



Fermi National Accelerator Laboratory

**D-Zero Detector Calorimeter Electronics
Run II b Upgrade Project**

Test Waveform Generator System Prototype

Date: June 9th, 2003
Revision Date: May 12th, 2004

N. George Wilcer, S. Marco Rapisarda

Document # ESE-D0-030609

TABLE OF CONTENTS

1	INTRODUCTION	3
2	SYSTEM	4
	Figure 2.1, System block diagram.....	4
3	XILINX X2S150 FPGA TEST CARD	5
	Figure 3.1, Avnet's Xilinx Spartan II Evaluation kit card.....	5
3.1	MEMORY MAPPING	7
	Figure 3.2, Address format	7
	Table 3.1, Memory map.....	7
	Table 3.2, Board Block memory map	8
3.1.1	<i>Board Control/Status registers.....</i>	<i>9</i>
	Table 3.3, Board Control/Status registers	9
	Figure 3.3, Board Control/Status register 0.....	10
	Table 3.4, Board Control/Status register 0.....	10
	Figure 3.4, Board Control/Status register 3.....	11
	Figure 3.5, Board Control/Status register 4.....	11
	Figure 3.6, Board Control/Status register 6.....	12
	Figure 3.7, Board Control/Status register 7.....	12
3.1.2	<i>Channel Control/Status registers.....</i>	<i>13</i>
	Table 3.5, Channel Control/Status registers.....	13
	Figure 3.8, Channel Control/Status register 0	14
	Table 3.6, Channel Control/Status register 0.....	15
	Figure 3.9, Channel Control/Status register 1 and 2	15
	Figure 3.10, Channel Control/Status register 3 and 4	15
	Figure 3.11, Channel Control/Status register 5	15
3.2	TRIGGER CONTROL.....	16
3.2.1	<i>Examples of trigger settings.....</i>	<i>19</i>
	Table 3.7, settings example for channel free-run	19
	Table 3.8, settings example for channel timed trigger	20
	Table 3.9, settings example for multi-channel timed trigger.....	21
	Table 3.10, settings example for multi-channel manual trigger	22
	Table 3.11, settings example for multi-channel external trigger	23
	Table 3.12, settings example for multi-channel SCL beam crossing trigger.....	24
	Table 3.13, settings example for multi-channel SCL turn trigger	25
	Table 3.14, settings example for multi-channel SCL beam crossing and turn trigger.....	26
3.3	SERIAL PORT INTERFACE.....	27
3.3.1	<i>Write command.....</i>	<i>28</i>
	Table 3.15, Protocol structure of the write command	28
3.3.2	<i>Read command.....</i>	<i>29</i>
	Table 3.16, Protocol structure of the read command.....	29
	Table 3.17, Command error word.....	30
3.4	BOARD SWITCHES/BUTTONS MAPPING	31
	Table 3.18, Board switches/buttons mapping.....	31
3.4.1	<i>Logic Analyzer Connectors.....</i>	<i>32</i>
	Table 3.19, HP Logic Analyzer connector J4 (suggested analyzer pods: 1 and 2).....	32
	Table 3.20, HP Logic Analyzer connector J5 (suggested analyzer pods: 3 and 4).....	33
	Table 3.21, HP Logic Analyzer connector J6 (suggested analyzer pods: 5 and 6).....	34
3.4.2	<i>Header Connectors.....</i>	<i>35</i>
	Table 3.22, Header connector JP5.....	36
	Table 3.23, Header connector JP6.....	37
	Table 3.24, Header connector P2-A (AvBus)	38
	Table 3.25, Header connector P2-B (AvBus).....	39
3.5	JTAG INTERFACE.....	40
	Figure 3.13, JTAG Connectors pinout	40
	Table 3.26, JTAG Chain Devices	40

4	AD9765 TEST CARD	41
	Figure 4.1, AD9765 evaluation board.....	41
	Table 4.1, Header connector P1 (DAC Channel 1).....	42
	Table 4.2, Header connector P2 (DAC Channel 2).....	43
5	AD8138 TEST CARD	44
	Figure 5.1, AD8138 evaluation board.....	44
6	THS4141 TEST CARD	45
	Figure 6.1, THS4141 evaluation board.....	45
6.1	INTERFACE WITH ADF SYSTEM.....	46
	Figure 6.2, Electrical diagram of the interface with ADF system.....	46
7	CUSTOM POWER SUPPLY CARD	47
8	SERIAL COMMAND LINK RECEIVER (SCLR) CARD	48
	Figure 8.1, SCL Receiver card.....	48
	Table 8.1, Header connector J1.....	49
	Table 8.2, Header connector J2.....	50
9	SCL RECEIVER TO XILINX TEST CARD PASSIVE ADAPTER	51
	Figure 9.1, Passive adapter for SCL Receiver and Xilinx Test card.....	51
	Figure 9.2, SCL Adapter, Xilinx Test card and SCL passive adapter assembly.....	51
	Table 9.1, SCL Receiver to Xilinx Test card adapter connections map.....	53
10	APPENDIX A - DESCRIPTION OF MS EXCEL SPREADSHEET INTERFACE	54
11	APPENDIX B - JUNE 11TH 2003 MEASUREMENTS	55
	Figure 11.1, block diagram of the June 11 th 2003 Test System.....	55
	Figure 11.2, Measure 1.....	56
	Figure 11.3, Measure 2.....	57
	Figure 11.4, Measure 3.....	58
12	GLOSSARY	59
	Figure 12.1, Switching standards.....	62
	Figure 12.2, Comparison of switching standards.....	62
13	REFERENCES	64

1 Introduction

This document describes the Test Waveform Generator (TWG) prototype system. The Test Waveform Generator System is part of the electronics of the D-Zero detector Calorimeter Trigger Test System at Fermilab. More information on the experiments performed at Fermilab is available on the laboratory web page:

<http://www.fnal.gov/>

More information on the D0 Detector is available on:

<http://www-d0.fnal.gov/>

The designers welcome suggestions and corrections [13], which can be addressed directly to the engineer responsible of the project. Contact information is available on the Electronics System Engineering (ESE) web page:

<http://www-ese.fnal.gov/>

More information and documentation on the Test Waveform Generator Project are available on:

http://www-ese.fnal.gov/D0Cal_TWG/

Contributing people: Gustavo Cancelo, Stefano Rapisarda, Neal Wilcer

2 System

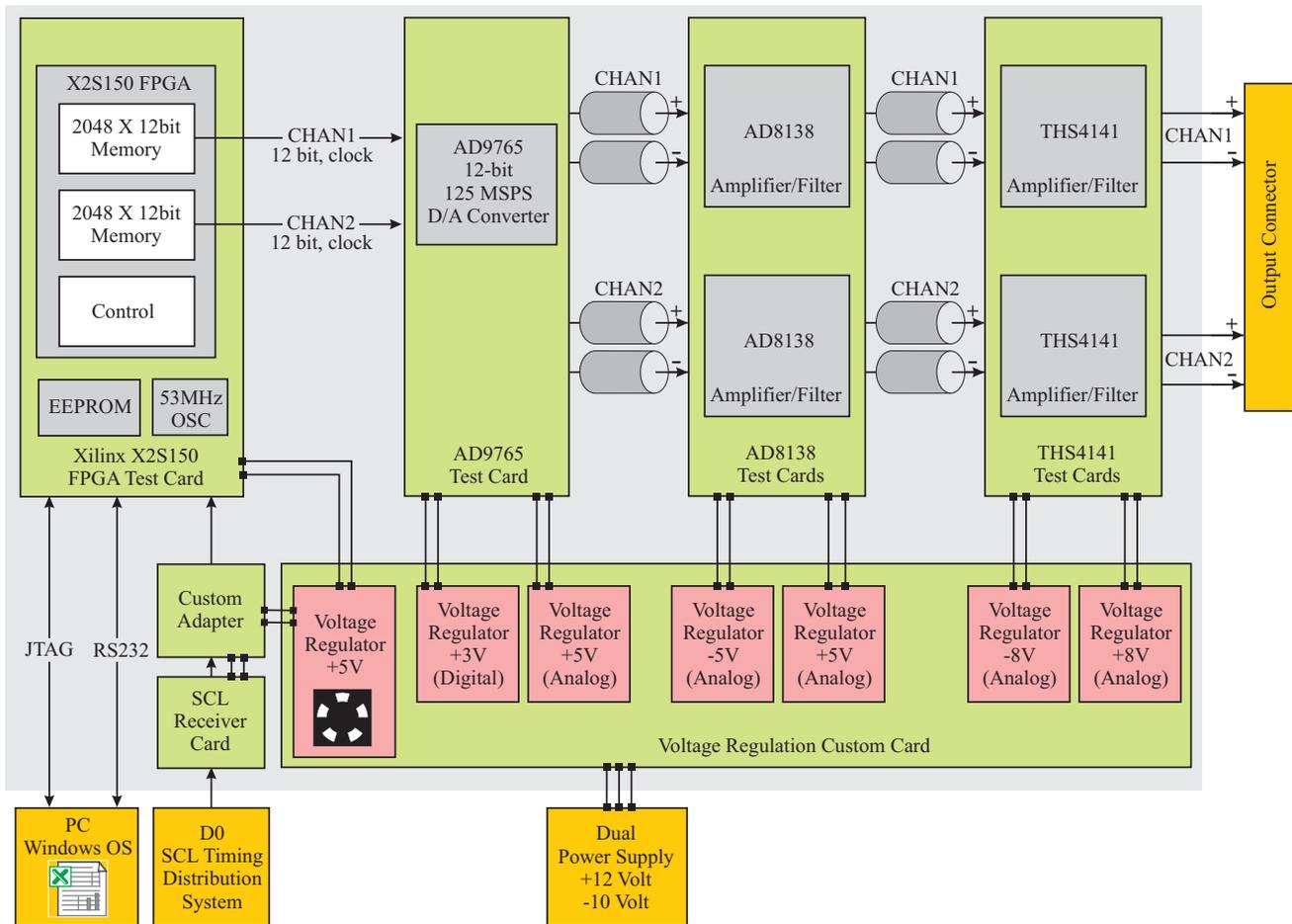


Figure 2.1, System block diagram

The *Test Waveform Generator* (TWG) prototype system is a two-channel arbitrary waveform generator that has been assembled using commercially available test cards.

The system needs an external dual power supply. Internal power distribution and regulation is done with a custom card.

An Avnet [12] Xilinx [11] Spartan II evaluation card is used to interface the system with a PC over a serial port (RS232). A Graphical User Interface based on Microsoft Excel is used to communicate with the system. The system can also be interfaced with most platforms that have a serial port. For this purpose a description of the protocol used is provided in paragraph 3.2 (page 16).

The PC is also used to reprogram the Xilinx FPGA's configuration EEPROM with the use of a Xilinx JTAG cable.

The FPGA on the Xilinx test card provides the memory and the controls used to generate the data streams at the input of a two channel Digital to Analog Converter board, an Analog Devices [7] AD9765 test card.

Each of the two output channels of the AD9765 card is fed to an Analog Devices AD8138 card, used for the first stage of amplification and filtering. The outputs of the AD8138 cards drive two Texas

Instruments THS4141 cards. These cards implement the second stage of amplification/filtering and drive the signals on the output connector.
 A Serial Command Link (SCL) receiver card is used to receive timing information from the D0 SCL Timing Distribution System. A custom passive adapter interfaces the SCL receiver with the Xilinx test card.

3 Xilinx X2S150 FPGA Test Card

The card used is the one provided in the Avnet [12] Spartan II Evaluation Kit (Avnet part number: ADS-XLX-SP2-EVL).

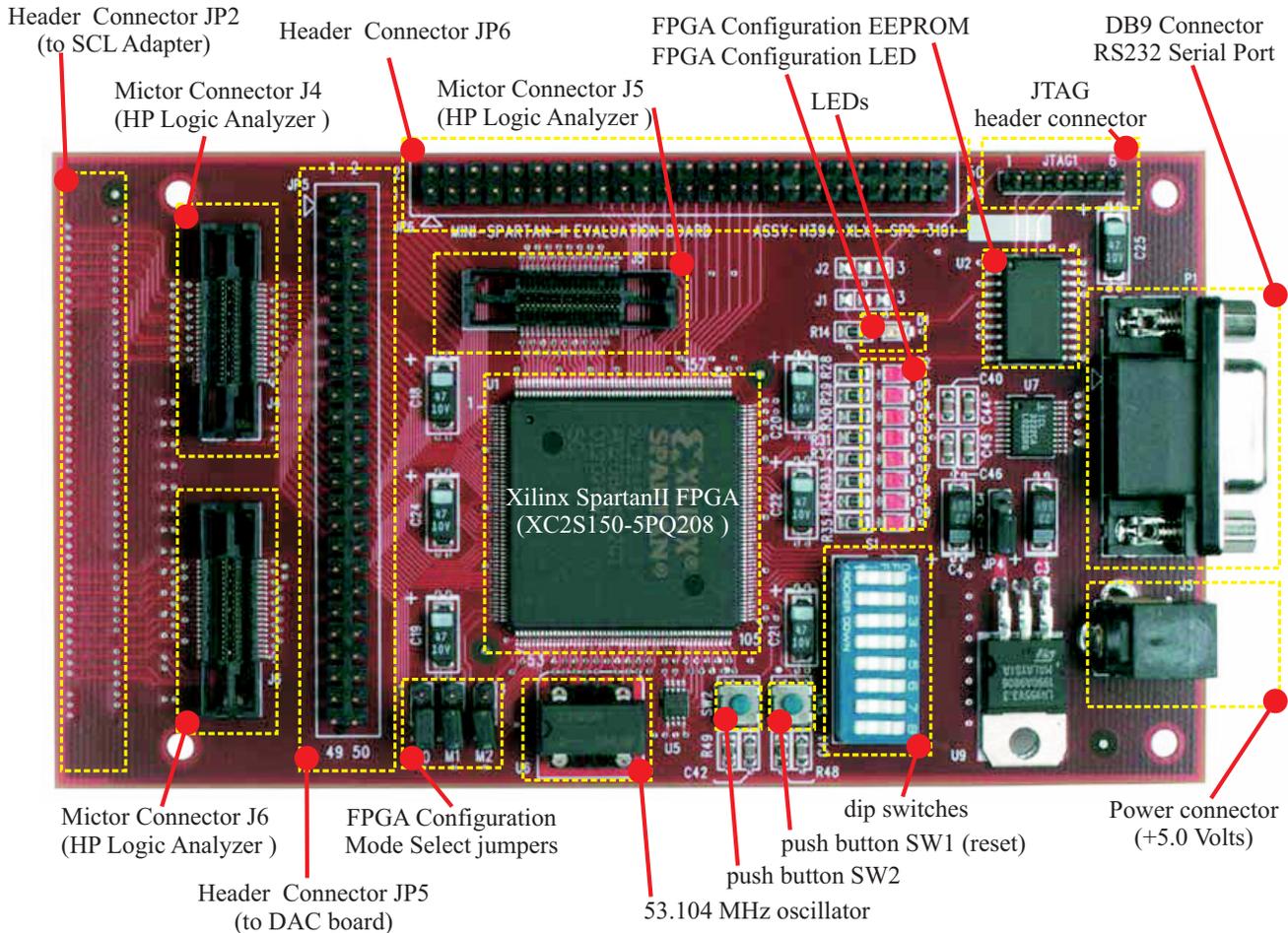


Figure 3.1, Avnet’s Xilinx Spartan II Evaluation kit card.

The Spartan II evaluation kit is supplied with VHDL source code. The code has been rewritten in order to meet the Test Waveform Generator prototype needs.

The primary purpose of the Xilinx Test Card is to provide the interface to a personal computer and the memory to store the arbitrary waveform data for the two-channel prototype system.

The amount of memory required for each channel depends on the desired amplitude resolution (DAC bits), the frequency of the DAC (f_{DAC}) and the length of the signal (t_{signal}) to be emulated (or the number of beam crossings). The beam crossing frequency is $f_{BXing} = \frac{f_{Tevatron}}{7}$ where the Tevatron frequency is $f_{Tevatron} = 53.104MHz$.

For the TWG prototype system the amplitude resolution has been established at 12 bits, so the number M of 12-bit memory words needed for a single channel to emulate N beam crossings is:

$$M_{Chan} = f_{DAC} \cdot t_{signal} = \left(\frac{f_{DAC}}{f_{Tevatron}} \right) \cdot 7 \cdot N = \left(\frac{f_{DAC}}{53.104MHz} \right) \cdot 7 \cdot N.$$

The number of beam crossing that can be emulated is:

$$N = \frac{M_{Chan}}{7} \cdot \left(\frac{53.104Mhz}{f_{DAC}} \right).$$

A memory of 2048 12-bit words has been allocated for each channel on the Xilinx FPGA. The Xilinx card runs on a 53.104 MHz clock and the same clock is used to drive the Analog Devices AD9765 DAC card. Using these values the number of beam crossings the TWG prototype system can emulate is:

$$N_{DAC@53.104MHz} = \frac{M_{Chan}}{7} \cong 292.$$

3.1 Memory Mapping

The memory space inside the Xilinx FPGA is implemented in locations two bytes wide. Each location is identified by a two-byte address (0x0000 to 0xFFFF). The memory is accessible in read/write through the serial port interface specified in paragraph 3.2 (page 16).

For convenience the memory can be considered as partitioned in blocks, with the block number specified by the most significant 4 bits of the 16-bit address word. Currently only three of the 16 possible blocks are used. Block 0 is used for the board control/status registers. Block 1 and block 2 are assigned respectively to channel 1 and channel 2 to store the data used to drive the Digital to Analog Converter (AD9765) card.

Block address	Row address		
bits 15..12	bits 11..8	bits 7..4	bits 3..0

Figure 3.2, Address format

Both channel 1 and 2 use the rows from 0x000 to 0x7FF to map the 12 bit wide “block RAM” locations (2048 words deep) used to store the information to drive the Digital to Analog Converters.

