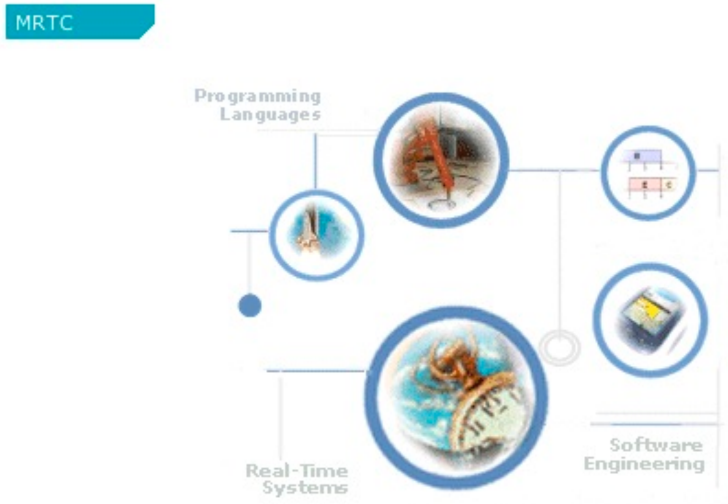
ECRTS 2012
10-13 July
Pisa, Italy

Timing Analysis for Mode Switch in Component-based Multi-mode Systems

Yin Hang, Hans Hansson
Mälardalen Real-Time Research Centre(MRTC),
Mälardalen University (MDH), Västerås, Sweden
Contact: young.hang.yin@mdh.se

MÄLARDALEN UNIVERSITY SWEDEN



Vetenskapsrådet

Overview

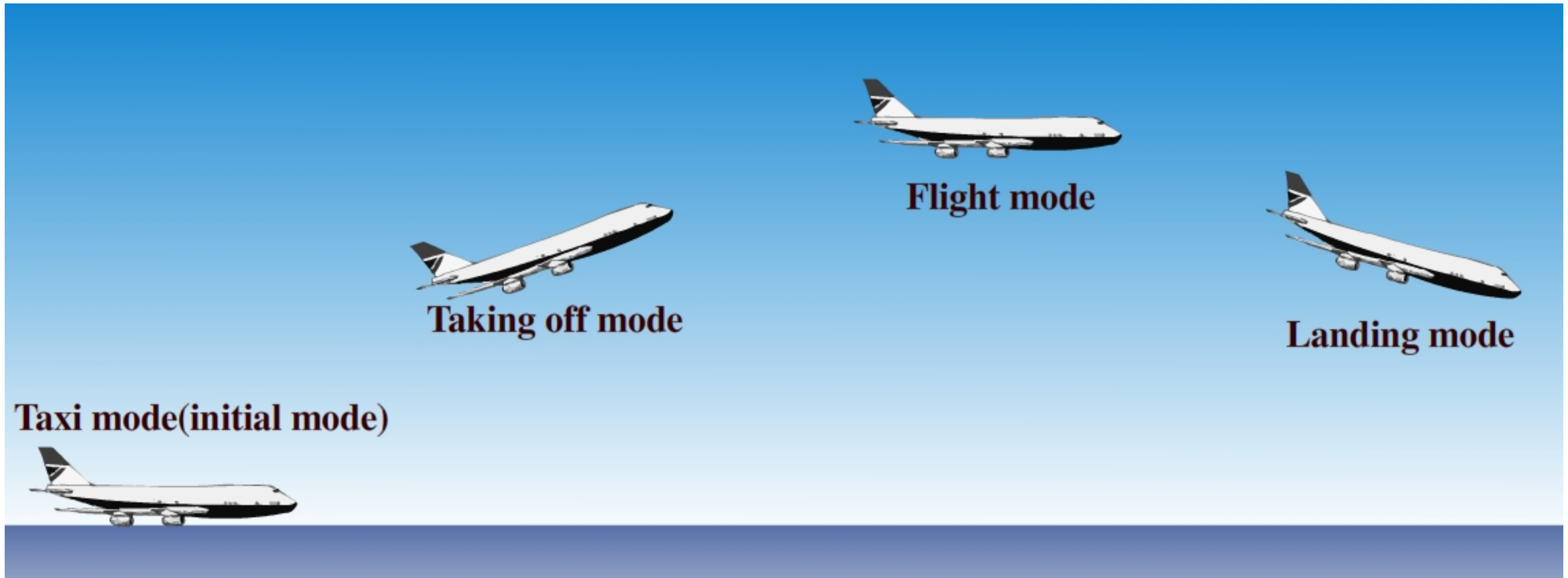
- Introduction
- The Mode Switch Logic (MSL)
- The handling of atomic component execution
- The mode switch timing analysis
- Calculating the worst-case atomic component execution time
- Conclusions and future work

Introduction

- Growing complexity of embedded systems
- A promising design paradigm: Component-Based Software Engineering(CBSE)

Component reuse

- Multi-mode systems:
  - Distinguished behaviors in different operational modes
  - To reduce complexity and improve efficiency
  - E.g. the control software of an airplane



Introduction

- Our target:
- Component-Based Software Engineering (CBSE)

