

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**BỘ MÔN MẠNG MÁY TÍNH
VÀ TRUYỀN THÔNG DỮ LIỆU**

Trần Trung Tín

**CÂU HỎI VÀ BÀI TẬP
MÔN HỌC: TỔ CHỨC MÁY TÍNH**

**TP. HỒ CHÍ MINH
2023**

Mục lục

1	SỐ NHỊ PHÂN VÀ MÃ	1
1.1	Phần câu hỏi trắc nghiệm	1
1.2	Phần bài tập	4
1.3	Các chủ đề sinh viên tự học	8
2	ĐẠI SỐ BOOLE	9
2.1	Phần câu hỏi trắc nghiệm	9
2.2	Phần bài tập	12
2.3	Các chủ đề sinh viên tự học	16
3	TỐI TIỂU MỨC CỒNG	18
3.1	Phần câu hỏi trắc nghiệm	18
3.2	Phần bài tập	22
3.3	Các chủ đề sinh viên tự học	24
4	MẠCH TỔ HỢP	26
4.1	Phần câu hỏi trắc nghiệm	27
4.2	Phần bài tập	29
4.3	Các chủ đề sinh viên tự học	30
5	MẠCH TUẦN TỰ	31
5.1	Phần câu hỏi trắc nghiệm	31
5.2	Phần bài tập	31
5.3	Các chủ đề sinh viên tự học	32
A	Ứng dụng Logism	33
A.1	Vẽ mạch logic từ biểu thức Boole	33
A.2	Vẽ mạch logic từ bảng sự thật	37
A.3	Rút gọn mạch logic và Bìa-K	38
A.4	Tìm biểu thức từ mạch logic	39

Chương 1

SỐ NHỊ PHÂN VÀ MÃ

Hệ nhị phân (hay hệ đếm cơ số hai hoặc mã nhị phân) là một hệ đếm dùng hai ký tự để biểu đạt một giá trị số, bằng tổng số các lũy thừa của 2. Hai ký tự đó thường là 0 và 1; chúng thường được dùng để biểu đạt hai giá trị hiệu điện thế tương ứng (có hiệu điện thế, hoặc hiệu điện thế cao là 1 và không có, hoặc thấp là 0). Do có ưu điểm tính toán đơn giản, dễ dàng thực hiện về mặt vật lý, chẳng hạn như trên các mạch điện tử, hệ nhị phân trở thành một phần kiến tạo căn bản trong các máy tính đương thời.

1.1 Phần câu hỏi trắc nghiệm

Câu hỏi 1. Một vùng nhớ kích thước 4 bytes thì tương đương với bao nhiêu bits?

- a. 4 bits.
- b. 40 bits.
- c. 32 bits.
- d. 4096 bits.

Gợi ý. Bit là đơn vị nhớ cơ bản của máy tính, mỗi bit sẽ lưu trữ giá trị 0 hoặc 1. Cụm 8 bits được gọi là 1 byte, vì thế 4 bytes sẽ tương đương 32 bits.

Câu hỏi 2. Để biểu diễn 43 giá trị cần ít nhất bao nhiêu bits?

- a. 10 bits.
- b. 8 bits.
- c. 6 bits.
- d. 5 bits.

Gợi ý. Để biểu diễn được n trạng thái trong thế giới thực, máy tính cần tối thiểu $\lceil \log_2 n \rceil$ bits. Vậy 6 bits là đáp án.

Câu hỏi 3. Máy tính có thể biểu diễn bao nhiêu trạng thái với 7 bits?

- a. 7 trạng thái.
- b. 14 trạng thái.
- c. 256 trạng thái.
- d. 49 trạng thái.

Gợi ý. Với n bits, máy tính có thể tổ hợp thành 2^n bộ giá trị riêng biệt tương đương với 2^n trạng thái. Vì vậy 256 trạng thái là đáp án. Ví dụ: bộ mã ASCII là bộ mã 7-bit nên biểu diễn được 256 kí tự khác nhau.

Câu hỏi 4. Số C trong hệ số đếm 16 khi chuyển sang hệ thập phân bằng:

- | | |
|------|-------|
| a. 8 | c. 11 |
| b. 9 | d. 12 |

Gợi ý. Trong hệ thập lục phân, ngoài các kí số từ 0 đến 9, kí hiệu A dành cho giá trị 10, B dành cho giá trị 11 và C, D, E, F lần lượt đại diện cho giá trị 12, 13, 14, 15.

Câu hỏi 5. Số thập phân 14.75 tương đương số nhị phân nào?

- | | |
|------------|------------|
| a. 1111.11 | c. 1110.11 |
| b. 1111.10 | d. 1111.01 |

Gợi ý. Lần lượt chuyển đổi $15 = 1110_2$ và $.75 = 0.5 + 0.25 = .11_2$ ta có được đáp án là 1110.11_2 .

Câu hỏi 6. Số 111100110 trong hệ nhị phân được đổi sang hệ bát phân là bao nhiêu?

- | | |
|--------|--------|
| a. 746 | c. 647 |
| b. 486 | d. 345 |

Gợi ý. Để chuyển từ hệ nhị phân sang hệ bát phân, hãy gom cụm 3-bit từ hàng đơn vị và chuyển từng cụm một thành một kí số bát phân. Cụ thể, $111100110 \rightarrow 111,100,110 \rightarrow 7,4,6 \rightarrow 746_8$.

Câu hỏi 7. Số 11010111100110 trong hệ nhị phân được đổi sang hệ thập lục phân là bao nhiêu?

- | | |
|---------|----------|
| a. D792 | c. 13798 |
| b. 35E6 | d. 3E36 |

Gợi ý. Để chuyển từ hệ nhị phân sang hệ thập lục phân, hãy gom cụm 4-bit từ hàng đơn vị và chuyển giá trị thành một kí số thập lục phân. Cụ thể, 11010111100110 được xem như $11,0101,1110,0110$ và đáp án là $35E6_{16}$.

Câu hỏi 8. Số bù 2 của số 1101 1100 0111 là bao nhiêu?

a. 0010 0011 1001

c. 0010 0011 1011

b. 0010 0011 1000

d. 0010 0011 1010

Gợi ý. Để tìm bù 2 của một số nhị phân, hãy lật bit để tìm bù 1 rồi cộng thêm 1 đơn vị sẽ có bù 2. Đáp án là 0010 0011 1001.

Câu hỏi 9. Số có dấu 5 bits lớn nhất có thể biểu diễn theo phương pháp dấu lượng (Sign and Magnitude) là bao nhiêu?

a. 15

c. 31

b. 16

d. 32

Gợi ý. Bit đầu tiên dành cho phần dấu (Sign) và 4 bit còn lại dành cho phần trị (Magnitude) với giá trị cực đại là $+1111 = 15$.

Câu hỏi 10. Số có dấu 5 bits nhỏ nhất có thể biểu diễn theo phương pháp dấu lượng (Sign and Magnitude) là bao nhiêu?

a. -15

c. -31

b. -16

d. -32

Gợi ý. Bit đầu tiên dành cho phần dấu (Sign) và 4 bit còn lại dành cho phần trị (Magnitude) với giá trị cực đại là $-1111 = -15$.

Câu hỏi 11. Đây là dạng chuẩn của phần định trị (mantissa) trong biểu diễn số thực dấu chấm động (Floating Point Number)?

a. 0.01101×2^5

c. 0.1101×2^4

b. 0.01101×2^6

d. 1.011×2^3

Gợi ý. Phần định trị luôn được chuẩn hóa thành 0.1xxxx trong biểu diễn số thực dấu chấm động bằng cách dịch chuyển dấu chấm ra liền trước bit 1 trọng số cao nhất và tăng giảm i trong số mũ 2^i tương ứng để giữ nguyên giá trị của số đó.

Câu hỏi 12. Phép toán nào bị tràn số (overflow)?

a. $0100 + 1001 = 1101$

c. $0011 + 0100 = 0111$

b. $1101 + 1010 = 10111$

d. $1110 + 1010 = 11000$

Gợi ý. Các phép toán đều thực hiện trên số hạng 4-bit. Nếu kết quả cũng là số 4-bit, phép toán không tràn. Nếu xuất hiện bit thứ 5 (hàng vạn) thì tràn số chỉ xảy ra khi (a) hai toán hạng cùng dấu) VÀ (b) bit thứ 4 của kết quả khác với bit thứ 4 của số hạng. Đáp án là b.

1.2 Phần bài tập

Câu hỏi 1. (Câu hỏi 1.1 - sách [1] trang 33) List the octal and hexadecimal numbers from 16 to 32. Using A and B for the last two digits, list the numbers from 8 to 28 in base 12.

Gợi ý.

- Hệ bát phân sử dụng 8 kí số từ 0 đến 7, cho nên hàng đơn vị tăng đến 7 thì sẽ quay về 0 đồng thời hàng chục tăng thêm 1 giá trị, vì thế bắt đầu từ 16 thì tiếp theo là 17, rồi đến 20, 21, 22, 23, 24, 25, 26, 27, 30, 31, 32.
- Hệ thập lục phân ngoài 10 kí số từ 0 đến 9 còn dùng thêm kí tự A đến F, cho nên ta có thể đếm 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, 32.
- Hệ thập nhị phân chỉ sử dụng thêm kí tự A và B, cho nên ta có thể đếm 16, 17, 18, 19, 1A, 1B, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 30, 31, 32.
- Đoạn code Python liệt kê các số từ 16 đến 32 trong các hệ cơ số được cho dưới đây.

```
1 print("\nList Octan numbers: ")
2 var = 0o16                      # start value
3 while (var <= 0o32):            # end value
4     print (oct(var))            # print out the number in octal form
5     var = var + 1              # increment varhex by 1
6
7 print("\nList Hexadecimal numbers:")
8 var = 0x16                      # start value
9 while (var <= 0x32):            # end value
10    print (hex(var))             # print out the number in hex form
11    var = var + 1               # increment varhex by 1
```

Listing 1: Đoạn mã 1

Câu hỏi 2. (Câu hỏi 1.2 - sách [1] trang 33) What is the exact number of bytes in a system that contains: (a) 32K bytes, (b) 64M bytes, and (c) 6.4G bytes?

Gợi ý.

