



Nuclear Physics Division  
*Fast Electronics Group*

## **Description and Technical Information for the VME LED Driver (VLD) Module**

J. William Gu ([jgu@jlab.org](mailto:jgu@jlab.org))

Initial version: Sept. 14, 2020

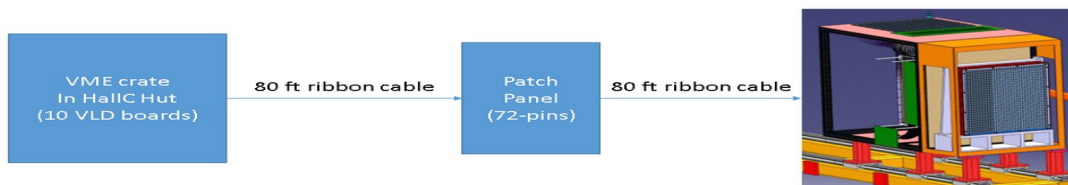
Last updated: Jan. 4, 2021

## Table of Contents

| <b>Section</b> | <b>Title</b>                          | <b>Page</b> |
|----------------|---------------------------------------|-------------|
| 1              | Introduction                          | 3           |
| 2              | Purpose of VLD Module                 | 4           |
| 3              | Function Description                  | 5           |
| 4              | Specifications                        | 7           |
| 5              | VLD operation                         | 9           |
| 6              | Programming Requirements              | 10          |
| 7              | Front Panel Connectors' Pinout tables | 13          |
| 8              | Examples of individual test           | 14          |
| 9              | Citation                              | 15          |
| Appendix A     | Schematics                            | 16          |
| Appendix B     | Fabrication Drawing                   | 25          |
| Appendix C     | Bill of Materials                     | 27          |
| Appendix D     | Revision history                      | 30          |

# 1 Introduction

The VME LED Driver (VLD) module is being designed for the scintillator calibration and bleaching of the NPS experiment in Jefferson Lab HallC [1]. The VLD boards are located in the VME crate in the radiation-shielded hut in the experimental hall. The VLD drives blue LEDs on the HV-LED board mounted on the back of the scintillator detector through ribbon cables totaling about 50 (max) meters in length. Figure1 shows the crate level diagram of VLD boards in the experiment.



**Figure 1 The VLD boards location and cables to 'power' the LEDs**

The VLD design will be based on the TI/TD [2] design for the VME interface with the ability of updating the firmware remotely, and uses the same FPGA (Xilinx Artix-7) as the VETROC [3] board for inexpensive and flexible control.

The blue LEDs (NSPB500AS by NICHIA) [4] are mounted on the HV-LED board on the back of the crystals (scintillators), and coupled with a quartz optic fiber to the back of each of the 1080 crystals of NPS detector. The Cathode of the LED is tied to the common ground, which simplifies the VLD design greatly (comparing with the initial assumption that the Anodes were tied to the common ground. The LED Anode is connected to the VLD output through several sections of the ribbon cables (and the PCB traces), totaling about 50 meters.

There are two working modes for the LED driver: the calibration mode, and the bleaching mode. In calibration mode, the LED can be powered up individually with an adjustable current of up to 110mA, the pulse rise time of several ns, and the pulse width adjustable (dozens of ns) to simulate a real scintillator signal.

In the bleaching mode, (to bleach off the scintillator radiation damage), all the channels can be powered up for up to several days (continuously) with the bleaching current of up to 35 mA per channel.

Here is the requirement documented by Carlos Munoz Camacho [4]:

*The calibration and optical bleaching of the NPS crystal imply the following requirements to the LED control system:*

- \_ Operate each LED both in continuous and pulsed mode.*
- \_ Be able to light up one single LED individually (in addition to 'all at once'), for both the continuous and pulsed modes.*
- \_ Be able to cycle all the channels, lighting up one LED at a time (in pulsed mode).*
- \_ Be able to trigger the DAQ on the pulsed mode operation.*
- \_ The system should have 'trigger/clock' NIM input in order to set the flash rate for enabled LEDs.*

*In continuous mode (for optical bleaching of the crystals) LEDs will need to be on for several hours (up to 48{72 hours).*

Here is the additional (clarifying) requirement documented by Brad Sawatzky [5]:

*1. Bleaching Mode: The LEDs will need to operate in two modes at 35 mA continuously for 48–72 hours during bleaching mode.*

*2. Pulse Mode: The LED pulses should be comparable in amplitude and timing to real pulses produced in the NPS crystal. Rise times on the order of a few nanoseconds, a pulse width on the order of 10–40 ns. Pulse amplitude is unclear (will depend strongly on the optical coupling of the LED → Fiber → crystal.*

*◦ It is not clear what level of granular control is desired over individual amplitude and pulse widths.*

*◦ It is not clear whether remote/programmable control of these pulse parameters is a “like” or a “must”.*

*3. Clarification for circuit drawing on p. 25:*

*◦ J1-1 and P1-1 are tied together and supply power to LED\_0*

*◦ J1-2 and P1-2 are the nominal common return for LED\_0 ie. are tied to the common ground plane)*

*◦ J1-3 and P1-3 are tied together and supply power to LED\_1*

*◦ J1-2 and P1-2 are the nominal common return for LED\_1 (ie. are tied to the common ground plane)*

***~~4. It is the anode (positive) pins on the LED that are tied to the common ground.~~***

***\*4. Later confirmed by the HV distribution PCB design that the LED cathodes are tied to common ground.***

*5. The ground of the LED boards are presently tied to the frame through screws (chassis ground). I have some concerns this will produce ground-bounce noise in adjacent PMTs that may be an issue.*

***\*(4): it is confirmed by the HV distribution PCB design that the LED cathodes are tied to common ground.***

In summary, there are totally 1080 channels (1080 LEDs distributed on 30 HV-LED boards with 36 per board). The total bleaching current is about 40 A continuous. There will be 10 VLD boards, with 108 channels each to cover all the 1080 LEDs (crystals).

Each VLD is estimated to cost about \$2.5K each for the order quantity of 12 (2 spares are added) and one prototype. The total would be about \$30K.

The additional components for the LED driver system include a VME crate with a CPU/VME\_controller, and many ribbon cables.

## **2 Purpose of VLD module**

The VLD board is loaded in the VME crate in the radiation-shielded hut in hall C as a VME slave board. Each VLD board drives 108 LEDs through the front panel connectors, and the attached long ribbon cables. The VLD can also accept external trigger signals, and initiate calibration pulses, or preprogrammed calibration sequences. The VLD can also generate DAQ readout trigger initiated by the LED calibration pulses.

The VLD can pulse any of the 108 channels with a programmable pulse (height and width). The VLD can also supply up to 35 mA continuous current to each LED, totaling about 4 A per VLD.

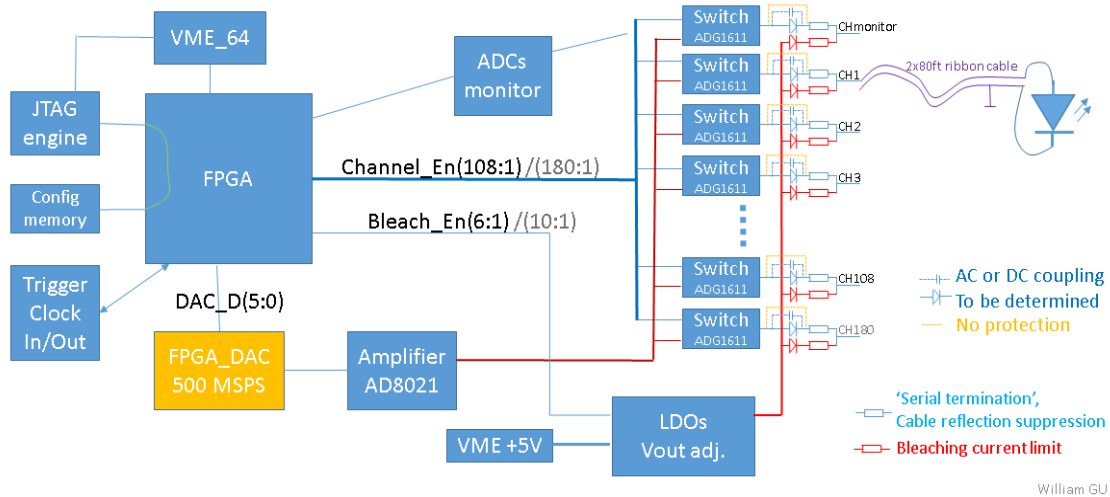
The VLD can work on its onboard oscillator clock, or the NIM input clock (to synchronize to a common clock). It can generate triggers when pulsing the LEDs, and it can also pulse the LEDs on the external trigger inputs. Figure 2 is a functional diagram of the VLD board.

