

سخنرانی MoCs: 9 جریان داده

سید حسین عطارزاده نیاکی

برخی از اسلایدها به خاطر ادوارد لی

سیستم های زمان واقعی جاسازی شده

1

بررسی کنید

• ترکیب ماشین های حالت

- ترکیب سنکرون

- ترکیب ناهمزمان

• ترکیب سلسله مراتبی: StateCharts

سیستم های زمان واقعی جاسازی شده

2

مدل های جریان داده

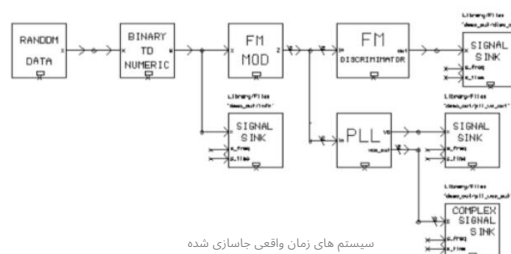
• ثابت شده است که برای مشخص کردن برنامه های پخش جریانی مفید است

به عنوان مثال، پردازش سیگنال و حوزه های ارتباطات

• شبیه سازی الگوریتم در حالت عملکردی یا رفتاری سطح

• ترکیب به نرم افزار (به عنوان مثال، یک برنامه C یا سخت افزار (به عنوان مثال، VHDL)

• برنامه ریزی بصری مبتنی بر بلوک دیگرام



سیستم های زمان واقعی جاسازی شده

3

سیگنال های جریان داده (جریان ها)

• در سیگنال های جریان داده، ارتباط بین بازیگران است

به صورت دنباله ای از پیام ها انجام می شود

• هر پیام یک نشانه نامیده می شود.

• سیگنال یک تابع جزئی از فرم V_s است

س: n

V_s نوع سیگنال است • سیگنال روی یک قطعه اولیه $N \in \{0, 1, \dots, n\}$ (برای

اجراهای بی نهایت) در کل مجموعه N تعریف می شود.

• برخلاف MoC واکنشی همزمان، هیچ رابطه زمانی لازم بین ورودی و خروجی یک بازیگر وجود ندارد.

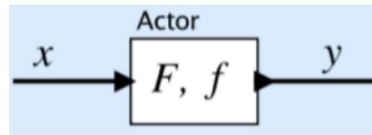
-آنها هماهنگ نیستند.

سیستم های زمان واقعی جاسازی شده

4

بازیگران جریان داده

یک بازیگر (معین) به عنوان تابعی توصیف می‌شود که
توالی‌های ورودی را به دنباله‌های خروجی نگاشت می‌کند.
یک تابع بازیگر F, f ، کل دنباله‌های ورودی را به کل دنباله‌های
خروجی نگاشت می‌کند. یک تابع شلیک، f ، بخش محدودی از
دنباله‌های ورودی را به دنباله‌های خروجی نگاشت می‌کند.
یک قانون شلیک مشخص می‌کند



$$F(x_1, x_2, x_3, \dots) = (ax_1, ax_2, ax_3, \dots)$$

به عنوان مثال، $f(x_1) = (ax_1)$

تعداد توکن‌های مورد نیاز در هر پورت ورودی برای فعال
کردن بازیگر

سیستم‌های زمان واقعی جاسازی شده

5

ترکیب بازیگران جریان داده: ارتباط بافر

• از آنجایی که شلیک بازیگران **ناهمزمان است**، رمز ارسال شده از یک بازیگر به بازیگر
دیگر باید **بافر** شود.

• هنگامی که بازیگر مقصد شلیک می‌کند، یک یا را **مصرف می‌کند**
توکن‌های ورودی بیشتر • توانایی اجرای یک مدل جریان داده برای همیشه،
اجرای نامحدود نامیده می‌شود.

- ما به سیاست‌های زمان‌بندی نیاز داریم که بافرهای محدود ارائه کنند.

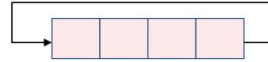
بن‌بست چالش دیگری است که در چرخه‌ها ظاهر می‌شود.



سیستم‌های زمان واقعی جاسازی شده

6

پیاده سازی بافرها برای مدل های جریان داده



• بافرهای نامحدود به طرح های تخصیص حافظه و توزیع نیاز دارند.

