

# VME to I<sup>2</sup>C Implementation on TID

J. William Gu

Data Acquisition Group

Jan. 7, 2011: Remove the options and minimize the protocol

Nov. 19, 2010: re-arrange the bits (A(9:2) for byte address

Oct. 5, 2010: Swap the START/STOP bit to match with the JTAG engine

Sept. 20, 2010: document the implemented (simulated in ISE) design

Sept. 2, 2010: Expanded the VME address, set the I<sup>2</sup>C byte address in the VME address.

Aug. 26, 2010: Initial release

## Introduction:

In the JLAB VXS crate, there is no VME bus in the switch slots, switch slot#A and switch slot#B. The data readout and switch board configuration cannot be implemented using VME bus. Two wires are used to implement an I<sup>2</sup>C interface between the Payload slot#18 and each switch slot. For the 12-GeV upgrade, the Trigger Interface & Distribution (TID) is located in Payload slot#18, and the Crate Trigger Processor (CTP), Signal Distribution (SD) and Global Trigger Processor (GTP) boards are located in the switch slots #A and #B respectively.

The TID implements two I<sup>2</sup>C buses, one for the CTP (or GTP) and the other for SD (or GTP/SD). Depending on the devices on the bus, it supports standard byte addressing, primary addressing and secondary addressing, one-byte, two-byte, or many bytes read/write. The TID is the I<sup>2</sup>C bus master; the CTP, SD and GTP are I<sup>2</sup>C slaves. The TID implements drive high then tri-state and pull high to improve the timing (signal rising time). To be simple, and compatible with the Silicon Lab's SI5326 PLL devices, the device numbers on I<sup>2</sup>C buses are b'1101xxx', that is each bus (CTP, GTP or SD) supports up to eight devices. For details about I<sup>2</sup>C bus, refer to the I<sup>2</sup>C bus specification [ii]. The FPGAs (on switch slot modules) are assigned b'1101000' I<sup>2</sup>C devices, which can have primary and secondary addresses.

## TID implementation (VME to I<sup>2</sup>C engine):

The TID boards are implemented as VME64x boards. They are expected to be in the JLAB VXS crate (minimum: standard 6U VME64 crate). The VME A24D32 is going to be used for VME to I<sup>2</sup>C engine. Each I<sup>2</sup>C device is treated as a block of VME addresses. The different addresses in the address block indicate the I<sup>2</sup>C START/STOP conditions and 1-byte/2-byte data transfer. Depending on the I<sup>2</sup>C slave devices, the 2-bytes data written may be treated as secondary addresses[i].

The VME D32 is divided into D(31:16) and D(15:0), two groups. The D(15:0) is the valid data for 2-byte transfer, and D(7:0) is the valid data for 1-byte transfer; the D(31:16) are DONOT CARE bits.

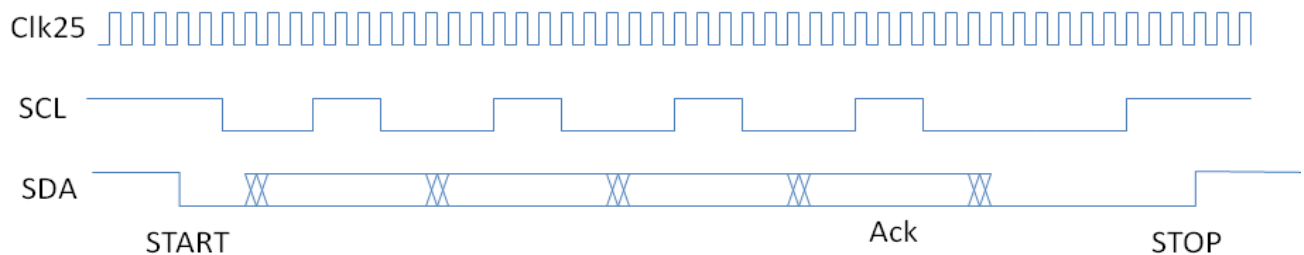
The table 1 shows the VME address allocation for the I<sup>2</sup>C engine.

