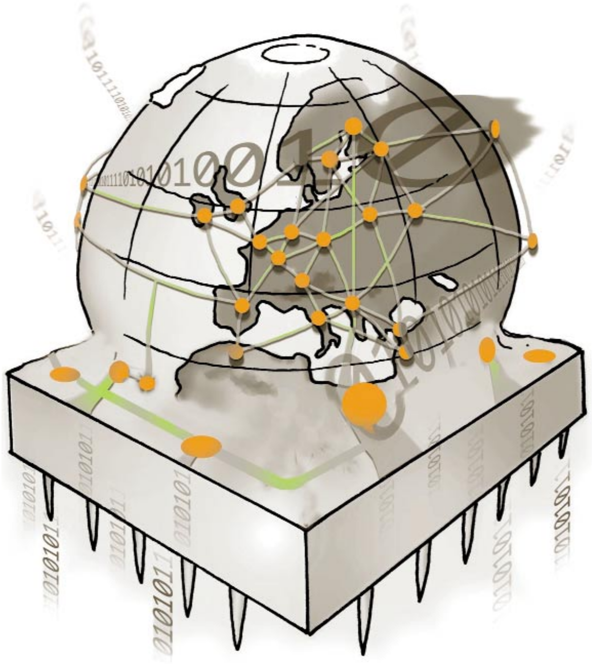
Evaluating TTA and some other novel ILP processor architectures



Martti Forsell
PhD, Senior Research Scientist
VTT Electronics, Oulu, Finland

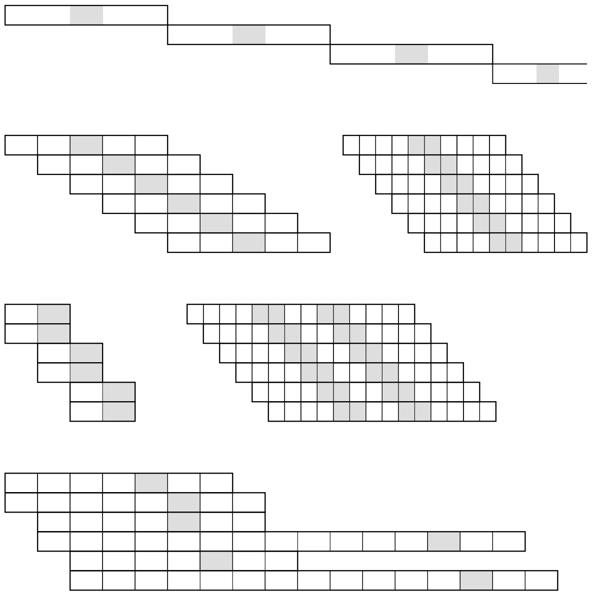
TTA Workshop
November 25-26, 2002
Nokia Research Center, Tampere

VTT ELECTRONICS

Contents

ELECTRONICS

Motivation



Processor architect has a wide spectrum of choices in designing processor execution pipeline:

- Number of pipeline stages
- Number of pipelines/units
- Order of execution
- Number of threads
- Organization of functional units
- Scheduling of segments/operations

ELECTRONICS

Problem

?

How some novel architectures like

TTA [Corporaal92, Corporaal98]
MPA [Forsell94, Forsell96]
MTAC [Forsell97]

perform compared to an out-of-order superscalar architecture (SS) [Hennessy90]?

ELECTRONICS

Compared processors

TTA	TTA processor with MOVE32INT-style 4-stage pipeline, *F* functional units and 2*F* transport channels
TTAm	TTA processor with MOVEInt32-style pipeline, *F* functional units and *F*/2 transport channels (minimized,cost effective TTA)
SS	Out-of-order *F*-issue superscalar processor with *F* functional units
MPA	Minimal pipeline architecture processor with *F* functional units
MTAC	Multithreaded architecture with chaining processor with *F* functional units and *T* threads

Our contribution

(i) We parameterize these architectures

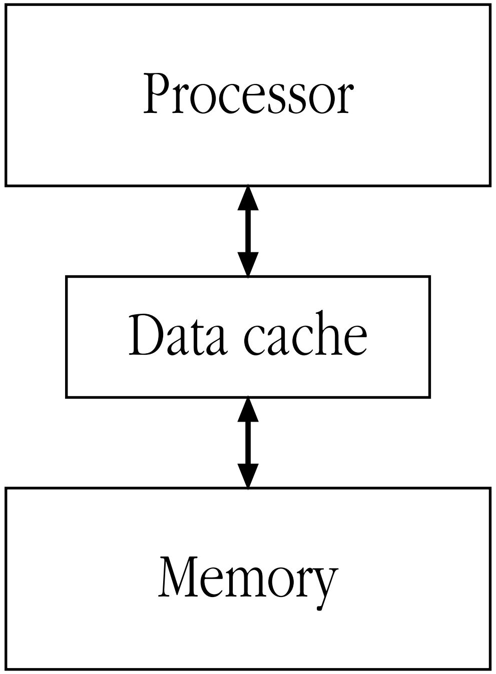
(ii) We will evaluate obtained architectures analytically using a parametrical test program in two system setups

(iii) We estimates roughly the area required by the forwarding/issuing network in single threaded architectures

These methodologies are described in details in [Forsell02b, Forsell96].

ELECTRONICS

Memory system setups — single processor setup (SPS)



A processor is connected to a hierarchized memory system

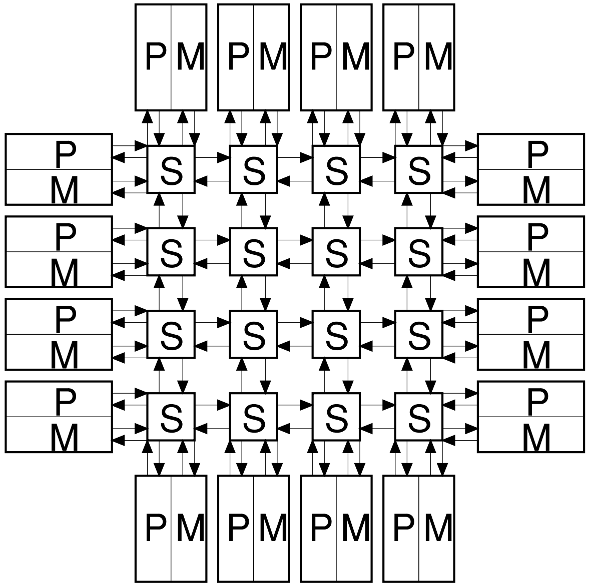
Modeled parameters:

- Probability of a cache miss
- Delay in a case of a cache miss
- Probability that an instruction references to memory

Instruction caches are not modeled

ELECTRONICS

Memory system setups
— multiprocessor setup (MPS)



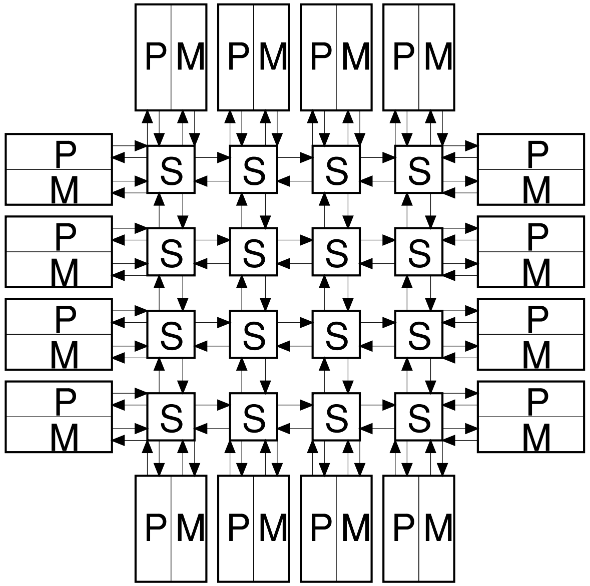
The most advanced parallel memory system available:

- *P* processors connected to a cacheless memory system featuring efficient shared memory abstraction [Ranade91]

