实验2

简单计算机系统基本模块设计B

主要内容

1、简单计算机系统实验任务简介

2、模块设计：

RAM，标志位寄存器

3、动手练习：仿真验证功能

实验任务简介

- 实现一种简单计算机系统的设计.
  - ✓ 精简的MIPS指令集
  - ✓ EDA仿真
- 编写程序，仿真验证所设计系统的功能
  - ✓ 用汇编格式编写程序，并翻译成机器码.
  - ✓ 将机器码程序放入ROM，通过仿真验证简单计算机系统的功能.

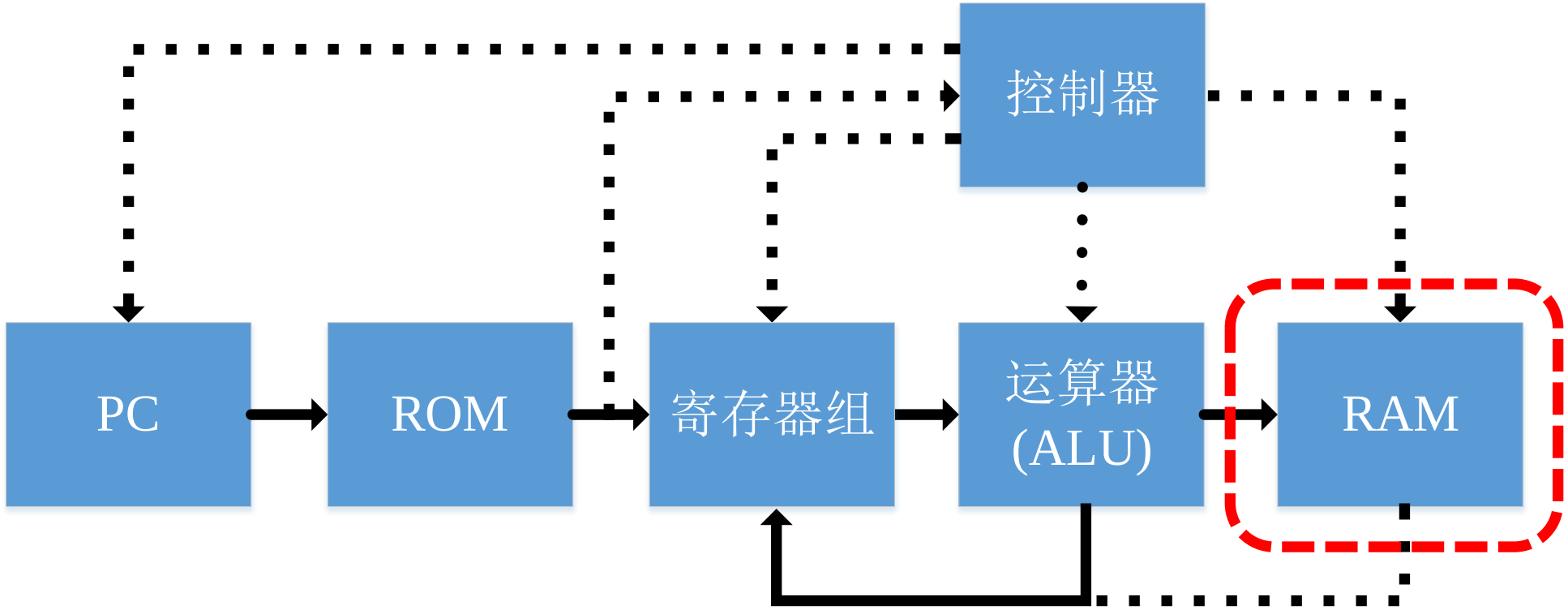
实验目的

- 学习、掌握计算机部件模块电路设计方法
- 熟悉计算机的组成与各部件的功能
- 熟悉简单计算机的指令集和数据通路

  掌握编写汇编语言程序和机器码程序
- 设计一个简单计算机系统

更好地理解和掌握计算机的组成与原理

计算机模型



1、RAM模块设计

采用Quartus Prime

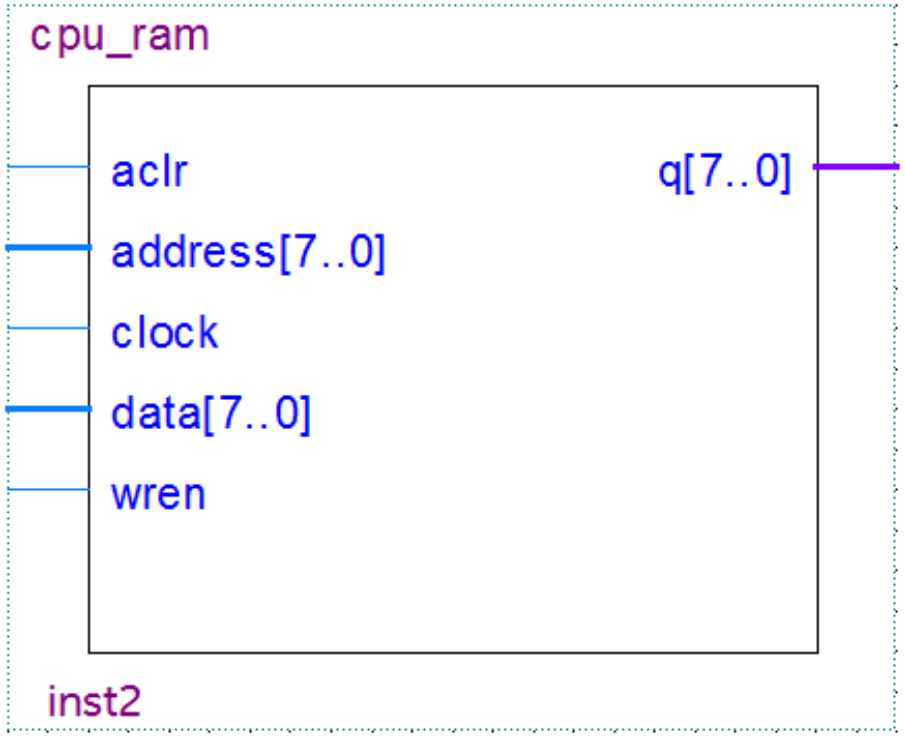
RAM模块

■ **输入信号**

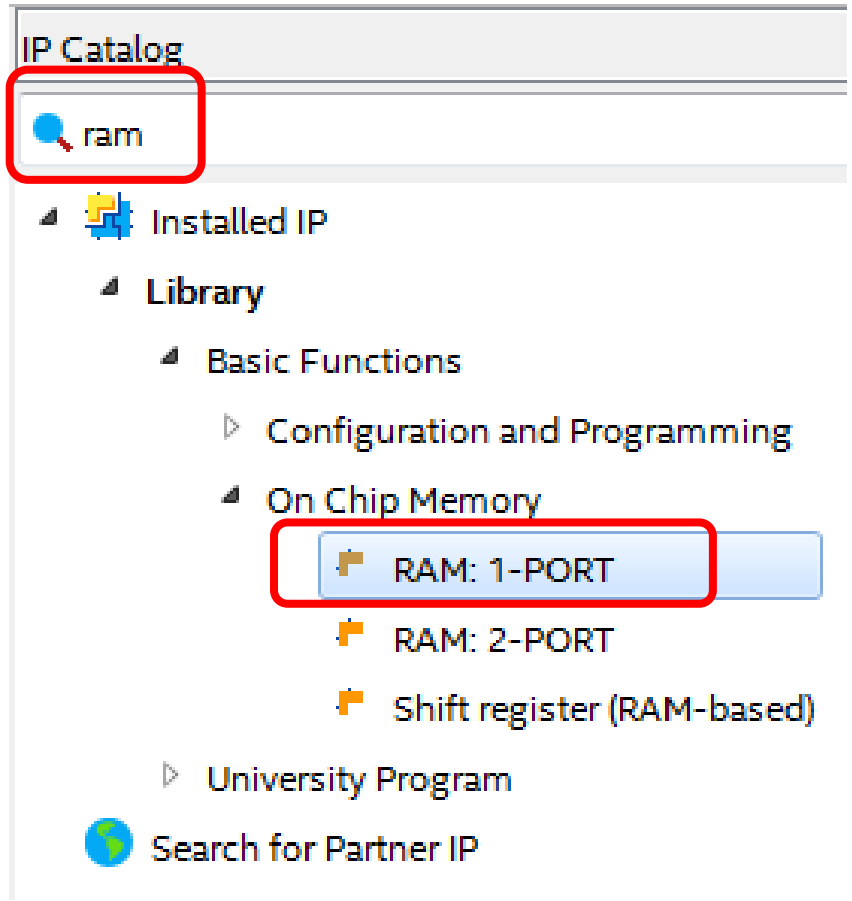
- ✓ address[7..0]：需要进行读/写操作的RAM单元地址
- ✓ data[7..0]：待写入RAM中的数据
- ✓ wren：写允许信号，为1时写；为0则读
- ✓ clock：时钟信号，上升沿将数据写入或读出
- ✓ aclr： 异步复位信号，对q[7..0]进行清0