A(23:19)	TID board address, set by the 5-bit switch or VME64x geographic address
A(18:16)	011: I <sup>2</sup> C bus for VXS switch slot#A, that is CTP (or GTP)
	100: I <sup>2</sup> C bus for VXS switch slot#B, that is SD (or GTP/SD)
	101: I <sup>2</sup> C bus for VME P2 Connector. (reserved)
A(15:13)	I <sup>2</sup> C devices. Together with higher 4-bits (1101), each I <sup>2</sup> C bus supports up to eight devices. The TID will combine the '1101' with A(15:13) to form the device ID. The device addresses are b'1101A <sub>15</sub> A <sub>14</sub> A <sub>13</sub> '. Right now, the higher 4-bits are defaulted to 0000, which is trying to be compatible with the older CTP/SD firmware, though the device address of 0000000 is for broadcast, not an individual I2C device address.

A(12)	Number of bytes in the data (D15:0). 0: one byte, D(7:0); 1: two bytes, D(15:0)
A(11)	I <sup>2</sup> C STOP condition. 0: no STOP; 1: STOP, that is the end of the I <sup>2</sup> C cycle
A(10)	I <sup>2</sup> C START condition. 0: no START; 1: START, that is a new I <sup>2</sup> C cycle
A(9:2)	I2C device byte address. For secondary addressing, this is the primary address. As limited by the available bits in A24 address space, the byte address is limited to 8-bits only. For larger I <sup>2</sup> C memory space, the secondary addressing is used, which is 16-bit secondary address and 16-bit data registers. If secondary address is used, the total I <sup>2</sup> C addressable memory is: $2^7(\text{primary address}) * 2^{16}(\text{secondary address}) * 2(\text{bytes/address})$ , which is about 16MByte.

Several examples are given later to show the VME Controller programming. In VME read/write, the actual register may be a mirror register on the TI or the SD/CTP/GTP for prompt respond to the VME cycles.

The I<sup>2</sup>C engine uses the FPGA configure clock (25 MHz) as the base clock. For easy control, the I<sup>2</sup>C clock (SCL) stays high normally. The SCL has a duty cycle of 37.5/62.5%. The SCL stays high for three configure clock cycles (120ns), and stays low for five configure clock cycles (200ns). The middle 40ns SCL low period supplies a cushion for the I<sup>2</sup>C data (SDA). The following figure shows the timing of the I<sup>2</sup>C bus:



The reduced functions (enough for our application) of the VME to I2C engine has been implemented in the FPGA, and simulated in ISE10.1 environment. The implemented engine can do one-byte read or two-byte read on a single VME read, one-byte write or two-byte write on a single VME write. The multi-byte (more than two bytes) operation has not been implemented yet. Because the two bytes read/write takes less than 20 us, we may not need to implement the multi-bytes read/write logic. The secondary addressing can be supported by the simple 2-byte operations in the FPGA. Here is one solution for the secondary addressing (affecting the SD/CTP/GTP design):

VME write 2-bytes to 1aaaaa: this will set the secondary address for primary address xaaaaa;

VME read 2-bytes from 1aaaaa: this will read the 16-bit secondary address back;

VME write to 0aaaaa: this will update the data register at primary address xaaaaa and secondary address (the secondary address is set by the previous VME write). After operation, the SD/CTP/GTP will automatically increase the secondary address by 1.

VME read from 0aaaaa: This will read the data register at primary address xaaaaa and secondary address (the secondary address is set by the previous VME write). After operation, the SD/CTP/GTP will automatically increase the secondary address by 1.

