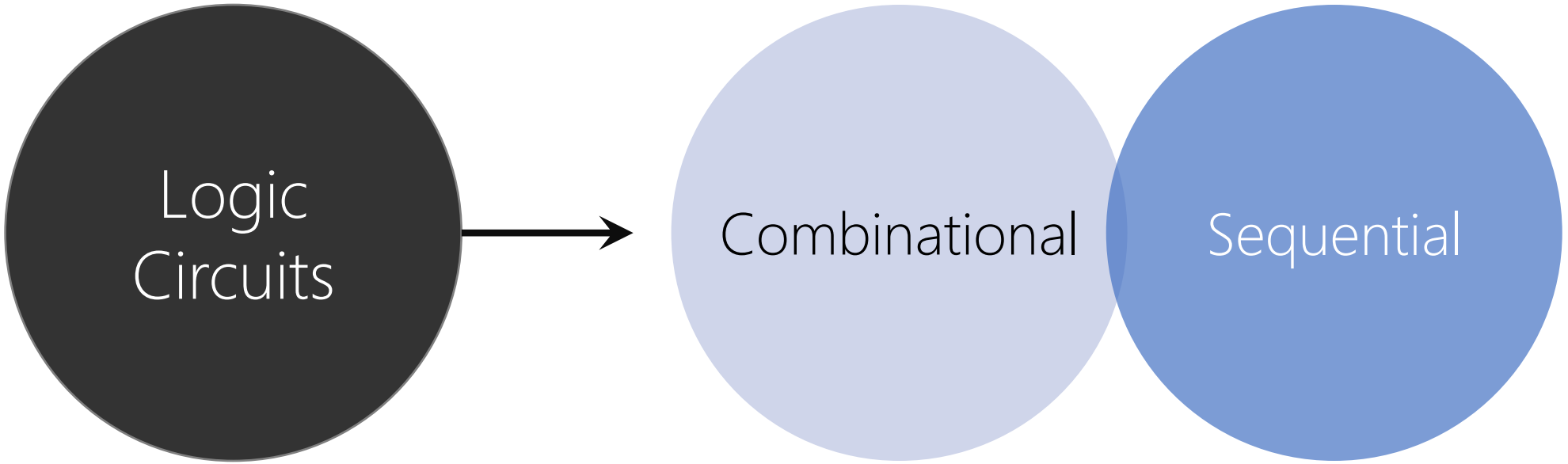

Logic
Circuits
Combinational
Sequential

# Chapter 4 Combinational Logic

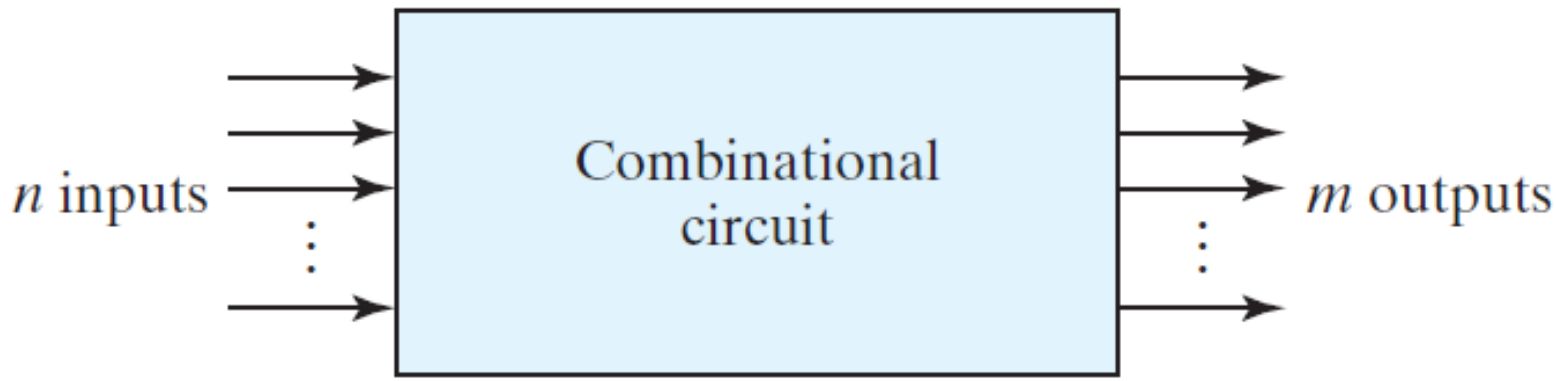
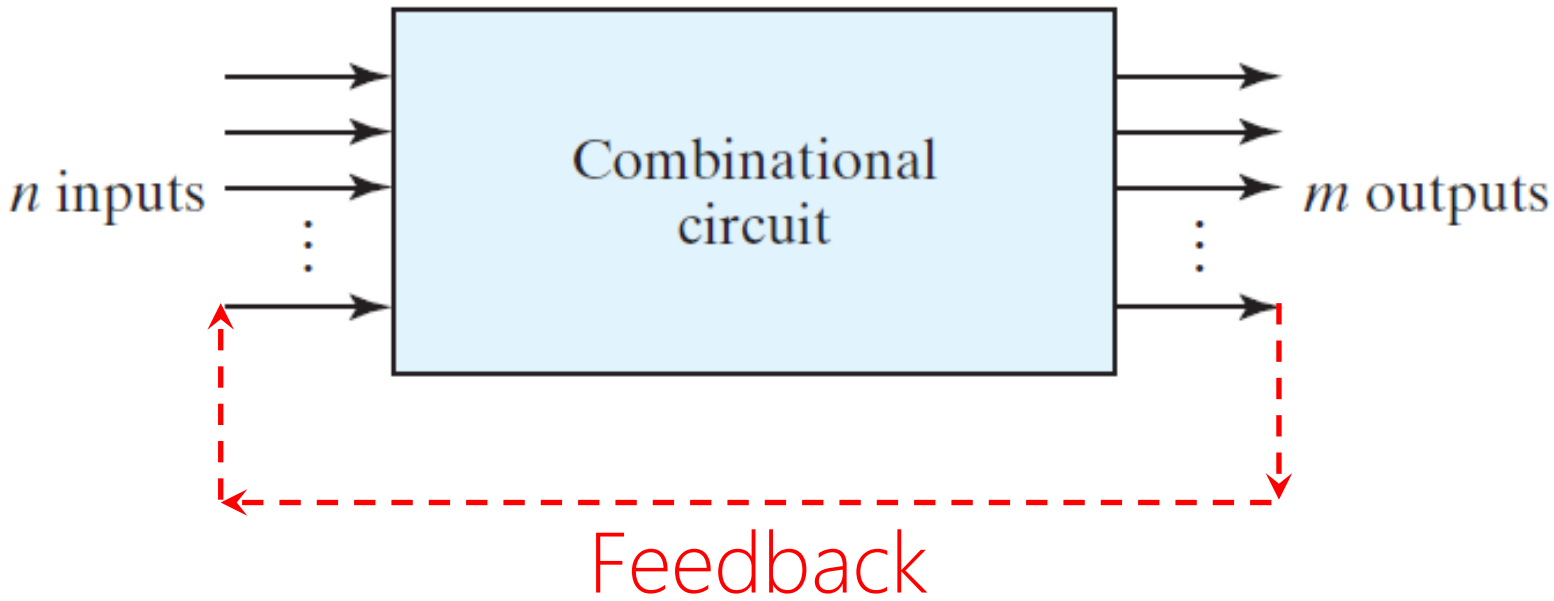



