



آزمایشگاه مدارهای منطقی و معماری کامپیوتر

دکتر مانی زارعی

(استادیار دانشگاه آزاد اسلامی واحد شهر قدس)

مهندس علی بی‌زاده

(عضو هیات علمی دانشگاه آزاد اسلامی واحد شهر قدس)

سرشناسه: زارعی، مانی، ۱۳۶۰؛ ولی زاده، علی، ۱۳۵۴

عنوان و نام پدید آورنده: آزمایشگاه مدارهای منطقی و معماری کامپیوتر/مانی زارعی (عضو هیأت علمی دانشگاه آزاد اسلامی واحد شهر قدس) علی ولی زاده (عضو هیأت علمی دانشگاه آزاد اسلامی واحد شهر قدس)

مشخصات نشر: تهران: انتشارات دانشگاه آزاد اسلامی، ۱۳۹۶

مشخصات ظاهری: ۲۸۰ص، جدول، شکل و نمودار

شابک: ۹۷۸-۹۶۴-۱۰-۴۷۲۰-۹

فهرستبندی: بر اساس اطلاعات فیا

یادداشت: منابع: ص: ۲۸۰-۲۷۹

موضوع: کامپیوتر

رده بندی: سر ۱۳۹۶ از ۷: TKV86A/

رده بندی: ص: ۱۵۰۷۸ ۶۲۱

شماره کتاب: ص: ۲۳۶

- نام کتاب: آزمایشگاه مدارهای منطقی و معماری کامپیوتر
- تألیف: مانی زارعی، علی ولی زاده
- ویراستار و صفحه آر: مانی زارعی
- چاپ اول: ۱۳۹۶
- تیراژ: ۲۰۰۰ نسخه
- ناشر: سازمان چاپ و انتشارات دانشگاه آزاد اسلامی یا همکاری واحد شهر قدس
- شابک: ۹۷۸-۹۶۴-۱۰-۴۷۲۰-۹
- قیمت: ۱۲۰۰۰۰ ریال
- قطع: وزیری
- محل فروش: دانشگاه آزاد اسلامی واحد شهر قدس و کتابفروشی های معتبر

در عصر حاضر کامپیوترها نقش تعیین کننده و غیر قابل انکاری در زندگی بشر ایفا می کنند به طوری که انجام خیلی از امور بدون استفاده از آنها امکان پذیر نمی باشد. یک سیستم کامپیوتری شامل دو بخش سخت افزاری و نرم افزاری است. هسته اصلی هر کامپیوتر پردازنده ای هست که به صورت یک تراشه داخل آن تعبیه شده است و اصلی ترین جزء سخت افزار به حساب می آید. برنامه های مختلفی که برای یک کامپیوتر نوشته می شوند، توسط پردازنده پردازش و اجرا می شوند که این برنامه ها طیف وسیعی از کاربردها را شامل می شوند. بنابراین، با بروز مشکل در عملکرد پردازنده ای یک سیستم، عملاً هیچ کاری توسط این سیستم به طور صحیح اجرا نمی شود.

در دوره های کارشناسی، پوس و ناپیوسته رشته مهندسی کامپیوتر و حتی مهندسی برق دروسی تحت عنوان مدارهای منطقی و معماری کامپیوتر ارائه می گردد. در مدار منطقی انواع مدارهای ترکیبی و ترتیبی معرفی شده و به تفکیک و با طراحی مدارهای مختلف، کاربرد هر یک توضیح داده می شود. در بخش تدارک دیده شده برای آزمایشگاه مدار منطقی کتاب حاضر، مدارهای سخت افزاری متنوعی در قالب آزمایش های مختلف به صورت عملی شبیه سازی می شود و لذا در این بخش، دانشجو با نحوه کار مدارهای ترکیبی نظیر مالتی پلکسرها، دیکدر، انکدر، تمام جمع کننده، جمع کننده BCD، سیستم گذرگاه مشترک و همچنین مدارهای ترتیبی نظیر شمارنده، حافظه LATCH، بافر، شیفت رجیستر، حافظه اصلی و بافر اولین ورودی - اولین خروجی آشنا می شود.

در معماری کامپیوتر معمولاً نحوه طراحی و ساخت پردازنده کامپیوتر به عنوان اصلی ترین بخش آن مورد بررسی قرار می گیرد. در درس معماری کامپیوتر مطالب متنوعی مشتمل بر معرفی واحدهای اصلی کامپیوتر، نحوه اجرای دستورالعمل ها توسط کامپیوتر، معماری مجموعه دستورالعمل ها، معیارها و محک های ارزیابی

