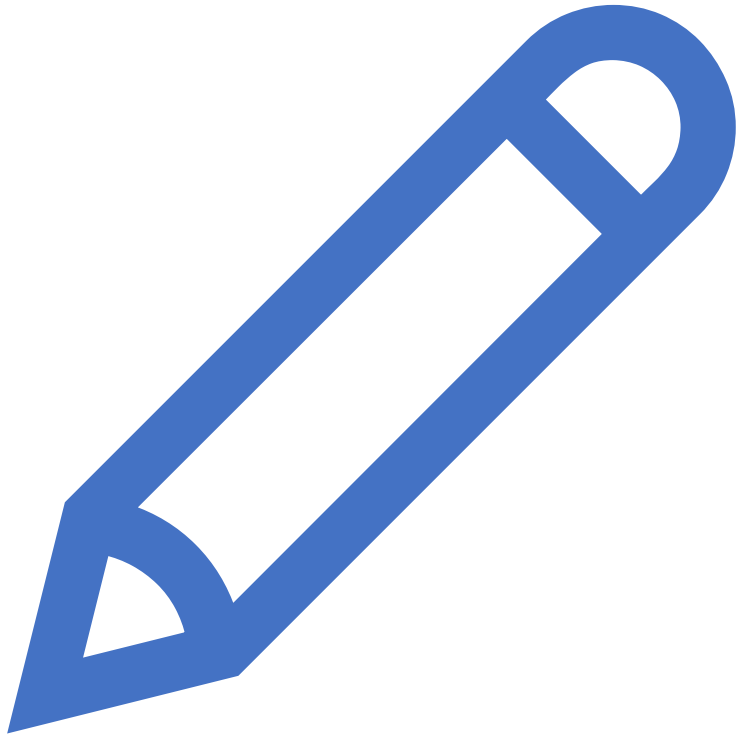


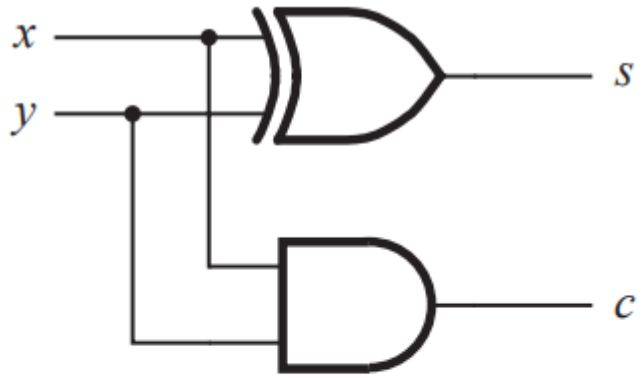
14주차
Sequence Detector

서기환(20181849)
전해찬(20211588)

목차

- FSM
- Moore machine
- Mealy machine
- sequence detector
- Partitioning Minimization Procedure

FSM(유한 상태 기계, finite-state machine)



(c) Circuit

Figure 5.2 Half-adder.

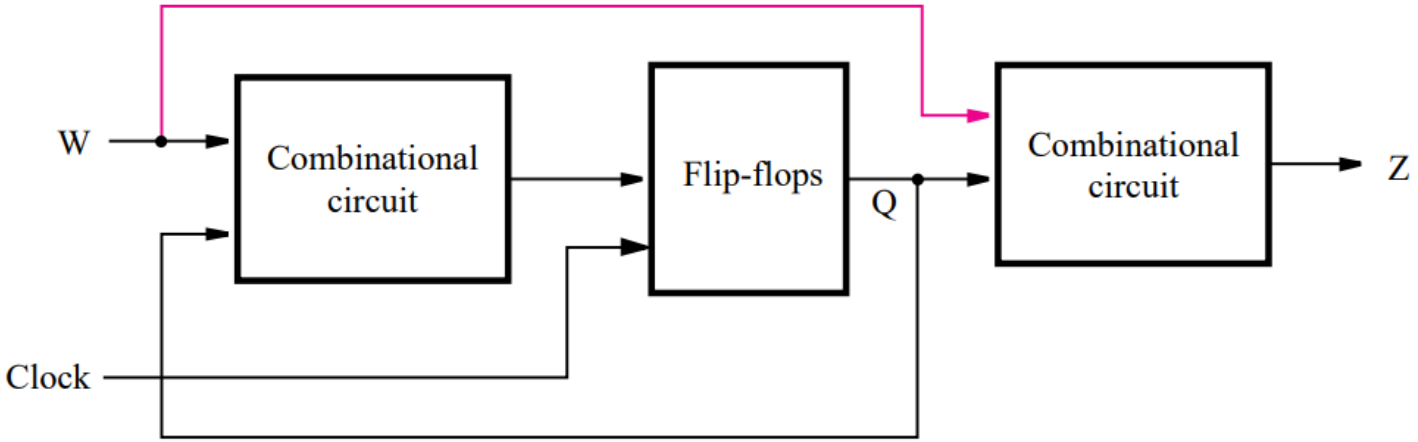


Figure 8.1 The general form of a sequential circuit.

