Nuclear Physics Division
Fast Electronics Group

Description and Instructions
for the
FPGA Data Acquisition and Control Board

Chris Cuevas
Benjamin Raydo
Jeff Wilson
2018 September 24

Contents

TABLE OF CONTENTS

1 Introduction

The Field Programmable Gate Array Data Acquisition(FPGA) and Control Board design offers a very compact method to interface custom electronics for a variety of detector readout systems.

The FPGA DAq and Control board is built on a small proprietary circuit card that is 4" by 2". The design uses the rich resources of a Xilinx Aritix 7 FPGA and makes use of the high number of input and output connections. The circuit board includes a high speed fiber optic LC transceiver that provides a downlink for controlling external circuit functions, and also a high speed (2.5Gpbs) uplink that allows high speed DAq from the Artix 7 Gigabit transceivers.

The circuit board includes a power supply connection, on-board power supply regulators for core FPGA and input/output voltages. Miniature coaxial connections are provided so the user can interface common trigger or synchronization signals for testing purposes. The main input and output connections to the board are high density connectors that provide up to 192 single ended signals to the FPGA. These input and output signals can be configured for a number of different type of single ended logic levels. We will use HSTL signal levels for the initial application.

2 Functional Description

The RICH FPGA module can be used in a stand-alone setup that interfaces to a PC using optical Ethernet (1000BASE-SX) or in a multi-RICH FPGA module setup that requires other proprietary Jefferson Lab hardware (the VXS SSP module at minimum). The RICH FPGA module has been interfaced to a MAROC3A ASIC front-end mezzanine board. The RICH FPGA was configured as a 192 channel TDC (1ns resolution) and ADC event builder, providing readout data though either 1Gbps Ethernet or via the VXS SSP. Slow controls for the MAROC3A ASIC and various monitoring features were also implemented (single event upset, voltage, temperature, and MAROC3A scaler rate monitors).

2.1 Stand-Alone Operation

In stand-alone operation, a single RICH FPGA module is used which can be triggered from an internal signal (pulser, or front-end) or from an external trigger fed into a coaxial TTL input. The module is controlled by a standard PC using short-range optical gigabit Ethernet (1000BASE-SX). TCP sockets used for event data transport to the PC and a separate TCP socket is used for device configuration and status monitoring. This setup is very easy to use because it can work with standard PCs and is the preferred setup when testing or characterizing front-end ASIC modules and/or front-end detectors that require a single RICH FPGA module. Multiple RICH FPGA modules can operate in parallel in this configuration, but they will run on independent oscillators, which degrade the jitter performance when comparing data between the modules and also present potential problems synchronizing events between modules. The problems with multi-RICH FPGA modules in stand-alone mode are addressed by the CLAS12 RICH system, which uses additional Jefferson Lab proprietary hardware.

2.2 System Synchronous Mode (CLAS12 RICH Example)

In CLAS12 RICH, there are 138 RICH FPGA modules used, which must remain synchronized to the master oscillator and trigger (the Trigger Supervisor). This multi-

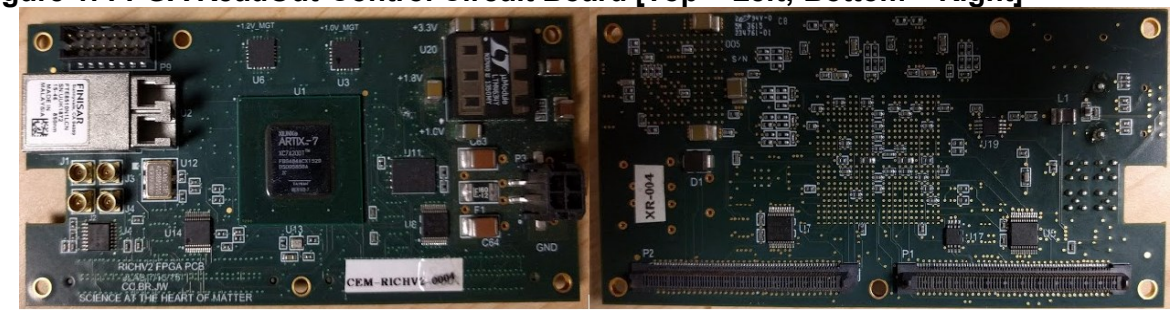
module synchronization requires a fixed latency optical link to deliver a clock, trigger, a data/address bus bridge, event data transport, and various prompt status signals. This is accomplished using the Xilinx GTP transceiver in a special mode that bypasses components that introduce timing uncertainty (RX buffer and RX comma alignment) and instead using the RX recovered clock and phase shifting this clock to achieve word alignment. In the TX direction, fixed latency is not needed so the local reference clock is used to send data using clock correction in the protocol to deal with small frequency differences between the RICH FPGA and remote end that would otherwise cause buffer overruns or underruns.

The Jefferson Lab Sub-System-Processor (SSP) is a VXS payload module with a VME interface to the crate CPU. The SSP contains also 32 full duplex optical transceivers that each can interface to a RICH FPGA board. The SSP receives the global clock and trigger signals from the Trigger Supervisor from the VXS backplane and uses these signals to implement the complementary parts of the fixed latency link that goes to the RICH FPGA (mainly, the SSP TX bypasses the GTX transmit buffers). The SSP memory maps a 4kByte region of A24 VME memory to each RICH FPGA so that the RICH FPGA registers appear to the VME CPU as if part of the VME bus. The SSP also contains a large, 4GByte, memory buffer that RICH FPGA events can be stored while waiting for the VME CPU to readout events in the normal way (using Jefferson Lab CODA running on VME controllers). Up to 16 SSP modules can be installed into a single VME crate, which could support up to 512 RICH FPGA modules. The limiting factor is the VME CPU readout bandwidth: if the RICH FPGA modules generated more data than the CPU can read over VME, then RICH FPGA modules and SSPs will have to be moved to another crate. In practice, CLAS12 RICH is far from that limit (limit is: 100MB/s using a 1GbE link on the CPU. CLAS12 RICH was reporting a peak data rate of ~15MB/s during the last run period).

3.0 Hardware Description

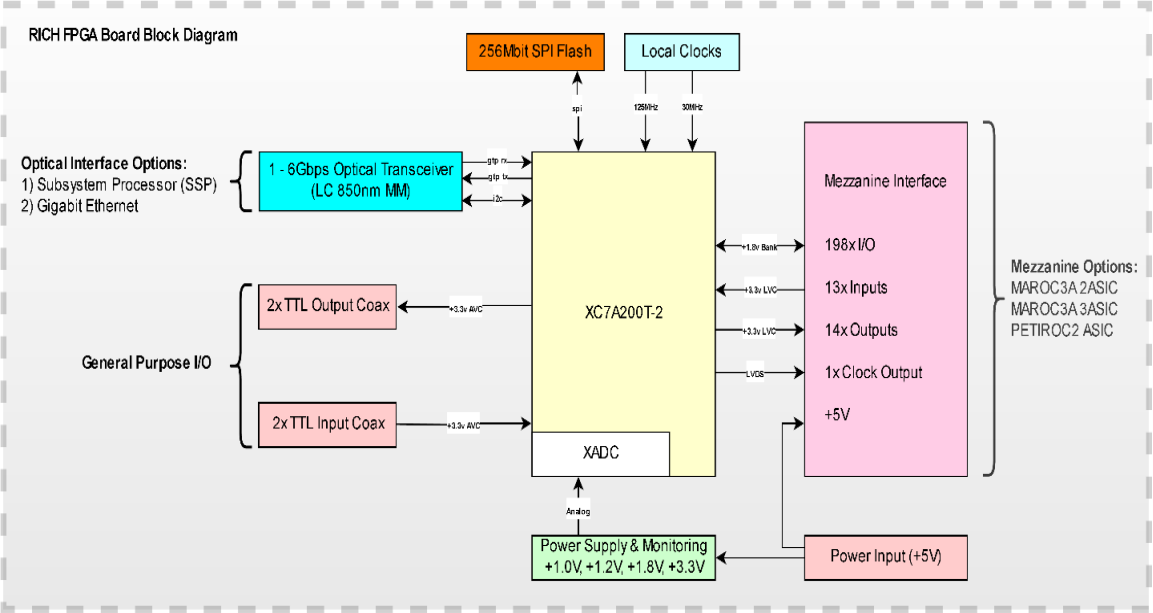
The RICH FPGA hardware was designed as an on-detector optical readout module for CLAS12 RICH; however, by keeping the detector specific analog front-end & digitization on a separate mezzanine interface the RICH FPGA module itself is very generic and can interface to future mezzanine cards for a variety of applications. Figure 1 shows the PCB design top and bottom views with all components visible. On top the LC fiber transceiver, JTAG header, 4 coaxial (2 input, 2 output), FPGA, oscillator, flash memory, and regulators can be seen. On the bottom the 2 high density mezzanine headers are visible.

