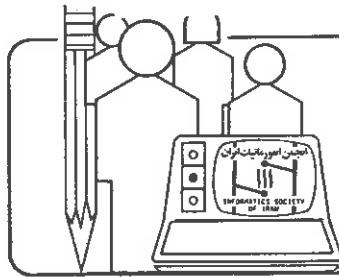


گزارش کامپیوتر

ماهنامه انجمن انفورماتیک ایران



سال سوم - شماره ۳
شماره پیاپی ۲۴
خرداد ۱۳۶۰
تک شماره ۵۰ ریال - اشتراک سالانه ۵۰۰ ریال



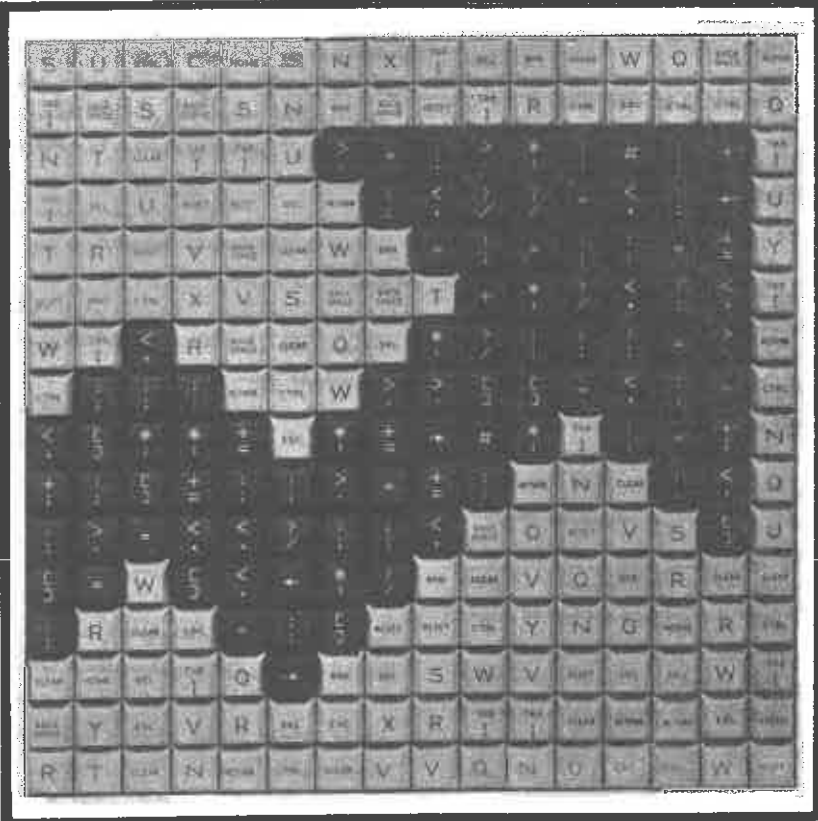
اخبار انجمن

سخنرانی ماهیانه انجمن برای تیر ۱۳۶۰

هفتمین سخنرانی ماهیانه انجمن طبق معمول همراه در آخرین چهارشنبه ماه بعد (۳۱ تیر ۱۳۶۰) از ساعت ۲/۵ بعد از ظهر در محل موسسه آموزش عالی آمار و انفورماتیک (خیابان پاسداران، ترسیده به دولت) برگزار خواهد شد. سخنران این جلسه آقای دکتر بهروز پرهامی هستند و عنوان سخنرانی "شناسائی متون چاپی فارسی توسط کامپیوتر" است که خلاصه آن در زیر بنظر خوانندگان گرامی میرسد.

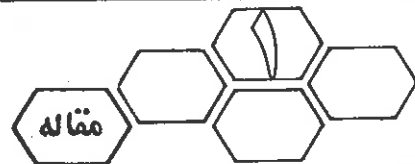
خلاصه: شناسائی متون چاپی فارسی نسبت به متون سایر زبانها بدلیل زیر مشکل تر است: (الف) اتصال حروف به یکدیگر، (ب) شباهت زیاد برخی از حروف، (پ) پهنای کلمات، متفاوت و (ت) روی هم قرار گرفتن حروف. در این سخنرانی، یک روش شناسائی متون چاپی فارسی و مراحل مختلف آن بشرح زیر مورد بحث قرار خواهد گرفت: (۱) رقمی ساختن متن، (۲) ویرایش، (۳) جدا سازی سطرها، (۴) جدا سازی زیرکلمه ها، (۵) جدا سازی نمادها، (۶) شناسائی و (۷) پس پردازش. همچنین الگوریتم های مربوط به مراحل فوق، بخصوص مراحل پنجم و ششم، همراه با کاربردهای عملی آنها در شناسائی متون چاپی درشت اراشه خواهد گردید. روش شناسائی بر اساس برخی از خواص هندسی حروف فارسی از قبیل پهنای نسبی، وجود فرورفتگی ها و وجود حلقه ها بنا شده است. رویهم رفته بیست خصیصه هندسی برای این منظور بکار میرود که یک بردار ۲۴ بیتی را برای هر علامت مورد بررسی تشکیل میدهند. این بردار با الگوهای موجود برای حروف و علائم فارسی مقایسه شده و در موارد نادری که تطابق کامل با هیچیک از الگوها موجود نباشد، جستجو برای حداکثر تطابق ممکن (با اولویت دادن به آندسته از خصیصه ها که دارای قابلیت اطمینان بیشتری هستند) صورت میگیرد. سیستم مورد بحث قادر است که الگوهای جدیدی را برای حروف

دنباله در صفحه ۱۶



در این شماره

- ۱ اخبار انجمن ... سخنرانی ماهیانه انجمن برای تیر ۱۳۶۰
- ۱۴ اخبار جهان تقای "تی آی" در بازار کامپیوترهای خانگی - ژنر بیشتر علی رغم رکود
- ۱۶ نوآوری ها صحبت با تلویزیون
- ۱۶ سرگرمی حروف بجای ارقام - حل جدول کلمات متقاطع شماره قبل
- ۵ مقاله ویژه ... اعداد بزرگ در کامپیوتر و ریاضیات (بهروز پرهامی)
- ۲ مقاله ۱ واقعیت های درباره ریزپردازنده ها (ترجمه حسن کا تویان)
- ۳ مقاله ۲ مدارهای مجتمع الکترونیکی: از تحرک الکترونی تا ساخت منطقی قسمت دوم (ترجمه محمد میرزا عبداللهی و بهروز پرهامی)
- ۱۱ مقاله ۳ برنامه نویسی بعنوان نظامی با طبیعت ریاضی (ترجمه ابراهیم مشایخ)
- ۱۳ مقاله ۴ کنترل ماده توسط ذهن: بازخورد زیستی برای کامپیوتر (ترجمه فرهاد ماحیان)
- ۱۵ مقاله ۵ آزمایش موتور اتمی بیل بکمی کامپیوتر (ترجمه ویدا دانی)



واقعیت‌هایی درباره ریزپردازنده‌ها

در اغلب بازارهای فروش دستگاه‌های کامپیوتری، ریزپردازنده‌ها مورد توجه بسیاری واقع شده‌اند. دلیل این توجه را می‌توان کوچکی، ارزانی و قدرتمندی دستگاه‌های فوق دانست. قسمت اساسی و مرکزی یک واحد پردازنده، مدار مجتمع و بسیار کوچک و پیچیده موسوم به "تراشه" (IC) است. تراشه با افزودن یک حافظه فقط خواندنی (ROM) و خروجی کمکی، ریز پردازنده تبدیل به یک ریزپردازنده می‌شود.

این پردازنده‌ها نخستین بار در اواخر دهه ۱۹۶۰ میلادی و در ماشین‌های حساب مورد استفاده قرار گرفتند. متعاقباً در سال ۱۹۷۲، شرکت اینتل نمونه‌ای از یک ریزپردازنده موفق بنام ۸۰۸۰ را ارائه نمود که هنوز نیز دارای میزان فروش بالایی است.

پیدایش ریزپردازنده‌ها این امید را بوجود آورد که هر نوع وسیله و سیستمی که دارای عنصر کنترل باشد، از کنترل الکترومکانیکی گرفته تا مدارهای منطقی قابل برنامه‌نویسی و از بازیهای الکترونیکی گرفته تا دستگاه‌های عظیم کامپیوتری، با کمک آنها کوچک‌تر، ارزان‌تر و دارای بازده بیشتری گردد. این واقعیت باعث کاربرد روزافزون ریزپردازنده‌ها در دنیای صنعت و تجارت شد.

بزرگترین مانع در مقابل توسعه بازار هم بیشتر ریزپردازنده‌ها را می‌توان در یک کلمه خلاصه کرد: "هزینه". گرچه شرکت‌ها با تولید هرچه بیشتر ریزپردازنده‌ها تجارت بسیاری می‌اندوزند که بازده کار آنها را بالا برده و هزینه‌هایشان را پائین می‌آورد، ولی سرمایه‌گذاری‌های اولیه برای طرح و تبلیغ هر محصول تنها از عهده معدودی از شرکتها بر می‌آید. بنابراین باید انتظار داشت که طی تقریباً ۱۰ سال آینده، فقط گروه کوچک و برگزیده‌ای از شرکت‌های سازنده هنوز قادر به ادامه تولید باشند.

بعنوان مثال، در زمینه کاربرد ریزپردازنده‌ها در صنعت شیرینی‌سازی، کشورهای مانند ایالات متحده آمریکا، آلمان و سوئیس با سرمایه‌گذاری‌های سنگین رفته‌رفته خود را در صدر بازار دنیا قرار می‌دهند. شرکت‌های دیگر از کشورهایی مانند انگلستان قادر به چنین سرمایه‌گذاری‌هایی نیستند. بنابراین انتظار می‌رود که شرکت‌های فوق در آینده از بازار رقابت عقب‌افتند و به رده‌های پائین‌تری تنزل نمایند.

رقابت در صنایع کامپیوتری توافقی با حرمی و طمع فراوان است. سازندگان این دستگاه‌ها

باید همچون شطرنج بازان، زیرکی فراوانی بخرج دهند. می‌توان گفت که تنها نور امید برای وضع اقتصادی خراب در بعضی از ممالک پیشرفته اروپائی نظیر انگلستان، همین صنعت ساخت سیستم‌های کامپیوتری است.

اما این صنعت خود می‌تواند نوعی دیگر از بحران را موجب شود. مثلاً صنایع انگلستان هرچه بیشتر در جهت کاستن از تعداد کارکنان و کاربرد سیستم‌های خودکار پیش می‌روند. این موضوع طبیعتاً می‌تواند باعث افزایش جمعیت بیکاران شود. البته برنامه کنترل خودکار در کارخانه‌ها هنوز بدان صورت حاد مطرح نیست و از آن بعنوان طرح بلند مدت یاد نمی‌کنند.

اما چندی پیش در یک نمایشگاه کامپیوتری در انگلستان فیلمی بر روی صحنه آمد که در آن نظری به وضعیت ادارات در آینده‌ای نه چندان دور انداخته شده بود. در فیلم فوق ضرورت تکنولوژی مدرن و کاربردهای آرایانه‌ای واضح نشان داده شده: مثلاً زوی میز مدیر یک اداره، کامپیوتری قرار داشت تا مدیر بتواند توسط آن با سرعت و سهولت به بانک‌های اطلاعاتی دسترسی داشته باشد. خدمات دیگر این کامپیوتر شامل ارتباط رادیو تلوویزیونی با سایر ارگان‌های اداری و نیز نمایش اطلاعات لازم بر صفحه پایانه کامپیوتر بود. البته آنچه در فیلم فوق الذکر نمایش داده شد اکنون نیز در برخی از کاربردهای خاص به واقعیت پیوسته است ولی هنوز استفاده وسیع از این دستگاه‌ها در ادارات معمولی مقرون به صرفه نمی‌باشد. اما در آینده مدیران و رؤسا بتدریج باین موضوع پی می‌آورند برد که عمل شیء‌سازی طرح‌ها روی کامپیوتر کاری بس ساده‌تر از رسم نمودار و جدول می‌باشد و باید بسوی آن رو آورند چرا که در غیر اینصورت در بازار رقابت عقب افتاده و نابود خواهند شد.

از همه اینها گذشته، باید دید که امروزه چه امکانات و وسعت عملی در کاربرد ریزپردازنده‌ها وجود دارد. تکنولوژی ریزپردازنده‌ها باعث تولد پدیده‌ای جدید در صحنه تولیدات کامپیوتری شده است: اشاره ما به بازیهای کامپیوتری خانگی است که در حال حاضر بازار کشورهای غربی را تسخیر نموده‌اند.

شرکت "آتاری" یکی از تولیدکنندگان اینگونه بازیهای بصری الکترونیکی است که بازی معروف "پشتازان فضا" را به بازار ارائه نموده است. ولی آیا اینگونه بازیها برای همیشه پررونق باقی خواهند ماند و اثرات تکنولوژیکی و تجاری مهمی بر صنعت کامپیوتر خواهند داشت یا اینکه پس از چندی همانطور که سرچا "مورد توجه واقع شدند سرعت نیز از خاطرها محو خواهند گردید؟ طبیعت بازار فروش این سرگرمی‌های الکترونیکی در حال حاضر بقدری متغیّر و آسیب‌پذیر است که پیش‌بینی جهت‌های آن مشکل می‌نماید. تولیدکنندگان اینگونه

محصولات امروزه دست‌بیکار مشترک زده‌اند تا بتوانند مشتری‌های هرچه بیشتری را جلب نمایند و شرکتها، که بسیاری از آنها کاملاً نوپا هستند، اگر در شرایط مناسب‌زمانی و دست‌بیکار شده باشند موفق شده و به ثروت می‌رسند.

اما باید دانست که بازار حالت ثبات ندارد. چندهای از ارائه یک محصول جدید نمی‌گذرد که بلافاصله نمونه‌ای جدیدتر توسط شرکت رقیب عرضه میگردد. مثلاً شرکت‌های "فیلیپس" و "تیل" توانسته‌اند سیستم‌های مشابه محصولات شرکت "آتاری" را بسازند. شرکت دیگری بنام "سامیت کوپنر" نیز، با وجود اینکه بیش از سه سال از تأسیس آن نمی‌گذرد، با جدیدترین روشها در ساخت سیستم‌های مبتنی بر ریزپردازنده‌ها وارد بازار شده و موفق هم بوده است.

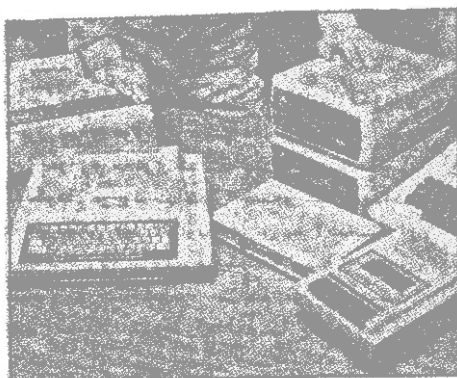
بموازات توسعه بازار فروش، دزدی الکترونیکی هم در غرب بازار داغی دارد. یکی از سازندگان عمده بازیهای الکترونیکی از این موضوع شکایت دارد که تعسّدی از شرکت‌های کوچک و خرده‌پا ناگهان با ارائه یک محصول تقلید شده از نمونه‌های اولیه آنها، در بازار عرضه اندام میکنند. این شرکتها گاه حتی طرح را کاملاً از نمونه اصلی کپی کرده و با نام دیگری در بازار بفروش می‌رسانند و چون هزینه تحقیق و طراحی ندارند، قیمت تمام شده محصولاتشان نسبت به انواع اصلی پائین‌تر است.

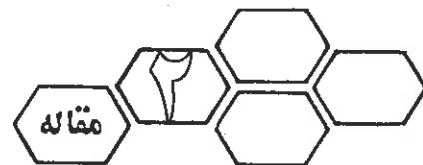
پیش‌بینی‌های منتشر شده در زمینه رشد صنایع الکترونیک در انگلستان نشان میدهند که بازار بازیهای الکترونیکی در سال ۱۹۸۰ به اوج خود رسیده و از این پس در سراسر سقوط قرار خواهد گرفت. طبق این پیش‌بینی‌ها، رقم ۴/۲ میلیارد واحد الکترونیکی فروخته شده در سال ۱۹۸۰ در انگلستان تا سال ۱۹۸۲ به نصف تقلیل خواهد یافت.

پادداشت‌ها

1. LSI chip
2. ROM = Read-only memory
3. RAM = Random-access memory

ترجمه و خلاصه شده توسط حسن کا تویان
دانشجوی دانشگاه صنعتی شریف
از "کامپیوتینگ"، ۲۹ ژانویه ۱۹۸۱





مدارهای مجتمع الکترونیکی:

از تئوریک الکترونی تا ساخت منطقی

(قسمت دوم)

توضیح و ویرایشگر: آنچه در زیر بنظر خوانندگان گرامی میرسد قسمت دوم از یک مقاله تومیسفی چند قسمتی است که از شماره سپتامبر ۱۹۸۰ مجله "کامپیوتینگ سورویز" به فارسی ترجمه شده است. قسمت اول مقاله "گزارش کامپیوتر" شماره بیای بی ۲۲، فروردین ۱۳۶۰ حاوی بخش هاشی با عناوین زیر بود:

- ۱- طراحی مدارهای مجتمع ترانزیستوری
 - ۱/۱- خواص فیزیکی جامدات
 - ۱/۱/۱- عایقها، هادیها و نیمه هادیها
 - ۱/۱/۲- غنی کردن نیمه هادی ها از نظر حامل های بار
 - ۱/۲- عناصر نیمه هادی
 - ۱/۲/۱- اتصال نیمه هادی: جنگ بارها
- اینک مطلب را از ابتدای بخش ۱/۲/۲ دنبال میکنیم.

۱/۲/۲ - ترانزیستورهای "تی تی ال" و "ماس"

در مدارهای رقمی، "ترانزیستور" نقش یک راه گزین را ایفا میکند، یعنی مؤلفه ای است که بسته به "روشن" یا "خاموش" بودن، به بار الکتریکی اجازه عبور داده یا از عبور آن جلوگیری میکند. در حال حاضر خانواده های مختلفی از مدارهای الکترونیکی وجود دارند که برخی از آنها عبارتند از: "تی تی ال"، "ماس"، "سی ماس"، "سی سی دی"، "سی سی ال" و "آی آی ال" یا اصطلاحاً "آی بیوان دو ال". بعضی از خانواده های فوق، در انواع "پی" و "ان" یا کم مصرف و پرمصرف و بالاخره سریع و کند در دسترس هستند و باین ترتیب فهرست انواع مدارهای موجود را طولانی تر میکنند. بحث ما فقط به ترانزیستورهای "تی تی ال" و "ماس" محدود میشود و در این محدوده نیز تکنولوژی "تی تی ال" با اختصار مورد بررسی قرار میگیرد تا فرصت بیشتری برای بحث در باره تکنولوژی ساده تر و در نتیجه قابل فهم تر "ماس" موجود باشد.

ترانزیستور "ماس" حدود ۵۰ سال پیش توسط "جی بی لیلیپیتال" ضمن پژوهش در زمینه تقویت بار الکتریکی اختراع شد (مآخذ ۳). گرچه ترانزیستور اختراع شده تا حدودی کار خود را انجام میداد، مبنای نظری طرز کار آن حدود ۲۰ سال بعد کشف شد. تقریباً در همان زمان بود که "جی باردین"، "دبلیو ایج براتین" و "دبلیو شاکلی" ترانزیستور دوقطبی را اختراع کردند (مآخذ ۲ و ۴). این اختراع بعدت یک دهه تمام توجه متخصصین این رشته را بخود معطوف کرد.

ترانزیستورهای دوقطبی اولیه بی قواره، غیر قابل اطمینان، پرهزینه و در عین حال

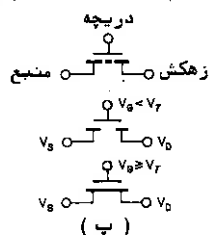
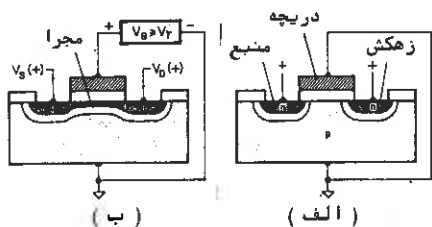
ولی اگر بصورت شکل ۸ - ب باطری A رانیز که ولتاژ خیلی کمتری دارد در مدار قرار دهیم، دو جریان مختلف بوجود خواهد آمد: حفره ها از طریق پایه وارد میشوند در حالیکه الکترون ها از هردو باطری به بیانه کننده فرستاده میشوند. جنگ بین پایه و سانه کننده تشدید میشود و الکترون های پُر انرژی از هردو باطری بسمت پایه رانده میشوند.

تلفات اولیه مربوط به الکترون هاشی است که توسط حفره ها خنثی شده اند. اما چون منطقه پایه بسیار نازک است، الکترون هاشی که با فشار زیاد به جلو رانده میشوند از پایه عبور کرده و بطرف روبنده میروند و در آنجا به همزمان خود می پیوندند. اگر باطری A قطع شود، وضعیت جنگی با بُن بست مواجه شده و جریان های الکتریکی فوراً قطع میشوند.

بنابراین می بینیم که جریان کوچکی در حلقه A جریان چندین برابر خود را در حلقه B کنترل میکند. در مدارهای "تی تی ال"، حلقه A مربوط به یک ترانزیستور جزئی از شاخه های حلقه B در یک ترانزیستور دیگر میباشد و این رابطه سلسله مراتبی بهمین ترتیب ادامه پیدا میکند.

علامت اختصاری یک ترانزیستور "ان پی ان" در مدارها در شکل ۸ - ب دیده میشود. توجه کنید که در شکل فوق الذکر، نوک پیکان در خلاف جهت جریان قرار دارد.

ترانزیستورهای "ماس" یک قطبی و بسیار ساده اند. شکل ۹ نوعی از این ترانزیستورها را با حامل "ان" نشان میدهد. دو منطقه مجزای نوع "ان" بنامهای "منبع" و "زهکش" از طریق دریچه هاشی در سطح سفرفه نوع "پی" نفوذ داده شده و از طریق پلر عایقی بنام "دریچه" بهم وصل شده اند. اگر دریچه سفرفه بصورت شکل ۹ - الف بیکدیگر متصل شوند، منبع و زهکش (بشرط اینکه هیچکدام از نظر الکتریکی منفی تر از دریچه نشود) از هم جدا خواهند بود.



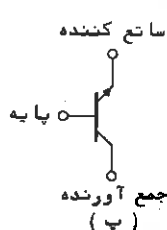
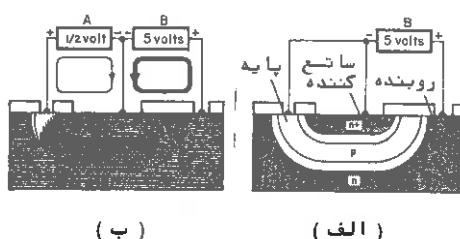
شکل ۹ - ترانزیستور "ماس" با مجرای نوع "ان" (الف) در حالت قطع، (ب) در حالت وصل و (پ) علامت های نمایی.

امیدوار کننده بودند. در این نمونه های اولیه، بجای سیلیسیم، ژرمانیوم مورد استفاده قرار میگرفت و ترانزیستور حاصل با مؤلفه ایده آل فاصله زیادی داشت. کار طراح عبارت بود از سرهم کردن تعداد زیادی ترانزیستور، مقداری از مؤلفه های دیگر و مقدار زیادی سیم برای جبران نمودن معایب و اشکالات موجود. این امر همواره به افزایش پیچیدگی مدارها منجر میشد.

با پیدایش روش نفوذ دادن ناخالصی در سیلیسیم، ساختمان شتوی امکان پذیر گشت و ابعاد فوق العاده کوچک، که برای ایجاد تغییر حالت در گری از میکرو ثانیه لازم بود، بدست آمد. ترانزیستورهای امروزی از نظر قابلیت اطمینان، هزینه و روشهای مهندسی ساخت و تولید، عناصر شگفت انگیزی هستند.

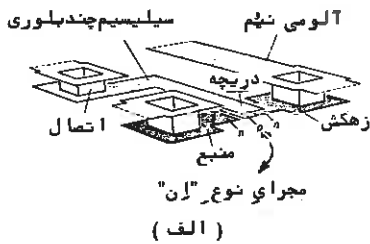
جالب اینکه پس از عطف توجه مجدد به ترانزیستورهای "ماس"، این نکته کشف شد که برخلاف مبانی نظری موجود، این ترانزیستورها کار خود را آنطور که انتظار میرفت انجام نمیدادند. پس از مطالعات گسترده و صرف وقت و انرژی زیاد، معلوم شد که عامل این عدم موفقیت، نمک (همان نمک خوراکی که بوفور در روی زمین یافت میشود) بوده است. باین ترتیب که مقادیر بسیار ناچیزی از نمک دی اکسید سیلیسیم را آلوده کرده و خاصیت عایق بودن آنرا زایل میکرده است. تا کید بیشتر بر تمیز بودن محیط و خالص سازی مواد مورد استفاده، بالاخره این مشکل راحل کرد.

ترانزیستورهای "تی تی ال"، همگی از نوع "ان پی ان" هستند (شکل ۸). سه منطقه ای که در این ترانزیستور وجود دارند بترتیب سانه کننده، پایه و روبنده نامیده میشوند. اگر باطری B مطابق شکل ۸ - الف به ترانزیستور متصل شود، مقداری بار الکتریکی از پایه و روبنده خارج شده و بر اثر تخلیه بار الکتریکی در نواحی مرزی، عبور جریان متوقف میشود.

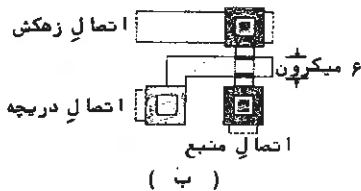


شکل ۸ - ترانزیستور "تی تی ال" (الف) در حالت قطع، (ب) در حالت وصل و (پ) علامت نمایی در مدارها.

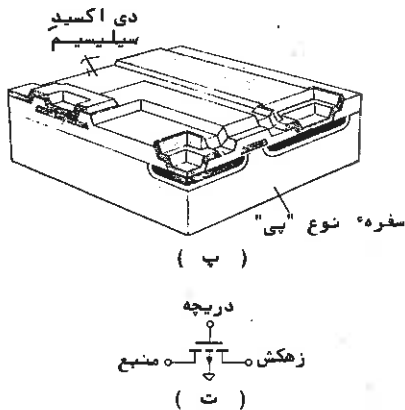
در شکل ۱۱- پ، ضخامت لایه‌های سطحی، برای روشن تر شدن موضوع، با اغراق نشان داده شده است. مثلاً، ضخامت لایه آلومی نیمی فقط نیم میکرون است و بطور کلی ساخت های نوع "ماس" بسیار کوچک و نازک هستند.



(الف)



(ب)



(پ)

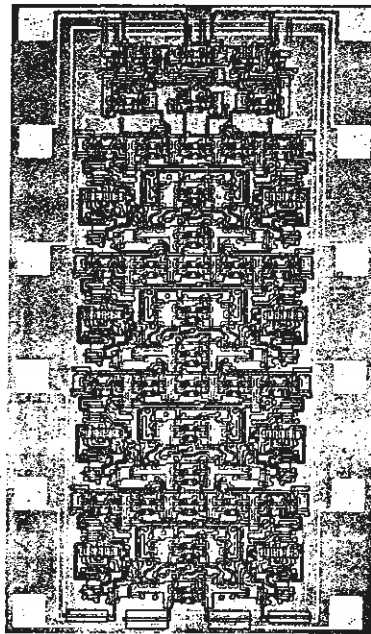
شکل ۱۱ - ترانزیستور با دریچه سیلیسیمی و مجرای نوع "ان" (الف) سطوح هندسی، (ب) طرح بندی مستطبی، (پ) ساخت و (ت) علامت نمایی که در آن وجود پیکان اختیاری است.

در این مقیاس های بسیار کوچک که مدارها ایداً قابل اطمینان نیستند، کوچولوی شیشه ای ما با مجراها و مسیرهای آلومی نیمی که بر روی بلندی ها و فرورفتگی های آن کشیده شده اند، روی یک لایه شکنده از بلور، میدان نبرد الکترون ها و حفره ها را تشکیل میدهد. آیا کسی دست به ساختن چنین چیزی خواهد زد؟

یادداشت ها (قسمت دوم)

4. MOS : Metal-Oxide Semiconductor
5. CCD : Charge-Coupled Device
6. ECL : Emitter-Coupled Logic
7. I²L : Integrated Injection Logic
8. Emitter; Base; Collector
9. Source; Drain

میشود. سپس قسمتهای دلخواه آن حذف شده و چنانکه بعداً ملاحظه خواهیم کرد، در مراحل مختلف ساخت با انواع ناخالصی ها بمنظور افزایش قابلیت هدایت، غنی میشود. برای مسیرهای هادی، آلومی نیم گماکان مورد استفاده قرار میگیرد ولی برخلاف گذشته که تنها دو لایه هادی در طرفین نیمه هادی برای برقرار ساختن اتصالات داخلی وجود داشت، امروزه معمولاً از سه لایه استفاده میشود تا علاوه بر حصول تراکم بیشتر، سرعت عمل مدار نیز بالا رود.



شکل ۱۰ - تراشه "ضرب کننده" لوله ای مضاعف از نوع "ماس".

شکل ۱۱- الف مراحل مختلف تشکیل سه لایه را نشان میدهد. لایه سیلیسیم چند بلوری توسط اکسید سیلیسیم از لایه فوقانی آلومی نیمی و لایه تحتانی جدا میشود و بنابراین میتواند برای ایجاد مسیرهای مستقلی که با لایه آلومی نیمی تماس ندارند مورد استفاده قرار گیرد.

چنین مسیرهایی در صورت تمایل طراح بر روی لایه تحتانی نیز قابل ایجاد است ولی هرگاه یک مسیر لایه تحتانی از زیر یک مسیر لایه چند بلوری عبور نماید (شکل ۱۱- ب)، یک ترانزیستور حاصل میشود! مشخصات این ترانزیستور بستگی به ضخامت لایه ها در محل تقاطع مسیرها دارد. در هر حال، طراح مدار قوانین مشخص و ساده ای را در زمینه مشخصات مستوی مسیرها دنبال میکند و مسائل مربوط به بعد سوم (ضخامت لایه ها) در مرحله ساخت توسط سازنده در نظر گرفته میشود.

شکل ۱۱- ب یک ترانزیستور متعسارف را نشان میدهد. پهنای دریچه در حدود ۶ میکرون است که با تکنولوژی سال ۱۹۸۰ (ماخذ ۶) به سادگی حاصل میشود. حال آنکه این مقدار تنها شش برابر طول موج نور قابل رویت است.

حال اگر مطابق شکل ۹ - ب ولتاژی حداقل معادل ولتاژ لازم برای بوجود آوردن یک "اتصال القاء شده توسط میدان" (در حدود یک ولت) بین دریچه و سطره ایجاد کنیم، منبع و زهکش از راه لایه وارون شده حامل به یکدیگر متصل میشوند. باین صورت الکترون ها از طریق مجرای نوع "ان" از منبع به سطره زهکش حرکت خواهند کرد. اینکه کدامیک از دو منطقه نوع "ان" را منبع و کدام را زهکش بدانیم به جهت جریان بستگی دارد.

مجرای بوجود آمده بین منبع و زهکش قابلیت هدایت کمی دارد چون اولاً سطح مقطع آن بسیار کوچک است و ثانیاً درجه ناخالصی آن از نظر حامل های بار خیلی پائین است. بعلاوه، بر اثر عبور الکترون ها از مجرا، حرارت زیادی که از برخورد الکترون ها با اتم های ثابت ناشی میشود بوجود می آید. الکترون های تضعیف شده (انرژی از دست داده) نهایتاً از طرق دیگری به منبع تغذیه باز میگردند.

۱/۳ - مجتمع کردن عناصر "ماس"

در سال ۱۹۵۲، یک مهندس انگلیسی بنام "چی کپلیو ای دایر" پیشگویی حیرت انگیز زیر را ارائه نمود (ماخذ ۵):

با پیدایش ترانزیستور و بطور کلی نیمه هادی ها، اکنون میتوان مدارهای الکترونیکی را بصورت گنجه های یک پارچه و بدون سیم های رابط جستم نمود. این گنجه ها از لایه های عایق، هادی، یکسوز و تقویت کننده تشکیل شده و قابلیت های مورد نظر مستقیماً با بریدن شکل های لازم بر روی این لایه ها بوجود آمده و بهم متصل می شوند.

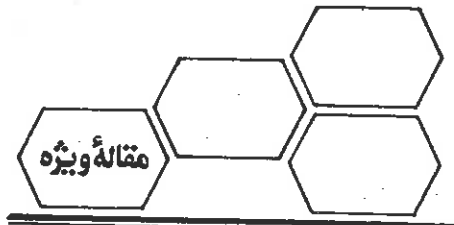
پس از گذشت چند سال از این پیشگویی، فعالیت در راه یافتن این نوع مولفه ها در انگلستان و ایالات متحده آمریکا آغاز شد. خود دایر در سال ۱۹۵۲ دستگاهی را بنمایش گذاشت که چگونگی استفاده از سیلیسیم را در مدارهای مجتمع نشان میداد. ولی اولین مداری یک قطبی عملی یک سال بعد توسط "چی اس کیلی" ارائه شد.

مدارهای "ماس" امروزی با روشی که توسط "آر اس نویس" ابداع گردید ساخته میشوند. شکل ۱۰ نمونه ای از این مدارها را نشان میدهد که بر روی یک سنجاق جا گرفته و مربوط به یک مدار ضرب کننده چند صد ترانزیستوری است.

۱/۳/۱ - ساخت فیزیکی عناصر "ماس"

امروزه در صنعت ساخت ترانزیستورهای یک قطبی، ترانزیستور با مجرای نوع "ان" و دریچه سیلیسیمی ترجیح داده میشود. نام "ماس" مربوط به دورانی است که دریچه های فلزی متداول بودند.

سیلیسیم چند بلوری روی سطح اکسیدی که سطره را میپوشاند پخش شده و با رسوب داده



اعداد بزرگ در کامپیوتر و ریاضیات

دکتر بهروز پرهامی
دانشیار دانشگاه صنعتی شریف

توضیح ویرایشگر: مقاله‌ای که در زیربنظرتان میرسد متن بسط یافته پنجمین سخنرانی ماهیانه انجمن انفورماتیک ایران است که روز چهارشنبه ۳۰ اردیبهشت ماه ۱۳۶۰ در محل دانشگاه صنعتی شریف توسط نگارنده ایراد گردید. این مقاله همچنین روز دوشنبه ۲۱ اردیبهشت ماه ۱۳۶۰ در سمینار "فلسفه، منطق و روش‌شناسی ریاضی" در محل دانشکده علوم دانشگاه تهران ارائه گردیده و قرار است جزو متون سخنرانی مربوط به مجموعه سمینارهای فوق‌الذکر، که توسط انجمن ریاضی ایران برگزار میشود، به چاپ برسد.

چکیده مقاله

ریاضی دانها گاه به اعداد متناهی با دیده حقارت می‌نگرند و آنها را "جالب توجه" بحساب نمی‌آورند. ولی برنامه‌نویسان میدانند که متناهی بودن اعداد نیست بلکه "خیلی محدود" بودن آنهاست. در این مقاله، ابتدا اعداد بزرگ در ریاضیات و روش نمایش آنها در کامپیوتر مورد بحث قرار می‌گیرد. ولی از آنجائیکه در علوم کامپیوتر، اعداد بزرگ نه بعنوان مقادیری که در محاسبات وارد میشوند بلکه عمدتاً "بمورد تعداد حالت‌های مختلف که در یک مسئله ترکیبی وجود دارد ظاهر میشوند، مطالب بعدی به کوشش‌هایی که بمنظور پیدا کردن راه‌هایی برای سروکله‌زدن با این نوع مسائل صورت گرفته است مربوط میشود. این کوشش‌ها، که یکی از جنبه‌های مهم رشته علوم و مهندسی کامپیوتر را تشکیل میدهند، به چهار دسته قابل تقسیم هستند: (۱) ازدیاد سرعت کامپیوترها تا حد امکان بمنظور بررسی تعداد حالت‌های بیشتری در یک زمان معین، (۲) یافتن الگوریتم بهتر برای حل یک مسئله ترکیبی بطوریکه بررسی تمام حالت‌های آن لازم نباشد، (۳) یافتن راه حل تقریبی یا آماری در مواردی که تعیین پاسخ دقیق مسئله خارج از محدوده امکانات موجود است یا دنبال آن رفتن با مبالغ "سرف نمیکنند" و (۴) اثبات اینکه برخی از مسائل درجه‌چار-چوب‌ر اماکنات موجود یا بطور کلی، راه حل عملی ندارند و راحت کردن خیال همه! این مقاله حاوی نکات و مثال‌هایی در مورد هر یک از موضوعات چهارگانه فوق‌بوده و در انتهای آن، مفهوم اعداد بزرگ "محاسبه ناپذیر" از طریق یک مثال ارائه شده است.

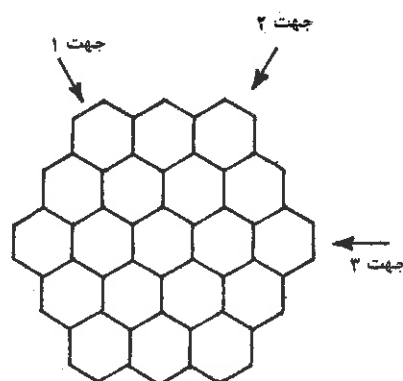
۱ - مقدمه: عدد بزرگ چیست؟

عبارت‌هایی که برای توصیف اعداد "بزرگتر از اندازه متعارف" بکار می‌روند متعدّدند. مثلاً، "اعداد بزرگ"، "اعداد خیلی بزرگ"، "اعداد غول‌آسا" و "اعداد نجومی". ولی مفهوم این عبارتها نزد افراد مختلف متفاوت است. عددی که بنظر یک دانش‌آموز دبستانی "خیلی بزرگ" می‌آید، برای یک ریاضیدان فوق‌العاده کوچک است. بنابراین این متوجه میشویم که بزرگی اعداد، مانند بسیاری از مفاهیم دیگر (مثلاً بلندی قد افراد، غلظت مایعات و گرمی هوا)، یک مفهوم نسبی یا آماری است؛ یعنی هرچه در مجموعه مشخصی از افراد، عده بیشتری یک عدد را "بزرگ" بدانند، آن عدد "بزرگتر" است. بر مبنای این تعریف، "صد" عدد بزرگی نیست، "یک میلیون" عدد نسبتاً بزرگی است و اعداد زیر "خیلی بزرگ" هستند (یعنی در مورد بزرگ بودن آنها، تقریباً همه متفق‌القولند):

$$\begin{aligned} 94\ 608\ 000\ 000\ 000\ 000\ 000\ 000 &= N_1 \\ 240\ 282\ 366\ 920\ 928\ 662 \\ 262\ 372\ 607\ 221\ 768\ 211\ 387 &= N_2 = 2^{128} + 1 \\ 121\ 625\ 100\ 408\ 822\ 000\ 000 &= N_3 = 19! \\ 42\ 282\ 002\ 272\ 489\ 886\ 000\ 000 &= N_4 \end{aligned}$$

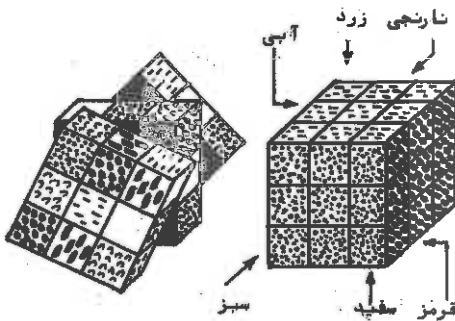
در تجربیات خود، ما بدو صورت متمایز با اعداد بزرگ‌سوارکار پیدا میکنیم: (الف) بعنوان مقادیری که کمیت‌های مادی را نشان داده یا در محاسبات وارد میشوند و (ب) بصورت تعداد حالت‌های مختلف که در یک مسئله ترکیبی وجود دارد.

بعنوان مثالی از اعداد بزرگ نوع (الف)، عدد N_1 در فوق فاصله دورترین کهکشان قابل "مشاهده" (حدود ۱۰ میلیارد سال نوری) را بر حسب کیلومتر بیان میکنند (اینکه چرا اعداد خیلی بزرگ را گاه "اعداد نجومی" می‌نامیم از این مثال معلوم میشود). همچنین برای تشخیص اول بودن یا نبودن عدد N_4 ، باید محاسباتی بر روی آن انجام داد.



شکل ۱ - اعداد ۱ تا ۱۹ را طوری در خانه‌های جدول فوق قرار دهید که مجموع اعداد هر ردیف در هر یک از سه جهت داده شده، روی شکل برابر شود.

بعنوان مثالی از اعداد بزرگ نوع (ب)، عدد N_4 در فوق تعداد صورت‌های مختلفی را که میتوان اعداد یک تا ۱۹ را در معنای شکل ۱ بمنظور پیدا کردن حل آن معما از راه‌برسی تمام حالت‌ها قرار داد نشان میدهد و عدد N_4 تعداد ترکیبات رنگی مختلفی است که در "مکتب جادویی" یا "مکتب روبیک" (شکل ۲) قابل ایجاد است.



شکل ۲ - "مکتب جادویی" که توسط "ارنو روبیک"، طراح و استاد رشته معماری اهل مجارستان، ابداع شده. از ۲۶ مکتب کوچک با سطوح رنگی (جمعاً ۶ رنگ مختلف) تشکیل شده و ساخت آن طوری است که هر یک از ۶ وجه مکتب بزرگ میتوانند حول مرکز خود بمقدار دلخواه دوران نمایند. هدف در این معما اینست که مکتب در هم ریخته را با چرخاندن وجوه آن بوضع اولیه برگردانیم بطوریکه هر یک از ۶ وجه آن تماماً "یک رنگ" باشد.

۲ - اعداد بزرگ در ریاضیات

ریاضیدانها گاه به اعداد متناهی، هر قدر هم که بزرگ باشند، با دیده حقارت می‌نگرند و آنها را "جالب توجه" بحساب نمی‌آورند. دلیل این امر اینست که در ریاضیات بسادگی میتوان اعدادی را که بزرگی شان مافوق تصور است نمایش داد. مثلاً در نمایش عدد $2^{128} + 1$ ، تنها شش علامت مورد استفاده قرار گرفته است و بهمان سادگی هم میتوانیم عدد 99999 را نمایش دهیم؛ عددی که حتی تعداد ارقامش بسادگی قابل تعیین نیست. تازه اکثریت قریب با اتفاق اعداد متناهی از این هم بزرگترند!

هنگامیکه اعداد از حدود درک و قدرت تصور ما بزرگتر شوند، دیگر مقدارشان و یا اختلاف بین آنها برای ما مفهوم نیست. مثلاً مدت زیادی طول میکشد تا تشخیص دهیم کدامیک از دو عدد زیر بزرگتر است، در حالیکه اختلاف بین آنها فوق‌العاده زیاد است:

$$\begin{aligned} 45678 \\ 87654 \end{aligned}$$

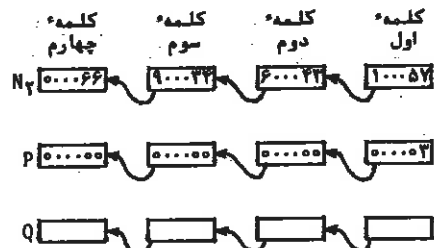
برای اینکه به اعداد خیلی بزرگتری دست بیابیم، از یک روش نشان‌گذاری ویژه استفاده میکنیم. اگر فرض کنیم که $x + y$ بمعنی "x بتواند y باشد" داریم:

$$x + y = x(x(x(x \dots (x) \dots)) \dots)$$

استفاده از علامت پیکان را میتوان بصورت زیر تعمیم داد:

ورق بزنید

داریم و چون حدود مقادیر p و q را نمی‌دانیم، همین مقدار فضا را هم باید به آنها اختصاص دهیم (شکل ۵).



شکل ۵ - نمایش اعداد صحیح بزرگ در چند کلمه کامپیوتر.

برای انجام عملیات بر روی اعدادی که باین ترتیب نمایش داده شده‌اند، باید برنامه‌های ویژه‌ای نوشت که جایی دستورالعمل‌های ماشین را بگیرند. باین ترتیب مثلاً عمل تقسیمی که بصورت ساده توسط ماشین ظرف یک میلیونیم ثانیه انجام می‌شد حالا دیگر احتمالاً چندین ده برابر این مقدار وقت خواهد گرفت. بجز جریمه زمانی که باین ترتیب می‌پردازیم و ظرفیت حافظه کامپیوتر، عملاً محدودیتی برای بزرگی اعدادی که باین ترتیب نمایش می‌دهیم وجود ندارد.

بفرض اینکه برنامه‌ای برای تعبیه قابلیت تقسیم N_2 بر p داشته باشیم، از نظر ریاضی راو حل مسئله روشن است. کافی است بخش پذیری N_2 را متوالیاً بر ۲، ۳، ۵، ۷، ... تحقیق کنیم (بعدها) خواهیم دید چرا این راو حل عملی نیست).

۲ - روشهای مقابله با محدودیت‌ها

با توجه به مثال فوق، بنظر می‌رسد که کار کردن با اعداد صحیح بزرگ در کامپیوتر ممکن است به دو نوع محدودیت بر بخورد: محدودیت در فضای حافظه و محدودیت در زمان یا سرعت محاسبه. محدودیت اول در عمل کمتر تولید اشکال میکند ولی محدودیت دوم یک مشکل واقعی است. برای تشریح موضوع، بد نیست ابتدا نگاهی به سرعت‌های محاسباتی در جدول ۲ بیاندازیم.

محاسبه کننده	تعداد عملیات در ثانیه
انسان	۵/۱
انسان بکمک چرتکه	۱
ماشین حساب مکانیکی	۱۰
کامپیوتر معمولی با سرعت متوسط	۱۰ ^۶
کامپیوتر سریع (ویژه محاسبات)	۱۰ ^۹

حال اگر مسئله طرح شده در بخش قبلی را در نظر بگیریم، با توجه باینکه مقسداً p تقریباً مساوی $10^{16} \times 16$ است، پیدا کردن

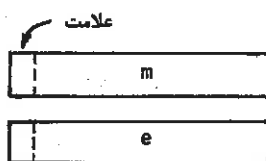
بنابیه قرارداد، کلمه شکل ۳ عدد $m.b^e$ را نشان میدهد که در آن b یک پایه قراردادی است. باین صورت، هرچه دقت مورد نیاز (تعداد ارقام مانعیت) کمتر باشد، حوزه وسیع تری از اعداد را میتوان با یک طول کلمه مشخص نمایش داد.

جدول ۱ ظرفیت تقریبی کامپیوترها را برای نمایش اعداد دقیق و نادقیق نشان میدهد. از آنجائیکه تعداد کل فرآیندهای موجود در قسمتهای شناخته شده از جهان هستی از 10^{100} متجاوز نیست و ثابتها و مقادیر فیزیکی نیز بمراتب از این مقدار کوچکترند، فقط در مورد کامپیوترهای کوچک ممکن است مشکل داشته باشیم که آنها با اختصاص دو کلمه برای نمایش هر عدد (بصورت شکل ۴) و برنامه نویسی متناسب با آن، قابل حل است. اگر فرض کنیم m و e دو عدد صحیح ۱۵ بیتی هستند و داشته باشیم $b = 16$ ، بزرگترین عددی که با این روش قابل نمایش است عبارت خواهد بود از:

$$N_2 = (2^{15} - 1)(16^{15} - 1) > 10^{29000}$$

جدول ۱ - ظرفیت تقریبی نمایش اعداد در کامپیوتر.

نوع کامپیوتر	طول تقریبی کلمه (به بیت)	اعداد با ممیز ثابت	اعداد با ممیز شناور
کامپیوتر کوچک و خیلی کوچک	۱۶	۱۰ ^۵	۱۰ ^{۴۰}
کامپیوتر معمولی (همه منظوره)	۳۲	۱۰ ^{۱۰}	۱۰ ^{۲۰۰}
کامپیوتر بزرگ (ویژه محاسبات)	۶۴	۱۰ ^{۲۰}	۱۰ ^{۱۰۰۰}



شکل ۴ - نمایش یک عدد "نادقیق" در دو کلمه کامپیوتر. توسط اعداد صحیح.

برای نمایش اعداد صحیح خیلی بزرگ در کامپیوتر نیز میتوان از روش مشابهی استفاده نمود، باین ترتیب که هر عدد بصورت برداری از اعداد کوچکتر در نظر گرفته شود. مثلاً فرض کنید که هدف ما تشخیص اول بودن یا نبودن عدد $N_2 = 2^{128} + 1$ است. بعبارت دیگر، میخواهیم کوچکترین مقدار p را پیدا کنیم بطوریکه $N_2 = p \times q$ باشد (طبیعی است که اگر $N_2 = p$ باشد، N_2 عددی اول خواهد بود).

اگر کامپیوتر مورد استفاده ما از نوع کامپیوترهای معمولی همه منظوره (جدول ۱) باشد، احتمالاً در آن فقط میتوانیم اعداد حداکثر ۱۰ رقمی را مستقیماً نشان دهیم. بنابراین برای نمایش N_2 که یک عدد ۲۹ رقمی است، به چهار کلمه کامپیوتر نیاز

$$x + y = \overbrace{x + (x + (x + \dots + (x) \dots))}^{\text{مترتبه } y}$$

$$x + y = \overbrace{x + (x + (x + \dots + (x) \dots))}^{\text{مترتبه } y}$$

$$x + y = \overbrace{x + (x + (x + \dots + (x) \dots))}^{\text{مترتبه } y}$$

$$x + y = \overbrace{x + (x + (x + \dots + (x) \dots))}^{\text{مترتبه } y}$$

در اینصورت، برای کوچکترین مقدار x و y که نتایج حاصله غیر بدیهی باشند (چون برای تمام مقادیر k همواره $p_{2,2}(k) = 4$) داریم:

$$p_{2,2}(1) = 2 + 2 = 4$$

$$p_{2,2}(2) = 2 + 2 = 2 + (2 + 2) = 2 + 4 = 6$$

$$p_{2,2}(3) = 2 + 2 = 2 + (2 + 2) = 2 + 4 = 6$$

$$p_{2,2}(4) = 2 + 2 = 2 + (2 + 2) = 2 + 4 = 6$$

$$N_5 = p_{2,2}(4) = 2 + 2 = 2 + (2 + 2) = 2 + 4 = 6$$

$$N_5 = p_{2,2}(4) = 2 + 2 = 2 + (2 + 2) = 2 + 4 = 6$$

$$N_5 = p_{2,2}(4) = 2 + 2 = 2 + (2 + 2) = 2 + 4 = 6$$

$$N_5 = p_{2,2}(4) = 2 + 2 = 2 + (2 + 2) = 2 + 4 = 6$$

$$N_6 = p_{2,2}(5) = 2 + 2 = 2 + (2 + 2) = 2 + 4 = 6$$

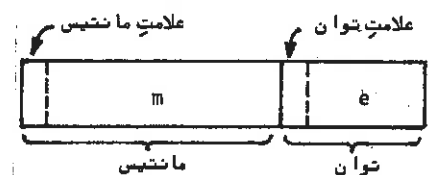
$$N_6 = p_{2,2}(5) = 2 + 2 = 2 + (2 + 2) = 2 + 4 = 6$$

$$N_6 = p_{2,2}(5) = 2 + 2 = 2 + (2 + 2) = 2 + 4 = 6$$

عدد N_6 بقدری بزرگ است که آنرا بدون توسل به روش نشان گذاری فوق نمیتوان حتی نمایش داد. تازه این عدد در مقابل $p_{2,2}(100)$ و $p_{2,2}(100)$ و $p_{2,2}(100)$ چیزی نیست! اگر توجه کنیم که تابع $p_{2,2}$ با این سرعت رشد خارق العاده، یک تابع "محاسبه پذیر" است و توابع "محاسبه ناپذیری" را در نظر بگیریم که سرعت رشد آنها از هر تابع محاسبه پذیری بیشتر است (بعدها نمونه‌ای از این نوع توابع را ارائه خواهیم کرد)، بخوبی روشن میشود که اعداد مثناهی در ریاضیات میتوانند فوق العاده بزرگ باشند.

۳ - نمایش اعداد بزرگ در کامپیوتر

حوزه نمایش اعداد صحیح در کامپیوتر عموماً بسیار محدود است و بندرت از 10^{20} تجاوز میکند. این امر در اکثر کاربردها مشکلی تولید نمیکند چون بندرت اتفاق می افتد که بخواهیم بر روی اعداد بزرگتر از این محاسبات "دقیق" انجام دهیم. اعداد بزرگتر در عمل بصورت "نادقیق" یا "بامیز شناور" نشان داده میشوند. باین ترتیب که قسمتی از یک کلمه کامپیوتر به مانعیت m و قسمتی دیگری به توان e اختصاص داده میشود.



شکل ۳ - نمایش اعداد نادقیق (با ممیز شناور) در کامپیوتر.

دهد. سپس بهمین ترتیب محلّهای ممکن برای یک ها، ۲ها، ... و ۹ها را بدست می آورد. و بالاخره جواب نهائی را با ترکیب کردن این جواب های جزئی پیدا میکند. باین ترتیب که مثلاً یکی از ۱۰۰ ترکیب ممکن از صفرها را انتخاب میکند. سپس از باین ترکیب های ممکن برای یک ها، ۲ها، ... و ۹ها با ترکیب انتخاب شده سازگار بودند برمیگزید و اینکار را برای ۲ها، ۳ها، ... ادامه میدهد تا با به جواب برسد و با اینکه به بن بست بر بخورد.

تخمین زده میشود که روش پارکر برای پیدا کردن مرتب های لاتین متعادل از مرتبه ۱۰ در حدود ۱۰^{۱۱} بار سریع تر از روش قبلی مربوط به سال ۱۹۵۲ است. با توجه باین نکته که طی مدت زمان فوق، کامپیوترها فقط حدود ۱۰ برابر سریع تر شدند، اهمیت یافتن راولر بهتر (در مقایسه با روش افزایش سرعت کامپیوترها) بخوبی روشن میشود.

اکنون دیگر معلوم شده است که مرتب های لاتین متعادل برای تمام مرتبه های بالاتر از ۶ وجود دارند. البته کامپیوتر در کشف این موضوع نقشی نداشته است چون با بالا رفتن مرتبه، میزان زمان لازم برای یافتن زوج متعادل سریعاً بالا میرود بطوریکه برای مرتبه ۱۵، حتی روش پارکریز نمیتواند ما را در یافتن زوج متعادل یاری نماید. کشف موضوع فوق بصورت دستی و با تعمیم آگوهائی که در مرتبه های کوچکتر مشاهده شده صورت گرفته است.

۶- راولرهای تقریبی یا آماری

در توری ۱۰x۱۰ شکل ۷، چند مسیر بدون حلقه از گوشه پائینی سمت چپ به گوشه بالائی سمت راست وجود دارد؟ مسأله ای از این قبیل مورد توجه شیمی دانها و فیزیک دانهای است که ساختمان ملکول های بزرگ را مطالعه میکنند. برای شروع به حلّ مسئله، توری های ۲x۲ و ۳x۳ را در نظر میگیریم. در این دو مورد، تعداد مسیرها با کمی فکر بترتیب مساوی ۱۲ و ۱۶ بدست می آید. ولی وقتی به توری ۴x۴ میرسیم، تعداد مسیرها بسیار زیاد میشود بطوریکه پیدا کردن و شمارش همه آنها با دست خیلی وقت گیر است. از آنجائیکه برای تعیین تعداد دقیق این مسیرها در توری ۱۰x۱۰، هیچ روش عملی که در مدت زمان معقولی به جواب برسد وجود ندارد، باید سعی نمود که یک جواب تقریبی (که با احتمال نسبتاً زیادی نزدیک به مقدار صحیح باشد) بدست آورد.

برای این منظور برخی از مسیرهای ممکن را بطور تصادفی میسازیم: یعنی از گوشه پائینی سمت چپ شروع کرده و در هر مرحله که تعداد امکانات برای ادامه مسیر بیش از یک باشد، با استفاده از یک مولد اعداد تصادفی، یکی از دو یا سه راه ممکن برای

۵- یافتن راولر بهتر

اهمیت این روش را با یک مثال جالب در مورد "مرتب های لاتین" نشان میدهم. مرتب لاتین از مرتبه n جدولی n در n از اعداد صفر تا n-۱ است که در آن هیچ سطری استونی حاوی اعداد تکراری نباشد. این مرتب ها، برای یک مرتبه خاص، منحصر بفرد نیستند. مثلاً تعداد مرتب های لاتین از مرتبه ۱۰ بیش از ۱۰^{۲۶} تخمین زده میشود.

شکل ۶ یک زوج مرتب لاتین از مرتبه ۱۰ را نشان میدهد. این زوج مرتب ها اصطلاحاً "متعادل" مینامیم چون در بین اعداد دورقی که از پهلوی هم گذاشتن آنها حاصل میشود (جمعا ص عدد) هیچ عدد تکراری وجود ندارد. مرتب های لاتین و زوج های متعادل در موارد مختلفی از قبیل طرح آزمایش های آماری، برنامه ریزی گردش کشت در کشاورزی و طرح حافظه کامپیوتر "با قابلیت تحمل خرابی" کاربرد دارند.

00	12	28	35	49	54	67	73	86	91
17	87	34	29	33	46	75	60	92	08
22	95	56	64	38	07	80	41	79	13
33	76	09	90	84	65	18	52	21	47
44	68	71	57	25	93	06	89	10	32
55	07	97	48	70	82	39	14	63	26
66	59	40	72	17	31	23	98	04	85
77	43	15	81	02	20	94	36	58	69
88	30	62	03	96	19	51	27	45	74
99	24	83	16	61	78	42	05	37	50

شکل ۶ - یک زوج مرتب لاتین از مرتبه ۱۰ که بر روی هم منطبق شده اند (ارقام مربوط به یکی از مرتب ها با حروف کج نشان داده شده اند).

اوایلر^۴، ریاضیدان بزرگ قرن هجدهم، روش ساخت زوج های متعادل از مرتب های لاتین را (جز برای مرتبه هائی بصورت کلی ۲k+۲) ارائه نموده و چنین گمان میکرد که اصولاً زوج های متعادل برای این مرتبه ها وجود ندارد. تحقیق این موضوع در مورد مرتبه ۲ کار ساده ای است و یک ریاضیدان فرانسوی^۵ در سال ۱۹۰۰ محنت این گمان را برای مرتبه ۶ از راه یک تحلیل جامع (در نظر گرفتن تمام حالت ها) نشان داده است.

در سال ۱۹۵۲، چند ریاضیدان سعی کردند که با کمک یک برنامه کامپیوتری مسئله را در مورد مرتبه ۱۰ مورد بررسی قرار دهند. آنها یک مرتب لاتین دلخواه را انتخاب نموده و سعی کردند که زوج متعادل آنها بیابند ولی محاسبات ماشین آنقدر طول کشید که مجبور به قطع آن شدند. تازه اگر هم محاسبه پس از چندین صد سال تمام میشد و جوابی تولید نمیکرد، هنوز میبایست همین کار را برای هریک از حدوداً ۱۰^{۲۶} مرتب لاتین دیگر از مرتبه ۱۰ انجام میدادند.

در سال ۱۹۵۹، پارکر^۶ توانست با استفاده از الگوریتم بهتری مسئله را ظرف یک دقیقه روی یک کامپیوتر نسبتاً "کند حل کند". روش او باین ترتیب بود که ابتدا سعی میکرد صفرهای مرتب متعادل را در جای خود قرار

آن از این طریق با کمک سریع ترین کامپیوترها به حدود ۵۰ سال وقت نیاز دارد (فرض شده است که هریک از تقریباً ۱۰^{۱۶} x عمل جمع و تقسیم چهار رقمی معادل ۵۰ عمل جمع ۱۰ رقمی وقت لازم داشته باشد).

بهمین ترتیب محلّی معما های ارائه شده در شکل های ۱ و ۲ با کمک کامپیوتر (در صورتیکه راولر^۷ بکار رفته مبتنی بر بررسی کلیه ترکیبها تمکن باشد و با این فرض که بررسی هریک از امکانات معادل ۵۰ عمل جمع وقت لازم داشته باشد) بترتیب ۲۰۰ و ۷۰۰۰ سال طول میکشد. همین وضع در بسیاری از مسائل ترکیبی با تعداد حالت های خیلی زیاد، که در کاربردهای واقعی علمی و تجاری کامپیوترها بوفور یافت میشوند، وجود دارد.

مسئله است که وضع فوق قابل قبول نبوده و باید برای آن چاره جویی نمود. رشته علوم و مهندسی کامپیوتر برای حلّ این مشکل چهار مسیر عمده دارد:

۱- سرعت بیشتر: سعی در افزایش سرعت کامپیوترها تا حد امکان بمنظور بررسی نمودن تعداد حالت های بیشتری در یک زمان معین.

۲- یافتن راولر بهتر: سعی در یافتن الگوریتم (راولر) بهتر برای حلّ یک مسئله ترکیبی، بطوریکه بررسی تمام حالت های آن لازم نباشد.

۳- راولر تقریبی یا آماری: سعی در یافتن راولر حلّ تقریبی یا آماری در مواردی که تعیین پاسخ دقیق مسئله خارج از محدوده امکانات موجود است و یا دنبال آن رفتن با اصطلاح "مرف نمیکند".

۴- اثبات پیچیدگی ذاتی: اثبات اینکه برخی از مسائل در چهارچوب امکانات موجود راولر عملی ندارند (یا اینکه اصولاً "غیر قابل حلّ هستند") و راحت کردن خیال همه!

در مورد اول محبتی نخواهیم کرد جز اینکه بگوئیم در این زمینه دو روش وجود دارد:

الف- روش افزایش سرعت در مدارهای کامپیوتری.

ب- روش تطبیق سازمان کامپیوتر با ساخت مسئله بمنظور افزایش کارآیی سیستم.

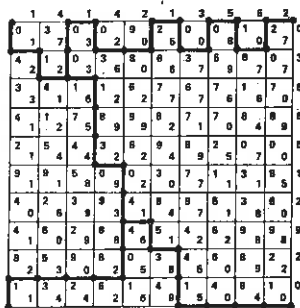
با وجود اینکه در طول سی و چند سال عملاً کامپیوترهای الکترونیکی، سرعت آنها بر اثر افزایش سرعت مدارها و پیشرفت رشته سازمان کامپیوتر چندین ده هزار برابر شده است، چنانکه بعداً خواهیم دید این ازدیاد سرعت بتنهایی نمیتواند پاسخگوی نیازهای محاسباتی ما باشد.

در دنباله مقاله، موارد دوم، سوم و چهارم را با ذکر مثال های لازم بررسی خواهیم کرد.

مسیر (که در مورد این مثال بخصوص منحصر-بفرد است)، از نقطه مقصد شروع کرده و در هر مرحله به یکی از نقاط مجاور میروسیم بطوریکه فاصله حداقل آن نقطه از مبدا به اندازه طول خط طی شده کمتر از فاصله نقطه قبلی از مبدا باشد.

مسئله دیگری که میتواند در رابطه با توری شکل ۸ مطرح شود، پیدا کردن کوتاهترین شبکه برای وصل کردن هر چهار گوشه توری است. چنین مسئلهای در طراحی مدارهای چاپی پیش می آید. شکل ۹ شبکه مورد نظر را، که "درخت استاینر" نام دارد، نشان میدهد.

تعداد درختهای استاینر در این توری برماتبیش از تعداد مسیرهای مورد بحث در بخش قبلی است. با اینحال یک الگوریتم "خوب"، که اجرایش بر روی کامپیوتر بیش از چند ثانیه طول نمیکشد، بما این اجازه را میدهد که از صحت پاسخ ارائه شده در شکل ۹ مطمئن باشیم. مقصود از الگوریتم "خوب" در اینجا الگوریتمی است که زمان لازم برای اجرای آن با افزایش اندازه مسئله بصورت بیرویه بالا نرود...



شکل ۹ - کوتاهترین راه برای وصل کردن هر چهار گوشه (درخت استاینر) در مورد توری شکل ۸.

الگوریتم فوق الذکر برای وصل کردن حداقل پنج نقطه در یک شبکه n نقطه ای تقریباً به n^4 واحد زمان (قدم) نیاز دارد. ولی هرگاه تعداد نقاطی که باید بهم وصل شوند افزایش یابد (مثلاً به حدود یک دهم مقدار n برسد)، هیچ الگوریتم خوبی برای پیدا کردن بهترین درخت استاینر موجود نیست.

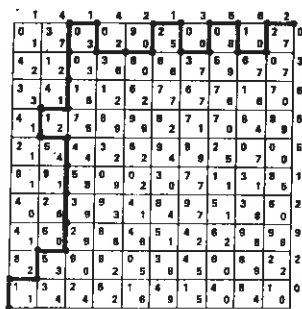
با شرح فوق، شاید از این واقعیت تعجب کنید که هرگاه متصل کردن تمام نقاط در توری شکل ۸ مورد نظر باشد، روش کار فوق العاده ساده میشود. در مورد این توری، تعداد شبکه های متصل کننده تمام نقاط، کسبه "درخت های پوشا" نامیده میشوند، واقعا زیاد است (تقریباً 4×10^{52}). ولی الگوریتمی که توسط کروسکال^{۱۰} کشف شده بما این اجازه را میدهد که حتی بصورت دستی ظرف چند دقیقه پاسخ مورد نظر را بیابیم.

روش کار با این ترتیب است که خطوط را برترتیب از کوتاهترین تا بلندترین آنها

باید در نظر داشت که حتی این نتایج بسیار مبنای کسر کوچکی از کل مسیرها بدست آمده اند و بنابراین نمیتوانیم از صحت آنها کاملاً مطمئن باشیم.

۷ - چند مثال جالب دیگر

اگر به هریک از خطوط در توری 10×10 شکل ۷، وزن دلخواهی (مثلاً یک رقم تصادفی) اضافه کنیم، یک توری نظیر شکل ۸ بدست می آید. البته شاید خواننده دقیقاً نتواند بتواند ارقام جذر عدد ۲ را در ردیف بسالا و ارقام عدد "بی" را در ستون دوم از سمت چپ تشخیص دهد، ولی وجود این نظم بخصوص در مثال هایی که خواهیم آورد تاءثیری ندارد.



شکل ۸ - توری 10×10 با خطوط وزن دار (بر اساس بیست مقدار ثابت ریاضی) و کوتاهترین مسیر روی آن از گوشه پایینی سمت چپ به گوشه بالائی سمت راست.

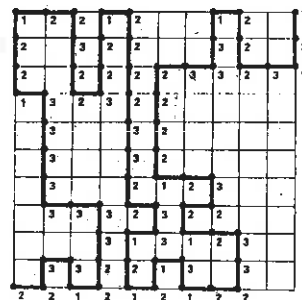
اگر وزنهای خطوط را بمناباه فاصله نقاط (طول جاده) بین دو نقطه یا زمان طی کردن آن در نظر بگیریم، یکی از مسائلی که میتواند در مورد شکل ۸ مطرح شود پیدا کردن کوتاهترین مسیر از گوشه پایینی سمت چپ توری به گوشه بالائی سمت راست آن است. طبیعی است که به علت زیاد بودن تعداد کل مسیرهای ممکن (مطابق آنچه در بخش قبلی دیدیم)، نمیتوان جواب مسئله را با بررسی و مقایسه طول یک یک مسیرها بدست آورد. خوشبختانه در مورد چنین مسائلی، دایکسترا^۷ الگوریتم ساده ای را ارائه داده است که حتی بصورت دستی با سهولت قابل اجراست.

الگوریتم برای این اساس کار میکند که هرگاه در یک مرحله، تمام نقاط توری که به فاصله مثلاً ۲۰ یا کمتر از نقطه شروع قرار دارند در دست باشند، بسادگی میتوان با در نظر گرفتن خطوط مجاور این نقاط، تمام نقاط توری را که به فاصله ۲۱ از نقطه شروع واقع شده اند پیدا کرد. با ادامه این عمل، پس از مدتی فاصله حداقل از نقطه شروع برای تمام نقاط توری و از جمله نقطه مقصد مورد نظر پیدا خواهد شد.

اگر این کار را در مورد شکل ۸ انجام دهیم، طول حداقل مسیر مورد نظر مساوی ۲۳ بدست می آید. برای پیدا کردن کوتاهترین

ادامه مسیر را انتخاب میکنیم. یک مسیر تصادفی از این نوع در شکل ۷ نشان داده شده است. احتمال تولید این مسیر از ضرب کردن معکوس تعداد انتخاب ها در هر مرحله بدست می آید:

$$p_1 = 2 \times 10^{-22} \approx 2 \times 10^{-22}$$



شکل ۷ - توری 10×10 و یک مسیر بدون حلقه تصادفی از گوشه پایینی سمت چپ به گوشه بالائی سمت راست. در هر نقطه از مسیر، تعداد امکانات (انتخاب ها) برای ادامه مسیر نشان داده شده است.

با تکرار این عمل برای چهاربار دیگر، مسیرهایی بدست آمده است که مشخصات آنها در جدول ۲ دیده میشود. حال اگر بخواهیم بر اساس داده های جدول ۲، تعداد کل مسیرها را تخمین بزنیم، یک قضیه آماربما میگوید که بهترین تخمین، از مقدار متوسط معکوس احتمالات بدست می آید.

$$22 \times 10^2 \approx \text{تعداد تخمینی مسیرها}$$

جدول ۳ - مشخصات ۵ مسیر تصادفی در توری 10×10 شکل ۷.

شماره مسیر	احتمال تولید این مسیر بخصوص	طول مسیر
۱	2×10^{-22}	۷۰
۲	3×10^{-12}	۳۰
۳	10^{-11}	۲۸
۴	7×10^{-11}	۴۰
۵	5×10^{-24}	۷۰

همچنین برای تخمین زدن میانگین طول مسیرها، باید طول هر مسیر را در معکوس احتمال مربوطه ضرب نمود و مجموع این مقادیر را بر مجموع معکوسات احتمال ها تقسیم کرد. با این ترتیب خواهیم داشت:

$70 \approx$ میانگین تخمینی طول مسیرها البته همانطور که میدانید، نمیتوان به نتایج حاصل بر مبنای فقط ۵ آزمایش اطمینان نمود. ولی در صورتیکه تولید مسیرهای تصادفی را توسط کامپیوتر انجام دهیم، میتوان چندین هزار مسیر را بسادگی بررسی نموده و نتایج دقیق تری بدست آورد. نتایج زیر که با این صورت بدست آمده اند، با درجه اطمینان زیادی صحیح هستند:

$$24 \times 10^2 (1/6 \pm 3/4) \approx \text{تعداد تخمینی مسیرها}$$

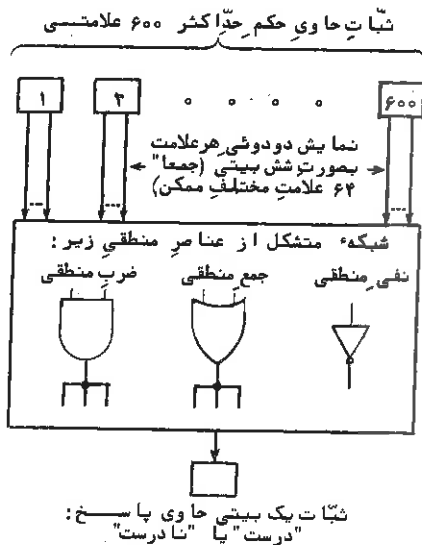
$$94 \pm 5 \approx \text{میانگین تخمینی طول مسیرها}$$

باشد، میتوان نتیجه گرفت که عدد طبیعی y وجود دارد بطوریکه y عضوی از S بوده و برای هر عدد طبیعی z در S بتوان گفت کسسه y کوچکتر یا مساوی z است. این حکم هم صحیح است چون در واقع میگوید که "در هر مجموعه غیر تهی از اعداد طبیعی، عنصری وجود دارد که از همه عناصر دیگر کوچکتر است".

حکم های منطقی از این نوع را در نظر میگیریم و قرار میگذاریم که تنها عمل جمع، محاسباتی قابل قبول در این حکمها، عمل جمع، آنهم فقط جمع کردن با یک مقدار ثابت (یعنی مثلاً استفاده از $x+y$ مجاز نیست)، باشد. تحقیق در مورد درستی یا نادرستی این احکام قاعدتا باید بسیار ساده باشد؛ مسلماً این کار به مشکلی اثبات قضایای ریاضی، که مرتباً توسط ریاضیدانان صورت میگیرد، نیست.

قضیه مشهوری از بوجی ۱۲ میگوید که میتوان در تعداد متناهی از قدمها، درستی یا نادرستی هر حکم از نوع فوق را تحقیق نمود. ولی دقت فوق الذکر ثابت کرده اند که اینکار بمرور عملی (توسط کامپیوتر) مقدور نیست، حتی اگر تعداد علامتهای بکار رفته در نمایش حکم را به حداکثر ۶۰۰ عدد محدود کنیم.

اثبات ارائه شده بر این اساس بنا شده که هرکاری را که توسط کامپیوتر قابل انجام باشد، میتوان توسط یک مدار منطقی متشکل از عناصر "نفی منطقی"، "جمع منطقی" و "ضرب منطقی" نیز انجام داد. فرض کنید که این مدار، مطابق شکل ۱۲، صورت دودویی شده حکم مورد نظر را دریافت کرده و پاسخ "درست" یا "نادرست" را در خروجی تولید نماید.



شکل ۱۲ - مدار منطقی فرضی برای تشخیص درستی یا نادرستی یک حکم منطقی (با حداکثر ۶۰۰ علامت) درباره اعداد طبیعی.

ورق بزنید

الف - حتی اگر ثابت شود که مسئله فروشنده دوره گرد "ذاتاً" مشکل است (یعنی الگوریتم خوبی برای حل آن وجود ندارد)، از نظر عملی فاجعه بزرگی رخ نداده است. الگوریتمهای تقریبی وجود دارند که مثلاً تحت شرایطی خاص، پاسخی را که حداکثر پنجاه درصد با بهترین پاسخ فرق دارد پیدا میکنند و چنین پاسخی در بسیاری از موارد عملی رضایت بخش است.

ب - حتی اگر الگوریتم خوبی برای حل این مسئله پیدا شود، ممکن است در بعضی از موارد، استفاده از آن "مصرف نکند": باین معنی که هزینه لازم برای اجرای الگوریتم (از نظر وقت کامپیوتر و میزان حافظه مصرفی) بیش از صرفه جویی حاصل از کاهش طول مسیر باشد.

۸ - اثبات پیچیدگی ذاتی

در سالهای اخیر، پژوهشگران رشته کامپیوتر موفق شده اند پیچیدگی ذاتی برخی از مسائل را اثبات نمایند. یکی از جالب ترین مثالها در این زمینه توسط دو نفر ۱۱ در دانشگاه "ام آی تی" در سال ۱۹۷۴ کشف شده است. مسئله مورد نظر، تشخیص درستی و نادرستی برخی از حکمهای منطقی در باره اعداد طبیعی است.

بد نیست مثال هایی از احکام مورد نظر را ارائه کنیم. حکم $1063 \leq 48$ مسلماً درست است. حال حکم زیر را در نظر میگیریم:

$$\forall n \exists m (m < n+1)$$

حکم فوق میگوید که "برای هر عدد طبیعی n ، عدد طبیعی m وجود دارد بطوریکه m کوچکتر از $n+1$ باشد". این حکم هم بوضوح درست است. ولی حکم زیر، که میگوید "برای هر عدد طبیعی n ، عدد طبیعی m وجود دارد بطوریکه $m+1$ کوچکتر از n باشد"، نادرست است:

$$\forall n \exists m (m+1 < n)$$

مثال بعدی کمی پیچیده تر است:

$$\forall x \forall y (y \geq x+2 \rightarrow \exists z (x < z \wedge z < y))$$

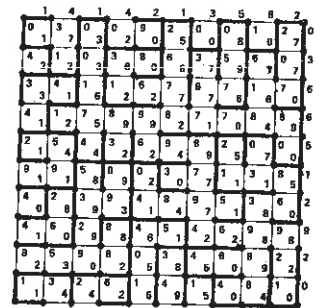
"برای هر زوج از اعداد طبیعی x و y ، هرگاه y بزرگتر یا مساوی $x+2$ باشد، میتوان نتیجه گرفت که عدد طبیعی z وجود دارد بطوریکه x کوچکتر از z و z کوچکتر از y باشد". این حکم هم صحیح است چون در واقع میگوید "هرگاه اختلاف دو عدد طبیعی حداقل مساوی ۲ باشد، یک عدد طبیعی بین آنها وجود دارد".

بالاخره، در حکم های مورد نظر، مجموعه های از اعداد طبیعی نیز میتوانند وارد شوند. مثلاً حکم

$$\forall S (\exists x (x \in S) \rightarrow \exists y (\exists z (y \in S \wedge z \in S \rightarrow y < z)))$$

میگوید که برای هر مجموعه S از اعداد طبیعی، اگر عدد طبیعی x در S وجود داشته

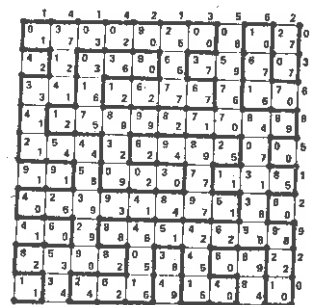
در نظر میگیریم (خطوط با طول مساوی را می توان بهر ترتیب دلخواهی در نظر گرفت). در هر مرحله از الگوریتم، خط مورد بررسی را در صورتی به شبکه مورد نظر اضافه می کنیم که اضافه کردن آن، به اتمام حداقل دو نقطه که توسط خطوط انتخاب شده قبلی بهم وصل نشده اند بیانجامد (شکل ۱۰).



شکل ۱۰ - کوتاه ترین راه برای وصل کردن تمام نقاط (درخت پوشا) در مورد توری شکل ۸.

سوال بعدی به مراتب مشکل تر است: کوتاه ترین مسیری که از گوشه پائینی سمت چپ به گوشه بالائی سمت راست توری رفته و در بین راه از تمام نقاط آن عبور میکند کدام است؟ این مسئله را "مسئله فروشنده دوره گرد" مینامند چون معادل مسئله پیکاروی فروشنده ای است که میخواهد به تمام شهرهای موجود در یک شبکه، با حداقل زمان یا هزینه مسافت سر بزند.

برای حل این مسئله، الگوریتمهای کشف شده اند که چندان "خوب" نیستند: باین معنی که زمان اجرای آنها با اندازه مسئله سریعاً افزایش می یابد، بطوریکه توری شکل ۸ بزرگترین شبکه ای است که پاسخ دقیق در مورد آن از این طریق بدست آمده است (شکل ۱۱). برای اجتناب از طولانی شدن بحث، از بررسی این الگوریتمها در اینجا خودداری میکنیم.



شکل ۱۱ - کوتاه ترین مسیری که از گوشه پائینی سمت چپ به گوشه بالائی سمت راست توری رفته و در بین راه از تمام نقاط عبور میکند.

این مثال آخر، زمینه خوبی را برای ذکر دو نکته فراهم ساخته است:

حال میتوان ماشین تورینگ دیگری را با
 $y+k+m+1$ حالت طوری طرح کرد که به تعداد
 $h(h(y))$ علامت X روی نوار خالی ثبت نماید.
 کافیت ابتدا با $y+k+1$ حالت، تعداد $h(y)$
 علامت X مطابق شرح فوق روی نوار ثبت کنیم
 و سپس با m حالت بقیه، مقدار تابع محاسبه
 پذیر h را پیدا کنیم. بنابراین میتوان
 نوشت:

$$f(y+k+m+1) \geq h(h(y)) \geq h(y^2)$$

برای مقادیر بزرگی y داریم

$$y^2 > y+k+m+1$$

بنابراین:

$$f(y+k+m+1) > h(y+k+m+1) \geq g(y+k+m+1)$$

برای y های بزرگ

پس ثابت میشود که:

$$f(z) > g(z)$$

برای z های بزرگ

چون $g(z)$ تابع محاسبه پذیر دلخواهی بود، f
 نمیتواند محاسبه پذیر باشد زیرا در این صورت
 از خودش بزرگتر خواهد شد.

۱۰ - نتیجه گیری

دیدیم که اعداد متناهی میتوانند فوق -
 العاده بزرگ شوند بطوریکه میزان بزرگی
 آنها از حدود درک و تصور ما خارج گردد
 (بخش دوم) یا اینکه اصولاً نتوانیم آنها را
 محاسبه کرده و یا حتی نمایش دهیم (بخش ششم).
 در مورد مسائل ترکیبی با تعداد حالت های
 خیلی زیاد، ملاحظه کردیم که ازدیاد سرعت
 کامپیوترها بحثهای نمیتواند پاسخگوی
 نیازهای محاسباتی ما باشد چون اولاً میزان
 ازدیاد سرعت ممکن، در مقابل اندازه برخی
 از این مسائل، ناچیز است و ثانیاً تجربه
 نشان داده است که با افزایش سرعت کامپیوتر -
 ها، مرتباً توقع ما نیز از آنها بالامیرود
 و مسائل بازم بزرگتری را برای حل توسط
 آنها مطرح میکنیم. در این قبیل موارد، کوشش
 در راه یافتن راول های بهتر (بخش های پنجم
 و هفتم)، ارائه راول های تقریبی یا آماری
 (بخش ششم) و یا اثبات اینکه برخی از این
 مسائل ذاتاً پیچیده هستند (بخش هشتم) ما
 را به مراتب زودتر به نتیجه میرساند.

یادداشت ها

1. Erno Rubik
2. Latin squares
3. Orthogonal
4. L. Euler
5. G. Tarry
6. E. T. Parker
7. E. W. Dijkstra
8. Steiner tree
9. Spanning trees
10. J. B. Kruskal
11. A. Meyer and L. J. Stockmeyer
12. J. R. Büchi
13. Turing machine

مآخذ

مثالهای بخش های دوم و پنجم تا هشتم این
 مقاله از مآخذ زیر گرفته شده اند:
 Knuth, D.E., "Mathematics and Computer
 Science: Coping with Finiteness," Science
 Vol. 194, No. 4271, 17 December 1976,
 pp. 1235-1242.

گرفته و بقیه نوار خالی است شروع بکار کنند،
 پس از توقف آن، تعداد $g(y)$ علامت X دنبال
 هم روی نوار موجود باشند.

چون تعداد ماشین های تورینگ n حالتی
 از نوع فوق متناهی است (زیرا تعداد کل
 توابع قابل تعریف Q و W و M بر حسب دو متغیر
 $\{B, X\}$ و $\{1, 2, \dots, n\}$ متناهی است)،
 در بین آنها ماشینی وجود دارد که در صورت
 شروع با یک نوار کاملاً خالی، متوقف شده و
 حداکثر تعداد ممکن علامت X را بر روی نوار
 بجا میگذارد (مسلماً است که اگر برخی از این
 ماشین ها هیچگاه با وضعیت شروع فوق متوقف
 نشوند، میتوانند تعدادی نامتناهی علامت X
 بر روی نوار ثبت کنند. این دسته از ماشین ها
 در اینجا مورد نظر نیستند).

فرض کنیم تعداد حداکثر فوق مساوی $f(n)$
 باشد. مقدار $f(n)$ برای مقادیر خیلی کوچک n
 بصورت دستی و یا توسط کامپیوتر قابل تعیین
 است. ولی هنگامیکه n بزرگ شود، میتوان ثابت
 کرد که مقدار $f(n)$ در نهایت از هر تابع
 محاسبه پذیری بزرگتر خواهد شد. چون $f(n)$
 نمیتواند از خودش بزرگتر باشد، نتیجه می -
 گیریم که f محاسبه ناپذیر است.

برای شروع اثبات، دو خاصیت زیر را

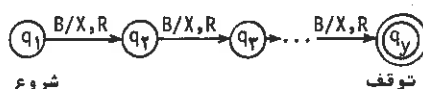
تابع f را در نظر میگیریم:

$$f(y) \geq y - 1$$

$$f(y) \leq f(y+1)$$

خاصیت اول از این واقعیت ناشی میشود که
 براحتی میتوان یک ماشین تورینگ y حالتی
 طرح نمود که پس از نوشتن $y-1$ علامت X بر
 روی نوار خالی متوقف شود (شکل ۱۳). خاصیت
 دوم هم واضح است چون افزایش تعداد حالتها
 نمیتواند منجر به کاهش حداکثر تعداد X قابل
 ثبت بر روی نوار شود.

راست است $S/S, R \rightarrow$ علامت خوانده شده.
 چپ حرکت نوک $\rightarrow$ علامت نوشته شده.



شکل ۱۳ - نمودار ماشین تورینگ y حالتی
 که پس از نوشتن $y-1$ علامت X بر روی
 نوار خالی متوقف میشود.

فرض کنیم $g(y)$ تابع محاسبه پذیر دلخواهی
 با خواص فوق الذکر باشد. میتوان با دگسی
 یک ماشین تورینگ با $y+k+1$ حالت طرح نمود
 بطوریکه پس از نوشتن $h(y) = g(y) + y^2$ علامت
 X بر روی نوار خالی متوقف شود: $y+1$ حالت
 برای نوشتن y علامت X بر روی نوار خالی
 بکار میرود و k حالت بقیه برای محاسبه
 تابع $g(y) + y^2$ ، که تابع محاسبه پذیری است.
 تابع $h(y)$ یک تابع صعودی است و داریم
 $h(y) \geq y^2$

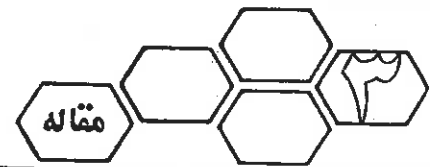
بر اساس قضیه بوجی، قاعدتاً باید
 بتوان چنین مداری را با استفاده از تعدادی
 متناهی از عناصر منطقی ساخت. ولی دو نفر
 فوق الذکر ثابت کرده اند که هر شبکه منطقی
 از نوع فوق باید دارای چیزی بیش از 10^{100}
 عنصر باشد. ولی در بخش سوم مقاله گفتیم
 که تعداد کل ذرات اتمی موجود در قسمتهای
 شناخته شده جهان هستی از 10^{100} تجاوز
 نمیکند. بنابراین مسلم است که ساخت چنین
 شبکه ای غیر ممکن خواهد بود.

۹ - اعداد محاسبه ناپذیر

آخرین مثالی که داریم از دو نظر بحث
 ما را کامل میکند: یکی از نظر نشان دادن
 این موضوع که اعداد خیلی بزرگ محاسبه نا -
 پذیر، که به مراتب از اعداد مورد بحث در
 بخش دوم مقاله بزرگترند، وجود دارند و دیگری
 از نظر آشنایی با مفهوم "محاسبه ناپذیری"
 بعنوان وضعیت جدی از مفهوم "پیچیدگی گسسی"
 محاسباتی که در بخش قبلی ارائه گردید.
 مثالی مورد بحث به "ماشین تورینگ" ۱۳
 مربوط میشود. خوانندگانی که با ماشین
 تورینگ آشنایی ندارند، کافیت نکات زیر را
 در مورد ماشین تورینگ خاصی که در اینجا مورد
 نظر است بدانند:

الف - ماشین تورینگ از سه قسمت تشکیل
 شده است: (۱) یک "نوار" متشکل از خانه های
 متوالی که در هر خانه آن یک علامت از الفبای
 دو علامتی $\{B, X\}$ قابل ثبت است. B نشان
 دهنده علامت "فاصله خالی" است. (۲) یک
 نوک خواندن و نوشتن که میتواند علامت ثبت
 شده در یک خانه از نوار را بخواند و به
 واحد کنترل اطلاع دهد و یا طبق دستور واحد
 کنترل، علامت X را در آن خانه بنویسد. پس
 از هر عمل خواندن و نوشتن، نوک طبق دستور
 واحد کنترل با اندازه یک خانه بر روی نوار
 به چپ یا به راست حرکت میکند. (۳) یک واحد
 کنترل با تعداد حالت های متناهی n که در بدو
 امر در حالت ابتدایی بخصوصی قرار گرفته و
 تغییر حالت آن تابعی است از حالت فعلی و
 علامت خوانده شده بر روی نوار. در واقع،
 ماشین تورینگ مورد نظر ما به تابع $Q(s, q)$ ،
 $W(s, q)$ و $M(s, q)$ مشخص میشود که به ترتیب
 حالت بعدی Q ، نوشتن یا ننوشتن علامت X
 بجای S و حرکت M که باید پس از نوشتن صورت
 گیرد (یک خانه به چپ یا یک خانه به راست)
 را بر حسب حالت فعلی q و علامت خوانده شده s
 نشان میدهند. (۴) یکی از حالت های واحد
 کنترل بعنوان حالت توقف در نظر گرفته میشود
 بطوریکه هرگاه ماشین تورینگ به این حالت
 برسد، کار خود را متوقف میکند.

ب - برای هر تابع محاسبه پذیر $g(y)$ ،
 ماشین تورینگی از نوع فوق وجود دارد (یعنی
 در واقع توابع Q و W و M را میتوان یافت)
 بطوریکه اگر ماشین تورینگ مورد بحث
 نواری که روی آن y علامت X دنبال هم قرار



برنامه نویسی بعنوان نظامی با طبیعت ریاضی

توضیح ویرایشگر: چه رابطه‌ای بین ریاضیات و علوم کامپیوتر وجود دارد؟ آیا علوم کامپیوتر بخشی و یا زیرمجموعه‌ای از ریاضیات است؟ آیا برای درک بهتر علوم کامپیوتر نیازی به دانستن ریاضیات وجود دارد؟ در شماره قبلی ماهنامه (سال دوم، شماره ۲، اردیبهشت ۱۳۶۰، شماره پیاپی ۲۲، صفحات ۱۳ تا ۱۵)، طی مقاله‌ای از نظرات "دانشکده کنت" در این زمینه مطلع شدیم. در این شماره، ترجمه مقاله‌ای از "دیگر دایکسترا"، یکی از صاحب‌نظران رشته کامپیوتر، را که مطالبی در رابطه با سوالات فوق در بر دارد میخوانیم.

در این مقاله می‌خواهم برنامه نویسی را بعنوان یک فعالیت ریاضی ارائه نمایم بدون اینکه سعی کنم تعاریفی دقیق از "ریاضیات" و "برنامه نویسی"، بنحوی که مورد قبول همه ریاضی دانان و برنامه نویسان باشد، ارائه دهم.

به اعتقاد من، غالباً "برای این بکنه" توافق داریم که بیشتر کارهای ریاضی دارای خصوصیات زیر میباشند:

۱ - ادعاهای ریاضی، در مقایسه با انواع دیگر از فعالیت‌های هوشمندانه، دارای دقتی غیر عادی است.

۲ - ادعاهای ریاضی عمومیت دارند، یعنی در یک رده بسیار بزرگ (و غالباً نامتناهی) از موارد مشابه صدق میکنند.

۳ - ریاضیات دارای نظام خاصی از استدلال است که به چنین ادعاهایی با درجه اطمینان بسیار بالا اجازه وجود میدهد.

روش ریاضی قدرت خود را از ترکیب سه خصوصیت فوق کسب میکند. از طرف دیگر، به عقیده من هنگامیکه یک فعالیت هوشمندانه این سه خصوصیت را تا حد زیادی دارا باشد، میتوان (بدون در نظر گرفتن اینکه موضوع آن تا چه حد برای ریاضی دانان آشناست) آنرا "فعالیتی با طبیعت ریاضی" نامید.

یک برنامه نویسی، الگوریتم‌ها را بمنظور اجرای مکانیکی و نیز کنترل ابزار محاسباتی (از نوع موجود یا قابل تصور) طراحی میکند. این دستگاه‌ها، که غالباً "الکترونیک" هستند، قدرت خود را از دو مشخصه پایه کسب میکنند: یکی مقدار اطلاعاتی که میتوانند ذخیره کنند و مقدار پردازشی که میتوانند در مدت زمان کوتاهی انجام دهند (که ایمن دو مقدار، هر دو بسیار بزرگ و ماورای قدرت تصور انسان است) و دیگری قابلیت اطمینان و "مطیع" بودن آنها بعنوان اجرا کننده‌های الگوریتم (تبعیت بی چون و چرای آنها از دستورات).

این مطیع بودن باعث میشود که برنامه نویسان بهنگام نوشتن برنامه، دقت زیادی صرف کنند، چه اگر برنامه بی معنی باشد، نتیجه‌ای که ماشین تولید میکند نیز بی معنی خواهد بود. برنامه نویسان کم تجربه، غالباً از این اطلاعات محض ماشین، بدلیل آنکه نمیتوان به درک و شعور خود آن متوسل شد، شکایت دارند. حال آنکه اغلب برنامه نویسان با تجربه معتقدند تنها بدلیل همین اطلاعات محض ماشین است که میتوان از آن با اطمینان استفاده نمود. در نتیجه، یک برنامه نویسی خوب، دقت دستگاه را بعنوان نقطه ضعف آن در نظر نمیگیرد و میدانند که بدون وجود چنین دقتی، قادر به کار کردن نخواهد بود (خاصیت اول).

کار برنامه نویسی نیز معمولاً "عمومیت" دارد، یعنی هر برنامه چنانچه روی داده‌های ورودی مختلفی اجرا شود، قادر است محاسبات مختلفی را انجام دهد. هر ادعا در مورد یک برنامه، در واقع ادعایی در باره تمام آن رده از محاسباتی است که تحت کنترل آن میتواند انجام گیرد و "طراحی یک الگوریتم" چیزی بیشتر یا کمتر از "طراحی کل یک رده از محاسبات" نیست. بنابراین کار برنامه نویسی دارای دومین خصوصیت از خصوصیات ذکر شده نیز میباشد.

سرانجام، درباره درجه اطمینان کار برنامه نویسی چه میتوان گفت؟ این درجه اطمینان، به دو دلیل باید خیلی بالا باشد. اولاً باین دلیل که یک برنامه بزرگ و پیچیده، تنها با استفاده دقیق از روش تقسیم مسئله به زیرمسائل ساده‌تر قابل پیاده کردن است. بنابراین چنین برنامه‌ای شامل مثلاً N مؤلفه تشکیل دهنده میباشد. حال اگر احتمال صحت هر مؤلفه، برابر p باشد، میتوانیم در مورد احتمال صحت کل برنامه (P) بگوئیم: $P \leq p^N$. بعبارت دیگر، اگر مقدار N بزرگ باشد، p باید خیلی به یک نزدیک باشد تا P از صفر قابل تشخیص گردد! چنانچه نتوانیم مؤلفه‌ها را بنحوی خوب طراحی کنیم، نمیتوان امیدوار بود که کل برنامه اصلاً کار کند.

ثانیاً، اگر ما بعنوان افراد جامعه، بخواهیم به اجرای الگوریتمی اعتماد داشته باشیم باید درجه اطمینان آن خیلی بالا باشد. و حتماً میدانید که مابه‌الگوریتم‌هایی که برای کنترل ترافیک هوایی، بانکداری، مراقبت از بیماران و پیش‌بینی زلزله وجود دارد عملاً اعتماد میکنیم.

ولی صداقت حکم میکند که اضافه کنم امروزه بطور متوسط درجه اطمینانی که در حرفه برنامه نویسی وجود دارد، به آن اندازه‌ای که باید باشد نرسیده است.

از نظر تاریخی، این نقطه ضعف قابل درک میباشد. سنت "برنامه نویسی" بسیار جوان است و تا همین اواخر که هنوز ماشین‌ها کوچک بودند، برنامه نویسی مسئله مهمی

بشمار نمی‌آمد (قبل از وجود ماشین‌ها، ابتدا مسئله‌ای نبود!) ولی در دهه بیست و یک سال اخیر، قدرت ماشین‌آلات محاسباتی حداقل هزار برابر شده و در نتیجه، زمینه حرفه برنامه نویسی کاملاً تغییر یافته است. روش قدیمی برنامه نویسی چنین بود که ابتدا برنامه‌ای را می‌نوشتند و سپس چند بار آنرا آزمایش میکردند. چنانچه آزمایش نتایج صحیحی به بار می‌آورد، نتیجه‌گیری میشد که برنامه "صحیح" است.

عدم کارآیی این نظریه با افزایش پیچیدگی الگوریتم‌ها بیشتر نمایان شد تا سرانجام، حدود پنج سال قبل، به "بحران نرم افزار" منجر گردید. یکی از اولین ملاحظات در زمینه‌ای که بعدها بعنوان "روش برنامه نویسی" شناخته شد، این سوال درباره درجه اطمینان بود که: "چگونه میتوان به الگوریتم‌ها اعتماد نمود؟"

تحلیل وضعیت نشان داد که آزمایش‌ش برنامه، تنها برای نشان دادن "حضور" اشتباه موثر است ولی هرگز نمیتواند "عدم حضور" اشتباه را متجسس نماید، زیرا تعداد دفعاتی که میتوانیم برنامه را آزمایش نمائیم در مقایسه با تعداد حالات ممکن مطلقاً ناچیز است. بنابراین، بنظر رسید که تنها راه خروج از این مخته، اثبات نمودن صحت برنامه است.

این فکر که میتوان صحت برنامه را اثبات کرد، ابتدا تولید شک نمود، چون بسیاری از افراد سالخورده بعنوان "قانون طبیعت" پذیرفته بودند که هر برنامه بی تردید حاوی اشتباهاتی است! البته این امر چندان هم بی دلیل نبود.

در ابتدا، مشخص نبود که اثبات صحت برنامه چه شکلی خواهد داشت. برای هراثبات باید اصول موضوعه‌ای در اختیار داشته باشیم. در باره موضوع بحث ما، باید تعریفی مبتنی بر اصول موضوعه از جنبه‌های معنایی زبان برنامه نویسی مورد استفاده موجود باشد. پس از کوشش‌هایی چند، یک روش برای تشریح معنی زبان، که میتوانست احتمالاً یک نقطه شروع عملی برای اثبات صحت برنامه باشد، پدیدار گردید.

ولی هنگامیکه در جهت اثبات صحت برنامه‌های موجود کوشش شد، نتیجه عمل ما بی‌سربسکننده بود: اثبات‌ها بقدری طولانی، پیچیده و درهم بود که شخص را ارضا نمیکرد؛ و یا حاصل از این موضوع بر تردیدها می‌افزود. کشف سه نکته اساسی منته را بکلی تغییر داد: اولین کشف این بود که مقدار کار لازم جهت اثبات صحت یک برنامه، بستگی زیادی به ساختمان آن برنامه دارد.

کشف دوم در رابطه با چند قضیه مفید در مورد ساختارهای برنامه‌ای بود که نیاز به مراجعه مستقیم به اصول موضوعه را در بسیاری از موارد از بین می‌برد.

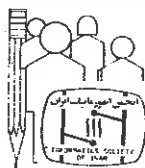
دنباله مقاله "از تحوّل الکترونی..."

فهرست مأخذ (قسمت دوم)

- [3] Lillienthal, J.E. U.S. patent no. 1900018, filed March 1928. (Compare Heil, O., British Patent no. 937457, 1935).
- [4] Shockley, W. "The theory of P-N junctions in semiconductors and P-N junction transistors," *Bell Syst. Tech. J.* 28(1949), 435.
- [5] Wolff, M.F. "The genesis of the integrated circuit," *IEEE Spectrum* (Aug. 1976), 45-53.
- [6] Mead, C., and Conway, L. *Introduction to VLSI systems*, Addison Wesley, Reading, Mass., 1980.

ترجمه محمد میرزا عبداللّهی و بهروز پیرهای

گزارش کامپیوتر



ما همامه انجمن انفورماتیک ایران - زیر نظر کمیته انتشارات انجمن سرپرست کمیته دکتر بهروز پیرهای می

نشانی: صندوق پستی ۴۵/۱۳۶، آریا شهر، تهران ۱۴، ایران

آخرین مهلت قبول مطلب و آگهی، دومین یکشنبه ماه انتشار است. نقل مطالب ما همامه "گزارش کامپیوتر" با ذکر مأخذ آزاد است.



ارسال مطالب برای "گزارش کامپیوتر"

از اعضای انجمن انفورماتیک ایران و سایر افراد علاقمند به همکاری با "گزارش کامپیوتر" تقاضا داریم به نکات زیر توجه کنند:

۱- اخبار و وقایع محل کار خود در زمینه انفورماتیک را حتی الامکان همراه با اسناد و مدارک مربوطه برای ما بفرستید. مثالهایی از مطالب مناسب عبارتند از: انعقاد یا فسخ قراردادهای، طرح و پیاده کردن سیستمها، کاربردهای نوین و برگزاری سمینارها و سخنرانیها.

۲- اگر در روزنامهها، مجلات، کتابها و سایر نشریات به مطالب جالبی برخوردید، بریده یا فتوکپی تمیز و خوانای آنها را با ذکر مشخصات کامل مأخذ (نام نشریه، تاریخ انتشار، ناشر، محل انتشار، شماره صفحه) برای ما بفرستید.

۳- مقالات باید بصورت ماشین شده (با ذکر مشخصات مأخذ و نام نویسنده و زمینه کردن نسخه‌ای از اصل مطلب در مورد مقالات ترجمه شده) در یک روی کاغذ و در دو نسخه ارسال شوند. ما همامه "گزارش کامپیوتر" در قبول یا رد مقالات رسیده، خلاصه کردن آنها و هر نوع حک و املاح دیگر آزاد است. در صورت تمایل میتوانند قبل از تهیه مقاله، تلفنی یا با درمورد آن مشورت کنید.

۴- مقالات و سایر مطالب رسیده مسترد نخواهند شد. نویسنده شخصاً مسئول نظرات ابراز شده در مقاله است و درج مقاله بمعنی تأیید محتوای آن توسط انجمن انفورماتیک ایران نیست.

خواهد شد که برای آن، بافت مناسب (از لحاظ زمان و مقدار حافظه) چندین برابر کوچکتر از بافت مناسب مربوطه در سطح قبلی است. در نتیجه، تعداد سطوحی که در یک ساختار سلسله-مراتبی قابل تشخیص و تمیز است تقریباً معادل لگاریتم نسبت بین بزرگترین و کوچکترین بافت می‌باشد.

در برنامه نویسی، در حالیکه کل محاسبات (برنامه) ممکنست به زمانی معادل یک ساعت نیاز داشته باشد، کوچکترین بافت زمانی (یک یا چند دستورالعمل) چیزی در حدود یک میکروثانیه است. عبارات دیگر، محیطی داریم که در آن، این نسبت میتواند بسادگی از یک میلیارد تجاوز نماید و من محیط دیگری را که در آن یک تکنولوژی واحد چنین حوزه وسیعی را دربر بگیرد، سراغ نداریم. بنظر میرسد که مطالب عنوان شده در

پاراگراف فوق بعضی از خصوصیات یگانه برنامه نویسی را بعنوان یک فعالیت هوشمندانه روشن کرده باشد. چنانچه این مفهوم از برنامه نویسی را با برداشتی که من از "برنامه" درسی استاندارد در ریاضیات دارم (که امیدوارم درست باشد) مقایسه نمایم، به تفاوت‌های زیر بر میخورم:

۱- در برنامه درسی استاندارد ریاضیات، دانشجو با مجموعه استانداردی از مفاهیم ریاضی آشنا (گاه حتی خیلی آشنا) میشود و کمتر امکان می‌یابد تا مفاهیم جدیدی را شخصاً ارائه نماید.

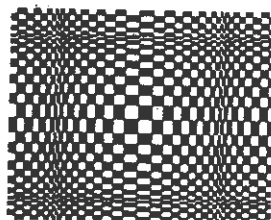
۲- در برنامه درسی استاندارد ریاضیات، دانشجو با مجموعه استانداردی از روشهای نشان گذاری آشنا (گاه حتی خیلی آشنا) میشود و کمتر امکان می‌یابد که در هنگام نیاز، نشان گذاری خاص خود را ابداع کند.

۳- در برنامه درسی استاندارد ریاضیات، دانشجو غالباً با مسأله "کوچک"، که از نظر معنایی در سطح واحدی قرار دارند، برخورد میکند. در نتیجه، بسیاری از دانشجویان بعوض اینکه به ریاضیات بصورت هنر سازماندهی به افکار بنگرند، آنها بصورت هنر تنظیم نمادها و بروی صفحه کاغذ می‌بینند.

ترجمه و خلاصه شده توسط ابراهیم نقیبزاده مشایخ از "آمریکن مانتیکال مانیتل"، ژوئن-ژوئیه ۷۴

یادداشت‌ها

1. Assertions
2. Axioms
3. Axiomatic definition
4. Hierarchy
5. Notational techniques
6. Symbols



آخرین و موثرترین کشف این بود که فرما "چنانچه اثبات صحت برنامه نیز مقدور شود، این کار در واقع بمشابه بستن گاری به جلوی اسباب است! روش خیلی بهتر این است که اثبات صحت و طرح خود برنامه را به موازات یکدیگر پیش ببریم: یعنی با انتخاب یک ساختار کلی برای روش اثبات صحت، برنامه‌های طرح کنیم که این اثبات بر روی آن قابل اعمال باشد.

چنانچه این مقاله را با عبارات خوش-بینانه فوق به پایان برم، این برداشت غلط تولید میشود که اکنون همه مشکلات فطری برنامه نویسی حل شده است، درحالیکه این مطلب صحیح نیست. بیشترین چیزی که میتوان بگویم اینست که اکنون دید بهتری نسبت به طبیعت مشکلات برنامه نویسی پیدا شده است.

یک برنامه نویس باید بتواند هم در یک زبان طبیعی و هم در سیستم‌های صوری، بنحو بسیار خوبی مقصود خود را بیان نماید. نیاز به تسلط به یک زبان طبیعی به دو دلیل است: اولاً، در اغلب موارد، مسئله در یک زبان طبیعی (مثلاً انگلیسی) برای برنامه نویسنده مطرح میگردد و او باید در همان زبان، تعبیر خود از مسئله یا اصلاحات آنرا تشریح نماید. این امر موجب بسیاری از سوء تفاهات بوده است تا جائیکه امروزه اعتقاد افراد بسیاری بر اینست که زبانهای طبیعی، مانع از انگلیسی، برای برقراری این نوع ارتباطات مناسب نیستند. من شخصاً با این نظر مخالفم و برعکس، همواره این احساس را داشته‌ام که زبان طبیعی ما با آنچه که ما آنرا "درک" می‌نامیم رابطه بسیار نزدیکی دارد و باید بتوانیم از آن برای بیان نمودن آنچه که درک کرده‌ایم استفاده کنیم.

ثانیاً "نباید چشم خود را براین واقعیت ببندیم که "صوری کردن" همواره پس از بروز واقعیت صورت می‌گیرد و لذا زبان طبیعی، علی‌الخصوص هنگامیکه باید مفاهیم جدیدی ارائه شود، وسیله‌ای ضروری برای فکر کردن است. و ارائه نمودن مفاهیم جدید نیز کار همیشگی برنامه نویسنده است: او باید مفاهیم جدیدی را که در صورت مسئله اصلی ظاهر نشده است، ارائه نماید تا قادر به یافتن، تشریح و درک راه حل خود برای مسئله باشد.

البته چنین نحوه عملی در مورد اغلب کارهای ریاضی امری غادی است ولی دلایلی برای این ادعا وجود دارد که در برنامه نویسی میزان کار سنگین تر از هر زمینه دیگری میباشد. علاوه بر مسائل مربوط به دقت و صراحت، مشکلی بنام اندازه مسئله پیشاروی برنامه نویسنده قرار میگیرد. بنظر میرسد که هنگامیکه با موضوع پیچیده‌ای سروکار داریم، مفهوم "سلسله مراتب" یک مفهوم کلیدی است. یعنی چیزی که در یک سطح بعنوان واحد تجزیه و تحلیل نشده منظور میگردد، در سطح بعدی بصورت یک موضوع ترکیبی در نظر گرفته

دستگاه باید دارای حفاظ های مخصوص باشد چون در غیر اینصورت بسامد ۵۰ هرتز در خروجی ظاهر خواهد شد.

کنترل کامپیوتر از راه بازخورد زیستی

شخمی را در نظر بگیرید که تقریباً " بطور کامل فلج می باشد. این سیستم را می توان به عنوان یک کلید قطع و وصل برای کنترل یک دستگاه پیچیده دیگر بکار برد (مثلاً) ممکن است با حرکت آبرو بتوان بر سیستم اثر گذاشت). یک مددکار کامپیوتری از این نوع برای اشخاص فلج ساخته شده است که می تواند حروف الفباء را بصورت ترتیبی نمایش دهد. حروف را می توان تک تک انتخاب کرده و سپس آنها را برای درست کردن جملات کامل با هم ترکیب نمود. اگر نرم افزار مناسبی با در نظر گرفتن زمان لازم برای عکس العمل مربوطه نوشته شود، می توان با یک ورودی یک بیتی کارهای زیادی را انجام داد.

برنامه های ساده ای که بزبان پیسیک برای ریز کامپیوترها نوشته شده اند برای منظوره های فوق مفیدند. مثلاً یکی از این برنامه ها، استفاده کننده را با هفت انتخاب روبرو میکند و بر اثر هر انتخاب خاص، برنامه فرعی مربوطه را انجام میدهد. در این برنامه فرض شده است که استفاده کننده عکس العمل مثبت خود را از طریق بالا بردن سطح منطقی یک بیت بخصوص (مثلاً از راه الکترودهای سطحی که به آبروی وی وصل شده اند) نشان میدهد.

خروجی سیستم دارای هفت چراغ می باشد که به بیت های یک تا هفت متصل شده اند. با روشن شدن اولین چراغ توسط برنامه، شخصی استفاده کننده باید راجع به انجام یا عدم انجام عملی که با بیت اول مشخص میشود تصمیم بگیرد. اگر جواب او مثبت باشد، فقط

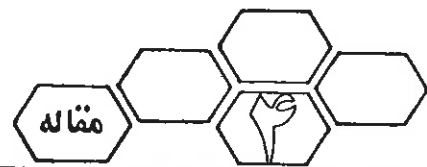
ثبت کردن فعالیت های الکتریکی در بدن

فعالیت داخل مغز یا یک ماهیچه را در نظر بگیرید. هر عصب و لئان های الکتریکی بسیار کوچکی را تولید میکند. با در نظر گرفتن میلیاردها سلولی که در سراسر بدن چنین ولتاژی را تولید میکنند، نتیجه حاصل قابل ملاحظه است. این وضعیت را می توانید با یک استاد بیوم فوتبال در جریان برگزاری یک مسابقه مقایسه نمایید. یک مجمع در خارج از استادیوم نمیتواند فریاد چند نفر را بشنود، ولی فریاد پنجاه هزار نفر داستان دیگری است.

موضوع دیگر مربوط به جریان پیشرفت مسابقه می باشد. صداهای بلند که از یک قسمت بخصوص از استادیوم بگوش میرسد نشان دهنده تشویق عده ای تماشاگر است. از روی چنین تشویقی در آخر بازی، میتوان برنده مسابقه را تشخیص داد. ملاحظه دقیق و تداعی کلیدهای اصلی مسئله هستند. علائم الکترود کاردیوگرام، الکترومایوگرام و الکترورنیف - لوگرام که ثبت میشوند باید بدقت مورد مطالعه و تفسیر قرار گیرند.

تمام علائم الکتریکی فوق الذکر را میتوان با استفاده از الکترودهای متصل به سطح بدن بدست آورد. موقعیکه یک ماهیچه دوسر حرکت میکند، ولتاژ کوچکی نسبت به نقاط دیگر بدن تولید میکند. برای ثبت کردن این ولتاژ، به تقویت کننده هایی با مقاومت ظاهری بسیار زیاد در ورودی و بسامد رادی ۵۰ هرتز نیاز داریم.

مداری که در شکل ۱ ملاحظه میکنید قادر است امواج فوق الذکر را آشکار نموده و آنها را به کامپیوتر بفراستد. این مدار دارای یک واحد مجزا کننده، دو تقویت کننده و یک مقایسه کننده (بمنظور آشکار کردن قله موج ها) می باشد. تمام اتصالات بین بدن و



کنترل ساده توسط ذهن: بازخورد زیستی برای کامپیوتر

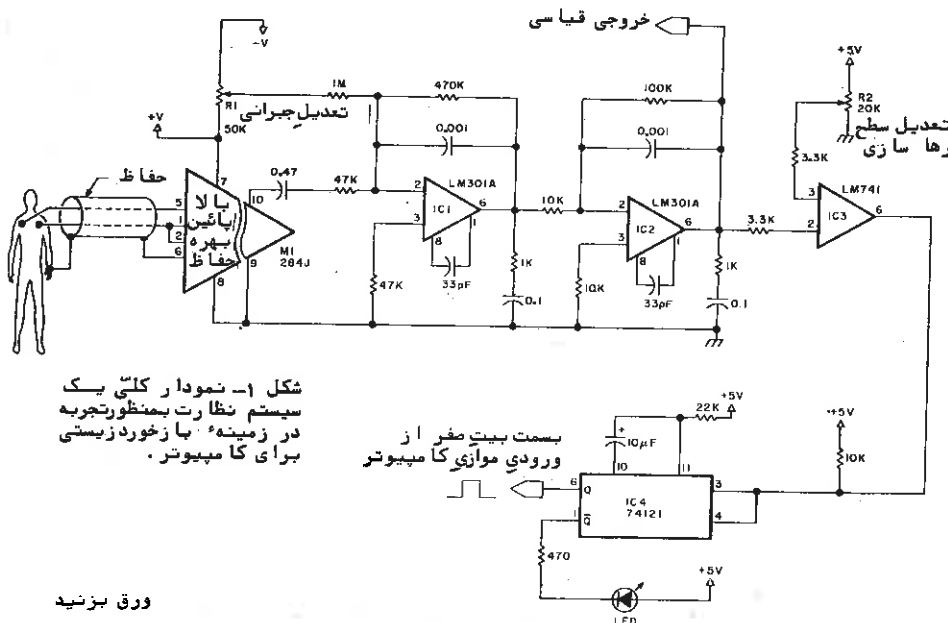
مقدمه

وسایل گوناگونی برای نظارت بر امواج مغزی بصورت تجارتي به بازار عرضه شده اند. امواج مغزی منبع یکی از انواع انرژی هستند که در رده "بازخورد زیستی" قرار میگیرند. اینگونه امواج بصورت تپش های الکتریکی می باشد که در نتیجه فعالیت مغز یا ماهیچه ها از درون بدن میگذرند. عصبها و ماهیچه های بدن انسان از راه یک فرآیند الکتروشیمیایی، نظیر آنچه در باطری ها وجود دارد، جریان الکتریکی تولید میکنند.

موقعیکه ما میخواهیم بازوی خود را بلند کنیم، مغز یک تپش الکتریکی به ماهیچه ها مربوطه میفرستد. مقدار انرژی الکتریکی که از مغز فرستاده میشود بسیار کم است: حداکثر در حدود چند صد میکروولت. آشنا ترین اینگونه علائم الکتریکی، ولتاژی است که بر اثر ضربان قلب تولید میشود. منحنی تغییرات این ولتاژ نسبت به زمان، الکتروکاردیوگرام نام دارد. در این منحنی، انقباض های ماهیچه ای متعددی باعث پیدایش یک طیف بسامدی از ۰/۱ تا ۱۰۰ هرتز میشوند.

گروه دیگری از علائم الکتریکی، امواج هستند که توسط ماهیچه های بزرگی متصل به اسکلت (نظیر ماهیچه های دوسر) تولید میشوند. صورت ثبت شده اینگونه امواج را الکترومایوگراف مینامند. این امواج فقط در زمان انقباض ماهیچه ها تولید میشوند و برخلاف علائم الکتریکی ناشی از ضربان قلب، دوره ای نیستند. بسامد اینگونه امواج بسیار کم است ولی دامنه آنها به ۵ تا ۱۰ میلی ولت میرسد. به علت این دامنه نسبتاً بزرگ است که ثبت اینگونه امواج ساده می باشد.

آخرین نوع علائم الکتریکی مهم در بدن، امواج با دامنه بسیار کوچک است که در خود مغز تولید میشوند. صورت ثبت شده اینگونه امواج را الکترورنیف لوگراف مینامند. دامنه اینگونه امواج حدود ۵۰ میکرو ولت است که یک بسامد بسامدی بین ۱ تا ۳۰ هرتز را می پوشانند. این علائم الکتریکی به موجهای "ولتا"، "تتا"، "لتا" و "یتا" تقسیم می شوند. این تقسیم بندی بر حسب درجه فعالیت در باندهای بسامدی بخصوص صورت میگیرد و گفته میشود که تفاوت های موجود در این موجها، نشانگر گرایش های خاص در شخصیت افراد می باشد.



شکل ۱- نمودار کتی یک سیستم نظارت بمنظور تجربه در زمینه بازخورد زیستی برای کامپیوتر.

بسمت بیت مغز از ورودی موزی کامپیوتر

این پیش بینی ها در گزارشی که هدف اصلی آن بررسی و پیش بینی توسعه بازار الکترونیک بطور کلی است درج شده است، ولی داده های زیادی نیز در مورد صنعت کامپیوتر در این گزارش یافت میشود. طبق آمار استخراج شده از این گزارش، رشد کلیسی سخت افزار در سال گذشته حدود ۱۵ درصد بوده و نیز حجم بازار در اروپای غربی به ۱۸/۳ میلیارد دلار رسیده است.

نرخ توسعه امسال حدود ۲۵ درصد پیش بینی میشود اما این گرایش در بین کشورهای مختلف بطور قابل ملاحظه ای متفاوت است. در سال گذشته، کل تولید کشورهای اروپای غربی ۱۵/۳ میلیارد دلار بوده است و بنابراین، این کشورها ۸۴ درصد احتیاجاتشان را خود تولید کرده اند. همچنین در سال گذشته، نرخ رشد تولیدات سریع تر از رشد فروش بوده است. ولی پیش بینی میشود که در سال ۱۹۸۱، این وضعیت دگرگون خواهد شد. یکی از نتایج جالب این بررسی، ازدیاد وابستگی بسیاری از شرکتهای اروپایی به تهیه کنندگان خارجی، بخصوص آمریکای شمالی، در زمینه تجهیزات داده پرداز است.

تخمین زده میشود که در سال ۱۹۸۰، میزان کسری در موازنه تجارت در حدود ۳ میلیارد دلار بوده و نیز انتظار میرود میزان این شکاف در سال جاری حدود ۴/۶ میلیارد دلار باشد. این امر نشان میدهد که سازندگان اروپایی در حال از دست دادن بازار هستند و احتمال میرود میزان واردات آنها در سال آینده افزایش یابد.

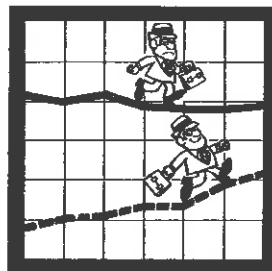
بررسی نشان میدهد که نرخ رشد در مورد تجهیزات جنبی بالاتر از مقدار متوسط خواهد بود. این بخش در سال ۱۹۸۱ احتمالاً حدود ۲۸ درصد رشد خواهد داشت در حالیکه رشد سیستم های کامپیوتری حدود ۲۰ درصد تخمین زده میشود.

از "کامپوتینگ"، ۲۲ ژانویه ۱۹۸۱

ولی ما تنها بدنبال بازار خانگی بوده ایم و در قیمت گذاری محصولات و تشخیص اهمیت سازگاری آنها با سیستم های مختلف تلویزیونی اشتباه کرده ایم.

با وجود این، شرکت "تگزاس اینسترومنتس" سیاست فروش محصولات خود در بازار خانگی را ادامه خواهد داد. سیستم ۹۹/۴، که در حدود دو سال پیش با تبلیغات زیادی به بازار عرضه گشت، تا این اواخر حدود ۱۰۰۰۰ دلار قیمت گذاری شده بود. با دعای شرکت "تگزاس اینسترومنتس"، این ماشین اولین کامپیوتر شخصی برای استفاده انحصاری در منازل است که از سیستم عامل سخت افزاری استفاده میکند. یکی از مهم ترین مسائل این ماشین عدم وجود نرم افزار و بسته های پیش نوشته شده کافی برای برآوردن انتظار مشتریان آن بود.

از "کامپوتینگ"، ۱۱ دسامبر ۱۹۸۰



رشد بیشتر علیرغم رکود اقتصادی

انتظار میرود که بازار کامپیوترها در اروپای غربی در سال ۱۹۸۱ گمakan نرخ رشد سریعی داشته باشد. طبق آخرین پیش بینی ها، علیرغم رکود اقتصادی، حجم کلی بازار در اروپای غربی امسال به بیش از ۲۳ میلیارد دلار و تا سال ۱۹۸۴ به ۴۳ میلیارد دلار خواهد رسید.

کافیت که آبروی خود را حرکت دهد تا کامپیوتر برنامه فرعی مربوطه را انجام دهد. اگر استفاده کننده در مورد اولین انتخاب عکس العملی از خود نشان ندهد، برنامه چراغ دوم را روشن میکند و مراحل فوق بهمین ترتیب ادامه می یابند.

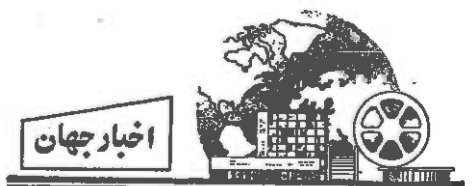
نتیجه گیری

شاید اینهمه کوشش برای وارد کردن یک داده تک بیتی غیرمستعمل بنظر برسد، ولی در مواقعی که راه دیگری برای تماس با کامپیوتر موجود نباشد، اهمیت این روش فوق العاده زیاد است. حداقل فایده روش فوق اینست که زمینه خوبی را برای تجربه در مورد بازخورد زیستی فراهم میکند. با اضافه کردن تقویت کننده ها و مافی های مخصوص برای ثبت امواج مغزی، میتوان مجموعه ای غنی و جالب از آزمایش ها را انجام داد. همچنین کنترل مستقیم یک سیستم توسط امواج مغزی راه را برای استفاده افراد معلول از کامپیوترهایی شخصی باز میکند.

یادداشت ها

1. Biofeedback
2. Electrocardiograph (ECG or EKG)
3. Electromyograph (EMG)
4. Electroencephalograph (EEG)

ترجمه و خلاصه شده توسط فرهاد ماحیان
استادیار دانشگاه ملی ایران
از مجله "بایت"، ژوئین ۱۹۷۹



تقلای "تی آی" در بازار کامپیوترهای خانگی

شرکت "تگزاس اینسترومنتس"، یکی از سازندگان آمریکایی مدارهای مجتمع الکترونیکی، با عوض کردن وضع فروش کامپیوترهای خانگی به اشتباه خود در بازاریابی اعتراف کرد. این شرکت همچنین اعلام کرد که بدلیل "برخی ضعف های بازار" در یکی از اداره های مرکزی به کارمندان خود "مرخصی اجباری" داده است.

بدنبال تقلیل قیمت کامپیوتر شخصی "تی آی ۹۹/۴" به میزان تقریباً ۳۰۰ دلار، یک مقام آگاه در اداره مرکزی بخش اروپایی این شرکت اعلام کرد که هنوز تقاضا برای این محصول "کم" است. وی افزود که مسئله اصلی دنبال کردن بازاری است که اصلاً وجود ندارد. اغلب رقبای ما مانند "آپل" و "ریدیو شک" کامپیوترهای شخصی خود را به ماشین های کوچک تجاری تبدیل کرده اند،

میزان تولید و فروش سخت افزار کامپیوتر در اروپای غربی (ارقام به میلیون دلار)

نام کشور	تولید سخت افزار			فروش سخت افزار		
	۱۹۷۹	۱۹۸۰	۱۹۸۱	۱۹۷۹	۱۹۸۰	۱۹۸۱
آلمان غربی	۳۲۳۱	۳۹۳۱	۴۵۰۴	۳۷۳۵	۳۹۶۸	۵۶۰۴
اسپانیا	۱۷۶	۲۰۱	۲۴۵	۵۰۶	۵۷۳	۶۹۱
آتریش	۵۷	۶۹	۸۱	۲۴۹	۳۱۰	۳۷۰
انگلستان	۲۶۷۲	۳۴۶۶	۴۳۰۷	۳۰۸۰	۳۷۰۴	۴۷۷۴
ایتالیا	۱۹۴۵	۲۲۴۵	۲۷۳۵	۲۱۷۵	۲۵۳۰	۳۰۶۰
بلژیک	۴۷۸	۵۲۴	۵۸۸	۷۴۸	۸۴۸	۹۸۲
دانمارک	۶۱	۶۶	۷۷	۲۵۲	۲۷۶	۳۲۷
سوئد	۳۹۳	۴۶۶	۵۵۳	۵۱۰	۶۰۶	۷۱۶
سوئیس	۱۵۵	۱۹۲	۲۲۲	۲۳۹	۵۱۵	۶۱۱
فرانسه	۳۱۲۲	۳۴۴۴	۴۲۶۴	۳۰۹۵	۳۵۹۰	۴۲۳۳
فنلاند	۶۵	۷۵	۸۴	۱۷۰	۲۱۲	۲۵۶
نروژ	۸۹	۱۰۰	۱۱۲	۱۹۴	۲۲۶	۲۶۴
هلند	۴۴۴	۵۲۹	۶۴۲	۸۵۶	۹۹۷	۱۱۹۸



آزمایش موتور اتومبیل بکم کامپیوتر

چند نفر از خوانندگان اصطلاح "نقشه موتور" را شنیده‌اند؟ برای کسانی که با این اصطلاح آشنا نیستند، منظور از "نقشه موتور" شرح کامپیوتری کارآیی یک موتور احتراق داخلی است. یکی از سوالاتی که طراحان سیستم کنترل موتور با آن مواجه هستند اینست که آیا کنترل کننده‌ها باید از این نقشه‌ها استفاده کنند و یا اینکه بهتر است بصورت مستقل و تطبیق پذیر عمل نمایند.

مسئله اصلی در بهینه‌سازی کارکرد موتور، سعی در کنترل زمانی جرقه‌ها یا نسبت هوا به سوخت برای تولید حداکثر قدرت و یا رسیدن به حداقل مصرف است. پاکیزگی کار موتور از نظر تولید دود و سهولت رانندگی نیز از عوامل مهم بشمار می‌روند.

هنگامیکه موتور بر روی سکوی معاینه قرار می‌گیرد، میتوان با تنظیم دستی زمان جرقه‌ها و نسبت هوا به سوخت، کارآیی را افزایش داد. درحالیکه این کار انجام میشود، یک کامپیوتر داده‌های لازم را جمع آوری کرده و نقشه‌ای از مشخصات تولید میکند. این نقشه میتواند در حافظه یک پردازنده مایکرو ذخیره شده و توسط کنترل کننده موتور مورد استفاده قرار گیرد. بایسن ترتیب، پردازنده مایکرو از نقشه مشخصات بعنوان یک "جدول مراجعه"، که در آن اطلاعاتی در باره چگونگی تأثیر زمان بندی الکتریکی و مخلوط سوخت بر سرعت، درجه حرارت و سایر شرایط موتور یافت میشود، استفاده میکند.

در یک آزمایش، مینی کامپیوتری که بمنظور جمع آوری اطلاعات برای ساخت نقشه بکار رفته بود با یک پردازنده مایکرو از نوع "اینتل ۸۰۸۰" تمویض شد. با استفاده از ۱۰ کیلوبیت حافظه در این سیستم، نقشه برداری از یک موتور کامل دو لیتری ظرف مدتی کمتر از ۵۰ ساعت امکان پذیر گردید. به عقیده گروهی، روش نقشه برداری بهترین راه برای کنترل کننده‌های الکترونیکی نیست. بر اثر کمبود روزافزون منابع سوختی، این نقشه‌ها باید بیش از پیش دقیق تهیه شوند، چون موتور اتومبیل‌ها درجه می‌نیم نسبت سوخت به هوا تنظیم شده، بطوریکه کوچکترین تغییر در این نسبت میتواند باعث توقف موتور گردد.

مهمتر اینکه نقشه فوق الذکر فقط برای یک دسته از موتورها تدوین شده و راسده‌ای برای منظور نمودن تأثیر تغییرات آماری که در تولید عمده وجود دارد، در این نقشه‌ها بنظر نمی‌رسد. باین دلیل پیشنهاد شده است که استفاده از کنترل کننده‌های تطبیق پذیر ممکن است راه حل بهتری باشد. این دستگاه‌ها بجای مراجعه به نقشه موتور، پارامترهای مهم موتور در حال کار را تحت نظارت قرار داده و از طریق کنترل بازخوردی، آنرا در شرایط بهینه قرار میدهند.

مشکل اصلی در این روش، دستگاه‌های سنجش است. احتمالاً شرایط هیچ محیط دیگری حتی در یک هواپیما، به سختی شرایط محیط موتور اتومبیل نیست. لرزش، روغن، بنزین، سروصدای حرارت زیاد و تداخل الکتریکی کار دستگاه‌های سنجش را فوق العاده مشکل میکند. در دانشگاه "ساوت همپتون"، فعالیت زیادی بر روی کنترل کننده‌های تطبیق پذیر بمنظور حداقل کردن مصرف ویژه سوخت (منظور مصرف سوخت بازای واحد قدرت است) صورت گرفته است. ویژگی این سیستم اینست که انرژی تولید شده در هر سیلندرا جداگانه

اندازه گیری میکند. اندازه گیری مصرف سوخت با تعیین زمان باز بودن دریچه تزریق سوخت برای هر سیلندر امکان پذیر است. با توجه به سرعت ۳۰۰۰ دور در دقیقه موتور، پردازش این اطلاعات نیاز به یک مینی کامپیوتر قوی دارد.

موفقیت روش فوق بحثی نبوده است. تنظیم زمانی بهینه را در کمتر از شش چرخه موتور بوجود آورده و میزان اتلاف قدرت در حالت پایدار کمتر از یک درصد بوده است. زمان عکس العمل در سرعت ۶۰۰۰ دور در دقیقه فقط ۱/۱۲ ثانیه اندازه گیری شده است. نتیجه نهائی کار این کنترل کننده صرفه جویی به میزان ۱۰ تا ۳۰ درصد در مصرف سوخت بوده است.

علیرغم تبلیغاتی که برای تنظیم کامپیوتری موتور میشود، سیستم تولید قدرت اتومبیل تنها بخشی آن نیست که توسط پردازنده‌های مایکرو مورد بررسی قرار می‌گیرد. اخیراً شرکت فیات کنترل جعبه دنده اتومبیل‌ها را نیز به سیستم افزوده است. یکی از سیستم‌های اولیه مورد استفاده توسط فیات با نظارت بر سرعت موتور، نسبت دنده و سرعت اتومبیل و با کمک یک جدول مراجعه، بهترین نسبت دنده را برای حداقل کردن مصرف سوخت تعیین میکرد. با دعای شرکت فیات، با این روش در حدود ۱۷ درصد صرفه جویی در مصرف سوخت مقدور بوده است.

روشی مشابه روش شرکت فیات در یکی از دانشگاه‌های انگلستان در مورد وسایل نقلیه سنگین ۱۵ تا ۲۲ تنی بکار رفته است. هدف این مطالعات نیز، که بکم یک مینی کامپیوتر "پی دی پی - ۸" صورت می‌گیرد، پیدا کردن مناسب ترین نسبت دنده برای صرفه جویی بیشتر در مصرف سوخت است. این سیستم هنوز در مرحله آزمایش است ولی سازنده آن عقیده دارد که میزان صرفه جویی حاصل در مصرف سوخت چیزی بین ۱۵ تا ۲۰ درصد خواهد بود.

ترجمه و خلاصه شده توسط ویدا دانشی
از "کامپیوتینگ"، ۵ فوریه ۱۹۸۱

اطلاعیه

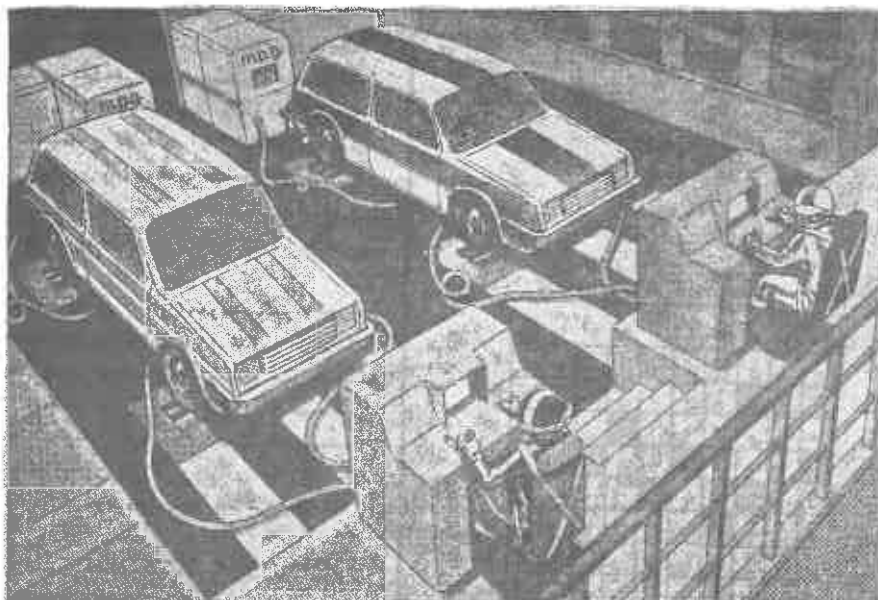
کمیته انتشارات انجمن انفورماتیک ایران
تقدیم میکند

دوره کامل سال اول (۱۳۵۸) و سال دوم (۱۳۵۹)

یا همان "گزارش کامپیوتر"

تیمت به ریال	اعفاء	سایرین
سال اول (۱۳۵۸) (مجموعه)	۲۰۰	۴۰۰
سال دوم (۱۳۵۹) (مجموعه)	۲۰۰	۴۰۰
سال دوم با جلد اعلاء	۵۰۰	۸۰۰

لطفأ با ارسال یک درجه انجمن
انفورماتیک ایران ویا رسید پرداخت
پول به حساب جاری شماره ۱۱۱۲ بانک
مخابرات شعبه شماره ۱۶۵۲ (دانشگاه
معلمی شریف) تهران، و درج نشانی کامل
پستی خود سفارش دهید. تحلیف مخصوص
اعفاء فقط برای نسخه اول منظور میشود.



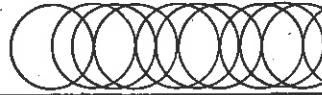
دنیاله اخبار انجمن

و علائم بر حسب نیاز ذخیره نمایند و بایستن ترتیب کارآئی خود را از طریق فراگویی افزایش دهد. تجربه نشان داده است که حداقل دقت لازم برای سخت افزار رقمی کننده متن، معادل چهار نمونه بازای ضخامت خط چاپی است. این امر بمعنی قابلیت برداشتن حداقل صد نمونه از هر سانتیمتر برای متون چاپی در کتاب های معمولی و متون ماشین شده میباشد. البته سخت افزار مورد استفاده برای سیستم مورد بحث دارای این درجه از دقت نبوده و بنا براین کاربرد عملی آن به شناسائی متون چاپی بزرگ (مناوین روزنامه ها) محدود گردیده است.

انجمن انفورماتیک ایران تقدیم میکند:

واژه نامه مقدماتی کامپیوتر و انفورماتیک (انگلیسی به فارسی)
حای ۶۰۰۰ واژه کامپیوتری - ۲۰۰ ریال
لطفاً با واریز کردن مبلغ فوق برای هر نسخه به حساب شماره ۱۱۱۷ بانک صادرات، شعبه شماره ۱۶۵۲ (دانشگاه صنعتی شریف تهران) و ارسال فیش بانکی سفارش دهید.

توضیح



تمحیح گزارش کمیته آموزش

در جدول ۲ از گزارش کمیته آموزش انجمن ("گزارش کامپیوتر"، شماره پیاپی ۲۳، فروردین ۱۳۶۰، صفحه ۷)، تأکید برنامۀ آموزشی دانشکده انفورماتیک و مدیریت سیستم دانشگاه ملی ایران با توجه به برنامه جدید آن دانشکده، بر روی "نرم افزار و سخت افزار" میباشد که بدینوسیله تمحیح میشود.

کمیته آموزش انجمن انفورماتیک ایران

برای تماس

برای تماس با انجمن از نشانی یا تلفن های زیر استفاده کنید:
نشانی: انجمن انفورماتیک ایران، صندوق پستی ۴۵/۱۳۶، آریا شهر، تهران ۱۳۴، ایران.
شماره تلفن مسئولین انجمن:

رئیس انجمن (بهرز پرها می) ۹۱۸۲۶۹
نایب رئیس (محمد میرزا عبداللهی) ۲۴۹۰۱۵-۸
دبیر انجمن (ابراهیم مشایخ) ۲۴۹۰۱۵-۸
خزانه دار (احمد مرآت نیا) ۶۱۲۱۲۹۶

شماره تلفن سرپرستان کمیته ها:

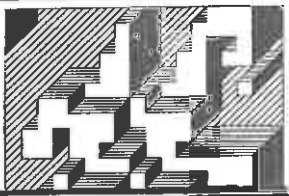
کمیته آموزش (خلیل فضل اللهی) ۲۴۹۰۶۰
کمیته انتشارات (بهرز پرها می) ۹۱۸۲۶۹
کمیته علمی و فنی (محمد میرزا عبداللهی) ۲۴۹۰۱۵-۸
کمیته عمومی و (ابراهیم مشایخ) ۲۴۹۰۱۵-۸
روابط عمومی

در خود دستگاه تلویزیون استفاده گردد، درمورد شناسائی پاشین خواهد آمد، حتی اگر مکانیزمی برای تخفیف اثر مدای خود تلویزیون در نظر گرفته شود.

برای بهبود قابلیت شناسائی سیستم در محیط های پارازیت دار، شرکت سانپو در نظر دارد آزمایشهایی با تقلیل مجموعه لغات سیستم و افزایش پیچیدگی نرم افزار آن انجام دهد. در سیستم مورد نظر، هشت صافی و یک پردازنده مایکرو از نوع "اینتل ۸۰۸۵" برای تحلیل فرکانس های صوتی مورد استفاده قرار میگیرد. در هر شش میلی ثانیه، خروجی صافی ها به اعداد ۸ بیتی تبدیل شده و بنا براین برای یک کلمه ۱/۸ ثانیه ای، جمعاً ۲۴۰۰ بیت اطلاع نتیجه میشود. چون ذخیره این مقدار اطلاع برای تمام کلمات در حافظه با دستیابی تصادفی عملی نیست، سیستم با ذخیره کردن این ۲۴۰۰ بیت در یک حافظه موقت و استخراج خلاصه ای از اطلاعات پراساسی ثرمالیزه کردن و انتخاب مشخصه های اصلی، نهایتاً عمل شناسائی را با مقایسه ۹۶ بیت اطلاع حاصله با مقادیر ذخیره شده انجام میدهد.

مقایسه بدین ترتیب صورت میگیرد که به هر بیت یک وزن اختصاص داده شده و در صورتیکه حاصل جمع وزنی بیت های یک فرمان نزدیک به یکی از کلمات ذخیره شده باشد، آن فرمان پذیرفته میشود. در غیر اینصورت، سیستم یک چراغ چشمک زن را که روی آن عبارت "تکرار کنید" دیده میشود روشن خواهد کرد.

از "آی تی ای اس کتریم"، ژانویه ۱۹۸۰



سرگرمی

حروف بجای ارقام

یکی از انواع جالب سرگرمی های ریاضی اینست که در یک عمل جمع، بجای هریک از ارقام، یکی از حروف لاتین را قرار دهیم (بطوریکه از قام مختلف با حروف متفاوت نشان داده شوند) و بخواهیم که از روی مورت حرفی عمل جمع، ارزش عددی حروف تعیین شود. یک مثال قدیمی در این زمینه عبارتست از:

$$SEND + MORE = MONEY$$

ولی زیبا ترین انواع این معماها آنهاشی هستند که در آنها حروف بکار رفته کلماتی با معانی عددی را تشکیل بدهند بطوریکه نتیجه عمل در مورد این اعداد نیز صحیح باشد.

آیا میتوانید پاسخ های دو مثال زیر از این نوع معماها را پیدا کنید؟

SEVEN +
SEVEN
SIX
TWENTY

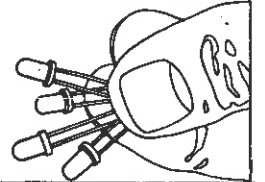
THIRTY +
TWELVE
TWELVE
TWELVE
TWELVE
NINETY

سفر زاپنی به اسامی "سابورو تامورا"، "ماسا هیگو تاناکا" و "ناشونوکی تامورا" توانسته اند با استفاده از یک مینی کامپیوتر "فاکوم - یو ۲۰۰" تمام معماهای از این نوع با حد اکثر شش عامل جمع را که در آنها اعداد یک تا بیست، سی، چهل، پنجاه، هشتاد، هشتاد و نود و صد بکار رفته باشند پیدا کنند. جدول زیر تعداد معماهای از این نوع را نشان میدهد.

تعداد عوامل جمع	تعداد معماهای دارای پاسخ منحصر بفرد	تعداد معماهای دارای پاسخ
۲	۶	۵
۳	۲۶	۲
۴	۶۱	۱۱
۵	۱۲۲	۲۹
۶	۲۸۱	۶۲

حل "جدول کلمات متقاطع" شماره قبل

۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
م	ا	م	ک	ن	د	ه	د	ه	د
ز	ا	ن	د	د	و	د	ن	م	م
ا	ت	ن	ا	ر	ا	س	ت	ه	ه
ز	ب	ت	ر	م	ت	ه	ه	ه	ه
ک	د	ر	ج	ت	ب	ل	ل	ل	ل
د	ر	و	ا	ز	ه	پ	ن	ی	ی
ک	د	ز	ی	ن	ک	ر	د	ن	ن
ا	ن	ف	و	ر	م	ا	ت	ی	ک



نوآوری ها

محبت با تلویزیون

آزمایشهایی که در زمینه کنترل تلویزیون بوسیله صوت صورت گرفته، راه را برای استفاده از این روش کنترل در سایر کاربردها نیز گشوده است. بعنوان مثال، یکی از سیستم های کنترل صوتی تلویزیون میتواند ۳۲ کلمه مختلف را (که معمولاً بصورت ۱۶ کلمه ادا شده توسط دوفر در نظر گرفته میشود) تشخیص دهد. هر کلمه با استفاده از ۹۶ بیت در حافظه سیستم ذخیره میگردد و هر فرمان سیستم نیاز به دو کلمه دارد. بعنوان مثال، فرمان "مدای پنج" یا "کانال سه" به ترتیب سطح صدا را روی درجه پنج و برنامه تلویزیون را روی کانال سه تنظیم میکند. بنابر نظر پژوهشگران شرکت سانپو در ژاپن، سیستم های کنترل ارزان قیمت و با واکنش سریع از این نوع قابل ساخت است ولی عملکرد مطمئن سیستم در محیط های پارازیت دار (ناشی از خود تلویزیون و یا مثلاً جاروی برقی) هنوز با مشکلاتی مواجه است.

متخصصین عقیده دارند که شناسائی صحیح ۹۵ درصد از کلمات با استفاده از یک میکروفون دستی بیسیم امکان پذیر است. ولی هنگامیکه از یک میکروفون تعبیه شده