Processor Pipelines and Static WCET Analysis

Jakob Engblom

& Bengt Jonsson

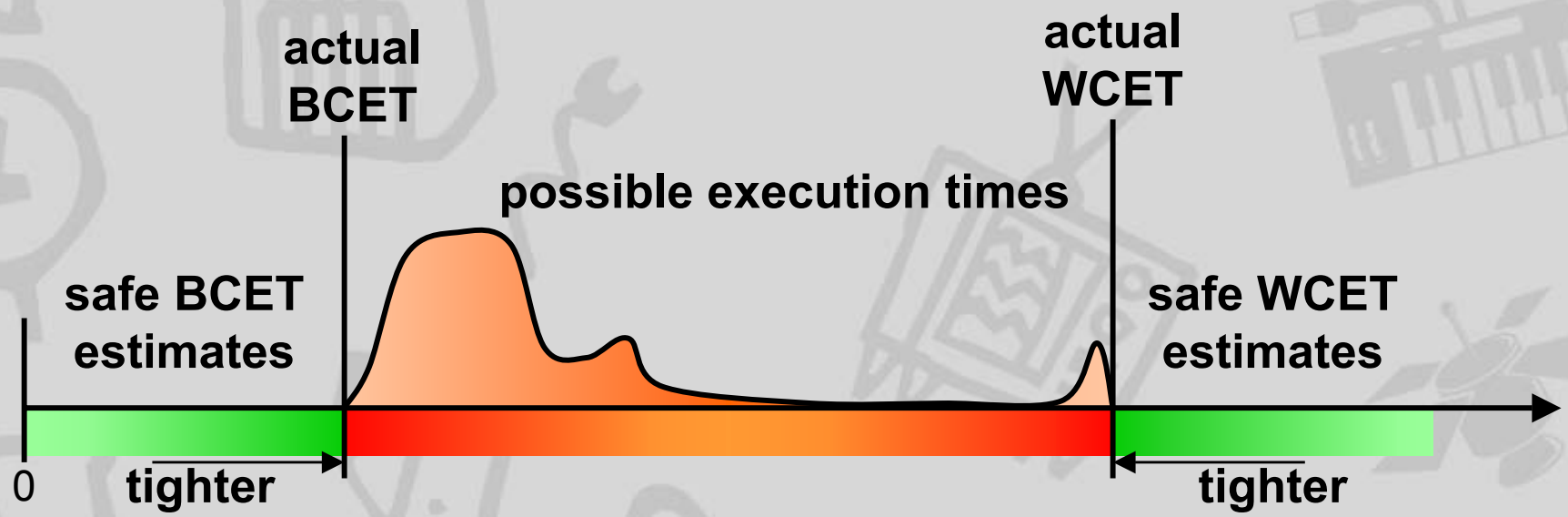
Uppsala University

jakob.engblom@it.uu.se

Currently at Virtutech, www.virtutech.com

And with IAR Systems, www.iar.com

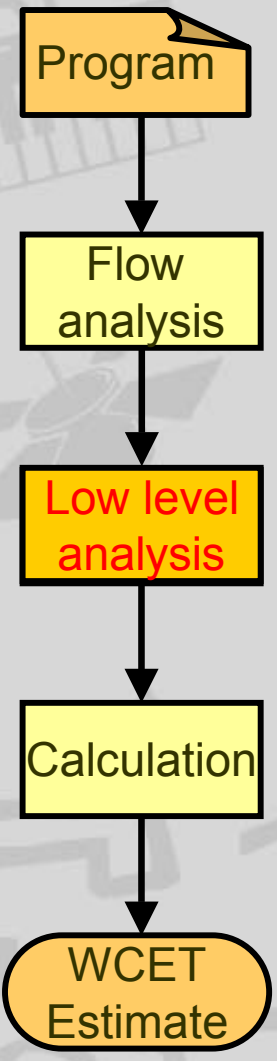
What is "WCET"?



- **WCET** = Worst case
- **BCET** = Best case
- **ACET** = Average case

Static WCET Analysis

- Find the WCET of a program
- Using of… alysis
- Need to … yze:
  - Program…
  - Cache b…
  - Pipeline…
- … and th… n



Previous Work

- Pipeline analysis by pairs of blocks
  - Kim et al. 1995
  - Ottoson & Sjödin 1997
  - Altenbernd & Stappert 1999
  - Bate et al. 2000
  - Colin & Puaut 2001
  - Atanssov et al. 2001
  - Lundqvist 2002
- Pipeline analysis along complete paths
  - Healy et al. 1999
  - Ziegenbein et al. 2001

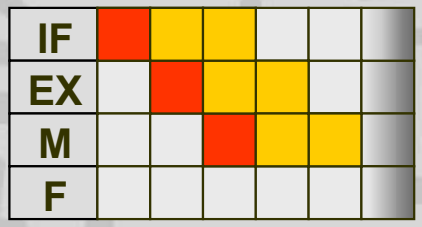
Previous Work

- By short sequences
  - Engblom & Ermedahl, 1999
  - Stappert et al., 2001
- By complete state exploration
  - Lim et al., 1998
  - Ferdinand et al., 2001
- ::All concrete methods to solve concrete analysis problems