• بافرهای اندازه محدود را می توان به صورت بافرهای دایره ای یا بافرهای حلقه ای، در یک آرایه تخصیص داده شده به صورت ایستا، درک کرد.

- یک اشاره گر خواندن r یک شاخص در آرایه است که به اولین مکان خالی اشاره دارد.

بعد از هر بار خواندن این مقدار را افزایش دهید.

- یک عدد n یک عدد بدون علامت است که به ما می گوید چند مورد داده در بافر وجود دارد. (یا یک اشاره گر دم)

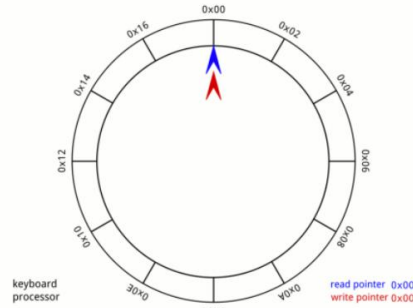
- مکان بعدی برای نوشتن $(r + n)$ است طول بافر مدولو

- اگر $n == 0$ باشد، بافر خالی است

- اگر n طول بافر باشد، بافر پر است

- می تواند n را به صورت یک شمارش پیاده سازی کند

- سمارفور، ایجاد طرد متقابل برای کدهایی که r را تغییر می دهند.



سیستم های زمان واقعی جاسازی شده

7

حقایق در مورد

Dynamic Dataflow (عمومی)

• آیا برنامه ای وجود دارد که وجود ندارد

بن بست غیرقابل تصمیم گیری است

• اینکه آیا برنامه ای وجود دارد که برای همیشه با حافظه محدود اجرا شود، غیر قابل

تصمیم گیری است .

محل های ذخیره داده در این معجزه است که هیچ داده ای در حافظه محدود وجود ندارد

بتواند به سوال در زمان محدود برای همه مدل های محدود پاسخ دهد.

در زمان محدود برای همه مدل های محدود.

با این حال، موارد خاصی وجود دارد که مدل ها قابل تجزیه و تحلیل هستند

سیستم های زمان واقعی جاسازی شده

8

جریان داده: بسیاری از انواع، حوزه تحقیقاتی هنوز فعال است

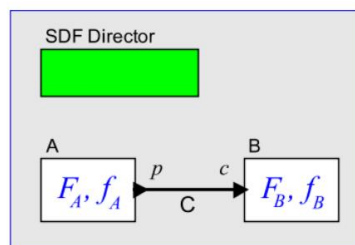
- Computation graphs [Karp & Miller - 1966]
- Process networks [Kahn - 1974]
- Static dataflow [Dennis - 1974]
- Dynamic dataflow [Arvind, 1981]
- K-bounded loops [Culler, 1986]
- Synchronous dataflow [Lee & Messerschmitt, 1986]
- Structured dataflow [Kodosky, 1986] now
- PGM: Processing Graph Method [Kaplan, 1987]
- Synchronous languages [Lustre, Signal, 1980's]
- Well-behaved dataflow [Gao, 1992]
- Boolean dataflow [Buck and Lee, 1993]
- Multidimensional SDF [Lee, 1993]
- Cyclo-static dataflow [Lauwereins, 1994]
- Integer dataflow [Buck, 1994]
- Bounded dynamic dataflow [Lee and Parks, 1995]
- Heterochronous dataflow [Girault, Lee, & Lee, 1997]
- Parameterized dataflow [Bhattacharya and Bhattacharyya 2001]
- Structured dataflow (again) [Thies et al. 2002]
- ...

Lee 09: 6

سیستم های زمان واقعی جاسازی شده

9

جریان داده همزمان (SDF)



• تعداد توکن های مصرف شده و تولید شده توسط

اخراج یک بازیگر ثابت است

• تجزیه و تحلیل استاتیک می تواند به ما بگوید که آیا می توانیم برنامه ریزی کنیم یا خیر

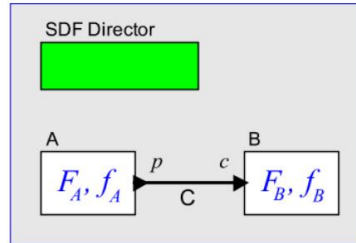
اخراج برای به دست آوردن یک اعدام مفید

• اگر چنین است، می توان یک نمایش محدود از یک برنامه زمانی برای چنین اجرائی ایجاد کرد.

