Computer Organization and Architecture

Qiben Yan

Department of Computer Science and Engineering
Michigan State University

January 26, 2026

Spring 2026

iClicker

How many minimum number of bits do we need to identify 6 types of bagels uniquely?

A 2

B 3

C 4

D 6

E 64

Truth Table

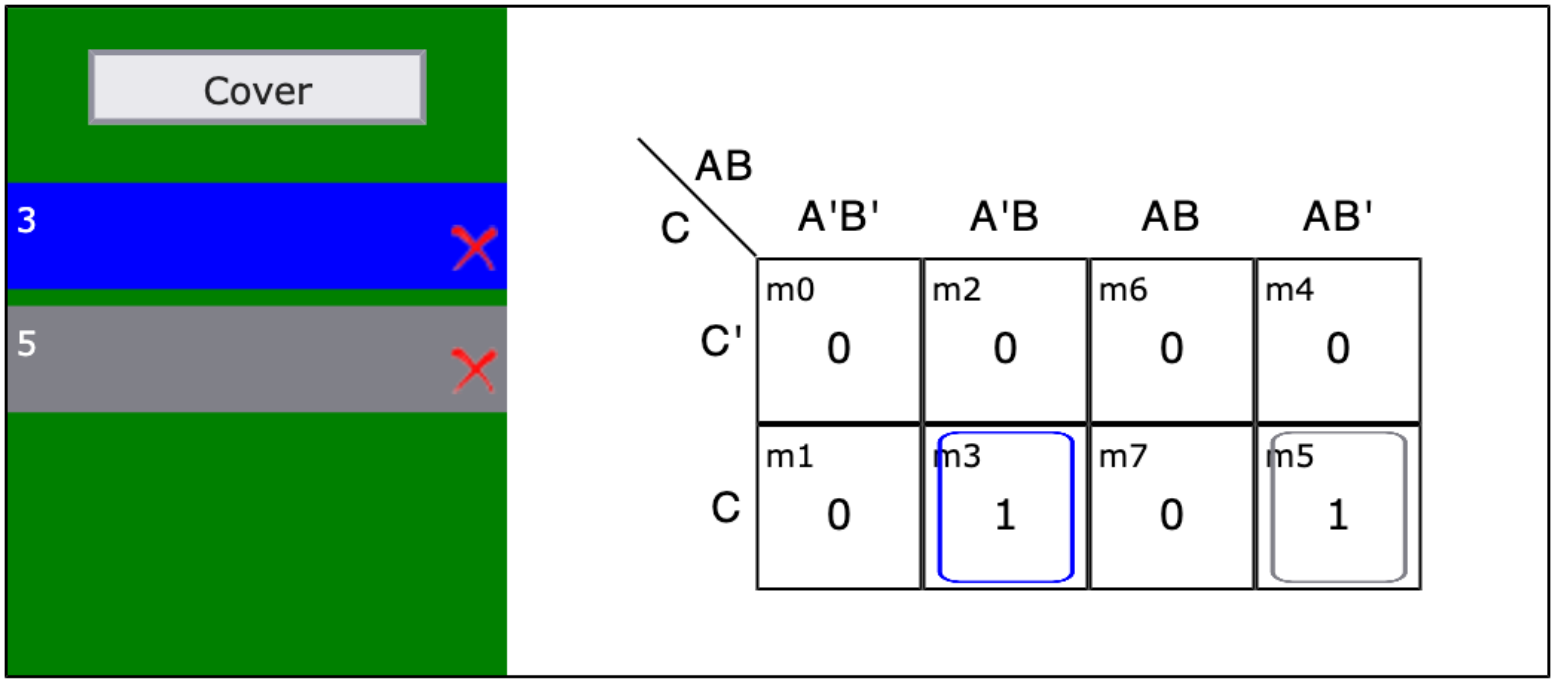
What's your favorite ?

Bagels	A	B	C	F(A, B, C)
Plain	0	0	0	0
Everything	0	0	1	0
Sesame	0	1	0	0
Cinnamon	0	1	1	1
Raisin	1	0	0	0
Jalapeno-Cheddar	1	0	1	1

Truth Table

Bagels	A	B	C	F(A, B, C)
Plain	0	0	0	0
Everything	0	0	1	0
Sesame	0	1	0	0
Cinnamon	0	1	1	1
Raisin	1	0	0	0
Jalapeno-Cheddar	1	0	1	1
	1	1	0	?
	1	1	1	?