Figure 1: FPGA ReadOut-Control Circuit Board [Top – Left; Bottom – Right]



In Figure 2 the hardware block diagram is shown. The RICH FPGA is a fairly generic design that allow it to adapt to various front-end PCB through the mezzanine connectors. These mezzanine connectors provide power, reference clocks, and >200 I/O pins that can be defined by the FPGA firmware. Currently MAROC3A front-end boards have been developed and a new PETIROC2 front-end design underway.

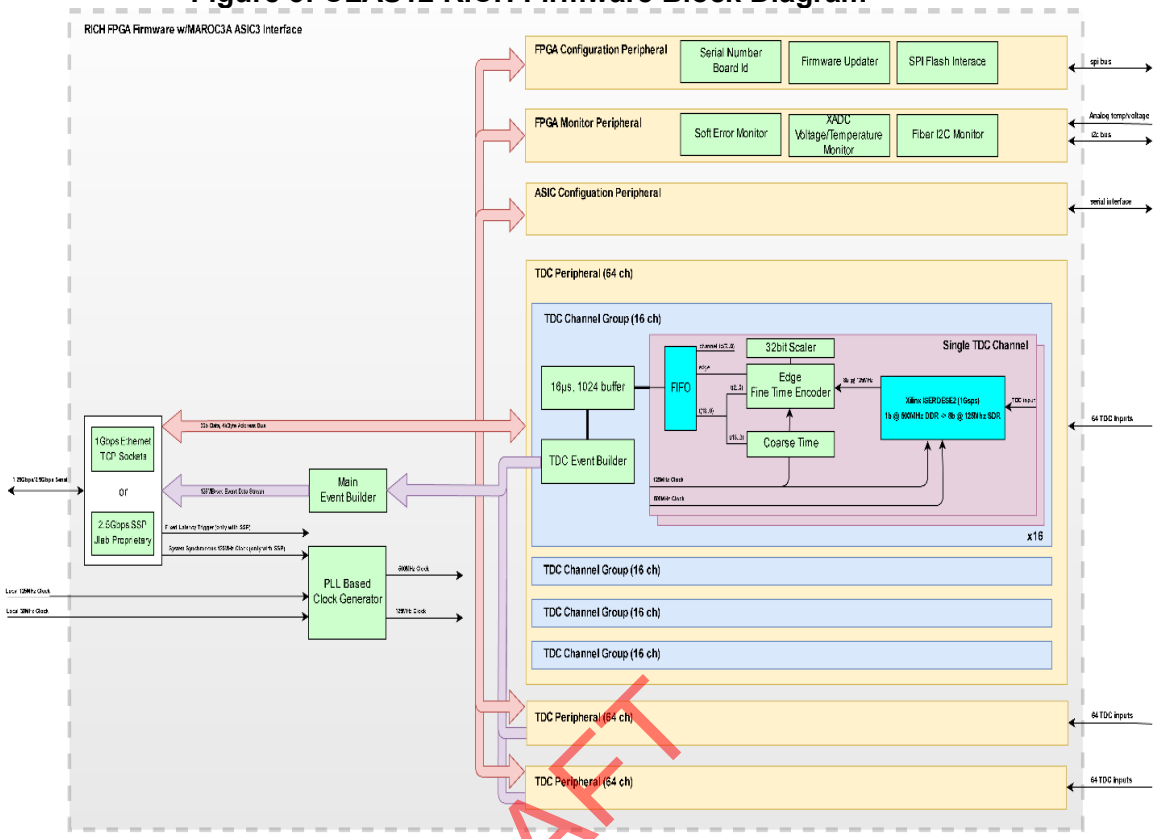
Figure 2: FPGA Data Acquisition and Control board



4 Firmware Description

The firmware for the RICH FPGA is modular by design consisting of peripherals, making it easy to change the design by adding/removing peripherals with minimal impact on existing firmware modules. A common peripheral bus connects to all peripheral modules that allow registers to be read and written (for configuration, monitoring, etc). The peripheral bus master either is a gigabit Ethernet controlled module (when operating in stand-alone mode) or controlled by a Jlab SSP module (when operating in a larger system synchronous mode). Peripherals can optionally generate event data when a trigger is received. A common event builder bus allows participating modules to connect to the main event builder which will build a complete event by waiting for event fragments from each participating peripheral. A clock generator is responsible for generating a necessary clocks/phases and resets needed by all peripherals in the system.[See Figure 3]

Figure 3: CLAS12 RICH Firmware Block Diagram



4.1 Gigabit Ehernet Optical Interface

Commercial IP was used to implement the 1Gbps Ethernet MAC and a subset of the TCP/IP protocols (UDP, TCP, ARP, PING). The Xilinx 1G Ethernet PCS/PMA IP core implemented the 1000BASE-SX physical layer. Two TCP socket servers run in the firmware: one that streams event data out of the FPGA and another that processes commands for register configuration

4.1.1 Event Streaming TCP Socket

The event stream TCP server listens on port 6103. When a connection is established, the FPGA will stream trigger event data on this socket a up to 1Gbps. The event-streaming socket is intended for heavy traffic to support high event rates. The event data is sent as 32-bit words (big endian) and the specific data format is defined by the peripherals that generate it.

4.1.2 Register Configuration TCP Socket [Table 1]

The register configuration TCP server listens on port 6102. When a connection is established, the FPGA waits for commands from the socket. The only two support commands are Read32, Write32, and the FPGA responds only with messages Read32_RSP after each Read32 command.

Table 1

	Write32	Read32

Request	typedef struct
	{
	  int len; // 16 only supported len
	  int type; // 4 Write32 message type
	  int wrcnt; // 1 only support wrcnt
	  int addr; // 32bit register address
	  int flags; // 0 only support flags
	  int vals[1]; //32bit data value to write
	} write_struct;
		typedef struct
		{
		  int len; // 12 only supported len
		  int type; // 3 Write32 message type
		  int rdcnt; // 1 only support rdcnt
		  int addr; // 32bit register address
		  int flags; // 0 only support flags
		} read_struct;

Response	none	typedef struct
		{
		  int len; // 8 only supported len
		  int type; // 0x80000003 only supported type
		  int rdcnt; // 1 only support rdcnt
		  int data[1]; // 32bit data value read
		} read_rsp_struct;

Example	void rich_write32(void *addr, int val)
	{
	  write_struct ws;
	
	  ws.len = 16;
	  ws.type = 4;
	  ws.wrcnt = 1;
	  ws.addr = (int)((long)addr);
	  ws.flags = 0;
	  ws.vals[0] = val;
	  write(sockfd_reg, &ws, sizeof(ws));
	}
		unsigned int rich_read32(void *addr)
		{
		  read_struct rs;
		  read_rsp_struct rs_rsp;
		  int len;
		
		  rs.len = 12;
		  rs.type = 3;
		  rs.rdcnt = 1;
		  rs.addr = (int)((long)addr);
		  rs.flags = 0;
		  write(sockfd_reg, &rs, sizeof(rs));
		
		  len = read(sockfd_reg, &rs_rsp, sizeof(rs_rsp));
		  if(len != sizeof(rs_rsp))
		  printf("Error in %s: socket read failed...\n", __func__);
		
		  return rs_rsp.data[0];
		}

4.3 Time-to-Digital Converter Peripheral Function

The TDC peripheral function is a 64 channel, 1ns resolution Time-to-Digital-Converter. It is an extremely simple implementation on the FPGA, which doesn't require special tricks to implement or require calibration to achieve the targeted resolution. Using the Xilinx ISERDESE2 primitive any pin can be sampled at up to 1.25GHz using a 625MHz reference clock in DDR mode. The TDC linearity therefore suffers by the clock duty cycle as it deviates from a nominal 50% (datasheet specifications limits this error to 200ps).

In CLAS12 RICH the MAROC3A ASIC front-end boards are used with the RICH FPGA and sampled at 1GHz (using a 500MHz reference clock in DDR mode). Each input pin has an ISERDESE2 in 1->8 mode at the input that serves as the fine time measurement, forming the lower 3bits of the time measurement (1ns resolution). A course timestamp (8ns resolution) is concatenated with the fine TDC time to form the full TDC timestamp. Each rising or falling edge is written to a FIFO containing the TDC timestamp, the TDC channel number, and the edge type seen. A group of 16 channels has their FIFOs funneled into a common memory buffer that can hold 1024 hits for a least 8us times. To prevent data loss, the sum of the hit rate in the group cannot exceed 125MHz over any 8μs time span. This limit corresponds to an average hit rate of 4MHz per channel, which is not a problem for CLAS12 RICH (which has average rates on the order of 10kHz per channel); however, it would be a simple matter of reducing the

channel group sizes from 16 and using more TDC event builders to improve this if ever needed.