## VME LED Driver (VLD) design

1080 (30x36) LEDs, through long ribbon cables.

- Calibration: individually controllable, up to 110 mA, ~ns pulses
- Bleaching: All ON, 35 mA, many hours

Design in progress



William GU

Figure 2 Functional diagram of VME LED Driver (VLD) board

## 3 Functional Descriptions

### 3.1 General description

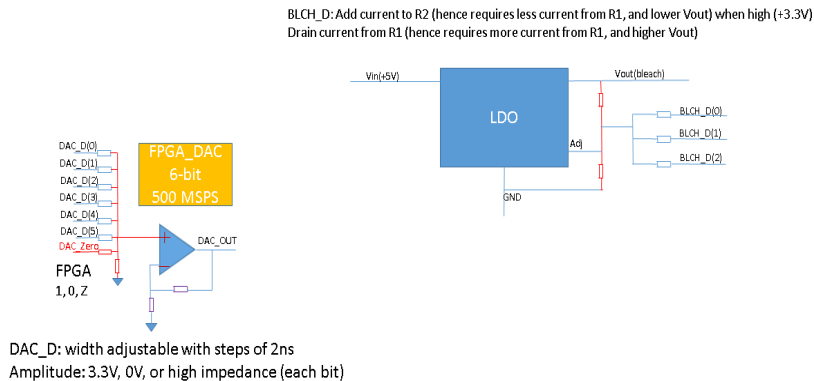
Figure 2 shows the block diagram of the VLD board, indicating the major components used in the design. The Xilinx XC7A200T is chosen for the controlling logic and VME interface, which has abundant resources, is less expensive, is LVTTTL33 I/O compatibility, and has been proven in the VETROC project.

### 3.2 Pulse shape generation (current/amplitude, and pulse width control):

The calibration pulse is generated by a high speed “DAC” working at 500 MSPS, which gives a precise user programmable shape (amplitude, rising time, falling time). The user can load the pulse into a memory in the FPGA.

Figure 3 shows the FPGA controlled DAC using an amplifier. The DAC (op-amp) output is further amplified/buffered per cables (36 LED channels). The buffered signals drives the LED through analog switches, which is used to disable/enable individual LEDs.

## VLD design



William GU

**Figure 3: FPGA DAC design**

### 3.3 Pulse (pulse mode) coupling to the ribbon cable:

Each channel is controlled by an analog switch, so that the calibration can be turned ON/OFF individually. Every 36 channels (one connector) share one pair of op-amps, which are connected in parallel to supply the needed current and maintain a faster rising time.

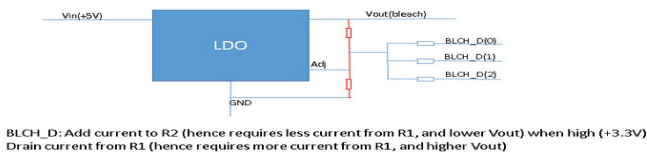
Two methods of coupling are being considered: AC and DC. In AC coupling, the pulse is coupled through a (0.1 uF) capacitor (for a <1 us pulse) to the current limiting resistor and the cable. In AC coupling mode, the voltage shift can be applied at the cable side of the capacitor. The disadvantage of the AC coupling is that the AC coupling would not work for crystal bleaching (almost DC current).

In DC coupling, a high speed diode may be to isolate from the bleaching mode driving circuit. A direct connection should be OK, if the bleaching circuit is diode isolated. If the diode is used for DC coupling, there will be an extra (~ 1V) dropout on the diode, and the voltage shift (for LED forward voltage) has to come from the amplifier too. This requires the op-amp output to have a larger swing, hence the diode coupling is not favorable.

### 3.4: Bleaching mode:

The amplifier (AD8021) is capable to drive 30 mA continuously, and even higher current with the current buffer circuit, but this design (using the calibration circuit for bleaching) is very inefficient, as most of the power will be consumed by the amplifier, and/or the current buffer for bleaching mode. It is impractical for the amplifier to work in the bleaching mode because of the VME power supply current limit. For the standard VME, each slot can consume 2A on the +12V (one VME connector pin). The bleaching mode requires 4 A for all the 108 channels. The solution is to use a DC-DC converter to get the bleaching using LDO regulators with adjustable output voltage. A diode is required to isolate the channels from each other (and from the pulsing mode current limiting resistor), though the bleaching current can be supplied by a common regulator (minimum partition). This can reduce the overall VLD power consumption to less than 30 Watt in full working (active) mode. The bleaching current is controlled by varying the LDO feedback current requirement (hence the LDO output voltage).

## VLD design



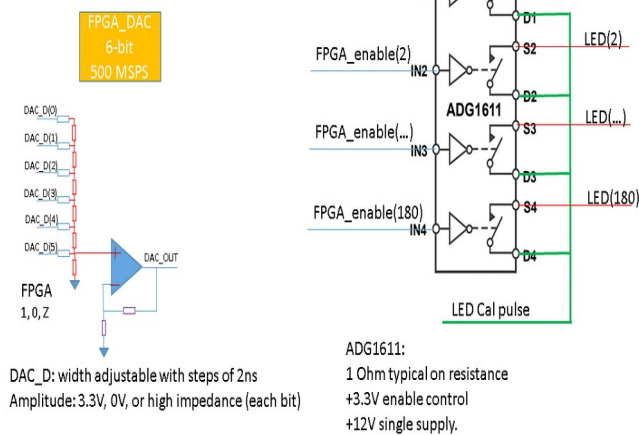
William GU

Figure 4 Bleaching voltage/current control

### 3.5: Channel enables:

For pulse mode: each channel can be enabled/disabled independently, and the channel enable/disable can be programmed to cycle through a certain sequence. When more than one channel are enabled, the enabled channels will have the same (amplitude and shape) pulse (calibration signals). The enabling/disabling is controlled by the analog switch, conduct or isolate.

## VLD design



William GU

Figure 5: ADG1611, anaog switch functional diagram

For bleaching mode, each connector (36 channels) can be enabled/disabled independently by enabling/disabling the LDO regulator directly.

### 3.6: Trigger/Clock Input and Output

The TTL inputs (of the trigger and clock) are received by a 3.3V receiver and connected to the FPGA. The clock input is connected to an FPGA clock capable pin. The clock output from the FPGA is driven to the LEMO connector (as TTL). The LEMO trigger output is the OR of the trigger input and the FPGA trigger output.

### 3.7 VME interface

The VLD board is a VME slave board. It is compatible with VME64x backplane, and the payload slots in the VXS crate. Normally, it is a VME slave board, with interrupt capability.

For simplicity, two kinds of VME address modifier codes are implemented. (1), User defined address modifier (0x19, 0x1A, 0x1C and 0x1D). This is similar to the A24 address modifier. It is used to load the FPGA programming memory by the 'emergency VME to JTAG engine' logic. (2), Standard A24 address modifier. This is used to set or readout the registers on the FPGA, slow controls of the VLD peripherals. This is implemented the same way as VETROC, except that there is no need to generate DAQ data for the VLD board.