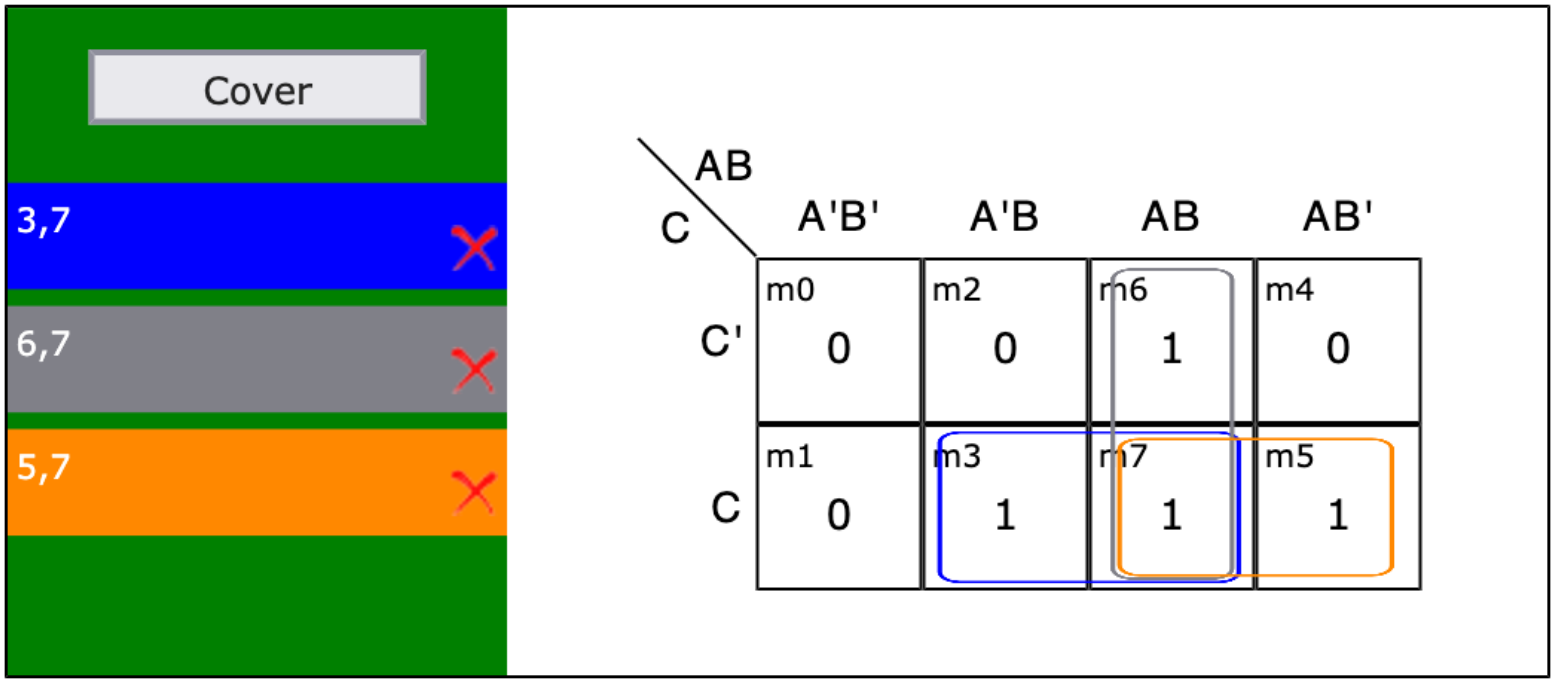
Truth Table



+ AB'C

A'BC

Truth Table



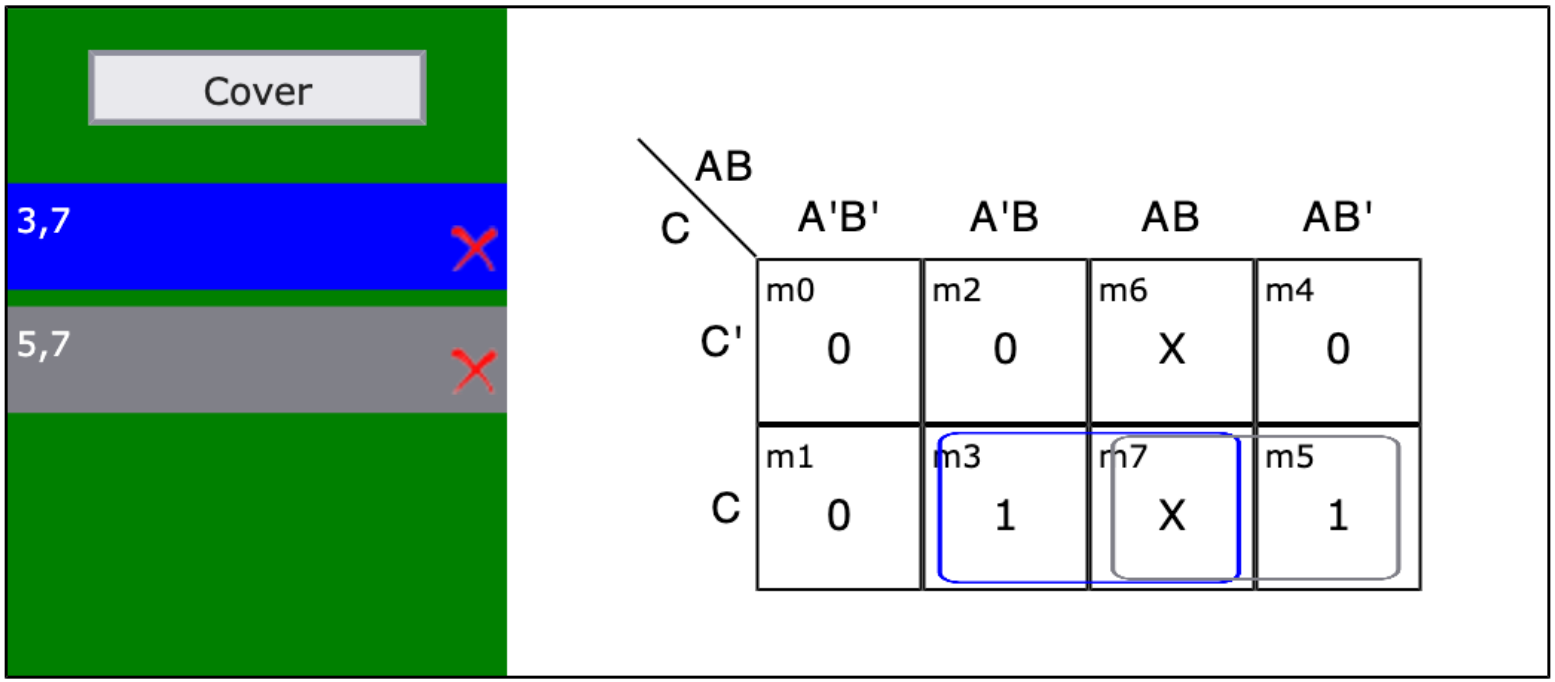
+ BC + AC

AB

Truth Table

Bagels	A	B	C	F(A, B, C)
Plain	0	0	0	0
Everything	0	0	1	0
Sesame	0	1	0	0
Cinnamon	0	1	1	1
Raisin	1	0	0	0
Jalapeno-Cheddar	1	0	1	1
	1	1	0	X
	1	1	1	X

