UNIVERSALITY

UNIVERSAL SET

Is it possible to implement ALL the possible Boolean functions using NOT, AND, OR, NAND, NOR? Yes!

UNIVERSAL SET

What if we are not given some!
What if some are very costly! E.g., NOT
Can we reduce this set? E.g., building NOT by NAND/ NOR

SET	UNIVERSAL SET
{NOT, AND, OR, NAND, NOR}	Yes! (a Full Set)
{NOT, AND, OR}	Yes, (a Full Set) NAND = AND-NOT NOR = OR-NOT
{NOT, AND}	If we could design OR
{NOT, OR}	If we could design AND
{NOT}	If we could design AND, OR
{AND}	If we could design NOT, OR
{OR}	If we could design NOT, AND
{NAND}	If we could design NOT, AND, OR
{NOR}	If we could design NOT, AND, OR

SET	UNIVERSAL SET
{NOT, AND, OR, NAND, NOR}	Yes! (a Full Set)
{NOT, AND, OR}	Yes, (a Full Set) NAND = AND-NOT NOR = OR-NOT
{NOT, AND}	If we could design OR (a Complete Set)
{NOT, OR}	If we could design AND (a Complete Set)
{NOT}	If we could design AND, OR
{AND}	If we could design NOT, OR
{OR}	If we could design NOT, AND
{NAND}	If we could design NOT, AND, OR
{NOR}	If we could design NOT, AND, OR

UNIVERSAL SET
{NOT, AND}

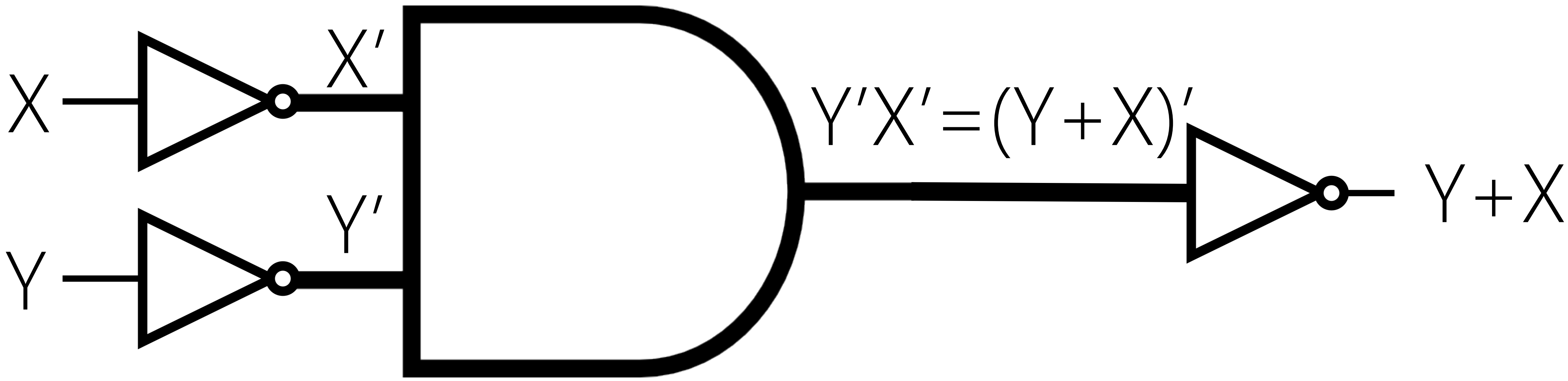
Augustus De Morgan (1806–1871) Mathematician Logician

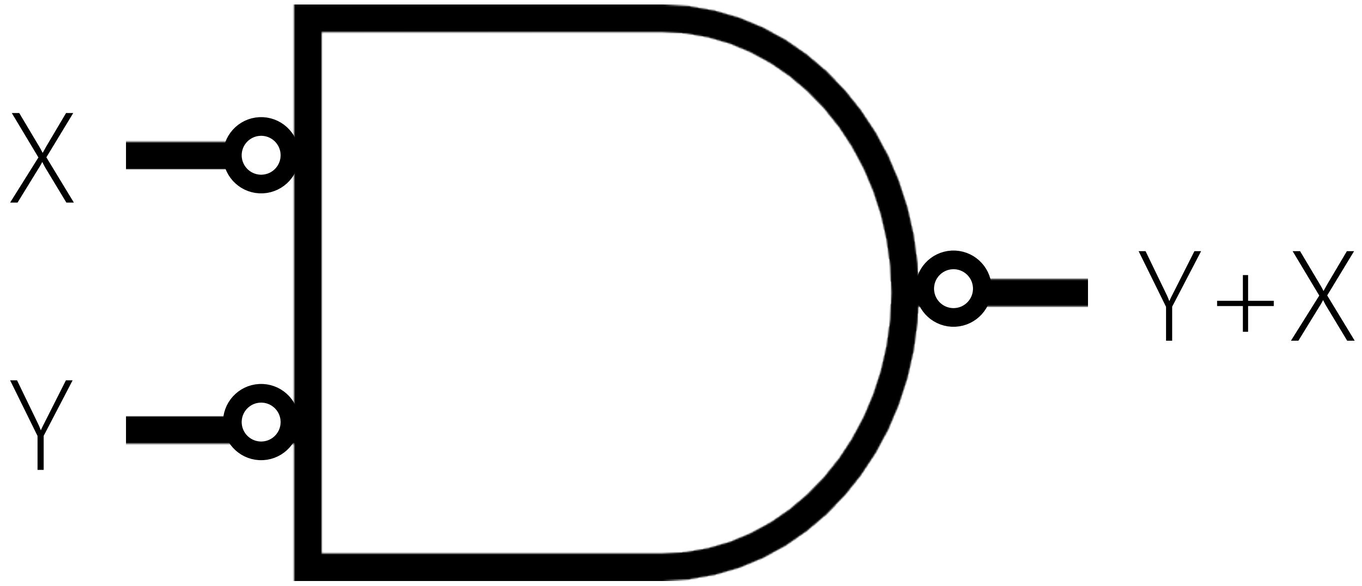
DE MORGAN'S LAWS

- $(Y+X)' = Y'X'$

$$((Y+X)')' = (Y'X')'$$

$$Y+X = (Y'X')'$$


X
Y
X′
Y′
Y′X′=(Y+X)′
Y+X



OR

SET	UNIVERSAL SET
{NOT, AND, OR, NAND, NOR}	Yes! (a Full Set)
{NOT, AND, OR}	Yes, (a Full Set) NAND = AND-NOT NOR = OR-NOT
{NOT, AND}	If we could design OR (a Complete Set)
{NOT, OR}	If we could design AND (a Complete Set)
{NOT}	If we could design AND, OR
{AND}	If we could design NOT, OR
{OR}	If we could design NOT, AND
{NAND}	If we could design NOT, AND, OR
{NOR}	If we could design NOT, AND, OR

UNIVERSAL SET
{NOT, OR}

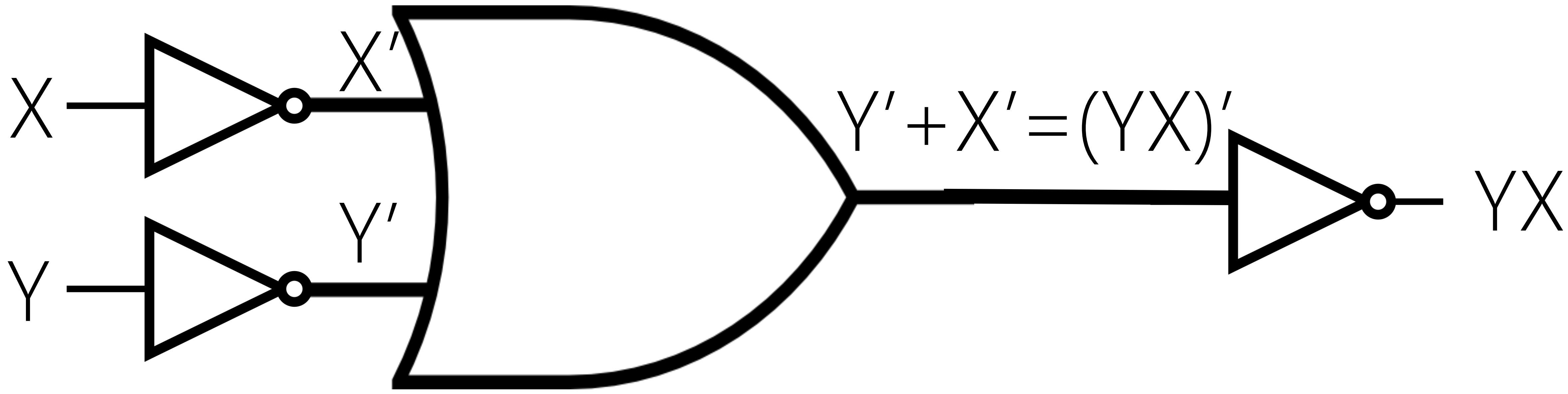
Augustus De Morgan
(1806–1871)
Mathematician
Logician

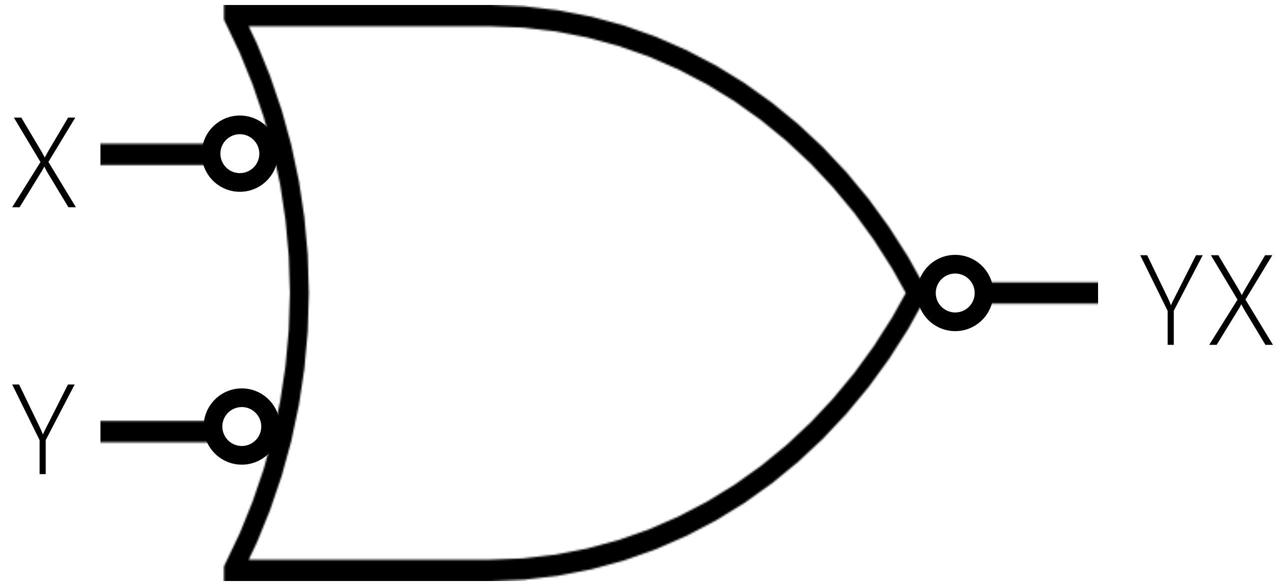
DE MORGAN'S LAWS

- $Y'+X' = (YX)'$

$$(Y'+X')' = ((YX)')'$$

$$(Y'+X')' = YX$$


X
Y
X′
Y′
Y′+X′=(YX)′
YX

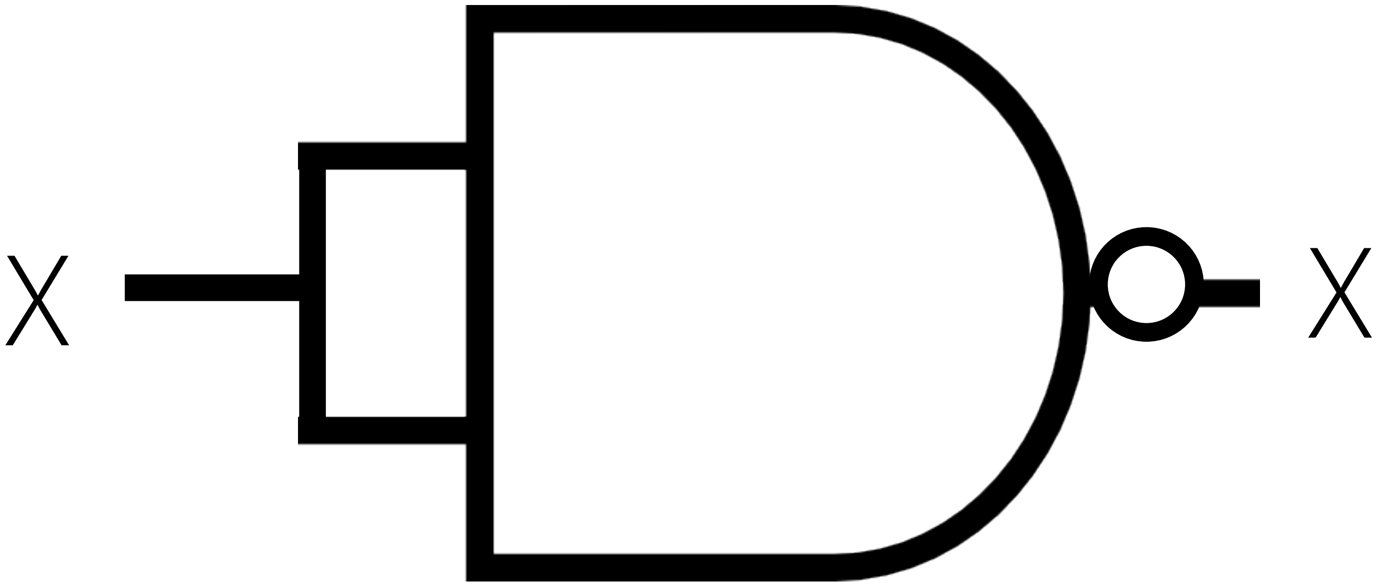


AND

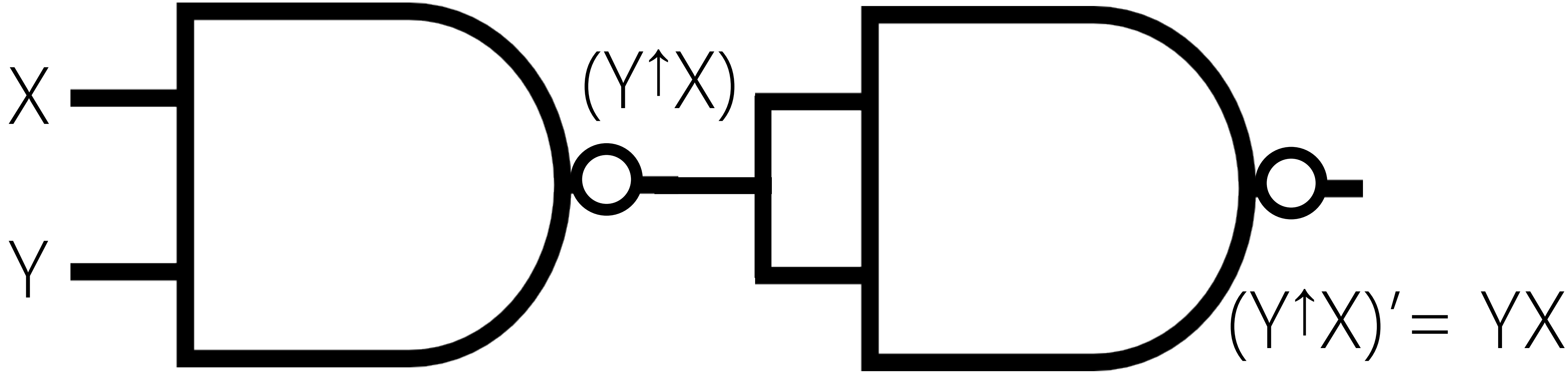
SET	UNIVERSAL SET
{NOT, AND, OR, NAND, NOR}	Yes! (a Full Set)
{NOT, AND, OR}	Yes, (a Full Set) NAND = AND-NOT NOR = OR-NOT
{NOT, AND}	If we could design OR (a Complete Set)
{NOT, OR}	If we could design AND (a Complete Set)
{NOT}	If we could design AND, OR
{AND}	If we could design NOT, OR
{OR}	If we could design NOT, AND
{NAND}	If we could design NOT, AND, OR
{NOR}	If we could design NOT, AND, OR