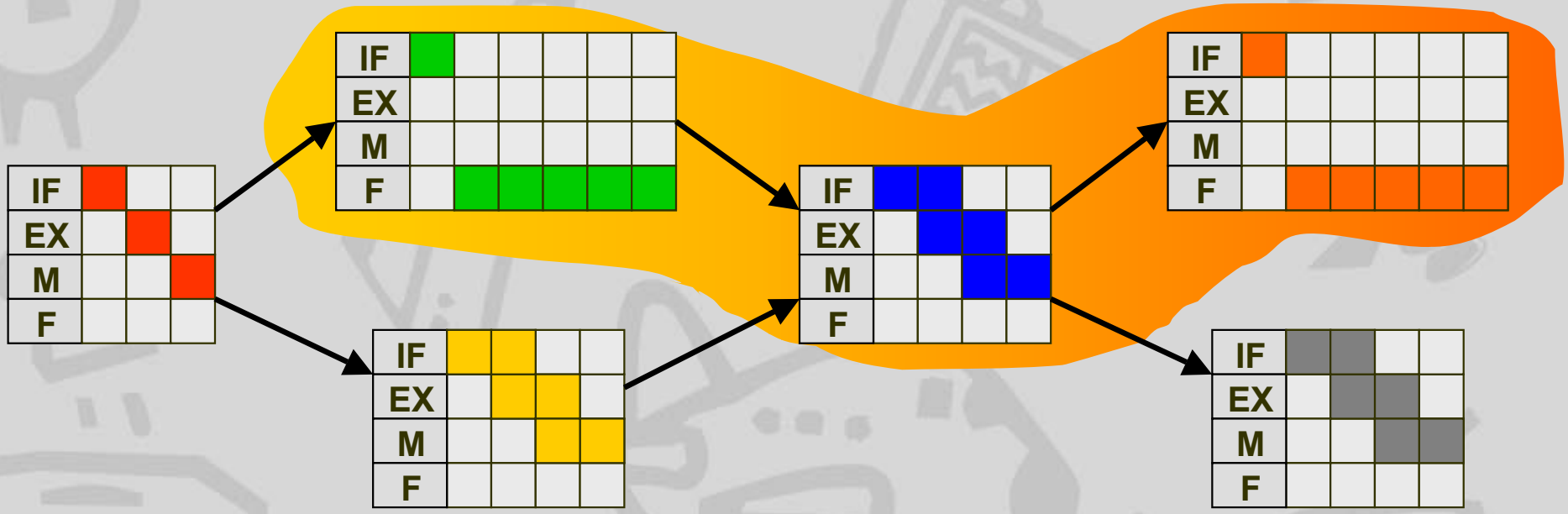
The Question

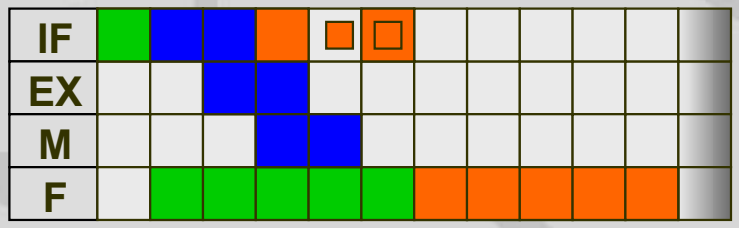
- How large pieces of a program have to be analyzed to find a perfect picture of pipeline behavior?

Pipeline Interactions

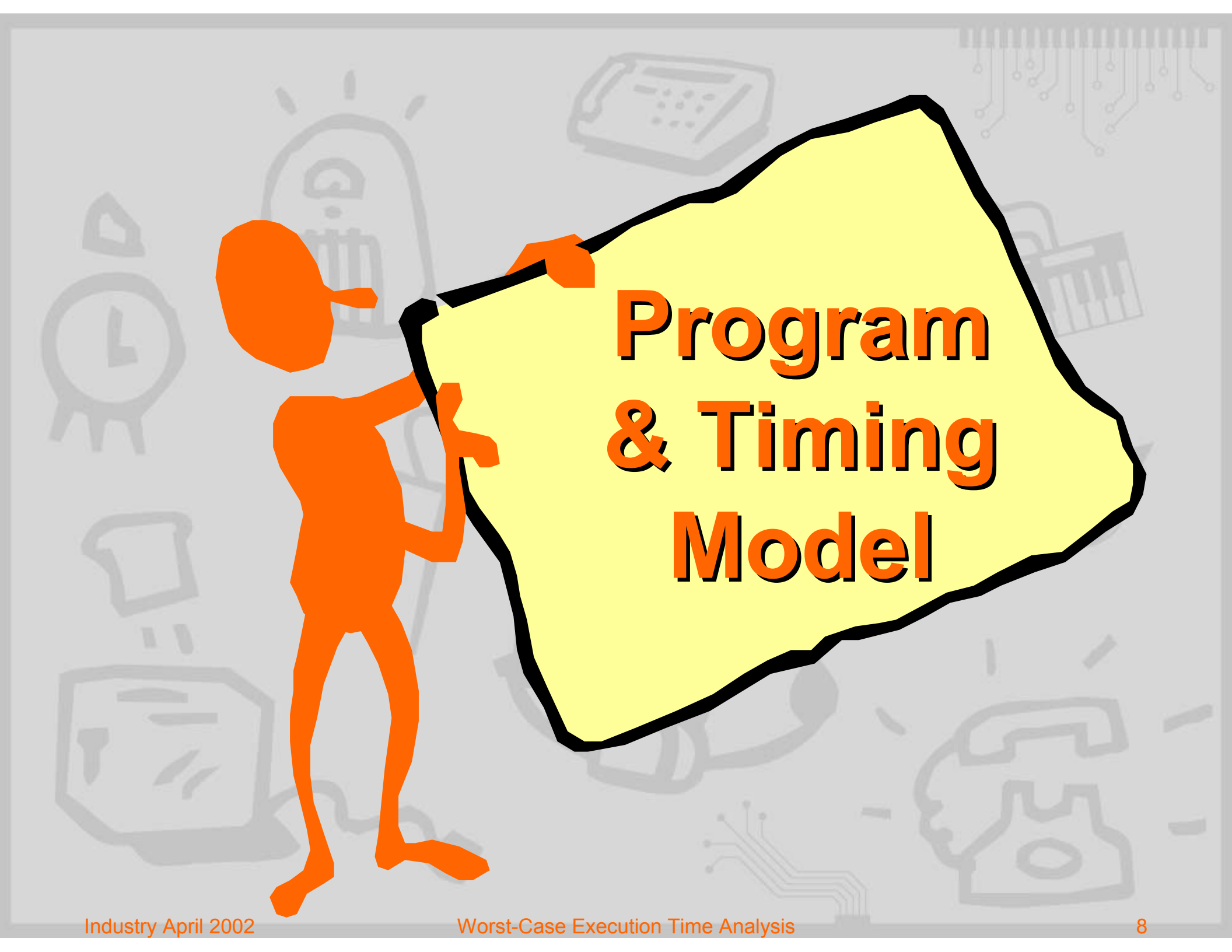


Pairwise overlap: speed-up that we want to account for



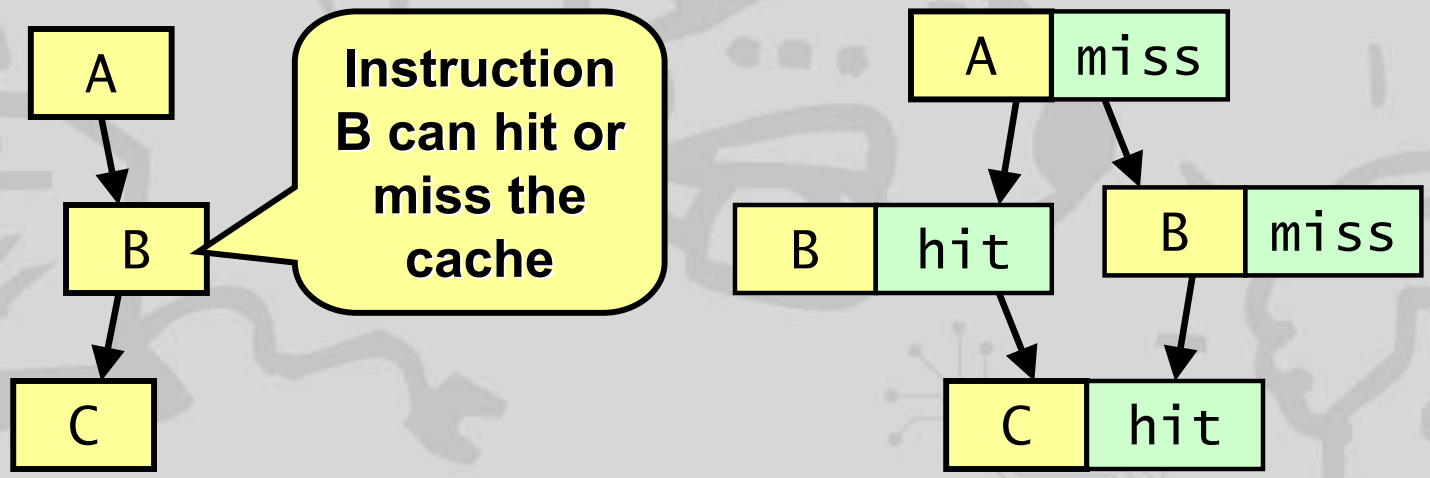


Interaction across three blocks!