Truth Table



+ BC

AC

K-Map: 3 variables

F(A, B, C) = m1, m3, m4, m5 and
don't care = m0, m7

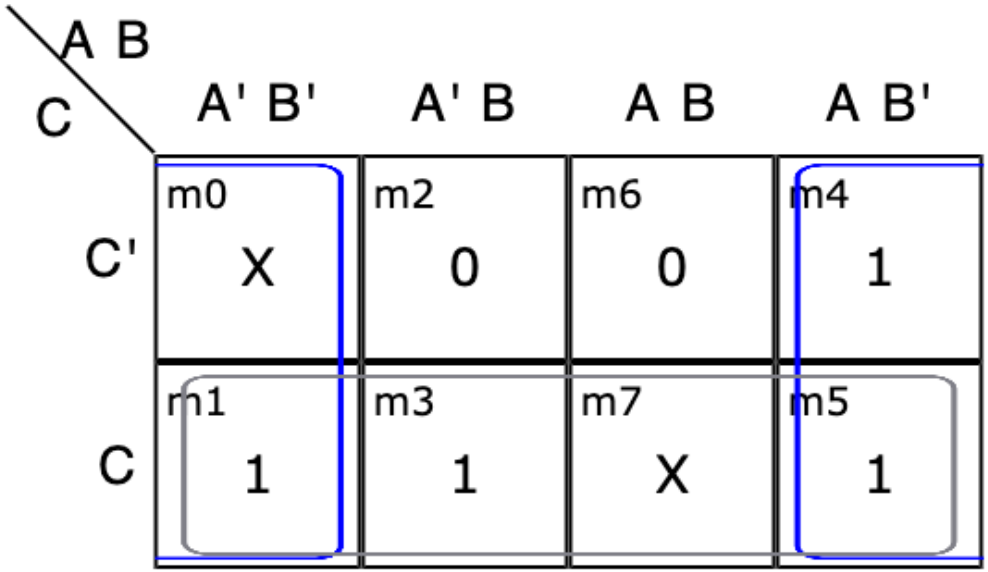
K-Map: 3 variables

F(A, B, C) = m1, m3, m4, m5 and
don't care = m0, m7

C \ A B	A' B'	A' B	A B	A B'
C'	m0 X	m2 0	m6 0	m4 1
C	m1 1	m3 1	m7 X	m5 1

K-Map: 3 variables

F(A, B, C) = m1, m3, m4, m5 and
don't care = m0, m7



F(A, B, C) = B' + C

K-Map: 3 variables

F(A, B, C) = m4, m5, m6, m7 and
don't care = m0, m2

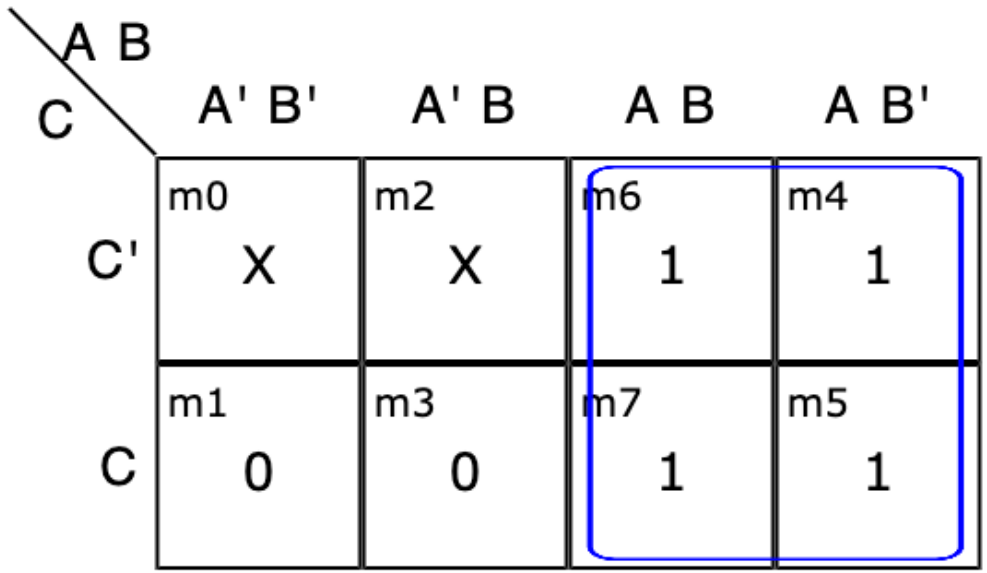
K-Map: 3 variables

F(A, B, C) = m4, m5, m6, m7 and
don't care = m0, m2

C \ A B	A' B'	A' B	A B	A B'
C'	m0 X	m2 X	m6 1	m4 1
C	m1 0	m3 0	m7 1	m5 1

K-Map: 3 variables

F(A, B, C) = m4, m5, m6, m7 and
don't care = m0, m2



F(A, B, C) = A

iClicker

F(A, B, C) = m3, m7

Don't care = m5, m6

C \ A B	A' B'	A' B	A B	A B'
C'	m0 0	m2 0	m6 X	m4 0
C	m1 0	m3 1	m7 1	m5 X

K-Map: 4 variables

F(A, B, C, D) = m2, m4, m6, m8, m11, m12, m13, m15
Don't care = m0, m9, m10

K-Map: 4 variables

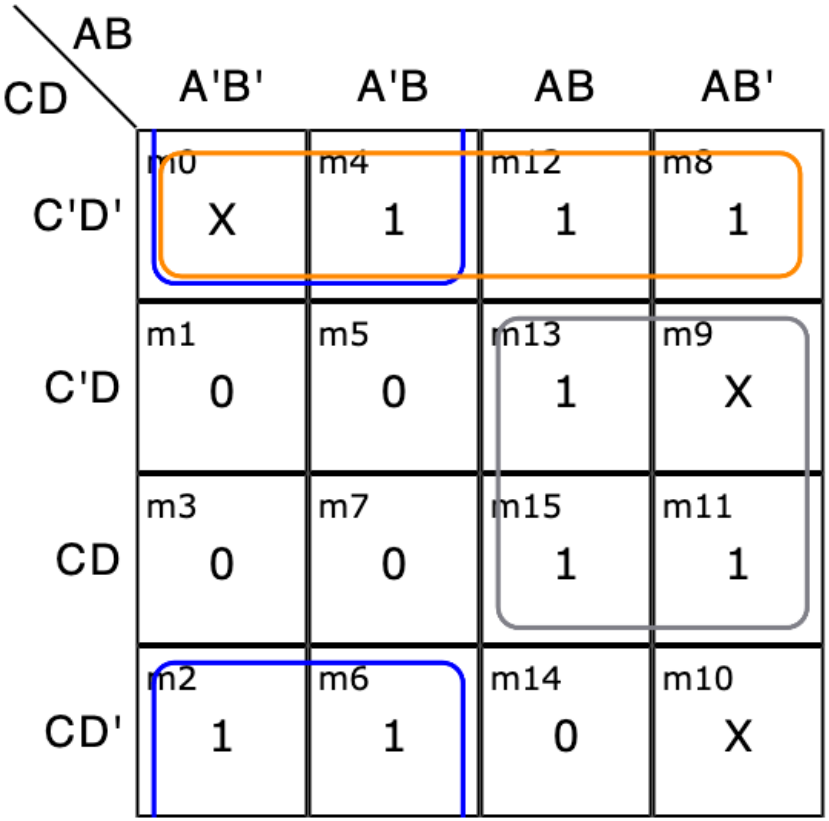
F(A, B, C, D) = m2, m4, m6, m8, m11, m12, m13, m15
Don't care = m0, m9, m10

CD \ AB	A'B'	A'B	AB	AB'
C'D'	m0 X	m4 1	m12 1	m8 1
C'D	m1 0	m5 0	m13 1	m9 X
CD	m3 0	m7 0	m15 1	m11 1
CD'	m2 1	m6 1	m14 0	m10 X

