Parallele Rechnerarchitektur III

Stefan Lang

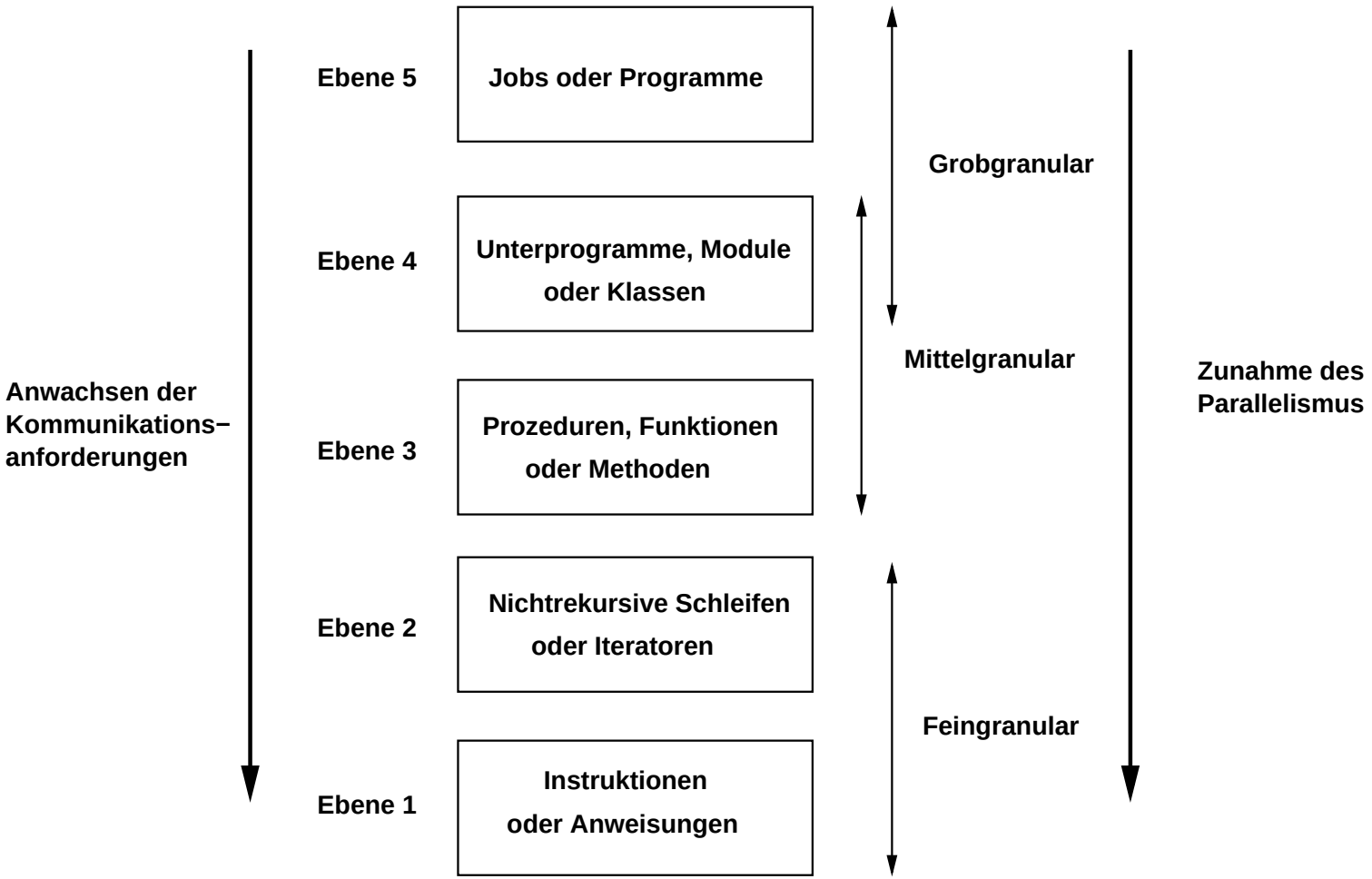
Interdisziplinäres Zentrum für Wissenschaftliches Rechnen
Universität Heidelberg
INF 368, Raum 532
D-69120 Heidelberg
phone: 06221/54-8264
email: Stefan.Lang@iwr.uni-heidelberg.de

WS 12/13

Parallele Rechnerarchitektur III

- Parallelität und Granularität
- Graphikkarten
- IO
- Detailstudie Hypertransport

