

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 THỐNG 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

| | |
|---|----|
| <i>MỤC LỤC</i> | 2 |
| <i>DANH MỤC HÌNH ẢNH</i> | 4 |
| <i>DANH MỤC BẢNG</i> | 5 |
| <i>I. Introduction</i> | 6 |
| <i>II. Implementation process</i> | 7 |
| <i>III. Design workflow</i> | 8 |
| <i>IV. Algorithm to multiply 2 matrices</i> | 9 |
| <i>V. Top Modules</i> | 10 |
| <i>VI. Data path</i> | 11 |
| <i>VII. ISA</i> | 12 |
| 1. <i>Instruction Set</i> | 12 |
| 2. <i>Memory Location</i> | 13 |
| 3. <i>Micro-Instruction</i> | 14 |
| 4. <i>State Diagram</i> | 16 |
| <i>VIII. Modules – RTL View</i> | 17 |
| 1. <i>Module Bus</i> | 17 |
| 2. <i>Module Register</i> | 18 |
| 3. <i>Module incRegister</i> | 19 |
| 4. <i>Module ALU</i> | 20 |
| 5. <i>Module isZeroReg</i> | 20 |
| 6. <i>Module ControlUnit</i> | 21 |
| 7. <i>Module Processor</i> | 22 |
| 8. <i>Module MultiCore</i> | 23 |
| 9. <i>Module InsMemory</i> | 24 |
| 10. <i>Module DataMemory</i> | 24 |

| | | |
|-----|------------------------------------|----|
| 11. | <i>Module Top</i> | 25 |
| IX. | <i>Test Bench</i> | 26 |
| 1. | <i>Module Bus_tb</i> | 26 |
| 2. | <i>Module Register_tb</i> | 27 |
| 3. | <i>Module incRegister_tb</i> | 28 |
| 4. | <i>Module ALU_tb</i> | 29 |
| 5. | <i>Module isZeroReg_tb</i> | 30 |