UNIVERSAL GATE
{NAND}

NOT ► $(XX)' = (X \uparrow X) = X'$



AND►NOT (NAND) = $((Y{\uparrow}X))' = YX$



OR ► DE MORGAN'S LAW

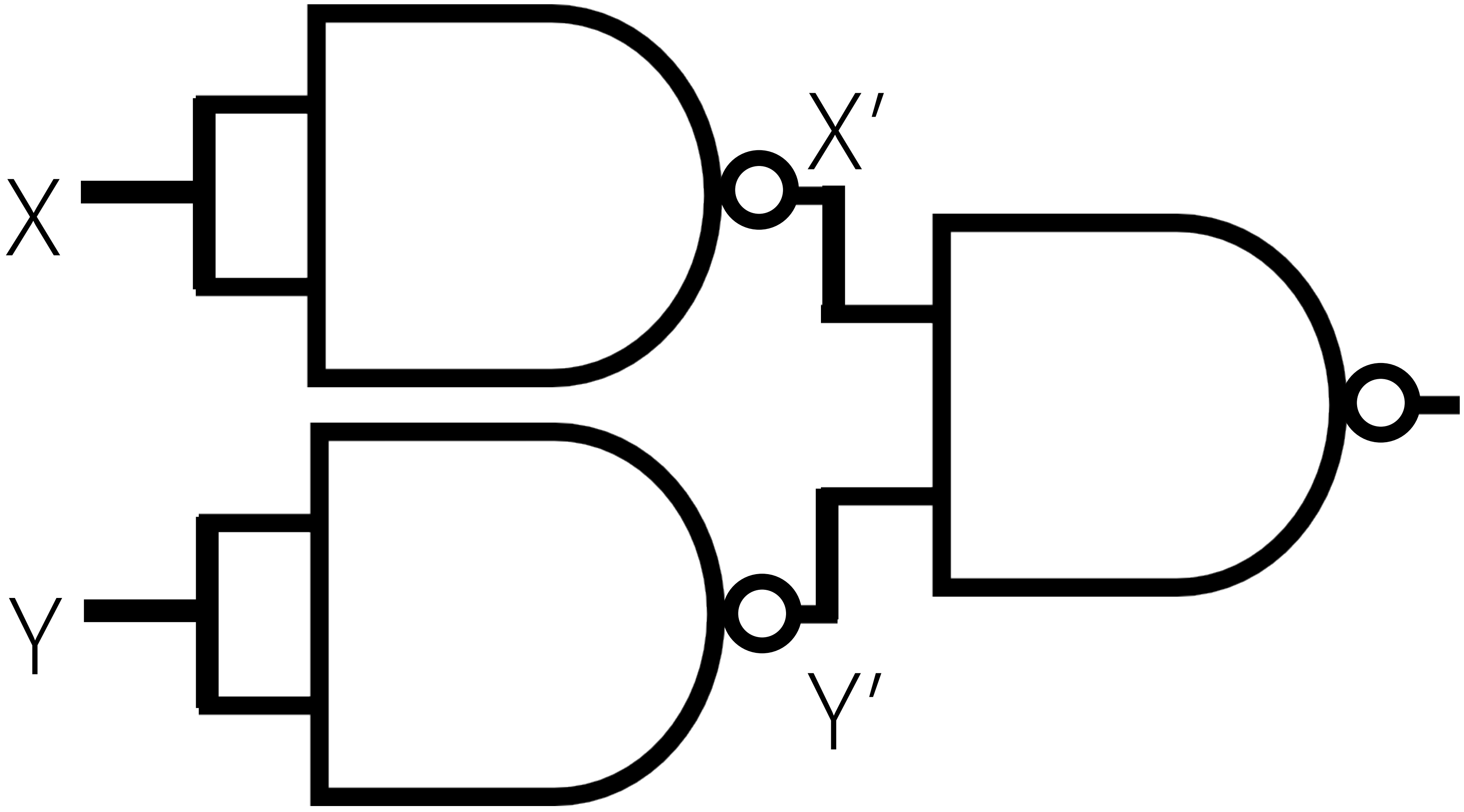
$$(Y+X)' = Y'X'$$

$$((Y+X)')' = (Y'X')'$$

$$Y + X = (Y'X')'$$

$$Y + X = Y' \uparrow X'$$

OR: DE MORGAN'S LAW



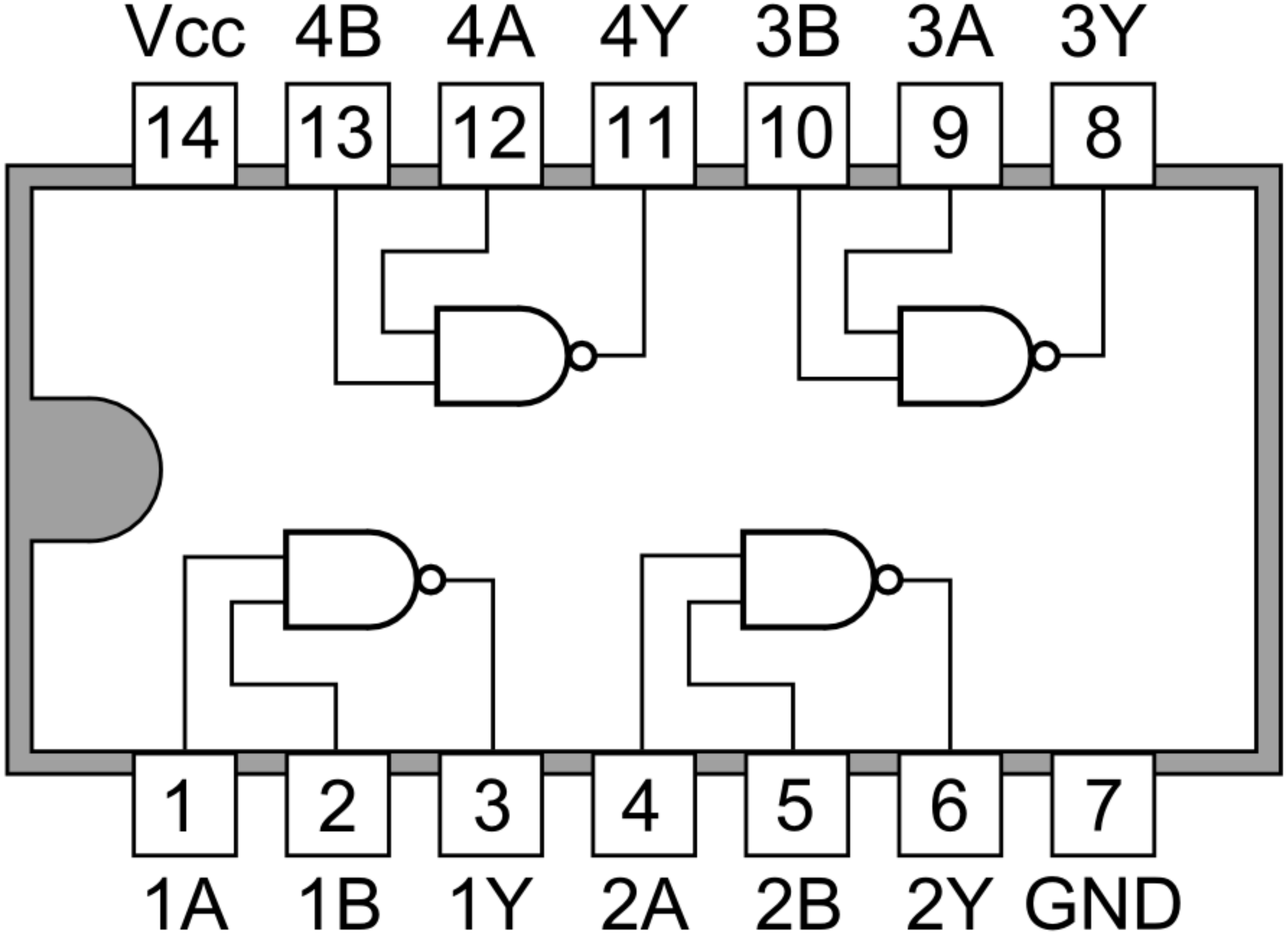
$(Y'X')' =$

$Y + X =$

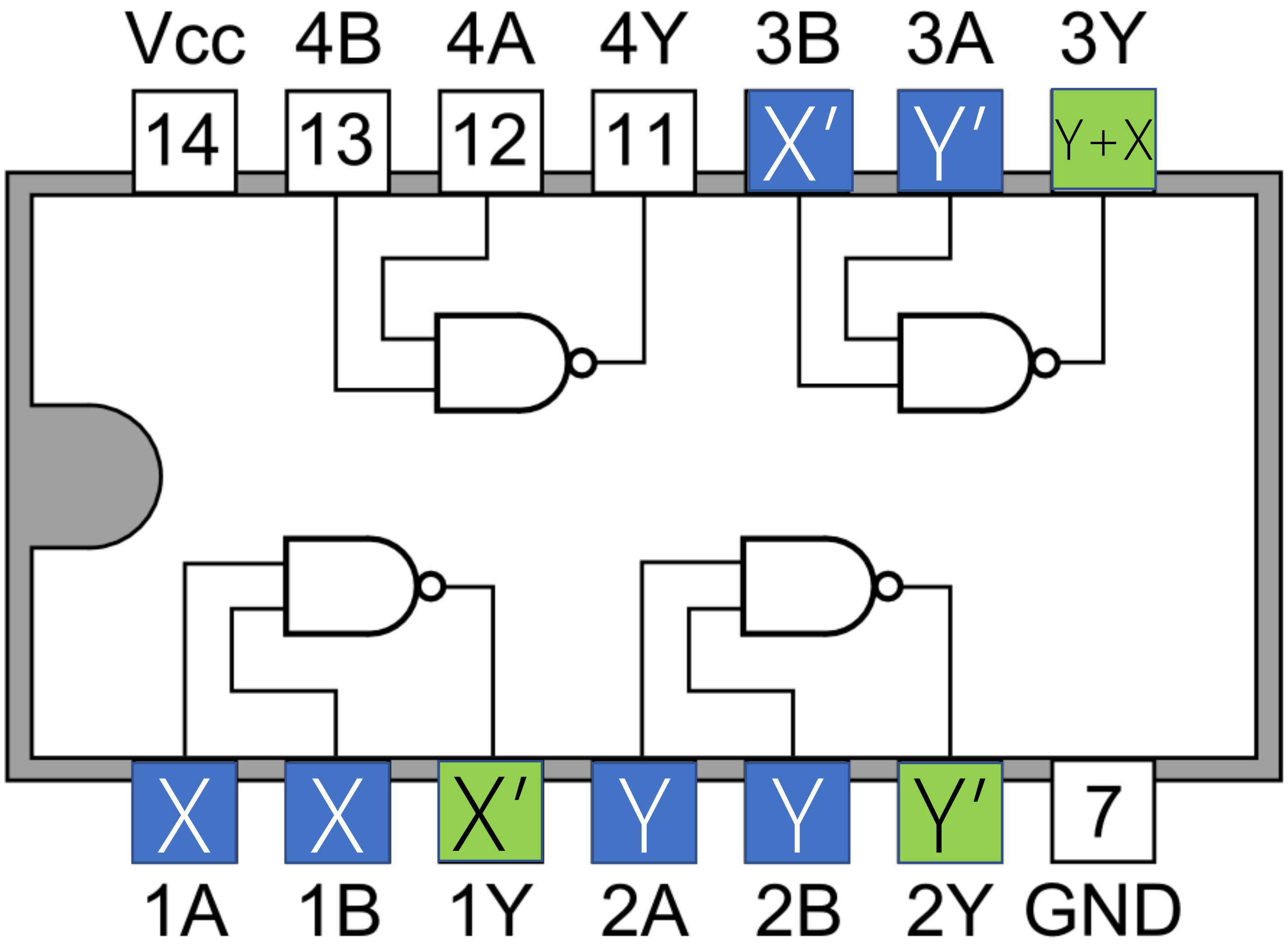
$Y' \uparrow X'$

7400 Quad 2-input NAND Gates

https://commons.wikimedia.org/wiki/7400_series_overview



7400 Quad 2-input NAND Gates



SET	UNIVERSAL SET
{NOT, AND, OR, NAND, NOR}	Yes! (a Full Set)
{NOT, AND, OR}	Yes, (a Full Set) NAND = AND-NOT NOR = OR-NOT
{NOT, AND}	If we could design OR (a Complete Set)
{NOT, OR}	If we could design AND (a Complete Set)
{NOT}	If we could design AND, OR
{AND}	If we could design NOT, OR
{OR}	If we could design NOT, AND
{NAND}	If we could design NOT, AND, OR
{NOR}	If we could design NOT, AND, OR

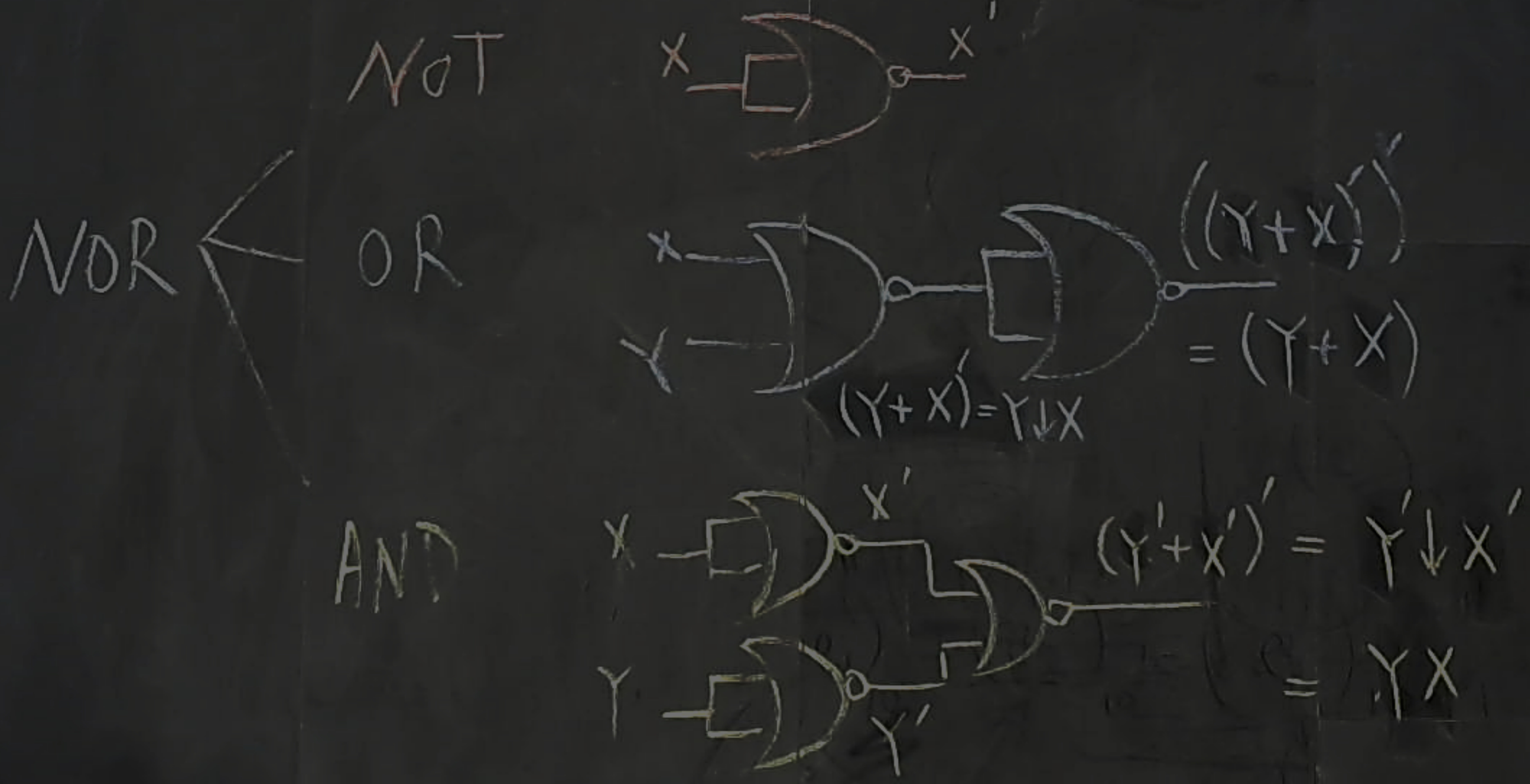
UNIVERSAL GATE
{NOR}

NOT ► $(X+X)' = (XX) = X'$

OR ► NOT (NOR)

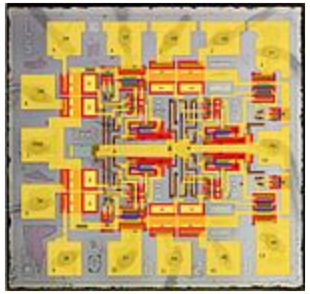
AND ► DE MORGAN'S LAW

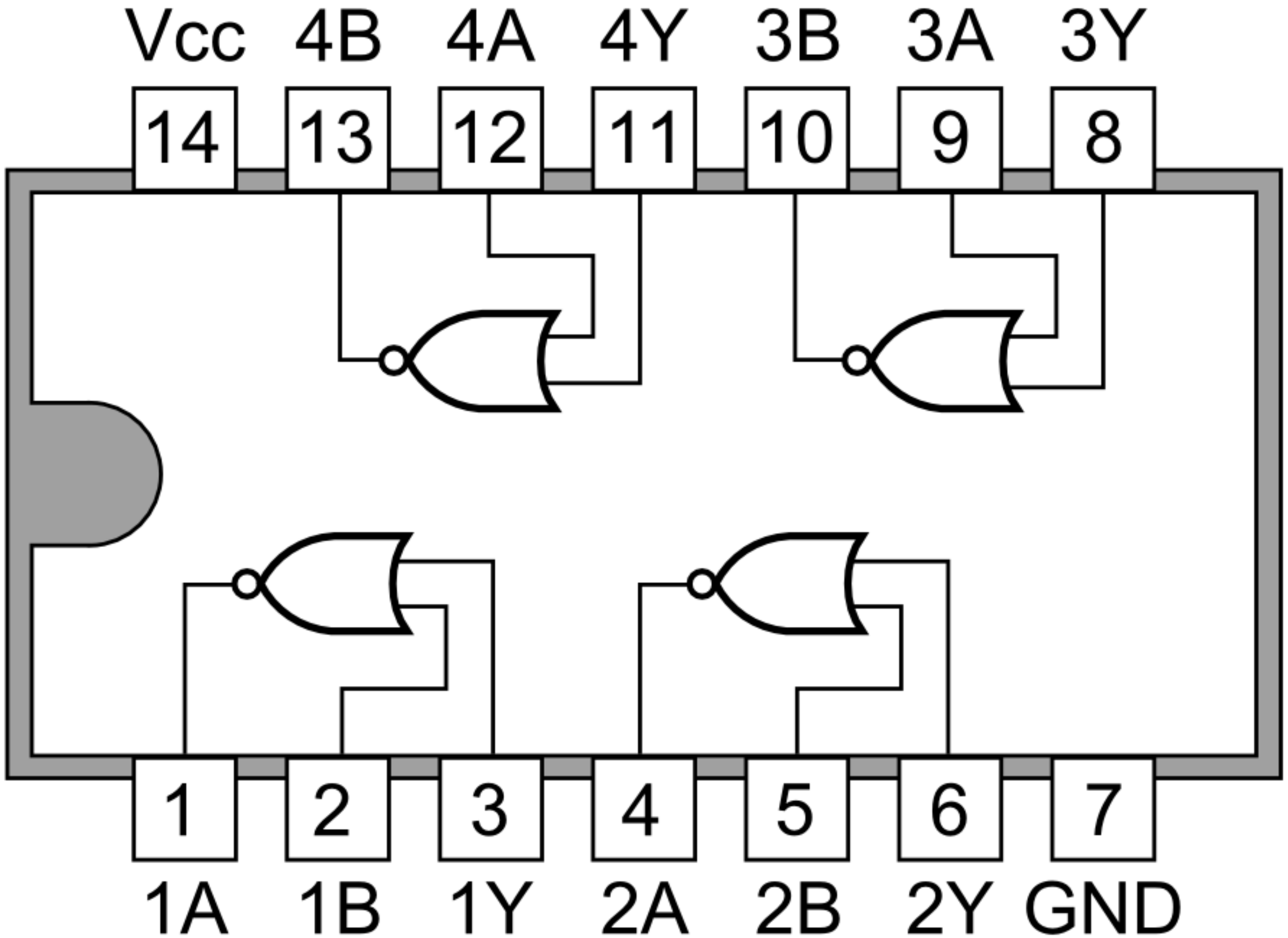
$(Y'+X')' = YX = (Y' \downarrow X')$


NOR
NOT
X
X'
OR
X
Y
((Y+X)')'
= (Y+X)
(Y+X)' = Y↓X
AND
X
X'
(Y'+X')' = Y'↓X'
= YX
Y
Y'