- Topology of a network is 3D coated mesh or sparse mesh [Leppänen96, Forsell02c]

ELECTRONICS

Memory system setups
— multiprocessor setup (continued)

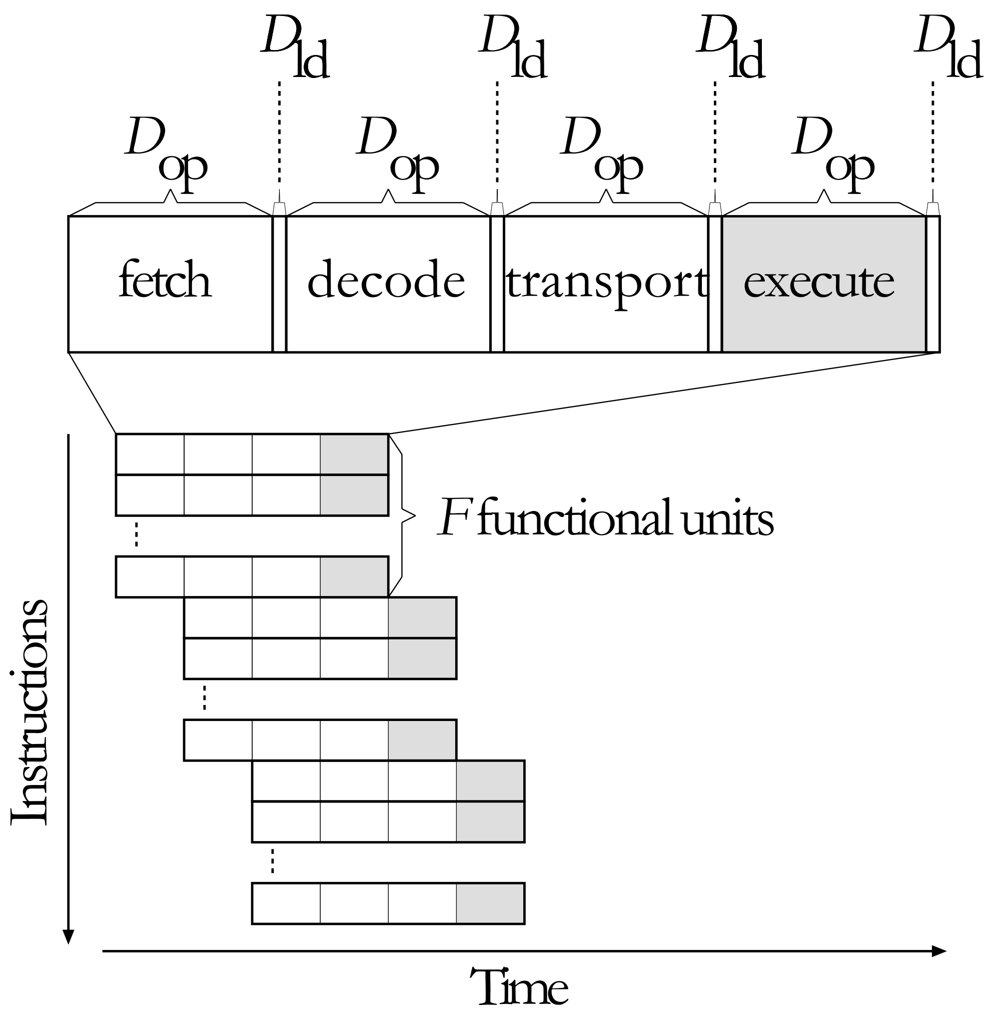


Modeled parameters:

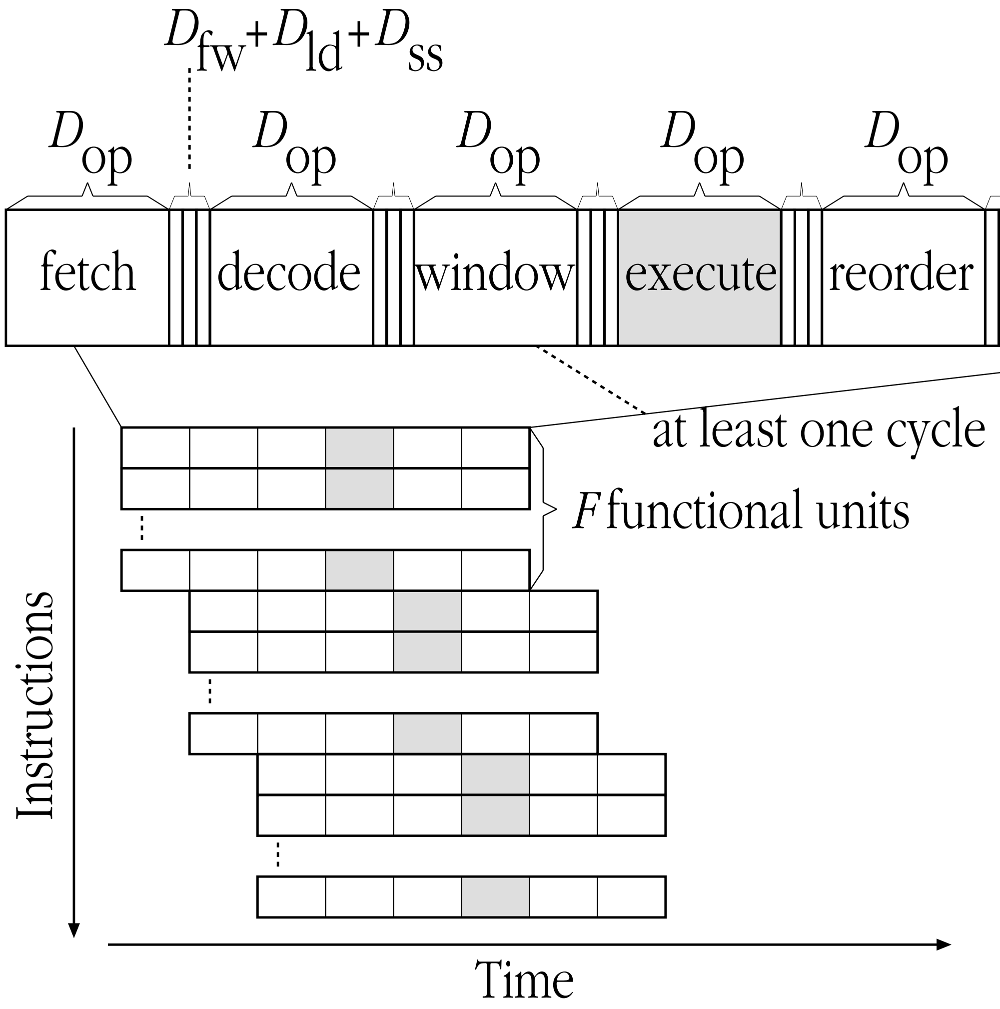
- The number of processors
- Memory read latency
- The number of threads
- Utilization of multithreading

