

Chapter 1

Thiết kế CPU RICSV 32 Pipeline 5 tầng

1.1 Tổng quát về Pipeline

Pipeline là một kỹ thuật mà trong đó các lệnh được thực thi theo kiểu chồng lấn lên nhau (overlap).

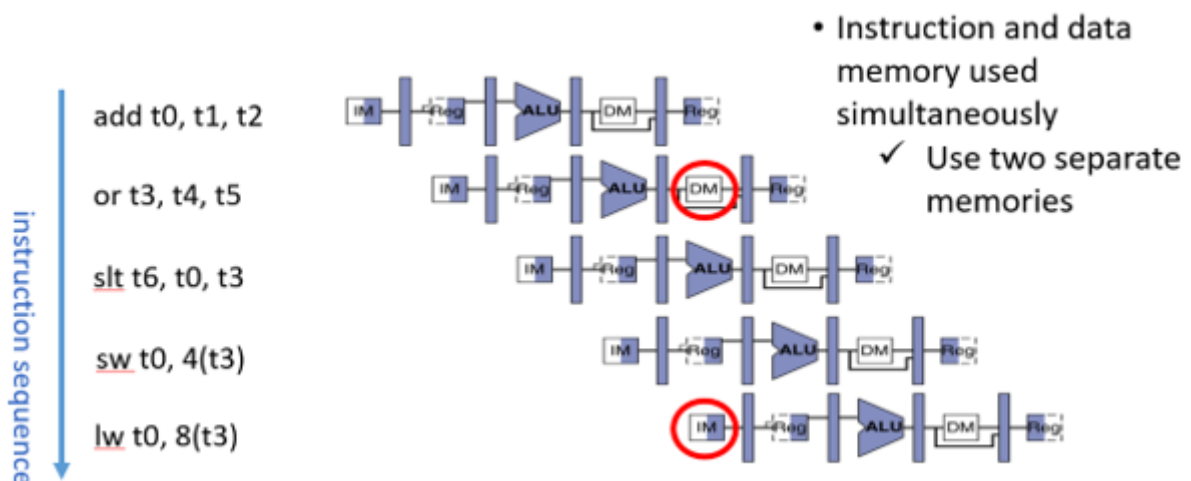
Dựa trên thiết kế của đơn chu kỳ ta thấy rằng, để thực hiện một câu lệnh cần ít nhất 5 trạng thái. Các trạng thái được nêu chi tiết trong bảng sau:

IF	ID	EX	MEM	WB
Instruction Fetch	Instruction decode/register file read	Execute/Address calculation	Memory access	Write back
Nạp lệnh	Decode	Tính toán	Truy cập MEM	Ghi vào RD

1.2 Pipelining Hazards

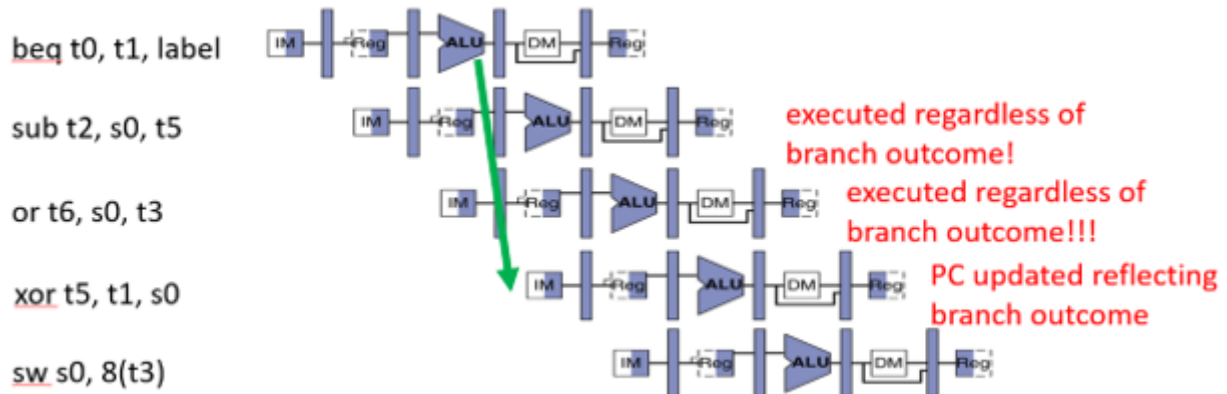
1.2.1 Structural Hazard

Hai câu lệnh cùng truy cập vào bank thanh ghi cùng lúc nhưng nó chỉ có thể đọc hoặc ghi dựa vào Clock.