| 6. | <i>Module ControlUnit_tb</i> | 31 |
| 7. | <i>Module MultiCore_tb</i> | 32 |
| 8. | <i>Module DataMemory_tb</i> | 34 |
| 9. | <i>Module InsMemory_tb</i> | 35 |
| 10. | <i>Module Top_tb</i> | 36 |

DANH MỤC HÌNH ẢNH

| | |
|--|-----------|
| <i>Hình 1 - Sơ đồ các modules</i> | <i>10</i> |
| <i>Hình 2 – Data path</i> | <i>11</i> |
| <i>Hình 3 - State Diagram.....</i> | <i>16</i> |
| <i>Hình 4 - Module Bus</i> | <i>17</i> |
| <i>Hình 5 - Module Register.....</i> | <i>18</i> |
| <i>Hình 6 - Module Increment Register</i> | <i>19</i> |
| <i>Hình 7 - Module ALU</i> | <i>20</i> |
| <i>Hình 8 - Module isZeroRegister</i> | <i>20</i> |
| <i>Hình 9 - Module Control Unit</i> | <i>21</i> |
| <i>Hình 10 - Module Processor.....</i> | <i>22</i> |
| <i>Hình 11 - Module Multi Core CPU</i> | <i>23</i> |
| <i>Hình 12 - Module Instruction Memory</i> | <i>24</i> |
| <i>Hình 13 - Module Data Memory.....</i> | <i>24</i> |
| <i>Hình 14 - Module Top.....</i> | <i>25</i> |
| <i>Hình 15 - Module Bus Test bench.....</i> | <i>26</i> |
| <i>Hình 16 – Module Register Test bench.....</i> | <i>27</i> |
| <i>Hình 17 – Module Increment Register Test bench.....</i> | <i>28</i> |
| <i>Hình 18 - Module ALU Test bench</i> | <i>29</i> |
| <i>Hình 19 – Module isZero Register Test bench.....</i> | <i>30</i> |
