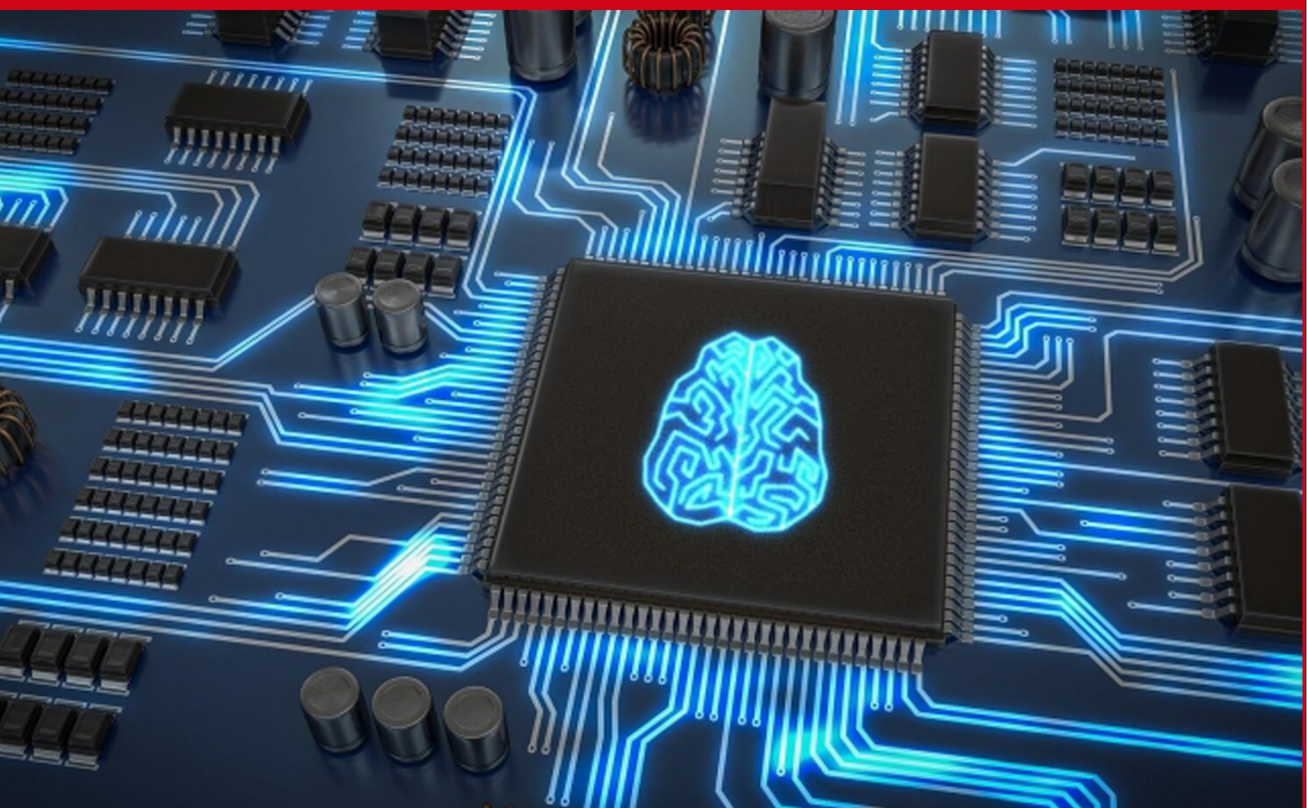
انتشارات موجک

سیستم وریلاگ برای توصیف سخت افزار

(طراحی مدار دیجیتال و روش‌های راستی آزمایی و آزمون)

تالیف: وایبهاو تاراته

ترجمه: علی منصور



به نام خداوند مهربانی‌ها

سیستم وریلاگ برای توصیف سخت افزار

(طراحی مدار دیجیتال و روش‌های راستی آزمایی و آزمون)

تالیف:

وایبهاو تاراته

ترجمه:

علی منصور

انتشارات موجک (ناشر دانشگاهی)

سرشناسه : تاراته، وایبهاو - Taraate, Vaibbhav
عنوان و نام پدیدآور : سیستم وریلاگ برای توصیف سخت‌افزار (طراحی مدار دیجیتال و روش‌های راستی‌آزمایی و آزمون)/ تالیف وایبهاو تاراته؛ ترجمه علی منصور.
مشخصات نشر : تهران: انتشارات موجک، ۱۴۰۱.
مشخصات ظاهری : ۳۲۴ ص.: جدول، نمودار.
شابک : ۲-۵۲۵-۹۹۴-۶۰۰-۹۷۸، ۲۵۲۰۰۰۰ ریال
وضعیت فهرست نویسی : فیپا
یادداشت : عنوان اصلی:
SystemVerilog for hardware description : RTL design and verification,2020.
موضوع : سیستم وریلاگ (زبان توصیف سخت افزار)
موضوع : SystemVerilog (Computer hardware description language)
شناسه افزوده : منصور، علی، ۱۳۵۳-، مترجم
رده بندی کنگره : TK ۷۸۸۵/۷
رده بندی دیویی : ۶۲۱/۳۹۲
شماره کتابشناسی ملی : ۸۹۵۷۷۶۳

