

جی بات فراگیری  $\rightarrow$  Ubiquitous Computing  
 (Universal Pervasive Computing)

عده کامپیوترها به صور مختلف و متنوع به صورت واحد حسابات استفاده می شوند.

❶ در سیستم های نوین جی بات دادلین های بین اندرونی سیستم و بیرونی

سرعت انتقالی زیاد  $\rightarrow$  Pipeline کردن با Cache کردن و این ها HPC

سرعت متوسط سیستم را بالا بریند. در سیستم های نوین حالت متوسط سیستم

ماله بی پذیری حالت امنیتی برقرار می شود.

❷ سیستم های نوین می خواهند Cyber-Physical متن کاری ننمایند

دانشگاهی و ارتباطی داشته باشند و باید این دانشگاهی های دیجیتال

این های را به سیستم پخته و در دست داشته باشند. در ادامه خروجی سیستم نوین کنترلر، Actuator کنترل کنند

که باعث ایجاد این درجات گذشتگی می شود.

❸ سیستم های نوین بصورت خودکار در محل و کنترل اند Green Computing

❹ پارادایم سیستم های نوین بصورت توزعی و آنلاین

هم و زندگی می باشند.

Embedded System = Software integrated with physical processes.

۱- استانداردهای ایمنی متن کار دادلین های میزبانی کرد و جدید دارد، بنابراین miss تردد ندارد.

۲- مسائلی می خوردند  $\rightarrow$  این متن کار سیستم های امنیتی این را داشته باشند و از

در اصل صرفاً یک متن کار دارند که در مقایسه سیستم های امنیتی دیگر از آنها سریع تر است.

Concurrency & parallel

Cyber-Physical (cy-phy) Systems = Embedded System + Physical Environment

۱۰) سیمہی نفته یا ~~ب~~ Robust بابت دربرابر حفاظت میں بہ عنوان مثال

۱۱) دربرابر External باستین مزفہ یا سیمہ کے درحال Disturbance

مانسٹر کر کر خود حفاظت مانگیں Shield

دین توسط گھولائی کر سکن است یعنی آن سبقتہ ایسی رہوں۔

۱۲) سیمہای نفته یا ~~ب~~ Responsive میں مانگیں جو بین حقاً کے دربرابر تسلیم

خالق کے دریافت سے کم در دلایت مشخص پاسخ فروختاں نہیں۔

۱۳) کاربر دھارہ سیمہای دیجیٹل : ۱- انقلابی سیمیں ایجاد کرنے والے افراد و تعمیر (عینہاں  
کاربر دھارہ سیمیں پڑھیں) ۲- امنیتی همینہی ۳- امنیت (Safety and Security) ۴- امنیت (Safety and Security)

۱۴) ایچ ال رس صفحہ ویڈ کے اچل نگت نہیں آن را تلقین کر کر۔ Reliability

ادعا ۱۵) این اچل بارے ~~پڑھیں~~ بالا بات۔ ایسے اکر حالے نہیں۔

هم رخ داد گایے تھے اندریں دربرابر آن را داشتے ہیں، برخطیں حالے ہیں

احدیتی کشم error

۲۴ اکتبر

- جلسه سوم -

سیم های نوچه - سیم های دوقطبی دستگاهی که برای این دو نوع سیم های دوقطبی می باشد.

پورتابل portable بعدن و توانایی حفظ بیشتر انرژی از دسترسی مسالم کام گوشی های موبایل دارد.

مقدار باری ۷۹٪ سیم های اسفلاد کنده از پردازنده های High-End سیم کنونی است.

قابلیت اعتماد (Dependability)

بعدن و بیشترین درد صراحت زیاد در هنگام خرابی fail-safe : Safety

در دسترسی ها کار و غیر علاوه بر سیم Security

قابلیت تعمیر و بازبود سیم را بتعطیف در برابر تغییرات Maintainability

در دسترسی بودن سیم در زمان) نیاز Availability

(Fault Tolerance) Reliability : آنچه از دسکارکردن سیم مطمئن باشیم.

لهم کاربردی که با جان انسان سرو کار دارد، بالاترین درجه امنیت را داشته باشد

حریت بیشتر انرژی (Energy efficiency) : حدود زمان از باتری های سیم

اسفلاد کنیم. به علاوه آنچه غریب نیست آن باتری های قدرت خواهد بود. چه اکسلن اس

تعریض باتری در یک هزار نیم بود سفارت ایالات (1) طراحی سیم کویی

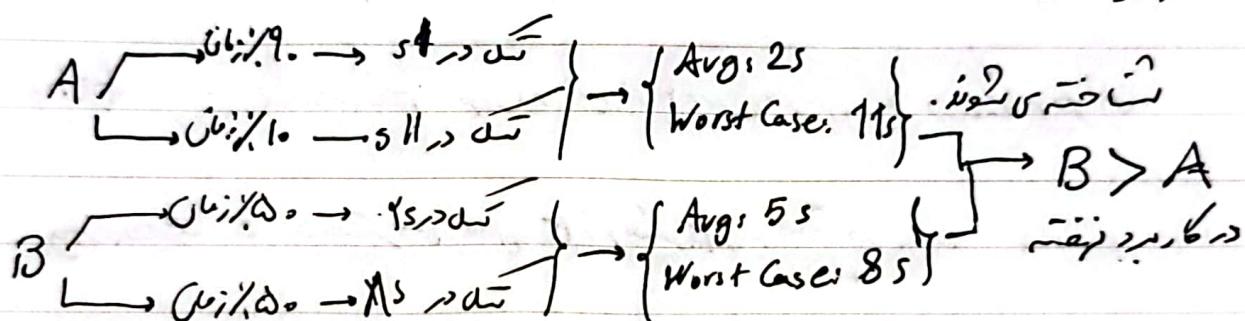
(2) استفاده از سیم کویی از انرژی بهتر

برای درستم های نسبتی عالی داریم از جنبه رسمیون به دلیلین (Performance) میگوییم.

آن Worst case و Average حالات میگویند. علاوه بر این هایی Case.

اهمیت دارد که در Worst Case Deadline بروز رسانید کنند.

جیزین حدودیت های زمانی در درستم های نسبتی را بعنوان Real-Time Constraint میگویند.



وزن بینی (Weight Efficient) سبک بودن درستم های نسبتی بعنوان معنی

از پیشنهاد راهی دارد.

(Cost Efficient) : سطحی دیگر : قیمت بینی

جهتی بودن سایز که memory ، حداکثر Code-size Efficient) کمودیت

و نیز توان به مرتبه ای داشت و بدینکه کمتر کردن آن هفتم سایز کرته  $\Leftrightarrow$  احتمال کمتر برخورد ذرات مغایر از دلایل آن Fault Tolerance ، هنریته کمتر ، سایز کمتر ... است.

خاص ساخته و بعد (Dedicated toward a certain App.)  
Dedicated user interface

اين سئله اخير محدود بـ خاتم را هم منحصر بـ شود و معنی لازم است که سیم خواه  
کشیم. به عنوان مدل Shield Robust (Avoid Fault)  
ات سیم نوشته آبی بینه که سر کند تا از خرد اسکال صبوری کشیم. با فرم  
در سیم هایی که دیجیتال آنالوگ راهنمای بایم دارند برهت است که این د  
Component را زم جایزه داشت که سیم های آنالوگ صفر پسیل را بدله اسکال ولتاژ زاید و من  
امرا سیمی داشته باشند این برا سیم های دیجیتال احوال ایجاد اسکال را ندارند.  
از خدمت توزیع های تولید نموده توسط سیم های دیجیتال همروز عکرد سیم های آنالوگ با میکرو  
لیکن لایه پیوسته از نمی کنند.

۱۴۰۲، ۱۳۹ سیم های نزفته - جلسه چهارم

۱) احوال سیم که تابعی از زمان است که بین سر کند سیم در زمان و تا باید سیم  
کند آنقدر در زمان و که درست کار کرد. است. در سیم های دیجیتال توزیع گشته شده  
از توزیع نهایی سپرده گشود که درین سیم های نزفته خرابی های عادی هستند. اما  
از سیم Weibull سپرده گشود که طبقه نهاد دارد.

از راهنمایی هر سه کردن ( $R(t)$ ) اس است که با  $\lambda$  را با شکل بندی و سایر جواب  
کا هست دیم.  $(R(t) = e^{-\lambda t})$

را کا هست دیم و با کم سود  $T$  هم اتر این را باید همینی را توانیم با

از آنرا از آنرا استفاده کنیم  $\rightarrow$  احتمال خطأ

$$R'(T) = 1 - (1 - R(T))^n \rightarrow \text{با نظر استفاده از } n \text{ مقطع}$$

در این حالت، لازم است که خروجی های instance های مختلف باشند.

کم در صورت نیاز (دست بودن جایگاه اولیه) سایر تغییرات را مشوون کنیم.

## Availability

اصل اینکه سیستم قادر باشد  $t$  در دسترس باشد. نرخ اگر میل سیستم  $\lambda$  باشد  
آنکه دفعه آن ۱ ساعت طول بگیرد، اصل در دسترس بودن آن  $\lambda$  خواهد بود.

در اصل، این ویژگی به نزدیک رسانی را نیز میگیرد.

## Maintainability

اصل اینکه سیستم قابل تعمیر باشند، خود را اطمینان داشت که خطای رفع شود

توانیم

## Safety

که احتمال که نفخ می‌دهد سیم یا درست کار کند یا آنکه دچار اسکال شود، مشکل بسیار

کمتر از افراد اطراف است بسیار نیاید. رابطه ای میان  $R(t)$  که در واحد:

$$S(t) = R(t) + (1-R(t)) f_s(t)$$

## Security:

افراد مجاز به سیم دسترسی داشته باشند و اندسترس های غیر مجاز به سیم حمله کنند.

۱۴۰۲، ۱۳

سیم های نوشتہ - جلسه نهم

دلایل سیم های نوشتہ دیگر کس علاوه بر اتحانی سیم اهمیت پیدا کند و آن رسین

بکی دلایل خاصی است که آنرا منع نمود موجب ابتلاء داده باشد.

این دلایل بمناسبت Real-time بودن.

دیگر کس بر اهمیت دیگر کس که اهمیت دارد، Low Power Design است. این دلایل

از این حرف اهمیت دارد که اعلی سیم های نوشتہ بستن بر باری است و از این حیث

من فوایدیم که طول عمر باتری خود را تا حد امکان زیاد کنیم. همچنین Power Density دوامیم

از باتری های انتزاعی از بقیه انتزاعی هاست.

آغاز

\* در میان ابجاد Redundancy آن دلایل کاری دارد را تا حد داشتم، بدین استقلال این

است که آنونیز هر سیم افزایش سریابی. از طرف نیز هم با کمترین دلایل باید

کاری داشت که این باید طول آن را ترسیم بازدید طعن طی شدن حرفه های لازم باشد!

اجرای Task حد تظرف سود. از طرف نیز رابطه  $R(T) = e^{-\lambda t}$  هم زمان اجرای

طبخ زیسته دهم و (نرخ خطا) زیادتر است و دینامیک این قابلیت اطمینان

کاربرد دارد.

\* در سیم های نوچه از Disk های ایجاد ننموده که اجزایی حساسی

محیطی هستند بنابرین قابلیت اطمینان سیم را کاهش عیوب دهنده. فرض که اگر

در یک سیم که جزو حکایتی داشته باشیم دهم یک جزو دیجیتال داشته باشیم

سیم در حقیقت قابلیت اطمینان جزو حکایتی کمال اسک. این من

$R_c(T)$  برابر با حاصل ضرب  $R(T)$  به طور کاربردی سیم (Disk usage،

خنثی کردن مواردی که تخلی سیم هستند.

Software Redundancy  
Hardware Redundancy  
Checksum → Information Redundancy  
Time Redundancy

سیم‌های Reactive می‌باشند و با جای خود در ارتباط هستند و بـ ④

اساس دارند که نفعه عمل خود را تنظیم می‌کنند.

بعد از این سیم‌ها براساس event گرفته شوند. در عین حال، هر خودادی ترتیلی است.