### 3.8: FPGA programming/PROM loading

The FPGA XC7A200T needs about 80Mbit flash memory to configure. This configure data is saved in the Micron MT25QL256A (N25Q256 for VETROC, but the flash memory is discontinued), which is 256 Mbit. The flash memory can save two versions of the FPGA firmware. This makes it possible for a fall back design, which is especially useful when the board is used in an area that is difficult to access (needs to be tested).

The FPGA is programmed in Master SPI mode with external clock of 50 MHz and 4-bit wide data loading (SPIx4). The expected FPGA program time is less than one second. The Micron flash memory can be loaded by the Xilinx iMPACT software through the on-board JTAG connector. The iMPACT software will load a special firmware to the FPGA through the JTAG connector and program the memory through the special firmware, which Xilinx calls it the indirect flash memory programming.

The FPGA and the Flash memory can be remotely loaded (through VME). To make it more robust for remote programming, a hardware (discrete logic) VME to JTAG engine is implemented on the board (copied from TS/TI/TD design) using the custom defined address modifier code (AM = 19), which will not get confused with the standard (VME specified) A24 address modifier codes. This engine can load the FPGA firmware even if the memory is corrupted (or simply, the memory is empty) or the FPGA failed to be loaded by the flash memory. The JTAG engine has been tested successfully on the VETROC. It takes about 30 seconds to load the FPGA through VME. It takes about ten minutes to load (program) the Flash memory (Micron N25Q256). In this JTAG engine, the VME data bit[1] is used for TDI, bit[0] is used for TMS, and all the other bits are unused. The higher bit of A24 address should match with the geographic address, and the lower A24 address is set to be 0x0FFFC. When the VLD is plugged into a standard VME64 crate, (no geographic address for slots), the on-board switch is used to set the 'geographic address'.

For more flexibility, two Micron flash memories are loaded on the board, and the flash memory is selected by a hardware switch just like that of the VETROC board. Two people can design independent firmware for the same piece of hardware (PCB).

### 3.9 Other functions

The VLD has several front panel LEDs to indicate the VLD status.

## 4. Specification Sheet

### 4.1 Mechanical

- Single width, 6Ux160mm VME64x slave module. It can be plugged into the VXS payload slots, or a standard VME64 crate slots (except slot#1).

### 4.2 Front panel inputs and outputs:

- Up to 250MHz (TTL) Clock Input & Output



- Trigger (TTL) Input & Output
- 108 channels (on three connectors) LED driver Output

### **4.3 LED Indicators: Front Panel:**

- Bit 1 (top): FPGA programmed and the clock (DCM locked) is ready;
- Bit 2: VME DTACK, VME activity;
- Bit 3: Trigger is active (either external input, or FPGA generated);
- Bit 4: flashing: Calibration pulse active, ON: bleaching active.

#### **On board:**

- Power OK near each regulator or DC-DC converter (LED is OFF when the power is OK);
- FPGA program DONE (LED is OFF when programmed);

### **4.4 Programming:**

- VME to JTAG A24D32 with user defined AM (Address Modifier) for remote loading with redundant On board JTAG connector;
- Custom VME to JTAG engine implemented in the FPGA using A24D32 for firmware loading;
- Up to four revisions of the firmware can be stored in the flash memories simultaneously.

### **4.5 Power requirements:**

- +5v @ 6 Amps; -12V @ 1 Amps; +12V @ 1 Amps (from VME P1 Backplane connector), all are fused.
- DC-DC converters for +3.3V (from +5V), +2.5V/3.3V, +1.2V, +1.0V, and LDO for +1.8V FPGA BRAM and +1.0V FPGA MGT power.

### **4.6 Environment:**

- Forced air cooling: Weiner standard VME64x/VXS
- Commercial grade (minimum) components ( 0-75 Celsius)

Figure 6 shows a typical populated VLD board: using the PCB layout as a place holder.

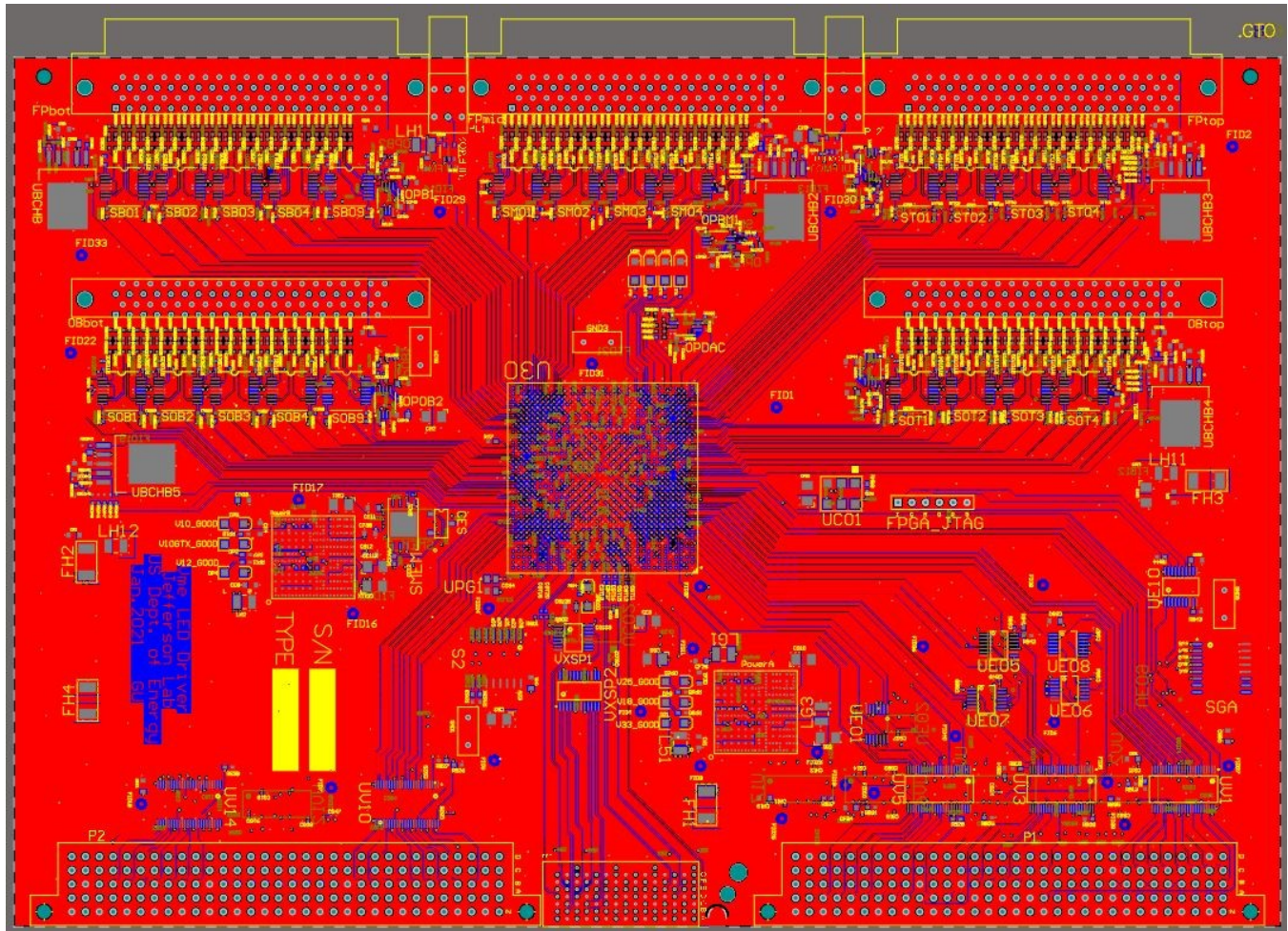


Figure 6: The VLD board (using the PCB layout as a placeholder)

## 5 VLD operation procedures:

The VLD needs to be properly set, and to be plugged into the proper crate and slot. Damage may happen to the VLD, the crate, or other PCBs in the crate if the right procedure is not followed.