CLAS12 RICH uses front-end ASIC boards with up to three MAROC3A ASIC. Each ASIC provides 64 TDC channels, for a total of 192 channels the RICH FPGA must support, which requires a total of three TDC peripherals. The event data format is documented below as the “TDC Hit” word type. Each TDC peripheral utilizes about 10% of the FPGA LUTs and 6% of the FPGA flip flops [FFs].

6 Readout Data Format

The readout data format utilizes the same encoding scheme defined for the JLAB FADC250. The word length for the readout data is 32bits. The event length is variable and depends on several factors (detector occupancy, headers, trailers, filler words).

Data Word Categories

Data words from the module are divided into two categories: Data Type Defining (bit 31 = 1) and Data Type Continuation (bit 31 = 0). Data Type Defining words contain a 4-bit data type tag (bits 30 - 27) along with a type dependent data payload (bits 26 - 0). Data Type Continuation words provide additional data payload (bits 30 – 0) for the *last defined data type*. Continuation words permit data payloads to span multiple words and allow for efficient packing of various data types spanning multiple data words. Any number of Data Type Continuation words may follow a Data Type Defining word.

Data Type List

0	Block Header
1	Block Trailer
2	Event Header
3	Trigger Time
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	TDC Hit
9	ADC
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Data Not Valid (empty module)
15	Filler Word (non-data)

Data Type: Block Header

Type: 0x0
Size: 1 word
Description: Indicates the beginning of a block of events. (High-speed readout of a board or a set of boards is done in blocks of events)

31	30	29	28	27	26	25	24
1	0	0	0	0	SLOTID		
23	22	21	20	19	18	17	16
SLOTID		0	0	0	0	BLOCK_NUMBER	
15	14	13	12	11	10	9	8
BLOCK_NUMBER							
7	6	5	4	3	2	1	0

BLOCK_SIZE

BLOCK_NUMBER:
Event block number (used to align blocks when building events)
BLOCK_SIZE:
Number of events in block
SLOTID:
Slot ID (set by VME64x backplane)

Data Type: Block Trailer

Type: 0x1
Size: 1 word
Description: Indicates the end of a block of events. The data words in a block are bracketed by the block header and trailer.

31	30	29	28	27	26	25	24
1	0	0	0	1	SLOTID		
23	22	21	20	19	18	17	16
SLOTID		NUM_WORDS					
15	14	13	12	11	10	9	8
NUM_WORDS							
7	6	5	4	3	2	1	0
NUM_WORDS							

NUM_WORDS:
Total number of words in block of events
SLOTID:
Slot ID (set by VME64x backplane)

Data Type: Event Header

Type: 0x2
Size: 1 word
Description: Indicates the start of an event. The included trigger number is useful to ensure proper alignment of event fragments when building events. The 21bit trigger number (2M count) is not a limitation, as it will be used to distinguish events within event blocks, or among events that are concurrently being built or transported.

31	30	29	28	27	26	25	24
1	0	0	1	0	DEV_ID		
23	22	21	20	19	18	17	16
DEV_ID			TRIGGER_NUMBER				
15	14	13	12	11	10	9	8
TRIGGER_NUMBER							
7	6	5	4	3	2	1	0
TRIGGER_NUMBER							

DEV_ID:
RICH FPGA programmable device ID

TRIGGER_NUMBER:
Accepted event/trigger number

Data Type: Trigger Time

Type: 0x3
Size: 2 words
Description: Time of trigger occurrence relative to the most recent global reset. The time is measured by a 48bit counter that is clocked from the 250MHz system clock. The assertion of the global reset clears the counter. The de-assertion of global reset enables counter and thus sets t=0 for the module. The trigger time is necessary to ensure system synchronization and is useful in aligning event fragments when building events.

Word 1:

31	30	29	28	27	26	25	24
1	0	0	1	1	0	0	0
23	22	21	20	19	18	17	16
TRIGGER_TIME_L							
15	14	13	12	11	10	9	8
TRIGGER_TIME_L							
7	6	5	4	3	2	1	0
TRIGGER_TIME_L							

TRIGGER_TIME_L:
This is the lower 24bits of the trigger time

Word 2:

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
TRIGGER_TIME_H							
15	14	13	12	11	10	9	8
TRIGGER_TIME_H							
7	6	5	4	3	2	1	0
TRIGGER_TIME_H							

TRIGGER_TIME_H:
This is the upper 24bits of the trigger time

Data Type: TDC Hit

Type: 0x8
Size: 1 words
Description: This data type identifies a TDC hit

Word 1:

31	30	29	28	27	26	25	24
1	1	0	0	0	EDGE	-	-
23	22	21	20	19	18	17	16
CHANNEL_NUMBER							
15	14	13	12	11	10	9	8
TDC_TIME							
7	6	5	4	3	2	1	0
TDC_TIME							

EDGE:

'0' – indicates hit is a leading edge (0->1 transition as seen by FPGA)
'1' – indicates hit is a trailing edge (1->0 transition as seen by FPGA)

CHANNEL_NUMBER:

0-191 – indicates which channel TDC corresponds to

TDC_TIME:

1ns TDC hit timestamp which is measured relative to the beginning of the defined readout window

Data Type: ADC

Type: 0x9
Size: 33 words
Description: This data type contains the measured ADC values for all 64 channels of a single MAROC ASIC.

Word 0:

31	30	29	28	27	26	25	24
1	1	0	0	1	-	-	-
23	22	21	20	19	18	17	16
ADC_HOLD2							
15	14	13	12	11	10	9	8
ADC_HOLD1							
7	6	5	4	3	2	1	0
ADC_MAX_BITS				-	-	MAROC_ID	

ADC_HOLD2:
MAROC HOLD2 delay in 8ns ticks

ADC_HOLD1:
MAROC HOLD1 delay in 8ns ticks

ADC_MAX_BITS:
11 – 12bit ADC mode
9 – 10bit ADC mode
7 – 8bit ADC mode

MAROC_ID:
MAROC ASIC Identifier

0	3-MAROC PCB: ASIC 0, 2-MAROC PCB: ASIC 0
1	3-MAROC PCB: ASIC 1
2	3-MAROC PCB: ASIC 2, 2-MAROC PCB: ASIC 1

Word 1-32:

31	30	29	28	27	26	25	24
0	-	-	-	ADC_UPPER			
23	22	21	20	19	18	17	16
ADC_UPPER							
15	14	13	12	11	10	9	8
-	-	-	-	ADC_LOWER			
7	6	5	4	3	2	1	0
ADC_LOWER							

ADC_UPPER:
MAROC ADC value for channel: (Word-1)*2+1

ADC_LOWER:
MAROC ADC value for channel: (Word-1)*2+0

Note: 8bit and 10bit ADC modes are left justified (i.e. lower bits of 12bit reported value are unused and set 0)

Data Type: Data Not Valid

Type: 0x14
Size: 1 word
Description: Module has no data available for readout. This can if the module is being read out too quickly after receiving (event building is in process and no data words have been put into the buffer yet) a trigger or if the module doesn't have any events to report.

31	30	29	28	27	26	25	24
1	1	1	1	0	UNDEFINED		
23	22	21	20	19	18	17	16
UNDEFINED							
15	14	13	12	11	10	9	8
UNDEFINED							
7	6	5	4	3	2	1	0
UNDEFINED							

Data Type: Filler Word

Type: 0x15
Size: 1 word
Description: Non-data word appended to the block of events. This is used to force the total number of 32-bit words read out of a module to be a multiple of 2 or 4 when

31	30	29	28	27	26	25	24
1	1	1	1	1	UNDEFINED		
23	22	21	20	19	18	17	16
UNDEFINED							
15	14	13	12	11	10	9	8
UNDEFINED							
7	6	5	4	3	2	1	0
UNDEFINED							

7. Registers

Register Summary:

Register Name	Description	Address Offset

Document Revision History

7/14/2016:

1) Initial document release. Preliminary documentation a firmware under development and subject to change significantly as design progresses.