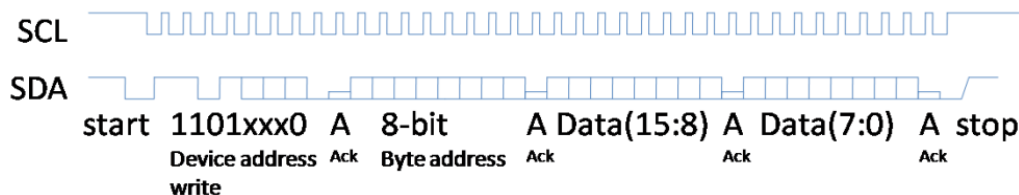
## CTP, SD and GTP implementation:

Right now, there are three I<sup>2</sup>C devices on CTP, which are all 2-byte registers. There are three I<sup>2</sup>C PLLs on the SD, which are 1-byte, 2-byte, many-byte registers, and another I<sup>2</sup>C slave similar to the CTP. 2-byte access is not sufficient for PLL control, as “Registers not listed (in the datasheet), such as register 64, should never be written to”. This means that the single-byte write has to be supported. Many bytes read/write can use the I<sup>2</sup>C bus more efficiently. As the two bytes read/write uses only 20 us, there is no need to do many-bytes read/write. The firmware and software will be simpler, and less prone to mistakes.

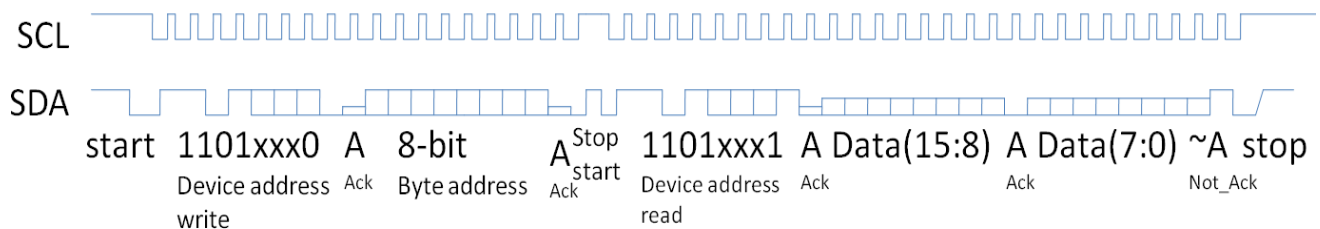
The secondary address was treated by the FPGAs on CTP, SD or GTP to expand the memory (register) space. For the FPGA (device CTP, device SD or device GTP), If the 8-bit I<sup>2</sup>C address is 1xxx,xxxx, the two bytes of data D(15:0) will be treated as secondary register address. This gives 2<sup>23</sup> secondary addresses, which correspond to 2<sup>24</sup> bytes (16 MB) of data memory. For PLLs, it uses the full 8-bit for primary address, and there is no secondary address. This may cause some confusion in the software, but should not be difficult. The SD/CTP/GTP may want to dedicate an I<sup>2</sup>C device specifically for secondary address support.

The VME software is simple and efficient. The complexity is hidden in the VME to I<sup>2</sup>C engine, which is part of the TID firmware design.

The following figure shows the process for two bytes writing to an I<sup>2</sup>C device:



The following figure shows the process for two bytes reading from an I<sup>2</sup>C device:



## Examples of VME software implementation:

### 1. 1-byte PLL register write:

For 1-byte register write, one VME write cycle is used:

AM(5:0) = 0x39 or 0x3A: A24D32;

A(23:0) = bbbb,bxxx,yyy0,11zz,zzzz,zz00: bbbb, TI module address; xxx, Switch #A or Switch #B; yyy, I<sup>2</sup>C device number; zzzzzzzz, byte address.

D(31:8): do not care;

D(7:0): 1-byte of data to be written to I<sup>2</sup>C device

This is implemented in the FPGA design, and functionally simulated in ISE10.1 Xilinx design.

## 2. 2-byte PLL register write or FPGA primary address register write:

For 2-byte register write, one VME write cycle is used:

AM(5:0) = 0x39 or 0x3A: A24D32;

A(23:0) = bbbb,bxxx,yyy1,11zz,zzzz,zz00: bbbbb, TI module address; xxx, Switch #A or Switch #B; yyy, I<sup>2</sup>C device number; zzzzzzz, byte address.

