CONV-TTL-BLO User Guide

June 25, 2013

Theodor-Adrian Stana (CERN/BE-CO-HT)

Revision history

Date	Version	Change
19-06-2013	1.00	First version
21-06-2013	1.01	Added termination resistors to Fig. 3, 4
25-06-2013	1.02	New title page and page layout

Contents

List of Figures

List of Tables

List of Abbreviations

FPGA	Field-Programmable Gate Array
I^2C	Inter-Integrated Circuit
PG	Pulse Generator
RTM	Rear Transition Module
SFP	Small Form-factor Pluggable (connector)
VME	VERSAmodule Eurocard

1 Introduction

CONV-TTL-BLO is an open hardware design [1] intended for replicating TTL and blocking pulses. The main features of the board are:

- VME64x form-factor
- Six independent pulse replication channels, each channel capable of replicating
  - TTL to blocking
  - TTL-BAR to blocking
  - Blocking to TTL
  - Blocking to TTL-BAR
  - Blocking to blocking
  - TTL to TTL
  - TTL-BAR to TTL-BAR
- Four general-purpose inverter channels
- Each channel has 50 Ω input termination
- Each channel capable of driving 50 Ω load
- SFP connector for White Rabbit [2]
- Remote monitoring and reprogramming over I^2C lines on VME P1 connector (not implemented)
- Status LEDs
- Pulse LEDs for each replication channel

CONV-TTL-BLO is a VME64x front-module that can be used standalone as a TTL or TTL-BAR pulse repeater using the six replication channels, or as a TTL to TTL-BAR (and viceversa) converter using the four general-purpose inverter channels.

By combining the CONV-TTL-BLO with the CONV-TTL-RTM rear-transition module (RTM) in the rear part of the VME crate, a flexible six-channel pulse conversion system can be obtained. Such a system is shown in Figure 1. TTL pulses arriving on an input TTL channel are regenerated in the FPGA and sent on the channel's TTL output as well as on the three blocking outputs on the RTM. Similarly, blocking pulses arriving on a blocking input channel are regenerated in the FPGA and replicated on both the TTL and blocking outputs of the channel.

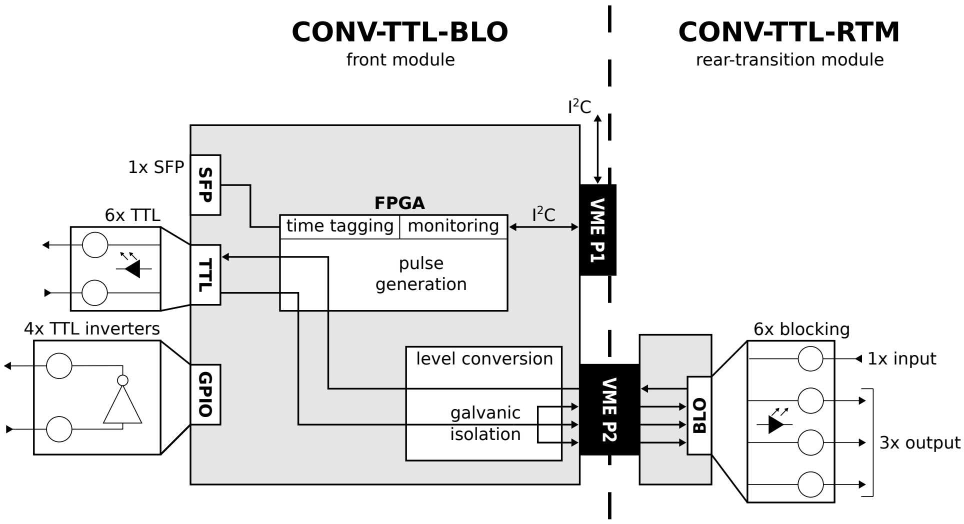

Figure 1: Simplified block diagram of the pulse conversion system

CONV-TTL-BLO contains all active circuitry in a pulse conversion system. It handles pulse generation and conversion from/to blocking, as well as galvanic isolation of blocking outputs. CONV-TTL-RTM is a passive module used as an interface from the rear part of the VME crate to the front module.

2 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 contains various connectors for TTL-level pulses and White Rabbit, as well as various status LEDs. The second is the *rear panel*, located on the other side of the VME backplane and corresponding to CONV-TTL-RTM boards. The rear panel offers blocking pulse connectors and status LEDs for pulse replication confirmation.

