

# Chương 3

## THIẾT KẾ MẠCH LOGIC TỔ HỢP BẰNG VHDL

### GIỚI THIỆU

#### THIẾT KẾ MẠCH GIẢI MÃ – MẠCH MÃ HOÁ

THIẾT KẾ MẠCH GIẢI MÃ

THIẾT KẾ MẠCH MÃ HOÁ

THIẾT KẾ MẠCH GIẢI MÃ LED 7 ĐOẠN LOẠI ANODE CHUNG

#### THIẾT KẾ MẠCH ĐA HỢP – MẠCH GIẢI ĐA HỢP

THIẾT KẾ MẠCH ĐA HỢP

THIẾT KẾ MẠCH GIẢI ĐA HỢP

#### CÂU HỎI ÔN TẬP VÀ BÀI TẬP

CÁC HÌNH VẼ

Hình 3-1. Sơ đồ khối mạch GM 2 - 4.

Hình 3-2. Sơ đồ khối mạch GM 3 - 8.

Hình 3-3. Sơ đồ khối mạch MH 4 - 2.

Hình 3-4. Sơ đồ khối mạch GM led 7 đoạn loại anode chung.

Hình 3-5. Sơ đồ khối mạch ĐH 4 vào.

Hình 3-6. Sơ đồ khối mạch GĐH 4 ra.

CÁC BẢNG

Bảng 3-1. BTT mạch GM 2 - 4.

Bảng 3-2. BTT mạch GM 3 - 8.

Bảng 3-3. BTT mạch MH 4 - 2.

Bảng 3-4. BTT mạch GM led 7 đoạn anode chung.

Bảng 3-5. BTT mạch đa hợp 4 ngõ vào.

Bảng 3-6. BTT mạch GĐH 4 ra.

Bản quyền thuộc về Trường ĐH Sư phạm Kỹ thuật TP. HCM

## I. GIỚI THIỆU:

Trong phần này sẽ thiết kế các mạch logic tổ hợp dùng ngôn ngữ VHDL và sử dụng thiết bị lập trình.

Các mạch logic tổ hợp bao gồm mạch giải mã n đường sang m đường, mạch mã hoá m đường sang n đường, mạch dồn kênh và mạch phân kênh, mạch giải mã led 7 đoạn loại anode chung và cathode chung.

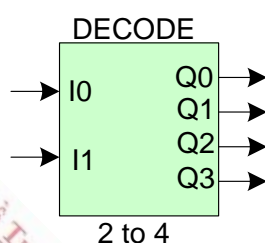
Các thiết bị lập trình có thể dùng CPLD XC9572, XC 95144, Coolrunner XC2C256.

## II. THIẾT KẾ MẠCH GIẢI MÃ – MẠCH MÃ HOÁ

### 1. THIẾT KẾ MẠCH GIẢI MÃ:

**Bài 3-1:** Thiết kế mạch giải mã 2 đường sang 4 đường với ngõ ra tích cực mức cao:

**Bước 1:** Vẽ sơ đồ khối của mạch:



Hình 3-1. Sơ đồ khối mạch GM 2 - 4.

**Bước 2:** Lập bảng trạng thái:

Ngõ vào		Ngõ ra			
I1	I0	Q3	Q2	Q1	Q0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Bảng 3-1. BTT mạch GM 2 - 4.

**Bước 3:** Viết chương trình:

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity giamma_24 is
    Port ( I : in  STD_LOGIC_VECTOR (1 downto 0);
          Q : out STD_LOGIC_VECTOR (3 downto 0));
end giamma_24;

architecture Behavioral of giamma_24 is

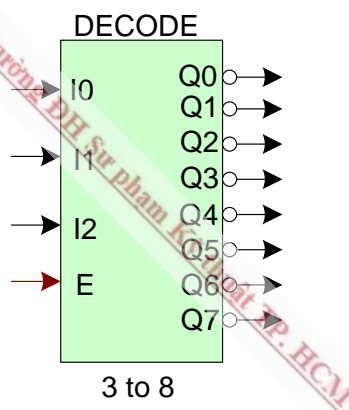
```

```

begin
  PROCESS (I)
  BEGIN
    CASE I IS
      WHEN "00" => Q <= "0001";
      WHEN "01" => Q <= "0010";
      WHEN "10" => Q <= "0100";
      WHEN "11" => Q <= "1000";
      WHEN OTHERS => NULL;
    END CASE;
  END PROCESS;
end Behavioral;
    
