Lecture 2 - State Machine Design with VHDL

Exercise 1: Which signal in the above diagrams indicates the current state?

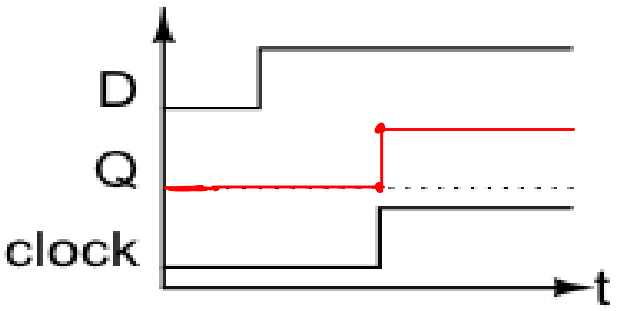
output of the memory.

Exercise 2: How may possible states are there for a CPU containing 10,000 flip-flops?

$2^{10,000}$ (very large)

$2^{10,000} \approx 2^{3.5 \times 3000} \approx 10^{3000}$

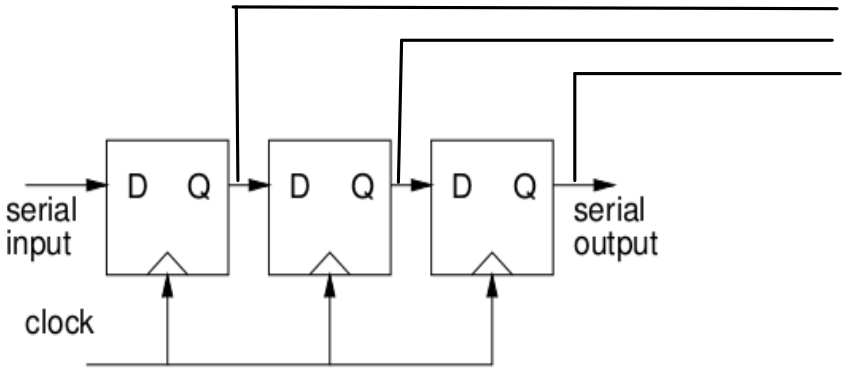
Exercise 3: Fill in the waveform for the `Q` signal in the diagram above.



Exercise 4: What would be another name for a 1-bit register?

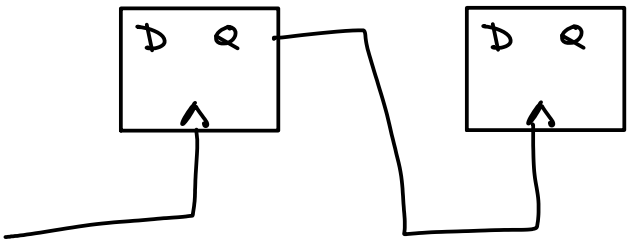
D flip-flop

Exercise 5: Add the parallel outputs to the shift register diagram.



Exercise 6: Is a ripple counter a synchronous logic circuit?

an asynchronous design.



Exercise 7: If we used 8-bits of state information, how many states could be represented? What if we used 8 bits of state but added the condition that exactly one bit had to be set at any given time (a so-called "one-hot encoding")?

