

**به نام خدا**

**زهرا معروفی**

**گزارش مربوط به چالش دوم**

**(آشنایی با معماری‌های ARM ، x86 و MIPS)**

# معماری‌های ARM ، x86 و MIPS

## ۱ – معماری ARM

### مقدمه

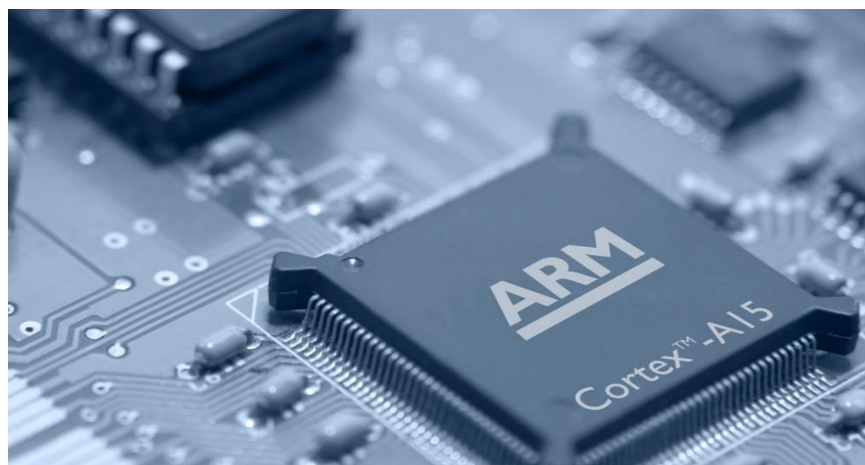
پس از انقلاب گوشی‌های هوشمند، استفاده از پردازنده‌های مبتنی بر معماری ARM بسیار افزایش پیدا کرده است، اما معماری ARM چیست و پردازنده‌های ARM چه ویژگی‌هایی دارند.

پس از آنکه گوشی‌های هوشمند ، موج جدیدی در دنیای فناوری ایجاد کردند و الگوی استفاده از گجت‌های دیجیتال را تغییر دادند، پردازنده‌ها نیز شاهد تغییر و تحولات و پا گرفتن معماری جدیدی بودند. با وجود تسلط اینتل و AMD بر پردازنده‌های مورد استفاده در پی‌سی که مبتنی بر معماری x86 هستند ( که در ادامه بررسی خواهد شد) ، گوشی‌های هوشمند توسط پردازنده‌های مبتنی بر معماری ARM قبضه شده‌اند. اهمیت معماری ARM و پردازنده‌های مبتنی بر این معماری به اندازه‌ای افزایش یافته است که مایکروسافت نیز درصدد همکاری با کوالکام، بزرگ‌ترین تولیدکننده تراشه‌های مبتنی بر معماری ARM برای گجت‌های موبایل، به‌منظور پشتیبانی از این تراشه‌ها در ویندوز ۱۰ است.

پردازنده‌های ARM توسط کمپانی Acorn Computers بر اساس معماری مبتنی بر RISC در دهه‌ی ۸۰ میلادی توسعه یافتند. Acorn Computers یک کمپانی بریتانیایی است که در سال ۱۹۷۸ در کمبریج انگلستان آغاز به کار کرد و محصولاتی نیز به بازار فرستاد که از جمله‌ی آن می‌توان به کامپیوتر BBC Micro اشاره کرد. در سال ۱۹۹۸ میلادی این کمپانی بریتانیایی تفکیک شد و زیرمجموعه‌های آن، امروزه به‌صورت مستقل به فعالیت می‌پردازند. یکی از موفق‌ترین زیرمجموعه‌های آکورن، Advanced RISC Machines نام دارد که بیشتر با عنوان ARM شناخته می‌شود. ARM در سال‌های اخیر به‌صورت مستقل فعالیت می‌کرد، اما کمپانی ژاپنی SoftBank چندی پیش با پرداخت بیش از ۳۲ میلیارد دلار، این کمپانی بریتانیایی را تصاحب کرد.

ARM Ltd یک کمپانی یا به بیان بهتر یک لابراتوار است که متخصصان آن به طراحی پردازنده مشغول هستند. البته عبارت اختصاری ARM دو مفهوم را شامل می‌شود. در مورد معماری و طراحی پردازنده‌های ARM ، این عبارت مخفف Acorn RISC Machines است، حال آنکه اگر منظور کمپانی توسعه‌دهنده‌ی این معماری باشد،

اختصار ARM کوتاه شده‌ی عبارت Advances RISC Machines است. ARM به هیچ وجه دستی در تولید تراشه ندارد و تنها معماری ARM را طراحی می‌کند، حال آنکه کمپانی‌هایی نظیر کوالکام، اپل و سامسونگ با دریافت گواهی استفاده از معماری آرم، تراشه‌های اختصاصی خود را مبتنی بر این معماری توسعه می‌دهند. البته این سه کمپانی شاخص‌ترین نام‌هایی هستند که گواهی معماری آرم را در پردازنده‌های خود استفاده می‌کنند و تعداد تولیدکنندگان پردازنده با استفاده از این معماری بالا است. اغلب گجت‌های الکترونیکی کوچک که از وجود باتری برای تأمین انرژی استفاده می‌کنند، از پردازنده‌های مبتنی بر آرم در واحد پردازشی خود بهره می‌برند.



## تاریخچه

معماری ARM که در اصل مخفف Acorn RISC Machine است، اولین بار توسط شرکت بریتانیایی Acorn computer توسعه یافت. نکته جالب اینکه این معماری جهت پردازش در رایانه‌های شخصی و در دهه ۱۹۸۰ توسعه پیدا کرده است. شرکت Acorn یکی از چندین استار آپ کوچکی بود که در دوره تب مایکرو رایانه‌ها در کمبریج تاسیس شد. در سال ۱۹۸۱ شرکت Acorn یک رایانه به نام BBC Micro برای شبکه بی بی سی ساخت تا آن‌ها با تولید یک سریال تلویزیونی به افزایش سواد رایانه‌ای در سراسر بریتانیا کمک کنند. در سال ۱۹۸۴ هشتاد درصد از مدارس سراسر بریتانیا، حداقل یک مدل از این رایانه را تهیه کرده بودند. از همان زمان به نظر می‌رسید این شرکت آینده درخشانی خواهد داشت.

اما با اضافه شدن رابط گرافیکی به رایانه های Acorn ، آن ها نیاز داشتند تا سرعت پردازش را افزایش دهند. خرید پردازنده‌ی میکرو از دیگر شرکت‌ها کار عاقلانه‌ای نبود زیرا آن پردازنده‌ها به اندازه‌ی کافی سریع نبودند. همچنین طراحی یک پردازنده برای شرکت کوچکی مثل Acorn آن هم تنها با ۴۰۰ کارمند، کار بسیار سختی به نظر می‌رسید. آن‌ها باید به ساختار ساده تری در طراحی دست پیدا می‌کردند که بهره وری بالایی نیز داشته باشد. در نهایت طراحان Acorn به ساختار تازه‌ای دست پیدا کردند و آن را “reduced instruction set computing” یا RISC نامیدند. رایانه‌هایی که با این پردازنده تولید شدند، Acorn RISC Machin یا ARM نام گرفتند. یکی از اولین کاربردهای تجاری، پردازشگرهای این شرکت در همکاری با شرکت اپل بود. اپل در حال ساخت نوعی رایانه قابل حمل به نام نیوتن که پدر تبلت های امروزی محسوب می شد، بود. پردازنده ARM بر روی نیوتن به خوبی جواب داد، زیرا این نوع پردازنده با توجه به قدرت پردازشی که فراهم می کرد، مصرف بسیار اندکی داشت. نیوتن که در زمان خود دستگاهی انقلابی محسوب می شد، نگاه‌ها را به معماری ARM معطوف کرد. گرچه این دستگاه حدود ۲۰ سال قبل از عصر تلفن های هوشمند، هیچوقت موفقیت تجاری بزرگی را برای اپل به همراه نداشت، اما نشان داد که می توان با استفاده از پردازنده های مبتنی بر معماری ARM دستگاه های قابل حمل قدرتمندی را تولید کرد.



**BBC Micro**

گرچه معماری خاص پردازنده های ARM ، آن ها را به بهترین گزینه برای دستگاه های قابل حمل تبدیل می کرد، اما همین معماری خاص باعث ضربه خوردن آن نیز شد و این نوع پردازنده ها را از بازار رایانه ها دور نگه داشت. انحصار وینتل ( مایکروسافت ویندوز و پردازنده های اینتل ) ، هرگز در بازار پردازش رایانه ها با خطری از سوی پردازنده های ARM مورد تحدید قرار نگرفت. متأسفانه طراحی این نوع پردازنده از نیاز بازار کمی جلو تر بود و در آن زمان شرکت ها قادر به ساختن دستگاه های قابل حمل انقلابی نبودند و به همین دلیل شرکت Acorn تحت سلطه بازار رایانه های شخصی برای مدتی از انظار محو شد. البته آن ها به فعالیت های خود در زمینه ساخت رایانه برای مدارس و دستگاه های همراه غیر هوشمند ادامه دادند تا این که سال ۲۰۰۷ فرا رسید. اپل با معرفی آیفون که در درون خود از پردازنده با معماری ARM که به دست سامسونگ ساخته شده بود، بهره می برد، دوران سلطنت این معماری بر پردازش موبایلی را آغاز کرد.

## طراحی ARM مبتنی بر معماری RISC

همان طور که اشاره کردیم، ARM پردازنده های خود را بر اساس دستورات RISC توسعه می دهد، اما RISC چیست؟ RISC مخفف عبارت Reduced Instruction Set Computing است. برخلاف پردازنده های ARM که از این معماری استفاده می کنند، پردازنده های اینتل و ای ام دی که قدرت پردازشی در لپ تاپ و پی سی شما را تأمین می کنند، از معماری CISC یا Complex Instruction Set Computing استفاده می کنند. دو معماری RISC و CISC برای استفاده در کاربردهای متفاوت طراحی شده اند. یک پردازنده مبتنی بر معماری RISC برای این منظور طراحی شده است تا تعداد دستورات ارسالی به پردازنده از سوی برنامه در حال اجرا کاهش یابد. در واقع مجموعه ی دستورات مورد استفاده در معماری RISC بسیار پایین تر است. با توجه به اینکه تعداد دستورات ارسالی در معماری RISC کاهش پیدا کرده، فرکانس پردازشی بالا است و پردازنده می تواند در هر ثانیه دستورات بیشتری در مقایسه با CISC اجرا کند.

زمانی که مجموعه ی دستورات اجرایی توسط پردازنده کاهش پیدا کند، پیچیدگی پردازنده نیز کاهش می یابد و می توان مدار تراشه را به شکل ساده تری طراحی کرد. پردازنده های RISC دارای ترانزیستورهای کمتری هستند که همین موضوع منجر به کاهش انرژی مصرفی توسط پردازنده می شود. سادگی طراحی پردازنده در کنار کاهش تعداد ترانزیستورها نتیجه ای جز کاهش سایز تراشه ندارد. سایز تراشه به سطح مقطعی اطلاق می شود که روی

ویفر سیلیکونی برای ساخت یک پردازنده تخصیص داده می‌شود. نتیجه کاهش سایز، امکان اضافه کردن کامپوننت‌های بیشتر روی پردازنده با اتصالات کمتر است، از این‌رو پردازنده‌های ARM کوچک‌تر هستند و انرژی کمتری مصرف می‌کنند.