### 5.1 VLD Power supply:

The VLD uses +5V, +12V, and -12V from the VME P1 connector, and generates +3.3V, +2.5V, +1.8V, +1.2V and +1.0V by the onboard DC-DC regulators or LDO regulators.

### 5.2 Hardware setting (Switch etc.):

CES: This is a one-bit, three-position switch used to select the flash memory to load the FPGA. The VLD FPGA can be loaded by two independent flash memories with SPIx4 interface. The switch SF needs be set to select the right flash memory. When the three position switch is set in the middle, no flash memory is selected, hence the FPGA will not be programmed.

S2: This is an 8-bit ON/OFF switch with pull up resistors (to +3.3V, so OFF = High, ON = Low).

- Bit[8:4]=A[23:19], VME address space in A24 mode. When the VLD is in a VME64x compatible crate, these bit settings do not matter, and the geographic address is used to match with A[23:19].

When the VLD is in a standard VME crate, the Bit[8:4] is usually set to its location in the VME crate (less confusing when addressing the VLD) to match with A[23:19] for VME address space.

- Bit[3:2]: to be determined
- Bit[1]: Clock source selection: 0: to select the on-board oscillator, 1: to select the external NIM input, this setting can be over-written by the software (register 0x2C).

SGA: This is an 8-bit ON/OFF switch with pull up resistors (to +3.3V, so OFF = High, ON = Low).

- Bit[8:4]= GA(4:0) if the VLD is plugged into a standard VME/VME64 crate (without Geographic address). These five bits should be set to high (OFF) if the VLD is plugged into a VME64x (or VXS) crate. This address is used in VME emergency JTAG engine (flash memory loading).
- Bit[3:1]: not used

### **5.3 Software setting:**

The VLD is powered up to a default low-power (inactive) state. No user interference is required if the user does not want to power up the LEDs on the detector.

Here are some example procedures to power up the LEDs.

#### **5.3.1 Scintillator calibration initiated by the internal command:**

- Load the pulse shape (to the FPGA memory) if the default pulse shape is not the right one, by write to VME address 0x6C repeatedly;
- Load the channel enable pattern (bits(18:0) of addresses 0x40, 0x44, 0x48, 0x4C, 0x50, 0x54, 0x58, 0x5C, 0x60, and 0x64); (bit0 should be the OR of the 18 channels bit(18:1) )
- Load the 'trigger' source enable register (0x20) to use internal pulsing;
- Load the number of pulses and period (address 0x8C), or load the random pulsing period (0x88)

#### **5.3.2 Scintillator calibration initiated by the external trigger:**

- Load the pulse shape (to the FPGA memory) if the default pulse shape is not the right one, by write to VME address 0x6C repeatedly;
- Load the channel enable pattern (bits(18:0) of addresses 0x40, 0x44, 0x48, 0x4C, 0x50, 0x54, 0x58, 0x5C, 0x60, and 0x64);
- Load the 'trigger' source enable register (0x20) to use external pulsing;
- Enable the external pulsing (this operation is NOT on the VLD)

#### **5.3.3 Scintillator calibration initiated by the internal sequence:**

To speed up the calibration, the channels can be cycled through automatically by enabling the internal calibration sequence.

- Load the pulse shape (to the FPGA memory) if the default pulse shape is not the right one, by write to VME address 0x6C repeatedly;
- Load the 'trigger' source enable register (0x20) to use automatic cycling through;
- Load the number of pulses per channel and period (address 0x8C).

#### **5.3.4 Scintillator bleaching:**

- Load the bleaching time (register address 0x68);
- Load the channel enable pattern (bits(31:28) of addresses 0x40, 0x44, 0x48, 0x4C, 0x50, and 0x54) by blocks of 18 channels;

## **6. VME Programming Requirements (This part will be updated as the firmware develops)**

There are two categories of Address Modifier codes supported on the VLD: the user-defined codes (A24) for emergency firmware loading; standard A24 for FPGA register read/write and slow control (JTAG engine and I2C engine).

## 6.1 VME to JTAG (flash memory) emergency loading engine:

The AM[5:0] user defined codes are used for this logic. This works even before the FPGA is programmed and working. It is almost the same as A24D32 mode. The valid AM codes are: 0x19, 0x1A, 0x1D and 0x1E. These AM codes are user defined, and similar to the AM codes 0x39, 0x3A, 0x3D and 0x3E.

The valid address bits are A[31:24] do not care; A[23:19]=GA[4:0] for VME64x crates, or A[23:19]=0 for non-VME64x crates; A[18:2]=b'0001111111111111.

Data bit[1] is TDI; data bit[0] is TMS.

For example, if the board is in slot#5 (that is ~GA(4:0)= 11010), you need write to A(23:0)=0x28fffc. If data(1:0)=00, both TMS and TDI will be low; if data(1:0)=01, TMS is high, TDI is low; if data(1:0)=10, TMS is low, TDI is high; if data(1:0)=11, both TDI and TMS are high. The normal A24 address should try to avoid this address (0x0fffc). The TCK is one cycle (low  $\Rightarrow$  High  $\Rightarrow$  low) on every VME write cycle.

A more advanced example: Instruction register shift (8-bit, shift in 0x5a) starting from/end up at the 'reset idle' JTAG state: 14 consecutive writes to the address 0x28fffc with AM=0x19, 1a, 1d or 1e, the data are 1, 1, 0, 0, 0, 2, 0, 2, 2, 0, 2, 1, 1, 0 respectively.

|      |    |    |    |    |   |   |   |   |   |   |   |   |    |    |
|------|----|----|----|----|---|---|---|---|---|---|---|---|----|----|
| Data | 1  | 1  | 0  | 0  | 0 | 2 | 0 | 2 | 2 | 0 | 2 | 1 | 1  | 0  |
| TMS  | H  | H  | L  | L  | L | L | L | L | L | L | L | H | H  | L  |
| TDI  | 0x | 0x | 0x | 0x | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0x | 0x |

- “TMS H” means logic High, “TMS L” means logic Low, “TDI 0” means 0 or Low, “TDI 1” means 1 or High, and “TDI 0x” means DO NOT CARE by the JTAG, but the set value is 0.

## 6.2 Configuration Registers:

A24D32 are used for register read/write. Similar to the emergency loading logic, the base address is determined by the Geographic Address in VME64x/VXS crate, and external switch for non-VME64x crate. That is, A[23:19]=GA[4:0], or SG[8:4].

➤ Address offset: 0x00000: Board ID:

Bit 7-0 (R/W): Crate ID; Reset default 0x00;

Bit 12-8 (R): A24 address, higher 5 bits; Reset default 000

Bit 13 (R): VME Crate info, 1: VME64x compatible, 0: standard VME;

Bit 15-14 (R): reserved;

Bit 23-16 (R): PCB related setting, 0x01: production board, 0x00; prototype board;

Bit 31-24 (R): Board type: 0x1D: vme Led Driver (VLD) board.

➤ Address offset: 0x0020 (R/W): Trigger source selection, default: 0, no trigger is enabled

Bit 0: Enable internal trigger, periodic (0x8C);

Bit 1: Enable internal trigger, random (0x88);

Bit 2: Enable internal trigger sequence;

Bit 3: not used;

Bit 4: Enable external trigger input;

Bit (31:5): not used.

➤ Address offset: 0x002C (R/W): Clock source selection, default: 0, oscillator clock

Bit 0: Clock source selection, default: 0

0: on-board oscillator;

1: External NIM input.