کارایی پردازنده‌ها و طراحی بخش‌های مختلف پردازنده از جمله واحد محاسبه و منطق، واحد کنترل، سیستم حافظه و مسائل گذرگاه، بخش‌های ورودی و خروجی و غیره مورد بررسی قرار می‌گیرد. دانشجو در بخش آزمایشگاه معماری کامپیوتر، پیاده‌سازی و ساخت کلیه مدارهای سخت افزاری لازم در داخل یک پردازنده را به صورت عملی به انجام رسانده و از نزدیک با نحوه کار مدارهایی نظیر مالتی پلکسر، دیکدر، انکدر، انواع حافظه، واحد محاسبه و منطق، بانک رجیستر، مسیرهای داده، واحد کنترل و نهایتاً یک پردازنده کامل آشنا می‌شود. پس از ساخته شدن همه قسمت‌ها به صورت سخت‌افزاری، دانشجو باید کل پردازنده را تست نماید. تست کامل یک پردازنده با یک برنامه که به زبان ماشین همان پردازنده نوشته شده است، امکان‌پذیر می‌باشد. یک برنامه جامع برای تست، برنامه‌ای است که شامل همه دستورات پیاده‌سازی شده در سخت‌افزار و همچنین همه ترتیبات ممکن دستورات باشد. جهت برنامه‌نویسی می‌توان از زبان‌های سطح بالای نظیر C نیز استفاده نمود ولی در صورت استفاده باید کامپایلر مناسبی جهت تبدیل برنامه از سطح بالا به زبان ماشین این پردازنده خاص در اختیار داشته باشیم.

امروزه کتاب‌های متنوعی برای آموزش مدارهای منطقی و معماری کامپیوتر نگاشته شده است ولی کتابی که بتواند به صورت عملی مطالب مطرح شده این درس را در قالب یک درس آزمایشگاهی آموزش دهد به دسترس ما نرسیده است. لذا نویسندگان کتاب حاضر بر آن شدند که با بهره‌گیری از تجربیات چند ساله تدریس و نیاز جامعه دانشگاهی به چنین منبعی، این کتاب را تهیه نمودند و در اختیار دانشجویان قرار دهند. خداوند متعال را سپاس می‌گوییم که توفیق چنین خدمتی را به ما عنایت فرمود.

بخش آزمایشگاه مدارهای منطقی کتاب حاضر بر اساس هر دو کتاب "طراحی دیجیتال" یا "Digital Design" و "معماری کامپیوتر" یا "Computer Architecture" و همچنین سایر مراجع مرتبط تنظیم شده است که توسط Morris Mano نگاشته شده است [۷-۱۴]. پیاده‌سازی مدارهای دیجیتالی مختلف

در بخش آزمایشگاه مدارهای منطقی در قالب هشت آزمایش مجزا به کمک نرم افزار Proteus انجام می شود. در هر آزمایش ابتدا عملکرد مدار معرفی شده توضیح داده می شود سپس این مدار به صورت عملی در محیط Proteus پیاده سازی و تست می شود.

بخش آزمایشگاه معماری کتاب حاضر بر اساس کتاب "سازمان کامپیوتر و طراحی آن..." یا "Computer Organization and Design the hardware/software interface" می باشد که توسط آقایان Hennessey و Patterson نگاشته شده و توسط انتشارات کافمن به چاپ رسیده است [۱۱]. در کتاب معرفی شده، پردازنده ای تحت عنوان MIPS معرفی و کلیه مفاهیم معماری کامپیوتر بر اساس آن توضیح داده شد. امروزه پردازنده MIPS جزء پردازنده های معروف در حوزه سیستم های نهفته (embedded systems) می باشد. بنابراین عمده مباحث مطرح شده در بخش آزمایشگاه معماری کتاب حاضر مربوط به پردازنده MIPS است و معماری مجموعه دستورات (ISA) این پردازنده جهت طراحی بخش های سخت افزاری و نرم افزاری مورد نظر خواهد بود. در این نوشته، ساخت یک پردازنده ساده MIPS به طور کامل آموزش داده خواهد شد. جهت پیشگیری از اضافه شدن حجم کتاب برای یک درس آزمایشگاهی، شش مطلب این کتاب به پیاده سازی و تست پردازنده MIPS اختصاص داده شده و نمت به طراحی پرداخته شده است. بعد از پیاده سازی نهایی دانشجو نیاز دارد پردازنده ساخته شده را تست کند و نحوه عملکرد آنچه پیاده سازی کرده است را عملاً ببیند، که به دلیل اهمیت موضوع تست در این کتاب به تفصیل و در قالب یک فصل به این موضوع پرداخته شده است.

مطالب این کتاب در قالب چهار فصل و یک پیوست تدوین شده است: در فصل اول مقدمه ای بر طراحی سخت افزار ارائه شده است. فصل دوم ابتدا به معرفی و آموزش نرم افزار Proteus می پردازد و سپس در قالب هشت آزمایش انواع مدارهای منطقی شبیه سازی و تحلیل می شوند. معرفی و کاربرد نرم افزار MAXplus و همچنین پیاده سازی یک پردازنده ساده MIPS به کمک این نرم افزار در قالب چهار آزمایش در فصل سوم توضیح داده شده است. در فصل چهارم طی چهار مثال جامع پردازنده

شبیه سازی شده از لحاظ عملکرد تست می شود. به منظور یادآوری و عدم نیاز به مراجعه به کتاب معماری، بخش پیوست کتاب به طور اجمالی به مرور پردازنده MIPS می پردازد.

