SKYNEX ACADEMY · ANALOG AND DIGITAL SYSTEMS

Digital Electronics Exercise Book

266 logic exercises, from number systems to PLC programming. Write your answer in the box under each question: the box turns **green** when it is correct and **red** when it is wrong. 206 questions are checked automatically; for the others, compare with the answer and mark yourself.

Chapter 1

Number systems and codes

Binary, octal and hexadecimal numbers, binary arithmetic, signed numbers and the codes used inside digital equipment.

1.1 Binary and decimal

Key ideas

- Each binary place is worth twice the place on its right: …, 16, 8, 4, 2, 1 . ½, ¼, ⅛, …
- Decimal → binary: divide by 2 again and again and read the remainders from the bottom up.
- Decimal fraction → binary: multiply the fraction by 2 again and again and read the whole-number parts from the top down.
- With n bits you can count from 0 to $2^n - 1$.

1.1.1 Basic

Convert these binary numbers to decimal:

(a) $(1011)_2$

(b) $(11010)_2$

(c) $(100111)_2$

(d) $(11111111)_2$

YOUR ANSWER

(a) number (b) number

(c) number (d) number

ANSWER

(a) $(11)_{10}$

(b) $(26)_{10}$

(c) $(39)_{10}$

(d) $(255)_{10}$

1.1.2 Basic

Convert these decimal numbers to binary:

(a) $(19)_{10}$

(b) $(46)_{10}$

(c) $(100)_{10}$

(d) $(255)_{10}$

YOUR ANSWER

(a) binary, e.g. 1011 (b) binary, e.g. 1011

(c) binary, e.g. 1011 (d) binary, e.g. 1011

ANSWER

(a) $(10011)_2$

(b) $(101110)_2$

(c) $(1100100)_2$

(d) $(11111111)_2$

1.1.3 Basic

Show every step of converting 46 to binary by repeated division by 2.

YOUR ANSWER

46 in binary — binary, e.g. 1011

ANSWER

Division	Quotient	Remainder
46 ÷ 2	23	0
23 ÷ 2	11	1
11 ÷ 2	5	1
5 ÷ 2	2	1
2 ÷ 2	1	0
1 ÷ 2	0	1

Read the remainders from the bottom up: $46 = (101110)_2$.

1.1.4 Practice

Convert these binary numbers (with fractions) to decimal:

(a) $(101.11)_2$

(b) $(1100.011)_2$

(c) $(0.1011)_2$

YOUR ANSWER

(a) number (b) number

(c) number

ANSWER

(a) $(5.75)_{10}$

(b) $(12.375)_{10}$

(c) $(0.6875)_{10}$

1.1.5 Practice

Convert to binary (give at most 6 bits after the point):

(a) $(6.75)_{10}$

(b) $(13.625)_{10}$

(c) $(0.3)_{10}$

YOUR ANSWER

(a) binary, e.g. 1011 (b) binary, e.g. 1011

(c) binary, e.g. 1011

ANSWER

(a) $(110.11)_2$

(b) $(1101.101)_2$

(c) $(0.010011)_2$ (0.3 never ends in binary; 0.010011… ≈ 0.297)

1.1.6 Basic

What is the largest decimal number you can write with (a) 6 bits (b) 8 bits (c) 12 bits?

YOUR ANSWER

(a) number (b) number

(c) number

ANSWER

(a) 2^6 − 1 = 63

(b) 2^8 − 1 = 255

(c) 2^12 − 1 = 4095

1.1.7 Practice

How many bits are needed to write (a) 1000 (b) 5000 (c) 65 535 in binary?

YOUR ANSWER

(a) number (b) number

(c) number

ANSWER

(a) 10 bits (2^9 = 512 is too small, 2^10 = 1024 is enough)

(b) 13 bits (2^12 = 4096 is too small, 2^13 = 8192 is enough)

(c) 16 bits (2^15 = 32768 is too small, 2^16 = 65536 is enough)

1.1.8 Challenge

A sensor sends a reading as 10 bits. (a) How many different values can it send? (b) If 0 means 0 °C and the largest value means 100 °C, what temperature does the reading 1000000000 stand for?

YOUR ANSWER

(a) values number (b) temperature e.g. 1.5 k°C

ANSWER

(a) $2^{10} = 1024$ values (0 to 1023).

(b) $1000000000_2 = 512$, so the temperature is $512 \times 100 / 1023 \approx 50$ °C.

1.2 Octal and hexadecimal

Key ideas

- Octal uses the digits 0–7; one octal digit = 3 bits.
- Hexadecimal uses 0–9 and A–F (A = 10 … F = 15); one hex digit = 4 bits.
- Binary → octal/hex: group the bits in 3s or 4s starting from the point, adding zeros at the ends if needed.

1.2.1 Basic

Convert these octal numbers to decimal:

(a) $(57)_8$

(b) $(144)_8$

(c) $(777)_8$

(d) $(1750)_8$

YOUR ANSWER

(a) number (b) number

(c) number (d) number

ANSWER

(a) $(47)_{10}$

(b) $(100)_{10}$

(c) $(511)_{10}$

(d) $(1000)_{10}$

1.2.2 Basic

Convert these decimal numbers to octal:

(a) $(100)_{10}$

(b) $(365)_{10}$

(c) $(1000)_{10}$

YOUR ANSWER

(a) octal (b) octal

(c) octal

ANSWER

(a) $(144)_{8}$

(b) $(555)_{8}$

(c) $(1750)_{8}$

1.2.3 Basic

Convert these binary numbers to octal:

(a) $(101110)_2$

(b) $(11011011)_2$

(c) $(1101.0111)_2$

YOUR ANSWER

(a) octal (b) octal

(c) octal

ANSWER

(a) $(56)_8$

(b) $(333)_8$

(c) $(15.34)_8$

1.2.4 Basic

Convert these octal numbers to binary:

(a) $(47)_8$

(b) $(362)_8$

(c) $(7.25)_8$

YOUR ANSWER

(a) binary, e.g. 1011

(b) binary, e.g. 1011

(c) binary, e.g. 1011

ANSWER

(a) $(100111)_2$

(b) $(11110010)_2$

(c) $(111.010101)_2$

1.2.5 Basic

Convert these hexadecimal numbers to decimal:

(a) $(2F)_{16}$

(b) $(A0)_{16}$

(c) $(3E8)_{16}$

(d) $(FFFF)_{16}$

YOUR ANSWER

(a) number (b) number

(c) number (d) number

ANSWER

(a) $(47)_{10}$

(b) $(160)_{10}$

(c) $(1000)_{10}$

(d) $(65535)_{10}$

1.2.6 Basic

Convert these decimal numbers to hexadecimal:

(a) $(75)_{10}$

(b) $(255)_{10}$

(c) $(4096)_{10}$

(d) $(2025)_{10}$

YOUR ANSWER

(a) hex, e.g. 3F (b) hex, e.g. 3F

(c) hex, e.g. 3F (d) hex, e.g. 3F

ANSWER

(a) $(4B)_{16}$

(b) $(FF)_{16}$

(c) $(1000)_{16}$

(d) $(7E9)_{16}$

1.2.7 Basic

Convert these binary numbers to hexadecimal:

(a) $(11010110)_2$

(b) $(101111100101)_2$

(c) $(1110.1011)_2$

YOUR ANSWER

(a) hex, e.g. 3F

(b) hex, e.g. 3F

(c) hex, e.g. 3F

ANSWER

(a) $(D6)_{16}$

(b) $(BE5)_{16}$

(c) $(E.B)_{16}$

1.2.8 Basic

Convert these hexadecimal numbers to binary:

(a) $(7C)_{16}$

(b) $(B3F)_{16}$

(c) $(1A.8)_{16}$

YOUR ANSWER

(a) binary, e.g. 1011 (b) binary, e.g. 1011

(c) binary, e.g. 1011

ANSWER

(a) $(0111\ 1100)_2$ (each hex digit becomes 4 bits)

(b) $(1011\ 0011\ 1111)_2$ (each hex digit becomes 4 bits)

(c) $(0001\ 1010.1000)_2$ (each hex digit becomes 4 bits)

1.2.9 Practice

Convert $(3F7)_{16}$ to octal by going through binary.

YOUR ANSWER

in octal octal

ANSWER

$(3F7)_{16} = 11\ 1111\ 0111 = 001\ 111\ 110\ 111 = (1767)_8$

1.2.10 Practice

Convert $(2735)_8$ to hexadecimal.

YOUR ANSWER

in hex hex, e.g. 3F

ANSWER

$(2735)_8 = 10111011101_2 = (5DD)_{16}$ (= 1501 in decimal)

1.2.11 Challenge

A memory chip uses the addresses 0000_{16} to $3FFF_{16}$. (a) How many memory locations does it have? (b) How many address lines (bits) does it need?

YOUR ANSWER

(a) locations number (b) address lines number

ANSWER

(a) $3FFF_{16} = 16383$, so there are 16384 locations (16 K).

(b) $16384 = 2^{14}$, so 14 address lines.

1.2.12 Challenge

Web colours are written as three hex bytes: #RRGGBB. Give the red, green and blue values in decimal for (a) #FF8000 (b) #1E90FF.

YOUR ANSWER

(a) R number (a) G number

(a) B number (b) R number

(b) G number (b) B number

ANSWER

(a) R = FF = 255, G = 80 = 128, B = 00 = 0 (orange)

(b) R = 1E = 30, G = 90 = 144, B = FF = 255 (blue)

1.3 Binary arithmetic

Key ideas

- Addition: 0+0 = 0, 0+1 = 1, 1+1 = 0 carry 1, 1+1+1 = 1 carry 1.
- Subtraction: 0−0 = 0, 1−0 = 1, 1−1 = 0, 0−1 = 1 borrow 1.
- Multiplication and division work like in decimal, but every partial product is either 0 or the number itself.

1.3.1 Basic

Add:

(a) 1011 + 0110

(b) 11101 + 10111

(c) 1011.01 + 110.11

(d) 1111 + 1111

YOUR ANSWER

(a) binary, e.g. 1011

(b) binary, e.g. 1011

(c) binary, e.g. 1011

(d) binary, e.g. 1011

ANSWER

(a) 10001 (11 + 6 = 17)

(b) 110100 (29 + 23 = 52)

(c) 10010 (11.25 + 6.75 = 18)

(d) 11110 (15 + 15 = 30)

1.3.2 Basic

Subtract:

(a) 1101 − 0110

(b) 101000 − 10011

(c) 1100.1 − 11.11

YOUR ANSWER

(a) binary, e.g. 1011

(b) binary, e.g. 1011

(c) binary, e.g. 1011

ANSWER

(a) 111 (13 − 6 = 7)

(b) 10101 (40 − 19 = 21)

(c) 1000.11 (12.5 − 3.75 = 8.75)

1.3.3 Practice

Multiply:

(a) 1011 × 101

(b) 1101 × 110

(c) 111 × 111

YOUR ANSWER

(a) binary, e.g. 1011

(b) binary, e.g. 1011

(c) binary, e.g. 1011

ANSWER

(a) 110111 (11 × 5 = 55)

(b) 1001110 (13 × 6 = 78)

(c) 110001 (7 × 7 = 49)

1.3.4 Practice

Divide:

(a) 110110 ÷ 110

(b) 1111000 ÷ 1010

(c) 1010001 ÷ 1001

YOUR ANSWER

(a) binary, e.g. 1011 (b) binary, e.g. 1011

(c) binary, e.g. 1011

ANSWER

(a) 1001 (54 ÷ 6 = 9)

(b) 1100 (120 ÷ 10 = 12)

(c) 1001 (81 ÷ 9 = 9)

1.3.5 Practice

Add the three numbers 1101 + 1011 + 111 and check your answer in decimal.

YOUR ANSWER

sum in binary binary, e.g. 1011

ANSWER

11111_2 = 31 (13 + 11 + 7 = 31).

1.3.6 Challenge

Multiplying a binary number by 2 shifts it one place to the left. What do you get when you shift 0010 1101 (a) one place left (b) two places right? Give the decimal values too.

YOUR ANSWER

(a) binary binary, e.g. 1011 (b) binary binary, e.g. 1011

ANSWER

(a) 0101 1010 = 90 (= 45 × 2)

(b) 0000 1011 = 11 (= 45 ÷ 4, the remainder 1 is lost)

1.4 Complements and signed numbers

Key ideas

- 1's complement: change every 0 into 1 and every 1 into 0.
- 2's complement: the 1's complement plus 1. It is how computers store negative numbers.
- In n-bit 2's complement the left bit is the sign (0 = +, 1 = –) and the range is -2^{n-1} to $+2^{n-1} - 1$.
- A – B = A + (2's complement of B); ignore the carry out of the left-most bit.

1.4.1 Basic

Give the 1's complement of:

(a) 1011

(b) 01101100

(c) 11110000

YOUR ANSWER

(a) e.g. 0110 (b) e.g. 0110

(c) e.g. 0110

ANSWER

(a) 0100

(b) 10010011

(c) 00001111

1.4.2 Basic

Give the 2’s complement of:

(a) 0101

(b) 00011010

(c) 10000000

(d) 11111111

YOUR ANSWER

(a) e.g. 0110 (b) e.g. 0110

(c) e.g. 0110 (d) e.g. 0110

ANSWER

(a) 1011

(b) 11100110

(c) 10000000

(d) 00000001

1.4.3 Basic

Write these numbers in 8-bit 2’s complement:

(a) +37

(b) –37

(c) –100

(d) –1

(e) –128

YOUR ANSWER

(a) e.g. 0110 (b) e.g. 0110

(c) e.g. 0110 (d) e.g. 0110

(e) e.g. 0110

ANSWER

(a) 00100101

(b) 11011011

(c) 10011100

(d) 11111111

(e) 10000000

1.4.4 Practice

What decimal values do these 8-bit 2’s complement numbers stand for?