| <i>Hình 20 - Module Control Unit test bench</i> | <i>31</i> |
| <i>Hình 21- Test bench Multicore CPU</i> | <i>32</i> |
| <i>Hình 22 - Test bench Data Memory.....</i> | <i>34</i> |
| <i>Hình 23 - Test bench Instruction Memory</i> | <i>35</i> |
| <i>Hình 24 - Test bench Top module</i> | <i>36</i> |

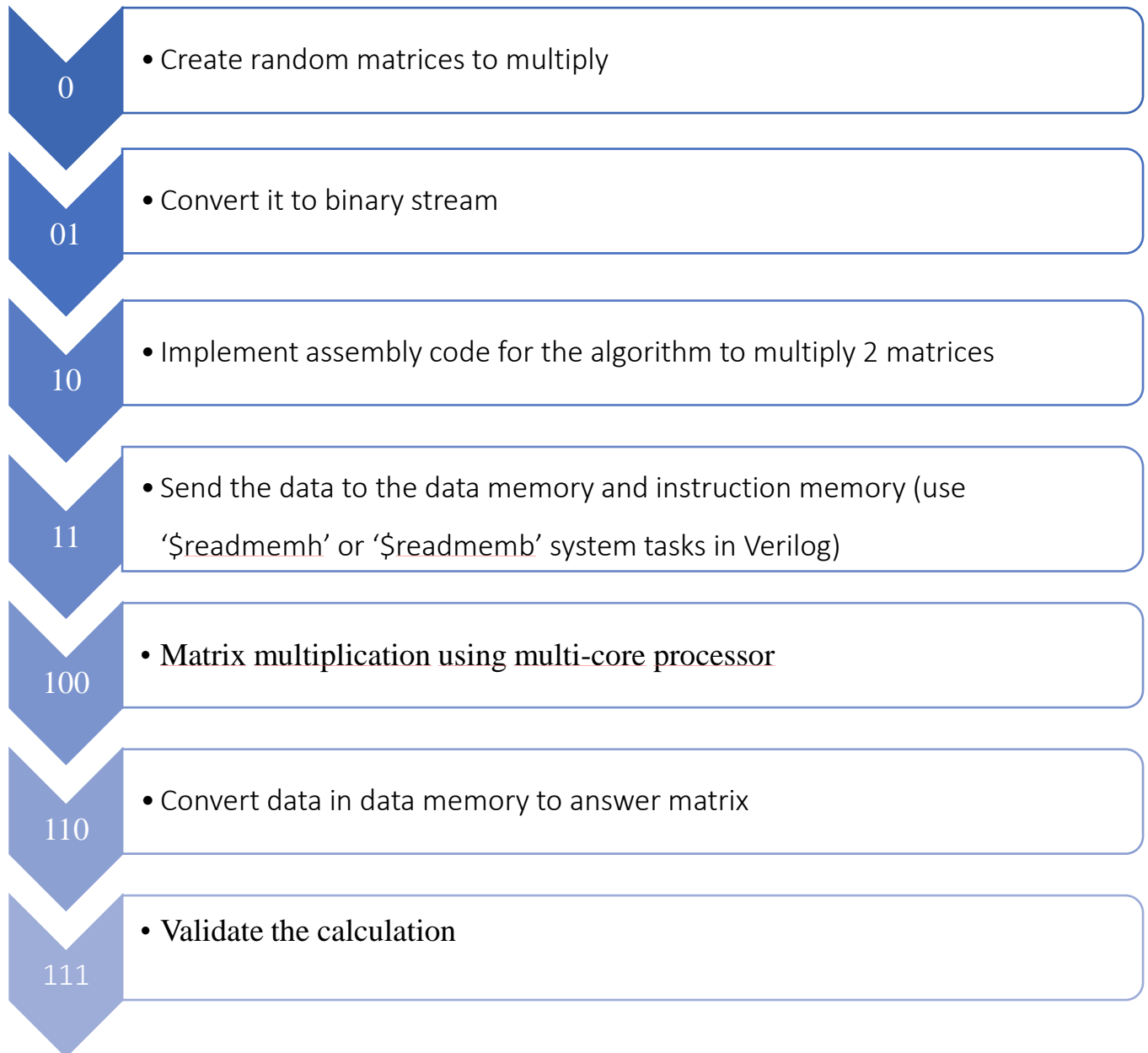
DANH MỤC BẢNG

| | |
|---|-----------|
| <i>Bảng 1 - Instruction Set</i> | <i>12</i> |
| <i>Bảng 2- Memory Location</i> | <i>13</i> |
| <i>Bảng 3 - Micro-Instruction</i> | <i>15</i> |
| <i>Bảng 4 - Bảng giá trị Bus Test bench</i> | <i>26</i> |
| <i>Bảng 5 - Bảng giá trị Register Test bench.....</i> | <i>27</i> |
| <i>Bảng 6 – Bảng giá trị Increment Register Test bench</i> | <i>28</i> |
| <i>Bảng 7 – Bảng giá trị ALU Test bench</i> | <i>29</i> |
| <i>Bảng 8- Bảng giá trị isZero Register Test bench.....</i> | <i>30</i> |
| <i>Bảng 9 - Bảng ma trận kết quả của test bench CPU</i> | <i>33</i> |
| <i>Bảng 10 - Bảng giá trị test bench Data Memory</i> | <i>34</i> |
| <i>Bảng 11 - Bảng giá trị test bench Instruction Memory</i> | <i>35</i> |

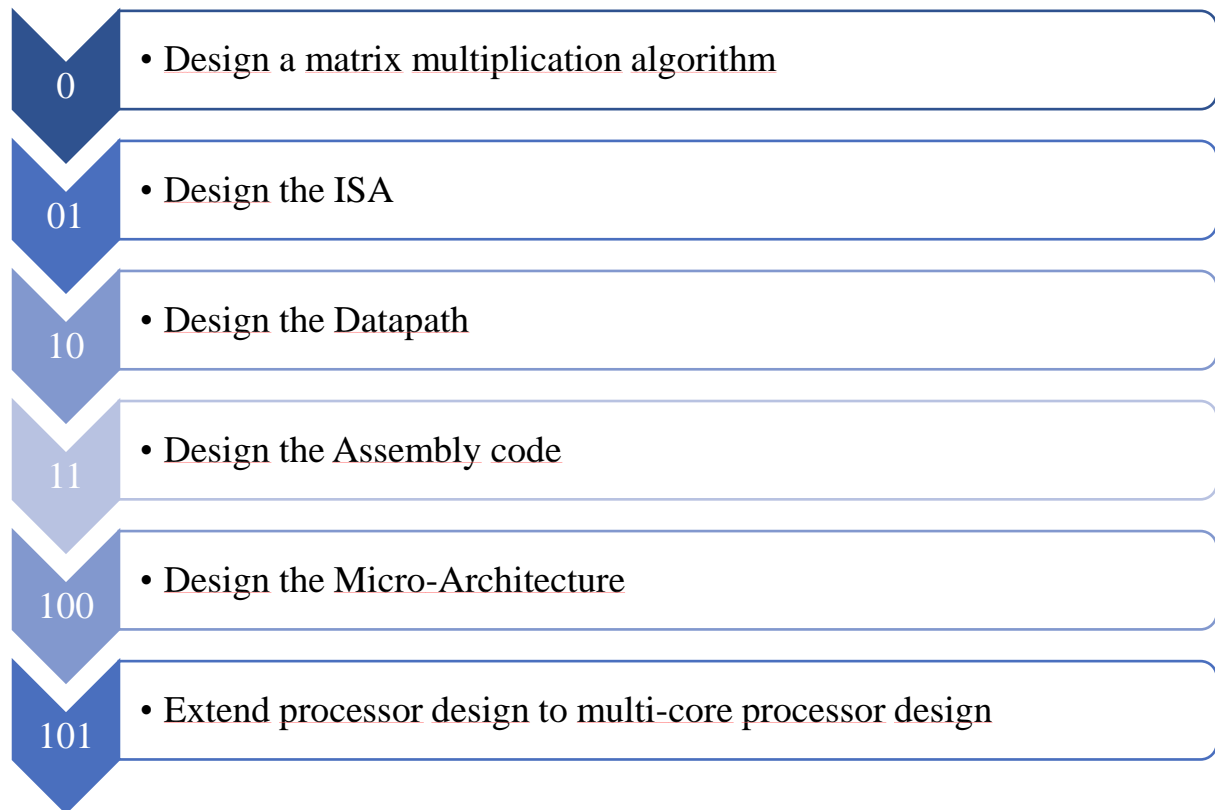
I. Introduction

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.

II. Implementation process



III. Design workflow



IV. 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 ++;