در متن کتاب برای هر یک از واژه های تخصصی معادل آن به زبان انگلیسی در پاورقی ذکر شده است. بدیهی است که این کتاب علی رغم تلاش فراوان نویسندگان نمی تواند خالی از ایراد و اشتباه باشد. لذا نویسندگان این کتاب از کلیه نظرات، انتقادات و پیشنهادات سازنده عزیزان خواننده استقبال می کنند.

مانی زارعی، علی ولی زاده

فهرست مطالب (خلاصه)

عنوان	صفحه
آزمایش ۱: طراحی و تست تمام جمع کننده به کمک مدار ورودی.....	۵۵
آزمایش ۲: شبیه سازی عملکرد انکدر با تقدم ورودی و تست آن به کمک مدار ورودی.....	۶۵
آزمایش ۳: معرفی عملکرد نمایشگر 7 Segment و بکار بردن تکنیک های مختلف در ساده سازی مدار.....	۷۴
آزمایش ۴: طراحی و تست شمارنده دودویی.....	۸۵
آزمایش ۵: طراحی و تست جمع کننده BCD به کمک مدار ورودی.....	۹۰
آزمایش ۶: طراحی سیستم گذرگاه مشترک به کمک بافرهای سه حالته و تست آن.....	۹۶
آزمایش ۷: طراحی و تست بافر اولین ورودی - اولین خروجی.....	۱۰۱
آزمایش ۸: شبیه سازی عملکرد حافظه اصلی و تست آن.....	۱۰۶
آزمایش ۹: طراحی و تست کنترل رجیستر.....	۱۴۳
آزمایش ۱۰: طراحی و تست ALU.....	۱۷۲
آزمایش ۱۱: طراحی و تست مسیر و اتصالات.....	۱۸۹
آزمایش ۱۲: طراحی حافظه داده، واحد گسترش بیت علامت، مدار شیفت دهنده.....	۲۰۴
آزمایش ۱۳: طراحی واحد کنترل ALU و واحد کنترل مرکزی.....	۲۱۳
آزمایش ۱۴: طراحی پردازنده MIPS به کمک مدارهای آزمایش های ۹ تا ۱۳.....	۲۱۷

فهرست مطالب

عنوان	صفحه
مقدمه	۱
فصل اول - مقدمه‌ای بر طراحی سخت افزار	۷
۱-۱ طراحی مدارهای دیجیتال با استفاده از تراشه‌های سری ۷۴	۱۰
۲-۱ پیاده سازی مدار با استفاده از تراشه‌های قابل برنامه ریزی	۱۱
۱-۲-۱ نحوه برنامه ریزی تراشه‌های قابل برنامه ریزی	۱۵
۳-۱ پیاپی سازی مدارات در سطح سیستم	۱۶
۴-۱ مراحل سخت‌افزار به کمک زبان‌های برنامه‌سازی	۱۶
۱-۴-۱ مراحل طراحی به کمک زبان‌های طراحی سخت‌افزار	۱۹
۵-۱ فرهنگ اصطلاحات این کتاب	۲۲
فصل دوم - آموزش نرم افزار Proteus و شبیه‌سازی مدارهای منطقی به کمک این نرم‌افزار	۲۵
۱-۲ آموزش شبیه‌سازی	۲۶
۱-۱-۲ انتخاب قطعات مورد نیاز	۳۰
۲-۱-۲ انتخاب زمین	۳۳
۳-۱-۲ انتخاب منبع تغذیه	۳۴
۴-۱-۲ قرار دادن قطعات در صفحه طراحی ISIS	۳۵
۵-۱-۲ حذف کردن قطعه از صفحه شماتیک	۳۶
۶-۱-۲ چرخاندن قطعه درون صفحه شماتیک	۳۷
۷-۱-۲ ایجاد تغییر در خصوصیات قطعات	۳۹
۸-۱-۲ اتصال قطعات به یکدیگر از طریق سیم‌کشی	۴۰
۹-۱-۲ روش‌های کاهش حجم سیم‌کشی	۴۱
۱-۹-۱-۲ ترمینال DEFAULT	۴۲
۲-۹-۱-۲ ترمینال‌های INPUT, OUTPUT و BIDIR	۴۵

