



دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

درس سیستم‌های نهفته

دکتر حسینی منزه

تمرین اول

مقدمه

مصطفی قدیمی

سؤال ۱.

تلفن همراه در گذشته به عنوان سیستم نهفته در نظر گرفته می‌شده است. به دلیل این‌که یک کار ساده مانند برقراری تماس و یا ارسال پیامک جزو تنها کاربردهای آن‌ها بوده است و با برقراری یک جریان الکتریکی انجام یک کار هوشمند انجام می‌داده‌اند و به همین دلیل سیستم نهفته محسوب می‌شدند (طبق تعریف). اما امروزه به دلیل این‌که به عنوان یک دستگاه با توان پردازشی بالا به حساب می‌آیند، آن‌ها را به عنوان کامپیوتر و نه سیستم نهفته می‌شمارند.

سؤال ۲.

یک ساعت دیجیتال معمولی را در نظر می‌گیریم.

• الف)

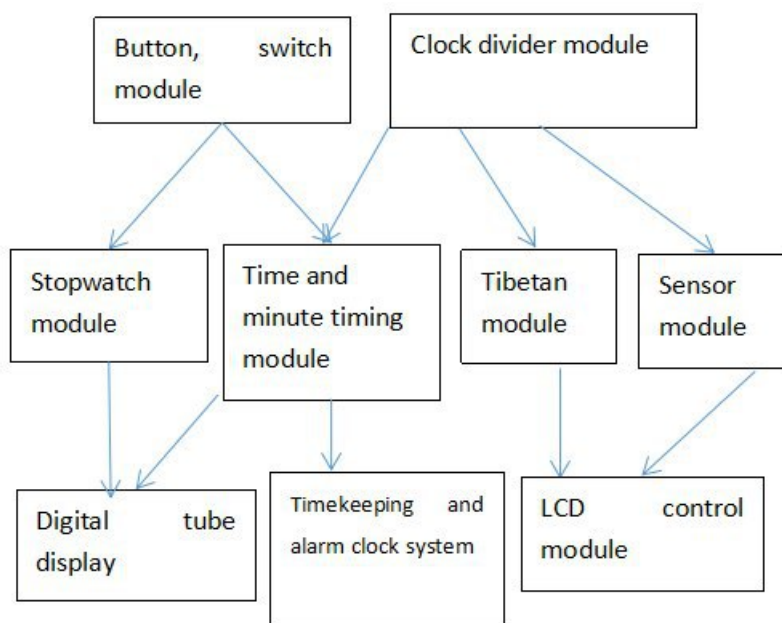
– اتکاپذیری:

- * دسترس‌پذیری: باید هر زمان که از ساعت می‌خواهیم استفاده کنیم، کار کند.
- * قابلیت اطمینان: باید زمان را به درستی نشان دهد.
- * امنیت: این ویژگی در ساعت اهمیت چندانی ندارد.
- * قابلیت نگهداری: باید بتوان از ساعت دیجیتال به درستی نگهداری کرد و قطعات آن در بازار موجود باشد.
- مصرف انرژی: مصرف انرژی آن باید تا حد امکان کم باشد.
- کارایی: اگر کارایی را مطابق با فرمول زیر در نظر بگیریم، باید کارایی آن ۱ باشد.

$$performance = \frac{1}{execution\ time}$$

- وزن: نباید وزن زیادی داشته باشد.
- قیمت: هزینه‌ی ساخت آن نباید زیاد شود.
- سایز کد: نباید کد اجرای آن بزرگ شود، تا از قطعات کم‌تری استفاده شود.
- همان‌طور که در سایز کد گفته شد، باید از کم‌ترین منابع با بیش‌ترین قدرت استفاده کرد.
- بی‌درنگ: چون کاربرد آن استفاده تشخیص زمان است، بنابراین، محدودیت بی‌درنگ بودن در آن از اهمیت بالایی برخوردار است.

