ETH Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Samuel Nobs

Prototype Board for Reconfigurable OS

Student Thesis SA-2003.22
Summer Term 2003

Tutor: Herbert Walder

Supervisor:
Prof. Dr. Lothar Thiele

4.7.2003

TIK Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

Contents

Chapter 1

Introduction

This chapter offers an overview of the background and the motivation of this semester thesis. As this thesis mainly consists of designing a printed circuit board (PCB) for the desired system, the underlying concepts and theoretical aspects are not the main key of concern, therefore the corresponding texts are kept short as these topics are dealt with in other documents in detail [1],[7].

1.1 Preliminaries

To prevent possible confusions, the following terms have to be defined:

M, k In memory context, the prefix *k* means a factor of $1024 = 2^{10}$, and *M* means a factor of $1024 \cdot 1024 = 2^{20}$. When talking about throughput and bandwidth, *k* is equivalent to a factor of 1'000, while *M* means a factor of 1'000'000.

word Throughout this text, there exist 16 bit words and 32 bit words depending on the component context.

1.2 Background

Silicon process technologies used for FPGA design have been constantly improved over years: layout densities and clock frequencies have been significantly increased and, in the meantime, reached a level where a 32 bit CPU including controllers and peripherals fits into such an FPGA without even using all available resources. For example, the Xilinx Virtex-II family is built on a 0.15 micron, 8-layer metal process with highspeed 0.12 micron transistors. The largest device in this family, the XC2V8000, offers 8 million system gates[1] in 23'296 configurable logic blocks (CLB), and the highest speed grade is suited for clock frequencies above 200 MHz.

Furthermore, current FPGA technologies allow for partial reconfiguration. This allows a device being altered in certain areas at runtime, leaving other areas untouched. So FPGAs are in a position now to be used as dynamically allocatable resources. Computationally complex hardware tasks that might be unefficiently treated by a general

[1] System gates are a combination of logic, memory, and custom circuit resources that would be utilized in a typical design. This term is used as a measure of FPGA density

core	area [CLB]
UART [4]	50
100-tap FIR filter [26], 12 bit data & coefficients	250
ADPCM [5]	250
DCT [27]	600
Triple-DES processor [26]	800
256 point complex FFT [26]	850
minimal protocol stack [4], Ethernet-MAC, IP, UDP	1050
MIDI Synthesizer [6], additive, 8 harmonics	1086
Discrete Wavelet Transform [27]	1800
LEON Sparc-V8 core, 32 bit mem I/F [28], 2Kbit I-cache, 2Kbit D-cache	2000
MPEG2 video decoder [27]	3650

Table 1-1: Area requirements for typical FPGA cores (mainly taken from [2])

purpose microprocessor can be implemented in dedicated hardware and loaded or unloaded on demand, boosting performance by orders of magnitude if the time lapse needed for (re-)configuration can be kept short. A selection of area requirements of such circuits, the so called HW-Tasks, is listed in table 1-1. When comparing these values with the FPGA densities available it comes clear that it is desirable to have several HW-Task running concurrently on an FPGA.

The special forms of resource allocation needed in the abovementioned application of FPGAs ask for a reconfigurable hardware operating system (RHWOS) providing an abstraction from the underlying technology by offering services like device drivers for I/O compontents (Ethernet, Audio), doing the bookkeeping about free user space on the FPGA and assign this space to HW-Tasks to be loaded, and managing task requests to internal (block RAM, FIFOs) and external memory. For more details about RHWOS's you might want to refer to [1].

1.3 Motivation

Configurable boards with FPGAs and CPLDs have become an important means for rapid prototyping and system development. A huge number of manufacturers (Xess, BurchED, XeSys, Memec Design, Sundance, etc.) are exploiting this market and offer a broad range of such prototype boards. None of the products found in this range fulfills the special board architecture demands of a RHWOS.

Due to the limitation of partial reconfiguration of the Xilinx FPGAs to chip scanning[2] and the desired topology of the architecture inside the FPGA (see figure 1-1), all I/O devices should be connected to an OS frame on the left and the right side of the FPGA. These OS frames will be left untouched during the reconfiguration process, so the connections to the I/O devices will persist. None of the commercially available boards respects this constraint.

To maximise the useability of such a board, all I/O devices and memory modules should be adressable independently, a requirement that is not consequently met on the

[2] Column-wise reconfiguration

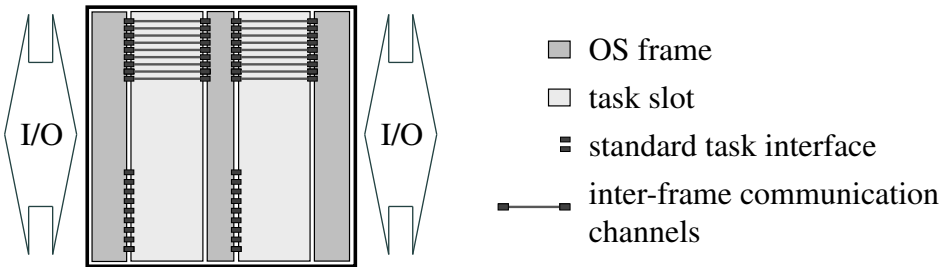


Figure 1-1: Location of FPGA I/Os suitable for RHWOS

board currently at hand, the XSV Board by XESS Corp.[8]: e.g. to use the LED bar, you have do disable the FlashRAM on the board.

The drawbacks being inherent in all these boards led to the decision to design a custom board that is tailored to the needs of a RHWOS.

Chapter 2

System Design

The design steps from the specification of the application concept and the hardware requirements of the board to the selection of design details will be rendered more precisely in this chapter. Up to that point, no schematic has been designed; please refer to chapter 3 for an explanation of the schematic.

2.1 Target Application

Mainly, this board will be used as a demonstrator for an RHWOS. The performance of such a system shall be proved by the implementation of computationally intensive and data trough-put-oriented tasks. These tasks can be found in the high-performance computing application domain:

- networking
- audio/video streaming
- multimedia
- encrypting/decrypting algorithms
- real-time signal processing

2.2 Design Requirements

The system should consist of a CPU coupled with an FPGA. The FPGA will be configured via the CPU. To allow for high-speed (partial) configuration and readback, broadband and bidirectional access to the SelectMAP configuration port (see [10] for details) is needed. The CPU needs to receive and send data (i.e. configuration bitstreams) to an external host, and some basic debugging mechanisms must be provided. Memory large enough to hold several configurations (full and partial) for the OS and for the various tasks has to be present, and this memory should be fast enough to allow for high-speed (re)configuration. For communication between the CPU and the FPGA a sufficiently high number of general purpose lines should be implemented. To allow for the target applications, I/O devices such as an audio CoDec, a video D/A-converter, an ethernet port and an additional data I/O port like RS-232 are mandatory. As the internal block

RAMs of the FPGA don't offer enough capacity for most of the applications, a huge amount of external memory is needed. For the reasons mentioned in section 1.3, all I/O devices should be connected to the left and the right side of the reconfigurable FPGA.

2.3 Design Decisions

2.3.1 FPGAs

Instead of using some standard CPU (e.g. ARM, PIC, MCore), we decided to use an FPGA and to implement a soft CPU core on it. With such soft cores, flexibility is much higher since they can be tightly adapted to our needs, and their detailed implementation may evolve during the lifetime of the XFBOARD. As soon as an update of the CPU is due, there is no need to solder out the component and replace it by an other one: the new configuration is downloaded to the FPGA, and in the next second the renewed CPU is up and running. Xilinx offers a 32 bit RISC processor suitable for our needs, the MicroBlaze soft processor core [11]. This processor core can be customized using a software tool provided by Xilinx, the Embedded Development Kit (EDK).

We decided to use the Xilinx FPGAs because Xilinx is the market leader in the sector of reconfigurable logic. Also, the Computer Engineering and Networks Laboratory has been using Xilinx products for years and therefore has some experience with those devices. Moreover, almost nothing is said about other manufacturer's FPGAs in research communities and forums.

The logic area needed by the MicroBlaze and the number of I/Os (for a detailed listing refer to section 3.1.1) used by the CPU's periphery dictate the minimum size and the package of the CPU FPGA (let's call it C-FPGA from now on): a Xilinx Virtex-II XC2V1000 in the FG456 package which contains 1M system gates (for a definition of this term refer to section 1.2) and offers 324 I/Os. Since this FPGA will behave like a general purpose processor, a rather fast speed grade is needed. Speed grades -4, -5 and -6 are available, with -6 being the fastest, supporting clock frequencies above 200 MHz[1]. We decided to use the -5 speed grade being a good compromise between speed and cost. The C-FPGA will be offered a 50 MHz clock which can be internally multiplied by an integer fraction using the FPGA's builtin digital clock managers (DCM)[10].

For the reconfigurable FPGA (R-FPGA), the number of I/Os used by the I/O-devices governs the package size: a Xilinx Virtex-II XC2V3000 in the FG676 package with 3M system gates and 484 I/O pins is needed due to the fact that only I/Os located at the package's left and right side may be used (refer to sections 1.3 and 3.2.1). As efficient and dedicated logic will be implemented on this FPGA, having a high clock frequency is not as crucial as with the C-FPGA, so the speed grade -4 has been chosen to prevent costs from exploding. The R-FPGA will have 4 clock frequencies offered by the C-FPGA and an additional 50 MHz clock from an external oscillator.

2.3.2 I/O Devices

In this subsection, some specific characteristcs of the I/O devices chosen will be denoted. For the exact device types and their manufacturers, please refer to chapter 3 and have a look at the text for the corresponding schematic.

[1] A maximum clock frequency is not that easy to declare as the longest path in the circuit highly depends on the design implementation

C-FPGA

For the communication from the external host to the C-FPGA, mainly used for the transmission of configuration data, a 100Mbps fast ethernet transceiver with an RJ45 connector has been selected.

To download the bitstream to the configuration PROM, a JTAG test access port is present. The C-FPGA is also connected to this JTAG chain to allow for an emergency configuration if all other means failed.

To communicate with the R-FPGA, a general purpose I/O (GPIO) bus is installed. This GPIO is 40 bits in width, e.g. for 32 bit data and 8 control signals. Additional 32 bits, the optional general purpose I/O (OGPIO), may be used if partial reconfiguration of the R-FPGA is not needed. Moreover, two 16 pin expansion headers are connected to the OGPIO bus that may be used even when partial reconfiguration of the R-FPGA is used.

The following debugging channels are provided: for a very basic input and output, two push-buttons and two LEDs are present on the board. For advanced debugging, two PS/2 connectors for a mouse and a keyboard and a simple 8 color VGA Output[2] are installed. For debugging using a host PC, two RS-232 ports can be accessed. An 8-LED bar monitors the level of 8 out of the 40 GPIO signals; this LED bar may be used for visual feedback. The two 16-pin headers connected to the OGPIO are available for debugging purposes, too.

Temperature sensors to monitor the core temperature of both FPGAs are installed.

R-FPGA

A 100Mbps fast ethernet transceiver with an RJ45 connector is used to send packets to the R-FPGA and to receive packets from the R-FPGA. This ethernet port is essential for all streaming and networking applications.

The R-FPGA is planned to be configured by the C-FPGA using the SelectMAP configuration method. In case of failure of this method, this task can be accomplished using the separate JTAG test access port for the R-FPGA.

For audio and signal processing applications, an audio CoDec working with CD-quality[3] audio data is installed. Two inputs and one output are available. Two inputs are needed whenever adaptive filtering of stereo signals is needed. The inputs and outputs can be used concurrently.

For video output, a VGA port driven by a video DAC is mounted to the XFBOARD. This DAC has three separate 8 bit pixel inputs, one each for red, green and blue video data. This DAC does not contain a built-in color lookup table, as this table may be implemented using the on-board memory or the SelectRAM blocks of the R-FPGA, if needed.

An RS-232 port is installed for communication with a host PC or another RS-232-aware device. This port can be used for debugging.

As mentioned above, the R-FPGA is connected with the C-FPGA by a 40 bits wide GPIO bus and a 32 bits wide OGPIO bus that may be used if the R-FPGA is not used for reconfiguration.

For a simple visual output, an 8-LED bar is connected to the GPIO signal. Furthermore, there are two push-buttons and another 2 LEDs connected for basic input and

[2] for 8 colors, the video DAC can be omitted. The red, green and blue signals may be directly fed with logic signals that have accurate levels.

[3] 44.1 kHz sampling frequency, 16 bit quantization.

output purposes.

A 36 pin expansion header is with 32 data lines, 3 power lines (+5 V, +3.3 V, +1.5 V) and a common ground can hold any extension board supporting 3.3 V LVC-MOS signal levels.

2.3.3 Memory

Memory Types

As memory requirements highly differ between the various HW-Tasks and the CPU duties, we decided to use various memory types on the XFBOARD. Furthermore, there exist a significant tradeoff between memory speed an costs per bit. In other words, the faster your memory is, the lower the memory size you get for the same amount of money. These considerations combined with some FPGA-intrinsic conditions led to the following memory selection:

- *SelectRAM blocks*. The fastest memory available on the board will be the FPGA's built-in SelectRAM blocks as the signal propagation paths are pretty short and these blocks can be clocked at the FPGA's system clock because they behave like registers. The fact that these RAMs are dual-ported is responsible for a substantial speed-up too. The number of such blocks is determined by the FPGA's type and size, the dimension of the SelectRAM blocks is not a completely free design parameter.

- *Static RAM (SRAM)*. For fast computations needing not too much memory space, fast static RAM is needed. SRAM works with a fast and simple protocol to read and write data. As a drawback, these memory modules are large and expensive compared to SDRAM.

- *Synchronous Dynamic RAM (SDRAM)*. For operations on large amounts of data with moderate speed, SDRAM is the correct type of memory. SDRAM is cheap, and they are available in higher densities than SRAM. The amount of storage bits fitting into a given package size when using SDRAM is the 16^{th} fold of the density possible with SRAM technology. The SDRAM data transfer is not as simple as with SRAM modules and needs more clock cycles for a read or write transaction. Fortunately, they support a so-called burst mode where larger amounts of contiguous data may be read or written, one word on each rising edge of the clock[4], which speeds up the data transfer.

- *FlashRAM*. To hold configuration bitstreams and application data even when no power supply is present, some non-volatile RAM modules are needed. FlashRAM is our choice because of the simple protocol used to read and write data. By mischance, access speed is very low, so these modules are even slower than the SDRAM elements. FlashRAM is rather expensive, but high densities are available.

- *Configuration PROM*. To enable automatic configuration of an FPGA on power-up, a configuration PROM that will hold the corresponding bitstream must be

[4]The modules present on the board (see section 3.2.5) allow for a burst length of at most 8 consecutive words

installed. As soon as the power supply is present, the FPGA will read the corresponding data from the PROM. The PROM must be programmable from the outside world.

For high-speed computations, ZBT-RAM has been considered which is able to follow a read cycle by a write cycle[5] without the obligatory idle cycle known from regular SRAM modules. But the speed increase is not that extraordinary to justify the significantly higher expenses yielded by ZBT-RAM.

Memory Structure C-FPGA

To load the MicroBlaze CPU soft core and a minimal bootstrap, a configuration PROM is connected to the C-FPGA. A full configuration bitstream for the XC2V1000 is 499 kB in size [14], so a module providing 512 kB is adequate. To store the microcode for the CPU, the SelectRAM blocks are extended with external SRAM memory. This memory will be used for the buffers of the ethernet protocol stack too. We decided to implement a generous 4 MB SRAM memory to allow for fairly large programs to be run on the MicroBlaze while still leaving enough free space for the ethernet protocol stack and eventually some frames for the VGA-Output.

As the VGA-output meant for debugging purposes only supports 8 colors (3 bits), we need the following amount of memory, assuming a resolution of 640 by 480 pixels:

$$640 \cdot 480 \text{ px } \cdot 3 \text{ bit} = 112.5 \text{ kB}$$

Several configuration bitstreams for the R-FPGA will be stored in the non-volatile FlashRAM, because at least the RHWOS must not be lost whenever a shut-down occurs. A full configuration bitstream for the XC2V3000 is 1.25 MB in size [14], so 16 MB of FlashRAM can hold 12 full configurations and even more partial configurations, which will be sufficient. The access times for FlashRAM are too high to enable a satisfyingly fast (re)configuration, so we decided to have the CPU load the configuration data from the FlashRAM into the SDRAM on startup, where it may be read from much faster. Alternatively, the SDRAM may be used as a video frame buffer too.

As a consequence of the MicroBlaze's 32 bit architecture, all memory modules are connected to provide 32 bit words.

memory type	C-FPGA (32 bit words)	R-FPGA (16 bit words)
SelectRAM Blocks	90 kB	216 kB
SRAM	4 MB	4 MB
SDRAM	64 MB	64 MB
FlashRAM	16 MB	16 MB
Configuration PROM	512 kB	-

Table 2-1: Memory Sizes present on the Board

Memory Structure R-FPGA

Because the R-FPGA gets configured by the C-FPGA, there is no need for memory holding configuration data at all, so there is no PROM and no FlashRAM connected

[5] and vice versa, of course

with the R-FPGA. For fast computations with moderate memory requirements, some SRAM is needed, and as a bulk memory to store larger amounts of data, SDRAM is used. To allow for as much parallelity and concurrency as possible, SRAM and SDRAM are each split in two halves to form two physically independent banks: if we had a handful of tasks one of which is time critical, one memory bank might be dedicated to that particular task while serving the other tasks with the second bank.

To calculate the proper dimension of the memory, we picked the target application consuming most of the memory: multimedia and video operations. To store one frame in the SRAM,

$$640 \cdot 480 \text{ px} \cdot 16 \text{ bit} = 600 \text{ kB}$$

are needed, assuming a resolution of 640 by 480 pixels and a 16 bit quantization of the color space[6]. Keeping this number in mind, 2 MB memory should be sufficient for each SRAM bank, providing enough space for 3 video frames at a time for fast and computation intensive video processing. As soon as it comes to buffering (frame buffering, video streaming), much more memory is needed, as the following calculation reveals:

$$640 \cdot 480 \text{ px} \cdot 16 \text{ bit} \cdot 25 \text{ fps} = 14.7 \text{ MB}$$

must be available to buffer one second of uncompressed video material, assuming a frame rate of 25 fps which corresponds to the PAL standard. As timing requirements are not that demanding for buffering purposes, buffering will be performed using SDRAM. So 32 MB per SDRAM bank, equivalent to 2 s buffering time, should be adequate.

The other half of multimedia, the audio sector, is not that challenging, as the memory space needed for one second of audio material in CD quality is only

$$44.1 \text{ kHz} \cdot 16 \text{ bit} \cdot 2 \text{ channels} = 172 \text{ kB}$$

Consistent with most signal processing applications, all memory devices are connected to work with 16 bit words. For an overview of the memory types and their dimensions refer to figure 2-1.

2.3.4 Power Supply

We decided to use standard PC power supplies with a 20-pin ATX connector for the XFBOARD. Since the XESS board uses the same supply, the it may be used for the XFBOARD. The ATX supply delivers +3.3 V and +5 V, so the only voltage that has to be generated is +1.5 V. This voltage gets derived from the +5 V present on the ATX connector. All voltages on the ATX connector, ± 12 V, ± 5 V and ± 3.3 V, are tied to an additional header which may be used to feed external devices, e.g. a hard drive.

Alternatively, we decided to include a connector for a +5 V supply, because these low voltage supplies with only one fixed voltage are much more handy than the large ATX devices. Whenever this option is chosen, the +3.3 V also have to be derived from the +5 V provided by the +5 V feed.

Please read section 5.1.13 carefully for a detailed description on how to connect the power supply as any connection error may destroy the circuit!

[6] 5 bits for the red color information, 5 bits for blue and 6 bits for green due to our eye's increased sensitivity for green light

There is an LED for each voltage on the XFBOARD which is lit when the corresponding voltage is present. For current measurements, there are fuses that may be removed to open the circuit to connect measurement gear.

Chapter 3

Schematic Entry

This chapter explains the schematic of the XFBOARD. It basically describes the schematic sheets in appendix A. The numbering of the sections is consistent with the numbering of the corresponding sheet in all digits except the first one, e.g. section 3.**1.4** delineates the sheet with the drawing number SA-2003.22-**1.4**. For rather conceptual considerations and reasons for the presence of particular I/O components refer to section 2; especially section 2.3 might serve your needs .

3.1 XF Top

3.1.1 CPU FPGA

The FPGA used as the boards CPU and configuration mechanism for the board's reconfigurable part is an XC2V500 of the Virtex-II series by Xilinx [9]. As this device has a huge number of I/O's, decoupling of the V_{CC} pins is crucial. Following Xilinx's application note on power distribution systems design [12], the number of decoupling capacitors needed highly depends on the application and the number of I/O's used simultaneosly. Due to the fact that the device is reconfigurable, the I/O usage is hardly predictable. To play it safe, I decided to assume full I/O usage, i.e. all I/O's available are used. This indeed overly pessimistic assumption results in the need for approximately one capacitor per V_{CC} pin. Given the number of discrete capacitors needed, a distribution of capacitor values adding up to that total number must be determined. Fortunately [12] does give me a hand: 50% of the total number of capacitors needed should be covered by 0.001 μF capacitors, and this number should be divided by two for every decade of increase in size up to 100 μF. This calculation yields the distribution shown in table 3-1. As my assumption concerning I/O usage is rather pessimistic I took the liberty of rounding off any noninteger result.

The FPGA is connected to be configured in Master Serial mode as described in the Virtex-II user guide [10]. Configuration data is read from an XC18V04 configuration PROM by Xilinx [14] which is connected to the same JTAG chain as the FPGA, being the first element in the chain and the FPGA the second element. A reset facility is provided by the program reset button, erasing the FPGA and initiating the configuration process. DONE and $\overline{\text{INIT}}$ are pulled to V_{CC} to guarantee defined signal levels at any time. To monitor the status of DONE , an LED is installed.

As there is no need for runtime reconfiguration of this FPGA, all I/O pins except

Capacitance value	$V_{CC_{int}}$ pins	V_{CC_o} pins
10.0 μF	0	1
1.0 μF	1	2
0.1 μF	3	6
0.01 μF	5	10
0.001 μF	11	20

Table 3-1: Number of decoupling capacitances used for the C-FPGA

Device	Pin Count
VGA Port	5
Switches	2
Control LEDs	2
Ethernet Tx/Rx	32
RS-232 Double Port	4
2x PS/2 Port	4
SRAM	57
SDRAM	54
FlashRAM	59
Clock	1
Clocks for R-FPGA	4
Configuration	2
SelectMAP connection to R-FPGA	14
General Purpose I/O	40
Optional General Purpose I/O	32
Temperature Sensors	2
Pins used	314
Pins available	324
Pins left unused	10

Table 3-2: Pin accounting for the C-FPGA

for those dedicated to the Master Serial configuration mode can be used. A summary of the pin usage is given in table 3-2.

To allow simple optical signalling and monitoring, active low LEDs are connected to two I/O's. The 50 MHz oscillator providing the clock for the two FPGAs is also included in this schematic.

The clock signal coming from the quarz oscillator is AC terminated as recommended in [25]. AC termination adds a capacitive load to the driver and a delay due to the RC time constant, however, it consumes low power. The termination elements have to be placed as near as possible to the FPGA's clock input pin in the layout.

3.1.2 Ethernet

The XFBOARD's Ethernet Tx/Rx Block has been built around the LXT970A Ethernet PHY by Intel [16] which supports full-duplex operation at 10 and 100 Mbps. The

fact that we decided to use the 0810-1XX1-03 Integrated Connector Module by Bel Fuse Inc.[15] made my life much more easier as this module includes the magnetics needed for operation. Additionally, it features two LEDs for signalling purposes. Some concepts have been adopted from the XSV manual [8].

Because the driver level of the PHY's Media Independent Interface (MII) is reduced, no termination resistances are needed at its outputs. The inputs for the fiber transceiver are left unconnected as we don't use the fiber interface.

The yellow LED of the connector module is connected to the active low pin LEDL of the PHY to indicate Link Valid status during 10 Mbps operation, and scrambler lock and receipt of valid idle codes during 100 Mbps operation, respectively. The two-color LED is connected to the active low pins LEDT and LEDR to indicate receiver activity with green light and transmitter activity with orange light.

The decoupling of the supply pins has been implemented as suggested in Figure 21 (Typical Interface Circuitry) in [16]. For debugging purposes, test pins have been provided at the quarz inputs of the Ethernet PHY, IX and OX .

3.1.3 1M x 32 SRAM

We decided to use the AS7C34096 by Alliance Semiconductor [20] for the SRAM block. As this memory module offers 512k halfwords (512k x 8 bits), I needed to merge the address space of two such blocks to get the desired 1M addresses. To get 32 bit words instead I had to connect two such pairs in parallel and concatenate their 8 bit data ports to the 32 bit wide signal DATA<31..0> .

To merge the address space of two such blocks, two alternatives have been considered:

- Extend the address width from 19 to 20 bits and use the MSB as a chip enable signal for the modules, inverted for the first module and non-inverted for the second module.

  While it would be nice to have a block looking like a 1M x 32 bits block from the outside, the drawback of this approach is the glue logic needed: the inverter is likely to introduce some unwanted delay in the order of 2-5 ns. As the memory modules used offer cycle times of 15 ns, this approach is not an option.

- Preserve the address width as it is and use two additional signals to select and deselect the appropriate memory module.

  Using this approach I don't need any glue logic, but there is now the possibility to have both memory modules enabled by mistake, resulting in bus contention in read mode. To avoid potential damage, the memory manager implemented in the FPGA has to guarantee never to activate both memory modules at a time.

Following this second approach, the memory block now has the address inputs ADDR<18..0> for the first 19 address bits and ADDR_19 and $\overline{\text{ADDR\_19}}$ for the MSB. For this purpose, there is no need to drive $\overline{\text{CE}}$ and $\overline{\text{OE}}$ with different signals, so these two inputs are shorted.

3.1.4 16M x 32 SDRAM

To get 32 bit wide memory entries, two HYB39S256160CT by Infineon Technologies [21] had to be connected in parallel because these devices offer 16M x 16 bit. $\overline{\text{CS}}$ is tied to GND and CKE to V_{CC} to permanently enable the memory modules.

Pin Number	Function
1	Data
2	Not connected
3	GND
4	VCC
5	Clock
6	Not connected

Table 3-3: Pin assignment for the 6 pin Mini-DIN connector

3.1.5 4M x 32 FlashRAM

The FlashRAM used in this circuit consists of Intel's 28F640J3 StrataFlash devices [24] which is organized as 16M x 8 bit or 8M x 16 bit. To get 8M x 32 bit, 2 such devices are connected in parallel, and the $\overline{\texttt{BYTE}}$ pins are tied to +3.3 V to enable 8M x 16 bit operation. In 8M x 16 bit operation, A<0> is ignored, so A<23..1> is connected to ADDR<22..0>.

CE0, CE1 and CE2 are tied to GND and $\overline{\texttt{RP}}$ is connected to +3.3 V because there is no need to deactivate the device. Also, $\overline{\texttt{OE}}$ is connected to GND because we don't want to deactivate the outputs as there is no other device that might want to drive these signals.

The status signal STS is an open drain output and is high impedance when the device is ready, so the two pins can be shorted and pulled up using a resistor. So STS is high only when both devices are ready.

3.1.6 Keyboard and Mouse

For the PS/2 ports, no additional components are needed except for two pullup resistors as the DATA and CLK signals are connected to an open drain output. For the connector pin assignment refer to table 3-3.

3.1.7 Simple VGA Output

A very simple 8 color VGA signal can be generated without even the use of a video DAC by connecting three digital 1 bit signals directly to the connector. Refer to table 3-4 for information on connector pin assignment.

3.1.8 RS-232 Double Port

The transmitting and receiving pins TXDI and RXDO are brought to the correct signal level using the Maxim MAX233A [17] and connected to RD and TD at the DB9 connector (for pin assignment refer to table 3-5).

While the normal PC hardware might well run with just TD, RD and GND connected, most driver software will wait forever for one of the handshaking lines to go to the corrrect level. Depending on the signal state it might sometimes work, other times it might not. The reliable solution is to loop back the handshake lines if they are not used. When the lines are handshake looped, the RTS output from the PC immediately activates the CTS input - so the PC effectively controls its own handshaking. For this reason I connected RTS with CTS and DTR with DSR and DCD.

Pin Number	Function
1	Red
2	Green
3	Blue
4	Monitor ID bit 2
5	GND
6	GND
7	GND
8	GND
9	Not connected
10	GND
11	Monitor ID bit 0
12	Monitor ID bit 1
13	Horizontal sync
14	Vertical sync
15	Not connected

Table 3-4: Pin assignment for the VGA DB15 high density connector

Pin Number	Short Name	Function
1	DCD	Data Carrier Detect
2	RD	Received Data
3	TD	Transmit Data
4	DTR	Data Terminal Ready
5	GND	Signal Ground
6	DSR	Data Set Ready
7	RTS	Request To Send
8	CTS	Clear To Send
9	RI	Ring Indicator

Table 3-5: Pin assignment for the RS-232 DB9 connector

Often it is desirable to interchange the RD and TD lines of the connector. Two jumpers have been provided for that purpose.