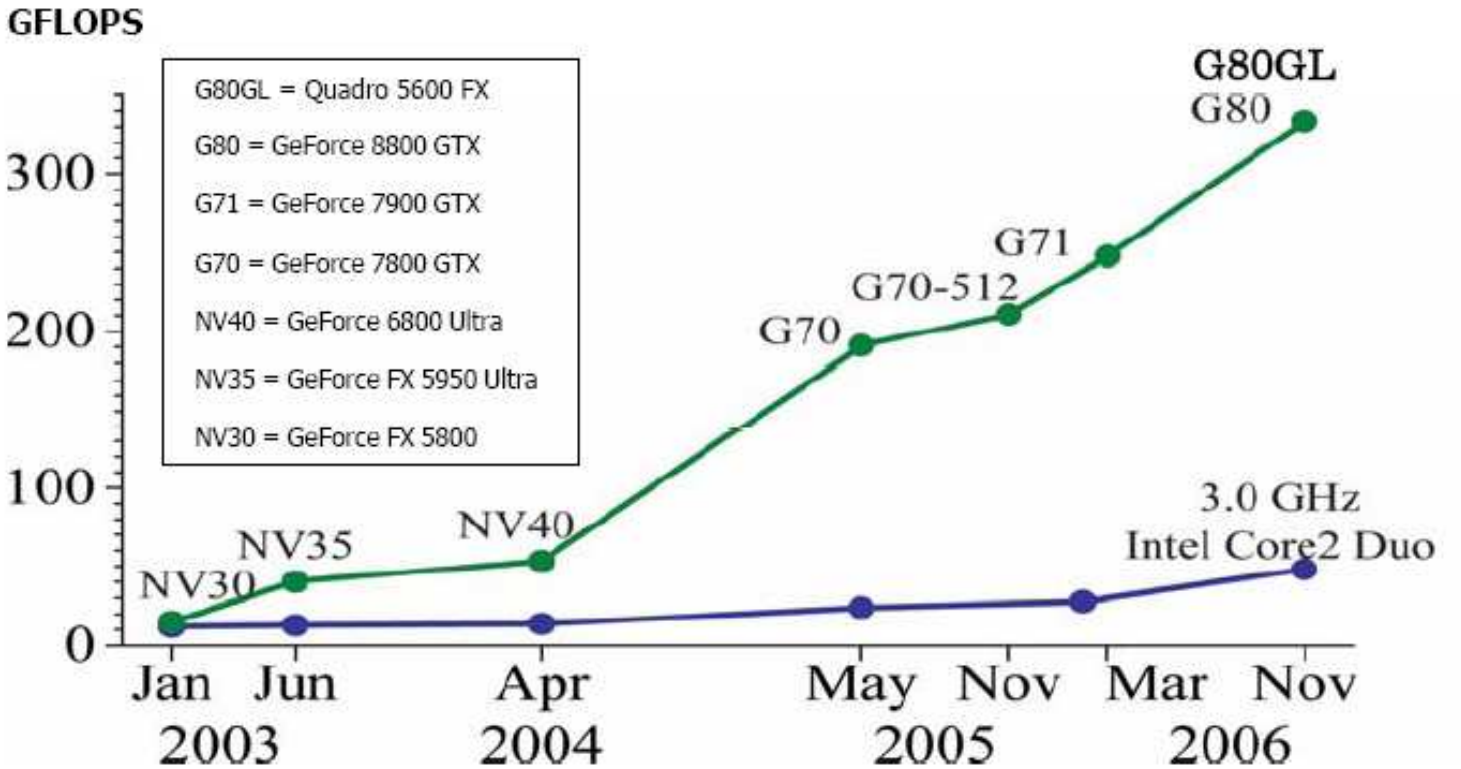
Parallelität und Granularität



Graphikkarten

- GPU = Graphics Processing Unit
- CUDA = Compute Unified Device Architecture
  - Tookit von NVIDIA zur direkten GPU Programmierung
  - Programmierung einer GPU ohne graphische API
  - GPGPU
- im Vergleich zu CPUs weitaus höhere Rechenleistung und Speicherbandbreite
- GPUs sind billig und breit etabliert