Board Block 0x0				Channel 1 Block 0x1				Channel 2 Block 0x2				Row	
MSB		LSB		MSB		LSB		MSB		LSB			
15..12	11..8	7..4	3..0	15..12	11..8	7..4	3..0	15..12	11..8	7..4	3..0		
Board Memory block See Table 3.2													0x000
													0x001
													•
													•
													•
													0x7FE
													0x7FF
													•
													•
													•
													•
													•
													0xFFFF

Table 3.1, Memory map

Board Block 0x0				Row
MSB		LSB		
15..12	11..8	7..4	3..0	
Board control/status register 0				0x000
Board control/status register 1				0x001
				•
				•
				•
Board control/status register 15				0x00F
Channel 1 control/status register 0				0x010
Channel 1 control/status register 1				0x011
				•
				•
				•
Channel 1 control/status register 15				0x01F
Channel 2 control/status register 0				0x020
Channel 2 control/status register 1				0x021
				•
				•
				•
Channel 2 control/status register 15				0x02F
				•
				•
				•
				•
				•
				0xFFFF

Table 3.2, Board Block memory map

3.1.1 Board Control/Status registers

Register	Address	Notes
Control/Status 0	0x0000	Trigger control and enable
Control/Status 1	0x0001	--
Control/Status 2	0x0002	--
Control/Status 3	0x0003	Multi-channel timed trigger. Clock ticks max count double-word (Least Significant Word)
Control/Status 4	0x0004	Multi-channel timed trigger. Clock ticks max count double-word (Most Significant Word)
Control/Status 5	0x0005	--
Control/Status 6	0x0006	SCL Current Turn Multi-channel trigger word
Control/Status 7	0x0007	SCL Current beam Crossing Multi-channel trigger half-words
Control/Status 8	0x0008	--
Control/Status 9	0x0009	--
Control/Status 10	0x000A	--
Control/Status 11	0x000B	--
Control/Status 12	0x000C	--
Control/Status 13	0x000D	--
Control/Status 14	0x000E	--
Control/Status 15	0x000F	--

Table 3.3, Board Control/Status registers

3.1.1.1 Board Control/Status register 0 (trigger control and enable)

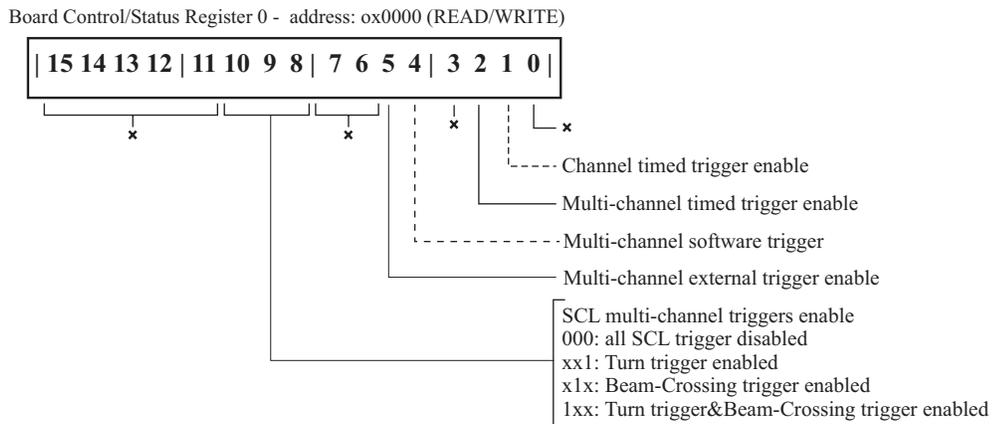


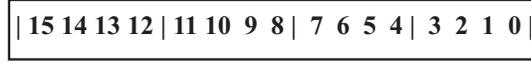
Figure 3.3, Board Control/Status register 0

Bit#	Name (Type)	Interpretation
15..11	--	Not used
10..8	SCL multi-channel trigger enable	Board level enable for the SCL triggers. The SCL triggers must also be individually enabled for each channel. There are three available triggers which can be enabled in any combination: 000: all SCL multi-channel trigger are disabled xx1: bit 8 enables the turn trigger. The trigger is generated when the “SCL current turn” matches the value specified in board control/status register 6. x1x: bit 9 enables the beam crossing trigger. The trigger is generated when the “SCL current beam crossing” matches one of the values specified in board control/status register 7. 1xx: bit 10 enables the Turn & Beam crossing trigger. The trigger is generated when BOTH the “SCL current turn” and the “SCL current beam crossing match the values specified in board control/status registers 6 and 7.
7..6	--	Not used
5	Multi-channel external trigger enable	Enable at board level the external trigger. The trigger must also be individually enabled for each channel. The trigger is generated by the push button SW2.
4	Multi-channel software trigger.	Generate a board level multi-channel trigger when the bit value transition from 0 to 1. The software trigger must also be individually enabled for each channel.
3	--	Not used
2	Multi-channel timed trigger enable	Enable at board level the multi-channel timed trigger. The trigger must also be individually enabled for each channel. A counter continuously loops between 0 and the 32 bit value specified in the <u>board</u> control/status registers 3 and 4. A trigger is generated at the beginning of each count (i.e. when the count value is zero).
1	Channel timed trigger enable	Enable at board level the channel timed trigger functionality. The trigger must also be individually enabled for each channel. The trigger is generated independently for each channel by a channel-dedicated counter that continuously loops between 0 and the 32 bit value specified in the <u>channel</u> control/status registers 3 and 4. A trigger is generated at the beginning of each count (i.e. when the count value is zero).
0	--	Not used

Table 3.4, Board Control/Status register 0.

3.1.1.2 Board Control/Status register 3 and 4 (multi-channel timed trigger)

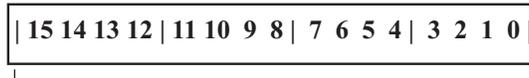
Board Control/Status Register 3 - address: 0x0003 (READ/WRITE)



Multi-channel timed trigger
Clock ticks count (least significant word)

Figure 3.4, Board Control/Status register 3

Board Control/Status Register 4 - address: 0x0004 (READ/WRITE)



Multi-channel timed trigger
Clock ticks count (most significant word)

Figure 3.5, Board Control/Status register 4

The Multi-channel timed trigger is used to trigger several channels simultaneously. The trigger logic consists in a counter using as maximum count value specified as double-word (32 bits) in the board status/control registers 3 (least significant word) and 4 (most significant word). The trigger is generated at the beginning of the count and has the effect of resetting both the channel memory address counter and the channel timed trigger counter (see Figure 3.12, page 18).

The trigger functionality is enabled at board level in the board control/status register 0. In order to operate it must also be enabled for the desired channels in their channel control/status register 0.

3.1.1.3 Board Control/Status register 6 and 7 (SCL triggers)

Board Control/Status Register 6 - address: 0x0006 (READ/WRITE)

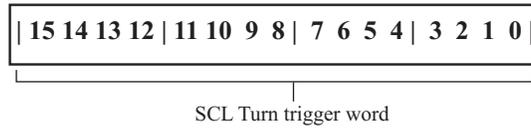


Figure 3.6, Board Control/Status register 6

Board register 6 defines the word (16 bits) used by the multi-channel SCL turn trigger.

Board Control/Status Register 7 - address: 0x0007 (READ/WRITE)

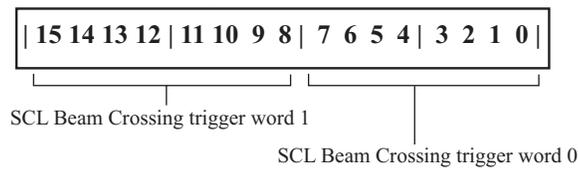


Figure 3.7, Board Control/Status register 7

Board register 7 defines two half-words (8 bits) used by the multi-channel SCL beam-crossing trigger.

The combination of board registers 6 and 7 defines the values used by the multi-channel SCL turn & beam-crossing trigger

3.1.2 Channel Control/Status registers

Register	Address	Notes
Control/Status 0	0x1000 (ch1) 0x2000 (ch2)	Channel trigger control and enable
Control/Status 1	0x1001 (ch1) 0x2001 (ch2)	Channel memory depth selection. Memory initial address (0x000 to 0x7FF)
Control/Status 2	0x1002 (ch1) 0x2002 (ch2)	Channel memory depth selection. Memory final address (0x000 to 0x7FF)
Control/Status 3	0x1003 (ch1) 0x2003 (ch2)	Channel timed trigger. Clock ticks max count (Least Significant Word)
Control/Status 4	0x1004 (ch1) 0x2004 (ch2)	Channel timed trigger. Clock ticks max count (Most Significant Word)
Control/Status 5	0x1005 (ch1) 0x2005 (ch2)	Multi-pulse counter max value. (0x0001 to 0xFFFF, default value is 0x0001)
Control/Status 6	0x1006 (ch1) 0x2006 (ch2)	
Control/Status 7	0x1007 (ch1) 0x2007 (ch2)	
Control/Status 8	0x1008 (ch1) 0x2008 (ch2)	
Control/Status 9	0x1009 (ch1) 0x2009 (ch2)	
Control/Status 10	0x100A (ch1) 0x200A (ch2)	
Control/Status 11	0x100B (ch1) 0x200B (ch2)	
Control/Status 12	0x100C (ch1) 0x200C (ch2)	
Control/Status 13	0x100D (ch1) 0x200D (ch2)	
Control/Status 14	0x100E (ch1) 0x200E (ch2)	
Control/Status 15	0x100F (ch1) 0x200F (ch2)	

Table 3.5, Channel Control/Status registers

3.1.2.1 Channel Control/Status register 0 (channel trigger control and enable)

Channel Control/Status Register 0 - address: 0x'n'000 (READ/WRITE)

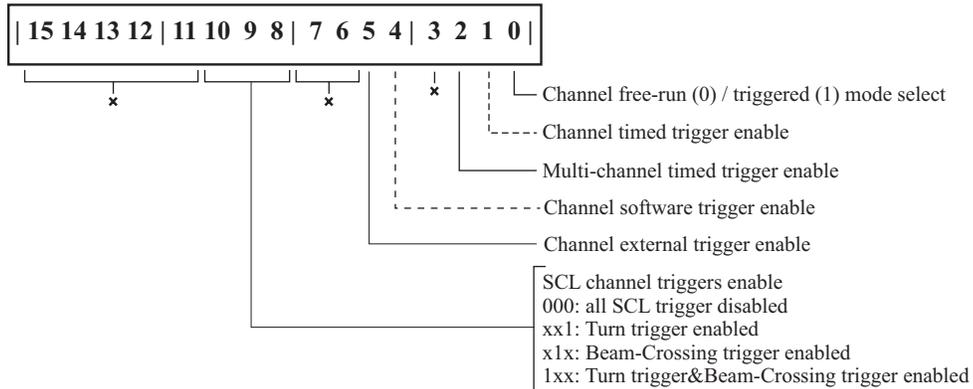


Figure 3.8, Channel Control/Status register 0

Bit#	Name (Type)	Interpretation
15..11	--	Not used
10..8	SCL multi-channel trigger enable	Channel level enable for the SCL triggers. The SCL triggers must also be enabled at board level. There are three available triggers which can be enabled in any combination: 000: all SCL multi-channel trigger are disabled xx1: bit 8 enables the turn trigger. The trigger is generated when the “SCL current turn” matches the value specified in board control/status register 6. x1x: bit 9 enables the beam crossing trigger. The trigger is generated when the “SCL current beam crossing” matches one of the values specified in board control/status register 7. 1xx: bit 10 enables the Turn & Beam crossing trigger. The trigger is generated when BOTH the “SCL current turn” and the “SCL current beam crossing match the value specified in board control/status registers 6 and 7.
7..6	--	Not used
5	Multi-channel external trigger enable	Enables at channel level the external trigger. The trigger must also be enabled at board level. The trigger is generated by the push button SW2.
4	Multi-channel software trigger.	Enables at channel level the multi-channel software trigger. The software trigger must also be enabled at board level.
3	--	Not used
2	Multi-channel timed trigger enable	Enable at channel level the multi-channel timed trigger. The trigger must also be enabled at board level. A counter continuously loops between 0 and the 32 bit value specified in board control/status registers 3 and 4. A trigger is generated at the beginning of each count (i.e. when the count value is zero).
1	Channel timed trigger enable	Enables at channel level the channel timed trigger functionality. The trigger must also be enabled for board level. The trigger is generated independently by a channel-dedicated counter. The counter continuously loops between 0 and the 32 bit value specified in the channel control/status registers 3 and 4. A trigger is generated at the beginning of each count (i.e. when the count value is zero).
0	Channel mode of operation selection	When set to 0 (default) the channel operates in “free run” mode. The channel address counter continuously loops between the memory limits specified in channel registers 1 and 2. When set to 1 enables the triggered mode of operation. This will allow the channel to be triggered by any trigger that is both enabled at channel and board level. When the channel is triggered the channel address counter scroll between the memory limits and stops. The number of time the channel address counter must scroll between the memory limits is specified in the channel control/status register 5

Table 3.6, Channel Control/Status register 0.

3.1.2.2 Channel Control/Status register 1 and 2 (memory depth selection)

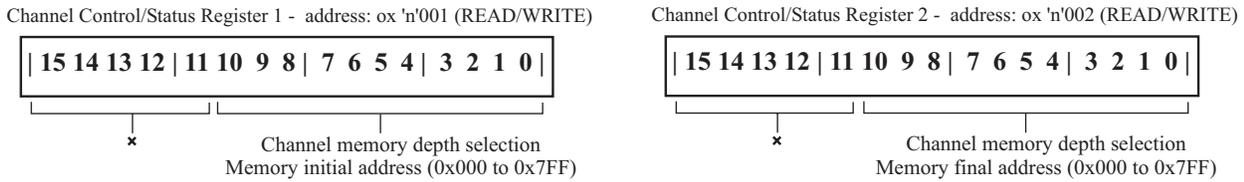


Figure 3.9, Channel Control/Status register 1 and 2

The two registers are used to specify the two addresses to be used by the channel memory address counter. The memory address counters operate in two modes (“free run” and “triggered”) selectable by the channel control/status register0.

3.1.2.3 Channel Control/Status register 3 and 4 (channel timed trigger)

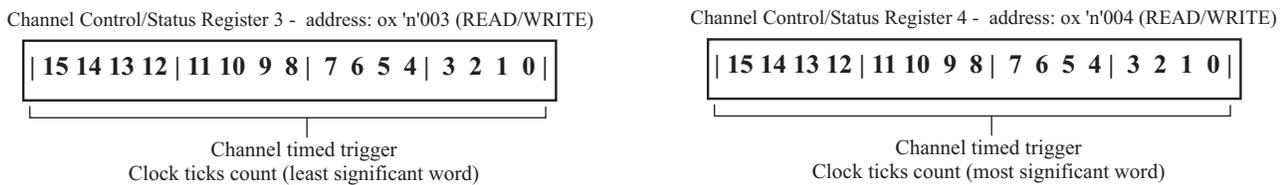


Figure 3.10, Channel Control/Status register 3 and 4

The two registers are used to specify the maximum count value for the channel timed trigger counter.

3.1.2.4 Channel Control/Status register 5 (multi-pulse control)

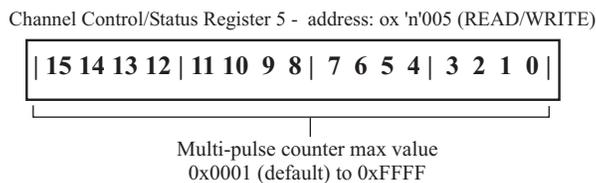


Figure 3.11, Channel Control/Status register 5

The register is used to specify how many times the channel memory address counter should loop before stopping. This function is active when the channel memory address counter is operating in “triggered mode”. The default content of the register is 1.

3.2 Trigger Control

Each TWG output channel has an independent memory address counter.

The counter can operate in two different modes, the “free-run” (default) and the “triggered”. The mode of operation is selected in channel control/status register 0.

In the “free-run” mode the counter continuously loops through the memory between two addresses. These addresses are specified in channel registers 1 (initial address, default is 0) and 2 (final address, default is 1024 (0x03FF)).

When triggered mode is selected the memory address counter scrolls between the initial and the final addresses and hold the last address value indefinitely. Independently of the counter status (counting or holding) the counting is restarted each time a trigger signal is received.

A “multi-pulse” function can force the memory counter to loop several times after a trigger. Channel register 5 is used to specify the number of times the memory address counter should scroll between the initial and the final addresses after a trigger is received, the default value is 1.

For a channel to be triggered ALL the following conditions must be verified:

- 1) One or more trigger modes is enabled at board level (board status/control register 0).
- 2) Triggered mode is selected for the channel to be triggered (bit 0 of the channel control/status register 0).
- 3) One or more trigger mode is selected for the channel to be triggered (channel control/status register 0).

The board level trigger enables control triggers generation for all the channels.

The channel level trigger enables control triggers for each individual channel. For a particular type of trigger to be generated it should be enabled at BOTH board and channel level.

The following is a description of the available triggering options.

No trigger (“free run” mode)

In the default “free-run” mode, channels run independently from the others using their own memory address counter. Synchronization between channels is not enforced. At startup/reset all the counters start at the same time, but because during a channel memory access the address counter of the accessed channel is reset, there will be a difference in the address count values of different channels. The use of a multi-channel trigger allows having a synchronized waveform between channels.

Channel timed trigger

Each channel has a dedicated timed trigger counter. The counter counts between the value zero and the one specified in channel register 3 (least significant word) and 4 (most significant word). A “timed trigger” signal is generated at the beginning of each count.

Multi-channel timed trigger

Is used to simultaneously trigger multiple channels. The function is based on a board level counter generating a trigger distributed to all the channels. The counter counts between the value zero

and the one specified in board register 3 (least significant word) and 4 (most significant word). A “multi-channel timed trigger” signal is generated at the beginning of each count.

SCL multi-channel turn trigger

The Multi-channel SCL turn trigger is used to trigger several channels simultaneously when the “SCL current turn” value matches the word (16 bits) specified in the board status/control registers 6.

SCL multi-channel beam-crossing trigger

The Multi-channel SCL beam-crossing trigger is used to trigger several channels simultaneously when the “SCL current beam crossing” value matches one of the two half-words (8 bits) specified in the board status/control register 7.

SCL multi-channel turn & beam-crossing trigger

The multi-channel SCL turn & beam-crossing trigger is used to trigger several channels simultaneously when BOTH the “SCL current turn” & “SCL current beam crossing” values match the values specified in the board status/control registers 6 and 7.

Use of more than one type of trigger

Each trigger is generated independently and the simultaneous use of different types of triggers is supported by the system.

For example the user may have the external trigger always enabled when operating in the triggered mode. This will allow the user to easily generate a trigger using push button SW2 even if another type of trigger is enabled.

