ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KỸ THUẬT MÁY TÍNH



**BÁO CÁO CUỐI KỲ**

**MÔN THIẾT KẾ HỆ THÔNG SỐ VỚI HDL – CE213**

|  |  |
| --- | --- |
| **HỌ VÀ TÊN:** | **Vòng Chí Cường – 21521910** |
| **LỚP:** | **CE213.O11** |

**GIẢNG VIÊN HƯỚNG DẪN**

**Hồ Ngọc Diễm**

**TP. HỒ CHÍ MINH – Tháng 12 năm 2023**

MỤC LỤC

[MỤC LỤC 2](#_Toc154654552)

[DANH MỤC HÌNH ẢNH 4](#_Toc154654553)

[DANH MỤC BẢNG 5](#_Toc154654554)

[I. Introduction 6](#_Toc154654555)

[II. Implementation process 7](#_Toc154654556)

[III. Design workflow 8](#_Toc154654557)

[IV. Algorithm to multiply 2 matrices 9](#_Toc154654558)

[V. Top Modules 10](#_Toc154654559)

[VI. Data path 11](#_Toc154654560)

[VII. ISA 12](#_Toc154654561)

[*1. Instruction Set 12*](#_Toc154654562)

[*2. Memory Location 13*](#_Toc154654563)

[*3. Micro-Instruction 14*](#_Toc154654564)

[*4. State Diagram 16*](#_Toc154654565)

[VIII. Modules – RTL View 17](#_Toc154654566)

[*1. Module Bus 17*](#_Toc154654567)

[*2. Module Register 18*](#_Toc154654568)

[*3. Module incRegister 19*](#_Toc154654569)

[*4. Module ALU 20*](#_Toc154654570)

[*5. Module isZeroReg 20*](#_Toc154654571)

[*6. Module ControlUnit 21*](#_Toc154654572)

[*7. Module Processor 22*](#_Toc154654573)

[*8. Module MultiCore 23*](#_Toc154654574)

[*9. Module InsMemory 24*](#_Toc154654575)

[*10. Module DataMemory 24*](#_Toc154654576)

[*11. Module Top 25*](#_Toc154654577)

[IX. Test Bench 26](#_Toc154654578)

[*1. Module Bus\_tb 26*](#_Toc154654579)

[*2. Module Register\_tb 27*](#_Toc154654580)

[*3. Module incRegister\_tb 28*](#_Toc154654581)

[*4. Module ALU\_tb 29*](#_Toc154654582)

[*5. Module isZeroReg\_tb 30*](#_Toc154654583)

[*6. Module ControlUnit\_tb 31*](#_Toc154654584)

[*7. Module MultiCore\_tb 32*](#_Toc154654585)

[*8. Module DataMemory\_tb 34*](#_Toc154654586)

[*9. Module InsMemory\_tb 35*](#_Toc154654587)

[*10. Module Top\_tb 36*](#_Toc154654588)

# **DANH MỤC HÌNH ẢNH**

[*Hình 1 - Sơ đồ các modules 10*](#_Toc154654613)

[*Hình 2 – Data path 11*](#_Toc154654614)

[*Hình 3 - State Diagram 16*](file:///F:\lession\TapVerilog\MulticoreCPU\Document.docx#_Toc154654615)

[*Hình 4 - Module Bus 17*](#_Toc154654616)

[*Hình 5 - Module Register 18*](#_Toc154654617)

[*Hình 6 - Module Increment Register 19*](#_Toc154654618)

[*Hình 7 - Module ALU 20*](#_Toc154654619)

[*Hình 8 - Module isZeroRegister 20*](#_Toc154654620)

[*Hình 9 - Module Control Unit 21*](#_Toc154654621)

[*Hình 10 - Module Processor 22*](#_Toc154654622)

[*Hình 11 - Module Multi Core CPU 23*](#_Toc154654623)

[*Hình 12 - Module Instruction Memory 24*](#_Toc154654624)

[*Hình 13 - Module Data Memory 24*](#_Toc154654625)

[*Hình 14 - Module Top 25*](#_Toc154654626)

[*Hình 15 - Module Bus Test bench 26*](#_Toc154654627)

[*Hình 16 – Module Register Test bench 27*](#_Toc154654628)

[*Hình 17 – Module Increment Register Test bench 28*](#_Toc154654629)

[*Hình 18 - Module ALU Test bench 29*](#_Toc154654630)

[*Hình 19 – Module isZero Register Test bench 30*](#_Toc154654631)

