

Modeling and Analysis: Measuring

by *Gerrit Muller* University of South-Eastern Norway-NISE

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

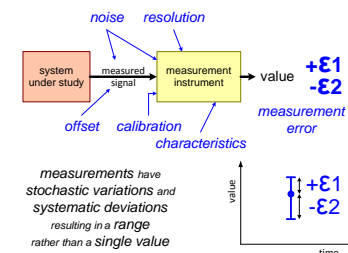
Abstract

This presentation addresses the fundamentals of measuring: What and how to measure, impact of context and experiment on measurement, measurement errors, validation of the result against expectations, and analysis of variation and credibility.

Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

September 9, 2018
status: preliminary
draft
version: 1.2



content

What and How to measure

Impact of experiment and context on measurement

Validation of results, a.o. by comparing with expectation

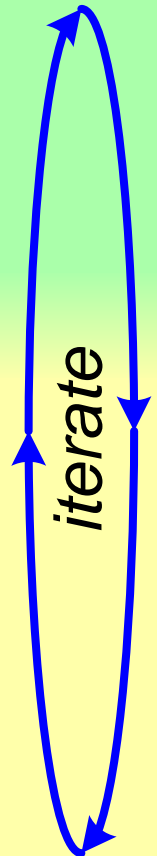
Consolidation of measurement data

Analysis of variation and analysis of credibility

Measuring Approach: What and How

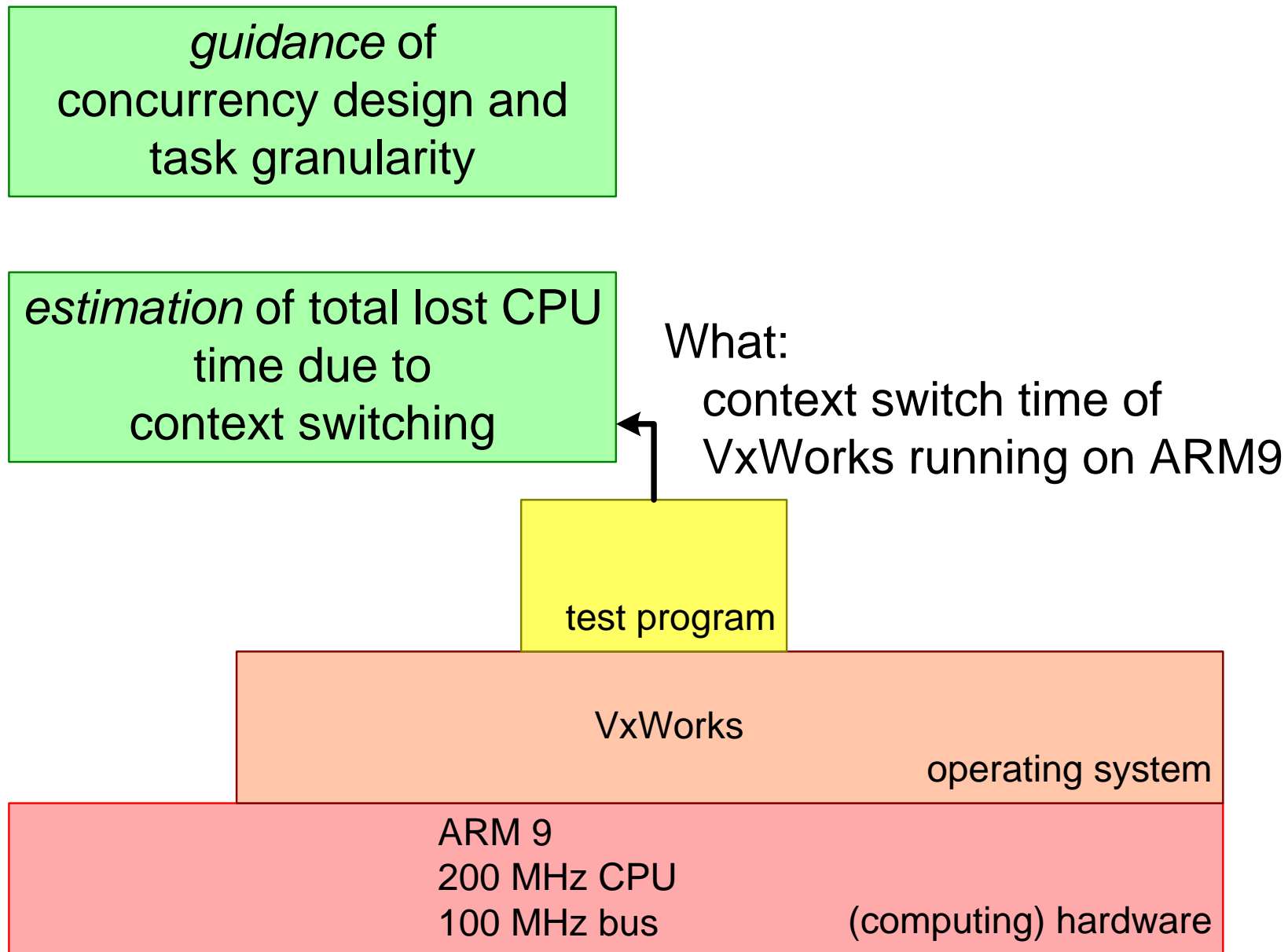
what

- | | |
|--|-----------------------------------|
| 1. What do we need to know? | |
| 2. Define quantity to be measured. | initial model |
| 3. Define required accuracy | purpose |
| 4A. Define the measurement circumstances | fe.g. by use cases |
| 4B. Determine expectation | historic data or estimation |
| 4C. Define measurement set-up | |
| 5. Determine actual accuracy | uncertainties, measurement error |
| 6. Start measuring | |
| 7. Perform sanity check | expectation versus actual outcome |



