NEC

Preliminary User's Manual

*start*WARE-GHS-VR4133

Starter Kit VR4133

Document No. U16916EE2V0UM00
Date Published March 2004

Printed in Germany

- The information in this document is current as of 04.03, 2004. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC sales representative for availability and additional information.

- No part of this document may be copied or reproduced in any form or by any means without prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.

- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such NEC Electronics products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.

- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of customer's equipment shall be done under the full responsibility of customer. NEC Electronics no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.

- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.

- NEC Electronics products are classified into the following three quality grades: “Standard”, “Special” and “Specific”.

The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated “quality assurance program” for a specific application. The recommended applications of NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

"Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.

"Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).

"Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is “Standard” unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact NEC Electronics sales representative in advance to determine NEC Electronics 's willingness to support a given application.

Notes: **1.** " NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.

2. " NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02.10

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics America Inc.
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC Electronics (Europe) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 1101
Fax: 0211-65 03 1327

Sucursal en España
Madrid, Spain
Tel: 091- 504 27 87
Fax: 091- 504 28 60

Succursale Française
Vélizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

Filiale Italiana
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

Branch The Netherlands
Eindhoven, The Netherlands
Tel: 040-244 58 45
Fax: 040-244 45 80

Branch Sweden
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

United Kingdom Branch
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.
Singapore
Tel: 65-6253-8311
Fax: 65-6250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377
Fax: 02-2719-5951

Preface

Readers

This manual is intended for users who want to understand the functions of the *start*WARE-GHS-VR4133 Starter Kit.

Purpose

This manual presents the hardware manual of *start*WARE-GHS-VR4133 Starter Kit..

Organization

This system specification describes the following sections:

- Board Features
- Detailed Functional Description
- Board Operation

Legend

Symbols and notation are used as follows:

Weight in data notation	:	Left is high-order column, right is low order column
Active low notation	:	$\overline{xxx}$ (pin or signal name is over-scored) or /xxx (slash before signal name)
Memory map address:	:	High order at high stage and low order at low stage
Note	:	Explanation of (Note) in the text
Caution	:	Item deserving extra attention
Remark	:	Supplementary explanation to the text
Numeric notation	:	Binary . . . xxxx or xxxB Decimal . . . xxxx Hexadecimal . . . xxxxH or 0x xxxx

Prefixes representing powers of 2 (address space, memory capacity)

K (kilo): $2^{10} = 1024$

M (mega): $2^{20} = 1024^2 = 1,048,576$

G (giga): $2^{30} = 1024^3 = 1,073,741,824$

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 System Requirements

Host PC	A PC supporting Windows 9x, Windows 2000 or Windows NT is required for Green Hills MULTI® 2000 Embedded Development Kit. Pentium 133 MHz (at least), 32 MB of RAM, 256-color display (1024 × 768), mouse, CD-ROM drive and 60 Mbytes of free hard disk space are required to install the GHS compiler and debugger package.
Host interface	Serial (RS232C) interface capable to handle communication at 9600, 19200, 38400 or 115200 baud.

1.2 Package Contents

Please verify that you have received all parts listed in the package contents list attached to the *start*-WARE-GHS-VR4133 package. If any part is missing or seems to be damaged, please contact the dealer from whom you purchased your *start*WARE-GHS-VR4133.

Note: Updates to this User Manual, additional documentation and/or utilities for *start*WARE-GHS-VR4133, if available, may be downloaded from the NEC WEB page(s) *http://www.nec.de/support*.

1.3 Related Documents

VR4133 Preliminary User's Manual Hardware, NEC Doc. Number U16620EJ3V0UM00
VR4133 Preliminary Data Sheet, NEC Doc. Number U16551EJ1V0DS00
VR4100 Series User's Manual Architecture, NEC Doc. Number U15509EJ2V0UM00
Green Hills Documentation is provided with the installation of the MULTI® software.

1.4 Used Abbreviations

There are some abbreviations used in this document, which may require additional information to be understood correctly.

- RFU - reserved for future use
- NC - not connected
- GND - ground
- GHS - Green Hills
- GNU - S/W supplied under General Public License (GLP)
- MIPS - Microprocessor without interlocked pipeline stages

Chapter 2 Board Features

*Start*ware-GHS-VR4133 is a low-cost evaluation board for NEC's VR4133 64-bit high-performance microprocessor. It allows evaluation of the processor's performance as well as potential system performance, because the VR4133 can also be operated together with its typical system environment.

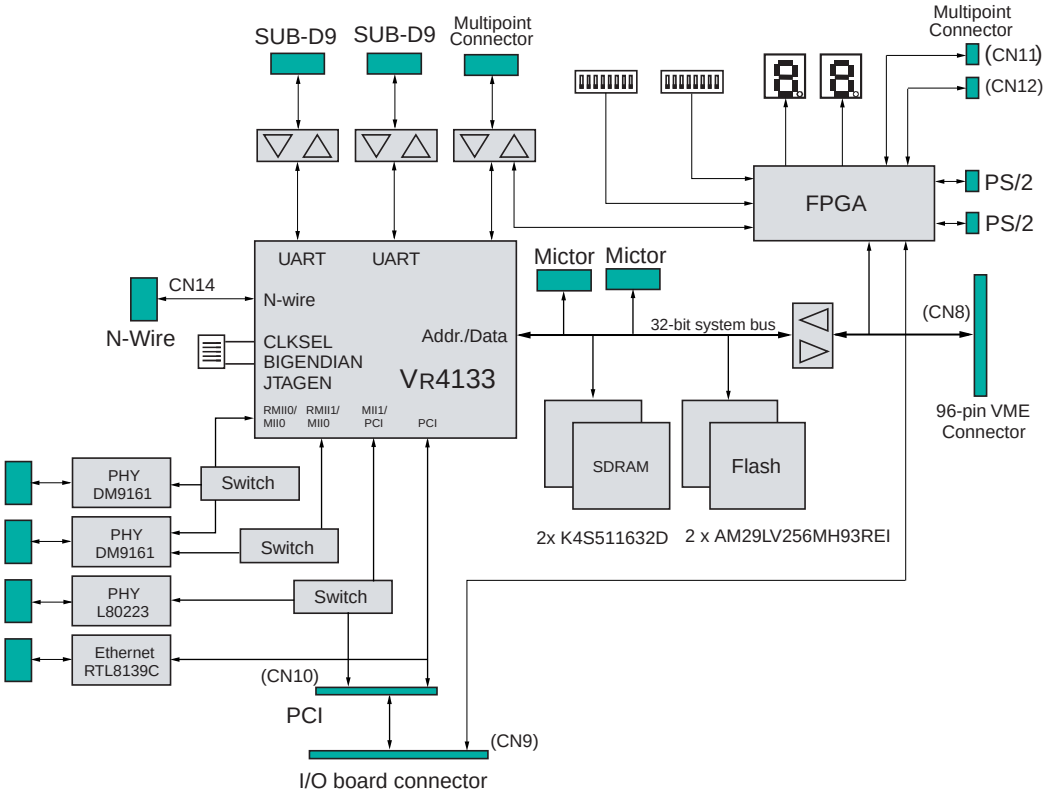
The *start*WARE-GHS-VR4133 board has a VR4133-266 processor soldered; processor speed is configurable via jumpers. The VR4133 is directly connected to 128 MByte of SDRAM, built up with 2 K4S511632D devices (total bus width 32-bit). The SDRAM supports full 133 MHz SDRAM clock. 64 MByte of flash are connected to the VR4133 memory bus directly. Mictor connectors are attached to the memory bus before the isolation buffer. A short summary of the board features is given below:

- VR4133-266 CPU
- 64 MByte Flash
- 128 MByte SDRAM
- N-wire debug interface (via KEL connector as used with the Midas ICE)
- 2 serial interfaces on conventional SUB-D connectors, buffered via RS-232 line drivers up to 3 Ethernet Interfaces (2 Ethernet Interfaces supported by the VR4133 and one Ethernet Interface realized with the Realtek RTL8139C controller)
- Clock-synchronous interface extended with 2 special input and 2 special output signals on a simple multipoint connector
- IrDA module – TFDS6502
- PCI bus on a standard PCI slot connector (32-bit, 66 MHz, R2.3)
- PCI bus with VR4133 specific extensions on a standard 160-pin VME connector
- System bus on a standard 96-pin VME connector
- Mictor Connectors for measurement purpose
- FPGA as an additional I/O device with several I/O ports on two Multipoint connectors
- two 7-segment LED displays for debugging purpose
- 16 DIP switches for selection of SDRAM clock, PCI clock, Monitor and user defined functions
- Single Voltage Power Supply

The board software consists of three monitor programs:

- A simple download monitor program named "S-Boot". The S-Boot monitor downloads files to the processor's SDRAM and/or Flash memory and optionally starts the program. A standard terminal emulation program can be used on the host PC side for communication.
- The Green Hills target monitor communicates with Green Hills Multi Compiler/Debugger on the host PC. An evaluation license of Multi is included in the package.
- The E-Boot download monitor for high-speed downloads via Ethernet.

Figure 2-1: Simplified Block Diagram



Chapter 3 Functional Description

The VR4133 Evaluation board (*start*WARE-GHS-VR4133) is designed to evaluate the VR4133 MIPS RISC processor and uses the Realtek RTL8139C chip for ethernet downloading of software. It can also be used as development platform for operating systems like WinCE, VxWorks, Linux or Integrity. The VR4133 Evaluation board (*start*WARE-GHS-VR4133) is designed to be used either in a stand-alone mode using a 15 V AC adapter or together with a Ravin-E Board as graphic interface and/or an I/O Board to increase the number of I/O interfaces. The VR4133 Evaluation board (*start*WARE-GHS-VR4133) provides 128 MB of SDRAM memory with two 512 Mbit chips and 64 MB of Flash memory with two 256 Mbit chips.
The I/O capabilities of the VR4133 Evaluation board (*start*WARE-GHS-VR4133) as stand alone board include 2 on-chip UARTs, an Ethernet Interface realized with RTL8139C, a PCI interface, a clock synchronous, serial interface, Real Time Clock (RTC) timers, and DMA channels. The FGPA increases the number of general purpose I/O pins. These boards are shipped with the NEC S-Boot monitor and two other monitors pre-installed in the Flash memory.

3.1 Flash Memory

The VR4133 Evaluation board (*start*WARE-GHS-VR4133) has 64 MByte on-board Flash memory where the user can store data or transfer code to. The system normally boots from the on-board Flash memory. The Flash memory is mapped at 0xBC00 0000 through 0xBFFF FFFF. In order to boot from Flash memory, the code must start at the physical address 0xBFC0 0000. Thus only the upper 4 MByte of Flash memory are available to the boot code. Flash memory is accessed via the VR4133 chip selects ROMCS1#.
Write operations to the Flash memory by VR4133 program execution are possible and controlled by the BCUCNTREG1 register at address 0xAF00 0000 in the VR4133. Typically the pre-installed monitor programs will be used for writing to Flash; however it is also possible to do this in a user application program.

3.2 Main Memory

The VR4133 Evaluation board (*start*WARE-GHS-VR4133) contains 128 MByte of main memory implemented in one physical bank and realized with two 512 Mbit SDRAM chips connected to CS0#. The physical address map for this memory is 0x0000 0000 to 0x07FF FFFF. The CPU address is 0xA000 0000 to 0xA7FF FFFF for uncached or 0x8000 0000 to 0x87FF FFFF for cached accesses.

3.3 FPGA Functions

The FPGA controls the power up and power down sequences and realizes several ports for predefined and user definable data I/O. Furthermore the FPGA routs the PCI interrupts INTA# to INTD# to GPIOs of the VR4133.
Predefined ports are the LED display, the DIP switches, two PS/2 interfaces (for keyboard and mouse), two I²S interfaces and the extension lines of the CSI interface. The other ports can be used as user definable GPIOs or to realize an additional connection between the FPGA and the VR4133.
The following block and state diagram show the functionality of the FPGA. The block diagram shows the part of the I/O lines except the power control logic. The functionality of this part is shown in the state diagram.

Figure 3-1: FPGA Block Diagram - I/O Circuits

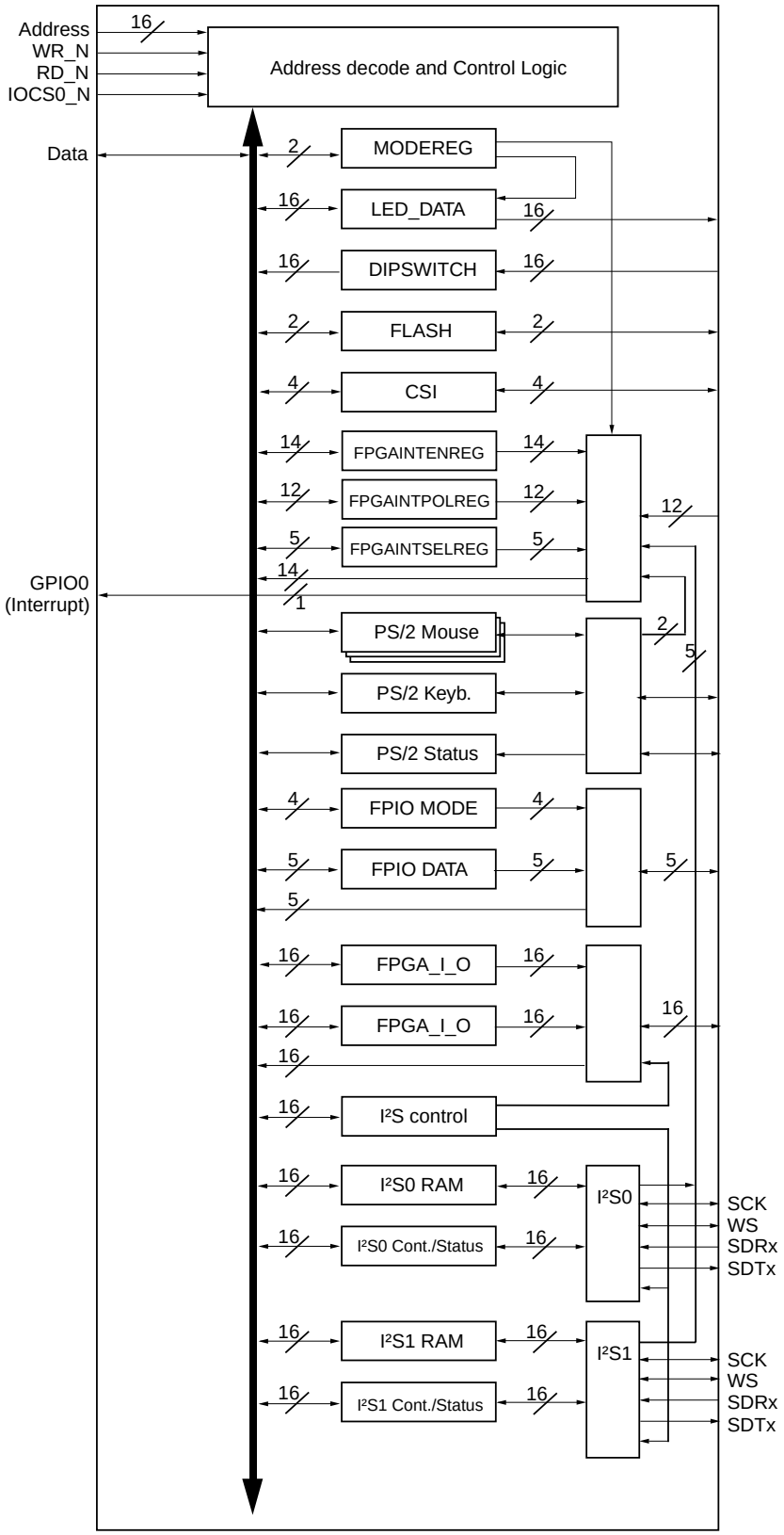
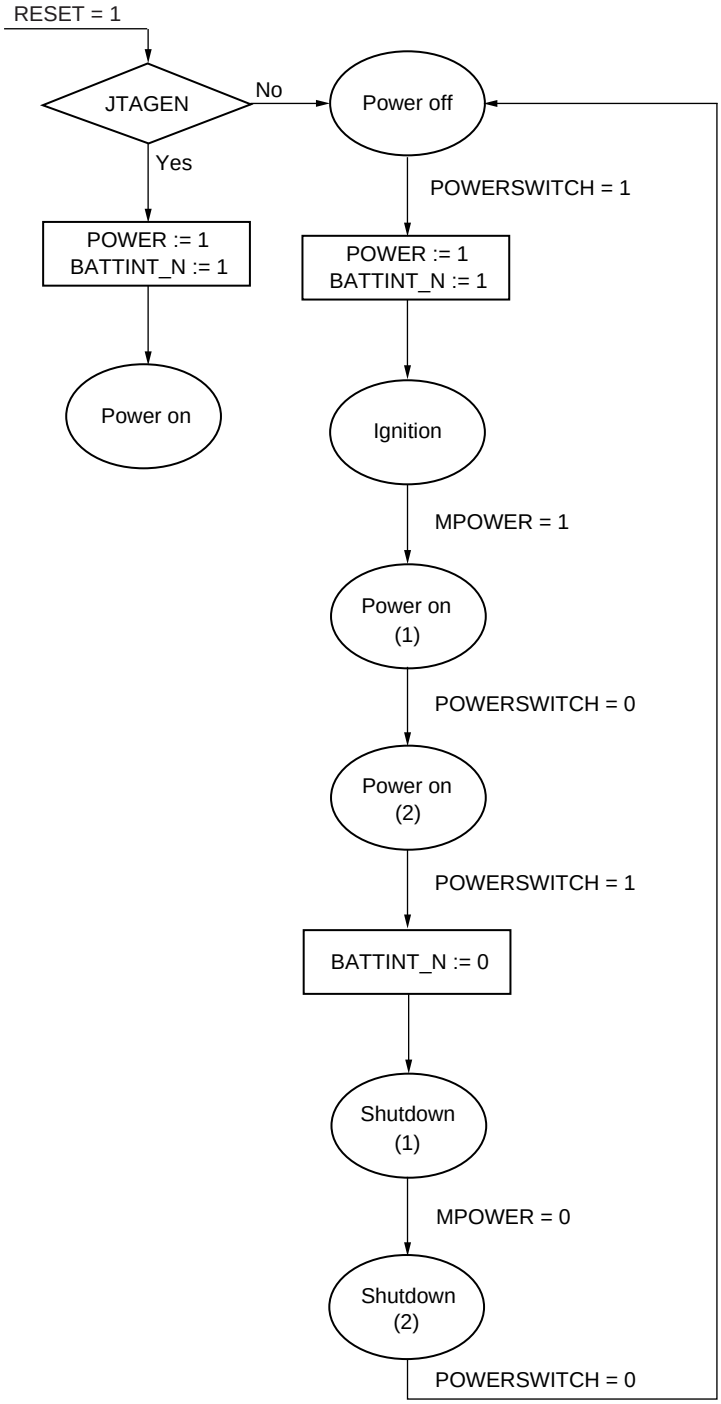


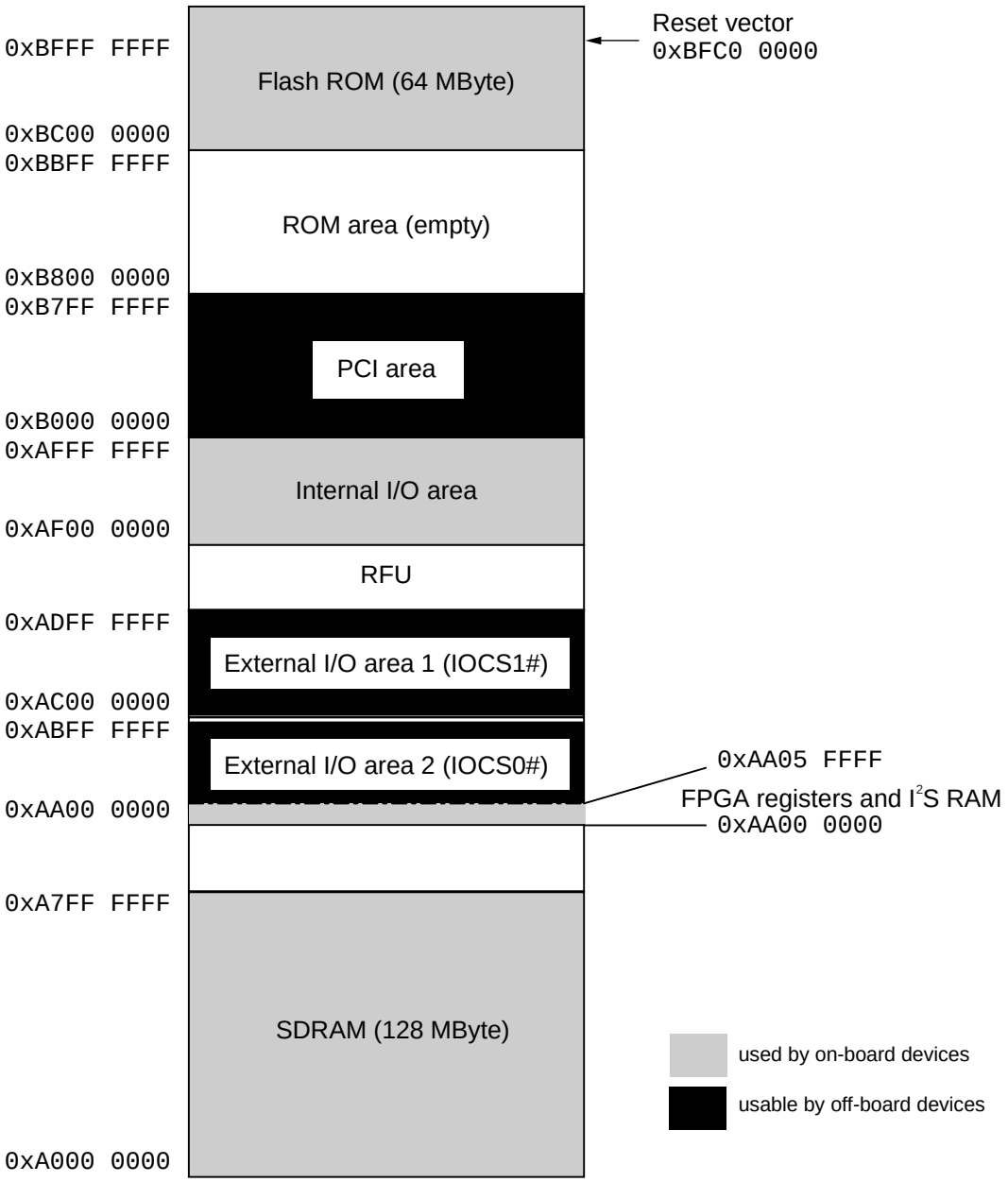
Figure 3-2: FPGA State Diagram – Power on Logic



3.4 Memory Mapping

The VR4133 processor regards external devices like the FPGA on the evaluation board as memory mapped I/O, which is controlled with one of the external chip select signals, IOCS0#. An access to an FPGA register is executed in the same way than a memory access. The 32-bit physical address space encompasses a total of 4 GByte; however these 4 GByte consist of 8 mirror images of a 512 MByte space subdivided in SDRAM, ROM and I/O spaces. The most significant address bits control, if the 512 MByte space is accessed cached or uncached respectively mapped or unmapped.
The following diagram shows the memory map for the *start*WARE-GHS-VR4133 board for uncached and unmapped addresses (kseg1) ranging from `0xA000 0000` to `0xBFFF FFFF`. Note that the same structure is seen for other address space segments like kseg0 (`0x8000 0000` to `0x9FFF FFFF`). Consult the processor documentation for details.

Figure 3-3: startWARE-GHS-VR4133 address map for kseg1 (unmapped, uncached)



Chapter 4 Detailed Functional Description

4.1 Usage of VR4133 GPIO Pins

Several GPIO pins of the VR4133 are directly used by the FPGA and other circuitry; therefore they can not be used as user defined I/O pins. There is a total of 2 GPIOs freely usable and 2 GPIOs can be used under certain circumstances (no Board inserted in CN8; no interrupt from the FPGA); 4 GPIOs can additionally be used, if the debug serial I/O of the VR4133 is not used.
The following table lists the usage of all GPIO pins:

Table 4-1: VR4133 GPIO pin usage (1/2)

GPIO	Available in mode	Proposed usage on *start*WARE-VR4133
GPIO0	all	is defined as a general interrupt input (default setting). The interrupt cause can be read in the FPGAINTREG. For using GPIO0 as free usable GPIO, the interrupt mode of the FPGA can be switched off.
GPIO1	all	is connected to CN8 and CN11 for usage as GPIO. If there is no connection via CN8 (Ravin Board or other), it can be used as GPIO at CN11.
GPIO2	all	is connected to CN11 for usage as GPIO
GPIO3	all	is connected to CN11 for usage as GPIO
GPIO4	D	is shared with Ethernet I/F pins M0RXD1/R0RXD1; not usable as GPIO on *start*WARE-VR4133
GPIO5	D	is shared with Ethernet I/F pins M0TXD3/R1TXD1; not usable as GPIO on *start*WARE-VR4133
GPIO6	all	is configured as SYSDIR signal
GPIO7	D	is shared with Ethernet I/F pins M0RXCLK/R1CRSDV; not usable as GPIO on *start*WARE-VR4133
GPIO8	D	is shared with Ethernet I/F pins M0RXDV/R0CRSDV; not usable as GPIO on *start*WARE-VR4133
GPIO9	D	is shared with Ethernet I/F pins M0MDC/R0MDC; not usable as GPIO on *start*WARE-VR4133
GPIO10	D	is shared with Ethernet I/F pins M0RXER/R1MDC; not usable as GPIO on *start*WARE-VR4133
GPIO11	B,D	is shared with Ethernet I/F pinM0COL; not usable as GPIO on *start*WARE-VR4133
GPIO12	D	is shared with Ethernet I/F pins M0MD/R0MD; not usable as GPIO on *start*WARE-VR4133
GPIO13	D	is shared with Ethernet I/F pins M0TXD1/R0TXD1; not usable as GPIO on *start*WARE-VR4133
GPIO14	D	is shared with Ethernet I/F pins M0TXD0/R0TXD0; not usable as GPIO on *start*WARE-VR4133
GPIO15	all	configured as DCD# signal
GPIO16 - GPIO31	all	is used as Data 16 –31 for the 32 Bit Data Bus
GPIO32 - GPIO35	all	is used as Debug Serial Interface; can be defined as GPIO, if the connections at CN29 are open
GPIO36	D	is shared with Ethernet I/F pins M0TXEN/R0TXEN; not usable as GPIO on *start*WARE-VR4133
GPIO37	D	is shared with Ethernet I/F pins M0RXD3/R1RXD1; not usable as GPIO on *start*WARE-VR4133

Table 4-1: VR4133 GPIO pin usage (2/2)

GPIO	Available in mode	Proposed usage on *start*WARE-VR4133
GPIO38	D	is shared with Ethernet I/F pins M0RXD0/R0RXD0; not usable as GPIO on *start*WARE-VR4133
GPO39-GPO46	C	is shared with PCI I/F pins AD18-26 and available on multipoint connector CN11
GPO47	C	is shared with PCI I/F pins AD18-26; not usable as GPIO on *start*WARE-VR4133

4.2 Jumper Settings

Before power-on, the board should be configured through the jumpers. The jumper settings are as shown in the following tables. The default setting is marked as a shaded area in the column entitled “Function”; an example for the default position 2-3 is given below:

Table 4-2: Example for default jumper setting marking

No.	Pos.	Function
1		Default
2		
3		
1		Not default
2		
3		

An overview about the placement of the jumpers on the evaluation board is shown in Figure 4-1 at the end of section 4.2 ”Jumper Settings” on page 23.

4.2.1 Pipeline Clock Setting

The VR4133 uses three pins (CLKSEL(2:0)), to configure the pipeline clock; these pins are sampled when the RTCRST# input is de-asserted. JP6, JP15 and JP16 are connected to these pins and select the pipeline clock as described in Table 4-3 below:

Table 4-3: Pipeline clock setting for VR4133

CLKSEL2 (JP16)	CLKSEL1 (JP15)	CLKSEL0 (JP6)	PClock	VTClock, SCLK				
				Div2	Div3	Div4	Div5	Div6
1	1	1	RFU	RFU	RFU	RFU	RFU	RFU
1	1	0	RFU	RFU	RFU	RFU	RFU	RFU
1	0	1	RFU	RFU	RFU	RFU	RFU	RFU
1	0	0	265.9 MHz	133.0 MHz	88.6 MHz	66.5 MHz	53.2MHz	44.3 MHz
0	1	1	199.1 MHz	99.5 MHz	66.4 MHz	49.8 MHz	39.8 MHz	RFU
0	1	0	165.9 MHz	82.9 MHz	55.3 MHz	41.5 MHz	RFU	RFU
0	0	1	149.0 MHz	74.5 MHz	49.7 MHz	37.2 MHz	RFU	RFU
0	0	0	133.0 MHz	66.5 MHz	44.3 MHz	RFU	RFU	RFU

The corresponding jumper positions for JP16, JP15 and JP6 look as shown in Table 4-4, “Jumper positions for pipeline clock setting,” on page 24.

