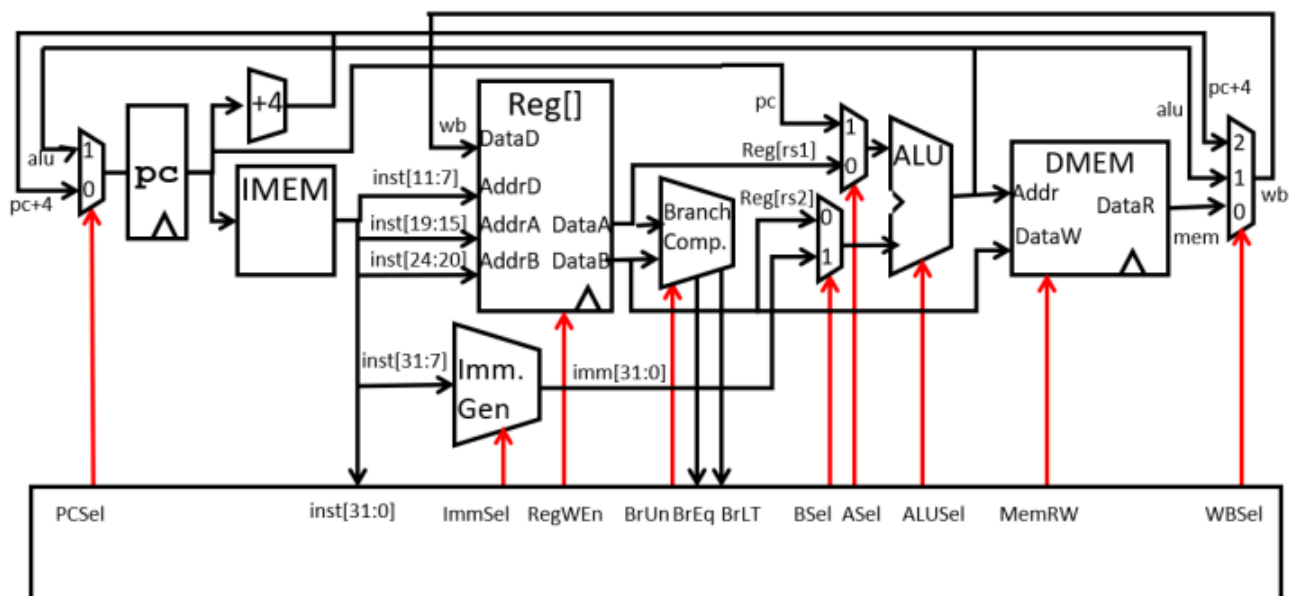


Chapter 1

Thiết kế CPU RICSV 32 đơn chu kỳ

1.1 Thiết kế phần cứng

Phần cứng được thiết kế theo sơ đồ sau:



Trong đó, các tín hiệu kết nối được đặt tên theo bảng sau:

Name	Khối bắt đầu	Khối đích	Ý nghĩa
clk	Input của CPU	PC, Register bank, DMEM	Xung clock điều khiển chu kỳ lệnh
pc_in	PCmux	PC	PC của lệnh tiếp theo
pc_out	PC	IMEM, PC+4, ALUmux1	PC hiện tại
pc_plus4_out	PC+4	PCmux	PC<-PC+4
rs1	IMEM	Register Bank	Địa chỉ của rs1
rs2	IMEM	Register Bank	Địa chỉ của rs2
rd	IMEM	Register Bank	Địa chỉ của rd
rs1_out	Register Bank	BranchComp, ALUmux1	Data của rs1
rs2_out	Register Bank	BranchComp, ALUmux2, DMEM	Data của rs2
imm_in	IMEM	ImmGen	Data vào ImmGen
imm_out	ImmGen	ALUmux2	Data sau khi qua khối ImmGen
alumux1_out	ALUmux1	ALU	Toán hạng 1 vào ALU
alumux2_out	ALUmux2	ALU	Toán hạng 2 vào ALU
aluout	ALU	DMEM, Wbmux, PCmux	Ngõ ra của ALU
dmem_out	DMEM	Wbmux	Data đọc của DMEM
wb_out	Wbmux	Register Bank	Data ghi ngược

