

디지털논리 Term Project

컴퓨터소프트웨어학부

2023085450, 고태우

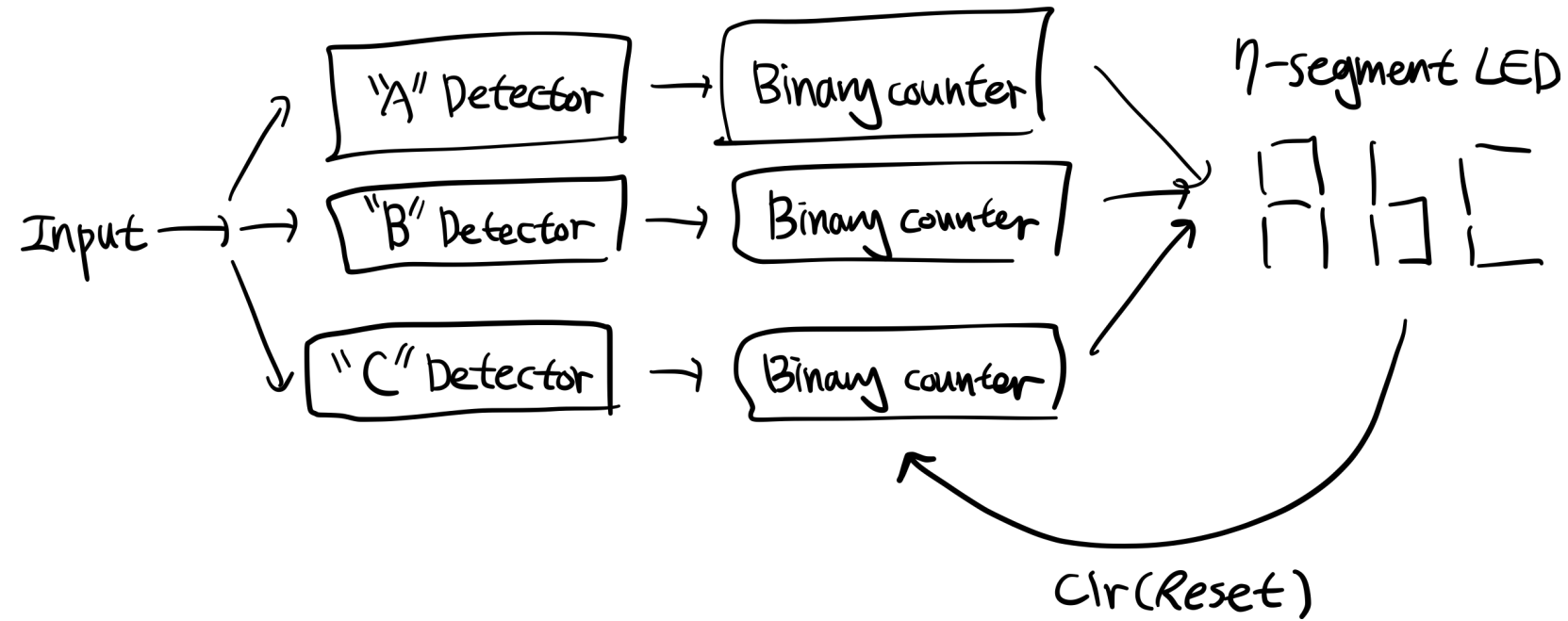
2023065350, 권도현

2023055787, 양호열

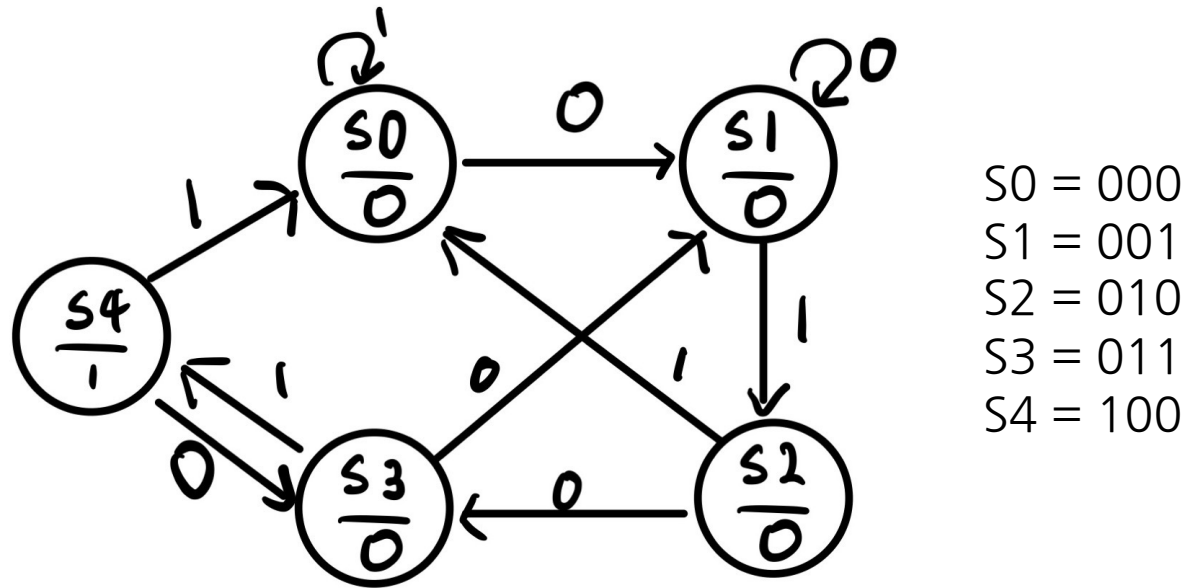
2023058022, 이동운

2023082579, 정영민

Circuit Overview



Code Detector of A("0101") - State Graph



Code Detector of A("0101") - Truth table, K-map

| present | Next | | output |
|----------|-------|-------|--------|
| ABC | X = 0 | X = 1 | Y |
| S0 = 000 | 001 | 000 | 0 |
| S1 = 001 | 001 | 010 | 0 |
| S2 = 010 | 011 | 000 | 0 |
| S3 = 011 | 001 | 100 | 0 |
| S4 = 100 | 011 | 000 | 1 |

| AB \ CX | 00 | 01 | 11 | 10 |
|-------------|----|----|----|----|
| 00 | 0 | 0 | x | 0 |
| 01 | 0 | 0 | x | 0 |
| 11 | 0 | 1 | x | x |
| 10 | 0 | 0 | x | x |
| $D_a = BCX$ | | | | |

| AB \ CX | 00 | 01 | 11 | 10 |
|----------------------------|----|----|----|----|
| 00 | 0 | 1 | x | 1 |
| 01 | 0 | 0 | x | 0 |
| 11 | 1 | 0 | x | x |
| 10 | 0 | 0 | x | x |
| $D_b = B'CX + BC'X' + AX'$ | | | | |

