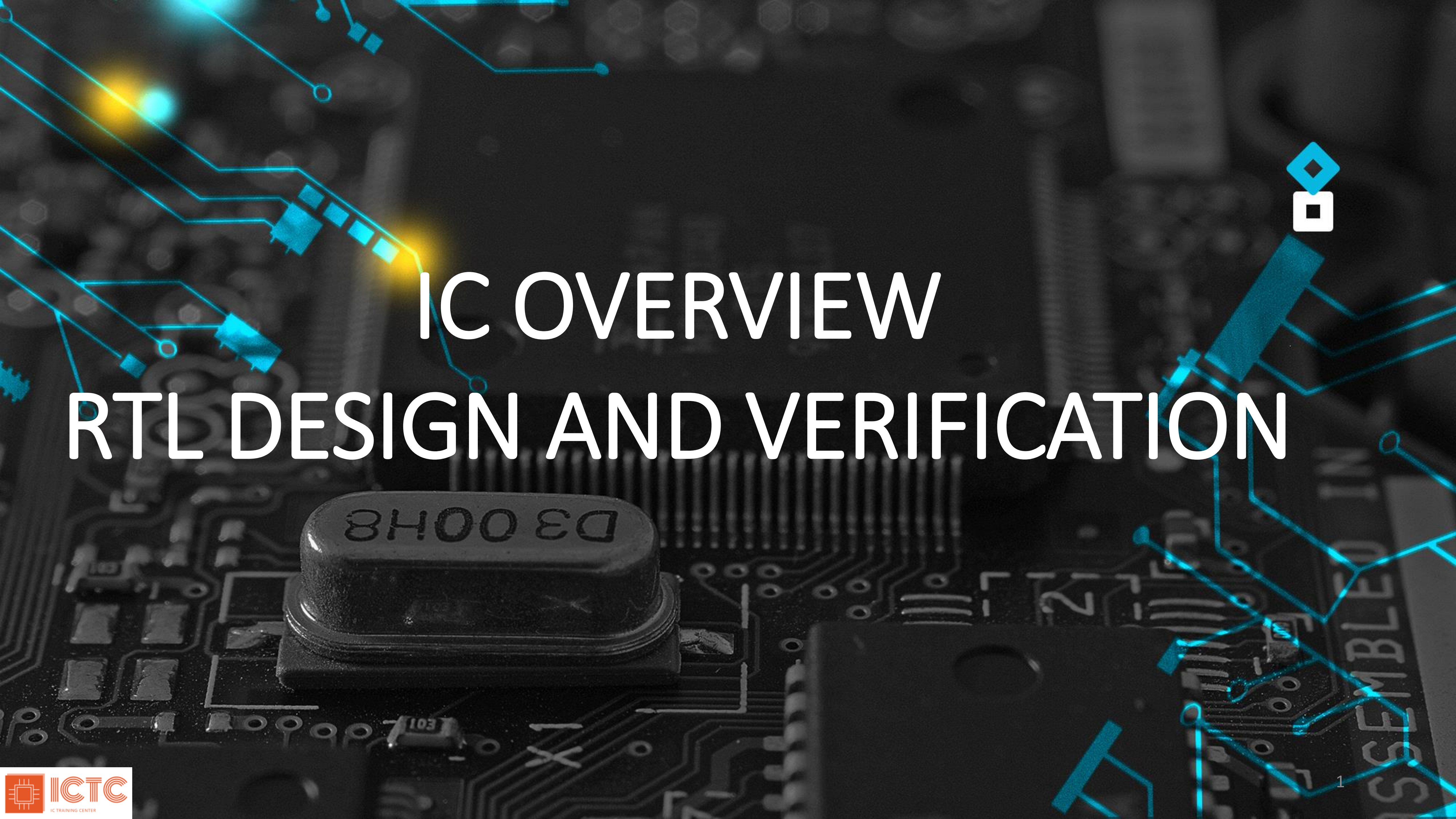
