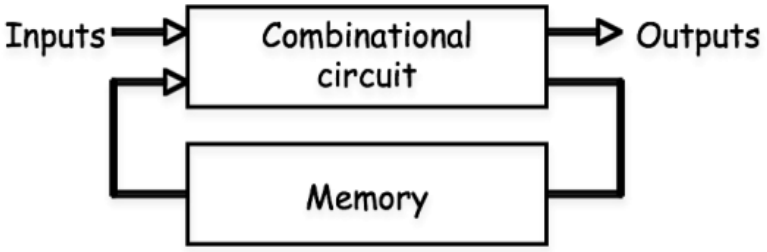
Ch.5 Synchronous Sequential Logic - part C

Sequential circuits analysis



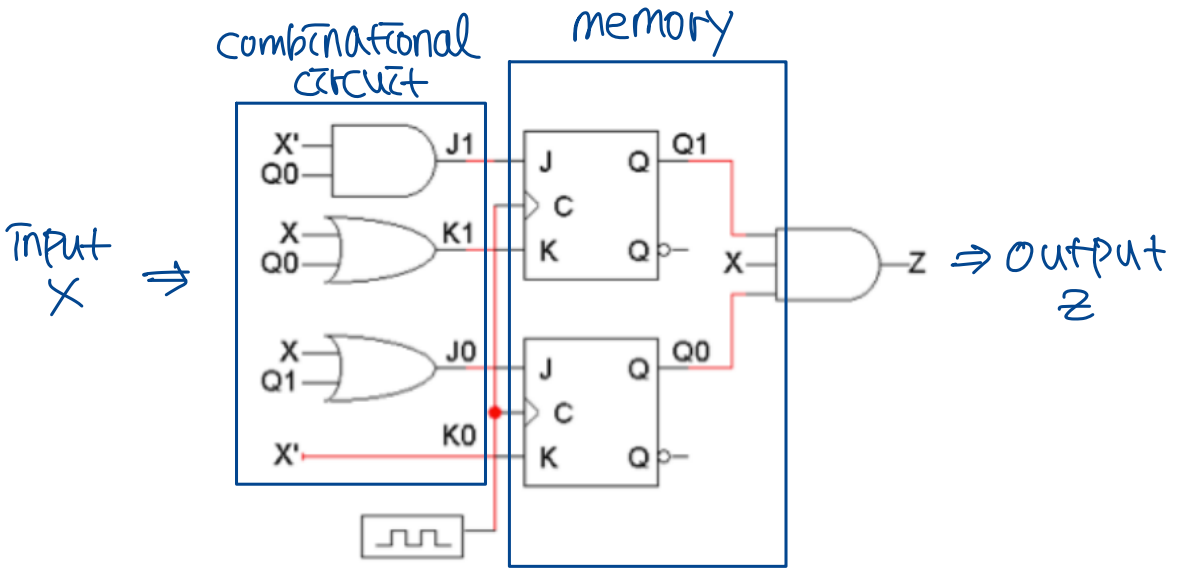
*순차회로 분석

1. **state tables**
   a. inputs
   b. outputs
   c. flip-flop state changes → current | next
2. **state diagrams** → 직관적인 state table (state tables의 equivalent way)
   - n개의 ff → 2^n 개의 node
   - m개의 input → 각 node(state)마다 2^m 개의 arrow

→ 최종적으로 state table 찾아내는 것이 목표

example

state table
① Output equation
② ff Input equation → combinational circuit
(J1K1, J0K0)
③ ff characteristic equation → ff next state
(Q1(n+1), Q0(n+1))
state diagram



- JK ff state : Q1Q0 → 00, 01, 10, 11 (ff의 개수에 따라 state 개수 달라짐. ff가 n개 일 때 총 2^n개)
- input : X
- output : Z

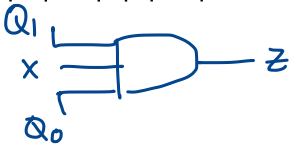
- **state table**
  1. 현재 state , input || 다음 state, output → 모든 경우의 수에 대해 다 작성

Present State Q_1	Present State Q_0	Inputs X	Next State Q_1	Next State Q_0	Outputs Z
0	0	0			0
0	0	1			0
0	1	0			0
0	1	1			0
1	0	0			0
1	0	1			0
1	1	0			0
1	1	1	1	1	1

J_1 K_1 J_0 K_0

$Q_0(t+1) = K_0' Q_0(n) + J_0 Q_0(n)'$

  2. output equation 찾기
     a. and gate → left가 다 1이어야 z가 1
     b. Z = Q1Q0X



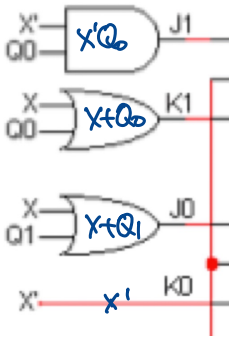
  3. **flip-flop equation** 찾기
     a. ff input boolean expression 찾기

$J_1 = X' Q_0$

$K_1 = X + Q_0$

$J_0 = X + Q_1$

$K_0 = X'$



b. 가능한 모든 현재 state, input 조합에 대하여 찐 ff input value 찾기 → table 완성

Present State Q_1	Present State Q_0	Inputs X	Flip-flop Inputs J_1	Flip-flop Inputs K_1	Flip-flop Inputs J_0	Flip-flop Inputs K_0
0	0	0	0	0	0	1
0	0	1	0	1	1	0
0	1	0	1	1	0	1
0	1	1	0	1	1	0
1	0	0	0	0	1	1
1	0	1	0	1	1	0
1	1	0	1	1	1	1
1	1	1	0	1	1	0

c. next state equation 찾기

i. JK ff equation : $Q(t+1) = K'Q(t) + JQ'(t)$

ii. example의 jk ff equation

$Q1(t+1) = K1'Q1(t) + J1Q1'(t)$

$Q0(t+1) = K0'Q0(t) + J0Q0'(t)$

혹은 상태표 생각하기 →

J	K	Q(t+1)	Operation
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	Complement

iii. table 확장

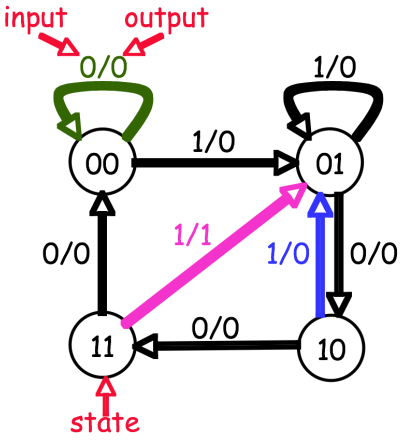
Present State Q_1	Present State Q_0	Inputs X	FF Inputs J_1	FF Inputs K_1	FF Inputs J_0	FF Inputs K_0	Next State Q_1	Next State Q_0
0	0	0	0	0	0	1	0	0
0	0	1	0	1	1	0	0	1
0	1	0	1	1	0	1	1	0
0	1	1	0	1	1	0	0	1
1	0	0	0	0	1	1	1	1
1	0	1	0	1	1	0	0	1
1	1	0	1	1	1	1	0	0
1	1	1	0	1	1	0	0	1

d. 전체 table 완성하기

Present State Q_1	Present State Q_0	Inputs X	FF Inputs J_1	FF Inputs K_1	FF Inputs J_0	FF Inputs K_0	Next State Q_1	Next State Q_0	Outputs Z
0	0	0	0	0	0	1	0	0	0
0	0	1	0	1	1	0	0	1	0
0	1	0	1	1	0	1	1	0	0
0	1	1	0	1	1	0	0	1	0
1	0	0	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0	1	0
1	1	0	1	1	1	1	0	0	0
1	1	1	0	1	1	0	0	1	1

- **state diagram**

Present State Q_1	Q_0	Inputs X	Next State Q_1	Q_0	Outputs Z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	1	0	0	0	0
1	1	1	0	1	1



  - 2개의 ff → 4개의 node(state) ⇒ m개의 ff → 2^n개의 state 가능
  - 1개의 input → 2개의 arrow ⇒ m개의 input → 각 state는 2^m개의 transition

present Q_1	Q_0	input X	ff J_1	K_1	J_0	K_0	next Q_1	Q_0	output Z
0	0	0	0	0	0	1	0	0	0
0	0	1	0	1	1	0	0	1	0
0	1	0	1	1	0	1	1	0	0
0	1	1	0	1	1	0	0	1	0
1	0	0	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0	1	0
1	1	0	1	1	1	1	0	0	0
1	1	1	0	1	1	0	0	1	1