Problem Statement: Sequential Summation System

Objective

Design and implement a sequential summation system in Verilog HDL that computes the sum of the first N natural numbers (i.e., 1 + 2 + 3 + ... + N) using an FSM-based controller and a datapath consisting of an accumulator and a counter. The system must enforce proper handshaking signals, validate input constraints, and avoid the use of multipliers.

System Description

• The system accepts an external 8-bit input N_in.
• The maximum valid value of N is 45.
• The system is controlled by the following states: IDLE, BUSY, DONE, with implicit RESET behavior.

Port Specifications

Inputs

• clk: System clock (positive-edge triggered).
• rst: Active-high synchronous reset; forces the system to reset into the IDLE state.
• go: Start signal (1-cycle pulse) to begin computation.
• N_in[7:0]: User-supplied number N.

Outputs

• idle: High when the system is waiting in IDLE state.
• busy: High while computation is ongoing in BUSY state.
• done: High for exactly 1 clock cycle when summation is completed.
• invalid: High if N = 0 or N > 45. In this case, go is ignored.
• sum_out[10:0]: Final computed summation value (fits in 11 bits since max = 1035).

Functional Requirements

1. Reset Behavior:
  - On rst=1, system enters IDLE state.
  - idle=1, busy=0, done=0, invalid=0, sum_out=0.

2. Input Validation:
  - If N_in = 0 or N_in > 45 → invalid=1 in IDLE.
  - While invalid=1, go has no effect.

3. Valid Operation:
  - If 1 ≤ N_in ≤ 45 and go=1 (in IDLE), system latches N_in and enters BUSY.
  - In BUSY, the system sequentially adds numbers 1 through N using only an incrementing counter and accumulator (no multipliers).
  - BUSY lasts exactly N clock cycles.

4. Completion:
  - After N additions, system moves to DONE.
  - In DONE, done=1 for exactly 1 cycle, sum_out holds the computed result.
  - System then automatically returns to IDLE.

5. Error Handling:
  - If N_in changes during BUSY, it is ignored; computation proceeds with the latched N.
  - Reset during BUSY aborts the computation and returns system to IDLE.

Deliverables

1. Synthesizable Verilog RTL code.
2. Self-checking testbench with at least the following cases:
  - N=0, N=46 (invalid cases).
  - N=1, N=5, N=10, N=45.
  - Change N_in during BUSY (ignored).
  - Reset during BUSY (abort).
3. Timing diagram or waveform for at least one valid and one invalid run.
4. Block diagram showing FSM + datapath (with ports).