

# CE118-Lab04

## Thiết kế vi xử lý đơn giản

### 1. Lý thuyết

Một vi xử lý đơn giản sẽ bao gồm 2 khối chính đó là Controller và Datapath.

- Controller có nhiệm vụ điều khiển đường đi của dữ liệu (điều khiển việc đọc ghi của Register File, ...) và điều khiển việc thực hiện tính toán (điều khiển Opcode của khối ALU, ...) trong khối Datapath.
- Datapath chứa các khối cần thiết để thực hiện việc tính toán (Register File, ALU, Bộ dịch,...) được điều khiển bởi Controller.

### 2. Thực hành

Sinh viên thực hiện thiết kế một vi xử lý đơn giản dùng để tính toán biểu thức sau:

$$D_3I_3 + D_2I_2 - D_1I_1 + D_0I_0$$

Trong đó:

- $D_x$  là 4 ký số cuối của MSSV (Ví dụ 4 ký số cuối MSSV là 6789 thì  $D_3 = 6$ ,  $D_2 = 7$ ,  $D_1 = 8$ ,  $D_0 = 9$ )
- $I_x$  là 4 ký số được nhập lần lượt tại ngõ vào ( $I_x$  có 4 bit, **số  $I_x$  sẽ được sinh ngẫu nhiên lúc báo cáo**, sinh viên có thể sử dụng chức năng sinh số ngẫu nhiên trong phần mềm mô phỏng quartus để kiểm tra thiết kế)

### HƯỚNG DẪN THỰC HÀNH:

Thực hiện thiết kế vi xử lý đơn giản để tính toán biểu thức:  $2I_3 - 1I_2 + 4I_1 - 2I_0$

**Lưu ý:** Phần hướng dẫn chỉ mang tính chất tham khảo, biểu thức trong phần hướng dẫn thực hành **KHÁC** với biểu thức trong phần thực hành.

Bước 1: Xác định các bước giải quyết bài toán

Bảng 1 - Các bước thực hiện để tính  $2I_3 - 1I_2 + 4I_1 - 2I_0$

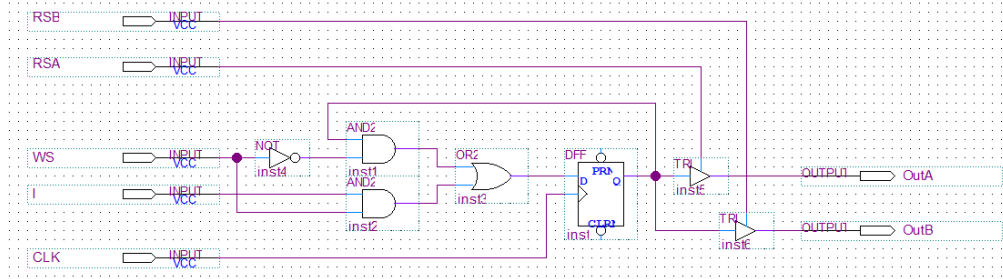
Bước	Công việc
1	$\text{Data0} \leftarrow I_0$
2	$\text{Data1} \leftarrow I_1$
3	$\text{Data2} \leftarrow I_2$
4	$\text{Data3} \leftarrow I_3$
5	$\text{Data0} \leftarrow \text{Data0} \ll 1$
6	$\text{Data1} \leftarrow \text{Data1} \ll 2$
7	$\text{Data2} \leftarrow \text{Data2} \& \text{Data2}$
8	$\text{Data3} \leftarrow \text{Data3} \ll 1$
9	$\text{Sum} \leftarrow \text{Data3} + \text{Data1}$
10	$\text{Temp} \leftarrow \text{Data2} + \text{Data0}$
11	$\text{Sum} \leftarrow \text{Sum} - \text{Temp}$
12	$\text{Output} \leftarrow \text{Sum} \& \text{Sum}$

Bước 2: Xác định các khối cần thiết để thực hiện tính toán

Từ bảng 1, có thể thấy:

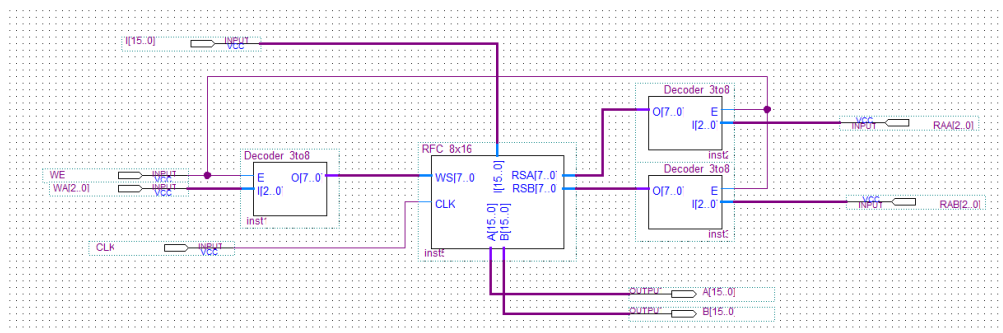
- Nhập dữ liệu ( $I_x$ ) cần 4 bit, nhưng phải mở rộng dấu lên 16 bit để thực hiện tính toán sau này.
- Có 2 nguồn gán giá trị cho biến (một là từ đầu vào và hai là từ các phép tính toán) nên sẽ cần sử dụng Mux 2 16 bit.
- Chúng ta sẽ sử dụng 6 biến có dấu (Data0, Data1, Data2, Data3, Sum, Temp). Vì thế cần sử dụng Register File có tối thiểu 8 thanh ghi 16 bit có chức năng đọc và ghi dữ liệu.
- ALU 16 bit để thực hiện các phép toán.
- Một bộ dịch trái tối đa 2 bit
- Xuất kết quả cần sử dụng thanh ghi 16 bit có khả năng cho phép/không cho phép xuất giá trị tại ngõ ra (sử dụng cổng tri-state)

a. Register File 8 thanh ghi 16 bit



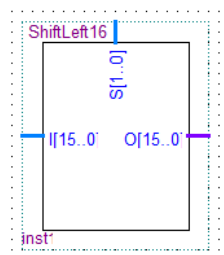
Hình 1 - Register File cell 1 bit

Tiến hành ghép nối nhiều RFC để tạo nên Register File 8 thanh ghi 16 bit có chức năng đọc và ghi dữ liệu.



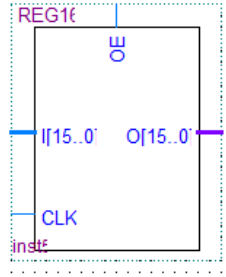
Hình 2 - Register File 8 thanh ghi 16 bit

- b. ALU có các phép toán cần thiết cho yêu cầu bài toán (đã hướng dẫn ở Lab 3)
- c. Bộ dịch trái: Sử dụng Mux và ghép nối dây phù hợp để tạo ra bộ dịch trái



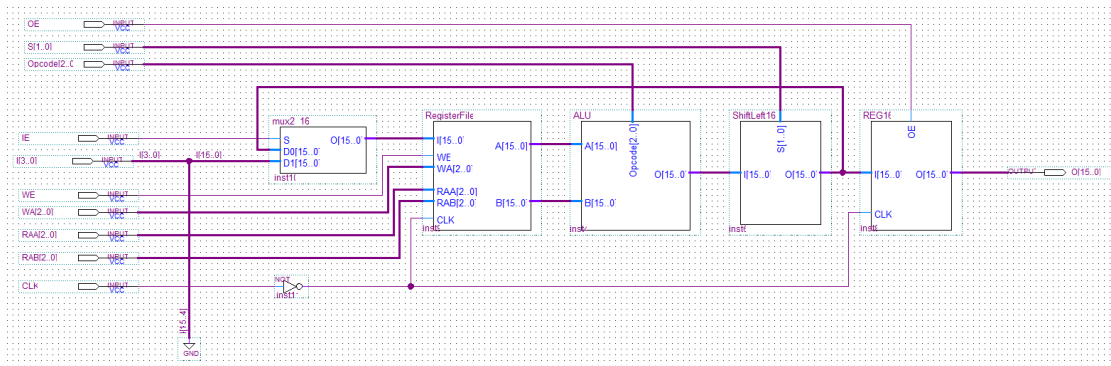
Hình 3 - Bộ dịch trái

- d. Thanh ghi 16 bit: Ghép nhiều thanh ghi 1 bit lại với nhau để được thanh ghi 16 bit sử dụng tín hiệu OE (output enable) để cho phép/không cho phép xuất kết quả (sử dụng tri-state)



Hình 4 - Thanh ghi 16 bit

e. Datapath: Thực hiện ghép nối các khối cần thiết của bài toán lại để có được Datapath



Hình 5 - Datapath mẫu

f. Controller: Thiết kế máy trạng thái để điều khiển việc hoạt động các khối trong Datapath.

g. Thiết kế bộ vi xử lý cho bài toán: Thực hiện ghép nối Controller và Datapath với nhau, sau đó mô phỏng kiểm tra hoạt động.

**Lưu ý:** Phần hướng dẫn chỉ mang tính chất tham khảo, biểu thức trong phần hướng dẫn thực hành **KHÁC** với biểu thức trong phần thực hành.