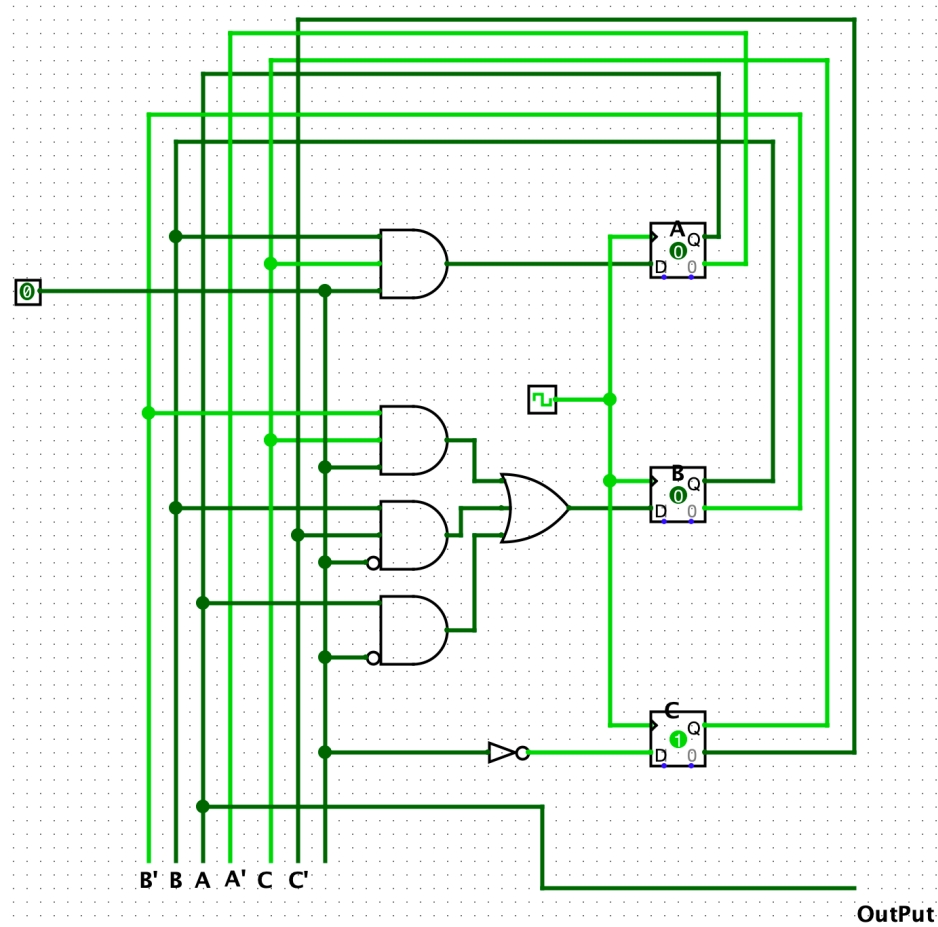
| AB \ CX | 00 | 01 | 11 | 10 |
|------------|----|----|----|----|
| 00 | 1 | 1 | x | 1 |
| 01 | 0 | 0 | x | 0 |
| 11 | 0 | 0 | x | x |
| 10 | 1 | 1 | x | x |
| $D_c = X'$ | | | | |

| A \ BC | 0 | 1 |
|---------|---|---|
| 00 | 0 | 1 |
| 01 | 0 | x |
| 11 | 0 | x |
| 10 | 0 | x |
| $Y = A$ | | |

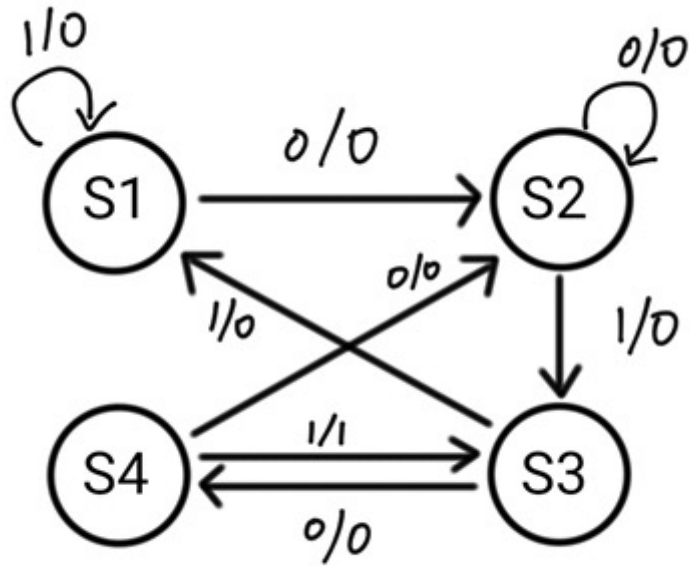
Code Detector of A("0101") - Circuit

우측 회로는 앞에서 구한 각 변수들의 Boolean equation을 회로로 표현한 것이다.

AND, OR 게이트와 3개의 D-FF를 활용하였다.



Additional Task : Code Detector of A("0101") with Mealy machine



S1 = 00

S2 = 01

S3 = 11

S4 = 10

S1 은 초기화한다

S2 은 0을 기억한다

S3 은 01을 기억한다

S4 은 010을 기억한다

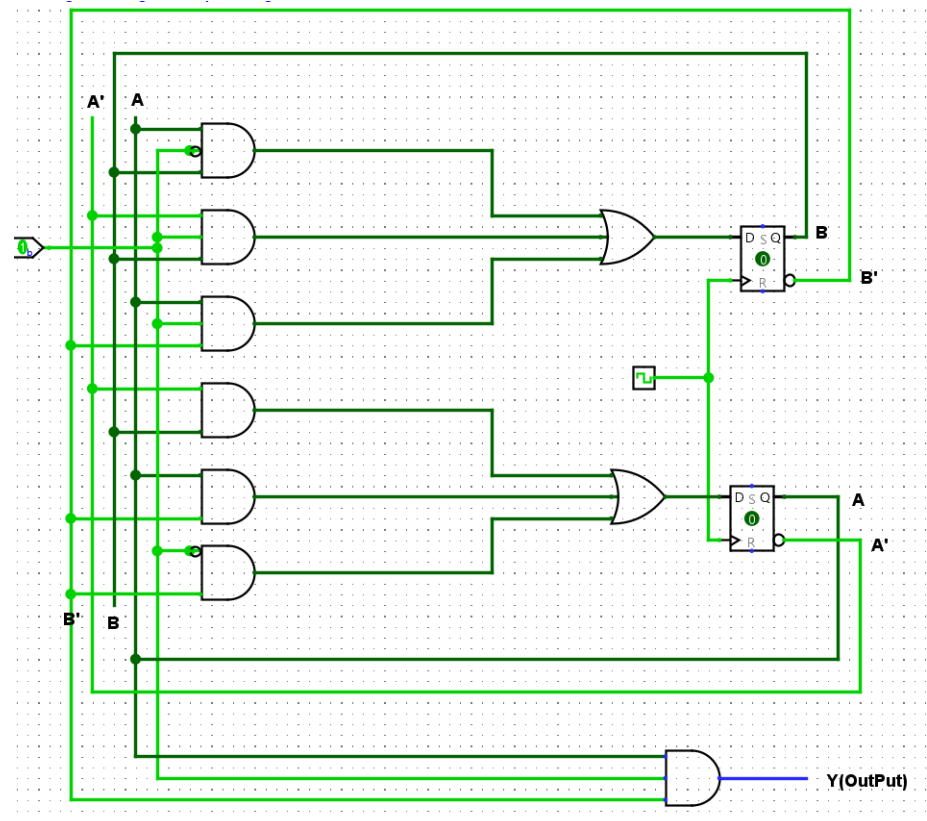
| Input X | Present | | Next | | Output Y | Da | Db |
|------------|---------|---|------|---|-------------|----|----|
| | A | B | A | B | | | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

Additional Task : Code Detector of A("0101") with Mealy machine

$$D_a = ABX' + A'BX + AB'X$$

$$D_b = A'B + AB' + A'B'X'$$

$$Y = AB'X$$



Code Detector of B("1 1 1 0")

