

## Lec1-5

• Safety: زمانی که سیستم فیل (Fail) می شود توان رد کردن را از دست می دهد side effect

زیادی نداشته باشد

• در سیستم های نرم افزاری reliability اهمیت بیشتری دارد (مثلاً فرض کنید با جابجایی اسلاید پروگرام)

• در سیستم های نرم افزاری Availability و security اهمیت زیادی دارد

نشان

$$P \propto f$$

نرخ

$$P \propto T_{00}$$

به منبع تغذیه

$$E \propto T_{00}$$

انرژی

• performance سیستم های نرم افزار: بین معادلات در برابری حالت اول سیستم نرم افزار از محاسبات زمانی مورد انتظار (deadline) بالاتر است

## Lec1-6

• هر چه قدر حافظه بیشتر داشته باشیم ES، اقل ریسک fault بیشتر می دهیم!

نکات رفت افولک بیشتر بشود می آید

• user interface به رابط کاربری / فرانت اند

robustness:

(fault avoidance) جلوگیری از رخداد خطا

قابلیت تطبیق

(fault tolerance) مقاومت در برابر خطا

## Lec1-7



• قابلیت اطمینان (Reliability):

$$R(t) = e^{-\lambda t}$$

در هر

زمان

نرخ افولک

Subject.

Date.

$$T_1 = 1^{\circ} \text{ms} / \lambda = 10^{-6} \frac{\text{p}}{\text{us}} \Rightarrow R(t) = 999997 \quad \text{مثال}$$

$$R(t) = 1 - \underbrace{(1 - R(T_1))(1 - R(T'_1))}_{\text{failure}}$$

و نوی کوانم در لغت افزای هم با کد، کد، کد، کد

قابلیت دسترسی پذیری (Availability) { نوع failure بازه repair } Lec 1-8

مثال: هر دو ساعت fail، هر یک ساعت repair، هر یک ساعت طول می کشد دسترسی پذیری

توی هر یک ساعت یک ساعت در دسترسه

$$= \frac{5}{6}$$

توی کل بازه بررسی نمی ریم ولی قابلیت اطمینان توی بازه خاصی هست

Lec 1-9

قابلیت maintainability

Lec 1-10

قابلیت safety (ایمنی)

سیستم درست کار کنه، داده درست کار کنه، کار کنه خودش رو سریع درست کنه

$$S(t) = R(t) + (1 - R(t))(F_{\text{safe}}(t))$$

fail safe

Lec 1-11

قابلیت امنیت (ایمنی)

یک سیستم میزنه اطلاعات که در دل سیستم هست (داده فزاد) لایه

ATM، و سیستم فزاد میزنه ES و سیستم فزاد

احراز  
لایه سیستم

Subject.

Date.

موضوع: عملیات ریاضی بر روی مجموعه‌های داده‌ها

Lect-12

موضوع: عملیات ریاضی بر روی مجموعه‌های داده‌ها

Real  
time



Subject.

Date.

مقاله سیستم  
موضوع: (تورن ۱) روی بردار FSM داده‌های (اسمیل) بردار Verity (۱۹ اسفند)

موضوع (۲) پیاده‌سازی Air conditioning ۲۲ اسفند

موضوع (۳) مدل ۷ اسفند True Time (مقیاس) (Mobile Notes) (۱۷ اسفند)

۴. مسائل در خطای طراحی (I-M) modeling (Lee 3-7)  
خطای پیاده‌سازی (I-M)  
خطای فرسودگی (aging - wearout)  
خطاهای محیطی - پیرایی (E-D)

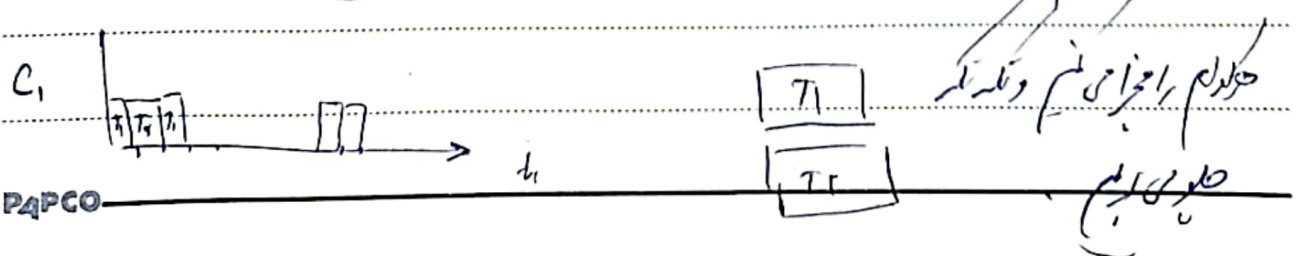
و شبیه‌سازی: یعنی با مقداردهی محدودیت‌کنی سیستم

Formal verification: با استفاده از فرمول‌ها و مطالبات به اثبات درستی کارکرد سیستم می‌پردازیم

ترتیب برنامه اهمیت ندارد

concurrency: یعنی ترتیب کار، برنامه (سینکرون) (Lee 3-8)

کامپیوترهای همزمان Core - concurrent سینکرون برنامه  
اول  $T_1$  به  $T_2$  می‌رسد  
پس به  $T_3$  می‌رسد



به بیان دیگر concurrency این است که چند فرایند همزمان با هم تولید شوند.