(a) 00101101

(b) 11010011

(c) 10000001

(d) 11111110

YOUR ANSWER

(a) number (b) number

(c) number (d) number

ANSWER

(a) +45

(b) −45

(c) −127

(d) −2

1.4.5 Practice

Use the 2's complement method to work out (8 bits): (a) 45 − 19 (b) 25 − 40 (c) −20 − 30.

YOUR ANSWER

(a) number (b) number

(c) number

ANSWER

(a) 00101101 + 11101101 = 1 00011010 → ignore the carry → 00011010 = +26

(b) 00011001 + 11011000 = 11110001 → no carry, so the result is negative: −(2's complement of 11110001) = −15

(c) 11101100 + 11100010 = 1 11001110 → ignore the carry → 11001110 = −50

1.4.6 Practice

Use 1's complement subtraction (with end-around carry) to work out 1011 − 0110.

YOUR ANSWER

result (binary) binary, e.g. 1011

ANSWER

1's complement of 0110 = 1001.

1011 + 1001 = 1 0100 → the carry 1 is added back to the result (end-around carry): 0100 + 1 = 0101.

Answer: 0101 = 5 (11 − 6 = 5).

1.4.7 Basic

What is the range of numbers you can store in (a) 8-bit (b) 16-bit 2’s complement?

YOUR ANSWER

(a) smallest number (a) largest number

(b) smallest number (b) largest number

ANSWER

(a) −128 to +127

(b) −32 768 to +32 767

1.4.8 Challenge

Add the 8-bit signed numbers 01110000 and 00110000. Is the answer correct? Explain what happened.

YOUR ANSWER

8-bit result e.g. 0110 Is it correct? (yes/no) answer

ANSWER

01110000 (+112) + 00110000 (+48) = 10100000. In 8-bit 2’s complement this reads −96, which is wrong: +160 is bigger than +127. This is an **overflow**: two positive numbers gave a negative-looking result. Overflow happens when both numbers have the same sign and the result has the other sign.

1.5 Binary codes: BCD, Excess-3, Gray and ASCII

Key ideas

- BCD (8421): every decimal digit is written as its own 4-bit group. 1010 to 1111 are not used.
- Excess-3: every digit is written as (digit + 3) in 4 bits. It is self-complementing.
- Gray code: only one bit changes between two neighbouring numbers. G = B ⊕ (B shifted right by one).
- ASCII: a 7-bit code for letters, digits and signs, e.g. ‘A’ = 65 = 41_{16}.

1.5.1 Basic

Write in BCD:

(a) 59

(b) 306

(c) 2025

YOUR ANSWER

(a) e.g. 0110 (b) e.g. 0110

(c) e.g. 0110

ANSWER

(a) 0101 1001

(b) 0011 0000 0110

(c) 0010 0000 0010 0101

1.5.2 Basic

Decode these BCD numbers:

(a) 1001 0011

(b) 0111 0000 0110

(c) 0001 0101 1000 0100

YOUR ANSWER

(a) number (b) number

(c) number

ANSWER

(a) 93

(b) 706

(c) 1584

1.5.3 Practice

Compare the binary and BCD forms of 97. Which needs more bits?

YOUR ANSWER

binary binary, e.g. 1011 BCD e.g. 0110

needs more bits answer

ANSWER

Binary: 1100001 (7 bits). BCD: 1001 0111 (8 bits). BCD needs more bits but is easier to show on a display.

1.5.4 Basic

Is 0110 1101 a valid BCD number? Explain.

YOUR ANSWER

valid? (yes/no) answer

ANSWER

No. The right group 1101 = 13 is not a decimal digit; BCD only uses 0000 to 1001.

1.5.5 Challenge

Add 47 + 35 in BCD. Show the correction step.

YOUR ANSWER

BCD sum e.g. 0110

ANSWER

0100 0111 (47)

+ 0011 0101 (35)

= 0111 1100

The right group 1100 is more than 9, so add 0110: 1100 + 0110 = 1 0010 → write 0010, carry 1 into the next group: 0111 + 1 = 1000.

Result: 1000 0010 = 82. ✓

1.5.6 Basic

Write in Excess-3:

(a) 4

(b) 27

(c) 905

YOUR ANSWER

(a) e.g. 0110 (b) e.g. 0110

(c) e.g. 0110

ANSWER

(a) 0111

(b) 0101 1010

(c) 1100 0011 1000

1.5.7 Practice

Decode the Excess-3 number 0110 1001 1011.

YOUR ANSWER

decimal number

ANSWER

Subtract 3 from each group: 0110 → 3, 1001 → 6, 1011 → 8. Answer: 368.

1.5.8 Challenge

Show that Excess-3 is self-complementing using the digit 2.

YOUR ANSWER

Excess-3 of 2 e.g. 0110 Excess-3 of 7 e.g. 0110

ANSWER

2 in Excess-3 = 0101. Its 9's complement is 7 = 1010. Inverting every bit of 0101 gives 1010, which is exactly the Excess-3 code of 7: the 1's complement of the code gives the code of the 9's complement.

1.5.9 Basic

Convert binary to Gray code:

(a) 0110

(b) 1011

(c) 1111

(d) 10010

YOUR ANSWER

(a) e.g. 0110 (b) e.g. 0110

(c) e.g. 0110 (d) e.g. 0110

ANSWER

(a) 0101

(b) 1110

(c) 1000

(d) 11011

1.5.10 Basic

Convert Gray code to binary:

(a) 1101

(b) 1000

(c) 0111

YOUR ANSWER

(a) e.g. 0110 (b) e.g. 0110

(c) e.g. 0110

ANSWER

(a) 1001

(b) 1111

(c) 0101

1.5.11 Practice

Write the full 3-bit Gray code sequence from 0 to 7.

YOUR ANSWER

Gray codes 0 → 7 e.g. 0110

ANSWER

Decimal	Binary	Gray
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

1.5.12 Practice

Why is Gray code used in rotary position encoders?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Only one bit changes from one position to the next, so the reader can never see a false value while several bits change at slightly different moments (e.g. 0111 → 1000 in binary could be read as 1111 for a moment).

1.5.13 Basic

Give the ASCII code of the letters S, K and Y in decimal, hex and 7-bit binary.

YOUR ANSWER

S decimal number | S hex hex, e.g. 3F

K decimal number | K hex hex, e.g. 3F

Y decimal number | Y hex hex, e.g. 3F

ANSWER

Letter	Decimal	Hex	Binary
S	83	53	1010011
K	75	4B	1001011
Y	89	59	1011001

1.5.14 Practice

What is the difference between the ASCII codes of ‘a’ and ‘A’? How can a program change a lowercase letter into a capital?

YOUR ANSWER

difference number

ANSWER

‘a’ = 97, ‘A’ = 65: the difference is 32 = 0100000_2. Clearing bit 5 (or subtracting 32) turns any lowercase letter into a capital.

1.5.15 Practice

The character ‘7’ and the number 7 are not the same in a computer. Give both in binary.

YOUR ANSWER

number 7 binary, e.g. 1011 character ‘7’ binary, e.g. 1011

ANSWER

Number 7 = 0000111. Character ‘7’ (ASCII 55) = 0110111.

Chapter 2

Logic gates

The basic gates, their truth tables and symbols, universal gates and tristate outputs.

2.1 Basic gates and truth tables

Key ideas

- NOT: Y = A'. AND: Y = A·B (1 only if all inputs are 1). OR: Y = A + B (1 if any input is 1).
- NAND = AND followed by NOT; NOR = OR followed by NOT.
- XOR: Y = A ⊕ B = A'B + AB' (1 when the inputs are different). XNOR: 1 when the inputs are equal.
- A gate with n inputs has 2^n rows in its truth table.

2.1.1 Basic

Write the truth table of a 2-input AND, OR, NAND, NOR, XOR and XNOR gate in one table.

YOUR ANSWER

AND column e.g. 0110 OR column e.g. 0110

NAND column e.g. 0110 NOR column e.g. 0110

XOR column e.g. 0110 XNOR column e.g. 0110

ANSWER

A	B	AND	OR	NAND	NOR	XOR	XNOR
0	0	0	0	1	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	1	0	0	0	1

2.1.2 Basic

Write the truth table of a 3-input NAND gate and a 3-input NOR gate.

YOUR ANSWER

NAND column e.g. 0110 NOR column e.g. 0110

ANSWER

A	B	C	NAND	NOR
0	0	0	1	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	0

2.1.3 Basic

How many rows does the truth table of a 5-input AND gate have? In how many of them is the output 1?

YOUR ANSWER

rows number rows with output 1 number

ANSWER

$2^5 = 32$ rows. The output is 1 in only one row (all inputs 1).

2.1.4 Practice

Two signals are sampled at 8 moments: A = 11001010, B = 10100110. Give the output of (a) AND (b) OR (c) XOR (d) NAND for each moment.

YOUR ANSWER

(a) e.g. 0110 (b) e.g. 0110

(c) e.g. 0110 (d) e.g. 0110

ANSWER

(a) 10000010

(b) 11101110

(c) 01101100

(d) 01111101

2.1.5 Practice

A gate gives these outputs for AB = 00, 01, 10, 11: 1, 0, 0, 0. Which gate is it? And for 0, 1, 1, 0?

YOUR ANSWER

1, 0, 0, 0 answer 0, 1, 1, 0 answer

ANSWER

1, 0, 0, 0 is a NOR gate. 0, 1, 1, 0 is an XOR gate.

2.1.6 Basic

Name the TTL chip that holds each of these: four 2-input NAND gates, six NOT gates, four 2-input AND gates, four 2-input OR gates, four 2-input XOR gates, four 2-input NOR gates.

YOUR ANSWER

NAND answer NOT answer

AND answer OR answer

XOR answer NOR answer

ANSWER

7400 (NAND), 7404 (NOT), 7408 (AND), 7432 (OR), 7486 (XOR), 7402 (NOR). On these 14-pin chips pin 14 is +5 V and pin 7 is ground.

2.1.7 Challenge

A NAND gate has inputs A and B. Its output goes to a NOR gate together with input C. Write the expression for the output Y and its truth table.

YOUR ANSWER

Y = e.g. AB' + C

ANSWER

Y = ((AB)' + C)' = AB·C'

A	B	C	Y
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	1
1	1	1	0

2.2 Negative-AND, Negative-OR, XOR uses and the INHIBIT gate

Key ideas

- Negative-AND (inputs inverted, then AND) = NOR. Negative-OR = NAND (De Morgan).
- XOR gives the parity of its inputs: the output is 1 when an odd number of inputs are 1.
- INHIBIT gate: Y = A·B'. Input B blocks (inhibits) A.

2.2.1 Basic

Show with a truth table that a Negative-AND gate (A'·B') behaves like a NOR gate.

YOUR ANSWER

A'·B' column e.g. 0110

ANSWER

A	B	A'·B'	(A+B)'
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

2.2.2 Basic

Show with a truth table that a Negative-OR gate (A' + B') behaves like a NAND gate.

YOUR ANSWER

A' + B' column e.g. 0110

ANSWER

A	B	A'+B'	(AB)'
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

2.2.3 Practice

An even-parity bit is added to 7-bit data so the total number of 1s is even. Find the parity bit for:

(a) 1011000

(b) 1110111

(c) 0110100

(d) 1000001

YOUR ANSWER

(a) number (b) number

(c) number (d) number

ANSWER

(a) 1 → sent as 10110001

(b) 0 → sent as 11101110

(c) 1 → sent as 01101001

(d) 0 → sent as 10000010

2.2.4 Practice

Which gate can you use to check whether two bits are equal? Draw the idea for comparing two 2-bit numbers A1A0 and B1B0.

YOUR ANSWER

gate answer

ANSWER

An XNOR gate gives 1 when its two inputs are equal. For 2-bit numbers: E = (A1 ⊙ B1)·(A0 ⊙ B0): two XNOR gates followed by an AND gate.

2.2.5 Basic

Write the truth table of an INHIBIT gate Y = A·B'. What does input B do?

YOUR ANSWER

Y column e.g. 0110

ANSWER

A	B	Y
0	0	0
0	1	0
1	0	1
1	1	0

When B = 1 the output is always 0 (A is blocked). When B = 0 the output follows A.

2.2.6 Challenge

A machine motor M may only run when the start switch S is on and the safety guard G is NOT open (G = 1 means open). Which gate do you need? Write the expression.

YOUR ANSWER

gate answer M = e.g. AB' + C

ANSWER

An INHIBIT gate: M = S·G'. The guard input inhibits the start switch.

2.3 Universal gates and tristate logic

Key ideas

- NAND and NOR are universal: any logic circuit can be built with only NAND gates or only NOR gates.
- A tristate (three-state) output can be 0, 1 or Z (high impedance, disconnected). The enable input decides.
- Several tristate outputs can share one bus wire, but only one may be enabled at a time.

2.3.1 Basic

Show how to make a NOT gate from (a) one NAND gate (b) one NOR gate.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

(a) Join both NAND inputs together: (A·A)' = A'.

(b) Join both NOR inputs together: (A + A)' = A'.

2.3.2 Practice

Build a 2-input AND gate and a 2-input OR gate using only NAND gates. How many gates does each need?

