**COUNTER 64 BIT**

Catalog

[1. OVERVIEW 1](#_Toc29413)

[1.1 Introduction: 1](#_Toc3427)

[1.2 fuction: 2](#_Toc7803)

[1.3 feature: 2](#_Toc3176)

[2.BLOCK DIAGRAM 2](#_Toc22069)

[3.I/O LIST 2](#_Toc6273)

[4. FUNCTION DESCRIPTION 3](#_Toc9009)

[4.1 APB Slave Block 3](#_Toc15448)

[4.1.1 APB slave Block 3](#_Toc31597)

[4.1.2 Detail logic diagram 3](#_Toc15363)

[4.1.3. Description 3](#_Toc26734)

[4.1.4. Waveform 4](#_Toc8598)

[4.2 Register Block 5](#_Toc24599)

[4.2.1 Block register 5](#_Toc14104)

[4.2.2 Detail logic diagram 6](#_Toc18746)

[4.2.3 Description 7](#_Toc8534)

[4.2.4 Waveform 7](#_Toc12711)

[4.3 Counter control Block 8](#_Toc10802)

[4.3.1 Block counter control 8](#_Toc10697)

[4.3.2 Detail logic diagram 8](#_Toc32334)

[4.3.3 Description 9](#_Toc6670)

[4.3.4 Waveform 9](#_Toc338)

[4.4.1 Counter 64 bit Block 9](#_Toc18606)

[4.4.1 Counter 64 bit block 9](#_Toc13592)

[4.4.2 Detail logic diagram 10](#_Toc14988)

[4.4.3 Description 10](#_Toc21381)

[4.4.4 Waveform 10](#_Toc12363)

[4.5 INTERRUPT BLOCK 11](#_Toc30680)

[4.5.1 Interrupt block 11](#_Toc18630)

[4.5.2 Detail logic diagram 11](#_Toc23470)

[4.5.3 Description 11](#_Toc23692)

[4.5.4 Waveform 11](#_Toc3358)

[5. REGISTER DESCRIPTION 11](#_Toc536)

[Detail register : 12](#_Toc28295)

[\* TCR : 12](#_Toc26837)

[\* TDR0 : 12](#_Toc5646)

[\* TDR1 : 13](#_Toc26924)

[\* TCMP0 : 13](#_Toc4579)

[\* TCMP1 : 13](#_Toc23096)

[\* TIER : 14](#_Toc32702)

[\* TISR : 14](#_Toc460)

[\* THCSR : 15](#_Toc6454)

1. **OVERVIEW**
   1. **Introduction:**

▪ Timer is an essential module for every chip.

▪ This is used to generate accurate timing interval or controlling the timing of various operations within the circuit. Timer can be used in various application: pulse generation, delay generation, event generation, PWM generation, Interrupt generation ….

▪ In this project, a timer module is customized from CLINT module of industrial RISC-V architecture. It is used to generate interrupt based on user settings.

▪ The spec of CLINT can be referred at:

<https://chromitem-soc.readthedocs.io/en/latest/clint.html>

**1.2 fuction:**

▪ This is timer IP can be used counter(64bit).

▪ [Advanced level] Halted mode.

▪ Timer interrupt.

▪ Counting mode.

▪ APB slave/register.

**1.3 feature:**

▪ count up(64-bit).

▪ Address space: 4KB (0x4000\_1000 – 0x4000\_1FFF)

▪ Register set is configured via APB bus (IP is APB slave).

▪ Support timer interrupt (can be enabled or disabled).

▪ System clock frequency is 200 MHz. Timer uses active low async reset.

▪ Counter can be counted based on system clock or divided up to 256.