انتشارات موجک (ناشر دانشگاهی)
واتساپ : ۰۹۳۶۳۰۳۱۲۵۸ کانال: telegram.me/mojak1
تلفن مرکز پخش : ۰۲۶۳۲۷۰۵۳۱۸ - ۰۲۶۳۲۷۰۲۶۵۹ - ۰۲۱۶۶۱۲۷۵۹۳ - ۰۲۱۶۶۴۲۹۷۳۳
ایمیل : mojakpublication@yahoo.com
سایت: www.mojak.ir اینستاگرام: mojakpublication

عنوان : سیستم وریلاگ برای توصیف سخت افزار (طراحی مدار دیجیتال و روش‌های راستی آزمایی و آزمون)
تالیف : وایبهاو تاراته
ترجمه : علی منصور
مشخصات ظاهری : ۳۲۴ صفحه، قطع وزیری
چاپ اول : تابستان ۱۴۰۱، تیراژ : ۵۰۰ جلد
قیمت : ۲۵۲۰۰۰۰ ریال، شابک : ۲-۵۲۵-۹۹۴-۶۰۰-۹۷۸

کلیه حقوق مادی و معنوی این اثر برای انتشارات موجک محفوظ است. هیچ شخص حقیقی و حقوقی حق چاپ و تکثیر این اثر را به هر شکل و صورت اعم از فتوکپی، چاپ کتاب و ... را ندارد. متخلفین به موجب بند ۵ ماده قانون حمایت از ناشرین تحت پیگرد قانونی قرار می‌گیرند.

تشکر و قدردانی

پس از به جای آوردن سپاس به درگاه احدیت، لازم می‌دانم مراتب قدردانی خود را نسبت به راهنمایی‌های ارزنده‌ی جناب آقای دکتر علی خیاطی در برگزیدن کتاب حاضر ابراز نمایم. از جناب آقای مهندس علی کریمیان به جهت کمک‌های وصف ناپذیرشان در پیشبرد ترجمه‌ی این کتاب تشکر می‌نمایم.

تقدیم به

مادر مهربان و فرزانه‌ام

که سخن را حرف به حرف و واژه به واژه،

بسان آواهای دلپذیر ترانه‌ای روح‌نواز به من آموخت.

فهرست مطالب

عنوان **صفحه**

مقدمه ۱۹

فصل اول: معرفی سیستم وریلاگ ۲۳

۱-۱ مقدمه ۲۳

۱-۲ روند طراحی ASIC ۲۳

۱-۳ راستی‌آزمایی ASIC ۲۶

۱-۳-۱ استراتژی‌های راستی‌آزمایی SOC ۲۶

۱-۴ سازه‌های VERILOG ۲۹

۱-۴-۱ دستورهای انتساب همروند ۲۹

۱-۴-۲ بلوک رویه‌ای ۳۰

۱-۵ مقدمه‌ای بر SYSTEM VERILOG ۳۳

۱-۶ SYSTEM VERILOG برای توصیف سخت افزار و راستی‌آزمایی ۳۵

۱-۷ خلاصه و بحث‌های آینده ۳۵

فصل دوم: مقادیر لیترال SYSTEM VERILOG و انواع داده ۳۷

۲-۱ مقدمه ۳۷

۲-۲ گیت‌های از پیش تعریف شده ۳۷

۲-۳ مدل سازی ساختاری ۳۹

۲-۴ ثابت‌های چند بیتی و الحاق ۴۱

۲-۵ لیترال‌ها ۴۲

۲-۵-۱ اعداد صحیح و منطقی ۴۲
۲-۵-۲ لیترال‌های real ۴۲
۲-۵-۳ لیترال زمان ۴۲
۲-۵-۴ لیترال‌های رشته‌ای ۴۳
۲-۵-۵ لیترال‌های آرایه ۴۴
۲-۵-۶ لیترال‌های ساختاری ۴۵
۲-۶ انواع داده ۴۶
۲-۶-۱ انواع داده‌های integer ۴۶
۲-۶-۲ انواع داده دو حالته یا چهار حالته ۴۷
۲-۶-۳ انواع داده علامت‌دار و بدون علامت ۴۷
۲-۶-۴ انواع داده real و shortreal ۴۷
۲-۶-۵ داده‌های چندل ۴۷
۲-۶-۶ انواع داده رویداد ۵۰
۲-۶-۷ نوع تعریف شده توسط کاربر ۵۰
۲-۷ خلاصه و بحث‌های آینده ۵۲