Table 4-4: Jumper positions for pipeline clock setting

	JP6	JP15	JP16	PClock
3				RFU
2				
1				
3				RFU
2				
1				
3				RFU
2				
1				
3				265.9 MHz
2				
1				
3				199.1 MHz
2				
1				
3				165.9 MHz
2				
1				
3				149.0 MHz
2				
1				
3				133.0 MHz
2				
1				

4.2.2 JTAG enable

JP23 enables/disables the N-Wire-Interface (JTAGEN) of the VR4133. If N-Wire / JTAG interface is enabled, the VR4133 shows different power on behaviour than in the “normal operating mode”. The MPOWER signal of the VR4133 is directly driven high and all needed Voltages supply the *start*WARE-GHS-VR4133. The POWER signal is generated automatically and the VR4133 is able to run from address 0xbcf00000.

Table 4-5: Jumper setting to enable N-Wire

	JP23	N-Wire interface Enable
3		N-Wire interface disabled
2		
1		
3		N-Wire interface enabled
2		
1		

4.2.3 Endianness Setting

Jumper JP26 selects the endianness as described below:

Table 4-6: Jumper positions for endianness setting

	JP26	Endianness
3		Little Endian
2		
1		
3		Big Endian
2		
1		

4.2.4 Data Bus Width Setting

The VR4133 can work with a 32-bit wide and with a 16-bit wide memory system. A 16-bit wide bus configuration reduces system performance but frees 16 data lines which can then be used as GPIOs. Selection is done via the DBUS32 pin which is sampled, when RTCRST# is de-asserted. Jumper JP24 selects the data bus width as described below:

Table 4-7: Jumper positions for data bus width setting

	JP24	Data bus width
3		16 bit data bus
2		
1		
3		32 bit data bus
2		
1		

4.2.5 RTC Clock Select Setting

VR4133 supports two different modes for RTC clock supply, which are selected using the RTCCLKSEL pin of VR4133. Both start-up supply modes are supported. The RTCCLKSEL pin has a jumper (JP27) for selectable pull up/down. Another jumper (JP7) allows reducing the reset time of the reset generator.

Table 4-8: Jumper positions for Start-up mode

	JP27		JP7	RTC Clock Supply
3		2		“Normal” Mode
2				
1		1		
3		2		“Fast start up” Mode
2				
1		1		

4.2.6 PCI Enable Setting

The VR4133 processor supports four different operation modes with different interface configurations:

Table 4-9: Jumper positions for PCI Enable setting

VR4133 mode	Interface configuration
A	PCI + MII0
B	PCI + RMII0 + RMII1
C	MII0 + MII1
D	PCI (VR4133 “compatible” mode)

Selecting one of these modes has to be done by hardware – detected during the power up and by software (initialization of register). The PCIEN input of the VR4133 signal enables the PCI interface. This signal is sampled when the RTCRST# signal becomes high level from low level.

Table 4-10: Jumper positions for PCI Enable setting

	JP25	PCI Enable
3		PCI disabled
2		
1		
3		PCI enabled
2		
1		

Note: This setting is completely independent from the data bus width configuration.

4.2.7 MII/RMII Setting

The VR4133 MAC0 interface connect to the responding on-board PHY device either via MII or RMII interface. JP14 selects which kind of interface is used after the start-up of the board. The actual jumper setting can be read in the corresponding register of the FPGA.

Table 4-11: Jumper positions for MII/RMII setting

	JP14	MII/RMII
3		MII selected
2		
1		
3		RMII selected
2		
1		

4.2.8 Jumpers for Ethernet Channel 1 Setting

The configuration of Ethernet Channel 1 has to be done by three jumpers, which allow the select the operation mode and the speed of the Ethernet Connection. The Auto Negotiations mode can be selected by JP1. In case of Auto Negotiation On the setting of the Speed jumper (JP8) is irrelevant.

Table 4-12: Jumper positions for Ethernet Channel 1 setting

	JP1	ANEG
3		Auto Negotiation off
2	■	
1	■	
3	■	Auto Negotiation on
2	■	
1		

	JP5	DPLX
3		Half Duplex
2	■	
1	■	
3	■	Full Duplex
2	■	
1		

	JP8	SPEED
3		10 Mbits/s
2	■	
1	■	
3	■	100 Mbits/s
2	■	
1		

4.2.9 Jumpers for Ethernet Channel 2 Setting

For realizing the Ethernet Channel 2 a DM 9161 from DAVICOM is used. This Ethernet PHY is able to support several operation modes. It is able to handle full and half duplex with a data rate of 10 or 100 Mbit/s. The setting will be done by jumper JP17, JP18 and JP19. The meaning of each combination is shown below.

Table 4-13: Jumper positions for Ethernet Channel 2 setting

	JP19	JP18	JP17	
3				Dual Speed 100/10 HDX
2	■	■	■	
1	■	■	■	
3			■	Reserved
2	■	■	■	
1	■	■		
3		■		Reserved
2	■	■	■	
1	■		■	
3		■	■	Manually Select 10TX HDX
2	■	■	■	
1	■			
3	■			Manually Select 10TX FDX
2	■	■	■	
1		■	■	
3	■		■	Manually Select 100TX HDX
2	■	■	■	
1		■		
3	■	■		Manually Select 100TX FDX
2	■	■	■	
1			■	
3	■	■	■	Auto – negotiation Enables all Capabilities
2	■	■	■	
1				

4.2.10 Jumpers for Ethernet Channel 3 Setting

For realizing the Ethernet Channel 2 a DM 9161 from DAVICOM is used. This Ethernet PHY is able to support several operation modes. It is able to handle full and half duplex with a data rate of 10 or 100 Mbit/s. The setting will be done by jumper JP20, JP21 and JP22. The meaning of each combination is shown below.

Table 4-14: Jumper positions for Ethernet Channel 3 setting

	JP22	JP21	JP20	
3				Dual Speed 100/10 HDX
2				
1				
3				Reserved
2				
1				
3				Reserved
2				
1				
3				Manually Select 10TX HDX
2				
1				
3				Manually Select 10TX FDX
2				
1				
3				Manually Select 100TX HDX
2				
1				
3				Manually Select 100TX FDX
2				
1				
3				Auto – negotiation Enables all Capabilities
2				
1				

4.2.11 Jumpers for Power Supply Control

Some jumpers on the evaluation board are just for current measurement purposes; they disconnect parts of the power supply completely when they are not present. JP2, JP3, JP4, JP9, JP10 and JP11 belong to this category. Please consult the circuit diagram for the influence of the different supply voltages. These jumpers control the supply voltages as follows:

Table 4-15: Jumper positions for power supply control

	JP2	VDD3.3P
2		VDD3.3P not supplied
1		
2		VDD3.3P supplied
1		

	JP3	VDD1.5P
2		VDD1.5P not supplied
1		
2		VDD1.5P supplied
1		

	JP4	VDD5.0
2		VDD5.0 not supplied
1		
2		VDD5.0 supplied
1		

	JP9	VDD3.3SDRAM
2		VDD3.3SDRAM not supplied
1		
2		VDD3.3SDRAM supplied
1		

	JP10	VDD3.3
2		VDD3.3 not supplied
1		
2		VDD3.3 supplied
1		

	JP11	VDD1.5
2		VDD1.5 not supplied
1		
2		VDD1.5 supplied
1		

4.2.12 Jumpers for Debug SIU and GPIOs

The pins for the debug serial interface of the VR4133 are shared with general purpose I/O pins and can be configured to either usage with the jumpers at CN29. They are normally configured for usage are debug SIU, as the S-Boot monitor uses this interface.

Table 4-16: Debug serial interface usage

CN29										Debug serial interface usage
1		3		5		7		9		Debug Serial Interface activated
2		4		6		8		10		
1		3		5		7		9		Debug Serial Interface not activated
2		4		6		8		10		

4.2.13 Jumpers for Ethernet Connector Shield Grounding

Several jumpers allow a direct connection between Ethernet connector shield and GND. Leave all these jumpers open.

Table 4-17: Ethernet Connector Shield Grounding

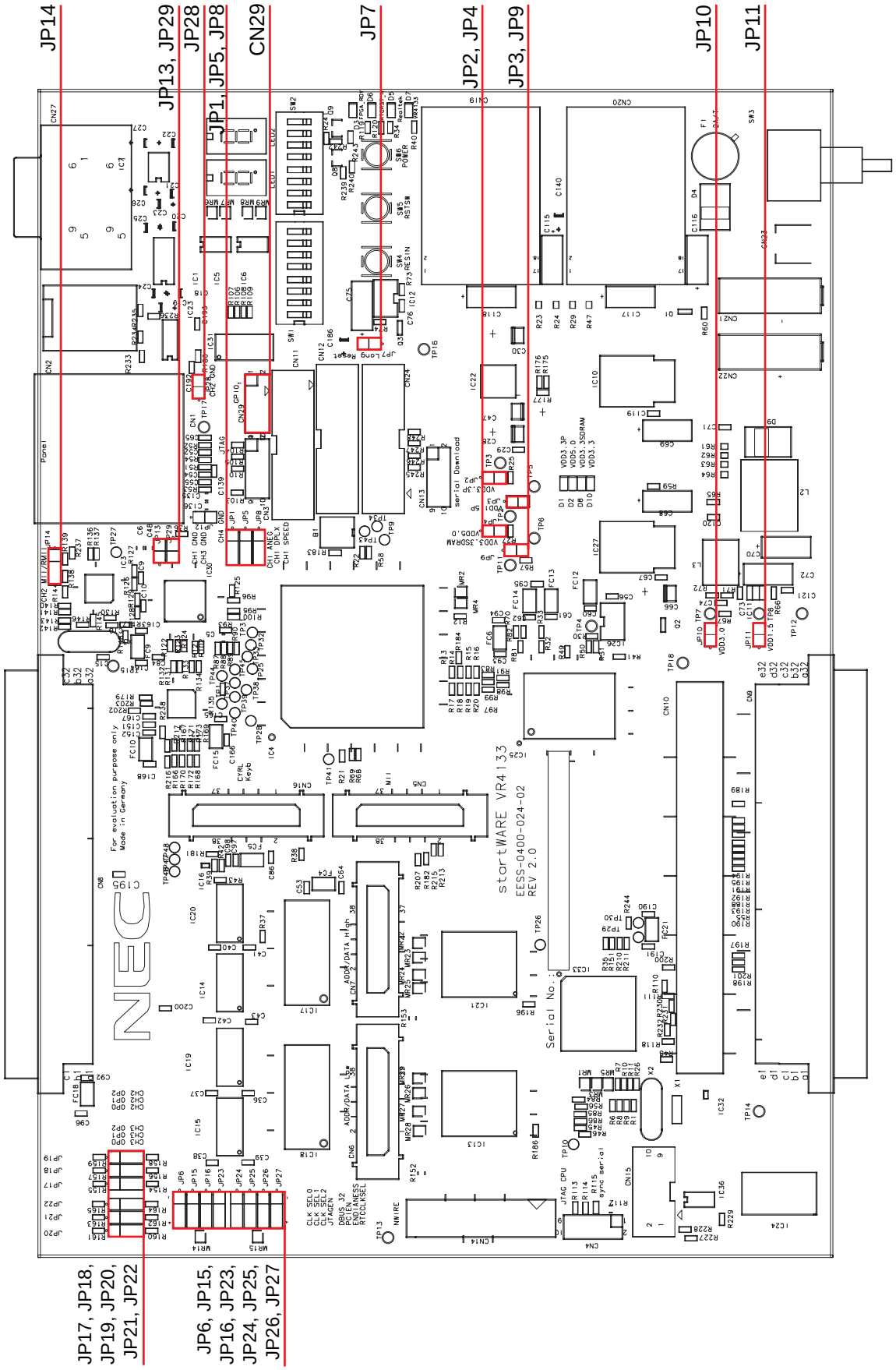
	JP13	GND_CH1
2		No direct connection with GND
1		
2		connection with GND
1		

	JP28	GND_CH2
2		No direct connection with GND
1		
2		connection with GND
1		

	JP29	GND_CH3
2		No direct connection with GND
1		
2		connection with GND
1		

	JP12	GND_CH4
2		No direct connection with GND
1		
2		connection with GND
1		

Figure 4-1: Jumper positions



4.3 Switch Settings

4.3.1 DIP Switch Settings

The VR4133 Evaluation Board has two 8-bit DIP-switches to define several settings.

Figure 4-2: DIP switch setting for SW1 and SW2

SW1 is defined as follows:

Table 4-18: DIP switch setting for SW1

SW1-1	SW1-2	Boot Monitor
OFF	OFF	S-Boot Monitor
OFF	ON	Jump to Flash memory base address
ON	OFF	Green Hills Monitor
ON	ON	E-Boot Monitor

SW1-5	SW1-6	SW1-7	SDRAM Clock/VTClock
OFF	OFF	OFF	re-use setting from CLKSEL pins
OFF	OFF	ON	reserved
OFF	ON	OFF	PClock/2
OFF	ON	ON	PClock/3
ON	OFF	OFF	PClock/4
ON	OFF	ON	PClock/5
ON	ON	OFF	PClock/6
ON	ON	ON	Reserved

SW1-8	FPGA test mode
OFF	Test mode off
ON	Test mode on

Please note that the DIP switches are read by software running on the VR4133 and that the setting according to the switch setting will be done by software as well. Not all combinations between pipeline clock (PClock) and SDRAM clock are allowed; Table 4-3, “Pipeline clock setting for VR4133,” on page 23 shows the details. The currently installed S-Boot Monitor takes care of the clock setting; other software does not necessarily do this.

4.3.2 Other Switch Settings

The board is equipped with 4 more switches related to powering up the board. The main power switch is SW3 on the front side of the board. It directly connects power to the switching regulators and activates LEDs D1and D2. Note that SW3 does not yet start the processor, if the JTAGEN setting is default (JTAGEN disabled). LED D6 indicates that the automatically generated RTCRST is active. While the RTCRST is active, the processor must not be started. The push button SW6 generates a pulse at the VR4133 POWER input and starts up the processor. SW6 should not be pressed before the FPGA configuration is completely terminated, which is indicated by LED D3.

If JTAGEN setting is done to use the N-Wire interface, the startWARE-GHS-VR4133 starts directly without any additional action from user side. LED D6 indicates that the automatically generated RTCRST is active. After the RTCRST is done, the processor needs not be started – it will be started automatically. Within this mode, it is impossible to use the RTCRST button to put the board into the “power off” state.

Push button SW5 generates the RSTSW# signal which is directly connected to the related pin of the processor. SW4 (“RTCRESET”) generates a reset signal that conforms to the VR4133 specification.
The position of the switches on the board can be found in Figure 4.1.

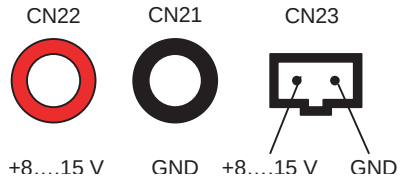
4.4 Connectors

The VR4133 evaluation board provides several connectors for testing purpose and for system extensions. This chapter will describe usage of the connectors one by one.

4.4.1 Power Supply Connectors (CN21, CN22 and CN23)

There are two connector types provided on the board to connect a power supply. You can either use the “classic” laboratory style connectors CN21 and CN22 or the texas style connector CN23. CN23 is used by the power supply that is delivered together with the board. The board consumes typically 250 mA at 15 V (does not include any externally connected boards). Feeding the board with lower voltages increases(!) the supply current due to the switched power supply circuits.

Figure 4-3: Power supply connectors (board front view)

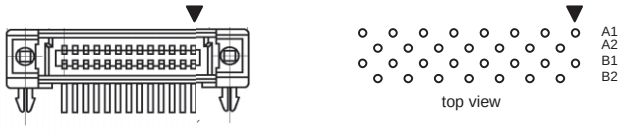


Caution: We urgently recommend to check polarity of the power supply that comes together with the board before connecting it to CN23.

4.4.2 Usage of the N-Wire ICE Connector (CN1)

For using the N-Wire ICE (RTE-1000-TP) connect the ICE to CN1.

Figure 4-4: N-Wire Connector



CN1 is directly connected to the VR4133 N-Wire-Interface.

GND	A1	B1	GND
GND	A2	B2	GND
GND	A3	B3	GND
GND	A4	B4	GND
GND	A5	B5	GND
GND	A6	B6	GND
RMODE_N/JTDI	A7	B7	GND
JTCK	A8	B8	GND
JTMS	A9	B9	GND
JTDO	A10	B10	GND
JTRST_N	A11	B11	NC.
BKTGIO_N	A12	B12	NC.
NC.	A13	B13	VDD3.3

JP23 enables/disables the N-Wire-Interface (JTAGEN) of the VR4133. If N-Wire / JTAG interface is enabled, the VR4133 shows different power on behaviour than in the “normal operating mode”. The MPOWER signal of the VR4133 is directly driven high and all needed Voltages supply the *start*WARE-GHS-VR4133. The POWER signal is generated automatically and the VR4133 is able to run from address 0xbcf00000.

Table 4-19: Jumper setting to enable N-Wire

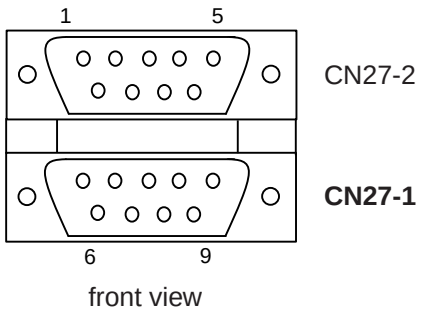
	JP23	N-Wire interface Enable
3		N-Wire interface disabled
2		
1		
3		N-Wire interface enabled
2		
1		

4.4.3 Using the Serial Interface (CN27-1)

The VR4133 has two on-chip serial interfaces, called the SIU (**s**erial **i**nterface **u**nit) and the DSIU (**d**ebug SIU). The SIU interface conforms to the RS232-C communication standard and supports up to 1.15 Mbps. This unit is functionally compatible with the NS16550. The address map for the SIU is 0xAF00 0800 to 0xAF00 080A. For details refer to the VR4133 Users Manual. The processor's serial interface is connected to a RS232-C electrical interface via a level converter. It uses a standard male 9-pin SUB/D connector CN16 on the CPU PCB. The pin definitions are described in the following figure.

Figure 4-5: Serial interface connector CN27-1

1	DCD_N
2	RXD
3	TXD
4	DTR_N
5	GND
6	DSR_N
7	RTS_N
8	CTS_N
9	NC



4.4.4 Usage of Debug Serial Interface (CN27-2)

The pins for the debug serial interface of the VR4133 are shared with GPIOs and are software configurable to either function. Both types of usage are supported by the VR4133 evaluation board; jumpers on CN29 need to be configured accordingly.
If the Debug Serial Interface is not activated, CN29 can be used as connector for connections to GPIO (35:32). These GPIOs are general purpose **outputs** – for details see also the VR4133 Users Manual.

Note: CN29 only affects the routing of the VR4133 signals on the board. The actual configuration is done by software on the VR4133.

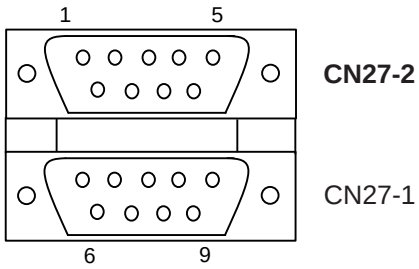
Table 4-20: Debug serial interface usage

CN29										Debug serial interface usage
1		3		5		7		9		Debug Serial Interface activated
2		4		6		8		10		
1		3		5		7		9		Debug Serial Interface not activated
2		4		6		8		10		

The address map for the DSIU is `0xAF00 0820` to `0xAF00 0827`. For details refer to VR4133 User's Manual. The processor's serial interface is connected to a RS232-C electrical interface via a level converter. It uses a standard male 9-pin SUB/D connector CN27-2 on the CPU PCB. The pin definitions are listed in the following table.

Figure 4-6: Serial interface connector CN27-2

1	NC.
2	DDIN
3	DDOUT
4	NC.
5	GND
6	NC.
7	DRTS_N
8	DCTS_N
9	NC.



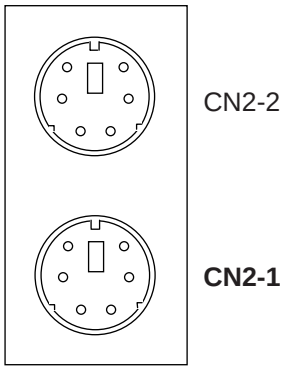
4.4.5 Usage of PS/2 Mouse interface (CN2-1)

The PS/2 Mouse interface is realized by the FPGA. The CPU controls this interface by reading or writing from/to FPGA registers. The implementation of the PS/2 interfaces supports polling and interrupt mode.

The address map for the PS/2 interfaces is `0xAA00 0054` to `0xAA00 0064`. For details refer to section 5.1 ”FPGA Register Set / Memory Map” on page 53. The PS/2 Ready register contains the status of both PS/2 interfaces, Mouse and Keyboard. For using the PS/2 interfaces in interrupt mode, interrupts have to be enabled in the FPGA address range `0xAA00 0018` to `0xAA00 0020` and in the VR4133. The interrupt generates a signal on GPIO (0) accordingly. For details to handle interrupts on VR4133 side refer to the VR4133 Users Manual.

Figure 4-7: PS/2 connector CN2-1

1	MOUSE_DAT
2	
3	GND
4	VDD5.0
5	MOUSE_CLK
6	
10	GND
11	GND
12	GND
13	GND



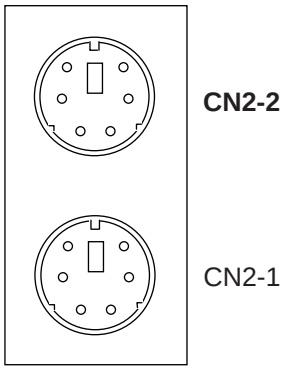
4.4.6 Usage of PS/2 Keyboard interface (CN2-2)

The PS/2 Keyboard interface is realized by the FPGA. The CPU controls this interface by reading or writing from/to FPGA registers. The implementation of the PS/2 interfaces supports polling and interrupt mode.

The address map for the PS/2 interfaces is `0xAA00 0054` to `0xAA00 0064`. For details refer to section 5.1 ”FPGA Register Set / Memory Map” on page 53. The PS/2 Ready register contains the status of both PS/2 interfaces, Mouse and Keyboard. For using the PS/2 interfaces in interrupt mode, interrupts have to be enabled in the FPGA address range `0xAA00 0018` to `0xAA00 0020` and in the VR4133. The interrupt generates a signal on GPIO (0) accordingly. For details to handle interrupts on VR4133 side refer to the VR4133 Users Manual.

Figure 4-8: PS/2 connector CN2-2

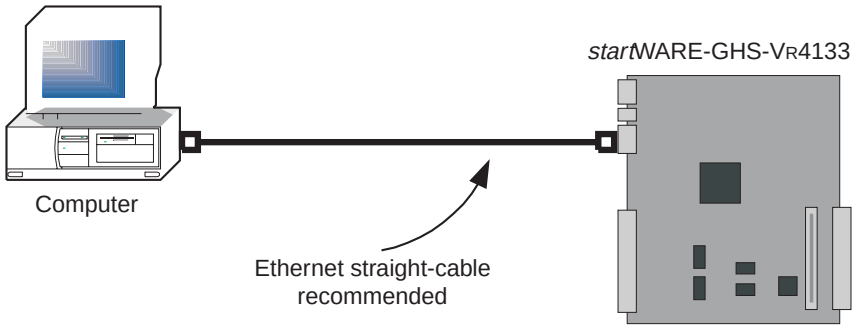
1	KEYB_DAT
2	
3	GND
4	VDD5.0
5	KEYB _CLK
6	
10	GND
11	GND
12	GND
13	GND



4.4.7 Usage of Ethernet interfaces (CN1)

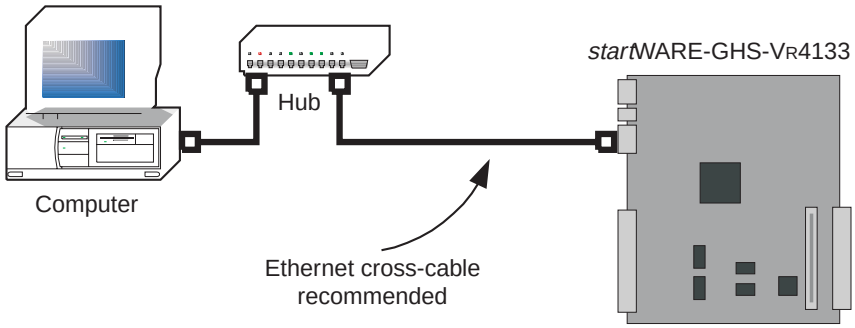
Caution: The transmit and receive signals have already been crossed on the PCB of the *start*-WARE-GHS-VR4133!
Therefore you may use a normal straight Ethernet cable to connect the board directly to a host computer. You don't need to use a cross cable for this purpose as you usually would have to.

Figure 4-9: SW4133 Ether connect direct



Caution: If you use a modern switch or hub this device should automatically adjust the right correspondence between transmit or receive signal. In that case you may also use a normal straight Ethernet patch cable. However if you experience problems using a simple hub device please use a cross-cable instead of a straight cable.

Figure 4-10: SW4133 Ether connect hub



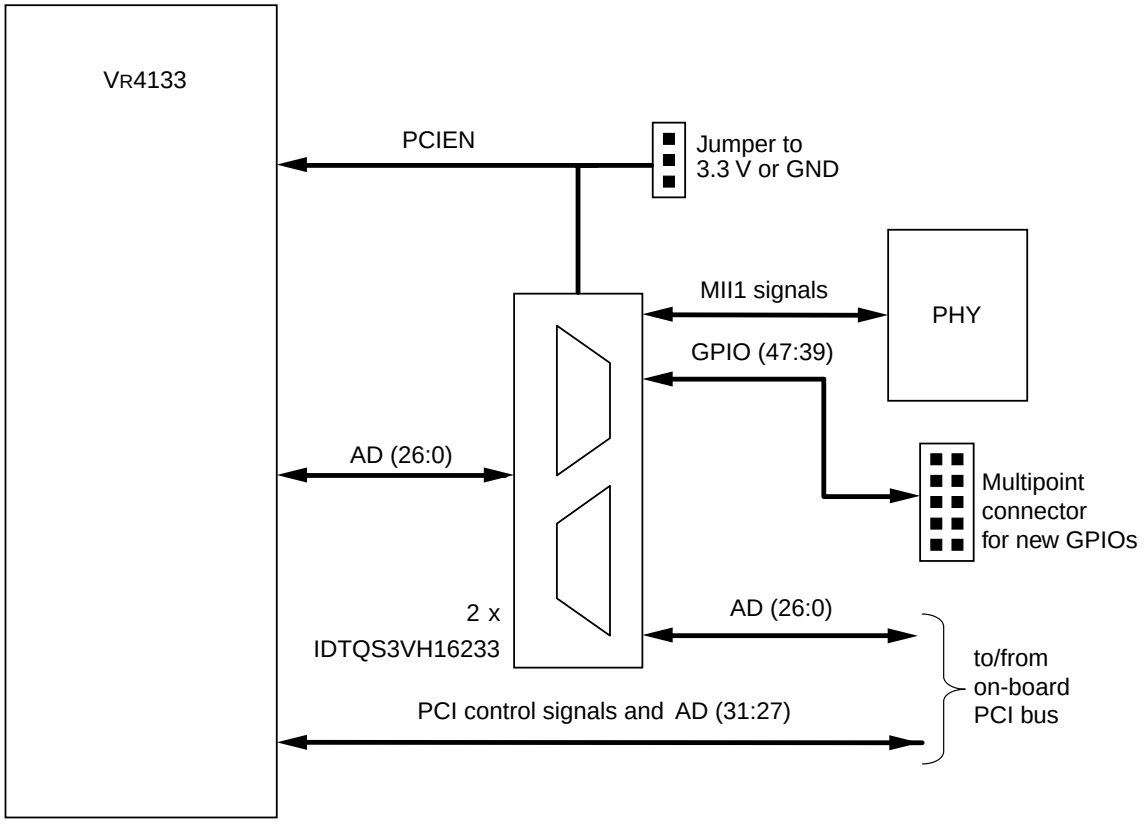
The VR4133 processor supports four different operation modes with different interface configurations:

Table 4-21: Usage of Ethernet interfaces (CN1)

VR4133 mode	Interface configuration
A	PCI + MII0
B	PCI + RMII0 + RMII1
C	MII0 + MII1
D	PCI (VR4133 “compatible” mode)

The *start*WARE-GHS-VR4133 supports all four modes. As the interfaces that are affected by the mode switching partially share pins, it is inevitable to add additional switching logic between the VR4133 and the connected resp. connectable PCI devices, if more than one mode is supported. To keep the switching logic as simple as possible the configuration is shown in the figure below:

Figure 4-11: Configuration switch for PCI/MII1 selection



To keep the switching logic simple an extra PHY (L80223 device from LSI Logic) exclusively for the MII1 interface is used for this purpose. As switching element two 16-bit bi-directional multiplexers from IDT are used. Selection between the different configuration will be done with a jumper that controls the level of the PCIEN pin of the VR4133 as well as the multiplexers (the polarity for the multiplexers is changed by the FPGA). Propagation delay of the multiplexers is app. 200ps.

Selecting the configuration of the RMII0, MII0 and RMII1 interfaces is done as shown below. The DM9161 from Davicom is used as PHY, because it supports MII and RMII interfaces. The RMII0, MII0 and RMII1 interfaces of VR4133 are mainly shared with GPIO pins. Most of these GPIOs do not have a dedicated function on startWARE-GHS-VR4133 and thus they will not be made usable on *start*WARE-GHS-VR4133 (except GPIO (10:8)). As switching devices the IDTQS3VH16233 are used as well.

