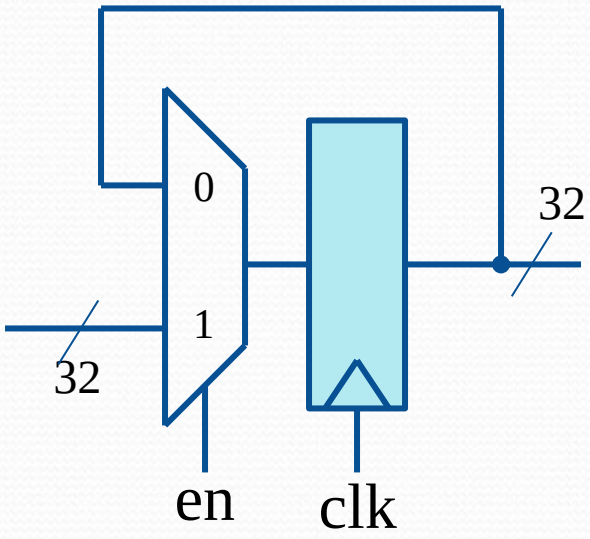
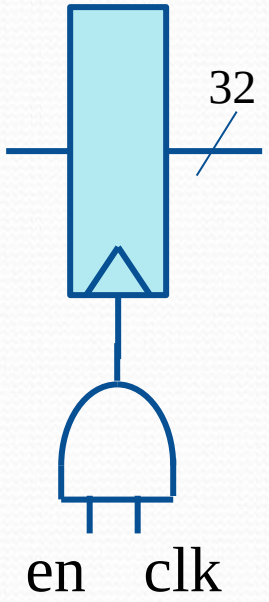
Exercise 22 Gated Clocks:

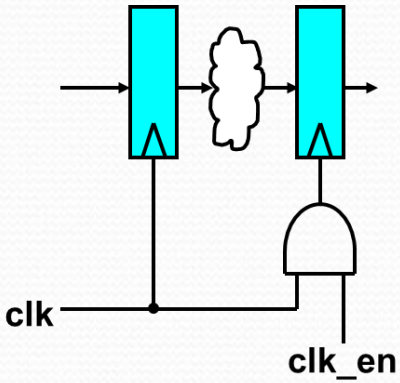




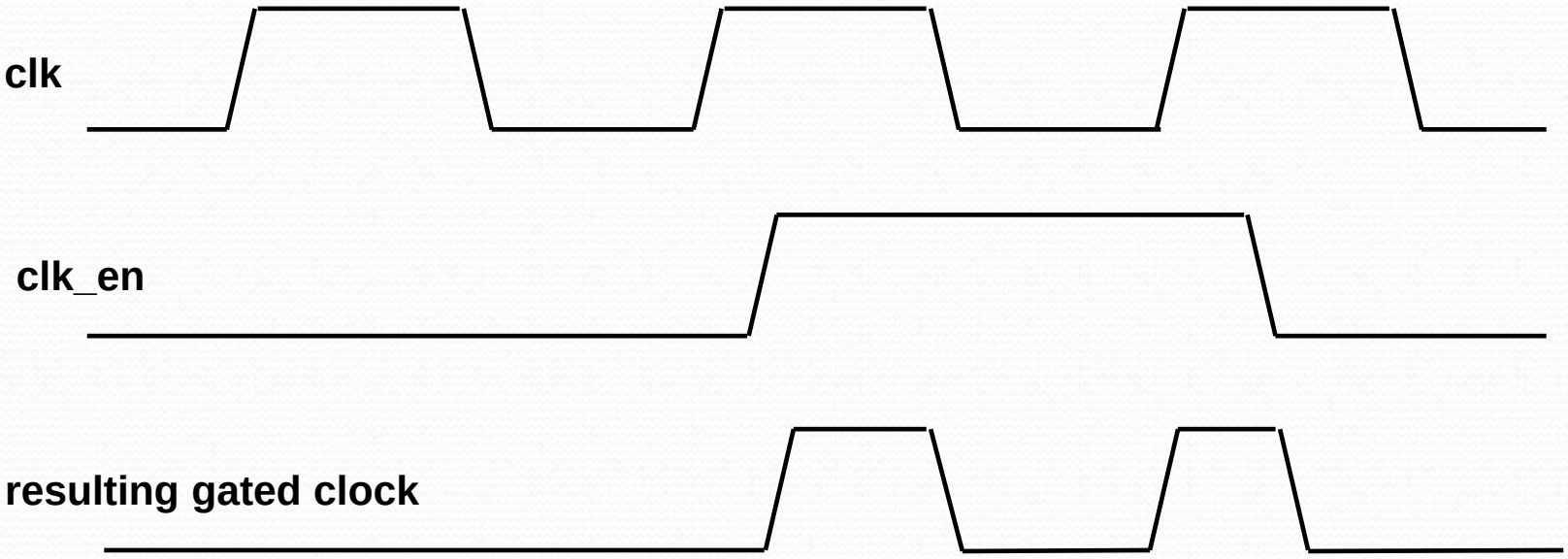
Recirculating seems rather wasteful we need a 2:1 mux for each flop. Plus all the cap and gates inside the flops on the clock network will be toggling.

Wouldn't it be better to just gate the clock? Pretty much same functionality but should save area and power? Yes, but you have to do it right.

