;
NET "prm_data_in[12]" LOC = AM18;
NET "prm_data_in[13]" LOC = AP22;
NET "prm_data_in[14]" LOC = AP21;
NET "prm_data_in[15]" LOC = AN19;
NET "prm_data_in[16]" LOC = AP19;
NET "prm_data_in[17]" LOC = AG26;
NET "prm_data_in[18]" LOC = AG27;
NET "prm_data_in[19]" LOC = AG25;
NET "prm_data_in[20]" LOC = AF24;
```

```

NET "prm_data_in[21]" LOC = AF26;
NET "prm_data_in[22]" LOC = AF25;
NET "prm_data_in[23]" LOC = AJ26;
NET "prm_data_in[24]" LOC = AH27;
NET "prm_data_in[25]" LOC = AK27;
NET "prm_data_in[26]" LOC = AK28;
NET "prm_data_in[27]" LOC = AJ29;
NET "prm_data_in[28]" LOC = AK29;
NET "prm_data_in[29]" LOC = AJ27;
NET "prm_data_in[30]" LOC = AK26;
NET "prm_data_in[31]" LOC = AF28;
NET "qpll_lvds40mhz_N" LOC = AG17;
NET "qpll_lvds40mhz_P" LOC = AH18;
NET "tim_clock_ok" LOC = L6;
NET "ttc_bcnt[0]" LOC = H10;
NET "ttc_bcnt[1]" LOC = C12;
NET "ttc_bcnt[2]" LOC = D12;
NET "ttc_bcnt[3]" LOC = J11;
NET "ttc_bcnt[4]" LOC = K11;
NET "ttc_bcnt[5]" LOC = D10;
NET "ttc_bcnt[6]" LOC = D11;
NET "ttc_bcnt[7]" LOC = H8;
NET "ttc_bcnt[8]" LOC = G8;
NET "ttc_bcnt[9]" LOC = G10;
NET "ttc_bcnt[10]" LOC = F10;
NET "ttc_bcnt[11]" LOC = F8;
NET "ttc_bcntres" LOC = T8;
NET "ttc_bcntstr" LOC = E6;
NET "ttc_brest[2]" LOC = F9;
NET "ttc_brest[3]" LOC = E8;
NET "ttc_brest[4]" LOC = E9;
NET "ttc_brest[5]" LOC = H9;
NET "ttc_brest[6]" LOC = A13;
NET "ttc_brest[7]" LOC = B12;
NET "ttc_breststr1" LOC = G7;
NET "ttc_breststr2" LOC = R8;
NET "ttc_clockllaccept" LOC = F5;
NET "ttc_dberstr" LOC = G6;
NET "ttc_dout[0]" LOC = J10;
NET "ttc_dout[1]" LOC = J9;
NET "ttc_dout[2]" LOC = K8;
NET "ttc_dout[3]" LOC = K9;
NET "ttc_dout[4]" LOC = B13;
NET "ttc_dout[5]" LOC = C13;
NET "ttc_dout[6]" LOC = L10;
NET "ttc_dout[7]" LOC = L11;
NET "ttc_doutstr" LOC = U10;
NET "ttc_dq[0]" LOC = E12;
NET "ttc_dq[1]" LOC = E13;
NET "ttc_dq[2]" LOC = N10;
NET "ttc_dq[3]" LOC = N9;
NET "ttc_evcnthstr" LOC = J7;
NET "ttc_evcnlstr" LOC = H7;
NET "ttc_evntres" LOC = U7;
NET "ttc_llaccept" LOC = G13;
NET "ttc_nf[0]" LOC = K6;
NET "ttc_nf[1]" LOC = K7;
NET "ttc_nf[2]" LOC = P6;
NET "ttc_nf[3]" LOC = P7;
NET "ttc_nf[4]" LOC = L5;
NET "ttc_nf[5]" LOC = L4;
NET "ttc_nf[6]" LOC = P5;
NET "ttc_nf[7]" LOC = N5;
NET "ttc_ready" LOC = F13;
NET "ttc_reset_b" LOC = T9;
NET "ttc_serial_b_ch" LOC = R9;
NET "ttc_sinerrstr" LOC = T11;
NET "ttc_subaddr[0]" LOC = G11;
NET "ttc_subaddr[1]" LOC = G12;
NET "ttc_subaddr[2]" LOC = M8;
NET "ttc_subaddr[3]" LOC = L8;
NET "ttc_subaddr[4]" LOC = F11;
NET "ttc_subaddr[5]" LOC = E11;
NET "ttc_subaddr[6]" LOC = M10;
NET "ttc_subaddr[7]" LOC = L9;
NET "ttc_tck" LOC = H5;
NET "ttc_tdi" LOC = R11;
NET "ttc_tdo" LOC = P10;
NET "ttc_tms" LOC = G5;
NET "ttc_trst" LOC = P9;
NET "usb_db[0]" LOC = K13;
NET "usb_db[1]" LOC = H23;
NET "usb_db[2]" LOC = G23;
NET "usb_db[3]" LOC = H12;
NET "usb_db[4]" LOC = J12;
NET "usb_db[5]" LOC = K14;
NET "usb_db[6]" LOC = L14;
NET "usb_db[7]" LOC = H22;
NET "usb_pwe_n" LOC = G22;
NET "usb_rd_n" LOC = J15;
NET "usb_rxf_n" LOC = K16;
NET "usb_txe_n" LOC = J22;
NET "usb_wr_n" LOC = L16;
CONFIG PROHIBIT = G16;
NET "ctrl_from_prm[0]" LOC = H19;
NET "ctrl_from_prm[1]" LOC = H13;
NET "ctrl_from_prm[2]" LOC = J14;
NET "ctrl_from_prm[3]" LOC = J21;
NET "ctrl_from_prm[4]" LOC = J20;
NET "ctrl_from_prm[5]" LOC = H15;
NET "ctrl_from_prm[6]" LOC = H14;
NET "ctrl_from_prm[7]" LOC = K19;
NET "ctrl_from_prm[8]" LOC = L19;
NET "ctrl_from_prm[9]" LOC = J17;
NET "ctrl_from_prm[10]" LOC = J16;
NET "ctrl_from_prm[11]" LOC = J19;
NET "ctrl_from_prm[12]" LOC = K18;
NET "ctrl_from_prm[13]" LOC = L18;
NET "ctrl_from_prm[14]" LOC = K17;
NET "ctrl_from_prm[15]" LOC = H18;
NET "flash_cs" LOC = M5;
NET "flash_resetn" LOC = N8;
NET "flash_sck" LOC = N7;
NET "flash_si" LOC = M7;
NET "flash_so" LOC = M6;
NET "flash_wp" LOC = R6;
NET "prm_strobe[0]" LOC = AH28;
NET "prm_strobe[1]" LOC = AG28;
NET "reset_cmds_in[0]" LOC = T6;
NET "reset_cmds_in[1]" LOC = J6;
NET "reset_cmds_out[0]" LOC = J5;
NET "reset_cmds_out[1]" LOC = R7;
NET "ck1_40_N" LOC = AH19;
NET "ck1_40_P" LOC = AH20;
NET "ck2_40_N" LOC = AH13;
NET "ck2_40_P" LOC = AH14;
NET "efb_edm_fifo_ef" LOC = AD24;
NET "efb_edm_fifo_empty_error" LOC = AE24;
NET "efb_edm_fifo_ff" LOC = AC24;
NET "efb_event_dynamic_mask[0]" LOC = AF21;
NET "efb_event_dynamic_mask[1]" LOC = AG21;
NET "efb_event_dynamic_mask[2]" LOC = AJ21;
NET "efb_event_dynamic_mask[3]" LOC = AK21;
NET "efb_event_dynamic_mask[4]" LOC = AL21;
NET "efb_event_dynamic_mask[5]" LOC = AE22;
NET "efb_event_dynamic_mask[6]" LOC = AG22;
NET "efb_event_dynamic_mask[7]" LOC = AH22;
NET "efb_event_dynamic_mask[8]" LOC = AJ22;
NET "efb_event_dynamic_mask[9]" LOC = AK22;
NET "efb_event_dynamic_mask[10]" LOC = AE23;
NET "efb_event_dynamic_mask[11]" LOC = AF23;
NET "efb_event_dynamic_mask[12]" LOC = AG23;

```