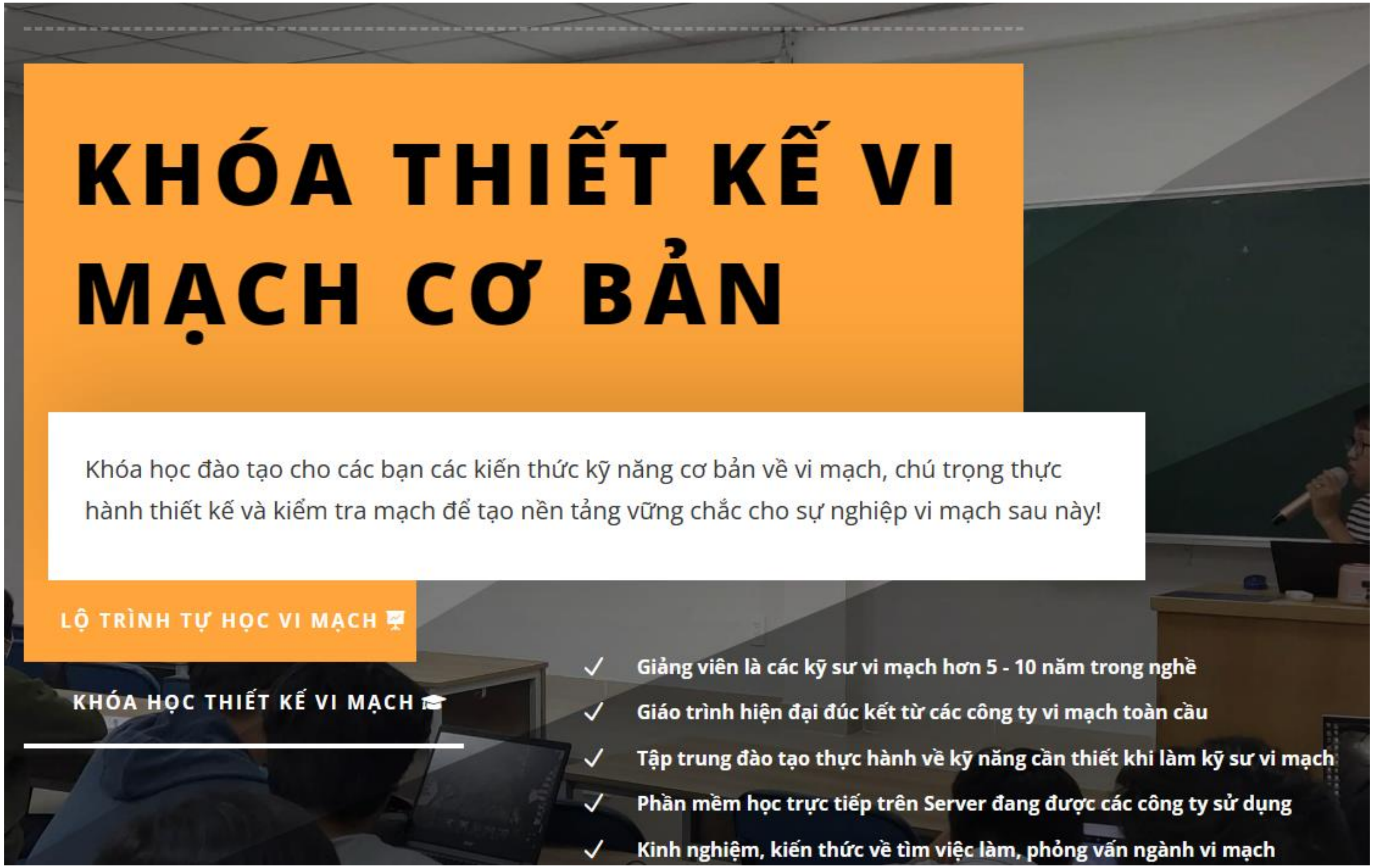