(Event-based) Action توسط سیم گشتند

براساس تشخیصی paradigm چه رسالاتی پاسخ داده می‌شوند:

۱- درودی و خودمی‌ها چه هستند؟

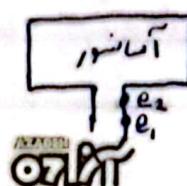
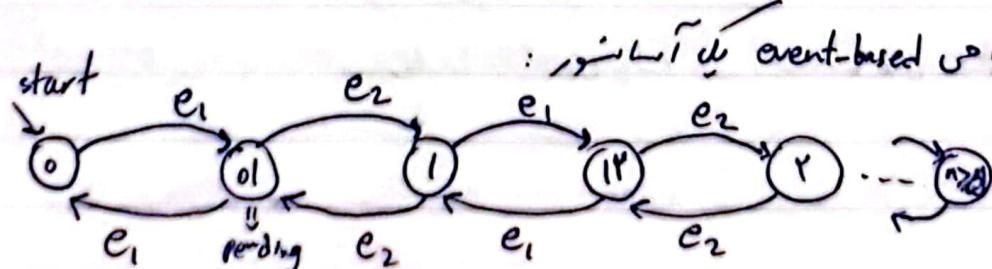
بعنوان فعل درودی و خودمی‌ها Von Neumann داده و درین فرضی سیم‌های

نخست events based Automata می‌گذارند. (نمایه ایجاد شده است)

۲- نظرسنجی بین افراد چه می‌باشد؟ (کشیده، سوپر، logical یا فیزیکی)

(- جمع، تجزیه، ... ) می‌باشد operation

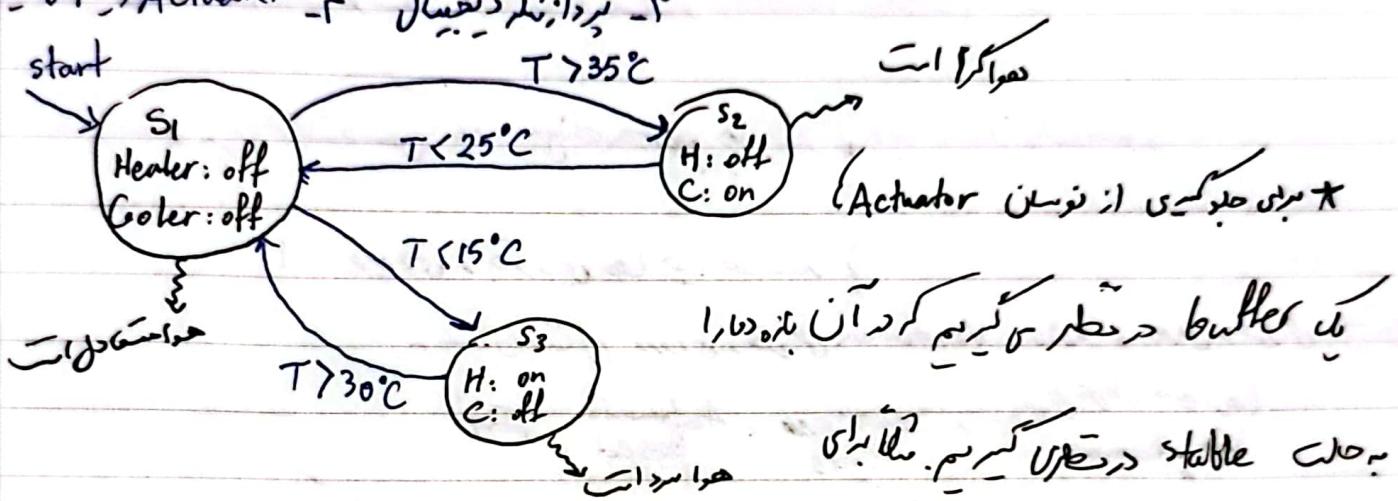
۳- از سطح عالی چه سطحی هستند؟ (Von Neumann, CISC, RISC) می‌باشد



specification  
در اصل به که FSM هر توانیم نیازمندی ها را نهان داشتم.

۱۴۰۲، ۱۲، N<sup>۳</sup> - حلبه ششم - سیستم های خرقت

عمل اجزایی: ۱- سنسور ۲- بُند ملک آنالوگ- دیجیتال  
Actuator - ۳- پردازنده دیجیتال



در اینجا  $T \rightarrow 25^\circ\text{C}$  و  $T \rightarrow 30^\circ\text{C}$  در برابر است  $S_2$  است. اگر این بازدید خیلی کوچک باشد

هر ۷ دانه این ایست های اول و دوم فرمان می کند. (در هر ایست ۳ هم بازدید  $\rightarrow 5$  داده)

نیزه ای داشتن چنین FSM ای را بگویید که while(1) نهست و پیاده سازی کرد. توابع که

Cooler, Heater و Driver را کنترل کردند on, off

و نیزه ای داشتن این action ها را صراحت نمایم. ~~fragile~~

محضی بین این انجام آئینه در بحث ایست  $\rightarrow$  فعل مستمر رخدادی داشتیم. این interrupt

از طرف سند سسیس مرتبه دار اطلاع دارد.

\* درستی های reactive استوار داریم که

دستور طور توانی به مالی ۷۰٪ نزدیک (در ۵۱) و در نتیجه مردم براش چنین وضعیت های

alert را تسطیح کرد.

Automaton-based

Programming

- از شرکت های FSM جلوه سیری می کنند. ترجیح مردم می آن را باشد در super state.

دانش بایم.

verify - Automata Module ساده تر خواهد بود.

- تولیه اتوماتیک code ساده تر خواهد بود.

۱۴۰۲، ۱۲، ۱۳

ستم های نهفته - جلسه هفتم

Model : مدل سازی از

زبان طراحی ساختاری؛ چنین زبان نباید تفسیر های متعدد داشته باشد بلکه توصیف

دک همراه آن بایان باشد. همین در توصیف این باید تعداد و ترتیب متن می

محض های تخفیف باشد. از طریق توصیف های حاصل نباید ناقص باشد بلکه ملک سیم

آزمایش

را به طور کامل توصیف کند.

\* لازم است که تمام داسته باشند که میتوانند ممکن است (ویژگی) به داده ساز را از خود توصیف داشته باشند.

⇒ دیگر کمی های لازم فبرای زبان توصیف و

۱- لازم است که زبان حاصل خصیت سلسله مراتب داشته باشد.

- سلسله مراتب ساختاری: با Component ها سروکار دارد. (نمای آرایه ای UML)

- سلسله حرایتی رفتاری: در سطح الگوریتمی است. ارتباط state های مختلف

۲- زبان بندی چه می باشد؟

- خود زبان منزدیکال می جنبه از تا خبر و زبان بندی است

- ترتیب اتفاقات (fine-logical) هم می جنبه از زبان بندی است.

\* زبان CTL یک زبان توصیف صرف است که قادر است execution ندارد.

۳- رفتار این زبان ها بر اساس وصفیت سیم است (state-oriented).

- اتفاقات ها چنین می باشند که رابطه ساده دارند.

۴- مانند بذای event-driven بجز سیم های نهضه لازم است که مایل

. (Reactive Systems) event handling

event می سکن است internal یا external باشد. مردم اول

توسط Component های خود سیم و دری توسط سیستم ایجاد می شود.

آزمایش

۶ - لازم است که support و بایس پاره سازی نظر اقتداری / سخت انتشاری انحصار شود.  
ماعینت مانیتورینگ را نمایرند.

- نحوه های خطا
- Design → ۱ - خطا در طراحی (D-M)  
آنچه نیز نهاده باشیم verify کلده باشیم.
  - Implementation ← ۲ - خطا در پاره سازی (I-M)  
در هنگام ساخته شدن تغیرات طراحی، خطا پیش بگیرد.
  - External → ۳ - خطاها در اثر گذشت زدن (aging-wear-out)  
نرخه کامپیوتمن فرکنس تغییر در اثر زمان
  - Disturbance ۴ - خطاها در گیرنده (E-D)  
ب یکدیگر شفید شوند

۵ - قابل اطمینان سیم را با استفاده از رفع ابرم های توانی و حصنی و همچنین

ایجاد کنیم verification

۶ - لازم است که exception handling داشته باشیم تا بایزی کامپیوٹرها را سلن

نمی این اطمینان را داشته باشیم که سیم حادثه ای - صورت این

به حالت fail safe چایه برسد.

۷ - باید هم زنگنه سازی هم پشتیبانی کنیم، معنی چندین تله به صورت هرگز مان بتواند

به حافظه و منبع سیم دسترسی داشته باشند. درجه امنی که تضاد منافع اینجاد می‌کند

با استفاده از - بلند Mutual Exclusion از خطاهای هم زمانی جلوگیری کنند. در اصل لازم است که Synchronization داشته باشند.

۹- الگان های نرم افزاری (switch-case, if-else) در دسترس باشند تا از آنها

نه بسی بزری تغییر ممکن است داشته باشند.

توضیح ۱: در زبان های برنامه نویسی (C, C++, Pascal, Fortran, Ada, COBOL, FORTRAN, COBOL, PL/I, C, C++, Pascal, Fortran, Ada, COBOL, PL/I, etc.) در صورت

کردن مراحل با هم مطابقت نشوند، تغییرات میان مراحل ناشی از اختلاف انتظارات را توصیف کنند.

Human-readable \_ ۱۰

۱۱- محوّل در I/O Device به صورت غیر اسازنده است. یعنی

بجا کیبورد و مادرس بجزی دستگاه دیده شوند. پردازشگر آنها را دریم.

۱۲- لازم است که متعاقبیت و معتبریت توصیف نیازمندی های غیر عکلاردي را داشته باشند.

Reliability - Size - Power Consumption : باید

۱۳- لازم است که مدل مکانیزم (Model of Computation) MOC

۱۴۰۲، ۱۲، ۱۷ - سیمراهی نهفته - حلبه هستم

۱۴- مدل های کیاسن (MOS) در دنیاگردی، از E/S های مخصوصی کنند.

توانی

۱- درین حاسوس هر component را تو صفت کن.

۲- درین اوضاع این component های تکی را هم تو صفت کن.

کراف ماستن ها (Dependence Graph) 

پس از تعدادی تسلیت تعدادی محدودیت (که به مکانیک اینها نیز داده شوند) دوست می شوند.

مثلاً این است که اطلاعات اضافی مانند Deadline، Arrival Time تسلیت داشته باشند.

مثلاً این است بفواهم منبع مشترک بیش از یک دارند. (

مثلاً shared memory باشد.

درین کروی اولیه مرتبه ترتیبی بازگشایی hierachical تعدادی از تسلیت ها را group نمی کنند.

این امر این احتمان را به حافظه دهد که سر برگاهی communication، کامپوننت، چشم دهم (با این درجه

مثلاً بازگشایی context switch، هر کدام force کرد این معنی دارد.

اجرا شوند). مثلاً این کار این است که دیرتر توجه خطا را شودم. حرکات دست

تسلیت ها همیزی باشند، علاوه بر کدامی checkpoint هستند.

Communication Models 

۱- shared mem : چند کسیست: اینها حقیقت مشترک استفاده کنند. بدلیل همزمان

مثلاً این سیستم های مشترکی این اینها زمانی هستند که همیزی داشته باشند.



مثلاً

۱ - ساختار buffer داشته باشیم و رسی Non-blocking Message Passing

آن message خود را بتوسیم دهی کار خود ادامه دهیم. از همان این روی

این buffer این C++ overflow است.

buffer, > read ( ) : Blocking Mess. Pass. , ساختار

انجام نمود و لیکن زمان که receive انجام نمود، سراغ تکه بعدی نمی خودم.

در این حالت ممکن است overflow باشد

گاهی از اینجا

علی Blocking MPN ۲ (Extended) : Acknowledgement-based msg. Pass. - ۱

کنیم ولی حداقت کنند و باز هم بین اینها که ارسال کنند

از حالت block خروج نمود. در این روش معلم اسکرپت buffer را خالی کنیم یا نیزم از الزاماً ندارد.

سازماندهی (Components) داده دهنده (Messages)

۱ -  $\text{FSM}_{\text{based}}$  : به کم state ها و مردمی خروجی هایی که در حالت های سودمندی

این state ها جای بجا می شود. در این این Component based

ها از بین می تصور کرنشه سیستم خود را هر چند  $\text{FSM}$  خود را دارد.

CCFSM  
Communicating FSM

بررسی دستگاه های مجزا در در خروجی دستگاه های مجزا : Discrete Event Systems - ۲

کتابت ا نظام می شود.

→ la Program : به صورت Von-Neumann - ۳  
حافظه و انتظام کتابت می بسیس آن انتظام می شود.

Machine Learning : رایج در دنیا Data Flow - F

## State Charts

۱- زبان عام برای نمایه بر حکم ریت های : ۱- نوع تعریف زمان (از logical به physical)  
۲- نهادن ساختار ۳- پیوین از FSM ها و جرد دانش را.

۱- داصلی افتراقی ای برای داده CFSM ایت. این طایفه component های است.  
۲- اساس حافظه ترکی است.

۱- نهادن نهادن به این دلیل از DFA های تفاوت می کنند هر کارکرده خواهم می سیند  
باشد و متاثر Design Time ای دستور همه جایب را بپیش.

super state □-۲ basic state ○-۱: در این طراحی دو دسته داریم

\* در برخی دستگاه های خاصی or داریم که کدی در هر لحظه، مقاطعه ای داشته باشند (OR Super State) می توانیم باشیم.



\* در درود و خروج به مُحلن اسَت Mediatority super state بعضی مُورّد

بعنی مُحلن اسَت بِلَى state خارجی به داخلی superstate دیده داشته باشند.

