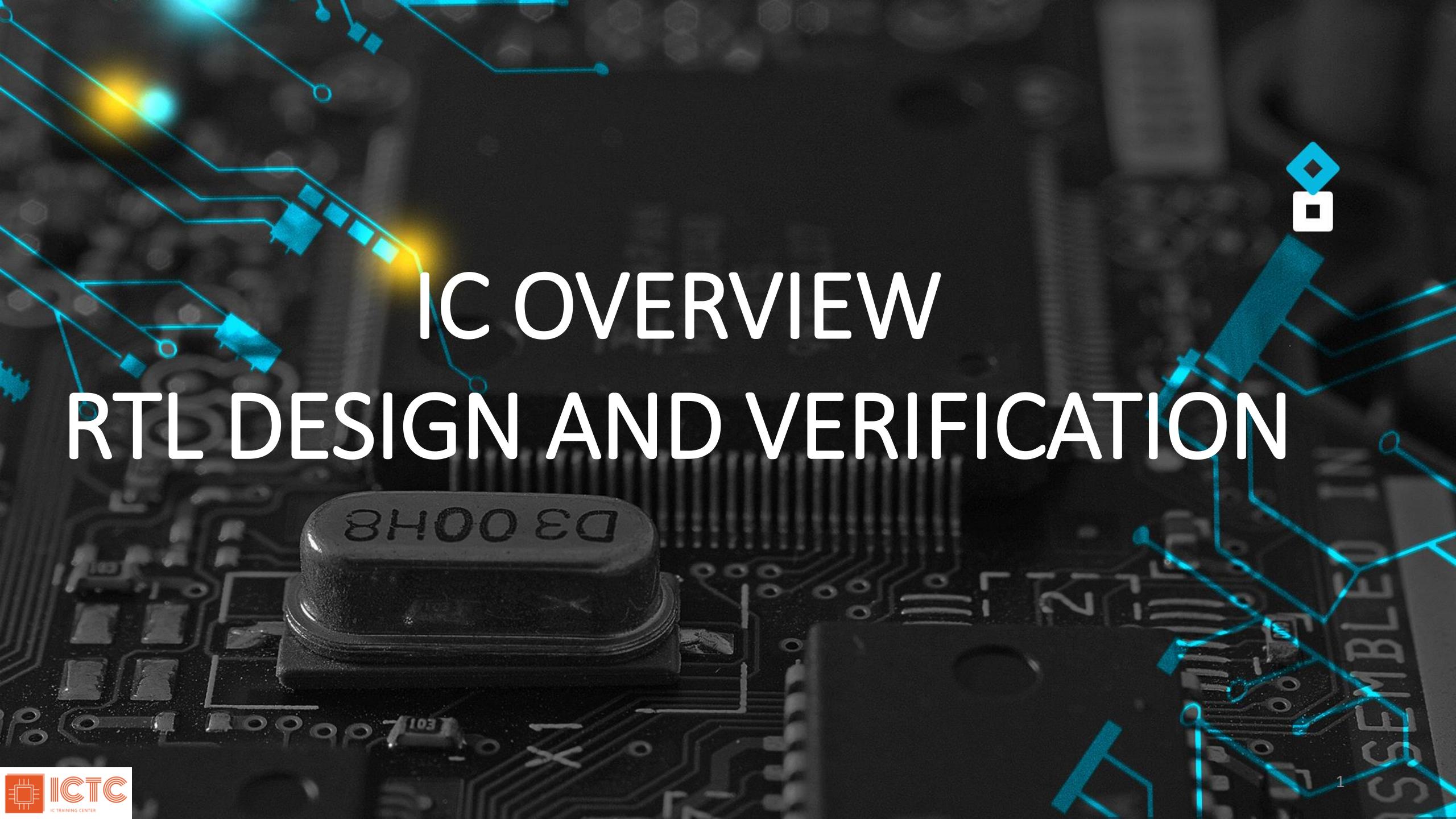