  +

- Multi-mode system

  =

- **Component-Based Multi-Mode System (CBMMS)**

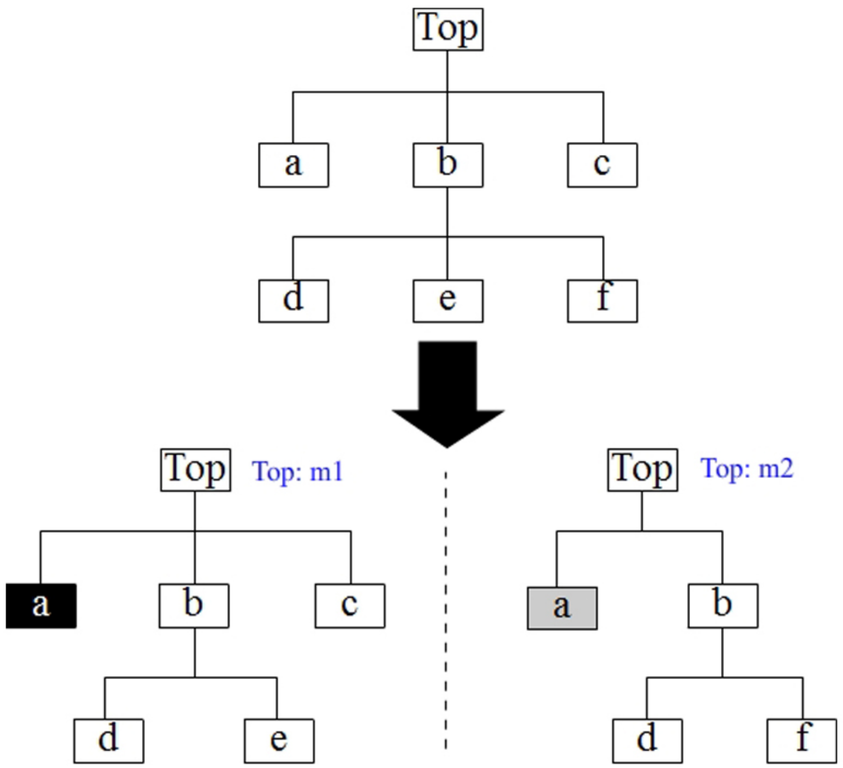
Transformers: Optimus Prime

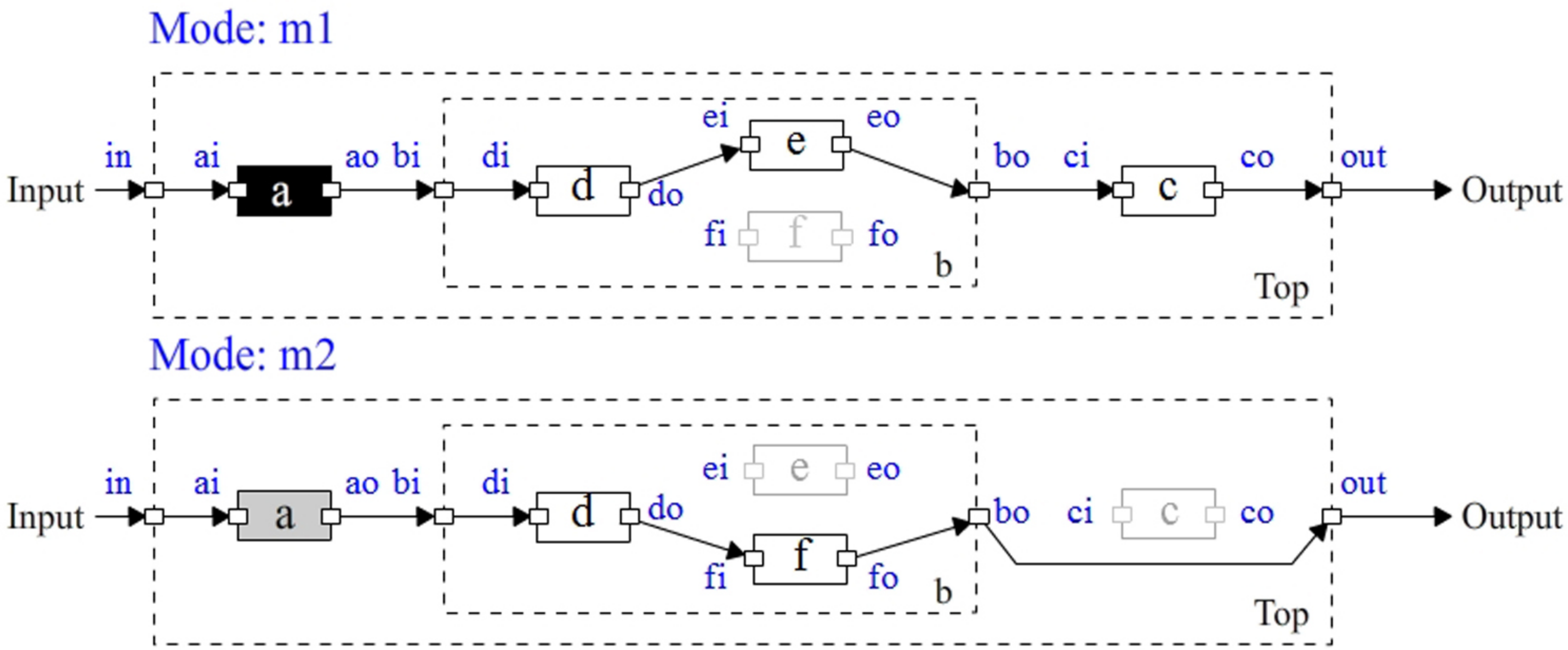
Human mode

Car mode

Introduction

- Component-based multi-mode system (CBMMS)