Combinational circuit

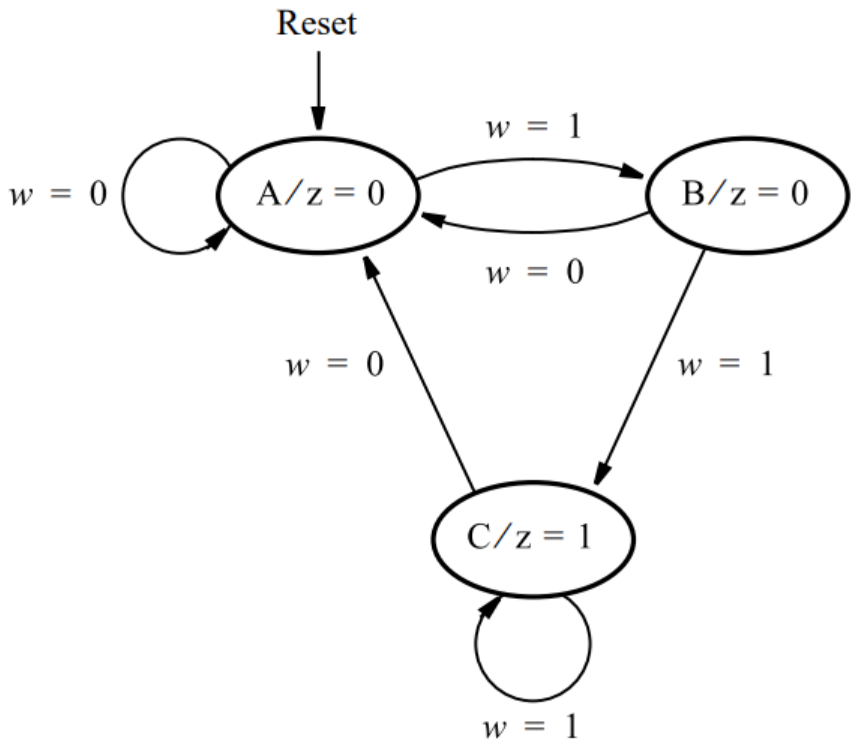
- output is dependent only by its current inputs

Sequential circuit (FSM)

- output depends not only on the present input but also on the history of the input
- functional behavior of these circuits can be represented using a finite number of states

Moore machine

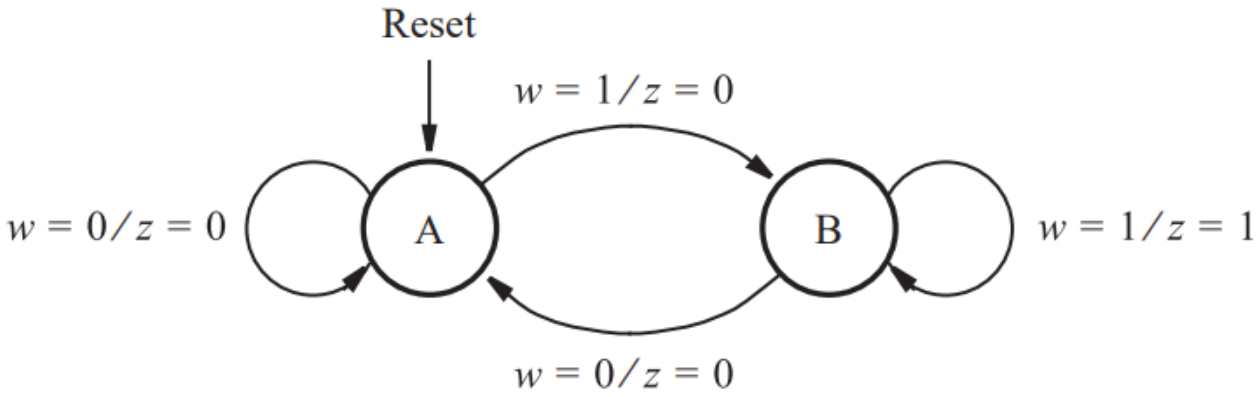
- outputs depend only on the state of the circuit



Present state	Next state		Output z
	$w = 0$	$w = 1$	
A	A	B	0
B	A	C	0
C	A	C	1

Mealy machine

- outputs depend on both the state and the inputs

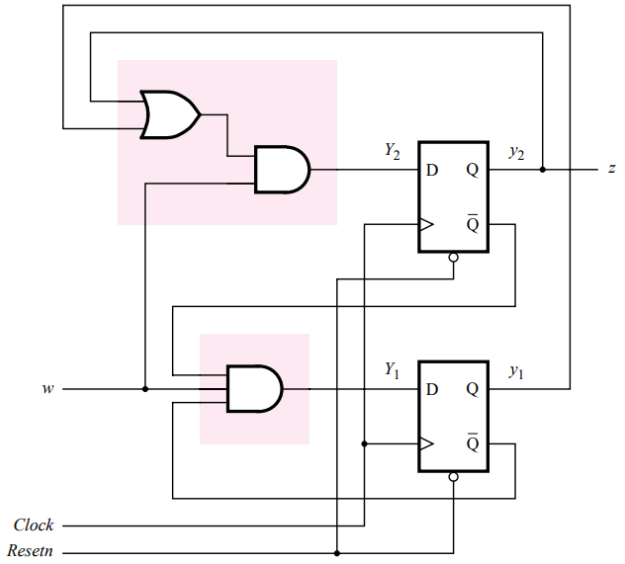


Present state	Next state		Output z	
	$w = 0$	$w = 1$	$w = 0$	$w = 1$
A	A	B	0	0
B	A	B	0	1

