RedPitaya - Timing Module

Firmware

Data types

The FPGA uses 40bit unsigned integer which is mapped to software registers as *uint64*. The elements of the sequence vector TIMES are 48 bits long and consist of a 40bit time value (bit 0:40) and a 8bit channel mask (40:48). The maximum number of elements of TIMES limited by the hardware constrains of the FPGA (≈ 45000 @ 48bit).

private parameters - registers

address	name	type	description
0x0000	STATUS	uint64[8]	Status register stores debug and state information of the FPGA (Table 1).
0x0008	INIT	uint8	Used to arm the device or abort an executed program.
0x0009	TRIG	uint8	Used to software trigger the device.
0x000A	CLEAR	uint8	Used to clear error from the status register.
0x000B	REINIT	uint8	Set if device in reinit mode.
0x000C	SAVE	uint8	Used to set up reinit mode.
0x000D	CLKSRC	uint8	Controls clock source: 0:internal (10Mhz) sync w/ clock, Bit1:external, Bit2:sync w/ trigger.
0x000E	INVERT	uint8	Stores bit information of channels that are set for inverted output.
0x000F	GATE	uint8	Stores bit information of channels that are set for gate mode.
0x0010	DELAY	uint64	Number of tick between the input trigger and the beginning of the first sequence.
0x0018	WIDTH	uint64	Number of ticks the output remains high after raised.
0x0020	PERIOD	uint64	Minimum number of ticks between two raising edges: defines the rate of the pulse train.
0x0028	BURST	uint64	Number of pulses in a pulse train.
0x0030	CYCLE	uint64	Total number of ticks before the cycle repeats.
0x0038	REPEAT	uint64	Number of cycle repetitions.
0x003C	COUNT	uint32	Number of pulse trains in a sequence.
0x0040	TIMES	uint64[]	Increasing tick counts since cycle beginning to raising edge of the first pulse in a train.

idx	description
0	Status byte (Table 2).
1	Next scheduled index (0:16).
2	Next scheduled time (0:40) and mask (40:48).
3	Current cycle count (0:40).
4	Current sequence count (0:40).
5	Current burst count (0:40).
6	Current repeat count (0:40).
7	Extended StatusX register (Table 3).

Table 1: Status register

bit	description
0	signal 0
1	signal 1
2	signal 2
3	is idle
4	is armed
5	is running
6	trigger high
7	clock high

Table 2: Status byte

bit	description	cleared	set
0	gstate[0]	idle/armed	run/wait
1	gstate[1]	idle/wait	run/armed
2	wait_for mode	delay	cycle
3	clear	not-	requested
4	inc_index	not-	requested
5	restart	not-	requested
6	restart_check	pending	confirmed
7	error	no error	occurred
8	gstate switch	not-	processed
9	trigger check	not-	processed
10	waiting	not-	processed
11	run program	not-	processed
12	run program 1	not-	processed
13	run sequence	not-	processed
14	rearm check	not-	processed
15	increment	not-	processed
56	signal 0	low	high
...			
63	signal 7	low	high

Table 3: Extended StatusX register

LED and DIO connections

pin	1	2	3	4	5	6	7	8	9	10	11	12	13
p(inner)	+3V3	ch0	ch1	ch2	ch3	ch4	ch5	ch6	ch7	nc	nc	nc	gnd
n(outer)	+3V3	trg_in	trg_out	clk_in	20MHz	clk_out	run_out	nc	nc	nc	nc	nc	gnd

input, output, *inverted*

LED

index	0	1	2	3	4	5	6	7	8	9	*P*	*D*
led	CH0	CH1	CH2	CH3	idle		active	clock				

input, output, *inverted*

Channels ch0-ch5 can be inverted or switched to gate mode using the invert() and gate() method, respectively.

Parameter contrains/defaults

These constrains should be checked against by the driver before programming.

name	constrain	default	default*
DELAY	≥ 0	0	$60,000,000$
WIDTH	≥ 1	WIDTH/2	5
PERIOD	$>$ WIDTH	10	10
BURST	≥ 0	1	0
CYCLE	$\geq \text{COUNT} * \text{PERIOD} * \text{BURST}$	$\text{COUNT} * \text{PERIOD} * \text{BURST}$	0
	$\geq \text{TIMES[end]} + \text{PERIOD} * \text{BURST}$	$\text{TIMES[end]} + \text{PERIOD} * \text{BURST}$	-
REPEAT	≥ 0	1	0
COUNT	≥ 0	1	1
	$\geq 0, \quad \leq \text{len(TIMES)}$	len(TIMES)	-

*) default value if nothing or only DELAY is provided, only valid for make_clock.