ELECTRONICS

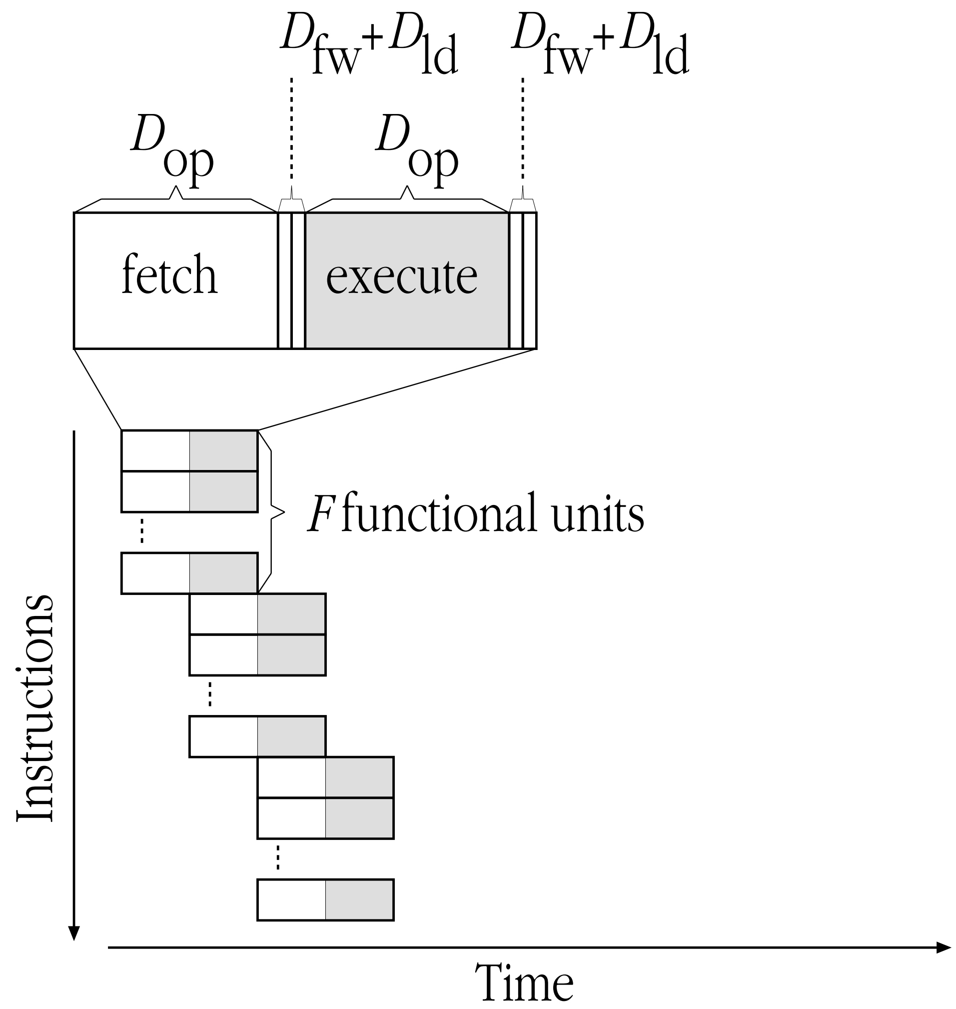
Pipeline organizations—TTA



Pipeline organizations—superscalar

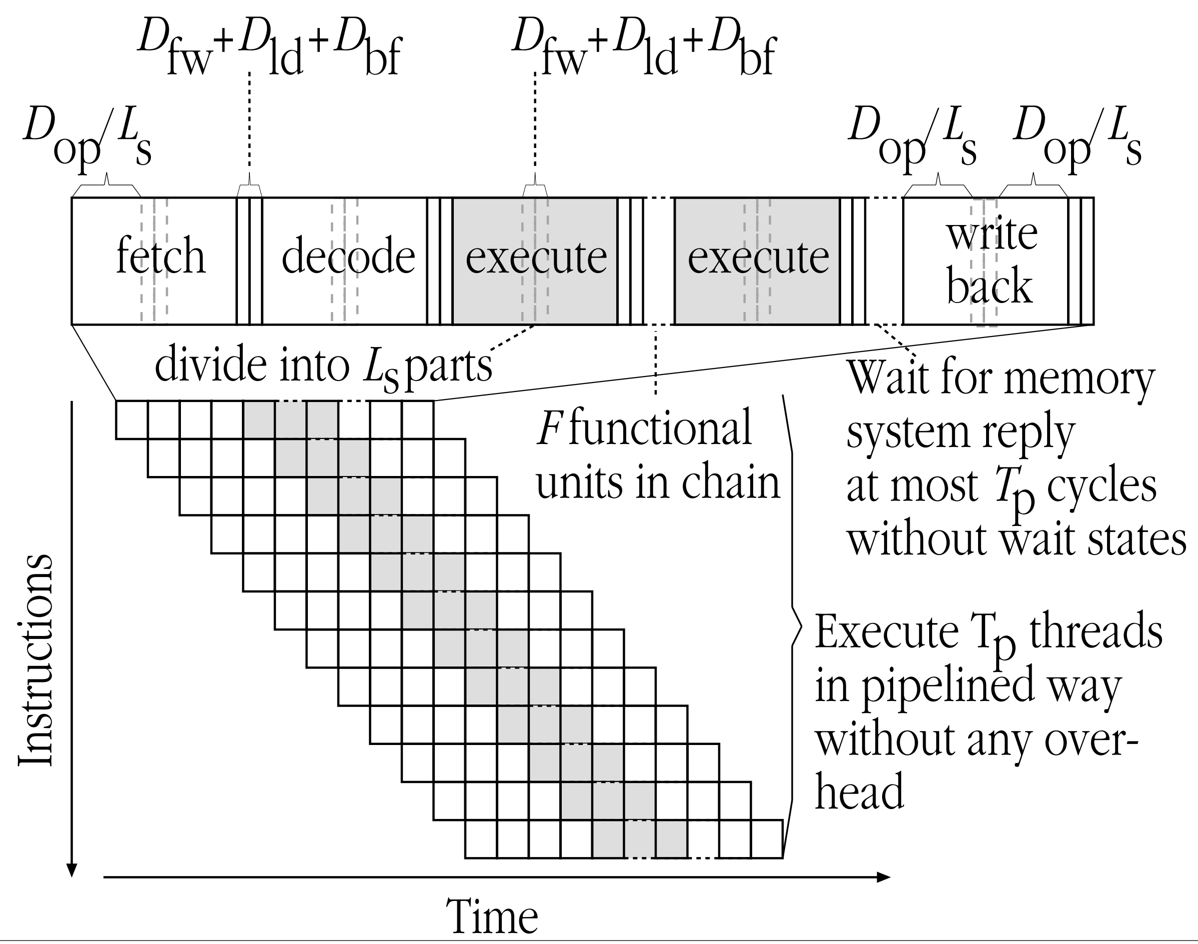


Pipeline organizations—MPA



ELECTRONICS

Pipeline organizations—MTAC



ELECTRONICS

Analytical performance models

For each architecture we analytically model

- minimum clock cycle time
- execution time of a parametric benchmark program called ***bench***

for both single processor and multiprocessor setups

We use simplified modeling by ignoring

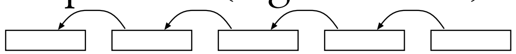
- conditional execution
- speculative loads

because they can be used in all architectures

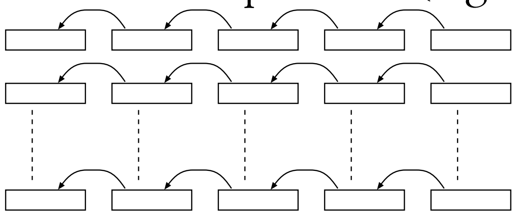
==> the models **can not be used as precise practical tools**, but as approximate models that are **accurate enough for determining the theoretical goodness of an architecture** with respect to the other evaluated architectures.