IC OVERVIEW
RTL DESIGN AND VERIFICATION

ICTC
IC TRAINING CENTER

COURSE INTRODUCTION

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



KHÓA THIẾT KẾ VI MẠCH CƠ BẢN

Khóa học đào tạo cho các bạn các kiến thức kỹ năng cơ bản về vi mạch, chú trọng thực hành thiết kế và kiểm tra mạch để tạo nền tảng vững chắc cho sự nghiệp vi mạch sau này!

- ✓ Giảng viên là các kỹ sư vi mạch hơn 5 - 10 năm trong nghề
- ✓ Giáo trình hiện đại đúc kết từ các công ty vi mạch toàn cầu
- ✓ Tập trung đào tạo thực hành về kỹ năng cần thiết khi làm kỹ sư vi mạch
- ✓ Phần mềm học trực tiếp trên Server đang được các công ty sử dụng
- ✓ Kinh nghiệm, kiến thức về tìm việc làm, phỏng vấn ngành vi mạch

COURSE INTRODUCTION

SUMMARY

HOMEWORK

QUESTION

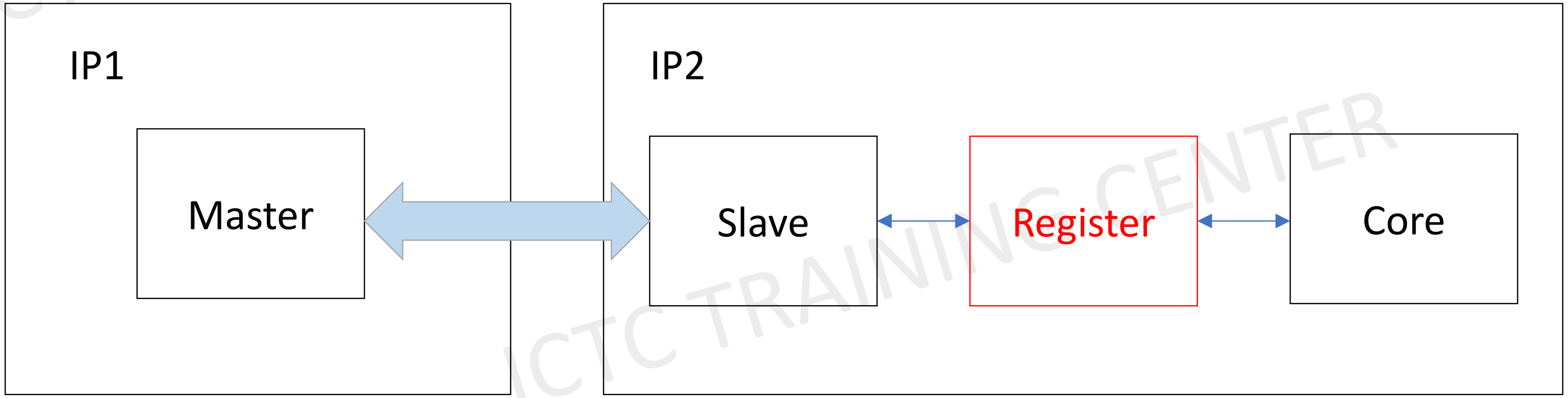
SELF-LEARNING

Session 12: Register Module

REGISTER MODULE

Register module is a fundamental building block in a IP design. It is used to:

- Store configuration to the IP
- Store IP status for monitoring/debug.
- Interact with other devices.



REGISTER SIZE

A register size is often in

- 8 bit (byte access)
- 16 bit (half word access)
- 32 bit (word access)

0x0	Register 0
0x1	Register 1
0x2	Register 2
	….
n	Register n

8 bit registers

0x0	Register 0
0x2	Register 1
0x4	Register 2
	….
2xn	Register n

16 bit registers

0x0	Register 0
0x4	Register 1
0x8	Register 2
	….
4xn	Register n

32 bit registers

0x0	8 bit
0x1	32 bit
0x5	16 bit
0x7	8 bit
0x8	8 bit

Mixed size registers

REGISTER ATTRIBUTE

Some common register attribute:

- Read Write (RW): a bit/register can be written and read value
- Read Only (RO): a bit/register can only be read, write does not affect
- Write Only (WO): a bit/register can only by write, read is not possible
- R/W1C: a bit/register is read only, write 1 to clear it.
- Reserved: Write Ignored, Read as Zero (WI/RAZ)

REGISTER TABLE

Register table example:

Bit	Name	Type	Default value	Description
31:0	FIELD_1	RW	32’h0000_0001	Description of FIELD_1

Single field register

Bit	Name	Type	Default value	Description
31:6	Reserved	RO	26’b0	Reserved area
5	FIELD_3	RO	1’b0	Description of FIELD_3
4:3	FIELD_2	R/W	2’b00	Description of FIELD_2
2:0	FILED_1	R/W	3’b010	Description of FIELD_1

Multiple fields register

REGISTER PRACTICE

Requirement

Practice:

Design the below register specification.