Bit(31:1): not used

- Address offset: 0x0040 (R/W): LED calibration channel enable, default: 0xffffffff, all disabled

Bit (18:0): Connector#1 (front panel, top), Channels#18-#1; 0: enable, 1: disable

Bit (27:19): not used

Bit (31:28): Bleach enable and Bleach current control

- Address offset: 0x0044 (R/W): LED calibration channel enable, default: 0xffffffff, all disabled

Bit (18:0): Connector#1(front panel, top), Channels#36-#19; 0: enable, 1: disable

Bit (27:19): not used

Bit (31:28): “1011”: These 36 channel bleaching enable, otherwise bleaching disabled

- Address offset: 0x0048 (R/W): LED calibration channel enable , default: 0xffffffff, all disabled

Bit (18:0): Connector#2(front panel, middle), Channels#18-#1; 0: enable, 1: disable

Bit (27:19): not used

Bit (31:28): Bleach enable and Bleach current control

- Address offset: 0x004C (R/W): LED calibration channel enable , default: 0xffffffff, all disabled

Bit (18:0): Connector#2(front panel, middle), Channels#36-#19; 0: enable, 1: disable

Bit (27:19): not used

Bit (31:28): “1011”: These 36 channel bleaching enable, otherwise bleaching disabled

- Address offset: 0x0050 (R/W): LED calibration channel enable , default: 0xffffffff, all disabled

Bit (18:0): Connector#3(front panel, bottom), Channels#18-#1; 0: enable, 1: disable

Bit (27:19): not used

Bit (31:28): Bleach enable and Bleach current control

- Address offset: 0x0054 (R/W): LED calibration channel enable , default: 0xffffffff, all disabled

Bit (18:0): Connector#3(front panel, bottom), Channels#36-#19; 0: enable, 1: disable

Bit (27:19): not used

Bit (31:28): “1011”: These 36 channel bleaching enable, otherwise bleaching disabled

- Address offset: 0x0058 (R/W): LED calibration channel enable , default: 0xffffffff, all disabled

Bit (18:0): Connector#4(Onboard connector, top), Channels#18-#1; 0: enable, 1: disable

Bit (27:19): not used

Bit (31:28): Bleach enable and Bleach current control

- Address offset: 0x005C (R/W): LED calibration channel enable , default: 0xffffffff, all disabled

Bit (18:0): Connector#4(on-board connector, top), Channels#36-#19; 0: enable, 1: disable

Bit (27:19): not used

Bit (31:28): “1011”: These 36 channel bleaching enable, otherwise bleaching disabled

- Address offset: 0x0060 (R/W): LED calibration channel enable , default: 0xffffffff, all disabled
  - Bit (18:0): Connector#5(on-board connector, bottom), Channels#18-#1; 0: enable, 1: disable
  - Bit (27:19): not used
  - Bit (31:28): Bleach enable and Bleach current control
- Address offset: 0x0064 (R/W): LED calibration channel enable , default: 0xffffffff, all disabled
  - Bit (18:0): Connector#3(on-board connector, bottom), Channels#36-#19; 0: enable, 1: disable
  - Bit (27:19): not used
  - Bit (31:28): “1011”: These 36 channel bleaching enable, otherwise bleaching disabled
- Address offset: 0x0068 (R/W): Bleaching setting, default:
  - Bit [31:0]: Bleaching timer (unit: second).
- Address offset: 0x006C (R/W): Pulse shape loading, default: 4ns rising, 25ns falling edge pulses
  - Bit [14:0]: DAC#1 pulse shape loading, the sample# automatically counts up. 1024 (or smaller) samples at 2ns per sample  $\Rightarrow$  total pulse width:  $\sim 2\mu s$ ;
  - Bit [15]: DAC#1 pulse shape loading reset (to sample#1)
  - Bit [30:16]: DAC#2 pulse shape loading, the sample# automatically counts up. 1024 samples at 2ns per sample  $\Rightarrow$  total pulse width:  $\sim 2\mu s$ ;
  - Bit [31]: DAC#2 pulse shape loading reset (to sample#1)
- Address offset: 0x0070 (R): Monitoring ADC readout
  - Bit [15:0]: ADC readout of Pulse amplitude,
  - Bit [31:16]: ADC readout of Bleaching current.
- Address offset: 0x0074 (R): Monitoring VLD board
  - Bit [9:0]: ADC readout of of VLD +5V supply current,
  - Bit [19:10]: ADC readout of VLD +12V supply current,
  - Bit [29:20]: ADC readout of VLD -12V supply current.
- Address offset: 0x0088 (R/W): Random Pulsing, default 0,
  - Bit [7:0]: Random pulsing speed setting (average period);
  - Bit [31:8]: not used
- Address offset: 0x008C (R/W): Periodic pulsing, default 0,
  - Bit [15:0]: number of repeat for periodic pulsing,
  - Bit [31:16]: Pulsing period
- Address offset: 0x00DC (R): Total number of triggers,
  - Bit [31:0]: number of trigger.
- Address offset: 0x00100 (W): Reset and one-shot registers. The signal will be one VME clock cycle. If the VME clock is 25 MHz, the one-shot will be 40ns wide. Positive logic.
  - Bit 1: if ‘1’, RESET signal to reset the VME\_to\_I2C engine;

Bit 2: if '1', RESET signal to reset the VME\_to\_JTAG engine;  
 Bit 4: if '1', RESET signal to reset the VME registers (VLD settings) to their default values;  
 Bit 8: if '1', Reset the CLK250/Clk200 DCM.  
 Bit 10: if '1', Reset the MGT (MultiGigabit Transceiver,) inside the FPGA (for JESD204).

### 6.3 VME to Serial engines:

A24D32 are used for VME to serial engines. The two engines are “VME to JTAG engine for the FPGA”, and “VME to I2C engine for the P2 connector (could be used for anything)”. In the I2C engines design, only the lower one-byte or 2-byte of the 32-bit data word is used. The higher bytes are not used. The A(1:0) are required to be “00”.

Address offset A(18:0): 0x2XXXX: JTAG for FPGA;  
 Address offset A(18:0): 0x4XXXX: I2C for the P2 connector;

## 7 Front panel IO Connectors' Pinout tables:

### 7.1 Front panel FPtop, FPmid, and FPbot Pinout Table (3M N3432-D302RB or 3M P50E-080P1-SR1), all are outputs. On-board connector OBtop, and OBbot

| Pin name | Signal Description | Signal Level | Direction |
|----------|--------------------|--------------|-----------|
| #1       | Ch#1 LED driver    | #2           | GND       |
| #3       | Ch#2 LED driver    | #4           | GND       |
| #5       | Ch#3 LED driver    | #6           | GND       |
| #7       | Ch#4 LED driver    | #8           | GND       |
| #9       | Ch#5 LED driver    | #10          | GND       |
| #11      | Ch#6 LED driver    | #12          | GND       |
| #13      | Ch#7 LED driver    | #14          | GND       |
| #15      | Ch#8 LED driver    | #16          | GND       |
| #17      | Ch#9 LED driver    | #18          | GND       |
| #19      | Ch#10 LED driver   | #20          | GND       |
| #21      | Ch#11 LED driver   | #22          | GND       |
| #23      | Ch#12 LED driver   | #24          | GND       |
| #25      | Ch#13 LED driver   | #26          | GND       |
| #27      | Ch#14 LED driver   | #28          | GND       |
| #29      | Ch#15 LED driver   | #30          | GND       |
| #31      | Ch#16 LED driver   | #32          | GND       |
| #33      | Ch#17 LED driver   | #34          | GND       |
| #35      | Ch#18 LED driver   | #36          | GND       |
| #37, #39 | GND                | #38, #40     | GND       |
| #41      | Ch#19 LED driver   | #42          | GND       |
| #43      | Ch#20 LED driver   | #44          | GND       |
| #45      | Ch#21 LED driver   | #46          | GND       |
| #47      | Ch#22 LED driver   | #48          | GND       |
| #49      | Ch#23 LED driver   | #50          | GND       |
| #51      | Ch#24 LED driver   | #52          | GND       |

|        |                  |        |     |
|--------|------------------|--------|-----|
| #53    | Ch#25 LED driver | #54    | GND |
| #55    | Ch#26 LED driver | #56    | GND |
| #57    | Ch#27 LED driver | #58    | GND |
| #59    | Ch#28 LED driver | #60    | GND |
| #61    | Ch#29 LED driver | #62    | GND |
| #63    | Ch#30 LED driver | #64    | GND |
| #65    | Ch#31 LED driver | #66    | GND |
| #67    | Ch#32 LED driver | #68    | GND |
| #69    | Ch#33 LED driver | #70    | GND |
| #71    | Ch#34 LED driver | #72    | GND |
| #73    | Ch#35 LED driver | #74    | GND |
| #75    | Ch#36 LED driver | #76    | GND |
| #77#79 | GND              | #78#80 | GND |