**FIGURE 4.1**
Block diagram of combinational circuit

# Sequential Logic

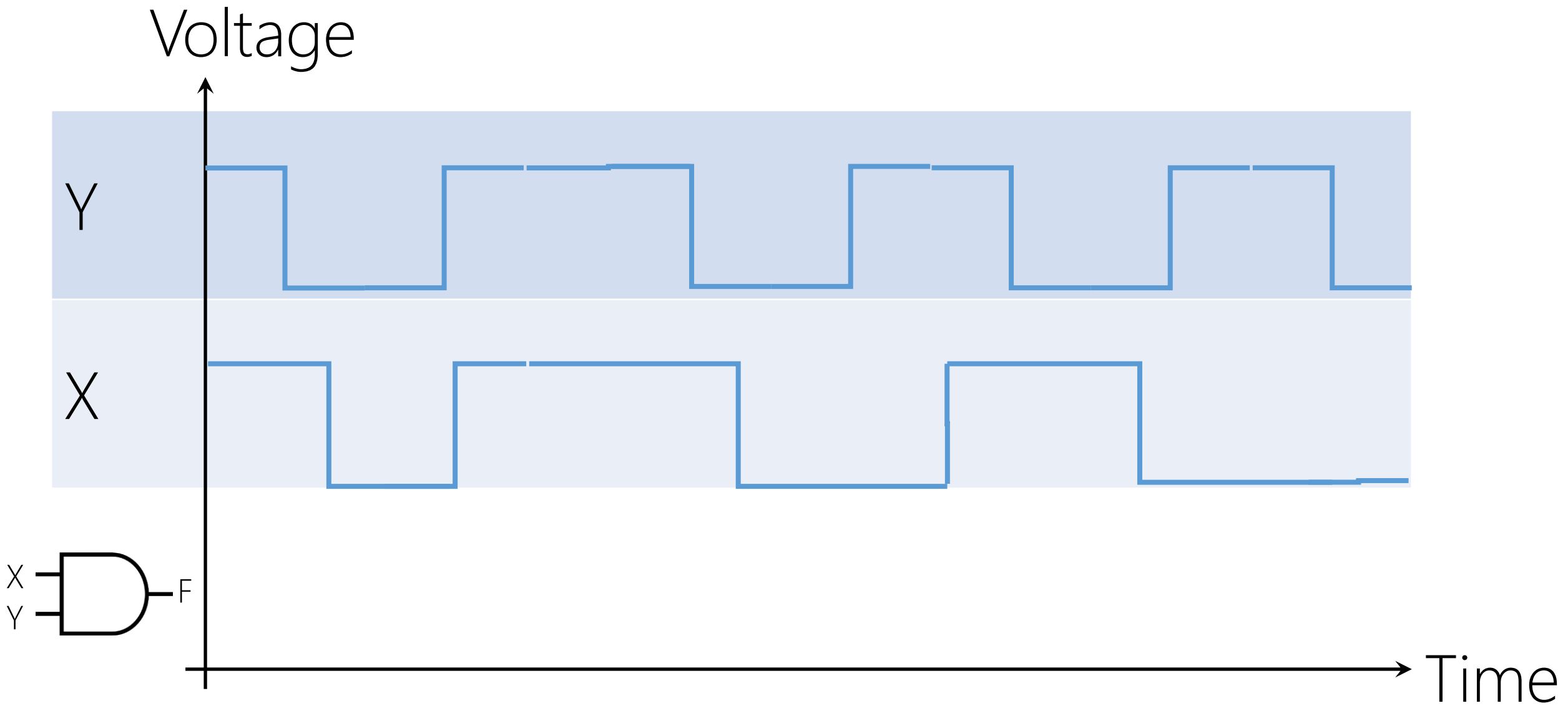


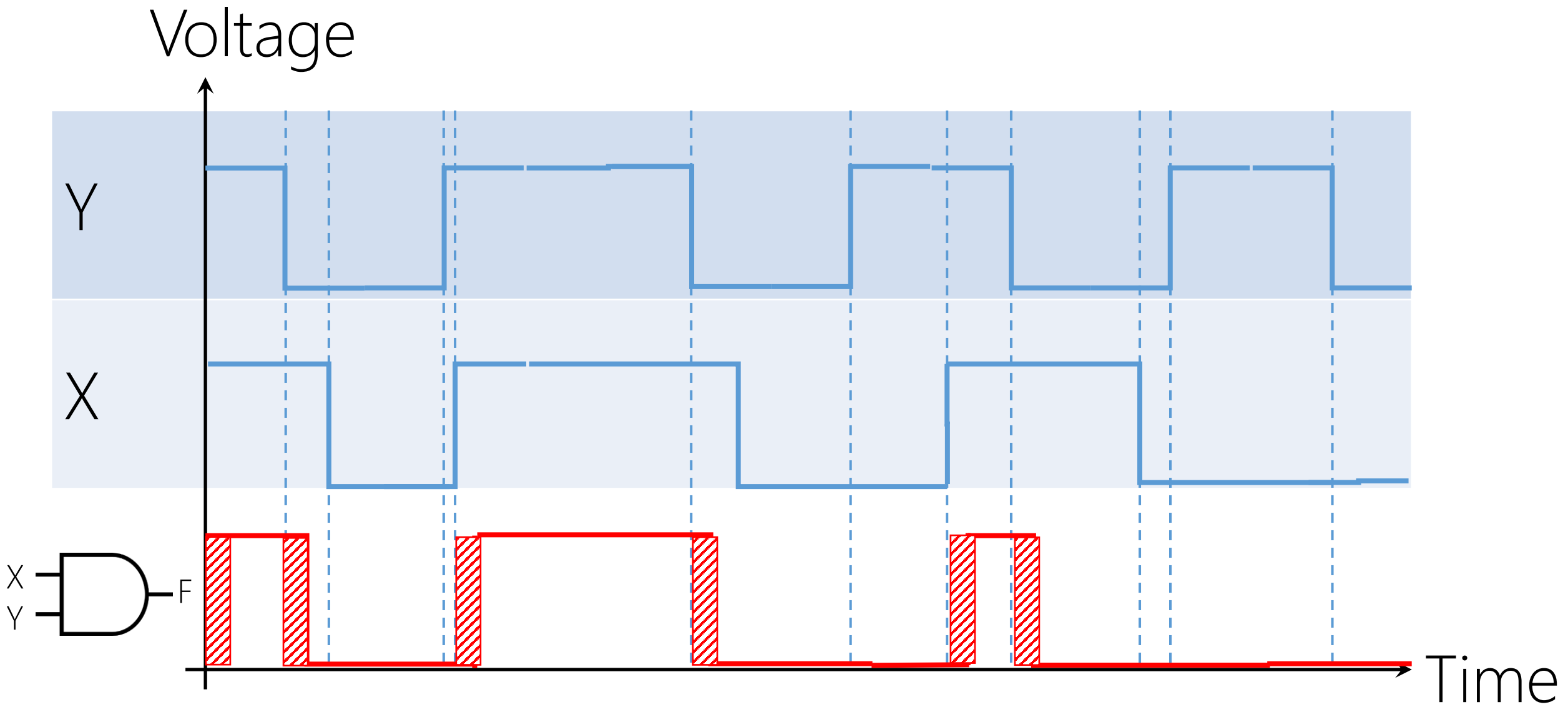
# Combinational Logic

aka. Combinational Circuit

Combination of logic gates on the present inputs → the outputs *at any time*!