Theo qui ước, 1 Giga bytes = 2^{30} bytes. 1 Mega bytes = 2^{20} bytes. 1 Kilo bytes = 2^{10} bytes. Vì thế:

- (a) 32,768 bytes
- (b) 67,108,864 bytes
- (c) 6,871,947,674 bytes

Câu hỏi 3. (Câu hỏi 1.4 - sách [1] trang 33) What is the largest binary number that can be expressed with 16 bits? What are the equivalent decimal and hexadecimal numbers?

Gợi ý. Số nhị phân đó là 1111111111111111_2 , và có giá trị thập phân là $2^{16} - 1 = 65535$, và được biểu diễn trong hệ thập lục phân là $0xFFFF$.

Câu hỏi 4. (Câu hỏi 1.5 - sách [1] trang 33) Determine the base of the numbers in each case for the following operations to be correct: (a) $14/2 = 5$, (b) $54/4 = 13$, and (c) $24 + 17 = 40$.

Gợi ý. (a) hệ cơ số 6 (b) hệ cơ số 8 (c) hệ cơ số 11

Câu hỏi 5. (Câu hỏi 1.9 - sách [1] trang 34) Express the following numbers in decimal:

(a) $(10110.0101)_2$

(b) $(16.5)_{16}$

(c) $(26.24)_8$

(d) $(DADA.B)_{16}$

(e) $(1010.1101)_2$

Gợi ý.

(a)(b)(c) 22.3125

(d) $(13 \times 16^3) + (10 \times 16^2) + (13 \times 16^1) + (10 \times 16^0) + (11 \times 16^{-1}) = (56026.6875)_{10}$

(e) 10.8125

Câu hỏi 6. (Câu hỏi 1.14 - sách [1] trang 34) Obtain the 1's and 2's complements of the following binary numbers:

(a) 00010000

(d) 10101010

(b) 00000000

(e) 10000101

(c) 11011010

(f) 11111111.

Gợi ý.

Số nhị phân	00010000	00000000	11011010	10101010	10000101	11111111
Biểu diễn bù 1	11101111	11111111	00100101	01010101	01111010	00000000
Biểu diễn bù 2	11110000	100000000	00100110	01010110	01111011	00000001

Trong trường hợp câu (b) thì bit 1 sẽ bị tràn và chuỗi 8-bit còn lại mang giá trị 0, vì thế, số bù 2 của 0 chính là 0.

Câu hỏi 7. (Câu hỏi 1.18 - sách [1] trang 34) Perform subtraction on the given unsigned binary numbers using the 2's complement of the subtrahend. Where the result should be negative, find its 2's complement and affix a minus sign.

$$(a) 10011 - 10010$$

$$(c) 1001 - 110101$$

$$(b) 100010 - 100110$$

$$(d) 101000 - 10101$$

Gợi ý.

(a)

$$\begin{array}{r} 10011 \\ - 10010 \\ \hline 00001 \end{array}$$

(c)

$$\begin{array}{r} 1001 \\ - \overset{\bullet\bullet\bullet\bullet}{110101} \\ \hline \textcolor{red}{1}010100 \end{array}$$

(b)

$$\begin{array}{r} 100010 \\ - \overset{\bullet\bullet\bullet\bullet}{100110} \\ \hline \textcolor{red}{1}111100 \end{array}$$

Bit **1** là bit tràn.

(d)

$$\begin{array}{r} 101000 \\ - \overset{\bullet\bullet\bullet\bullet}{10101} \\ \hline 010011 \end{array}$$

Bit **1** là bit tràn.

Câu hỏi 8. (Câu hỏi 1.20 - sách [1] trang 35) Convert decimal +49 and +29 to binary, using the signed-2's-complement representation and enough digits to accommodate the numbers. Then perform the binary equivalent of (+29) + (-49), (-29) + (+49), and (-29) + (-49). Convert the answers back to decimal and verify that they are correct.

Gợi ý. Giá trị lớn nhất là 49 và nên miền giá trị của phép toán cộng hoặc trừ trên hai số hạng sẽ là $[-98; 98]$, số bit cần thiết cho miền giá trị này là 8 bit.

$$29 = 0001\ 1101_2$$

$$-29 = 11100011_{2s}$$

$$49 = 0011\ 0001_2$$

$$-49 = 11001111_{2s}$$

(a)

$$\begin{array}{r} 29 \\ - 49 \\ \hline -20 \end{array}$$

$$\begin{array}{r} \overset{\bullet\bullet\bullet\bullet}{00011101} \\ + 11001111 \\ \hline 11101100 \end{array}$$

Đáp án 11101100 là một số âm vì vậy được chuyển đổi thành giá trị thập phân là $11101100 \rightarrow 00010011_{1s} \rightarrow 00010100_{1s} \rightarrow -20$.

(b)

$$\begin{array}{r} -29 \\ + 49 \\ \hline 20 \end{array}$$

$$\begin{array}{r} \overset{\bullet}{1}\overset{\bullet}{1}100\overset{\bullet}{0}\overset{\bullet}{1}1 \\ +00110001 \\ \hline \textcolor{red}{1}00010100 \end{array}$$

$\textcolor{red}{1}$ là bit tràn nên đáp án là 00010100, đây là một số dương và có giá trị thập phân là 20.

(c)

$$\begin{array}{r} -29 \\ - 49 \\ \hline -78 \end{array}$$

$$\begin{array}{r} \overset{\bullet}{1}\overset{\bullet}{1}\overset{\bullet}{1}00\overset{\bullet}{0}\overset{\bullet}{1}1 \\ +11001111 \\ \hline \textcolor{red}{1}10110010 \end{array}$$

$\textcolor{red}{1}$ là bit tràn nên đáp án là 10110010, đây là một số âm vì vậy được chuyển đổi thành giá trị thập phân là $10110010 \rightarrow 01001101_{1s} \rightarrow 01001110_{1s} \rightarrow -78$.

Câu hỏi 9. (Câu hỏi 1.23 - sách [1] trang 35) Represent the unsigned decimal numbers 791 and 658 in BCD, and then show the steps necessary to form their sum.

Gợi ý.

791 được biểu diễn bởi $0111\ 1001\ 0001_{BCD}$

658 được biểu diễn bởi $0110\ 0101\ 1000_{BCD}$

$$\begin{array}{r} \overset{\bullet}{0}111 \\ \text{hundreds} + 0110 \\ \hline 1110 \end{array}$$

$$\begin{array}{r} 1001 \\ \text{tens} + 0101 \\ \hline 1110 \end{array}$$

$$\begin{array}{r} 0001 \\ \text{units} + 1000 \\ \hline 1001 \end{array}$$

Hàng đơn vị có kết quả là $1001_2 = 9$. Hàng chục có kết quả là $1110_2 = 14$ thì lấy 4 và mang nhớ 1 sang hàng trăm. Hàng trăm có kết quả là $1110_2 = 14$. Kết quả của phép toán là $791 + 658$ là 1449.

Câu hỏi 10. (Câu hỏi 1.25 - sách [1] trang 35) Represent the decimal number 6,248 in (a) BCD, (b) excess-3 code.

Gợi ý. Mã BCD có được sau khi chuyển mỗi kí số được chuyển thành cụm nhị phân 4-bit. Trong khi mã excess-3 có được từ BCD bằng cách +3 vào mỗi cụm nhị phân.

$$6,248 = 0110\ 0010\ 0100\ 1000_{BCD}$$

$$6,248 = 1001\ 0101\ 0111\ 1011_{\text{excess}-3}$$

Câu hỏi 11. (Câu hỏi 1.29 - sách [1] trang 35) Decode the following ASCII code: 1010011
1110100 1100101 1110110 1100101 0100000 1001010 1101111 1100010
1110011.

Gợi ý. Steve Jobs

1.3 Các chủ đề sinh viên tự học

Biểu diễn số nhị phân và kí tự

Video: [Link](#).

Biểu diễn số thực

Video: [Link](#).

Lịch sử máy tính điện tử

Bộ phim tài liệu đầy đủ nhất của BBC: [Link](#).

Tìm kiếm trên web

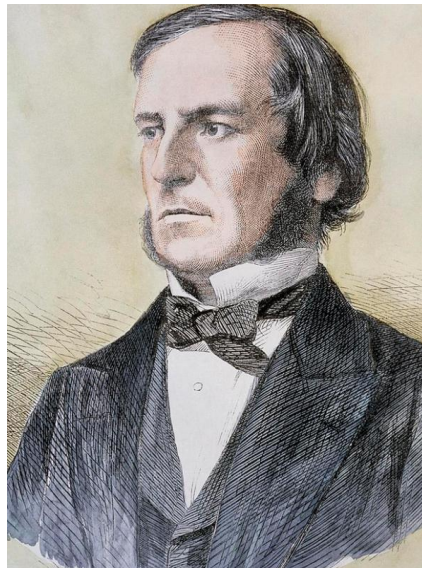
Hãy sử dụng từng từ khóa sau đây để tìm video hay bài đọc về các chủ đề tương ứng.

1. BCD code
2. ASCII
3. Storage register
4. Binary logic
5. BCD addition
6. Binary codes
7. Binary numbers
8. Excess-3 code

Chương 2

ĐẠI SỐ BOOLE

Trong đại số trừu tượng, đại số Boole (hay đại số Boolean) là một cấu trúc đại số có các tính chất cơ bản của cả các phép toán trên tập hợp và các phép toán logic. Cụ thể, các phép toán trên tập hợp được quan tâm là phép giao, phép hợp, phép bù; và các phép toán logic là Và, Hoặc, Không. Đại số Boole được đặt tên theo George Boole (1815–1864), một nhà toán học người Anh. Đại số Boole làm việc với các đại lượng chỉ nhận giá trị Đúng hoặc Sai và có thể thể hiện hệ thống số nhị phân, hoặc các mức điện thế trong mạch điện logic. Do đó đại số Boole có nhiều ứng dụng trong kỹ thuật điện và khoa học máy tính, cũng như trong logic toán học.



Hình 2.1: Giáo sư George Boole (1815 - 1864)

2.1 Phần câu hỏi trắc nghiệm

Câu hỏi 1. Công thức của F là gì theo bảng sự thật dưới đây?

