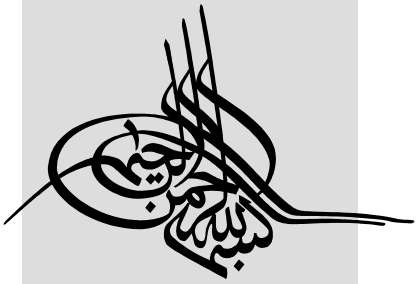


سری کتابهای کمک آموزشی کارشناسی ارشد

معماری کامپیوتر

مجموعه مهندسی کامپیوتر و IT

مؤلف: مهدی منوچهرزاده

منوچهرزاده، مهدی
کامپیوتر/ کارشناسی ارشد رشته مهندسی کامپیوتر و IT / مهدی منوچهرزاده
مهر سبحان ۱٤۰۱ معماری
۲۱۵ ص: شرح ـ نکته (آمادگی آزمون کارشناسی ارشد مهندسی کامپیوتر و IT)
ISBN: 978-964-164-988-5
فهرستنویسی بر اساس اطلاعات فیپا.
فارسی ـ چاپ سوم
۱- معماری کامپیوتر ۲- آزمونها و تمرینها (عالی)
۳- آزمون دوره‌های تحصیلات تکمیلی ٤- دانشگاهها و مدارس عالی ـ ایران ـ آزمونها
مهدی منوچهرزاده
ج ـ عنوان
۱۳۹۲ ٦ م ۸۲۷ م /LB۲۳۵۳ ۳۷۸/۱٦٦٤
کتابخانه ملی ایران ۳۲۲۲۰۹۲

☑ نام کتاب: معماری کامپیوتر
☑ مولف: مهدی منوچهرزاده
☑ ناشر: مهر سبحان
☑ نوبت و تاریخ چاپ سوم/ ۱٤۰۱
☑ تیراژ: ۳۰۰۰ نسخه
☑ قیمت: ۲/۲۹۰/۰۰۰
☑ شابک: ISBN ۹۷۸-۹٦٤-۱٦٤-۹۸۸-۵

انتشارات مهر سبحان: خیابان ولیعصر، بالاتر از تقاطع مطهری، روبروی قنادی هتل بزرگ تهران، جنب بانک ملی، پلاک ۲۰۵۰ تلفن: ٤- ۸۸۱۰۰۱۱۳

کلیه حقوق مادی و معنوی این اثر متعلق به موسسه آموزش عالی آزاد ماهان می‌باشد. هر گونه اقتباس و کپی‌برداری از این اثر بدون اخذ مجوز پیگرد قانونی دارد.

مقدمه ناشر

آیا آنانکه می‌دانند با آنانکه نمی‌دانند برابرند؟ (قرآن کریم)

پس از حمد و سپاس و ستایش به درگاه بی همتای احدیت و درود بر محمد مصطفی، عالی نمونه بشریت که در تاریک دور تاریخ، بنا به فرمان نافذ صمدیت از میان مردمی برخاست که خود بودند در پست‌ترین حد توحش و ضلال و بربریت و آنگاه با قوانین شامل خویش هم ایشان را راهبری نمود و رهانید از بدویت و استعانت جوییم از قرآن کریم، کتابی که هست جاودانه و بی‌نقص تا ابدیت.

کتابی که در دست دارید آخرین ویرایش از مجموعه کتب خودآموز مؤسسه آموزش عالی آزاد ماهان است که برمبنای خلاصه درس و تأکید بر نکات مهم و کلیدی و تنوع پرسش‌های چهار گزینه‌ای جمع‌آوری شده است. در این ویرایش ضمن توجه کامل به آخرین تغییرات در سرفصل‌های تعیین شده جهت آزمون‌های ارشد تلاش گردیده است که مطالب از منابع مختلف معتبر و مورد تأکید طراحان ارشد با ذکر مثال‌های متعدد بصورت پرسش‌های چهار گزینه‌ای با کلید و در صورت لزوم تشریح کامل ارائه گردد تا دانشجویان گرامی را از مراجعه به سایر منابع مشابه بی نیاز نماید.

لازم به ذکر است شرکت در آزمون‌های آزمایشی ماهان که در جامعه آماری گسترده و در سطح کشور برگزار می‌گردد می‌تواند محک جدی برای عزیزان دانشجو باشد تا نقاط ضعف احتمالی خود را بیابند و با مرور مجدد مطالب این کتاب، آنها را برطرف سازند که تجربه سال‌های مختلف موکد این مسیر به عنوان مطمئن‌ترین راه برای موفقیت می‌باشد.

لازم به ذکر است از پورتال ماهان به آدرس www.mahanportal.ir می‌توانید خدمات پشتیبانی را دریافت دارید.

و نیز بر خود می‌بالیم که همه ساله میزان تطبیق مطالب این کتاب با سؤالات آزمون‌های ارشد- که از شاخصه‌های مهم ارزیابی کیفی این کتاب‌ها می‌باشد- ما را در محضر شما سربلند می‌نماید.

در خاتمه بر خود واجب می‌دانیم که از همه اساتید بزرگوار و دانشجویان ارجمند از سراسر کشور و حتی خارج از کشور و همه همکاران گرامی که با ارائه نقطه نظرات سازنده خود ما را در پربارتر کردن ویرایش جدید این کتاب یاری نمودند سپاسگزاری نموده و به پاس تلاش‌های بی چشمداشت، این کتاب را به محضرشان تقدیم نماییم.

مؤسسه آموزش عالی آزاد ماهان

معاونت آموزش

فهرست مطالب

عنوان **صفحه**

فصل اول: آشنایی با معماری کامپیوتر و نمایش اطلاعات در سیستم ---------- ۹
مدل انتزاعی کامپیوتر ---------- ۱۵
روش طراحی ---------- ۱۵
گراف مدل سازی سیستم ---------- ۱۵
سطوح طراحی ---------- ۱۸
طراحی سلسله مراتبی ---------- ۱۸
مالتی پلکسر یا تسهیم کننده ---------- ۱۹
رمزگشا ---------- ۲۲
انواع فلیپ فلاپ‌ها ---------- ۲۸
ثبات‌ها ---------- ۳۰
طراحی پردازنده ---------- ۴۱
سؤالات چهار گزینه‌ای فصل اول ---------- ۵۰
پاسخ سؤالات چهار گزینه‌ای فصل اول ---------- ۵۱

فصل دوم: طراحی واحد پردازنده و اعمال حسابی ---------- ۵۳
مراحل اجرایی یک دستورالعمل ---------- ۵۵
موارد تعیین‌شده در طراحی مجموعه دستورالعمل ---------- ۵۶
روش‌های آدرس دهی ---------- ۵۶
تعداد آدرس‌ها ---------- ۶۰
دستورالعمل‌های حسابی ---------- ۶۱
جمع و تفریق حسابی ---------- ۶۱
سؤالات چهار گزینه‌ای فصل دوم ---------- ۸۴
پاسخ سؤالات چهار گزینه‌ای فصل دوم ---------- ۹۰

فصل سوم: واحد کنترل ---------- ۹۳
توالی دستورالعمل‌ها ---------- ۹۵
تفسیر دستورالعمل ---------- ۹۹
مراحل طراحی واحد کنترل ---------- ۱۰۰
ریز عملیات منطقی ---------- ۱۰۴

روش‌های پیاده‌سازی واحد کنترل ---------- ۱۰۷
طرح اولیه wilkes ---------- ۱۱۰
سؤالات چهار گزینه‌ای فصل سوم ---------- ۱۱۸
پاسخ سؤالات چهار گزینه‌ای فصل سوم ---------- ۱۲۱

فصل چهارم: سازمان حافظه ---------- ۱۲۳
دسته‌بندی حافظه‌ها از نظر انواع خواندن ---------- ۱۲۶
مراحل خواندن از حافظه ---------- ۱۲۶
مراحل نوشتن در حافظه ---------- ۱۲۶
دسته‌بندی حافظه‌ها از نظر منبع تغذیه ---------- ۱۲۷
حافظه‌های مدار مجمتع نیمه هادی ---------- ۱۲۷
سازمان حافظه ---------- ۱۲۹
حافظه‌های کمکی ---------- ۱۳۰
دیسک مغناطیسی ---------- ۱۳۱
حافظه انجمنی ---------- ۱۳۲
جابه‌جا نمودن صفحات ---------- ۱۳۷
حفاظت حافظه ---------- ۱۳۸
سؤالات چهار گزینه‌ای فصل چهارم ---------- ۱۳۹
پاسخ سؤالات چهار گزینه‌ای فصل چهارم ---------- ۱۴۱

فصل پنجم: طراحی کامپیوتر پایه ---------- ۱۴۳
وظیفه و ساختار ---------- ۱۴۵
کدهای دستورالعمل‌ها ---------- ۱۴۵
ثبات‌های کامپیوتر پایه ---------- ۱۴۶
دستورالعمل‌های کامپیوتر ---------- ۱۴۷
زمانبندی و کنترل ---------- ۱۴۸
انواع سازمان کنترل ---------- ۱۴۸
سیکل دستورالعمل ---------- ۱۴۹
فلوچارت کنترل ---------- ۱۵۵
دستورات ورودی خروجی ---------- ۱۵۶
کنترل ثبات‌ها و حافظه ---------- ۱۶۰
کنترل فلیپ فلاپ‌های منفرد ---------- ۱۶۱

