MIPS Single-cycle Processor

Assignment Outline

This assignment is to test your understanding of the MIPS-Lite single-cycle processor design presented in the lectures.

A sample Verilog code is available at Appendix A. You will need to add an Instruction memory to the desig[illegible]h your assembly code. The procedure for process is given in the Appen[illegible]

Currently the memory [illegible] the instruction memory ROM is empty. You can write the machine co[illegible]ons into the memory locations of the initialisation file.

The assignment is to [illegible] series of steps which you should complete in order. To ensure that it is your own work that is being assessed, Appendix C shows the specific requirements for each student.

1. Write a programme to: [15 Marks]

a. Load the data stored in the X and Y locations of the data memory into the X and Y registers.

b. Add the X and Y registers and store the result in the Z register.

c. Store the data from the Z register into the Z memory location.

d. Load the data in the Z memory location into the T register.

2. Simulate your program to show the contents of the X, Y, and T registers. [10 Marks]

3. Modify the *program.mif* file to simulate the operation of the **BEQ** instruction. [10 Marks]

4. Modify the design of the processor so that the **Jump** (J) instruction is implemented. [15 Marks] Modify the *program.mif* file to simulate the operation of the **Jump** instruction. [10 Marks]

5. Modify the designs so that the additional instructions given in the column “Design 1” is correctly executed. [15 Marks]

6. Modify the *program.mif* file to simulate the operation of the additional instruction. [10 Marks]

7. Change the processor’s data bus width from 32 to the one indicated in the table. Repeat the simulation of step 2 using your new processor. [15 Marks]

MIPS instructions’ reference data is provided.

程序代写代做 CS编程辅导
WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
https://tutorcs.com

Reports

Your report should include the following contents:

程序代写代做 CS编程辅导

1. The MIPS code and how the corresponding machine code is decided.

2. Simulation waveforms for the PC, opcode, ALUResultOut, DReadData and relevant registers (X, Y, T) mu[illegible]ou should clearly indicate why the simulations show that the operation is c[illegible]

2. Description of the m[illegible]e to the Verilog code for implementing the Jump instruction. Highlight [illegible] in the code.

3. Description of the modifications made to implement the additional instruction. Highlight the changes made in the code.

WeChat: cstutorcs

4. Description of the modifications made to change the data bus width.

Submission Date

Assignment Project Exam Help

A single PDF file should be submitted to Learning Mall by **2pm on Friday 15th December 2023 (Week 13)**. This file should be named as "surname_forename_A2.pdf". The contents in each **submission must be consistent with the tasks assigned** to a specific student; otherwise a zero mark will be awarded. There is no demonstration lab session for this assignment.

Email: tutorcs@163.com

QQ: 749389476

https://tutorcs.com

Appendix A

```
// MIPS single Cycle processor originaly developed for simulation by Patterson and
Hennesy
// Modified for synthesis using the QuartusII package by Dr. S. Ami-Nejad. Feb. 2009

// Register File
module RegisterFile [illegible] itereg,WriteData,RegWrite, Data1,
Data2,clock,reset);

    input  [4:0] Re[illegible] g; // the registers numbers to read or write
    input  [31:0] W[illegible]    // data to write
    input  RegWrite;              // The write control
    input  clock, reset;          // The clock to trigger writes
    output [31:0] Data1, Data2;   // the register values read;
    reg    [31:0] RF[31:0];       // 32 registers each 32 bits long
    integer   k;

    // Read from registers independent of clock
    assign Data1 = RF[Read1];
    assign Data2 = RF[Read2];
    // write the register with new value on the falling edge of the clock if RegWrite
is high
    always @(posedge clock or posedge reset)
        if (reset) for(k=0;k<32;k=k+1) RF[k]<=32'h00000000;
        // Register 0 is a read only register with the content of 0
        else   if (RegWrite & (Writereg!=0)) RF[Writereg] <= WriteData;
endmodule

//ALU Control
module ALUControl (ALUOp, FuncCode, ALUCtl);

    input  [1:0]  ALUOp;
    input  [5:0]  FuncCode;
    output [3:0]  ALUCtl;
    reg    [3:0]  ALUCtl;

    always@( ALUOp, FuncCode)
    begin
    case(ALUOp)
    2'b00: ALUCtl = 4'b0010;
    2'b01: ALUCtl = 4'b0110;
    2'b10: case(FuncCode)
            6'b 100000: ALUCtl = 4'b 0010;
            6'b 100010: ALUCtl = 4'b 0110;
            6'b 100100: ALUCtl = 4'b 0000;
            6'b 100101: ALUCtl = 4'b 0001;
            6'b 101010: ALUCtl = 4'b 0111;
```

程序代写代做 CS编程辅导
WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
https://tutorcs.com

```
                default:   ALUCtl = 4'b xxxx;
                endcase
        default:   ALUCtl = 4'b xxxx;
        endcase
        end
endmodule

//ALU
module MIPSALU (ALU[illegible]t, Zero);
        input   [3:0]  A[illegible]
        input   [31:0] A[illegible]
        output  [31:0] A[illegible]
        output  Zero;
        reg     [31:0] ALUOut;

        assign Zero = (ALUOut==0); //Zero is true if ALUOut is 0
        always @(ALUctl, A, B) begin //reevaluate if these change
        case (ALUctl)
            0: ALUOut <= A & B;
            1: ALUOut <= A | B;
            2: ALUOut <= A + B;
            6: ALUOut <= A - B;
            7: ALUOut <= A < B ? 1:0;
            // .... Add more ALU operations here
            default: ALUOut <= A;
            endcase
        end
endmodule

// Data Memory
module DataMemory(Address, DWriteData, MemRead, MemWrite, clock, reset, DReadData);
input   [31:0] Address, DWriteData;
input          MemRead, MemWrite, clock, reset;
output  [31:0] DReadData;
reg     [31:0] DMem[7:0];

assign  DReadData = DMem[Address[2:0]];
always @(posedge clock or posedge reset)begin
        if (reset) begin
            DMem[0]=32'h00000005;
            DMem[1]=32'h0000000A;
            DMem[2]=32'h00000055;
            DMem[3]=32'h000000AA;
            DMem[4]=32'h00005555;
            DMem[5]=32'h00008888;
            DMem[6]=32'h00550000;
            DMem[7]=32'h00004444;
            end else
```