Analytical performance models

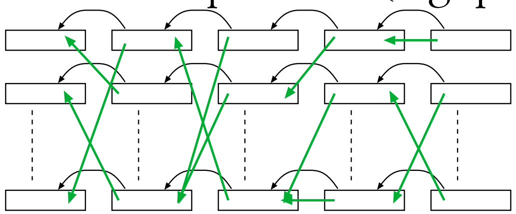
Sequential (e.g. IIR-filter)



Parallel independent (e.g. matmul)



Parallel dependent (e.g. prefix sum)



IL dependence
TL dependence

Models of

- Minimum clock cycle time
- Memory reference latency
- Thread switch delay
- Execution time of the benchmark program composed of portions of three types of code
  - sequential
  - independent parallel
  - dependent parallel

VTT ELECTRONICS

Models—TTA

$$D_{cycle} = \max(D_{op} + D_{ld}, D_{soc} + D_{fw} + D_{de} + D_{ld})$$

$$
\begin{aligned}
T_{tta} = O\, D_{cycle}(\\
&F_s (1 - P_{mnd} - P_{bnd}) (2 + P_m P_{cm} C_{cm} + P_b) \\
&+ F_{ip} (1 + P_m P_{cm} C_{cm}) / F U_f P \\
&+ F_{dp} \min((1 - P_{mnd} - P_{bnd}) \\
&(2 + P_m P_{cm} C_{cm} + P_b), \\
&(\lfloor 1/P \rfloor (1 - P_{mnd} - P_{bnd}) \\
&(4 + C_{sw} + P_m P_{cm} C_{cm}) + \\
&\lceil (P-1)/P \rceil (1 - P_{mnd} - P_{bnd} + P_{rd}) \\
&(4 + C_{sw}))/P))
\end{aligned}
$$

VTT ELECTRONICS

Models—TTA (shortly explained)

fraction of sequential operations

Delay caused by a cache miss

Delay caused by a delayed branch

$$T_{tta} = O/D_{cycle}($$

$$F_s\,(1 - P_{mnd} - P_{bnd})\,(2 + P_m P_{cm} C_{cm} + P_b)$$

Sequential portions

$$+\ F_{ip}(1 + P_m P_{cm} C_{cm})/FU_f P$$

Independent parallel portions

$$+\ F_{dp}\min(\,(1 - P_{mnd} - P_{bnd})$$
$$(2 + P_m P_{cm} C_{cm} + P_b),$$
$$(\lfloor 1/P \rfloor\,(1 - P_{mnd} - P_{bnd})$$
$$(4 + C_{sw} + P_m P_{cm} C_{cm}) +$$
$$\lceil (P\text{-}1)/P \rceil\,(1 - P_{mnd} - P_{bnd} + P_{rd})$$
$$(4 + C_{sw}))/P))$$

Dependent parallel portions

Minimum of execution time in one processor and execution time in all processors (requiring thread switches)

ELECTRONICS

Models—TTAm

$$D_{cycle} = \max(D_{op} + D_{ld}\,,\, D_{soc} + D_{fw} + D_{de} + D_{ld})$$

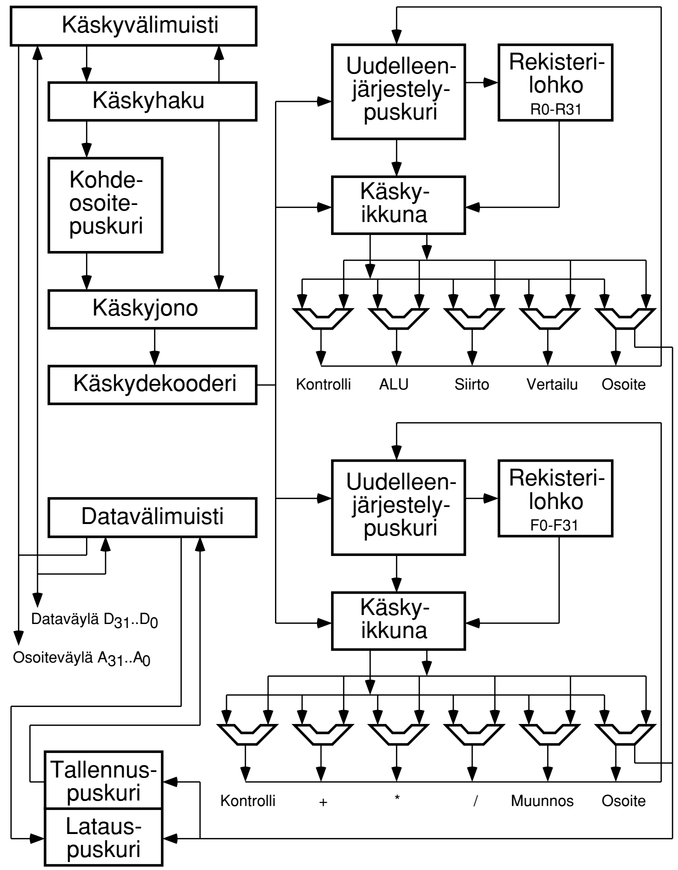
$$\begin{aligned}
T_{ttam} = O\, D_{cycle}(& \\
& F_s (1 - P_{mnd} - P_{bnd})(2 + P_m P_{cm} C_{cm} + P_b) \\
& + \mathbf{4} F_{ip}(1 + P_m P_{cm} C_{cm})/F\, \boldsymbol{U}_{\mathbf{ft}} P \\
& + F_{dp} \min((1 - P_{mnd} - P_{bnd}) \\
& (2 + P_m P_{cm} C_{cm} + P_b), \\
& (\lfloor 1/P \rfloor (1 - P_{mnd} - P_{bnd}) \\
& (4 + C_{sw} + P_m P_{cm} C_{cm}) + \\
& \lceil (P\text{-}1)/P \rceil (1 - P_{mnd} - P_{bnd} + P_{rd}) \\
& (4 + C_{sw}))/P))
\end{aligned}$$

ELECTRONICS

Models—SS

$$D_{cycle} = D_{op} + D_{fw} + D_{ld} + D_{ss}$$

$$\begin{aligned}
T_{ss} = O\,D_{cycle}(& \\
& F_s\,(1 - P_{mnd} - P_{bnd}) \\
& (1 + P_m P_{cm} C_{cm} + P_b P_{mp} C_{mp}) \\
& + F_{ip}(1 + P_m P_{cm} C_{cm})/FU_f P \\
& + F_{dp}\min((1 - P_{mnd} - P_{bnd}) \\
& (1 + P_m P_{cm} C_{cm} + P_b P_{mp} C_{mp}), \\
& (\lfloor 1/P \rfloor (1 - P_{mnd} - P_{bnd}) \\
& (6 + C_{sw} + P_m P_{cm} C_{cm}) + \\
& \lceil (P\text{-}1)/P \rceil (1 - P_{mnd} - P_{bnd} + P_{rd}) \\
& (6 + C_{sw}))/P))
\end{aligned}$$

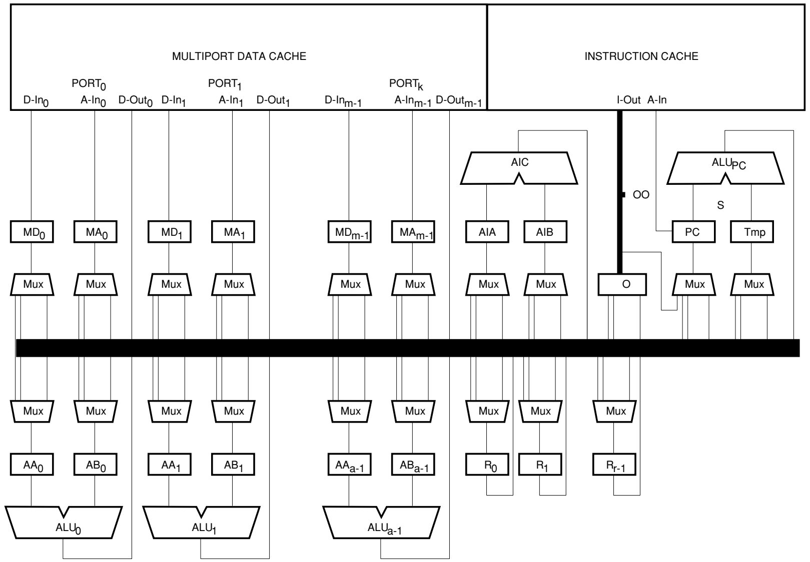


VTT ELECTRONICS

Models—MPA

$$D_{\text{cycle}} = D_{\text{op}} + D_{\text{fw}} + D_{\text{ld}}$$

$$\begin{aligned}
T_{\text{mpa}} = O\,D_{\text{cycle}}(\;& \\
& F_s\,(1 - P_{\text{mnd}} - P_{\text{bnd}})\,(1 \\
& + P_m P_{\text{cm}} C_{\text{cm}}) \\
& + F_{\text{ip}}(1 + P_m P_{\text{cm}} C_{\text{cm}})/FU_f P \\
& + F_{\text{dp}}\min((1 - P_{\text{mnd}} - P_{\text{bnd}}) \\
& (1 + P_m P_{\text{cm}} C_{\text{cm}}), \\
& (\lfloor 1/P \rfloor\,(1 - P_{\text{mnd}} - P_{\text{bnd}}) \\
& (2 + C_{\text{sw}} + P_m P_{\text{cm}} C_{\text{cm}}) + \\
& \lceil (P\text{-}1)/P \rceil\,(1 - P_{\text{mnd}} - P_{\text{bnd}} + P_{\text{rd}}) \\
& (2 + C_{\text{sw}}))/P))
\end{aligned}$$