فصل ششم: دستگاه‌های ورودی خروجی و وقفه‌ها ---------- ۱۶۵
ورودی خروجی برنامه‌ریزی شده ---------- ۱۶۷
آدرس‌دهی ورودی خروجی ---------- ۱۶۷
دسترسی مستقیم و وقفه ---------- ۱۶۹
دسترسی مستقیم به حافظه ---------- ۱۶۹

ورودی خروجی مبتنی بر وقفه ---------- ۱۷۱
سرشماری ---------- ۱۷۲
وقفه‌های برداری ---------- ۱۷۳
سؤالات چهار گزینه‌ای فصل ششم ---------- ۱۷۶
پاسخ سؤالات چهار گزینه‌ای فصل ششم ---------- ۱۷۸
مجموعه سوالات کنکور سال‌های۸۹-۹۳ ---------- ۱۷۹
منابع ---------- ۲۱۵

فصل اول

آشنایی با معماری کامپیوتر و نمایش اطلاعات در کامپیوتر

عناوین اصلی

- مدل انتزاعی کامپیوتر
- روش طراحی
- گراف مدل سازی سیستم
- سطوح طراحی
- طراحی سلسله مراتبی
- مالتی پلکسر یا تسهیم کننده
- رمزگشا
- انواع فلیپ فلاپ‌ها
- ثبات‌ها و کاربرد آنها
- طراحی پردازنده

فصل اول

آشنایی با معماری کامپیوتر و نمایش اطلاعات در کامپیوتر

این درس در رابطه با ساختار و چگونگی طراحی کامپیوترهای دیجیتالی می‌باشد. این مطلب به "معماری کامپیوتر" معروف است که شامل معماری مجموعه دستورالعمل و سازماندهی سخت‌افزاری می‌باشد.

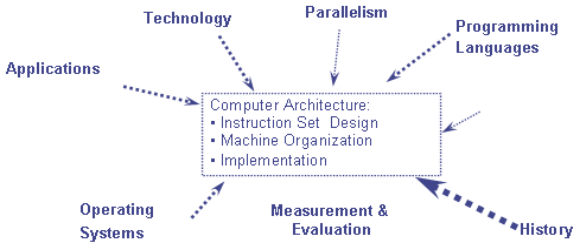
معماری کامپیوتر اولین بار توسط Fred Brooks ابداع گردید. و تعاریف زیر از آن وجود دارد:

- معماری کامپیوتر، یعنی کامپیوتر از دید کاربر(Amdhal et al, 1964)
- "ما توسط معماری، ساختار واحدهای تشکیل‌دهنده یک سیستم کامپیوتری را هدفمند می‌نماییم. (Stone, H 1987)
- معماری یک کامپیوتر عبارتست ازمحیط یا فضای بین ماشین و نرم‌افزار (Andris Pages IBM 360/370 Architect)
- معماری کامپیوتر شامل بخش‌های زیر می‌باشد:
- **ساختار:** نظم و ترتیب دادن به بخش‌های ثابت (نقشه)
- **سازماندهی:** فعل و انفعال پویای این بخش‌ها و مدیریت آنها
- **پیاده‌سازی:** طراحی کردن بلوک بخش‌های دارای هدف خاص
- **ارزیابی کارایی:** مطالعه رفتار سیستم (Decorative Treatment)

معماری از دیدگاه یک معمار عبارت است از:

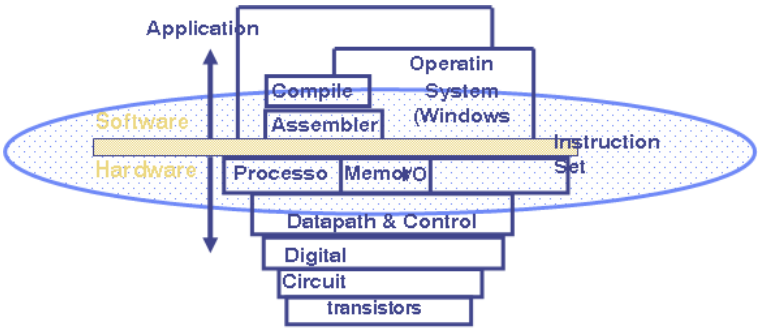
- پیاده‌سازی
- سازماندهی: منظر سطح بالا
- سیستم حافظه
- ساختار گذرگاه (bus)
- طراحی داخلی پردازنده
- سخت‌افزار
- طراحی منطقی
- تکنولوژی بسته بندی(Packaging)
- معماری مجموعه دستورالعمل

عوامل مهم در معماری کامپیوتر در شکل زیر نشان داده شده است:

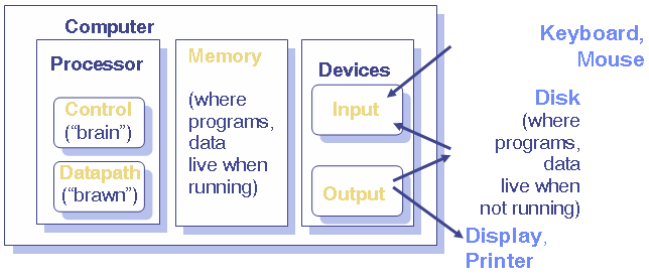


- هم سخت‌افزار و هم نرم‌افزار مرکب از لایه‌های سلسله مراتبی هستند. با هر لایه، جزئیاتی از دید سطح بالاتر مخفی می‌شوند. این اصل تجرید، سبب می‌شود که طراحان سخت‌افزار و نرم‌افزار از عهدهٔ پیچیدگی سیستم‌های کامپیوتری برآیند.

در شکل زیر سطوح کامپیوتر نشان داده شده است که سطح معماری کامپیوتر در بین این سطوح نشان داده شده است:



بخش‌های مختلف یک سیستم کامپیوتری در شکل زیر نشان داده شده است:



سرعت رشد تکنولوژی بخش‌هایی از کامپیوتر در سال‌های مختلف به صورت زیر می‌باشد:

- پردازنده

هر یکسال و نیم، ۲ برابر در سرعت (از سال ۱۹۸۵). کارایی ۱۰۰ برابر، در دهه گذشته

- حافظه

ظرفیت DRAM: ۲برابر در هر دو سال (از سال ۱۹۹۶)

بهبود اندازه × ۶۴ در دهه گذشته

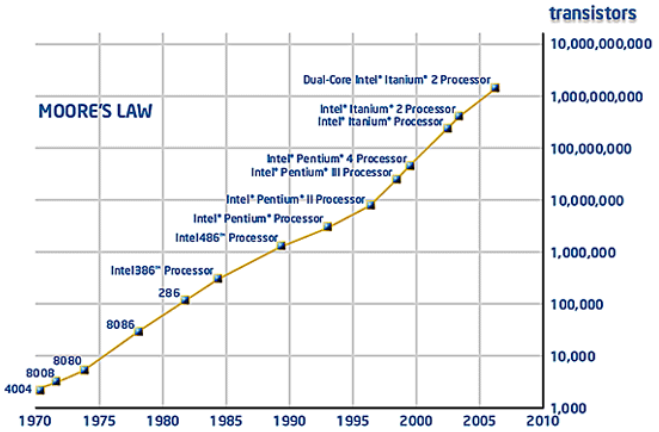
- دیسک

ظرفیت: ۲برابر در هر سال(از سال ۱۹۹۷)

بهبود اندازه × ۲۵۰ در دهه گذشته

- ریزپردازنده

قانون مور: هر ۱/۵ الی ۲ سال پیچیدگی ترانزیستور/تراشه ۲ برابر می‌شود. پیچیدگی ریز پردازنده در شکل زیر نشان داده شده است:



مشخصات پردازنده‌های شرکت اینتل در جدول زیر آمده است:

۱) پردازنده Intel 486™ DX • Design 1986 – 1989 • 25 MHz, 33 MHz • 1.2 M transistors • 1.0 micron • 5 stage pipeline • Unified 8 KByte code/data cache (write-through) • First IA-32 processor capable of executing 1 instruction per clock cycle	**۲) پردازنده Pentium Design 1989 – 1993** • 60 MHz, 66 MHz • 3.1 M transistors • 0.8 micron • 5 stage pipeline • 8 KByte instruction and 8 KByte data caches (writeback) • Branch predictor • Pipelined floating point • First superscalar IA-32: capable of executing 2 instructions per clock
۳) پردازنده Pentium II • Design 1995 – 1997 • 233 MHz, 266 MHz, 300 MHz • 7.5 M transistors • 0.35 micron • 16 KByte L1I, 16 KByte L1D, 512 KByte off-die L2 • First compaction of P6 microarchitecture	**۴) پردازنده Pentium III(Katmai)** • Introduced: 1999 • 450 MHz, 500 MHz, 533 MHz, 600MHz • 9.5 M transistors • 0.25 micron • 16 KByte L1I, 16 KByte L1D, 512 KByte off-chip L2 • Addition of SSE instructions.
۵) پردازنده Pentium III(Coppermine) • Introduced: 1999 • 500MHz … 1133MHz • 28 M transistors • 0.18 micron • 16 KByte L1I, 16 KByte L1D, 256KByte on-chip L2 • Integrate L2 cache on chip, It topped out at 1GHz.	**۶) پردازنده Pentium IV** • Introduced: 2000 • 1.3GHz … 2GHz … 3.4GHz • 42M … 55M … 125 M transistors • 0.18 … 0.13 … 0.09 micron • Latest one: 16 KByte L1I, 16 KByte L1D, 1M on-chip L2 • Very high clock speed and SSE performance