3.1.9 I/O Header

Two 16 pin headers are connected to the optional general purpose I/O lines which may be used to extend the capabilities of the board

3.1.10 Switches

Two push buttons are present to provide means for simple user input. As there are no external pull-up resistors installed, they must be instantiated in the FPGA.

3.1.11 JTAG Header

To allow configuration in Boundary Scan mode, a JTAG header is present.

3.2 XF Bottom

3.2.1 Reconfigurable FPGA

The FPGA used as the reconfigurable unit is an XC2V3000 of the Virtex-II series by Xilinx [9]. Refer to section 3.1.1 for details about the decoupling capacitor problem; a listing of the number of capacitors used for decoupling is given in table 3-1.

The FPGA is connected to be configured in Slave SelectMap mode (for information about this configuration mode please refer to [10], page 256).

Capacitance value		$V_{CC_{int}}$ pins	V_{CC_o} pins
10.0	μF	0	1
1.0	μF	1	3
0.1	μF	3	8
0.01	μF	5	14
0.001	μF	11	28

Table 3-6: Number of decoupling capacitances used for the R-FPGA

When connecting peripheral devices to the FPGA, lots of I/O pins must be left unconnected due to the need for reconfiguration. The OS frames in the FPGA remain static, so the I/O pins located in such frames may be used for peripheral devices only. We decided to use only the Inputs and Outputs neighbouring the left and the right OS frame (see figure 1-1). As a result of this restriction there are mere 150 pins available for peripheral devices on each side of the FPGA. An overview of the number of pins used for the various devices connected to the FPGA is given in table 3-7. To retain flexibility, some additional 32 lines to the C-FPGA connecting from the top side of the R-FPGA have been included for the case of non-reconfigurable use.

To allow simple optical signalling and monitoring, active low LEDs are connected to two I/O's. The state of the FPGA's DONE configuration signal is monitored by a LED too. The configuration signals $\overline{\text{PROG}}$ and CCLK are connected to the C-FPGA

Device	Pin Count	
	left	right
SRAM	38	38
SDRAM	37	37
VGA Port	27	-
Ethernet	-	32
RS-232 Single Port	-	2
Switches	-	2
Control LEDs	-	2
Audio CoDec	-	11
General Purpose I/O	16	24
I/O Slot	32	-
Pins used	150	148
Pins available	150	150
Pins left unused	0	2

Table 3-7: Pin accounting for the R-FPGA

and would not assume a valid signal level as long as the C-FPGA is not configured, so they need to be pulled to V_{CC}.

To provide temperature control of the two FPGA's, two MAX1617 temperature sensors by Maxim [13] are installed which are connected to the FPGA's temperature-sensing diode pins, `DXN` and `DXP` . The temperature values can be read using the SMBus serial protocol described in [13], and the addresses of the two sensors are set to `1A` for the C-FPGA and to `4C` for the R-FPGA.

The clock signal coming from the quarz oscillator is AC terminated as recommended in [25]. AC termination adds a capacitive load to the driver and a delay due to the RC time constant, however, it consumes low power. The termination elements have to be placed as near as possible to the FPGA's clock input pin in the layout. The clock signals coming from the C-FPGA are not terminated as these signals are buffered and are of high quality.

3.2.2 Audio CoDec

The Audio CoDec used in this circuit is the AK4563A by AKM [18]. It is a 16 bit CoDec with two analog stereo inputs and one analog stereo output. The schematic supplied in the datasheet for this device (figure 23 in [18]) has been adopted without modifications. The value for the AC coupling capacitors at the inputs has been calculated to let the cutoff frequency f_c be at most 16 Hz as this is the lower end of the human ear's frequency response. The formula used was $f_c = 1/2\pi RC$ with R being the CoDec's input resistance[1] and C being the desired capacitance. This calculation yields a value of 1 μF for C. Using the same capacitance for AC decoupling at the output guarantees f_c to be at most 16 Hz for devices with an input resistance greater than or equal to 1 kΩ. This means that for headphone appearing as a 32 Ω load at the output will raise f_c to a significantly higher value (approximately 500 Ω). Hence, the

[1] this value depends on the gain table chosen. R is 10 kΩ at microphone gain table and 125 kΩ at line gain table, see [18]

Pin Number	Function
1	Ground
2	Left channel
3	Not connected
4	Not connected
5	Right channel

Table 3-8: Pin assignment for the audio connector

use of a line amplifier is recommended.

3.2.3 Video DAC, VGA Out

To convert the digital, 24bit wide color data for a pixel into analog values for the red, green and blue components of the video signal, Intersil's video DAC HI1178 is [19] used. Two modifications to the application circuit in figure 11 in [19] have been performed:

- to get the +2 V needed for the VREF pin from the +5 V supply, a voltage divider with fixed values is used instead of a potentiometer.
- some additional bypassing capacitors have been included.

The connector pin assignment can be found in table 3-4 on page 23. The $\overline{\text{VSYNC}}$ and $\overline{\text{HSYNC}}$ signals are directly fed through from the FPGA.

3.2.4 1M x 16 SRAM

This circuit's concept is based on the 1M x 16 SRAM described in section 3.1.3. The only difference is the use of 2 instead of 4 modules that are connected in parallel to get 16 bit words.

3.2.5 16M x 16 SDRAM

The memory modules used, the HYB39S256160CT by Infineon Technologies [21] come in the desired size and width, so the circuit is straightforward. $\overline{\text{CS}}$ is tied to GND and CKE to V_{CC} to permanently enable the memory module.

3.2.6 8-LED Bar

8 LEDs are included as a means of debugging and displaying. To connect the 8 LEDs to the general purpose I/O's of the FPGA without compromising bus signal integrity, the bus drivers of the 74 series (MM74HC244 by Fairchild Semiconductor [22]) have been used.

3.2.7 I/O Slot

To extend the capabilities of the board, a 36 pin header is installed. 32 pins are signal pins, the remaining 4 pins offer three supply voltages (+5 V, +3.3 V and +1.5 V) and ground.

3.2.8 RS-232 Single Port

This block is mainly the same as the one described in section 3.1.8 on page 22 except for the fact that only one driver pair of the MAX233A is used and one DB9 connector is present.

3.2.9 Ethernet

This block is identical to the one described in section 3.1.2 on page 20.

3.2.10 Switches

This block is identical to the one described in section 3.1.10 on page 24.

3.2.11 JTAG Header

Refer to section 3.1.11 on page 24 for details about this block.

3.3 Power Supply

To connect a mains adaptor to the XFBOARD, two different connectors are included: a 20 pin ATX connector used by standard PC power supplies, and a low-voltage connector to connect a +5 V power supply unit. The connectors tip has to be connected to +5 V, the sleeve to GND.

As neither variant shows up with the +1.5 V needed for the FPGA cores, this voltage has to be derived from the +5 V available. For that purpose, a TPS54613 step-down converter from Texas Instruments [23] has been installed. To generate the +3.3 V when no ATX supply is used, a TPS54616 is used. To disable or enable this converter, jumpers are present to be populated or depopulated according to the supply variant chosen. The converter circuits have been copied from the TIKDIMM Board [3].

All voltages are independently fused (+5 V, +3.3 V, +1.5 V); voltages generated by a step-down converter are fused after the converter. As a means for visual control of presence of the supply voltage, every voltage has its own status LED.

To measure the current drawn from the converters two jumpers are installed that may be depopulated to open the circuit to connect the measurement gear. All voltages present on the board are fed to an additional header. This header may be used to feed extension boards or external devices, e.g. a hard drive.

Chapter 4

Physical Realization

This short chapter explains the most important decisions and thoughts correlated with the design and manufacturing steps from the schematic to the physical board.

4.1 Trace Width and Spacing

As suggested in [10], the width of the signal traces on the top and the bottom layer of the PCB and the spacings between such traces is set to 5 mil (5 milli-Inches). A slightly lower trace width can be used in internal layers than the width used in top and bottom external or exposed traces. The trace width in inner layers is set to 4 mil. These values yield a 69Ω impedance for the top and bottom layers and a 56Ω impedance for the inner layers [29].

4.2 Number of Layers

Signal escaping (traces leaving the pin/ball area) can be quite difficult with the large FG packages of the FPGAs (see figures 4-1 and 4-2). Since the selected trace width is 5 mil (4 mil in inner layers), only one signal can be escaped between two pads. As a consequence, only two rows of pins can be escaped per layer as shown in figure 4-3. The XC2V3000 used as the R-FPGA, there are 8 rows of pins to be connected, therefore at least 4 signal layers are necessary. Additionally, there is a need for a ground plane and for one power plane per voltage present on the board. As the two inner signal layers' usage is high only in the perimeter of the FPGAs, they may be filled up with a power plane too. Two power-only planes, one for ground and one for the 3.3 V needed all over the board, and the two inner layers filled up with the two other voltages (1.5 V and 5 V) seem to be reasonable. Have a look at figure 4-4 for details about the PCB layer stackup.

4.3 Maximum Trace Length

To be sure not to have reflection problems on the signal traces which are transmission lines for high frequencies, the trace lengths must not exceed an upper limit. The clock frequency does not really matter, as the spectral composition of the signal depends on the edges' slopes and these slopes quickly induce frequency components above the

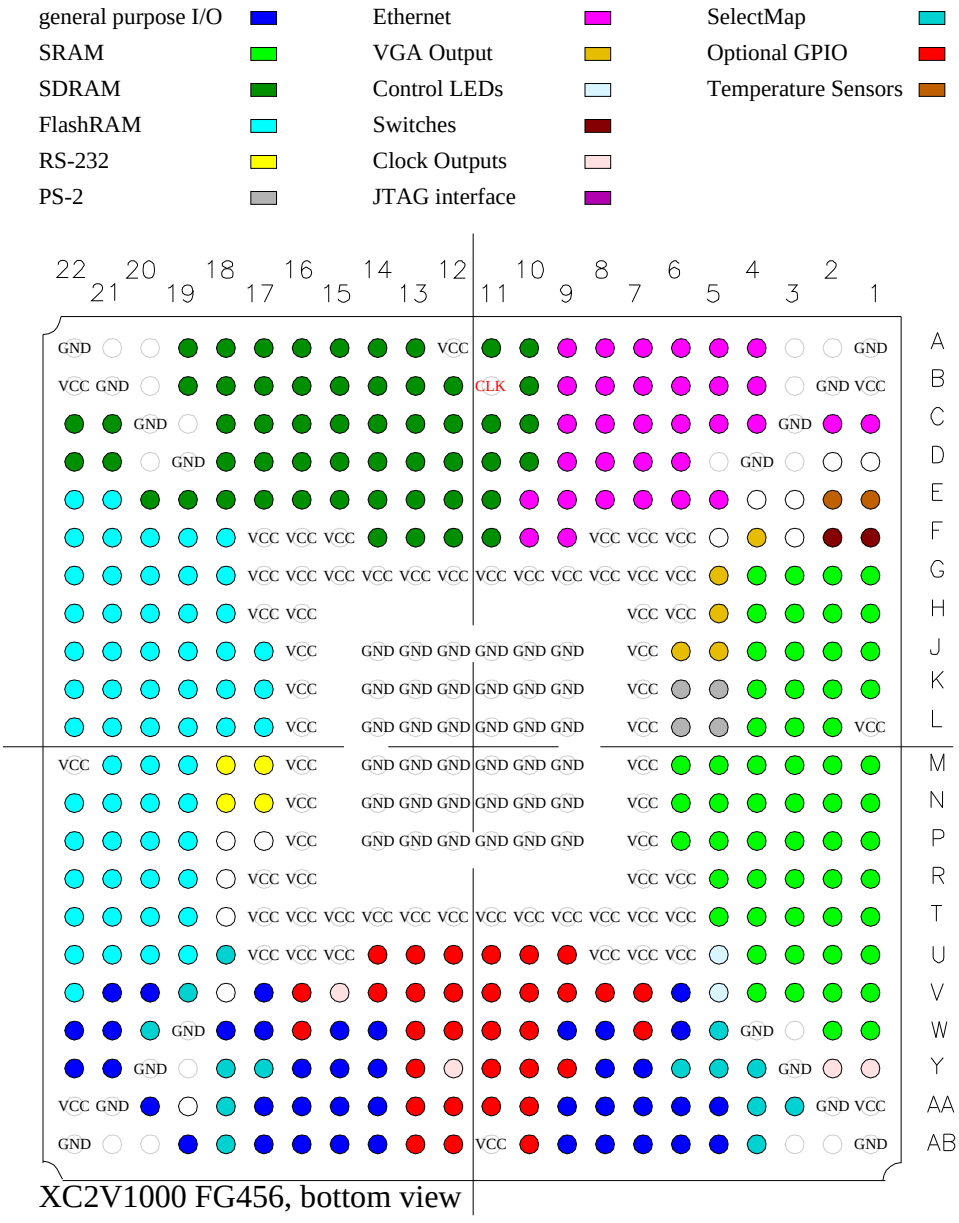


Figure 4-1: Pinout Diagram of the C-FPGA

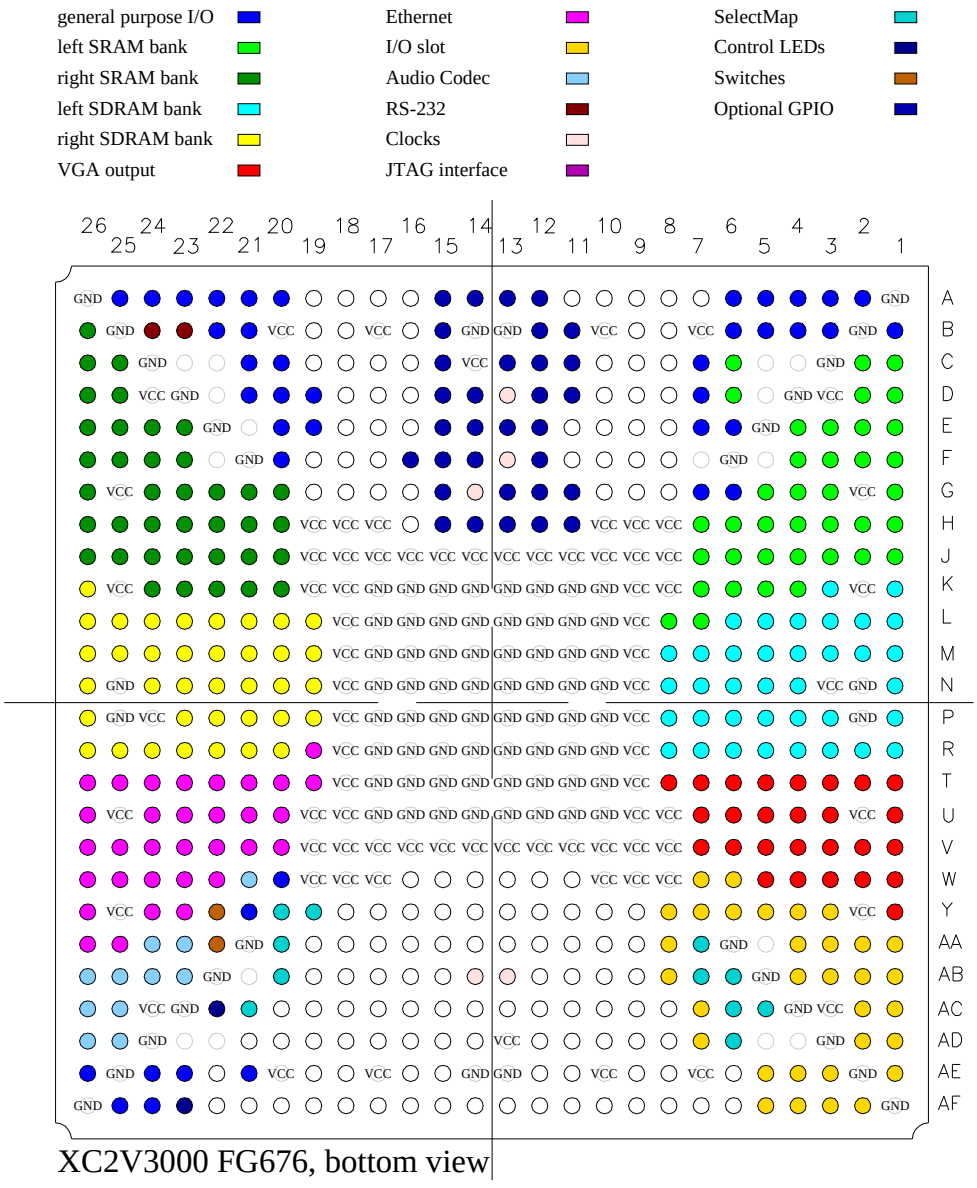


Figure 4-2: Pinout Diagram of the R-FPGA

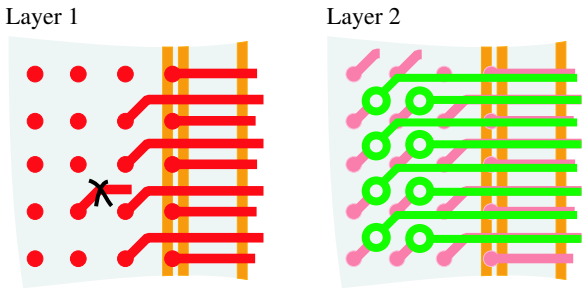


Figure 4-3: Signal Escaping Problem

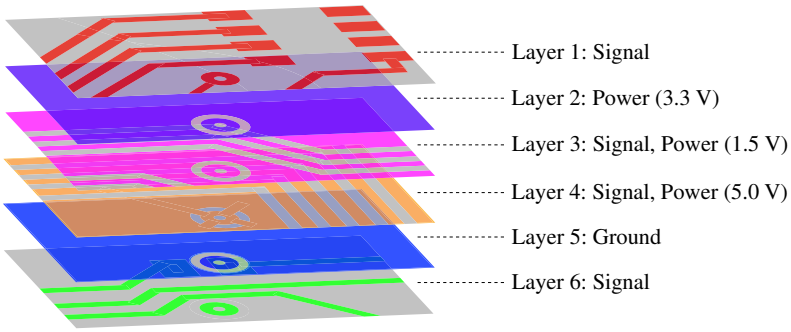


Figure 4-4: PCB Layer Stackup

base clock frequency. With the SRAM being the component with the fastest signal transients and assuming a transition time t_{tran} (rise or fall) of 3 ns[1], the following rule of thumb yields a maximum trace length of

$$l_{max} < t_{tran} \cdot 3\,\frac{cm}{ns} = 9\,cm$$

which can easily be guaranteed not to be exceeded. So signal termination is omitted except for the clock signals coming from the quarz oscillator (see sections 3.1.1 and 3.2.1).

[1]Unfortunately, the datasheet [20] does not provide an exact value, but 3 ns being 20% of the access time (15 ns) seem to be a reasonable value

Chapter 5

Programming Model

This chapter shall give you the information needed for successful programming of the XFBOARD. First, you get an overview of the components available for use, then you learn the concept of programming the board.

5.1 Available Components

To get an idea of the components available on the board you may want to catch a glimpse of figure 5-1 which provides a block diagram of the XFBOARD. Figure 5-2 offers a component side view of the PCB including the reference designators for the most important components; this may help you to locate them on the board. In the following subsections, some more information is given in textual and graphical form for the functional blocks in this diagram. However, details with low importance have been omitted in the graphics; for a more thorough discussion see the corresponding section in chapter 3 and the schematics in appendix A. Also, it may be useful to consult the relevant datasheets. Listings with all pin connections between the FPGAs and external components are available in appendix B. These pin listings may be used to generate the `*.ucf` constraints file; on the CD-ROM, there are two ready-to-use constraints files: `C-FPGA.ucf` and `R-FPGA.ucf`.

5.1.1 Configuration Section

As shown in figure 5-3 there are three elements to be configured from the outside world, the BootPROM for the C-FPGA, the C-FPGA itself and the R-FPGA. The BootPROM and the C-FPGA are connected to the same JTAG chain while the R-FPGA has its own JTAG connector. Please note that configuration of the two FPGAs via JTAG is not to be performed when using the board for those purposes it is intended for. As stated in the description of the development cycle in section 5.2, only the BootPROM is configured via JTAG. The option to configure the FPGAs via JTAG chain is present for higher flexibility. Also, the JTAG interface may be used for debugging via boundary scan.

For partial reconfiguration of the R-FPGA, its SelectMAP configuration port may be used. Refer to table B-11 on how to connect to the R-FPGA from the C-FPGA.

The two LEDs LED_CDONE and LED_RDONE are dark after power-up and get lit as soon as the C-FPGA or the R-FPGA is configured, respectively. You can reset the C-FPGA and trigger its reconfiguration using the PROGRAM_RESET switch.

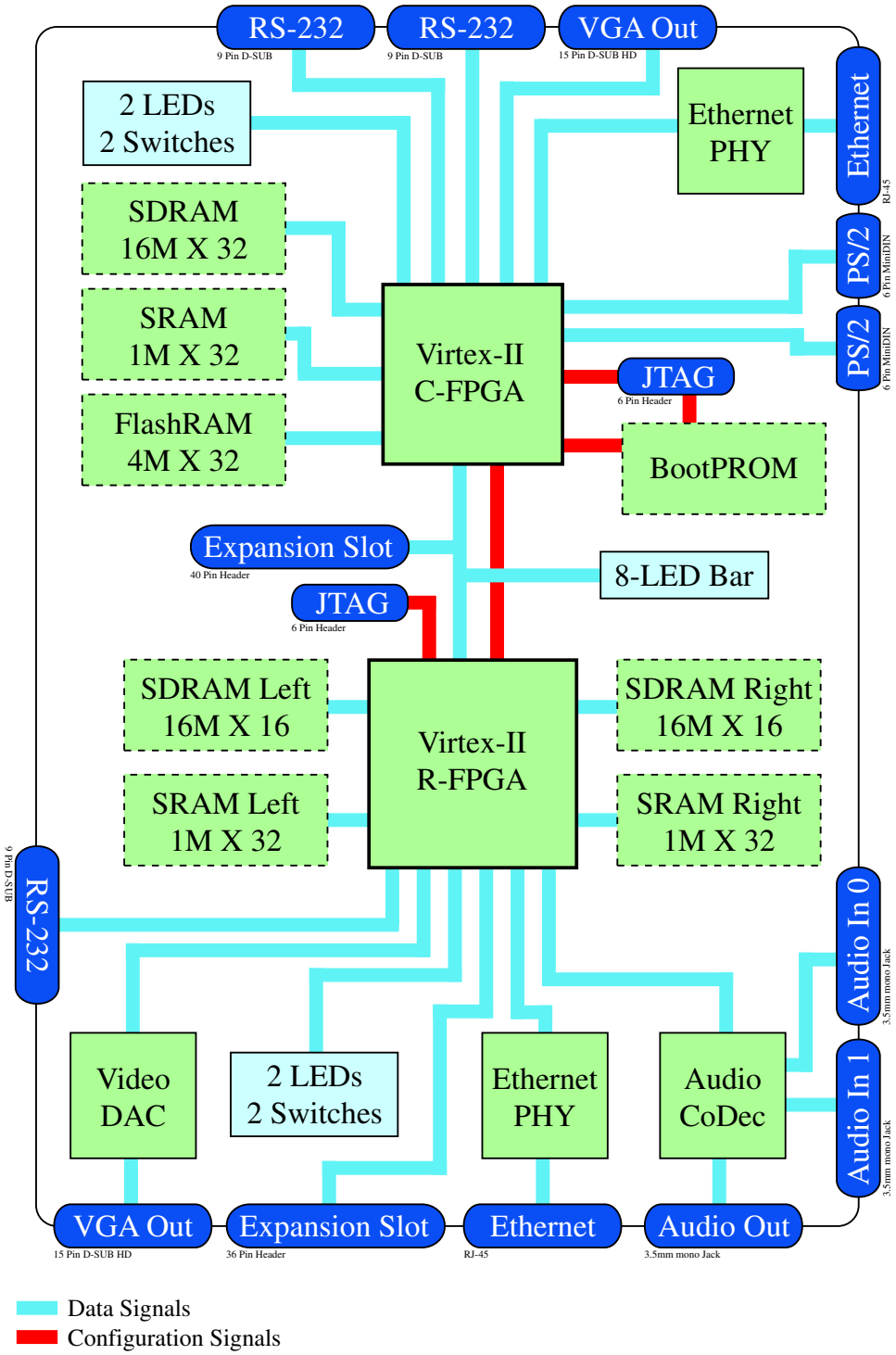


Figure 5-1: Block Diagram of the XF-Board (Power Supply omitted)

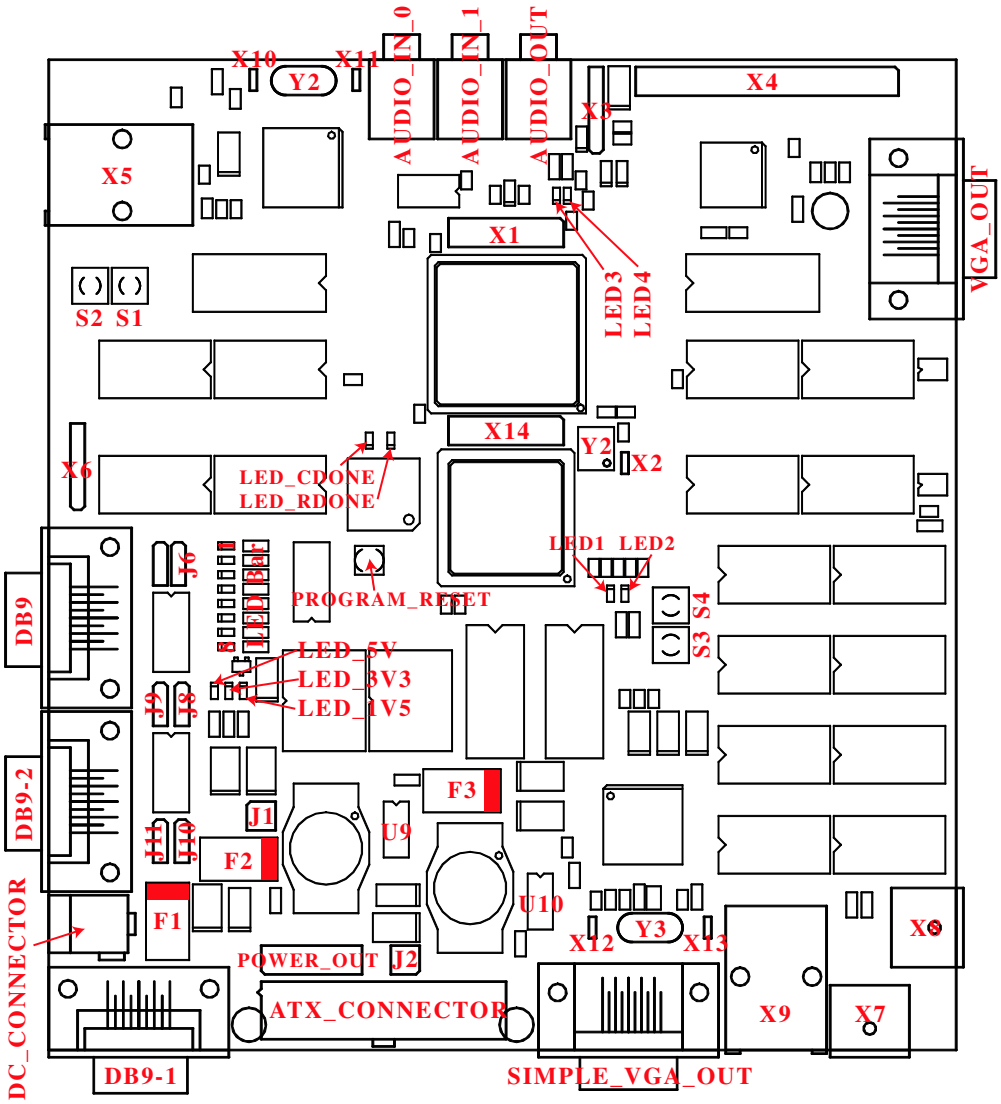


Figure 5-2: Component Side View of the XF-Board with Reference Designators

The R-FPGA may be reset via the SelectMAP port. Consulting [10], [14] and [9] is recommended for additional information on (re)configuration.

