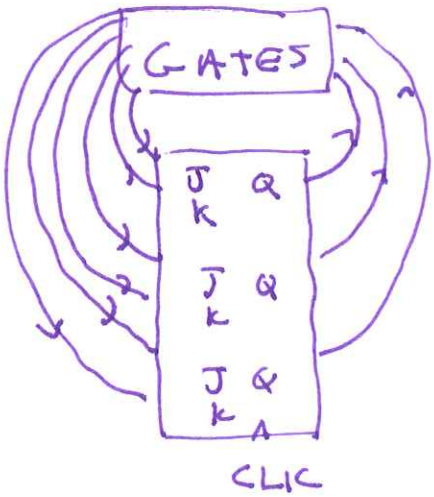
Using JKFF to "solve" state diagram problems

↳ 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 simple circuit

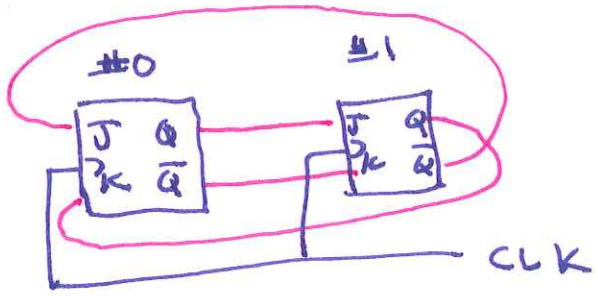
transitions:

	J	K
0→0	0	x
0→1	1	x
1→0	x	1
1→1	x	0

Eg - Gray Counter: $00 \to 01 \to 11 \to 10$

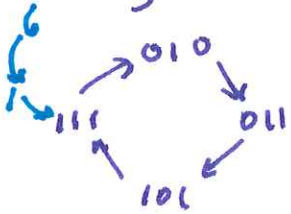
Q_1	Q_0	J_1	K_1	J_0	K_0
0	0	0	x	1	x
0	1	1	x	x	0
1	1	x	0	x	1
1	0	x	1	0	x

$J_1 = Q_0$, $K_1 = \bar{Q}_0$, $J_0 = \bar{Q}_1$, $K_0 = Q_1$



often can find Boolean expressions just by looking

Eg - Prime number Counter



check: what happens to excluded state 6?

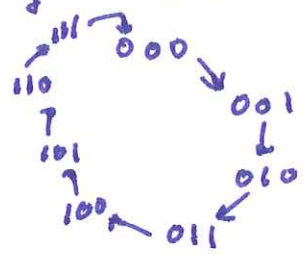
	J_2 K_2	J_1 K_1	J_0 K_0
110	0 1	1 1	1 1
001	1 0	1 1	1 0
111			

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

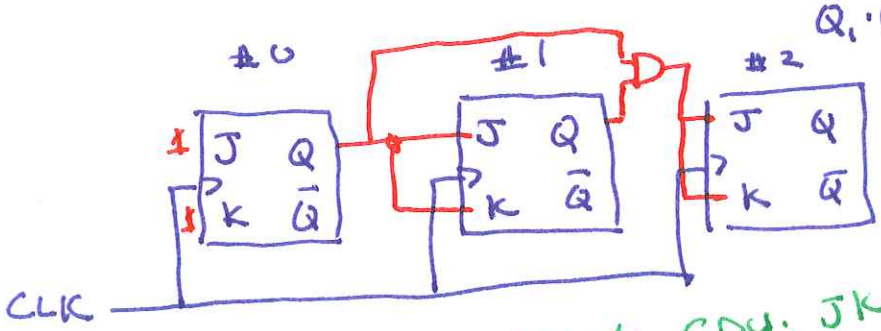
$J_2 = Q_0$, $K_2 = Q_1$, $J_1 = 1$, $K_1 = Q_2 \oplus Q_0$, $J_0 = 1$, $K_0 = Q_1 \cdot Q_2$

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

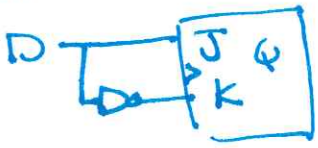
Eg synchronous binary counter (3 bits)



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
			↑	↑ ↑	↑ ↑ 1
			$Q_1 \cdot Q_0$	$Q_1 \cdot Q_0$	Q_0 Q_0 1



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



Remark: make a DFF from JKFF

Some packaged functions

Counters — size (in bits) decade a 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