۴۷	Buses Mode و Wire Label Mode (۳-۹-۱-۲)
۵۰	تغییر اندازه صفحه طراحی (۱۰-۱-۲)
۵۱	جابجایی پنجره صفحه طراحی (۱۱-۱-۲)
۵۳	اجرای برنامه (۱۲-۱-۲)
۵۵	آزمایش ۱
۵۵	طراحی تمام جمع کننده (۲-۲)
۵۸	طراحی و شبیه‌سازی تمام جمع کننده در Proteus (۱-۲-۲)
۶۵	آزمایش ۲
۶۵	شبیه‌سازی عملکرد انکدر با تقدم ورودی و تست آن به کمک مدار ورودی (۳-۲)
۶۵	ترسیم ۴۱۰۷ (۱-۳-۲)
۶۸	شبیه‌سازی ۷۴۱۳۹ در Proteus (۱-۱-۳-۲)
۷۴	آزمایش ۳
۷۴	معرفی نمایشگر 7 Segment، استفاده از قابلیت‌های نرم‌افزار در ساده‌سازی مدار (۴-۲)
۷۴	نمایشگر 7 Segment (۱-۴-۲)
۷۶	اتصال انکدر به نمایشگر 7 Segment (۱-۱-۴-۲)
۷۸	ساده‌سازی مدار انکدر (۲-۴-۲)
۷۸	ساده‌سازی به کمک ترمینال DEFAULT (۱-۲-۴-۲)
۷۹	ساده‌سازی به کمک ترمینال‌های INPUT و OUTPUT (۲-۲-۴-۲)
۸۲	ساده‌سازی به کمک ابزارهای Buses Mode و Wire Label Mode (۳-۲-۴-۲)
۸۵	آزمایش ۴
۸۵	شمارنده دودویی (۵-۲)
۸۸	شبیه‌سازی شمارنده دودویی در Proteus (۱-۵-۲)
۹۰	آزمایش ۵
۹۰	جمع کننده BCD (۶-۲)
۹۳	شبیه‌سازی جمع کننده BCD در Proteus (۱-۶-۲)
۹۶	آزمایش ۶

۷-۲	سیستم گذرگاه مشترک با استفاده از بافرهای سه حالت	۹۶
۷-۲-۱	شبیه‌سازی سیستم گذرگاه مشترک با استفاده از بافرهای سه حالت	۹۹
۷	آزمایش	۱۰۱
۷-۲-۸	بافر اولین ورودی اولین خروجی	۱۰۱
۷-۲-۱۸	شبیه‌سازی بافر اولین ورودی اولین خروجی	۱۰۴
۸	آزمایش	۱۰۶
۷-۲-۹	حافظه اصلی RAM-6116	۱۰۶
۷-۲-۱۹	شبیه‌سازی حافظه اصلی RAM-6116	۱۱۰
فصل سوم - آموزش نرم‌افزار MAXPLUS و طراحی پردازنده MIPS به کمک این		
	نرم‌افزار	۱۱۳
۷-۲-۱۰-۱	آشنایی با MAXplus	۱۱۴
۷-۲-۱۰-۱-۱	ایجاد پروژه و فایل	۱۱۴
۷-۲-۱۰-۱-۲	ایجاد مدار	۱۱۶
۷-۲-۱۰-۱-۳	ذخیره مدار	۱۲۰
۷-۲-۱۰-۱-۴	کامپایل مدار	۱۲۱
۷-۲-۱۰-۱-۵	شبیه‌سازی مدار	۱۲۴
۷-۲-۱۰-۱-۶	انتخاب FPGA	۱۳۲
۷-۲-۱۰-۱-۷	استفاده از مدارهای آماده در Maxplus	۱۳۳
۷-۲-۱۰-۲	طراحی و پیاده‌سازی یک پردازنده ساده MIPS	۱۳۸
۷-۲-۱۰-۲-۱	مسیر داده	۱۴۱
۹	آزمایش	۱۴۳
۷-۲-۱۰-۲-۲	بانک رجیستر	۱۴۳
۷-۲-۱۰-۲-۲-۱	پیاده‌سازی رجیستر ۳۲ بیتی	۱۴۷
۷-۲-۱۰-۲-۲-۲	پیاده‌سازی دیکدر	۱۵۸
۷-۲-۱۰-۲-۲-۳	طراحی مالتی پلکسرها	۱۶۵