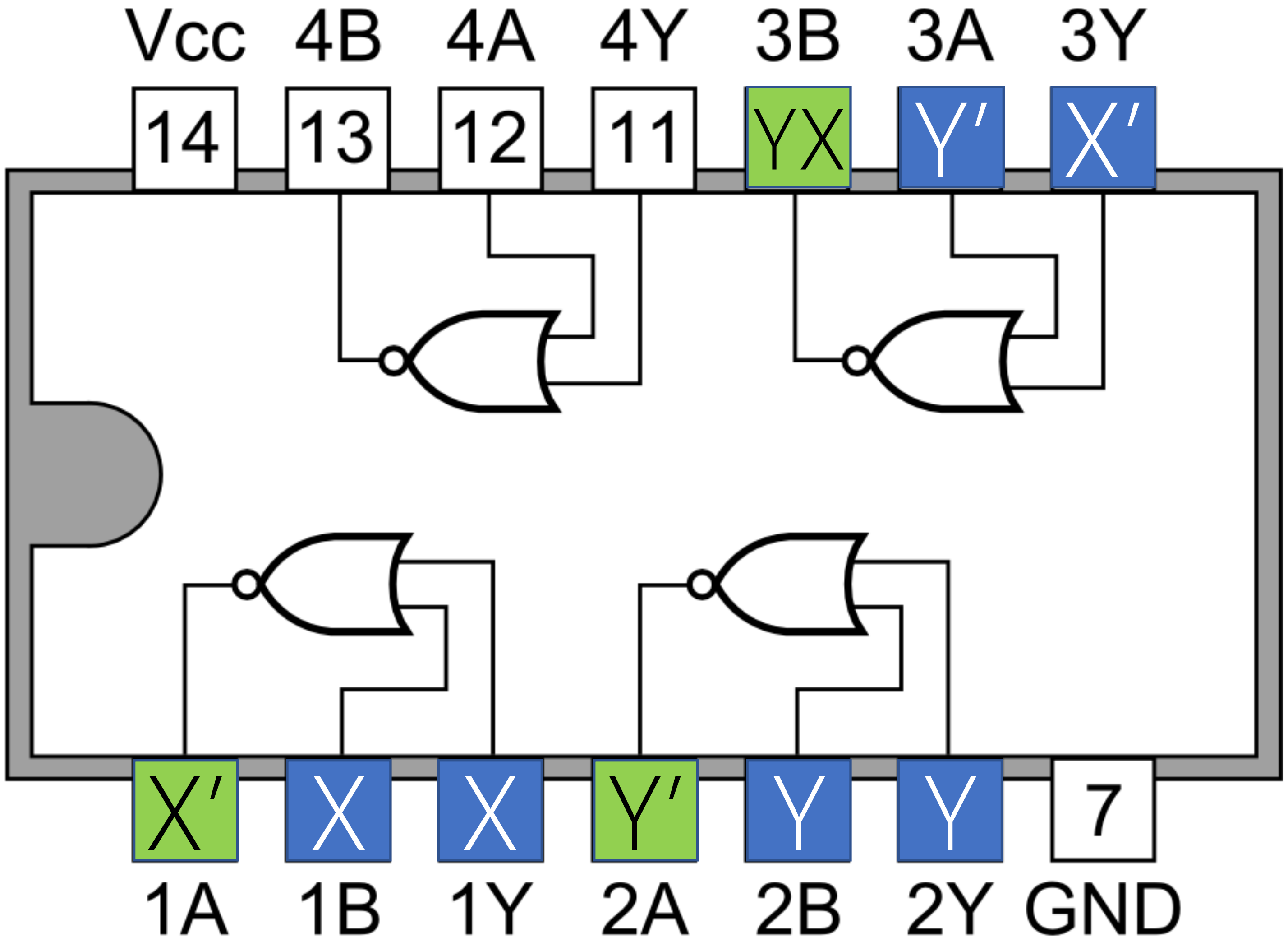
7402 Quad 2-input NOR Gates

https://commons.wikimedia.org/wiki/7400_series_overview



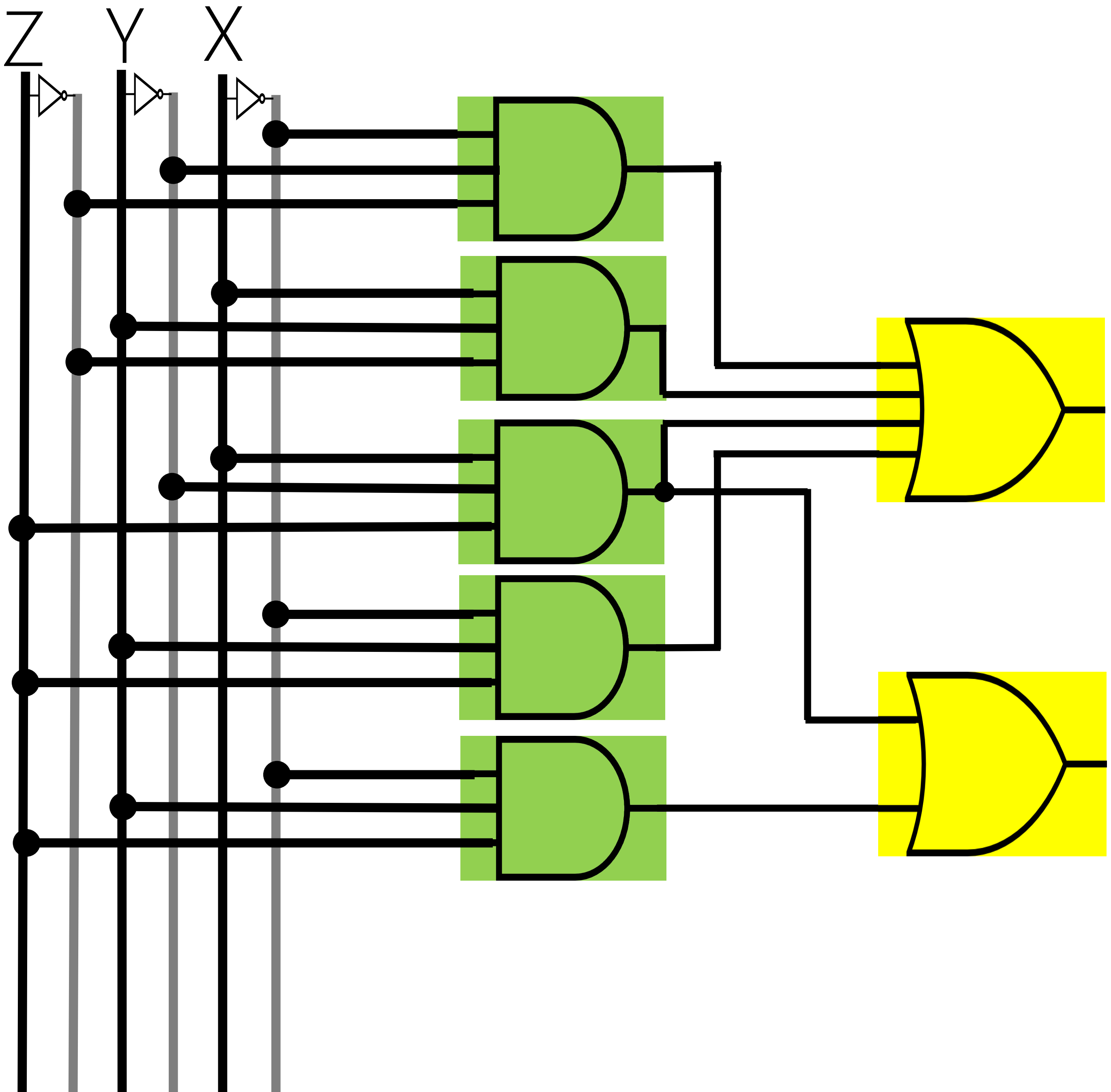


7402 Quad 2-input NOR Gates



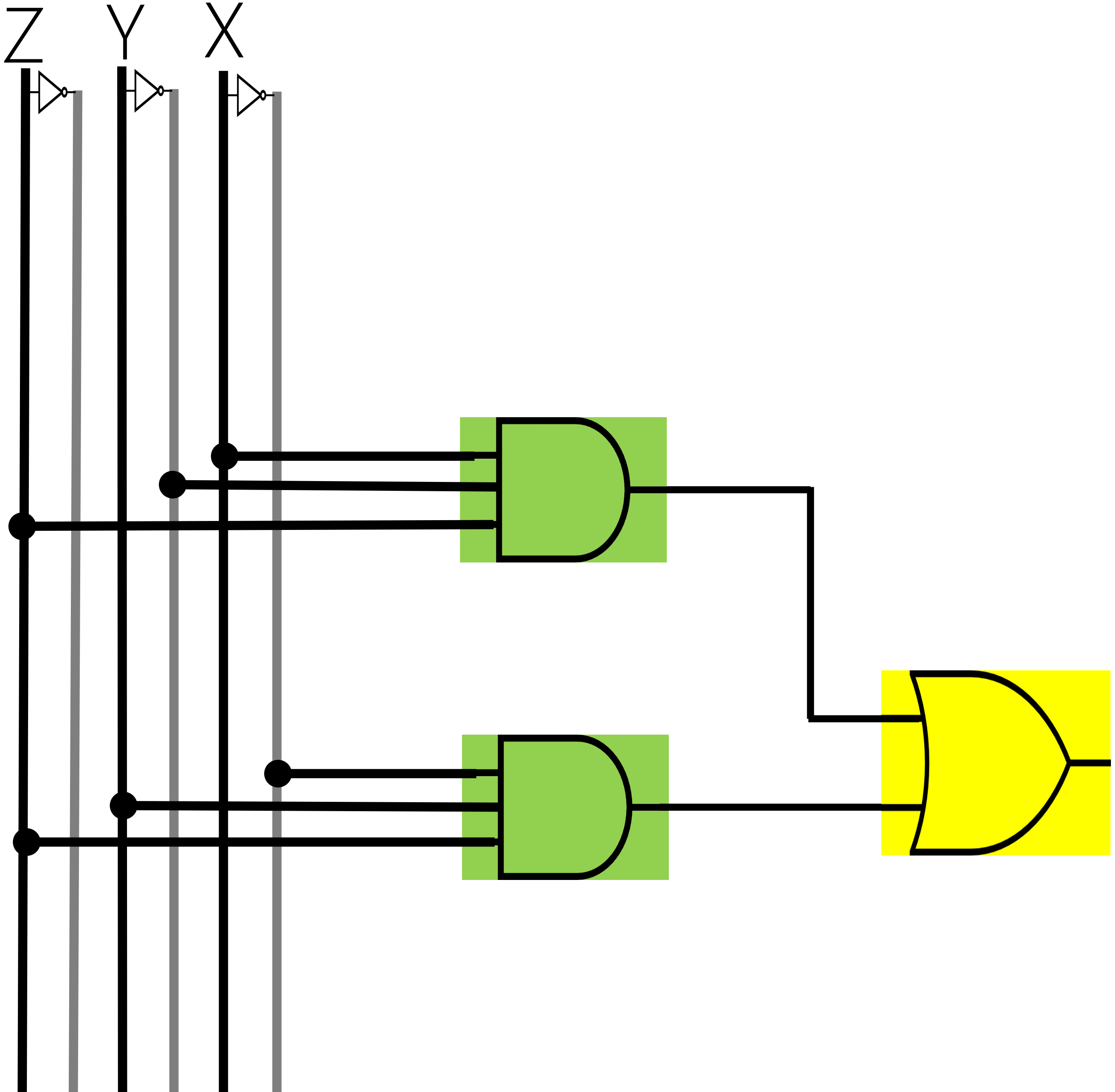
UNIVERSAL GATE

SoP → {NAND}

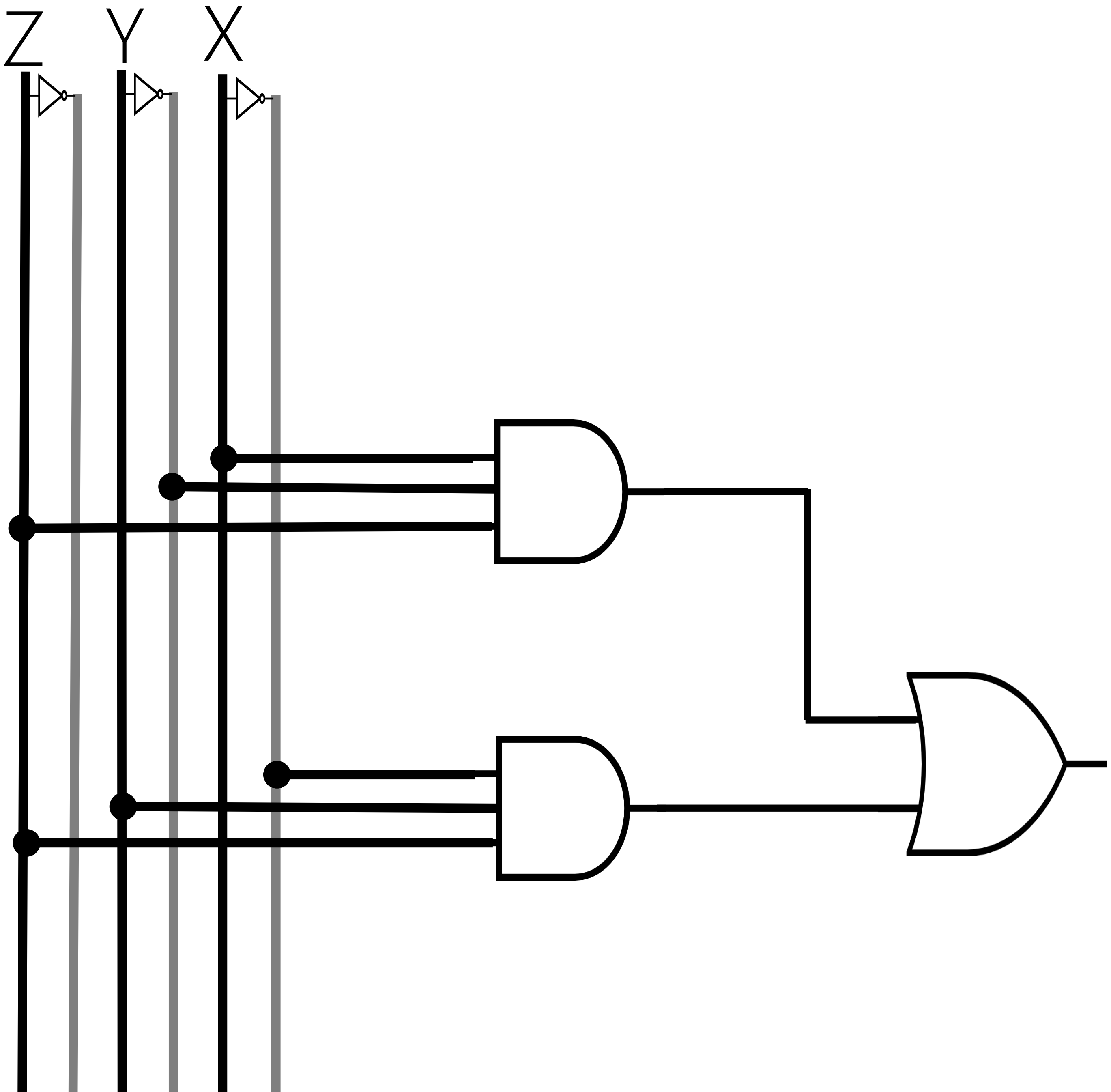


$F_1 = Z'Y'X' + Z'YX + ZY'X + ZYX'$

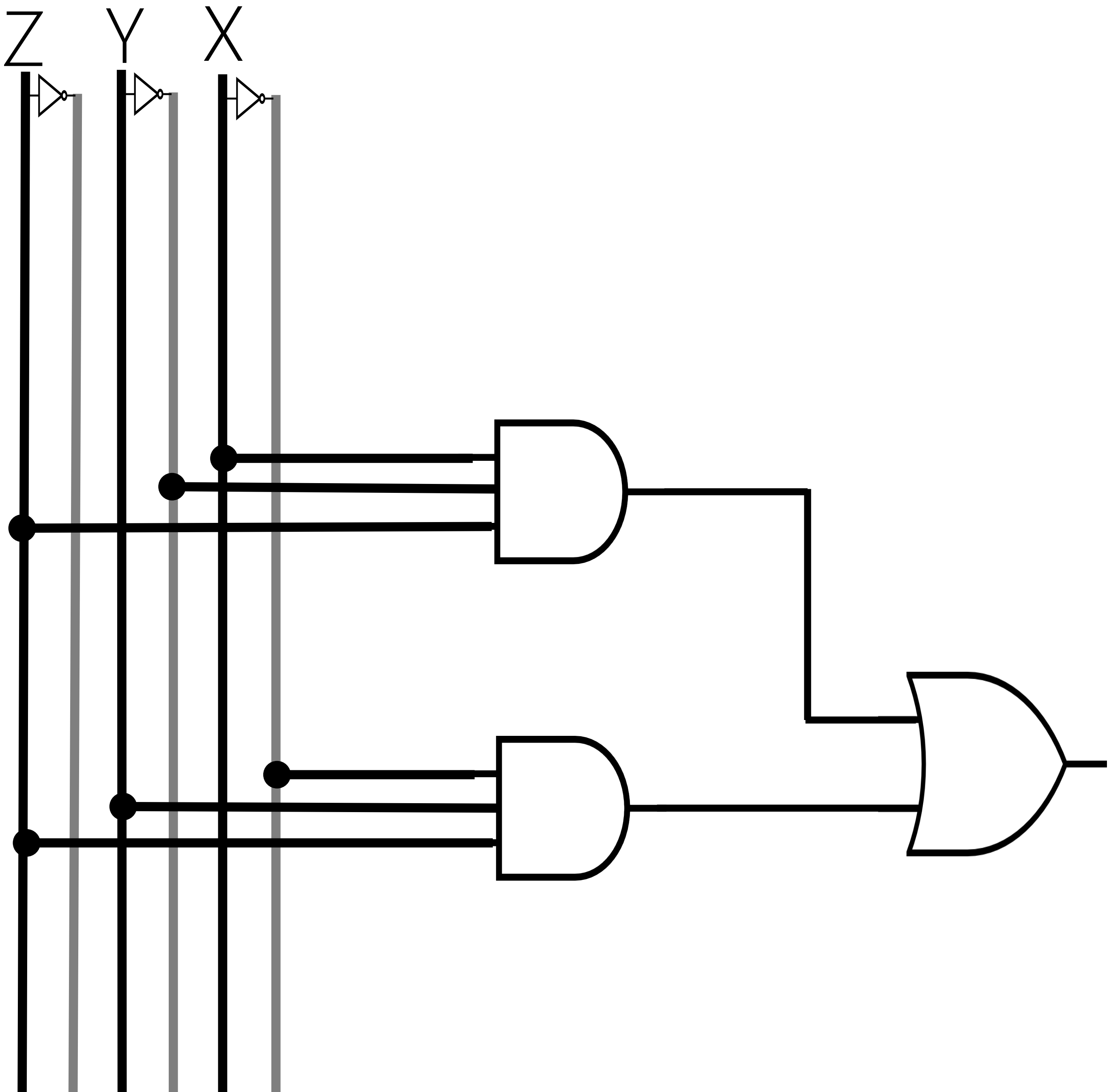
$F_2 = ZY'X' + ZY'X$



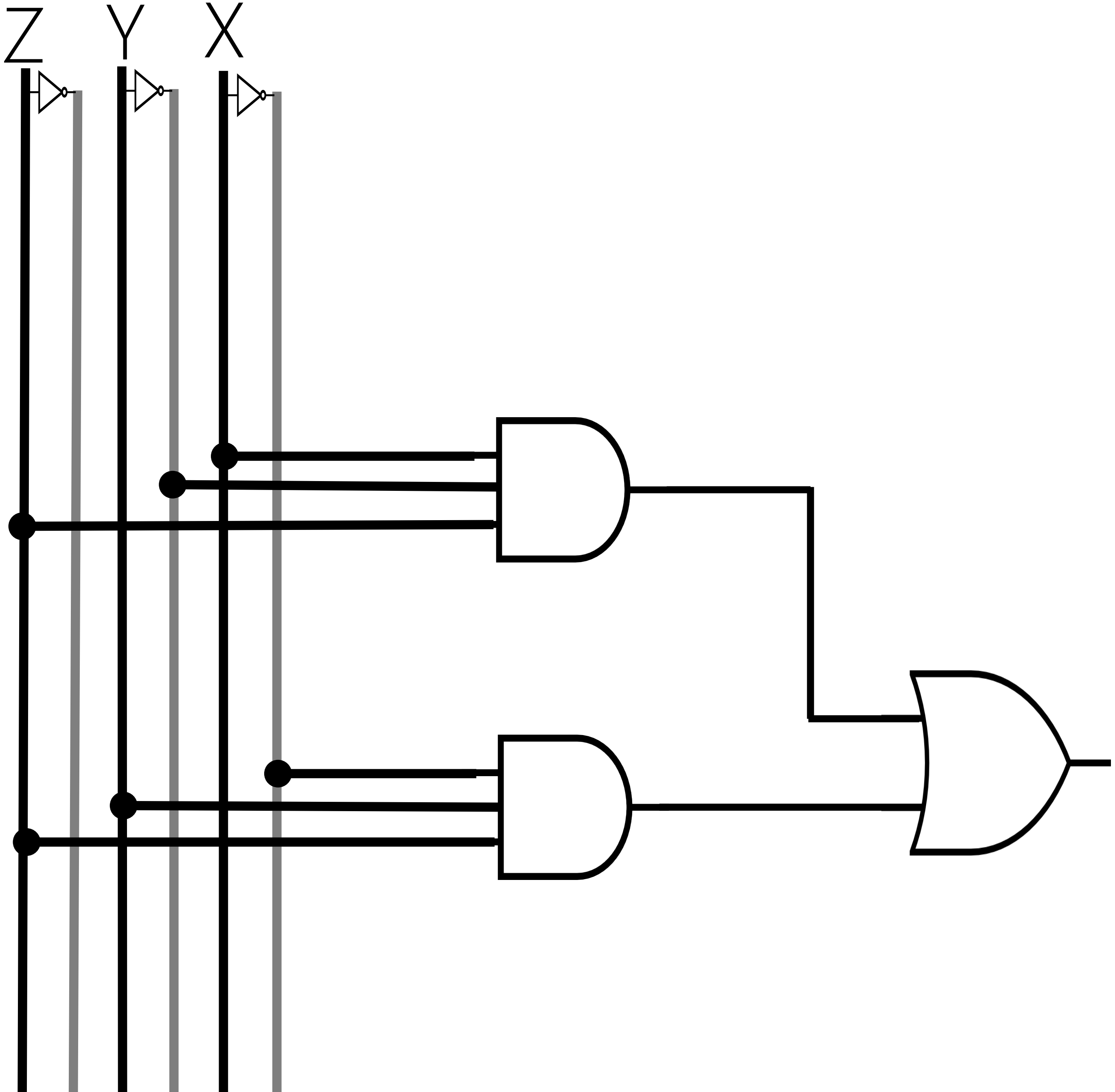
$F_2 = ZY'X' + ZY'X$



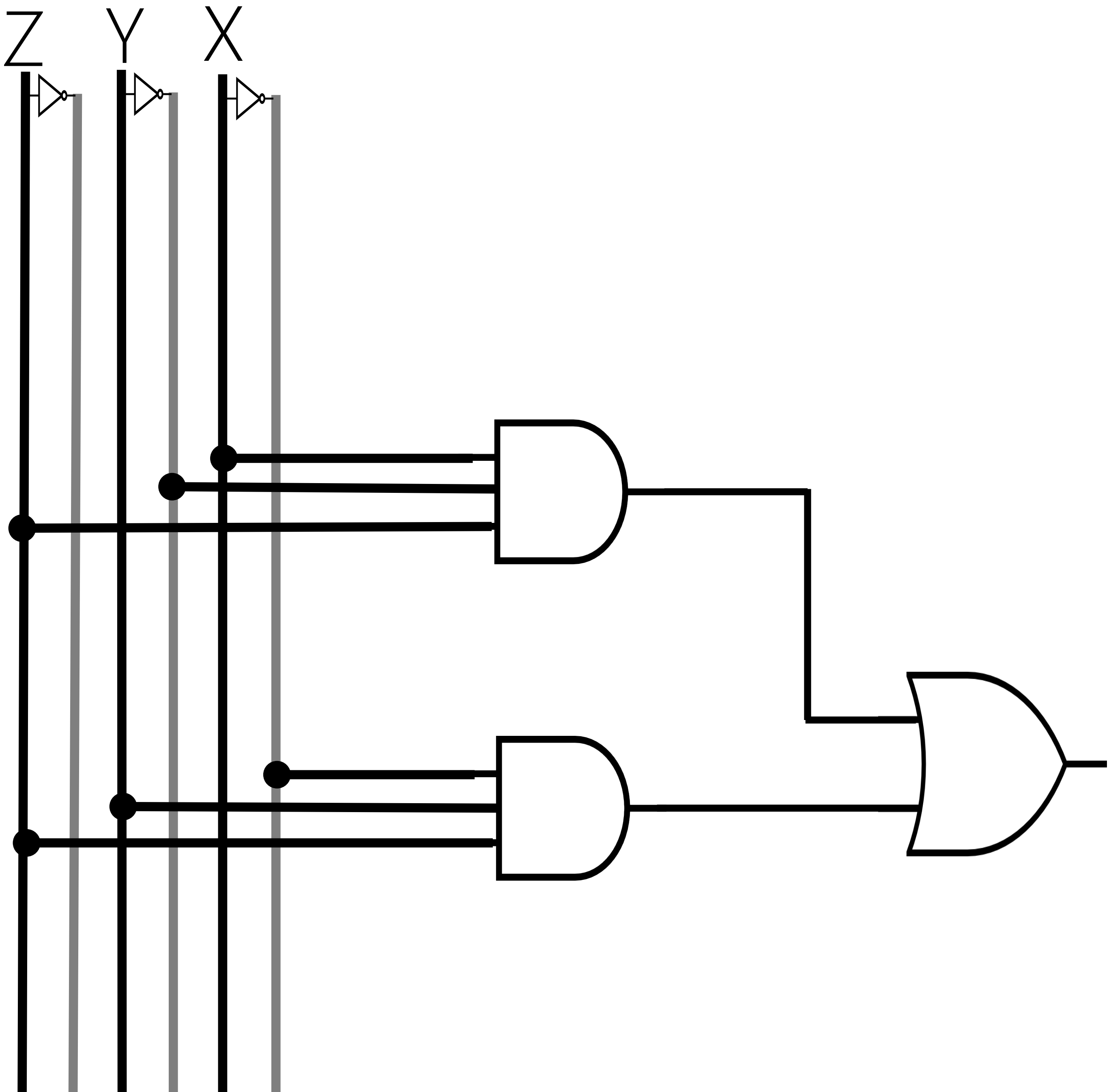
