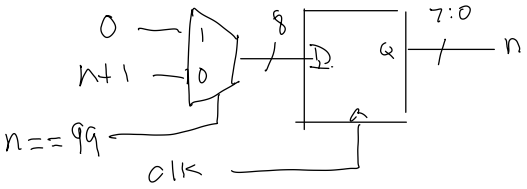
Sequential Logic and State Machines

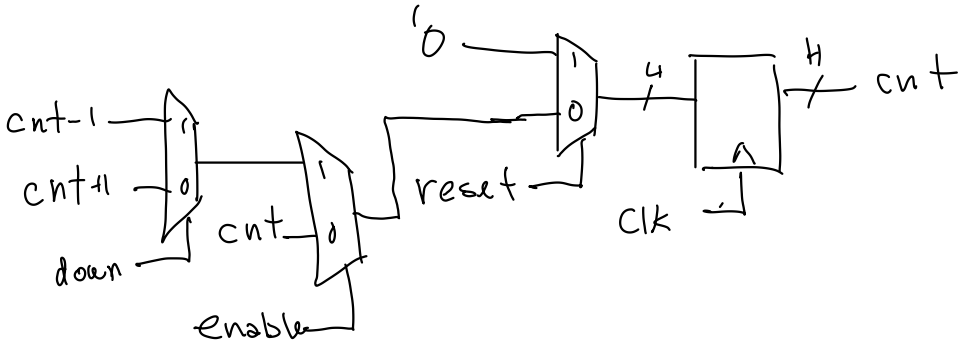
Exercise 1: Write the Verilog for an 8-bit counter named '**n**'. Modify it so that the register value is set to 0 each time it reaches 99. Draw the corresponding block diagram. Identify the next-state combinational logic.

```
logic [7:0] n;
always_ff @(posedge clk)
    n <= n==99 ? 0 : n+1 ;
```

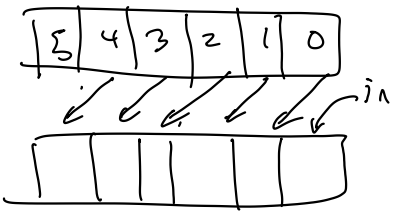


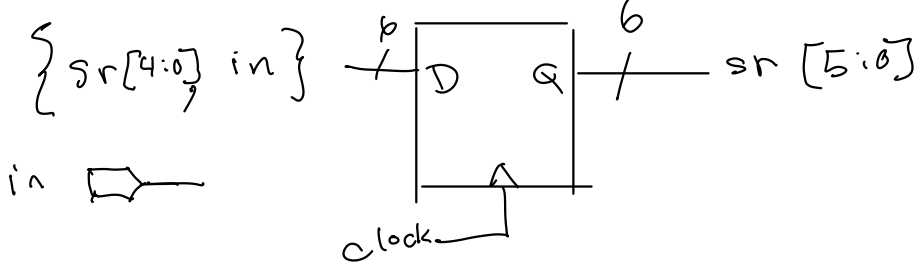
Exercise 2: Write a 4-bit counter named '**cnt**' with **reset**, **enable**, and **down** inputs. **reset** should have priority over **enable**. Draw the block diagram.

```
logic [3:0] cnt;
always_ff @(posedge clk)
    cnt <= reset ? '0 :
           enable ? (down ? cnt-1 : cnt+1) : cnt ;
```



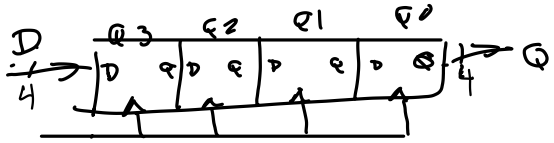
Exercise 3: The example above is an N-bit shift register that shifts the bits right. Draw a block diagram and write the Verilog for a 6-bit shift register that shifts left.

