

به نام خدا

تمرین ۱ ساختار و زبان کامپیوتر

خانم دکتر ارشدی

سارا آذرنوش ۹۸۱۷۰۶۶۸

(۱)

۱ $O(n^2) = \text{radical}(k)$ است و پس از 6 سال (3^{82}) برابر میشود با $8 * \text{radical}(k)$

۲ $O(\log n) = \log(k)$ است و پس از 6 سال (3^{82}) برابر میشود با $8 * \log(k)$

۳ $O(n) = k$ است و پس از 6 سال (3^{82}) برابر میشود با $8 * k$

(۲)

۱- embedded system

۲- desktop

۳- server

(۳)

۱- server

۲- embdded system

۳- desktop

۴- desktop

۵- embdded system

(۴)

میپس ۱ دارای سی و دو رجیستر ۳۲ بیتی عمومی است. رجیستر ۰ \$ سخت افزاری صفر است و و هر آنچه در آن نوشته شود دور انداخته می شود. رجیستر ۳۱ \$ رجیستر پیوند است. برای دستورهای ضرب و تقسیم اعداد صحیح، که مستقل از دستورت دیگر اجرا می شود، یک جفت رجیستر ۳۲ بیتی، HI و LO ارائه شده است. فلوت نیز ۳۲ بیت است.

HI : مانند باقی مانده تقسیم است.

LO : مانند خارج قسمت است.

قالب‌های دستورها

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 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.

Registers for O32 calling convention

Name	Number	Use	Callee must preserve?
\$zero	\$0	constant 0	N/A
\$at	\$1	assembler temporary	No
\$v0-\$v1	\$2-\$3	values for function returns and expression evaluation	No
\$a0-\$a3	\$4-\$7	function arguments	No
\$t0-\$t7	\$8-\$15	temporaries	No
\$s0-\$s7	\$16-\$23	saved temporaries	Yes
\$t8-\$t9	\$24-\$25	temporaries	No
\$k0-\$k1	\$26-\$27	reserved for OS kernel	N/A
\$gp	\$28	global pointer	Yes (except PIC code)
\$sp	\$29	stack pointer	Yes
\$fp	\$30	frame pointer	Yes
\$ra	\$31	return address	N/A

Registers for N32 and N64 calling conventions

Name	Number	Use	Callee must preserve?
\$zero	\$0	constant 0	N/A

\$at	\$1	assembler temporary	No
\$v0-\$v1	\$2-\$3	values for function returns and expression evaluation	No
\$a0-\$a7	\$4-\$11	function arguments	No
\$t4-\$t7	\$12-\$15	temporaries	No
\$s0-\$s7	\$16-\$23	saved temporaries	Yes
\$t8-\$t9	\$24-\$25	temporaries	No
\$k0-\$k1	\$26-\$27	reserved for OS kernel	N/A
\$gp	\$28	global pointer	Yes
\$sp	\$29	stack pointer	Yes
\$s8	\$30	frame pointer	Yes
\$ra	\$31	return address	N/A

(۵)

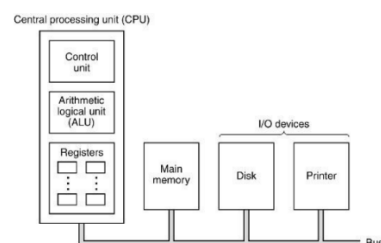
۱- نادرست، رجیسترها حافظه و حجم کمتر و سرعت بیشتری دارند.

۲- نادرست، از execute, examine, fetch تشکیل شده است.

۳- درست، بنا به شکل داخل اسلاید



- The engine that interprets (*executes*) instructions stored in main memory



۴- نادرست، هم دستور هم عمل ها هم اطلاعات به صورت ۰ و ۱ هستند.