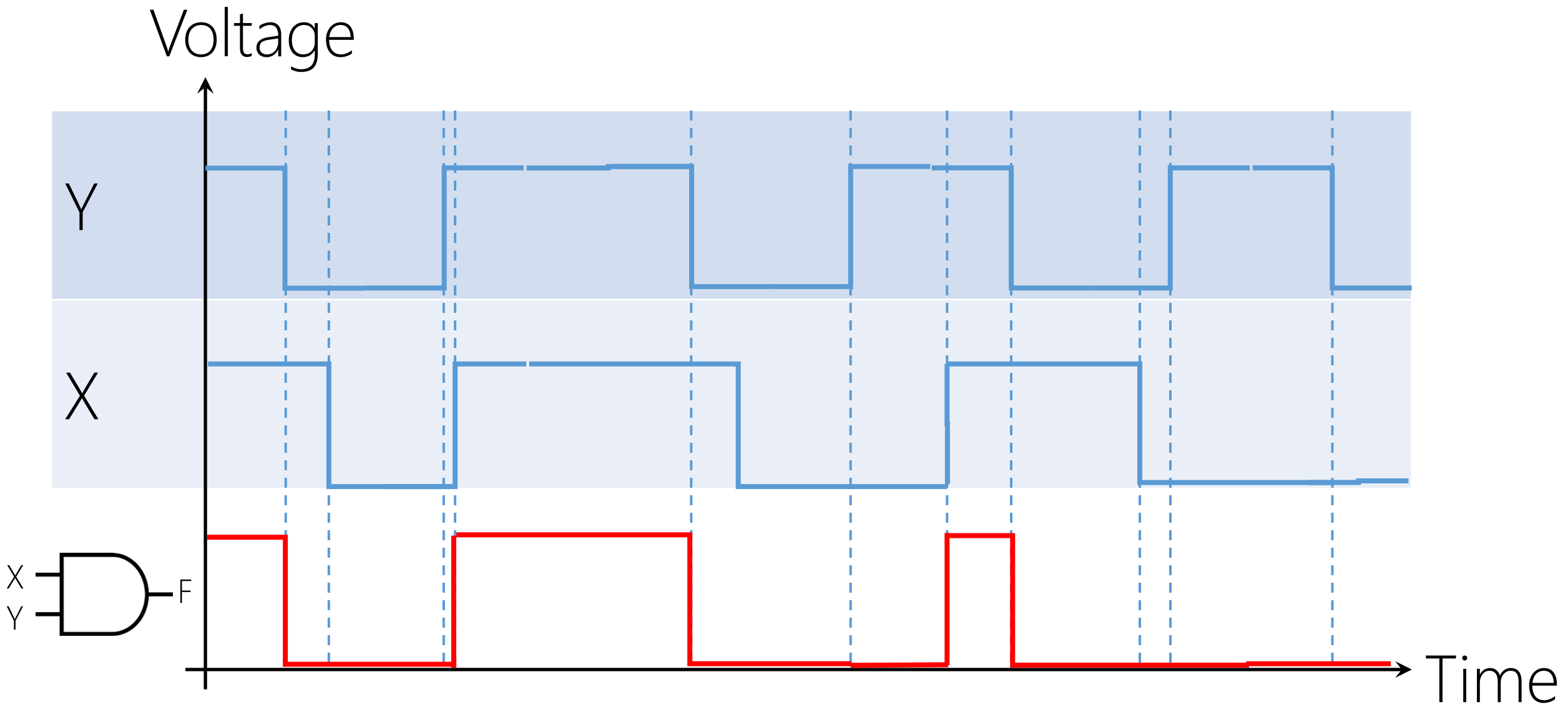
A combinational circuit performs an operation that can be specified logically by a set of Boolean functions.