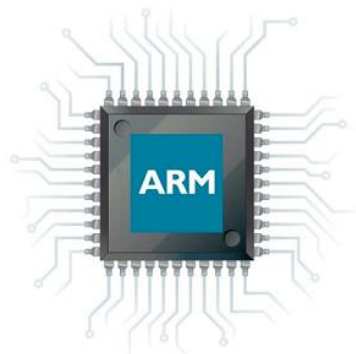
پردازنده‌های سریع، کوچک و کم‌مصرف بهترین گزینه برای استفاده در گوشی‌های هوشمند هستند. گوشی‌های هوشمند این روزها قدرت پردازشی بالایی دارند. سیستم‌عامل‌ها و اپلیکیشن‌های توسعه‌یافته برای گوشی‌های هوشمند نیز، به‌منظور کاهش دستورات ارسالی به پردازنده، بهینه شده‌اند تا بهترین نتیجه در زمان به‌کارگیری تراشه‌های ARM حاصل شود.

البته تمام مواردی که در بالا به آن‌ها اشاره کردیم به این معنی نیست که پردازنده‌های ARM قدرت پردازشی محدودی دارند. معماری ARM امروزه امکان استفاده از طراحی ۳۲ و ۶۴ بیتی را در اختیار کاربران قرار می‌دهد. همچنین باید به مجموعه‌ای از قابلیت‌های دیگر اشاره کرد که علاوه بر گوشی‌های هوشمند، پردازنده‌های ARM را برای استفاده در کاربردهای دیگری نظیر ابررایانه‌ها مناسب می‌کند.

نسبت عملکرد به ازای هر وات از انرژی مصرفی در پردازنده‌های ARM بسیار قابل قبول است. در صورتی که نرم‌افزار توسعه‌یافته برای پردازنده‌های ARM بهینه باشد، این پردازنده‌ها می‌توانند عملکرد بهتری در مقایسه با معماری x86 ارائه کنند، از این‌رو اهمیت پردازنده‌های ARM در کاربردهایی نظیر به‌کارگیری آن‌ها در سرورها و ابررایانه‌ها بیش از پیش پررنگ می‌شود.

## پردازنده‌ی ARM و ویژگی‌های آن

آرم هولدینگز (ARM Holdings plc) یک شرکت نیمه‌هادی سطح اول است که محصولات آن پردازنده‌ها، سیستم‌های برون‌تراشه‌ای (System-on-Chips)، نرم‌افزارهای کامپیوتری و غیره است. معماری آرم یا ARM که مخفف Advanced RISC Machine به معنی ماشین‌های RISC پیشرفته است، در سال ۱۹۹۰ ساخته شد RISC یا ریسک، مخفف Reduced Instruction Set Computer به معنی رایانه کم‌دستور و نوع معماری ساخت کامپیوتر یا ریزپردازنده است که در آن به جای استفاده از دستورالعمل‌های خاص سایر معماری‌ها، از یک مجموعه دستورالعمل حداقلی و بسیار بهینه‌سازی شده استفاده می‌شود. ریسک (RISC) اصلی‌ترین استراتژی طراحی CPU است که در پردازنده‌های آن اجرا می‌شود.



ARM ارائه دهنده‌ی پیشرو ریزپردازنده‌های مبتنی بر RISC و سایر IP های نیمه‌هادی با بیش از ۸۵ میلیارد تراشه مبتنی بر ARM است. ARM مانند سایر شرکت‌های تولید ریزپردازنده همچون اینتل (Intel)، هیتاچی (Hitachi)، فری‌اسکیل (Freescale) و غیره، پردازنده یا سایر ادوات نیمه‌هادی را تولید نمی‌کند بلکه، به عنوان مالکیت فکری (IP) مجوز ساخت هسته‌های نیمه‌هادی را به سایر شرکت‌های نیمه‌هادی مانند اتمل (ATMEL)، فیلپس (اکنون NXP)، سامسونگ (Samsung) و غیره می‌دهد. مهم‌ترین مالکیت‌های فکری ARM، شامل ریزپردازنده‌های RISC توان‌پایین، کم‌هزینه و با کارایی بالا، سیستم برون‌تراشه‌ای و سایر لوازم جانبی است.

جدا از پردازنده‌ها و مالکیت فکری هسته‌ها، ARM ابزارهای توسعه نرم‌افزاری کاملی مانند Keil و DS-5 را برای توسعه سیستم‌های کامل مبتنی بر میکروکنترلر ARM و سیستم برون‌تراشه‌ای ارائه می‌دهد.

امروزه پردازنده‌های ARM تقریباً در هر زمینه‌ای مانند دستگاه‌های الکترونیکی دستی، دستگاه‌های برقی، رباتیک، اتوماسیون و غیره یافت می‌شوند. پردازنده‌های تولید شده از مالکیت فکری ARM در سیستم‌های تعبیه شده یا توکار یا نهفته (Embedded) مانند تلویزیون‌های هوشمند، ساعت‌های هوشمند، تبلت‌ها و غیره استفاده می‌شوند.

پردازنده‌های ARM مبتنی بر معماری رایانه کم‌دستور یا ریسک (RISC) هستند، اما بر اساس الزامات سیستم‌های تعبیه شده، برخی از اصلاحات نیز در معماری RISC انجام می‌شود. پردازنده‌های ARM معماری نوع انتقال داده (Load-Store) را دنبال می‌کنند که پردازش داده‌ها فقط بر روی محتویات ثابت‌ها یا رجیسترها (نه مستقیماً روی حافظه) انجام می‌شود. دستورالعمل پردازش داده‌ها در رجیسترها با دسترسی به حافظه متفاوت است.

مجموعه دستورالعمل ARM یکنواخت و طول آن ثابت است. پردازنده‌های آرم ۳۲ بیتی دو مجموعه دستورالعمل دارند: به طور کلی، مجموعه دستورالعمل آرم ۳۲ بیتی و مجموعه دستورالعمل Thumb با ۱۶ بیت است (Thumb)

شیوه‌ای برای فشرده‌سازی دستورات پرکاربرد ۳۲ بیتی به صورت ۱۶ بیتی جهت کاهش حجم برنامه، در ازای کاهش سرعت اجرای آن است .

پردازنده ARM از چندین مرحله خط لوله (Pipeline) برای سرعت بخشیدن به جریان دستورالعمل‌ها پشتیبانی می‌کند. در خط لوله سه مرحله‌ای ساده، دستورالعمل‌ها از سه مرحله پیروی می‌کنند: واکشی (fetch)، رمزگشایی (decode) و اجرا (execute).

برخی از ویژگی‌های عمومی ARM به شرح زیر هستند:

- پردازنده‌های ARM از سرعت مناسب نسبت به توان مصرفی برخوردار هستند.
- طیف فرکانس ساعت (کلاک) میکروپروسسورهای ARM گسترده و از ۱ مگاهرتز تا چند گیگاهرتز است.
- از اجرای مستقیم کدهای جاوا با استفاده از Java Jazelle DBX ARM پشتیبانی می‌کنند.
- پردازنده‌های ARM به صورت سخت‌افزاری برای اشکال‌زدایی (Debugging) ساخته شده‌اند.
- از دستورالعمل‌های پیشرفته برای عملیات پردازش سیگنال دیجیتال یا DSP بهره می‌برند.

## انواع پردازنده‌های ARM

پردازنده‌های ARM را می‌توان به پردازنده‌های کلاسیک (ARM Classic Processors)، پردازنده‌های نهفته (ARM Embedded Processors) و پردازنده‌های کاربردی (ARM Application Processors) تقسیم‌بندی کرد.

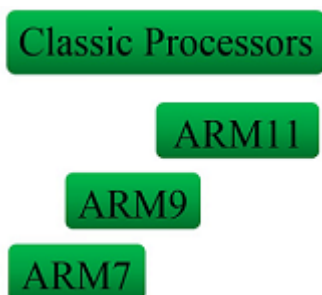
Classic Processors

Embedded Processors

Application Processors

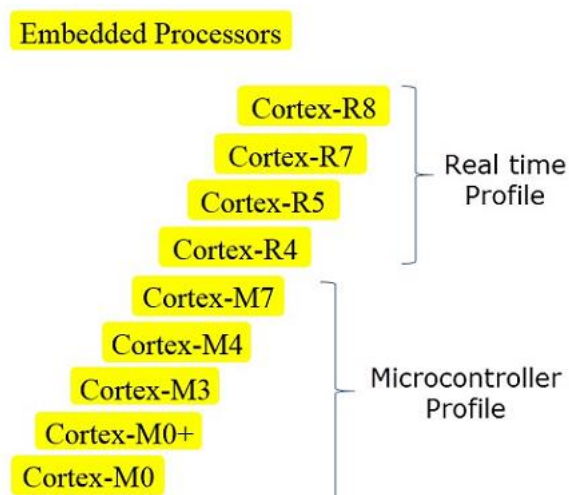


پردازنده‌های کلاسیک ARM شامل خانواده‌های ARM9، ARM7 و ARM11 هستند و ARM7TMDI هنوز پرستفاده‌ترین پردازنده ۳۲ بیتی است. پردازنده‌های مبتنی بر ARM7 هنوز هم در بسیاری از دستگاه‌های کوچک و ساده ۳۲ بیتی استفاده می‌شوند.

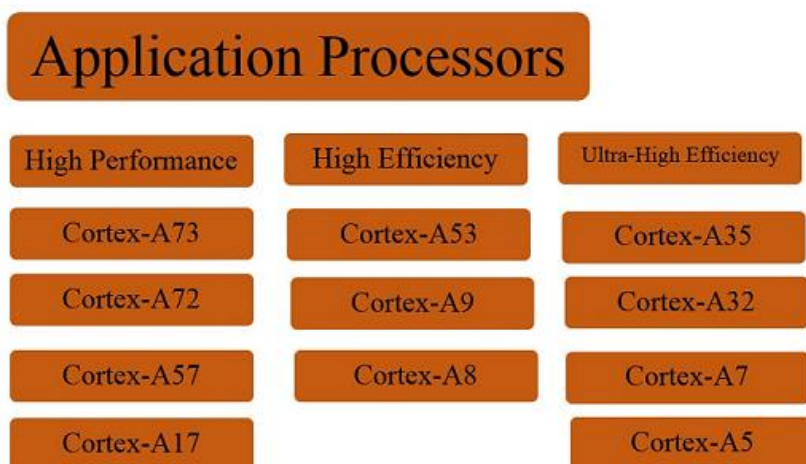


می‌توان از ARM7 یا سایر پردازنده‌های کلاسیک ARM برای سیستم‌های تعبیه شده در مقیاس کوچک استفاده کرد که با استفاده از پردازنده‌های پیشرفته تعبیه شده ARM یا پردازنده‌های Cortex-M و پردازنده‌های Cortex-R ساخته می‌شوند. پردازنده‌های Cortex-M دارای مشخصات میکروکنترلری هستند، در حالی که پردازنده‌های Cortex-R زمان واقعی‌اند.

پردازنده‌های Cortex-M دارای انرژی کارآمد بوده و برای اجرا ساده هستند و عمدتاً برای برنامه‌های پیشرفته تعبیه شده توسعه یافته‌اند. پردازنده‌های Cortex-M ARM به چندین دسته پردازنده مانند Cortex-M0، Cortex-M0+، Cortex-M3، Cortex-M4 و Cortex-M7 تقسیم می‌شوند.



سری Cortex-A بالاترین عملکرد را در بین پردازنده‌های ARM دارند. این پردازنده‌ها در دستگاه‌های تلفن همراه، محصولات فناوریانه مانند تجهیزات شبکه، لوازم برقی، سیستم‌های اتوماسیون، خودروها و سایر سیستم‌ها توکار به کار می‌روند.



پردازنده‌های Cortex-A خود به انواع پردازنده‌های عملکرد بالا (High Performance)، بازدهی بالا (High Efficiency) و بازدهی بسیار بالا (Ultra-high Efficiency) تقسیم می‌شوند. هریک از این‌ها نیز خود انواع هسته‌های متخلفی دارند.

