



سیستم زمان‌بندی تمرین‌های ورزشی

۱- مقدمه

سلامتی یکی از ارزشمندترین نعمت‌هایی است که هر فردی می‌تواند داشته باشد و ورزش منظم نقش کلیدی در حفظ و بهبود آن ایفا می‌کند. تمرین‌های ورزشی نه تنها باعث افزایش تناسب اندام و تقویت سیستم ایمنی بدن می‌شوند، بلکه در کاهش استرس، بهبود خلق و خو و افزایش سطح انرژی روزانه نیز مؤثرند. با این حال، یکی از چالش‌های اصلی در انجام تمرین‌های ورزشی، مدیریت صحیح زمان و پایداری به برنامه‌های تمرینی است. اینجاست که سیستم‌های دیجیتال هوشمند می‌توانند به کمک ما بیایند. فناوری‌های هوشمند حوزه سلامت با ترکیب دانش پزشکی و پیشرفت‌های فناوری دیجیتال، انقلابی در مراقبت‌های بهداشتی ایجاد کرده‌اند. این فناوری‌ها که شامل سیستم‌های پوشیدنی، نرم‌افزارهای نظارت بر سلامت، دستگاه‌های تشخیصی هوشمند و سیستم‌های مدیریت تمرین می‌شوند، امکان نظارت دقیق‌تر، پیشگیری مؤثرتر و درمان شخصی‌سازی شده را فراهم می‌کنند.

این پروژه با تمرکز بر طراحی سیستم دیجیتال زمان‌بندی تمرین‌ها روی FPGA، گامی به سوی توسعه فناوری‌های هوشمندی است که می‌توانند در ارتقای سلامت فردی و اجتماعی نقش مؤثری ایفا کنند. پیاده‌سازی این سیستم روی FPGA برای دانشجویان فرصتی ارزشمند است تا دانش نظری خود در زمینه طراحی دیجیتال و سیستم‌های نهفته را به صورت عملی به کار گیرند. این پروژه به آن‌ها امکان می‌دهد با مفاهیم مهمی مانند طراحی سلسله‌مراتبی، ماشین‌های حالت محدود، زمان‌بندی دیجیتال و رابط‌های ورودی/خروجی به صورت ملموس کار کنند و مهارت‌های توصیف سخت‌افزار با HDL را در یک پروژه کاربردی توسعه دهند. از طرفی، کار با FPGA دانشجویان را با چالش‌های دنیای واقعی مانند بهینه‌سازی منابع، ستر دیجیتال و مدیریت زمان مواجه می‌سازد که برای ورود به بازار کار صنعت الکترونیک دیجیتال ضروری است. همچنین، ماهیت انعطاف‌پذیر FPGA این امکان را فراهم می‌کند تا دانشجویان بتوانند قابلیت‌های پروژه را به تدریج گسترش دهند و خلاقیت خود را در طراحی سیستم‌های دیجیتال به نمایش بگذارند.

یک تمرین ورزشی چند دقیقه‌ای با شدت متوسط تا بالا می‌تواند شامل چند حرکت ورزشی مانند پروانه، درازونشست، شنای سوئدی، اسکات، پلانک و ... با ۳۰ ثانیه اجرای هر حرکت و ۱۰ ثانیه استراحت بین آن‌ها باشد. شکل ۱ نمونه‌ای از این تمرین را نشان می‌دهد. این تمرین با شدت مناسب می‌تواند ضربان قلب را به ۶۰ تا ۸۰ درصد حداکثر ظرفیت برساند و حدود ۱۰۰ تا ۱۵۰ کالری بسوزاند که برای بهبود استقامت قلبی-عروقی، تقویت عضلات و افزایش سوخت‌وساز بدن مؤثر است. چنین تمرین‌های کوتاه‌مدت اما پرتکرار، به‌ویژه برای افراد پر مشغله، راه‌حلی عملی برای حفظ تناسب اندام و سلامت عمومی محسوب می‌شود.



شکل ۱ - نمونه‌ای از حرکات‌های ورزشی روزانه

توجه: کلیه مطالب پروژه در مورد سلامت، ورزش و فرمول‌های ذکر شده، صرفاً جهت انجام یک پروژه مقدماتی مطرح و ساده سازی شده‌اند و از نظر علمی موثق نیستند. لذا جهت کسب اطلاعات دقیق و قابل اعتماد در این خصوص می‌بایست به منابع معتبر یا متخصصان این حوزه مراجعه نمایید.

یکی از اطلاعات رایجی که وسایل هوشمند ورزشی ارائه می‌دهند نمایش میزان کالری مصرفی در یک تمرین ورزشی است. کالری مصرفی به مدت زمان تمرین، شدت فعالیت (میزان ضربان قلب در طول ورزش)، وزن و جنسیت فرد وابسته است. برای تعیین مدت زمان مورد نیاز جهت سوزاندن مقدار مشخصی کالری، می‌توان از فرمول زیر استفاده کرد.

$$T = \frac{Cal \times 60}{MET \times W} \times G \quad (1)$$

فرمول (۱) مدت زمان مورد نیاز (T) بر حسب دقیقه برای سوزاندن مقدار مشخصی کالری (Cal) را محاسبه می‌کند. در این فرمول MET نشان‌دهنده شدت فعالیت (مثلاً ۴ برای پیاده‌روی، ۶ برای شنا و ۸ برای دویدن)، W وزن فرد بر حسب کیلوگرم و G ضریب تصحیح جنسیت است. ضریب جنسیت به دلیل تفاوت‌های متابولیکی

بدن برای زنان حدود ۱/۱ و برای مردان برابر ۱ در نظر گرفته می‌شود. برای مثال، یک زن ۶۰ کیلوگرمی ($G=1.1$) برای سوزاندن ۱۰۰ کالری از طریق پیاده روی ($MET=4$)، به $T = \frac{100 \times 60}{4 \times 60 \times 0.9} = 27.8$ دقیقه تمرین نیاز دارد. توجه شود که نتایج تقریبی هستند و عوامل دیگری مانند سن و سطح آمادگی نیز تأثیر گذارند.

۲- مشخصات پروژه

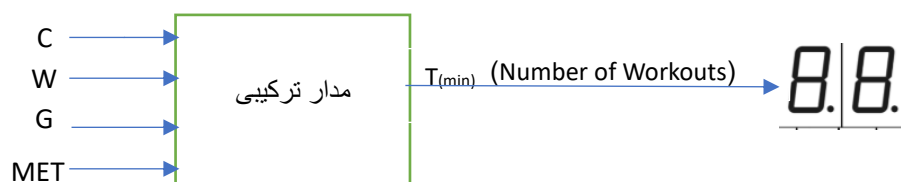
این پروژه در سه فاز (مرحله) اجرا می‌شود: در فاز طراحی، ابتدا معماری سیستم شامل واحد محاسبه و واحد کنترل (ماشین حالت) طراحی می‌شود. در فاز دوم شبیه‌سازی عملکرد منطقی سیستم تحت تست‌های جامع (شامل تنظیم پارامترهای مختلف و سناریوهای تمرینی) انجام شده و با ابزارهای شبیه‌سازی اعتبارسنجی می‌شود. در آخر، در فاز پیاده‌سازی، مدار طراحی شده همراه با رابط‌های کاربری مورد نیاز (نظیر نمایشگرهای دیجیتال، کلیدها و سیستم هشدار صوتی)، پس از فرآیند سنتز و بهینه‌سازی منابع، روی برد هدف (FPGA) برنامه‌ریزی و پیاده‌سازی می‌شود.

۲-۱- فاز طراحی

در این پروژه می‌خواهیم ابتدا با گرفتن اطلاعات کاربر، بر اساس فرمول (۱) مدت زمان مورد نیاز برای سوزاندن کالری مد نظر کاربر را تخمین زده و نمایش دهیم، سپس تعدادی تمرین ورزشی یک دقیقه‌ای را با شدت فعالیت مد نظر کاربر زمان‌بندی کنیم. فاز طراحی پروژه شامل دو بخش واحد محاسبه (طراحی مدار ترکیبی) و واحد زمان‌بندی (طراحی ماشین حالت) است.

۲-۱-۱ واحد محاسبه

در بخش محاسبه می‌خواهیم یک مدار ترکیبی مانند شکل ۲ طراحی کنیم که وزن (W)، میزان کالری مدنظر (Cal)، شدت فعالیت (MET) و جنسیت (G) فرد را بگیرد و از رابطه (۱) مدت زمان مورد نیاز برای تمرین را محاسبه کند، سپس تعداد و زمان هر حرکت ورزشی را نمایش دهد.



شکل ۲- مدار محاسبه تعداد و زمان حرکات ورزشی

برای سادگی طراحی مدار ترکیبی مفروضات زیر را برای ورودی‌ها در نظر بگیرید:

- Cal دو بیتی است و کالری مورد نظر کاربر را بین ۵۰، ۱۰۰، ۱۵۰ و ۲۰۰ مشخص می‌کند.
- MET دو بیتی است و شدت فعالیت را بین ۱، ۲، ۴ و ۸ مشخص می‌کند.
- ضریب جنسیت به صورت داده یک بیتی است که اگر صفر باشد، ضریب جنسیت $G=1$ (برای مردان) و اگر یک باشد ضریب جنسیت برابر $G=1.125$ (برای زنان) در نظر گرفته می‌شود.
- W سه بیتی است و طبق جدول زیر وزن بین ۵۰ تا ۱۲۰ کیلوگرم را مشخص می‌کند.

جدول ۱- مقادیر ممکن ورودی‌های Cal، W و MET

Cal	کالری هدف
00	50
01	100
10	150
11	200

MET	شدت فعالیت
00	1
01	2
10	4
11	8

W	وزن (کیلوگرم)
000	50
001	60
010	70
011	80
100	90
101	100
110	110
111	120

با این مفروضات ورودی سیستم در مجموع ۸ بیت خواهد بود. بعنوان مثال برای یک کاربر زن ($G=1.125$) با وزن ۶۰ کیلوگرم و با هدف سوزاندن ۱۰۰ کالری در یک تمرین با شدت نسبتاً شدید ($MET=4$)، اطلاعات ورودی به صورت زیر است و خروجی در این حالت $T=28$ دقیقه خواهد بود.

W	Cal	MET	G
0	0	1	0
0	1	1	0
1	0	1	1

فرض بر این است که تمرین‌های ما شامل تعدادی تمرین یک دقیقه‌ای پشت سر هم هستند. بنابراین بعد از محاسبه زمان کل تمرین‌ها بر حسب دقیقه (T)، تعداد T تمرین یک دقیقه‌ای را باید زمان‌بندی کنیم. هر تمرین شامل یک حرکت ورزشی ۴۵ ثانیه‌ای و ۱۵ ثانیه استراحت بعد از آن است.

محدودیت: در پیاده سازی مدار ترکیبی نباید از توصیف رفتاری و عملگرهای ضرب و تقسیم در Verilog استفاده کنید و مدار باید به صورت ساختاری (در سطح گیت و جریان داده) توصیف شود.

راهنمایی: برای پیاده سازی مدار بدون استفاده از واحدهای ضرب و تقسیم، می توان از ترکیبی از عملگر شیف، مدار مالتی پلکسر و پیاده سازی بر اساس جدول درستی به شرح زیر استفاده کرد.

برای این منظور فرمول را به صورت زیر به سه بخش تقسیم می کنیم:

$$T = \frac{Cal \times 60}{MET \times W \times G} = \frac{Cal \times 60}{W} \times G \times \frac{1}{MET}$$

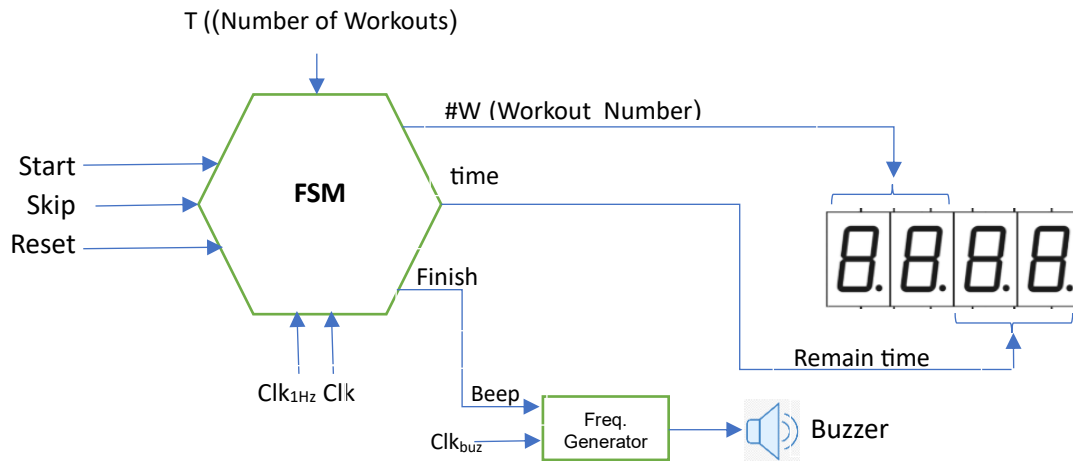
برای محاسبه بخش اول ($\frac{Cal \times 60}{W}$)، یک جدول درستی رسم می کنیم و تمام مقادیر ممکن ورودیهای W و Cal در آن می نویسیم. چون این دو ورودی با هم ۵ بیت هستند جدول درستی ۳۲ سطر خواهد داشت. سپس برای هر سطر جدول، مقدار $\frac{Cal \times 60}{W}$ را محاسبه کرده، آنرا به باینری تبدیل و بعنوان خروجی در جدول قرار می دهیم. بعد هر بیت (ستون) خروجی را با جدول کارنو ساده کرده و مدار ترکیبی را بدست می آوریم.

برای محاسبه بخش دوم ($G \times$)، می توان از یک مالتی پلکسر 2x1 برای انتخاب ضریب ۱ (مردان) یا ضریب ۱/۱۲۵ (برای زنان) استفاده کرد. برای محاسبه ضریب ۱/۱۲۵ (معادل یک و یک هشتم)، می توان خروجی بخش قبل را با سه بار شیف به راست خودش جمع کرد.

برای محاسبه بخش سوم ($\times \frac{1}{MET}$) نیز با توجه به اینکه MET چهار مقدار ممکن می تواند داشته باشد که همه توانهای ۲ هستند، می توان از مالتی پلکسر 4x1 و انتخاب شیف به راست های مختلف از خروجی قبل استفاده کرد.

۲-۱-۲ بخش زمان بندی ترتیبی

در این بخش یک ماشین حالت محدود (FSM) طراحی می شود که تعدادی حرکت ورزشی با زمان مشخص و استراحت بین آنها را زمان بندی کرده و پایان هر حرکت را به کاربر اعلام کند. در شکل ۳ ورودی ها و خروجی های ماشین حالت رسم شده اند؛ این ماشین حالت سه ورودی یک بیتی حساس به لبه، به نام های Start، Skip و Reset دارد. در حالت اولیه سیستم، با زدن دکمه Start زمان بندی تمرین ها شروع می شود. در حین تمرین های ورزشی کاربر می تواند با زدن دکمه Skip حرکت فعلی صرف نظر و به حرکت بعد برود و با زدن دکمه Reset نیز در هر حالتی سیستم به حالت اولیه برمی گردد.



شکل 3 - ماشین حالت

در هر حالت باید شماره تمرین فعلی و زمان باقیمانده حرکت ورزشی فعلی (با شمارنده کاهشی با فرکانس ۱ هرتز) روی 7-segment ها نمایش داده شود و بعد از پایان زمان حرکت، زمان استراحت به صورت شمارنده کاهشی نمایش داده می شود. پس از پایان زمان استراحت، آلام (صدای beep از Buzzer) به نشانه پایان تمرین فعلی به صدا در آمده و پس از آن تمرین بعدی شروع می شود. بعد از پایان همه تمرین ها سیستم متوقف شده و پایان تمرین ها صدای beep با فرکانس متفاوت اعلام می شود.

گزارش فاز ۱: در پایان فاز طراحی، برنامه Verilog توصیف کننده مدارهای طراحی شده را نوشته و همراه گزارش آن (شامل رسم کامل مدار منطقی ترکیبی و رسم کامل ماشین حالت، برنامه ها و توضیح آنها) در سامانه دروس (courses) بارگذاری کنید.

۲-۲- فاز شبیه سازی

در این مرحله با نوشتن برنامه تست مناسب (Testbench)، صحت عملکرد بخش محاسبه فاز ۱ را بررسی کرده و در صورت لزوم آن را اصلاح می کنیم. برنامه تست باید به گونه ای نوشته شود که سناریوی تست را از فایل inputs.txt خوانده و به مدار وارد کند و سپس خروجی های مدار متناظر با سناریوی تست در فایل output.txt ذخیره نماید (مثالی از روش خواندن و نوشتن فایل های متنی در Verilog در پیوست ۱ آمده است).

نمونه یک سناریوی تست و خروجی آن در جدول ۲ نشان داده شده‌اند. هنگام تحویل پروژه ممکن است سناریوی تست متفاوت باشد و مدار باید برای همه مقادیر ممکن ورودی جواب صحیح را محاسبه کند.

جدول ۲-نمونه‌ای سناریوی تست و خروجی آن برای مدار محاسبه شکل ۲

input.txt	output.txt	
{W, Cal, MET, G}	Inputs	outputs
00101101	W=60, Cal=100, MET=4, G=1.125	=> T=28min
01001000	W=70, Cal=100, MET=1, G=1	=> T=85min
00000010	W=50, Cal=50 , MET=2, G=1	=> T=30min
10011111	W=90, Cal=200, MET=8, G=1.125	=> T=18min
01110100	W=80, Cal=150, MET=4, G=1	=> T=28min
...	...	

دانشجویان برای شبیه‌سازی می‌توانند از شبیه‌ساز ISIM در ISE (مورد استفاده در طول ترم) یا هر ابزار شبیه‌سازی دیگر استفاده کنند. برنامه Testbench فاز ۲، باید همزمان با تحویل نهایی پروژه در سامانه دروس نیز بارگذاری شوند.

۲-۳ - فاز پیاده سازی

در این مرحله بخش‌های ورودی/خروجی به بلوک‌های طراحی شده قبل، متصل شده و پروژه قابل سنتز و پیاده سازی روی FPGA خواهد شد. ورودی/خروجی‌های سیستم و مدارهای مورد نیاز آنها به صورت زیر هستند:

- کلاک (Clock): یکی از ورودی‌های مهم سیستم کلاک ساعت است، سیستم طراحی شده احتمالاً به کلاک با فرکانس متفاوت نیاز دارد (برای زمان‌بندی شمارنده، برای نمایشگر دیجیتال، برای آلارم و ..). بنابراین ماژول تقسیم فرکانسی برای تولید کلاک‌های مختلف مورد نیاز سیستم باید پیاده‌سازی شود. فرکانس کلاک اصلی روی بُرد آزمایشگاه 40Mhz (روی پایه P184) است.
- نمایشگر 7-segment چهار رقمی: اطلاعات تعداد و زمان تمرین‌ها در حالت اولیه سیستم و پس از آن زمان باقیمانده هر تمرین روی نمایشگر ۴ رقمی نمایش داده می‌شود. نمایشگر 7-segment روی بُرد آزمایشگاه، شامل ۴ عدد 7-segment کاتد (Cathode) مشترک است و برای نمایش یک عدد چهار رقمی به صورت همزمان روی این نمایشگر، باید از روشی با نام مالتی پلکس ارقام (پیوست ۲) استفاده شود. این قسمت را به صورت یک ماژول نوشته و تست کنید، سپس آن را به سیستم اصلی متصل نمایید.

- سوئیچ‌ها و دکمه‌های فشاری: ورودی‌های تک بیتی سیستم شامل ۹ سوئیچ (Switch) برای ورود اطلاعات کاربر (وزن، کالری و ..) و سه دکمه فشاری (Push Button) برای ورودی‌های FSM هستند، دکمه‌های فشاری برای رفع اثر لرزش دست نیاز به مدار لرزش گیر (Denouncer) دارند، در غیر اینصورت همراه ورودی‌ها، داده‌های تصادفی وارد سیستم خواهد شد. در پیوست ۳ یک مدار لرزش گیر معرفی شده است.
- بازر (Buzzer): یکی از خروجی‌های دیگر سیستم بازر است که برای تولید صدای beep در پایان هر تمرین استفاده می‌شود. بازر روی بُرد آزمایشگاه از نوع پسیو (Passive) بوده و به پایه P13 تراشه FPGA متصل است. برای ایجاد صدا روی بازر، باید یک پالس مربعی با فرکانس در محدوده شنوایی انسان تولید کرد و به پایه بازر فرستاد. فرکانس قابل شنیدن برای beep بین 500Hz تا 2kHz است. فرکانس‌های پایین‌تر (محدوده 200-500Hz) صدای بم‌تر و فرکانس‌های بالاتر (محدوده 2-3kHz) صدای زیرتر شبیه بوق تولید می‌کنند. برای ایجاد صدای beep با مدت زمان مشخص (مثلاً ۱ ثانیه روشن و ۱ ثانیه خاموش)، از یک ماژول زمان‌سنج با زمان یک ثانیه می‌توان استفاده کرد.

۲-۴ - بخش اختیاری

(انجام این بخش اختیاری بوده و با تحویل آن همراه با تحویل پروژه، ۲۰ درصد به نمره پروژه اضافه خواهد شد).

قسمتی به پروژه اضافه کنید که همزمان با هر تمرین ورزشی، نام حرکت ورزشی و زمان باقیمانده را روی LCD کاراکتری موجود روی بُرد FPGA نمایش دهد. نام ۸ حرکت ورزشی رایج در شکل ۱ نشان داده شده است، اگر تعداد تمرین‌ها بیشتر بود بصورت دوره این ۸ حرکت تکرار شوند. نوع قطعه نمایشگر LCD موجود روی بُرد آزمایشگاه، lcd 16L است که دارای دو ردیف ۱۶ کاراکتری برای نمایش کاراکترهای اسکی است. برای کسب اطلاعات بیشتر می‌توانید از نمونه پروژه لینک زیر کمک بگیرید.

<https://github.com/josh-macfie/FPGAtoLCD>

۳- زمان بندی تحویل

جدول ۳- زمان بندی انجام و تحویل پروژه

تاریخ	فعالیت	موارد تحویلی	نمره از ۱۰۰
هفته اول خرداد	- توضیح پروژه توسط مدرس آزمایشگاه - پاسخ به سوالات در خصوص پروژه ^۱	-	-
دوشنبه ۱۲ خرداد	تحویل گزارش فاز ۱	- بارگذاری گزارش فاز ۱ در کورسز	۵۰
دوشنبه ۲ تیر	تحویل فاز ۲ و ۳	- تحویل حضوری شبیه سازی و سنتر پروژه ^۳ - بارگذاری فایل ۷ در کورسز	۵۰

^۱ به غیر از کلاس های هفته اول خرداد، جلسه پرسش و پاسخ در روز دوشنبه ۵ خرداد ساعت ۴/۵ تا ۶ عصر با حضور مدرسان آزمایشگاه برگزار خواهد شد، دانشجویان اگر سوالی در مورد پروژه یا روش حل آن دارند می توانند در این جلسه شرکت کنند.
^۲ کانال تلگرامی با آدرس t.me/DLDDLab با حضور همه مدرسان، جهت پاسخ به سوالات احتمالی دانشجویان تشکیل شده است.
^۳ زمان دقیق تحویل حضوری برای هر تیم از طریق سامانه درس ها اطلاع رسانی می شود.

نکات مهم

- پروژه به صورت تیم های حداکثر دو نفره انجام می شود ولی به صورت تک نفره تحویل گرفته خواهد شد (هر دو نفر باید آمادگی کامل برای پاسخگویی به سوالات را داشته باشند).
- در صورت عدم اشراف دانشجو به بخشی از پروژه تحویلی و عدم توانایی پاسخگویی به سوالات، نمره کل پروژه برای آن دانشجو صفر خواهد شد.
- گزارش فاز ۱ می تواند دست نویس یا تایپی باشد، گزارش در یک فایل pdf با نام **Ggr_ID.pdf** که در آن **gr** شماره گروه آزمایشگاه و **ID** شماره دانشجویی اعضای تیم است، در سامانه کورسز بارگذاری شود (مثال **G03_402232301_403131910.pdf**). به ازای هر روز تاخیر در ارسال گزارش فاز ۱، حداکثر تا ۵ روز، ۱۰ درصد نمره کسر خواهد شد.
- همزمان با تحویل نهایی پروژه، همه برنامه های Verilog نوشته شده (از جمله ماژول تست) را در یک فایل ۷. با نام **Ggr_ID.۷** در سامانه کورسز بارگذاری کنید. برای زمان تحویل نهایی هیچ تاخیری در بارگذاری فایل قابل قبول نیست.
- **بسیار مهم:** برنامه های دریافتی از همه گروه های آزمایشگاه با ابزارهای هوش مصنوعی شباهت سنجی خواهند شد. در صورت شباهت بالا در ساختار یا منطق برنامه با تیم های دیگر یا استفاده از مولدهای هوش مصنوعی (مانند ChatGPT و ...)، نمره پروژه صفر خواهد شد.

با آرزوی موفقیت، عباس نیک آبادی، بهار ۱۴۰۴

پیوست ۱ – ورودی/ خروجی به فایل متنی در Verilog

برنامه تست زیر داده‌های ۴ بیتی فایل ورودی input.txt را به صورت خط به خط خوانده، دو بیت کوچکتر و بزرگتر داده هر خط را توسط نمونه سازی ماژول adder_2bit با هم جمع کرده و حاصل جمع ۳ بیتی حاصل را در فایل خروجی output.txt می‌نویسد.

```
`timescale 1ns / 1ps
module testfile( );
    integer fin, fout, tmp , i=0;
    reg [3:0] data_in;
    wire [2:0] data_out;

    //instantiation
    adder_2bit f1(
        .out(data_out),
        .in1(data_in[3:2]),
        .in2(data_in[1:0])
    );

    initial begin
        fin = $fopen("input.txt", "r");
        fout = $fopen("output.txt", "w");

        while (!$feof(fin)) begin
            tmp=$fscanf(fin, "%4b\n", data);
            i=i+1;
            $fdisplay(fout, "line = %2d : %3b , ", i, data_out);
        End

        $fclose(fin);
        $fclose(fout);
        $finish;
    end
endmodule
```

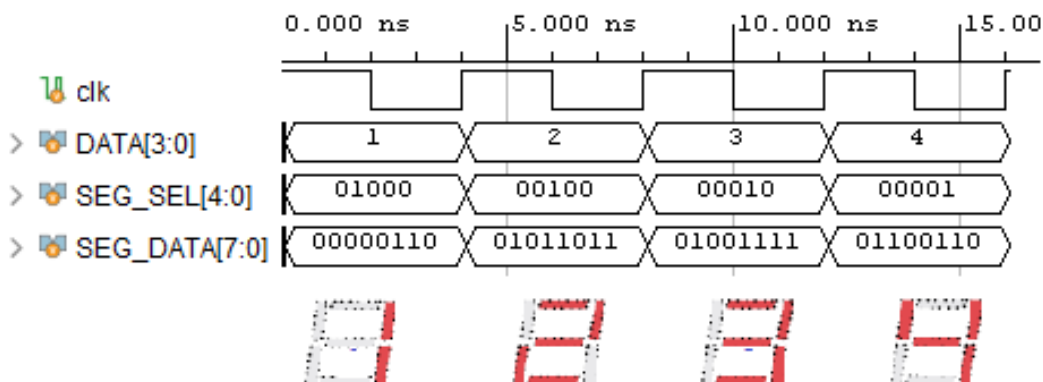
پیوست ۲ - مالتی پلکس ارقام

روش مالتی پلکس ارقام (Digit Multiplexing) یکی از تکنیک‌های رایج برای نمایش ارقام چند رقمی روی نمایشگرهای 7-segment است. در این روش، به جای روشن کردن همه سگمنت‌ها به طور همزمان، تنها یک رقم در هر لحظه روشن می‌شود و به سرعت بین ارقام مختلف سوئیچ می‌شود تا به چشم انسان این طور به نظر برسد که همه ارقام همزمان نمایش داده می‌شوند. نمایشگر مورد آزمایشگاه (AVA3S400) طبق شکل زیر از ۵ بخش (چهار رقم 7-segment و کاراکتر :) تشکیل شده است. خطوط داده در هر چهار 7-segment مشترک بوده و کاتدهای آنها به طور مجزا قابل مقداردهی است. با یک کردن کاتد ارقام می‌توانیم مشخص کنیم داده ارسالی در کدام رقم (7-segment) نمایش داده شود.



شکل ۴- نمایشگر ۴ رقمی پورده AVA3S400

برای نمایش یک عدد ۴ رقمی با روش مالتی پلکس ارقام، هر کدام از ۴ رقم به صورت جداگانه با فاصله زمانی کوتاه نمایش داده می‌شوند (در هر لحظه فقط یکی از ارقام روشن است). با سوئیچینگ سریع بین ارقام با سرعت حدود ۶۰ هرتز تا ۵۰۰ هرتز (یعنی هر ۲ میلی ثانیه تا ۱۶ میلی ثانیه یک رقم تغییر می‌کند)، چشم انسان این تغییرات را به صورت یک نمایش ثابت مشاهده می‌کند. این روند برای تمام ۴ رقم به طور مکرر تکرار می‌شود (ابتدا رقم اول روشن می‌شود و کاتد رقم اول فعال می‌شود، سپس رقم دوم، سوم و چهارم به طور پیوسته نمایش داده می‌شوند. بعد از نمایش چهارم، دوباره به اولین رقم برمی‌گردد و این روند ادامه می‌یابد). شکل زیر زمان‌بندی یک دور تکرار برای نمایش عدد 1234 را نشان می‌دهد. در این شکل DATA داده اصلی، SEG_SEL سیگنال انتخاب کاتد 7-segment ها برای نمایش یک رقم و SEG_DATA داده‌های لازم برای روشن شدن LEDهای 7-segment برای نمایش رقم مورد نظر است.

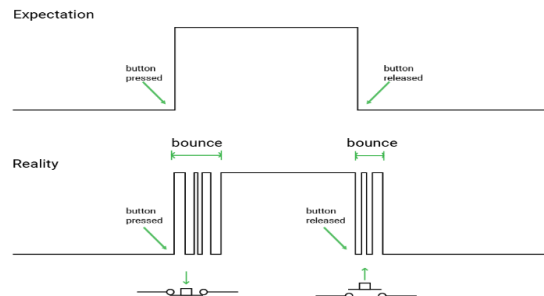


شکل ۵- زمان‌بندی یک دور به روزرسانی برای نمایش عدد 1234

بنابراین برای نمایش اعداد روی نمایشگر ۴ رقمی، باید مازولی بنویسید که به صورت تکراری با فرکانس ۶۰ تا ۵۰۰ هرتز ارقام آن عدد را توسط مازول مبدل BCD به 7-segment (آزمایش ۹) به پایه‌های نمایشگر ارسال کند.

پیوست ۳ - رفع لرزش کلید

لرزش کلید (Switch Bounce) یک پدیده فیزیکی است که در هنگام فشردن یا رها کردن یک کلید مکانیکی (مانند دکمه یا سوئیچ) اتفاق می‌افتد. هنگامی که کلید فشرده می‌شود، تماس الکتریکی بین دو پین ممکن است چندین بار به صورت بسیار سریع قطع و وصل شود.

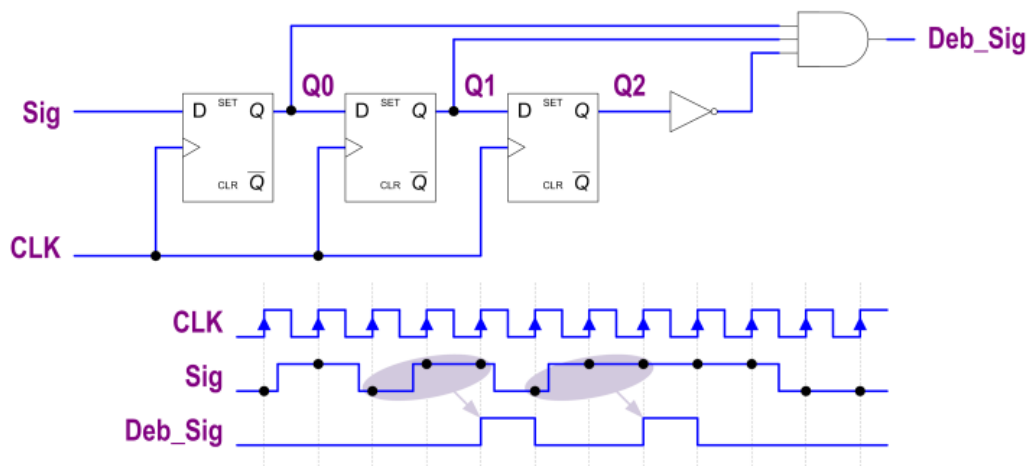


شکل ۶ - اثر لرزش کلید

لرزش کلید باعث می‌شود که سیگنال‌های دیجیتال که باید تنها یک تغییر واحد (یک "۰" یا "۱") را نشان دهند، چندین بار تغییر کنند. این تغییرات ناپایدار می‌تواند باعث ایجاد نویز در مدار شود و ممکن است باعث اشتباهات منطقی به خصوص در در مدارهای شمارش گر (که در آنها برای هر فشار کلید باید تنها یک تغییر رخ دهد) شود.

برای مقابله با اثر لرزش کلید، از روش‌های مختلفی می‌توان استفاده کرد که از جمله این روش‌ها می‌توان (۱) استفاده از مدارهای RC در کلید ورودی، (۲) استفاده از مدارهای مقایسه کننده (Schmitt Trigger، ۳) تاخیر و توقف چند میلی‌ثانیه‌ای پس از ورودی و (۴) استفاده از مدارهای منطقی را نام برد.

مدارهای منطقی لرزش گیر کلید معمولاً از یک فیلتر یا تاخیر دهنده استفاده می‌کنند تا از ثبت لرزش‌های سریع و موقتی جلوگیری و تنها تغییرات پایدار را شناسایی کند. شکل زیر یک نمونه از این مدارها را نشان می‌دهد که از ۳ فلیپ فلاپ D و یک گیت AND سه ورودی ساخته شده است.



شکل ۷ - یک نمونه مدار debouncer سخت افزار