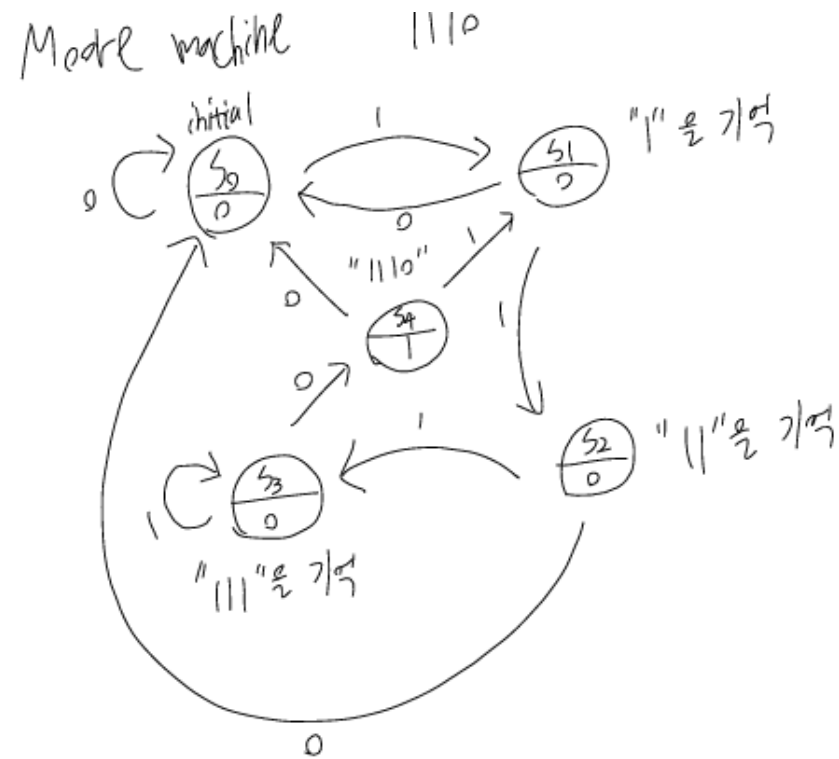
State Table

State Table

$s_0 = 000, s_2 = 010$
 $s_1 = 001, s_3 = 011$
 $s_4 = 100$

| Present State (ABC) | Next State | | Output (Z) |
|------------------------|------------|-----|------------|
| | X=0 | X=1 | |
| 000 | 000 | 001 | 0 |
| 001 | 000 | 010 | 0 |
| 010 | 000 | 011 | 0 |
| 011 | 100 | 011 | 0 |
| 100 | 000 | 001 | 1 |

State Graph



K-Map

| BC \ A | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | X | X | X |
| 2 | 0 | X | X | X |
| 3 | 0 | 0 | 0 | 0 |

| BC \ A | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | X | X | X |
| 2 | 0 | X | X | X |
| 3 | 0 | 1 | 1 | 1 |

| BC \ A | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | X | X | X |
| 2 | 1 | X | X | X |
| 3 | 1 | 0 | 1 | 1 |

| BC \ A | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | X | X | X |

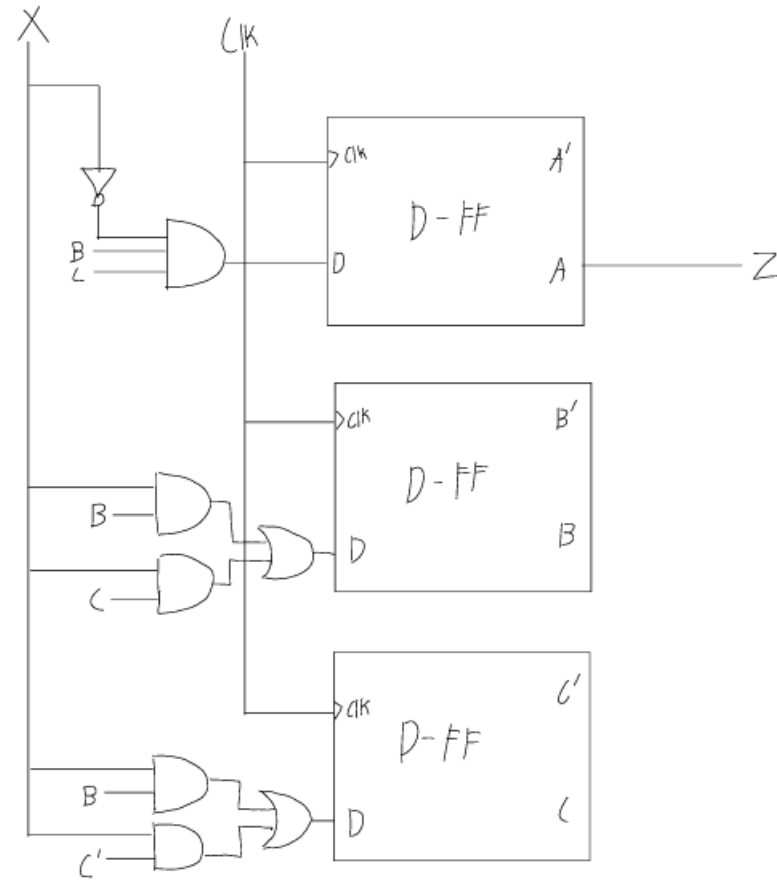
$$A^+ = X'BC$$

$$B^+ = XB + XC$$

$$C^+ = XB + XC'$$

$$Z = A$$

Final Circuit using D-FF



Additional Task : Code Detector of B("1 1 1 0") with Mealy machine

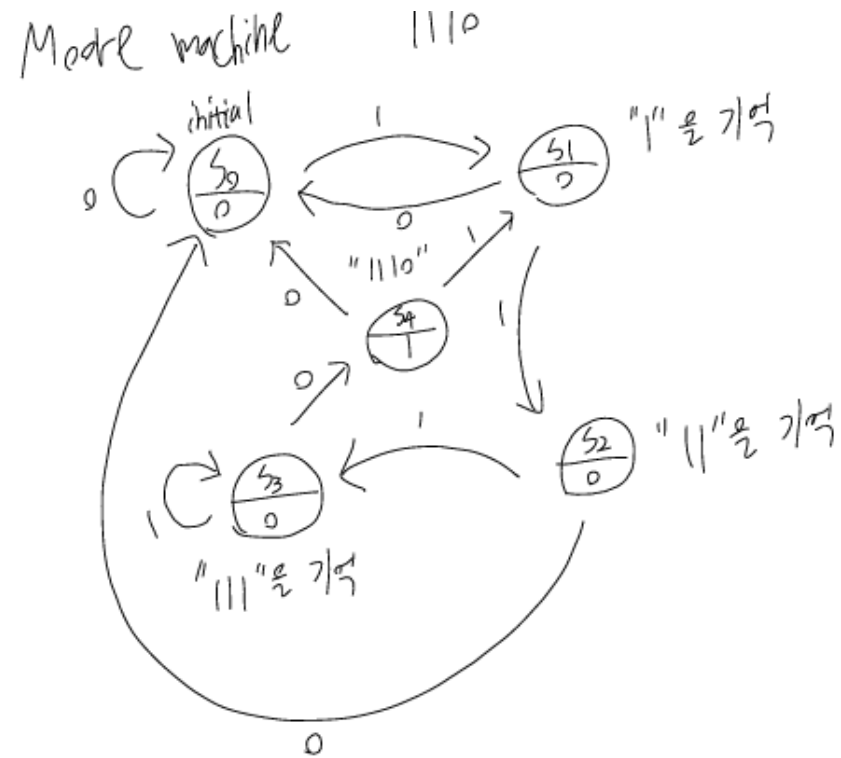
State Table

State Graph

State Table

$s_0 = 000, s_2 = 010$
 $s_1 = 001, s_3 = 011$
 $s_4 = 100$

| present state (ABC) | next state | | 0y+1y+2z |
|------------------------|------------|-----|----------|
| | x=0 | x=1 | |
| 000 | 000 | 001 | 0 |
| 001 | 000 | 010 | 0 |
| 010 | 000 | 011 | 0 |
| 011 | 100 | 011 | 0 |
| 100 | 000 | 001 | 1 |



