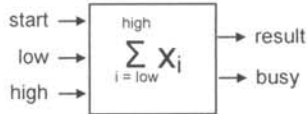
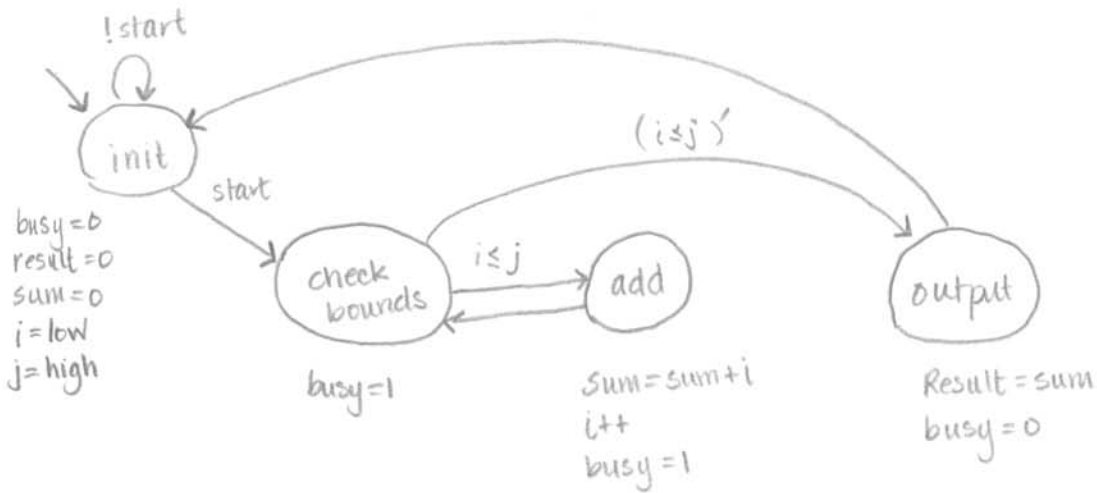


Problem 3 (20 points) – Using the RTL design method, create a circuit to perform a summation operation. A `start` bit indicates when to perform the operation, once the operation has begun a `busy` signal is asserted (set to 1) to indicate the circuit is busy calculating the result. The summation bounds are specified by two 4-bit inputs, `low` and `high`. Only when the circuit has completed the summation operation, the resulting value is outputted on a 7-bit signal named `result`, and the `busy` signal is de-asserted (set to 0). (Example: `low = 2, high = 5, result = 2 + 3 + 4 + 5 = 14`).