- 1KB address space
- 32-bit access only
- Data is available in the register 1 cycle after a write request.
- Data is returned immediately when there is a read access.
- Read data is always 0 when not access to valid address.
- Support continuous read/write (read and write request do not occur at the same time).

Data 0 register: 32 bit addr = 0x0

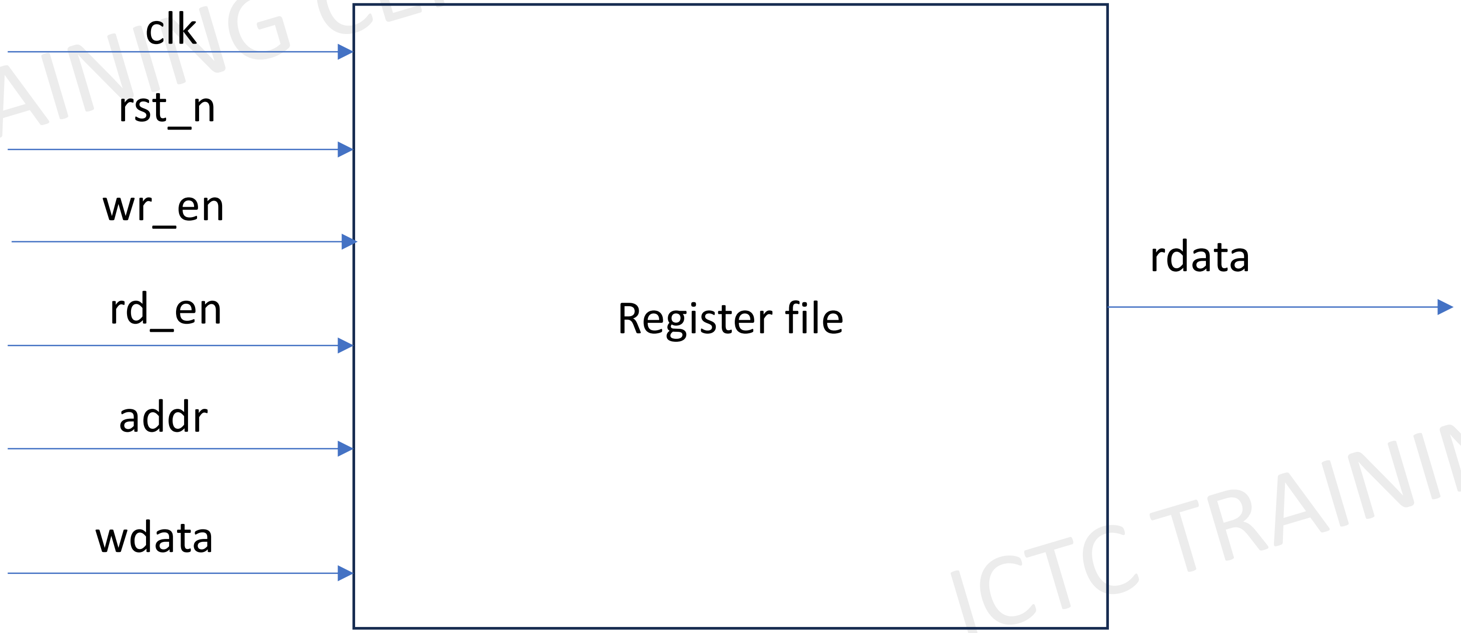
Bit	**Name**	**Type**	**Default value**	**Description**
31:0	DATA0	RW	32’h0000_0000	DATA0 data register

Status data 0 register: 32 bit addr = 0x4

Bit	**Name**	**Type**	**Default value**	**Description**
31:0	SR_DATA0	RO	32’h0000_0000	Store value of Data 0 register

