Control-X. User Manual

C. F. Bedoya, A. Navarro, C. Willmott.

mayo 2013

CMS

Versión 1.0

Índice

1. Introduction

Control-X is a VME multipurpose board designed to operate Readout Boards (ROB's) under all possible circumstances.

It includes:

- Trigger synchronization, delay, counting, generation through VME, and interfaces with several common standards.
- Two test pulse handling with independent delays, and pulse generation via VME.
- Interrupt generation: externally from two different sources, or internally at the end of a programmable trigger count.
- Full ROBUS operation.

1.1 40 MHz Clock

The clock to operate the board can be selected through a jumper (see Fig. x) from an on boards quartz, and external NIM input, or an external TTL input. The selected clock is output at the front panel: LVDS OUT CLK0..7 and LVPECL OUT CLK.

1.2 Trigger handling

All trigger sources are merged into one common internal trigger signal. This signal is output as LVDS OUT T0 in the front panel. Then it is synchronized with the 40 MHz clock and fed into the shift delay line. The selected outputs of the shift delay line (Reg. $02) are OR-ed and send to the various trigger outputs (NIM OUT TRG, LVPECL OUT TRG, LVDS

OUT TRGSY, and ROBUS trigger line) only when:

a) trigger count bypass is selected: bit_3 in Reg. $00 set to 0 or
b) trigger count bypass is de-selected (previous bit ser to 1) and trigger count has not reached the programmed trigger max count (Reg. $04).

1.3 Trigger counting

When the trigger counting option is selected by setting bit_3 in Reg. $00 to 1 and a programmed trigger count is written into Reg. $04, this number of triggers is sent to the various trigger outputs. After that, trigger output is stopped and subsequent arriving input triggers are counted in Reg. $0E + $0C (Trigger_Count_Out). A new sequence can be restarted by writing a 1 in bit_8 Reg. $1E. At that moment Trigger_Count_Out register is reset and its contents is transferred to Trigger_Count_Out_Latch ($0A + $08).

A high level on input IN VETO also stops triggers being send to output connectors and initiates Trigger_Count_Out mechanism.

1.4 Test pulses

Test pulses can be generated through VME (bit_4 and bit_5 in Reg $1E) or via front panel connectors: TTLIN PLS0, TTLIN PLS1. After a programmable delay (Reg $16 and Reg $18) pulses are output on front panel connectors LVDS OUT PLS0 and LVDS OUT PLS1.

1.5 Interrupts

Interrupts can be generated via two independent front panel input connectors, NIM IN INT0 and NIM IN INT1, or at the end of Trigger_Count_In. All three interrupts have independent registers ($10, $12 and $14) to select interrupt levels and vectors. In addition Interupt_0 and Interupt_1 can be programmed to accept:

a) Polarity: active low or active high (bit_11)
b) Pulse: level or edge (bit_12)
c) Source (input Lemo connector): LVDS OUT PLS0 and LVDS OUT PLS1.

1.6 ROBUS

A 40-pin Yamaishi connector interconnects to all ROBUS signals. More specifically:

a) Reg $1A: 7 ROB ON signals
b) Reg $1C: JTAG interface
c) Reg $1E, in addition to trigger signal it provides: Bunch_Counter_Reset, Reset, Event_Reset, Test_Mode_Advance, and Test_Mode_Reset. The signal Test_Mode is implemented in bit_4 Reg $00.
d) Reg $20, $22, $24, and $26: TDIO bus interface

2. Control-X VME Interface

The VME interface to the Control-X board is performed through standard A16 access 16 bit words. The address selection is done through switch S3.