فصل سوم: توصیف سخت افزار با استفاده از SYSTEM VERILOG..... ۵۳
۳-۱ مقدمه ۵۳
۳-۲ چگونه می‌توانیم شروع کنیم؟ ۵۴
۳-۲-۱ اعداد و ثابت‌ها ۵۵
۳-۲-۲ عملگرها ۵۵
۳-۳ انواع داده شبکه ۵۸
۳-۴ بیاید در مورد عناصر ترکیبی فکر کنیم! ۶۰
۳-۴-۱ تخصیص پیوسته ۶۰
۳-۴-۲ بلوک رویه‌ای always_comb ۶۳

۳-۵ بیاید از ALWAYS_COMB برای پیاده سازی مبدل‌های کد استفاده کنیم ۶۵
۳-۵-۱ تبدیل کد دودویی به gray ۶۵
۳-۵-۲ مبدل کد gray به دودویی ۶۷
۳-۶ بیایید درک اولیه از همزمانی داشته باشیم ۶۸
۳-۷ بلوک رویه ALWAYS_LATCH ۶۹
۳-۸ بلوک رویه ALWAYS_FF ۷۰
۳-۹ بیایید از ALWAYS_FF برای پیاده سازی طراحی متوالی استفاده کنیم ۷۱
۳-۱۰ نمونه‌سازی با استفاده از اتصالات پورت نامگذاری شده (سبک VERILOG) ۷۲
۳-۱۱ نمونه‌سازی با استفاده از اتصال پورت مختلط ۷۴
۳-۱۲ خلاصه و بحث‌های آینده ۷۴

فصل چهارم: SYSTEMVERILOG و پشتیبانی از OOPS..... ۷۷
۴-۱ مقدمه ۷۷
۴-۲ انواع داده شمارشی ۷۷
۴-۲-۱ روش‌های تخصصی ۷۹
۴-۲-۲ روش‌های شمارش ۸۰
۴-۲-۳ انواع شمارشی در بیان عددی ۸۰
۴-۲-۴ تحویل خودکار به نوع پایه ۸۰
۴-۳ ساختارها ۸۱
۴-۳-۱ ساختار بسته بندی نشده و بسته بندی شده ۸۳
۴-۴ اتحادها ۸۴
۴-۵ آرایه‌ها ۸۵
۴-۵-۱ آرایه استاتیک ۸۵
۴-۵-۲ آرایه‌های بسته بندی شده تک بعدی ۸۶
۴-۵-۳ آرایه‌های بسته بندی شده چند بعدی ۸۶

۴-۵-۴ آرایه‌های ساختارها و اتحادها ۸۸
۴-۵-۵ آرایه‌های بسته بندی شده و بسته بندی نشده و سنتز ۸۸
۴-۵-۶ آرایه‌های پویا ۸۹
۴-۵-۷ آرایه‌های انجمنی ۸۹
۴-۵-۸ صف‌ها ۹۰
۴-۶ خلاصه و بحث‌های آینده ۹۰

فصل پنجم: مزایای مهم SYSTEMVERILOG ۹۳
۵-۱ مقدمه ۹۳
۵-۲ بلوک رویه‌ای VERILOG ۹۳
۵-۳ بلوک‌های رویه‌ای SYSTEMVERILOG ۹۵
۵-۳-۱ مدلسازی ترکیبی با استفاده از always_comb ۹۶
۵-۳-۲ طراحی‌های مبتنی بر لچ با استفاده از always_latch ۹۷
۵-۳-۳ طراحی‌های متوالی با استفاده از always_ff ۹۸
۵-۴ برچسب بلوک ۹۹
۵-۵ برچسب بیانیه ۱۰۰
۵-۶ برچسب مدول ۱۰۰
۵-۷ مزایای TASK و تابع ۱۰۱
۵-۸ تابع VOID ۱۰۴
۵-۹ حلقه‌ها ۱۰۴
۵-۹-۱ Verilog برای حلقه ۱۰۵
۵-۹-۲ System Verilog برای حلقه ۱۰۶
۵-۹-۳ مزایای حلقه SystemVerilog ۱۰۶
۵-۹-۴ حلقه Verilog while ۱۰۷
۵-۹-۵ حلقه System Verilog do-while ۱۰۸

۵-۱۰ رهنمودها ۱۰۸
۵-۱۱ خلاصه و بحث‌های آینده ۱۰۹

فصل ششم: طراحی ترکیبی با استفاده از SYSTEMVERILOG..... ۱۱۱
۶-۱ مقدمه ۱۱۱
۶-۲ نقش بلوک رویه‌ای ALWAYS_COMB ۱۱۲
۶-۳ IF-ELSE تو در تو و اولویت منطقی ۱۱۴
۶-۴ پارامتر و کاربرد آن در طراحی ۱۱۶
۶-۵ عملگر شرطی و استفاده برای استنتاج منطق MUX..... ۱۱۸
۶-۶ دیکدرها..... ۱۱۹
۶-۶-۱ دیکدر پارامتری ۱۲۲
۶-۶-۲ دیکدر با استفاده از تابع ۱۲۳
۶-۷ انکودر اولویت دار ۱۲۳
۶-۸ خلاصه و بحث‌های آینده ۱۲۵