程序代写代做 CS编程辅导
WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
https://tutorcs.com

```
            if (MemWrite) DMem[Address[2:0]] <= DWriteData;
        end
endmodule

// Main Controller
module Control
(opcode,RegDst,Bran[illegible]Reg,ALUOp,MemWrite,ALUSrc,RegWrite);

input  [5:0]  opco[illegible]
output [1:0]  ALUOp[illegible]
output RegDst,Branc[illegible]eg,MemWrite,ALUSrc,RegWrite;
reg    [1:0]  ALUOp[illegible]
reg    RegDst,Branch,MemRead,MemtoReg,MemWrite,ALUSrc,RegWrite;

parameter R_Format = 6'b000000, LW = 6'b100011, SW = 6'b101011, BEQ=6'b000100;
always @(opcode)begin
    case(opcode)
        R_Format:{RegDst,ALUSrc,MemtoReg,RegWrite,MemRead,MemWrite,Branch,ALUOp}=
9'b 100100010;
        LW:     {RegDst,ALUSrc,MemtoReg,RegWrite,MemRead,MemWrite,Branch,ALUOp}=
9'b 011110000;
        SW:     {RegDst,ALUSrc,MemtoReg,RegWrite,MemRead,MemWrite,Branch,ALUOp}=
9'b x1x001000;
        BEQ:    {RegDst,ALUSrc,MemtoReg,RegWrite,MemRead,MemWrite,Branch,ALUOp}=
9'b x0x000101;
        // .... Add more instructions here
        default: {RegDst,ALUSrc,MemtoReg,RegWrite,MemRead,MemWrite,Branch,ALUOp}=
9'b xxxxxxxxx;
        endcase
    end
endmodule

// Datapath
module DataPath(RegDst, Branch, MemRead, MemtoReg, ALUOp, MemWrite,
ALUSrc, RegWrite, clock, reset, opcode,/* RF1,  RF2, RF3,*/ALUResultOut ,DReadData);

input  RegDst,Branch,MemRead,MemtoReg,MemWrite,ALUSrc,RegWrite,clock, reset;
input  [1:0]  ALUOp;
output [5:0]  opcode;
output [31:0] /*RF1,  RF2, RF3,*/ ALUResultOut ,DReadData;

reg    [31:0] PC, IMemory[0:31];
wire   [31:0] SignExtendOffset, PCOffset, PCValue, ALUResultOut,
       IAddress, DAddress, IMemOut, DmemOut, DWriteData, Instruction,
       RWriteData, DReadData, ALUAin, ALUBin;
wire   [3:0] ALUctl;
wire   Zero;
wire   [4:0] WriteReg;
```

程序代写代做 CS编程辅导

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

https://tutorcs.com

```
//Instruction fields, to improve code readability
wire [5:0]    funct;
wire [4:0]    rs, rt, rd, shamt;
wire [15:0] offset;

//Instantiate local [illegible]
ALUControl alucontr[illegible]t,ALUctl);

// Instantiate ALU [illegible]
MIPSALU ALU(ALUctl, [illegible] ALUResultOut, Zero);

// Instantiate Register File
RegisterFile REG(rs, rt, WriteReg, RWriteData, RegWrite, ALUAin,
DWriteData,clock,reset);

// Instantiate Data Memory
DataMemory datamemory(ALUResultOut, DWriteData, MemRead, MemWrite, clock, reset,
DReadData);

// Instantiate Instruction Memory
IMemory    IMemory_inst (
    .address ( PC[6:2] ),
    .q ( Instruction )
    );

// Synthesize multiplexers
assign WriteReg    = (RegDst)           ? rd                 : rt;
assign ALUBin      = (ALUSrc)           ? SignExtendOffset   : DWriteData;
assign PCValue        = (Branch & Zero) ? PC+4+PCOffset   : PC+4;
assign RWriteData     = (MemtoReg)      ? DReadData          : ALUResultOut;

// Acquire the fields of the R_Format Instruction for clarity
assign {opcode, rs, rt, rd, shamt, funct} = Instruction;
// Acquire the immediate field of the I_Format instructions
assign offset = Instruction[15:0];
//sign-extend lower 16 bits
assign SignExtendOffset = { {16{offset[15]}} , offset[15:0]};
// Multiply by 4 the PC offset
assign PCOffset = SignExtendOffset << 2;
// Write the address of the next instruction into the program counter
always @(posedge clock ) begin
if (reset) PC<=32'h00000000; else
    PC <= PCValue;
end
endmodule

module MIPS1CYCLE(clock, reset,opcode, ALUResultOut ,DReadData);
```

程序代写代做 CS编程辅导
WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
https://tutorcs.com

```
    input  clock, reset;
    output [5:0]  opcode;
    output [31:0] ALUResultOut ,DReadData; // For simulation purposes

    wire [1:0] ALUOp;
    wire [5:0] opcode;
    wire [31:0] Sig[illegible]Out ,DReadData;
    wire RegDst,Bra[illegible]Reg,MemWrite,ALUSrc,RegWrite;

    // Instantiate [illegible]
    DataPath MIPSDP[illegible]MemRead,MemtoReg,ALUOp,
    MemWrite,ALUSrc[illegible]reset, opcode, ALUResultOut ,DReadData);

    //Instantiate the combinational control unit
    Control MIPSControl
(opcode,RegDst,Branch,MemRead,MemtoReg,ALUOp,MemWrite,ALUSrc,RegWrite);
endmodule
```