Exercise 22 Gated Clocks:

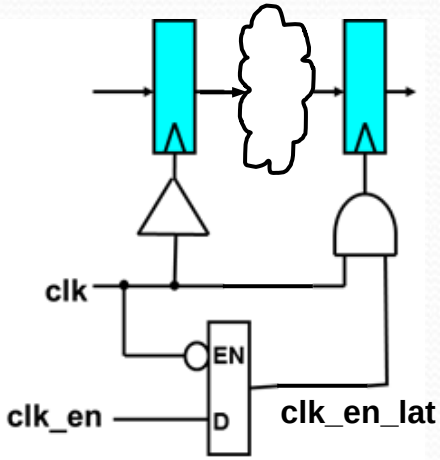


Something this simple does not work. Recall the enable function is probably coming from a SM. So it changes slightly after the rising edge of clock.



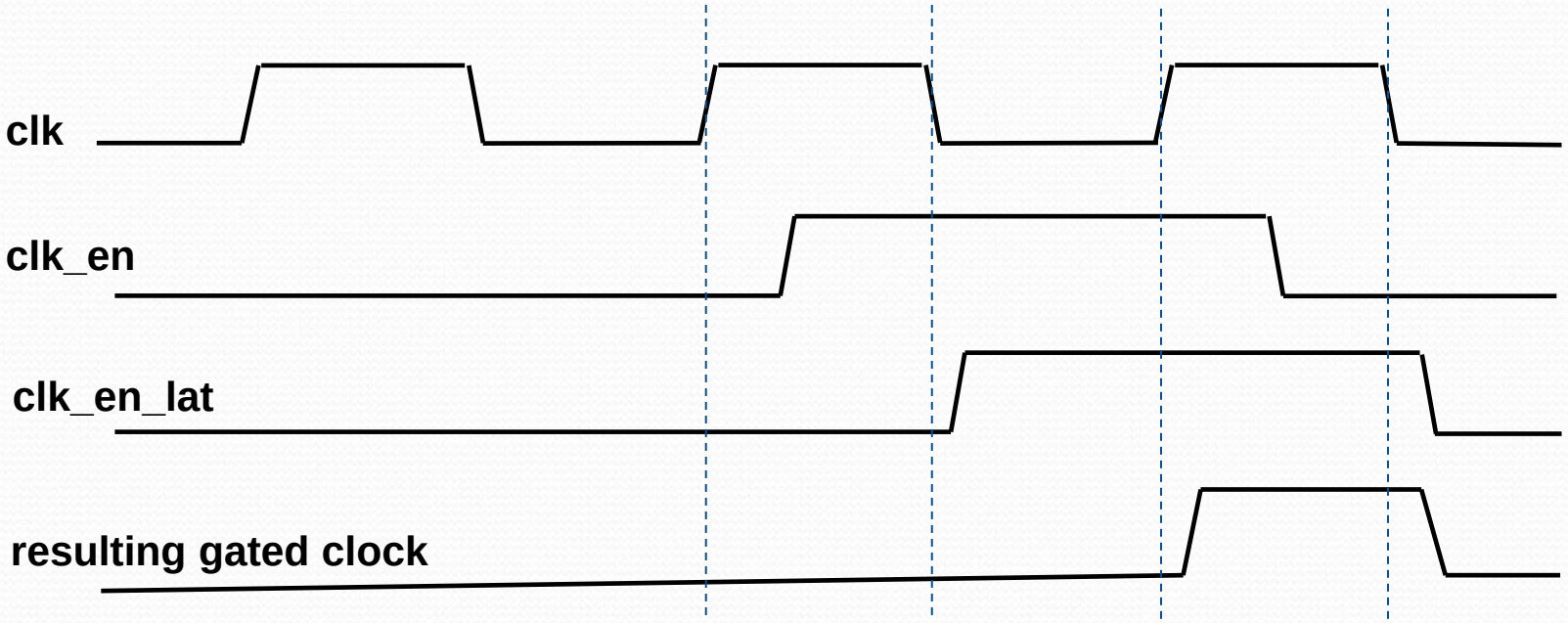
That's pretty messed up!

Exercise 22 Gated Clocks:



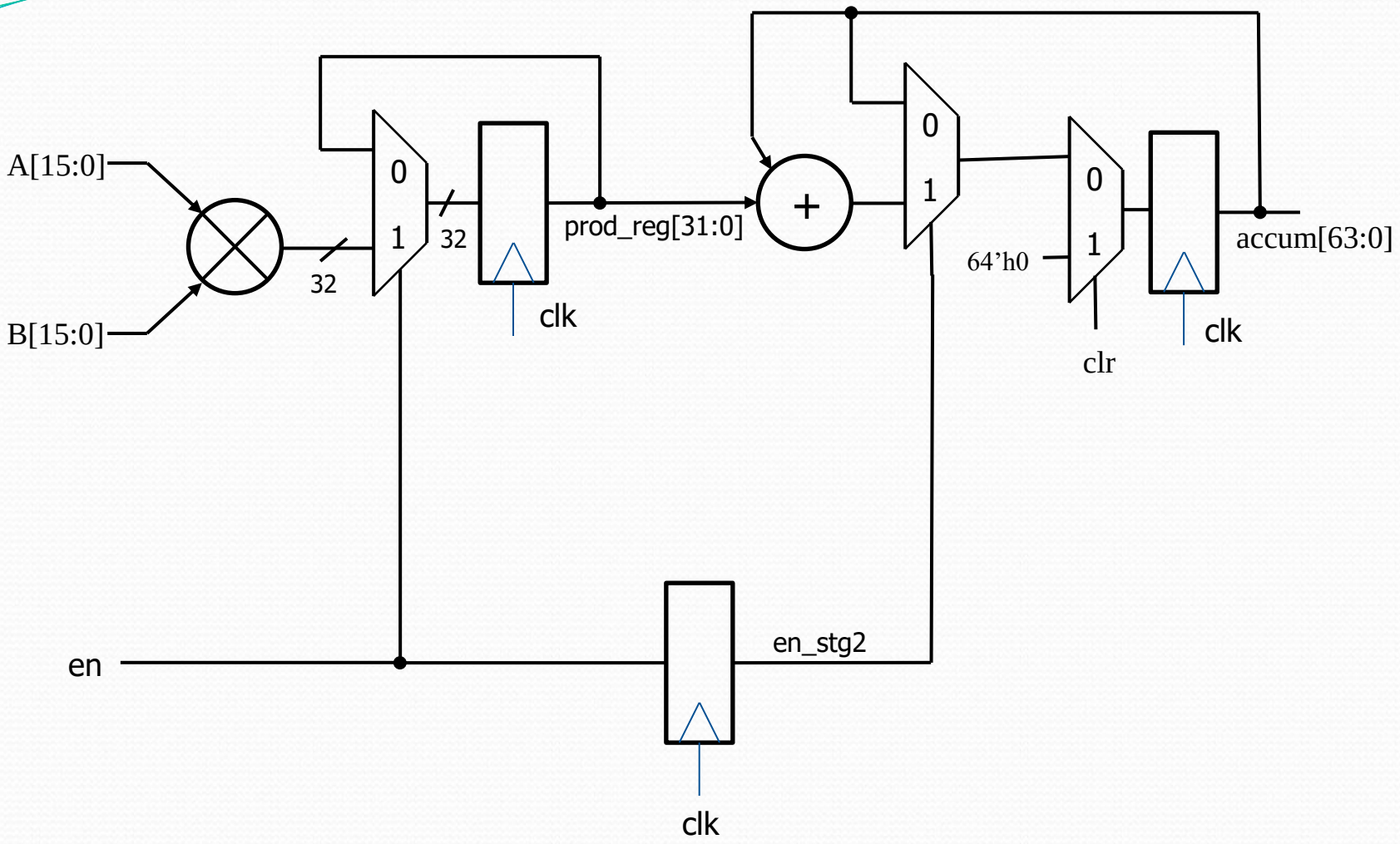
Have to use an enable low latch to latch the clock enable signal. This ensures the enable is setup and ready for the next high phase of clock