2.1 Front panel

The front panel of CONV-TTL-BLO boards is shown in Figure 2. It consists of status LEDs and several ports; these are, from top to bottom:

- System status LEDs
- Small form-factor pluggable (SFP) connector
- TTL pulse connectors and associated pulse LEDs
- General-purpose inverter channels

2.1.1 System status LEDs

There are twelve bicolor status LEDs on the CONV-TTL-BLO front panel. The implemented status LEDs are presented in Table 1. Unimplemented system status LEDs are *off*.

2.1.2 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 provide to show the status of the White Rabbit link.

White Rabbit is currently not supported by the FPGA firmware.

2.1.3 TTL inputs and outputs

Six of the LEMO 00 connectors on the CONV-TTL-BLO board are TTL repeater channels. Both front-panel inputs and outputs are TTL-level. The signal type and the inputs and outputs can be either TTL or TTL-BAR, as selected by the TTL switch (SW2.4, see Section 4.2).

A simplified diagram of pulse repetition is shown in Figure 3, more details can be found in Section 4.3. If a TTL (TTL-BAR) pulse arrives on a channel

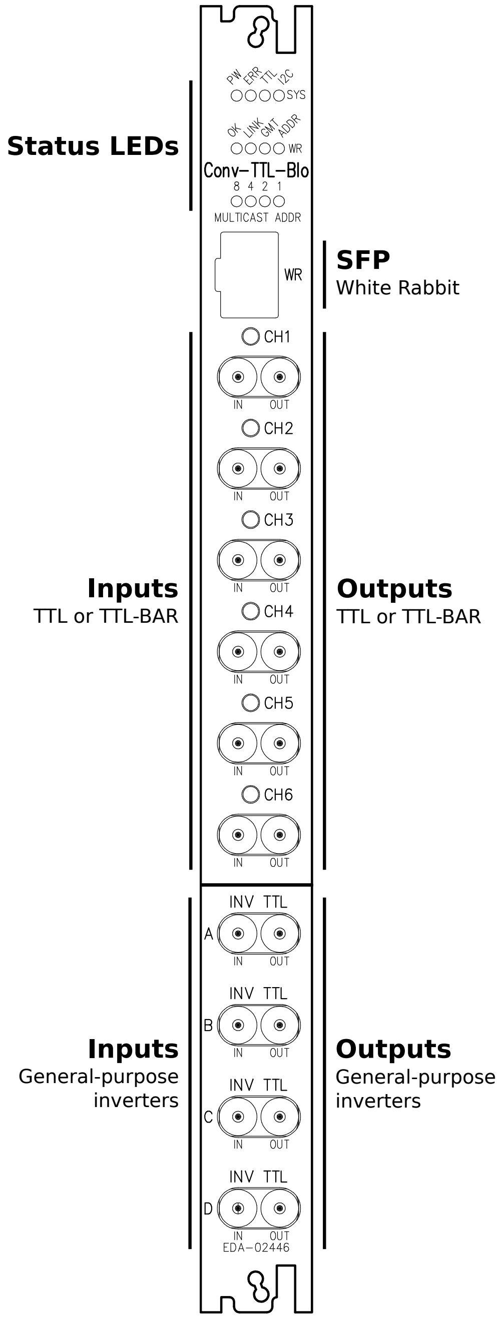

Figure 2: CONV-TTL-BLO panel (front panel)

Table 1: System status LEDs on the CONV-TTL-BLO front panel

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 RTM board is present, *off* if a valid RTM is present.
TTL	TTL status LED. Lights *green* when TTL logic is selected via the on-board selection switch, *off* when TTL-BAR logic is selected.
I2C	I^2C status LED. Flashes *green* when an I^2C transfer is taking place. Lights *red* when a transfer error occurs, or when the CONV-TTL-BLO register being addressed does not exist. After a transfer error, the LED will still flash *green* on transfer and return to a *red* color after the transfer has ended; it can only be turned back *off* via a system reset.

input, it is regenerated on the output of the same channel in TTL (TTL-BAR), as well as the blocking outputs of the same channel on the rear panel, if a CONV-TTL-RTM board with an attached CONV-TTL-RTM-BLO is present. Similarly, if a blocking pulse arrives on the back panel, it is replicated on the TTL output channel.