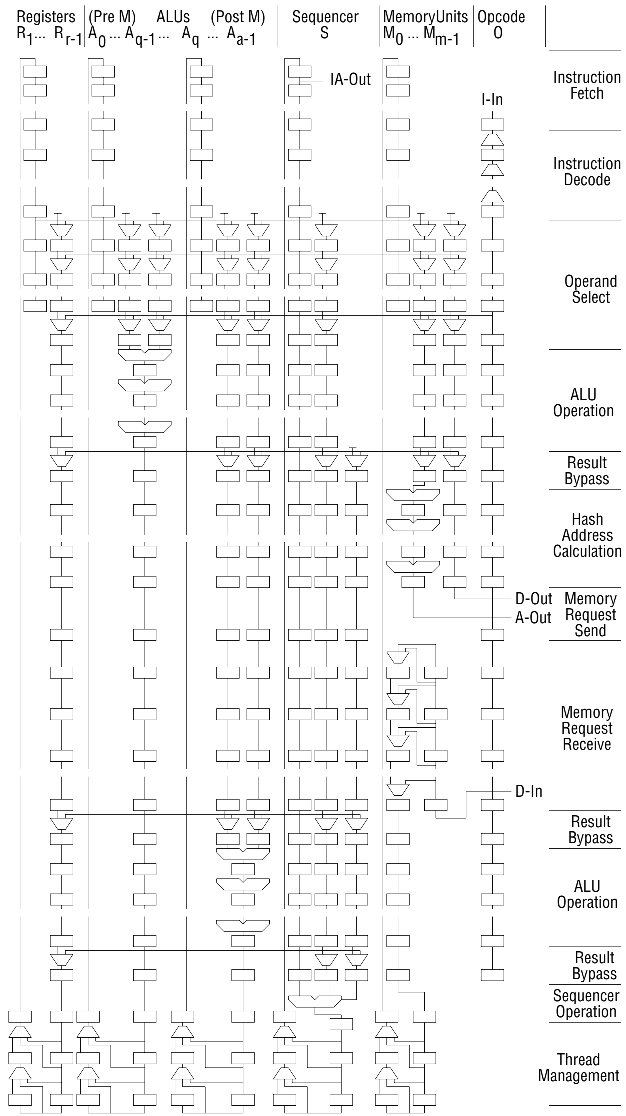


ELECTRONICS

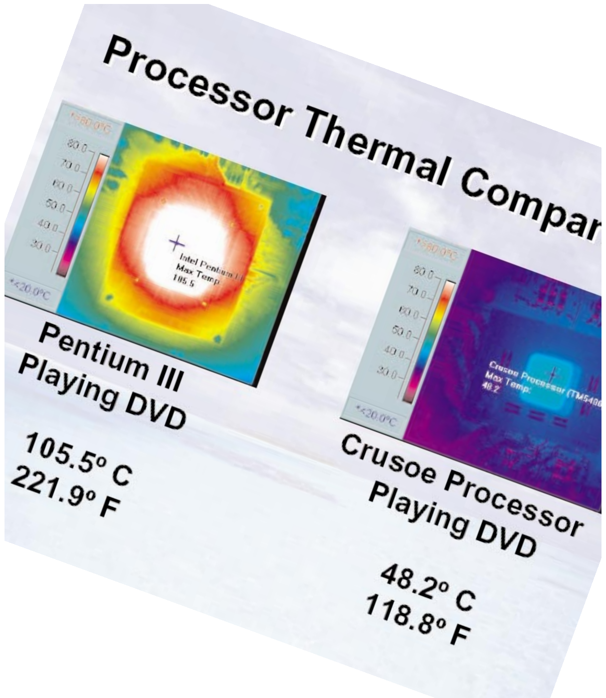
Models—MTAC

$$D_{\mathrm{cycle}} = D_{\mathrm{op}}/L_{\mathrm{s}} + D_{\mathrm{bf}} + D_{\mathrm{fw}} + D_{\mathrm{ld}}$$

$$T_{\mathrm{mtac}} = O\, D_{\mathrm{cycle}}(F_{\mathrm{s}} \max(D_{\mathrm{mem}}/D_{\mathrm{cycle}}, T_{\mathrm{p}}) + (F_{\mathrm{ip}} + F_{\mathrm{dp}})/(P\, U_{\mathrm{t}}))/L_{\mathrm{c}}\, U_{\mathrm{c}}$$



Comparison



The performance of these architectures are compared to each other with “realistic” parameter values assuming the same program, speed of logic and number of pipeline segments for each architecture

- Values are taken from quantitative architecture studies and experiments [Hennessy90, Forsell97]

Parameters

Number of additional cycles taken by a cache miss	Ccm	
Number of additional cycles taken by a main memory reference	Cmem	
Number of additional cycles taken by a branch misprediction	Cmp	3
Number of additional cycles taken by a context switch	Csw	
Delay caused by balancing pipeline stages	Dbf	
Delay caused by a processor cycle	Dcycle	
Delay caused by decoding an instruction	Dde	700 ps
Delay caused by executing an ALU instruction/referencing a data cache	Dex	800 ps
Delay caused by forwarding	Dfw	100 ps
Delay caused by an instruction fetch	Dif	800 ps
Delay caused by inter-stage latching in pipeline	Dld	100 ps
Delay caused by a memory system operation	Dmem	64000 ps
Delay caused by an operation	Dop	800 ps
Delay caused by managing an instruction window	Dss	100 ps
Delay caused by a register write back	Dwb	300 ps
Delay caused by a write socket operation	Dsoc	100 ps

ELECTRONICS

Parameters (continued)

Number of functional units	F	8
Fraction of code that is dependent parallel	Fdp	0.5
Fraction of code that is independent parallel	Fip	0.5
Fraction of code that is sequential	Fs	0
Level of chaining	Lc	8
Level of superpipelining	Ls	1
Number of operations	O	1.00E+12
Number of processors	P	400
Probability that an instruction is a branch	Pb	0.2
Probability that an instruction is an independent branch (e.g. jump)	Pbnd	0.1
Probability of a cache miss	Pcm	0.25
Probability that an instruction is a memory reference	Pm	0.3
Probability that an instruction is an independent memory reference	Pmnd	0.2
Probability of a branch misprediction	Pmp	0.2
Probability that an instruction is a read	Prd	0.2