سیستم های زمان واقعی جاسازی شده

10

معادلات تعادل



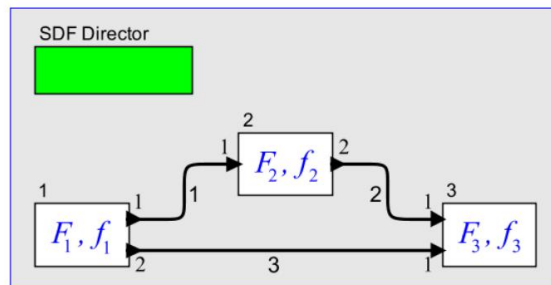
فرض کنید q_A , تعداد شلیک بازیگران A و B باشد.
فرض کنید c_C , تعداد توکن های تولید و مصرف شده در اتصال C باشد.
سیس سیستم در تعادل است اگر برای همه اتصالات $c_C = q_B$ $p_C = q_A$ که در آن توکن هایی را در C تولید می کند و B آنها را مصرف می کند.

سیستم های زمان واقعی جاسازی شده

11

مثال

این مثال را در نظر بگیرید، جایی که بازیگران و کمان ها شماره گذاری می شوند:

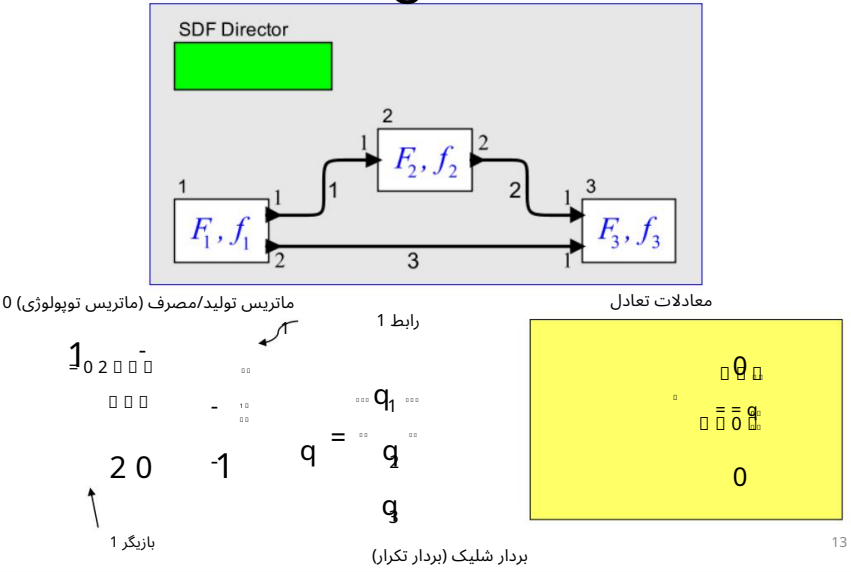


معادلات تعادل نشان می دهد که بازیگر 3 باید دو برابر دو بازیگر دیگر شلیک کند.

سیستم های زمان واقعی جاسازی شده

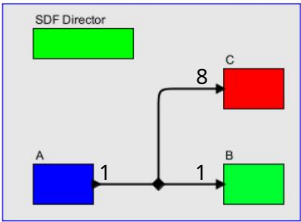
12

به طور فشرده نشان می دهد معادلات تعادل



سوال

•ماتریس تولید/مصرف در این مورد چیست؟



$$K = \begin{bmatrix} 1 & 0 & -1 \\ 1 & -8 & 0 \end{bmatrix}$$

مثال

SDF Director

حل معادلات تعادل:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

این به ما می گوید که بازیگر 3 باید دو برابر بازیگران 1 و 2 شلیک کند.

15

مثال

SDF Director

اما راه حل های زیادی برای معادلات تعادل وجود دارد:

وجود ندارد برای مدل های «خوش رفتار»، یک جواب 2 حداقل مثبت 0 منحصر به فرد 1

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

سپستم های زمان واقعی جاسازی شده

16

حداقل راه حل عدد صحیح مثبت به معادلات تعادل

نتیجه داشته باشید که: معادله تعادل تعداد توکن‌های تولید و مصرف شده در اتصال، C ، اعداد

$$q_A p_C = q_B c_C$$

دالیت دارد:

q_A منطقی است اگر و فقط اگر q_B منطقی باشد.

q_A مثبت است اگر و فقط اگر q_B مثبت باشد.