YOUR ANSWER

NAND gates for AND number NAND gates for OR number

ANSWER

AND: NAND followed by a NAND used as an inverter: ((AB)')' = AB → 2 gates.

OR: invert each input with a NAND, then NAND them: (A'·B')' = A + B → 3 gates.

2.3.3 Practice

Build a 2-input OR gate and a 2-input AND gate using only NOR gates.

YOUR ANSWER

NOR gates for OR number NOR gates for AND number

ANSWER

OR: NOR followed by a NOR inverter → 2 gates.

AND: invert each input with a NOR, then NOR them: (A' + B')' = AB → 3 gates.

2.3.4 Challenge

Build an XOR gate from four 2-input NAND gates. Give the output of each gate.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

G1 = (AB)'. G2 = (A·G1)'. G3 = (B·G1)'. Y = (G2·G3)' = A·G1 + B·G1 = (A + B)(AB)' = A'B + AB' = A ⊕ B.

2.3.5 Basic

A tristate buffer has data input A and active-high enable E. Write its truth table.

YOUR ANSWER

Y for EA = 00, 01, 10, 11 e.g. 0110

ANSWER

E	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1

2.3.6 Practice

Two tristate buffers drive the same bus wire. Buffer 1 outputs 1 and buffer 2 outputs 0, and both are enabled. What happens, and how is it avoided?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Bus contention: one output pushes the wire high while the other pulls it low. A large current flows, the level is unknown and the chips can be damaged. A decoder is used so that only one enable is active at a time.

Chapter 3

Boolean algebra and Karnaugh maps

The laws of Boolean algebra, SOP and POS forms, minterms and maxterms, and simplification with Karnaugh maps.

3.1 Laws and algebraic simplification

Key ideas

- A + 0 = A, A·1 = A, A + 1 = 1, A·0 = 0, A + A = A, A·A = A, A + A' = 1, A·A' = 0, (A')' = A.
- Absorption: A + AB = A, A(A + B) = A, A + A'B = A + B.
- Distributive: A(B + C) = AB + AC and A + BC = (A + B)(A + C).
- De Morgan: (A + B)' = A'B' and (AB)' = A' + B'.

3.1.1 Basic

Name the law used in each line: (a) A + 1 = 1 (b) A·A' = 0 (c) A + AB = A (d) (AB)' = A' + B' (e) A(B + C) = AB + AC

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

(a) annulment / OR with 1 (b) complement law (c) absorption (d) De Morgan’s theorem (e) distributive law

3.1.2 Basic

Simplify: Y = AB + AB' + A'B

YOUR ANSWER

Y = e.g. AB' + C

ANSWER

Y = A + B

3.1.3 Basic

Simplify: Y = A + AB

YOUR ANSWER

Y = e.g. AB' + C

ANSWER

Y = A

3.1.4 Basic

Simplify: Y = A(A + B)

YOUR ANSWER

Y = e.g. AB' + C

ANSWER

Y = A

3.1.5 Basic

Simplify: Y = A + A'B

YOUR ANSWER

Y = e.g. AB' + C

ANSWER

Y = A + B

3.1.6 Practice

Simplify: Y = (A + B)(A + C)

YOUR ANSWER

Y = e.g. AB' + C

ANSWER

Y = A + BC

3.1.7 Practice

Simplify: Y = AB + A'C + BC

YOUR ANSWER

Y = e.g. AB' + C

ANSWER

Y = AB + A'C

3.1.8 Practice

Simplify: Y = A'B'C + A'BC + AB'C + ABC

YOUR ANSWER

Y = e.g. AB' + C

ANSWER

Y = C

3.1.9 Practice

Simplify: Y = (A + B')(A' + B)

YOUR ANSWER

Y = e.g. AB' + C

ANSWER

Y = AB + A'B'

3.1.10 Practice

Simplify: Y = A'B'C' + A'BC' + AB'C' + ABC'

YOUR ANSWER

Y = e.g. AB' + C

ANSWER

Y = C'

3.1.11 Practice

Simplify: Y = ABC + ABC' + AB'C

YOUR ANSWER

Y = e.g. AB' + C

ANSWER

Y = AB + AC

3.1.12 Practice

Use De Morgan’s theorem to write these without a bar over more than one letter: (a) (A + B + C)' (b) (AB + C)' (c) ((A + B)C')'

YOUR ANSWER

(a) e.g. AB' + C (b) e.g. AB' + C

(c) e.g. AB' + C

ANSWER

(a) A'B'C'

(b) (AB)'·C' = (A' + B')C'

(c) (A + B)' + C = A'B' + C

3.1.13 Challenge

Prove with algebra that AB + A'C + BC = AB + A'C (the consensus theorem).

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

BC = BC(A + A') = ABC + A'BC.

So AB + A'C + BC = AB + ABC + A'C + A'BC = AB(1 + C) + A'C(1 + B) = AB + A'C. ✓

3.1.14 Practice

Check with a truth table that A + A'B = A + B.

YOUR ANSWER

A + A'B column e.g. 0110 A + B column e.g. 0110

ANSWER

A	B	A + A'B	A + B
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	1

3.2 SOP, POS, minterms and maxterms

Key ideas

- A minterm is an AND term that contains every variable once; it is 1 for exactly one row. m3 for ABC is A'BC.
- A maxterm is an OR term that is 0 for exactly one row. M3 for ABC is (A + B' + C').
- F = Σm(rows where F = 1) = ΠM(rows where F = 0).

3.2.1 Basic

Write the minterm and the maxterm for row 5 of a 3-variable truth table (A is the MSB).

YOUR ANSWER

m5 = e.g. AB' + C M5 = e.g. AB' + C

ANSWER

Row 5 = 101 → minterm m5 = AB'C, maxterm M5 = (A' + B + C').

3.2.2 Basic

F(A, B, C) = Σm(1, 4, 6, 7). Write F as a canonical SOP expression and as a list of maxterms.

YOUR ANSWER

F (SOP) = e.g. AB' + C maxterm numbers e.g. a, b, c

ANSWER

F = A'B'C + AB'C' + ABC' + ABC

F = ΠM(0, 2, 3, 5)

3.2.3 Practice

Write the truth table of F(A, B, C) = Σm(0, 3, 5, 6). What simple gate circuit does it describe?

YOUR ANSWER

F column e.g. 0110 gate answer

ANSWER

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

F is 1 when an even number of inputs are 1: F = (A ⊕ B ⊕ C)' (an XNOR of three inputs).

3.2.4 Practice

Convert F(A, B, C, D) = ΠM(0, 2, 5, 7, 8, 10, 13, 15) into Σm form.

YOUR ANSWER

minterm numbers e.g. a, b, c

ANSWER

F = Σm(1, 3, 4, 6, 9, 11, 12, 14)

3.3 Karnaugh maps

Key ideas

- Rows and columns of a K-map are in Gray order (00, 01, 11, 10) so neighbours differ by one variable.
- Group the 1s in rectangles of 1, 2, 4, 8 or 16 cells; groups may wrap around the edges and may overlap.
- Make the groups as big as possible and use as few groups as possible.
- Group the 0s instead to get the POS form.

3.3.1 Basic

Use a 3-variable K-map to simplify F(A, B, C) = Σm(1, 3, 5, 7). Also give the POS form.

YOUR ANSWER

F = e.g. AB' + C

ANSWER

F = C (POS: F = C)

3.3.2 Basic

Use a 3-variable K-map to simplify F(A, B, C) = Σm(0, 2, 4, 6).

YOUR ANSWER

F = e.g. AB' + C

ANSWER

F = C'

3.3.3 Basic

Use a 3-variable K-map to simplify F(A, B, C) = Σm(1, 2, 3, 7).

YOUR ANSWER

F = e.g. AB' + C

ANSWER

F = BC + A'B + A'C

3.3.4 Basic

Use a 3-variable K-map to simplify F(A, B, C) = Σm(0, 1, 2, 4, 5).

YOUR ANSWER

F = e.g. AB' + C

ANSWER

F = B' + A'C'

3.3.5 Basic

Use a 3-variable K-map to simplify F(A, B, C) = Σm(2, 3, 4, 5, 6).

YOUR ANSWER

F = e.g. AB' + C

ANSWER

F = AB' + A'B + BC'

3.3.6 Practice

Use a 4-variable K-map to simplify F(A, B, C, D) = Σm(0, 1, 2, 5, 8, 9, 10).

YOUR ANSWER

F = e.g. AB' + C

ANSWER

F = B'C' + B'D' + A'C'D

3.3.7 Practice

Use a 4-variable K-map to simplify F(A, B, C, D) = Σm(4, 5, 6, 7, 12, 13, 14, 15).

YOUR ANSWER

F = e.g. AB' + C

ANSWER

F = B

3.3.8 Practice

Use a 4-variable K-map to simplify F(A, B, C, D) = Σm(0, 2, 5, 7, 8, 10, 13, 15).

YOUR ANSWER

F = e.g. AB' + C

ANSWER

F = BD + B'D'

3.3.9 Practice

Use a 4-variable K-map to simplify F(A, B, C, D) = Σm(1, 3, 4, 6, 9, 11, 12, 14).

YOUR ANSWER

F = e.g. AB' + C

ANSWER

F = BD' + B'D

3.3.10 Practice

Use a 4-variable K-map to simplify F(A, B, C, D) = Σm(0, 1, 3, 7, 8, 9, 11, 15).

YOUR ANSWER

F = e.g. AB' + C

ANSWER

F = CD + B'C'

3.3.11 Practice

Use a 4-variable K-map to simplify F(A, B, C, D) = Σm(2, 3, 6, 7, 10, 11, 13, 15).

YOUR ANSWER

F = e.g. AB' + C

ANSWER

F = A'C + B'C + ABD

3.3.12 Practice

Give the minimal POS form of F(A, B, C, D) = Σm(0, 1, 2, 3, 4, 6, 8, 9, 10, 11).

YOUR ANSWER

F (POS) = e.g. AB' + C

ANSWER

F = (A' + B')(B' + D') (SOP for comparison: F = B' + A'D')

3.3.13 Challenge

Draw the 4-variable K-map of F = A'B'C'D' + A'B'CD' + AB'C'D' + AB'CD' and simplify it. Why is the answer so short?

YOUR ANSWER

F = e.g. AB' + C

ANSWER

F = B'D'. The four 1s are the four corners of the map; corners are neighbours because the map wraps around in both directions.

3.4 Don’t-care conditions and design problems

Key ideas

- A don’t-care (x or d) is an input combination that never happens or whose output does not matter.
- Use a don’t-care as a 1 when it makes a group bigger; otherwise ignore it.
- BCD inputs 1010 to 1111 never happen, so they are don’t-cares.

3.4.1 Practice

Simplify F(A, B, C, D) = Σm(1, 3, 7, 11, 15) with don’t-cares d(0, 2, 5).

YOUR ANSWER

F = e.g. AB' + C

ANSWER

F = CD + A'B'

3.4.2 Practice

Simplify F(A, B, C, D) = Σm(0, 4, 8, 12) with don’t-cares d(2, 6, 10, 14).

YOUR ANSWER

F = e.g. AB' + C

ANSWER

F = D'

3.4.3 Practice

Simplify F(A, B, C, D) = Σm(5, 6, 7) with don’t-cares d(10, 11, 12, 13, 14, 15).

YOUR ANSWER

F = e.g. AB' + C

ANSWER

F = BC + BD

3.4.4 Practice

Three judges vote with switches A, B and C. A lamp lights when at least two judges press their switch. Write the truth table and the simplest expression.

YOUR ANSWER

L = e.g. AB' + C

ANSWER

A	B	C	L
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

L = AB + AC + BC

3.4.5 Challenge

A BCD digit ABCD drives a 7-segment display. Segment a (the top bar) is on for the digits 0, 2, 3, 5, 6, 7, 8 and 9. Find the simplest expression for segment a, using don’t-cares.

YOUR ANSWER

a = e.g. AB' + C

ANSWER

a = A + C + BD + B'D'

3.4.6 Challenge

Design a circuit with a BCD input ABCD whose output is 1 when the digit can be divided by 3 (0 counts as not divisible). Use don’t-cares.

YOUR ANSWER

F = e.g. AB' + C

ANSWER

Digits 3, 6, 9 ⟶ F = AD + BCD' + B'CD

3.4.7 Challenge

Implement F = AB + CD using only NAND gates.

YOUR ANSWER

NAND gates number

ANSWER

F = AB + CD = ((AB)'·(CD)')' ⟶ NAND A with B, NAND C with D, then NAND the two results: 3 NAND gates.

3.4.8 Practice

A car alarm Z sounds when the engine is off (E = 0) and either a door is opened (D = 1) or the glass breaks (G = 1). Write the expression and the truth table.

YOUR ANSWER

Z = e.g. AB' + C

ANSWER

Z = E'(D + G)

E	D	G	Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Chapter 4

Combinational circuits

Adders, subtractors, comparators, multiplexers, demultiplexers, encoders and decoders.

4.1 Adders and subtractors

Key ideas

- Half adder: S = A ⊕ B, C = AB.
- Full adder: S = A ⊕ B ⊕ Cin, Cout = AB + Cin(A ⊕ B). Two half adders + an OR gate make a full adder.
- Half subtractor: D = A ⊕ B, Bout = A'B. Full subtractor: D = A ⊕ B ⊕ Bin, Bout = A'B + (A ⊕ B)'Bin.
- An XOR gate with a control input M is a controlled inverter: M = 0 passes B, M = 1 inverts B.

