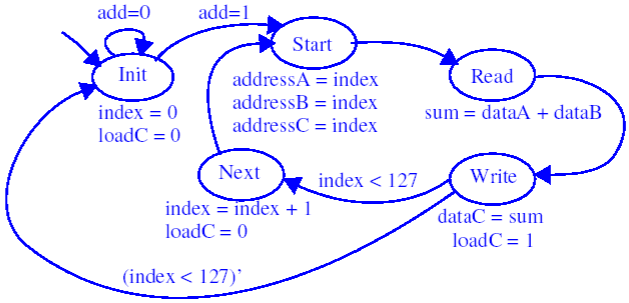
5.5 (a) Create a high-level state machine that adds each register of one 128x8 register file to the corresponding registers of another 128x8 register file, storing the results in a third 128x8 register file. The system should only begin the addition when a bit-input `add` is `1`, and should not perform the addition again until it has finished adding (only adding again if `add` is `1`). (b) Extend this system to either add or subtract, using an additional bit-input `op`, where `op=1` means `add`, and `op=0` means subtract.

(a)

Inputs: dataA (8 bits), dataB (8 bits), add (bit)
Outputs: addressA (7 bits), addressB (7 bits), addressC (7 bits), dataC (8 bits), loadC (bit)
Local registers: sum (8 bits), index (7 bits)



(b)

Inputs: dataA (8 bits), dataB (8 bits), go (bit), op (bit)
Outputs: addressA (7 bits), addressB (7 bits), addressC (7 bits), dataC (8 bits), loadC (bit)
Local registers: result (8 bits), index (7 bits)