SLOVE AND FAST DIVISION ALGORITHM IN COMPUTER ARCHITECTURE

Module: Rest_div

- Description: The module implements a division algorithm using a finite state machine (FSM). It performs division on a 4-bit quotient (Q) and a 4-bit divisor (M) and provides the quotient, remainder, and valid output.
- Inputs:
    - clk: Clock signal for synchronous operation.
    - rst: Reset signal for initializing the module.
    - Q: 4-bit input representing the quotient. ● M: 4-bit input representing the divisor.
- Outputs:
    - quot: 4-bit output representing the quotient.
    - rem: 4-bit output representing the remainder.
    - valid: Output signal indicating when the quotient and remainder are valid.
- Registers:
    - Acc: 8-bit register holding the current value of the accumulator.
    - next_Acc: 8-bit register holding the next value of the accumulator.
    - present_state: 2-bit register holding the current state of the FSM.
    - next_state: 2-bit register holding the next state of the FSM.
    - count: 2-bit register holding the current count value.
    - next_count: 2-bit register holding the next count value.
    - valid: Register indicating the current validity status of the quotient and remainder.
    - next_valid: Register indicating the next validity status of the quotient and remainder.
- Parameters:
    - IDLE: 2-bit parameter representing the IDLE state of the FSM.
    - SHIFT: 2-bit parameter representing the SHIFT state of the FSM.
    - SUBTRACT: 2-bit parameter representing the SUBTRACT state of the FSM. ● RESTORE: 2-bit parameter representing the RESTORE state of the FSM.

- Always Block (Synchronous):
- Combinational Assignments:
    - rem: Assigns the value of Acc[7:4] to rem when valid is 1, otherwise assigns 3'b0.
    - quot: Assigns the value of Acc[3:0] to quot when valid is 1, otherwise assigns 3'b0.
    - Executes on a positive edge of clk or a negative edge of rst.
    - Handles the initialization and updates of registers based on the state transitions.
- Always_comb Block (Combinational):
    - Executes whenever there is a change in the inputs.
    - Implements the FSM using a case statement to perform the division algorithm.
    - Handles the state transitions and updates the next values of registers accordingly.

The module represents a division algorithm that calculates the quotient and remainder using a finite state machine. It performs shifting, subtracting, and restoring operations based on the current state of the FSM. The division process starts in the IDLE state, shifts the accumulator in the SHIFT state, subtracts in the SUBTRACT state, and restores the accumulator in the RESTORE state. The division is complete when the count reaches 2'b11, and the valid signal is set to 1 to indicate that the quotient and remainder are valid.