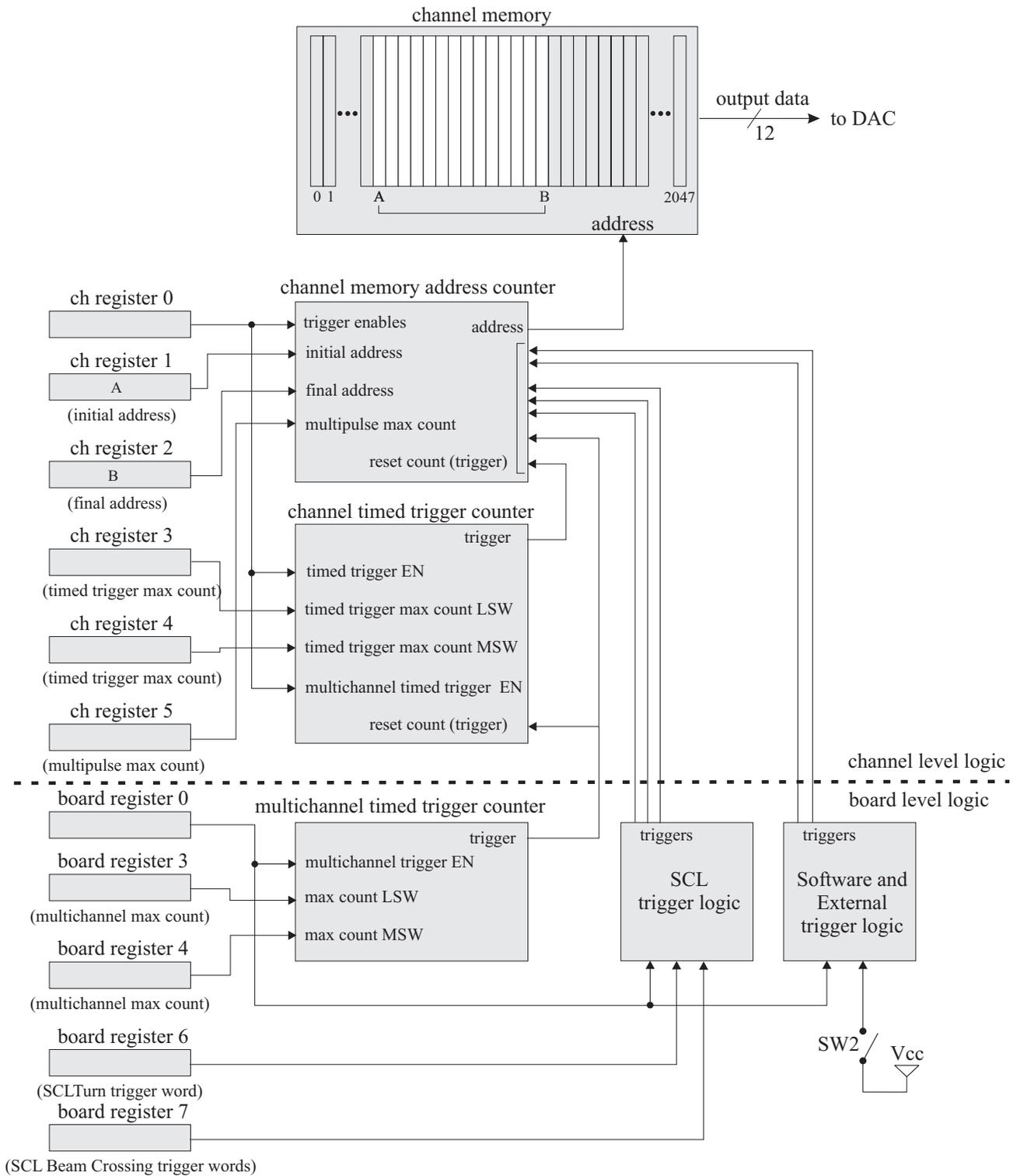


Figure 3.12, Channel trigger control

3.2.1 Examples of trigger settings

Channel free run

When a channel operate in “free-run” mode (channel register 0) the board registers have no effect on its operations. Channel registers 1 and 2 are used to set the two addresses representing the boundaries of the channel memory to be used. A channel in free-run mode runs independently from other channels and the channel’s memory address continuously loops between its memory boundaries (ch. registers 1 and 2). No synchronization is enforced between channels.

In the setting example of Table 3.7 the channel memory continuously loops between the memory boundaries set address 0x0000 (ch. register 1) and 0x07FF (ch. register 2).

Board registers			Channel registers		
#	Address	Content	#	Address	Content
0	0x 0000	N.A.	0	0x n000	bxxxx xxxx xxxx xxx0
1	0x 0001	N.A.	1	0x n001	b0000 0000 0000 0000
2	0x 0002	bxxxx xxxx xxxx xxxx	2	0x n002	b0000 0111 1111 1111
3	0x 0003	bxxxx xxxx xxxx xxxx	3	0x n003	bxxxx xxxx xxxx xxxx
4	0x 0004	N.A.	4	0x n004	bxxxx xxxx xxxx xxxx
5	0x 0005	bxxxx xxxx xxxx xxxx	5	0x n005	bxxxx xxxx xxxx xxxx
6	0x 0006	bxxxx xxxx xxxx xxxx	6	0x n006	N.A.
7	0x 0007	bxxxx xxxx xxxx xxxx	7	0x n007	N.A.

Table 3.7, settings example for channel free-run

Channel timed trigger

When channels are in the “channel timed trigger” mode (channel register 0) the board registers from 1 through 7 have no effect on channel operations. Channel registers 1 and 2 are used to set the two addresses representing the boundaries of the channel’s memory to be used. A channel dedicated counter loops continuously between zero and a maximum value (channel registers 3 and 4) and generate a trigger signal at the beginning of each count.

A channel in “channel timed trigger” mode runs independently from other channels, no synchronization is enforced between channels.

In the setting example of Table 3.8, each time a trigger is received, the channel memory address loops five times (ch. register 5) between memory address 0x0200 (ch. register 1) and 0x03FF (ch. register 2). A trigger is generated each time the counter restarts. The counter maximum value (ch. registers 3 and 4) is set to 1,000,000 (0xF4240), which with a 53.104MHz clock correspond to about 53 triggers each second.

Board registers			Channel registers		
#	Address	Content	#	Address	Content
0	0x 0000	b0000 0000 0000 0010	0	0x n000	bxxxx xxxx xxxx xx11
1	0x 0001	N.A.	1	0x n001	b0000 0010 0000 0000
2	0x 0002	N.A.	2	0x n002	b0000 0011 1111 1111
3	0x 0003	bxxxx xxxx xxxx xxxx	3	0x n003	b0100 0010 0100 0000
4	0x 0004	bxxxx xxxx xxxx xxxx	4	0x n004	b0000 0000 0000 1111
5	0x 0005	N.A.	5	0x n005	b0000 0000 0000 0101
6	0x 0006	bxxxx xxxx xxxx xxxx	6	0x n006	N.A.
7	0x 0007	bxxxx xxxx xxxx xxxx	7	0x n007	N.A.

Table 3.8, settings example for channel timed trigger

Multi-channel timed trigger

When channel are in the “multi-channel timed trigger” mode (channel register 0) the trigger is generated at board level. A counter loops continuously between zero and a maximum value (board registers 3 and 4) and generate a trigger signal at the beginning of each count.

A channel in “multi-channel timed trigger” mode runs synchronously to other channels in the same mode due to the trigger common source. Each channel memory boundaries are specified in channel registers 1 and 2.

In the setting example of Table 3.9 each time a trigger is received both channel A and B memory addresses scroll between the established boundaries (ch. registers 1 and 2).

Channel A memory address loops four times (ch. register 5) between memory address 0x0000 (ch. register 1) and 0x03FF (ch. register 2) each time a trigger is received.

Channel B memory address scroll just one time (ch. register 5) between memory address 0x0000 (ch. register 1) and 0x07FF (ch. register 2) each time a trigger is received.

A trigger is generated each time the board counter restarts. The counter maximum value (board registers 3 and 4) is set to 53,104,000 (0x032A4D80), which with the 53.104MHz clock cycle correspond to 1 trigger each second.

Board registers			Channel registers			
#	Address	Content	#	Address	Content Ch A	Content Ch B
0	0x 0000	b0000 0000 0000 0100	0	0x n000	bxxxx xxxx xxxx x1x1	bxxxx xxxx xxxx x1x1
1	0x 0001	N.A.	1	0x n001	b0000 0000 0000 0000	b0000 0000 0000 0000
2	0x 0002	N.A.	2	0x n002	b0000 0011 1111 1111	b0000 0111 1111 1111
3	0x 0003	b0100 1101 1000 0000	3	0x n003	bxxxx xxxx xxxx xxxx	bxxxx xxxx xxxx xxxx
4	0x 0004	b0000 0011 0010 1010	4	0x n004	bxxxx xxxx xxxx xxxx	bxxxx xxxx xxxx xxxx
5	0x 0005	N.A.	5	0x n005	b0000 0000 0000 0100	b0000 0000 0000 0001
6	0x 0006	bxxxx xxxx xxxx xxxx	6	0x n006	N.A.	N.A.
7	0x 0007	bxxxx xxxx xxxx xxxx	7	0x n007	N.A.	N.A.

Table 3.9, settings example for multi-channel timed trigger

Multi-channel software trigger

When channel are in the “multi-channel software trigger” mode (channel register 0) the trigger is generated at board level by a 0 to 1 transition of bit 4 in board register 0.

A channel in “multi-channel software trigger” mode runs synchronously to other channels in the same mode due to the trigger common source. The memory boundaries for each channel are specified in channel registers 1 and 2.

In the setting example of Table 3.10 each time a trigger is received both channel A and B memory addresses scroll between the established boundaries (ch. registers 1 and 2).

Channel A memory address loops sixteen times (ch. register 5) between memory address 0x0000 (ch. register 1) and 0x03FF (ch. register 2) each time a trigger is received.

Channel B memory address loops ten times (ch. register 5) between memory address 0x0000 (ch. register 1) and 0x00FF (ch. register 2) each time a trigger is received.

Board registers			Channel registers			
#	Address	Content	#	Address	Content Ch A	Content Ch B
0	0x 0000	b0000 0000 000↑ 0000	0	0x n000	bxxxx xxxx xxx1 xxx1	bxxxx xxxx xxx1 xxx1
1	0x 0001	N.A.	1	0x n001	b0000 0000 0000 0000	b0000 0000 0000 0000
2	0x 0002	N.A.	2	0x n002	b0000 0011 1111 1111	b0000 0000 1111 1111
3	0x 0003	bxxxx xxxx xxxx xxxx	3	0x n003	bxxxx xxxx xxxx xxxx	bxxxx xxxx xxxx xxxx
4	0x 0004	bxxxx xxxx xxxx xxxx	4	0x n004	bxxxx xxxx xxxx xxxx	bxxxx xxxx xxxx xxxx
5	0x 0005	N.A.	5	0x n005	b0000 0000 0001 0000	b0000 0000 0000 1010
6	0x 0006	bxxxx xxxx xxxx xxxx	6	0x n006	N.A.	N.A.
7	0x 0007	bxxxx xxxx xxxx xxxx	7	0x n007	N.A.	N.A.

Table 3.10, settings example for multi-channel manual trigger

Multi-channel external trigger

When channel are in the “multi-channel external trigger” mode (channel register 0) the trigger is generated at board level by push button SW2.

A channel in “multi-channel external trigger” mode runs synchronously to other channels in the same mode due to the trigger common source. The memory boundaries for each channel are specified in channel registers 1 and 2.

In the setting example of Table 3.11 each time a trigger is received, both channel A and B memory addresses scroll between the established boundaries (ch. registers 1 and 2).

Channel A memory address loops two times (ch. register 5) between memory address 0x0300 (ch. register 1) and 0x03FF (ch. register 2) each time a trigger is received.

Channel B memory address loops three times (ch. register 5) between memory address 0x0700 (ch. register 1) and 0x07FF (ch. register 2) each time a trigger is received.

A trigger is generated each time the push button SW2 is pressed.

Board registers			Channel registers			
#	Address	Content	#	Address	Content Ch A	Content Ch B
0	0x 0000	b0000 0000 0010 0000	0	0x n000	bxxxx xxxx xx1x xxx1	bxxxx xxxx xx1x xxx1
1	0x 0001	N.A.	1	0x n001	b0000 0011 0000 0000	b0000 0111 0000 0000
2	0x 0002	N.A.	2	0x n002	b0000 0011 1111 1111	b0000 0111 1111 1111
3	0x 0003	bxxxx xxxx xxxx xxxx	3	0x n003	bxxxx xxxx xxxx xxxx	bxxxx xxxx xxxx xxxx
4	0x 0004	bxxxx xxxx xxxx xxxx	4	0x n004	bxxxx xxxx xxxx xxxx	bxxxx xxxx xxxx xxxx
5	0x 0005	N.A.	5	0x n005	b0000 0000 0000 0010	b0000 0000 0000 0011
6	0x 0006	bxxxx xxxx xxxx xxxx	6	0x n006	N.A.	N.A.
7	0x 0007	bxxxx xxxx xxxx xxxx	7	0x n007	N.A.	N.A.

Table 3.11, settings example for multi-channel external trigger

Multi-channel SCL beam crossing trigger

When channel are in the “multi-channel SCL beam-crossing trigger” mode (channel register 0) the trigger is generated at board level each time the “SCL current beam-crossing” matches one of the two values specified in board register 7.

A channel in “multi-channel SCL beam-crossing trigger” mode runs synchronously to other channels in the same mode due to the trigger common source. The memory boundaries for each channel are specified in channel registers 1 and 2.

In the setting example of Table 3.12 each time a trigger is received both channel A and B memory addresses scroll between the established boundaries (ch. registers 1 and 2).

Each time a trigger is received both channel A and B memory addresses scrolls one time (ch. register 5) through the $159 \times 7 = 1113$ memory locations between address 0x0000 (ch. register 1) and 0x0458 (ch. register 2). The memory size will allow the coverage of a complete accelerator turn (1 turn = 159 beam crossings, 1 beam crossing = seven 53MHz clock cycles).

A trigger is generated each time the current beam-crossing value is 0 or 159 (0x9F, register 7). In the SCL system the valid beam crossing values are only those between 1 and 159.

Board registers			Channel registers			
#	Address	Content	#	Address	Content Ch A	Content Ch B
0	0x 0000	b0000 0010 0000 0000	0	0x n000	bxxxx xx1x xxxx xxx1	bxxxx xx1x xxxx xxx1
1	0x 0001	N.A.	1	0x n001	b0000 0000 0000 0000	b0000 0000 0000 0000
2	0x 0002	N.A.	2	0x n002	b0000 0100 0101 1000	b0000 0100 0101 1000
3	0x 0003	bxxxx xxxx xxxx xxxx	3	0x n003	bxxxx xxxx xxxx xxxx	bxxxx xxxx xxxx xxxx
4	0x 0004	bxxxx xxxx xxxx xxxx	4	0x n004	bxxxx xxxx xxxx xxxx	bxxxx xxxx xxxx xxxx
5	0x 0005	N.A.	5	0x n005	b0000 0000 0000 0001	b0000 0000 0000 0001
6	0x 0006	bxxxx xxxx xxxx xxxx	6	0x n006	N.A.	N.A.
7	0x 0007	b0000 0000 1001 1111	7	0x n007	N.A.	N.A.

Table 3.12, settings example for multi-channel SCL beam crossing trigger

Multi-channel SCL turn trigger

When channel are in the “multi-channel SCL turn trigger” mode (channel register 0) the trigger is generated at board level each time the “SCL current turn” matches one the value specified in board register 6.

A channel in “multi-channel SCL turn trigger” mode runs synchronously to other channels in the same mode due to the trigger common source. The memory boundaries for each channel are specified in channel registers 1 and 2.

In the setting example of Table 3.13 each time a trigger is received both channel A and B memory addresses scroll between the established boundaries (ch. registers 1 and 2).

Each time a trigger is received both channel A and B memory address scrolls one time (ch. register 5) through the $159 \times 7 = 1113$ memory locations between address 0x0000 (ch. register 1) and 0x0458 (ch. register 2). The memory size will allow the coverage of a complete accelerator turn (1 turn = 159 beam crossings, 1 beam crossing = seven 53MHz clock cycles).

A trigger is generated each time the current turn value is 16 (0x0010, register 6). In the SCL system the turn values are generated by a 16 bit counter (0 to 65535).

Board registers			Channel registers			
#	Address	Content	#	Address	Content Ch A	Content Ch B
0	0x 0000	b0000 0001 0000 0000	0	0x n000	bxxxx xxx1 xxxx xxx1	bxxxx xxx1 xxxx xxx1
1	0x 0001	N.A.	1	0x n001	b0000 0000 0000 0000	b0000 0000 0000 0000
2	0x 0002	N.A.	2	0x n002	b0000 0100 0101 1000	b0000 0100 0101 1000
3	0x 0003	bxxxx xxxx xxxx xxxx	3	0x n003	bxxxx xxxx xxxx xxxx	bxxxx xxxx xxxx xxxx
4	0x 0004	bxxxx xxxx xxxx xxxx	4	0x n004	bxxxx xxxx xxxx xxxx	bxxxx xxxx xxxx xxxx
5	0x 0005	N.A.	5	0x n005	b0000 0000 0000 0001	b0000 0000 0000 0001
6	0x 0006	b0000 0000 0001 0000	6	0x n006	N.A.	N.A.
7	0x 0007	bxxxx xxxx xxxx xxxx	7	0x n007	N.A.	N.A.

Table 3.13, settings example for multi-channel SCL turn trigger

Multi-channel SCL beam crossing and turn trigger

When channel are in the “multi-channel SCL turn & beam-crossing trigger” mode (channel register 0) the trigger is generated at board level each time both the “SCL current turn” and the “SCL current beam-crossing” matches the values specified in board register 6 and 7.

A channel in “multi-channel SCL turn & beam crossing trigger” mode runs synchronously to other channels in the same mode due to the trigger common source. The memory boundaries for each channel are specified in channel registers 1 and 2.

In the setting example of Table 3.14 each time a trigger is received both channel A and B memory addresses scroll between the established boundaries (ch. registers 1 and 2).

Each time a trigger is received both channel A and B memory address scrolls two times (ch. register 5) through the $159 \times 7 = 1113$ memory locations between address 0x0000 (ch. register 1) and 0x0458 (ch. register 2).

A trigger is generated each time the current turn value is 1 (0x0001, register 6) and the current beam-crossing is 159 (0x9F, register 7). In the SCL system the turn values are generated by a 16 bit counter (0 to 65535) and the valid beam crossing values are only those between 1 and 159.

Board registers			Channel registers			
#	Address	Content	#	Address	Content Ch A	Content Ch B
0	0x 0000	b0000 0100 0000 0000	0	0x n000	bxxxx x1xx xxxx xxx1	bxxxx x1xx xxxx xxx1
1	0x 0001	N.A.	1	0x n001	b0000 0000 0000 0000	b0000 0000 0000 0000
2	0x 0002	N.A.	2	0x n002	b0000 0100 0101 1000	b0000 0100 0101 1000
3	0x 0003	bxxxx xxxx xxxx xxxx	3	0x n003	bxxxx xxxx xxxx xxxx	bxxxx xxxx xxxx xxxx
4	0x 0004	bxxxx xxxx xxxx xxxx	4	0x n004	bxxxx xxxx xxxx xxxx	bxxxx xxxx xxxx xxxx
5	0x 0005	N.A.	5	0x n005	b0000 0000 0000 0010	b0000 0000 0000 0010
6	0x 0006	b0000 0000 0001 0000	6	0x n006	N.A.	N.A.
7	0x 0007	b0000 0000 1001 1111	7	0x n007	N.A.	N.A.