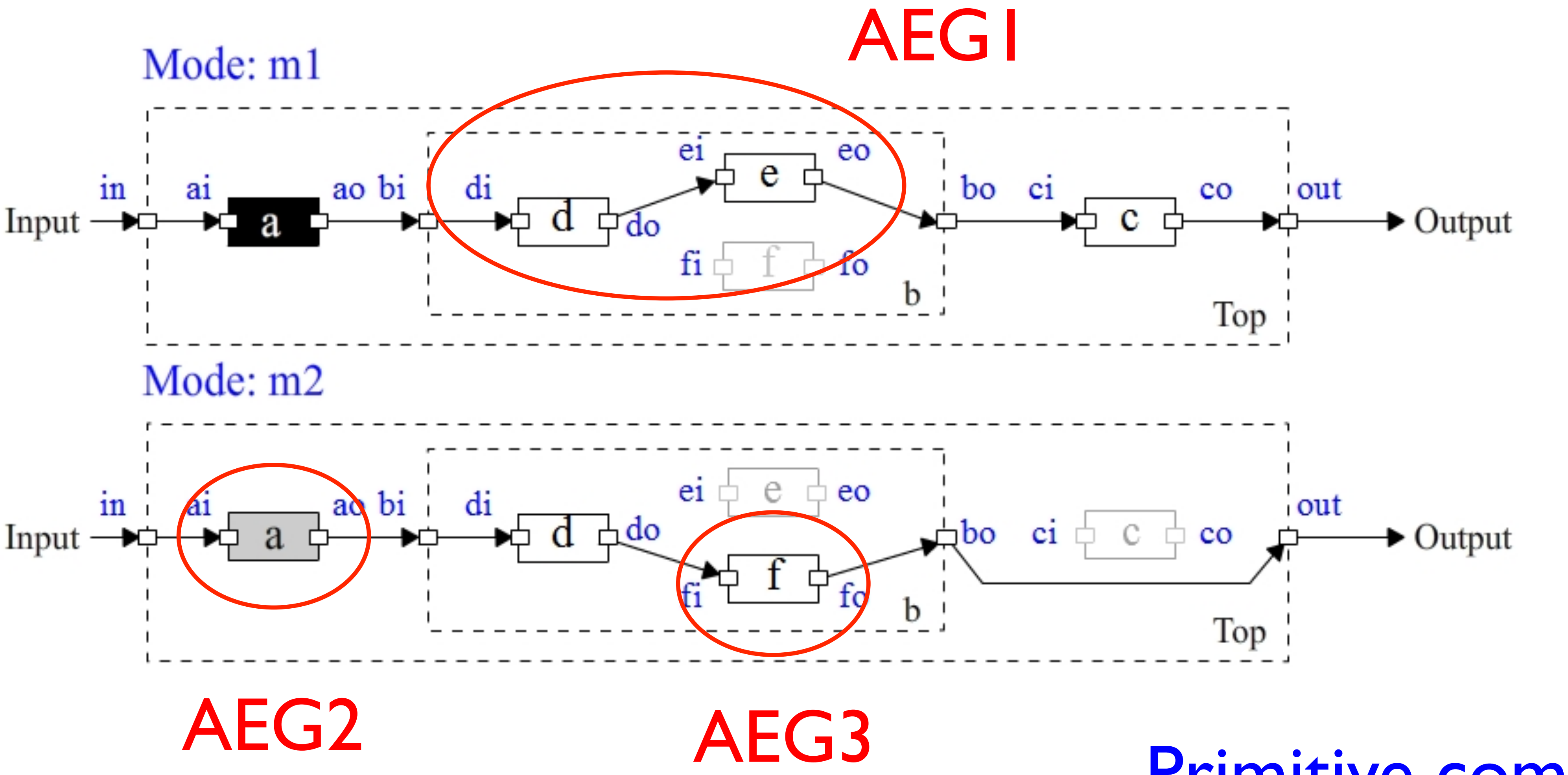
Primitive components: a, c, d, e, f

Composite components: Top, b

Composable mode switch? **Mode Switch Logic (MSL)**

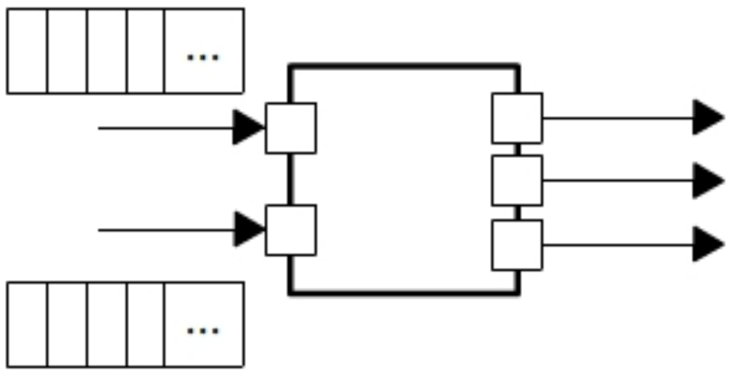
Introduction

- Atomic component execution--**Atomic Execution Group (AEG)**



Primitive component

Input buffers



Overview

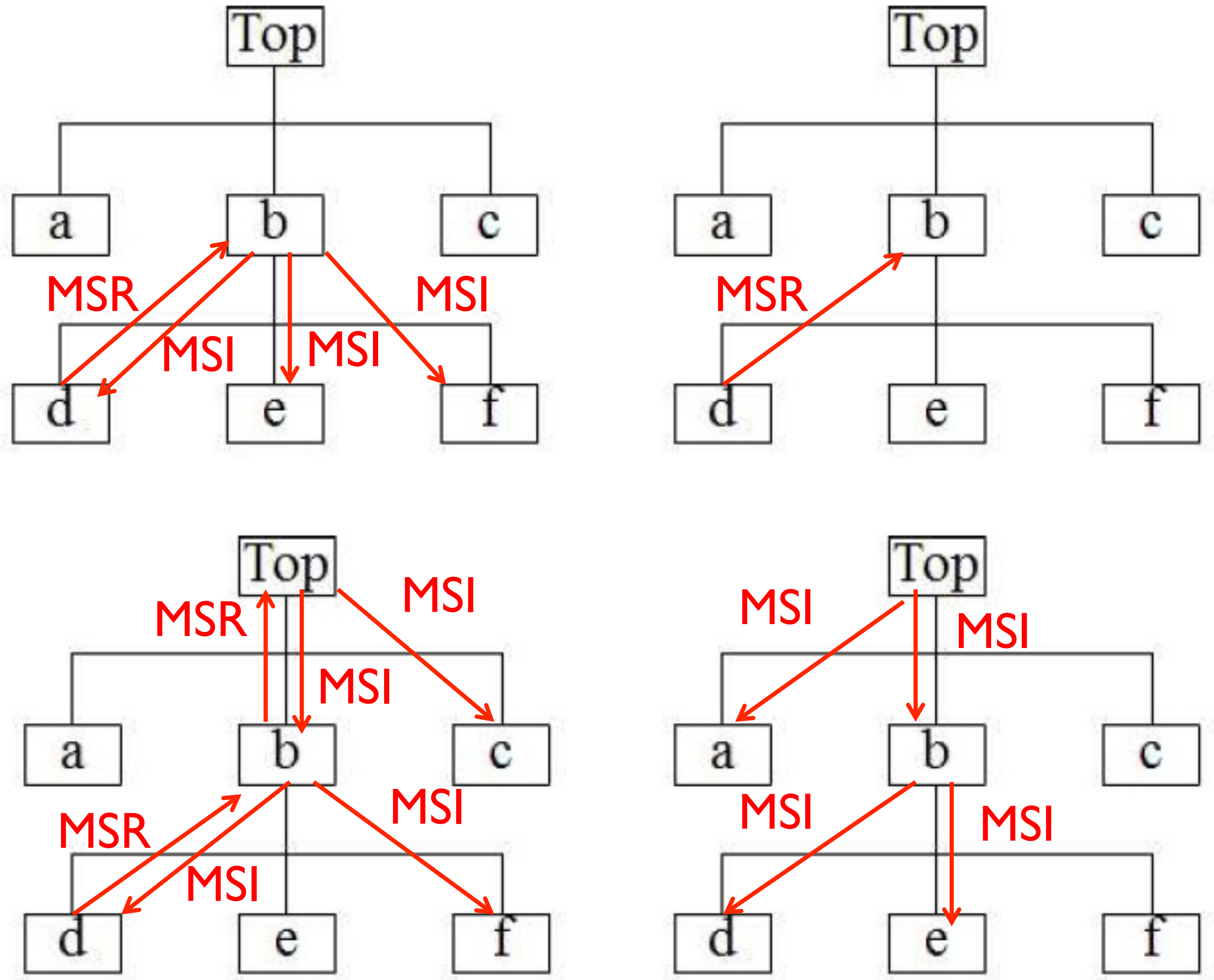
- Introduction
- **The Mode Switch Logic (MSL)**
- The handling of atomic component execution
- The mode switch timing analysis
- Calculating the worst-case atomic component execution time
- Conclusions and future work

The Mode Switch Logic (MSL)

- Mode-aware component model
- Mode mapping mechanism
- Mode Switch runtime mechanism

Mode Switch Logic (MSL)

- The **Mode Switch (MS) propagation mechanism**



MSR: Mode Switch Request

MSI: Mode Switch Instruction

MSS: Mode Switch Source

MSDM: Mode Switch Decision Maker

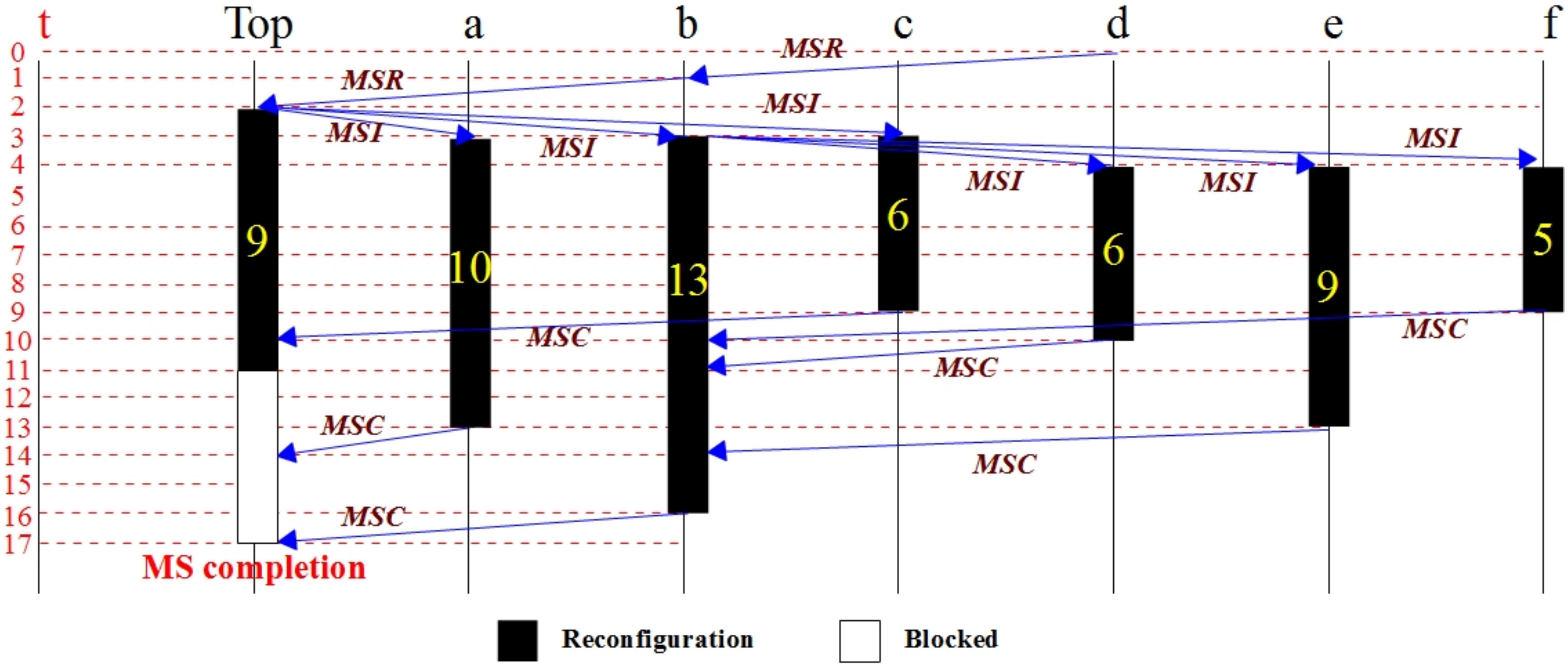
The mode switch runtime mechanism

- Guaranteeing mode consistency—**Mode switch dependency rule**
  - A component starts mode switch after receiving or actively issuing an MSI.
  - As a component starts its mode switch, if its target mode is different from its current mode, it will do the reconfiguration.
  - A component who has received an MSI from its parent must send an **MSC (Mode Switch Completion)** back after it completes its mode switch
  - Conditions for mode switch completion:
    - Primitive component: after reconfiguration
    - Composite component: (after reconfiguration)+the collection of all expected MSC from the subcomponents