• نتیجه: در هر جزء متصل، اگر برای معادلات تعادل راه حل غیر صفر وجود داشته باشد، یک راه حل حداقل مثبت یکتا وجود دارد.

سیستم های زمان واقعی جاسازی شده

17

رتبه یک ماتریس

• **رتبه** یک ماتریس \square تعداد سطرها یا ستون های مستقل خطی است. معادله

$$\square q \square$$

ترکیبی خطی از ستون های \square را تشکیل می دهد.

• چنین ترکیب خطی فقط می تواند صفر را به دست دهد اگر ستون ها به صورت خطی وابسته باشند (این همان چیزی است که به معنای وابستگی خطی است).

• اگر دارای یک ستون و ردیف b باشد، رتبه نمی تواند از آن بیشتر شود.
دقیقه (الف، ب).

• اگر ستون ها یا ردیف های \square دوباره مرتب شوند، ماتریس به دست آمده همان رتبه \square را دارد.

سیستم های زمان واقعی جاسازی شده

18

رتبه از

ماتریس تولید/مصرف

• فرض کنید a تعداد بازیگران یک نمودار متصل باشد.
سپس رتبه ماتریس تولید/مصرف \square
 a است. (چرا؟)

• اگر مدل یک درخت پوشا باشد (به این معنی که به ندرت اتصالات کافی برای اتصال آن وجود دارد) آنگاه \square دارای یک ردیف و یک ستون 1 - است. • قضیه [Lee-Messerschmitt'87]: رتبه آن 1 است.

-تمرین: آن را ثابت کنید. (نکته: از القاء استفاده کنید).

• نتیجه: رتبه هر تولید/مصرف
ماتریس یک نمودار متصل یا a یا 1 - است. (چرا؟)

مدل های سازگار



• تعداد بازیگران یک مدل متصل را در نظر بگیرید.
اگر \square دارای رتبه 1 - باشد، مدل سازگار است.

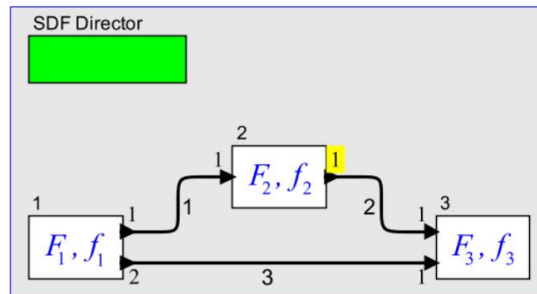
• اگر رتبه a باشد، معادلات تعادل فقط دارند
یک راه حل بی اهمیت (صفر شلیک).

• هنگامی که \square دارای رتبه 1 a باشد، آنگاه معادلات موازنه همیشه
یک راه حل غیر ضروری دارند.

مثالی از یک مدل ناسازگار:

بدون راه حل غیر پیش پا افتاده برای معادلات تعادل

$$\begin{array}{ccc} 1 & \square & 10 \square \\ 0 & \square & -1 \square \\ 2 & 0 & -1 \square \end{array}$$



• این ماتریس تولید/مصرف دارای رتبه 3 است، بنابراین هیچ راه حل غیر ضروری برای معادلات تعادل وجود ندارد.

• توجه داشته باشید که این مدل می تواند برای همیشه اجرا شود، اما نیاز دارد حافظه بی حد و حصر

سیستم های زمان واقعی جاسازی شده

21

شرایط لازم و کافی

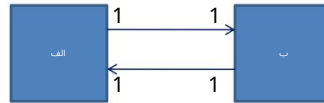
سازگاری شرط لازم برای اجرای بی نهایت (حافظه محدود) است.

آیا کافی است؟

سیستم های زمان واقعی جاسازی شده

22

بن بست 1

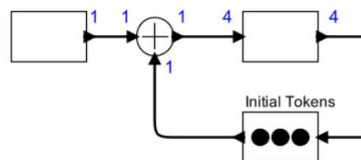


آیا این نمودار سازگار است؟

سیستم های زمان واقعی جاسازی شده

23

بن بست 2



برخی از مدل های جریان داده نمی توانند برای همیشه اجرا شوند. در مدل فوق، حلقه بازخورد توکن های اولیه را تزریق می کند ، اما برای اجرای مدل کافی نیست .