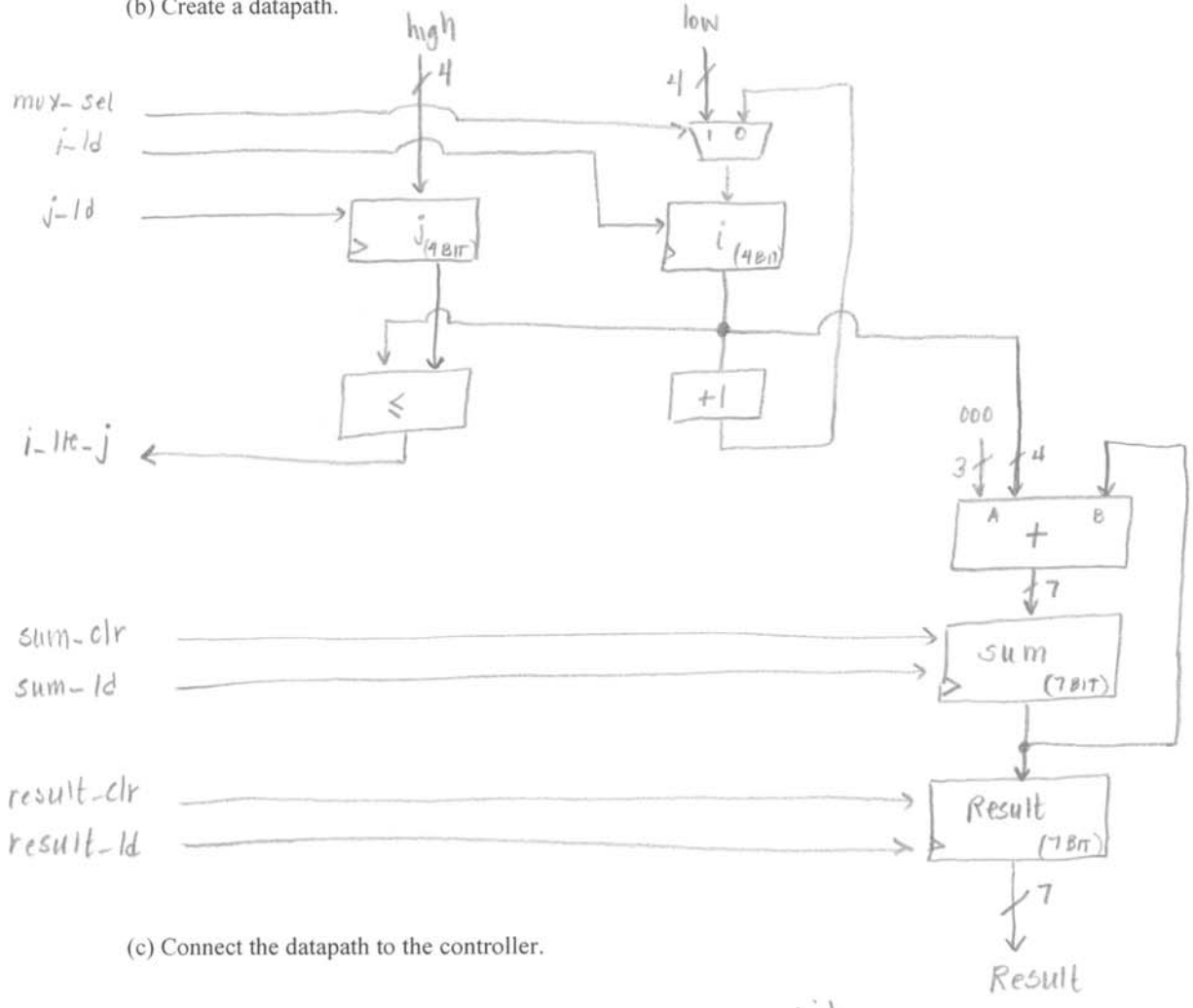
(a) Create a high-level state machine.

inputs: `start` (bit), `low` (4 bits), `high` (4 bits)
 outputs: `result` (7 bits), `busy` (bit)
 local reg: `sum` (7 bits), `i` (4 bits), `j` (4 bits)

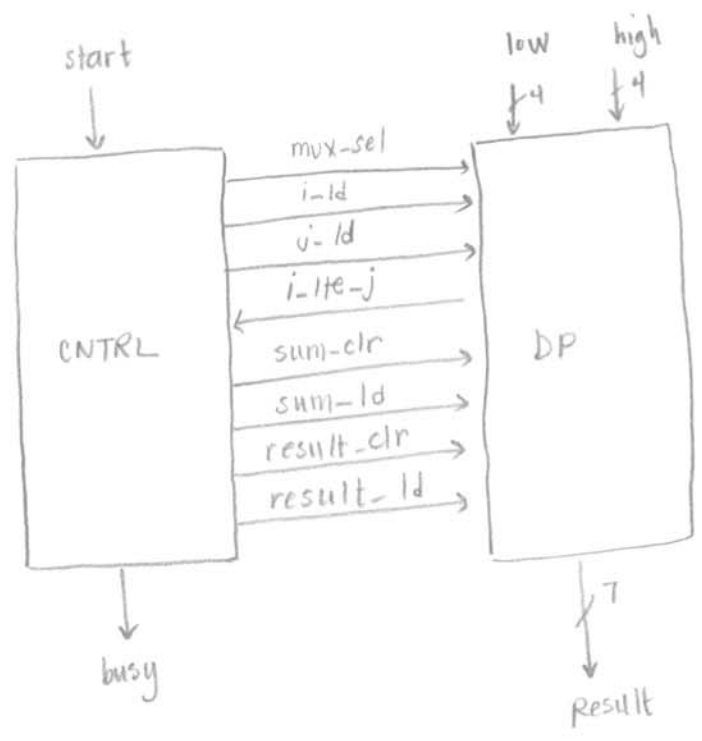


Note: `low` & `high` are stored locally to ensure their values do not change during the summation operation.

(b) Create a datapath.



(c) Connect the datapath to the controller.



(d) Derive the controller's FSM.

inputs: start, $i \leq j$

outputs: busy, mux_sel, i_ld, j_ld, sum_clr, sum_ld, result_clr, result_ld