Moore VS Mealy machine

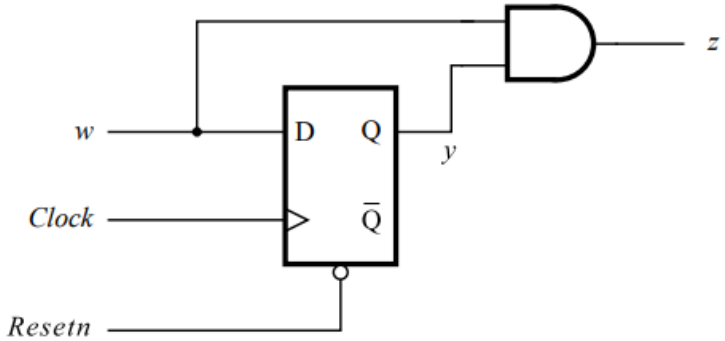
- Moore

	Present state y_2y_1	Next state $w=0$ Y_2Y_1	Next state $w=1$ Y_2Y_1	Output z
A	00	00	01	0
B	01	00	10	0
C	10	00	10	1
	11	*dd*	*dd*	*d*



- Mealy

	Present state y	Next state $w=0$ Y	Next state $w=1$ Y	Output $w=0$ z	Output $w=1$ z
A	0	0	1	0	0
B	1	0	1	0	1



Moore VS Mealy machine

- Moore

Clock cycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	0	1	0	0	1	1	0

- Mealy

Clock cycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	1	0	0	1	1	0	0

Sequence detector

입력 시퀀스의 비트를 받아 타겟 시퀀스 가 검출 될 때마다 출력 1을 생성하는 순차 상태 머신

Mealy

Moore

1. Overlapping : last bit of one sequence becomes the first bit of the next sequence
2. Non-Overlapping : the last bit of one sequence does not become the first bit of the next sequence

Sequence Detector 예시: 101 Mealy

Mealy

Non overlapping

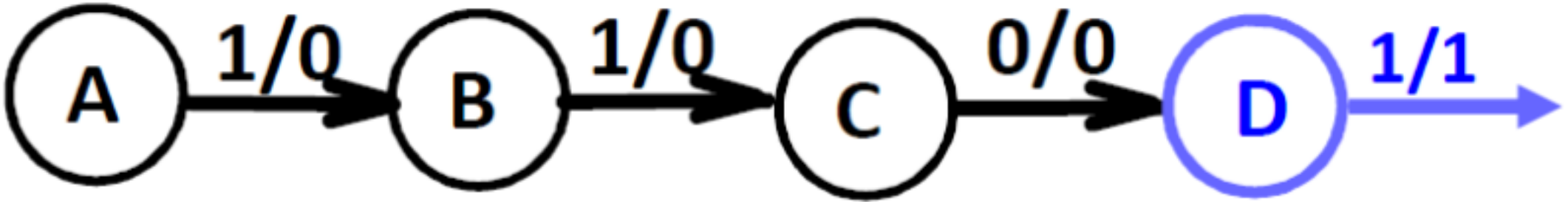
0	1	1	0	1	0	1	0	1	1	0	0	1
0	0	0	0	1	0	0	0	1	0	0	0	0

overlapping

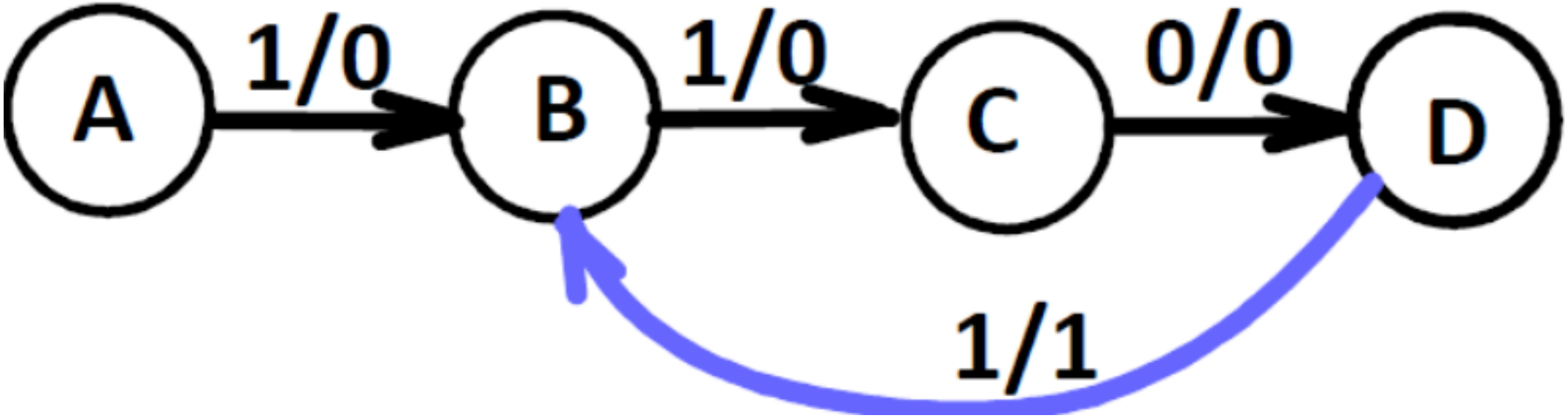
0	1	1	0	1	0	1	0	1	1	0	0	1
0	0	0	0	1	0	1	0	1	0	0	0	0