$$F_2 = m_4 + m_5$$
$$= ((F_2)')'$$



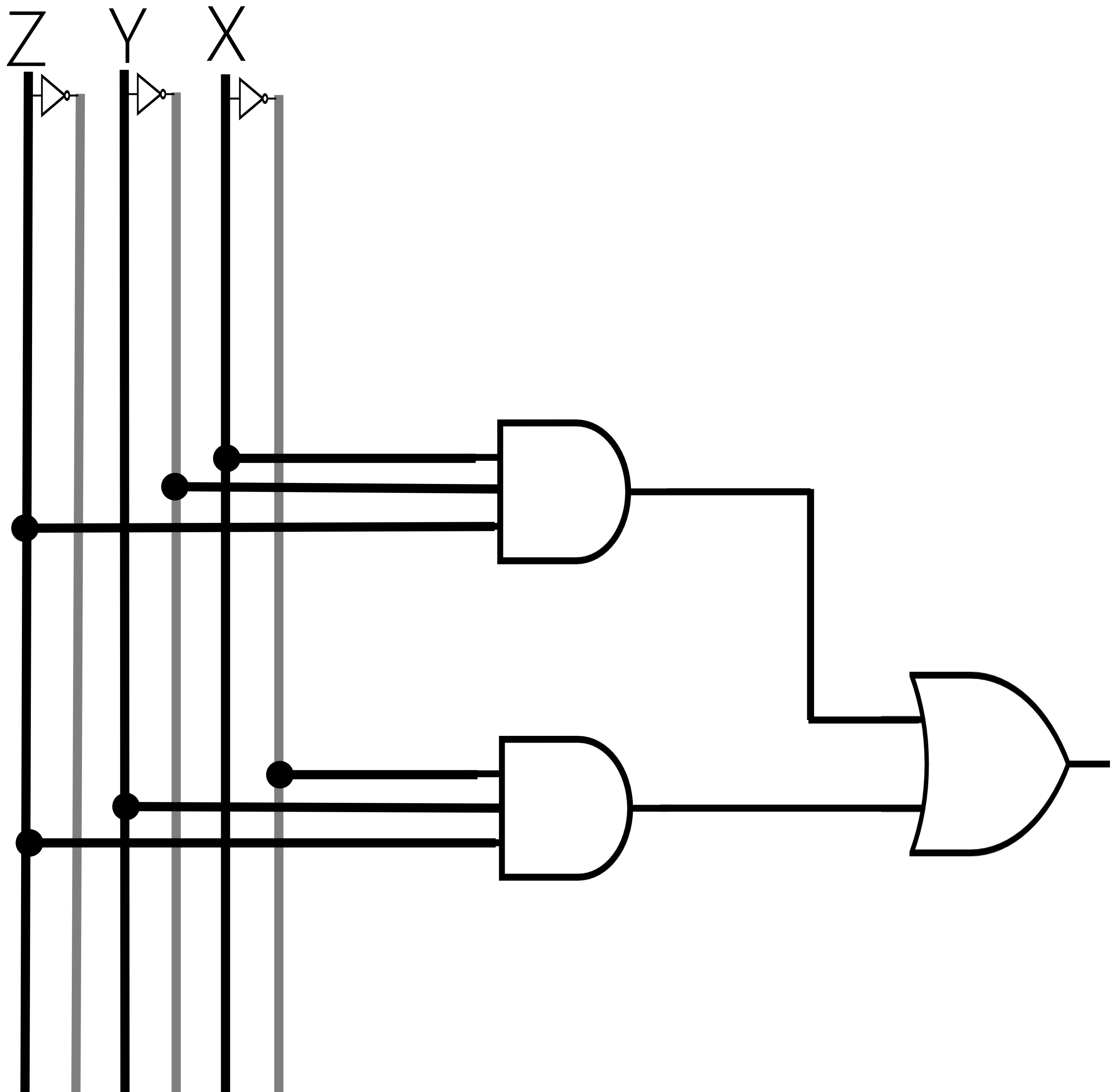
$$
\begin{aligned}
F_2 &= m_4+m_5 \\
&= ((F_2)')' \\
&= ((m_4+m_5)')'
\end{aligned}
$$



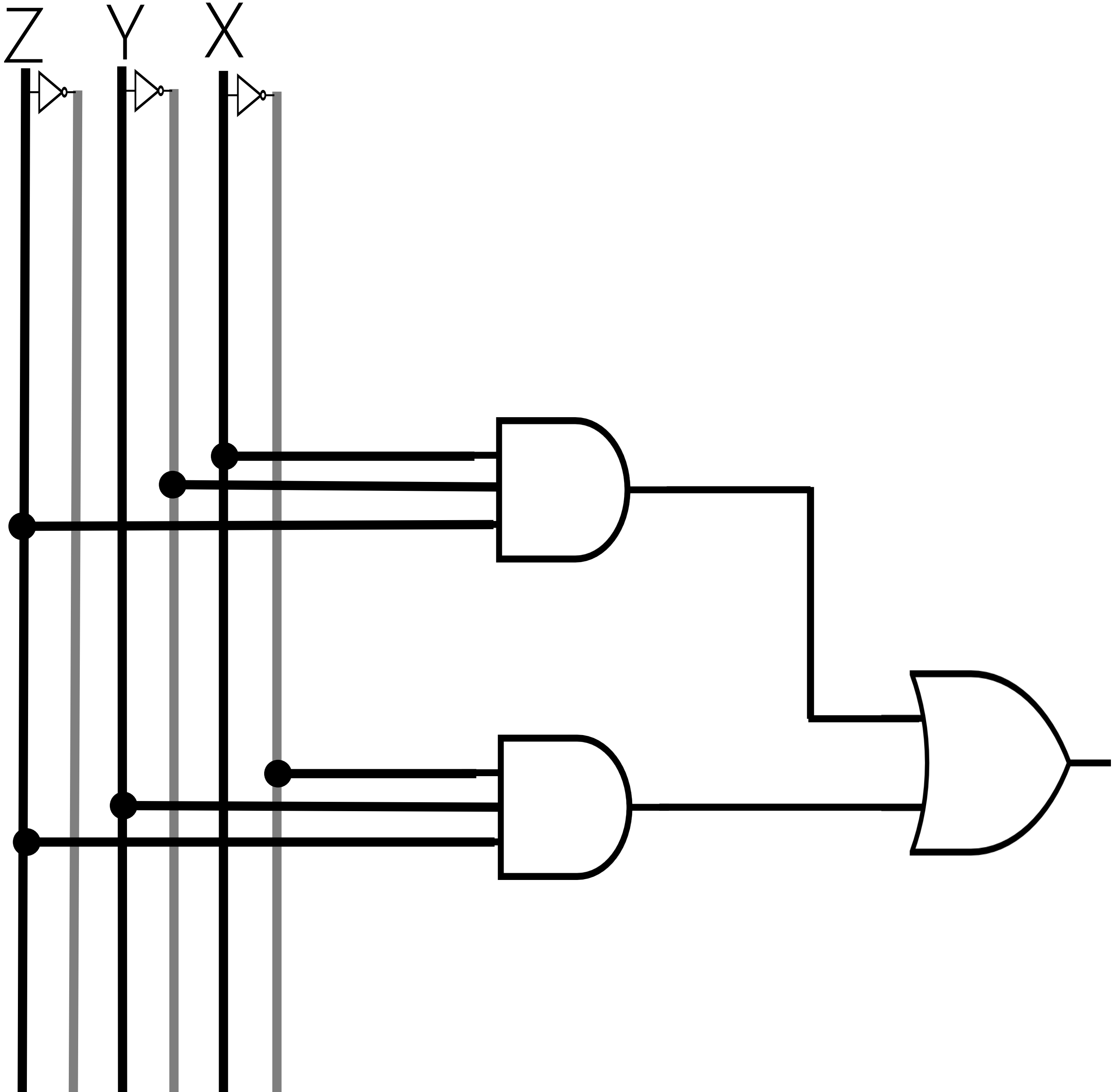
$$
\begin{aligned}
F_2 &= m_4 + m_5 \\
&= ((F_2)')' \\
&= ((m_4 + m_5)')' \\
&= (m'_4 m'_5)'
\end{aligned}
$$



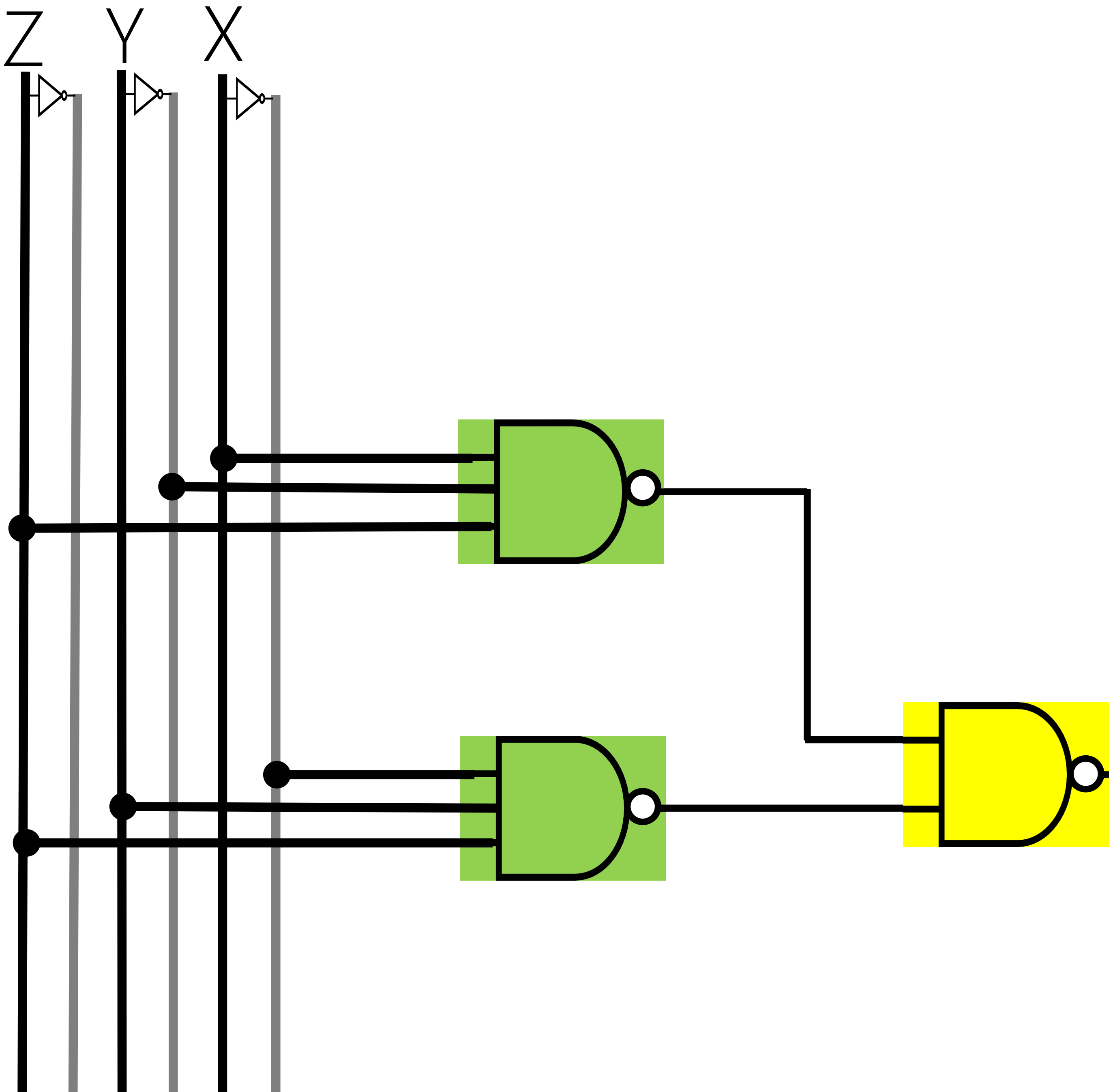
$$
\begin{aligned}
F_2 &= m_4 + m_5 \\
&= ((F_2)')' \\
&= ((m_4+m_5)')' \\
&= (m'_4 m'_5)' \\
&= ((ZY'X')' \, (ZY'X)')'
\end{aligned}
$$