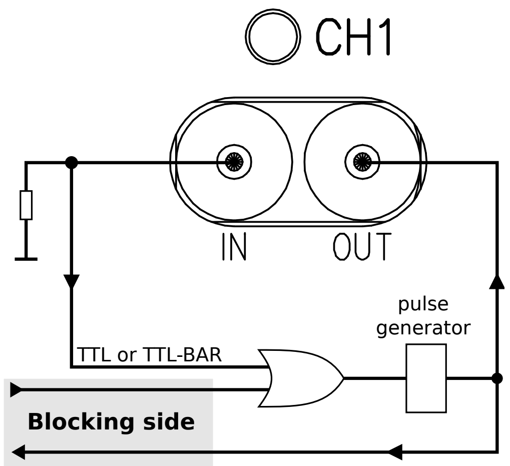

Figure 3: Pulse repetition on front channel

Each TTL replication channel has a pulse LED which flashes shortly whenever a pulse is replicated on the channel.

All TTL input channels are terminated with 50Ω resistors; TTL output channels are not terminated.

2.1.4 General-purpose inverters

Four general-purpose TTL inverter channels can be found in the lower part of the front panel. The output of a channel is always an inverted version of the channel input (Figure 4). No regeneration is performed on the input signal, nor is it in any way connected to the blocking outputs on the RTM. The input signal is simply passed through an inverter and presented at the channel output.

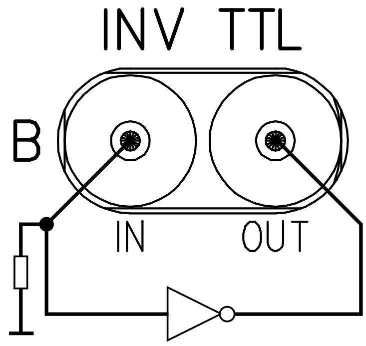

Figure 4: TTL general-purpose inverter channel

All general-purpose inputs are terminated with 50Ω resistors; the outputs are not terminated.

2.2 Rear panel

The rear panel on CONV-TTL-BLO-RTM boards is shown in Figure 5. It contains the input and output connectors, as well as pulse status LEDs for six blocking-level pulse channels. A blocking pulse at the input connector of a channel is regenerated at the three outputs of the same channel in blocking level and in TTL level at the output connector of the corresponding TTL channel on the front panel.

All blocking input channels have internal termination with 50 Ω resistors. Blocking outputs are not terminated. Each output on a channel has a separate blocking driver capable of driving a 50 Ω load.

When a pulse is repeated on the output connector of a channel, the pulse status LED flashes briefly.

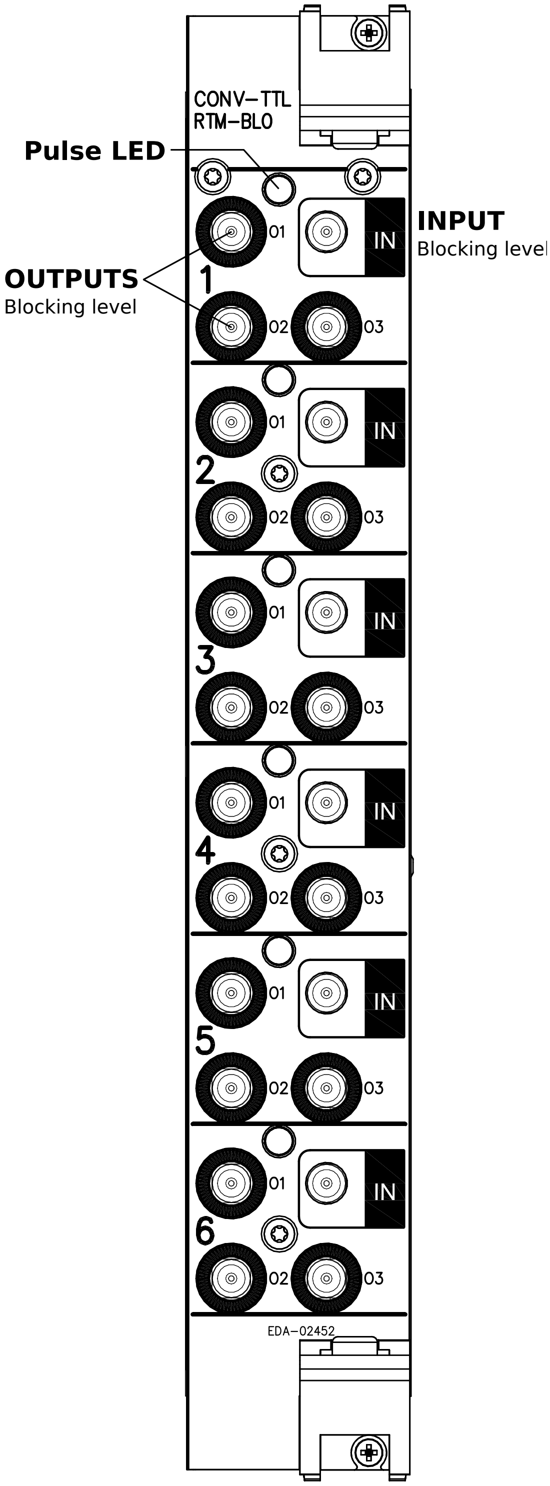

Figure 5: CONV-TTL-BLO-RTM panel (rear panel)