The mode switch completion of a system=the mode switch completion of the MSDM

The mode switch runtime mechanism

- The complete mode switch process

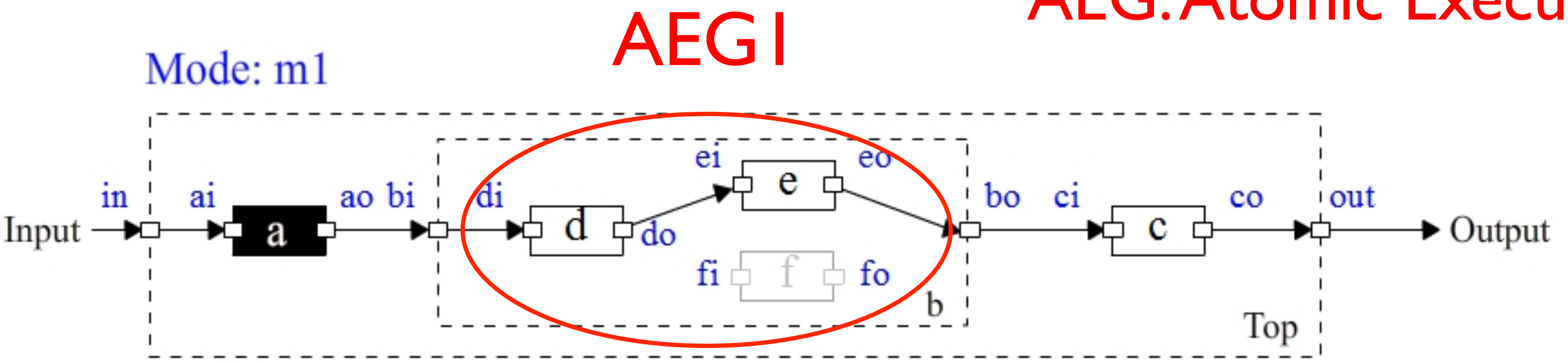


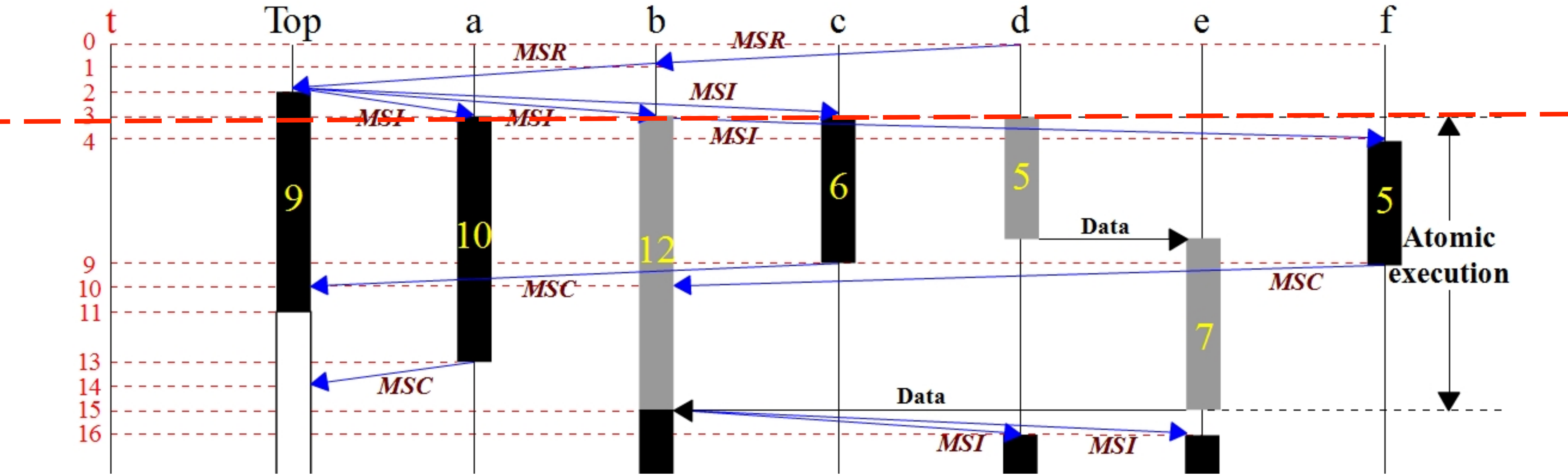
Overview

- Introduction
- The Mode Switch Logic (MSL)
- **The handling of atomic component execution**
- The mode switch timing analysis
- Deriving the worst-case atomic component execution time
- Conclusions and future work

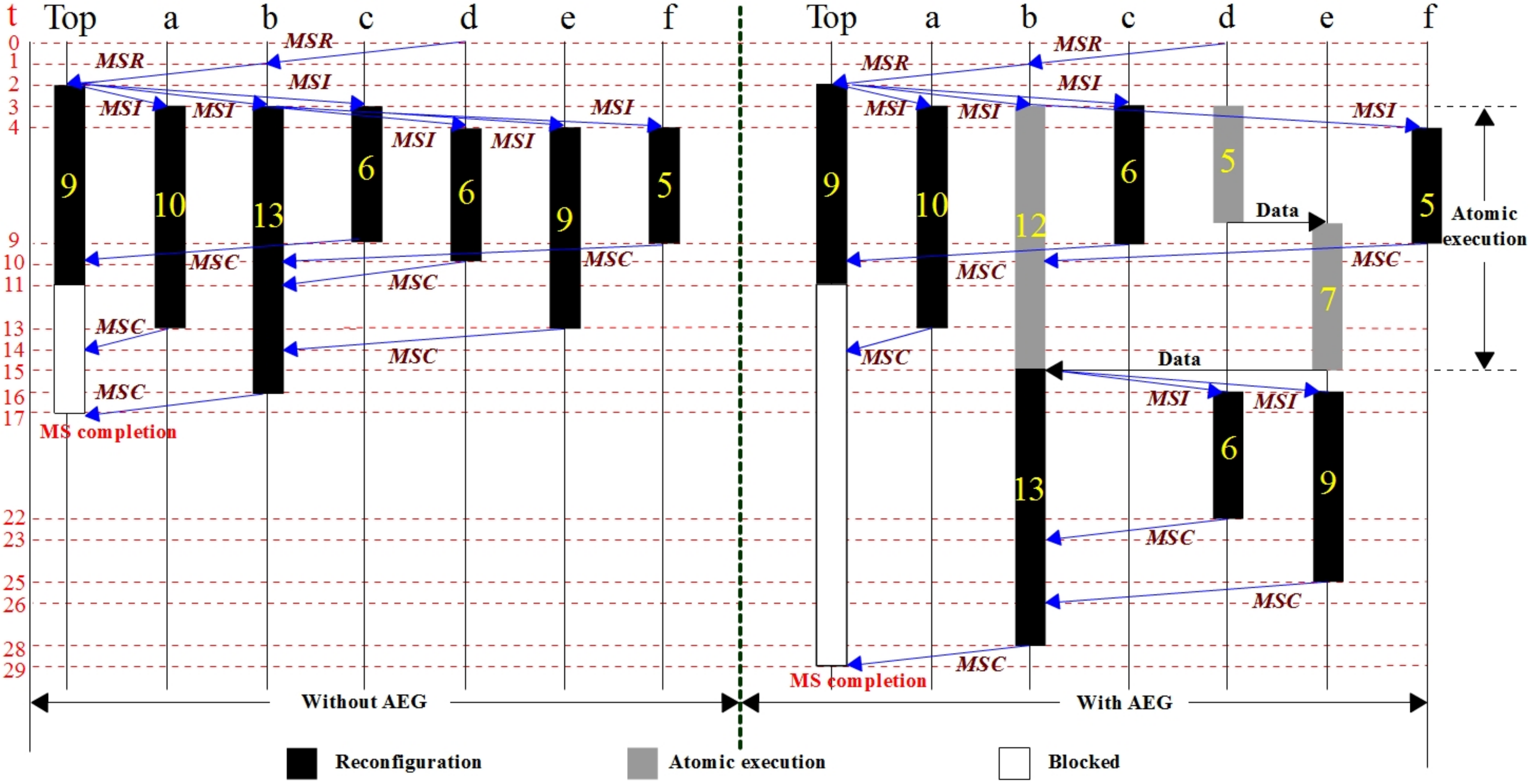
The handling of atomic component execution