```

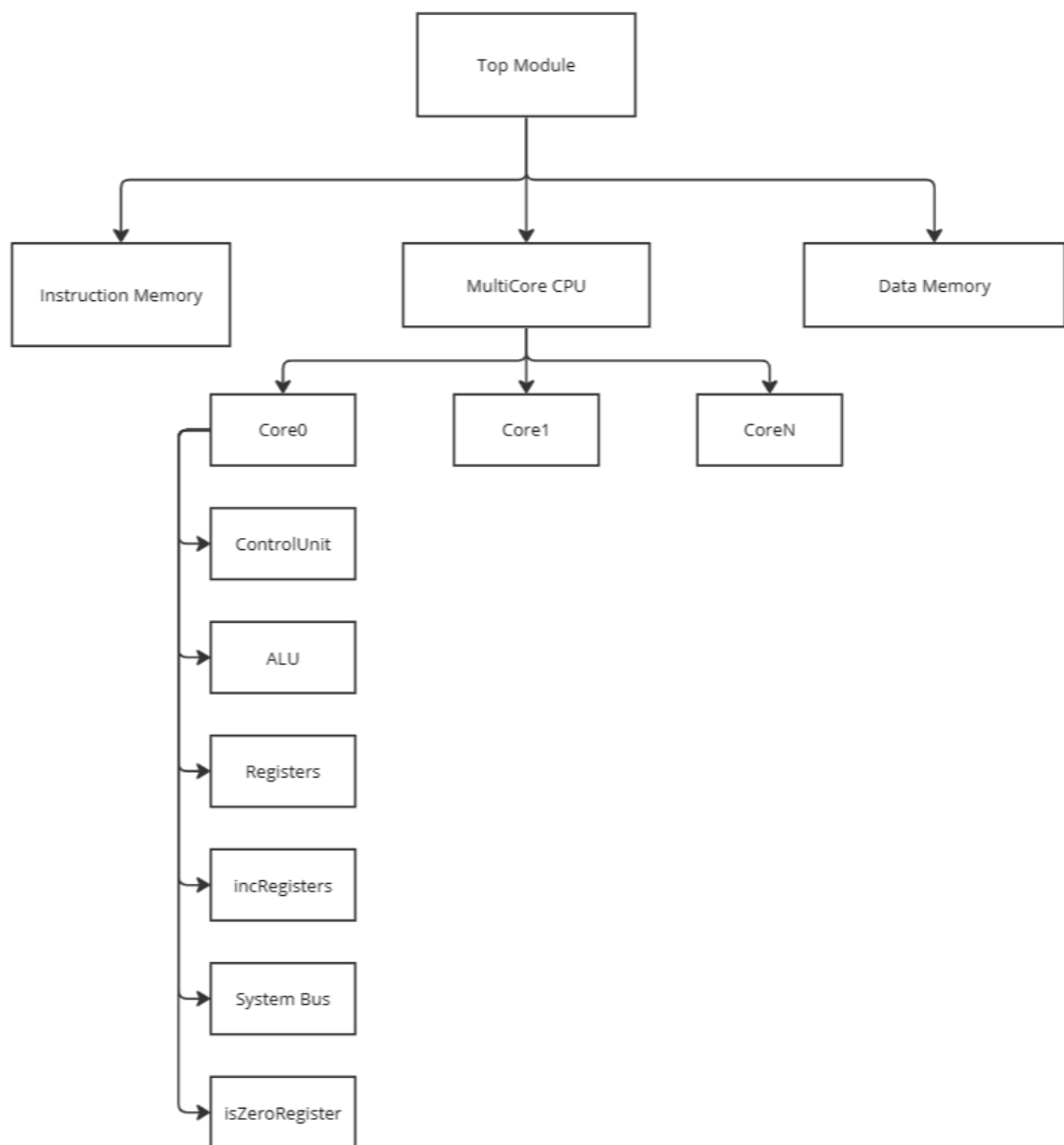
```
        output_value = 0;
```

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

```
            output_value += matrixP[i][count] × matrixQ[count][j];
```

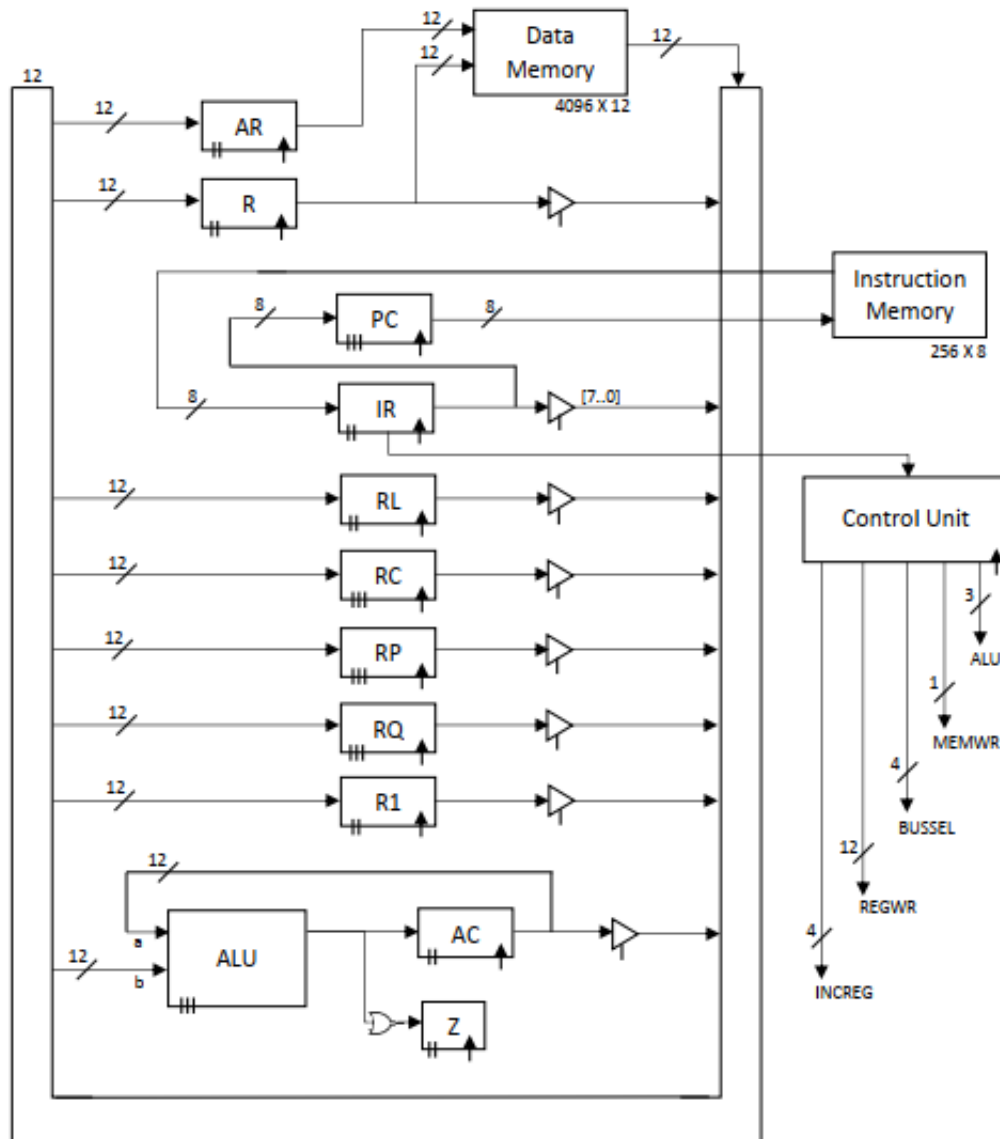
```
        save matrixR[i; j] <- output_value;
```

V. Top Modules



Hình 1 - Sơ đồ các modules

VI. Data path



Hình 2 – Data path

VII. ISA

1. Instruction Set

| INSTRUCTION | OPCODE | OPERATION |
|-------------|--------|-------------------------------------|
| NOP | 0 | No operation |
| ENDOP | 1 | End operation |
| CLAC | 2 | AC 0 |
| LDIAC | 3 | AC dataMemfaddr _g |
| LDAC | 4 | AC dataMemfAC _g |
| STR | 5 | dataMemfAC _g R |
| STIR | 6 | dataMemfaddr _g R |
| JUMP | 7 | PC instructionMemfaddr _g |
| 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 1 - Instruction Set

2. 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 2- Memory Location

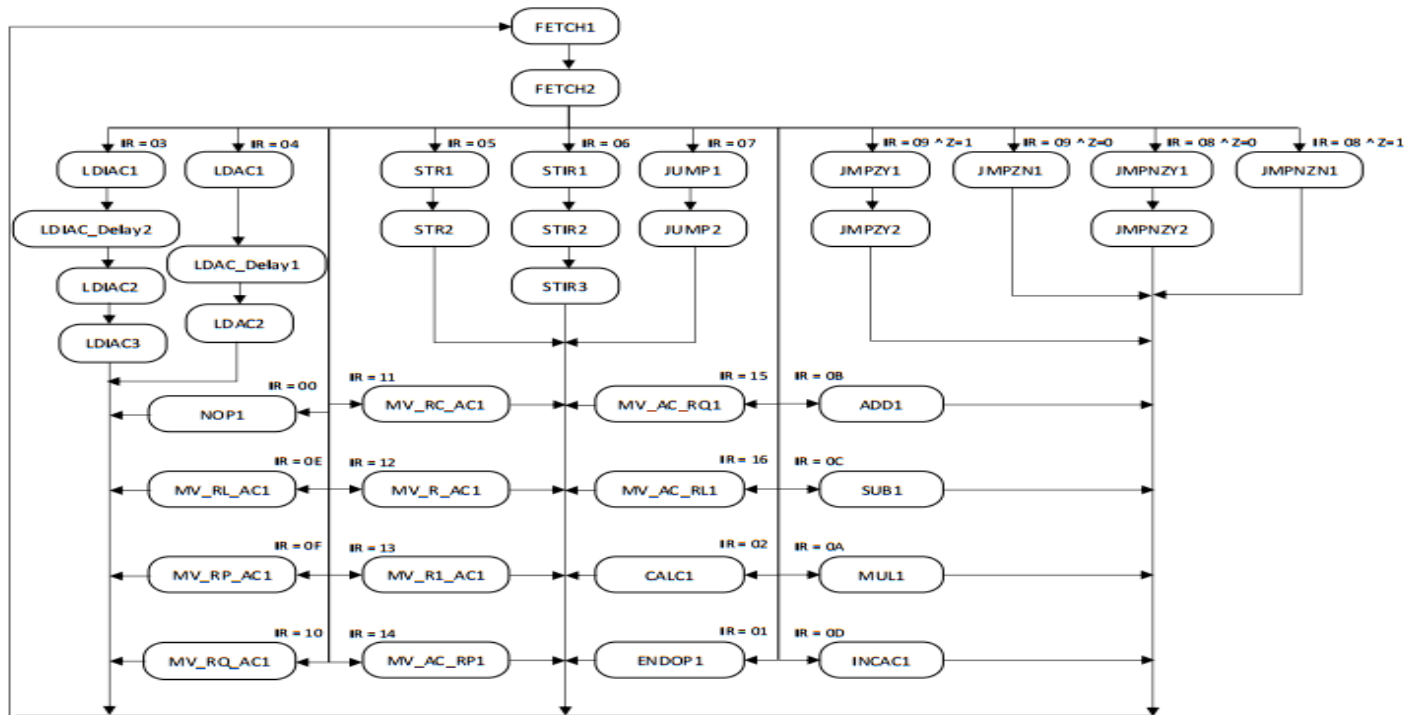
3. Micro-Instruction

| INSTRUCTION | STATE | MICROINSTRUCTION |
|-----------------|--|---|
| FETCH | FETCH_DELAY1 FETCH1 FETCH2 | IR \leftarrow Instruction memory, Instruction read IR \leftarrow Instruction memory, Instruction read PC \leftarrow PC+1 |
| NOP | NOP1 | No operation |