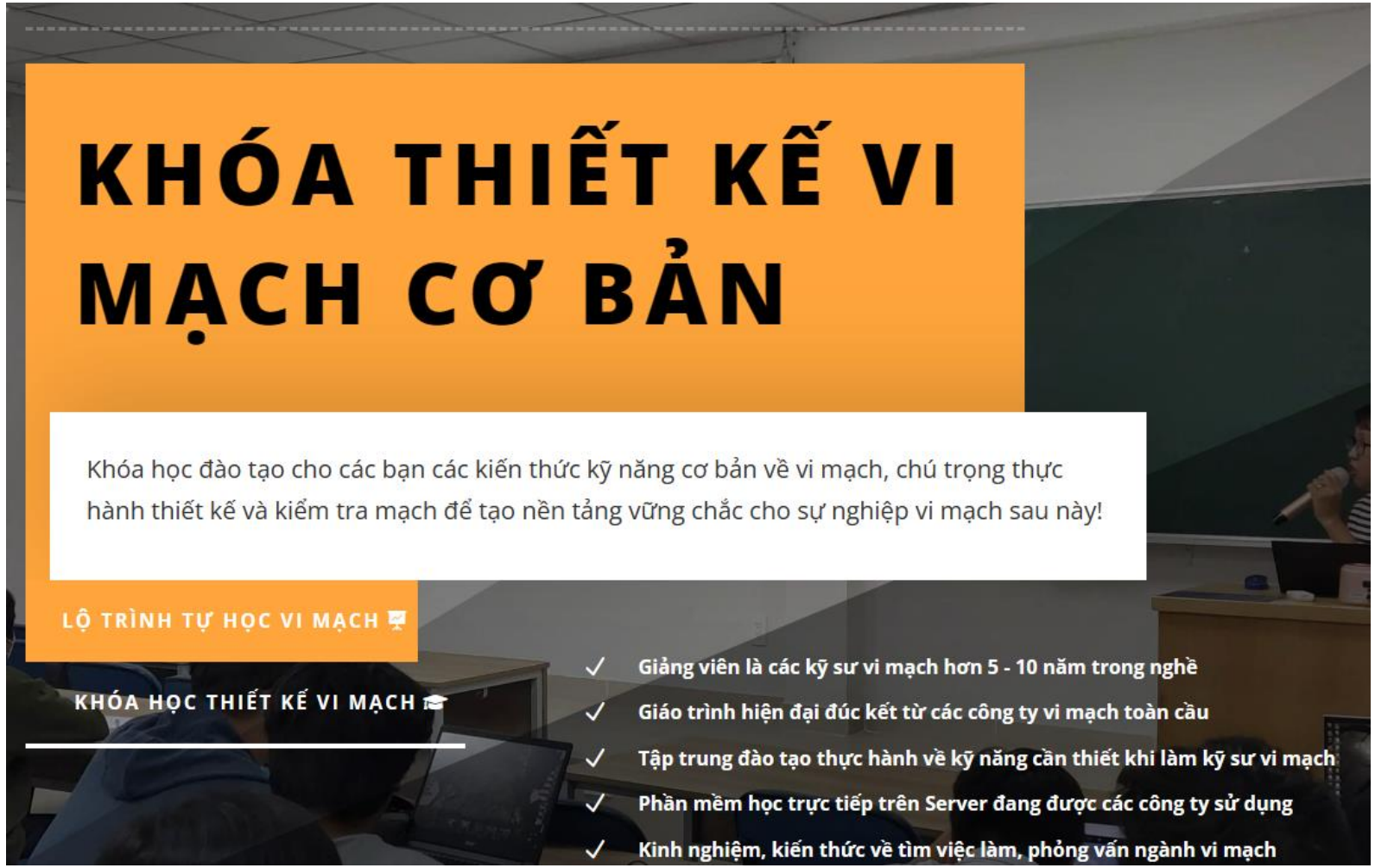