Table 3.14, settings example for multi-channel SCL beam crossing and turn trigger

3.3 Serial Port Interface

A Universal Asynchronous Receiver/Transmitter has been implemented in the Xilinx FPGA, which allows interfacing the card with a computer having a serial port. The Xilinx card uses an Analog Devices ADM3222 as RS-232 transceiver. The card behaves like a Data Communication Equipment (DCE) with the following settings: BAUD rate 115200, 1 start bit, 2 stop bits, no parity and no handshake. Paragraphs 3.3.1 and 3.3.2 describe the protocol for the WRITE and READ commands used to access the FPGA memory.

3.3.1 Write command

The write command protocol has the following structure.

Sequence	Action	Number of words	Command value	Data value	Valid data values
Step 1	Block Write Command Marker	2	0x1	0x0	0x0
			0x0	0x1 (write)	0x1
Step 2	Specify Initial Address	4	0x0	Address of row, LS Nibble	0x000 to 0xFFFF
			0x0	Address of row, Middle Nibble	
			0x0	Address of row, MS Nibble	
			0x0	Base Address of Memory Block	0x0, 0x1, 0x2
Step 3	Specify Number of 16 bits words	4	0x0	LS Nibble	0x0000 to 0x0FFF
			0x0	Middle Low Nibble	
			0x0	Middle High Nibble	
			0x0	MS Nibble (always 0x0)	
Step 4	Specify Data	1 + N*4	0x0	LS Nibble of 1 st data word	0x0000 to 0xFFFF
			0x0	Middle Low Nibble of 1 st data word	
			0x0	Middle High Nibble of 1 st data word	
			0x0	MS Nibble of 1 st data word	
			0x0	LS Nibble of 2 nd data word	0x0000 to 0xFFFF
			0x0	Middle Low Nibble of 2 nd data word	
			0x0	Middle High Nibble of 2 nd data word	
			0x0	MS Nibble of 2 nd data word	
			•	•	•
			0x0	LS Nibble of last data word	0x0000 to 0xFFFF
			0x0	Middle Low Nibble of last data word	
			0x0	Middle High Nibble of last data word	
0x0	MS Nibble of last data word				
Step 5	End command marker	2	0x1	0xF	0xF
			0x0	0x1 (write)	0x1

Table 3.15, Protocol structure of the write command

Length in bytes of a write command:

2 (Step1 bytes) + 4 (Step2 bytes) + 4 (Step3 bytes) +

N*4 (Step 4 with N 16-bit data words) + 2 (Step5 bytes) = 12 + N*4 bytes

The transmission is fully echoed. At the end of the transmission the user will find 12 + N*4 bytes in the RS232 receiver serial buffer of the computer.

3.3.2 Read command

The read command protocol has the following structure.

Sequence	Action	Number of words	Command value	Data value	Valid data values
Step 1	Block Read/Write Command Marker	2	0x1	0x0	0x0
			0x0	0x0 (read)	0x0
Step 2	Specify Initial Address	4	0x0	Address of row, LS Nibble	0x000 to 0xFFFF
			0x0	Address of row, Middle Nibble	
			0x0	Address of row, MS Nibble	
			0x0	Base Address of Memory Block	0x0, 0x1, 0x2
Step 3	Specify Number of 16 bits words	4	0x0	LS Nibble	0x0000 to 0x0FFF
			0x0	Middle Low Nibble	
			0x0	Middle High Nibble	
			0x0	MS Nibble (always 0x0)	
Step 4	End command marker	2	0x1	0xF	0xF
			0x0	0x0 (read)	0x0
Step 5	<u>Readback Data</u>	1 + N*4	0x0	LS Nibble of 1 st data word	0x0000 to 0xFFFF
			0x0	Middle Low Nibble of 1 st data word	
			0x0	Middle High Nibble of 1 st data word	
			0x0	MS Nibble of 1 st data word	0x0000 to 0xFFFF
			0x0	LS Nibble of 2 nd data word	
			0x0	Middle Low Nibble of 2 nd data word	
			0x0	Middle High Nibble of 2 nd data word	0x0000 to 0xFFFF
			0x0	MS Nibble of 2 nd data word	
			•	•	
			0x0	LS Nibble of last data word	0x0000 to 0xFFFF
			0x0	Middle Low Nibble of last data word	
			0x0	Middle High Nibble of last data word	
0x0	MS Nibble of last data word				

Table 3.16, Protocol structure of the read command

Step 5 in Table 3.16 (gray background) is not part of the read command; it is the block of read data appended to the command echo by the board RS232 controller on the Xilinx FPGA.

Length in bytes of a read command:

$$2 \text{ (Step1 bytes)} + 4 \text{ (Step2 bytes)} + 4 \text{ (Step3 bytes)} + 2 \text{ (Step5 bytes)} + M*4 \text{ bytes (M 16-bit data words)} = 12 + M*4 \text{ bytes}$$

The transmission is fully echoed. At the end of the transmission the user will find 12 bytes in the RS232 receiver serial buffer of the computer. In addition to the 12 echoed bytes the user will also find the M*4 bytes, were M is the number of 16 bits words to be read. The words read comply with the same format used in the protocol, LS nibble first, MS nibble last.

The following table provides the decoding information for the “command error word”. The word is generated each time a command is received by the RS-232 controller on the Xilinx FPGA. The error word can be accessed through the board LEDs with an appropriate setting of the board DIP switches (see Table 3.18).

Bit	Notes
0	Set if the command state machine is/has been in State 0
1	Set if the command state machine is/has been in State 1
2	Set if the command state machine is/has been in State 2
3	Set if the command state machine is/has been in State 3
4	Set if the command state machine is/has been in State 4
5	Set if the command state machine is/has been in State 5
6	Set if the command state machine is/has been in State 6
7	Set if the command state machine is/has been in State 7
8	Set if the command state machine is/has been in State 8
9	Set if the command state machine is/has been in State 9
10	Set if the command state machine is/has been in State 10
11	Set if the command state machine is/has been in State 11
12	Set if the command state machine is/has been in State 12
13..15	000: No error 001: Command error 010: Protocol error 011: End Command Header error 100: End Command Marker error 101: 110: 111: Out of state error

Table 3.17, Command error word

3.4 Board switches/buttons mapping

The dip-switches on the test board allows to select which signals are used to drive the set of 8 LEDs. The two buttons are used one for reset and the other to trigger the transmission of a message by the RS232 transmitter.

Name	PCB Name	Standard Setting	Notes
SWITCH[3..0]	S1 switch[4..1]		Used to select the signals driving the LEDs 0000 LED scan 0001 Temperature 0010 RS232 Receiver (received data) 0011 RS232 Transmitter (data driving the transmitter input) 0100 Command Error Word lower byte 0101 Command Error Word upper byte 0110 Board Control/Status Register 0 lower byte 0111 Board Control/Status Register 0 upper byte 1000 1001 1010 1011 1100 1101 1110 1111
SWITCH[4]	S1 switch[5]		
SWITCH[5]	S1 switch[6]	ON	Command Mode Enable (enables command recognition)
SWITCH[6]	S1 switch[7]	ON	RS232 receiver enable
SWITCH[7]	S1 switch[8]	ON	RS232 transmitter enable
SWITCH[8]	SW1 push button		Reset button
SWITCH[9]	SW2 push button		The button is used for two functions: 1) Serial port message button (identify firmware version) when command recognition is disabled. 2) External trigger generation when command recognition is enabled.

Table 3.18, Board switches/buttons mapping.

3.4.1 Logic Analyzer Connectors

For debugging/diagnostic purposes the Xilinx card has three Amp "Mictor 38" connectors (AMP 2-767004-2) to fit the Agilent (formerly HP) High Density adapter cables (Agilent E5346A). Each adapter cable fits two test pods of an Agilent 16550A Logic Analyzer allowing monitoring up to 38 signals. The following three tables list the signals connected to the Xilinx test card's logic analyzer connector.

Notes [FPGA Pin#]	Schematic name	Pin #	Pin name	Pin name	Pin #	Schematic name	Notes [FPGA Pin#]
	N.C.	1	+5VDC	SCL	2	N.C.	
	N.C.	3	GND	SDA	4	N.C.	
[P182]	CLK_OUT	5	CLK1	CLK2	6	OSC	[P80]
DAC_CH1_DATA(7) [P45]	ADDRESS31	7	P1.15	P2.15	8	ADDRESS15	[P22]
[P44]	ADDRESS30	9	P1.14	P2.14	10	ADDRESS14	DAC_CH2_DATA(4) [P21]
DAC_CH1_DATA(6) [P43]	ADDRESS29	11	P1.13	P2.13	12	ADDRESS13	[P13]
[P42]	ADDRESS28	13	P1.12	P2.12	14	ADDRESS12	DAC_CH2_DATA(5) [P18]
DAC_CH1_DATA(5) [P41]	ADDRESS27	15	P1.11	P2.11	16	ADDRESS11	[P17]
[P37]	ADDRESS26	17	P1.10	P2.10	18	ADDRESS10	DAC_CH2_DATA(6) [P16]
DAC_CH1_DATA(4) [P36]	ADDRESS25	19	P1.09	P2.09	20	ADDRESS9	[P15]
[P35]	ADDRESS24	21	P1.08	P2.08	22	ADDRESS8	DAC_CH2_DATA(7) [P14]
DAC_CH1_DATA(3) [P34]	ADDRESS23	23	P1.07	P2.07	24	ADDRESS7	DAC_CH1_CLK [P10]
DAC_CH2_DATA(0) [P33]	ADDRESS22	25	P1.06	P2.06	26	ADDRESS6	DAC_CH2_DATA(8) [P9]
DAC_CH1_DATA(2) [P31]	ADDRESS21	27	P1.05	P2.05	28	ADDRESS5	[P8]
DAC_CH2_DATA(1) [P30]	ADDRESS20	29	P1.04	P2.04	30	ADDRESS4	DAC_CH2_DATA(9) [P7]
DAC_CH1_DATA(1) [P29]	ADDRESS19	31	P1.03	P2.03	32	ADDRESS3	[P6]
DAC_CH2_DATA(2) [P27]	ADDRESS18	33	P1.02	P2.02	34	ADDRESS2	DAC_CH2_DATA(10) [P5]
DAC_CH1_DATA(0) [P24]	ADDRESS17	35	P1.01	P2.01	36	ADDRESS1	[P4]
DAC_CH2_DATA(3) [P23]	ADDRESS16	37	P1.00	P2.00	38	ADDRESS0	DAC_CH2_DATA(11) [P3]

Table 3.19, HP Logic Analyzer connector J4 (suggested analyzer pods: 1 and 2)

Notes [FPGA Pin#]	Schematic name	Pin #	Pin name	Pin name	Pin #	Schematic name	Notes [FPGA Pin#]
	N.C.	1	+5VDC	SCL	2	N.C.	
	N.C.	3	GND	SDA	4	N.C.	
[P185]	GCK3	5	CLK1	CLK2	6	GCLK1	[P77]
[P162]	DATA31	7	P1.15	P2.15	8	DATA15	SCL_CURRENT_TURN[15] [P187]
[P163]	DATA30	9	P1.14	P2.14	10	DATA14	SCL_CURRENT_TURN[14] [P188]
[P164]	DATA29	11	P1.13	P2.13	12	DATA13	SCL_CURRENT_TURN[13] [P189]
SCL_SPARE_PERIOD [P165]	DATA28	13	P1.12	P2.12	14	DATA12	SCL_CURRENT_TURN[12] [P191]
SCL_COSMIC_GAP [P166]	DATA27	15	P1.11	P2.11	16	DATA11	SCL_CURRENT_TURN[11] [P192]
SCL_SYNC_GAP [P167]	DATA26	17	P1.10	P2.10	18	DATA10	SCL_CURRENT_TURN[10] [P193]
SCL_BEAM_PERIOD [P168]	DATA25	19	P1.09	P2.09	20	DATA9	SCL_CURRENT_TURN[9] [P194]
SCL_FIRST_PERIOD [P172]	DATA24	21	P1.08	P2.08	22	DATA8	SCL_CURRENT_TURN[8] [P195]
SCL_CURRENT_BX[7] [P173]	DATA23	23	P1.07	P2.07	24	DATA7	SCL_CURRENT_TURN[7] [P199]
SCL_CURRENT_BX[6] [P174]	DATA22	25	P1.06	P2.06	26	DATA6	SCL_CURRENT_TURN[6] [P200]
SCL_CURRENT_BX[5] [P175]	DATA21	27	P1.05	P2.05	28	DATA5	SCL_CURRENT_TURN[5] [P201]
SCL_CURRENT_BX[4] [P176]	DATA20	29	P1.04	P2.04	30	DATA4	SCL_CURRENT_TURN[4] [P202]
SCL_CURRENT_BX[3] [P178]	DATA19	31	P1.03	P2.03	32	DATA3	SCL_CURRENT_TURN[3] [P203]
SCL_CURRENT_BX[2] [P179]	DATA18	33	P1.02	P2.02	34	DATA2	SCL_CURRENT_TURN[2] [P204]
SCL_CURRENT_BX[1] [P180]	DATA17	35	P1.01	P2.01	36	DATA1	SCL_CURRENT_TURN[1] [P205]
SCL_CURRENT_BX[0] [P181]	DATA16	37	P1.00	P2.00	38	DATA0	SCL_CURRENT_TURN[0] [P206]

Table 3.20, HP Logic Analyzer connector J5 (suggested analyzer pods: 3 and 4)

Notes [FPGA Pin#]	Schematic name	Pin #	Pin name	Pin name	Pin #	Schematic name	Notes [FPGA Pin#]
	N.C.	1	+5VDC	SCL	2	N.C.	
	N.C.	3	GND	SDA	4	N.C.	
SCL_CLK_7 [P182]	CLK_OUT	5	CLK1	CLK2	6	CLK_IN	SCL_CLK_53 [P147]
[P154]	DOUT	7	P1.15	P2.15	8	CNTL15	[P81]
[P109]	SWITCH9	9	P1.14	P2.14	10	CNTL14	[P75]
[P110]	SWITCH8	11	P1.13	P2.13	12	CNTL13	[P74]
[P94]	TEMP_SDO	13	P1.12	P2.12	14	CNTL12	[P73]
[P95]	TEMP_SDI	15	P1.11	P2.11	16	CNTL11	[P71]
[P96]	TEMP_CE	17	P1.10	P2.10	18	CNTL10	[P70]
[P97]	TEMP_SCLK	19	P1.09	P2.09	20	CNTL9	[P69]
[P149]	RS232SD_N	21	P1.08	P2.08	22	CNTL8	[P68]
[P150]	RS232EN_N	23	P1.07	P2.07	24	CNTL7	DAC_CH1_DATA(11) [P67]
[P151]	RS232TX	25	P1.06	P2.06	26	CNTL6	[P63]
[P152]	RS232RX	27	P1.05	P2.05	28	CNTL5	DAC_CH1_DATA(10) [P62]
SCL_ACK [P87]	CNTL20	29	P1.04	P2.04	30	CNTL4	[P61]
SCL_DATA_ERROR [P86]	CNTL19	31	P1.03	P2.03	32	CNTL3	DAC_CH1_DATA(9) [P60]
SCL_SYNC_ERROR [P84]	CNTL18	33	P1.02	P2.02	34	CNTL2	[P59]
SCL_READY [P83]	CNTL17	35	P1.01	P2.01	36	CNTL1	DAC_CH1_DATA(8) [P58]
[P82]	CNTL16	37	P1.00	P2.00	38	CNTL0	DAC_CH2_CLK [P57]

Table 3.21, HP Logic Analyzer connector J6 (suggested analyzer pods: 5 and 6)

3.4.2 Header Connectors

The Xilinx card has two 25x2 Header connectors. One of the header connectors (JP5) is used to interface the Xilinx card to the Analog Devices AD9765 DAC card. MEMCH1_DATOUT_D(11..0) and MEMCH1_DATOUT_D(11..0) are the 12-bit wide buses used to drive the DAC's input channel 1 and input channel 2. The signal "CLK" is a copy of the FPGA 53.104 MHz clock and is used as DAC clock on the AD9765 DAC card.

The following two tables provide the pinouts of the Xilinx card's header connectors.