Have to balance from clock root to all leaf nodes of clock

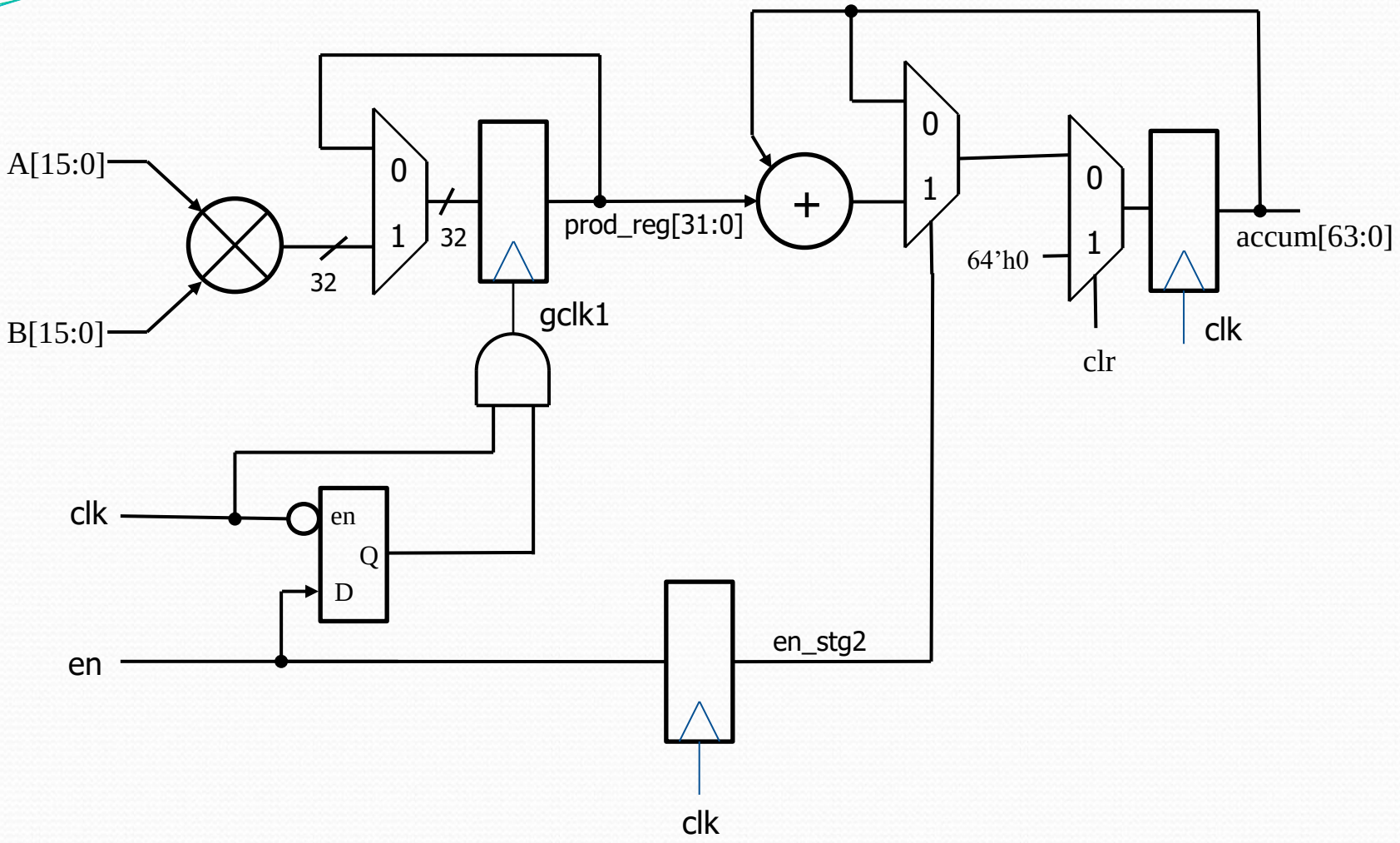


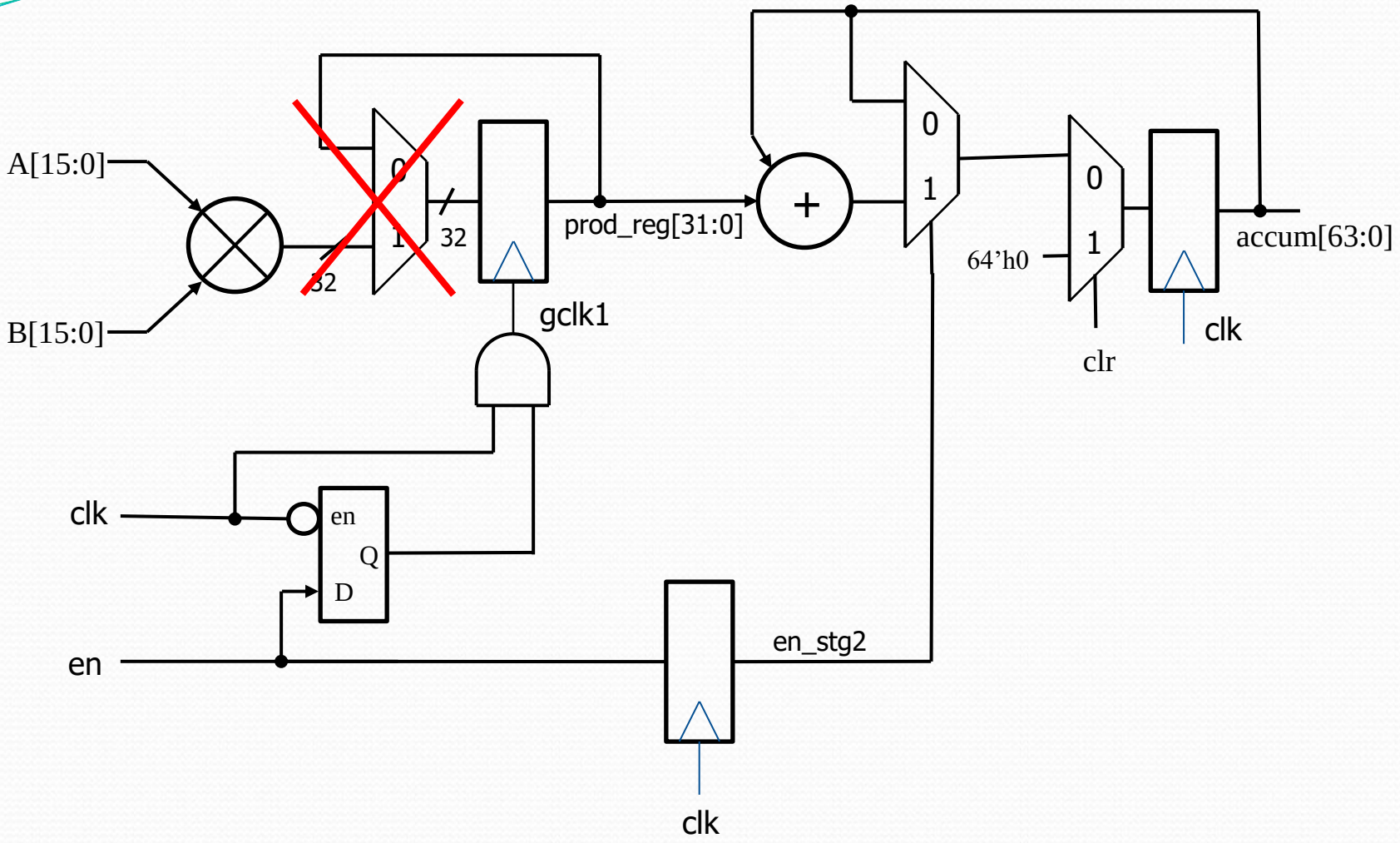
Exercise 22 Gated Clocks:

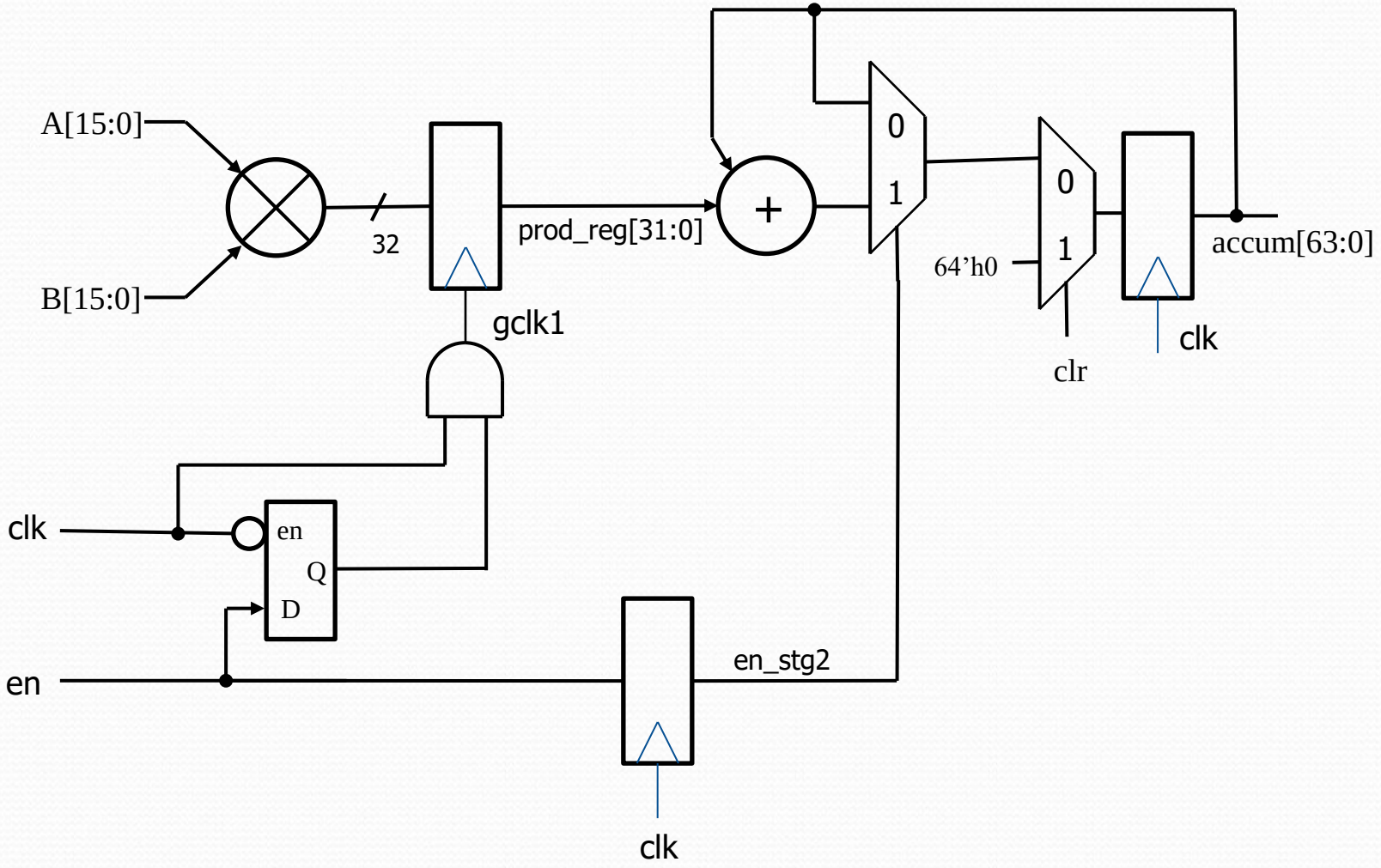
- Work in groups of two
- On the website you will find two files:
  - **mult_accum.sv** (design that uses recirculating flops)
  - **mult_accum.dc** (synthesis script)
- Synthesize **mult_accum.sv** using the provided script
  - Note the area, and the power estimate
  - Note whether or not it passes timing
- Now modify the design to use gated clocks instead of recirculating. i.e. copy **mult_accum.sv** and make **mult_accum_gated.sv**
- Modify the synthesis script as needed (minor change to read in **mult_accum_gated.sv** instead of **mult_accum.sv**)
- Again synthesize and remark on the difference in both area and power.
- Submit the area and dynamic power numbers for both designs to the dropbox for this exercise. (both people submit)

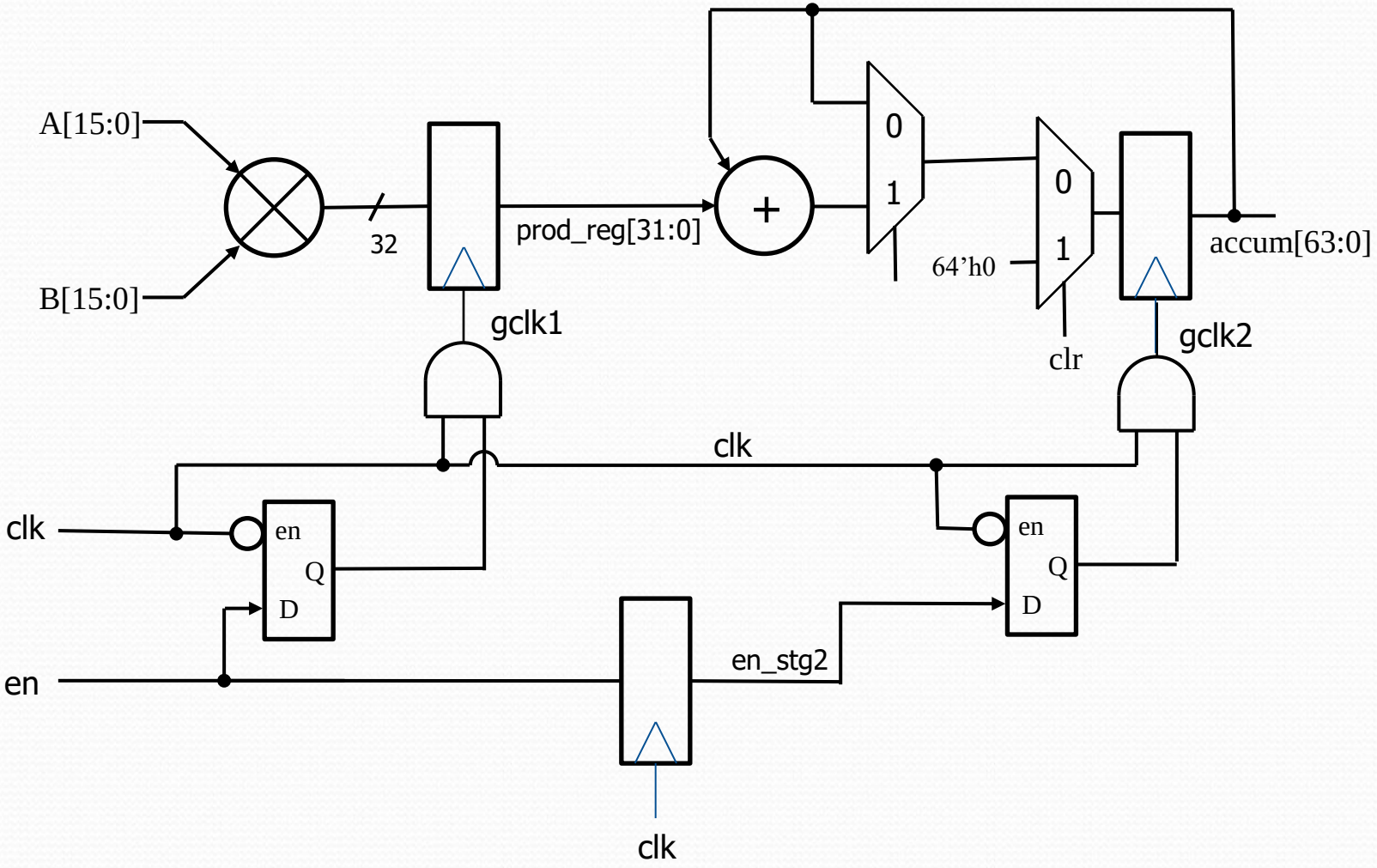


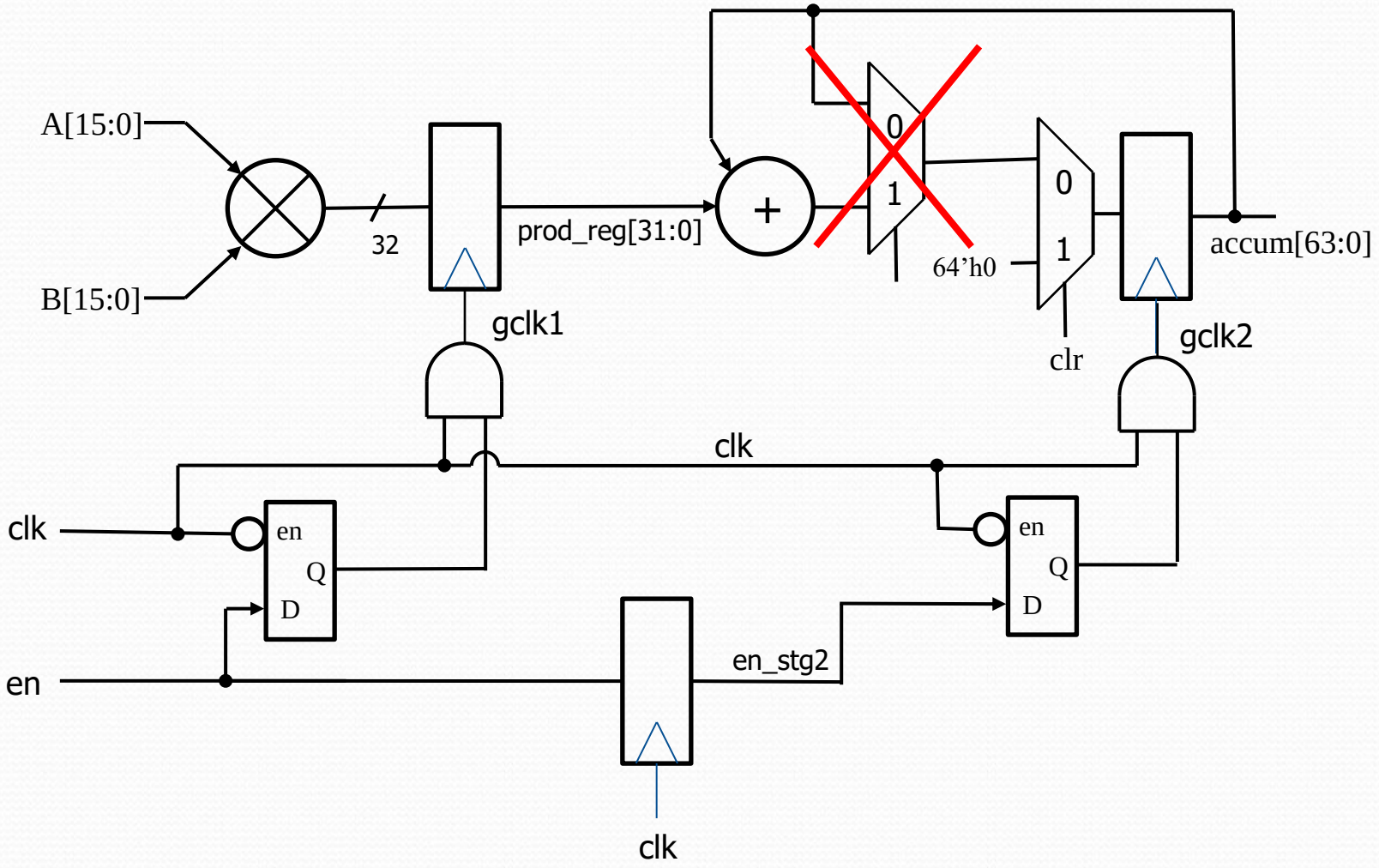
This is what we have to start

