



پروژه درس طراحی سیستم‌های دیجیتال

استاد درس: دکتر خدادادی

افراد گروه:

محمد اصولیان

محمدحسین عباسپور

فرزان رحمانی

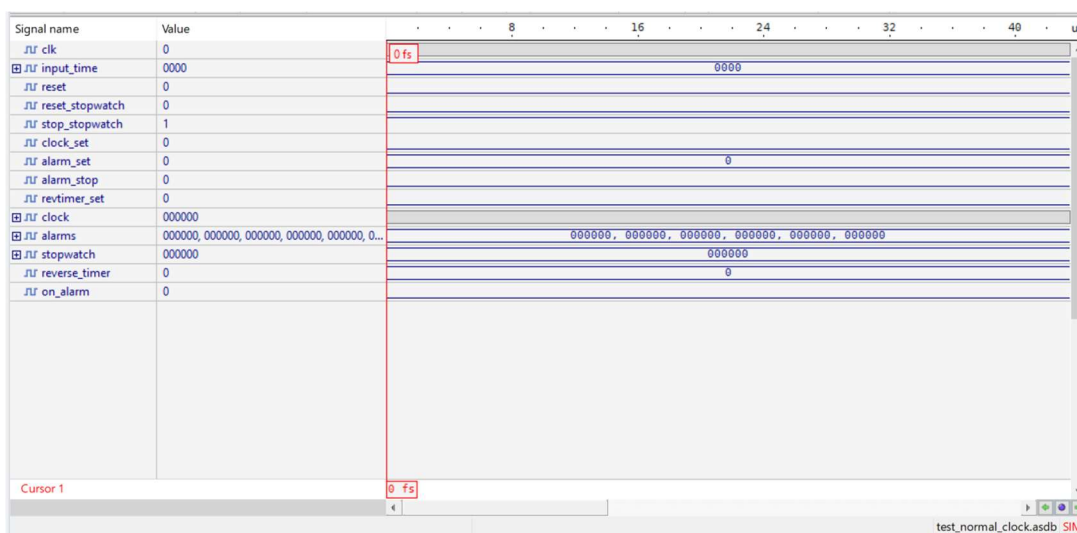
آیلین نائبزاده

بهار ۱۴۰۲

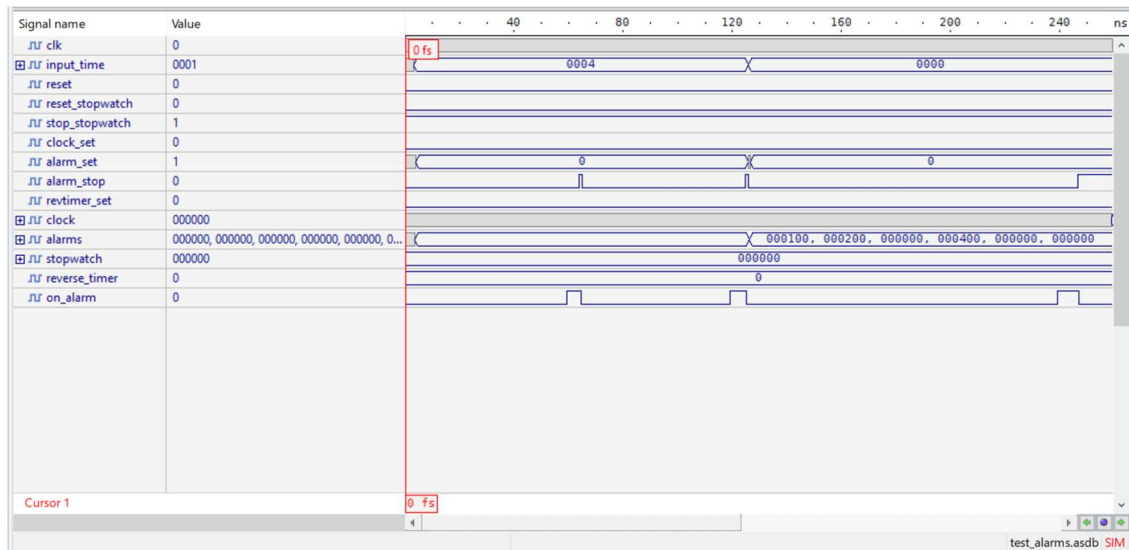
## توضیحات پروژه:

در این پروژه به پیاده‌سازی یک ساعت با استفاده از زبان vhdl می‌پردازیم. این برنامه چهار عملکرد متفاوت را پشتیبانی می‌کند.