K-Map: 4 variables

F(A, B, C, D) = m2, m4, m6, m8, m11, m12, m13, m15

Don't care = m0, m9, m10



F(A, B, C, D) = A'D' + AD + C'D'

iClicker

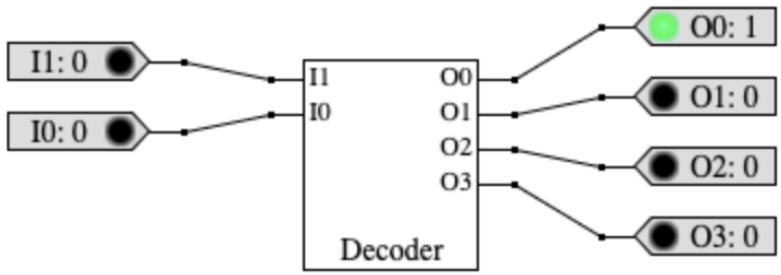
F(A, B, C, D) = m0, m1, m2, m3, m5, m12, m13, m14

Don't care = m4, m8, m9, m10, m15

CD \ AB	A'B'	A'B	AB	AB'
C'D'	m0 1	m4 X	m12 1	m8 X
C'D	m1 1	m5 1	m13 1	m9 X
CD	m3 1	m7 0	m15 X	m11 0
CD'	m2 1	m6 0	m14 1	m10 X

Decoder

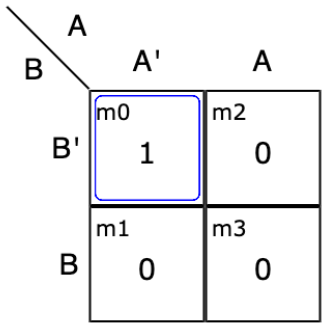
- A decoder recognizes which encoded value is the input to the circuit
- Input: n
- Output: 2^n
- One of the 2^n outputs is asserted, based on which encoded value is input.



