آز طراحی سیستم‌های دیجیتال

نیم‌سال اول ۱۴۰۰-۱۴۰۱

مدرس: دکتر سیاوش بیات سرمدی

دستیار آموزشی: خانم زینب رشیدی

دانشکده مهندسی کامپیوتر

گزارش آزمایش اول

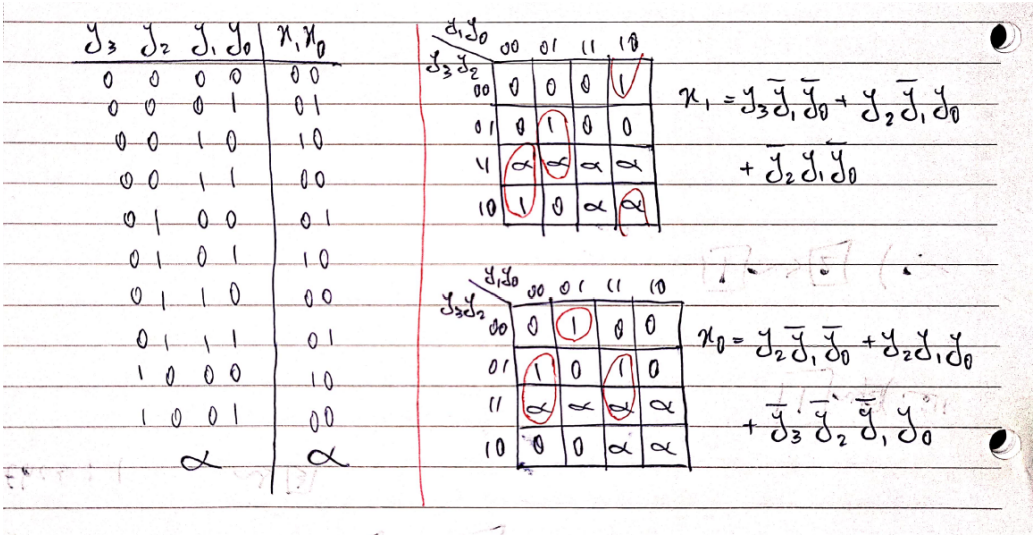
نام و نام‌خانوادگی: پرهام چاوشیان

شماره دانشجویی: ۹۸۱۰۰۱۱۸

برای بررسی بخش‌پذیری بر ۳ از قاعده زیر استفاده شده است:

$$\overline{abcd}\ mod\ 3 = (a\ mod\ 3) + (b\ mod\ 3) + (c\ mod\ 3) + (d\ mod\ 3)$$

برای استفاده از این قاعده ماژولی با نام $one\_bcd\_3\_checker$ نوشته شده است که ورودی آن یه عدد ۴بیتی BCD است و در خروجی باقی‌مانده آن عدد بر ۳ را می‌دهد. علت ۴ بیتی بودن خروجی انجام راحتتر عملیات جمع در ماژول‌های دیگر است. روابط استفاده شده از جدول صحت و جدول کارنوهای زیر آمده‌اند:



سپس ماژول دیگری به نام $bcd\_3\_checker$ داریم که یک عدد ۴رقمی BCD را می‌گیرد و باقی مانده تمام ارقام آن عدد بر سه را با یکدیگر جمع می‌کند. مشخص است که این عدد حداکثر می‌تواند ۸ باشد بنابراین با استفاده از همان ماژول اولیه، باقی ماتده این عدد را هم بر ۳ می‌یابیم که درواقع برابر باقی مانده عدد بر ۳ است. در پایان به کمک یک or و not بررسی می‌کنیم که اگر هر دو رقم باقی مانده برابر ۰ بود خروجی ۱ (به معنای بخش‌پذیری) و در غیر این صورت خروجی ۰ شود.