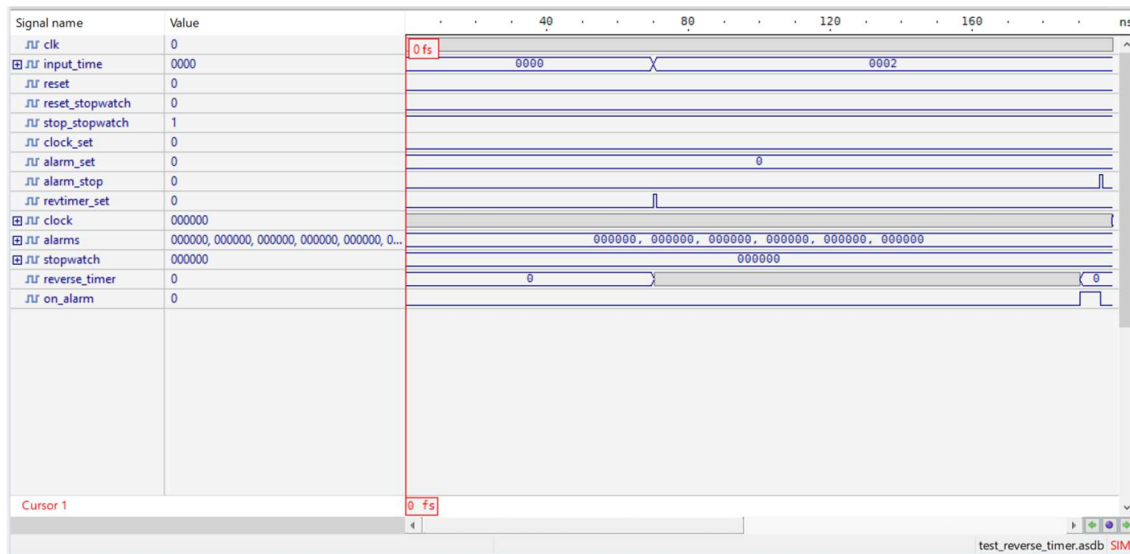
۱. ساعت عادی: این بخش ساعت را نمایش می‌دهد. می‌توان ساعت جدیدی را set/reset کرد. همچنین می‌توان ساعت را متوقف و دوباره اجرا کرد. امواج خروجی مربوط به این بخش را در تصویر زیر می‌توانید مشاهده کنید.



