

سخنرانی: 10 همزمان MoC های واکنشی و زمان دار

سید حسین عطارزاده نیاکی

چند اسلاید از ادوارد لی و تام هنزینگر

سیستم های زمان واقعی جاسازی شده

1

بررسی کنید

• CoM های جریان داده

• قابلیت تجزیه و تحلیل مدل های جریان داده

- برنامه ریزی

- اندازه بافر

• جریان داده همزمان • انواع دیگر CoM های Dataflow

- HSDF

- BDF

• مقایسه CoM های جریان داده

سیستم های زمان واقعی جاسازی شده

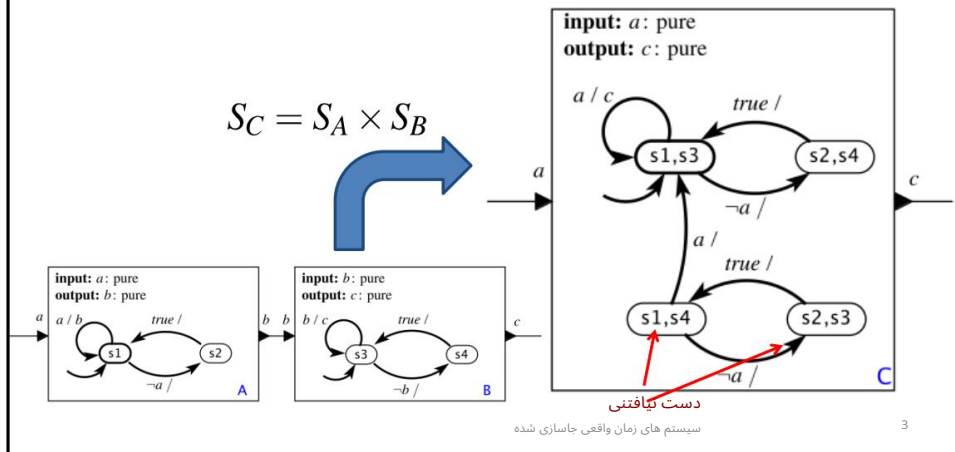
2

خلاصه:

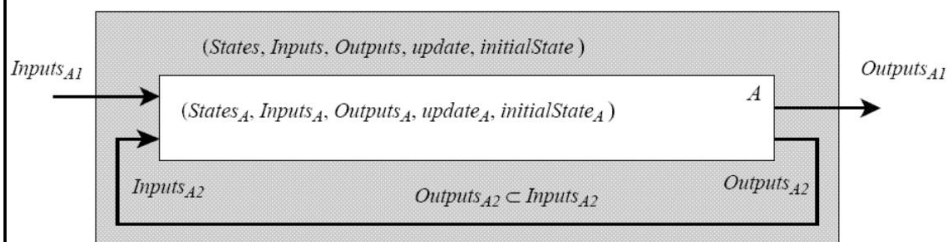
مدل راکتیو سنکرون محاسبات

واکنش ها همزمان و آنی هستند

در این مدل، نباید فکر کنید که ماشین A در زمان فیزیکی قبل از ماشین B واکنش می دهد. اگر این کار را می کرد، حالت های دست نیافتنی غیرقابل دسترس نبودند.



ترکیب بازخورد



معلوم شد همه چیز را می توان به عنوان ترکیب بازخورد مشاهده کرد...

ترکیب همزمان FSM ها: راه حل های مختلف

مراحل خرد: یک واکنش دنباله ای از مراحل خرد ابتدایی است

• اتخاذ شده توسط VHDL, Verilog, StateCharts

Acyclic: حلقه های بازخورد با تاخیر صفر ممنوع هستند

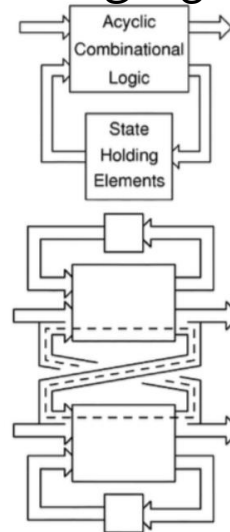
• طراحی سخت افزار همزمان، درختش

نقطه ثابت منحصر به فرد: واکنش ها راه حل های معادلات نقطه ثابت هستند

لی آپمک • سخت تر، Esterel

رابطه یا محدودیت: واکنش ها به عنوان محدودیت

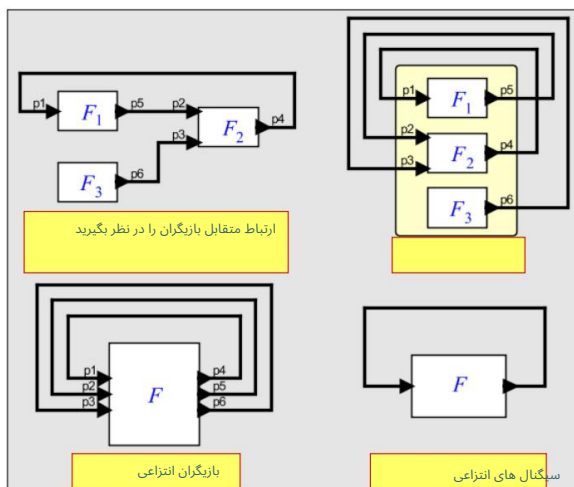
• راه حل های متعدد به عنوان رفتار غیر قطعی، سیگنال