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

a. $F = A.B$

c. $F = A'.B'$

b. $F = A + B$

d. $F = A \oplus B.$

Câu hỏi 2. Thứ tự ưu tiên từ thấp đến cao trong biểu thức boole ra sao?

a. or, and, not.

c. and, not, or.

b. not, and, or.

d. or, not, and.

Gợi ý. Trong các phép toán của đại số Boole, phép or có độ ưu tiên thấp nhất và được thực hiện sau cùng, phép toán and có độ ưu tiên cao hơn và cao nhất là phép toán phủ định (not).

Câu hỏi 3. Định lý De Morgan được phát biểu như thế nào?

a. $(A+B)' = A'.B'$

c. $(A.B)' = A'+B'$

b. $A.(B+C) = A.B+A.C$

d. $A.A' = 0$

Gợi ý. Lựa chọn a. và b. chính là hai thể hiện của định lý De Morgan. Lựa chọn b. là luật phân phối, lựa chọn d. là luật nghịch đảo.

Câu hỏi 4. Biểu thức Boole nào được lượng giá là 0 với tổ hợp (x=0, y=1, z=0)?

a. $(x \text{ OR } (\text{NOT } y)) \text{ AND } z$

c. $x'.y+z$

b. $(x \text{ OR } y) \text{ AND } (\text{NOT } z)$

d. $x.y+z'$

Gợi ý. Thay x=0, y=1, z=0 vào từng công thức và thực hiện lượng giá theo luật đại số Boole, chúng ta sẽ có được $x'.y+z \rightarrow (0)'.1 + 0 = 0$ là đáp án.

Câu hỏi 5. Cho bảng sự thật như trong bảng 2.1.

Đâu là Maxterm của hàm F3 là gì?

a. $x'+y'+z'$

c. $x+y+z$

b. $x.y.z$

d. $x'+y+z'$

x	y	z	F1	F2	F3
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	1	1	0
1	1	1	0	1	0

Bảng 2.1: Bảng sự thật của $F_i(x,y,z)$

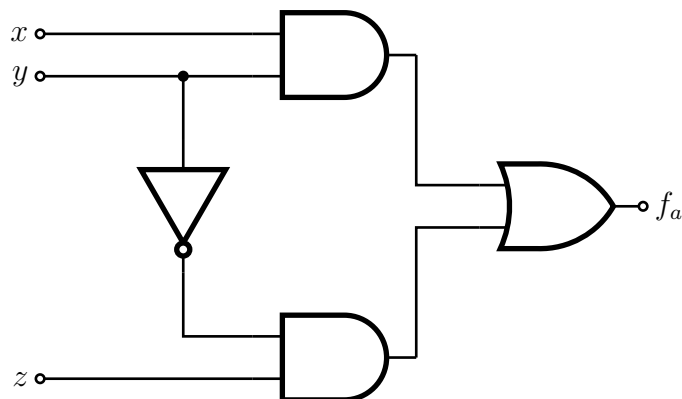
Gợi ý. Lựa chọn (b) là sai vì đó là dạng của minterm. Lựa chọn (d) là M5 ($x'+y+z'$) \rightarrow 101) không thuộc hàm F3. Lựa chọn (a) là M7 và lựa chọn (c) là M0, đều thuộc hàm F3 và là đáp án.

Câu hỏi 6. Cổng logic nào sau đây là cổng đa năng?

- a. AND
- b. NAND
- c. XOR
- d. OR
- e. NOR
- f. NOT

Gợi ý. Các cổng logic NOT, AND và OR là cổng cơ bản của đại số Boole. Cổng XOR là cổng mở rộng. Chỉ có cổng NAND hoặc NOR mới có khả năng thay thế cả 3 cổng cơ bản, nên chúng là cổng đa năng.

Câu hỏi 7. Công thức nào biểu diễn cho mạch logic dưới đây?



- a. $(x+y).(y'+z)$
- b. $x.y + y'.z$
- c. $x.y.z + y'$
- d. $(y' + z).x$

Gợi ý. Có thể thấy mạch được xây dựng bằng dạng SOP và công thức của mạch là $f_a = x.y + y'.z$.

2.2 Phần bài tập

Câu hỏi 1. (Câu hỏi 2.2 - sách [1] trang 69) Simplify the following Boolean expressions to a minimum number of literals:

(a) $x.y + x.y'$

(e) $(a + b + c')(a'.b' + c)$

(b) $(x + y).(x + y')$

(f) $a'.b.c + a.b.c' + a.b.c + a'.b.c'$

(c) $x.y.z + x'.y + x.y.z'$

(d) $(A + B)'.(A' + B')$

Gợi ý.

(a) x

(b) x

(c) y

(d) 0

(e) $a'.b'.c' + b.c + a.c$

(f) b




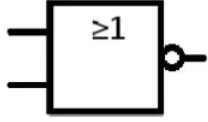
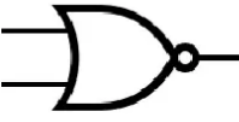
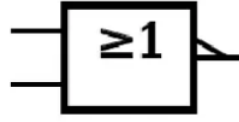
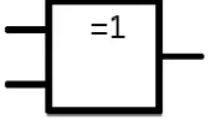

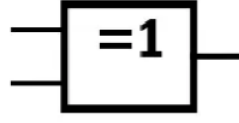
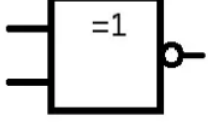
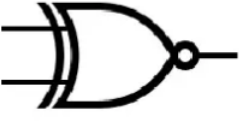
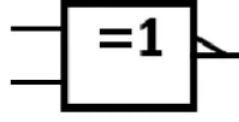
Câu hỏi 2. Hãy cho biết kí hiệu các cổng cơ bản, đang năng và mở rộng khi thể hiện trên các mạch logic.

Gợi ý. Các cổng logic có nhiều tiêu chuẩn kí hiệu khác nhau như ANSI/IEEE hay IEC. Dưới đây là một vài hình thức trình bày thông dụng trong sách giáo khoa.

Operation	IEC 60617-12 Symbol	ANSI/IEEE Distinctive Shape	ANSI/IEEE Rectangular Shape
NOT			
AND			
OR			

Hình 2.2: Các cổng logic cơ bản

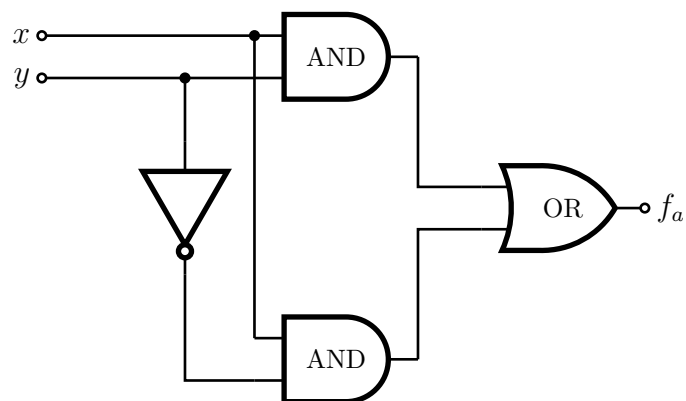
Câu hỏi 3. (Câu hỏi 2.5 - sách [1] trang 69) Draw logic diagrams of the circuits that implement the original and simplified expressions in **Câu hỏi 1.**

Operation	IEC 60617-12 Symbol	ANSI/IEEE Distinctive Shape	ANSI/IEEE Rectangular Shape
NAND			
NOR			
XOR			
XNOR			

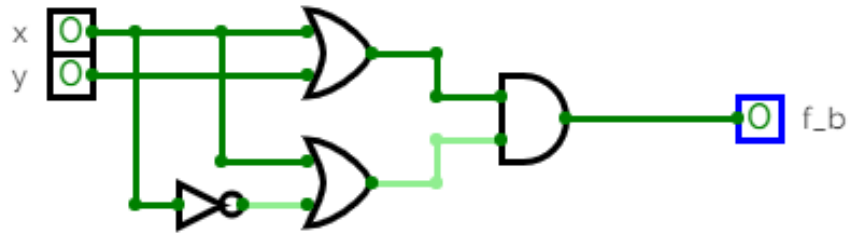
Hình 2.3: Các cổng logic đa dụng và mở rộng

Gợi ý.

(a) (illustrated by circuitikz)

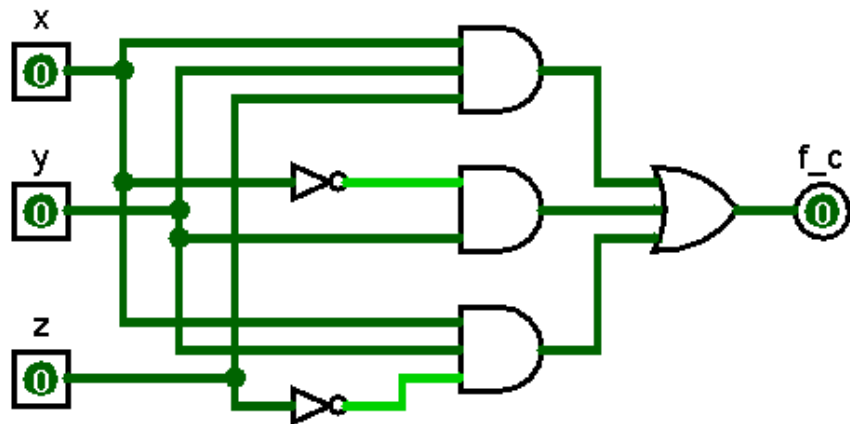


(b) (illustrated by circuitverse)



Hình 2.4: Mạch luận lý của biểu thức $(x + y) \cdot (x + y')$

(c) (illustrated by logism)



Hình 2.5: Mạch luận lý của biểu thức $(x + y) \cdot (x + y')$

(d-f) Sinh viên sử dụng phần mềm Logism để vẽ tự động (Hướng dẫn tại Phụ lục A.1).

Câu hỏi 4. (Câu hỏi 2.11 - sách [1] trang 69) List the truth table of the function:

(a) $F_a = x \cdot y + x \cdot y' + y' \cdot z$

(b) $F_b = b \cdot c + a' \cdot c'$

Gợi ý.