| LDAC | LDAC1 LDAC_DELAY1 LDAC2 | AR \leftarrow AC, Data read R \leftarrow Data Memory R \leftarrow Data Memory |
| LDIAC | LDIAC_DELAY1 LDIAC1 LDIAC2 LDIAC_DELAY2 LDIAC3 | IR \leftarrow Instruction Memory IR \leftarrow Instruction Memory AR \leftarrow IR, PC \leftarrow PC+1, Data read AC \leftarrow Data Memory AC \leftarrow Data Memory |
| STR | STR1 STR_DELAY1 STR2 | AR \leftarrow AC, Data write Data Memory \leftarrow R Data Memory \leftarrow R |
| STIR | STIR_DELAY1 STIR1 STIR2 STIR_DELAY2 STIR3 | IR \leftarrow Instruction Memory IR \leftarrow Instruction Memory AR \leftarrow IR, PC \leftarrow PC+1, Data write Data Memory \leftarrow R Data Memory \leftarrow R |
| MV RL AC | MV_RL_AC1 | RL \leftarrow AC |
| MV R AC | MV_R_AC1 | R \leftarrow AC |
| MV RP AC | MV_RP_AC1 | RP \leftarrow AC |
| MV RQ AC | MV_RQ_AC1 | RQ \leftarrow AC |
| MV RC AC | MV_RC_AC1 | RC \leftarrow AC |
| MV R1 AC | MV_R1_AC1 | R1 \leftarrow AC |
| MV AC RP | MV_AC_RP1 | AC \leftarrow RP |
| MV AC RQ | MV_AC_RQ1 | AC \leftarrow RQ |
| MV AC RL | MV_AC_RL1 | AC \leftarrow RL |

| | | |
|--------------|--|--|
| JUMP | JUMP_DELAY1 JUMP1 JUMP2 | IR \leftarrow Instruction Memory IR \leftarrow Instruction Memory PC \leftarrow IR |
| JMPZ | JMPZY_DELAY1 JMPZY1 JMPZY2 JMPZN1 | IR \leftarrow Instruction Memory IR \leftarrow Instruction Memory PC \leftarrow IR PC \leftarrow PC+1 |