**▪ Standard level:**

o Only support APB 32-bit transfer with no wait states and no error handling

**▪ Advanced level:**

o Support wait state (1 cycle is enough) and error handling

o Support byte access

o Support halt (stop) in debug mode

**2.BLOCK DIAGRAM**

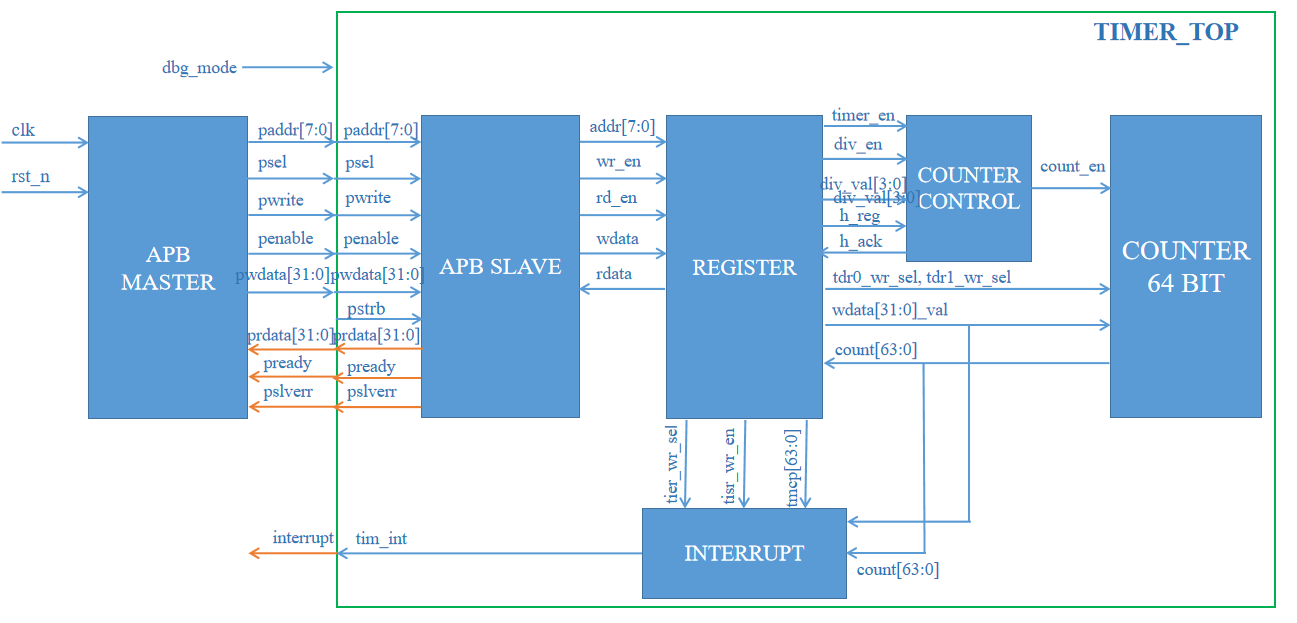


Figure 1: Block Diagram

**3.I/O LIST**

|  |  |  |  |
| --- | --- | --- | --- |
| sys\_clk | 1 | input | Clock. The rising edge of PCLK times all transfers on the APB |
| sys\_rst\_n | 1 | input | Reset. The APB reset signal is active LOW. This signal is normally connected directly to the system bus reset signal. |
| tim\_psel | 1 | input | Select. Active high select signal. The APB bridge unit generates this signal to each peripheral bus slave such as: TDR, TCR, TCMP0, TCMP1, TIER. TISR and THCSR |
| tim\_pwrite | 1 | input |  |
| tim\_penable | 1 | input | Enable. This signal indicates the second and subsequent cycles of an APB transfer. |
| tim\_paddr | 8 | input | Address. This is the APB address bus. |
| tim\_pwdata | 8 | input | Write data. |
| tim\_prdata | 8 | output | Read data |
| tim\_pstrb | 1 | input | Signal which bytes need to be written and which not. |
| tim\_pready | 1 | output | Ready. The slave uses this signal to extend an APB transfer. |
| tim\_pslverr | 1 | output | This signal indicates a transfer failure. |
| tim\_int | 1 | output |  |
| dbg\_mode | 1 | input | Debug mode |

1. **FUNCTION DESCRIPTION**

**4.1 APB Slave Block**

4.1.1 APB slave Block

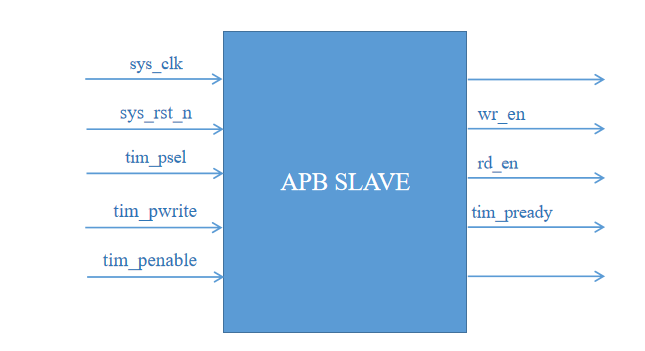


Figure 2 Register block

4.1.2 Detail logic diagram

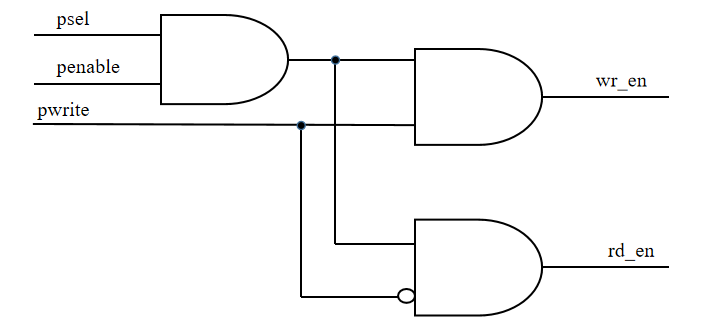


Figure 3 Logic diagram output APB Slave

4.1.3. Description

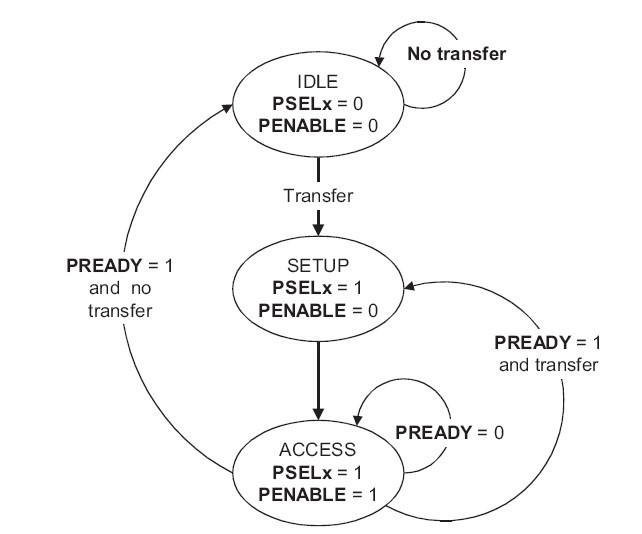


Figure 4 FSM\_STATE

- The APB Slave performs read/write operations based on the coordination of the PSEL, PENABLE, and PWRITE signals:

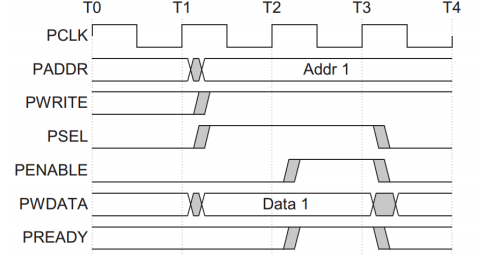


Figure 5 Write data

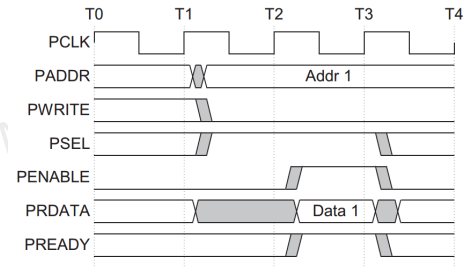


Figure 6 Read data

4.1.4. Waveform

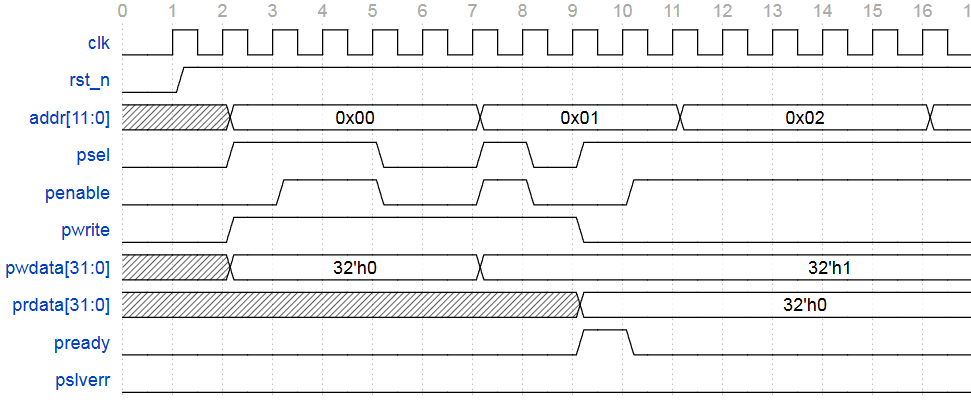


Figure 7 Waveform APB RW

**4.2 Register Block**

4.2.1 Block register

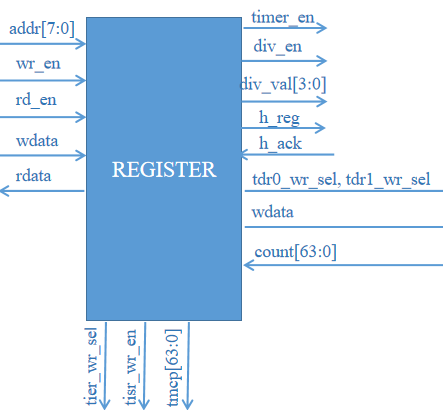


Figure 8 Block register

**Table : Port I/O Block Register**

|  |  |  |  |
| --- | --- | --- | --- |
| addr[11:0] | RW | input | address bus |
| wr\_en | RW | input | 1: module access for writing  0: module access for reading |
| rd\_en | RW | input | 1: module access for reading  0: module access for writing |
| wdata[31:0] | RW | input | register write data |
| rdata[31:0] | RO | output | Clock input |
| timer\_en | RW | output | 1: start counting  0: stop counting |
| div\_en | RW | output | External clock or pulse input |
| div\_val[3:0] | RW | output |  |
| tdr0\_wr\_sel | RW | output | Select signal used to write to timer tdr0 |
| tdr1\_wr\_sel | RW | output | Select signal used to write to timer tdr1 |
| wdata\_en[31:0] | RW | output | register write data |
| count[63:0] | RW | input |  |
| tcmp[63:0] | RW | output | Timer compare |
| tier\_wr\_sel | RW | output | Select signal used to write to timer tier |
| tier\_wr\_sel | RW | output | Select signal used to write to timer tisr |
| h\_reg | RW | output |  |
| h\_ack | RO | input |  |