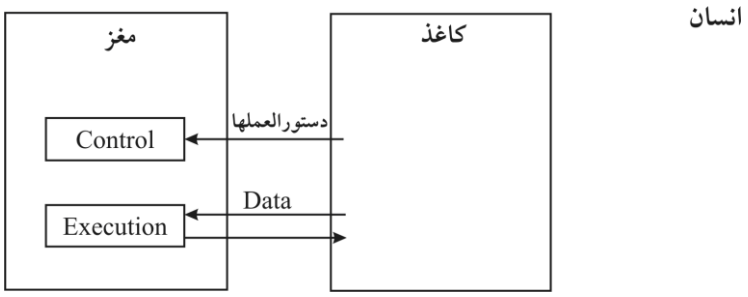
۷) پردازنده Intel® Itanium® • Design 1993 – 2000 • 733 MHz, 800 MHz • 25 M transistors • 0.18 micron • 3 levels of cache • 16 KByte L1I, 16 KByte L1D • 96 KByte L2 • 4 MByte off-die L3 • Superscalar degree 6, in-order machine • First implementation of 64-bit Itanium architecture	**۸) پردازنده Intel® Itanium 2®** • Introduced: 2002 • 1GHz • 221 M transistors • 0.18 micron • 3 levels of cache • 32 KByte I&D L1 • 256 KByte L2 • integrated 1.5MByte L3 • Based on EPIC architecture • Enhanced Machine Check Architecture (MCA) with extensive Error Correcting Code (ECC)

طبیعت کار کامپیوترها

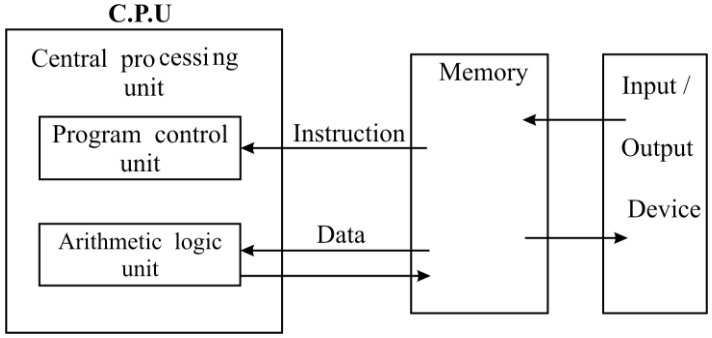
۱- سرعت کار کامپیوتر محدود است.

۲- انسان جایزالخطاست به طوریکه نتیجه حاصل از محاسبات دستی پیچیده نامطمئن می‌باشد زیرا با توجه به منابع خطای انسان (حواس‌پرتی، بی دقتی، خستگی و.....) ممکن است اشتباهاتی در محاسبات رخ دهد ولی ماشین از منابع خطای انسان متأثر نمی‌شود و بنابراین نتایج دور از خطا خواهد بود.

مؤلفه‌های لازم برای تفکر انسان در شکل زیر نشان داده شده است:



مؤلفه‌های اصلی کامپیوتر نیز مشابه با مؤلفه‌های مغز انسان به صورت زیر می‌باشد:



تفاوت اصلی بین ماشین و انسان در نحوه نمایش اطلاعات می‌باشد بنابراین برای تبدیل اطلاعات از زبان ماشین به زبان انسان و بالعکس وسیله‌ای لازم است که I / O Device نامیده می‌شود.

بنابراین مؤلفه‌های لازم برای کامپیوتر عبارتند از:

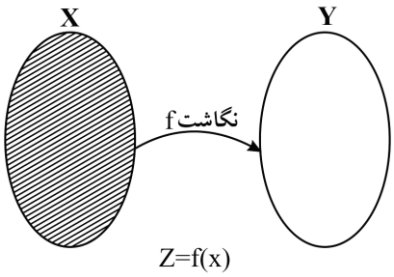
۱- **پردازنده:** که قادر به تفسیر و اجرای برنامه است.

۲- **حافظه:** برای ذخیره نمودن برنامه‌ها و داده‌ها

۳- وسیله‌ای برای انتقال اطلاعات بین حافظه و پردازنده و بین کامپیوتر و دنیای خارج.

مدل انتزاعی کامپیوتر

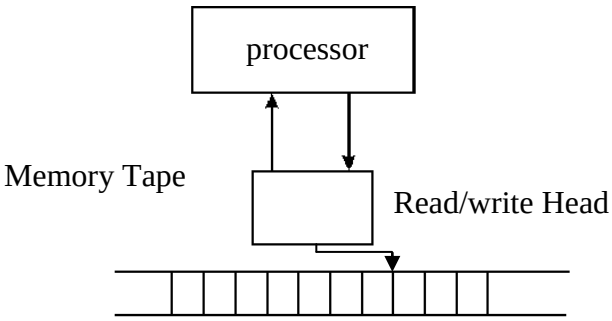
منظور از مدل انتزاعی، مدلی بدون در نظر گرفتن جزئیات برای درک بهتر کارکرد کلی کامپیوتر است. محاسبه را می‌توان به عنوان ارزیابی تابعی مانند $f(x)$ تلقی نمود که در آن x ورودی و $z = f(x)$ خروجی است.



برای اینکه بتوان $f(x)$ را به وسیله کامپیوتر ارزیابی نمود. باید قادر باشیم آن را توسط مجموعه‌ای از دستورالعمل‌ها مانند: $f_1, f_2, \ldots f_n$ به صورت زیر بیان نمود:

$$y_1 = f_1(x) \ , \ y_2 = f_2(y_1), \ldots, y_n = z_n = f_n(y_{n-1})$$

چنین ماشینی باید قادر باشد محاسبه را طی مراحلی متناهی و محدود انجام دهد. مدل انتزاعی به وسیله ریاضیدان انگلیسی آلن تورینگ (Alen Turing) معرفی شده است که همه مؤلفه‌های لازم به وسیله کامپیوتر را در بردارد.



R / W Head قادر است علامت‌های روی نوار را بخواند و آنها را به علامات دیگری جایگزین کند و یا یک خانه به طرف چپ یا راست تغییر مکان دهد.

روش طراحی (Design Methodology)

کامپیوتر دیجیتال یک سیستم است. سیستم مجموعه‌ای است از اجزاء که با اتصال منطقی به یکدیگر یک شئ جدید با وظیفه و هدف مشخص به وجود می‌آورد. ولی در اینجا منظور ما سیستم پردازش اطلاعات می‌باشد که وظیفه آن تبدیل ورودی A به خروجی B می‌باشد و این تبدیل را تابع f از A به B نشان داد.

محاسبات $f : A \to B$

$\forall a \in A \qquad \exists b \in B \qquad ; b = f(a)$

گراف مدلسازی سیستم (System Graph Modeling)

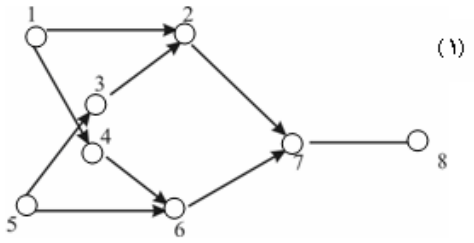
گراف یک سیستم دیجیتال شامل دو مؤلفه است:

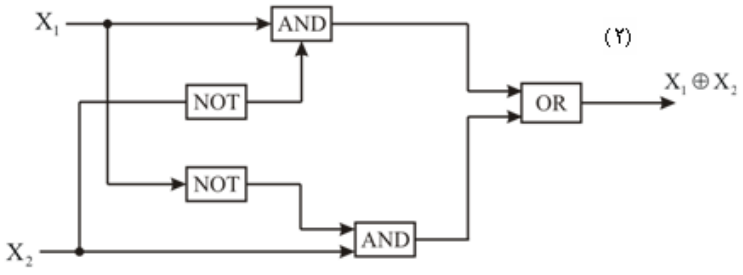
۱- مجموعه‌ای از اجزاء پردازش (C)

۲- مجموعه‌ای از سیگنال‌ها (S) که وظیفه انتقال اطلاعات بین اجزاء پردازش را به عهده دارد.

در مدلسازی سیستم اجزای پردازش C رئوس و سیگنال‌ها S یال‌ها می‌باشند و گراف حاصل را نمودار بلوکی (Block Diagram) گویند.

مثال: در شکل زیر ساختار xor در یک گراف با ۸ رأس و ۹ یال نشان داده شده است:





نمودار بلوکی X - OR

ساختار سیستم به عنوان گراف انتزاعی است که هیچ اطلاعی درباره وظیفه سیستم بیان نمی‌کند مثلاً شکل (۱) ساختار سیستم شکل (۲) را نشان می‌دهد، بنابراین توصیف رفتاری اجزاء تشکیل‌دهنده سیستم را بیان می‌کند و اتصال منطقی بین آنها را معین می‌کند، ولی توصیف رفتاری به شخص امکان می‌دهد که برای هر سیگنال ورودی A به سیستم سیگنال خروجی $f(a)=b$ را معین کند و f را به عنوان رفتار سیستم در نظر می‌گیریم.

برای نمایش رفتار سیستم می‌توان ازجدول صحت استفاده کرد.