Sequence Detector: 1101 Mealy - Overlapping

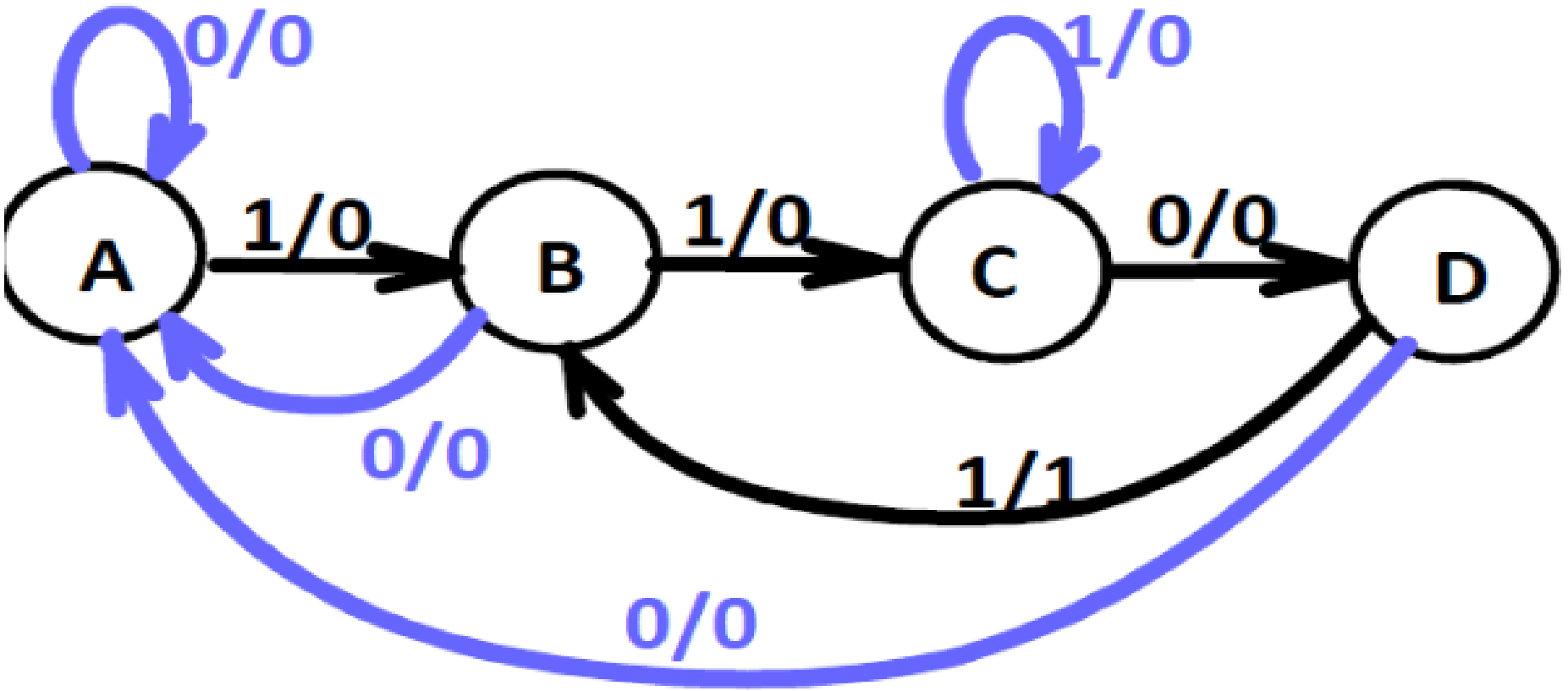
State Diagram



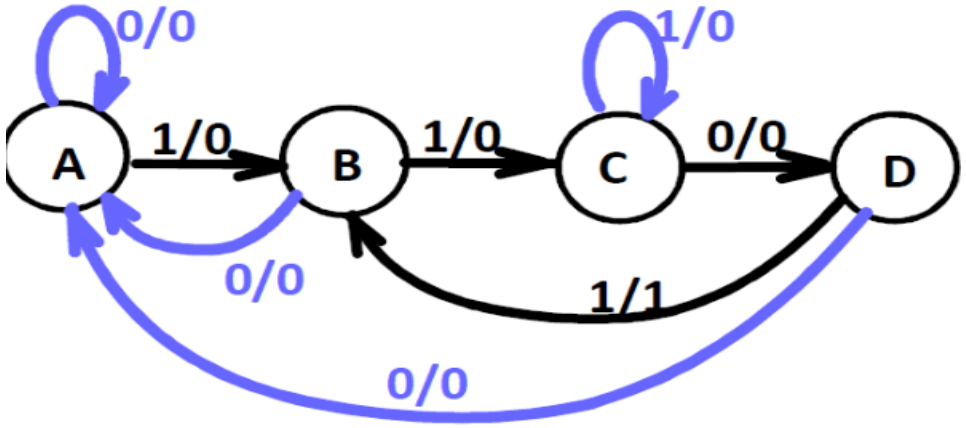
Sequence Detector: 1101 Mealy



Sequence Detector: 1101 Mealy



Sequence Detector: 1101 Mealy



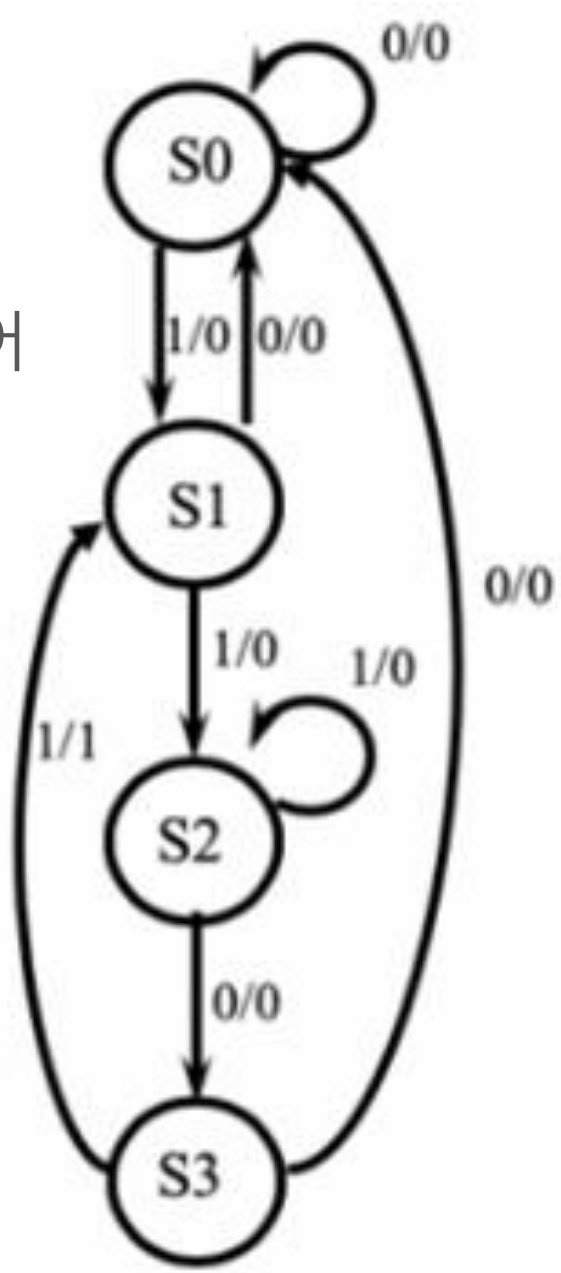
Present State	Next State x=0	Next State x=1	Output x=0	Output x=1
A	A	B	0	0
B	A	C	0	0
C	D	C	0	0
D	A	B	0	1

Sequence Detector : 1101 Mealy

1. Overlapping

1이 출력된 후:
마지막 bit(1)이 S0이 되어
S1로 감!

Present State	Next State		Output	
