پیش‌گفتار

چرا ARM؟

"میکروکنترلرهای بر مبنای پردازنده‌ی ARM، قطعاتی ۱۶/۳۲ بیتی با کارایی بالا هستند. در میان میکروکنترلرها، بازار ARM به شدت در حال رشد است. قیمت میکروکنترلرهـای ARM در حال حاضر مشابه قطعات ۸ بیتی است، درحالی‌که توانایی و سخت‌افزارهای جنبـی آن‌هـا بـسیار بیشتر است.

سازندگان مختلفی ایـن میکروکنترلرهـا را عرضـه مـی‌کننـد : Atmel، Analog Devices، Cirrus Logic، OKI، Philips، ST، TI، Intel، Motorola، Samsung، Sharp و غیره.

بسیاری کارشناسان عقیده دارند که در ۵ سال آینده، تقریبـاً در تمـام کاربردهـا، میکروکنترلرهـای ARM جایگزین استاندارد صنعتی 8051 خواهند شد.

مزایای ARM

- بسیار سریع : بیشتر هسته‌های ARM7 می‌تواننـد تـا فرکـانس 60 MHz کـار کننـد و هسته‌های ARM9 نیز سرعتی بیش از 150 MHz دارند که در عمـل، توانـایی آن‌هـا از یک پردازنده‌ی 386 بیشتر خواهد بود.
- توان مصرفی کم : هسته‌ی ARM تنها بین 0.5-1 mA به ازای هر MHz جریان می‌کشد.
- سـخت‌افزارهـای جنبـی متعـدد : ADC، DAC، USB، SPI، UART، I2C، CAN، Ethernet، SDRAM.
- حافظه‌ی داخلی زیاد : 32 KB-1 MB حافظه‌ی فلش و 4-256 KB حافظه‌ی RAM

اشکال ARM

برای فرد مبتدی، پیچیده است؛ قطعاً نباید ARM اولین گام شما در کار با میکروکنترلرها باشد[1]."

نقل قول از سایت شرکت Olimex (سازنده‌ی ابزارها و بردهای آموزشی).

چرا LPC2000؟

شرکت‌های مختلفی با استفاده از پردازنده‌های ARM7، میکروکنترلر تولید کرده‌اند، در این کتاب از قطعات سری LPC2000 شرکت NXP (تأسیس شده به‌وسیله‌ی فیلیپس) استفاده می‌کنیم. این مسئله دلایل فراوانی دارد که برخی از آن‌ها عبارتند از :

۱. قطعات سری LPC2000 یکی از متنوع‌ترین خانواده‌های میکروکنترلرهای با هسته‌ی ARM7 هستند و قطعات موجود در آن، در مقایسه با سایر میکروکنترلرهای مبتنی بر ARM7 قیمت کمتری دارند. به عنوان مثال قیمت قطعه‌ی LPC2101 حدود ۲ دلار است که این از بسیاری از میکروکنترلرهای ۸ بیتی (مانند ATmega16) کمتر است.

۲. اجرای برنامه از حافظه‌ی فلش بسیار سریع است. بدلیل دسترسی ۱۲۸ بیتی به حافظه‌ی فلش و وجود واحد شتاب‌دهنده‌ی حافظه، قطعات سری LPC2000 می‌توانند در مُد ARM با حداکثر سرعت ۶۰ تا ۷۵ MHz به حافظه‌ی فلش دسترسی داشته باشند؛ این در حالی است که سایر قطعات مشابه با سرعتی کمتر از نصف این مقدار کد برنامه را اجرا می‌کنند. علاوه براین، در مقایسه با سایر قطعات، فرکانس کاری میکروکنترلرهای LPC2000 نسبتاً بالاست (جداول ۲.۱ و ۳.۱ را ملاحظه کنید).

۳. راه‌اندازی سخت‌افزاری جنبی قطعات LPC2000 ساده‌تر است. اکثر سخت‌افزارهای جنبی به شکلی طراحی شده‌اند که لازم است رجیسترهای کمتری تنظیم شوند و بسیاری از آن‌ها را می‌توان به حالت پیش‌فرض رها کرد. این مسئله باعث محبوبیت این میکروکنترلرها شده است و به همین دلیل بسیاری از افراد ترجیح می‌دهند کار با میکروکنترلرهای ARM را با این قطعات شروع کنند.