Thiết kế khối Control

Tiến hành lập bảng bao gồm các lệnh, các tín hiệu vào và tín hiệu ra, sau đó phân tích từng lệnh về điền vào bảng như hình sau:

Index	No	Type	MNEMONIC	Inst[30]	Inst[14:12]	Inst[6:2]	BrEq	BrLT	PcSel	ImmSel	RegWEN	BrUn	Bsel	Asel	ALUsel	MemRW	DataIn	DataOutAdd	WBsel
							0 => Not EQ 1 => EQ	0 => Not LT 1 => LT	0 => PC + 4 1 => ALU	000 => Ins[31:20], ext sign, 12 001 => Ins[31:20], ext usign, 12 010 => Ins[24:20], ext usign, 5 011 => Ins[31:25][11:7], ext sign, 12 100 => Ins[31:7][30:25][11:8]0, ext sign, 12 101 => Ins[31:12]0_0 110 => Ins[31][19:12][20][30:21]0, ext sign, 20	0 => Read 0 => Signed 1 => Write 1 => Unsigned	0 => Data B 1 => Imm Gen	0 => Data 1 => PC	0000 => ADD 0001 => SUB 0010 => SLL 0011 => SLT 0100 => SLTU 0101 => XOR 0110 => SRL 0111 => SRA 1000 => OR 1001 => AND 1110 => PC + ImmGen + 4 1111 => Sel B	0 => Read Only 1 => Read Write	00 => SB 01 => SHW 11 => SW	000 => exB 001 => exHW 010 => W 011 => exBU 100 => exHU	00 => DMEM 01 => ALU 10 => PC + 4	
0	0 R	ADD	0	000	01100	x	x	0	x		1	x	0	0	0000	0	x	x	01
1	1 R	SUB	1	000	01100	x	x	0	x		1	x	0	0	0001	0	x	x	01
2	2 R	SLL	0	001	01100	x	x	0	x		1	x	0	0	0010	0	x	x	01
3	3 R	SLT	0	010	01100	x	x	0	x		1	x	0	0	0011	0	x	x	01
4	4 R	SLTU	0	011	01100	x	x	0	x		1	x	0	0	0100	0	x	x	01
5	5 R	XOR	0	100	01100	x	x	0	x		1	x	0	0	0101	0	x	x	01
6	6 R	SRL	0	101	01100	x	x	0	x		1	x	0	0	0110	0	x	x	01
7	7 R	SRA	1	101	01100	x	x	0	x		1	x	0	0	0111	0	x	x	01
8	8 R	OR	0	110	01100	x	x	0	x		1	x	0	0	1000	0	x	x	01
9	9 R	AND	0	111	01100	x	x	0	x		1	x	0	0	1001	0	x	x	01

Sau đó, chuyển bảng vừa lập thành khối Control. Viết theo kiểu ROM

Thiết kế các khối chức năng

Phân tích từng khối: chức năng, ngõ vào, ngõ ra và viết riêng từng module.