## 7.2 LEMO connector:

|         |              |         |                |
|---------|--------------|---------|----------------|
| LEMO#11 | Clock_input  | LEMO#12 | Trigger_input  |
| LEMO#21 | Clock_output | LEMO#22 | Trigger_output |

## 8 VLD Operation examples:

The following is some operating procedures at VxWorks interactive mode. This is just a memo for the quick test of the VLD board. They may change as the VLD debug proceeds. First, one needs to login the VME controller. Here is the sequence:

From any xterm on daq network computers, telnet to DAVW5 (neither username, nor password is needed). Only one telnet process is supported for the MVME6100 module). The address mapping for A24 is 0x90xxxxxx.

Then, load the custom vme Address Modifier library to the CPU: `ld < usrTempeDma_AM.o`

If necessary, load the user software code: `ld < trigger.o`

The following commands are assuming that the VLD is in slot#13 VME64x compatible crate (Geographic address available). `0x68 = 01101xxx`, that is the Geographic Address `GA=13` or `01101`.

### 8.1 A24 register echoing (write and read):

➔ `*(0x90680000)=0x64`; the same register should be read out, with Bit(31:8) non-zero.

### 8.2 Readout the FPGA user\_ID:

The FPGA user code is readout through VME\_to\_JTAG engine. The FPGA user\_ID is firmware specific. If this code matches, it is VLD and the right version of the firmware is loaded.

- ➔ `*(0x90680100)=0x04`: VME\_to\_JTAG engine logic reset
- ➔ `*(0x906a003C)=0x0`: Reset FPGA JTAG to 'reset\_idle' state
- ➔ `*(0x906a092C)=0x3C8`: enable user\_ID readback
- ➔ `*(0x906a1F1C)=0x00`: shift in 32-bit data, and the readback is user\_ID. The user\_ID should be `1D0xnmmm`: n is the major version of the firmware, mmm is the revision of the firmware.

### 8.3 I2C operation:



The VME\_to\_I2C engine hides the serialization, and the two-bytes write (and/or read) can be finished by a single VME write (and/or read) command. The I2C chain is 100; for device 1101000, the I2C device at the FPGA is 000; assuming the byte address is 0xAD (10101101).

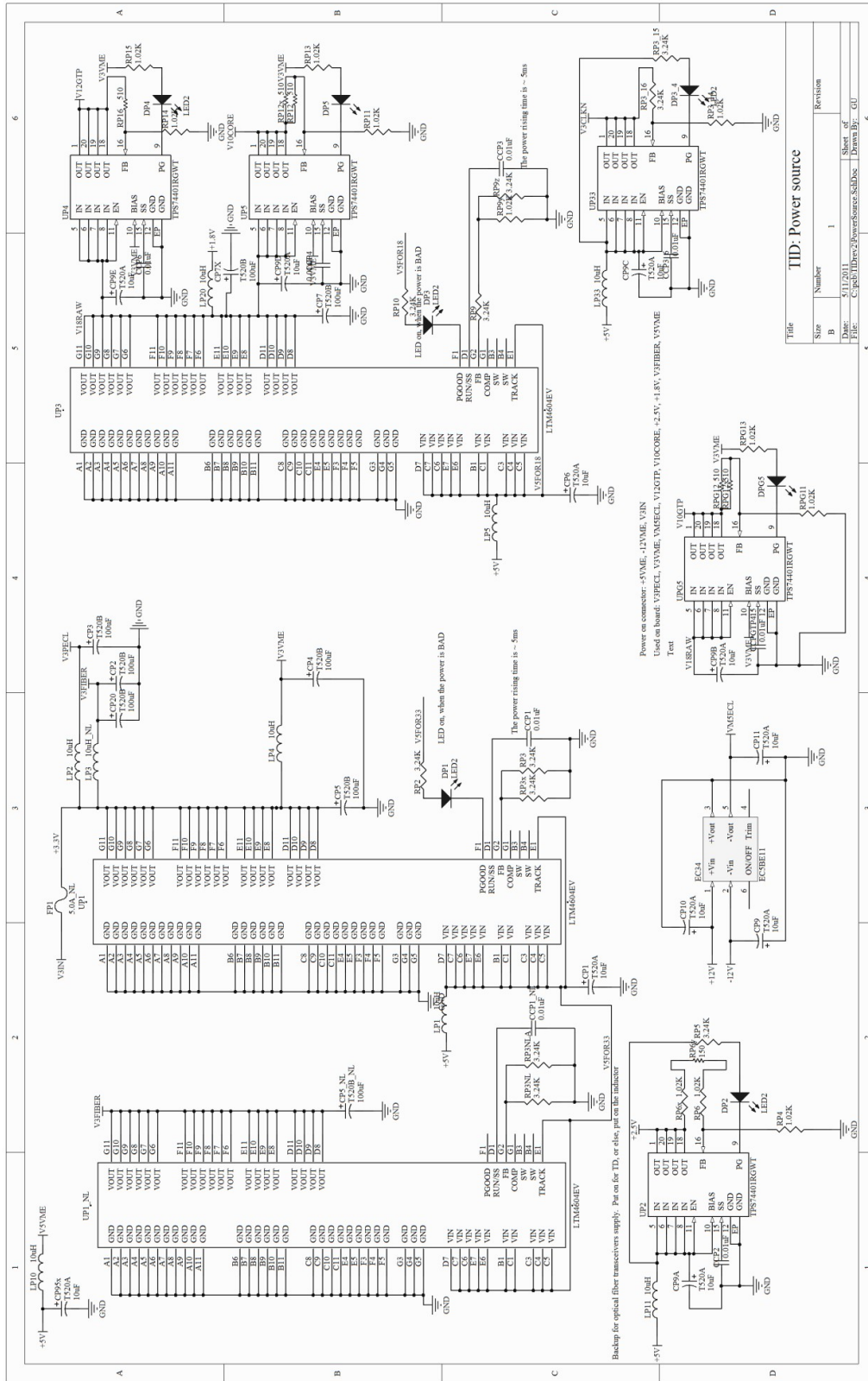
- ➔ \*(0x90680100)=0x01: VME\_to\_I2C engine reset
- ➔ \*(0x906C1EB4)=0xDADA: VME\_to\_I2C write
- ➔ m 0x906C1EB4, 4: VME to I2C read. ignore the higher two bytes
- ➔ \*(0x90680004)=0x100: Set the I2C address to 0b1101xxx
- ➔ \*(0x906C1eb4)=0xdata: VME to I2C write