مثال: جدول صحت XOR به صورت مقابل می‌باشد:

ورودی‌ها		$b=f(a)$
x_1	x_2	$x_1 \oplus x_2$
۰	۰	۰
۰	۱	۱
۱	۰	۰
۱	۱	۰

(۳)

حال می‌توانیم با در دست داشتن توصیف رفتاری و اجزای تشکیل‌دهنده سیستم اجزاء را طوری به یکدیگر متصل کنیم که با

حداقل هزینه رفتار موردنظر را انجام دهد. باید توجه کرد که توصیف رفتاری و ساختاری شکل (۱) و (۳) مستقل از یکدیگرند در حالیکه نمودار بلوکی هم توصیف ساختاری و هم رفتاری است. از شکل (۲) می‌توان شکل (۱) و (۳) را بدست آورد.

داده و کنترل (Data and Control)

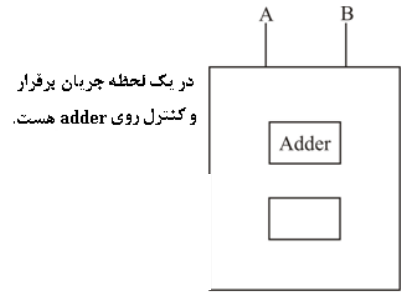
سیستم رقمی به دو جزء تقسیم می‌شود:

۱- واحدپردازش داده‌ها (Data Processing Unit)

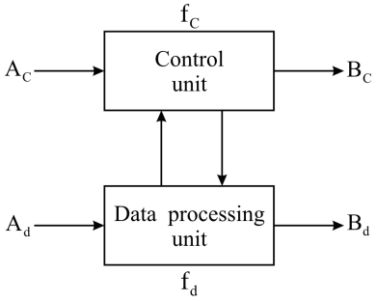
۲- واحد کنترل (Control Unit)

مجموعه اجزاء و مسیرهای پیموده شده به وسیلهٔ داده را واحد پردازش داده‌ها گویند و سایر قسمت‌ها واحد کنترل نامیده می‌شود. عملیات انجام داده شده در واحد پردازش و مسیرهای پیموده شده به وسیلهٔ داده‌ها در هر لحظه از زمان به وسیلهٔ واحد کنترل مشخص می‌گردد.

تمایز کنترل و داده را می‌توان با استفاده از تابع رفتاری ماشین و با تجزیهٔ A و B و f به دو قسمت کنترل و داده انجام داد.



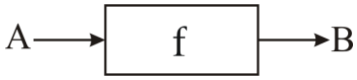
در صورتیکه دو تابع کنترل و داده را به صورت جداگانه پیاده‌سازی کنیم نمودار دو بلوکی زیر بدست می‌آید.



واحد کنترل را می‌توان به دو صورت طراحی نمود:

۱. اگر f_c ثابت و دایمی باشد واحد کنترل را سیم بندی شده (Hard wired) گویند.

۲. اگر f_c به کمک ذخیره نمودن اطلاعات کنترل درحافظه انجام گیرد واحد کنترل را زیر برنامه‌سازی شده (Microprogrammed) گویند.



روش‌های توصیفی: اجزای سیستم در نمودار بلوکی به صورت فوق نشان داده می‌شود که در آن ساختار داخلی مشخص نیست ولی رفتار آن به طور ناقص یا کامل مشخص است. اجزاء یا مؤلفه‌های سیستم در واقع ماشین با حالت‌های متناهی است که در آن ورودی A به دو جزء تقسیم می‌شود.

۱. سیگنال‌های ورودی اصلی X که به وسیلهٔ منابع خارجی فراهم می‌شود.

۲. حالات درونی Y که اطلاعات ذخیره شده درونی را تشکیل می‌دهد.

همچنین خروجی B به دو جزء زیر تقسیم می‌شود.

۱. سیگنال‌های خروجی اصلی Z

۲. حالات جدید درونی y

$$f : x \times y \rightarrow y \times z$$

$$\forall < x,y > \in x \times y \qquad \exists < y,z > \in y \times z \quad \ni < y,z > = f < x_1 y >$$

و این تابع را می‌توان به وسیلهٔ State Table نیز نشان داد که در آن سطرها نشان‌دهندهٔ حالات و ستون‌ها نشاندهنده ورودی‌ها می‌باشد.
روش دیگر برای نشان دادن رفتار سیستم استفاده ازState Transition می‌باشد، که معادلات آن به صورت زیر می‌باشد:

$$< y_1, z_1 > \leftarrow f < x_0, y_0 >$$
$$< y_2, z_2 > \leftarrow f < x_1, y_1 >$$
$$\vdots \qquad\qquad \vdots$$

که با توصیف رفتاری ماشین تورینگ یکسان است.
هریک از معادلات فوق یک حکم (Statement) نامیده می‌شود که به دو صورت زیر تفسیر می‌شود:

۱- خبری: بیانی از آنچه که ماشین انجام می‌دهد.
۲- دستوری: بیانی از آنچه ماشین باید انجام دهد.

RTL (Register Transfer Language): مجموعه‌ای از احکام دستوری است که رفتار مورد نظر ماشین را مشخص می‌کند و مقدمه یک طراحی می‌باشد.

سطوح طراحی

طراحی کامپیوترهای دیجیتال در سه سطح انجام می‌شود.

Design level	Components	Information Unit	Time Unit
Processor LSI (Larg Scale Integrated Circuit)	CPU, I/O Devices. I/O processors, Memories	Block of Data	10^{-3} _ 10^{3} S
Register MSI (Medium Scale Integrated circuit)	Register, Combinational simple Sequential ckt	Word	10^{-9} _ 10^{-6} S
Gate SSI (small scale Integrated circuit)	Logical Gates,FFs	Bit	10^{-10} _ 10^{-8} S

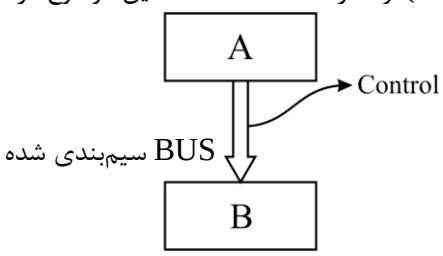
طراحی سلسله مراتبی (Hierarchy Design)

برای طراحی بالا به پایین یک سیستم مراحل زیر را انجام می‌دهیم:

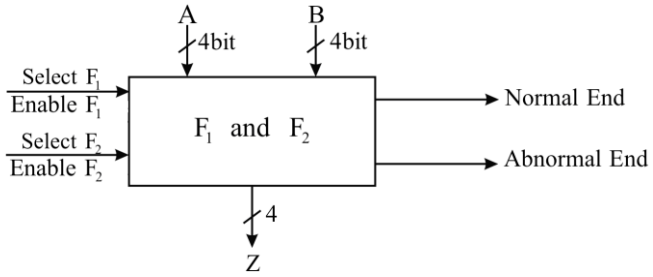
۱- ساختار سیستم را در سطح پردازنده‌ها مشخص می‌کنیم.
۲- ساختار هریک از مؤلفه‌های مرحله اول را در سطح ثباتها مشخص می‌کنیم.
۳- ساختار هریک از مؤلفه‌های مرحله دوم را در سطح گیت‌ها مشخص می‌کنیم.

سطح ثباتی (Register level)

واحد اطلاعاتی مجموعه‌ای از بیت‌ها (word) و گذرگاه (BUS) است. این موضوع در شکل زیر نشان داده شده است.



نمودار بلوکی مولفه‌ای نیز در سطح ثبات به صورت زیر است:

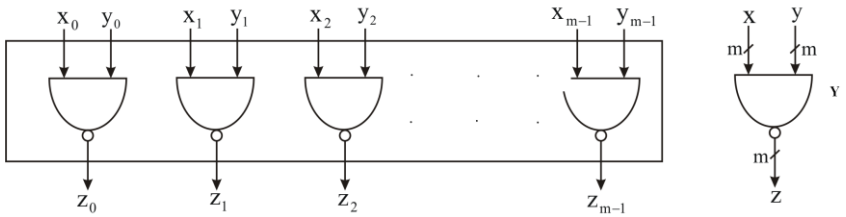


گیت کلمه‌ای (Word Gate)

فرض کنید $X=(x_0,x_1,.....,x_{m-1})$ و $Y=(y_0,y_1,.....,y_{m-1})$ دو کلمه m بیتی باشند و f عضوی از مجموعهٔ عمل‌گرهای $\{AND, OR, NOT, NAND, NOR, XOR\}$ باشد، آنگاه $z=f(x,y)$ اگر $z_i=f(x_i,y_i)$ برای $i=0,1,.....,m-1$ و این تعمیم جبر بول دو ارزشی $B=\{0,1\}$ از تابع $f:B^n\to B$ به تابع $f(B^m)^n\to B^m$ می‌باشد.