8 Power Supply and Current Consumption

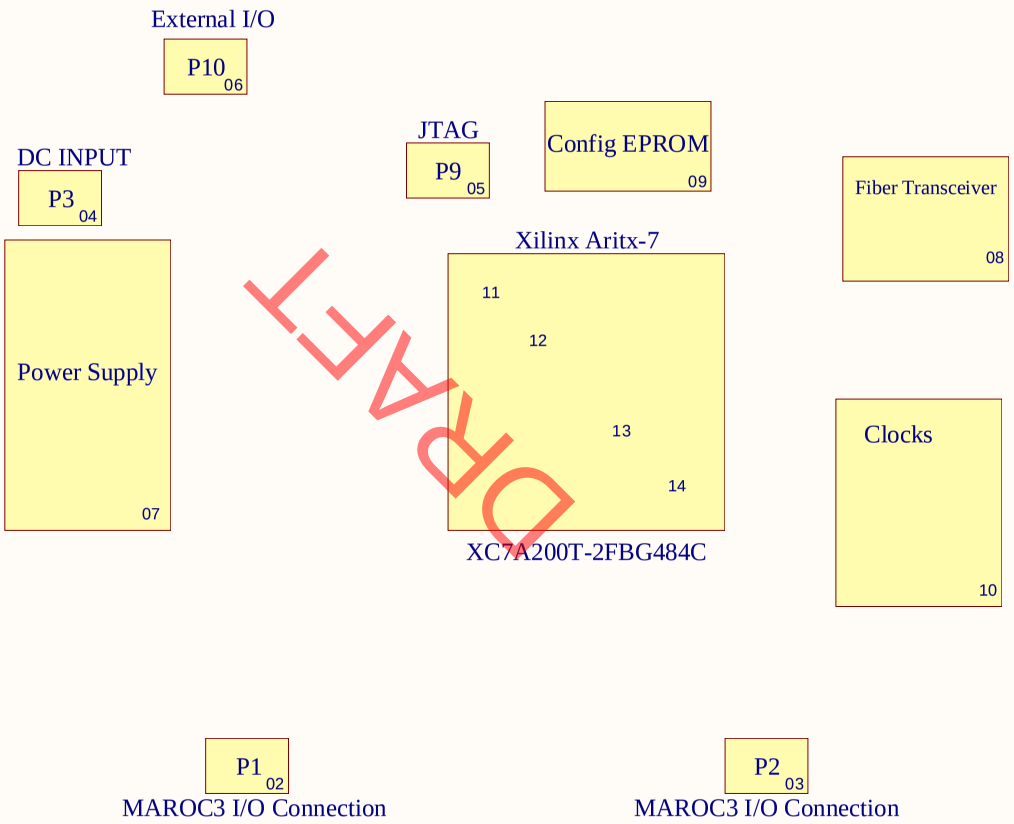
The total current required for the FPGA-ROC board is N, with an input Voltage range of X. There are several voltages that are created on board using voltage regulators. Table 2 lists these Voltage levels including a brief description of what the Voltage output is driving.

Appendix A: Schematics

RICH FPGA Readout Board

Schematic Sheets

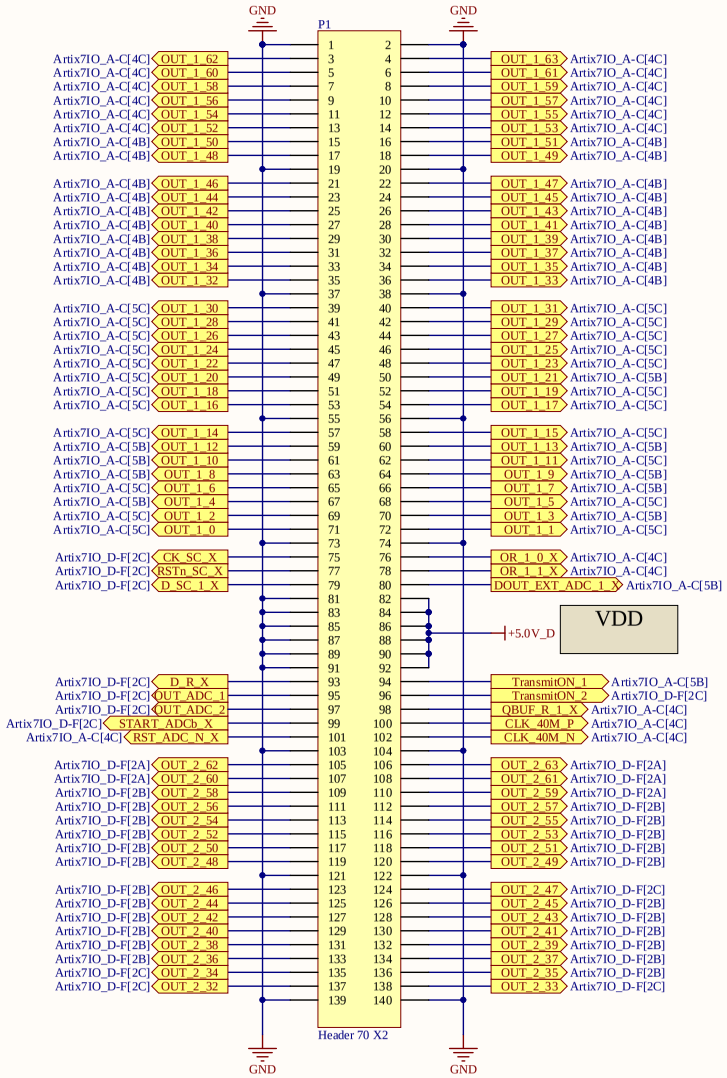
TOP Level RICH FPGA Readout Board Block Diagram
02) P1 I/O connections to ASiC board
03) P2 I/O connections to ASiC board
04) P3 Molex DC Input connector
05) P9 JTAG Interface connector
06) P10 External I/O connections
07) Power supply
08) Fiber Interface
09) Configuration EPROM
10) CLOCKS
11) Artix 7 BankA-C
12) Artix 7 BankD-F
13) Artix 7 BankG-I
14) Artix 7 BankJ-K

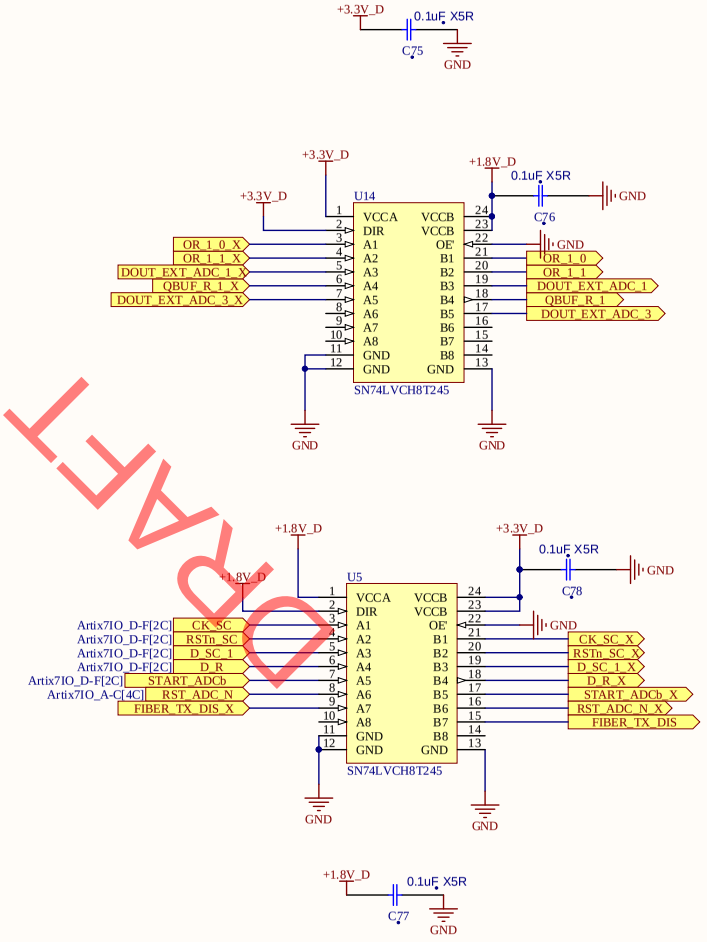


Title: *RICH FPGA Board*			*Jefferson Lab* *Fast Electronics* *12000 Jefferson Avenue* *Newport News, VA* *23606*
Size: B	Number:	Revision:	
Date: 2/11/2016 Time: 2:13:24 PM	Sheet 1 of 27		
File: RICH_3Tube_Block_Diagram.SchDoc			