فصل هفتم: طراحی ترتیبی با استفاده از SYSTEMVERILOG..... ۱۲۷
۷-۱ مقدمه ۱۲۷
۷-۲ لچ‌های عمدی با استفاده از ALWAYS_LATCH..... ۱۲۸
۷-۳ ثبات PIPO با استفاده از ALWAYS_FF..... ۱۲۹
۷-۴ بیایید از RESET ناهمزمان استفاده کنیم ۱۳۱
۷-۵ بیایید از RESET همزمان استفاده کنیم ۱۳۲
۷-۶ شمارنده UP-DOWN..... ۱۳۳
۷-۷ ثبات شیفت..... ۱۳۵
۷-۸ شمارنده حلقه ۱۳۶
۷-۹ شمارنده جانسون ۱۳۸

۷-۱۰ بیایید RTL را برای واحد حساب کلاک شده پیاده سازی کنیم ۱۳۹
۷-۱۱ بیایید RTL را برای واحد منطقی کلاک پیاده سازی کنیم ۱۴۳
۷-۱۲ خلاصه و بحث‌های آینده ۱۴۵

فصل هشتم: رهنمودهای طراحی و سنتز RTL ۱۴۷
۸-۱ مقدمه ۱۴۷
۸-۲ بیانیه NON-FULL CASE ۱۴۹
۸-۳ بیانیه FULL CASE ۱۵۱
۸-۴ دستورالعمل SYNOPSYS FULL_CASE ۱۵۲
۸-۵ UNIQUE CASE ۱۵۲
۸-۶ بیانیه CASEZ ۱۵۴
۸-۷ THE PRIORITY CASE ۱۵۵
۸-۸ بیانیه UNIQUE IF ۱۵۶
۸-۹ دیکدر با دستورالعمل SYNOPSYS FULL_CASE ۱۵۷
۸-۱۰ بیانیه PRIORITY IF ۱۵۸
۸-۱۱ مشکلات سنتز هنگام استفاده از PRIORITY CASE یا SYNOPSYS FULL_CASE ۱۵۹
۸-۱۲ کلاک‌های تولید شده ۱۶۰
۸-۱۳ کلاک گیت دار ۱۶۱
۸-۱۴ مولد کلاک چندگانه ۱۶۲
۸-۱۵ کلاک چند فازی ۱۶۳
۸-۱۶ بهینه سازی سطح ۱۶۴
۸-۱۷ بهبود سرعت ۱۶۸
۸-۱۸ بهبود توان و بهینه سازی ۱۷۱
۸-۱۹ خلاصه و بحث‌های آینده ۱۷۳

فصل نهم: طراحی RTL و استراتژی برای طراحی‌های پیچیده ۱۷۵
۹-۱ مقدمه ۱۷۵
۹-۲ استراتژی برای طرح‌های پیچیده ۱۷۶
۹-۳ ALU ۱۷۷
۹-۴ BARREL SHIFTER ۱۷۹
۹-۵ حافظه تک پورت و دو پورت ۱۸۱
۹-۵-۱ خواندن ناهمزمان ۱۸۳
۹-۵-۲ خواندن و نوشتن همزمان ۱۸۴
۹-۵-۳ RAM توزیع شده ۱۸۴
۹-۵-۴ BRAM ۱۸۶
۹-۵-۵ رم دو پورت ۱۸۷
۹-۶ BUS ARBITERS و استراتژی‌های طراحی ۱۸۹
۹-۷ دامنه‌های کلاک چندگانه ۱۹۰
۹-۸ طراحی و استراتژی‌های FIFO ۱۹۰
۹-۸-۱ FIFO ۱۹۰
۹-۸-۲ حافظه FIFO ۱۹۲
۹-۸-۳ همگام‌ساز دامنه خواندن به نوشتن ۱۹۳
۹-۸-۴ همگام‌ساز دامنه نوشتن به خواندن ۱۹۴
۹-۸-۵ Write full ۱۹۵
۹-۸-۶ Read empty ۱۹۶
۹-۹ خلاصه و بحث‌های آینده ۱۹۷