Voltage
Y
X
X
Y
F
Time




# Propagation Delay (Gate Delay) ≈ Δt

https://en.wikipedia.org/wiki/Propagation_delay#Electronics




Propagation Delay (Gate Delay) ≈ Δt ≈ 0

https://en.wikipedia.org/wiki/Propagation_delay#Electronics

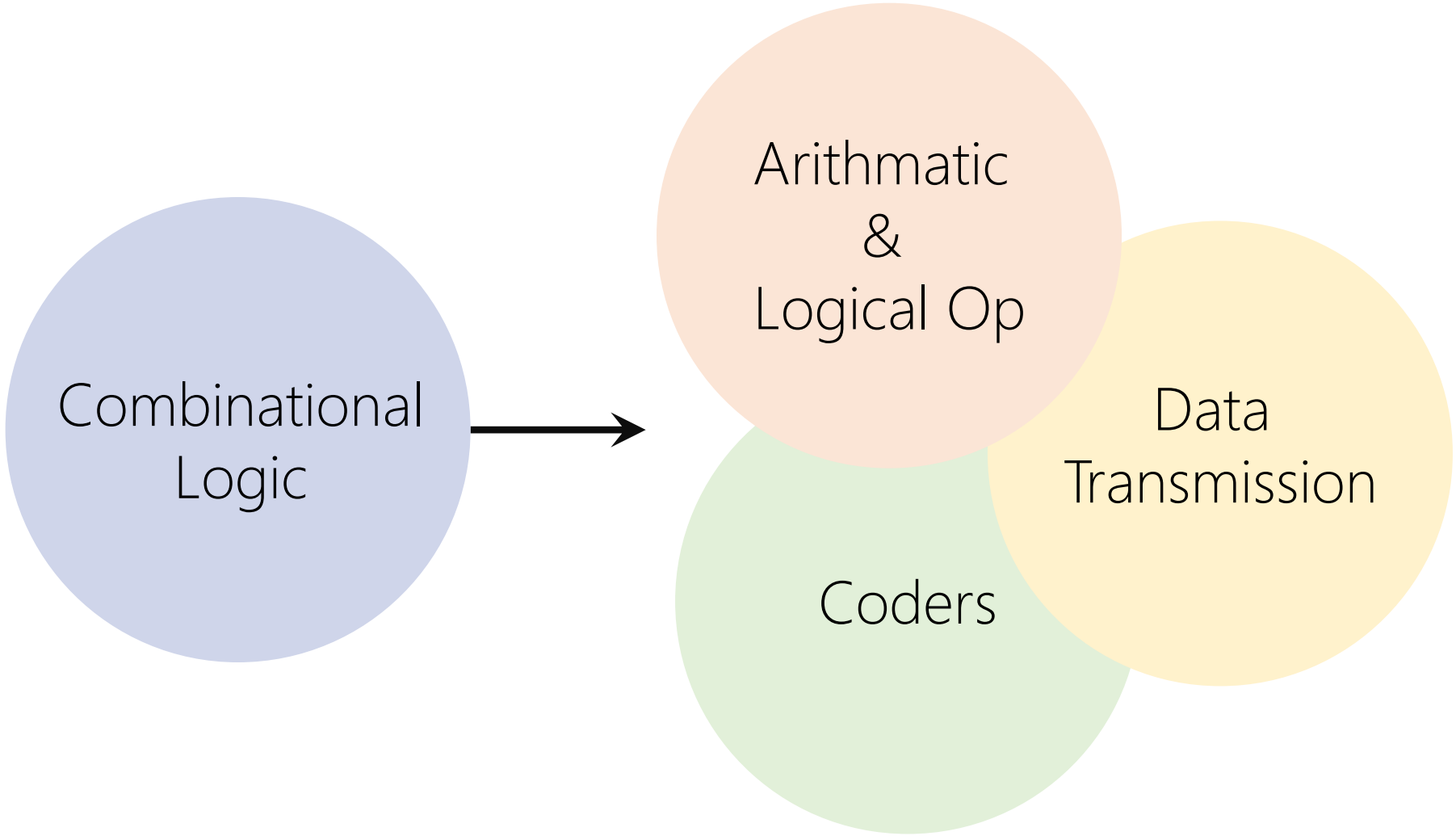
# What we've done so far

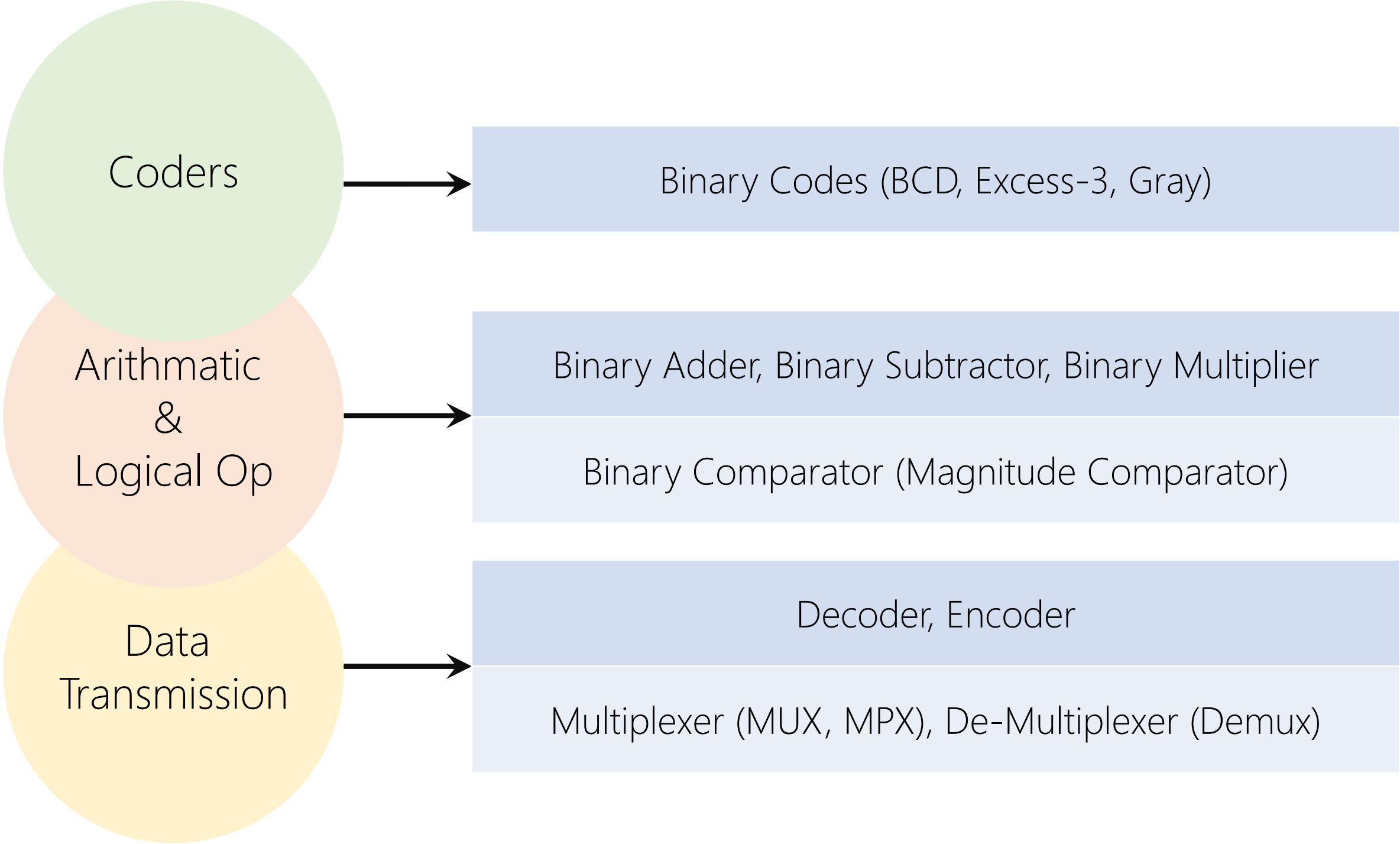
Combinational Logic aka. Combinational Circuit

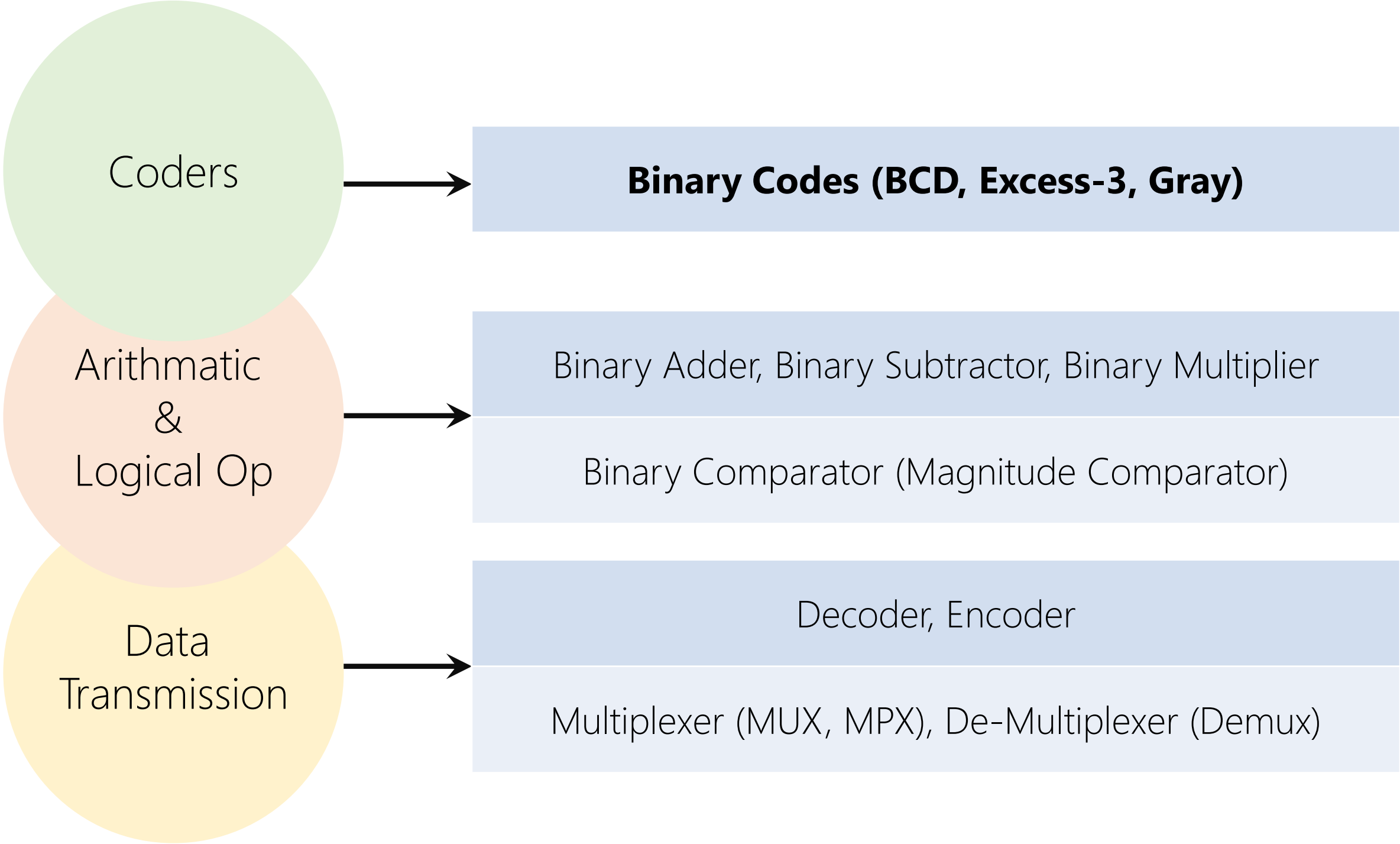
Design a combinational logic circuit:

1. Truth Table (Inputs, Outputs)
2. Output Boolean Functions (SoP: ∑m | PoS: ∏M)
3. Minimization
   - Algebraically | K-Map | Quine-McCluskey

4. Logic Diagram | Circuit


Combinational
Logic
Arithmatic
&
Logical Op
Data
Transmission
Coders


Coders
Binary Codes (BCD, Excess-3, Gray)
Arithmatic
&
Logical Op
Binary Adder, Binary Subtractor, Binary Multiplier
Binary Comparator (Magnitude Comparator)
Data
Transmission
Decoder, Encoder
Multiplexer (MUX, MPX), De-Multiplexer (Demux)


Coders
Binary Codes (BCD, Excess-3, Gray)
Arithmatic
&
Logical Op
Binary Adder, Binary Subtractor, Binary Multiplier
Binary Comparator (Magnitude Comparator)
Data
Transmission
Decoder, Encoder
Multiplexer (MUX, MPX), De-Multiplexer (Demux)

# Coding

A → Encode→B

A ← Decode ← B

# A ↔ [Enc][Dec]code ↔ B

By a convention

- Math, e.g., conversion in radix numbering system
- Non-math, e.g., in base-64, the value of characters
- Engineering
- etc

# 1-way Coding

A → Encode → B

A ↚ Decode ↚ B

# 2-way Coding

A ↔ Look up Table ↔ B

# Base-64

A ↔ Look up Table ↔ B

| Digit | Value |
|---|---|
| A | 0 |
| B | 1 |
| C | 2 |
| D | 3 |
| E | 4 |
| F | 5 |
| G | 6 |
| H | 7 |
| I | 8 |
| J | 9 |
| K | 10 |
| L | 11 |
| M | 12 |
| N | 13 |
| O | 14 |
| P | 15 |

→

| Digit | Value |
|---|---|
| Q | 16 |
| R | 17 |
| S | 18 |
| T | 19 |
| U | 20 |
| V | 21 |
| W | 22 |
| X | 23 |
| Y | 24 |
| Z | 25 |
| a | 26 |
| b | 27 |
| c | 28 |
| d | 29 |
| e | 30 |
| f | 31 |

→

| Digit | Value |
|---|---|
| g | 32 |
| h | 33 |
| i | 34 |
| j | 35 |
| k | 36 |
| l | 37 |
| m | 38 |
| n | 39 |
| o | 40 |
| p | 41 |
| q | 42 |
| r | 43 |
| s | 44 |
| t | 45 |
| u | 46 |
| v | 47 |

→

| Digit | Value |
|---|---|
| w | 48 |
| x | 49 |
| y | 50 |
| z | 51 |
| 0 | 52 |
| 1 | 53 |
| 2 | 54 |
| 3 | 55 |
| 4 | 56 |
| 5 | 57 |
| 6 | 58 |
| 7 | 59 |
| 8 | 60 |
| 9 | 61 |
| + | 62 |
| / | 63 |

# Binary Codes

## Assigning binary numbers to things

A ↔ Look up Table ↔ Binary Number

# Binary Coded Decimal
# BCD (8421)

Decimal ↔ Look up Table ↔ Binary Number

**Table 1.4**
*Binary-Coded Decimal (BCD)*

| Decimal Symbol | BCD Digit |
|---|---|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |

| Decimal | BCD (Binary Code) | Binary Number |
|---|---|---|
| 10 | 0001 0000 | 0000 1010 |
| 11 | 0001 0001 | 0000 1011 |
| 12 | 0001 0010 | 0000 1100 |
| 13 | 0001 0011 | 0000 1101 |
| 14 | 0001 0100 | 0000 1110 |
| 15 | 0001 0101 | 0000 1111 |
| 16 | 0001 0110 | 0001 0000 |
| 17 | 0001 0111 | 0001 0001 |
| 18 | 0001 1000 | 0001 0010 |
| 19 | 0001 1001 | 0001 0011 |
| 20 | 0010 0000 | 0001 0100 |
| 21 | 0010 0001 | 0001 0101 |
| 22 | 0010 0010 | 0001 0110 |
| 23 | 0010 0011 | 0001 0111 |
| … | … | … |

| Decimal | BCD (Binary Code) | Binary Number |
|---|---|---|
| 10 | 0001 0000 | 0000 1010 |
| 11 | 0001 0001 | 0000 1011 |
| 12 | 0001 0010 | 0000 1100 |
| 13 | 0001 0011 | 0000 1101 |
| 14 | 0001 0100 | 0000 1110 |
| 15 | 0001 0101 | 0000 1111 |
| 16 | 0001 0110 | 0001 0000 |
| 17 | 0001 0111 | 0001 0001 |
| 18 | 0001 1000 | 0001 0010 |
| 19 | 0001 1001 | 0001 0011 |
| 20 | 0010 0000 | 0001 0100 |
| 21 | 0010 0001 | 0001 0101 |
| 22 | 0010 0010 | 0001 0110 |
| 23 | 0010 0011 | 0001 0111 |
| … | … | … |

$$(185)_{10} = (?)_{BCD} = (?)_{2}$$

$$(185)_{10} = (0001)_{BCD} = (?)_2$$

$$(185)_{10} = (0001\ 1000)_{BCD} = (?)_2$$

$$(185)_{10} = (0001\ 1000\ 0101)_{BCD} = (?)_2$$

$$(185)_{10} = (0001\ 1000\ 0101)_{BCD} = (?)_2$$

$$(185)_{10} = (0001\ 1000\ 0101)_{BCD} = (?)_2$$

| | Remainder |
|---|---|
| 185÷2 | 1 |
| 92÷2 | 0 |
| 46÷2 | 0 |
| 23÷2 | 1 |
| 11÷2 | 1 |
| 5÷2 | 1 |
| 2÷2 | 0 |
| 1÷2 | 1 |
| 0 | |