REGISTER PRACTICE

Input output diagram



Let’s identify the bit-width of addr, wdata, rdata

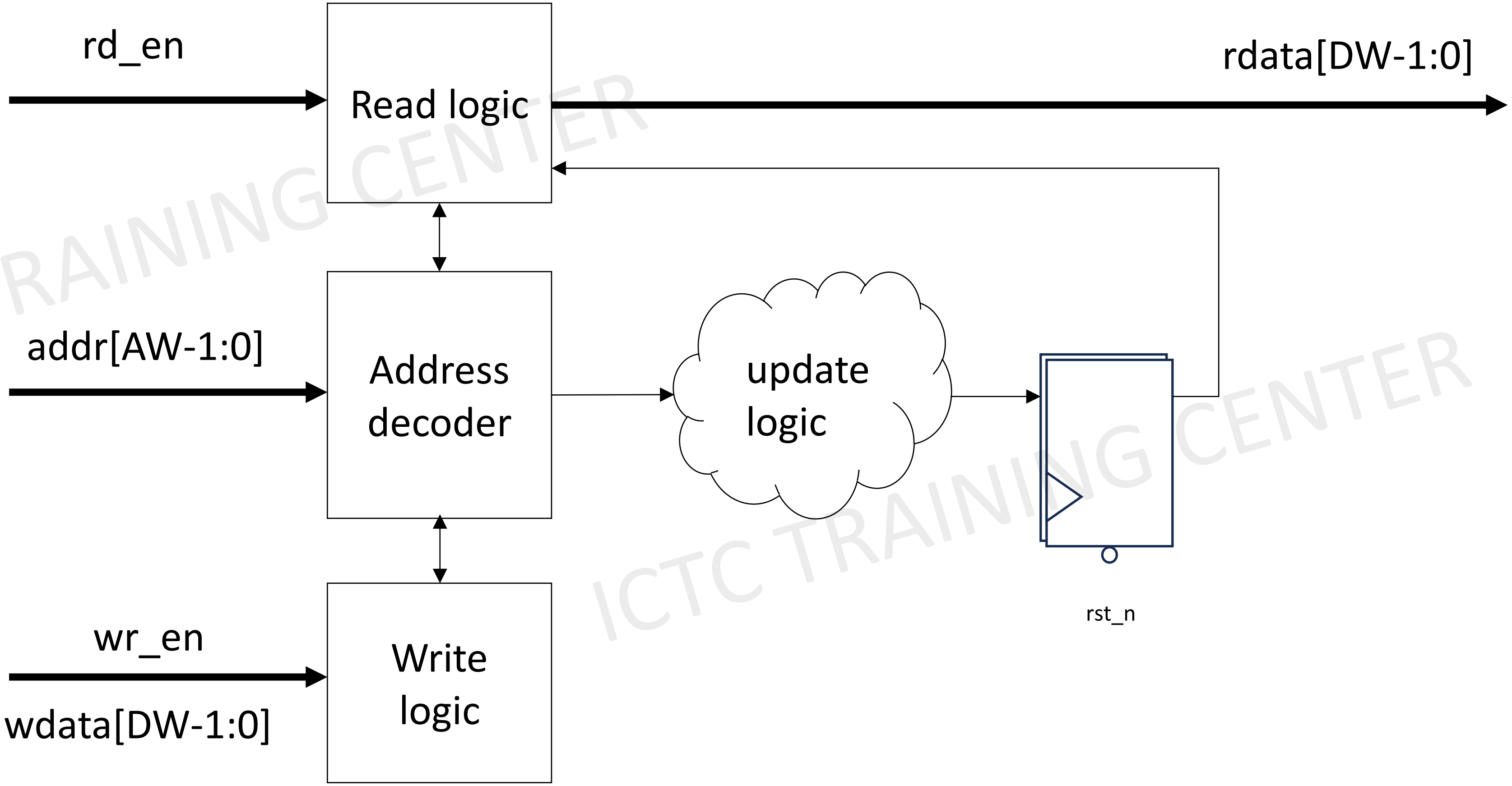
REGISTER PRACTICE

IO DESCRIPTION

Signal	Direction	Bit-width	Description
clk	Input	1	Input clock
rst_n	Input	1	Active low asynchronous reset 0: counter is in reset state 1: counter is in non-reset state
wr_en	Input	1	Write request enable 0: no write request 1: write request enable. Write data is available and is updated into registers
rd_en	Input	1	
addr	Input		
wdata	Input		
rdata	Output		