۱۶۷	 ۴-۲-۲-۳ ترکیب قطعات برای ایجاد بانک رجیستر
۱۷۰	 ۵-۲-۲-۳ تست عملکرد بانک رجیستر
۱۷۲	 ۱۰ آزمایش
۱۷۲	 ۳-۲-۳ واحد محاسبه و منطق
۱۷۸	 ۴-۲-۳ پیاده‌سازی واحد محاسبه و منطق در Maxplus
۱۷۸	 ۱-۴-۲-۳ پیاده‌سازی تمام جمع‌کننده برای عملیات‌های جمع و تفریق در ALU
۱۸۱	 ۲-۴-۲-۳ پیاده‌سازی ALU یک بیتی برای طبقات 0 تا 30
۱۸۳	 ۲-۲-۳ پیاده‌سازی ALU یک بیتی برای طبقه آخر
۱۸۴	 ۲-۴-۲-۳ پیاده‌سازی ALU ۳۲ بیتی
۱۸۶	 ۵-۴-۲-۳ تست ALU
۱۸۹	 ۱۱ آزمایش
۱۸۹	 ۵-۲-۳ بررسی مسیر و کش دست‌رعمل
۱۹۰	 ۱-۵-۲-۳ رجیستر PC
۱۹۱	 ۲-۵-۲-۳ جمع‌کننده ۳۲ بیتی که یک ورودی آن عدد ثابت است
۱۹۴	 ۳-۵-۲-۳ پیاده‌سازی حافظه دستورات
۱۹۸	 ۴-۵-۲-۳ پیاده‌سازی مسیر واکنشی دستورات
۱۹۹	 ۵-۵-۲-۳ تست مسیر واکنشی دستورات
۲۰۴	 ۱۲ آزمایش
۲۰۴	 ۶-۲-۳ بررسی مسیر داده دستورات مراجعه به حافظه
۲۰۴	 ۱-۶-۲-۳ پیاده‌سازی واحد گسترش بیت علامت
۲۰۷	 ۲-۶-۲-۳ پیاده‌سازی حافظه داده‌ها
۲۰۹	 ۳-۶-۲-۳ ملاحظات پیاده‌سازی مسیر داده دستورات حافظه‌ای
۲۱۰	 ۴-۶-۲-۳ بررسی مسیر داده دستور شرطی
۲۱۱	 ۵-۶-۲-۳ پیاده‌سازی واحد شیفت دهنده
۲۱۳	 ۱۳ آزمایش
۲۱۳	 ۷-۲-۳ واحد کنترل

۲۱۳	 واحد کنترل ALU (۱-۷-۲-۳)
۲۱۵	 واحد کنترل اصلی (۲-۷-۲-۳)
۲۱۷	 آزمایش ۱۴
۲۱۷	 پیاده‌سازی مدار CPU (۸-۲-۲)
۲۲۱	 فصل چهارم - تست پردازنده
۲۲۲	 (۱-۴) تست پردازنده طراحی شده
۲۲۲	 (۱-۱-۴) توصیه‌های سخت‌افزاری
۲۲۷	 (۲-۱-۴) مثال‌های برنامه‌نویسی و تست پردازنده با برنامه نوشته شده
۲۲۸	 (۱-۱-۴) مثال ۱: محاسبه عمل جمع
۲۳۸	 (۲-۱-۴) مثال ۲: محاسبه n جمله اول سری فیبوناچی
۲۴۴	 (۱-۲-۱-۴) مثال ۳: محاسبه n امین جمله سری فیبوناچی
۲۴۸	 (۴-۲-۱-۴) مثال ۱: محاسبه عمل ضرب
۲۵۲	 (۳-۱-۴) توصیه‌های نرم‌افزاری
۲۵۳	 (۴-۱-۴) چند پروژه جهت توسعه پردازنده طراحی شده
۲۵۳	 (۱-۴-۱-۴) اضافه نمودن دستور add به مجموعه دستورات پردازنده
۲۵۳	 (۲-۴-۱-۴) اضافه نمودن دستور jump به مجموعه دستورات پردازنده
۲۵۴	 (۳-۴-۱-۴) اضافه نمودن دستور bne به مجموعه دستورات پردازنده
۲۵۵	 پیوست - مروری بر معماری و زبان اسمبلی پردازنده MIPS
۲۵۸	 (۱-۵) قالب دستورات پردازنده MIPS
۲۵۹	 (۱-۱-۵) قالب نوع R
۲۶۰	 (۲-۱-۵) قالب نوع I
۲۶۱	 (۳-۱-۵) قالب نوع J
۲۶۲	 (۲-۵) زبان اسمبلی MIPS
۲۷۳	 (۳-۵) نحوه استفاده کامپایلر از رجیسترها
۲۷۴	 (۴-۵) میدان‌های آدرس دستورات پرش شرطی و غیر شرطی در زبان ماشین