```

**Bài 3-2:** Thiết kế mạch giải mã 3 đường sang 8 đường với ngõ ra tích cực mức thấp và 1 ngõ cho phép E:

**Bước 1:** Vẽ sơ đồ khối của mạch:



**Hình 3-2.** Sơ đồ khối mạch GM 3 - 8.

**Bước 2:** Lập bảng trạng thái:

Ngõ vào				Ngõ ra							
E	I2	I1	I0	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0
0	x	x	x	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	0
1	0	0	1	1	1	1	1	1	1	0	1
1	0	1	0	1	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1	0	1	1	1
1	1	0	0	1	1	1	0	1	1	1	1
1	1	0	1	1	1	0	1	1	1	1	1
1	1	1	0	1	0	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1	1	1	1

**Bảng 3-2.** BTT mạch GM 3 - 8.

**Bước 3:** Viết chương trình:

```

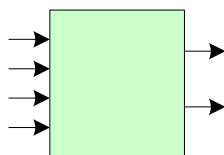
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity giaima_38 is
    Port ( I : in STD_LOGIC_VECTOR (2 downto 0);
          Q : out STD_LOGIC_VECTOR (7 downto 0);
          E : in STD_LOGIC);
end giaima_38;
architecture Behavioral of giaima_38 is
BEGIN
    PROCESS (I,E)
    BEGIN
        IF E = '0' THEN Q <= "11111111";
        ELSE
            CASE I IS
                WHEN "000" => Q <= "11111110";
                WHEN "001" => Q <= "111111101";
                WHEN "010" => Q <= "11111011";
                WHEN "011" => Q <= "11110111";
                WHEN "100" => Q <= "11101111";
                WHEN "101" => Q <= "11011111";
                WHEN "110" => Q <= "10111111";
                WHEN "111" => Q <= "01111111";
                WHEN OTHERS => NULL;
            END CASE;
        END IF;
    END PROCESS;
end Behavioral;

```

## 2. THIẾT KẾ MẠCH MÃ HOÁ

**Bài 3-3:** Thiết kế mạch mã hoá 4 đường sang 2 đường với ngõ vào tích cực mức cao.

**Bước 1:** Vẽ sơ đồ khối của mạch:



Hình 3-3. Sơ đồ khối mạch MH 4 - 2.

Bước 2: Lập bảng trạng thái:

Ngõ vào				Ngõ ra	
I3	I2	I1	I0	Q1	Q0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Bảng 3-3. BTT mạch MH 4 - 2.

Bước 3: Viết chương trình:

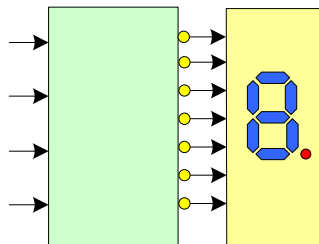
```

library IEEE;
use IEEE.STD_LOGIC_4164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity maha42 is
    Port ( I : in STD_LOGIC_VECTOR (3 downto 0);
          Q : out STD_LOGIC_VECTOR (1 downto 0));
end maha42;
architecture Behavioral of maha42 is
begin
    PROCESS(I)
    BEGIN
        CASE I IS
            WHEN "0001" => Q <= "00";
            WHEN "0010" => Q <= "01";
            WHEN "0100" => Q <= "10";
            WHEN "1000" => Q <= "11";
            WHEN OTHERS => NULL;
        END CASE;
    END PROCESS;
end Behavioral;
    
```

E  
I0  
I1  
I2  
I3

**Bài 3-4:** Thiết kế mạch giải mã led 7 đoạn loại anode chung

**Bước 1:** Vẽ sơ đồ khối của mạch:



**Hình 3-4.** Sơ đồ khối mạch GM led 7 đoạn loại anode chung.

**Bước 2:** Lập bảng trạng thái:

Số tp	Ngõ vào				Ngõ ra						Số hex	
	I3	I2	I1	I0	$\bar{g}$	$\bar{f}$	$\bar{e}$	$\bar{d}$	$\bar{c}$	$\bar{b}$		$\bar{a}$
0	0	0	0	0	1	0	0	0	0	0	0	40
1	0	0	0	1	1	1	1	1	0	0	1	79
2	0	0	1	0	0	1	0	0	1	0	0	24
3	0	0	1	1	0	1	1	0	0	0	0	30
4	0	1	0	0	0	0	1	1	0	0	1	19
5	0	1	0	1	0	0	1	0	0	1	0	22
6	0	1	1	0	0	0	0	0	0	1	0	02
7	0	1	1	1	0	1	1	1	0	0	0	78
8	1	0	0	1	0	0	0	0	0	0	0	00
9	1	0	0	1	0	0	1	0	0	0	0	10
tắt	1	0	1	0	1	1	1	1	1	1	1	7F
tắt	1	0	1	1	1	1	1	1	1	1	1	7F
tắt	1	1	0	0	1	1	1	1	1	1	1	7F
tắt	1	1	0	1	1	1	1	1	1	1	1	7F
tắt	1	1	1	0	1	1	1	1	1	1	1	7F
tắt	1	1	1	1	1	1	1	1	1	1	1	7F

**Bảng 3-4.** BTT mạch GM led 7 đoạn anode chung.

**Bước 3:** Viết chương trình:

Để đơn giản nên trong chương trình thì ngõ ra được đặt tên theo dạng vector và tên là Y.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity giaima7doan is
    Port ( I : in STD_LOGIC_VECTOR (3 downto 0);
          Y : out STD_LOGIC_VECTOR (6 downto 0));
end giaima7doan;
    
```

SEGM  
DECC

10

11

12

13

```

architecture Behavioral of giaima7doan is
begin
  PROCESS (I)
  BEGIN
    CASE I IS
      when "0000" => Y <= "1000000"; -- so 0
      when "0001" => Y <= "1111001"; -- so 1
      when "0010" => Y <= "0100100"; -- so 2
      when "0011" => Y <= "0110000"; -- so 3

      when "0100" => Y <= "0011001"; -- so 4
      when "0101" => Y <= "0010010"; -- so 5
      when "0110" => Y <= "0000010"; -- so 6
      when "0111" => Y <= "1111000"; -- so 7

      when "1000" => Y <= "0000000"; -- so 8
      when "1001" => Y <= "0010000"; -- so 9
      when others => Y <= "1111111"; -- tat

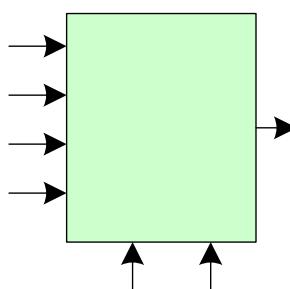
    END CASE;
  END PROCESS;
end Behavioral;
    
```

### III. THIẾT KẾ MẠCH ĐA HỢP – MẠCH GIẢI ĐA HỢP

#### 1. THIẾT KẾ MẠCH ĐA HỢP:

**Bài 3-5:** Thiết kế mạch đa hợp 4 ngõ vào, 1 ngõ ra, 2 ngõ lựa chọn.

**Bước 1:** Vẽ sơ đồ khối của mạch:



Hình 3-5. Sơ đồ khối mạch ĐH 4 vào.



**Bước 2:** Lập bảng trạng thái:

Ngõ vào						Ngõ ra
S1	S0	I3	I2	I1	I0	Q
0	0	x	x	x	I0	I0
0	1	x	X	I1	X	I1
1	0	X	I2	x	X	I2
1	1	I3	x	x	X	I3

**Bảng 3-5.** BTT mạch đa hợp 4 ngõ vào.

**Bước 3:** Viết chương trình:

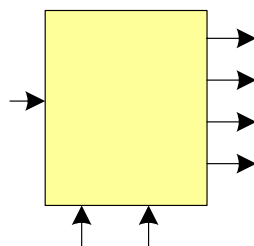
```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity GMA is
    Port ( I : in STD_LOGIC_VECTOR (3 downto 0);
          S : in STD_LOGIC_VECTOR (1 downto 0);
          Q : out STD_LOGIC);
end GMA;
architecture Behavioral of GMA is
begin
    PROCESS(I,S)
    BEGIN
        CASE S IS
            WHEN "00" => Q <= I(0);
            WHEN "01" => Q <= I(1);
            WHEN "10" => Q <= I(2);
            WHEN "11" => Q <= I(3);
            WHEN OTHERS => NULL;
        END CASE;
    END PROCESS;
end Behavioral;
    
```

## 2. THIẾT KẾ MẠCH GIẢI ĐA HỢP

**Bài 3-6:** Thiết kế mạch giải đa hợp 1 ngõ vào, 4 ngõ ra, 2 ngõ lựa chọn.

**Bước 1:** Vẽ sơ đồ khối của mạch:



Hình 3-6. Sơ đồ khối mạch GDH 4 ra.