مثلاً به اتوماتا می‌توانیم concurrency را توصیف کنیم.

⑤ هنگام سادی ۱. ارتباطات درست بین پردازنده‌های مختلف.

سه می‌توان در زبان برنامه‌نویسی را با ارسال خط‌ها مشخص است. برنامه به صورت concurrency نوشت (مثل C).

flexibility: تغییر در سیستم من به همان اندازه باید تغییر در توصیف من.  
مثلاً: داده تغییر توهم. داده تولید توصیف تغییر من.

به عنوان مثال io device ها به توی automate توصیف کرد.

Lee3-11 concurrency / فرایند ثابت  
variable.

① وقتی ۲ state را با هم در یک box قرار دهیم، هزینه  
جایابی بین آنها را switching اکم می‌کنیم.

② یک سری ترانسپل کردن آنها با هم این است که در state اول دچار delay شود، بعد از  
این state دوم متوجه این مورد می‌شود.

مثال: زمانی که سوئیچ می‌کنیم، قابلیت اطمینان ممکن است کاهش یابد (Reliability).

\* چرا نیاز است به از FSM های Deterministic

استفاده کرد؟  
نیاز به نیاز داریم قطبیت بالایی داشته باشیم تا بتوانیم  
سیستم زنده خود را محاسبه کنیم (به بیانی دیگر یعنی پذیرا باشند)  
چون در بعضی موارد با همان مسائل ها برخورد داریم.

• خود super state ها می‌توانند reaction داشته باشند.

با این صورت که در ضمن ورودی و خروجی super state با reaction می‌داریم.  
این reaction که انجام شده سپس به سرخ state (فعالیت) می‌رود.

مثال: زمانی که می‌خواهیم یک FSM را به single core اجرا کنیم، یعنی این را به Flex کنیم.

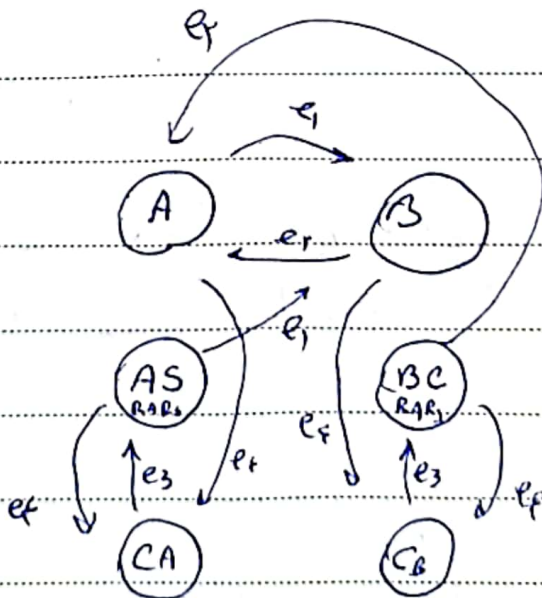
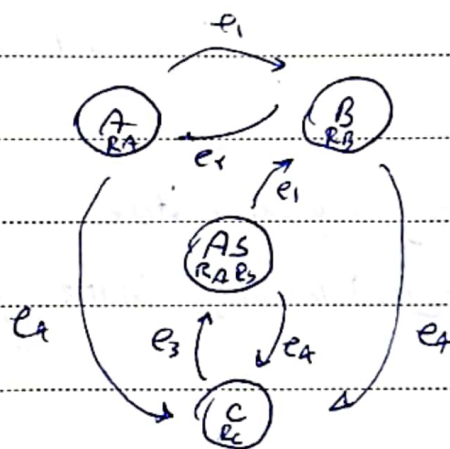
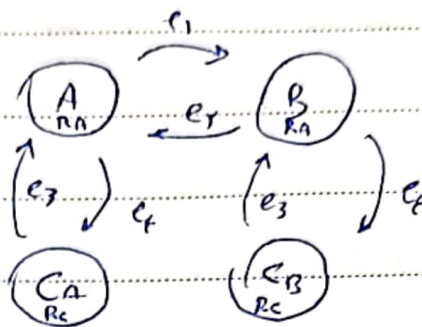
مثال: هم وقتی به معنی مولدات نیست / در هم ریزی تمام و تخریب ما هم نیست.