8 bits: 2**8 = 256

8 bits in "one-hot": 8

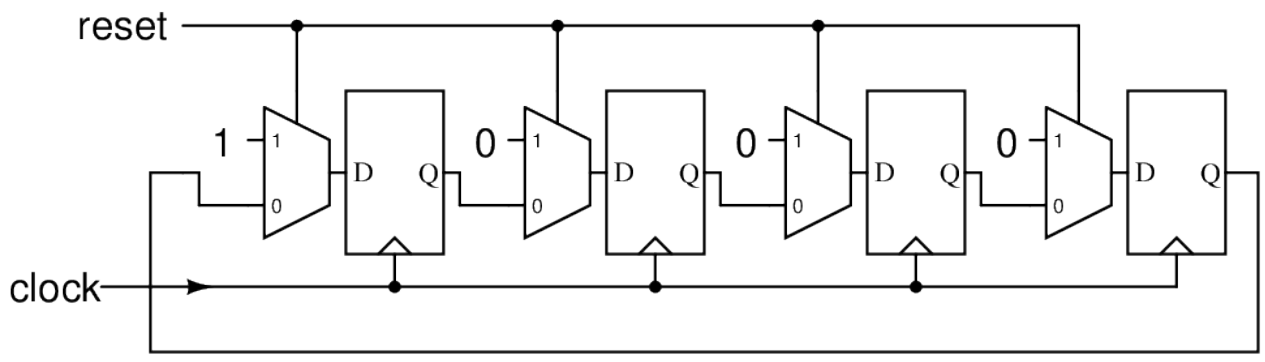
binary

0 0
0 1
1 0
1 1

01 — one-hot
10

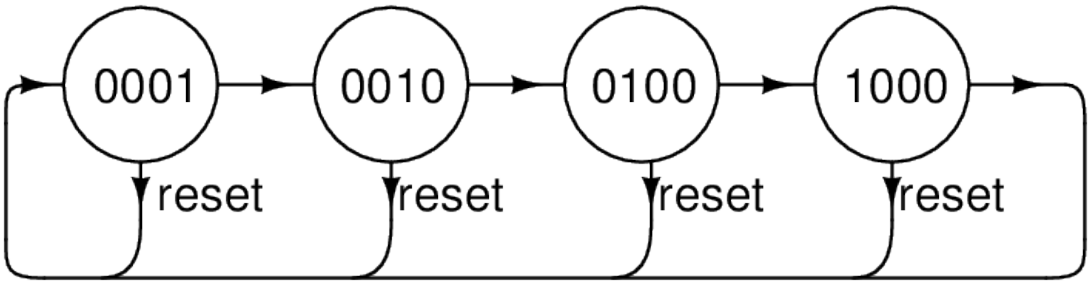
Exercise 8: Write the tabular description and draw the schematic of a resetable 2-bit counter with demultiplexed outputs (only one of the four outputs is true at any time). You can assume the counter will always be reset before being used. How does this counter compare to the previous one in terms of number of flip-flops and the complexity of the combinational logic?

current state	R	next state
1000	0	0100
0100	0	0010
0010	0	0001
0001	0	1000
XXXX	1	1000



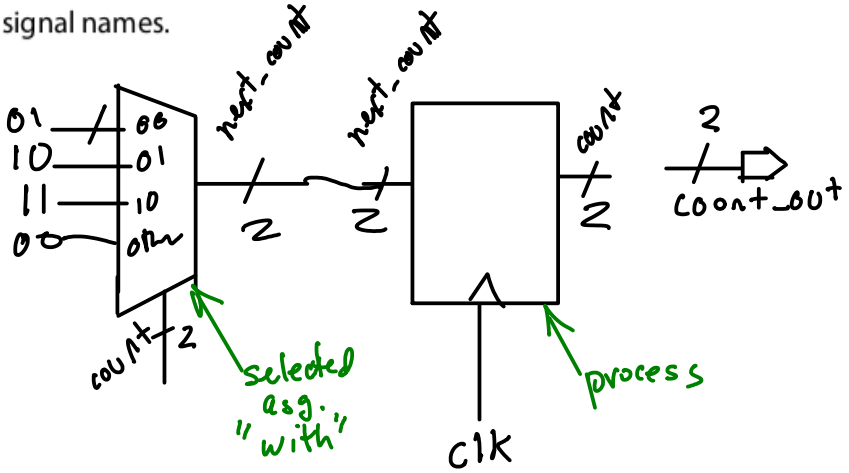
4 FFs instead of 2
simpler? combinational logic

Exercise 9: Draw the state transition diagram.

