**THỰC HÀNH NHẬP MÔN MẠCH SỐ - LỚP PH002.N14**

**BÀI THỰC HÀNH 3: PHÂN TÍCH VÀ THIẾT KẾ MẠCH SỐ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Giảng viên hướng dẫn** | **Đỗ Trí Nhựt** | | **ĐIỂM** |
| **Sinh viên thực hiện** | **Lại Quan Thiên** | **22521385** |  |
| **Sinh viên thực hiện** |  |  |  |

1. **Mục tiêu**

* Phân tích, thiết kế, đánh giá mạch số từ đặc tả kỹ thuật
* Làm quen với IC7447 để hiện thị giá trị của một số BCD

1. **Nội dung**
2. **Vận dụng ở nhà**

**Câu 1: Thiết kế bộ báo động (A=1) cho lái xe với các tình huống: Bugi bật (B=1) và Cửa mở (C=0), hoặc chưa Cài dây an toàn (D=0) và Bugi bật (B=1)**

* Xác định các ngõ vào và ngõ ra của mạch:

+ Ngõ vào: Bugi, cửa, dây an toàn.

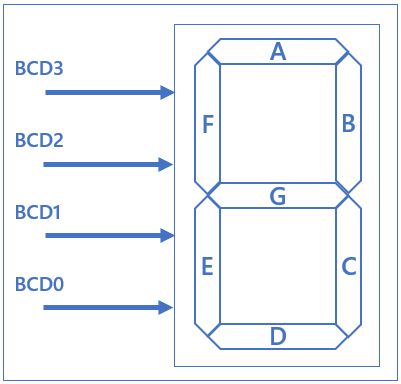
+ Ngõ ra: Bộ báo động.

* Hoàn thành bảng chân trị bên dưới:

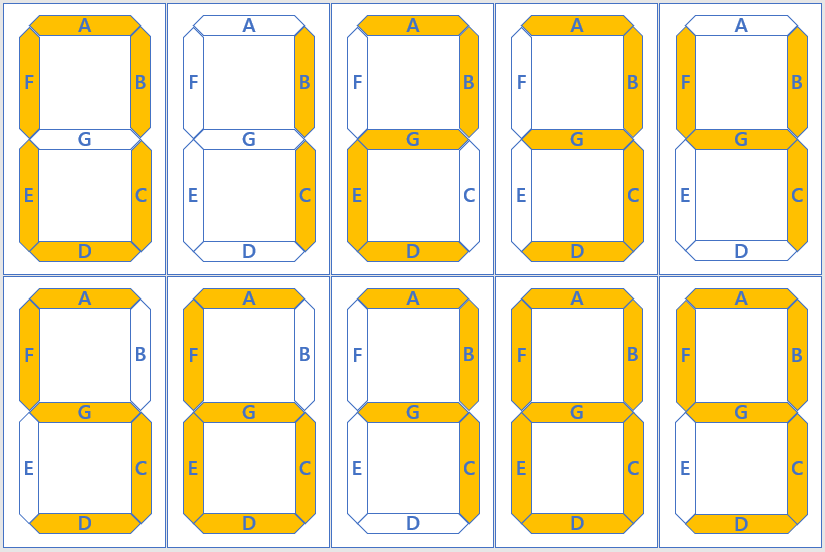
|  |  |  |  |
| --- | --- | --- | --- |
| B | C | D | A |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