Let's finish this IO table

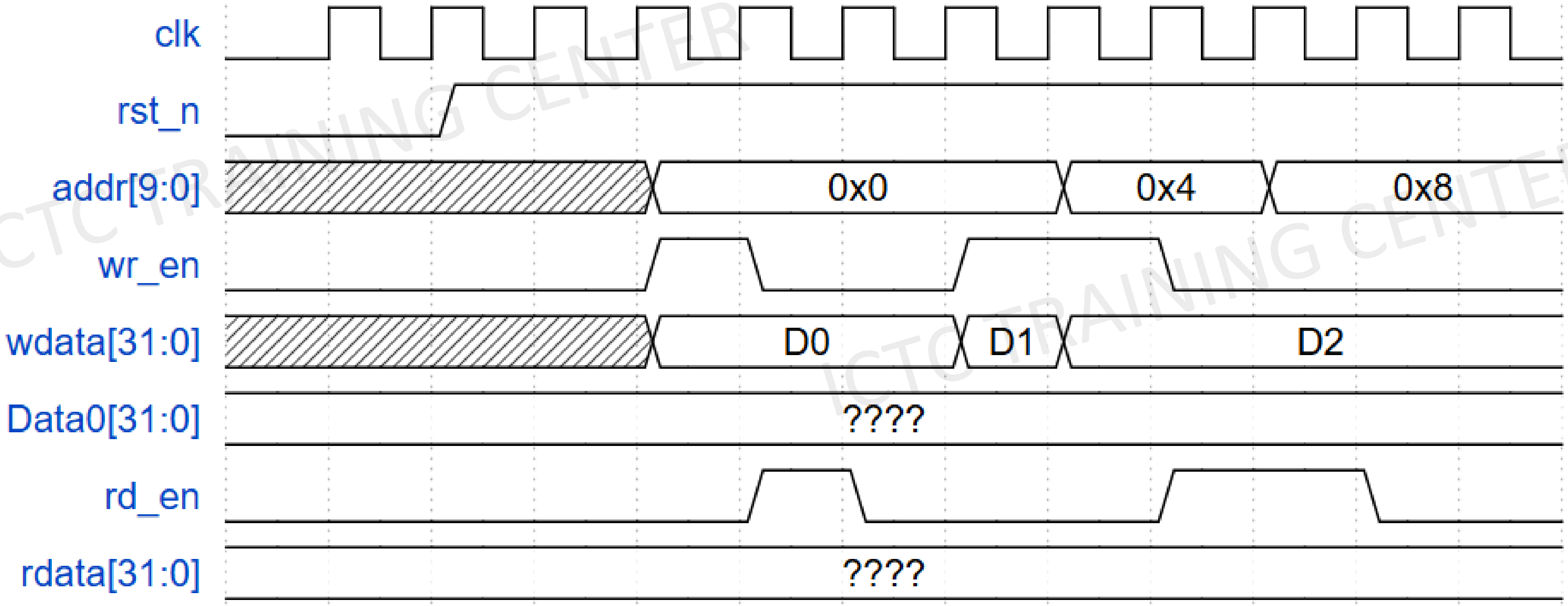
REGISTER BLOCK DIAGRAM



- RW register can store data, each register bit is a flip-flop, which can be updated or read value.
- RO register holds the value of signals/other F.Fs. It doesn’t have to use F.F to store value.

REGISTER TIMING DIAGRAM

Practice: Complete the timing diagram below



REGISTER TIMING DIAGRAM

Code wavedrom for write timing

```
{signal: [
  {name: 'clk', wave: '0p...........'},
  {name: 'rst_n', wave: '0.1..........'},
  {name: 'addr[9:0]', wave: 'x...=...=.=..',data:[“0x0",“0x4",“0x8"]},
  {name: 'wr_en', wave: '0...10.1.0...',},
  {name: 'wdata[31:0]', wave: 'x...=..==....',data:["D0","D1","D2"]},
  {name: 'Data0[31:0]', wave: '=............',data:["????"]},
    {name: 'rd_en', wave: '0....10..1.0.',},
  {name: 'rdata[31:0]', wave: '=............',data:["????"]},

]}
```

REGISTER LOGIC DIAGRAM

Let’s draw logic diagram for write and read access.

VERILOG CODE

Let’s write Verilog code to describe above behavior.