D(31:16): do not care;

D(15:0): 2-byte of data to be written to I<sup>2</sup>C device. The D(7:0) will be shifted before D(15:8), although the shifts are MSB first. The bit shift order is: D7, D6, ..., D1, D0, D15, D14, ..., D9, D8.

This is implemented in the FPGA design, and functionally simulated in ISE10.1 Xilinx design.

## 3. PLL register read, or primary address register read:

Each PLL register read needs only one VME read cycle. One-byte or 2-byte reading needs only one VME read cycle.

AM(5:0) = 0x39 or 0x3A: A24D32;

A(23:0) = bbbb,bxxx,yyyw,11zz,zzzz,zz00: bbbbb, TI module address; xxx, Switch #A or Switch #B; yyy, I<sup>2</sup>C device number; zzzzzzz, byte address; w=0 for 1-byte, z=1 for 2-byte.

The 1-byte and 2-bytes register reads are implemented in the FPGA design, and functionally simulated in ISE10.1 environment.

## 4. 2-byte secondary address register write

For the primary registers with secondary address registers, there is no data register corresponding to the primary address. The data corresponding to the primary register is the secondary address register.

For 2-byte secondary address register write, two VME write cycles are used. As the secondary address only applies to the FPGAs, the secondary address capable registers can be defined as a separate I<sup>2</sup>C device address, for example: b'1101111'.

First VME write cycle: set the START and STOP conditions, set the primary address in the VME address space, and set the secondary address in VME data bits(15:0):

AM(5:0) = 0x39 or 0x3A: A24D32;

A(23:0) = bbbb,bxxx,yyy1,111z,zzzz,zz00: bbbbb, TI module address; xxx, Switch #A or Switch #B; yyy, device address, 1zzzzzz, primary address, which support secondary addressing. The highest bit is 1.

D(15:0): the secondary address;

Second VME write cycle: set the START and STOP conditions, set the primary address in the VME address space, and set the data in VME data bits(15:0). The secondary address will be the data(15:0) set in the first VME write cycle. After the VME write, the I<sup>2</sup>C device (FPGA) will automatically increase the secondary address by 1, so the next write will be in the next secondary address automatically:

AM(5:0) = 0x39 or 0x3A: A24D32;

A(23:0) = bbbb,bxxx,yyy1,110z,zzzz,zz00: bbbbb, TI module address; xxx, Switch #A or Switch #B; yyy, device address, 0zzzzzz, primary address, the highest bit is 0. (if the address is 1zzzzzz, the data will be secondary address, if the address is 0zzzzzz, the data will be the secondary address register data)

D(15:0): 2-bytes worth of data for the secondary register;

## 5. 2-byte secondary address register read

This is similar to the secondary address write. It needs one VME write cycles and one VME read cycles. The VME write cycle will write the secondary address to the primary address register, which is the same as the first VME write in the secondary data write. The VME read cycle will be similar to the standard two-bytes FPGA data read.

First, VME write: set the START/STOP condition, write the secondary address to the primary address. The data byte indicator (1-byte or 2-byte) is set to 1.

AM(5:0) = 0x39 or 0x3A: A24D32;

A(23:0) = bbbb,bxxx,yyy1,111z,zzzz,zz00: bbbbb, TI module address; xxx, Switch #A or Switch #B; yyy, device address, 1zzzzzz, primary address, which support secondary addressing. The highest bit is 1.

D(15:0): the secondary address;

Second, VME read: read the 2-byte of data from the previously set secondary address. After the VME read, the I<sup>2</sup>C device (FPGA) will automatically increase the secondary address by 1, so the next read will be in the next secondary address automatically:

AM(5:0) = 0x39 or 0x3A: A24D32;

A(23:0) = bbbb,bxxx,yyy1,110z,zzzz,zz00: bbbbb, TI module address; xxx, Switch #A or Switch #B; yyy: device address capable of secondary addressing; 0zzzzzz, primary address.

---

<sup>i</sup> Hai Dong, TI, CTP, SD I2C implementation, Apr. 8, 2008;

<sup>ii</sup> Philips Semiconductors, The I<sup>2</sup>C-bus Specification, Version 2.1, January, 2000;