سیستم های زمان واقعی جاسازی شده

24

SDF: از تحلیل استاتیک تا زمان‌بندی

• داده شده: نمودار SDF
یافتن: یک برنامه زمان‌بندی بافر محدود، در صورت وجود

• مرحله 0: بررسی کنید که آیا نمودار سازگار است یا خیر. اگر نه، هیچ برنامه بافر محدودی وجود ندارد.

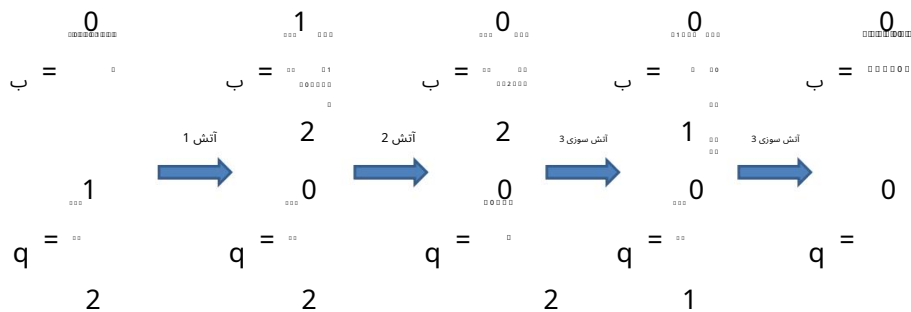
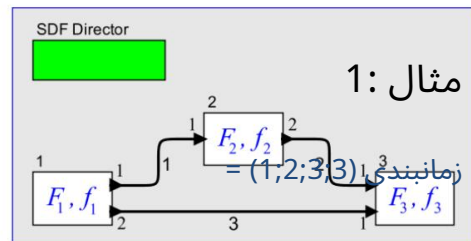
• مرحله 1: یک راه حل عدد صحیح برای $q=0$ پیدا کنید.

• مرحله 2: "تجزیه" محلول q را به یک جدول زمانی، اطمینان حاصل کنید که بافرها هرگز منفی نمی‌شوند.

سیستم‌های زمان واقعی جاسازی شده

25

مرحله 2: "تجزیه" بردار شلیک



سیستم‌های زمان واقعی جاسازی شده

26

مرحله 2: "تجزیه" بردار شلیک



اگر بخواهیم فرار کنیم چه اتفاقی می افتد
رویه قبلی
در این مثال؟

بنابراین، ما هر دو لازم را داریم
و شرایط کافی برای
زمان بندی نمودارهای SDF

سیستم های زمان واقعی جاسازی شده

27

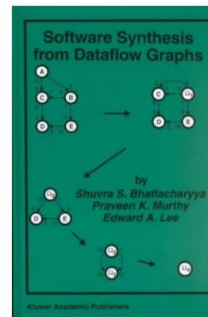
یک سوال کلیدی: اگر بیش از یک بازیگر در مرحله 2 قابل آتش نشاندن باشد، چگونه یکی را انتخاب کنم؟

معیارهای بهینه سازی که ممکن است اعمال شوند:

• اندازه بافر را به حداقل برسانید.

• تعداد فعال سازی بازیگر را به حداقل برسانید.

• اندازه نمایش برنامه (اندازه کد) را به حداقل برسانید.



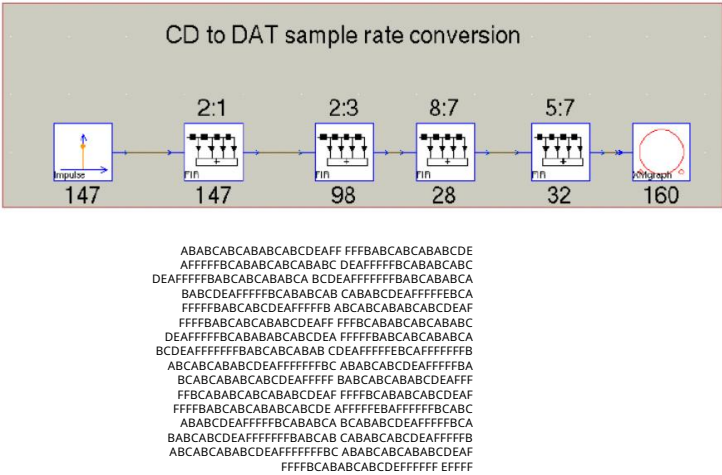
رجوع کنید به: Software Synthesis from Dataflow Graphs, Kluwer Academic Press, 1996. و SS Bhattacharyya, PK Murthy, EA Lee.