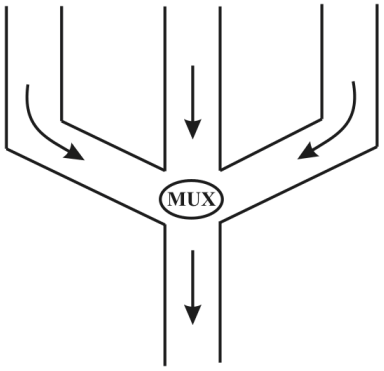
به عنوان مثال $z=\overline{xy}$ نشاندهنده یک گیت NAND با ورودی m – بیتی به صورت زیر خواهد بود.

$$(z_0,z_1,.....,z'_{m-1})=(\overline{x_0y_0}\ ,\ \overline{x_1y_1},.....,\overline{x_{m-1}y_{m-1}})$$



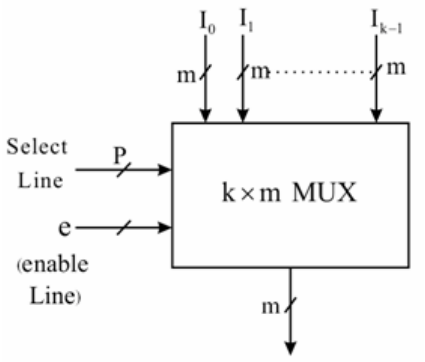
مالتی پلکسر یا تسهیم‌کننده (Multiplexor)

مدار تسهیم‌کننده یک مدار ترکیبی است، بنابراین از حافظه استفاده نمی‌کند. این مدار از بین چندین منبع ورودی یکی را انتخاب و آن را به خروجی می‌فرستد. تصویری از عملکرد این مدار در شکل زیر نشان داده شده است.



گزینش منابع ورودی به وسیلهٔ خطوط انتخاب صورت می‌گیرد. اگر 2^n منبع ورودی، آنگاه به n خط انتخاب نیاز است. اگر تسهیم‌کننده دارای k منبع ورودی و هر منبع شامل اطلاعات m بیتی باشد در آن‌صورت برای این اطلاعات به یک تسهیم‌کننده $k\times m$ بیتی احتیاج داریم که در آن k تعداد ورودی‌ها و m تعداد بیت‌های هر منبع می‌باشد. k را می‌توان از رابطه $k=2^p$ محاسبه کرد که در آن p تعداد خطوط انتخاب می‌باشد.

$$K=2^P$$



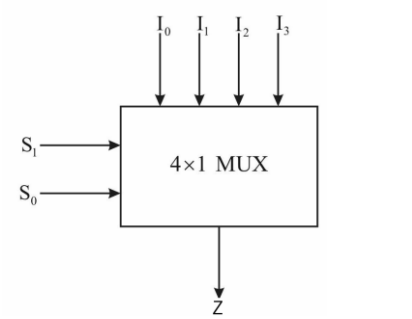
نمودار بلوکی

مثال: یک تسهیم‌کننده ۱×۴طراحی کنید.

حل: تعداد خطوط انتخاب در این مالتی‌پلکسر را می‌توان به صورت زیر محاسبه نمود:

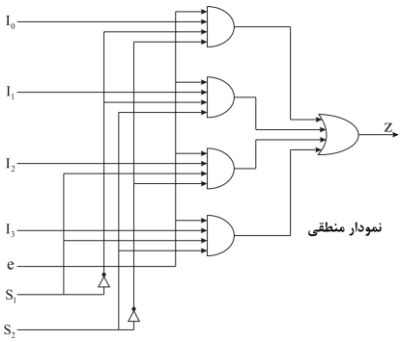
$$K = ۴ = ۲^{۲} \quad \Rightarrow P = ۲$$
$$K = ۲^{P}$$

بنابراین نمودار بلوکی و جدول صحت این مالتی‌پلکسر به صورت زیر خواهد بود:



$S_۰$	$S_۱$	Z
۰	۰	$I_۰$
۰	۱	$I_۱$
۱	۰	$I_۲$
۱	۱	$I_۳$

با توجه به جدول صحت، می‌توان این مالتی‌پلکسر را به صورت زیر طراحی نمود:



نمودار منطقی

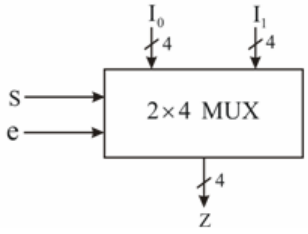
نکته: می‌توان از خط enable برای افزایش ساخت مالتی‌پلکسرهای بزرگتر با به هم وصل کردن مالتی‌پلکسرهای کوچک‌تر استفاده نمود.

مثال: با استفاده از مالتی‌پلکسرهای ۱×۲ یک مالتی پلکسر ۴×۲ طرح کنید.

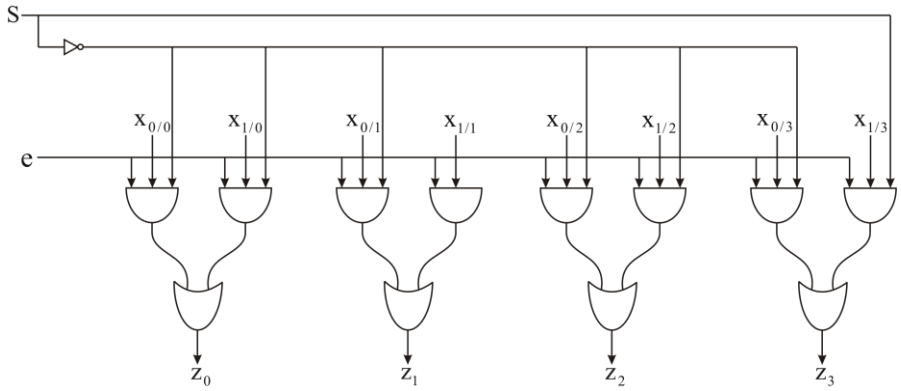
حل:

e	S	Z
۰	–	۰
۱	۰	$I_۰$
۱	۱	$I_۱$

$$K = ۲ = ۲^{۱} \Rightarrow P = ۱$$



و چون خروجی چهاربیتی است بنابراین به چهار عدد تسهیم‌کننده۱×۲ احتیاج داریم: اندیس xها در شکل زیر بدین گونه می‌باشد که $x_{i/j}$ بیانگر منبع i ام بیت j ام می‌باشد.



$x_{\circ} = (x_{\circ/\circ},\ x_{\circ/۱},\ x_{\circ/۲},\ x_{\circ/۲})$

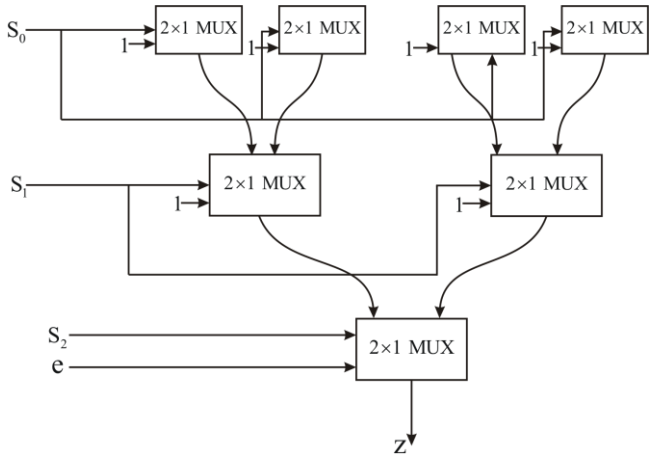
$x_{۱} = (x_{۱/\circ},\ x_{۱/۱},\ x_{۱/۲},\ x_{۱/۳})$

اتصال درختی تسهیم‌کننده‌ها

با اتصال درختی تسهیم‌کننده‌های k ورودی می‌توان تسهیم‌کننده‌های با تعداد منابع ورودی بیشتر از K به وجود آورد. در این روش بایستی خطوط انتخاب متناظر با هر سطح به تسهیم‌کنندگان آن سطح به طور مشترک وصل گردند.

مثال: با استفاده مالتی‌پلکسرهای ۱×۲ یک مالتی پلکسر۱×۸ طرح کنید.

حل: با توجه به اینکه k برابر ۲ است و $۸ = ۲^q$ که در آن q تعداد سطوح می‌باشد، نتیجه می‌گیریم که q برابر سه خواهد بود. شکل زیر این طراحی را نشان می‌دهد:



اتصال درختی MUX جهت ازدیاد منابع ورودی

$S_۲$	$S_۱$	$S_\circ$	خروجی
$۲^۲$	$۲^۲$	۲°	Z
۰	۰	۰	$X_\circ$
۰	۰	۱	$X_۱$
۰	۱	۰	$X_۲$
۰	۱	۱	$X_۳$
۱	۰	۰	$X_۴$
۱	۰	۱	$X_۵$
۱	۱	۰	$X_۶$
۱	۱	۱	$X_۷$

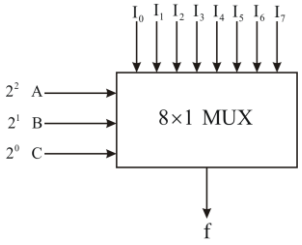
نکته: کاربردهای رایج تسهیم‌کننده عبارتند از:

۱. برای هدایت اطلاعات از بین چندین منبع ورودی به یک مقصد خروجی مشترک
۲. برای طراحی گذرگاهها
۳. برای پیاده‌سازی توابع بولی و ترکیبی

نکته: هر تابع n متغیر را می‌توان به وسیلهٔ یک $mux\,۲^n \times ۱$ پیاده‌سازی داد به شرط آنکه متغیرهای تابع را به خطوط انتخاب تسهیم‌کننده متصل سازیم.

مثال: تابع $f(A,B,C)=\Sigma m(۱,۳,۵,۶)$ را به وسیله مالتی پلکسر پیاده‌سازی کنید.