Figure 4-12: Configuration switch for RMII0, MII0 and RMII1 configuration

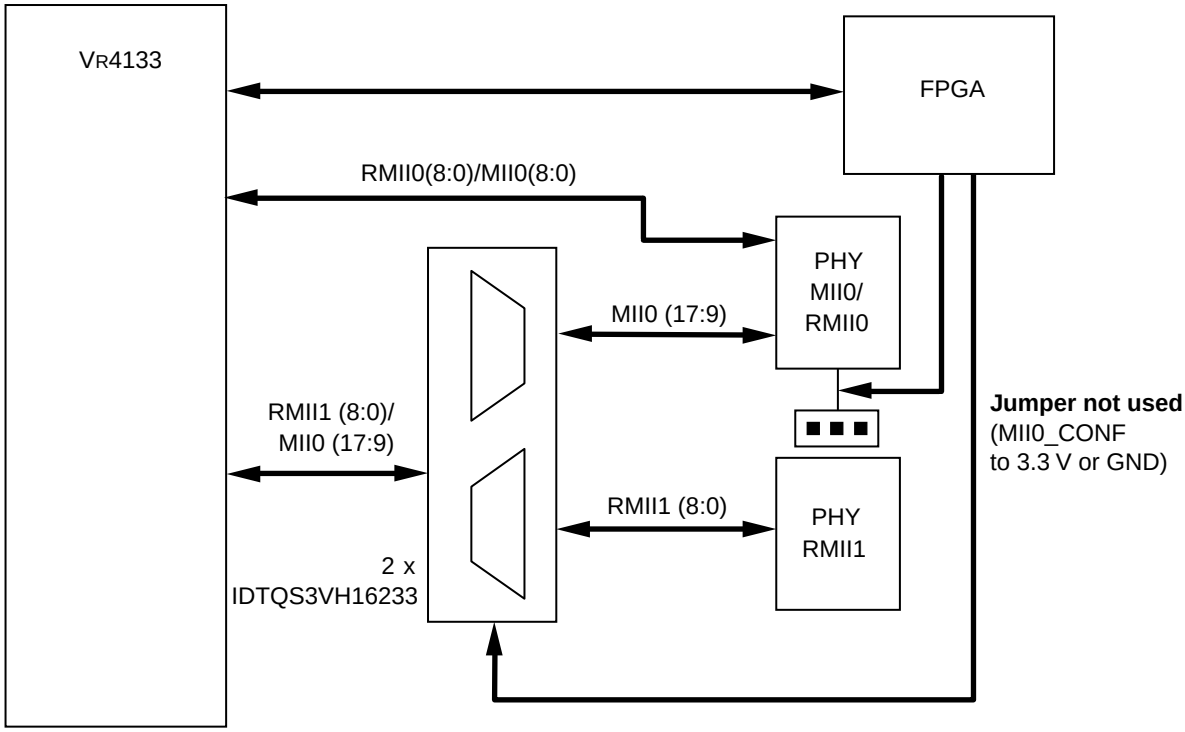
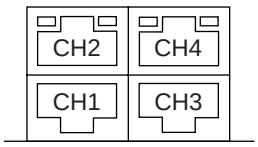


Figure 4-13: Ethernet connectors CN1

Connector	PHY	VR4133 interface	Usable in Mode
CH1	LSI_Logic L80223	MII1	C
CH2	Davicom DM9161	MII0/RMII0	A, B, C
CH3	Davicom DM9161	RMII1	B
CH4		Realtek RTL8139C	A, B, D



Note: CH4 only available, if PCI interface is usable (Mode A, B and D).

As described, the *start*WARE-GHS-VR4133 has up to three on-board Ethernet controller, for which three MAC addresses per board have been reserved in order to avoid network collisions. The MAC addresses are generally given as:

00:00:4C:80:5A:mm
00:00:4C:80:5A:mm+1
00:00:4C:80:5A:mm+2

“nnnn” of the NEC serial number defines “mm” of the MAC addresses.

Please note that the MAC address is counted up as a hexadecimal number while the NEC serial number is counted up in decimal. Here are three examples:

NEC serial number 0001	first MAC address	00:00:4C:80:5A:00
NEC serial number 0002	first MAC address	00:00:4C:80:5A:03
NEC serial number 0017	first MAC address	00:00:4C:80:5A:30

The MAC addresses have been programmed into the on-board Flash memory and the first MAC address is stored in the EEPROM for using the Ethernet connection realized by the Realtec Ethernet Controller. If required, the MAC addresses can be updated by using the Etherinit Software, which is available on CDROM.

Caution: Ethernet provides a reliable MAC address initialisation only, if the Flash memory is erased. Changing a valid MAC address requires a *complete* new initialisation of the evaluation board!

The MAC addresses are stored into the on-board Flash at address:

0xBFC0A350	ID for the RTL8139 controller
0xBFC0A360	ID for the VR4133 internal MACs
0xBFC0A370	ID for the VR4133 internal MACs

For security the MAC address can also be found on an additional sticker on the board.

4.4.8 Using the IrDA Interface

For using the IrDA interface the N-Wire ICE (RTE-1000-TP-EE) must be disconnected from CN1 and jumper JP23 (JTAGEN) must connect Pin 1 and 2 of the jumper field (that means disable JTAG/N-Wire).

4.4.9 Using the CSI Interface (CN15)

The CSI interface of the VR4133 is available on connector CN15. This interface is extended by four signals controlled by the FPGA. Additionally CN15 provides 3.3 V power supplies. Signals on the connector are assigned as follows:

Figure 4-14: CSI connector pin assignment (CN15)

3.3 V	1			2	SIN
SOUT	3			4	3.3 V
RQC0	5			6	RQC1
GNTC0	7			8	GNCT1
SECLK	9			10	GND

SIN, SOUT and SECLK are connected via permanently enabled, non-inverting buffers to the corresponding pins of the VR4133; the other signals are provided to support multi-master operation of the CSI. RQC0 and RQC1 are (input) signals to request the CPU's CSI interface. The CPU can confirm this request via the GNTC0 and GNTC1 (output) signals. These lines are fully software controlled by the CPU by reading and writing the CSICONTREG in the FPGA. The usage of these extra signals is not mandatory.

4.4.10 General Purpose IOs at CN11

There are several possibilities to connect user hardware to the board; one of them is using I/O ports. The VR4133 evaluation board provides two kinds of I/O ports: **G**PIO's are realized in the CPU (VR4133) and controlled using the respective GIU control registers in the VR4133. These GPIO's are not available in all operating modes of the VR4133 – please refer to VR4133 User's Manual.

At the multipoint connector CN11 the pin assignment is as follows:

Figure 4-15: GPIO pin assignment at CN11

	1			2	VDD 3.3 V
	3			4	
GPO39	5			6	
GPO40	7			8	
GPO46	9			10	
GPO45	11			12	GPIO3
GPO44	13			14	GPIO2
GPO43	15			16	GPIO1
GPO42	17			18	GPIO0
GPO41	19			20	GND

4.4.11 General Purpose IO's at CN12

Another kind of I/O ports is realized in the FPGA. These FPGA_I_O's are connected to the FPGA. The CPU controls these I/O's and each I/O can be defined as input or output separately. Control is done via a set of two registers in the FPGA: FPGA_I_O_MODEREG and FPGA_I_O_REG.

Alternatively these signal can be used by the I^2S blocks implemented in the FPGA. Please refer to the I^2S section for details.

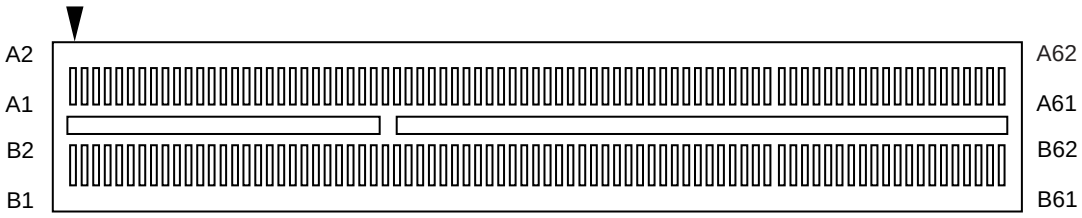
Figure 4-16: Pin assignment at CN12 (FPGA_I_O usage)

	1			2	VDD 3.3 V
GND	3			4	FPGA_I_O(13)
FPGA_I_O(12)	5			6	FPGA_I_O(11)
FPGA_I_O(10)	7			8	GND
FPGA_I_O(9)	9			10	FPGA_I_O(8)
FPGA_I_O(7)	11			12	FPGA_I_O(6)
GND	13			14	FPGA_I_O(5)
FPGA_I_O(4)	15			16	FPGA_I_O(3)
FPGA_I_O(2)	17			18	FPGA_I_O(1)
FPGA_I_O(0)	19			20	GND

4.4.12 PCI Connector CN10

The PCI interface of the VR4133 processor is routed to a standard PCI connector CN10. The VR4133 supports up to three external PCI devices; a device connected to CN10 is using REQ1_N and GNT1_N signals of the CPU. The PCI interrupt pins INTA_N, … INTD_N are routed to and processed in the FPGA. Modification of the FPGA content will allow implementation of a simple PCI interrupt controller; however this function is currently not implemented.

Figure 4-17: PCI connector pin assignment



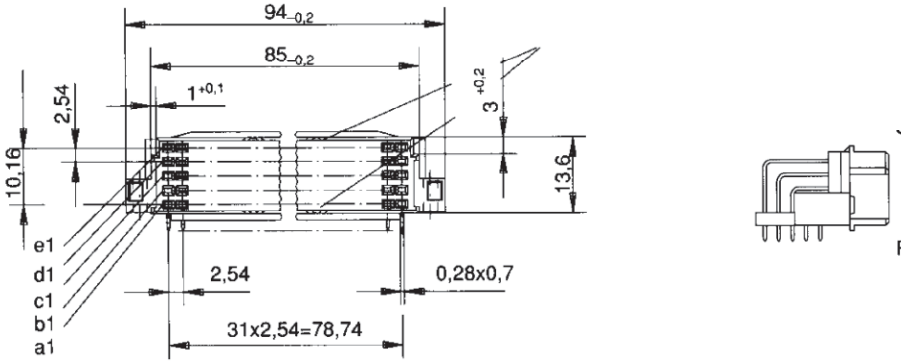
	A1	B1	
	A2	B2	
	A3	B3	GND
	A4	B4	
VDD 5.0	A5	B5	VDD 5.0
INTA1_N	A6	B6	VDD 5.0
INTC1_N	A7	B7	INTB1_N
VDD 5.0	A8	B8	INTD1_N
	A9	B9	
VDD 3.3	A10	B10	
	A11	B11	
	A12	B12	
	A13	B13	
	A14	B14	
RST_N	A15	B15	GND
VDD 3.3	A16	B16	PCLK1
GNT1_N	A17	B17	GND
GND	A18	B18	REQ1_N
	A19	B19	VDD 3.3
AD30	A20	B20	AD31
VDD 3.3	A21	B21	AD29
AD28	A22	B22	GND
AD26	A23	B23	AD27
GND	A24	B24	AD25
AD24	A25	B25	VDD 3.3
IDSEL1	A26	B26	CBE3
VDD 3.3	A27	B27	AD23
AD22	A28	B28	GND
AD20	A29	B29	AD21
GND	A30	B30	AD19
AD18	A31	B31	VDD 3.3
AD16	A32	B32	AD17
VDD3.3	A33	B33	CBE2
FRAME_N	A34	B34	GND
GND	A35	B35	IRDY_N
TRDY_N	A36	B36	VDD 3.3
GND	A37	B37	DEVSEL_N
STOP_N	A38	B38	GND
VDD 3.3	A39	B39	LOCK_N
	A40	B40	PERR_N

	A41	B41	VDD 3.3
GND	A42	B42	SERR_N
PAR	A43	B43	VDD 3.3
AD15	A44	B44	CBE1
VDD 3.3	A45	B45	AD14
AD13	A46	B46	GND
AD11	A47	B47	AD12
GND	A48	B48	AD10
AD9	A49	B49	
GND	A50	B50	GND
GND	A51	B51	GND
CBE0	A52	B52	AD8
VDD 3.3	A53	B53	AD7
AD6	A54	B54	VDD 3.3
AD4	A55	B55	AD5
GND	A56	B56	AD3
AD2	A57	B57	GND
AD0	A58	B58	AD1
VDD 3.3	A59	B59	VDD 3.3
	A60	B60	
VDD 5.0	A61	B61	VDD 5.0
VDD 5.0	A62	B62	VDD 5.0

4.4.13 I/O Board Connector CN9

The VR4133 evaluation board offers two “specialised” extension connectors; one is referred to as I/O Board connector. It is a 160-pin (5 × 32) VME-style connector that carries the complete PCI bus (with REQ2_N and GNT2_N) and several additional signals like GPIOs from the CPU, FPIOs and FPGA_I_O‘s from the FPGA and power control signals of the VR4133.

Figure 4-18: I/O board connector pin assignment

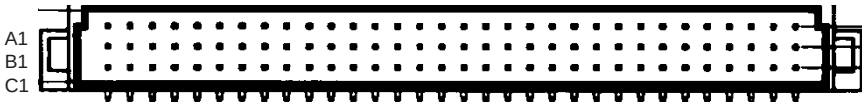


A1		B1		C1		D1		E1	
A2		B2		C2	GND	D2		E2	GND
A3	VDD 5.0	B3	INTA2-N	C3	VDD 5.0	D3	VDD 5.0	E3	INTE-N
A4	INTC2-N	B4	VDD 5.0	C4	INTB2-N	D4	INTD2-N	E4	GPIO0
A5		B5	VDD 3.3	C5		D5		E5	GPIO1
A6		B6		C6		D6		E6	GPIO2
A7		B7		C7		D7		E7	GPIO3
A8	RST-N	B8	VDD 3.3	C8	GND	D8	PCLK	E8	GND
A9	GNT2-N	B9	GND	C9	GND	D9	REQ2-N	E9	GND
A10		B10	AD30	C10	VDD 3.3	D10	AD31	E10	FPIO0
A11	VDD 3.3	B11	AD28	C11	AD29	D11	GND	E11	FPIO1
A12	AD26	B12	GND	C12	AD27	D12	AD25	E12	FPIO2
A13	AD24	B13	IDSEL2	C13	VDD 3.3	D13	CBE3	E13	FPIO3
A14	VDD 3.3	B14	AD22	C14	AD23	D14	GND	E14	GND
A15	AD20	B15	GND	C15	AD21	D15	AD19	E15	FPGA_I_O (0)
A16	AD18	B16	AD16	C16	VDD 3.3	D16	AD17	E16	FPGA_I_O (1)
A17	VDD 3.3	B17	FRAME-N	C17	CBE2	D17	GND	E17	FPGA_I_O (4)
A18	GND	B18	TRDY-N	C18	IRDY-N	D18	VDD 3.3	E18	FPGA_I_O (5)
A19	GND	B19	STOP-N	C19	DEVSEL-N	D19	GND	E19	GND
A20	VDD 3.3	B20		C20	LOCK-N	D20	PERR-N	E20	MPOWER
A21		B21	GND	C21	VDD 3.3	D21	SERR-N	E21	SPOWER
A22	PAR	B22	AD15	C22	VDD 3.3	D22	CBE1	E22	POWERON
A23	VDD 3.3	B23	AD13	C23	AD14	D23	GND	E23	FPGA_I_O (6)
A24	AD11	B24	GND	C24	AD12	D24	AD10	E24	FPGA_I_O (7)
A25	AD9	B25	GND	C25		D25	GND	E25	FPGA_I_O (8)
A26	GND	B26	CBE0	C26	GND	D26	AD8	E26	FPGA_I_O (9)
A27	VDD 3.3	B27	AD6	C27	AD7	D27	VDD 3.3	E27	FPGA_I_O (10)
A28	AD4	B28	GND	C28	AD5	D28	AD3	E28	FPGA_I_O (11)
A29	AD2	B29	AD0	C29	GND	D29	AD1	E29	FPGA_I_O (12)
A30	VDD3.3	B30		C30	VDD3.3	D30		E30	FPGA_I_O (13)
A31	VDD5.0	B31	VDD5.0	C31	VDD5.0	D31	VDD5.0	E31	FPGA_I_O (14)
A32	DDOUT	B32	DDIN	C32	FPGA_I_O (2)	D32	FPGA_I_O (3)	E32	FPGA_I_O (15)

4.4.14 Ravin Board Connector CN8

The other “specialised” connector CN8 uses a 96-pin (3 × 32) VME-style connector, that carries the CPU’s buffered address and data bus and I/O control signals. One CPU GPIO, two DMA control signals and eight FPGA_I_O signals are routed to the connector as well. The intention of CN8 is to connect to a board with the RAVIN-E graphic display controller (µPD72255), that is offered as *start*WARE-GHS-RAVINE.

Figure 4-19: Ravin board connector pin assignment



A1	VDD 5.0	B1	VDD 5.0	C1	VDD 5.0
A2	GND	B2	BD31	C2	BD30
A3	BD29	B3	BD28	C3	BD27
A4	BD26	B4	BD25	C4	BD24
A5	GND	B5	BD23	C5	BD22
A6	BD21	B6	BD20	C6	BD19
A7	BD18	B7	BD17	C7	BD16
A8	GND	B8	BD15	C8	BD14
A9	BD13	B9	BD12	C9	BD11
A10	BD10	B10	BD9	C10	BD8
A11	GND	B11	BD7	C11	BD6
A12	BD5	B12	BD4	C12	BD3
A13	BD2	B13	BD1	C13	BD0
A14	GND	B14	BEB0	C14	BEB1
A15	BEB2	B15	BEB3	C15	GND
A16	RESET	B16	FPGA_I_O (7)	C16	FPGA_I_O (6)
A17	GND	B17	BA23	C17	BA22
A18	BA21	B18	BA20	C18	BA19
A19	BA18	B19	BA17	C19	BA16
A20	GND	B20	BA15	C20	BA14
A21	BA13	B21	BA12	C21	BA11
A22	BA10	B22	BA9	C22	BA8
A23	GND	B23	BA7	C23	BA6
A24	BA5	B24	BA4	C24	BA3
A25	BA2	B25	RDY	C25	FPGA_I_O (0)
A26	GND	B26	BIOCS0_N	C26	BIOCS1_N
A27	GND	B27	BWR_N	C27	BRD_N
A28	BROMCS0	B28	BROMCS1	C28	FPGA_I_O (1)
A29	GPIO1	B29	HLDRQ_N/DRQ0	C29	HLDAK_N/DAK0
A30	FPGA_I_O (2)	B30	VDD 3.3	C30	FPGA_I_O (3)
A31	GND	B31	GND	C31	GND
A32	FPGA_I_O (4)	B32	GND	C32	FPGA_I_O (5)

[MEMO]

Chapter 5 FPGA Function and Software Description

5.1 FPGA Register Set / Memory Map

The FPGA has several registers to control extra I/O ports, a LED display DIP switches and several other functionalities like I^2S- and PS/2 interfaces. This chapter explains the register programming functions register by register. The processor should access these registers with uncached load/store operations at the addresses shown in the table below:

Table 5-1: FPGA register set

Address	R/W	Register Symbol	Function
0x0A00 0000	R	REVREG	FPGA Revision
0x0A00 0004	R/W	MODEREG	Modes for internal functions
0x0A00 0008	R/W	LEDREG	LED control register
0x0A00 000C	R	DIPSWITCHREG	DIP switch register
0x0A00 0010	R/W	FLASHACCREG	Flash access register
0x0A00 0014	R/W	CSICONTREG	CSI control register for additional control signals
0x0A00 0018	R/W	FPGAINTREG	Interrupt cause from FPGA
0x0A00 001C	R/W	FPGAINTENREG	Interrupt enable from FPGA
0x0A00 0020	R/W	FPGAINTPOLREG	Interrupt polarity from FPGA
0x0A00 0024	R/W	FPGAINTSELREG	Select interrupt source in FPGAINTREG
0x0A00 0050	R/W	ETHER_CONTREG	Ethernet control signals
0x0A00 0054	R	PS2_READYREG	Mouse and Keyboard ready
0x0A00 005c	R	PS2_MOUSE_XDATAREG_RIGHT_BUTTON	X Increments of Mouse movement / right button
0x0A00 0060	R	PS2_MOUSE_YDATAREG_LEFT_BUTTON	Y Increments of Mouse movement / left button
0x0A00 0064	R	PS2_KEYBSCANREG	Keyboard scan code
0x0A00 0068	R/W	FPIO_MODEREG	FPIO input/output setting register
0x0A00 006C	R/W	FPIO_DATAREG	FPIO input/output data register
0x0A00 0070	R/W	FPGA_I_O_MODEREG	FPGA_I_O input/output setting register
0x0A00 0074	R/W	FPGA_I_O_DATAREG	FPGA_I_O input/output data register
0x0A00 0080	R/W	I2S_CONTROLREG	I2S pin select general control register
0x0A00 0090	R/W	I2S0_CONFREG	I2S0 configuration register
0x0A00 0094	R/W	I2S0_RX_STATREG	I2S0 reception status register
0x0A00 0098	R/W	I2S0_TX_STATREG	I2S0 transmission status register
0x0A00 00A0	R/W	I2S1_CONFREG	I2S1 configuration register
0x0A00 00A4	R/W	I2S1_RX_STATREG	I2S1 reception status register
0x0A00 00A8	R/W	I2S1_TX_STATREG	I2S1 transmission status register

5.2 Description of Basic FPGA Functions / Registers

5.2.1 REVREG (0x0A00 0000)

Bit	15	14	13	12	11	10	9	8
Name	BOARD _TYPE (3)	BOARD _TYPE (2)	BOARD _TYPE (1)	BOARD _TYPE (0)	FPGA_REV (3)	FPGA_REV (2)	FPGA_REV (1)	FPGA_REV (0)
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Name	FPGA _SREV(3)	FPGA _SREV (2)	FPGA _SREV (1)	FPGA _SREV (0)	BOARD _REV (3)	BOARD _REV (2)	BOARD _REV (1)	BOARD _REV (0)
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

This register allows identifying the Board type and several revision numbers, which might be relevant for the implemented features of the board.

Bit	Name	Function
15:12	BOARD_TYPE (3:0)	BOARD_TYPE (3:0) = 0000: *start*WARE-GHS-VR4133 BOARD_TYPE (3:0) = 0001: *start*WARE-GHS-VR4133 others: reserved
11:8	FPGA_REV (3:0)	**Main Revision of the FPGA** FPGA_REV (3:0) is decoded as following: FPGA_REV (3:0) = 0001: Rev.: 1.x FPGA_REV (3:0) = 0010: Rev.: 2.x FPGA_REV (3:0) = 0011: Rev.: 3.x others: reserved
7:4	FPGA_SREV (3:0)	**Sub Revision of the FPGA** FPGA_SREV (3:0) is decoded as following: FPGA_SREV (3:0) = 0000: Rev.: y.0 others: reserved
3:0	BOARD_REV (3:0)	**Revision of the Board (*start*WARE-GHS-VR4133)** BOARD_REV (3:0) is decoded as following: BOARD_REV (3:0) = 0000: Rev.: 1.0 BOARD_REV (3:0) = 0001: Rev.: 2.0 BOARD_REV (3:0) = 0010: Rev.: 2.1 BOARD_REV (3:0) = 0011: Rev.: 2.2 BOARD_REV (3:0) = 0100: Reserved …… BOARD_REV (3:0) = 1111: Reserved

5.2.2 MODEREG (0x0A00 0004)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTMODE	Reserved	Reserved	POWER-MODE	Reserved	Reserved	Reserved	LEDMODE
R/W	R/W	R	R	R/W	R	R	R	R/W
Reset	0	0	0	0	0	0	0	0

This register determines how information in the LED register is used to control the 7-segment display on the board. The user can select between a BCD decoder and a bit-wise assignment of LED segments.
Furthermore the VR4133 can control by writing to this register, whether the SDRAM is switched off when the processor goes to hibernate mode, or not.

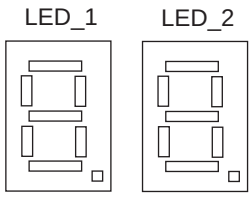
Bit	Name	Function	
15:8	Reserved		
7	INTMODE	INTMODE = 0:	The interrupt output of the FPGA is connected to the GPIO0 and each interrupt of the FPGA will be reported to the VR4133.
		INTMODE = 1:	The interrupt output is disabled. For using the GPIO0 as a standard GPIO pin, this bit allows to define an interrupt disable mode. The output pin of the FPGA is switched to high impedance and no interrupt of the FPGA will be reported to the VR4133.
6:5	Reserved		
4	POWERMODE	POWERMODE = 0:	MPOWER controls SDRAM power supply (SDRAM is switched off, when VR4133 goes to hibernate mode)
		POWERMODE = 1:	SPOWER controls SDRAM power supply (SDRAM is **not** switched off, when VR4133 goes to hibernate mode)
3:1	Reserved		
0	LEDMODE	LEDMODE = 0:	LED (7:0) will be decoded (binary to 7 segment)
		LEDMODE = 1:	LED (15:8) control Segments of LED – left side LED (7:0) control Segments of LED – right side

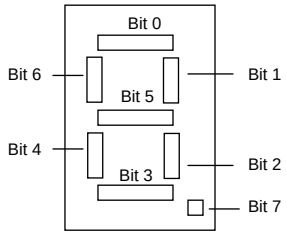
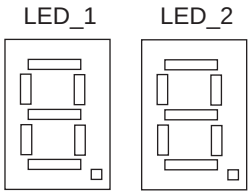
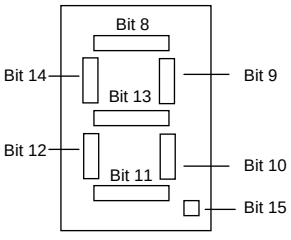
5.2.3 LEDREG (0x0A00 0008)

Bit	15	14	13	12	11	10	9	8
Name	LED15	LED 14	LED 13	LED 12	LED 11	LED 10	LED 9	LED 8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	LED 7	LED 6	LED 5	LED 4	LED 3	LED 2	LED 1	LED 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

The information in the LEDREG is used to control the 7-segment display on the board. The user can select between a BCD decoder and a bit-wise assignment of LED segments. The mode can be defined in the MODEREG. If the DIP-switch SW1-8 is switched ON the test mode of the 7-segment display is selected and the contents of the LEDREG will not be reflected in the 7-segment display. The register contents will not be changed during the test mode. After switching OFF DIP-switch SW1-8, the value will be displayed according to the selected LEDMODE.

Bit	Name	Function
(7:0)	LED (7:0)	 If LEDMODE = 0, LED (7:4) will be decoded as following: LED (7:4)= 0000: LED_1(7:0) = 1010_0000 0 LED (7:4)= 0001: LED_1(7:0) = 1111_1001 1 LED (7:4)= 0010: LED_1(7:0) = 1100_0100 2 LED (7:4)= 0011: LED_1(7:0) = 1101_0000 3 LED (7:4)= 0100: LED_1(7:0) = 1001_1001 4 LED (7:4)= 0101: LED_1(7:0) = 1001_0010 5 LED (7:4)= 0110: LED_1(7:0) = 1000_0010 6 LED (7:4)= 0111: LED_1(7:0) = 1111_1000 7 LED (7:4)= 1000: LED_1(7:0) = 1000_0000 8 LED (7:4)= 1001: LED_1(7:0) = 1001_0000 9 LED (7:4)= 1010: LED_1(7:0) = 1000_1000 A LED (7:4)= 1011: LED_1(7:0) = 1000_0011 B LED (7:4)= 1100: LED_1(7:0) = 1010_0110 C LED (7:4)= 1101: LED_1(7:0) = 1100_0001 D LED (7:4)= 1110: LED_1(7:0) = 1000_0110 E LED (7:4)= 1111: LED_1(7:0) = 1000_1110 F

Bit	Name	Function
(7:0)	LED (7:0)	If LEDMODE = 0, LED (3:0) will be decoded as following: LED (3:0)= 0000: LED_2(7:0) = 1010_0000 0 LED (3:0)= 0001: LED_2(7:0) = 1111_1001 1 LED (3:0)= 0010: LED_2(7:0) = 1100_0100 2 LED (3:0)= 0011: LED_2(7:0) = 1101_0000 3 LED (3:0)= 0100: LED_2(7:0) = 1001_1001 4 LED (3:0)= 0101: LED_2(7:0) = 1001_0010 5 LED (3:0)= 0110: LED_2(7:0) = 1000_0010 6 LED (3:0)= 0111: LED_2(7:0) = 1111_1000 7 LED (3:0)= 1000: LED_2(7:0) = 1000_0000 8 LED (3:0)= 1001: LED_2(7:0) = 1001_0000 9 LED (3:0)= 1010: LED_2(7:0) = 1000_1000 A LED (3:0)= 1011: LED_2(7:0) = 1000_0011 B LED (3:0)= 1100: LED_2(7:0) = 1010_0110 C LED (3:0)= 1101: LED_2(7:0) = 1100_0001 D LED (3:0)= 1110: LED_2(7:0) = 1000_0110 E LED (3:0)= 1111: LED_2(7:0) = 1000_1110 F If LEDMODE = 1, LED (7:0) controls LED_2 directly. 
(15:8)	LED (15:8)	 If LEDMODE = 0, LED (15:8) are not used. The written data can be read, but will be ignored for LED control. If LEDMODE = 1, LED (15:8) controls LED1 directly 

5.2.4 DIPSWITCHREG (0x0A00 000C)

Bit	15	14	13	12	11	10	9	8
Name	SW1-1	SW1-2	SW1-3	SW1-4	SW1-5	SW1-6	SW1-7	SW1-8
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	SW2-1	SW2-2	SW2-3	SW2-4	SW2-5	SW2-6	SW2-7	SW2-8
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

The DIPSWITCHREG register is a read only register that reflects the setting of the switches SW1 and SW2. Please see section 4.3 ”Switch Settings” on page 35 for the meaning of the different switch settings. The register is updated according to current switch positions any time, when it is read by the processor; this ensures that the processor “sees” the current switch position.