سیستم های زمان واقعی جاسازی شده

5

مشاهده: هر ترکیبی الف است ترکیب بازخورد



سازماندهی مجدد

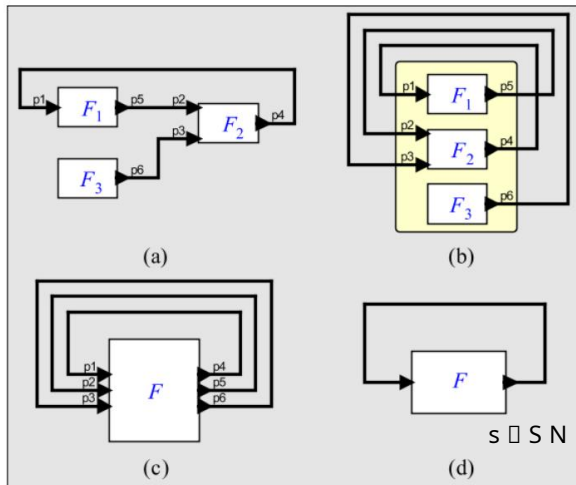
$s \sqsubseteq S N$

رفتار سیستم به طور کلی یک "نقطه ثابت" است.

سیستم های زمان واقعی جاسازی شده

6

معناشناسی نقطه ثابت



ما به دنبال یک $s \subseteq S \times N$ هستیم که

$F(s) = s$ را برآورده کند.

چنین s یک نقطه ثابت نامیده می شود.

ما دوست داریم نقطه ثابت وجود داشته باشد

و منحصر به فرد باشد و ما می خواهیم یک

سازنده

روش برای پیدا کردن آن

رفتار نظام است .

سیستم های زمان واقعی جاسازی شده

7

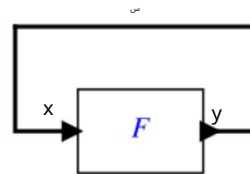
بازخورد خوب شکل گرفته

At the n -th reaction, we seek $s(n) \in V_y \cup \{absent\}$ such that

$$s(n) = (f(n))(s(n))$$

There are two potential problems:

1. It does not exist.
2. It is not unique.



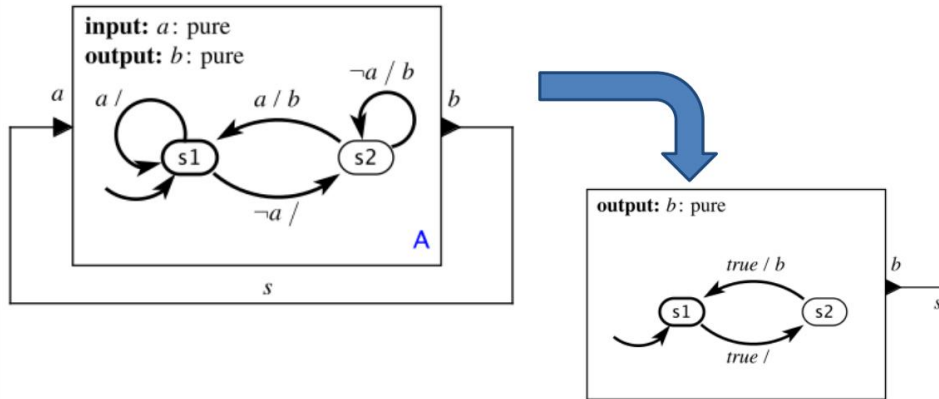
In either case, we call the system **ill formed**. Otherwise, it is **well formed**.

Note that if a state is not reachable, then it is irrelevant to determining whether the machine is well formed.

سیستم های زمان واقعی جاسازی شده

8

نمونه بازخورد خوب شکل گرفته

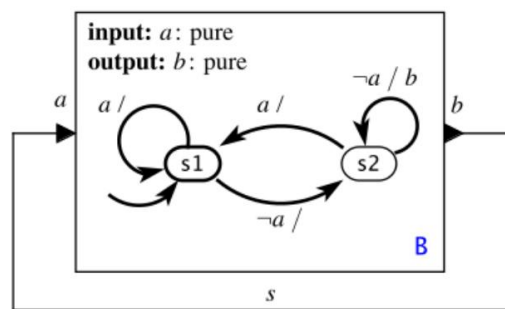


In state **s1**, we get the unique $s(n) = \text{absent}$.
 In state **s2**, we get the unique $s(n) = \text{present}$.
 Therefore, s alternates between *absent* and *present*.