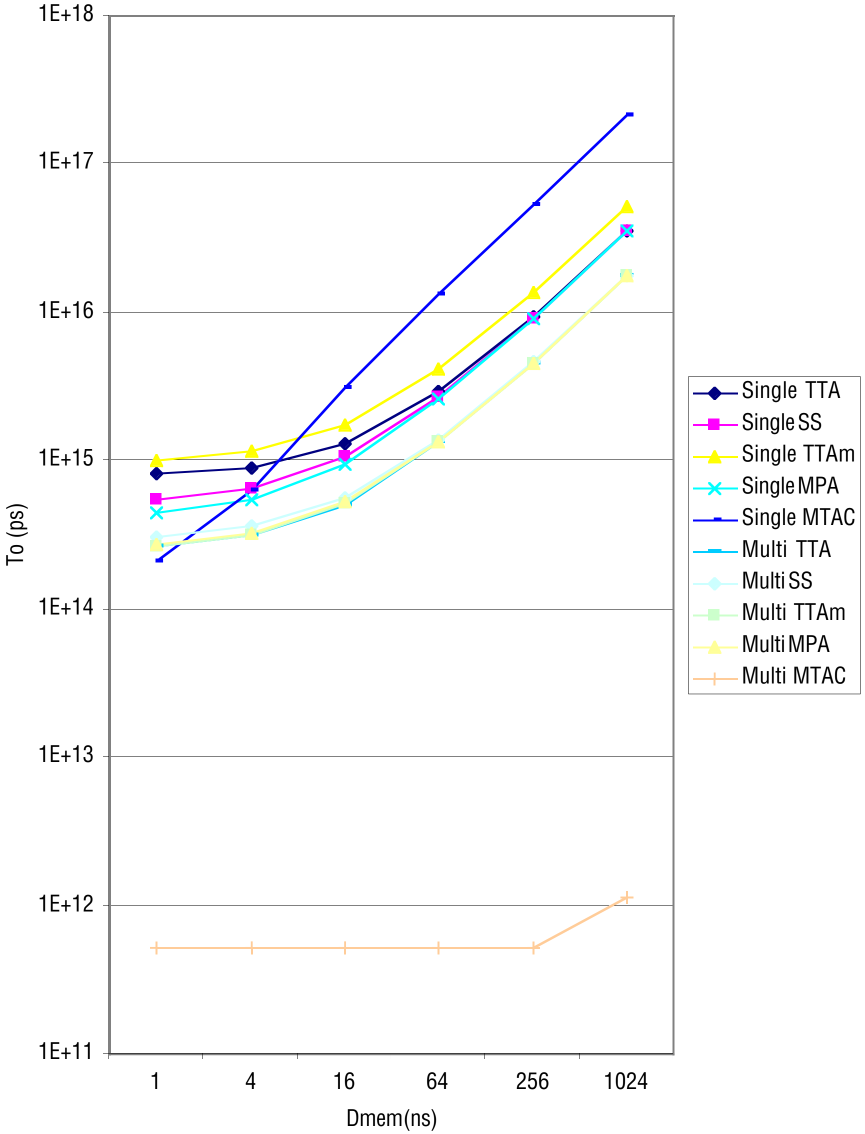
ELECTRONICS

Parameters (continued)

Number of switches	S	
Number of threads in a program	T	1000000
Fan-in constant	t	4
Time taken by executing a program with O operations	To	
Number of threads per processor	Tp	500
Utilization of chained functional units	Uc	0.6
Utilization of functional units	Uf	0.6
Utilization of multithreading	Ut	
Utilization of functional units in limited transport capacity situation	Uft	0.8

ELECTRONICS

Results

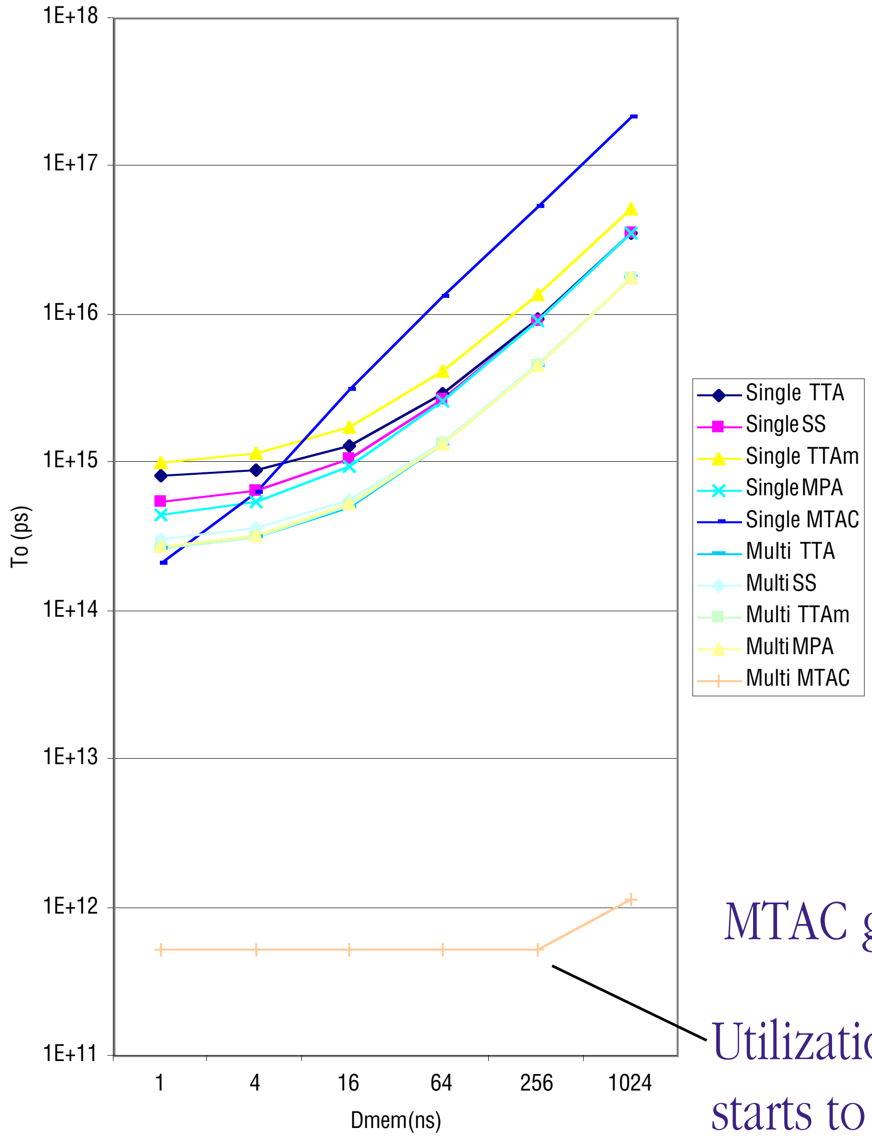


Execution time as the function of memory system delay

- Assumptions: F=8, F_{dp}=0.5, F_{ip}=0.5, P=400, T_p=500.

ELECTRONICS

Results (continued)



Single processor setups give somewhat expected behavior

MPA gives the best performance with SPS

Multiprocessor setups

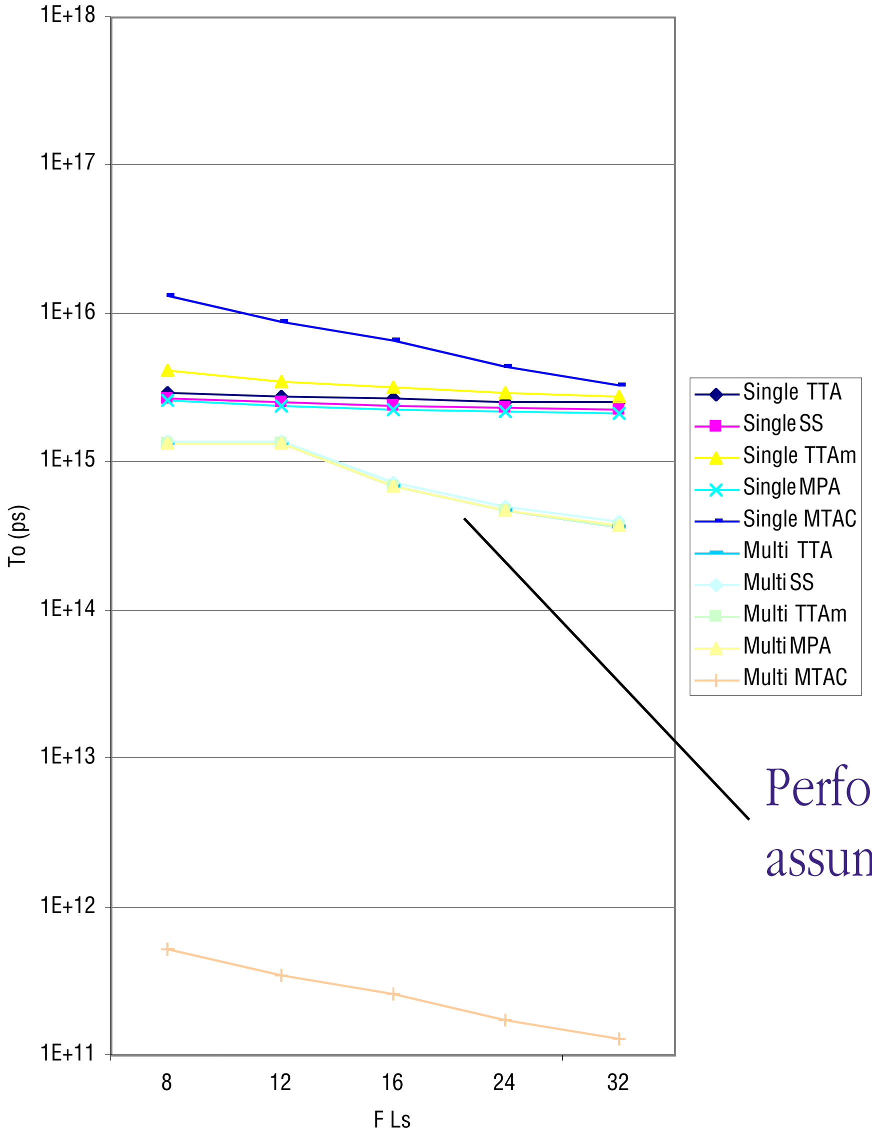
MTAC gives the best performance with MPS