این مُثُد را با در درجه توان ارضی کرده دید این نام Sup. State داشتند.

مُثُد Default State - I Mechanism super state : Default State - I

مُثُد default state شروع کریں.

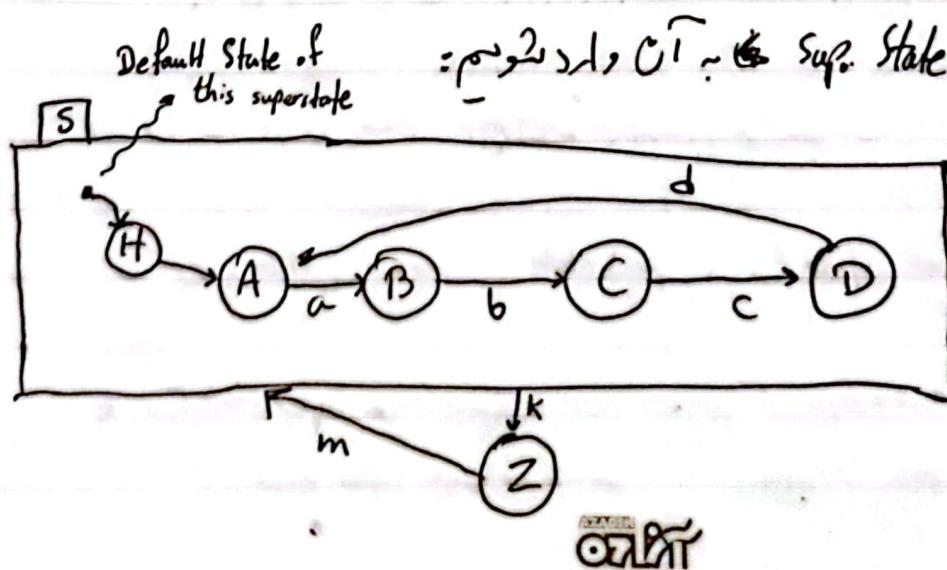
: مُحلن اسَت که برگشتن به حالت default باعث History Mechanism - I شود

شود که خطای Default Sup State, overflow. به معنای مُحلن A (بعنی رعنی)

در حافظه باشد و آنکه سر از حافظه، حافظه خواهد بود و سر از ریکارڈ از حافظه

در بارهی A به کردم، حافظه روند، اگر پیشتر سر جدید روند

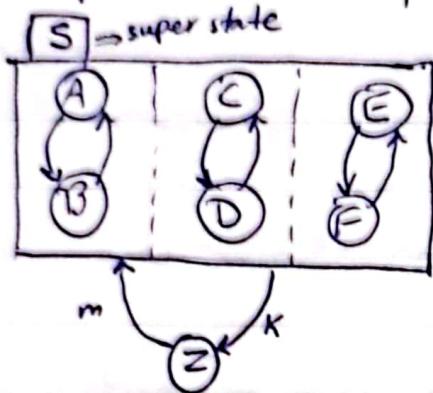
کند. بنابرین خوب است که history state نله داشتم که سر از بازگشت



۱۴۰۵ / ۱۱ / ۲۰

سیمای نوشتہ

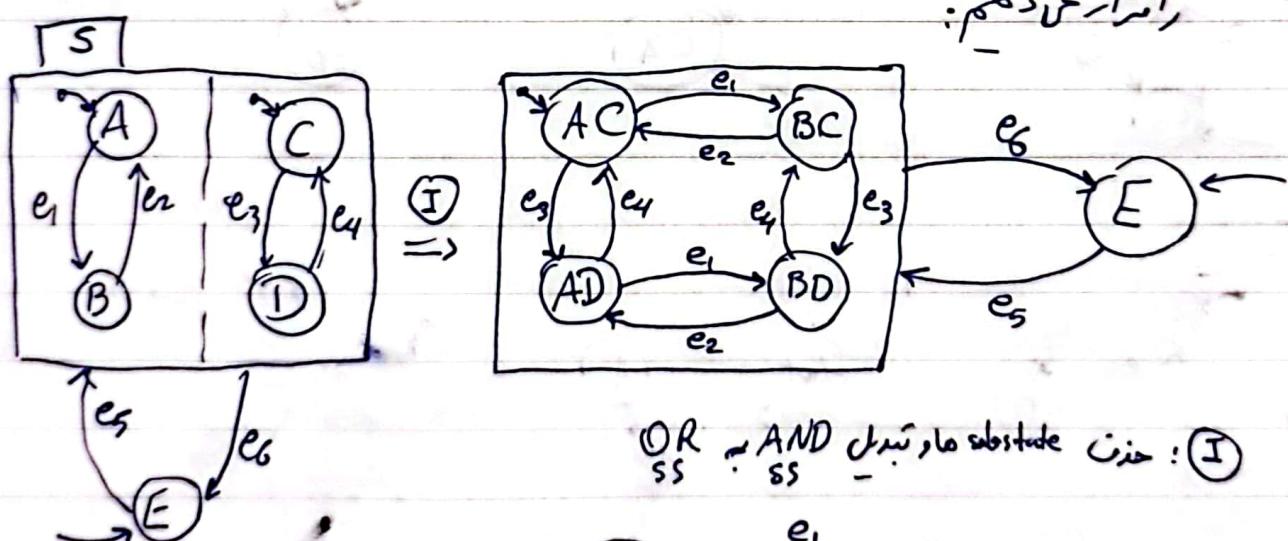
super-state یا substitute میں چال میں AND Super States



عمل: single core + برس پر دستور State-chart flat نحو:

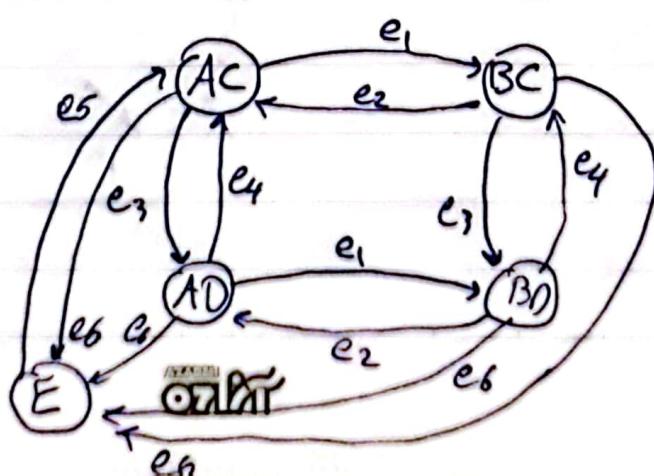
basic state کام کر کر داده، احتفظ کر داده، super-state ہمارا باز کر کر داده module

راترا-سی دھم:



superstate میں basic state

(II)



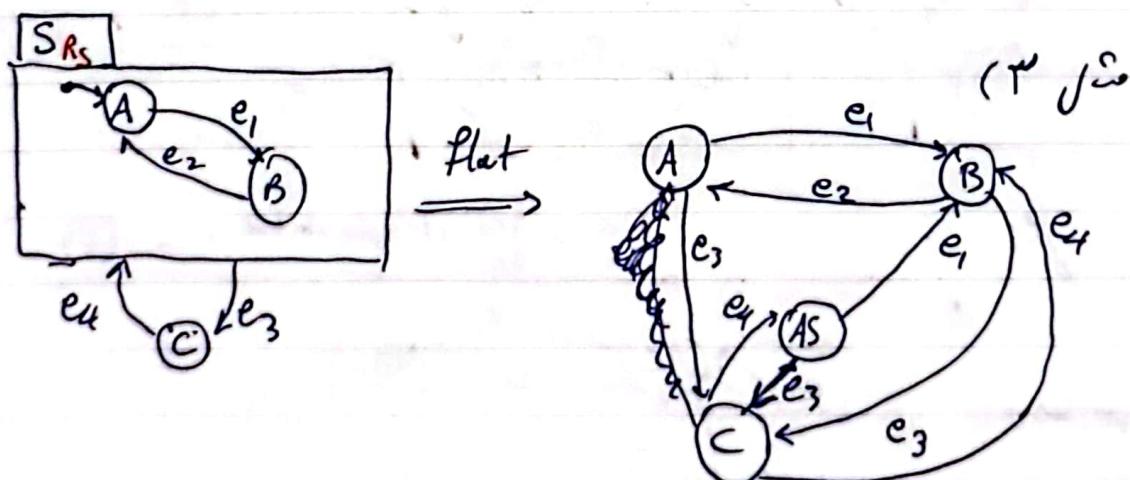
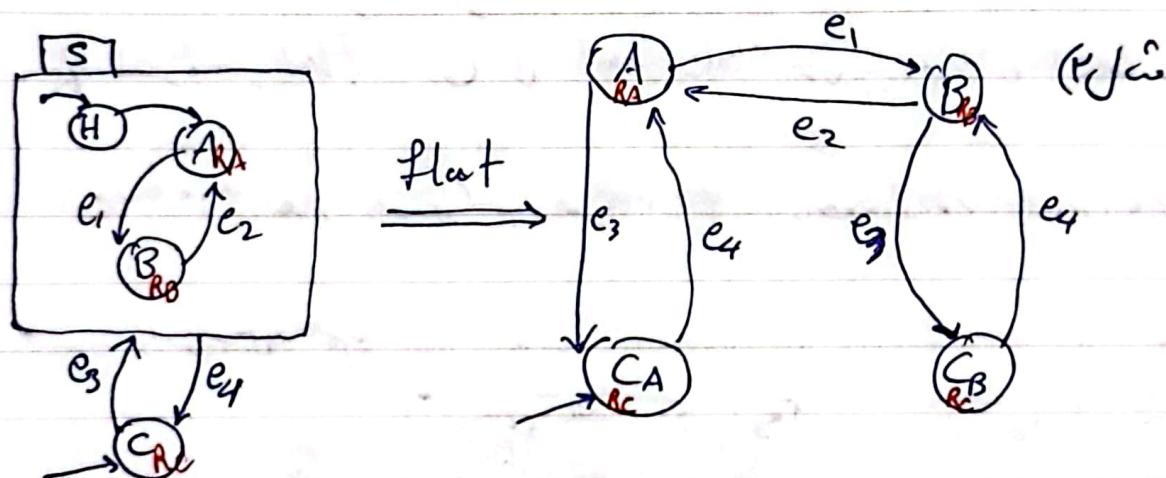
عملیاتی (Reaction) (Moore) میں داداںی اس کے باوجود FSM کو مکمل کر سکتے ہیں

action کو redundancy میں (basic step) کہا جاتا ہے



دوسرا صورت بارہ ایتھے جگہ اینے اکسن ہے

دستخط لیبری سے کہ خرفا تباہی کی وجہ سے انجام نہ ہو.



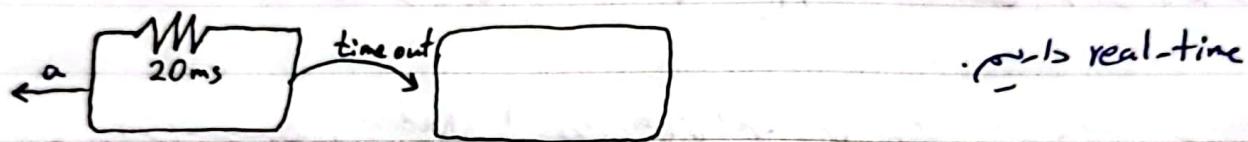
## سیم های زمانی - جلسه دهم

### Timer

سیم های زمانی به دید physical نسبت به زمان داریم. از طرفی مدل توصیفی

این کابیلتی را به ماس داده که Timer داشته باشیم. یکی از علائم این

آن دیدی به زمان از حیث ارضا کردن دلالین های زمان است که در سیم های



در شکرته بالا به کلایلی تایمر ۲۰ میلی ثانیه ای، یک دلالین ۲۰ms برای رخ داده

اینست که از اینم.

\* سن است که داخل علاوه بر زمان دلالین، دیگر reaction

برای خود در پطر تغییر میم.

### Edge Labels

یک مدل FSM ها نداشتن variable بعد. درین صورت برای طراحی counter ها

نیاز نمی باشد از اینها state و count های اضافه در تقدیر نماییم.

در اینم با استفاده از edge label خاص دیگر از این condition خصی از

سیم، به عنوان State Chart که variable، reaction را تغییر دیم.