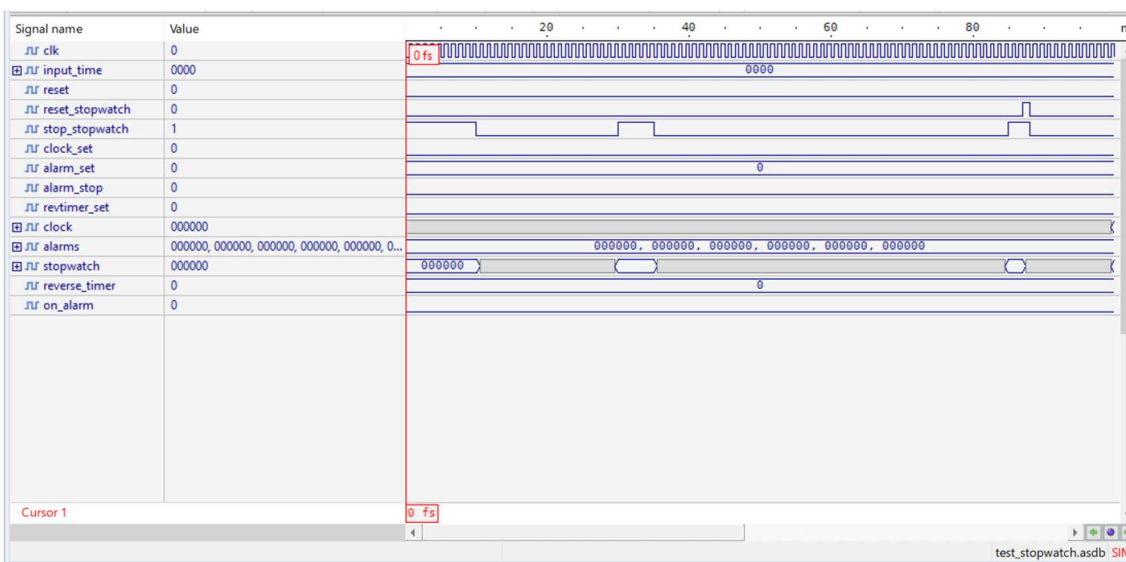
۲. آلام: می‌توان برای یک تایم خاص آلام تنظیم کرد. با رسیدن آلام صدای زنگ فعال می‌شود. می‌توان با یک دکمه آلام رو خاموش کرد. همچنین می‌توان چندین آلام را تنظیم کرد. (تا پنج آلام)



۳. شمارش معکوس: می‌توان شمارش معکوس تنظیم کرد. با پایان رسیدن زمان، ساعت زنگ می‌خورد و می‌توان زنگ را خاموش کرد. در هر لحظه فقط یک شمار معکوس وجود خواهد داشت.



۴. کرنومتر: می‌توان برای سنجش بازه زمانی (به عنوان کرنومتر) از قطعه استفاده کرد و چهار عمل شروع و پایان و توقف و ادامه را دارد.



## توضیحات کد package.vhdl:

این کد یک بسته (package) با نام **mypackage** است که شامل دو بخش **package** و **package body** می‌شود. این بسته ماژول‌ها و توابعی را برای کار با سیگنال‌های منطقی (logic signals) در محیط VHDL فراهم می‌کند.

در بخش **package**، سه کتابخانه استاندارد IEEE استفاده شده است:

- **ieee.std\_logic\_1164.all**: این کتابخانه شامل تعریف‌ها و توابع مربوط به سیگنال‌های منطقی در استاندارد IEEE 1164 است.

- **ieee.MATH\_REAL.all**: این کتابخانه شامل توابع ریاضی برای اعداد حقیقی است.

- **ieee.NUMERIC\_STD.all**: این کتابخانه شامل توابع ریاضی برای اعداد صحیح است.

در بخش **package**، یک نوع از داده‌ها و یک تابع تعریف شده است:

- **alarmarr**: این نوع داده، یک آرایه‌ای از اعداد بدون علامت با طول ۶ و بیت‌های ۲۱ تا ۰ را تعریف می‌کند. این نوع داده به عنوان یک نوع سفارشی در بسته استفاده می‌شود.

- **increase\_clock**: این تابع یک سیگنال منطقی به نام **intime** را به عنوان ورودی دریافت می‌کند و سیگنال منطقی **clock\_time** را به عنوان خروجی باز می‌گرداند. این تابع سیگنال **clock\_time** را به عنوان یک ساعت مجازی در نظر می‌گیرد و با هر فراخوانی به آن، مقدار ساعت را یک واحد افزایش می‌دهد.

در بخش **package body**، تابع **increase\_clock** پیاده‌سازی شده است. این تابع به عنوان ورودی یک سیگنال منطقی به نام **intime** دریافت می‌کند و سیگنال **clock\_time** را به عنوان متغیر محلی تعریف می‌کند و مقدار اولیه آن را برابر با **intime** قرار می‌دهد.