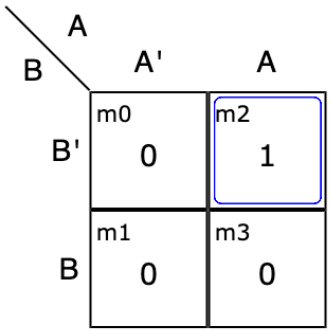
Decoder

I_1	I_0	O_3	O_2	O_1	O_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

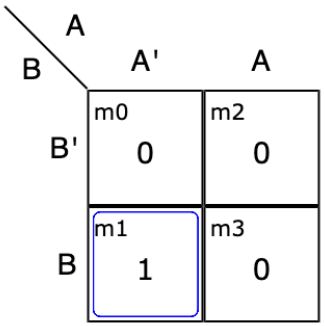
Decoder



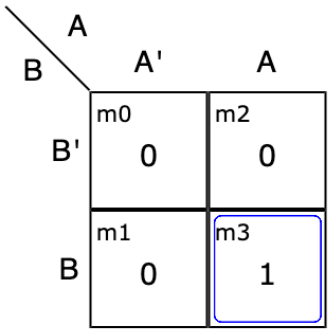
O_0 = A'B' = $I_1' I_0'$



O_2 = AB' = $I_1 I_0'$

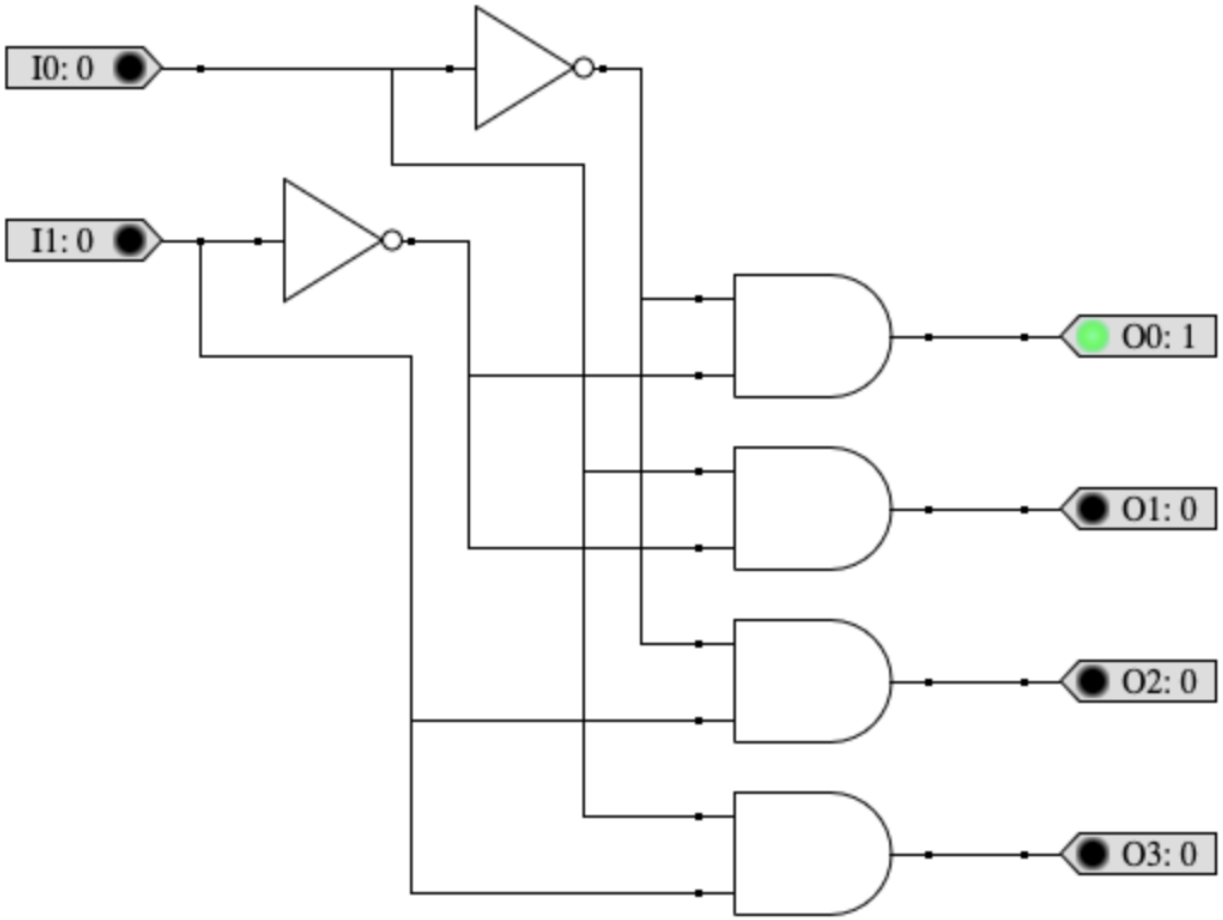


O_1 = A'B = $I_1' I_0$



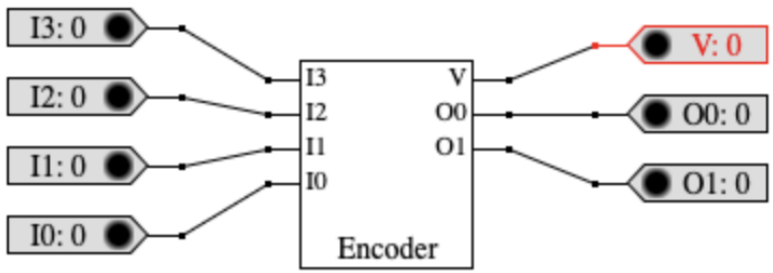
O_3 = AB = $I_1 I_0$

Decoder



Encoder

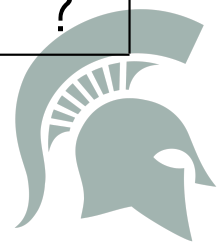
- Input: 2^n
- Output: n



Encoder

I_3	I_2	I_1	I_0		O_1	O_0
0	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		?	?

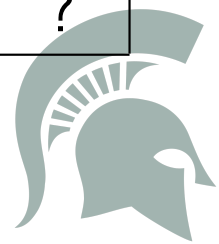
I_3	I_2	I_1	I_0		O_1	O_0
1	0	0	0		?	?
1	0	0	1		?	?
1	0	1	0		?	?
1	0	1	1		?	?
1	1	0	0		?	?
1	1	0	1		?	?
1	1	1	0		?	?
1	1	1	1		?	?



Encoder

I_3	I_2	I_1	I_0		O_1	O_0
0	0	0	0		?	?
0	0	0	1		0	0
0	0	1	0		0	1
0	0	1	1		?	?
0	1	0	0		1	0
0	1	0	1		?	?
0	1	1	0		?	?
0	1	1	1		?	?

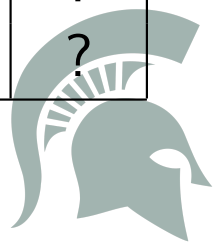
I_3	I_2	I_1	I_0		O_1	O_0
1	0	0	0		1	1
1	0	0	1		?	?
1	0	1	0		?	?
1	0	1	1		?	?
1	1	0	0		?	?
1	1	0	1		?	?
1	1	1	0		?	?
1	1	1	1		?	?



Encoder

I_3	I_2	I_1	I_0	V	O_1	O_0
0	0	0	0	0	?	?
0	0	0	1	1	0	0
0	0	1	0	1	0	1
0	0	1	1	0	?	?
0	1	0	0	1	1	0
0	1	0	1	0	?	?
0	1	1	0	0	?	?
0	1	1	1	0	?	?

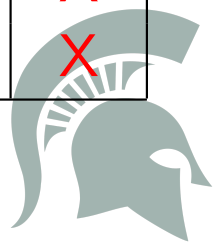
I_3	I_2	I_1	I_0	V	O_1	O_0
1	0	0	0	1	1	1
1	0	0	1	0	?	?
1	0	1	0	0	?	?
1	0	1	1	0	?	?
1	1	0	0	0	?	?
1	1	0	1	0	?	?
1	1	1	0	0	?	?
1	1	1	1	0	?	?



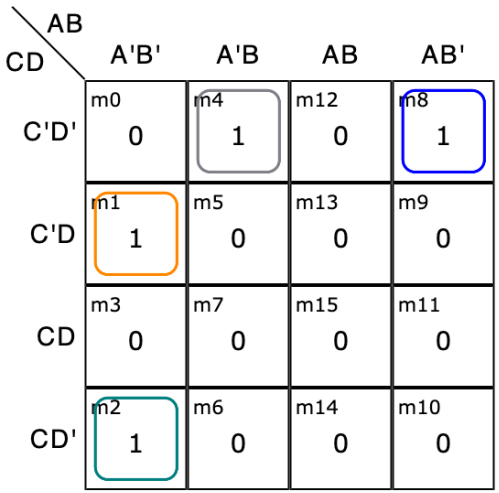
Encoder

I_3	I_2	I_1	I_0	V	O_1	O_0
0	0	0	0	0	X	X
0	0	0	1	1	0	0
0	0	1	0	1	0	1
0	0	1	1	0	X	X
0	1	0	0	1	1	0
0	1	0	1	0	X	X
0	1	1	0	0	X	X
0	1	1	1	0	X	X

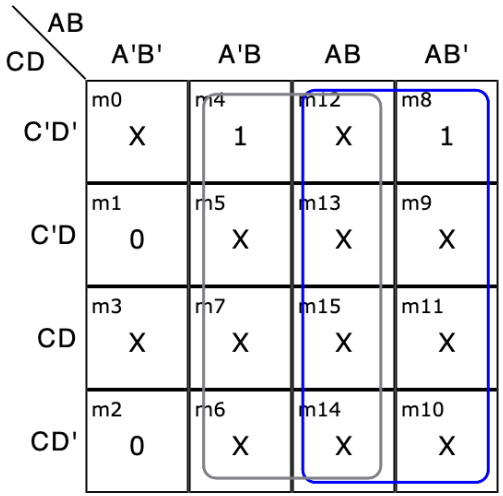
I_3	I_2	I_1	I_0	V	O_1	O_0
1	0	0	0	1	1	1
1	0	0	1	0	X	X
1	0	1	0	0	X	X
1	0	1	1	0	X	X
1	1	0	0	0	X	X
1	1	0	1	0	X	X
1	1	1	0	0	X	X
1	1	1	1	0	X	X



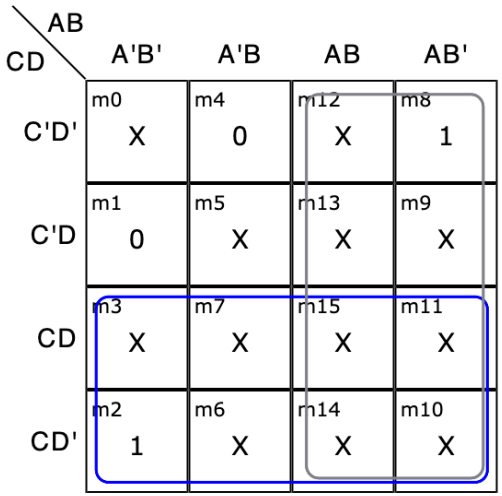
Encoder



$$V = AB'C'D' + A'BC'D' + A'B'CD' + A'B'C'D =$$
$$I_3 I_2' I_1' I_0' + I_3' I_2 I_1' I_0' + I_3' I_2' I_1 I_0' + I_3' I_2' I_1' I_0$$