```

NET "efb_event_dynamic_mask[13]" LOC = AH23;
NET "efb_event_dynamic_mask[14]" LOC = AK23;
NET "efb_event_dynamic_mask[15]" LOC = AL23;
NET "efb_event_dynamic_mask fifo_rst n" LOC = AK24;
NET "efb_event_dynamic_mask fifo_we" LOC = AJ24;
NET "form_mode_bits[0]" LOC = AF18;
NET "form_mode_bits[1]" LOC = AG18;
NET "form_mode_bits[2]" LOC = AK18;
NET "form_mode_bits[3]" LOC = AL18;
NET "form_mode_bits[4]" LOC = AD19;
NET "form_mode_bits[5]" LOC = AE19;
NET "form_mode_bits[6]" LOC = AF19;
NET "form_mode_bits[7]" LOC = AJ19;
NET "form_mode_bits[8]" LOC = AK19;
NET "form_mode_bits[9]" LOC = AL19;
NET "form_mode_bits[10]" LOC = AD20;
NET "form_mode_bits[11]" LOC = AF20;
NET "form_trailer_info[0]" LOC = AH9;
NET "form_trailer_info[1]" LOC = AJ9;
NET "form_trailer_info[2]" LOC = AK9;
NET "form_trailer_info[3]" LOC = AF10;
NET "form_trailer_info[4]" LOC = AG10;
NET "form_trailer_info[5]" LOC = AH10;
NET "form_trailer_info[6]" LOC = AJ10;
NET "form_trailer_info[7]" LOC = AL10;
NET "form_trailer_info[8]" LOC = AD11;
NET "form_trailer_info[9]" LOC = AE11;
NET "form_trailer_info[10]" LOC = AF11;
NET "form_trailer_info[11]" LOC = AG11;
NET "rod_bus_boc_busy" LOC = AB10;
NET "rod_bus_ce1_s6" LOC = AJ11;
NET "rod_bus_ce2_s6" LOC = V10;
NET "rod_bus_ce_boc" LOC = W10;
NET "rod_bus_data[0]" LOC = AC7;
NET "rod_bus_data[1]" LOC = AD7;
NET "rod_bus_data[2]" LOC = AE7;
NET "rod_bus_data[3]" LOC = U8;
NET "rod_bus_data[4]" LOC = V8;
NET "rod_bus_data[5]" LOC = Y8;
NET "rod_bus_data[6]" LOC = AA8;
NET "rod_bus_data[7]" LOC = AB8;
NET "rod_bus_data[8]" LOC = AC8;
NET "rod_bus_data[9]" LOC = AE8;
NET "rod_bus_data[10]" LOC = V9;
NET "rod_bus_data[11]" LOC = W9;
NET "rod_bus_data[12]" LOC = Y9;
NET "rod_bus_data[13]" LOC = AA9;
NET "rod_bus_data[14]" LOC = AC9;
NET "rod_bus_data[15]" LOC = AD9;
NET "rod_bus_rnw" LOC = AA10;
NET "setup_bus_dir" LOC = AC10;
NET "setup_bus_oe" LOC = AD10;
NET "spare_s6A[0]" LOC = AL15;
NET "spare_s6A[1]" LOC = AM15;
NET "spare_s6A[2]" LOC = AN15;
NET "spare_s6A[3]" LOC = AP15;
NET "spare_s6A[4]" LOC = AE16;
NET "spare_s6A[5]" LOC = AF16;
NET "spare_s6A[6]" LOC = AG15;
NET "spare_s6A[7]" LOC = AJ16;
NET "spare_s6A[8]" LOC = AK16;
NET "spare_s6A[9]" LOC = AL16;
NET "spare_s6A[10]" LOC = AM16;
NET "spare_s6A[11]" LOC = AP16;
NET "spare_s6A[12]" LOC = AE17;
NET "spare_s6A[13]" LOC = AH15;
NET "spare_s6A[14]" LOC = AD25;
NET "spare_s6A[15]" LOC = AD26;
NET "spare_s6A[16]" LOC = AM17;
NET "spare_s6A[17]" LOC = AN17;
NET "spare_s6B[0]" LOC = AK12;
NET "spare_s6B[1]" LOC = AM12;
NET "spare_s6B[2]" LOC = AN12;
NET "spare_s6B[3]" LOC = AP12;
NET "spare_s6B[4]" LOC = AE13;
NET "spare_s6B[5]" LOC = AF13;
NET "spare_s6B[6]" LOC = AG13;
NET "spare_s6B[7]" LOC = AK13;
NET "spare_s6B[8]" LOC = AL13;
NET "spare_s6B[9]" LOC = AM13;
NET "spare_s6B[10]" LOC = AN13;
NET "spare_s6B[11]" LOC = AE14;
NET "spare_s6B[12]" LOC = AF14;
NET "spare_s6B[13]" LOC = AJ14;
NET "spare_s6B[14]" LOC = AK14;
NET "spare_s6B[15]" LOC = AL14;
NET "spare_s6B[16]" LOC = AN14;
NET "fe_cmdpulse_form_outA" LOC = AL20;
NET "fe_cmdpulse_form_outB" LOC = AE21;
NET "form_ds" LOC = Y24;
NET "form_hwob" LOC = AA24;
NET "form_mb_fifo_efA" LOC = AL11;
NET "form_mb_fifo_efB" LOC = AM11;
NET "form_mb_fifo_ffA" LOC = AE26;
NET "form_mb_fifo_ffB" LOC = AE27;
NET "form_mode_bits_rst_nA" LOC = AG20;
NET "form_mode_bits_rst_nB" LOC = AJ20;
NET "form_mode_bits_weA" LOC = AH24;
NET "form_mode_bits_weB" LOC = AJ15;
NET "header_trailer_limit_inA" LOC = AH12;
NET "header_trailer_limit_inB" LOC = AJ12;
NET "rod_bus_ackA" LOC = AE18;
NET "rod_bus_ackB" LOC = AF15;
NET "rod_bus_addr[0]" LOC = AB5;
NET "rod_bus_addr[1]" LOC = AA5;
NET "rod_bus_addr[2]" LOC = AB7;
NET "rod_bus_addr[3]" LOC = AB6;
NET "rod_bus_addr[4]" LOC = AC5;
NET "rod_bus_addr[5]" LOC = AC4;
NET "rod_bus_addr[6]" LOC = AD4;
NET "rod_bus_addr[7]" LOC = AD5;
NET "rod_bus_addr[8]" LOC = AA6;
NET "rod_bus_addr[9]" LOC = Y7;
NET "rod_bus_addr[10]" LOC = AD6;
NET "rod_bus_addr[11]" LOC = AE6;
NET "rod_bus_addr[12]" LOC = W6;
NET "rod_bus_addr[13]" LOC = Y6;
NET "rod_bus_addr[14]" LOC = W7;
NET "rod_bus_addr[15]" LOC = V7;
NET "rod_bus_addr[16]" LOC = Y11;
NET "rod_bus_addr[17]" LOC = W11;
NET "rod_busy_inA" LOC = AE12;
NET "rod_busy_inB" LOC = AG12;
NET "show_trailer_flagsA" LOC = AP17;
NET "show_trailer_flagsB" LOC = AP14;
NET "spare_s6B[17]" LOC = AB25;
NET "xc_A[0]" LOC = AF5;
NET "xc_A[1]" LOC = AG5;
NET "xc_A[2]" LOC = AH5;
NET "xc_A[3]" LOC = AF6;
NET "xc_A[4]" LOC = AG6;
NET "xc_A[5]" LOC = AJ6;
NET "xc_A[6]" LOC = AK6;
NET "xc_A[7]" LOC = AG7;
NET "xc_B[0]" LOC = AH7;
NET "xc_B[1]" LOC = AJ7;
NET "xc_B[2]" LOC = AK7;
NET "xc_B[3]" LOC = AK11;
NET "xc_B[4]" LOC = AG8;
NET "xc_B[5]" LOC = AH8;
NET "xc_B[6]" LOC = AK8;
NET "xc_B[7]" LOC = AF9;

```