$$(185)_{10} = (0001\ 1000\ 0101)_{BCD} = (10111001)_2$$

# Other Binary Codes

A ↔ Look up Table ↔ B

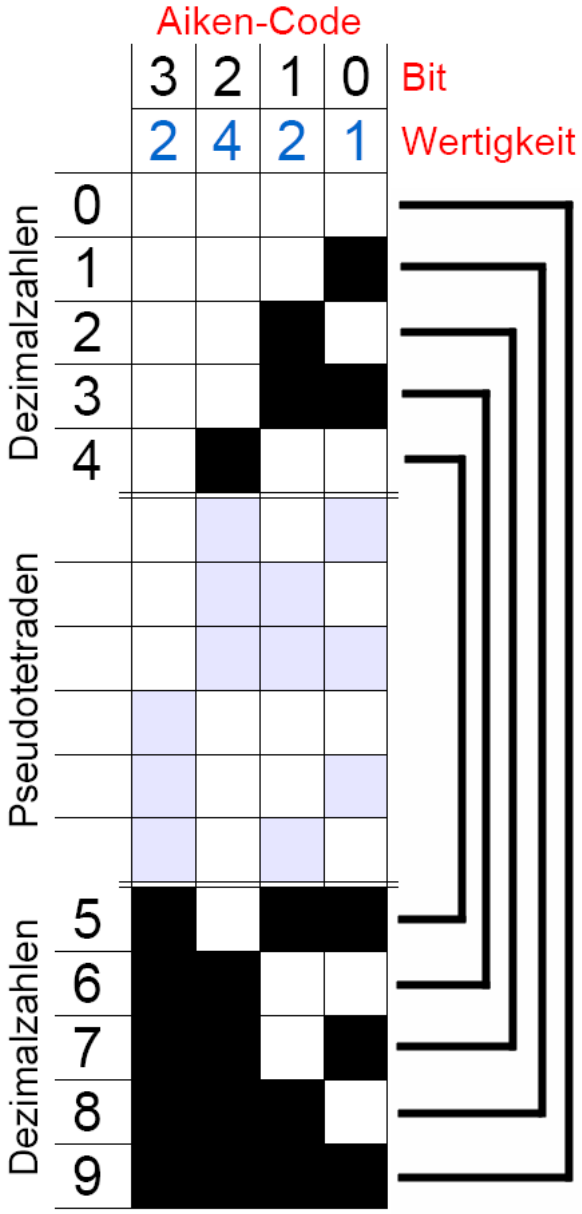
**Table 1.5**
*Four Different Binary Codes for the Decimal Digits*

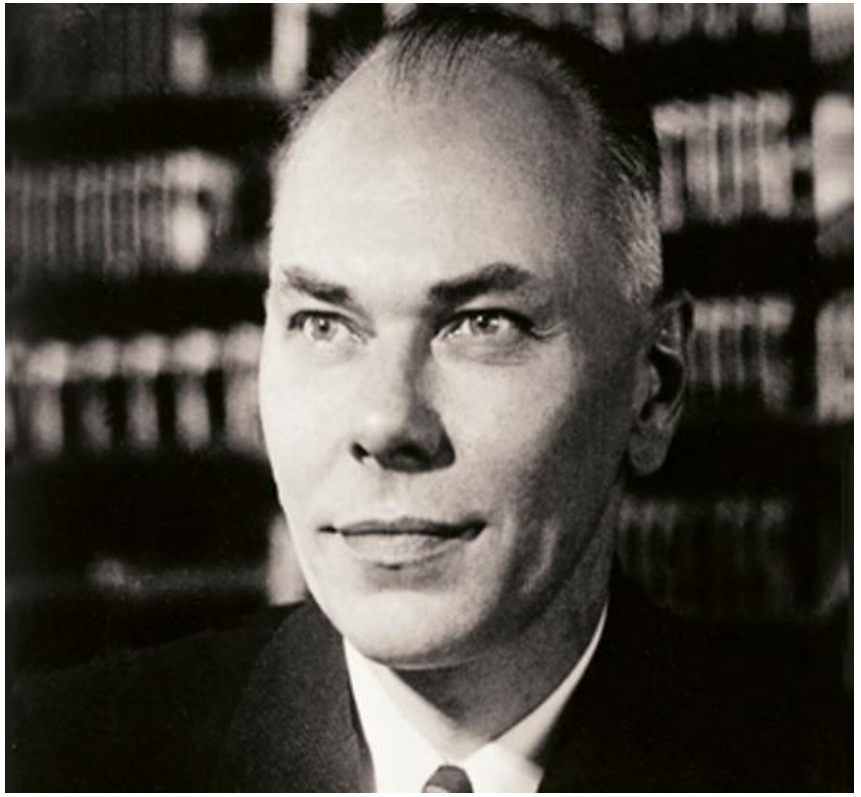
| Decimal Digit | BCD 8421 | Aiken 2421 | Excess-3 | 8, 4, −2, −1 |
|---|---|---|---|---|
| 0 | 0000 | 0000 | 0011 | 0000 |
| 1 | 0001 | 0001 | 0100 | 0111 |
| 2 | 0010 | 0010 | 0101 | 0110 |
| 3 | 0011 | 0011 | 0110 | 0101 |
| 4 | 0100 | 0100 | 0111 | 0100 |
| 5 | 0101 | 1011 | 1000 | 1011 |
| 6 | 0110 | 1100 | 1001 | 1010 |
| 7 | 0111 | 1101 | 1010 | 1001 |
| 8 | 1000 | 1110 | 1011 | 1000 |
| 9 | 1001 | 1111 | 1100 | 1111 |

# Other Binary Codes
# Aiken (2421)

/ˈeɪkən/

https://en.wikipedia.org/wiki/Aiken_code



# Howard Hathaway Aiken

(March 8, 1900 – March 14, 1973)

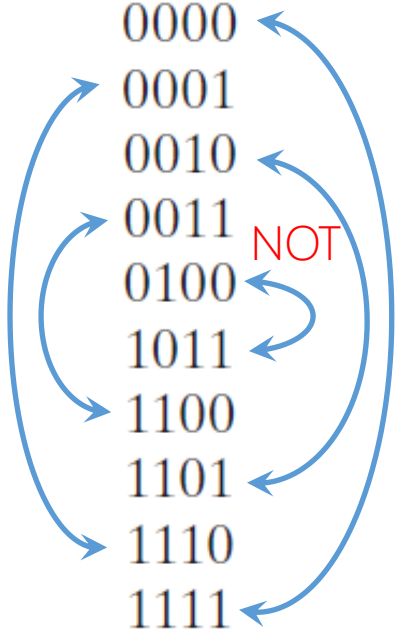
Physicist

Pioneer in computing

Original conceptual designer behind IBM's Harvard Mark I

**Table 1.5**
*Four Different Binary Codes for the Decimal Digits*

| Decimal Digit | BCD 8421 | Aiken 2421 | Excess-3 | 8, 4, −2, −1 |
|---|---|---|---|---|
| 0 | 0000 | 0000 | 0011 | 0000 |
| 1 | 0001 | 0001 | 0100 | 0111 |
| 2 | 0010 | 0010 | 0101 | 0110 |
| 3 | 0011 | 0011 | 0110 | 0101 |
| 4 | 0100 | 0100 | 0111 | 0100 |
| 5 | 0101 | 1011 | 1000 | 1011 |
| 6 | 0110 | 1100 | 1001 | 1010 |
| 7 | 0111 | 1101 | 1010 | 1001 |
| 8 | 1000 | 1110 | 1011 | 1000 |
| 9 | 1001 | 1111 | 1100 | 1111 |



$$(185)_{10} = (0001\ 1000\ 0101)_{\text{BCD (8421)}}$$
$$= (10111001)_2$$
$$= (0001\ \text{NOT(1)}\ \text{NOT(4)})_{\text{Aiken (2421)}}$$

$$(185)_{10} = (0001\ 1000\ 0101)_{\text{BCD (8421)}}$$
$$= (10111001)_2$$
$$= (0001\ \text{NOT}(0001)\ \text{NOT}(0100))_{\text{Aiken (2421)}}$$

$$
\begin{aligned}
(185)_{10} &= (0001\ 1000\ 0101)_{\text{BCD (8421)}} \\
&= (10111001)_{2} \\
&= (0001\ 1110\ 1011)_{\text{Aiken (2421)}}
\end{aligned}
$$