3 On-board switches

There are eight switches provided on-board the CONV-TTL-BLO, not all of which are used. Figure 6 shows the switches and highlights the used ones; the used switches are also listed in Table 2.

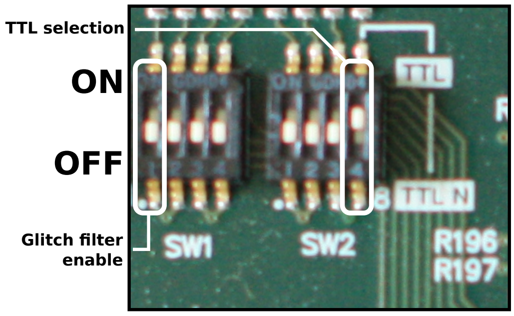

Figure 6: Switches on the CONV-TTL-BLO board

Table 2: Switches on CONV-TTL-BLO

Switch	Description
SW1.1	Glitch filter enable (see Section 4.4)
	ON – glitch filter enabled, output jitter present
	OFF – glitch filter disabled, no output jitter (default)
SW2.4	TTL/TTL-BAR selection switch (see Section 4.2)
	ON – TTL channels receive and generate TTL (default)
	OFF – TTL channels receive and generate TTL-BAR

Note that both switches in Table 2 are board-wide switches; selecting one position or the other yields a selection valid for all six pulse replication channels.

4 Pulse replication

4.1 Pulse signal definition

Three pulse types are defined in the context of CONV-TTL-BLO, depending on signal amplitude, rise and fall times; pulse widths and frequencies are the same. TTL and TTL-BAR pulses are input and output on the front panel of the boards. TTL-BAR is essentially an inverted version of TTL signals (see Section 4.2). Blocking pulses [3] are differential signals and are input and output on the rear panel (via a CONV-TTL-RTM).

The various characteristics of the pulse signals are defined in Figure 7 and outlined in Table 3 for TTL and TTL-BAR pulses, and in Table 4 for blocking pulses.

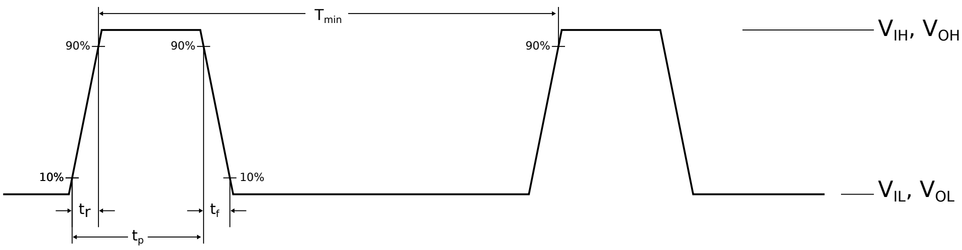

Figure 7: Pulse signal characteristics

Table 3: TTL and TTL-BAR pulse characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input pulse high-level amplitude (1)(2)	1		5.5	V
V_{IL}	Input pulse low-level amplitude (2)	0.7		1.6	V
V_{OH}	Output pulse high-level amplitude	2.4	3.3	5	V
V_{OL}	Output pulse low-level amplitude		0	0.7	V
$t_{p,i}$	Input pulse width	50			ns
$t_{p,o}$	Output pulse width		1.2		μs
T_{min}	Period of pulse signal (3)	4.8			μs
t_r	Rise time	1	3.2	4.9	ns
t_f	Fall time	2	4	5.6	ns

Note 1: Pulse amplitude for which a $t_{p,o}$ pulse is replicated at the output.
Note 2: V_{IH}, V_{IL} correspond to the thresholds of input Schmitt triggers.
Note 3: Max. pulse frequency dictated by blocking output max. frequency.