## ۲- معماری x86

### معرفی و بررسی اجمالی

x86 به خانواده‌ای از مجموعه دستورالعمل‌ها که بر پایه پردازنده Intel 8086 است، اشاره دارد. ۸۰۸۶ در سال ۱۹۷۸ به عنوان پردازنده ۱۶ بیتی و توسعه‌داده شده‌ی پردازنده ۸ بیتی ۸۰۸۰، به کار گرفته شد. بعدها پردازنده‌هایی عرضه شدند که همگی آن‌ها مانند Intel 8086 به عدد ۸۶ ختم می‌شدند. مانند: ۸۰۲۸۶، ۸۰۳۸۶، ۸۰۴۸۶، ۸۰۵۸۶ و... (از اواسط سال ۱۹۸۰ تا ۱۹۹۰).

در طول سال‌ها، توسعه‌ها و موارد اضافی بسیاری به دستورالعمل x86 اضافه شده است که عمدتاً از خاصیت "سازگاری عقب‌رو (backward compatible)" پشتیبانی می‌کنند.

این معماری در پردازنده‌های اینتل، Cyrix، AMD، VIA Technologies و بسیاری از شرکت‌های دیگر پیاده‌سازی شده است. همچنین پیاده‌سازی‌های باز، مانند پلت فرم Zet SoC (در حال حاضر غیرفعال) نیز موجود است. با این وجود، از این تعداد، تنها Intel، AMD، VIA Technologies و P Electronics & DM دارای مجوزهای معماری x86 هستند و از این تعداد، فقط دو مورد اول به طور فعال طرح‌های مدرن ۶۴ بیتی را تولید می‌کنند.

از سال ۲۰۲۱، بیشتر رایانه‌های شخصی، لپ‌تاپ‌ها و کنسول‌های بازی فروخته شده، براساس معماری x86 ساخته می‌شوند، در حالی که دسته‌های تلفن همراه مانند تلفن‌های هوشمند یا تبلت‌ها تحت سلطه ARM هستند. امروزه، x86 همچنان بر محاسبات فشرده در ایستگاه‌های کاری و بخش‌های رایانش ابری تسلط دارد، در حالی که سریع‌ترین ابر رایانه مبتنی بر ARM است.

در دهه ۱۹۸۰ و اوایل دهه ۱۹۹۰، زمانی که ۸۰۸۸ و ۸۰۲۸۶ هنوز هم به‌صورت رایج، استفاده می‌شدند، اصطلاح x86 معمولاً هر CPU سازگار با ۸۰۸۶ را نشان می‌داد. با این حال امروزه، x86 معمولاً سازگاری دودویی را با مجموعه دستورات ۳۲ بیتی ۸۰۳۸۶ نیز در پی دارد. این امر، به دلیل این واقعیت است که این مجموعه دستورالعمل برای بسیاری از سیستم عامل‌های مدرن به کمترین مخارج مشترک تبدیل شده است و احتمالاً به این دلیل است که این اصطلاح، پس از معرفی ۸۰۳۸۶ در سال ۱۹۸۵ رایج شد.

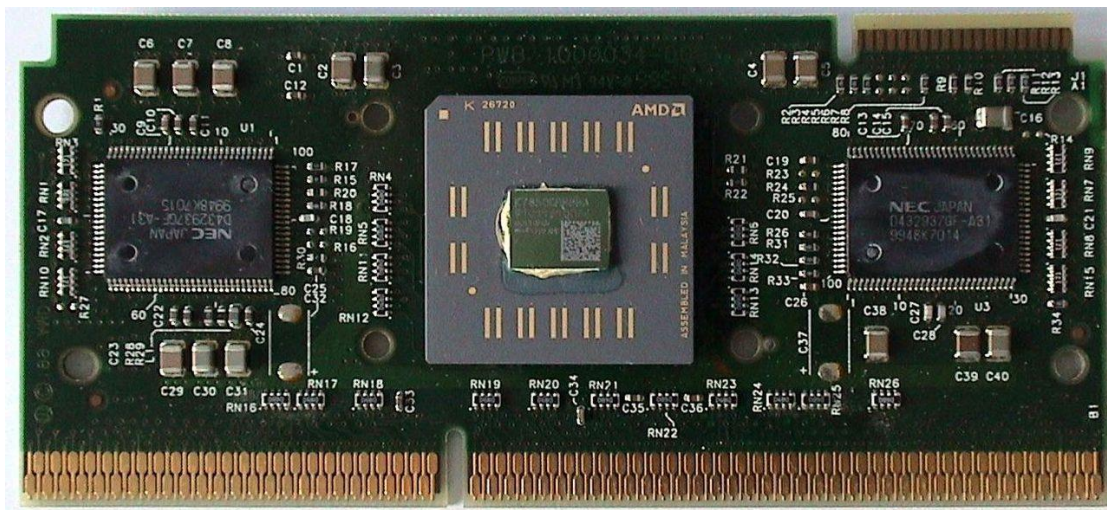
چند سال پس از معرفی ۸۰۸۶ و ۸۰۸۸، اینتل به طرح نامگذاری و اصطلاحات فنی خود، پیچیدگی‌هایی افزود زیرا iAPX، پردازنده بلندپرواز اما بد سرنوشت اینتل iAPX 432، که با تراشه‌های موفق تر خانواده ۸۰۸۶ امتحان شده بود، به عنوان نوعی پیشوند، در سطح سیستم اعمال شد. یک سیستم ۸۰۸۶، شامل پردازنده‌های بزرگ مانند ۸۰۸۷ و ۸۰۸۹، و تراشه‌های سیستمی ساده‌تر مختص Intel بوده که در نهایت به عنوان یک سیستم iAPX 86 توصیف شده است. همچنین اصطلاحاتی مانند iRMX (برای سیستم عامل‌ها)، iSBC (برای رایانه‌های تک برد) و iSBX (برای بردهای چند ماژولی و مبتنی بر معماری ۸۰۸۶) وجود داشت که همه با هم تحت عنوان Microsystem 80 شناخته می‌شدند. با این حال، این طرح نامگذاری کاملاً موقتی بود و در اوایل دهه ۱۹۸۰ چند سال به طول انجامید.

اگرچه ۸۰۸۶ در درجه اول برای سیستم‌های جاسازی شده و رایانه‌های کوچک چند کاربره یا تک کاربره ساخته شد، اما خط x86، عمدتاً به عنوان پاسخی به Zilog Z80 سازگار با ۸۰۸۰، به سرعت در ویژگی‌ها و قدرت پردازش رشد کرد. امروزه x86 در رایانه‌های شخصی ثابت و قابل حمل در همه جا وجود دارد و همچنین در رایانه‌های میان رده‌ای، ایستگاه‌های کاری، سرورها و بیشتر خوشه‌های ابر رایانه‌ای جدید موجود در لیست TOP500 استفاده می‌شود. تعداد زیادی از نرم افزارها، از جمله لیست بزرگی از سیستم عامل‌های x86، از سخت افزارهای مبتنی بر x86 استفاده می‌کنند.

x86 مدرن در سیستم‌های جاسازی شده نسبتاً غیرمعمول است، از طرفی، اپلیکیشن‌های کم مصرف (با استفاده از باتریهای ریز) و مارکت‌های ریزپردازنده‌های ارزان قیمت مانند لوازم خانگی و اسباب بازی‌ها، فاقد حضور قابل توجه x86 هستند. در این‌جا، معماری‌های ساده ۸ و ۱۶ بیتی معمول است، اگرچه VIA C7، VIA Nano، AMD's Geode، Athlon Neo و Intel Atom سازگار با x86، نمونه‌هایی از طرح‌های ۳۲ و ۶۴ بیتی هستند که در برخی از سناریوهای نسبتاً کم مصرف و کم هزینه مورد استفاده قرار گرفته‌اند.



Intel Core 2 Duo, an example of an x86-compatible, 64-bit multicore processor



AMD Athlon (early version), a technically different but fully compatible x86 implementation

## ویژگی‌های اساسی موجود در معماری

معماری x86، مجموعه‌ای از دستورالعمل‌ها با طول متغیر است که در درجه اول آن، معماری CISC با تأکید بر سازگاری عقب‌رو قرار دارد. مجموعه دستورالعمل‌ها معمولاً CISC نیستند، اما در اصل نسخه‌ی توسعه یافته‌ای از معماری‌های ساده‌ی هشت بیتی ۸۰۰۸ و ۸۰۸۰ هستند. در این حالت، آدرس دهی بایت امکان‌پذیر است و کلمات با ترتیب little-endian byte در حافظه ذخیره می‌شوند. دسترسی حافظه به آدرس‌های غیرهمتراز برای همه اندازه‌های معتبر کلمه، مجاز است. بزرگترین اندازه محلی برای محاسبات عدد صحیح و آدرس‌های حافظه (یا)، ۱۶، ۳۲ یا ۶۴ بیت است که به نسل معماری بستگی دارد (پردازنده‌های جدید شامل پشتیبانی مستقیم از عددهای صحیح کوچکتر نیز هستند). مقادیر متعدد اسکالر را می‌توان همزمان از طریق واحد SIMD موجود در نسل‌های بعدی، پشتیبانی کرد. انحرافات آدرس دهی فوری (Immediate addressing offsets) و داده‌های فوری (immediate data) ممکن است به عنوان مقادیر ۸ بیتی، برای مواردی که مکرراً در حال وقوع است و یا زمینه‌هایی که یک دامنه‌ی ۱۲۷..۱۲۸- برای آن‌ها کافی است، بیان شود. بنابراین دستورالعمل‌های معمولی، دارای طول ۲ یا ۳ بایت هستند (اگرچه بعضی از آنها بسیار طولانی‌تر و برخی دیگر تک بایت هستند).

برای حفظ بیشتر فضای رمزگذاری، بیشتر رجیسترها با استفاده از سه یا چهار بیت در آپ‌کدها بیان می‌شوند (دومی از طریق پیشوند کد در حالت ۶۴ بیتی)، در حالی که حداکثر یک عملوند برای یک دستورالعمل می‌تواند یک جایگاه حافظه باشد. با این حال، این عملوند حافظه ممکن است مقصد (یا یک منبع و مقصد ترکیبی) باشد، در حالی که عملوند دیگر، منبع، می‌تواند رجیستر (register) یا فوری (immediate) باشد. در میان عوامل دیگر، این امر به اندازه‌ی کد کمک می‌کند که با ماشین‌های هشت بیتی رقابت کند و استفاده‌ی کارآمد از حافظه نهان دستورالعمل را امکان‌پذیر می‌سازد. تعداد نسبتاً کمی از رجیسترهای عمومی (که از اعداد ۸ بیتی آن به ارث رسیده است)، آدرس دهی نسبی رجیستر (با استفاده از offset‌های فوری کوچک) را به روشی مهم برای دستیابی به عملوندها، به ویژه روی پشته (stack) تبدیل کرده است.

یک پردازنده‌ی نقطه‌شمار اختصاصی با رجیسترهای داخلی ۸۰ بیتی (یعنی ۸۰۸۷)، برای ۸۰۸۶ اصلی ساخته شده است. این ریزپردازنده متعاقباً به ۸۰۳۸۷ تأمین‌یافته، توسعه پیدا کرد و پردازنده‌های بعدی، یک نسخه با قابلیت "سازگاری عقب‌رو (backward compatible)" از این عملکرد را، در همان ریزپردازنده، مشابه پردازنده‌ی اصلی، ادغام کردند.

علاوه بر این، طراحی‌های مدرن x86 شامل یک واحد SIMD نیز است که در آن دستورالعمل‌ها می‌توانند به صورت موازی بر روی (یک یا دو) کلمه ۱۲۸ بیتی کار کنند، که هر کدام حاوی دو یا چهار عدد با نقطه‌شمار



(هرکدام به ترتیب به پهنای ۶۴ یا ۳۲) ، یا به جای آن ، ۲ ، ۴ ، ۸ یا ۱۶ عدد صحیح (به کدام به ترتیب به پهنای ۶۴ ، ۳۲ ، ۱۶ یا ۸ بیت) می باشد.