■ **输出信号**

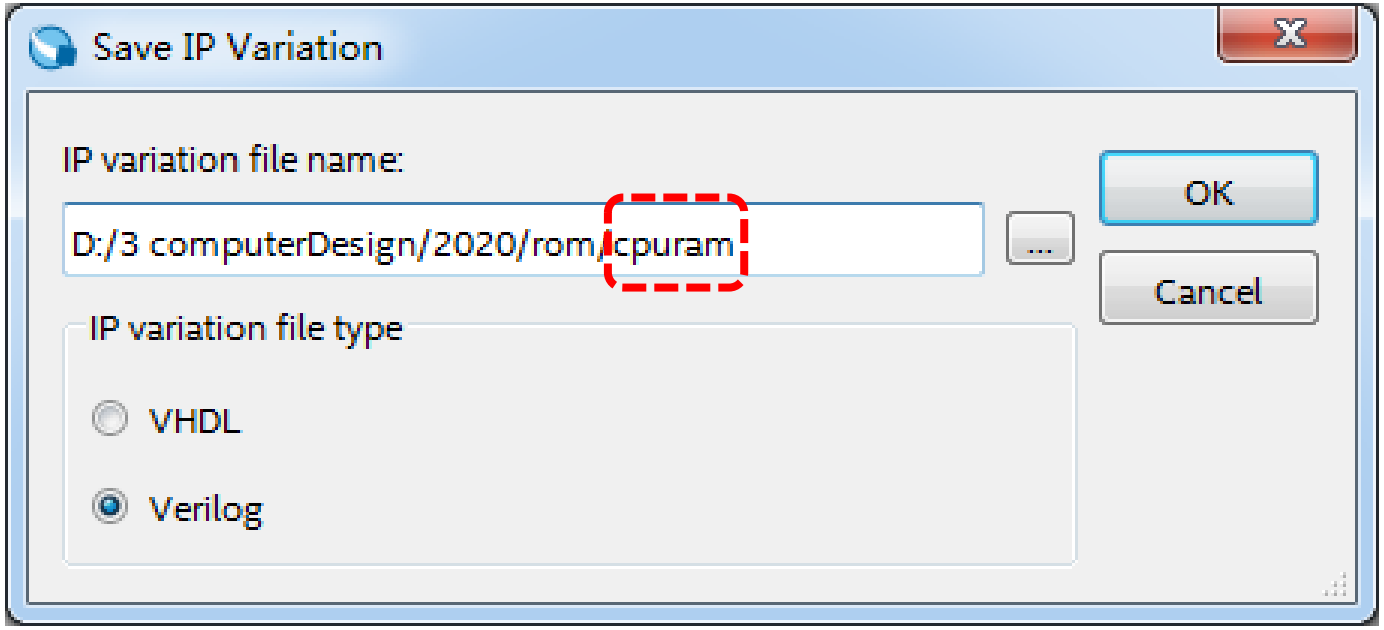
- ✓ q[7..0]：输出address [7..0]指定的RAM单元的内容



RAM IP


IP Catalog
ram
Installed IP
Library
Basic Functions
Configuration and Programming
On Chip Memory
RAM: 1-PORT
RAM: 2-PORT
Shift register (RAM-based)
University Program
Search for Partner IP

命名


Save IP Variation
IP variation file name:
D:/3 computerDesign/2020/rom/cpuram
...
OK
Cancel
IP variation file type
VHDL
Verilog

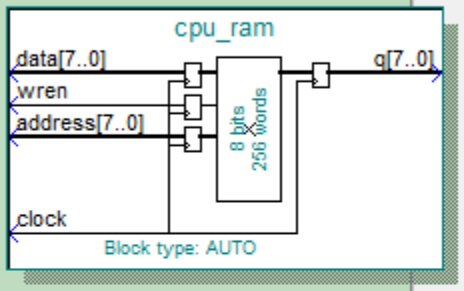
MegaWizard Plug-In Manage [page 1 of 6]

RAM: 1-PORT 共6步

About | Documentation

1 Parameter Settings | 2 EDA | 3 Summary

Widths/Blk Type/Clks > Regs/Clken/Byte Enable/Aclrs > Read During Write Option > Mem Init



Currently selected device family: Cyclone 10 LP

☑ Match project/default

How wide should the 'q' output bus be? 8 bits

How many 8-bit words of memory? 256 words

Note: You could enter arbitrary values for width and depth

What should the memory block type be?

- (●) Auto
- ( ) MLAB
- ( ) M9K
- ( ) M144K
- ( ) LCs

Options...

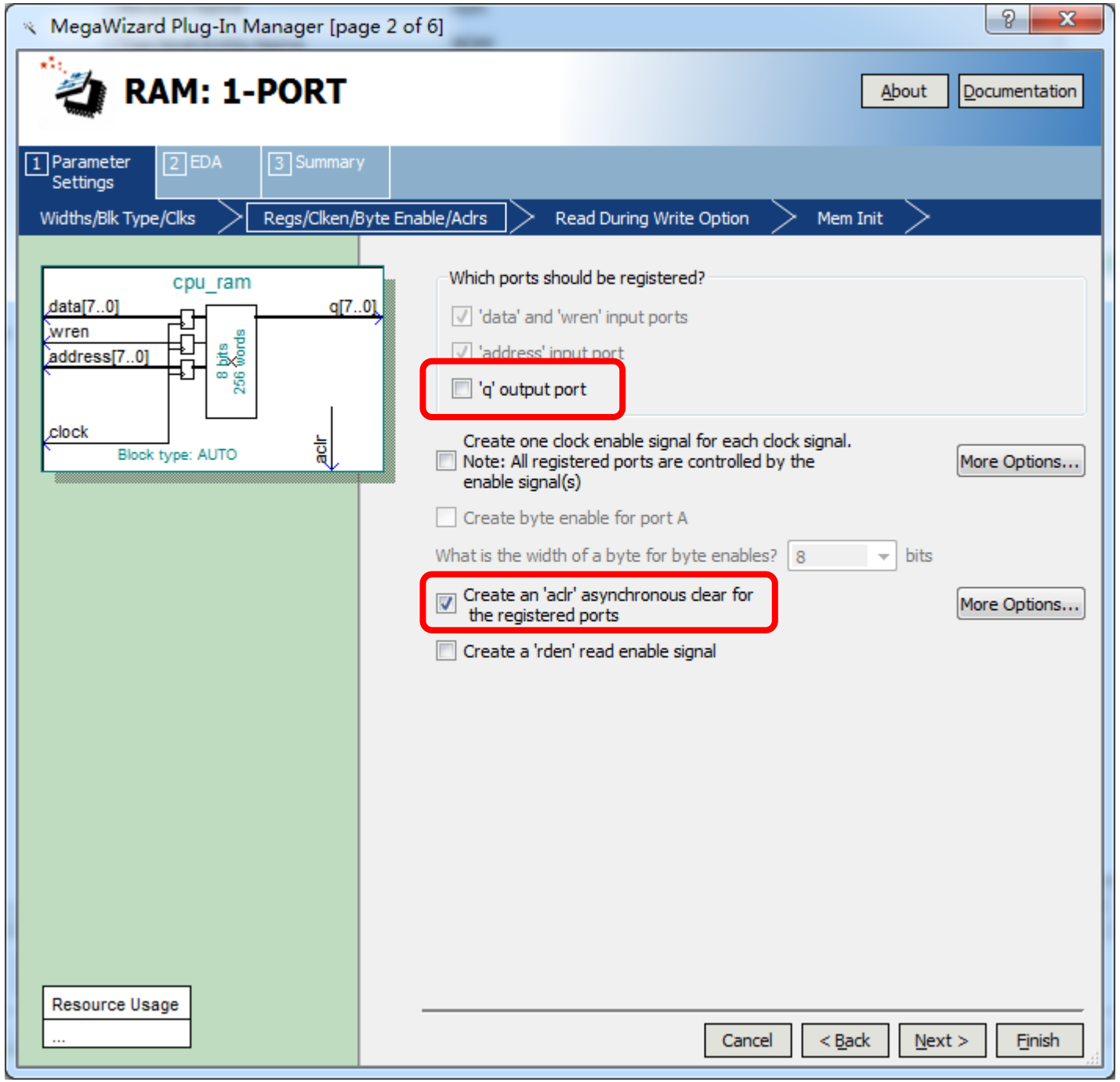
Set the maximum block depth to Auto words

What clocking method would you like to use?