5.2.5 FLASHACCREG (0x0A00 0010)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	FLASH_ RDY4	FLASH_ RDY3	FLASH_ RDY2	FLASH_ RDY1
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

This register permits the processor to check the programming status of the Flash devices. FLASH_RDY1 to FLASH_RDY4 reflects the status of the RY/BY# Pin of each Flash device.

Bit	Name	Function
15:4	Reserved	
3	FLASH_ RDY4	RY/BY# signal of IC38 (not assembled) 1: High level 0: Low level
2	FLASH_ RDY3	RY/BY# signal of IC37(not assembled) 1: High level 0: Low level
1	FLASH_ RDY2	RY/BY# signal of IC21 1: High level 0: Low level
0	FLASH_ RDY1	RY/BY# signal of IC13 1: High level 0: Low level

5.2.6 CSICONTREG (0x0A00 0014)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	RQC1	RQC0	GNTC1	GNTC0
R/W	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

The CSICONTREG provides a software handshake for multi-master configurations on the CSI interface of the VR4133. This register reads the RQC (1:0) pins and writes the GNTC (1:0) pins. The correspondence is 1 bit per pin.
Writing a value to the RQC (1:0) bit does not affect the RQC(1:0) pin (the write data is ignored). When reading the CSICONTREG, the RQC (1:0) pin's state and the written GNTC (1:0) bits will be read.

Bit	Name	Function
15:4	Reserved	
3	RQC1	RQC1 pin input data 1: High level 0: Low level
2	RQC0	RQC0 pin input data 1: High level 0: Low level
1	GNTC1	GNTC1 pin output data 1: High level 0: Low level
0	GNTC0	GNTC0 pin output data 1: High level 0: Low level

5.2.7 FPGAINTREG (0x0A00 0018)

Bit	15	14	13	12	11	10	9	8
Name	INTA1_N	INTB1_N	INTC1_N	INTD1_N	INTA2_N / SxOVR	INTB2_N / S1FUL	INTC2_N / S0FUL	INTD2_N / S1EMPTY
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTE_N / S0EMPTY	FPIO4	RQC0	RQC1	PS2 _KEYB	PS2 _MOUSE	ETH0	ETH1
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

This register reflects the current value of the interrupt signals of the PCI slot, the I/O Board, the on-board Ethernet controller, the PS/2 keyboard interface, the PS/2 Mouse interface and the Request signals of the CSI extension. The interrupt-input signals are forwarded to the CPU via GPIO0.

Bit	Name	Function
15	INTA1_N	Interrupt INTA1_N from PCI slot
14	INTB1_N	Interrupt INTB1_N from PCI slot
13	INTC1_N	Interrupt INTC1_N from PCI slot
12	INTD1_N	Interrupt INTD1_N from PCI slot
11	INTA2_N / SxOVR	Interrupt INTA2_N from I/O-Board or I2S Slave 1/0 overrun**Note**
10	INTB2_N / S1FULL	Interrupt INTB2_N from I/O-Board or I2S 1 RX FULL**Note**
9	INTC2_N /S0FULL	Interrupt INTC2_N from I/O-Board or I2S 0 RX FULL**Note**
8	INTD2_N / S1EMPTY	Interrupt INTD2_N from I/O-Board or I2S 1 TX EMPTY**Note**
7	INTE_N / S0EMTY	Interrupt INTE_N from I/O-Board or I2S 0 TXEMPTY**Note**
6	FPIO4	Interrupt signal (FPIO4) from Ethernet Controller RTL8139C
5	RQC0	Request RQC0 from CSI interface
4	RQC1	Request RQC1 from CSI interface
3	PS2_KEYB	Interrupt PS2_KEYB from PS/2 Keyboard
2	PS2_MOUSE	Interrupt PS2_MOUSE from PS/2 Mouse
1	ETH0	Link status interrupt from IC3 (ETH0 in RMII and MII mode)
0	ETH1	Link status interrupt from IC2 (ETH1 in RMII mode)

Note: Selection of interrupt source is done in FPGAINTSELREG

5.2.8 FPGAINTENREG (0x0A00 001C)

Bit	15	14	13	12	11	10	9	8
Name	INTA1_EN	INTB1_EN	INTC1_EN	INTD1_EN	INTA2_EN	INTB2_EN	INTC2_EN	INTD2_EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
Name	INTE_EN	FPIO4_EN	RQC0_EN	RQC1_EN	PS2_KEYB_EN	PS2_MOUSE_EN	ETH0_EN	ETH1_EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	0	0	0	0	0	0

This register allows to enable/disable the interrupt signals coming from the PCI slot, the I/O Board and the on-board Ethernet controller. The interrupt-input signals are forwarded to the CPU via GPIO0. Additional the RQC0 and RQC1 signal can be used as an interrupt request source. Within the FPGA two PS/2 interfaces are implemented. Both interfaces can be used in interrupt mode. All these signals can be enabled/disabled and are also forwarded to the GPIO0 interrupt. This interrupt output is always high active.

When this bit is set to 1, the corresponding pin/function is enabled. The default value is selected to secure the same functionality as *start*WARE-GHS-VR4133. This register allows disabling unused interrupt signals.

Bit	Name	Function	
15	INTA1_EN	Interrupt enable for INTA1_N from PCI slot	1: interrupt enabled 0: interrupt disabled
14	INTB1_EN	Interrupt enable for INTB1_N from PCI slot	1: interrupt enabled 0: interrupt disabled
13	INTC1_EN	Interrupt enable for INTC1_N from PCI slot	1: interrupt enabled 0: interrupt disabled
12	INTD1_EN	Interrupt enable for INTD1_N from PCI slot	1: interrupt enabled 0: interrupt disabled
11	INTA2_EN**Note**	Interrupt enable for INTA2_N from I/O-Board	1: interrupt enabled 0: interrupt disabled
10	INTB2_EN**Note**	Interrupt enable for INTB2_N from I/O-Board	1: interrupt enabled 0: interrupt disabled
9	INTC2_EN**Note**	Interrupt enable for INTC2_N from I/O-Board	1: interrupt enabled 0: interrupt disabled
8	INTD2_EN**Note**	Interrupt enable for INTD2_N from I/O-Board	1: interrupt enabled 0: interrupt disabled
7	INTE_EN**Note**	Interrupt enable for INTE_N from I/O-Board	1: interrupt enabled 0: interrupt disabled
6	FPIO4_EN	Interrupt enable for FPIO4 signal from Ethernet Controller RTL8139C	1: interrupt enabled 0: interrupt disabled
5	RQC0_EN	Enable interrupts on RQC0 signal from CSI interface	1: interrupt enabled 0: interrupt disabled
4	RQC1_EN	Enable interrupts on RQC1 signal from CSI interface	1: interrupt enabled 0: interrupt disabled
3	PS2_KEYB _EN	Enable PS2_KEYB interrupts from PS/2 Keyboard	1: interrupt enabled 0: interrupt disabled
2	PS2_MOUSE _EN	Enable PS2_MOUSE interrupts from PS/2 Mouse	1: interrupt enabled 0: interrupt disabled
1	ETH0_EN	Enable link status interrupt from IC3 (ETH0 in RMII and RMI mode)	1: interrupt enabled 0: interrupt disabled
0	ETH1_EN	Enable link status interrupt from IC2 (ETH1 in RMII mode)	1: interrupt enabled 0: interrupt disabled

Note: This interrupt source can be enabled only, if the FPGAINTSELREG register disables the secondary interrupt source.

5.2.9 FPGAINTPOLREG (0x0A00 0020)

Bit	15	14	13	12	11	10	9	8
Name	INTA1_POL	INTB1_ POL	INTC1_ POL	INTD1_ POL	INTA2_ POL	INTB2_ POL	INTC2_ POL	INTD2_ POL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	INTE_ POL	FPIO4_ POL	RQC0_ POL	RQC1_ POL	Reserved	Reserved	Reserved	Reserved
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	0	0

This register allows selecting the polarity of the interrupt signals coming from the PCI slot, the I/O Board and the on-board Ethernet controller. The interrupt-input signals are forwarded to the CPU via GPIO0. Additional the RQC0 and RQC1 signal can be used as an interrupt request source. These signals are forwarded to the GPIO0 interrupt as well.

Bit	Name	Function	
15	INTA1_POL	Interrupt polarity of INTA1_N from PCI slot	1: active high 0: active low
14	INTB1_ POL	Interrupt polarity of INTB1_N from PCI slot	1: active high 0: active low
13	INTC1_ POL	Interrupt polarity of INTC1_N from PCI slot	1: active high 0: active low
12	INTD1_ POL	Interrupt polarity of INTD1_N from PCI slot	1: active high 0: active low
11	INTA2_ POL	Interrupt polarity of INTA2_N from I/O-Board	1: active high 0: active low
10	INTB2_ POL	Interrupt polarity of INTB2_N from I/O-Board	1: active high 0: active low
9	INTC2_ POL	Interrupt polarity of INTC2_N from I/O-Board	1: active high 0: active low
8	INTD2_ POL	Interrupt polarity of INTD2_N from I/O-Board	1: active high 0: active low
7	INTE_ POL	Interrupt enable for INTE_N from I/O-Board	1: active high 0: active low
6	FPIO4_ POL	Interrupt polarity of FPIO4 signal from Ethernet Controller RTL8139C	1: active high 0: active low
5	RQC0_ POL	Interrupt polarity of RQC0 signal from CSI interface	1: active high 0: active low
4	RQC1_ POL	Interrupt polarity of RQC1 signal from CSI interface	1: active high 0: active low
3:0	Reserved		

Remark: The interrupt output signals of the FPGA (GPIO0) is always high active.

5.2.10 FPGAINTSELREG (0x0A00 0024)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	SxOVR_EN	S1FULL_EN	S0FULL_EN	S1EMPTY_EN
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	S0EMPTY_EN	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R/W	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

The FPGA features more interrupt sources than can be notified in the 16 bits of the FPGAINTREG. For this reason it is possible to select an alternative interrupt source for some interrupt bit in FPGAINTREG. Set a bit in the FPGAINTSELREG and the corresponding bit in FPGAINTREG will indicate an alternative interrupt as described below.

Bit	Name	Function	
15:12	Reserved		
11	SxOVR_EN	Enables OVERRUN interrupt from both I2S RX 1 and 0	1: interrupt enabled 0: interrupt disabled
10	S1FULL_EN	Enables FULL interrupt from I2S 1 RX	1: interrupt enabled 0: interrupt disabled
9	S0FULL_EN	Enables FULL interrupt from I2S 0 RX	1: interrupt enabled 0: interrupt disabled
8	S1EMPTY_EN	Enables EMPTY interrupt from I2S 1 TX	1: interrupt enabled 0: interrupt disabled
7	S0EMPTY_EN	Enables EMPTY interrupt from I2S 0 TX	1: interrupt enabled 0: interrupt disabled
6:0	Reserved		

Remark: If the alternative interrupt source/s is/are disabled, the FPGAINTENREG register can enable the primary interrupt source/s.

5.2.11 ETHER_CONTREG (0x0A00 0050)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	RMII_SEL_JP14	IC3_RST_N	PCI_EN	MII0_RMII0
R/W	R	R	R	R	R	R/W	R	R/W
Reset	0	0	0	0	JP14	0	0	0

This register is used to control IC3 of the *start*WARE-GHS-VR4133. The Ethernet interface (CH2) can be used in two modes – MII and RMII.

Bit	Name	Function	
15:4	Reserved		
3	RMII_SEL_JP14	Reads RMII/MII selection for IC3 at JP14 (Ethernet PHY for VR4133 MAC0)	1: JP14 selects RMII 0: JP14 selects MII
2	IC3_RST_N	Sets or reads reset signal for IC3 (Ethernet PHY for VR4133 MAC0)	1: Reset not asserted 0: Reset asserted
1	PCI_EN	Reads PCI enable (jumper setting)	1: PCI enabled 0: PCI disabled
0	MII0_RMII0	Reads active MII0/RMII0 selection	1: RMII 0: MII

5.2.12 FPIO_MODEREG (0x0A00 0068)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	FPIOMOD (4)	FPIOMOD (3)	FPIOMOD (2)	FPIOMOD (1)	FPIOMOD (0)
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

This register sets the I/O mode of the FPIO (3:0) pins. The correspondence is 1 bit per pin. When the FPIOMOD bit is set to 1, the corresponding FPIO pin is set to output and the value that has been written to the corresponding FPIO bit in the FPIO_REG register is output.
When this bit is set to 0, the corresponding FPIO pin is set to input.

Bit	Name	Function
15:5	Reserved	
4	FPIOMOD (4)	always 0: FPIO(4) is always input FPIO(4) shows the status of the INTAB Pin of the RTL8139C – Ethernet Controller
3:0	FPIOMOD (3:0)	FPIOMOD (3:0) = 0: FPIO (3:0) is input 1: FPIO (3:0) is output

5.2.13 FPIO_REG (0x0A00 006C)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	FPIO (4)	FPIO (3)	FPIO (2)	FPIO (1)	FPIO (0)
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

This register reads and/or writes the FPIO (4:0) pins. The correspondence is 1 bit per pin. When 1 is set to the corresponding bit in the FPIO_MODEREG register, the data in FPIO_REG is written to the corresponding FPIO pin.
When the value of the corresponding bit in the FPIO_MODEREG register is 0, writing a value to the FPIO bit does not affect the FPIO pin (the write data is ignored).
When the value of the corresponding bit in the FPIO_MODEREG register is 0, reading the FPIO bit enables the corresponding FPIO pin's state to be read.

Note: FPIO(4) can only be used as input pins and it is already hard-wired on the board to pins of the Ethernet controller as shown in the table below. FPIO (3:0) pins are available on the I/O board connector CN9.

Bit	Name	Function
15:5	Reserved	
4	FPIO (4)	FPIO(4) pin input data 1: High level 0: Low level FPIO(4) shows the value of the INTAB Pin of the RTL8139C – Ethernet Controller
3:0	FPIO (3:0)	FPIO(3:0) pin input/output data 1: High level 0: Low level

5.2.14 FPGA_I_O_MODEREG (0x0A00 0070)

Bit	15	14	13	12	11	10	9	8
Name	FPGA_I_O_MOD (15)	FPGA_I_O_MOD (14)	FPGA_I_O_MOD (13)	FPGA_I_O_MOD (12)	FPGA_I_O_MOD (11)	FPGA_I_O_MOD (10)	FPGA_I_O_MOD (9)	FPGA_I_O_MOD (8)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	FPGA_I_O_MOD (7)	FPGA_I_O_MOD (6)	FPGA_I_O_MOD (5)	FPGA_I_O_MOD (4)	FPGA_I_O_MOD (3)	FPGA_I_O_MOD (2)	FPGA_I_O_MOD (1)	FPGA_I_O_MOD (0)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

This register sets the I/O mode of the FPGA_I_O (15:0) pins. The correspondence is 1 bit per pin.
When the FPGA_I_O_MOD bit is set to 1, the corresponding FPGA_I_O pin is set to output and the value that has been written to the corresponding FPGA_I_O bit in the FPGA_I_O_REG register is output.
When this bit is set to 0, the corresponding FPGA_I_O pin is set to input.
FPGA_I_O (15:0) are connected to CN9 and FPGA_I_O (13:0) are connected to CN11. Some of these FPGA_I_O pins can be redefined to I²S input/output pins. Please refer to section 5.4 ”I^2S Function” on page 76.

Bit	Name	Function
15:0	FPGA_I_O_MOD (15:0)	FPGA_I_O_MOD (15:0) = 0: FPGA_I_O (15:0) is input 1: FPGA_I_O (15:0) is output

5.2.15 FPGA_I_O_REG (0x0A00 0074)

Bit	15	14	13	12	11	10	9	8
Name	FPGA_I_O (15)	FPGA_I_O (14)	FPGA_I_O (13)	FPGA_I_O (12)	FPGA_I_O (11)	FPGA_I_O (10)	FPGA_I_O (9)	FPGA_I_O (8)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	FPGA_I_O (7)	FPGA_I_O (6)	FPGA_I_O (5)	FPGA_I_O (4)	FPGA_I_O (3)	FPGA_I_O (2)	FPGA_I_O (1)	FPGA_I_O (0)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

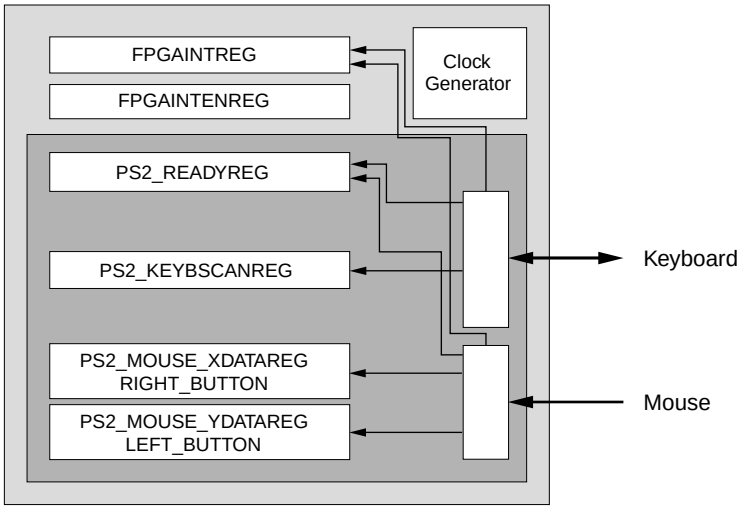
This register reads and/or writes the FPGA_I_O (15:0) pins. The correspondence is 1 bit per pin. When 1 is set to the corresponding bit in the FPGA_I_O_MODEREG register, the corresponding data is written to the FPGA_I_O pin.
When the value of the corresponding bit in the FPGA_I_O_MODEREG register is 0, writing a value to the FPGA_I_O bit does not affect the FPGA_I_O pin (the write data is ignored).
When the value of the corresponding bit in the FPGA_I_O_MODEREG register is 0, reading the FPGA_I_O bit enables the corresponding FPGA_I_O pin's state to be read.
FPGA_I_O (15:0) are connected to CN9 and FPGA_I_O (13:0) are connected to CN11. Some of these FPGA_I_O pins can be redefined to I²S input/output pins. Please refer to section 5.4 "I²S Function" on page 76.

Bit	Name	Function
15:0	FPGA_I_O (15:0)	FPGA_I_O (15:0) pin input/output data 1: High level 0: Low level

5.3 PS2 Function

The FPGA supports two PS/2 interfaces (for Keyboard and Mouse). These interfaces can be used in interrupt and polling mode. The following block diagram shows the relevant parts of the PS/2 block.

Figure 5-1: PS/2 Block Diagram



For using the PS/2 interfaces in interrupt mode, interrupts have to be enabled in the FPGAINTENREG (0xAA00 001C) and in the VR4133. The FPGAINTREG shows the interrupt source, if an interrupt within the FPGA has been occurred. The interrupt generates a signal on GPIO (0) accordingly. For details to handle interrupts on VR4133 side refer to the VR4133 Users Manual.

5.3.1 PS2 Overview

The following table summarizes the PS2 related registers:

Table 5-2: PS/2 - Registers

Address	R/W	Register Symbol	Function
0x0A00 0054	R	PS2_READYREG	Mouse and Keyboard ready
0x0A00 005c	R	PS2_MOUSE_XDATAREG_RIGHT_BUTTON	X Increments of Mouse movement
0x0A00 0060	R	PS2_MOUSE_YDATAREG_LEFT_BUTTON	Y Increments of Mouse movement
0x0A00 0064	R	PS2_KEYBSCANREG	Keyboard scan code

5.3.2 PS2_READYREG (0x0A00 0054)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MOUSE_ DATA_RDY	KEYB_ DATA_RDY
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

This register is used to detect the status of the PS/2 interfaces of the *start*WARE-GHS-VR4133. Two bits of this register show, if a character has been received from the keyboard or the mouse has been moved since the latest read cycle from the PS/2 Mouse interface.
Each bit is cleared on read access to the corresponding keyboard - or mouse register.

Bit	Name	Function
15:2	Reserved	
1	MOUSE_DATA_RDY	Mouse data ready 1: data received, Data in PS2_MOUSE_XDATAREG, PS2_MOUSE_YDATAREG and PS2_MOUSE_BUTTONREG valid 0: no data received, no valid data
0	KEYB_DATA_RDY	Keyboard data ready 1: data received, Data in PS2_KEYBSCANREG valid 0: no data received, no valid data

5.3.3 PS2_MOUSE_XDATAREG (0x0A00 005C)

Bit	15	14	13	12	11	10	9	8
Name	RIGHT_ BUTTON	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MOUSE_ XDATA (8)
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	MOUSE_ XDATA (7)	MOUSE_ XDATA (6)	MOUSE_ XDATA (5)	MOUSE_ XDATA (4)	MOUSE_ XDATA (3)	MOUSE_ XDATA (2)	MOUSE_ XDATA (1)	MOUSE_ XDATA (0)
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

This register contains the latest data (x-direction and right button) received by the PS/2 Mouse interfaces of the *start*WARE-GHS-VR4133. The x-direction data format is 9-bit 2's complement.
The MOUSE_DATA_RDY bit of the PS2_READYREG indicates valid data.

Bit	Name	Function
15	RIGHT_BUTTON	Received Mouse Button status - right button 1: pressed 0: not pressed
14:9	Reserved	
8:0	MOUSE_XDATA (8:0)	Received Mouse data (x-direction) move left - negative value move right - positive value

5.3.4 PS2_MOUSE_YDATAREG (0x0A00 0060)

Bit	15	14	13	12	11	10	9	8
Name	LEFT_ BUTTON	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MOUSE_ YDATA (8)
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	MOUSE_ YDATA (7)	MOUSE_ YDATA (6)	MOUSE_ YDATA (5)	MOUSE_ YDATA (4)	MOUSE_ YDATA (3)	MOUSE_ YDATA (2)	MOUSE_ YDATA (1)	MOUSE_ YDATA (0)
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

This register contains the latest data (y-direction and left button) received by the PS/2 Mouse interfaces of the *start*WARE-GHS-VR4133. The y-direction data format is 9-bit 2's complement.
The MOUSE_DATA_RDY bit of the PS2_READYREG indicates valid data.

Bit	Name	Function
15	LEFT_BUTTON	Received Mouse Button status - left button 1: pressed 0: not pressed
14:9	Reserved	
8:0	MOUSE_YDATA (8:0)	Received Mouse data (y-direction) move down - negative value move up - positive value

5.3.5 PS2_KEYBSCANREG (0x0A00 0064)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	KEYB_ DATA (7)	KEYB_ DATA (6)	KEYB_ DATA (5)	KEYB_ DATA (4)	KEYB_ DATA (3)	KEYB_ DATA (2)	KEYB_ DATA (1)	KEYB_ DATA (0)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

This register contains the latest data received by the PS/2 interfaces of the *start*WARE-GHS-VR4133. The data format is ASCII. Reading this register causes resetting the KEYB_ DATA_RDY bit of the PS2_READYREG register. Writing to this register allows to sent control characters to the PS/2 keyboard. This control characters define the keyboard mode, LED status etc.

Bit	Name	Function
15:8	Reserved	
7:0	KEYB_DATA	Keyboard data (scan code)

Figure 5-2: The AT Scan Codes

F1	F2	`	1	2	3	4	5	6	7	8	9	0	-	=	\	BS	ESC	NUML	SCRL	SYSR
05	06	0E	16	1E	26	25	2E	36	3D	3E	46	45	4E	55	5D	66	76	77	7E	84

F3	F4	TAB	Q	W	E	R	T	Y	U	I	O	P	[	]	HOME	Up	PgUp	PrtSc
04	0C	0D	15	1D	24	2D	2C	35	3C	43	44	4D	54	5B	6C	75	7D	7C

F5	F6	CNTL	A	S	D	F	G	H	J	K	L	;	’	ENTER	Left	5	Right	-
03	0B	14	1C	1B	23	28	34	33	38	42	4B	4C	52	5A	6B	73	74	7B

F7	F8	LSHFT		Z	X	C	V	B	N	M	,	.	/	RSHFT	End	Dn	PgDn	+
83	0A	12		1A	22	21	2A	32	31	3A	41	49	4A	59	69	72	7A	79

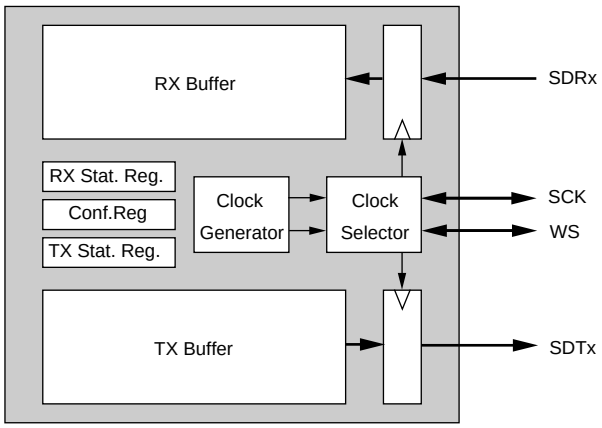
F9	F10	ALT	SPC	CAPLOCK	Ins	Del
01	09	11	29	58	70	71

5.4 I²S Function

5.4.1 I²S Overview

The FPGA features two I²S blocks. The following block diagram shows the relevant parts of the I²S block.

Figure 5-3: I²C Block Diagram



Each block uses the following signals, which are accessible outside the FPGA: serial receive data (SDRx), serial transmit data (SDTx), continuous serial clock (SCK) and word select/strobe (WS). It can be selected, whether the serial clock and word select signals are input or are generated by the I²S block itself.
There are three registers to set the configuration and to show the receive and the transmit status. Each direction has a buffer memory of selectable size, which is split into two pages. Usually one page is accessed by the CPU, while the other is used by the I²S sequencer.

The following table summarizes the registers relevant for the I²S function:

Table 5-3: I²S - Registers

Address	R/W	Register Symbol	Function
0x0A00 0080	R/W	I2S_CONTROLREG	I2S pin select general control register
0x0A00 0090	R/W	I2S0_CONFREG	I2S0 configuration register
0x0A00 0094	R/W	I2S0_RX_STATREG	I2S0 reception status register
0x0A00 0098	R/W	I2S0_TX_STATREG	I2S0 transmission status register
0x0A00 00A0	R/W	I2S1_CONFREG	I2S1 configuration register
0x0A00 00A4	R/W	I2S1_RX_STATREG	I2S1 reception status register
0x0A00 00A8	R/W	I2S1_TX_STATREG	I2S1 transmission status register

The following table summarizes the memory areas used by the I^2S blocks:

Table 5-4: I²S Memory Areas

Address	R/W	Memory Area	Function
0x0A00 1000 - 0x0A00 13FF	R	I2S0_RX_BUFFER	Receive Buffer of I2S0
0x0A00 2000 – 0x0A00 23FF	R	I2S1_RX_BUFFER	Receive Buffer of I2S1
0x0A00 3000 – 0x0A00 33FF	R/W	I2S0_TX_BUFFER	Transmit Buffer of I2S0
0x0A00 4000 – 0x0A00 43FF	R/W	I2S1_TX_BUFFER	Transmit Buffer of I2S1

5.4.2 I^2S Signals on CN12

The following figure shows how the signals of the I^2S blocks are shared with the FPGA_I_O which are available on CN12:

Figure 5-4: Pin assignment at CN12 (I^2S usage)

	1			2	VDD 3.3
GND	3			4	S0_SCK [FPGA_I_O(13)]
[FPGA_I_O(12)] S0_WS	5			6	S0_SDRx [FPGA_I_O(11)]
[FPGA_I_O(10)] S0_SDTx	7			8	GND
[FPGA_I_O(9)] S1_WS	9			10	S1_SDRx [FPGA_I_O(8)]
[FPGA_I_O(7)] S1_SCK	11			12	S1_SDTx [FPGA_I_O(6)]
GND	13			14	FPGA_I_O(5)
FPGA_I_O(4)	15			16	FPGA_I_O(3)
FPGA_I_O(2)	17			18	FPGA_I_O(1)
FPGA_I_O(0)	19			20	GND

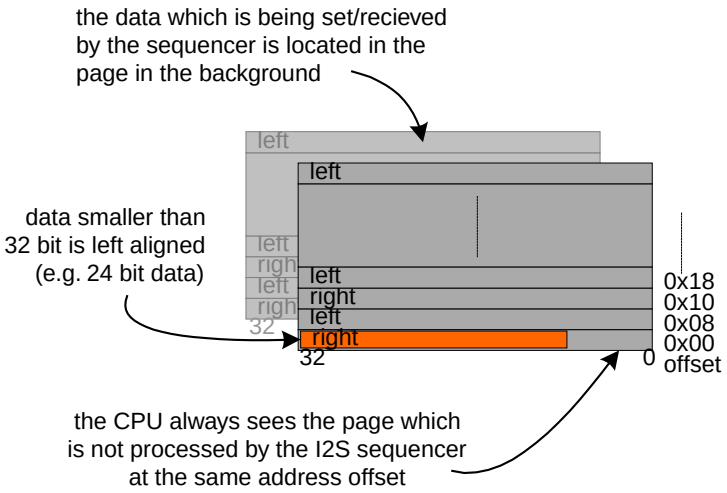
Each I^2S channel can be activated separately. The signals related to a disabled channel are assigned to the FPGA_I_O functionality and can be used as input or output depending on the FPGA_I_O_MODEREG.

