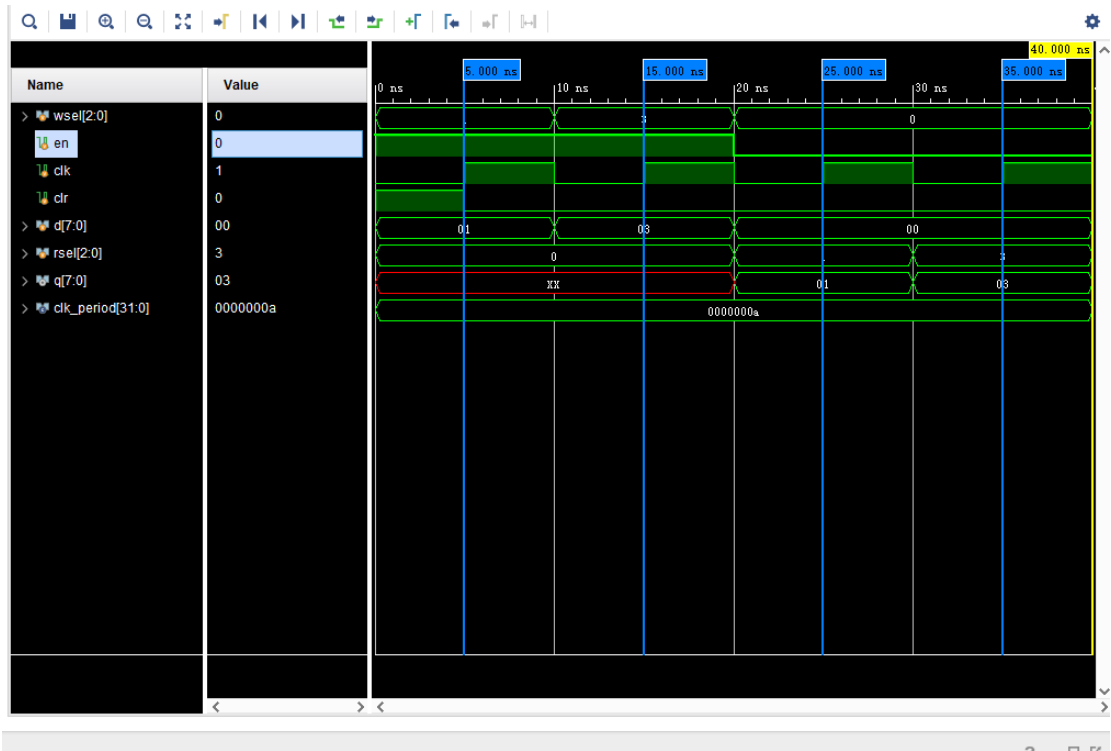


下面是 8 位寄存器的波形图：



波形分析：

clk 为时钟信号，clr 为复位信号，en 为输入使能，wsel 为输入选择，rsel 为输出选择，d 为寄存器输入，q 为寄存器输出。

(1) 第一个时钟上升沿 5ns 时，clr 为 0，en 为 1，处于写入状态，wsel 为 1，rsel 为 0，d 为 1，向寄存器的编号 1 的触发器中写入 d，不输出，q 为 x 符合预期；

