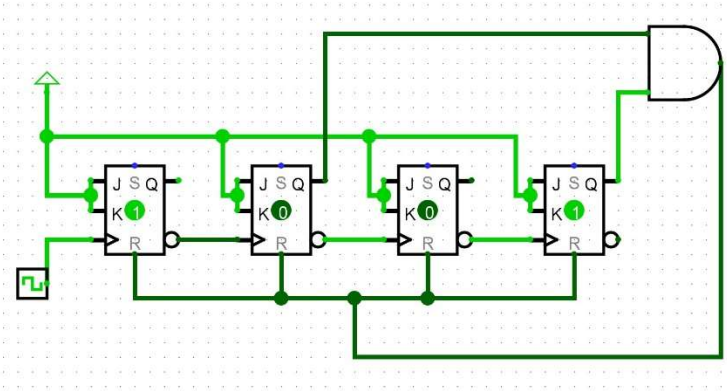
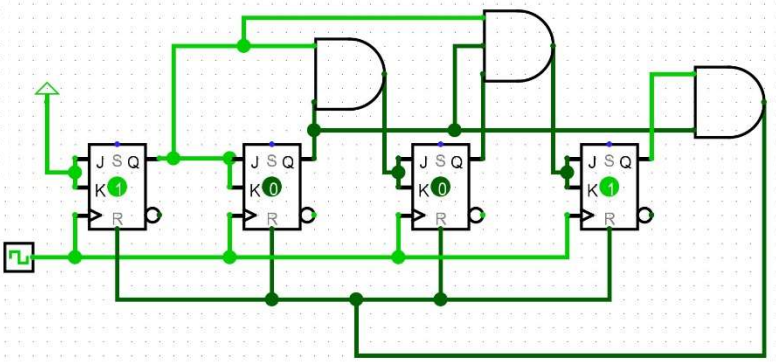
**Problem A: Mod-10 synchronous JK counter**

Construct a mod-10 (counts up to decimal 9 / binary 1001, then loops back to 0) counter) using JK flip flops. ~~**It must be synchronous – no glitch states!**~~ Use JK in "toggle" mode. The below configuration has been provided for you as a starting point.




**synchronous ->**

**<- asynchronous**




**Problem B: Custom synchronous 3-bit counter**

Given this state table, construct a **synchronous custom counter** using **D-type flip flops**. Hint: Construct Karnaugh maps by filling in where $D_c$, $D_b$, and $D_a$ are 0, 1 or x respectively.

Note: None of the $t_n$ states of the counter have a $t_{n+1}$ state of 010 or 110. However, if the counter were to ever enter those states (say, a random glitch happened), it should return to the sequence and not get stuck in those states.

| $t_n$ | | | | | | $t_{n+1}$ | | |
|---|---|---|---|---|---|---|---|---|
| $Q_c$ | $Q_b$ | $Q_a$ | $D_c$ | $D_b$ | $D_a$ | $Q_c$ | $Q_b$ | $Q_a$ |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | x | x | x | x | x | x |
| 1 | 1 | 0 | x | x | x | x | x | x |

Dc

| Qc\QbQa | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | x |
| 1 | 1 | 0 | 0 | x |

**!QcQa + Qc!Qa**

**Qc XOR Qa**

Db

| Qc\QbQa | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | x |
| 1 | 1 | 0 | 0 | x |

!Qb!Qa + Qb!Qa

**= !Qa**

Da

| Qc\QbQa | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | x |
| 1 | 1 | 1 | 0 | x |

**!Qa + !QcQb + Qc!Qb**

**!Qa + Qc XOR Qb**