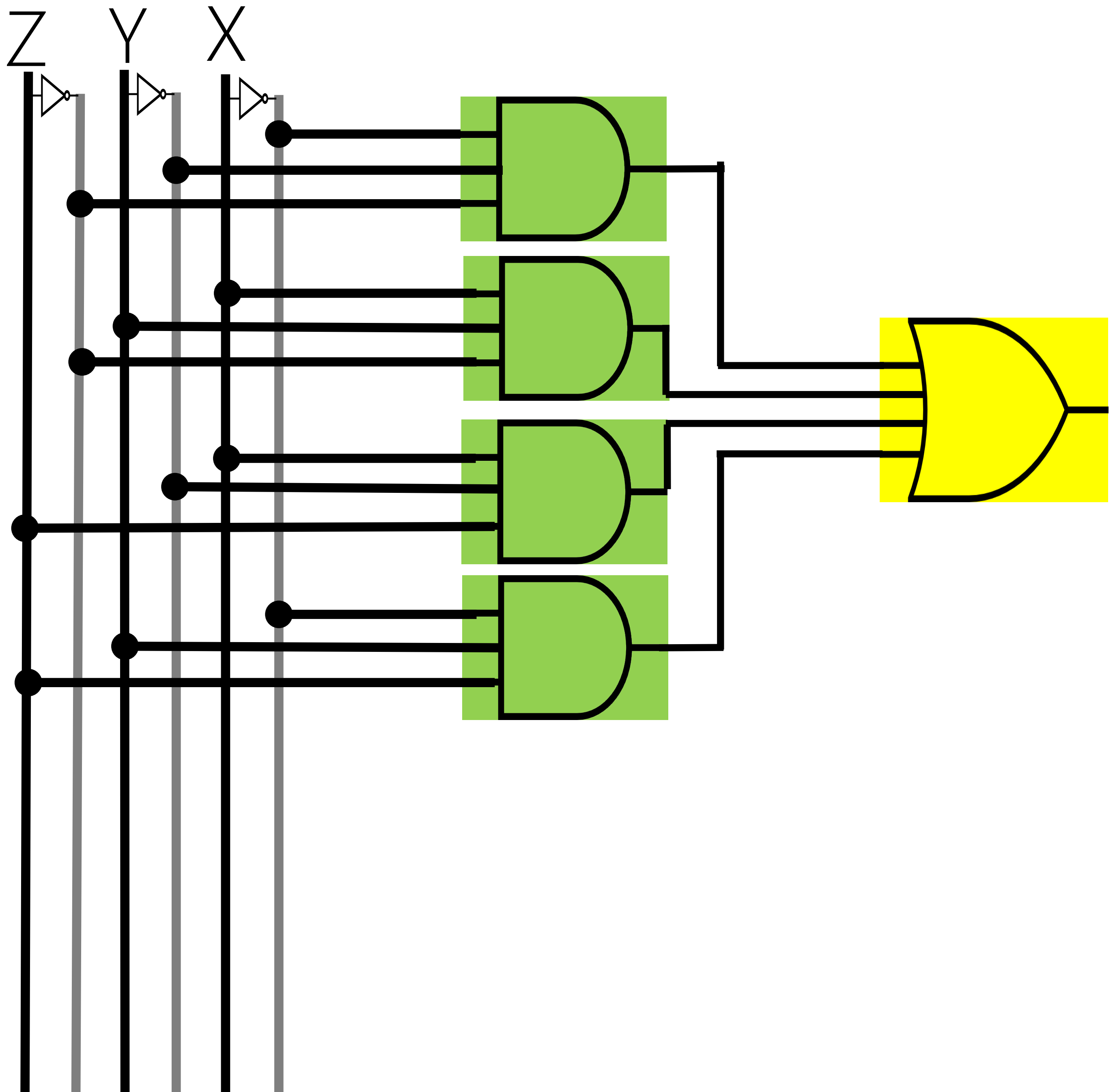
$$F_2 = m_4 + m_5$$
$$= ((F_2)')'$$
$$= ((m_4+m_5)')'$$
$$= (m'_4 m'_5)'$$
$$= ((ZY'X')'\,(ZY'X)')'$$
$$= ((Z \uparrow Y' \uparrow X')\,(Z \uparrow Y' \uparrow X))'$$



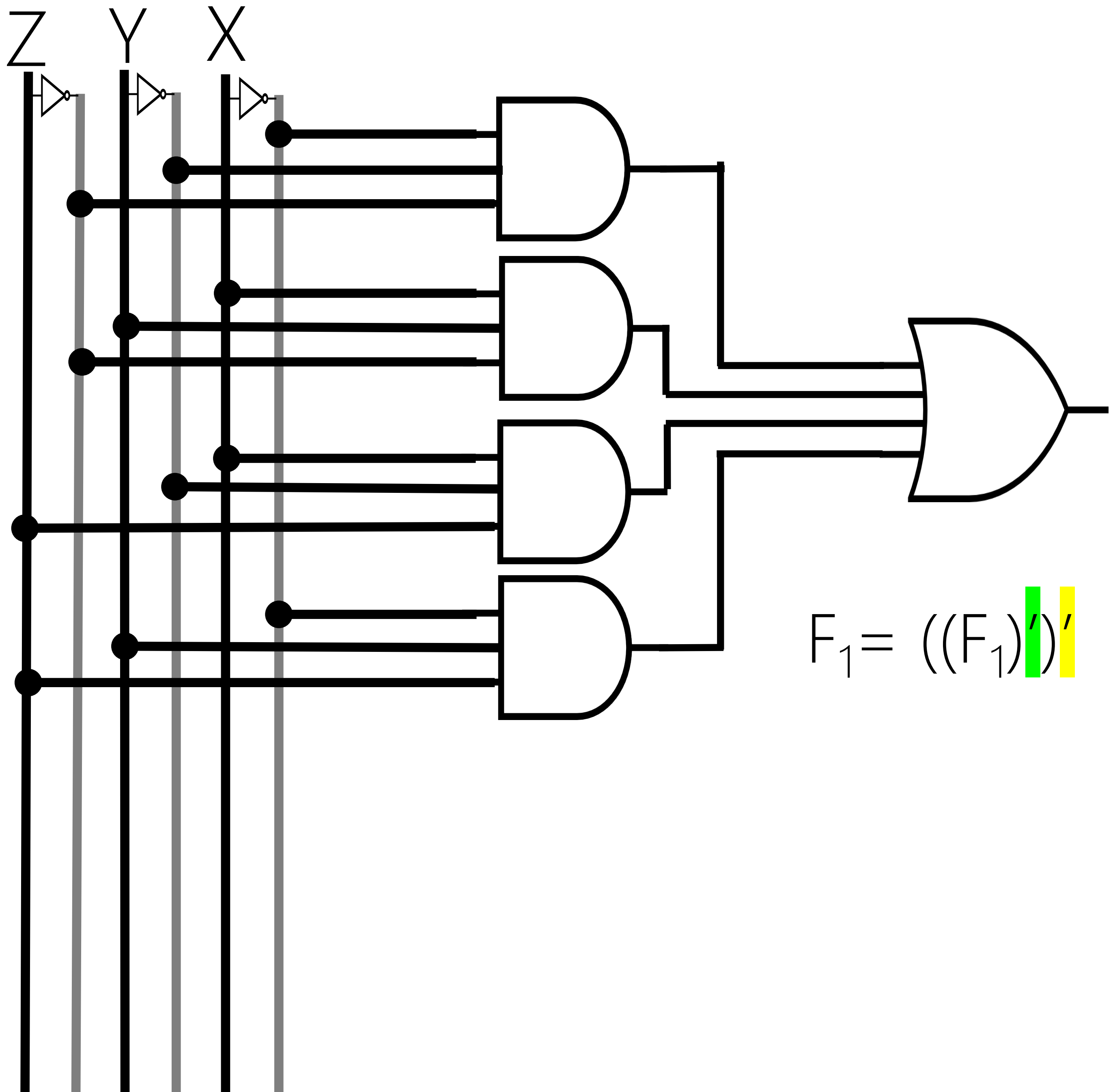
$$
\begin{aligned}
F_2 &= m_4 + m_5 \\
&= ((F_2)')' \\
&= ((m_4 + m_5)')' \\
&= (m'_4 m'_5)' \\
&= ((ZY'X')' \, (ZY'X)')' \\
&= ((Z \uparrow Y' \uparrow X') \, (Z \uparrow Y' \uparrow X))' \\
&= ((Z \uparrow Y' \uparrow X') \uparrow (Z \uparrow Y' \uparrow X))
\end{aligned}
$$



