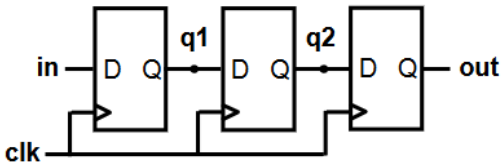
Report Task 1

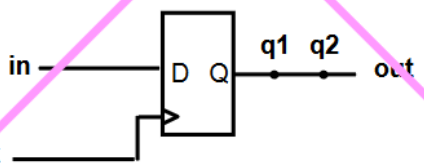
```
always @ (posedge clk)
begin
   q1 <= in;
   q2 <= q1;
   out <= q2;
end
```

"At each rising clock edge, *q1*, *q2*, and *out* simultaneously receive the old values of *in*, *q1*, and *q2*."




```
always @ (posedge clk)
begin
   q1 = in;
   q2 = q1;
   out = q2;
end
```

"At each rising clock edge, *q1* = *in*. After that, *q2* = *q1* = *in*; After that, *out* = *q2* = *q1* = *in*; Finally *out* = *in*."

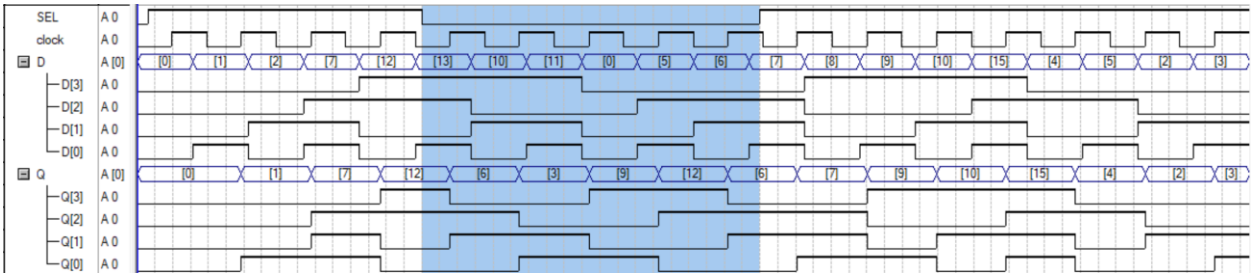


Report Task 2

Verilog Code:

```
module exp4task2(D, SEL, clock, Q);

parameter n = 4;

input [n-1:0] D;

input SEL, clock;

output reg [n-1:0] Q;


always @(posedge clock)

if (SEL) Q <= D;

else Q[n-1:0] <= {Q[0], Q[n-1:1]};

endmodule
```

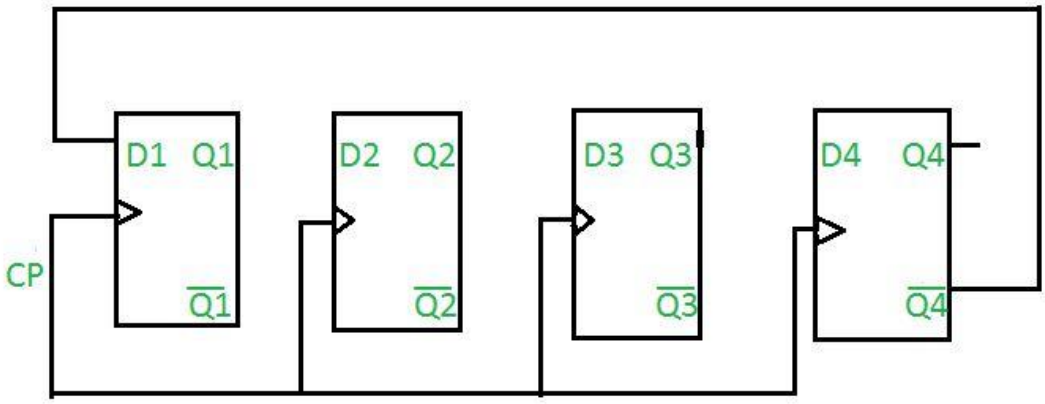
Vector Waveform:



When SEL=1, the register loads data at the positive edge of the clock. When SEL=0, the register right shifts the data one bit at the positive edge of the clock. The highlighted part indicates the shifting operation of the register.

Report Task 3

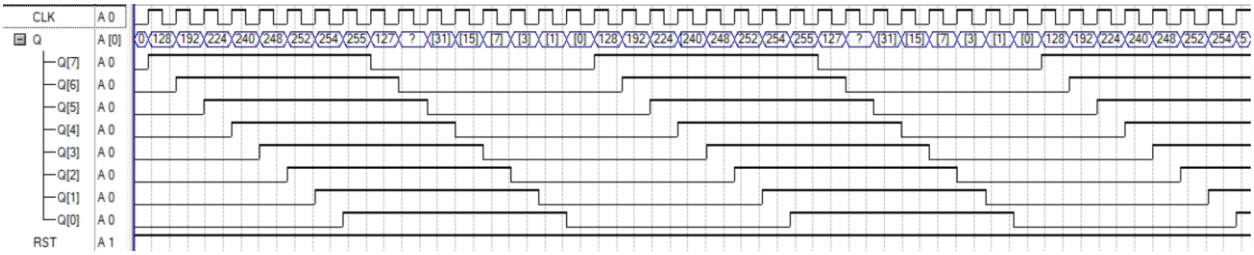
Diagram:



Truth Table:

CP	Q1	Q2	Q3	Q4
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0

Vector Waveform:

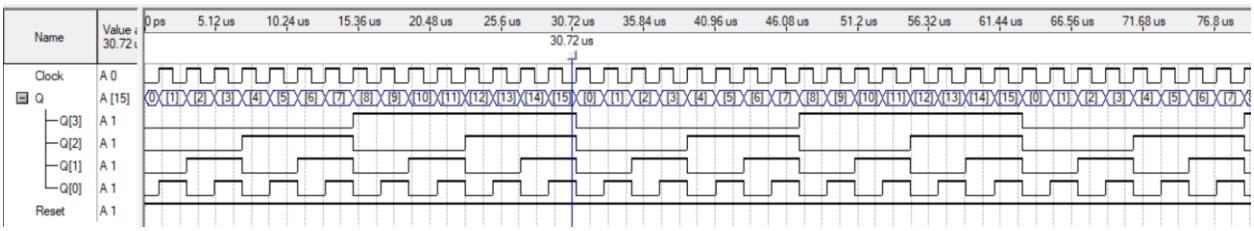


Report Task 4

Verilog Code:

```
module task4exp4(Reset, Clock, Q);

input Reset, Clock;

output reg [3:0] Q;


always @(posedge Clock)

begin

if (Reset == 0) Q <= 0;

else Q <= Q + 1;

end

endmodule
```

Vector Waveform:



The main clock pulse has a period of 2us and a frequency of 0.5MHz. The frequency divider is executed using an up counter. The MSB Q[3] has a clock period of 31.111us and a frequency of 0.032143 MHz which is 1/16 times of the main clock frequency.