- (●) Single clock
- ( ) Dual clock: use separate 'input' and 'output' clocks

Resource Usage
1 M9K

Cancel | < Back | Next > | Finish


MegaWizard Plug-In Manager [page 2 of 6]
RAM: 1-PORT
About
Documentation
1 Parameter Settings
2 EDA
3 Summary
Widths/Blk Type/Clks
Regs/Clken/Byte Enable/Adrs
Read During Write Option
Mem Init
cpu_ram
data[7..0]
wren
address[7..0]
clock
q[7..0]
8 bits
256 words
aclr
Block type: AUTO
Which ports should be registered?
'data' and 'wren' input ports
'address' input port
'q' output port
Create one clock enable signal for each clock signal.
Note: All registered ports are controlled by the enable signal(s)
More Options...
Create byte enable for port A
What is the width of a byte for byte enables?
8
bits
Create an 'aclr' asynchronous clear for the registered ports
More Options...
Create a 'rden' read enable signal
Resource Usage
...
Cancel
< Back
Next >
Finish

第3/4/5步骤

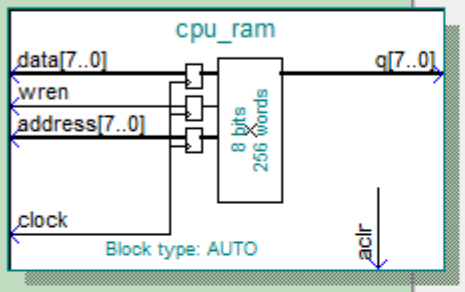
取默认值

MegaWizard Plug-In Manager [page 6 of 6]

RAM: 1-PORT

About | Documentation

1 Parameter Settings | 2 EDA | 3 Summary



Turn on the files you wish to generate. A gray checkmark indicates a file that is automatically generated, and a green checkmark indicates an optional file. Click Finish to generate the selected files. The state of each checkbox is maintained in subsequent MegaWizard Plug-In Manager sessions.

The MegaWizard Plug-In Manager creates the selected files in the following directory:

D:\3 computerDesign\2020\rom\

File	Description
☑ cpu_ram.v	Variation file
☐ cpu_ram.inc	AHDL Include file
☐ cpu_ram.cmp	VHDL component declaration file
☐ cpu_ram.bsf	Quartus Prime symbol file
☑ cpu_ram_inst.v	Instantiation template file
☑ cpu_ram_bb.v	Verilog HDL black-box file

Resource Usage
1 M9K

Cancel | < Back | Next > | Finish

Quartus Prime IP Files

When you create an Intel IP variation, a Quartus Prime IP File is generated. Quartus Prime IP Files are used to represent the Intel IP in your design. Do you want to add the Quartus Prime IP File to the project?

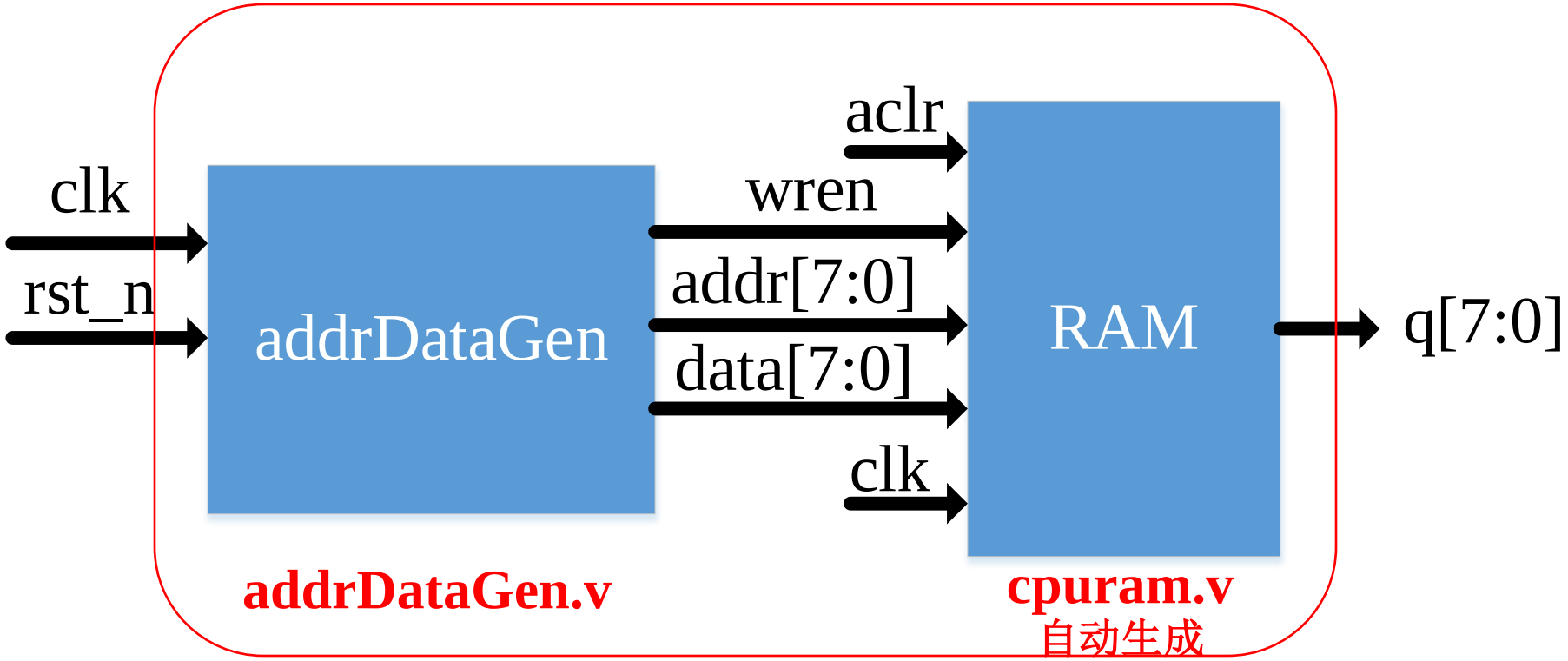
☑ D:\3 computerDesign\2020\rom\cpu_ram.qip

☐ Automatically add Quartus Prime IP Files to all projects

(Note: Turning on this option permanently suppresses this dialog box. You can change this setting in the Options dialog box)

Yes | No | Help

测试RAM模块



RAM.v

顶层

```
module addrDataGen(clk,rst_n,wren,aclr,addr,data);
  input clk;
  input rst_n;
  output reg wren;
  output reg aclr;
  output reg [7:0] addr;
  output reg [7:0] data;
  reg state;
  always @ (posedge clk or negedge rst_n)
    begin
      ......
      ......
    end
endmodule
```

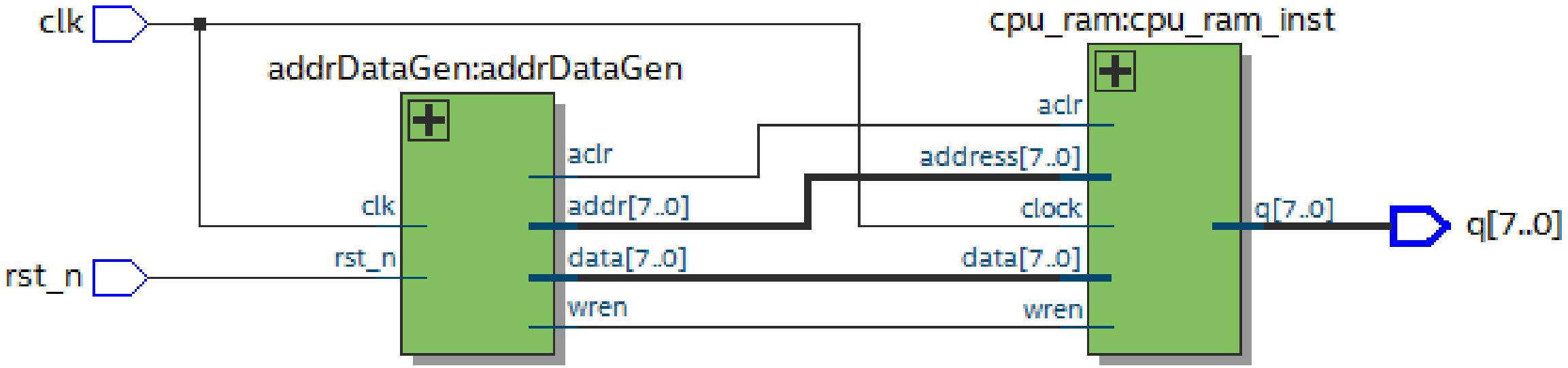
addrDataGen.v

```
module RAM(clk,rst_n,q);
  input clk;
  input rst_n;
  output [7:0] q;
  ......
  ......
endmodule
```