Program
& Timing
Model

Program Graph

- **Deterministic assumption:**
  - Each node has precisely one time
  - Execution facts fix all variables
  - Use multiple nodes to model variability
- **Example:**



Program Timing Model

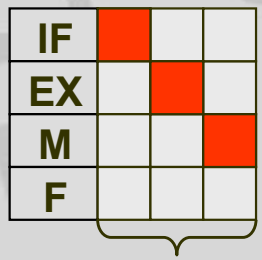
- Times (t_A)
  - For single nodes
- Timing effects (δ_{ABC})
  - For sequences of nodes
  - Pairwise or long (>2 nodes)
  - Effect of differing previous nodes
- Execution time: $T(A..Z)=\Sigma\, t + \Sigma\, \delta$
  - All times & all subsequences for A..Z

Program Timing Model

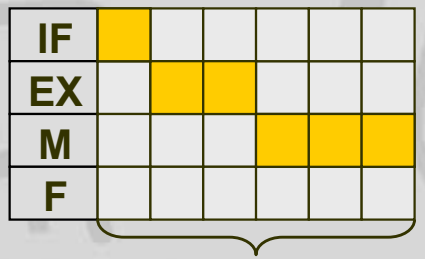
- Deriving timing effects:
  - For sequence
  - $\delta_{A..Z}$=T(AB..YZ)-T(AB..Y)-T(B..YZ)+T(B..Y)
  - Negative for speedup
  - Positive for slowdown
  - δ_{AB}=T(AB)-T(A)-T(B): Pairwise effects
- $\delta_{A..Z} \neq 0$ means that A affects Z

Pairwise Timing Effects

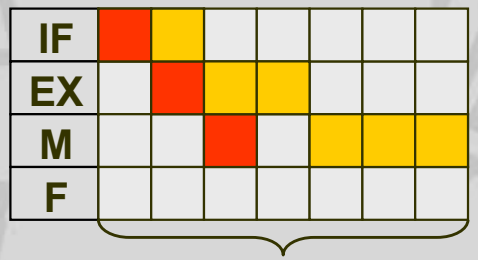
- Effects between two instr:



T(A)=3

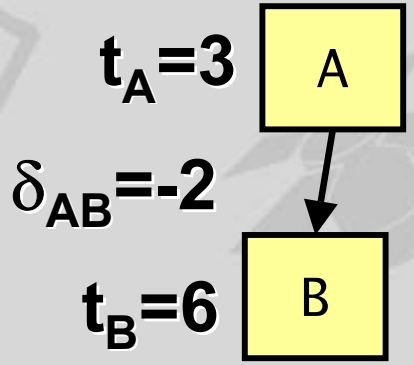


T(B)=6



T(AB)=7

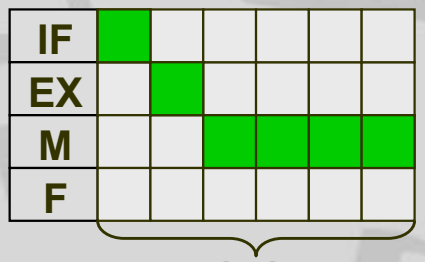
δ_{AB}=7-6-3+0=-2



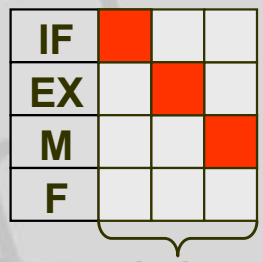
- "Always" appears in pipelines
- Always negative or zero

Long Timing Effect (LTE)

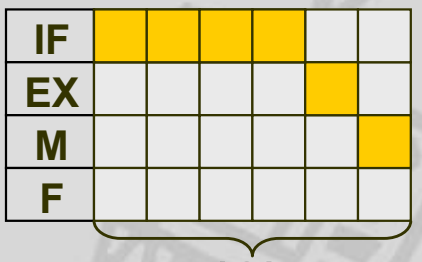
- In a single pipeline (already seen ||)