حضور رجیسترهای گسترده SIMD به این معنی است که پردازنده های x86 موجود ، می توانند حداکثر ۱۲۸ بیت از داده های حافظه را در یک دستورالعمل ، بارگیری یا ذخیره کنند و همچنین عملیات بیتی را روی مقادیر کامل ۱۲۸ بیتی به صورت موازی انجام دهند. پردازنده های Intel's Sandy Bridge با افزودن دستورالعمل های Advanced Vector Extensions (AVX) ، رجیسترهای SIMD تا ۲۵۶ بیت گسترش دادند. دستورالعمل های Intel Initial Many Core که توسط پردازنده های Knights Corner Xeon Phi پیاده سازی شده و دستورالعمل های AVX-512 که توسط پردازنده های Knights Landing Xeon Phi و Skylake-X اجرا شده اند ، از رجیسترهای عریض SIMD به پهنای ۵۱۲ بیت استفاده می کنند.

## رجیسترهای x86

### ۱۶-بیتی

اینتل اصلی ۸۰۸۶ و ۸۰۸۸ دارای چهارده رجیستر ۱۶ بیتی هستند. چهار مورد از آنها (AX, BX, CX, DX) رجیسترهای عمومی (GPR) هستند ، اگرچه هر یک ممکن است یک هدف اضافی نیز داشته باشند. به عنوان مثال ، فقط CX می تواند به عنوان شمارنده با دستورالعمل حلقه (loop) مورد استفاده قرار گیرد. هریک توسط دو بایت جداگانه قابل دسترسی پیدا کرد ( بنابراین می توان به بایت بالای BX به صورت BH و بایت پایین آن، به صورت BL دسترسی داشت).

دو رجیستر اشاره گر نقش ویژه ای دارند: SP (اشاره گر پشته) به "بالای" پشته اشاره می کند و BP (اشاره گر پایه) اغلب برای اشاره کردن به مکان دیگری از پشته ، به طور معمول بالای متغیرهای محلی، استفاده می شود. رجیسترهای SI ، DI ، BX و BP ، رجیسترهای آدرس هستند و ممکن است برای اندیس گذاری آرایه ای نیز مورد استفاده قرار گیرند.

چهار رجیستر (CS, DS, SS و ES) برای تشکیل یک آدرس حافظه استفاده می شوند. رجیستر FLAGS شامل پرچم هایی مانند پرچم حمل (carry flag) ، پرچم سرریز (overflow flag) و پرچم صفر (zero flag) است. در نهایت ، اشاره گر دستورالعمل (IP) ، به دستورالعمل بعدی اشاره می کند که از حافظه واکشی شده و سپس اجرا می شود. این رجیستر نمی تواند ، به طور مستقیم، توسط یک برنامه قابل دسترسی (خواندن یا نوشتن) باشد.

اینتل ۸۰۱۸۶ و ۸۰۱۸۸ اساساً یک ورژن به‌روزشده از CPU های ۸۰۸۶ و ۸۰۸۸ (با تجهیزات جانبی اضافه شده روی تراشه) هستند و همان CPU registers های مشابه ۸۰۸۶ و ۸۰۸۸ را دارند (علاوه بر رجیسترهای رابط برای تجهیزات جانبی).

۸۰۸۶، ۸۰۸۸، ۸۰۱۸۶ و ۸۰۱۸۸ می‌توانند از یک کمک پردازنده‌ی نقطه‌شناور اختیاری، تحت عنوان ۸۰۸۷، استفاده کنند. ۸۰۸۷ به عنوان بخشی از CPU برای برنامه‌نویس به نظر می‌رسد و هشت رجیستر ۸۰ بیتی، یعنی st (0) تا st (7)، را اضافه می‌کند که هر کدام از آنها می‌توانند داده‌های عددی را در یکی از این هفت قالب نگه دارند: نقطه‌شناور با ۳۲، ۶۴ یا ۸۰ بیت، عدد صحیح باینری با ۱۶، ۳۲ یا ۶۴ بیت و عدد صحیح دهدهی بسته بندی شده با ۸۰ بیت.

در Intel 80286، برای پشتیبانی از حالت محافظت شده، سه رجیستر خاص آدرس جدول توصیف کننده را نگه می‌دارند (GDTR، LDTR، IDTR) و رجیستر چهارمی تحت عنوان TR (task register) برای تغییر task استفاده می‌شود. ۸۰۲۸۷ یک پردازنده کمکی با نقطه‌ی شناور برای ۸۰۲۸۶ است و همان رجیسترهای موجود در ۸۰۸۷ با همان قالب‌های داده را دارد.

## ۳۲-بیتی

با ظهور پردازنده ۳۲ بیتی ۸۰۳۸۶، رجیسترهای عمومی ۱۶ بیتی، رجیسترهای پایه، رجیسترهای اندیس، اشاره‌گر دستورالعمل و رجیستر FLAGS، به جز رجیسترهای segment، به ۳۲ بیت گسترش یافتند و نامگذاری، این موضوع را با اضافه کردن یک پیشوند E (گرفته‌شده از ابتدای "extended")، به نام‌های رجیسترهای موجود در زبان اسمبلی x86 نشان داد. بنابراین، رجیستر AX مربوط به ۱۶ بیت کم‌ارزش‌تر رجیستر ۳۲ بیتی EAX جدید، SI مربوط به ۱۶ بیت کم‌ارزش ESI و... است. رجیسترهای عمومی، رجیسترهای پایه و رجیسترهای اندیس می‌توانند به عنوان پایه در حالت‌های آدرس‌دهی مورد استفاده قرار گیرند و همه این رجیسترها، به جز نشانگر پشته، می‌توانند به عنوان شاخص در حالت‌های آدرس‌دهی به کار گرفته شوند.

در ادامه، دو رجیستر segment جدید تحت عنوان FS و GS اضافه شدند و با استفاده از تعداد بیشتری از رجیسترها، دستورالعمل‌ها و عملوندها، قالب کد ماشین توسعه پیدا کرد. برای ایجاد "سازگاری به عقب"، بخش‌هایی با کد اجرایی را می‌توان علامت‌گذاری کرد که حاوی دستورالعمل‌های ۱۶ بیتی یا ۳۲ بیتی است. پیشوندهای ویژه اجازه می‌دهند که دستورالعمل‌های ۳۲ بیتی را در یک بخش ۱۶ بیتی قرار دهید یا بالعکس.

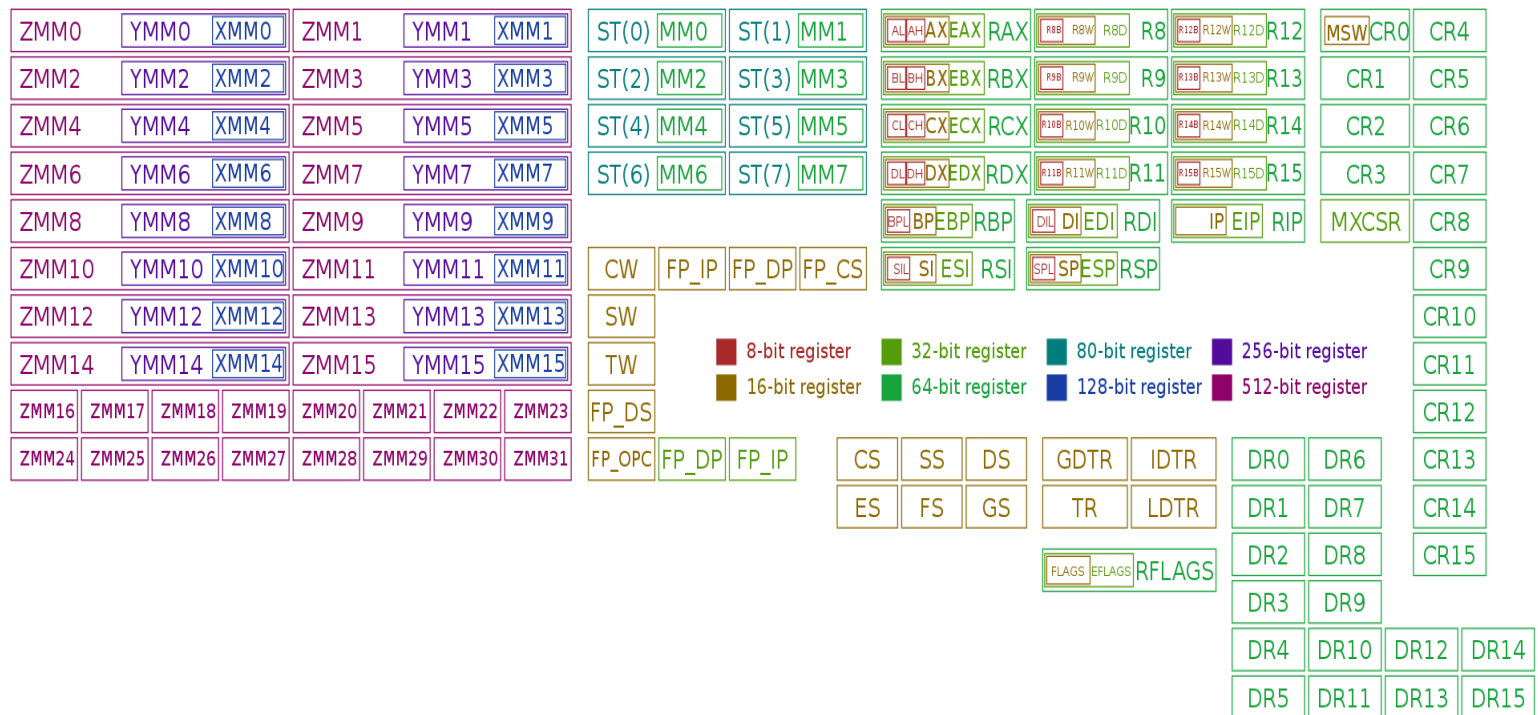


۸۰۳۸۶ دارای یک کمک پردازنده‌ی نقطه شناور اختیاری ، تحت عنوان ۸۰۳۸۷ بود که مانند ۸۰۸۷ و ۸۰۲۸۷، هشت رجیستر گسترده ۸۰ بیتی داشت: st (0) تا st (7). ۸۰۳۸۶ همچنین می‌توانست از یک کمک پردازنده‌ی ۸۰۲۸۷ استفاده کند. با استفاده از ۸۰۴۸۶ و تمام مدل های بعدی x86 ، واحد پردازش نقطه شناور (FPU)، روی تراشه مجتمع شده است.

## ۶۴-بیتی

با شروع به کار پردازنده AMD Opteron ، معماری x86 ، رجیسترهای ۳۲ بیتی را به رجیسترهای ۶۴ بیتی گسترش داد و این عمل با روشی مشابه نحوه توسعه رجیسترهای ۱۶ به ۳۲ بیتی ، صورت پذیرفت. یک پیشوند R ( گرفته شده از ابتدای "register" ) ، رجیسترهای ۶۴ بیتی ( RAX ، RBX ، RCX ، RDX ، RSI ، RDI ، RBP ، RSP ، RIP ، RFLAGS ) و هشت رجیستر عمومی ۶۴ بیتی اضافی ( R8-R15 ) را مشخص می‌کند که در ایجاد x86-64 معرفی شده‌اند. با این حال ، این توسعهها فقط در حالت ۶۴ بیتی قابل استفاده هستند که درواقع یکی از دو حالتی است که تنها در long mode قابل دسترسی می‌باشند. حالت‌های آدرس‌دهی به طرز چشمگیری از حالت ۳۲ بیتی تغییر نکردند ، با این تفاوت که آدرس دهی به ۶۴ بیت افزایش یافت. در حال حاضر، آدرس‌های مجازی به ۶۴ بیت (جهت جلوگیری از بیت‌های حالت در آدرس‌های مجازی) گسترش یافته‌اند و سایر جزئیات انتخابگر به طرز چشمگیری کاهش یافته است .