## 9. Citations:

### Works Cited

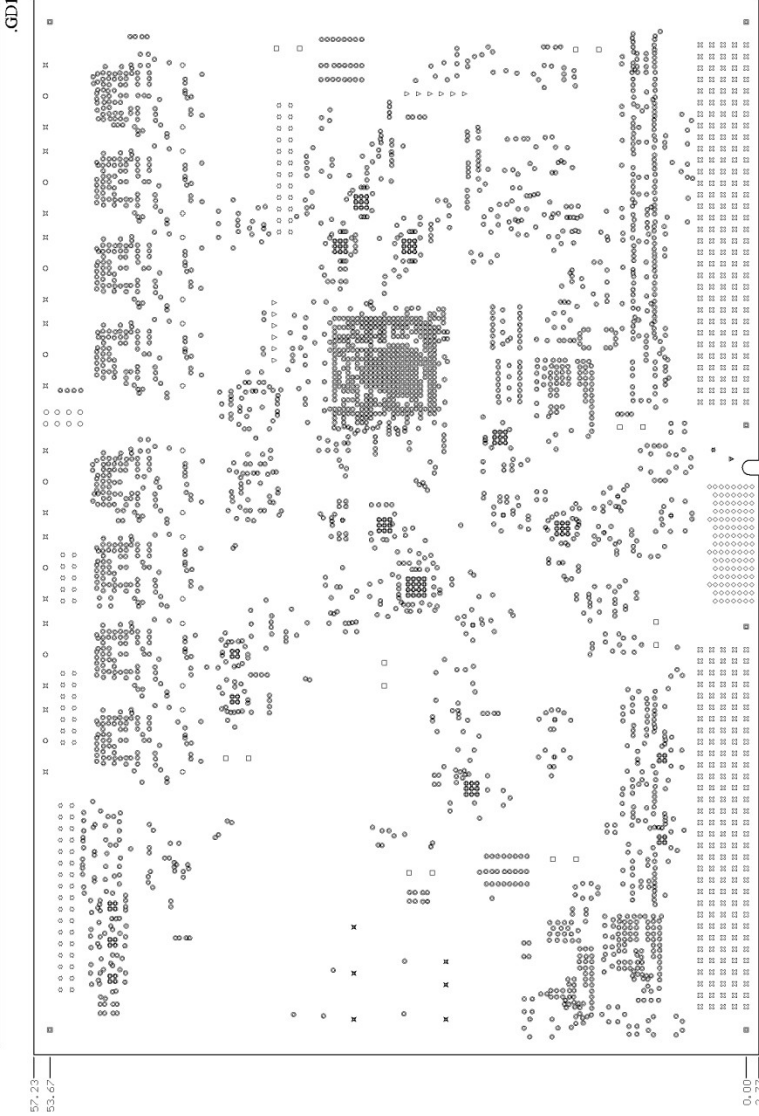
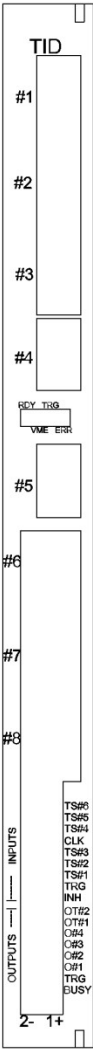
1. NPS experiment
2. GU, TI/TD design document
3. GU, VETROC design document
4. Carlos, LED driver requirement
5. Brad, Further explanation of LED driver requirement
6. GU, VME to JTAG engine implementation
7. GU, VME to I2C engine implementation

# Appendix A: VLD Schematics:





Appendix B: VLD fabrication drawing:



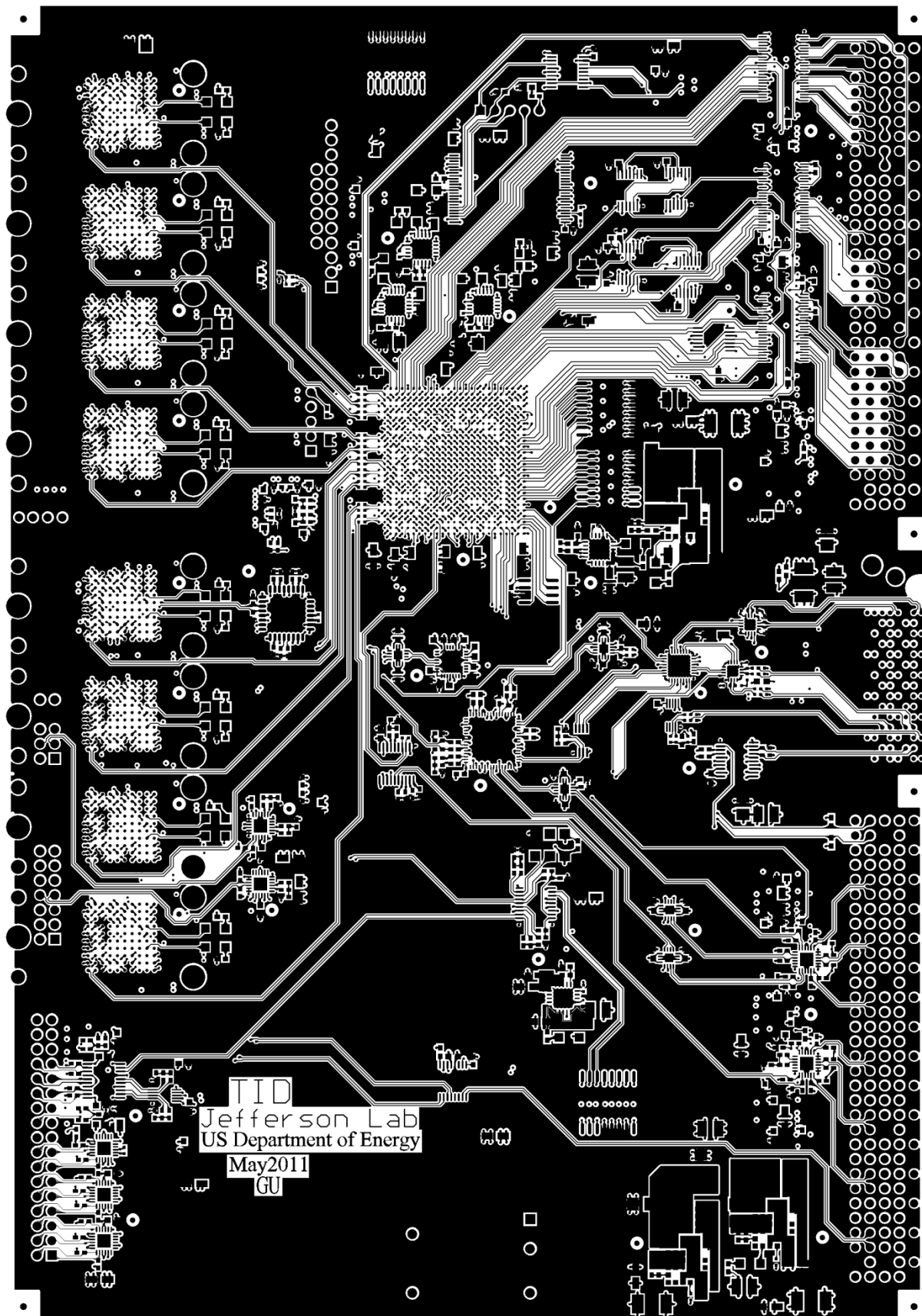
**DRILL FACILITIES & SIZES**  
The following table lists the drill facilities and sizes used in the fabrication of the VLD. The facilities are listed in the first column, and the sizes are listed in the second column. The sizes are given in both inches and millimeters.