$$F_2 = ((Z \uparrow Y' \uparrow X') \uparrow (Z \uparrow Y' \uparrow X))$$

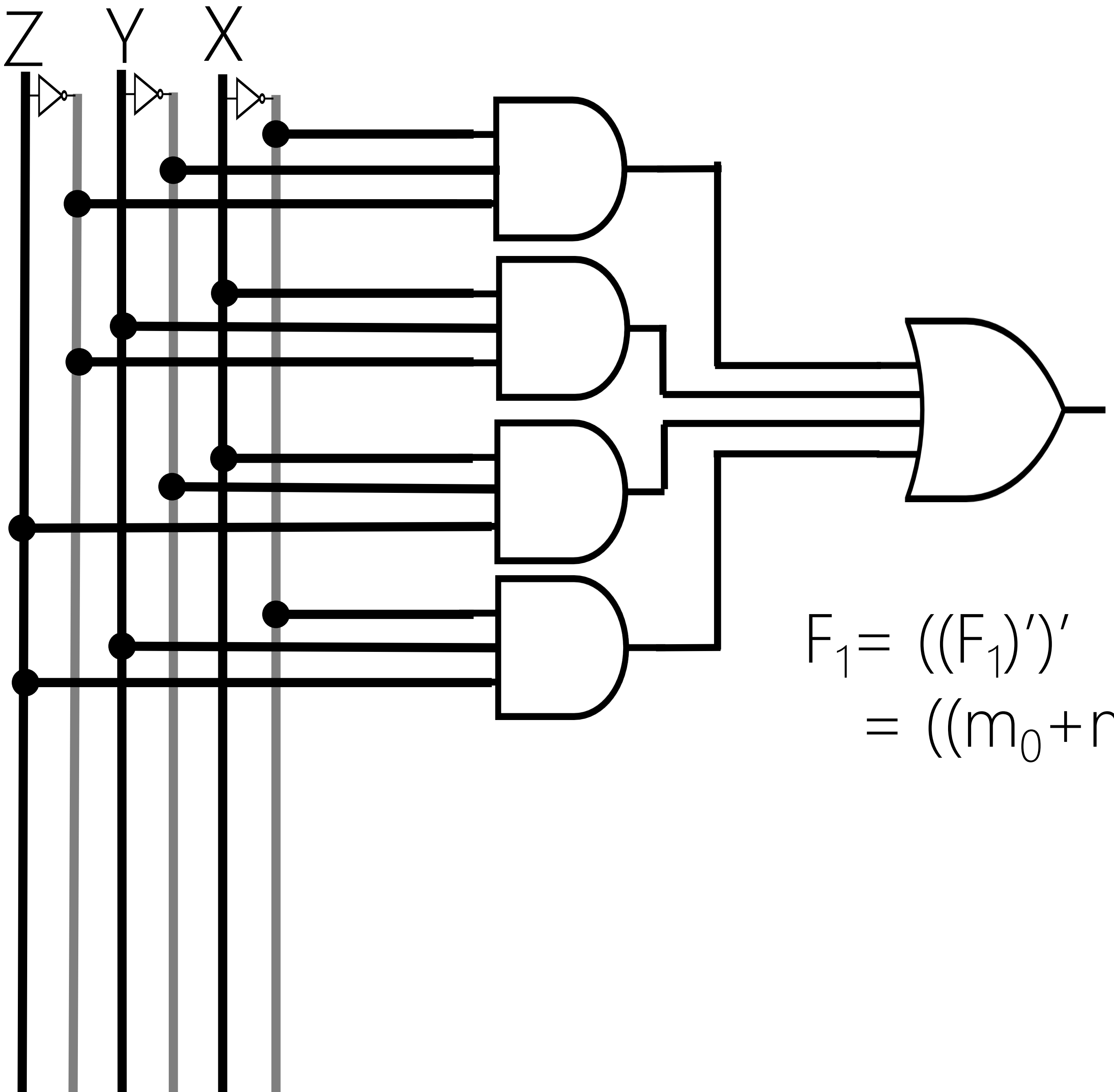


$$F_1 = Z'Y'X' + Z'YX + ZY'X + ZYX'$$



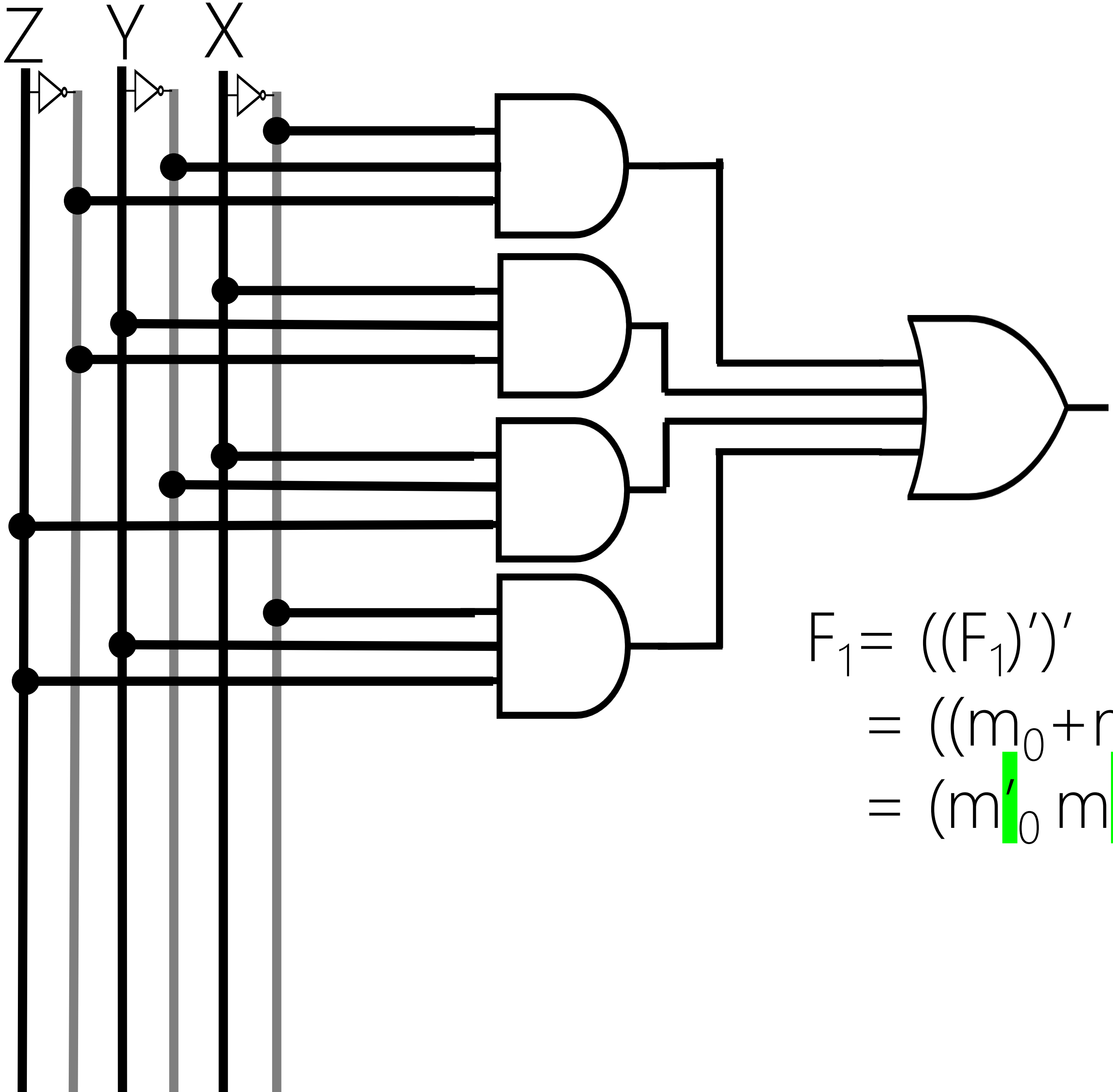
$$F_1 = Z'Y'X' + Z'YX + ZY'X + ZYX'$$

$$F_1 = ((F_1)')'$$



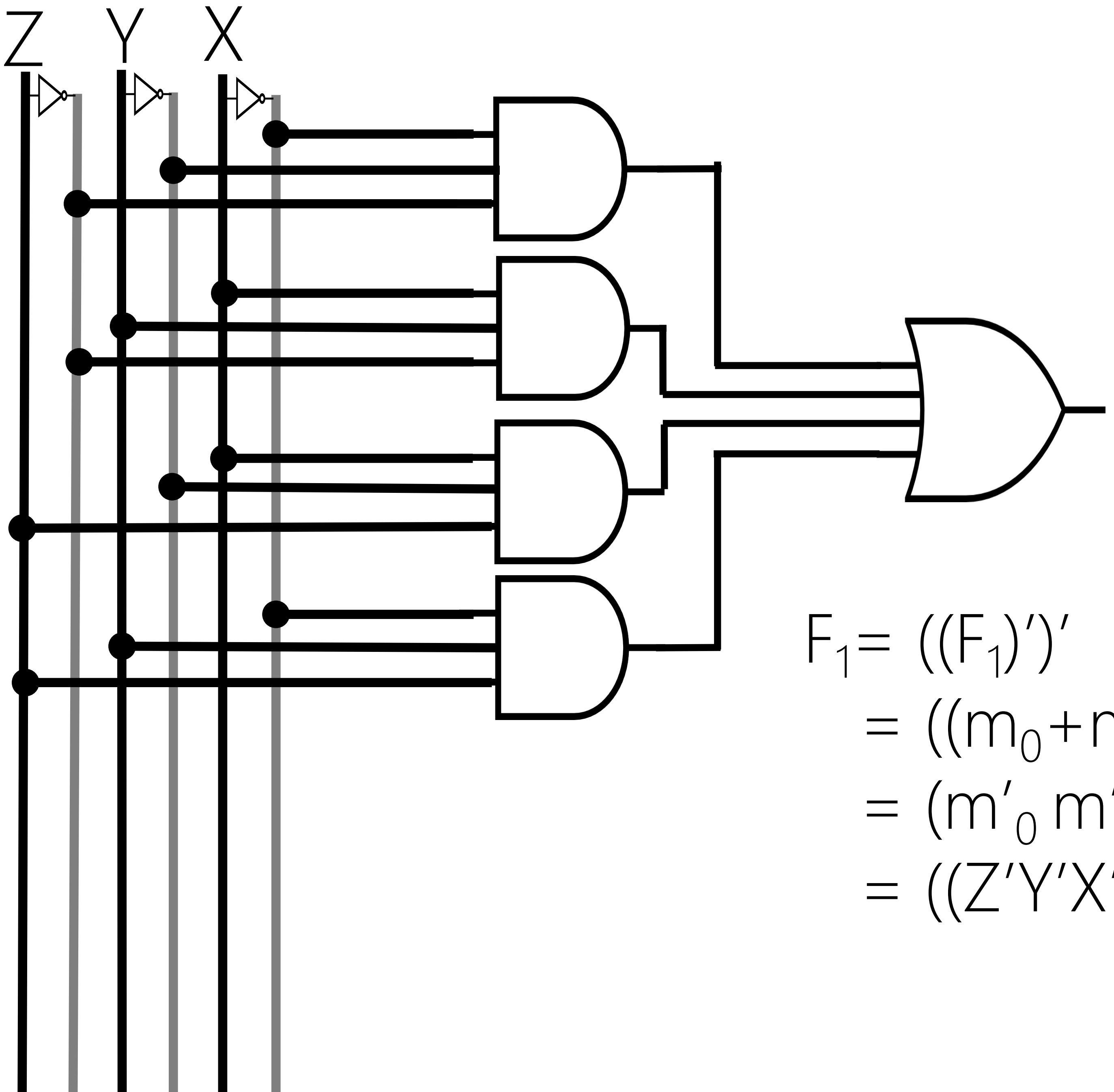
$$F_1 = Z'Y'X' + Z'YX + ZY'X + ZYX'$$

$$\begin{aligned} F_1 &= ((F_1)')' \\ &= ((m_0 + m_3 + m_5 + m_6)')' \end{aligned}$$



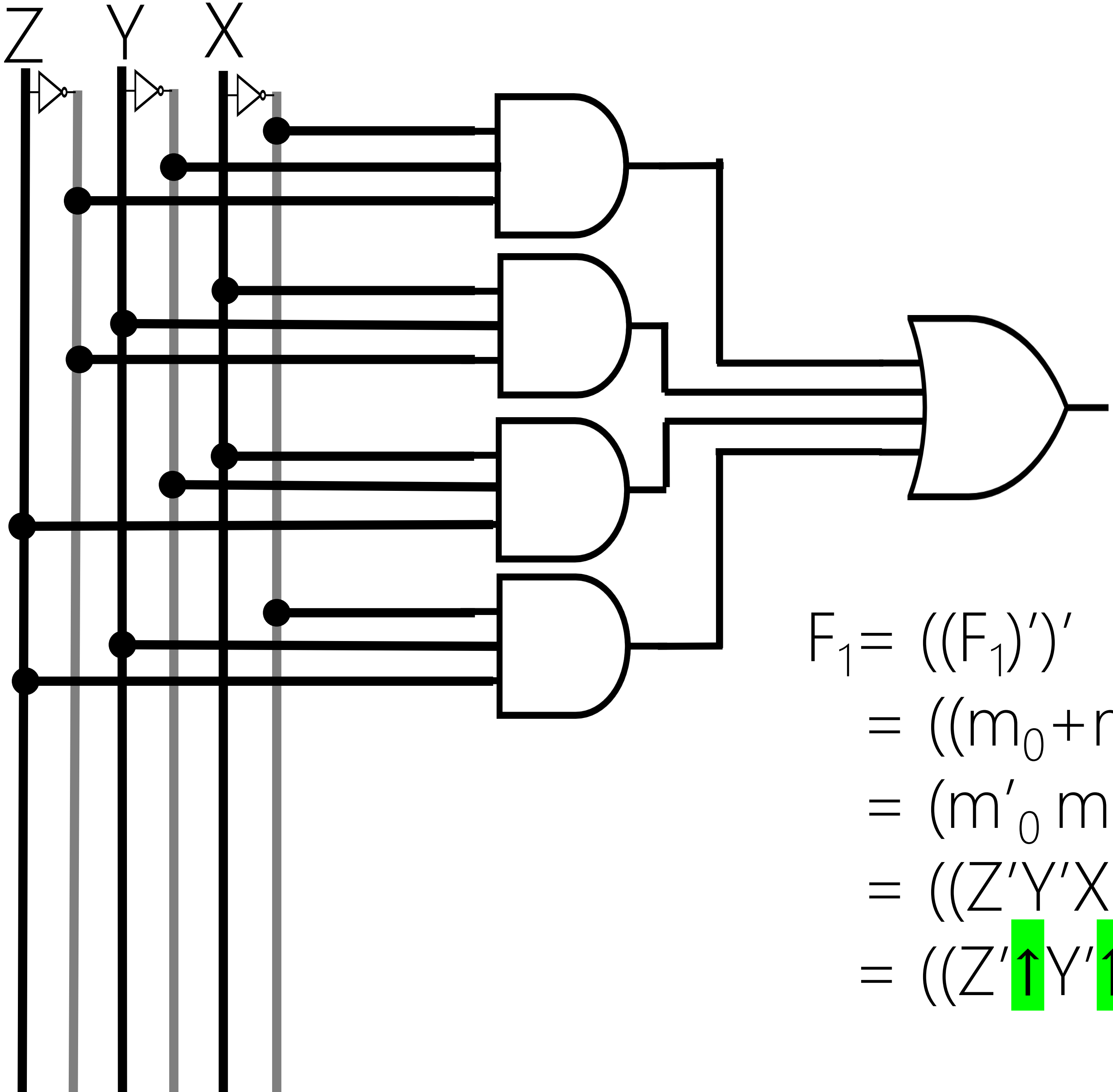
$$F_1 = Z'Y'X' + Z'YX + ZY'X + ZYX'$$

$$\begin{aligned} F_1 &= ((F_1)')' \\ &= ((m_0 + m_3 + m_5 + m_6)')' \\ &= (m'_0\, m'_3\, m'_5\, m'_6)' \end{aligned}$$



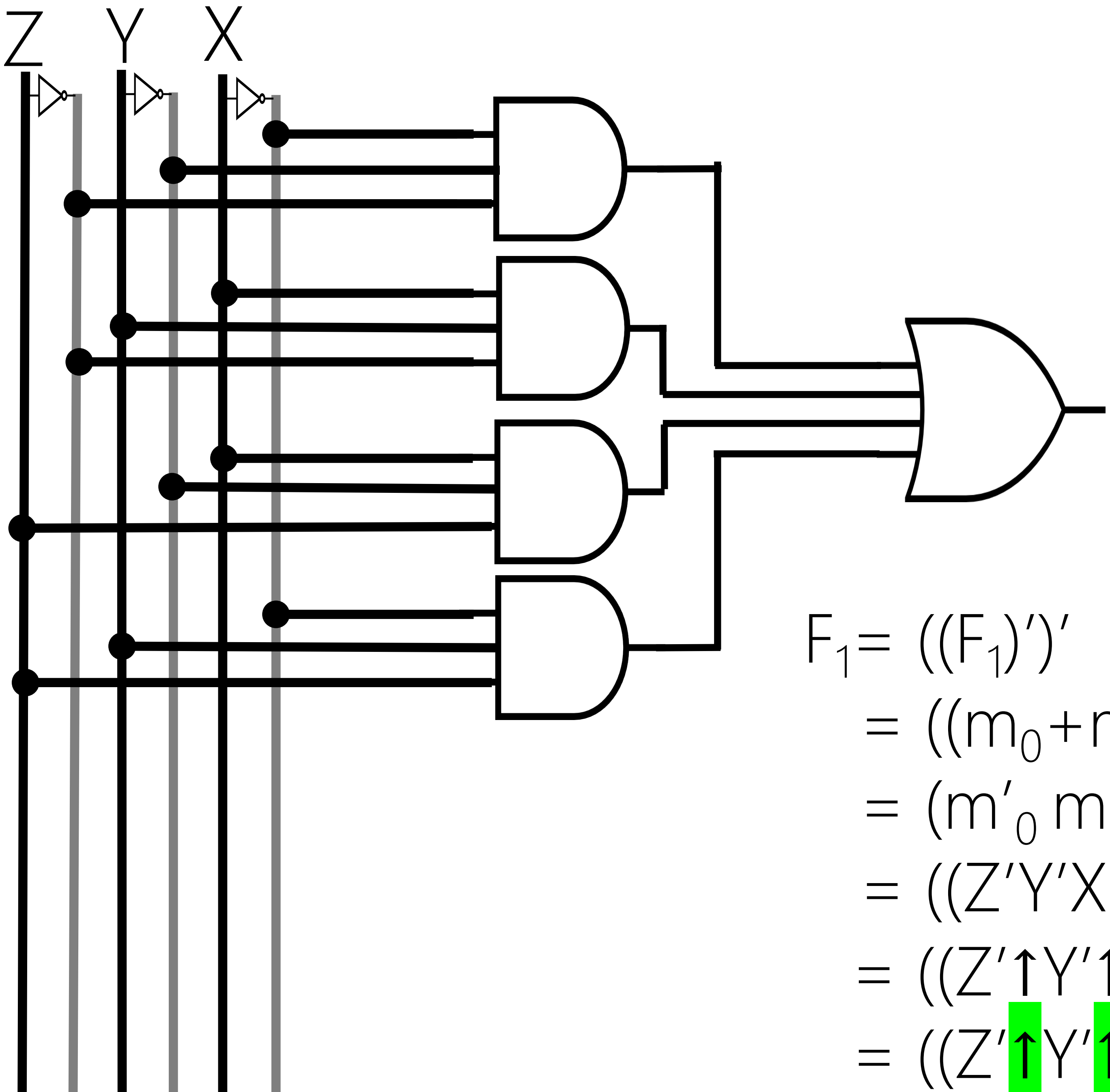
$$F_1 = Z'Y'X' + Z'YX + ZY'X + ZYX'$$

$$\begin{aligned}
F_1 &= ((F_1)')' \\
&= ((m_0 + m_3 + m_5 + m_6)')' \\
&= (m'_0\, m'_3\, m'_5\, m'_6)' \\
&= ((Z'Y'X')'\,(Z'YX)'\,(ZY'X)'\,(ZYX')')'
\end{aligned}$$



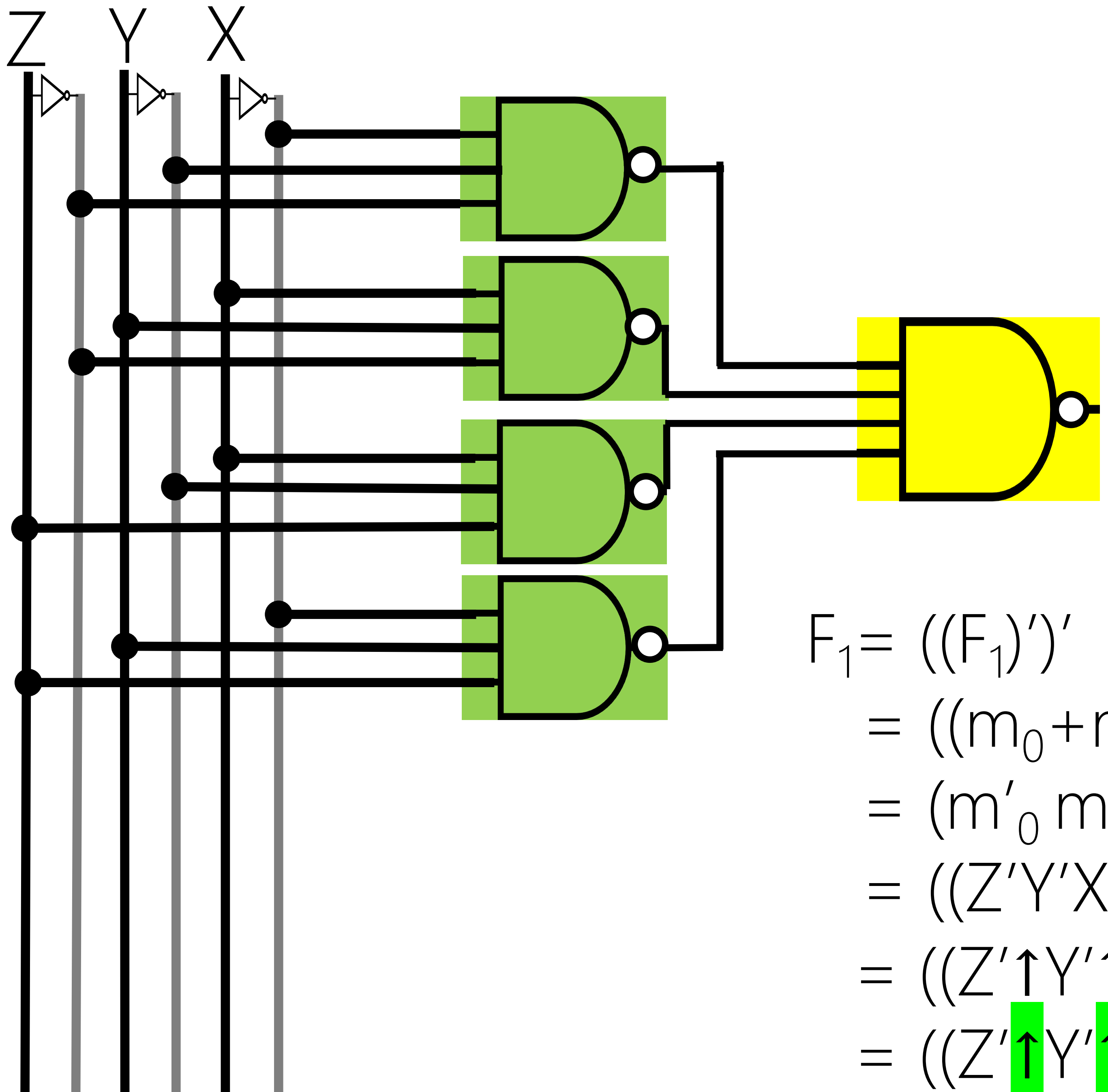
$$F_1 = Z'Y'X' + Z'YX + ZY'X + ZYX'$$

$$\begin{aligned}
F_1 &= ((F_1)')' \\
&= ((m_0 + m_3 + m_5 + m_6)')' \\
&= (m'_0\, m'_3\, m'_5\, m'_6)' \\
&= ((Z'Y'X')'\, (Z'YX)'\, (ZY'X)'\, (ZYX')')' \\
&= ((Z' \uparrow Y' \uparrow X')\, (Z' \uparrow Y \uparrow X)\, (Z \uparrow Y' \uparrow X)\, (Z \uparrow Y \uparrow X'))'
\end{aligned}$$