4.1.1 Basic

Write the truth table of a half adder and give the expressions for S and C.

YOUR ANSWER

S column e.g. 0110 C column e.g. 0110

S = e.g. AB' + C C = e.g. AB' + C

ANSWER

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

S = A ⊕ B, C = AB

4.1.2 Basic

Write the truth table of a full adder.

YOUR ANSWER

S column e.g. 0110 Cout column e.g. 0110

ANSWER

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

4.1.3 Practice

Show that Cout = AB + Cin(A ⊕ B) is the same as the majority function AB + ACin + BCin.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Both are 1 in exactly the rows Σm(3, 5, 6, 7) (when at least two inputs are 1). K-map of Σm(3, 5, 6, 7): Cout = A·B + A·Cin + B·Cin.

4.1.4 Practice

A 4-bit parallel adder (7483) adds A3A2A1A0 + B3B2B1B0 with carry-in C0. Find the sum and the carry out:

(a) A = 0101, B = 0011, C0 = 0

(b) A = 1001, B = 0111, C0 = 0

(c) A = 1111, B = 0001, C0 = 1

(d) A = 1100, B = 1010, C0 = 0

YOUR ANSWER

(a) S	e.g. 0110	(a) C4	number
(b) S	e.g. 0110	(b) C4	number
(c) S	e.g. 0110	(c) C4	number
(d) S	e.g. 0110	(d) C4	number

ANSWER

(a) S = 1000, C4 = 0 (5 + 3 + 0 = 8)

(b) S = 0000, C4 = 1 (9 + 7 + 0 = 16)

(c) S = 0001, C4 = 1 (15 + 1 + 1 = 17)

(d) S = 0110, C4 = 1 (12 + 10 + 0 = 22)

4.1.5 Basic

Write the truth table of a half subtractor (A – B).

YOUR ANSWER

D column e.g. 0110 Bout column e.g. 0110

ANSWER

A	B	D	Bout
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

4.1.6 Practice

Write the truth table of a full subtractor (A – B – Bin).

YOUR ANSWER

D column e.g. 0110 Bout column e.g. 0110

ANSWER

A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

4.1.7 Challenge

A 4-bit adder–subtractor uses XOR gates on the B inputs and the mode line M as carry-in. Work out A − B for A = 0110, B = 0011 and for A = 0011, B = 0110. Explain the carry out.

YOUR ANSWER

(a) A − B number (b) A − B number

ANSWER

M = 1: B is inverted and 1 is added (2’s complement).

(a) 0110 + 1100 + 1 = 1 0011 → carry out 1 means the result is positive: 0011 = +3.

(b) 0011 + 1001 + 1 = 0 1101 → carry out 0 means the result is negative, in 2’s complement: 1101 = −3.

4.2 Comparators

Key ideas

- 1-bit comparator: (A > B) = AB', (A < B) = A'B, (A = B) = (A ⊕ B)'.
- A multi-bit comparator looks at the most significant bits first; the 7485 compares two 4-bit numbers and can be cascaded.

4.2.1 Basic

Write the truth table and the three output expressions of a 1-bit comparator.

YOUR ANSWER

A > B = e.g. AB' + C A = B = e.g. AB' + C

A < B = e.g. AB' + C

ANSWER

A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

(A > B) = AB', (A = B) = A'B' + AB, (A < B) = A'B

4.2.2 Practice

Write the expression for A = B for two 2-bit numbers A1A0 and B1B0.

YOUR ANSWER

(A = B) = e.g. AB' + C

ANSWER

(A = B) = (A1 ⊙ B1)·(A0 ⊙ B0), where ⊙ is XNOR.

4.2.3 Practice

Give the 7485 outputs (A>B, A=B, A<B) for: (a) A = 1011, B = 1001 (b) A = 0110, B = 0110 (c) A = 0111, B = 1000

YOUR ANSWER

(a) e.g. 0110 (b) e.g. 0110

(c) e.g. 0110

ANSWER

(a) 1, 0, 0 — decided at bit A1

(b) 0, 1, 0 (with the cascade input A = B held at 1)

(c) 0, 0, 1 — decided at the MSB

4.2.4 Challenge

How do you connect two 7485 chips to compare two 8-bit numbers?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

The low chip compares bits 0–3; its three outputs go into the cascade inputs (A>B, A=B, A<B) of the high chip, which compares bits 4–7. The low chip's cascade inputs are tied A=B = 1, A>B = A<B = 0. The high chip's outputs give the final answer.

4.3 Multiplexers and demultiplexers

Key ideas

- A multiplexer (MUX) connects one of 2^n data inputs to its output; n select lines choose which.
- Y of a 4-to-1 MUX = S1'S0'D0 + S1'S0D1 + S1S0'D2 + S1S0D3.
- Any function of n variables can be made with a 2^n-to-1 MUX (or a 2^{n-1}-to-1 MUX with the last variable on the data inputs).
- A demultiplexer (DEMUX) sends one input to one of 2^n outputs.

4.3.1 Basic

How many select lines do these multiplexers need: 4-to-1, 8-to-1, 16-to-1, 32-to-1?

YOUR ANSWER

4-to-1 number 8-to-1 number

16-to-1 number 32-to-1 number

ANSWER

2, 3, 4, 5

4.3.2 Basic

An 8-to-1 MUX has D0…D7 = 1, 0, 0, 1, 1, 1, 0, 1. What is the output for S2S1S0 = 011, 101, 110?

YOUR ANSWER

011 number 101 number

110 number

ANSWER

011 → D3 = 1; 101 → D5 = 1; 110 → D6 = 0

4.3.3 Practice

Implement F(A, B, C) = Σm(1, 2, 6, 7) with an 8-to-1 MUX. What goes on each data input?

YOUR ANSWER

D0 … D7 e.g. 0110

ANSWER

A, B, C go to S2, S1, S0. D0…D7 = 0, 1, 1, 0, 0, 0, 1, 1.

4.3.4 Challenge

Implement the same function F(A, B, C) = Σm(1, 2, 6, 7) with a 4-to-1 MUX, using A and B as select lines.

YOUR ANSWER

D0 = answer D1 = answer

D2 = answer D3 = answer

ANSWER

For each value of AB, compare F with C:

AB	F for C=0, C=1	Data input
00	0, 1	D0 = C
01	1, 0	D1 = C'
10	0, 0	D2 = 0
11	1, 1	D3 = 1

4.3.5 Challenge

Implement F(A, B, C) = Σm(0, 2, 3, 5) with a 4-to-1 MUX (A, B on the select lines).

YOUR ANSWER

D0 = answer D1 = answer

D2 = answer D3 = answer

ANSWER

D0 = C', D1 = 1, D2 = C, D3 = 0

4.3.6 Basic

A 1-to-4 DEMUX has data input D = 1 and select S1S0 = 10. Which output is 1?

YOUR ANSWER

output answer

ANSWER

Y2 = 1; Y0, Y1 and Y3 are 0.

4.3.7 Practice

Write the four output expressions of a 1-to-4 demultiplexer.

YOUR ANSWER

Y0 = e.g. AB' + C Y1 = e.g. AB' + C

Y2 = e.g. AB' + C Y3 = e.g. AB' + C

ANSWER

Y0 = S1'S0'D, Y1 = S1'S0D, Y2 = S1S0'D, Y3 = S1S0D

4.4 Encoders and decoders

Key ideas

- An encoder turns one active input out of 2^n into an n-bit code. A priority encoder outputs the code of the highest active input.
- A decoder does the opposite: an n-bit code makes exactly one of 2^n outputs active. The 74LS138 is a 3-to-8 decoder with active-LOW outputs.
- A decoder plus an OR gate can make any function: OR together the outputs of its minterms.
- The 7447 decodes BCD for a common-anode 7-segment display.

4.4.1 Basic

Write the expressions of a 2-to-4 decoder with inputs A, B.

YOUR ANSWER

Y0 = e.g. AB' + C Y1 = e.g. AB' + C

Y2 = e.g. AB' + C Y3 = e.g. AB' + C

ANSWER

Y0 = A'B', Y1 = A'B, Y2 = AB', Y3 = AB

4.4.2 Basic

An 8-to-3 encoder has input 5 active. What are its outputs A2A1A0?

YOUR ANSWER

A2A1A0 e.g. 0110

ANSWER

101

4.4.3 Practice

A 4-to-2 priority encoder gets inputs D3D2D1D0 = 0110. What is the output? Which input wins?

YOUR ANSWER

output e.g. 0110 winning input answer

ANSWER

D2 has the highest priority among the active inputs, so the output is 10. A valid-output flag V would be 1.

4.4.4 Practice

A 74LS138 has its enables active and CBA = 110. Give all eight outputs (they are active-LOW).

YOUR ANSWER

Y7 … Y0 e.g. 0110

ANSWER

Y6 = 0; all the others (Y0–Y5, Y7) = 1.

4.4.5 Challenge

Build a full adder from a 3-to-8 decoder and two OR gates.

YOUR ANSWER

S: decoder outputs e.g. a, b, c Cout: decoder outputs e.g. a, b, c

ANSWER

Put A, B, Cin on the decoder inputs.

S = OR of outputs 1, 2, 4, 7 (S = Σm(1, 2, 4, 7))

Cout = OR of outputs 3, 5, 6, 7 (Cout = Σm(3, 5, 6, 7))

4.4.6 Practice

Which segments of a 7-segment display light for the digits 2, 4 and 7?

YOUR ANSWER

digit 2 e.g. a, b, c digit 4 e.g. a, b, c

digit 7 e.g. a, b, c

ANSWER

(a) 2: a, b, d, e, g

(b) 4: b, c, f, g

(c) 7: a, b, c

4.4.7 Challenge

Design a 3-bit binary-to-Gray converter. Give the expressions for G2, G1 and G0.

YOUR ANSWER

G2 = e.g. AB' + C G1 = e.g. AB' + C

G0 = e.g. AB' + C

ANSWER

G2 = B2, G1 = B2 ⊕ B1, G0 = B1 ⊕ B0

B2	B1	B0	G2	G1	G0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Chapter 5

Sequential logic and the 555 clock

Synchronous and asynchronous circuits, clock signals and how a 555 timer makes them.

5.1 Sequential circuits and clock signals

Key ideas

- A combinational circuit's output depends only on its present inputs; a sequential circuit also remembers its past (it has memory).
- Synchronous circuits change state only on a clock edge; asynchronous circuits change as soon as an input changes.
- Clock period T = 1/f. Duty cycle = time HIGH ÷ period × 100 %.

5.1.1 Basic

Give two differences between a combinational and a sequential circuit, with one example of each.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Combinational: no memory, output depends only on the present inputs (e.g. adder, decoder).

Sequential: has memory (flip-flops), output depends on the inputs and the stored state (e.g. counter, shift register).

5.1.2 Basic

Why are synchronous circuits used more often than asynchronous ones?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

All flip-flops change together on the clock edge, so the timing is predictable. Asynchronous circuits can have races and short false states because signals arrive at slightly different times.

5.1.3 Basic

A clock signal has a frequency of 25 kHz and is HIGH for 10 µs in each period. Find the period and the duty cycle.

YOUR ANSWER

period e.g. 1.5 ks duty cycle e.g. 1.5 k%

ANSWER

T = 1/25 kHz = 40 µs. Duty cycle = 10/40 × 100 = 25 %.

5.1.4 Practice

A clock has a period of 0.5 ms and a duty cycle of 60 %. How long is it HIGH and how long LOW? What is its frequency?

YOUR ANSWER

HIGH time e.g. 1.5 ks LOW time e.g. 1.5 ks

frequency e.g. 1.5 kHz

ANSWER

HIGH = 0.3 ms, LOW = 0.2 ms, f = 1/0.5 ms = 2 kHz.

5.2 The 555 timer

Key ideas

- Pins: 1 GND, 2 trigger, 3 output, 4 reset, 5 control voltage, 6 threshold, 7 discharge, 8 +Vcc.
- Inside: two comparators (at ⅓ Vcc and ⅔ Vcc), an SR flip-flop, a discharge transistor and an output stage.
- Astable: tH = 0.693(RA + RB)C, tL = 0.693·RB·C, f = 1.44 / ((RA + 2RB)C), duty = (RA + RB)/(RA + 2RB).
- Monostable: one pulse of width T = 1.1·R·C each time pin 2 is pulled below ⅓ Vcc.

5.2.1 Basic

Name the eight pins of the 555 timer and say what pins 2, 4 and 6 do.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

1 GND, 2 trigger, 3 output, 4 reset, 5 control voltage, 6 threshold, 7 discharge, 8 +Vcc.

Pin 2 starts a cycle when it drops below ⅓ Vcc; pin 4 resets the timer when pulled LOW (tie to Vcc if not used); pin 6 ends the HIGH time when it rises above ⅔ Vcc.

5.2.2 Basic

With Vcc = 9 V, at what voltages do the two comparators of a 555 switch?

YOUR ANSWER

lower comparator e.g. 1.5 kV upper comparator e.g. 1.5 kV

ANSWER

⅓ × 9 = 3 V (trigger comparator) and ⅔ × 9 = 6 V (threshold comparator).

5.2.3 Basic

A 555 astable has RA = 1 kΩ, RB = 10 kΩ and C = 10 µF. Find tH, tL, the frequency and the duty cycle.

