Features:

- 64-bit timer with 12-bit prescaler and 8-bit step register
- Compliant with RISC-V privileged specification v1.11
- Configurable number of timers per hart and number of harts
- Can be used with APB (Advanced Peripheral Bus) or TL-UL (Tile link).

Block Diagram:

Hardware interface

Signal Name	Width	Direction	Description
clk_i	1 bit	input	Primary clock
rst_ni	1 bit	input	Active low asynchronous reset
reg_we	1 bit	input	Write enable signal
reg_re	1 bit	input	Read enable signal
reg_addr	9 bit	input	Address for read/write
reg_wdata	32 bit	input	Register write data
reg_be	4 bit	input	Byte selection bits for Write
reg_rdata	32 bit	output	Read data
reg_error	1 bit	output	Error indication
intr_timer_expired_0_0_o	1 bit	output	Interrupt signal

Programmers Guide:

Initialization

The software should configure the step and prescaler bits before enabling the timer control in order to correctly increment the timer value. If software wants to change step or prescaler it should first disable the timer by writing 0x0 to CTRL register and then it can change the step and prescaler by changing the CFG register.The first step should always be to disable the timer by writing 0x0 to the timer CTRL register. Secondly the CFG register should be modified to the desired prescaler and step value. Now the timer compare value should be written to the CMP_LOWER as well as CMP_UPPER if UPPER. If a programmer needs to use only the first 32 bits then you must set the upper 32 bits of the CMP register to 0x0. Moving forward now the CTRL register needs to be written 0x1 in order to activate the timer to start the counter. Now if the programmer needs to reset the timer can do it by activating the hardware reset pin. If there

is a software reset then the software should write 0xFFFF_FFFF to CMP_LOWER as well as CMP_UPPER. And should write 0x0 to all the other registers and the interrupt status register to 0x1.

Register Access

This timer is capable of handling a 64 bit timer compare and 64 bit timer register value, however the registers for timer and compare are divided into two registers of 32 bit for each timer and compare value. As the CPU cannot access 64 bit in a single instruction. Whereas it can split it into two read instructions.Here comes a possibility that the timer may get incremented between the two requests.

Configuration Steps:

1. Disable the timer by writing 0x0 in CTRLregister which is at address 0x0.
2. Now set the prescaler and step value by writing into the CFG register which is at address 0x100.
3. Now set the desired time delay by writing to the CMP_LOWER and CMP_UPPER registers which are at 0x10c and 0x110 respectively.
4. FInally now to activate the timer, write 0x1 to the CTRL register at 0x0.

Register Map:

rv_timer.INTR_STATE0 @0x118

Interrupt status

Reset Default = 0x0,mask 0x1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															IS_0

Bits	Type	Reset	Name	Description

rv_timer.INTR_TEST0 @0x11c

Interrupt test register

Reset Default = 0x0,mask 0x1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															T_0

Bits	Type	Reset	Name	Description
0	wo	x	T_0	Interrupt test for timer

rv_timer.INTR_ENABLE0 @0x114

Interrupt Enable

Reset Default = 0x0,mask 0x1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															IE_0

Bits	Type	Reset	Name	Description
0	rw	0x0	IE_0	Interrupt Enable for timer

rv_timer.COMPARE_UPPER_0 @0x110

Timer Value Upper

Reset Default = 0xffffffff,mask 0xffffffff

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
upper compare value															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
upper compare value															

Bits	Type	Reset	Name	Description
31:0	rw	0xffffffff	value	Timer compare value[63:32]

rv_timer.CFG0 @0x100

Configuration For Hart 0

Reset Default = 0x10000,mask 0xff0fff

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								step							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				Prescale											

Bits	Type	Reset	Name	Description
11:0	rw	0x0	prescale	Prescaler to generate bit
15:12				Reserved
23:16	rw	0x1	step	Incremental value for each tick

rv_timer.COMPARE_LOWER_0 @0x10c

Timer Value lower

Reset Default = 0xffffffff,mask 0xffffffff

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
lower compare value															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
lower compare value															

Bits	Type	Reset	Name	Description
31:0	rw	0xffffffff	value	Timer compare value[31:0]

---

rv_timer.TIMER_V_UPPER0 @0x108

Timer Value Upper

Reset Default = 0x0,mask 0xffffffff

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
upper value															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Upper Value															

Bits	Type	Reset	Name	Description
31:0	rw	0x0	value	Timer Value[63:32]

rv_timer.TIMER_V_LOWER0 @0x104

Timer Value Lower

Reset Default = 0x0,mask 0xffffffff

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
lower Value															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
lower Value															

Bits	Type	Reset	Name	Description
31:0	rw	0x0	value	Timer Value[31:0]

rv_timer.CTRL @0x0

Control register

Reset Default = 0x0,mask 0x1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															active_0

Bits	Type	Reset	Name	Description
0	rw	0x0	active_0	if 1 timer operates