4.2.2 Detail logic diagram

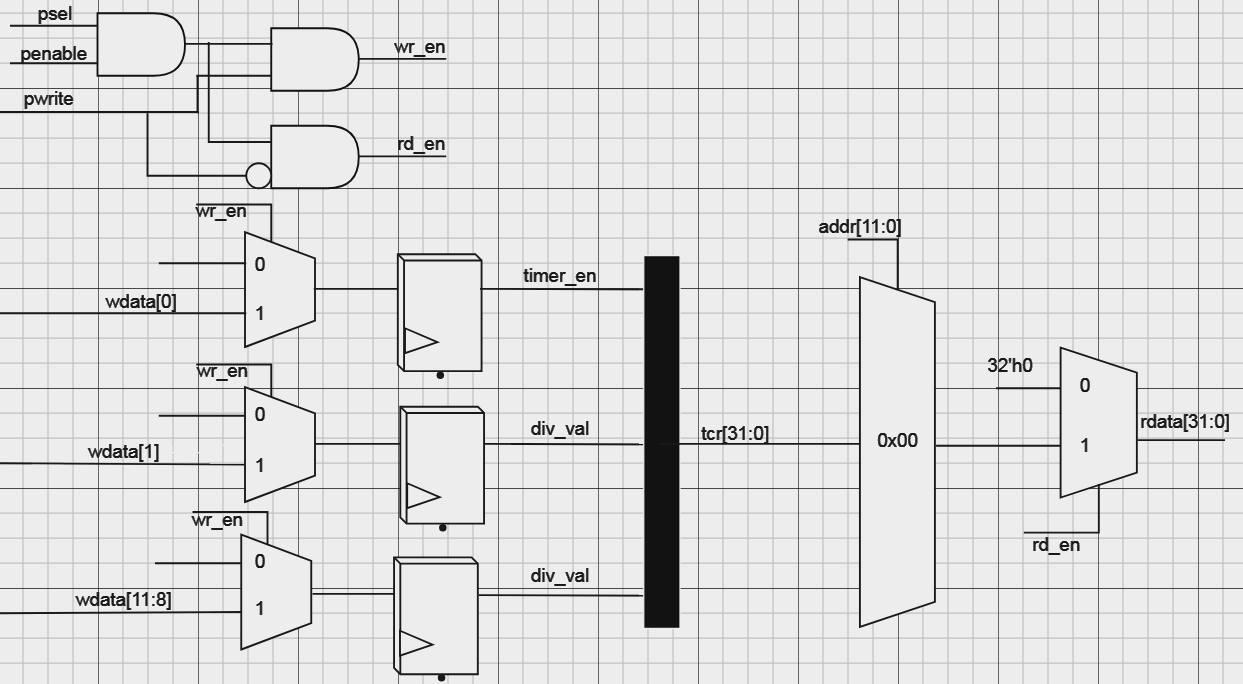


Figure 9 TCR[31:0] Register

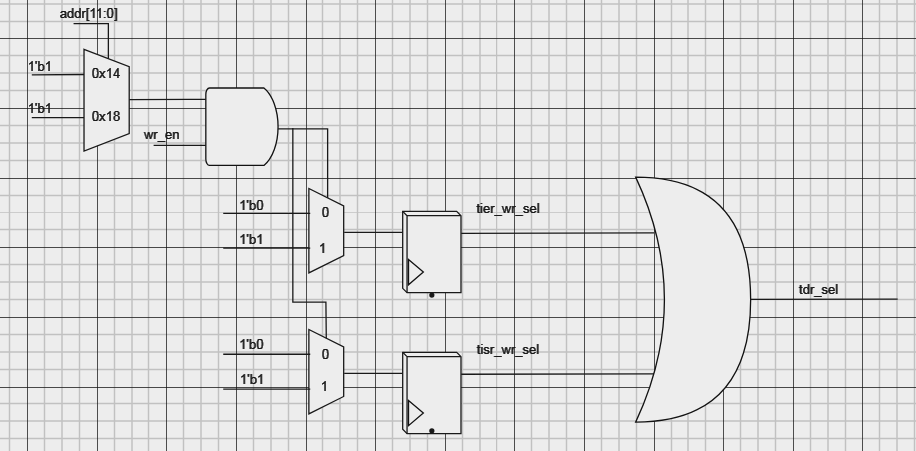


Figure 10 TDR0, TDR1 register

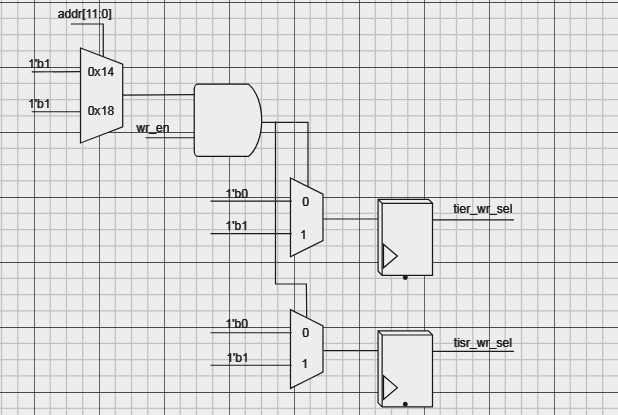


Figure 11 TIER, TISR\_WR\_SEL

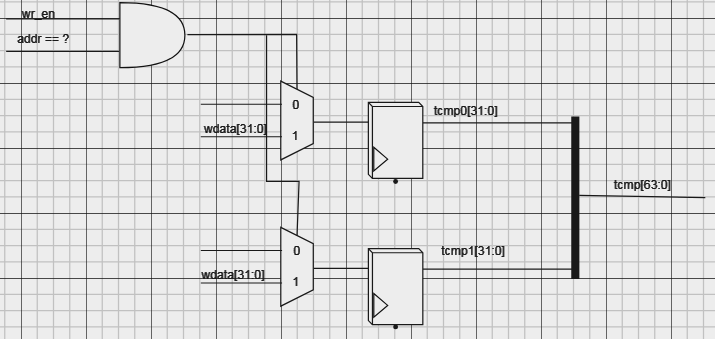


Figure 12 TCMP Register

4.2.3 Description

- TCR timer is the register used to control the counting operation of the counter:

timer\_en[0] = 1 starts counting.

timer\_en[0] = 0 stops counting.

div\_en = 1 counts according to the external clock pulse.

div\_val[3:0] pulse divider.