Notes [FPGA Pin#]	Schematic name	Pin #		Pin #	Schematic name	Notes [FPGA Pin#]
DAC_CH2_DATA(11) [P3]	ADDRESS0	1		2	ADDRESS1	[P4]
DAC_CH2_DATA(10) [P5]	ADDRESS2	3		4	ADDRESS3	[P6]
DAC_CH2_DATA(9) [P7]	ADDRESS4	5		6	ADDRESS5	[P8]
DAC_CH2_DATA(8) [P9]	ADDRESS6	7		8	ADDRESS7	DAC_CH1_CLK [P10]
DAC_CH2_DATA(7) [P14]	ADDRESS8	9		10	ADDRESS9	[P15]
DAC_CH2_DATA(6) [P16]	ADDRESS10	11		12	ADDRESS11	[P17]
DAC_CH2_DATA(5) [P18]	ADDRESS12	13		14	ADDRESS13	[P13]
DAC_CH2_DATA(4) [P21]	ADDRESS14	15		16	ADDRESS15	[P22]
DAC_CH2_DATA(3) [P23]	ADDRESS16	17		18	ADDRESS17	DAC_CH1_DATA(0) [P24]
DAC_CH2_DATA(2) [P27]	ADDRESS18	19		20	ADDRESS19	DAC_CH1_DATA(1) [P29]
DAC_CH2_DATA(1) [P30]	ADDRESS20	21		22	ADDRESS21	DAC_CH1_DATA(2) [P31]
DAC_CH2_DATA(0) [P33]	ADDRESS22	23		24	ADDRESS23	DAC_CH1_DATA(3) [P34]
[P35]	ADDRESS24	25		26	ADDRESS25	DAC_CH1_DATA(4) [P36]
[P37]	ADDRESS26	27		28	ADDRESS27	DAC_CH1_DATA(5) [P41]
[P42]	ADDRESS28	29		30	ADDRESS29	DAC_CH1_DATA(6) [P43]
[P44]	ADDRESS30	31		32	ADDRESS31	DAC_CH1_DATA(7) [P45]
DAC_CH2_CLK [P57]	CNTL0	33		34	CNTL1	DAC_CH1_DATA(8) [P58]
[P59]	CNTL2	35		36	CNTL3	DAC_CH1_DATA(9) [P60]
[P61]	CNTL4	37		38	CNTL5	DAC_CH1_DATA(10) [P62]
[P63]	CNTL6	39		40	CNTL7	DAC_CH1_DATA(11) [P67]
[P68]	CNTL8	41		42	CNTL9	[P69]
[P70]	CNTL10	43		44	PWDN#	[P55]
[P80] (*)	OSC	45		46	STATUS	[P56]
[P154]	DOUT	47		48	GND	
	GND	49		50	GND	

Table 3.22, Header connector JP5

Notes [FPGA Pin#]	Schematic name	Pin #		Pin #	Schematic name	Notes [FPGA Pin#]
SCL_CURRENT_TURN[0] [P206]	DATA0	1		2	DATA1	SCL_CURRENT_TURN[1] [P205]
SCL_CURRENT_TURN[2] [P204]	DATA2	3		4	DATA3	SCL_CURRENT_TURN[3] [P203]
SCL_CURRENT_TURN[4] [P202]	DATA4	5		6	DATA5	SCL_CURRENT_TURN[5] [P201]
SCL_CURRENT_TURN[6] [P200]	DATA6	7		8	DATA7	SCL_CURRENT_TURN[7] [P199]
SCL_CURRENT_TURN[8] [P195]	DATA8	9		10	DATA9	SCL_CURRENT_TURN[9] [P194]
SCL_CURRENT_TURN[10] [P193]	DATA10	11		12	DATA11	SCL_CURRENT_TURN[11] [P192]
SCL_CURRENT_TURN[12] [P191]	DATA12	13		14	DATA13	SCL_CURRENT_TURN[13] [P189]
SCL_CURRENT_TURN[14] [P188]	DATA14	15		16	DATA15	SCL_CURRENT_TURN[15] [P187]
SCL_CURRENT_BX[0] [P181]	DATA16	17		18	DATA17	SCL_CURRENT_BX[1] [P180]
SCL_CURRENT_BX[2] [P179]	DATA18	19		20	DATA19	SCL_CURRENT_BX[3] [P178]
SCL_CURRENT_BX[4] [P176]	DATA20	21		22	DATA21	SCL_CURRENT_BX[5] [P175]
SCL_CURRENT_BX[6] [P174]	DATA22	23		24	DATA23	SCL_CURRENT_BX[7] [P173]
SCL_FIRST_PERIOD [P172]	DATA24	25		26	DATA25	SCL_BEAM_PERIOD [P168]
SCL_SYNC_GAP [P167]	DATA26	27		28	DATA27	SCL_COSMIC_GAP [P166]
SCL_SPARE_PERIOD [P165]	DATA28	29		30	DATA29	[P164]
[P163]	DATA30	31		32	DATA31	[P162]
[P71]	CNTL11	33		34	CNTL12	[P73]
[P74]	CNTL13	35		36	CNTL14	[P75]
[P81]	CNTL15	37		38	CNTL16	[P82]
SCL_READY [P83]	CNTL17	39		40	CNTL18	SCL_SYNC_ERROR [P84]
SCL_DATA_ERROR [P86]	CNTL19	41		42	CNTL20	SCL_ACK [P87]
[P77]	GCK1	43		44	CLK_IN	SCL_CLK_53 [P147] (*)
[P185]	GCK3	45		46	CLK_OUT	SCL_CLK_7 [P182] (*)
[P182] (*)	CLK_OUT_F B	47		48	GND	
	GND	49		50	GND	

Table 3.23, Header connector JP6

Notes [FPGA Pin#]	Schematic name	Pin #	Pin #	Schematic name	Notes [FPGA Pin#]
	NC (Res. 5.0V)	1	71	ADDRESS0	DAC_CH2_DATA(11) [P3]
[P4]	ADDRESS1	2	72	GND	
DAC_CH2_DATA(10) [P5]	ADDRESS2	3	73	ADDRESS3	[P6]
	GND	4	74	ADDRESS4	DAC_CH2_DATA(9) [P7]
[P8]	ADDRESS5	5	75	GND	
DAC_CH2_DATA(8) [P9]	ADDRESS6	6	76	ADDRESS7	DAC_CH1_CLK [P10]
	GND	7	77	ADDRESS8	DAC_CH2_DATA(7) [P14]
[P15]	ADDRESS9	8	78	3.3V	
DAC_CH2_DATA(6) [P16]	ADDRESS10	9	79	ADDRESS11	[P17]
	GND	10	80	ADDRESS12	DAC_CH2_DATA(5) [P18]
[P13]	ADDRESS13	11	81	GND	
DAC_CH2_DATA(4) [P21]	ADDRESS14	12	82	ADDRESS15	[P22]
	NC (Res. 5.0V)	13	83	ADDRESS16	DAC_CH2_DATA(3) [P23]
DAC_CH1_DATA(0) [P24]	ADDRESS17	14	84	GND	
DAC_CH2_DATA(2) [P27]	ADDRESS18	15	85	ADDRESS19	DAC_CH1_DATA(1) [P29]
	GND	16	86	ADDRESS20	DAC_CH2_DATA(1) [P30]
DAC_CH1_DATA(2) [P31]	ADDRESS21	17	87	GND	
DAC_CH2_DATA(0) [P33]	ADDRESS22	18	88	ADDRESS23	DAC_CH1_DATA(3) [P34]
	GND	19	89	ADDRESS24	[P35]
DAC_CH1_DATA(4) [P36]	ADDRESS25	20	90	3.3V	
[P37]	ADDRESS26	21	91	ADDRESS27	DAC_CH1_DATA(5) [P41]
	GND	22	92	ADDRESS28	[P42]
DAC_CH1_DATA(6) [P43]	ADDRESS29	23	93	GND	
[P44]	ADDRESS30	24	94	ADDRESS31	DAC_CH1_DATA(7) [P45]
	NC (Res. 5.0V)	25	95	DATA0	SCL_CURRENT_TURN(0) [P206]
SCL_CURRENT_TURN(1) [P205]	DATA1	26	96	GND	
SCL_CURRENT_TURN(2) [P204]	DATA2	27	97	DATA3	SCL_CURRENT_TURN(3) [P203]
	GND	28	98	DATA4	SCL_CURRENT_TURN(4) [P202]
SCL_CURRENT_TURN(5) [P201]	DATA5	29	99	GND	
SCL_CURRENT_TURN(6) [P200]	DATA6	30	100	DATA7	SCL_CURRENT_TURN(7) [P199]
	GND	31	101	DATA8	SCL_CURRENT_TURN(8) [P195]
SCL_CURRENT_TURN(9) [P194]	DATA9	32	102	3.3V	
SCL_CURRENT_TURN(10) [P193]	DATA10	33	103	DATA11	SCL_CURRENT_TURN(11) [P192]
	GND	34	104	DATA12	SCL_CURRENT_TURN(12) [P191]
SCL_CURRENT_TURN(13) [P189]	DATA13	35	105	GND	
SCL_CURRENT_TURN(14) [P188]	DATA14	36	106	DATA15	SCL_CURRENT_TURN(15) [P187]
	NC (Res. 5.0V)	37	107	DATA16	SCL_CURRENT_BX(0) [P181]

Table 3.24, Header connector P2-A (AvBus)

Notes [FPGA Pin#]	Schematic name	Pin #		Pin #	Schematic name	Notes [FPGA Pin#]
SCL_CURRENT_BX(1) [P180]	DATA17	38		108	GND	
SCL_CURRENT_BX(2) [P179]	DATA18	39		109	DATA19	SCL_CURRENT_BX(3) [P178]
	GND	40		110	DATA20	SCL_CURRENT_BX(4) [P176]
SCL_CURRENT_BX(5) [P175]	DATA21	41		111	GND	
SCL_CURRENT_BX(6) [P174]	DATA22	42		112	DATA23	SCL_CURRENT_BX(7) [P173]
	GND	43		113	DATA24	SCL_FIRST_PERIOD [P172]
SCL_BEAM_PERIOD [P168]	DATA25	44		114	3.3V	
SCL_SYNC_GAP [P167]	DATA26	45		115	DATA27	SCL_COSMIC_GAP [P166]
	GND	46		116	DATA28	SCL_SPARE_PERIOD [P165]
[P164]	DATA29	47		117	GND	
[P163]	DATA30	48		118	DATA31	[P162]
	NC (Res. 5.0V)	49		119	CNTL0	DAC_CH2_CLK [P57]
DAC_CH1_DATA(8) [P58]	CNTL1	50		120	GND	
[P59]	CNTL2	51		121	CNTL3	DAC_CH1_DATA(9) [P60]
	GND	52		122	CNTL4	[P61]
DAC_CH1_DATA(10) [P62]	CNTL5	53		123	GND	
[P63]	CNTL6	54		124	CNTL7	DAC_CH1_DATA(11) [P67]
	GND	55		125	CNTL8	[P68]
[P69]	CNTL9	56		126	3.3V	
[P70]	CNTL10	57		127	CNTL11	[P71]
	GND	58		128	CNTL12	[P73]
[P74]	CNTL13	59		129	GND	
[P75]	CNTL14	60		130	CNTL15	[P81]
	NC (Res. 5.0V)	61		131	CNTL16	[P82]
SCL_READY [P83]	CNTL17	62		132	GND	
SCL_SYNC_ERROR [P84]	CNTL18	63		133	CNTL19	SCL_DATA_ERROR [P86]
	GND	64		134	CNTL20	SCL_ACK [P87]
SCL_CLK_53 [P147] (*)	CLK_IN	65		135	GND	
[P182] (*)	CLK_OUT_FB	66		136	CLK_OUT	SCL_CLK_7 [P182] (*)
	GND	67		137	JTAG_TMS	[P100]
[P99]	JTAG_TDO	68		138	3.3V	
[P98]	JTAG_TCK	69		139	JTAG_TDI	[P101]
	GND	70		140	JTAG_TRS	[P102]

Table 3.25, Header connector P2-B (AvBus)

3.5 JTAG Interface

See a description of JTAG in Paragraph 12. One four pin Test Access Port (TAP) is provided on the Xilinx card, it accesses the Xilinx FPGA and its configuration EEPROM (Table 3.26).

☒	1 Vcc (+3.3V)
☒	2 TDI
☒	3 TMS
☒	4 TCK
☒	5 TDO
☒	6 GND

Figure 3.13, JTAG Connectors pinout

JTAG Chain#1, connector CON3		
Device #	Device Type	Notes
1	XC18V01SO20C	In-System-Programmable Configuration PROM.
2	XC2S150-5PQ208C	Spartan II FPGA

Table 3.26, JTAG Chain Devices

4 AD9765 Test Card

The Analog Devices AD9765 Evaluation Board is used in the TWG prototype system as a two channel Digital to Analog Converter card. The DAC input data and the clock are provided by the Xilinx card. The outputs are configured in differential mode and in a way to obtain a 1 Volt peak-to-peak waveforms at the input of the next set of cards, the AD8138.

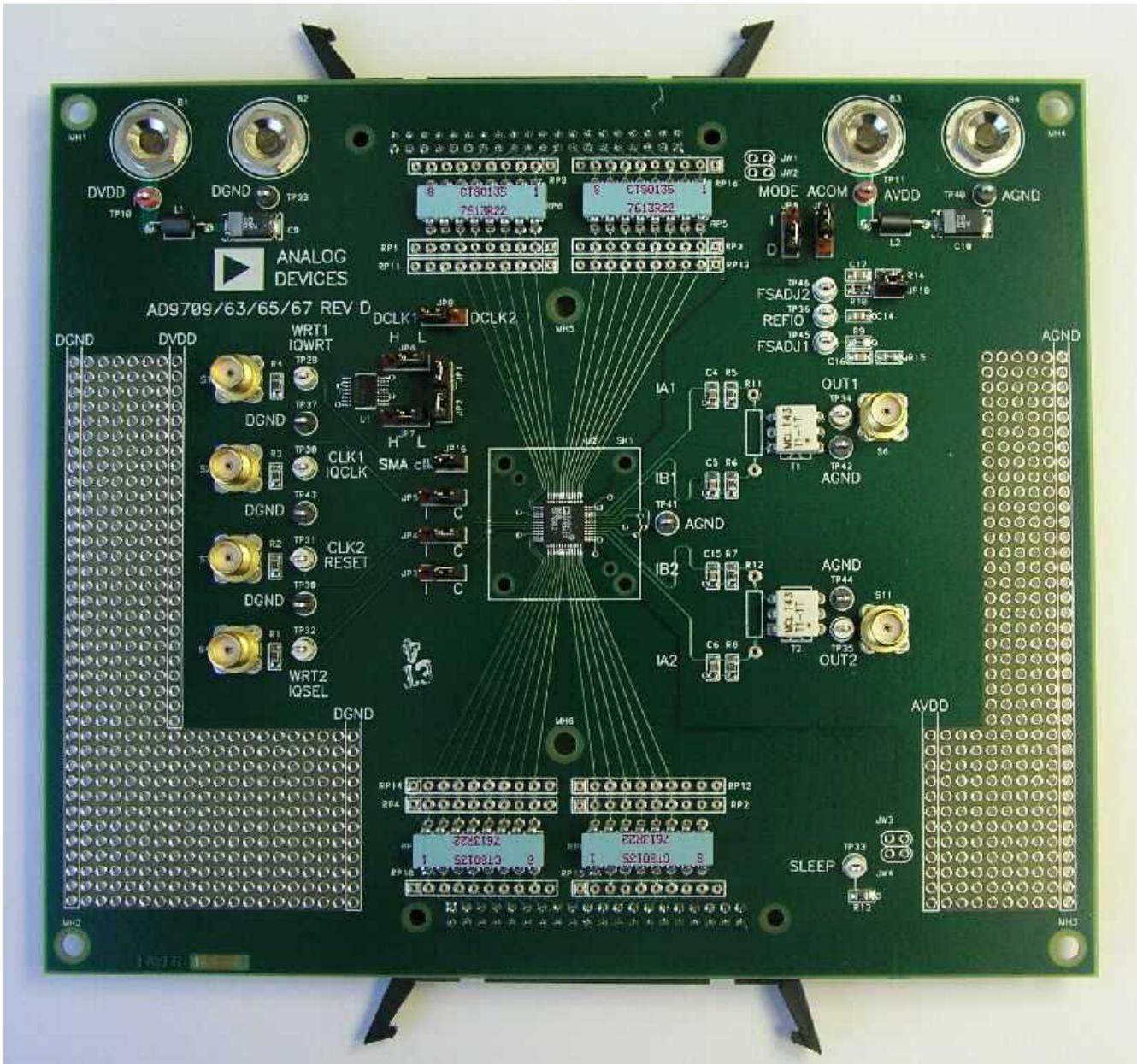


Figure 4.1, AD9765 evaluation board

Notes AD9765 pin name (Connector P1) [AD9765 Pin#]	Xilinx Card Schematic name	Pin #		Pin #	Xilinx Card Schematic name	Notes AD9765 pin name (Connector P1) [AD9765 Pin#]
DB11-P1MSB (DUTP1) [1]	CNTL7	1		2	Flat Cable N.C.	(GND)
DB10-P1 (DUTP2) [2]	CNTL5	3		4	Flat Cable N.C.	(GND)
DB9-P1 (DUTP3) [3]	CNTL3	5		6	Flat Cable N.C.	(GND)
DB8-P1 (DUTP4) [4]	CNTL1	7		8	Flat Cable N.C.	(GND)
DB7-P1 (DUTP5) [5]	ADDRESS31	9		10	Flat Cable N.C.	(GND)
DB6-P1 (DUTP6) [6]	ADDRESS29	11		12	Flat Cable N.C.	(GND)
DB5-P1 (DUTP7) [7]	ADDRESS27	13		14	Flat Cable N.C.	(GND)
DB4-P1 (DUTP8) [8]	ADDRESS25	15		16	Flat Cable N.C.	(GND)
DB3-P1 (DUTP9) [9]	ADDRESS23	17		18	Flat Cable N.C.	(GND)
DB2-P1 (DUTP10) [10]	ADDRESS21	19		20	Flat Cable N.C.	(GND)
DB1-P1 (DUTP11) [11]	ADDRESS19	21		22	Flat Cable N.C.	(GND)
DB0-P1 (DUTP12) [12]	ADDRESS17	23		24	Flat Cable N.C.	(GND)
NC (DUTP13) [13]	ADDRESS15	25		26	Flat Cable N.C.	(GND)
NC (DUTP14) [14]	ADDRESS13	27		28	Flat Cable N.C.	(GND)
(Open)	ADDRESS11	29		30	Flat Cable N.C.	(GND)
(Open)	ADDRESS9	31		32	Flat Cable N.C.	(GND)
CLK1 (DCLKIN1) [18]	ADDRESS7	33		34	Flat Cable N.C.	(GND)
(Open)	ADDRESS5	35		36	Flat Cable N.C.	(GND)
(Open)	ADDRESS3	37		38	Flat Cable N.C.	(GND)
(Open)	ADDRESS1	39		40	Flat Cable N.C.	(GND)

Table 4.1, Header connector P1 (DAC Channel 1)

Notes AD9765 pin name (Connector P1) [AD9765 Pin#]	Xilinx Card Schematic name	Pin #		Pin #	Xilinx Card Schematic name	Notes AD9765 pin name (Connector P1) [AD9765 Pin#]
DB11-P2MSB (DUTP23) [23]	ADDRESS0	1		2	Flat Cable N.C.	(GND)
DB10-P2 (DUTP24) [24]	ADDRESS2	3		4	Flat Cable N.C.	(GND)
DB9-P2 (DUTP25) [25]	ADDRESS4	5		6	Flat Cable N.C.	(GND)
DB8-P2 (DUTP26) [26]	ADDRESS6	7		8	Flat Cable N.C.	(GND)
DB7-P2 (DUTP27) [27]	ADDRESS8	9		10	Flat Cable N.C.	(GND)
DB6-P2 (DUTP28) [28]	ADDRESS10	11		12	Flat Cable N.C.	(GND)
DB5-P2 (DUTP29) [29]	ADDRESS12	13		14	Flat Cable N.C.	(GND)
DB4-P2 (DUTP30) [30]	ADDRESS14	15		16	Flat Cable N.C.	(GND)
DB3-P2 (DUTP31) [31]	ADDRESS16	17		18	Flat Cable N.C.	(GND)
DB2-P2 (DUTP32) [32]	ADDRESS18	19		20	Flat Cable N.C.	(GND)
DB1-P2 (DUTP33) [33]	ADDRESS20	21		22	Flat Cable N.C.	(GND)
DB0-P2 (DUTP34) [34]	ADDRESS22	23		24	Flat Cable N.C.	(GND)
NC (DUTP35) [35]	ADDRESS24	25		26	Flat Cable N.C.	(GND)
NC (DUTP36) [36]	ADDRESS26	27		28	Flat Cable N.C.	(GND)
(Open)	ADDRESS28	29		30	Flat Cable N.C.	(GND)
(Open)	ADDRESS30	31		32	Flat Cable N.C.	(GND)
CLK2 (DCLKIN2) [19]	CNTL0	33		34	Flat Cable N.C.	(GND)
(Open)	CNTL2	35		36	Flat Cable N.C.	(GND)
(Open)	CNTL4	37		38	Flat Cable N.C.	(GND)
(Open)	CNTL6	39		40	Flat Cable N.C.	(GND)

Table 4.2, Header connector P2 (DAC Channel 2)

5 AD8138 Test Card

Two Analog Devices [7] AD8138 Evaluation Boards are used to implement the first stage of amplification/filtering (differential) for the two channels. The 1 Volt peak-to-peak signals from the DAC are amplified by a factor of 7 to 7 Volt peak-to-peak signals with a common mode of 0 Volts. The differential outputs of the AD8138 cards drive the second stage of amplification implemented with THS4141 cards.

