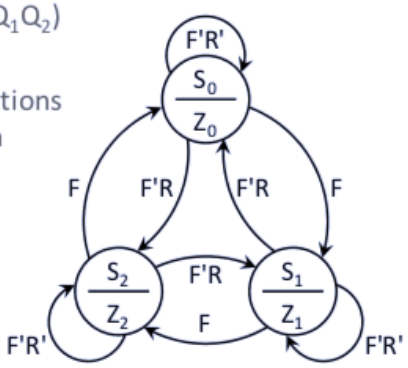
❑ Determine the flip flop input equations from the **next-state equations** using K-maps
- ➢ Always copy X's from next state maps onto input maps first
- ➢ Fill in the remaining squares with 0's

Type of FF	Input	Q = 0		Q = 1		Rules for forming input map from next state map	
		$Q^+ = 0$	$Q^+ = 1$	$Q^+ = 0$	$Q^+ = 1$	Q = 0 Half of Map	Q = 1 Half of Map
D	D	0	1	0	1	No change	No change
T	T	0	1	1	0	No change	Complement
S-R	S	0	1	0	X	No change	Replace 1's with X's
	R	X	0	1	0	Replace 0's with X's Replace 1's with 0's	Complement
J-K	J	0	1	X	X	No change	Fill in with X's
	K	X	X	1	0	Fill in with X's	Complement

Type	Q^+
D Flip-Flop	D
S-R Flip-Flop	S + R'Q
J-K Flip-Flop	JQ' + K'Q
T Flip-Flop	TQ' + T'Q
D-CE Flip-Flop	D(CE) + Q(CE)'

One-hot state assignment: One flip-flop for each state

- ➢ Example: 3 flip-flops for 3 states ($Q_0Q_1Q_2$)
  - $S_0 = 100$, $S_1 = 010$, $S_2 = 001$
- ➢ Write next-state and output (Z) equations directly by inspecting the state graph
  - $Q_0^+ = F'R'Q_0 + F'RQ_1 + FQ_2$
  - $Q_1^+ = F'R'Q_1 + F'RQ_2 + FQ_0$
  - $Q_2^+ = F'R'Q_2 + F'RQ_0 + FQ_1$
  - $Z = Z_0Q_0 + Z_1Q_1 + Z_2Q_2$



How about Don't Cares?

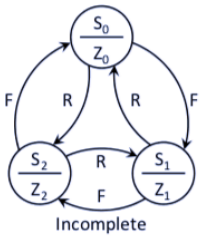
❑ A state table is incompletely specified if don't cares are present
- ➢ Certain sequences will never occur as inputs

Present State	Next State X=0	Next State X=1	Present Output X=0	Present Output X=1
A	B	D	0	---
B	C	D	---	0
C	B	A	1	0
D	C	D	---	---

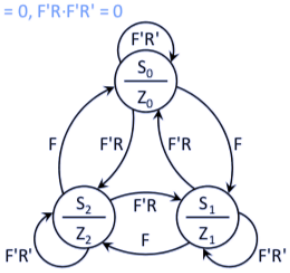
Present State	Next State X=0	Next State X=1	Present Output X=0	Present Output X=1
A	B	A	1	0
B	B	A	0	1
C	B	A	1	0
D	C	A	0	1

Completely Specified State Graph

❑ Properties
- ➢ **OR** together all input labels on arcs emanating from a state, the result can reduce to 1
  - Cover all conditions: F + F'R +F'R' = F + F' = 1
- ➢ **AND** together any pair of input labels on arcs emanating from a state, the result can reduce to 0
  - Only one arc is valid: F·F'R = 0, F·F'R' = 0, F'R·F'R' = 0



Incomplete



State Equivalence

❑ Definition
- ➢ N_1, N_2: sequential circuits (not necessarily different)
- ➢ X: a sequence of inputs of arbitrary length
- ➢ Then, state p in N_1 ≡ state q in N_2 if and only if $\lambda_1(p,X) = \lambda_2(q,X)$ for every possible input sequence X
  - λ: output
- ➢ Difficult to check the equivalence using this definition!
  - Infinite number of input sequences

❑ Theorem
- ➢ Two states p and q of a sequential circuit are equivalent if and only if for every single input X, the outputs are the same and the next states are equivalent, i.e., $\lambda(p,X) = \lambda(q,X)$ and $\delta(p,X) \equiv \delta(q,X)$
  - δ: next state
  - Note that the next state do not have to be equal, just equivalent

Static-1: 在 1 框框中間加上 bridge 的 1 框框
Static-0: 在 0 框框中間加上 bridge 的 0 框框

Multi-level 中,

- SOP: Static-1 一樣用發生在需要 bridge 時; Static-0 發生在 (AA')x 這種 complement 時。

essential hazard: caused by different delay in same input gate.

Shannon expansion theorem: 函數的降微，$f(A,B,C,D) = D' \cdot f(A,B,C,0) + D \cdot f(A,B,C,1)$

Latches:（閘）

- SR-Latch (Set to 1, Reset to 0): $Q^+ = S + R'Q$
- Gated-D Latch (Gated-Direct): $Q^+ = G'Q + GD$

Flip-Flop:（注意看是 rising-edge trigger 還是 falling-edge trigger）

- D-FF: D 是多少就是多少
- SR-FF: S=1 時 Set to 1; R=0 時 Reset to 0 (S=R=1 is illegal)
- JK-FF: J=1 時 Jump to 1; R=0 時 Clear to 0 (S=R=1 is toggle, 0 變 1, 1 變 0)
- T-FF: T=1 時 Toggle