```

NET "boc_clock_ok" LOC = L20;
NET "eth_rxclk" LOC = H17;
NET "eth_txclk" LOC = D22;
NET "v5_busy_out" LOC = E7;
NET "mdsp_ext_iack[4]" LOC = D24;
NET "mdsp_ext_iack[5]" LOC = E24;
NET "mdsp_ext_iack[6]" LOC = E23;
NET "mdsp_ext_iack[7]" LOC = F23;
NET "mdsp_ext_iack[8]" LOC = F24;
NET "ddr2_scl" LOC = AF30;
NET "ddr2_sda" LOC = AG30;
NET "ext_trigger" LOC = K12;
NET "leds[0]" LOC = AA28;
NET "leds[1]" LOC = AB28;
NET "leds[2]" LOC = AD27;
NET "leds[3]" LOC = AC28;
NET "leds[4]" LOC = AB26;
NET "leds[5]" LOC = AC27;
NET "leds[6]" LOC = AB27;
NET "leds[7]" LOC = AA26;
NET "qpll_reset" LOC = L15;
NET "qpll_lvds40mhz_N" IOSTANDARD = LVDS_25;
NET "qpll_lvds40mhz_P" IOSTANDARD = LVDS_25;
NET "eth_rxclk" IOSTANDARD = LVCMOS33;
NET "eth_txclk" IOSTANDARD = LVCMOS33;
NET "eth_txclk" DRIVE = 12;
NET "eth_txclk" SLEW = SLOW;
NET "boc_clock_ok" IOSTANDARD = LVCMOS33;
NET "ck1_40_P" IOSTANDARD = LVDS_25;
NET "ck1_40_N" IOSTANDARD = LVDS_25;
NET "ck2_40_P" IOSTANDARD = LVDS_25;
NET "ck2_40_N" IOSTANDARD = LVDS_25;
NET "ddr2_scl" IOSTANDARD = LVCMOS18;
NET "ddr2_scl" DRIVE = 12;
NET "ddr2_scl" SLEW = SLOW;
NET "ddr2_sda" IOSTANDARD = LVCMOS18;
NET "ddr2_sda" DRIVE = 12;
NET "ddr2_sda" SLEW = SLOW;
NET "ext_trigger" IOSTANDARD = LVCMOS33;
NET "qpll_reset" IOSTANDARD = LVCMOS33;
NET "leds[0]" IOSTANDARD = LVCMOS25;
NET "leds[0]" DRIVE = 12;
NET "leds[0]" SLEW = SLOW;
NET "leds[1]" IOSTANDARD = LVCMOS25;
NET "leds[1]" DRIVE = 12;
NET "leds[1]" SLEW = SLOW;
NET "leds[2]" IOSTANDARD = LVCMOS25;
NET "leds[2]" DRIVE = 12;
NET "leds[2]" SLEW = SLOW;
NET "leds[3]" IOSTANDARD = LVCMOS25;
NET "leds[3]" DRIVE = 12;
NET "leds[3]" SLEW = SLOW;
NET "leds[4]" IOSTANDARD = LVCMOS25;
NET "leds[4]" DRIVE = 12;
NET "leds[4]" SLEW = SLOW;
NET "leds[5]" IOSTANDARD = LVCMOS25;
NET "leds[5]" DRIVE = 12;
NET "leds[5]" SLEW = SLOW;
NET "leds[6]" IOSTANDARD = LVCMOS25;
NET "leds[6]" DRIVE = 12;
NET "leds[6]" SLEW = SLOW;
NET "leds[7]" IOSTANDARD = LVCMOS25;
NET "leds[7]" DRIVE = 12;
NET "leds[7]" SLEW = SLOW;
NET "ctrl_from_prm[0]" IOSTANDARD = LVTTL;
NET "ctrl_from_prm[1]" IOSTANDARD = LVTTL;
NET "ctrl_from_prm[2]" IOSTANDARD = LVTTL;
NET "ctrl_from_prm[3]" IOSTANDARD = LVTTL;
NET "ctrl_from_prm[4]" IOSTANDARD = LVTTL;
NET "ctrl_from_prm[5]" IOSTANDARD = LVTTL;
NET "ctrl_from_prm[6]" IOSTANDARD = LVTTL;
NET "ctrl_from_prm[7]" IOSTANDARD = LVTTL;
NET "ctrl_from_prm[8]" IOSTANDARD = LVTTL;
NET "ctrl_from_prm[9]" IOSTANDARD = LVTTL;
NET "ctrl_from_prm[10]" IOSTANDARD = LVTTL;
NET "ctrl_from_prm[11]" IOSTANDARD = LVTTL;
NET "ctrl_from_prm[12]" IOSTANDARD = LVTTL;
NET "ctrl_from_prm[13]" IOSTANDARD = LVTTL;
NET "ctrl_from_prm[14]" IOSTANDARD = LVTTL;
NET "ctrl_from_prm[15]" IOSTANDARD = LVTTL;
NET "reset_cmds_in[0]" IOSTANDARD = LVTTL;
NET "reset_cmds_in[1]" IOSTANDARD = LVTTL;
NET "reset_cmds_out[0]" IOSTANDARD = LVTTL;
NET "reset_cmds_out[0]" DRIVE = 12;
NET "reset_cmds_out[0]" SLEW = SLOW;
NET "reset_cmds_out[1]" IOSTANDARD = LVTTL;
NET "reset_cmds_out[1]" DRIVE = 12;
NET "reset_cmds_out[1]" SLEW = SLOW;
NET "flash_si" IOSTANDARD = LVCMOS33;
NET "flash_si" DRIVE = 12;
NET "flash_si" SLEW = SLOW;
NET "flash_sck" IOSTANDARD = LVCMOS33;
NET "flash_sck" DRIVE = 12;
NET "flash_sck" SLEW = SLOW;
NET "flash_resetn" IOSTANDARD = LVCMOS33;
NET "flash_resetn" DRIVE = 12;
NET "flash_resetn" SLEW = SLOW;
NET "flash_cs" IOSTANDARD = LVCMOS33;
NET "flash_cs" DRIVE = 12;
NET "flash_cs" SLEW = SLOW;
NET "flash_so" IOSTANDARD = LVCMOS33;
NET "flash_wp" IOSTANDARD = LVCMOS33;
NET "flash_wp" DRIVE = 12;
NET "flash_wp" SLEW = SLOW;
NET "rod_bus_data[0]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[0]" DRIVE = 12;
NET "rod_bus_data[0]" SLEW = SLOW;
NET "rod_bus_data[1]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[1]" DRIVE = 12;
NET "rod_bus_data[1]" SLEW = SLOW;
NET "rod_bus_data[2]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[2]" DRIVE = 12;
NET "rod_bus_data[2]" SLEW = SLOW;
NET "rod_bus_data[3]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[3]" DRIVE = 12;
NET "rod_bus_data[3]" SLEW = SLOW;
NET "rod_bus_data[4]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[4]" DRIVE = 12;
NET "rod_bus_data[4]" SLEW = SLOW;
NET "rod_bus_data[5]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[5]" DRIVE = 12;
NET "rod_bus_data[5]" SLEW = SLOW;
NET "rod_bus_data[6]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[6]" DRIVE = 12;
NET "rod_bus_data[6]" SLEW = SLOW;
NET "rod_bus_data[7]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[7]" DRIVE = 12;
NET "rod_bus_data[7]" SLEW = SLOW;
NET "rod_bus_data[8]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[8]" DRIVE = 12;
NET "rod_bus_data[8]" SLEW = SLOW;
NET "rod_bus_data[9]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[9]" DRIVE = 12;
NET "rod_bus_data[9]" SLEW = SLOW;
NET "rod_bus_data[10]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[10]" DRIVE = 12;
NET "rod_bus_data[10]" SLEW = SLOW;
NET "rod_bus_data[11]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[11]" DRIVE = 12;
NET "rod_bus_data[11]" SLEW = SLOW;
NET "rod_bus_data[12]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[12]" DRIVE = 12;

```

```

NET "rod_bus_data[12]" SLEW = SLOW;
NET "rod_bus_data[13]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[13]" DRIVE = 12;
NET "rod_bus_data[13]" SLEW = SLOW;
NET "rod_bus_data[14]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[14]" DRIVE = 12;
NET "rod_bus_data[14]" SLEW = SLOW;
NET "rod_bus_data[15]" IOSTANDARD = LVCMOS25;
NET "rod_bus_data[15]" DRIVE = 12;
NET "rod_bus_data[15]" SLEW = SLOW;
NET "rod_bus_addr[0]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[0]" DRIVE = 12;
NET "rod_bus_addr[0]" SLEW = SLOW;
NET "rod_bus_addr[1]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[1]" DRIVE = 12;
NET "rod_bus_addr[1]" SLEW = SLOW;
NET "rod_bus_addr[2]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[2]" DRIVE = 12;
NET "rod_bus_addr[2]" SLEW = SLOW;
NET "rod_bus_addr[3]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[3]" DRIVE = 12;
NET "rod_bus_addr[3]" SLEW = SLOW;
NET "rod_bus_addr[4]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[4]" DRIVE = 12;
NET "rod_bus_addr[4]" SLEW = SLOW;
NET "rod_bus_addr[5]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[5]" DRIVE = 12;
NET "rod_bus_addr[5]" SLEW = SLOW;
NET "rod_bus_addr[6]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[6]" DRIVE = 12;
NET "rod_bus_addr[6]" SLEW = SLOW;
NET "rod_bus_addr[7]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[7]" DRIVE = 12;
NET "rod_bus_addr[7]" SLEW = SLOW;
NET "rod_bus_addr[8]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[8]" DRIVE = 12;
NET "rod_bus_addr[8]" SLEW = SLOW;
NET "rod_bus_addr[9]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[9]" DRIVE = 12;
NET "rod_bus_addr[9]" SLEW = SLOW;
NET "rod_bus_addr[10]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[10]" DRIVE = 12;
NET "rod_bus_addr[10]" SLEW = SLOW;
NET "rod_bus_addr[11]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[11]" DRIVE = 12;
NET "rod_bus_addr[11]" SLEW = SLOW;
NET "rod_bus_addr[12]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[12]" DRIVE = 12;
NET "rod_bus_addr[12]" SLEW = SLOW;
NET "rod_bus_addr[13]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[13]" DRIVE = 12;
NET "rod_bus_addr[13]" SLEW = SLOW;
NET "rod_bus_addr[14]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[14]" DRIVE = 12;
NET "rod_bus_addr[14]" SLEW = SLOW;
NET "rod_bus_addr[15]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[15]" DRIVE = 12;
NET "rod_bus_addr[15]" SLEW = SLOW;
NET "rod_bus_addr[16]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[16]" DRIVE = 12;
NET "rod_bus_addr[16]" SLEW = SLOW;
NET "rod_bus_addr[17]" IOSTANDARD = LVCMOS25;
NET "rod_bus_addr[17]" DRIVE = 12;
NET "rod_bus_addr[17]" SLEW = SLOW;
NET "rod_bus_ce_boc" IOSTANDARD = LVCMOS25;