| JMPNZ | JMPNZY_DILAY1 JMPNZY1 JMPNZY2 JMPNZN1 | IR \leftarrow Instruction Memory IR \leftarrow Instruction Memory PC \leftarrow IR PC \leftarrow PC+1 |
| CLAC | CLAC1 | AC \leftarrow 0 |
| ADD | ADD1 | AC \leftarrow AC+R |
| SUB | SUB1 | AC \leftarrow AC-RC |
| MUL | MUL1 | AC \leftarrow AC*R1, RP \leftarrow RP+1, RQ \leftarrow RQ+1, RC \leftarrow RC+1 |
| INCAC | INCAC1 | AC \leftarrow AC+1 |
| ENDOP | ENDOP1 | End Operation |

Bảng 3 - Micro-Instruction

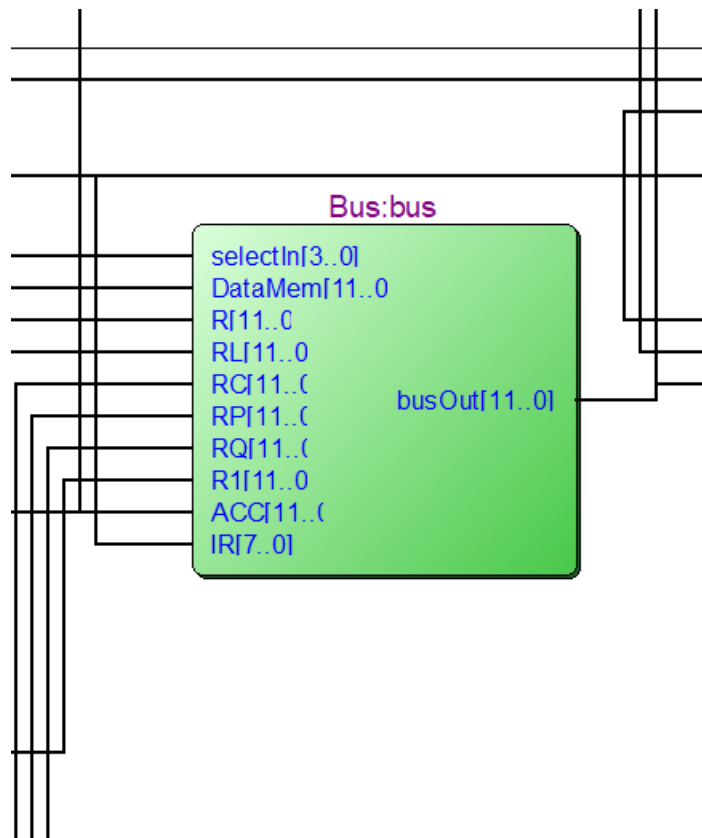
4. State Diagram



Hình 3 - State Diagram

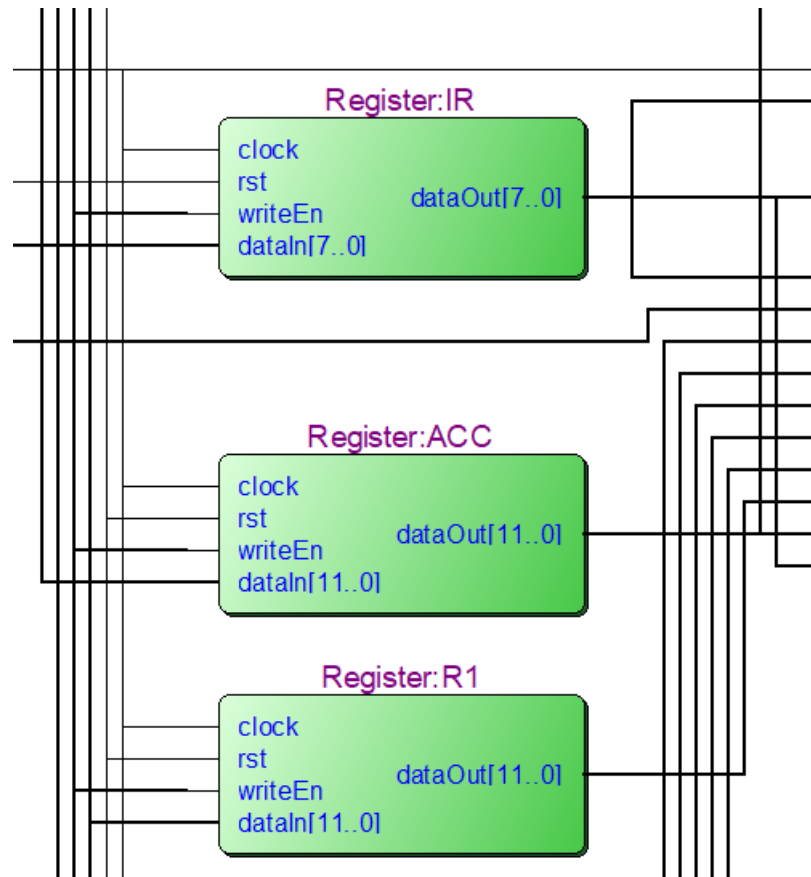
VIII. Modules – RTL View

1. Module Bus



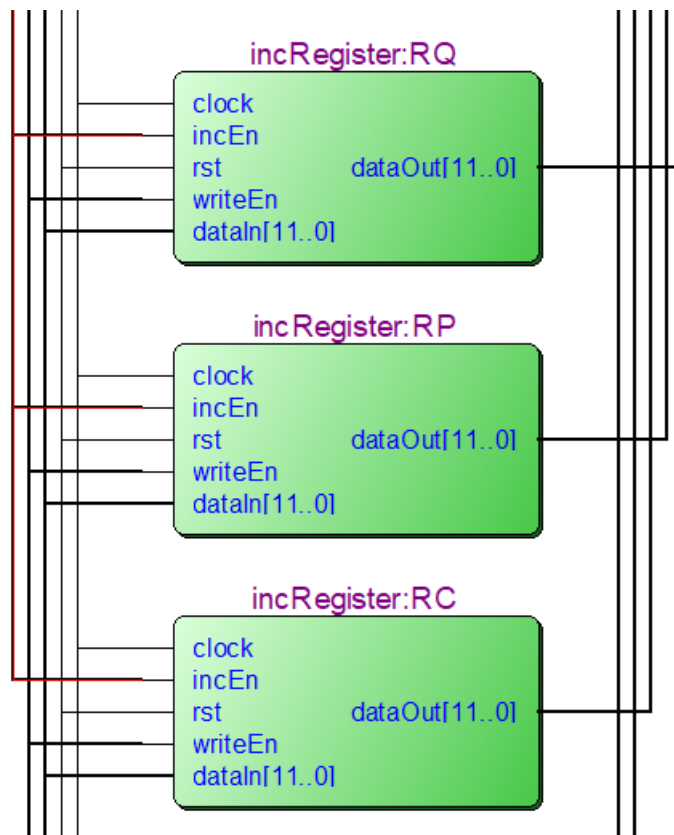
Hình 4 - Module Bus

2. Module Register



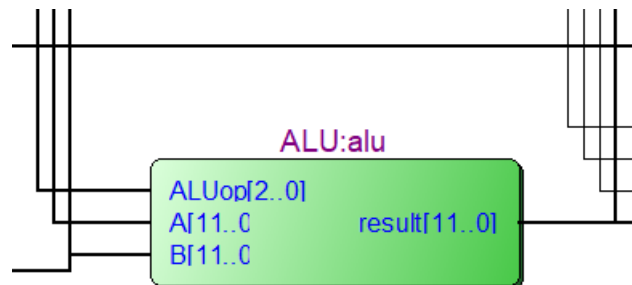
Hình 5 - Module Register

3. Module incRegister



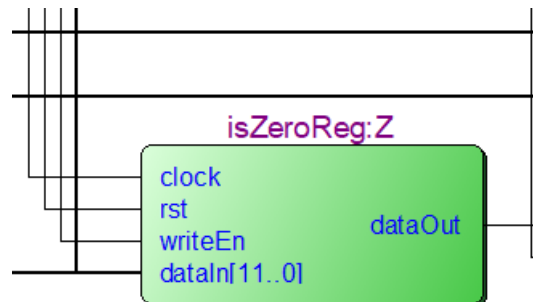
Hình 6 - Module Increment Register

4. Module ALU



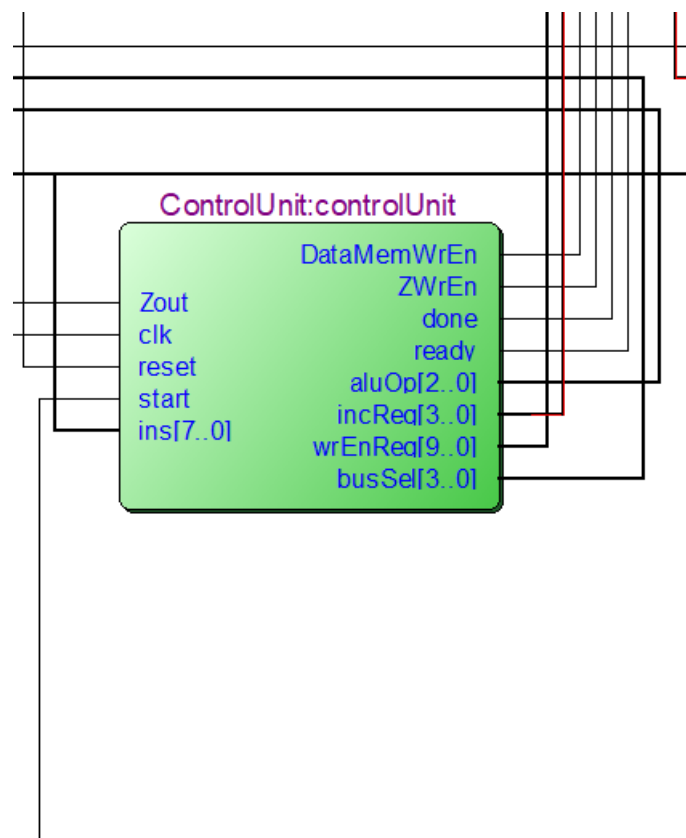
Hình 7 - Module ALU

5. Module isZeroReg



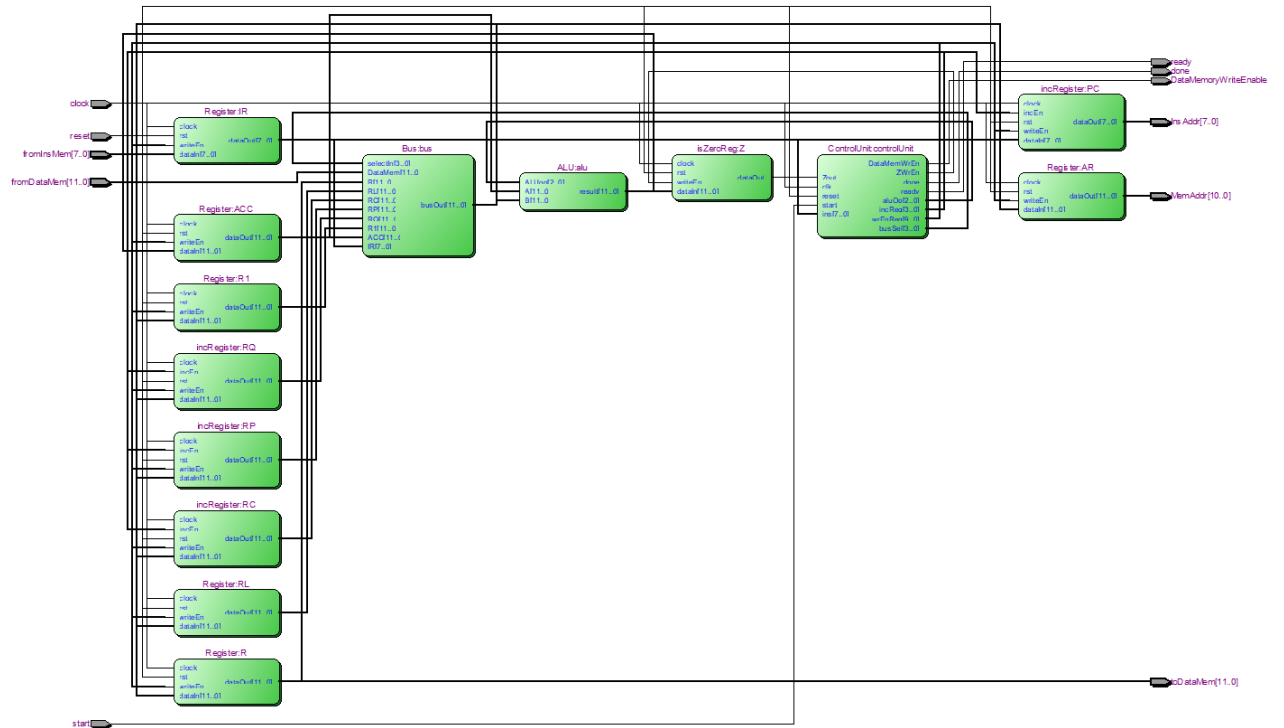
Hình 8 - Module isZeroRegister

6. Module ControlUnit



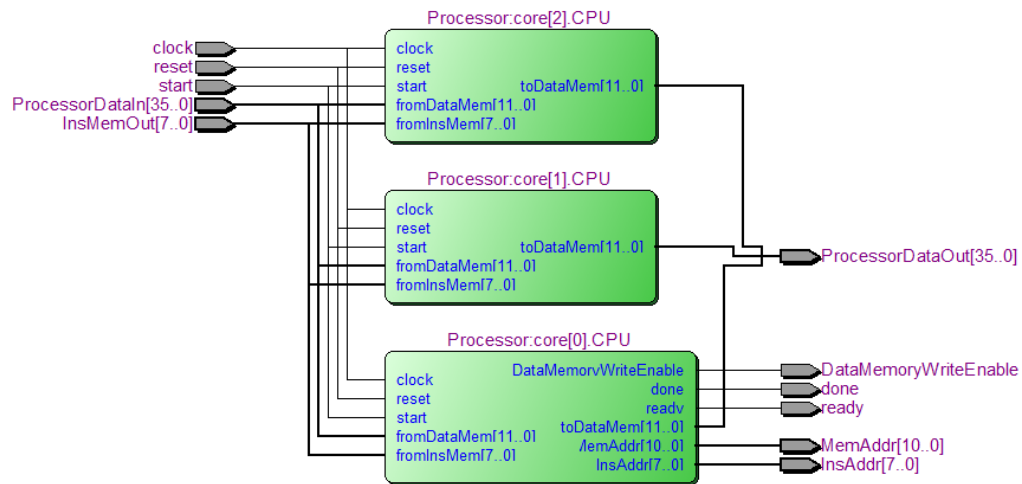
Hình 9 - Module Control Unit

7. Module Processor



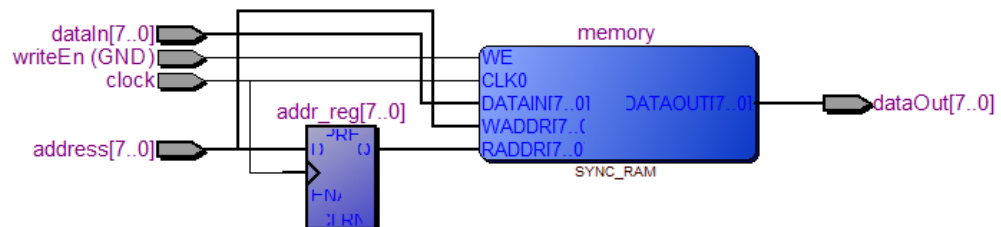
Hình 10 - Module Processor

8. Module MultiCore



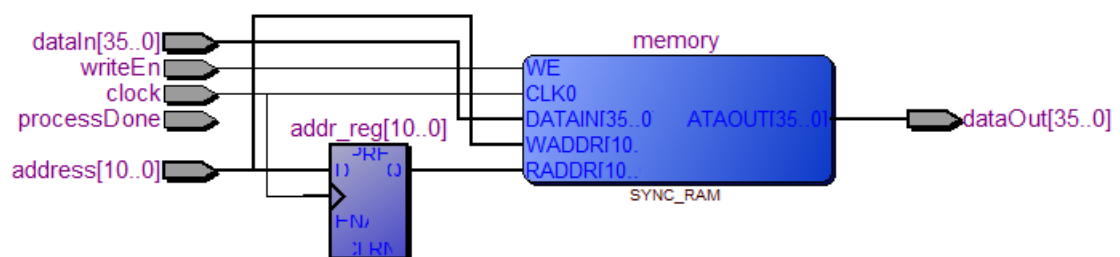