5.4.3 I²S Buffer Structure

Both buffers for transmit and receive direction are separated into two pages. The CPU only sees one page at the same address offset. The other page is processed in the background by the sequencer, i.e. it is filled by the received data or used to transmit data. Once the sequencer has processed the whole page and the CPU has filled/read out the whole page, the pages are switched.

The following figure illustrates the use of the buffers.

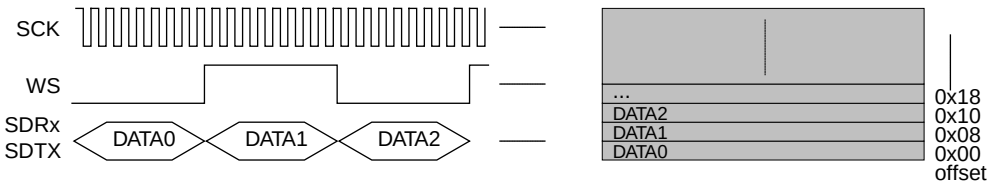
Figure 5-5: I²S Buffer Structure



The size of the buffer can be set in the CONFREG of each I²S block from 32 words to 256 words. The data length is selected in the CONFREG as well, in case the I²S is clock master. When an external device generates the clock signals, the data width is determined by the length of the word select (WS) pulse related to the serial clock signal (SCK).

The following figure illustrates how the serial data is placed in the buffer:

Figure 5-6: Location of received and transmitted Data



If the data length is smaller than 32-bit the data will be stored in the buffer left aligned. The lower bits will remained unused (i.e. you can only take advantage of the full buffer size if you select 32-bit data width).

Note: As the FPGA is only connected to the CPU with 16 bit width you need to perform two 16-bit operations to access one line of the buffer (except if you choose 16 bit data length of course).

5.4.4 I²S Initialization

To initialize the I²S block the following steps are recommended:

- Set the I2S_CONTROLREG register to 0 to reset the I²S blocks
- Set the I2S_CONTROLREG register bits:
  to enable the desired I²S block (I2Sx_EN),
  to select whether the corresponding I²S block is clock master (I2Sx_CLK_EN) and
  to set the correct CPU data access width to the I²S block.
- Set the I2Sx_CONFREG register of the corresponding I²S block:
  to set the frequency of the transmit clocks (I2S_P bits) – if the I²S block is clock master
  to set the buffer size for transmission and reception (I²S_PL bits)
  to set the data length (I2S_DL bits) – if the I²S block is clock master
- Set the FPGAINTSELREG register to enable the desired I²S interrupts

5.4.5 I²S Transmit Operation

The following is a example of a typical transmit operation:

- Fill the transmit buffer with the number of words as set in I2S_PL in the I2Sx_CONFREG
  start address is I2Sx_TX_BUFFER
  increment the address according to the access width set in the I2S_CONTROLLREG if the data width is less than 32-bit (I2S_DL in I2Sx_CONFREG) the data is left aligned
  please refer also to section 5.4.7 ”Accessing the Transmit or Receive Page Buffer” on page 81 for details about the page buffer access
- When the last word to fill the buffer is written the following happens:
  The PAGE_FULL bit in the I2Sx_TX_STATREG is set
  The sequencer will start to process (i.e. send) the buffer page which has just been filled
  Now the second page (which is empty) is accessible at the address I2Sx_TX_BUFFER
- Fill the second page with the number of words as set in I2S_BL in the I2Sx_CONFREG
  The PAGE_FULL bit in the I2Sx_TX_STATREG is set
- Wait until the PAGE_EMPTY_IRQ bit in the I2Sx_TX_STATREG register is set
  The FPGA will generate an interrupt to the CPU if enabled in FPGAINTSELREG
  The PAGE_EMPTY_IRQ will be cleared after the next write access to the transmit buffer
- Loop “Fill page” and “wait for PAGE_EMPTY_IRQ”
- If both buffer pages have been sent on the I²S interface and the CPU has not filled a buffer page with new data an underrun occurs. This will be notified by setting the UNDERRUN bit in the I2Sx_TX_STATREG. However the send process will continue and the old buffer data will be sent again.

Note: If the corresponding I²S block is set to be the clock master – the clock generation is started as soon as the sequencer starts operation as described above. At the same time the receive operation is also started and the receive buffer is filled with data sampled from the Sx_SDRx pin. If you do not intend to receive data just ignore the signals related with the data reception (e.g. mask the corresponding interrupts).

5.4.6 I^2S Receive Operation

The reception of data (i.e. filling the receive buffer with data sampled from the Sx_SDRx pin) will start under the following circumstances:

- I^2S block is clock master and I2Sx_SYNC_MODE in I2S_CONTROLREG is cleared:
  Reception starts when transmission is started after filling first page of transmit data
- I^2S block is clock master and I2Sx_SYNC_MODE in I2S_CONTROLREG is set:
  Reception starts once the I2S_RX_EN is set in the I2Sx_CONFREG
- External I^2S device is clock master:
  Reception starts once the external device generates the SCK clock signal.

The following shows an example of a typical receive operation:

- The receive buffer page is filled with data
- Once one page is filled, the PAGE_FULL_IRQ in the I2Sx_RX_STATREG register is set
  The page which has just been filled is now accessible at the address I2S0_RX_BUFFER
  The sequencer will continue to receive data in the background, filling the second page
- The CPU reads out the receive buffer page
  completely read the number of words as set in I2S_PL in the I2Sx_CONFREG
  increment the address according to the access width set in the I2S_CONTROLLREG
  if the data width is less than 32-bit (I2S_DL in I2Sx_CONFREG) the data is left aligned
  please refer also to section 5.4.7 ”Accessing the Transmit or Receive Page Buffer” on page 81 for details about the page buffer access
  when the last data is read, the READ_ALL_DATA bit in I2Sx_RX_STATREG is set
  after the whole page is read, the PAGE_FULL_IRQ is cleared
- Loop “wait for PAGE_FULL_IRQ” “read out buffer”

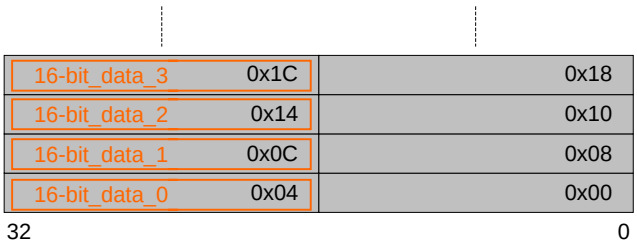
If the CPU does not read out the complete receive buffer page in the time the second (background) page is filled by the sequencer, there will be no place to store further received data. In this case the PAGES_OVF bit in the I2Sx_RX_STATREG register is set to indicate a buffer overflow and potential data loss.

5.4.7 Accessing the Transmit or Receive Page Buffer

Please keep in mind that the access to the buffer differs depending on the data length (set by I2S_DL in I2Sx_CONFREG). This is due to the fact that the CPU can only read or write 16 bits at a time since the physical connection to the FPGA is only 16-bit wide.

In the simple case of a 16-bit data length the data is located in the buffer page as illustrated in the following figure:

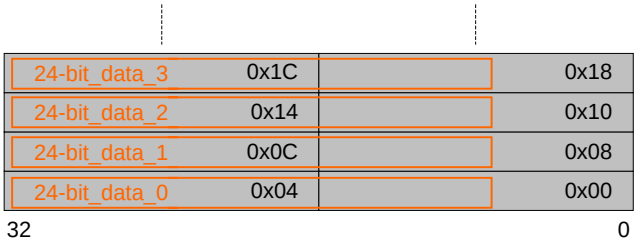
Figure 5-7: 16-bit Data Format



This mean to read or write the buffer page the CPU has to access the first data at the address offset 0x4, the second data at the offset 0x0C and so on. I.e. the CPU has to perform halfword accesses starting from the offset 0x04, incrementing the address by 0x8.

If the data length is selected larger than 16 bits, the CPU has to perform two halfword accesses to read or write a data in the buffer page. The following figure shows the example how 24-bit data are located in the buffer page:

Figure 5-8: 24-bit Data Format



This means that the CPU has to write the lower bits of the first data (bits 7 to 0) left-aligned to the address offset 0x0. The upper bits (bits 23 to 8) have to be written to the address offset0x04. The lower bits of the second data are located at the address offset 0x08, the upper bits at the offset 0x0C and so on.

5.4.8 I^2S Interrupts

Interrupts generated by the I2S blocks can be enable in the FPGAINTSELREG register and are notified in the FPGAINTREG register.

Here is the relation of the interrupts and the I^2S registers:

- **S0EMPTY** interrupt:
  This interrupt is asserted when the bit PAGE_EMPTY_IRQ is set in I2S0_TX_STATREG.
  It is cleared the I2S0_TX_BUFFER area is completely filled with new data.
- **S1EMPTY** interrupt:
  This interrupt is asserted when the bit PAGE_EMPTY_IRQ is set in I2S1_TX_STATREG.
  It is cleared when the I2S1_TX_BUFFER is completely filled with new data.
- **S0FULL** interrupt:
  This interrupt is asserted when the bit PAGE_FULL_IRQ is set in I2S0_RX_STATREG.
  It is cleared after the whole page is read from the I2S0_RX_BUFFER area
- **S1FULL** interrupt:
  This interrupt is asserted when the bit PAGE_FULL_IRQ is set in I2S1_RX_STATREG.
  It is cleared after the whole page is read from the I2S1_RX_BUFFER area
- **SxOVF** interrupt:
  This interrupt is set when the PAGES_OVF bit in either the I2S0_RX_STATREG or I2S1_RX_STATREG is set.
  It can be cleared by resetting the corresponding I2S block (clear I2Sx_EN bit in the I2S_CONTROLREG register)

5.4.9 I2S_CONTROLREG (0x0A00 0080)

Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	I2S1_SYNC_MODE	I2S0_SYNC_MODE	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R/W	R/W	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	I2S1_CLK_EN	I2S0_CLK_EN	I2S1_TEST	I2S0_TEST	I2S1_EN	I2S0_EN
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

This register is used to set some general features of the two I^2S blocks:

The corresponding FPGA pins can either be assigned to the FPGA_I_O unit or to the I^2S unit of the FPGA. If the I^2S blocks are not enabled, these pins can be used as FPGA_I_O pins.

Disabling a I^2S block also shuts down the whole block. In that way disabling and enabling an I^2S block will also perform a reset of this unit.

Also the direction of the I^2S clock pins can be selected. This determines whether the I^2S block acts as a clock master or clock slave (i.e. the clock signals are output or input).

Bit	Name	Function
15:12		Reserved
13	I2S1_SYNC_MODE	Enables that the reception operation of I2S1 block starts synchronous with the transmission process (i.e. when one TX buffer page is completely filled) 0: not synchronous (RX will start once RX is enabled) 1: synchronous (RX will start when TX starts)
12	I2S0_SYNC_MODE	Enables that the reception operation of I2S0 block starts synchronous with the transmission process (i.e. when one TX buffer page is completely filled) 0: not synchronous (RX will start once RX is enabled) 1: synchronous (RX will start when TX starts)
11:6		Reserved
5	I2S1_CLK_EN	Enables I2S1 clock signal output mode 0: S1_WS and S1_SCK are input signals (external device generates clock signals) 1: S1_WS and S1_SCK are output signals (I2S1 block generates clock signals)
4	I2S0_CLK_EN	Enables I2S0 clock signal output mode 0: S0_WS and S0_SCK are input signals (external device generates clock signals) 1: S0_WS and S0_SCK are output signals (I2S0 block generates clock signals)
3	I2S1_TEST	Enable the test mode for I2S1 0: normal operation 1: test mode (test data is automatically generated)
2	I2S0_TEST	Enable the test mode for I2S0 0: normal operation 1: test mode (test data is automatically generated)
1	I2S1_EN	Enables I2S1 pins and I2S1 block
0	I2S0_EN	Enables I2S0 pins and I2S0 block

Note: Make sure only one I^2S block and only one direction is selected at any time!

Setting the I2Sx_TEST bit will start the generation of WS and SCK signals of the corresponding I^2S interface. SCK and WS are generated according to the setting of I2S_DL and I2S_BL in the I2Sx_CONFREG. If the I2S_TX_EN bit is set in this register, some pre-defined data stream will be transmitted to the Sx_SDTx pin.

5.4.10 I2S0_CONFREG (0x0A00 0090)

Bit	15	14	13	12	11	10	9	8
Name	I2S_TX_EN	I2S_RX_EN	I2S_DL2	I2S_DL1	I2S_DL0	I2S_BL1	I2S_BL0	I2S_P8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	I2S_P7	I2S_P6	I2S_P5	I2S_P4	I2S_P3	I2S_P2	I2S_P1	I2S_P0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	1	1

This register configures the I²S interface. Within this register the data length of the incoming bit stream, the FIFO length and the divider for selecting the Bit rate have to be defined.

Bit	Name	Function
15	I2S_TX_EN	Setting this bit will enable the I2S0 transmit function
14	I2S_RX_EN	Setting this bit will enable the I2S0 receive function
13:11	I2S_DL[2:0] **Note 1**	Data length I2S_DL[2:0] = 000: 16-Bit I2S_DL[2:0] = 001: 17-Bit I2S_DL[2:0] = 010: 18-Bit I2S_DL[2:0] = 011: 19-Bit I2S_DL[2:0] = 100: 20-Bit I2S_DL[2:0] = 101: 24-Bit I2S_DL[2:0] = 110: 28-Bit I2S_DL[2:0] = 111: 32-Bit
10:9	I2S_PL[1:0]	Page length **Note 2** I2S_PL[1:0] = 00: 32 words I2S_PL[1:0] = 01: 64 words I2S_PL[1:0] = 10: 128 words I2S_PL[1:0] = 11: reserved
8:0	I2S_P[8:0] **Note 1** **Note 3**	Prescaler value for S0_SCK (S0 bit clock) generation if I2S0 is generating the clocks (depending on I2S_PINASSIGNREG setting) 0: disable clock generation other value: S0_SCK = 50 MHz / ((I2S_P[8:0] +1)*2)

Notes: 1. The settings for the data length and the clock frequency (set by I2S_DL[2:0] and I2SP[8:0] are only effective if this I2S block is configured to be the clock master. If an external device is clock master, it determines the frequency of SCK and WS. In that case the data length is set by the number of SCK cycles within one WS half-cycle.

2. One word always consists of the number of bits as defined in the I2S_DL field.
Example: I2S_PL=0 (32 words) and I2S_DL=0 (16 bit) will result in a page size of 64 bytes and a total buffer size of 128 byte. (See also section 5.4.7 ”Accessing the Transmit or Receive Page Buffer” on page 81)

3. Once the prescaler value I2S_P[8:0] is set, it should not be changed while the I^2S block is operating. I2S_P[8:0] has to be set to 0 before setting a different value.

5.4.11 I2S0_RX_STATREG (0x0A00 0094)

Bit	15	14	13	12	11	10	9	8
Name	PAGE_ FULL_IRQ	PAGES_ OVF	READ_ ALL_DATA	INIT_DONE	Reserved	Reserved	Reserved	Reserved
R/W	R	R/W	R/W	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	PAGE_FULL_IRQ	Indicates that one complete page has been received and is ready to be read by the CPU 0: not enough data received to fill one page 1: received data filled one, which can be read by the CPU
14	PAGES_OVF	Indicates that a receive data overflow has happened (i.e. received data filled more that two pages without any read-out by the CPU)
13	READ_ALL_DATA	Indicates that still some unread data is remaining in the active read-out page 0: still some unread data in the active read-out page 1: all data of the active read-out page is read
12	INIT_DONE	Indicates that the I2S block has been initialized 1: Initialization completed 0: Initialization not completed
11:0		Reserved

The bits PAGE_FULL_IRQ and PAGES_OVF will generate an FPGA interrupt signal to the CPU (FPGAINTREG) if the register FPGAINTSELREG is set accordingly.

5.4.12 I2S0_TX_STATREG (0x0A00 0098)

Bit	15	14	13	12	11	10	9	8
Name	PAGE_ EMPTY_ IRQ	UNDER RUN	Reserved	INIT_DONE	PAGE_BIT	A10	A9	A8
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	A7	A6	A5	A4	A3	A2	A1	A0
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	PAGE_EMPY_IRQ	Indicates that a buffer page has been completely sent and is now ready to be filled with new data by the CPU 1: page empty 0: page not empty (transmission still ongoing)
14	UNDERRUN	Indicates a buffer page has not been refilled in time. This will result in multiple re-send of the previous buffer data 1: underrun occurred 0: no underrun occurred
13		Reserved
12	INIT_DONE	Indicates that the I^2S block has been initialized 1: Initialization completed 0: Initialization not completed
11	PAGE_BIT	Indicates which of the two pages is currently being processed by the transmission sequencer
10:0	A[10:0]	Address showing which data in the active-sent page is currently being processed

The bit PAGE_EMPTY will generate an FPGA interrupt signal to the CPU (FPGAINTREG) if the register FPGAINTSELREG is set accordingly.

5.4.13 I2S1_CONFREG (0x0A00 00A0)

Bit	15	14	13	12	11	10	9	8
Name	I2S_TX_EN	I2S_RX_EN	I2S_DL2	I2S_DL1	I2S_DL0	I2S_BL1	I2S_BL0	I2S_P8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	I2S_P7	I2S_P6	I2S_P5	I2S_P4	I2S_P3	I2S_P2	I2S_P1	I2S_P0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	1	1

This register configures the I^2S interface. Within this register the data length of the incoming bit stream, the FIFO length and the divider for selecting the Bit rate have to be defined.

Bit	Name	Function
15	I2S_TX_EN	Setting this bit will enable the I2S0 transmit function
14	I2S_RX_EN	Setting this bit will enable the I2S0 receive function
13:11	I2S_DL[2:0] **Note 1**	Data length I2S_DL[2:0] = 000: 16-Bit I2S_DL[2:0] = 001: 17-Bit I2S_DL[2:0] = 010: 18-Bit I2S_DL[2:0] = 011: 19-Bit I2S_DL[2:0] = 100: 20-Bit I2S_DL[2:0] = 101: 24-Bit I2S_DL[2:0] = 110: 28-Bit I2S_DL[2:0] = 111: 32-Bit
10:9	I2S_PL[1:0]	Page length **Note 2** I2S_PL[1:0] = 00: 32 words I2S_PL[1:0] = 01: 64 words I2S_PL[1:0] = 10: 128 words I2S_PL[1:0] = 11: reserved
8:0	I2S_P[8:0] **Note 1** **Note 3**	Prescaler value for S0_SCK (S0 bit clock) generation if I2S0 is generating the clocks (depending on I2S_PINASSIGNREG setting) 0: disable clock generation other value: S0_SCK = 50 MHz / ((I2S_P[8:0] +1)*2)

Notes: 1. The settings for the data length and the clock frequency (set by I2S_DL[2:0] and I2SP[8:0] are only effective if this I^2S block is configured to be the clock master. If an external device is clock master, it determines the frequency of SCK and WS. In that case the data length is set by the number of SCK cycles within one WS half-cycle.

2. One word always consists of the number of bits as defined in the I2S_DL field.
Example: I2S_PL=0 (32 words) and I2S_DL=0 (16 bit) will result in a page size of 64 bytes and a total buffer size of 128 byte. (See also section 5.4.7 ”Accessing the Transmit or Receive Page Buffer” on page 81)

3. Once the prescaler value I2S_P[8:0] is set, it should not be changed while the I^2S block is operating. I2S_P[8:0] has to be set to 0 before setting a different value.

5.4.14 I2S1_RX_STATREG (0x0A00 00A4)

Bit	15	14	13	12	11	10	9	8
Name	PAGE_ FULL_IRQ	2PAGES_ OVF	READ_ ALL_DATA	INIT_DONE	Reserved	Reserved	Reserved	Reserved
R/W	R	R/W	R/W	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	PAGE_FULL_IRQ	Indicates that one complete page has been received and is ready to be read by the CPU 0: not enough data received to fill one page 1: received data filled one, which can be read by the CPU
14	PAGES_OVF	Indicates that a receive data overflow has happened (i.e. received data filled more that two pages without any read-out by the CPU)
13	READ_ALL_DATA	Indicates that still some unread data is remaining in the active read-out page 0: still some unread data in the active read-out page 1: all data of the active read-out page is read
12	INIT_DONE	Indicates that the I^2S block has been initialized 1: Initialization completed 0: Initialization not completed
11:0		Reserved

The bits PAGE_FULL_IRQ and PAGES_OVF will generate an FPGA interrupt signal to the CPU (FPGAINTREG) if the register FPGAINTSELREG is set accordingly.

5.4.15 I2S1_TX_STATREG (0x0A00 00A8)

Bit	15	14	13	12	11	10	9	8
Name	PAGE_ EMPTY_ IRQ	UNDER RUN	Reserved	INIT_DONE	PAGE_BIT	A10	A9	A8
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Name	A7	A6	A5	A4	A3	A2	A1	A0
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function
15	PAGE_EMPY_IRQ	Indicates that a buffer page has been completely sent and is now ready to be filled with new data by the CPU 1: page empty 0: page not empty (transmission still ongoing)
14	UNDERRUN	Indicates a buffer page has not been refilled in time. This will result in multiple re-send of the previous buffer data 1: underrun occurred 0: no underrun occurred
13		Reserved
12	INIT_DONE	Indicates that the I^2S block has been initialized 1: Initialization completed 0: Initialization not completed
11	PAGE_BIT	Indicates which of the two pages is currently being processed by the transmission sequencer
10:0	A[10:0]	Address showing which data in the active-sent page is currently being processed

The bit PAGE_EMPTY will generate an FPGA interrupt signal to the CPU (FPGAINTREG) if the register FPGAINTSELREG is set accordingly.

Chapter 6 Board Operation

This chapter explains practical usage of the *start*WARE-GHS-VR4133 evaluation board.

6.1 Getting Started

This chapter gives a “step-by-step” description, how to start up the board.

- Check if all jumpers are in their default position
- Check if all DIP switches are in their default position
- Connect the *start*WARE-GHS-VR4133 Evaluation board with the serial cable to a host PC with a terminal emulation program (115200 baud, 8 bit, no parity, 1 stop bit, no handshake); use the VR4133 Debug Serial interface on CN27-2 /upper Sub-D) for this purpose
- Connect the power supply to the board
- Start the terminal emulation program
- Operate power switch SW3 (LEDs D1 and D2 should be activated)
- FPGA configuration will be done, which is indicated by LED D3
- LED D6 shows that the automatically generated RTCRST is active
- Push POWER button SW6 if JTAG (N-Wire) interface is disabled
- LEDs D8 and D10 should be activated.

Note: The startup sequence is different depending on the jumper setting JP23 (JTAGEN). If JTAG (N-Wire) interface is enabled (default jumper setting), the board starts automatically after switching on SW3. Otherwise push POWER button SW6 for starting the board.

Now the board is ready to communicate with the host PC using the S-Boot monitor program that is described in the next chapter.

6.2 S-Boot Operation

6.2.1 Features

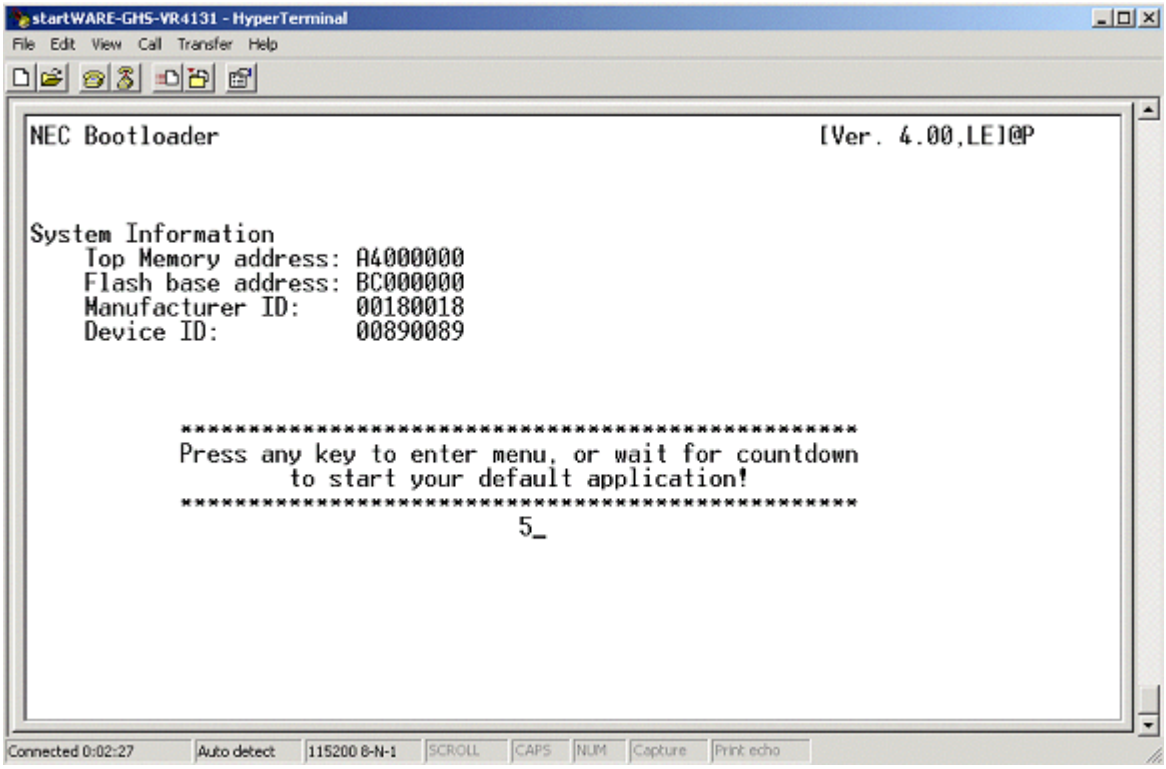
A simple boot monitor named SBOOT is implemented on the *start*WARE-GHS-VR4133 starter kit. This monitor offers the following features:

- A fast serial connection using the CPUs DSIU at 115200 Baud,
- Download capabilities to SDRAM,
- Download to FLASH,
- Automatic, programmable start of applications available in RAM or FLASH,
- A small Flash File System with simple directory structure.

6.2.2 S-Boot Startup Message

Once the starter kit is powered up turned on, you will find this screen on the host PC.

Figure 6-1: Startup Screen



The screen gives information about the current system configuration, please see Figure 6-1.

Remarks: **1.** Version information (Example - 4.0)

2. Endianess:
LE - Little Endian,
BE - Big Endian

3. Run mode:
@ - start from Flash
* - start from RAM

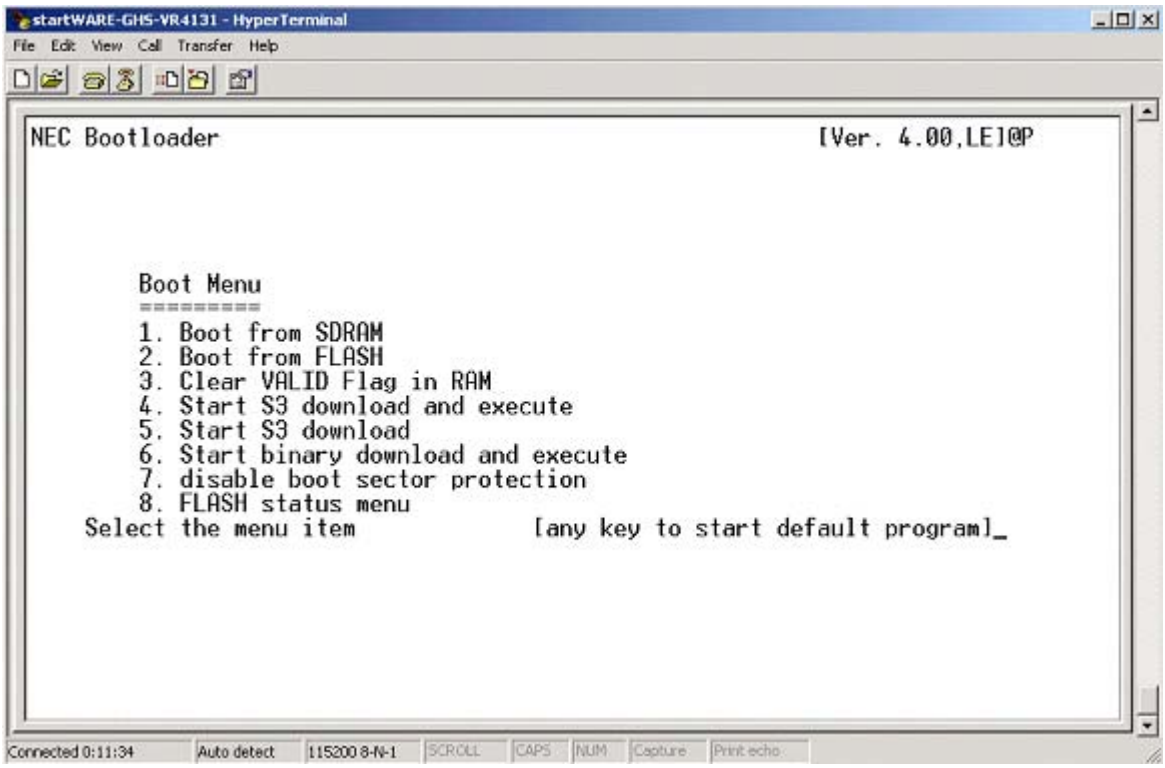
4. Protection settings:
P - Protected,
N - Not protected

The running countdown can be stopped by any input. Once the countdown reaches 0, the system will try to boot. In the first instance, it will try booting from SDRAM, if any information is available that there is something loaded. If none is found, it continues to check for information in the FLASH memory. If there is an application available in any of the FLASH locations, it starts this application. Please see item ‘Flowchart’ for a more detailed description of the procedure.