(a)

x	y	z	Fa	Fb
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Câu hỏi 5. (Câu hỏi 2.12 - sách [1] trang 69) We can perform logical operations on strings of bits by considering each pair of corresponding bits separately (called bitwise operation). Given two eight-bit strings A = 10110001 and B = 10101100, evaluate the eight-bit result after the following logical operations: (a)AND (b) OR (c)XOR (d)NOT A

Gợi ý.

(a)

$$\begin{array}{r} 10110001 \\ \&10101100 \\ \hline 10100000 \end{array}$$

(b)

$$\begin{array}{r} 10110001 \\ |10101100 \\ \hline 10111101 \end{array}$$

(c)

$$\begin{array}{r} 10110001 \\ \wedge 10101100 \\ \hline 00011101 \end{array}$$

(d)

$$\begin{array}{r} \neg 10110001 \\ \hline 01001110 \end{array}$$

Câu hỏi 6. (Câu hỏi 2.20 - sách [1] trang 71) Express the complement of the following functions in sum-of-minterms form:

(a) $F1(A,B,C,D) = \sum(2, 4, 7, 10, 12, 14)$

(b) $F2(x,y,z) = \prod(3, 5, 7)$

Gợi ý. (a) F1 có biểu thức được cho dưới dạng Sum of minterm và có thể triển khai như sau:

minterm	4-bit binary	term
m2	0010	$A'.B'.C.D'$
m4	0100	$A'.B.C'.D'$
m7	0111	$A'.B.C.D$
m10	1010	$A.B'.C.D'$
m12	1100	$A.B.C'.D'$
m14	1110	$A.B.C.D'$

$$F1 = A'.B'.C.D' + A'.B.C'.D' + A'.B.C.D + A.B'.C.D' + A.B.C'.D' + A.B.C.D'$$

(b) F2 có biểu thức được cho dưới dạng Product of MAXTERM và có thể triển khai như sau:

MAXTERM	3-bit binary	term
M3	011	$x + y' + z'$
M5	101	$x' + y + z'$
M7	111	$x' + y' + z'$

$$F2 = (x + y' + z').(x' + y + z').(x' + y' + z')$$

2.3 Các chủ đề sinh viên tự học

Giới thiệu đại số Boole

Video: [Link](#).

Giới thiệu về IC số

Bộ phim tài liệu đầy đủ nhất của BBC: [Link](#).

Tìm kiếm trên web

Hãy sử dụng từng từ khóa sau đây để tìm video hay bài đọc về các chủ đề tương ứng.

1. Algebraic field
2. Boolean logic
3. Boolean gates
4. Bipolar transistor
5. Field-effect transistor
6. Emitter-coupled logic

7. TTL logic
8. CMOS logic
9. CMOS process

Chương 3

TỐI TIỂU MỨC CỒNG

Bìa Karnaugh, hay sơ đồ Các-nô, biểu đồ Veitch, là một công cụ để thuận tiện trong việc đơn giản các biểu thức đại số Boole. Bìa Karnaugh độc đáo ở chỗ giữa các ô chỉ có sự thay đổi của một biến mà thôi; hay nói cách khác, các hàng và cột được sắp xếp theo nguyên lý mã Gray. Được Edward W. Veitch sáng tạo vào năm 1952, biểu đồ Veitch được Maurice Karnaugh, một kỹ sư viễn thông làm việc tại Bell Labs, phát triển thêm vào năm 1953. Từ đó bìa Karnaugh còn được gọi là bìa Karnaugh–Veitch.

3.1 Phần câu hỏi trắc nghiệm

Câu hỏi 1. Những phương pháp nào có thể dùng để rút gọn một biểu thức logic?

- Phương pháp biến đổi đại số
- Phương pháp dùng bìa Karnaugh
- Phương pháp dùng đồ thị
- Phương pháp Quine-McCluskey

Gợi ý.

- Phương pháp biến đổi đại số là sử dụng các luật và định lý của đại số Boole để biến đổi một biểu thức trở thành tối thiểu.
- Phương pháp bìa Karnaugh là một công cụ để thuận tiện trong việc đơn giản các biểu thức đại số Boole bằng cách bố trí các minterm lên các ô lân cận nhau.
- Thuật toán Quine–McCluskey (QMC) được sử dụng để cực tiểu hóa các hàm Boolean được Willard V. Quine phát triển vào năm 1952 và được mở rộng bởi Edward J. McCluskey vào năm 1956, có thể xem thêm tại [link](#).
- Không có phương pháp tối thiểu biểu thức nào sử dụng đồ thị.

Câu hỏi 2. Cho sơ đồ Karnaugh sau, biểu thức $F(A,B,C,D)$ sẽ có dạng nào?

		AB			
		00	01	11	10
CD	00			1	1
	01				
	11	1	1	1	
	10		1	1	

- a. $F(A,B,C,D) = A.C'.D' + A.C' + B.C$
- b. $F(A,B,C,D) = A.B'.C'.D' + A.C'.D' + B'.C' + A'.C.D$
- c. $F(A,B,C,D) = C.D.A' + C'.D'.A + C.B$
- d. $F(A,B,C,D) = A.B'.C'.D' + A'.C.D$

Gợi ý. Có 3 implicant trong bìa K, một implicant chứa 4 minterm nên tên gọi sẽ chứa 2 biến số - chính là $C.B$ (lưu ý nhãn gắn cho hai cạnh của bìa K); hai implicant còn lại trong tên gọi sẽ có 3 biến số.

Câu hỏi 3. Cho sơ đồ Karnaugh sau, cách nhóm thành implicant như thế là đúng hay sai?

		X_1X_0			
		00	01	11	10
X_3X_2	00	1			1
	01				
	11				
	10	1			1

- a. Đúng
- b. Sai

Gợi ý. 4 góc của bìa K chính là các minterm m_0, m_2, m_8 và m_{10} - là các lân cận của nhau. Đây chính là implicant $X_2'.X_0'$.

Câu hỏi 4. Cho sơ đồ Karnaugh sau, cách nhóm thành implicant như thế là đúng hay sai?

		X_1X_0			
		00	01	11	10
X_3X_2	00				
	01				
	11	1	1	1	
	10	1	1	1	

a. Đúng

b. Sai

Gợi ý. Một implicant luôn chứa 2^n term. Vì thế 6 là một gom nhóm sai.

Câu hỏi 5. Cho sơ đồ Karnaugh sau, cách nhóm thành implicant như thế là đúng hay

		X_1X_0			
		00	01	11	10
X_3X_2	00				
	01				
	11	1	1	1	
	10	1	1	1	

sai?

a. Đúng

b. Sai

Gợi ý. Các implicant có thể giao nhau nên cách gom nhóm này không vi phạm.

Câu hỏi 6. Cho sơ đồ Karnaugh sau, cách nhóm thành implicant như thế là đúng hay

$$X_1 X_0$$

	00	01	11	10
$X_3 X_2$	00			
01		1		
11			1	
10				

sai?

a. Đúng

b. Sai

Gợi ý. Hai minterm lân cận phải có cạnh chung. Rõ ràng m5 (0101) và m15(1111) không thể thành lập một implicant.

Câu hỏi 7. Cho sơ đồ Karnaugh sau, biểu thức $F(w,x,y,z)$ sẽ có dạng nào?

$$WX$$

	00	01	11	10
YZ	00			
01	1	1		
11			1	1
10			1	1

a. $F(w, x, y, z) = w.x+y.z$

b. $F(w, x, y, z) = w+x+y+z$

c. $F(w, x, y, z) = w.x.y.z$

d. $F(w, x, y, z) = w'.x.y' + w.y$

Gợi ý. Implicant của cụm 4 term $\sum(10,11,14,15)$ có biểu thức là $w.y$ và implicant còn lại có biểu thức là $w'.x.y'$.

3.2 Phần bài tập

Câu hỏi 1. (Câu hỏi 3.1 - sách [1] trang 118) Simplify the following Boolean functions, using three-variable maps:

(a) $F(x, y, z) = \sum(0, 2, 4, 5)$

(c) $F(x, y, z) = \sum(0, 1, 2, 3, 5)$

(b) $F(x, y, z) = \sum(0, 2, 4, 5, 6)$

(d) $F(x, y, z) = \sum(1, 2, 3, 7)$

Gợi ý. (a)

		yz			
		00	01	11	10
x	0	1			1
	1	1	1		

Vì thế $F(x,y,z) = x.y' + x'.z'$

(b)

		yz			
		00	01	11	10
x	0	1			1
	1	1	1		1

Vì thế $F(x,y,z) = x.y' + z'$

(c)

		yz			
		00	01	11	10
x	0	1	1	1	1
	1		1		

(d)

		yz			
		00	01	11	10
x	0		1	1	1
	1			1	

Câu hỏi 2. (Câu hỏi 3.3 - sách [1] trang 118) Simplify the following Boolean expressions, using three-variable maps:

(a) $x.y + x'.y'.z' + x'.y.z'$

(b) $x'.y' + y.z + x.y.z'$

(c) $F(x, y, z) = x'.y + y.z' + y'.z'$

(d) $F(x, y, z) = x'.y.z + x.y'.z' + x.y'.z$

Câu hỏi 3. (Câu hỏi 3.5 - sách [1] trang 119) Simplify the following Boolean functions, using four-variable maps:

(a) $F(w, x, y, z) = \sum(1, 4, 5, 6, 12, 14, 15)$ (b) $F(A, B, C, D) = \sum(2, 3, 6, 7, 12, 13, 14)$ (c) $F(w, x, y, z) = \sum(1, 3, 4, 5, 6, 7, 9, 11, 13, 15)$ (d) $F(A, B, C, D) = \sum(0, 2, 4, 5, 6, 7, 8, 10, 13, 15)$

Câu hỏi 4. (Câu hỏi 3.6 - sách [1] trang 119) Simplify the following Boolean expressions, using four-variable maps:

(a) $A'.B'.C'.D' + A.C'.D' + B'.C.D' + A'.B.C.D + B.C'.D$

(b) $x'.z + w'.x.y' + w.(x.y' + x'.y)$

(c) $A'.B'.C'.D + A.B'.D + A'.B.C' + A.B.C.D + A.B'.C$

(d) $A'.B'.C'.D' + B.C'.D + A'.C'.D + A'.B.C.D + A.C.D'$

Câu hỏi 5. (Câu hỏi 3.9 - sách [1] trang 119) 3.9 Find all the prime implicants for the following Boolean functions, and determine which are essential:

(a) $* F(w, x, y, z) = \sum(0, 2, 4, 5, 6, 7, 8, 10, 13, 15)$

(b) $* F(A, B, C, D) = \sum(0, 2, 3, 5, 7, 8, 10, 11, 14, 15)$

(c) $F(A, B, C, D) = \sum(2, 3, 4, 5, 6, 7, 9, 11, 12, 13)$

(d) $F(w, x, y, z) = \sum(1, 3, 6, 7, 8, 9, 12, 13, 14, 15)$

(e) $F(A, B, C, D) = \sum(0, 1, 2, 5, 7, 8, 9, 10, 13, 15)$

(f) $F(w, x, y, z) = \sum(0, 1, 2, 5, 7, 8, 10, 15)$

Câu hỏi 6. (Câu hỏi 3.23 - sách [1] trang 130) Implement the following Boolean function F , together with the don't-care conditions d , using no more than two NOR gates:

$$F(A, B, C, D) = \sum(2, 4, 10, 12, 14)$$

$$d(A, B, C, D) = \sum(0, 1, 5, 8)$$

Assume that both the normal and complement inputs are available.