MTAC gives exceptional performance!

Utilization of threading starts to drop below one

ELECTRONICS

Results (continued)



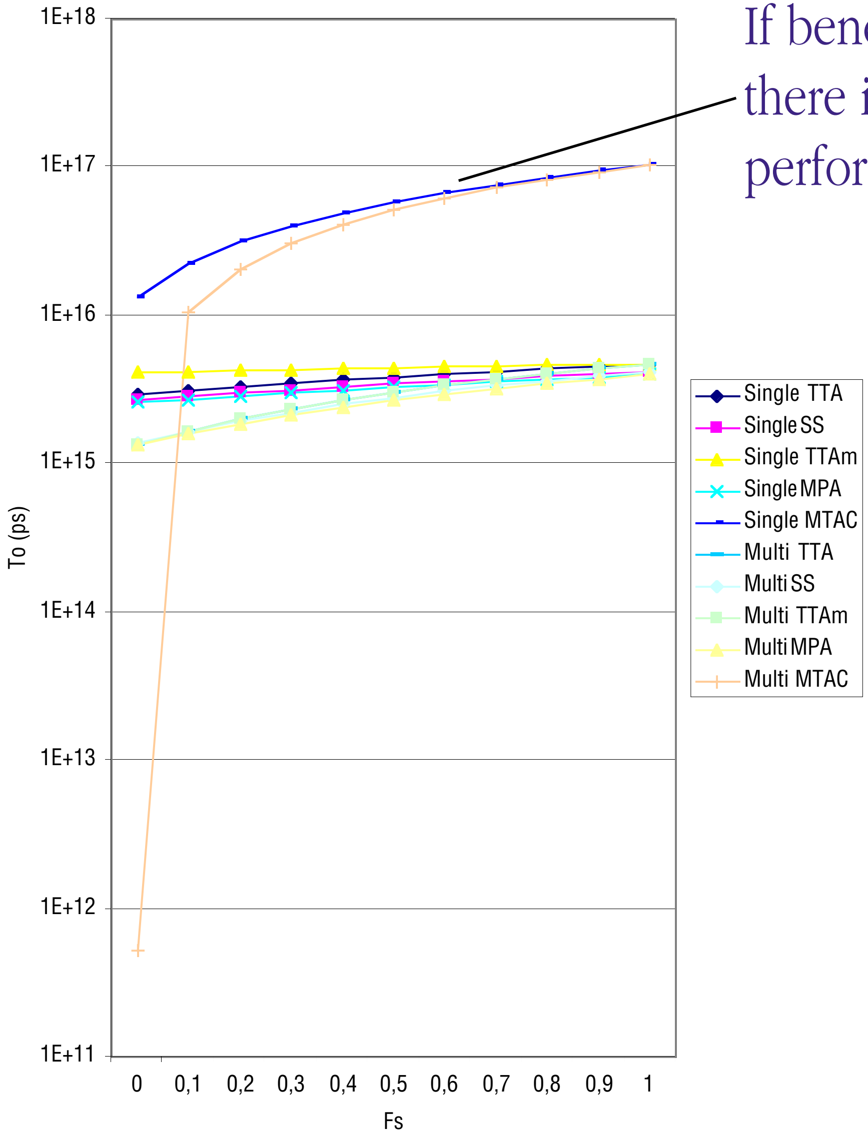
Execution time as the function of number of pipeline segments

- Assumptions: D_{mem}=64, F_{dp}=0.5, F_{ip}=0.5, P=400, T_p=500.

Performance increases, because we assume that every eighth unit is MU!

ELECTRONICS

Results (continued)



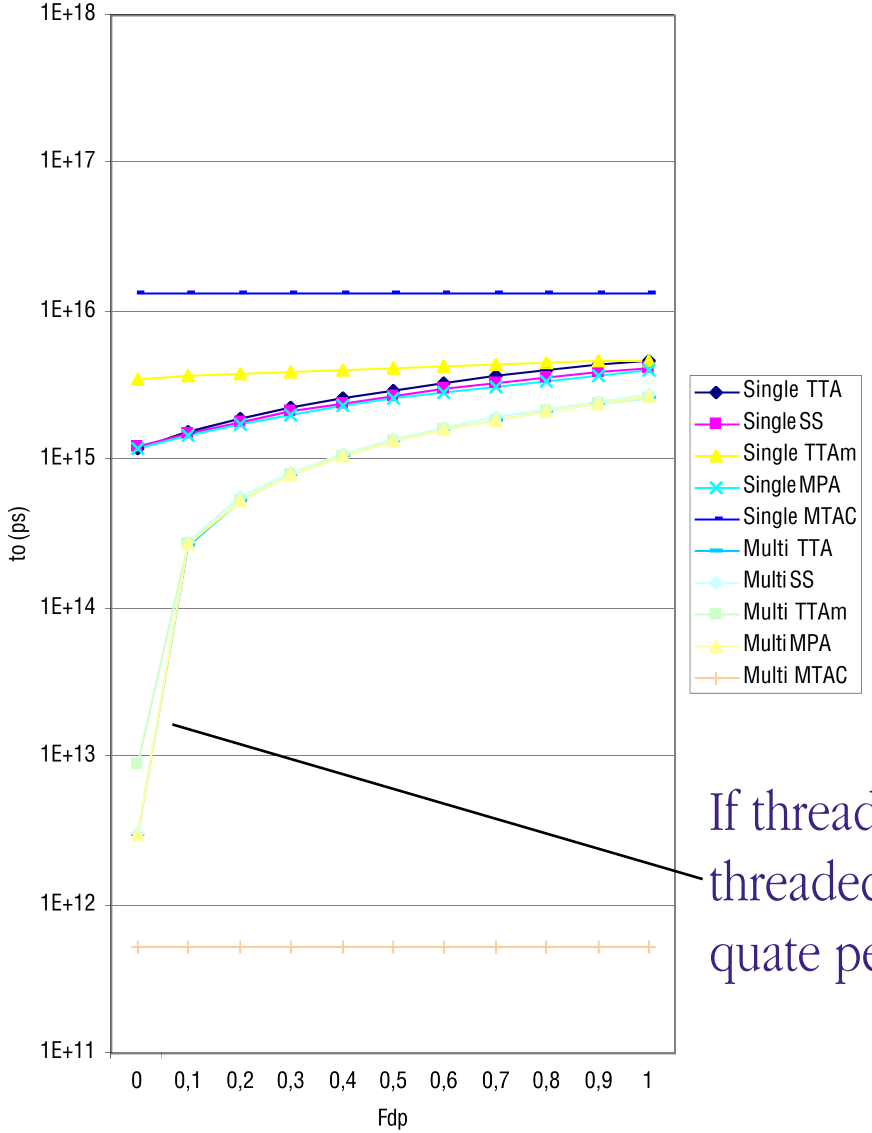
If bench contains partitions in which there is only one sequential thread the performance of MTAC drops dramatically!