سیستم های زمان واقعی جاسازی شده

9

مثال نادرست 1 (وجود)

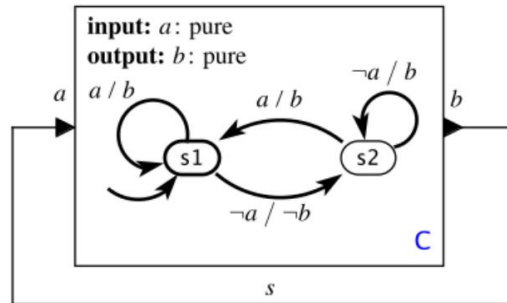


In state **s1**, we get the unique $s(n) = \text{absent}$.
 In state **s2**, there is no fixed point.
 Since state **s2** is reachable, this composition is ill formed.

سیستم های زمان واقعی جاسازی شده

10

مثال نادرست 2 (یگانه بودن)



In $s1$, both $s(n) = \text{absent}$ and $s(n) = \text{present}$ are fixed points.

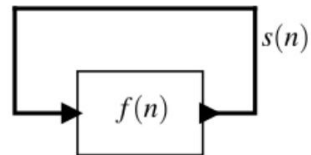
In state $s2$, we get the unique $s(n) = \text{present}$.

Since state $s1$ is reachable, this composition is ill formed.

سیستم های زمان واقعی جاسازی شده

11

معناشناسی سازنده: سیگنال واحد



1. Start with $s(n)$ unknown.
2. Determine as much as you can about $(f(n))(s(n))$.
3. If $s(n)$ becomes known (whether it is present, and if it is not pure, what its value is), then we have a unique fixed point.

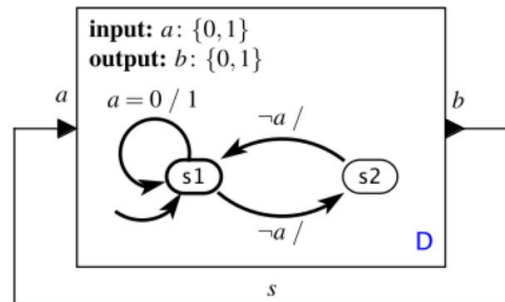
A state machine for which this procedure works is said to be **constructive**.

سیستم های زمان واقعی جاسازی شده

12

غیر سازنده به خوبی شکل گرفته است

ماشین حالت



In state $s1$, if the input is unknown, we cannot immediately tell what the output will be. We have to try all the possible values for the input to determine that in fact $s(n) = \text{absent}$ for all n .

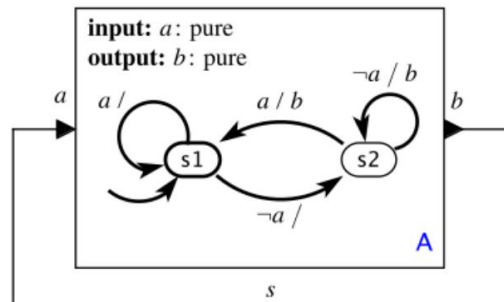
For non-constructive machines, we are forced to do **exhaustive search**. This is only possible if the data types are finite, and is only practical if the data types are small.

سیستم های زمان واقعی جاسازی شده

13

باید / می

تجزیه و تحلیل



For the above constructive machine, in state $s1$, we can immediately determine that the machine *may not* produce an output. Therefore, we can immediately conclude that the output is *absent*, even though the input is unknown.

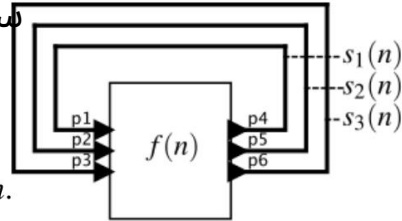
In state $s2$, we can immediately determine that the machine *must* produce an output, so we can immediately conclude that the output is *present*.

سیستم های زمان واقعی جاسازی شده

14

معناشناسی سازنده:

سیگنال های چندگانه



1. Start with $s_1(n), \dots, s_N(n)$ *unknown*.
2. Determine as much as you can about $(f(n))(s_1(n), \dots, s_N(n))$.
3. Using new information about $s_1(n), \dots, s_N(n)$, repeat step (2) until no information is obtained.
4. If $s_1(n), \dots, s_N(n)$ all become known, then we have a unique fixed point and a constructive machine.

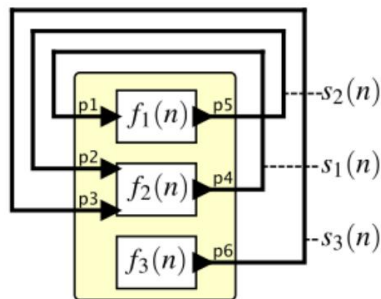
A state machine for which this procedure works is said to be **constructive**.