6.2.3 Flash Monitor of startWARE-GHS-VR4133

When the countdown is interrupted, the SBOOT stops boot procedure and an additional screen is shown:

Figure 6-2: startWARE-GHS-VR4133 Flash Monitor Boot Menu



This is the main menu of the system and provides access to various other options of the SBOOT. The user may manually boot from RAM or FLASH.

(1) Boot from SDRAM
SBOOT tries to boot from memory location last recently loaded with an application.

(2) Boot from FLASH
SBOOT tries to boot from a valid FLASH memory location available in the directory list and marked as ‘default’ start-up.

(3) Clear VALID Flag in RAM
makes an application located in main memory unavailable for next boot process.

(4) Start S3 download and execute
implies the option to download an S-record type file and start it directly after download.

(5) Start S3 download.
Behaves like item 4, but does not automatically start the application.

(6) Start binary download and execute.
Using this option starts a download of a raw binary file, recommended for larger images. Once this option is invoked, the user is asked for

a) a flash memory address, where the application is downloaded to,
b) a start address of the application.
This is necessary for images providing a different entry point than the first target address is.

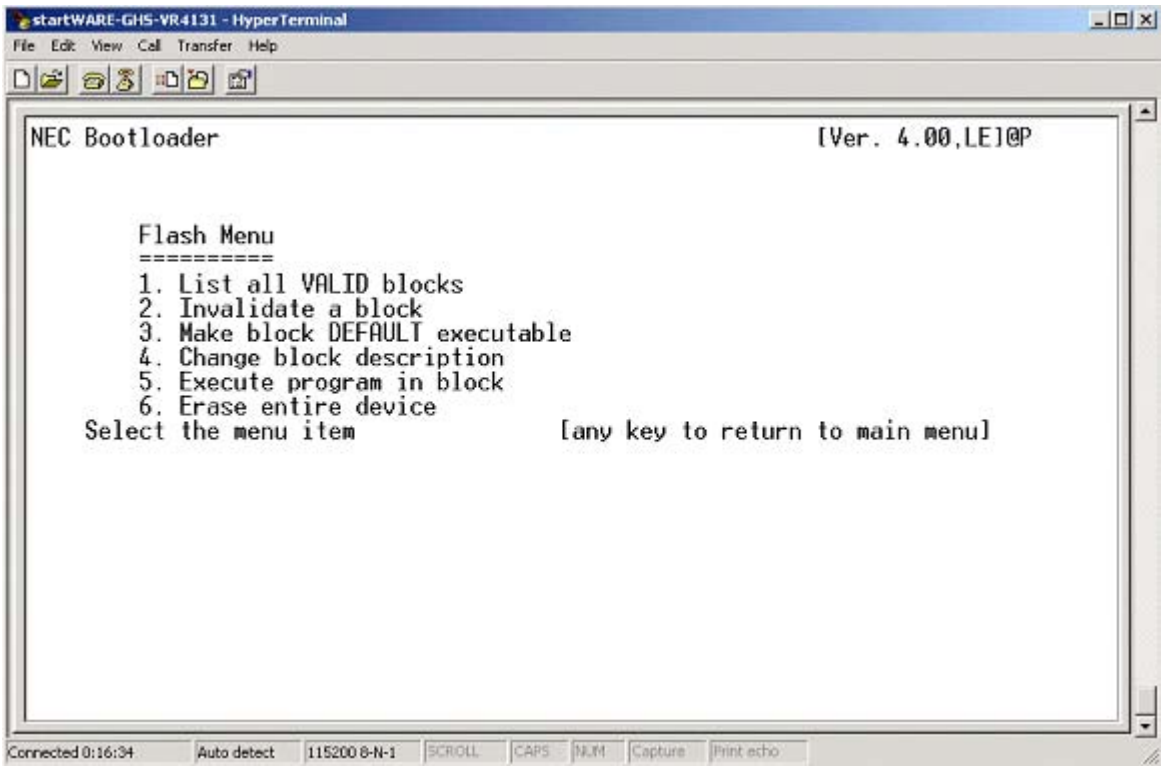
(7) Disable boot sector protection
is not available for the Starter Kit.

(8) FLASH status menu
opens a submenu handling the various FLASH options.

6.2.4 FLASH Options

There are several option available making the handling of the FLASH easier. Several target applications may reside in the FLASH memory and can be managed by SBOOT.

Figure 6-3: startWARE-GHS-VR4133 Flash Menu



(1) List all VALID blocks
displays a list of applications available in the FLASH memory. The default screen looks as shown in Figure 6-4.

(2) Invalidate a block
disables the directory entry and makes the application unavailable for the next boot process.

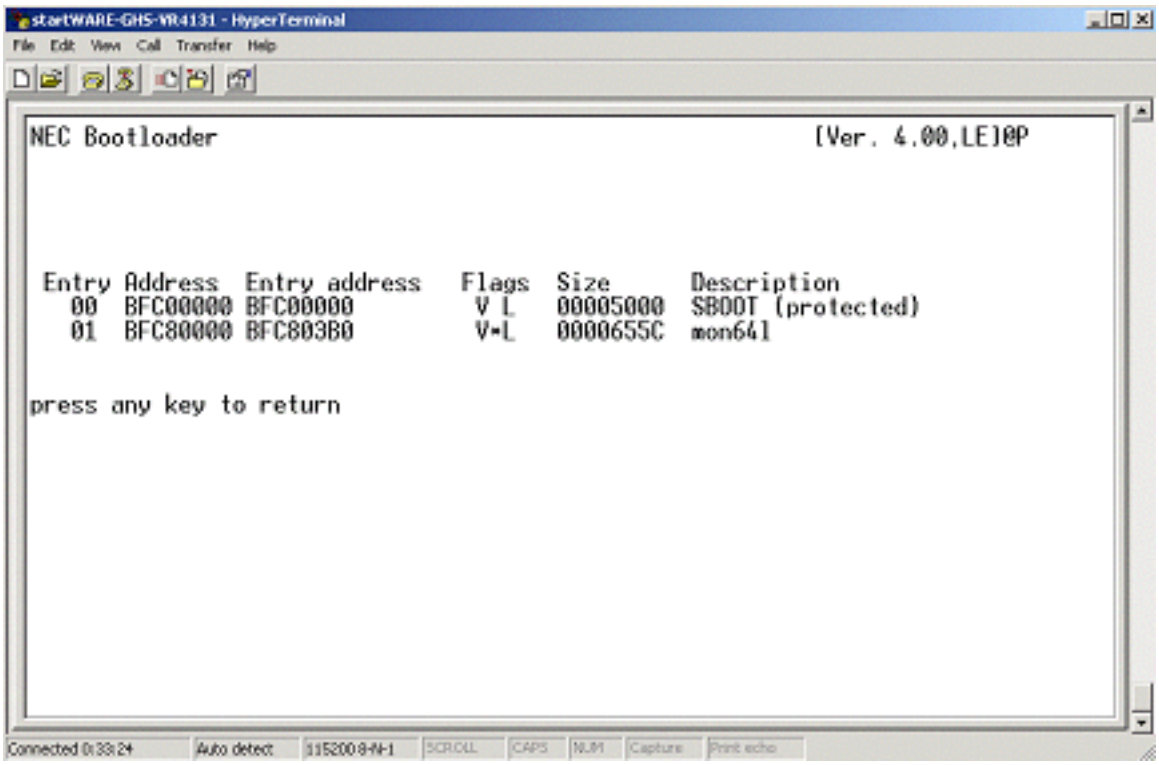
(3) Make block DEFAULT executable
gives the option to boot an application from a block specified by the User. The application must have a valid entry in the directory block structure, before it can be started by default boot process. In the Starter Kit, the GHS target monitor, here named mon64l is the default application to start with. Any last recently downloaded application to FLASH will automatically get the DEFAULT boot flag. This menu entry gives the User the option to change it manually.

(4) Change block description
of a downloaded application, if the automatically provided name is not sufficient. It is a simple text entry to name that directory entry into a more specific description. A maximum number of 32 characters is allowed here.

(5) Execute program in block
offers the option to start any of the available applications manually.

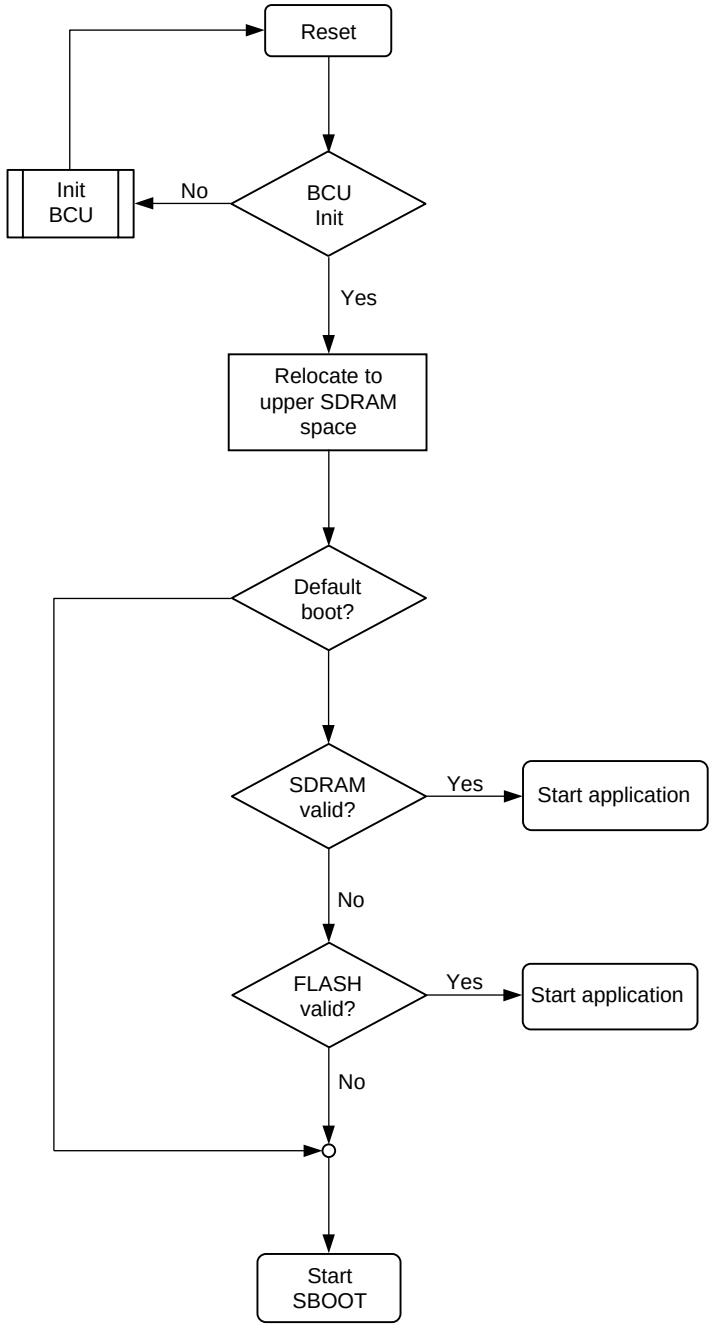
(6) The entire FLASH device is erased, except for the boot block. All applications are lost and the directory entry list is emptied as well.

Figure 6-4: startWARE-GHS-VR4133 Directory List



6.2.5 Reset Process Flow

Figure 6-5: startWARE-GHS-VR4133 Boot Flow Chart



6.2.6 SBOOT Specification

(1) Interrupts

SBOOT does not use any interrupt. The serial units are used in polling mode only. All interrupts are routed to addresses 0x80000000, 0x80000080, 0x80000100 and 0x80000180. It is recommended that all user applications are making usage of the BEV bit, so that own interrupt handling routines can be entered at the desired locations. The non-maskable interrupt (NMI) is treated as ordinary reset – no special handling implemented in here.

(2) Initialization

The caches are initialized prior to any download. A timer is not implemented. Only for the operation of SBOOT necessary peripheral units are supplied with a clock and are initialized.

(3) Memory Usage

The memory area used by SBOOT is from ranging from 0xA3FFB000 to 0xA7FFFFFF.

The SDRAM boot flags are stored in ‘Top memory location’ –0x8 and 0x4. Once an application is loaded into RAM, one address is containing the text ‘EXEC’; the other one contains the execution address of the last recently loaded application.

SBOOT allocates the reset vector location of the VR-CPU, i.e. the FLASH block located at address 0XBFC00000 cannot be used by any target applications. Usually, there is one more monitor delivered with the Starter Kit – the Green Hills target monitor located at 0xBFC80000. It is running from FLASH memory area, which prohibits a programming of the FLASH, unless the FLASH burning algorithm is located in SDRAM. In addition, SBOOT is protected by H/W bit, so that no accidental erase may happen.

6.3 Reprogramming the FPGA

Caution: The functions described in this chapter should only be used by experienced customers. Reprogramming the FPGA in the wrong way, may damage the board. So watch your step!

A short summary how the FPGA is reprogrammed is given below. The current FPGA content has been developed with the Quartus II (Rev. 3.0) tool which can be downloaded from the Altera website (www.altera.com). This tool is capable of downloading FPGA content via JTAG with special cables that are also available from Altera. (During development of the board the ByteBlasterII™ cable was used.).

- Connect the *start*WARE-GHS-VR4133 evaluation board via the download cable to your host PC (CN13 on the evaluation board)
- Power on the evaluation board with SW3
- Download the FPGA code
- Power off the evaluation board with SW3
- Power on the evaluation board with SW3 and the new FPGA content.

Appendix A Circuit Diagrams

Figure A-1: CPU (VR4133) Circuit Diagram (1/20)

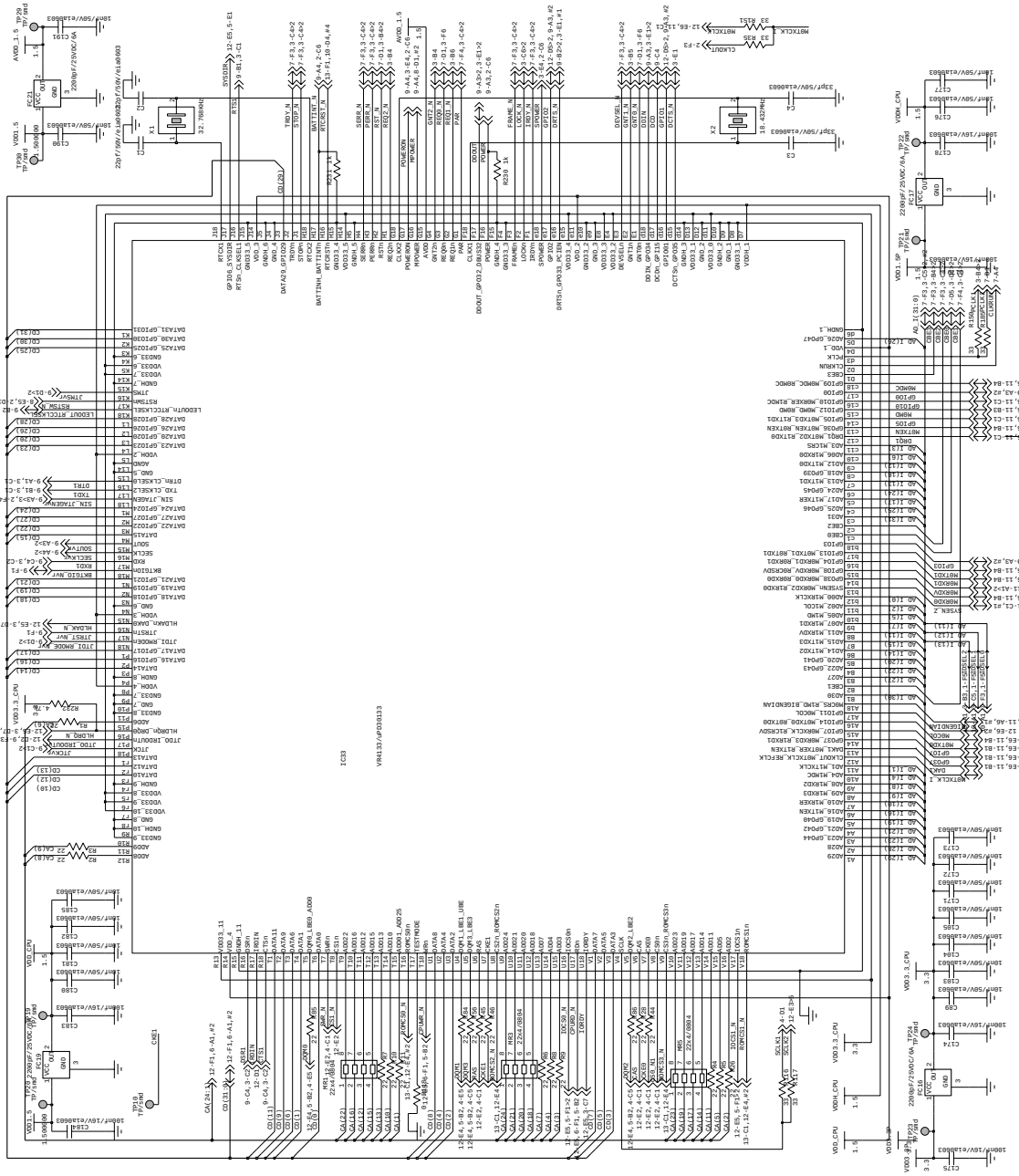


Figure A-2: FPGA Circuit Diagram (2/20)

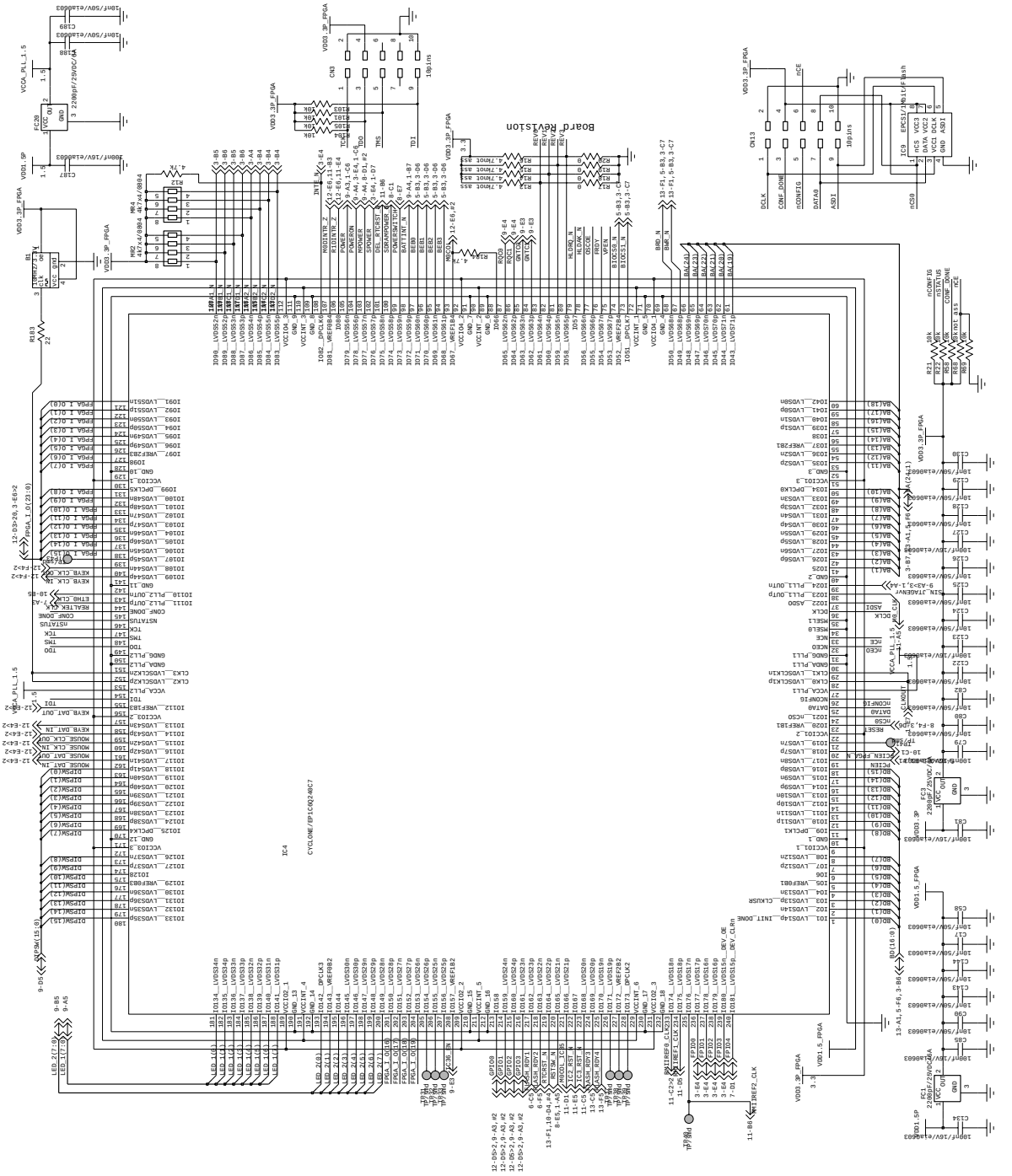


Figure A-3: Board Connectors and Serial Interfaces Circuit Diagram (3/20)

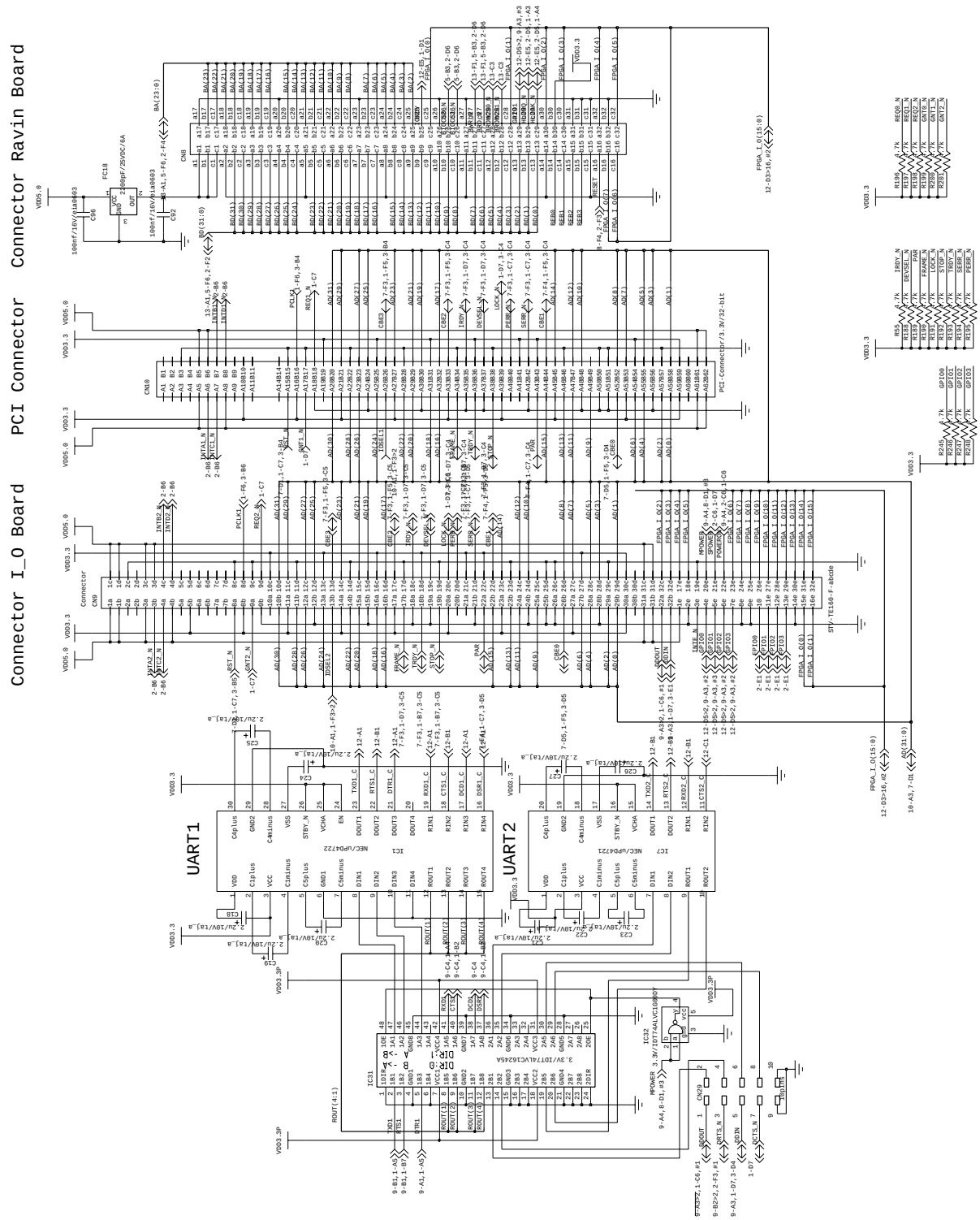


Figure A-4: SDRAM Circuit Diagram (4/20)

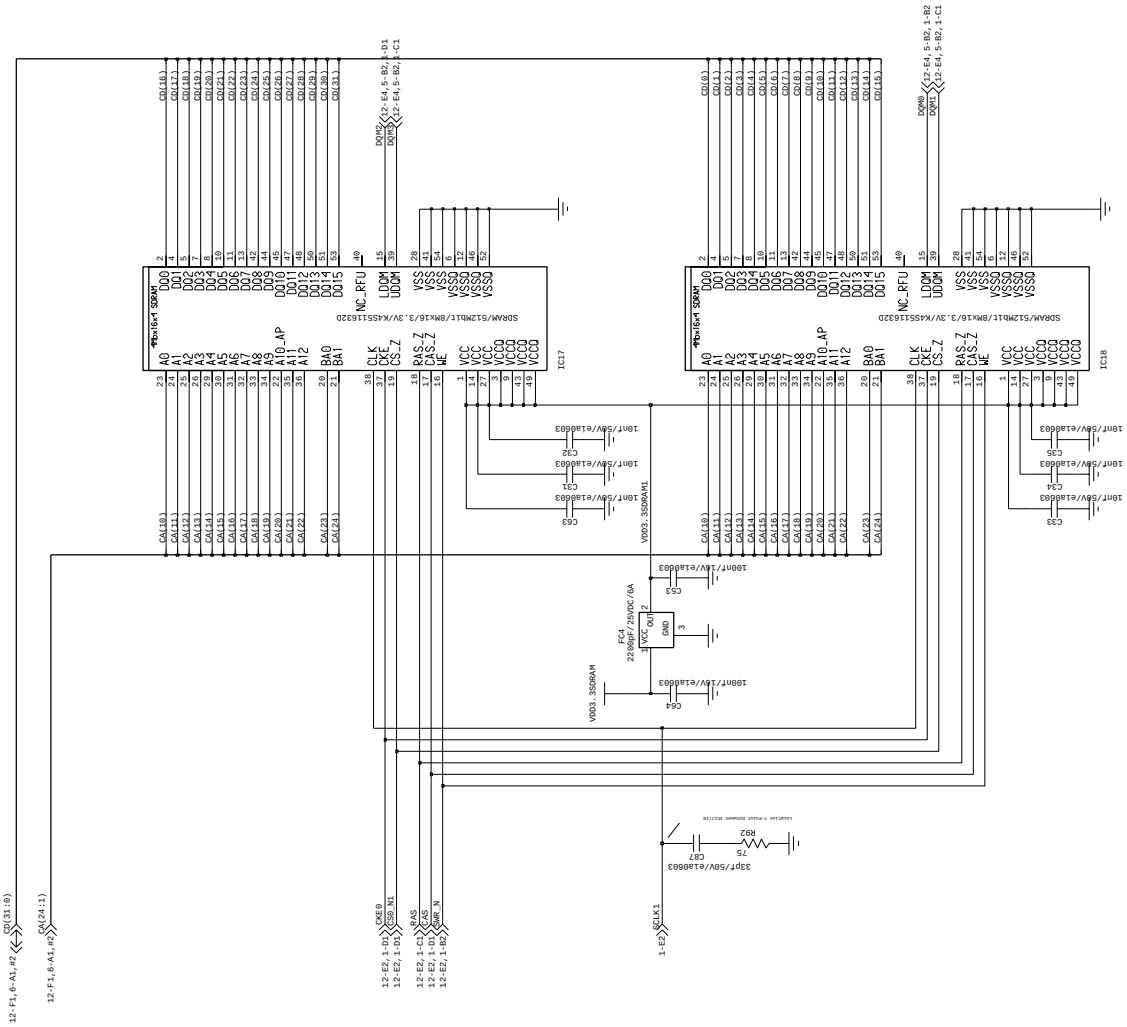


Figure A-5: Address + Data Buffer Circuit Diagram (5/20)

