Midterm 2

Create a circuit this is capable of doing division between two **5 bit unsigned integer numbers.**
Remember that given a fraction $\frac{A}{B}$, A is called the dividend and B is called the divisor. The result of
the division is called the quotient. Your division circuit should be implemented using sequential logic
and should **complete the division within 6 clock cycles** of receiving its input. If you implement your
circuit using only combinational logic you will receive no credit. You may also not implement
implement the combinational circuit in the book for division and put the result in a register.

Restricted Components
You **MAY NOT USE**: RAM, ROM, or the Divider.

Inputs

Name	Width	Description
Dividend	5	The dividend. This input will only be valid during the first clock tick that Divide is 1. This means you will have to store the Dividend.
Divisor	5	The divisor. This input will only be valid during the first clock tick that Divide is 1. This means you will have to store the Divisor.
Divide	1	When divide is 1 your process for division of the dividend by the divisor should start. Once your process begins it should ignore Divide until the division is complete. When divide is 0 your outputs should remain the same.

Outputs

Name	Width	Description
Ready	1	1 if your circuit is ready to perform the next division and 0 if it is currently doing division. When Ready is 1 it also signifies that you have completed your division.
Quotient	5	The quotient. On start up this value should be 0. If after completing the division, divide is 0 then this value should continue to hold the quotient of the most recent division. While doing division this value is a don't care.
Remainder	5	The remainder. If after completing the division, divide is 0 then this value should continue to hold the remainder of the most recent division. While doing division this value is a don't care.

Hints
1. A program that divides 32 bit numbers has been provided to you. This will help you to think about how to implement it in hardware. Try to keep in mind the limitations of the hardware and the fact that your solution must complete within 6 clock cycles.
   1. Completing within 6 clock basically means that you can't have a for/while loop that runs for more than 6 iterations.
   2. When converting to hardware all lines of a for loop are in parallel so make sure you have all the values need.
2. Components I used but not their counts: Register, Shift Register, Comparator, Mux, basic logic gates, bit shifter, bit extender, Subtractor, Counter, Constant, Ground, and Splitter.