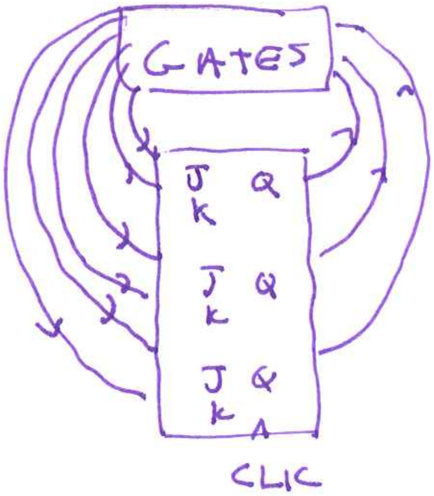
Using JKFF to "solve" state diagram problems

↳ ie find circuit that follows diagram




Basic idea same as DFF circuits:
GATES produce the future from present

Advantage: lots of Xs for JK make
GATES a simpler circuit

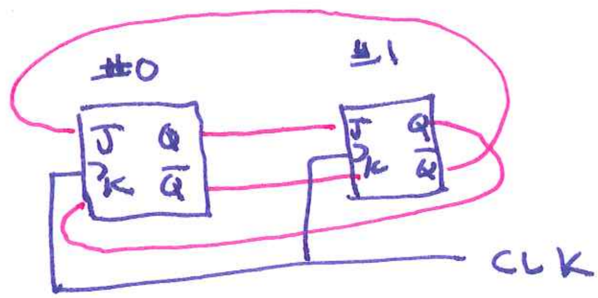
transitions:

| | J | K |
|---|---|---|
| 0→0 | 0 | X |
| 0→1 | 1 | X |
| 1→0 | X | 1 |
| 1→1 | X | 0 |

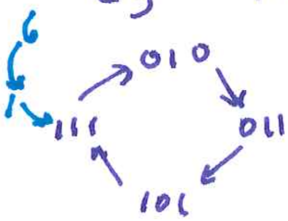
Eg – Gray Counter: 00 → 01 → 11 → 10 (→ 00)

| $Q_1$ | $Q_0$ | $J_1$ | $K_1$ | $J_0$ | $K_0$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | X | 1 | X |
| 0 | 1 | 1 | X | X | 0 |
| 1 | 1 | X | 0 | X | 1 |
| 1 | 0 | X | 1 | 0 | X |
| | | $Q_0$ | $\bar{Q}_0$ | $\bar{Q}_1$ | $Q_1$ |

often can find boolean expression just by looking




Eg – Prime number Counter



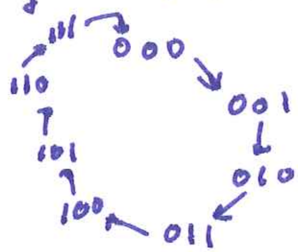

check: what happens to excluded state 6?

| | $J_2$ | $K_2$ | $J_1$ | $K_1$ | $J_0$ | $K_0$ |
|---|---|---|---|---|---|---|
| 110 | 0 | 1 | 1 | 1 | 1 | 1 |
| 001 | 1 | 0 | 1 | 1 | 1 | 0 |
| 111 | | | | | | |

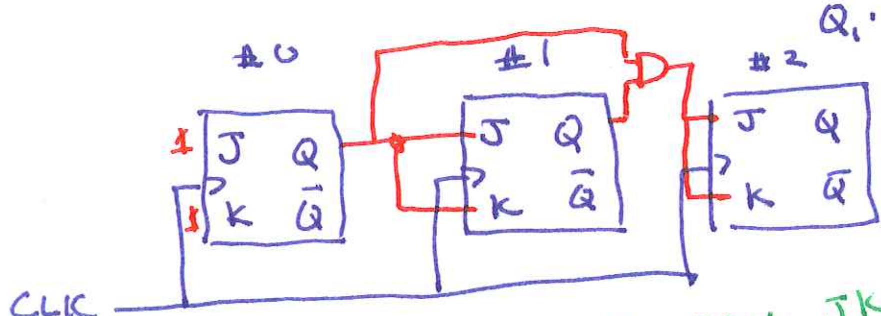
| $Q_2$ | $Q_1$ | $Q_0$ | $J_2$ | $K_2$ | $J_1$ | $K_1$ | $J_0$ | $K_0$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | X | X | 0 | 1 | X |
| 0 | 1 | 1 | 1 | X | X | 1 | X | 0 |
| 1 | 0 | 1 | X | 0 | 1 | X | X | 0 |
| 1 | 1 | 1 | X | 1 | X | 0 | X | 1 |
| | | | $Q_0$ | $Q_1$ | 1 | $Q_2 \oplus Q_0$ | 1 | $Q_1 \cdot Q_2$ |

found just by looking but could K-map or nonterm if required

Eg synchronous binary counter (3 bits)