程序代写代做 CS编程辅导

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

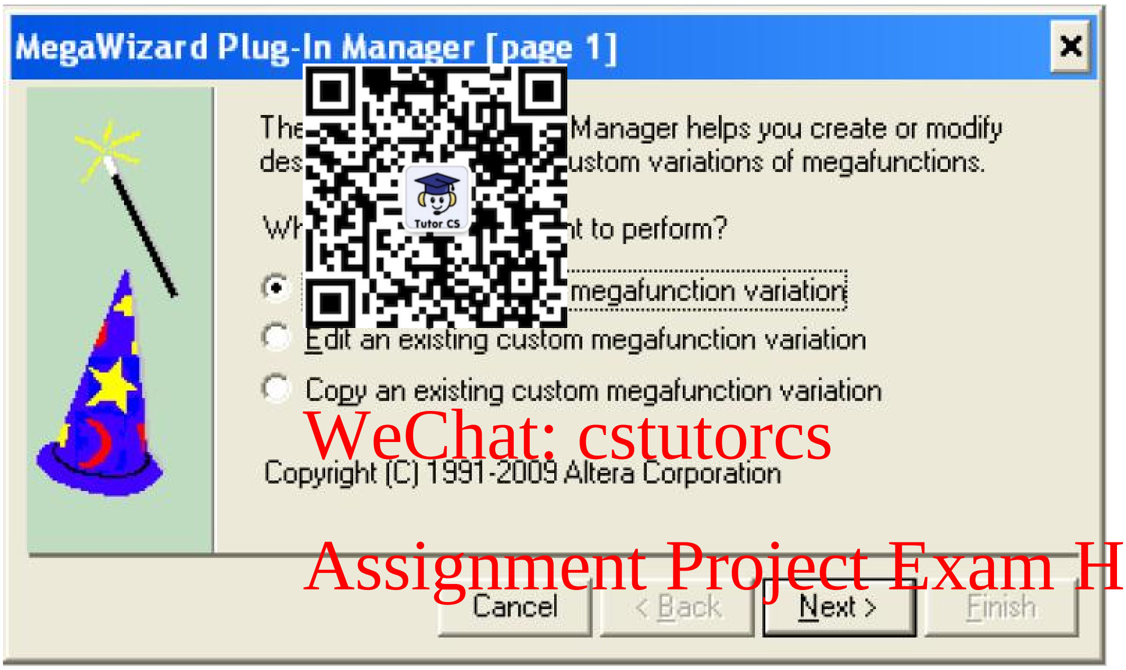
https://tutorcs.com

Appdendix B

Inserting LPM ROM

Select MegaWizard Plug-In Manager from the tools menu and click on the Next button.

程序代写代做 CS编程辅导



MegaWizard Plug-In Manager [page 1]

The Manager helps you create or modify des ustom variations of megafunctions.

Wh t to perform?

● megafunction variation

○ Edit an existing custom megafunction variation

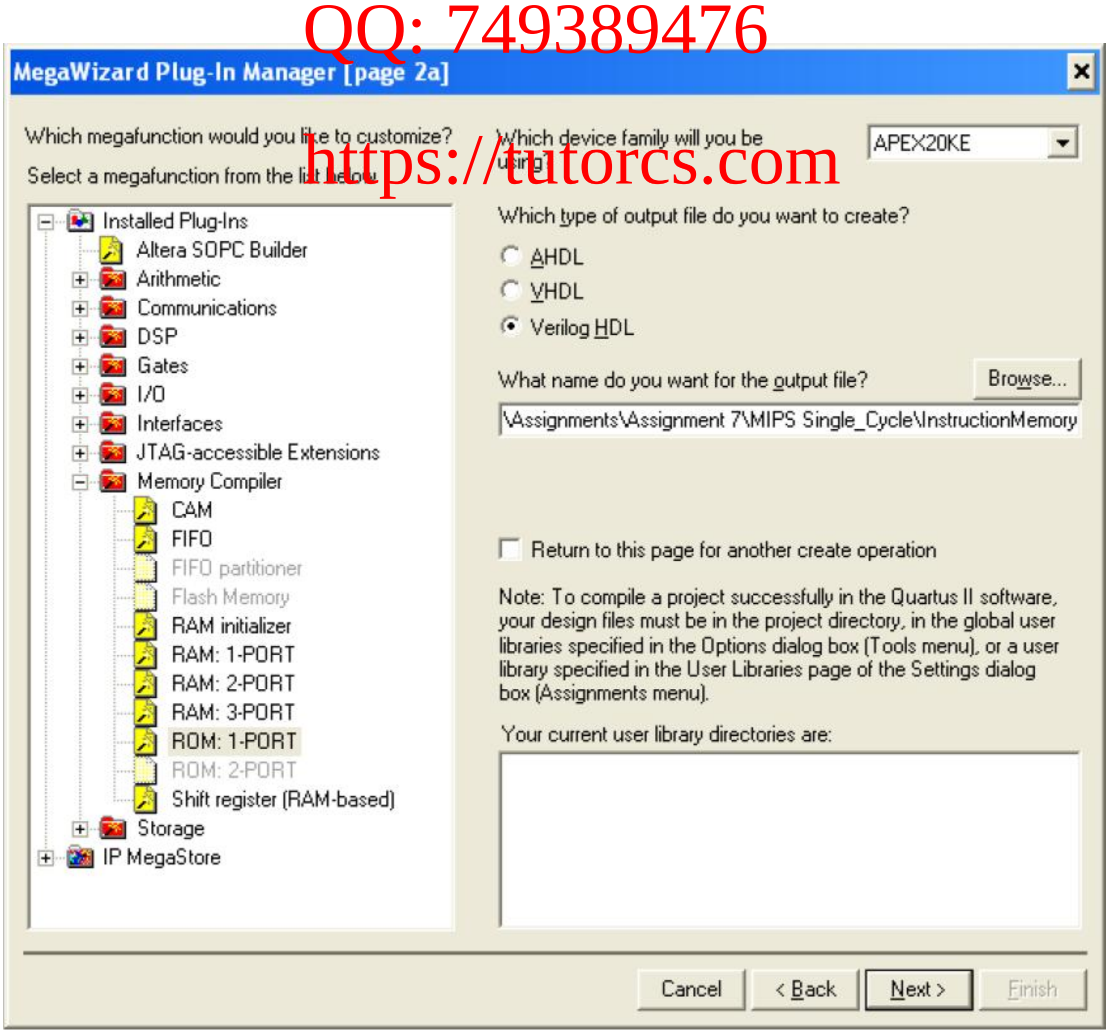
○ Copy an existing custom megafunction variation

Copyright (C) 1991-2009 Altera Corporation

WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com

Select the ROM: 1-PORT and give a name to the output file. This name should be the same as the one which has been used for Instruction memory instantiation in your Verilog code.

QQ: 749389476
https://tutorcs.com



MegaWizard Plug-In Manager [page 2a]

Which megafunction would you like to customize?
Select a megafunction from the list below

Which device family will you be using? APEX20KE

Which type of output file do you want to create?

○ AHDL

○ VHDL

● Verilog HDL

What name do you want for the output file? Browse...

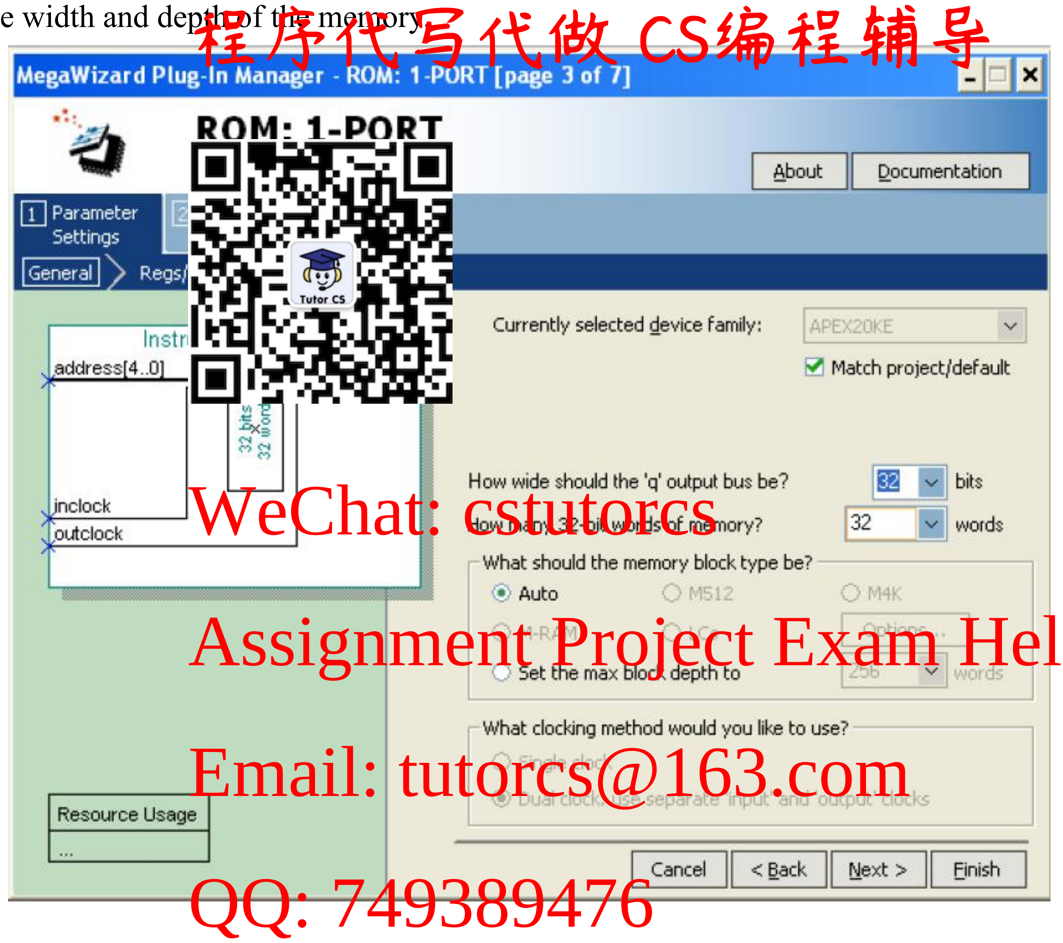
\Assignments\Assignment 7\MIPS Single_Cycle\InstructionMemory

□ Return to this page for another create operation

Note: To compile a project successfully in the Quartus II software, your design files must be in the project directory, in the global user libraries specified in the Options dialog box (Tools menu), or a user library specified in the User Libraries page of the Settings dialog box (Assignments menu).

Your current user library directories are:

Select the width and depth of the memory



MegaWizard Plug-In Manager - ROM: 1-PORT [page 3 of 7]

ROM: 1-PORT

1 Parameter Settings

General > Regs/

Currently selected device family: APEX20KE

☑ Match project/default

How wide should the 'q' output bus be? 32 bits

How many 32-bit words of memory? 32 words

What should the memory block type be?

● Auto ○ M512 ○ M4K

○ M-RAM ○ LCs

○ Set the max block depth to 256 words

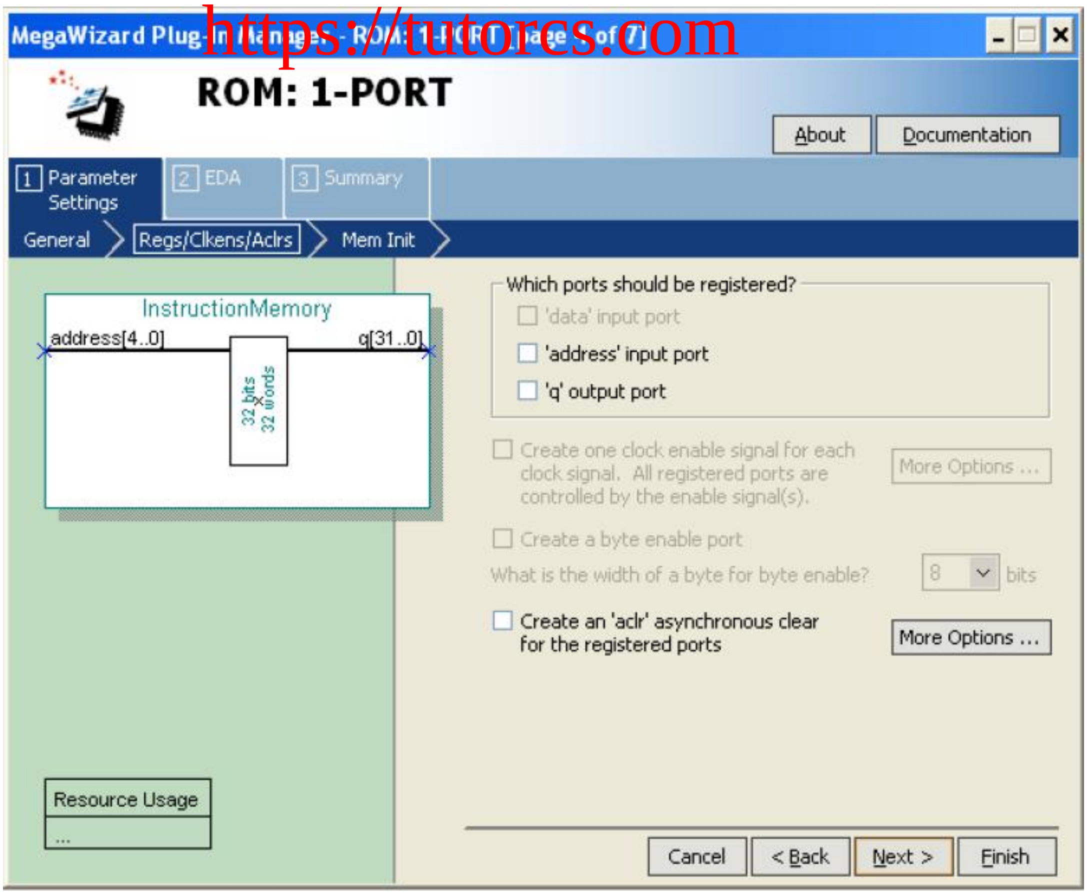
What clocking method would you like to use?

○ Single clock

● Dual clock: use separate 'input' and 'output' clocks

程序代写代做 CS编程辅导
WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
https://tutorcs.com

Deselect the input and output registers.



MegaWizard Plug-In Manager - ROM: 1-PORT [page 5 of 7]

ROM: 1-PORT

1 Parameter Settings | 2 EDA | 3 Summary

General > Regs/Clkens/Aclrs > Mem Init

Which ports should be registered?

□ 'data' input port

□ 'address' input port

□ 'q' output port

□ Create one clock enable signal for each clock signal. All registered ports are controlled by the enable signal(s).

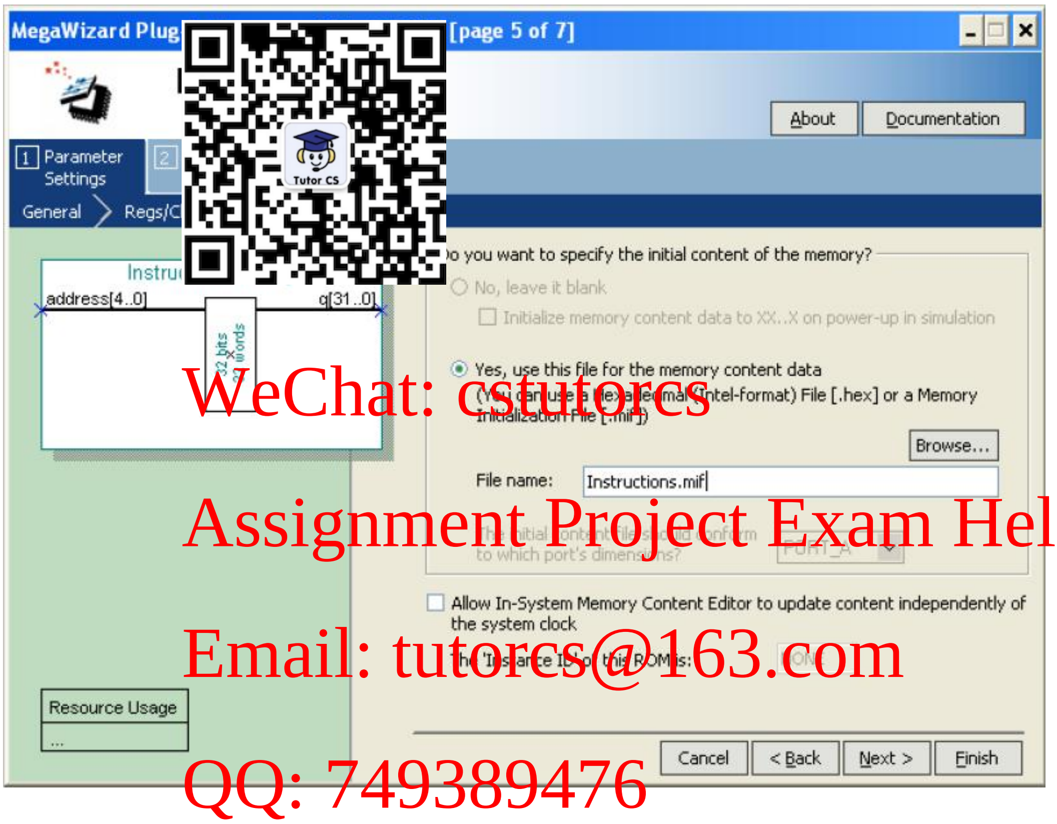
□ Create a byte enable port

What is the width of a byte for byte enable? 8 bits

□ Create an 'aclr' asynchronous clear for the registered ports

程序代写代做 CS编程辅导

Give a name to the memory initialisation file, e.g. Instruction.mif. You can use your name as the file name. In this case you should edit the Verilog module name in the provided code.



Do you want to specify the initial content of the memory?

○ No, leave it blank

□ Initialize memory content data to XX..X on power-up in simulation

● Yes, use this file for the memory content data
(You can use a Hexadecimal (Intel-format) File [.hex] or a Memory Initialization File [.mif])

