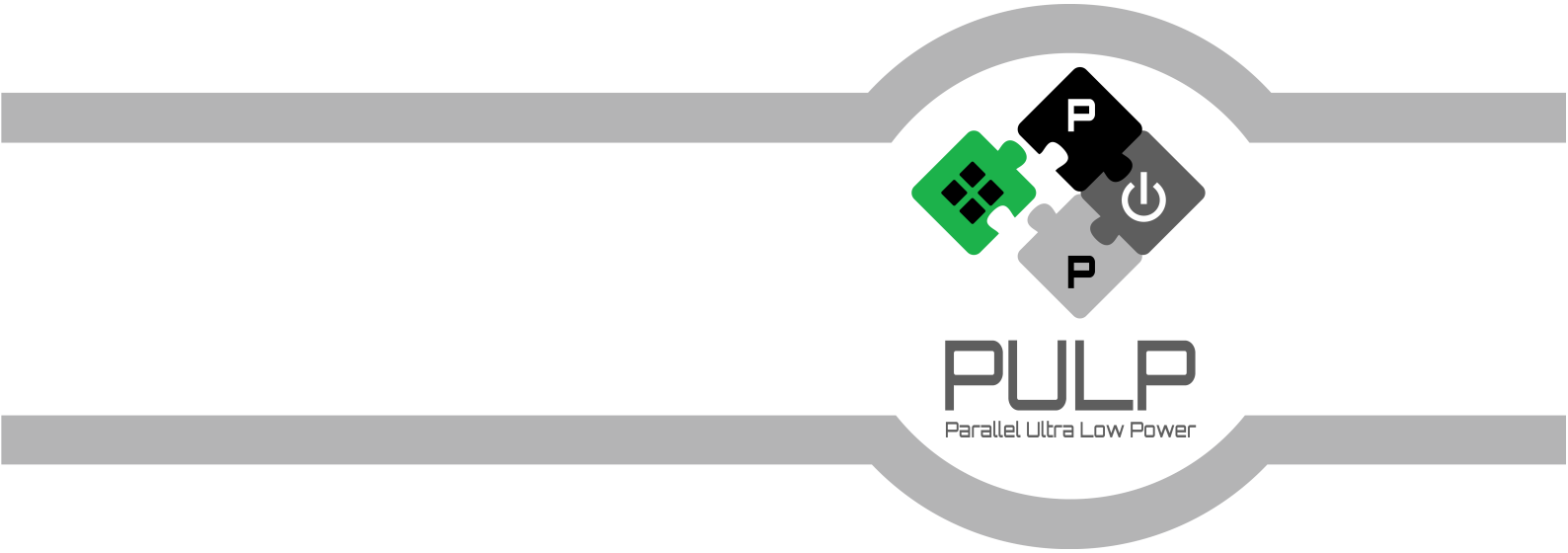


PULP Timer User Manual

April 2016
Revision 1.0
Davide Rossi (davide.rossi@unibo.it)

Micrel Lab and Multitherman Lab
University of Bologna, Italy

Integrated Systems Lab
ETH Zürich, Switzerland

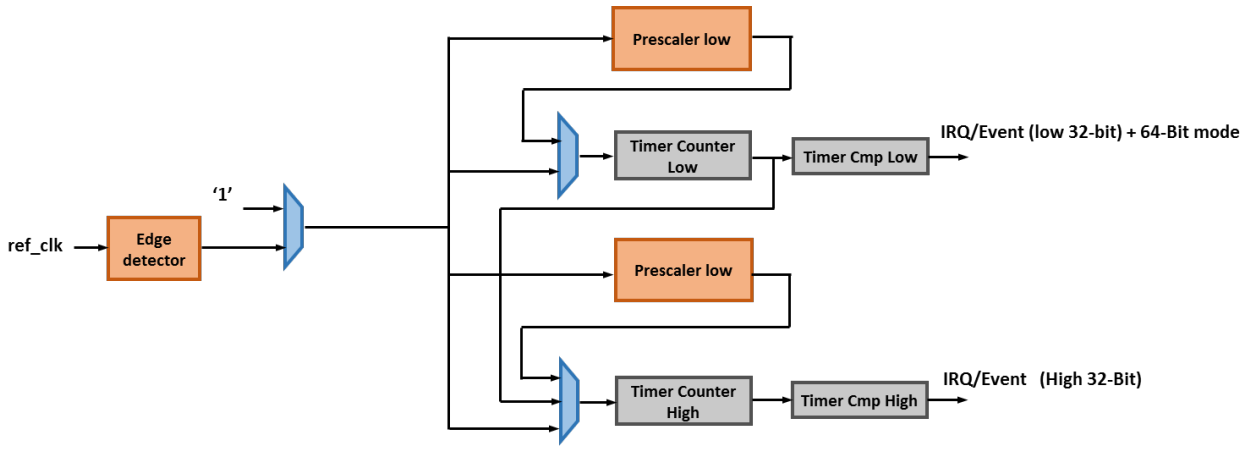
Document Revisions

Rev.	Date	Author	Description
1.0	14/04/16	Davide Rossi	First Draft

Table of Contents

1 Introduction

Timer for PULP platform.



