CMP 334: *Sixth Class*

HW 4 solution

Non-standard truth table formats

Boolean formula → combinational circuit practice

Computer arithmetic part 1 (non-negative numbers)

- Addition (and carry)
- Subtraction

Inverters, decoders, multiplexers

TINY instruction set architecture

For next class: HW 5 (lookover HW 6) read A.3, 2.1-4

HW 4: Boolean Formulas

1) Present (i) the truth table and (ii) a circuit diagram for the following Boolean formulas:

a) $(a + b)\,(\bar{a} + \bar{b})$

b) $xyz + \overline{xy}z + \bar{x}y\bar{z} + x\overline{yz}$

c) $(a + \bar{b})\,(b + \bar{c})\,(c + \bar{a})$

d) $(x + y + z)\,\overline{(xyz)}$

2) Give a truth table to prove or disprove the following identity:

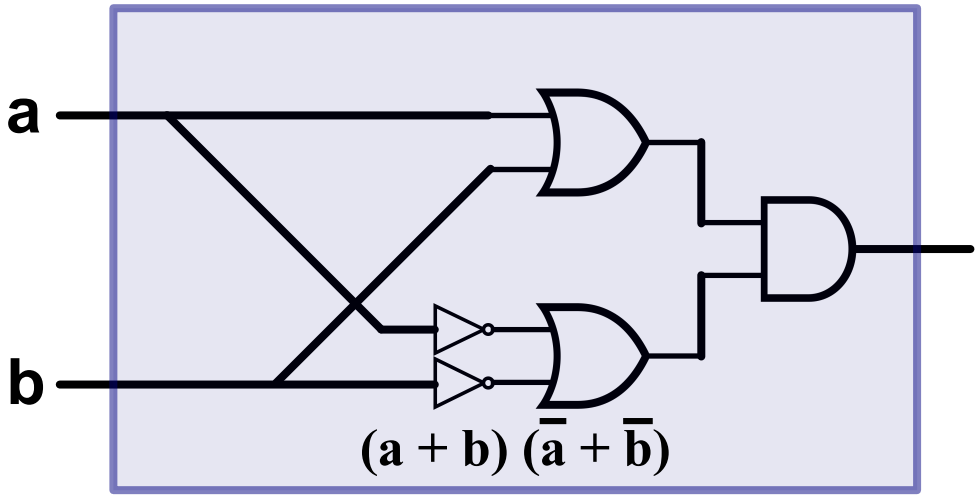
$$(\bar{x} + y)\,(y + \bar{z}) = y + \overline{(x+z)}$$

HW 4.1a: $(a + b)(\bar{a} + \bar{b})$

a	b	$(a + b)$	$\bar{a}$	$\bar{b}$	$(\bar{a} + \bar{b})$	$(a + b)(\bar{a} + \bar{b})$
0	0	0	1	1	1	0
0	1	1	1	0	1	1
1	0	1	0	1	1	1
1	1	1	0	0	0	0

HW 4.1a: $(a + b)(\bar{a} + \bar{b})$

a	b	$(a + b)$	$\bar{a}$	$\bar{b}$	$(\bar{a} + \bar{b})$	$(a + b)(\bar{a} + \bar{b})$
0	0	0	1	1	1	0
0	1	1	1	0	1	1
1	0	1	0	1	1	1
1	1	1	0	0	0	0



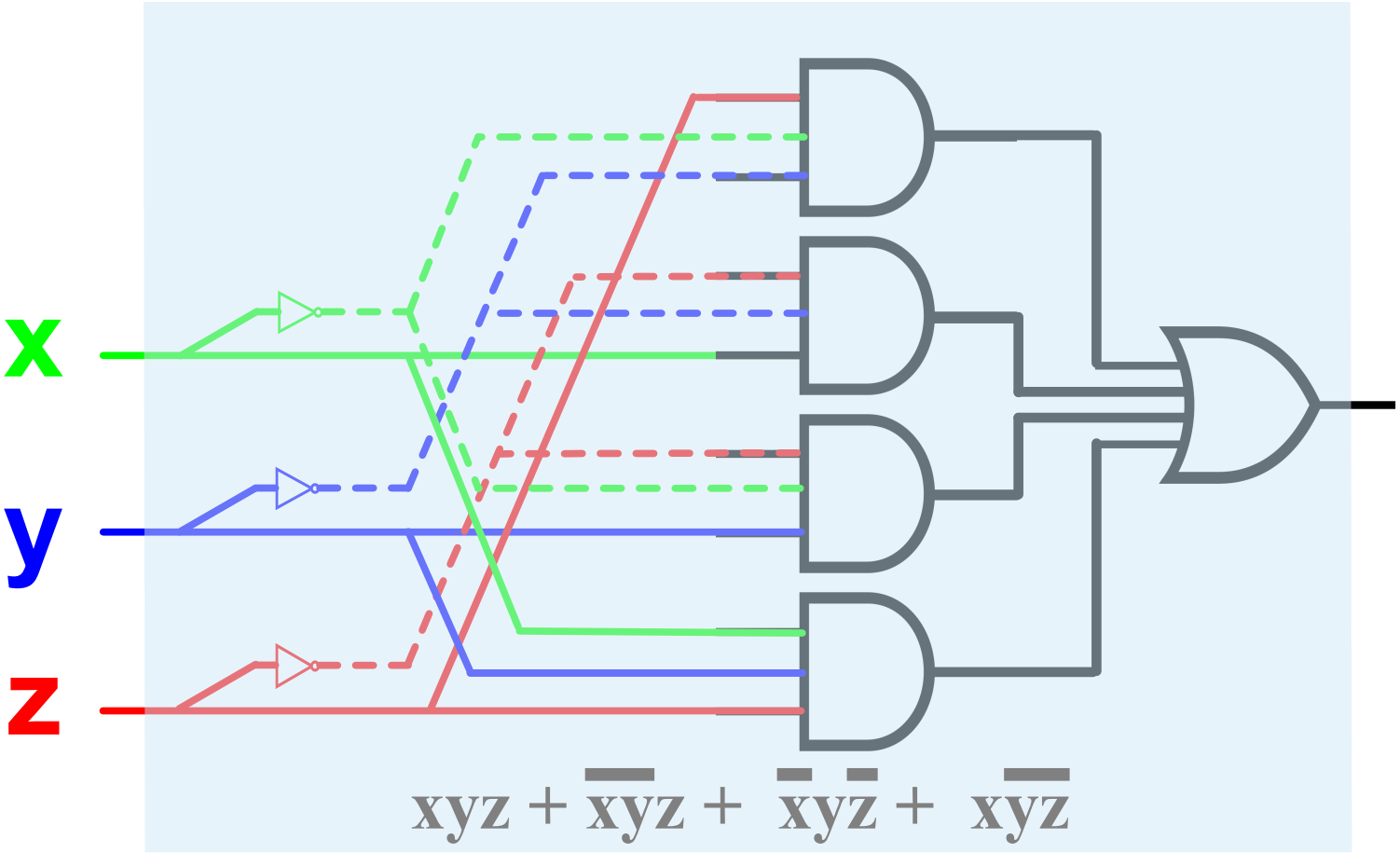
HW 4.1b: $xyz + \overline{x}\overline{y}z + \overline{x}y\overline{z} + x\overline{y}\overline{z}$

x	y	z	xyz	$\overline{x}\overline{y}z$	$\overline{x}y\overline{z}$	$x\overline{y}\overline{z}$	$xyz + \overline{x}\overline{y}z + \overline{x}y\overline{z} + x\overline{y}\overline{z}$
0	0	0	0	0	0	0	**0**
0	0	1	0	1	0	0	**1**
0	1	0	0	0	1	0	**1**
0	1	1	0	0	0	0	**0**
1	0	0	0	0	0	1	**1**
1	0	1	0	0	0	0	**0**
1	1	0	0	0	0	0	**0**
1	1	1	1	0	0	0	**1**

HW 4.1b: $xyz + \overline{xy}z + \overline{x}y\overline{z} + x\overline{yz}$

x	y	z	xyz	$\overline{xy}z$	$\overline{x}y\overline{z}$	$x\overline{yz}$	$xyz + \overline{xy}z + \overline{x}y\overline{z} + x\overline{yz}$
0	0	0	0	0	0	0	**0**
0	0	1	0	1	0	0	**1**
0	1	0	0	0	1	0	**1**
0	1	1	0	0	0	0	**0**
1	0	0	0	0	0	1	**1**
1	0	1	0	0	0	0	**0**
1	1	0	0	0	0	0	**0**
1	1	1	1	0	0	0	**1**

HW 4.1b: $xyz + \overline{x}\overline{y}z + \overline{x}y\overline{z} + x\overline{y}\overline{z}$

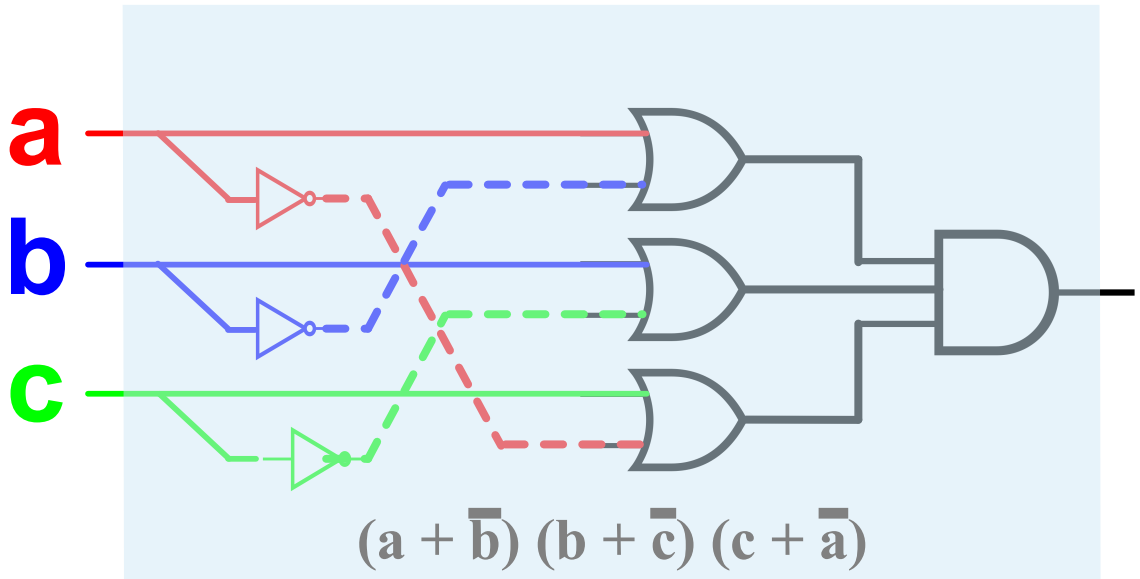


HW 4 1c: $(a + \overline{b})\,(b + \overline{c})\,(c + \overline{a})$

a	b	c	$a+\overline{b}$	$b+\overline{c}$	$c+\overline{a}$	$(a + \overline{b})\,(b + \overline{c})\,(c + \overline{a})$
0	0	0	1	1	1	**0**
0	0	1	1	0	1	**1**
0	1	0	0	1	1	**1**
0	1	1	0	1	1	**1**
1	0	0	1	1	0	**1**
1	0	1	1	0	1	**1**
1	1	0	1	1	0	**1**
1	1	1	1	1	1	**0**

HW 4 1c: $(a + \overline{b})\,(b + \overline{c})\,(c + \overline{a})$

a	b	c	$a+\overline{b}$	$b+\overline{c}$	$c+\overline{a}$	$(a + \overline{b})\,(b + \overline{c})\,(c + \overline{a})$
0	0	0	1	1	1	**0**
0	0	1	1	0	1	**1**
0	1	0	0	1	1	**1**
0	1	1	0	1	1	**1**
1	0	0	1	1	0	**1**
1	0	1	1	0	1	**1**
1	1	0	1	1	0	**1**
1	1	1	1	1	1	**0**

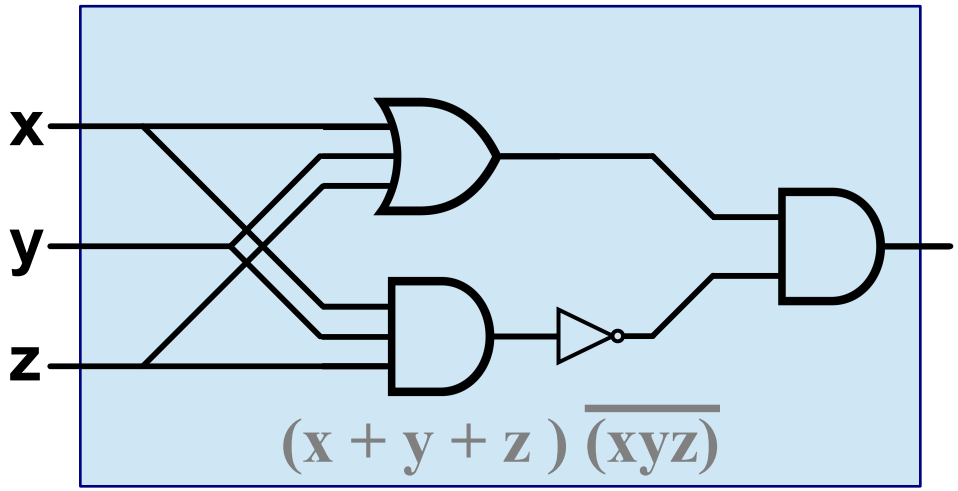


HW 4 1d: $(x + y + z)\,\overline{(xyz)}$

x	y	z	x+y+z	xyz	$\overline{xyz}$	$(x + y + z)\,\overline{(xyz)}$
0	0	0	0	0	1	**0**
0	0	1	1	0	1	**1**
0	1	0	1	0	1	**1**
0	1	1	1	0	1	**1**
1	0	0	1	0	1	**1**
1	0	1	1	0	1	**1**
1	1	0	1	0	1	**1**
1	1	1	1	1	0	**0**

HW 4 1d: $(x + y + z)\,\overline{(xyz)}$

x	y	z	x+y+z	xyz	$\overline{xyz}$	$(x + y + z)\,\overline{(xyz)}$
0	0	0	0	0	1	**0**
0	0	1	1	0	1	**1**
0	1	0	1	0	1	**1**
0	1	1	1	0	1	**1**
1	0	0	1	0	1	**1**
1	0	1	1	0	1	**1**
1	1	0	1	0	1	**1**
1	1	1	1	1	0	**0**



HW 4.2: $(\bar{x} + y)(y + \bar{z}) \equiv y + \overline{(x+z)}$

x	y	z	$\bar{x}+y$	$y+\bar{z}$	$(\bar{x}+y)(y+\bar{z})$	$y+\overline{(x+z)}$	$\overline{x+z}$	$x+z$
0	0	0	1	1	**1**	**1**	1	0
0	0	1	1	0	**0**	**0**	0	1
0	1	0	1	1	**1**	**1**	1	0
0	1	1	1	1	**1**	**1**	0	1
1	0	0	0	1	**0**	**0**	0	1
1	0	1	0	0	**0**	**0**	0	1
1	1	0	1	1	**1**	**1**	0	1
1	1	1	1	1	**1**	**1**	0	1

Combinational Circuit Design

Combinational circuit

Output determined by input

Design process

1. Specify semantics
   - Black Box input and output
   - Truth Table (input determines output)
2. Truth table → Boolean formula
3. **Minimize Boolean formula** (optional)
   - Boolean algebra
   - Karnaugh maps
4. **Boolean formula → combinational circuit**

D E F (Appendix A.2)

D is **true** iff at least one of A, B, and C is **true**

E is **true** iff exactly two of A, B, and C are **true**

F is **true** iff all three of A, B, and C are **true**

A	B	C	D	E	F
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	0	1

D E F *Truth Tables*

D = {1, 2, 3, 4, 5, 6, 7}

E = {3, 5, 6}

F = {7}

	A B C	D E F
0	0 0 0	
1	0 0 1	D
2	0 1 0	D
3	0 1 1	D E
4	1 0 0	D
5	1 0 1	D E
6	1 1 0	D E
7	1 1 1	D F

D E F *Truth Tables*

D = {1, 2, 3, 4, 5, 6, 7}

E = {3, 5, 6}

F = {7}

	A B C	D E F
0	0 0 0	
2	0 1 0	D
4	1 0 0	D
6	1 1 0	D E

	A B C	D E F
1	0 0 1	D
3	0 1 1	D E
5	1 0 1	D E
7	1 1 1	D F

D E F *Truth Tables*

D = {1, 2, 3, 4, 5, 6, 7}

E = {3, 5, 6}

F = {7}

	$\overline{C}$		C		
$\overline{A}\,\overline{B}$	0		1	D	
$\overline{A}B$	2	D	3	D	E
$A\overline{B}$	4	D	5	D	E
AB	6	D E	7	D	F

D E F *Truth Tables* (Gray Order)

D = {1, 2, 3, 4, 5, 6, 7}

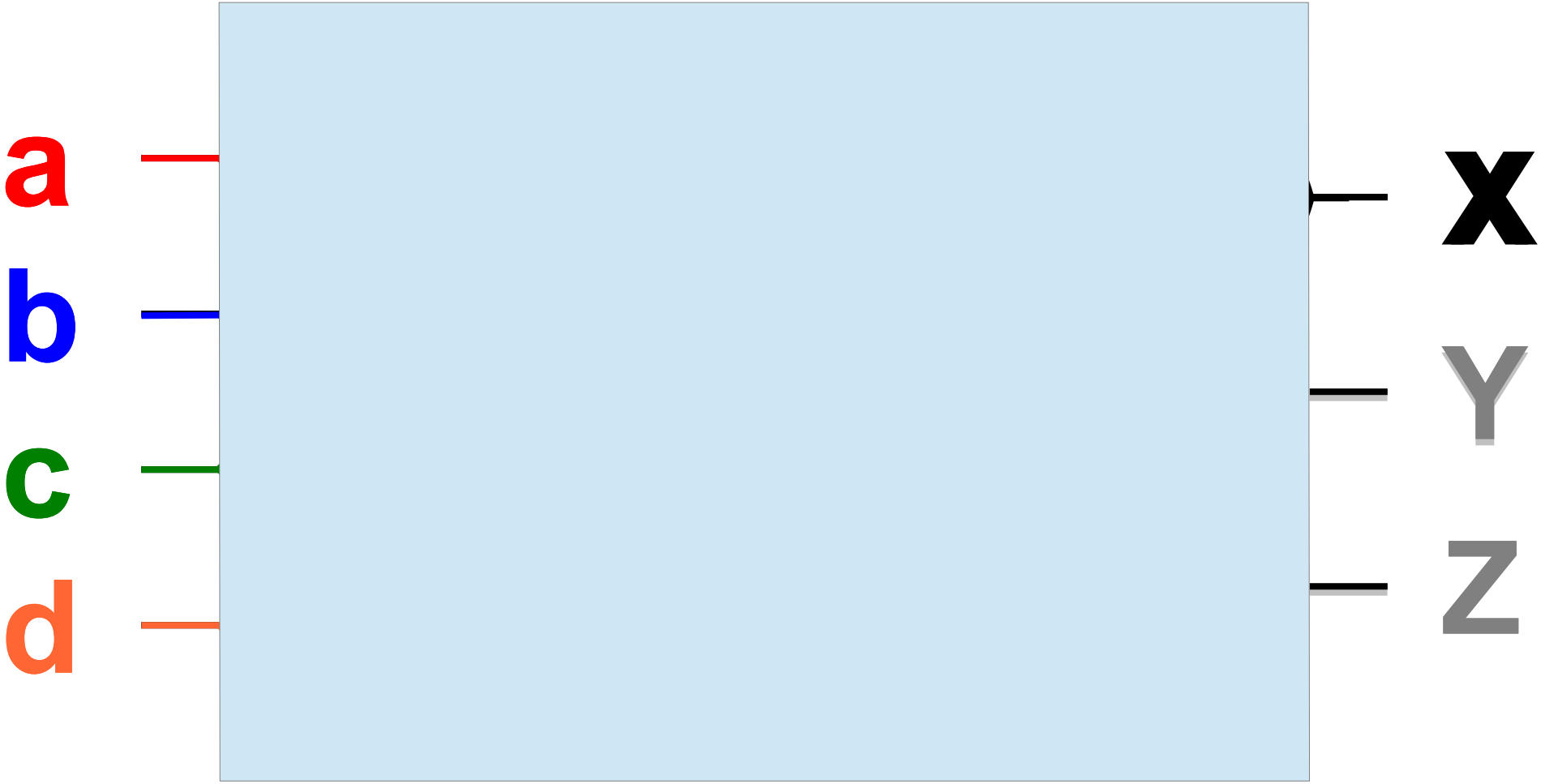
E = {3, 5, 6}

F = {7}

	$\overline{C}$		C	
$\overline{A}\,\overline{B}$	0		1	**D**
$\overline{A}B$	2	**D**	3	**D E**
AB	6	**D E**	7	**D F**
$A\overline{B}$	4	**D**	5	**D E**