• ب)



شکل ۱: نمودار بلوکی ساعت دیجیتال

- (ج) اگر در این قسمت قیمت تمام شده را در نظر بگیریم، چه برای شرکت تولیدکننده چه برای تولید شخصی محصول، اختلاف قیمت وجود خواهد شد. دلیل وجود این اختلاف قیمت، حذف شدن هزینه‌ی طراحی اولیه و مهندسی (که در سوال ۳ راجع به آن توضیح داده شده است) می‌باشد. به همین دلیل قیمت برای شرکت تولیدکننده کمتر از تولید آن با جنس، کیفیت و ... مشابه همان محصول به طور شخصی است.

سؤال ۳.

• الف) به هزینه‌ی اولیه‌ای که صرف تحقیق، طراحی، ایجاد و تست یک محصول جدید یا بهبود محصول می‌شود، اصطلاحاً هزینه‌ی NRE یا Non-Recurring Engineering گفته می‌شود. با وجود این‌که این هزینه فقط یک بار برای محصول صرف می‌شود، اما هنگام بررسی سوددهی حتماً باید در نظر گرفته شود.

• ب)

– اگر بخواهیم از ریزپردازه استفاده کنیم:

$$cost = 5000 + 25x$$

– اگر بخواهیم از ASIC استفاده کنیم:

$$cost = 100000 + 5x$$

با مساوی قرار دادن این دو معادله به دست می‌آوریم که $x = 4750$ و به این معنی است که اگر بخواهیم تا ۴۷۵۰ محصول تولید کنیم، استفاده از ریزپردازه به صرفه‌تر است، در غیر این صورت (بیش از ۴۷۵۰ محصول)، استفاده از ASIC.

سؤال ۴.

یک پردازنده‌ی سیگنال دیجیتال، همان‌طور که از نامش مشخص است سیگنال‌های دیجیتال مانند صدا، ویدیو، دما، و... را دریافت می‌کند و بر روی آن‌ها عملیات ریاضی انجام می‌دهد. این دستگاه‌ها برای انجام عملیات سریع ریاضی (مانند جمع، تفریق، ضرب و تقسیم) بهینه‌سازی و ساخته شده‌اند. سیگنال‌ها در آن پردازش می‌شوند بنابراین اطلاعاتی که دارند می‌توانند نمایش داده شوند، تحلیل و بررسی روی آن‌ها صورت گیرد و یا به یک سیگنال با نوع دیگر تبدیل شوند؛ برای مثال صدا یک سیگنال آنالوگ است و نیاز است تا ابتدا به کمک یک تبدیل‌کننده، به سیگنال دیجیتال تبدیل شده و به عنوان ورودی به Digital Signal Processor داده شود. از مهم‌ترین الگوریتم‌هایی که در پردازش سیگنال‌های دیجیتال استفاده می‌شود، می‌توان به موارد زیر اشاره کرد:

• Fast Fourier Transform

• Finite Impulse Response Filters

برای کاربردهای پردازش سیگنال‌های دیجیتال نیز می‌توان به موارد زیر اشاره کرد:

• پردازش گفتار

• استفاده در رادار

• ارتباطات راه دور

• پردازش صدا

• پردازش تصویر

• ردیاب آوایی

• ...

سؤال ۵.

میکروکنترلر AVR دارای معماری RISC^۱ می باشد. در هر کلاک می توانند یک دستور ساده انجام دهند؛ مانند انتقال عدد از حافظه به رجیستر. این میکروکنترلرها معمولا ۸ بیتی هستند. از آن ها برای انجام کارهای کنترلی استفاده می شود. به بیان دیگر در جاهایی که نیاز به محاسبات پردازشی وجود ندارد، از آن ها استفاده می شود؛ مانند کنترل روشنایی میکروکنترلر ARM دارای معماری RISC می باشد. این میکروکنترلرها معمولا ۳۲ بیتی هستند. به دلیل توانایی بالا در اجرای سیستم عامل های مختلف مانند لینوکس، دارای کاربرد وسیعی در حوزه های مختلف می باشند؛ مانند:

- تبلت

- دوربین های دیجیتال

- تلفن های همراه

- ...