[*Hình 20 - Module Control Unit test bench 31*](#_Toc154654632)

[*Hình 21- Test bench Multicore CPU 32*](#_Toc154654633)

[*Hình 22 - Test bench Data Memory 34*](#_Toc154654634)

[*Hình 23 - Test bench Instruction Memory 35*](#_Toc154654635)

[*Hình 24 - Test bench Top module 36*](#_Toc154654636)

# **DANH MỤC BẢNG**

[*Bảng 1 - Instruction Set 12*](#_Toc154654637)

[*Bảng 2- Memory Location 13*](#_Toc154654638)

[*Bảng 3 - Micro-Instruction 15*](#_Toc154654639)

[*Bảng 4 - Bảng giá trị Bus Test bench 26*](#_Toc154654640)

[*Bảng 5 - Bảng giá trị Register Test bench 27*](#_Toc154654641)

[*Bảng 6 – Bảng giá trị Increment Register Test bench 28*](#_Toc154654642)

[*Bảng 7 – Bảng giá trị ALU Test bench 29*](#_Toc154654643)

[*Bảng 8- Bảng giá trị isZero Register Test bench 30*](#_Toc154654644)

[*Bảng 9 - Bảng ma trận kết quả của test bench CPU 33*](#_Toc154654645)

[*Bảng 10 - Bảng giá trị test bench Data Memory 34*](#_Toc154654646)

[*Bảng 11 - Bảng giá trị test bench Instruction Memory 35*](#_Toc154654647)

1. Introduction

Một bộ xử lý đa nhân là một vi mạch xử lý trên một vi mạch tích hợp duy nhất với hai hoặc nhiều đơn vị xử lý riêng biệt được gọi là các lõi (ví dụ, hai lõi hoặc bốn lõi), mỗi lõi đọc và thực hiện các chỉ thị chương trình. Các chỉ thị này là các chỉ thị CPU thông thường (như cộng, chuyển động dữ liệu và nhảy nhánh) nhưng bộ xử lý đơn có thể thực hiện các chỉ thị trên các lõi riêng biệt cùng một lúc, tăng tốc độ tổng thể cho các chương trình hỗ trợ đa luồng hoặc các kỹ thuật tính toán song song khác. Thông thường, các nhà sản xuất tích hợp các lõi vào một viên mạch tích hợp duy nhất (được gọi là bộ xử lý đa lõi hoặc CMP) hoặc trên nhiều viên mạch tích hợp trong một gói chip duy nhất. Các vi xử lý hiện nay được sử dụng trong hầu hết tất cả các máy tính cá nhân đều là đa nhân.

1. Implementation process
2. Design workflow
3. Algorithm to multiply 2 matrices

Algorithm :

Initialization:

a <- row\_P; b <- column\_P(row\_Q); c <- column\_Q;

RP <- start\_P, RQ <- start\_Q

Mem[6] <- end\_P, Mem[7] <- end\_Q

Mem[8] <- start\_P, Mem[9] <- start\_Q, Mem[10] <- start\_R

AC, R, RC <- 0

for i = 0; i < a; i ++;

    for j = 0; j < c; j ++;

        output\_value = 0;

        for count = 0; count < b; count++;

            output\_value+ = matrixP[i][count] × matrixQ[count][j];

        save matrixR[i; j] <- output\_value;

1. Top Modules

Ảnh có chứa biểu đồ, văn bản, Kế hoạch, Bản vẽ kỹ thuật

Mô tả được tạo tự động

Hình 1 - Sơ đồ các modules

1. Data path

Ảnh có chứa văn bản, biểu đồ, Kế hoạch, Bản vẽ kỹ thuật

Mô tả được tạo tự động

Hình 2 – Data path

1. ISA
2. Instruction Set

|  |  |  |
| --- | --- | --- |
| **INSTRUCTION** | **OPCODE** | **OPERATION** |
| NOP | 0 | No operation |
| ENDOP | 1 | End operation |
| CLAC | 2 | AC 0 |
| LDIAC | 3 | AC dataMemfaddrg |
| LDAC | 4 | AC dataMemfACg |
| STR | 5 | dataMemfACg R |
| STIR | 6 | dataMemfaddrg R |
| JUMP | 7 | PC instructionMemfaddrg |
| JMPNZ | 8 | Jump if z!=0 |
| JMPZ | 9 | Jump if z==0 |
| MUL | 10 | AC AC\*R1 |
| ADD | 11 | AC AC+R |
| SUB | 12 | AC AC-RC |
| INCAC | 13 | AC AC+1 |
| MV RL AC | 14 | RL AC |
| MV RP AC | 15 | RP AC |
| MV RQ AC | 16 | RQ AC |
| MV RC AC | 17 | RC AC |
| MV R AC | 18 | R AC |
| MV R1 AC | 19 | R1 AC |
| MV AC RP | 20 | AC RP |
| MV AC RQ | 21 | AC RQ |
| MV AC RL | 22 | AC RL |