AEG: Atomic Execution Group





The handling of atomic component execution



Overview

- Introduction
- The Mode Switch Logic (MSL)
- The handling of atomic component execution
- **The mode switch timing analysis**
- Deriving the worst-case atomic component execution time
- Conclusions and future work

The mode switch timing analysis without AEG

- Notations
  - $\mathbf{t_{MSR}}$, $\mathbf{t_{MSI}}$, $\mathbf{t_{MSC}}$: The transmission time of an MSR, MSI or MSC
  - $\mathbf{RCT_{ci}}$: The reconfiguration time of c_i
  - $\mathbf{MS_{ci}}$: The mode switch time of c_i
- Two phases
  - $\mathbf{T_{MSR}}$: MSR propagation (upstream)
  - $\mathbf{T_{MSI}}$: MSI propagation (downstream) and mode switch

The mode switch timing analysis without AEG

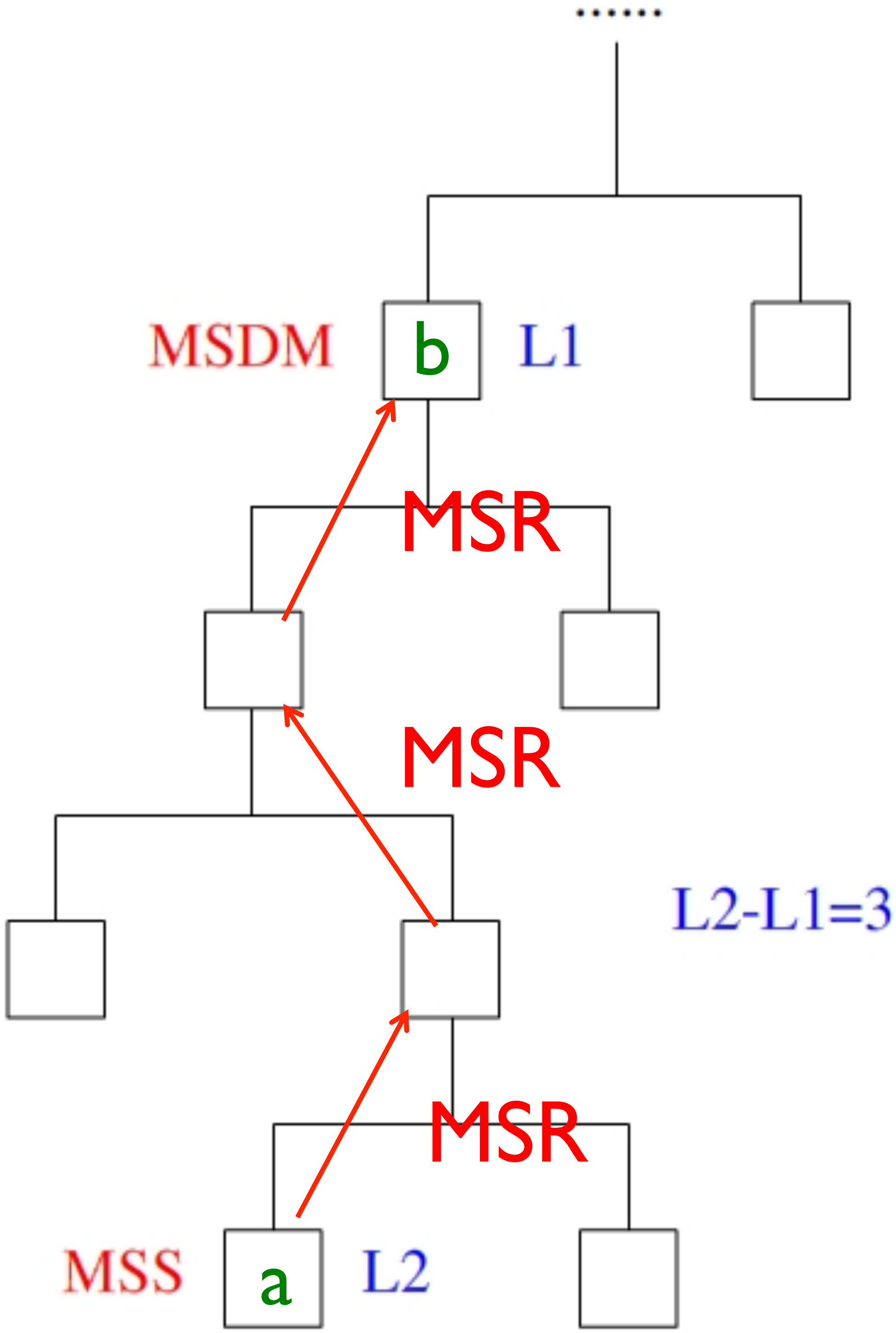
- The first phase--T_{MSR}
  - Depth level: L
  - MSR propagation time

$$T_{MSR} = t_{MSR} * \Delta L$$

$$\Delta L = L_{MSS} - L_{MSDM}$$

MSS: Mode Switch Source

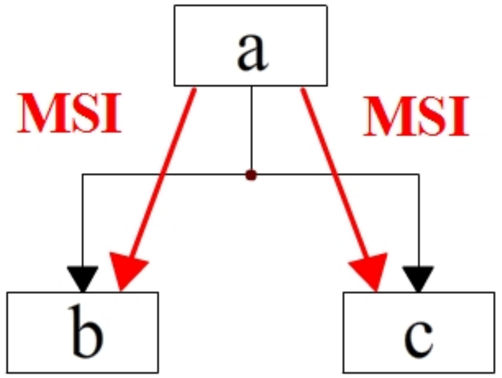
MSDM: Mode Switch Decision Maker



The mode switch timing analysis without AEG

- The second phase--T$_{MSI}$

$T_{MSI}=MS_{MSDM}$



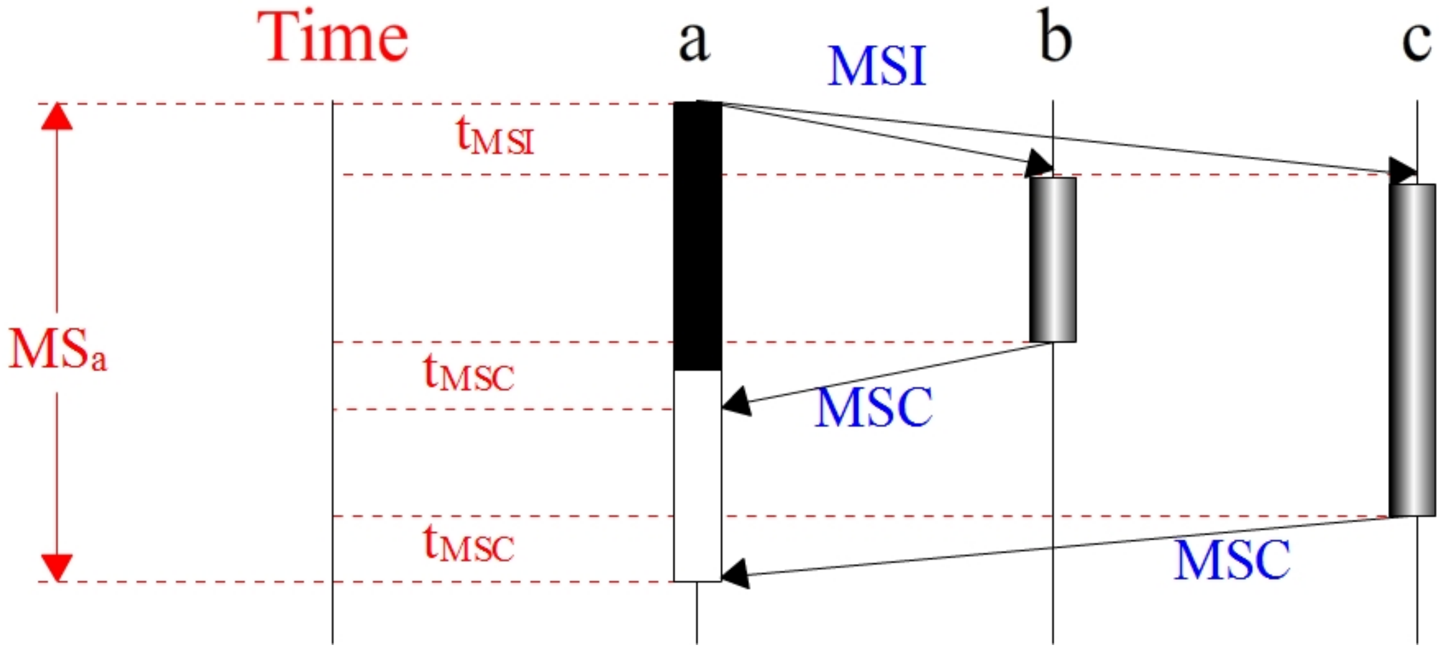
■ Reconfiguration

▨ Mode switch

□ Blocked

Total mode switch time:

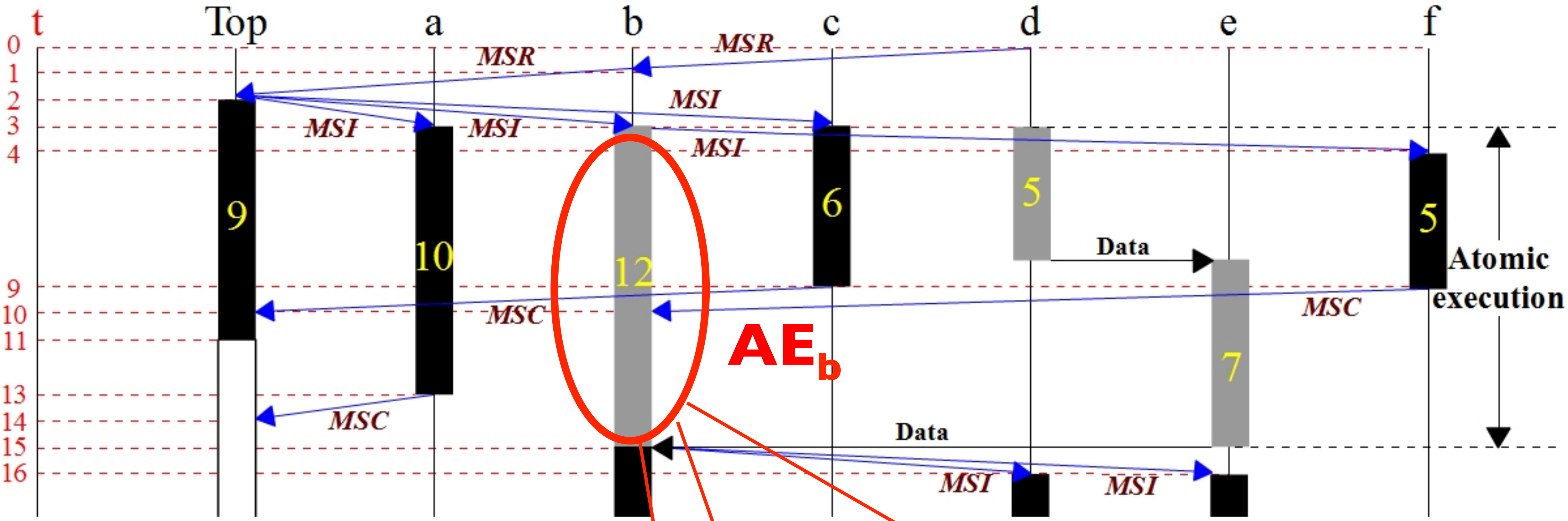
$MS=T_{MSR}+T_{MSI}$



$$MS_a = \max\{RCT_a, t_{MSI} + MS_b + t_{MSC}, t_{MSI} + MS_c + t_{MSC}\}$$

For a primitive component x: $MS_x = RCT_x$

The mode switch timing analysis with AEG



- A constant delay, the worst-case execution time of an AEG (**AE**), is added to the reconfiguration starting time of the AEG component and its activated subcomponents

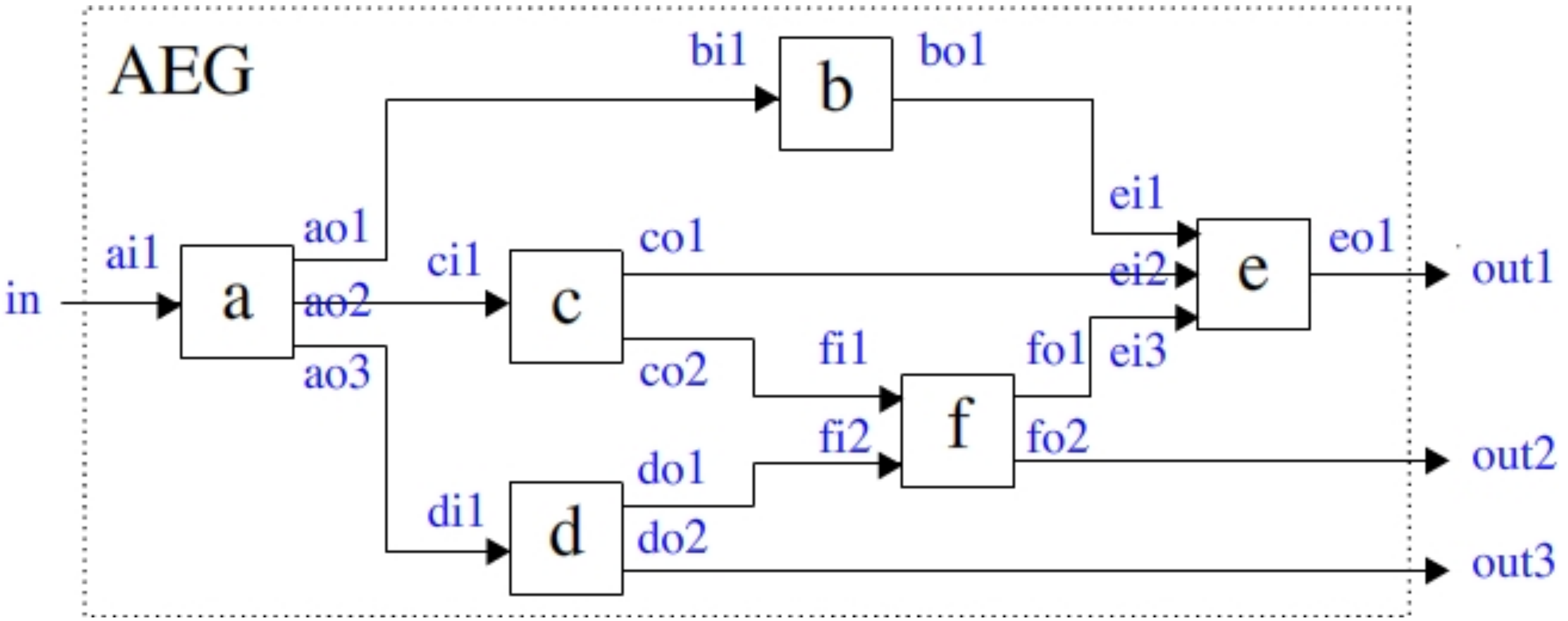
$$MS_b = \max\{RCT_b + AE_b, t_{MSI} + MS_d + t_{MSC} + AE_b, \\ t_{MSI} + MS_e + t_{MSC} + AE_b, t_{MSI} + MS_f + t_{MSC}\}$$

Overview

- Introduction
- The Mode Switch Logic (MSL)
- The handling of atomic component execution
- The mode switch timing analysis
- **Calculating the worst-case atomic component execution time**
- Conclusions and future work