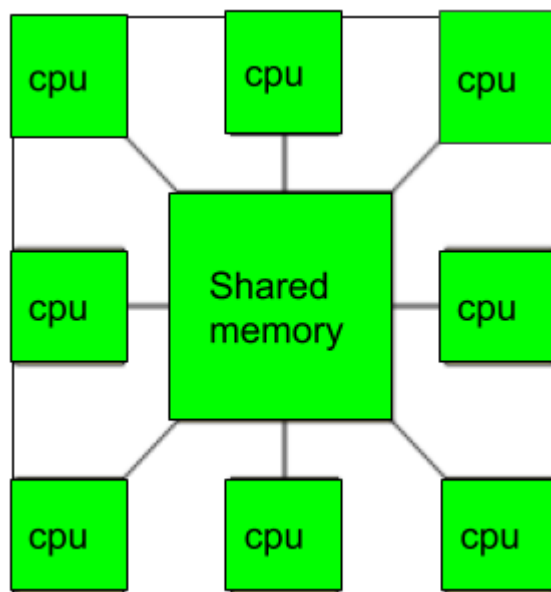
۶) ۱. چند پردازنده:

Multiprocessor یک سیستم رایانه ای با دو یا چند واحد پردازش مرکزی (CPU) است که دسترسی کامل به یک RAM مشترک را دارند. هدف اصلی استفاده از چند پردازنده افزایش سرعت اجرای سیستم اهداف تحمل خطا و تطبیق برنامه است.

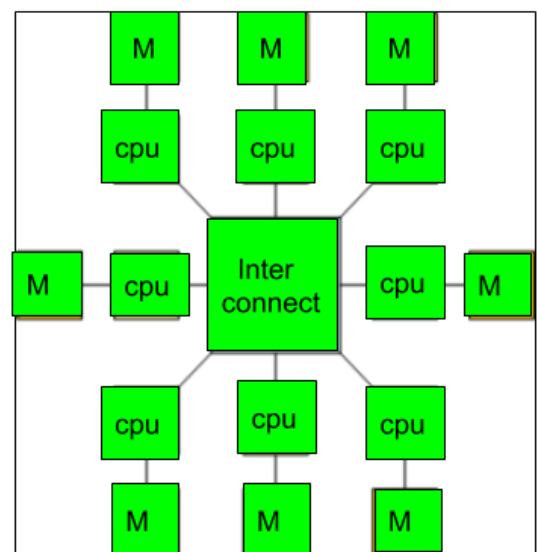
چند پردازنده دو نوع دارد، یکی چند پردازنده حافظه مشترک و دیگری پردازنده حافظه توزیع شده نامیده می شود. در چند پردازنده حافظه مشترک، همه پردازنده ها حافظه مشترک دارند اما در یک پردازنده چند حافظه توزیع شده، هر پردازنده حافظه خصوصی خود را دارد.

۲. چند رایانه:

سیستم چند رایانه ای یک سیستم رایانه ای با چندین پردازنده است که برای حل یک مشکل به هم متصل شده اند. هر پردازنده حافظه خاص خود را دارد و توسط پردازنده خاص قابل دسترسی است و این پردازنده ها می توانند از طریق یک شبکه اتصال به یکدیگر ارتباط برقرار کنند. از آنجا که چند رایانه توانایی ارسال پیام بین پردازنده ها را دارد، برای انجام کار می توان کار را بین پردازنده ها تقسیم کرد. از این رو، می توان از یک چند رایانه برای محاسبات توزیعی استفاده کرد. ساخت چند رایانه نسبت به پردازنده چند منظوره مقرون به صرفه و آسان است.



multiprocessor



multicomputers

MULTIPROCESSOR VERSUS MULTICOMPUTER

MULTIPROCESSOR	MULTICOMPUTER
A system with two or more CPUs that allows simultaneous processing of programs	A set of processors connected by the communication network that works jointly to solve a computation problem
Easier to process	Less easy to program
More difficult and costly to build	Easier and cost effective to build
Supports parallel computing	Supports distributed computing
	Visit www.PEDIAA.com

(۷)

۱- نادرست، با افزایش قیمت سرعت افزایش و حجم کاهش میابد.