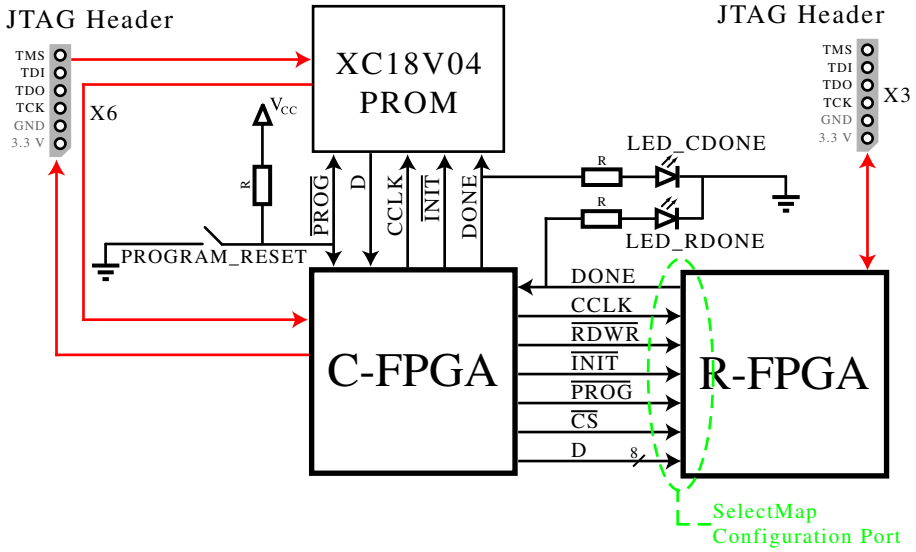


Figure 5-3: Configuration Section

5.1.2 FlashRAM

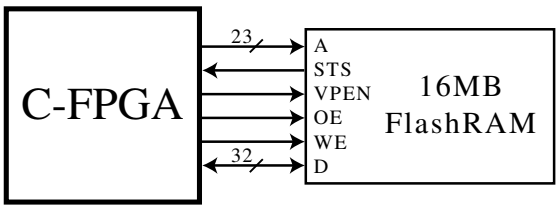


Figure 5-4: FlashRAM for the C-FPGA

Two Intel 28F640J3 with 16 MB of storage (4M x 32) are connected to the C-FPGA as shown in figure 5-4. In the graphic, the two FlashRAM modules are merged into one block as they are connected in parallel to provide 32 bit words. After power-up, the FPGA can read and/or write the FlashRAM. Refer to the datasheet [24] for details on reading from and writing to the flash array. The schematic is described in section 3.1.5. The pins of the C-FPGA connected to the FlashRAM are listed in table B-5.

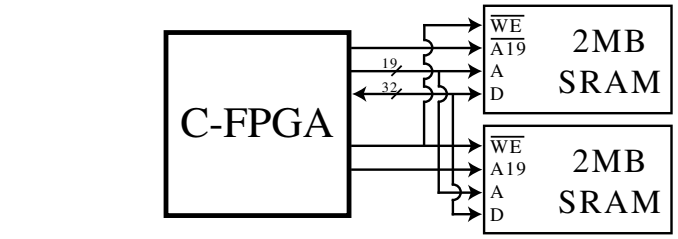
5.1.3 SRAM

The FPGAs have access to 4 MB of SRAM built from eight Alliance AS7C34096. The SRAM for the C-FPGA is organized as 1M x 32 bits as shown in figure 5-5(a),

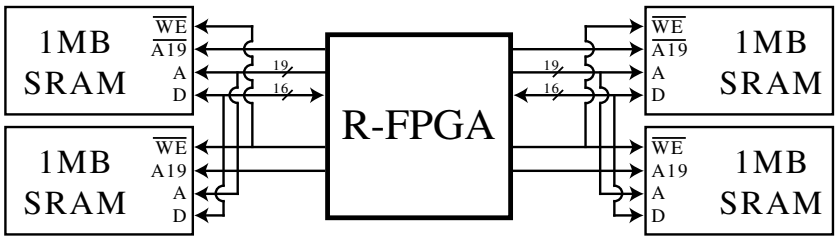
while the SRAM for the R-FPGA is organized in two independent banks as 1M x 16 (figure 5-5(b)). Please refer to sections 3.1.3 and 3.2.4 for the schematic, to tables B-13, B-23 and B-26 for the pins connected to the SRAM banks, and to the datasheet [20] on how to read and write from the SRAM.

Please note that `A19` and $\overline{\texttt{A19}}$ are in fact chip select signals that are used to extend the address space over a second group of memory modules.

`A19` and $\overline{\texttt{A19}}$ have to be complementary at every time to avoid bus contention which may yield device damage



(a) C-FPGA



(b) R-FPGA

Figure 5-5: SRAM for both FPGAs

5.1.4 S-DRAM

Both FPGAs have access to 64 MB of SDRAM. The C-FPGA is connected to two Infineon HYB39S256160CT to get a 16M x 32 bit structure. In figure 5-6(a), these two modules are merged into one block. The R-FPGAs SDRAM is organized in two independent banks each of which consists of an Infineon HYB39S256160CT, providing 16M x 16 bits (figure 5-6(b)). Sections 3.1.4 and 3.2.5 offer some additional information about the schematics. The FPGA pins connected to the SDRAM banks are listed in the tables B-10, B-22 and B-25, while details about the communication protocol may be gathered in [21].

5.1.5 Ethernet PHY

The XFBOARD interfaces to an Ethernet LAN at 10 or 100Mbps. Each FPGA is connected to an Intel LXT970A Ethernet PHY chip as can be seen in figure 5-7. The FPGA acts as a MAC and manages the transfer of data packets to and from the PHY

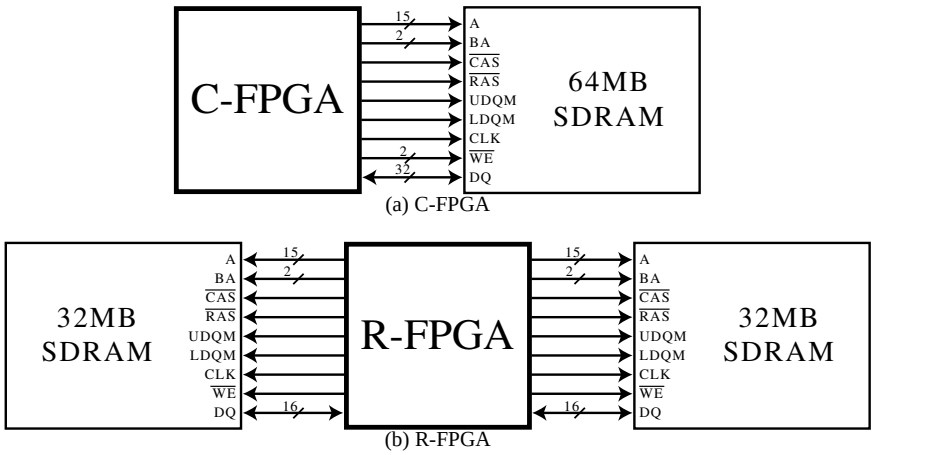


Figure 5-6: SDRAM for both FPGAs

chip. The FPGA also controls the configuration pins determining the operational mode of the PHY chip. How the Ethernet PHY chips are connected to the FPGAs is visible in the tables B-4 and B-19. Schematic details are described in section 3.1.2.

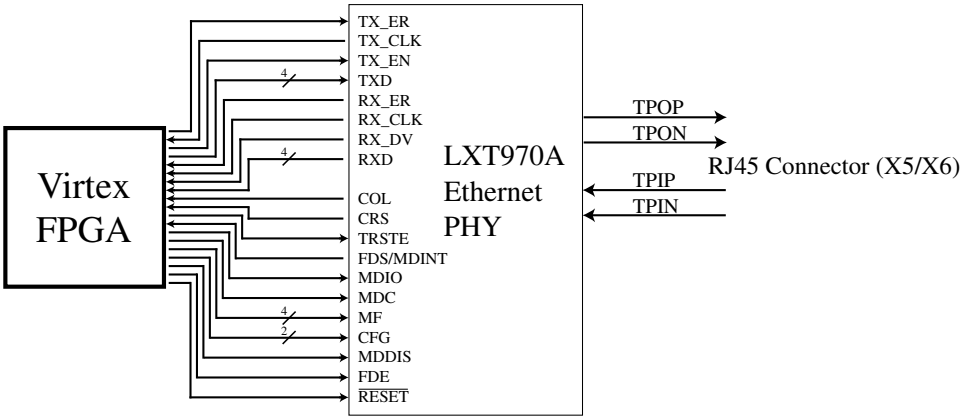


Figure 5-7: Ethernet Interfaces for both FPGAs

To send data to the LAN, the FPGA needs to enable the transmitter using `TX_EN` and then send the data bits on TXD. These operations have to be synchronized with the signal `TX_CLK` . If a transmission error ocurred, the MAC (i.e. the FPGA) needs to inform the PHY by asserting `TX_ER` . As soon as valid data has been received by the PHY, the FPGA gets notified using `RX_DV` , and this data is fed to the FPGA via `RXD` in sync with the receiver clock `RX_CLK` . The PHY asserts `RX_ER` when it receives invalid symbols from the network.

The FPGA can disable the interface to the PHY by asserting the tristate control `TRSTE` . Otherwise, the FPGA passes management information to and from the PHY over the serial data line `MDIO` in sync with the clock `MDC` . The FPGA can be alerted to changes in PHY chip status by the `FDS/MDINT` interrupt line.

The FPGA can configure the PHY using the MF and the CFG lines. The 5 bits on MF can set operation functions like auto-negotiation, repeater mode, scrambler mode etc. The 2 CFG bits are used to chose 10Mbps or 100Mbps operating speed and to enable/disable the link test function. FDE determines either full-duplex or half-duplex operation, and MDDIS disables the management information interface.

The meaning of the LEDs at the front side of the RJ45 connectors is as follows:

yellow	link active
green	receiver active
orange	transmitter active

You may want to learn how to communicate with the PHY chip in more detail using the datasheet [16].

5.1.6 Audio CoDec

The XFBOARD uses an AK4563A to provide two stereo inputs and one stereo output. For details about the connections between the FPGA and the CoDec, please refer to figure 5-8 and to table B-16. You may want to visit section 5.1.6 for an explanation of the schematic.

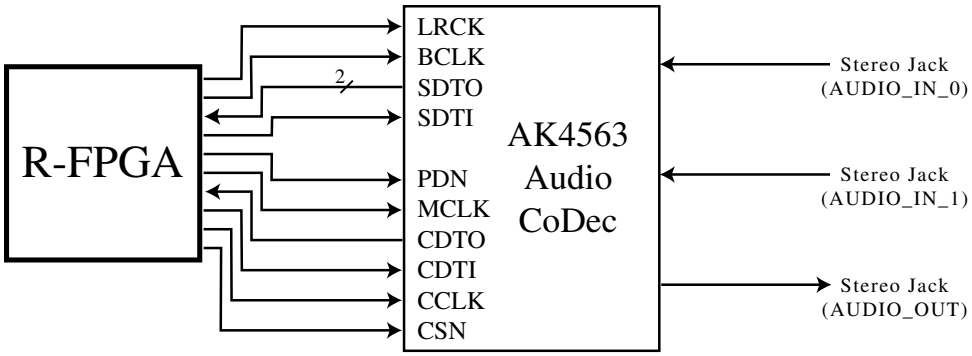


Figure 5-8: Audio In- and Output

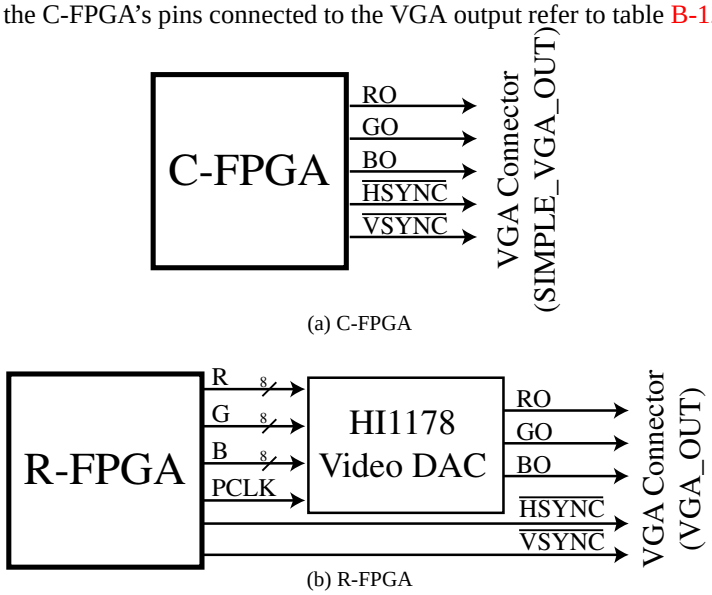
The audio signals coming from the jack connectors are digitized and fed to the FPGA using the I^2S protocol: serial audio data is offered on SDTO<0> and SDTO<1> in sync to the bit clock $\overline{\text{BCLK}}$. Left-right information is given by the clock LRCK; both clocks are generated by the FPGA. Data to be converted to the analog domain and output via the jack connector has to be present on the SDTI line, again in sync to the two abovementioned clocks. The master clock MCLK from the FPGA synchronizes all the internal operations of the codec.

Additional information on operation status and operation functions (e.g. the peak-meter) can be read to and written from the control register using the signals CSN, CCLK, CDTI and CDTO. To reset the CoDec, pull down the PDN.

The datasheet [18] offers some additional information on the CoDec.

5.1.7 VGA Outputs

Both FPGAs can generate video signals for display on a VGA monitor. The C-FPGA directly feeds the color information to the VGA connector (figure 5-9(a)) using the one-bit signals RO , GO and BO , yielding 3 bit color resolution (8 colors). The vertical and horizontal sync pulses $\overline{\text{VSYNC}}$ and $\overline{\text{HSYNC}}$ have to be generated by the FPGA. Schematic details about this simple VGA output can be found in section 3.1.7, for a listing of the C-FPGA's pins connected to the VGA output refer to table B-12.



(a) C-FPGA

(b) R-FPGA

Figure 5-9: VGA Outputs for both FPGAs

The R-FPGA generates the signals using the Intersil HI1178 video DAC [19]. The DAC accepts three 8 bit data signals, R , G and B , as a color information, synchronized to the pixel clock PCLK generated by the FPGA. The analog color information is then fed to the VGA output using RO , GO and BO . Again, the sync pulses have to be generated by the R-FPGA. The schematic of this circuit is described in section 3.2.3. The corresponding pin listing is available in table B-29.

5.1.8 PS/2 Ports

The C-FPGA has access to two PS/2 ports to connect a keyboard and a mouse. The FPGA receives two signals from each interface: a clock signal and a serial data stream that is synchronized with the falling edges on the clock signal. Figure 5-10 illustrates the connection between the R-FPGA and the PS/2 ports. The pin listing is located in table B-8. A good resource for in-depth PS/2 information is [30].

5.1.9 Serial Port

As shown in figure 5-11, the C-FPGA is connected to two RS-232 ports, while the R-FPGA has access to one RS-232 port. The Maxim MAX233 are used to shift the signals to the correct levels. Use the jumpers to set the correct communications function (DTE

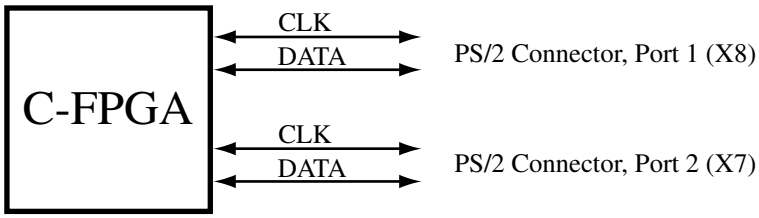


Figure 5-10: PS/2 Interfaces, C-FPGA

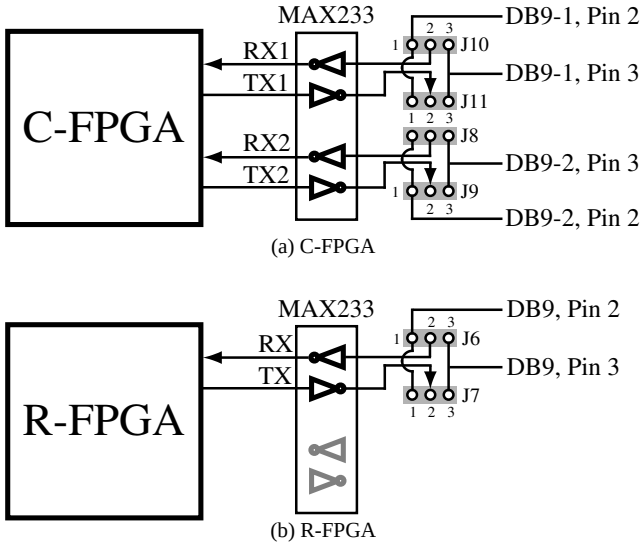


Figure 5-11: RS-232 Interfaces for both FPGAs

vs. DCE); figure 5-12 may give you a hand in deciding the correct jumper position. For the corresponding schematics visit section 3.1.8. The pin mapping is given in table B-9 and B-27. A good resource for the RS-232 specification is [31].

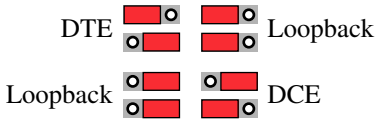


Figure 5-12: RS-232 Jumper Settings

5.1.10 General Purpose I/Os, Expansion Headers, LED Bar

The two FPGAs are connected with 40 general purpose lines (`GPIO`) to allow for intercommunication. 8 of these lines are also connected to an LED bar. These LEDs are active-high meaning that an LED segment will glow when a logic-high is applied to it.

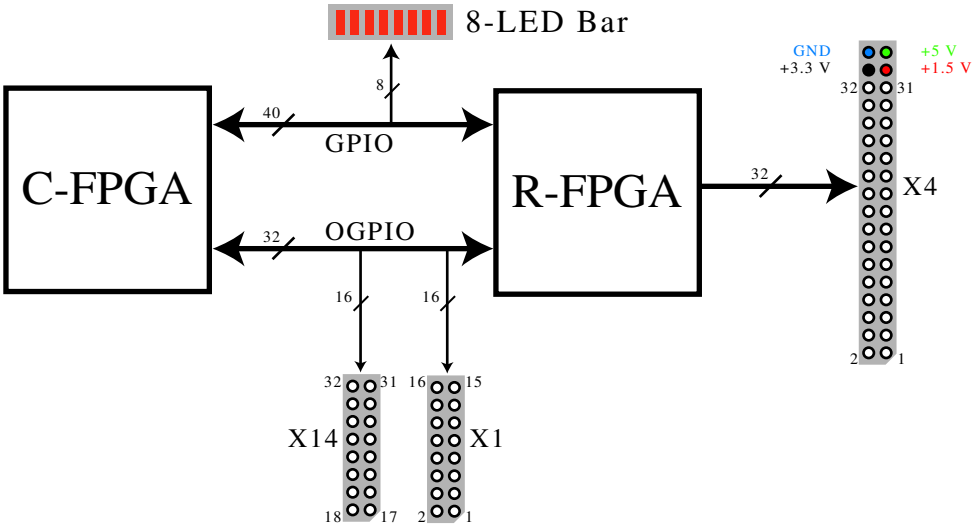


Figure 5-13: General Purpose I/Os, Expansion Headers, LED Bar

Optionally, there are 32 additional lines (`OGPIO`) that may be used for interconnection between the two FPGAs. Overmore, two 16 pin headers expansion headers are provided. Nevertheless, if the R-FPGA is meant to be partially reconfigured, the `OGPIO` should not be connected at the R-FPGA side (i.e. left unconnected in the `*.ucf` for the R-FPGA), still allowing access to the headers from the C-FPGA side.

An additional 36 pin expansion header is connected to the R-FGPA. 32 pins are signal pins, the remaining four pins are used for power supply of the expansion board. As a means for expansion board fixture, some mounting holes are provided.

You may want to read the schematics' explanations in the sections 3.1.9, 3.2.7 and 3.2.6 and the pin mappings in the tables B-6 and B-7 for the C-FPGA and the tables B-20, B-24 and B-21 for the R-FPGA.

5.1.11 LEDs and Pushbuttons

As shown in figure 5-14, each FPGA is connected to two pushbuttons and two LEDs. The LEDs are active-low, i.e. an LED glows when it is fed with a logic-low signal. To use the switches, internal pull-down resitors must be instantiated in the corresponding IOB of the FPGA.

5.1.12 Temperature Sensors

The Virtex-II FPGAs contain an integrated temperature-sensing diode. The analog signals delivered by these diodes are digitized by two SMBus temperature ICs (Maxim MAX1617 [13]). The digital representation of the temperature can be read by the C-FPGA. Refer to figure 5-15 and table B-15 on how to connect to these modules. How to work with SMBus devices is described in [32].

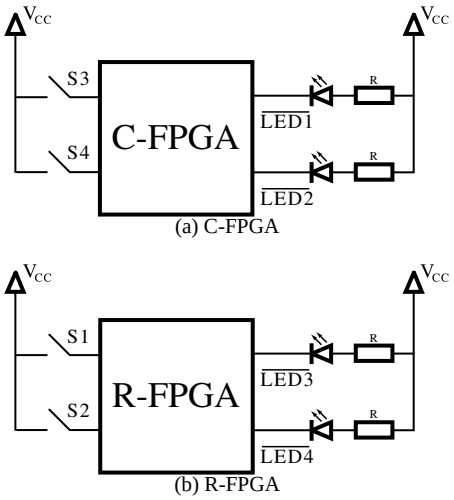


Figure 5-14: Buttons and LEDs for both FPGAs

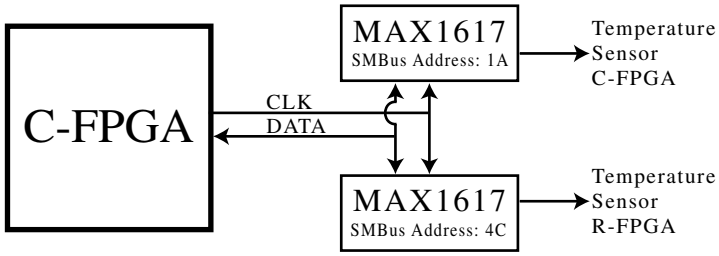


Figure 5-15: Temperature Sensors

5.1.13 Power Supply

There are two possibilities to feed the XFBOARD with power:

- Using a standard PC power supply

  Before connecting the power supply to the ATX connector, be sure to depopulate J1 and J2 to prevent the DC/DC converter from being destroyed.

  The PC power supply offers the +5 V and +3.3 V needed on the board, the +1.5 V for the core supply of the FPGAs is derived from the +5 V. The ATX connector is fed with an additional voltages (±12 V, -5 V, -3.3 V); these are not needed on the board, but they are forwarded to a power header to be used for possible external devices.

- Using a 5 V DC supply

  With this option used, the +3.3 V drop out and need to be generated on-board by an additional DC/DC converter. To put that converter in operation you have to

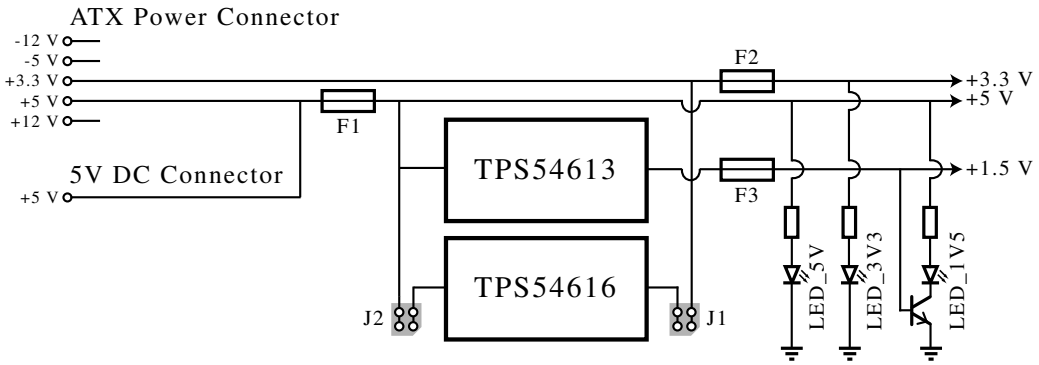


Figure 5-16: Power Supply

populate J1 and J2 . The other voltages delivered by the ATX connector drop out too and therefore won't be available at the power header mentioned above.

Never ever use both power connectors at the same time as this will destroy the XFBOARD

Three LEDs, one for every voltage, are glowing as long as all voltages required for proper operation are present. All voltages are fused; as soon as a fuse is blown, the corresponding LED ceases. As the +1.5 V are derived from the +5 V, this voltage will be lost as soon as the +5 V fuse is gone. When using the 5 V DC supply option, the same applies for the +3.3 V.

Figure 5-16 is a simplified view of the power supply section, for a view at a higher level of detail refer to section 3.3 explaining the schematic.

5.2 Development Cycle

As the exact workflow highly depends on the implementation of the CPU running on the C-FPGA, only a rough conceptual approach can be depicted here. The development flow can be divided into eight steps as shown in figure 5-17:

1. Store the firmware into the BootPROM, figure 5-17(a)

   A basic boot-loader and the CPU soft core (MicroBlaze) have to be written to the FlashPROM. This configuration data will be automatically loaded on power-up. It is recommended to include configuration data for a simple system check that will be performed shortly after configuration of the C-FPGA.

2. Load the firmware from the BootPROM, figure 5-17(b)

   As soon as power is supplied the firmware is loaded and the CPU soft core (MicroBlaze) gets installed. Optionally, a basic system check can be performed. This process gets reinitiated whenever the reset button PROGRAM_RESET is hit.

3. Load the CPU's program, figure 5-17(c)

   Now, the firmware is ready to receive program code for the CPU. The IP packets' contents get stored into the SRAM serving as program memory.

4. Start program execution, figure 5-17(d)

   As soon as the last package is received, the program counter jumps to the first instruction in the SRAM and the CPU starts to execute the program

5. Optional: partial reconfiguration of the C-FPGA, figure 5-17(e)

   The C-FPGA may be partially reconfigured using the SelectMAP configuration port, e.g. to adapt the DCMs for different clock frequency settings. At the time writing, this procedure has not been evaluated and we have no idea wether this approach is feasible at all.

6. Write configuration data to the FlashRAM, figure 5-17(f)

   Now, the configuration bitstreams for the R-FPGA are received in IP packets and stored in the FlashRAM. Storing them into the FLashRAM means that the configuration data will be available again when power is shut down and then applied again.

7. Copy the FlashRAM's contents to the SDRAM, figure 5-17(g)

   When the system is started up again, the FlashRAM's contents get transferred to the SDRAM to be available there for a fast configuration of the R-FPGA. Alternatively, this transaction may be inferred by an explicit command, either to copy every configuration or only a subset of the configurations to the SDRAM.

8. Load OS and tasks to the R-FPGA, figure 5-17(h)

   Finally, the configuration data for the OS frames and the HW tasks can be loaded to the R-FPGA and put into operation. The XFBOARD is ready to perform the tasks it is intended to!

To debug the software running on the XFBOARD board, the following debugging channels may be used:

- The simplest approach for debugging is the use of the LEDs and the pushbuttons (see section 5.1.11). The LEDs can be used to monitor the logic state of some signals, interrupts or test signals can be generated using the push buttons. If the two LEDs for each FPGA don't suffice, the LED bar connected to the general purpose I/O may be used (section 5.1.10). A logic analyzer or similar equipment may be connected to the expansion headers to monitor a larger group of signals. For all monitoring processes on internal signals, these signals must be mapped to an I/O to allow access from the outer world.

- A more sophisticated approach is using the RS-232 interfaces to read and write data to the FPGAs. This debugging alternative runs on a higher level of abstraction because simple RS-232 drivers and a useful debugging protocol have to implemented.

- For skillful monitoring, the simple VGA output of the C-FPGA may be used. When intending to use the VGA output for debugging, be aware of the fact that a moderately complex 3 bit VGA driver has to be implemented.

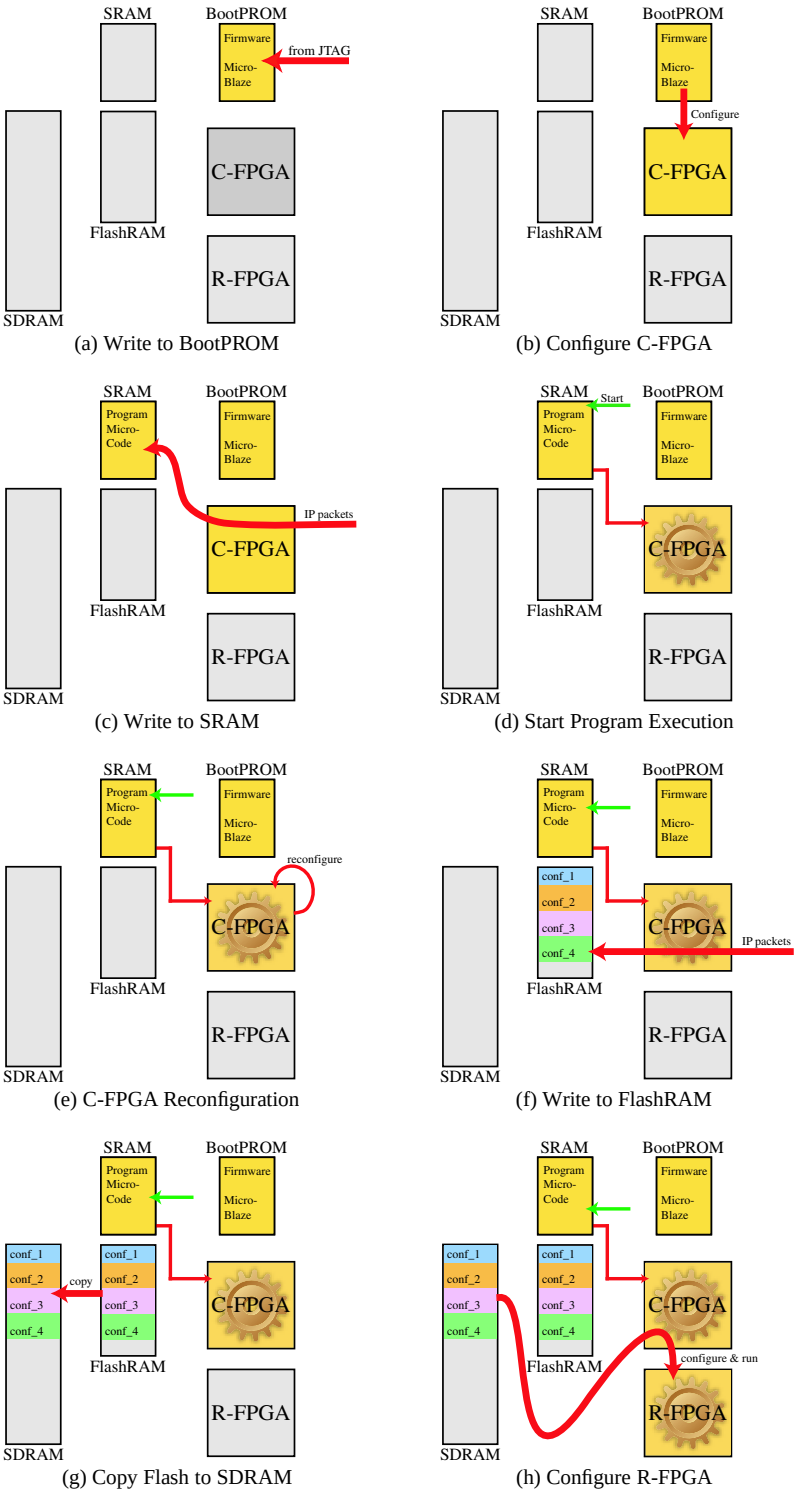


Figure 5-17: XF-Board Startup Procedure and Software Loading

- Of course, you can use the IP protocol for debugging, too. But you need to be in the know of what you're doing...
- You may use the JTAG headers to read back the configuration of the FPGAs to analyze the bitstreams.