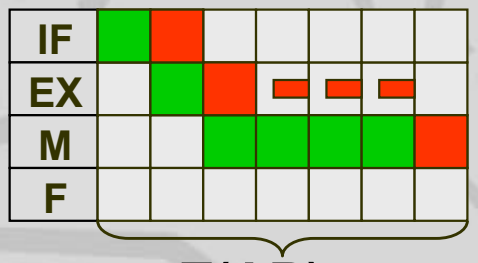
T(A)=6



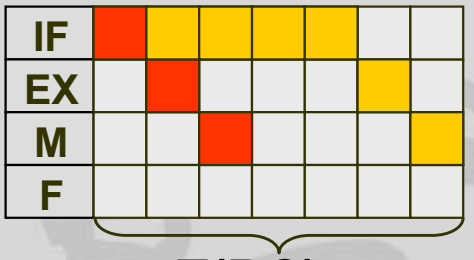
T(B)=3



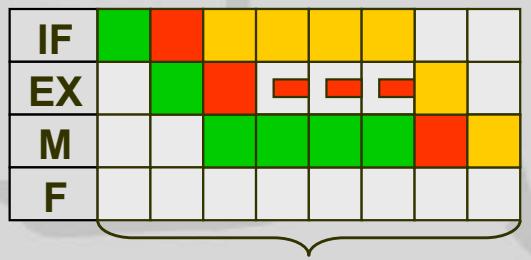
T(C)=6



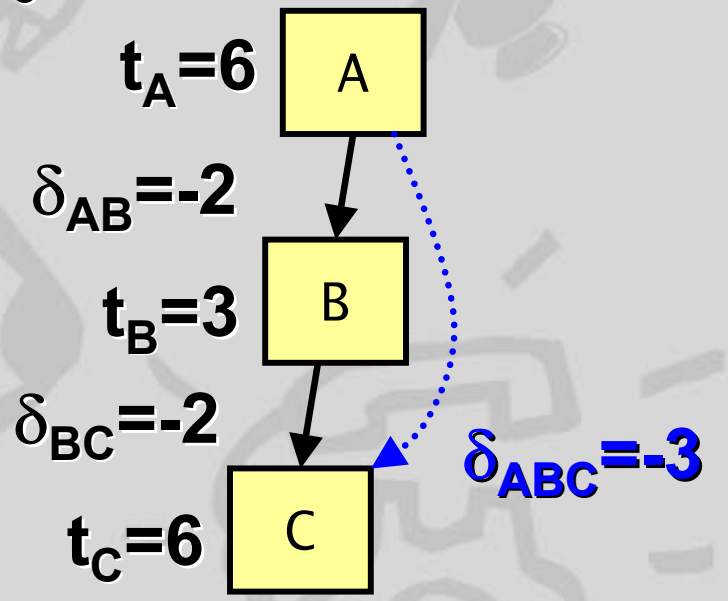
T(AB)=7



T(BC)=7



T(ABC)=8

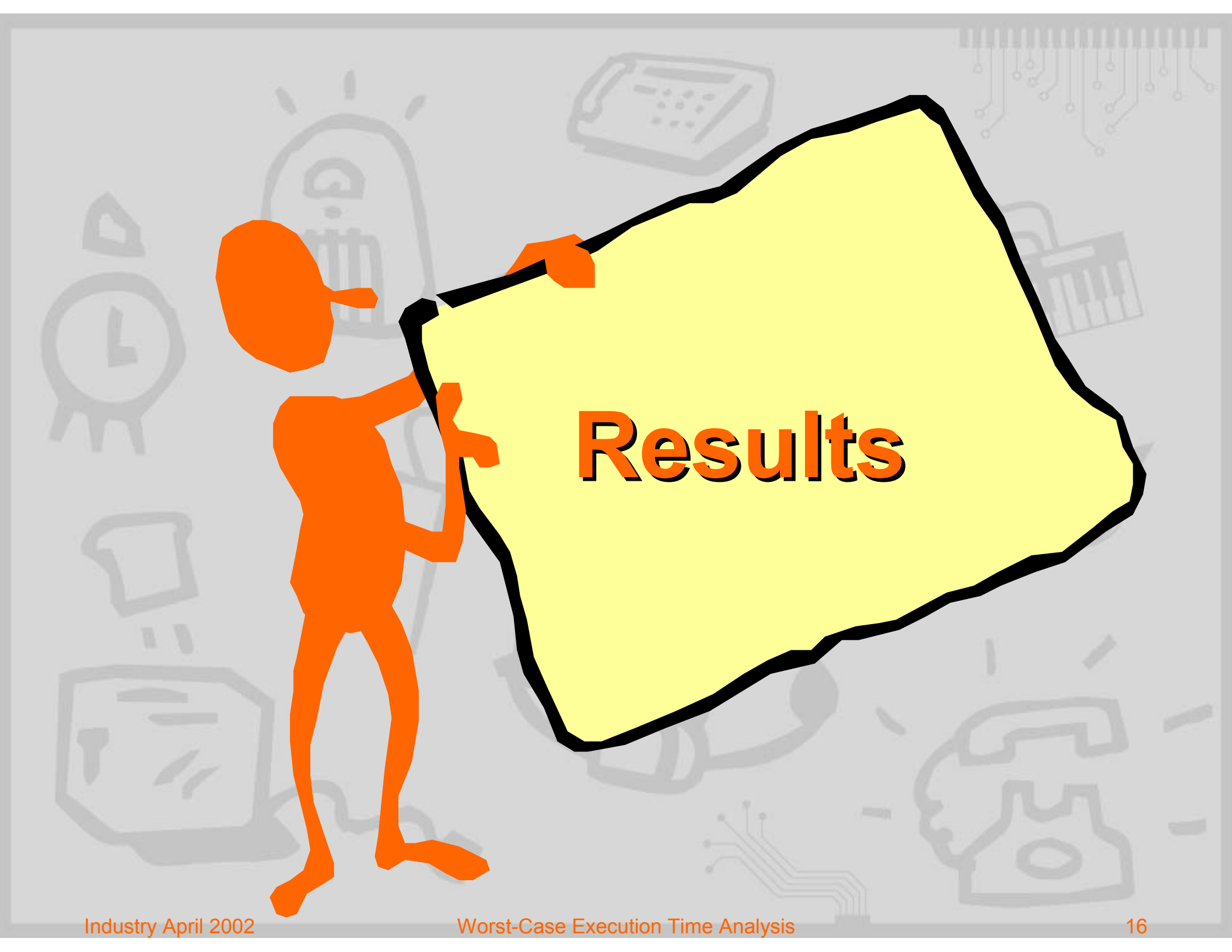


Long Timing Effect (LTE)

- Property of the pipeline, not of our particular model
- Negative ($\delta_{A..Z} < 0$):
  - Can be safely ignored
- Positive ($\delta_{A..Z} > 0$):
  - Have to be considered for safety
  - The main problem in pipeline modeling for static WCET analysis

Pipeline Behavior Model

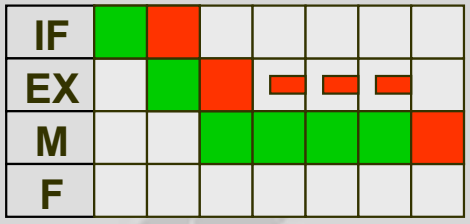
- Question: when do LTEs occur?
- To answer this, we built a constraint model of pipelines
  - Sufficient for in-order pipelines
  - Single-issue (or VLIW) required
- Details in the paper

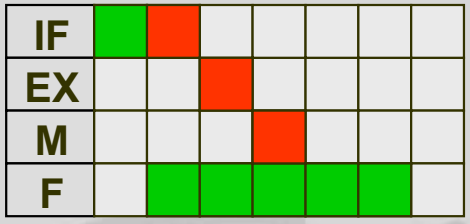

Results

Source of LTEs

- We have proven that LTEs over I_1 .. I_m require that:
  - The first instruction (I_1) stalls one of its successors

    OR
  - The first instruction (I_1) reaches past I_2 .. I_{m-1}





- NB: not sufficient condition

Source of LTEs

- One stall can give several LTEs
- A single stall can give
  - Positive, and
  - Negative timing effects
- A stall need not give LTEs
  - Depends on exact timing of instrs
  - Many such cases on the V850E