| Facility              | Size               |
|-----------------------|--------------------|
| 1. 1/16" Dia. Drill   | 0.0625" (1.5875mm) |
| 2. 1/8" Dia. Drill    | 0.125" (3.175mm)   |
| 3. 1/4" Dia. Drill    | 0.25" (6.35mm)     |
| 4. 3/8" Dia. Drill    | 0.375" (9.525mm)   |
| 5. 1/2" Dia. Drill    | 0.5" (12.7mm)      |
| 6. 5/8" Dia. Drill    | 0.625" (15.875mm)  |
| 7. 3/4" Dia. Drill    | 0.75" (19.05mm)    |
| 8. 7/8" Dia. Drill    | 0.875" (22.225mm)  |
| 9. 1" Dia. Drill      | 1.0" (25.4mm)      |
| 10. 1 1/8" Dia. Drill | 1.125" (28.575mm)  |
| 11. 1 1/4" Dia. Drill | 1.25" (31.75mm)    |
| 12. 1 3/8" Dia. Drill | 1.375" (34.925mm)  |
| 13. 1 1/2" Dia. Drill | 1.5" (38.1mm)      |
| 14. 1 5/8" Dia. Drill | 1.625" (41.275mm)  |
| 15. 1 3/4" Dia. Drill | 1.75" (44.45mm)    |
| 16. 1 7/8" Dia. Drill | 1.875" (47.625mm)  |
| 17. 2" Dia. Drill     | 2.0" (50.8mm)      |
| 18. 2 1/8" Dia. Drill | 2.125" (53.975mm)  |
| 19. 2 1/4" Dia. Drill | 2.25" (57.15mm)    |
| 20. 2 3/8" Dia. Drill | 2.375" (60.325mm)  |
| 21. 2 1/2" Dia. Drill | 2.5" (63.5mm)      |
| 22. 2 5/8" Dia. Drill | 2.625" (66.675mm)  |
| 23. 2 3/4" Dia. Drill | 2.75" (69.85mm)    |
| 24. 2 7/8" Dia. Drill | 2.875" (73.025mm)  |
| 25. 3" Dia. Drill     | 3.0" (76.2mm)      |

| Symbol | Hit Count | Tool Size          | Plated | Hole Type |
|--------|-----------|--------------------|--------|-----------|
| ●      | 2903      | 0.3mm (11.811mil)  | PTH    | Round     |
| ○      | 115       | 0.33mm (12.992mil) | PTH    | Round     |
| ◇      | 109       | 0.56mm (22.047mil) | PTH    | Round     |
| □      | 16        | 0.7mm (27.559mil)  | PTH    | Round     |
| ▽      | 12        | 0.9mm (35.433mil)  | PTH    | Round     |
| ⊠      | 320       | 0.9652mm (38mil)   | PTH    | Round     |
| ⊙      | 82        | 1mm (39.37mil)     | PTH    | Round     |
| ⊖      | 8         | 1.016mm (40mil)    | PTH    | Round     |
| ⊗      | 6         | 1.27mm (50mil)     | PTH    | Round     |
| ⊘      | 16        | 1.7018mm (67mil)   | PTH    | Round     |
| ⊙      | 24        | 2.6924mm (106mil)  | PTH    | Round     |
| ⊖      | 6         | 2.794mm (110mil)   | PTH    | Round     |
| ⊗      | 1         | 2.794mm (110mil)   | NPTH   | Round     |
| ⊘      | 1         | 3.2mm (125.984mil) | NPTH   | Round     |
|        | 3619      | Total              |        |           |

**DRILL FILES**  
1) 1/16" Dia. Drill - 1/16" Drill Bit  
2) 1/8" Dia. Drill - 1/8" Drill Bit  
3) 1/4" Dia. Drill - 1/4" Drill Bit  
4) 3/8" Dia. Drill - 3/8" Drill Bit  
5) 1/2" Dia. Drill - 1/2" Drill Bit  
6) 5/8" Dia. Drill - 5/8" Drill Bit  
7) 3/4" Dia. Drill - 3/4" Drill Bit  
8) 7/8" Dia. Drill - 7/8" Drill Bit  
9) 1" Dia. Drill - 1" Drill Bit  
10) 1 1/8" Dia. Drill - 1 1/8" Drill Bit  
11) 1 1/4" Dia. Drill - 1 1/4" Drill Bit  
12) 1 3/8" Dia. Drill - 1 3/8" Drill Bit  
13) 1 1/2" Dia. Drill - 1 1/2" Drill Bit  
14) 1 5/8" Dia. Drill - 1 5/8" Drill Bit  
15) 1 3/4" Dia. Drill - 1 3/4" Drill Bit  
16) 1 7/8" Dia. Drill - 1 7/8" Drill Bit  
17) 2" Dia. Drill - 2" Drill Bit  
18) 2 1/8" Dia. Drill - 2 1/8" Drill Bit  
19) 2 1/4" Dia. Drill - 2 1/4" Drill Bit  
20) 2 3/8" Dia. Drill - 2 3/8" Drill Bit  
21) 2 1/2" Dia. Drill - 2 1/2" Drill Bit  
22) 2 5/8" Dia. Drill - 2 5/8" Drill Bit  
23) 2 3/4" Dia. Drill - 2 3/4" Drill Bit  
24) 2 7/8" Dia. Drill - 2 7/8" Drill Bit  
25) 3" Dia. Drill - 3" Drill Bit

Drilling Details.



.GTL

## Appendix C: VLD Bill of materials:

# Component list

Source Data From:

Project:

Variant:

Report

Date: 5/12/2011

Print Date: 12-May-11

| Description                                    | Footprint  | Quantity | Designator  |
|--|------------|----------|---|
| Capacitor                                      | 0402       | 11       | C0001, C0002, C0003, C0004, C0007, C0008, C0013, C0014, C0017, C0018, CUC04A  |
| Capacitor (Semiconductor SIM Model)            | 1608[0603] | 28       | C0005, C0006, CCL01, CCL02, CCL03, CCL04, CCL05, CCL06, CCL07, CCL08, CCL09, CCL11, CCL12, CCL13, CCL14, CCL15, CCL16, CCL17, CCL18, CP01, CP02, CP03, CP04, CP05, CP06, CP07, CUP07A, CUP07B   |
| Capacitor, Capacitor (Semiconductor SIM Model) | 0402       | 121      | C9510A, C9510B, C9510C, C9510D, CCAU1, CCAU2, CCAU3, CCAU4, CCAU5, CCAU6, CCAU7, CCAU8, CCF01, CCF02, CCF03, CCF04, CCF11, CCF12, CCF13, CCF14, CCF15, CCF16, CCF17, CCF18, CCF21, CCF22, CCF23, CCF24, CCF25, CCF26, CCF27, CCF28, CCF31, CCF32, CCF33, CCF34, CCF35, CCF36, CCF37, CCF38, CCF41, CCF42, CCF43, CCF44, CCF45, CCF46, CCF47, CCF48, CCF51, CCF52, CCF53, CCF54, CCF55, CCF56, CCF57, CCF58, CCF61, CCF62, CCF63, CCF64, CCF65, CCF66, CCF67, CCF68, CCF71, CCF72, CCF73, CCF74, CCF75, CCF76, CCF77, CCF78, CCF81, CCF82, CCF83, CCF84, CCF85, CCF86, CCF87, CCF88, CCF101, CCF102, CCF103, CCF104, CCF119, CCF129, CCF139, CCF149, CCF159, CCF169, CCF179, CCF189, CCLK04, CCLK05, CCT11, CCT12, CFB21, CFB22, CFB31, CFB32, CFB41, CFB42, CFB61, CFB62, CFB71, CFB72, CFB81, CFB82, CGTP1, CGTP2, CU201, CU202, CU203, CU204, CU205, CU206, CU207, CU208, CUC04, CUC08A, CUC08B |

## **Appendix D: Document revision history:**

Document started on Sept. 4<sup>th</sup>, 2020: Using the TI documentation format;

First release on Sept. 14<sup>th</sup>, 2020: to circulate to the NPS group for design feedback

Jan. 4, 2021: Added the FPGA DAC and Bleach regulator controls. Confirmed that the LED's Cathodes are tied to common ground.