سیستم های زمان واقعی جاسازی شده

15

معناشناسی سازنده:

چند بازیگر



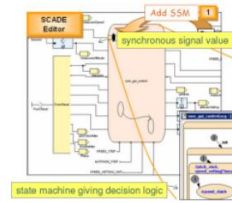
• رویه یکسان است.

سیستم های زمان واقعی جاسازی شده

16

کاربرد

• ابزارهای مبتنی بر SR MoC در حوزه سیستم های تعبیه شده
از نظر ایمنی بسیار موفق بوده است.



Qualified Code Generator. از جمله SCADE Suite.
SCADE KGC توسط AIRBUS و بسیاری از تامین کنندگان
اصلی آن برای توسعه بیشتر نرم افزارهای حیاتی A380 و
A400M و برای سیستم فرماندهی ثانویه پرواز A340-500/600
استفاده می شود. هواپیما در حال استفاده عملیاتی از اوت
2002.



Lustre/SCADE from <http://www.esterel-technologies.com>

• فرانسوا پیلارسکی، چارچوب مهندسی سیستم - مهندسی مدیر ارشد،
سیستم ها و تست های یکپارچه سازی. ایرباس فرانسه



سیستم های زمان واقعی جاسازی شده

17

مدل های زمان بندی شده محاسباتی

زمان-
باعث شد
مدل ها

گسسته-رویداد
(DE) مدل ها

پیوسته-
مدل های زمان

سیستم های زمان واقعی جاسازی شده

18

چرا به مفهوم زمان نیاز داریم؟



سیستم های زمان واقعی جاسازی شده

19

مدل های فعال در زمان: معماری زمان تریگر

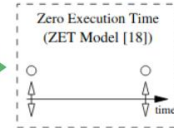
- به طور دوره ای محاسبات توزیع شده را بر اساس یک ساعت توزیع شده فعال کنید
- استفاده عملی در طراحی سیستم های اویونیک حیاتی ایمنی و خودرو
- برخلاف SR MoC، محاسبات زمان بر است
- زمان اجرای منطقی

سیستم های زمان واقعی جاسازی شده

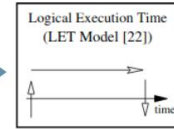
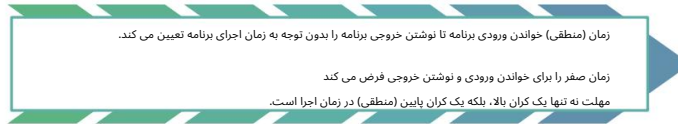
20

انتزاعات برنامه نویسی بلادرنگ

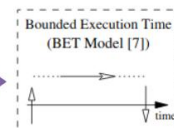
زمان اجرای صفر (ZET)



زمان اجرای منطقی (LET)



زمان اجرای محدود (BET)



کیرش، کریستف ام. و آنا سوکولووا. "پارادایم زمان اجرای منطقی." In Advances in Real-Time Systems. 103-120. اسپرینگر، برلین، هایدلبرگ. 2012.

سیستم های زمان واقعی جاسازی شده

مدل های تحریک شده با زمان

• ورودی های محاسبات با تیک های ساعت جهانی ارائه می شوند

• خروجی ها برای محاسبات دیگر تا تیک بعدی ساعت جهانی قابل مشاهده نیستند

-عدم تعامل بین محاسبات

-مشکلات همزمانی مانند شرایط مسابقه وجود ندارد

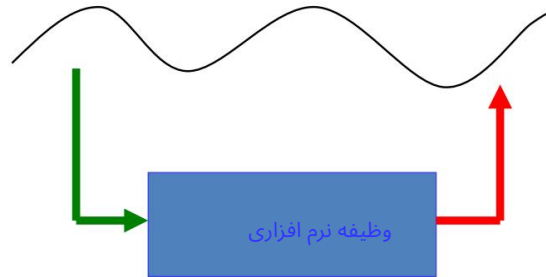
وجود نداشته باشد

-محاسبات (منطقی) آنی نیستند ->

هیچ مشکلی با بازخورد وجود ندارد

-همه مدل ها سازنده هستند

مدل برنامه نویسی LET (زمان اجرای منطقی).



خواندن ورودی
سنسور در زمان t

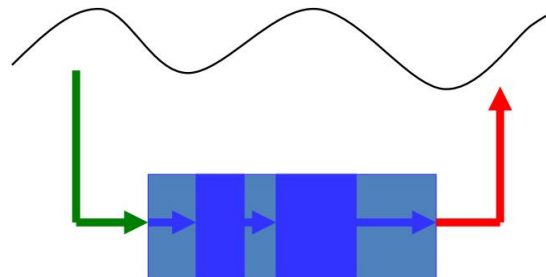
خروجی محرک را در زمان
 $t+d$ برای d مشخص
بنویسید

مثال: Giotto, TDL, ...

سیستم های زمان واقعی جاسازی شده

23

مدل برنامه نویسی LET (زمان اجرای منطقی).



