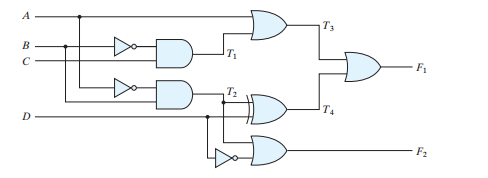
Bài Tập chương 5: Combinational Logic

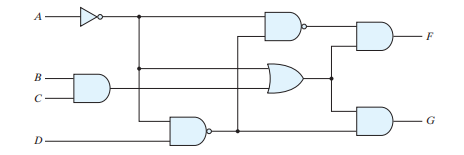
1. Xem mạch được hiển thị trên hình.



1. Rút gọn biểu thức inputs từ T1 tới T4, và tìm ra biểu thức outputs F1 và F2
2. Viết bảng chân trị với 4 giá trị inputs gồm từ T1 tới T4 và 2 giá trị outputs F1 F2.
3. vẽ bảng Kmaps với bảng chân trị ở câu b) và rút gọn biểu thức tương đương câu a)

Biểu thức được đơn giản hóa của output của F và G theo các biến input.

1. Cho mạch như hình:



1. Viết biểu thức với 4 outputs theo các biến đầu vào
2. Nếu mạch được mô tả bởi bảng chân trị, thì có bao nhiêu hàng và bao nhiêu cột trong bảng
3. Thiết kế mạch với 3 input và 1 output
4. Output là 1 khi giá trị của input nhỏ hơn 3. Ngược lại thì 0
5. Output là 1 khi giá trị của input là số chẵn.
6. Thiết kế mạch với 3 inputs x,y,z và 3 outputs A,B,C.Khi input tính theo hệ nhị phân là 0,1,2, hoặc 3 thì output theo hệ nhị phân lớn hơn input. Khi input theo hệ nhị phân 4,5,6 hoặc 7 thì output theo hệ nhị phân nhỏ hơn input.
7. A Majority

Thiết kế mạch Majority bằng bảng chân trị, biểu thức, biểu đồ logic

1. Thiết kế mạch mà mạch đó chuyển 4 bit trong mã Gray sang 4 bit trong số nhị phân

Vẽ mạch trên với cổng exclusive-or

1. Thiết kế mạch chuyển đổi từ số thập phân sang:
2. 84-2-1 sang to BCD
3. 84-2-1 sang mã Gray
4. Thiết kế mạch 4bitđể chuyển đổi số nhị phân sang bù 2’s (output tạo ra bù 2’s của số nhị phân đầu vào). Mạch có thể được biểu diễn với cổng XOR. Dự đoán hàm output với 5bit của phần bù.
5. Sử dụng mạch có 4 half-adders:
6. Thiết kế mạch cộng 1 vào số 4bit nhị phân
7. Thiết kế mạch trừ 1 vào số 4bit nhị phân
8. Thiết kế mạch tổ hợp so sánh hai số 4bit để kiểm tra xem chúng có bằng nhau không.

Đầu ra của mạch bằng 1 nếu hai số bằng nhau và bằng 0 nếu ngược lại.

1. Thiết kế bộ exess-3 thành nhị phân bằng cách sử dụng các tổ hợp( không sử dụng được thì không cần quan tâm)
2. 4.23 Vẽ sơ đồ logic của 2-to-4-line decoder sử dụng:
3. chỉ cổng NOR
4. chỉ cổng NAND.

Bao gồm một đầu vào kích hoạt. (enable)

1. Thiết kế một decoder BCD sang số thập phân bằng cách sử dụng các tổ hợp mã BCD(không được sử dụng dưới dạng BCD thì không cần quan tâm).
2. Xây dựng một 4-to-16-line decoder với năm 2-to-4-line decoder với enable(E).
3. Một mạch tổ hợp được xác định bởi ba hàm Boolean sau:

F (A, B, C) =2(1,4,6)

F2(A, B, C) =2(3,5)

F3(A, B, C) =2(2,4,6,7)

Thực hiện mạch với bộ giải mã được xây dựng bằng cổng NAND (tương tự như Hình 4.19) và cổng NAND hoặc AND được kết nối với đầu ra của bộ giải mã. Sử dụng sơ đồ khối cho bộ giải mã. Giảm thiểu số lượng đầu vào trong các cổng bên ngoài.

1. ử dụng bộ giải mã và các cổng bên ngoài, hãy thiết kế mạch tổ hợp được xác định bởi ba hàm Boolean sau:

(a) F₁ = x'yz' +xz

F₂ = xy'z' + x'y

F3 = x'y'z' + xy

(b) F₁ = (y + x)z

F₂ = y'z'x'y + yz'

F3 = (x + y)z

1. Chỉ định bảng chân trị của encoder ưu tiên bát phân sang nhị phân. Cung cấp một output V để chỉ ra rằng có ít nhất một trong inputs. Input có số chỉ số dưới cao nhất có mức ưu tiên cao nhất. Giá trị của input sẽ là bao nhiêu nếu input D2 và D6 đồng thời là 1?
2. Xây dựng bộ ghép kênh 16 x 1 với hai bộ ghép kênh 8 X 1 và một bộ ghép kênh 2 × 1. Sử dụng sơ đồ khối.
3. thực hiện chức năng Boolean sau với multiplexer

a. F (a, b, c, d) = ∑(0, 2, 5, 8, 10, 14)

b.F (A, B, C, D) = π(2,6,11)

1. thực hiện một full adder với 2 4x1 multiplexers
2. Một multiplexers 8 x 1 có inputs A, B và C được kết nối với Selection inputs S₂, S₁ và S0. Các đầu vào dữ liệu I0 thông qua I7 như sau:
3. I1=I2=I7=0; I3=I5=1; I0=I4=D; I6=D
4. I1=I2=0; I3=I7=1; I5=I4=D; I6=I0=D

Xác định chức năng boolean mà multiplexers thực hiện.

1. Thực hiện chức năng boolean sau multiplexers 4 x 1 và cổng external.

(a) f₁ (a, b, c, d) =∑ (1, 3, 4, 11, 12, 13, 14, 15)

(b) f₂ (a, b, c, d) =∑ (1, 2, 5, 7, 8, 10, 11, 13, 15)

Kết nối đầu vào A và B với selections. Các yêu cầu đầu vào cho bốn dữ liệu

Các dòng sẽ là một hàm của các biến C và D. Các giá trị được bằng cách biểu thị F như

một hàm của C và D cho mỗi bốn trường hợp khi AB= 00,01, 10 và 11. các hàm này

có thể được thực hiện với external gates.