Methods (public methods)

make_clock (method = 'C')

The parameters DELAY, WIDTH, PERIOD, BURST, CYCLE, and REPEAT are transmitted. TIMES defaults to [0].

make_sequence (method = 'S')

The parameters DELAY, WIDTH, PERIOD, BURST, CYCLE, REPEAT, COUNT, and TIMES are transmitted.

arm (method = 'A')

INIT is set. This will cause the program to react on trig events.

disarm (method = 'D')

INIT is cleared. This will interrupt the program if running. The device is idling until the next init.

trig (method = 'T')

TRIG is set and if armed the device will start the program, i.e. software trigger.

reinit (method = 'R')

REINIT mode is set up (¿=0) or deactivated (-1/None).

clksrc (method = 'E')

CLKSRC is configured. This selects between internal 10MHz (0) and external (1) clock. The second bit activates the trgsync feature in which the output aligns with the trigger instead of being tied to the clock.

gate (method = 'G')

GATE is configured as bit register controlling the output channels whether to be in signal (0) or gate (1) mode.

invert (method = 'I')

INVERT is configured as bit register controlling the output channels whether to be in normal (0) or inverted (1) mode.

state (method = 's')

Requests the current state of the device.

params (method = 'p')

Requests the parameters the device is currently configured with.

error (method = 'e')

Requests the error/message string.

status (method = 'x')

Requests the status buffer holding current index, sample, sequence, cycle, burst, and repeat value in addition to a flag register.

try_extclk (method = 't')

Test if external clock can be detected (side effect: debug(0) and disarm()).

Timing

The board cycle can be synchronized to an external clock or will use its internal 10Mhz clock. The jitter is minimal. The timestamps of the TTLs can be calculate with little uncertainty based on the clock. The uncertainty is mainly influenced by the quality of edge of the pulse.

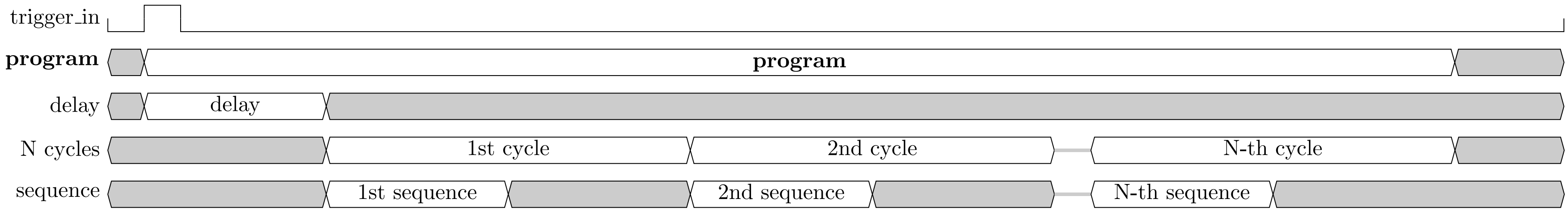
Trigger schema in reinit mode

At W7X there will be a trigger $T_0 = -60$s before the experiment starts at $T_1 = 0$s. This trigger will engage the initialization of all diagnostics as well as the timing module. In order to supports output signals even seconds before T_1 it is required to trigger the timing module with T_0. In reinit mode the triggered module enters the delay phase set to 60s. During the delay phase the module accepts a reconfiguration (makeClock, makeSequence) without loosing track of the tick count with respect to T_0. The updated delay and timing configuration are used for the rest of the program. If the device is not configured before the end of delay, it returns to armed state without generating any pulse.

Timing

Program

A program contains the list of strictly monotonic increasing tick counts relative to the trigger input for all pulses that will be generated when a trigger arrives. It is the lowest level of control. A program can be compiled as a initial DELAY and REPEAT identical, subsequent cycles.

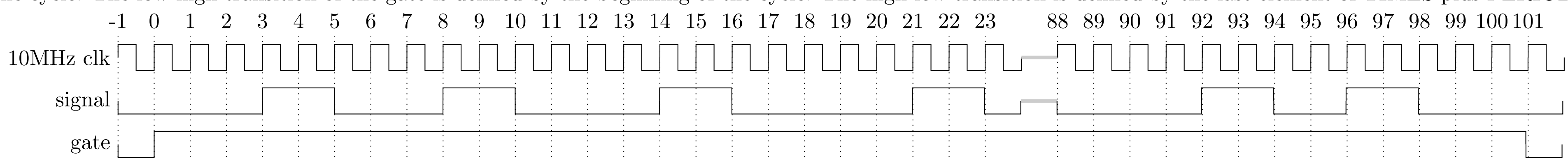


Cycle

During every cycle a digital signal is generated. A gate signal will be high during the sequence and low during idle time. The duration of one cycle is defined by CYCLE and controls the repetition rate.

Sequence $(\mathrm{WIDTH} = 2, \mathrm{PERIOD} = 5, \mathrm{TIMES} = [3, 8, 14, 21, \cdots, 92, 96], \mathrm{MASK} = [1, 3, 2, 5, \cdots, 1, 255])$, make_sequence

The digital signal is generated as a sequence of pulses of a given WIDTH. The timing of the pulses is defined by TIMES that contains tick counts relative to the beginning of the cycle. The low-high transition of the gate is defined by the beginning of the cycle. The high-low transition is defined by the last element of TIMES plus PERIOD.



Pulse train $(\mathrm{WIDTH} = 3, \mathrm{PERIOD} = 10, \mathrm{BURST} = 10)$, make_clock

In pulse train mode the parameter PERIOD and BURST are used generate a sequence of equidistant pulses.

TIMES = [0]