مصحح



Date.

Date.



**PAPCO**

Subject.

Date.

مقاله در Plat کردن اگرچه در مورد نیاز است ابتدا از لایه های پایینی شروع کرد و سپس طرح بالا بردیم.

به مدت 20ms فقط داریم، event a اتفاق می افتد اگرچه این زمان در نظر می آید، timeout انجام می شود.  
اگرچه در timeout یک event اتفاق می افتد، برای ما اهمیت دارد و هم تفاوت دارد.

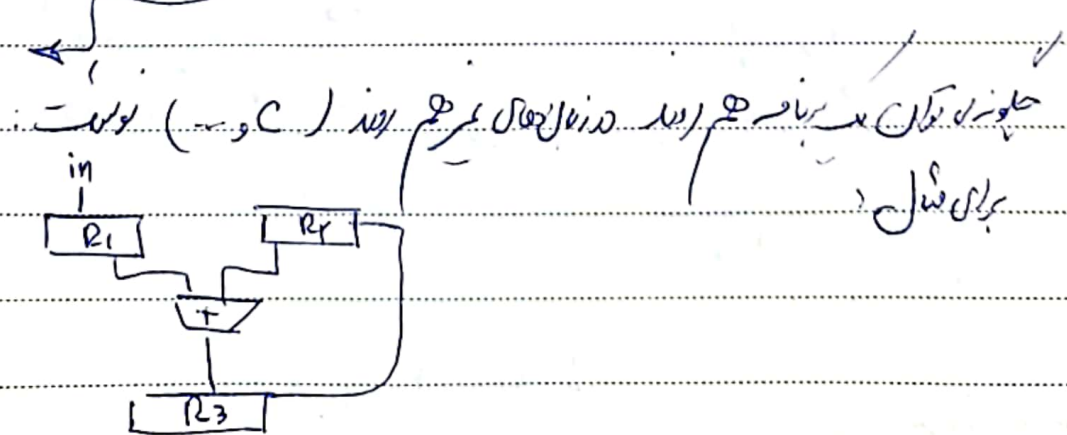
مقاله در خود timer ها و زمان reaction نسبت به یکدیگر.

dead state: حالتی که در آنجا super state -  
همیشه که یعنی حالت انتهای و در این state حرکات یا (توانایی) نوری نیستند.

Job reaction state ها -> مور

Lee 3.40: evaluation assignment -> تمام shadow ها مقداردهی می شود  
مقداردهی می شود Main var ها

Atomic EYE





Subject.

Date.

while (1) {

$S_1 = in$

$S_2 = R_3$

evaluation

$S_3 = R_1 + R_2$

$R_1 = S_1$

assignment

$R_2 = S_2$

$R_3 = S_3$

}

## Embedded system Hardware

(Lee 4-20)

تبدیل سیگنال پیوسته به نمونه  
شماره ۱ به ۲ به ۳

$D_T \rightarrow D_V$

مشکل مدار sample-and-hold چیست؟

داده ورودی  
داده خروجی

cash miss در سیستم های مبتنی بر سیسٹم حافظه من و تفاوت فرکانس  
انرژی (توان) مصرفی به می توانی با افزایش دما و فرکانس fan را کم کنی  
است نه سیستم و از بین های نرم افزاری یا تغییر در ولتاژ که باعث کم شدن توان و  
در مصرف کم شدن را خواهد شد

(Lee 4-23) Runtime error

از اتصالاتی که در زمان run-time به وجود می آید. استاندارد نمی آید و توان بهینه تر عمل کرد