Gợi ý.

		X_1X_0			
		00	01	11	10
X_3X_2	00	X	X	0	1
	01	1	X	0	0
	11	1	0	0	1
	10	X	0	0	1

$$\text{Vì thế } F = X'_2.X'_0 + X'_1.X'_0 + X_3.X'_0$$

Chương trình Logism có thể được sử dụng để giải bài toán này, tham khảo phụ lục.

3.3 Các chủ đề sinh viên tự học

Giới thiệu đại số Boole

Video: [Link](#).

Tìm kiếm trên web

Hãy sử dụng từng từ khóa sau đây để tìm video hay bài đọc về các chủ đề tương ứng.

1. Boolean minimization
2. Karnaugh map
3. Wired logic
4. Emitter-coupled logic
5. Open-collector logic
6. Quine McCluskey method
7. Espresso software

8. Consensus theorem
9. Don't-care conditions

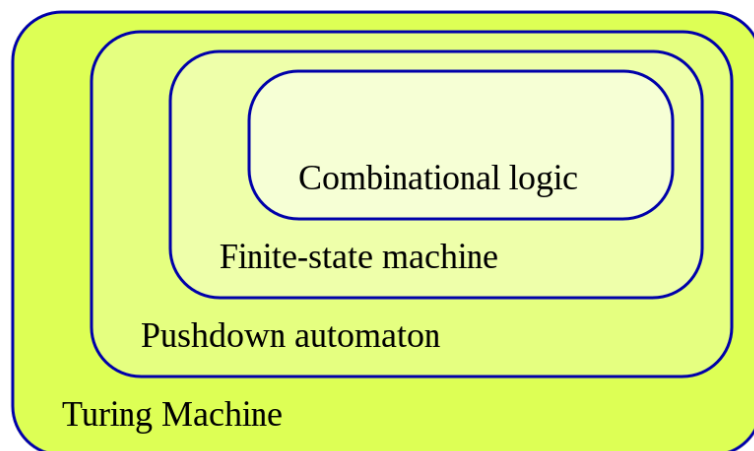
Chương 4

MẠCH TỔ HỢP

Trong lý thuyết automata, logic tổ hợp là một loại logic kỹ thuật số được thực hiện bởi các mạch Boole, trong đó đầu ra chỉ là một hàm thuần túy của đầu vào hiện tại. Điều này trái ngược với logic tuần tự, trong đó đầu ra không chỉ phụ thuộc vào đầu vào hiện tại mà còn phụ thuộc vào lịch sử của đầu vào. Nói cách khác, logic tuần tự có bộ nhớ trong khi logic tổ hợp thì không.

Logic tổ hợp được sử dụng trong các mạch máy tính để thực hiện đại số Boole trên tín hiệu đầu vào và trên dữ liệu được lưu trữ. Các mạch máy tính thực tế thường chứa hỗn hợp logic tổ hợp và logic tuần tự. Ví dụ: một phần của đơn vị logic số học hoặc ALU thực hiện các phép tính toán học được xây dựng bằng logic tổ hợp. Các mạch khác được sử dụng trong máy tính, chẳng hạn như bộ cộng bán phần, bộ cộng toàn phần, các bộ trừ, bộ ghép kênh, bộ tách kênh, bộ mã hóa và bộ giải mã cũng được thực hiện bằng cách sử dụng logic tổ hợp.

Automata theory



Hình 4.1: Vai trò của mạch tổ hợp trong hệ thống máy Turing

4.1 Phần câu hỏi trắc nghiệm

Câu hỏi 1. Mạch tổ hợp (Combinational Circuit) có tính chất nào sau đây?

- a. trạng thái đầu ra chỉ phụ thuộc vào trạng thái các đầu vào.
- b. trạng thái đầu ra không phụ thuộc vào trạng thái các đầu vào.
- c. có trạng bị bộ nhớ để ghi lại trạng thái quá khứ.
- d. trạng thái đầu ra phụ thuộc vào ngữ cảnh và trạng thái các đầu vào.

Gợi ý. Mạch tổ hợp không có trạng bị bộ nhớ, không phụ thuộc vào ngữ cảnh và độc lập với thời gian. Các trạng thái ở đầu ra chỉ được quyết định bởi tổ hợp trạng thái của các đầu vào.

Câu hỏi 2. Mạch con nào sau đây chỉ có 1 đầu ra?

- a. Bộ mã hóa
- b. Bộ ghép kênh (Multiplexer)
- c. Các mạch Flip-flop
- d. Bộ giải mã

Gợi ý. Các mạch Flip-flop luôn cung cấp 2 đầu ra mang tín hiệu bù nhau. Các bộ mã hóa có nhiều đầu ra là một cụm bit tín hiệu. Bộ giải mã có nhiều đầu ra tương ứng với kết quả giải mã từ cụm bit đầu vào. Chỉ có bộ ghép kênh là có một đầu ra duy nhất, là đường tín hiệu được chọn từ n cổng vào và được chọn bởi cụm tín hiệu SELECT.

Câu hỏi 3. Bất cặp mạch con và chức năng của nó.

- Các mạch con:

- | | |
|-------------------------|--------------------------|
| 1. Bộ mã hóa ENCODER | 5. Bộ cộng 4-BIT ADDER |
| 2. Bộ ghép kênh MUX | 6. Bộ so sánh COMPARATOR |
| 3. Bộ cộng toàn phần FA | 7. Bộ tách kênh DeMUX |
| 4. Bộ giải mã DECODER | 8. Bộ cộng bán phần HA |

- Các chức năng:

- a. cộng hai số nhị phân 4-bit
- b. so sánh hai số nhị phân 4-bit
- c. giải mã thông tin $n - bit$ từ đầu vào đến 1 trong 2^n đầu ra
- d. chọn 1 trong số các đầu ra để kết nối với đầu vào thông qua tín hiệu điều khiển SELECTION
- e. chọn 1 trong số các đầu vào để kết nối với đầu ra thông qua tín hiệu điều khiển SELECTION

- f. mã hóa thông tin từ 1 trong 2^n đầu vào thành cụm $n - bit$ ở đầu ra.
- g. cộng một vị trí trọng số trong phép cộng 2 số nhị phân có xem xét số nhớ tràn vào.
- h. cộng hai bit ở hàng đơn vị trong phép cộng nhị phân

Gợi ý.

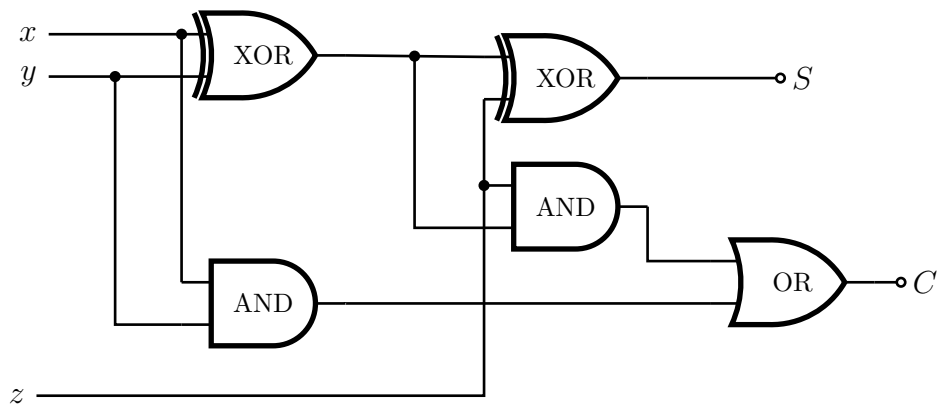
- 1 - f
- 4 - c
- 7 - d
- 2 - e
- 5 - a
- 8 - h
- 3 - g
- 6 - b

Câu hỏi 4. Đâu là khác biệt giữa bộ cộng toàn phần FULL ADDER và bộ cộng bán phần HALF ADDER?

- a. FA có 3 đầu ra trong khi HA chỉ có 2 đầu ra
- b. FA có 3 đầu vào trong khi HA chỉ có 2 đầu vào
- c. FA cộng số nhị phân trong khi HA cộng số thập phân
- d. FA được trang bị bộ nhớ số tràn trong khi HA thì không

Gợi ý. Đáp án là câu (b). Mạch HA có thêm một đầu vào là bit tràn vào (Carry in). Cả hai mạch đều có 2 đầu ra là kết quả S và số tràn ra C.

Câu hỏi 5. Cho mạch logic như hình bên dưới. Giả sử rằng các tín hiệu đầu vào đã ổn định tại thời điểm 0 giây và mỗi cổng logic có độ trễ là t , tín hiệu ở mỗi đầu ra có độ trễ là bao nhiêu?