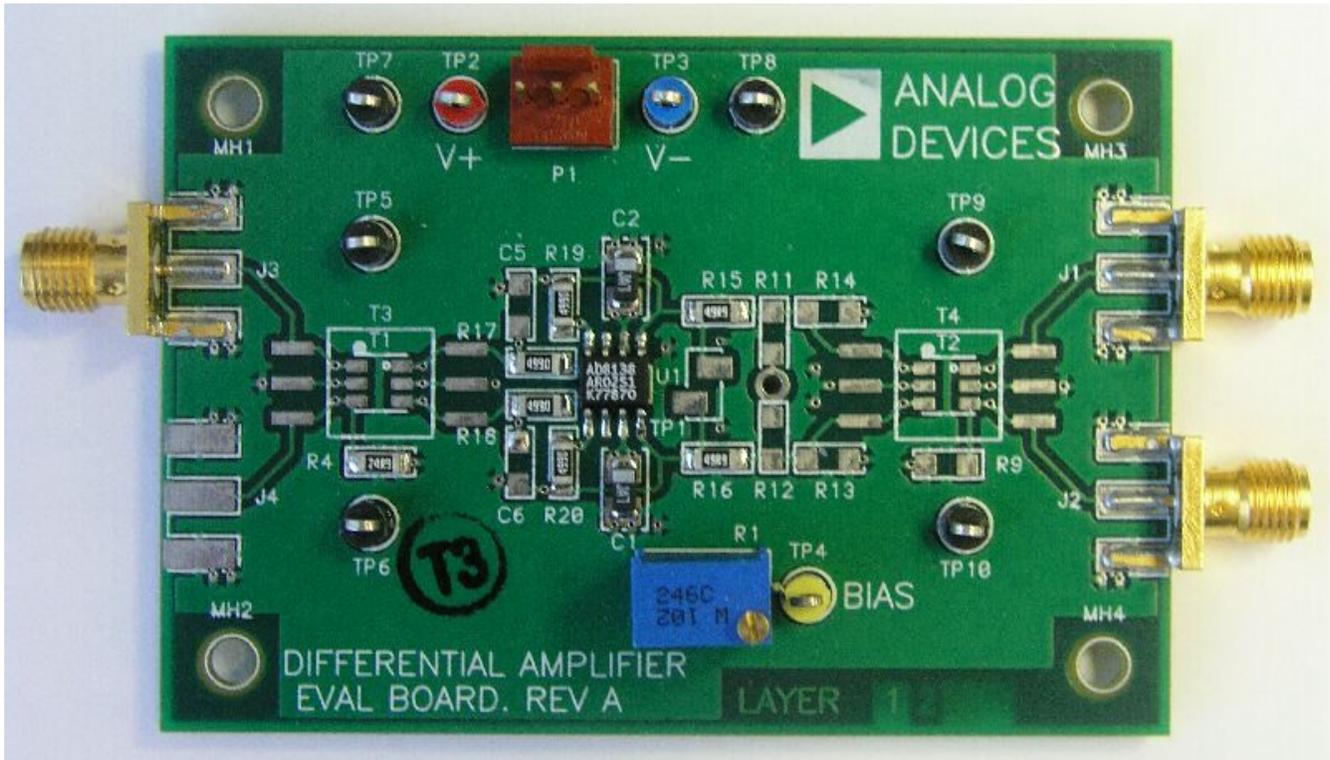


Figure 5.1, AD8138 evaluation board

6 THS4141 Test Card

Two Texas Instrument [9] THS4141 Evaluation Boards are used implement the second stage of amplification/filtering (differential) for the two channels. The 7 Volt peak-to-peak signals from the DAC are amplified by a factor of 2 to 14 Volt peak-to-peak signals with a common mode of 0 Volts. The differential outputs of the THS4141 cards are also the output of the TWG prototype system and cabled to an appropriate connector for interfacing to the L1 Cal ADF system.

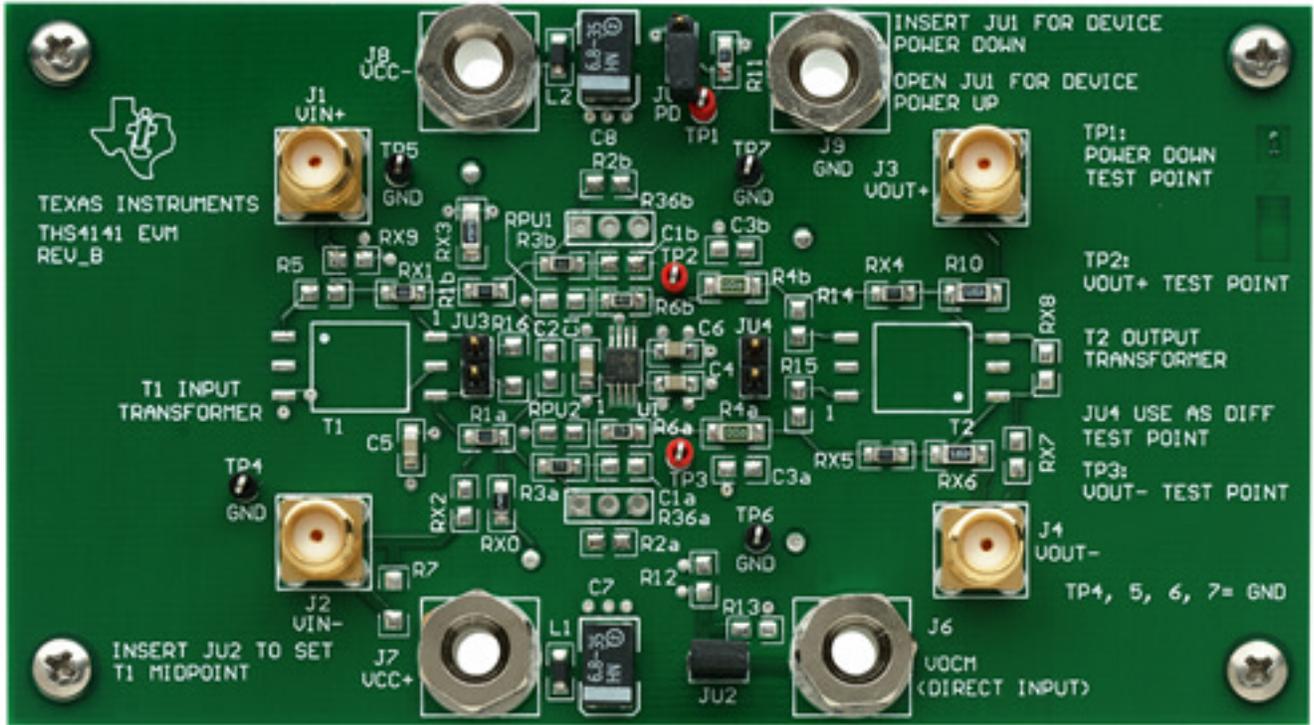


Figure 6.1, THS4141 evaluation board

6.1 Interface with ADF system

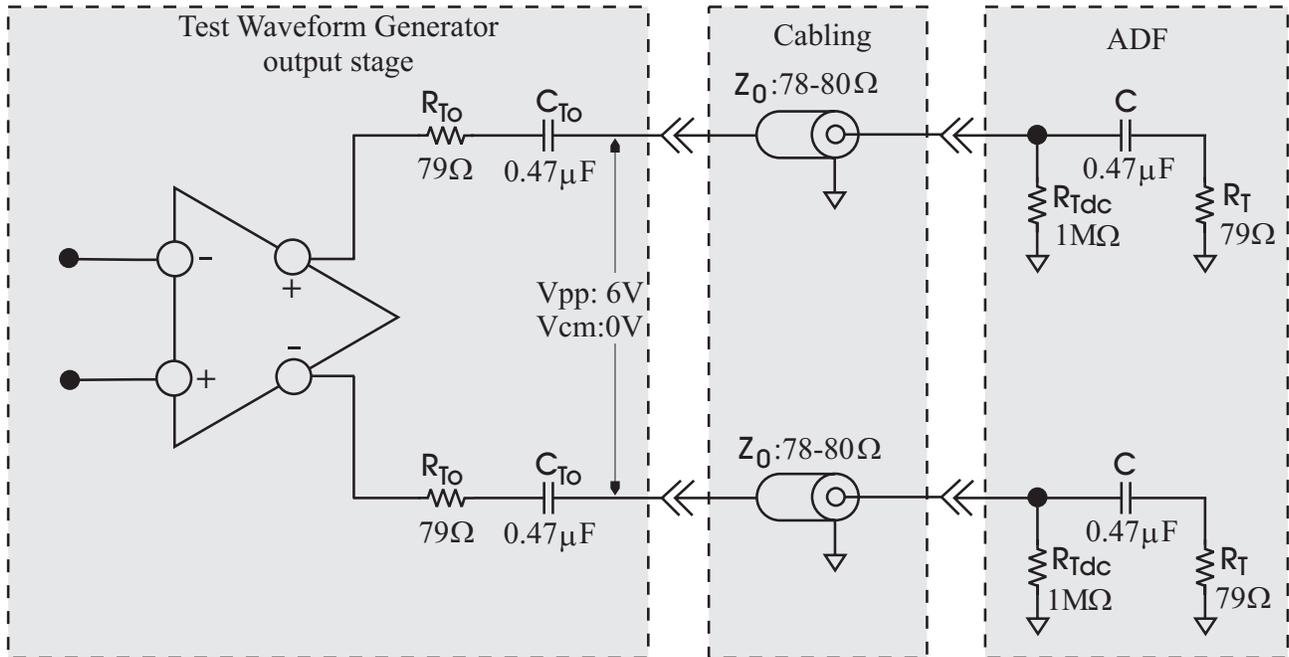


Figure 6.2, Electrical diagram of the interface with ADF system

7 Custom power supply card

The TWG prototype system requires the use of an external dual ± 10 Volt DC power supply. Internally power regulation and distribution is achieved with the use of a custom card.

A set of two negative and five positive regulators are used to generate the TWG prototype system voltages.

8 Serial Command Link Receiver (SCLR) card

The SCLR receives serial trigger and timing information over a 1.062 Gbit/sec serial link. The data is de-serialized to 20 bit parallel by the AMCC chip and handed to the Altera CPLD. The CPLD performs 10B/8B decoding and further de-multiplexes the data providing at its outputs 75 data bits at a 7.59MHz rate. Data and status bits are exchanged with a host board over two 64-pin high density connectors. The signal mapping for these connectors is provided in Table 8.1 and Table 8.2.

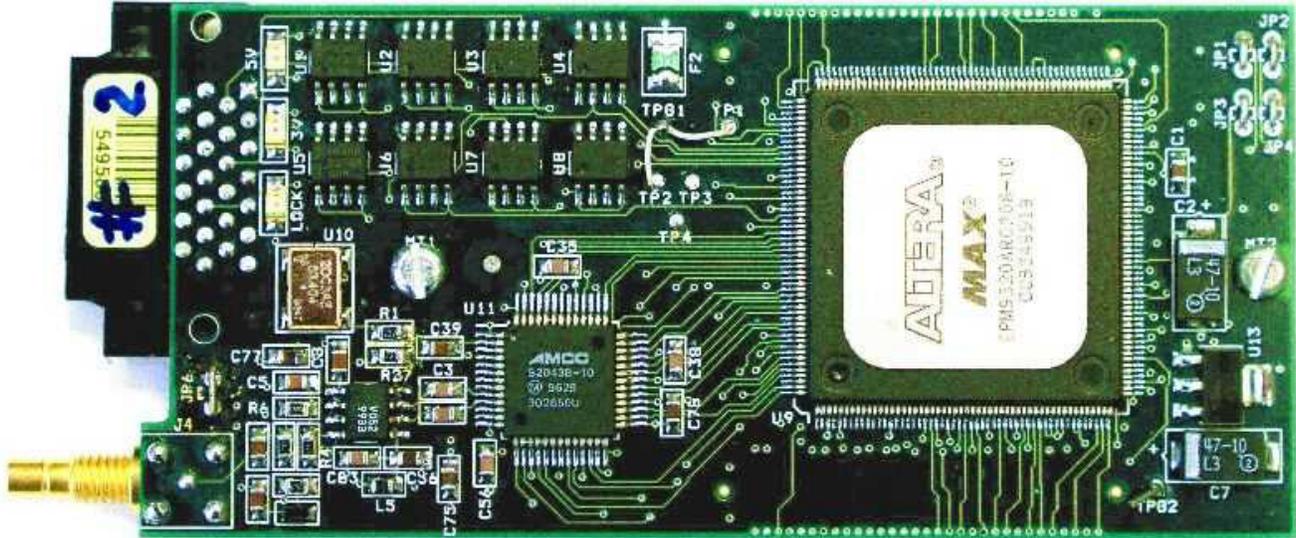


Figure 8.1, SCL Receiver card

Signal	Type	Pin #		Pin #	Type	Signal
+5V	Power	1		2	Power	+5V
SCL_READY	Output	3		4	Output	SCL_SYNCERROR
Ground	Ground	5		6	Output	CURRENT_TURN[0]
CURRENT_TURN[1]	Output	7		8	Output	CURRENT_TURN[2]
CURRENT_TURN[3]	Output	9		10	Ground	Ground
CURRENT_TURN[4]	Output	11		12	Output	CURRENT_TURN[5]
Ground	Ground	13		14	Output	CURRENT_TURN[6]
CURRENT_TURN[7]	Output	15		16	Output	CURRENT_TURN[8]
CURRENT_TURN[9]	Output	17		18	Ground	Ground
CURRENT_TURN[10]	Output	19		20	Output	CURRENT_TURN[11]
Ground	Ground	21		22	Output	CURRENT_TURN[12]
CURRENT_TURN[13]	Output	23		24	Output	CURRENT_TURN[14]
CURRENT_TURN[15]	Output	25		26	Ground	Ground
CURRENT_BX[0]	Output	27		28	Output	CURRENT_BX[1]
Ground	Ground	29		30	Output	CURRENT_BX[2]
CURRENT_BX[3]	Output	31		32	Output	CURRENT_BX[4]
CURRENT_BX[5]	Output	33		34	Ground	Ground
CURRENT_BX[6]	Output	35		36	Output	CURRENT_BX[7]
Ground	Ground	37		38	Output	FIRST_PERIOD
BEAM_PERIOD	Output	39		40	Output	L1_PERIOD
L1_ACCEPT	Output	41		42	Ground	Ground
SPARE_PERIOD	Output	43		44	Output	L2_PERIOD
Ground	Ground	45		46	Output	L2_REJECT
L2_ACCEPT	Output	47		48	Output	SCL_DATA_ERROR
RESERVED	Output	49		50	Ground	Ground
L1_BUSY	Input	51		52	Input	L1_ERROR
Ground	Ground	53		54	Input	L2_BUSY
L2_ERROR	Input	55		56	Input	INIT_ACK
SYNC_LOST	Input	57		58	Ground	Ground
SPARE_STATUS[0]	Input	59		60	Input	SPARE_STATUS[1]
Ground	Ground	61		62	Ground	Ground
Ground	Ground	63		64	Output	CLK_53

Table 8.1, Header connector J1

Signal	Type	Pin #		Pin #	Type	Signal
+5V	Power	1		2	Power	+5V
INIT_SECTION	Output	3		4	Input	SCL_ACK
Ground	Ground	5		6	Output	L1_TURN[0]
L1_TURN[1]	Output	7		8	Output	L1_TURN[2]
L1_TURN[3]	Output	9		10	Ground	Ground
L1_TURN[4]	Output	11		12	Output	L1_TURN[5]
Ground	Ground	13		14	Output	L1_TURN[6]
L1_TURN[7]	Output	15		16	Output	L1_TURN[8]
L1_TURN[9]	Output	17		18	Ground	Ground
L1_TURN[10]	Output	19		20	Output	L1_TURN[11]
Ground	Ground	21		22	Output	L1_TURN[12]
L1_TURN[13]	Output	23		24	Output	L1_TURN[14]
L1_TURN[15]	Output	25		26	Ground	Ground
L1_BX[0]	Output	27		28	Output	L1_BX[1]
Ground	Ground	29		30	Output	L1_BX[2]
L1_BX[3]	Output	31		32	Output	L1_BX[4]
L1_BX[5]	Output	33		34	Ground	Ground
L1_BX[6]	Output	35		36	Output	L1_BX[7]
Ground	Ground	37		38	Output	L1_QUALFIER[0]
L1_QUALFIER[1]	Output	39		40	Output	L1_QUALFIER[2]
L1_QUALFIER[3]	Output	41		42	Ground	Ground
L1_QUALFIER[4]	Output	43		44	Output	L1_QUALFIER[5]
Ground	Ground	45		46	Output	L1_QUALFIER[6]
L1_QUALFIER[7]	Output	47		48	Output	L1_QUALFIER[8]
L1_QUALFIER[9]	Output	49		50	Ground	Ground
L1_QUALFIER[10]	Output	51		52	Output	L1_QUALFIER[11]
Ground	Ground	53		54	Output	L1_QUALFIER[12]
L1_QUALFIER[13]	Output	55		56	Output	L1_QUALFIER[14]
L1_QUALFIER[15]	Output	57		58	Ground	Ground
SYNC_GAP	Output	59		60	Output	COSMIC_GAP
Ground	Ground	61		62	Ground	Ground
Ground	Ground	63		64	Output	CLK_7