NET "rod_bus_ce_boc" DRIVE = 12;
NET "rod_bus_ce_boc" SLEW = SLOW;
NET "rod_bus_rnw" IOSTANDARD = LVCMOS25;
NET "rod_bus_rnw" DRIVE = 12;
NET "rod_bus_rnw" SLEW = SLOW;
NET "rod_bus_boc_busy" IOSTANDARD = LVCMOS25;
NET "setup_bus_dir" IOSTANDARD = LVCMOS25;
NET "setup_bus_dir" DRIVE = 12;
NET "setup_bus_dir" SLEW = SLOW;
NET "setup_bus_oe" IOSTANDARD = LVCMOS25;
NET "setup_bus_oe" DRIVE = 12;
NET "setup_bus_oe" SLEW = SLOW;
NET "form_ds" IOSTANDARD = LVCMOS25;
NET "form_ds" DRIVE = 12;
NET "form_ds" SLEW = SLOW;
NET "form_hwob" IOSTANDARD = LVCMOS25;
NET "form_hwob" DRIVE = 12;
NET "form_hwob" SLEW = SLOW;
NET "xc_B[0]" IOSTANDARD = LVCMOS25;
NET "xc_B[0]" DRIVE = 12;
NET "xc_B[0]" SLEW = SLOW;
NET "xc_B[1]" IOSTANDARD = LVCMOS25;
NET "xc_B[1]" DRIVE = 12;
NET "xc_B[1]" SLEW = SLOW;
NET "xc_B[2]" IOSTANDARD = LVCMOS25;
NET "xc_B[2]" DRIVE = 12;
NET "xc_B[2]" SLEW = SLOW;
NET "xc_B[3]" IOSTANDARD = LVCMOS25;
NET "xc_B[3]" DRIVE = 12;
NET "xc_B[3]" SLEW = SLOW;
NET "xc_B[4]" IOSTANDARD = LVCMOS25;
NET "xc_B[4]" DRIVE = 12;
NET "xc_B[4]" SLEW = SLOW;
NET "xc_B[5]" IOSTANDARD = LVCMOS25;
NET "xc_B[5]" DRIVE = 12;
NET "xc_B[5]" SLEW = SLOW;
NET "xc_B[6]" IOSTANDARD = LVCMOS25;
NET "xc_B[6]" DRIVE = 12;
NET "xc_B[6]" SLEW = SLOW;
NET "xc_B[7]" IOSTANDARD = LVCMOS25;
NET "xc_B[7]" DRIVE = 12;
NET "xc_B[7]" SLEW = SLOW;
NET "form_trailer_info[0]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[1]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[2]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[3]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[4]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[5]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[6]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[7]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[8]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[9]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[10]" IOSTANDARD = LVCMOS25;
NET "form_trailer_info[11]" IOSTANDARD = LVCMOS25;
NET "spare_s6A[0]" IOSTANDARD = LVCMOS25;
NET "spare_s6A[0]" DRIVE = 12;
NET "spare_s6A[0]" SLEW = SLOW;
NET "spare_s6A[1]" IOSTANDARD = LVCMOS25;
NET "spare_s6A[1]" DRIVE = 12;
NET "spare_s6A[1]" SLEW = SLOW;
NET "spare_s6A[2]" IOSTANDARD = LVCMOS25;
NET "spare_s6A[2]" DRIVE = 12;
NET "spare_s6A[2]" SLEW = SLOW;
NET "spare_s6A[3]" IOSTANDARD = LVCMOS25;
NET "spare_s6A[3]" DRIVE = 12;
NET "spare_s6A[3]" SLEW = SLOW;
NET "spare_s6A[4]" IOSTANDARD = LVCMOS25;
NET "spare_s6A[4]" DRIVE = 12;
NET "spare_s6A[4]" SLEW = SLOW;
NET "spare_s6A[5]" IOSTANDARD = LVCMOS25;
NET "spare_s6A[5]" DRIVE = 12;

```



```

NET "form_mode_bits[9]" IOSTANDARD = LVCMOS25;
NET "form_mode_bits[9]" DRIVE = 12;
NET "form_mode_bits[9]" SLEW = SLOW;
NET "form_mode_bits[10]" IOSTANDARD = LVCMOS25;
NET "form_mode_bits[10]" DRIVE = 12;
NET "form_mode_bits[10]" SLEW = SLOW;
NET "form_mode_bits[11]" IOSTANDARD = LVCMOS25;
NET "form_mode_bits[11]" DRIVE = 12;
NET "form_mode_bits[11]" SLEW = SLOW;
NET "spare_s6B[0]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[0]" DRIVE = 12;
NET "spare_s6B[0]" SLEW = SLOW;
NET "spare_s6B[1]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[1]" DRIVE = 12;
NET "spare_s6B[1]" SLEW = SLOW;
NET "spare_s6B[2]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[2]" DRIVE = 12;
NET "spare_s6B[2]" SLEW = SLOW;
NET "spare_s6B[3]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[3]" DRIVE = 12;
NET "spare_s6B[3]" SLEW = SLOW;
NET "spare_s6B[4]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[4]" DRIVE = 12;
NET "spare_s6B[4]" SLEW = SLOW;
NET "spare_s6B[5]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[5]" DRIVE = 12;
NET "spare_s6B[5]" SLEW = SLOW;
NET "spare_s6B[6]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[6]" DRIVE = 12;
NET "spare_s6B[6]" SLEW = SLOW;
NET "spare_s6B[7]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[7]" DRIVE = 12;
NET "spare_s6B[7]" SLEW = SLOW;
NET "spare_s6B[8]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[8]" DRIVE = 12;
NET "spare_s6B[8]" SLEW = SLOW;
NET "spare_s6B[9]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[9]" DRIVE = 12;
NET "spare_s6B[9]" SLEW = SLOW;
NET "spare_s6B[10]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[10]" DRIVE = 12;
NET "spare_s6B[10]" SLEW = SLOW;
NET "spare_s6B[11]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[11]" DRIVE = 12;
NET "spare_s6B[11]" SLEW = SLOW;
NET "spare_s6B[12]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[12]" DRIVE = 12;
NET "spare_s6B[12]" SLEW = SLOW;
NET "spare_s6B[13]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[13]" DRIVE = 12;
NET "spare_s6B[13]" SLEW = SLOW;
NET "spare_s6B[14]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[14]" DRIVE = 12;
NET "spare_s6B[14]" SLEW = SLOW;
NET "spare_s6B[15]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[15]" DRIVE = 12;
NET "spare_s6B[15]" SLEW = SLOW;
NET "spare_s6B[16]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[16]" DRIVE = 12;
NET "spare_s6B[16]" SLEW = SLOW;
NET "spare_s6B[17]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[17]" DRIVE = 12;
NET "spare_s6B[17]" SLEW = SLOW;
NET "spare_s6B[18]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[18]" DRIVE = 12;
NET "spare_s6B[18]" SLEW = SLOW;
NET "spare_s6B[19]" IOSTANDARD = LVCMOS25;
NET "spare_s6B[19]" DRIVE = 12;
NET "spare_s6B[19]" SLEW = SLOW;
NET "efb_edm_fifo_ff" IOSTANDARD = LVCMOS25;
NET "efb_edm_fifo_ef" IOSTANDARD = LVCMOS25;