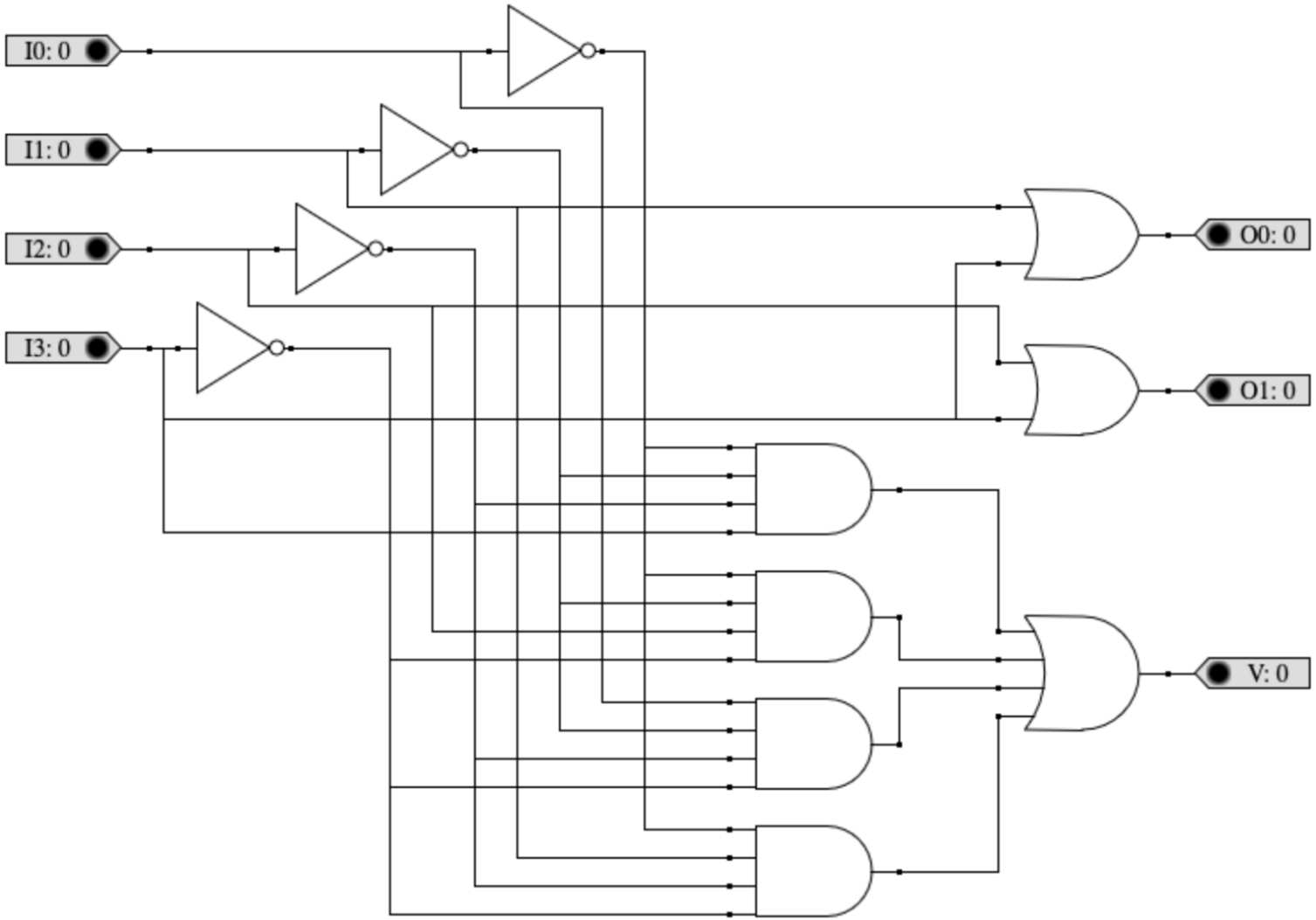


$$O_1 = A + B = I_3 + I_2$$



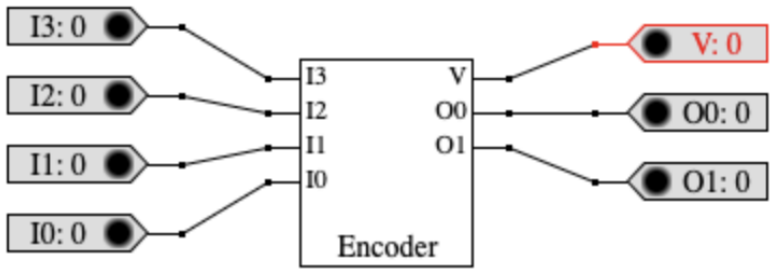
$$O_0 = A + C = I_3 + I_1$$

Encoder



Priority Encoder

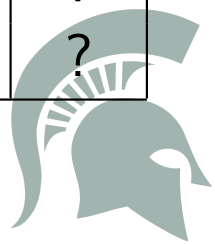
- Piority Encoder acts like an encoder, but there is a priority established to handle more than one input being asserted
- Input: 2^n
- Output: n
- One or more of the 2^n inputs is asserted, highest priority value is encoded
- Priority Order: $I_3 > I_2 > I_1 > I_0$



Priority Encoder

I_3	I_2	I_1	I_0	V	O_1	O_0
0	0	0	0		?	?
0	0	0	1	1	0	0
0	0	1	0	1	0	1
0	0	1	1		?	?
0	1	0	0	1	1	0
0	1	0	1		?	?
0	1	1	0		?	?
0	1	1	1		?	?

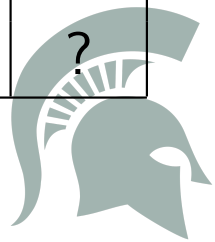
I_3	I_2	I_1	I_0	V	O_1	O_0
1	0	0	0	1	1	1
1	0	0	1		?	?
1	0	1	0		?	?
1	0	1	1		?	?
1	1	0	0		?	?
1	1	0	1		?	?
1	1	1	0		?	?
1	1	1	1		?	?



Priority Encoder

I_3	I_2	I_1	I_0	V	O_1	O_0
0	0	0	0		?	?
0	0	0	1	1	0	0
0	0	1	0	1	0	1
0	0	1	1	1	0	1
0	1	0	0	1	1	0
0	1	0	1		?	?
0	1	1	0		?	?
0	1	1	1		?	?