# Other Binary Codes
# Excess-3 (XS-3)

https://en.wikipedia.org/wiki/Excess-3

# George Robert Stibitz

(April 30, 1904 – January 31, 1995)

Bell Labs researcher

One of the fathers of the modern first digital computer

**Table 1.5**

*Four Different Binary Codes for the Decimal Digits*

| Decimal Digit | BCD 8421 | Aiken 2421 | +3 Excess-3 | 8, 4, −2, −1 |
|---|---|---|---|---|
| 0 | 0000 | 0000 | 0011 | 0000 |
| 1 | 0001 | 0001 | 0100 | 0111 |
| 2 | 0010 | 0010 | 0101 | 0110 |
| 3 | 0011 | 0011 | 0110 | 0101 |
| 4 | 0100 | 0100 | 0111 | 0100 |
| 5 | 0101 | 1011 | 1000 | 1011 |
| 6 | 0110 | 1100 | 1001 | 1010 |
| 7 | 0111 | 1101 | 1010 | 1001 |
| 8 | 1000 | 1110 | 1011 | 1000 |
| 9 | 1001 | 1111 | 1100 | 1111 |

$$
\begin{aligned}
(185)_{10} &= (0001\ 1000\ 0101)_{\text{BCD (8421)}} \\
&= (10111001)_{2} \\
&= (0001\ 1110\ 1011)_{\text{Aiken (2421)}} \\
&= ((1+3)\ (8+3)\ (5+3))_{\text{Excess-3}}
\end{aligned}
$$

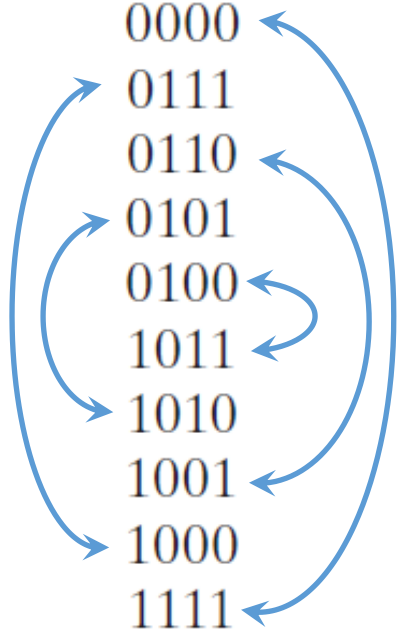
$$
\begin{aligned}
(185)_{10} &= (0001\ 1000\ 0101)_{\text{BCD (8421)}} \\
&= (10111001)_{2} \\
&= (0001\ 1110\ 1011)_{\text{Aiken (2421)}} \\
&= ((4)\ (11)\ (8))_{\text{Excess-3}}
\end{aligned}
$$

$$
\begin{aligned}
(185)_{10} &= (0001\ 1000\ 0101)_{\text{BCD (8421)}} \\
&= (10111001)_{2} \\
&= (0001\ 1110\ 1011)_{\text{Aiken (2421)}} \\
&= (0100\ 1011\ 1000)_{\text{Excess-3}}
\end{aligned}
$$

# Other Binary Codes
# 84(-2)(-1)

**Table 1.5**
*Four Different Binary Codes for the Decimal Digits*

| Decimal Digit | BCD 8421 | Aiken 2421 | Excess-3 | 8, 4, −2, −1 |
|---|---|---|---|---|
| 0 | 0000 | 0000 | 0011 | 0000 |
| 1 | 0001 | 0001 | 0100 | 0111 |
| 2 | 0010 | 0010 | 0101 | 0110 |
| 3 | 0011 | 0011 | 0110 | 0101 |
| 4 | 0100 | 0100 | 0111 | 0100 |
| 5 | 0101 | 1011 | 1000 | 1011 |
| 6 | 0110 | 1100 | 1001 | 1010 |
| 7 | 0111 | 1101 | 1010 | 1001 |
| 8 | 1000 | 1110 | 1011 | 1000 |
| 9 | 1001 | 1111 | 1100 | 1111 |



NOT

$$
\begin{aligned}
(185)_{10} &= (0001\ 1000\ 0101)_{\text{BCD (8421)}} \\
&= (10111001)_{2} \\
&= (0001\ 1110\ 1011)_{\text{Aiken (2421)}} \\
&= (0100\ 1011\ 1000)_{\text{Excess-3}} \\
&= (0111\ 1000\ 1011)_{\text{84-2-1}}
\end{aligned}
$$

# What's nice about *some* binary codes?

# Self-complementing

The 9's complement of the decimal number

=

The 1's complement (NOT) of its binary code

$$(185)_{10} = (0001\ 1110\ 1011)_{\text{Aiken (2421)}}$$
$$= (0100\ 1011\ 1000)_{\text{Excess-3}}$$
$$= (0111\ 1000\ 1011)_{\text{84-2-1}}$$

$$9\text{'s-comp}(185)_{10} = (814)_{10}$$
$$= \text{NOT}(0001\ 1110\ 1011)_{\text{Aiken (2421)}}$$
$$= \text{NOT}(0100\ 1011\ 1000)_{\text{Excess-3}}$$
$$= \text{NOT}(0111\ 1000\ 1011)_{\text{84-2-1}}$$

$$(185)_{10} = (0001\ 1110\ 1011)_{\text{Aiken (2421)}}$$
$$= (0100\ 1011\ 1000)_{\text{Excess-3}}$$
$$= (0111\ 1000\ 1011)_{\text{84-2-1}}$$

$$9\text{'s-comp}(185)_{10} = (814)_{10}$$
$$= (1110\ 0001\ 0100)_{\text{Aiken (2421)}}$$
$$= (1011\ 0100\ 0111)_{\text{Excess-3}}$$
$$= (1000\ 0111\ 0100)_{\text{84-2-1}}$$

# Other Binary Codes
# Gray

**Table 1.6**
*Gray Code*

| Gray Code | Decimal Equivalent |
|---|---|
| 0000 | 0 |
| 0001 | 1 |
| 0011 | 2 |
| 0010 | 3 |
| 0110 | 4 |
| 0111 | 5 |
| 0101 | 6 |
| 0100 | 7 |
| 1100 | 8 |
| 1101 | 9 |
| 1111 | 10 |
| 1110 | 11 |
| 1010 | 12 |
| 1011 | 13 |
| 1001 | 14 |
| 1000 | 15 |

# Gray Code
## Analog → Digital

**Table 1.6**
*Gray Code*

| Gray Code | Decimal Equivalent |
|---|---|
| 0000 | 0 |
| 0001 | 1 |
| 0011 | 2 |
| 0010 | 3 |
| 0110 | 4 |
| 0111 | 5 |
| 0101 | 6 |
| 0100 | 7 |
| 1100 | 8 |
| 1101 | 9 |
| 1111 | 10 |
| 1110 | 11 |
| 1010 | 12 |
| 1011 | 13 |
| 1001 | 14 |
| 1000 | 15 |

1-bit change

# Gray Code
## Analog → Digital

*Straight binary* number sequence for 7 to 8: 0111 → 1000; causes all four bits to change values.
*Gray* code for 7 → 8: 0100 to 1100; only the first bit changes from 0 to 1; the other three bits remain the same.