میکروکنترلر ۸۰۵۱، ۸ بیتی است. دارای فضای stack بسیار محدود ۱۲۸ بیتی است که حتی نوشتن یک کامپایلر برای زبان C را در آن ها با چالش روبه رو می کند. به ازای هر instruction به چندین clock نیاز دارد و توسط شرکت Intel طراحی شده است.

- Bus width

- 8051 : ۸ بیتی

- AVR : ۳۲ / ۸ بیتی

- ARM : اغلب ۳۲ بیتی است اما ۶۴ بیتی هم دارد.

- سرعت

- 8051 : $12 \text{ clock/instruction cycle}$

- AVR : $1 \text{ clock/instruction cycle}$

- ARM : $1 \text{ clock/instruction cycle}$

- ISA

- 8051 : CLSC

- AVR : RISC

- ARM : RISC

- مصرف انرژی

- 8051 : متوسط

- AVR : کم

- ARM : کم

- Memory Architecture

- 8051 : Von Neumann

- AVR : Modified

- ARM : Modified Harvard Architecture

^۱ Reduced Instruction Set Architecture

سؤال ۶.

• الف)

– Hard Real-time: به سیستم‌هایی گفته می‌شود که ضمانت شود به همه‌ی ددلاین‌ها در محدوده زمانی مشخص شده، پاسخ می‌دهد؛ مانند سیستم‌های هسته‌ای، سیستم‌های برقی هواپیما، دستگاه‌های ضربان‌ساز در پزشکی و ...

– Firm Real-time: به سیستم‌هایی گفته می‌شود که تعداد از دست دادن ددلاین در آن‌ها بسیار کم است و به عنوان سیستمی که fail شده آن‌ها را در نظر نمی‌گیرند. در این مواقع به دلیل فاصله مناسب خطاها، سیستم می‌تواند نجات پیدا کند، در حالی که ارزش انجام کار به صفر می‌رسد یا غیرممکن می‌شود؛ مانند تولید سیستم‌های دارای خط مونتاژ روباتی که از دست دادن ددلاین باعث درست وصل نشدن قطعه می‌شود. اما تا زمانی که تعداد قطعات خراب به اندازه کافی کم و نادر باشند که توسط کنترل کیفیت رد بشوند و هزینه زیادی هم نداشته باشند، تولید ادامه دارد. یک مثال دیگر می‌تواند کابل‌های دیجیتال set-up box باشد که زمان را موقعی که نیاز است تا روی صفحه فریم‌ها نمایش داده شوند، رمزگشایی می‌کند. چون فریم‌ها به ترتیب زمانی حساسند، از دست دادن یک ددلاین باعث لرزش و کاهش کیفیت تصویر می‌شود. اگر فریم از دست رفته بعداً در دسترس قرار بگیرد، نمایش آن باعث لرزش بیش‌تر تصویر می‌شود بنابراین بی‌فایده است. بیننده تا زمانی که تعداد این لرزش‌ها زیاد و مکرر نباشد، از برنامه می‌تواند لذت ببرد.

– Soft Real-time: به سیستم‌هایی گفته می‌شود که مکرراً در آن‌ها ممکن است ددلاین‌ها از دست می‌روند اما چون تسک‌ها به‌طور زمان‌بندی شده اجرا می‌وند نتایج آن‌ها مقادیری دارد. مثلاً در ایستگاه‌های هواشناسی سنسورهای زیادی برای خواندن و اندازه‌گیری دما، رطوبت، سرعت باد و ... وجود دارد. خواندن اطلاعات آن‌ها باید در بازه‌های زمانی مشخص انجام گیرد و انتقال داده شوند. در صورتی که این فرآیند، هم‌آهنگ نیست. با این وجود، هم‌چنان این عددها می‌توانند به اندازه‌ی کافی نزدیک باشند. نمونه دیگر آندر کنسول‌های بازی است که یک نرم‌افزار را برای موتور بازی اجرا می‌کند. منابع زیادی وجود دارند که باید بین تسک‌ها به اشتراک گذاشته شوند. هم‌چنین، این تسک‌ها باید کامل شوند با توجه به برنامه‌ی بازی تا به‌درستی نمایش داده شوند. در صورتی که بتوانند نزدیک به هم اجرا شوند که لذت‌بخش خواهد بود اما در غیر این صورت در بازی وقفه‌ی کوچکی به وجود می‌آید.