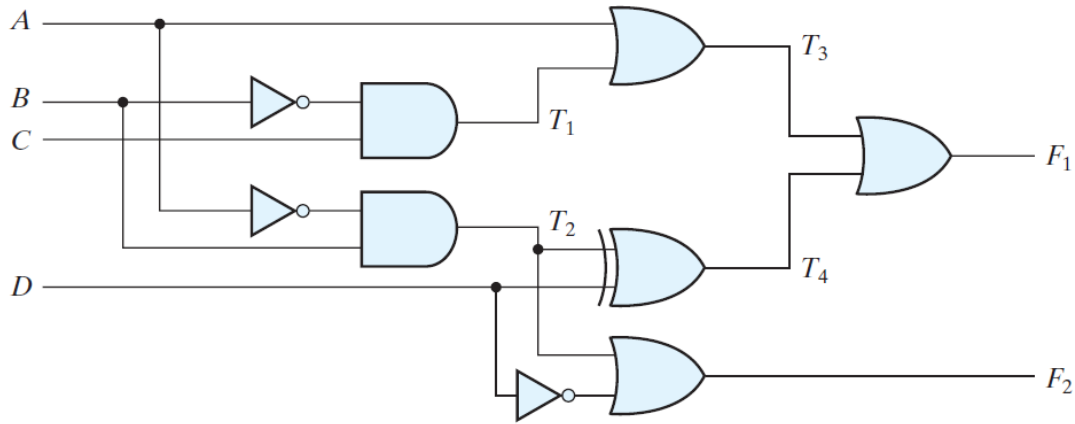
Gợi ý.

$$Delay_S = t + \max(x \oplus y, z) = t + t = 2 * t$$

$$Delay_C = t + \max(x.y, (x \oplus y).z) = t + \max(t, t + \max(x \oplus y, z)) = 3 * t$$

4.2 Phần bài tập

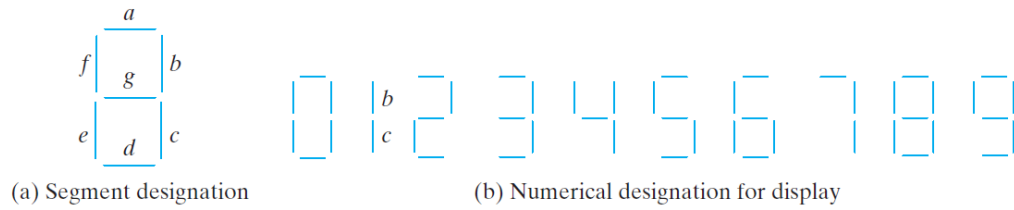
Câu hỏi 1. (Câu hỏi 4.1 - sách [1] trang 182) Consider the combinational circuit shown in Fig. 4.2



Hình 4.2: Mạch logic cho **Câu hỏi 1.**

- Derive the Boolean expressions for T_1 through T_4 . Evaluate the outputs F_1 and F_2 as a function of the four inputs.
- List the truth table with 16 binary combinations of the four input variables. Then list the binary values for T_1 through T_4 and outputs F_1 and F_2 in the table.
- Plot the output Boolean functions obtained in part (b) on maps and show that the simplified Boolean expressions are equivalent to the ones obtained in part (a).

Câu hỏi 2. (Câu hỏi 4.9 - sách [1] trang 183) An ABCD-to-seven-segment decoder is a combinational circuit that converts a decimal digit in BCD to an appropriate code for the election of segments in an indicator used to display the decimal digit in a familiar form. The seven outputs of the decoder (a, b, c, d, e, f, g) select the corresponding segments in the display, as shown in Fig. 4.3(a). The numeric display chosen to represent the decimal digit is shown in Fig. 4.3(a). Using a truth table and Karnaugh maps, design the BCD-to-seven-segment decoder using a minimum number of gates. The six invalid combinations should result in a blank display.



Hình 4.3: Mạch logic cho **Câu hỏi 2**.

4.3 Các chủ đề sinh viên tự học

Thiết kế bộ giải mã BCD to LED 7 đoạn

Video: [Link](#).

Thiết kế bộ cộng 4-bit

Video: [Link](#).

Tìm kiếm trên web

Hãy sử dụng từng từ khóa sau đây để tìm video hay bài đọc về các chủ đề tương ứng.

1. Boolean equation
2. Combinational logic
3. Truth table
4. Exclusive-OR
5. Comparator
6. Multiplexer
7. Decoder
8. Priority encoder
9. Three-state inverter
10. Three-state buffer

Chương 5

MẠCH TUẦN TỰ

Mạch tuần tự là mạch có trạng thái đầu ra không những phụ thuộc vào tổ hợp các đầu vào mà còn phụ thuộc trạng thái đầu ra trước đó, ta nói mạch tuần tự có tính nhớ. Mạch tuần tự vận hành dưới tác động của xung đồng hồ và được chia làm 2 loại: Đồng bộ và Không đồng bộ. Ở mạch đồng bộ, các phần tử của mạch chịu tác động đồng thời của xung đồng hồ.

5.1 Phần câu hỏi trắc nghiệm

Câu hỏi 1. Linh kiện nào khác nhóm với những linh kiện còn lại?

- | | |
|----------------|------------------|
| a. D flip-flop | c. J-K flip-flop |
| b. S-R latch | d. Decoder |

Gợi ý. Decoder là mạch tổ hợp còn các linh kiện khác là mạch tuần tự.

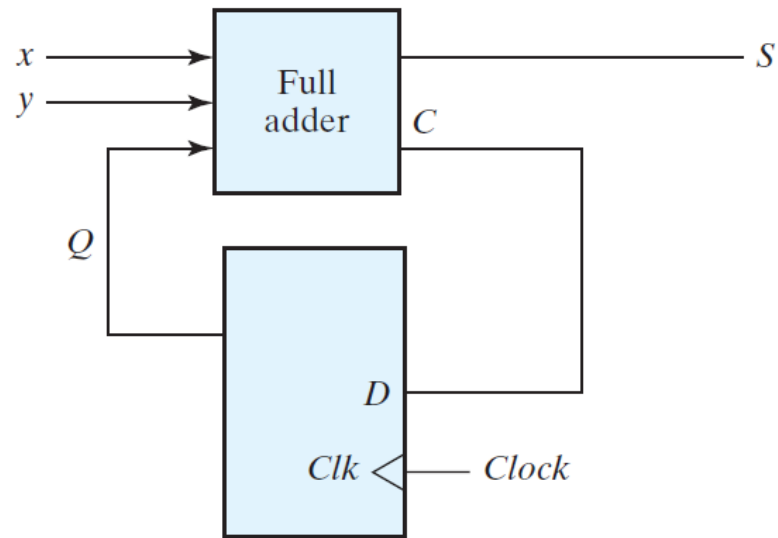
Câu hỏi 2. Phần tử nào sau đây cần cấp xung đồng hồ để hoạt động?

- | | |
|----------------|------------|
| a. Multiplexer | c. Counter |
| b. Full Adder | d. Decoder |

Gợi ý. Counter.

5.2 Phần bài tập

Câu hỏi 1. (Câu hỏi 5.7 - sách [1] trang 246) A sequential circuit has one flip-flop Q, two inputs x and y, and one output S. It consists of a full-adder circuit connected to a D flip-flop, as shown in Fig. 5.1. Derive the state table and state diagram of the sequential circuit.



Hình 5.1: Mạch logic cho **Câu hỏi 1**.

5.3 Các chủ đề sinh viên tự học

Giới thiệu về bộ đếm số

Video: [Link](#).

Tìm kiếm trên web

Hãy sử dụng từng từ khóa sau đây để tìm video hay bài đọc về các chủ đề tương ứng.

1. Finite State Machine
2. Synchronous state machine
3. Asynchronous state machine
4. D-type flip-flop
5. Toggle flip-flop
6. J-K type flip-flop
7. Binary counter
8. State diagram
9. Mealy state machine
10. Moore state machine
11. One-hot/cold codes

Phụ lục A

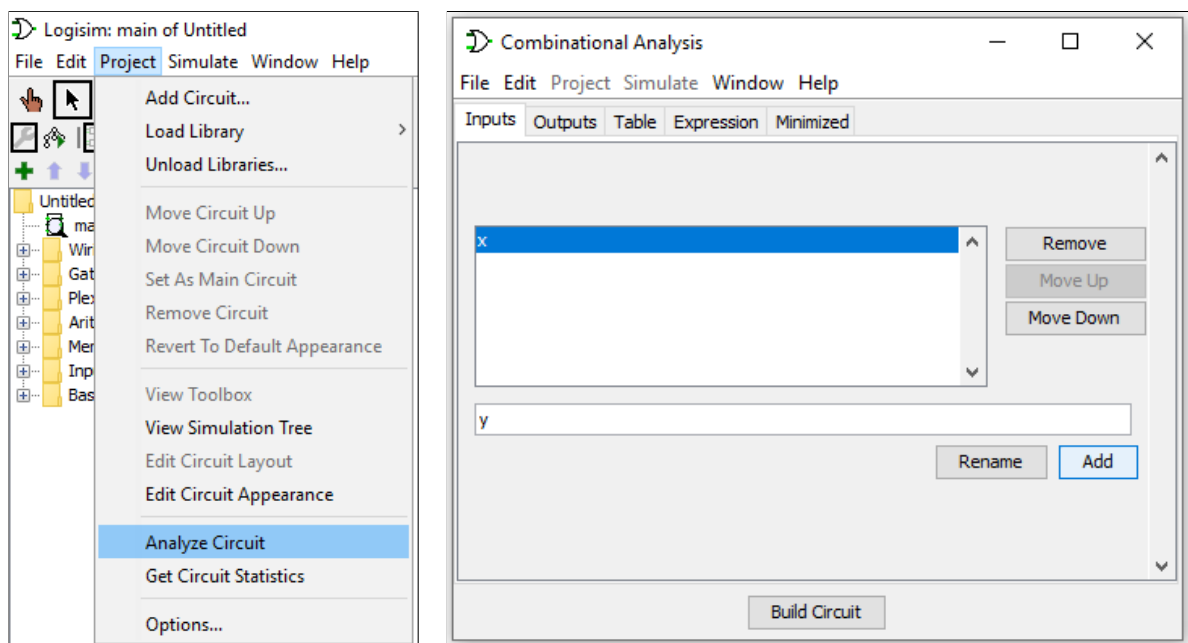
Ứng dụng Logism

A.1 Vẽ mạch logic từ biểu thức Boole

Bước 1: Từ màn hình làm việc, chọn **Project \ Analyze Circuit**.

Bước 2: Nhập các biến số vào (và sắp xếp thứ tự nếu cần thiết) tại thẻ **Inputs** như hình A.1.