حل: از آنجاییکه $۸ = ۲^۳$، بنابراین می‌توان این تابع را توسط یک $Mux\,۸\times ۱$ پیاده‌سازی نمود. تعداد خطوط این مالتی پلکسر برابر سه است. دیاگرام بلوکی و جدول صحت این تابع به صورت زیر می‌باشد:

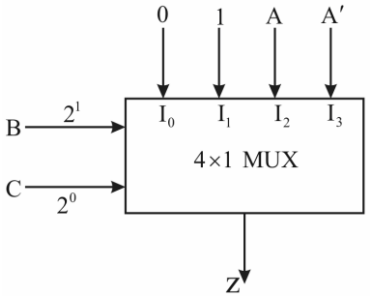


	A	B	C	f(A,B,C)
$m_۰$	۰	۰	۰	۰
$m_۱$	۰	۰	۱	۱
$m_۲$	۰	۱	۰	۰
$m_۳$	۰	۱	۱	۱
$m_۴$	۱	۰	۰	۰
$m_۵$	۱	۰	۱	۱
$m_۶$	۱	۱	۰	۱
$m_۷$	۱	۱	۱	۰

نکته: هر تابع n متغیر را می‌توان به وسیلهٔ یک $Mux\,۲^{n-۱}\times ۱$ پیاده‌سازی نمود، به شرط آنکه $(n-۱)$ تا از متغیرهای تابع را به خطوط انتخاب وصل کنیم و از متغیر دیگر در ورودیهای MUX استفاده کنیم.

مثال: تابع مثال قبل را به این روش پیاده‌سازی کنید.

حل:



جدول این تابع در زیر آمده است. در این جدول دور مینترم‌های تابع، دایره می‌کشیم.

	I_0	I_1	I_2	I_3
A′	0	①	2	③
A	4	⑤	⑥	7
	0	1	A	A′

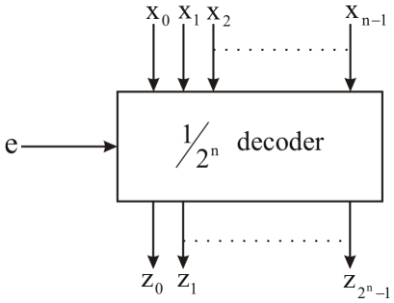
رمز گشا (Decoder)

رمز گشا مداری ترکیبی است که اطلاعات واقع در n خط ورودی را حداکثر به ۲^n خط خروجی رمزگشایی می‌کند. (یعنی هر خروجی متناظر با یک مینترم می‌باشد.) و در صورتیکه حالات بی‌اهمیت وجود داشته باشد تعداد خروجی‌ها کمتر خواهد بود. اگر

m تعداد خروجی‌ها باشد، داریم:

$m \le ۲^n$

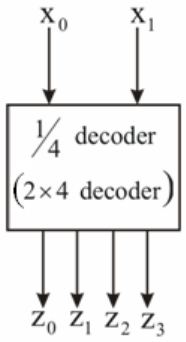
بنابراین رمز گشای ۱ از ۲^n $\left(۱/۲^n \text{ decoder}\right)$مدار ترکیبی است با n خط ورودی و ۲^n خط خروجی که در آن هریک از ۲^n ترکیبات ورودی تنها یکی از خطوط خروجی را فعال می‌کند. به این مدار $\text{Dec } n\times 2^n$ نیز می‌گویند.



مثال: یک $\frac{۱}{۴}$decoder طرح کنید.

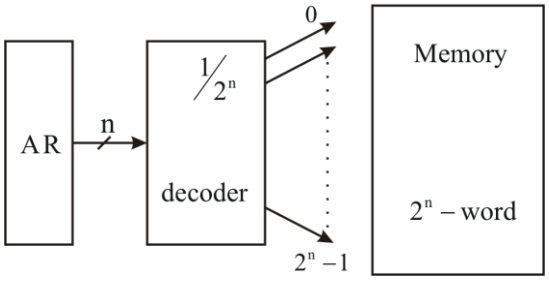
حل: جدول صحت و بلوک دیاگرام این مدار به صورت زیر می‌باشد:

$x_۰$	$x_۱$	$z_۰$	$z_۱$	$z_۲$	$z_۳$
۰	۰	۱	۰	۰	۰
۰	۱	۰	۱	۰	۰
۱	۰	۰	۰	۱	۰
۱	۱	۰	۰	۰	۱



به رمز گشا مبدل رمز نیز می‌گویند مانند مبدل رمز BCD به Seven Segment decoder

کاربرد اصلی رمز گشا در آدرس دهی کلمات حافظه است که در آن n ورودی به عنوان آدرس تلقی می‌شود که برای گزینش یکی از ۲^n کلمه حافظه مورد استفاده قرار می‌گیرد.



در این شکل AR ، ثبات آدرس می‌باشد که آدرس را به رمزگشا می‌دهد.

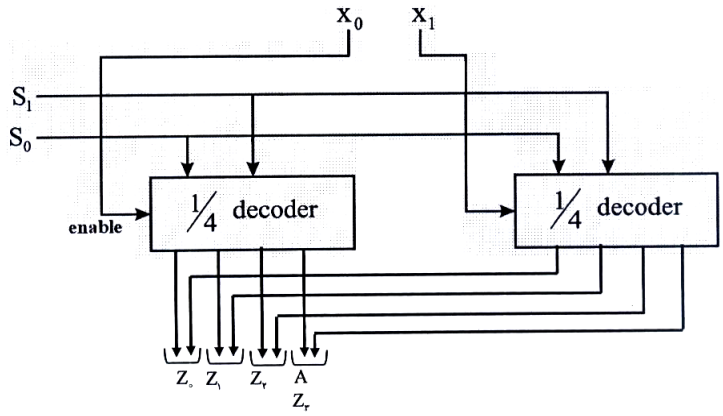
اگر خط enable رمزگشاها را به عنوان منبع ورودی در نظر بگیریم، می‌توان با اتصال درختی رمزگشاها $\frac{۱}{۲^n}$، بیش از n خط ورودی را رمزگشایی کنیم.

دی‌مالتی پلکسر (DeMux)

برای ارسال داده از یک منبع مشترک به یکی از چندین مقصد می‌توان از دی‌مالتی پلکسر استفاده نمود. بنابراین این مدار دارای یک خط ورودی و ۲^n خط خروجی می‌باشد.

با اتصال شبکه‌ای دی‌مالتی پلکسرهای یک بیتی با k خروجی می‌توان دی‌مالتی پلکسر m بیتی با k خروجی به وجود آورد.

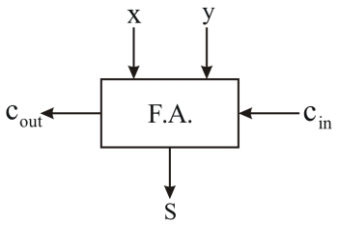
مثلاً دی‌مالتی پلکسر ۲ بیتی با ۴ خروجی را می‌توان به وسیله دو $\frac{۱}{۴}$ decoder به صورت زیر طراحی نمود:



هر تابع ترکیبی با n ورودی و m خروجی را می‌توان به وسیلهٔ یک $\frac{۱}{۲^n}$decoder و m تا گیت or پیاده‌سازی کرد.

مثال: با استفاده از decoder، گیت or یک تمام جمع‌کننده طراحی کنید.

حل: بلوک دیاگرام و جدول صحت یک تمام جمع‌کننده به صورت زیر می‌باشد:

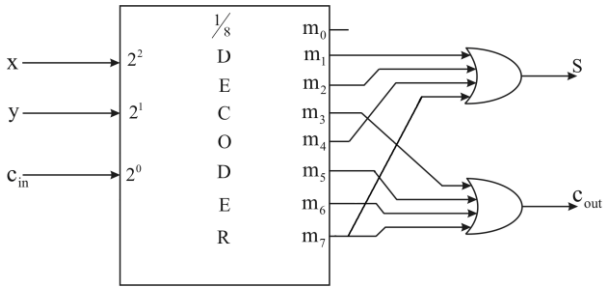


	x	y	c_{in}	S	c_{out}
$m_۰$	۰	۰	۰	۰	۰
$m_۱$	۰	۰	۱	۱	۰
$m_۲$	۰	۱	۰	۱	۰
$m_۳$	۰	۱	۱	۰	۱
$m_۴$	۱	۰	۰	۱	۰
$m_۵$	۱	۰	۱	۰	۱
$m_۶$	۱	۱	۰	۰	۱
$m_۷$	۱	۱	۱	۱	۱

$$S(x,y,c_{in}) = \Sigma m(۱,۲,۴,۷)$$
$$C_{out}(x,y,c_{in}) = \Sigma m(۳,۵,۶,۷)$$

بنابراین می‌توان مدار را به صورت زیر طراحی نمود:

$$n = ۳ \Rightarrow \frac{۱}{۸}\text{decoder}$$
$$m = ۲ \Rightarrow ۲ - \text{OR gate}$$



در صورتیکه تعداد مینترم‌ها زیاد باشد می‌توان بجای f از f' و بجای OR از NOR استفاده نمود.

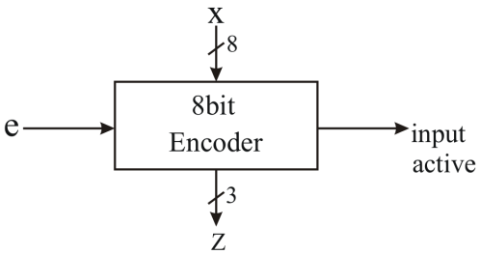
رمز گذار (Encoder)

رمزگذار یک مدار ترکیبی با 2^k ورودی و k خروجی است که آدرس و یا نام خط ورودی فعال (active input line) را تولید می‌کند بنابراین رفتار آن عکس رفتار decoder می‌باشد.

