Processor Pipelines and Static WCET Analysis

Jakob Engblom & Bengt Jonsson

Uppsala University

And with IAR Systems, www.iar.com

Currently at Virtutech, www.virtutech.com

jakob.engblom@it.uu.se
What is “WCET”?

» WCET = Worst case
» BCET = Best case
» ACET = Average case
Static WCET Analysis

* Find the WCET of a program
* Using offline static analysis
* Need to model & analyze:
  - Program flow
  - Cache behavior
  - Pipeline
  - … and their interaction

Program

Flow analysis

Low level analysis

Calculation

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

**Example:**

- Instruction B can hit or miss the cache
Program Timing Model

Times ($t_A$)
- For single nodes

Timing effects ($\delta_{ABC}$)
- For sequences of nodes
- Pairwise or long (>2 nodes)
- Effect of differing previous nodes

Execution time: $T(A..Z) = \sum t + \sum \delta$
- All times & all subsequences for A..Z
Program Timing Model

★ Deriving timing effects:

★ For sequence

★ δ_{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)

★ δ_{A..Z} ≠ 0 means that A affects Z
Pairwise Timing Effects

Effects between two instr:

\[ \delta_{AB} = 7 - 6 - 3 + 0 = -2 \]

"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

A → B: t_A = 6, \delta_{AB} = -2

B → C: t_B = 3, \delta_{BC} = -2

C: t_C = 6, \delta_{ABC} = -3
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
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 \( l_1 .. l_m \) require that:

- The first instruction \( (l_1) \) stalls one of its successors
- OR
- The first instruction \( (l_1) \) reaches past \( l_2 .. l_{m-1} \)

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 instructions
  - Many such cases on the V850E
Only Negative LTE

- Prove absence of positive $\delta$
- 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
In a single pipeline

- **T(A)** = 6
- **T(B)** = 7
- **T(C)** = 6
- **T(AB)** = 9
- **T(BB)** = 10
- **T(BC)** = 9

- t_A = 6
- δ_{AB} = -4
- t_B = 7
- δ_{BB} = -4
- t_C = 6

Unbounded LTE
Unbounded LTE

*In a single pipeline*

- \( T(A) = 6 \)
- \( T(B) = 7 \)
- \( T(C) = 6 \)
- \( T(AB \ldots B) = 6 + 3n \)

- \( t_A = 6 \)
- \( \delta_{AB} = -4 \)
- \( t_B = 7 \)
- \( \delta_{BB} = -4 \)
- \( t_C = 6 \)
- \( \delta_{BC} = -4 \)
Unbounded LTE

*In a single pipeline*

\[ T(A) = 6 \]
\[ T(B) = 7 \]
\[ T(C) = 6 \]

\[ T(AB...BC) = 7 + 3n \]
Unbounded LTE

In a single pipeline

T(A) = 6
T(B) = 7
T(C) = 6

T(AB…BC) = 7 + 3n
T(B…B) = 4 + 3n

δ_{AB} = -4
δ_{BB} = -4
δ_{BC} = -4

t_A = 6
t_B = 7
t_C = 6
Unbounded LTE

*In a single pipeline*

IF
EX
M
F

T(A)=6

T(B)=7

T(C)=6

T(AB…BC)=7+3n

δ_{AB}=-4

δ_{BB}=-4

δ_{BC}=-4

t_{A}=6

t_{B}=7

t_{C}=6
Unbounded LTE

*In a single pipeline*

\[
\begin{align*}
\delta_{AB..BC} &= T(AB..BC) - T(AB..B) - T(B..BC) + T(B..B) \\
&= 7 + 3n - 6 - 3n - 6 - 3n + 4 + 3n = 11 - 12 = -1
\end{align*}
\]
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
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!