دانشکده مهندسی کامپیوتر

دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)

پروژه‌ی پایانی درس مدارهای منطقی

Fighting Game

اساتید درس:
جناب دکتر صاحب‌الزمانی
جناب دکتر صدیقی

طراحی پروژه:
امیرحسین کاشانی
فرزانه ارزاقی

دی ماه ۱۴۰۲

توضیحات

۱. پروژه به صورت گروهی (گروه‌هایی که در آزمایشگاه پیش از این داشته‌اید) تحویل گرفته می‌شود. اگر کسی مشکلی با گروه‌بندی فعلی خود دارد با داشتن پیشنهاد جایگزین به مدرسین درس پیام دهد. (جابه‌جایی افراد گروه بهتر است بین دانشجویان همان کلاس انجام شود زیرا هنگام ارائه شناخت مدرس آزمایشگاه از دانشجویان کلاس خود در ارزیابی صحیح ایشان تاثیرگذار است. همچنین هماهنگی با سایر افراد گروه جهت جابه‌جایی و تغییر گروه بر عهده خود دانشجو می‌باشد.)

۲. هنگام ارائه مدرسین از هر دانشجو به صورت جداگانه سوال می‌پرسند پس همه افراد گروه باید در انجام پروژه همکاری کنند و تسلط کامل داشته باشند.

۳. نمره هر عضو گروه با توجه به ارائه‌اش جداگانه محاسبه می‌شود. به طور مثال ممکن است یک نفر تلاش زیادی برای انجام پروژه کرده باشد و کاملا بر روی تمام بخش‌های پیاده‌سازی مسلط باشد در مقابل عضو دیگری از گروه تسلط چندانی نداشته باشد به همین دلیل نمرات هر فرد پس از ارائه جداگانه محاسبه می‌شود و اعضای گروه نمره مشابه نمی‌گیرند.

۴. روز سه‌شنبه ۰۲/۱۱/۱۰ برای تحویل پروژه مشخص شده است.

۵. اینکه هر گروه چه ساعتی برای تحویل پروژه به آزمایشگاه مراجعه کند توسط مدرس هر آزمایشگاه به دانشجویان آن کلاس اطلاع داده خواهد شد.

۶. تحویل پروژه فقط به صورت حضوری پذیرفته است.

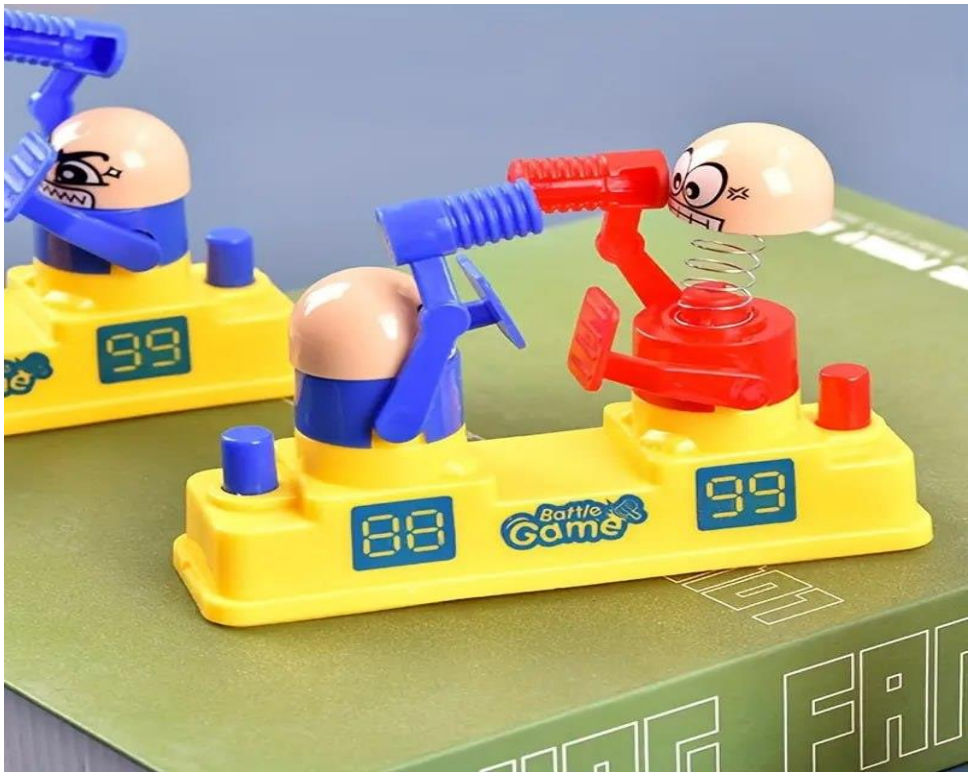
۷. در صورتی که در هر بخش از پروژه ابهام یا سوالی داشتید در گروه درس سوال خود را بپرسید.

۸. اگر مایل به انجام بخش امتیازی بودید در ساعاتی که آقای خلیلی‌خو حضور دارند به آزمایشگاه بروید و از ایشان برد برای پیاده‌سازی خود بگیرید.

مقدمه

در این پروژه قصد داریم با استفاده از زبان توصیف سخت‌افزار یک بازی مبارزه ساده طراحی کنیم. در این بازی دو بازیکن مقابل یکدیگر قرار می‌گیرند و با استفاده از یک سری اکشن تعریف شده شروع به مبارزه می‌کنند. در ابتدا هر بازیکن تعدادی جان در اختیار دارد. پس از هر اکشن که به بازیکن برخورد کند تعداد جان مشخصی از او کم خواهد شد. در نهایت بازیکنی که زودتر جان‌های خود را از دست بدهد بازنده خواهد بود. دانشجویان باید منطق این بازی را به شکلی که در بخش‌های بعدی توضیح داده شده است، پیاده‌سازی کنند سپس سناریوای از بازی را بر روی کد خود اجرا کنند. این سناریو به دانشجویان داده شده است و کافی است اکشن خواسته شده برای هر یک از بازیکنان اجرا شود.

در ادامه جزئیات مورد نیاز برای پیاده‌سازی این بازی قرار داده‌شده‌است. برای گرفتن نمره این پروژه شبیه‌سازی آن با استفاده از نوشتن تست‌بنچ کافی است. اگر به دنبال نمره امتیازی هستید می‌توانید به سراغ بخش امتیازی پروژه بروید و بازی را بر روی برد FPGA پیاده‌سازی کنید.



شرح بازی

در این بازی دو بازیکن مقابل یکدیگر قرار دارند و به مبارزه می‌پردازند. فضای حرکت دو بازیکن ۶ بلاک هست و کارکترها نمی‌توانند از این ۶ بلاک خارج شوند. در این بازی هر کارکتر می‌تواند به چپ و راست حرکت کند، با حرکت پرش در مقابل ضربه حریف جاخالی دهد و یا بدون هیچ اکشن و حرکتی ثابت بایستد. همچنین هر بازیکن با دو حرکت punch و kick می‌تواند به حریف حمله کند. اجرای هر کدام از این حرکات دارای شرایط مخصوص به خود و اولویت جداگانه است که در بخش قوانین مربوط به ضربه‌ها توضیح داده شده است. در ابتدای بازی هر بازیکن دارای ۳ جان است که در طول بازی و با برخورد هر یک از ضربات حریف تعدادی از جان‌ها از بین می‌رود. در نهایت بازیکنی که زودتر جان خود را از دست بدهد بازنده خواهد بود.

قوانین حرکتی بازی

۱. دو بازیکن از ۶ بلاک موجود نمی‌توانند خارج شوند.

۲. هر بازیکن ۳ بلاک در قسمت خود دارد و تنها می‌تواند تا مرز خود حرکت کند و اجازه عبور از آن و وارد شدن به محدوده حریف را ندارد.

۳. در طول بازی هر بازیکن می‌تواند ایستادن را انتخاب کند به عبارتی دیگر هیچ کاری نکند. نام این حرکت را wait گذاشته‌ایم.

۶. اگر بازیکنی این ریسک را بپذیرد که ۲ بار پشت سر هم wait را انجام دهد در صورتی که جانش کمتر از ۳ باشد ۱ جان برای این استراحت و تجدید قوا به دست می‌آورد.

۷. اگر هر دو بازیکن همزمان یک حمله را انجام دهند هر دو یک بلاک به عقب پرتاب می‌شوند. (هر دو یک واحد از هم دور می‌شوند.)

قوانین ضربه‌های بازی

۱. هر بازیکن برای حمله می‌تواند از دو ضربه punch و kick استفاده کند.

۲. بازیکنان فقط در حالتی که از هم فاصله‌ای ندارند یا تنها یک بلاک با هم فاصله دارند می‌توانند از ضربه kick استفاده کنند. در صورت برخورد این ضربه با حریف ۱ جان از او کم می‌کند.

۳. بازیکنان فقط در حالتی که از هم فاصله‌ای ندارند می‌توانند از ضربه punch استفاده کنند. در صورت برخورد این ضربه با حریف ۲ جان از او کم می‌کند.

۴. در جدول زیر وضعیت الویت ضربات نسبت به یکدیگر آورده شده است.

سمت چپ جدول فاصله دو بازیکن از یکدیگر، حرکت بازیکن اول و حرکت بازیکن دوم نوشته شده است. ستون سمت راست نشان می‌دهد در حالت نوشته شده کدام ضربه برنده است و زننده آن ضربه می‌تواند از حریف جان کم کند.

Distance	Action 1	Action 2	Winner
0	Punch	Kick	Action 1
1	Punch	Kick	Action 2

با توجه به جدول بالا در صورتی که بین دو بازیکن فاصله‌ای نباشد، الویت اعمال ضربه‌ها به صورت زیر است.

Kick < Punch

توجه داشته باشید زمانی که فاصله دو بازیکن ۱ است بر روی بازیکنان فقط ضربه kick اثرگذار است و در صورت زدن ضربه punch برخوردی با حریف صورت نمی‌گیرد. اما وقتی دو بازیکن فاصله‌ای با هم ندارند هر دو ضربه برخورد خواهد داشت اما اولویت متفاوتی دارند.

۵. اکشن‌های حرکتی اولویت بیشتری نسبت به ضربات دارند. به طور مثال اگر فاصله‌ای بین دو بازیکن نباشد و یکی از بازیکن‌ها ضربه punch را انتخاب کند و حریفش حرکت به سمت مخالف را برگزیند و از این طریق از بازیکن مقابل فرار کند، پس از آمدن کلاک و اعمال اکشن‌ها، اول حرکت چپ و راست (در این مثال فرار کردن) اعمال می‌شود در این حالت فاصله دو بازیکن ۱ می‌شود و در فاصله ۱ ضربه punch با حریف برخورد نخواهد داشت.

راهنمایی جهت پیاده‌سازی بازی

با توجه به قوانین گفته شده برای هر بازیکن یک ماشین حالت طراحی کنید. بدیهی است ماشین حالت هر دو بازیکن تقریبا یکسان است. در طراحی خود می‌توانید از کدگذاری برای حالت‌ها و ضربات مختلف استفاده کنید. در آزمایش ۱۰ دستور کار خود با نحوه پیاده‌سازی و نوشتن کد وریلاگ ماشین حالت آشنا شده‌اید. به همان شکل ماشین حالت‌های طراحی شده را پیاده‌سازی و شبیه‌سازی کنید. برای شبیه‌سازی نیز مانند آزمایش ۱۰ دستور کار از تست‌بنچ استفاده می‌شود. در ادامه سناریویی به شما داده شده است که حالت‌های مختلف این سناریو را به عنوان ورودی در تست‌بنچ وارد کنید و خروجی را در هر مرحله پس از آمدن کلاک ارزیابی کنید. توجه کنید که در تست‌بنچ بین اعمال ورودی‌های مختلف تاخیری ایجاد کنید که بتوانید خروجی valid برای هر ورودی را به درستی ارزیابی کنید.

به این نکته توجه کنید برای اینکه هر دو بازیکن بتوانند به طور همزمان بازی کنند و هر کدام اکشن‌های خود را اجرا کنند به دو ماشین حالت تقریبا یکسان (با تفاوت‌های جزئی در جهت حرکت بازیکن‌ها) اما جداگانه نیاز داریم. این دو ماشین حالت به صورت موازی با یکدیگر اجرا می‌شوند هر کدام مسئول پیاده‌سازی اکشن‌های یک بازیکن خواهد بود. به طور مثال ماشین حالت ۱ برای بازیکن شماره ۱ قرار داده شده است پس تمام حرکت‌های مربوط به خود بازیکن شماره ۱ به همراه ضربات بازیکن شماره ۲ به عنوان ورودی ماشین حالت ۱ خواهند بود تفاوت در نوع طراحی شماست که حرکات مربوط به خود بازیکن باعث جابه‌جایی بین حالت‌ها و حرکات حریف باعث تصمیم‌گیری در آن حالت می‌شود. این موضوع برای ماشین حالت بازیکن شماره ۲ برعکس خواهد بود.

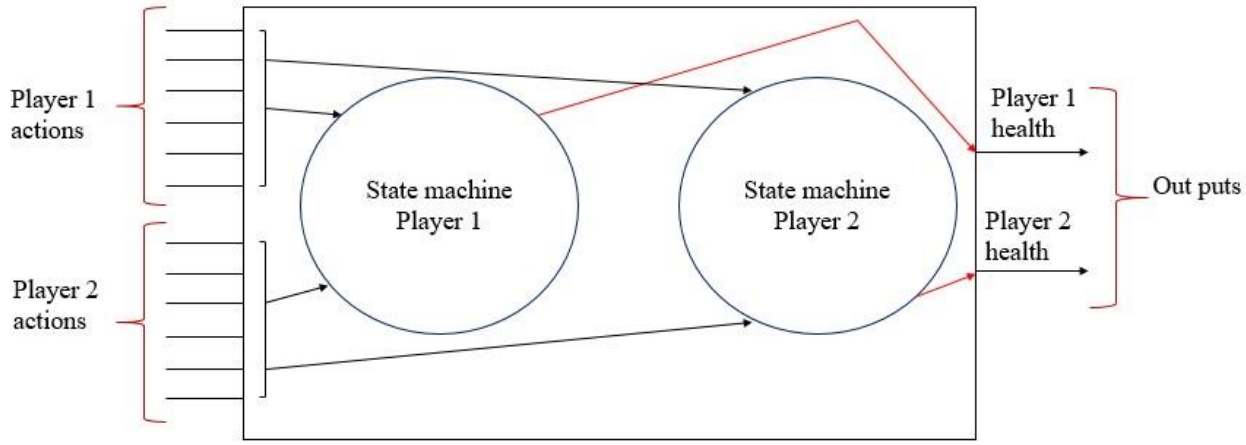
برای اینکه ابهامی در این بخش باقی نماند بار دیگر درباره عملکرد ماشین حالت فکر کنیم. ماشین حالت با توجه به حالت فعلی سیستم (بازیکن) و ورودی آن تصمیم می‌گیرد که سیستم (بازیکن) به کدام حالت برود و در آن حالت (ماشین مور) یا با این ورودی و رفتن به آن حالت (ماشین میلی) چه خروجی (جان از دست بدهد یا نه) داشته باشد. حال ورودی‌های سیستم ما (بازیکن ما) در این بازی چیست؟ اول ضرباتی که خودش به حریف می‌زند سپس ضرباتی که از حریف دریافت می‌کند. با توجه به اکشنی که خود بازیکن در نظر دارد اجرا کند می‌توان بین حالت‌ها جابه‌جا شد حال در آن حالت با توجه به اکشن بازیکن مقابل و اولویت‌ها و شرایط گفته شده تصمیم گرفته می‌شود که جان از خود بازیکن کم یا به آن اضافه شود یا از جان‌های بازیکن حریف کم شود.

با توجه به توضیحاتی که پیش‌تر داده شده است لیست اکشن‌هایی که هر بازیکن می‌تواند انجام دهد به صورت زیر است. به این ترتیب سیستم ۶ ورودی برای بازیکن ۱ و ۶ ورودی برای بازیکن ۲ خواهد داشت.

لیست اکشن‌های هر بازیکن
kick
punch
wait
jump
move left
move right

توجه داشته باشید از بین کل اکشن‌ها تنها یک اکشن برای هر بازیکن قابل اجرا است. به عبارتی دیگر در هر زمان هر بازیکن قادر به انجام تنها یک اکشن خواهد بود و دو یا چند اکشن همزمان توسط یک بازیکن انجام نمی‌شود.

بلاک دیاگرام طراحی که باید انجام دهید به صورت زیر است.



پس از اینکه طراحی را انجام دادید باید توسط یک تست‌بنچ مورد ارزیابی قرار بگیرد. در ادامه سناریویی نوشته شده است که از دانشجویان انتظار می‌رود وروی‌ها را براساس این سناریو در تست‌بنچ وارد کنند و بر روی منطق بازی خود اعمال کنند. خروجی دو عدد دوبیتی خواهد بود که تعداد جان هر بازیکن را پس از هر مرحله از اعمال اکشن‌ها نشان می‌دهد.

سناریوی بازی جهت پیاده‌سازی و ارزیابی

دانشجویان باید این سناریو را بر روی منطق بازی خود اجرا کنند و هنگام ارائه پروژه نتیجه هر مرحله از اعمال اکشن‌ها و تغییرات در تعداد جان بازیکن‌ها را به مدرس نشان دهند. در این سناریو چه کسی برنده بازی است؟

CLK	Player 1	Player 2
1	move right	move left
2	move right	move left
3	punch	jump
4	punch	kick
5	Punch	punch
6	move right	wait
7	jump	wait

8	punch	kick
9	kick	jump
10	move left	kick
11	kick	move left

همراه این پروژه دو پاورپوینت به نام‌های sample و testbench وجود دارد. در sample یک نمونه از بازی، نحوه اعمال اکشن‌ها و کسر یا دریافت امتیاز براساس منطق بازی آورده شده است. در testbench سناریویی که پیش‌تر توضیح داده شده که شما باید در تست‌بنچ خود وارد کنید و جهت پیاده‌سازی و ارزیابی منطق بازی شما است، نمایش داده شده است.

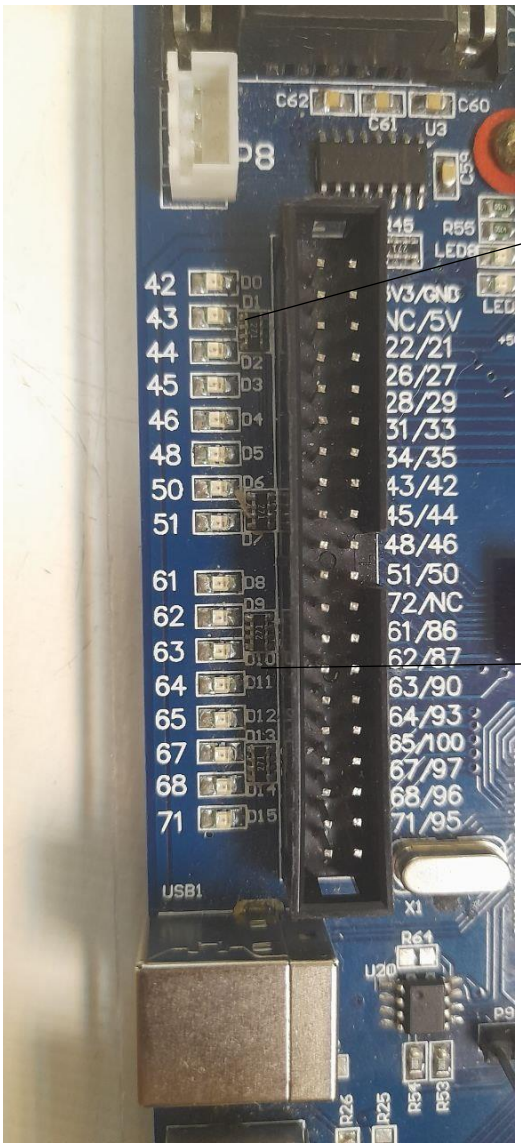
بخش امتیازی

پیاده‌سازی بر روی FPGA

بازی خود را بر روی FPGA اجرا کنید. با استفاده از ۱۲ عدد از دیپ سوئیچ‌های روی برد ورودی‌ها را اعمال کنید. موقعیت بازیکنان را در ۳ خانه محدوده خود با روشن شدن LEDها در هر سمت نشان دهید. خروجی که جان بازیکن‌ها است را بر روی سون‌سگمت نشان دهید. هر بار با تغییر جان هر بازیکن عدد روی سون‌سگمنت باید تغییر کند و مطابق جان جدید بازیکنان شود. (اختیاری: اگر دوست داشتید می‌توانید برای صدای اکشن‌ها از بازر روی برد استفاده کنید.)

دیپ سوئیچ برای اعمال اکشن‌های بازیکن شماره ۲

دیپ سوئیچ برای اعمال اکشن‌های بازیکن شماره ۱



نکته: به دلیل اینکه بتوانید خروجی خود را درست ارزیابی کنید و به درستی به شکلی که قابل مشاهده باشد نمایش دهید از یکی از دیپ سوئیچ‌ها به عنوان اعمال کننده ورودی استفاده کنید. به این صورت که در طراحی خود باید این ورودی را در نظر بگیرید و در هر حالتی زمانی که این ورودی (ورودی که توسط دیپ سوئیچ انتخابی اعمال می‌شود) غیر فعال است در state خود باقی می‌مانید و در صورت فعال شدن این ورودی با آمدن کلاک با توجه به وضعیت بازی در صورت لزوم state عوض می‌شود.

مواردی که باید توسط دانشجویان تحویل داده شود و مورد ارزیابی قرار می‌گیرد تا نمره پروژه را دریافت کنند:

۱. ارائه کد کامل منطق بازی

۲. پیاده‌سازی سناریو داده شده و ارائه نتایج شبیه‌سازی با استفاده از تست‌بنچ

۳. آمادگی و تسلط بر روی کد و ماشین حالت‌های طراحی شده برای توضیح یا تغییر هر بخش از آن‌ها هنگام ارائه

۴. در صورتی که نمره بخش امتیازی را می‌خواهید باید پیاده‌سازی کامل بر روی FPGA انجام دهید.

۵. اگر پیاده‌سازی بر روی FPGA کامل انجام شده باشد و خروجی به همان شکلی که خواسته شده است قابل مشاهده باشد، نیاز به نشان دادن نتایج شبیه‌سازی نیست و نتایج از روی برد قابل ارزیابی خواهد بود. (خودتان برای ارزیابی درستی کدتان به شبیه‌سازی نیاز دارید اما برای روز ارائه در صورتی که پیاده‌سازی عملی داشته باشید، ارائه آن کافی است.)

- هنگام ارائه اگر ماشین حالت‌های طراحی شده را به همراه داشته باشید و با استفاده از آن‌ها منطق کد پیاده‌سازی شده خود را توضیح دهید، توضیحات شما گویاتر خواهد بود.