how

1. What do We Need? Example Context Switching

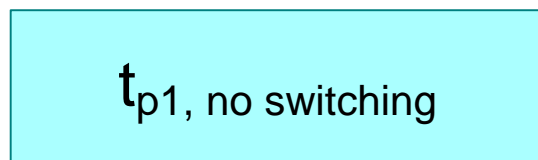


2. Define Quantity by Initial Model

What (original):
context switch time of
VxWorks running on ARM9

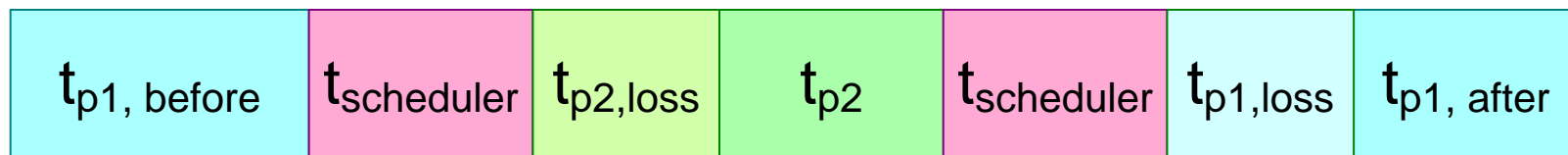
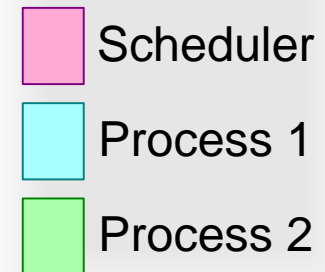
What (more explicit):
The amount of lost CPU time,
due to context switching on
VxWorks running on ARM9
on a heavy loaded CPU

$$t_{\text{context switch}} = t_{\text{scheduler}} + t_{p1, \text{loss}}$$



time →

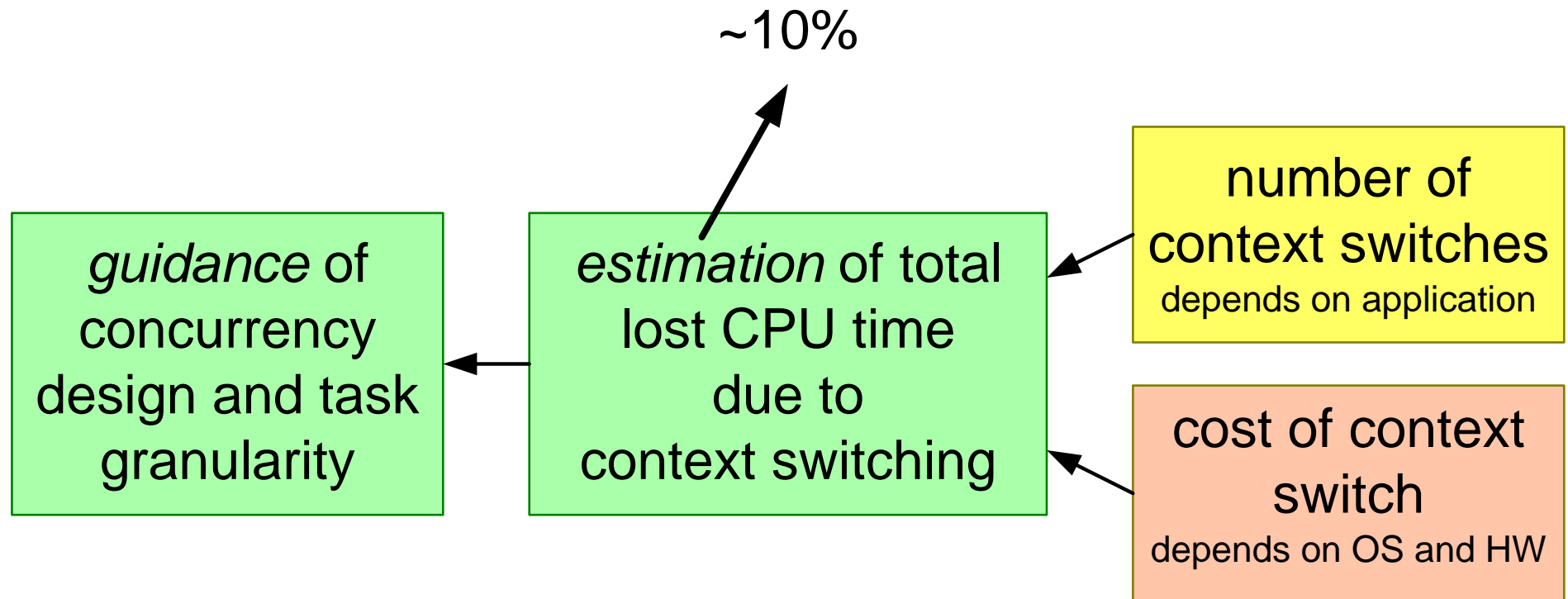
legend



p2 pre-empts p1

*p1 resumes
= lost CPU time*

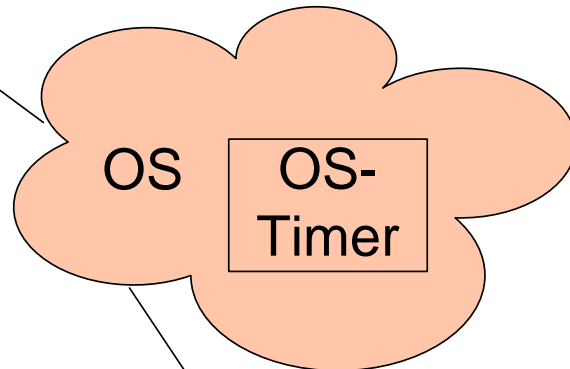
3. Define Required Accuracy



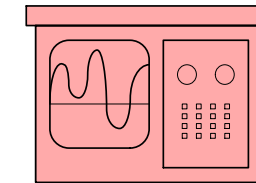
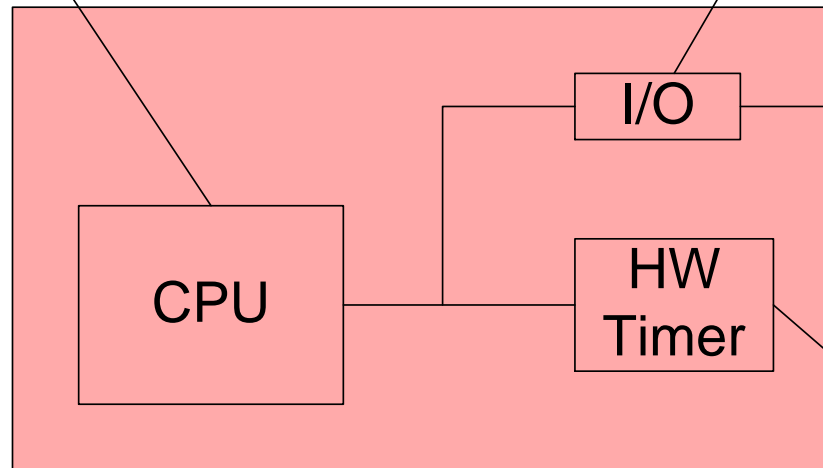
purpose drives required accuracy

Intermezzo: How to Measure CPU Time?

Low resolution (~ μs - ms)
Easy access
Lot of instrumentation



High resolution (~ 10 ns)
requires
HW instrumentation

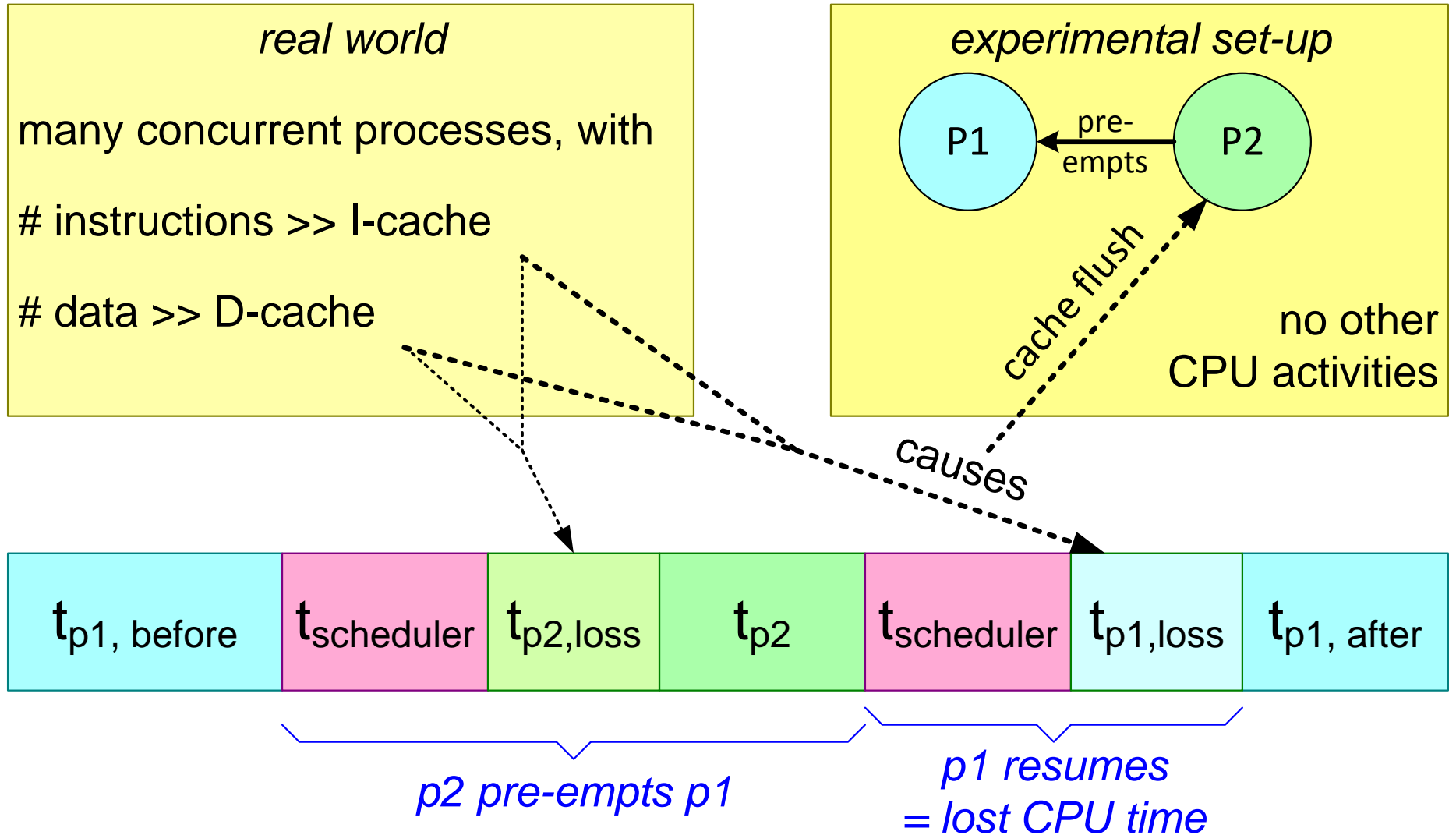


Logic analyzer /
Oscilloscope

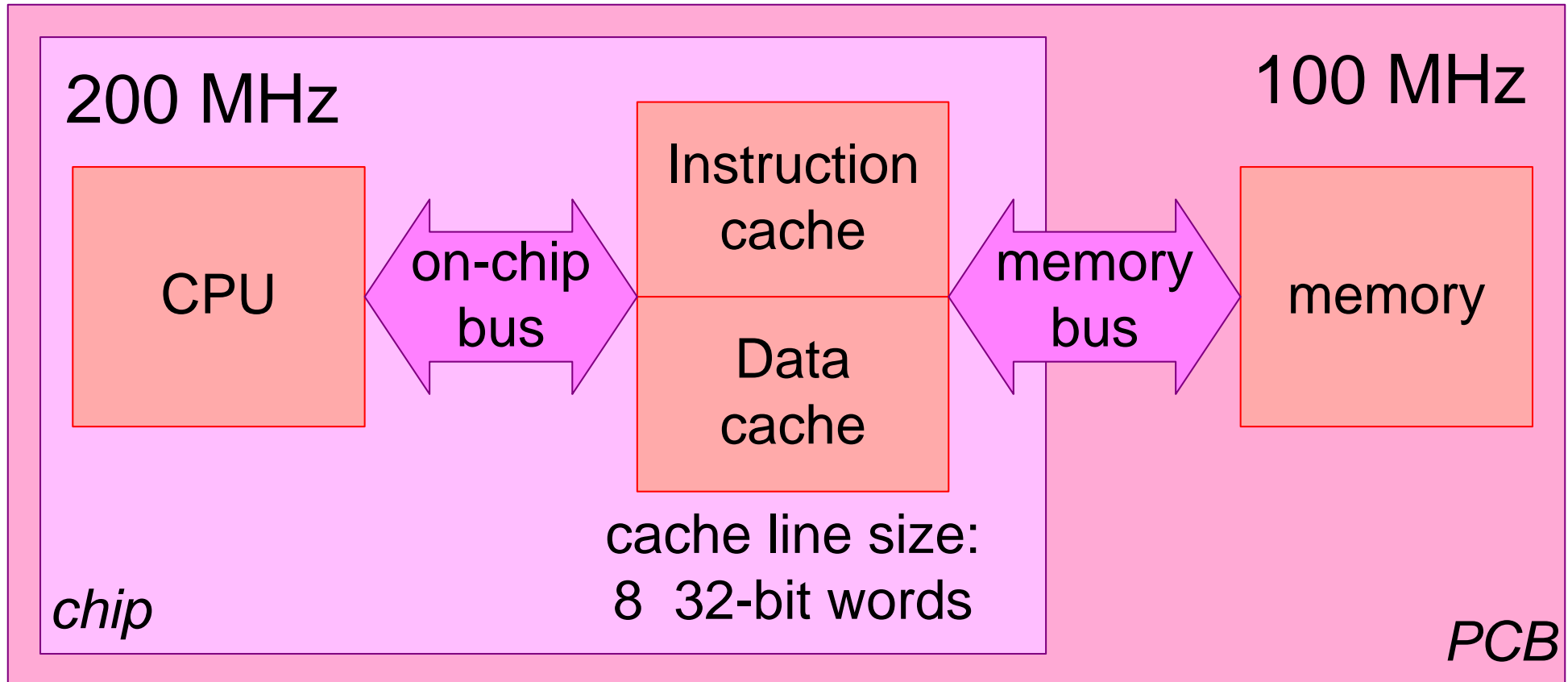
High resolution (~ 10 ns)
Cope with limitations:
- Duration (16 / 32 bit
counter)
- Requires Timer Access

4A. Define the Measurement Set-up

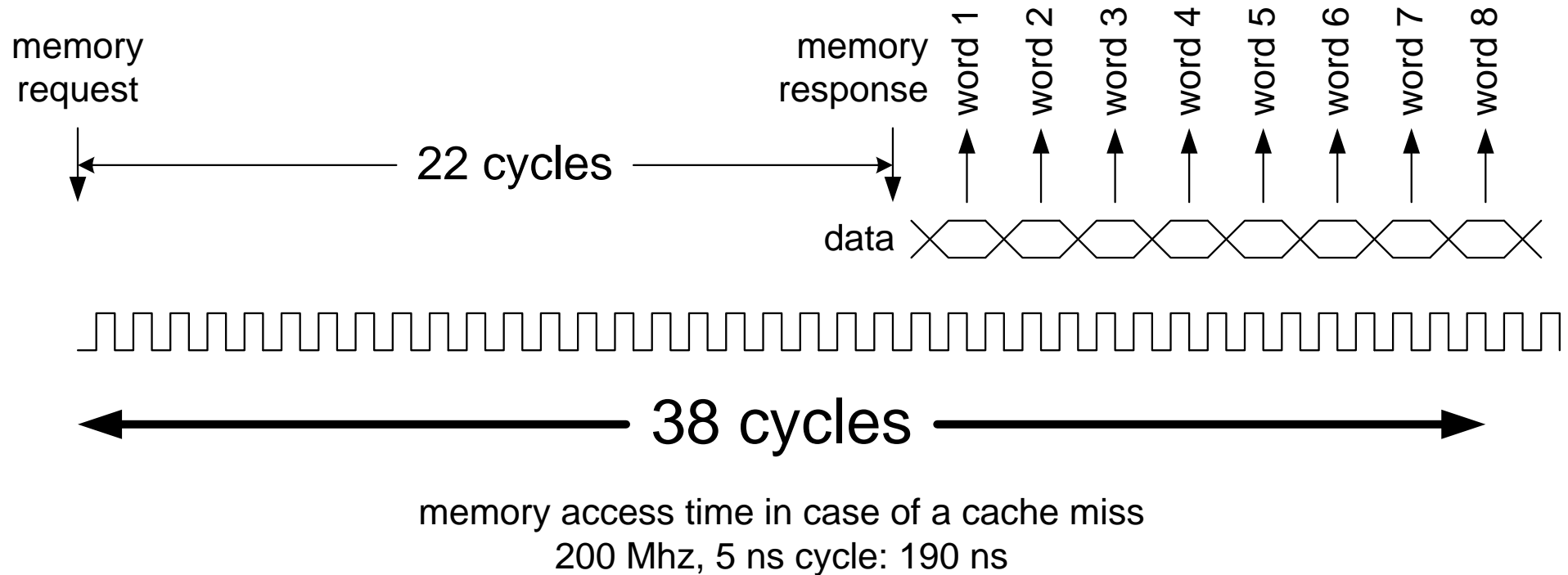
Mimick relevant real world characteristics



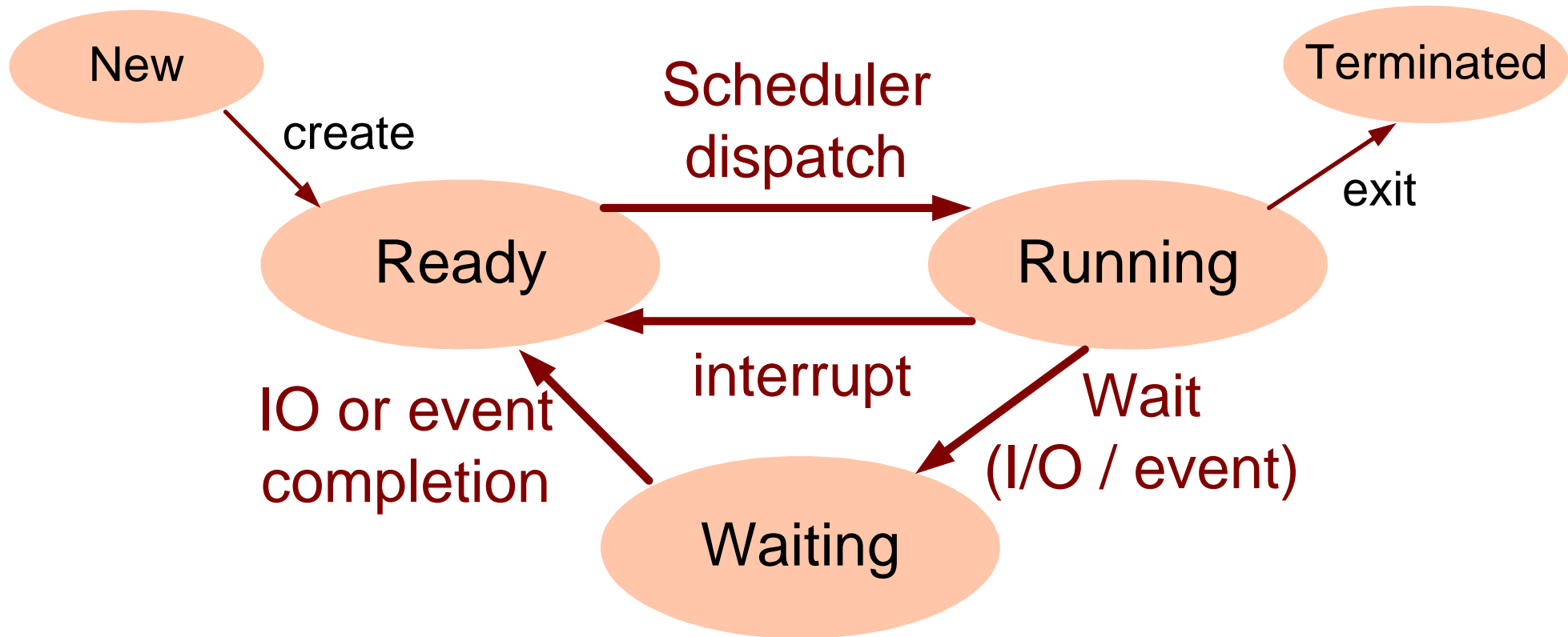
4B. Case: ARM9 Hardware Block Diagram



Key Hardware Performance Aspect



OS Process Scheduling Concepts



Determine Expectation

simple SW model of context switch:

save state P1

determine next runnable task

update scheduler administration

load state P2

run P2

input data HW:

$t_{\text{ARM instruction}} = 5 \text{ ns}$

$t_{\text{memory access}} = 190 \text{ ns}$

Estimate how many
instructions and memory accesses
are needed per context switch

Calculate the estimated time
needed per context switch

Determine Expectation Quantified

instructions
memory
accesses

10	1
50	2
20	1
10	1
10	1
<hr/>	
100	6

simple SW model of context switch:

save state P1

determine next runnable task

update scheduler administration

load state P2

run P2

Estimate how many
instructions and memory accesses
are needed per context switch

500 ns
1140 ns
<hr/>
1640 ns

input data HW:

$t_{\text{ARM instruction}} = 5 \text{ ns}$

$t_{\text{memory access}} = 190 \text{ ns}$

Calculate the estimated time
needed per context switch

round up (as margin) gives expected $t_{\text{context switch}} = 2 \mu\text{s}$

4C. Code to Measure Context Switch

Task 1

Time Stamp End
Cache Flush
Time Stamp Begin
Context Switch

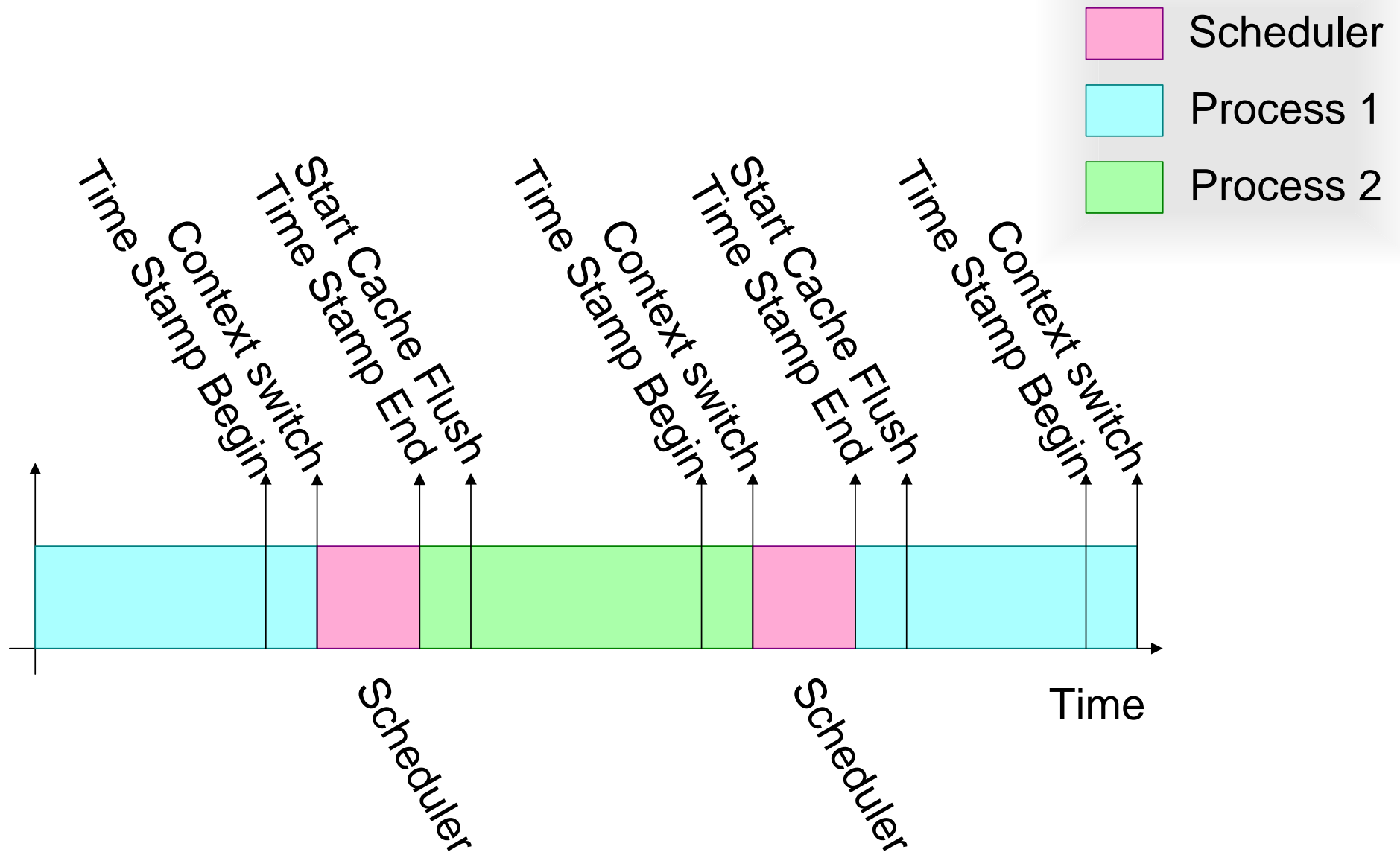
Time Stamp End
Cache Flush
Time Stamp Begin
Context Switch

Task 2

Time Stamp End
Cache Flush
Time Stamp Begin
Context Switch

Time Stamp End
Cache Flush
Time Stamp Begin
Context Switch

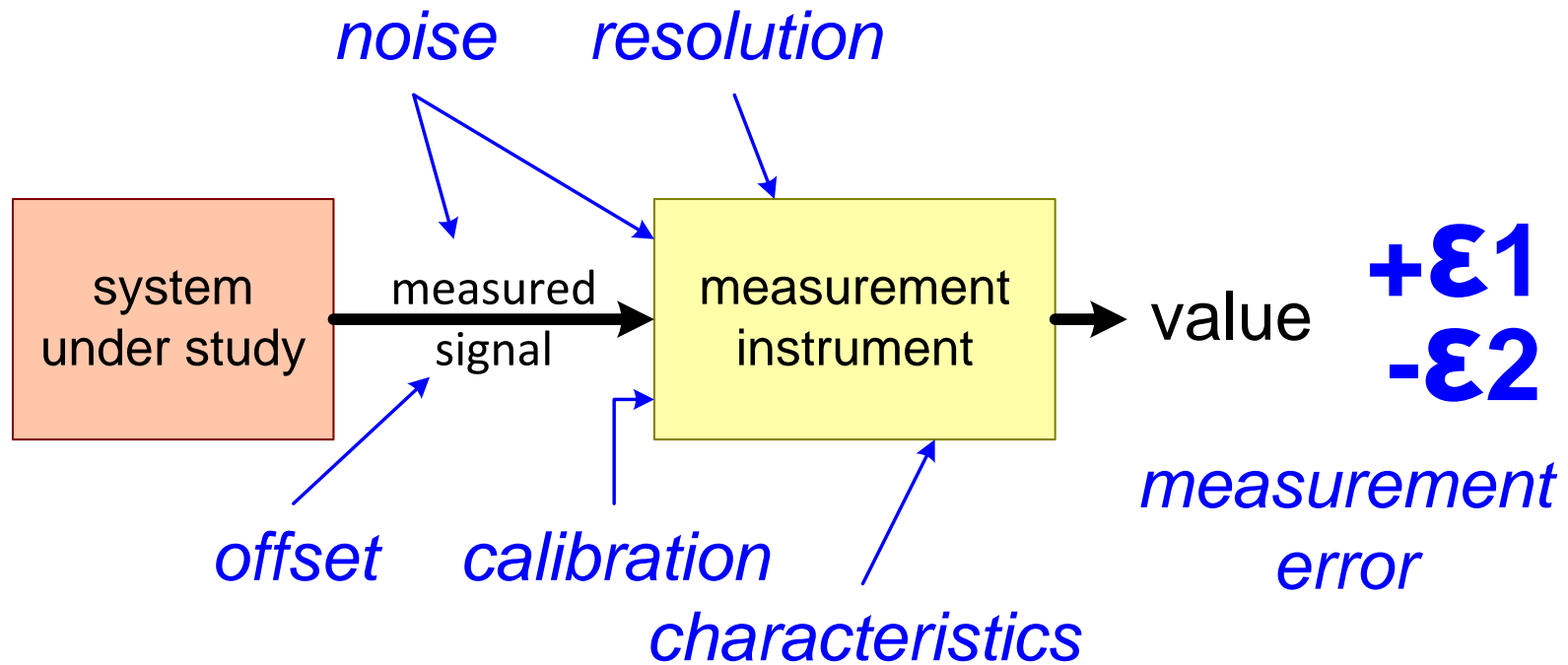
Measuring Task Switch Time



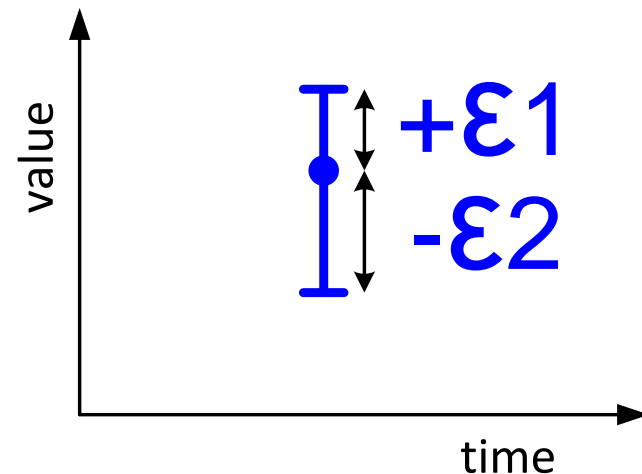
Understanding: Impact of Context Switch



5. Accuracy: Measurement Error



measurements have stochastic variations and systematic deviations resulting in a range rather than a single value



Accuracy 2: Be Aware of Error Propagation

$$t_{\text{duration}} = t_{\text{end}} - t_{\text{start}}$$

$$t_{\text{start}} = 10 \pm 2 \mu\text{s}$$

$$t_{\text{end}} = 14 \pm 2 \mu\text{s}$$

$$t_{\text{duration}} = 4 \pm ? \mu\text{s}$$

systematic errors: add linear

stochastic errors: add quadratic

Measurements have

stochastic variations and systematic deviations

resulting in a range rather than a single value.

The inputs of modeling,

"facts", assumptions, and measurement results,

also have stochastic variations and systematic deviations.

Stochastic variations and systematic deviations

propagate (add, amplify or cancel) through the model

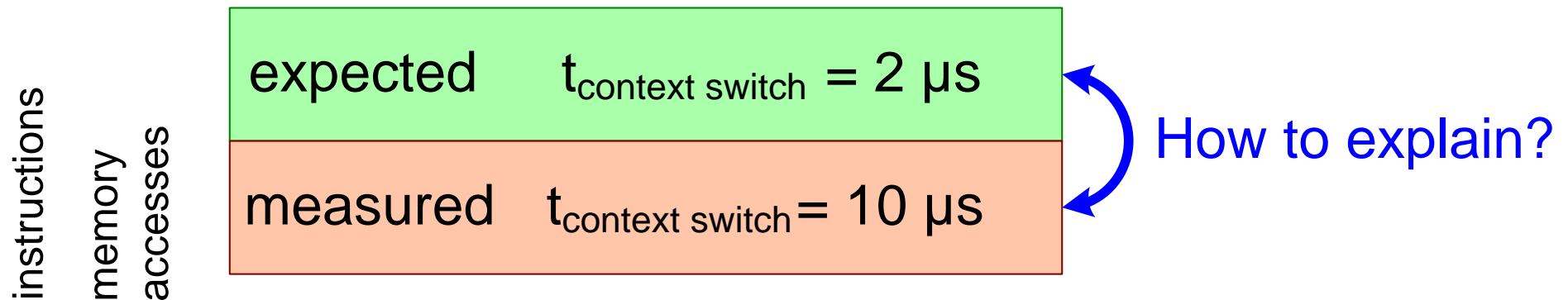
resulting in an output range.

6. Actual ARM Figures

ARM9 200 MHz $t_{\text{context switch}}$
as function of cache use

cache setting	$t_{\text{context switch}}$
From cache	2 μs
After cache flush	10 μs
Cache disabled	50 μs

7. Expectation versus Measurement



simple SW model of context switch:	
10	1 save state P1
50	2 determine next runnable task
20	1 update scheduler administration
10	1 load state P2
10	1 run P2
+	
100	6
input data HW:	
500 ns	$t_{\text{ARM instruction}} = 5 \text{ ns}$
1140 ns	$t_{\text{memory access}} = 190 \text{ ns}$
+	
1640 ns	

potentially missing in expectation:

- memory accesses due to instructions
 - ~10 instruction memory accesses ~ = 2 μs
- memory management (MMU context)
- complex process model (parents, permissions)
- bookkeeping, e.g performance data
- layering (function calls, stack handling)
- the combination of above issues

However, measurement seems to make sense

Conclusion Context Switch Overhead

$$t_{\text{overhead}} = n_{\text{context switch}} * t_{\text{context switch}}$$

$n_{\text{context switch}}$ (s^{-1})	$t_{\text{context switch}} = 10\mu s$		$t_{\text{context switch}} = 2\mu s$	
	t_{overhead}	CPU load overhead	t_{overhead}	CPU load overhead
500	5ms	0.5%	1ms	0.1%
5000	50ms	5%	10ms	1%
50000	500ms	50%	100ms	10%

Summary Context Switching on ARM9

goal of measurement

Guidance of concurrency design and task granularity

Estimation of context switching overhead

Cost of context switch on given platform

examples of measurement

Needed: context switch overhead ~10% accurate

Measurement instrumentation: HW pin and small SW test program

Simple models of HW and SW layers

Measurement results for context switching on ARM9

Conclusions

Measurements are an important source of factual data.

A measurement requires a well-designed experiment.

Measurement error, validation of the result determine the credibility.

Lots of consolidated data must be reduced to essential understanding.

Techniques, Models, Heuristics of this module

experimentation

error analysis

estimating expectations

This work is derived from the EXARCH course at CTT developed by *Ton Kostelijk* (Philips) and *Gerrit Muller*.

The Boderc project contributed to the measurement approach. Especially the work of

Peter van den Bosch (Océ),

Oana Florescu (TU/e),

and *Marcel Verhoef* (Chess)

has been valuable.