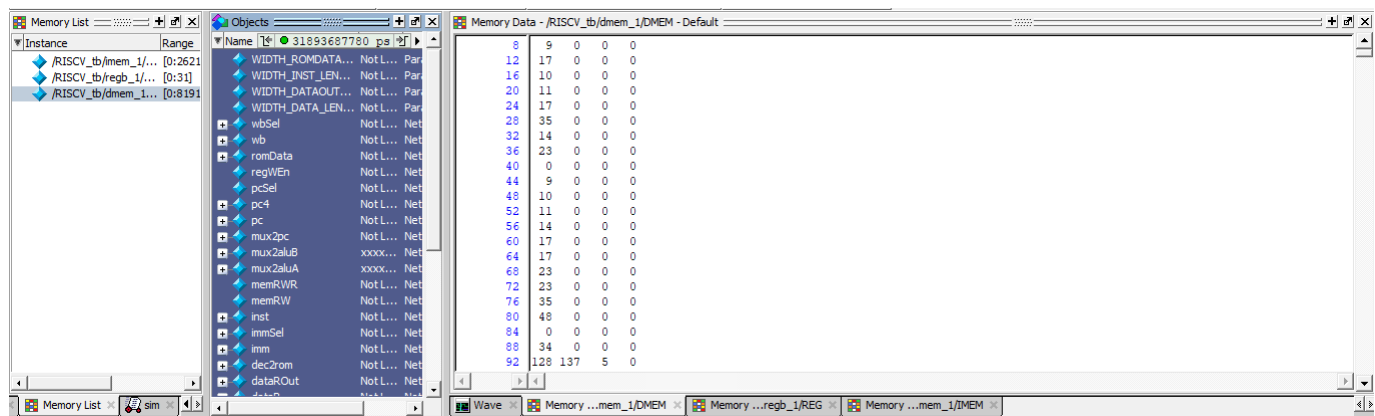
1.2 Thực hiện mô phỏng

1.2.1 Viết đoạn chương trình test

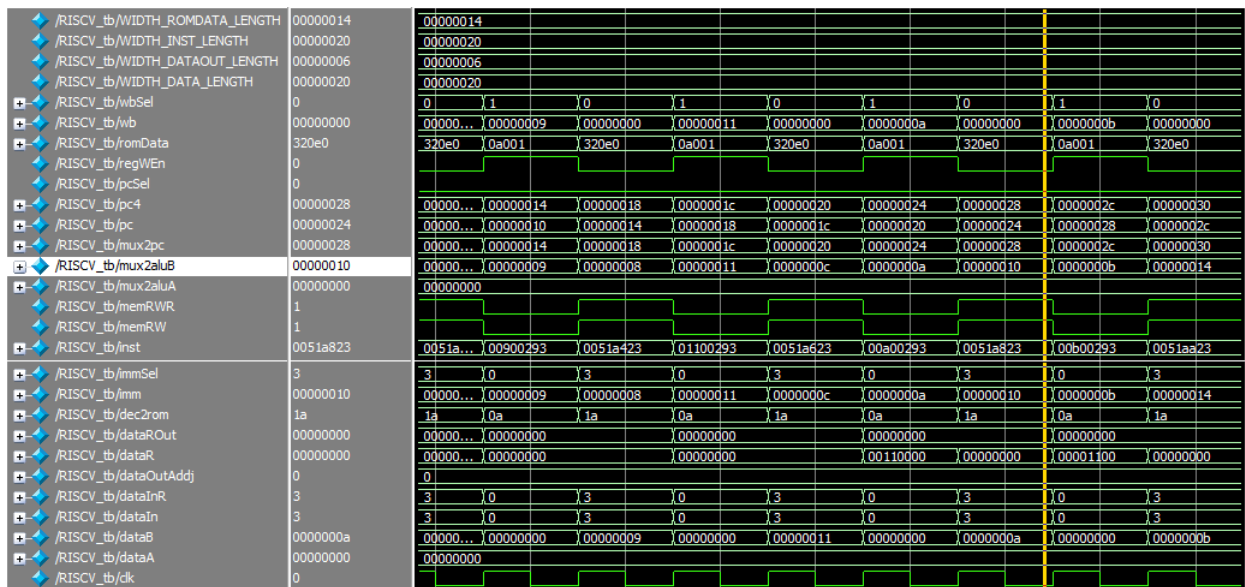
- Lấy 10 số lưu trong DMEM và sắp xếp lại rồi lưu vào DMEM ở 10 vị trí tiếp theo.
- Tính giai thừa số lớn nhất và lưu ở vị trí tiếp theo.
- Tính số Fibonacci của số lớn nhất và lưu ở vị trí tiếp theo.