NET "efb_edm_fifo_empty_error" IOSTANDARD = LVCMOS25;
NET "form_mode_bits_rst_nB" IOSTANDARD = LVCMOS25;
NET "form_mode_bits_rst_nB" DRIVE = 12;
NET "form_mode_bits_rst_nB" SLEW = SLOW;
NET "efb_event_dynamic_mask_fifo_we" IOSTANDARD = LVCMOS25;
NET "efb_event_dynamic_mask_fifo_we" DRIVE = 12;
NET "efb_event_dynamic_mask_fifo_we" SLEW = SLOW;
NET "efb_event_dynamic_mask_fifo_rst_n" IOSTANDARD = LVCMOS25;
NET "efb_event_dynamic_mask_fifo_rst_n" DRIVE = 12;
NET "fe_cmddpulse_form_outA" IOSTANDARD = LVCMOS25;
NET "fe_cmddpulse_form_outA" DRIVE = 12;
NET "fe_cmddpulse_form_outA" SLEW = SLOW;
NET "fe_cmddpulse_form_outB" IOSTANDARD = LVCMOS25;
NET "fe_cmddpulse_form_outB" DRIVE = 12;
NET "fe_cmddpulse_form_outB" SLEW = SLOW;
NET "rod_busy_inA" IOSTANDARD = LVCMOS25;
NET "rod_busy_inB" IOSTANDARD = LVCMOS25;
NET "form_mode_bits_rst_nA" IOSTANDARD = LVCMOS25;
NET "form_mode_bits_rst_nA" DRIVE = 12;
NET "form_mode_bits_rst_nA" SLEW = SLOW;
NET "form_mode_bits_weA" IOSTANDARD = LVCMOS25;
NET "form_mode_bits_weA" DRIVE = 12;
NET "form_mode_bits_weA" SLEW = SLOW;
NET "form_mode_bits_web" IOSTANDARD = LVCMOS25;
NET "form_mode_bits_web" DRIVE = 12;
NET "form_mode_bits_web" SLEW = SLOW;
NET "form_mb_fifo_efA" IOSTANDARD = LVCMOS25;
NET "form_mb_fifo_efB" IOSTANDARD = LVCMOS25;
NET "header_trailer_limit_inA" IOSTANDARD = LVCMOS25;
NET "header_trailer_limit_inB" IOSTANDARD = LVCMOS25;
NET "form_mb_fifo_ffA" IOSTANDARD = LVCMOS25;
NET "form_mb_fifo_ffB" IOSTANDARD = LVCMOS25;
NET "rod_bus_ackA" IOSTANDARD = LVCMOS25;
NET "rod_bus_ackB" IOSTANDARD = LVCMOS25;
NET "show_trailer_flagsA" IOSTANDARD = LVCMOS25;
NET "show_trailer_flagsA" DRIVE = 12;
NET "show_trailer_flagsA" SLEW = SLOW;
NET "show_trailer_flagsB" IOSTANDARD = LVCMOS25;
NET "show_trailer_flagsB" DRIVE = 12;
NET "show_trailer_flagsB" SLEW = SLOW;
NET "ttc_bcnt[0]" IOSTANDARD = LVCMOS33;
NET "ttc_bcnt[1]" IOSTANDARD = LVCMOS33;
NET "ttc_bcnt[2]" IOSTANDARD = LVCMOS33;
NET "ttc_bcnt[3]" IOSTANDARD = LVCMOS33;
NET "ttc_bcnt[4]" IOSTANDARD = LVCMOS33;
NET "ttc_bcnt[5]" IOSTANDARD = LVCMOS33;
NET "ttc_bcnt[6]" IOSTANDARD = LVCMOS33;
NET "ttc_bcnt[7]" IOSTANDARD = LVCMOS33;
NET "ttc_bcnt[8]" IOSTANDARD = LVCMOS33;
NET "ttc_bcnt[9]" IOSTANDARD = LVCMOS33;
NET "ttc_bcnt[10]" IOSTANDARD = LVCMOS33;
NET "ttc_bcnt[11]" IOSTANDARD = LVCMOS33;
NET "ttc_brcst[2]" IOSTANDARD = LVCMOS33;
NET "ttc_brcst[3]" IOSTANDARD = LVCMOS33;
NET "ttc_brcst[4]" IOSTANDARD = LVCMOS33;
NET "ttc_brcst[5]" IOSTANDARD = LVCMOS33;
NET "ttc_brcst[6]" IOSTANDARD = LVCMOS33;
NET "ttc_brcst[7]" IOSTANDARD = LVCMOS33;
NET "ttc_dout[0]" IOSTANDARD = LVCMOS33;
NET "ttc_dout[1]" IOSTANDARD = LVCMOS33;
NET "ttc_dout[2]" IOSTANDARD = LVCMOS33;
NET "ttc_dout[3]" IOSTANDARD = LVCMOS33;
NET "ttc_dout[4]" IOSTANDARD = LVCMOS33;
NET "ttc_dout[5]" IOSTANDARD = LVCMOS33;
NET "ttc_dout[6]" IOSTANDARD = LVCMOS33;
NET "ttc_dout[7]" IOSTANDARD = LVCMOS33;
NET "ttc_subaddr[0]" IOSTANDARD = LVCMOS33;
NET "ttc_subaddr[1]" IOSTANDARD = LVCMOS33;

```

```

NET "ttc_subaddr[2]" IOSTANDARD = LVCMOS33;
NET "ttc_subaddr[3]" IOSTANDARD = LVCMOS33;
NET "ttc_subaddr[4]" IOSTANDARD = LVCMOS33;
NET "ttc_subaddr[5]" IOSTANDARD = LVCMOS33;
NET "ttc_subaddr[6]" IOSTANDARD = LVCMOS33;
NET "ttc_subaddr[7]" IOSTANDARD = LVCMOS33;
NET "ttc_dq[0]" IOSTANDARD = LVCMOS33;
NET "ttc_dq[1]" IOSTANDARD = LVCMOS33;
NET "ttc_dq[2]" IOSTANDARD = LVCMOS33;
NET "ttc_dq[3]" IOSTANDARD = LVCMOS33;
NET "ttc_ready" IOSTANDARD = LVCMOS33;
NET "ttc_llaccept" IOSTANDARD = LVCMOS33;
NET "ttc_bentstr" IOSTANDARD = LVCMOS33;
NET "ttc_doutstr" IOSTANDARD = LVCMOS33;
NET "ttc_reset_b" IOSTANDARD = LVCMOS33;
NET "ttc_reset_b" DRIVE = 12;
NET "ttc_reset_b" SLEW = SLOW;
NET "ttc_brcstr1" IOSTANDARD = LVCMOS33;
NET "ttc_dberrstr" IOSTANDARD = LVCMOS33;
NET "ttc_sinerrstr" IOSTANDARD = LVCMOS33;
NET "ttc_brcstr2" IOSTANDARD = LVCMOS33;
NET "ttc_bcntres" IOSTANDARD = LVCMOS33;
NET "ttc_clockllaccept" IOSTANDARD = LVCMOS33;
NET "ttc_tdo" IOSTANDARD = LVCMOS33;
NET "ttc_tdi" IOSTANDARD = LVCMOS33;
NET "ttc_tdi" DRIVE = 12;
NET "ttc_tdi" SLEW = SLOW;
NET "ttc_tms" IOSTANDARD = LVCMOS33;
NET "ttc_tms" DRIVE = 12;
NET "ttc_tms" SLEW = SLOW;
NET "ttc_tck" IOSTANDARD = LVCMOS33;
NET "ttc_tck" DRIVE = 12;
NET "ttc_tck" SLEW = SLOW;
NET "ttc_trst" IOSTANDARD = LVCMOS33;
NET "ttc_trst" DRIVE = 12;
NET "ttc_trst" SLEW = SLOW;
NET "ttc_serial_b_ch" IOSTANDARD = LVCMOS33;
NET "ttc_eventhstr" IOSTANDARD = LVCMOS33;
NET "ttc_evnctlstr" IOSTANDARD = LVCMOS33;
NET "ttc_evnctres" IOSTANDARD = LVCMOS33;
NET "usb_db[0]" IOSTANDARD = LVCMOS33;
NET "usb_db[0]" DRIVE = 12;
NET "usb_db[0]" SLEW = SLOW;
NET "usb_db[1]" IOSTANDARD = LVCMOS33;
NET "usb_db[1]" DRIVE = 12;
NET "usb_db[1]" SLEW = SLOW;
NET "usb_db[2]" IOSTANDARD = LVCMOS33;
NET "usb_db[2]" DRIVE = 12;
NET "usb_db[2]" SLEW = SLOW;
NET "usb_db[3]" IOSTANDARD = LVCMOS33;
NET "usb_db[3]" DRIVE = 12;
NET "usb_db[3]" SLEW = SLOW;
NET "usb_db[4]" IOSTANDARD = LVCMOS33;
NET "usb_db[4]" DRIVE = 12;
NET "usb_db[4]" SLEW = SLOW;
NET "usb_db[5]" IOSTANDARD = LVCMOS33;
NET "usb_db[5]" DRIVE = 12;
NET "usb_db[5]" SLEW = SLOW;
NET "usb_db[6]" IOSTANDARD = LVCMOS33;
NET "usb_db[6]" DRIVE = 12;
NET "usb_db[6]" SLEW = SLOW;
NET "usb_db[7]" IOSTANDARD = LVCMOS33;
NET "usb_db[7]" DRIVE = 12;
NET "usb_db[7]" SLEW = SLOW;
NET "usb_pwe_n" IOSTANDARD = LVCMOS33;
NET "usb_pwe_n" DRIVE = 12;
NET "usb_pwe_n" SLEW = SLOW;
NET "usb_rd_n" IOSTANDARD = LVCMOS33;
NET "usb_rd_n" DRIVE = 12;
NET "usb_rd_n" SLEW = SLOW;
NET "usb_rxf_n" IOSTANDARD = LVCMOS33;

NET "usb_rxf_n" DRIVE = 12;
NET "usb_rxf_n" SLEW = SLOW;
NET "usb_txe_n" IOSTANDARD = LVCMOS33;
NET "usb_txe_n" DRIVE = 12;
NET "usb_txe_n" SLEW = SLOW;
NET "usb_wr_n" IOSTANDARD = LVCMOS33;
NET "usb_wr_n" DRIVE = 12;
NET "usb_wr_n" SLEW = SLOW;
NET "eth_rxd[0]" IOSTANDARD = LVCMOS33;
NET "eth_rxd[1]" IOSTANDARD = LVCMOS33;
NET "eth_rxd[2]" IOSTANDARD = LVCMOS33;
NET "eth_rxd[3]" IOSTANDARD = LVCMOS33;
NET "eth_rxd[4]" IOSTANDARD = LVCMOS33;
NET "eth_rxd[5]" IOSTANDARD = LVCMOS33;
NET "eth_rxd[6]" IOSTANDARD = LVCMOS33;
NET "eth_rxd[7]" IOSTANDARD = LVCMOS33;
NET "eth_txd[0]" IOSTANDARD = LVCMOS33;
NET "eth_txd[0]" DRIVE = 12;
NET "eth_txd[0]" SLEW = SLOW;
NET "eth_txd[1]" IOSTANDARD = LVCMOS33;
NET "eth_txd[1]" DRIVE = 12;
NET "eth_txd[1]" SLEW = SLOW;
NET "eth_txd[2]" IOSTANDARD = LVCMOS33;
NET "eth_txd[2]" DRIVE = 12;
NET "eth_txd[2]" SLEW = SLOW;
NET "eth_txd[3]" IOSTANDARD = LVCMOS33;
NET "eth_txd[3]" DRIVE = 12;
NET "eth_txd[3]" SLEW = SLOW;
NET "eth_txd[4]" IOSTANDARD = LVCMOS33;
NET "eth_txd[4]" DRIVE = 12;
NET "eth_txd[4]" SLEW = SLOW;
NET "eth_txd[5]" IOSTANDARD = LVCMOS33;
NET "eth_txd[5]" DRIVE = 12;
NET "eth_txd[5]" SLEW = SLOW;
NET "eth_txd[6]" IOSTANDARD = LVCMOS33;
NET "eth_txd[6]" DRIVE = 12;
NET "eth_txd[6]" SLEW = SLOW;
NET "eth_txd[7]" IOSTANDARD = LVCMOS33;
NET "eth_txd[7]" DRIVE = 12;
NET "eth_txd[7]" SLEW = SLOW;
NET "eth_col" IOSTANDARD = LVCMOS33;
NET "eth_crs" IOSTANDARD = LVCMOS33;
NET "eth_gtxclk" IOSTANDARD = LVCMOS33;
NET "eth_gtxclk" DRIVE = 12;
NET "eth_gtxclk" SLEW = SLOW;
NET "eth_int" IOSTANDARD = LVCMOS33;
NET "eth_mdc" IOSTANDARD = LVCMOS33;
NET "eth_mdc" DRIVE = 12;
NET "eth_mdc" SLEW = SLOW;
NET "eth_mdio" IOSTANDARD = LVCMOS33;
NET "eth_mdio" DRIVE = 12;
NET "eth_mdio" SLEW = SLOW;
NET "eth_res" IOSTANDARD = LVCMOS33;
NET "eth_res" DRIVE = 12;
NET "eth_res" SLEW = SLOW;
NET "eth_rxdv" IOSTANDARD = LVCMOS33;
NET "eth_rxer" IOSTANDARD = LVCMOS33;
NET "eth_txen" IOSTANDARD = LVCMOS33;
NET "eth_txen" DRIVE = 12;
NET "eth_txen" SLEW = SLOW;
NET "eth_txer" IOSTANDARD = LVCMOS33;
NET "eth_txer" DRIVE = 12;
NET "eth_txer" SLEW = SLOW;
NET "mdsp_ed[0]" IOSTANDARD = LVTTL;
NET "mdsp_ed[0]" DRIVE = 12;
NET "mdsp_ed[0]" SLEW = SLOW;
NET "mdsp_ed[1]" IOSTANDARD = LVTTL;
NET "mdsp_ed[1]" DRIVE = 12;
NET "mdsp_ed[1]" SLEW = SLOW;
NET "mdsp_ed[2]" IOSTANDARD = LVTTL;
NET "mdsp_ed[2]" DRIVE = 12;

```

```

NET "mdsp_ed[2]" SLEW = SLOW;
NET "mdsp_ed[3]" IOSTANDARD = LVTTL;
NET "mdsp_ed[3]" DRIVE = 12;
NET "mdsp_ed[3]" SLEW = SLOW;
NET "mdsp_ed[4]" IOSTANDARD = LVTTL;
NET "mdsp_ed[4]" DRIVE = 12;
NET "mdsp_ed[4]" SLEW = SLOW;
NET "mdsp_ed[5]" IOSTANDARD = LVTTL;
NET "mdsp_ed[5]" DRIVE = 12;
NET "mdsp_ed[5]" SLEW = SLOW;
NET "mdsp_ed[6]" IOSTANDARD = LVTTL;
NET "mdsp_ed[6]" DRIVE = 12;
NET "mdsp_ed[6]" SLEW = SLOW;
NET "mdsp_ed[7]" IOSTANDARD = LVTTL;
NET "mdsp_ed[7]" DRIVE = 12;
NET "mdsp_ed[7]" SLEW = SLOW;
NET "mdsp_ed[8]" IOSTANDARD = LVTTL;
NET "mdsp_ed[8]" DRIVE = 12;
NET "mdsp_ed[8]" SLEW = SLOW;
NET "mdsp_ed[9]" IOSTANDARD = LVTTL;
NET "mdsp_ed[9]" DRIVE = 12;
NET "mdsp_ed[9]" SLEW = SLOW;
NET "mdsp_ed[10]" IOSTANDARD = LVTTL;
NET "mdsp_ed[10]" DRIVE = 12;
NET "mdsp_ed[10]" SLEW = SLOW;
NET "mdsp_ed[11]" IOSTANDARD = LVTTL;
NET "mdsp_ed[11]" DRIVE = 12;
NET "mdsp_ed[11]" SLEW = SLOW;
NET "mdsp_ed[12]" IOSTANDARD = LVTTL;
NET "mdsp_ed[12]" DRIVE = 12;
NET "mdsp_ed[12]" SLEW = SLOW;
NET "mdsp_ed[13]" IOSTANDARD = LVTTL;
NET "mdsp_ed[13]" DRIVE = 12;
NET "mdsp_ed[13]" SLEW = SLOW;
NET "mdsp_ed[14]" IOSTANDARD = LVTTL;
NET "mdsp_ed[14]" DRIVE = 12;
NET "mdsp_ed[14]" SLEW = SLOW;
NET "mdsp_ed[15]" IOSTANDARD = LVTTL;
NET "mdsp_ed[15]" DRIVE = 12;
NET "mdsp_ed[15]" SLEW = SLOW;
NET "mdsp_ed[16]" IOSTANDARD = LVTTL;
NET "mdsp_ed[16]" DRIVE = 12;
NET "mdsp_ed[16]" SLEW = SLOW;
NET "mdsp_ed[17]" IOSTANDARD = LVTTL;
NET "mdsp_ed[17]" DRIVE = 12;
NET "mdsp_ed[17]" SLEW = SLOW;
NET "mdsp_ed[18]" IOSTANDARD = LVTTL;
NET "mdsp_ed[18]" DRIVE = 12;
NET "mdsp_ed[18]" SLEW = SLOW;
NET "mdsp_ed[19]" IOSTANDARD = LVTTL;
NET "mdsp_ed[19]" DRIVE = 12;
NET "mdsp_ed[19]" SLEW = SLOW;
NET "mdsp_ed[20]" IOSTANDARD = LVTTL;
NET "mdsp_ed[20]" DRIVE = 12;
NET "mdsp_ed[20]" SLEW = SLOW;
NET "mdsp_ed[21]" IOSTANDARD = LVTTL;
NET "mdsp_ed[21]" DRIVE = 12;
NET "mdsp_ed[21]" SLEW = SLOW;
NET "mdsp_ed[22]" IOSTANDARD = LVTTL;
NET "mdsp_ed[22]" DRIVE = 12;
NET "mdsp_ed[22]" SLEW = SLOW;
NET "mdsp_ed[23]" IOSTANDARD = LVTTL;
NET "mdsp_ed[23]" DRIVE = 12;
NET "mdsp_ed[23]" SLEW = SLOW;
NET "mdsp_ed[24]" IOSTANDARD = LVTTL;
NET "mdsp_ed[24]" DRIVE = 12;
NET "mdsp_ed[24]" SLEW = SLOW;
NET "mdsp_ed[25]" IOSTANDARD = LVTTL;
NET "mdsp_ed[25]" DRIVE = 12;
NET "mdsp_ed[25]" SLEW = SLOW;
NET "mdsp_ed[26]" IOSTANDARD = LVTTL;
NET "mdsp_ed[26]" DRIVE = 12;
NET "mdsp_ed[26]" SLEW = SLOW;
NET "mdsp_ed[27]" IOSTANDARD = LVTTL;
NET "mdsp_ed[27]" DRIVE = 12;
NET "mdsp_ed[27]" SLEW = SLOW;
NET "mdsp_ed[28]" IOSTANDARD = LVTTL;
NET "mdsp_ed[28]" DRIVE = 12;
NET "mdsp_ed[28]" SLEW = SLOW;
NET "mdsp_ed[29]" IOSTANDARD = LVTTL;
NET "mdsp_ed[29]" DRIVE = 12;
NET "mdsp_ed[29]" SLEW = SLOW;
NET "mdsp_ed[30]" IOSTANDARD = LVTTL;
NET "mdsp_ed[30]" DRIVE = 12;
NET "mdsp_ed[30]" SLEW = SLOW;
NET "mdsp_ed[31]" IOSTANDARD = LVTTL;
NET "mdsp_ed[31]" DRIVE = 12;
NET "mdsp_ed[31]" SLEW = SLOW;
NET "mdsp_ea[2]" IOSTANDARD = LVTTL;
NET "mdsp_ea[3]" IOSTANDARD = LVTTL;
NET "mdsp_ea[4]" IOSTANDARD = LVTTL;
NET "mdsp_ea[5]" IOSTANDARD = LVTTL;
NET "mdsp_ea[6]" IOSTANDARD = LVTTL;
NET "mdsp_ea[7]" IOSTANDARD = LVTTL;
NET "mdsp_ea[8]" IOSTANDARD = LVTTL;
NET "mdsp_ea[9]" IOSTANDARD = LVTTL;
NET "mdsp_ea[10]" IOSTANDARD = LVTTL;
NET "mdsp_ea[11]" IOSTANDARD = LVTTL;
NET "mdsp_ea[12]" IOSTANDARD = LVTTL;
NET "mdsp_ea[13]" IOSTANDARD = LVTTL;
NET "mdsp_ea[14]" IOSTANDARD = LVTTL;
NET "mdsp_ea[15]" IOSTANDARD = LVTTL;
NET "mdsp_ea[16]" IOSTANDARD = LVTTL;
NET "mdsp_ea[17]" IOSTANDARD = LVTTL;
NET "mdsp_ea[18]" IOSTANDARD = LVTTL;
NET "mdsp_ea[19]" IOSTANDARD = LVTTL;
NET "mdsp_ea[20]" IOSTANDARD = LVTTL;
NET "mdsp_ea[21]" IOSTANDARD = LVTTL;
NET "mdsp_be_n[0]" IOSTANDARD = LVTTL;
NET "mdsp_be_n[1]" IOSTANDARD = LVTTL;
NET "mdsp_be_n[2]" IOSTANDARD = LVTTL;
NET "mdsp_be_n[3]" IOSTANDARD = LVTTL;
NET "mdsp_tinp[0]" IOSTANDARD = LVTTL;
NET "mdsp_tinp[0]" DRIVE = 12;
NET "mdsp_tinp[0]" SLEW = SLOW;
NET "mdsp_tinp[1]" IOSTANDARD = LVTTL;
NET "mdsp_tinp[1]" DRIVE = 12;
NET "mdsp_tinp[1]" SLEW = SLOW;
NET "mdsp_ext_int[4]" IOSTANDARD = LVTTL;
NET "mdsp_ext_int[4]" DRIVE = 12;
NET "mdsp_ext_int[4]" SLEW = SLOW;
NET "mdsp_ext_int[5]" IOSTANDARD = LVTTL;
NET "mdsp_ext_int[5]" DRIVE = 12;
NET "mdsp_ext_int[5]" SLEW = SLOW;
NET "mdsp_ext_int[6]" IOSTANDARD = LVTTL;
NET "mdsp_ext_int[6]" DRIVE = 12;
NET "mdsp_ext_int[6]" SLEW = SLOW;
NET "mdsp_ext_int[7]" IOSTANDARD = LVTTL;
NET "mdsp_ext_int[7]" DRIVE = 12;
NET "mdsp_ext_int[7]" SLEW = SLOW;
NET "mdsp_ext_int[8]" IOSTANDARD = LVTTL;
NET "mdsp_ext_int[8]" DRIVE = 12;
NET "mdsp_ext_int[8]" SLEW = SLOW;
NET "mdsp_toutp[0]" IOSTANDARD = LVTTL;
NET "mdsp_toutp[1]" IOSTANDARD = LVTTL;
NET "mdsp_ext_iack[4]" IOSTANDARD = LVCMOS33;
NET "mdsp_ext_iack[5]" IOSTANDARD = LVCMOS33;
NET "mdsp_ext_iack[6]" IOSTANDARD = LVCMOS33;
NET "mdsp_ext_iack[7]" IOSTANDARD = LVCMOS33;
NET "mdsp_ext_iack[8]" IOSTANDARD = LVCMOS33;
NET "mdsp_ce0_n" IOSTANDARD = LVTTL;
NET "mdsp_aoe" IOSTANDARD = LVTTL;

```