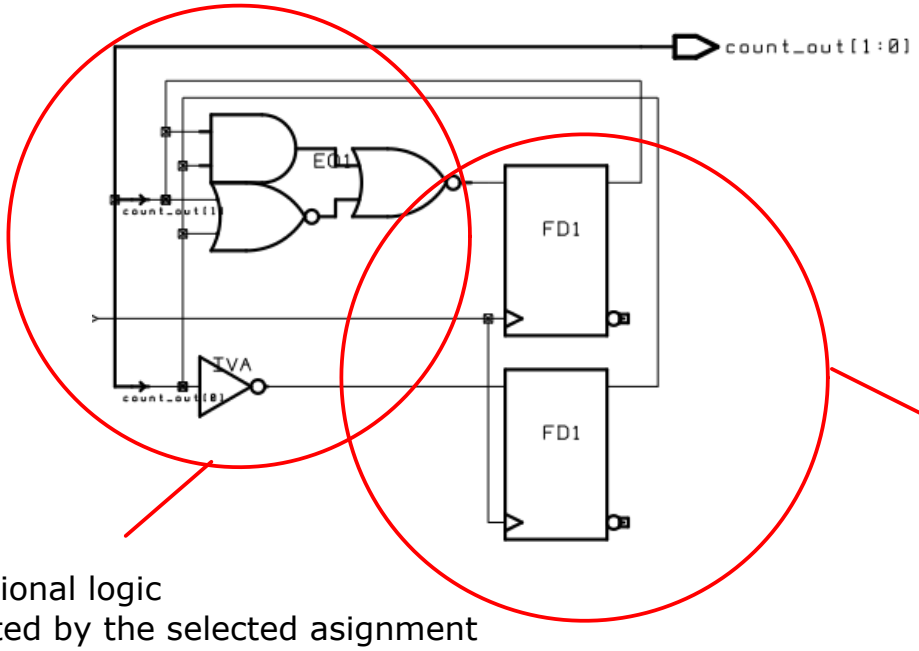


Exercise 10: Draw the block diagram corresponding to this VHDL description. Use a multiplexer for the selected assignment and and a register for the state variables. Label the connections, inputs and outputs with the corresponding VHDL signal names.

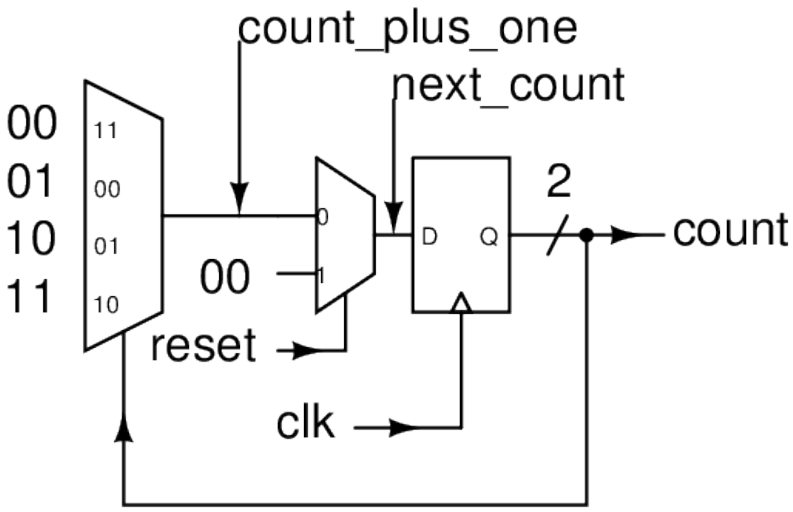
```
with count select next_count <=
    "01" when "00",
    "10" when "01",
    "11" when "10",
    "00" when others ;

-- combinational logic for output
count_out <= count ;

-- sequential logic
process(clk)
begin
   if clk'event and clk = '1' then
      count <= next_count ;
   end if ;
end process ;
```



Exercise 11: Identify the components in the schematic that were created ("*instantiated*") the different parts of the VHDL code.



Exercise 12: Modify the diagram of the 2-bit counter by adding a (synchronous) reset input and a 2-input, 2-bit multiplexer. Draw a block diagram showing the modified circuit. Label the connections and i/o with appropriate signal names. Write the corresponding VHDL description.



changes:

```
    clk, reset : in bit ;


-- combinational logic for next state
with count select count_plus_one <=
   "01" when "00",
   "10" when "01",
   "11" when "10",
   "00" when others ;


 -- reset logic
 with reset select next_count <=
       count_plus_one when '1',
       "00" when others ;
```