Only Negative LTE

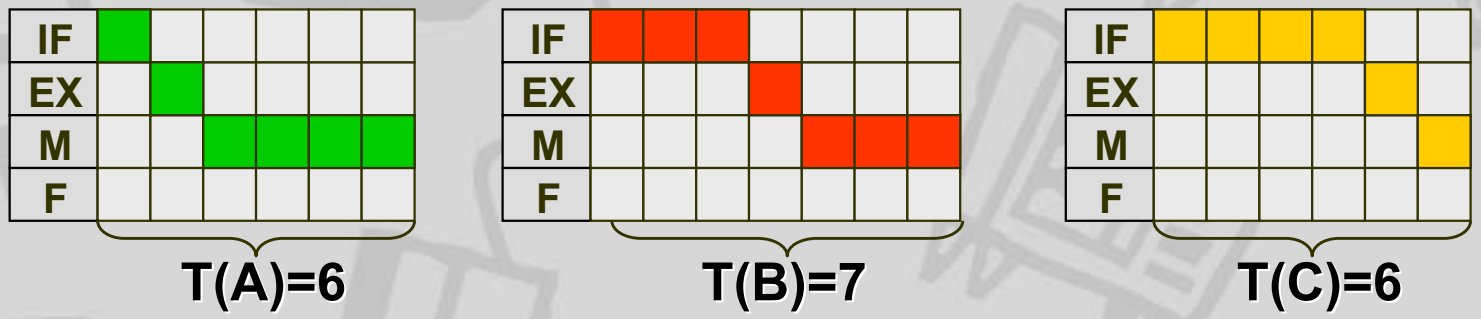
- Prove absence of positive δ
- Only negative LTEs occur when:
  - Single pipeline
  - All data dependences go between adjacent instructions
- Multiple pipelines means +
- Non-adjacent data dep means +

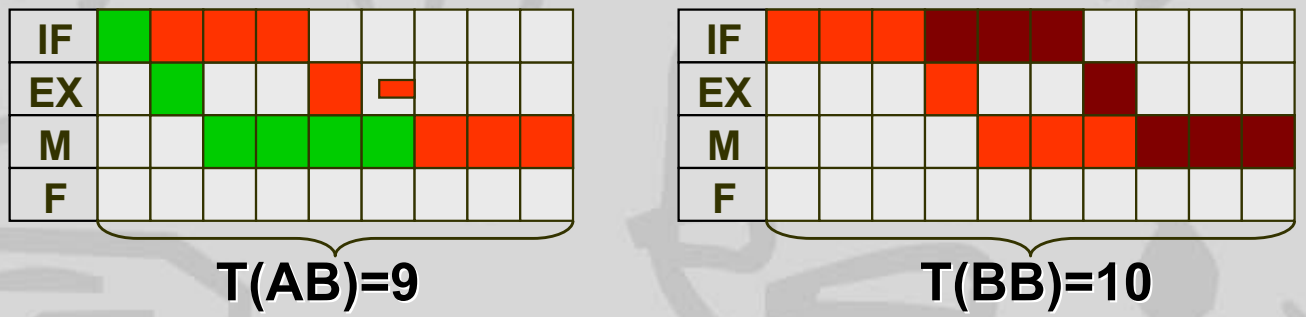
Only Negative LTEs

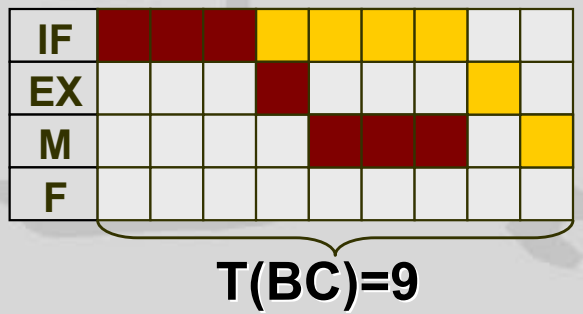
- Consequences:
  - Some lucky processors have no LTEs
- Examples:
  - NEC V850
  - ARM7, ARM9
  - Most five-stage RISC pipelines
- Allows safe approximate analysis
  - Analyze sequences up to some length l
  - Can be very tight & efficient