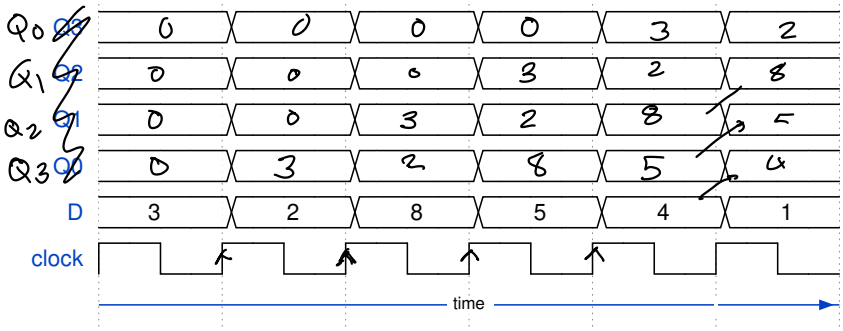





```
logic sr[5:0];
always_ff @(posedge clock)
    sr <= {sr[4:0], in};
```

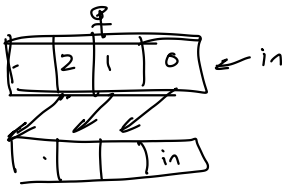
Exercise 4:

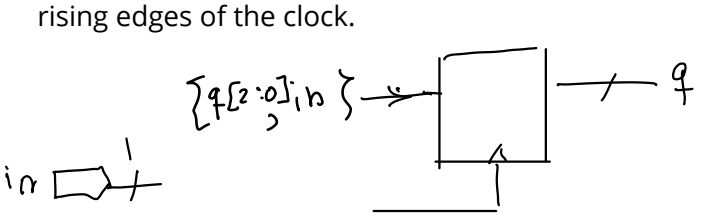


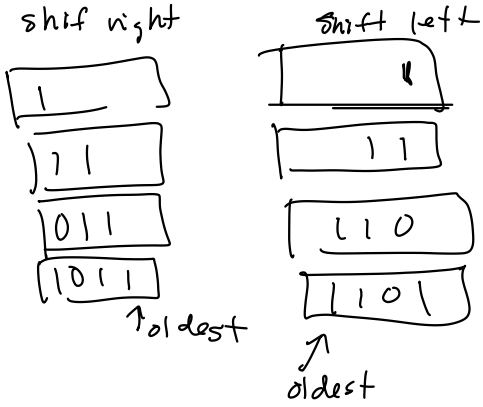


Fill in the diagram above for a 4-bit ($N = 4$) right-shift shift register. Assume the initial value is zero. Which bit is the oldest (first) value in the waveform? Which bit of the shift register holds the oldest value?

Exercise 5: Draw a block diagram and write the Verilog for a circuit that sets an output named **`detect`** high when the sequence of values 1, 1, 0, 1 has appeared on an input named **`in`** on successive rising edges of the clock.








```
module srdetect (input logic in, clk,
                 output logic detect);
  logic [3:0] q;
  always_ff @(posedge clk)
    q <= {q[2:0], in};
  assign detect = q == 4'b1101;
endmodule
```

Exercise 6: What value of N would result in a 20 ms delay if the clock frequency is 50 MHz? How many bits are needed for this timer's register?

$$T = \frac{1}{f} = \frac{1}{50 \times 10^6} = 20 \times 10^{-9} \quad (20\text{ns})$$

$$20\text{ms} = 20 \times 10^{-3}$$

$$NT = \text{delay} \qquad N = \frac{\text{delay}}{T} = \frac{20 \times 10^{-3}}{20 \times 10^{-9}} = 1 \times 10^6$$

$$N = 1 \times 10^6 = 1\,000\,000$$

$$N-1 = 999\,999 \ldots 0$$

with b bits can represents from 0 to $2^b - 1$

test: $b = 10 \quad 0 \ldots 2^{10} - 1 = 1023$

$b = 20 \quad 0 \ldots 2^{20} - 1 \simeq 10^6$

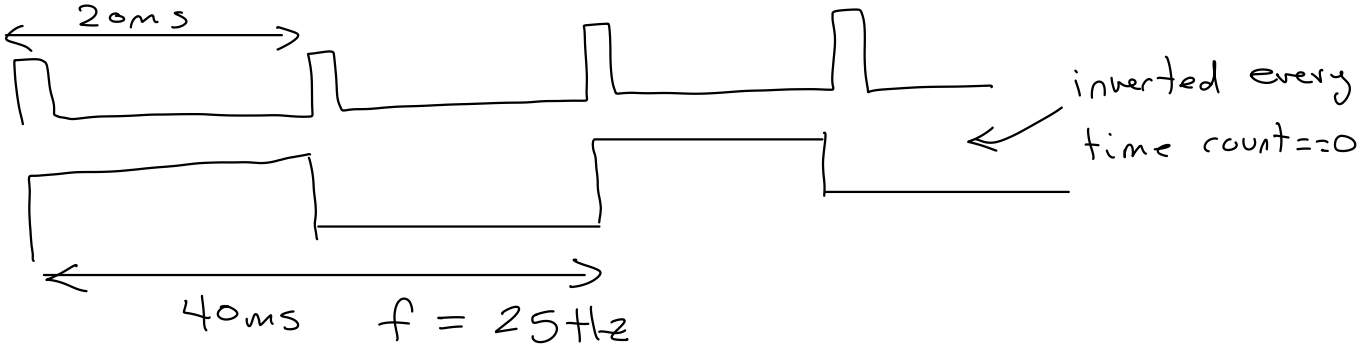
$$\text{max} = 2^b - 1 \qquad b = \log_2(\text{max} + 1)$$