File name: Instructions.mif

The initial content file should conform to which port's dimensions? PORT_A

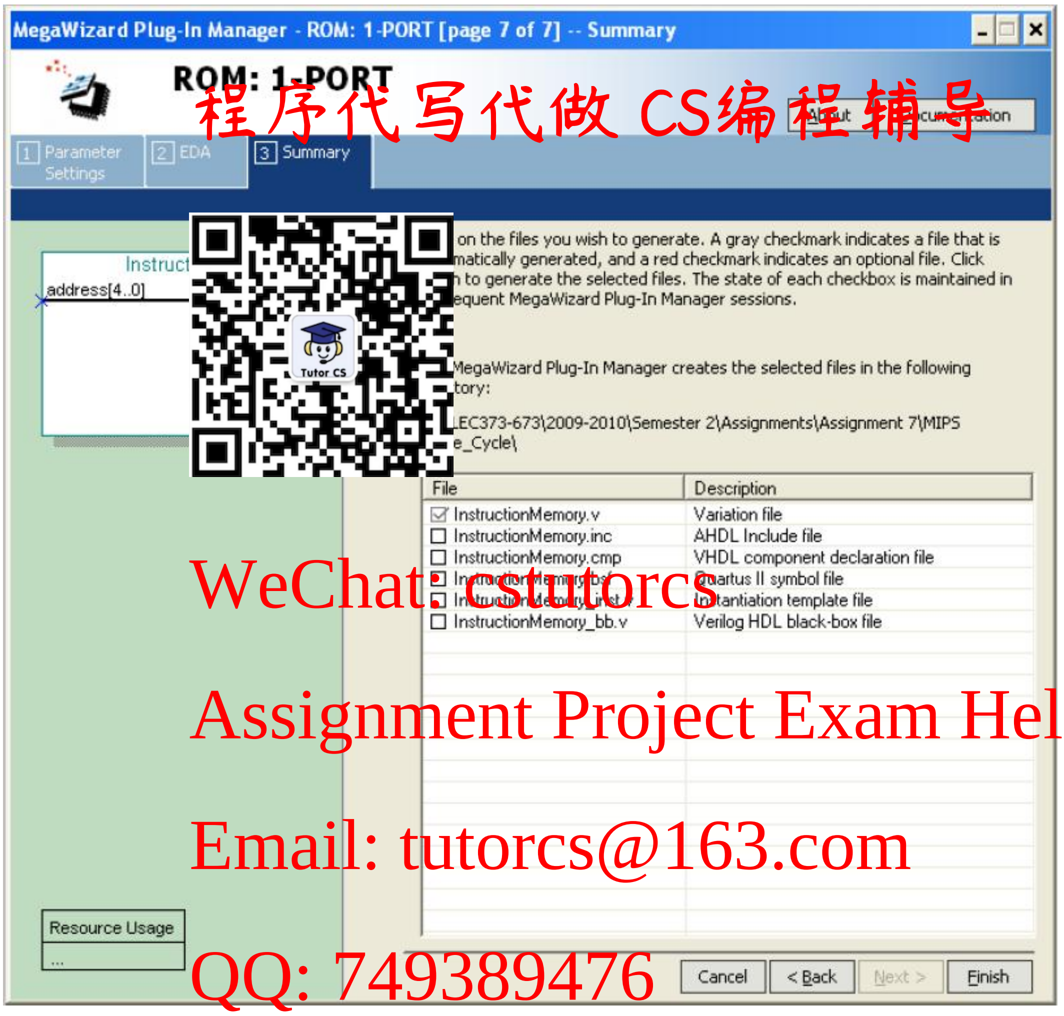
□ Allow In-System Memory Content Editor to update content independently of the system clock

The 'Instance ID' of this ROM is: NONE

WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476

Deselect the creation of the optional files and click on Finish button.

https://tutorcs.com



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

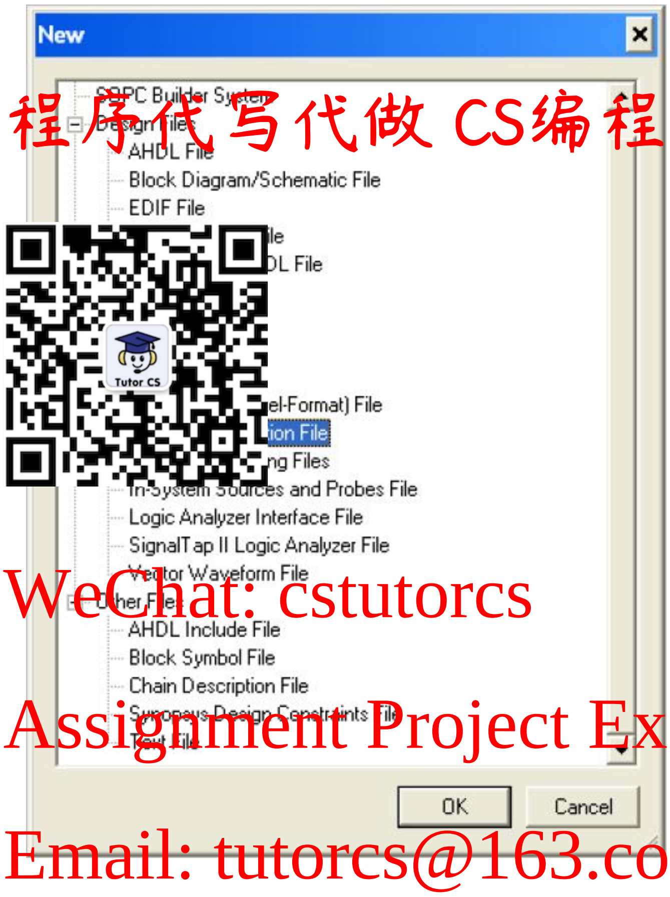
...MegaWizard Plug-In Manager creates the selected files in the following ...tory:

...LEC373-673\2009-2010\Semester 2\Assignments\Assignment 7\MIPS ...e_Cycle\

File	Description
☑ InstructionMemory.v	Variation file
☐ InstructionMemory.inc	AHDL Include file
☐ InstructionMemory.cmp	VHDL component declaration file
☑ InstructionMemory.bsf	Quartus II symbol file
☐ InstructionMemory_inst.v	Instantiation template file
☐ InstructionMemory_bb.v	Verilog HDL black-box file

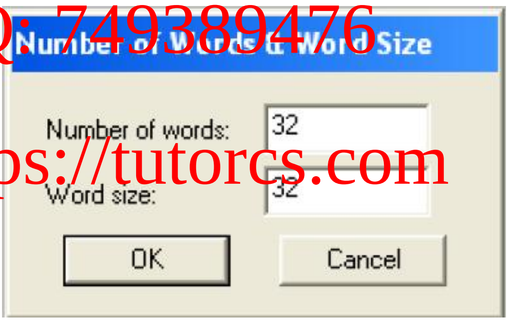
程序代写代做 CS编程辅导
WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
https://tutorcs.com

Click on the New icon and select the Memory Initialization File.