Calculating the worst-case atomic component execution time—AE

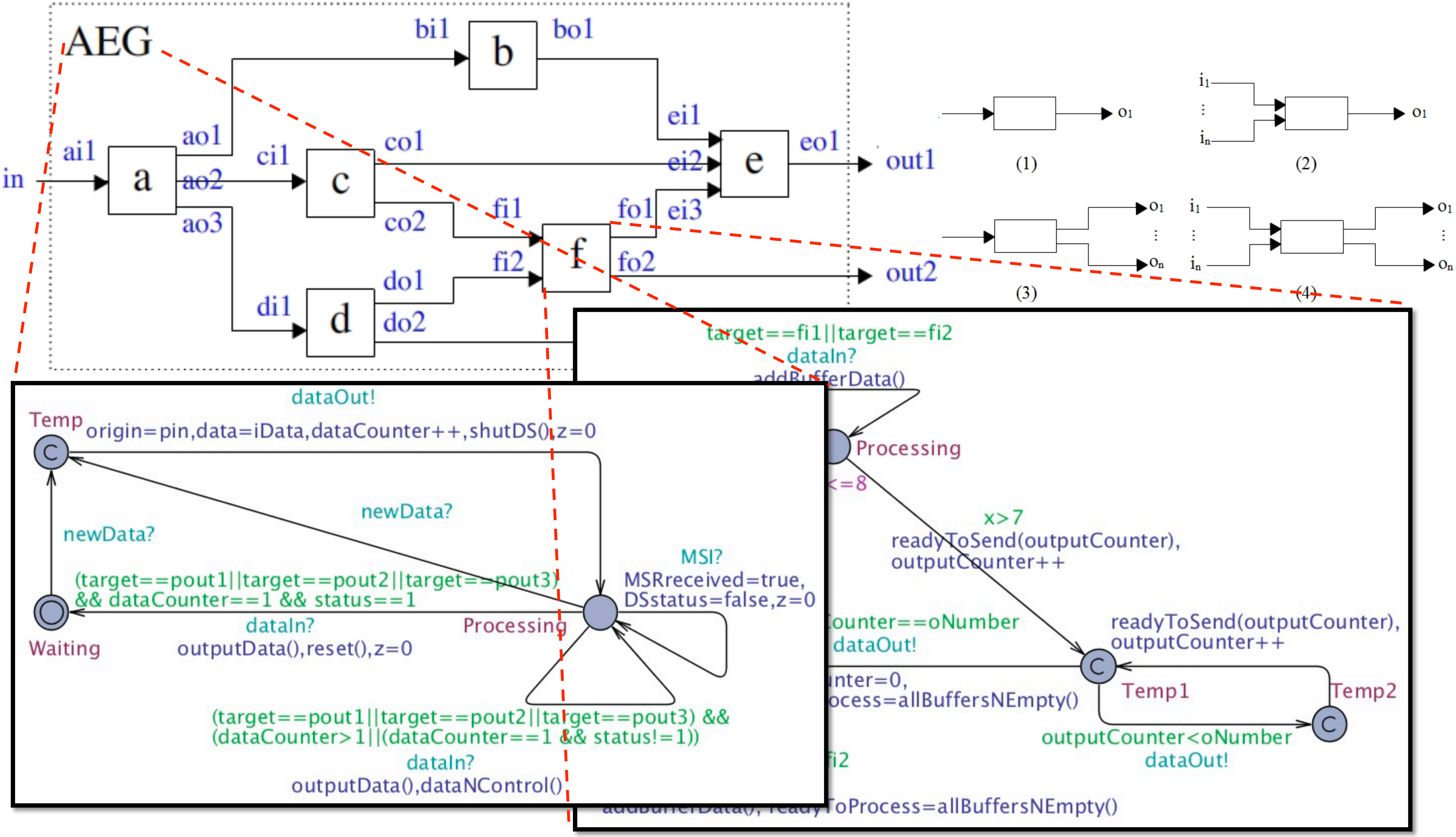
Input data



Output data

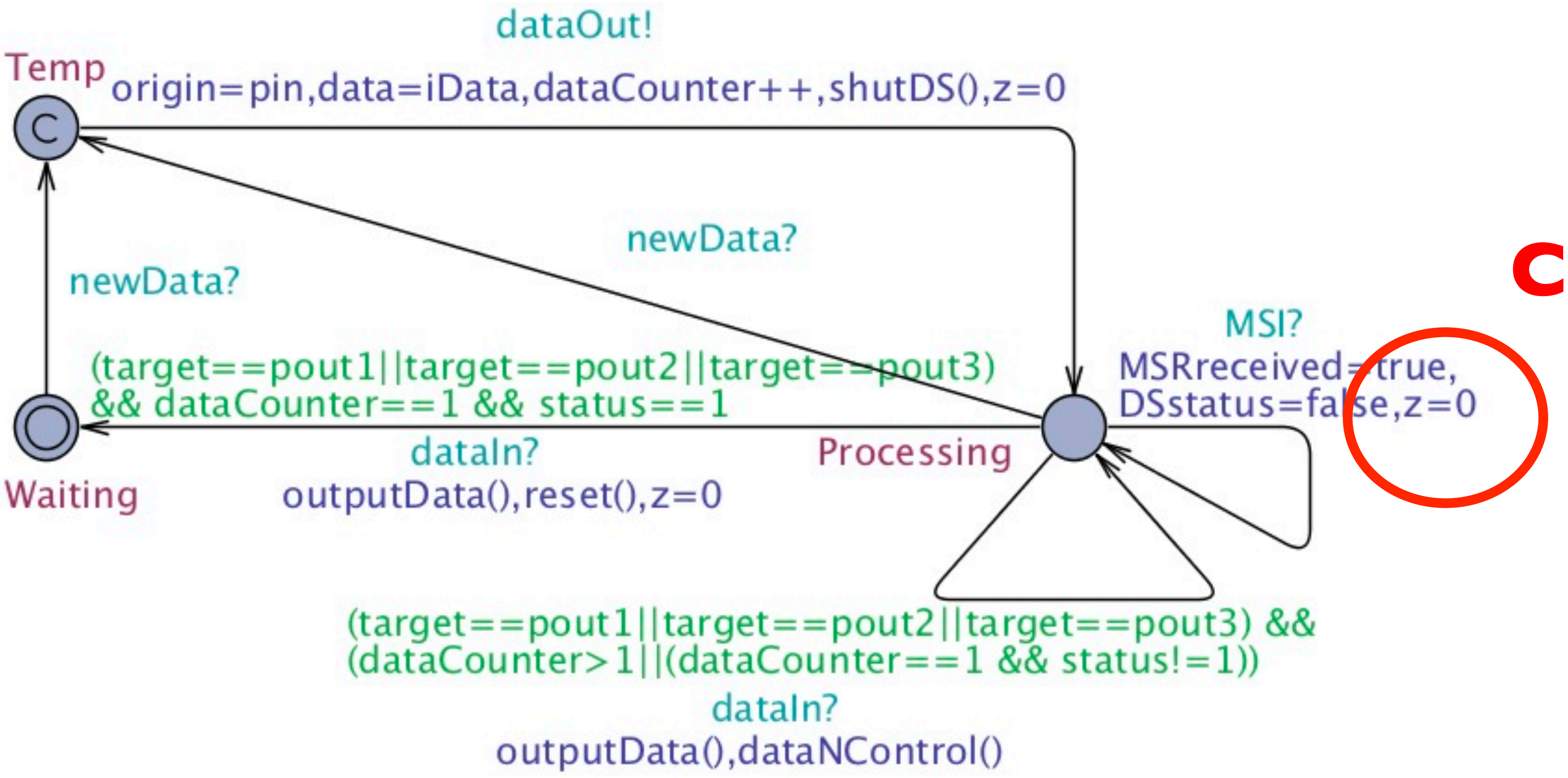
- A model-checking approach to deriving **AE**--UPPAAL
- Parameters:
  - Input data rate: R
  - The data processing time of each primitive component: C
  - Maximum of data elements in the AEG: N