2 Pinout

Port	Width	Direction	Description
Global Signals Bus			
clk_i	1	Input	Global Clock signal
rst_ni	1	Input	Global Reset signal
APB Slave			
req_i	1	Input	Slave Select
ben_i	4	Input	Byte Enable
wen_i	1	Input	Type: 0 = WR, 1 = RD
addr_i	32	Input	Address
wdata_i	32	Input	Input data
id_i	ID_WIDTH	Input	Input ID
gnt_o	1	Output	Grant
r_valid_o	1	Output	Response Valid
r_opc_o	1	Output	Response Opcode
r_id_o	ID_WIDTH	Output	Response ID
r_rdata_o	32	Output	Output data
Interrupts and Events			
event_lo_i	1	Input	Input event for timer 1
event_hi_i	1	Input	Input event for timer 2
irq_lo_o	1	Output	Output IRQ for timer 1

Port	Width	Direction	Description
irq_hi_o	1	Output	Output IRQ for timer 2

3 Register Map

Register Name	Offset	Description
CFG_REG_LOW	0x00	Configuration register for lower 32-bit
CFG_REG_HIGH	0x04	Configuration register for upper 32-bit
TIMER_VALUE_LOW	0x08	Timer value for lower 32-bit
TIMER_VALUE_HIGH	0x0C	Timer value for upper 32-bit
TIMER_CMP_LOW	0x10	Timer comparator value for lower 32-bit
TIMER_CMP_HIGH	0x14	Timer comparator value for upper 32-bit
TIMER_START_LOW	0x18	Additional Start timer register for lower 32-bit
TIMER_START_HIGH	0x1C	Additional Start timer register for upper 32-bit
TIMER_RESET_LOW	0x20	Additional Reset timer register for lower 32-bit
TIMER_RESET_HIGH	0x24	Additional Reset timer register for upper 32-bit

Table 1: Timer Register Map

Bit #	R/W	Description
31	R/W	**64-bit mode:** enables 64-bit mode 0: 64-bit mode disabled. Two 32-bit counters are available 1: 64-bit mode disabled. One 32-bit counter is available
30:9		**Reserved**
15:8	R/W	**Prescaler value:** value of prescaler clock division factor: FreqTimer = 1/(1+PrescalerValue[7:0])
7	R/W	**Clock Source**: 0: FLL / FLL + prescaler 1: 32 KHz ref clock
6	R/W	**Prescaler enable:** Enables prescaler 0: Prescaler is disabled 1: Prescaler is enabled
5	R/W	**One shot:** One shot feature 0: Keeps counting when TIMER_CMP_LOW is reached 1: Disables timer when TIMER_CMP_LOW is reached

Bit #	R/W	Description
4	R/W	**CMP&CLR:** Compare and clear feature 0: Timer keeps going on after reaching TIMER_CMP_LOW 1: Set timer to 0 when TIMER_CMP_LOW is reached
3	R/W	**IEM:** Input event mask 0: Input event is disabled 1: Input event is enabled
2	R/W	**IRQ Enable:** Rise an interrupt request when TIMER_CMP_LOW is reached 0: IRQ is enabled 1: IRQ is disabled
1	W	**RESET:** Resets timer value (this field is cleared after timer reset)
0	R/W	**ENABLE:** Enables Timer 0: Timer idle 1: Timer active

Table 2: CFG_REG_LOW

Bit #	R/W	Description
31:7		Reserved
7	R/W	**Enable Ref Clock**: 0: FLL / FLL + prescaler 1: 32 KHz ref clock
6	R/W	**Prescaler enable:** Enables prescaler 0: Prescaler is disabled 1: Prescaler is enabled
5	R/W	**One shot:** One shot feature 0: Keeps counting when TIMER_CMP_HIGH is reached 1: Disable timer when TIMER_CMP_HIGH is reached
4	R/W	**CMP&CLR:** Compare and clear feature 0: Timer keeps going on after reaching TIMER_CMP_HIGH 1: Set timer to TIMER_VALUE_HIGH when TIMER_CMP_HIGH is reached
3	R/W	**IEM:** Input event mask 0 Input event is disabled 1 Input event is enabled
2	R/W	**IRQ Enable:** Rise an interrupt request when TIMER_CMP_HIGH is reached 0: IRQ Off 1: IRQ On
1	W	**RESET:** Resets timer value (this field is cleared after timer reset)

Bit #	R/W	Description
0	R/W	**ENABLE:** Enables Timer 0: Timer active 1: Timer idle

Table 3: CFG_REG_HIGH