YOUR ANSWER

tH e.g. 1.5 ks tL e.g. 1.5 ks

f e.g. 1.5 kHz duty cycle e.g. 1.5 k%

ANSWER

tH = 0.693 × (1 kΩ + 10 kΩ) × 10 µF = 76.2 ms

tL = 0.693 × 10 kΩ × 10 µF = 69.3 ms

f = 1 / (tH + tL) = 6.87 Hz

Duty cycle = 52.4 %

5.2.4 Basic

A 555 astable has RA = 4.7 kΩ, RB = 47 kΩ and C = 1 µF. Find tH, tL, the frequency and the duty cycle.

YOUR ANSWER

tH e.g. 1.5 ks tL e.g. 1.5 ks

f e.g. 1.5 kHz duty cycle e.g. 1.5 k%

ANSWER

tH = 0.693 × (4.7 kΩ + 47 kΩ) × 1 µF = 35.8 ms

tL = 0.693 × 47 kΩ × 1 µF = 32.6 ms

f = 1 / (tH + tL) = 14.6 Hz

Duty cycle = 52.4 %

5.2.5 Practice

A 555 astable has RA = 10 kΩ, RB = 10 kΩ and C = 100 nF. Find tH, tL, the frequency and the duty cycle.

YOUR ANSWER

tH e.g. 1.5 ks tL e.g. 1.5 ks

f e.g. 1.5 kHz duty cycle e.g. 1.5 k%

ANSWER

tH = 0.693 × (10 kΩ + 10 kΩ) × 100 nF = 1.39 ms

tL = 0.693 × 10 kΩ × 100 nF = 693 µs

f = 1 / (tH + tL) = 481 Hz

Duty cycle = 66.7 %

5.2.6 Practice

A 555 astable has RA = 2.2 kΩ, RB = 68 kΩ and C = 10 nF. Find tH, tL, the frequency and the duty cycle.

YOUR ANSWER

tH e.g. 1.5 ks tL e.g. 1.5 ks

f e.g. 1.5 kHz duty cycle e.g. 1.5 k%

ANSWER

tH = 0.693 × (2.2 kΩ + 68 kΩ) × 10 nF = 486 µs

tL = 0.693 × 68 kΩ × 10 nF = 471 µs

f = 1 / (tH + tL) = 1.04 kHz

Duty cycle = 50.8 %

5.2.7 Basic

A 555 monostable has R = 100 kΩ and C = 10 µF. How long is the output pulse?

YOUR ANSWER

pulse width e.g. 1.5 ks

ANSWER

T = 1.1 × 100 kΩ × 10 µF = 1.1 s

5.2.8 Basic

A 555 monostable has R = 47 kΩ and C = 100 µF. How long is the output pulse?

YOUR ANSWER

pulse width e.g. 1.5 ks

ANSWER

T = 1.1 × 47 kΩ × 100 µF = 5.17 s

5.2.9 Basic

A 555 monostable has R = 1 MΩ and C = 2.2 µF. How long is the output pulse?

YOUR ANSWER

pulse width e.g. 1.5 ks

ANSWER

T = 1.1 × 1 MΩ × 2.2 µF = 2.42 s

5.2.10 Practice

Design a 555 astable that gives about 1 kHz with C = 100 nF and RA = 1 kΩ. Find RB.

YOUR ANSWER

RB e.g. 1.5 kΩ

ANSWER

1.44 / ((RA + 2RB)·C) = 1000 → RA + 2RB = 1.44 / (1000 × 100 nF) = 14.4 kΩ → RB = (14.4 − 1) / 2 = 6.7 kΩ (use 6.8 kΩ: f ≈ 986 Hz).

5.2.11 Challenge

Why can a basic 555 astable never give a duty cycle of 50 % or less? How can a diode across RB fix this?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

The capacitor charges through RA + RB but discharges only through RB, so tH is always longer than tL. A diode in parallel with RB lets the capacitor charge through RA only: tH = 0.693·RA·C and tL = 0.693·RB·C. With RA = RB the duty cycle is 50 %.

5.2.12 Practice

You need a 10-second delay from a 555 monostable with C = 100 µF. Which resistor do you need?

YOUR ANSWER

R e.g. 1.5 kΩ

ANSWER

R = T / (1.1 C) = 10 / (1.1 × 100 µF) = 90.9 kΩ (use 91 kΩ or a 100 kΩ trimmer).

Chapter 6

Flip-flops

Latches and flip-flops: SR, clocked SR, D, JK, T and master–slave.

6.1 SR latch and clocked SR flip-flop

Key ideas

- A flip-flop stores one bit. Q is the stored bit and Q' its complement.
- SR: S = 1 sets Q to 1, R = 1 resets Q to 0, S = R = 0 holds, S = R = 1 is not allowed.
- A NAND latch has active-LOW inputs (S' and R'); a NOR latch has active-HIGH inputs.
- A clocked (gated) SR only reacts while the clock/enable is active.

6.1.1 Basic

Write the characteristic (truth) table of a clocked SR flip-flop.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

S	R	Q^+	Action
0	0	Q	hold
0	1	0	reset
1	0	1	set
1	1	?	not allowed

6.1.2 Basic

Why is S = R = 1 not allowed for a NOR SR latch?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Both outputs are forced to 0, so Q and Q' are no longer complements. When S and R return to 0 together, the final state depends on which gate is faster: it cannot be predicted.

6.1.3 Practice

A clocked SR flip-flop starts with Q = 0. At each clock pulse the inputs (S, R) are: (1,0), (0,0), (0,1), (0,0), (1,0), (1,1). Give Q after each pulse.

YOUR ANSWER

Q after each pulse answer

ANSWER

Q = 1, 1, 0, 0, 1, ? (the last pulse has S = R = 1: invalid, the state cannot be predicted).

6.1.4 Practice

Draw (describe) an SR latch made of two NAND gates and write its truth table.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Each NAND output feeds one input of the other NAND. The free inputs are S' and R' (active LOW).

S'	R'	Q	Action
1	1	Q	hold
0	1	1	set
1	0	0	reset
0	0	1 (Q = Q' = 1)	not allowed

6.1.5 Challenge

Give the characteristic equation of the SR flip-flop and check it for S = 1, R = 0 and Q = 0.

YOUR ANSWER

Q^+ = e.g. AB' + C

ANSWER

$Q^+ = S + R'Q$ (with SR = 0).

S = 1, R = 0, Q = 0: $Q^+ = 1 + 1 \cdot 0 = 1$ ✓ (set).

6.2 D, JK and T flip-flops

Key ideas

- D: $Q^+ = D$. It removes the forbidden state of the SR by using S = D and R = D'.
- JK: like SR, but J = K = 1 toggles. $Q^+ = JQ' + K'Q$.
- T: T = 1 toggles, T = 0 holds. $Q^+ = T \oplus Q$. A JK with J = K = T is a T flip-flop.
- Edge-triggered flip-flops change only on the rising (↑) or falling (↓) clock edge.

6.2.1 Basic

Write the truth tables of the D, JK and T flip-flops.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

D	Q^+
0	0
1	1

J	K	Q^+
0	0	Q (hold)
0	1	0 (reset)
1	0	1 (set)
1	1	Q' (toggle)

T	Q^+
0	Q (hold)
1	Q' (toggle)

6.2.2 Basic

A rising-edge D flip-flop starts with Q = 0. D at each rising edge is 1, 1, 0, 1, 0, 0, 1. Give Q after each edge.

YOUR ANSWER

Q after each edge e.g. 0110

ANSWER

Q = 1, 1, 0, 1, 0, 0, 1 (Q simply copies D at each edge).

6.2.3 Practice

A JK flip-flop starts with Q = 0. The inputs (J, K) at each clock edge are (1,0), (1,1), (1,1), (0,0), (0,1), (1,1), (1,0). Give Q after each edge.

YOUR ANSWER

Q after each edge e.g. 0110

ANSWER

Q = 1, 0, 1, 1, 0, 1, 1

6.2.4 Practice

A T flip-flop starts with Q = 1. T at each clock edge is 1, 1, 0, 1, 1, 1, 0. Give Q after each edge.

YOUR ANSWER

Q after each edge e.g. 0110

ANSWER

Q = 0, 1, 1, 0, 1, 0, 0

6.2.5 Practice

Show how to turn (a) a JK flip-flop into a D flip-flop (b) a JK into a T flip-flop (c) a D flip-flop into a T flip-flop.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

(a) J = D, K = D' (an inverter between J and K).

(b) Join J and K together: J = K = T.

(c) D = T ⊕ Q (an XOR gate feeding back Q).

6.2.6 Practice

Write the excitation tables of the JK, D and T flip-flops (the inputs needed for each change Q → Q^+).

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Q	Q^+	J	K	D	T
0	0	0	x	0	0
0	1	1	x	1	1
1	0	x	1	0	1
1	1	x	0	1	0

6.2.7 Challenge

Prove the JK characteristic equation $Q^+ = JQ' + K'Q$ from its truth table with a K-map.

YOUR ANSWER

$Q^+ =$ e.g. AB' + C

ANSWER

Q^+ is 1 for (J, K, Q) = 001, 100, 101, 110. On a 3-variable map, 100 and 110 give JQ' and 001 and 101 give K'Q. So $Q^+ = JQ' + K'Q$.

6.3 Master–slave flip-flops, preset and clear

Key ideas

- A master–slave flip-flop is two latches: the master follows the inputs while CLK = 1, the slave copies the master when CLK goes to 0.
- This stops the ‘race-around’ problem of a level-triggered JK with J = K = 1.
- Preset (PRE) and clear (CLR) are asynchronous inputs: they act at once, without waiting for the clock.

6.3.1 Basic

What is the race-around condition and how does the master–slave arrangement stop it?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

In a level-triggered JK with J = K = 1, Q keeps toggling as long as the clock is HIGH, so the final state is unknown. In a master–slave flip-flop the master is cut off from the output while it changes, and the slave changes only once, on the falling edge.

6.3.2 Practice

A 74LS76 JK flip-flop has active-LOW PRE and CLR. What is Q when (a) PRE = 0, CLR = 1 (b) PRE = 1, CLR = 0 (c) PRE = CLR = 1?

YOUR ANSWER

(a) Q number (b) Q number

ANSWER

(a) Q = 1 at once (preset)

(b) Q = 0 at once (clear)

(c) normal clocked JK operation

6.3.3 Practice

Why do we say that preset and clear are asynchronous inputs?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

They change Q immediately, whatever the clock is doing. Normal inputs (J, K, D) only act on a clock edge.

Chapter 7

Frequency division and counters

Dividing frequencies with toggle flip-flops, ripple and synchronous counters, up/down and presettable counters, cascading and counter design.

7.1 Frequency division

Key ideas

- A toggle flip-flop (D with D = Q', or JK with J = K = 1) changes state on every clock edge: its output frequency is f/2.
- n toggle flip-flops in a chain divide by 2^n.

7.1.1 Basic

How do you connect a D flip-flop so that it divides the clock frequency by 2?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Connect Q' back to D. On every active edge Q takes the opposite value, so Q has half the clock frequency (and a 50 % duty cycle).

7.1.2 Basic

A 32.768 kHz watch crystal clock goes through a chain of toggle flip-flops. What is the frequency after (a) 1 (b) 5 (c) 15 flip-flops?

YOUR ANSWER

(a) e.g. 1.5 kHz (b) e.g. 1.5 kHz

(c) e.g. 1.5 kHz

ANSWER

(a) 16.4 kHz

(b) 1.02 kHz

(c) 1 Hz

7.1.3 Practice

How many toggle flip-flops are needed to turn 1 MHz into a frequency below 1 kHz? What frequency do you get?

YOUR ANSWER

flip-flops number frequency e.g. 1.5 kHz

ANSWER

$2^{10} = 1024 \rightarrow$ 10 flip-flops give 1 MHz / 1024 = 977 Hz.

7.1.4 Practice

A 50 Hz mains signal is shaped into a square wave and divided by 2 six times. What is the output frequency and period?

YOUR ANSWER

frequency e.g. 1.5 kHz period e.g. 1.5 ks

ANSWER

50 / 64 = 0.7813 Hz, period = 1.28 s.

7.2 Ripple (asynchronous) counters

Key ideas

- In a ripple counter only the first flip-flop gets the clock; each next flip-flop is clocked by the previous output.
- n flip-flops count 0 to $2^n - 1$ (modulus 2^n).
- For a modulus $N < 2^n$, a NAND gate detects N and clears all flip-flops.
- Each stage adds its delay: the maximum clock frequency is about 1 / (n × tpd).

7.2.1 Basic

How many flip-flops are needed for a counter of modulus (a) 8 (b) 10 (c) 60 (d) 100?

YOUR ANSWER

(a) number (b) number

(c) number (d) number

ANSWER

(a) 3 flip-flops

(b) 4 flip-flops

(c) 6 flip-flops

(d) 7 flip-flops

7.2.2 Basic

A 4-bit binary ripple counter starts at 0000. What does it show after (a) 9 (b) 16 (c) 37 clock pulses?

YOUR ANSWER

(a) e.g. 0110 (b) e.g. 0110

(c) e.g. 0110

ANSWER

(a) 1001 (9 mod 16 = 9)

(b) 0000 (16 mod 16 = 0)

(c) 0101 (37 mod 16 = 5)

7.2.3 Practice

Design a MOD-6 ripple counter. How many flip-flops do you need, and which outputs go to the reset NAND gate?

YOUR ANSWER

flip-flops number outputs to the NAND gate e.g. a, b, c

ANSWER

3 flip-flops. The counter must clear when it reaches 6 = 110, so the NAND gate takes Q2 and Q1; its output drives the CLR inputs. The counter shows 0 to 5.