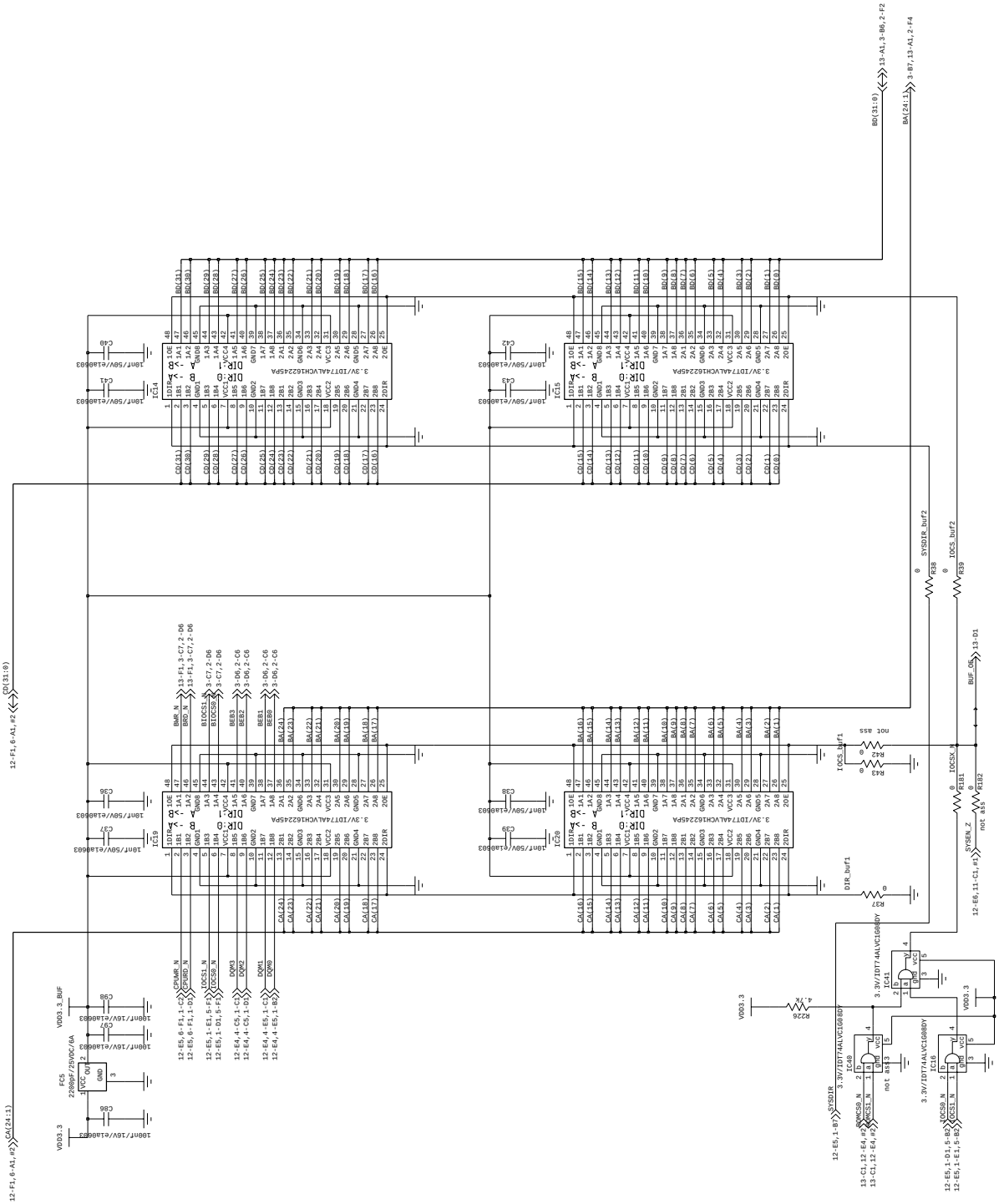


Figure A-6: FLASH ROM Circuit Diagram (6/20)

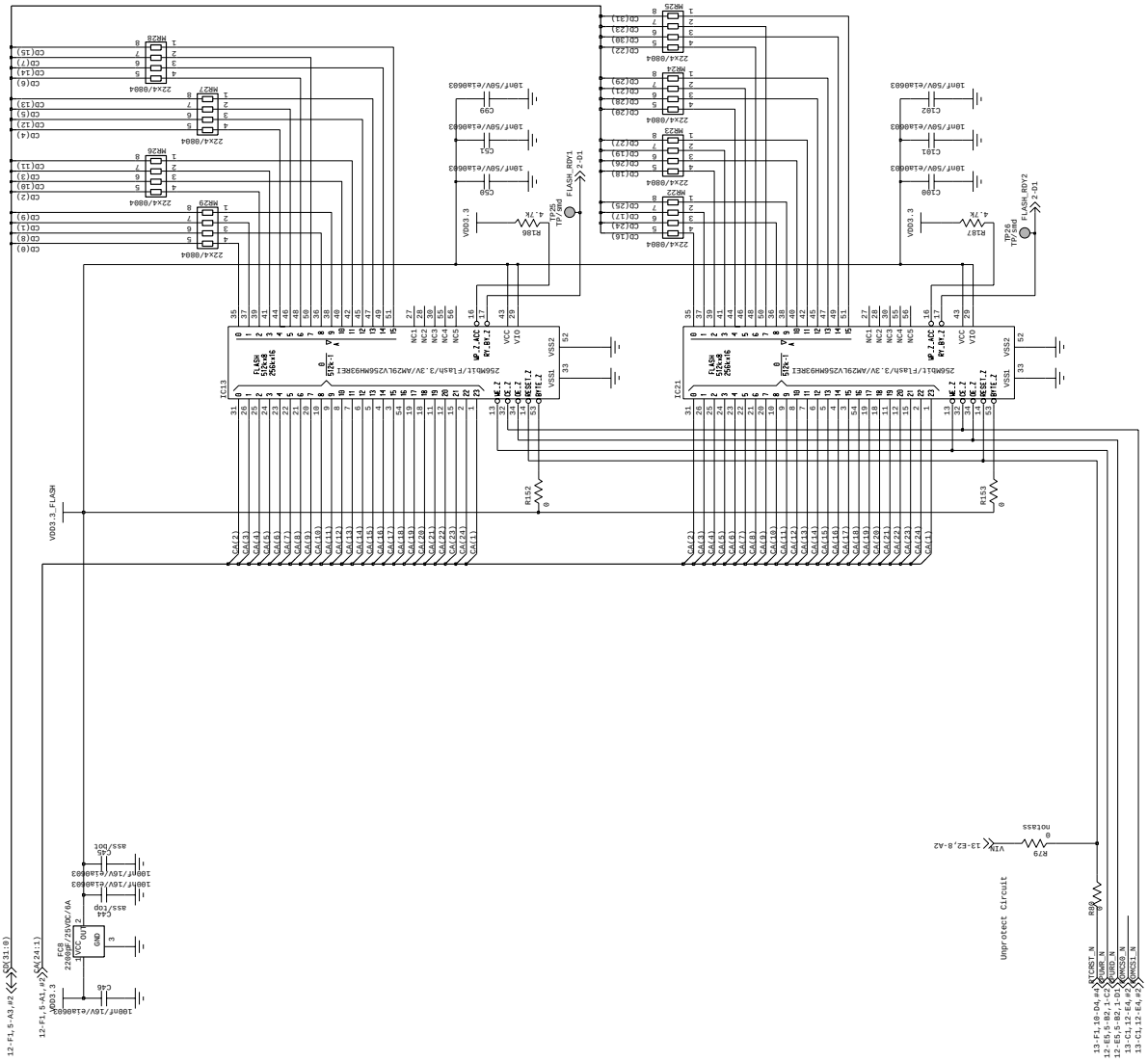


Figure A-7: Ethernet Interface (Realtek Controller) Circuit Diagram (7/20)

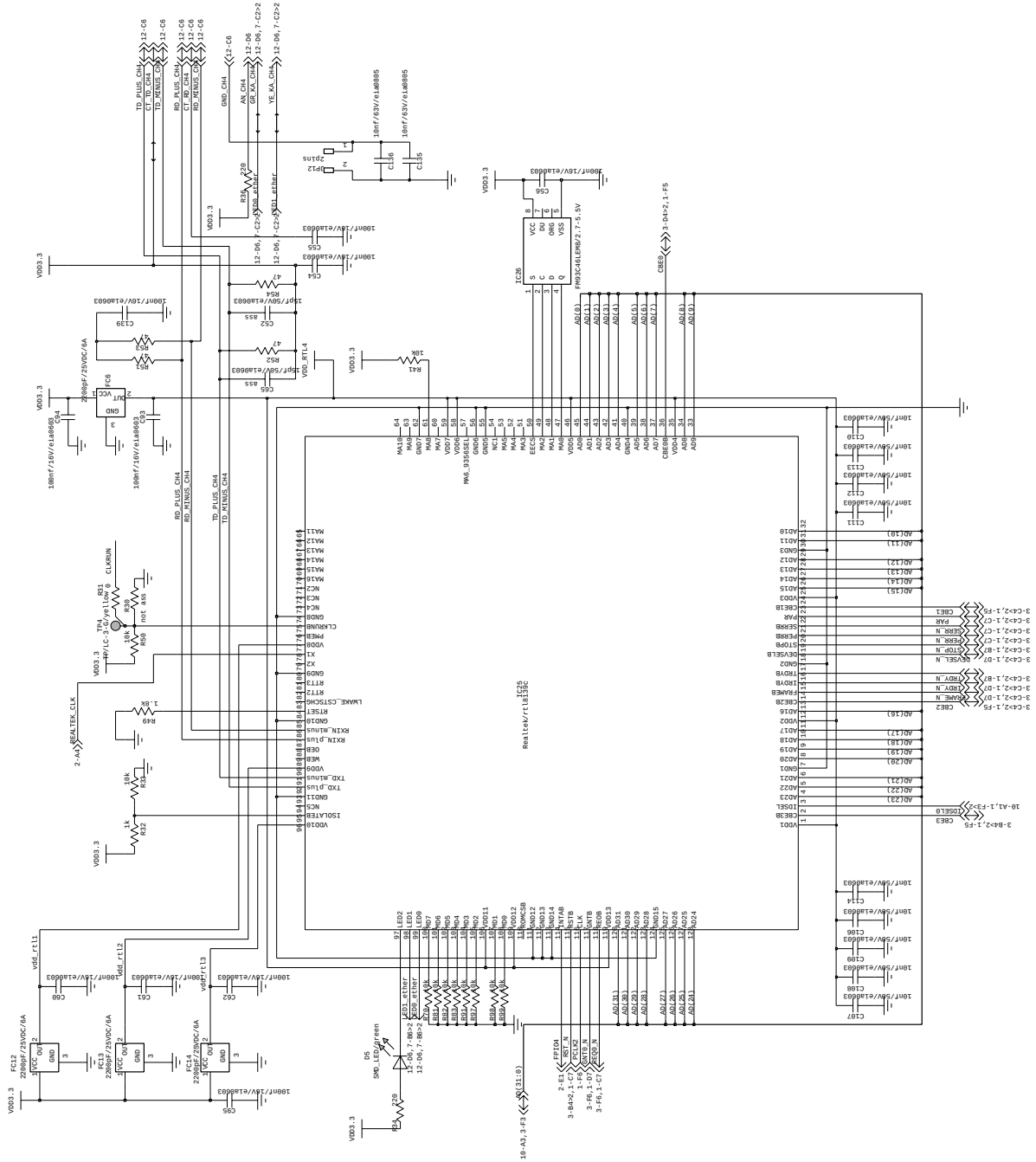


Figure A-8: Power Supply Circuit Diagram (8/20)

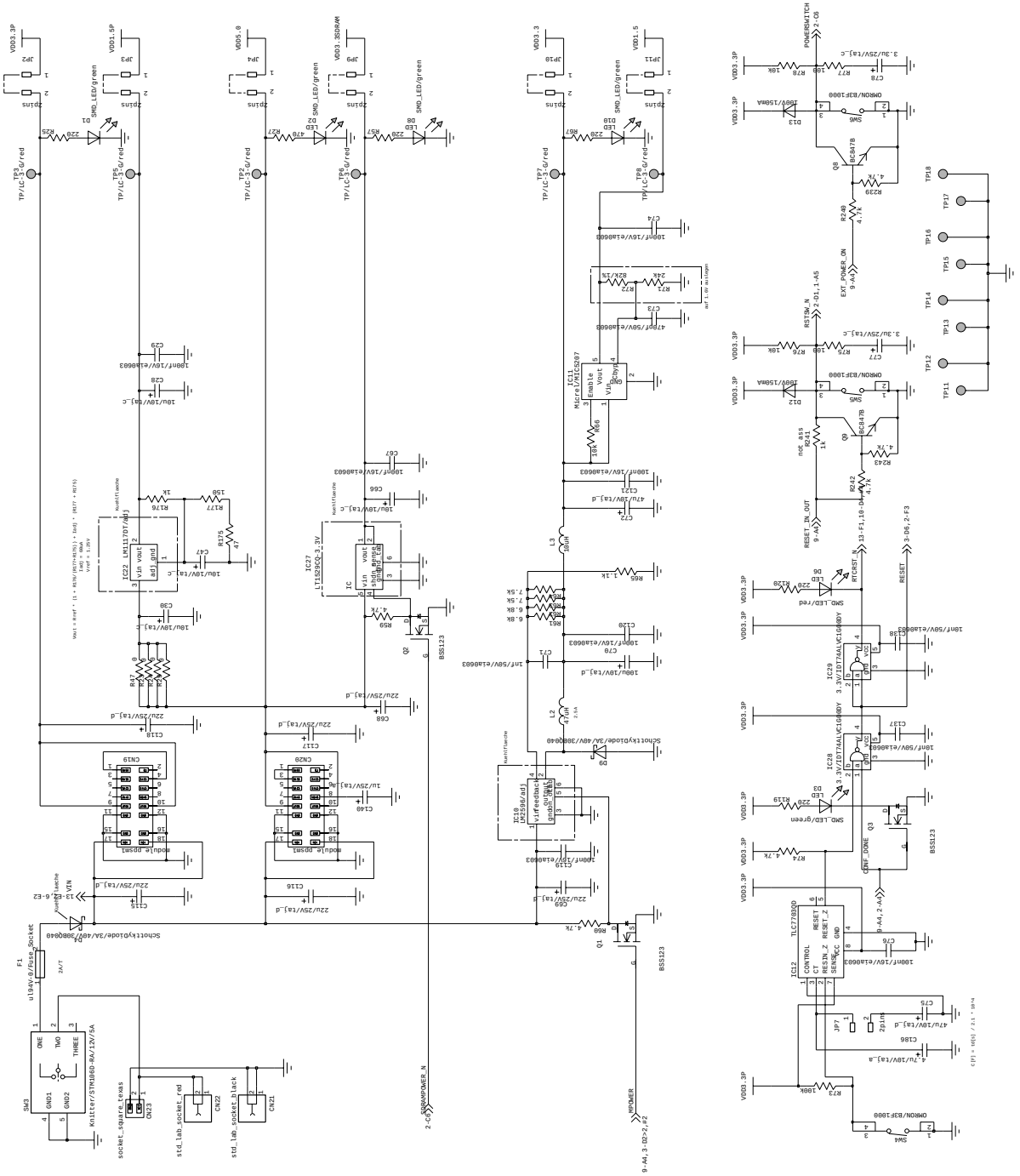


Figure A-9: Display, Setup, DIP switch, NWIRE, CSI Interface Circuit Diagram (9/20)

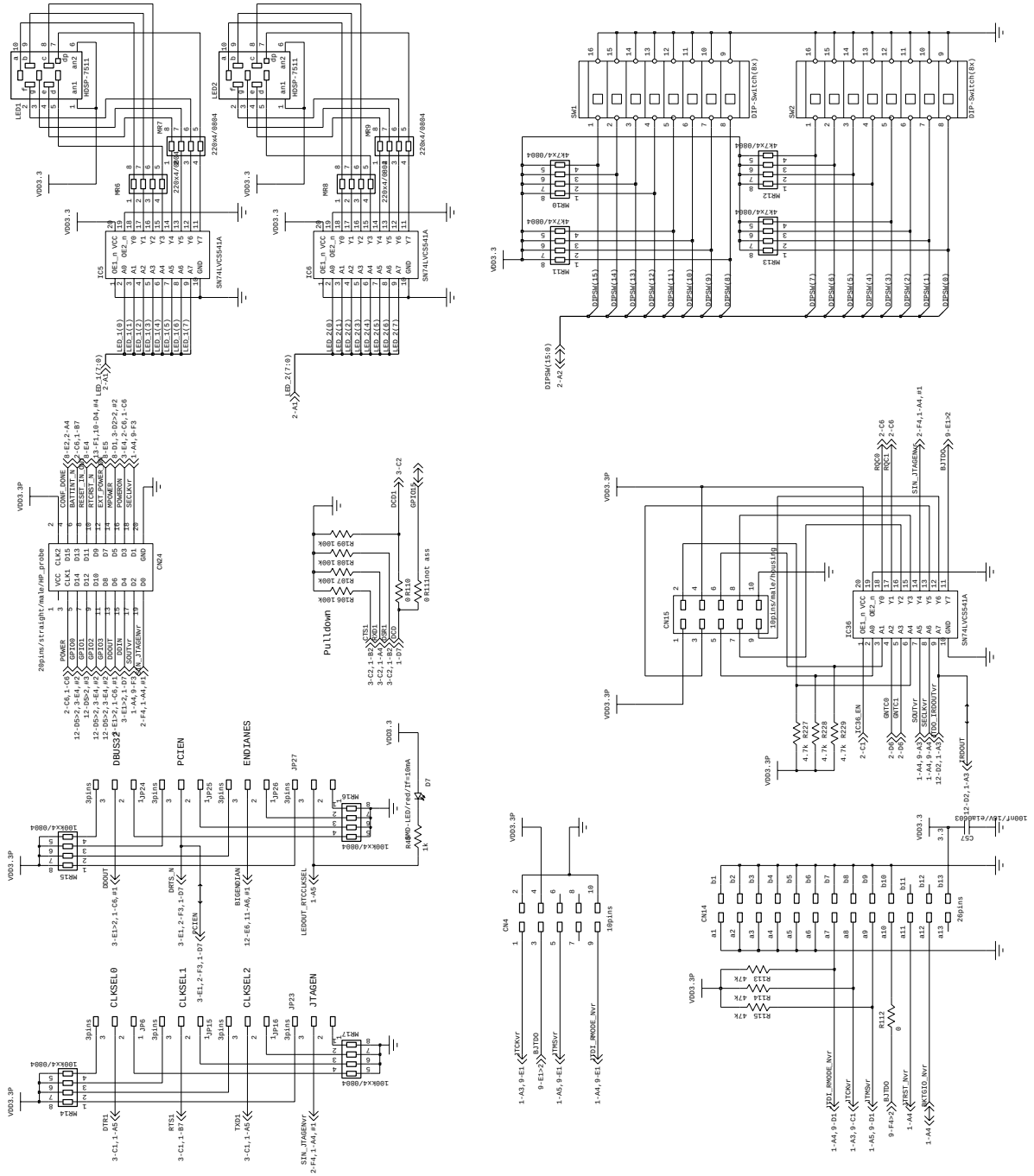


Figure A-10: Ethernet Channel 1 Circuit Diagram (10/20)

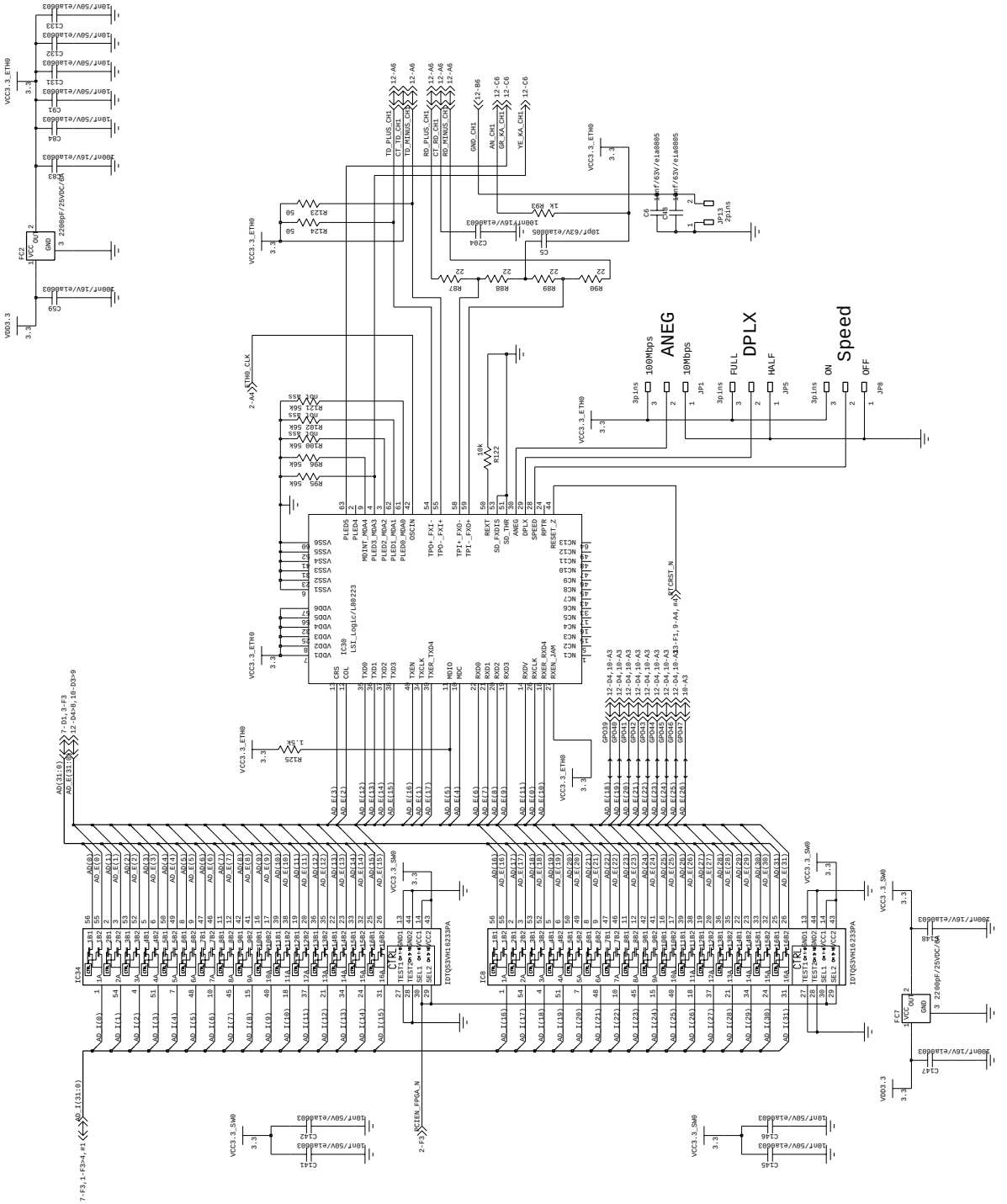


Figure A-11: Ethernet Channels 2 + 3 Circuit Diagram (11/20)

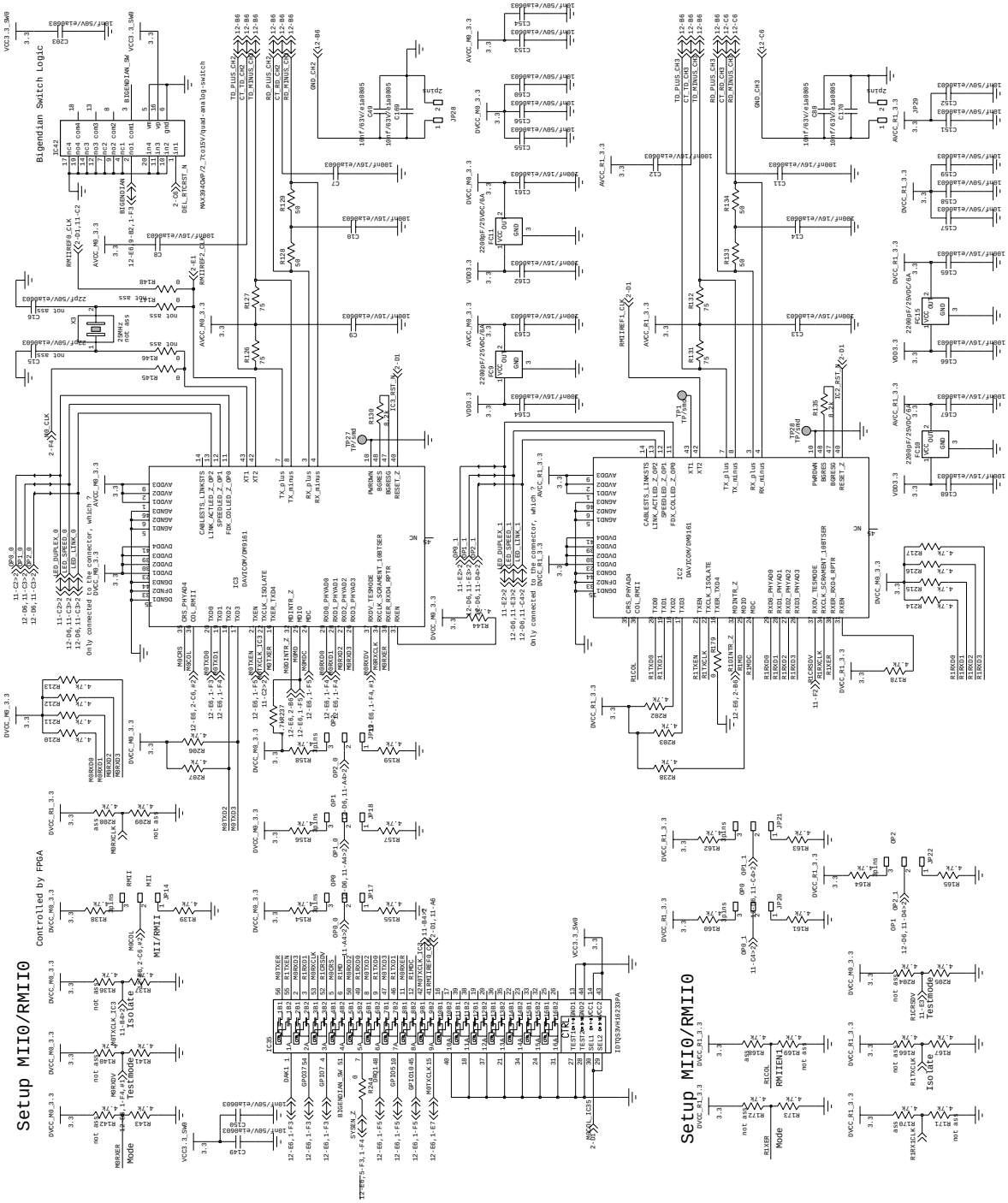


Figure A-12: Connectors Circuit Diagram (12/20)