**-** TDR is data timer(tdr={tdr1, tdr0})

**-**

4.2.4 Waveform

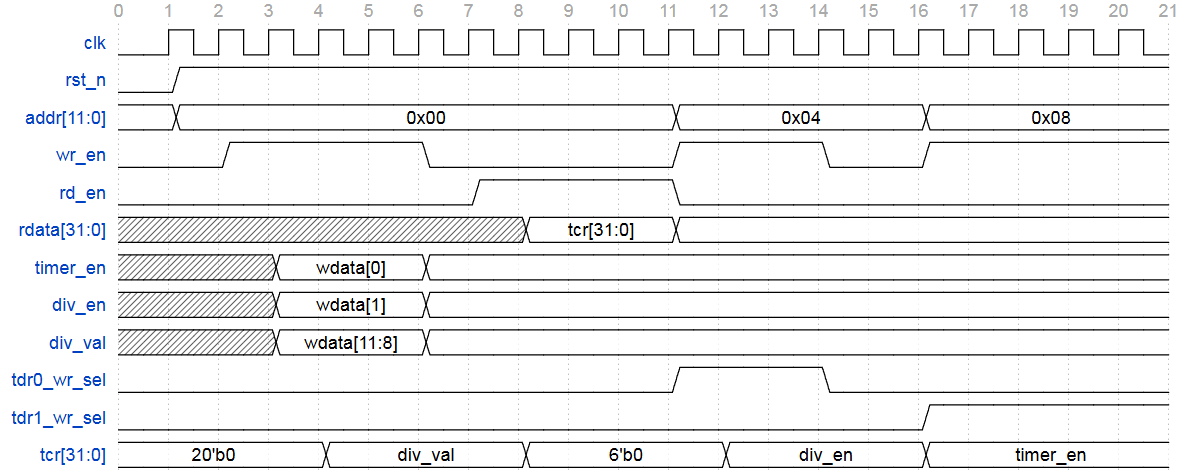


Figure 13 Waveform RW TCR register

**4.3 Counter control Block**

4.3.1 Block counter control

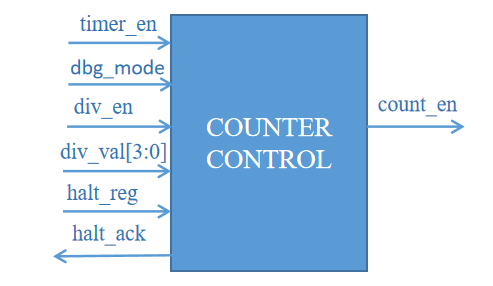


Figure 14 Counter control Block

4.3.2 Detail logic diagram

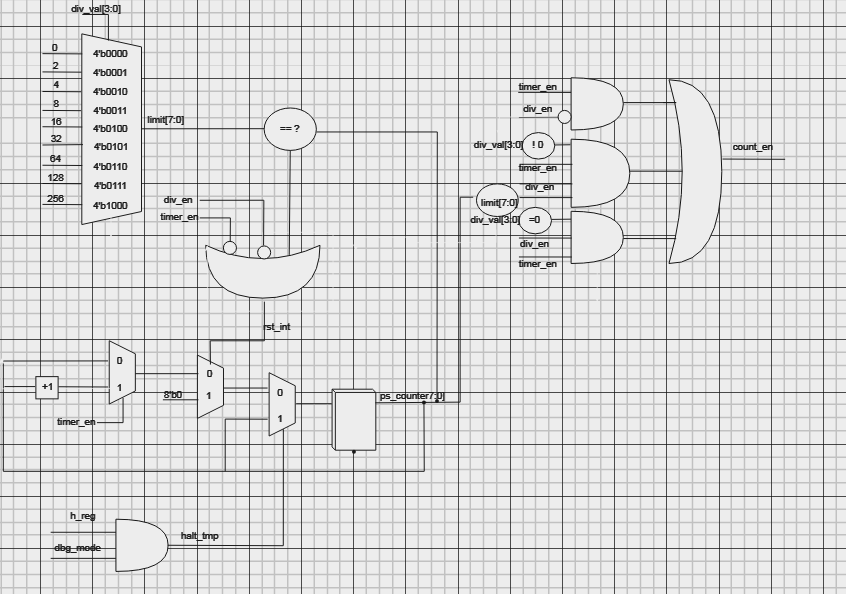


Figure 15 Logic diagram Counter Control Block

4.3.3 Description

**-** The function block has the function of selecting the counting mode for the Counter 64 bit block.

- Timer\_en =1 Counter can be counted based on system clock.

- div\_en =1 divided up to 256(div\_val[3:0]).

- halt\_reg and dbg\_mode when high(1) will immediately stop the running counter mode

4.3.4 Waveform

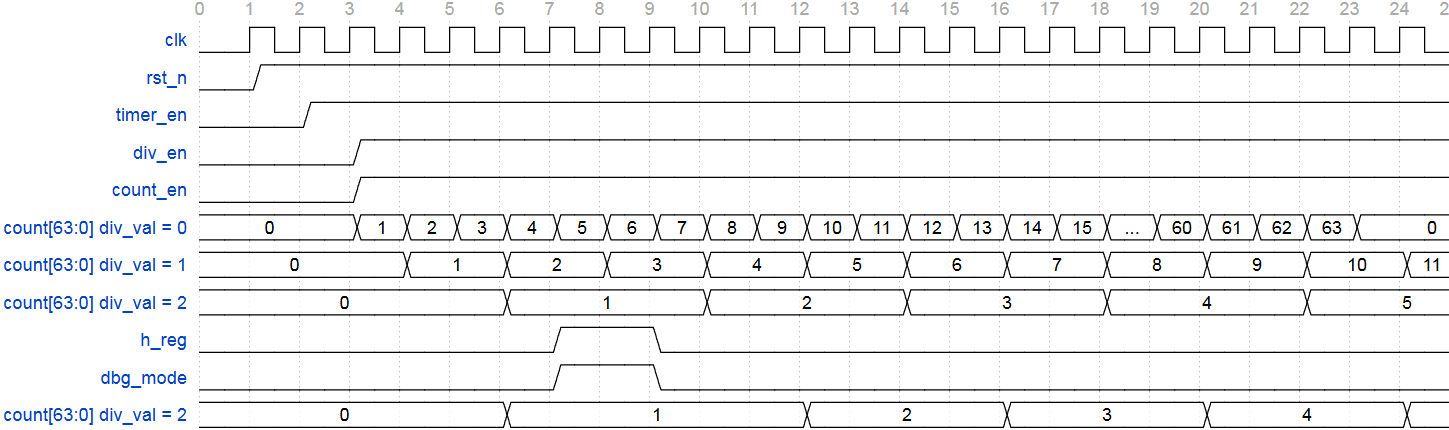


Figure 16 Counting mode

**4.4.1 Counter 64 bit Block**

4.4.1 Counter 64 bit block

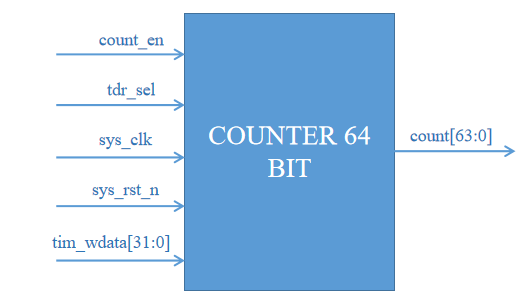


Figure 17 Counter 64 Bit Block

4.4.2 Detail logic diagram

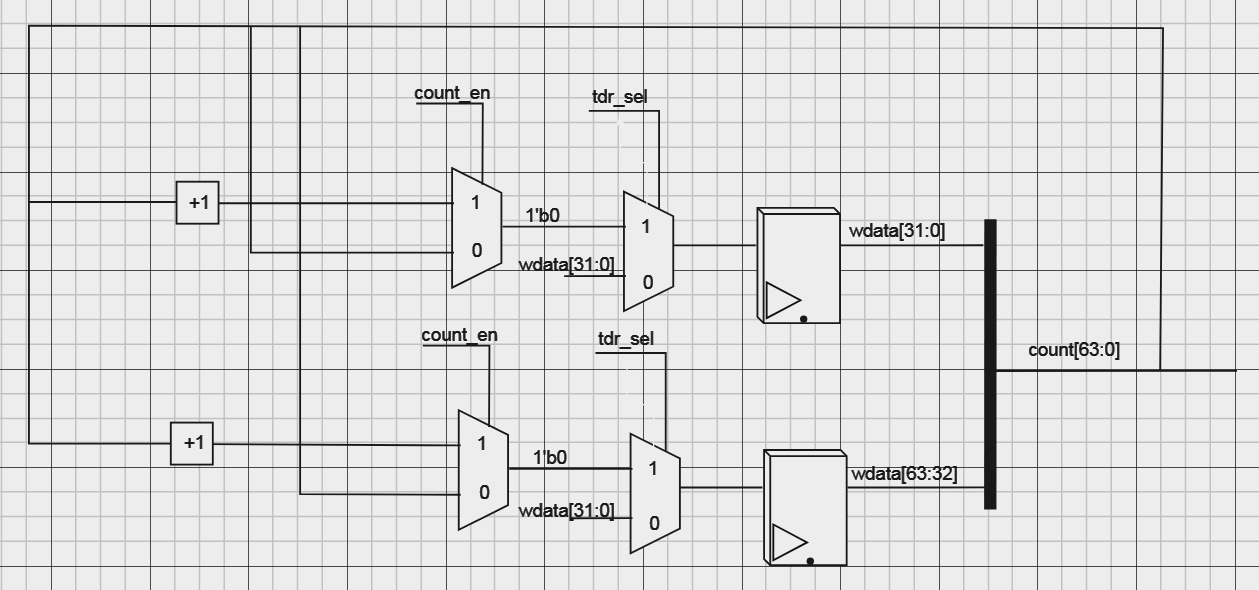


Figure 17 Logic diagram counter Block

4.4.3 Description

- wr\_en & (addr == 0x00) is enabled(1) then:

+ timer\_en <= wdata[0]

+ div\_en <= wdata[1].

+ div\_val <= wdata[11:8].

4.4.4 Waveform

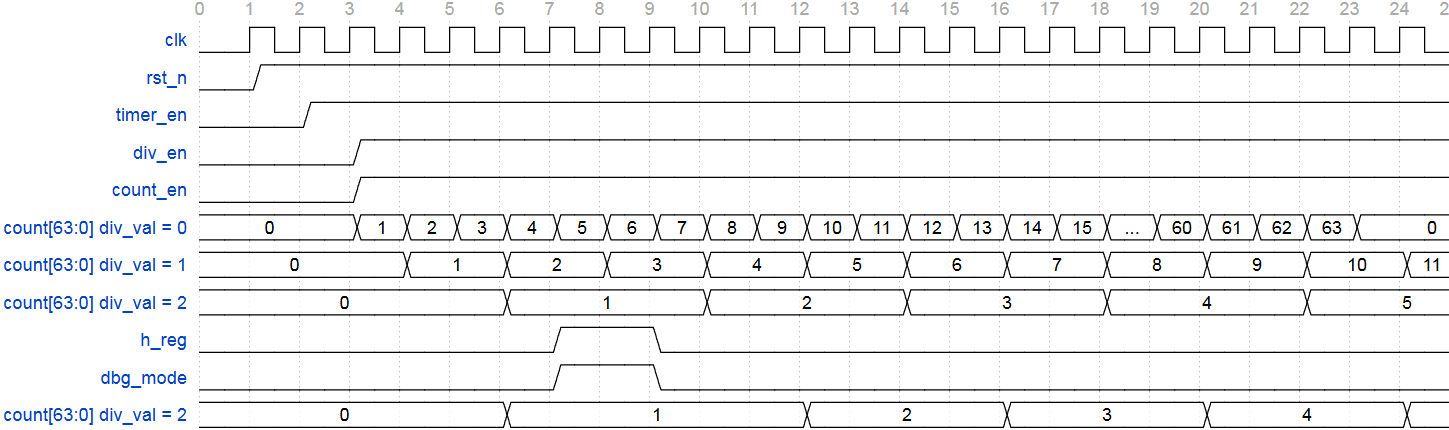


Figure 18 Waveform Counter

**4.5 INTERRUPT BLOCK**

4.5.1 Interrupt block

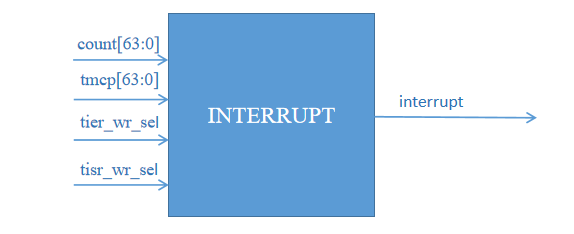


Figure 19 Interrupt Block

4.5.2 Detail logic diagram

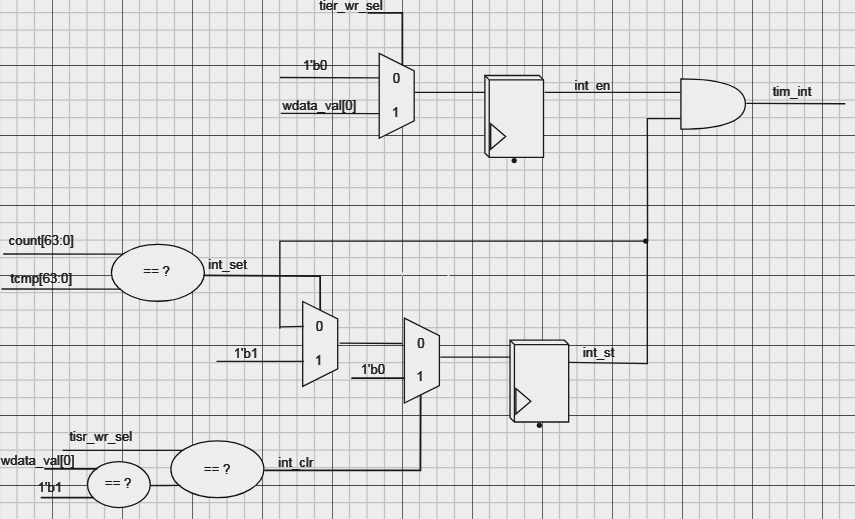


Figure 20 Logic diagram Interrupt Block

4.5.3 Description

- int\_set is enabled when the count value is equal to the compare value (tmcp).

- int\_clr is enabled (1) when the tisr\_wr\_sel signal is active and wdata[0]\_val is equal to , used to reset the int\_st signal to 0.

- tim\_int is asserted when int\_en and int\_set both assert.

- Once asserted, the timer interrupt (tim\_int) remains unchange until it is cleared by writing 1 to TISR.int\_st bit or the interrupt is disabled.