1.2.2 Test dạng sóng trên ModelSim và kết quả mô phỏng

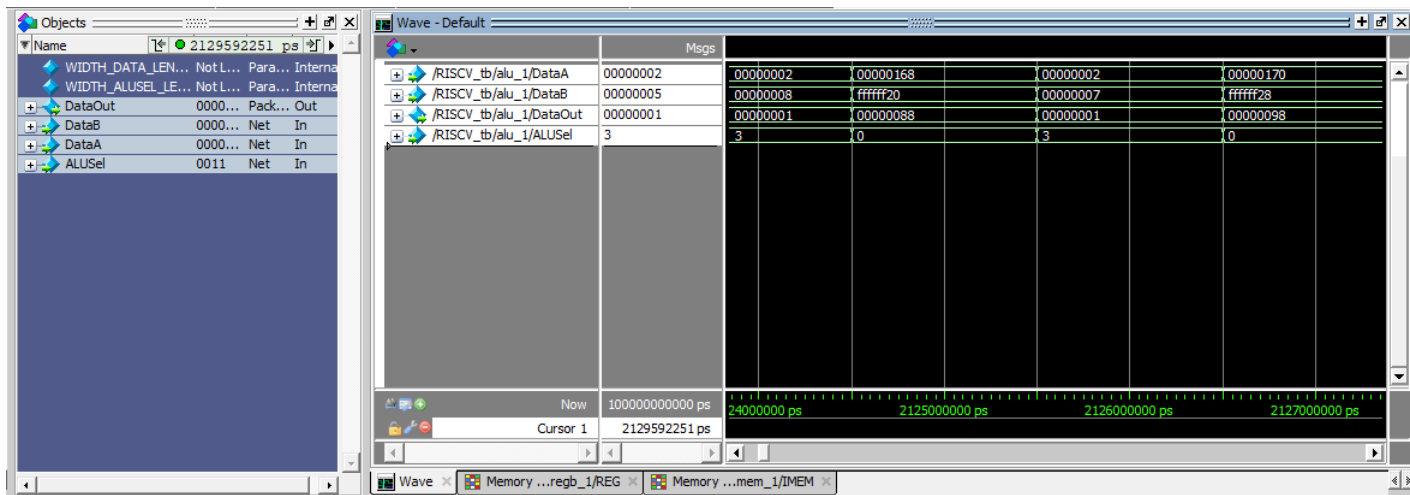
Kết quả sau khi chạy đoạn code trên:



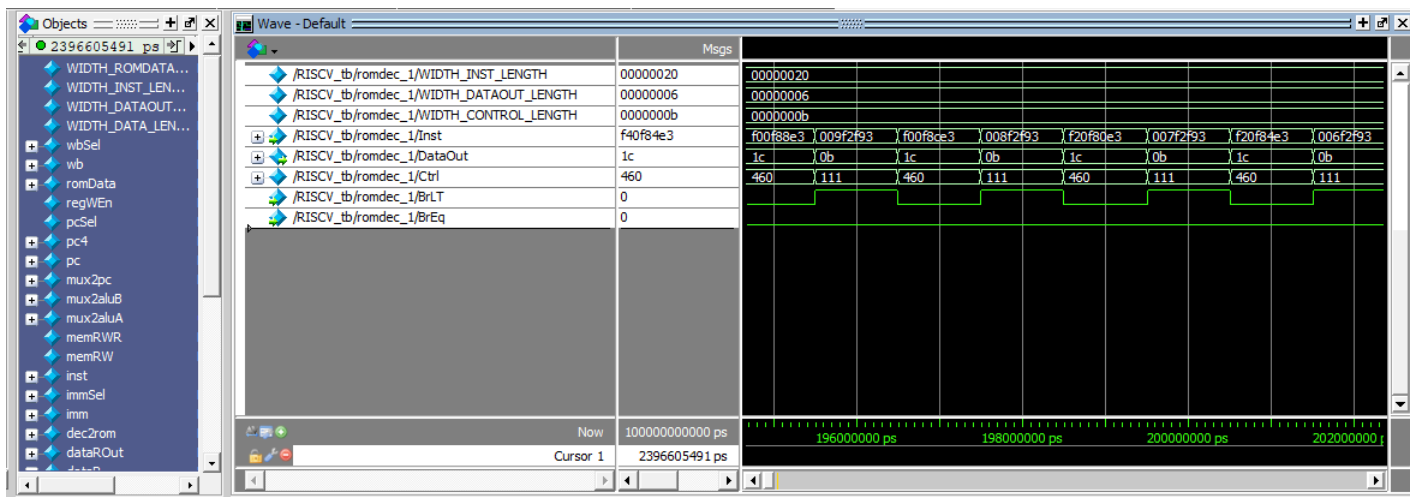
Dạng sóng của đoạn code trên:



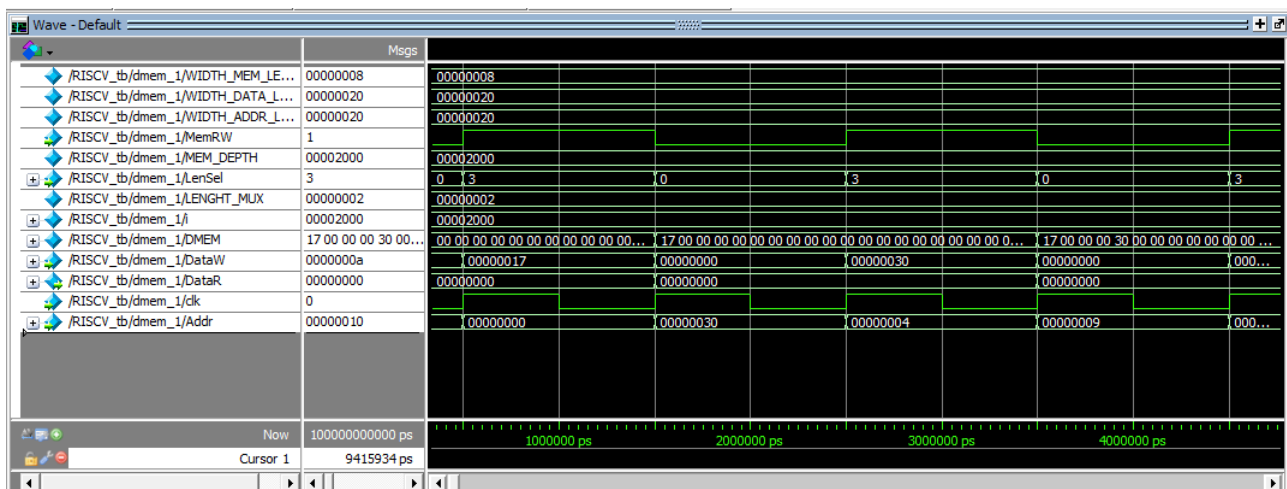
Kết quả mô phỏng của khối ALU:



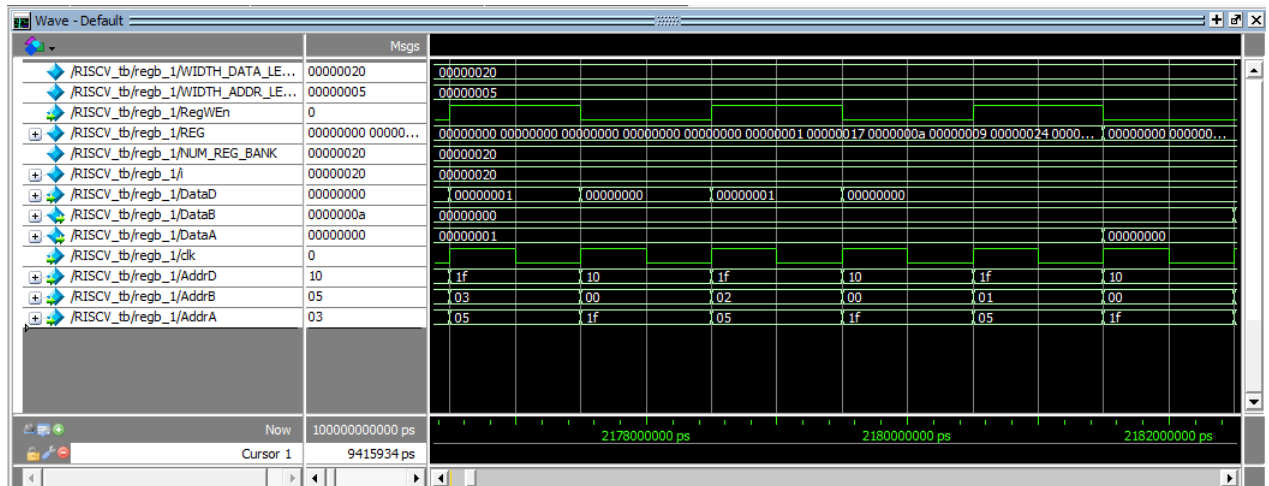
Kết quả mô phỏng khối Branch Comp:



Kết quả mô phỏng khối DMEM:



Kết quả mô phỏng khối Reg:



Kết quả mô phỏng khối Imm:

