

به نام خداوند بخشنده مهربان

نام و نام خانوادگی: داریوش امیری

شماره دانشجویی: ۹۸۱۰۱۱۸۲

گزارش کار آزمایش اول:

ابتدا ماژول az1 را به عنوان ماژول اصلی آزمایش تعریف میکنیم.

سپس دو ماژول bcd_mod11, bcd_mod3 را به ترتیب برای بخش پذیری بر ۳ و ۱۱ تعریف میکنیم.

یک ماژول نیز برای جمع دو عدد ۴ بیتی به همراه بیت carry تعریف میکنیم.

یک ماژول دیگر نیز برای مقایسه برابری دو عدد ۵ بیتی تعریف میکنیم.

حال روند تعریف ماژول bcd_mod11 را توضیح خواهیم داد:

از آنجایی که عدد ورودی BCD است و یافتن بخش پذیری بر ۱۱ در اعداد دسیمال به این صورت است که ارقام را به صورت یک در میان با مثبت و منفی جمع میکنیم و در صورتی که حاصل ۰ یا از مضارب ۱۱ بود، بر ۱۱ بخش پذیر میباشد، در نتیجه رقم اول و سوم یعنی [11:8], [3:0] عدد BCD ورودی را جمع کرده و برای دو رقم دیگر نیز به طور مشابه انجام میدهیم.

حال مقایسه میکنیم آیا برابرند یا خیر. اگر برابر بودند یعنی طبق توضیحات بر ۱۱ بخش پذیر است. اگر نه:

ممکن است بر ۱۱ بخش پذیر باشد اما حاصل جمع کردن ۴ رقم مضربی از ۱۱ باشد.

چون عدد ورودی ۴ رقمی است بنابراین نهایتاً حاصل جمع میتواند ۱۸- تا ۱۸ باشد که برای بخش پذیری بر ۱۱ یعنی برابر ۰ یا ۱۱ یا ۱۱-.

حال در دو حالت چک میکنیم که قدر مطلق اختلاف دو عدد آیا ۱۱ میباشد یا خیر.

در حالت اول حاصل جمع [11:8], [3:0] را با ۱۱ جمع کرده و سپس برابرش را با حاصل جمع [15:12], [7:4] چک میکنیم.

در حالت دوم نیز حاصل جمع [15:12], [7:4] را با ۱۱ جمع کرده و سپس برابرش را با حاصل جمع [11:8], [3:0] چک میکنیم.

و در نهایت or ۳ تا بولین بدست آمده را به عنوان خروجی قرار میدهیم.

در آخر ماژول دیگری به نام bcd_3_add_each تعریف میکنیم که دو عدد ۴ بیتی را جمع کرده سپس با کم کردن مضارب ۳ از آن، اطمینان پیدا کنیم که از ۱۶ کمتر است. بیت carry حاصل جمع در جایگاه بیت ۵ قرار میگیرد یعنی اگر به دسیمال تبدیل شود ۲۸۴ خواهد بود یعنی ۱۶ که به

پیمانه ۳ همان ۱ میباشد یعنی جایگاه بیت اول یعنی ۲۸۰. به طریق مشابه بیت ۴ ام حاصل جمع نیز در دسیمال ضربدر ۲۸۳ یعنی ۸ خواهد شد که به پیمانه ۳ برابر ۲ خواهد بود یعنی ۲۸۱ یعنی جایگاه بیت دوم.

بنابراین بدین صورت عمل میکنیم که ۳ بیت سمت راست حاصل جمع (به همراه صفر در جایگاه بیت ۴ ام) را با عدد ۴ بیتی با بیت اول که بیت carry است، بیت دوم که در واقع بیت ۴ ام حاصل جمع اول است به همراه بیت ۳ ام و ۴ ام که دارای مقادیر ۰ هستند را جمع میکنیم و به عنوان خروجی قرار میدهیم.

حال در ماژول `bcd_mod3` ابتدا ۴ رقم اول و دوم عدد ورودی را با استفاده از ماژول `bcd_3_add_each` جمع کرده سپس حاصل را با ۴ رقم سوم مجدد با استفاده از ماژول فوق، جمع کرده و در نهایت حاصل را با ۴ رقم ۴ از سمت راست جمع کرده و ذخیره میکنیم. سپس عدد بدست آمده را که عددی ۴ بیتی است را با استفاده از گیت های پایه `xor, and, nor` چک میکنیم که آیا بر ۳ بخش پذیر است یا خیر.

هر یک از دو ماژول `bcd_mod11, bcd_mod3` یک بیت (یک بولین) به عنوان خروجی خواهند داشت که اگر ۱ باشد نشان می دهد بخش پذیر است وگرنه به معنای آن است که بر عدد مورد نظر بخش پذیر نمیشد سکه به ترتیب در `out[1], out[0]` ذخیره میکنیم.

در ادامه تصویری از تست پنج برای ۳ عدد ۴۵۶۵ و ۴۵۶۴ و ۳۲۶۷ یعنی ۰۱۰۰۰۱۰۱۰۱۱۰۰۱۰۰ و ۰۱۰۰۰۱۰۱۰۱۱۰۰۱۱۱ و ۰۰۱۱۰۰۱۰۱۱۰۰۱۱۱ آمده است.

