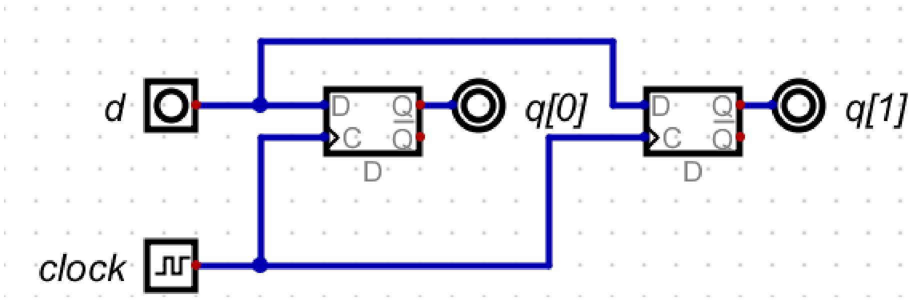
2.1 It give the same result because both code describe a module that function the same.

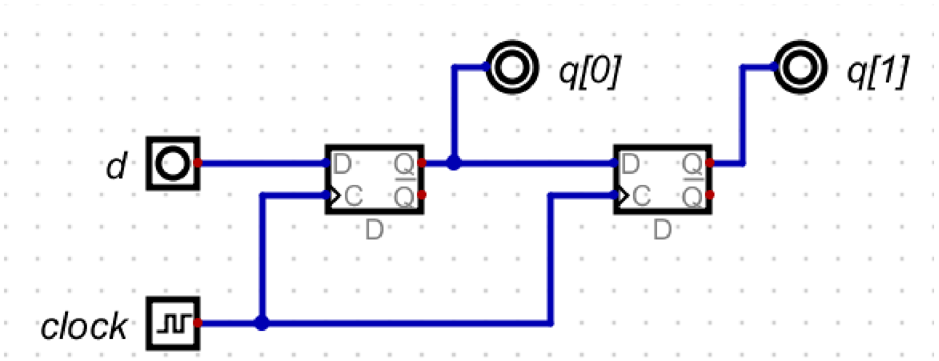
4.1 ShiftA both q[0] and q[1] will have the value of d when the positive edge of clock happen but shiftB q[0] will have the value of d and q[1] will also have the value of q[0] when the positive edge of clock happen

5.1 :

ShiftA :



ShiftB :



5.2: In the always block the blocking assignment (=) is executed in series so the result we get will be depending on the sequence of the instruction but the non-blocking assignment (<=) is executed in parallel so that the value that we assign to our register will be assigned to the register simultaneously at the end of the processing.

5.3: Yes by using Parameter

```
module N_bit_Shift
#(
    parameter N = 2
)
(
    output reg [N-1:0] q,
    input wire clock,
    input wire d
);
always @(posedge clock) begin
    q[N-1:1] <= q[N-2:0];
    q[0] <= d;
end
endmodule
```

We can access this module by

```
N_bit_Shift #(.N(4))u1(q,clock,d);
```

this is a 4 bit shif register