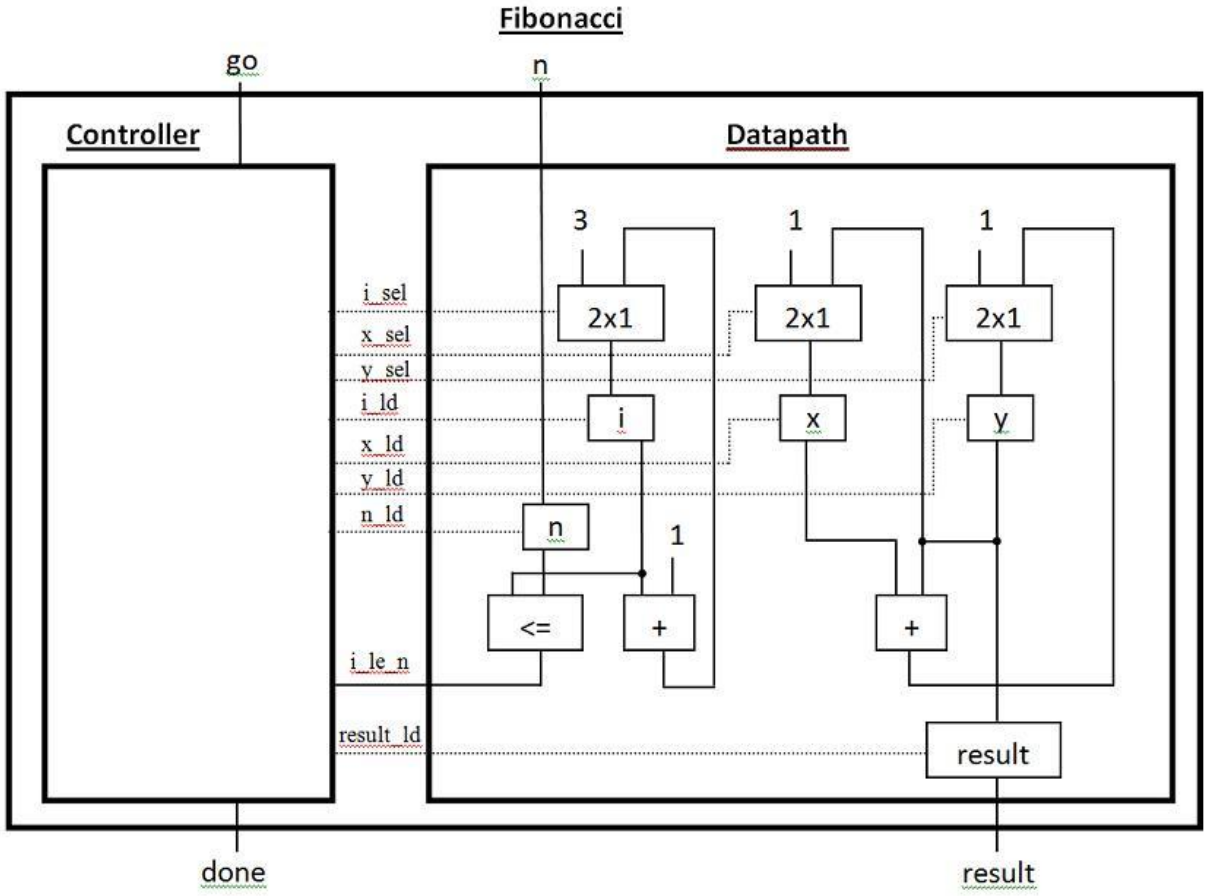
Suppose that you want to implement a circuit in VHDL that calculates Fibonacci numbers. Here is the pseudocode that describes the behavior of the circuit:

```
Input: n (specifies that the Fibonacci number to be calculated)
Output: result (the nth Fibonacci number)

i = 3;
x = 1;
y = 1;
while (i <= n)
{
  temp = x+y;
  x = y;
  y = temp;
  i ++;
}
result = y;
```

The following controller and datapath can be used to implement the code shown above. A block diagram of the circuit is shown below:



The circuit has 4 inputs: *go*, *n (32-bit)*, and *clock/reset* (not shown). There are also 2 outputs: *done* and *result (32-bit)*. The circuit should initially wait until go is asserted (active high), and then read in input *n*, control the datapath as needed to generate the nth fibonacci number, and then store the result in the result register that is connected to the output *result*. After a result has been generated, *done* should be asserted (active high) until *go* is set back to 0 and then reasserted (i.e., until the circuit starts again). The circuit should not start calculating another number until go is reset to 0 and then set to 1.