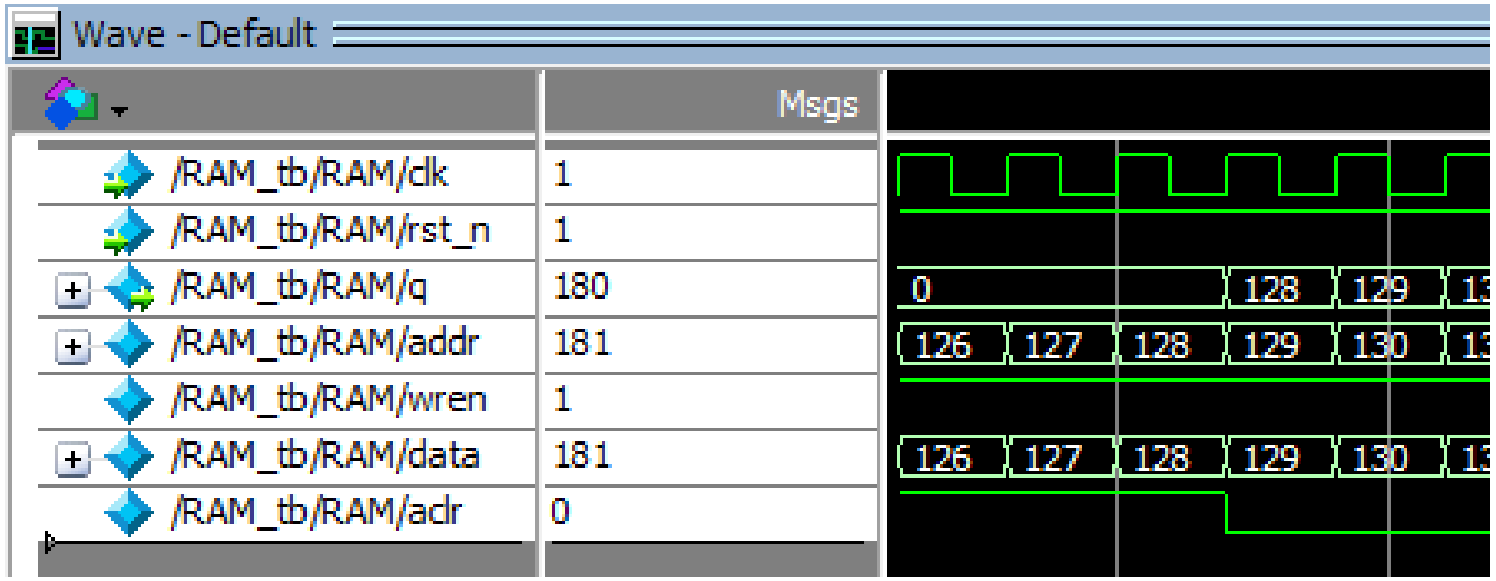
RAM.v

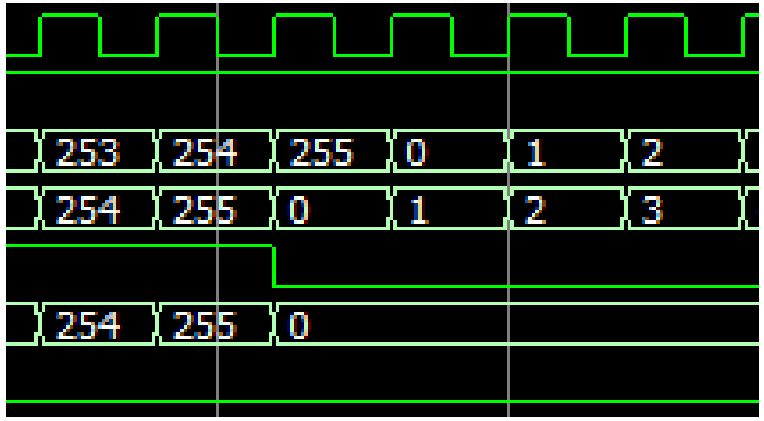
```
`timescale 1ns/1ps
module RAM_tb;
  reg clk;
  reg rst_n;
  wire [7:0] q;
  initial begin
    ......
  end
  ......
endmodule
```

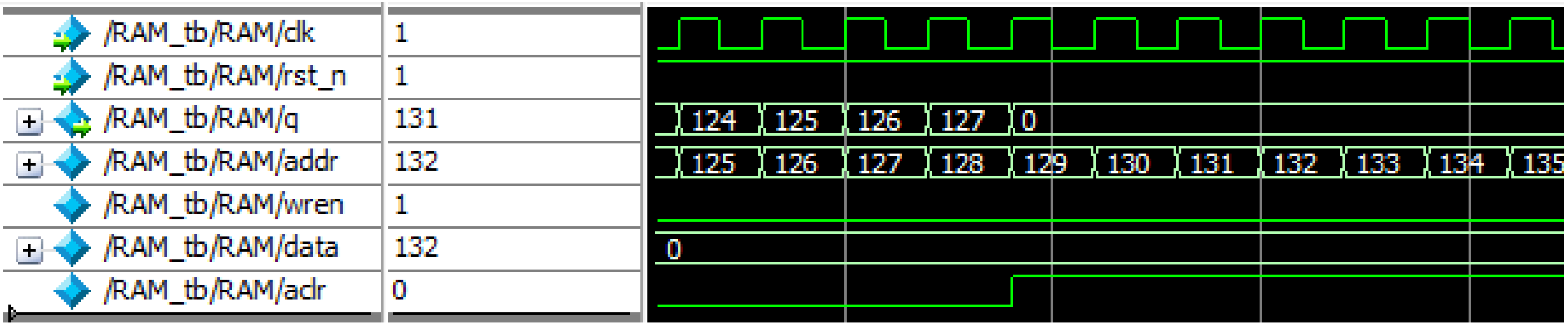
RAM_tb.v

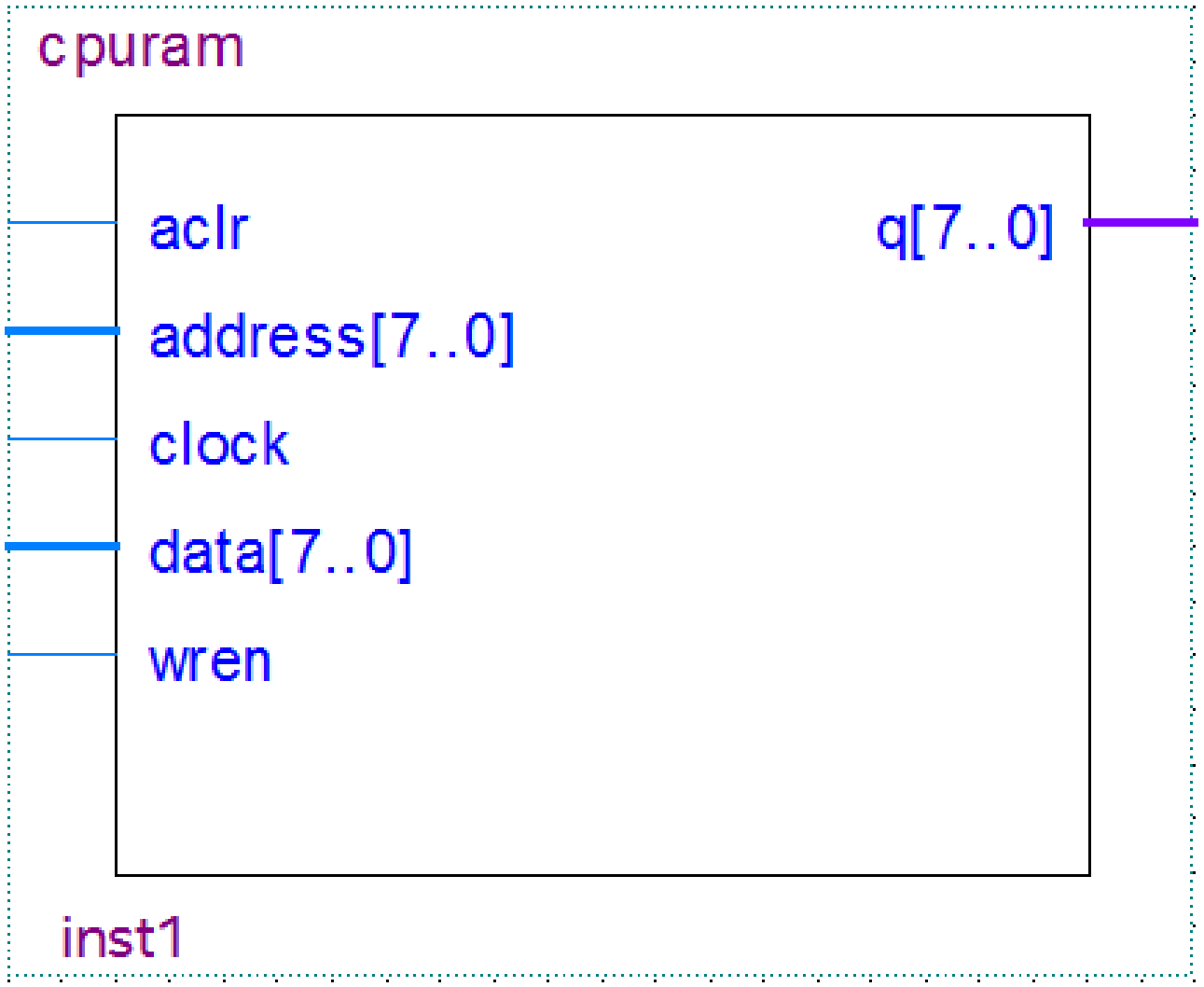

clk
rst_n
addrDataGen:addrDataGen
clk
rst_n
aclr
addr[7..0]
data[7..0]
wren
cpu_ram:cpu_ram_inst
aclr
address[7..0]
clock
data[7..0]
wren
q[7..0]
q[7..0]

验证：存、取RAM的过程


Wave - Default
Msgs
/RAM_tb/RAM/clk 1
/RAM_tb/RAM/rst_n 1
/RAM_tb/RAM/q 180
/RAM_tb/RAM/addr 181
/RAM_tb/RAM/wren 1
/RAM_tb/RAM/data 181
/RAM_tb/RAM/adr 0
0 128 129
126 127 128 129 130
126 127 128 129 130


253 254 255 0 1 2
254 255 0 1 2 3
254 255 0


/RAM_tb/RAM/clk 1
/RAM_tb/RAM/rst_n 1
/RAM_tb/RAM/q 131
/RAM_tb/RAM/addr 132
/RAM_tb/RAM/wren 1
/RAM_tb/RAM/data 132
/RAM_tb/RAM/adr 0
124 125 126 127 0
125 126 127 128 129 130 131 132 133 134 135
0


cpuram
aclr
address[7..0]
clock
data[7..0]
wren
q[7..0]
inst1

实验任务2

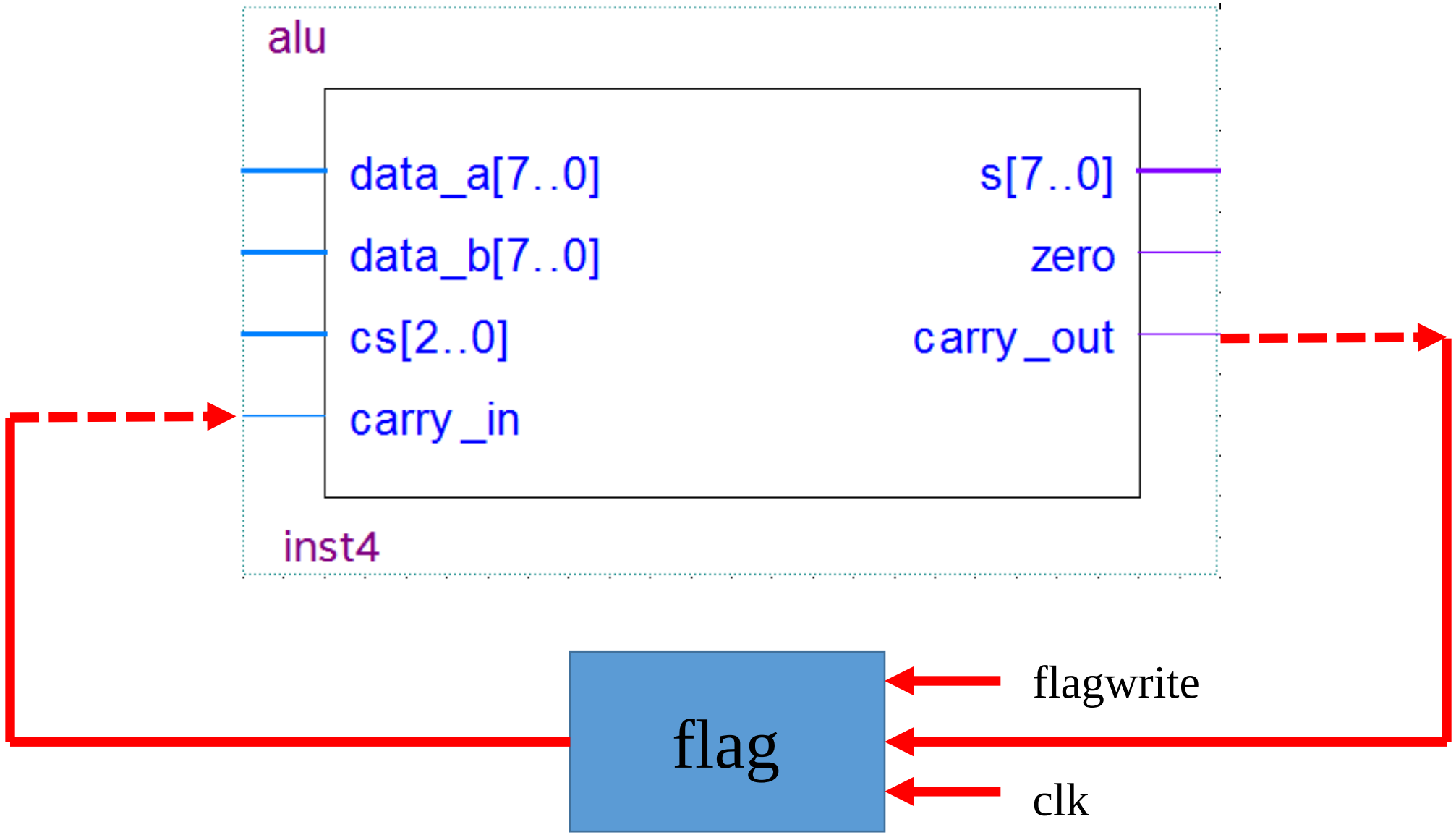
任务2.1

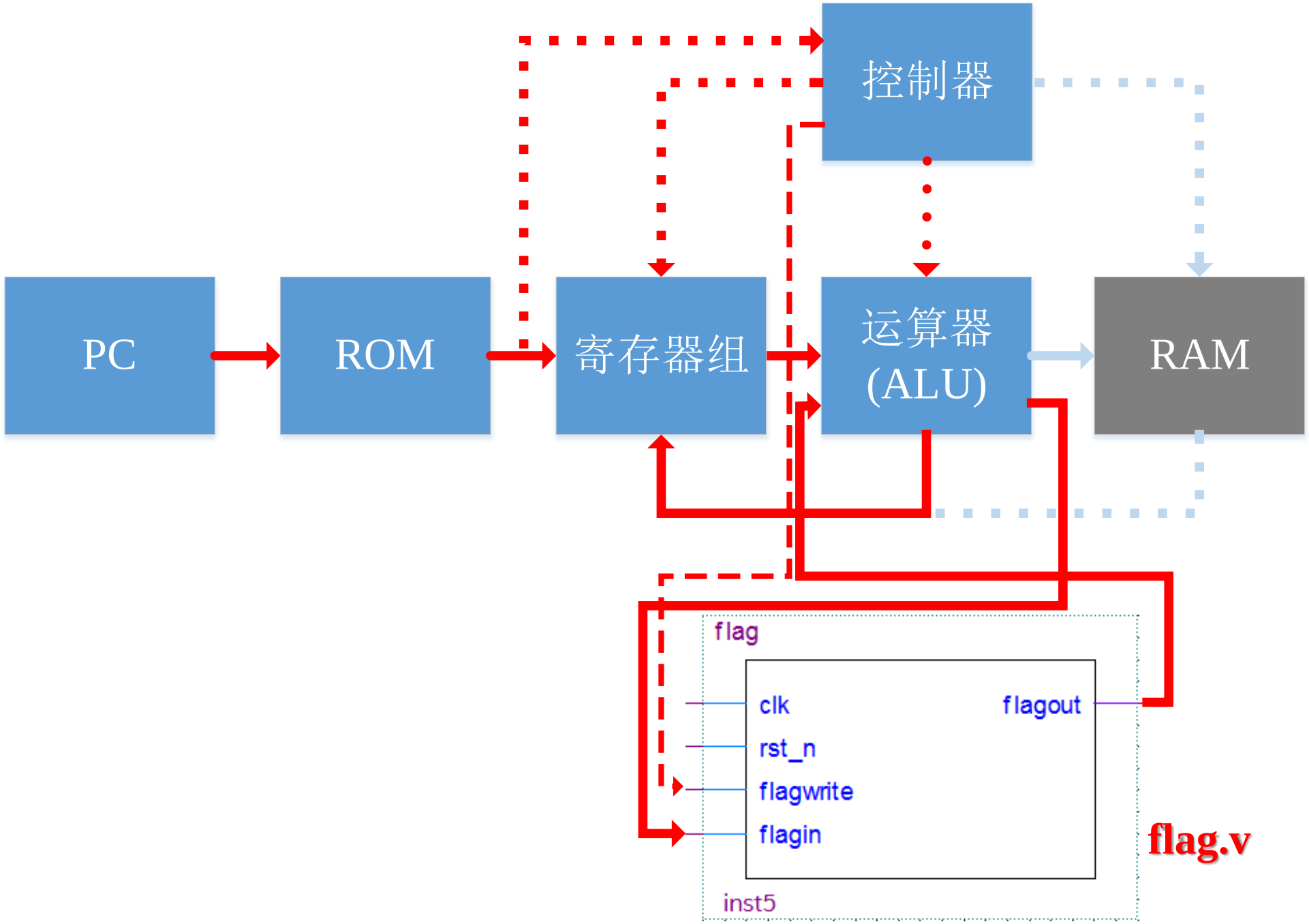
（1）参照上面的步骤，实现对RAM模块【8位】的仿真验证；分析仿真结果.