Always keep in mind: the more advanced the debugging channel you use, the less immediate results you get.

Chapter 6

Outlook and Acknowledgements

In this short chapter, some final remarks are due. I want to give some hints for future improvements of the XF*BOARD*, tell what I learned during the design process, and thank all those people that gave me a hand during this semester thesis.

6.1 Future Work and Improvements

The board may be sped up without being redesigned by using faster speed grades for the critical components, i.e. the FPGAs and the memory modules. Nevertheless, the board's limits may be reached by replacing these components as an in-depth exploration of line impedance has not been performed due to lack of time. For very fast speeds, the lines might ask for proper termination.

If the schematic proofs workable, the layout process may be optimized. The length of the interconnections between critical components may be shortened by applying a higher component placement density. This will help improve signal integrity and reduce interconnect delays. However, this means manufacturing a new board.

Of course, existing software modules and configuration bitsreams have to be ported to the new hardware, and new modules and bitsreams have to be implemented. The RHWOS has to be adapted to seamlessly work with the XF*BOARD*, and additional services have to be implemented.

An evaluation has to be performed to get a measure for the usability and performance gain of XF*BOARD* when compared to the XESS board. Also part of future work are the extension boards for the expansion slots.

6.2 Lessions Learned

If I started again a project with the same dimension, I would allocate much more time for the whole design process. Many steps have not been possible to be thought-out as careful as needed.

As the board has not been put into operation until now, all design decisions have not been proved nor disproved. Therefore I am not able to reconsider these decisions at the time writing.

6.3 Acknowledgements

First of all, I want to thank Herbert Walder, my advisor, for guiding me through this project. His ideas and his experience with the reconfigurable OS and the corresponding hardware have been very helpful.

I also wish to thank Matthias Dyer for being my co-advisor. Many thanks go to other assistants that have been bothered with my concerns, especially the members of the workgroup involved in the specification elaboration process, i.e. Herbert Walder, Marco Platzner, Matthias Dyer, Christian Plessl and Andreas Schweizer. A special thank you goes to Andreas Schweizer for transferring his knowledge acquired when designing the TIKDIMM board.

I want to thank the students working in the same room who helped in resolving minor problems and offered useful tips for working with the software environment.

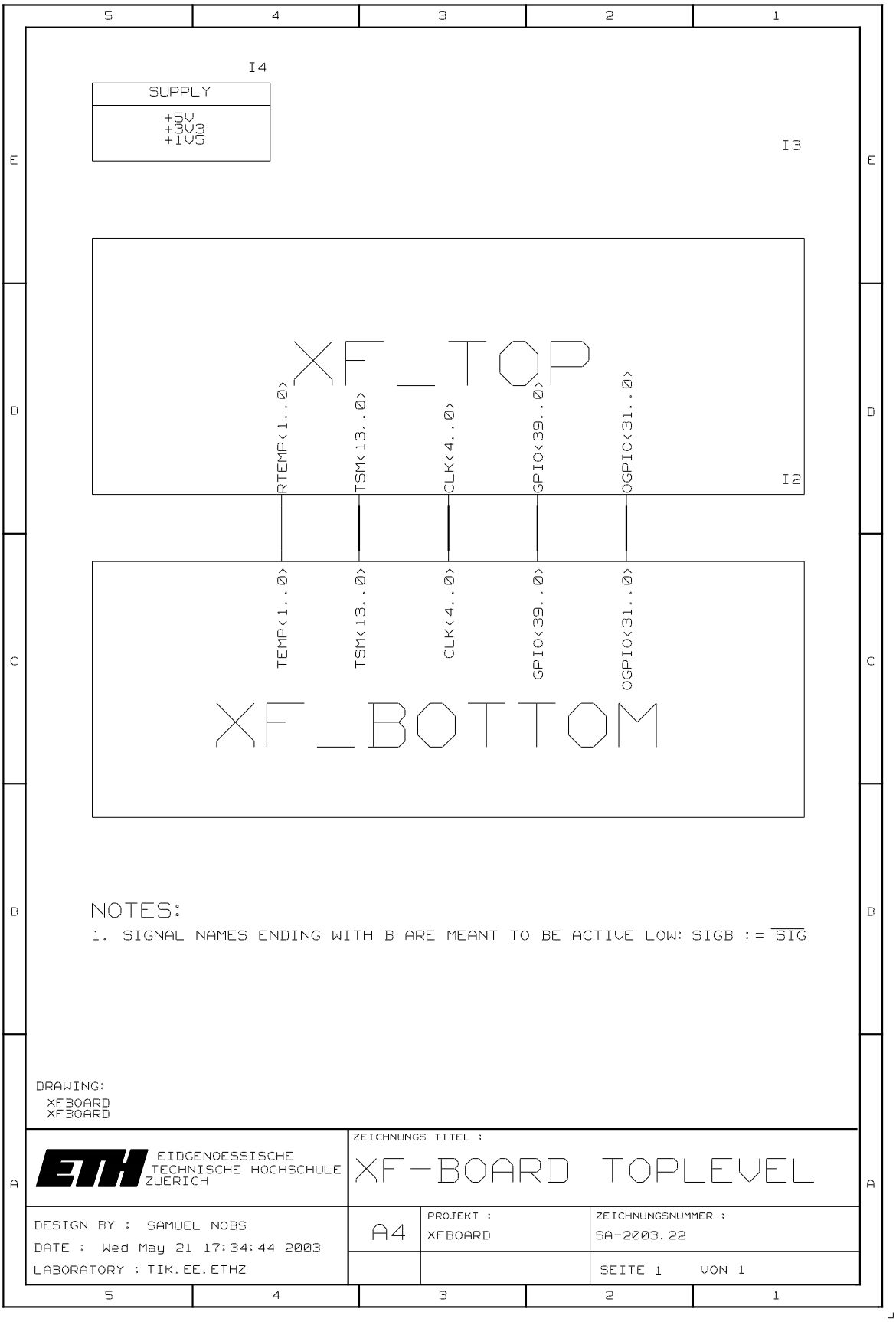
A thank you is aimed at the Support Group (Dienstgruppe) for helping with and installing the computer environment. Luckily, they perform a daily incremental backup of the home directories...

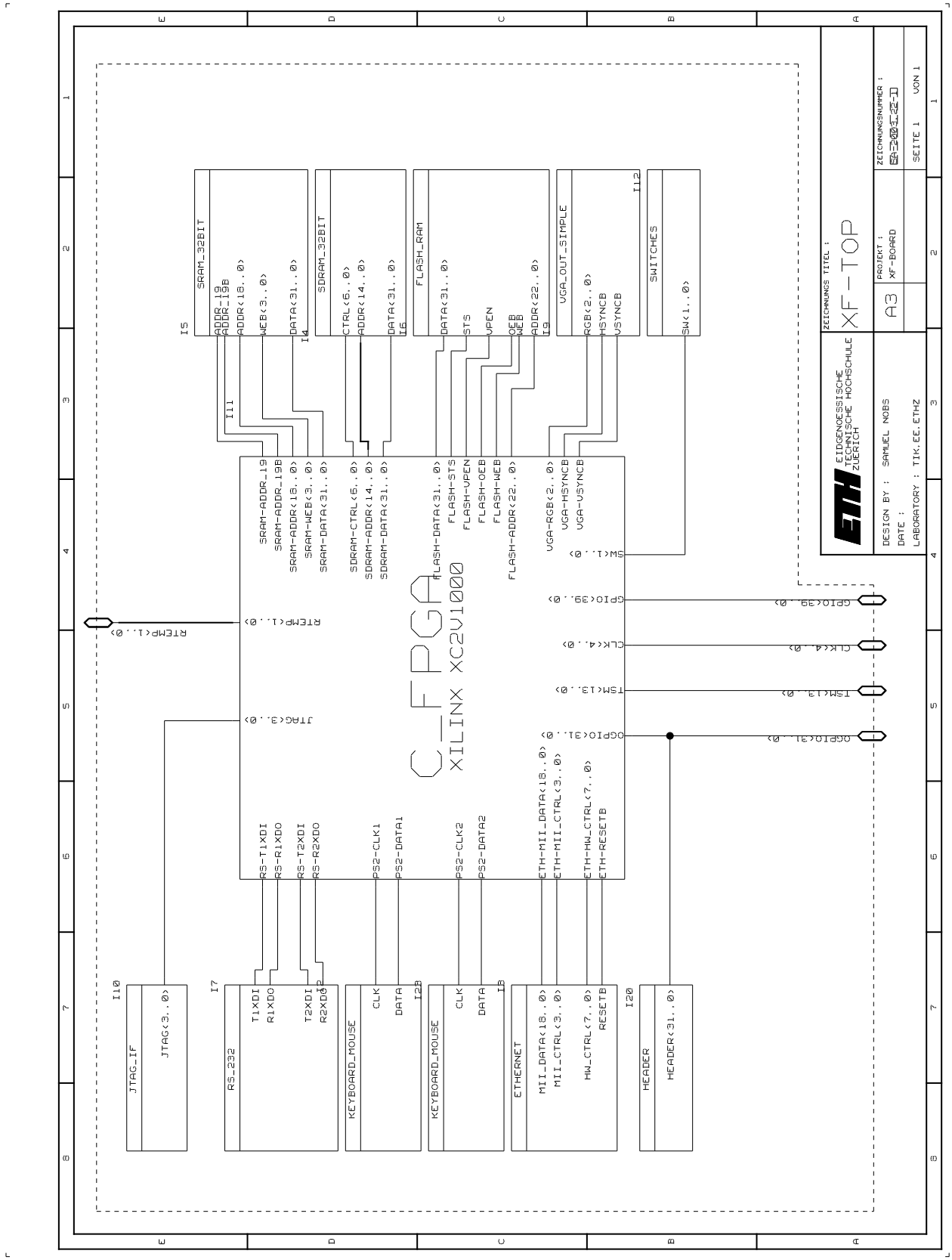
I also appreciate the work of Ruedi Koeppel, staff of the DZ (Design Zentrum), who organized a CAD course in the beginning of the semester. Without that course and Ruedi's unhesitant help, the XFBOARD would not have been feasible at all.

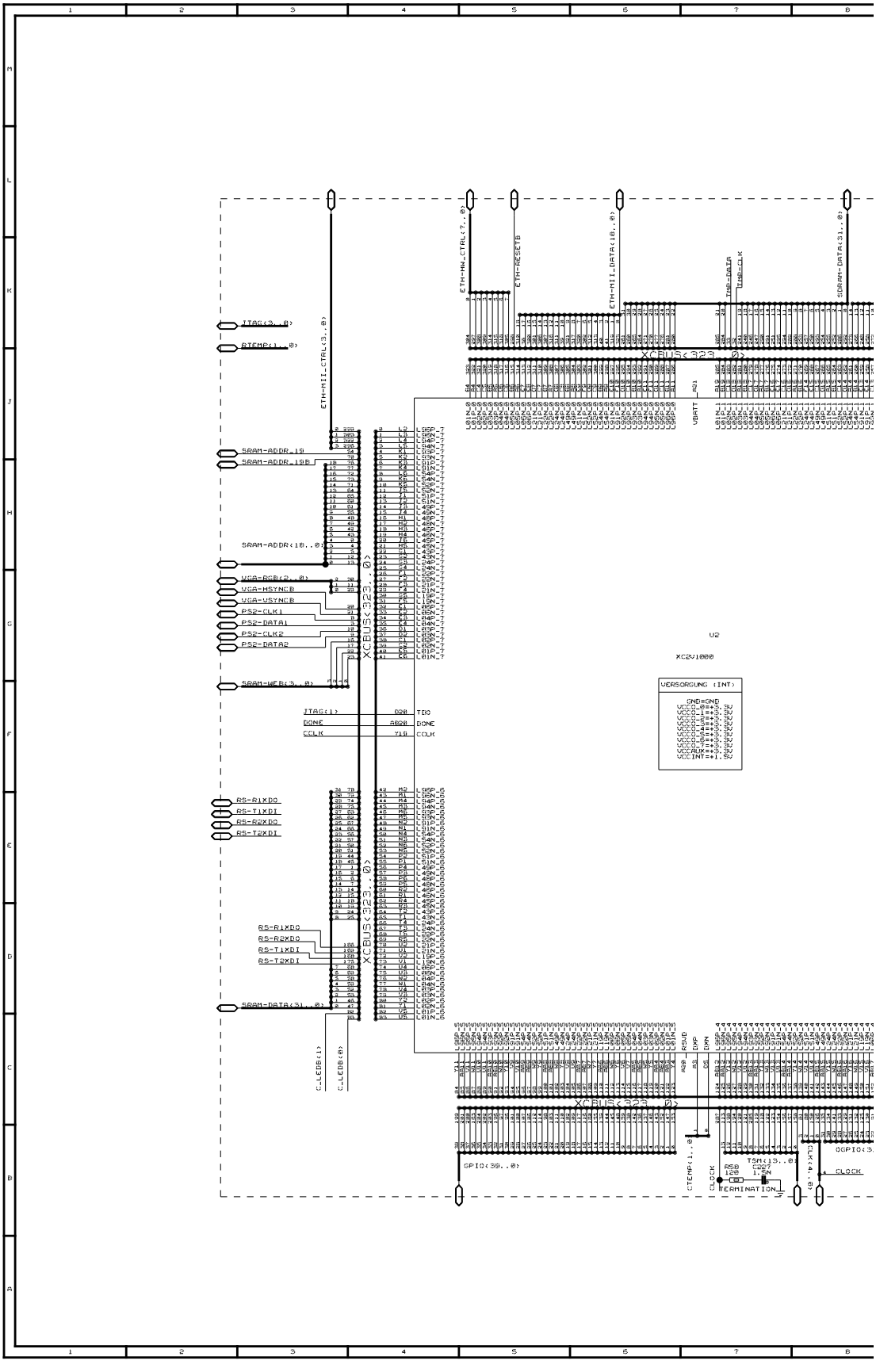
I wish to thank Prof. Dr. Lothar Thiele for being my supervisor and for his confidence in this project.

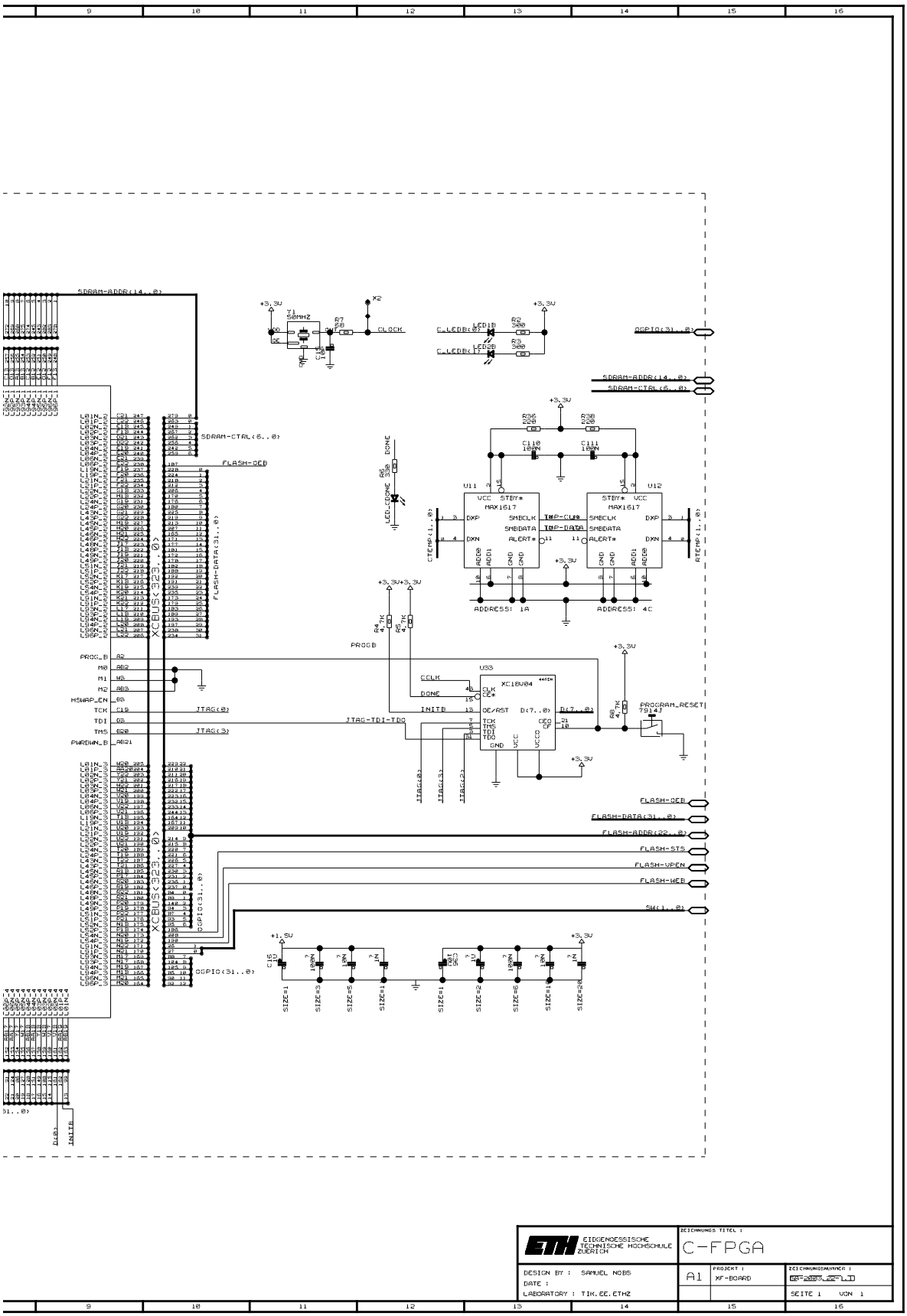
Appendix A

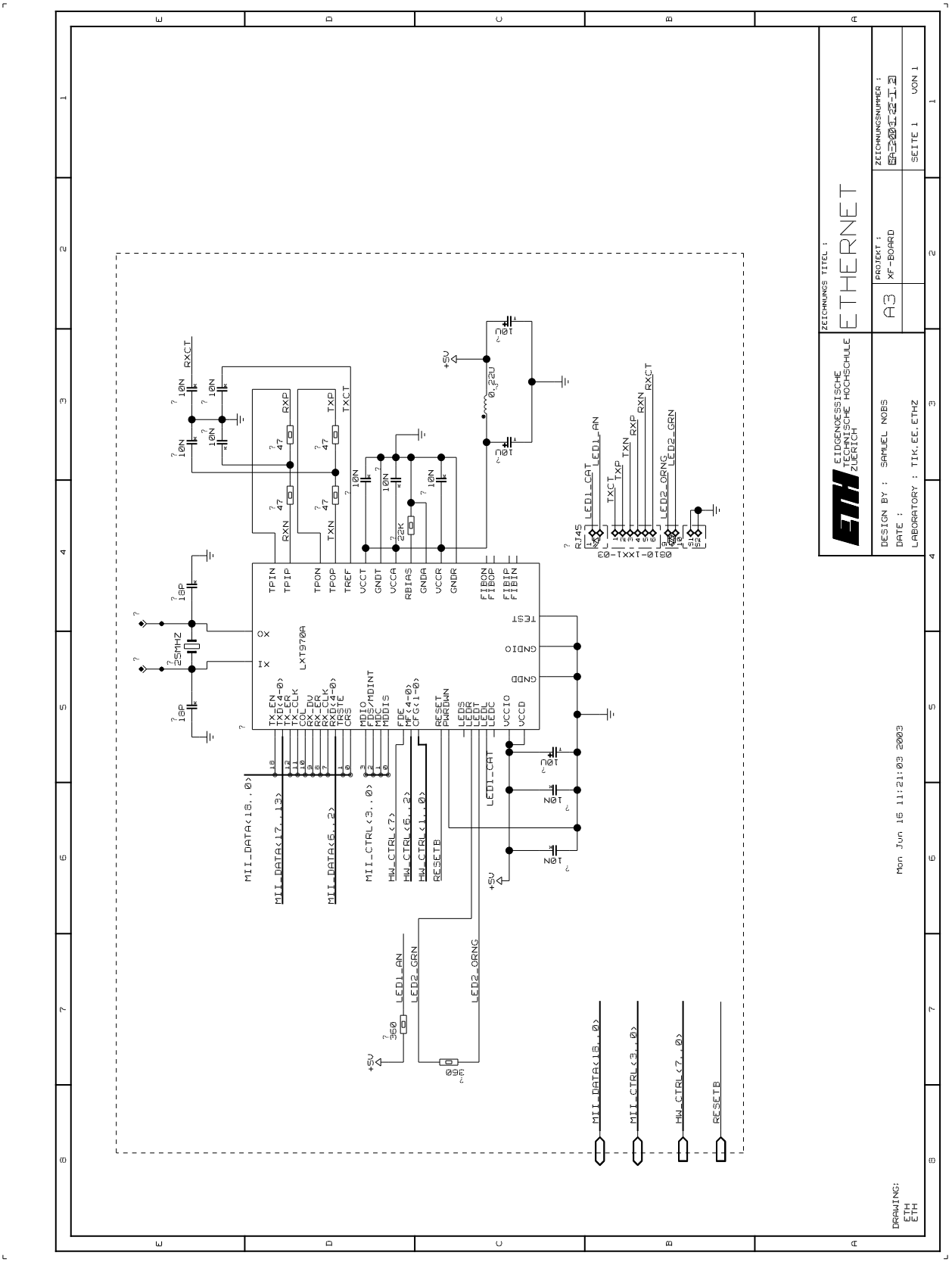
Schematics

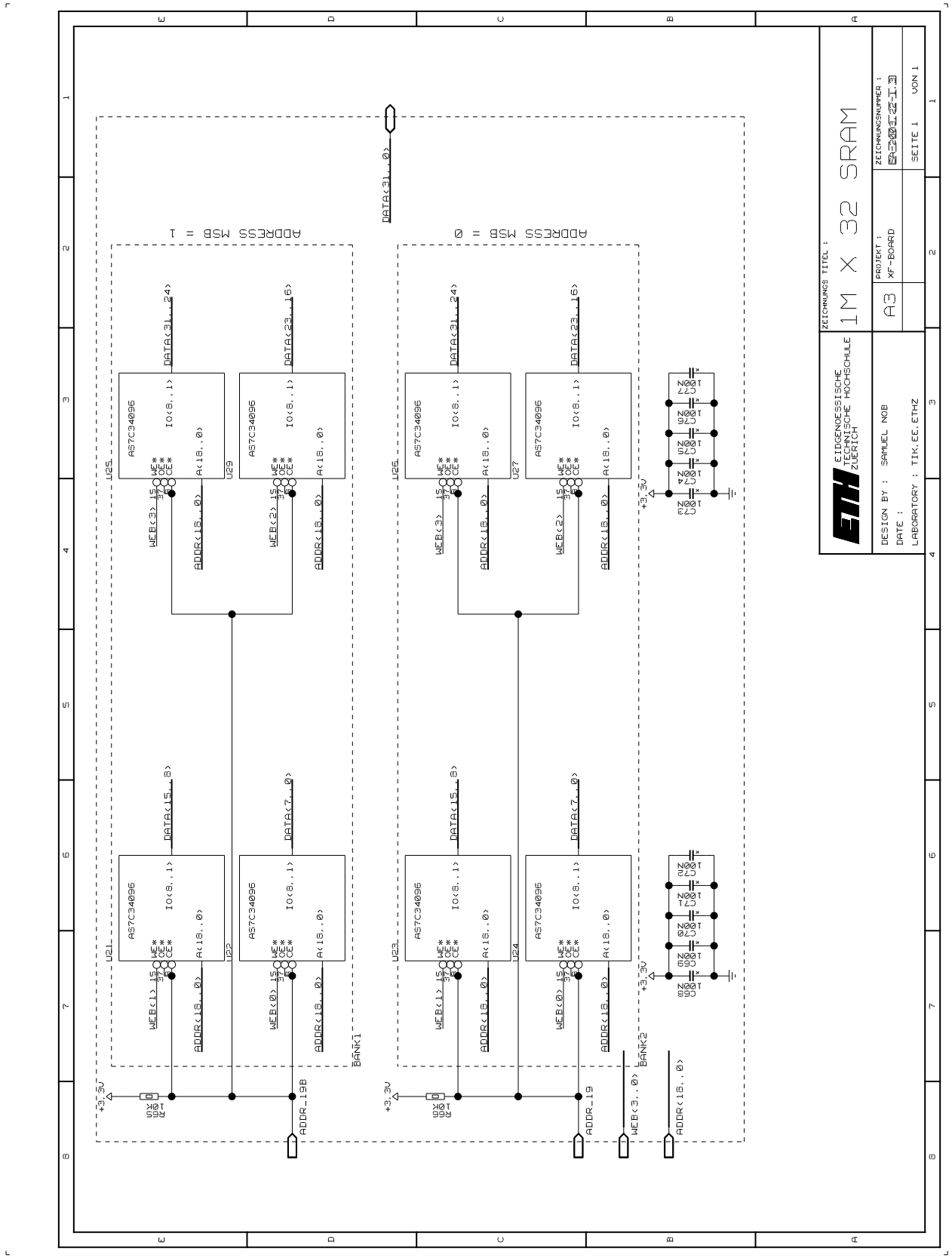

I4
SUPPLY
+5V
+3V3
+1V5
I3
XF_TOP
RTEMP<1..0>
TSM<13..0>
CLK<4..0>
GPIO<39..0>
OGPIO<31..0>
I2
TEMP<1..0>
TSM<13..0>
CLK<4..0>
GPIO<39..0>
OGPIO<31..0>
XF_BOTTOM
NOTES:
1. SIGNAL NAMES ENDING WITH B ARE MEANT TO BE ACTIVE LOW: SIGB := SIG
DRAWING:
XFBOARD
XFBOARD
EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH
ZEICHNUNGS TITEL :
XF-BOARD TOPLEVEL
DESIGN BY : SAMUEL NOBS
DATE : Wed May 21 17:34:44 2003
LABORATORY : TIK.EE.ETHZ
A4
PROJEKT :
XFBOARD
ZEICHNUNGSNUMMER :
SA-2003.22
SEITE 1 VON 1