زمان t

اجرای واقعی روی
CPU

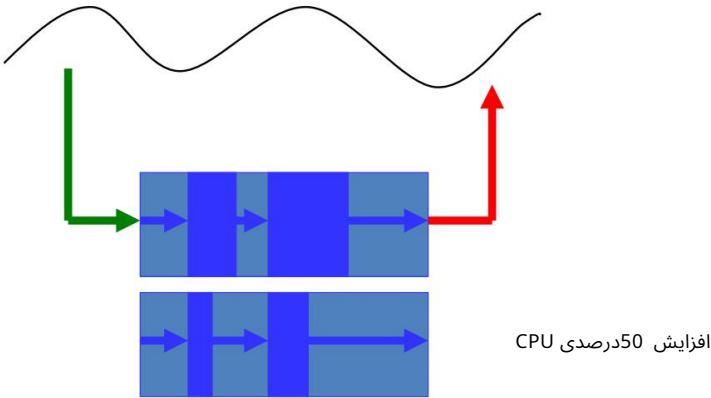
خروجی بافر

زمان $t+d$

سیستم های زمان واقعی جاسازی شده

24

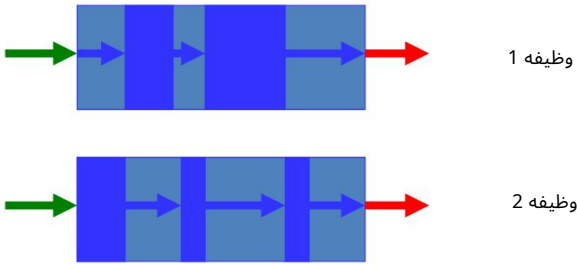
قابل حمل بودن



سیستم های زمان واقعی جاسازی شده

25

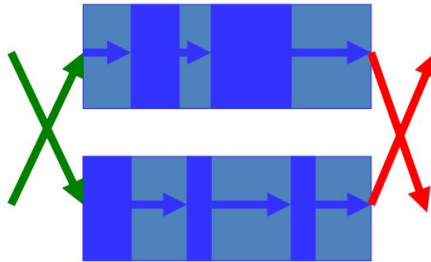
ترکیب پذیری



سیستم های زمان واقعی جاسازی شده

26

جبرگرای

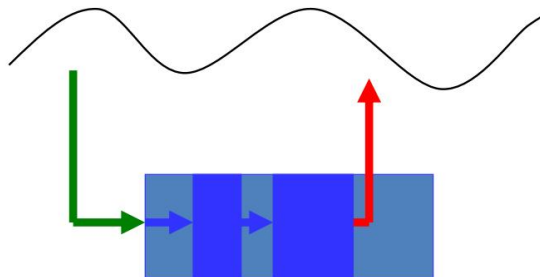


قابلیت پیش بینی زمان: حداقل لرزش
قابلیت پیش بینی عملکرد: بدون شرایط مسابقه

سیستم های زمان واقعی جاسازی شده

27

LET را با تمرین استاندارد مقایسه کنید

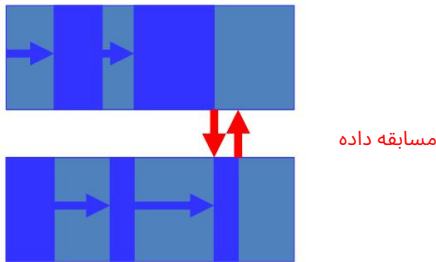


خروجی را به محض آماده شدن
در دسترس قرار دهید

سیستم های زمان واقعی جاسازی شده

28

LET را با تمرین استاندارد مقایسه کنید

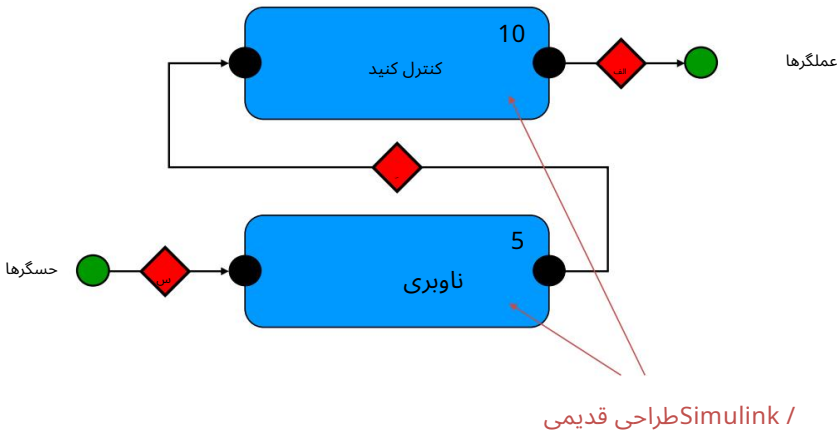


مسابقه داده

سیستم های زمان جاسازی شده

29

نرم افزار هلیکوپتر ساده شده

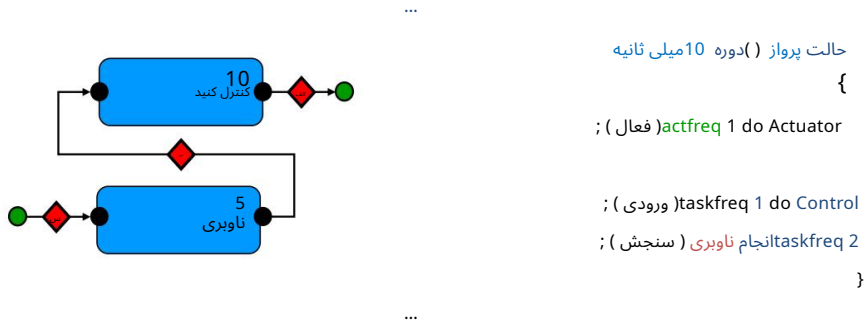


Simulink / طراحی قدیمی

سیستم های زمان جاسازی شده

30

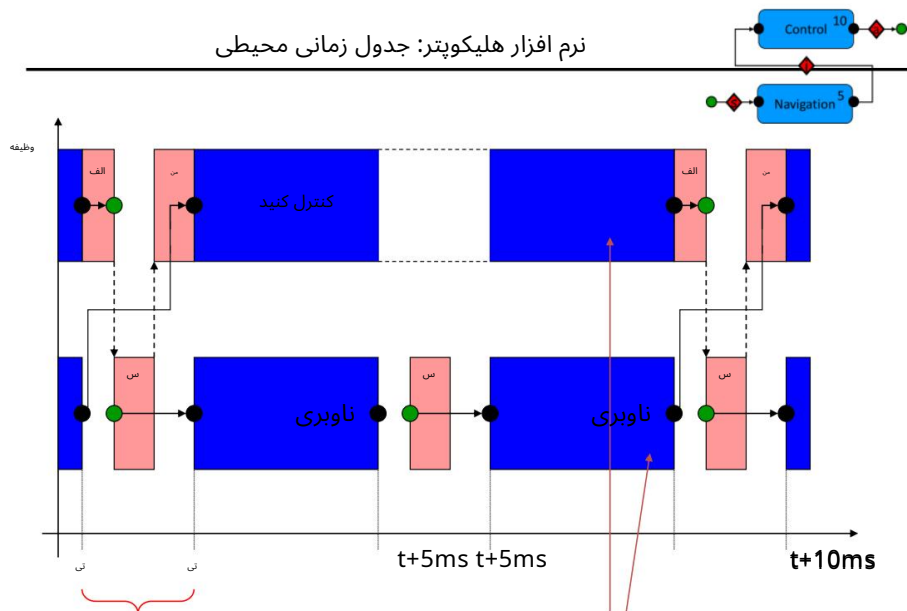
نرم افزار هلیکوپتر: Giotto Syntax



سیستم های زمان واقعی جاسازی شده

31

نرم افزار هلیکوپتر: جدول زمانی محیطی



بلوک های کد در زمان مشخص شده