Execution time as the function of fraction of sequential code

- Assumptions: D_{mem}=64, F=8, F_{dp}=(1-F_s)/2, F_{ip}=(1-F_s)/2, P=400, T_p=500

VTT ELECTRONICS

Results (continued)

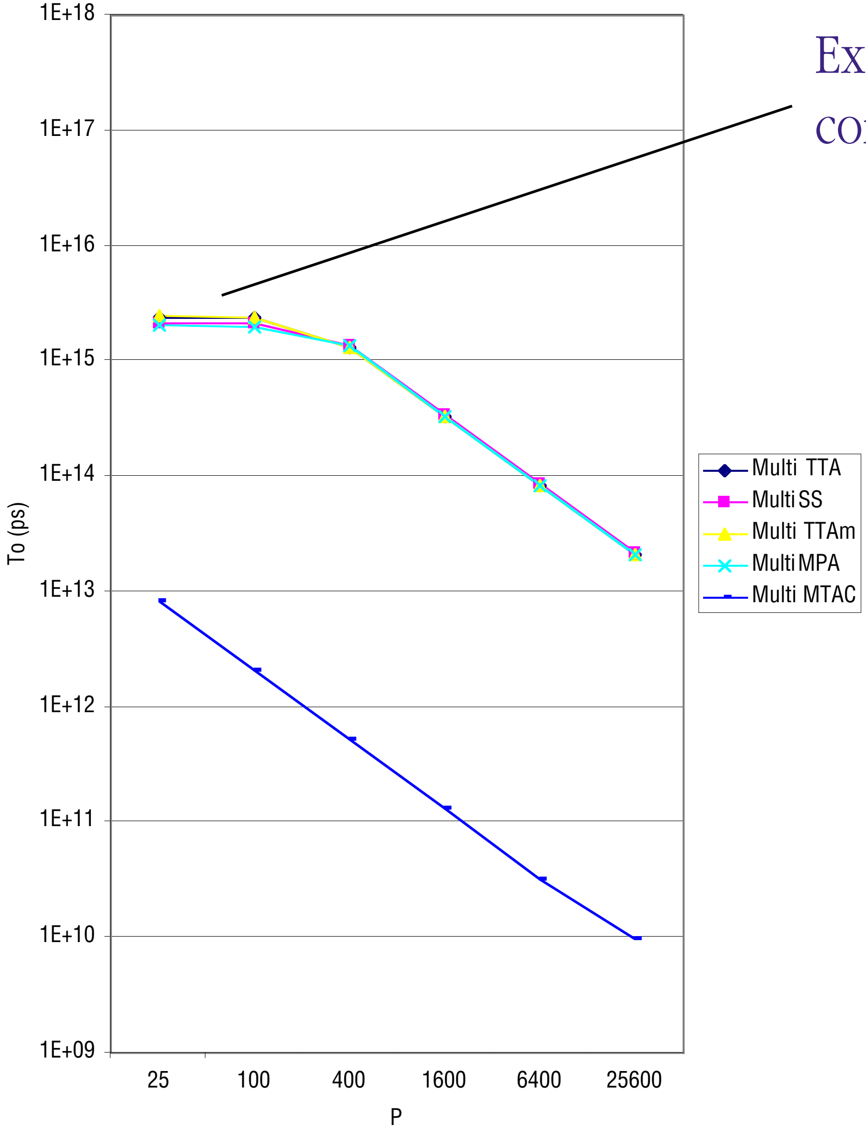


Execution time as the function of fraction of dependent parallel code

- Assumptions: D_{mem}=64, F=8, F_s=0, P=400, T_p=500

If threads are independent then even single threaded processors are able to give adequate performance in multiprocessor setup

Results (continued)



Execution is faster locally due to communication overhead!

Execution time as the function of number of processors

- Assumptions: D_{mem}=64, F=8, F_{dp}=0.5, F_{ip}=0.5, P=400, T_p=500

ELECTRONICS

Results—wiring area

Very simple wiring area model of the issue/forwarding/transport network (assuming unit are placed in row and one metal layer is used):

#FUs	TTA	TTAm	SS	MPA
4	1.44k	0.48k	4.86k	6.46k
10	7.40k	2.12k	17.9k	16.8k
25	40.8k	10.8k	78.4k	54.8k

ELECTRONICS

Conclusions (single processor setup)

- TTA provides 56-100% of the performance of optimal architecture due to inefficient pipeline organization, if a high capacity transport network is used. The forwarding network takes 22-75% of the area of the optimal architecture and 30-52% of the area of the superscalar alternative, respectively.

- If a cost effective transport network is used the performance decreases to 34-86% of the optimal architecture due to limited transport capacity. The forwarding network drops to 7-20% of the area of the optimal architecture.

- The superscalar architecture provides also better performance, but with the cost of increased forwarding/issuing network area. The performance advantage may actually be smaller than measured due to limitations of the dynamic scheduling of instructions.

Conclusions (multiprocessor setup)

- Like the other single threaded architectures TTA is not able to compete against the efficient multithreaded architectures like MTAC, although data partitioning in TTA and other single threaded systems is assumed unrealisticly perfect.
- TTA-based system is able to deliver only a small fraction (0,006-22 %) of the performance of the MTAC based system. This is due to inability to hide latency of the memory system, inefficient thread switching and limited number of registers.
- The area of MTAC is estimated to be the same as TTA + caches, because while MTAC needs a lot of space for threads it does not use caches at all.

Conclusions (recommendations)

- The cost effective version of TTA seems especially suitable for embedded systems which are area and power constrained

- Due to performance overheads we consider TTA not perfect for general purpose computing

- Multiprocessor systems and single processor high speed (or real-time) systems with high number of concurrent tasks need new multithreaded architectures like MTAC which can efficiently exploit both ILP and TLP

- MTAC can even execute fully sequential threads with the rate of multiple instructions per step. (This is not true for single threaded processors or multithreaded processors featuring parallel organization of FUs.)

References

[Hoogerbrugge92] J. Hoogerbrugge and H. Corporaal, Comparing Software Pipelining for Operation-Triggered and a Transport-Triggered Architecture, 1992.

[Corporaal94] H. Corporaal, MOVE32INT Architecture and Programmer's Reference Manual, Technical Report 1-68340-44-(1993)01, Department of Computer Architecture, Delft University of Technology, the Netherlands, 1994.

[Corporaal98] H. Corporaal, Microprocessor Architecture from VLIW to TTA, John Wiley & Sons, Chichester, 1998.

[Ranade91] A. Ranade, How to Emulate Shared Memory, Journal of Computer and System Sciences 42, (1991), 307--326.

VTT ELECTRONICS

References

[Hennessy90] J. Hennessy and D. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann Publishers Inc., Palo Alto, 1990.

[Leppänen96] V. Leppänen, Studies on the realization of PRAM, Dissertation 3, Turku Centre for Computer Science, University of Turku, Turku, 1996.

[Forsell94] Martti Forsell, Design and analysis of some chip level parallel architectures, Licentiate thesis, Department of Computer Science, University of Joensuu, Joensuu, 1994

[Forsell96] M. Forsell, Minimal Pipeline Architecture-an Alternative to Superscalar Architecture, Microprocessors and Microsystems 20, 5 (1996), 277-284.

[Forsell97a] M. Forsell, MTAC—A Multithreaded VLIW Architecture for PRAM Simulation, Journal of Universal Computer Science 3, 9 (1997), 1037-1055.

[Forsell02a] M. Forsell, Sequential Processor Architecture Research at VTT, In TTA Workshop Notes 2002, Editors J. Lilius and S. Virtanen, TUCS General Publications 21, Turku Centre for Computer Science, Turku, 2002.

[Forsell02b] M. Forsell, Architectural differences of efficient sequential and parallel computers, Journal of Systems Architecture 47, 13 (July 2002), 1017-1041.

[Forsell02c] M. Forsell, A Scalable High-Performance Computing Solution for Network on Chips, IEEE Micro 22, 5 (September-October 2002), 46-55.