سپس با استفاده از تعریف‌های **alias**، قسمت‌های مختلف سیگنال **clock\_time** که برای نمایش ساعت مجازی استفاده می‌شود را تعریف می‌کند. به عنوان مثال، **secondl** و **secondb** به ترتیب رقم یک و دهم ساعت را نشان می‌دهند.

سپس با استفاده از دستورات شرطی، مقادیر قسمت‌های مختلف ساعت مجازی تغییر داده می‌شوند. به طور مثال، با افزایش **secondl** به یک واحد، اگر مقدار آن برابر با "۱۰۱۰" شود، به "۰۰۰۰" تنظیم می‌شود و همچنین **secondb** نیز یک واحد افزایش می‌یابد.

در نهایت، مقدار ساعت مجازی به عنوان خروجی تابع برگشت داده می‌شود.

با استفاده از این بسته، می‌توان سیگنال‌های منطقی را به عنوان ساعت مجازی استفاده کرده و مقدار آن‌ها را با فراخوانی تابع **increase\_clock** به صورت پیوسته افزایش داد.

## توضیحات کد clock.vhdl:

این کد یک مدل VHDL برای ساخت یک ساعت هشدار است که از بسته (`package`) قبلی استفاده می‌کند. این مدل با استفاده از یک سیگنال `clk` که ورودی است و چندین سیگنال دیگر که خروجی هستند، عملکرد ساعت هشدار را اجرا می‌کند و مقادیر ساعت، زمان‌های هشدار، تایمر معکوس و وضعیت هشدار را به خروجی می‌دهد.

در بخش `entity`، موارد زیر تعریف شده‌اند:

- ورودی‌ها:

- `clk`: سیگنال کلاک ورودی.

- `input\_time`: زمان ورودی که به عنوان تنظیم ساعت استفاده می‌شود.

- `reset`: سیگنال بازنشاندن (ریست) که در صورت فعال بودن، ساعت را صفر می‌کند.

- `reset\_stopwatch`: سیگنال بازنشاندن (ریست) تایمر استاپ واچ که در صورت فعال بودن، تایمر را صفر می‌کند.

- `stop\_stopwatch`: سیگنال توقف تایمر استاپ واچ که در صورت فعال بودن، تایمر را متوقف می‌کند.

- `clock\_set`: سیگنال تنظیم ساعت که در صورت فعال بودن، ساعت را با زمان ورودی `input\_time` تنظیم می‌کند.

- `alarm\_set`: عدد هشدار که در صورت برابر بودن با یکی از اعداد ۱ تا ۵، زمان هشدار متناظر را با زمان ورودی `input\_time` تنظیم می‌کند.

- `alarm\_stop`: سیگنال توقف هشدار که در صورت فعال بودن، هشدار را غیرفعال می‌کند.

- `revtimer\_set`: سیگنال تنظیم تایمر معکوس که در صورت فعال بودن، تایمر معکوس را با زمان ورودی `input\_time` تنظیم می‌کند.

- خروجی‌ها:

- `clock`: ساعت مجازی که به صورت یک عدد با طول ۲۲ بیتی نشان داده می‌شود.

- `alarms`: مجموعه زمان‌های هشدار که به صورت یک آرایه از اعداد با طول ۲۲ بیتی نشان داده می‌شود.

- `stopwatch`: تایمر استاپ واچ که به صورت یک عدد با طول ۲۲ بیتی نشان داده می‌شود.