برای بررسی بخش‌پذیری بر ۱۱ از قاعده زیر استفاده شده است:

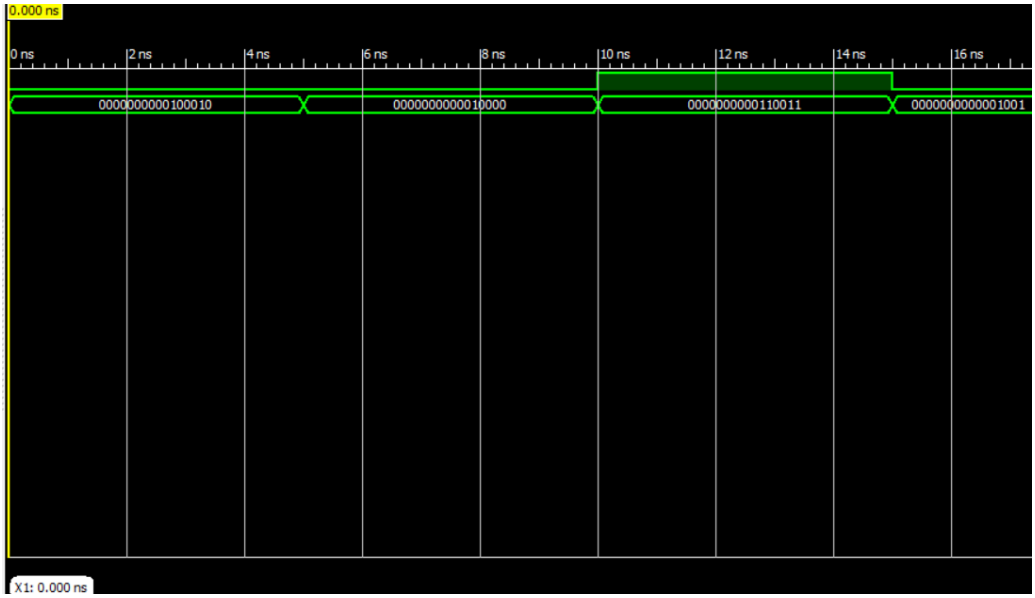
$$\overline{abcd}\ mod\ 11 = (b + d - a - c)\ mod\ 11$$

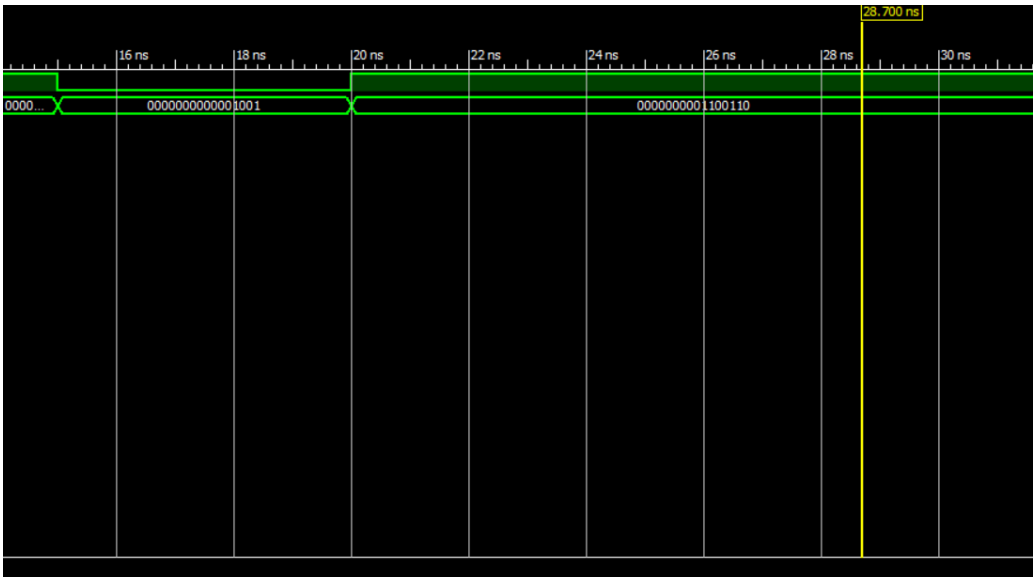
باتوجه به محدوده اعداد به سادگی می‌توان فهمید که اگر $(b + d - a - c) = -11 \vee 0 \vee 11$ باشد عدد بر ۱۱ بخش پذیر است و در غیر این صورت بخش‌پدیر نیست. به جای تفریق کردن ما مقادیر $x = b + d$ و $y = a + c$ را محاسبه می‌کنیم (به کمک یک $FullAdder$) و سپس به کمک یک مقایسه‌گر (که در ماژول $five\_bit\_comprator$ نوشته شده است) ابتدا x و y و سپس $y + 11$ و سپس $x + 11$ و y را مقایسه می‌کنیم و در صورتی که حتی یکی از این مقایسه‌گرها اعلام برابری کند می‌گوییم که عدد بر ۱۱ بخش‌پذیر است. (همچنین دقت کنید که مقایسه‌گر صرفا می‌گوید که آیا اعداد ورودی برابر هستند یا خیر)

گزارش برنامه برای تعداد LUT به شکل زیر است:

Slice Logic Utilization:
Number of Slice LUTs: 38 out of 63400 0%
Number used as Logic: 38 out of 63400 0%

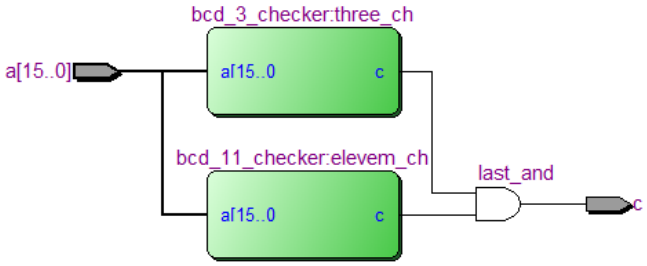
نتایج برنامه برای شبیه‌سازی در زیر آمده است (عکس‌ها در صورت نیاز برای بررسی بیشتر پیوست شده‌اند) :



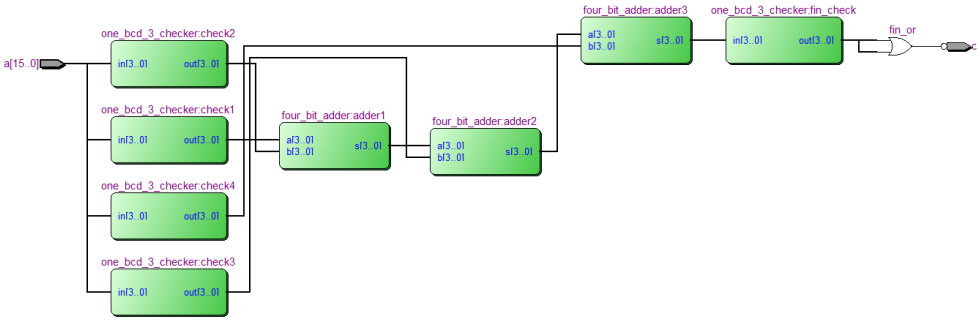


فایل کد اصلی در فایل $checker.v$ قرار دارد و تست بنچ فایل ماژول در $checker\_tb.v$ قرار دارد. تمامی تست بنچ‌ها در فایل $checker\_tb\_all.v$ قرار دارند.
در ادامه نیز شکل مدار قرار دارد، به علت شلوغی کد برای نمایش شکل از بلوک‌بندی استفاده کرده‌ایم:

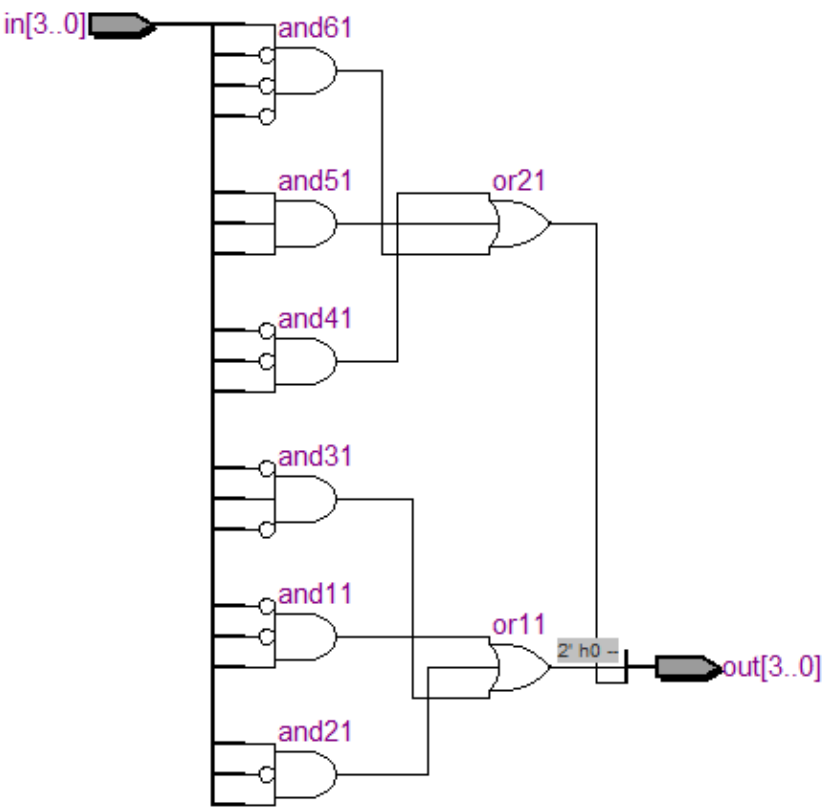
ماژول *checker*:



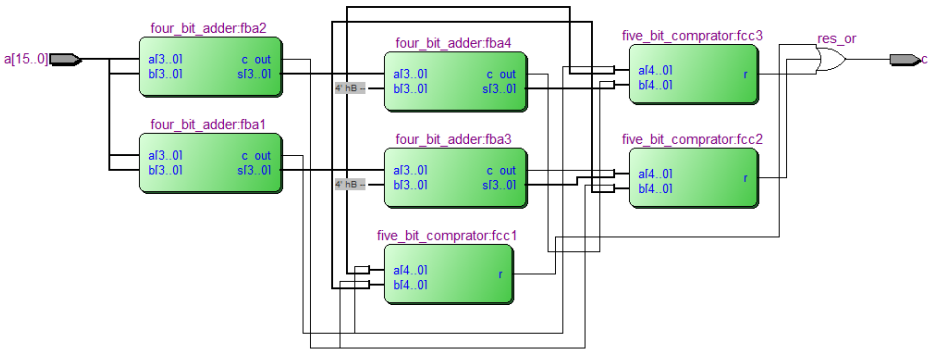
ماژول *bcd_3_checker*:



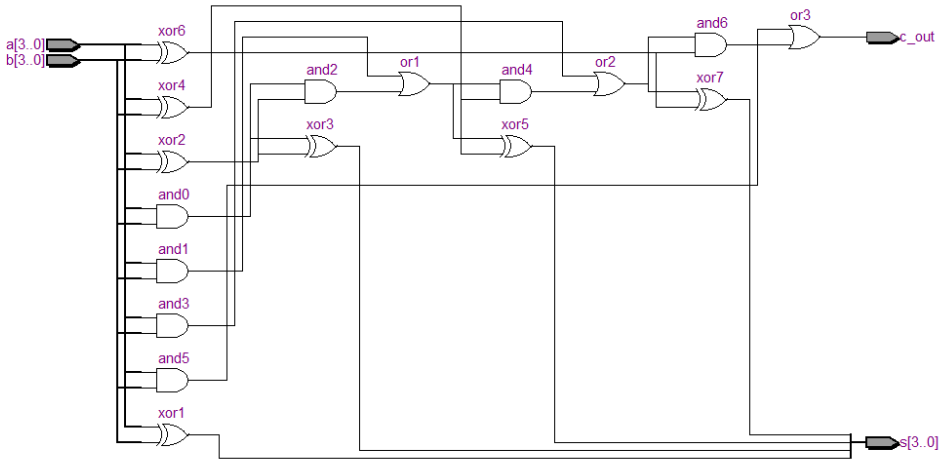
ماژول *one_bcd_3_checker*:



ماژول $bcd\_11\_checker$:



ماژول $four\_bit\_adder$:



ماژول $five\_bit\_comprator$: