

Precision Timed Infrastructure

IHI Meeting

February 20, 2013



David Broman

broman@eecs.berkeley.edu

UC Berkeley and
Linköping University

PRET Infrastructure at Berkeley

David Broman	Edward A. Lee
Jian Cai	Aviral Shrivastava
Hokeun Kim	Chris Shaver
Yooseong Kim	Michael Zimmer

PRET Machine Collaborators and Alumni

Steven A. Edwards	Isaac Liu	Jan Reineke
Jeff Jensen	Slobadan Matic	Sanjit Seshia
Sungjun Kim	Hiren Patel	Jia Zou

A Story...

2

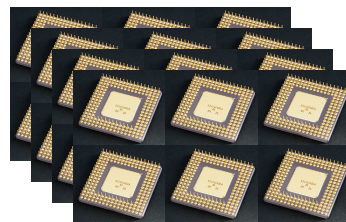
broman@eecs.berkeley.edu



Fly-by-wire technology
controlled by software.

Safety critical →
Rigorous validation and certification

Success?



They have to purchase and store
microprocessors for at least 50 years
production and maintenance...

Why?

Apparently, the software does not
specify the behaviour that has
been validated and certified!

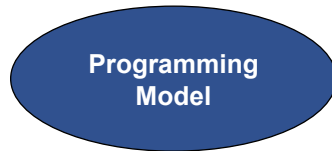
What is PRET?

broman@eecs.berkeley.edu

Timing is not part of the software semantics

Correct execution of programs (e.g., in C, C++, C#, Java, Scala, Haskell, OCaml) has nothing to do with how long time things takes to execute.

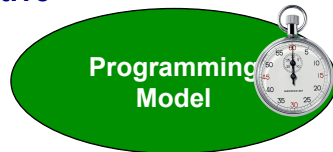
Traditional Approach



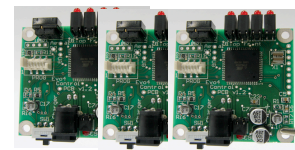
Timing Dependent on the Hardware Platform



Our Objective



Make time an abstraction within the programming model



Timing is independent of the hardware platform (within certain constraints)

What is Precision Timed (PRET) Infrastructure?

broman@eecs.berkeley.edu

A vision of making time first class citizen in both software and hardware.

PRET Infrastructure

- PRET Language (Language with timing semantics)
- PRET Compiler (Timing aware compilation)
- PRET Machine (Computer Architecture)

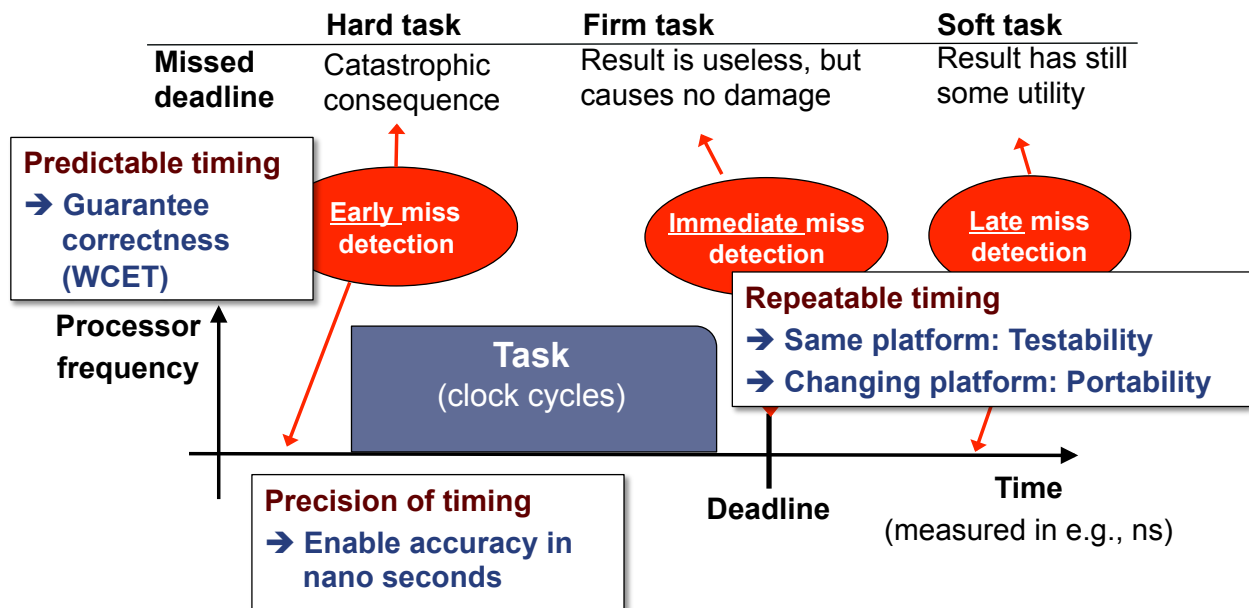


What do mean by precision, predictable, and repeatable timing?

5

broman@eecs.berkeley.edu

Focus on cyber-physical systems with real-time constraints



Languages with timing semantics

6

broman@eecs.berkeley.edu

Modeling Languages

Simulink/Stateflow
(Mathworks)

Modelica
(Modelica Associations)

Ptolemy II
(Eker et al., 2003)

Giotto
(Henzinger, Horowitz, and Kirsch, 2003)

Modelyze
(Broman and Siek, 2012)

Programming Languages

Real-time Concurrent C
(Gehani and Ramamritham, 1991)

PRET-C
(Andalam et al., 2009)

Assembly Languages

The assembly languages for today's processors lack the notion of time

Precision Timed Machine

broman@eecs.berkeley.edu

Rethink the ISA

Timing has to be a *correctness* property not only a *performance* (quality) property

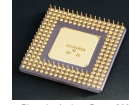


Photo by Andrew Dunn, 2005

PRET Machine

- Repeatable and predictable execution time
- Repeatable memory access time
- Timing instructions for handling missed deadline detection

PRET Infrastructure

broman@eecs.berkeley.edu

**Modeling
Languages**

**Simulink/
Stateflow**
(Mathworks)

Modelica
(Modelica
Associations)

Ptolemy II
(Eker et al., 2003)

Giotto
(Henzinger, Horowitz,
and Kirsch, 2003)

Modelyze
(Broman and
Siek, 2012)

**Programming
Languages**

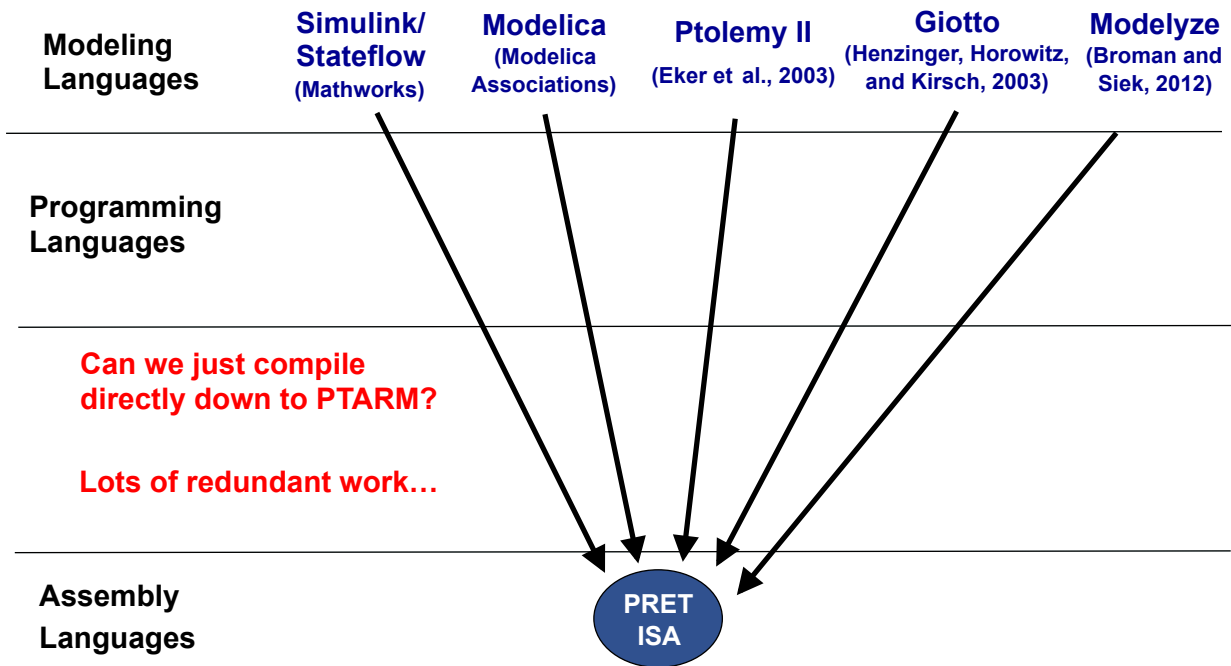
**Semantic gap between
timed high level modeling
languages and PRET ISA**