Bảng - Instruction Set

1. Memory Location

|  |  |  |
| --- | --- | --- |
| 0 | a | No. of rows of matrix P |
| 1 | b | No. of columns of matrix P (No. of rows of matrix Q) |
| 2 | c | No. of columns of matrix Q |
| 3 | start addres P | Start address of matrix P |
| 4 | start addr Q | Start address of matrix Q |
| 5 | start addr R | Start address of matrix R |
| 6 | end addr P | End address of matrix P |
| 7 | end addr Q | End address of matrix Q |
| 8 | current addr P | The address of currently processing data of matrix P |
| 9 | current addr Q | The address of currently processing data of matrix Q |
| 10 | current addr R | The address of currently processing data of matrix R |
| 11 | current c | Current row count |
| 12 | current a | Current column count |

Bảng - Memory Location

1. Micro-Instruction

|  |  |  |
| --- | --- | --- |
| INSTRUCTION | STATE | MICROINSTRUCTION |
| FETCH | FETCH\_DELAY1  FETCH1  FETCH2 | IR ← Instruction memory, Instruction read  IR ← Instruction memory, Instruction read  PC ← PC+1 |
| NOP | NOP1 | No operation |
| LDAC | LDAC1  LDAC\_DELAY1  LDAC2 | AR ← AC, Data read  R ← Data Memory  R ← Data Memory |
| LDIAC | LDIAC\_DELAY1  LDIAC1  LDIAC2  LDIAC\_DELAY2 LDIAC3 | IR ← Instruction Memory  IR ← Instruction Memory  AR ← IR, PC ← PC+1, Data read  AC ← Data Memory  AC ← Data Memory |
| STR | STR1  STR\_DELAY1  STR2 | AR ← AC, Data write  Data Memory ← R  Data Memory ← R |
| STIR | STIR\_DELAY1  STIR1  STIR2  STIR\_DELAY2  STIR3 | IR ← Instruction Memory  IR ← Instruction Memory  AR ← IR, PC ← PC+1, Data write  Data Memory ← R  Data Memory ← R |
| MV RL AC | MV\_RL\_AC1 | RL ← AC |
| MV R AC | MV\_R\_AC1 | R ← AC |
| MV RP AC | MV\_RP\_AC1 | RP ← AC |
| MV RQ AC | MV\_RQ\_AC1 | RQ ← AC |
| MV RC AC | MV\_RC\_AC1 | RC ← AC |
| MV R1 AC | MV\_R1\_AC1 | R1 ← AC |
| MV AC RP | MV\_AC\_RP1 | AC ← RP |
| MV AC RQ | MV\_AC\_RQ1 | AC ← RQ |
| MV AC RL | MV\_AC\_RL1 | AC ← RL |
| JUMP | JUMP\_DELAY1  JUMP1  JUMP2 | IR ← Instruction Memory  IR ← Instruction Memory  PC ← IR |
| JMPZ | JMPZY\_DELAY1  JMPZY1  JMPZY2  JMPZN1 | IR ← Instruction Memory  IR ← Instruction Memory  PC ← IR  PC ← PC+1 |
| JMPNZ | JMPNZY\_DILAY1  JMPNZY1  JMPNZY2  JMPNZN1 | IR ← Instruction Memory  IR ← Instruction Memory  PC ← IR  PC ← PC+1 |
| CLAC | CLAC1 | AC ← 0 |
| ADD | ADD1 | AC ← AC+R |
| SUB | SUB1 | AC ← AC-RC |
| MUL | MUL1 | AC ← AC\*R1, RP ← RP+1, RQ ← RQ+1,  RC ← RC+1 |
| INCAC | INCAC1 | AC ← AC+1 |
| ENDOP | ENDOP1 | End Operation |