Jefferson Lab
Thomas Jefferson National Accelerator Facility

P1 I/O connector

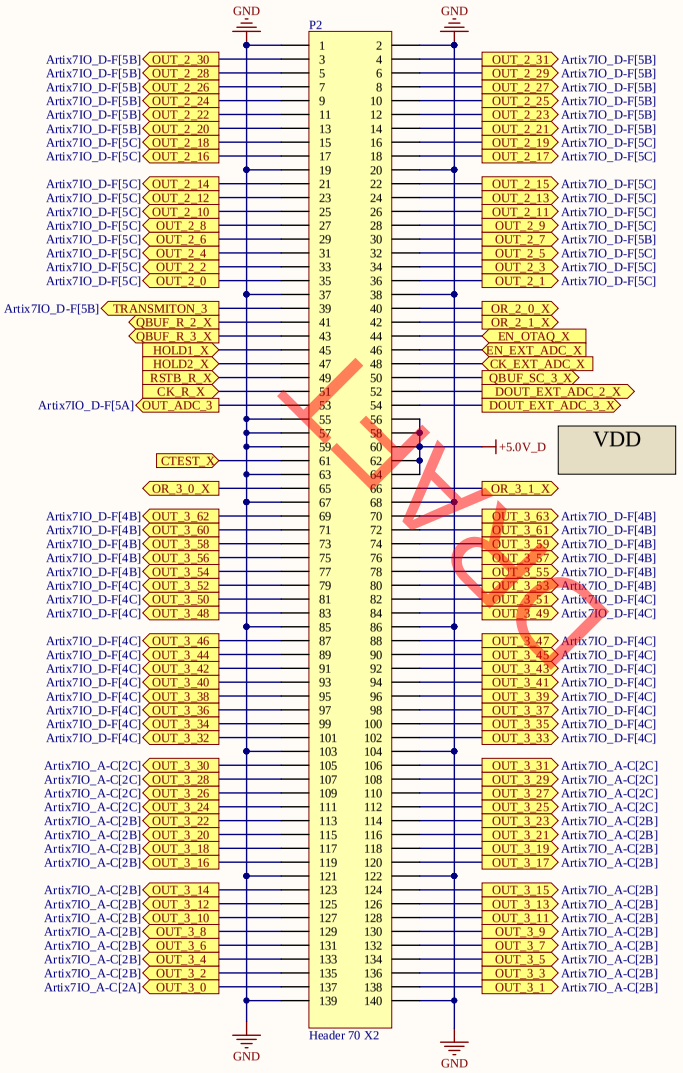


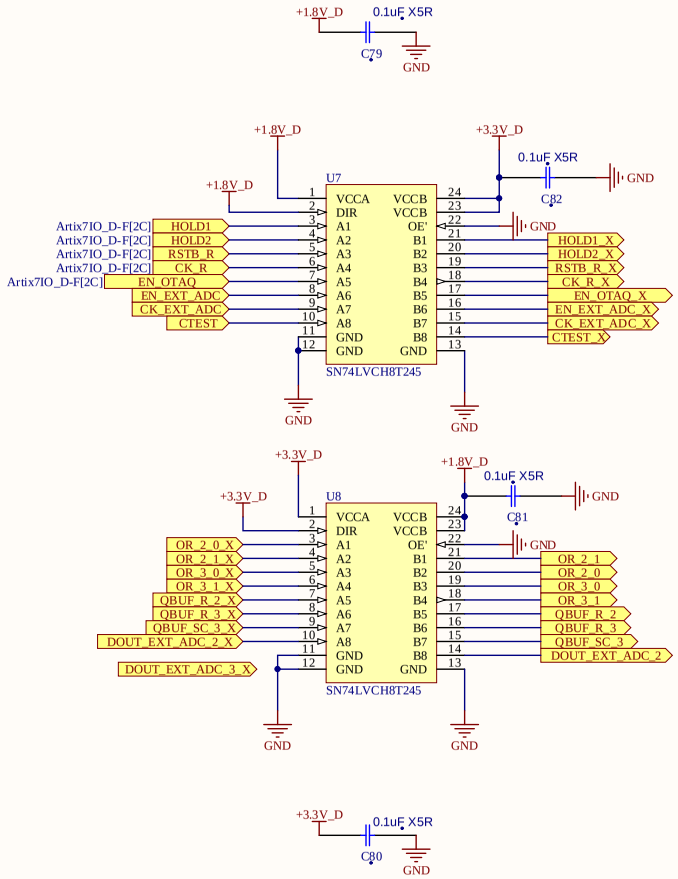


Title: **RICH FPGA Board**			Jefferson Lab Fast Electronics 12000 Jefferson Avenue Newport News, VA 23606
Size: B	Number:	Revision:	
Date: 2/11/2016	Time: 2:14:42 PM	Sheet 1 of 27	
File: P1Connector.SchDoc			