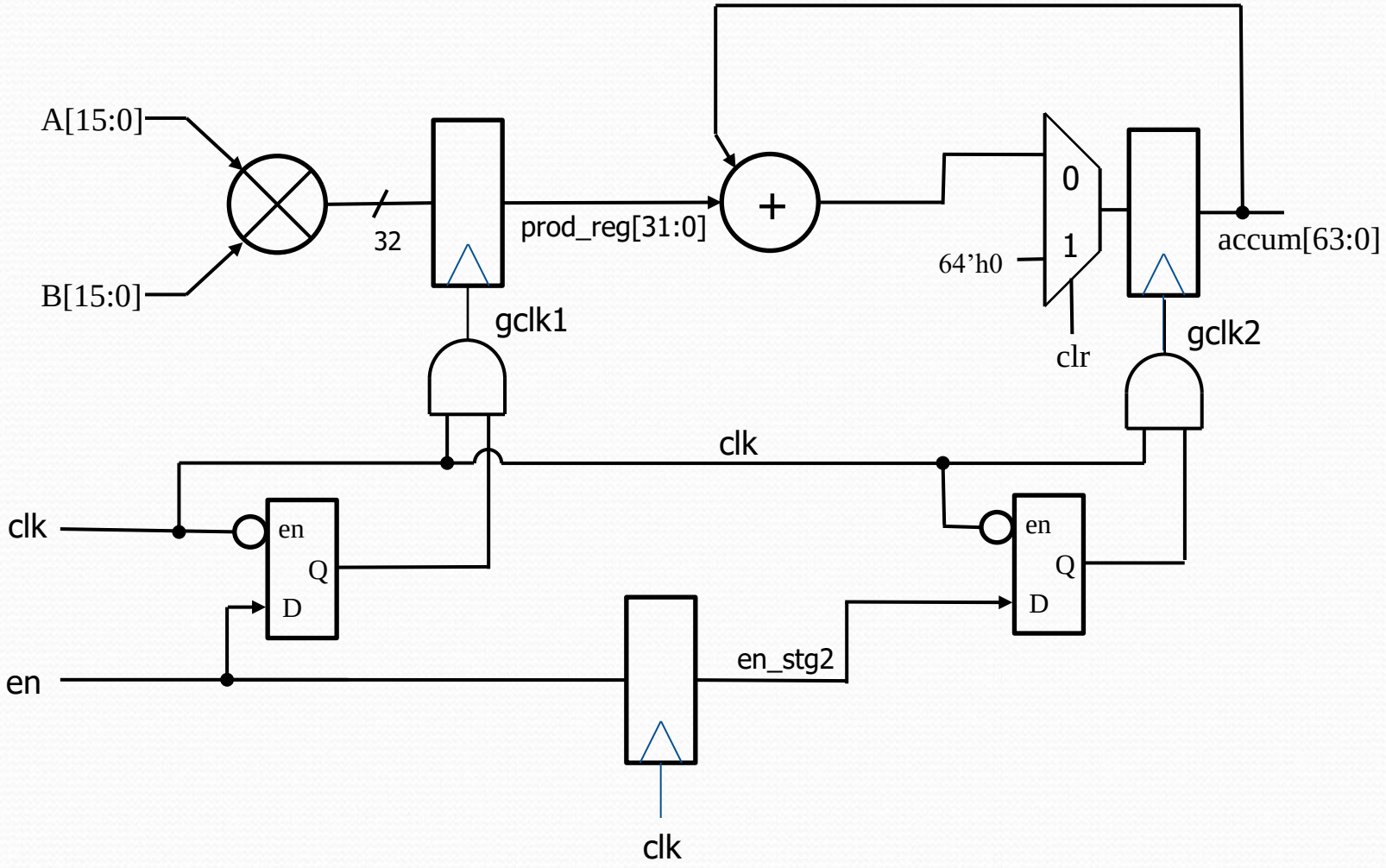

A[15:0]
B[15:0]
32
0
1
32
prod_reg[31:0]
gclk1
+
0
1
64'h0
0
1
clr
clk
accum[63:0]
clk
en
Q
D
en
en_stg2
clk