程序代写代做 CS编程辅导
WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
https://tutorcs.com

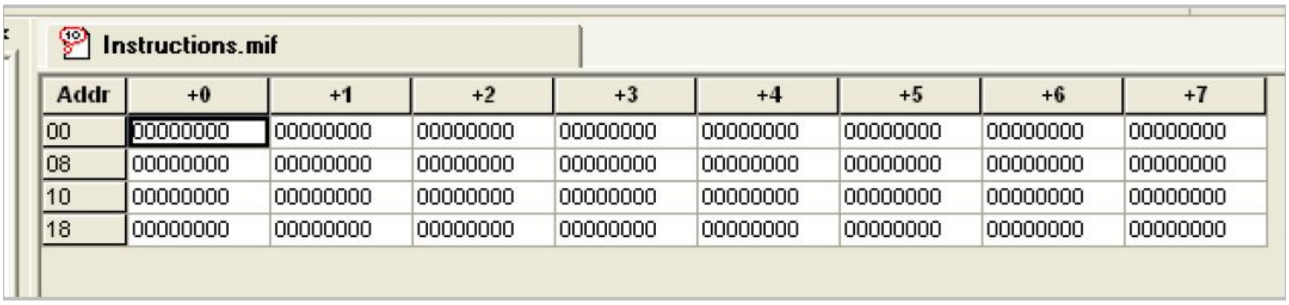
Input 32 for the Number of words and 32 for the Word size.


Number of Words & Word Size

Number of words: 32

Word size: 32

Now you can insert your machine code into the Instruction memory locations. You should be noted that this memory is not byte oriented and addresses are incremented by one not by 4.


Instructions.mif

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
08	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
10	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
18	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

Appendix C: The tasks for each student

程序代写代做 CS编程辅导

ID	English N[illegible]	Memory [illegible]s [illegible]T)	Data bus size	Design
2037696	A[illegible]	[illegible]	24	Ori
2033767	Bair[illegible]	[illegible]	28	Andi
2035143	Be[illegible]	[illegible]	20	Ori
1927872	B[illegible]	[illegible]	24	Andi
2033368	Boang Tian	5,6,7,1	28	Ori
2035925	Bolin Wang	2,3,4,5	20	Andi
2036029	Boran Zhu	6,7,1,2	24	Ori
2034136	Bowen Gu	3,4,5,6	28	Andi
2034115	Boyi Lu	7,1,2,3	20	Ori
2036108	Chang Huan	4,5,6,7	24	Andi
2036604	Cheng Jin	1,2,3,4	28	Ori
2033958	Chengxiao Zhu	5,6,7,1	20	Andi
1612332	Chengyi Wei	2,3,4,5	24	Ori
1927607	Chenming Shen	6,7,1,2	28	Andi
2034718	Chentao Zhang	3,4,5,6	20	Ori
2035284	Chenxi Zhang	7,1,2,3	24	Andi
2034672	Chenxin Yu	4,5,6,7	28	Ori
2035936	Chenxu Wei	1,2,3,4	20	Andi
2036637	Chongxiang Zhang	5,6,7,1	24	Ori
2035660	Dexuan Li	2,3,4,5	28	Andi
2037672	Dongtian Bai	6,7,1,2	20	Ori
2037645	Fanbin Lin	3,4,5,6	24	Andi
2033893	Fanxiang Chu	7,1,2,3	28	Ori
2038630	Feng Zhu	4,5,6,7	20	Andi
1824823	Guorui Xu	1,2,3,4	24	Ori
2035624	Guoxuan Sun	5,6,7,1	28	Andi
1929623	Haiyang Li	2,3,4,5	20	Ori
2036584	Hanyu Shi	6,7,1,2	24	Andi
1822806	Hao Dong	3,4,5,6	28	Ori
1824879	Haohan Zhang	7,1,2,3	20	Andi
2033955	Haoliang Zhu	4,5,6,7	24	Ori
2035427	Haoran Zhang	1,2,3,4	28	Andi
2033385	Haotao You	5,6,7,1	20	Ori
2034762	Haoxun Shen	2,3,4,5	24	Andi

WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
https://tutorcs.com

2036317	Haoyu Yang	6,7,1,2	28	Ori
2033854	Haozhong Zuo	3,4,5,6	20	Andi
1927818	Henry Li	7,1,2,3	24	Ori
1717106	Heyiyi Cheng	4,5,6,7	28	Andi
2145463	Hon Yu Andy Shing	1,2,3,4	20	Ori
2036585	Hu [illegible]	[illegible]	24	Andi
2032344	Ivan Au [illegible]	[illegible]	28	Ori
2032936	Janis [illegible] An [illegible]	[illegible]	20	Andi
1612508	Jia [illegible]	[illegible]	24	Ori
2035918	Jiahao Li	7,1,2,3	28	Andi
2034067	Jiaheng Zhang	4,5,6,7	20	Ori
2035086	Jiajie Huang	1,2,3,4	24	Andi
2033561	Jiakai Sun	5,6,7,1	28	Ori
1931380	Jiangchen Song	2,3,4,5	20	Andi
2035350	Jianzhi Yang	6,7,1,2	24	Ori
2036831	Jiarao Zhang	3,4,5,6	28	Andi
2034063	Jiatao Ma	7,1,2,3	20	Ori
2033379	Jiawei Tan	4,5,6,7	24	Andi
2035919	Jifeng Li	1,2,3,4	28	Ori
2033390	Jijia Zhang	5,6,7,1	20	Andi
1824838	Jinghang Ma	2,3,4,5	24	Ori
1929268	Junjie Qiu	6,7,1,2	28	Andi
2036019	Junzhe Tan	3,4,5,6	20	Ori
1927808	Kai Zhu	7,1,2,3	24	Andi
1823120	Kaiwen Du	4,5,6,7	28	Ori
1822280	Leyu Wang	1,2,3,4	20	Andi
2036660	Likun Fan	5,6,7,1	24	Ori
2035649	Lingyu Li	2,3,4,5	28	Andi
2035398	Liwei Jin	6,7,1,2	20	Ori
2033369	Luchang Liu	3,4,5,6	24	Andi
2034264	Lulu Song	7,1,2,3	28	Ori
2036597	Machao Yang	4,5,6,7	20	Andi
2035625	Man Ni	1,2,3,4	24	Ori
2036524	Minghao Tang	5,6,7,1	28	Andi
1825229	Mingjie Dong	2,3,4,5	20	Ori
2035959	Mingqiao Du	6,7,1,2	24	Andi
2037104	Mingtian Tan	3,4,5,6	28	Ori
1822925	Mingze Zhang	7,1,2,3	20	Andi
2037244	Mutian Yuan	4,5,6,7	24	Ori

程序代写代做 CS编程辅导
Tutor CS
WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
https://tutorcs.com