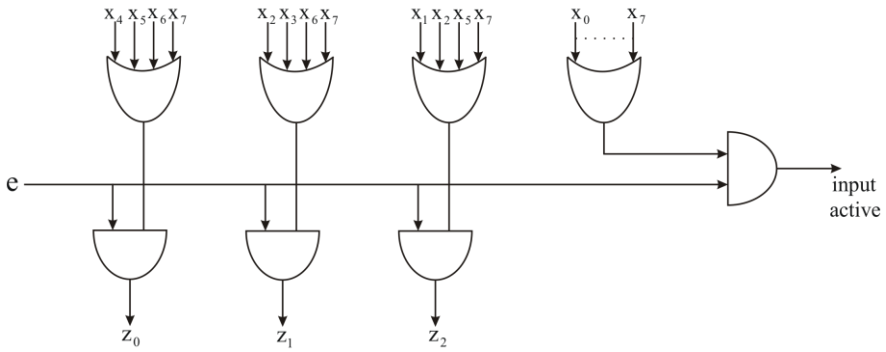
مثال: یک رمزگذار با ۸ ورودی و ۳ خروجی طرح کنید.

حل: بلوک دیاگرام و جدول صحت یک رمزگذار به صورت زیر می‌باشد:

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	z_0	z_1	z_2
۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۱
۰	۰	۱	۰	۰	۰	۰	۰	۰	۱	۰
۰	۰	۰	۱	۰	۰	۰	۰	۰	۱	۱
۰	۰	۰	۰	۱	۰	۰	۰	۱	۰	۰
۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۱
۰	۰	۰	۰	۰	۰	۱	۰	۱	۱	۰
۰	۰	۰	۰	۰	۰	۰	۱	۱	۱	۱



بنابراین مدار این رمزگذار به صورت زیر می‌باشد:



مشکل اول این مدار این است که مدار نمی‌تواند بین حالتی که تنها x_0 فعال است و حالتی که هیچکدام از ورودیها فعال نیستند تمایز قایل شود برای از بین بردن این عیب می‌تواند از یک خط ورودی اضافی به نام input active استفاده کرد.

مشکل دوم این مدار این است که در این مدار اگر بیش از یک خط ورودی به طور همزمان فعال شوند آدرس نادرست تولید خواهد شد مثلاً اگر $x_1 = x_2 = ۱$ باشد $z = ۰۱۱$ خواهد بود که متناظر با نشانی x_3 است و نادرست است. به همین علت از رمزگذار اولویت‌دار (Priority Encoder) استفاده می‌شود.

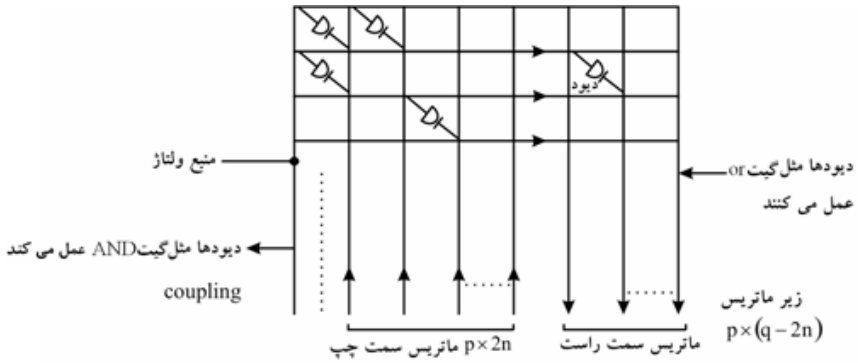
در این نوع از رمزگذار برای از بین بردن این ابهام برای ورودیها حق تقدم قایل شده اند، به طوریکه نشانی حاصل در خروجی encoder متناظر با خط ورودی است که حق تقدم آن بیشتر است.

اگر $i > j$، آنگاه حق تقدم x_i بالاتر از x_j خواهد بود.

در برخی کاربردها مخصوصاً در طراحی واحد کنترل مجموعه توابعی به صورت تصادفی می‌باشد و دسته بندی آنها آسان نیست. در چنین مواقعی از آرایه‌های منطقی (logic Array Elements) برای پیاده‌سازی توابعی ترکیبی تصادفی استفاده می‌شود که این مؤلفه‌ها در شرایط زیر صدق می‌کند.

۱- ساختمان اساسی آنها یکنواخت هستند بنابراین با استفاده از تکنولوژی ساخت مدارهای مجتمع می‌توان تعداد زیادی از آنها را تولید نمود. (Mass production)

۲- ساختمان اصلی آن به هنگام ساخت چنان در نظر گرفته می‌شود که در آینده می‌توان تغییراتی در آن به وجود آورد و توابع جدید موردنظر را پیاده‌سازی کرد. یکی از این مؤلفه‌ها معروف به آرایه منطقی همه منظوره یا PLA (Programmable Logic Array) می‌باشد که می‌تواند مجموعهٔ توابع ترکیبی را به صورت مدار دو سطحی متناظر با مجموع حاصلضرب‌ها تولید کند. ساختمان داخلی آن از p سطر و q ستون از هادی‌ها به صورت زیر تشکیل شده است.



این ماتریس شبکه‌ای است از هادیهاست که شامل p سطر و q ستون می‌باشد و عناصر جفت ساز مانند دیود و ترانزیستور، خطوط افقی و قائم را به یکدیگر متصل می‌کند.

هر خط افقی یک جمله حاصلضرب تولید می‌کند که به عنوان ورودی زیر ماتریس $p\times(q-2n)$ مورد استفاده قرار می‌گیرد و هر خط قائم در زیر ماتریس سمت راست یک تابع به صورت مجموع حاصلضرب‌ها تولید می‌کند و هر یک از خطوط قائم در زیر ماتریس سمت چپ پایه متغیر ورودی و یا به متمم آن متصل می‌شود. به طور کلی یک PLA با ابعاد $p\times q$ می‌تواند مجموعه‌ای از $q-2n$ تابع n متغیری را به شرطی که تعداد کل جملات حاصلضرب توابع از p تجاوز نکند، پیاده‌سازی کند.

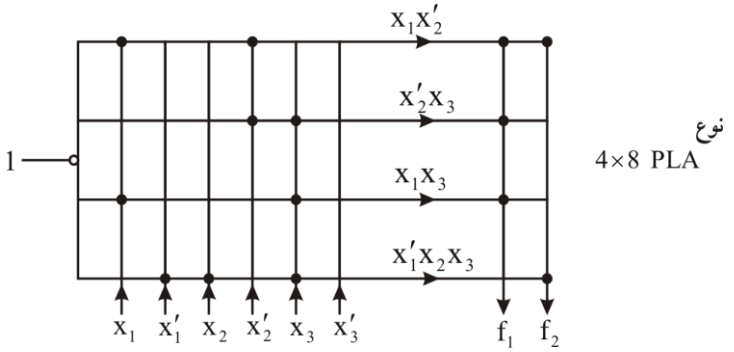
مثال: توابع

$$\begin{cases} f_1 = x_1x_2' + x_2'x_3 + x_1x_3 \\ f_2 = x_1x_2' + x_1'x_2x_3 \end{cases}$$

را پیاده‌سازی کنید.

حل: تعداد جملات حاصلضرب برابر ۴ $p=4$ می‌باشد بنابراین و داریم:

تعداد توابع $= q-2n=2\ ,\ n=3 \Rightarrow q=8$

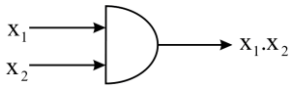


Arithmetic Element

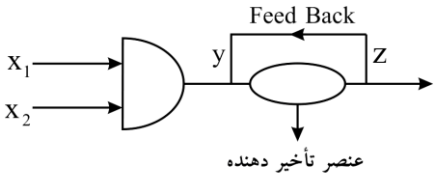
مدارهای ترتیبی

در مدارهای ترتیبی برخلاف مدارهای ترکیبی خروجی نه تنها به ورودی فعلی بلکه به ورودی‌های قبل نیز وابسته است و این نوع مدارها قادر به ذخیره اطلاعات می‌باشند.

مدار زیر یک مدار ترکیبی می‌باشد:



با اعمال تغییر زیر این مدار به مدار ترتیبی تبدیل می‌شود.



$$z(t+\Delta) = y(t)$$

$$z(t+\Delta) = x_1(t) \wedge x_2(t)$$

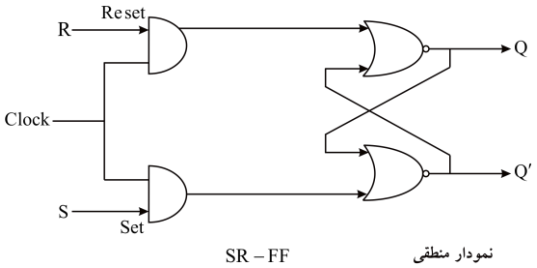
عنصرتأخیر دهنده، یک انباره موقتی است (یعنی بعد از زمان Δ، اطلاعات از عنصر تأخیردهنده عبور می‌کند)

ولی اطلاعات ذخیره شده y بعد از Δ واحد زمان از بین می‌رود یعنی اگر ورودیها صفر شوند اطلاعات از بین می‌رود. برای بدست آوردن یک مدار منطقی که قادر باشد که اطلاعات را به طور نامحدود از زمان در خود ذخیره کند. به یک چرخه (cycle) یا یک مسیر بازخورد (feed back) نیاز داریم.