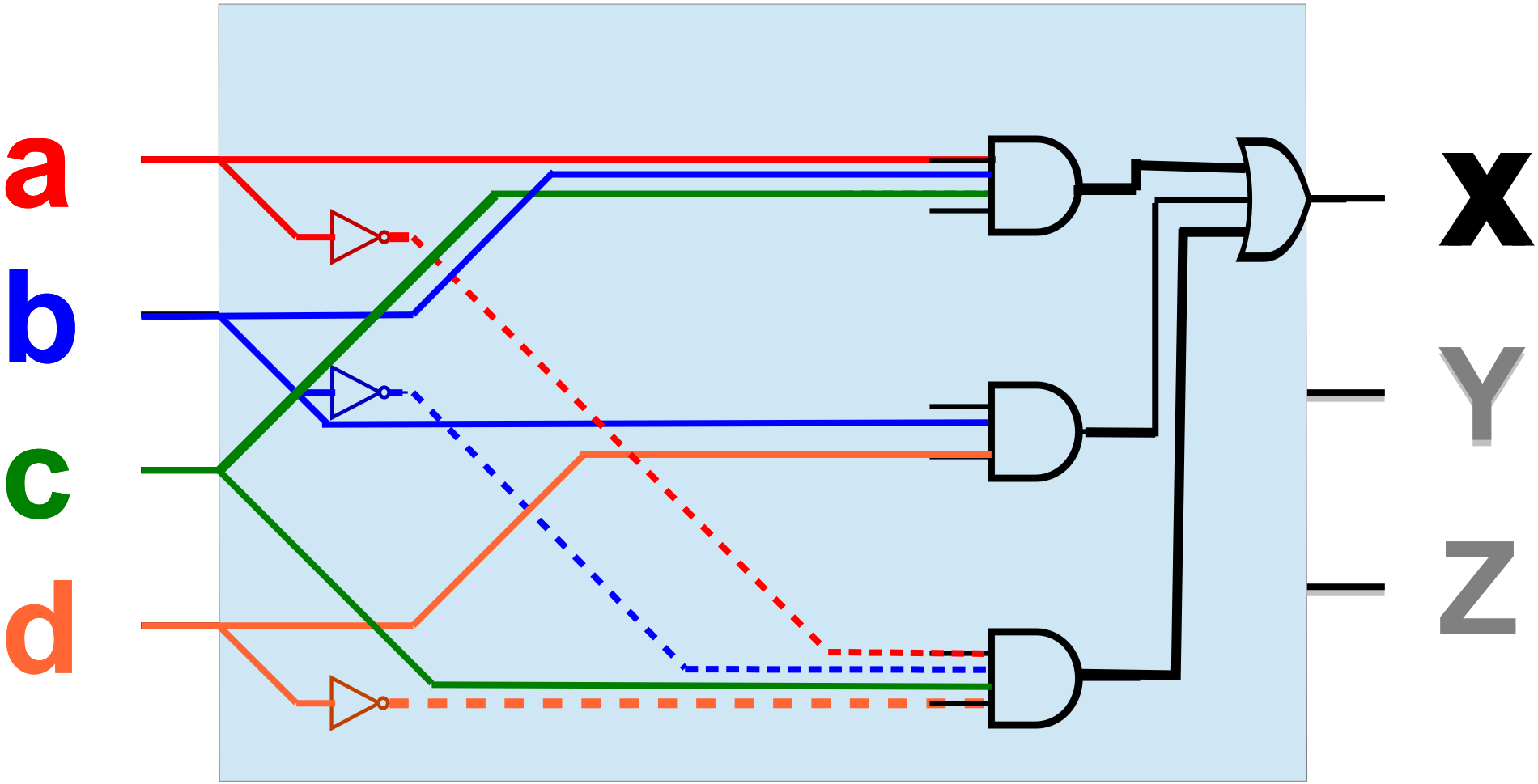
X Y Z

$$\underline{bd} + \underline{abc} + \underline{\overline{a}\overline{b}c\overline{d}}$$



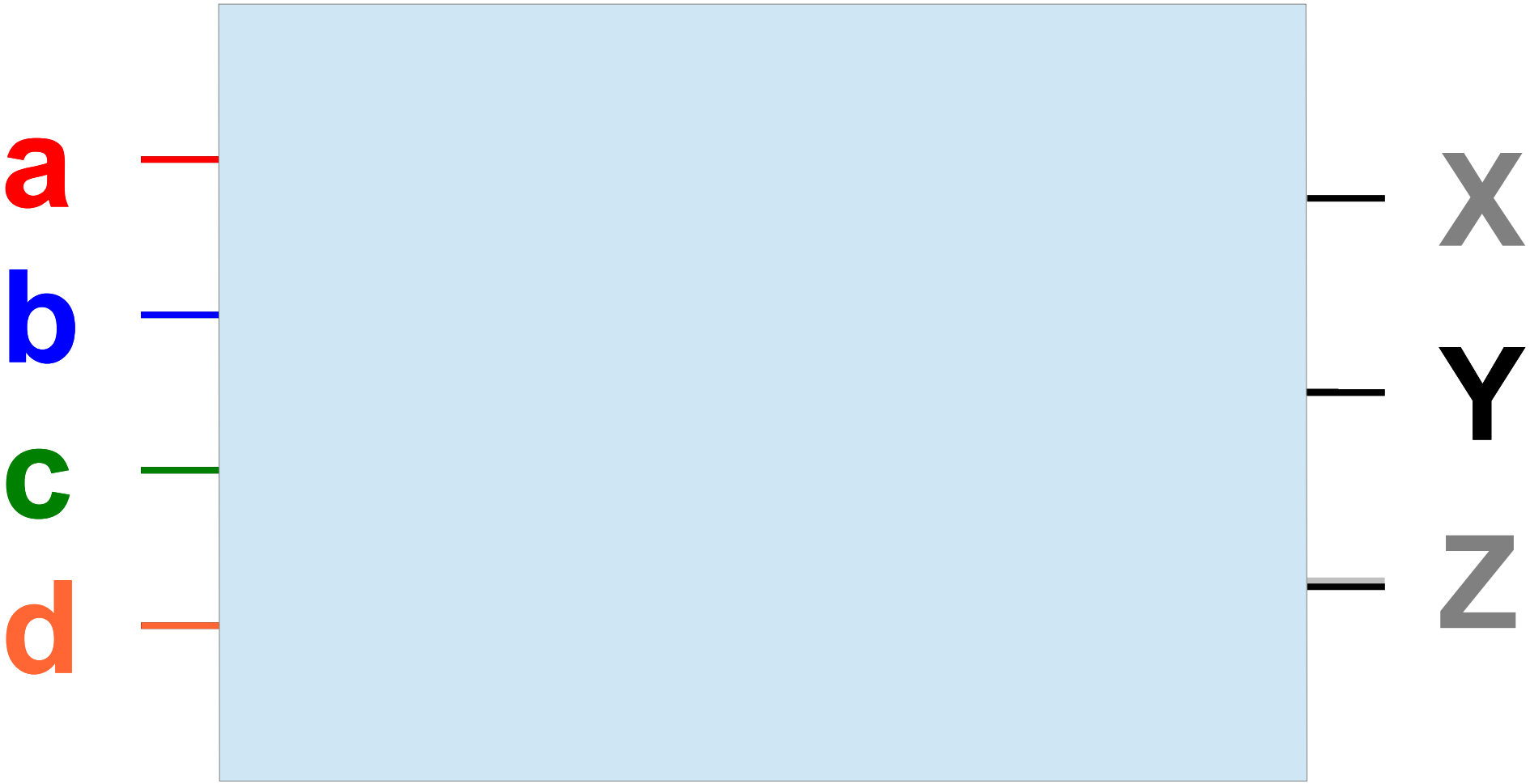
X Y Z

$$\underline{bd} + \underline{abc} + \underline{\overline{a}\overline{b}c\overline{d}}$$



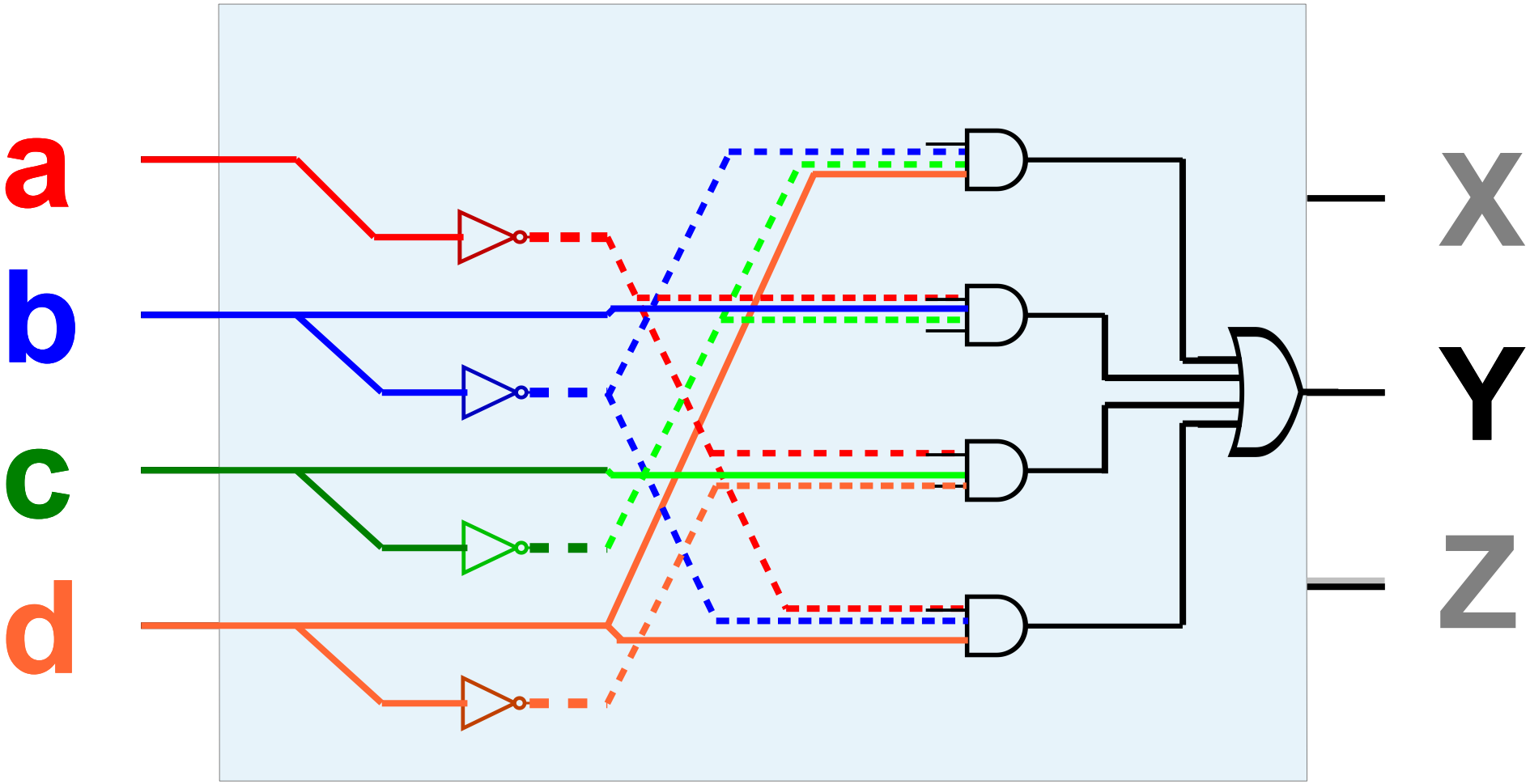
X Y Z

$$\overline{b}\,\overline{c}\,d + \overline{a}\,b\,\overline{c} + \overline{a}\,c\,\overline{d} + \overline{a}\,\overline{b}\,d$$



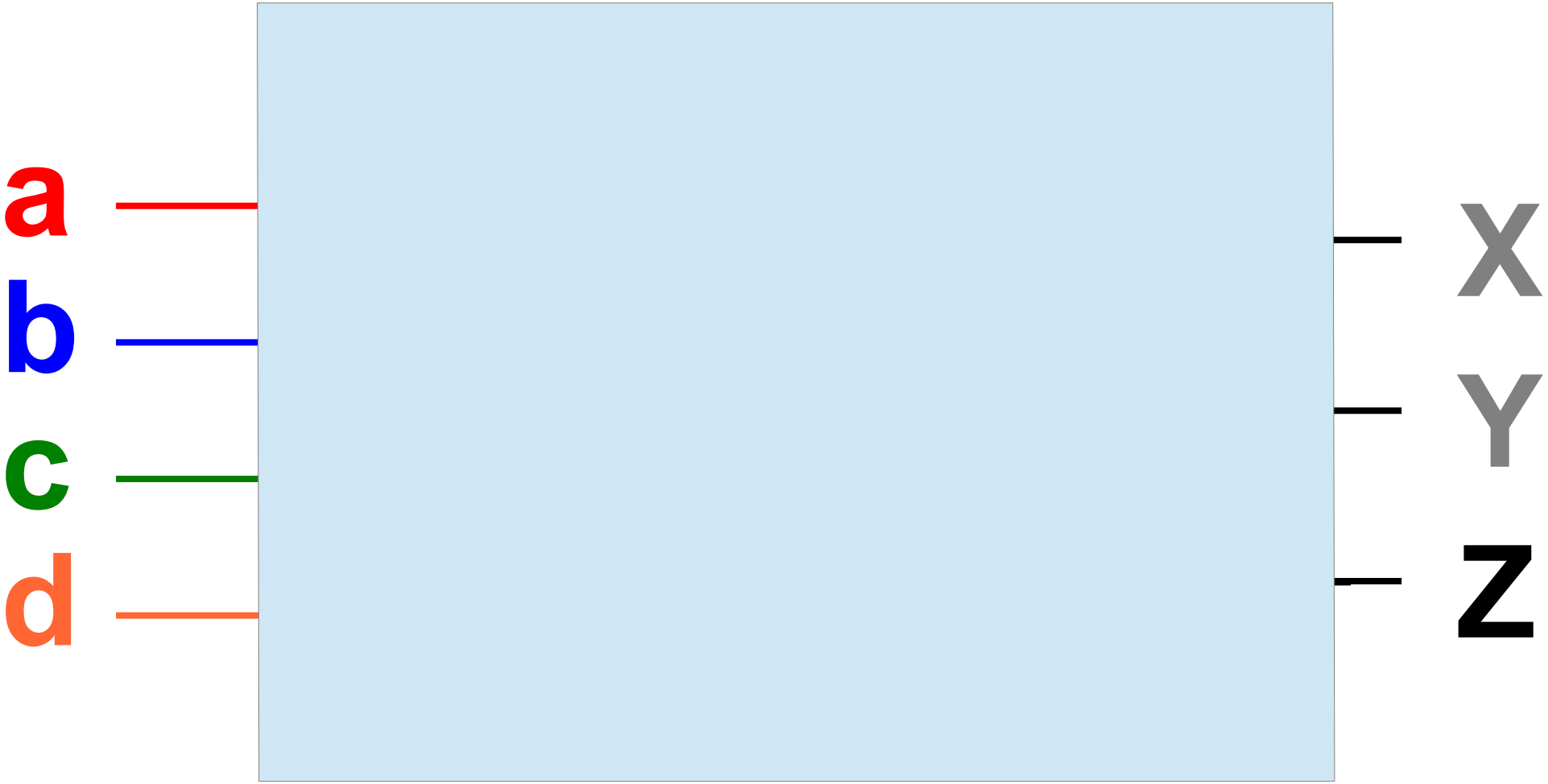
X Y Z

$$\overline{b}\,\overline{c}\,\overline{d} + \overline{a}\,b\,\overline{c} + \overline{a}\,c\,\overline{d} + \overline{a}\,\overline{b}\,\overline{d}$$



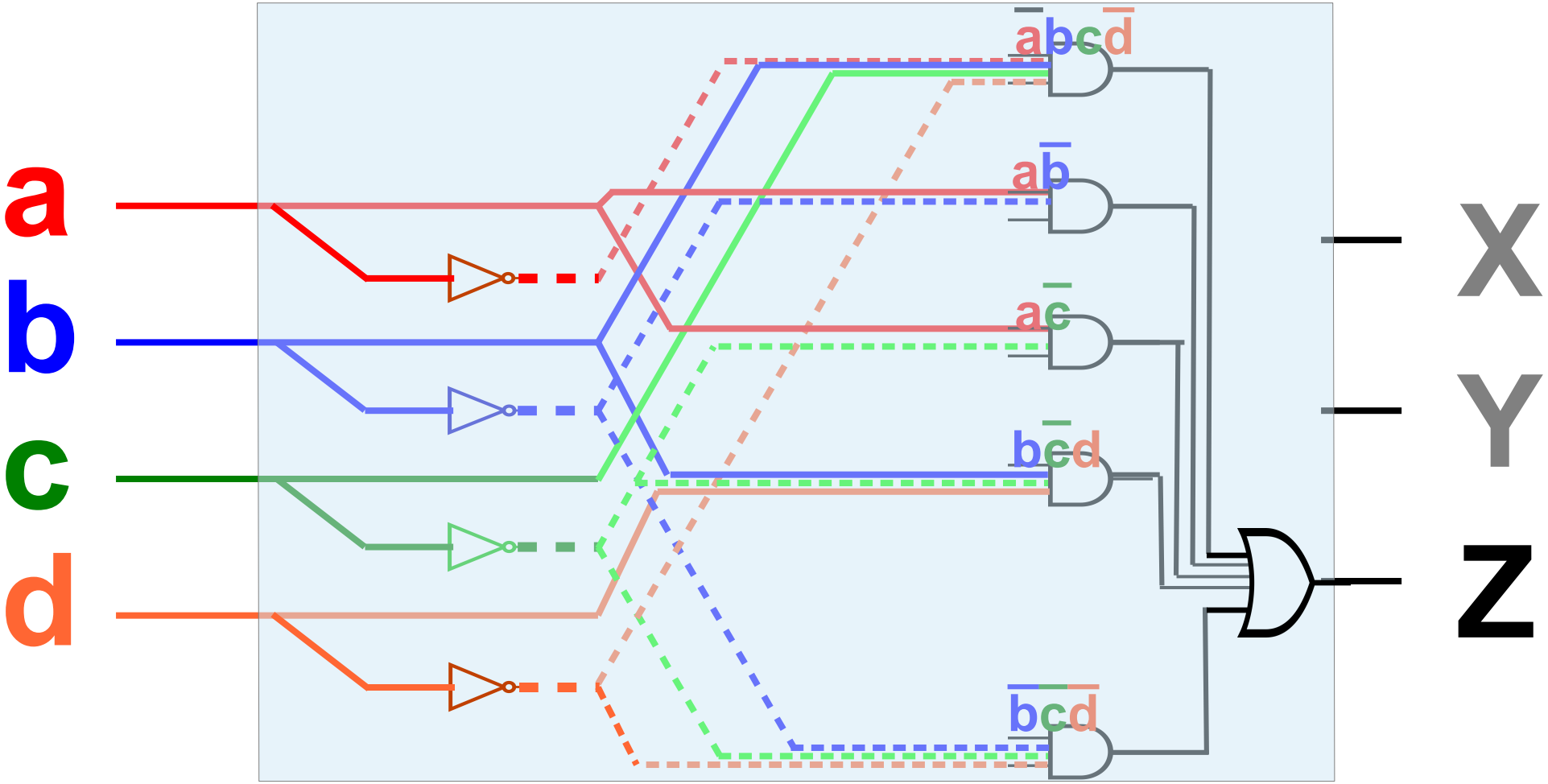
X Y Z

$$a\bar{c} + a\bar{b} + b\bar{c}d + \overline{bcd} + \bar{a}bc\bar{d}$$



X Y Z

$$a\bar{c} + a\bar{b} + b\bar{c}d + \overline{bcd} + \bar{a}bc\bar{d}$$



Some Boolean Formulas

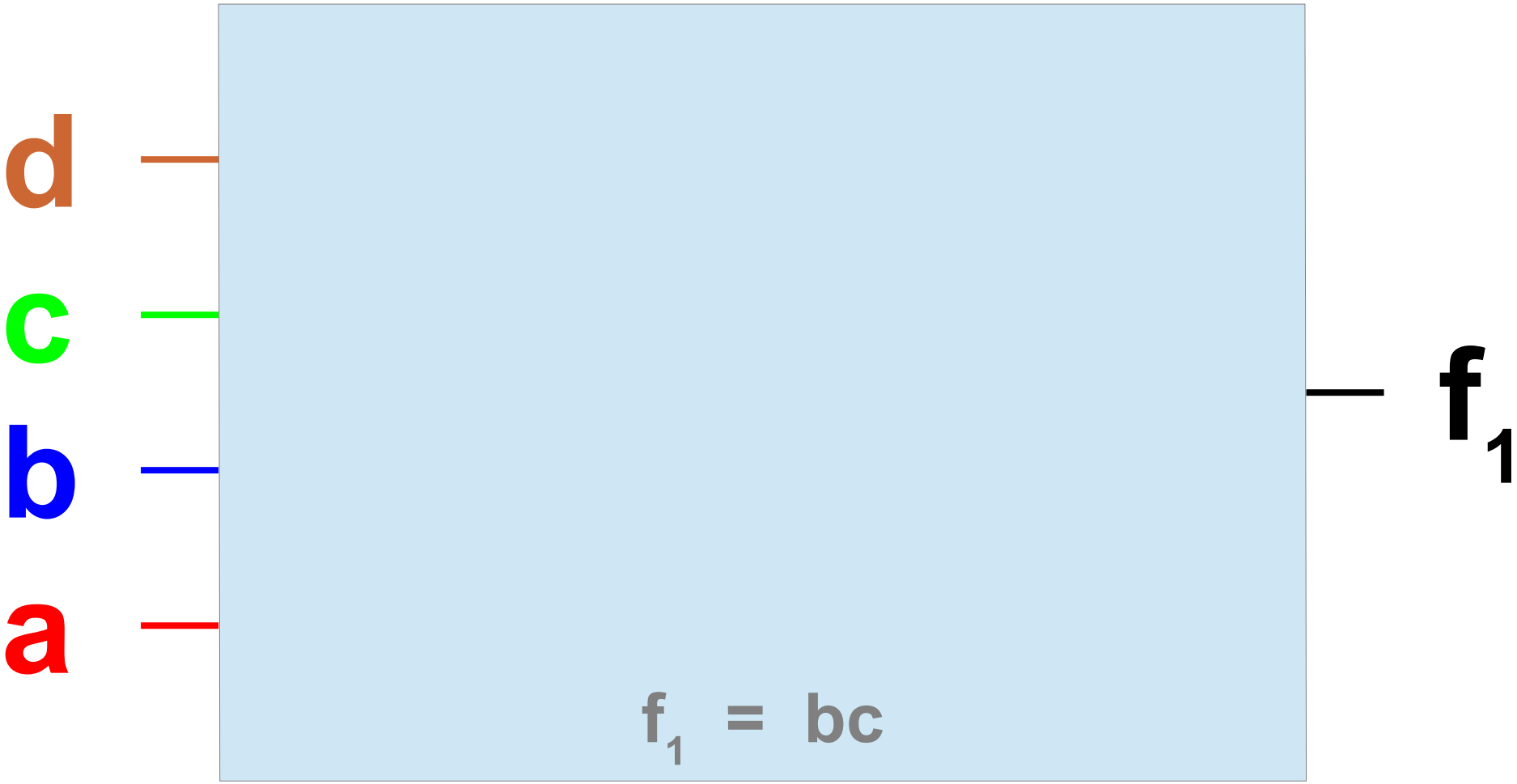
1) f_1 is {6, 7, E, F}
   $f_1 = bc$

2) f_2 is {0, 1, 5, 8, 9, D}
   $f_2 = \overline{bc} + \overline{c}d$

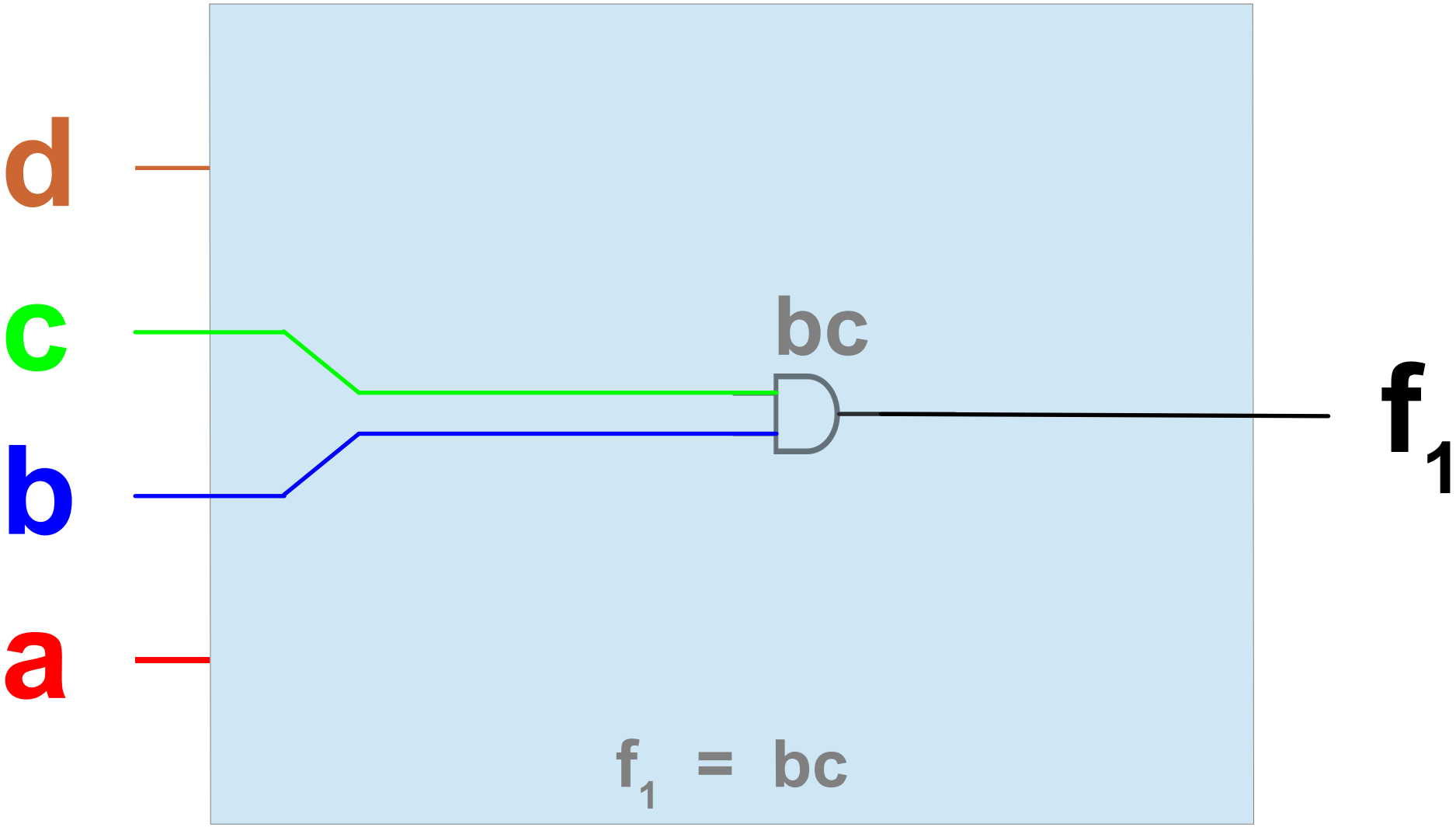
3) f_3 is {2, 3, 5, 7, B, D}
   $f_3 = \overline{abc} + \overline{a}bc + \overline{b}cd + b\overline{c}d$

4) f_4 is {0, 2, 6, 8, B, C, E}
   $f_4 = \overline{abd} + bc\overline{d} + a\overline{cd} + a\overline{b}cd$

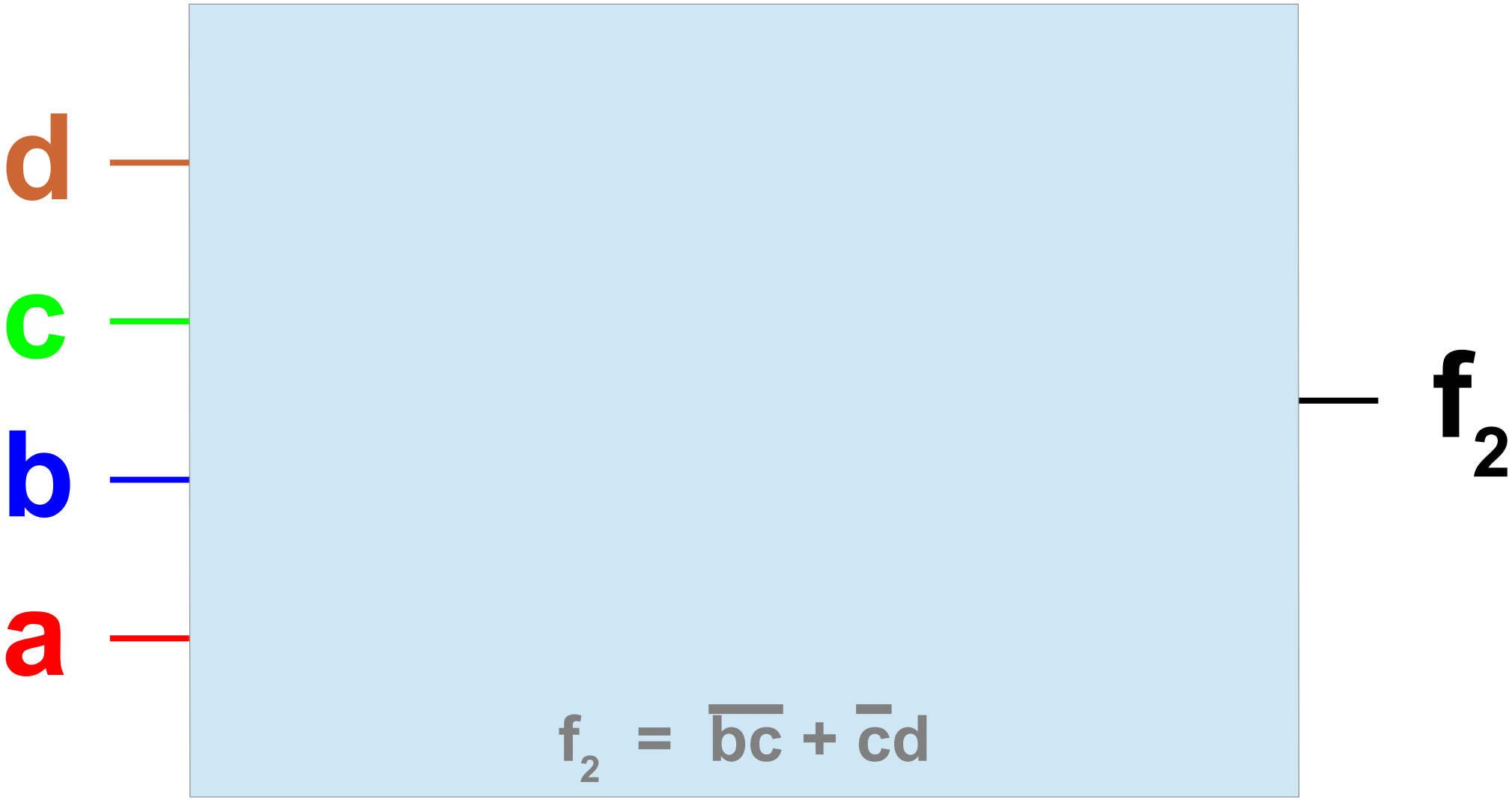
Circuit for f_1



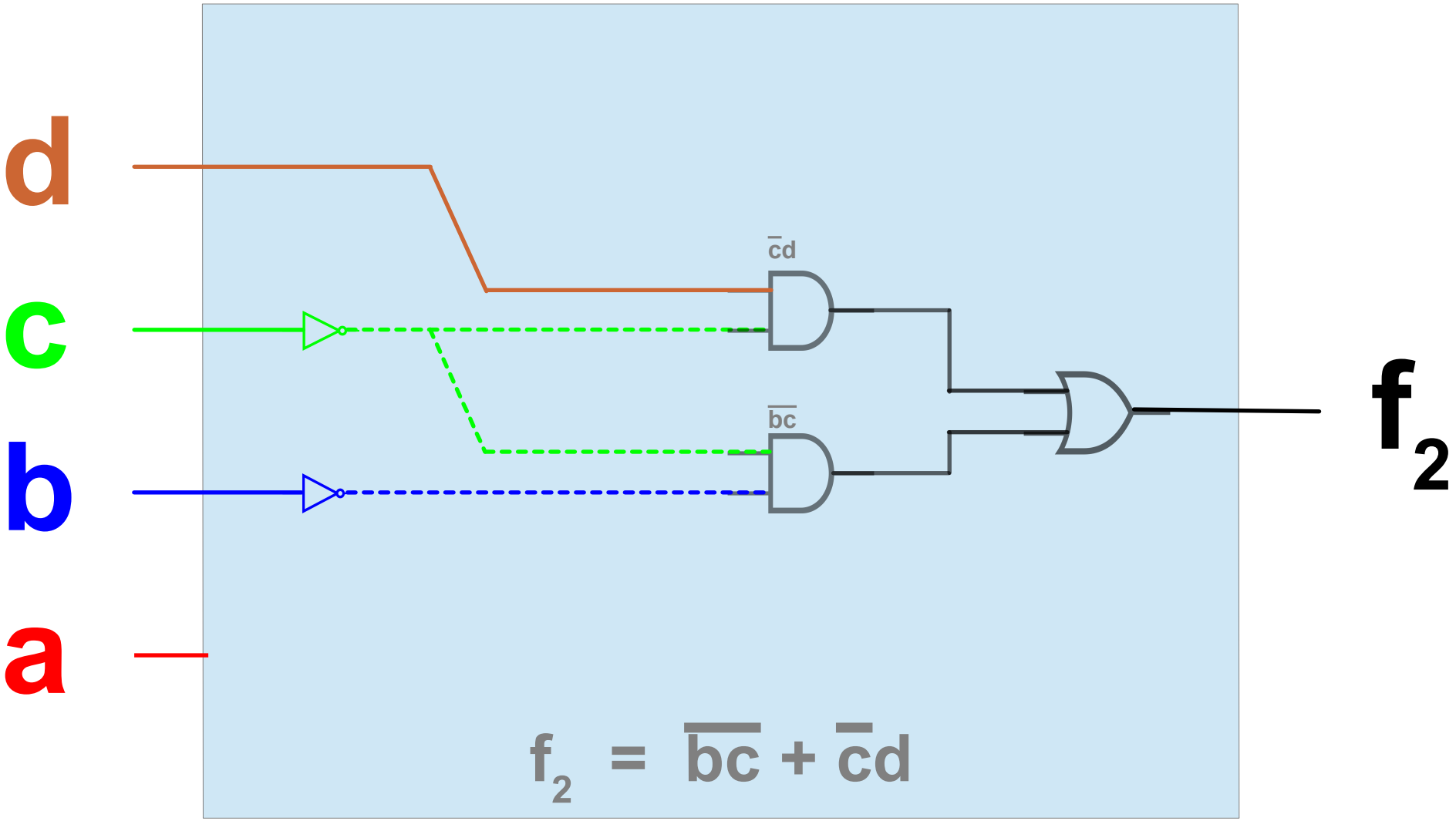
Circuit for f_1



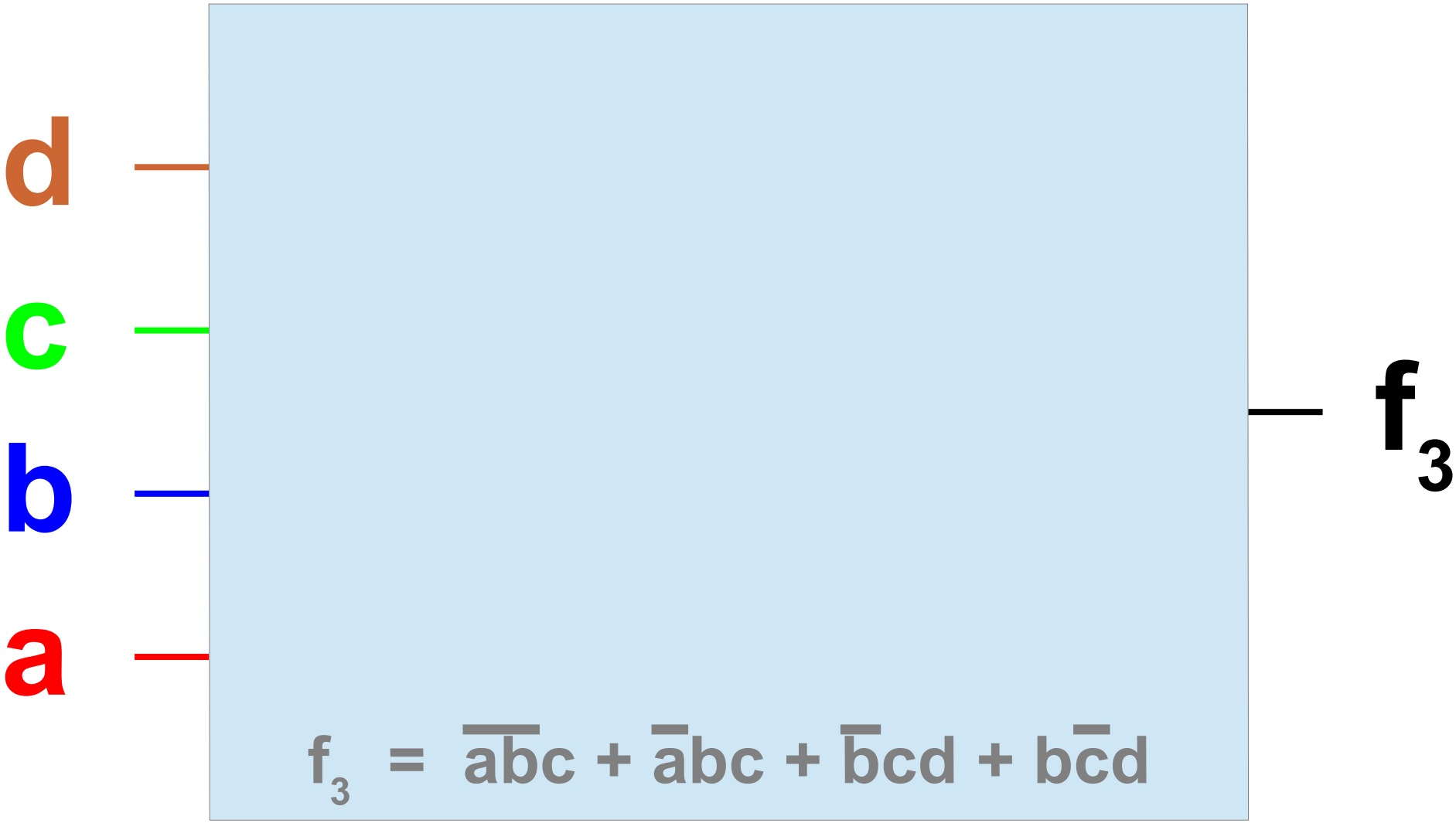
Circuit for f_2



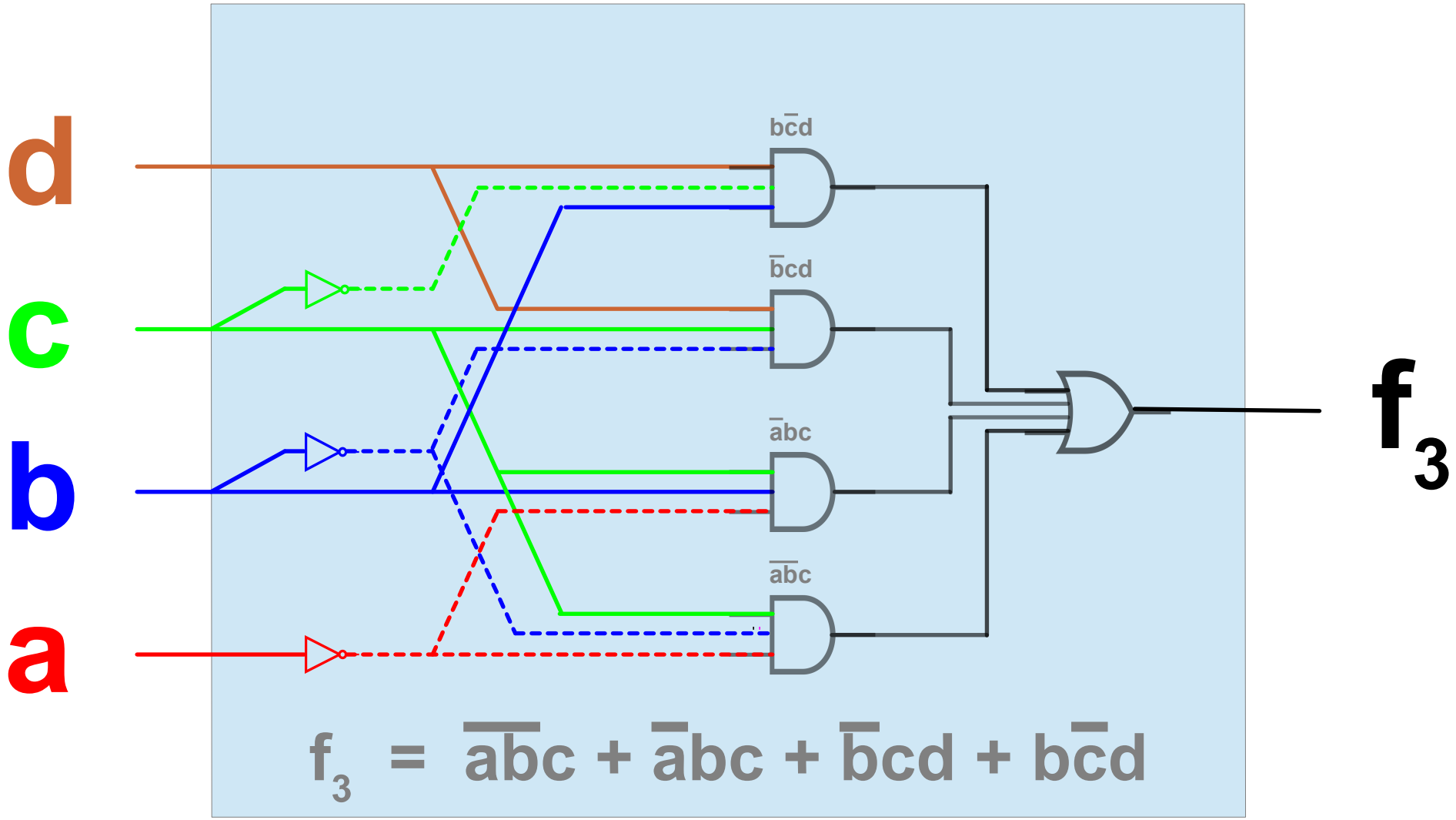
Circuit for f_2



Circuit for f_3

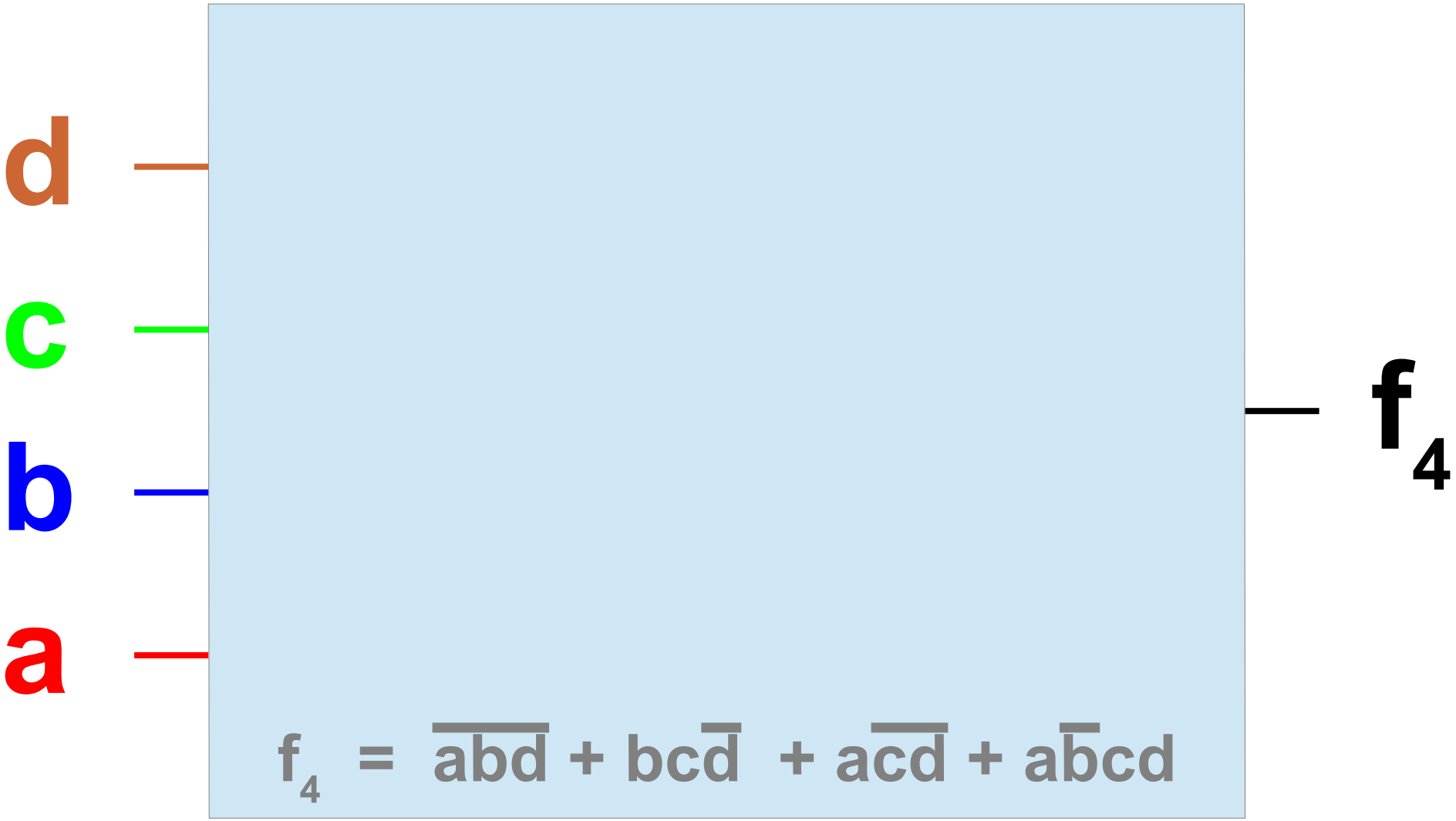


Circuit for f_3



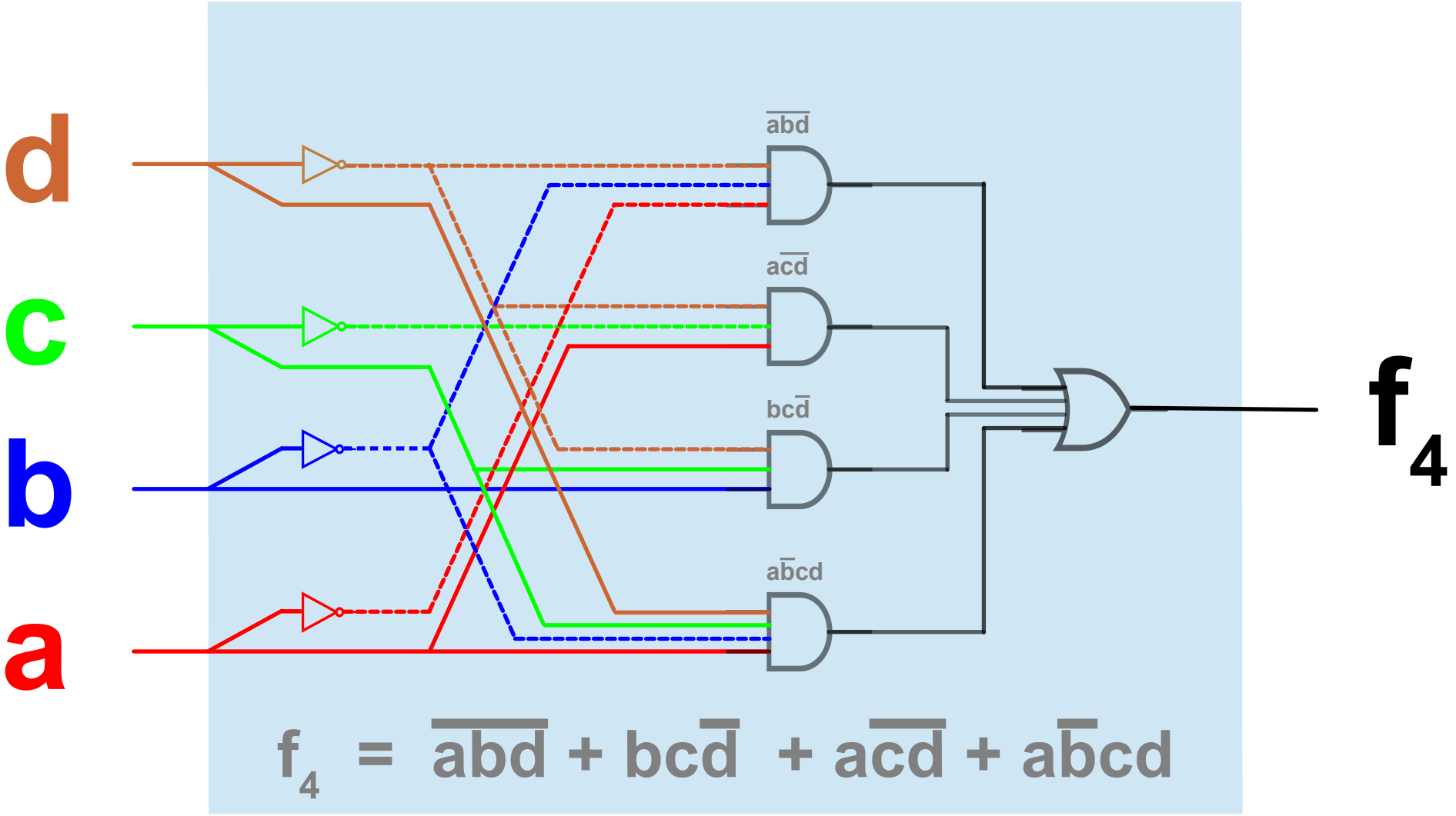
$$f_3 = \bar{a}\bar{b}c + \bar{a}bc + \bar{b}cd + b\bar{c}d$$

Circuit for f_4



$$f_4 = \overline{a}\overline{b}\overline{d} + bc\overline{d} + a\overline{c}\overline{d} + a\overline{b}cd$$

Circuit for f_4



Human Addition

$$
\begin{array}{r}
8\ 2\ 7\ 1\ 5\ 2\ 4 \\
+\quad 6\ 7\ 1\ 8\ 4\ 9\ 4 \\
\hline
\end{array}
$$

Human Addition

	8	2	7	1	5	2	4
+	6	7	1	8	4	9	4
						0	
							8

Human Addition

	8	2	7	1	5	2	4
+	6	7	1	8	4	9	4
					1	0	
						1	8

Human Addition

	8	2	7	1	5	2	4
+	6	7	1	8	4	9	4
				1	1	0	
					0	1	8

Human Addition

	8	2	7	1	5	2	4
+	6	7	1	8	4	9	4
			1	1	1	0	
				0	0	1	8

Human Addition

	8	2	7	1	5	2	4
+	6	7	1	8	4	9	4
		0	1	1	1	0	
			9	0	0	1	8

Human Addition

	8	2	7	1	5	2	4
+	6	7	1	8	4	9	4
	0	0	1	1	1	0	
		9	9	0	0	1	8

Human Addition

		8	2	7	1	5	2	4
+		6	7	1	8	4	9	4
	1	0	0	1	1	1	0	
		4	9	9	0	0	1	8

Human Addition

	8	2	7	1	5	2	4
+	6	7	1	8	4	9	4
1	0	0	1	1	1	0	
1	4	9	9	0	0	1	8

Human Addition (Binary)

$$
\begin{array}{r}
1\ 1\ 0\ 1\ 1\ 1\ 0 \\
+\quad 1\ 0\ 0\ 0\ 0\ 1\ 0 \\
\hline
\end{array}
$$

Human Addition (Binary)

	1	1	0	1	1	1	0
+	1	0	0	0	0	1	0
						0	
							0

Human Addition (Binary)

	1	1	0	1	1	1	0
+	1	0	0	0	0	1	0
					1	0	
						0	0

Human Addition (Binary)

	1	1	0	1	1	1	0
+	1	0	0	0	0	1	0
				1	1	0	
					0	0	0

Human Addition (Binary)

	1	1	0	1	1	1	0
+	1	0	0	0	0	1	0
			1	1	1	0	
				0	0	0	0

Human Addition (Binary)

		1	1	0	1	1	1	0
+		1	0	0	0	0	1	0
			0	1	1	1	0	
				1	0	0	0	0

Human Addition (Binary)

	1	1	0	1	1	1	0
+	1	0	0	0	0	1	0
	0	0	1	1	1	0	

1	1	0	0	0	0

Human Addition (Binary)

		1	1	0	1	1	1	0
+		1	0	0	0	0	1	0
	1	0	0	1	1	1	0	
		0	1	1	0	0	0	0

Human Addition (Binary)

		1	1	0	1	1	1	0
+		1	0	0	0	0	1	0
	1	0	0	1	1	1	0	
	1	0	1	1	0	0	0	0

Fixed Width Binary Addition

	1	1	1	0
+	0	0	1	0

Fixed Width Binary Addition

	1	1	1	0
+	0	0	1	0
			0	
				0

Fixed Width Binary Addition

	1	1	1	0
+	0	0	1	0
		1	0	
			0	0

Fixed Width Binary Addition

	1	1	1	0
+	0	0	1	0
	1	1	0	
		0	0	0

Fixed Width Binary Addition

	1	1	1	0
+	0	0	1	0
1	1	1	0	
	0	0	0	0

Fixed Width Binary Addition

	1	1	1	0
+	0	0	1	0
1	1	1	0	
	0	0	0	0

Carry out

Fixed Width Binary Addition

		1	1	1	0
+		0	0	1	0
	1	1	1	0	0
		0	0	0	0

Carry out

Carry in

What about Subtraction? **A – B**

A ≥ B

A ~ B ≡ *A – B (ordinary arithmetic)*

A < B

1) **B ~ B = 0**
2) **(A ~ B) + C = (A + C) ~ B**

~B ≡ 0 ~ B = $2^n - B = \overline{B} + 1$ (2's complement)

$$\boxed{A \sim B \equiv A + \overline{B} + 1}$$

Fixed Width Binary Addition

	1	1	1	0
+	1	1	1	0
	1	1	0	1
			Carry in	
	0	0	0	0

Arithmetic / Logical Unit

Building Blocks

- AND, OR, and NOT gates
- ***Inverters, Decoders, Multiplexers***

Inputs (*operands*): **A** and **B** buses

Output (*result*): **C** bus

- Logical
  - Bitwise: $\overline{\mathbf{A}}$, **A** & **B**, **A** | **B**, **A** ^ **B**, **A** ↑ **B**, **A** ↓ **B**, ...
- Arithmetic
  - **A** + **B**, **A** – **B**, **A** · **B**, **A** div **B**, **A** mod **B**
- Comparison
  - **A** < **B**, **A** = **B**, **A** ≥ **B**, etc. and **X** < 0, **X** = 0, **X** ≥ 0, etc.

TINY Arithmetic / Logical Unit

Building Blocks

- AND, OR, and NOT gates
- ***Inverters, Decoders, Multiplexers***

Inputs (*operands*): **A** and **B** buses

Output (*result*): **C** bus

- Logical
  - Bitwise: $\overline{\mathbf{A}}$, **A** & **B**, **A** | **B**, **A** ^ **B**, **A** ↑ **B**, **A** ↓ **B**, ...
- Arithmetic
  - **A** + **B**, **A** – **B**, **A** · **B**, **A** div **B**, **A** mod **B**
- Comparison
  - **A** < **B**, **A** = **B**, **A** ≥ **B**, etc. and **X** < 0, **X** = 0, **X** ≥ 0, etc.

Inverters, Decoders, Multiplexer

Inverter: select data input *or its negation*

- 1 data input
- 1 selector input
- 1 output

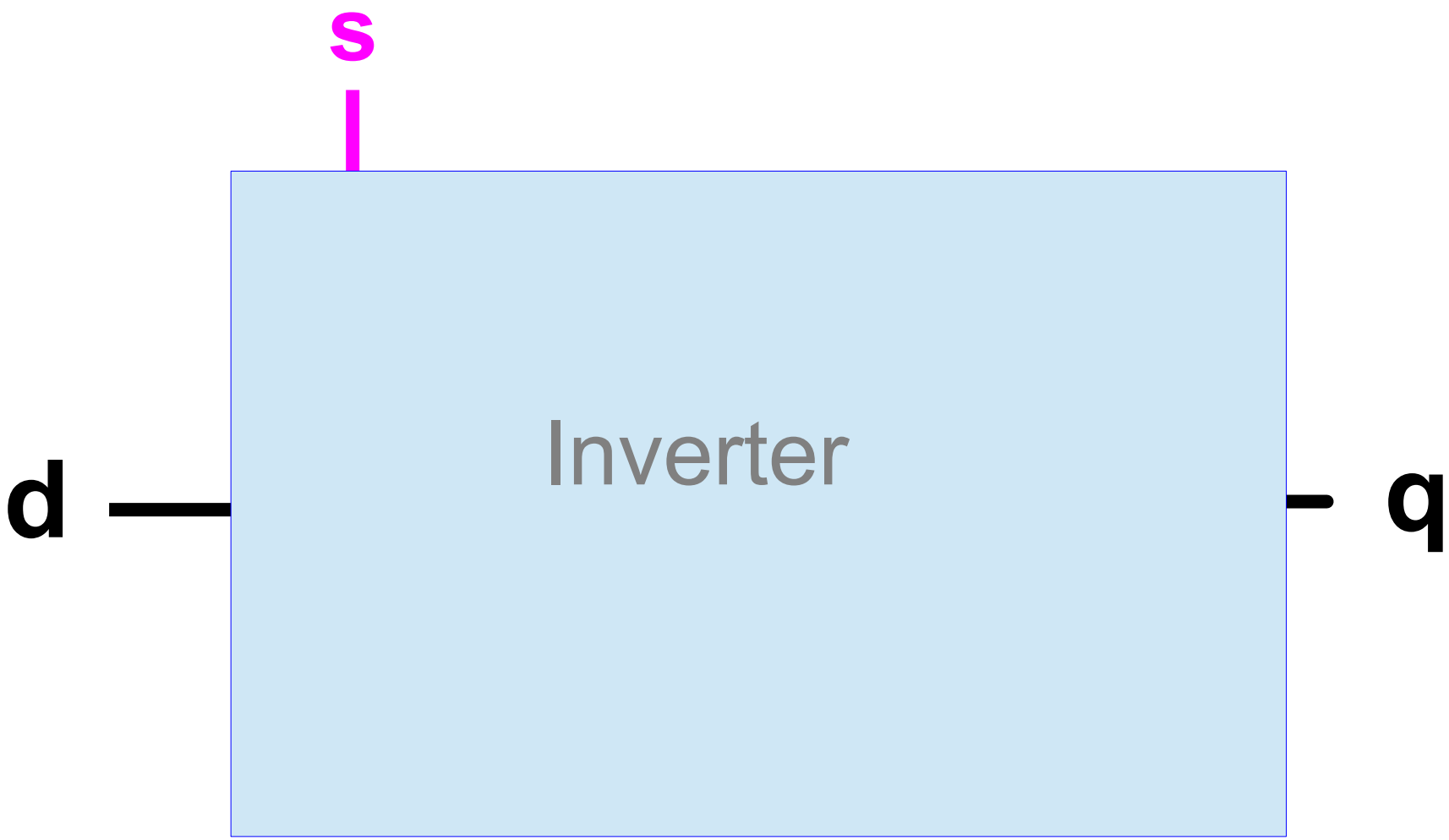
Decoder: select unique output to be 1 (true)

- N selector inputs
- 2^N outputs

Multiplexer: select unique data input to be output

- 2^N data inputs
- N selector inputs
- 1 output

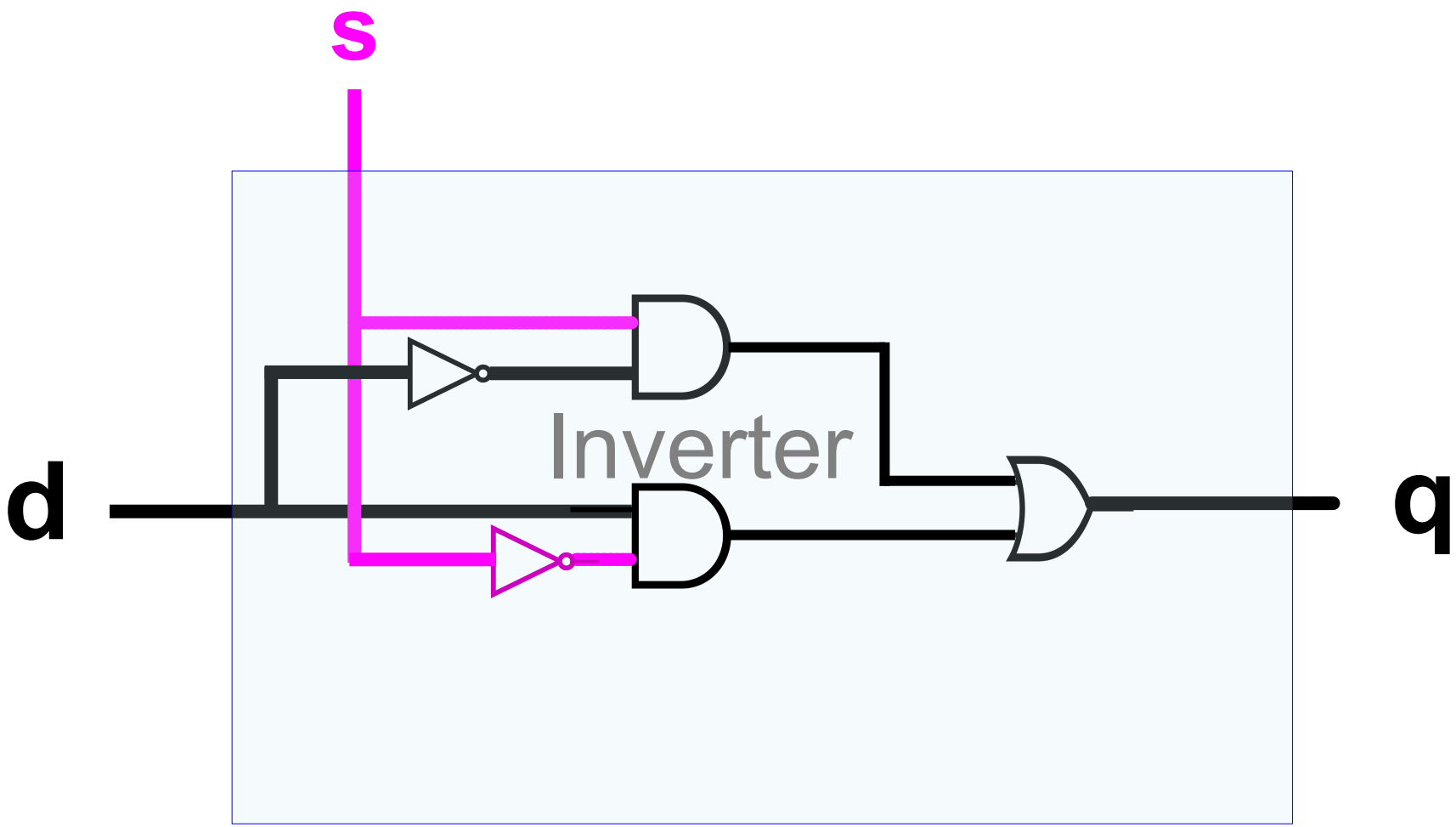
Inverter



Inverter Truth Table

#	s	d	q
0	0	0	**0**
1	0	1	**1**
2	1	0	**1**
3	1	1	**0**

Inverter



N-Bit Decoder

Each different combination of N input bits uniquely specifies one of 2^N outputs. An output is 1 if and only if the corresponding input combination is active. For any input, exactly one output is 1.

N-Bit Decoder Truth Table:

- N input columns
- 2^N output columns
- 2^N rows

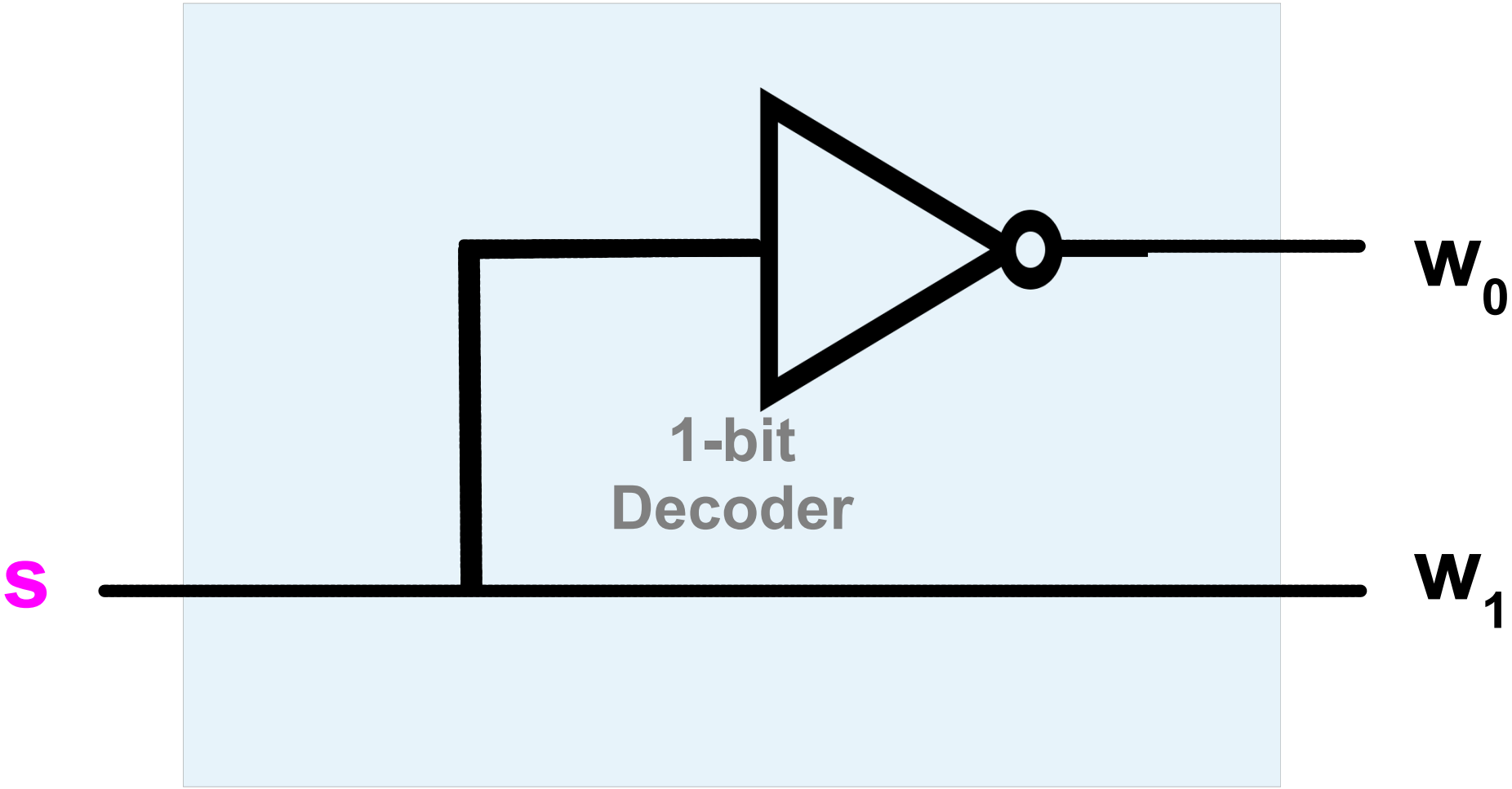
Exactly one 1 in each output column

Exactly one 1 in each output row

1-Bit Decoder Truth Table

#	**S**	$\mathbf{x_0}$	$\mathbf{x_1}$
0	0	1	**0**
1	1	0	**1**

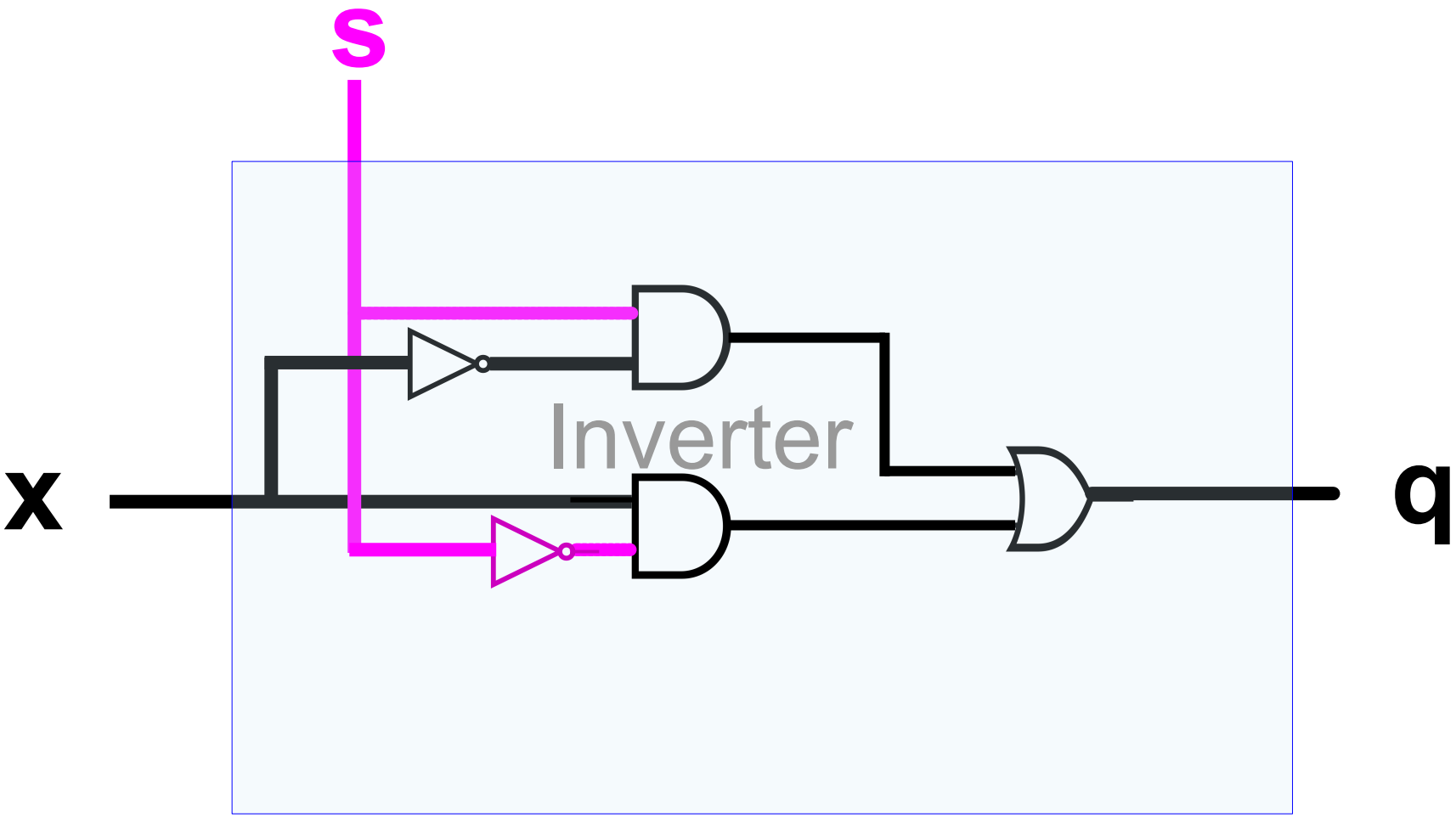
1-Bit Decoder



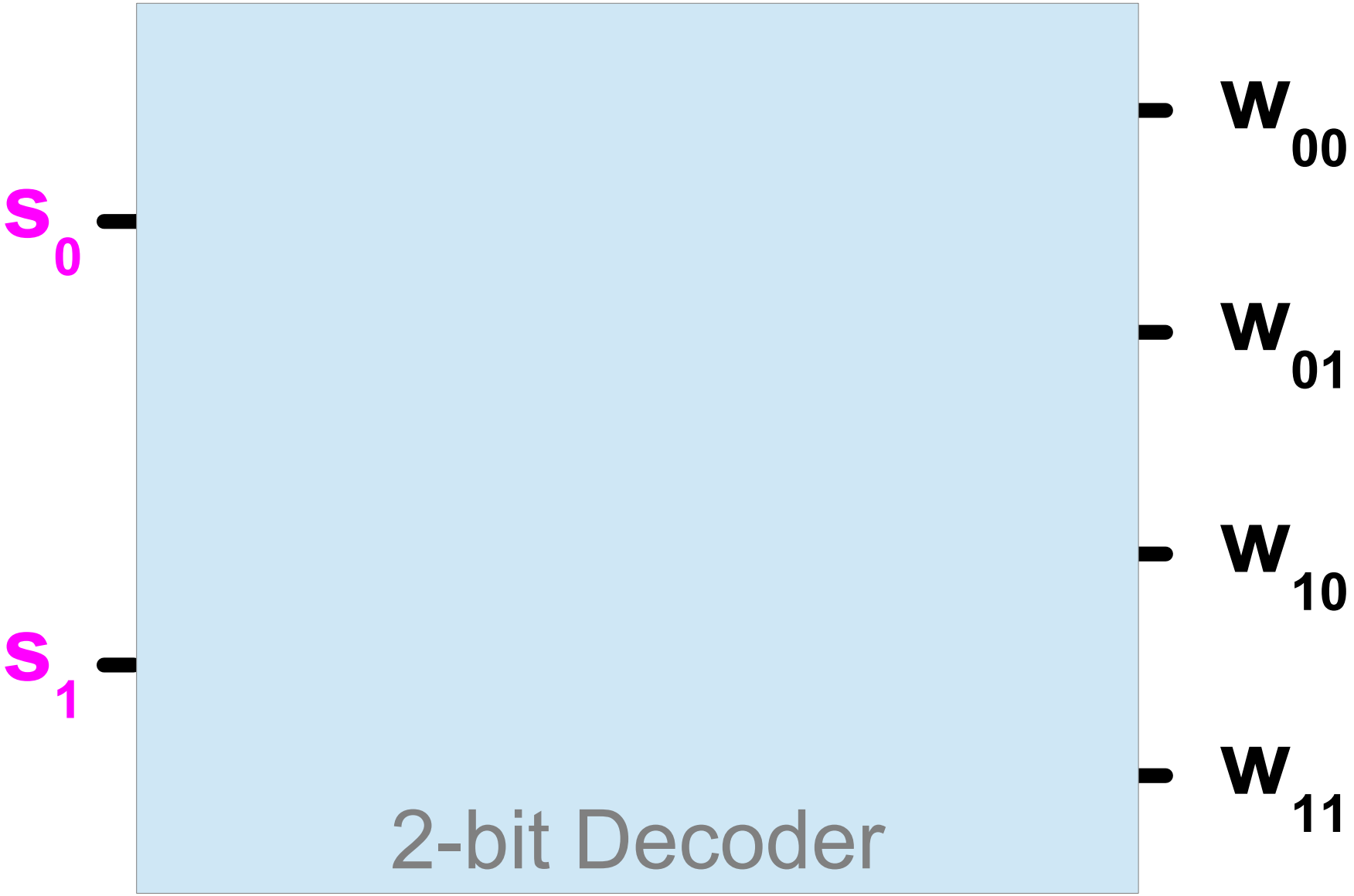
Inverter Truth Table

#	s	d	q
0	0	0	**0**
1	0	1	**1**
2	1	0	**1**
3	1	1	**0**

Inverter



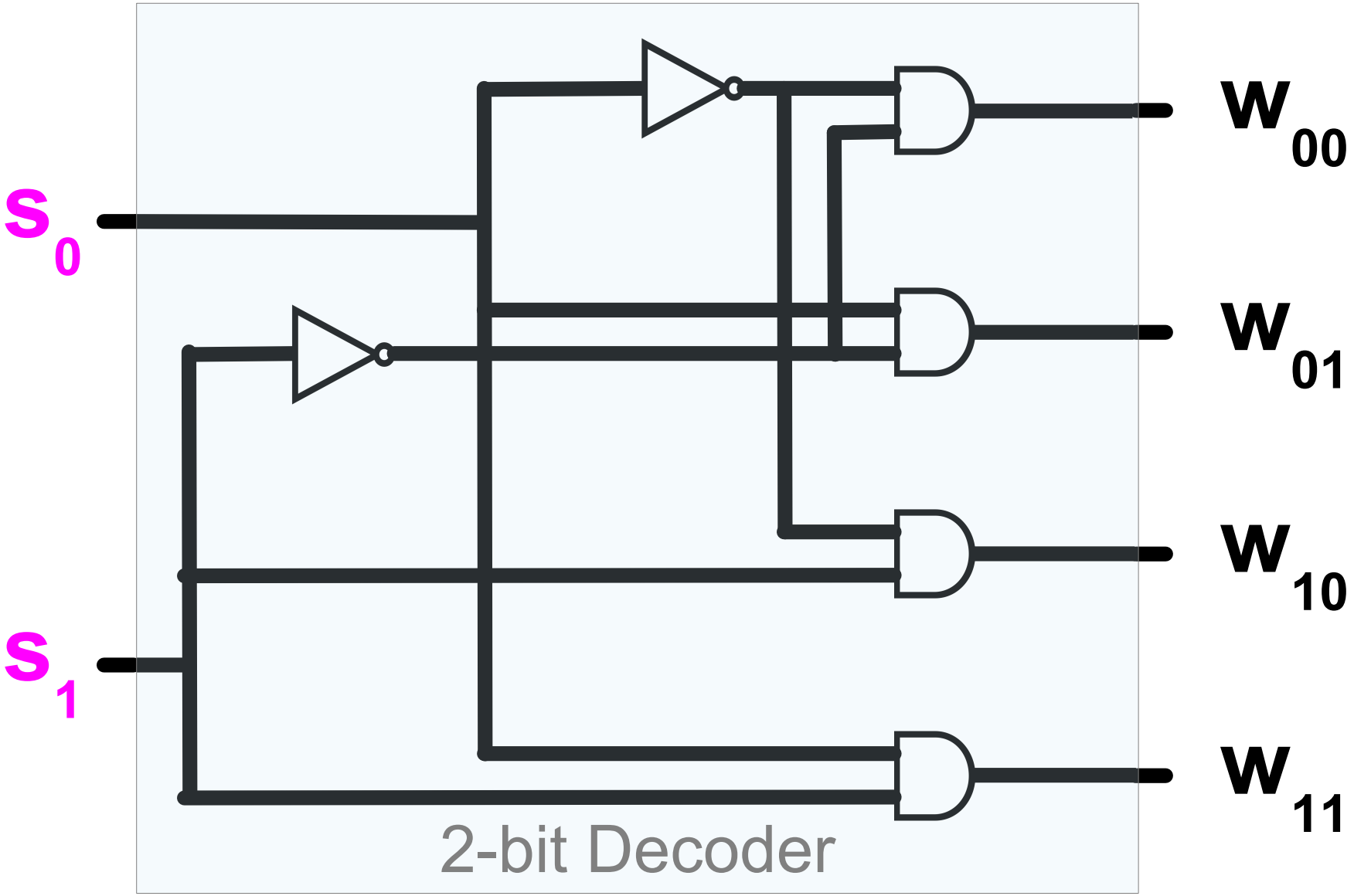
2-Bit Decoder



2-bit Decoder Truth Table

#	s_1	s_0	w_{00}	w_{10}	w_{10}	w_{11}
0	0	0	**1**	**0**	**0**	**0**
1	0	1	**0**	**1**	**0**	**0**
2	1	0	**0**	**0**	**1**	**0**
3	1	1	**0**	**0**	**0**	**1**

2-Bit Decoder



3-bit Decoder Truth Table

s_2	s_1	s_0	w_{000}	w_{010}	w_{010}	w_{011}	w_{100}	w_{111}	w_{110}	w_{111}
0	0	0	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	0	1	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
0	1	0	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**
0	1	1	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**
1	0	0	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**
1	0	1	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
1	1	0	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**
1	1	1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**

M-way Multiplexer ($M \equiv 2^N$)

N (lg M) selector inputs choose one of 2^N (M) data inputs to output

2^N-way multiplexer truth table:

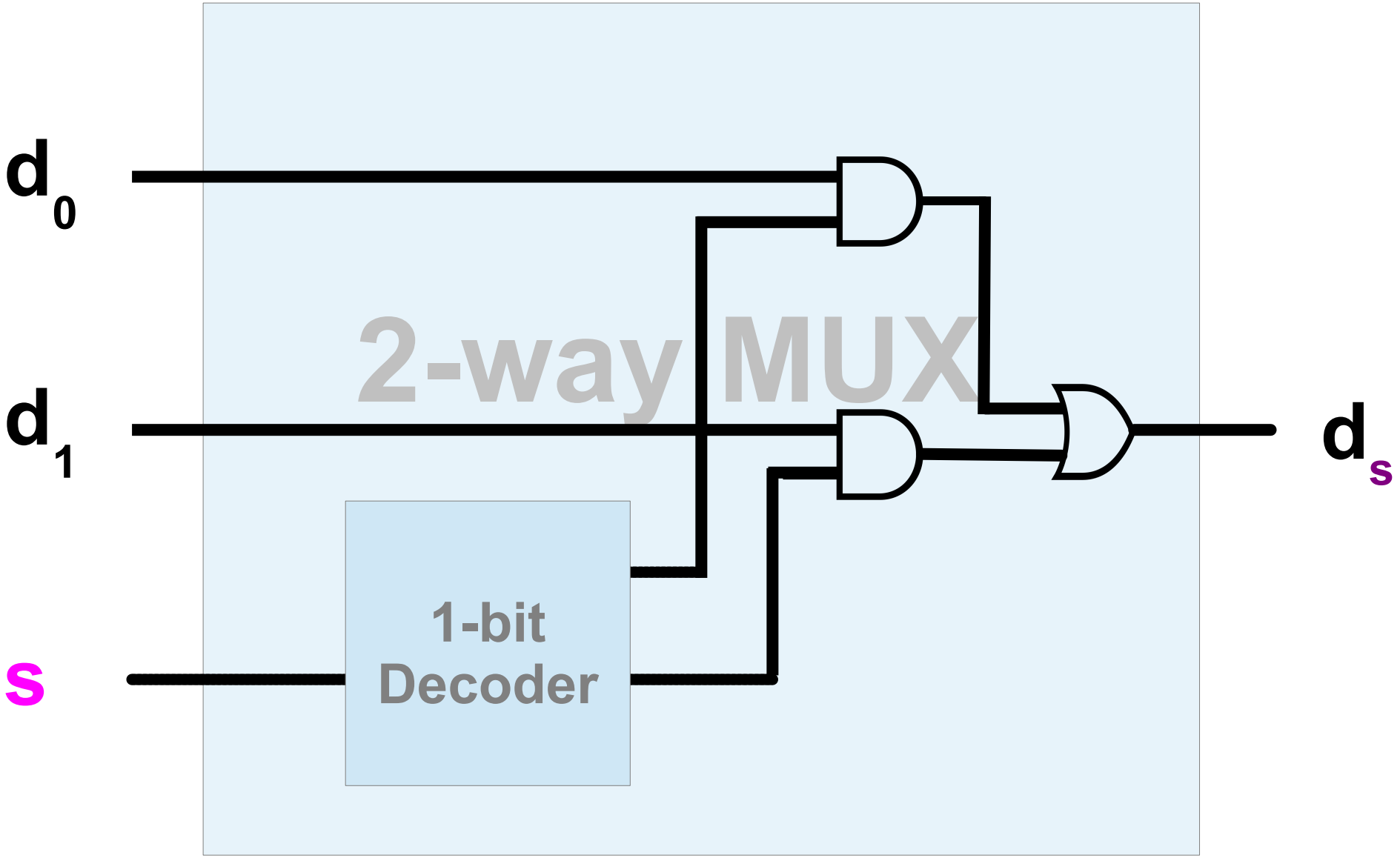
2^N data input columns	($\mathbf{d}_i$ $0 \le \mathbf{i} < 2^N$)
N selector input columns	($\mathbf{s}_j$ $0 \le \mathbf{s} < 2^N$)
1 output column	($\mathbf{x} = \mathbf{d}_s$)

2-Way **Multiplexer**

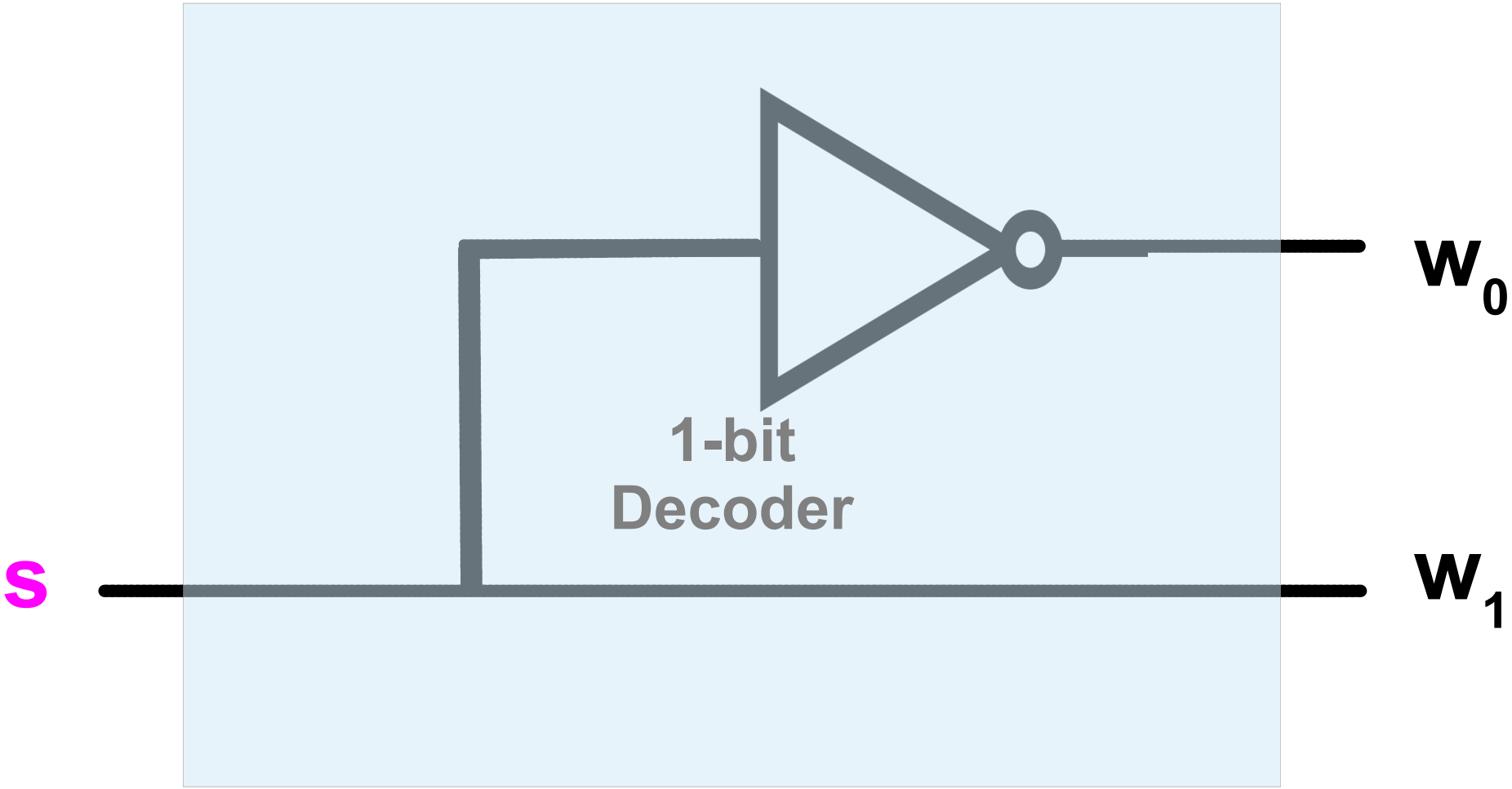
2-Way **Multiplexer**

#	s	d_1	d_0	q
0	**0**	0	**0**	**0**
1	**0**	0	**1**	**1**
2	**0**	1	**0**	**0**
3	**0**	1	**1**	**1**
4	**1**	**0**	0	**0**
5	**1**	**0**	1	**0**
6	**1**	**1**	0	**1**
7	**1**	**1**	1	**1**