قالب تعریف کردن edge label

Ex) KeyOff [Battery == full] / on: 0

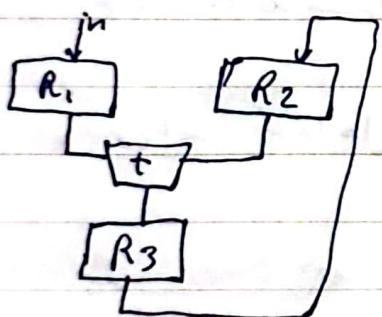
### Execution of StateCharts

در این زمان، در زمان اجرا به صورت Atomic اجرا می‌کنیم. برای اینکار، ابتدا

از خود پر اصل (main variables) سازیم (ماز) و در ادامه همین ناز shadow

پس از تغییراتی shadow را تقدیر دهیم کنیم. پس در نهاد (User) خود پر shadow

امضا دهیم و مقادیر اصلی را تقدیر دهیم کنیم (ماز)



Eval:

$$R_1\_sh = in$$

$$R_2\_sh = R_3$$

$$R_3\_sh = R_1 + R_2$$

Assign:  $\frac{R_1 = R_1\_sh}{R_2 = R_2\_sh}$

$$R_3 = R_3\_sh$$

(↓)

Atomic  
Execution

در حقیقت در درست بالا نشان داده شد که هر کدام از خطاهای موجود

در هر ناز را می‌توان با هم جابه جا کنید و همچنان دستگیر کنید

نیز.

Mealy or Moore

در طراحی های خود می توانیم هم حالات Moore را داشته باشیم و هم حالات

مختصر رخدان دهنده (Moore state) می باشد که با reaciton و Moore تابع

کہ در پر میں ایسا ہے جس کا خاص اتفاق ہو رہا ہے۔

1403 / 1 / 19

## جلسہ یازدهم

تیمهای تجربه نمودن و loop مخصوص به خودشان (Hardware in loop)

## (Embedded System HW Components)

- ① Sensors: Captures real-world data and sends it to the processing unit (physical quantities)

- ② A/D (Analog to Digital converter): Here we want to convert continuous signals, possibly generated by sensors, to discrete ones that can be processed by digital (binary) systems. To do so, we can use a sampler,

- ② Sample-and-Hold circuit : Clocked transistor + capacitor  
 ب دخل سریز  $\rightarrow$  transistor که لحظه‌ای  $\rightarrow$  clock می‌خورد و خازن  $\rightarrow$  از سریز می‌شود. علی‌  
 تازیان طلک بعدی به مقدار صحیح binary را نهاده می‌دارد. همین آنکه ولتاژ مایس  
 پس و (با صفر نزدیک شدن) مقدار صفر را ذخیره می‌کند.

- ### ③ Embedded Processor:

پس از اجتناب کردن processor های نیاز است حافظه ایت. همان طور که قبلاً توضیح داشتیم، بهتر است از حافظه های پیشینی و ذخیره (miss) (Cache) دارند و کایلیست پیشینی را در آنها استفاده کنیم.

لیکن دلیل از ورگیرندهای Processor هایی است که لازم است کم توان و کم مصرف باشد.

ویا که هم در مصرف باشند نباشند بهجهنی سازی داریم. از طرفی دست ترا کم توان

با این روش برای سیم خطرساز خواهد شد هر کس قواند دلای سیم را بالا ببرد

رباعی سطح interconnect های خرابی سفت اتکاری نیست.

از طرفی این دلای زیاد لازم است که ببکشید و درجه حریقی را کنترل کنید

را خنک کنیم. که خودا بین باعث نشود که اس ( bottleneck کاتیک ) شوند سیم

( قابلیت اطمینان و محقق را دارد )  
و از لحاظ failure از fan است که از استفاده مطلق

مالی سعی کنیم عوامل دلیر تسدیل ( بطری و پژو ( حالت peak ) را کنسل کنیم .

### (Energy vs Power) Consumption

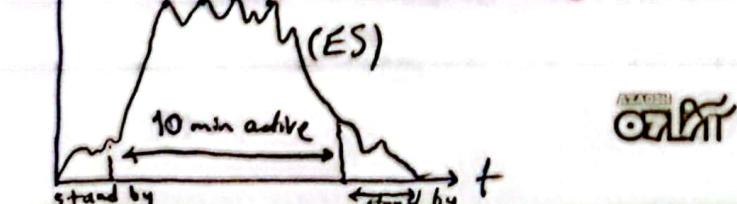
پس از آن ساخت مخصوص بین سیم های نسبتی و سیم های HPC در این است که در ES های فرعی

مصرف  $P_{cool}$  به صورت short term، کوتاه مدت، bursty، داری باشیم - و مصالح

سیم های HPC باید زمان طولانی و با توان بالا لازم است که کار کند و در آن ماباید

آن نشاط peak - لفظ  
خط سازی برای این اهمیت دارد.

در صورتی دیگر  $P_{cool}$  به صورت long-term cooling داشته باشیم .

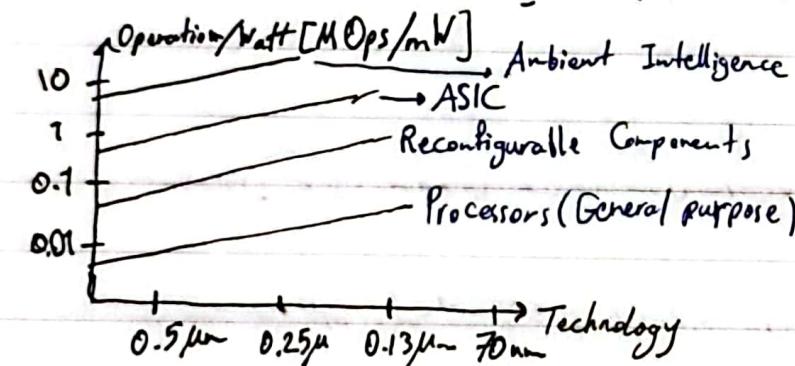


از آنچه که مذکوری C ASIC بین گردیدهای حاصل مطلعه بهینه سازی می شوند مقدار علیم

که باز از هر دوست صورت انعامی مهد بین از مدارهای عام مطلعه است. حس

تیمهای (Ambient Intelligence) AMI از هدفی این حواردهم نهاده است.

عملیاتی تصریحی **AMI** : آینه آن به حالتی است؟



تیمهای نیمه سیمی ISA محصول خود را دارند و دلیل آن این است که سخت افزار خاص

متوجه دارند که برای بجزی پارامترها مثل مصرف انرژی و ... بینهایت زیاد باشند.

PC compatible

### Microprocessor vs Microcontroller

میکروکنترلرها یک پلیچ کامل از اخباری تیم، یعنی میکروسور اسکرینز می باشند

که بـ **میکروسورها** دارند دلیل اسکرینز در ساخت کنترل اجزا را که رهم نمایند.

آن از طرفی باعث می شود که **reliability** بالا بود چرا که ساخت کنترل از دارای

اَصْحَالَات هم که کاهن تر می‌شود و در نتیجه اَحْلَائِه خطا در سیستم اَسْعَال داده  
امداد شود که شود. از طرفی همیک کرد که بعد اَصْحَالَات و جنس onchip interconnection  
این باید می‌شود که تغییر مدار ساقم کم بشود.

۱۴۰۳، ۲، ۲۱ - حلہ دو از دھم -

### Memory in ES:

لزمان است که به لحاظ صرف اُنُرُز بینه باشد، درین حال سرعان می‌باشد داشته  
بُشَّاب طور که predictable باشد، بیش در صورت نیاز به آنگه هم داده ای را  
توانیم در صورت بازگشتن دستیافت کنیم.

در پرسی توان سیم‌های ثابت هم توان static leakage مطلع است و هم توان dynamic  
حال در سیم‌های نفخه جامع این توان ها مستطیلیان است و نیاز داریم که  
این جمیع توان حصری بین ۰.۵ و ۰.۶ داشت. باگذرا زمان دباقتر اسی پیشنهاد  
مدارات بُشَّابی صرف اُنُرُز بیشتر لازم است که این حدیثت قوان را جبری تر دنبیل  
کنیم. مُحَلَّ اَيْلَه توان زیاد بگو در این اس کم باید اتراسی دعا کیم  
حر شود که خود این اس بینه روس اجزای سازنده مداراتی کند از دو می‌توانند خواهند

محاسبه افزاشی ایجاد کنند. از طرفی حلقو آمار پشتگاه در سال ۲۰۰۵،

حدود ۲۰٪ از معرفی اینتری قرط <sup>memory</sup>، حدود ۲۰٪ از معرفی اینتری <sup>processor</sup>

است.

۵ اندام حافظه. ① SRAM ← ۳ ترانزیستور، + خازن  
← ۴ ترانزیستور

به دلیل استفاده از خازن ساختن ات DRAM دلیل تصنیف دلار سود دل از نمای

در حافظه های Timed refresh، باعث معرفی اینتری حافظه حالت عالی داشته باشند که خود را باعث معرفی اینتری حافظه حالت عالی می کنند.

در حافظه های DRAM معرفی اینتری بیشتر دارد و در عین آن هم کمتر از دل

از طرفی از این تک خواهد بود.

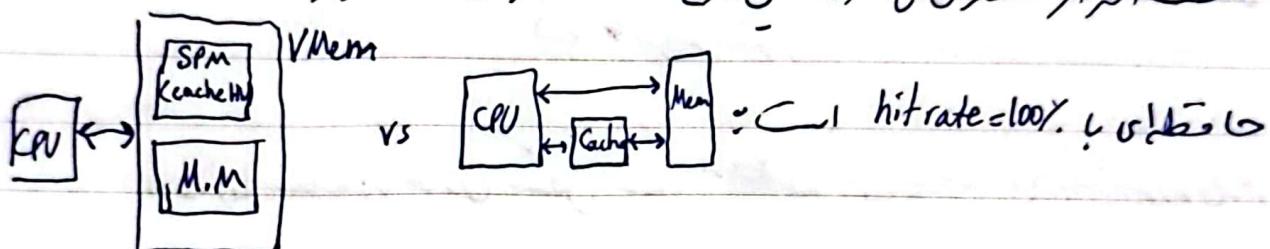
\* معرفی تک دل زمان دسترسی به یک خانه از حافظه با اتراکس سایر حافظه های موجود

خطه افزایشی می باشد. (برای این این تعریف است).

لی راهکار برای اینه بتوانیم از سرعت cache استفاده کنیم در عین اینه عدم قطعیت آن را از سی بیزیم، از SPM (Scratch pad Mem) استفاده می کنیم. این قطعه بدهانه ساخته

می باشد و نکاته ای که در جدید دارد این است که تو سطح زیر این را میتوانیم انتظام

می شود. در اصل یک حافظهٔ جازی سی DRAM و SPM در مطری کسریم که علاوه بر این می شود که مفهای آدرس دهن Code می شوند. این در حلقات کردن می شود که قویتر نصی آدرس دهن بر ORAM تناظر داشته و حافظه‌ای فرضی می شود که قویتر نصف افزار کترل می شود. این نوع استفاده از سخن‌افزار Cache باشد می شود که



بعد تبدیل داده آنلوك به دیجیتال در صورتی که نرخ تحویله برداری

از سیلول وردی ۲ برابر بالاترین ترکیب درین تراجم Single Merit به تبدیل خود رسانیده

۶۰ تراجم بعداً از روی این سیلول آنلوك اصلاح راه رانیده کنیم

۱۴۰۳، ۱، ۲۴

### سیم عامل در سیم‌های منفذه

برای ساختن سختی‌های IP ها از Intellectual Property برای ساختن سختی‌های IP ها از

کرده ایم این IP ها ساده‌ترند: (1) Hard (2) Firm (3) Soft

دست‌ادل یک تعلیمه دریلک یا VHDL را توانند باشند (می‌توانند قابلیت edit داشته باشند)

دست‌دادم را توانند که قطعه که مستلزم شده (netlist) باشد در زیر داشته باشد

زنگنه

آخرین Layout سافت افزاری است. نهایتی نیازی نیست از آبتدا ملیت خود را طراحی کنیم.

۳) دینگری های OS های ES :

- چنین سیستم عامل های لازم است که قابلیت شخصی سازی داشته باشد Configurable

به دلیل خصوصیت داشته باشد. نرم افزاری بجز پارامترها Compile-time و اترانسایت

دهم تابعیت در runtime افزایش داده باشد. یعنی Scheduler، ریدر طوری

۴) تغییر دهنده (Processor)، لایه سفت افزاری از توافقی از سیستم های FPGA استفاده کند و ISA کم کنیم که آن ها کابل تغییر هستند.

- در سیستم های دینگری، UI به صورت سفارشی شده Disk and Network handled by Tasks

و رسماً وجود ندارد دینگری نیازی به Driver هایی است که دستورات نیاز داشتند

Task کنترل دستورات I/O انجام می‌شوند.

۵) آنچه که احتیاجی که Protection is optional ES های سیستم های اینتل

متند و بر اساس این حدود هستند لازمه اند، نیازی به حافظت

نداشند.



Task Interrupts : هنوز کردن اینترپرتوت Task برای ترسیم خود از OS محدود نیست.

انطم پر گشته.

- دنبالهای ES ذاتی هم را دارای قدرت React-time Capability می‌باشد.