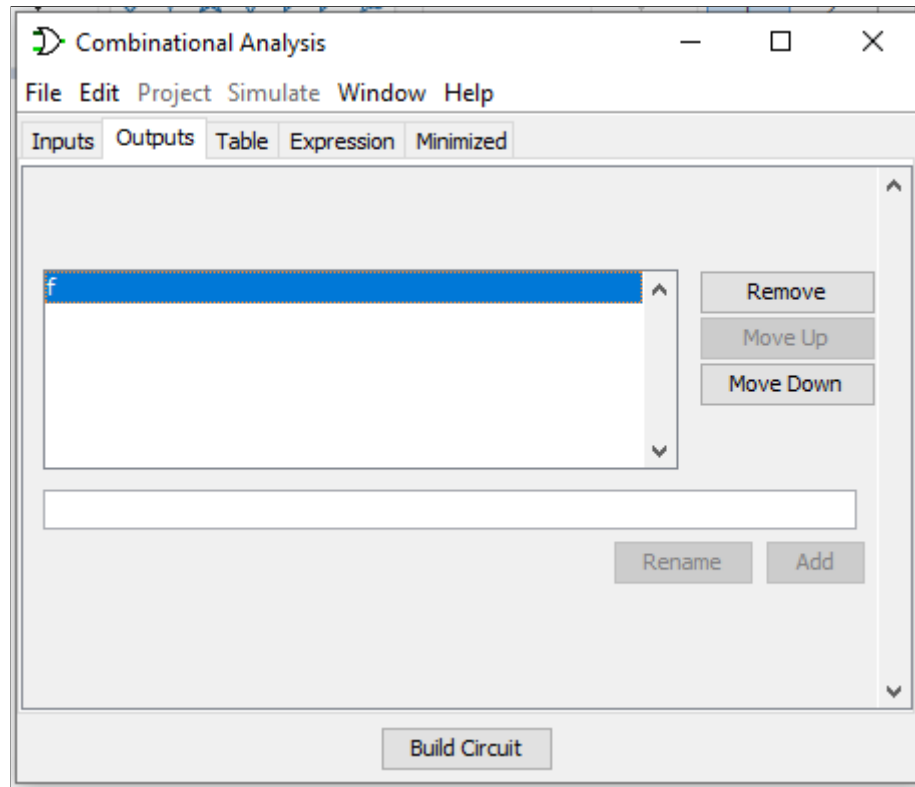
Bước 3: Tương tự nhập các đầu ra tại thẻ **Outputs** như hình A.2.



(a) Lựa chọn "Phân tích mạch"

(b) Cửa sổ "Phân tích mạch" với thẻ "Đầu vào"

Hình A.1: Giao diện chức năng Phân tích mạch của Logism

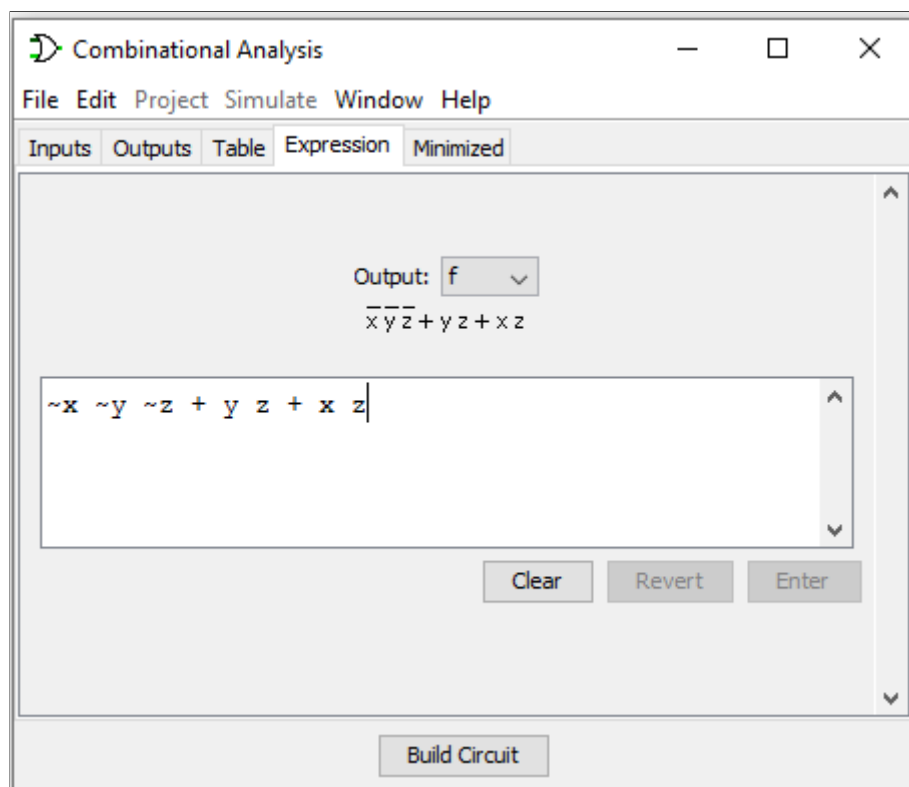


Hình A.2: Chức năng "Phân tích mạch logic"

Bước 4: Chuyển đến thẻ **Expression** và nhập biểu thức cho từng đầu ra như hình A.3 với cách viết các kí hiệu được quy ước tại bảng A.1.

Biểu thức	Cách viết trong Logism	Ghi chú
0 / 1	0 / 1	Hằng số false / true
NOT X	$\sim X$ hoặc !X hoặc X'	Phủ định X'
X AND Y	X Y hoặc X & Y	Khoảng trắng giữa biến số
X XOR Y	$X \wedge Y$	Phép toán XOR
X OR Y	X + Y hoặc X Y	Phép toán HOẶC
Ví dụ 1	$X \sim Y + \sim X Y$	$X'.Y + X.Y'$
Ví dụ 2	$(X \wedge Y) (\sim Y Z + 1)$	$(X \oplus Y). (Y.Z' + 1)$
Ví dụ 3	$a' (b + c)$ $!a \& (b c)$ NOT a AND (b OR c)	$a'.(b+c)$

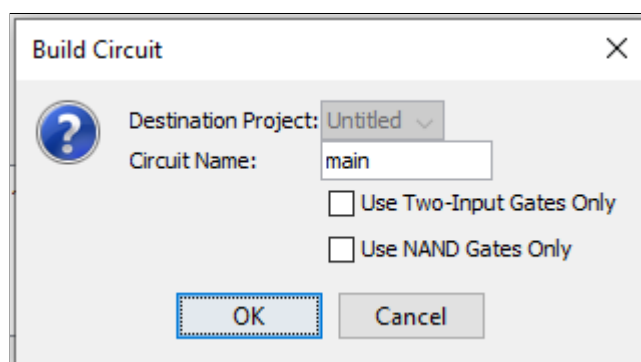
Bảng A.1: Cách viết biểu thức trong Logism



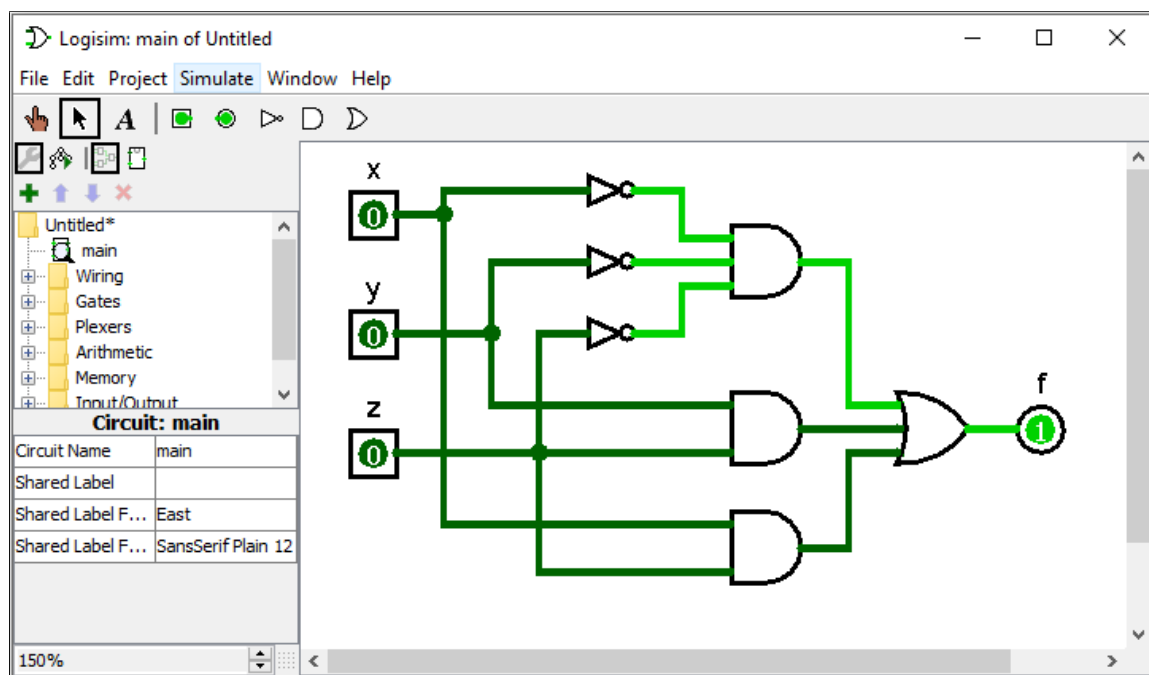
Hình A.3: Chức năng "Nhập vào biểu thức cho từng đầu ra"

Bước 5: Click chuột vào nút **Build Circuit**, tại cửa sổ **Build Circuit** như hình A.4, chúng ta có thể chọn lựa "*Chỉ sử dụng cổng có 2 đầu vào*" và/hoặc "*Chỉ sử dụng cổng đa dụng NAND*".

Bước 6: Mạch kết quả sẽ được thể hiện trong cửa sổ làm việc chính như hình A.5.