(غیر قابل پیش بینی) شده (قابل پیش گرفتن)

32

MoC رویداد گسسته

• ده ها سال برای ساخت استفاده شده است

شبیه سازی برای کاربردهای مختلف

• DEVS توسعه ماشین های مور است که طول عمر غیر صفر را با

هر حالت مرتبط می کند

• در DE رویدادها با مهر زمان وقف می شوند

سیستم های زمان واقعی جاسازی شده

33

MoC رویداد گسسته

• مدل DE شبکه ای از بازیگران است که در آن بازیگران

-به رویدادهای ورودی به ترتیب مهر زمانی واکنش نشان دهید

-تولید رویدادهای خروجی به ترتیب مهر زمان

• برای اجرا، از صف رویداد، که لیستی از رویدادها است که بر اساس مهر

زمانی مرتب شده اند، استفاده می شود

-اولین رویداد را در صف رویداد انتخاب کنید و

تعیین اینکه کدام بازیگر باید آن رویداد را دریافت کند

-بازیگر واکنش نشان می دهد ("آتش") و می تواند خروجی تولید کند

رویدادها (قرار گرفته در صف)

سیستم های زمان واقعی جاسازی شده

34

سخنرانی بعدی

• پردازنده های جاسازی شده

سیستم های زمان واقعی جاسازی شده

35

اسلایدهای یدکی

سیستم های زمان واقعی جاسازی شده

36

تمديد SR MoC با زمان: Ptides MoC

• تیک های ساعت همزمان را برای رخ دادن در یک زمان در برخی از مدل های زمان تعریف کنید.

سیگنال های بین بازیگران اکنون شامل رویدادهای مهر زمانی است .