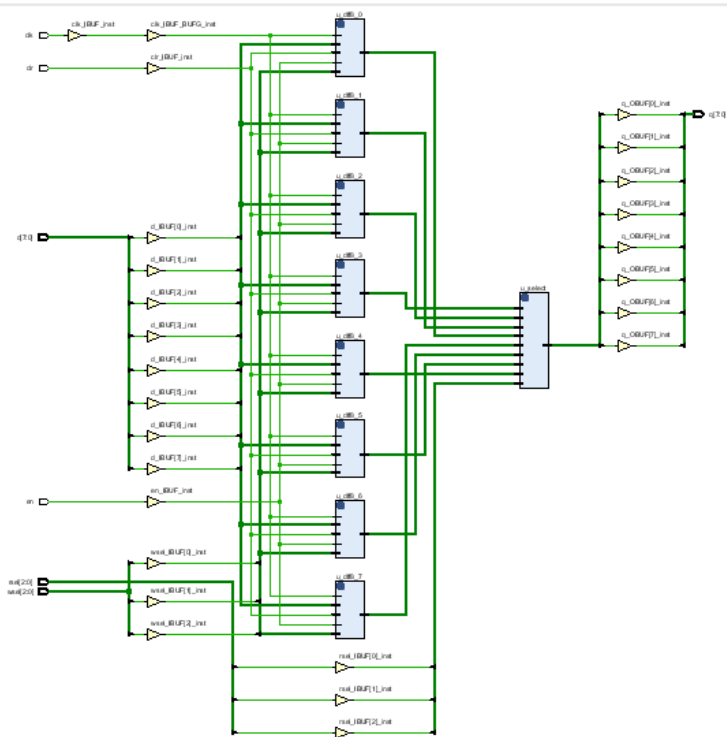
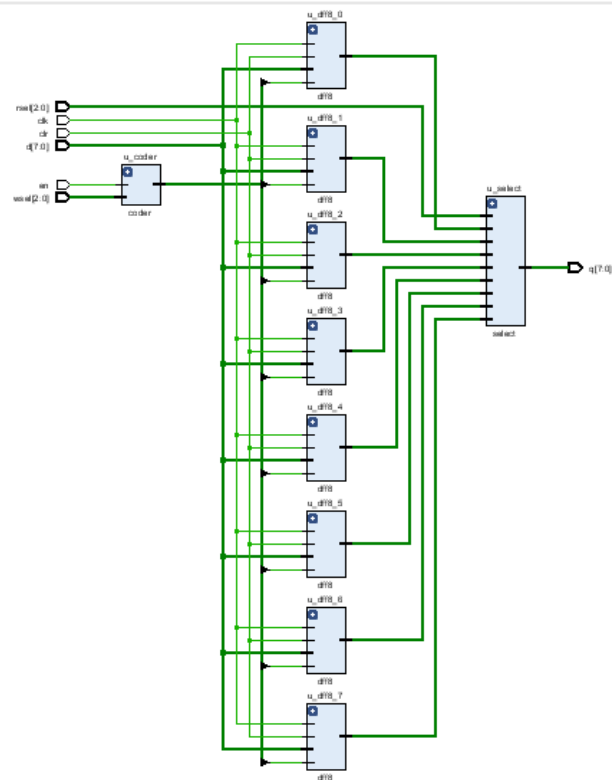
(2) 第二个时钟上升沿 15ns 时，clr 为 0，en 为 1，处于写入状态，wsel 为 2，rsel 为 0，d 为 3，向寄存器的编号 2 的触发器中写入 d，不输出，q 为 x 符合预期；

(3) 25ns 时，clr 为 0，en 为 0，无法写入，wsel 为 0，rsel 为 1，d 为 0，此时读取寄存器的编号 1 的触发器中的数据，输出 q 为 1 符合预期；

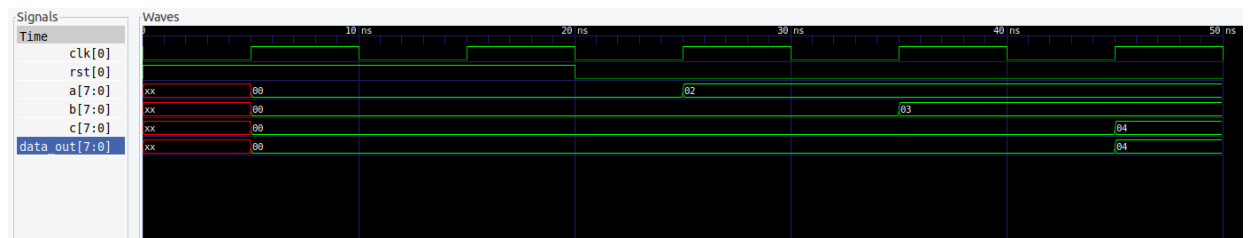
(4) 30ns 时，clr 为 0，en 为 0，无法写入，wsel 为 0，rsel 为 2，d 为 0，此时读取寄存器的编号 2 的触发器中的数据，输出 q 为 3 符合预期。

综上所述，8 位寄存器模块正确实现了功能。

下面分别为 8 位寄存器的 RTL Analysis schematic 和 Synthesis schematic：



下面为课后作业的波形：



波形分析：

在 25ns 时 a 变为 2，但 b 在下一个时钟上升沿 35ns 时变为 3，c 在再下一个时钟上升沿 45ns 时变为 4。

从而得出结论：

每个 always 块被触发时，reg 变量的赋值计算用的是所涉及到的变量触发时刻的值，但并不受同时触发的其他 always 计算结果影响。