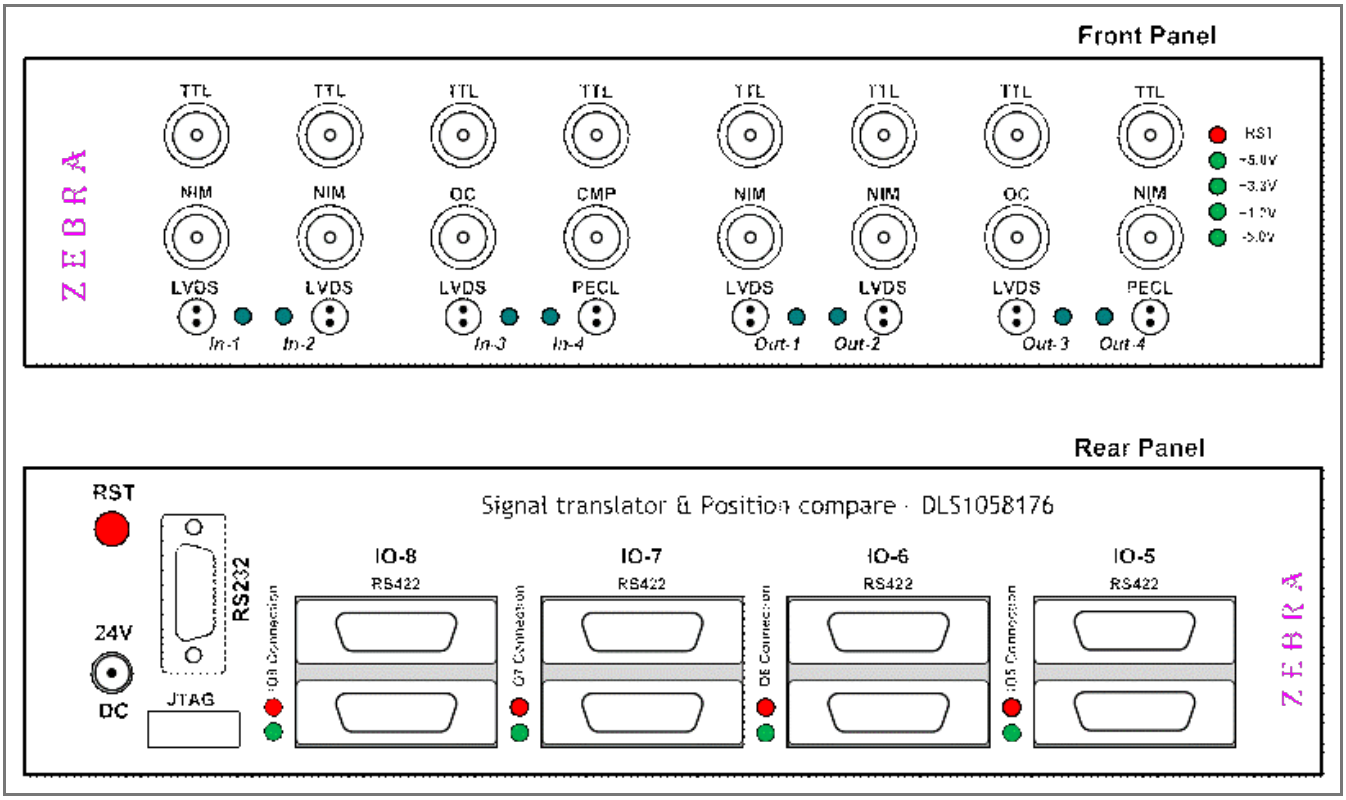
Beamline signal converter box

Introduction

Many times on beamlines it has been necessary to convert one signal type to another at short notice. There is also a requirement to be able to synchronise motor positions and detector triggers regardless of the motor controller. This specification outlines a box that can do both of the above.

Sketch



Requirements

Inputs

It should have the following inputs:

- 4 TTL
- 3 LVDS
- 1 PECL
- 2 NIM
- 1 Open collector
- 1 Comparator (with pot to adjust)
- 4 Quadrature encoders (3 RS422 channels each, A, B, Z, plus error)

For the encoders, if the input is open circuited it should set in error state

Each input should be labelled, and connected through appropriate input circuitry to an input of an FPGA

The inputs should be in groups of 3 as shown in the image at the bottom of the page, each group should have an LED attached

For the LVDS inputs, if the output is disconnected, it should translate to a logic 0 in the FPGA

Output

It should have the following inputs:

- 4 TTL (capable of driving a 50 ohm output)
- 3 LVDS
- 1 PECL
- 4 NIM
- 4 Quadrature encoders (3 RS422 channels each, A, B, Z, plus error)

For the encoders, if it is in error state, the outputs should be open circuited, all non-RS422 signals should be passed straight through

Each output should be labelled, and connected through appropriate output circuitry to an output of an FPGA

The outputs should be in groups of 3 as shown in the image at the bottom of the page, each group should have an LED attached

Form Factor

There should be a single PCB that can be mounted in a 1U rackmount enclosure, or a standalone box

Power

It should be powered from the mains

FPGA

The FPGA would look something like this and have an RS232/ethernet connection, a reset button, and retain settings between power cycles. If the reset button is depressed when the unit is turned on it should load default settings rather than saved settings.

The FPGA will be clocked at 50MHz, so the shortest pulse width is 20ns

Logic blocks

There will be a number of logic blocks that can be soft-wired together to give functionality

4 input AND gate

The logical AND of its inputs, each of which can be inverted

4 input OR gate

The logical OR of its inputs, each of which can be inverted

Single pulse generator

Edge triggered pulse generator with width and delay control

Pulse divider

Divisor D splits pulses into every Dth, and every other

Gate generator

Edge triggered flip flop with set and reset inputs

Position compare

3 signals, Arm, Gate, Pulse. When position compare is armed, a series of gates is output, and a series of pulses output during each gate.

Arm

- Mode: Soft / Hardware triggered

Gate

- Mode: Time / Position / External
- Start: Where to set the gate high
- Width: How long to keep the gate high
- NGates: How many gates to output
- Step: Increment between gates if more than one gate

Pulse

- Mode: Time / Position / External
- Delay: Delay from start of gate to first pulse
- Width: How long to keep the pulse
- Step: Increment between pulses

Data capture

On each pulse output of position compare, the 4 encoder counters or the 4 divisor counters should be captured. A circular buffer in the FPGA should store these values with a timestamp then send them over the serial line. There should be enough internal memory to store 100000 of these events.