	X = 0	X = 1	X =0	X = 1
S0	S0	S1	0	0
S1	S0	S2	0	0
S2	S3	S2	0	0
S3	S0	S1	0	1



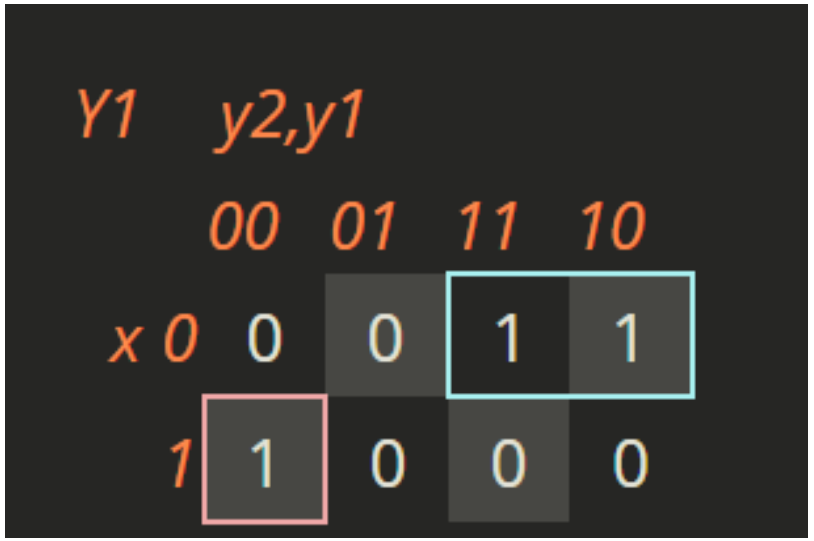
Sequence Detector : 1101 Mealy - Overlapping

1.

Present State	Next State		Output	
	X = 0	X = 1	X =0	X = 1
S0	S0	S1	0	0
S1	S0	S2	0	0
S2	S3	S2	0	0
S3	S0	S1	0	1

Present State	Next State		Output	
	X = 0	X = 1	X =0	X = 1
y2y1	Y2Y1	Y2Y1	Z	
00	00	01	0	0
01	00	10	0	0
10	11	10	0	0
11	00	01	0	1

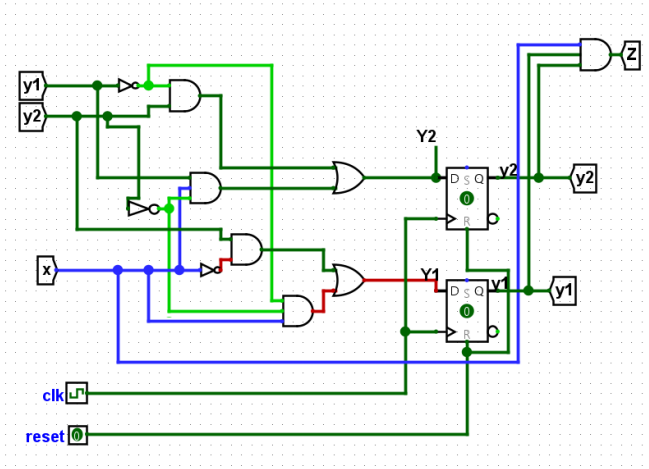
Sequence Detector : 1101 Mealy - Overlapping