به صورت خاص سازنده RTOS هستند. این OS های توکن است.

زنجیرهای دوایسه بهم را نهاده می‌بینند که

### ۳) تعریف و نیازمندی‌های RTOS

\* چنین OS ماین توجه اصل خود را در رساندن دلایل تکمیل کننده اند. یعنی

باید پسینی زنگیری را در سیم داشته باشیم. یعنی باید نتایج تکلیفها را

خرموده طول می‌کنند و نتیجه هی توان در به اتمام رسیدن آن مایه

از آن زمان مخصوص اطمینان داشت. این دلیل این نیازمندی دینیک ترتیلها

نمی‌بینند زمان را به مامن خود. در عمل، چنین سیم عامل‌های باید سینه درستی باند.

بنابراین از سریع Scheduling در RTOS های تراکنی برخی تضمین هارا به صورت

routine بلاییم و با شخص کدن زمان بندی اجرای تکلیف‌های دلیل خود را در زمان offline

مرغی به تخفیض تکلیف‌ها بیندازیم.

CS CamScanner

## RTOS Kernel ②

که اصل این kernel حدیرت منبع و timer ایجاد می‌کند. از طرف دیگر

ارتباط تسلیم هم می‌شوند.

\* در تفاوت Standard OS ها را به صور افونه‌ای در RTOS می‌سازند که تهدید

crash کردن Stand. OS را در RTOS می‌کنند. در اصل non-RT

نخواهد بود.

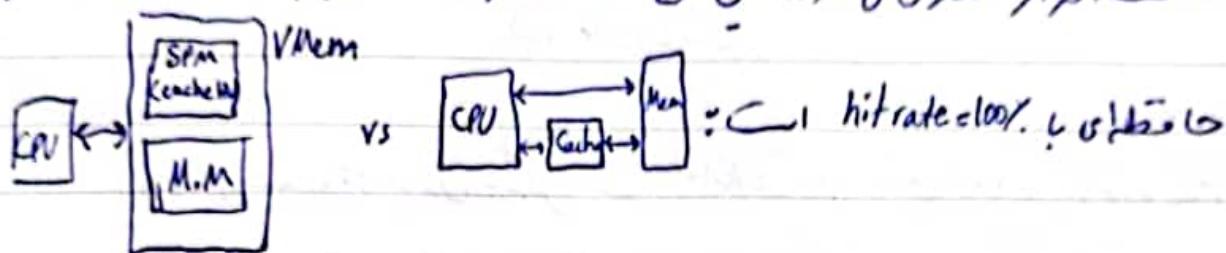
\* دفعه انرژی در سیم‌های نیمه حیی است. معمولی RTOS I می‌تواند در میان ۲ برابر

از RTOS 2 بسیار کم باشد حبود استفاده می‌شود و نهایتاً

Power, Energy, and Reliability  

سیستم استور. در اصل یک هارتفلچه حیزی سیستم (DRAM و SPM) نه مقداری کمیر سیم که علاوه بر آن داشتند که مقدار آدرس دهنده DRAM و SPM بینشند. این در جمله است که در مقدار آدرس دهنده DRAM متناظر با سیستم هارتفلچه ای داشتمه ای نه تنها مجموع سیستم استور که قویتر

سیستم افزایش کرده استور است. این نوع انتخاب داده از سیستم افزایش کرده استور که



در بعده تبدیل داده آنالوگ به دیجیتال در صورتی که نرخ تغونه بردار  $\bullet$  D/A Converter.

از سیلول ورد در ۲ برابر بالاترین تراکنسندریون تراجم Single مربوط به تبدیل فوریه باشند،

و توانیم بعد از آنها  $\rightarrow$  sample و  $\rightarrow$  میانوار آنالوگ اصله را بازبینی کنیم

۱۴۰۳، ۱۷

### ■ سیستم عامل و سیستم های ذهنی

برای ساخت بخوبیدن پردازش تولید ES ها از توان از Intellectual Property ایجاد کنند.

کرده این IP ها ۳ دسته هستند: (1) soft (2) firm (3) Hard

دسته اول یک آنالوگ دریلک یا VHDL و توانند باشند (سیستم قابلیت edit).

دسته دوم و توانند یک قابلیت که متزر شود (netlist) باشد در نزدیک داشته

آنالوگ

آپلیکیشن Layout ساخت افزاری است. نامی بخوبی بسته از آنها می‌شود را طراحی کنیم.

### ④ دستگاه‌های OS مای ES ها:

۱- Configurable: جیز سیستم عامل فایل لازم است که قابلیت شخصی سازی داشته باشد.

بهای ES خاص داشته باشد. نرم‌بتن‌هایی مخصوص پردازهای FPGA را تراویس

دهم، نیز در runtime افزایش داده باشد. رسیده Scheduler و صورت

۲- تغیر دهنده افزایشی (Object Oriented): میان ES و FPGA اینها کنترل

یا از processor می‌باشد که ISA آنها تابع تغیر می‌شوند.

۳- دستگاهی: Disk and Network handled by Tasks

مرسوم وجود ندارد و بین نیازی به Driver داشته و در صورت نیاز تو ساخته

کنترل دستگاه‌های I/O انجام می‌شود.

۴- آنچه که احتیاجی کر غایب ES ها: Protection is optional

هندسه برآوردی در آن حدود هسته را می‌تواند تأثیر نیازی به چانکه

نداشته

و Task چون که در مورد یک Interrupt باشد : Interrupts not restricted to OS.

انظمه من شعره.

ذات نشاط ES دلایل تکرار اس و بایرین Real-time capability - ۱۵

به صورت خاص سیستم RTOS (Real-time OS) های تراویته را می‌توانند.

ز شخصیه‌ای دوسته بهم را دهتر مدریس کند.

## ۴) تعریف و میزان نهادهای ATOS

\* چین ۵۰ هاین توجه اصل خود را رساندن دلایل سند های حکم که از نظر معنی

باید دیگرین زنگیری را دستیم داشته باشیم. هنری برا نیم سه ساعت تکه‌ها در

و سید آن مایه ورثت-Case حی متعدد طول کشیده در نتیجه هی توان در بسیار اهم رسیدن

از که زن مخصوص اطهان داشت. این دیگر اسما نباشد زنی دستی ترسیده

و درست زمان را به مامادر. در عمل، حينی سیم عاملهاین باید مبدأ و سرعت باشند

پیک چلوکسیل از سری RTOS Scheduling های توانیم مرتبه هایی هارا به صورت

offline نتیجه را با شخص کدن زمان بین اجرای آنها در یک جبرد و در زمان routine

مرنی به تخفیف تسلیهای پردازیم.

حال توجه داریم که  $P_{total} = P_S + P_{D(static)}$  ،  $P_S \propto V_{DD}I$  ،  $P_D$  ثابت است.

میں پہلے  $P_{static}$  کو سمجھتا ہوں اور  $P_{dynamic}$  کو سمجھنا بے نتیجہ ہے۔

تمان یا کامن گایا بود و در نتیجه محل اتفاق فوتی گایا بود.

مثال : میدان زمینه crusoe میانه خود را در 32 سطح می ۹.۱۷

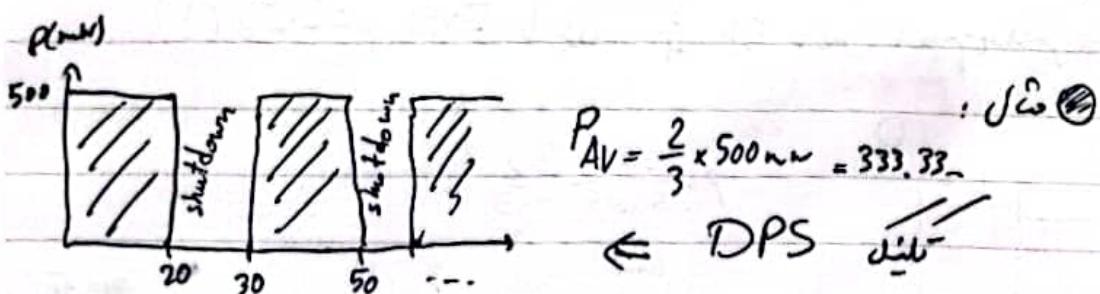
گردن، همین زمانه ساره هم می تواند بین  $33\text{MHz}$  و  $700\text{MHz}$  باشد

(V<sub>dd</sub>  $\propto$  f) inverso

برای تغییر فرکانس دارایی دستگاه خود نماینده freq. Register است و با توجه به آن DVFS می‌تواند انجام داد.

مطح DC/DC Converter کی مدد سے اس کا ولتی کنٹرولر اور اس کا ایک ایجاد کرنے والا set 1, 2, 3, 4 میں مذکور ہے

کے VCO فرکننس طرزی مدار را بعد از اساس V جدید تغییر دهد.



۴۶ f shutdown م استفاده کنم و در مقابل DVFS چیزی

$$P = \frac{t_{exe}}{t_{exe} + slack} = \frac{20}{20+10} = \frac{2}{3}$$

مسار را که حدیغ داشت می‌توان کسر را بدست:

$$P_D^{(1)} = \frac{\alpha C_L f V_{dp}^2}{\rho} \times P^3 = \left(\frac{2}{3}\right)^3 \times P_D^{(0)}$$

$$\Rightarrow f \times \frac{2}{3}, T \times \frac{3}{2} \quad \text{outfit} \quad P_D = \frac{\alpha C_L + P_{AP}}{P_D^{(0)1}} \times P = \left(\frac{2}{3}\right)^3 \times P_D$$

$$\Rightarrow P_{AP} = 500 \text{ mW} \times \left(\frac{2}{3}\right)^3 = 148.15 \text{ mW}$$

۴) خاتمه خطاهای اولیه (original) و خطاهای follow-up را متوجه شویم و آن internal disturbance، aging wear-out را نقلی کنیم (متوجه خرابی: ۱) خرابی ۲) بیادگاری ۳) فرسودگی ۴) خود

بعدتر لازم است که Cause & Effect ها را بایسیم و عصینی: Timing آنها

میزان زمان طول کشیدن و تسانی خود خطاهای هم توجه داشته باشیم.

۵) <sup>نحوی نزدیک</sup> بازگشته به interface های خوب تعریف کنیم اجزای داخلی را به خوبی پیمان کنیم.

۶) لازم است که اجزای سیستم خطاهای مستقل از هم داشته باشند و در هم ارتباط نداشته باشند.

۷) <sup>اعتماده نشوند</sup> اجزای سیستم فرض براین است که هر چند در آن بدست کاری کند ممکن است اینهای را

در اجزای خلاف این را اعلام کنند و بلوغی کنند از اجزای دلیرد چار failure شده

۸) خطاهای fault باید اجزای از منطق و عملرد اصلی component از های باشند tolerance

مدل تلخی ANR کردن <sup>HW</sup> دریم و درنگیت سان



خروجی های ALU ها به ترتیب voting انجام می دهند.

\* اصطلاح <sup>HW</sup> تلخی ANR خطاهای ایجاد شده را mask کرده ایم.

۹) سیستم باید خطاهای سیستم میبیند و خود داشته باشد.

۱۰) لازم است که <sup>ture & easy-to-use</sup> interface داشتم اول باید ساده و سعیی علاوه

باشد. داشتم بعدی نیز است که علی رغم سادگی و سادگی است فرد سوانح به

۱۴۰۳ / ۹

Principles  
Kopetz Design

بر از تحیله practice در failure این اصل بست آنها نداشت.

۱) در مرحله از طراحی باید به نیازمندی های safety critical failure داشته باشیم.

۲) باید حالت های خاص در failure مادرست کرته شوند و فرض های دست در استانداری برای توصیه safety خود داشته باشیم.

۳) دسترسی FCL و راستایی کنیم (Fault containment Region)

در اصل region نباید بزرگ داشته باشیم طوری که fault های هر مرتبه

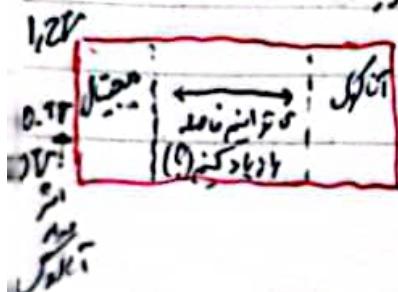
ایزوله نشده باشند. البته که trade off بعنوان هم داریم. مثال: سیم که

هر دو نوع آنالوگ و دیجیتال را در خود داشته باشد، آنلگ جریان زیاد را به زمین

خواهد برد بنابراین صفر پسیل زیاد می شود و حساسیت نویز (باشه ولذتی قیمت دیجیتال)

کاهش سیماید و راحت تر خطها می توانند در میان دیجیتال ایجاد شوند. همچنان خود نویز

نمی تواند در میان دیجیتال هم بعثت fault در میان آنالوگ ایجاد شود.



