CONV-TTL-BLO
User Guide

Theodor-Adrian Stana
BE-CO-HT

March 11, 2013

Abstract

This document describes the CONV-TTL-BLO board, a blocking pulse repeater board in double height VME format, intended to replace the following boards:

- 8 channel repeater
- 16 channel repeater
- CTDAC
- LA-BLO-TTL
- LAF-BLO-TTL
- LASB-TTL-BLO
- LA-GATE
- LA-TTL-BLO
- LAPF-TTL-BLO [1]

[1] These boards have a 4 μs pulse width.

Contents

List of Figures

List of Tables

List of abbreviations

FPGA	Field-Programmable Gate Array
RTM	Rear Transition Module
RTMM	RTM Motherboard
RTMP	RTM Piggyback
SFP	Small form-factor pluggable (in the context of SFP connectors)

1 Introduction

CONV-TTL-BLO is a board intended for replicating blocking and TTL pulses, offering six totally independent replication channels. The shape of the pulses is defined in Sec. 4. CONV-TTL-BLO works together with two more boards: CONV-TTL-RTM and CONV-TTL-RTM-BLO.

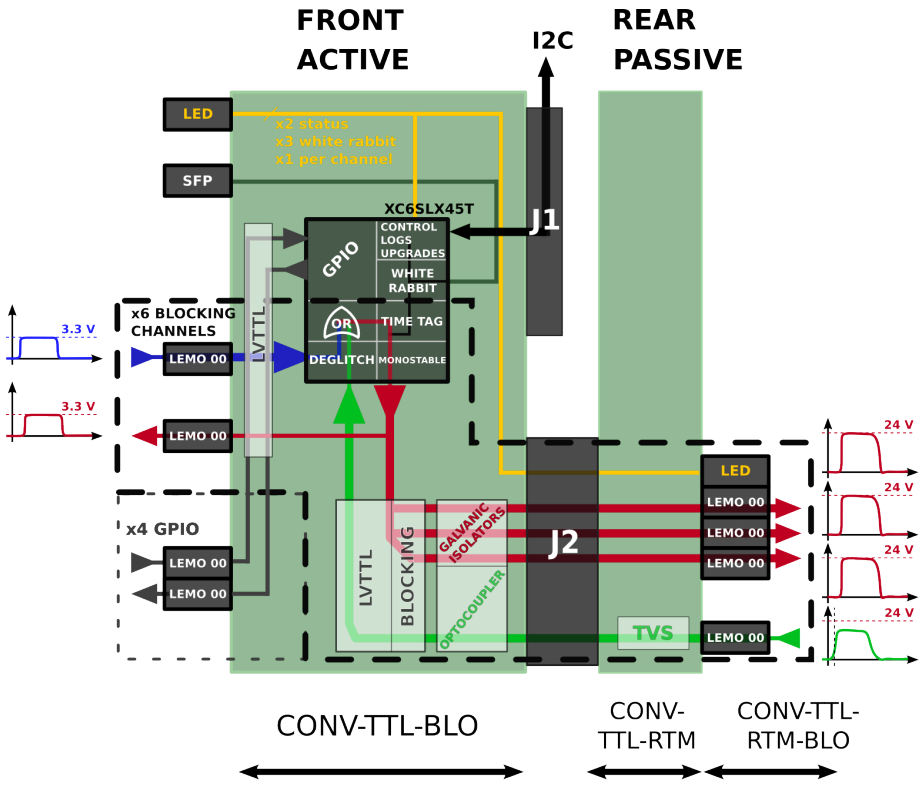

Figure 1: Pulse Repetition system

CONV-TTL-BLO contains all the active circuitry and is connected as a front module to a VME64x backplane. CONV-TTL-RTM and CONV-TTL-RTM-BLO are both connected to the rear part of the crate and provide, in the rear panel, the connectivity of the I/O blocking lines. Every channel offers, in the rear panel, three blocking pulse outputs and one blocking pulse input.