|  |  |
| --- | --- |
| Viết biểu thức dạng SOP (không rút gọn):  FA(B, C, D)= BC’D + BCD’ + BC’D’ | Viết biểu thức dạng POS (không rút gọn):  FA(B, C, D)= (B+C+D)(B+C+D’)(B+C’+D)(B+C’+D’)(B’+C’+D’) |
| Tính chi phí biểu thức ở trên theo dạng SOP:  C(FA)= 3 + 3 + 3 +3 = 12 | Tính chi phí biểu thức ở trên theo dạng POS:  C(FA)= 5 + 3 + 3 + 3 + 3 + 3 = 20 |
| Rút gọn luận lý biểu thức ở trên bằng phương  pháp Đại số Bool:  FA(B, C, D) = BC’D + BCD’ + BC’D’  = BC’(D+D’) + BCD’  = BC’ + BCD’  = BD’ + BC’  Rút gọn luận lý biểu thức ở trên bằng phương pháp K-map:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | A | BC | |  |  |  | | D |  | 00 | 01 | 11 | 10 | |  | 0 |  |  | 1 | 1 | |  | 1 |  |  |  | 1 |   FA(B, C, D)=BD’+BC’ | Rút gọn luận lý biểu thức ở trên bằng phương  pháp Đại số Bool:  FA(B, C, D) = (B+C+D)(B+C+D’)(B+C’+D)(B+C’+D’)(B’+C’+D’)  = (B+BC+C)(B+BC’+C’)(B’+C’+D’)  = (B+BC’+BC’+BC+BC)(B’+C’+D’)  = B(B’+C’+D’) = BC’+BD’  Rút gọn luận lý biểu thức ở trên bằng phương pháp K-map:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | A | BC | |  |  |  | | D |  | 00 | 01 | 11 | 10 | |  | 0 | 0 | 0 |  |  | |  | 1 | 0 | 0 | 0 |  |   FA(B, C, D)=B(C’+D’) |
| Tính chi phí biểu thức sau khi đã rút gọn:  O(FA)= 2 + 2 + 2 = 6 | Tính chi phí biểu thức sau khi đã rút gọn:  O(FA)= 2 + 0 + 2 = 4 |
| Nhận xét chi phí khi chưa rút gọn và sau khi rút gọn: Chi phí giảm | Nhận xét chi phí khi chưa rút gọn và sau khi rút gọn: Chi phí giảm đáng kể sau khi rút gọn. |

**Câu 2: Thiết kế Bộ giải mã BCD\_2\_7SEGMENT**



* Đầu vào là số BCD 4bits: {BCD3, BCD2, BCD1, BCD0}
* Đầu ra là 1 tổ hợp 7 tín hiệu đèn, được sắp xếp để biểu diễn số BCD: A, B, C, D, E, F, G
* Khi Đầu vào từ 0000 đến 1001 thì, tổ hợp 7 tín hiệu sẽ biểu diễn từ số 0 đến số 9 tương ứng như hình bên dưới. Nếu đầu vào là 1111 thì không tín hiệu nào sáng. Còn lại thì 7 tín hiệu đầu ra nhận giá trị tùy định (X), chú ý: đèn LED sáng khi giá trị của nó là 0:



* Hoàn thảnh bảng chân trị bên dưới (Chú ý, I viết thay cho BCD):

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **I3** | **I2** | **I1** | **I0** | **A** | **B** | **C** | **D** | **E** | **F** | **G** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| **I3** | **I2** | **I1** | **I0** | **A** | **B** | **C** | **D** | **E** | **F** | **G** |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | X | X | X | X | X | X | X |
| 1 | 0 | 1 | 1 | X | X | X | X | X | X | X |
| 1 | 1 | 0 | 0 | X | X | X | X | X | X | X |
| 1 | 1 | 0 | 1 | X | X | X | X | X | X | X |
| 1 | 1 | 1 | 0 | X | X | X | X | X | X | X |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

