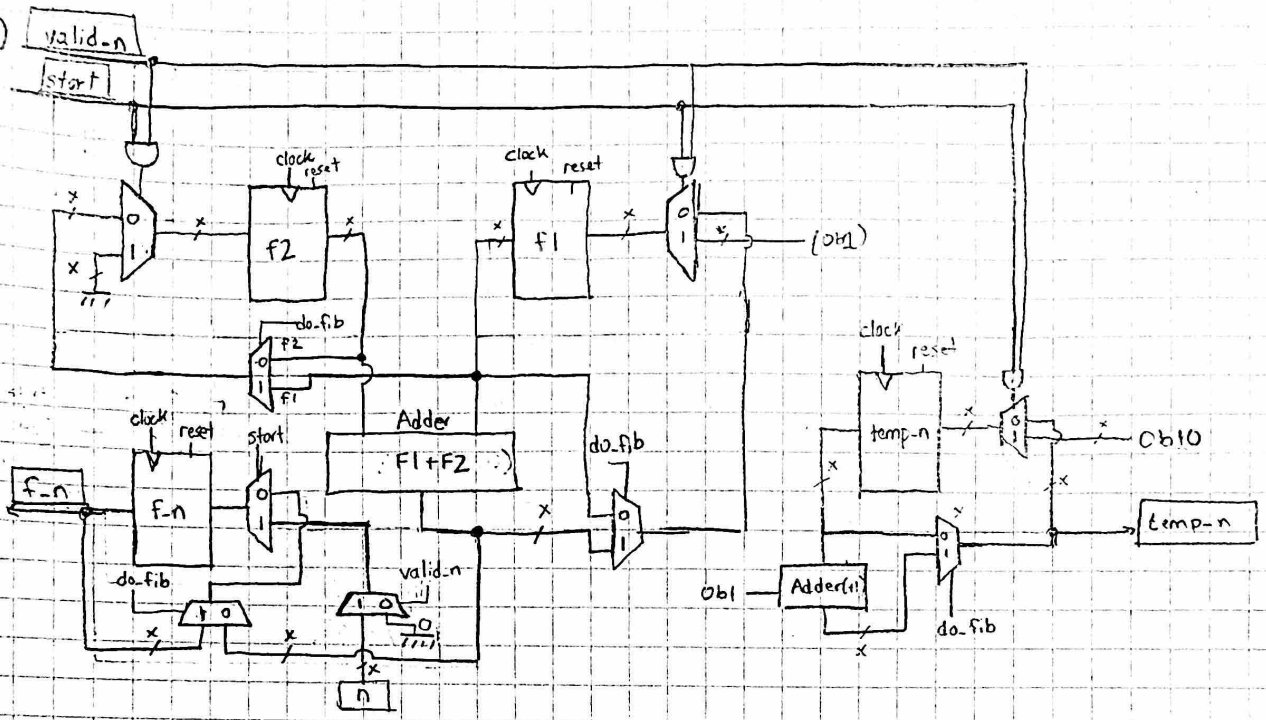
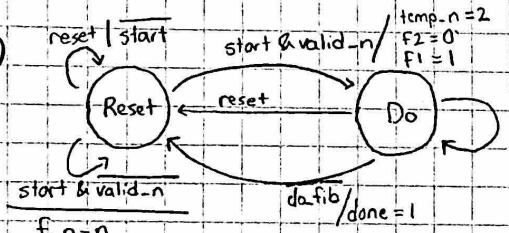
2) a) The system will have inputs: and outputs:
- Clock, Reset, n, start
- f_n, done

b)



c)



The system operates by iteratively computing the fibonnaci sequence from $F_0 \rightarrow F_N$. The control signals used in this DP/FSM are (valid_n): which simply looks if (n) is ≥2, indicating a valid input, and do_fib which controls when to stop looping, by comparing if temp_n, the current loop number, is less than or equal to n, the total number of loops. this is initialized to 2 since F_0 and F_1 are seeded. start is also used as a one-clock start pulse for the computation.

3

```
module testbench();
  parameter CLK_PRD = 100; parameter HOLD_TIME = CLK_PRD*0.3; parameter MAX_X_VAL = 15; parameter NUM_TRANS = 20;
  logic clock, reset, valid, ready; logic [31:0] x_factorial; rand logic [31:0] data;
  fifo_interface dut(.*);
  initial begin clock <= 0; forever #(CLK_PRD/2) clock = ~clock; end
  initial begin: sim
    reset = 0; valid = 0; data = 0; #CLK_PRD;
    @(posedge clock); #HOLD_TIME; reset = 1; repeat(2) #CLK_PRD;
    $monitor(data, x_factorial); repeat(NUM_TRANS) begin
      valid = 1; data.randomize() with {data <= MAX_X_VAL;}; #CLK_PRD;
      while (x_factorial == 0) begin
        valid = 0; #CLK_PRD;
      end
    $finish;
  end: sim
endmodule: testbench
```