آنالوگ

دست بایستی کرکند، نایابی سیم لایسنس و محدودیت های انتخابی  
و safety را حفظ کنند.

(۱۱) هرگونه ناهنجاری که در سیم رخ روده لازم است که ضبط شود. خواهد کرد  
سلن اس دهن سیم باشند و خطایین در سطح interface مانند مسند و مُضمن  
exception لازم است که رکود شوند. هنر خطای mask (به این معنی که

handle را می بینند).

Never Give up داشته باشیم. به این معنای برای تمام error  
interface را بازیابی کنیم. سلاخ برخ handle را

از pop-up window می شود.

Highly reliable  
Low-Power  
Real-time

{ safety critical بیمهای نزدیکی

Reliability را در صورتی که در حالت ایجاد اشغال دنباره شد خود را اجرا کنم دارم

رادیو برابر کاصه Real-time بودن، اتراسیمی دهم. یاد نماید AMR دارم

در ازای حصرف انرژی بیشتر توسل component کی نکن، به Reliability پردازم.

## RTOS Kernel ②

کاراصل این kernel مدیریت منابع و اتمام تهدیات است. از طرف دیگر این باید تعلیماتی همیز شود.

\* در تفاوت Standard OS ها را به صور اتفاقی در RTOS می‌دانیم که تهدیات

بر RTOS ایجاد نمی‌شوند و Stand. OS crash کردن ایجاد نمی‌شوند اما RTOS اینها را می‌توان ایجاد کرد. در اصل اینها احتمالاً non-RT خواهد بود.

\* دفعه از زی دستیمها نیسته باید ای. و می‌تواند در RTOS ۲ برابر باشد.

آنچه زی دستیمها می‌باشد حدود استفاده بجزء سیر دنیا می‌باشد.

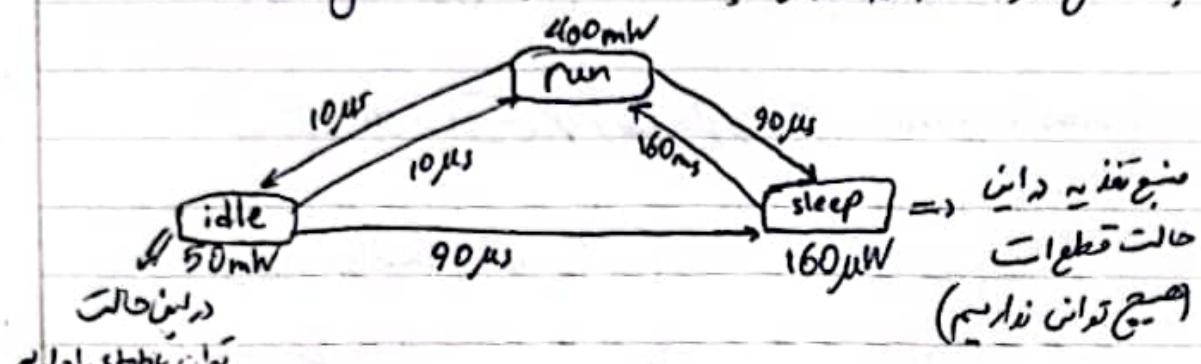
Power, Energy, and Reliability ☑ ★

Idle Dynamic Power Management : این اصل خوب کردن افرازی عالی

این خوب این تلکیل این است که general است و می تواند بسی اجزای

دیجیتال، آنالوگ یا حافظه اعمال شود.

(Strong Arm 1100 در پردازنده وات) DPM ①



\* مبنی استناده از همین حالات لازم است که دیگر بین از آینه

ستم که داشته باشیم و مثلاً اگر تا کله دقیقه آینده تکه برای اجرای این سیم خوب

است که سیم حافظه شود (sleep).

(Dynamic-Voltage Scaling) DVS-Enabled Proc

درین تلکیل supply voltage را تغییر مدهن و متناسب با آن load change می کنیم

هم بروای خواهد بود. طبق  $\frac{C_L \cdot V_{dd}}{(V_{dd} - V_{th})^\alpha}$

که می بینیم delay زیاد شود و به عبور فریم کا صرف یافته است.

از طرف طبق  $P_{sw} = \alpha C_L V_{dd}^2 f$  توان dynamic مدار است که به  $V_{dd}$  داشته است

پولتیکس ۱۰۰mW (2) طرح مُنْدَسَّ (3) طرح مُنْدَسَّ

نمودار داریم .  
 DPM:  $P_{avg} = \frac{1}{2} \times [500 + 100] + \frac{1}{3} [700 + 100] + \frac{3}{10} \times 0$

:  $P_s = IV$  که  $P_s \propto P_s$  نویسید ازین که DVFS حل برای

DVFS:  $P_{avg} = \frac{71.5}{100} [0.71^3 \times 500 + 0.7 \times 100]$

heuristic ①  $+ \frac{28.5}{100} [(0.7)^3 \times 700 + 0.7 \times 100] = ...$

DVFS:  $P_{avg} = \frac{62.5}{100} [(0.8)^3 \times 500 + (0.8) \times 100]$

heuristic ②  $+ \frac{37.5}{100} [(0.53)^3 \times 700 + (0.53) \times 100] = ...$

نمودار داریم . ۲ نتیجه داریم : ۱- Dual-core ۲- یک Core

$T_1: 50\text{ms}$ ,  $T_2: 20\text{ms}$ ,  $D = 60\text{ms}$

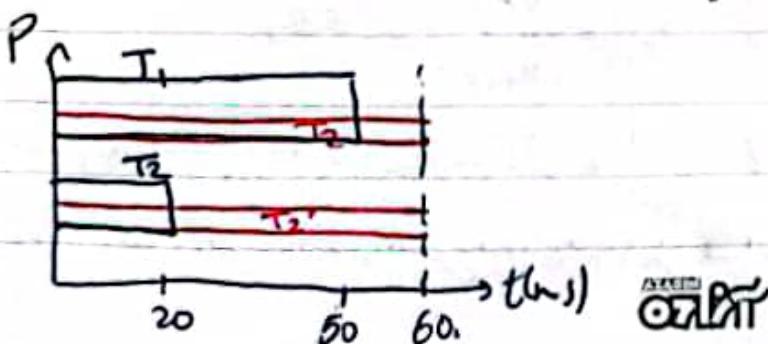
( $P_{avg}(D) = 500\text{ mW}$ ,  $P_{avg}(D) = 700\text{ mW}$ )

نمودار داریم static (بنزه‌ای به مانع core)

DPM:  $P_{avg}(C_1) = \frac{5}{6} \times 600 + \frac{1}{6} \times 0 = 500\text{ mW}$

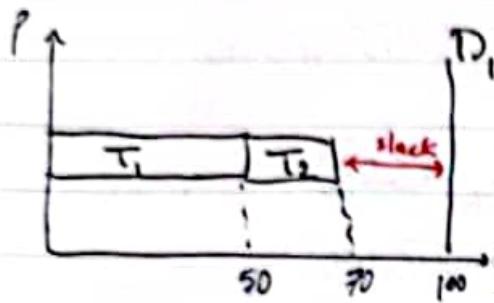
$P_{avg}(C_2) = \frac{2}{6} \times 800 + \frac{4}{6} \times 0 = 267\text{ mW}$

$\Rightarrow P_{avg(\text{tot})} = 767\text{ mW} = P_{avg}(C_1) + P_{avg}(C_2)$



$$T_1 = 50 \text{ ns}, D_1 = 100 \text{ ns}, P_{\text{avg}(Dy)} = 500 \text{ nW} \quad (4) \quad V_{\text{max}} = 3.3 \text{ V}, f_{\text{max}} = 1 \text{ GHz}$$

$$T_2 = 20 \text{ ns}, D_2 = 100 \text{ ns}, P_{\text{avg}(Dy)} = 700 \text{ nW}$$



$$\text{DPM: } P_{\text{avg}} = \frac{1}{2} \times 500 \text{ nW} + \frac{1}{5} \times 700 \text{ nW} + \frac{3}{10} \times 0 \text{ nW} = 390 \text{ nW}$$

$$\text{DVFS: } P_{T_1} = \frac{?}{?}, P_{T_2} = \frac{?}{?}$$

راه اصلی پیدا کردن  $P_{\text{avg}}$  استفاده از معادله متن دینامیکی ایست

برای این ایست  $P_{T_1}$  و  $P_{T_2}$  را با استفاده از همین ساده سیستم  $\text{slack}$  را بخواهیم  $\text{heuristic (I)}$

$$\text{slack}_{T_1} = \frac{50}{70} \times 30 \text{ ns} = 21.5, \text{slack}_{T_2} = \frac{20}{70} \times 30 \text{ ns} = 8.5; \text{ وظایف تخصیص دهنده}$$

$$\Rightarrow P_{T_1} = \frac{50}{50+21.5} = 0.7, P_{T_2} = \frac{20}{20+8.5} = 0.7$$

$$\Rightarrow P_{\text{tot}}^{(\text{avg})} = \frac{71.5}{100} \times (0.7)^3 \times 500 + \frac{28.5}{100} \times (0.7)^3 \times 700 = 191.05 \text{ mW}$$

دوم این ایست که بین متریط نویں مصرف،  $\text{slack}$  محاسبه شود:

$$\text{slack}_{T_1} = \frac{500}{1200} \times 30 = 12.5, \text{slack}_{T_2} = \frac{700}{1200} \times 30 = 17.5$$

$$\Rightarrow P_{T_1} = \frac{50}{50+12.5} = 0.8, P_{T_2} = \frac{20}{20+17.5} = 0.53$$

$$\Rightarrow P_{\text{tot}}^{(\text{avg})} = \frac{62.5}{100} \times (0.8)^3 \times 500 \text{ nW} + \frac{37.5}{100} \times (0.53)^3 \times 700 \text{ nW} = 199.5 \text{ mW}$$

اگر بین همان دو ایست را بر اساس درستگیری تقریبی و ایست  $\text{avg}$  ترجیح بگیریم..

۵ نتیجه

دست و زدن بهی

۱۴۰۳، ۲، ۲۳

ابن افوارتین  $\Delta$  PC، MOC، ES، ES کیم:

مسئلاً سیم هر PC عن صند، دستیم هر ES مکاری ها نهان متن.

مقدار  $\Delta$  MOC، ES از نوع Discrete Event است.

در ES ها هن بین بازی چنی درسته.

PC های اصل که خواهی بین باشد.

در ES ها معملاً شرکت های سخن دسته هستند.

[چند مورد دیگر در اسلاید بعد ...]

حالی بینیم که توصیف ملے ماجه است؟ داده شده: جو عای از application هاو

ساده نتریکه این نرم اثراها به نحوی استفاده می شوند. تعدادی مکاری کندیم داریم.

خواهیم بیا کیم: که mapping از app های بردگانها

۱۵۰۳، ۲، ۱۱

$$T_1: 50\text{ms} \quad (\text{زمان})$$

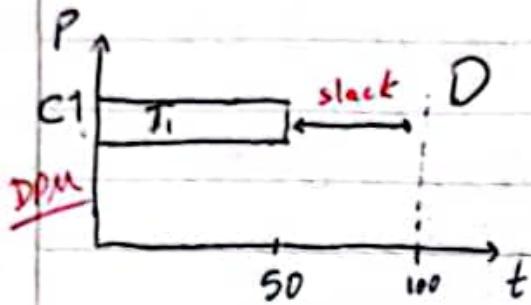
$$\begin{matrix} G_{lb} \\ f_{max} \end{matrix}$$

$$P_{avg} = 500\text{mW}$$

(J2)

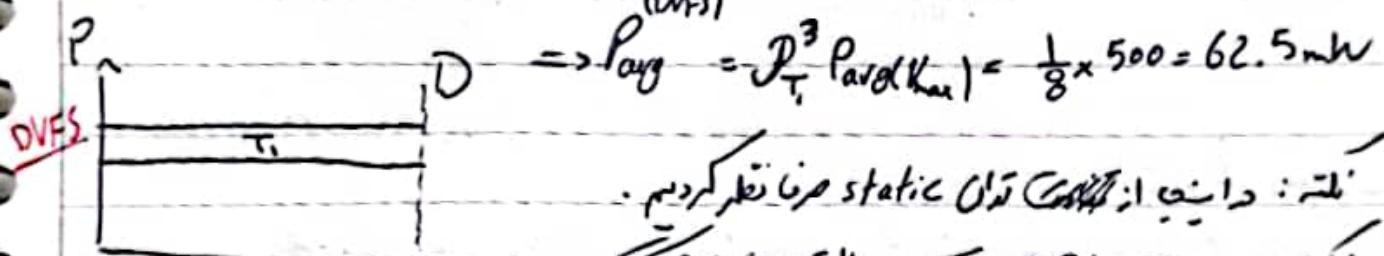
$$D_1: 100\text{ms} \quad (\text{مطالعه زمان})$$

$$V_{max} = 3.3V, f = 1\text{GHz}$$



$$\text{DPM}, P_{avg} = \frac{1}{2} \times 500\text{mW} = 250\text{mW} \quad \left( \frac{f_1 \times 500 + f_2 \times 0}{2} \right)$$

$$\text{DVFS: } \rho_{T_1} = \frac{t_{exe}}{t_{exe} + t_{slack}} = \frac{50}{50+50} = \frac{1}{2}$$



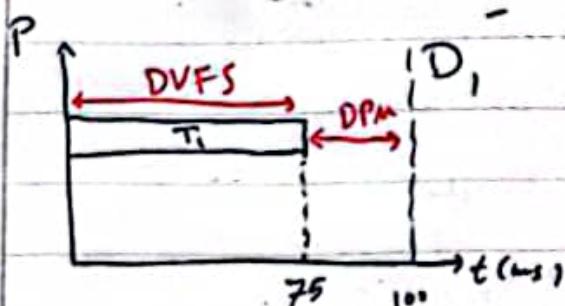
نتیجه: داینامیک از  $C_{min}$  تا  $C_{max}$  صرف نظر کردیم.

