NAME

trlo − Setup file for TRLO II.

SYNOPSIS

.trlo

DESCRIPTION

With around 500 setup register of 60 kinds and 200 signals of almost as many variations in TRLO II, efficient handling can be a daunting task. The *libtrlo_ctrl* and *trlo_ctrl* companion library and command-line interface program for control, monitoring and read-out help with this. Complex configurations can be loaded from setup files, while one-off commands for testing can be quickly issued from a shell.

A *trlo* file is used to store the connections to be used within a module running the TRLO II. The file can be loaded with the *trlo_ctrl* program (or *libtrlo_ctrl* library), and sections executed. Executing a section means to perform the actual programming of the hardware setup registers. The separation in sections is to be able to describe (slightly) different configurations that are used in different modes, e.g. when a system is operated stand-alone or as multi-branch subsystem.

SYNTAX

The file has a C-like syntax. Expressions are enclosed in function-like sections.

SECTIONS

A command section has the form

SECTION(*name*)
{
expression;
[*expressions...*]
}

The section is executed by NAME and can also be CALLed from within another section.

EXPRESSIONS

The signal multiplexer is controlled by assigning a source to a destination.

destination <= *source*;

(Previously, '=' was used. This is still accepted, but deprecated.)

Functional units which can operate on several signals are assigned in similar ways, with logic operators in between.

LMU_OUT <= *LMU_IN* [**and|or** *LMU_IN* ...];

Special output signals which are assigned using bitmasks, and therefore can be assigned to multiple outputs are given in a reverse order, e.g.

out_mask => *destination*[, *destination*];

(Previously, '=' was used. This is still accepted, but deprecated.)

Setup registers are assigned values as expected.

setupreg = *x*;

Values representing times are given with units. In cases where the TRLO II firmware imposes a minimum value (such that a 0 register value corresponds to a non-zero offset), this is taken into account by the execution. The setup file holds the effective time.

setupreg = *t* ns;

Period times can also be given as frequencies (in Hz or kHz).

setupreg = *f* Hz;

Where the values represent choices, the assignment is with a describing name

setupreg = *MODE*;

CONTROL FLOW EXPRESSIONS

The contents of another section can be executed by:

CALL(*name*);

ALIASES

Aliases can be declared. Both for values, and signal names.

This can also be used to give meaningful names to values, as well as signals. Such aliases are declared outside the scope of command **SECTION**s:

some_signal_alias := *signal*;
some_value_alias := *x*;
some_time_alias := *t* ns;
some_freq_alias := *f* Hz;

(Previously, '=' was used. This is still accepted, but deprecated.)

EXAMPLES

Would be nice...

AUTHOR

Håkan T. Johansson <f96hajo@chalmers.se>

SEE ALSO

trloctrl(1),