7.2.4 Practice

Design a MOD-10 ripple counter. How many flip-flops do you need, and which outputs go to the reset NAND gate?

YOUR ANSWER

flip-flops number outputs to the NAND gate e.g. a, b, c

ANSWER

4 flip-flops. The counter must clear when it reaches 10 = 1010, so the NAND gate takes Q3 and Q1; its output drives the CLR inputs. The counter shows 0 to 9.

7.2.5 Practice

Design a MOD-12 ripple counter. How many flip-flops do you need, and which outputs go to the reset NAND gate?

YOUR ANSWER

flip-flops number outputs to the NAND gate e.g. a, b, c

ANSWER

4 flip-flops. The counter must clear when it reaches 12 = 1100, so the NAND gate takes Q3 and Q2; its output drives the CLR inputs. The counter shows 0 to 11.

7.2.6 Practice

Design a MOD-5 ripple counter. How many flip-flops do you need, and which outputs go to the reset NAND gate?

YOUR ANSWER

flip-flops number outputs to the NAND gate e.g. a, b, c

ANSWER

3 flip-flops. The counter must clear when it reaches 5 = 101, so the NAND gate takes Q2 and Q0; its output drives the CLR inputs. The counter shows 0 to 4.

7.2.7 Practice

Each flip-flop in a 4-bit ripple counter has a delay of 25 ns. What is the longest settling time and the highest safe clock frequency?

YOUR ANSWER

settling time e.g. 1.5 ks fmax e.g. 1.5 kHz

ANSWER

4 × 25 ns = 100 ns ⟶ fmax ≈ 1 / 100 ns = 10 MHz.

7.2.8 Challenge

Why does a ripple counter that goes from 0111 to 1000 show false states for a short time? Which false states can appear?

YOUR ANSWER

false states e.g. a, b, c

ANSWER

The flip-flops change one after the other: 0111 ⟶ 0110 ⟶ 0100 ⟶ 0000 ⟶ 1000. The short ‘glitch’ states 0110, 0100 and 0000 can be decoded by mistake. Synchronous counters avoid this.

7.2.9 Practice

The 7490 is a decade counter and the 7493 a 4-bit binary counter. How do you get MOD-10 and MOD-16 from them?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

7490: join QA to input B (CLK B) and clock input A; the reset pins R0(1), R0(2) are held LOW → counts 0–9.

7493: join QA to input B and clock input A; R0(1), R0(2) LOW → counts 0–15.

7.3 Synchronous, up/down and presettable counters

Key ideas

- In a synchronous counter every flip-flop gets the same clock. A flip-flop toggles when all lower bits are 1 (count up).
- fmax = 1 / (tpd of a flip-flop + tpd of the AND gates).
- An up/down counter toggles a stage when all lower bits are 1 (up) or all lower bits are 0 (down).
- A presettable counter (74190–74193) can be loaded with any starting value.

7.3.1 Basic

Give the J and K inputs of each stage of a 4-bit synchronous binary up-counter.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

J0 = K0 = 1, J1 = K1 = Q0, J2 = K2 = Q0Q1, J3 = K3 = Q0Q1Q2.

7.3.2 Practice

In a synchronous counter each flip-flop has tpd = 20 ns and each AND gate 10 ns. What is the highest clock frequency? Compare it with a 4-bit ripple counter made of the same flip-flops.

YOUR ANSWER

synchronous fmax e.g. 1.5 kHz ripple fmax e.g. 1.5 kHz

ANSWER

Synchronous: 1 / (20 + 10) ns = 33.3 MHz.

Ripple: 1 / (4 × 20 ns) = 12.5 MHz. The synchronous counter is faster, and the difference grows with more bits.

7.3.3 Basic

A 4-bit up/down counter is at 0110. Write the next five states when counting (a) up (b) down.

YOUR ANSWER

(a) next five (up) e.g. 0110 (b) next five (down) e.g. 0110

ANSWER

(a) 0110 → 0111 → 1000 → 1001 → 1010 → 1011

(b) 0110 → 0101 → 0100 → 0011 → 0010 → 0001

7.3.4 Practice

A down counter starts at 0011. What does it show after 5 clock pulses?

YOUR ANSWER

after 5 pulses e.g. 0110

ANSWER

1110 (it wraps from 0000 to 1111).

7.3.5 Practice

A 74193 is loaded with 1001 and then counts up. What does it show after 10 pulses? Which output tells you it passed 1111?

YOUR ANSWER

after 10 pulses e.g. 0110 output answer

ANSWER

9 + 10 = 19 → 19 − 16 = 3 → 0011. The carry output (CO) gives a pulse when the count rolls over from 1111 to 0000.

7.3.6 Challenge

Use a presettable 4-bit counter to make a MOD-12 counter that counts from 4 to 15.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Use a counter with synchronous load, e.g. the 74163. Set the data inputs to 0100 and feed the ripple-carry output (RCO = 1 at 15) through an inverter to the active-LOW LOAD input. At the clock edge after 15 the counter loads 4 instead of going to 0: 4, 5, …, 15, 4, … = 12 states.

7.4 Cascading counters

Key ideas

- Counters in cascade multiply their moduli: MOD-10 followed by MOD-6 = MOD-60.
- Two cascaded BCD (decade) counters count 00 to 99.

7.4.1 Basic

What is the total modulus of (a) a MOD-10 and a MOD-6 counter in cascade (b) three MOD-10 counters (c) MOD-16 and MOD-16?

YOUR ANSWER

(a) number (b) number

(c) number

ANSWER

(a) 60 (b) 1000 (c) 256

7.4.2 Practice

A digital clock needs a 1 Hz pulse from a 1 kHz oscillator. Which counters do you cascade?

YOUR ANSWER

decade counters number

ANSWER

Three decade (MOD-10) counters: 1 kHz ÷ 10 ÷ 10 ÷ 10 = 1 Hz.

7.4.3 Practice

How do you build the seconds part (00–59) of a clock from counters?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

A decade counter for the units (0–9) clocks a MOD-6 counter for the tens (0–5). The tens counter is reset when it reaches 6 (0110); its reset pulse also clocks the minutes counter.

7.4.4 Challenge

Two cascaded BCD counters show 3 7. What do they show after 68 more pulses?

YOUR ANSWER

display shows number

ANSWER

37 + 68 = 105 → the display only has two digits, so it shows 05 (and has rolled over once).

7.5 Designing counters with any sequence

Key ideas

- 1. Draw the state diagram. 2. Make the state table (present state → next state).
- 3. Use the excitation table of the chosen flip-flop to find its inputs.
- 4. Simplify each input with a K-map (unused states are don’t-cares). 5. Draw the circuit.

7.5.1 Practice

Design a synchronous MOD-5 counter (0 → 4) with JK flip-flops.

YOUR ANSWER

J2 = e.g. AB' + C K2 = e.g. AB' + C

J1 = e.g. AB' + C K1 = e.g. AB' + C

J0 = e.g. AB' + C K0 = e.g. AB' + C

ANSWER

Q2	Q1	Q0	$Q2^+$	$Q1^+$	$Q0^+$	J2	K2	J1	K1	J0	K0
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
1	0	0	0	0	0	x	1	0	x	0	x

J2 = Q1·Q0

K2 = 1

J1 = Q0

K1 = Q0

J0 = Q2'

K0 = 1

Unused states 101, 110, 111 are don’t-cares.

7.5.2 Challenge

Design a synchronous counter with JK flip-flops for the sequence 000 → 010 → 101 → 110 → 111 → 000 …

YOUR ANSWER

J2 = e.g. AB' + C K2 = e.g. AB' + C

J1 = e.g. AB' + C K1 = e.g. AB' + C

J0 = e.g. AB' + C K0 = e.g. AB' + C

ANSWER

Q2	Q1	Q0	$Q2^+$	$Q1^+$	$Q0^+$	J2	K2	J1	K1	J0	K0
0	0	0	0	1	0	0	x	1	x	0	x
0	1	0	1	0	1	1	x	x	1	1	x
1	0	1	1	1	0	x	0	1	x	x	1
1	1	0	1	1	1	x	0	x	0	1	x
1	1	1	0	0	0	x	1	x	1	x	1

J2 = Q1

K2 = Q1·Q0

J1 = 1

K1 = Q0 + Q2'

J0 = Q1

K0 = 1

Unused states 001, 011, 100 were used as don’t-cares.

7.5.3 Challenge

Design a synchronous counter with D flip-flops that counts only the odd numbers 1, 3, 5, 7.

YOUR ANSWER

D2 = e.g. AB' + C

D1 = e.g. AB' + C

D0 = e.g. AB' + C

ANSWER

Q2	Q1	Q0	Q2⁺	Q1⁺	Q0⁺	D2	D1	D0
0	0	1	0	1	1	0	1	1
0	1	1	1	0	1	1	0	1
1	0	1	1	1	1	1	1	1
1	1	1	0	0	1	0	0	1

D2 = Q2·Q1' + Q2'·Q1

D1 = Q1'

D0 = 1

(Q0 stays at 1 all the time.)

7.5.4 Challenge

Design a synchronous counter with T flip-flops for the sequence 000 → 001 → 011 → 111 → 110 → 100 → 000 …

YOUR ANSWER

T2 = e.g. AB' + C

T1 = e.g. AB' + C

T0 = e.g. AB' + C

ANSWER

Q2	Q1	Q0	$Q2^+$	$Q1^+$	$Q0^+$	T2	T1	T0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	0
0	1	1	1	1	1	1	0	0
1	1	1	1	1	0	0	0	1
1	1	0	1	0	0	0	1	0
1	0	0	0	0	0	1	0	0

T2 = Q2·Q1' + Q2'·Q1

T1 = Q1·Q0' + Q1'·Q0

T0 = Q2·Q0 + Q2'·Q0'

7.5.5 Practice

After designing a counter you must check the unused states. Why?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

At power-on the counter can start in an unused state. You must make sure every unused state leads back into the main sequence (self-starting counter); otherwise it can get stuck in a loop of unused states.

Chapter 8

Shift registers and shift-register counters

SISO, SIPO, PISO, PIPO, bidirectional and universal shift registers, ring counters and Johnson counters.

8.1 Shift registers

Key ideas

- A shift register is a chain of D flip-flops with a common clock. Each pulse moves the data one place.
- SISO: serial in, serial out (a delay line). SIPO: serial in, parallel out. PISO: parallel in, serial out. PIPO: parallel in, parallel out.
- An n-bit SISO needs n clock pulses to load n bits and n more to read them out.
- A universal shift register (74194) can shift left, shift right, load in parallel or hold, chosen by S1S0.

8.1.1 Basic

Name the four kinds of shift register and give one use for SIPO and one for PISO.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

SISO, SIPO, PISO, PIPO. SIPO: turning serial data from a UART into a parallel byte, or driving many LEDs from three wires (74HC595). PISO: reading eight buttons on one input wire (74HC165).

8.1.2 Basic

A 4-bit shift-right register holds 0000. The serial input gets 1, 0, 1, 1 (one bit per clock pulse). Give the contents after each pulse.

YOUR ANSWER

pulse 1 e.g. 0110 pulse 2 e.g. 0110

pulse 3 e.g. 0110 pulse 4 e.g. 0110

ANSWER

Pulse 1: 1000

Pulse 2: 0100

Pulse 3: 1010

Pulse 4: 1101

8.1.3 Practice

A 4-bit shift-right register holds 1010. The serial input gets 0, 1, 1, 0, 0. Give the contents after each pulse.

YOUR ANSWER

pulse 1 e.g. 0110 pulse 2 e.g. 0110

pulse 3 e.g. 0110 pulse 4 e.g. 0110

pulse 5 e.g. 0110

ANSWER

Pulse 1: 0101

Pulse 2: 1010

Pulse 3: 1101

Pulse 4: 0110

Pulse 5: 0011

8.1.4 Basic

How many clock pulses do you need to load a byte into an 8-bit SIPO register? How many to read it out of a SISO register?

YOUR ANSWER

pulses to load number pulses in total for SISO (load + read) num

ANSWER

8 pulses to load. A SISO needs 8 pulses to load and 8 more to take it all out at the serial output.

8.1.5 Practice

A SISO register has 8 stages and is clocked at 1 MHz. How long is a bit delayed between input and output?

YOUR ANSWER

delay e.g. 1.5 ks

ANSWER

8 × 1 µs = 8 µs.

8.1.6 Practice

The 74194 has mode inputs S1S0. What does it do for 00, 01, 10 and 11?

YOUR ANSWER

00 answer 01 answer

10 answer 11 answer

ANSWER

00 hold, 01 shift right, 10 shift left, 11 parallel load.

8.1.7 Challenge

A 4-bit register holds 0011 (= 3). What value does it hold after shifting left once and after shifting right twice (0s shifted in)? What arithmetic does this do?

YOUR ANSWER

after shifting left e.g. 0110 after shifting right twice e.g. 0110

ANSWER

Shift left: 0110 = 6 (× 2). From 0011, shift right twice: 0000 (3 ÷ 4 = 0, the remainder is lost).

Shifting left multiplies by 2; shifting right divides by 2.

8.2 Ring and Johnson counters

Key ideas

- Ring counter: Q of the last stage goes back to D of the first; one 1 circulates. n flip-flops → n states.
- Johnson (twisted-ring) counter: Q' of the last stage goes back; n flip-flops → 2n states.
- Each Johnson state can be decoded with one 2-input AND gate.

8.2.1 Basic

List the states of a 4-bit ring counter that starts at 1000.

YOUR ANSWER

the four states e.g. 0110

ANSWER

1000 → 0100 → 0010 → 0001 → 1000 …

8.2.2 Basic

List the states of a 4-bit Johnson counter that starts at 0000.

YOUR ANSWER

the eight states e.g. 0110

ANSWER

0000 → 1000 → 1100 → 1110 → 1111 → 0111 → 0011 → 0001 → 0000 …

8.2.3 Basic

How many states do you get from (a) a 5-bit ring counter (b) a 5-bit Johnson counter (c) a 5-bit binary counter?

YOUR ANSWER

(a) number (b) number

(c) number

ANSWER

(a) 5 (b) 10 (c) 32

8.2.4 Practice

You need a MOD-10 counter with outputs that never glitch. How many flip-flops does a Johnson counter need, and how many a ring counter?

YOUR ANSWER

Johnson flip-flops number ring flip-flops number

ANSWER

Johnson: 5 flip-flops (2 × 5 = 10). Ring: 10 flip-flops.

8.2.5 Practice

A 3-bit Johnson counter is clocked at 600 Hz. What is the frequency at each output?

YOUR ANSWER

output frequency e.g. 1.5 kHz

ANSWER

The sequence has 6 states and each output is HIGH for 3 of them, so every output is a 600/6 = 100 Hz square wave (50 % duty), shifted in time from the others.

8.2.6 Challenge

Which two outputs must an AND gate use to decode the state 1100 of a 4-bit Johnson counter (sequence 0000, 1000, 1100, …)?

YOUR ANSWER

AND gate = e.g. AB' + C

ANSWER

1100 is the only state where Q1 = 1 and Q2 = 0 (with the bits written Q0Q1Q2Q3), so the gate is Q1·Q2'.

Chapter 9

Logic families

TTL and CMOS, their sub-families and the parameters that describe a logic gate: voltage levels, noise margin, fan-out, delay and power.

9.1 Families and parameters

Key ideas

- TTL (bipolar transistors, 5 V): 74, 74LS, 74S, 74AS, 74ALS, 74F. CMOS (MOSFETs): 4000, 74HC, 74HCT, 74AC, 74LVC.
- Standard TTL: VIH = 2 V, VIL = 0.8 V, VOH = 2.4 V, VOL = 0.4 V.
- Noise margins: NMH = VOH − VIH and NML = VIL − VOL.
- Fan-out = the smaller of IOH/IIH and IOL/IIL.
- Speed–power product = propagation delay × power per gate (in pJ).

9.1.1 Basic

Name three TTL sub-families and three CMOS sub-families.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

TTL: 74 (standard), 74LS (low-power Schottky), 74F (fast). CMOS: 4000 series, 74HC (high-speed CMOS), 74HCT (HC with TTL-level inputs).

9.1.2 Basic

Find the HIGH and LOW noise margins of standard TTL.

YOUR ANSWER

NMH e.g. 1.5 kV NML e.g. 1.5 kV

ANSWER

NMH = 2.4 − 2.0 = 0.4 V

NML = 0.8 − 0.4 = 0.4 V

9.1.3 Practice

A 74HC gate at 5 V has VOH = 4.4 V, VIH = 3.5 V, VIL = 1.5 V, VOL = 0.1 V. Find its noise margins and compare them with TTL.

YOUR ANSWER

NMH e.g. 1.5 kV NML e.g. 1.5 kV

ANSWER

NMH = 4.4 − 3.5 = 0.9 V, NML = 1.5 − 0.1 = 1.4 V. CMOS has much bigger noise margins, so it is less sensitive to noise.

9.1.4 Basic

Standard TTL: IOH = 400 µA, IIH = 40 µA, IOL = 16 mA, IIL = 1.6 mA. What is the fan-out?

YOUR ANSWER

fan-out number

ANSWER

HIGH: 400/40 = 10, LOW: 16/1.6 = 10 → fan-out = 10.

9.1.5 Practice

A 74LS gate has IOH = 400 µA, IOL = 8 mA, and each 74LS input needs IIH = 20 µA, IIL = 0.4 mA. How many 74LS inputs can it drive?

YOUR ANSWER

fan-out number

ANSWER

HIGH: 400/20 = 20, LOW: 8/0.4 = 20 → fan-out = 20.

9.1.6 Challenge

A 74LS output (IOL = 8 mA, IOH = 400 µA) must drive standard 74 inputs (IIL = 1.6 mA, IIH = 40 µA). What is the fan-out?

YOUR ANSWER

fan-out number

ANSWER

LOW: 8/1.6 = 5, HIGH: 400/40 = 10 ⟶ fan-out = 5 (the LOW state limits it).

9.1.7 Practice

A gate has tPLH = 11 ns and tPHL = 7 ns. What is its average propagation delay?

YOUR ANSWER

tpd e.g. 1.5 ks

ANSWER

tpd = (11 + 7) / 2 = 9 ns.

9.1.8 Practice

A TTL chip with four gates draws ICCH = 8 mA and ICCL = 22 mA at 5 V. Find the average power per gate. With tpd = 10 ns, what is the speed–power product?

YOUR ANSWER

power per gate e.g. 1.5 kW speed–power product e.g. 1.5 kJ

ANSWER

Average ICC = (8 + 22)/2 = 15 mA → P = 5 V × 15 mA = 75 mW per chip = 18.75 mW per gate.

Speed–power product = 10 ns × 18.75 mW = 187.5 pJ.

9.1.9 Basic

Give two advantages of CMOS over TTL and one advantage of TTL.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

CMOS: very low power when not switching, wide supply range (3–15 V for the 4000 series), large noise margins, high fan-out.

TTL: less sensitive to static electricity; older TTL was faster than early CMOS.

9.2 Inside the gates: TTL, PMOS, NMOS and CMOS

Key ideas

- Standard TTL NAND: multi-emitter input transistor, phase splitter and totem-pole output.
- Open-collector outputs need a pull-up resistor and can be joined (wired-AND).
- CMOS inverter: a P-MOSFET to Vdd and an N-MOSFET to ground; one of them is always off, so almost no current flows when idle.
- CMOS NAND: two P-MOSFETs in parallel to Vdd and two N-MOSFETs in series to ground.

9.2.1 Basic

Explain how a CMOS inverter works for input 0 and input 1.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Input 0: the P-MOSFET is on and the N-MOSFET is off → the output is connected to Vdd (1).

Input 1: the N-MOSFET is on and the P-MOSFET is off → the output is connected to ground (0).

9.2.2 Practice

In a 2-input CMOS NAND gate, which transistors are on when A = 1 and B = 0? What is the output?

YOUR ANSWER

output number

ANSWER

The P-MOSFET of B is on (it is in parallel to Vdd), the N-MOSFET of A is on but the N-MOSFET of B is off, so the series path to ground is open. Output = 1.

9.2.3 Practice

Why must unused CMOS inputs never be left floating?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

A floating CMOS input picks up noise and static, can sit half-way between 0 and 1, turn both MOSFETs on (extra current, heat) and make the output oscillate. Tie it to Vdd or ground (or to a used input).

9.2.4 Practice

What is a totem-pole output? Why must two totem-pole outputs never be connected together?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Two transistors stacked between Vcc and ground: one pulls HIGH, the other pulls LOW. If one output is HIGH and the other LOW, a large current flows from Vcc to ground through both and can destroy them.

9.2.5 Challenge

Three open-collector outputs are joined to one pull-up resistor. What logic function does the joined wire give?

YOUR ANSWER

function answer

ANSWER

Wired-AND: the wire is HIGH only if all three outputs are HIGH (off); any output that is LOW pulls the wire LOW.

Chapter 10

Introduction to control systems

Transducers, sensors, actuators, and open-loop versus closed-loop control.

10.1 Transducers, sensors and actuators

Key ideas

- A transducer changes one form of energy into another.
- A sensor (input transducer) measures a physical quantity and gives an electrical signal.
- An actuator (output transducer) turns an electrical signal into action: movement, heat, light, sound.

10.1.1 Basic

Sort these into sensors and actuators: thermistor, DC motor, LDR, solenoid valve, microphone, loudspeaker, push-button, relay, ultrasonic sensor, heater.

YOUR ANSWER

sensors e.g. a, b, c actuators e.g. a, b, c

ANSWER

Sensors: thermistor, LDR, microphone, push-button, ultrasonic sensor.

Actuators: DC motor, solenoid valve, loudspeaker, relay, heater.

10.1.2 Basic

Give the energy change made by: (a) a microphone (b) a loudspeaker (c) a solar cell (d) an LED (e) a thermocouple.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

(a) sound → electrical (b) electrical → sound (c) light → electrical (d) electrical → light (e) heat → electrical

10.1.3 Practice

Choose a sensor for each job: (a) detect a person in a room (b) measure the level of water in a tank (c) count bottles on a conveyor (d) measure motor speed.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

(a) PIR motion sensor (b) ultrasonic or float sensor (c) infrared beam (photoelectric) sensor (d) tachogenerator or optical encoder

10.1.4 Practice

What is the difference between an analogue and a digital sensor? Give one example of each.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

An analogue sensor gives a signal that changes continuously with the quantity (thermistor, LDR).

A digital sensor gives only ON/OFF or a coded number (limit switch, digital temperature sensor).

10.2 Open-loop and closed-loop control

Key ideas

- A control system has an input (set point), a controller, an actuator and the process (plant) being controlled.
- Open loop: no measurement of the output; the controller cannot correct errors.
- Closed loop: a sensor feeds the output back; the controller compares it with the set point (error = set point – feedback) and corrects it.

10.2.1 Basic

Say whether each system is open-loop or closed-loop: (a) a toaster with a timer (b) an oven thermostat (c) a washing machine that runs a fixed program (d) a car's cruise control (e) traffic lights on a timer (f) a water tank with a float valve.

YOUR ANSWER

(a) answer (b) answer

(c) answer (d) answer

(e) answer (f) answer

ANSWER

(a) open (b) closed (c) open (d) closed (e) open (f) closed

10.2.2 Basic

Draw (describe) the block diagram of a closed-loop temperature control system and name each block.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Set point → comparator (error detector) → controller → actuator (heater) → process (room/oven) → output temperature. A sensor (thermistor) measures the temperature and feeds it back to the comparator.

10.2.3 Practice

Give two advantages and two disadvantages of closed-loop control compared with open-loop.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Advantages: corrects disturbances automatically; more accurate.

Disadvantages: more complex and costly (needs sensors); can become unstable (oscillate) if badly tuned.

10.2.4 Practice

A room heater has its set point at 22 °C. The sensor reads 19 °C. What is the error and what does the controller do?

YOUR ANSWER

error e.g. 1.5 k°C

ANSWER

Error = 22 − 19 = +3 °C. The controller switches the heater on (or increases its power) until the error is about 0.

10.2.5 Challenge

Why does a simple ON/OFF thermostat use a small dead band (hysteresis), e.g. ON below 21.5 °C and OFF above 22.5 °C?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Without a dead band the heater would switch on and off very quickly around 22 °C, wearing out the relay. The dead band makes the temperature swing a little but keeps the switching slow.

Chapter 11

Sequential function charts (SFC / GRAFCET)

Steps, transitions and actions; action qualifiers; and the rules for serial, selection and parallel branches.

11.1 Steps, transitions and actions

Key ideas

- A step is a stable state of the machine; the initial step has a double square. An active step has a token.
- A transition sits between two steps and has a condition (receptivity). It fires when the step before it is active and its condition is true.
- Actions are written in a rectangle beside the step. Qualifiers: N (while active), S (set, stored), R (reset), L (time limited), D (delayed), P (pulse).

11.1.1 Basic

What do these SFC symbols mean: a double square, a single square, a short horizontal bar on a link, a rectangle beside a step?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Double square: initial step. Single square: normal step. Bar on the link: transition (with its condition written beside it). Rectangle: the action done while the step is active.

11.1.2 Basic

Give the meaning of the action qualifiers N, S, R, L, D and P.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

N: action on while the step is active. S: set (stays on after the step ends). R: reset an action set earlier. L: on for a limited time. D: starts after a delay. P: a single pulse when the step becomes active.

11.1.3 Practice

A press works like this: when the start button S is pressed, the ram goes down (DOWN) until the low limit switch L1; then it goes up (UP) until the high limit switch L0; then it waits for the next start. Write the SFC.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

```
[[0]]   (wait)
  |
  + S·L0
  |
 [1] — DOWN
  |
  + L1
  |
 [2] — UP
  |
  + L0
  |
 back to step 0
```

11.1.4 Practice

A mixer tank: press START → open valve V1 until the level sensor H is reached → run the mixer M for 30 s → open the drain valve V2 until the low sensor B → back to the start. Write the SFC with the timer.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

```
[[0]]
  + START
 [1] – V1
  + H
 [2] – M ; start timer T1 = 30 s
  + T1 done
 [3] – V2
  + B
 back to 0
```

11.1.5 Challenge

Why does an action with the S qualifier need a later step with the R qualifier?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

S keeps the action on even after its step is no longer active. Without a step that resets it (R), the action would stay on for ever.

11.2 Rules of evolution: serial, selection and parallel branches

Key ideas

- Serial: steps follow each other one by one.
- Selection (OR) branch: a single horizontal line; only one of the paths is taken, the one whose transition is true first. Conditions should exclude each other.
- Parallel (AND) branch: a double horizontal line; all paths start together, and the chart only continues when every path has finished (synchronisation).
- A transition fires when (1) all the steps before it are active and (2) its condition is true. Firing deactivates the steps before and activates the steps after it, at the same moment.

11.2.1 Basic

What is the difference between a selection branch and a parallel branch in an SFC? How are they drawn?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Selection: one path out of several is chosen (single horizontal line, a transition on each path).

Parallel: all paths run at the same time (double horizontal line, one common transition before and after).

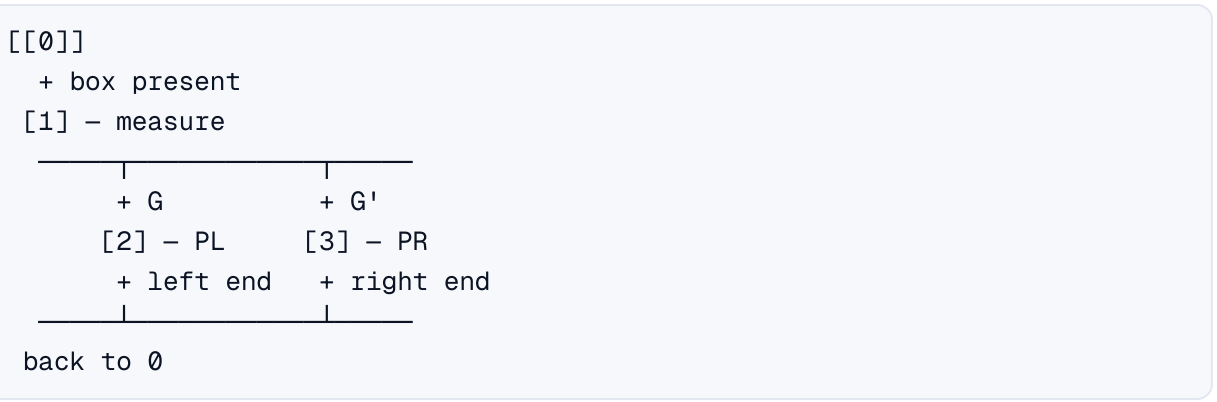
11.2.2 Practice

A sorting machine measures a box: if it is big (G) it pushes it left (PL); if it is small (G') it pushes it right (PR); then it returns to waiting. Draw the SFC with a selection branch.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER



11.2.3 Practice

A machine must drill (step D) and polish (step P) at the same time; when both are finished it unloads the part. Which kind of branch do you use and when does the unload step start?

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

A parallel (AND) branch: after the start transition both steps D and P become active together. The unload step only starts when both branches have reached their last step and the common transition after the double line is true.

11.2.4 Challenge

Steps 4 and 7 are active. The transition after them (with a double line) has the condition ‘ready’ = 1. Step 5 is not active. Which steps are active after the transition fires if it leads to step 8?

YOUR ANSWER

active step(s) after firing e.g. a, b, c

ANSWER

Only step 8. Firing deactivates 4 and 7 and activates 8 at the same moment (step 5 plays no part).

Chapter 12

Programmable logic controllers (PLC)

What a PLC is, how it scans its program, and programming in ladder logic, statement list and function blocks.

12.1 PLC structure and operation

Key ideas

- A PLC is an industrial computer: power supply, CPU, memory, input modules, output modules and a programming device.
- Scan cycle: 1) read all inputs, 2) run the program, 3) update all outputs, 4) housekeeping/communication, then repeat.
- Advantages over relay logic: easy to change, fewer wires, fault finding, timers and counters built in, small size.

12.1.1 Basic

Name the main parts of a PLC and say what each one does.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Power supply (gives the PLC its voltages), CPU (runs the program), memory (stores the program and data), input module (connects switches and sensors), output module (drives lamps, relays, motors through contactors), programming device (PC or handheld to write the program).

12.1.2 Basic

Describe the four steps of the PLC scan cycle.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

1. Read inputs into the input image table. 2. Execute the program rung by rung, top to bottom. 3. Write the results to the outputs. 4. Housekeeping and communication. Then start again.

12.1.3 Practice

A PLC has a scan time of 10 ms. A push-button is pressed for only 4 ms. Can the PLC miss it? Explain.

YOUR ANSWER

can it miss it? (yes/no) answer

ANSWER

Yes. Inputs are only read once per scan; if the 4 ms press happens between two input reads, the PLC never sees it. Fast signals need a high-speed input or an interrupt.

12.1.4 Basic

Give four advantages of a PLC over hard-wired relay control.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

The program can be changed without rewiring; much less wiring and fewer relays; built-in timers and counters; easy fault finding with status lights and the programming software; smaller, more reliable, can communicate with other systems.

12.2 Ladder logic

Key ideas

- The left rail is the power line. A rung is read from left to right; rungs are read from top to bottom.
- —| |— normally open contact (true when the input is 1). —|/|— normally closed contact (true when the input is 0). —( )— output coil.
- Contacts in series = AND. Contacts in parallel = OR.
- A seal-in (latching) contact keeps a motor running after the start button is released.

12.2.1 Basic

Draw the ladder rung for Y = A·B (AND) and for Y = A + B (OR).

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

```
AND:  |--[ A ]--[ B ]--------( Y )--|

OR:   |--+--[ A ]--+-------( Y )--|
         |         |
         +--[ B ]--+
```

12.2.2 Basic

Draw the ladder rung for Y = A' (NOT) and for Y = A·B'.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

```
NOT:   |--[/A ]-----------( Y )--|

A·B':  |--[ A ]--[/B ]----( Y )--|
```

12.2.3 Practice

Draw a motor start/stop circuit with a seal-in contact. START is a normally open button I0.0, STOP a normally closed button I0.1, the motor M is Q0.0.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

```
|--+--[ I0.0 ]--+--[ I0.1 ]--( Q0.0 )--|
   |            |
   +--[ Q0.0 ]--+
```

Pressing START energises Q0.0; its own contact in parallel keeps it on after START is released. STOP (wired NC, so the input is 1 when not pressed) opens the rung and stops the motor.

12.2.4 Practice

Write the ladder rung for Y = (A + B)·C'.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

```
|--+--[ A ]--+--[/C ]--( Y )--|
   |        |
   +--[ B ]--+
```

12.2.5 Practice

Write the ladder rung for an XOR: Y = A·B' + A'·B.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

```
|--+--[ A ]--[/B ]--+--( Y )--|
   |                |
   +--[/A ]--[ B ]--+
```

12.2.6 Practice

Write the statement list (STL) for the start/stop rung with seal-in (Siemens style).

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

```
A(
O   I0.0
O   Q0.0
)
A   I0.1
=   Q0.0
```

12.2.7 Challenge

A tank pump P must run when the low-level switch L is ON and must stop when the high-level switch H turns ON. It must keep running between the two levels. Write the ladder logic.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

```
|--+--[ L ]--+--[/H ]--( P )--|
   |         |
   +--[ P ]--+
```

L starts the pump, the P contact holds it on while the level rises, and H breaks the rung when the tank is full.

12.2.8 Challenge

Two buttons A and B must both be pressed (two-hand safety control) to start a press, and a light curtain C (1 = clear) must be clear. Write the logic and the ladder rung.

YOUR ANSWER

PRESS = e.g. AB' + C

ANSWER

PRESS = A·B·C

```
|--[ A ]--[ B ]--[ C ]--( PRESS )--|
```

12.2.9 Practice

Name the three common PLC programming languages besides ladder logic and explain function block diagrams (FBD) briefly.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

Statement list (STL / instruction list), function block diagram (FBD) and sequential function chart (SFC); structured text (ST) is also common. In FBD the logic is drawn as boxes (AND, OR, timers) with the inputs on the left and the outputs on the right, like a logic circuit.

12.2.10 Practice

An ON-delay timer T1 has a preset of 5 s and is enabled by input I0.2. Describe what output Q0.1 (driven by the timer's done bit) does when I0.2 is switched on for 3 s and then for 8 s.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

3 s: the timer does not reach 5 s, so Q0.1 stays off and the timer resets when I0.2 goes off.

8 s: Q0.1 turns on 5 s after I0.2 goes on, stays on for the remaining 3 s and goes off together with I0.2.

Chapter 13

Mixed revision test

Exam-style questions that mix every chapter. Try them without looking back, then check your answers.

13.1 Part A: short questions

13.1.1 Basic

Convert $(222)_{10}$ to binary, octal, hexadecimal and BCD.

YOUR ANSWER

binary: binary, e.g. 1011 | octal: octal

hex: hex, e.g. 3F | BCD: e.g. 0110

ANSWER

$(11011110)_2 = (336)_8 = (DE)_{16}$; BCD: 0010 0010 0010

13.1.2 Basic

Write −58 in 8-bit 2's complement.

YOUR ANSWER

8 bits e.g. 0110

ANSWER

11000110

13.1.3 Basic

Give the Gray code of 1100_2.

YOUR ANSWER

Gray code e.g. 0110

ANSWER

1010

13.1.4 Basic

Which two gates are called universal gates and why?

YOUR ANSWER

the two gates e.g. a, b, c

ANSWER

NAND and NOR: any other gate (and so any logic circuit) can be built from only NAND gates or only NOR gates.

13.1.5 Basic

How many select lines does a 64-to-1 multiplexer need?

YOUR ANSWER

select lines number

ANSWER

6 ($2^6 = 64$)

13.1.6 Basic

What is the frequency at the output of the 4th flip-flop of a ripple counter clocked at 48 kHz?

YOUR ANSWER

frequency e.g. 1.5 kHz

ANSWER

48 kHz / 2^4 = 3 kHz

13.1.7 Basic

How many states does a 6-bit Johnson counter have?

YOUR ANSWER

states number

ANSWER

12

13.1.8 Basic

Is a street light switched on by a light sensor at dusk an open-loop or a closed-loop system? Explain.

YOUR ANSWER

open or closed loop? answer

ANSWER

Open-loop: the sensor measures the daylight, not the light the lamp produces, so the output is not fed back.

13.2 Part B: longer problems

13.2.1 Practice

Simplify F(A, B, C, D) = Σm(0, 1, 4, 5, 6, 7, 12, 13) with a K-map. Give the SOP and the POS forms.

YOUR ANSWER

SOP: F = e.g. AB' + C POS: F = e.g. AB' + C

ANSWER

SOP: F = A'B + BC' + A'C'

POS: F = (A' + C')(A' + B)(B + C')

13.2.2 Practice

Design a circuit that compares two 2-bit numbers A = A1A0 and B = B1B0 and gives G = 1 when A > B.

YOUR ANSWER

G = e.g. AB' + C

ANSWER

G = 1 for (A1A0B1B0) = Σm(4, 8, 9, 12, 13, 14)

G = A1·B1' + A1·A0·B0' + A0·B1'·B0'

13.2.3 Practice

A 555 astable has RA = 2.2 kΩ, RB = 4.7 kΩ and C = 47 nF. Find the frequency and the duty cycle.

YOUR ANSWER

frequency e.g. 1.5 kHz duty cycle e.g. 1.5 k%

ANSWER

tH = 225 µs, tL = 153 µs, f = 2.65 kHz, duty = 59.5 %

13.2.4 Challenge

Design a synchronous MOD-6 counter (0 → 5) with D flip-flops.

YOUR ANSWER

D2 = e.g. AB' + C

D1 = e.g. AB' + C

D0 = e.g. AB' + C

ANSWER

Q2	Q1	Q0	Q2⁺	Q1⁺	Q0⁺	D2	D1	D0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	0	0	0	0	0	0

D2 = Q2·Q0' + Q1·Q0

D1 = Q1·Q0' + Q2'·Q1'·Q0

D0 = Q0'

13.2.5 Practice

Write the truth table and the simplest expression for a circuit with three inputs whose output is 1 when exactly one input is 1.

YOUR ANSWER

Y = e.g. AB' + C

ANSWER

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Y = AB'C' + A'BC' + A'B'C (no simpler SOP exists)

13.2.6 Practice

A 4-bit shift-left register holds 0110. 1, 1, 0 are shifted in at the right. Give the contents after each pulse.

YOUR ANSWER

pulse 1 e.g. 0110 pulse 2 e.g. 0110

pulse 3 e.g. 0110

ANSWER

Pulse 1: 1101

Pulse 2: 1011

Pulse 3: 0110

13.2.7 Challenge

Draw the SFC and the ladder logic for a garage door: button B opens it (OPEN) until the top switch T; after 10 s it closes (CLOSE) until the bottom switch D; a light beam L (1 = blocked) during closing makes it open again.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

SFC:

```
[[0]]
  + B
 [1] — OPEN
  + T
 [2] — timer 10 s
  + 10 s done
 [3] — CLOSE
  ——┬———————┬——
    + D       + L
  back to 0   back to 1 (open again)
```

Ladder (one rung per step, using memory bits M1…M3 for the steps):

```
|--[ M1 ]-----------------( OPEN )--|
|--[ M3 ]--[/L ]-----------( CLOSE )-|
```

13.2.8 Practice

Compare TTL and CMOS: fill in the table.

YOUR ANSWER

Write your answer here, then compare it with the answer.

ANSWER

	TTL (74LS)	CMOS (74HC)
Transistors	bipolar	MOSFET
Supply	5 V ± 5 %	2–6 V
Idle power	a few mW per gate	almost zero
Noise margin	about 0.4 V	about 1 V or more
Unused inputs	float HIGH (still tie them)	must never float

Notation: A' means NOT A (A with a bar), · or writing letters together means AND, + means OR, ⊕ means XOR and ⊙ means XNOR. Σm lists the minterms (rows where the output is 1), ΠM the maxterms (rows where it is 0).