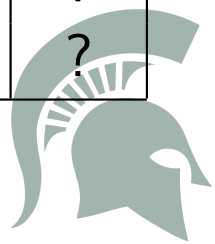
I_3	I_2	I_1	I_0	V	O_1	O_0
1	0	0	0	1	1	1
1	0	0	1		?	?
1	0	1	0		?	?
1	0	1	1		?	?
1	1	0	0		?	?
1	1	0	1		?	?
1	1	1	0		?	?
1	1	1	1		?	?



Priority Encoder

I_3	I_2	I_1	I_0	V	O_1	O_0
0	0	0	0		?	?
0	0	0	1	1	0	0
0	0	1	0	1	0	1
0	0	1	1	1	0	1
0	1	0	0	1	1	0
0	1	0	1	1	1	0
0	1	1	0	1	1	0
0	1	1	1	1	1	0

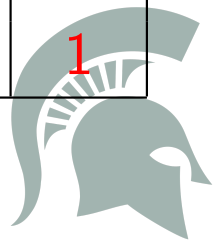
I_3	I_2	I_1	I_0	V	O_1	O_0
1	0	0	0	1	1	1
1	0	0	1		?	?
1	0	1	0		?	?
1	0	1	1		?	?
1	1	0	0		?	?
1	1	0	1		?	?
1	1	1	0		?	?
1	1	1	1		?	?



Priority Encoder

I_3	I_2	I_1	I_0	V	O_1	O_0
0	0	0	0		?	?
0	0	0	1	1	0	0
0	0	1	0	1	0	1
0	0	1	1	1	0	1
0	1	0	0	1	1	0
0	1	0	1	1	1	0
0	1	1	0	1	1	0
0	1	1	1	1	1	0

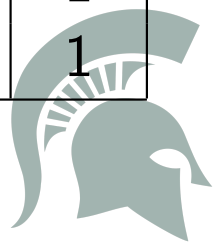
I_3	I_2	I_1	I_0	V	O_1	O_0
1	0	0	0	1	1	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1



Priority Encoder

I_3	I_2	I_1	I_0	V	O_1	O_0
0	0	0	0	0	X	X
0	0	0	1	1	0	0
0	0	1	0	1	0	1
0	0	1	1	1	0	1
0	1	0	0	1	1	0
0	1	0	1	1	1	0
0	1	1	0	1	1	0
0	1	1	1	1	1	0

I_3	I_2	I_1	I_0	V	O_1	O_0
1	0	0	0	1	1	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

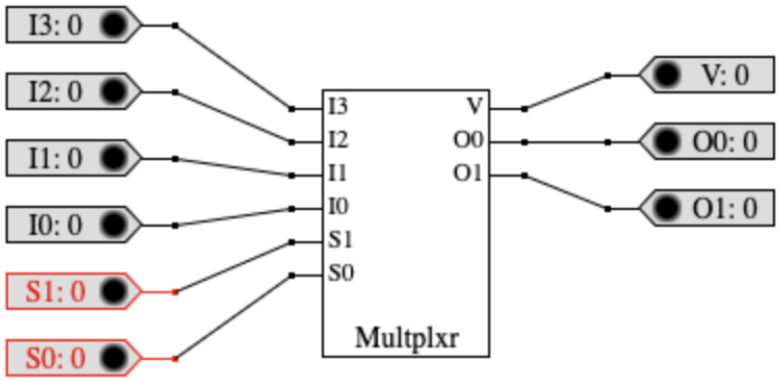


Priority Encoder

Expression and Circuit: Quiz !

Multiplexer

- A multiplexer is used to select one input (out of several inputs) to be the circuit output
- Input: 2^n
- Selection input: n
- Output: 1
- The n selection bits determine which of the 2^n data inputs is routed to the output