Table 8.2, Header connector J2

9 SCL Receiver to Xilinx Test card passive adapter

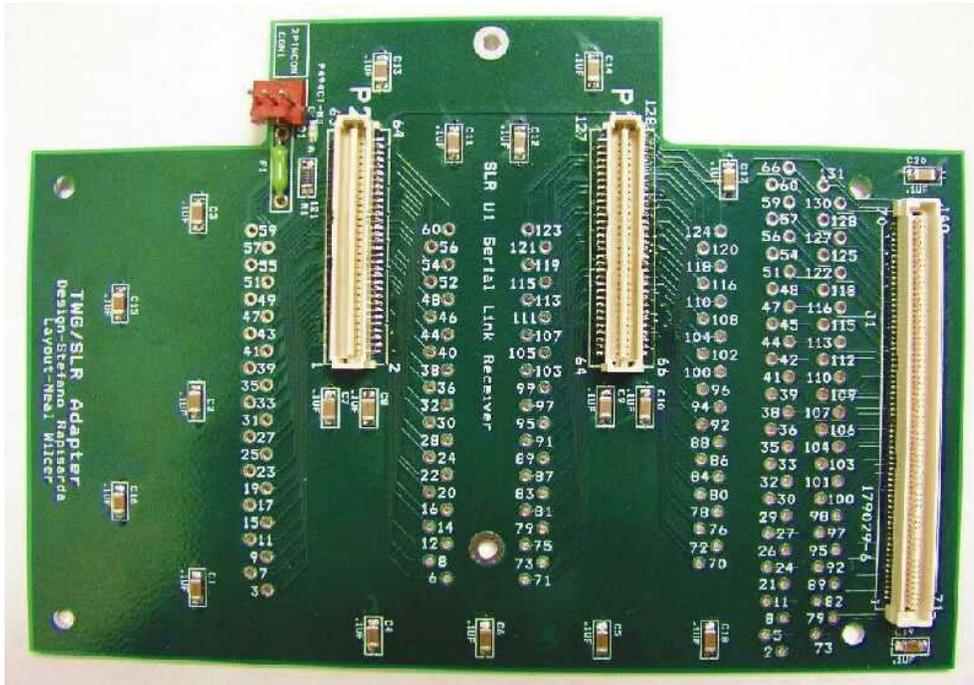


Figure 9.1, Passive adapter for SCL Receiver and Xilinx Test card

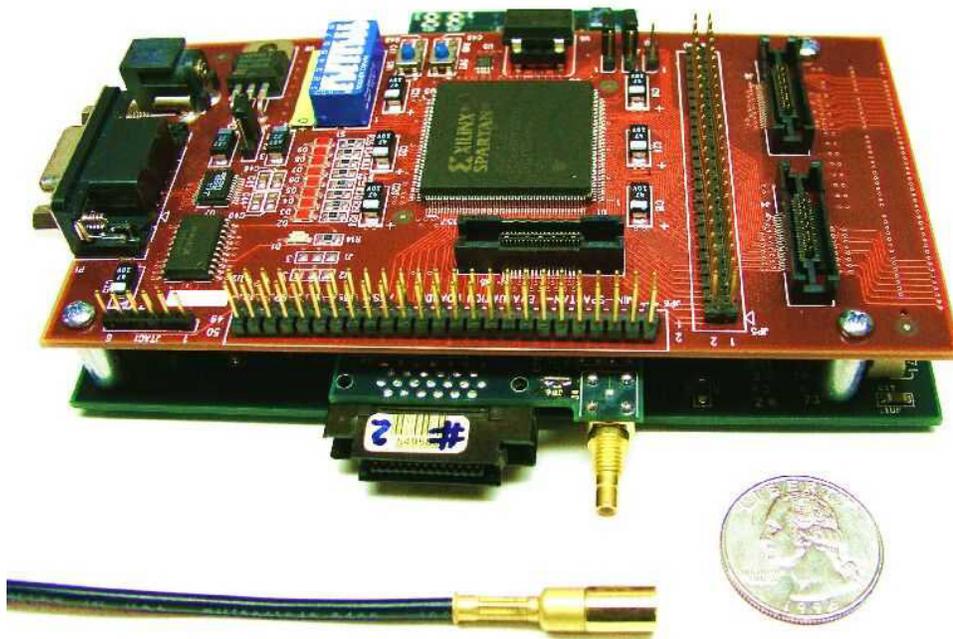


Figure 9.2, SCL Adapter, Xilinx Test card and SCL passive adapter assembly

SCL Receiver card (connectors J1 and J2)			Xilinx Test Card (connector P2-A/B)	
Signal	Type	pin#	pin#	Signal
			2	ADDRESS(1)
			73	ADDRESS(3)
			5	ADDRESS(5)
			8	ADDRESS(9)
			79	ADDRESS(11)
			11	ADDRESS(13)
			82	ADDRESS(15)
			89	ADDRESS(24)
			21	ADDRESS(26)
			92	ADDRESS(28)
			24	ADDRESS(30)
CURRENT_TURN(0)		70 (J1-P6)	95	DATA(0)
CURRENT_TURN(1)		71 (J1-P7)	26	DATA(1)
CURRENT_TURN(2)		72 (J1-P8)	27	DATA(2)
CURRENT_TURN(3)		73 (J1-P9)	97	DATA(3)
CURRENT_TURN(4)		75 (J1-P11)	98	DATA(4)
CURRENT_TURN(5)		76 (J1-P12)	29	DATA(5)
CURRENT_TURN(6)		78 (J1-P14)	30	DATA(6)
CURRENT_TURN(7)		79 (J1-P15)	100	DATA(7)
CURRENT_TURN(8)		80 (J1-P16)	101	DATA(8)
CURRENT_TURN(9)		81 (J1-P17)	32	DATA(9)
CURRENT_TURN(10)		83 (J1-P19)	33	DATA(10)
CURRENT_TURN(11)		84 (J1-P20)	103	DATA(11)
CURRENT_TURN(12)		86 (J1-P22)	104	DATA(12)
CURRENT_TURN(13)		87 (J1-P23)	35	DATA(13)
CURRENT_TURN(14)		88 (J1-P24)	36	DATA(14)
CURRENT_TURN(15)		89 (J1-P25)	106	DATA(15)
CURRENT_BX(0)		91 (J1-P27)	107	DATA(16)
CURRENT_BX(1)		92 (J1-P28)	38	DATA(17)
CURRENT_BX(2)		94 (J1-P30)	39	DATA(18)
CURRENT_BX(3)		95 (J1-P31)	109	DATA(19)
CURRENT_BX(4)		96 (J1-P32)	110	DATA(20)
CURRENT_BX(5)		97 (J1-P33)	41	DATA(21)
CURRENT_BX(6)		99 (J1-P35)	42	DATA(22)
CURRENT_BX(7)		100 (J1-P36)	112	DATA(23)
FIRST_PERIOD		102 (J1-P38)	113	DATA(24)
BEAM_PERIOD		103 (J1-P39)	44	DATA(25)
SYNC_GAP		59 (J2-P59)	45	DATA(26)
COSMIC_GAP		60 (J2-P60)	115	DATA(27)
SPARE_PERIOD		107 (J1-P43)	116	DATA(28)
			47	DATA(29)
			48	DATA(30)
			118	DATA(31)
			51	CNTL(2)
			122	CNTL(4)
			54	CNTL(6)
			125	CNTL(8)
			56	CNTL(9)
			57	CNTL(10)
			127	CNTL(11)
			128	CNTL(12)

			59	CNTL(13)
			60	CNTL(14)
			130	CNTL(15)
			131	CNTL(16)
SCL_READY		67 (J1-P3)	62	CNTL(17) hardwired
SCL_SYNC_ERROR		68 (J1-P4)	63	CNTL(18) hardwired
SCL_DATA_ERROR		112 (J1-P48)	133	CNTL(19) hardwired
SCL_ACK		4 (J2-P4)	134	CNTL(20) hardwired
CLK_53		128 (J1-P64)	65	CLK_IN hardwired
CLK_7		64 (J2-P64)	136	CLK_OUT hardwired

Table 9.1, SCL Receiver to Xilinx Test card adapter connections map

10 Appendix A - Description of MS Excel Spreadsheet Interface

The TGW features can be easily accessed and controlled through an interface based on Microsoft Excel. An excel spreadsheet allows the user to access the TWG prototype memory and registers using the serial port interface.

Waveform worksheet

High-level interface that provide the user a simple way of controlling the waveform generated by the TWG.

Triggers worksheet

High-level interface to control the triggers used by the TWG.

Manual Serial Test worksheet

Low-level interface employed to test serial interface protocol used by the Xilinx FPGA and the MS Excel spreadsheet. The user has access to all the bytes sent and received.

Serial Loop Test worksheet

Medium level interface used to test the reliability of the serial interface between the computer and the Xilinx FPGA. The protocol is hidden and the user has access to the 2-bytes data words sent and received. The test use the data echoed by the Xilinx FPGA serial interface and compare it with the data sent.

Memory Loop Test worksheet

Medium level interface used to test the reliability of the serial interface between the computer and the Xilinx FPGA. The protocol is hidden and the user has access to the 2-bytes data words sent and received. The test consists in writing data to the memory on the Xilinx FPGA and read it back to verify it.

Read Write worksheet

Medium level interface used to access in read/write the memory on the Xilinx FPGA.

Allows the user to fully control the TWG prototype functions specifying content of both board/channel registers and the content of the channel's memory used to drive the DACs.

Serial Port Settings worksheet

Used to specify the computer serial port setting. The settings should match those of the Xilinx FPGA serial port interface.

11 Appendix B - June 11th 2003 Measurements

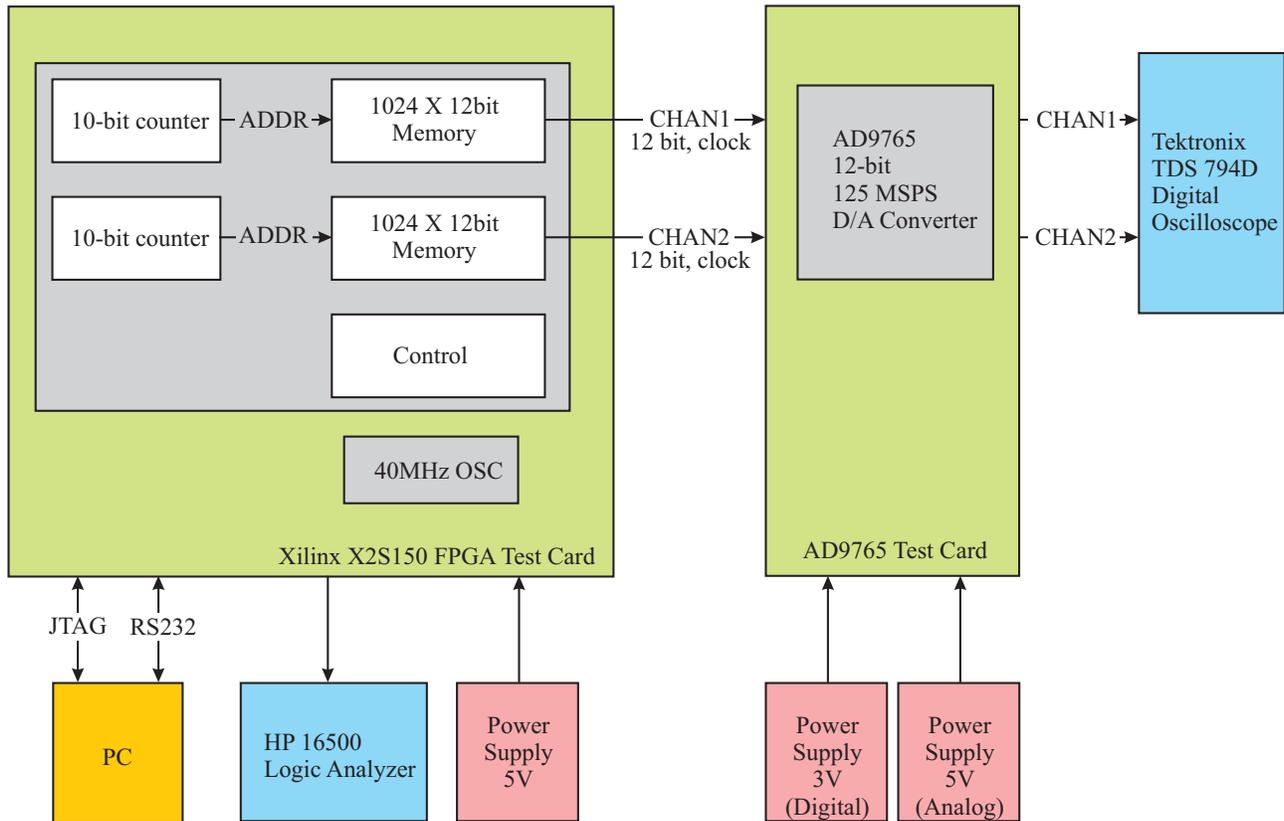
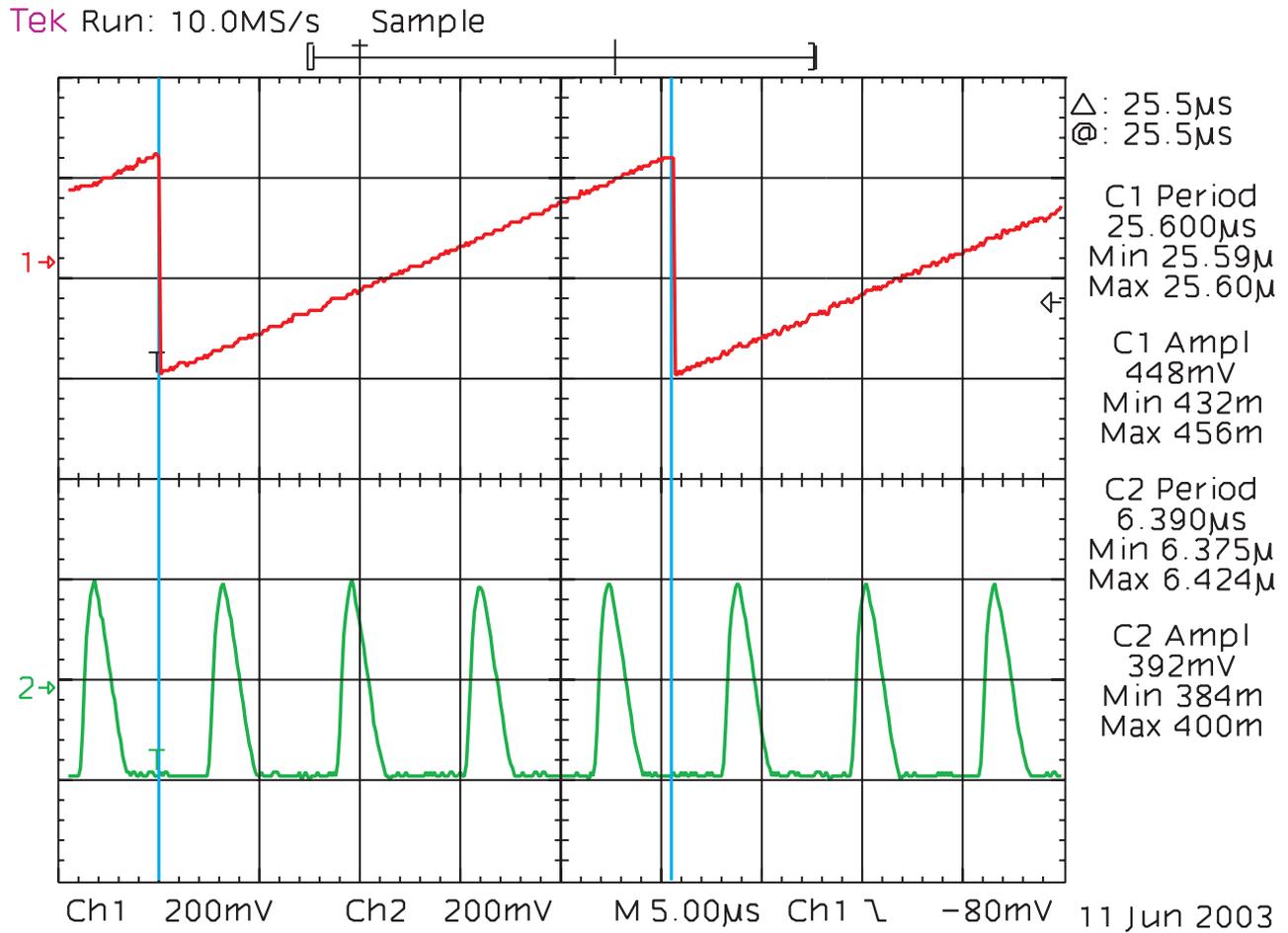


Figure 11.1, block diagram of the June 11th 2003 Test System



Channel 1: DAC CH1 Output.

DAC CH1 input driven with repetition of sequence of 1024 12bits samples (ramp).

2 Least Significant bits not used.

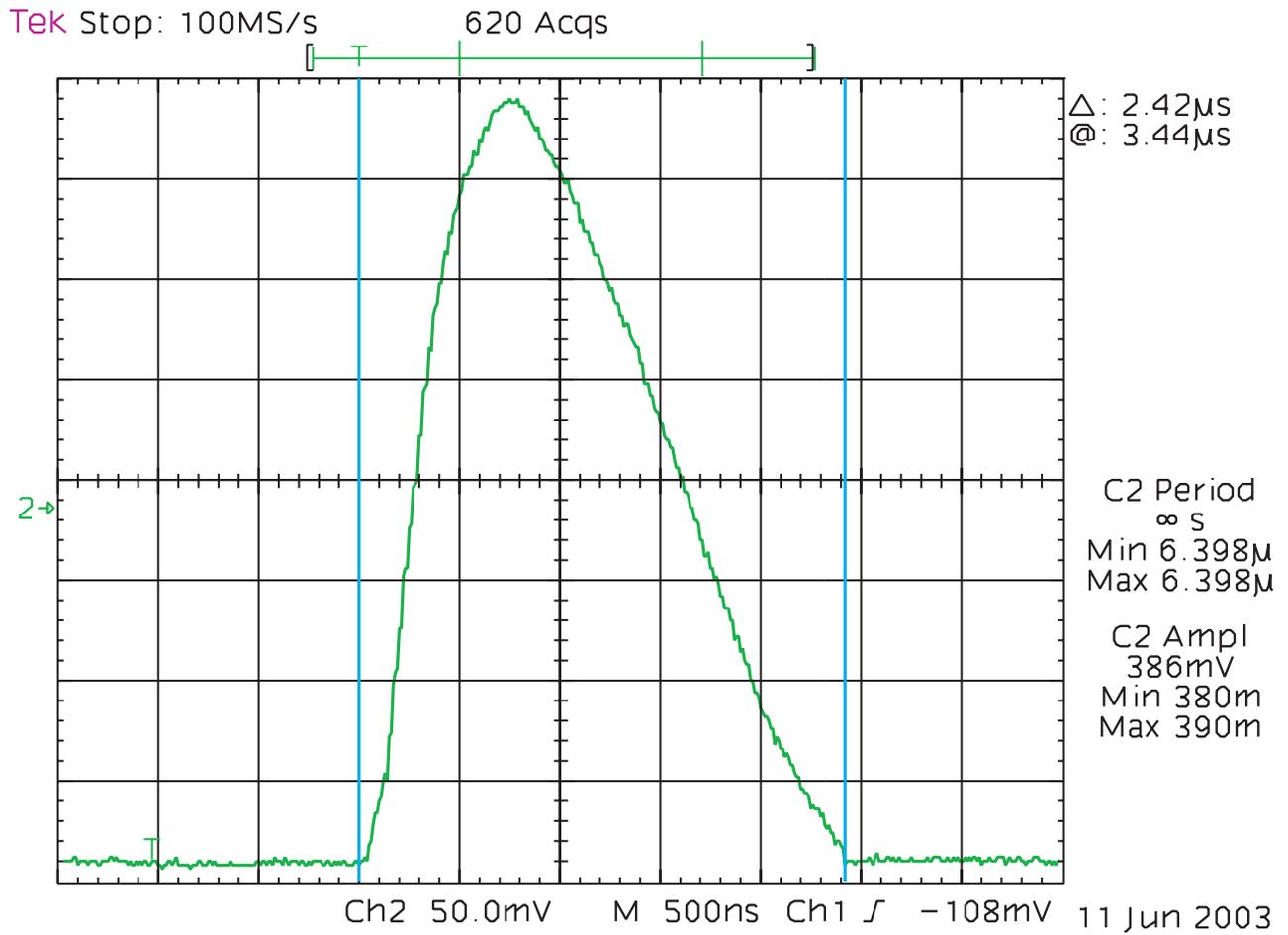
DAC CH1 Clock: 40 MHz.