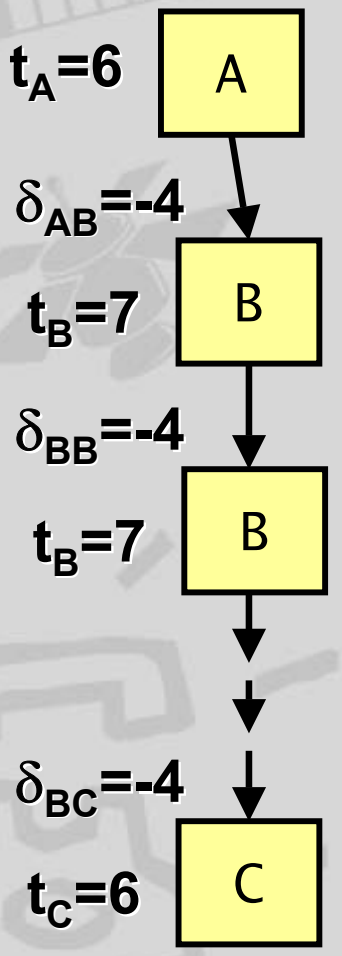
Unbounded LTE

- In a single pipeline



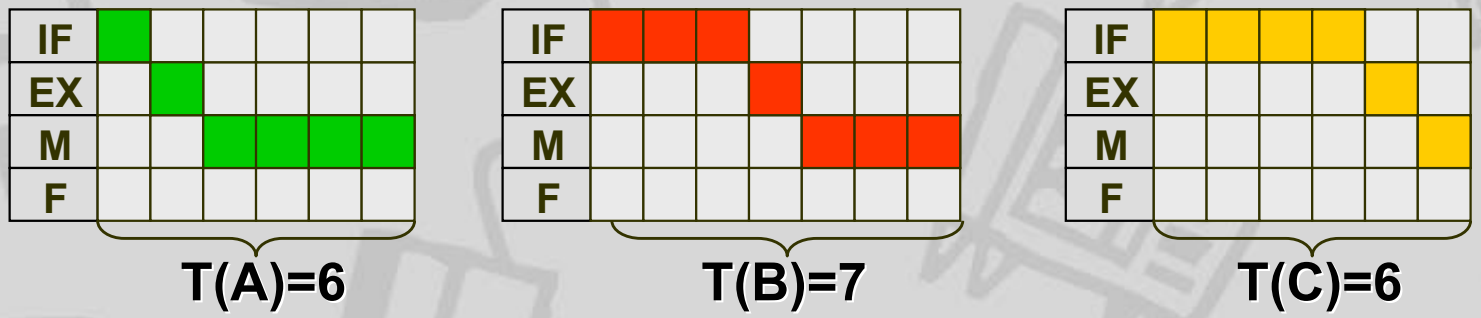


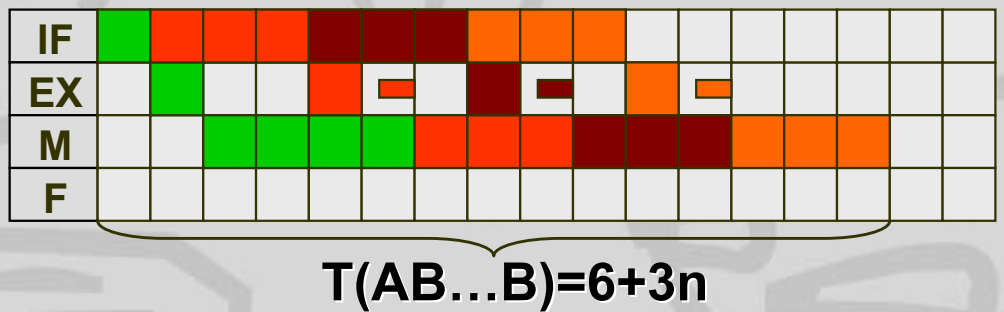


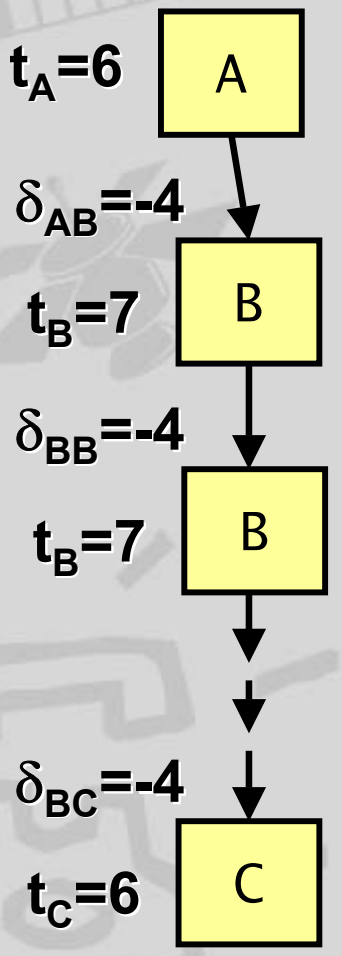


Unbounded LTE

- In a single pipeline

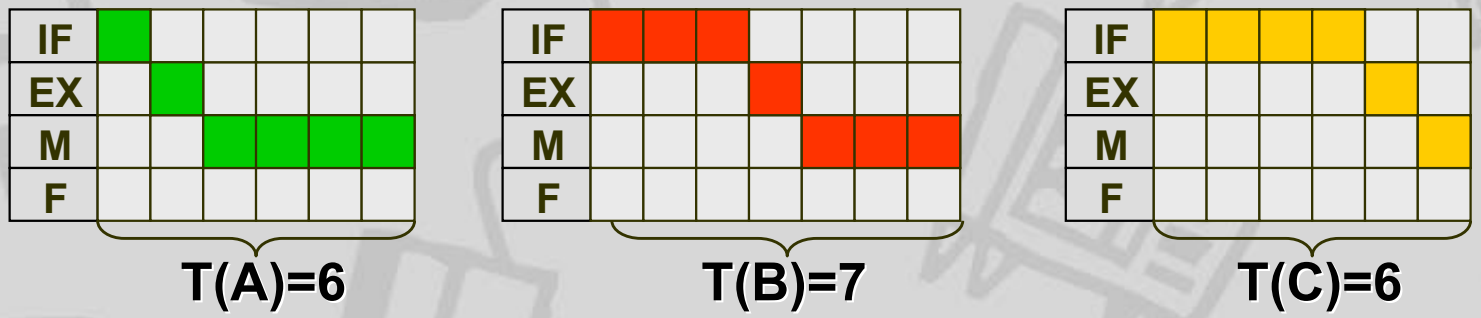


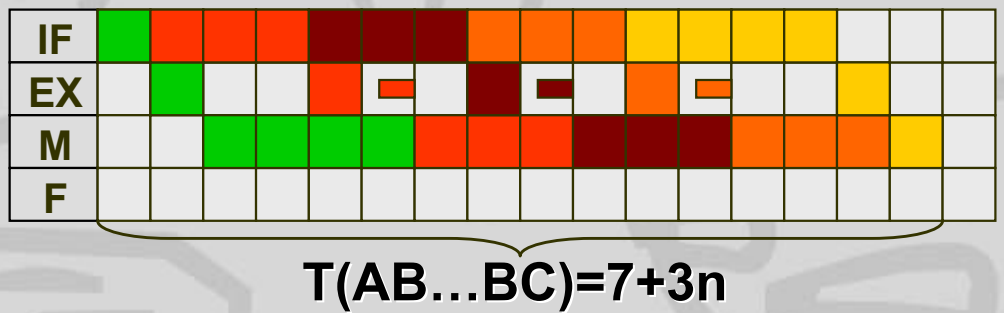


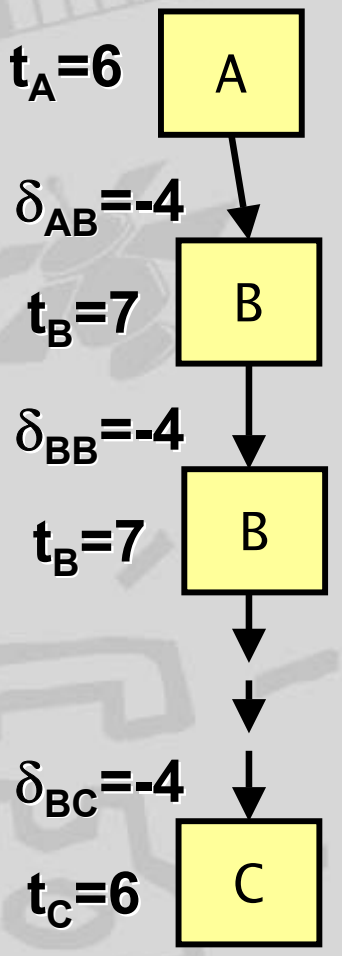


Unbounded LTE

- In a single pipeline

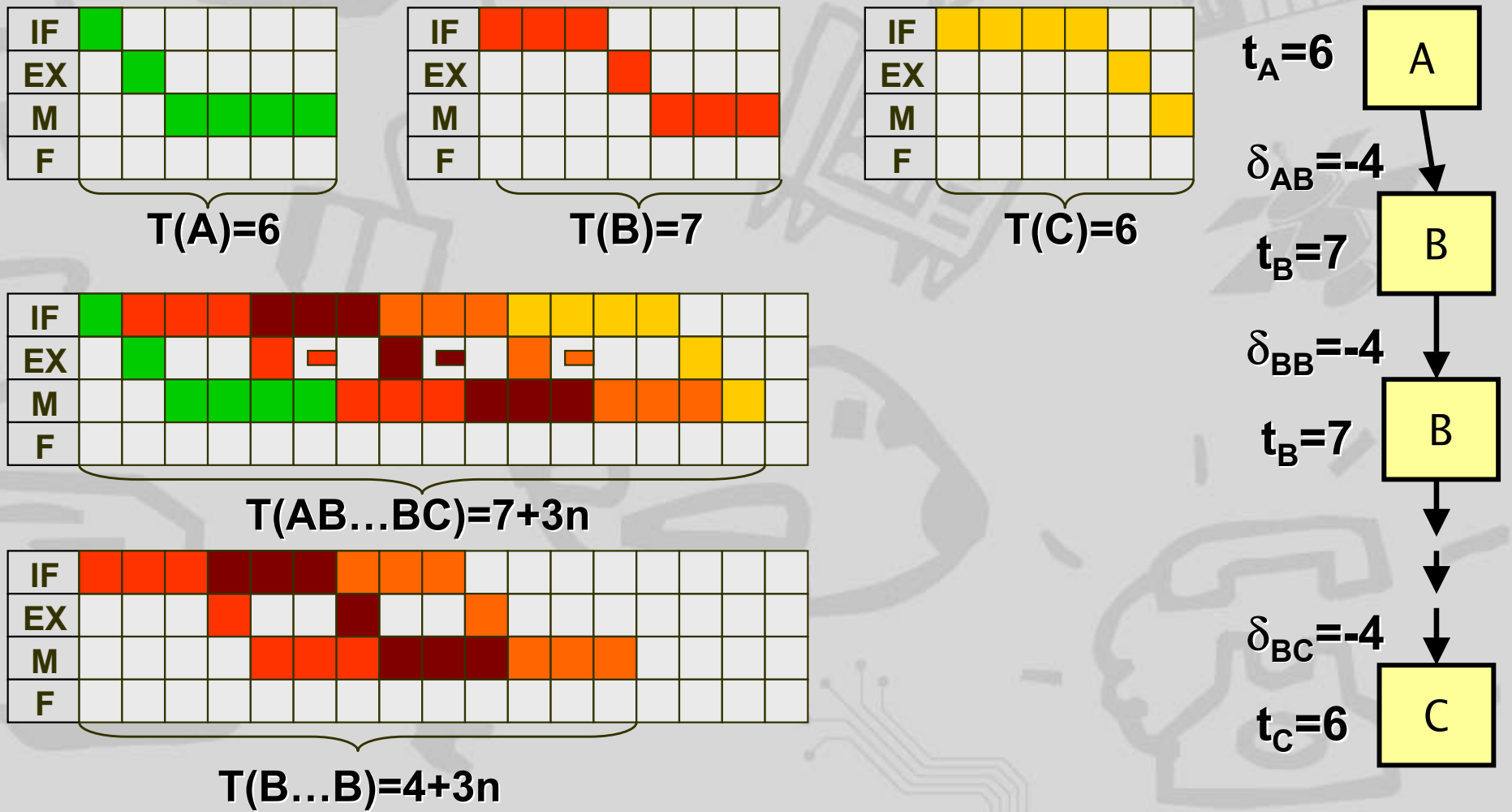






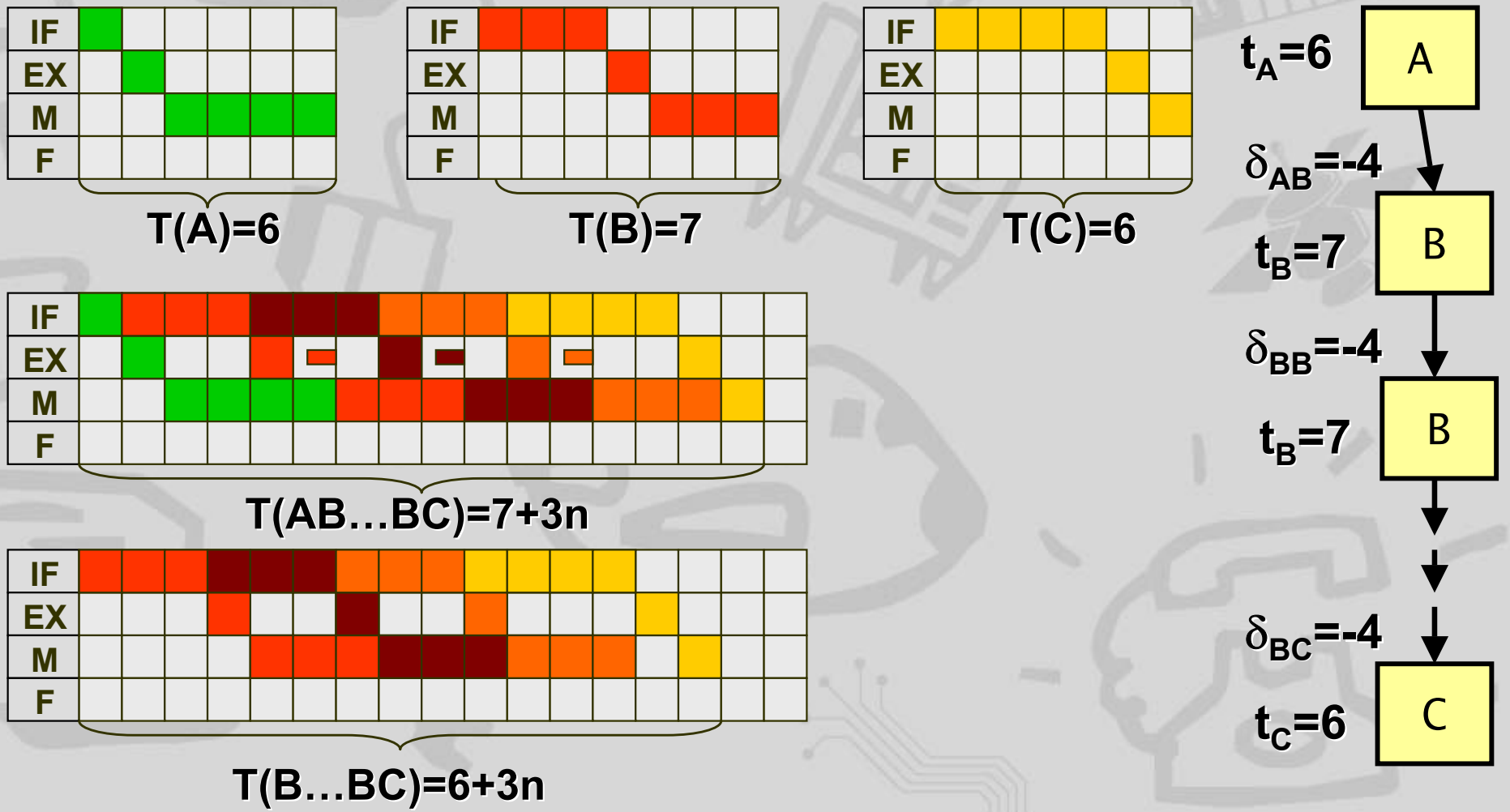
Unbounded LTE