۴. میکروکنترلرهای LPC2000 دارای برخی سخت‌افزارهای جنبی هستند که به ندرت در سایر میکروکنترلرهای با هسته‌ی ARM7 دیده می‌شود. مثلاً تایمر ۳۲ بیتی با

---

۱. نظر مولف این است که در چند سال آینده اینطور نخواهد بود و ARM می‌تواند اولین تجربه‌ی کار با میکروکنترها باشد، همانطور که زمانی توصیه می‌شد AVR برای شروع مناسب نیست و در حاضر بسیاری از افراد کار با میکروکنترلرها را با AVR آغاز می‌کنند!

پیش‌تقسیم‌کننده‌ی ۳۲ بیتی، DAC، RTC، LIN، SSP، MMC/SD Controller، USB Host/OTG، Fast GPIO، XGA LCD Controller و غیره.

۵. قطعات LPC دارای تعداد I/O بیشتری هستند. مثلاً قطعه‌ی LPC2132 که یک قطعه‌ی ۶۴ پایه است دارای ۴۷ پایه‌ی GPIO است در حالیکه بسیاری از قطعات مشابه ۶۴ پایه، دارای ۳۲ خط I/O هستند.

۶. شرکت NXP (سازنده‌ی قطعات LPC2000) مستندات مناسبی برای میکروکنترهای با هسته‌ی ARM ارائه کرده است. در وب‌سایت این شرکت، برای راه‌اندازی بسیاری از سخت‌افزارهای جنبی نکات کاربردی (Application Note) ارائه شده و برای هر قطعه، یک مجموعه کد (Code Bundle) موجود است که تمام سخت‌افزارهای جنبی در آن راه‌اندازی شده است.

۷. امکان شبیه‌سازی قطعه و اشکال‌زدایی[۱] برنامه در نرم‌افزار پروتئوس وجود دارد. اگرچه نهایتاً لازم است برنامه را بر روی سخت‌افزار آزمایش کنید اما شبیه‌سازی امکان ارزشمندی در یادگیری میکروکنترلر است. خوشبختانه نرم‌افزار پروتئوس از هسته‌ی ARM7 و بسیاری از قطعات خانواده‌ی LPC2000 پشتیبانی می‌کند، این مسئله می‌تواند نقش مؤثری در تسریع یادگیری میکروکنترلرهای ARM داشته باشد.

---

۱. برای اشکال‌زدایی برنامه کافی است در IAR Embedded Workbench خروجی با فرمت elf ایجاد کنید و این فایل را در پروتئوس به جای فایل hex. بارگذاری کنید. از آنجایی که ممکن است برخی از افراد دسترسی به دیباگر سخت‌افزاری نداشته باشند این قابلیت نیز می‌تواند برای آن‌ها مفید واقع شود.

مقدمه

میکروکنترلرهای ۸ بیتی به دلیل ساختار و معماری ساده‌شان بسیار جذاب و محبوب‌اند، اما کاربردهایی زیادی وجود دارد که در توانایی این میکروکنترلرها نیست. در این مواقع نیاز به یک میکروکنترلر سریع‌تر و با امکانات بیشتر مانند میکروکنترلرهای با هسته‌ی ARM احساس می‌شود. از سوی دیگر به دلیل اینکه انتظار کاربران سیستم‌های الکترونیکی روزبه‌روز بیشتر می‌شود، نیاز به طراحی سیستم‌هایی است که دارای واسطِ کاربر پسند، توان مصرفی کم، امکانات بیشتر و قیمت مناسب‌تر باشند؛ بنابراین پیچیدگی سخت‌افزار و نرم‌افزار این سیستم‌ها رو به افزایش است. در این شرایط لازم است تا طراحی سیستم‌ها با روش‌ها و ابزارهای به‌روز و کارآمدتری انجام شود.

در صنعت کشورمان صورت مسئله‌های زیادی وجود دارد که راه‌حلشان می‌تواند در استفاده از امکانات میکروکنترلرهای ARM باشد؛ بنابراین بسیاری از متخصصان و طراحان، آشنایی با این قطعات را در برنامه‌ی خود قرار داده‌اند. مشکلی که برای این دسته از افراد وجود دارد کمبود (و یا نبود) اطلاعات در مورد این میکروکنترلرهاست و این باعث شده است که از این میکروکنترلرهای قدرتمند در کشورمان کمتر استفاده شود. آنچه من و امثال من در مورد میکروکنترلرهای ARM آموختیم از طریق منابع اندک و صرف وقت و تلاش فراوان بود اما اکنون بسیار خرسند از این که این دانسته‌ها مکتوب شده تا شاید دیگران بتوانند این مسیر را سریع‌تر طی کنند.