Session 12

Homework1:
Add following registers to the design

Data 1 register: 32 bit, addr =0x8

Bit	Name	Type	Default value	Description
31:0	DATA1	RW	32'hFFFF_FFFF	DATAA1 data register

Status data 1 register: 32 bit addr = 0xC

Bit	Name	Type	Default value	Description
31:0	SR_DATA1	RO	32'hFFFF_FFFF	Store value of Data 1 register

Finish the RTL Code and Write a simple testbench to test the design based on your understanding.

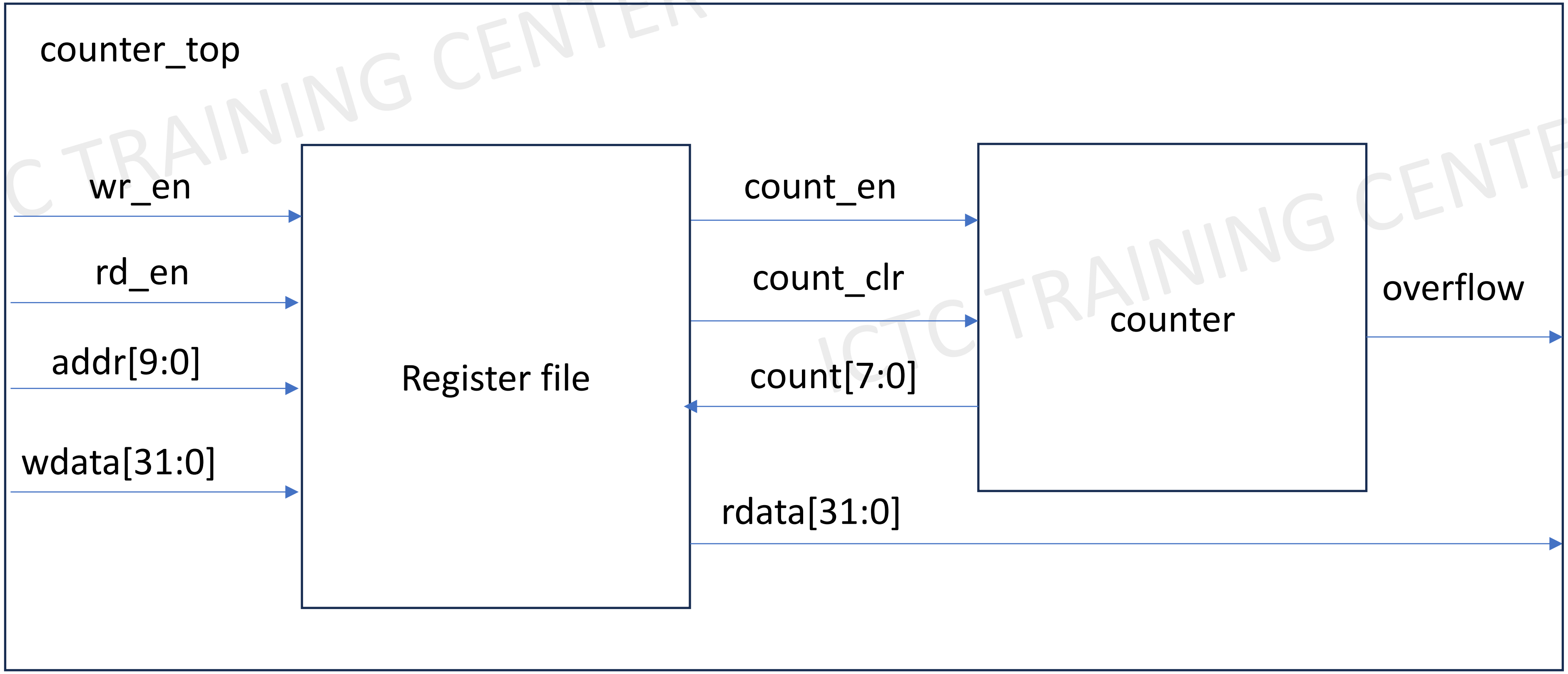
Session 12

Homework2(*):

Design register module to control for the counter module (already designed before).

Write simple testbench to test the behavior of register and counter.

The register and counter use same clock and active low asynchronous reset.