- In a single pipeline



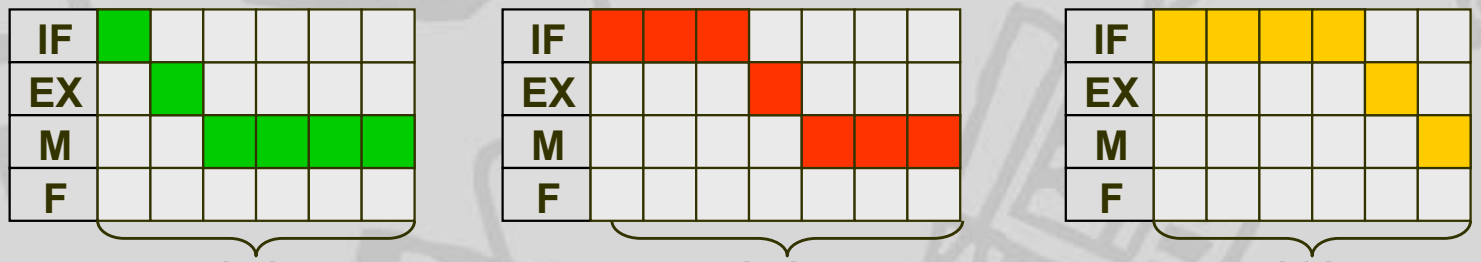
Unbounded LTE

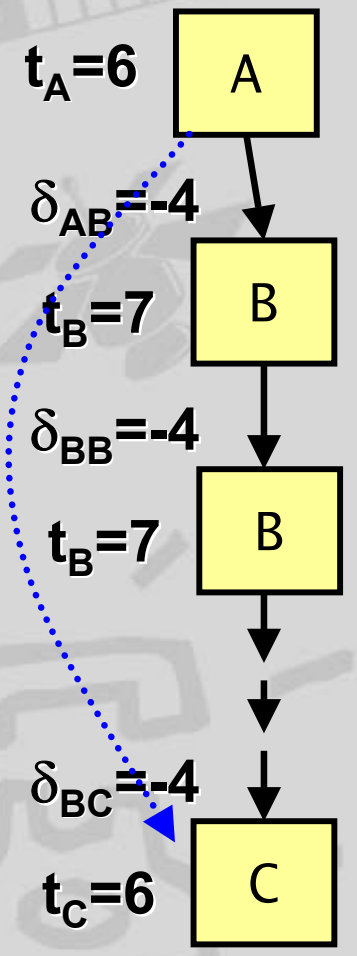
- In a single pipeline



Unbounded LTE

- In a single pipeline





T(AB..BC)=7+3n T(B..B)=4+3n

T(AB..B)=6+3n T(B..BC)=6+3n

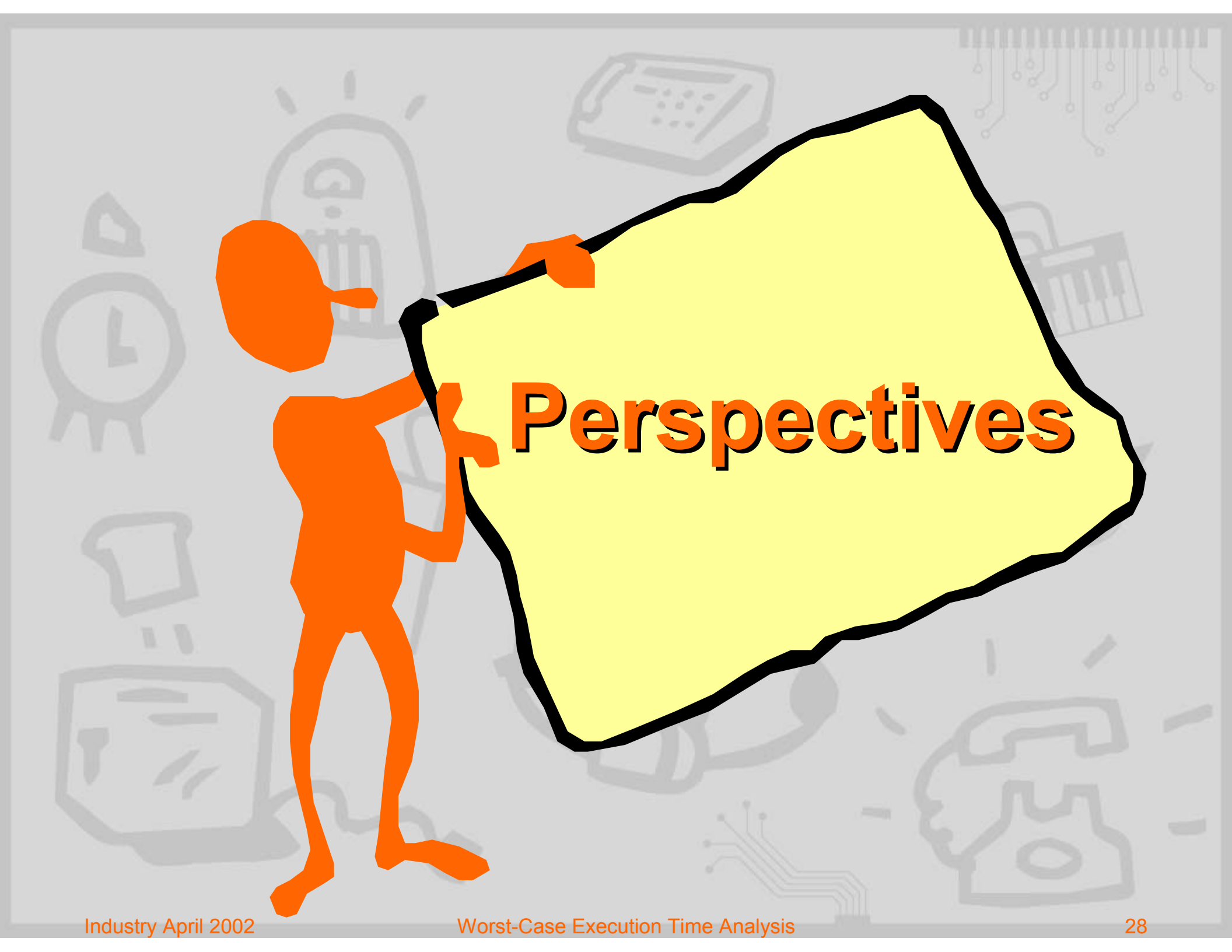
$\delta_{AB...BC}$=-1

$\delta_{AB..BC}$=T(AB..BC)-T(AB..B)-T(B..BC)+T(B..B)

$\delta_{AB..BC}$=7+3n-6-3n-6-3n+4+3n=11-12=-1

Unbounded LTE

- Not just negative
  - Example of positive effect in thesis
- Can appear in the simplest of pipelines (unexpected)
- Consequences:
  - Not obvious that analyzing up to some fixed maximal distance is safe
  - Analysis can get very complex


Perspectives

Conclusions

- Pairwise analysis usually unsafe
  - A few lucky pipelines
- Efficient analysis possible
  - When few & bounded LTEs
  - In-order scalar CPUs without speculation
- General analysis: check all states
  - Applicable to any pipeline with a model
  - More LTEs means more complex analysis
  - Work in Saarbrücken on Coldfire & PPC

The End!