Channel 2: DAC CH2 Output.

DAC CH2 input driven with repetition of sequence of 1024 12bits samples (4 pulses).

DAC CH2 Clock: 40 MHz.

Figure 11.2, Measure 1.

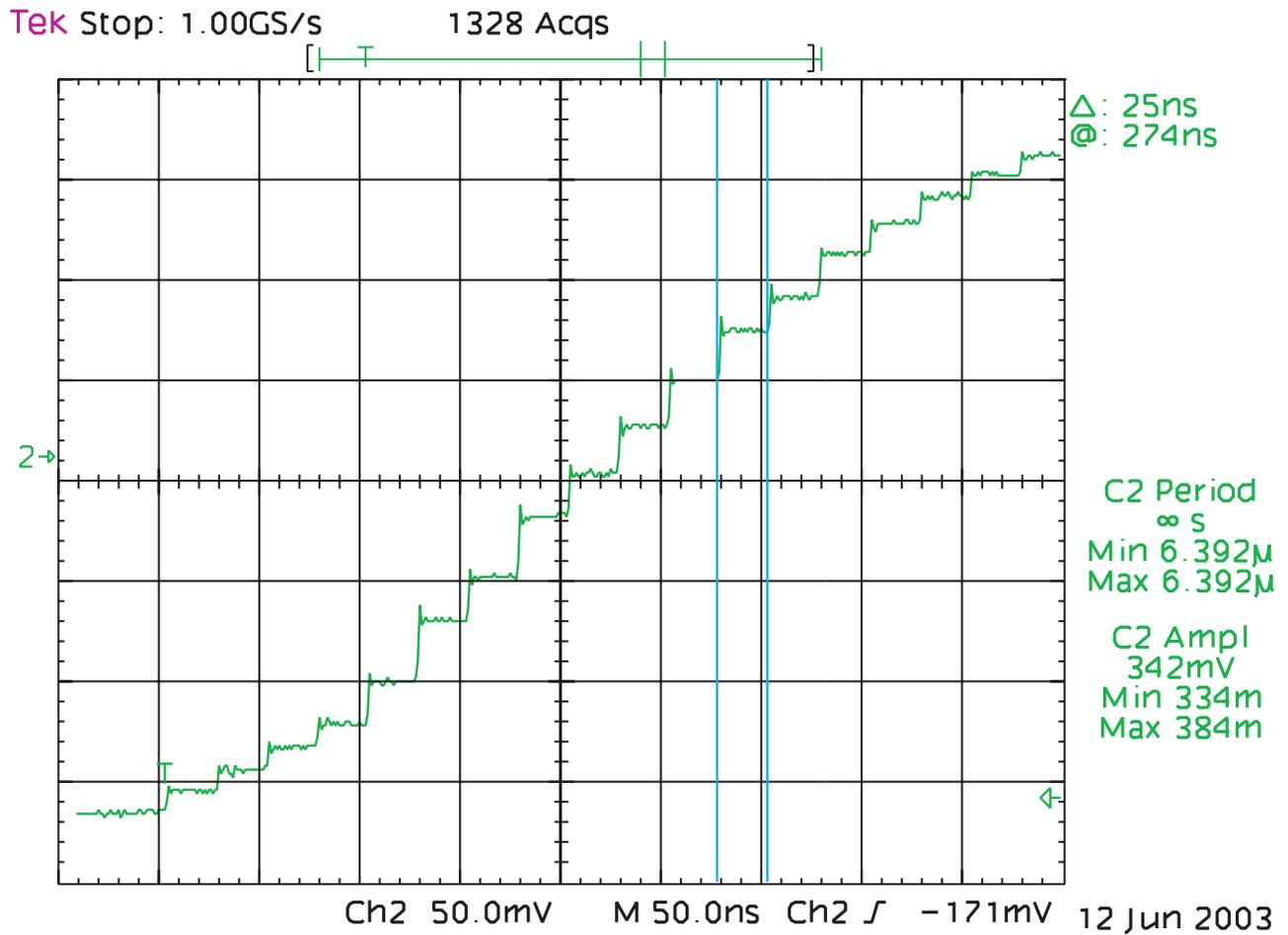


Channel 2: DAC CH2 Output.

DAC CH2 input driven with repetition of sequence of 1024 12bits samples (4 pulses).

DAC CH2 Clock: 40 MHz.

Figure 11.3, Measure 2.



Channel 2: DAC CH2 Output.

DAC CH2 input driven with repetition of sequence of 1024 12bits samples (4 pulses).

DAC CH2 Clock: 40 MHz.

Figure 11.4, Measure 3.

12 Glossary

ADF: Analog to Digital converter and Filter board. Component of the Run IIb D0 Level 1 Calorimeter System. See system diagram ().

BLS: Summer and Baseline Subtractor card. BLSs are a component of the Run IIb D0 Level 1 Calorimeter System. See system diagram ().

TAB: Trigger Algorithm Board. TABs are a component of the Run IIb D0 Level 1 Calorimeter System. A TAB run the physics selection algorithms on one eighth of the data produced by the ADF (Analog to Digital converter and Filter) boards. See system diagram ().

GAB: Global Algorithm Board. GABs are a component of the Run IIb D0 Level 1 Calorimeter System. See system diagram ().

SCL: Serial Command Link.

BSDL: Boundary Scan Description Language. IEEE 1149.1-1993b defines a language that describes IEEE 1149.1 architecture for an integrated circuit. This language is known as the Boundary Scan Description Language (BSDL). Updated BSDL files for the XILINX devices used on the Mixer Board can be found at:

http://support.xilinx.com/support/sw_bsd.htm

EDIF: Electronic Data Interchange Format. Industry-standard for specifying a logic design in text (ASCII) form.

File types:

JEDEC files

JEDEC files are CPLD programming files generated by the CPLD fitter. They are ASCII text files containing programming information and, optionally, functional test vectors that can be used to verify the correct functional behavior of the programmed device. One JEDEC file is required for each device in the JTAG programming chain. The extension for JEDEC files is .jed.

BSDL Summary files

The Boundary-Scan Description Language (BSDL) files use a subset of VHDL to describe the boundary scan features of a device. The Xilinx JTAG Programmer automatically extracts the length of the instruction register from the BSDL file to place non-Xilinx devices in bypass mode. Xilinx BSDL files are located automatically by the JTAG Programmer. The name of the BSDL file is assumed to be <device name>.bsd.

BIT Files

Bit files are Xilinx FPGA configuration files generated by the Xilinx FPGA design software. They are proprietary format binary files containing configuration information. One BIT file is required for each Xilinx FPGA in the JTAG boundary-scan chain. The extension for BIT files is ".bit".

(MCS/EXO) PROM Files

PROM programming files are generated by the Xilinx PROM file formatter. They are ascii text files used to specify configuration data. One PROM file is required for each Xilinx PROM in the JTAG boundary-scan chain. Use the device properties (Edit>Properties) dialog to specify the location of the MCS/EXO files for each Xilinx PROM. The required extensions for MCS and EXO files are ".mcs" and ".exo" respectively.

FPGA: Field Programmable Gate Array. An integrated circuit that contains configurable (programmable) logic blocks and configurable (programmable) interconnect between these blocks.

IEEE: Institute of Electrical and Electronics Engineers, Inc. More information is available on the Internet:

<http://www.ieee.org/>

Jedec: The JEDEC Solid State Technology Association (once known as the Joint Electron Device Engineering Council), is the semiconductor engineering standardization body of the Electronic Industries Alliance (EIA), a trade association that represents all areas of the electronics industry. More information is available on the Internet:

<http://www.jedec.org/>

Jitter: The JEDEC Standard No. 65 (EIA/JESD65) defines jitter as the magnitude of the time deviation of a controlled edge from its nominal position.

JTAG: Joint Test Action Group. Older name for IEEE 1149.1 boundary scan, a method to test printed circuit boards and also integrated circuits. See also BSDL. Design complexity, difficulty of loaded board testing, and the limited pin access of surface mount technology led industry leaders to seek accord on a standard to support the solution of these problems.

The standard defines a hardware architecture and the mechanisms for its use.

The standard itself defines instructions that can be used to perform functional and interconnect tests as well as built-in self test procedures. Vendor-specific extensions to the standard have been developed to allow execution of maintenance and diagnostic applications as well as programming algorithms for re-configurable parts.

The top level schematic of the test logic defined by IEEE Std 1149.1 includes three key blocks:

The *TAP Controller*

This responds to the control sequences supplied through the test access port (TAP) and generates the clock and control signals required for correct operation of the other circuit blocks.

The *Instruction Register*

This shift register-based circuit is serially loaded with the instruction that selects an operation to be performed.

The *Data Registers*

These are a bank of shift register based circuits. The stimuli required by an operation are serially loaded into the data registers selected by the current instruction. Following execution of the operation, results can be shifted out for examination.

The *JTAG Test Access Port* (TAP) contains four pins that drive the circuit blocks and control the operations specified. The TAP facilitates the serial loading and unloading of instructions and data. The four pins of the TAP are: TMS (Test Mode Select), TCK (Test Clock), TDI (Test Data In) and TDO (Test Data Out). The function of each TAP pin is as follows:

TCK - this pin is the JTAG test clock. It sequences the TAP controller as well as all of the JTAG registers.

TMS - this pin is the mode input signal to the TAP Controller. The TAP controller is a FSM that provides the control logic for JTAG. The state of TMS at the rising edge of TCK determines the sequence of states for the TAP controller. TMS has an internal pull-up resistor on it to provide a logic 1 to the system if the pin is not driven.

TDI - this pin is the serial data input to all JTAG instruction and data registers. The state of the TAP controller as well as the particular instruction held in the instruction register determines which register is fed by TDI for a specific operation. TDI has an internal pull-up resistor on it to provide a logic 1 to the system if the pin is not driven. TDI is sampled into the JTAG registers on the rising edge of TCK.

TDO - this pin is the serial data output for all JTAG instruction and data registers. The state of the TAP controller as well as the particular instruction held in the instruction register determines which register feeds TDO for a specific operation. Only one register (instruction or data) is allowed to be the active connection between TDI and TDO for any given operation. TDO changes state on the falling edge of TCK and is only active during the shifting of data through the device. This pin is three-stated at all other times.

LVTTL: Low Voltage TTL. Is one of the several switching standards used in digital electronics.

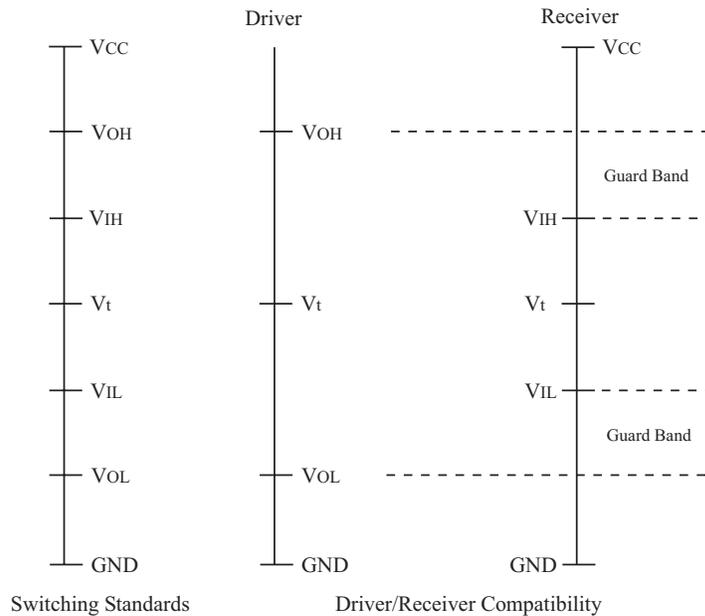


Figure 12.1, Switching standards.

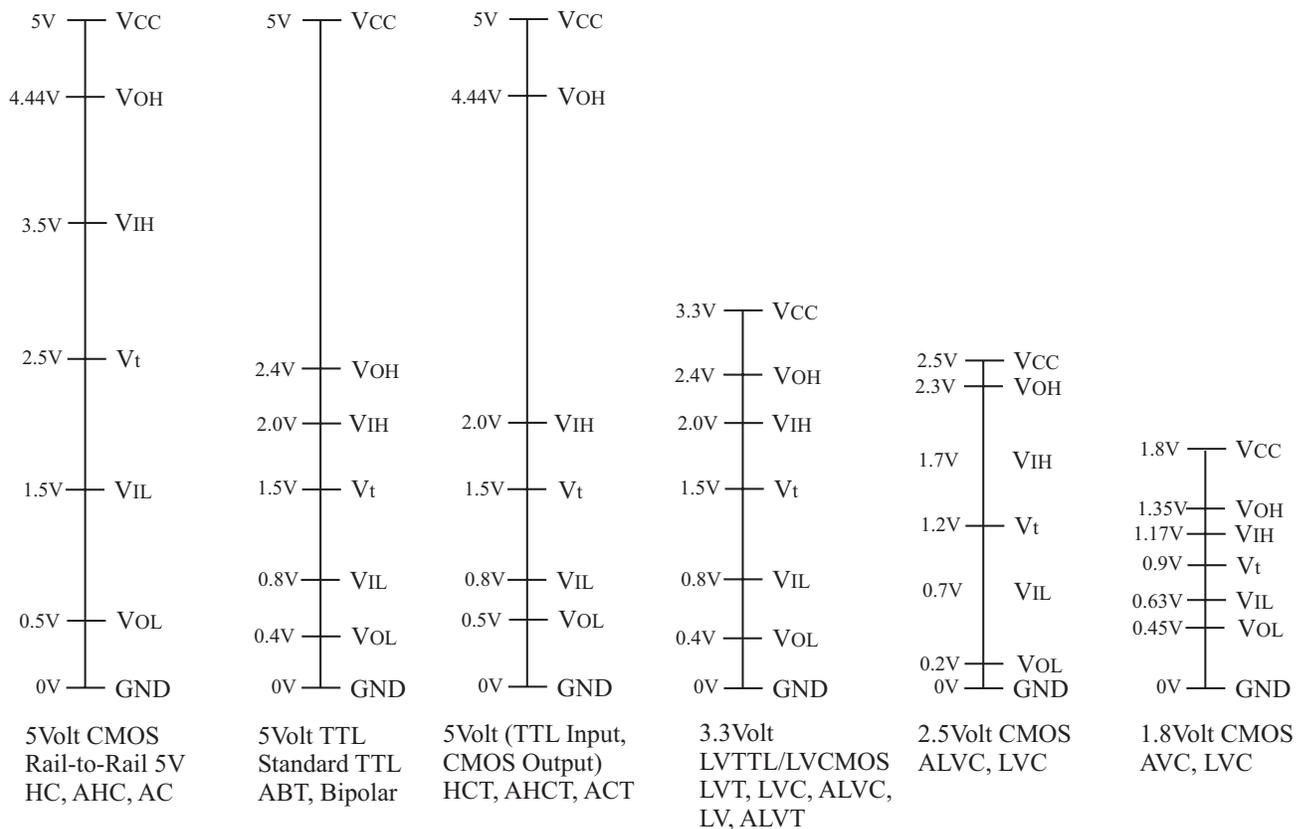


Figure 12.2, Comparison of switching standards.

MCU: Micro-Controller Unit.

Skew: The JEDEC Standard No. 65 (EIA/JESD65) defines skew as The magnitude of the time difference between two events that ideally would occur simultaneously.

VHDL: "V" stands for Very High Speed Integrated Circuit and "HDL" stands for Hardware Description Language.

13 References

- [1] Paul Horowitz, Winfield Hill, "The Art of Electronics", 1995, Cambridge University Press. ISBN 0-521-37095-7.
 - [2] Howard Johnson, Martin Graham, "High Speed Digital Design", 1993, Prentice Hall PTR. ISBN 0-13-395724-1.
 - [3] Test Waveform Generator System documentation
http://www-ese.fnal.gov/D0Cal_TWG/
 - [4] D0 Calorimeter RunIIb Level 1 Trigger documentation:
<http://www.nevis.columbia.edu/~evans/l1cal/index.html>
 - [5] Agilent (formerly Hewlett-Packard) documentation:
 - a) Probing Solutions for Agilent Technologies Logic Analysis Systems. Information available on Internet:
<http://www.agilent.com/>
 - [6] Aldec. Web site: <http://www.aldec.com/>.
 - [7] Analog Devices. Web site: <http://www.analog.com/>.
 - [8] Corel Draw is a drawing software package by Corel. Web site: <http://www.corel.com>
 - [9] Texas Instruments. Web site: <http://www.ti.com/>
 - [10] Tektronix. Web site: <http://www.tek.com/>
 - [11] Xilinx documentation. Information available on Internet:
<http://www.xilinx.com/>
 - [12] Avnet Inc. Web site:
<http://www.avnet.com/>
 - [13] Feedback is very welcome.
- A small ASCII art drawing of a person with a speech bubble. The speech bubble contains the text: "It's not a BUG it's a FEATURE!"
- [14] Serial Command Link Receiver (SCLR), June 6, 2000. Document available on the web:
<http://www-ese.fnal.gov/>
 - [15] Serial Command Link Description, D-Zero Run II Trigger DAQ System, April 24th, 1997.