Table 4: Blocking pulse characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Input pulse high-level amplitude (1)	3.7		25	V
V_{IL}	Input pulse low-level amplitude	-5			V
V_{OH}	Output pulse high-level amplitude (2)	23	24	25	V
V_{OL}	Output pulse low-level amplitude (2)		0		V
$t_{p,i}$	Input pulse width	50		2000	ns
$t_{p,o}$	Output pulse width		1.2		μs
T_{min}	Min. period of pulse signal	4.8			μs
t_r	Rise time	75	140	225	ns
t_f	Fall time	75	160	225	ns

Note 1: Pulse amplitude for which a $t_{p,o}$ pulse is replicated at the output.
Note 2: Voltage amplitude between the differential signal lines

4.2 TTL vs. TTL-BAR

The two signal types that may be replicated on the front panel are TTL or TTL-BAR. As Figure 8 shows, TTL-BAR is an inverted version of TTL.

Selection between these two signal types is done by means of the TTL selection switch, SW2.4 (Figure 9). When the switch is **ON** (default), TTL pulses arriving on the front panel input or blocking pulses arriving on the rear panel input are replicated to TTL pulses on the front panel. When the switch is **OFF**, TTL-BAR pulses arriving on the front panel or blocking pulses arriving on the rear panel are replicated to TTL-BAR pulses on the front panel.

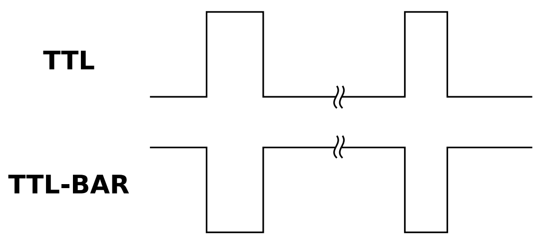

Figure 8: TTL and TTL-BAR signals

The TTL selection switch is valid board-wide, i.e., if it is set for TTL inputs (**ON**), TTL signals should be input on all TTL channels. Inputting TTL-BAR signals on a channel while the TTL/TTL-BAR selection switch is set to TTL is not an intended functionality for the board and is not encouraged.

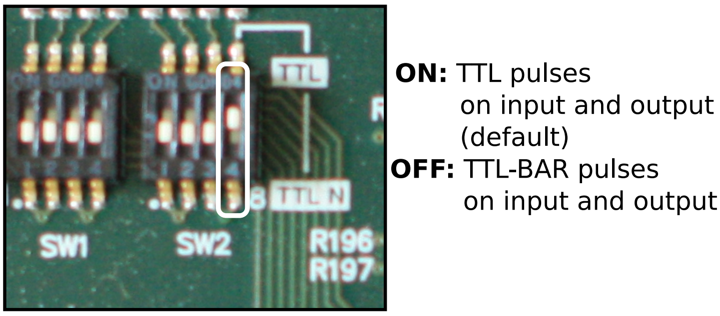

Figure 9: TTL/TTL-BAR selection switch

4.3 Pulse replication mechanism

Figure 10 shows a diagram of how pulses are replicated on a channel inside the FPGA. The figure also shows the shape of the different types of pulse signals after they pass through a part of the circuit. The grey DC signals are the signals when no wire is plugged into a channel.

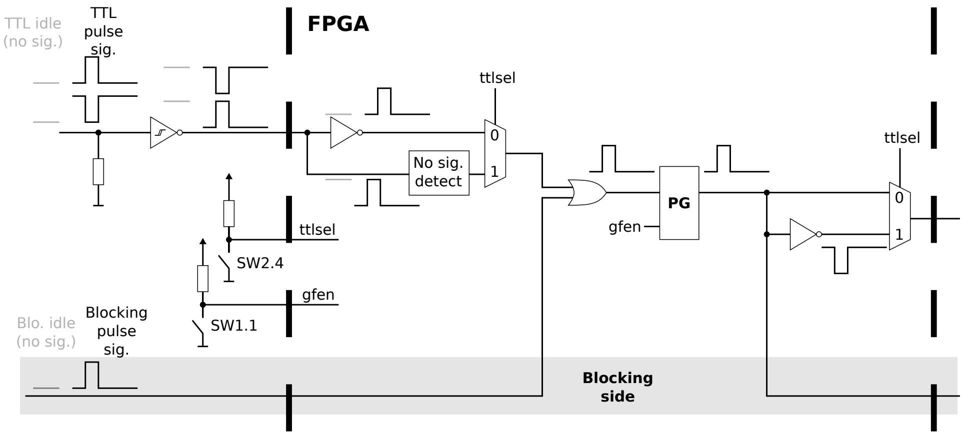

Figure 10: Pulse repetition mechanism

The pulse generator (PG) block in the FPGA generates $t_{p,o}$ wide (Table 3) TTL pulses at its output on the rising edge of its input. It therefore expects TTL pulses at its input. The rest of the logic external to this block is used to accomodate for TTL-BAR and blocking pulses.