تلاش مؤلف بر این بوده است که این کتاب، کاربردی باشد؛ به این منظور، در انتهای بیشتر بخش‌ها، نمونه برنامه‌ای وجود دارد. این نمونه برنامه‌ها تماماً بر روی سخت‌افزار آزمایش شده و برای افرادی که دسترسی به سخت‌افزار ندارد، شبیه‌سازی نیز شده‌اند. نکته‌ی دیگر این است که کتاب حاضر، مکمل مستندات ارائه شده توسط سازنده‌ی قطعه و کامپایلر است تا افراد بتوانند با صرف کمترین زمان، اطلاعات موجود در برگه‌داده (Data Sheet)، راهنمای کاربر (User Manual) و نکات کاربردی (Application Note) را جمع‌بندی کرده و به‌کار گیرند. به عبارت دیگر در طی فصول این کتاب مطالبی که از طریق سایت سازنده‌ی قطعه و راهنمای کامپایلر قابل تهیه است، کمتر یافت می‌شود و به موازات مطالعه‌ی کتاب لازم است از مستندات ارائه شده توسط سازنده‌ی قطعه و کامپایلر

استفاده کنید[1]. این سبک جدیدی در تألیف کتاب‌های میکروکنترلر در کشورمان است و شیوه‌ی ارائه‌ی بسیاری از کتاب‌های موفق دنیاست. با این حال مولف از نظرات شما در مورد سبک، محتوای کتاب و اشکالات آن استقبال می‌کند. لطفاً مطالب مربوط به این موارد را به آدرس books@microsam.ir یا info@fadakbook.ir ارسال نمایید تا در ویرایش‌های بعدی مدنظر قرار گیرند. در ضمن خوانندگان علاقمند می‌توانند پرسش‌های علمی خود را در تالار گفتگوی forum.microsam.ir مطرح فرمایند.

در پایان از آقای مهندس رضا کرمی‌شاهنده مدیریت پژوهش و تولید، سرکار خانم مریم یوزباشی به خاطر صفحه‌آرایی زیبای متن، مدیریت محترم انتشارات، جناب آقای مجیدرضا زروئی به خاطر تلاش و حمایت‌های جدی از این کار، سپاسگزاری می‌نمایم.

رضا سپاس‌یار

پاییز ۸۸

---

۱. تمام منابعی که غیر از این کتاب به آن‌ها نیاز خواهید داشت در CD همراه کتاب وجود دارند.

فهرست مطالب

فصل ۱ آشنایی با پردازنده‌های ARM	**۱-۱۴**
۱.۱ تاریخچه‌ی ARM	۳
۲.۱ معماری ARM	۳
۳.۱ انواع معماری، خانواده و هسته‌ی CPU	۳
۴.۱ مجموعه دستورالعمل‌های Thumb	۶
۵.۱ خط لوله‌ی دستورالعمل (Pipeline)	۷
۶.۱ رجیسترهای CPU	۸
۷.۱ اکسپشن (Exception) و مُدهای کاری	۱۱
۸.۱ میکروکنترلرهای مبتنی بر پردازنده‌های ARM	۱۲

فصل ۲ آشنایی با میکروکنترلرهای LPC2000	**۱۵-۴۴**
۱.۲ مقدمه	۱۷
۲.۲ ساختار گذرگاه (Bus)	۱۷
۳.۲ نقشه‌ی حافظه (Memory Map)	۱۹
۴.۲ واحد شتاب‌دهنده‌ی حافظه	۲۱
۵.۲ منابع کلاک و PLL	۲۴
۶.۲ مقسم VPB	۳۳
۷.۲ پروگرام کردن حافظه‌ی فلش میکروکنترلر	۳۴
۸.۲ منابع ریسِت (Reset)	۳۷
۹.۲ مدیریت توان مصرفی	۳۸
۱۰.۲ بلوک اتصال پین	۳۸
۱۱.۲ وقفه‌ها در خانواده‌ی LPC2000	۳۹

فصل ۳ آموزش کوتاه IAR Embedded Workbench	**۴۵-۵۹**
۱.۳ چرا IAR Embedded Workbench؟	۴۷
۲.۳ ساختار کامپایلرهای C/C++	۴۷
۳.۳ لزوم برنامه‌نویسی به سبک ماژولار	۴۸
۴.۳ مفهوم ماژول نرم‌افزاری	۴۹
۵.۳ ایجاد پروژه در EWARM	۵۱
۶.۳ افزودن فایل‌ها به پروژه	۵۲
۷.۳ افزودن فایل startup	۵۳
۸.۳ انجام تنظیمات پروژه	۵۴
۹.۳ کامپایل کردن ماژول‌ها و ساخت فایل نهایی	۵۵
۱۰.۳ شبیه‌سازی و اشکال‌زدایی برنامه	۵۶
۱۱.۳ ایجاد خروجی HEX و LIST	۵۷