Session 12

Homework2(*):

Register specification

Control register (addr = 0)

Bit	Name	Type	Default value	Description
31:2	Reserved	RO	30’b0	Reserved
1	count_clr	RW	1’b0	0: not clear 1: clear cnt
0	count_start	RW	1’b0	0: idle 1: count start

Status register (addr = 4)

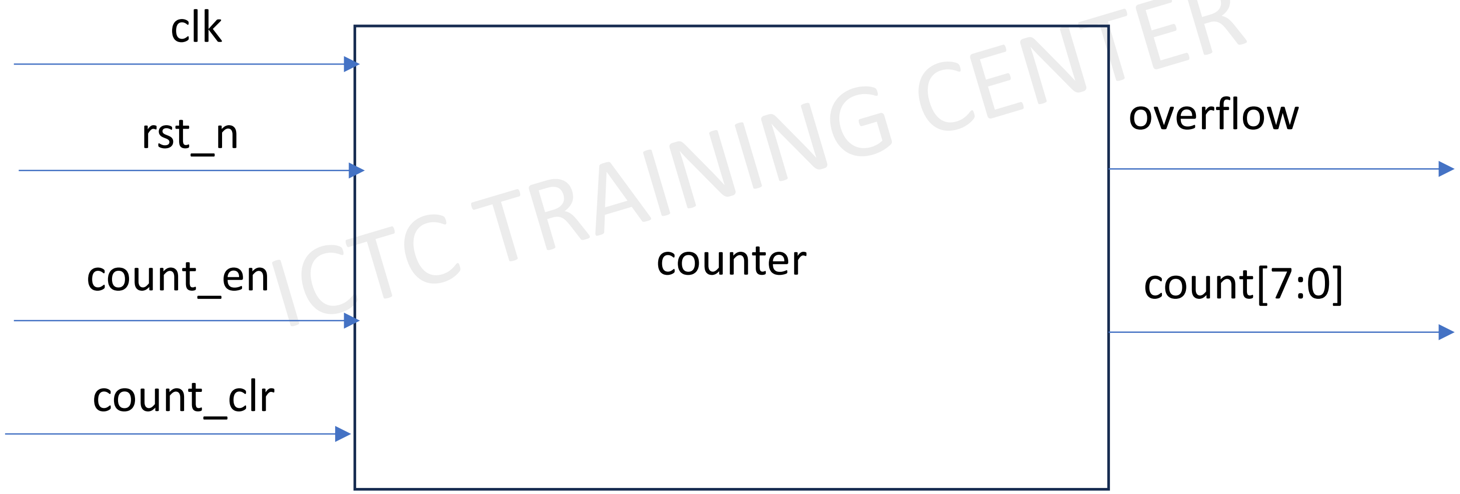
Bit	Name	Type	Default value	Description
31:8	Reserved	RO	24’b0	Reserved
7:0	Cnt	RO	8’h0	Cnt value

SESSION 12

Homework2(*)

Specification of 8bit counter in previous sessions:

- 8-bit counter using D-FF, low active async reset.
- Reset (initial) value is 8'h00
- When counter reach max value, it overflowed (overflow = 1) and count again.
- "overflow" is assert only when counter is overflowed and negate after that
- Counter only start counting when input "count_en" is High. Otherwise, keep current value.
- Counter is cleared to its initial value (8'h00) when "count_clr" is High regardless of count_en. Note that count_clr is not a reset signal, it's just a data signal.



PREPARING FOR FINAL PROJECT

In ss11, we did a homework to make a transfer using task.
That is the APB protocol that we will used in our final project.
Let’s investigate the APB protocol by yourselves to practice the specification research.
Download the specification: AMBA_APB_Specification.pdf

- The class is divided into 2 groups by the lecturer based on your current ranking.
- Each group investigate and prepare for the group presentation.
- The presentation should be in maximum 20min
- All group member should be presented in the presentation date.
- The presentation will be held in ss16 (2weeks later).
- Points will be assigned for ss15 advanced level points (ss15 does not have advanced level homework).
- Team members that do not present will be divided the points by 2.
- Team members that absent on the presentation day without proper reason will get zero points.

The presentation should cover:

- Introduction
- Detail on signals and transfers
- Practical usage

…