P4PCO

انگا، در حال اجرای کد های بهینه سازی کرد

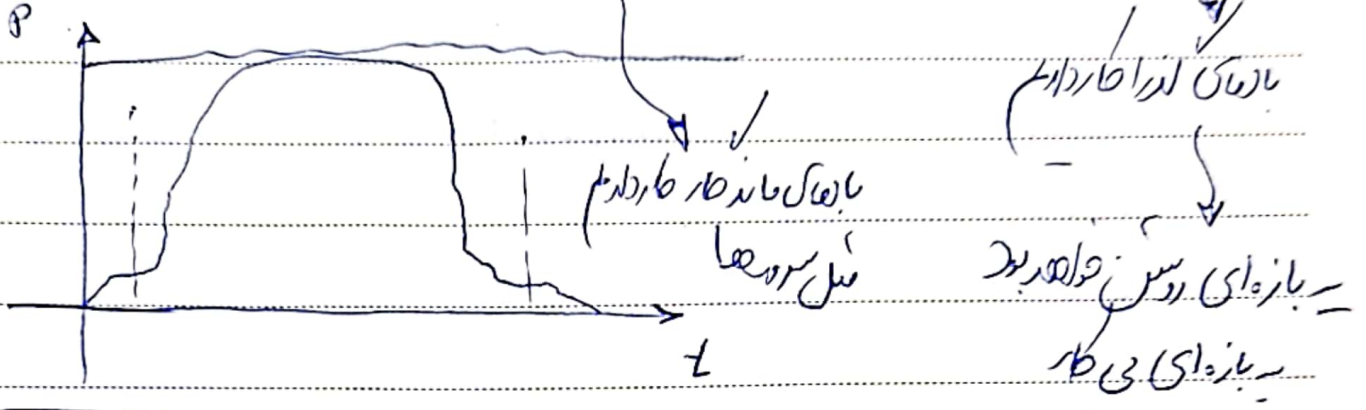
Subject.

Date.

Lec 4-26

regulator نو سارت فسخ تقویر کارایی می کنند

long term cooling VS. short term cooling



Lec 4-28

Ambient Intelligence : آشنایی با

Lec 4-30

microcontroller/microprocessor

سیستم‌های مبتنی بر microprocessor ارتباط بین پردازنده‌های آن و دایره‌های ارتباطی را برقرار می‌کند. قابلیت انجام کارهای محاسباتی و کنترل را دارد. میکروپروسسور در قلب سیستم‌های دیجیتال قرار دارد. میکروپروسسور در قلب سیستم‌های دیجیتال قرار دارد. میکروپروسسور در قلب سیستم‌های دیجیتال قرار دارد.

# Lee4-38

Subject.

Date.

مسئله در مورد داده به حافظه (1 تا 100) و (2 تا 100) در حافظه  
 { Dram ~ ضمیمه  
 { Sram ~ FF (4 تا 100)

• به وسیله ی Dram و Refresh می توان به توان دلی و ف توان بیشتر صرف داده  
 • Sram درون تر از Dram است به عنوان main استفاده می (رم)  
 • حجم زیادی هم داده به عنوان حافظه اتصال به

## Lee4-39

هر دو سال (18 تا 20) سرعت به دو برابر می (عمل 20)

هر دو سال 1.7 ده سرعت حافظ

• نکته: حال که سرعت ram به دو برابر می (عمل 20) از cash استفاده می  
 • این اتفاق در سرعت و حال کند

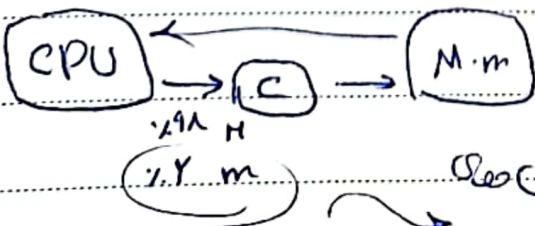
• در عمل های به دو برابر می (عمل 20) به دو برابر می (عمل 20)

## Lee4-41

ما هر می توان صرفی کاملاً خطی نیست و زیاد به چن تعداد مقایسه ها  
 بیشتر

## Lee4-42

• نکته: در هر های نقطه می توان از cash استفاده کرد زیرا  
 Predictable نیست



• در این 2 miss شدن پیش می (عمل 20)  
 ما به هم می (عمل 20)



اگر کنترل در دسترس به cash داشته باشیم به مشکل ⑤

در حالت دفعای بدون cash، چون آدرس دهی آن متفاوت است از رم

Lec 4 - 43

SPM : فضای آدرس دهی آن برابر Main mem است

تعمیم داده های آن را من به نام فیزیس، در سیستم به سه عنوان Hit  
کنترل SPM نرم افزار فیزیس است (cash به کنترل)

آدرس دهی همان آدرس است (cash دفعای بی)

Lee 5 8  
A-5

مهم ترین ویژگی OS های RT اینست که در آن زمان واقعی توجه کنند

OS resource management در آن زمان به بلان که محدودیت در

منابع انرژی دارند (battery) این هم در

safety-critical : سیستم هایی که در صورت بروز خطا دچار اتلاف بحرانی می شود

Lee 2

$$Pof \text{ (Probability of failure)} = 1 - \prod_{i=1}^n R_{ei} \quad \begin{matrix} R_{ei} & R_{er} \\ R_{er} & R_{ef} \end{matrix}$$

(sys.)

قابلیت اعتماد این بلان به واقعی است : راه حل : ترکیبی از testiney  
reasoning

مسلط ریاضی و عمل های احتمالات است

که نتایج حاصل از مدل ۸ با توجه به شرایط محیطی و سیستم

تجزیه می گردد