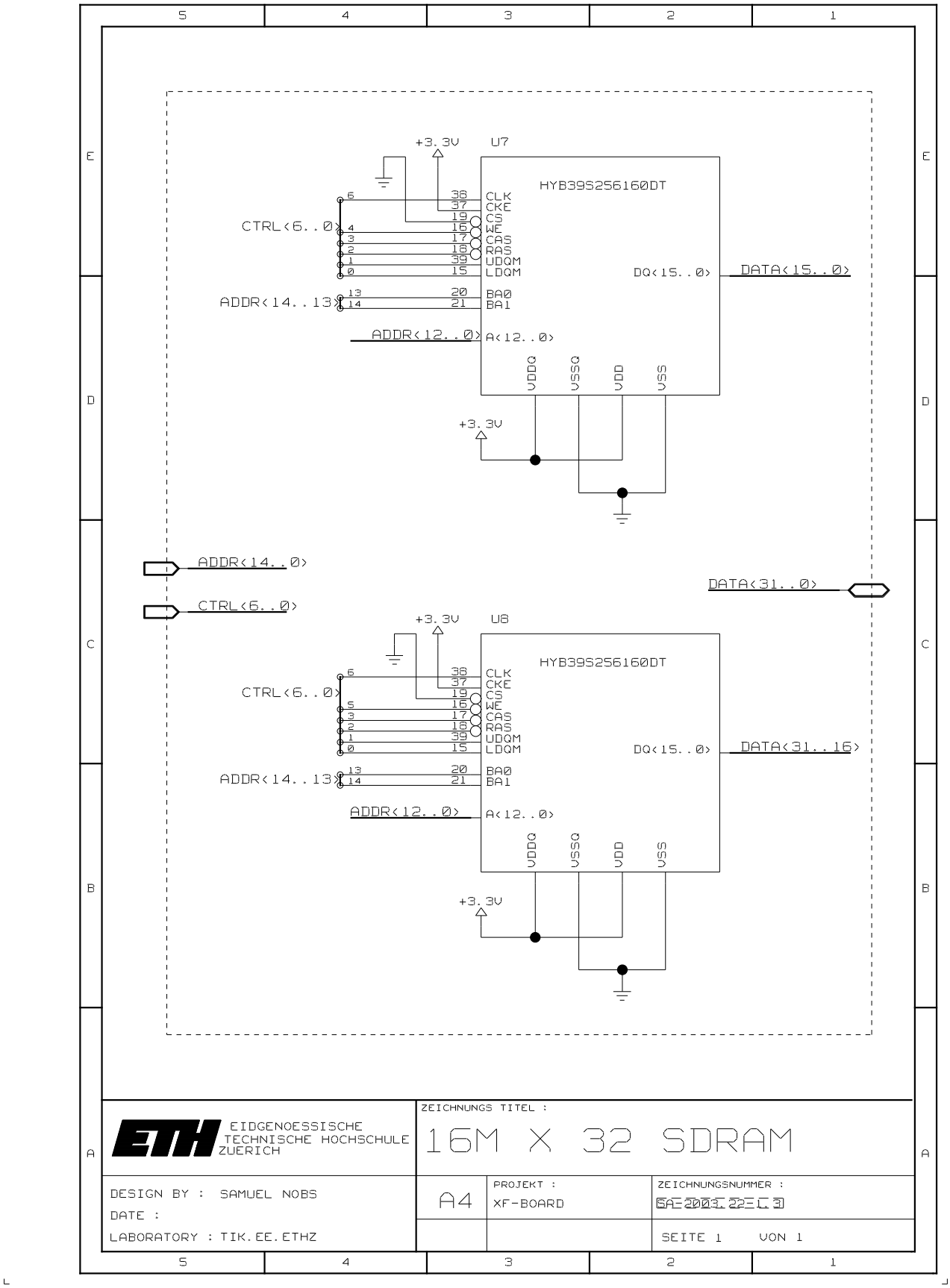

SRAM_32BIT
SDRAM_32BIT
FLASH_RAM
VGA_OUT_SIMPLE
SWITCHES
C_FPGA
XILINX XC2V1000
RS_232
KEYBOARD_MOUSE
KEYBOARD_MOUSE
ETHERNET
JTAG_IF
HEADER
EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH
XF-TOP
PROJEKT : XF-BOARD
A3
SEITE 1 VON 1
DESIGN BY : SAMUEL NOBS
LABORATORY : TIK.EE.ETHZ


ETH-MII_CTRL(3..0)
SRAM-ADDR(19)
SRAM-ADDR(18)
SRAM-ADDR(18..0)
VGA-RGB(2..0)
VGA-HSYNC
VGA-VSYNC
PS2-CLK1
PS2-DATA1
PS2-CLK2
PS2-DATA2
SRAM-WEB(3..0)
JTAG(1)
DONE
CCLK
RS-R1XD0
RS-T1XD1
RS-R2XD0
RS-T2XD1
SRAM-DATA(31..0)
C_LED0(1)
C_LED0(0)
ETH-MII_CTRL(7..0)
ETH-RESETB
ETH-MII_DATA(18..0)
SRAM-DATA(31..0)
XCBUS(323)
U2
XC2V1000
VERSORGUNG (INT)
GPIO(39..0)
TERMINATION
CLOCK

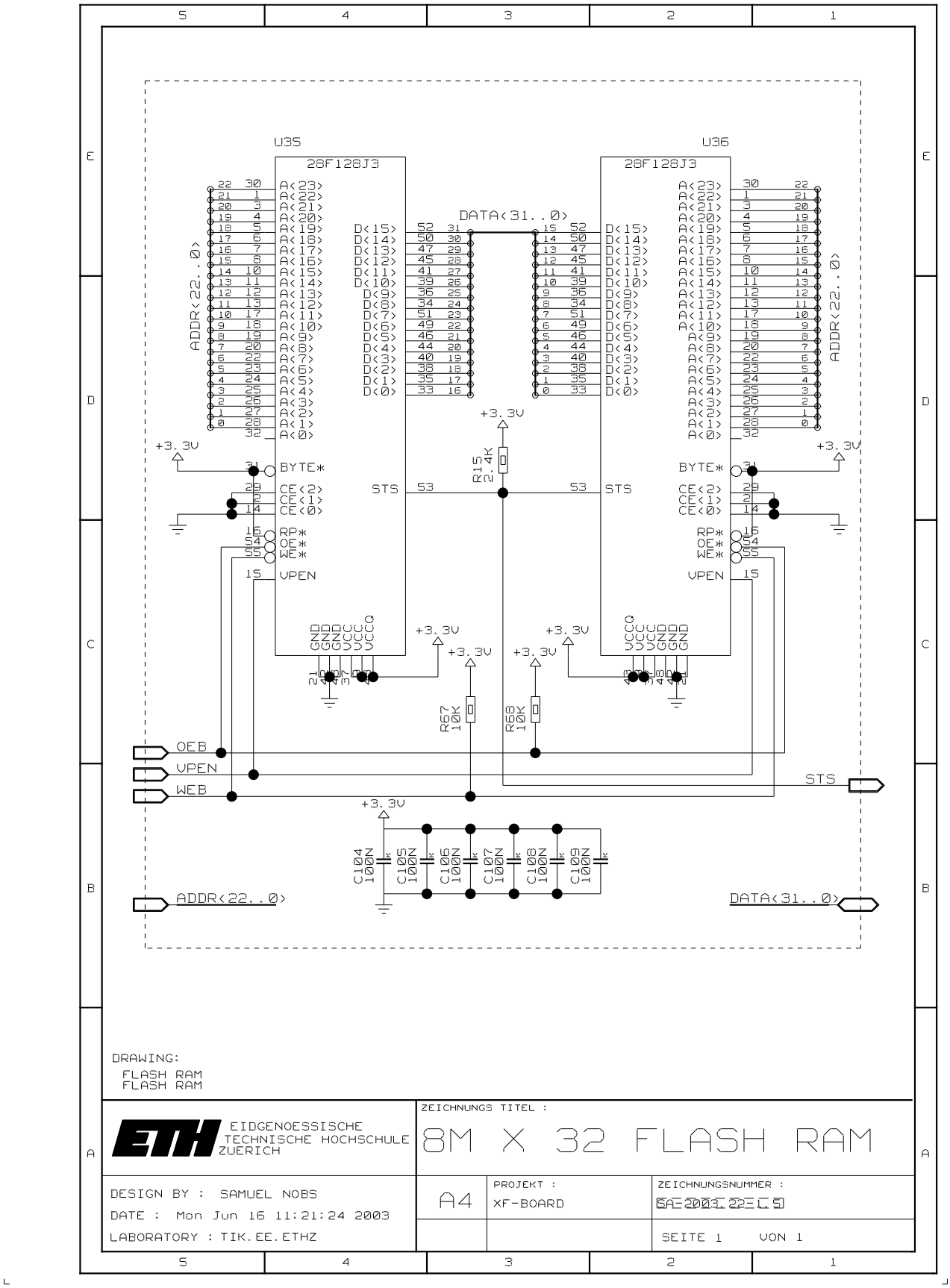

SDRAM-ADDR<14..0>
SDRAM-CTRL<6..0>
FLASH-OEB
FLASH-DATA<31..0>
XCBUS<23..0>
XCBUS<31..24>
OGPIO<31..0>
CLOCK
C_LED0<7..0>
C_LED0<7>
LED1B
LED2B
TEMP-CLK
TEMP-DATA
ADDRESS: 1A
ADDRESS: 4C
LED_DONE
DONE
PROGB
CCLK
INITB
JTAG-TDI-TDO
JTAG<0>
JTAG<3>
JTAG<2>
PROGRAM_RESET
FLASH-ADDR<22..0>
FLASH-STS
FLASH-WPEN
FLASH-WEB
SW<1..0>
HSWAP_EN
PWRDWN_B
EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH
C-FPGA
DESIGN BY : SAMUEL NOBS
DATE :
LABORATORY : TIK.EE.ETHZ
A1
PROJEKT : XF-BOARD
SEITE 1 VON 1


ETHERNET
EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH
DESIGN BY : SAMUEL NOBS
DATE :
LABORATORY : TIK.EE.ETHZ
PROJEKT : XF-BOARD
A3
SEITE 1 VON 1
LXT970A
RJ45
25MHZ
Mon Jun 16 11:21:03 2003
DRAWING:
ETH
ETH


ZEICHNUNGS TITEL :
1M X 32 SRAM
EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH
A3
PROJEKT : XF-BOARD
ZEICHNUNGSNUMMER :
SEITE 1 VON 1
DESIGN BY : SAMUEL NOB
DATE :
LABORATORY : TIK.EE.ETHZ
ADDRESS MSB = 1
ADDRESS MSB = 0
DATA<31..0>
AS7C34096
U21
U22
U23
U24
U25
U26
U27
U29
BANK1
BANK2
+3.3V
ADDR_19B
ADDR_19
WEB<3..0>
ADDR<18..0>
C72
C73
C74
C75
C76
C77

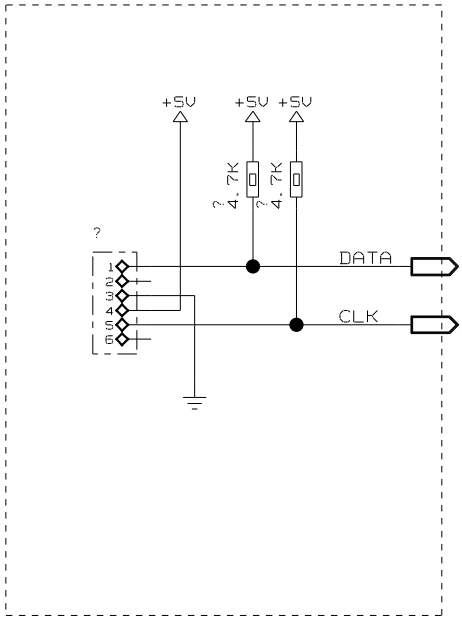


EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH	ZEICHNUNGS TITEL : 16M X 32 SDRAM		
DESIGN BY : SAMUEL NOBS DATE : LABORATORY : TIK.EE.ETHZ	A4	PROJEKT : XF-BOARD	ZEICHNUNGSNUMMER : [illegible]
			SEITE 1 VON 1



DRAWING:
FLASH RAM
FLASH RAM

EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH	ZEICHNUNGS TITEL : 8M X 32 FLASH RAM		
DESIGN BY : SAMUEL NOBS DATE : Mon Jun 16 11:21:24 2003 LABORATORY : TIK.EE.ETHZ	A4	PROJEKT : XF-BOARD	ZEICHNUNGSNUMMER : [illegible]
			SEITE 1 VON 1



DRAWING:
KEYBOARD/MOUSE
KEYBOARD/MOUSE

EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH

ZEICHNUNGS TITEL :
KEYBOARD AND MOUSE

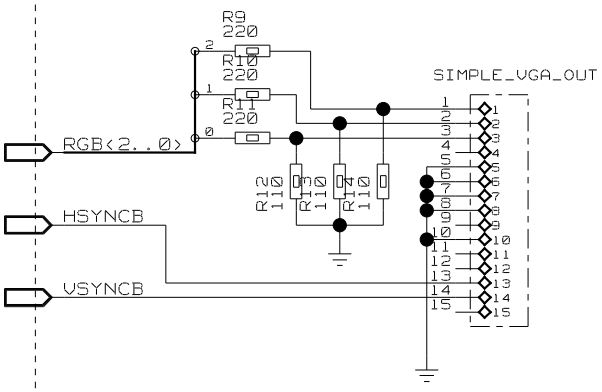
DESIGN BY : SAMUEL NOBS
DATE : Mon Jun 16 11:21:20 2003
LABORATORY : TIK.EE.ETHZ

A4

PROJEKT :
XF-BOARD

ZEICHNUNGSNUMMER :
SA-2003-22-C-6

SEITE 1 VON 1

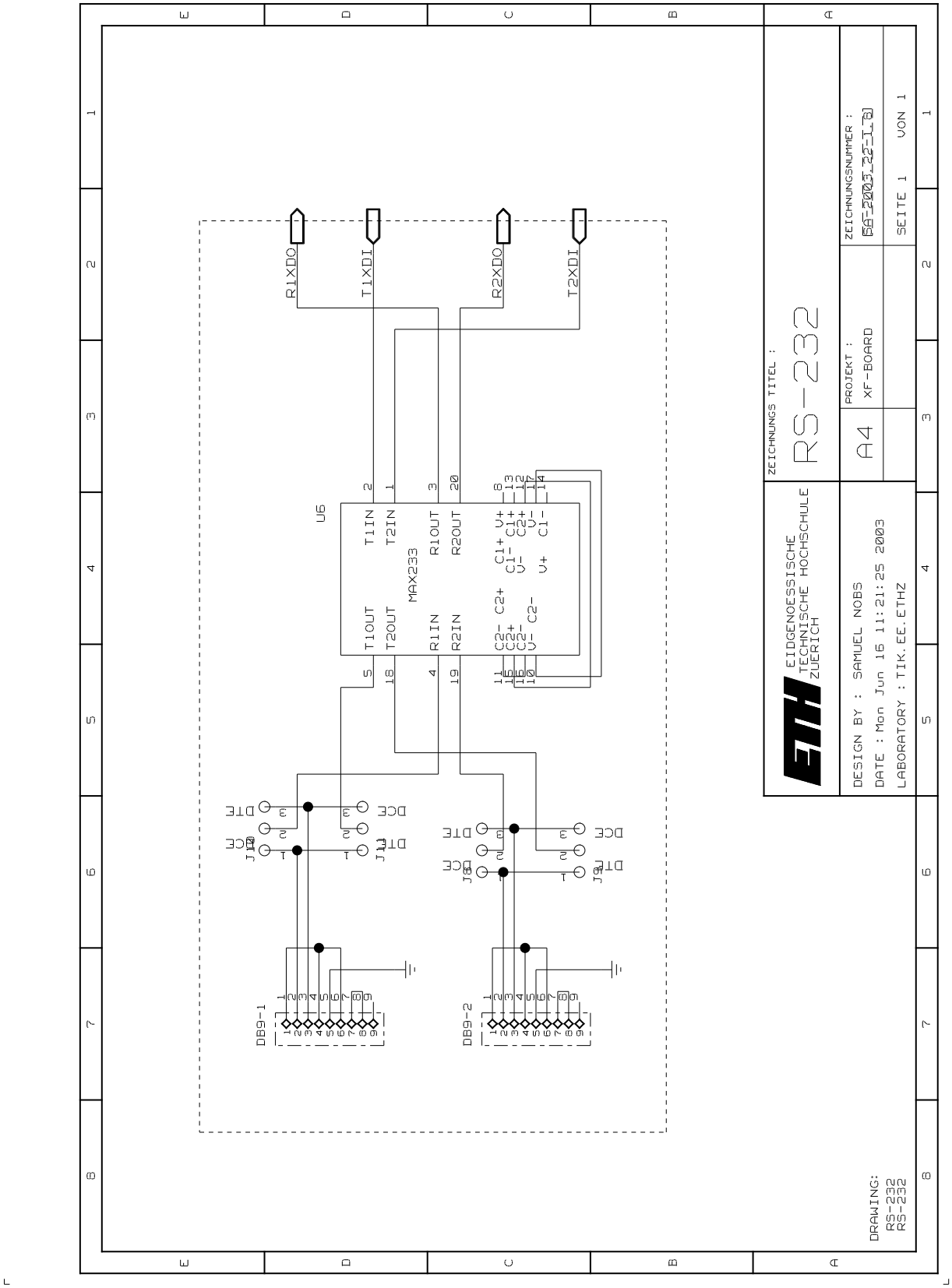


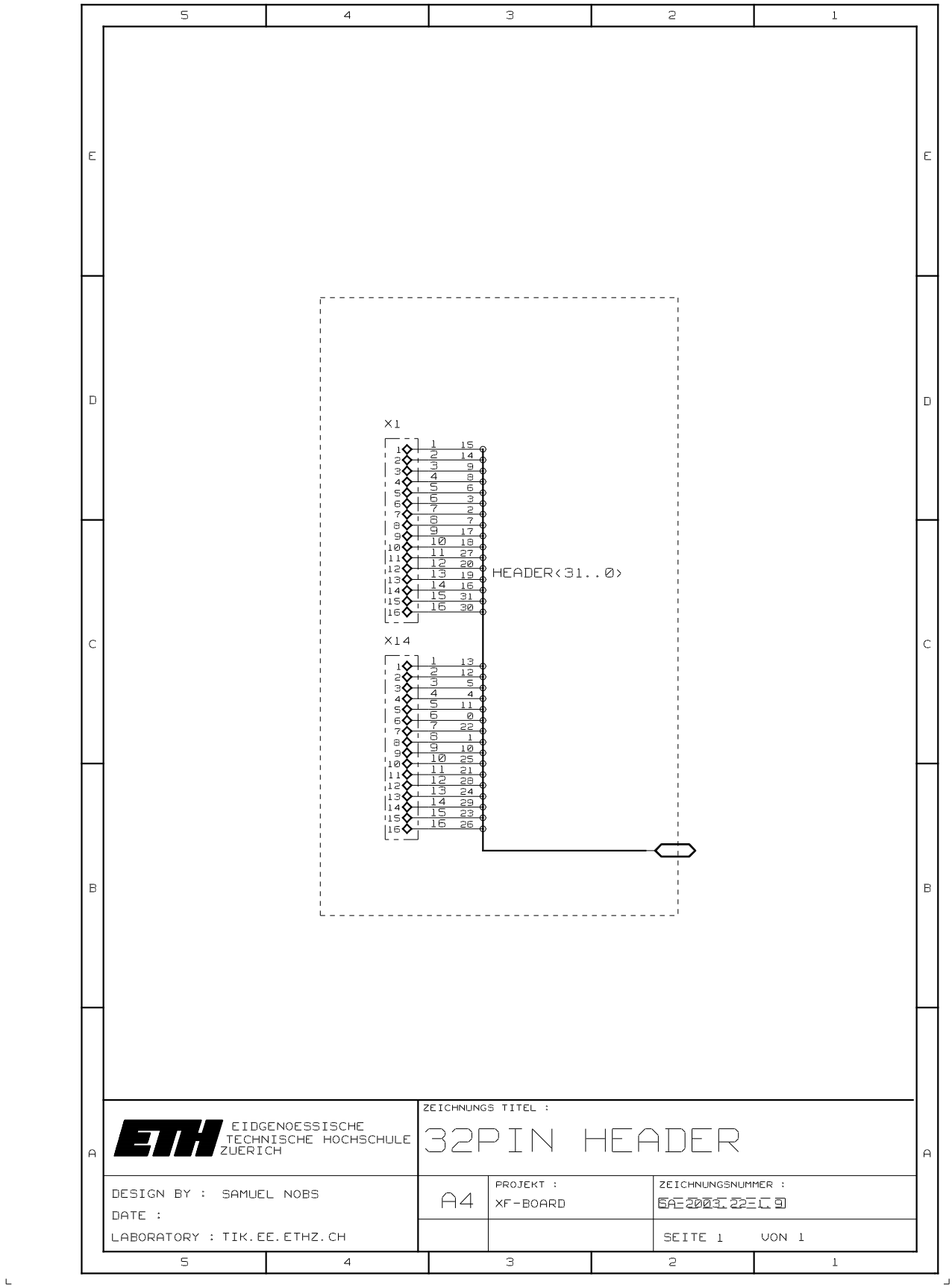
EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH

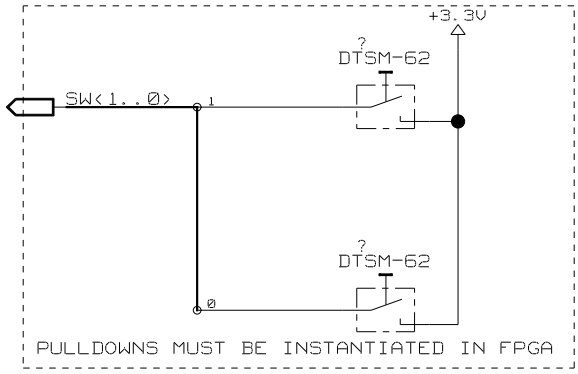
ZEICHNUNGS TITEL :

SIMPLE VGA OUT

DESIGN BY : SAMUEL NOBS DATE : LABORATORY : TIK.EE.ETHZ	A4	PROJEKT : XF-BOARD	ZEICHNUNGSNUMMER : SA-2003.22-1.7
			SEITE 1 VON 1


R1XDO
T1XDI
R2XDO
T2XDI
U6
MAX233
T1IN
T2IN
R1OUT
R2OUT
T1OUT
T2OUT
R1IN
R2IN
C1+ V+
C1- C1+
V- C2+
V- C1-
C2- C2+
C2+
C2-
V- C2-
V+
DB9-1
DB9-2
DTE
DCE
ZEICHNUNGS TITEL :
RS-232
EIDGENOESSISCHE
TECHNISCHE HOCHSCHULE
ZUERICH
DESIGN BY : SAMUEL NOBS
DATE : Mon Jun 16 11:21:25 2003
LABORATORY : TIK.EE.ETHZ
A4
PROJEKT :
XF-BOARD
ZEICHNUNGSNUMMER :
SEITE 1 VON 1
DRAWING:
RS-232
RS-232


X1
1 15
2 14
3 9
4 8
5 6
6 3
7 2
8 7
9 17
10 18
11 27
12 20
13 19
14 16
15 31
16 30
HEADER<31..0>
X14
1 13
2 12
3 5
4 4
5 11
6 0
7 22
8 1
9 10
10 25
11 21
12 28
13 24
14 29
15 23
16 26
EIDGENOESSISCHE
TECHNISCHE HOCHSCHULE
ZUERICH
ZEICHNUNGS TITEL :
32PIN HEADER
DESIGN BY : SAMUEL NOBS
DATE :
LABORATORY : TIK.EE.ETHZ.CH
A4
PROJEKT :
XF-BOARD
ZEICHNUNGSNUMMER :
SEITE 1 VON 1



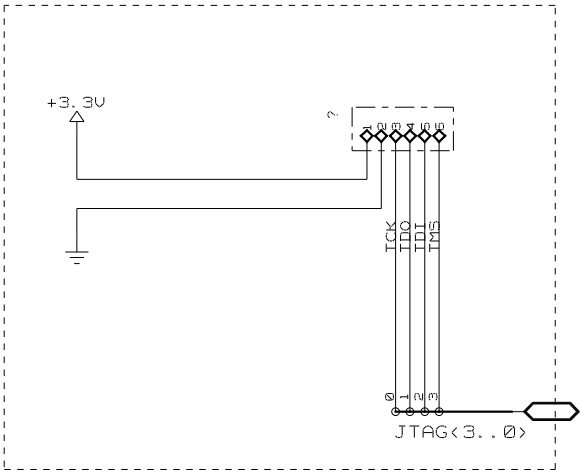
EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH

ZEICHNUNGS TITEL : SWITCHES

DESIGN BY : SAMUEL NOBS
DATE :
LABORATORY : TIK.EE.ETHZ

A4 | PROJEKT : XF-BOARD | ZEICHNUNGSNUMMER : SA-2003.22-1.10

SEITE 1 VON 1



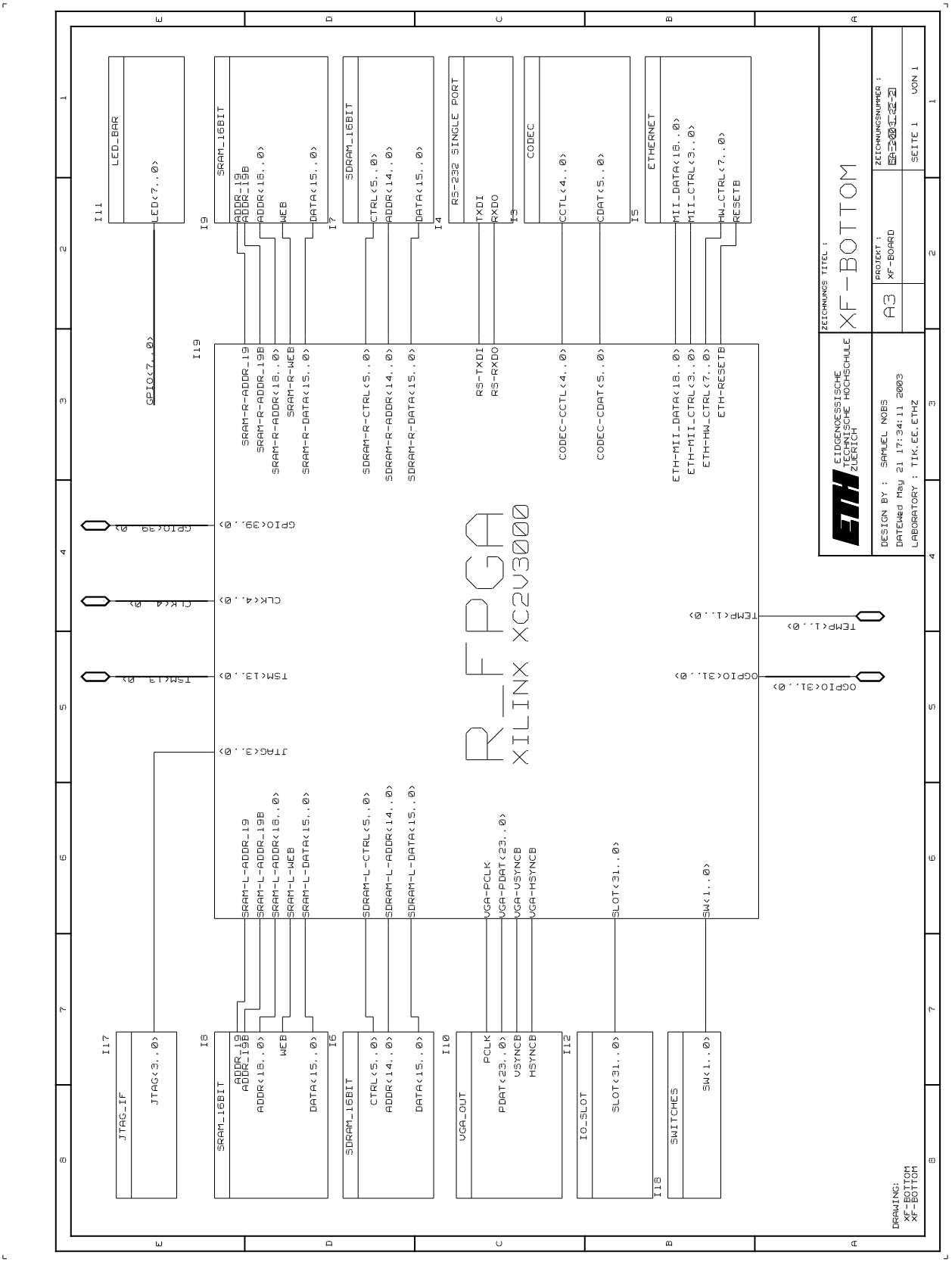
EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH

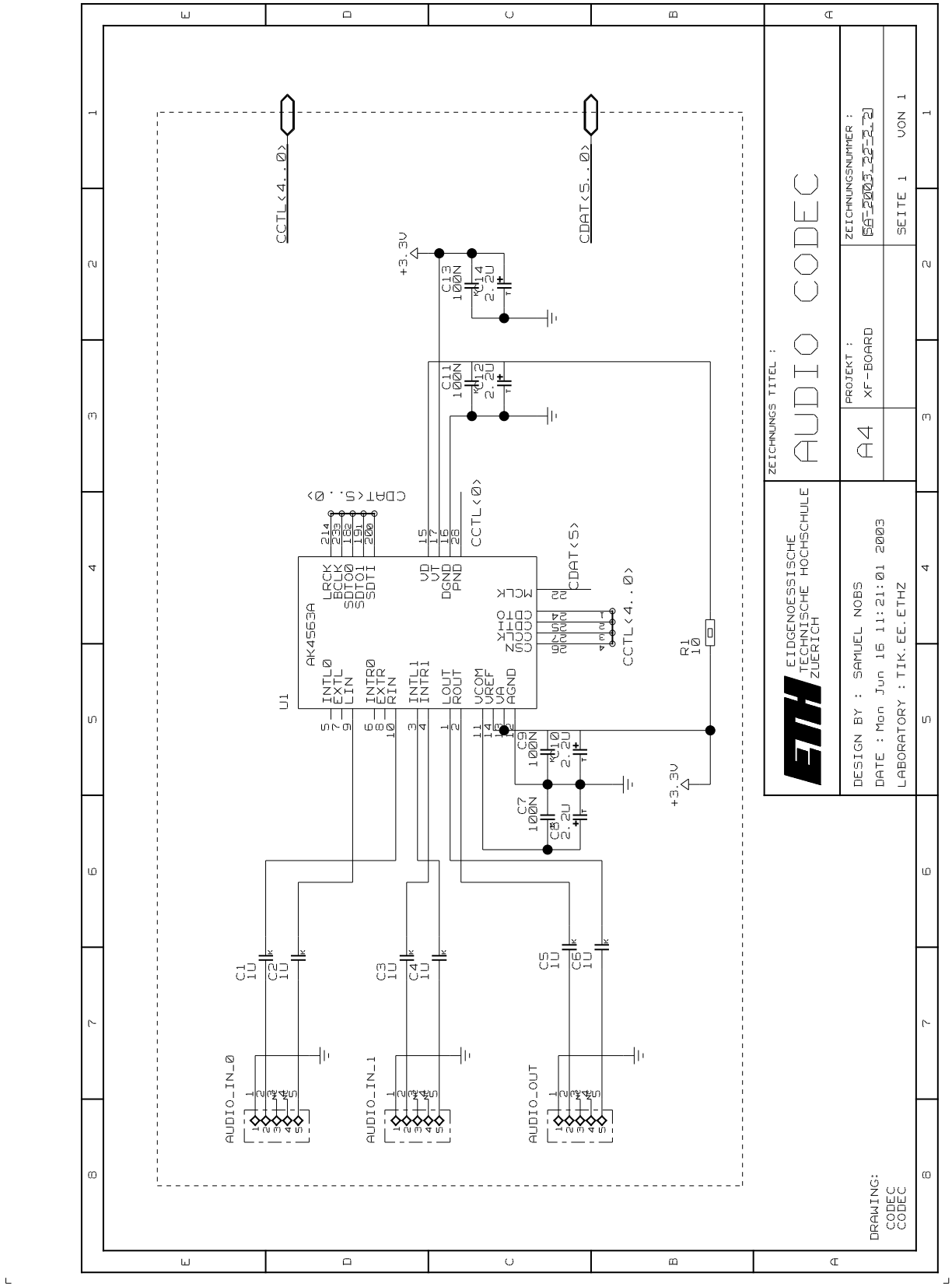
ZEICHNUNGS TITEL :

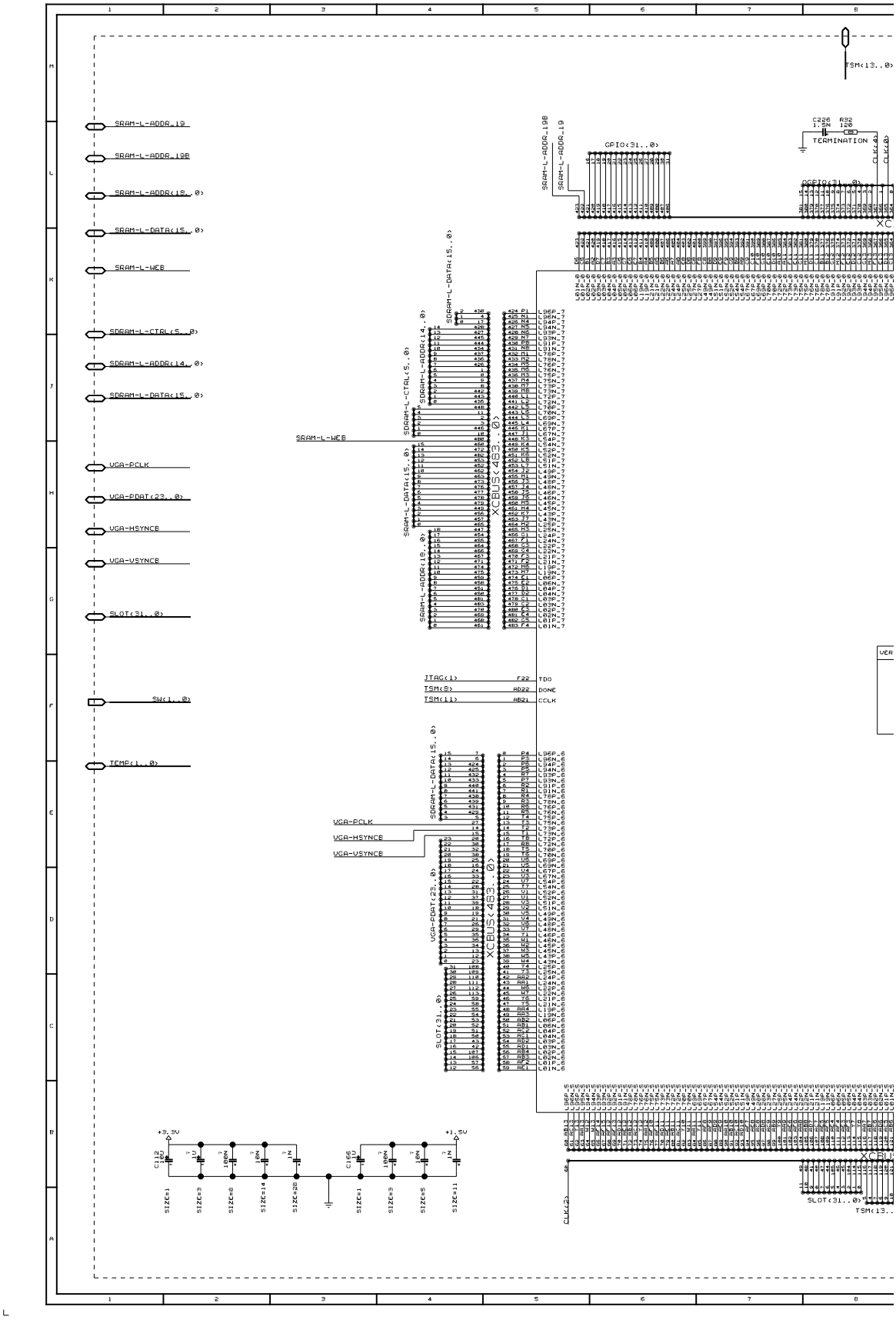
JTAG INTERFACE

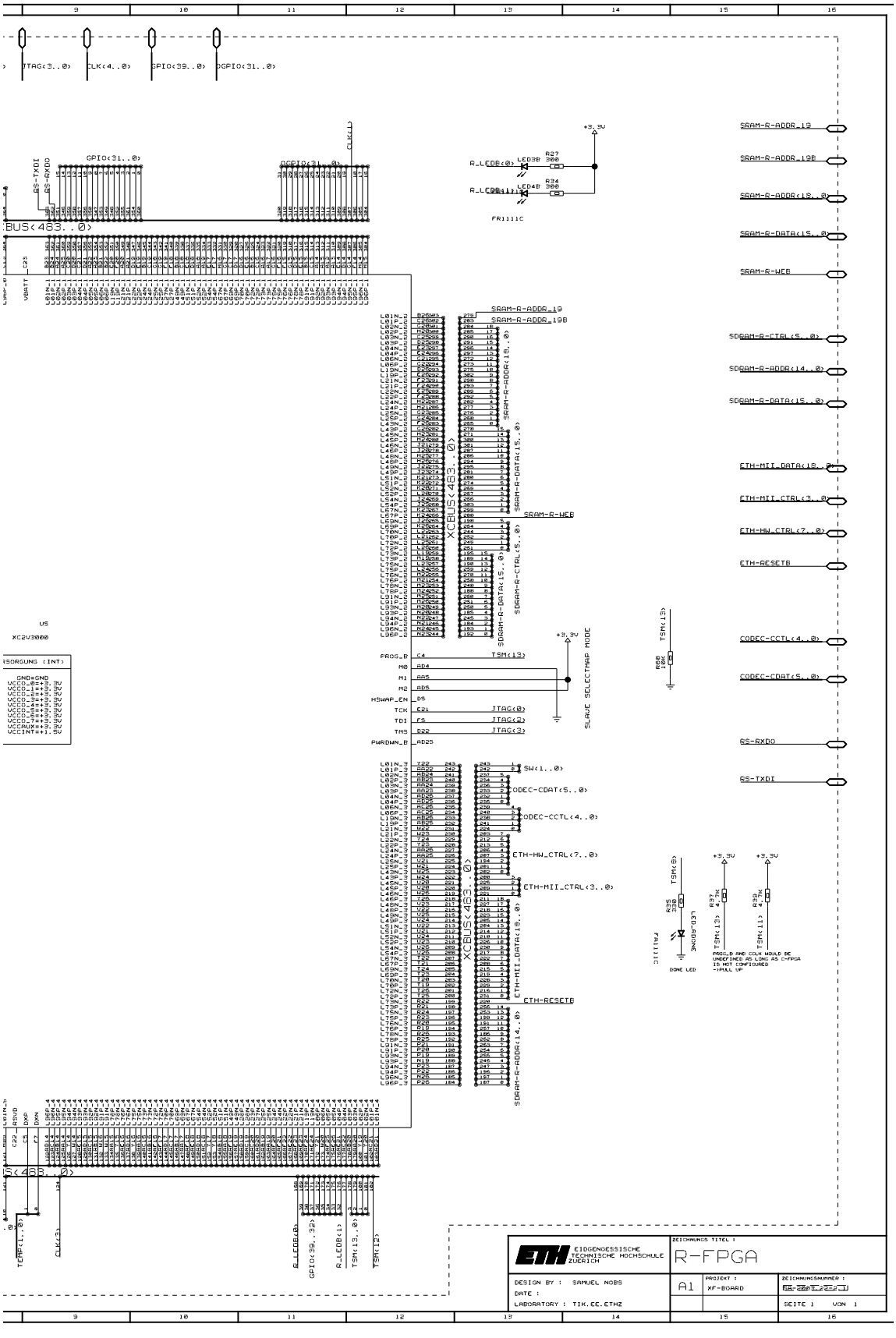
DESIGN BY : SAMUEL NOBS
DATE :
LABORATORY : TIK.EE.ETHZ

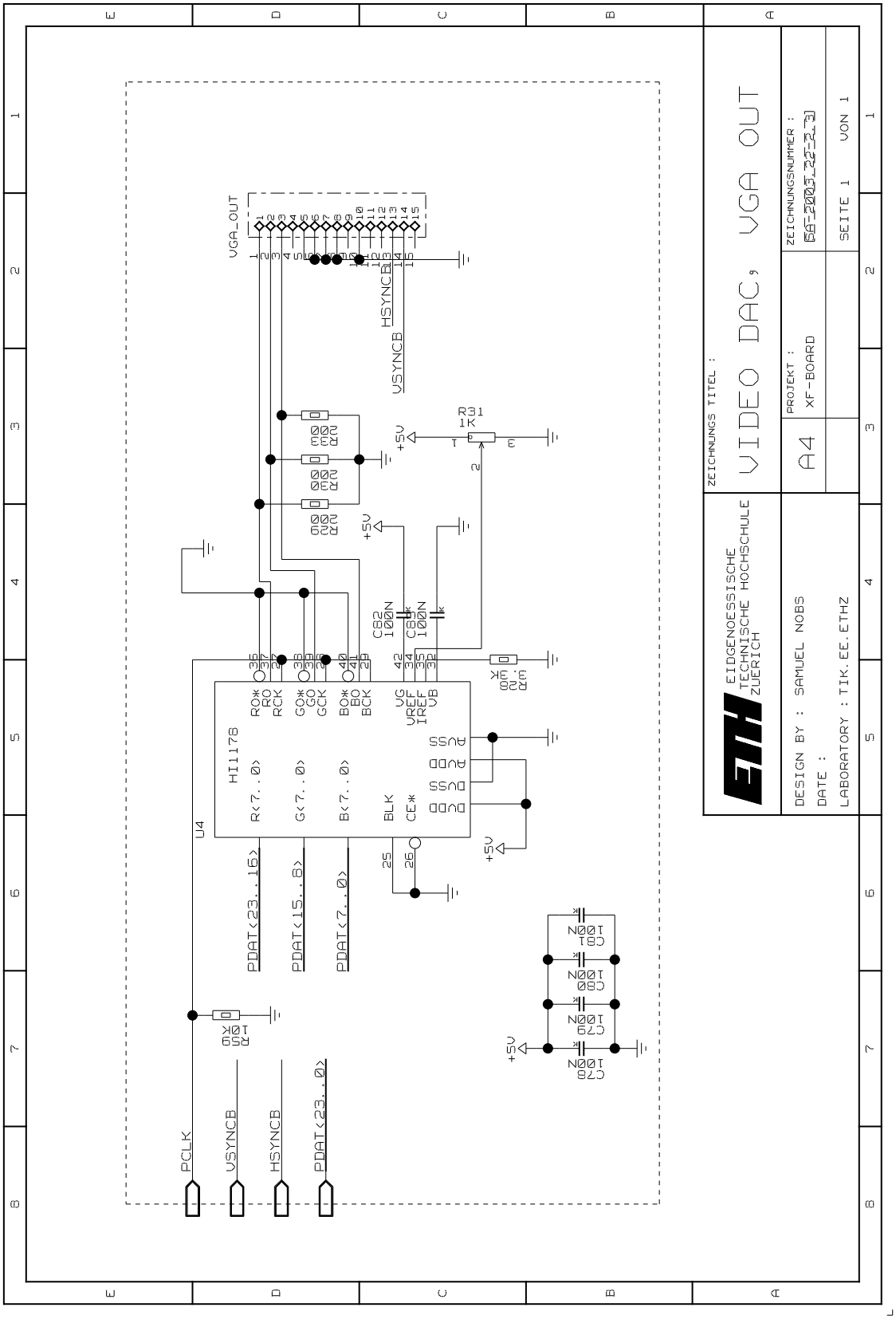
A4	PROJEKT : XFBOARD	ZEICHNUNGSNUMMER : SA-2003.22-1.11
		SEITE 1 VON 1

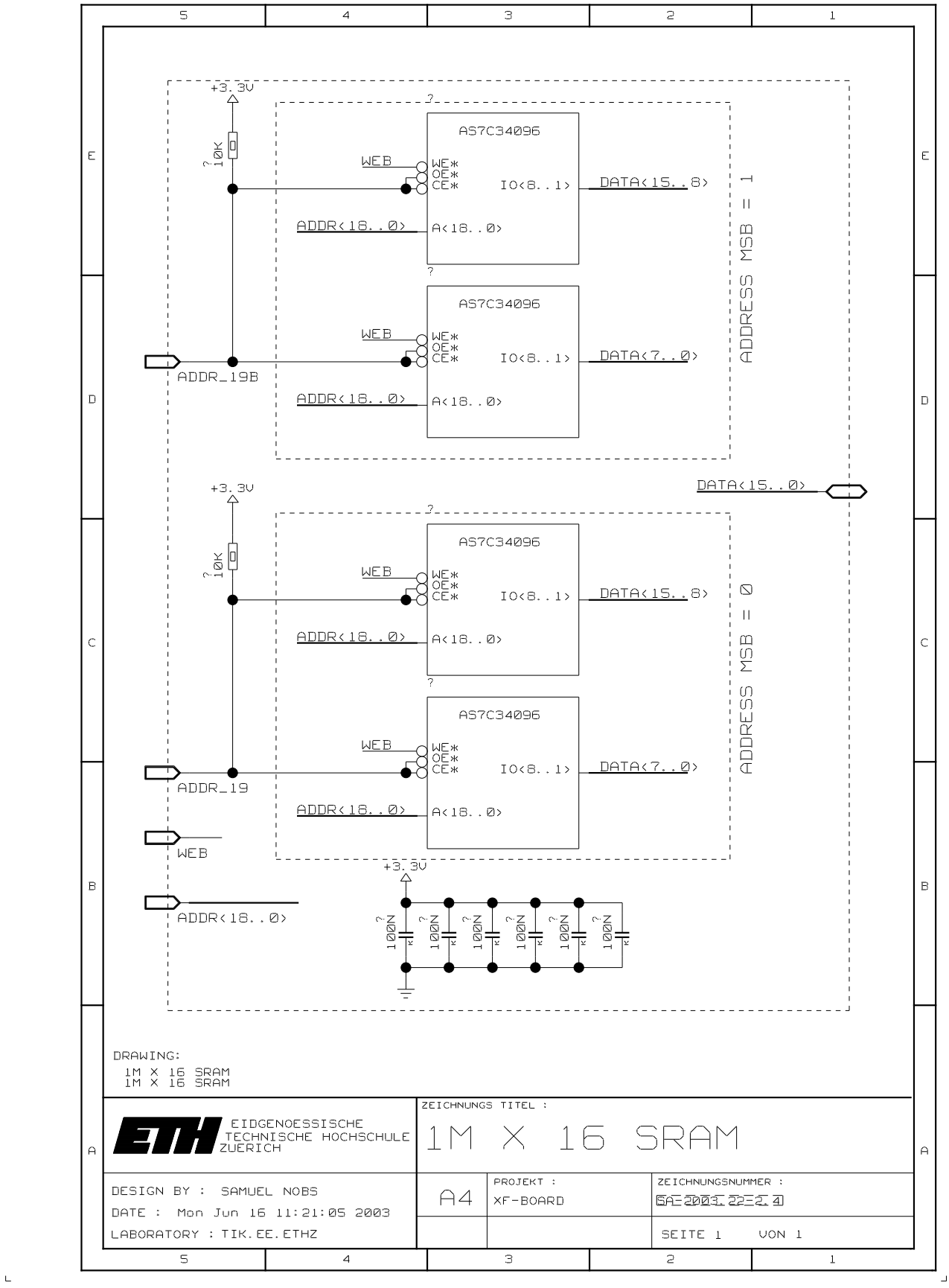

LED_BAR
SRAM_16BIT
SDRAM_16BIT
RS-232 SINGLE PORT
CODEC
ETHERNET
R_FPGA
XILINX XC2V3000
JTAG_IF
VGA_OUT
IO_SLOT
SWITCHES
EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH
DESIGN BY : SAMUEL NOBS
DATE:Wed May 21 17:34:11 2003
LABORATORY : TIK.EE.ETHZ
ZEICHNUNGS TITEL :
XF-BOTTOM
PROJEKT : XF-BOARD
A3
SEITE 1 VON 1
DRAWING:
XF-BOTTOM
XF-BOTTOM

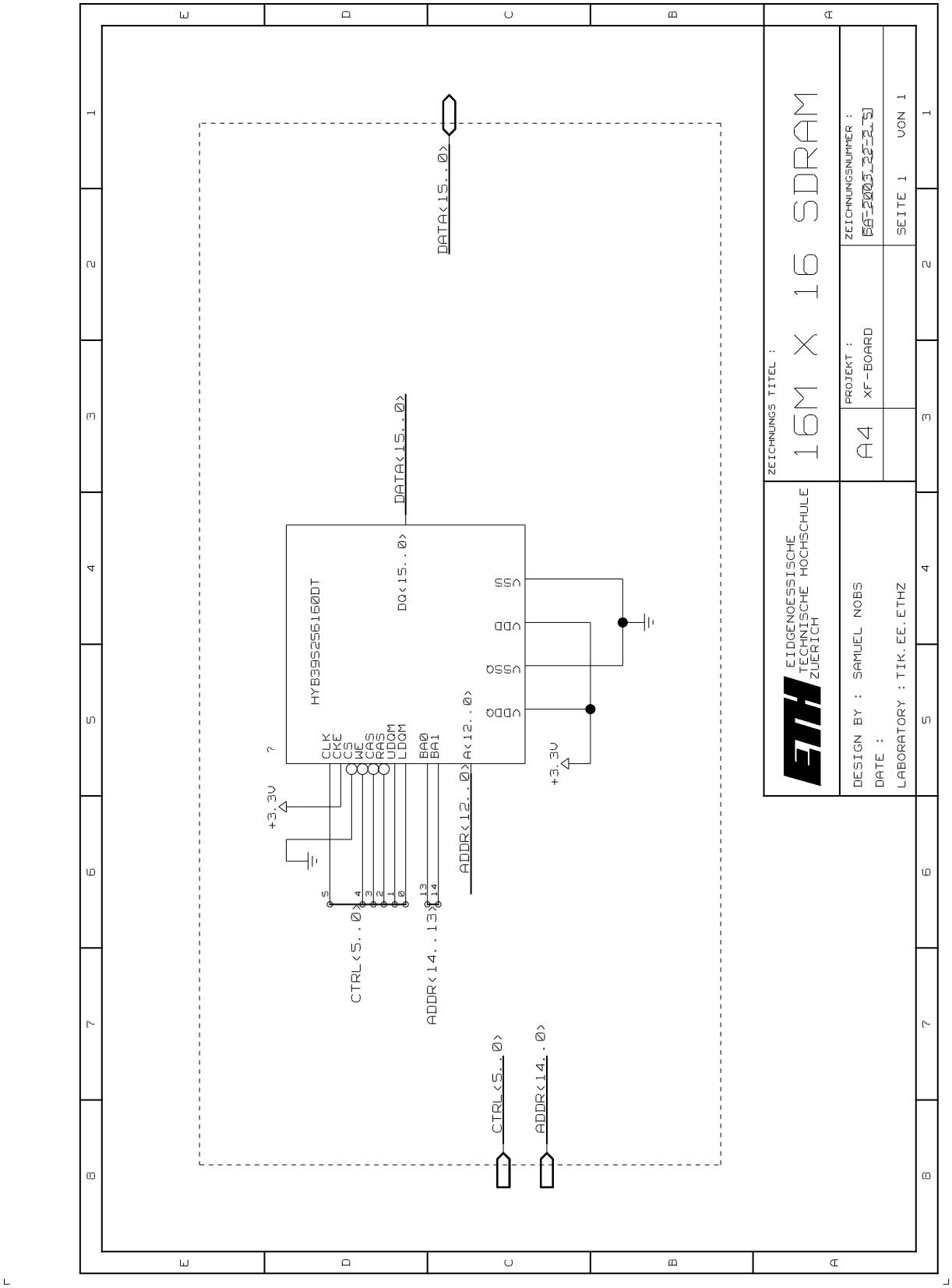

CCTL<4..0>
CDAT<5..0>
+3.3V
C13
100N
C14
2.2U
C11
100N
C12
2.2U
CDAT<5..0>
CCTL<0>
U1
AK4563A
LRCK
BCLK
SDTO0
SDTO1
SDTI
VD
VT
DGND
PND
INTL0
EXTL
LIN
INTR0
EXTR
RIN
INTL1
EXTR1
LOUT
ROUT
VCOM
VREF
VA
AGND
XTI
XTO
MCKO
MCKI
CDTO
CDTI
CCTL
CCLK
CDAT
CCTL<4..0>
R1
10
C9
100N
C10
2.2U
C7
100N
C8
2.2U
+3.3V
C1
1U
C2
1U
C3
1U
C4
1U
C5
1U
C6
1U
AUDIO_IN_0
AUDIO_IN_1
AUDIO_OUT
ZEICHNUNGS TITEL :
AUDIO CODEC
EIDGENOESSISCHE
TECHNISCHE HOCHSCHULE
ZUERICH
PROJEKT :
XF-BOARD
A4
ZEICHNUNGSNUMMER :
SEITE 1 VON 1
DESIGN BY : SAMUEL NOBS
DATE : Mon Jun 16 11:21:01 2003
LABORATORY : TIK.EE.ETHZ
DRAWING:
CODEC
CODEC

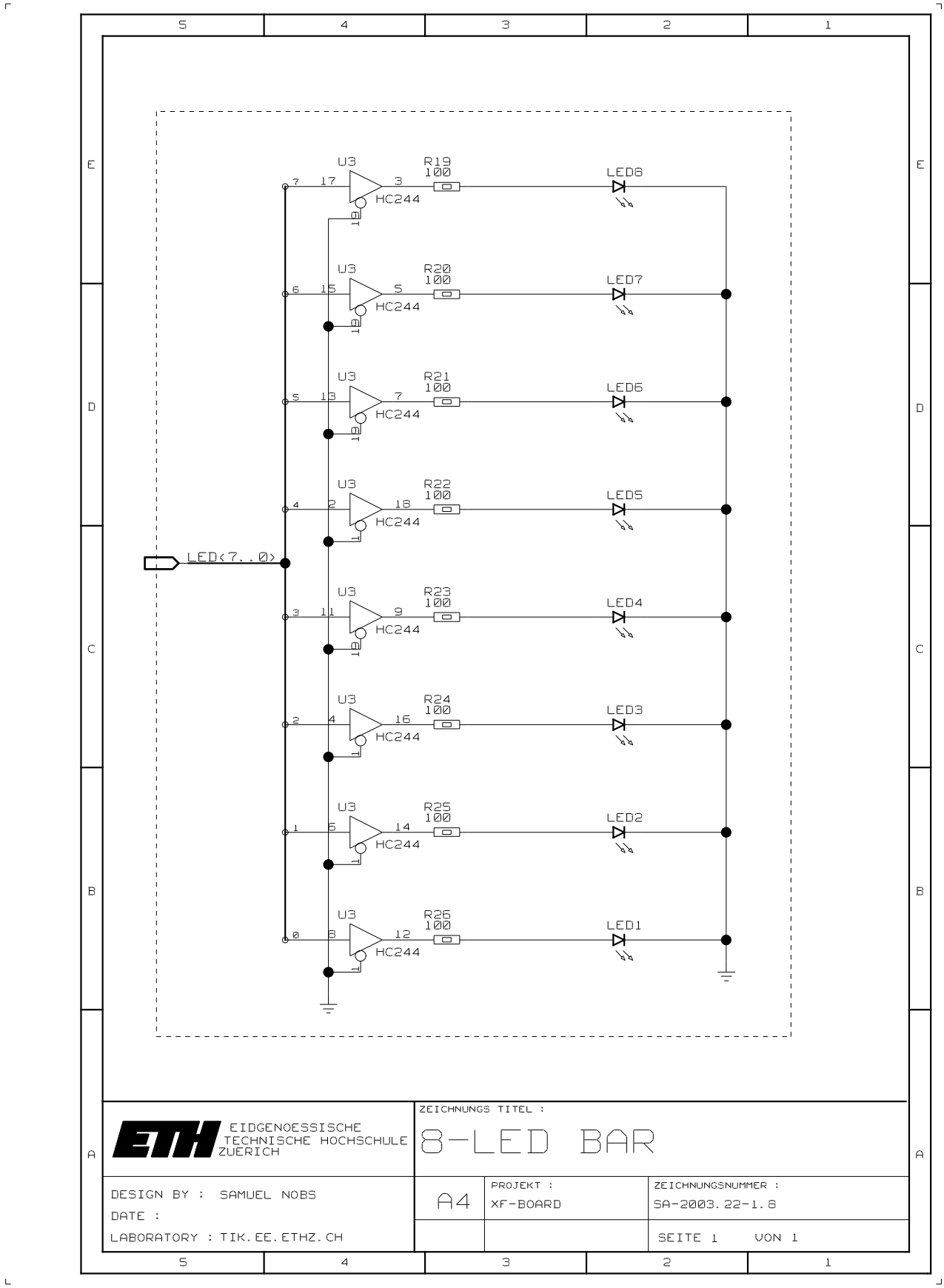

SRAM-L-ADDR.19
SRAM-L-ADDR.19B
SRAM-L-ADDR<18..0>
SRAM-L-DATA<15..0>
SRAM-L-WEB
SDRAM-L-CTRL<5..0>
SDRAM-L-ADDR<14..0>
SDRAM-L-DATA<15..0>
VGA-PCLK
VGA-PDAT<23..0>
VGA-HSYNCE
VGA-VSYNCE
SLOT<31..0>
SW<1..0>
TEMP<1..0>
GPIO<31..0>
TERMINATION
JTAG<1>
TSM<5>
TSM<11>
TDO
DONE
CCLK
+3.3V
+1.5V