Jefferson Lab
Thomas Jefferson National Accelerator Facility

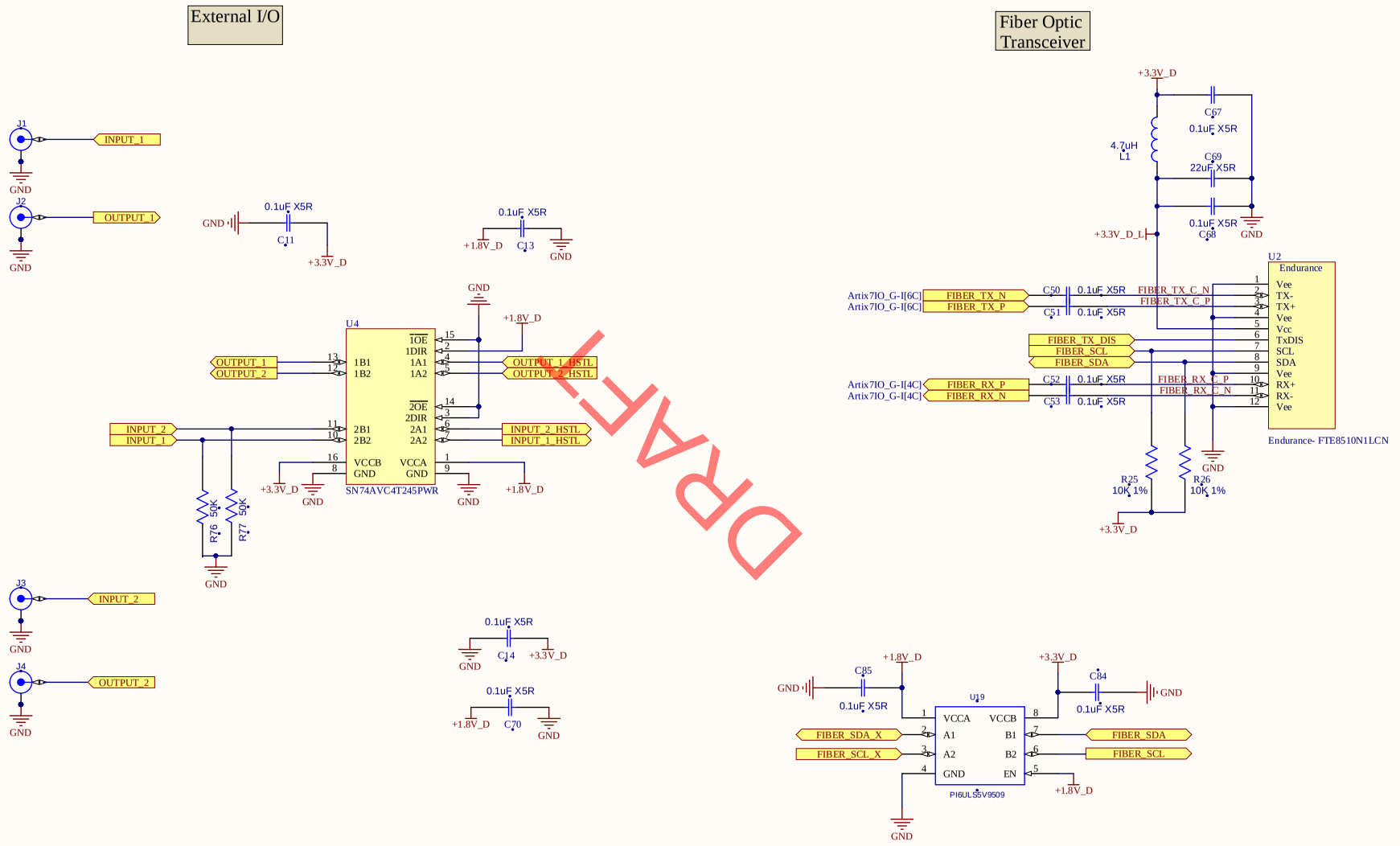
P2 I/O connector



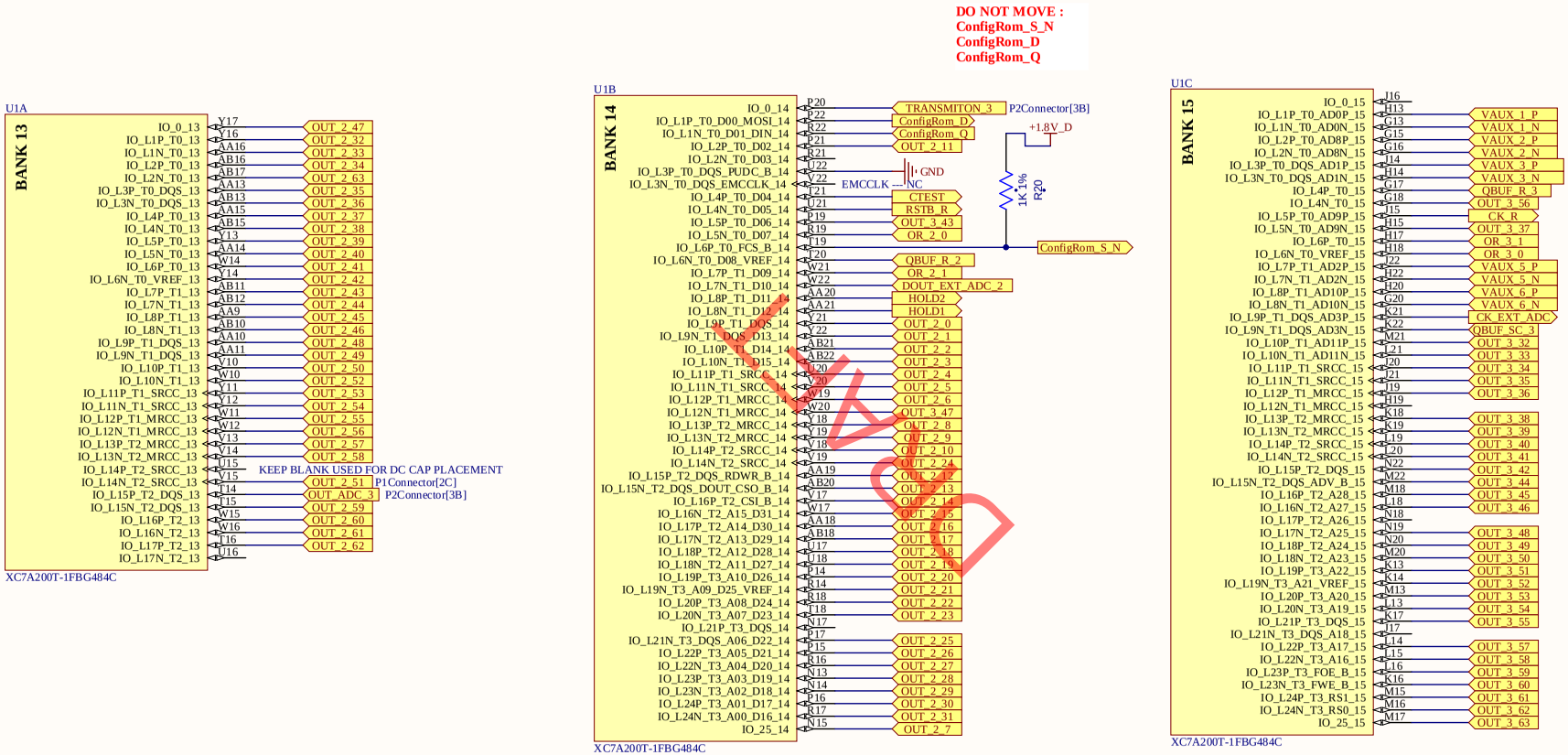


Title	**RICH FPGA Board**		*Jefferson Lab* *Fast Electronics* *12000 Jefferson Avenue* *Newport News, VA* *23606*
Size: B	Number:	Revision:	
Date: 2/11/2016	Time: 2:14:55 PM	Sheet 1 of 27	
File: P2Connector.SchDoc			

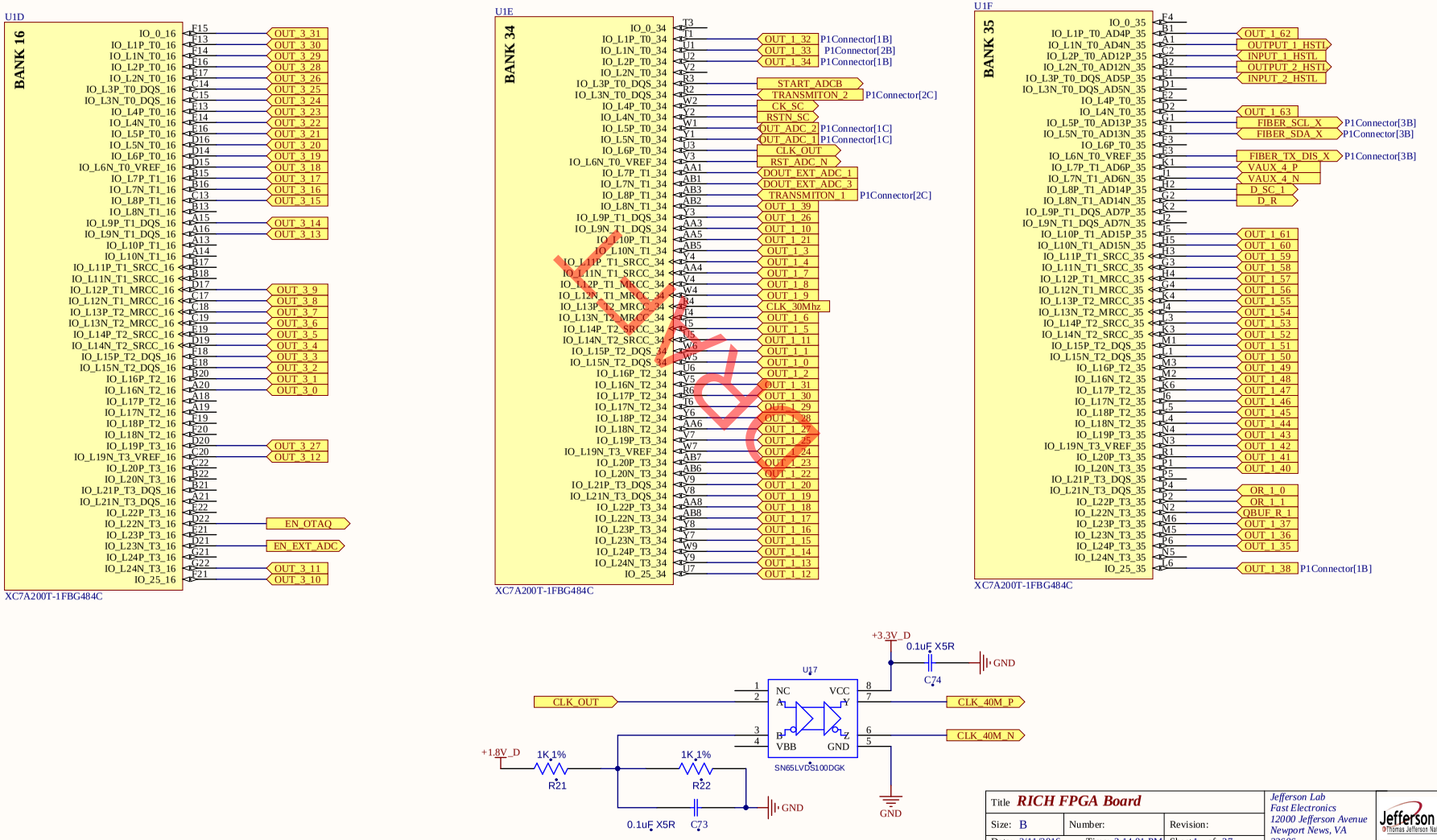
Jefferson Lab
Thomas Jefferson National Accelerator Facility


External I/O
Fiber Optic Transceiver
J1
INPUT_1
J2
OUTPUT_1
J3
INPUT_2
J4
OUTPUT_2
GND
0.1uF X5R
C11
C13
C14
C70
+3.3V_D
+1.8V_D
U4
SN74AVC4T245PWR
1OE
1DIR
1B1
1B2
1A1
1A2
2OE
2DIR
2B1
2B2
2A1
2A2
VCCB
VCCA
OUTPUT_1_HSTL
OUTPUT_2_HSTL
INPUT_2_HSTL
INPUT_1_HSTL
R76
R77
50K
DRAFT
+3.3V_D
C67
4.7uH
L1
C69
22uF X5R
C68
+3.3V_D_L
U2
Endurance
Endurance- FTE8510N1LCN
Artix7IO_G-I[6C]
Artix7IO_G-I[4C]
FIBER_TX_N
FIBER_TX_P
FIBER_RX_P
FIBER_RX_N
C50
C51
C52
C53
FIBER_TX_C_N
FIBER_TX_C_P
FIBER_RX_C_P
FIBER_RX_C_N
FIBER_TX_DIS
FIBER_SCL
FIBER_SDA
R25
R26
10K 1%
U19
PI6ULS5V9509
C85
C84
FIBER_SDA_X
FIBER_SCL_X
VCCA
VCCB
A1
A2
B1
B2
GND
EN
Title RICH FPGA Board
Size: B
Number:
Revision:
Date: 2/11/2016
Time: 2:13:32 PM
Sheet 1 of 27
File: External_Io.SchDoc
Jefferson Lab
Fast Electronics
12000 Jefferson Avenue
Newport News, VA
23606
Jefferson Lab
Thomas Jefferson National Accelerator Facility

ARTIX7 IO BANK 13, 14, 15 Assignments

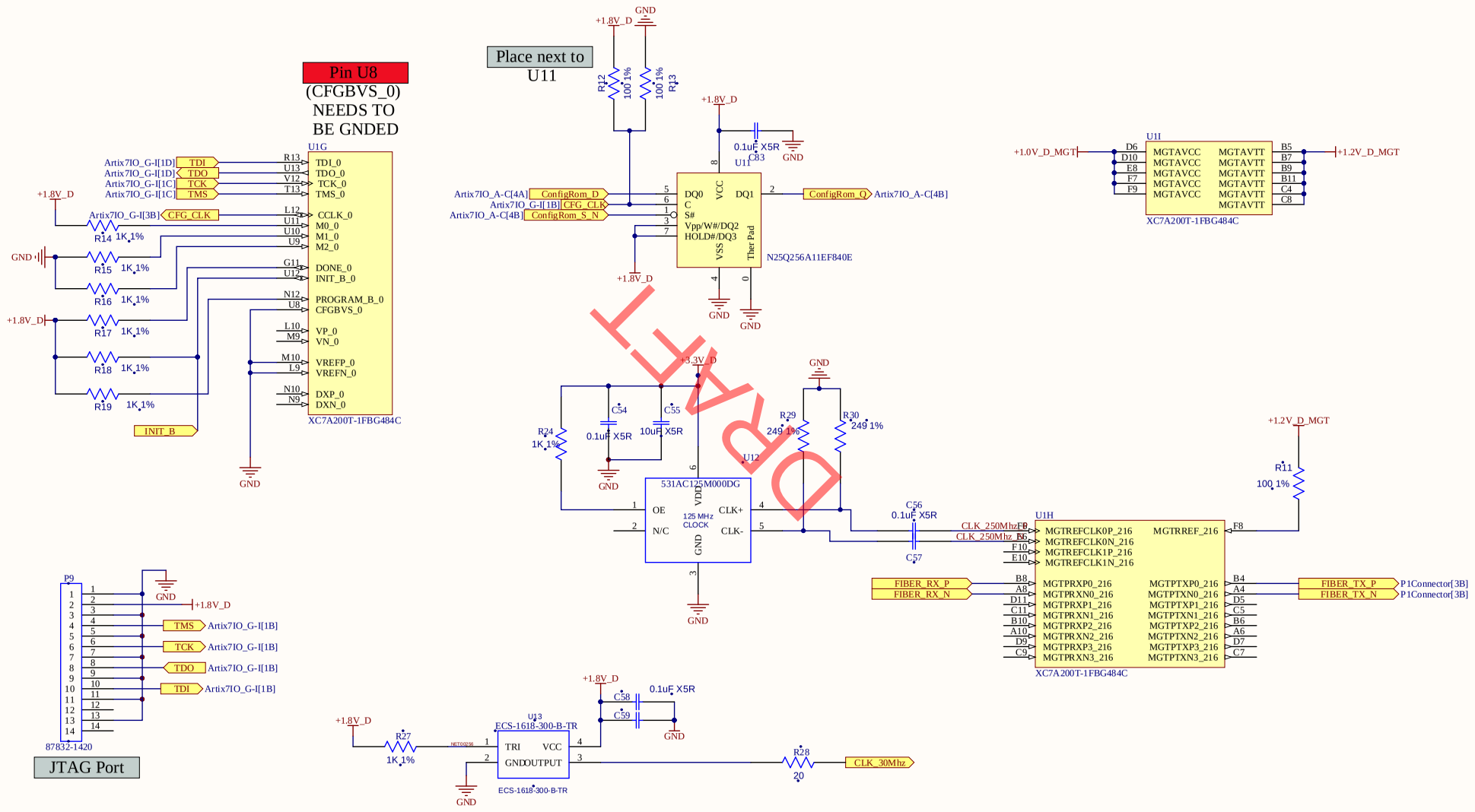


ARTIX7 IO BANK 16, 34, 35 Assignments



Title: **RICH FPGA Board**			Jefferson Lab Fast Electronics 12000 Jefferson Avenue Newport News, VA 23606
Size: B	Number:	Revision:	
Date: 2/11/2016	Time: 2:14:01 PM	Sheet 1 of 27	
File: Artix7IO_D-F.SchDoc			

ARTIX7 IO BANK G - I Assignments

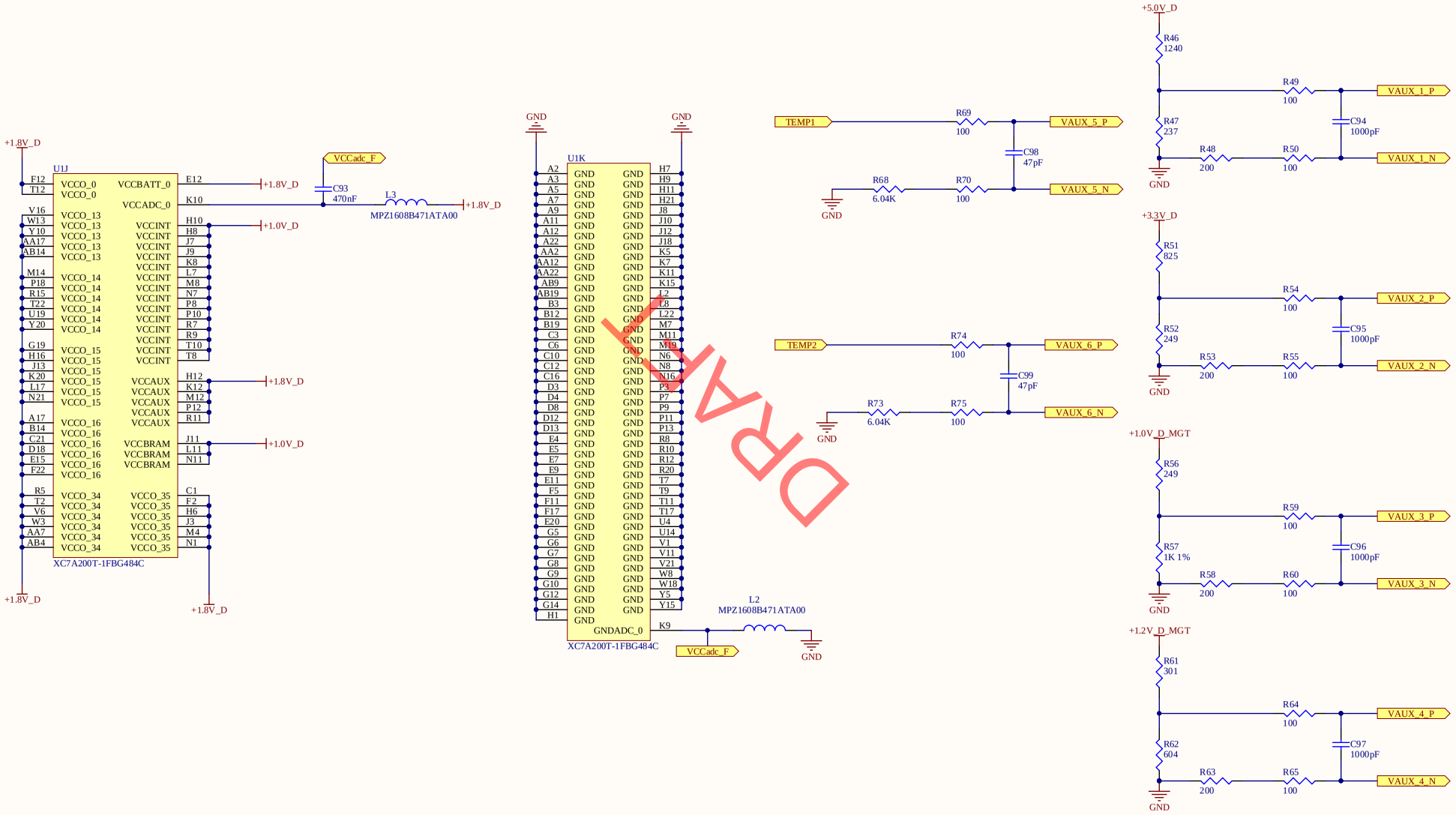


Title	**RICH FPGA Board**		
Size: B	Number:	Revision:	
Date: 2/11/2016	Time: 2:14:14 PM	Sheet 1 of 27	
File: Artix7IO_G-I.SchDoc			