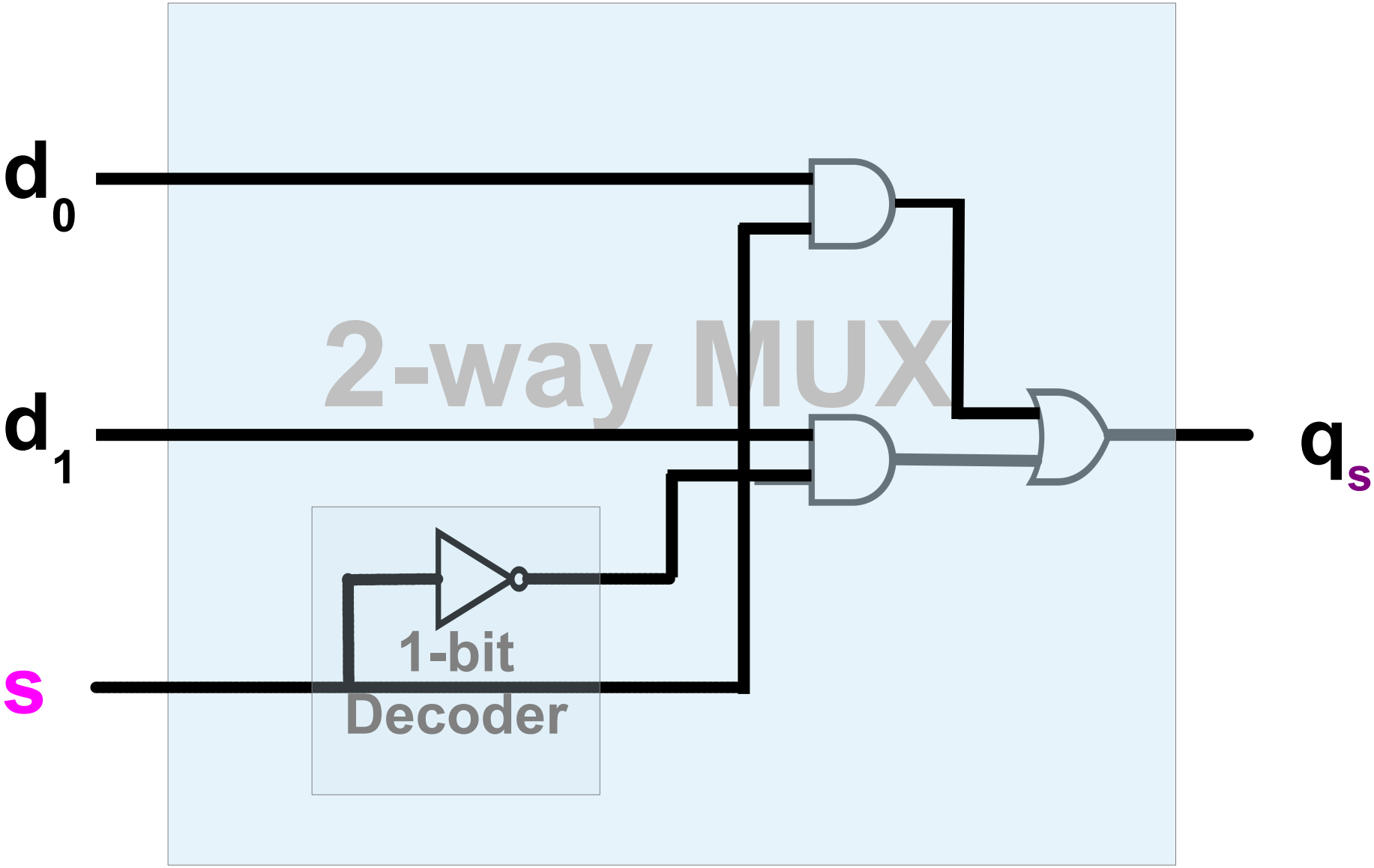
2-Way **Multiplexer**



1-Bit Decoder



2–Way **Multiplexer**

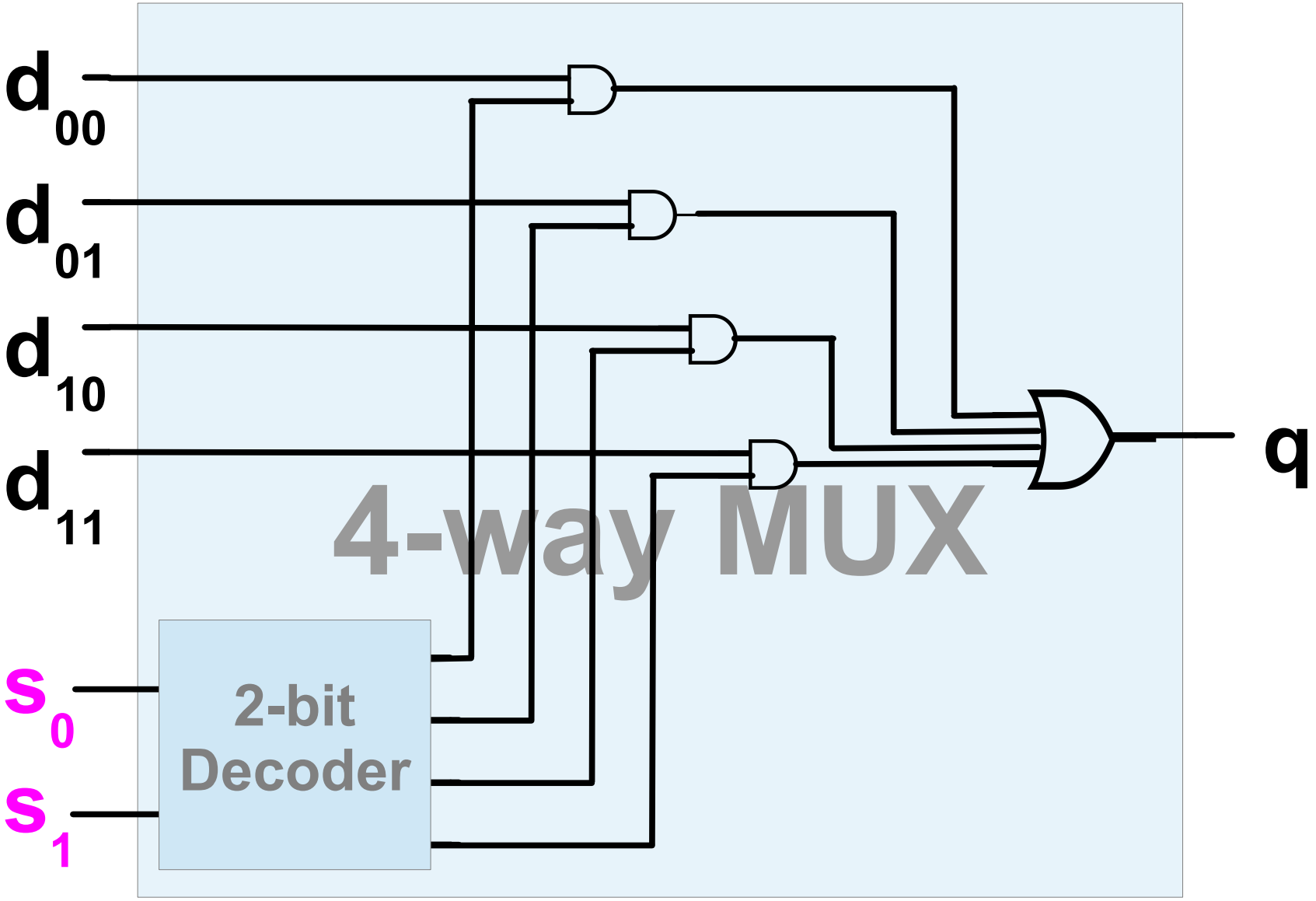


Multiplexer

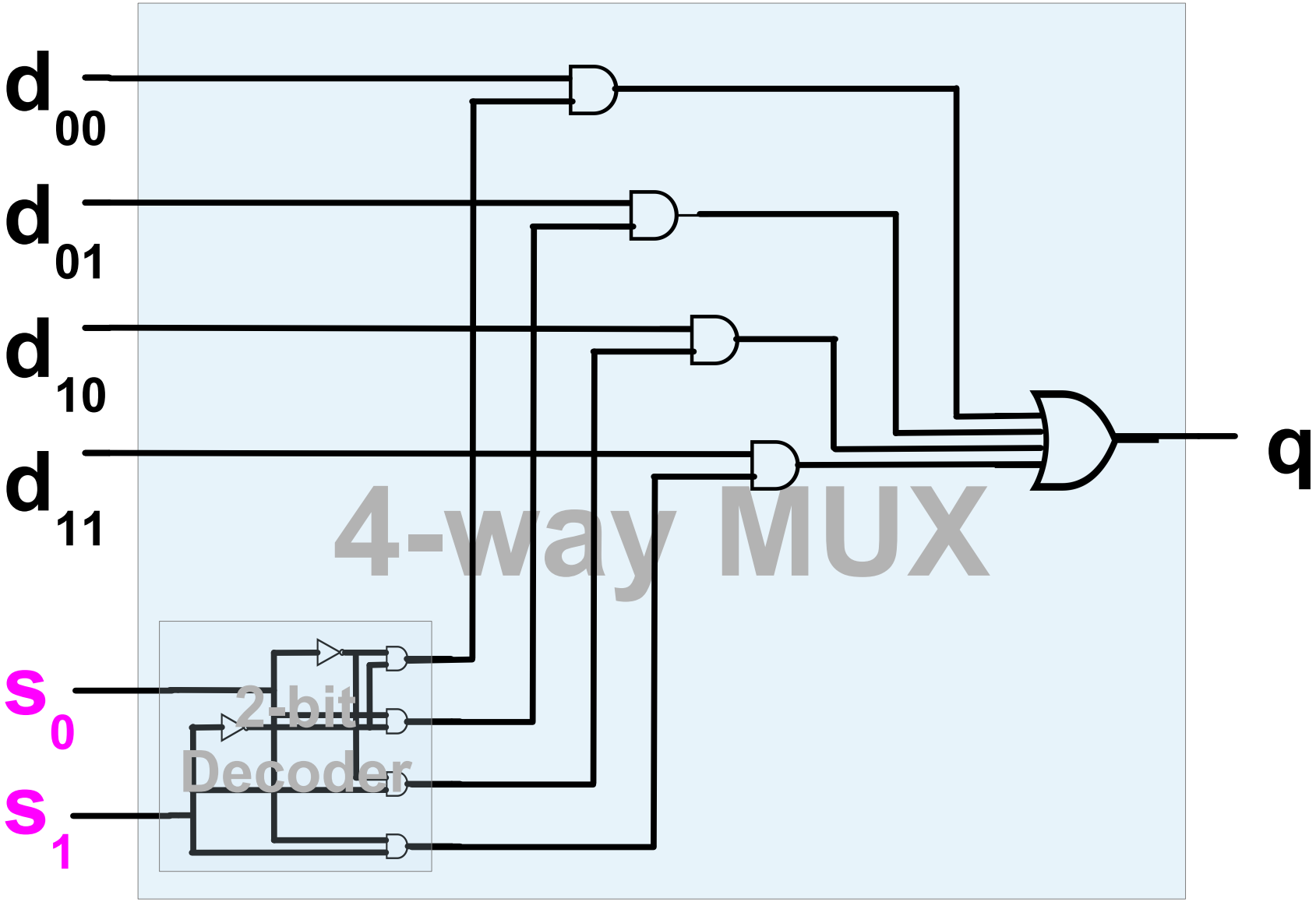
4-Way **Multiplexer**

s_1	s_0	d_{11}	d_{10}	d_{01}	d_{00}	q
0	0	√	√	√	0	0
0	0	√	√	√	1	1
0	1	√	√	0	√	0
0	1	√	√	1	√	1
1	0	√	0	√	√	0
1	0	√	1	√	√	1
1	1	0	√	√	√	0
1	1	1	√	√	√	1