1.2.2 Control Hazard

Hazard control sẽ xảy ra khi dùng các lệnh nhảy như beq, bne,... Sau khi lệnh nhảy đến tầng ALU nhưng địa chỉ kế tiếp chứa câu lệnh thực hiện kế tiếp không biết là nhảy hay không nhảy và nhảy đến đâu

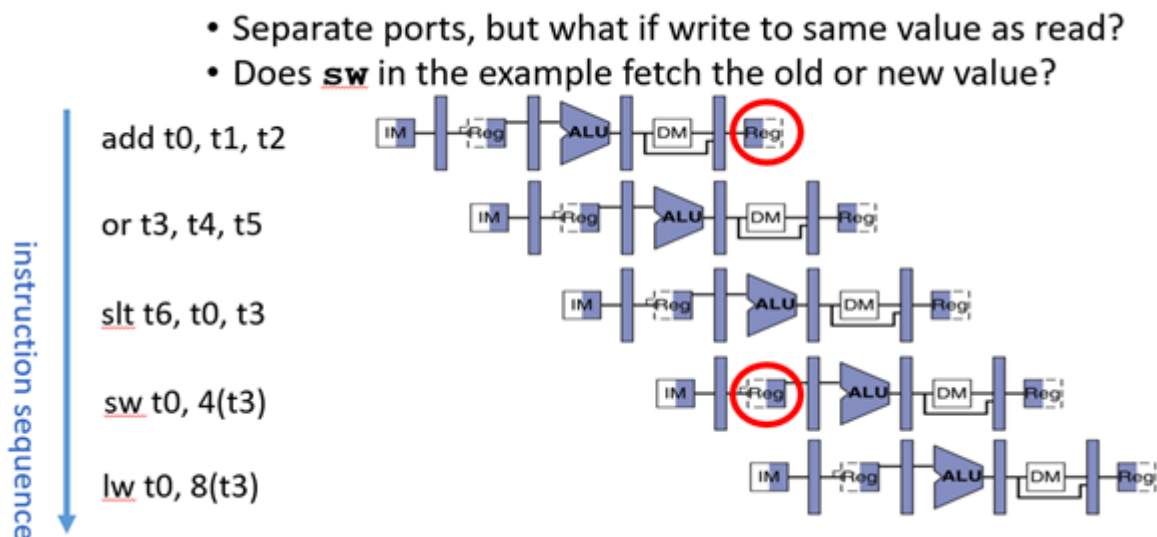


1.2.3 Data hazard

Có hai loại Data Hazard

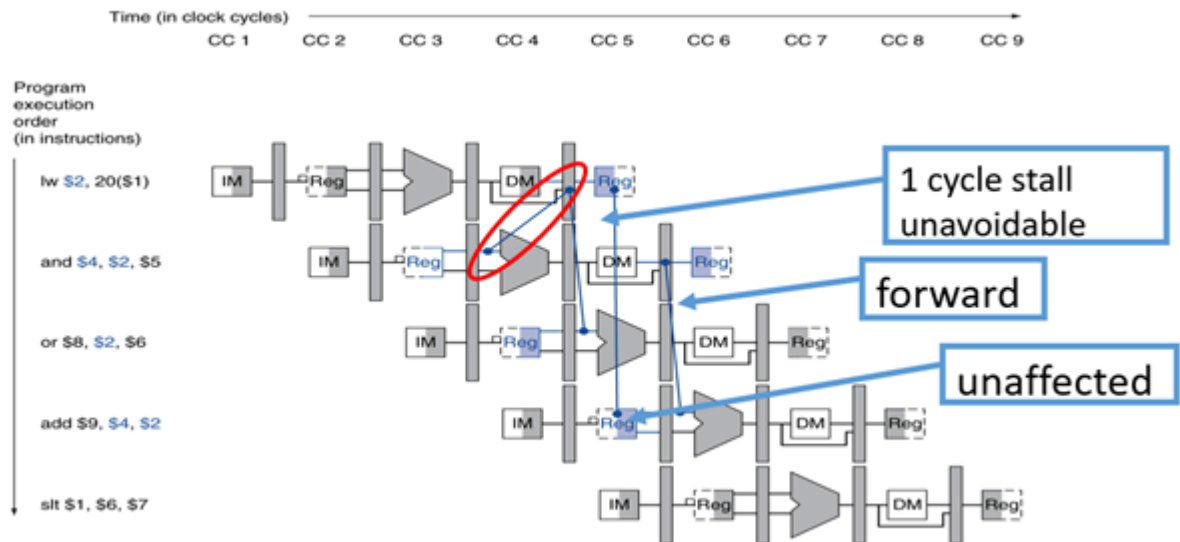
- Hazard R type instruction

Ở đây ta dùng phương pháp Forwarding để lấy giá trị ở các tầng khác đưa về tầng đang cần dùng giá trị đó. Cấu trúc như sau:



Việc Forward này có tác dụng đưa các giá trị cần dùng về tầng X khi ở tầng X cần dùng kết quả được tính trong cùng một thanh ghi nhưng nó vẫn chưa Writeback được về băng thanh ghi, ở đây ta có thể có hai vị trí Writeback về đó là ở tầng M và tầng W. Việc Forward này đều có thể xảy ra được trên cả rs1 và rs2.

- Load data hazard

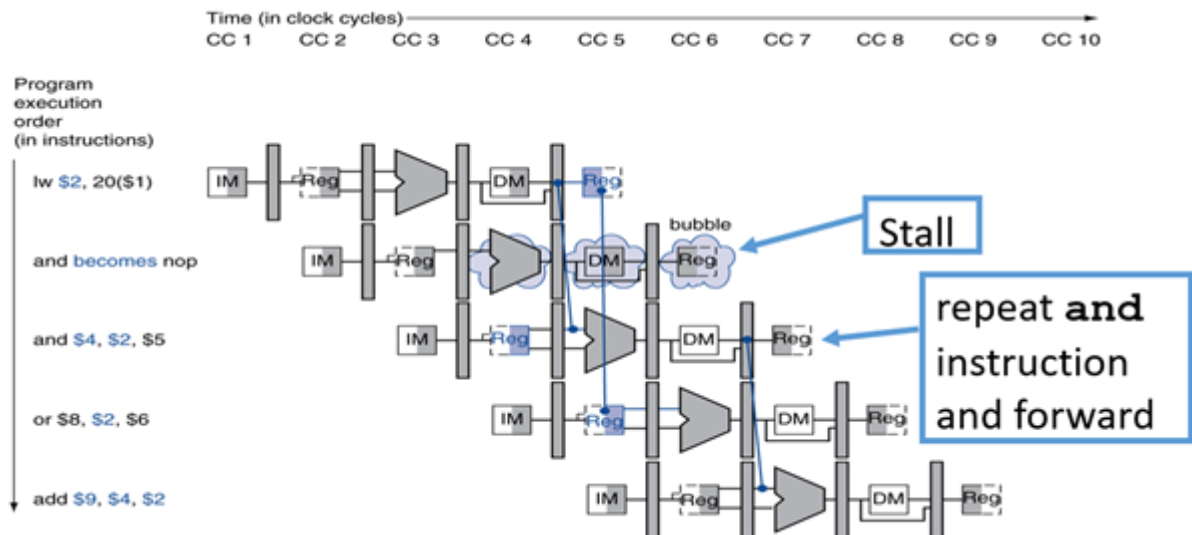


1c

Lecture 13: Pipelining

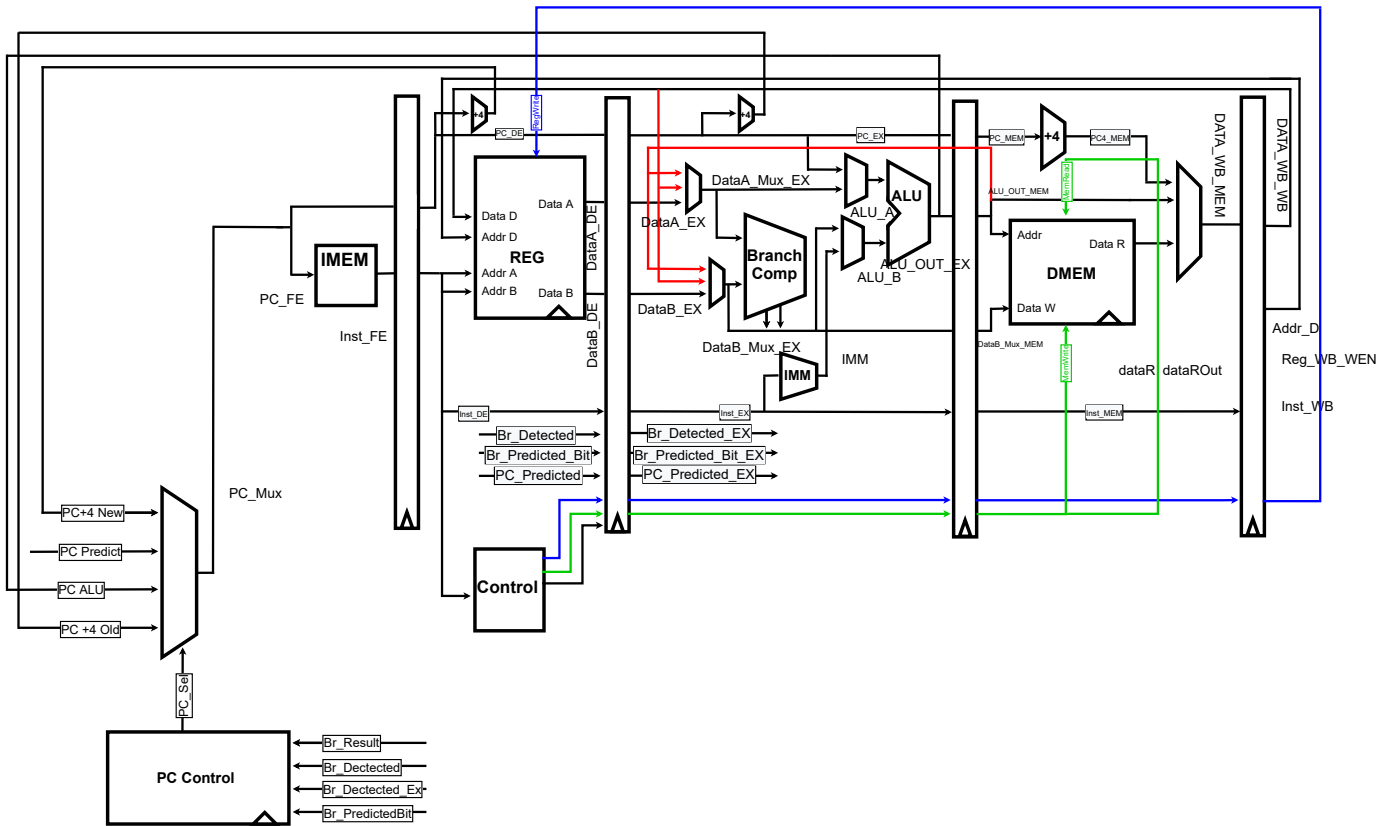
30

Lỗi ở đây là việc khi dùng lệnh load vào một thanh ghi nào đó nếu tại thời điểm đó mà cũng chính là thời điểm bộ ALU cần giá trị của thanh ghi đó để tính toán nên ta phải dừng quá trình này lại 1 chu kỳ hay chèn một lệnh NOP vào giữa hai lệnh đó.



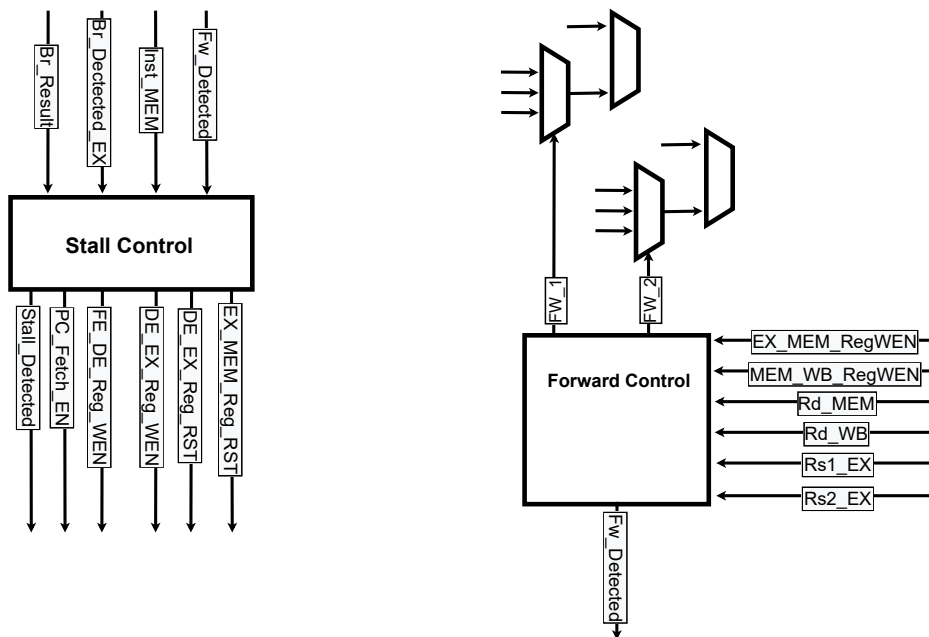
1.3 Thiết kế Pipeline 5 tầng

Kết quả phân cứng được thiết kế lại sau khi Pipeline:

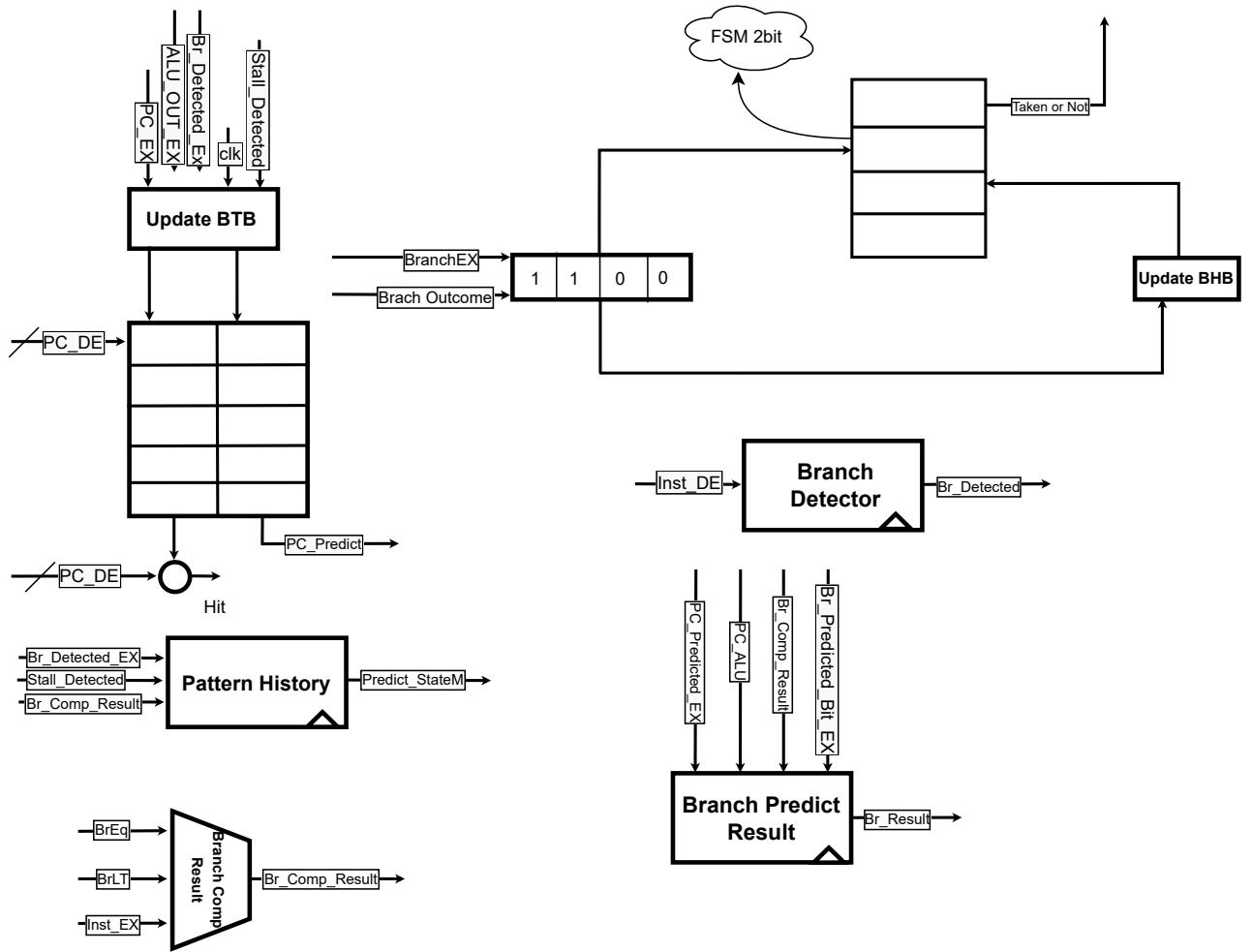


Ngoài ra, một số khối được thiết kế thêm để giải quyết các vấn đề về Hazard:

– Bộ Stall và Forward



– Bộ Branch predict



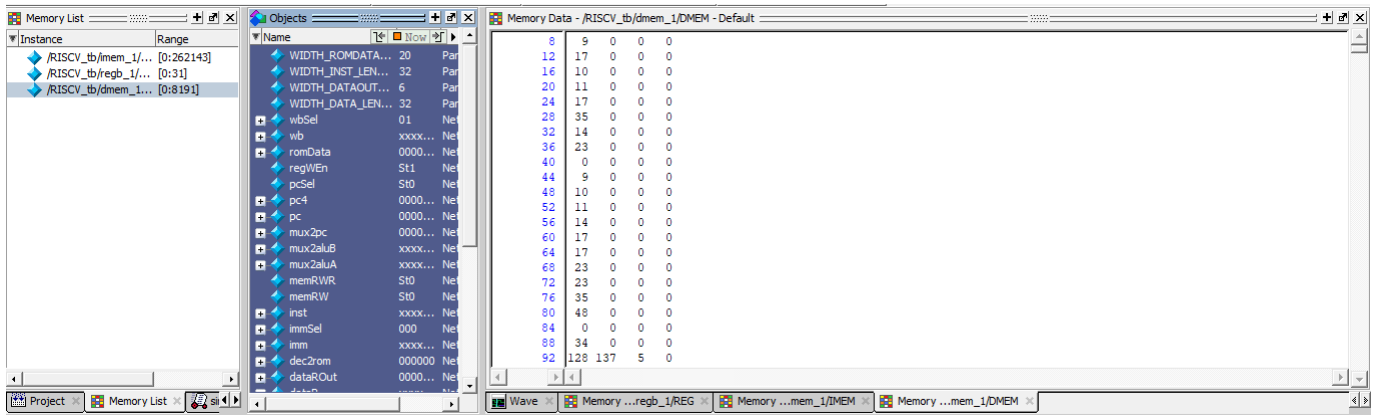
1.4 Thực hiện mô phỏng

1.4.1 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 Fibonanci của số lớn nhất và lưu ở vị trí tiếp theo.

1.4.2 Test dạng sóng trên ModelSim và kết quả mô phỏng

Kết quả sau khi chạy đoạn code trên:



Dạng sóng của đoạn code trên:

