


Single Processor Design Report

 Author 111511198 鄭恆安

1. 指令生成方法 (Generate Pattern Method)

根據專案規範（spec），處理對應的指令碼（opcode），並正確計算以下兩項：

- **golden_inst_addr**：指令的目標地址。
- **golden_r**：運算的正確結果。

在指令解碼的過程中，確保輸出符合給定的測試數據，以驗證設計的正確性。

2. 設計架構 (Design Architecture)

Non-Pipeline Architecture

採用有限狀態機（FSM）設計，共包含以下五個狀態，並按順序執行：

1. **IF (Instruction Fetch)**: 提取指令。
2. **ID (Instruction Decode)**: 解碼指令。
3. **EX (Execute)**: 執行運算。
4. **MEM (Memory Access)**: 存取記憶體。
5. **WB (Write Back)**: 寫回結果。

FSM 的狀態定義如下：

```
localparam IF   = 3'b000;
localparam ID   = 3'b001;
localparam EX   = 3'b010;
localparam MEM  = 3'b011;
localparam WB   = 3'b100;
```

Pipeline Architecture

定義四個階段（Stage）：

- **Stage 0**: 解碼（Decode）。
- **Stage 1**: 運算（EX）。
- **Stage 2**: 記憶體存取（MEM）。
- **Stage 3**: 寫回（WB）。

每個時鐘週期（cycle），將當前計算結果及相關數值傳遞到下一個階段，實現pipeline提升效能。

3. 課程回饋 (Course Feedback)

老師的講解內容非常清楚且用心，幫助我們快速理解處理器設計的基本原理。但整學期課程中未涉及 Verilog 的介紹，讓期末專案從零開始撰寫 Verilog 程式碼時，感到有些吃力。建議未來能在課程中加入 Verilog 的基礎教學，幫助學生更好地應對類似的專案挑戰。