فراتر از محدوده ما در اینجا، اما اشاره می کند که این یک مشکل جالب است...

سیستم های زمان واقعی جاسازی شده

28

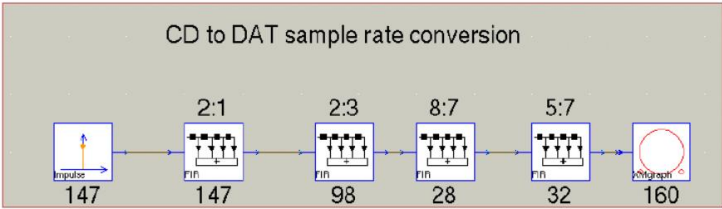
برنامه حداقل بافر



منبع: شوورا باتاچاريا

29

زمان بندی معاملات
(Bhattacharyya, Parks, Pino)



استراتژی برنامه ریزی	داده های کد
حداقل برنامه بافر، بدون حلقه	13735 32
حداقل برنامه بافر، با حلقه	9400
بدترین برنامه حداقل اندازه کد	1020
بهترین برنامه حداقل اندازه کد	200

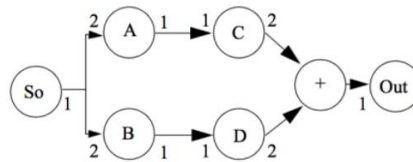
منبع: شوورا باتاچاريا

سیستم های زمان واقعی جاسازی شده

30

SDF همگن

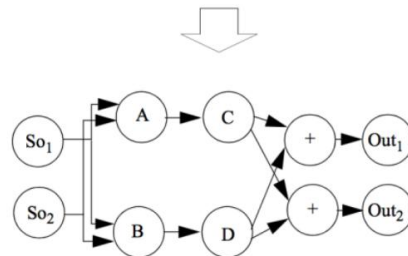
• یک نمودار SDF که در آن هر بازیگر از هر یک از آن ها فقط یک نشانه مصرف و تولید می کند



ورودی ها و خروجی ها

• کلی، سازگار

نمودار SDF که یک گراف نیست
HSDFG همیشه می تواند به HSDFG
معادل تبدیل شود



سیستم های زمان واقعی جاسازی شده

31

تکنیک های مبتنی بر نمودار برای آنالیز (H)SDF

• به دست آوردن زمان بندی های امکان پذیر با استفاده از کوتاه ترین مسیرها در نمودارها

• حداکثر میانگین چرخه (MCM) برای به دست آوردن حداکثر توان عملیاتی در دسترس یک نمودار HSDF زمان بندی شده استفاده می شود .

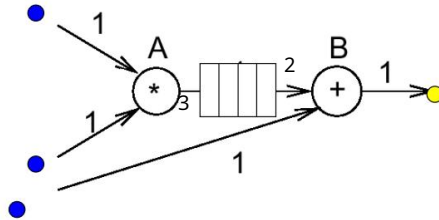
$$MCM(G) = \max_{\text{cycle } C \text{ in } G} \left\{ \frac{\sum_{v \text{ is on } C} t(v)}{\text{Delay}(C)} \right\}$$

سیستم های زمان واقعی جاسازی شده

32

انعکاس

• چگونه می توانیم اندازه های بافر محدود را در SDF مدل کنیم؟

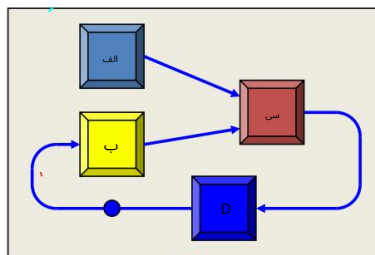


سیستم های زمان واقعی جاسازی شده

33

برنامه ریزی موازی مدل های SDF

SDF مناسب برای
نقشه برداری خودکار روی پردازنده های
موازی و سنتز مدارهای موازی.



بسیاری از مسائل بهینه
سازی زمان بندی را می
توان فرموله کرد.