۲- درست، **pipelining** مجموعه‌ای از عناصر (مراحل) پردازش داده است که به صورت سری به یکدیگر متصلند و ورودی هر عنصر، خروجی عنصر قبلی است. در تکنیک **pipeline**، چند دستور می‌توانند در یک زمان اجرا شوند. یعنی دستورهای از لحاظ زمان اجرا دارای همپوشانی هستند.

۳- نادرست، کلمه دستورالعملی بسیار طولانی (VLIW) به معماری واحد پردازنده مرکزی اشاره می‌کند که برای بهره‌برداری از فواید توازی در سطح دستورالعمل (ILP) طراحی شده است. پردازشگری که هر دستورالعمل را بعد از دیگری اجرا می‌کند (معماری غیر خط لوله‌ای دارد)، ممکن است منابع پردازش را به نحو فریبنده‌ای

استفاده کند که این امر می‌تواند باعث عملکرد ضعیف شود. کارایی می‌تواند با اجرای همزمان زیر گامهای مختلف از دنباله‌های دستورها (Pipelining) یا حتی اجرای دستورها چندتایی به شکل کاملاً همزمان درست مانند معماری‌های Superscalar، بهبود یابد. برای بهبود بیشتر می‌توان دستورالعملها را با نظم متفاوت از آنچه در برنامه نشان داده شده‌اند اجرا کرد. به این عمل اجرای خارج از نظم گفته می‌شود.

(۸)

معماری فون نویمان :

یک طرح نظری مبتنی بر مفهوم رایانه‌های برنامه ذخیره شده است که در آن داده‌های برنامه و داده‌های دستورالعمل در همان حافظه ذخیره می‌شوند. ایده موجود در معماری‌های فون نویمان امکان ذخیره دستورالعمل‌ها در حافظه به همراه داده‌هایی است که دستورالعمل‌ها روی آن کار می‌کنند. به طور خلاصه، معماری فون نویمان به یک چارچوب کلی اشاره دارد که باید سخت افزار، برنامه نویسی و داده‌های رایانه از آن پیروی کند.

معماری فون نویمان از سه مؤلفه مجزا تشکیل شده است: واحد پردازش مرکزی (CPU)، واحد حافظه و رابط ورودی / خروجی CPU. (I / O) قلب سیستم رایانه ای است که از سه مؤلفه اصلی تشکیل شده است: واحد حساب و منطق (ALU)، واحد کنترل (CU)، و ثبت‌ها.

معماری هاروارد:

یک معماری رایانه ای است که دارای مسیرهای ذخیره سازی و سیگنال جداگانه برای داده‌ها و دستورالعمل‌های برنامه است. برخلاف معماری فون نویمان که از یک اتوبوس واحد استفاده می‌کند تا بتواند دستورالعمل‌های حافظه را منتقل کند و داده‌ها را از یک بخش از رایانه به بخش دیگر انتقال دهد، معماری هاروارد فضای حافظه جداگانه ای برای داده‌ها و دستورالعمل‌ها دارد.

هر دو مفهوم به جز نحوه دسترسی به خاطرات مشابه هستند. ایده معماری هاروارد تقسیم حافظه به دو بخش است - یکی برای داده‌ها و دیگری برای برنامه‌ها. این اصطلاحات بر روی رایانه اصلی مبتنی بر رله هاروارد مارک اول ساخته شده بود که دارای سیستمی بود که امکان انجام همزمان داده‌ها و نقل و انتقالها را نیز فراهم می‌آورد.

طراحی رایانه‌های دنیای واقعی در واقع بر پایه معماری اصلاح شده هاروارد استوار است و معمولاً در میکروکنترلرها و DSP (پردازش سیگنال دیجیتال) مورد استفاده قرار می‌گیرد.

تفاوت بین فون نویمان و معماری هاروارد

مبانی معماری :

معماری فون نویمان یک طراحی نظری رایانه است که مبتنی بر مفهوم برنامه ذخیره شده است که در آن برنامه‌ها و داده‌ها در همان حافظه ذخیره می‌شوند. در حال حاضر به عنوان پایه تقریباً همه رایانه‌های مدرن خدمت

می کند. معماری هاروارد بر پایه مدل رایانه ای مبتنی بر رله هاروارد مارک اول بود که از اتوبوس های جداگانه برای داده ها و دستورالعمل ها استفاده می کرد.

سیستم حافظه معماری:

معماری فون نویمان تنها یک اتوبوس دارد که برای واگذاری دستورالعمل ها و انتقال داده ها استفاده می شود، و این عملیات باید برنامه ریزی شود زیرا نمی توانند همزمان انجام شوند. از طرف دیگر معماری هاروارد از فضای حافظه جداگانه ای برای دستورالعمل ها و داده ها برخوردار است، که از لحاظ جسمی سیگنال ها و ذخیره سازی برای کد و حافظه دیتا را از هم جدا می کند که به نوبه خود امکان دستیابی همزمان به هر یک از سیستم های حافظه را فراهم می کند.

پردازش دستورالعمل:

در معماری فون نویمان، واحد پردازش برای تکمیل یک دستورالعمل به دو چرخه ساعت نیاز دارد. پردازنده در چرخه اول دستورالعمل را از حافظه می گیرد و آن را رمزگشایی می کند، و سپس داده ها از چرخه دوم از حافظه گرفته می شود. در معماری هاروارد، واحد پردازش می تواند در صورت وجود راهکارهای مناسب لوله کشی (، دستورالعمل را در یک چرخه انجام دهد.

هزینه معماری:

از آنجا که دستورالعمل ها و داده ها از همان سیستم اتوبوس در معماری فون نویمان استفاده می کنند، طراحی و توسعه واحد کنترل را ساده می کند، که در نهایت هزینه تولید را به حداقل می رساند. توسعه واحد کنترل در معماری هاروارد به دلیل معماری پیچیده ای که از دو اتوبوس برای دستورالعمل و داده استفاده می کند، گرانتر از سابق است.

استفاده:

معماری فون نویمان عمدتاً در هر دستگاهی که از رایانه های رومیزی و نوت بوک ها گرفته تا رایانه ها و ایستگاه های کاری با کارایی بالا مشاهده می کنید، استفاده می شود. معماری هاروارد یک مفهوم نسبتاً جدید است که در درجه اول در میکروکنترلرها و پردازش سیگنال های دیجیتال مورد استفاده قرار می گیرد.

خلاصه:

معماری فون نویمان مشابه معماری هاروارد است، مگر اینکه از یک اتوبوس منفرد برای انجام هر دو بخش دستورالعمل و انتقال داده استفاده کند، بنابراین عملیات باید برنامه ریزی شود. از طرف دیگر معماری هاروارد از دو آدرس حافظه جداگانه برای داده ها و دستورالعمل ها استفاده می کند که باعث می شود همزمان داده ها در هر دو اتوبوس امکان پذیر باشد. با این حال، معماری پیچیده فقط به هزینه توسعه واحد کنترل در برابر هزینه توسعه پایین تر معماری کمتر پیچیده فون نویمان که از یک حافظه پنهان یکپارچه استفاده می کند، اضافه می کند.

VON NEUMANN ARCHITECTURE VERSUS HARVARD ARCHITECTURE

It is a theoretical design based on the stored-program computer concept.	It is a modern computer architecture based on the Harvard Mark I relay-based computer model.
It uses same physical memory address for instructions and data.	It uses separate memory addresses for instructions and data.
Processor needs two clock cycles to execute an instruction.	Processor needs one cycle to complete an instruction.
Simpler control unit design and development of one is cheaper and faster.	Control unit for two buses is more complicated which adds to the development cost.
Data transfers and instruction fetches cannot be performed simultaneously.	Data transfers and instruction fetches can be performed at the same time.
Used in personal computers, laptops, and workstations.	Used in microcontrollers and signal processing.