IC OVERVIEW
RTL DESIGN AND VERIFICATION

COURSE INTRODUCTION

[Khóa Học Thiết Kế Vi Mạch Cơ Bản - Trung Tâm Đào Tạo Thiết Kế Vi Mạch ICTC]



COURSE INTRODUCTION

SUMMARY

HOMEWORK

QUESTION

SELF-LEARNING

FINAL PROJECT

1. Requirement Specification

2. Requirement for Submission

REGISTER ITEM LIST

We finally reach here!!! Awesome work !!!

Let’s do the last practice - the most challenge and interesting thing in this course!!!

THE FINAL PROJECT – TIMER IP

Final Project Requirement

- 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

Final Project Requirement

The timer has following features:

- 64-bit count-up.
- Address space: 4KB (0x4000_1000 – 0x4000_1FFF)
- Register set is configured via APB bus (IP is APB slave).
- Standard level:
  - Only support APB 32-bit transfer with no wait states and no error handling
- Advanced level:
  - Support wait state (1 cycle is enough) and error handling
  - Support byte access
  - Support halt (stop) in debug mode
- 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.
- Support timer interrupt (can be enabled or disabled).

DETAIL FUNCTIONAL DESCRIPTION

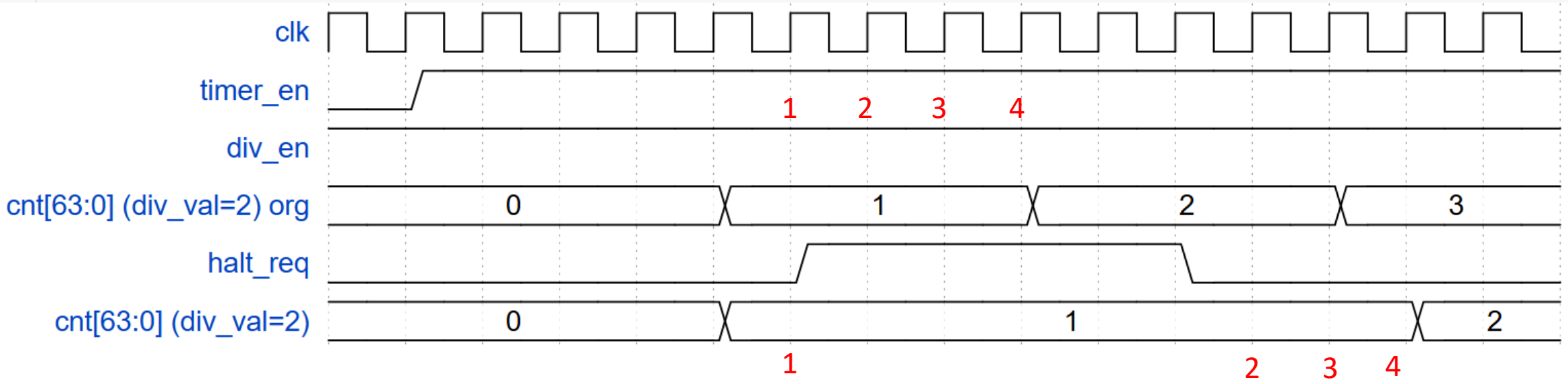
Counter

- Counter
  - 64-bit count-up.
  - Counting mode:
    - Default mode: counter's speed is same as system clock.
    - Control mode: when enabled by writing 1 to TCR.div_en bit, the counter's speed is determined by the divisor value set in TCR.div_val
  - Counter continues counting when interrupts occurs.
  - Counter continues counting when overflow occurs.
  - [Standard only]: when timer_en is H->L, counter needs to be initialized by software. When timer_en is L->H again, timer can work normally.
  - [Advanced] support halted mode describe in next page.
  - [Advanced] When timer_en changes from High to Low, the counter is cleared to its initial value. When timer_en is L->H again, timer can work normally.
  - Note: the div_en and div_val is not related to frequency divisor (clock divider). Those settings only control the counter when to count.

DETAIL FUNCTIONAL DESCRIPTION

Halted

- [Advanced level] Halted mode
  - Counter can be halted (stopped) in debug mode when both below conditions occur:
    - Input debug_mode signal is High,
    - THCR.halt_req is 1.
  - THCSR.halt_ack is 1 after a halt request indicates that the request is accepted.
  - After halted, counter can be resumed to count normally when clearing the halt request to 0.
  - The period of each counting number needs to be same when halt and not halt as described in the below waveform example (div_val=2).



DETAIL FUNCTIONAL DESCRIPTION

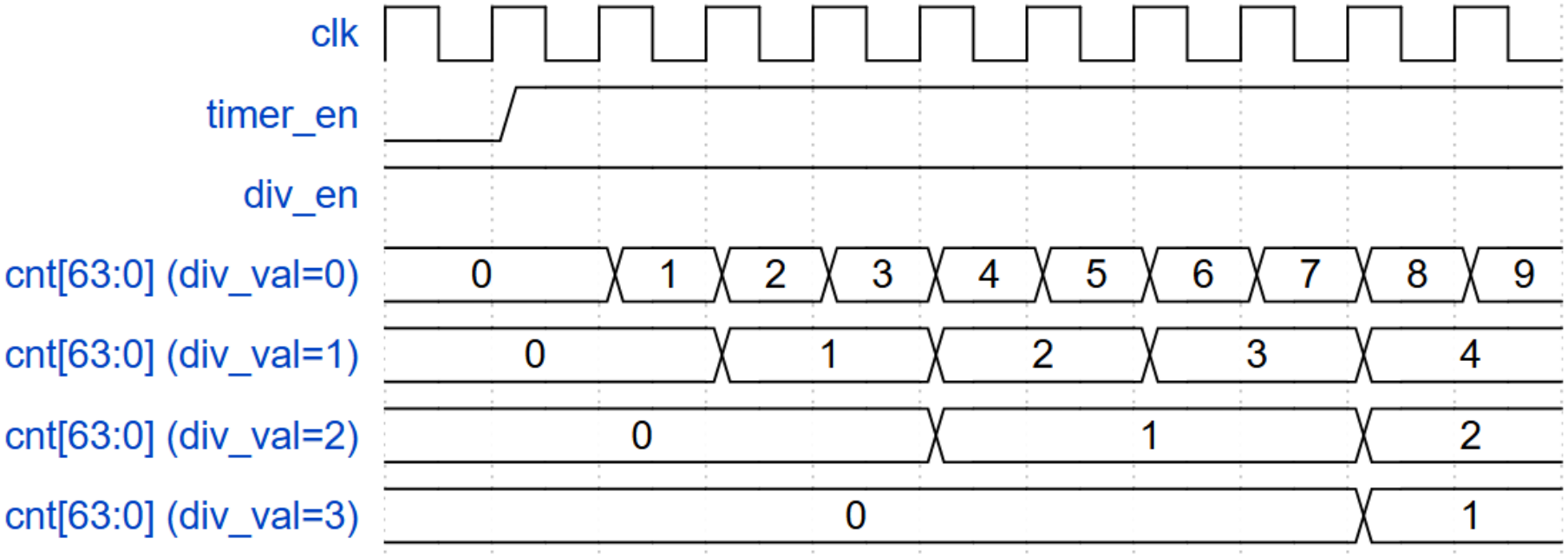
Timer Interrupt

- Timer Interrupt
  - Timer interrupt (tim_int) is asserted (set) when interrupt is enabled and counter's value matches (equal) the compare value.
  - 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.

DETAIL FUNCTIONAL DESCRIPTION

Counting Mode

- Counting mode:
  - In default mode, counting speed depend on system clock (same as div_val=0 case as below waveform).
  - The counting's speed can be controlled by register settings by setting into TCR.div_en and TCR.div_val[3:0] as the register specification
  - div_en and div_val can not be changed during timer_en is High.
    - Standard level: testbench should not change div_en and div_val.
    - Advanced level: not allow to change div_en and div_val while timer_en is High by an error response when user mistakenly accesses.
  - Example waveform of counter if control mode.