فصل دهم: ماشین‌های حالت محدود ۱۹۹
۱۰-۱ مقدمه ۱۹۹

۱۰-۲ ماشین حالت محدود (FSM) ۲۰۰
۱۰-۳ ماشین MOORE ۲۰۱
۱۰-۴ ماشین MEALY ۲۰۱
۱۰-۵ آشکارساز توالی غیر همپوشانی ماشین MOORE ۲۰۳
۱۰-۶ آشکارساز توالی همپوشانی ماشین MOORE ۲۰۵
۱۰-۷ آشکارساز توالی غیر همپوشانی ماشین MEALY ۲۰۸
۱۰-۸ آشکارساز توالی همپوشانی ماشین MEALY ۲۱۰
۱۰-۹ روش کدگذاری دودویی ۲۱۳
۱۰-۱۰ روش رمزگذاری ONE-HOT ۲۱۵
۱۰-۱۱ ماشین حالت با CASE معکوس ۲۱۶
۱۰-۱۲ کنترلر FSM ۲۱۹
۱۰-۱۳ سنتز مسیر داده و کنترل ۲۲۰
۱۰-۱۴ بهینه سازی FSM ۲۲۲
۱۰-۱۵ خلاصه و بحث‌های آینده ۲۲۳

فصل یازدهم: پورت‌ها و رابط‌های SYSTEMVERILOG ۲۲۵

۱۱-۱ مقدمه ۲۲۵
۱۱-۲ اتصالات پورت نام گذاری شده VERILOG ۲۲۵
۱۱-۳ اتصالات پورت ضمنی .NAME ۲۲۷
۱۱-۴ اتصالات پورت ضمنی .* ۲۲۹
۱۱-۵ ماژول‌های تو در تو ۲۳۱
۱۱-۶ ماژول خارجی ۲۳۲
۱۱-۷ رابط‌ها ۲۳۳
۱۱-۷-۱ اعلامیه رابط ۲۳۳
۱۱-۸ رابط با استفاده از بسته نام گذاری شده ۲۳۵

۱۱-۹ رابط با استفاده از بسته نرم افزاری عمومی ۲۳۶
۱۱-۱۰ پورت رابط ۲۳۸
۱۱-۱۱ THE MODPORTS ۲۳۹
۱۱-۱۱-۱ نام Modport در اعلامیه ماژول ۲۴۰
۱۱-۱۱-۲ ماژول نمونه و Modport ۲۴۱
۱۱-۱۲ روش‌های رابط ۲۴۲
۱۱-۱۳ رابط مجازی ۲۴۳
۱۱-۱۴ سمافور [۱] ۲۴۵
۱۱-۱۴-۱ new() ۲۴۵
۱۱-۱۴-۲ put() ۲۴۶
۱۱-۱۴-۳ get() ۲۴۶
۱۱-۱۴-۴ try_get() ۲۴۷
۱۱-۱۵ MAILBOXES ۲۴۸
۱۱-۱۶ خلاصه و بحث‌های آینده ۲۴۹

فصل دوازدهم: سازه‌های راستی‌آزمایی ۲۵۱
۱۲-۱ مقدمه ۲۵۱
۱۲-۲ بلوک رویه‌ای INITIAL ۲۵۲
۱۲-۳ مولد کلاک ۲۵۳
۱۲-۴ تولید کلاک با چرخه وظیفه متغیر ۲۵۴
۱۲-۵ منطق نسل RESET ۲۵۶
۱۲-۶ مکانیسم نظارت بر پاسخ ۲۵۶
۱۲-۷ چگونه پاسخ را حذف کنیم؟ ۲۵۸
۱۲-۸ چگونه بردارهای آزمایشی را از فایل اضافه کنیم؟ ۲۵۹
۱۲-۹ بیایید TESTBENCH را توصیف کنیم ۲۶۰

