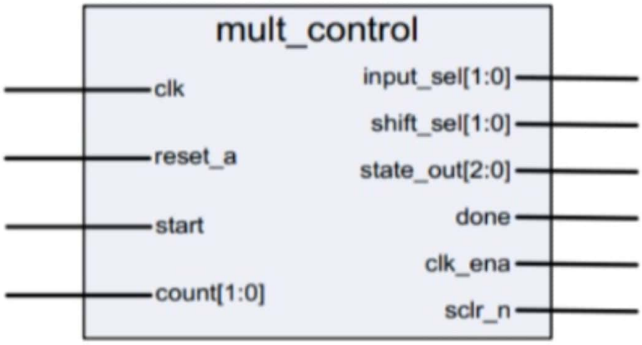
Multiplier controller



This state machine will manage all the operations that occur within the 8x8 multiplier using 6 defined states: **idle, lsb, mid, msb, calc_done, and err.**

The state machine in the **LSB** state multiplies the lowest 4 bits of the two 8-bit multiplicands ((a[3..0] * b[3..0]) * 2^0). This intermediate result is saved in an accumulator.

The state machine in the **MID** state performs cross multiplication ((a[3..0] * b[7..4]) * 2^4) and ((a[7..4] * b[3..0]) * 2^4). This is done in successive clock cycles.

The products of both multiply operations are added to the content of the accumulator as they are completed and clocked back into the accumulator.

The state machine in the **MSB** state multiplies the highest 4 bits of the two 8-bit multiplicands ((a[7..4] * b[7..4]) * 2^8).

This product is added with the content of the accumulator and clocked back into the accumulator. This result is the final product:

result[15..0] = a[7..0] * b[7..0] = ((a[7..4] * b[7..4]) * 2^8) +

((a[7..4] * b[3..0]) * 2^4) +

((a[3..0] * b[7..4]) * 2^4) +

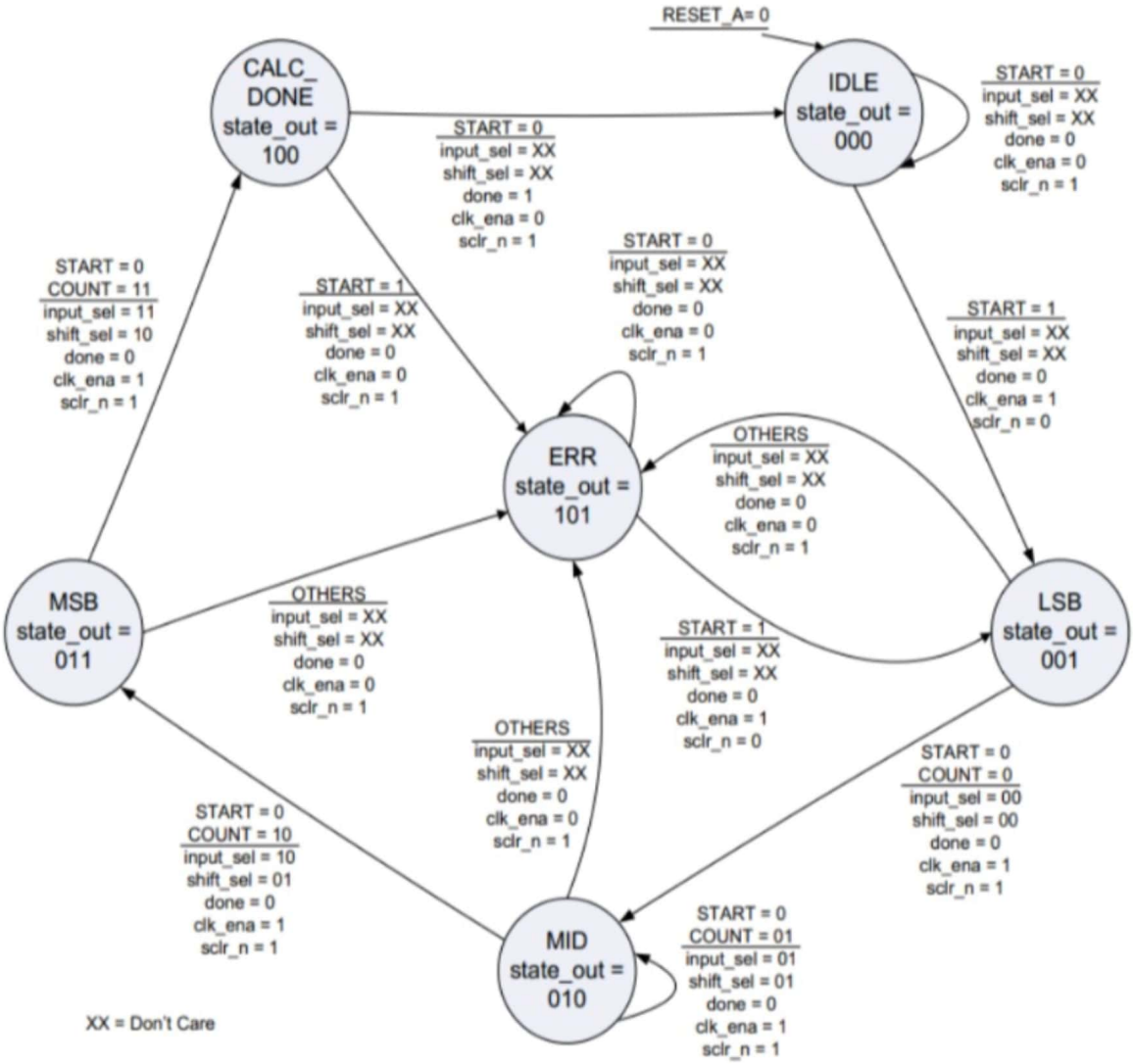
((a[3..0] * b[3..0]) * 2^0)

The state machine in the CALC_DONE state asserts the **done_flag** output to indicate the final product has been calculated and is ready for reading by downstream logic.

The state machine in the ERR state indicates incorrect inputs have been received.

There are two inputs to the state machine: start and count. The start signal is asserted for one clock cycle to begin an 8x8 multiply operation on the next clock cycle. The start signal must only be asserted for one clock cycle. The count signal is used by the state machine to track the multiplication cycles.

The outputs of **mult_control** state machine control the other blocks in the design.



- Write_Verilog code to perform this state machine.
- Write its testbench.
- Put screenshots of testbench results and circuit schematic
  (hint: to show schematic in Quiestasim, type vsim -debugDB mult_control.v in Quiestasim transcript window).

Finally, put all together a top-level design as shown in page 2 using instantiations required.

GOOD LUCK!