علاوه بر این ، یک حالت آدرس‌دهی اضافه شد تا به منابع حافظه ، نسبت به RIP ( اشاره گر دستورالعمل ) ، اجازه دهد که اجرای کد مستقل از موقعیت را ، که در کتابخانه های مشترک در برخی سیستم عامل‌ها استفاده می‌شود، سهولت بخشد.



Registers available in the x86-64 instruction set

## RISC در مقابل CISC یا ARM در مقابل x86

RISC که مخفف Reduced instruction Set Computing یا مجموعه دستورات ساده شده است در واقع نوعی از طراحی CPU است که پایه و اساس آن، ساده سازی دستورات است که منجر به بازده بالا و سرعت بخشیدن به اجرای دستورات می شود. پردازنده ای که براساس این طراحی ساخته می شود را RISC (ریسک) می نامند. مهمترین و معروفترین معماری که براساس RISC طراحی شده، ARM است. درست نقطه مقابل ریسک، طراحی دیگری با نام CISC وجود دارد که مخفف Complex Instruction Set Computing یا مجموعه دستورات پیچیده است که معماری x86 اینتل براساس آن طراحی شده و پردازنده کامپیوترهای رومیزی و لپ تاپ ها و بسیاری از ابزارهای دیگر از آن بهره می برند.

ایده اصلی RISC اولین بار توسط جان کوکی از IBM و در سال ۱۹۷۴ شکل گرفت، نظریه او به این موضوع اشاره داشت که یک کامپیوتر تنها از ۲۰ درصد از دستورات نیاز دارد و ۸۰ درصد دیگر، دستورات غیرضروری هستند. پردازنده‌های ساخته شده براساس این طراحی از دستورات کمی پشتیبانی می‌کنند به این ترتیب به ترانزیستور کمتری نیز نیاز دارند و ساخت آنها نیز کم هزینه است. با کاهش تعداد ترانزیستورها و اجرای دستورات کمتر، پردازنده در زمان کمتری دستورات را پردازش می‌کند. کمی بعد اصطلاح RISC توسط یک استاد دانشگاه کالیفورنیا به نام دیوید پترسون ایجاد شد.

هر دو طراحی RISC و CISC به مراتب در انواع و اقسام ابزارها بکار گرفته می‌شوند، اما مفهوم کلی RISC در واقع سیستمی است که در آن به پردازش دستورات کوچک و به شدت بهینه شده پرداخته می‌شود، درست برخلاف CISC که در آن دستورات پیچیده ارسال می‌شوند. یکی از تفاوت‌های عمده بین RISC و CISC نیز در نحوه دسترسی به حافظه و ذخیره و اجرای اطلاعات بر روی آن است. در ریسک دسترسی به حافظه تنها از طریق دستورالعمل‌های خاصی قابل انجام است و به عنوان مثال نمی‌توان از بخشی از دستور add به حافظه دسترسی داشت.

علاوه بر ARM شرکت‌های بسیار دیگری از جمله Intel i860, AMD 29k , ARC و غیره از طراحی RISC برای ساخت پردازنده استفاده می‌کنند، اما به لطف گسترش تلفن و تبلت‌ها، معماری ARM به عنوان برجسته‌ترین معماری مبتنی بر RISC شناخته می‌شود.

در سیسک (CISC) اوضاع دقیقاً برعکس ریسک است و پردازنده قادر به پردازش دستورات پیچیده است به همین دلیل نیاز به تعداد بیشتر ترانزیستور و همچنین طراحی پیچیده‌تر و پردازنده‌های گران قیمت‌تر دارد. ایده اصلی پشت این طراحی این است که برنامه نویسان ساده‌تر بتوانند نرم افزارهای خود را تولید کنند و دستورات را ساده‌تر به CPU ارجاع دهند. به لطف پشتیبانی اینتل و تولیدکنندگان نرم افزار، CISC به شدت محبوب شد و تمام کامپیوترها از پردازنده مبتنی بر این طراحی بهره بردند.

برخی تصور می‌کنند که ریسک قادر به اجرای دستورات زیاد نیست اما در حقیقت ریسک به اندازه سیسک می‌تواند دستورات مختلف را اجرا کند اما مهمترین تفاوت این دو در این است که در RISC تمام دستورات با یک فرمت، دقیقاً یک فرمت صادر می‌شوند و پردازش تمام دستورات یک زمان مشخص طول می‌کشد، معمولاً در ریسک در هر سیکل، پردازنده یک دستور را اجرا می‌کند.

اما در CISC مجموعه‌ای از دستورات بصورت فشرده و با آدرس دهی مختلف به یکباره پردازش می‌شوند، مثل اعداد اعشاری یا تقسیم که در طراحی RISC وجود ندارند. از آنجایی که دستورات در RISC ساده‌تر هستند پس سریعتر اجرا می‌شوند و نیاز به ترانزیستور کمتری دارند، ترانزیستور کمتر هم به معنی دمای کمتر، مصرف پایین‌تر و فضای کمتر است که آن را برای ابزارهای موبایل مناسب می‌کند.

معماری پردازنده‌های مبتنی بر طراحی RISC طی سال‌های گذشته پیشرفت چشم‌گیری داشته و اجرای دستورات پیچیده را نیز میسر کرده است و تولیدکنندگان نرم‌افزاری نیز به سمت ساخت نرم‌افزارهای مبتنی بر این معماری گرایش پیدا کرده‌اند. لازم است بدانید که کامپیوترهای اولیه مک نیز از پردازنده مبتنی بر RISC بهره می‌بردند.

اما در واقع پردازنده‌های CISC بسیار سریعتر و پرقدرت‌تر از RISC ها هستند و قادر به پردازش امور سنگین می‌باشند اما در عوض گران‌قیمت‌تر، پرمصرف‌تر بوده و دمای بیشتری نیز تولید می‌کنند. در CISC تمرکز بر روی سخت‌افزار است و در RISC بر روی نرم‌افزار، در CISC دستورات بصورت پیچیده به پردازنده ارسال می‌شوند ولی در RISC نرم‌افزار دستورات را ساده کرده و به عنوان مثال یک عملیات پیچیده را در قالب چندین دستور ساده به پردازنده ارسال می‌کند و پردازنده دستورات ساده را به سرعت پردازش نموده و نتیجه را باز می‌گرداند. پس کدهای نرم‌افزارهای سازگار با RISC طولانی‌تر ولی کدهای مربوط به نرم‌افزارهای CISC کوتاه‌تر و پیچیده‌تر هستند. البته این بدین معنا نیست که مثلاً اگر قرار است برای اندروید یا iOS برنامه بنویسید باید چند هزار خط بیشتر از معادل کامپیوتر ویندوزی آن کد نویسی کنید، در واقع کامپایلرها کدها را به دستورات کوچک زیاد تبدیل می‌کنند و برنامه نویس به سختی متوجه نوع پردازش دستورات می‌شود.

لازم به ذکر است که هسته اصلی CPU در ARM نیاز به ۳۵ هزار ترانزیستور دارد اما در پردازنده‌های X86، نیاز به میلیون‌ها ترانزیستور در بخش هسته سی‌پی‌یو هست و همین امر نیز منجر به بالا بودن سرعت و قیمت آن می‌شود. در حال حاضر استفاده از پردازنده ARM در گوشی‌های تلفن همراه و تبلت‌ها رواج بیشتری یافته است و یکی از علل آن مصرف پایین انرژی می‌باشد. البته باید این موضوع را در نظر داشت که ARM و X86 هر دو یک هدف را دنبال می‌کنند.

ARM و x86 در اصل هدف یکسانی را دنبال می‌کنند: خلق یک «معماری مجموعه دستورات» قدرتمند و انعطاف پذیر تا بتواند در کامپیوترهای شخصی مدرن مورد استفاده قرار گیرد. البته این دو تکنولوژی در ابتدا کاملاً انشعاب یافتند و هنگامی که x86 در دنیای کامپیوترهای شخصی برنده شد، ARM در این فضا به گوشه‌ای رانده شد.

این می توانست پایان کار ARM و نابودی این خانواده باشد. اما در عوض، ARM نیروهای خود را جمع کرده و با قدرت در فضایی شروع به رشد و گسترش نموده که اینتل با x86 چندان تلاشی برای حضور در آن نکرده بود: سیستم های جاسازی شده.

اگر چه اینتل قابلیت های x86 را گسترش می داد و به توسعه پردازنده هایی روی می آورد که به شکل روز افزونی قدرتمند بودند، اما این فرآیند تکامل در یک دوره زمانی توجه چندان به راندمان مصرف انرژی و بهینه سازی آن نداشت. علاوه بر این، اینتل در آن زمان و حتی اکنون به شکل خارق العاده ای از حق امتیاز و لیسانس خود محافظت می کند. و این یعنی توسعه تکنولوژی شدیداً محدود به جهتی است که اینتل می خواهد برود. و تا چندی پیش حضور x86 در قلب سیستم های جاسازی شده یا ابزارهای همراه برای این شرکت محلی از اعراب نداشت.

و به همین دلیل، ARM امروزه «معماری مجموعه دستورالعمل» ساده تری دارد. برای یک پردازنده با معماری بر اساس ARM بسیار ساده است با سرعت یک گیگاهرتز، مصرف انرژی خود را به کمتر از ۲ وات TDP برساند. اما پردازنده های اتم اینتل با بهترین راندمان هنوز برای رسیدن به اعداد کمتر از ۵ وات TDP با مشکل روبرو هستند. شاید این موضوع چندان مهمی به نظر نرسد، اما هنگامی شما درباره ابزارهایی به کوچکی یک تلفن هوشمند صحبت می کنید، تاثیر تفاوت نیروی مصرفی با توجه به کمبود فضا چندین برابر می شود. طراحی هایی که نیروی بیشتری مصرف کنند باید این مشکل را با باتری بزرگ تر حل کنند و این دردسر بزرگی در شکل کلی ابزار کار و اندازه آن است. در صورتی که این مشکل در ابزارهای بزرگتر، همچون لپ تاپ ها، کمتر به چشم می آید.

به هر حال، ریزپردازنده های x86 اینتل هنوز به شکل واضحی، رهبر بلامنازع داستان کارایی هستند. پشتیبانی از دستورالعمل های پیچیده بیشتر توسط x86 ممکن است پردازنده های آنها را بزرگتر و با راندمان انرژی کمتر نشان دهد؛ اما نتیجه کار، CUP های بسیار سریع تر است. همچنین x86 اکنون به شکل موثری از «معماری مجموعه دستورالعمل» ۶۴ بیتی هم بهره می برد. در این میان باید ممنون AMD باشیم، زیرا پیشگام داستان ۶۴ بیتی ها بود. اما ARM هنوز در این زمینه هیچ گامی بر نداشته و تنها ۳۲ بیتی است (البته ۱۶ بیتی آن هم در برخی ابزارهای سطح پایین دیده می شود). تاثیر این تفاوت ها را به خوبی می توان در سیستم عامل های مورد پشتیبانی هر یک از «معماری مجموعه دستورالعمل» ها دید ARM: پایگاه اندروید، آی او اس، گوگل کروم و مانند آن است. در حالی که x86 با ویندوز، مک، لینوکس و... دم خور است.