۱۲-۱۰ خلاصه و بحث‌های آینده ۲۶۳

فصل سیزدهم: تکنیک‌های راستی‌آزمایی و اتوماسیون ۲۶۵

۱۳-۱ مقدمه ۲۶۵

۱۳-۲ زمان‌بندی رویدادهای طبقه بندی شده ۲۶۶

۱۳-۳ تاخیر و مدل‌های تاخیر ۲۶۸

۱۳-۴ فرآیندها و موضوعات ۲۷۰

۱۳-۴-۱ رشته fork join ۲۷۰

۱۳-۴-۲ رشته fork join-any ۲۷۲

۱۳-۴-۳ رشته fork join-none ۲۷۴

۱۳-۵ حلقه‌ها و استفاده از آنها در TESTBENCHها ۲۷۵

۱۳-۵-۱ حلقه forever (بی نهایت) ۲۷۵

۱۳-۵-۲ Repeat (تکرار) ۲۷۶

۱۳-۵-۳ حلقه foreach ۲۷۶

۱۳-۶ بلوک‌های کلاک ۲۷۷

۱۳-۶-۱ انحراف ۲۷۸

۱۳-۶-۲ بلوک کلاک با رابط ۲۷۹

۱۳-۷ اتوماسیون و TESTBENCH ۲۸۰

۱۳-۸ خلاصه و بحث‌های آینده ۲۸۳

فصل چهاردهم: ساختارهای راستی‌آزمایی پیشرفته ۲۸۵

۱۴-۱ مقدمه ۲۸۵

۱۴-۲ تصادفی‌سازی ۲۸۵

۱۴-۳ تصادفی سازی محدود ۲۸۸

۱۴-۴ راستی‌آزمایی مبتنی بر ادعا ۲۸۸

۱۴-۵ بلوک PROGRAM ۲۹۱
۱۴-۶ مثال راستی‌آزمایی ۲۹۲
۱۴-۶-۱ testbench سطح بالا ۲۹۳
۱۴-۶-۲ توصیف کلاس تراکنش ۲۹۵
۱۴-۶-۳ توصیف کلاس مولد ۲۹۶
۱۴-۶-۴ توصیف درایور ۲۹۷
۱۴-۶-۵ بیایید محیط را داشته باشیم ۲۹۹
۱۴-۶-۶ بیایید آزمون تصادفی را انجام دهیم ۳۰۱
۱۴-۷ خلاصه بحث‌های آینده ۳۰۲

فصل پانزدهم: بررسی CASE راستی‌آزمایی ۳۰۳
۱۵-۱ مقدمه ۳۰۳
۱۵-۲ اهداف راستی‌آزمایی ۳۰۳
۱۵-۳ طراحی RTL (طرح در دست راستی‌آزمایی) ۳۰۴
۱۵-۳-۱ بیایید هدف معامله را داشته باشیم ۳۰۶
۱۵-۳-۲ بیایید کلاس مولد را توصیف کنیم ۳۰۷
۱۵-۳-۳ بیایید درایور را داشته باشیم ۳۰۸
۱۵-۳-۴ بیایید کلاس مانیتور را داشته باشیم ۳۰۹
۱۵-۳-۵ بیایید کلاس scoreboard را داشته باشیم ۳۱۰
۱۵-۳-۶ بیایید کلاس محیط را داشته باشیم ۳۱۱
۱۵-۳-۷ بیایید آزمون را انجام دهیم ۳۱۲
۱۵-۳-۸ بیایید رابط را داشته باشیم ۳۱۳
۱۵-۴ آینده طراحی و راستی‌آزمایی ۳۱۳

پیوست‌ها ۳۱۵

پیوست اول ۳۱۵
پیوست دوم ۳۱۶
پیوست سوم ۳۱۹
پیوست چهارم: کلمات مخفف ۳۲۳

مقدمه

در طول دو دهه گذشته، پیچیدگی طراحی به طور تصاعدی افزایش یافته است و برای داشتن SOCها و محصولات بدون اشکال، تلاش بیشتری در زمینه راستی‌آزمایی لازم است. برنامه ریزی راستی‌آزمایی که یک تعریف معماری راستی‌آزمایی است به ما امکان می‌دهد محصولات بدون اشکال و طرح‌های SOC را راه اندازی کنیم. هدف تیم راستی‌آزمایی، یافتن اشکالات عملکردی در مراحل اولیه طراحی است.

با افزایش نمایی پیچیدگی طراحی، تعداد بیشتری از اعضای تیم برای انجام کار در حوزه راستی‌آزمایی RTL و حتی در راستی‌آزمایی فیزیکی مورد نیاز هستند. این سناریو از سال ۲۰۰۵ تغییر کرده است زیرا به ساعات کار بیشتر در سطوح راستی‌آزمایی نیاز است. هدف راستی‌آزمایی پوشش محور و مبتنی بر ادعا است.

بسیاری از ما در طول دهه گذشته از Verilog-1995، Verilog-2001 و Verilog-2005 استفاده می‌کردیم، اما مشکل واقعی عدم وجود ویژگی‌های برنامه نویسی شی‌گرا بود. به همین دلیل، راستی‌آزمایی فرآیندی زمان‌بر بود. زبان‌های جدید در طی سال‌های ۱۹۹۵ تا ۲۰۰۵ برای رفع نیاز به راستی‌آزمایی ASIC و SOC تکامل یافتند. سیستم C با TLM برای راستی‌آزمایی سیستم و SystemVerilog که ابرمجموعه Verilog برای راستی‌آزمایی قوی ASIC و SOCها از اهداف اولیه این زبان‌ها است.

از سال ۲۰۰۵، به روز رسانی‌های زیادی برای SystemVerilog منتشر شده است و نسخه پایدار فعلی IEEE 1800-2017 است. SystemVerilog از الگوی شی‌گرا C و C++ استفاده می‌کند و به طور گسترده برای طراحی و راستی‌آزمایی ASIC و SOC استفاده می‌شود. به عبارت ساده، می‌توان گفت که این زبان برای مهندسان طراح و برای مهندسین راستی‌آزمایی، هدف را برآورده می‌کند، بنابراین زبان توصیف و راستی‌آزمایی سخت افزار است.

هدف اصلی این کتاب تشویق مهندسان و متخصصان به استفاده از SystemVerilog برای توصیف سخت افزاری است. خواه طراحی مبتنی بر ASIC یا FPGA باشد، این زبان می‌تواند برای توصیف RTL با استفاده از ساختارهای قابل سنتز توانمند و برای راستی‌آزمایی با استفاده از ساختارهای غیرقابل سنتز استفاده شود.

با در نظر گرفتن تمام سناریوهای فوق، کتاب در ۱۵ فصل سازماندهی شده است و مبانی SystemVerilog و توصیف سخت افزاری و راستی‌آزمایی با استفاده از SystemVerilog را پوشش می‌دهد. این کتاب از تعاریف نحوی SystemVerilog از راهنمای مرجع زبان (LRM) و شماتیک RTL با استفاده از ابزار Xilinx Vivado EDA استفاده می‌کند. خوانندگان می‌توانند برای اطلاعات بیشتر در مورد خانواده‌ها و ابزارها و مجوزهای FPGA به www.xilinx.com مراجعه کنند!

فصل ۱: مقدمه: اصول جریان طراحی ASIC، راستی‌آزمایی و استراتژی‌های راستی‌آزمایی را پوشش می‌دهد. این فصل حتی برای درک سبک‌های طراحی Verilog-2001، Verilog-2005 RTL و اصول اولیه SystemVerilog مفید است.

فصل ۲: مقادیر لیترال SystemVerilog و انواع داده: لیترال‌های SystemVerilog، انواع داده، گیت‌های از پیش تعریف شده و سبک مدل سازی ساختاری را پوشش می‌دهد. حتی این فصل برای درک انواع داده رشته‌ای و روش‌های ویژه رشته مفید است.

فصل ۳: توصیف سخت افزار با استفاده از SystemVerilog: هدف این فصل آشنایی با عملگرها، انواع داده و ساختارهای اولیه SystemVerilog است. حتی این فصل بر روی بلوک‌های رویه‌ای همزمان و رویه‌ای که در سراسر کتاب برای مدل‌سازی طرح‌های ترکیبی و ترتیبی استفاده می‌شوند، تمرکز دارد.

فصل ۴: SystemVerilog و پشتیبانی OOPS: این فصل در مورد انواع داده، کلاس، ساختار، اتحادها و آرایه‌ها و استفاده از آنها در طول طراحی و راستی‌آزمایی بحث می‌کند.

فصل ۵: پیشرفت‌های مهم SystemVerilog: استاندارد فعلی نسخه پایدار SystemVerilog IEEE 1800-2017 است. در این زمینه، این فصل در مورد ساختارهای مهم SystemVerilog و سایر پیشرفت‌های مهم SystemVerilog بحث می‌کند. این فصل

برای درک حلقه ها، توابع، وظایف، برچسب هایی که در سراسر این کتاب استفاده می‌شود مفید است!

فصل ۶: طراحی ترکیبی با استفاده از SystemVerilog: ساختارهای قابل سنتز و توصیف سخت افزار برای بلوک‌های طراحی ترکیبی مهم مانند مالتی پلکسرها، دی مالتی پلکسرها، دیکدرها، انکودرها و انکودرهای اولویت‌دار را پوشش می‌دهد. این فصل برای درک بلوک رویه‌ای always_comb، پارامترها، تخصیص شرطی و همزمان در حین مدل‌سازی برای طراحی ترکیبی مفید است.

فصل ۷: طراحی متوالی با استفاده از SystemVerilog: بلوک‌های رویه‌ای مانند always_latch و always_ff و استفاده از آن‌ها برای توصیف سخت‌افزار عناصر طراحی متوالی مانند لچ‌ها، فلیپ فلاپ‌ها، شمارنده‌ها و ثبات‌های شیفت، محاسبه کلاک‌شده و واحدهای منطقی را پوشش می‌دهد. این فصل حتی در مورد استفاده از ساختارهای کارا و مفهوم reset همزمان و ناهمزمان بحث می‌کند.

فصل ۸: دستورالعمل‌های طراحی و سنتز RTL: دستورالعمل‌های سنتز و بهینه‌سازی را با استفاده از ساختارهای قابل سنتز SystemVerilog پوشش می‌دهد. این فصل case-end، full-case، case موازی و if-else تودرتو را با سوئیچ‌های منحصربه‌فرد و اولویت‌دار و استفاده از آنها پوشش می‌دهد. حتی این فصل برای درک توصیف سخت افزاری برای بهینه سازی سطح و بهبود سرعت و توان برای طراحی مفید است.

فصل ۹: طراحی RTL و استراتژی‌های طراحی‌های پیچیده: استفاده از ساختارهای SystemVerilog برای توصیف طرح‌های پیچیده مانند ALU، barrel shifters، arbiterها، حافظه‌هایی مانند RAM تک پورت و دو پورت، FIFO و نتیجه سنتز آن‌ها را پوشش می‌دهد.

فصل ۱۰: ماشین‌های حالت محدود: طراحی FSM، moore و mealy را پوشش می‌دهد. آشکارسازهای توالی، دو و سه بلوک FSM بلوک always، طراحی کنترل کننده، مسیر داده و سنتز مسیر کنترل را مسدود می‌کنند. حتی این فصل برای درک تکنیک‌های بهینه سازی FSM مفید است.

فصل ۱۱: پورت‌ها و رابط‌های SystemVerilog: SystemVerilog انواع مختلفی از اتصالات پورت، رابط‌ها و modportها را اضافه می‌کند. اینها سازه‌های توانمندی هستند که در طول طراحی و راستی‌آزمایی استفاده می‌شوند. در این سناریو، این فصل در مورد نمونه سازی ماژول، رابط‌ها، modportها، سمافور و mailboxها بحث می‌کند.

فصل ۱۲: ساختارهای راستی‌آزمایی: ساختارهای غیرقابل سنتز SystemVerilog مانند بلوک رویه‌ای initial، کلاک، منطق تولید reset، testcaseها، بردارهای تست و اصول راستی‌آزمایی و testbenchها را پوشش می‌دهد. این فصل برای درک مولدهای محرک، بررسی کننده پاسخ و testbenchها با استفاده از SystemVerilog مفید است.

فصل ۱۳: تکنیک‌های راستی‌آزمایی و اتوماسیون: برنامه زمان‌بندی رویداد طبقه‌بندی شده SystemVerilog، تأخیرها، راستی‌آزمایی رویدادها و چرخه‌ها و اتوماسیون در طول راستی‌آزمایی را پوشش می‌دهد. این فصل برای درک testbench خودکار و نقش بلوک‌های کلاک مفید است.

فصل ۱۴: ساختارهای راستی‌آزمایی پیشرفته: تکنیک‌های راستی‌آزمایی پیشرفته، تصادفی سازی، تصادفی سازی محدود با راستی‌آزمایی مبتنی بر ادعا در این فصل مورد بحث قرار می‌گیرد. حتی این فصل مطالعه case راستی‌آزمایی را برای مدل حافظه ساده با استفاده از اجزای مختلف testbench را پوشش می‌دهد.

فصل ۱۵: مطالعه case راستی‌آزمایی: این فصل در مورد مطالعه case با استفاده از اجزای testbench مانند DUV، رابط، مولد، درایور، مانیتور و scoreboard بحث می‌کند.

این کتاب برای درک توصیف سخت افزار با استفاده از SystemVerilog و اصول راستی‌آزمایی با استفاده از SystemVerilog مفید است. از خوانندگان درخواست می‌شود که با تحولات و پیشرفت‌های جدید در زمینه طراحی و راستی‌آزمایی همگام باشند تا فرصت‌های شغلی بهتری را بدست آورند.

SystemVerilog for Hardware Description

RTL Design and Verification

© Springer Nature Singapore. 2020

Author: Vaibbhav Taraate

Translator: Ali Mansoor

در طول دو دهه گذشته، پیچیدگی طراحی به طور تصاعدی افزایش یافته است، بطوری که برای داشتن SOCها و محصولات بدون اشکال، تلاش بیشتری در زمینه راستی آزمایی لازم است. برنامه‌ریزی راستی‌آزمایی یا به بیانی دیگر، معماری راستی‌آزمایی امکان برپاسازی محصولات بدون اشکال و طرح‌های SOC را فراهم می‌نماید. هدف تیم راستی‌آزمایی، یافتن اشکالات عملکردی در مراحل اولیه طراحی است.

با افزایش نمایی پیچیدگی طراحی، تعداد اعضای بیشتری در تیم مربوط به راستی آزمایی RTL و حتی تیم مربوط به راستی‌آزمایی فیزیکی مورد نیاز هستند. این سناریو از سال ۲۰۰۵ همراه با افزایش تعداد نفر ساعت مورد نیاز در سطوح راستی‌آزمایی تغییر کرده است. هدف، راستی‌آزمایی پوشش محور و مبتنی بر ادعا است.

هدف اصلی این کتاب تشویق مهندسان و متخصصان به استفاده از SystemVerilog برای توصیف سخت افزاری است. خواه طراحی مبتنی بر ASIC یا FPGA باشد، این زبان می‌تواند برای توصیف RTL با استفاده از ساختارهای قابل سنتز توانمند و برای راستی‌آزمایی با استفاده از ساختارهای غیرقابل سنتز استفاده شود.

انتشارات موجک

9786009945252