مقدمه

کامپیوتر از دو بخش اصلی سخت‌افزار و نرم‌افزار تشکیل شده است [۱]. سخت‌افزار به بخش‌های فیزیکی و قابل مشاهده کامپیوتر اطلاق می‌شود و نرم‌افزار به برنامه‌هایی گفته می‌شود که سخت‌افزار را قابل استفاده می‌نماید. مثال نرم‌افزار به سخت‌افزار همانند عقل و قوه تفکر انسان به کل اجزای قابل مشاهده او است [۲]. در دروس مهندسی کامپیوتر درس‌های فراوانی وجود دارند که برخی از آنها به مفاهیم سخت‌افزاری، برخی به مفاهیم نرم‌افزاری و برخی به هر دو مبحث می‌پردازند. در میان دروس، سخت‌افزاری، دروس مدارهای منطقی و معماری کامپیوتر مهم‌ترین دروس به شمار می‌رود به طوری که درک کامل آنها، کمک قابل توجهی بر درک مفاهیم سخت‌افزار دیگر و همچنین درک مؤثرتر مفاهیم نرم‌افزاری و همچنین نوشتن برنامه‌های کاربردی می‌نماید. دروس مدارهای منطقی و معماری کامپیوتر به صورت تئوری ارائه می‌گردند و برای اینکه مفاهیم مطرح شده در آنها ملموس‌تر و قابل فهم‌تر باشد، درس آزمایشگاه مدارهای منطقی و معماری کامپیوتر به عنوان مکمل این دو درس به کار گرفته می‌شود. بنابراین درس آزمایشگاه مدارهای منطقی و معماری کامپیوتر به صورت صد در صدی و به کمک دو شبیه‌ساز متفاوت ارائه می‌شود و در هر یک از آنها انواع مدارها مورد آزمون قرار می‌گیرند.

در بخش آزمایش‌های مدارهای منطقی، با استفاده از نرم‌افزار Proteus، انواع مدارهای ترکیبی نظیر تمام جمع‌کننده، انکودر با قدم ورودی، نمایشگر 7 Segment، دیکدر، جمع‌کننده BCD و سیستم گذرگاه شتیب و همچنین انواع مدارهای ترتیبی نظیر شمارنده دودویی، شیفت رجیستر، بافر اولین ورودی-اولین خروجی، حافظه لچ و حافظه اصلی طراحی، شبیه‌سازی و مورد تحلیل قرار می‌گیرند. در بخش آزمایش‌های معماری کامپیوتر با استفاده از نرم‌افزار MaxPlus II، طراحی، پیاده‌سازی، برنامه‌نویسی و تست یک پردازنده کامل از ابتدا تا انتها توضیح داده می‌شود. دانشجوی در بخش آزمایش‌های معماری کامپیوتر، طراحی و ساخت کلیه مدارهای سخت‌افزاری لازم را خودش به انجام رسانده و از نزدیک با نحوه کار

مدارهایی نظیر مالتی پلکسر، دیکدر، حافظه، ALU، بانک رجیسترها، مسیرهای داده و واحد کنترل و... آشنا می‌گردد. پس از ساخته شدن همه قسمت‌های سخت-افزاری، دانشجو باید کل پردازنده را تست نماید. تست کامل یک پردازنده فقط و فقط با یک برنامه که به زبان ماشین همان پردازنده نوشته شده است، امکان پذیر است. بهترین برنامه‌ها برای تست، برنامه‌هایی هستند که شامل همه دستورات پیاده‌سازی شده در سخت‌افزار و همچنین همه ترتیبات ممکن دستورات^۱ باشد.

هر پردازنده‌ای دارای معماری مجموعه دستورات یا ISA^۲ خاص خودش می‌باشد. ISA، عنوان یک نقطه مشترک و یک پروتکل مشترک (یک قرارداد مشترک) بین سخت‌افزار و نرم‌افزار سطح پایین (کامپایلر) عمل می‌کند. هر سخت‌افزاری مبتنی بر یک ISA طراحی می‌گردد. هر کامپایلری نیز برای یک ISA نوشته می‌شود تا دستوراتی که تولید می‌شوند بر روی آن معماری قابل اجرا باشد. ISA، قالب دستورات را که مشتمل بر دستورات مختلف دستور از جمله نوع دستور (opcode)، عملوندهای (operand) مبدأ و مقصد، لام برای این دستور و غیره را مشخص می‌نماید. همچنین، ISA شامل اطلاعاتی در ارتباط با تعداد مراحل پایپلین، نوع حافظه پشته، قراردادهای لازم در فراخوانی توابع و روتین‌های وقفه و غیره می‌باشد.

مباحث مطرح شده در بخش آزمایش‌های معماری کامپیوتر مربوط به پردازنده MIPS خواهد بود و ISA این پردازنده جهت طراحی باشد. سخت‌افزاری و نرم-افزاری مورد نظر خواهد بود. برای تست پردازنده، دانشجو باید با ISA پردازنده MIPS آشنا گردد و بتواند به زبان اسمبلی این پردازنده کامپایلر نماید. پس از اینکه برنامه به زبان اسمبلی نوشته شد، دانشجو باید بتواند به صورت دستی برنامه زبان اسمبلی را به برنامه زبان ماشین تبدیل نماید (می‌توان فرآیند تبدیل را به کمک یک ابزار اسمبلر MIPS نیز انجام داد). پس از آماده شدن برنامه به زبان ماشین، می‌توان کد باینری آن را به داخل حافظه پردازنده طراحی شده بار نمود و برنامه را توسط پردازنده اجرا کرد. اگر برنامه نوشته شده توسط دانشجو بر روی

^۱ - Instruction Order

^۲ - Instruction Set Architecture

پردازنده طراحی شده توسط خودش اجرا گردد، یکی از لحظات باشکوه برای آن دانشجو در طول دوره تحصیلش اتفاق خواهد افتاد؛ چرا که این دانشجو همه کارهای سخت‌افزاری و نرم‌افزاری لازم را با موفقیت انجام داده و مانند این است که خودش کامپیوتری ساخته که قابل استفاده می‌باشد.