Hình A.4: Mạch logic được vẽ từ biểu thức được nhập vào



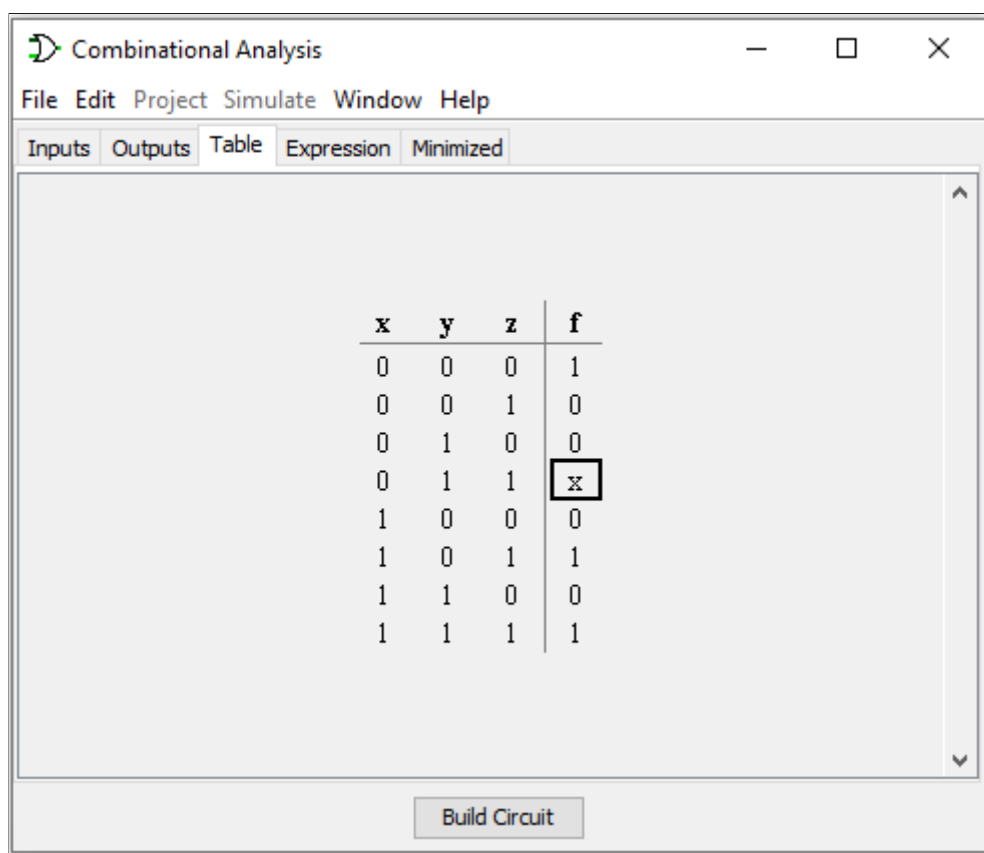
Hình A.5: Chức năng "Nhập vào biểu thức cho từng đầu ra"

A.2 Vẽ mạch logic từ bảng sự thật

Bước 1: Thực hiện như Bước 1 đến Bước 3 của phụ lục A.1.

Bước 2: Chuyển đến thẻ **Table** và click chuột lên từng bit của cột đầu ra để xác định giá trị mong muốn như hình A.6. Trong kí hiệu: 0 = false; 1 = true; X = don't care.

Bước 3: Click chuột vào nút **Build Circuit** để có kết quả mạch logic trong cửa sổ làm việc chính như hình A.5.



Hình A.6: Mạch logic được vẽ từ biểu thức được nhập vào

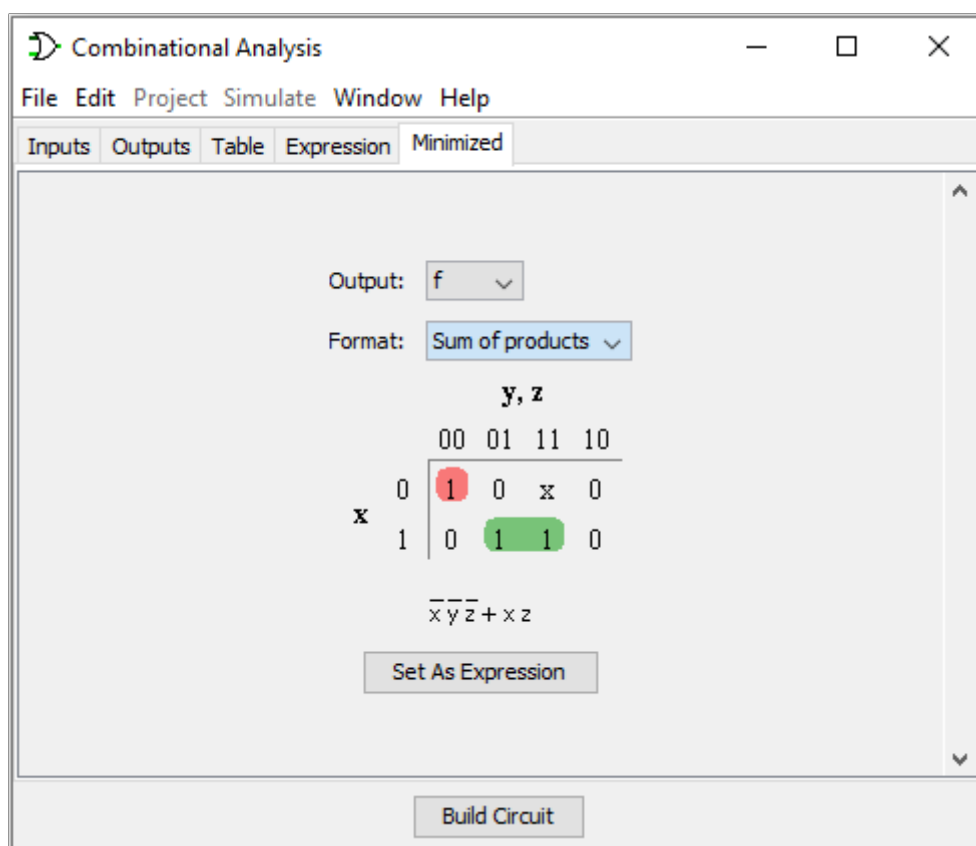
A.3 Rút gọn mạch logic và Bìa-K

Bước 1: Thực hiện như Bước 1 đến Bước 3 của phụ lục A.1.

Bước 2: Chuyển đến thẻ **Minimized** và chọn đầu ra cùng với mục tiêu rút gọn là SOP hay POS.

Bước 3: Tùy chọn **Set As Expression** sẽ chọn biểu thức rút gọn thay thế vào biểu thức ban đầu.

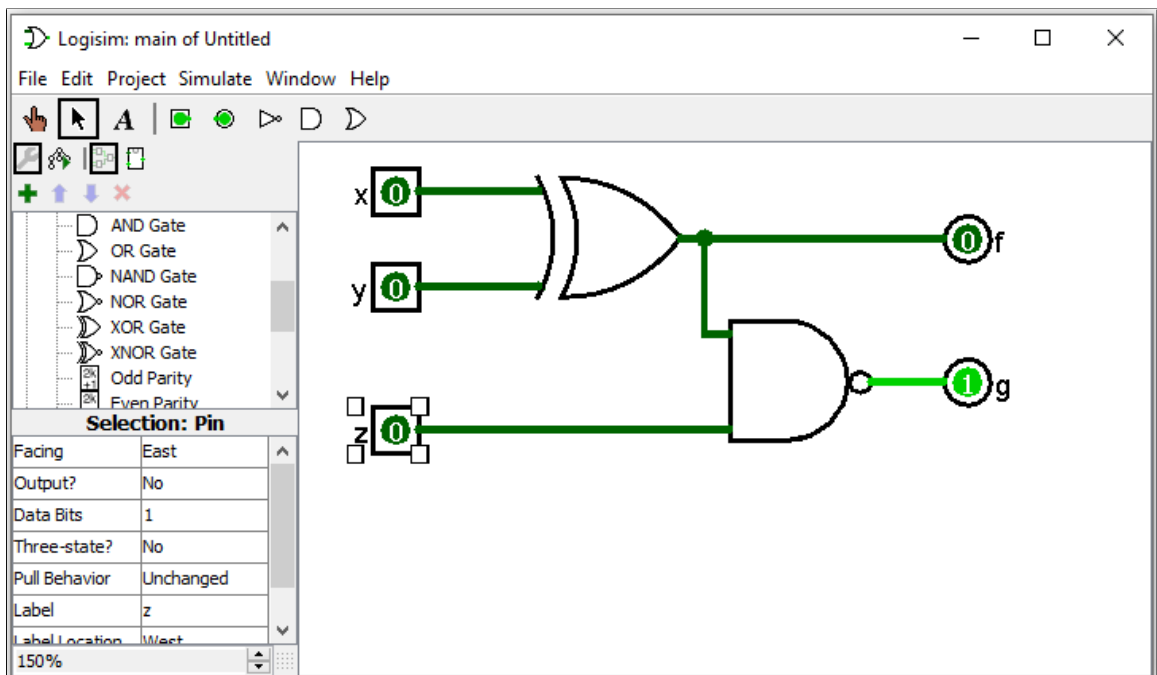
Bước 4: Bìa-K được thể hiện như hình A.7.



Hình A.7: Bìa-K cùng với các cụm được chọn

A.4 Tìm biểu thức từ mạch logic

- Bước 1:** Tại màn hình làm việc chính, bắt đầu vẽ mạch logic cần phân tích như hình A.8.
- Bước 2:** Mỗi đầu vào được định nghĩa bằng một Pin input, được thêm vào bằng nút hình vuông **Add pin** (nút thứ 4 trên thanh công cụ Toolbar). Có thể thêm nhanh bằng tổ hợp phím **Ctrl + 4**.
- Bước 3:** Mỗi đầu ra được định nghĩa bằng một Pin output, được thêm vào bằng nút hình tròn **Add pin** (nút thứ 5 trên thanh công cụ Toolbar). Có thể thêm nhanh bằng tổ hợp phím **Ctrl + 5**.
- Bước 4:** Đặt các cổng cần thiết vào và kết nối dây dẫn. Tham khảo thêm Hướng dẫn của Logism cho các thao tác này.
- Bước 5:** Khi hoàn thành, chọn **Project \ Analyze Circuit**. Bảng sự thật thể hiện ở thẻ **Table**; Biểu thức được thể hiện ở thẻ **Expression**.



Hình A.8: Mạch logic được ghép nối từ các cổng định nghĩa sẵn