[به لی و ژنگ، "بهره گیری از اصول زبان همزمان برای مدل سازی و طراحی ناهمگن سیستم های جاسازی شده مراجعه کنید."
EMSOFT، 30 سپتامبر، 13 اکتبر، 2007 سالزبورگ، اتریش]

سیستم های زمان واقعی جاسازی شده

37

ریشه های ایده

Using Time Instead of Timeout for Fault-Tolerant Distributed Systems

LESLIE LAMPORT
SRI International



A general method is described for implementing a distributed system with any desired degree of fault-tolerance. Instead of relying upon explicit timeouts, processes execute a simple clock-driven algorithm. Reliable clock synchronization and a solution to the Byzantine Generals Problem are assumed.

Categories and Subject Descriptors: C.2.4 [Computer-Communications Networks]: Distributed Systems—network operating systems; D.1.3 [Programming Techniques]: Concurrent Programming; D.4.1 [Operating Systems]: Process Management—synchronization; D.4.3 [Operating Systems]: File Systems Management—distributed file systems; D.4.5 [Operating Systems]: Reliability—fault-tolerance; D.4.7 [Operating Systems]: Organization and Design—distributed systems; real-time systems

General Terms: Design, Reliability

Additional Key Words and Phrases: Clocks, transaction commit, timestamps, interactive consistency, Byzantine Generals Problem

تراکنش های ACM روی زبان ها و سیستم های برنامه نویسی، 1984.

سیستم های زمان واقعی جاسازی شده

38

Ptides - A Robust Distributed DE MoC

برای برنامه های IoIT

In Proceedings of the 13th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 07), Bellevue, WA, United States.

A Programming Model for Time-Synchronized Distributed Real-Time Systems

Yang Zhao
EECS Department
UC Berkeley

Jie Liu
Microsoft Research
One Microsoft Way

Edward A. Lee
EECS Department
UC Berkeley

چکیده: مدل‌های رویداد گسسته (DE) مشخصات سیستم رسمی هستند که رفتارهای قطعی قابل تحلیل دارند. با استفاده از یک مفهوم جهانی و ثابت از زمان، اجزای DE از طریق رویدادهای دارای مهر زمانی ارتباط برقرار می‌کنند. مدل‌های DE عمدتاً در مدل‌سازی و شبیه‌سازی عملکرد استفاده می‌شوند، جایی که مهرهای زمانی یک ویژگی مدل‌سازی هستند که هیچ ارتباطی با زمان واقعی در طول اجرای مدل ندارند. در این مقاله، مدل‌های DE را با قابلیت ربط دادن رویدادهای خاص به زمان فیزیکی گسترش می‌دهیم.

سیستم های زمان واقعی جاسازی شده

39

Google Spanner یک اختراع مجدد -

Spanner: Google's Globally-Distributed Database

James C. Corbett, Jeffrey Dean, Michael Epstein, Andrew Fikes, Christopher Frost, JJ Furman, Sanjay Ghemawat, Andrey Gubarev, Christopher Heiser, Peter Hochschild, Wilson Hsieh, Sebastian Kanthak, Eugene Kogan, Hongyi Li, Alexander Lloyd, Sergey Melnik, David Mwaura, David Nagle, Sean Quinlan, Rajesh Rao, Lindsay Rolig, Yasushi Saito, Michal Szymaniak, Christopher Taylor, Ruth Wang, Dale Woodford

Google, Inc.

Abstract

Spanner is Google's scalable, multi-version, globally-distributed, and synchronously-replicated database. It is the first system to distribute data at global scale and support externally-consistent distributed transactions. This paper describes how Spanner is structured, its feature set, the rationale underlying various design decisions, and a novel time API that exposes clock uncertainty. This API and its implementation are critical to supporting external consistency and a variety of powerful features: non-blocking reads in the past, lock-free read-only transactions, and atomic schema changes, across all of Spanner.

tency over higher availability, as long as they can survive 1 or 2 datacenter failures.

Spanner's main focus is managing cross-datacenter replicated data, but we have also spent a great deal of time in designing and implementing important database features on top of our distributed-systems infrastructure. Even though many projects happily use Bigtable [9], we have also consistently received complaints from users that Bigtable can be difficult to use for some kinds of applications: those that have complex, evolving schemas, or those that want strong consistency in the presence of wide-area replication. (Similar claims have been made by other authors [37].) Many applications at Google

مجموعه مقالات OSDI 2012

سیستم های زمان واقعی جاسازی شده

40

گوگل به طور مستقل
تکنیک بسیار مشابهی را
توسعه داد و آن را در پایگاه
های داده توزیع شده به کار
برد.

PTIDES: معنانشناسی رویداد گسسته

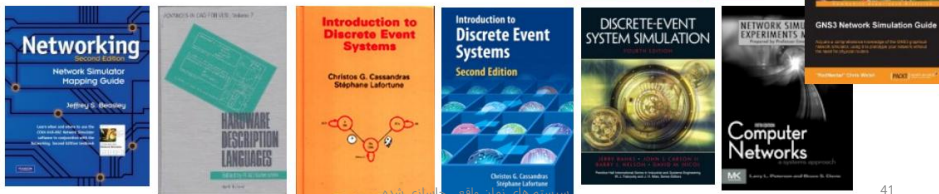
+ ساعت های همگام + حسگرها و محرک ها

رویدادهای دارای مهر زمانی که به ترتیب مهر زمانی پردازش می شوند.