Calculating the worst-case atomic component execution time—AE



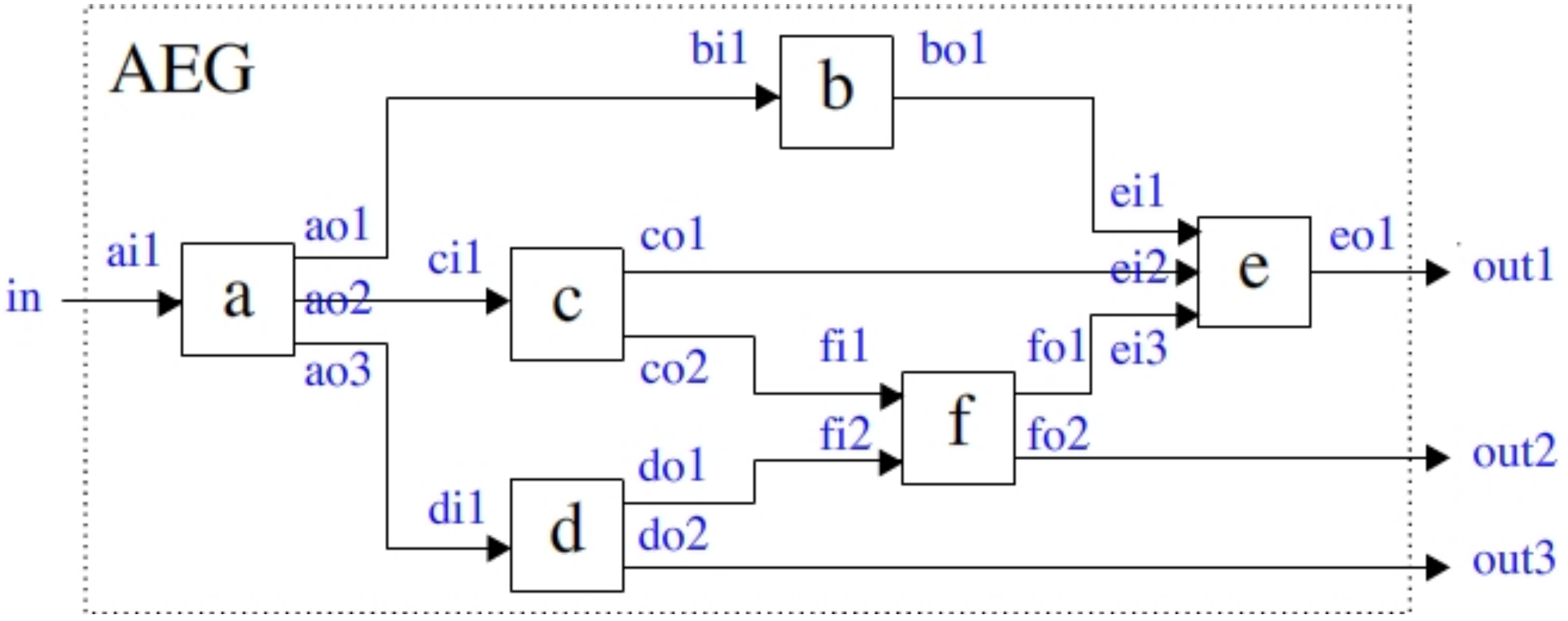
Calculating the worst-case atomic component execution time—AE

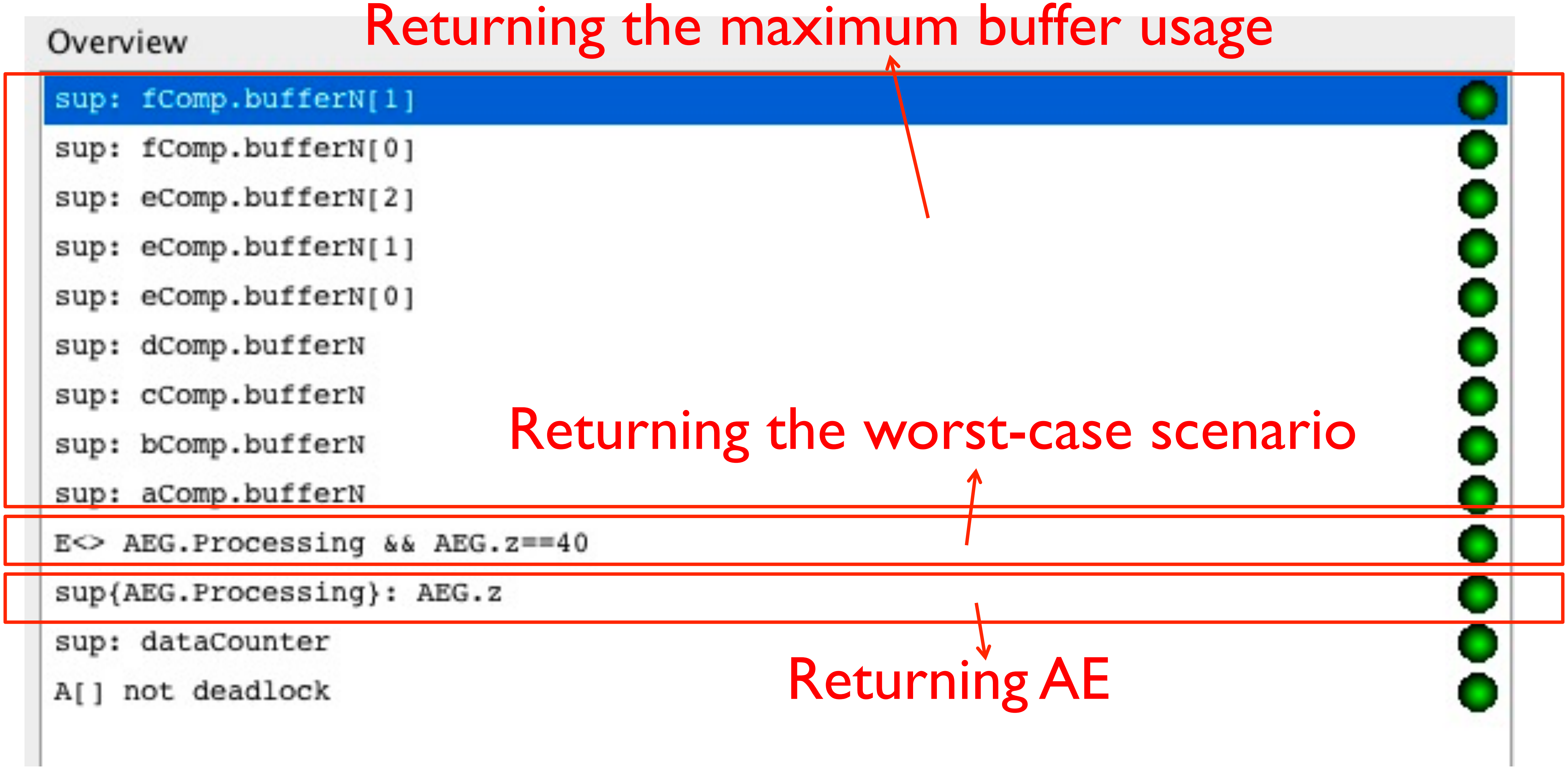
- The "sup" operator: automatically returning the maximum value of a variable or clock


dataOut!
Temp
origin=pin,data=iData,dataCounter++,shutDS(),z=0
C
newData?
newData?
MSI?
MSReceived=true,
DSstatus=false,z=0
(target==pout1||target==pout2||target==pout3)
&& dataCounter==1 && status==1
dataIn?
outputData(),reset(),z=0
Waiting
Processing
(target==pout1||target==pout2||target==pout3) &&
(dataCounter>1||(dataCounter==1 && status!=1))
dataIn?
outputData(),dataNControl()
Clock z

AE: the maximum value of Clock z

Verification





Overview

```
sup: fComp.bufferN[1]
sup: fComp.bufferN[0]
sup: eComp.bufferN[2]
sup: eComp.bufferN[1]
sup: eComp.bufferN[0]
sup: dComp.bufferN
sup: cComp.bufferN
sup: bComp.bufferN
sup: aComp.bufferN
E<> AEG.Processing && AEG.z==40
sup{AEG.Processing}: AEG.z
sup: dataCounter
A[] not deadlock
```

Returning the maximum buffer usage

Returning the worst-case scenario

Returning AE

Conclusions and future work

- Mode Switch Logic (MSL) for component-based multi-mode systems (CBMMSs)
- The handling of atomic component execution in the MSL
- The mode switch timing analysis
- Deriving the worst-case atomic component execution time by model checking
- **Future work**
  - Resolving the conflict of multiple mode switch triggering+the mode switch timing analysis
  - More general model for the Atomic Execution Group (AEG)
  - Component connections with feedback loops
  - ……

MÄLARDALEN UNIVERSITY
SWEDEN

Thank you!