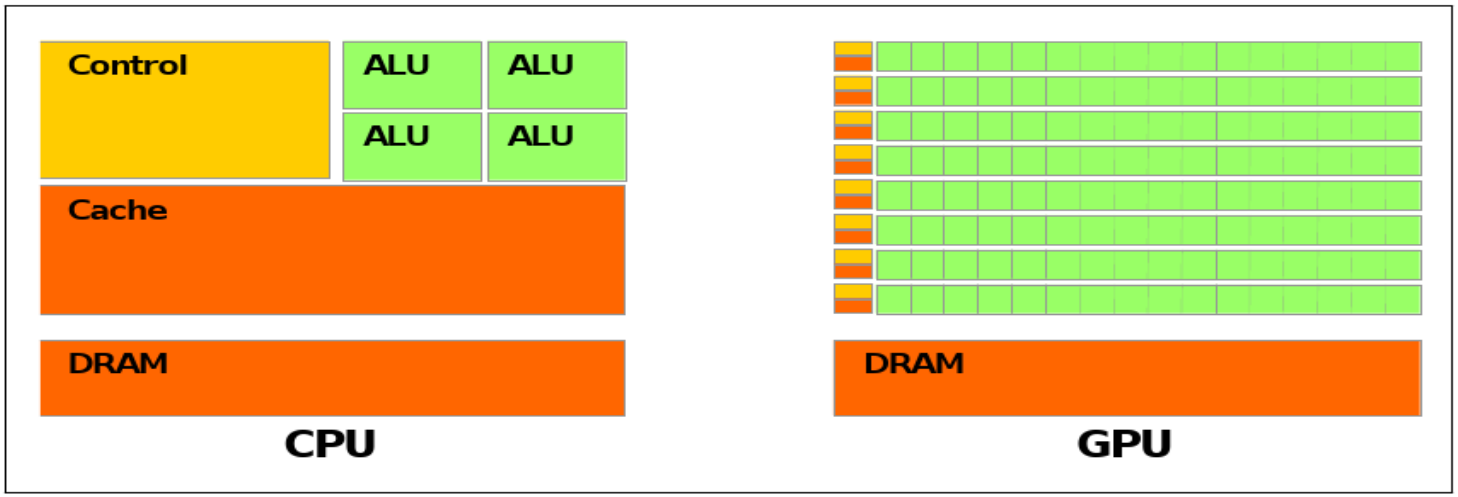
Leistungsentwicklung: CPU vs. GPU



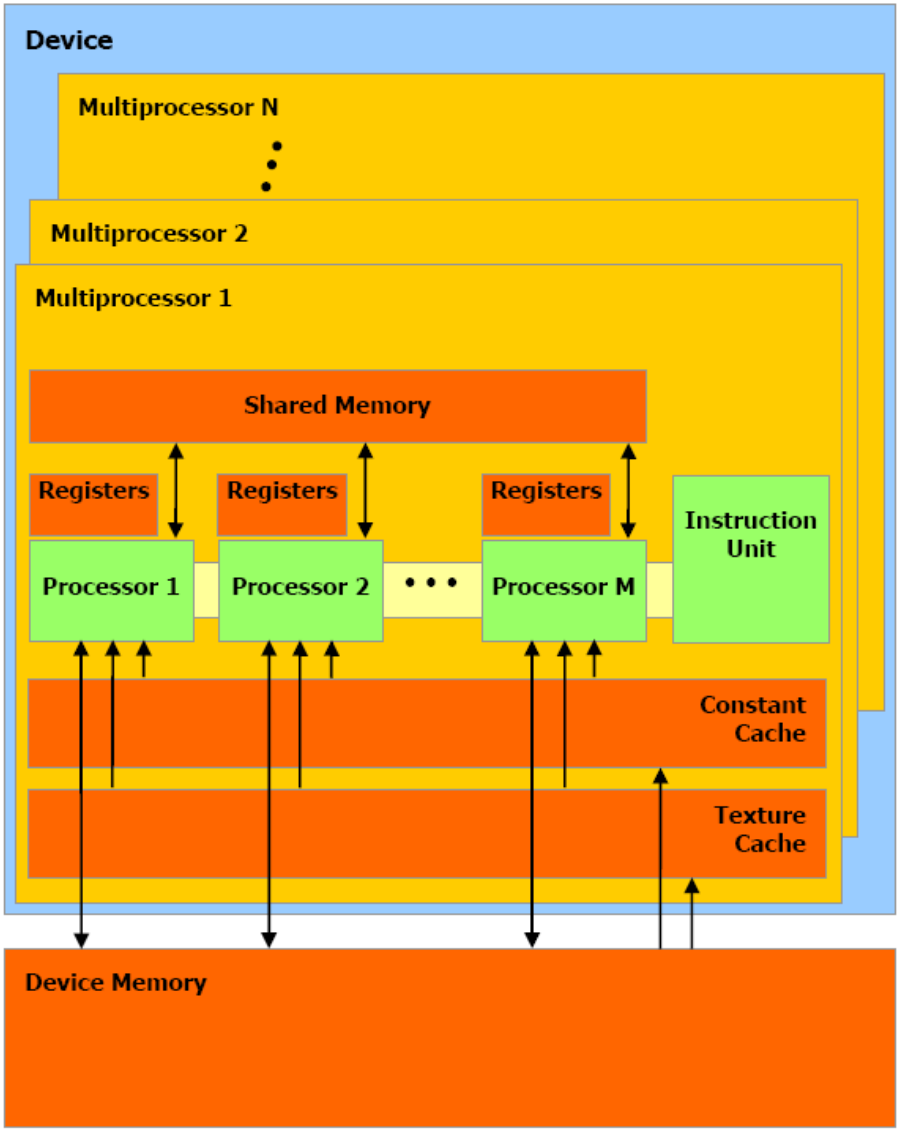
Graphikkarte: Hardwaredaten

GeForce GTX 285	
Fab (nm)	55
Transistors (million)	1400
Memory (MB)	1024
Multiprocessors	30
Streaming Processors	240
Shader Clock (MHz)	1476
Memory Bandwidth (GB/s)	159
Memory Bus width (bit)	512
SP GFLOPs (MADD + MUL)	1063
DP GFLOPs (MADD)	89
TDP (Watt)	183
Price (EUR)	~350