Y2(x, y2, y1) = y2y1' + xy2'y1

Y1(x, y2, y1) = x'y2 + xy2'y1'

Z(x, y2, y1) = xy2y1

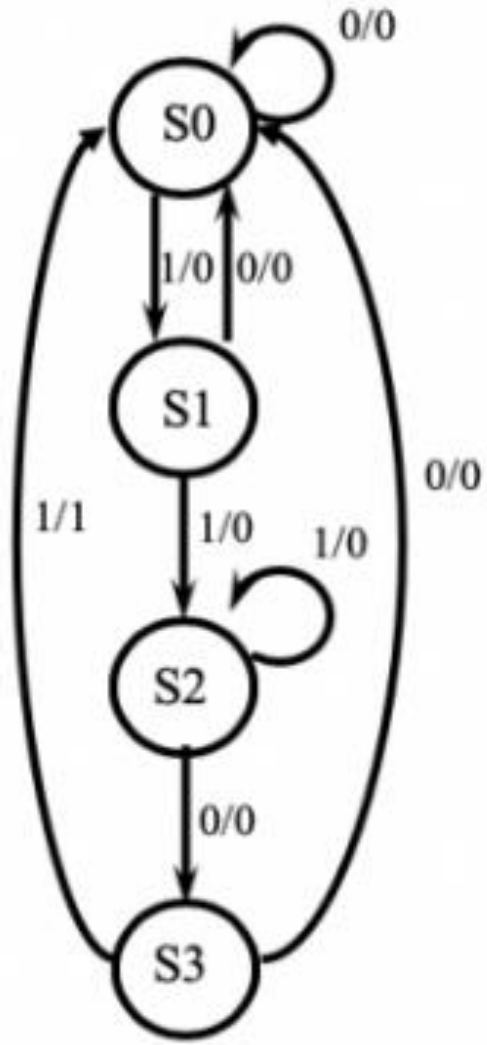


Sequence Detector : 1101 Mealy

2. Non overlapping

1이 출력된 후:
마지막 bit(1)이 S0이 되지 않음!

Present State	Next State		Output	
	X = 0	X = 1	X =0	X = 1
S0	S0	S1	0	0
S1	S0	S2	0	0
S2	S3	S2	0	0
S3	S0	S0	0	1



Sequence Detector : 1101 Mealy - Non overlapping

Present State	Next State		Output	
	X = 0	X = 1	X =0	X = 1
S0	S0	S1	0	0
S1	S0	S2	0	0
S2	S3	S2	0	0
S3	S0	S0	0	1

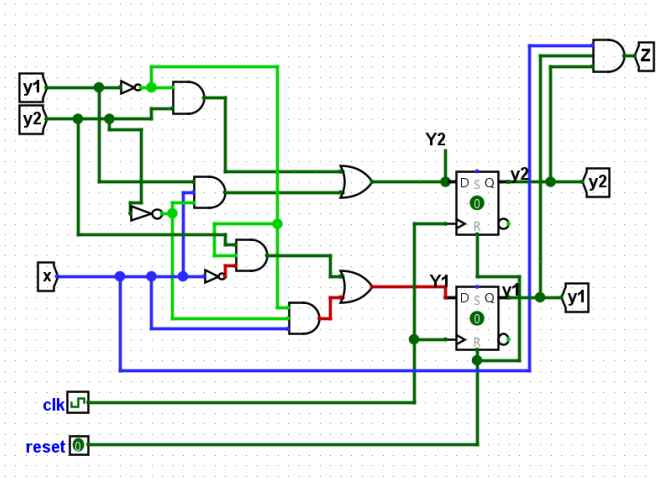
Present State	Next State		Output	
	X = 0	X = 1	X =0	X = 1
y2y1	Y2Y1	Y2Y1	Z	
00	00	01	0	0
01	00	10	0	0
10	11	10	0	0
11	00	00	0	1

Sequence Detector : 1101 Mealy - Non overlapping

Y2(x, y2, y1) = y2y1' + xy2'y1

Y1(x, y2, y1) = x'y2y1' + xy2'y1'

