به نام خدا

گزارش‌‌کار فاز اول پروژه‌ی پایانی مدار های منطقی

طراحی بخش های ترکیبی

استاد کارگاه : خانم جلیلی

بهار کاویانی۹۷۳۱۰۵۱

یاسمن حق بین ۹۷۳۱۰۱۷

**بخش ۱: مدار فشار خون**

***افزونگی بیت توازن***

***الف.***از مدار طراحی شده در آزمایش ۴ می‌دانیم که اگه بیت توازن (که در این مثال بیت ۶ ام مدنظر است چون خود اطلاعات ۵ بیتی است) یک باشد یعنی ورودی دارای خطا بوده‌است. طبق صورت سوال قرارداد شده که بیت توازن زوج است. پس اگر به تعداد فرد یک داشته باشیم، ورودی خطاست.

***تشخیص دهنده‌ی غیر طبیعی بودن فشار خون***

***ج.*** تابع مجموعه مینترم های مدار:

F(b, c, d, e, f) = ∑ m(0, 1, 2, 3, 4, 5, 6, 7, 23, 24, 25, 26, 27, 28, 29, 30, 31)

***د.***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 11 | 01 | 00 |  | 10 | 11 | 01 | 00 | cd  ef |
| 1 | 1 | 0 | 0 | 00 | 0 | 0 | 1 | 1 | 00 |
| 1 | 1 | 0 | 0 | 01 | 0 | 0 | 1 | 1 | 01 |
| 1 | 1 | 1 | 0 | 11 | 0 | 0 | 1 | 1 | 11 |
| 1 | 1 | 0 | 0 | 10 | 0 | 0 | 1 | 1 | 10 |

b = 0 b = 1

SOP: F(b, c, d, e, f) = b’c’ + bc + c’def

***بخش۲: مدار غلظت خون***

***الف و ب. مدار تفکیک‌کننده‌ی گروه خونی***

F

1

0

C

0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| F | F | C | B | A |  |
| 1 | 1 | 0 | 0 | 0 | O- |
| 1 | 1 | 0 | 0 | O+ |
| 0 | 0 | 0 | 1 | 0 | A- |
| 0 | 1 | 1 | 0 | A+ |
| C | 0 | 0 | 0 | 1 | B- |
| 1 | 1 | 0 | 1 | B+ |
| 0 | 0 | 0 | 1 | 1 | AB- |
| 0 | 1 | 1 | 1 | AB+ |

A B

F(A, B, C) = A’B’ + AB’C

***د.***

*تابع مربوط به abnormalityP*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10 | 11 | 01 | 00 | ab cd |
| 0 | 1 | 1 | 1 | 00 |
| 1 | 1 | 1 | 1 | 01 |
| 1 | 1 | 0 | 1 | 11 |
| 1 | 1 | 1 | 1 | 10 |

abnormalityP(a, b, c, d) = (a’+b+c+d)(a+b’+c’+d’)

a

b

c

d

*تابع مربوط به abnormalityQ*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10 | 11 | 01 | 00 | ab cd |
| 0 | 1 | 1 | 1 | 00 |
| 0 | 1 | 1 | 1 | 01 |
| 1 | 1 | 0 | 1 | 11 |
| 1 | 1 | 0 | 1 | 10 |

abnormalityQ(a, b, c, d) = (a’+b+c)(a+b’+c’)

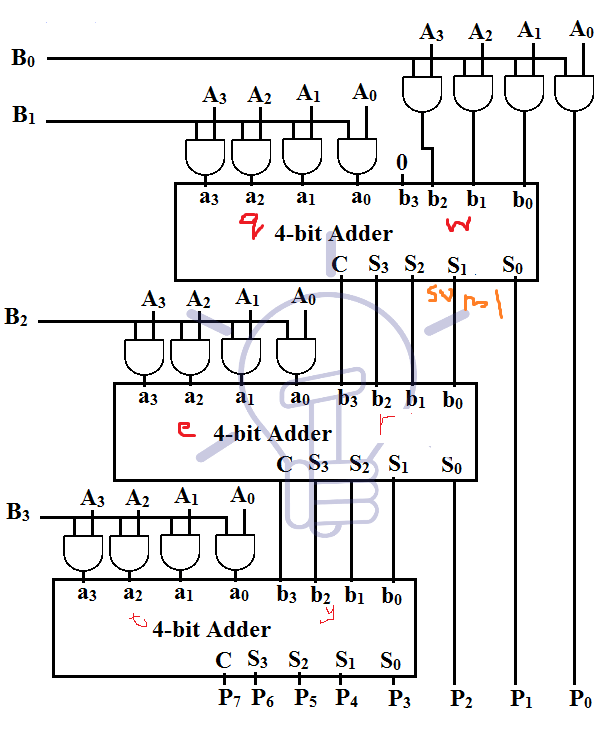
a

b

c

***بخش۴: مدار محاسبه‌کننده‌ی دمای بدن***

***ماژول ضرب‌کننده‌ی ۴ بیتی***



بخش های مختلف و سیم های نام‌گذاری شده در کد این بخش روی شکل نمایش داده شده‌اند.

