به نام خداوند بخشنده مهربان

نام و نام خانوادگی: داریوش امیری

شماره دانشجویی: 98101182

**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 3 و 11 تعریف میکنیم.

یک ماژول نیز برای جمع دو عدد 4 بیتی به همراه بیت carry تعریف میکنیم.

یک ماژول دیگر نیز برای مقایسه برابری دو عدد 5 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 11 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 0 یا از مضارب 11 بود، بر 11 بخش پذیر میباشد، در نتیجه رقم اول و سوم یعنی [3:0] , [11:8] 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 11 بخش پذیر است.

اگرنه:

ممکن است بر 11 بخش پذیر باشد اما حاصل جمع کردن 4 رقم مضربی از 11 باشد.

چون عدد ورودی 4 رقمی است بنابراین نهایتا حاصل جمع میتواند 18- تا 18 باشد که برای بخش پذیری بر 11 یعنی برابر 0 یا 11 یا 11-.

حال در دو حالت چک میکنیم که قدر مطلق اختلاف دو عدد آیا 11 میباشد یا خیر.

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

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

و در نهایت or 3 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 4 بیتی را جمع کرده سپس با کم کردن مضارب 3 از آن، اطمینان پیدا کنیم که از 16 کمتر است. بیت carry حالصل جمع در جایگاه بیت 5 ام قرار میگیرد یعنی اگر به دسیمال تبدیل شود 4^2 خواهد بود یعنی 16 که به پیمانه 3 همان 1 میباشد یعنی جایگاه بیت اول یعنی 0^2. به طریق مشابه بیت 4 ام حاصل جمع نیز در دسیمال ضربدر 3^2 یعنی 8 خواهد شد که به پیمانه 3 برابر 2 خواهد بود یعنی 1^2 یعنی جایگاه بیت دوم.

بنابراین بدین صورت عمل میکنیم که 3 بیت سمت راست حاصل جمع (به همراه صفر در جایگاه بیت 4 ام) را با عدد 4 بیتی با بیت اول که بیت carry است، بیت دوم که در واقع بیت 4 ام حاصل جمع اول است به همراه بیت 3 ام و 4 ام که دارای مقادیر 0 هستند را جمع میکنیم و به عنوان خروجی قرار میدهیم.

حال در ماژول bcd\_mod3 ابتدا 4 رقم اول و دوم عدد ورودی را با استفاده از ماژول bcd\_3\_add\_each جمع کرده سپس حاصل را با 4 رقم سوم مجدد با استفاده از ماژول فوق، جمع کرده و در نهایت حاصل را با 4 رقم 4 از سمت راست جمع کرده و ذخیره میکنیم.

سپس عدد بدست آمده را که عددی 4 بیتی است را با استفاده از گیت های پایه xor, and, nor چک میکنیم که آیا بر 3 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 1 باشد نشان می دهد بخش پذیر است وگرنه به معنای آن است که بر عدد مورد نظر بخش پذیر نمیباشد سکه به ترتیب در out[1], out[0] ذخیره میکنیم.

در ادامه تصویری از تست بنچ برای 3 عدد 4565 و 4564 و 3267 یعنی 0100010101100101 و 0100010101100100 و 0011001001100111آمده است.