First, the OR gate at the PG input indicates the condition for a pulse to be regenerated. When a pulse arrives at either of the two inputs, TTL or TTL-BAR on the front panel, or blocking on the rear panel, a pulse is generated on the output.

On the blocking side, the voltage level of blocking pulses arriving on the RTM is adapted to a voltage level suitable for the FPGA by on-board circuitry external to the FPGA. What ends up in the FPGA is a TTL type pulse, so this may be passed directly to the PG's input through the OR gate. The output of the PG block is passed to the FPGA output and to the three blocking pulse outputs of a channel, where the blocking-level pulse is generated.

On the TTL side, pulse signals go through a Schmitt trigger inverter buffer to the FPGA. The termination resistor pulls the input line to ground when there is no signal, so this gets inverted to V_{cc} by the Schmitt trigger. Based on the setting of the TTL switch (see Section 4.2), the multiplexer assures a TTL signal at the OR gate input.

The *no signal detect* block at the multiplexer input on the TTL-BAR side detects the lack of a signal by checking for a continuous high level on the line. This is important when the TTL selection switch is set to TTL-BAR, since no signal would mean a DC high-level signal appears at the OR gate input and this signal would inhibit pulses arriving from the blocking side.

4.4 Pulse jitter and delay

The PG block incorporates a glitch filter that can prevent pulses being generated as a result of a glitch occuring on input channels. The glitch filter can be enabled via SW1.1 (Figure 11).

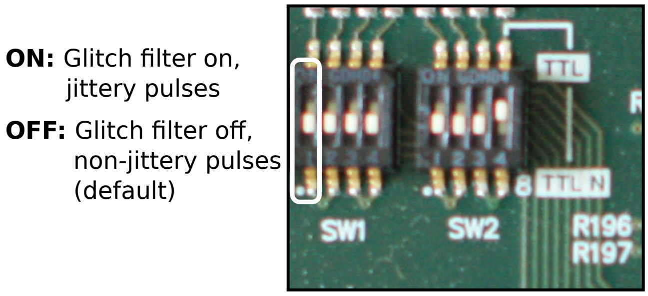

Figure 11: Glitch filter enable switch

Placing SW1.1 in the **ON** position will enable the glitch filter inside the PG block. The pulse signal is sampled with a 125 MHz on-board clock and passed through the glitch filter which rejects any pulses narrower than 40 ns, but introduces an 8 ns jitter on the leading edge of the output signal. Jitter appears in the form of pulses being triggered either 8 ns before or 8 ns after the ideal edge, based on when the input pulse is sampled. This is shown in Figure 12.

When SW1.1 is in the **OFF** (default) position, the giltch filter is disabled and the pulse signal is regenerated at the output without being sampled with an on-board clock. This yields jitter-free pulses at the output, but a glitch on the input will lead to a pulse being generated at the output.

The glitch filter internal to the PG block may be enabled when the environment where the board operates is noisy. A noisy environment may lead to glitches induced on the signal lines and thus unwanted pulses on the output of the CONV-TTL-BLO. When the environment is not so noisy, or when the 8 ns jitter is deemed to be an issue, SW1.1 can be left in its default position.

Figure 12 defines propagation delay from input to output and output jitter and shows how propagation delay was measured per each signal type. Table 5 presents the characteristics measured on the CONV-TTL-BLO.