چند متن که از IDE MoC استفاده می کنند

این MoC به طور گسترده در شبیه سازی و HDL ها استفاده می شود.

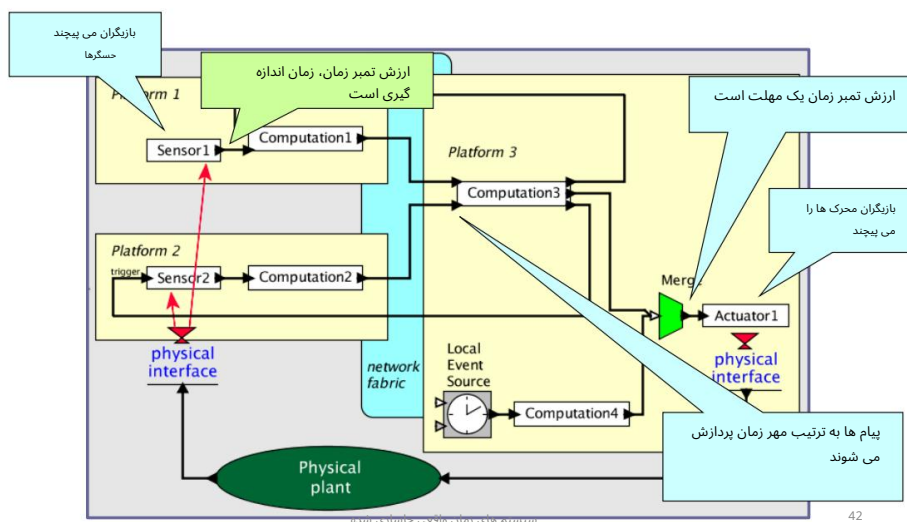
با توجه به ورودی های مهر زمانی، آن یک MoC همزمان قطعی است.



41

Ptides: مهرهای زمانی به زمان واقعی در حسگرها و محرک ها

متصل می شوند



42

قطعی توزیع شده

زمان واقعی

حدود را در نظر بگیرید:

• خطای همگام سازی ساعت

• تأخیر شبکه

سپس رویدادها به ترتیب مهر زمانی در هر جزء پردازش می شوند. اگر علاوه بر آن فرض کنیم

• محدودیت در زمان اجرا

سپس رویدادها به موقع به محرک ها تحویل داده می شوند.

به <http://chess.eecs.berkeley.edu/ptides> مراجعه کنید

سیستم های زمان واقعی جاسازی شده

43

اینهمه فرضیه؟

ساعت های غیر همگام

• همه مفروضات با فناوری امروزی قابل دستیابی هستند و به هر حال برای سیستم های بلادرنگ نیازمندی هستند.

• یک مدل Ptides الزامات را واضح می کند.

• نقض الزامات به عنوان رویدادهای خارج از نظم قابل تشخیص است و می تواند به عنوان خطا در نظر گرفته شود.

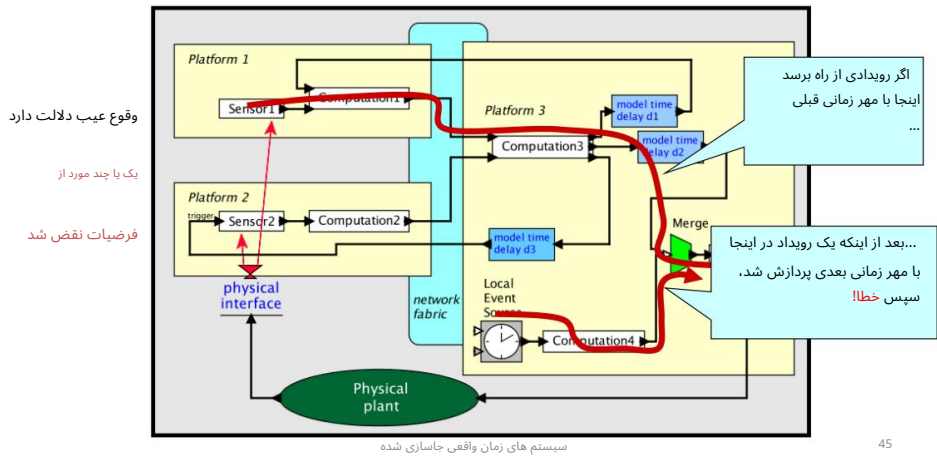


سیستم های زمان واقعی جاسازی شده

44

رسیدگی به عیوب

یک خطا به صورت رویدادهای خارج از نظم ظاهر می شود.



پتیدها

• همگام سازی ساعت با دقت بالا در همه جا حاضر خواهد شد.

• شبکه ها قادر خواهند بود تاخیرهای محدود را "تضمین" کنند.

• زمان مناسب است.