module FourBitMultiplier(

input [3:0]A,

input [3:0]B,

output [7:0]result

);

wire [3:0]w, q, e, r, t, y;

wire [3:0]sum1, sum2, sum3;

wire cout1, cout2, cout3;

and

g0(result[0], A[0], B[0]),

g1(w[0], A[1], B[0]),

g2(q[0], A[0], B[1]),

g3(w[1], A[2], B[0]),

g4(q[1], A[1], B[1]),

g5(e[0], A[0], B[2]),

g6(w[2], A[3], B[0]),

g7(q[2], A[2], B[1]),

g8(e[1], A[1], B[2]),

g9(t[0], A[0], B[3]),

g10(q[3], A[3], B[1]),

g11(e[2], A[2], B[2]),

g12(t[1], A[1], B[3]),

g13(e[3], A[3], B[2]),

g14(t[2], A[2], B[3]),

g15(t[3], A[3], B[3]);

assign

w[3] = 0,

result[1] = sum1[0],

r[3] = cout1,

r[2] = sum1[3],

r[1] = sum1[2],

r[0] = sum1[1],

result[2] = sum2[0],

y[3] = cout2,

y[2] = sum2[3],

y[1] = sum2[2],

y[0] = sum2[1],

result[3] = sum3[0],

result[4] = sum3[1],

result[5] = sum3[2],

result[6] = sum3[3],

result[7] = cout3;

FourBitAdder

g16(q, w, sum1, cout1),

g17(e, r, sum2, cout2),

g18(t, y, sum3, cout3);

endmodule

***بخش امتیازی:***

چون در صورت سوال قدر مطلق عدد خواسته شده،‌ پس اعداد علامت‌دار هستند. در صورت سوال اشاره نشده که چطور علامت دار هستند در نتیجه ما ورودی را از نوع 2’s complement در نظر گرفتیم.

می‌دانیم در اعداد 2’s complement برای قرینه کردن عدد باید تک‌تک بیت ها را قرینه کرده و حاصل را با یک جمع کنیم.

در بخش زیر تک‌تک بیت ها را قرینه کردیم.

not g1(notbloodSensor[7],bloodSensor[7]),

g2(notbloodSensor[6],bloodSensor[6]),

g3(notbloodSensor[5],bloodSensor[5]),

g4(notbloodSensor[4],bloodSensor[4]),

g5(notbloodSensor[3],bloodSensor[3]),

g6(notbloodSensor[2],bloodSensor[2]),

g7(notbloodSensor[1],bloodSensor[1]),

g8(notbloodSensor[0],bloodSensor[0]);

در خط زیر هم عدد را با یک جمع کردیم.

EightBitAdder g9 (notbloodSensor , {7'b0, 1} , towsComplement)

حال باید تشخیص بدهیم که عدد مثبت است یا منفی. برای این بخش ابتدا از یک ماکس استفاده می‌کنیم. اگر بیت آخر ۱ بیت بود، باید قرینه‌ی عدد را داشته باشیم، اگر صفر بود، خود عدد. یک ماکس ۸ بیتی داریم که در تعریف ماژول آن از ۸ ماکس دو بیتی استفاده شده.

EightBitMux g10(bloodSensor,towsComplement,bloodSensor[7],absoluteNum);

در این خط خود عدد و 2’s complement آن به عنوان ورودی به ماکس داده شده. که select ما بیت آخر عدد است و خروجی یعنی قدر مطلق عدد در absoluteNum ذخیره می‌شود.

حال برای محاسبه‌ی تعداد یک های عدد، اول در بخش زیر

g11({7'b0, absoluteNum[7]},{7'b0, absoluteNum[6]},sum1),

g12({7'b0, absoluteNum[5]},{7'b0, absoluteNum[4]},sum2),

g13({7'b0, absoluteNum[3]},{7'b0, absoluteNum[2]},sum3),

g14({7'b0, absoluteNum[1]},{7'b0, absoluteNum[0]},sum4),

هر دو بیت را با هم جمع می‌کنیم و بعد در بخش زیر

g15(sum1,sum2,sum5),

g16(sum3,sum4,sum6),

g17(sum5,sum6,endsum);

آن جمع ها را با هم جمع می‌کنیم.

در آخر هم که این جمع را به خروجی ماژول assign می‌کنیم.