Bảng - Micro-Instruction

Ảnh có chứa văn bản, biểu đồ, Kế hoạch, Bản vẽ kỹ thuật

Mô tả được tạo tự động

Hình - State Diagram

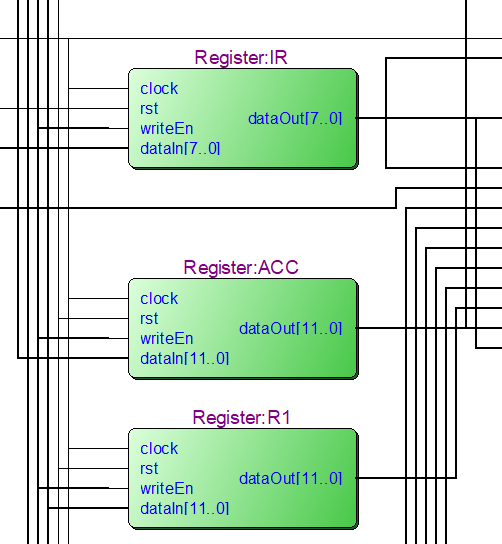
1. State Diagram
2. Modules – RTL View
3. Module Bus

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

Hình 4 - Module Bus

1. Module Register



Hình 5 - Module Register

1. Module incRegister

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

Hình 6 - Module Increment Register

1. Module ALU

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

Hình 7 - Module ALU

1. Module isZeroReg

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, biểu đồ

Mô tả được tạo tự động

Hình 8 - Module isZeroRegister

1. Module ControlUnit

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, biểu đồ

Mô tả được tạo tự động

Hình 9 - Module Control Unit

1. Module Processor

Ảnh có chứa biểu đồ, ảnh chụp màn hình, Kế hoạch, Bản vẽ kỹ thuật

Mô tả được tạo tự động

Hình 10 - Module Processor

1. Module MultiCore

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

Hình 11 - Module Multi Core CPU

1. Module InsMemory

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

Hình 12 - Module Instruction Memory

1. Module DataMemory

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

Hình 13 - Module Data Memory

1. Module Top

Ảnh có chứa biểu đồ, văn bản, Kế hoạch, hàng

Mô tả được tạo tự động

Hình 14 - Module Top

1. Test Bench
2. Module Bus\_tb

Ảnh có chứa ảnh chụp màn hình, Phần mềm đa phương tiện, phần mềm, Phần mềm đồ họa

Mô tả được tạo tự động

Hình 15 - Module Bus Test bench

|  |  |  |  |
| --- | --- | --- | --- |
| # Time = 10 | Test case 1: DataMem\_sel | selectIn = 0000 | busOut = 123 |
| # Time = 20 | Test case 2: R\_sel | selectIn = 0001 | busOut = 456 |
| # Time = 30 | Test case 3: IR\_sel | selectIn = 0010 | busOut = 0aa |
| # Time = 40 | Test case 4: RL\_sel | selectIn = 0011 | busOut = 789 |
| # Time = 50 | Test case 5: RC\_sel | selectIn = 0100 | busOut = abc |
| # Time = 60 | Test case 6: RP\_sel | selectIn = 0101 | busOut = def |
| # Time = 70 | Test case 7: RQ\_sel | selectIn = 0110 | busOut = f00 |
| # Time = 80 | Test case 8: R1\_sel | selectIn = 0111 | busOut = fed |
| # Time = 90 | Test case 9: ACC\_sel | selectIn = 1000 | busOut = 987 |
| # Time = 100 | Test case 10: idle | selectIn = 1001 | busOut = 000 |

Bảng 4 - Bảng giá trị Bus Test bench

1. Module Register\_tb

Ảnh có chứa ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện, Phần mềm đồ họa

Mô tả được tạo tự động

Hình 16 – Module Register Test bench

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # Time = 10 | Test case 1: Reset | dataIn = 123 | writeEn = 0 | rst = 1 | dataOut = 000 |
| # Time = 20 | Test case 2: Write | dataIn = 456 | writeEn = 1 | rst = 0 | dataOut = 456 |
| # Time = 30 | Test case 3: No write | dataIn = 789 | writeEn = 0 | rst = 0 | dataOut = 456 |

Bảng 5 - Bảng giá trị Register Test bench

1. Module incRegister\_tb

Ảnh có chứa ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện, Phần mềm đồ họa

Mô tả được tạo tự động

Hình 17 – Module Increment Register Test bench

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # Time = 10 | Test case 1: Write operation | dataIn = 123 | writeEn = 1 | rst = 0 | incEn = 0 | dataOut = 123 |
| # Time = 20 | Test case 2: Increment operation | dataIn = 000 | writeEn = 0 | rst = 0 | incEn = 1 | dataOut = 124 |
| # Time = 30 | Test case 3: Reset operation | dataIn = 000 | writeEn = 0 | rst = 1 | incEn = 0 | dataOut = 000 |