2.1 Register 0 ($00) G0.0: General Control & Status

0	enable external interrupt 0	R/W	When an interruption is sent, the corresponding enable signal is put to 0 until it is set again through VME. For end_count, it only controls the interrupt request, not the normal function of the trigger counting mode.
1	enable external interrupt 1	R/W	
2	enable end_count (interrupt 2)	R/W	
3	trigger count bypass	R/W	0=bypass
4	enable test mode (track)	R/W	➔ ROBUS Test Pulse
5	ROB power fault latch	R/W	1=fault
6	ROB power fault	R	← ROBUS FLTB (active low)
7	ROB error	R	← ROBUS ROBerror (active high)
8	interrupt requested 0	R	Not latched. Thus, these bits may be down even if have provoked an interrupt request.
9	interrupt requested 1	R	
10	trigger end count	R	Not latched. However, once the irq is sent this bit remains active until reset by restarting the trigger count.

11	board global reset	W	Causes the FPGA registers to initialize (to reload firmware from PROM, send a sysreset VME signal)

2.2 Register 1 ($02) G0.1: Trigger delay

Default 0x8 (1.6 us)

0-15	trigger shift delay (200ns step; max 51.2 us)	R/W	

2.3 Register 2 ($04) G0.2: Trigger max count

0-14	trigger max count	R/W	Max number of triggers allowed when trg count bypass is deselected
15	trigger end-count interrupt[1]	R/W	If set, an end count interrupt will be generated continuously once the count is restarted with register 15 bit8

2.4 Register 3 ($06) G0.3: Trigger count in

0-15	trigger count in	R	

2.5 Register 4 ($08) G1.0: Trigger count out latch 0

0-15	trigger count out latch 0-15	R	

2.6 Register 5 ($0A) G1.1: Trigger count out latch 1

0-15	trigger count out latch 16-31	R	

2.7 Register 6 ($0C) G1.2: Trigger count out register 0

0-15	trigger count out register 0-15	R/W	

2.8 Register 7 ($0E) G1.3: Trigger count out register 1

0-15	trigger count out register 16-31	R/W	

1 Internally, the max count register is a countdown register that is loaded with the value of this register and generates an interruption when reaches negative values (MSB = 1). Thus, if the max count has already set the MSB, it launches an interruption as soon as it is restarted, and continues to request it continuously.

2.9 Register 8 ($10) G2.0: Interrupt 0

0-7	Interrupt Vector_0	R/W	
8-10	Interrup Level_0 (1-7)	R/W	
11	Polarity_0	R/W	
12	Pulse_0	R/W	
13	Select Lemo0/Lemo1	R/W	

2.10 Register 9 ($12) G2.1: Interrupt 1

0-7	Interrupt Vector_1	R/W	
8-10	Interrup Level_1 (1-7)	R/W	
11	Polarity_1	R/W	
12	Pulse_1	R/W	
13	Select Lemo1/Lemo0	R/W	

2.11 Register 10 ($14) G2.2: Interrupt 2 (end_count)

0-7	Interrupt Vector_2	R/W	
8-10	Interrup Level_2 (1-7)	R/W	

2.12 Register 11 ($16) G2.3: Pulse 0 delay

0-6	Delay 0.5ns step (max 64 ns)	R/W	
7-13	Delay 25 ns step (max 3200 ns)	R/W	

2.13 Register 12 ($18) G3.0: Pulse 1 Delay

0-6	Delay 0.5ns step (max 64 ns)	R/W	
7-13	Delay 25 ns step (max 3200 ns)	R/W	

2.14 Register 13 ($1A) G3.1: ROB on_off register

Default 0x7F (all ROBs on)

0	RON 0	R/W	➔ ROBUS
1	RON 1	R/W	
2	RON 2	R/W	

3	RON 3	R/W	
4	RON 4	R/W	
5	RON 5	R/W	
6	RON 6	R/W	

2.15 Register 14 ($1C) G3.2: JTAG

0	JTCK	R/W	➔ ROBUS
1	JTMS	R/W	
2	JTDI	R/W	
3	JTADD0	R/W	
4	JTADD1	R/W	
5	JTADD2	R/W	
6	JTADD3	R/W	
7	JTDO	R	<--- ROBUS

2.16 Register 15 ($1E) G3.3: ROB pulse register

(pulses of 25 ns width)

0	Bunch Counter Reset	W	
1	Trigger	W	
2	Reset	W	
3	Event Reset	W	
4	Pulse 0 (50 ns)	W	
5	Pulse 1 (50 ns)	W	
6	test mode advance	W	
7	test mode reset	W	
8	set trigger count in	W	

Writing ones in this register, bits 0 – 7 triggers the corresponding event, which can also be triggered through some of the front panel connectors.