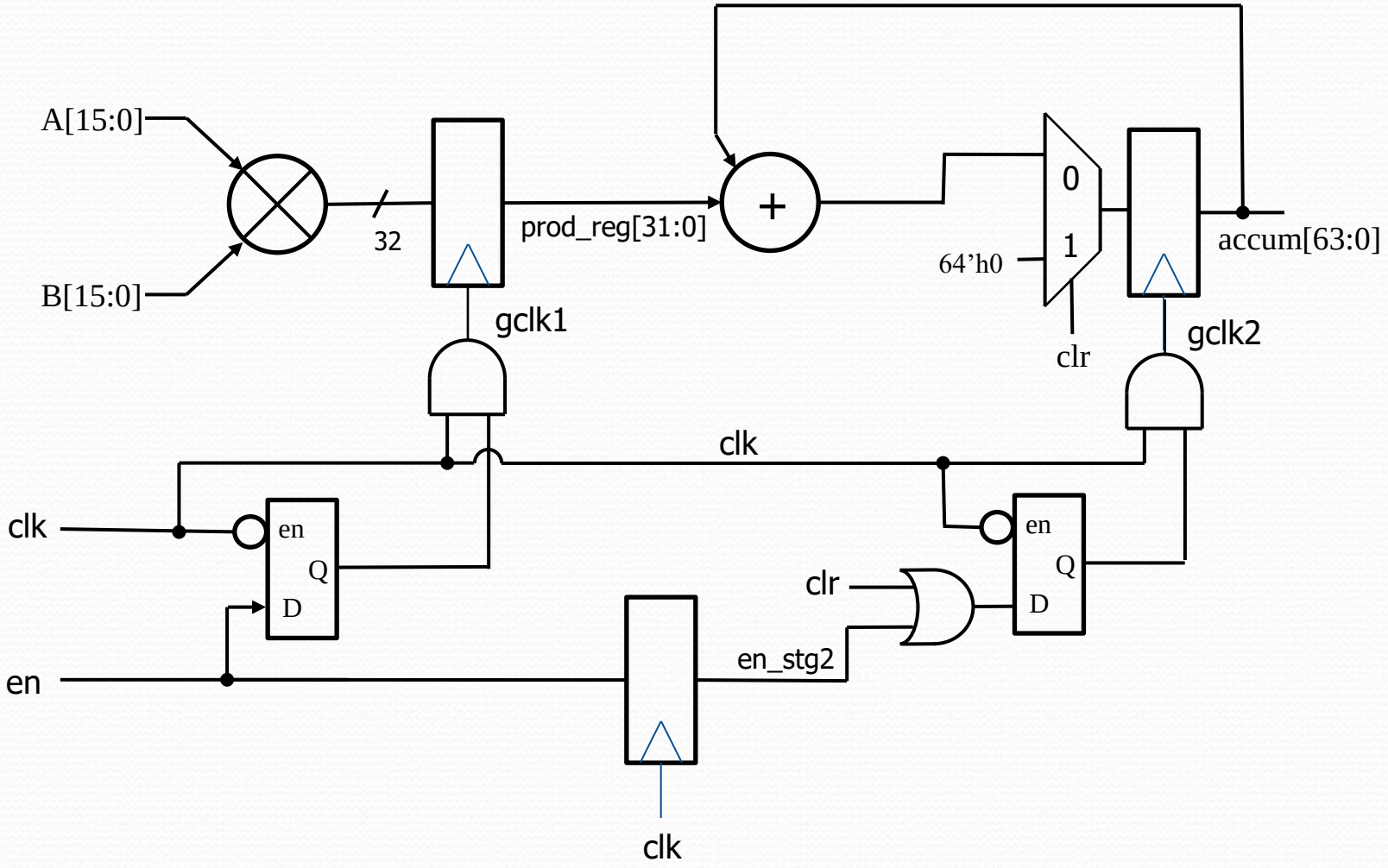

A[15:0]
B[15:0]
0
1
32
32
prod_reg[31:0]
+
0
1
0
1
64'h0
clr
clk
accum[63:0]
gclk1
clk
en
Q
D
en
en_stg2
clk


A[15:0]
B[15:0]
32
prod_reg[31:0]
gclk1
0
1
0
1
64'h0
clr
clk
accum[63:0]
clk
en
Q
D
en
en_stg2
clk


A[15:0]
B[15:0]
32
prod_reg[31:0]
+
0
1
64'h0
0
1
clr
accum[63:0]
gclk1
gclk2
clk
clk
en
Q
D
en
Q
D
en_stg2
clk


A[15:0]
B[15:0]
32
prod_reg[31:0]
+
0
1
64'h0
0
1
clr
accum[63:0]
gclk1
gclk2
clk
clk
en
Q
D
en
Q
D
en_stg2
clk


A[15:0]
B[15:0]
32
prod_reg[31:0]
+
0
1
64'h0
clr
accum[63:0]
gclk1
gclk2
clk
clk
en
Q
D
en
Q
D
en
en_stg2
clk


A[15:0]
B[15:0]
32
prod_reg[31:0]
+
0
1
64'h0
clr
accum[63:0]
gclk1
gclk2
clk
clk
en
Q
D
en
Q
D
clr
en_stg2
en
clk