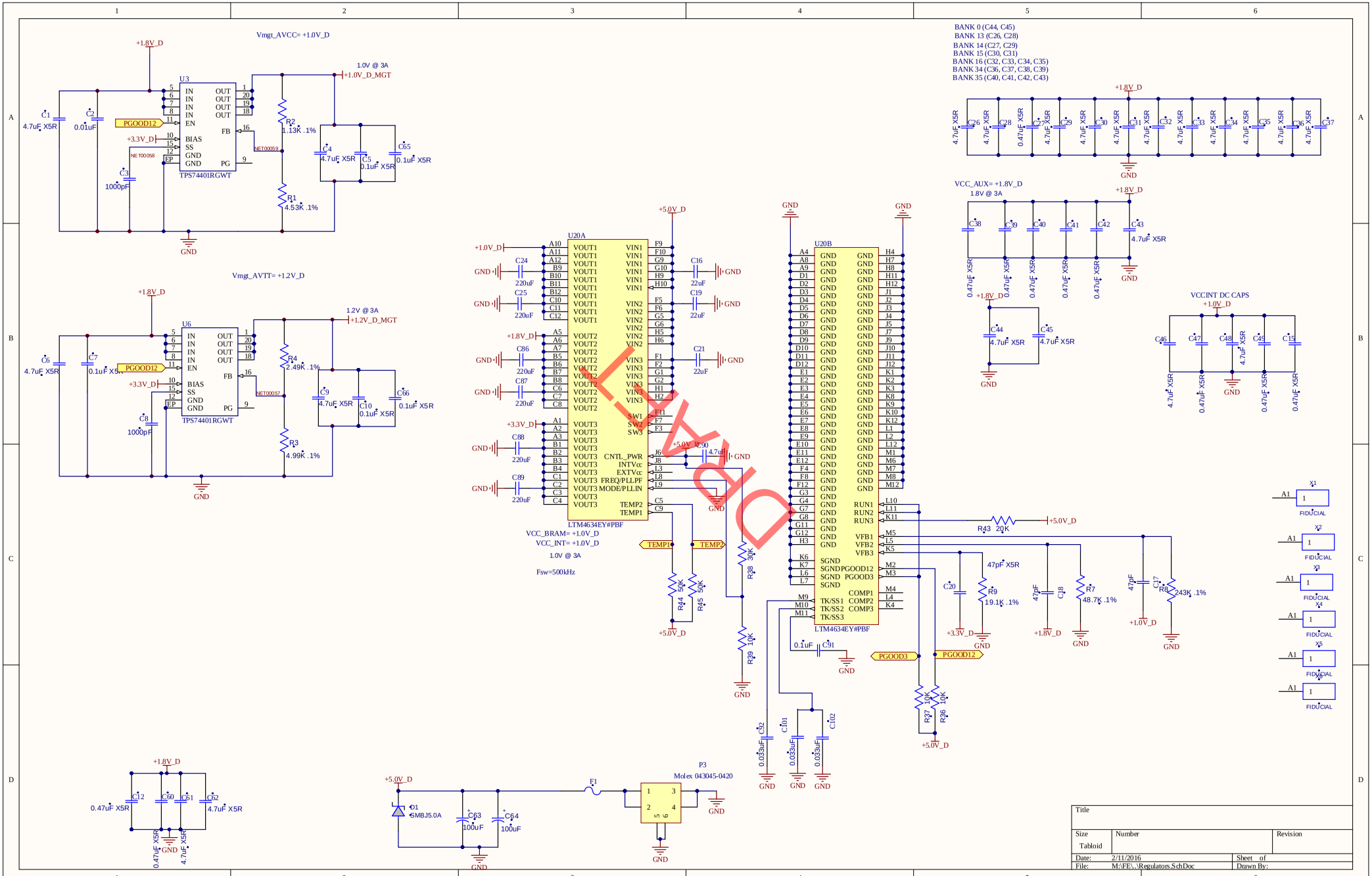
Jefferson Lab
Fast Electronics
12000 Jefferson Avenue
Newport News, VA
23606

Jefferson Lab
Thomas Jefferson National Accelerator Facility

ARTIX7 IO BANK J - K Assignments



Title	**RICH FPGA Board**		Jefferson Lab Fast Electronics 12000 Jefferson Avenue Newport News, VA 23606
Size: B	Number:	Revision:	
Date: 2/11/2016	Time: 2:14:28 PM	Sheet 1 of 27	
File: Artix7IO_J-K.SchDoc			


Vmgt_AVCC= +1.0V_D
+1.8V_D
1.0V @ 3A
+1.0V_D_MGT
U3
TPS74401RGWT
PGOOD12
+3.3V_D
NET00058
NET00059
R2 1.13K .1%
R1 4.53K .1%
C1 4.7uF X5R
C2 0.01uF
C3 1000pF
C4 4.7uF X5R
C5 0.1uF X5R
C65 0.1uF X5R
Vmgt_AVTT= +1.2V_D
1.2V @ 3A
+1.2V_D_MGT
U6
TPS74401RGWT
NET00057
R4 2.49K .1%
R3 4.99K .1%
C6 4.7uF X5R
C7 0.1uF X5R
C8 1000pF
C9 4.7uF X5R
C10 0.1uF X5R
C66 0.1uF X5R
U20A
LTM4634EY#PBF
VCC_BRAM= +1.0V_D
VCC_INT= +1.0V_D
1.0V @ 3A
Fsw=500kHz
+5.0V_D
+1.0V_D
+3.3V_D
+5.0V_D_0.90
TEMP1
TEMP2
R38 30K
R39 10K
R44 50K
R45 50K
C16 22uF
C19 22uF
C21 22uF
C24 220uF
C25 220uF
C86 220uF
C87 220uF
C88 220uF
C89 220uF
U20B
LTM4634EY#PBF
R43 20K
R9 19.1K .1%
R7 48.7K .1%
R8 243K .1%
C20 47pF X5R
C18 47pF
C17 47pF
C91 0.1uF
PGOOD3
PGOOD12
R37 10K
R36 10K
C32 0.033uF
C101 0.033uF
C102 0.033uF
BANK 0 (C44, C45)
BANK 13 (C26, C28)
BANK 14 (C27, C29)
BANK 15 (C30, C31)
BANK 16 (C32, C33, C34, C35)
BANK 34 (C36, C37, C38, C39)
BANK 35 (C40, C41, C42, C43)
C26 4.7uF X5R
C28 4.7uF X5R
C27 0.47uF X5R
C29 4.7uF X5R
C30 4.7uF X5R
C31 4.7uF X5R
C32 4.7uF X5R
C33 4.7uF X5R
C34 4.7uF X5R
C35 4.7uF X5R
C36 4.7uF X5R
C37 4.7uF X5R
VCC_AUX= +1.8V_D
1.8V @ 3A
C38 0.47uF X5R
C39 0.47uF X5R
C40 0.47uF X5R
C41 0.47uF X5R
C42 0.47uF X5R
C43 4.7uF X5R
C44 4.7uF X5R
C45 4.7uF X5R
VCCINT DC CAPS
C46 4.7uF X5R
C47 0.47uF X5R
C48 4.7uF X5R
C49 0.47uF X5R
C15 0.47uF X5R
X1 FIDUCIAL
X2 FIDUCIAL
X3 FIDUCIAL
X4 FIDUCIAL
X5 FIDUCIAL
C12 0.47uF X5R
C60 0.47uF X5R
C61 4.7uF X5R
C62 4.7uF X5R
D1 SMBJ5.0A
C63 100uF
C64 100uF
F1
P3
Molex 043045-0420
Title
Size Tabloid
Number
Revision
Date: 2/11/2016
Sheet of
File: M:\FE\..\Regulators.SchDoc
Drawn By:
COPY

Appendix B: Fabrication Drawing

FPGA DAQ and Control board.

Appendix C: Bill of Materials