## ۳- معرفی پردازنده MIPS

میپس به انگلیسی MIPS ، مخفف Microprocessor without Interlocked Pipelined Stages ، یک معماری مجموعه دستورالعمل (ISA) برای کامپیوترهای کم دستور (RISC) است که توسط MIPS Technologies ( MIPS Computer Systems سابق ) توسعه یافته است. معماری اولیه میپس ۳۲ بیتی بود، و نسخه های ۶۴ بیتی بعداً به آن اضافه شد. نسخه های متعددی از میپس وجود دارد از جمله: میپس I, II, III, IV, V ؛ و همچنین پنج توزیع میپس ۳۲/۶۴ (به ترتیب برای پیاده سازی ۳۲ و ۶۴ بیتی). از آوریل ۲۰۱۷، آخرین نسخه میپس ۳۲/۶۴ توزیع ۶ است. تفاوت اساسی بین میپس ۳۲/۶۴ و میپس V-I ، تعریف پردازنده کمکی برای کنترل سیستم در حالت کرنل دارای حق ویژه ، علاوه بر معماری حالت کاربر است.

چندین افزونه اختیاری نیز موجود است، از جمله میپس سه بعدی که مجموعه ای ساده از دستورات ممیز شناور چند داده است که به وظایف سه بعدی معمول اختصاص دارد MDMX یا MadMaX که یک مجموعه ای جامع تر چند داده برای اعداد صحیح است و از رجیسترهای ممیز شناور ۶۴ استفاده می کند، MIPS16e است که فشرده سازی را به جریان دستورها اضافه می کند تا برنامه ها فضای کمتری را اشغال کنند و میپس MT ، که قابلیت چندریسمانی را اضافه می کند .

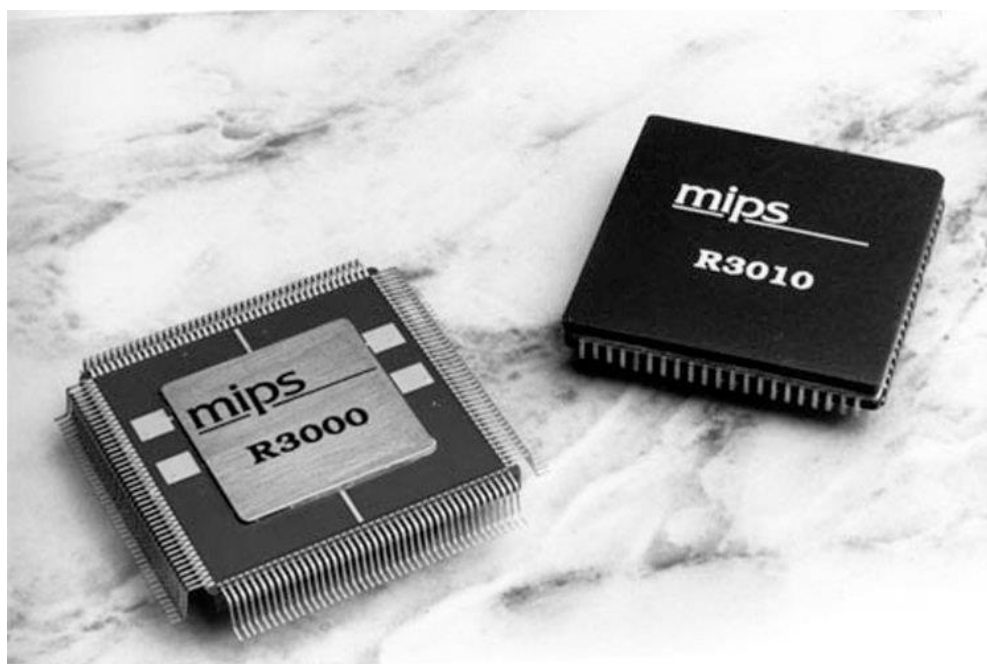
اغلب دوره های معماری کامپیوتر در دانشگاه ها و مدارس فنی، معماری میپس را مطالعه می کنند ؛ معماری های ریسک جدیدتر مانند آلفا تا حد زیادی از این معماری تأثیر پذیرفته اند.

از آوریل ۲۰۱۷، پردازنده های میپس در سامانه های نهفته مانند دروازه های محلی و روترها استفاده می شود. در آغاز، میپس برای محاسبات عمومی طراحی شده بود. در دهه های ۱۹۸۰ و ۱۹۹۰، شرکت های بسیاری مانند SiCortex ، Pyramid Technology ، NEC ، MIPS Computer Systems ، Digital Equipment Corporation ، Siemens Nixdorf ، Silicon Graphics و Tandem Computers از پردازنده های میپس برای کامپیوترهای شخصی، ایستگاه های کار و کامپیوترهای سرور استفاده می کردند. در طول تاریخ، کنسول های بازی مانند نینتندو ۶۴، پلی استیشن، پلی استیشن ۲ و پلی استیشن همراه از پردازنده های میپس استفاده می کردند. همچنین در دهه ۱۹۹۰ پردازنده های میپس محبوبیت زیادی در ابر رایانه ها داشتند، اما تمام این سیستم ها از فهرست TOP500 (پانصد ابرایانه برتر) حذف شده اند. این استفاده ها در ابتدا توسط کاربردهای جاسازی شده جا افتاد، اما در دهه ۱۹۹۰، میپس بخش بزرگی از بازار پردازنده های جاسازی شده را به خود اختصاص داده بود و در دهه ی

۲۰۰۰ اکثر پردازنده‌های میپس در این زمینه به کار می‌رفتند. در اواسط تا اواخر دهه ۱۹۹۰ تخمین زده شد که یک در هر سه ریزپردازنده ریسک تولید شده، یک پردازنده میپس بوده است.

میپس یک معماری مدولار است که تا چهار پردازنده کمکی را پشتیبانی می‌کند (CP 0/1/2/3). در اصطلاح میپس، CP0 پردازنده‌ی کمکی کنترل سیستم است (بخشی حیاتی از پردازنده که در میپس V-I در پیاده‌سازی تعریف شده است)، CP1 یک واحد اختیاری ممیز شناور (FPU) است و CP 2/3 پردازنده‌های کمکی اختیاری اند که در پیاده‌سازی تعریف می‌شوند. (میپس III، CP3 را حذف کرد و آپ کد آن را برای امور دیگر استفاده کرد). به عنوان مثال، در کنسول بازی پلی استیشن، CP2 موتور تبدیل هندسی (GTE) است که پردازش هندسی را در گرافیک کامپیوتری سه بعدی تسهیل می‌کند.

در دسامبر ۲۰۱۸، Wave Computing، صاحب جدید معماری میپس، اعلام کرد که میپس ISA در برنامه‌ای که MIPS Open initiative نامیده می‌شود، متن باز می‌شود. هدف برنامه‌ای که برای سال ۲۰۱۹ برنامه‌ریزی شده بود، این بود که دسترسی آزاد به اکثر نسخه‌های اخیر طراحی‌های ۳۲ بیتی و ۶۴ بیتی را بدون نیاز به هیچ گونه مجوز یا هزینه حق امتیاز فراهم کند و همچنین مجوز استفاده از اختراعات ثبت شده‌ی میپس موجود را در اختیار شرکت کنندگان قرار دهد.



## MIPS I

اولین نسخه معماری میپس توسط MIPS Computer Systems برای ریزپردازنده R2000 ، اولین پیاده‌سازی میپس، طراحی شد. میپس و R2000 با هم در سال ۱۹۸۵ معرفی شدند . هنگامی که میپس II معرفی شد، میپس به میپس I تغییر نام داد تا از نسخه جدید متمایز شود.

MIPS یک معماری بارگذاری / ذخیره‌سازی (نام دیگر آن رجیستر رجیستر) است؛ به جز دستورهای بارگذاری / ذخیره‌سازی استفاده شده برای دسترسی به حافظه، تمام دستورها در رجیسترها عمل می‌کنند.

### رجیسترها

میپس I دارای سی و دو رجیستر ۳۲ بیتی عمومی است. رجیستر ۰ \$ سخت‌افزاری صفر است و هر آنچه در آن نوشته شود، دور انداخته می‌شود. رجیستر ۳۱ \$ رجیستر پیوند است. برای دستورهای ضرب و تقسیم اعداد صحیح، که مستقل از دستور دیگر اجرا می‌شود، یک جفت رجیستر ۳۲ بیتی، HI و LO ارائه شده‌است. مجموعه‌ی کوچکی از دستورها برای کپی کردن داده‌ها بین رجیسترهای کلی و رجیسترهای HI / LO وجود دارد. شمارنده برنامه ۳۲ بیت دارد. از آنجا که طول دستورهای میپس I ، ۳۲ بیت است و هم اندازه‌ی حد طبیعی یک کلمه است، دو بیت مرتبه پایین‌تر همواره صفر است.

### قالب‌های دستورات

دستورها به سه نوع تقسیم می‌شوند: R ، I و J . هر دستور با یک آپ کد ۶ بیتی شروع می‌شود. علاوه بر آپ کد، دستورهای نوع R ، سه رجیستر، یک فیلد برای اندازه‌ی شیفت و یک فیلد برای تابع را مشخص می‌کنند. دستورهای نوع I ، دو رجیستر و یک مقدار فوری ۱۶ بیتی را مشخص می‌کند؛ دستورهای نوع J از آپ کد همراه با مقصد پرش ۲۶ بیتی پیروی می‌کنند.

در زیر سه قالب مورد استفاده برای مجموعه دستورهای اصلی آمده‌است:



Type	format (bits)					
R	opcode (6)	rs (5)	rt (5)	rd (5)	shamt (5)	funct (6)
I	opcode (6)	rs (5)	rt (5)	immediate (16)		
J	opcode (6)	address (26)				

## دستورهای CPU

### بارگذاری‌ها و ذخیره‌سازی‌ها

میپس I دستورهای برای بارگذاری و ذخیره بایت‌های ۸ بیتی، نیم کلمه‌های ۱۶ بیتی و کلمات ۳۲ بیتی دارد. تنها یک حالت آدرس دهی پشتیبانی می‌شود: پایه + جابجایی. از آنجا که میپس I یک معماری ۳۲ بیتی است، برای بارگذاری مقادیر کمتر از ۳۲ بیت داده باید آن‌ها را یا با علامت یا با صفر به ۳۲ بیت رساند. دستورهای بارگذاری با پیشوند "unsigned" داده‌ها را با صفر توسعه می‌دهند و در غیر اینصورت با علامت. منبع دستورهای بارگذاری برای پایه، محتوای یک رجیستر عمومی (rs) است و نتیجه را در یک رجیستر عمومی دیگر (rt) می‌نویسند. منبع دستورهای ذخیره‌سازی نیز برای پایه، محتوای یک رجیستر عمومی (rs) و برای داده‌های ذخیره شونده یک رجیستر عمومی دیگر (rt) است. تمام دستورهای بارگذاری و ذخیره، برای محاسبه آدرس، مقدار پایه را با مقدار فوری ۱۶ بیتی که با علامت به ۳۲ بیت رسانده شده جمع می‌کنند. در میپس I همه‌ی دسترسی‌های حافظه باید بر مرزهای طبیعی کلمه‌شان منطبق باشند، در غیراین صورت یک استثنا پرتاب (signal) می‌شود. برای پشتیبانی از دسترسی‌های بهینه‌ی غیر منطبق حافظه، دستورهای بارگذاری/ذخیره‌سازی کلمه با پیشوندهای «left» یا «right» وجود دارد. تمام دستورهای بارگذاری با یک شکاف تأخیر بارگذاری دنبال می‌شود. دستورهایی که در شکاف تأخیر بارگذاری می‌آیند نمی‌توانند از داده‌های بارگذاری شده توسط دستور بارگذاری

استفاده کنند. شکاف تأخیر بارگذاری را می توان با دستورهای که به بارگذاری بستگی ندارند پر کرد. اگر چنین دستورهایی یافت نشد یک دستور بی عمل جایگزین آن می شود.

## واحد محاسبه و منطق

میپس I دستورهایی برای جمع و تفریق دارد. این دستورها عمل وندهای خود را از دو رجیستر عمومی (rs و rt) می گیرند، و نتیجه را در یک رجیستر عمومی سوم (rd) ذخیره می کنند. علاوه بر این، یکی از عملوندهای جمع می تواند یک مقدار فوری ۱۶ بیتی باشد (که با علامت به ۳۲ بیت گسترش یافته است). دستورهای جمع و تفریق دو نوع اند: به طور پیش فرض، اگر نتیجه سرریز شود، یک استثنا پرتاب (signal) می شود؛ دستورهایی که پسوند "unsigned" دارند استثنا پرتاب نمی کنند. کنترل سرریز نتیجه را به صورت یک عدد صحیح متمم دو ۳۲ بیتی تفسیر می کند.

میپس I دستورهای برای انجام اعمال منطقی AND, OR, XOR و NOR بر روی بیتها دارد. این دستورها، عملوندهای خود را از دو رجیستر عمومی می گیرند و نتیجه را در رجیستر سوم ذخیره می کنند. همچنین یکی از عملوندهای دستورهای AND, OR و XOR می تواند یک مقدار فوری ۱۶ بیتی (که با صفر تا ۳۲ بیت توسعه یافته است) باشد.

مجموعه دستورهای رابطه ای، بسته به برقرار بودن یا نبودن رابطه در رجیستر مقصد، مقدار صفر یا یک را ذخیره می کند. عملوندهای این دستورها دو رجیستر عمومی یا یک رجیستر و یک مقدار فوری ۱۶ بیتی (که با علامت تا ۳۲ بیت توسعه یافته است) هستند، و نتیجه نیز در یک رجیستر عمومی سوم ذخیره می شود. به طور پیش فرض عملوندها به عنوان اعداد صحیح علامت دار تفسیر می شوند. فرمهایی از این دستورها که پسوند "unsigned" دارند عملوندها را اعداد صحیح بدون علامت تفسیر می کنند (حتی آنهایی که از توسعه ی با علامت یک مقدار فوری ۱۶ بیتی حاصل شده اند).

دستور بارگذاری بالایی مقدار فوری، مقدار فوری ۱۶ بیتی را در ۱۶ بیت مرتبه بالاتر یک رجیستر عمومی کپی می کند. این دستور همراه با دستور Or استفاده می شود تا یک مقدار فوری ۳۲ بیتی را در یک رجیستر بارگذاری کند.

## شیفت‌ها

میپس I دستورهای برای شیفت‌های چپ و راست منطقی و راست جبری دارد. عملگر از یک رجیستر عمومی (rt) گرفته می‌شود، و نتیجه نیز در یک رجیستر عمومی دیگر (rd) ذخیره می‌شود. مقدار شیفت یا از یک رجیستر عمومی (rs) گرفته می‌شود یا یک "مقدار شیفت" ۵ بیتی (فیلد "sa").

## ضرب و تقسیم

میپس I دستورهایی برای ضرب و تقسیم اعداد صحیح با علامت و بدون علامت دارد. عملوندهای این دستورها، دو رجیستر عمومی اند و حاصل را در یک جفت رجیستر ۳۲ بیتی به نام‌های HI و LO ذخیره می‌کنند، آن‌ها ممکن است جدا از (و همزمان با) دستورهای دیگر CPU اجرا شوند. در ضرب، نیمه‌های مرتبه بالا و مرتبه پایین حاصل ضرب ۶۴ بیتی به ترتیب در HI و LO ذخیره می‌شوند. در تقسیم، خارج قسمت در LO و باقی‌مانده در HI نوشته می‌شوند. برای دسترسی به نتایج دو دستور (انتقال از HI و انتقال از LO) فراهم شده‌اند که محتویات HI و LO را در یک رجیستر عمومی کپی می‌کنند. این دستورها به هم پیوسته‌اند. خواندن HI و LO قبل از اتمام عمل جبری ای که در HI و LO نوشته خواهد شد، میسر نیست. دو دستور دیگر (انتقال به HI و انتقال به LO)، محتویات یک رجیستر عمومی را در HI و LO کپی می‌کند. کاربرد این دستورها در رسیدگی به استثناهاست. با کمک آن‌ها مقادیر قبلی HI و LO را پس از رسیدگی به استثنا بازیابی می‌کنیم. بین دستورهایی که HI و LO را می‌خوانند باید دو دستور دیگر بیاید که در HI و LO نمی‌نویسند.

## پرش و انشعاب

پس از تمام دستورهای کنترل جریان میپس I، یک شکاف تأخیر انشعاب می‌آید. اگر این شکاف با دستوری که کار مفیدی انجام دهد پر نشده باشد، یک دستور بی‌عمل در آن قرار می‌گیرد. دستورهای انشعاب میپس I محتوای یک رجیستر عمومی (rs) را با صفر یا یک رجیستر عمومی دیگر (rt) به عنوان اعداد صحیح علامت دار مقایسه می‌کنند و اگر شرط تعیین شده برقرار باشد انشعاب می‌کنند. برای محاسبه‌ی مقصد پرش، آفست ۱۶ بیتی دو بیت به چپ شیفت داده می‌شود، نتیجه‌ی ۱۸ بیتی با علامت، گسترش داده می‌شود، و حاصل ۳۲ بیتی به حاصل جمع شمارنده‌ی برنامه (آدرس دستور) و ۸ اضافه می‌شود. پرش‌ها دو نوع اند: مطلق و غیر مستقیم. پرش‌های مطلق ("پرش" و "پرش و پیوند") برای محاسبه‌ی آدرس مقصد instr\_index را دو بیت به چپ شیفت می‌دهند و حاصل را به ۴ بیت مرتبه بالای آدرس دستوری که در شکاف تأخیر انشعاب آمده، پیوند می‌زنند. پرش‌های

غیرمستقیم، آدرس دستور مقصد را از یک رجیستر عمومی (rs) می‌گیرند. آدرسی که در رجیستر است باید بر اساس کلمه، تراز شده باشد، وگرنه پس از آن که دستوری که در شکاف تأخیر انشعاب قرار دارد اجرا شود یک استثنا پرتاب می‌شود. دستورهای انشعاب و پرشی که پیوند برقرار می‌کنند (به غیر از "پرش و پیوند رجیستر") آدرس بازگشت را در رجیستر عمومی ۳۱ ذخیره می‌کنند. دستور "پرش و پیوند رجیستر" اجازه می‌دهد آدرس برگشت در هر رجیستر عمومی قابل استفاده ای ذخیره شود.

## MIPS II

میپس II شکاف تأخیر بارگذاری را حذف کرد و مجموعه‌های مختلفی از دستورها را اضافه کرد. برای مثال، دستورهایی برای پردازش چندگانه با حافظه مشترک، بارگذاری کلمه‌ی پیوندیافته و ذخیره‌ی کلمه‌ی مشروط اضافه کرد. برای مثال، مجموعه‌ای از دستورهای Trap-on-Condition اضافه شد. این دستورها در صورت برقراری شرط، باعث استثنا می‌شوند. برای تمام دستورهای انشعاب موجود، نسخه‌های انشعاب گونه ای اضافه شد که هنگام انشعاب، دستورهای داخل شکاف تأخیر انشعاب را اجرا می‌کنند در برخی موارد این دستورها با ایجاد امکان قرار دادن دستورهای مفید در شکاف تأخیر انشعاب باعث بهبود عملکرد می‌شوند. همچنین دستورهای بارگذاری و ذخیره‌ی دوکلمه برای 3 – COP1 اضافه شد. همانند دستورهای دسترسی حافظه‌ی دیگر، در این بارگذاری‌ها و ذخیره‌ها دوکلمه باید به‌طور طبیعی تراز شده باشد.

همچنین دستورهای مختلفی به مجموعه دستورهای پردازنده کمکی نقطه شناور اضافه شد. برای نمونه، یک دستور جذر نقطه شناور سازگار با IEEE754 اضافه شد. این دستور از هردو نوع عملوند دقت یگانه و دقت دوگانه پشتیبانی می‌کند. علاوه‌بر آن، مجموعه دستورهایی برای تبدیل اعداد نقطه شناور دقت یگانه و دقت دوگانه به کلمات ۳۲ بیتی اضافه شد. این دستورها این امکان را فراهم می‌کردند که حالت گرد کردن IEEE، به جای کنترل نقطه شناور و رجیستر وضعیت، توسط دستورها مشخص شود و به این وسیله، دستورهای تبدیل موجود را کامل‌تر کردند.

ریزپردازنده R6000 از شرکت MIPS Computer Systems (۱۹۸۹) اولین پیاده‌سازی میپس II است. این ریزپردازنده برای سرورها طراحی شد. R6000 توسط Bipolar Integrated Technology ساخته و به بازار عرضه شد، اما با شکست تجاری مواجه شد. در اواسط دهه ۱۹۹۰، بسیاری از پردازنده‌های جدید ۳۲ بیتی میپس برای سیستم‌های جاسازی شده، پیاده‌سازی میپس II بودند، زیرا با معرفی معماری میپس III ۶۴ بیتی در ۱۹۹۱، میپس II تا معرفی میپس ۳۲ در ۱۹۹۹، جدیدترین معماری میپس ۳۲ بیتی باقی ماند.

## MIPS III

میپس III ، یک توسیع سازگار با گذشته از میپس II است که پشتیبانی از عملیات اعداد صحیح و آدرس دهی ۶۴ بیتی حافظه را اضافه کرده است. نوع داده‌ی ۶۴ بیتی دو کلمه نامیده می‌شود و میپس III رجیسترهای عمومی، رجیسترهای HI/LO و شمارنده‌ی برنامه را به ۶۴ بیت توسعه داد تا بتواند از آن پشتیبانی کند. دستورهای جدیدی برای بارگذاری و ذخیره‌ی دو کلمه‌ها، انجام جمع، تفریق، ضرب و تقسیم اعداد صحیح و همچنین عملیات‌های شیفت بر روی آن‌ها و انتقال دو کلمه‌ها بین رجیسترهای عمومی و HI/LO اضافه شد. دستورهای موجود که در اصل برای عمل کردن بر روی کلمه‌های ۳۲ بیتی تعریف شده بودند، هرجا نیاز بود بازتعریف شدند تا بتوانند حاصل‌های ۳۲ بیتی را با علامت توسعه دهند و اکثر دستورها بتوانند با کلمه‌ها و دو کلمه‌ها یکسان برخورد کنند. از میان این دستورهای بازتعریف شده می‌توان به بارگذاری کلمه اشاره کرد. در میپس III این دستور کلمه‌ها را با علامت به ۶۴ بیت توسعه می‌دهد. برای کامل کردن بارگذاری کلمه، نسخه‌ای اضافه شد که با صفر توسعه می‌داد.

ناتوانی قالب R ، در مشخص کردن مقدار کامل شیفت برای شیفت‌های ۶۴ بیتی (فیلد ۵ بیتی آن برای مقدار شیفت بسیار کوتاه‌تر از آن است که مقدار شیفت برای دو کلمه‌ها را نشان دهد) باعث شد میپس III ، سه نسخه ۶۴ بیتی از هر دستور شیفت میپس I ارائه دهد. اولین نسخه، نسخه‌ی ۶۴ بیتی دستورهای شیفت اصلی است، که برای مشخص کردن مقدار شیفت‌های ۰ تا ۳۱ بیتی استفاده می‌شود. دومین نسخه مشابه اولی است، اما به مقدار فیلد مقدار شیفت ، ۳۲ را می‌افزاید تا مقدارهای ثابت شیفت ۳۲ تا ۶۴ بیت را بتوان مشخص کرد. نسخه‌ی سوم ، مقدار شیفت را از ۶ بیت مرتبه‌ی پایین یک رجیستر عمومی به دست می‌آورد.