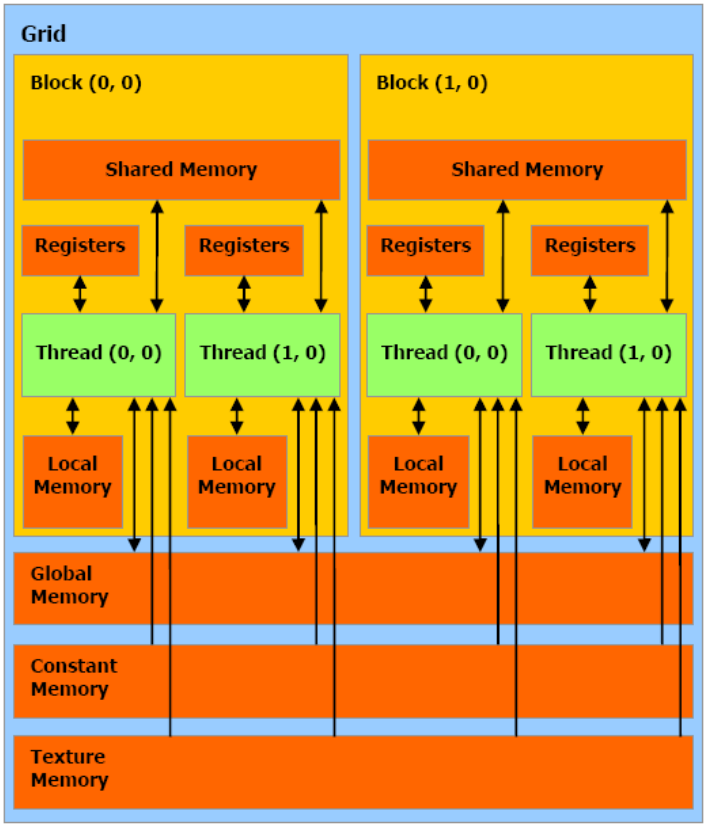
Chiparchitektur: CPU vs. GPU



Graphikkarte: Hardwareaufbau

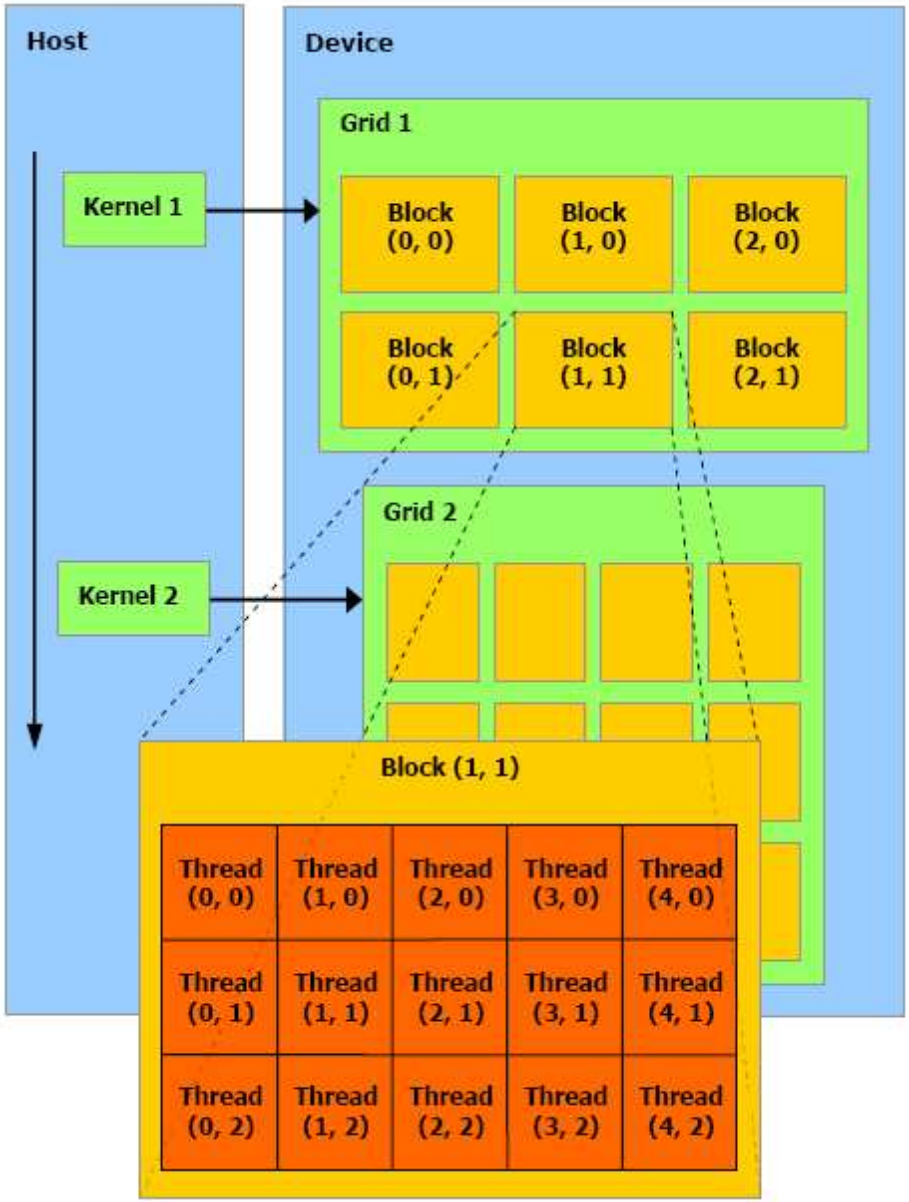


Graphikkarte: Speicheraufbau

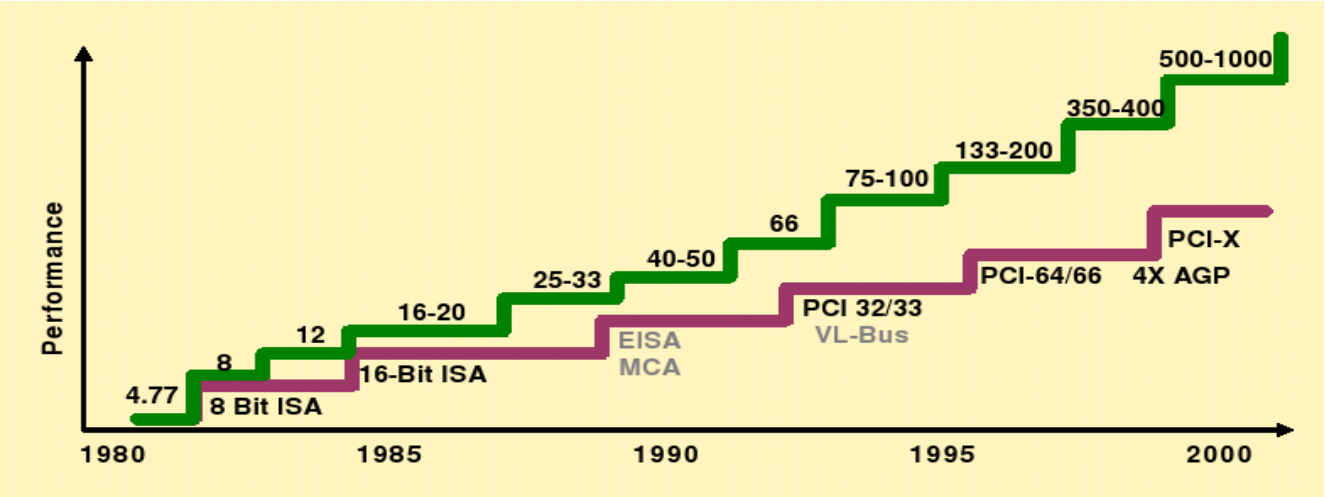


- 8192 Register (32-bit), ingesamt 32KB pro Multiprozessor
- 16KB schnelles shared memory pro Multiprozessor
- Grosses globales memory (Hunderte von MB, e.g. 512MB-4GB)
- Globales memory ungecacht, Latenzzeit 400-600 Taktzyklen
- Lokales memory ist eigentlich Teil vom globalen Speicher
- Read-only konstanter Speicher
- Read-only Texturspeicher
- Register und shared memory werden zwischen Blöcken, welche auf einem Multiprozessor ausgeführt werden, aufgeteilt
- Globaler Speicher, konstanter Speicher und Texturspeicher sind von der CPU zugreifbar

Graphikkarte: Programmiermodell



Leistungsentwicklung im Bereich des I/O

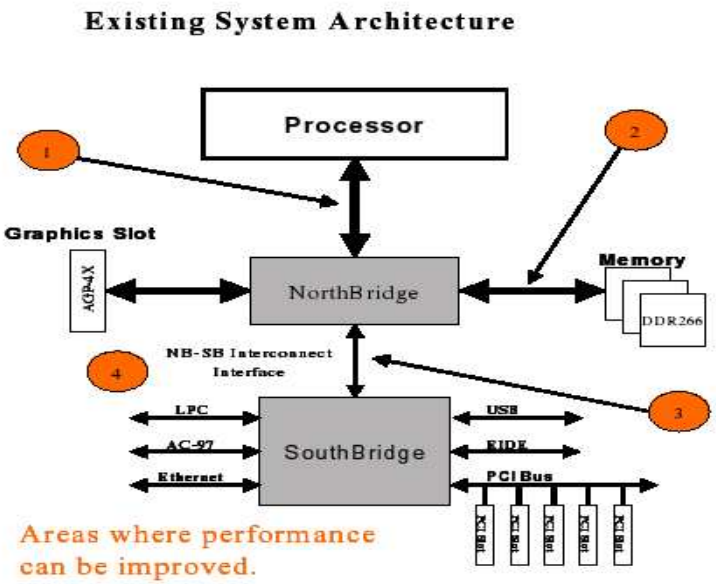


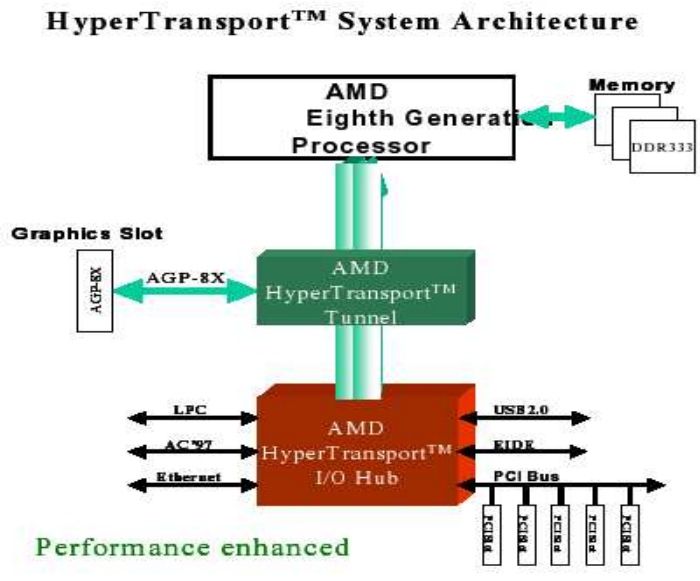
- Mikroprozessor Leistung verdoppelt sich in ca. 18 Monaten
- Leistung der I/O Architektur verdoppelt sich in ca. 36. Monaten
- Server und Workstations benötigen verschiedene Hochgeschwindigkeitsbusse (PCI-X, PCI-E, AGP)

→ hohe Komplexität und unnötige Vielfalt bei eingeschränkter Leistung

- Anforderungen an Bandbreite wachsen: Hochauflösende 3D-Graphik und Video (CPU – Graphikprozessor), Interprozessor (CPU – CPU), Hochleistungsnetzwerke wie Gigabit Ethernet und Infiniband (CPU – I/O)

Engpäße im I/O-Bereich





Mögliche Engpäße sind:

- Front-Side Bus
- Speicherinterface
- Chip-to-Chip Verbindung
- I/O zu anderen Bussystemen

Standardisierung von I/O Anforderungen

Entwicklung der Hypertransport (HT) I/O Verbindungsarchitektur (seit 1997)

- HyperTransport ist In-The-Box Lösung (Intraconnect Technologie)
- komplementäre Technik zu Netzwerkprotokollen wie Infiniband und 10 Gb/ 100 Gb Ethernet als Box-to-box Lösungen (Interconnect Technologie)
- HT ist offener Industriestandard, kein Produkt (keine Lizenzgebühren)
- Weiterentwicklung und Standardisierung durch Konsortium von Industriepartnern, z.B. AMD, Nvidia, IBM, Apple
- früherer Kodename: Lightning Data Transfer (LDT)

Funktionsüberblick

Feature/Function	HyperTransport Technology
Bus Type	Dual, unidirectional, point-to-point links
Link Width	2, 4, 8, 16, or 32 bits
Protocol	Packet-based, with all packets multiples of four bytes (32 bits). Packet types include Request, Response, and Broadcast, any of which can include commands, addresses, or data.
Bandwidth (Each Direction)	100 to 6400 Mbytes/s
Data Signaling Speeds	400 MHz to 1.6 GHz
Operating Frequencies	400, 600, 800, 1000, 1200, and 1600 Megatransfers/second
Duplex	Full
Max Packet Payload or Burst Length	64-byte packet
Power Management	ACPI-compatible
Signaling	1.2-V Low-Voltage Differential Signaling (LVDS) with a 100-ohm differential impedance
Multiprocessing Support	Yes
Environment	Inside the box
Memory model	Coherent and noncoherent

Designziele bei der Gestaltung von HyperTransport

- Verbesserung der Systemleistung
  - höhere I/O Bandbreite (high bandwidth)
  - Vermeidung von Flaschenhälsen durch langsame Geräte in kritischen Informationspfaden
  - geringere Anzahl von Systembussen
  - niedrige Antwortzeiten (low latency)
  - reduzierter Energieverbrauch
- Vereinfachung des Systemaufbaus
  - Einheitliches Protokoll für In-box Verbindungen
  - Nutzung kleiner Pinzahlen um hohe Packungsdichte und niedrige Kosten sicherzustellen
- Höhere I/O-Flexibilität
  - Modulare Brückenarchitektur
  - Unterschiedliche Bandbreiten in Upstream/Downstream Richtung
- Kompatibilität mit bestehendem Systemen
  - Ergänzung zu standardisierten, externen Bussen
  - geringe Auswirkungen auf bestehende Betriebssysteme und Treiber
- Erweiterbarkeit zu Systemnetzwerk Architekturen (SNA Busse)
- Hohe Skalierbarkeit in Mehrprozessorsystemen

Flexible I/O Architektur

Hypertransport Architektur ist in 5 Schichten unterteilt
Struktur orientiert sich am **Open-Sytem-Interconnection** (OSI) Referenzmodell

- **Bitübertragungsschicht:** Physikalische und elektrische Eigenschaften von Daten, Kontroll, Taktleitungen
- **Datenverbindungsschicht:** Initialisierung und Konfiguration von Verbindungen, periodisch zyklische Redundanz (CRC), Trennung/Wiederverbindung, Paketierung Kontrollfluß und Fehlermanagement,
- **Protokollschicht:** Virtuelle Kanäle und Kommandos
- **Transaktionsschicht:** Schreibe- und Leseaktionen unter Nutzung der Datenverbindungsschicht
- **Sitzungsschicht:** Power-, Interrupt- und Systemmanagement

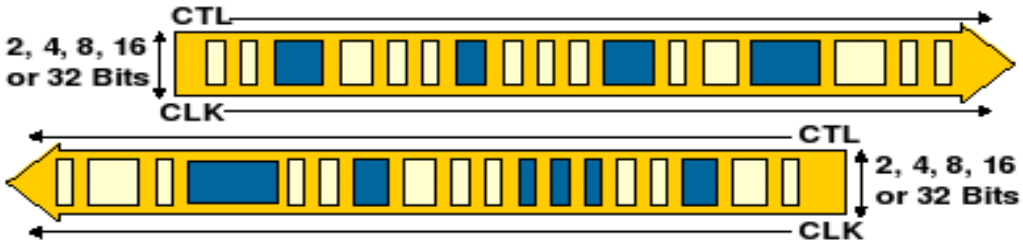
Geräteklassen und -konfigurationen

3 Geräteklassen werden hinsichtlich Funktion und Lage innerhalb der HT-Kette unterschieden: Cave, Tunnel und Bridge

- HT Bridge: Vermittler zwischen primärer Seite (CPU bzw. Speicher) und sekundärer Seite (HT-Geräten) einer Kette
- HT Tunnel: besitzt zwei Seiten mit jeweils einer Empfangs- und einer Sendeeinheit, z.B. Netzwerkkarte oder Bridge zu weiteren Protokoll
- HT Cave: markiert Ende der Kette und besitzt nur eine Kommunikationsseite. Durch die Verschaltung von mindestens einer HT Bridge und einem HT Cave kann eine einfache HT-Kette aufgebaut werden.

Bitübertragungsschicht I

Aufbau einer HT-Verbindung



- zwei unidirektionalen Punkt-zu-Punkt Datenpfade
- Datenpfadbreiten: 2,4 8 und 16 Bit je nach Geräteanforderungen
- Kommandos, Adressen und Daten (CAD) verwenden die selben Signalleitungen
  $\rightarrow$ geringere Leitungsanzahl, kleinere Pakete, geringerer Energieverbrauch, bessere thermische Eigenschaften
- Pakete enthalten CAD und sind Vielfache von 4 Byte (32 bit)
- Verbindungen mit weniger als 32 bit verwenden aufeinanderfolgende Takte um Pakete vollständig zu übertragen

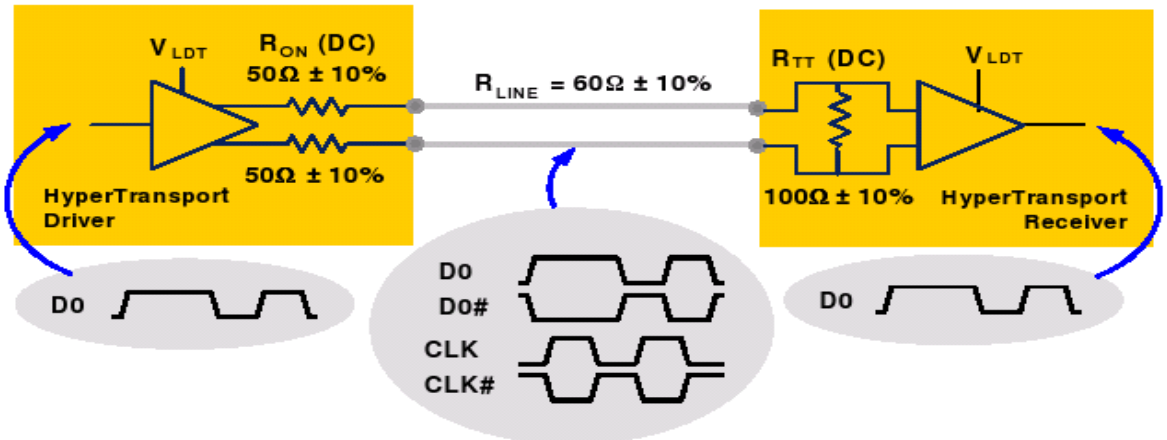
Hohe Performanz und Skalierbarkeit

- hohe Datenraten aufgrund von low voltage differential signaling (LVDS, 1,2V $\pm$ 5%)
- skalierbare Bandbreite mittels Skalierung von Übertragungsfrequenz und Linkbreite

Bitübertragungsschicht II

Hypertransport Low Voltage Differential Signaling

- Differenzielle Signalübertragung über zwei Leitungen: Spannungsdifferenz ($\pm$ 0,3 V) entspricht Logikzustand Logikwechsel durch Umpolen der Leitungen (symmetrische Signalübertragung, double-ended)
- HT Signalübertragung entspricht erweitertem IEEE LVDS Standard (1,2 V statt 2,5 V)
- 60 cm maximale Leitungslänge bei 800 Mbit/s
- Übertrager sind in Controllchips integriert, 100 Ω Impedanz verhindert Reflexionen
- einfach realisierbar auf Standard 4-Schicht PCBs (Printed Circuit Boards) und zukunftsfähig



Bitübertragungsschicht III

Elektrische Signale

Signal Name	Description	Comment
CAD	**Commands, Addresses and Data:** Carries command, address, or data information.	CAD width can be different in each direction.
CTL	**Control:** Used to distinguish control packets from data packets.	
CLK	**Clock:** Forwarded clock signal.	Each byte of CAD has a separate clock signal. Data is transferred on each clock edge.
PWROK	**Power OK:** Power and clocks are stable.	Single-ended.
RESET#	**HyperTransport Technology Reset:** Resets the chain.	Single-ended.
LDTSTOP#	**HyperTransport Technology Stop:** Enables and disables links during system state transitions.	Used in systems requiring power management. Single-ended.
LDTREQ#	**HyperTransport Technology Request:** Requests re-enabling links for normal operation.	Used in systems requiring power management. Single-ended.

Für je 8 Bit Datenbreite gibt es eine Taktleitung, welche vom Sender zum Empfänger verläuft, mit der die Daten auf den 8 Datenleitungen beim Empfänger abgetastet werden (quell-synchrone Taktung)
$\rightarrow$ Abweichungen vom Sendertakt werden minimiert

Bitübertragungsschicht IV

Bandbreitenskalierung

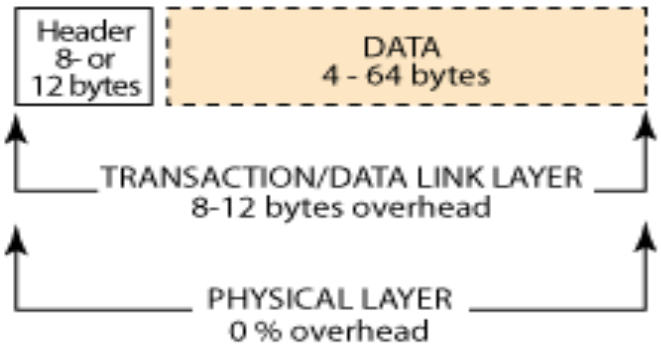
- Datenübertragung von CAD bei steigender und fallender Flanke des Taktsignals (DDR)
  $\rightarrow$ Verbindungstakt von 800 MHz entspricht 1600 MHz Datentakt
- Anzahl der Leitungen ermöglicht Anpassung an Bandbreitenerfordernisse
  $\rightarrow$ 2 x 16 CAD bits + 800 GHz clock = 2 x 3,2 GByte Bandbreite (103 Pins)
  $\rightarrow$ 2 x 2 CAD bits + 400 MHz clock = 2 x 200 MByte Bandbreite (24 Pins)

Link Width (Each Way)	2	4	8	16	32
Data Pins (total)	8	16	32	64	128
Clock Pins (total)	4	4	4	8	16
Control Pins (total)	4	4	4	4	4
Subtotal (High Speed)	**16**	**24**	**40**	**76**	**148**
V_{LDT}	2	2	3	6	10
GND	4	6	10	19	37
PWROK	1	1	1	1	1
RESET#	1	1	1	1	1
Total Pins	**24**	**34**	**55**	**103**	**197**

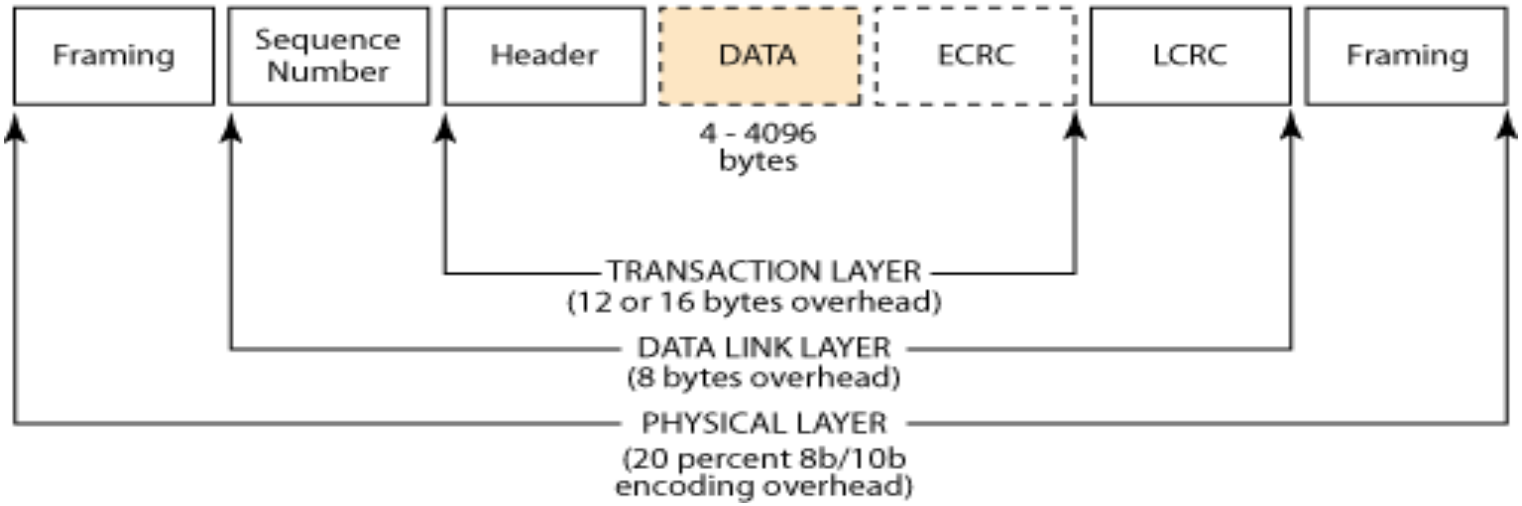
Verbindungsschicht

Packetaufbau von HT im Vergleich zu PCI-E

HyperTransport Packet Format



PCI Express Packet Format



Protokoll- und Transaktionsschicht

- Protokollschicht: Kommandos, virtuelle Kanäle und Flußkontrolle
- Transaktionsschicht: Durchführung von Aktionen wie Leseanforderungen und Antworten

Kommandoübersicht

Virtual Channel	Command	Comment
Posted	Posted Write	Followed by data packet(s).
	Broadcast	Issued by host bridge downstream to communicate information to all devices.
	Fence	All posted requests in a stream cannot pass it.
Non-Posted	Non-Posted Write	
	Read	Designates whether response can pass posted requests or not.
	Flush	Forces all posted requests to complete.
	Atomic Read-Modify-Write	Generated by I/O devices or bridges and directed to system memory controlled by the host.
Responses	Read Response	Response to read command, is followed by data packet(s).
	Target Done	A transaction not requiring returned data has completed at its target.

Paketstruktur

Bit-Time	7	6	5	4	3	2	1	0
0	SeqID[3:2]		Cmd[5:0]					
1	PassPW	SeqID[1:0]	UnitID[4:0]					
2	*Command-Specific*							
3	*Command-Specific*							
4	Addr[15:8]							
5	Addr[23:16]							
6	Addr[31:24]							
7	Addr[39:32]							

Paketweiterleitung

Weiterleitungsmethoden der HT-3.0 Spezifikation

- Store-and-Forward Routing: Ein Paket wird vollständig empfangen und zwischengespeichert, die Checksumme (CRC) berechnet und mit der im Paket verglichen, der Paketheader wird zur Empfängerermittlung dekodiert, dann wird das Paket entweder selbst verarbeitet oder über die Verbindung in Richtung des Empfängers weitergeleitet (1 hop)
- Cut-Through Routing: Weiterleitung des Pakets sobald der Paketheader empfangen und dekodiert ist, das Paket wird ohne Zwischenspeicherung durchgeleitet
  Problem: weitergeleitete Pakete mit CRC-Fehler Lösung: Abbruch der Weiterleitung, Empfänger ermittelt Paketfehler und verwirft das Paket
- Spekulative Weiterleitung: Spekulation auf Korrektheit und Empfängerport, nur noch Feststellen ob korrektes Kommando im Paket vorliegt (1 Byte)
- Spekulative Weiterleitung mit aggr. Implementierung: Paket wird sofort, also mit dem ersten Leitungstakt und ohne irgendeine Dekodierung weitergeleitet

Alle Methoden können ohne Veränderung der Verbindungsspezifikation implementiert werden!

Paketweiterleitung

Paket mit n bytes (inkl. CRC), Linkbreite w bit, Pfadlänge d hops
t_{wire} Übertragungszeit, t_{proc} Zeit für CRC-Check + Empfängerermittlung
h zu empfangende Bytes bis zum Forwarding
Latenzzeit der verschiedenen Weiterleitungsmethoden

- Store-and-Forward Switching

$$L = d \cdot (n \cdot 8/w + t_{wire} + t_{proc})$$

- Cut-Through Switching

$$L = d \cdot (max(1, 8 * h/w) + t_{wire} + t_{proc}) + 8/w \cdot (n - max(h, w/8))$$

- Spekulative Weiterleitung

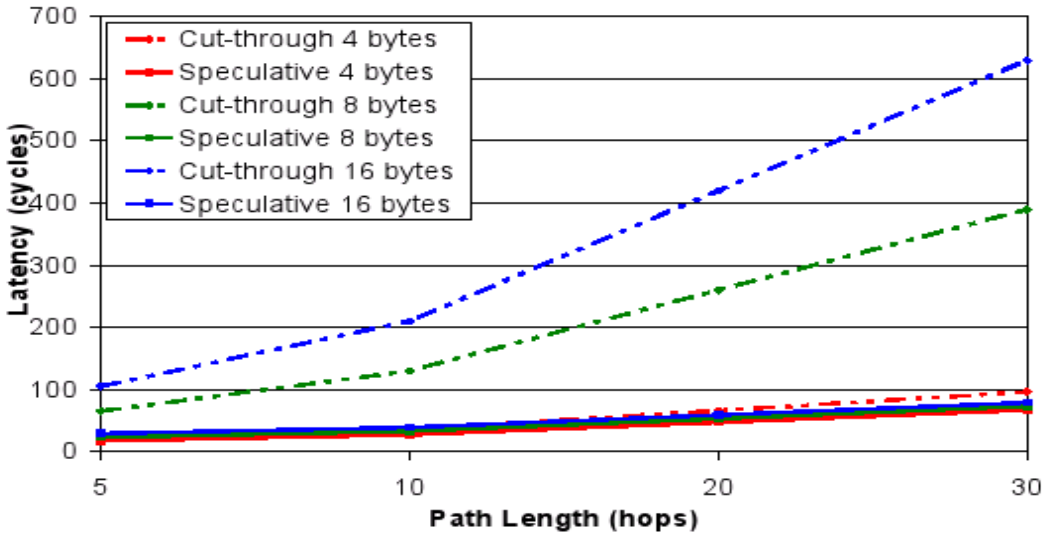
$$L = d \cdot (max(1, 8/w) + t_{wire} + t_{proc}) + 8/w \cdot (n - max(1, w/8))$$