فلیپ فلاپ‌ها

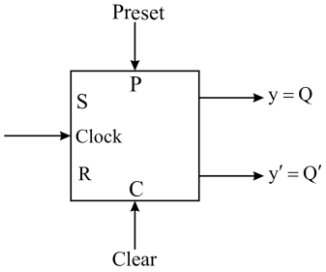
انواع فلیپ فلاپ‌های مختلف را می‌توان با اضافه نمودن clock به Latch به وجود آورد. خطوط کنترل به مدار Latch یکی از دو حالت ۰ یا ۱ می‌باشد. دو خروجی فلیپ فلاپ مکمل یکدیگرند.

فلیپ‌ها فلاپ‌ها را می‌توان توسط Preset و Clear مقداردهی اولیه کرد.



اگر Clear = ۱ آنگاه فلیپ‌فلاپ صفر می‌شود.

اگر Preset = ۱ آنگاه فلیپ‌فلاپ یک می‌شود.

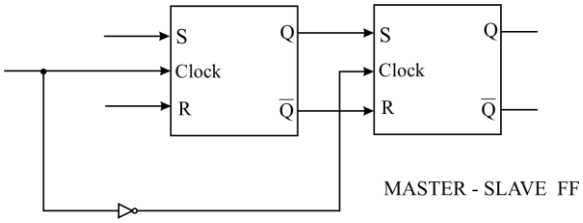


مسأله مسابقه(Race Problem)

مدت زمانی لازم است تا Clock بتواند در فلیپ فلاپ تأثیر بگذارد. یعنی به اندازهٔ کافی زمان لازم است تا فلیپ فلاپ‌ها بتوانند با توجه به ورودی، خروجی مناسب را تولید کنند و همچنین هنگامیکه clock فعال است، باید ورودی‌ها ثابت بمانند. حال اگر در

خروجی تغییراتی ایجاد شود، به دلیل وجود مسیر بازخورد تغییراتی مجدد در خروجی‌ها ایجاد خواهد شد. از طرف دیگر خروجی فلیپ فلاپ می‌تواند ورودی فلیپ فلاپ دیگر باشد.

معمولاً برای ساخت فلیپ فلاپ‌ها از مدارهای مجتمع Master-Slave استفاده می‌شود که این کار به صورت زیر می‌باشد:

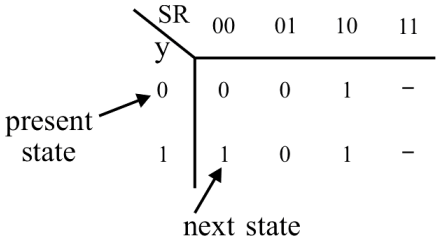


در اینجا کل مدار تنها یک بیت را ذخیره می‌کند و از این مدار برای جلوگیری از تأخیرات ناهماهنگ فلیپ فلاپ‌ها و از بین بردن آنها، که همان Race problem می‌باشد، استفاده می‌شود.

انواع فلیپ فلاپ‌ها

فلیپ فلاپ SR

که نمودار منطقی و بلوکی آن قبلاً ذکر شد. جدول انتقال این فلیپ فلاپ به صورت زیر می‌باشد.

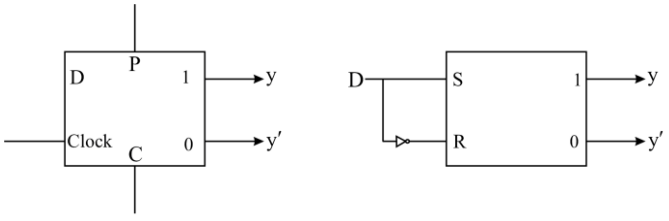


جدول تحریک این فلیپ فلاپ به صورت زیر می‌باشد:

ورودی‌ها			
$R(t)$	$S(t)$	$y(t+1)$	$y(t)$
–	۰	۰	۰
۰	۱	۱	۰
۱	۰	۰	۱
۰	–	۱	۱

فلیپ فلاپ D

بلوک دیاگرام و مدار این فلیپ فلاپ به صورت زیر می‌باشد.



جدول تحریک و جدول انتقال نیز به صورت زیر می‌باشد:

y \ D	0	1
0	0	1
1	0	1

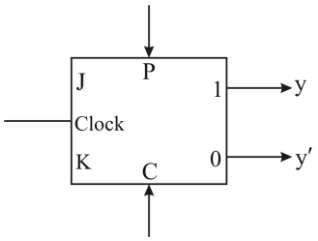
$y(t+1)= D(t)$

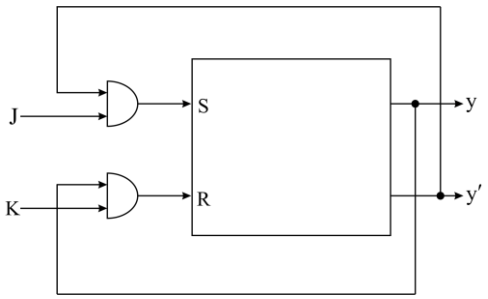
$y(t)$	$y(t+1)$	$D(t)$
0	0	0
0	1	1
1	0	0
1	1	1

این فلیپ فلاپ تنها با تأخیر یک واحد زمان همان ورودی را در خروجی تولید می‌کند.

فلیپ فلاپ JK

بلوک دیاگرام و مدار این فلیپ فلاپ به صورت زیر می‌باشد:





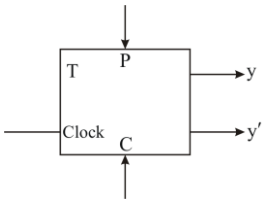
جدول تحریک و حالت این فلیپ فلاپ نیز به صورت زیر می‌باشد:

y \ JK	00	01	10	11
0	0	0	1	1
1	1	0	1	0

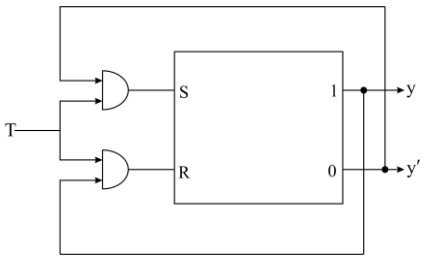
$y(t)$	$y(t+۱)$	$J(t)$	$k(t)$
۰	۰	۰	–
۰	۱	۱	–
۱	۰	–	۱
۱	۱	–	۰

فلیپ فلاپ T

دیاگرام بلوکی این فلیپ فلاپ به صورت زیر می‌باشد:



نکته: T-FF همان JK-FF است که ورودی‌های J و K را به هم وصل کرده اند. ساختار این فلیپ فلاپ و جدول حالت آن به صورت زیر می‌باشد:



y	T=0	T=1
0	0	1
1	1	0

همچنین جدول تحریک این فلیپ فلاپ به صورت زیر می‌باشد:

$y(t)$	$y(t+1)$	$T(t)$
۰	۰	۰
۰	۱	۱
۱	۰	۱
۱	۱	۰

ثبات‌ها (Registers)

ثبات m بیتی مجموعه مرتبی از m فلیپ فلاپ است که برای ذخیره نمودن یک کلمهٔ m بیتی بکار می‌رود که در آن هر بیت در یک فلیپ فلاپ ذخیره می‌شود.

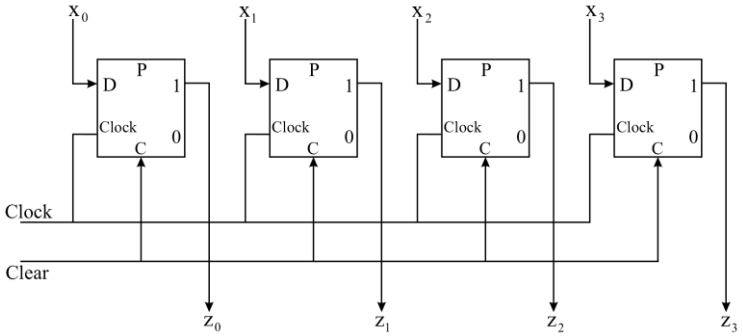
بیت‌ها

$$Z=(Z_0, Z_1, \ldots, Z_{m-1})$$

چون اطلاعات ذخیره شده در یک ثبات به صورت یک کلمه واحد در نظر گرفته می‌شود بنابراین سیگنال‌های کنترل مانند clock و clear در همهٔ فلیپ فلاپ‌ها باید یکسان و مشترک باشد و از آنجاییکه اطلاعات را می‌توان به طور همزمان به درون و بیرون فلیپ فلاپ‌های ثبات انتقال داد بنابراین به آنها ***parallel input / output*** گویند. ثباتها را می‌توان با هر نوع فلیپ فلاپی ساخت ولی معمولاً از فلیپ فلاپ‌های Master Slave استفاده می‌شود، این مساله به خاطر Race Problem می‌باشد.

مثال: با استفاده از فلیپ فلاپ D یک ثبات چهار بیتی طراحی کنید.

حل: چون $m = ۴$ است بنابراین نیاز به ۴ فلیپ فلاپ داریم، می‌توان این کار را به صورت زیر انجام داد:



$x=(x_0, x_1, x_2, x_3)$: کل ورودی‌ها را می‌توان با x در نظر گرفت.