

دانشگاه علم و صنعت

دانشکده مهندسی کامپیوتر

پروژه نهایی

Animation

اعضای گروه

یاسمن توکلی / پرنیان شاکریان

سپهر میرشاهی / الینا گودرزی / ارشیا انصاری

استاد:

دکتر خدادادی

سال تحصیلی: تیر 1402

شرح کلی

هر انیمیشن از كنار هم قرار گرفتن چند تصویر تشكیل می شود. انیمیشن های ساخته شده توسط این دستگاه با رزولوشن 8\*8 و در 4 فریم تولید می شوند. ابزاری وجود دارد شامل 8\*8 كلید كه طراح با استفاده از آن یك فریم از انیمیشن را طراحی می كند و سپس با فشردن دكمه ای جداگانه آنرا در اولین رجیستر خالی دستگاه ذخیره میكند. همچنین در دستگاه دكمه Playتعبیه شده است كه با فشردن آن تمامی فریم های ذخیره شده به نوبت نمایش داده میشوند و انیمیشن به وجود می آید. در این پروژه قصد داریم کد چنین طرحی را با زبان vhdl پیاده سازی کنیم.

شرح پروژه

در فایل اول پروژه شروع به ساخت package مورد نیاز خود میکنیم:

1. f عدد صحیح ثابت با مقدار 8 است.
2. w عدد صحیح ثابت با مقدار 2 است.
3. data\_type آرایه دو بعدی با ابعاد w-1 و w-1 بوده که هر عنصر آن یک بردار 24 بیتی (std\_logic\_vector) میباشد.
4. memory\_type آرایه یک بعدی با دامنه 0 تا f-1 میباشد.

entity انیمیشن توسط پنج پورت اعلام میشود:

* clk سیگنال ساعت ورودی است.
* reset سیگنال بازنشانی ورودی است.
* play سیگنال کنترل ورودی است که play یا ضبط را نشان می دهد.
* data\_in سیگنال ورودی دو بعدی از نوع types.data\_type است.
* data\_out سیگنال خروجی دو بعدی از نوع types.data\_type است.

architecture شامل نمونه ای از مولفه serial\_playback با استفاده از حلقه است. این حلقه بر روی ابعاد آرایه های data\_in و data\_out (i و j) تعریف و در package تکرار می شود. در داخل حلقه، مولفه serial\_playback به صورت sp نمایش داده می شود. پورت های component به سیگنال ها و متغیرهای مربوطه متصل می شوند که امکان play یا ضبط همزمان داده ها را به صورت موازی بر اساس مقادیر سیگنال های کنترلی play، clk و reset می دهد:

* پورت‌های clk، reset و play sp به سیگنال‌های clk، reset و play متصل هستند.
* پورت shift\_reg sp به عنصر data\_in(i,j) سیگنال data\_in متصل است.
* پورت data\_out sp به عنصر data\_out(i,j) سیگنال data\_out متصل است.

در فایل دوم پروژه Entity serial\_playback با پنج پورت اعلام خواهد شد:

1. clk سیگنال ساعت ورودی است.
2. reset سیگنال بازنشانی ورودی است.
3. play سیگنال کنترل ورودی است که play یا ضبط را نشان می دهد.
4. shift\_reg سیگنال ورودی 24 بیتی بوده که نشان دهنده داده هایی است که باید ثبت شوند.
5. data\_out سیگنال خروجی 24 بیتی بوده که نشان دهنده داده های در حال play است.

در ادامه برای serial\_playback برخی از سیگنال های داخلی مورد استفاده در architecture را تعریف میکنیم:

* memory سیگنالی از نوع memory\_type بوده که در package animation.types تعریف میشود.
* counter\_play و counter\_record سیگنال‌های عدد صحیحی هستند که به عنوان شمارنده استفاده شده و محدوده‌ای از 0 تا type.f-1 دارد.

architecture ما حاوی یک process statement حساس به تغییرات در سیگنال های clk و reset است. (در process، رفتار بر اساس مقادیر سیگنال‌های reset و play تعریف می‌شود.) اگر reset زیاد باشد ('1')، شمارنده های counter\_play و counter\_record هر دو به صفر بازنشانی می شوند. اگر ساعت ما rising\_edge باشد، process بر اساس مقدار سیگنال play، بین دو حالت تفاوت قائل می شود.

1. اگر play کم باشد ('0')، که نشان دهنده حالت ضبط است، process در هر لبه بالارونده ساعت فعال می شود. در داخل دستور if، مقدار counter\_record را بررسی کرده و اقدامات زیر را انجام میدهد:

* اگر counter\_record به types.f-1 برسد، مقدار shift\_reg فعلی آرایه memory در index ذخیره می شود. سپس counter\_record به صفر بازنشانی می شود. در غیر این صورت، مقدار shift\_reg در memory counter\_record ذخیره شده و counter\_record یک واحد افزایش می یابد.

2. اگر play زیاد باشد ('1')، که نشان دهنده حالت play است، process در هر لبه بالارونده ساعت فعال می شود. در داخل دستور if، مقدار counter\_play را بررسی کرده و اقدامات زیر را انجام میدهد:

* اگر counter\_play به types.f-1 برسد، داده‌های memory موجود در index به data\_out اختصاص داده، با رشته counter\_play اجرا شده و counter\_play به صفر بازنشانی می‌شود.در غیر این صورت، داده های memory موجود درindex به data\_out اختصاص داده شده ، یک دستور که مقدار counter\_play را به عنوان عدد صحیح نمایش می دهد اجرا می شود و counter\_play یک واحد افزایش می یابد.

موج testbench