K-Map

| AB \ X | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |

A^+

| AB \ X | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |

B^+

| AB \ X | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |

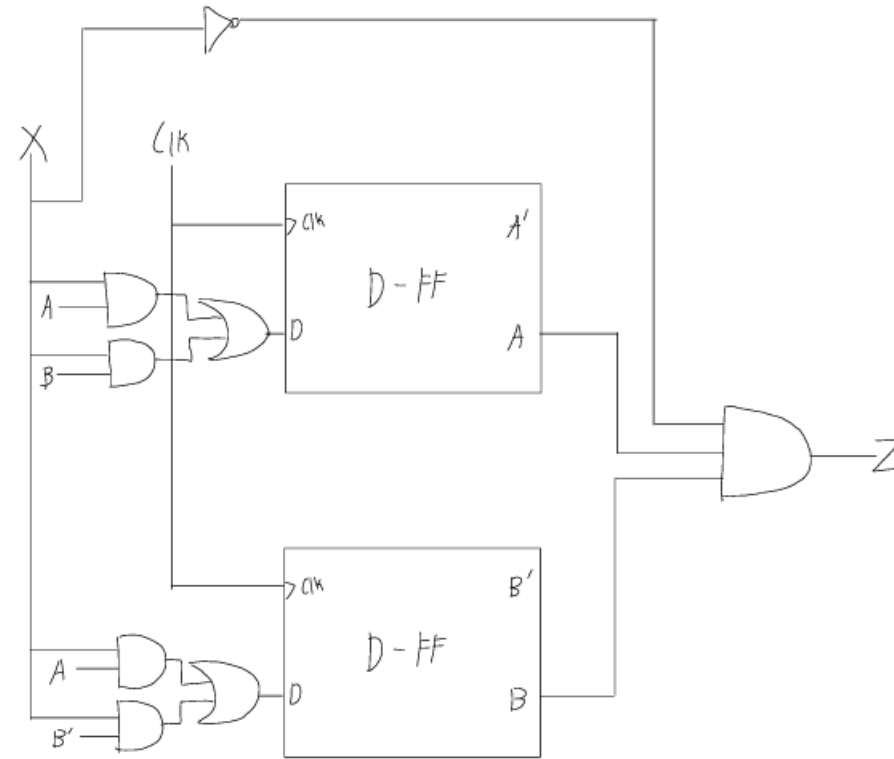
Z

$$A^+ = XA + XB$$

$$B^+ = XA + XB'$$

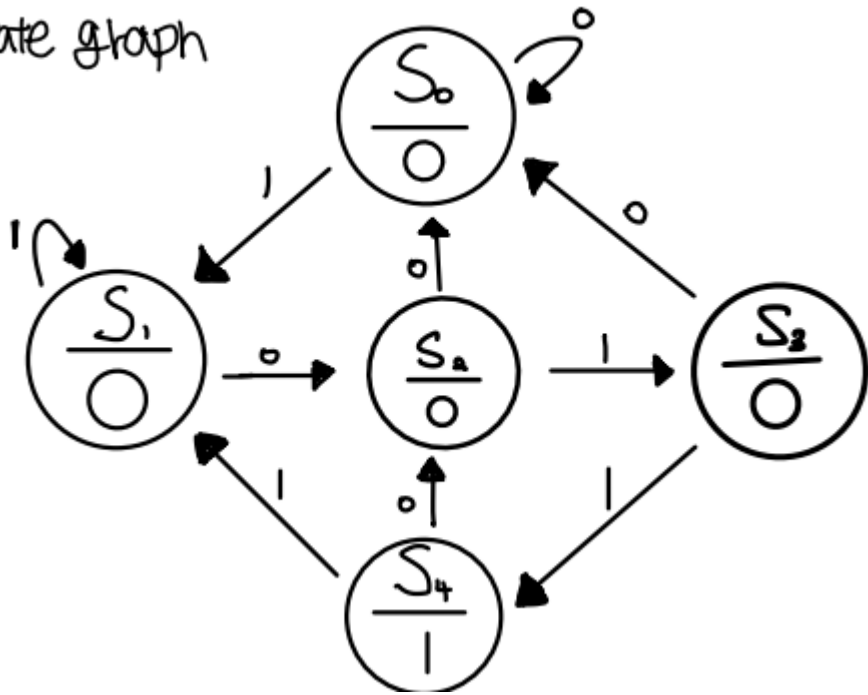
$$Z = X'AB$$

Final Circuit using D-FF



Code Detector of C("1011") - State Graph

State graph



- S0: Initial state
- S1: '1'을 저장
- S2: '10'을 저장
- S3: '101'을 저장
- S4: '1011'이 검출되었음을 알려준다.

Code Detector of C("1011") - State Table

| Present state | Next state | | Output | Present state | Next state | | Output |
|----------------|----------------|----------------|--------|---------------|------------|-----|--------|
| | x=0 | x=1 | | | x=0 | x=1 | |
| S ₀ | S ₀ | S ₁ | 0 | 000 | 000 | 001 | 0 |
| S ₁ | S ₂ | S ₁ | 0 | 001 | 010 | 001 | 0 |
| S ₂ | S ₀ | S ₃ | 0 | 010 | 000 | 011 | 0 |
| S ₃ | S ₀ | S ₄ | 0 | 011 | 000 | 100 | 0 |
| S ₄ | S ₂ | S ₁ | 1 | 100 | 010 | 001 | 1 |

Code Detector of C("1011") - Karnaugh Map

| \overline{AB} C | 00 | 01 | 11 | 10 |
|------------------------|----|----|----|----|
| 00 | 0 | 0 | x | x |
| 01 | 0 | 0 | x | x |
| 11 | 0 | 1 | x | x |
| 10 | 0 | 0 | x | x |

$$J_A = BCX$$

| \overline{AB} C | 00 | 01 | 11 | 10 |
|------------------------|----|----|----|----|
| 00 | x | x | x | 1 |
| 01 | x | x | x | 1 |
| 11 | x | x | x | x |
| 10 | x | x | x | x |

$$K_A = 1$$

| \overline{AB} C | 00 | 01 | 11 | 10 |
|------------------------|----|----|----|----|
| 00 | 0 | x | x | 1 |
| 01 | 0 | x | x | 0 |
| 11 | 0 | x | x | x |
| 10 | 1 | x | x | x |

$$J_B = AX' + CX'$$

| \overline{AB} C | 00 | 01 | 11 | 10 |
|------------------------|----|----|----|----|
| 00 | 0 | 0 | x | 1 |
| 01 | 0 | 0 | x | 1 |
| 11 | 0 | 0 | x | x |
| 10 | 0 | 0 | x | x |

$$Z = A$$

| \overline{AB} C | 00 | 01 | 11 | 10 |
|------------------------|----|----|----|----|
| 00 | x | 1 | x | x |
| 01 | x | 0 | x | x |
| 11 | x | 1 | x | x |
| 10 | x | 1 | x | x |

$$K_B = X' + C$$

| \overline{AB} C | 00 | 01 | 11 | 10 |
|------------------------|----|----|----|----|
| 00 | 0 | 0 | x | 0 |
| 01 | 1 | 1 | x | 1 |
| 11 | x | x | x | x |
| 10 | x | x | x | x |

