



Vorstellung der SUN Rock-Architektur

Hauptseminar

Ronald Rist

Dresden, 14.01.2009

Inhalt

Einleitung

Kurzübersicht SUN SPARC-Prozessoren

Übersicht SUN Rock – Features

Spekulative Ausführung

- **Scout Threading**
- *Execute Ahead (EXE)*
- *Simultaneous Speculative Threading (SST)*
- **Transactional Memory**

Ausblick/Kritik

Quellen

Einleitung

- kommerzieller Highend Multithreading-Multicore-Mikroprozessor der SPARC-Familie
- derzeit in der Entwicklung (mehrmals verschoben)
- separat abseits Niagara-Familie (UltraSPARC T1, T2, (T3))
- höhere Pro-Thread-Performance, höhere FP-Performance
- Back-End-Datenbank-Server, FP-intensives High-Performance Computing
- Rock-basierte Server nun erwartet für 2. Hälfte 2009

Kurzübersicht SUN SPARC-Prozessoren

Sun SPARC-Mikroprozessor-Spezifikation													
Modell	Frequenz [MHz]	Architektur Version	Jahr	Prozess [µm]	Transistoren [Millionen]	Kernabmessung [mm ²]	IO Pins	Aufnahme [W]	Spannung [V]	L1 Dcache [K]	L1 Icache [K]	L2 Cache [K]	L3 Cache [K]
SPARC	14,28–40	V7	1987-1992	0,8–1,3	~0,1–1,8	--	160–256	--	--	0–128		kein	kein
microSPARC I	50	V8	1992	0,8	0,8	225	288	2,5	5	4	2	kein	kein
SuperSPARC I	33-65	V8	1992	0,8	3,1	256	293	14,3	5	16	20	1024	kein
microSPARC II	60-125	V8	1992	0,5	2,3	233	321	5	3,3	8	16	kein	kein
SuperSPARC II	75-90	V8	1994	0,8	3,1	299	--	16	--	16	20	2048	kein
TurboSPARC	170-180	V8	1995	0,35	--	--	--	7	--	16	16	1024	kein
UltraSPARC I	140-200	V9	1995	0,5	5,2	315	521	30	3,3	16	16	1024	kein
UltraSPARC II	250-480	V9	1997	0,25	5,4	156	521	21	3,3	16	16	8192	kein
UltraSPARC Ili	270-650	V9	1998	0,25	5,75	148	370	17,6	1,7	16	16	2048	kein
UltraSPARC Iie	400-500	V9	2000	0,18	5,4	--	370	13	1,7	16	16	256	kein
UltraSPARC III	600-1200	V9	2001	0,13	29	330	1368	53	1,6	64	32	8192	kein
UltraSPARC IIIi	1064-1600	V9	2003	0,13	87,5	206	959	52	1,3	64	32	1024	kein
UltraSPARC IV	1050-1350	V9	2004	0,13	66	356	1368	108	1,35	64	32	16384	kein
UltraSPARC IV+	1500-2100	V9	2005	0,09	295	336	1368	90	1,1	64	64	2048	32768
UltraSPARC T1	1000-1400	V9	2005	0,09	279	379	1934	79	1,3	8	16	3072	kein
UltraSPARC T2	900-1400	V9	2007	0,065	503	342	1831	95	1,2	8	16	4096	kein

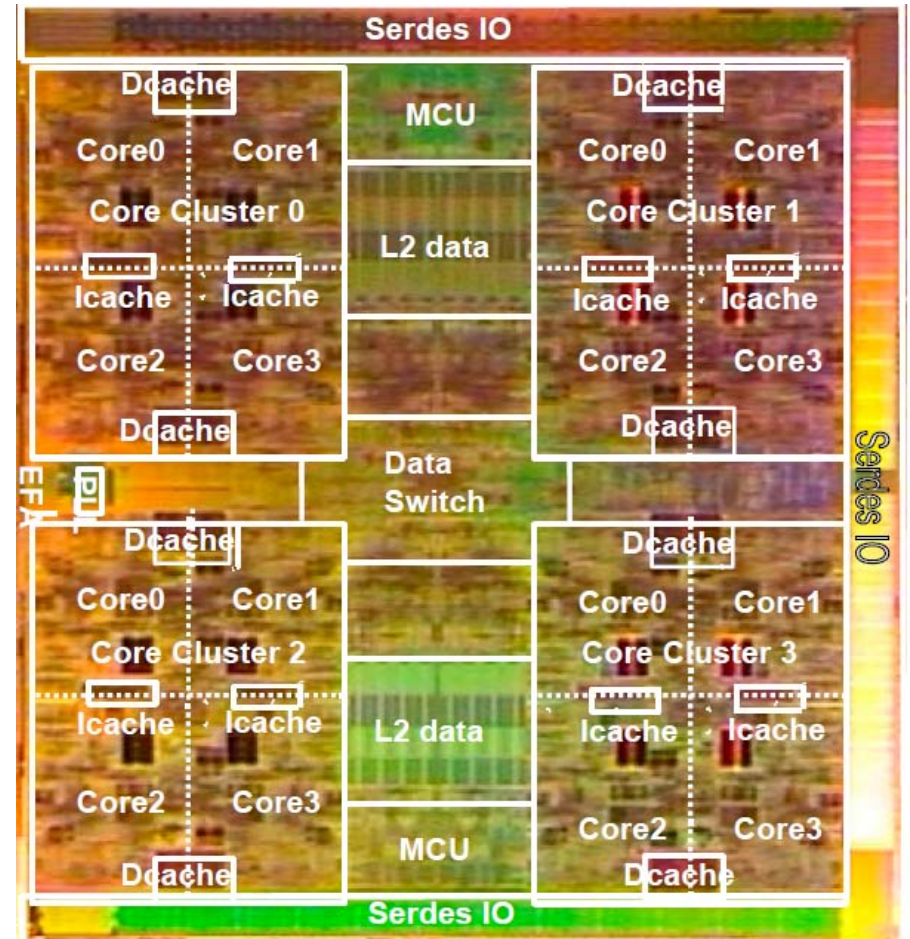
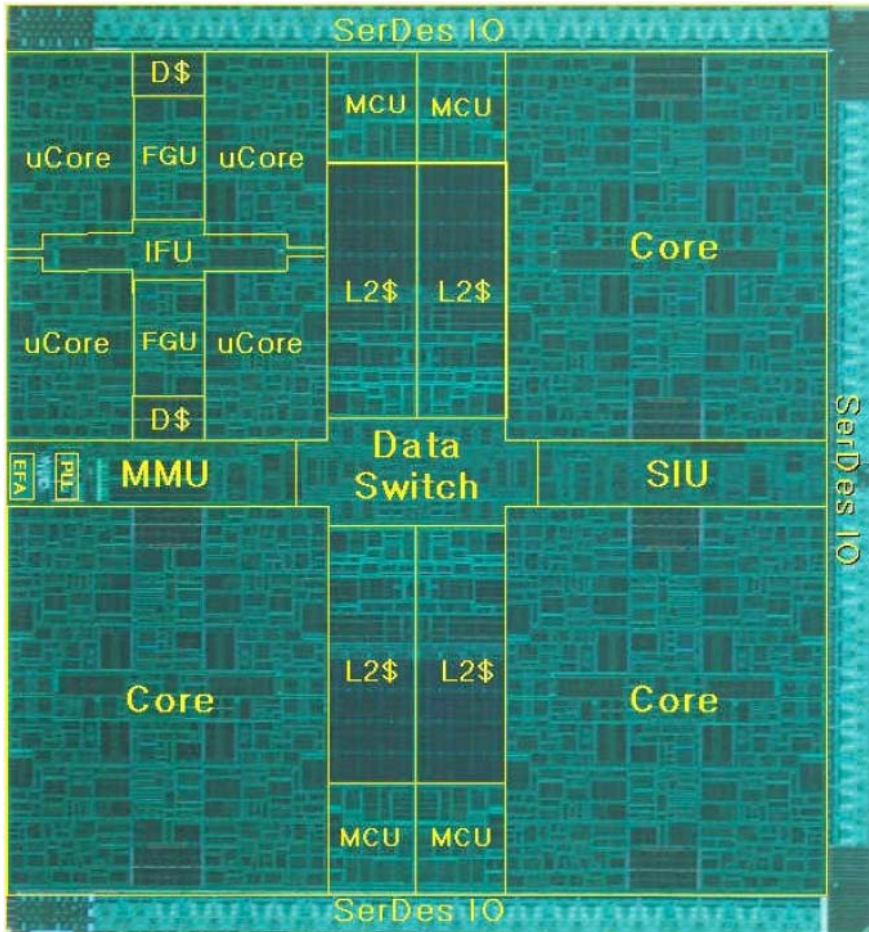
Übersicht SUN Rock – Features

Eckdaten

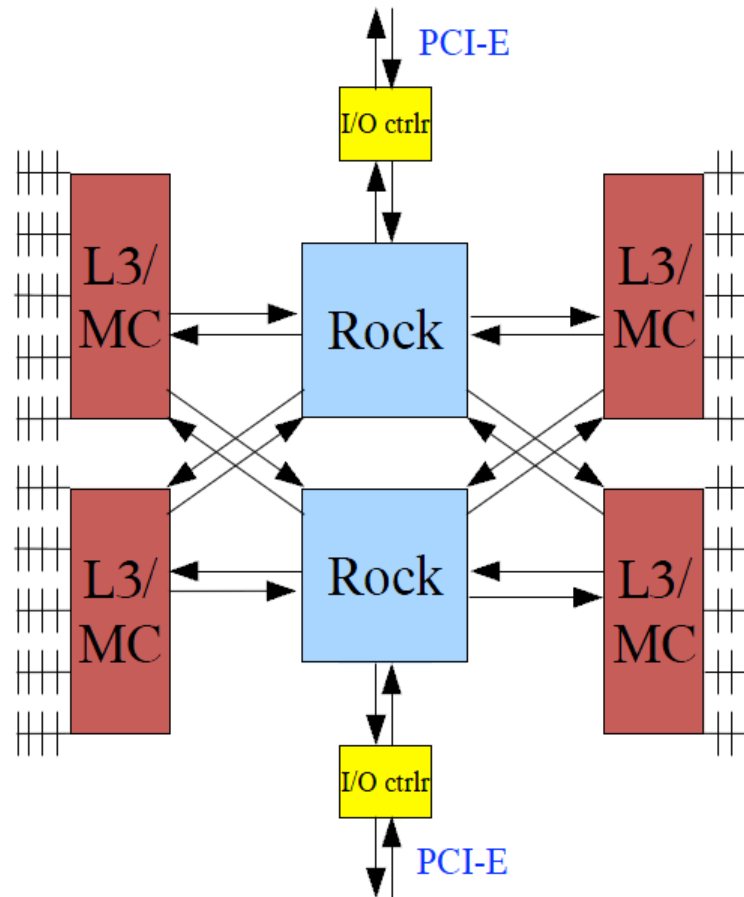
Cores	4
Microcores (µCore)	16
L1 Data Cache	8 x 32KB
L1 Instruction Cache	4 x 32KB
L2 Cache	2 MB
L3 Cache (nicht on-chip)	16 MB / Chip
Threads / µCore	1 oder 2
Threads / Chip	16 bis 32
I/O Durchsatz (Peak)	8 GB/s
Memory Durchsatz (Peak)	48 GB/s
Chipfläche (65nm)	396 mm²
Taktfrequenz	2.1 GHz
Gatter	5.5 Million
Flop Count	1.1 Million
Transistoren	321 Million
Power	250W



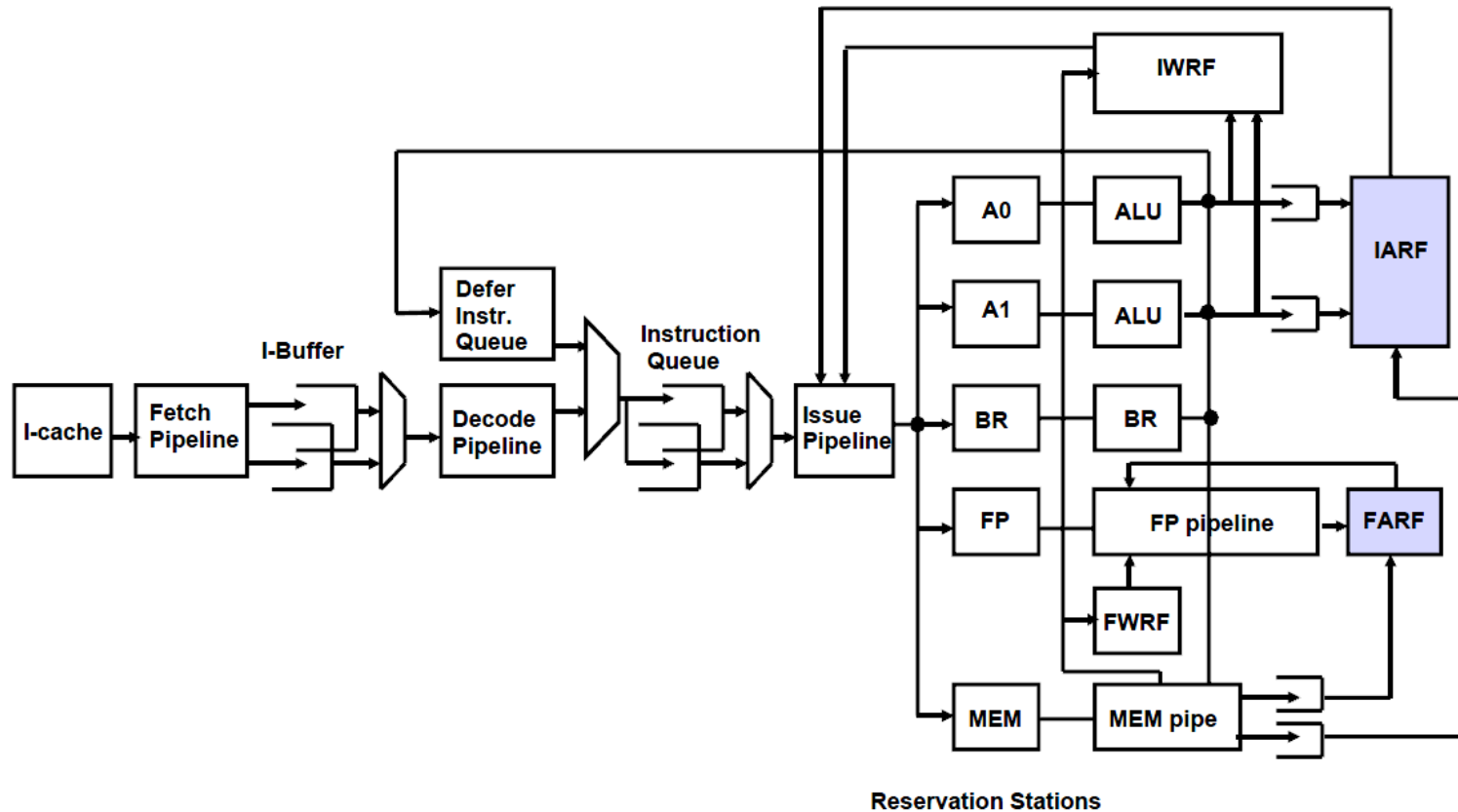
Chip



Mehrprozessortopologie



Spekulative Ausführung



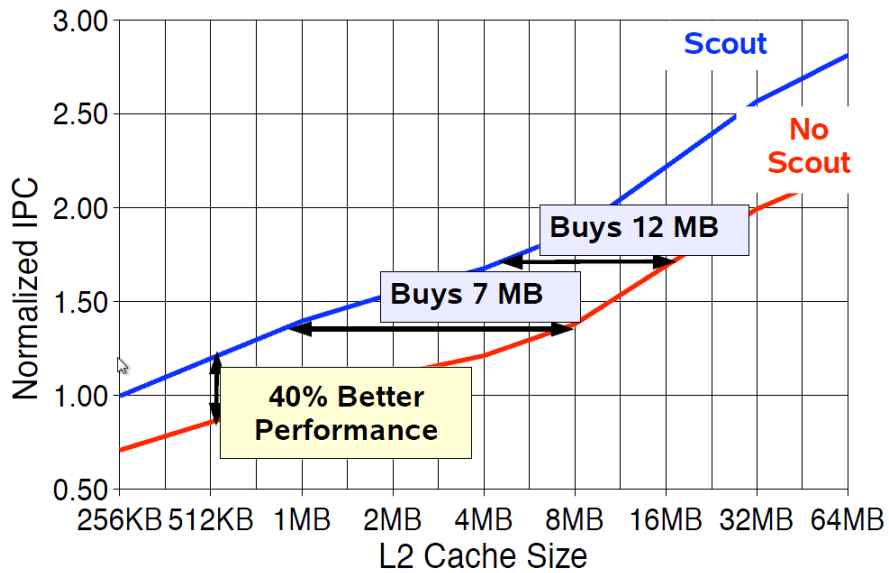
Scout Threading

- Idee:
- lange andauernde Instruktionen (*L1 Data-Cache Miss, TLB Miss, Division, ...*) starten für SW transparenten HW-Scout
 - HW-Scout erledigt *Ld/St-Prefetch* („Memory Level Parallelism“ MLP) und startet *Branch Prediction*
 - lange andauernde Instruktionen wird fertig, Ausführung wird fortgesetzt

- Umsetzung:
- 2 Kopien jedes Registers (eingefrorener *Checkpoint* + *Arbeitsregister* mit den spekulativen Updates)
 - Register haben sog. „NotThere“-Bit
 - Checkpoint einfrieren latenzfrei (0-cycle)
 - Rückkehr zum Code nach dem Checkpoint hat allerdings refetch-Latenz

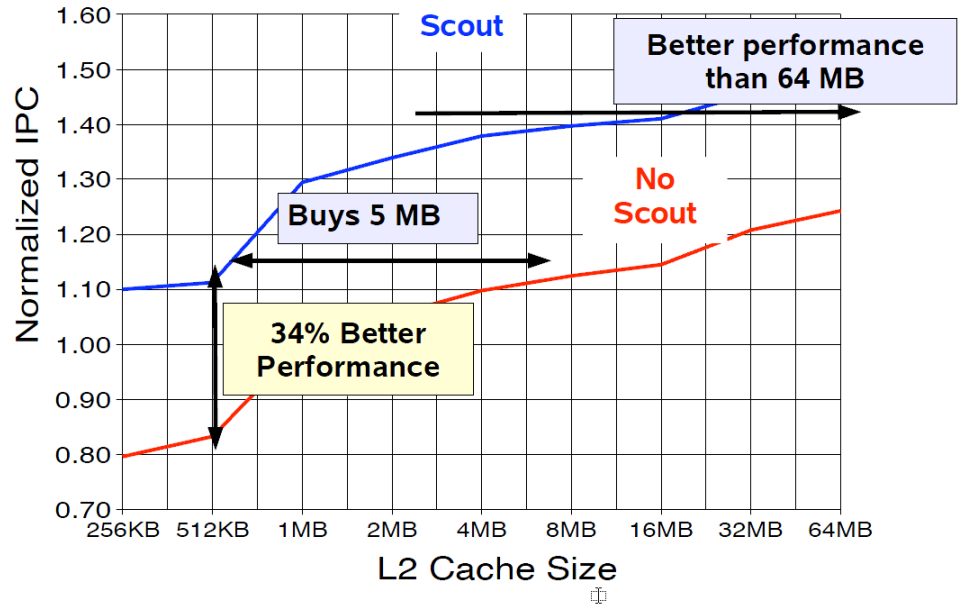
Scout Threading

Scouting Improvement - Database



Source: ROCK Simulator running TPCC traces

Scouting Improvement - SPECfp2000



Source: ROCK Simulator running SPECfp traces

Execute Ahead (EXE)

- lange andauernde Instruktionen starten für SW transparenten *Execute Ahead* Ausführungsstrang
- Verschiebeschlange („deferred queue“ DQ) für abhängige Instruktionen
- verbleibende Instruktionen durch Spekulation, keine erneute Ausführung
- Verschiebeschlange nach Abschluss der lange andauernden Instruktion ausführen → alle erfolgreich: latenzfreier "join"; Fehlschlag: resume

Simultaneous Speculative Threading (SST)

- nur noch ein Thread pro μ Core
- entspricht Execute Ahead bis Abschluss lange andauernde Instruktion, dann:
ein Ausführungstrang arbeitet DQ ab **„*behind strand*“**
ein Ausführungstrang bereits weiter spekulativ **„*ahead strand*“**