درس آزمایشگاه مدارهای منطقی و معماری کامپیوتر را می‌توان به سه روش مختلف برگزار نمود:

۱- می‌توان از چند بُرد bread board استفاده نموده و همه بخش‌های لازم را بر روی آن پیاده‌سازی نمود. در این صورت هر ماژول سخت‌افزاری که مورد نیاز باشد باید بر روی بُرد قرار داده شود و سیم‌بندی بین ماژول‌ها انجام گیرد. در این روش، ماژول‌هایی نظیر حافظه، ALU، مدار تولید کلاک^۳، مدار تولید سیگنال، بازنشانی^۴، رجیسترها، واحد کنترل و غیره، همگی باید بر روی بُرد پیاده‌سازی گردد. برای تست پردازنده در این روش می‌توان برنامه نویسی و نوشتن برنامه به داخل حافظه را به طور دستی و یا در محیط PC انجام داد.

۲- استفاده از بُردهای آموزشی آماده‌شده؛ در این روش از بُردهایی استفاده می‌شود که بر روی آنها تراشه‌های قابل برنامه‌ریزی نظیر FPGA (در مورد این تراشه‌ها در ادامه این کتاب صحبت خواهد شد) قرار دارد. این بُردها از طریق یکی از پورتهای کامپیوتر نظیر پورت USB^۵ به کامپیوتر وصل می‌گردند و با استفاده از نرم‌افزاری که بر روی کامپیوتر نصب می‌گردد قابل کنترل هستند. این محیط نرم‌افزاری، یک محیط طراحی کامل بوده (IDE^۵) و تقریباً همه کارهای نرم‌افزاری و سخت‌افزاری لازم را می‌توان توسط آن انجام داد. دانشجو در محیط PC طراحی سخت‌افزار را با روش‌هایی که بعداً در مورد آن صحبت خواهد شد، انجام داده و پس از اتمام کار،

^۳ - Clock

^۴ - Reset

^۵ - Integrated Development Environment

مدار طراحی شده را با استفاده از محیط IDE و توسط کابل واسط بین کامپیوتر و بُرد، بر روی تراشه‌های قابل برنامه ریزی قرار می‌دهد.

۳- استفاده از نرم افزارهای شبیه سازی سخت افزار: در این روش می‌توان از ابزارهایی نظیر Proteus, MAXplus II و غیره استفاده نمود. این ابزارها، امکانات سخت افزاری قابل توجهی در اختیار قرار می‌دهند. از جمله این امکانات می‌توان به انواع گیت‌های منطقی، مالتی پلکسرها، دیکدرها، مالتی پلکسرها، رجیسترها و غیره اشاره نمود. هر گیت منطقی در این ابزارها به مانند گیت‌های واقعی دارای تأخیر می‌باشد و امکان کار موازی گیت‌ها وجود دارد. برنامه نویسی، شبیه سازی، تست و دیباگ از امکانات دیگر این گونه ابزارها می‌باشد.

از سه روش گفته شد، روش اول دارای پیچیدگی‌های زیادی است و امکان جواب گرفتن از پردازنده طراحی شده در این روش بسیار ضعیف است. در این روش، امکان خراب بودن تک تک بُردها و تراشه‌ها وجود دارد و به دلیل زیاد بودن تعداد سیم‌ها و اتصالات، احتمال خطا در سیم بندی‌ها نیز وجود دارد. همچنین، به دلیل درگیر شدن دانشجو با مسائل سطح پایین سخت افزار، امکان آموزش بسیاری از مفاهیم معماری کامپیوتر وجود ندارد.

در روش دوم دانشجو دیگر درگیر سیم بندی و بستن مدار بر روی بُرد نخواهد بود. دانشجو در سطحی بالاتر فقط با مسائل طراحی و مفاهیم معماری کامپیوتر درگیر خواهد بود. در این روش، سیم بندی‌ها و بستن مدار توسط ابزار بر روی تراشه‌های قابل برنامه ریزی انجام خواهد شد. این روش، از یک سو شبیه ریس اول بوده و بر روی سخت افزار واقعی انجام می‌گیرد و از یک سو به دلیل کارهایی که کاربر در محیط کامپیوتر می‌کند و خیلی وارد مسائل سطح پایین سخت افزاری (نظیر سیم بندی واقعی) نمی‌شود شبیه روش سوم می‌باشد.

