**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**



**BÁO CÁO TIẾN ĐỘ ĐỒ ÁN LẦN 2**

**MÔN THIẾT KẾ HỆ THỐNG SỐ VỚI HDL**

|  |  |  |
| --- | --- | --- |
| **THÀNH VIÊN NHÓM** | **NGUYỄN XUÂN TÙNG**    **NGUYỄN ĐẶNG ANH KIỆT** | **21521649**  **21520312** |
| **LỚP CE213.O11** |  | |

I. Tổng quan đề tài  
- Để thực hiện đề tài thiết kế một bộ xử lý pipeline theo kiến trúc RISC-V 32-bit, ta chia việc thiết kế thành 3 giai đoạn cụ thể như sau:

* Giai đoạn 1 (tiến độ lần 1): Xác định các thành phần phần cứng và vẽ datapath dựa vào tập lệnh và định dạng mã hóa lệnh. Thiết kế controller cho datapath đã có.
* Giai đoạn 2 (tiến độ lần 2): Ứng dụng cơ chế pipeline vào datapath đã có.
* Giai đoạn 3 (báo cáo cuối kì): Xử lý các xung đột và hoàn thiện bộ xử lý (có thể sẽ không làm được).

- Trong bản báo cáo tiến độ lần 1 này ta sẽ trình bày về giai đoạn 2 trong việc thiết kế, ứng dụng cơ chế pipeline vào datapath đã thiết kế ở giai đoạn 1.

# II. Datapath & Controller & Pipeline

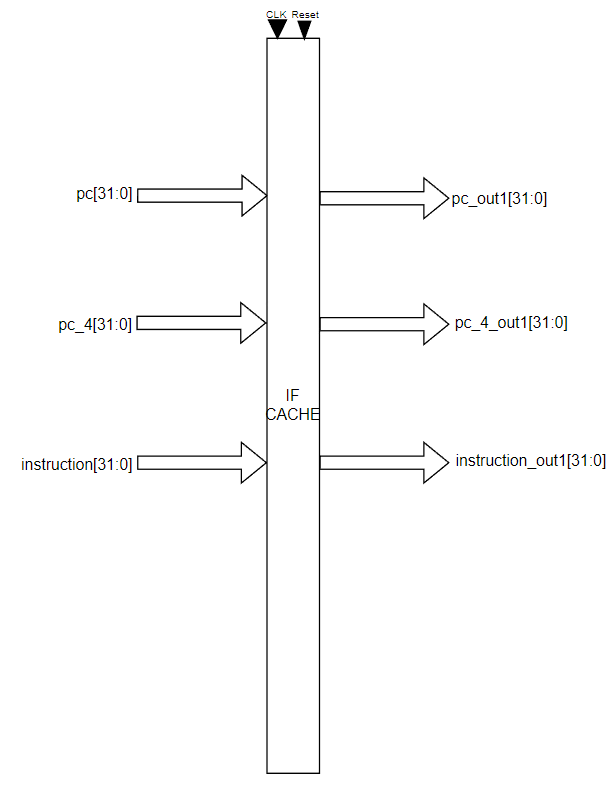
Hình 2: Datapath & Controller & Pipeline.

- Sử dụng cơ chế pipeline cho 5 giai đoạn (Fetch,Decode,Execute,Memory,Writeback).

- Pipline 5 giai đoạn ta sẽ cần 4 pipeline register dùng để kết nối 5 giai đoạn pipeline với nhau :

* IF Cache: Pipeline register thứ nhất, dùng để lưu các dữ liệu pc, pc\_4, instruction.
* ID Cache: Pipeline register thứ hai, dùng để lưu các dữ liệu reg\_dest, branch, w\_dm, r\_dm, alu\_op, alu\_src, pc\_sel, pc, pc\_4, data\_register, immdediate.
* ALU Cache: Pipeline register thứ ba, dùng để lưu các dữ liệu reg\_dest, branch, w\_dm, r\_dm, pc\_4, alu\_result, branch signal từ khối ALU, data2 từ register, pc\_branch
* MEM Cache: Pipeline register cuối cùng, dùng để lưu các dữ liệu: reg\_dest, branch, branch signal từ khối ALU, alu\_result, data từ khối memory, pc\_branch.

# III. Mô tả chung khối pipeline register



Ví dụ về khối pipeline register 1: IF Cache.

Input: clk, reset, pc[31:0], pc\_4[31:0], instruction[31:0].

Output: pc\_out1[31:0], pc\_4\_out1[31:0], instruction[31:0].

Khi có tín hiệu reset = 1 thì pc\_out1 = 0, pc\_4\_out1 = 0 và instruction\_out1 = x.

Khi có tín hiệu cạnh lên của clock thì pipeline register sẽ xuất giá trị lưu trong register và sau đó 1 khoảng thời gian delay thì register sẽ được lưu giá trị từ ngõ vào. Cách này chỉ sử dụng khi mô phỏng trạng thái hoạt động của pipeline.

# IV. Kết quả kiểm thử

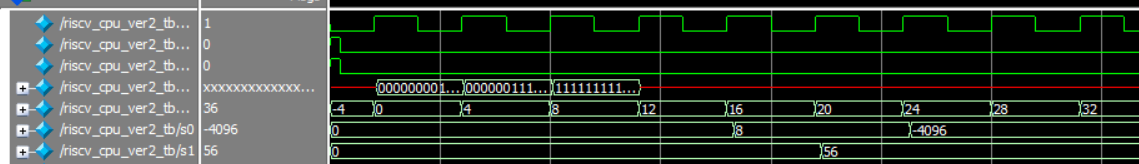
Phần mềm sử dụng: ModelSim.

Instructions được kiểm tra:

addi $0, $1, 8 //$0 = 8

addi $1, $0, 56 //Vì xung đột dữ liệu, $0 lúc này không mang giá trị 8. $1 sẽ bằng 56

lui $0, 0xFFFFF // $0 = -4096



lệnh 3 hoàn thành

lệnh 2 hoàn thành

lệnh 1 hoàn thành

chu kì 1 chu kì 2 chu kì 3 chu kì 4 chu kì 5

instruction

pc

$0

$1

Vì là pipeline 5 giai đoạn nên lệnh đầu tiên sẽ được hoàn thành ở chu kì 5. Các lệnh tiếp theo chỉ cần tốn 1 chu kì kể từ lệnh thứ nhất.