CONV-TTL-RTM is a motherboard attached to the rear transition module (RTM) of the P2 VME64x connector. It connects CONV-TTL-BLO to CONV-TTL-RTM-BLO and provides overvoltage protection for all the I/Os of all the channels.

CONV-TTL-RTM-BLO is an RTM piggyback (RTMP) board mounted

on CONV-TTL-RTM. It contains all the LEMO 00 connectors and channel LEDs that are offered in the rear panel.

Table 1: Boards for Blocking repetition

Board	**Connection**	**Front panel ports**
CONV-TTL-BLO	Front	SFP TTL Blocking triggers TTL Blocking output replica inverters
CONV-TTL-RTM	Back	-
CONV-TTL-RTM-BLO	Back	Blocking Pulse input Blocking Pulse outputs

2 Getting Started

This section provides a description on testing CONV-TTL-BLO boards for basic functionality. The following steps should be followed in order to test the board.

1. Plug in a front module card to the ELMA crate. Turn on power to the crate and program the Spartan-6 FPGA.
2. Check that the *PW* LED lights *green* and the *ERR* LED lights *red*. The *TTL_N* LED may also be lit. If it is and the LED is *green*, then the LEVEL switch is set for INV-TTL pulses.
3. Make sure the LEVEL switch is set for TTL pulses (see Sec. **??**).
4. First, connect one end of a cable with LEMO 00 connectors at both ends to the TTL input port of channel 1 on the front panel.
5. Configure a pulse generator to output TTL level pulses (*max. 5V*) at a frequency of about 1 Hz with a pulse length of approx. 1 μs and connect the other end of the cable to the pulse generator. The LED of the corresponding channel should light for 96 ms when a pulse arrives. A TTL pulse should be replicated at the TTL output of channel 1. Check (using e.g., an oscilloscope) that the TTL pulse has a width of 1 μs and a 3.3 V amplitude.
6. Connect a CONV-TTL-RTM board (with attached CONV-TTL-BLO-RTM) to the back-plane of the ELMA, on the same VME slot as the front module. The *ERR* LED should turn off. The rear panel pulse status LED on channel 1 should be lit for 96 ms to signal pulses are being output on the channel. Check that the pulse width on the output connectors of the rear panel is approx. 1 μs and the amplitude 24 V.

7. Disconnect the LEMO cable from the front panel and configure the pulse generator for 15 V pulse amplitude, keeping the pulse width to approx. 1 μs.

8. Connect the LEMO cable to the input port of channel 1 on the rear panel. The channel 1 LEDs on both front and rear panels should be lit for 96 ms. Measure that the output pulse on channel 1 is a blocking level pulse with approx. 1 μs pulse width and 24 V in amplitude.

9. Finally, measure on the front panel of the front module that on channel 1 the output pulse is 1 μs long and 3.3 V.

10. Repeat steps 4-9 for all remaining five channels.

3 Front and Rear Panels

Two panels exist in the context of the pulse repeater boards. The first of these is the *front panel*, which corresponds to CONV-TTL-BLO boards and offers various status LEDs, as well as various connectors for TTL and INV-TTL (see Sec. 4) pulses and White Rabbit. The second is the *rear panel*, located on the other side of the backplane and corresponding to CONV-TTL-RTM-BLO boards. The rear panel offers blocking pulse connectors and status LEDs for pulse arrival confirmation.

3.1 Front panel

The front panel of CONV-TTL-BLO boards is shown in Fig. 2. It consists of status LEDs and several ports, divided in four sections from top to bottom:

- System status LEDs;
- Small form-factor pluggable (SFP) connector;
- TTL pulse connectors;
- INV-TTL pulse connectors.

3.1.1 System status LEDs

In the current version of the CONV-TTL-BLO boards, only several of the system status LEDs present on the board are used, due to limited firmware support in the FPGA. The implemented LEDs are presented in Table 2. Unimplemented system status LEDs are off by default.

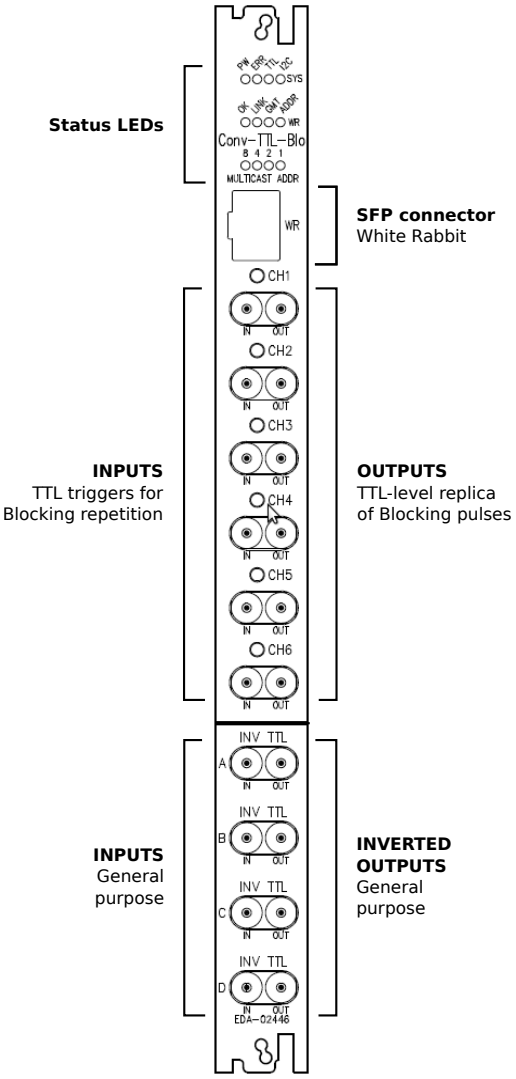

Figure 2: CONV-TTL-BLO panel (front panel)

Table 2: System status LEDs on CONV-TTL-BLO front panels

LED	Description
PW	Power LED. Lights *green* when a valid CONV-TTL-BLO firmware is loaded to the FPGA.
ERR	Error LED. Lights *red* when no rear transition module board is present.
TTL_N	Negated-TTL status LED. Lights *green* when negated TTL logic is selected via the 8^{th} position of the on-board selection switch.

3.1.2 SFP connector

This connector is used to add White Rabbit support to the CONV-TTL-BLO boards. If an optic fibre cable is connected to this socket, White Rabbit precise time-stamping can be added to CONV-TTL-BLO. Four status LEDs above the connector are provisioned to show the status of the White Rabbit link.

White Rabbit is currently not supported in the CONV-TTL-BLO firmware.

3.1.3 TTL triggers

One side of the dual LEMO 00 (type EPY) connector on the CONV-TTL-BLO boards are used for the TTL trigger inputs. By connecting an external trigger source to one of these connectors, a Blocking pulse is generated at the rear panel and a TTL-level pulse is generated at the front panel. The triggers can be either TTL, or INV-TTL level.

All input channels are line-terminated with 50Ω resistors.

3.1.4 Repeated TTL pulses

The other side of the dual LEMO 00 connector is used to output a TTL-level replica of the blocking pulse received at the rear panel, or of the trigger signal arrived on the front panel. The pulse width of this output is similar to the pulse output in the rear panel; the rise time and top pulse level are however different from the Blocking output.

When the pulse is output, the LED of the corresponding channel blinks for 96 ms.

TTL output lines are not internally terminated.

3.1.5 General purpose

Four dedicated inverted-TTL connectors can be found in the lower part of the front panel. Inverted-TTL outputs are not internally terminated.

3.2 Rear panel

The rear panel on CONV-TTL-BLO-RTM boards is shown in Fig. 3. It contains the input and output connectors, as well as pulse status LEDs for six blocking-level pulse channels. A blocking-level pulse at the input connector of a channel is repeated at the three outputs of the same channel in blocking level and TTL level at the output connector of the corresponding channel on the front panel.