Hình 11 - Module Multi Core CPU

9. Module InsMemory



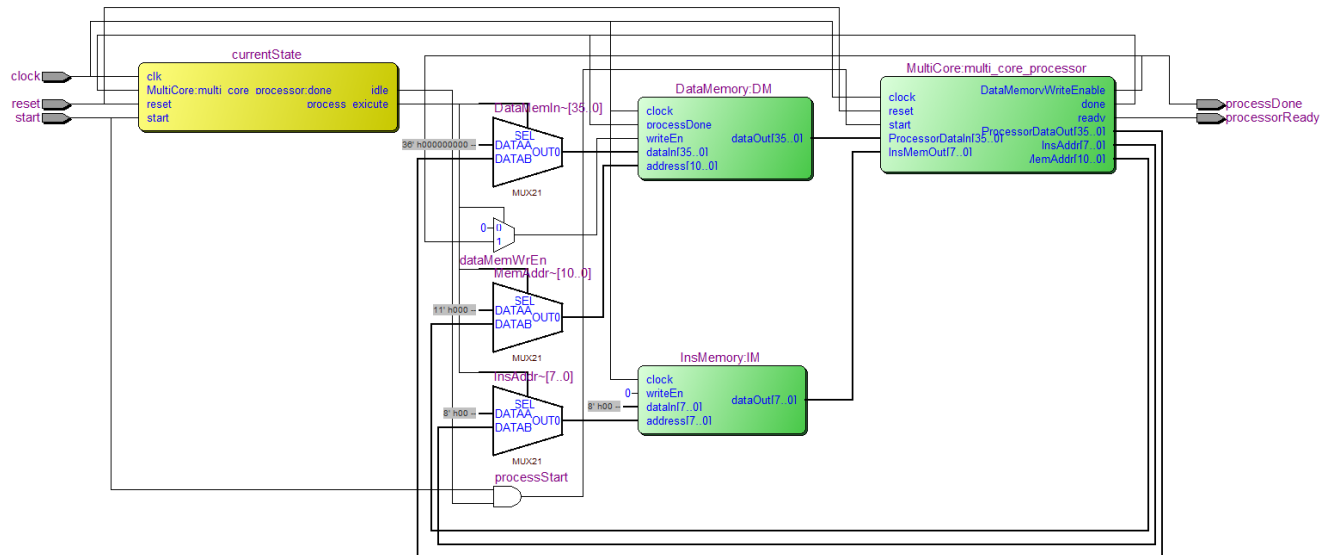
Hình 12 - Module Instruction Memory

10. Module DataMemory



Hình 13 - Module Data Memory

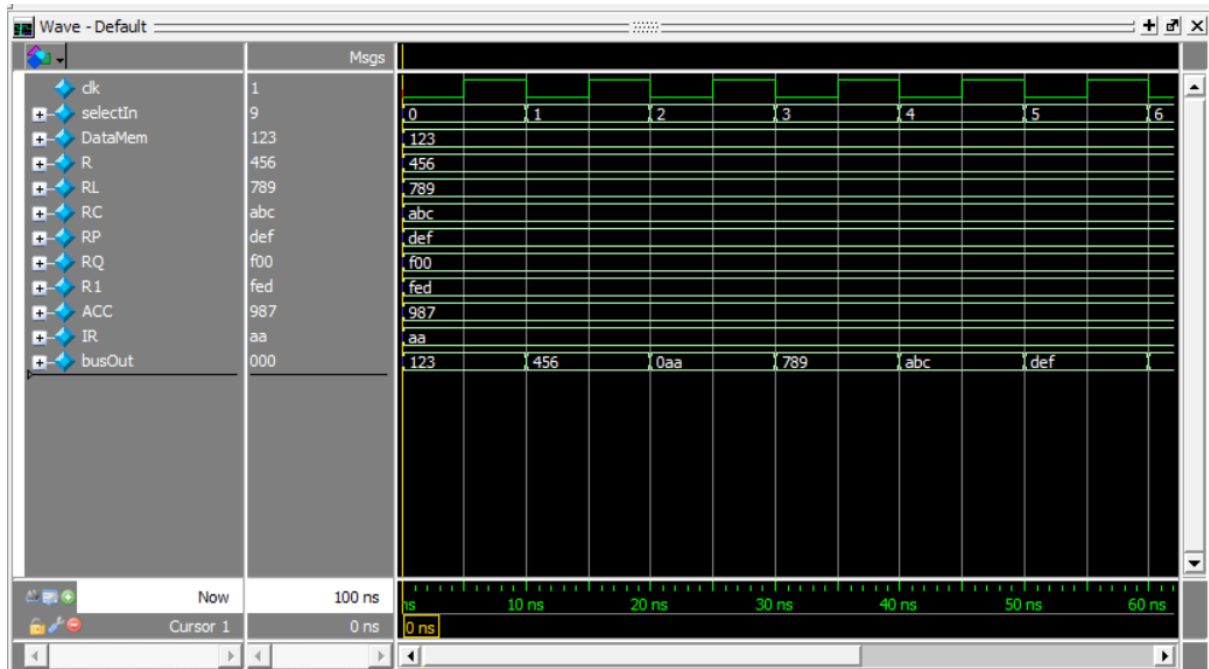
11. Module Top



Hình 14 - Module Top

IX. Test Bench

1. Module Bus_tb

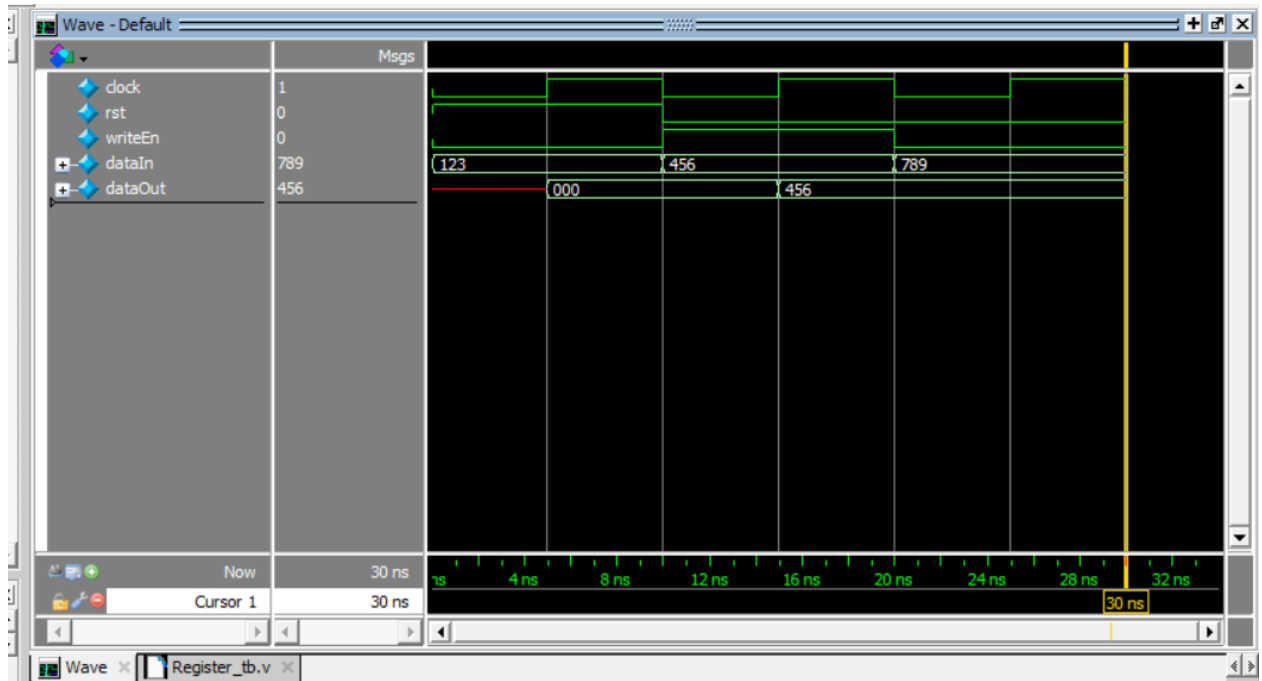


Hình 15 - Module Bus Test bench

| | | | |
|--------------|--------------------------|-----------------|--------------|
| # 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

2. Module Register_tb

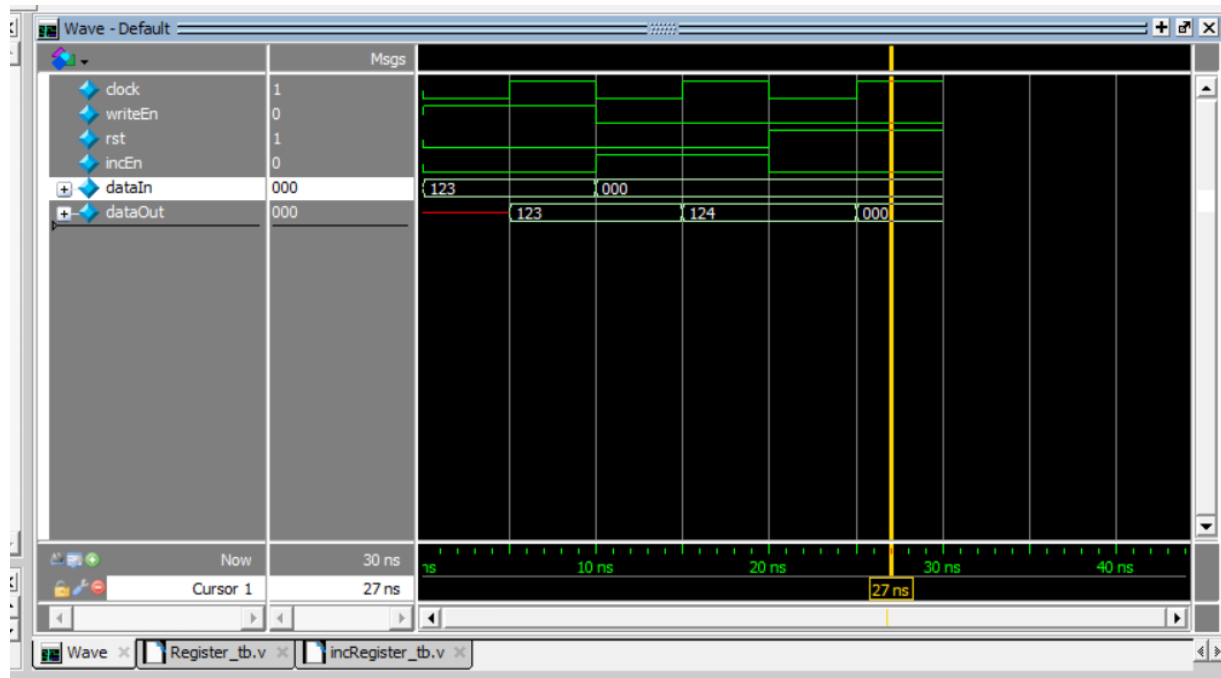


Hình 16 – Module Register Test bench

| | | | | | |
|-------------|-----------------------|--------------|-------------|---------|---------------|
| # 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

3. Module incRegister_tb

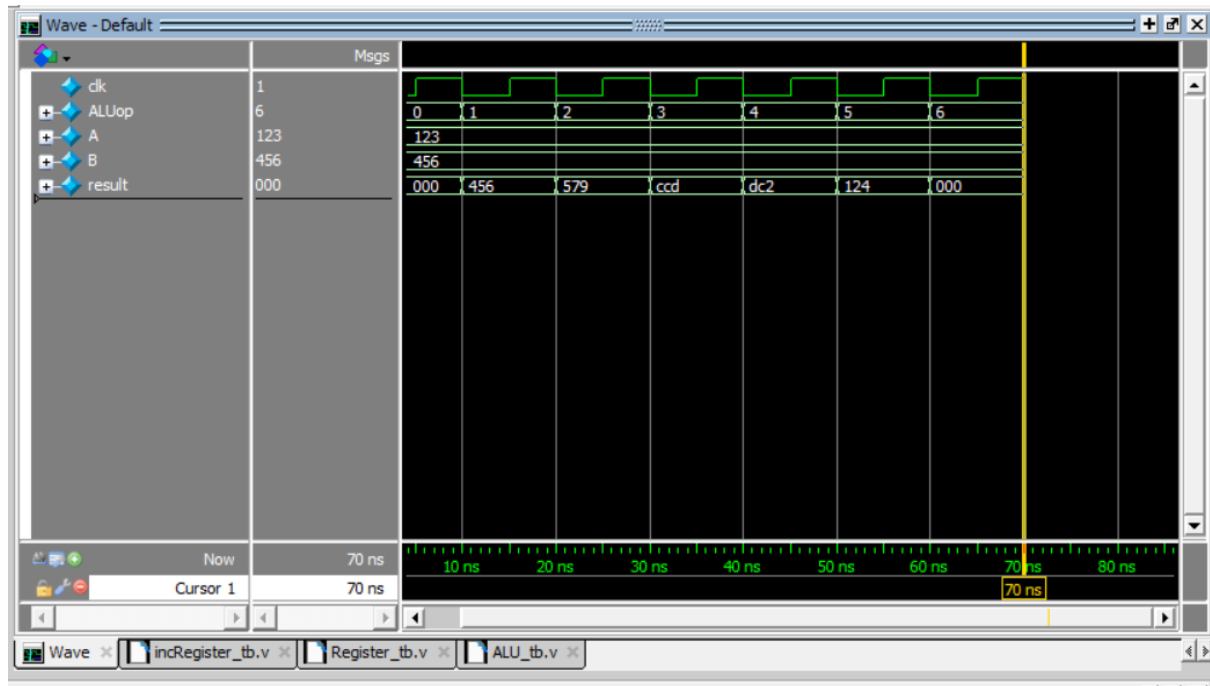


Hình 17 – Module Increment Register Test bench

| | | | | | | |
|-------------|--|--------------|----------------|---------|-----------|---------------|
| # 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

4. Module ALU_tb

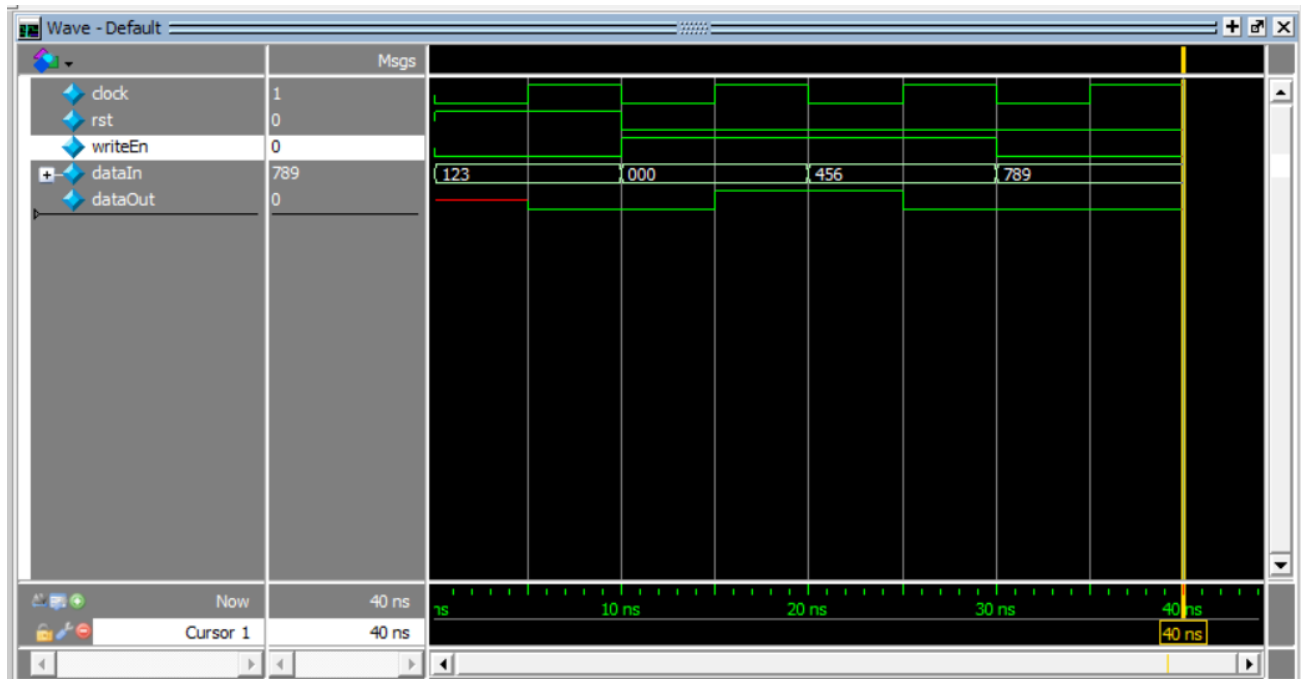


Hình 18 - Module ALU Test bench

| | | | | | |
|-------------|-------------------|---------|---------|-------------|--------------|