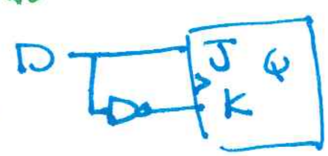



| $Q_2$ | $Q_1$ | $Q_0$ | $J_2$ | $K_2$ | $J_1$ | $K_1$ | $J_0$ | $K_0$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | X | 0 | X | 1 | X |
| 0 | 0 | 1 | 0 | X | 1 | X | X | 1 |
| 0 | 1 | 0 | 0 | X | X | 0 | 1 | X |
| 0 | 1 | 1 | 1 | X | X | 1 | X | 1 |
| 1 | 0 | 0 | X | 0 | 0 | X | 1 | X |
| 1 | 0 | 1 | X | 0 | 1 | X | X | 1 |
| 1 | 1 | 0 | X | 0 | X | 0 | 1 | X |
| 1 | 1 | 1 | X | 1 | X | 1 | X | 1 |
| | | | $Q_1 \cdot Q_0$ | $Q_1 \cdot Q_0$ | $Q_0$ | $Q_0$ | 1 | 1 |




Remark: JKFF as simple CPU; JK as "instruction"

Remark: make a DFF from JKFF




Some packaged functions

Counters – size (in bits) decade or binary — top count = 9; aka BCD counter

"ripple" = asynchronous or synchronous

up/down ; carry ; enables

clear, preset = load (synchronous or asynchronous)

Shift Register


size (in bits – are all bits available on pins or internal)

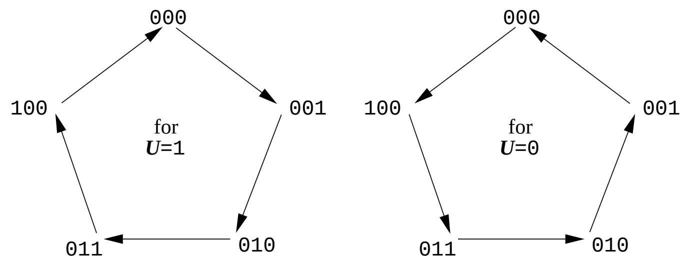
L R shifts

clear, preset = load (synchronous or asynchronous)

use: parallel ↔ serial

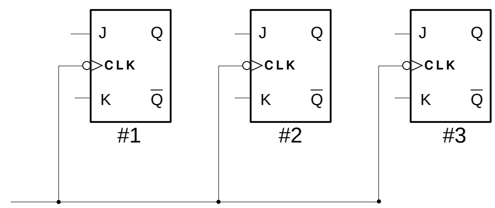
multiply by 2

37. Design a synchronous circuit built from three edge-triggered JKFFs that follows the below state diagram, where the three binary digits represent the values of $Q_1Q_2Q_3$:




(i.e., when control line $U$=1 the circuit counts up to 4 (i.e., mod-5); when control line $U$=0 the circuit counts down from 4). Your job is to determine the gate arrangement needed to make this cycle run, i.e., connecting the outputs of the three JKFFs: $Q_i$ (and/or $\overline{Q}_i$) and the $U$ line to the inputs of the three JKFFs: $J_iK_i$ possibly using the usual (AND, OR,...) gates.

| Transition: | $J$ | $K$ |
|---|---|---|
| $0 \longrightarrow 0$ | | |
| $0 \longrightarrow 1$ | | |
| $1 \longrightarrow 0$ | | |
| $1 \longrightarrow 1$ | | |




(a) Begin by considering the possible transitions of a single JKFF. What values of $JK$ allow a particular transition? Fill in the above table. Hint: in every row either $J$ or $K$ will be an $X$ for "don't care".

(b) Fill in the below table which displays the desired cycles

| $U$ | $Q_1$ | $Q_2$ | $Q_3$ | $J_1$ | $K_1$ | $J_2$ | $K_2$ | $J_3$ | $K_3$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | | | | | | |
| 1 | 0 | 0 | 1 | | | | | | |
| 1 | 0 | 1 | 0 | | | | | | |
| 1 | 0 | 1 | 1 | | | | | | |
| 1 | 1 | 0 | 0 | | | | | | |
| 0 | 1 | 0 | 0 | | | | | | |
| 0 | 0 | 1 | 1 | | | | | | |
| 0 | 0 | 1 | 0 | | | | | | |
| 0 | 0 | 0 | 1 | | | | | | |
| 0 | 0 | 0 | 0 | | | | | | |

Note that there are additional "don't care" possibilities in the full truth table.

(c) Maxterm the 0s for $J_3$ to produce a product-of-sums.

(d) Make a Karnaugh map of $J_2$ using the four logical variables $U, Q_1, Q_2, Q_3$. Don't forget to include the Xs (don't care) in your map! Circle appropriate groups and report the resulting simplest possible boolean expression for $J_2$. Please carefully label your Karnaugh maps so I know what each row and column of the map represents!