- `reverse_timer`: تایمر معکوس که به صورت یک عدد نشان داده می‌شود.
  - `on_alarm`: وضعیت هشدار که به صورت یک سیگنال منطقی نشان داده می‌شود.
- در بخش `architecture`، عملکرد ساعت هشدار تعریف شده است. داخل فرآیند `process` که با تغییر `clk` فعال می‌شود، عملیات زیر انجام می‌شود:
- عملکرد تایمر معکوس (`revtimer`): اگر مقدار `revtimer` برابر با صفر نباشد، آن را یک واحد کاهش داده و در صورت برابر بودن با صفر، سیگنال `on_alarm` را فعال می‌کند.
  - عملکرد ساعت مجازی (`clock_time`): با استفاده از تابع `increase_clock` از بسته `mypackage`، مقدار ساعت مجازی را افزایش می‌دهد.
  - عملکرد تایمر استاپ واچ (`stopwatch_time`): اگر سیگنال `stop_stopwatch` غیرفعال باشد، مقدار تایمر را با استفاده از تابع `increase_clock` از بسته `mypackage`، افزایش می‌دهد.
  - عملکرد بازنشاندن (ریست) ساعت: اگر سیگنال `reset` فعال باشد، ساعت مجازی را به صفر تنظیم می‌کند.
  - عملکرد بازنشاندن (ریست) تایمر استاپ واچ: اگر سیگنال `reset_stopwatch` فعال باشد، تایمر استاپ واچ را به صفر تنظیم می‌کند.
  - عملکرد تنظیم ساعت: اگر سیگنال `clock_set` فعال باشد، بخش ساعت مجازی را با مقدار ورودی `input_time` تنظیم می‌کند.
  - عملکرد تنظیم هشدار: اگر مقدار سیگنال `alarm_set` برابر با یکی از اعداد ۱ تا ۵ باشد، بخش مربوط به هشدار در آرایه `alarm_times` را با مقدار ورودی `input_time` تنظیم می‌کند.
  - عملکرد فعال‌سازی هشدار: اگر بخش ساعت مجازی با یکی از زمان‌های هشدار برابر باشد، سیگنال `on_alarm` را فعال می‌کند.
  - عملکرد غیرفعال‌سازی هشدار: اگر سیگنال `alarm_stop` فعال باشد، سیگنال `on_alarm` را غیرفعال می‌کند.
  - عملکرد تنظیم تایمر معکوس: اگر سیگنال `revtimer_set` فعال باشد، مقدار تایمر معکوس را با استفاده از مقادیر ورودی `input_time` تنظیم می‌کند.
- در نهایت، مقادیر ساعت مجازی (`clock_time`)، زمان‌های هشدار (`alarm_times`)، تایمر استاپ واچ (`stopwatch_time`) و تایمر معکوس (`revtimer`) به خروجی داده می‌شوند. همچنین، وضعیت هشدار (`on_alarm`) به عنوان خروجی برای نشان دادن وضعیت فعال یا غیرفعال بودن هشدار تعیین می‌شود.



## توضیحات کد alarmclock\_TB.vhdl:

این کد یک `testbench` برای `entity` با نام `alarmclock` است که در بالا کد اصلی آن بررسی و توضیح داده شده است. `testbench` برای تست و بررسی صحت عملکرد `entity` استفاده می‌شود.

در بخش `architecture TB\_ARCHITECTURE`، یک سری سیگنال ورودی و خروجی برای `entity` تعریف شده است. این سیگنال‌ها برای ایجاد ورودی‌ها و نظارت بر خروجی‌ها در `testbench` استفاده می‌شوند.

در بلاک `process` اول، سیگنال `clk` با توجه به مقدار فعلی خود تغییر می‌کند تا یک سیگنال کلاک ساده ایجاد شود. این کلاک در حلقه `wait for 0.5 ns` با فاصله زمانی کوتاه تغییر می‌کند.

در بلاک `process` دوم، اقدامات تستی برای `entity` انجام می‌شوند. ابتدا مقادیر ورودی `reset`، `input\_time`، `reset\_stopwatch`، `clock\_set`، `alarm\_set` و `alarm\_stop` و `revtimer\_set` به مقادیر مشخص شده تنظیم می‌شوند.