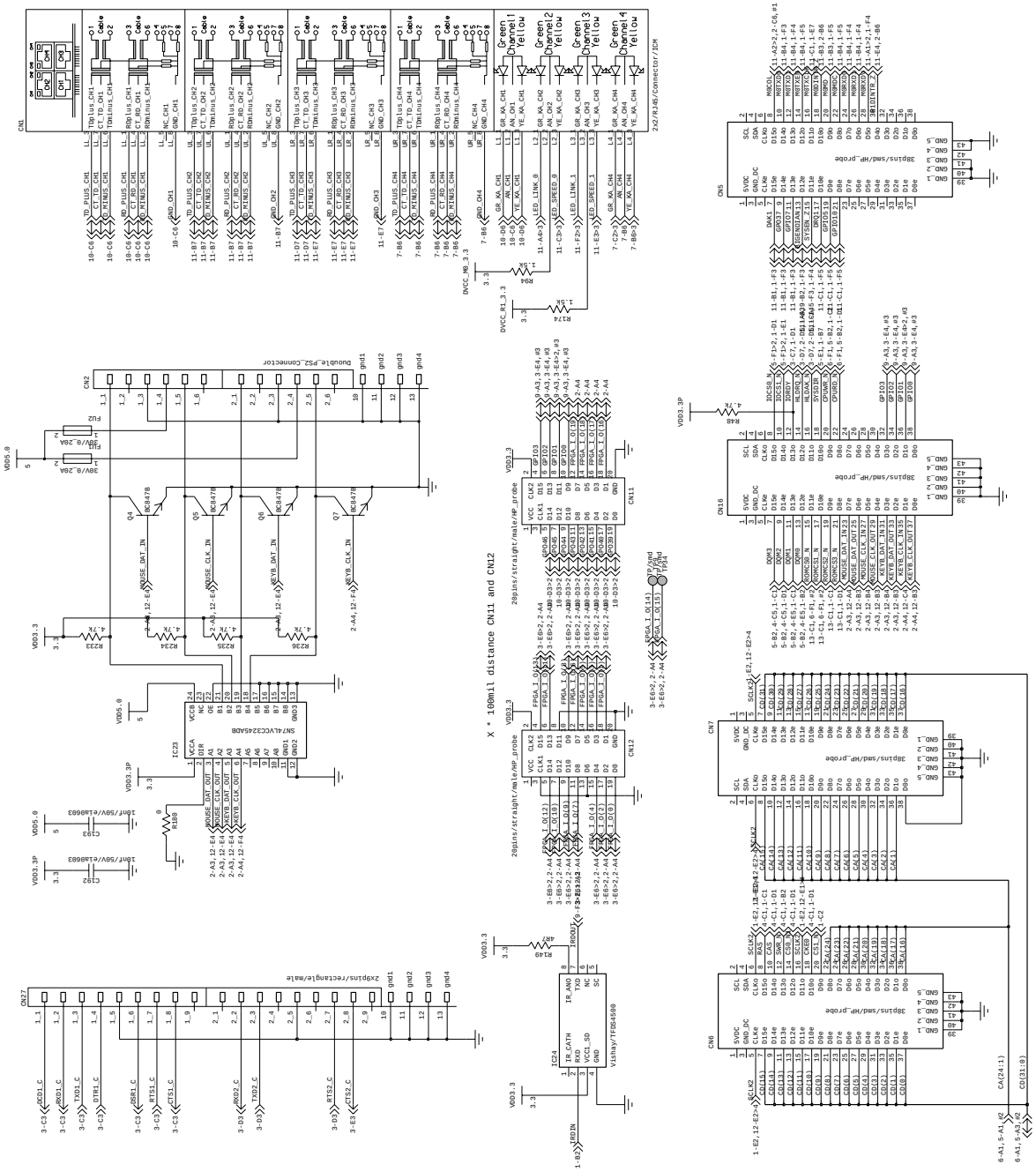


Figure A-13: Buffered FLASH ROM (not assembled) Circuit Diagram (13/20)

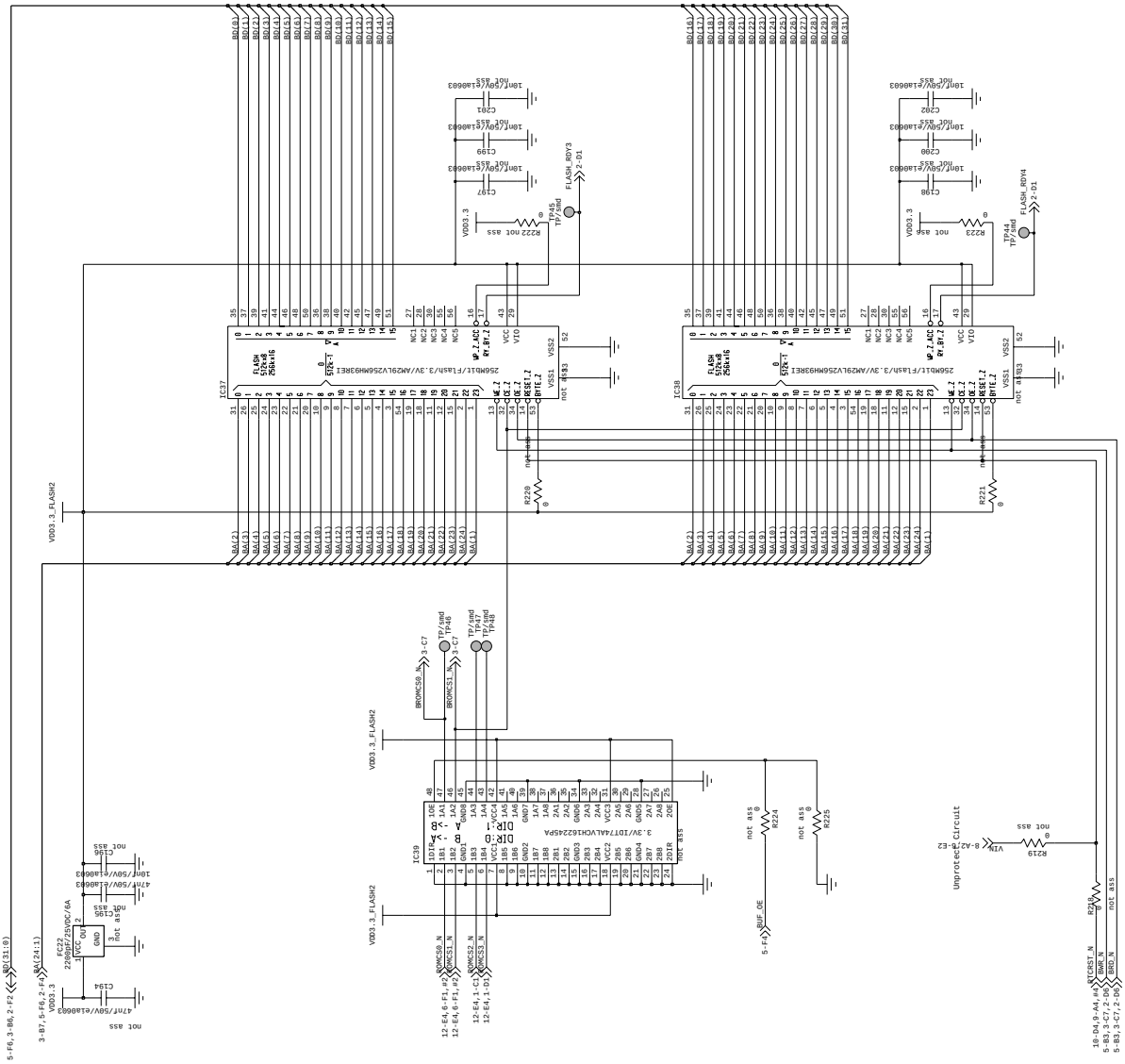


Figure A-14: XREF 1 Circuit Diagrams

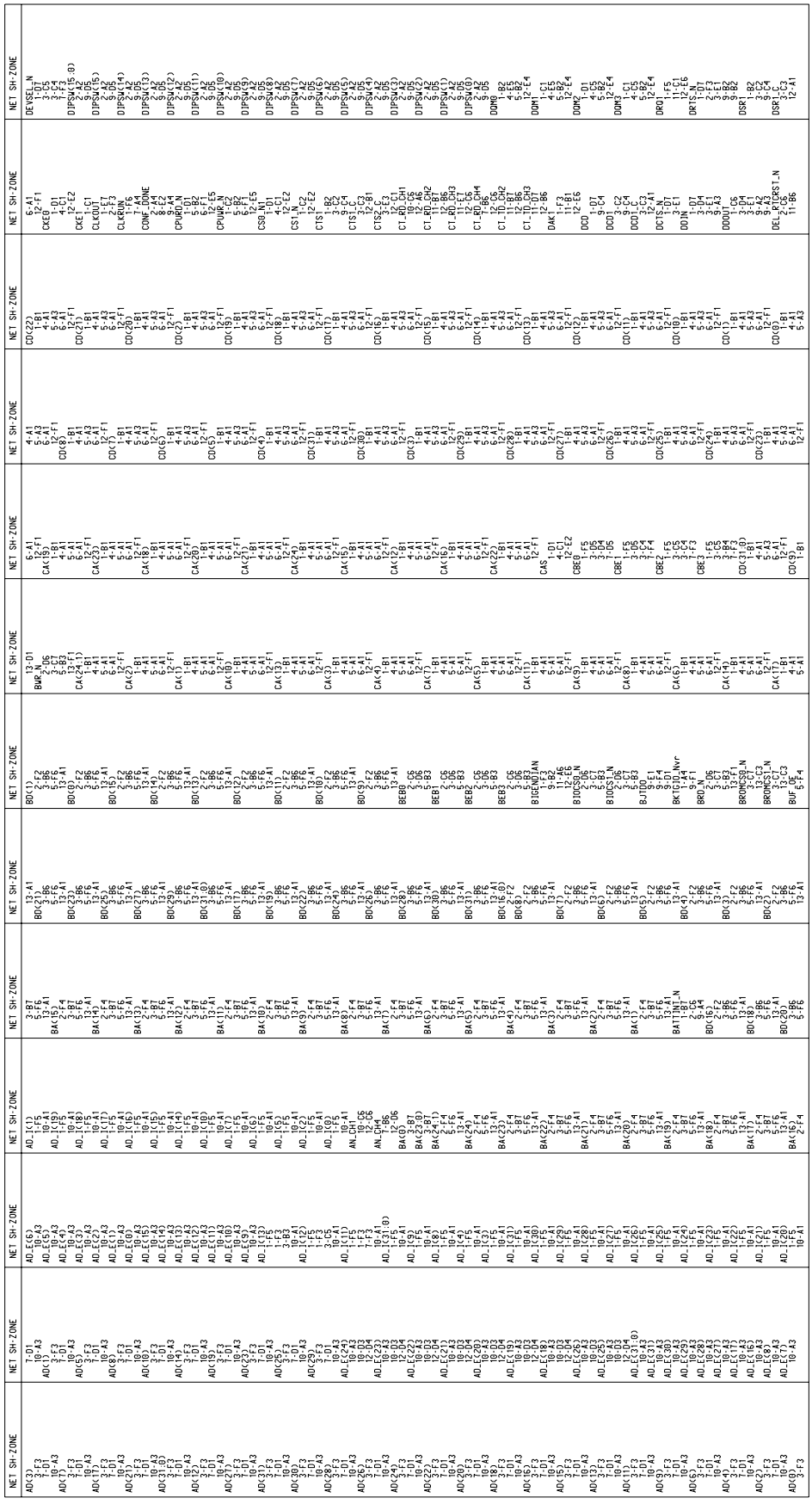


Figure A-15: XREF 2 Circuit Diagrams

Figure A-16: XREF 3 Circuit Diagrams

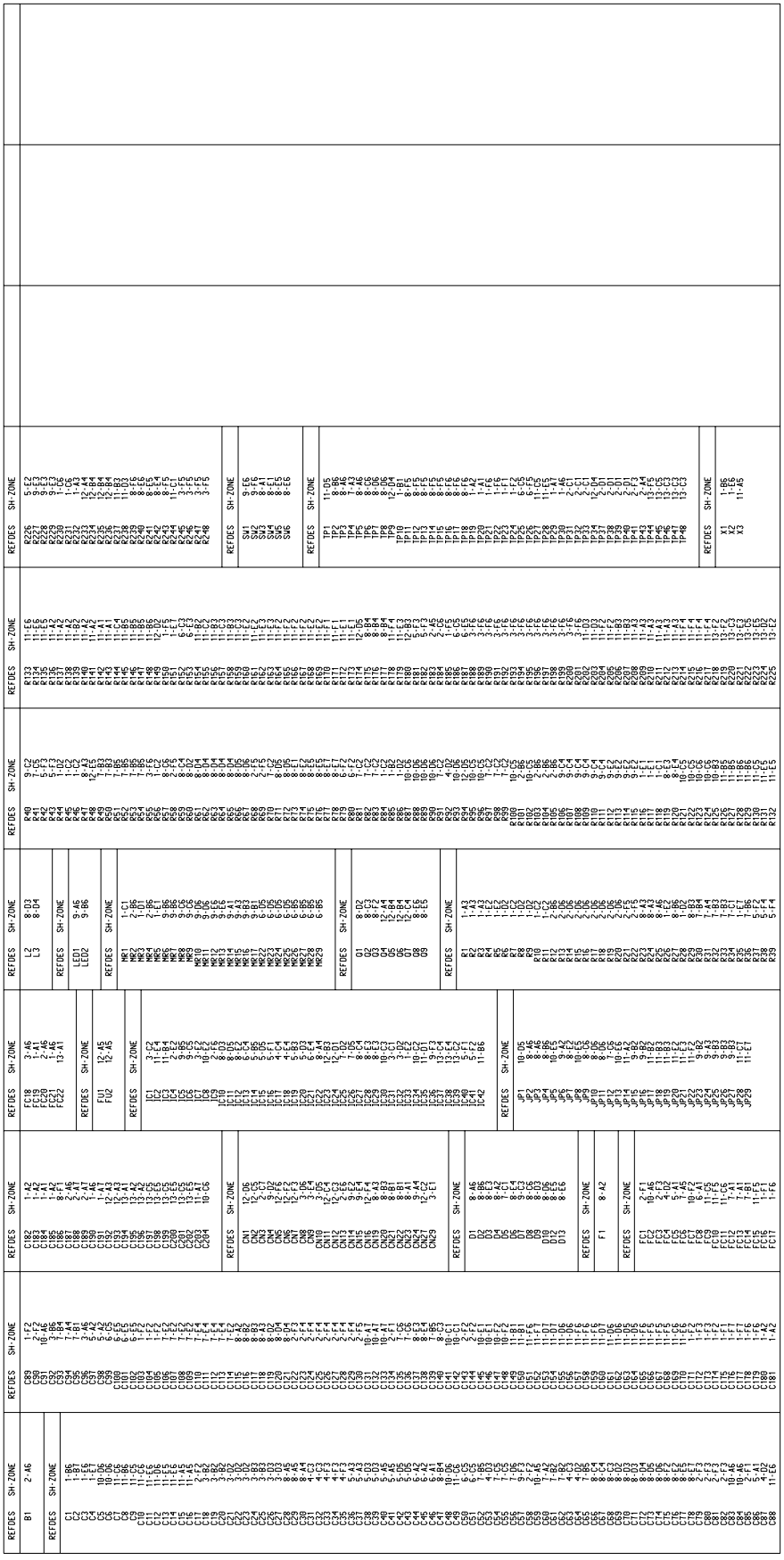


Figure A-17: Silk Screen Top

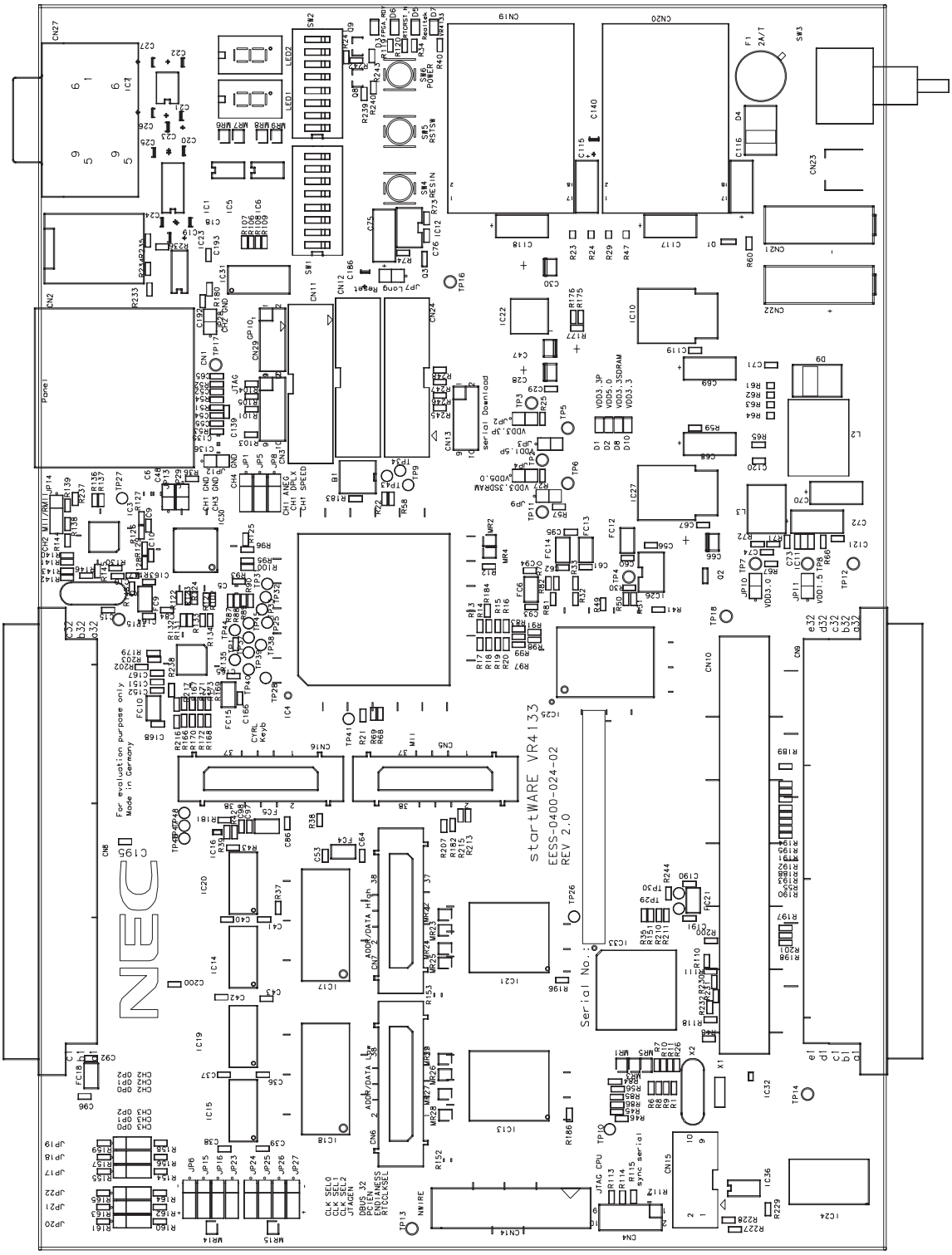


Figure A-18: Silk Screen Bottom

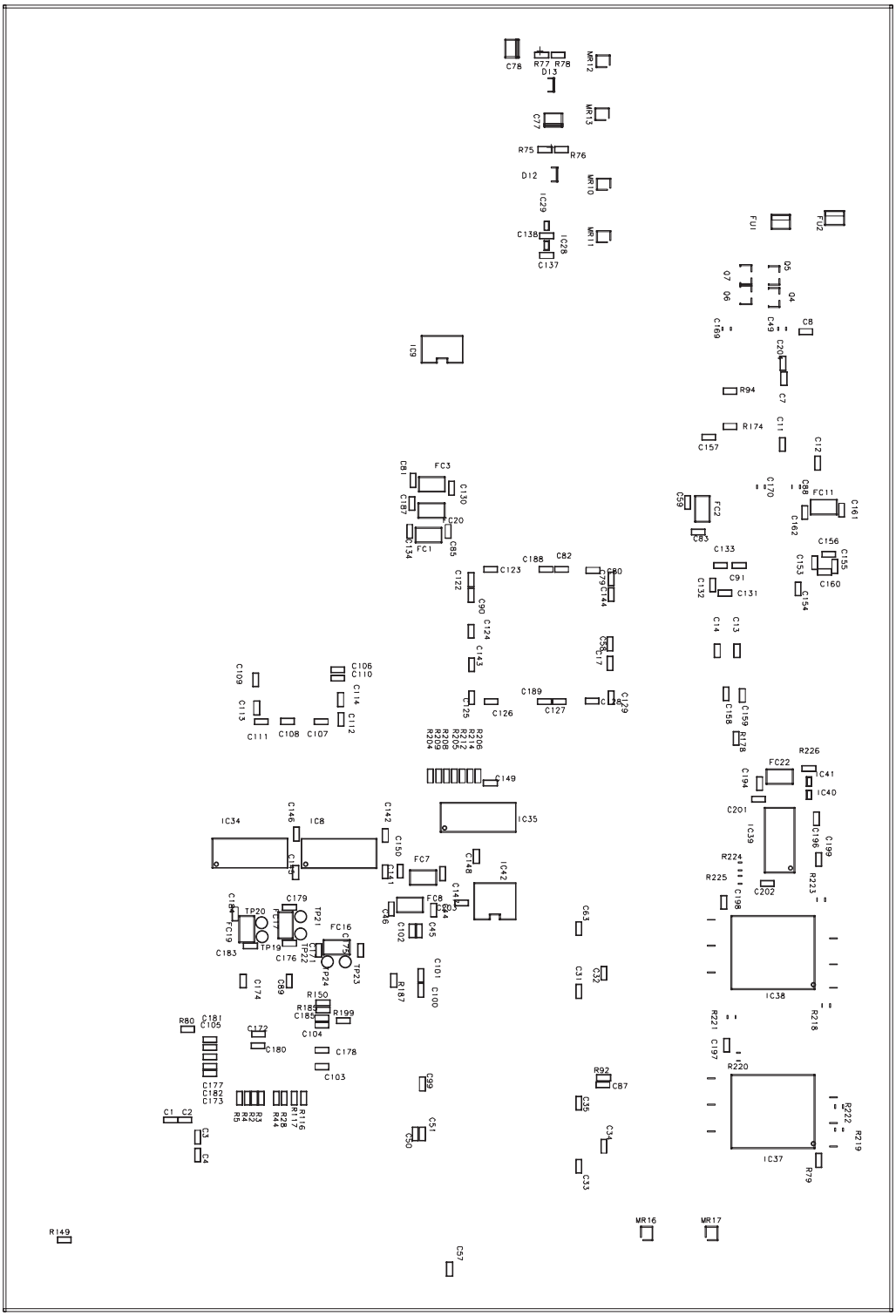


Figure A-19: Placement Top

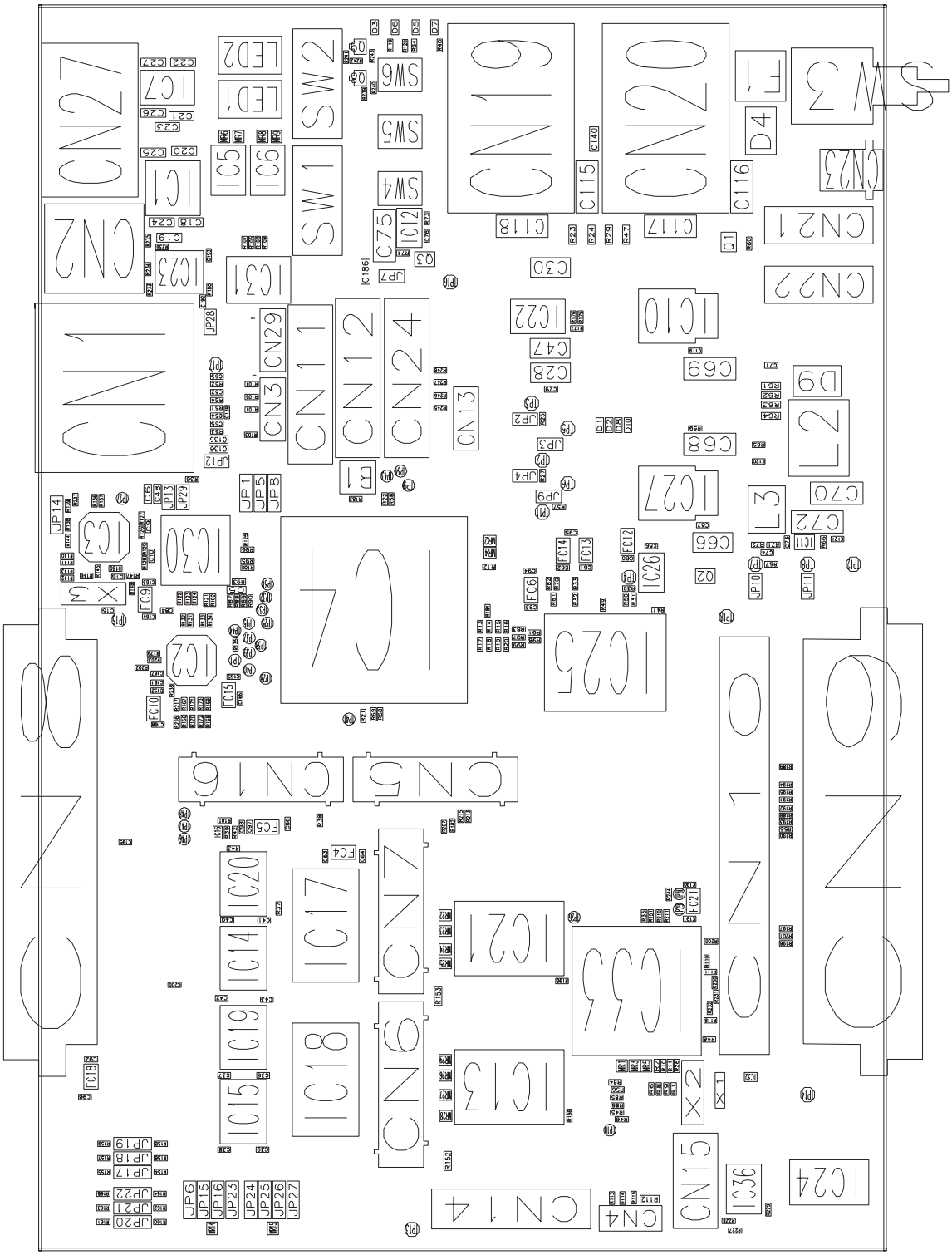
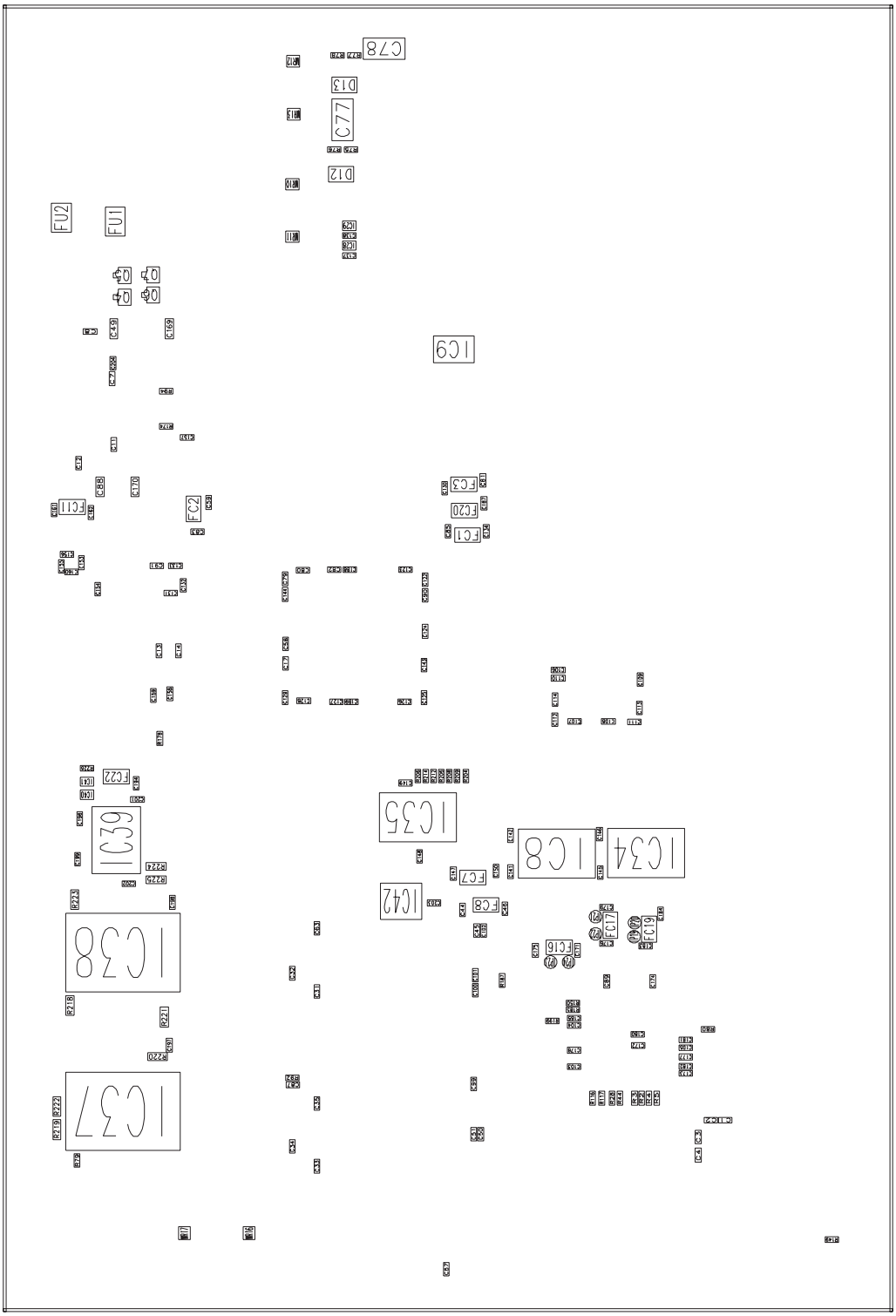


Figure A-20: Placement Bottom



Appendix B Revision History

Item	Date Published	Document No.	Comment
1	November 2003	U16916EE1V0UM00	1st release
2	March 2004	U16916EE2V0UM00	New FPGA contents (Rev. 3x) New functionality of PS/2 Interface Ethernet link status interrupts implemented Description - MAC addresses - added

[MEMO]

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics America Inc.
Corporate Communications Dept.
Fax: 1-800-729-9288
1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Market Communication Dept.
Fax: +49(0)-211-6503-1344

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: 02-528-4411

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-6250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: +81- 44-435-9608

I would like to report the following error/make the following suggestion:

Document title: _____

Document number: _____ Page number: _____

_____

_____

_____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	☐	☐	☐	☐
Technical Accuracy	☐	☐	☐	☐
Organization	☐	☐	☐	☐