نتیجه: در حالت DPM ببینیم که حالت بعد همچنان

در میان در 50ms دستگاه را کمالاً خودش می‌کنیم و در حالت sleep تراویح داریم از صرفه است.

که در حالت sleep میان تغییر پردازش static می‌باشد و در حالت DPM نیست.

۱) توانیم DVFS را بازی  $\frac{3}{4}$  بازه مطالعه اندازد و داشتیم، به جای  $T_1$ :



$$P = \frac{50}{50+25} = \frac{2}{3} \Rightarrow$$

$$P_{avg}^{(new)} = \frac{3}{4} \times \left( \rho_{T_1}^3 \times P_{avg}(new) \right) + \frac{1}{4} \times 0 \approx 111.11$$

تمام

نکات و نمانه ها

۱۴۰۳، ۲، ۲۳

ابناء آنوات میں LOPC، LMOE، ES، برسوں کیمی:

عملاً نیمی PC میں ہے، دیگر نیمی ES میانگین میں ہے۔

عملاً درج LOPC، LMOE، ES از نوع Discrete Event است.

در ES میں بینی سازی چنبر) درج ہے۔

ایدیا PC میں اصل خواص بینی ہے۔

در ES میں ترتیب میں شخص و فنہ کو مسترد کر، PC میں

[جنہی مدد دیا گیم در اسلامیہ بعد ...] دیگر وجہ دناری۔

حالی بینیم کہ تو صنعت میں ماجھ اس؟ دارہ تھے: جمعی ای از application میں

کہاں نہ رکھ کے ایں لرم اٹر رہا بھی نہیں استفادہ من ہے۔ تعدادی میانگین کندیم داریم،

خواص پیا کیم: بلکہ mapping از app میں برداشت میں

$$P_{T_1} = \frac{50}{60+40} = \frac{5}{10}, \quad P_{T_2} = \frac{20}{20+40} = \frac{1}{3}$$

DVFS:  $P_{avg}(C_1) = \left(\frac{5}{6}\right)^3 \times 500 + \frac{5}{6} \times 100 = \frac{372}{372} \text{ mW}$

$$P_{avg}(C_2) = \left(\frac{1}{3}\right)^3 \times 700 + \frac{1}{3} \times 100 = \frac{59.22}{59.22} \text{ mW}$$

$$P_{avg(sys)} = \frac{372}{372} + 59.22 = \frac{431.22}{431.22} \text{ mW}$$

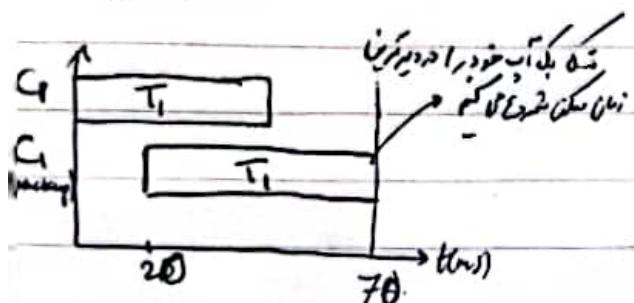
ناتیجہ: سامت ٹائی کم فردا ر  $P_t$  پر تکمیل کیتی جائے، دھنہ دھنہ اور اسیں بنتی

ایسے میرے بزرگ تر باشد، سطح دلیل؟ فرمائیں اسے دو شرط سُر خواهد بود.

(Reliability backup time) Stand up Starting 't' (5 نئی)  
 $T_1 = 50\text{ms}$ ,  $D_{T_1} = 70\text{ms}$

$$P_D = 500\text{mW} \quad P_S = 100\text{mW}$$

$$S-S =$$



در الگوریتم های offline مادام تپا تمام دلاین ها و زمان اجرای تکه های شخصی شوند.

را داریم و از میان اجرای کس زمان های شروع را شخصی کنیم و در زمان

اجرا می خواهیم invoke کرد هر تکه در زمان مشخص لازم است که انجام شود.

این نوع زمان بندی به حافظه predictive دارد و نیازی نمی داشته باشد.

تکه های از جنس periodic است. زمان بین طبقه online در زمان routine

هم زمان شروع تکه های شخصی کرده هم invoke کرده آنها را انتظامی کرده.

\* برخی جالش های که ممکن است در زمان بندی سطوح شوند ممکن است

که فرآیند نوع بردازندگان مختلف ساختگاری باشند. در این صورت ممکن است

زمان اجراهای یک تکه هایی بین بردازندگان مختلف نباشند.

☒ زمان بندی پریوس: مجموعه ای از تکه های زمان بندی برای این تعداد تکه های سطح،

که صورت کرده و بعد داشته باشند (زمان بندی از اجرای تکه های آن سطح) را ارضا کنند.

get Function های مسائل ممکن است حوار دشمنی های داشته باشند.

① Maximum lateness =  $\max_{\text{all tasks}} (\text{Completion time} - \text{Deadline})$

اگر این مساعی متداری مسابت داشته باشد بعضی دست کم میان از تکه های دلاین

فود را miss کرده اند.

صفات مرتبه : ۱- نُسْتَ UCL های مطابق سیم های

۲- متعلقه به زمانی ، این زمان های شروع تکه ها هم زمان باشند.

۳- partition کردن نرم افزارها<sup>۴</sup> در حال اجرا سخت افزار موجدد . [ تعدادی مورد دیگر اسلامی ]

### سطوح تکف نهاده

۱- در تکمیل توزیع ساده سخت افزار نوع آنها ( Allocation )

۲- در تکمیل توزیع افزایشی تکه ها ( Assignment )

۳- مشخص کردن زمان اجرای هر تکه در بازه ها

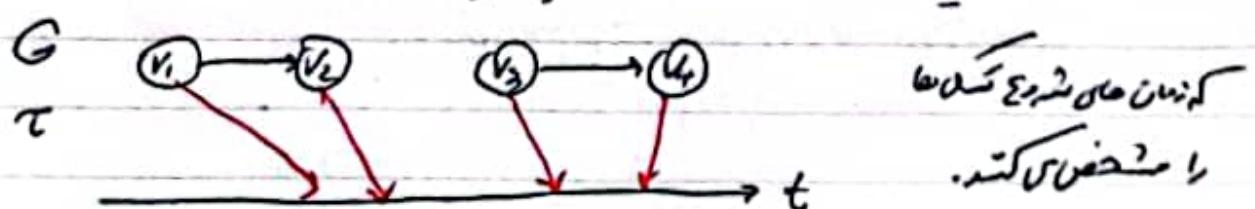
[ تعدادی مورد دیگر در اسلامی : ) ]

امثله الگوریتم های زمان بندی وابسته به dependent بودن یا غیرdependent مسائل

است و تغییر کند.

$G(V,E)$  : یک گراف از تکه های خود ( دایتل ها ) داریم <sup>۵</sup> Schedule

ساخته ایم نهاده جای G را پیدا کنیم به صورت زیر :



\* که دلایل Hard است اگر این فرضیه آن مسجد نویجه ها ( خانی ، گالری )

بشد و درست بگیر آن soft دلایلی ، دلایلی است که ابعاد گللات ابعاد نشده بزرگ نیست .

\* تک‌ها چند من دسته دارند.

میں اسے ملن اتے aperiodic & periodic تھے۔ تھیں کہ پہلے دنیا میں اجرا شوندہ، periodیت متنہ، نرودتی اور aperiodic ما تکھیں sporadic ہتھ کر نامٹیں مرکام اور اجراتی آؤں ہا دت کیم چھپیں ہیں۔ بھی ملن اسے گزار شوندہ ولی ان کلراویں ہا تے و بندی۔

\* زمان بندی ها هم حیندسته دارند:

- پیشنهاد مبنی بر pre-emptive و non-pre-emptive است. در صورت اول تکه های نزدیکی به اتهام رسیدن از بردازه خارج نمی شودند. مبنی بر دو دلایل این است که response از ماهیات تکه های مختلف خیلی ریز است. در صورت دوم هم مبنی است که مبنی است تکه های بزرگ پایین ~~stare~~ <sup>starve</sup> می شوند.
- پیشنهاد مبنی بر dynamic/online باشد یا static/offline است. البته مبنی است تباشیم مبنی بر اطلاعات تغیری کنیم که static بدون از <sup>online</sup> ~~other~~ <sup>dynamic</sup> دون مستقر باشند. (البته

۱۷.۵.۲۰۲۴

قریب نرمال : Aperiodic Sch.

۳) T<sub>i</sub> : جزء نتیجه ای است. باز این ترتیب به زمان اجرای نباید دلایل اجرا کرد

مدت زمان شروع کار دیگری زمان معلم پس از آن بیان رسیدن ترتیب است (d<sub>i</sub>) و به زمان

برابر با slack<sub>i</sub> است.

۴) آن بردازنده زمان های انتیا می باشند.

تقریب که دلاین آن نزدیکتر است را اجرای کنیم. Earliest Due Date (EDD)

(پیچیدگی: O(n log(n)) که مرتب شدن sort نیاز است).



: Earliest arrival " "

۵) درین حالت preemption داریم و تقریب جدیدی که سرمه اگر

زمان Deadline نزدیکتر باشد، جگالنرن ترتیب در حال اجرا می شود.

در صورت که از صفت های sorted استفاده نکنیم، پیچیدگی O(n log(n)) خواهد بود.

درین سمت پیچیدگی n<sup>2</sup> خواهد بود.

۶) درین متعدد و مختلفیات که زمان slack کمتری دارد به Least Slack Time First (LSF)

عنوان ترتیب اجرا شونده انتخاب می شود. نظریه بُت این الgoritم همان است

که هر چه slack<sub>i</sub> بیشتر از slack<sub>j</sub> باشد، دلاین آن بیشتر است و ضرور است

از آنها

شیوه دارد که اول اجرا شود. latency در هر مرور جدید تسلیم به صورت نبی  
relative deadline

$\text{latency} = (\text{Absolute Deadline} - \text{Current Time})$  بازی محاسب شود و یعنی:  
-  $(\text{Duration} - \text{Previously Executed Time})$

❶ خوبی این الگوریتم این است که هر قوان نزد متوجه Deadline Miss ها شویم مرا  
که اکنون محاسب  $\text{latency}$  شود، متوجه شویم که آن تسلیم به دلاین نخواهد رسید.

❷ نکته جالب این الگوریتم این است که علاوه بر آن priority تسلیم هادا نیز در حال  
تغییر است و dynamic priority شناخته شده است.

❸ نکته سفر این الگوریتم context switch را زیاد کردن است.

❹ لیم: باید الگوریتم زمان بندی در صورتی که deadline نداشت باشد، باید

idle time optimal سلیمانی باشد.

\* missed deadline این است که missed deadline های خود را کمینه کنیم.

❺ نکته از lemon lemma مطرح شده این است که وقت دسترسی periodic داریم

نهان که باید در اثر اجرا نهان تسلیم اول و دوم، تسلیم اجازه:

نکته نداشت باشد، دلاینی miss شود در نتایج بعدی به جای  
اجرا کردن آن، مقداری زمان idle داریم تا پردازندۀ سعده نبیند که هستم  
و سین  $T_2$  آن را اجرا کنند. (با این فرض که سین از اجرای  $T_2$  نهان خواهد بود)

رسیدن آن به دلایلی وحشود دارد.

❷ زمان بندی با واسطه مین تسلیق (Precedence Constraints) که  
دیگر همی داشت و ابتدا هسته

تسلیق تاکتیکی تاکتیکی حدود که حسنه را بازگشایانده دارد

درین یعنی صفتی برای اولویت که بعد از تسلیق همیکم Deadline خود را ندارد دارند،

زندتر اجرای شونه (در حالی که به جای صفت از stack استفاده کنیم، درین این stack خود

تسلیق هایی دلایلی دیرتر را زندتر اسازیم و نتایجی LDF از این طریق منسوس نیابد).