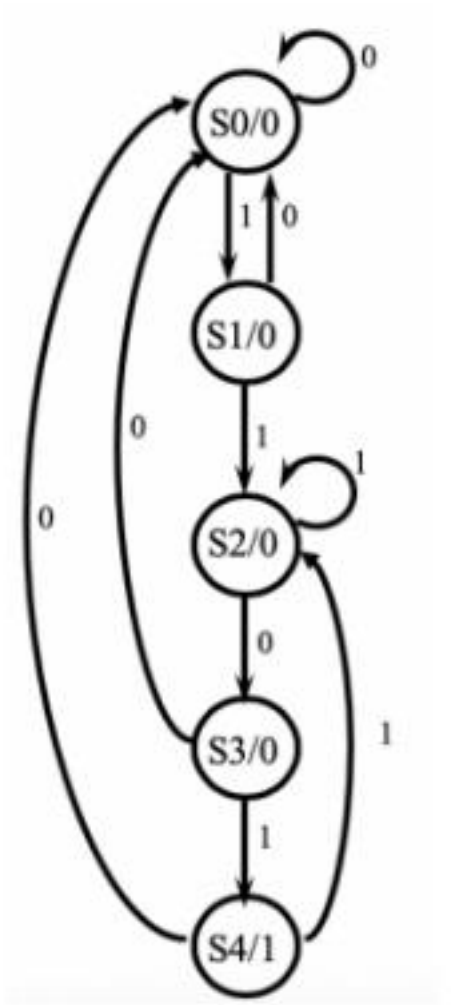
Z(x, y2, y1) = xy2y1



Sequence Detector : 1101 Moore

1.Overlapping

Present State	Next State		Output
	X = 0	X = 1	Y
S0	S0	S1	0
S1	S0	S2	0
S2	S3	S2	0
S3	S0	S4	0
S4	S0	S2	1

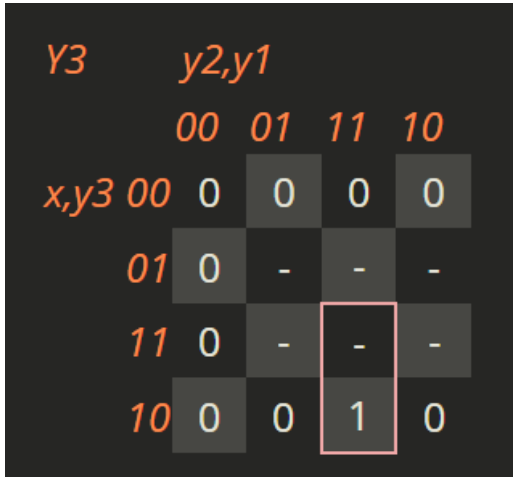


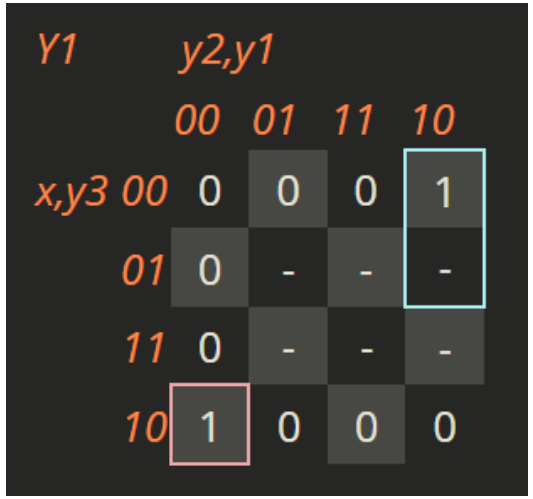
Sequence Detector : 1101 Moore - Overlapping

Present State	Next State		Output
	X = 0	X = 1	Y
S0	S0	S1	0
S1	S0	S2	0
S2	S3	S2	0
S3	S0	S4	0
S4	S0	S2	1

Present State	Next State		Output
	X = 0	X = 1	z
y3y2y1	Y3Y2Y1	Y3Y2Y1	
000	000	001	0
001	000	010	0
010	011	010	0
011	000	100	0
100	000	010	1

Sequence Detector : 1101 Moore - Overlapping



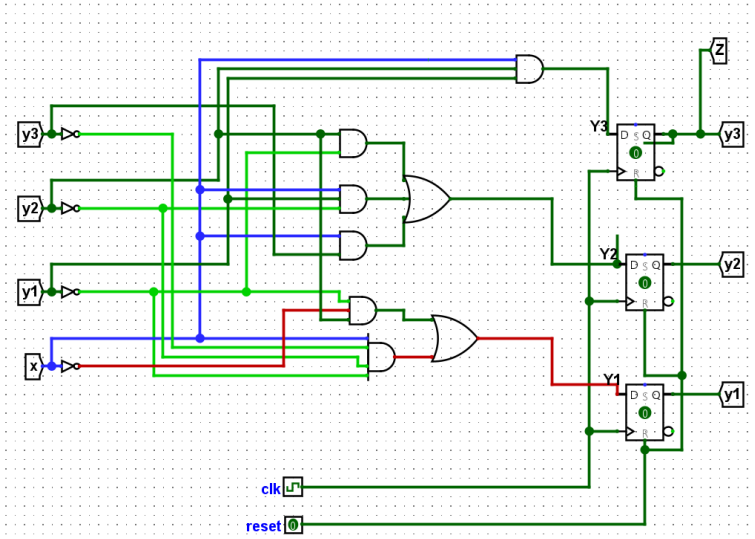


Y3(x, y3, y2, y1) = xy2y1

Y2(x, y3, y2, y1) = y2y1' + xy2'y1 + xy3

Y1(x, y3, y2, y1) = x'y2y1' + xy3'y2'y1'