When a pulse is repeated on the output connector of a channel, the pulse status LED is lit for 96 ms.

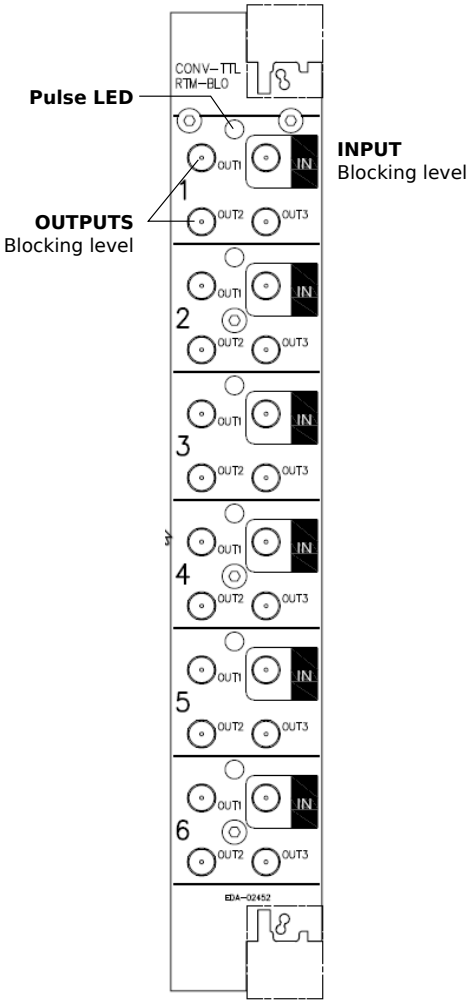

Figure 3: CONV-TTL-BLO-RTM panel (rear panel)

4 Output Pulse Signal

In order for CONV-TTL-BLO boards to work as repeaters, logic is implemented in the on-board FPGA that reacts to a trigger at either rear or front panel and generates a pulse at the output.

Extensive work was made by Carlos Gil-Soriano to research existing boards at CERN and define a standard for pulse levels in repeater boards [1]. Based on this document and on further tests with two of the existing repeater boards at CERN, output pulse widths and amplitudes were selected for the converter boards.

Three types of pulses are defined in the context of CONV-TTL-BLO boards. They differ only in signal amplitude and signal rise and fall times, due to the circuitry used to generate them; pulse widths are the same for all three types. Table 3 presents the different types of pulses and Fig. 4 shows a graphic representation of the pulse signal.

Table 3: Trigger sources

Type	**Pk-pk amplitude**	**Comments**
TTL	3.3 V	
INV-TTL	3.3 V	Inverted version of TTL pulse
Blocking	24 V	Same as TTL, but different level and rise and fall times

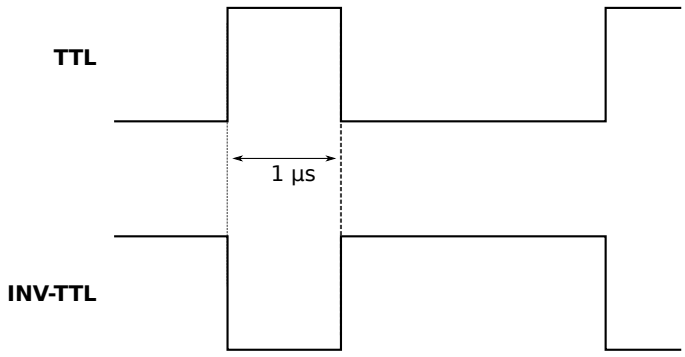

Figure 4: Pulse signal shape

5 FPGA Logic

5.1 Block diagram

A block diagram of the FPGA design is presented in Fig. 5. First, an internal clock signal is generated from the on-board differential 125 MHz clock