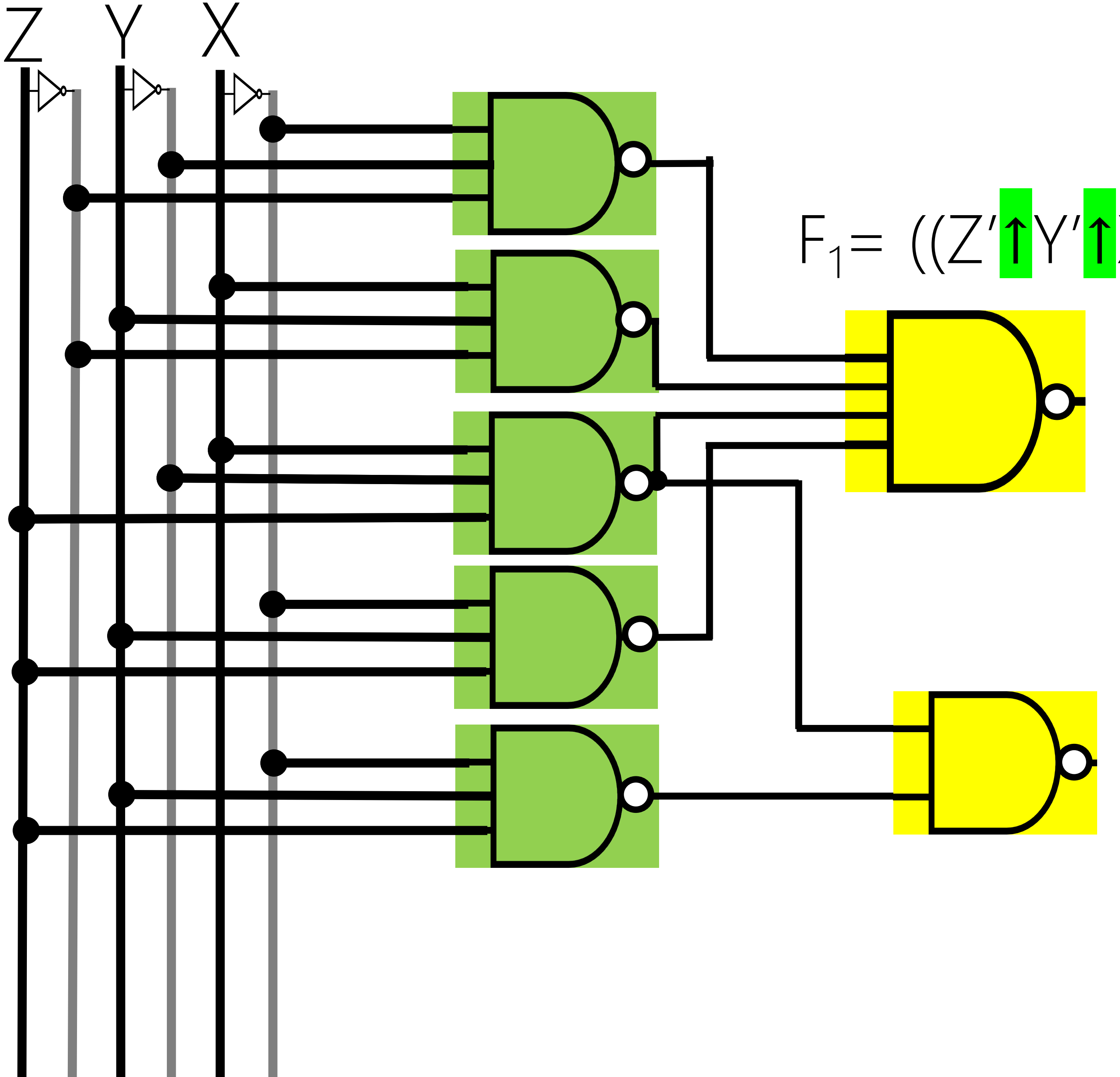


$$F_1 = Z'Y'X' + Z'YX + ZY'X + ZYX'$$

$$\begin{aligned}
F_1 &= ((F_1)')' \\
&= ((m_0 + m_3 + m_5 + m_6)')' \\
&= (m'_0\, m'_3\, m'_5\, m'_6)' \\
&= ((Z'Y'X')'\ (Z'YX)'\ (ZY'X)'\ (ZYX'))' \\
&= ((Z'\uparrow Y'\uparrow X')\ (Z'\uparrow Y\uparrow X)\ (Z\uparrow Y'\uparrow X)\ (Z\uparrow Y\uparrow X'))' \\
&= ((Z'\uparrow Y'\uparrow X')\uparrow(Z'\uparrow Y\uparrow X)\uparrow(Z\uparrow Y'\uparrow X)\uparrow(Z\uparrow Y\uparrow X'))
\end{aligned}$$

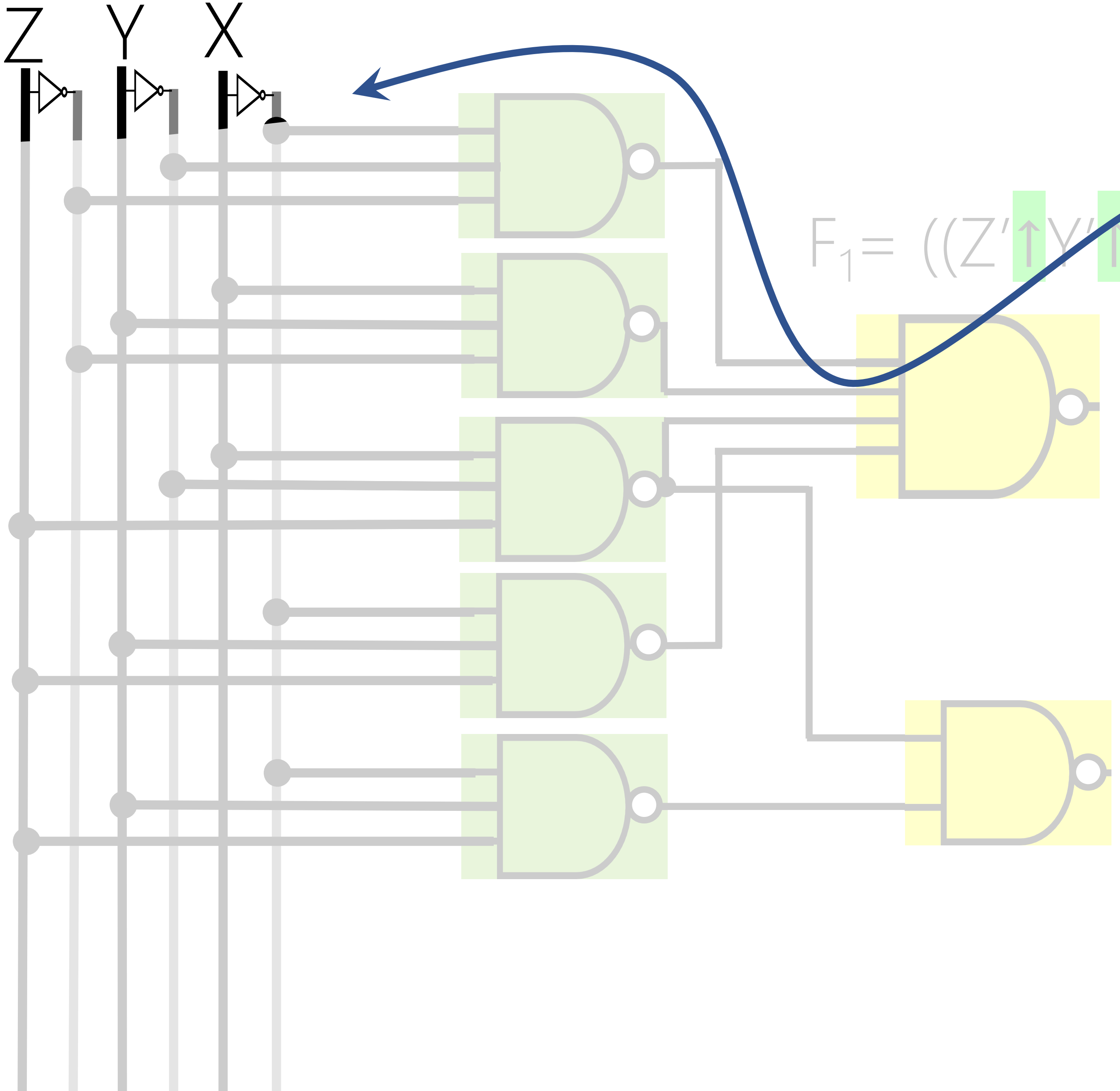


$$F_1 = ((F_1)')'$$
$$= ((m_0+m_3+m_5+m_6)')'$$
$$= (m'_0\, m'_3\, m'_5\, m'_6)'$$
$$= ((Z'Y'X')'\ (Z'YX)'\ (ZY'X)'\ (ZYX'))'$$
$$= ((Z'\uparrow Y'\uparrow X')\ (Z'\uparrow Y\uparrow X)\ (Z\uparrow Y'\uparrow X)\ (Z\uparrow Y\uparrow X'))'$$
$$= ((Z'\uparrow Y'\uparrow X')\uparrow(Z'\uparrow Y\uparrow X)\uparrow(Z\uparrow Y'\uparrow X)\uparrow(Z\uparrow Y\uparrow X'))$$



$$F_1 = ((Z'\uparrow Y'\uparrow X')\uparrow(Z'\uparrow Y\uparrow X)\uparrow(Z\uparrow Y'\uparrow X)\uparrow(Z\uparrow Y\uparrow X'))$$

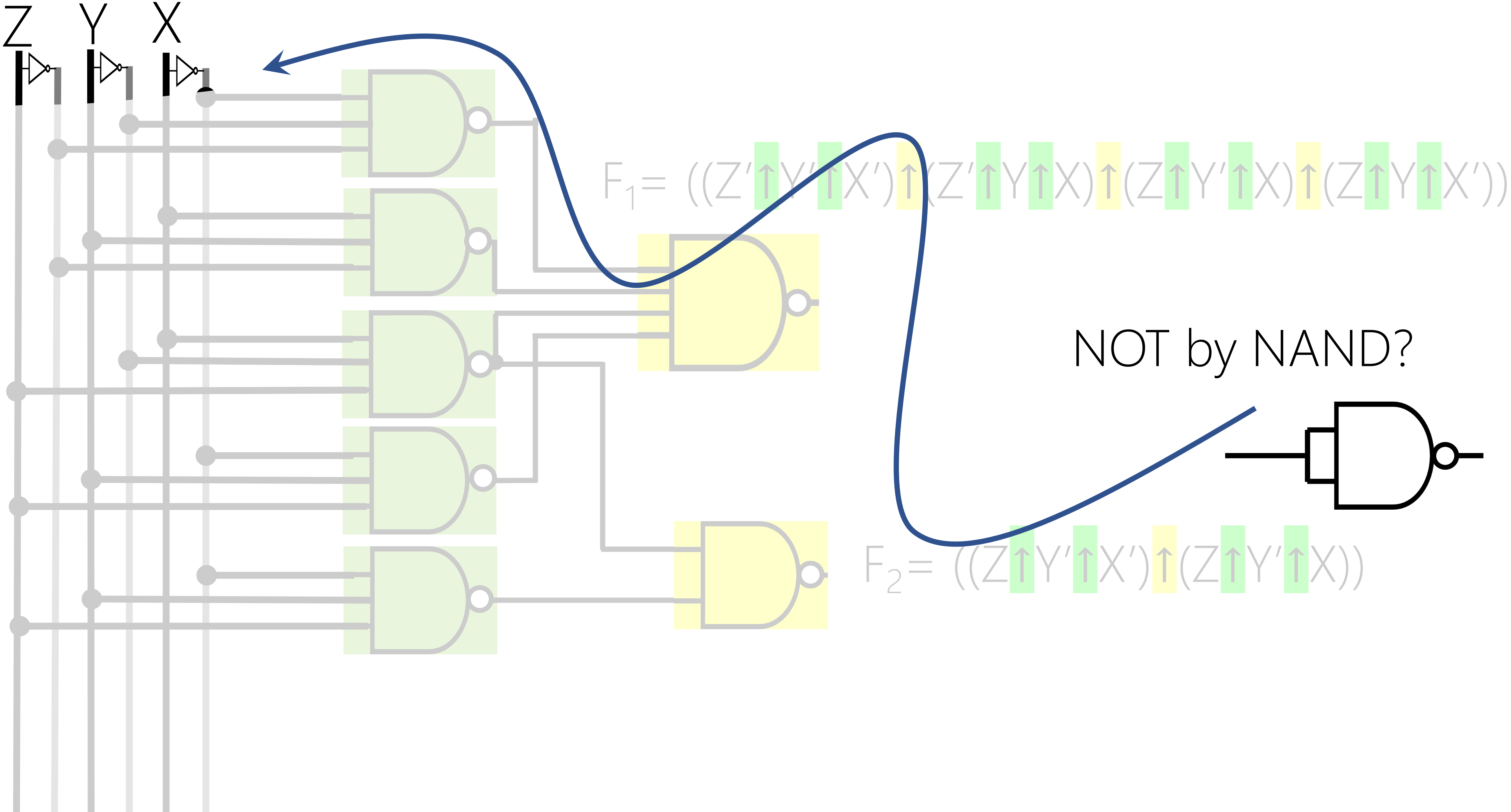
$$F_2 = ((Z\uparrow Y'\uparrow X')\uparrow(Z\uparrow Y'\uparrow X))$$



$F_1 = ((Z' \uparrow Y' \uparrow X') \uparrow (Z' \uparrow Y \uparrow X) \uparrow (Z \uparrow Y' \uparrow X) \uparrow (Z \uparrow Y \uparrow X'))$

$F_2 = ((Z \uparrow Y' \uparrow X') \uparrow (Z \uparrow Y' \uparrow X))$

NOT by NAND?


Z Y X
$F_1 = ((Z' \uparrow Y' \uparrow X') \uparrow (Z' \uparrow Y \uparrow X) \uparrow (Z \uparrow Y' \uparrow X) \uparrow (Z \uparrow Y \uparrow X'))$
NOT by NAND?
$F_2 = ((Z \uparrow Y' \uparrow X') \uparrow (Z \uparrow Y' \uparrow X))$

UNIVERSAL GATE {NAND}

$F=(F')'=(SoP')'$

UNIVERSAL GATE
{NOR}

UNIVERSAL GATE
PoS → {NOR}

$$F_{PoS} = (F')'$$

Lecture Assignment

RECAP

Any Boolean Function F:

- Sum (OR) of Products (ANDs)
- Sum of minterms for Entries with 1
  - ANDs-OR
  - NAND via (F')'

- Product (AND) of Sums (ORs)
- Product of MAXTERMS (NOT minterms) for Entries with 0
  - ORs-AND
  - NOR via (F')'

UNIVERSAL GATE

SoP → {NAND}

SoP → {NOR} ?

PoS → {NOR}

PoS → {NAND} ?

Lecture Assignment