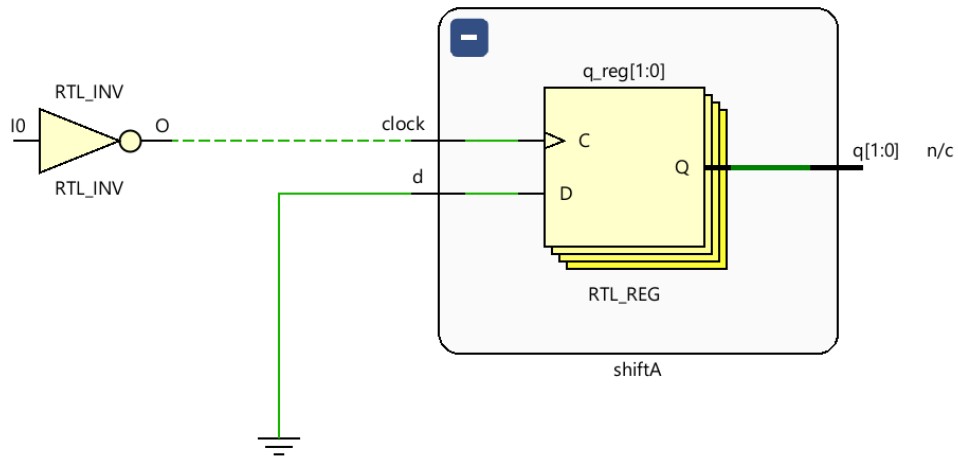
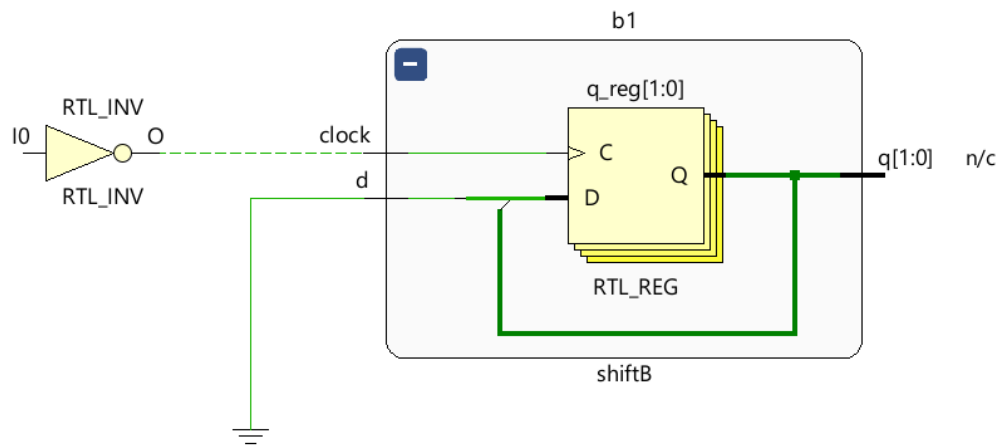


5.1) Please draw a schematic representing the logical blocks of both shiftA and shiftB in exercise 4.

shiftA:



shiftB:



5.2) What is the difference between blocking and non-blocking assignments?

Blocking assignment:

- ลำดับในการ assign ค่ามีผล เนื่องจาก Statement แรกจะทำงานก่อน แล้วค่อยทำงาน statement ต่อไป
- เวลาที่มี always หลายตัว อาจเกิด race condition
- นิยมใช้กับการทำ Combinational Logic

Non-blocking assignment:

- ลำดับในการ assign ค่าไม่มีผล เนื่องจาก Statement จะทำงานแบบ concurrent
- ลดปัญหาการเกิด race condition เนื่องจาก statement ด้านขวาจะถูก evaluate ก่อนเสมอ
- นิยมใช้กับการทำ Sequential Logic

5.3) Is it possible to apply parameters to the design in exercise 4 to create shiftRegister with any number of bits? If Yes, please explain how.

ผมใช้ parameter ชื่อ n ในการบอกว่า q จะมีทั้งหมดกี่ bits และใช้การมัลติเพล็กซ์สายไฟตั้งแต่ตัวที่รองสุดท้ายจนถึงแรกสุด แล้วเพิ่ม D ลงไปท้ายสุด แล้วนำไปแทนใน q เลย โดยใช้แบบ non-blocking assignment ทำให้สามารถสร้าง shifter ที่รองรับจำนวน n bits ได้

```
module nBitsShift #(parameter n=8)
(q,clock,d);
output [n-1:0] q;
input clock, d;
reg [n-1:0] q;
always @ (posedge clock)
begin
    q <= {q[n-2:0], d};
end
endmodule
```

```
nBitsShift #(16) shifter00 (q,clock,d);
```