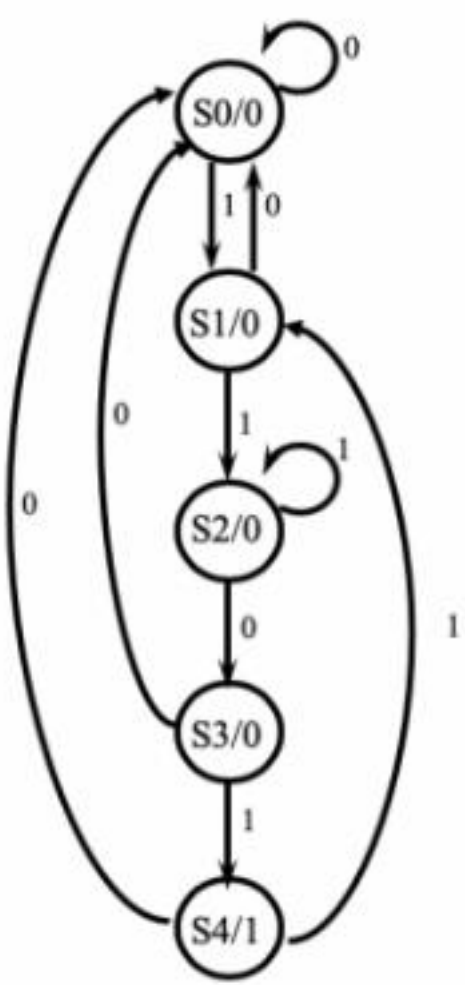
Z = y3



Sequence Detector : 1101 Moore

2. Non overlapping

Present State	Next State		Output
	X = 0	X = 1	Y
S0	S0	S1	0
S1	S0	S2	0
S2	S3	S2	0
S3	S0	S4	0
S4	S0	S1	1



Sequence Detector : 1101 Moore - Non overlapping

Present State	Next State		Output
	X = 0	X = 1	Y
S0	S0	S1	0
S1	S0	S2	0
S2	S3	S2	0
S3	S0	S4	0
S4	S0	S1	1

Present State	Next State		Output
	X = 0	X = 1	z
y3y2y1	Y3Y2Y1	Y3Y2Y1	
000	000	001	0
001	000	010	0
010	011	010	0
011	000	100	0
100	000	001	1

Sequence Detector : 1101 Moore - Non overlapping

Y3	y2,y1			
x,y3	00	01	11	10
00	0	0	0	0
01	0	-	-	-
11	0	-	-	-
10	0	0	1	0

Y2	y2,y1			
x,y3	00	01	11	10
00	0	0	0	1
01	0	-	-	-
11	0	-	-	-
10	0	1	0	1

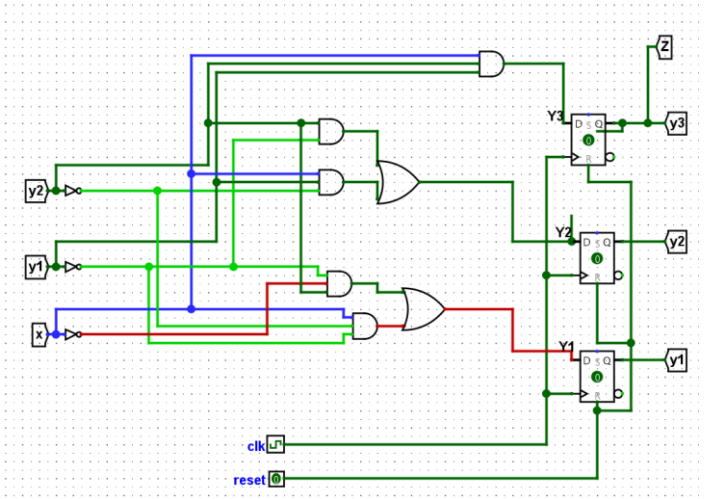
Y1	y2,y1			
x,y3	00	01	11	10
00	0	0	0	1
01	0	-	-	-
11	1	-	-	-
10	1	0	0	0

Y3(x, y3, y2, y1) = xy2y1

Y2(x, y3, y2, y1) = y2y1' + xy2'y1

Y1(x, y3, y2, y1) = x'y2y1' + xy2'y1'

Z = y3



Partitioning Minimization Procedure

states in each block are equivalent

Present state	Next state		Output
	w =0	w = 1	z
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0

Partitioning Minimization Procedure

P1 = (ABCDEFG)

Present state	Next state		Output
	w =0	w = 1	z
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0

Partitioning Minimization Procedure

P1 = (ABCDEFG)

P2 = (ABD)(CEFG)

Present state	Next state		Output
	w =0	w = 1	z
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0

Partitioning Minimization Procedure

P1 = (ABCDEFG)

P2 = (ABD)(CEFG)

P3 = (ABD)(CEG)(F)

Present state	Next state		Output
	w =0	w = 1	z
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0

Partitioning Minimization Procedure

P1 = (ABCDEFG)

P2 = (ABD)(CEFG)

P3 = (ABD)(CEG)(F)

P4 = (AD)(B)(CEG)(F)

Present state	Next state		Output
	w =0	w = 1	z
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0

Partitioning Minimization Procedure

(AD) => A

(B) => B

(CEG) => C

(F) => F

Present state	Next state		Output
	w =0	w = 1	z
A	B	C	1
B	A	F	1
C	F	C	0
F	C	A	0

출처 및 팀원별 기여도

S. Brown and Z. Vranesic, 'Fundamentals of Digital Logic with VHDL Design', 3rd Edition, McGraw Hill, 2009

https://www.geeksforgeeks.org/design-101-sequence-detector-mealy-machine/

https://yue-guo.com/2019/03/25/sequence-detector-1101-moore-machine-mealy-machine-overlapping-non-overlapping/

https://youtu.be/5wceSNz0TAE?si=zWT74xR3MIVi-ME8 **-WIT Solapur - Professional Learning Community**

- **기여도**

  서기환(50%), 전해찬(50%)