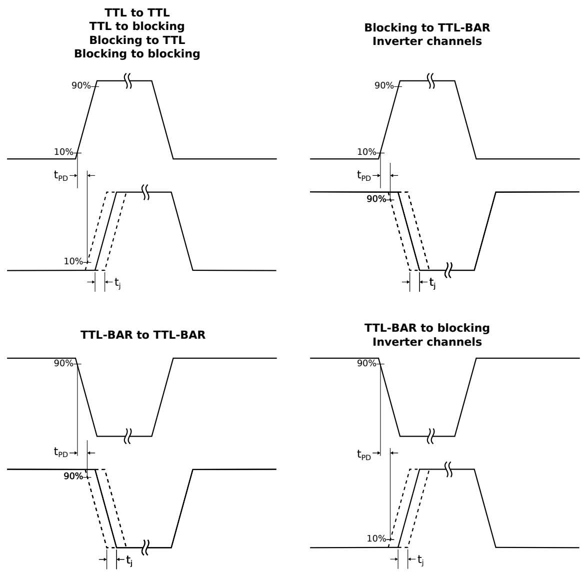

Figure 12: Output pulse delay and jitter

Table 5: Output pulse delay and jitter

Symbol	**Parameter**	**Value**	**Unit**
t_j	Leading edge jitter		
	Without glitch filter	0	ns
	With glitch filter	8	ns
t_{PD}	Propagation delay (1)		
	TTL to TTL	40	ns
	TTL to blocking	80	ns
	Blocking to TTL	80	ns
	TTL-BAR to TTL-BAR	50	ns
	TTL-BAR to blocking	90	ns
	Blocking to TTL-BAR	90	ns
	Blocking to blocking	120	ns
	Inverter channel	30	ns

Note 1: If glitch filter is enabled, it adds an extra 40 ns delay to t_{PD}.

5 Communicating with the CONV-TTL-BLO

It is possible to communicate to the CONV-TTL-BLO remotely via the VME P1 I^2C interface. This section describes how to connect to the VME64x crate and communicate to the board.

5.1 The ELMA I^2C protocol

In order to connect to a CONV-TTL-BLO board in an ELMA VME crate, a higher-level protocol based on I^2C is defined [4]. The protocol uses the serial lines on the VME P1 connector (*SERCLK*, *SERDAT*).

By this protocol, 2^{12} (12 address bits) 32-bit registers can be read from or written to byte by byte. The user accesses the VME crate using Telnet and sends commands which the ELMA SysMon board translates to I^2C transfers to the board.

Two telnet commands (see Table 6) can be used to transfer data to the board. As their names suggest, *readreg* reads a board register, whereas *writereg* writes to a board register.

Table 6: The *readreg* and *writereg* commands

Command	Description
writereg *slot reg val*	Writes the value *val* to register number *reg* of board in slot number *slot*
readreg *slot reg*	Returns the value of register number *reg* of board in slot number *slot*

The register numbers are integer numbers starting from one. The actual addresses sent to the board are word-aligned starting from 0x000. For example, *reg 1* translates to address 0x000, *reg 2* to 0x004 and so on. The following relation exists between ELMA *reg* numbers to on-board register addresses:

$$addr = (reg - 1) * 4$$

An example of retrieving the CONV-TTL-BLO ID from *reg 1* of a CONV-TTL-BLO plugged into VME slot 2 of the crate *some-crate* is given below. If the board is present in slot 2, the command should yield the ASCII string **BLO2**.

```
tstana@tstana-unit:~$ telnet some-crate
Trying 137.138.192.90...
Connected to some-crate.cern.ch.
Escape character is '^]'.
login:user
```

```
password:**********
%>readreg 2 1
 Read Data: 424C4F32
%>
```

First, a telnet connection is made with the crate, after which the *readreg* command is issued to read the value of *reg 1* (address 0x000). The value of the register can be confirmed to be the hex value of the ASCII string **BLO2**, so the board is indeed present in the slot.

Another example of running the same command, this time with the board removed from the crate, is given below. As expected, when the board is removed, it can no longer acknowledge the I^2C access, thus the message:

```
Connected to some-crate.cern.ch.
Escape character is '^]'.
login:user
password:**********
%>readreg 2 1
 Not Acknoledged!
%>
```

5.2 Memory map

The memory map of registers accessible through the ELMA protocol is given in Table 7.

Table 7: CONV-TTL-BLO memory map

Address	ELMA reg	Description
0x000	1	Board ID register *access*: R/W *default*: 0x424C4F32 (ASCII **BLO2**)