Bảng 6 – Bảng giá trị Increment Register Test bench

1. Module ALU\_tb

Ảnh có chứa ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện, Phần mềm đồ họa

Mô tả được tạo tự động

Hình 18 - Module ALU Test bench

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # Time = 10 | Test case 1: clr | A = 123 | B = 456 | ALUop = 000 | Result = 000 |
| # Time = 20 | Test case 2: pass | A = 123 | B = 456 | ALUop = 001 | Result = 456 |
| # Time = 30 | Test case 3: add | A = 123 | B = 456 | ALUop = 010 | Result = 579 |
| # Time = 40 | Test case 4: sub | A = 123 | B = 456 | ALUop = 011 | Result = ccd |
| # Time = 50 | Test case 5: mul | A = 123 | B = 456 | ALUop = 100 | Result = dc2 |
| # Time = 60 | Test case 6: inc | A = 123 | B = 456 | ALUop = 101 | Result = 124 |
| # Time = 70 | Test case 7: idle | A = 123 | B = 456 | ALUop = 110 | Result = 000 |

Bảng 7 – Bảng giá trị ALU Test bench

1. Module isZeroReg\_tb

Ảnh có chứa ảnh chụp màn hình, Phần mềm đa phương tiện, phần mềm, Phần mềm đồ họa

Mô tả được tạo tự động

Hình 19 – Module isZero Register Test bench

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # Time = 10 | Test case 1: Reset | dataIn = 123 | writeEn = 0 | rst = 1 | dataOut = 0 |
| # Time = 20 | Test case 2: Write 0 | dataIn = 000 | writeEn = 1 | rst = 0 | dataOut = 1 |
| # Time = 30 | Test case 3: Write non-zero | dataIn = 456 | writeEn = 1 | rst = 0 | dataOut = 0 |
| # Time = 40 | Test case 4: No write | dataIn = 789 | writeEn = 0 | rst = 0 | dataOut = 0 |

Bảng 8- Bảng giá trị isZero Register Test bench

1. Module ControlUnit\_tb

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Hình - Module Control Unit test bench

1. Module MultiCore\_tb

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Hình - Test bench Multicore CPU

|  |
| --- |
| # Matrix P |
| # 004 004 005 002 005 |
| # 001 003 005 005 005 |
| # 004 003 004 001 001 |
| # 003 000 001 001 003 |
| # 000 005 001 001 004 |
| # 003 003 004 001 003 |
| # 003 000 004 002 001 |
| # Matrix Q |
| # 008 008 006 000 003 00a |
| # 004 003 003 00a 004 000 |
| # 000 009 007 003 008 007 |
| # 007 005 00a 000 002 002 |
| # 004 007 005 001 002 007 |
| # Matrix R |
| # 052 086 074 03c 052 072 |
| # 04b 07a 07d 032 04b 05a |
| # 037 059 04c 02b 03c 04d |
| # 02b 03b 032 006 019 03c |
| # 02b 039 034 039 026 025 |
| # 037 05f 050 02d 03d 051 |
| # 02a 04d 047 00d 02f 045 |

Bảng - Bảng ma trận kết quả của test bench CPU

1. Module DataMemory\_tb

Ảnh có chứa phần mềm, Phần mềm đa phương tiện, Phần mềm đồ họa, văn bản

Mô tả được tạo tự động

Hình - Test bench Data Memory

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # clock=1 | writeEn=1 | address=0 | dataIn=abc | dataOut=abc | processDone=x |
| # clock=1 | writeEn=1 | address=1 | dataIn=123 | dataOut=123 | processDone=x |
| # clock=1 | writeEn=0 | address=1 | dataIn=123 | dataOut=123 | processDone=x |

Bảng - Bảng giá trị test bench Data Memory

1. Module InsMemory\_tb

Ảnh có chứa ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện, Phần mềm đồ họa

Mô tả được tạo tự động

Hình - Test bench Instruction Memory

|  |  |  |
| --- | --- | --- |
| # Write: writeEn=1 | address=00 | dataIn=da |
| # Write: writeEn=1 | address=01 | dataIn=33 |
| # Read: writeEn=0 | address=01 | dataOut=33 |

Bảng - Bảng giá trị test bench Instruction Memory

1. Module Top\_tb

Ảnh có chứa ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện, Phần mềm đồ họa

Mô tả được tạo tự động

Hình - Test bench Top module

**--- Hết ---**