

初探REGISTER

- 做完了**DECODER**，根據**5-STAGE**模型我們得知下個步驟是執行

*複習**5-STAGE**

INSTRUCTION FETCH->INSTRUCTION DECODE->EXECUTE->MEMORY ACCESS->WRITEBACK

提取指令

解碼

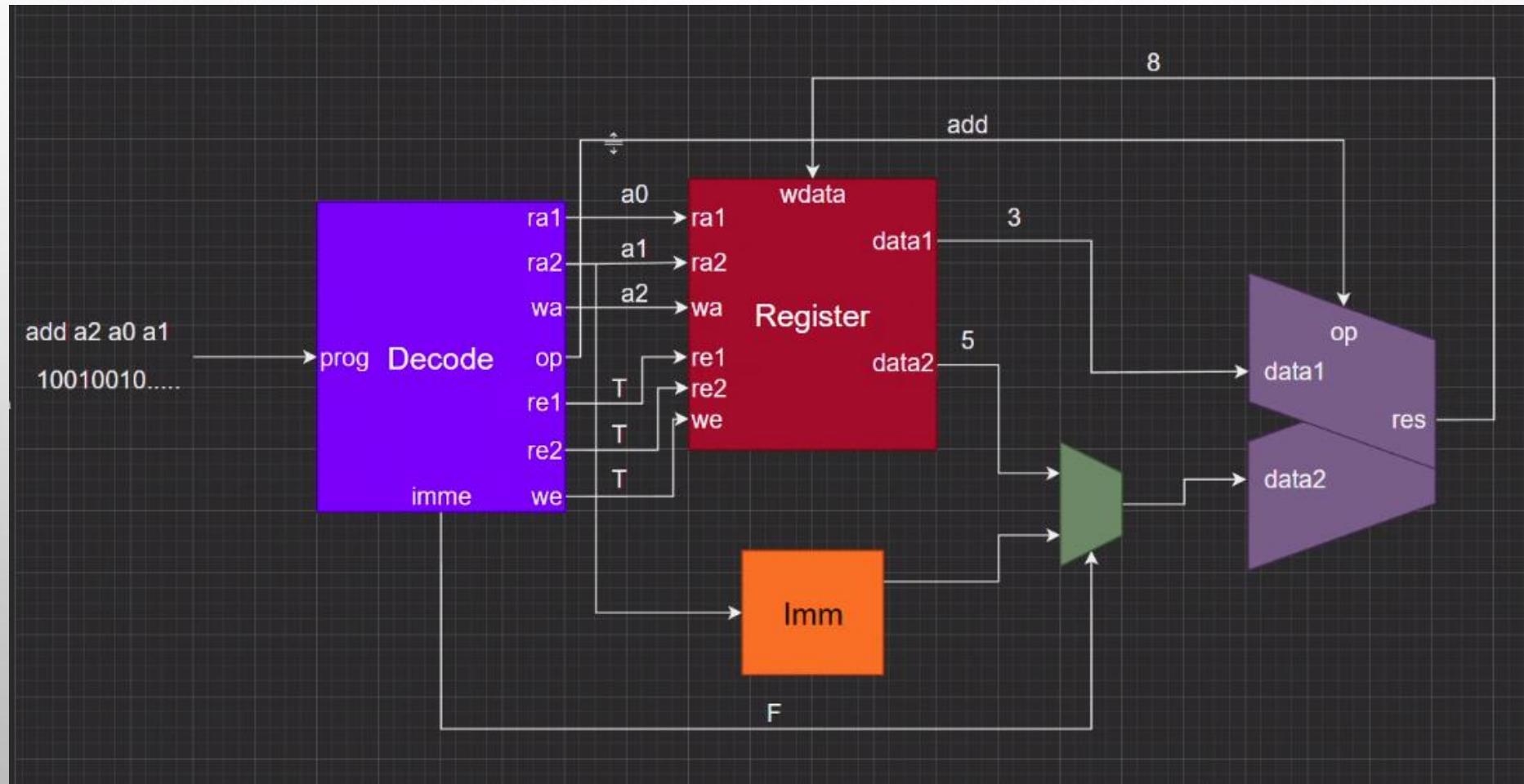
執行

存取記憶體

寫入

- 雖然運算單元是**ALU**，但執行需要多個元件參與，其中包含**REGISTER**
-**DECODER**解出的**OPCODE**是交給**ALU**，**RS1.RS2.RD**則連結到**REGISTER**

目前要實現的模型概圖



接腳

主要輸入接口:

INPUT

- CLK&RESET

- RS1和RS2的位址

- RS1和RS2各一個讀取開關

- 寫入資料與位址

- 寫入開關

OUTPUT

- RS1和RS2輸出

用VERILOG實現

- 功能:

- 要是地址為0則直接輸出0(RISC規則)
- 持續讀進寫入的資料並決定是否存入REG陣列
- 分為純讀取和讀取後寫出

```
module register(  
    input clk,  
    input reset,  
    input [4:0]rs1addr, //rs1位址  
    input [4:0]rs2addr, //rs2位址  
  
    input ren1,          //rs1 read enable  
    input ren2,          //rs2 read enable  
    input wren,          //write enable  
  
    input [31:0]wb_data, //寫入資料  
    input [4:0]wb_addr,  //寫入位址(即為rd)  
  
    output reg [31:0]rs1, //rs1輸出  
    output reg [31:0]rs2, //rs2輸出  
);  
    reg [31:0]regbus[15:0]; //register陣列  
    parameter yes= 1'b1;    //參數定義  
    parameter no= 1'b0;
```

```
always@(posedge clk) begin //如果寫入的位址為0 則不寫入
    if(reset!=yes) begin
        if((wren==yes) && (wb_addr!=5'b00000))
            regbus[wb_addr]<=wb_data;
    end
end
```

```
always@(*) begin                //如果讀取的位置為0 則直接給0
    if((ren1==yes) && (rs1addr==5'b00000))
        rs1=0;
    else if((ren1==yes) && (rs1addr==wb_addr) && (wren==yes))
        //可同時寫入及讀取的話 就將輸入直接給輸出
        rs1=wb_data;
    else if((ren1==yes) && (rs1addr!=5'b00000)) //正常讀取
        rs1=regbus[rs1addr];
    else                          //預設輸出0
        rs1=0;
end
```



```

always@(*) begin
    if((ren2==yes) && (rs2addr==5'b00000))    //如果讀取的位置為0 則直接給0
        rs2=0;
    else if((ren2==yes) && (rs2addr!=5'b00000) && (wren==yes))
        //可同時寫入及讀取的話 就將輸入直接給輸出
        rs2=wb_data;
    else if((ren2==yes) && (rs2addr!=5'b00000))    //正常讀取
        rs2=regbus[rs2addr];
    else
        //預設輸出0
        rs2=0;
end
endmodule

```