دانسته که معنود را پر کردیم و تسلیق باشی خانه، میتوانیم از منع تسلیق های اسخانی و

آن ها را اجرا کنیم.

❸ Modified EDF: در این روش سه مرحله ترتیب مین تسلیق ها را ازین بیریم و

پس پر کو EDF تسلیق ها را اجرا کنیم.

❹ ASAP: از ریشه درخت شروع کنیم، start time را برابر با زمان زمان

start time + duration بازیم. finishing time خر این تسلیق هم را شروع شد

حل کردن های متوجه دادیم را بست آمدیم و تقاضای زمان بندی را انجام دهیم. اگر

فرض کنیم محدودیت منابع نداریم، به تعداد بیشترین تسلیق هم تسلیق که هزمان میگیرد است

ZDT

## Diff Sigaling بہمنی : CAN کانٹرول

\* هنری کر  $\rightarrow$  ES های از این هنری می‌شوند. اول latency و در دو orient-based

هستم، دفعه داده های سایر کم است و در علیه نظریت مسجدود در Ethnret هم نیز نباید

میتوان سرگیری با استفاده از CAN باقیه خود را آغاز کنیم.

\* در این پروتکل استفاده از Arbitration برای CSMA/CA نیست.

برخواهی Ethernet که آدرس ندارد، message ایجاد کرد و آدرس دارد. بنابراین

در عمل هم می توانیم یک node جدید را اضافه کنیم و یک node را حذف کنیم (dynamic, scalable)

\* از مکانیزم تخلیقی اسکیم ARC است که timeout را بازگشت (Automatic retrig command)

جس کی دنگوں کے سنتھریں ہیں، اس نوائیں خداوار استاداں، handle کیم۔

wired-and ; استفاده از : collision هندل کردن

آہادیم، مہاں دنخوات دادن

نکته ۲۰: خود را ارسال نمایند، پس از که آنها دریافت شده باشند بروز

Bus تراکر گرتے اے، اداہے ہی دھن۔ داخل ٹکریں Bus رانغراہد صفر را،

Ban در این دارویی نمی‌نماید. اگرچه ~~نکره~~ بتواند را پذیرش نماید، معملاً صفر است.

**دستورات** هر چیز دارای یک اندیشه است و صفات همان اینست که از آنها برای

اچرا آئونه، پردازنه ده مقتدری لیستم.

مُبَعْد مُنْهَا (Late) ، ALAP ، ASAP ، از بَرَكَه مُسْتَعِد مُكْنِم Finishing time ، از کُنْهَا

صفر ده تظاهر کیسے ہے؟ تراویح دھم "start-time = f\_startduration". بڑا تسلیمان لائیٹ بالا رہے

زمان شروع و زمان پایانی هر کار را start time و finishing time می‌نامند.

در لایه بالاتر، به بهترین سلسله میان که به آن داشتنی دارند نظر نمی‌گیرند). این از نزدیک

نذریں کنم۔ تاہب باللہ

150, 2, 20

number of tasks

زمنیہ اسکے لئے 

Average utilization:  $\mu = \frac{1}{m} \sum_{i=1}^n \frac{C_i}{P_i}$  executing period ;  $\frac{C_i}{P_i} < 1 \Rightarrow \mu \leq m$  number of processors

دالان تکنیک (ii) :  $d_i - p_i$  (iii) : مفرزها (iv) : Rate Monotic Scheduling

(iii)  $\text{NaCl}$  هم تردی periodic داشته و مساحتی هست.

(v) زینتی contact switch تاںل حیثیت دوئیاں ہے۔

$$\mu = \sum_{i=1}^n \frac{c_i}{p_i} \leq n(2^{\frac{1}{n}} - 1) \rightarrow \text{مبنی بر این نتیجه است} \quad \therefore \quad \text{معنی}.$$

نہ بینی اکر مل جید سندھی سادھ کران بالائی ذکر مٹو صدی کند، آن مجدعہ

تمامی کارهای اکسپریم RM مابین زمان بینی است که در عین‌این صورت، همچو.

\* در عرصه فضایی محدود و محدود از مدت کوتاه مراجع محنت هایی بسیاری بیشتر.

- \* در این روش که static priority داریم که وابسته به مالیات هاست.
- \* با تراصیر تعداد تکه های کراس مالیات ذکر شده که روش خوب نباید بیشترین بیشترین
- \* کسر تراصیر روز هر روز بردازند قرار دسیم هم همنه ری باشد.
- \* هر هنوزن  $\frac{d}{d+1}$  تکه بیشتر باشد، این باید بالاتر خواهد بود. لازم است ذکر است که بدبختی ادلویی دهن به ترتیب  $n$  preemption هم استفاده شود کنیم.

$$T_1: c_1 = 3, p_1 = 5 \quad T_2: c_2 = 3, p_2 = 8 \quad (\text{JL})$$

$$\therefore \mu = \frac{3}{5} + \frac{3}{8} \approx 0.975 , \quad \text{upperbound}(RM) = 2(2^{\frac{1}{2}} - 1) \approx 0.828$$

$$\mu > (2^{\frac{1}{n}} - 1) \quad \text{نیا سبب زمان سند بزر نہیں!}$$

\* اگر هنر بردازنه داشته باشیم باید ابتداءاً هارا به نماید m بردازنه خود  
کنیم و بآنها هر بردازنه صیغه که کان بالا (partition utilization) کنیم

\* به طور سُفترض نازنینم تعدادی بسیار کم را نزد خود کنید که صفر است.

\* اکثر بیم مزرس اند (اممیت بیم) جو شدید بیم capacity را توانیم زمان بینی خود را

انجمن دهشم.

با تکمیل تکه ها، partition کردن آنها من توانیم تعداد هسته های فعلی را در لحظه

را اندیسی کنیم و با همین می توانیم دمای راهبردی لحظه کسر کنیم و با این توانی TAT بخوبی

شود) فرض کنیم که تکه های زیر را داریم:

$$T_1: WC_1 = 20, P_{P_{max1}} = 13$$

$$T_2: WC_2 = 15, P_{P_{max2}} = 10$$

$$T_3: WC_3 = 10, P_{P_{max3}} = 14$$

→ no dependency  
same deadline →  $d_1 = d_2 = d_3$ ,  
same release time

بازه شکاف (partitioning slot) برابر 15 است (و بجز این حد این اسے که می خواهد)

تکه های کوچکی هستند که در از لحاظ کسری روح از لحاظ سایر نیازهای راه را می توانند

T <sub>11</sub>	T <sub>12</sub>	T <sub>13</sub>	T <sub>14</sub>
T <sub>21</sub>	T <sub>22</sub>	T <sub>23</sub>	X
T <sub>31</sub>	T <sub>32</sub>	X	X

به ترتیب های 5 گانه ای هستند:

۱- در زمان بینی خود نهاده کنیم که کامپیوٹر بالاترین حد را داشته باشد

تعداد حضور را در دست آن بیشترین اولویت را از دهم و ایش بروزگراندی اصل

من کنیم. هر تکه به باقی میان بروزگراندی spare می بینیم تکه را استخراج کنیم و این اسے

بایسی ترین اولویت را پیدا کنیم و سطحی بیش از هفتمین هزار میلی اجرای

جذبی تکه پرتوان در بروزگراندی تکلف را کاهش دهد.

در صورتی که در تکه (یا جذبی) هزار (هزار) بیشتر از این داشته باشیم

آن

\* bit ها در آن اولویت را به صورت serial ارسال می‌کنند.

مونهای بایم هادارند هر یک ارسال یک عدد and می‌شوند و هر لازم هر این سه حالت اعذار

دهن متوجه شوند و صرف کسانی که صفر را شنید در ادامه راه است به این Bas می‌باشد. دنیا پی

کس که کوهیک ترین اولدی را داشته باشد، صاحب خط سیمود.

\* محدوده در bit های ارسال یک سری ACK field هم داریم که به صورت سیف من برابر ۱ است.

و صورت که لازم باشد NACK بگیریم (?:)

\* دنوع بایم آزادیم، <sup>(Frame)</sup> Remote Frame بخوبی data، remote، ریسخ هادر

قابل Data Frame ارسال می‌شوند.

\* باسته از اسم identifier عدد ارسال شده بعنوان data تفسیر می‌شوند.

\* لذت های هفظی که می خواهیم ۶ صربت هم ببریم بنهوی آن را تغییر دهیم که دیر <sup>(Bit Stuffing)</sup>

این خود را نیست آهون ۶ صربت هم نهی دهنده یک سلسله Critical Error است.

\* اگر ارسال کنند بایم ۶ بیت هم ارسال کنند، دارد اعلام کنند که خود را حفظ انجام داده اند

\* collision میان است رخدادی این سلسله با استفاده از اولویت های کجا باشد بایم ها

مشخص می‌شود، هندلر سعد Non destructive Collision

ایک الگوریتم دیگر اے کہ ہر اس زمانہ تک مارک اتنا دا منداد ہے۔ EDF

بیان آن کہ مارک اتنا دا حیبہ ہے کہ hyper-period (hyper-period). اگر تہائیم تک مارک اتنا نہیں کیا تو مارک اتنا نہیں کیا۔ دراسن دوسری اولادت dynamic ہے تکمیل کرنے اجرا تفسیری کئے۔

در ہان (نیل) ذکر شدہ در صفحہ قبل،  $T_i$  کے دوسری ماں خود میں ملک تواند  $T_i$

در حال اجرا را preempt کر کر deadlie کے  $T_i$  کا دوسری ماں درحال اجرا تک ملک تراست۔

بی طور معمول ہری OS ہاں fixed priority ہے اسے RMS ایں ملکیتی EDF

راداں۔

EDF از کھل کوان کا باس بردازندہ اتنا دا منداد RMS محدود (1-2) ہے۔

نہیں بھس  $\text{NP-Hard}$  ہے اسے heuristic dependent، periodic دیا جاتا ہے۔

را استفادہ کئیں۔

### Power-aware Scheduling

→ ملن اے کہ کیا کوان طراحی Thermal Design Power (TDP) کا لحاظ کر دا شایستہ Motivation

با کنڑا ایک نئی سسٹم میں اے دھارنا باید اسی میں ہوں

TDP میزبانی بود و یک slot تعلیم خود را (راسته بروی primary & spare) داشت.

عقب این هلو) سیستمی داشتم.

\* برای هر slot از تعلیم خود یک  $P_{max}$  جواہر حساب می کنیم و به صورت

دقیق تر در هر slot بتوانیم worst case را تعیین کنیم.

به عنوان مثال  $P_{max14}, P_{max13}, P_{max12}, P_{max11}, P_{max10}, T_1, T_2$  را برای  $P_{max}$  جواهه کنیم.

حساب کنیم و داریم:  $\{P_{max}\} = \max\{P_{max14}, P_{max13}, P_{max12}, P_{max11}, P_{max10}\}$

۱۵۳، ۲، ۱

★ در ES ماتطابقه نهاده  $\rightarrow$  خالی آن از Distributed ES استعدادی کنیم. Communication

Bus - ۲ P2P - ۱

\* P2P ب لحاظ هزینه ای کمتر است دلیل توافق در هر زیرسیستم از روی این تهدید باشد.

جزء سیستم  
\* عملاً در P2P میان هر دو Link واحد دارد و در Bus پر میان هر دو مشترک می باشد.

خبرها مجدد دارند.

②

①

در درجه اول Decentralized, Centralized(Arbiter), Bus-based وجد دارد:

(Single Point of Failure) ① در این حالت Arbiter یک تعمیم کرده Bus و لخطه فعلی دست چیزی باشد.

② در این حالت یک component با لخطه ای Bus sense یا Ethernet می کنند. اگر

CS CamScanner

مشمول باشند که عدد رزیم تولیدی می‌کنند و درخواست خود را بباخته‌ای فریم اعماقند. این دو مسئ

مذهبی شود که عدم تعلق داشت باشیم و نتوانیم با این بین (west-east) را ساخت کنیم.

کل درست کردن سایی از سطح جوش های Poling است. در این روش (با ارجویت یا مبنای) Based

مدت زمان‌هایی را به مرکز از اجزاء تخصصی دهم. درین «کل سُلْطانیات» ک

از بیام خار Emergency رخوبی نه تنایم سایر کنیم هر که خطا در زمان

که سعفان یا شه هماز می‌نماید آن نمی‌توانیم صرف کتنه از خوارانفس داشیم.

: Ethernet، پرتوانی

## Collision detection

radio wave multiple access/collision

(ii) carrier sense multiple access/collision detection (CSMA/CD) - I detection

داده نهن سورد.  
CSMA/CA -۲ : نتیجہ درستگی میں درستگی اسٹ کاملاً اجازہ رخ دارن avoidance