فصل ٤ وسایل جنبی داخلی	**۶۱-۱۱۰**
۱.۴ مقدمه	۶۳
۲.۴ پورت‌های موازی با کاربرد عمومی	۶۳
۳.۴ تایمر/کانترهای با کاربرد عمومی	۸۲
۴.۴ واحد UART	۸۹
۵.۴ مبدل آنالوگ به دیجیتال	۹۶
۶.۴ مبدل دیجیتال به آنالوگ	۱۰۲
۷.۴ مدولاتور PWM	۱۰۵

فصل ۵ نمونه پروژه‌ی کاربردی (کنترل موتور DC)	**۱۱۱-۱۳۴**
۱.۵ مقدمه	۱۱۳
۲.۵ معرفی موتورهای DC جاروبک‌دار	۱۱۳
۳.۵ راه‌اندازی موتورهای DC جاروبک‌دار	۱۱۴
۴.۵ کنترل سرعت	۱۱۶
۵.۵ فیدبک جریان	۱۱۶

۶.۵ اندازه‌گیری سرعت موتور ۱۱۷
۷.۵ انتخاب MOSFET ۱۱۸
۸.۵ راه‌انداز MOSFET ۱۱۹
۹.۵ کنترل سرعت و جهت ۱۲۱
۱۰.۵ شماتیک اتصالات ۱۲۱
۱۱.۵ عملکرد برنامه ۱۲۴
۱۲.۵ کدهای برنامه ۱۲۵

فصل ۶ طراحی یک سیستم‌عامل بی‌درنگ (RTOS) **۱۳۵-۱۵۸**
۱.۶ مقدمه ۱۳۷
۲.۶ ارزیابی معماری حلقه‌ی بی‌پایان ۱۳۷
۳.۶ راهکار مناسب‌تر ۱۳۹
۴.۶ انواع Scheduler ۱۴۰
۵.۶ طراحی یک Scheduler نوع Co-operative ۱۴۱

منابع **۱۵۹**

پیوست الف – مشخصات میکروکنترلرهای LPC **۱۶۱**

پیوست ب – بخشی از برگه‌داده‌ی LPC2101/02/03 **۱۷۱**

پیوست ج – بخشی از برگه‌داده‌ی LPC2141/42/44/46/48 **۱۸۱**

پیوست د – برنامه‌های فصل چهارم(بصورت تک‌فایلی) **۱۹۵**

واژه‌نامه **۲۰۷**

فهرست الفبایی(انگلیسی) **۲۱۱**

فهرست الفبایی(فارسی) **۲۱۷**

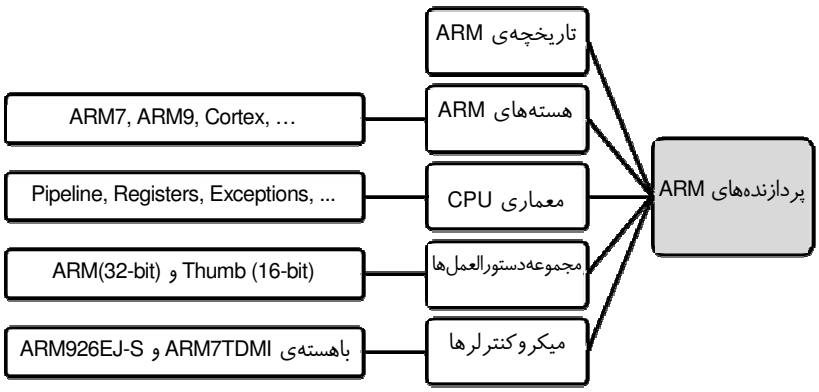
فصل

۱

آشنایی با پردازنده‌های ARM

Introduction to ARM Processors

ساختار فصل:



ARM نخستین نمونه‌ی تجاری از پردازنده‌های RISC است. معماری ARM به دلیل سادگی، قیمت مناسب، توان مصرفی کم و کارایی بالا از موفق‌ترین و مطرح‌ترین پردازنده‌های ۳۲ بیتی در دنیاست. در این فصل ضمن بررسی خانواده‌ها و هسته‌های مختلف ARM به معرفی کلی این پردازنده‌ها و معماری آنها می‌پردازیم. در پایان فصل دو نمونه از میکروکنترلرهای پرکاربرد که از هسته‌ی ARM استفاده می‌کنند معرفی خواهند شد.