# Gray Code
## Algorithm

Step 0: Convert the decimal number to binary number.
Step 1: The MSB (Most Significant Bit) of a gray code and binary code is **the same**.
Step 2: The next digit of gray code is the XOR of the previous and current digit in the binary code.

Step 0: Convert the decimal number to binary number.

| $(20)_{10}$ | Binary Number | 1 | 0 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|
| | Gray Code | | | | | |

Step 1: The MSB (Most Significant Bit) of a gray code and binary code is **the same**.

| | | | | | | |
|---|---|---|---|---|---|---|
| $(20)_{10}$ | Binary Number | 1 | 0 | 1 | 0 | 0 |
| | Gray Code | 1 | | | | |

Step 2: The next digit of gray code is the XOR of the previous and current digit in the binary code.

| $(20)_{10}$ | Binary Number | 1 | 0 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|
| | Gray Code | 1 | $1 \oplus 0 = 1$ | | | |

Step 2: The next digit of gray code is the XOR of the previous and current digit in the binary code.

| $(20)_{10}$ | Binary Number | 1 | 0 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|
| | Gray Code | 1 | 1 | $0 \oplus 1 = 1$ | | |

Step 2: The next digit of gray code is the XOR of the previous and current digit in the binary code.

| $(20)_{10}$ | Binary Number | 1 | 0 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|
| | Gray Code | 1 | 1 | 1 | $1 \oplus 0 = 1$ | |

Step 2: The next digit of gray code is the XOR of the previous and current digit in the binary code.

| $(20)_{10}$ | Binary Number | 1 | 0 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|
| | Gray Code | 1 | 1 | 1 | 1 | $0 \oplus 0 = 0$ |

Step 2: The next digit of gray code is the XOR of the previous and current digit in the binary code.

| $(20)_{10}$ | Binary Number | 1 | 0 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|
| | Gray Code | 1 | 1 | 1 | 1 | 0 |

| $(21)_{10}$ | Binary Number | 1 | 0 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|
| | Gray Code | 1 | 1 | 1 | 1 | 1 |

# ASCII Code

American Standard Code for Information Interchange

# USASCII code chart

| $b_7$ $b_6$ $b_5$ → | 0 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1 0 0 | 1 0 1 | 1 1 0 | 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| Bits $b_4$ $b_3$ $b_2$ $b_1$ ↓ Row / Column → | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 0 0 0 | 0 | NUL | DLE | SP | 0 | @ | P | ` | p |
| 0 0 0 1 | 1 | SOH | DC1 | ! | 1 | A | Q | a | q |
| 0 0 1 0 | 2 | STX | DC2 | " | 2 | B | R | b | r |
| 0 0 1 1 | 3 | ETX | DC3 | # | 3 | C | S | c | s |
| 0 1 0 0 | 4 | EOT | DC4 | $ | 4 | D | T | d | t |
| 0 1 0 1 | 5 | ENQ | NAK | % | 5 | E | U | e | u |
| 0 1 1 0 | 6 | ACK | SYN | & | 6 | F | V | f | v |
| 0 1 1 1 | 7 | BEL | ETB | ' | 7 | G | W | g | w |
| 1 0 0 0 | 8 | BS | CAN | ( | 8 | H | X | h | x |
| 1 0 0 1 | 9 | HT | EM | ) | 9 | I | Y | i | y |
| 1 0 1 0 | 10 | LF | SUB | * | : | J | Z | j | z |
| 1 0 1 1 | 11 | VT | ESC | + | ; | K | [ | k | { |
| 1 1 0 0 | 12 | FF | FS | , | < | L | \ | l | \| |
| 1 1 0 1 | 13 | CR | GS | - | = | M | ] | m | } |
| 1 1 1 0 | 14 | SO | RS | . | > | N | ^ | n | ~ |
| 1 1 1 1 | 15 | SI | US | / | ? | O | _ | o | DEL |

"0" = $(011\ 0000)_2 = (48)_{10}$

**Table 1.7**
*American Standard Code for Information Interchange (ASCII)*

| $b_4b_3b_2b_1$ | $b_7b_6b_5$ = 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| 0000 | NUL | DLE | SP | 0 | @ | P | ` | p |
| 0001 | SOH | DC1 | ! | 1 | A | Q | a | q |
| 0010 | STX | DC2 | " | 2 | B | R | b | r |
| 0011 | ETX | DC3 | # | 3 | C | S | c | s |
| 0100 | EOT | DC4 | $ | 4 | D | T | d | t |
| 0101 | ENQ | NAK | % | 5 | E | U | e | u |
| 0110 | ACK | SYN | & | 6 | F | V | f | v |
| 0111 | BEL | ETB | ' | 7 | G | W | g | w |
| 1000 | BS | CAN | ( | 8 | H | X | h | x |
| 1001 | HT | EM | ) | 9 | I | Y | i | y |
| 1010 | LF | SUB | * | : | J | Z | j | z |
| 1011 | VT | ESC | + | ; | K | [ | k | { |
| 1100 | FF | FS | , | < | L | \ | l | \| |
| 1101 | CR | GS | – | = | M | ] | m | } |
| 1110 | SO | RS | . | > | N | ^ | n | ~ |
| 1111 | SI | US | / | ? | O | – | o | DEL |

**Control Characters**

| | | | |
|---|---|---|---|
| NUL | Null | DLE | Data-link escape |
| SOH | Start of heading | DC1 | Device control 1 |
| STX | Start of text | DC2 | Device control 2 |
| ETX | End of text | DC3 | Device control 3 |
| EOT | End of transmission | DC4 | Device control 4 |
| ENQ | Enquiry | NAK | Negative acknowledge |
| ACK | Acknowledge | SYN | Synchronous idle |
| BEL | Bell | ETB | End-of-transmission block |
| BS | Backspace | CAN | Cancel |
| HT | Horizontal tab | EM | End of medium |
| LF | Line feed | SUB | Substitute |
| VT | Vertical tab | ESC | Escape |
| FF | Form feed | FS | File separator |
| CR | Carriage return | GS | Group separator |
| SO | Shift out | RS | Record separator |
| SI | Shift in | US | Unit separator |
| SP | Space | DEL | Delete |

# Combinational Logic

## Binary Codes

# Combinational Logic
## Code Conversion

| Decimal Equivalent | BCD 8421 | Aiken 2421 | Excess-3 | 8, 4, −2, −1 | Gray Code |
|---|---|---|---|---|---|
| 0 | 0000 | 0000 | 0011 | 0000 | 0000 |
| 1 | 0001 | 0001 | 0100 | 0111 | 0001 |
| 2 | 0010 | 0010 | 0101 | 0110 | 0011 |
| 3 | 0011 | 0011 | 0110 | 0101 | 0010 |
| 4 | 0100 | 0100 | 0111 | 0100 | 0110 |
| 5 | 0101 | 1011 | 1000 | 1011 | 0111 |
| 6 | 0110 | 1100 | 1001 | 1010 | 0101 |
| 7 | 0111 | 1101 | 1010 | 1001 | 0100 |
| 8 | 1000 | 1110 | 1011 | 1000 | 1100 |
| 9 | 1001 | 1111 | 1100 | 1111 | 1101 |
| 10 | | | | | 1111 |
| 11 | | | | | 1110 |
| 12 | | | | | 1010 |
| 13 | | | | | 1011 |
| 14 | | | | | 1001 |
| 15 | | | | | 1000 |