| # 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

5. Module isZeroReg_tb

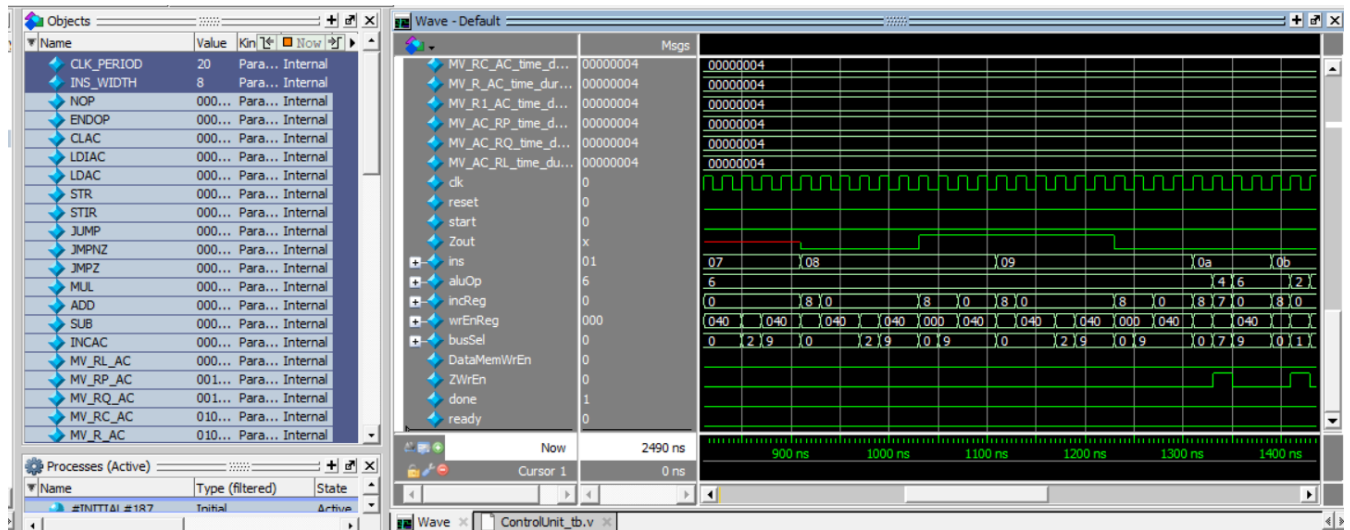


Hình 19 – Module isZero Register Test bench

| | | | | | |
|-------------|-----------------------------|--------------|-------------|---------|-------------|
| # 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 |

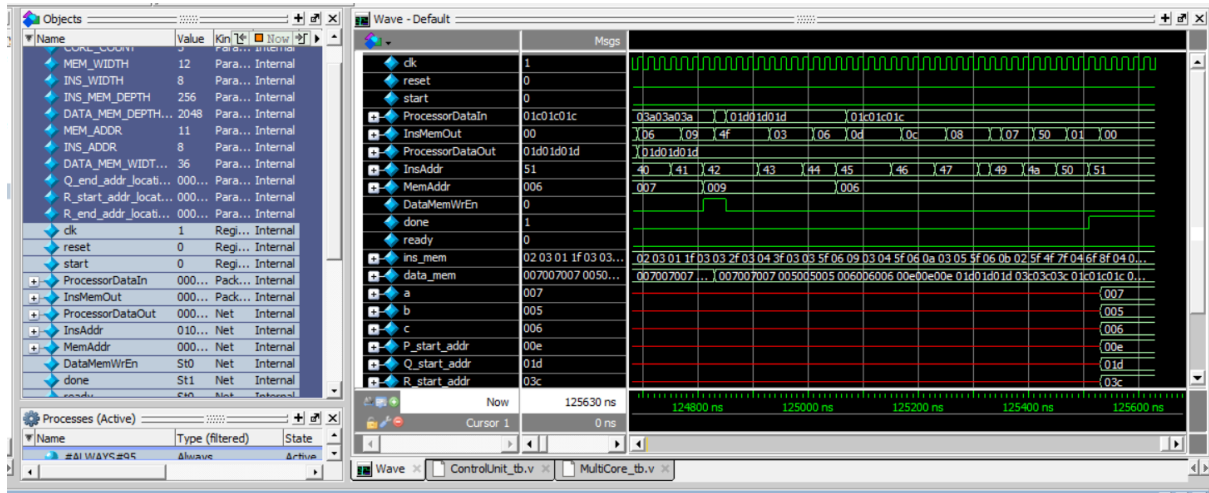
Bảng 8- Bảng giá trị isZero Register Test bench

6. Module ControlUnit_tb



Hình 20 - Module Control Unit test bench

7. Module MultiCore_tb



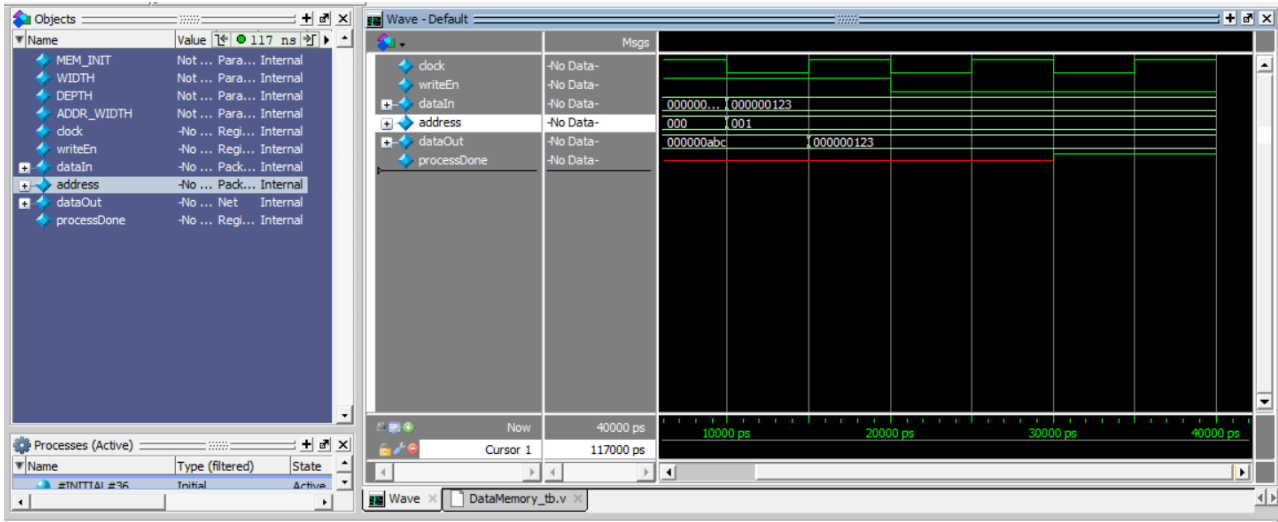
Hình 21- Test bench Multicore CPU

| |
|---------------------------|
| # 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 9 - Bảng mã trộn kết quả của test bench CPU

8. Module DataMemory_tb

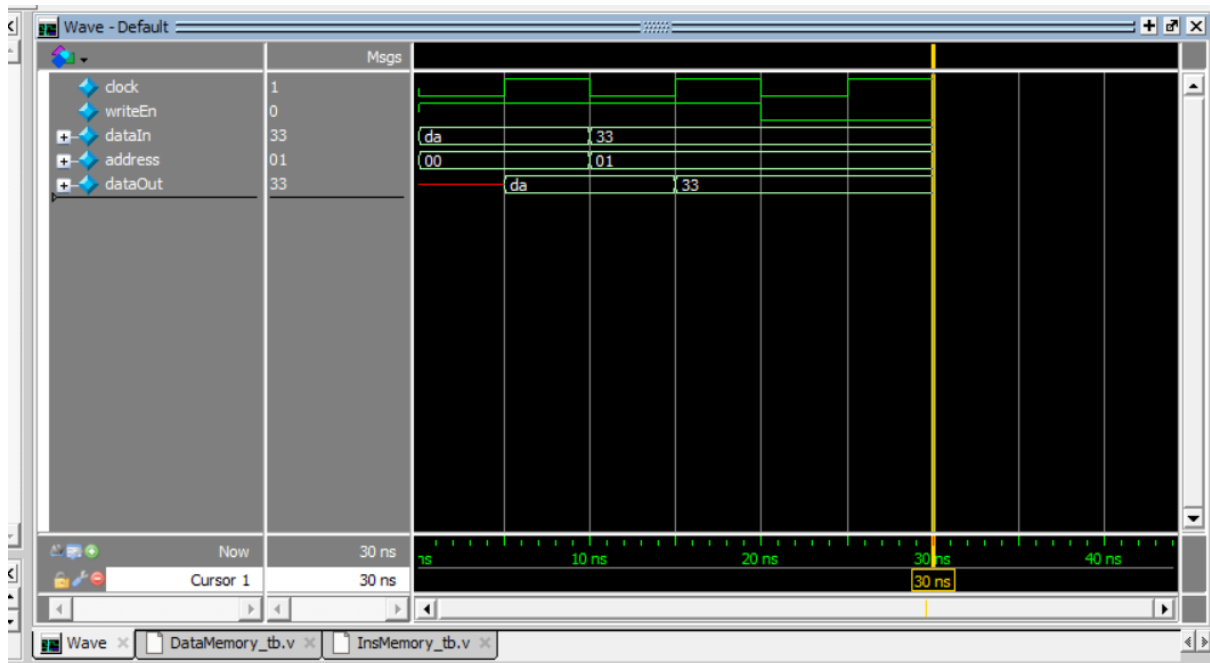


Hình 22 - Test bench Data Memory

| | | | | | |
|-----------|-----------|-----------|------------|-------------|---------------|
| # 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 10 - Bảng giá trị test bench Data Memory

9. Module InsMemory_tb

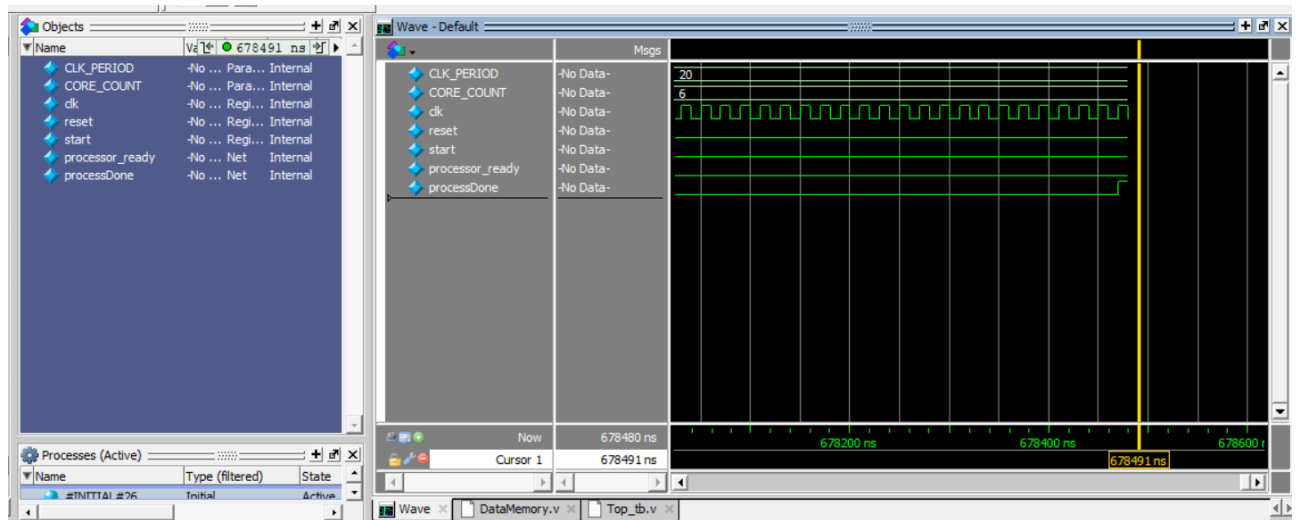


Hình 23 - Test bench Instruction Memory

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

Bảng 11 - Bảng giá trị test bench Instruction Memory

10. Module Top_tb



Hình 24 - Test bench Top module

--- Hết ---