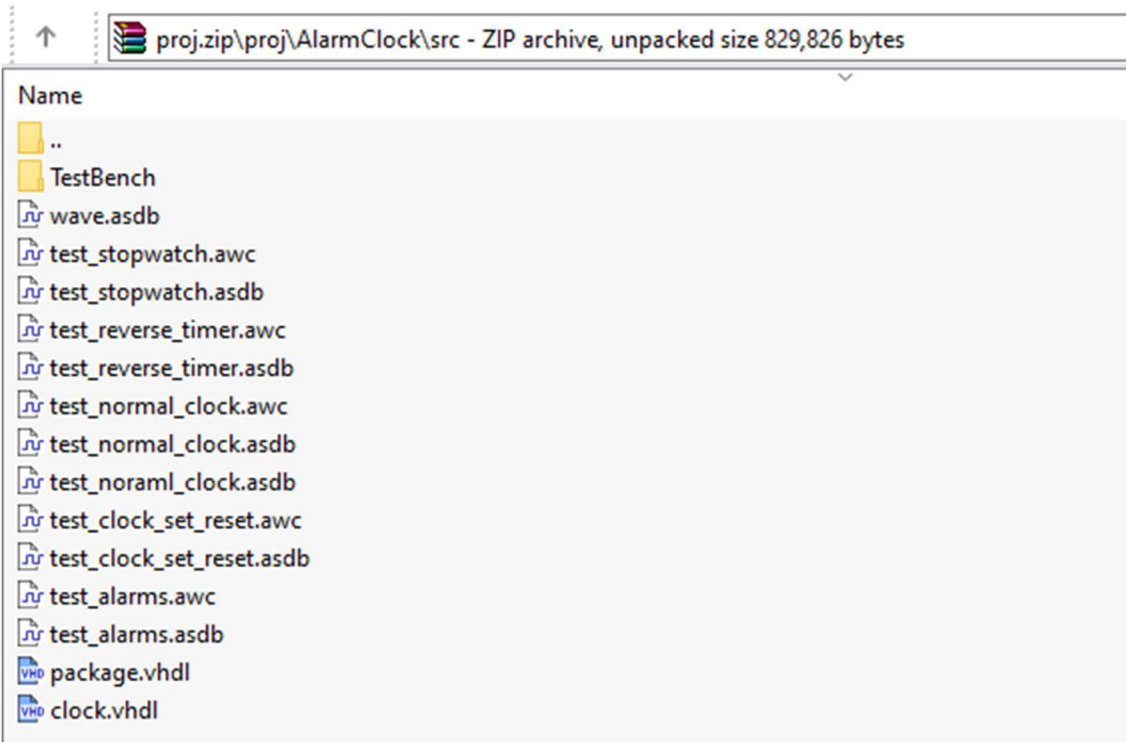
سپس با استفاده از دستورات `wait for` و `>=`، ورودی‌های `input\_time` و `clock\_set` تغییر می‌کنند و سپس به مقادیر اولیه باز می‌گردند. همچنین، ورودی `reset` به مدت کوتاهی فعال شده و سپس غیرفعال می‌شود.

سپس تست هشدارها (`alarms`) صورت می‌گیرد. ابتدا چهار هشدار فعال می‌شوند و سپس پس از مدت زمان مشخصی غیرفعال می‌شوند. همچنین، یکی از هشدارها برای مدت زمانی تنظیم شده و سپس متوقف می‌شود. در نهایت، تست تایمر معکوس (`reverse\_timer`) و تست تایمر استاپ واچ (`stopwatch`) انجام می‌شود.

بعد از انجام اقدامات تستی، مقادیر خروجی‌های `clock`، `alarms`، `stopwatch`، `reverse\_time` و `on\_alarm` که به صورت سیگنال‌ها تعریف شده‌اند، تست شده و عملکرد `entity` بررسی می‌شود.

در بخش `configuration TESTBENCH\_FOR\_alarmclock`، برای `testbench` برای `entity alarmclock` تعیین شده است. این بخش برای پیکربندی `testbench` استفاده می‌شود.

در نهایت نیز میتوانید صحت عملکرد کد های package، entity، و architectue و testbench ها با مراجعه به فایل های .awc که در مسیر proj\AlarmClock\src وجود دارند را مشاهده کنید.



پایان