（2）设计32位位宽的RAM模块；给出仿真结果.

2、标志位寄存器模块

考虑带进位的加减运算时，还需处理进位、借位的问题……

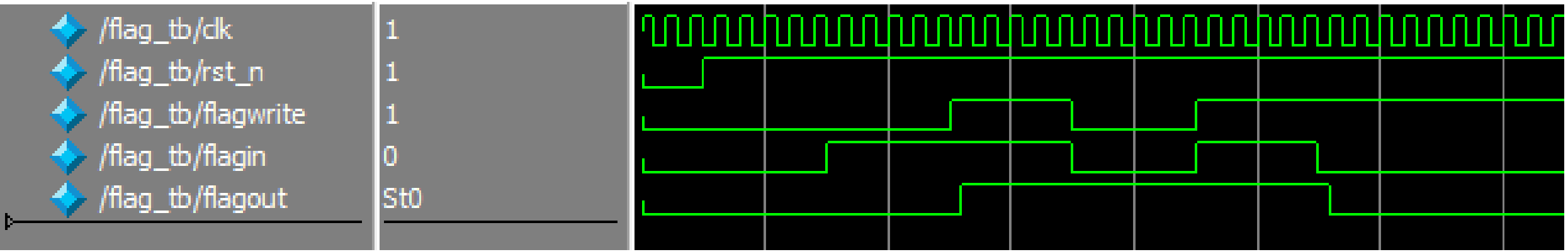


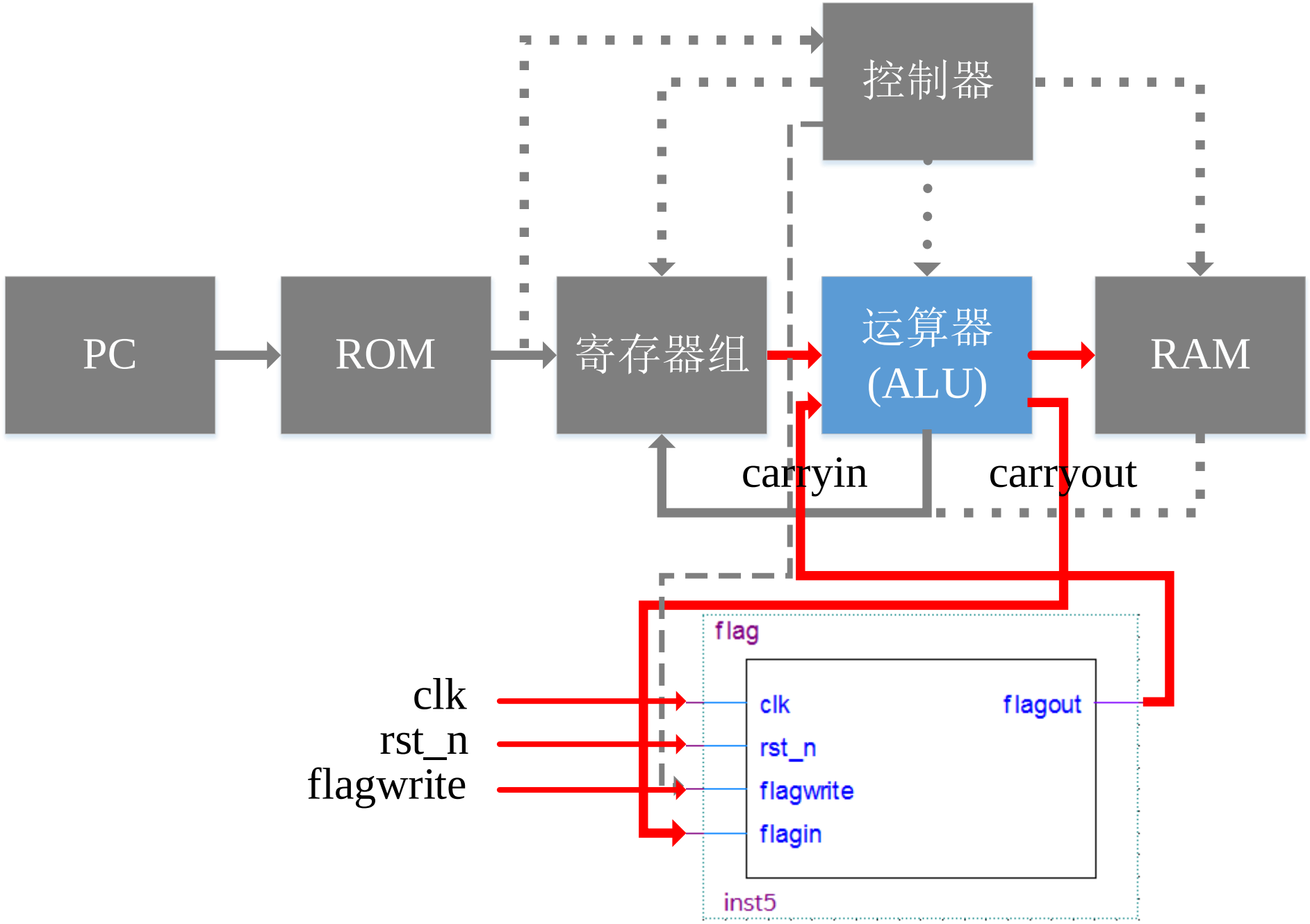

控制器
PC
ROM
寄存器组
运算器
(ALU)
RAM
flag
clk
rst_n
flagwrite
flagin
flagout
inst5
flag.v

```
module flag(clk,rst_n,flagwrite,flagin,flagout);
  input clk;
  input rst_n;
  input flagin;
  input flagwrite;
  output reg flagout;
  ……
endmodule
```

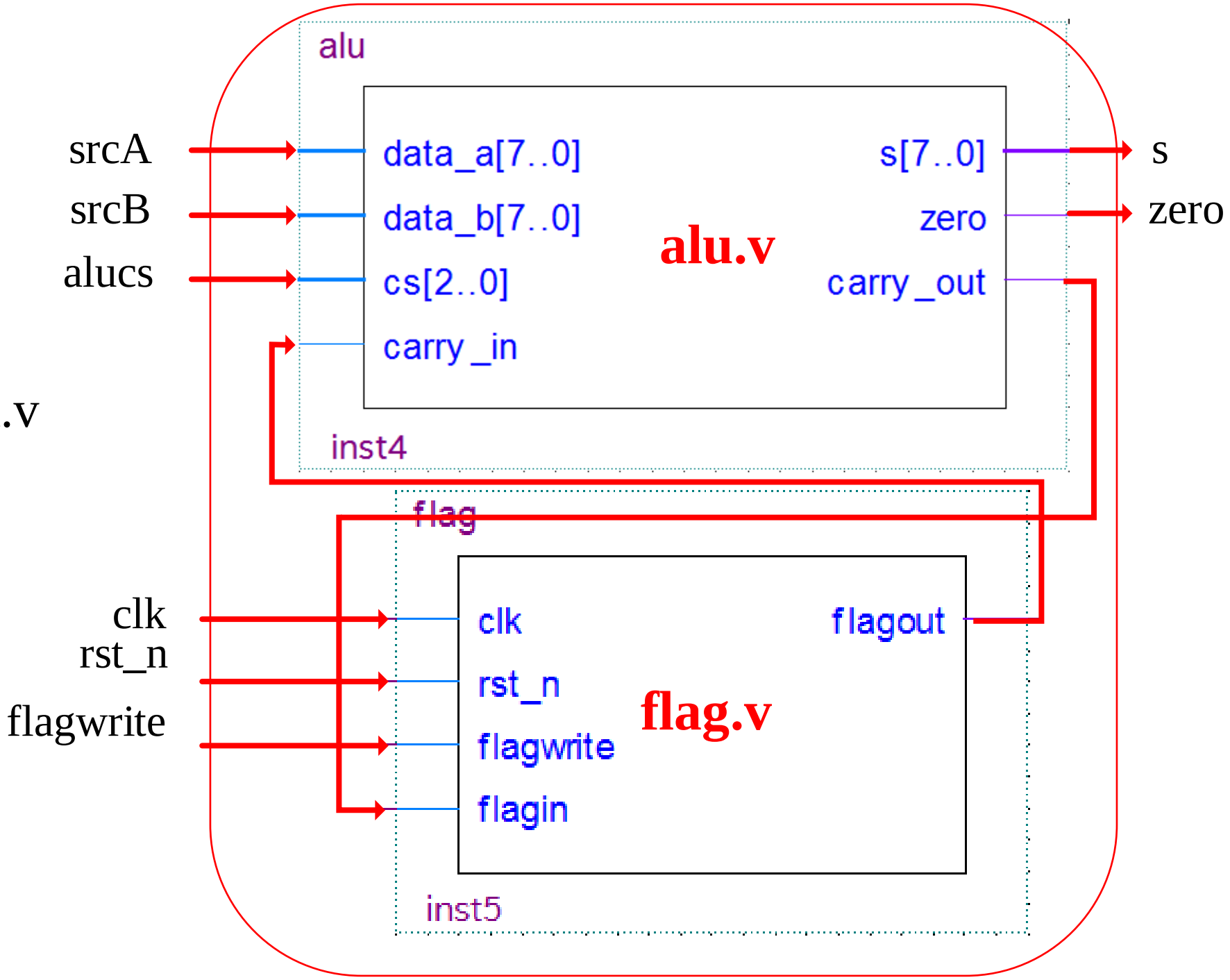
flag.v

```
……
initial begin
  clk = 0;
  rst_n = 0;
  flagwrite = 0;
  flagin = 0;
  #50.1 rst_n = 1;
  #100 flagin = 1;
  #100 flagwrite = 1;
  #100 flagwrite = 0;
         flagin = 0;
  ……
end
……
```

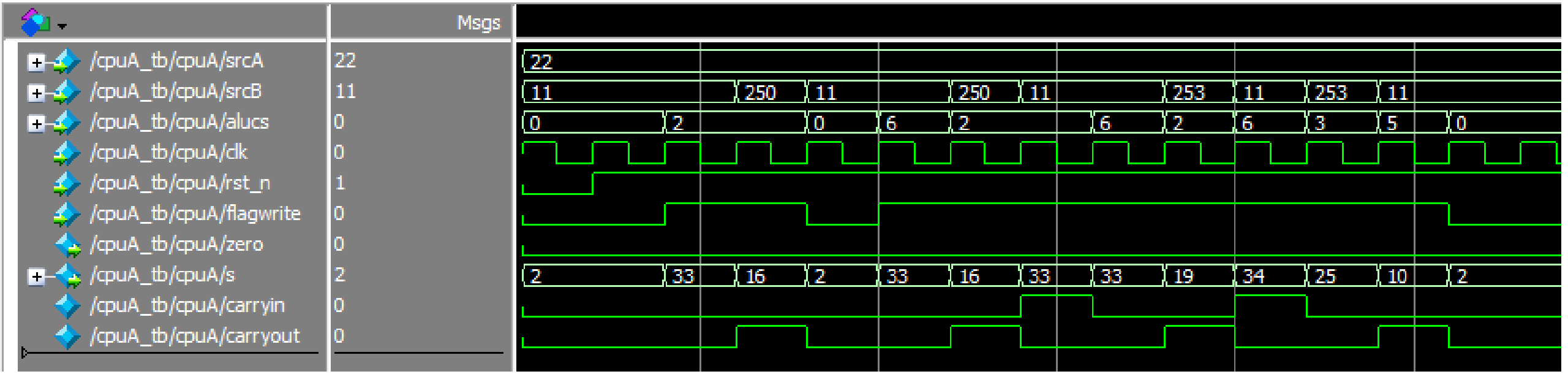



控制器
PC
ROM
寄存器组
运算器
(ALU)
RAM
carryin
carryout
flag
clk
rst_n
flagwrite
clk
rst_n
flagwrite
flagin
flagout
inst5

- 编写顶层文件cpuA.v
- 编写测试文件



```
initial begin
  clk = 1;
  rst_n = 0;
  srcA = 22;
  srcB = 11;
  alucs = 0;
  flagwrite = 0;
  #20 rst_n = 1;
  #20 alucs = 2;
        flagwrite = 1;
  #20 alucs = 2;
        srcB = 250;
        flagwrite = 1;
  #20  alucs = 0;
        srcB = 11;
        flagwrite = 0;
  #20  alucs = 6;
        srcB = 11;
        flagwrite = 1;
  #20 alucs = 2;
        srcB = 250;
        flagwrite = 1;
……
……
end
```



实验任务2

任务2.2

（1）参照上面的步骤，设计、仿真flag模块.

（2）编写顶层文件cpuA.v及其测试文件；测试加减与或等操作；设置适当的初值，对包含进位、借位的操作进行验证；分析仿真结果.

（3）如果要求alu输出信号zero，也存入标志寄存器，应修改哪些相关模块？如何修改？给出具体实现.

THE END

仿真中查看中间变量的方法

ModelSim - INTEL FPGA STARTER EDITION 2020.1

File Edit View Compile Simulate Add Structure Tools Layout Bookmarks Window Help

sim - Default

Instance	Design unit	Design unit typ
cpuA_tb	cpuA_tb	Module
cpuA		
#ALWAYS#63		
#vsim_capacity#		

View Declaration
View Instantiation
UVM
UPF
Add Wave Ctrl+W
Add Wave To
Add Dataflow Ctrl+D
Add to
Copy Ctrl+C
Find... Ctrl+F
Save Selected...
Expand Selected
Collapse Selected
Collapse All
Code Coverage
Test Analysis
XML Import Hint
Show

Objects

Name	Value	Now	
clk	St1	Net	In
rst_n	St1	Net	In
flagwrite	St0	Net	In
srcA	0001...	Net	In
srcB	0000...	Net	In
alucs	000	Net	In
s	0000...	Net	Out
zero	St0	Net	Out
carry_in	St1	Net	Internal
carry_out	St0	Net	Internal

Processes (Active)

Name	Type (filtered)

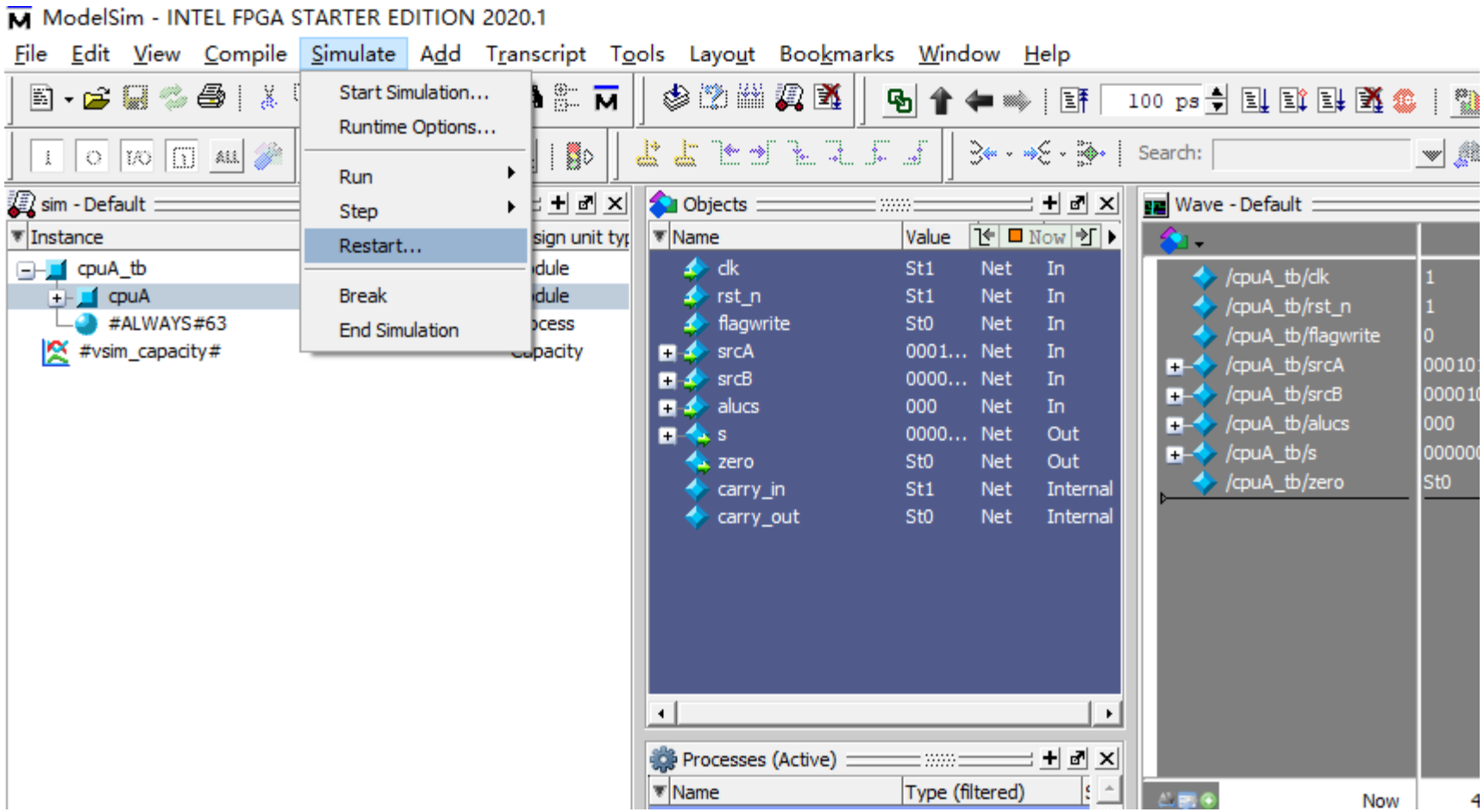
Wave - Default

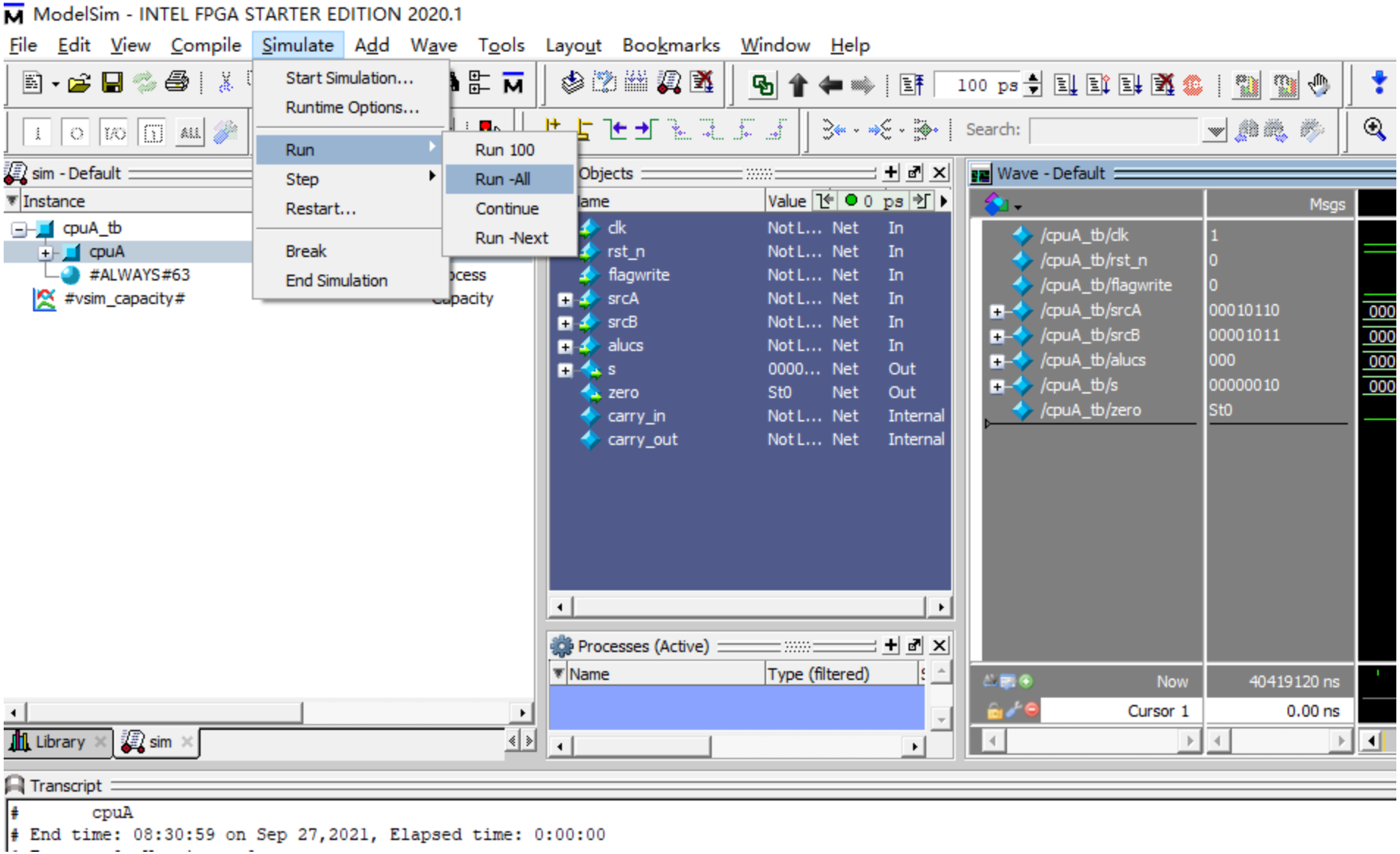
	Msgs
/cpuA_tb/clk	1
/cpuA_tb/rst_n	1
/cpuA_tb/flagwrite	0
/cpuA_tb/srcA	00010110
/cpuA_tb/srcB	00001011
/cpuA_tb/alucs	000
/cpuA_tb/s	00000010
/cpuA_tb/zero	St0
Now	40419120 ns
Cursor 1	0.00 ns

40419119.2 ns

Transcript

```
#        cpuA
# End time: 08:30:59 ime: 0:00:00
# Errors: 0, Warnings
#
# vlog -vlog01compat -work work +incdir+D:/intelFPGA_lite/user {D:/intelFPGA_lite/user/cpuA_tb.v}
# Model Technology ModelSim - Intel FPGA Edition vlog 2020.1 Compiler 2020.02 Feb 28 2020
# Start time: 08:30:59 on Sep 27,2021
# vlog -reportprogress 300 -vlog01compat -work work "+incdir+D:/intelFPGA_lite/user" D:/intelFPGA_lite/user/cpuA_tb.v
```


ModelSim - INTEL FPGA STARTER EDITION 2020.1
File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks Window Help
Start Simulation...
Runtime Options...
Run
Step
Restart...
Break
End Simulation
100 ps
Search:
sim - Default
Instance
cpuA_tb
cpuA
#ALWAYS#63
#vsim_capacity#
Objects
Name Value Now
clk St1 Net In
rst_n St1 Net In
flagwrite St0 Net In
srcA 0001... Net In
srcB 0000... Net In
alucs 000 Net In
s 0000... Net Out
zero St0 Net Out
carry_in St1 Net Internal
carry_out St0 Net Internal
Processes (Active)
Name Type (filtered)
Wave - Default
/cpuA_tb/clk 1
/cpuA_tb/rst_n 1
/cpuA_tb/flagwrite 0
/cpuA_tb/srcA
/cpuA_tb/srcB
/cpuA_tb/alucs 000
/cpuA_tb/s
/cpuA_tb/zero St0
Now


ModelSim - INTEL FPGA STARTER EDITION 2020.1
File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help
Start Simulation...
Runtime Options...
Run
Step
Restart...
Break
End Simulation
Run 100
Run -All
Continue
Run -Next
sim - Default
Instance
cpuA_tb
cpuA
#ALWAYS#63
#vsim_capacity#
Objects
Value
clk Not L... Net In
rst_n Not L... Net In
flagwrite Not L... Net In
srcA Not L... Net In
srcB Not L... Net In
alucs Not L... Net In
s 0000... Net Out
zero St0 Net Out
carry_in Not L... Net Internal
carry_out Not L... Net Internal
Processes (Active)
Name
Type (filtered)
Wave - Default
Msgs
/cpuA_tb/clk 1
/cpuA_tb/rst_n 0
/cpuA_tb/flagwrite 0
/cpuA_tb/srcA 00010110
/cpuA_tb/srcB 00001011
/cpuA_tb/alucs 000
/cpuA_tb/s 00000010
/cpuA_tb/zero St0
Now 40419120 ns
Cursor 1 0.00 ns
Library
sim
Transcript
cpuA
End time: 08:30:59 on Sep 27,2021, Elapsed time: 0:00:00