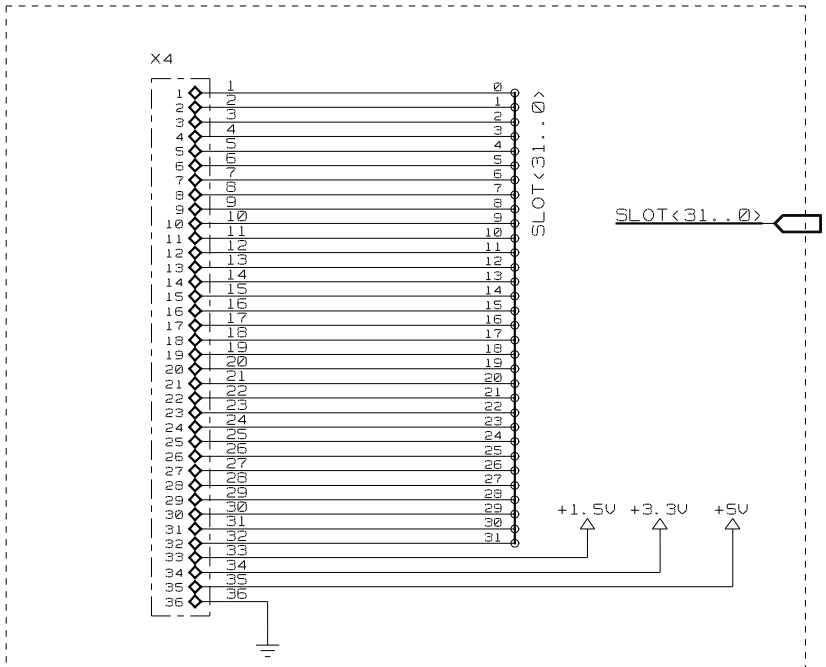

JTAG<3..0>
CLK<4..0>
GPIO<39..0>
DGPIO<31..0>
RS-TXDI
RS-RXDO
GPIO<31..0>
DGPIO<31..0>
CLK<1>
XCBUS<483..0>
R_LED0<0>
R_LED0<1>
LED3B
LED4B
R27 300
R34 300
+3.3V
FA1111C
SRAM-R-ADDR_19
SRAM-R-ADDR_19B
SRAM-R-ADDR<18..0>
SRAM-R-DATA<15..0>
SRAM-R-WEB
SDRAM-R-CTRL<5..0>
SDRAM-R-ADDR<14..0>
SDRAM-R-DATA<15..0>
ETH-MII_DATA<15..0>
ETH-MII_CTRL<3..0>
ETH-HW_CTRL<7..0>
ETH-RESETB
CODEC-CCTL<4..0>
CODEC-CDAT<5..0>
SW<1..0>
SDRAM-R-ADDR<14..0>
SDRAM-R-DATA<15..0>
SDRAM-R-CTRL<5..0>
SDRAM-R-ADDR<14..0>
SDRAM-R-ADDR<18..0>
SLAVE SELECTMAP MODE
PROG_B
M0
M1
M2
HSWAP_EN
TCK
TDI
TMS
PWRDWN_B
TSM<13>
TSM<6>
TSM<12>
TSM<11>
R35 68K
R37 4.7K
R39 4.7K
LED_BOARD
BOARD LED
PROG_B AND CCLK WOULD BE UNDEFINED AS LONG AS C-FPGA IS NOT CONFIGURED -PULL UP
U5
XC2V3000
TEMP<1..0>
CLK<3>
R_LED0<0>
GPIO<39..32>
R_LED0<1>
TSM<13..0>
TSM<12>
CIDGENOSSISCHE TECHNISCHE HOCHSCHULE ZUERICH
DESIGN BY : SAMUEL NOBS
DATE :
LABORATORY : TIK.EE.ETHZ
ZEICHNUNGS TITEL :
R-FPGA
A1
PROJEKT :
XF-BOARD
ZEICHNUNGSNUMMER :
SEITE 1 VON 1


VIDEO DAC, VGA OUT
EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH
ETH
ZEICHNUNGS TITEL :
PROJEKT : XF-BOARD
A4
ZEICHNUNGSNUMMER :
SEITE 1 VON 1
DESIGN BY : SAMUEL NOBS
DATE :
LABORATORY : TIK.EE.ETHZ
VGA_OUT
HSYNCB
VSYNCB
HI1178
U4
PCLK
PDAT<23..16>
PDAT<15..8>
PDAT<7..0>
+5V


+3.3V
?
10K
AS7C34096
WEB
WE*
OE*
CE*
IO<8..1>
DATA<15..8>
ADDR<18..0>
A<18..0>
AS7C34096
DATA<7..0>
ADDR_19B
ADDRESS MSB = 1
DATA<15..0>
ADDRESS MSB = 0
ADDR_19
WEB
ADDR<18..0>
100N
DRAWING:
1M X 16 SRAM
1M X 16 SRAM
EIDGENOESSISCHE
TECHNISCHE HOCHSCHULE
ZUERICH
ZEICHNUNGS TITEL :
1M X 16 SRAM
DESIGN BY : SAMUEL NOBS
DATE : Mon Jun 16 11:21:05 2003
LABORATORY : TIK.EE.ETHZ
A4
PROJEKT :
XF-BOARD
ZEICHNUNGSNUMMER :
SEITE 1 VON 1

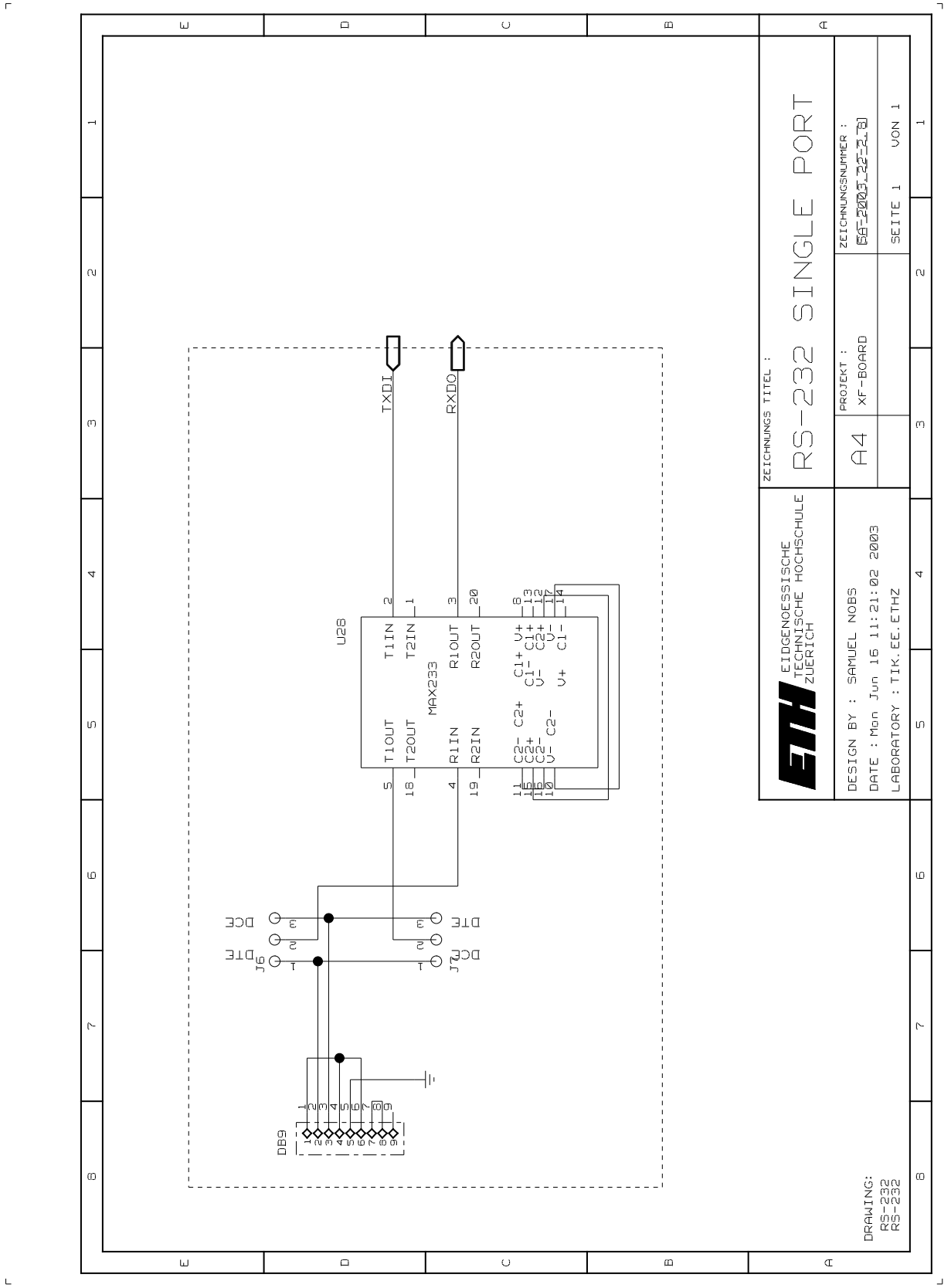

DATA<15..0>
DATA<15..0>
HYB39S256160DT
DQ<15..0>
CLK
CKE
CS
WE
CAS
RAS
UDQM
LDQM
BA0
BA1
A<12..0>
+3.3V
+3.3V
CTRL<5..0>
ADDR<14..13>
ADDR<12..0>
CTRL<5..0>
ADDR<14..0>
ZEICHNUNGS TITEL :
16M X 16 SDRAM
EIDGENOESSISCHE
TECHNISCHE HOCHSCHULE
ZUERICH
A4
PROJEKT :
XF-BOARD
ZEICHNUNGSNUMMER :
SEITE 1 VON 1
DESIGN BY : SAMUEL NOBS
DATE :
LABORATORY : TIK.EE.ETHZ

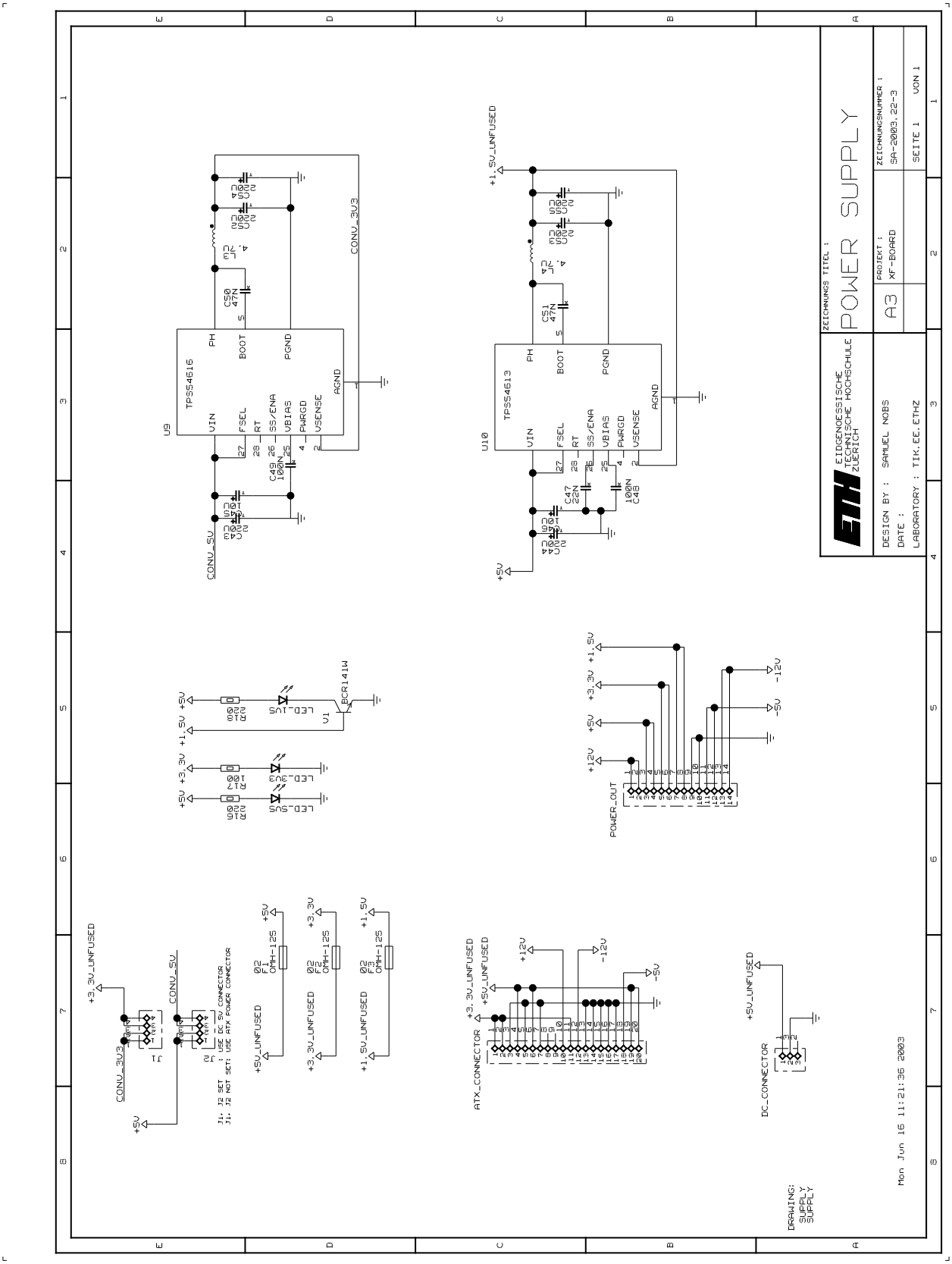

U3
HC244
R19
100
LED8
R20
100
LED7
R21
100
LED6
R22
100
LED5
R23
100
LED4
R24
100
LED3
R25
100
LED2
R26
100
LED1
LED<7..0>
EIDGENOESSISCHE
TECHNISCHE HOCHSCHULE
ZUERICH
ZEICHNUNGS TITEL :
8-LED BAR
DESIGN BY : SAMUEL NOBS
DATE :
LABORATORY : TIK.EE.ETHZ.CH
A4
PROJEKT :
XF-BOARD
ZEICHNUNGSNUMMER :
SA-2003.22-1.8
SEITE 1 VON 1



DRAWING:
IO SLOT
IO SLOT

EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH	ZEICHNUNGS TITEL : I/O SLOT		
DESIGN BY : SAMUEL NOBS DATE : Mon Jun 16 11:21:09 2003 LABORATORY : TIK.EE.ETHZ.CH	A4	PROJEKT : XF-BOARD	ZEICHNUNGSNUMMER : SA-2003.22-2.7
			SEITE 1 VON 1


TXDI
RXDO
U28
MAX233
T1IN
T2IN
R1OUT
R2OUT
T1OUT
T2OUT
R1IN
R2IN
C1+
C1-
C2+
C2-
V+
V-
DB9
J6
J7
DTE
DCE
ZEICHNUNGS TITEL :
RS-232 SINGLE PORT
EIDGENOESSISCHE
TECHNISCHE HOCHSCHULE
ZUERICH
A4
PROJEKT :
XF-BOARD
ZEICHNUNGSNUMMER :
SEITE 1 VON 1
DESIGN BY : SAMUEL NOBS
DATE : Mon Jun 16 11:21:02 2003
LABORATORY : TIK.EE.ETHZ
DRAWING:
RS-232
RS-232


EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH
ZEICHNUNGS TITEL : POWER SUPPLY
DESIGN BY : SAMUEL NOBS
DATE :
LABORATORY : TIK.EE.ETHZ
A3
PROJEKT : XF-BOARD
ZEICHNUNGSNUMMER : SA-2003.22-3
SEITE 1 VON 1
TPS54616
TPS54613
U9
U10
CONV_5V
CONV_3V3
+1.5V_UNFUSED
+5V_UNFUSED
+3.3V_UNFUSED
BCR141W
POWER_OUT
ATX_CONNECTOR
DC_CONNECTOR
J1: J2 SET
J1: J2 NOT SET: USE ATX POWER CONNECTOR
J1 : USE DC 5V CONNECTOR
Mon Jun 16 11:21:36 2003

Appendix B

Pin Tables

The pin tables explain which pins of the I/O devices and of other components are connected to which pins of the FPGAs. This information may be useful when writing your own constraints file (`*.ucf`).

B.1 Pin Tables for the C-FPGA

Table B-1: Pin mapping for the Clock Input

BGA Name	I/O Name	CLOCK Pin
B11	L96N_0	System Clock, 50MHz

Table B-2: Pin mapping for the Clock Output

BGA Name	I/O Name	CLK Pin
Y12	L95P_4	Clock 0, to R-FPGA
V15	L19P_4	Clock 1, to R-FPGA
Y2	L02P_6	Clock 2, to R-FPGA
Y1	L02N_6	Clock 3, to R-FPGA

Table B-3: Pin mapping for the Control LEDs

BGA Name	I/O Name	C_LEDB Pin
U5	L01N_6	$\overline{\texttt{LED1}}$
V5	L01P_6	$\overline{\texttt{LED2}}$

Table B-4: Pin mapping for the Ethernet Transceiver

BGA Name	I/O Name	ETH Pin
A8	L49P_0	CFG(0)
E10	L91N_0	CFG(1)

Table B-4: Pin mapping for the Ethernet Transceiver (cont'd)

BGA Name	I/O Name	ETH Pin
C2	L02N_7	COL
B4	L01N_0	CRS
B8	L49N_0	FDE
A4	L01P_0	FDS/MDINT
E9	L51N_0	MDC
B9	L54N_0	MDDIS
F10	L91P_0	MDIO
A7	L22P_0	MF(0)
B7	L22N_0	MF(1)
A6	L05P_0	MF(2)
B6	L05N_0	MF(3)
A5	L03P_0	MF(4)
A9	L54P_0	$\overline{\text{RESET}}$
E6	L01N_7	RXD(0)
E5	L01P_7	RXD(1)
E7	L06N_0	RXD(2)
E8	L06P_0	RXD(3)
F9	L51P_0	RXD(4)
C5	L02P_0	RX_CLK
C4	L02N_0	RX_DV
D6	L04N_0	RX_ER
B5	L03N_0	TRSTE
C8	L24P_0	TXD(0)
D9	L52N_0	TXD(1)
C9	L52P_0	TXD(2)
C1	L02P_7	TXD(3)
C7	L21P_0	TXD(4)
D7	L21N_0	TX_CLK
D8	L24N_0	TX_EN
C6	L04P_0	TX_ER

Table B-5: Pin mapping for the FlashRAM

BGA Name	I/O Name	FLASH Pin
K20	L54P_2	A(10)
L19	L94N_2	A(11)
M19	L94N_3	A(12)
M20	L96P_3	A(13)
F18	L02P_2	A(14)
G18	L22N_2	A(15)
H18	L22P_2	A(16)
J17	L48N_2	A(17)
J18	L48P_2	A(18)
K17	L52N_2	A(19)
F19	L19N_2	A(1)

Table B-5: Pin mapping for the FlashRAM (cont'd)

BGA Name	I/O Name	FLASH Pin
K18	L52P_2	A(20)
L17	L93N_2	A(21)
L18	L93P_2	A(22)
G21	L43N_2	A(23)
F20	L19P_2	A(2)
G19	L24N_2	A(3)
G20	L24P_2	A(4)
H19	L45N_2	A(5)
H20	L45P_2	A(6)
J19	L49N_2	A(7)
J20	L49P_2	A(8)
K19	L54N_2	A(9)
J21	L51N_2	D(10)
K21	L91N_2	D(11)
L21	L96N_2	D(12)
M21	L96N_3	D(13)
N22	L91N_3	D(14)
P22	L51N_3	D(15)
R22	L48N_3	D(16)
N19	L54P_3	D(17)
P19	L49P_3	D(18)
R19	L46P_3	D(19)
G22	L43P_2	D(1)
T19	L24P_3	D(20)
U19	L21P_3	D(21)
U22	L22N_3	D(22)
E21	L06N_2	D(23)
F21	L21N_2	D(24)
N20	L54N_3	D(25)
P20	L49N_3	D(26)
R20	L46N_3	D(27)
T20	L24N_3	D(28)
U20	L21N_3	D(29)
H22	L46P_2	D(2)
V22	L06N_3	D(30)
E22	L06P_2	D(31)
F22	L21P_2	D(32)
J22	L51P_2	D(3)
K22	L91P_2	D(4)
L22	L96P_2	D(5)
N21	L91P_3	D(6)
P21	L51P_3	D(7)
R21	L48P_3	D(8)
H21	L46N_2	D(9)
T22	L43N_3	$\overline{OE}$

Table B-5: Pin mapping for the FlashRAM (cont'd)

BGA Name	I/O Name	FLASH Pin
T21	L43P_3	STS
L20	L94P_2	VPEN
U21	L22P_3	$\overline{WE}$

Table B-6: Pin mapping for the GPIO and LED-Bar

BGA Name	I/O Name	GPIO Pin
AA17	L06N_4	GPIO(0) / LED 1
Y16	L21P_4	GPIO(10)
W15	L24N_4	GPIO(11)
Y15	L24P_4	GPIO(12)
W14	L52N_4	GPIO(13)
W17	L05N_4	GPIO(14)
V17	L02P_4	GPIO(15)
AA5	L04P_5	GPIO(16)
AB5	L04N_5	GPIO(17)
Y7	L21N_5	GPIO(18)
W8	L49P_5	GPIO(19)
AA16	L22N_4	GPIO(1) / LED 2
AA6	L19P_5	GPIO(20)
AB6	L19N_5	GPIO(21)
Y8	L49N_5	GPIO(22)
W9	L52P_5	GPIO(23)
V6	L05P_5	GPIO(24)
W6	L06P_5	GPIO(25)
AA7	L22P_5	GPIO(26)
AB7	L22N_5	GPIO(27)
AA8	L51P_5	GPIO(28)
AB8	L51N_5	GPIO(29)
AB17	L06P_4	GPIO(2) / LED 3
AA9	L54P_5	GPIO(30)
AB9	L54N_5	GPIO(31)
V21	L06P_3	GPIO(32)
Y22	L02N_3	GPIO(33)
Y21	L02P_3	GPIO(34)
AA20	L01P_3	GPIO(35)
AB19	L01N_4	GPIO(36)
W21	L03P_3	GPIO(37)
W22	L03N_3	GPIO(38)
V20	L04N_3	GPIO(39)
AA15	L49N_4	GPIO(3) / LED 4
AB16	L22P_4	GPIO(4) / LED 5
AA14	L54N_4	GPIO(5) / LED 6
AB14	L54P_4	GPIO(6) / LED 7
AB15	L49P_4	GPIO(7) / LED 8

Table B-6: Pin mapping for the GPIO and LED-Bar (cont'd)

BGA Name	I/O Name	GPIO Pin
Y14	L52P_4	GPIO(8)
W18	L03N_4	GPIO(9)

Table B-7: Pin mapping for the OGPIO and Header

BGA Name	I/O Name	OGPIO Pin
Y11	L96P_5	OGPIO(0) / pin 22
AA11	L96N_5	OGPIO(10) / pin 25
AA10	L93P_5	OGPIO(11) / pin 21
W10	L92P_5	OGPIO(12) / pin 2
Y9	L52N_5	OGPIO(13) / pin 17
V7	L05N_5	OGPIO(14) / pin 2
W7	L21P_5	OGPIO(15) / pin 1
W16	L21N_4	OGPIO(16) / pin 14
V16	L19N_4	OGPIO(17) / pin 9
V12	L94P_4	OGPIO(18) / pin 10
W12	L95N_4	OGPIO(19) / pin 13
U11	L94N_5	OGPIO(1) / pin 24
V11	L95P_5	OGPIO(20) / pin 12
AB12	L96P_4	OGPIO(21) / pin 27
AB10	L93N_5	OGPIO(22) / pin 23
AB13	L93P_4	OGPIO(23) / pin 31
AA12	L96N_4	OGPIO(24) / pin 29
Y13	L92P_4	OGPIO(25) / pin 26
AA13	L93N_4	OGPIO(26) / pin 32
U12	L94N_4	OGPIO(27) / pin 11
W13	L92N_4	OGPIO(28) / pin 28
U14	L51N_4	OGPIO(29) / pin 30
V14	L51P_4	OGPIO(2) / pin 7
V13	L91P_4	OGPIO(30) / pin 16
U13	L91N_4	OGPIO(31) / pin 15
V9	L91P_5	OGPIO(3) / pin 6
W11	L95N_5	OGPIO(4) / pin 20
Y10	L92N_5	OGPIO(5) / pin 19
V10	L91N_5	OGPIO(6) / pin 5
U10	L94P_5	OGPIO(7) / pin 8
V8	L24P_5	OGPIO(8) / pin 4
U9	L24N_5	OGPIO(9) / pin 3

Table B-8: Pin mapping for the PS/2 Ports

BGA Name	I/O Name	PS2 Pin
L6	L54P_7	CLK port 1
K5	L52P_7	CLK port 2

Table B-8: Pin mapping for the PS/2 Ports (cont'd)

BGA Name	I/O Name	PS2 Pin
L5	L94N_7	DATA port 1
K6	L54N_7	DATA port 2

Table B-9: Pin mapping for the RS-232 Interface

BGA Name	I/O Name	RS Pin
M18	L94P_3	R1
M17	L93N_3	R2
N17	L93P_3	T1
N18	L52N_3	T2

Table B-10: Pin mapping for the SDRAM bank

BGA Name	I/O Name	SDRAM Pin
C17	L04N_1	A(0)
D16	L21P_1	A(10)
E14	L91P_1	A(11)
F13	L96P_1	A(12)
D17	L04P_1	A(1)
C18	L02N_1	A(2)
D18	L02P_1	A(3)
D21	L03N_2	A(4)
E18	L02N_2	A(5)
E17	L06P_1	A(6)
E16	L06N_1	A(7)
E15	L24P_1	A(8)
F14	L24N_1	A(9)
D15	L49P_1	BA(0)
C16	L21N_1	BA(1)
D14	L52P_1	$\overline{\text{CAS}}$
E13	L91N_1	CLK
A10	L93P_0	DQ(0)
D11	L95N_0	DQ(10)
C10	L92P_0	DQ(11)
D10	L92N_0	DQ(12)
E12	L95N_1	DQ(13)
E11	L94N_0	DQ(14)
F11	L94P_0	DQ(15)
C21	L01N_2	DQ(16)
C22	L01P_2	DQ(17)
E20	L04P_2	DQ(18)
E19	L04N_2	DQ(19)
A11	L96P_0	DQ(1)
B19	L01P_1	DQ(20)

Table B-10: Pin mapping for the SDRAM bank (cont'd)

BGA Name	I/O Name	SDRAM Pin
A19	L01N_1	DQ(21)
B18	L03P_1	DQ(22)
A18	L03N_1	DQ(23)
B17	L05P_1	DQ(24)
A17	L05N_1	DQ(25)
B16	L22P_1	DQ(26)
A16	L22N_1	DQ(27)
B15	L51P_1	DQ(28)
A15	L51N_1	DQ(29)
B12	L94P_1	DQ(2)
B14	L54P_1	DQ(30)
A14	L54N_1	DQ(31)
A13	L93N_1	DQ(3)
B13	L93P_1	DQ(4)
B10	L93N_0	DQ(5)
D12	L95P_1	DQ(6)
C13	L92N_1	DQ(7)
C12	L94N_1	DQ(8)
C11	L95P_0	DQ(9)
C14	L52N_1	LDQM
C15	L49N_1	$\overline{\texttt{RAS}}$
F12	L96N_1	UDQM
D13	L92P_1	$\overline{\texttt{WE}}$ (0)
D22	L03P_2	$\overline{\texttt{WE}}$ (1)

Table B-11: Pin mapping for the SelectMap connection to R-FPGA

BGA Name	I/O Name	TSM Pin
U18	L19P_3	CCLK
AB4	L02N_5	$\overline{\texttt{CS}}$
Y18	L03P_4	D(0)
AA18	L04N_4	D(1)
AB18	L04P_4	D(2)
Y17	L05P_4	D(3)
Y4	L01N_5	D(4)
AA3	L01P_5	D(5)
W5	L03P_5	D(6)
Y5	L03N_5	D(7)
W20	L01N_3	DONE
V19	L04P_3	$\overline{\texttt{INIT}}$
Y6	L06N_5	$\overline{\texttt{PROG}}$
AA4	L02P_5	$\overline{\texttt{RDWR}}$

Table B-12: Pin mapping for the Simple VGA Output

BGA Name	I/O Name	VGA Pin
F4	L21N_7	BO
J5	L52N_7	GO
J6	L45P_7	$\overline{\texttt{HSYNC}}$
G5	L19P_7	RO
H5	L45N_7	$\overline{\texttt{VSYNC}}$

Table B-13: Pin mapping for the SRAM bank

BGA Name	I/O Name	SRAM Pin
J2	L51N_7	A(0)
R1	L46N_6	A(10)
R2	L46P_6	A(11)
T1	L43N_6	A(12)
T2	L43P_6	A(13)
U1	L21N_6	A(14)
V1	L19N_6	A(15)
V2	L19P_6	A(16)
W1	L04N_6	A(17)
W2	L04P_6	A(18)
P2	L51P_6	A(19)
J1	L51P_7	A(1)
K2	L93N_7	A(2)
K1	L93P_7	A(3)
L2	L96P_7	A(4)
M1	L96N_6	A(5)
M2	L96P_6	A(6)
N1	L91N_6	A(7)
N2	L91P_6	A(8)
P1	L51N_6	A(9)
U2	L21P_6	$\overline{\texttt{A}}$ (19)
M5	L93N_6	D(0)
H4	L46N_7	D(10)
H3	L46P_7	D(11)
J4	L49N_7	D(12)
J3	L49P_7	D(13)
K4	L91N_7	D(14)
K3	L91P_7	D(15)
L4	L94P_7	D(16)
L3	L96N_7	D(17)
M3	L94N_6	D(18)
M4	L94P_6	D(19)
M6	L93P_6	D(1)
N3	L54N_6	D(20)
N4	L54P_6	D(21)
P3	L49N_6	D(22)

Table B-13: Pin mapping for the SRAM bank (cont'd)

BGA Name	I/O Name	SRAM Pin
P4	L49P_6	D(23)
T4	L24P_6	D(24)
T3	L24N_6	D(25)
R4	L45P_6	D(26)
R3	L45N_6	D(27)
U3	L06N_6	D(28)
U4	L06P_6	D(29)
N5	L52N_6	D(2)
V3	L03N_6	D(30)
V4	L03P_6	D(31)
N6	L52P_6	D(3)
P5	L48N_6	D(4)
P6	L48P_6	D(5)
R5	L22N_6	D(6)
T5	L22P_6	D(7)
G4	L24N_7	D(8)
G3	L24P_7	D(9)
G2	L43N_7	$\overline{\text{WE}}$ (0)
G1	L43P_7	$\overline{\text{WE}}$ (1)
H2	L48N_7	$\overline{\text{WE}}$ (2)
H1	L48P_7	$\overline{\text{WE}}$ (3)