4.5.4 Waveform

1. **REGISTER DESCRIPTION**

Base Address :

|  |  |  |
| --- | --- | --- |
| **Offset** | **Abbreviation** | **Resister name** |
| 0x00 | TCR | Timer Control Register |
| 0x04 | TDR0 | Timer Data Register 0 |
| 0x08 | TDR1 | Timer Data Register 1 |
| 0x0C | TCMP0 | Timer Compare Register 0 |
| 0x10 | TCMP1 | Timer Compare Register 1 |
| 0x14 | TIER | Timer Interrupt Enable Register |
| 0x18 | TISR | Timer Interrupt Status Register |
| 0x1C | THCSR | Timer Halt Control Status Register |
| Others | Reserved |  |

**Detail register :**

**\* TCR :**

▪ **address** : 0x00

▪ **size** : 32-bit

▪ **description** : Controls timer operation, including starting, stopping.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **bit** | **name** | **type** | **default value** | **description** |
| 11:8 | Div\_val | RW | 4’b0001 | • 4’b0000: Counting speed is not divided  • 4’b0001: Counting speed is divided by 2 (default)  • 4’b0010: Counting speed is divided by 4  • 4’b0011: Counting speed is divided by 8  • 4’b0100: Counting speed is divided by 16  • 4’b0101: Counting speed is divided by 32  • 4’b0110: Counting speed is divided by 64  • 4’b0111: Counting speed is divided by 128  • 4’b1000: Counting speed is divided by 256 |
| 7:2 | Reserved | RO | 6’b0 | Reserved |
| 1 | div\_en | RW | 1’b0 | Counter control mode enable.  • **0**: Disabled. Counter counts with normal speed based on system clock  • **1**: Enabled. The couting speed of counter is controlled based on div\_val  Note: user must not change div\_en while timer\_en is High  (\*): add hardware logic to ensure div\_en is prohibited to change when timer\_en is High. Access is error response in this case. |
| 0 | timer\_en | RW | 1’b0 | • **0**: Disabled. Counter does not count.  • **1**: Enabled. Counter starts counting.  (\*) timer\_en changes from H->L will initialize the TDR0/1 to their initial value |

**\* TDR0 :**

▪ **address** : 0x04

▪ **size** : 32-bit

▪ **description** : Stores the current count value or initial value for Timer 0.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **bit** | **name** | **type** | **default value** | **description** |
| 31:0 | TDR0 | RW | 32’h0000\_0000 | • Lower 32-bit of 64-bit counter.  H -> L : clear |

-----------------------------------------------------------------------------------------------

**\* TDR1 :**

▪ a**ddress** : 0x08

▪ **size** : 32-bit

▪ **description** : Stores the current count value or initial value for Timer 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **bit** | **name** | **type** | **default value** | **description** |
| 31:0 | TDR1 | RW | 32’h0000\_0000 | • Upper 32-bit of 64-bit counter.  H -> L : clear |

-----------------------------------------------------------------------------------------------

**\* TCMP0 :**

▪ **address** : 0xC

▪ **size** : 32-bit

▪ **description** : Set the compare value for Timer 0, trigger an interrupt when this value is reached

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **bit** | **name** | **type** | **default value** | **description** |
| 31:0 | TCMP0 | RW | 32’hFFFF\_FFFF | • The TCMP0 register contains the low 32 bits of the 64 bits of the compare value.  • The interrupt will be triggered when the count value equals the lower 32bit value of this timer.  **EX:** Compare Value (64-bit) = 0x1234\_5678\_9ABC\_DEF0  TCMP0: 0x9ABC\_DEF0  TCMP1: 0x1234\_5678 |

**\* TCMP1 :**

▪ **address** : 0x10

▪ **size** : 32-bit

▪ **description** : Set the compare value for Timer 1, trigger an interrupt when this value is reached

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **bit** | **name** | **type** | **default value** | **description** |
| 31:0 | TCMP1 | RW | 32’hFFFF\_FFFF | • The TCMP0 register contains the upper 32 bits of the 64 bits of the compare value.  • The interrupt will be triggered when the count value equals the upper 32bit value of this timer.  **EX:** Compare Value (64-bit) = 0x1234\_5678\_9ABC\_DEF0  TCMP0: 0x9ABC\_DEF0  TCMP1: 0x1234\_5678 |

**\* TIER :**

▪ **address** : 0x14

▪ **size** : 32-bit

▪ **description** : Controls the interrupt or non-interrupt operation from the timer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **bit** | **name** | **type** | **default value** | **description** |
| 31:1 | Reserved | RO | 31’b0 | Reserved |
| 0 | int\_en | R/W | 1’b0 | **0**: Timer interrupt is disabled.  **1**: Timer interrupt is enabled.  • When this bit is 0, no timer interrupt is output.  • When this bit is 1, timer interrupt can be output when reaching  trigger condition.  … |

**\* TISR :**

▪ **address** : 0x18

▪ **size** : 32-bit

▪ d**escription** : Indicates the status of the timer interrupt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **bit** | **name** | **type** | **default value** | **description** |
| 31:1 | Reserved | RO | 31’b0 | Reserved |
| 0 | int\_st | R/W1C | 1’b0 | Timer interrupt trigger condition status bit (interrupt pending bit)  **0**: the interrupt trigger condition does not occur.  **1**: the interrupt trigger condition occurred.  Write 1 when this bit is 1 to clear it  Write 0 when this bit is 1 has no effect  Write to this bit when it is 0 has no effect.  **Note:** When interrupt trigger condition occurred (counter reached  compare value), counter continues to count normally |

**\* THCSR :**

▪ **address** : 0x1C

▪ **size** : 32-bit

▪ d**escription** :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **bit** | **name** | **type** | **default value** | **description** |
| 31:2 | reserved | RO | 30’h0 | reserved |
| 1 | halt\_ack | RO | 1’b0 |  |
| 0 | halt\_reg | RW | 1’b0 |  |