

15 TIMERS

15.1 General

The timer block consists of:

- A fixed clock divider
- Clock prescaling (i.e. a programmable clock divider)
- Two programmable 8-bit timers: *timer0* and *timer1*
- One watchdog timer

The fixed clock divider is also used as a baud rate generator for the asynchronous and synchronous serial ports. The two programmable timers can be cascaded to form one 16-bit timer.

15.2 Timer Registers

Table 15-1 below provides a brief description of the timer registers. For more detailed information, please see chapter 18.4 *Timer Registers*.

Register	Function
R_TIMER_CTRL	A 32-bit write only register to control the operation, clock selection, and divide factor for timer0 and timer1.
R_TIMER_DATA	A 32-bit read only register for the high and low byte of the fixed clock divider, and the current value of timer0 and timer1.
R_TIMER01_DATA	A 16-bit read only register for the combined count value of timer0 and timer1. Typically used when the timers are cascade coupled.
R_TIMER0_DATA	A byte size read only register for the current count value of timer0.
R_TIMER1_DATA	A byte size read only register for the current count value of timer1.
R_WATCHDOG	A 32-bit write only register for enabling and restarting the watchdog timer.
R_CLOCK_PRESCALE	A 32-bit write only register for the divide factor for timer- and serial clock prescaling.
R_TIMER_PRESCALE	A 16-bit write only register for the divide factor for timer clock prescaling.
R_PRESCALE_STATUS	A 32-bit read only register for the current value of the serial- and timer divide value.
R_TIM_PRESC_STATUS	A 16-bit read only register for the current count value of the timer divide factor.

Table 15-1 *Timer registers*

15.3 Clock Prescaling: The Programmable Clock Divider

The programmable clock divider divides a 25MHz clock with a 16-bit value. This 16-bit value ranges from 2 to 65535, and results in a new clock available for the timers that ranges from 12.5MHz down to 381.5Hz. The value for the new available clock is written and read in the internal register R_TIMER_PRESCALE.

15.4 Programmable Timers

15.4.1 Timer Operation

The ETRAX 100LX has two programmable timers, timer0 and timer1, which are configured in the register R_TIMER_CTRL. Each one is an 8-bit binary down counter.

Each timer can be loaded with a divide factor between 1 and 256, in R_TIMER_CTRL. When started, the timer:

- 1 Counts down to 1
- 2 Generates an interrupt (**timer0** or **timer1** respectively)
- 3 Restarts from the programmed divide factor

If the divide factor is changed while the timer is running, it will not take effect until the ongoing count has expired.

Each timer has two mode fields that control its operation:

- timer0: **tm0** and **i0**
- timer1: **tm1** and **i1**

The **tm0** or **tm1** mode field controls the following:

Timer mode:	Operation:
00	Stop the timer and load it with the divide factor
01	Stop the timer and preserve current count value
10	Run
11	Reserved, do not use

Each timer also has a **i0** or **i1** (clear interrupt) mode field. Setting it to 1 clears the interrupt, and setting it to 0 has no effect.

The current count value of both timers can be read in R_TIMER_DATA.

15.5 Timer Input Clock

The timer input clock can be individually selected for each timer. In addition to the programmable clock, the following input frequencies can be selected from the fixed clock divider:

Clock sel:	Nominal frequency: (Note 1)
0	0.3 kHz
1	0.6 kHz
2	1.2 kHz
3	2.4 kHz
4	4.8 kHz
5	9.6 kHz
6	19.2 kHz
7	38.4 kHz
8	57.6 kHz
9	115.2 kHz
10	230.4 kHz
11	460.8 kHz
12	921.6 kHz
13	1843.2 kHz
14	6250.0 kHz

Note 1: The actual frequency is 64 ppm higher than the nominal value (given an exact clock reference input), for all frequencies except 6250 kHz.

15.5.1 Timer0 Input Clock

Timer0 can use the external clock by connecting to the general IO pin **pb6**, or it can use the prescaled clock.

In order to use the prescaled clock (from the programmable clock divider) or the external clock, the **clkse0** field of **R_TIMER_CTRL** must be set to **flexible** (15), as shown in table 15-2 below.

To use the prescaled clock, **presc_ext** must be cleared, and to use the external clock **presc_ext** must be set.

clkse0	presc_ext	clock rate
15	0	timer prescaled clock
15	1	external clock

Table 15-2 *clkse0 settings*

When using the external clock, timer0 will act as a pulse counter with a maximum of 256 pulses. The external clock must be enabled in **R_GEN_CONFIG_II**.

Timer0 can also be cascaded with timer1, resulting in a maximum of 65536 pulses.

15.5.2 Timer1 Input Clock

In order to use the prescaled clock (from the programmable clock divider), **presc_timer1** of **R_TIMER_CTRL** must be set:

presc_timer1	clock rate
0	normal (default)
1	timer prescaled clock

Note 2: Setting **presc_timer1** in **R_TIMER_CTRL** overrides all settings on **clkssel1**.

15.6 Cascade Mode

The two programmable counters can be cascaded to form one 16-bit timer. Cascade mode is selected by setting the **clkssel1** field for timer1 to **cascade0**. The **timer0** interrupt is used to signal the end count of the cascaded counter. The **timer1** interrupt is not used in cascade mode, and should be masked off in **R_IRQ_MASK0_SET**.

When the counters operate in cascade mode, the timer modes, **tm0** and **tm1**, should be set to the same value for both counters with one single write operation.

15.7 Watchdog Timer

When the watchdog timer is started, it generates an NMI if the watchdog is not restarted or stopped within 0.1 s. If it still is not restarted or stopped after an additional 3.3 ms, the watchdog timer resets the chip. The watchdog timer is stopped after reset. The watchdog timer is controlled by the register **R_WATCHDOG**, which contains an **enable** field and a 3-bit **key** value. The effect of writing to the register is described in the table below:

Watchdog	Value written:		Operation:
	To enable:	To key:	
stopped	0	X	No effect
stopped	1	key_val	Start watchdog with key = key_val
started	0	~key (Note 3)	Stop watchdog
started	1	~key (Note 3)	Restart watchdog with key = ~key
started	X	new_key_val	Change key to new_key_val

Note 3: '~' is the bitwise NOT operator.

15.8 Timer Interrupts

There are two timer interrupts, one each for timer0 and timer1.

timer1

This interrupt is set whenever timer1 reaches its terminal count. It is cleared by setting the **i1** field in R_TIMER_CTRL.

timer0

This interrupt is set whenever timer0 reaches its terminal count. It is cleared by setting the **i0** field in timer register R_TIMER_CTRL.

For more information about ETRAX 100LX interrupts, please refer to chapter 17 *Interrupts*.

