Spec Document

Microprocessor Specification Document

Introduction

- The microprocessor is a simplified 8-bit design intended for basic arithmetic, data transfer, and I/O operations.
- It supports six general-purpose registers (A–F), an 8-bit instruction set, and a shared internal bus for data movement.
- The processor is designed to execute five basic instructions: **ADD, SUB, MOV, IN, and OUT**.

Instruction Set Overview

The microprocessor supports the following 8-bit instructions:

- **ADD R**: Adds the value in register R to the accumulator (A) and stores the result back in A.
- **SUB R**: Subtracts the value in register R from the accumulator (A) and stores the result back in A.
- **MOV R1, R2**: Copies the value from register R2 into register R1.
- **IN R**: Loads an 8-bit external value (`data_in`) into register R.
- **OUT R**: Place the contents of register R onto the output bus (`data_out`).

Inst[7:6]	**Inst[5:3]**	**Inst[2:0]**	**Operation**	
00	DDD	000	IN D	D ← input
00	DDD	SSS	Mov D,S	D ← S
00	000	SSS	OUT S	output ← S
01	0XX	S	ADD S	A ← A+S
01	1XX	S	SUB S	A ← A-S

Module Interface

Top-level Module

```
module micro(
    input clk,
    input [7:0] inst,
    input [7:0] data_in,
    output [7:0] data_out
);
```

Port Description

Signal	Direction	Width	Description
clk	input	1	System clock. All register operations occur on the rising edge.
inst	input	8	Instruction word controlling the operation (opcode and register selection).
data_in	input	8	External 8-bit data input bus.
data_out	output	8	External 8-bit data output bus.

Internal Architecture

The microprocessor consists of the following components:

1. **Registers (A–F)**
   - Six 8-bit registers store intermediate data and results.
   - Register A functions as the **accumulator**, used in arithmetic operations.

2. **ALU (Arithmetic Logic Unit)**
   - Performs addition and subtraction between register A and the shared data bus.
   - Controlled by instruction bit `inst[5]`.
     - `inst[5] = 0`: Perform ADD.
     - `inst[5] = 1`: Perform SUB.

3. **Decoder (3-to-8)**
   - Two decoders are used:
     - **Load Decoder (LD)**: Determines which register or output is enabled for writing.
     - **Output Enable Decoder (OE)**: Determines which register drives the shared bus.

4. **Multiplexers and Tri-state Buffers**
    - Control data flow between registers, the ALU, and the bus.
    - Ensure only one source drives the bus at a time.

5. **D Flip-Flops**
    - Each register (A–F) is implemented using an 8-bit DFF.
    - Data is updated on the rising edge of the clock.

Expected Operation

- **Arithmetic Operations**: The ALU must correctly perform addition and subtraction according to `inst[5]`.

- **Register Loads**: All registers (A–F) must be loadable via `LD` signals.

- **Data Transfers**: MOV and IN instructions should correctly transfer values into the selected register.

- **Outputs**: OUT instructions must place the correct register contents on the output bus without floating (`Z`) states.

Observed Issues

During simulation of the current RTL design, the following problems were observed:

1. **Arithmetic Inversion**

    - ADD instructions behave like subtraction, and SUB behaves like addition.

    - This suggests incorrect control mapping in the ALU's `add_sub` logic.

2. **Register C Not Updating**

    - MOV and IN instructions targeting register C do not modify its value.

    - Likely due to missing or incorrect assignment logic in the register load block.

3. **OUT A Produces High-Impedance (Z)**

    - When executing OUT A, the bus shows `Z` instead of A's value.

    - Indicates a wiring issue in the tri-state buffer connecting register A to the bus.

Debugging Guidelines

1. **Check ALU Mapping**
    - Verify that `inst[5]` correctly selects between addition and subtraction.

2. **Inspect Register Load Logic**
    - Ensure all registers (B–F) have proper load paths from the bus.
    - Confirm that the decoder correctly asserts `LD[i]` for all registers.

3. **Validate Tri-state Buffer Wiring**
    - For all registers, ensure its tri-state buffer connects to the bus when `OE[i]` is asserted.
    - Verify bus contention does not occur with other outputs.

Timing and Latency

The microprocessor operates synchronously with the following characteristics:

Operation	Latency (Clock Cycles)
Arithmetic (ADD, SUB)	1 cycle (results written to A on next rising edge)
Register Load	1 cycle
IN Operation	1 cycle
OUT Operation	1 cycle

Deliverables

- Corrected Verilog RTL code .
- Simulation results showing correct execution of ADD, SUB, MOV, IN, OUT instructions.
- A short bug report describing the identified issues and fixes applied.