روش سوم یک روش کم هزینه ولی با قابلیت‌های فراوان می‌باشد و برای منظورهای آموزشی بهترین گزینه به حساب می‌آید. در این روش، امکان انجام هر نوع طرحی وجود دارد و هر طور خواسته باشیم و با هر برنامه‌ای می‌توانیم طرح خود را تست

نمائیم. در این روش، هیچ وقت نگران اشتباهات خود نیستیم که مثلاً با یک اشتباه یک تراشه و یا بخشی از مدار بسوزد و از رده خارج گردد. همه کارهایی که با این روش انجام می‌گیرد به راحتی قابل استفاده در روش‌های دوم و سوم خواهد بود. به دلیل ساده بودن این روش، احتمال جواب گرفتن از پردازنده در این روش بسیار بالا خواهد بود. بنابراین، به دلیل مزیت‌های غیر قابل انکار روش سوم، مطالب مطرح شده در این کتاب مبتنی بر این روش خواهد بود.

مواردی که در این کتاب آموزش داده خواهد شد به شرح زیر خواهد بود:

- فصل ۱: مقدمه‌ای بر طراحی سخت افزار. در این فصل مراحل مختلف طراحی، پیاده‌سازی و ساخت مدارات سخت افزاری مورد بررسی قرار خواهد گرفت. بنابراین نگاه نوایم داشت بر تراشه‌های قابل برنامه‌ریزی، پروسه طراحی سخت‌افزار و ابزارهای مورد استفاده در پروسه طراحی.
- فصل دوم: معرفی نرم‌افزار Proteus. در این فصل ابتدا نرم‌افزار Proteus بطور کامل معرفی می‌شود و سپس با استفاده از این نرم‌افزار آزمایش‌های متنوعی برای شبیه‌سازی و تست مدارهای منطقی صورت می‌گیرد. بخش‌های مختلف این فصل به شرح زیر خواهد بود:
 - آزمایش ۱: طراحی تمام جمع‌کننده و تست آن به کمک مدار ورودی
 - آزمایش ۲: شبیه‌سازی عملکرد انکودر و تست آن به کمک مدار ورودی
 - آزمایش ۳: معرفی عملکرد نمایشگر ۷ Segment و کار بردن تکنیک‌های مختلف در ساده‌سازی مدار
 - آزمایش ۴: طراحی شمارنده دودویی و تست آن
 - آزمایش ۵: طراحی جمع‌کننده BCD و تست آن به کمک مدار ورودی
 - آزمایش ۶: طراحی سیستم گذرگاه مشترک به کمک بافرهای سه حالت و تست آن
 - آزمایش ۷: طراحی بافر اولین ورودی - اولین خروجی و تست آن

○ آزمایش ۸: شبیه‌سازی عملکرد حافظه اصلی و تست آن

- فصل سوم: معرفی و کاربرد نرم‌افزار MAXplus II. در این فصل ابتدا نرم‌افزار MAXplus II بطور کامل معرفی می‌شود و سپس به کمک این نرم‌افزار پردازنده MIPS به طور سلسله‌مراتبی شبیه‌سازی می‌شود. اغلب بخش‌های مهم پردازنده در حین طراحی از لحاظ عملکرد صحت‌سنجی می‌شوند. بخش‌های مختلف این فصل به شرح زیر خواهد بود:

○ آزمایش ۹: طراحی بانک رجیستر و تست آن

○ آزمایش ۱۰: طراحی ALU و تست آن

○ آزمایش ۱۱: طراحی مسیر واکنشی دستورات و تست آن

○ آزمایش ۱۲: طراحی حافظه داده، واحد گسترش بیت علامت، مدار شیفت‌دهنده

○ آزمایش ۱۳: طراحی واحد کنترل ALU و واحد کنترل مرکزی

○ آزمایش ۱۴: طراحی پردازنده MIPS به کمک مدارهای آزمایش‌های ۹ تا ۱۳.

- فصل چهارم: تست پردازنده MIPS. از کارهای مهمی که باید در سیستم‌های نرم‌افزاری و سخت‌افزاری به نحو شایسته‌ای انجام دهیم، تست می‌باشد. به همین دلیل فصل جداگانه‌ای در این کتاب برای این منظور تعبیه شده است. در این فصل ابتدا مقدمه‌ای در ارتباط با برنامه‌های اسمبلی MIPS خواهیم داشت و سپس ۴ برنامه نمونه به این زبان جهت تست پردازنده آورده خواهد شد. چند راهنمایی سازنده در ارتباط با تست پردازنده طراحی شده نیز در انتهای این فصل بیان خواهد شد.

- پیوست: مروری بر معماری MIPS. ساختار آزمایش‌های معماری کامپیوتر در این کتاب بر اساس معماری پردازنده MIPS است. بنابراین، در این بخش به صورت اجمالی معماری این پردازنده توضیح داده خواهد شد.