* Rút gọn luận lý các hàm luận lý trên bằng phương pháp K-map:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A** | I1I0 | |  |  |  |  |  |  | **B** | I1I0 | |  |  |  |  |  |
| I3I2 | |  | 00 | 01 | 11 | 10 |  |  | I3I2 | |  | 00 | 01 | 11 | 10 |  |  |
|  |  | 00 |  | 1 |  |  |  |  |  |  | 00 |  |  |  |  |  |  |
|  |  | 01 | 1 |  |  |  |  |  |  |  | 01 |  | 1 |  | 1 |  |  |
|  |  | 11 | X | X | 1 | X |  |  |  |  | 11 | X | X | 1 | X |  |  |
|  |  | 10 |  |  | X | X |  |  |  |  | 10 |  |  | X | X |  |  |
| FA(I3, I2. I1, I0) =  = I3I1 + I2I1’I0’ + I3’I2’I1’I0 | | | | | | | | | FB(I3, I2. I1, I0) = I3I1 + I2I1’I0 + I2I1I0’  = I3I1 + I2(I1’I0 + I1I0’)  = I3I1 + I2(I1 xor I0) | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **C** | I1I0 | |  |  |  |  |  |  | **D** | I1I0 | |  |  |  |  |  |
| I3I2 | |  | 00 | 01 | 11 | 10 |  |  | I3I2 | |  | 00 | 01 | 11 | 10 |  |  |
|  |  | 00 |  |  |  | 1 |  |  |  |  | 00 |  | 1 |  |  |  |  |
|  |  | 01 |  |  |  |  |  |  |  |  | 01 | 1 |  | 1 |  |  |  |
|  |  | 11 | X | X | 1 | X |  |  |  |  | 11 | X | X | 1 | X |  |  |
|  |  | 10 |  |  | X | X |  |  |  |  | 10 |  |  | X | X |  |  |
| FC(I3, I2. I1, I0) = I3I1 + I2’I1I0’ | | | | | | | | | FD(I3, I2. I1, I0) =  = I2I1I0 + I2I1’I0’ + I3’I2’I1’I0  = I2(I1 xnor I0) + I3’I2’I1’I0 | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **E** | I1I0 | |  |  |  |  |  |  | **F** | I1I0 | |  |  |  |  |  |
| I3I2 | |  | 00 | 01 | 11 | 10 |  |  | I3I2 | |  | 00 | 01 | 11 | 10 |  |  |
|  |  | 00 |  | 1 | 1 |  |  |  |  |  | 00 |  | 1 | 1 | 1 |  |  |
|  |  | 01 | 1 | 1 | 1 |  |  |  |  |  | 01 |  |  | 1 |  |  |  |
|  |  | 11 | X | X | 1 | X |  |  |  |  | 11 | X | X | 1 | X |  |  |
|  |  | 10 |  | 1 | X | X |  |  |  |  | 10 |  |  | X | X |  |  |
| FE(I3, I2. I1, I0)= I0 + I2I1’I0’ | | | | | | | | | FF(I3, I2. I1, I0)= I1I0 + I2’I1 + I3’I2’I0 | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **G** | I1I0 | |  |  |  |  |  |
| I3I2 | |  | 00 | 01 | 11 | 10 |  |  |
|  |  | 00 | 1 | 1 |  |  |  |  |
|  |  | 01 |  |  |  |  |  |  |
|  |  | 11 | X | X | 1 | X |  |  |
|  |  | 10 |  |  | X | X |  |  |
| FG(I3, I2. I1, I0)= I3’I2’I1’ + I3I2I1I0 | | | | | | | | |

* Vẽ và đóng gói riêng rẽ 7 mạch trên Quartus II. Sau đó đóng gói 7 module này thành một module mới để hoàn thiện yêu cầu thiết kế BCD\_2\_7SEGMENT:

|  |
| --- |
| Mạch A:  Diagram, schematic  Description automatically generated  Mạch B:  Diagram  Description automatically generated  Mạch C:  Diagram  Description automatically generated  Mạch D:  Diagram, schematic  Description automatically generated  Mạch E:  Diagram  Description automatically generated  Mạch F:  Diagram, schematic  Description automatically generated  Mạch G:  Diagram, schematic  Description automatically generated  Mạch BCD\_2\_7\_SEGMENT (Mạch đóng gói A, B, C, D, E, F, G):Diagram  Description automatically generated  Mạch BCD\_2\_7\_SEGMENT đóng gói:  Diagram  Description automatically generated  Waveform mạch BCD\_2\_7\_SEGMENT:  A computer screen capture  Description automatically generated with low confidence |

* Trong đường dẫn tải về có một thư mục là BCD\_2\_7SEG, khi đó sẽ có một symbol là BCD\_2\_7SEG trong thư viện, đây là mạch số giống với yêu cầu của đề bài. Ngoài ra, trong đường dẫn tải về có một thư mục là cmp\_1bit, sao chép các tệp tin trong thư mục vào project vừa thực hiện, khi đó sẽ có một symbol là cmp\_1bit trong thư viện, đây là mạch số có chức năng so sánh 2 bits có bằng nhau hay không, nếu bằng nhau thì kết quả là 1. Sử dụng BCD\_2\_7SEG và cmp\_1bit để kiểm tra tính đúng đắn của mạch vừa đóng gói (16 trường hợp cần kiểm tra):