DETAIL FUNCTIONAL DESCRIPTION

APB slave / Register

- APB slave / Register
  - Address space: 4KB (0x4000_1000 – 0x4000_1FFF)
  - Read/write to reserved area is RAZ/WI (read as zero, write ignored)
  - System clock frequency is 200 MHz.
  - Standard level:
    - Only support APB 32-bit transfer
    - No wait states and no error handling
  - Advanced level:
    - Support byte access: bus can access to individual bytes in the register.
    - Support wait state (1cycle) to improve the timing.
    - Support error handling for some prohibited access:
      - write prohibited value to TCR.div_val
      - div_en or div_val changes during timer is operating
    - When error occurs, data is not written into register bit/fields

REGISTER SUMMARY

Base address: 0x4000_1000

Offset	Abbreviation	Register 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	

REGISTER SPECIFICATION

Timer Control Register -TCR

Bit	Name	Type	Default value	Description
31:12	Reserved	-	20'h0	Reserved
11:8	div_val	RW	4'b0001	Counter control mode setting: • 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 • Others: reserved, (*)prohibit settings. When setting the prohibit value, div_val is not changed. Note: user must not change div_en while timer_en is High (*): add hardware logic to ensure div_val is prohibited to change when timer_en is High. Access is error response in this case. (*)access is "error response" when setting prohibit value to div_val
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	Timer enable • 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

(*): advanced level

REGISTER SPECIFICATION

Timer Data Register 0 –TDR0

Bit	Name	Type	Default value	Description
31:0	TDR0	RW	32’h0000_0000	Lower 32-bit of 64-bit counter. [Advanced level]: value of this register is cleared to initial value when timer_en changes from H->L.

REGISTER SPECIFICATION

Timer Data Register 1 –TDR1

Bit	Name	Type	Default value	Description
31:0	TDR1	RW	32’h0000_0000	Upper 32-bit of 64-bit counter. [Advanced level]: value of this register is cleared to initial value when timer_en changes from H->L.

REGISTER SPECIFICATION

Timer Compare Register 0 –TCMP0

Bit	Name	Type	Default value	Description
31:0	TCMP0	RW	32’hFFFF_FFFF	Lower 32-bit of 64-bit compare value. Interrupt is asserted when counter value is qual to compare value.

REGISTER SPECIFICATION

Timer Compare Register 1 –TCMP1

Bit	Name	Type	Default value	Description
31:0	TCMP1	RW	32’hFFFF_FFFF	Upper 32-bit of 64-bit compare value. Interrupt is asserted when counter value is qual to compare value.

REGISTER SPECIFICATION

Timer Interrupt Enable Register–TIER

Bit	Name	Type	Default value	Description
31:1	Reserved	RO	31'h0	Reserved
0	int_en	R/W	1'b0	Timer interrupt enable 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. Clearing this bit to 0 while interrupt is asserting will mask the interrupt to 0 but does not affect the interrupt pending bit TISR.int_st bit

REGISTER SPECIFICATION

Timer Interrupt Status Register–TISR

Bit	Name	Type	Default value	Description
31:1	Reserved	RO	31'h0	Reserved
0	int_st	RW1C	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.

REGISTER SPECIFICATION

Timer Halt Control Status Register–THCSR

Bit	Name	Type	Default value	Description
31:2	Reserved	RO	30'h0	Reserved
1	halt_ack	RO	1'b0	[Standard level] This bit is reserved bit [Advanced level] Timer halt acknowledge 0: timer is NOT halted 1: timer is halted Timer accepts the halt request only in debug mode, indicates by debug_mode input signal
0	halt_req	RW	1'b0	[Standard level] This bit is normal R/W but has no function related. [Advanced level] Timer halt request 0: no halt req. 1: timer is requested to halt.

IO Port List

Top module name: timer_top

Signal name	Width	Direction	Description
sys_clk			
sys_rst_n			
tim_psel			
tim_pwrite			
tim_penable			
tim_paddr			Note: the address bit-width is calculated based on address space. The bit-width must be enough to contain all the address in the address space
tim_pwdata			
tim_prdata			
tim_pstrb			
tim_pready			
tim_pslverr			
tim_int			
dbg_mode			

RELEASE DATA FOR FINAL PROJECT

Release data:

- ❑ Design:
  - Design spec full (Power Point or Word)
  - Clean RTL code
- ❑ Verification
  - Verification plan (excel format)
  - Verification environment (testbench, script)
  - 100% testcases passed with checker
  - Coverage report
    - Standard level: 90%
    - Advanced level: 100%


THANK YOU