**Bước 2:** Lập bảng trạng thái:

Ngõ vào			Ngõ ra			
I	S1	S0	Y3	Y2	Y1	Y0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Bảng 3-6. BTT mạch GDH 4 ra.

**Bước 3:** Viết chương trình:

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity GDH is
    Port ( I : in STD_LOGIC;
          S : in STD_LOGIC_VECTOR (1 downto 0);
          Q : out STD_LOGIC_VECTOR (3 downto 0));
end GDH;
architecture Behavioral of GDH is
begin
    PROCESS(I,S)
    BEGIN
        CASE S IS
            WHEN "00" => Q(0) <= I;
            WHEN "01" => Q(1) <= I;
            WHEN "10" => Q(2) <= I;
            WHEN "11" => Q(3) <= I;
            WHEN OTHERS => NULL;
        END CASE;
    END PROCESS;
end Behavioral;
    
```

```
END PROCESS;
```

```
end Behavioral;
```

#### IV. CÂU HỎI ÔN TẬP VÀ BÀI TẬP

**Bài tập 3-1:** Thiết kế mạch giải mã 2 đường sang 4 đường với ngõ ra tích cực mức thấp và có một tín hiệu cho phép E tích cực mức cao.

**Bài tập 3-2:** Thiết kế mạch giải mã 2 đường sang 4 đường với ngõ ra tích cực mức thấp và có một tín hiệu cho phép E1 tích cực mức cao, và một tín hiệu cho phép E2 tích cực mức thấp.

**Bài tập 3-3:** Thiết kế mạch giải mã bên trong có 4 mạch giải mã 2 đường sang 4 đường với ngõ ra tích cực mức thấp.

**Bài tập 3-4:** Thiết kế mạch giải mã 3 đường sang 8 đường với ngõ ra tích cực mức thấp và có một tín hiệu cho phép E1 tích cực mức cao, và một tín hiệu cho phép E2 tích cực mức thấp.

**Bài tập 3-5:** Thiết kế mạch mã hoá 8 đường sang 3 đường với các ngõ vào tích cực mức thấp.

**Bài tập 3-6:** Thiết kế mạch mã hoá 8 đường sang 3 đường với các ngõ vào tích cực mức cao.

**Bài tập 3-7:** Thiết kế mạch giải mã led 7 đoạn loại cathode chung.

**Bài tập 3-8:** Thiết kế mạch giải mã led 7 đoạn loại giống như IC 74247.

**Bài tập 3-9:** Thiết kế mạch giải mã led 7 đoạn loại giống như IC 4511.

**Bài tập 3-10:** Thiết kế mạch đa hợp 8 ngõ vào, 1 ngõ ra và 3 ngõ lựa chọn.

**Bài tập 3-11:** Thiết kế mạch đa hợp 16 ngõ vào, 1 ngõ ra và 4 ngõ lựa chọn.

**Bài tập 3-12:** Thiết kế mạch giải đa hợp 1 ngõ vào, 8 ngõ ra và 3 ngõ lựa chọn.

**Bài tập 3-13:** Thiết kế mạch giải đa hợp 1 ngõ vào, 16 ngõ ra và 4 ngõ lựa chọn.

**Bài tập 3-14:** Thiết kế mạch giải đa hợp giống như IC 74151.

**Bài tập 3-15:** Thiết kế mạch so sánh 2 số 8 bit A và B và có 3 led hiển thị LEDLH, LEDBA, LEDNH. Nếu  $A > B$  thì LEDLH sáng, nếu  $A = B$  thì LEDBA sáng, nếu  $A < B$  thì LEDNH sáng.

**Bài tập 3-16:** Thiết kế mạch chuyển đổi số nhị phân 8 bit thành số BCD.

**Bài tập 3-17:** Thiết kế mạch chuyển đổi số 2 số BCD thành số nhị phân.

**Bài tập 3-18:** Thiết kế mạch kiểm tra chẵn lẻ của một số nhị phân 8 bit, nếu là số chẵn thì đèn chẵn sáng, nếu là số lẻ thì đèn lẻ sáng.

# end