Transactional Memory

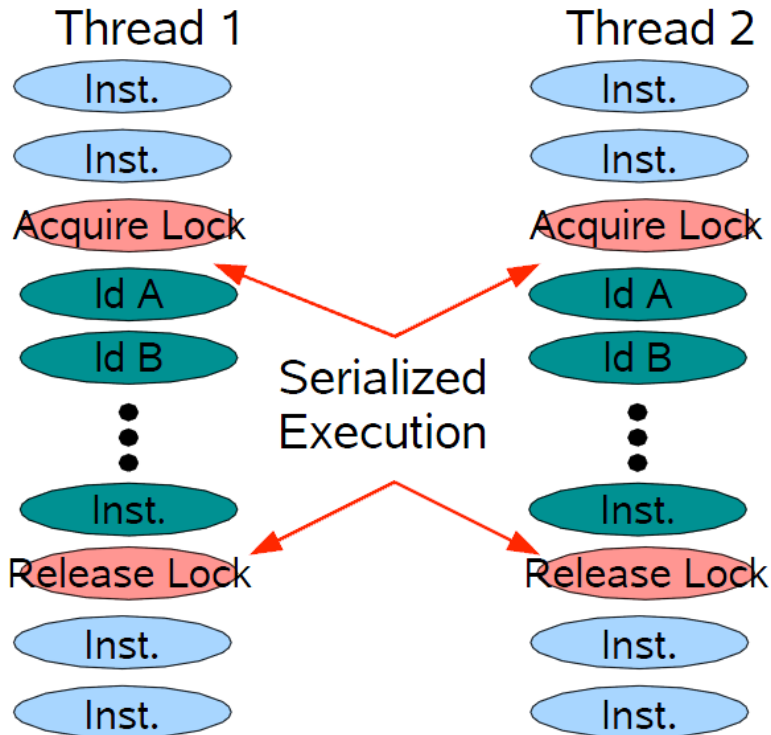
- „best effort“ HTM (Hardware Transactional Memory) → kleine Transaktionen effizient in HW ausführen
- von SW angestoßen, realisiert über 2 neue Instruktionen
 - „chkpt“ (Start einer Transaktion mit fail_PC zur Stelle, an der bei Fehlschlag fortgesetzt werden soll)
 - „commit“ (Transaktion erfolgreich abgeschlossen)
 - „cps“ (auslesbares Checkpoint-Status-Register mit Ursacheninformation)

Name	Syntax
Checkpoint	chkpt <fail pc>
Commit	commit
Read Checkpoint	Status rd %cps, <dest reg>

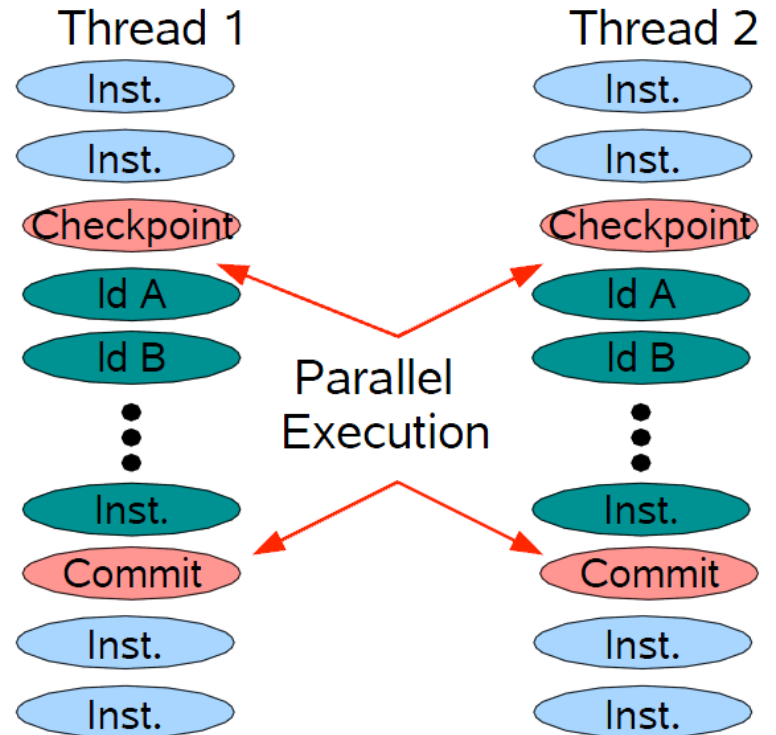
- hybrider Ansatz: HW-Transaktion wenn möglich, sonst SW
- L2-Cache puffert Store-Operationen u. -Konflikte bis Commit, dann Lock der Zeilen bis Update (→ L2-Cache begrenzt Komplexität)

Transactional Memory

Beispiel für **Lock**:

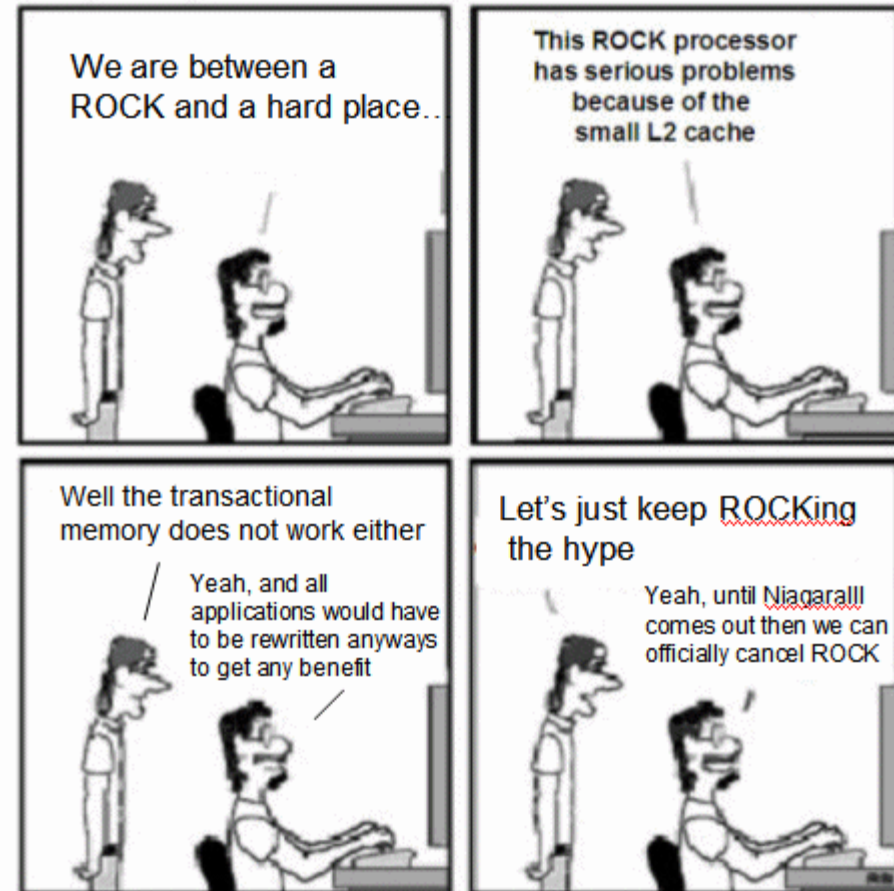
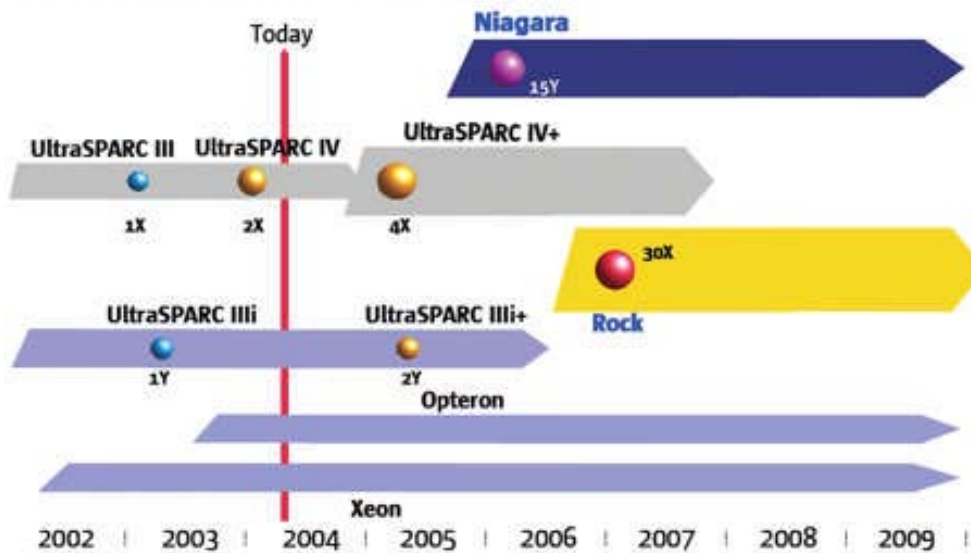


Beispiel für **Transactional Memory**:



Ausblick/Kritik

Sun CPU Roadmap



Quellen

http://de.wikipedia.org/wiki/Sun_SPARC

<http://research.sun.com/scalable/pubs/TRANSACT2008-ATMTP-Apps.pdf>

<http://www.opensparc.net/pubs/preszo/08/RockISSCC08.pdf>

<http://www.opensparc.net/pubs/preszo/08/RockHotChips.pdf>

<http://www.unine.ch/transact08/slides/Moir-Adaptive.pdf>

<http://www.geocities.com/sunrockchip/>