**Assembly
Languages**

**PRET
ISA**

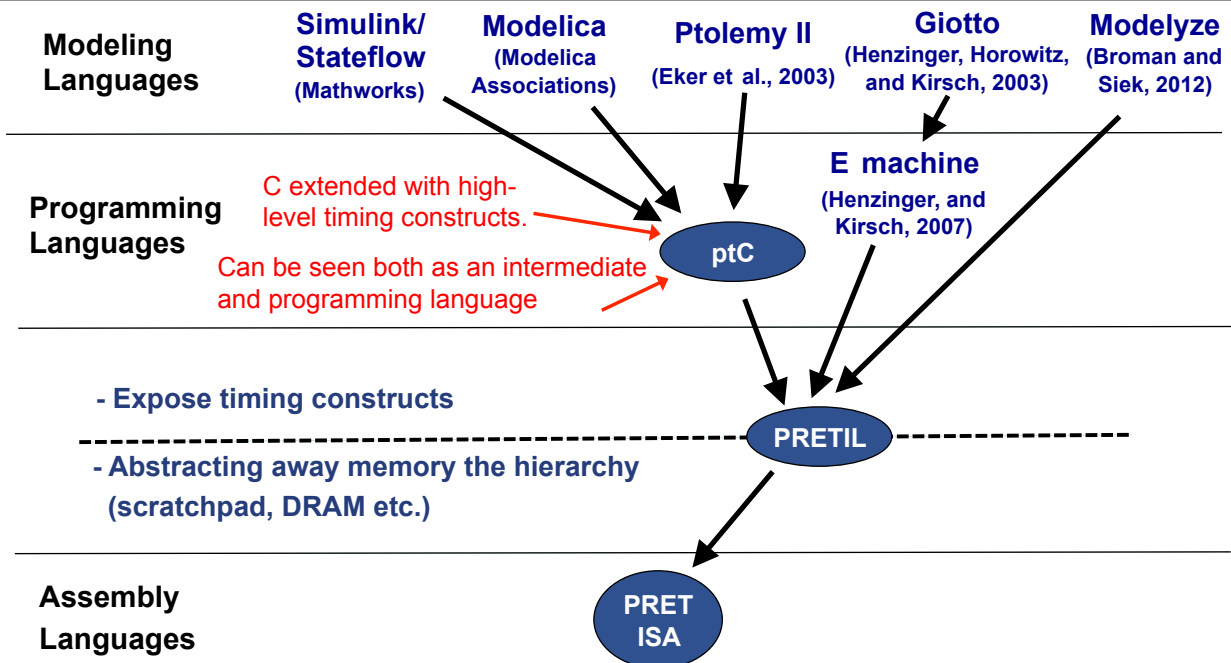
PRET Infrastructure

broman@eecs.berkeley.edu



PRETIL vision

broman@eecs.berkeley.edu



PRETIL vision

broman@eecs.berkeley.edu