• ب) سیستم‌عامل‌های همه‌منظوره یک جزء ضروری در هر دستگاه موبایل، سیستم کامپیوتری و ... است. برای انجام چندین کار به‌طور هم‌زمان بسیار عالی عمل می‌کنند ولی مشکلاتی نظیر تاخیر و هم‌آهنگی آن‌ها را برای کاربردهایی که به زمان حساس هستند، غیر ایده‌آل می‌کند.
در مقابل سیستم‌عامل‌های بی‌درنگ پلتفرم‌های نرم‌افزاری هستند که برای کاربردهایی طراحی شده‌اند که زمان در آن‌ها اهمیت زیادی دارد.

– RTOS

- * برنامه‌ریزی همیشه بر اساس اولویت است.
- * یک تسک با اهمیت کم‌تر توسط یک تسک با اهمیت بیش‌تر متوقف می‌شود؛ حتی اگر در حال اجرا کردن kernel باشد.
- * در جایی که توسعه مهم است، کد kernel سیستم‌عامل‌های بی‌درنگ طوری طراحی شده‌اند تا مقیاس‌پذیر باشند، بنابراین توسعه‌دهنده می‌تواند این کار را به راحتی انجام دهد.

- * اجرای برنامه‌ها قطعی^۲ است و هیچ الگوی اجرای تصادفی در آن وجود ندارد.
- * زمان اجرای و دریافت پاسخ قابل پیش‌بینی است.
- * محدودیت زمانی دارد.
- * چون از آن‌ها در میکروکنترلرها استفاده می‌شود، نباید بیش‌تر از ده درصد از فضای حافظه را اشغال کند.
- * به‌طور بهینه پیاده‌سازی نمی‌شود.

- GPOS

- * برنامه‌ریزی همیشه بر اساس اولویت برنامه‌ریزی تسک‌ها همیشه براساس اولویت و اهمیت نیست.
 - * هر چه تعداد ترد بیش‌تر باشد، برنامه‌ریزی و شروع اجرای آن‌ها زمان بیش‌تری می‌گیرد.
 - * در جایی که توسعه مهم است، در حالت کلی کد سیستم‌عامل‌های همه‌منظوره ماهیت modular ندارد.
 - * الگوی اجرایی تصادفی دارد.
 - * تضمینی برای مدت زمان دریافت پاسخ وجود ندارد.
 - * نگاشت حافظه به صورت پویا انجام می‌شود.^۳
 - * به‌طور میانگین برای اکثر کارها بهینه پیاده‌سازی شده است.^۴
 - * حافظه زیادی دارد.
- با توجه به مقایسه‌ای که انجام شده، بسته به کاربرد هر کدام از این‌ها می‌توانند مزیت و یا عیب در نظر گرفته شوند. برای نمونه سیستم‌عامل‌های RTOS چون کم‌ترین تعداد تسک‌ها رو انجام می‌دهند برای کاربردهایی که انجام چندین تسک به‌طور هم‌زمان، پردازش سبک اما با سرعت و کارایی مد نظر است، به کار نمی‌آیند.

• ج)

- سیستم کنترل آلودگی هوا
- سیستم رزرو هواپیمایی
- صنایع دفاعی مانند رادار
- سیستم‌هایی که فوراً به‌روزرسانی می‌شوند.
- و ...

^۲ deterministic

^۳ Dynamic Memory Mapping

^۴ average case