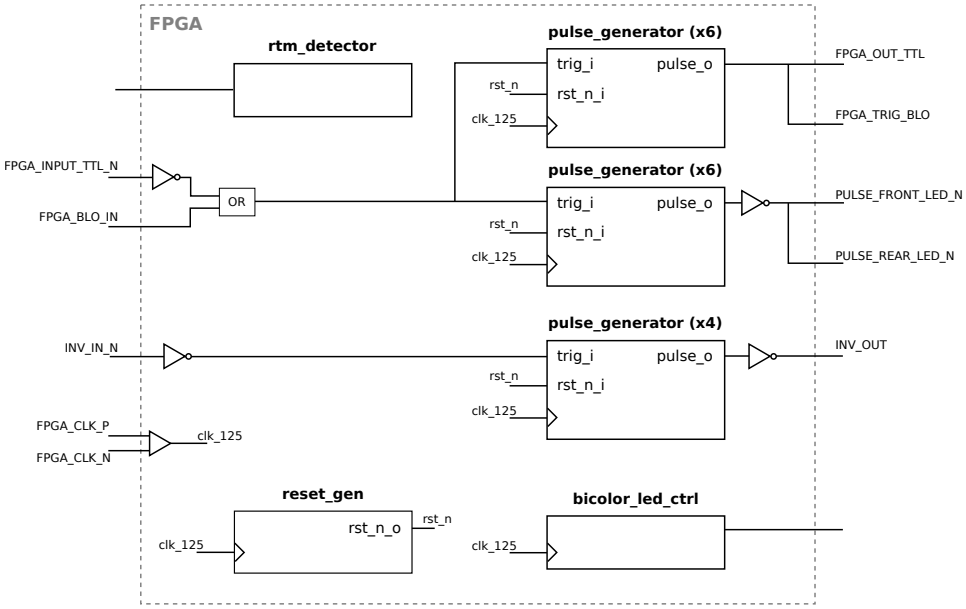

Figure 5: Block diagram of FPGA logic design

oscillator. This internal clock signal is used as the clock for all synchronous logic internal to the FPGA.

The *reset_gen* module generates an internal active-low reset signal that is input to all synchronous with reset. RTM presence is detected via the *rtm_detector* module. Signals generated by this module can be used in various ways in the design; in the current version of the design, they are used as control signals for one of the bicolor LEDs on the board (the *ERR* LED).

Bicolor LEDs on the board are controlled via the *bicolor_led_ctrl* module. Based on input control signals, this module generates signals at a preset refresh rate to light the various status LEDs on board.

Finally, the *pulse_generator* modules generate predefined-width pulses to be transmitted on the blocking, TTL and INV-TTL channels. The same *pulse_generator* module with a different pulse width is used to light the pulse arrival LEDs on front an rear panels.

5.2 Reset generation

The reset generator module (*reset_gen*) implemented inside the FPGA is responsible with generating a predefined-width reset signal when power is applied to the FPGA.

When a power-on reset occurs on the Xilinx FPGA, a counter inside the *reset_gen* module starts counting up. While this counter is counting up, the active-low reset signal is kept low, resetting synchronous logic inside the

Port/generic	Description
g_reset_time	Reset time, in number of *clk_i* clock cycles
clk_i	Input clock signal
rst_n_o	Output reset signal, active-low

FPGA. When the counter reaches the value of the reset width (specified via the *g_reset_time* generic at synthesis time), the reset signal is de-asserted, the counter is disabled and the *reset_gen* module remains inactive until the next power-on reset.

Note that the VHDL of this module is Xilinx and XST-specific and porting to a different FPGA architecture is not guaranteed to provide the same results. The *reset_gen* module has an initial value set for the counter signal after power-up, which is guaranteed by XST to be set after the FPGA's GSR signal is de-asserted.

By default, the reset time is set to 96 ms.

5.3 RTM detection

A simple RTM detection mechanism is employed on CONV-TTL-BLO boards. Three lines on the VME P2 connector are dedicated for RTMM detection, and three lines for RTMP detection. On the CONV-TTL-BLO side, these lines are pulled up to VCC with pull-up resistors. Thus, when no RTMM is plugged in, all six lines (RTMM and RTMP) are logic high due to the pull-up. When an RTM is plugged in, the lines corresponding to the RTMM/P is connected to ground and a logic low will be detected at the FPGA input.