میپس III ، به سطوح امتیاز کرنل و کاربر که از قبل موجو بودند ، یک سطح ناظر افزود. این قابلیت تنها پردازنده‌ی کنترل سیستم تعریف شده در پیاده‌سازی (Coprocessor 0) را تحت تأثیر قرار داد.

ریزپردازنده‌ی R4000 شرکت MIPS Computer Systems (در سال ۱۹۹۱) ، اولین پیاده‌سازی میپس III بود. این ریزپردازنده برای استفاده در کامپیوترهای شخصی، ایستگاه‌های کاری و سرورها طراحی شده بود. MIPS Computer Systems به‌شدت معماری میپس و R4000 را ترویج داد و کنسرسیوم محیط محاسباتی پیشرفته (ACE) را برای پیشبرد استاندارد محاسبه‌ی ریسک پیشرفته (ARC) ایجاد کرد، که هدف آن، جا انداختن میپس به عنوان پلتفرم غالب برای محاسبات شخصی بود. ARC چندان موفقیتی در کامپیوترهای شخصی پیدا نکرد، اما R4000 ( و R4400 مشتق آن) به‌طور گسترده در کامپیوترهای ایستگاه‌های کاری و سرورها استفاده می‌شد،

به خصوص توسط بزرگترین کاربر آن Silicon Graphics. سایر کاربردهای R4000 شامل سیستم‌های جاگذاری شده پیشرفته و سوپر کامپیوترها می‌باشند.

پس از گذر زمان، میپس III توسط تعدادی ریزپردازنده‌ی جاگذاری شده پیاده‌سازی شد. ریزپردازنده‌ی R4600 شرکت Quantum Effect Design (سال ۱۹۹۳) و مشتقات آن، به‌طور گسترده، در سیستم‌های جاگذاری شده‌ی پیشرفته و ایستگاه‌های کاری ابتدایی و سرورها استفاده می‌شدند. ریزپردازنده‌ی R4200 شرکت MIPS Technologies (سال ۱۹۹۴) برای سیستم‌های جاگذاری شده، لپ‌تاپ‌ها و کامپیوترهای شخصی طراحی شد. یک مشتق آن، R4300i، ساخته شده توسط NEC Electronics، در کنسول بازی نینتندو ۶۴ استفاده شد. نینتندو ۶۴ در کنار پلی استیشن از بزرگترین استفاده‌کننده‌های پردازنده‌های معماری میپس در میانه‌ی دهه ۱۹۹۰ بودند.

## MIPS IV

میپس IV، چهارمین نسخه‌ی معماریست. این معماری، یک فوق مجموعه‌ی میپس III بوده و با تمام نسخه‌های موجود میپس سازگار است. انگیزه‌ی اصلی طراحی میپس IV، بهبود عملکرد نقطه شناور (FP) بود. برای بهبود دسترسی به عملوندها، یک حالت آدرس‌دهی اندیس‌گذاری شده (پایه + اندیس که منبع هردو رجیسترهای عمومی است)، برای بارگذاری‌ها و ذخیره‌سازی‌های FP اضافه شد، دستورهای فراخوانی زودرس نیز برای انجام فراخوانی زودرس حافظه و مشخص کردن راهنماهای کش (این‌ها، هر دو حالت آدرس‌دهی پایه + آفست و پایه + اندیس را پشتیبانی می‌کردند)، اضافه شد.

میپس IV، قابلیت‌های گوناگونی برای بهبود موازی کاری در سطح دستورها اضافه کرد. برای سبک کردن بار گلوگاهی که در اثر یک بیت شرطی به وجود می‌آید، هفت بیت کد شرطی به رجیسترهای نقطه شناور کنترل و وضعیت اضافه شدند و مجموع را به هشت رساندند. دستورهای مقایسه و انشعاب FP بازتعریف شدند تا بتوانند مشخص کنند کدام بیت شرطی، به ترتیب، نوشته یا خوانده می‌شود و شکاف تأخیر بین یک انشعاب FP که یک بیت شرطی را می‌خواند و مقایسه FP قبلی که آن بیت را نوشته است حذف شد. پشتیبانی از پیش‌بینی جزئی در قالب دستورهای انتقال شرطی، برای هردوی رجیسترهای عمومی و نقطه شناور اضافه شد و یک پیاده‌سازی می‌توانست بین استثنائات دقیق یا مبهم تله‌های IEEE 754، انتخاب کند.

میپس IV دستورهای جبری FP جدیدی برای هردوی اعداد نقطه شناور دقت یگانه و دوگانه اضافه کرد : ترکیب جمع یا تفریق با ضرب ، معکوس، معکوس ریشه‌ی دوم. دستورهای ترکیب جمع یا تفریق و ضرب نقطه شناور، یک یا دو گردکردن انجام می‌دهند (این امر در پیاده‌سازی تعریف می‌شود) ، به ترتیب برای رسیدن به دقت مورد نیاز IEEE754 یا رفتن فراتر از آن . دستورهای نقطه مشترک معکوس و معکوس ریشه دوم از دقت مورد نیاز IEEE754 پیروی نمی‌کنند و نتایج آن‌ها با دقت مورد نیاز ، در یک یا دو واحد جایگاه آخر، اختلاف دارد (در پیاده‌سازی تعریف می‌شود) . این دستورها هنگامی کاربرد دارند که تأخیر دستور، مهم‌تر از دقت آن است .

اولین پیاده‌سازی میپس IV ، ریزپردازنده‌ی R8000 شرکت MIPS Technologies بود (۱۹۹۴) . طراحی R8000 در شرکت Silicon Graphics,inc آغاز شد و تنها در ایستگاه‌های کاری و سرورهای پیشرفته، برای کاربردهای علمی و فنی که در آن کارایی بالا در بار کاری نقطه شناور زیاد مهم بود ، به کار رفت. پیاده‌سازی‌های بعدی R10000 از شرکت MIPS Technologies (سال ۱۹۹۶) و R5000 از شرکت Quantum Effect Device (سال ۱۹۹۶) و RM7000 (سال ۱۹۹۸) بود . R10000 نیز توسط NEC Electronics و Toshiba ساخته و به بازار عرضه شد . همچنین مشتقات آن توسط NEC و Pyramid Technology و Silicon Graphics,inc و Tandem Computers (درمیان دیگر شرکت‌ها) در ایستگاه‌های کاری، سرورها و ابررایانه‌ها به کاررفت علاوه بر آن، R5000 و R7000 در سیستم‌های جاگذاری شده پیشرفته شده، رایانه‌های شخصی، ایستگاه‌های کاری و سرورهای ابتدایی به کار رفت.

## MIPS V

میپس V ، معرفی شده در ۲۱ اکتبر ۱۹۹۶ در سمینار میکروپروسسور سال ۱۹۹۶ همراه با افزونه Digital Media Extensions یا MDMX ، برای بهبود عملکرد تحولات گرافیکی سه بعدی طراحی شد. در میانه‌ی دهه ۱۹۹۰، یک کاربرد اصلی ریزپردازنده‌های غیر جاگذاری شده‌ی میپس، ایستگاه‌های کاری گرافیکی SGI بود. میپس V با افزونه‌ی اعداد صحیح MDMX تکمیل شد تا یک سیستم کامل برای بهبود عملکرد برنامه‌های گرافیک سه بعدی فراهم آورد.

هیچگاه برای میپس V پیاده‌سازی‌ای معرفی نشد. در ۱۲ مه ۱۹۹۷، شرکت SGI خبر از ریزپردازنده‌های «H1» ملقب به «Beast» و «H2» ملقب به «Captain» داد. اولی قرار بود اولین پیاده‌سازی میپس V باشد و بنا بود در

نیمه‌ی اول ۱۹۹۹ معرفی شود. پروژه‌های «H1» و «H2» بعدها ترکیب شده و در نهایت در سال ۱۹۹۸ لغو شدند. با این که هیچ پیاده‌سازی‌ای از میپس V وجود ندارد، میپس ۶۴ توزیع ۱ (۱۹۹۹) بر پایه‌ی آن است که تمام قابلیت‌های آن را در یک پردازنده‌ی کمکی اختیاری (FPU) به نام Paired-Single حفظ کند.

میپس V نوع داده‌ی جدیدی معرفی کرد، یگانه جفت شده (PS)، که متشکل است از دو عدد نقطه شناور دقت یگانه (۳۲ بیتی) که در رجیسترهای ۶۴ بیتی نقطه شناور موجود ذخیره شده‌اند. صورت‌های دیگری از دستورهای نقطه شناور موجود برای عملیات جبری، مقایسه و انتقال شرطی اضافه شدند تا با این نوع داده به صورت SIMD کار کنند. دستورهای جدیدی برای بارگذاری، بازآرایی و تبدیل داده‌های PS اضافه شد. این اولین مجموعه دستور برای استفاده از SIMD نقطه شناور با منابع موجود بود.

## مقایسه‌ی معماری‌های MIPS و ARM

تعدادی از تفاوت‌های بین MIPS و ARM را می‌توان شناسایی کرد هر چند که هر دو در یک خانواده از مجموعه های دستورالعمل قرار دارند. برای همین موضوع، MIPS و ARM دو معماری با مجموعه دستورالعمل (ISA) هستند که در دنیای ریزپردازنده‌ها در دسترس هستند. MIPS و ARM هر دو مبتنی بر Reduction Instruction Set Computing (RISC) هستند و در نوع ثبت نام هستند. هر دو مجموعه دستورالعمل دارای اندازه دستورالعمل ثابت ۳۲ بیت / ۶۴ بیت (فضای آدرس) هستند و هر دو مجموعه دستورالعمل را می‌توان در endianness بزرگ و همچنین ماندگاری کمی پیکربندی کرد. هر دو معماری از سازگاری عقب پشتیبانی می‌کنند. معماری‌های هر دو MIPS و ARM در پردازنده‌های تلفن‌های هوشمند و رایانه‌های لوحی مانند آیفون‌ها، تبلت‌های اندرویدی و ویندوز RT استفاده می‌شود، اما در رایانه‌های اصلی جریان مانند لپ‌تاپ و سرور وجود ندارد.

### تفاوت بین MIPS و ARM چیست؟

- (۱) MIPS و ARM دو مجموعه معماری متفاوت در خانواده از مجموعه دستورالعمل RISC هستند.
- (۲) گرچه هر دو مجموعه دستورالعمل دارای یک اندازه دستورالعمل ثابت و یکسان هستند، ARM فقط ۱۶ ثابت دارد در حالی که MIPS دارای ۳۲ رجیستر است.



۳) ARM دارای توان بالا و راندمان بسیار خوبی نسبت به MIPS است زیرا پردازنده های ARM از اتوبوس های داده ۶۴ بیتی بین هسته و انبارها پشتیبانی می کنند.

۴) به منظور امکان تغییر کارآمد در زمینه ، معماری MIPS از اجرای چندین بانک ثبت نام پشتیبانی می کند. ARM برای اهداف حسابی و سایر توابع دیگر فقط ثبات هایی را در اختیار شما قرار می دهد ، اما MIPS برای ثبت نتایج عملیات ضرب ، دو رجیستر جداگانه فراهم می کند.

۵) MIPS هیچ دستورالعمل معادل با دستورالعمل ARM MOV ندارد.

۶) دستورالعمل MIPS ADD به طوعادی، استثنائی در سرریز ایجاد می کند ، بنابراین به ندرت از ARM استفاده می شود.

۷) تمام دستورالعمل های پردازش داده های ARM کدهای شرط ALU را بطور پیش فرض تنظیم می کنند ، اما MIPS SLT را برای مقایسه فراهم می کند.

و بطور کلی، در دنیای ریز پردازنده ها ، MIPS و ARM به نمایندگی از معماری های مجموعه آموزشی خود خدمات بزرگی را انجام می دهند. MIPSis در درجه اول در سیستم های تعبیه شده پیاده سازی شده است. اما ، در حال حاضر ، ARM در صنعت از MIPS محبوب تر شده است.