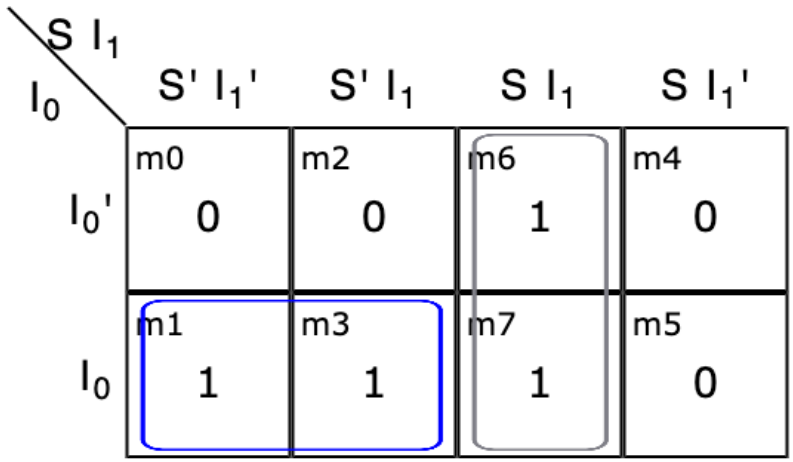
2-to-1 Multiplexer

S	O
0	I_0
1	I_1

S	I_1	I_0	O
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

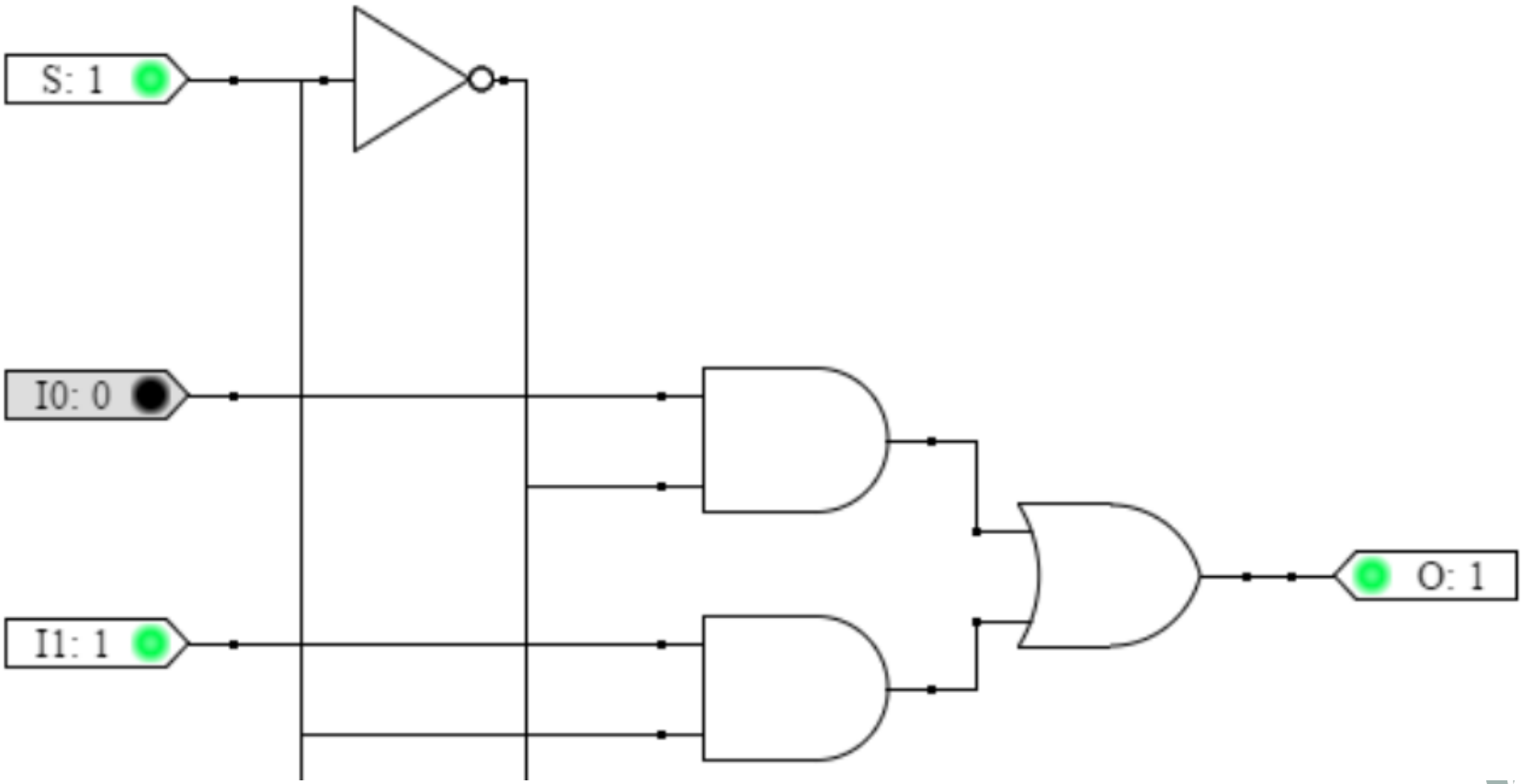
2-to-1 Multiplexer

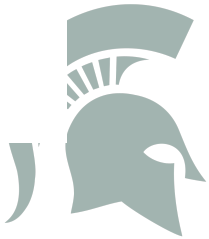
S	I_1	I_0	O
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



$SI_1 + S'I_0$

2-to-1 Multiplexer





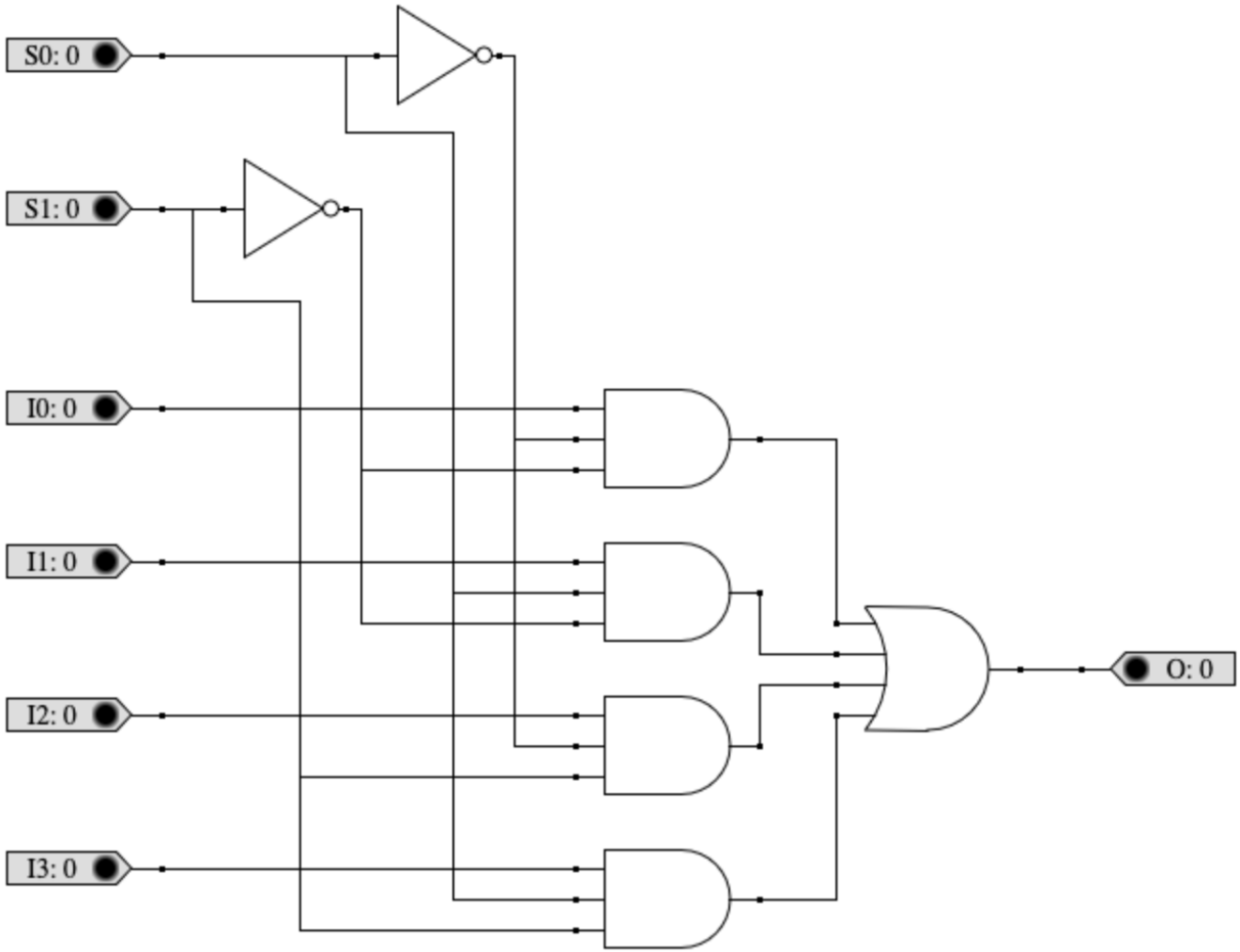
4-to-1 Multiplexer

S_1	S_0	O
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

4-to-1 Multiplexer

6-inputs, so the characteristic table will contain $2^6 = 64$ entries

4-to-1 Multiplexer



Demultiplexer

S_1	S_0	O_3	O_2	O_1	O_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0