Port/generic	Description
rtmm_i	Input from RTMM detection lines
rtmp_i	Input from RTMP detection lines
rtmm_ok_o	Status of RTMM lines (*high* means RTMM plugged in)
rtmp_ok_o	Status of RTMP lines (*high* means RTMP plugged in)

The *rtm_detector* module simply sets the *rtmm_ok* and *rtmp_ok* signals low if the *rtmm_i* and *rtmp_i* input signals are respectively all-ones.

While more advanced RTM detection can be employed if need be, the RTM signals are currently used to signal an error via the *ERR* status LED on the CONV-TTL-BLO front panel. The *ERR* led is lit when both *rtmp_ok* and *rtmm_ok* outputs are low.

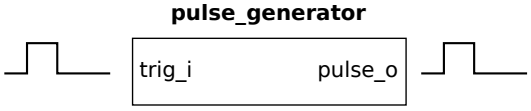

Figure 6: Pulse generator trigger and output polarity

5.4 Pulse generation

The *pulse_generator* module is used to generate pulses of predefined width based on a trigger input. To avoid glitches on the input, the trigger input is taken through a variable-length glitch filter (set by the user at synthesis time via the *g_glitch_filt_len* generic). The glitch filter consists of a series of flip-flops that, when all high, trigger the generation of a variable-width pulse at the output. The width of the pulse is set via the *g_pulse_width* generic.

Port/generic	Description
g_pulse_width	Width of the pulse, in number of *clk_i* clock cycles
g_glitch_filt_len	Length of glitch filter. A length of *1* means the pulse generator block is sensitive to glitches of more than one clock cycle
clk_i	Clock input
rst_n_i	Active-low reset signal
trig_i	Trigger signal, must be active-high
pulse_o	Active-high pulse signal

Assuming active-high triggers arrive at the *pulse_generator* module trigger input, high-level active pulses are generated at the pulse output of the module (Fig. 6). In order to avoid output jitter, the pulse output is selected between the trigger input and the internally-generated pulse signal. The latter is generated using an internal counter, which starts counting once the glitch filter has settled to all-ones. When a trigger arrives at the input it is directed to the output; when the glitch filter settles to all-ones, the pulse signal at the input is extended to the pulse width value set by the *g_pulse_width* generic.

Finally, because the *pulse_generator* module is used to drive the transformers generating the blocking pulses, a pulse rejection mechanism is employed. This mechanism rejects pulses longer than the desired output pulse width by setting a signal that is used to select the signal routed to the output. The pulse rejection signal is set as soon as a pulse is generated via the internal counter, thus once the input has passed through the glitch filter successfully. The pulse rejection signal is cleared when the trigger input settles back to zero and the cycle restarts when a new input pulse arrives. It is assumed that the input pulse frequency is sufficiently low to allow the

transformer to give away all the energy stored in its magnetic field.

Multiple *pulse_generator* modules are instantiated in the design and used to generate pulse signals. Six of these are configured to output 1 μs pulses on both TTL and blocking outputs based on a trigger signal which is the *OR* of TTL and blocking level input pulses. Four are configured to output 1 μs pulses based on a trigger from the INV-TTL channels. Finally, six *pulse_generator* modules are configured to output 96 ms pulses to light the pulse status LEDs corresponding to the six blocking and TTL channels; they are sensitive to the same trigger input as the TTL and blocking pulse generators.

All pulse generator modules instantiated in the design have glitch filters with length four, thus the input trigger pulse has to have a width of at least 32 ns, considering the 125 MHz clock input.

References

[1] C. G. Soriano, "Standard Blocking Output Signal Definition for CTDAH board," Sept. 2011. http://www.ohwr.org/documents/109.

[2] "Under the Hood of Flyback SMPS Designs." http://focus.ti.com/asia/download/Topic_1_Picard_42pages.pdf.