```

NET "mdsp_are" IOSTANDARD = LVTTL;
NET "mdsp_awe" IOSTANDARD = LVTTL;
NET "mdsp_aridy" IOSTANDARD = LVTTL;
NET "mdsp_aridy" DRIVE = 12;
NET "mdsp_aridy" SLEW = SLOW;
NET "mdsp_dr0" IOSTANDARD = LVTTL;
NET "mdsp_dr0" DRIVE = 12;
NET "mdsp_dr0" SLEW = SLOW;
NET "mdsp_dx0" IOSTANDARD = LVTTL;
NET "mdsp_clk0" IOSTANDARD = LVTTL;
NET "mdsp_clk0" DRIVE = 12;
NET "mdsp_clk0" SLEW = SLOW;
NET "mdsp_clkx0" IOSTANDARD = LVTTL;
NET "mdsp_clkx0" DRIVE = 12;
NET "mdsp_clkx0" SLEW = SLOW;
NET "mdsp_fsr0" IOSTANDARD = LVTTL;
NET "mdsp_fsr0" DRIVE = 12;
NET "mdsp_fsr0" SLEW = SLOW;
NET "mdsp_fx0" IOSTANDARD = LVTTL;
NET "mdsp_fx0" DRIVE = 12;
NET "mdsp_fx0" SLEW = SLOW;
NET "mdsp_dr1" IOSTANDARD = LVTTL;
NET "mdsp_dr1" DRIVE = 12;
NET "mdsp_dr1" SLEW = SLOW;
NET "mdsp_dx1" IOSTANDARD = LVTTL;
NET "mdsp_clk1" IOSTANDARD = LVTTL;
NET "mdsp_clk1" DRIVE = 12;
NET "mdsp_clk1" SLEW = SLOW;
NET "mdsp_fsr1" IOSTANDARD = LVTTL;
NET "mdsp_fsr1" DRIVE = 12;
NET "mdsp_fsr1" SLEW = SLOW;
NET "mdsp_fsr1" IOSTANDARD = LVTTL;
NET "mdsp_fsr1" DRIVE = 12;
NET "mdsp_fsr1" SLEW = SLOW;
NET "mdsp_int_n" IOSTANDARD = LVTTL;
NET "prm_ctrl_out[0]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[0]" DRIVE = 12;
NET "prm_ctrl_out[0]" SLEW = SLOW;
NET "prm_ctrl_out[1]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[1]" DRIVE = 12;
NET "prm_ctrl_out[1]" SLEW = SLOW;
NET "prm_ctrl_out[2]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[2]" DRIVE = 12;
NET "prm_ctrl_out[2]" SLEW = SLOW;
NET "prm_ctrl_out[3]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[3]" DRIVE = 12;
NET "prm_ctrl_out[3]" SLEW = SLOW;
NET "prm_ctrl_out[4]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[4]" DRIVE = 12;
NET "prm_ctrl_out[4]" SLEW = SLOW;
NET "prm_ctrl_out[5]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[5]" DRIVE = 12;
NET "prm_ctrl_out[5]" SLEW = SLOW;
NET "prm_ctrl_out[6]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[6]" DRIVE = 12;
NET "prm_ctrl_out[6]" SLEW = SLOW;
NET "prm_ctrl_out[7]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[7]" DRIVE = 12;
NET "prm_ctrl_out[7]" SLEW = SLOW;
NET "prm_ctrl_out[8]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[8]" DRIVE = 12;
NET "prm_ctrl_out[8]" SLEW = SLOW;
NET "prm_ctrl_out[9]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[9]" DRIVE = 12;
NET "prm_ctrl_out[9]" SLEW = SLOW;
NET "prm_ctrl_out[10]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[10]" DRIVE = 12;
NET "prm_ctrl_out[10]" SLEW = SLOW;
NET "prm_ctrl_out[11]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[11]" DRIVE = 12;
NET "prm_ctrl_out[11]" SLEW = SLOW;
NET "prm_ctrl_out[12]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[12]" DRIVE = 12;
NET "prm_ctrl_out[12]" SLEW = SLOW;
NET "prm_ctrl_out[13]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[13]" DRIVE = 12;
NET "prm_ctrl_out[13]" SLEW = SLOW;
NET "prm_ctrl_out[14]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[14]" DRIVE = 12;
NET "prm_ctrl_out[14]" SLEW = SLOW;
NET "prm_ctrl_out[15]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[15]" DRIVE = 12;
NET "prm_ctrl_out[15]" SLEW = SLOW;
NET "prm_ctrl_out[16]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[16]" DRIVE = 12;
NET "prm_ctrl_out[16]" SLEW = SLOW;
NET "prm_ctrl_out[17]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[17]" DRIVE = 12;
NET "prm_ctrl_out[17]" SLEW = SLOW;
NET "prm_ctrl_out[18]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[18]" DRIVE = 12;
NET "prm_ctrl_out[18]" SLEW = SLOW;
NET "prm_ctrl_out[19]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_out[19]" DRIVE = 12;
NET "prm_ctrl_out[19]" SLEW = SLOW;
NET "prm_ctrl_in[0]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[1]" IOSTANDARD = LVCMOS25;
NET "prm_ctrl_in[2]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[0]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[1]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[2]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[3]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[4]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[5]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[6]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[7]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[8]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[9]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[10]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[11]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[12]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[13]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[14]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[15]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[16]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[17]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[18]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[19]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[20]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[21]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[22]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[23]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[24]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[25]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[26]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[27]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[28]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[29]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[30]" IOSTANDARD = LVCMOS25;
NET "prm_data_in[31]" IOSTANDARD = LVCMOS25;
NET "prm_strobe[0]" IOSTANDARD = LVCMOS25;
NET "prm_strobe[1]" IOSTANDARD = LVCMOS25;
NET "ddr2_a[0]" IOSTANDARD = SSTL18_II;
NET "ddr2_a[1]" IOSTANDARD = SSTL18_II;
NET "ddr2_a[2]" IOSTANDARD = SSTL18_II;
NET "ddr2_a[3]" IOSTANDARD = SSTL18_II;
NET "ddr2_a[4]" IOSTANDARD = SSTL18_II;
NET "ddr2_a[5]" IOSTANDARD = SSTL18_II;
NET "ddr2_a[6]" IOSTANDARD = SSTL18_II;

```


Main components in the BOM

Quantity	Instance name	Component	Package	Code
1	SW2	SW PUSHBUTTON	BOURNS_SDTM	Tyco FSM2JAH
1	SW3	SW DIP-10	SW_DIP10	Tyco GDH10S04 (1.27 mm)
3	SW4,SW8,SW9	SW DIP-2	SW_DIP2	Tyco GDH02S04 (1.27 mm)
2	SW5,SW6	A6A-16RF	SW_ROTARY	Digikey A6A-16RF-ND
1	SW10	SW DIP-4	SW_DIP4	APEM NHDS04T (1.27 mm)
1	S1	SO DIMM 144 Socket	SODIMM-144	RS France 670-4442
3	U1,U3,U5	PTH04T220WAD	THROUGH_HOLE	RS 661-7834
1	U2	LM3880MFE-1AB	SOT23-6	Digikey LM3880MF-1AECT-ND
2	U4,U6	PTH05050W	THROUGH_HOLE	RS 661-7904
1	U7	PTH04T230W	THROUGH_HOLE	RS 661-7831
1	U8	PTH08T240W	THROUGH_HOLE	RS 661-7991
2	U9,U10	TPS75625	TO220-5	RS 661-6415
1	U11	PTH08T240W	THROUGH_HOLE	RS 661-7991
2	U12,U15	TPS75618	SOT223-5	RS 661-4096
2	U13,U14	REF3330	SOT23-3	RS 661-9489
4	U16,U17,U47,U48	TPS51100DGQ	DGQ-10	Digikey 296-23595-2-ND
3	U18,U34,U36	74LVT162245B	TSSOP-48	RS 662-6765
6	U20,U33,U35,U37,U38,U44	74LVT162244B	TSSOP-48	Digikey 74LVT162244MTD-ND
1	U39	74LVTH125M	TSSOP-14	Digikey 74LVTH125MTCXTR-ND
2	U40,U42	7407	SOIC-14	Digikey 296-1018-2-ND
1	U41	74FCT138/SO	SO16-1	RS 663-0672
1	U43	74LVX14	TSSOP-14	74LVX14MTCXTR-ND
1	U45	MAX6315US44D3-T	SOT-143	RS 173-436
1	U46	MAX6315US31D3-T	SOT-143	RS 173-442
2	U49,U50	MT47H128M16RT-25E	FBGA-84	MT47H128M16RT-25E:C
1	U51	TMS320C6201GJC200	S-PBGA-N352	TMS320C6201GJCA200
1	U52	SST39VF040-70-4C-NH	PLCC-32	SST39VF040-70-4C-NH
3	U53,U54,U55	DP83865	PQFP-128	RS 504-4803
1	U56	FLASH_SPI	CASON-8	Atmel AT45DB642D
1	U57	ispCLOCK5620A	TQFP-100	Lattice semiconductor
1	U58	SY100EL92ZC	SOIC-20	Micrel
1	U59	CB3LV-3C-40M0000	4-SMD, No Lead	Digikey CTX280LVCT-ND
1	U60	XC6SLX45FGG484	FGG484	RoHS
1	U61	XCF16PVO48	VO48	PROM Xilinx
2	U62,U63	XC6SLX150FGG900	FGG900	RoHS
3	U64,U66,U75	XCF32PVO48	VO48	PROM Xilinx

2	U65,U67	XCF08PVO48	VO48	PROM Xilinx
4	U68,U69,U70,U71	CY7C1370D-250AXC-ND	TQFP-100	Digikey CY7C1370D-250AXC-ND
1	U72	tterq J1	STRIP-25X2 femmina	2.54 mm vertical female strip
1	U73	tterq J2	STRIP-25X2 femmina	2.54 mm vertical female strip
1	U74	xc5vfx70tff1136	FF1136	RoHS
1	U78	TPS75633	TO-263-5	RS 661-4100
3	Y1,Y2,Y3	25MHz 50ppM	SMD Crystal 12.5x4.6mm	RS 703-1887

References

- [1] <https://twiki.cern.ch/twiki/bin/view/Main/AtlasSiliconRodGroup>
- [2] <https://twiki.cern.ch/twiki/bin/viewauth/Atlas/IBLreadout>
- [3] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.34.3055&rep=rep1&type=pdf>