Setting the bit 8 causes the trigger count to restart with the value in register 2 and the trigger count out to be latched and reset. After a trigger count interrupt is sent, if the corresponding interrupt enable is set in reg. 0, an interrupt will be immediately requested if the count is not restarted.

2.17 Register 16 ($20) G4.0: TDIO write

0-7	Bit to be written	R/W	

If the next operation is a bit (not byte) write, only the LSB will be used.

2.18 Register 17 ($22) G4.1: TDIO read

0-7	Bit read during last command	R	

If last command was a 1 bit read or init, bits 1-7 are low.

If last command was a byte read, all 8 bits are actve.

2.19 Register 18 ($24) G4.2: TDIO command

0	Initialization command	W	750 us
1	Write command	W	70 us
2	Read command	W	70 us
3	Byte write command	W	700 us
4	Byte read command	W	700 us

The approximate duration of each command is given plus a security margin

2.20 Register 19 ($26) G4.3: CRC

0-7	CRC register	R/W	

The 8-bit CRC calculated for the stream of bits read since the last time this register was reset.

This register is reset automatically right after the init operation and is updated with every bit read from the line.

2.21 Register 20 ($28) G5.0: Trigger control

0-3	Auto-trigger frequency	R/W	Frequency (in 10^5 Hz) – 1. Example: 300 kHz => 3 (in 10^5 Hz) – 1 => 2
4	Selector for trigger delay	R/W	0 => Delay Trigger-IN signal; 1 => Delays Auto-

			trigger signal
5-6	Selector for TRG-SY output	R/W	0 => Redirects Trigger-IN signal to output, 1 => Redirects formed Trigger-IN pulse to output, 2 => Redirects Auto-trigger signal to output, 3 => Redirects delayed signal to output (it depends on bit-4 selection)
7	T0 Output selector	R/W	0 => Trigger-IN signal; 1 => Auto-trigger signal

3. Notes about FLASH-CONTROL-X

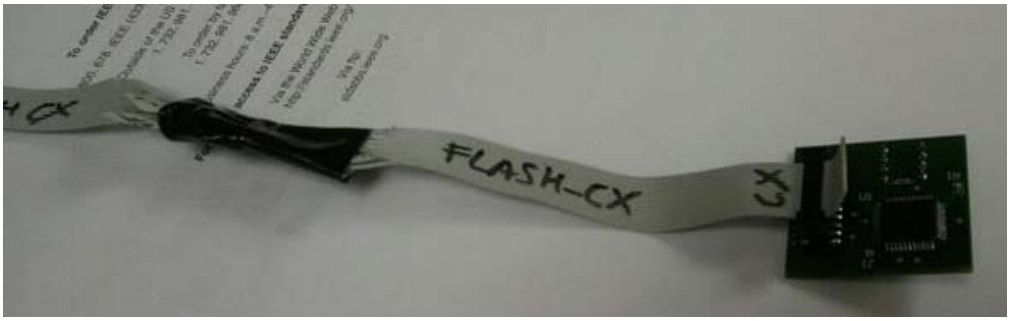
El conector de Xilinx está mal el footprint, no se puede usar el cable paralelo estándar. Hay que usar uno homemade.

Hay que crear el *.mcs para una Spartan 2: XC2S50-PQ208 -6

```
F:\CRIS\CMS\VHDL_ISE6_2010\CONTROLX_2011\ControlX_spartan2.mcs
```

Los 4 pines más cercanos al silkscreen P2 hay que cortarlos. El Bump del connector tiene que mirar hacia la Flash (hacia arriba)


CONTROL-X
XILINX
SPARTAN
DONE
PROG


FLASH-CX