2035803	Peihao Huang	1,2,3,4	28	Andi
2037000	Qi An	5,6,7,1	20	Ori
2036940	Qi Zhang	2,3,4,5	24	Andi
2035451	Qian Cheng	6,7,1,2	28	Ori
2037287	Qianye Yang	3,4,5,6	20	Andi
2033881	Qih[illegible]	[illegible]	24	Ori
1929545	Qin[illegible]	[illegible]	28	Andi
2034930	Qi[illegible]	[illegible]	20	Ori
2036499	Qix[illegible]	[illegible]	24	Andi
2035344	Qua[illegible]	[illegible]	28	Ori
2036476	Quanhua Zeng	6,7,1,2	20	Andi
2036774	Rui Li	3,4,5,6	24	Ori
2033922	Rui Wang	7,1,2,3	28	Andi
2035811	Rui Wu	4,5,6,7	20	Ori
2035982	Ruiheng Liu	1,2,3,4	24	Andi
1825081	Ruixin Xu	5,6,7,1	28	Ori
2035507	Ruochen Qi	2,3,4,5	20	Andi
2032806	Sanjeev Thaiyal Bagam	6,7,1,2	24	Ori
2036493	Shiqiang Wu	3,4,5,6	28	Andi
2033903	Shuang Jiang	7,1,2,3	20	Ori
2037372	Shuyuan Zhang	4,5,6,7	24	Andi
2033595	Sixuan Li	1,2,3,4	28	Ori
2034719	Siyuan He	5,6,7,1	20	Andi
1929804	Siyuan Jiang	2,3,4,5	24	Ori
2036766	Songhao Li	6,7,1,2	28	Andi
1927559	Sunghu Moon	3,4,5,6	20	Ori
2033506	Tiancheng Cui	7,1,2,3	24	Andi
2033758	Tianwen Shen	4,5,6,7	28	Ori
2037925	Tianxing Xu	1,2,3,4	20	Andi
2034632	Tianyu Zhao	5,6,7,1	24	Ori
1614022	Weijian Zhang	2,3,4,5	28	Andi
2036648	Weiqin Xu	6,7,1,2	20	Ori
2036999	Wensheng Sun	3,4,5,6	24	Andi
2037025	Wenyi Dong	7,1,2,3	28	Ori
2035430	Xiang Yao	4,5,6,7	20	Andi
1927812	Xiaoliang Su	1,2,3,4	24	Ori
2035533	Xiaoou Jiang	5,6,7,1	28	Andi
2034204	Xingyu Chen	2,3,4,5	20	Ori
2037248	Xingze Xu	6,7,1,2	24	Andi
2035090	Xinlei Shu	3,4,5,6	28	Ori

程序代写代做 CS编程辅导
WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
https://tutorcs.com

2033772	Xiyin Xue	7,1,2,3	20	Andi
2034760	Yang Jin	4,5,6,7	24	Ori
2034129	Yang Xu	1,2,3,4	28	Andi
2033512	Yanhang Zhu	5,6,7,1	20	Ori
1930583	Yanwei Li	2,3,4,5	24	Andi
2035335	Ya[illegible]	[illegible]	28	Ori
2034171	Yao[illegible]	[illegible]	20	Andi
2036659	Yet[illegible]	[illegible]	24	Ori
2036641	Yi[illegible]	[illegible]	28	Andi
2034010	Yiha[illegible]	[illegible]	20	Ori
2034189	Yilin Cao	5,6,7,1	24	Andi
2036145	Yilin Yang	2,3,4,5	28	Ori
1613819	Yiming Huang	6,7,1,2	20	Andi
2034898	Yiming Wang	3,4,5,6	24	Ori
1715729	Yingkai Xu	7,1,2,3	28	Andi
2037237	Yingyi Cheng	4,5,6,7	20	Ori
2145498	Yiu Ting Wong	1,2,3,4	24	Andi
2036005	Yiwen Liu	5,6,7,1	28	Ori
2035381	Yixuan Huang	2,3,4,5	20	Andi
1715797	Yixuan Wang	6,7,1,2	24	Ori
2034777	Yiyang Lao	3,4,5,6	28	Andi
2035406	Yu Chen	7,1,2,3	20	Ori
2037909	Yuang Qiao	4,5,6,7	24	Andi
1929062	Yuanhan Zhou	1,2,3,4	28	Ori
1930219	Yuanzhen Liu	5,6,7,1	20	Andi
2037981	Yuchen Chen	2,3,4,5	24	Ori
2035048	Yuchen Lu	6,7,1,2	28	Andi
2036456	Yuhang Liu	3,4,5,6	20	Ori
2035998	Yuxi Wang	7,1,2,3	24	Andi
1930004	Yuxin Ding	4,5,6,7	28	Ori
2036027	Yuxing Sun	1,2,3,4	20	Andi
2035108	Ze Geng	5,6,7,1	24	Ori
2036116	Ze Zhang	2,3,4,5	28	Andi
2037225	Zehui Zhang	6,7,1,2	20	Ori
2033563	Zexi Peng	3,4,5,6	24	Andi
1928031	Zhan Wang	7,1,2,3	28	Ori
2033935	Zhangyan Heng	4,5,6,7	20	Andi
2033684	Zhen Qiu	1,2,3,4	24	Ori
1928961	Zhenbin Zhang	5,6,7,1	28	Andi
2036460	Zhenhua Liu	2,3,4,5	20	Ori

程序代写代做 CS编程辅导
Tutor CS
WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
QQ: 749389476
https://tutorcs.com

2037245	Zhenhua Mo	6,7,1,2	24	Andi
2035542	Zhewen Zhang	3,4,5,6	28	Ori
2035306	Zhijie Kong	7,1,2,3	20	Andi
2036997	Zhike Yang	4,5,6,7	24	Ori
2035981	Zhiyang Zhao	1,2,3,4	28	Andi
2035385	Zhiy	[illegible]	20	Ori
2036468	Zhon	[illegible]	24	Andi
1930302	Zhua	[illegible]	28	Ori
1929138	Zhuc	[illegible]	20	Andi
2035605	Ziji	[illegible]	24	Ori
2037685	Zijie Xu	4,5,6,7	28	Andi
2033763	Zijin Jiang	1,2,3,4	20	Ori
2035236	Ziqi Guo	5,6,7,1	24	Andi
2037436	Zirui Guo	2,3,4,5	28	Ori
2036679	Ziyan Wen	6,7,1,2	20	Andi

程序代写代做 CS编程辅导

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

https://tutorcs.com