متوالی



موازی

سیستم های زمان واقعی جاسازی شده

34

جریان داده بولی (BDF)

SDF نمی تواند مستقیماً شلیک مشروط را بیان کند

-به عنوان مثال، جایی که یک بازیگر تنها در صورتی شلیک می کند که یک نشانه دارای یک علامت باشد

ارزش خاص

• دو بازیگر اصلی SWITCH و SELECT در جریان داده بولی اضافه

می شوند

• هر ماشین تورینگ را می توان به صورت BDF بیان کرد

نمودار

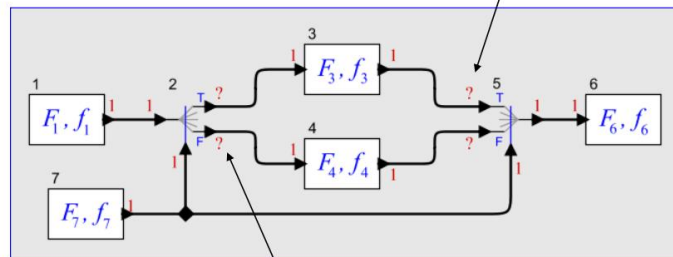


سیستم های زمان واقعی جاسازی شده

35

جریان داده بولی

چه میزان مصرف؟



ضروری است
معادل:

در حالی که (درست) {
چه میزان تولید؟

$x = f1();$

$b = f7();$

مدل SDF if-then-else نیست.

{ اگر (b) }

اما ما به وضوح می توانیم یک برنامه شبه استاتیک

{ دیگر }

محدود برای آن ارائه دهیم:

$y = f4(x);$

(1, 7, 2, b?3, !b?4, 5, 6)

}
 $f6(y);$

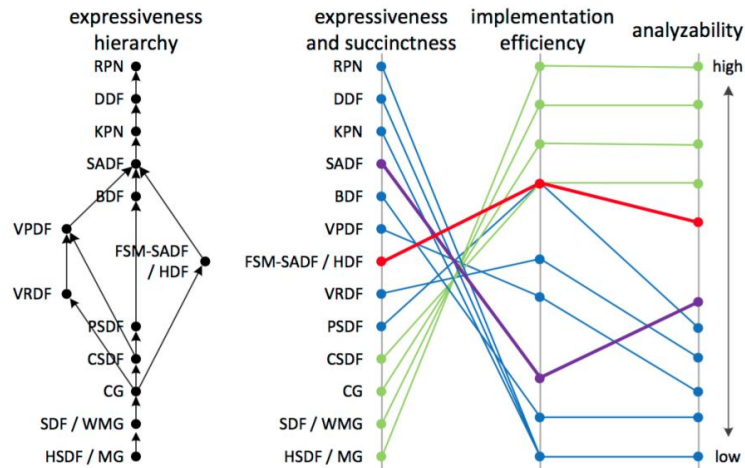
}

نگهبان

سیستم های زمان واقعی جاسازی شده

36

مقایسه CoM های جریان داده



Stuijk و همکاران، SAMOS'11

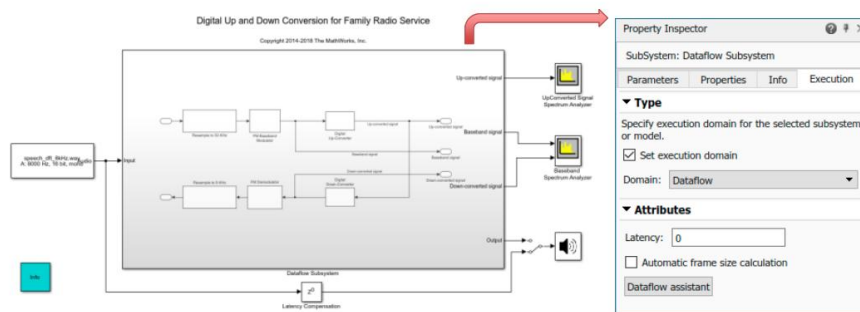
سیستم های زمان واقعی جاسازی شده

37

مدلسازی جریان داده در سیمولینک استفاده از Dataflow Domain در جعبه ابزار DSP

• بلوک های داخل دامنه جریان داده بر اساس در دسترس بودن داده ها بر خلاف زمان نمونه سیمولینک اجرا می شوند.

• Simulink به طور خودکار سیستم را به پارتیشن بندی می کند رشته های همزمان برای افزایش توان داده برای شبیه سازی سریع و تولید کد چند هسته ای.



سیستم های زمان واقعی جاسازی شده

38