Exercise 7: Assume the timer above is reset to $N-1$ each time it reaches 0. What is the time between each time the count reaches zero? What is this frequency? For how long does the register have the value 0? What are the period and frequency of a signal that is inverted each time the count reaches 0?

reaches 0 every N clock cycles = NT seconds
= 20ms.

$$\text{frequency} = \frac{1}{20\text{ms}} = 50\text{ Hz}$$

value is 0 for 1 clock period



Exercise 8: For the state machine described above, if the current state is 01, what will be the next state? When will the state change? What is the output in state 00? In state 01? In state 10?

00 → 0
01 → 0
10 → 1

01 → 10 (if reset not asserted).

state changes on rising edge of clock.

Exercise 9: What will be the next state if the state is 00 and the **reset** input is 1? If the state is 00 and the **reset** input is 0? When does the state change? When does **reset** affect the output?

00 → 00 → use first row
00 → 01 → use third row

state changes on ↑ clock

reset affects O/P is if is 1.

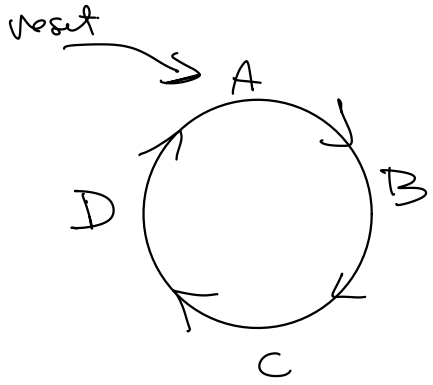
Exercise 10: Write the above state transition and output tables using state names A, B, C, and D.

e.g.

A = 00
B = 01
C = 10
D = 11

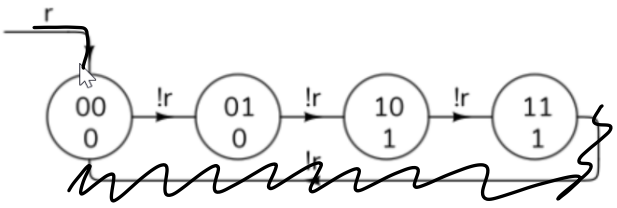
state	reset	next state
xx	1	00
11	0	00
n	0	$n+1$

state	reset	next
X	1	A
D	0	A
A	0	B
B	0	C
C	0	D



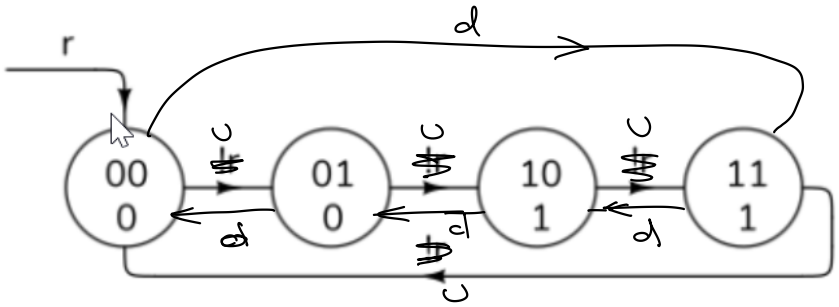
Exercise 11: Modify the diagram so the stat and stops.

remove transition from 11 to 00:



r: reset==1

Exercise 12: Add a **down** input that cause the values to count down.



r: reset==1

d: !reset && down

U: !reset && !down

d	r	next state
1	0	down
0	0	up
1	1	00
0	1	00
X	1	

Exercise 13: How many bits need to be considered to detect a specific state when a binary encoding is used? How many need to be considered if a one-hot encoding is used?

binary: need to consider all bits

one-hot: one bit (if it's 1, then it's in that state).

Exercise 14: If we used 8-bits of state information, how many states could be represented? What if we used 8 bits of state but used a "one-hot" encoding?

binary: $2^8 = 256$

one-hot: 8