Table B-14: Pin mapping for the Switches

BGA Name	I/O Name	SW Pin
F2	L22N_7	S1
F1	L22P_7	S2

Table B-15: Pin mapping for the Temperature sensors

BGA Name	I/O Name	TMP Pin
E1	L06P_7	CLK
E2	L06N_7	DATA

B.2 Pin Tables for the R-FPGA

Table B-16: Pin mapping for the Audio CoDec

BGA Name	I/O Name	CODEC Pin
AD25	L04P_3	BCLK
AB23	L02P_3	CCLK
AA23	L03P_3	CDTI
AB24	L02N_3	CDTO
AA24	L03N_3	CSN
AC25	L06P_3	LRCK
AD26	L04N_3	MCLK
W21	L25P_3	PDN
AC26	L06N_3	SDTI
AB26	L19N_3	SDTO(0)
AB25	L19P_3	SDTO(1)

Table B-17: Pin mapping for the Clock Inputs

BGA Name	I/O Name	CLK Pin
D13	L96N_0	Clock 0, from C-FPGA
G14	L95N_1	Clock 1, from C-FPGA
AB13	L96P_5	Clock 2, from C-FPGA
AB14	L95P_4	Clock 3, from C-FPGA
F13	L95N_0	Clock 4, from 50MHz Oscillator

Table B-18: Pin mapping for the Control LEDs

BGA Name	I/O Name	R_LEDB Pin
AF23	L21P_4	$\overline{\text{LED3}}$
AC22	L04N_4	$\overline{\text{LED4}}$

Table B-19: Pin mapping for the Ethernet Transceiver

BGA Name	I/O Name	ETH Pin
T19	L70P_3	CFG(0)
T26	L72N_3	CFG(1)
AA25	L24P_3	COL
W22	L21N_3	CRS
T20	L70N_3	FDE
V21	L25N_3	FDS/MDINT
U26	L54N_3	MDC
U20	L45N_3	MDDIS
T25	L72P_3	MDIO
R19	L76P_3	MF(0)

Table B-19: Pin mapping for the Ethernet Transceiver (cont'd)

BGA Name	I/O Name	ETH Pin
T22	L67N_3	MF(1)
T21	L67P_3	MF(2)
U22	L51N_3	MF(3)
U21	L51P_3	MF(4)
V20	L45P_3	$\overline{\text{RESET}}$
Y24	L22N_3	RXD(0)
Y23	L22P_3	RXD(1)
W26	L46N_3	RXD(2)
V25	L49N_3	RXD(3)
V26	L54P_3	RXD(4)
W24	L43P_3	RX_CLK
W23	L21P_3	RX_DV
V23	L48N_3	RX_ER
V22	L48P_3	TRSTE
T24	L69N_3	TXD(0)
W25	L43N_3	TXD(1)
Y26	L46P_3	TXD(2)
AA26	L24N_3	TXD(3)
U24	L52N_3	TXD(4)
U23	L52P_3	TX_CLK
T23	L69P_3	TX_EN
V24	L49P_3	TX_ER

Table B-20: Pin mapping for the GPIO and LED-Bar

BGA Name	I/O Name	GPIO Pin
A25	L02P_1	GPIO(0) / LED 1
D21	L04P_1	GPIO(10)
C21	L04N_1	GPIO(11)
D20	L03P_1	GPIO(12)
C20	L03N_1	GPIO(13)
E19	L22P_1	GPIO(14)
E20	L19N_1	GPIO(15)
B1	L02N_0	GPIO(16)
A2	L02P_0	GPIO(17)
D7	L03N_0	GPIO(18)
C7	L03P_0	GPIO(19)
A23	L05P_1	GPIO(1) / LED 2
B3	L04N_0	GPIO(20)
A3	L04P_0	GPIO(21)
G6	L05N_0	GPIO(22)
G7	L05P_0	GPIO(23)
E6	L06N_0	GPIO(24)
E7	L06P_0	GPIO(25)
B4	L19N_0	GPIO(26)

Table B-20: Pin mapping for the GPIO and LED-Bar (cont'd)

BGA Name	I/O Name	GPIO Pin
A4	L19P_0	GPIO(27)
B5	L21N_0	GPIO(28)
A5	L21P_0	GPIO(29)
A24	L02N_1	GPIO(2) / LED 3
B6	L22N_0	GPIO(30)
A6	L22P_0	GPIO(31)
AE21	L04P_4	GPIO(32)
AE26	L05N_4	GPIO(33)
AF25	L05P_4	GPIO(34)
W20	L06N_4	GPIO(35)
Y21	L06P_4	GPIO(36)
AE24	L19N_4	GPIO(37)
AF24	L19P_4	GPIO(38)
AE23	L21N_4	GPIO(39)
A22	L05N_1	GPIO(3) / LED 4
B22	L06P_1	GPIO(4) / LED 5
A21	L21P_1	GPIO(5) / LED 6
A20	L21N_1	GPIO(6) / LED 7
B21	L06N_1	GPIO(7) / LED 8
D19	L22N_1	GPIO(8)
F20	L19P_1	GPIO(9)

Table B-21: Pin mapping for the I/O Slot

BGA Name	I/O Name	SLOT Pin
Y8	L04N_5	pin 1
Y3	L25N_6	pin 10
AA4	L19P_6	pin 11
AA3	L19N_6	pin 12
AB4	L02P_6	pin 13
AB3	L02N_6	pin 14
AD7	L21P_5	pin 15
AC7	L21N_5	pin 16
AA2	L24P_6	pin 17
AA1	L24N_6	pin 18
AB2	L06P_6	pin 19
Y7	L04P_5	pin 2
AB1	L06N_6	pin 20
AC2	L04P_6	pin 21
AC1	L04N_6	pin 22
AD2	L03P_6	pin 23
AD1	L03N_6	pin 24
AF2	L01P_6	pin 25
AE1	L01N_6	pin 26
AF3	L05N_5	pin 27

Table B-21: Pin mapping for the I/O Slot (cont'd)

BGA Name	I/O Name	SLOT Pin
AE3	L05P_5	pin 28
AF4	L06N_5	pin 29
AA8	L22P_5	pin 3
AE4	L06P_5	pin 30
AF5	L19N_5	pin 31
AE5	L19P_5	pin 32
W7	L22N_6	pin 4
Y6	L21P_6	pin 5
AB8	L22N_5	pin 6
W6	L22P_6	pin 7
Y5	L21N_6	pin 8
Y4	L25P_6	pin 9

Table B-22: Pin mapping for the left SDRAM bank

BGA Name	I/O Name	SDRAM-L Pin
M6	L76N_7	A(0)
M5	L76P_7	A(10)
L3	L69P_7	A(11)
L4	L69N_7	A(12)
L6	L70N_7	A(1)
L5	L70P_7	A(2)
R4	L78P_6	A(3)
R3	L78N_6	A(4)
P4	L96P_6	A(5)
P3	L96N_6	A(6)
N4	L94P_7	A(7)
M3	L75P_7	A(8)
M4	L75N_7	A(9)
N5	L94N_7	BA(0)
N6	L93P_7	BA(1)
P6	L94P_6	$\overline{\text{CAS}}$
K3	L54P_7	CLK
R8	L72N_6	DQ(0)
M2	L78N_7	DQ(10)
M1	L78P_7	DQ(11)
N1	L96N_7	DQ(12)
P1	L96P_7	DQ(13)
R2	L91P_6	DQ(14)
R1	L91N_6	DQ(15)
R7	L93P_6	DQ(1)
P8	L91P_7	DQ(2)
P7	L93N_6	DQ(3)
N7	L93N_7	DQ(4)
N8	L91N_7	DQ(5)

Table B-22: Pin mapping for the left SDRAM bank (cont'd)

BGA Name	I/O Name	SDRAM-L Pin
M8	L73N_7	DQ(6)
M7	L73P_7	DQ(7)
L2	L72N_7	DQ(8)
L1	L72P_7	DQ(9)
R6	L76P_6	LDQM
P5	L94N_6	$\overline{\text{RAS}}$
K1	L67P_7	UDQM
R5	L76N_6	$\overline{\text{WE}}$

Table B-23: Pin mapping for the left SRAM bank

BGA Name	I/O Name	SRAM-L Pin
C6	L01P_0	A19
D6	L01N_0	$\overline{\text{A19}}$
H4	L45N_7	A(0)
E2	L06N_7	A(10)
E1	L06P_7	A(11)
F2	L21N_7	A(12)
F1	L24N_7	A(13)
G1	L24P_7	A(14)
H2	L25P_7	A(15)
H1	L49N_7	A(16)
J2	L49P_7	A(17)
J1	L67N_7	A(18)
G3	L22P_7	A(1)
G4	L22N_7	A(2)
F3	L21P_7	A(3)
F4	L01N_7	A(4)
E4	L02N_7	A(5)
K5	L52P_7	A(6)
K6	L52N_7	A(7)
J5	L46P_7	A(8)
J6	L46N_7	A(9)
H3	L25N_7	D(0)
K7	L43P_7	D(10)
L8	L51P_7	D(11)
L7	L51N_7	D(12)
G5	L01P_7	D(13)
H6	L19P_7	D(14)
H5	L45P_7	D(15)
J4	L48N_7	D(1)
J3	L48P_7	D(2)
K4	L54N_7	D(3)
C2	L03N_7	D(4)
C1	L03P_7	D(5)

Table B-23: Pin mapping for the left SRAM bank (cont'd)

BGA Name	I/O Name	SRAM-L Pin
D2	L04N_7	D(6)
D1	L04P_7	D(7)
H7	L19N_7	D(8)
J7	L43N_7	D(9)
E3	L02P_7	$\overline{WE}$

Table B-24: Pin mapping for the OGPIO and Header

BGA Name	I/O Name	OGPIO Pin
C13	L96P_0	OGPIO(0) / pin 22
B12	L78P_0	OGPIO(10) / pin 25
B11	L78N_0	OGPIO(11) / pin 21
C11	L76P_0	OGPIO(12) / pin 2
D11	L76N_0	OGPIO(13) / pin 17
H11	L75P_0	OGPIO(14) / pin 2
G11	L75N_0	OGPIO(15) / pin 1
H15	L96P_1	OGPIO(16) / pin 14
H14	L96N_1	OGPIO(17) / pin 9
F14	L95P_1	OGPIO(18) / pin 10
D14	L94P_1	OGPIO(19) / pin 13
E13	L95P_0	OGPIO(1) / pin 24
E14	L94N_1	OGPIO(20) / pin 12
A13	L93P_1	OGPIO(21) / pin 27
A12	L93N_1	OGPIO(22) / pin 23
A15	L92P_1	OGPIO(23) / pin 31
A14	L92N_1	OGPIO(24) / pin 29
C15	L91P_1	OGPIO(25) / pin 26
B15	L91N_1	OGPIO(26) / pin 32
E15	L78P_1	OGPIO(27) / pin 11
D15	L78N_1	OGPIO(28) / pin 28
G15	L76P_1	OGPIO(29) / pin 30
H13	L94P_0	OGPIO(2) / pin 7
F15	L76N_1	OGPIO(30) / pin 16
F16	L75P_1	OGPIO(31) / pin 15
G13	L94N_0	OGPIO(3) / pin 6
C12	L93P_0	OGPIO(4) / pin 20
D12	L93N_0	OGPIO(5) / pin 19
E12	L92P_0	OGPIO(6) / pin 5
F12	L92N_0	OGPIO(7) / pin 8
H12	L91P_0	OGPIO(8) / pin 4
G12	L91N_0	OGPIO(9) / pin 3

Table B-25: Pin mapping for the right SDRAM bank

BGA Name	I/O Name	SDRAM-R Pin
P23	L94N_3	A(0)
L23	L75N_2	A(10)
P21	L91N_3	A(11)
R22	L73N_3	A(12)
R24	L75N_3	A(1)
R23	L75P_3	A(2)
N22	L94N_2	A(3)
N21	L94P_2	A(4)
M22	L76N_2	A(5)
M21	L76P_2	A(6)
L22	L70N_2	A(7)
L21	L70P_2	A(8)
P22	L94P_3	A(9)
M23	L78N_2	BA(0)
L24	L75P_2	BA(1)
N23	L96P_2	$\overline{\text{CAS}}$
R21	L73P_3	CLK
R25	L78P_3	DQ(0)
M19	L73P_2	DQ(10)
L20	L52P_2	DQ(11)
L19	L73N_2	DQ(12)
P20	L91P_3	DQ(13)
P19	L93N_3	DQ(14)
R20	L76N_3	DQ(15)
R26	L78N_3	DQ(1)
P26	L96P_3	DQ(2)
N24	L96N_2	DQ(3)
N26	L96N_3	DQ(4)
M26	L91P_2	DQ(5)
M25	L91N_2	DQ(6)
L26	L72P_2	DQ(7)
N19	L93P_3	DQ(8)
N20	L93P_2	DQ(9)
L25	L72N_2	LDQM
M24	L78P_2	$\overline{\text{RAS}}$
M20	L93N_2	UDQM
K26	L69P_2	$\overline{\text{WE}}$

Table B-26: Pin mapping for the right SRAM bank

BGA Name	I/O Name	SRAM-R Pin
J21	L46N_2	A19
F26	L43N_2	$\overline{\text{A19}}$
J26	L69N_2	A(0)
J22	L49N_2	A(10)

Table B-26: Pin mapping for the right SRAM bank (cont'd)

BGA Name	I/O Name	SRAM-R Pin
K21	L51N_2	A(11)
K22	L51P_2	A(12)
E23	L04N_2	A(13)
E24	L04P_2	A(14)
F23	L21N_2	A(15)
F24	L21P_2	A(16)
G23	L25N_2	A(17)
G24	L25P_2	A(18)
J25	L54P_2	A(1)
H26	L48P_2	A(2)
H25	L48N_2	A(3)
G26	L43P_2	A(4)
E26	L19P_2	A(5)
E25	L22N_2	A(6)
D26	L19N_2	A(7)
D25	L03P_2	A(8)
C26	L01P_2	A(9)
C25	L03N_2	D(0)
H21	L24P_2	D(10)
H22	L24N_2	D(11)
G20	L02N_2	D(12)
H20	L02P_2	D(13)
K20	L52N_2	D(14)
J20	L46P_2	D(15)
B26	L01N_2	D(1)
K24	L67P_2	D(2)
K23	L67N_2	D(3)
J24	L54N_2	D(4)
J23	L49P_2	D(5)
H24	L45P_2	D(6)
H23	L45N_2	D(7)
G21	L06N_2	D(8)
G22	L06P_2	D(9)
F25	L22P_2	$\overline{WE}$

Table B-27: Pin mapping for the RS-232 Interface

BGA Name	I/O Name	RS Pin
B24	L01P_1	R2OUT
B23	L01N_1	T1IN

Table B-28: Pin mapping for the Switches

BGA Name	I/O Name	SW Pin
AA22	L01P_3	S3
Y22	L01N_3	S4

Table B-29: Pin mapping for the Video DAC/VGA Output

BGA Name	I/O Name	VGA Pin
U3	L67N_6	B(0)
T4	L75P_6	B(1)
T3	L75N_6	B(2)
Y1	L46P_6	B(3)
W2	L45P_6	B(4)
W1	L46N_6	B(5)
V2	L51N_6	B(6)
V1	L52P_6	B(7)
U5	L69N_6	G(0)
T6	L70N_6	G(1)
T5	L70P_6	G(2)
W4	L43N_6	G(3)
W3	L45N_6	G(4)
V4	L49N_6	G(5)
V3	L51P_6	G(6)
U4	L67P_6	G(7)
T2	L73P_6	$\overline{\text{HSYNC}}$
U1	L52N_6	PCLK
V7	L48N_6	R(0)
U7	L54P_6	R(1)
T8	L72P_6	R(2)
T7	L54N_6	R(3)
W5	L43P_6	R(4)
V6	L48P_6	R(5)
V5	L49P_6	R(6)
U6	L69P_6	R(7)
T1	L73N_6	$\overline{\text{VSYNC}}$

Appendix C

Initial Operation

The following sequence of tests shall be applied to every board befor it may be used:

- DC/DC converter tests
- On-board oscillator and quartz tests
- JTAG interface tests

It is recommended to write test procedures for all I/O components on the board to speed up tracking and locating potential errors. At the time writing, no such procedure does exist.

C.1 Preparations

Consult figure 5-2 to locate the components brought up in this appendix. First, all fuses (F1 , F2 , F3) have to be removed from the module. These are installed in later sections of this appendix. Depopulate J1 and J2 , if populated.

Providing power to the board when J1 and J2 are spuriously populated can damage the board

C.2 DC/DC Converter Tests

The XFBOARD contains two on-board DC/DC converters to generate supply voltages for the +1.5 V and the +3.3 V power planes. The +1.5 V converter's reference designator is U10 and the +3.3 V converter's reference designator is U9 . Use this test to check if the two converters function correctly. Continue **only** if you have completed the preparations (previous section).

If available, connect a standard PC power supply to the ATX_CONNECTOR . Do the following measurements on the fuse holders:

- Fuse F1 (pin tagged red in figure 5-2) to GND. Passed if +5 V.
- Fuse F1 (untagged pin) to GND. Passed if 0 V.

- Fuse F2 (tagged pin) to GND. Passed if +3.3 V.
- Fuse F2 (untagged pin) to GND. Passed if 0V.

Resolve detected problems.

Now, install fuse F1 (5 A). This connects the +1.5 V converter to the +5 V power plane. The LED_5V should start to glow. Measure the voltage:

- Fuse F1 (untagged pin) to GND. Passed if +5 V.
- Fuse F3 (tagged pin) to GND. Passed if +1.5V.
- Fuse F3 (untagged pin) to GND. Passed if 0 V.

If no problem is detected, install fuses F2 and F3 (5 A each). As soon as both LED_3V3 and LED_1V5 are lit you know that the board is powered up. In that configuration, the +3.3 V converter is not used.

Do not populate J1 and J2 whenever you are using the ATX connector for power supply

Now, power supply using the +5 V DC adapter is going to be tested. Again, remove all fuses (F1 , F2 , F3). Then connect the adapter to DC_CONNECTOR . Measure the voltages:

- Fuse F1 (pin tagged red in figure 5-2) to GND. Passed if +5 V.
- Fuse F1 (untagged pin) to GND. Passed if 0 V.

Resolve detected problems.

Now, install F1 . This connects the +1.5 V converter to the +5 V power plane. The LED_5V should start to glow. Measure the voltage:

- Fuse F1 (untagged pin) to GND. Passed if +5 V.
- Fuse F3 (tagged pin) to GND. Passed if +1.5V.
- Fuse F3 (untagged pin) to GND. Passed if 0 V.

If no problem is detected, populate J1 and J2 to put the +3.3 V converter into operation. Perform these measurements:

- Fuse F2 (tagged pin) to GND. Passed if +3.3 V.
- Fuse F2 (untagged pin) to GND. Passed if 0 V.

As soon as you are able to verify theses voltages, install F2 and F3 . LED_3V3 and LED_1V5 start to glow, prooving that the board is completely powered up now.

As no configuration data is present on the board, neither in the BootPROM nor in one of the FPGAs, LED_CDONE and LED_RDONE should be dark.

C.3 Onboard Oscillator and Quartz Tests

The on-board oscillator for the FPGAs, Y2 , is located near the header between C-FPGA and R-FPGA. It generates a 50 MHz signal that is available to both FPGAs. Use an oscilloscope to check the waveform at test point X2 . Now check the quartz crystals for the Ethernet PHYs: quartz Y2 using test points X10 and X11 , quartz Y3 using X12 and X13 . Both should provide a 25 MHz clock.

C.4 JTAG Interface Tests

The XFBOARD has two separate JTAG connectors, X3 and X6 . Connect an appropriate cable to X6 and start the Xilinx iMPACT software on the host PC. Initialize the boundary scan chain and see if both the C-FPGA and the configuration PROM appear in the chain layout. Do the same with X3 . Here, the R-FPGA should be seen in the chain.

Glossary

C-FPGA CPU FPGA on the XFBOARD running the soft CPU core (MicroBlaze).

CMOS Complementary Metal Oxide Semiconductor. This term refers both to the semiconductor devices built in this technology and the I/O signal levels required by these devices.

CoDec Coder/Decoder. Device being DAC and ADC at a time.

CPLD Complex Programmable Logic Device. A combination of a fully programmable AND/OR array and a bank of macrocells. CPLDs are nonvolatile.

CPU Central Processing Unit. Central unit in a computer containing the logic circuitry that performs the instructions of a computer's programs

DAC Digital-to-Analog Converter. Converts a digitally represented number into a voltage or current.

DCE Data Circuit-terminating Equipment or Data Communications Equipment. Usually a modem in an RS-232 environment.

DCM Digital Clock Manager. The Xilinx Virtex-II devices have 4 to 12 such clock managers used for clock de-skew, frequency synthesis and phase shifting using a delay-locked loop (DLL).

DTE Data Terminal Equipment. This role is usually played by a computer or terminal in an RS-232 communication.

FlashRAM Flash Random Access Memory. Retains data bits in memory even when power is removed. It is organized so that a section of memory cells is erased in a singel action or "flash".

FPGA Field-Programmable Gate Array. An integrated circuit that can be programmed in the field after manufacture. FPGAs are volatile.

JTAG IEEE 1149.1 standard used for boundary-scan testing of circuits and in-system programming of configurable devices and memories.

LED Light Emitting Diode.

LVCMOS Low voltage version of the CMOS I/O standard.

MAC Media Access Controller. A sublayer of the Data-Link-Layer of the TCP protocol. There is a different MAC sublayer for each physical device type. The other sublayer level in tthe Data-Link-Layer is the Logical Link Control sublayer

PCB Printed Circuit Board. Board with an electrical circuit to mount electronic components.

PROM Programmable Read-Only Memory. Memory that can be modified once by a user. As this leaves no margin for error, most PROM chips are so called EPROMS which are eraseable and reprogrammable.

R-FPGA Reconfigurable FPGA on the XFBOARD running the RHWOS.

RHWOS Reconfigurable Hardware Operation System. Operating system running on reconfigurable hardware platforms, offering services to the application developer and being responsible for resource managment and allocation.

SDRAM Synchronous Dynamic Random Access Memory. Retains data bits in memory as long as power is being supplied. Bits are stored in small capacitances. Due to charge loss in these capacitances, the cells need to be refreshed periodically.

SRAM Static Random Access Memory. Retains data bits in memory as long as power is being supplied. Bits are stored in a register-like structure.

VGA Video Graphics Array. The accepted minimum standard for PC monitors. Introduced by IBM in 1987.

ZBT-RAM Zero Bus Turnaround Random Access Memory. Fast static RAM allowing 100% use of bus cycles during back-to-back read/write and write/read cycles by eliminating the idle cycle between read and write cycles which is mandatory with ordinary RAM.

Bibliography

[1] H. Walder, M. Platzner. Swiss Federal Institute of Technology Zurich (ETH), Computer Engineering and Networks Laboratory, TIK Report Nr. 168. *Reconfigurable Hardware OS Prototype*, April 2003.
http://www.tik.ee.ethz.ch/∼walder/XFORCES/TIKR168.pdf

[2] H. Walder, M. Platzner. Swiss Federal Institute of Technology Zurich (ETH), Computer Engineering and Networks Laboratory. *Non-preemptive Multitasking on FPGA: Task Placement and Footprint Transform*, June 2002.
http://www.tik.ee.ethz.ch/∼walder/XFORCES/ERSA02.pdf

[3] A. Schweizer. Swiss Federal Institute of Technology Zurich (ETH), Computer Engineering and Networks Laboratory, Master's Thesis. *Reconfigurable Computing on a DIMM Module (TIKDIMM)*, 2003.

[4] M. Lerjen, C. Zbinden. Master's Thesis, Swiss Federal Institute of Technology Zurich (ETH), Computer Engineering and Networks Laboratory. *Reconfigurable BlueTooth Ethernet Bridge*, 2002.
http://www.tik.ee.ethz.ch/∼walder/SADA/BluetoothEthernetBridge/EBB_Report.pdf

[5] M. Dyer, M. Wirz. Master's Thesis, Swiss Federal Institute of Technology Zurich (ETH), Computer Engineering and Networks Laboratory. *Reconfigurable System on FPGA*, 2002.
http://www.tik.ee.ethz.ch/∼walder/SADA/ReconfigurableSystemOnFPGA/RSOF_Report.pdf

[6] S. Nobs, D. Engeler. Semester Thesis, Swiss Federal Institute of Technology Zurich (ETH), Integrated Systems Laboratory. *MIDI Synthesizer*, February 2003.

[7] G. Brebner. In Proceedings of the 6th International Workshop on Field-Programmable Logic and Applications (FPL), pages 327 336. *A Virtual Hardware Operating System for the Xilinx XC6200*, Springer, 1996.

[8] Xess Corp. *XSV Board V1.1 Manual*, May 2001.
http://www.xess.com/manuals/xsv-manual-v1_1.pdf

[9] Xilinx, Inc. *Virtex-II Platform FPGAs: Advance Product Specification*, 2002.
http://direct.xilinx.com/bvdocs/publications/ds031.pdf

[10] Xilinx, Inc. *Virtex-II Platform FPGA User Guide*, December 2002.
http://www.xilinx.com/publications/products/v2pro/ug_pdf/ug012.pdf

[11] Xilinx, Inc. *MicroBlaze RISC 32-Bit Soft Processor*, August 2001.
http://www.xilinx.com/ipcenter/catalog/logicore/docs/
microblaze_risc_32bit_proc_final.pdf

[12] Xilinx, Inc. *Power Distribution System (PDS) Design: Using Bypass/Decoupling Capacitors*, August 2002.
http://www.xilinx.com/xapp/xapp623.pdf

[13] Maxim Integrated Products. *Remote/Local Temperature Sensor with SMBus Serial Interface*, March 1998
http://pdfserv.maxim-ic.com/arpdf/MAX1617.pdf

[14] Xilinx, Inc. *XC18V00 Series In-System Programmable Configuration PROMs*, April 2003
http://direct.xilinx.com/bvdocs/publications/ds026.pdf

[15] Bel Fuse Inc. *Integrated Connector Modules*, 2000.
http://www.belfuse.com/Data/DBObject/bm00801.pdf

[16] Intel Corp. *LXT970A Dual-Speed Fast Ethernet Transceiver*, January 2001.
http://www.intel.com/design/network/products/lan/datashts/24909901.pdf

[17] Maxim Integrated Products. *+5V-Powered, Multichannel RS-232 Drivers/Receivers*, 2000.
http://pdfserv.maxim-ic.com/arpdf/MAX220-MAX249.pdf

[18] Asahi Kasei Microsystems Co., Ltd. *AK4563A Low Power 16bit 4ch ADC & 2ch DAC with ALC*, December 2000.
http://www.asahi-kasei.co.jp/akm/en/product/ak4563a/ek4563a.pdf

[19] Intersil Corporation. *HI1178 Triple 8-bit, 40MSPS, RGB, 3-Channel D/A Converter*, 2000.
http://www.intersil.com/design/images/DataSheet.jpg

[20] Alliance Semiconductor. *AS7C4096/AS7C34096, 5V/3.3V 52K x 8 CMOS SRAM*, March 2002, v.1.8.
http://www.alsc.com/pdf/sram.pdf/fa/AS7C34096.pdf

[21] Infineon Technologies. *HYB39S256400/800/160CT(L) 256MBit Synchronous DRAM*, August 2000.
http://www.infineon.com/cmc_upload/0/000/018/241/SDRAM_256M.pdf

[22] Fairchild Semiconductor. *MM74HC244 Octal 3-STATE Buffer*, August 2000.
http://www.fairchildsemi.com/ds/MM/MM74HC244.pdf

[23] Texas Instruments, Inc. *3 V to 6 V Input, 6 A Output Synchronous Buck PWM Switcher with Integrated FETs (SWIFT)*, January 2002.
http://focus.ti.com/lit/ds/symlink/tps54611.pdf

[24] Intel Corp. *3 Volt Intel StrataFlash Memory*, February 2003.
ftp://download.intel.com/design/flcomp/datashts/29066713.pdf

[25] Pericom Semiconductor Corp. Application Brief 23. *Clock Termination Techniques*, November 1999.
http://www.pericom.com/docs/AB23.pdf

[26] Xilinx, Inc.
http://www.xilinx.com

[27] Amphion Semiconductor Ltd.
http://www.amphion.com

[28] Gaisler Research.
http://www.gaisler.com

[29] In-Circuit Design Pty Ltd. *Interactive Multilayer Impedance Calculator*, 1998.
http://www.icd.com.au/board.html

[30] Adam Chapweske. *PS/2 Mouse/Keyboard Protocol*, 1999.
http://panda.cs.ndsu.nodak.edu/ achapwes/PICmicro/PS2/ps2.htm

[31] Christopher E. Strangio, CAMI Research Inc. *The RS-232 Standard*, 1997.
http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html

[32] SBS Implementers Forum. *System Management Bus Specification*, August 2000.
http://www.smbus.org/specs/smbus20.pdf

Please note that the URLs noted below the reference entries have been valid at the time writing this document. They may be outdated in the meantime and therefore point to nowhere.