$$J_C = X$$

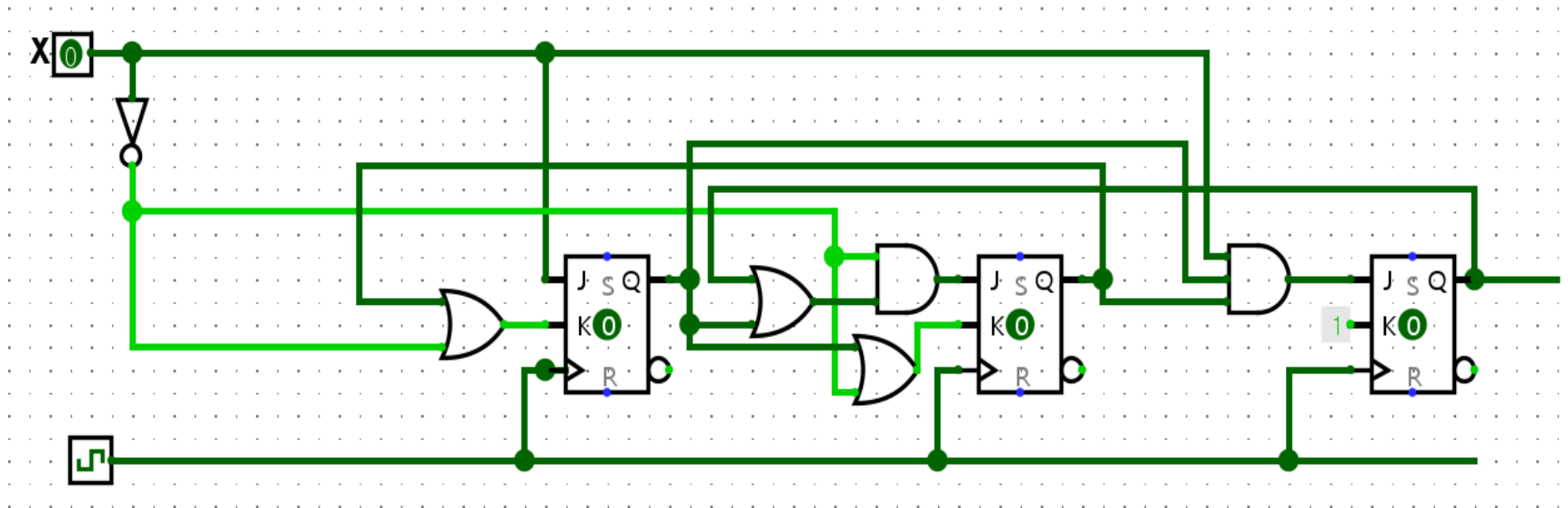
| \overline{AB} C | 00 | 01 | 11 | 10 |
|------------------------|----|----|----|----|
| 00 | x | x | x | x |
| 01 | x | x | x | x |
| 11 | 0 | 1 | x | x |
| 10 | 1 | 1 | x | x |

$$K_C = X' + B$$

Code Detector of C("1011") - Boolean Algebra

| Ja | Ka | Jb | Kb | Jc | Kc | Z |
|-----|----|-------|--------|----|--------|---|
| BCX | 1 | AX' | $X'+C$ | X | $X'+B$ | A |

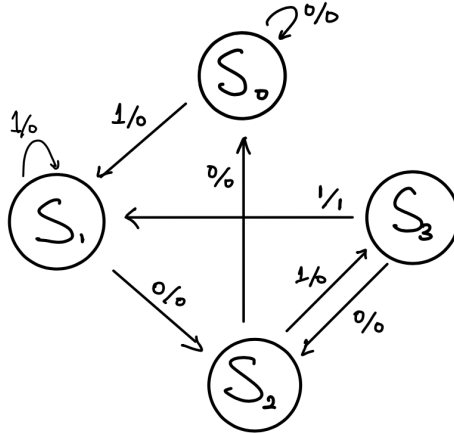
Code Detector of C("1011") using JK-FFs



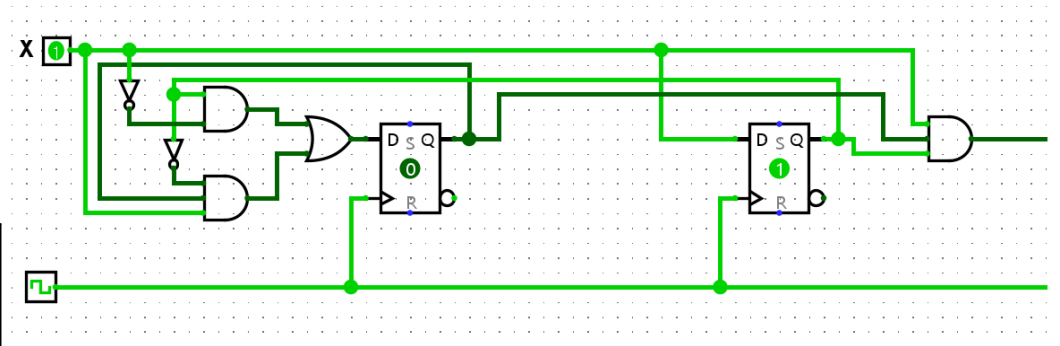
Additional Task : Code Detector of C("1011") with Mealy machine

2. Mealy machine

S_0 : Initial state (0 0)
 S_1 : Remember '1' (0 1)
 S_2 : Remember '10' (1 0)
 S_3 : Remember '101' (1 1)



| Present state | Next state | | output (Z) | |
|---------------|------------|-------|------------|-----|
| | x=0 | x=1 | x=0 | x=1 |
| S_0 | S_0 | S_1 | 0 | 0 |
| S_1 | S_2 | S_1 | 0 | 0 |
| S_2 | S_0 | S_3 | 0 | 0 |
| S_3 | S_2 | S_1 | 0 | 1 |



5bits-counter-truth_table

| A | B | C | D | E | X | Z | A+ | B+ | C+ | D+ | E+ | DA | DB | DC | DD | DE |
|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

좌측 truth_table은
A,B,C,D,E,X(input),Z(output),A+,B+,
C+,D+,E+,DA,DB,DC,DD,DE
각각에 대한 truth_table 이다.

5bits-counter-Boolean equation

QM 알고리즘을 이용하여, Z, DA, DB, DC, DD, DE에 대해 Boolean equation을 구하면 다음과 같다.

$$Z=ABCDEX$$

$$DA=A'BCDEX+AB'+AB'+AC'+AD'+AE'+AX'$$

$$DB=B'CDEX+BC'+BD'+BE'+BX'$$

$$DC=C'DEX+CD'+CE'+CX'$$

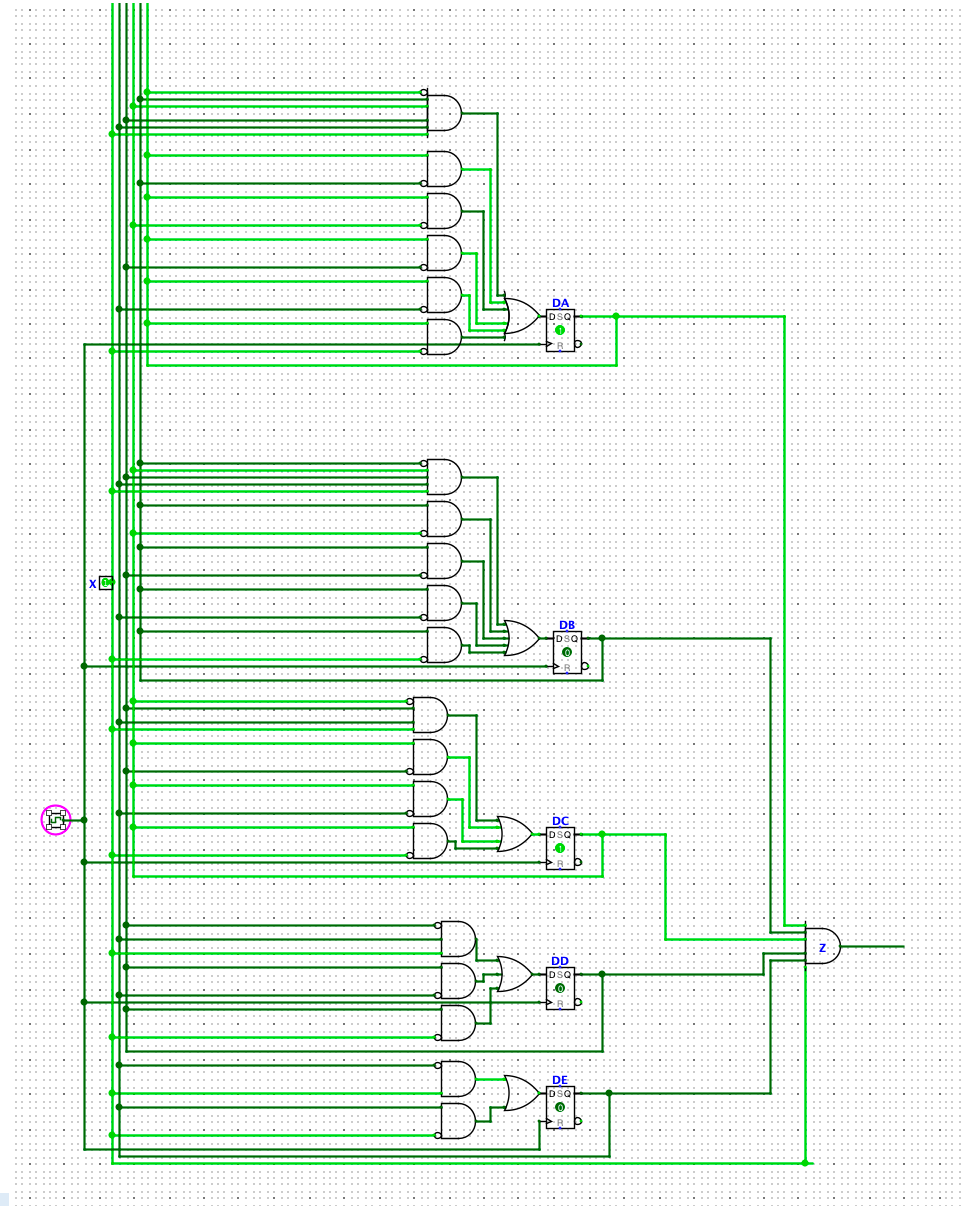
$$DD=D'EX+DE'+DX'$$

$$DE=E'X+EX'$$

5-bits-counter-circuit

우측 회로는 앞에서 구한 각 변수들의 Boolean equation을 회로로 표현한 것이다.

5개의 D-FF과 AND, OR, NOT Gate를 활용하여 구현하였다.

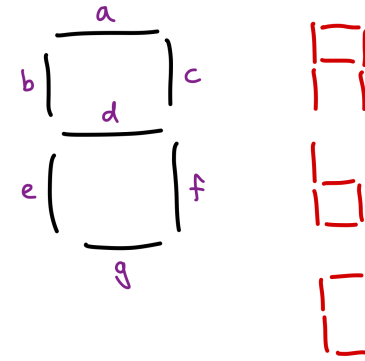


Short Discription

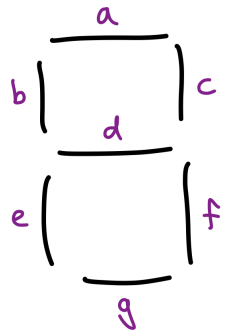
3개의 input(A, B, C)를 받아 들어오는 input을 LED로 표시한다.
동시에, Binary counter 부분으로 Clr=1을 보낸다.

A, B, C 중 2개 이상이 1인 경우는 없다.

3-input, 1-output with 7-segment LED



Truth table, K-map of LED inputs



$A: a.b.c.d.e.f$

$B: b.d.e.f.g$

$C: a.b.e.g$

| A | B | C | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

| c \ AB | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0 | 0 | 0 | 0 | x |
| 1 | 1 | x | x | x |

$$a = A + C$$

| c \ AB | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0 | 0 | 0 | 1 | x |
| 1 | 1 | x | x | x |

$$b = A + B + C = e$$

| c \ AB | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0 | 0 | 0 | 0 | x |
| 1 | 1 | 0 | x | x |

$$c = A$$

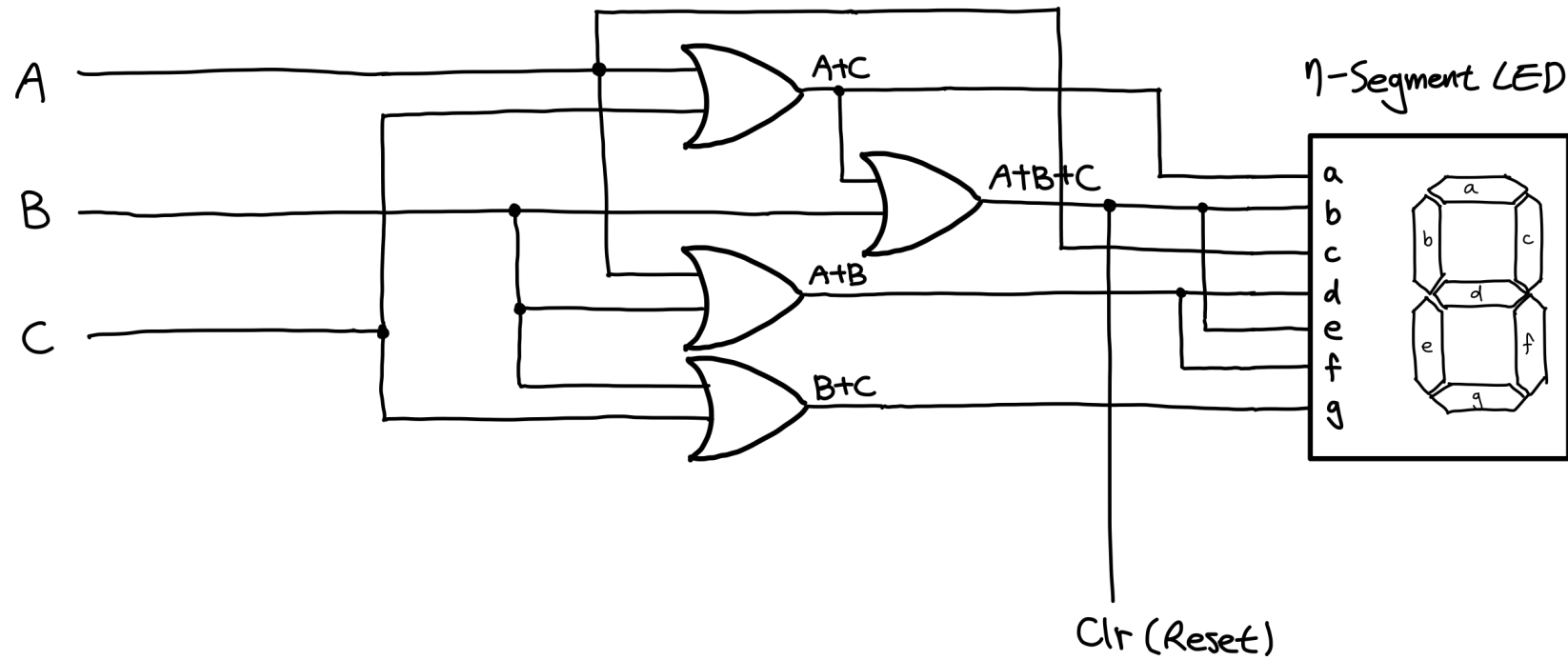
| c \ AB | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0 | 0 | 0 | 1 | x |
| 1 | 0 | x | x | x |

$$d = A + B = f$$

| c \ AB | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0 | 0 | 0 | 1 | x |
| 1 | 1 | x | x | x |

$$g = B + C$$

Logic Gate



Discussion : Display time of LED

LED가 표시되는 시간을 따로 생각하지 않고 구현하면,
Counter들이 Reset되는 시점, 즉 1클락 동안만 LED가 표시된다.
(LED로의 입력 신호를 받자마자 Reset=1을 보낼 예정이기 때문)

기본) 1클락동안 표시 - 너무 짧다. 사용자가 LED를 확인하기도 전에 꺼져버릴 것

Idea 1. 모종의 레지스터를 이용해 n클락동안 표시

Idea 2. 새로운 31번 등장한 코드가 나오기 전까지 이전 결과를 계속해서 표시
정도의 아이디어를 생각해볼 수 있다.

그냥 1클락동안 표시되는 회로를 짤 수도 있겠지만,
보다 실용적인 회로를 위해 2번째 아이디어를 채택해 이전에 출력된 값을 레지스터에 저장하고,
계속해서 표시하는 기능을 추가한다.

Additional function design : Idea

3개의 T flip-flop을 이용, A, B, C의 값을 저장한다.

이전 Gate의 A, B, C를 위 플립플롭의 output QA, QB, QC로 대체한다.

효율적인 게이트 구성을 위해, 세 input(A, B, C)의 값이 0 0 0일 경우 변화가 없도록 한다.

T flip-flop이므로, 플립플롭 값의 변화는 :

원래 1이었다면, 그 플립플롭에 해당하는 input 이외의 input 중 하나가 1일 때 toggle

원래 0이었다면, 그 플립플롭에 해당하는 input이 1일 때 toggle

중요한 점은, 다른 플립플롭의 output의 영향을 받지 않는다는 것이다.

Additional function design : K-map

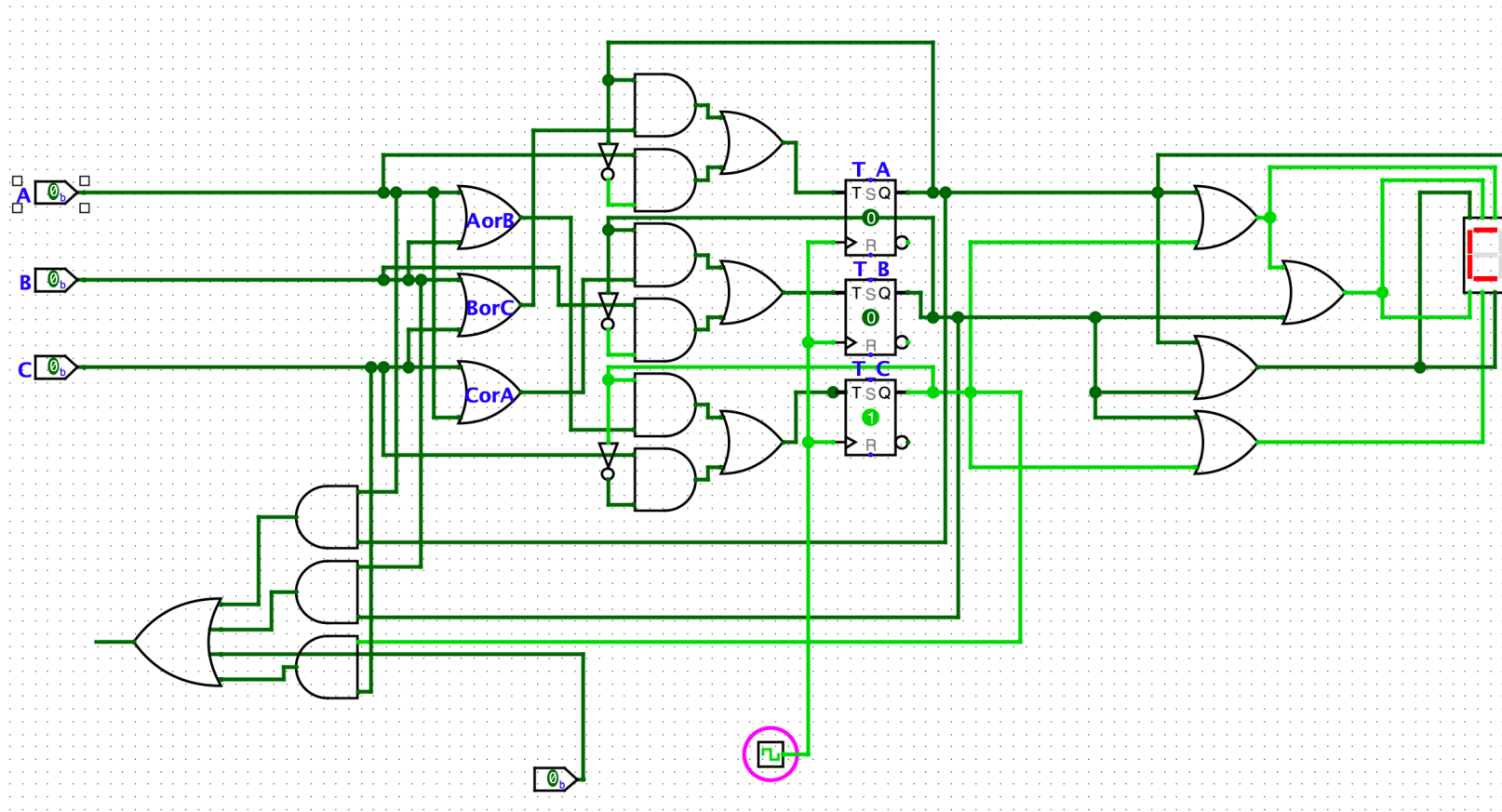
| A | B | C | Q _A | Q _A ' | T _A |
|---|---|---|----------------|------------------|----------------|
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |

| AB \ CQ _A | 00 | 01 | 11 | 10 |
|----------------------|----|----|----|----|
| 00 | 0 | 0 | X | 1 |
| 01 | 0 | 1 | X | 0 |
| 11 | 1 | X | X | X |
| 10 | 0 | X | X | X |

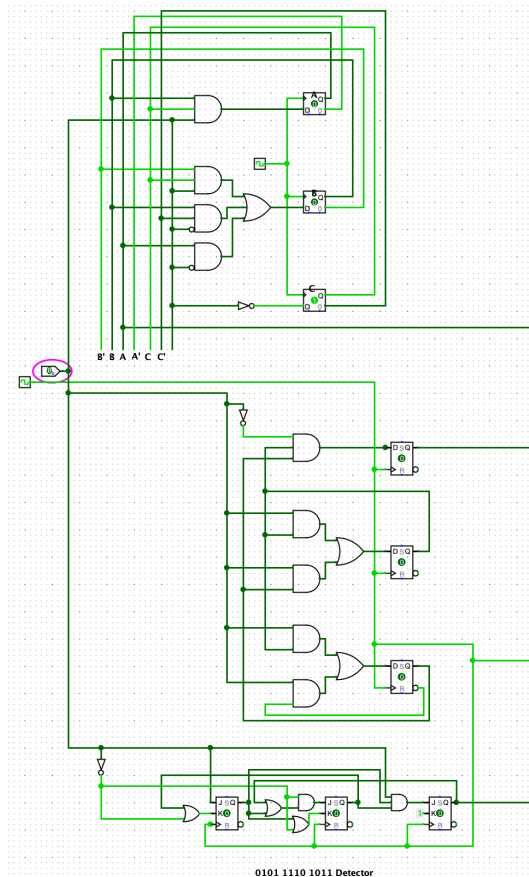
$$\begin{aligned}
 T_A &= CQ_A + Q_A'A + QAB \\
 &= Q_A(B+C) + Q_A'A
 \end{aligned}$$

T_b, T_c는 문자만 바뀐
비슷한 식이 나온다.

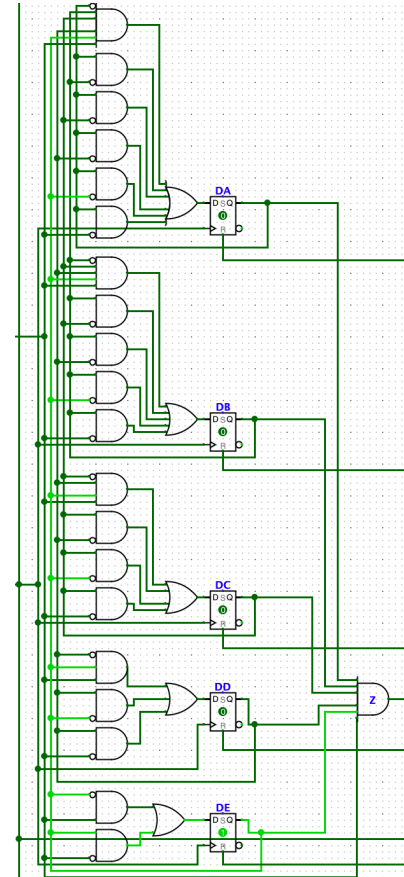
Logic Gate (with additional function)



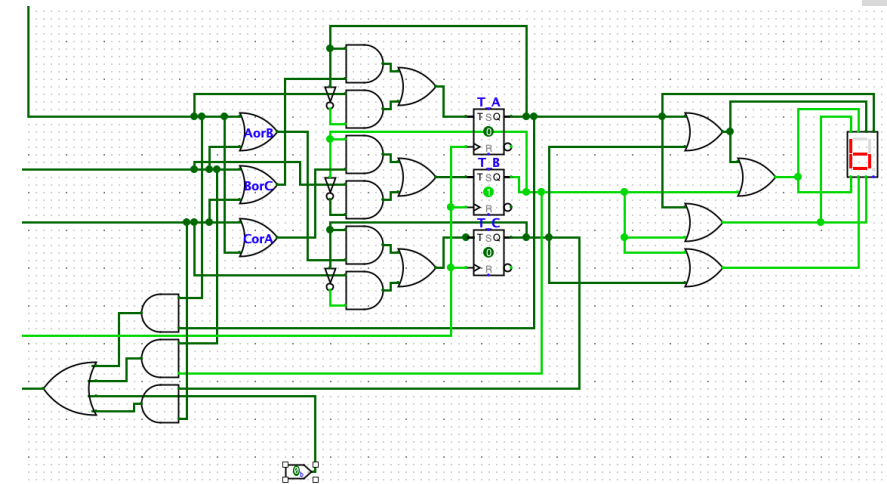
Combined Circuit



Three Code Detectors



Binary counter



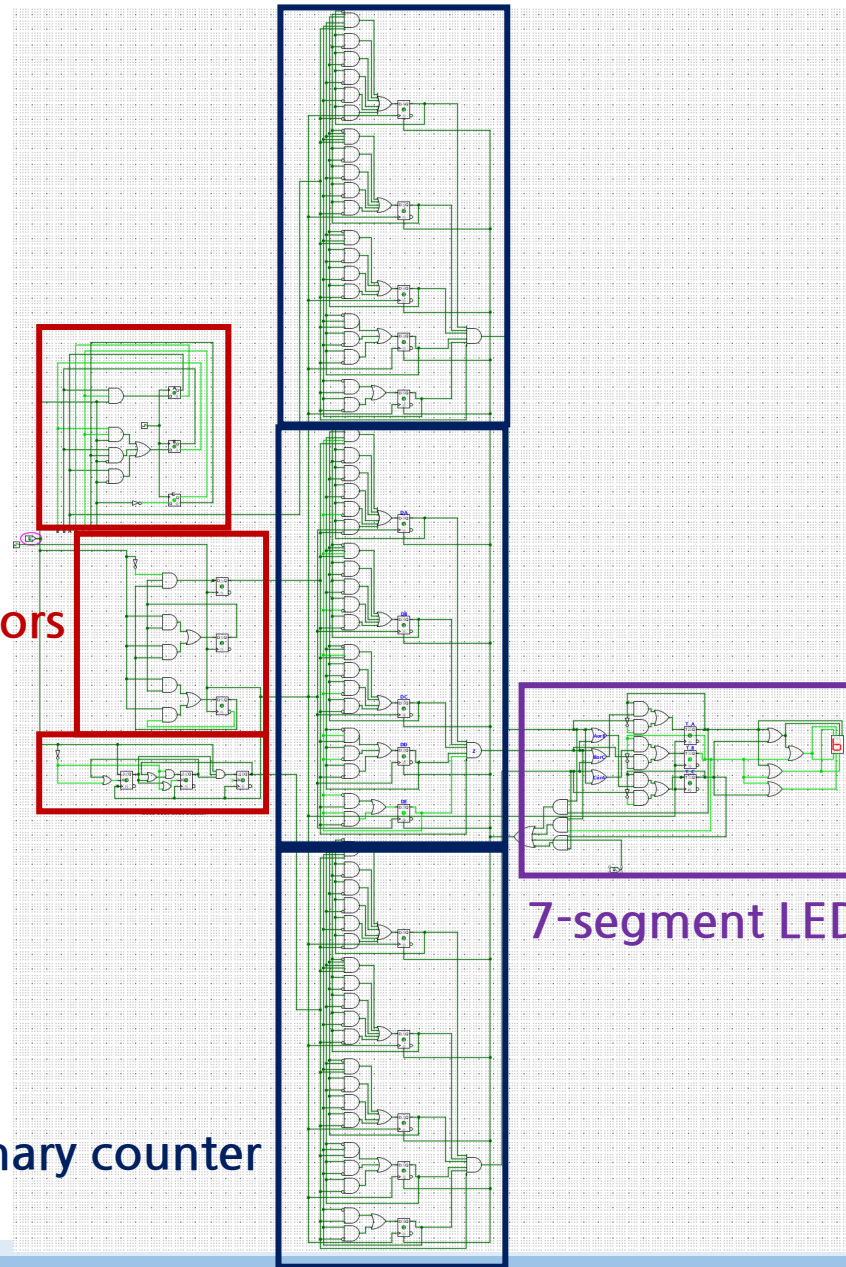
7-segment LED

Combined Circuit

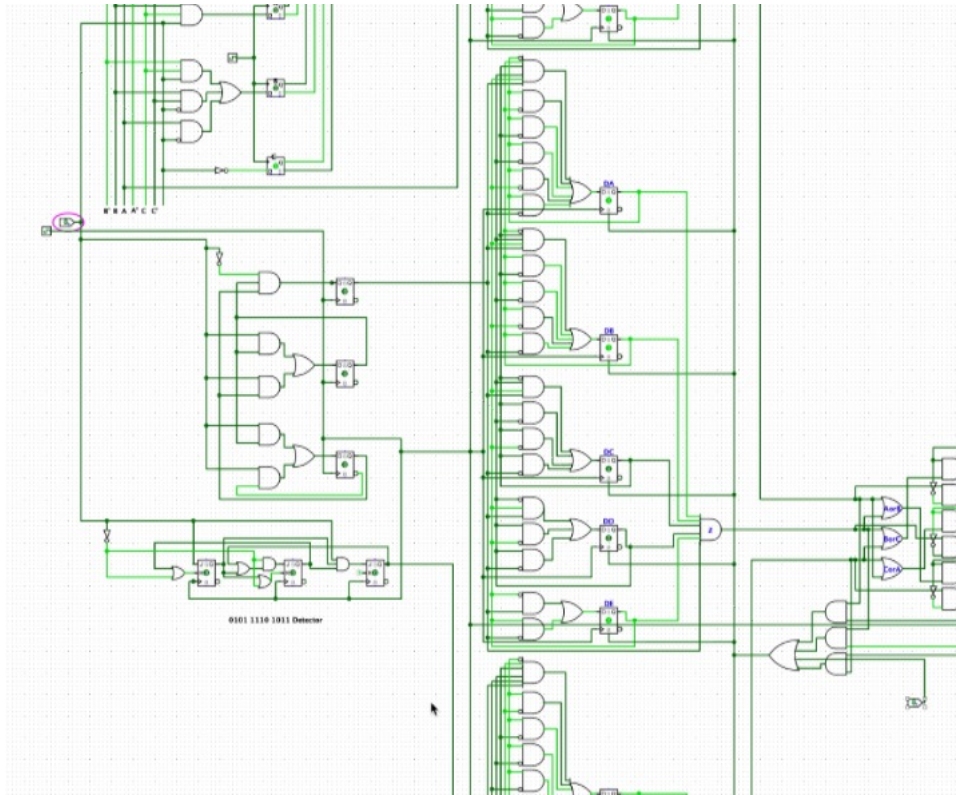
Three Code Detectors

Binary counter

7-segment LED



Circuit Simulation



B를 여러 번 입력하고(1110),
Counter를 거쳐 LED에 표시되기까지의
과정을 영상으로 담았다.

(LMS에 함께 제출)

팀 프로젝트 파트 분배

고태우 : Circuit Overview/Combining, 7-segment LED connection

이동윤 : Code Detector of A("0101")

정영민 : Code Detector of B("1110")

권도현 : Code Detector of C("1011")

양호열 : 5-bit Binary Counter