Appendices

A Getting started with the CONV-TTL-BLO

1. Plug in the CONV-TTL-RTM board into the rear part of the VME crate.
2. Remove the CONV-TTL-BLO board from its box and ESD-protective bag.
3. Based on the type of pulses desired on the front panel, set the TTL selection switch (SW2.4, see Section 4.2) to the appropriate position:
   - TTL pulses – set the switch to the **ON** position (default)
   - TTL-BAR pulses – set the switch to the **OFF** position
4. Set the glitch filter enable switch (SW1.1, see Section 4.4) to the appropriate position:
   - Glitch filter disabled, jitter-free signal at output – set the switch to the **OFF** position (default)
   - Glitch filter enabled, introduces output jitter – set the switch to the **ON** position
5. Insert the CONV-TTL-BLO board into the VME crate and power on the crate.
6. Check that the ***PW*** status LED is lit ***green***. If there is no RTM in the rear side of the crate, the ***ERR*** LED will light ***red***. The ***TTL*** status LED should also be lit ***green*** if you set the TTL switch to the **ON** position in step 3.
7. Input a TTL (or TTL-BAR) signal into a front panel input channel. When a pulse arrives on the input, it is replicated on the output of the same channel. If an RTM board is present in the rear part of the VME crate, the pulse will also be replicated on the three blocking outputs of the same channel on the rear-panel. The channel pulse LEDs on both the front and rear panels flash briefly when a pulse arrives.
8. Input a blocking signal on a rear panel channel; the pulse LED of the channel will flash and the pulse will be replicated on the three blocking outputs of the same channel, as well as the TTL channel output on the front panel. If the TTL switch is **OFF**, the pulse is replicated in TTL-BAR.

B Typical use cases

B.1 Repeating one blocking pulse to twelve separate ones

Such a setup is outlined in Figure 13. Only one external blocking signal is required and it is input to CH1 on the rear panel. A daisy-chain is created on the front panel starting from CH1 to CH2, up to CH4. The front panel is preferred here due to smaller delay in replicating pulses (see Section 4.4).

Blocking pulses arriving on CH1 then get replicated through the daisy chain from CH1 to CH4. By connecting all outputs of channels 1 through 4 on the rear panel, the desired pulse conversion system can be obtained.

Each channel will add a 40 ns delay (80 ns with glitch filter), in addition to the 160 ns (240 ns with glitch filter) of the CH1 and CH4 blocking conversions.

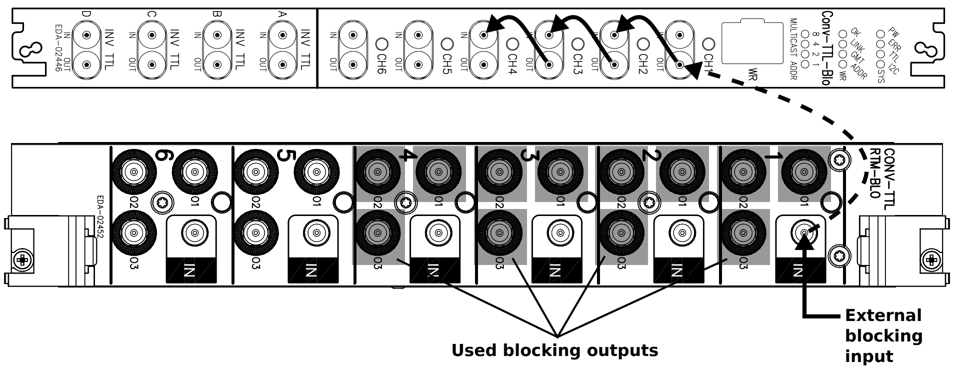

Figure 13: Setup for converting one blocking signal into 16 separate ones

B.2 Repeating TTL pulses in TTL-BAR

When the board has already been plugged in and the switch has been set in, e.g., the **OFF** position, only TTL-BAR pulses can be input on a front panel replication channel. If the user desires to input a TTL pulse and repeat it into TTL-BAR, one of the four general-purpose inverter channels can be used. Figure 14 shows a setup for inverting TTL pulses into TTL-BAR on inverting channel A and repeating them on front panel channel 6.

The inverter channel will add a 30 ns delay to the input TTL signal.

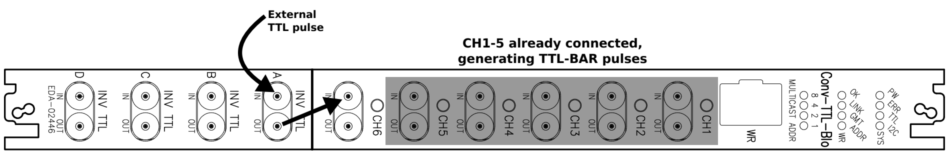

Figure 14: Setup for repeating TTL pulses in TTL-BAR

References

[1] “Conv TTL Blocking webpage on OHWR.” `http://www.ohwr.org/projects/conv-ttl-blo`.

[2] “White Rabbit.” `http://www.ohwr.org/projects/white-rabbit`.

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

[4] ELMA, “Access to board data using SNMP and I2C.” `http://www.ohwr.org/documents/227`.

[5] “CONV-TTL-BLO Schematics.” `https://edms.cern.ch/file/1278535/1/EDA-02446-V2-1_sch.pdf`.