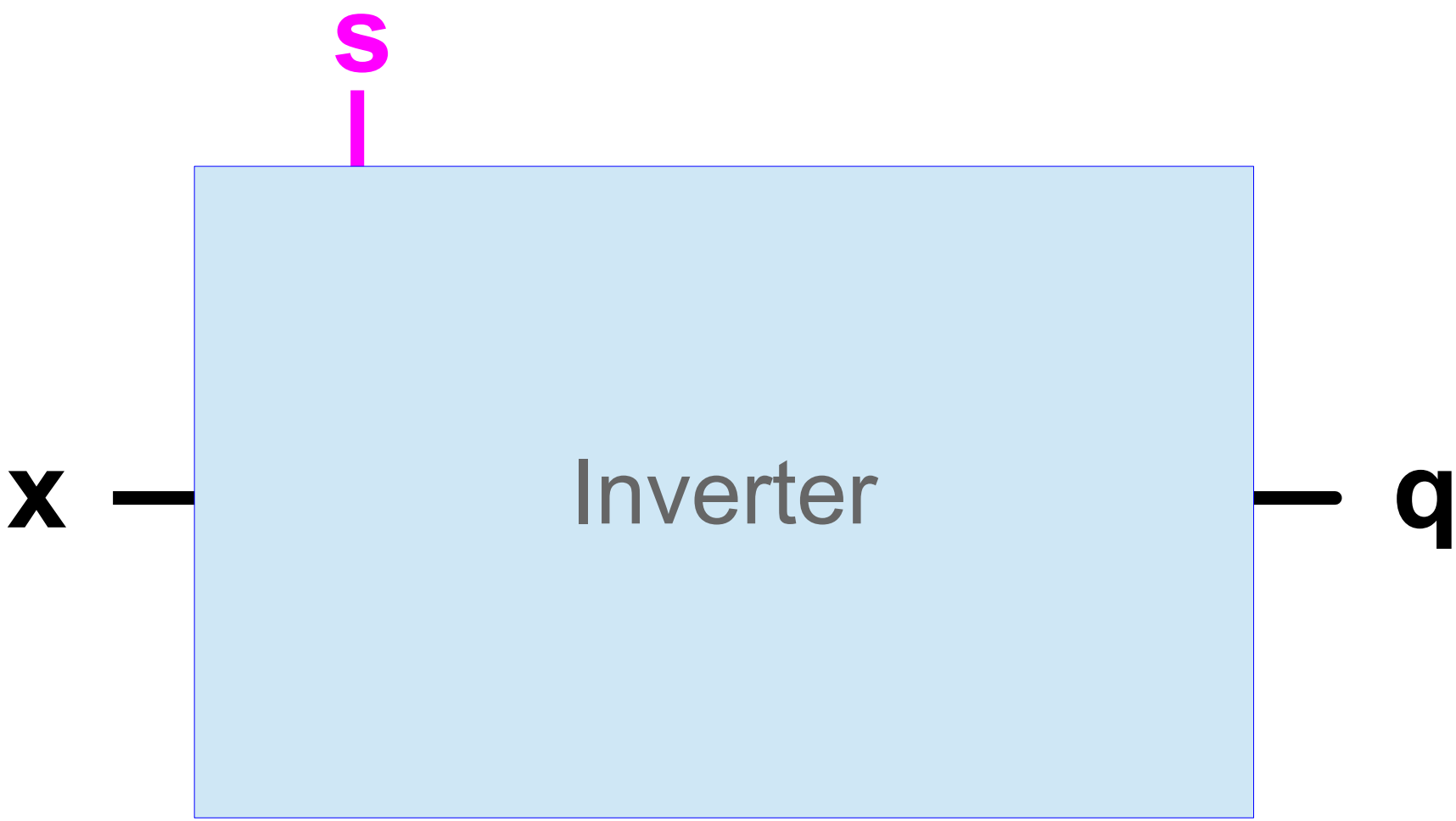
Multiplexer



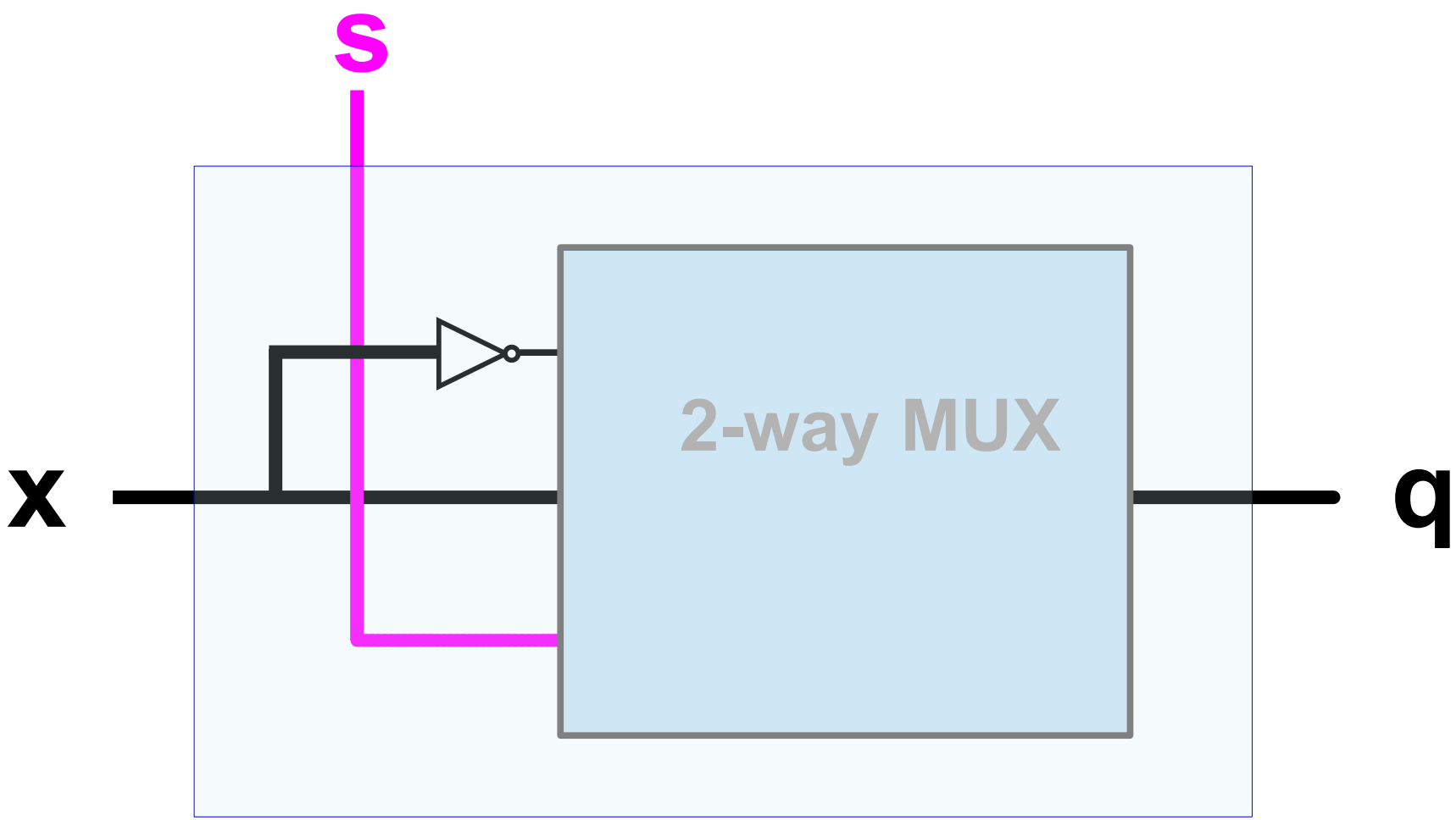
Multiplexer



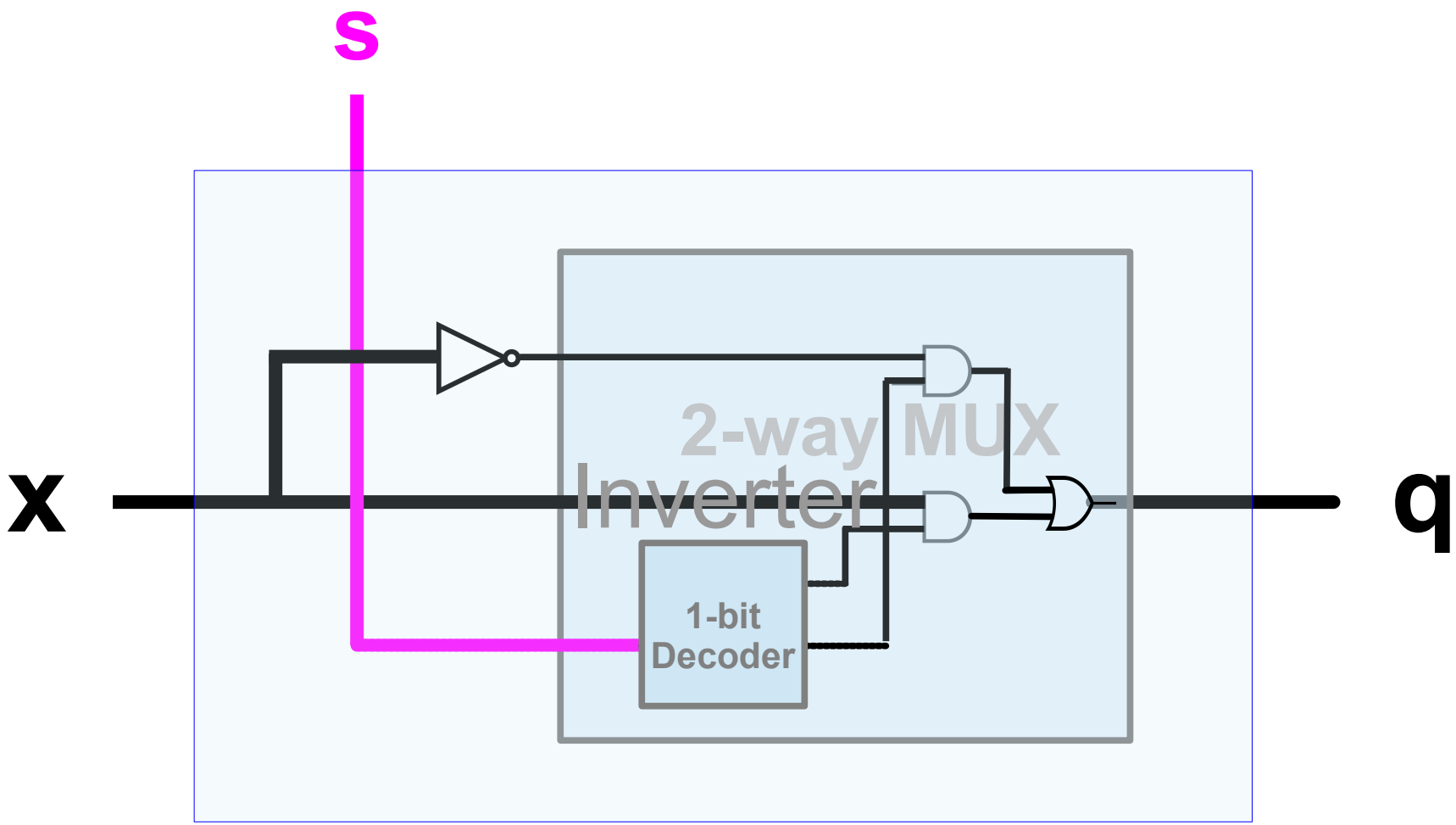
Inverter



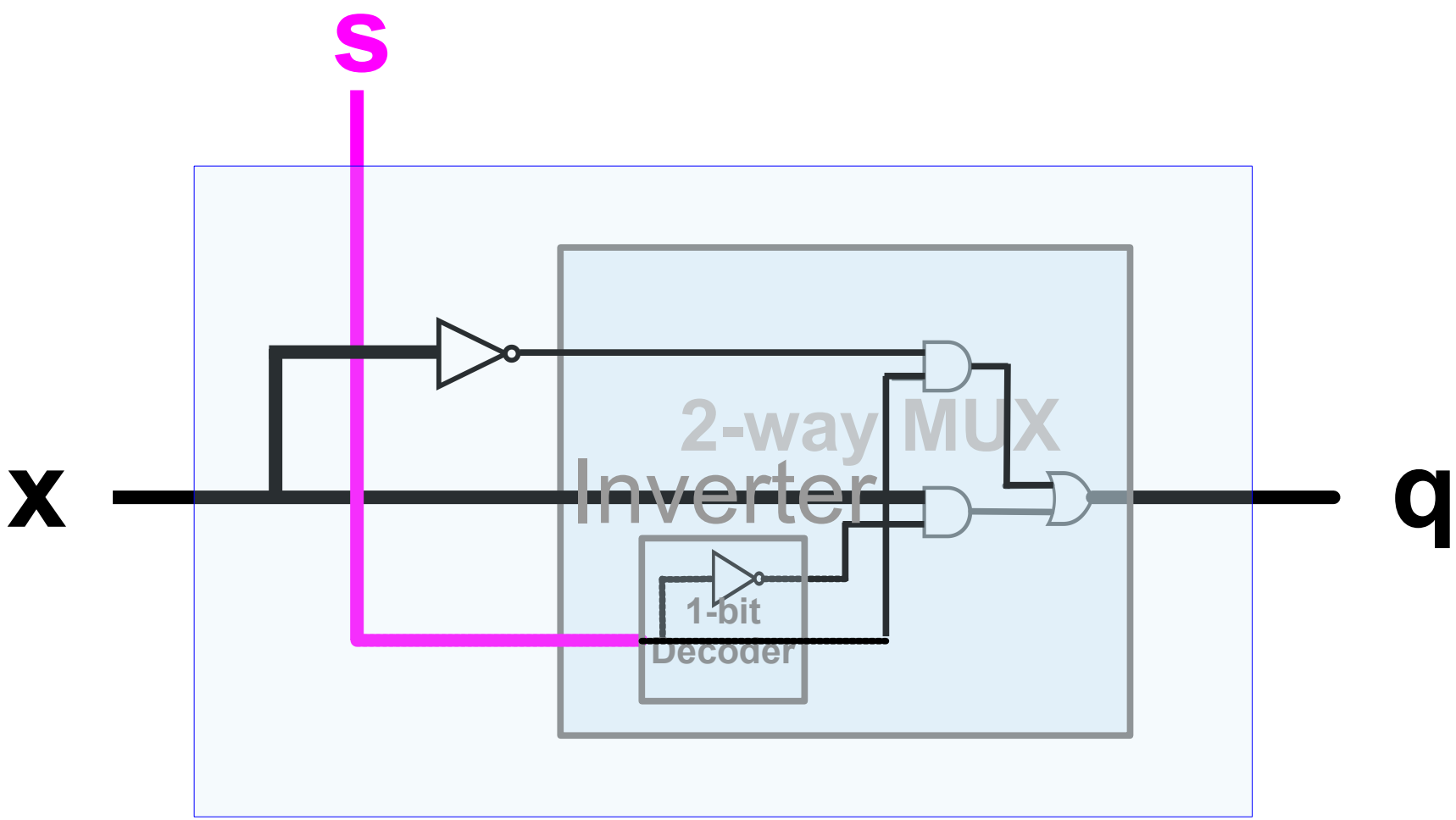
Inverter



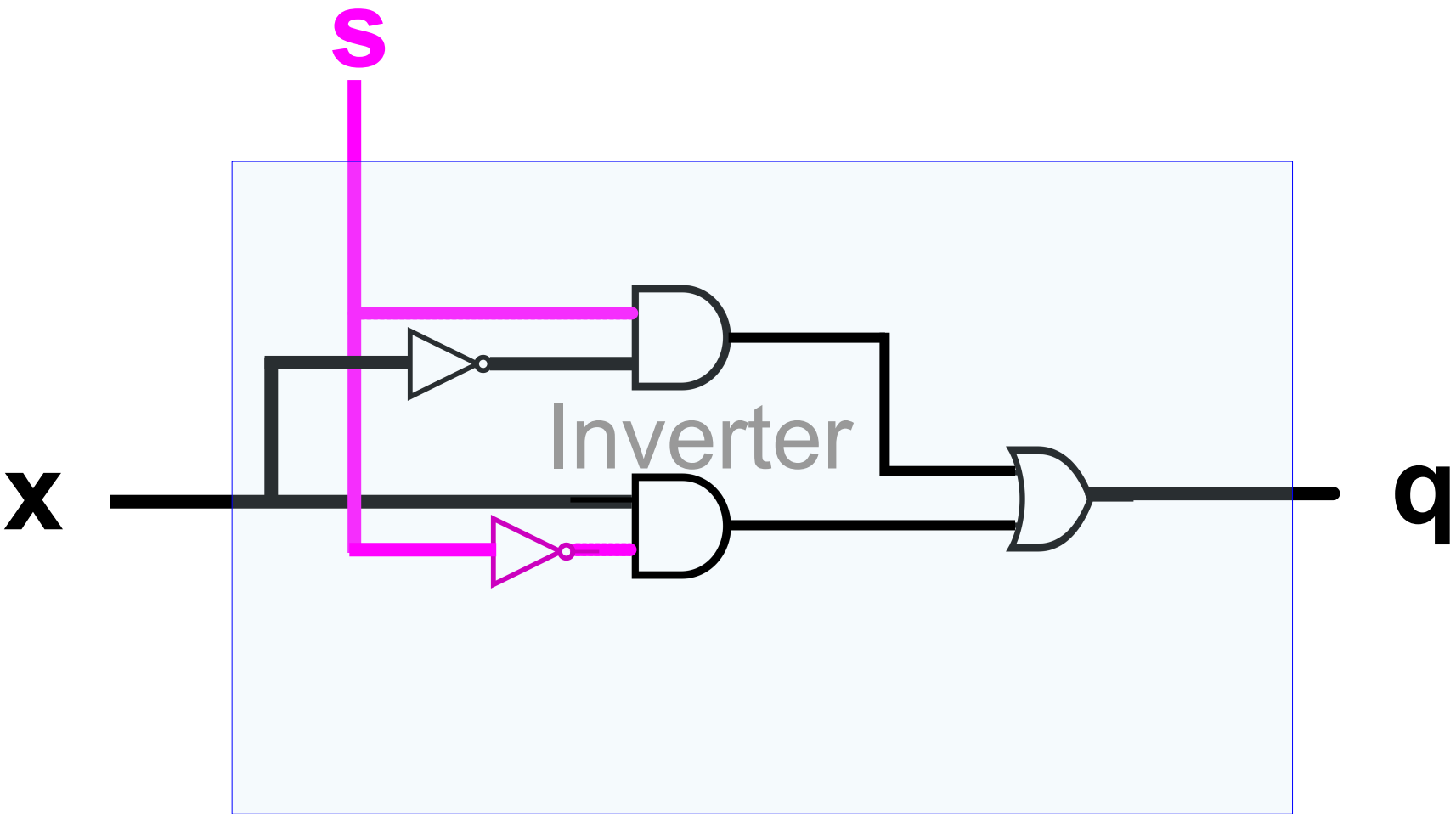
Inverter



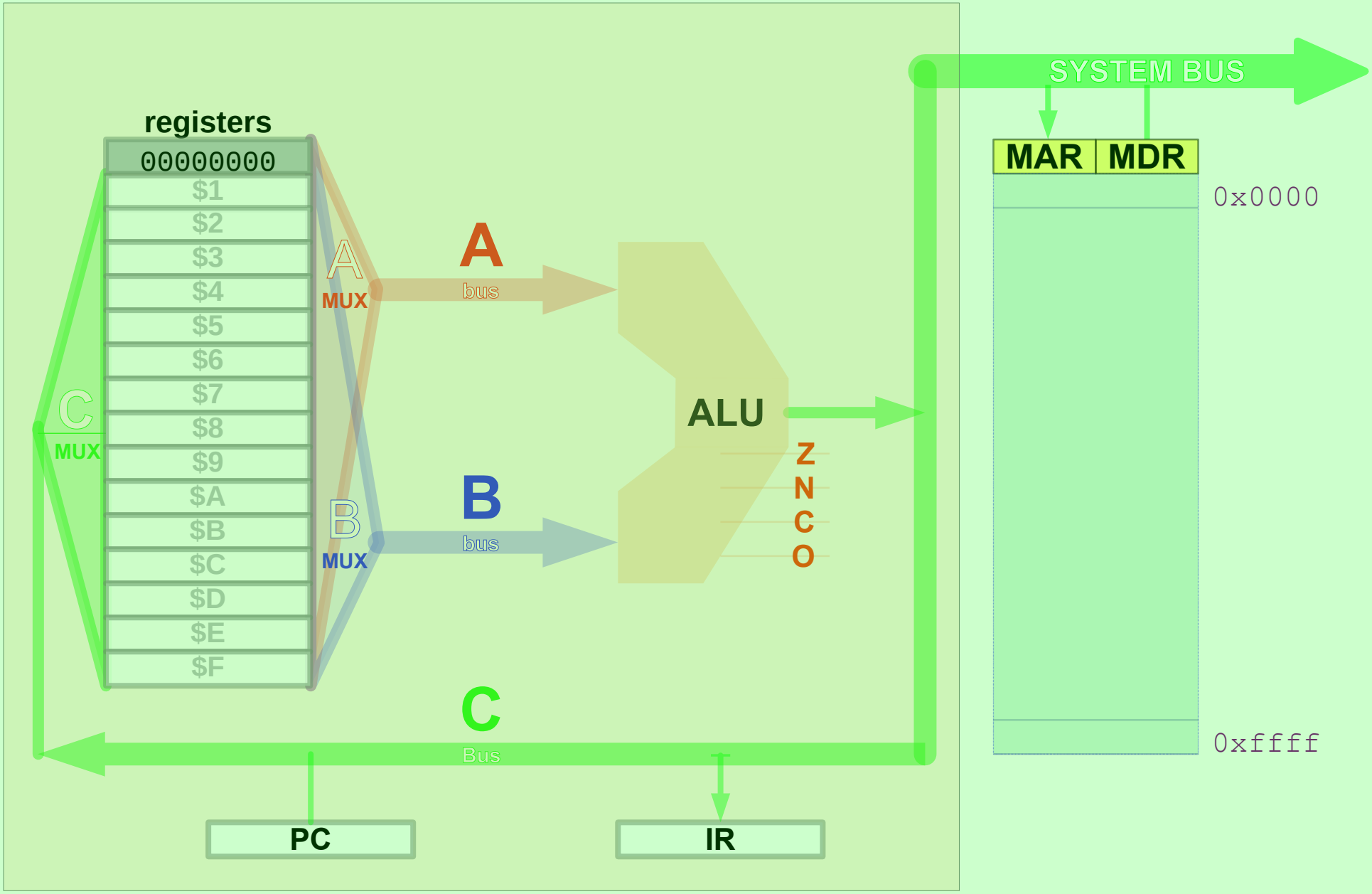
Inverter



Inverter



The TINY Computer



TINY Instruction Set Architecture

Reference Data Card

Main Memory 65536 16-bit words

M[n] – n^{th} memory address

^M[n] – content of M[n]

Register File 16 16-bit "registers"

15 real registers: $1 … $F

1 pseudo-register: $0 [$0] = 0

Immediate values

In — n-bit signed int

Un — n-bit unsigned int

CC — 4-bit condition code

Instructions [0]

ADD rT ← [rA]+[rB] [1,2]

AND rT ← [rA]&[rB] [1,3]

BRC PC ← [rA]+U4+1 ⫽ CC

BRU rL ← PC, PC ← [rA]+[rB] [1]

LDI rT ← ^M[[rA]+U4+1] [1]

LDX rT ← ^M[[rA]+[rB]] [1]

LIH $rT_{15..8}$ ← I8 [1]

NOR rT ← $\overline{[rA]|[rB]}$ [1,3]

SLL rT ← [rA]<<I4 [1,3]

SRS rT ← [rA]>>I4 [1,3]

SRU rT ← [rA]>>>I4 [1,3]

STI M[[rA]+U4+1] ← [rS]

STX M[[rA]+[rB]] ← [rS]

SUB rT ← [rA]-[rB] [1, 2]

SYS system call [4]

Arithmetic / Logical

0100	**ADD**	rT	rA	rB
0101	**SUB**	rT	rA	rB
0110	**AND**	rT	rA	rB
0111	**NOR**	rT	rA	rB

Shift / Load Immediate

1000	**LIH**	rT	I8	
1001	**SLL**	rT	rA	U4
1010	**SRS**	rT	rA	U4
1011	**SRU**	rT	rA	U4

Load/Store

0111	**LDI**	rT	rA	U4
0110	**LDX**	rT	rA	rB
0101	**STI**	rS	rA	U4
0100	**STX**	rS	rA	RB

Branch/Special

0011	**BRC**	C	rA	U4
0010	**BCU**	rL	rA	rB
0001	reserved			
0000	**SYS**	U12		

Condition Codes

0000	**true**	TT
0001	**false**	FF
0010	A = B signed	EQ
0011	A B signed	NE
0100	A < B signed	LT
0101	A B signed	GE
0110	A B signed	LE
0111	A > B signed	GT
1000	true	
1001	false	
1010	A = B unsigned	
1011	A B unsigned	
1100	A < B unsigned	LTU
1101	A B unsigned	GEU
1110	A B unsigned	LEU
1111	A > B unsigned	GTU

Notes

[0] PC ← PC+1 ***before*** instruction execution

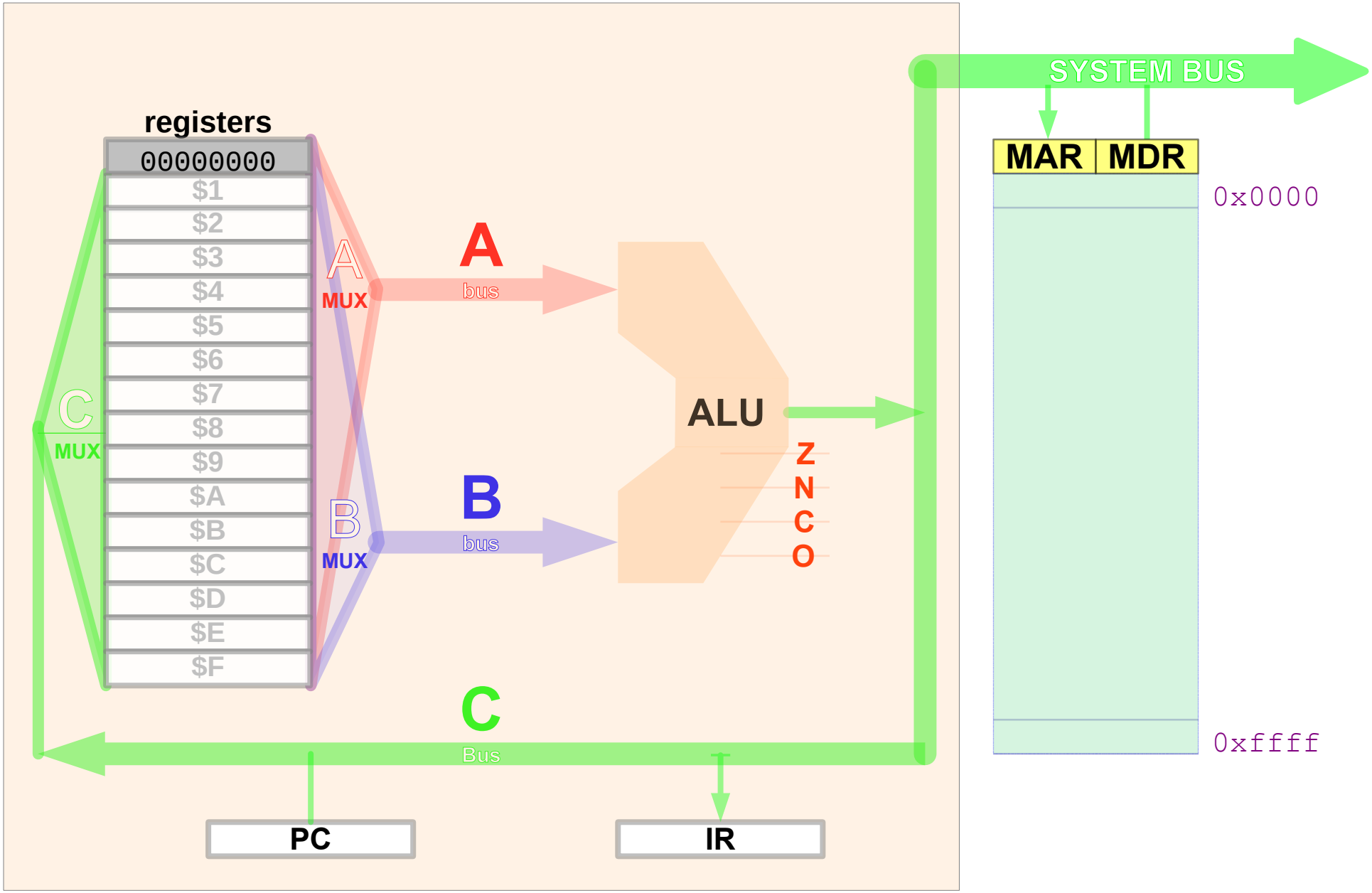
[1] $0 ***not*** changed

[2] Determines flags: **z, n, c, o**

[3] Determines flags: **z, n,**

[4] No op ⫽ U12 = 0

The TINY Computer



TINY Instruction Set Architecture

Reference Data Card

Main Memory 65536 16-bit words

M[n] – n^{th} memory address

^M[n] – content of M[n]

Register File 16 16-bit "registers"

15 real registers: $1 … $F

1 pseudo-register: $0 [$0] = 0

Immediate values

In — n-bit signed int

Un — n-bit unsigned int

CC — 4-bit condition code

Instructions [0]

ADD rT ← [rA]+[rB] [1,2]

AND rT ← [rA]&[rB] [1,3]

BRC PC ← [rA]+U4+1 ∥ CC

BRU rL ← PC, PC ← [rA]+[rB] [1]

LDI rT ← ^M[[rA]+U4+1] [1]

LDX rT ← ^M[[rA]+[rB]] [1]

LIH $rT_{15..8}$ ← I8 [1]

NOR rT ← $\overline{[rA]|[rB]}$ [1,3]

SLL rT ← [rA]<<I4 [1,3]

SRS rT ← [rA]>>I4 [1,3]

SRU rT ← [rA]>>>I4 [1,3]

STI M[[rA]+U4+1] ← [rS]

STX M[[rA]+[rB]] ← [rS]

SUB rT ← [rA]-[rB] [1, 2]

SYS system call [4]

Arithmetic / Logical

0100	**ADD**	rT	rA	rB
0101	**SUB**	rT	rA	rB
0110	**AND**	rT	rA	rB
0111	**NOR**	rT	rA	rB

Shift / Load Immediate

1000	**LIH**	rT	I8	
1001	**SLL**	rT	rA	U4
1010	**SRS**	rT	rA	U4
1011	**SRU**	rT	rA	U4

Load/Store

0111	**LDI**	rT	rA	U4
0110	**LDX**	rT	rA	rB
0101	**STI**	rS	rA	U4
0100	**STX**	rS	rA	RB

Branch/Special

0011	**BRC**	C	rA	U4
0010	**BCU**	rL	rA	rB
0001	reserved			
0000	**SYS**	U12		

Condition Codes

0000	**true**	TT
0001	**false**	FF
0010	A = B signed	EQ
0011	A B signed	NE
0100	A < B signed	LT
0101	A B signed	GE
0110	A B signed	LE
0111	A > B signed	GT
1000	true	
1001	false	
1010	A = B unsigned	
1011	A B unsigned	
1100	A < B unsigned	LTU
1101	A B unsigned	GEU
1110	A B unsigned	LEU
1111	A > B unsigned	GTU

Notes

[0] PC ← PC+1 ***before*** instruction execution

[1] $0 ***not*** changed

[2] Determines flags: **z, n, c, o**

[3] Determines flags: **z, n,**

[4] No op ∥ U12 = 0