- Spekulative Weiterleitung mit aggr. Implementierung

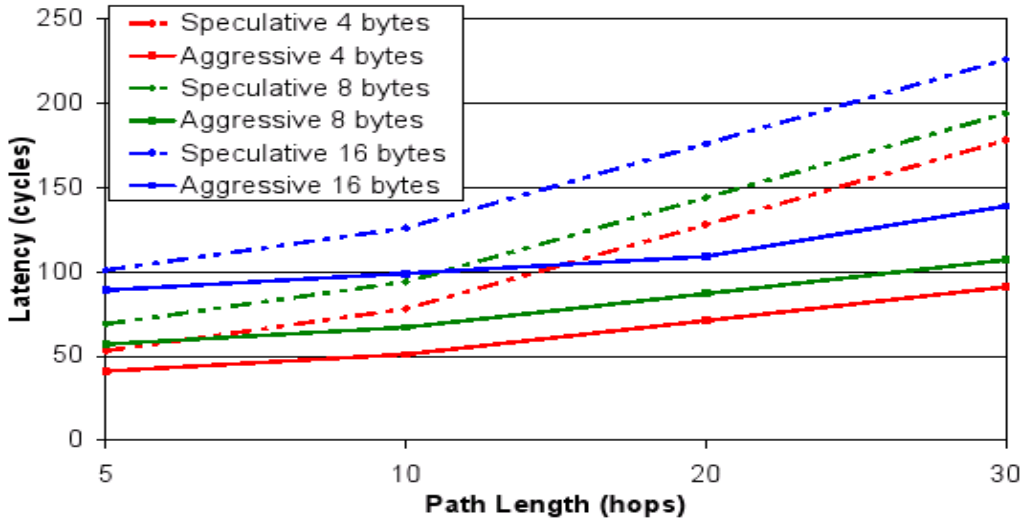
$$L = d \cdot (1 + t_{wire} + t_{proc}) + 8/w \cdot (n - 1)$$

Paketweiterleitung

Latenzzeit bei standard bzw. aggressiver spekulativer Weiterleitung



8-Bit Verbindungsbreite



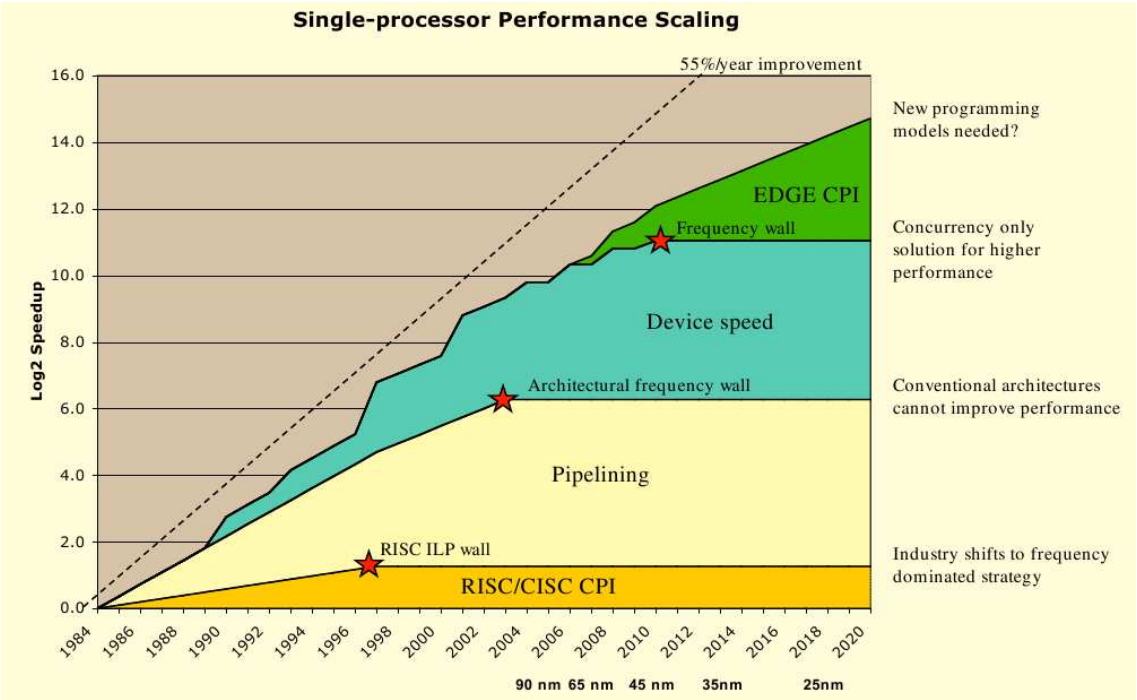
Innovative Architekturansätze

Zwei Entwicklungsrichtungen

- Design jenseits der RISC-Architektur
  - TRIPS
  - WaveCore
- Many-Core Prozessoren
  Many-Core Chip Entwicklung (chronologisch)
  - Intel Tera
  - Intel Larrabee
  - Tilera Tile64
  - Adapteva Epiphany

TRIPS Prozessor

- TRIPS = Tera-op, Reliable, Intelligently adaptive Processing System
- Forschungsprozessor (UTA, Texas)
- Prozessorarchitektur ermöglicht einfaches Hinzufügen weiterer Kerne
- Projektförderung durch u.a. IBM, Intel, DARPA, NSF
- EDGE-Architektur (Explicit Data Graph Execution) basierend auf Blöcken, welche elementare Anweisungen unabhängig voneinander ausführen, sowie datengesteuerte (out-of-order) Anweisungsausführung.
- Ziel: Realisierung von Prozessoren mit Multi-Teraflop Leistung



Aufbau der TRIPS Architektur I

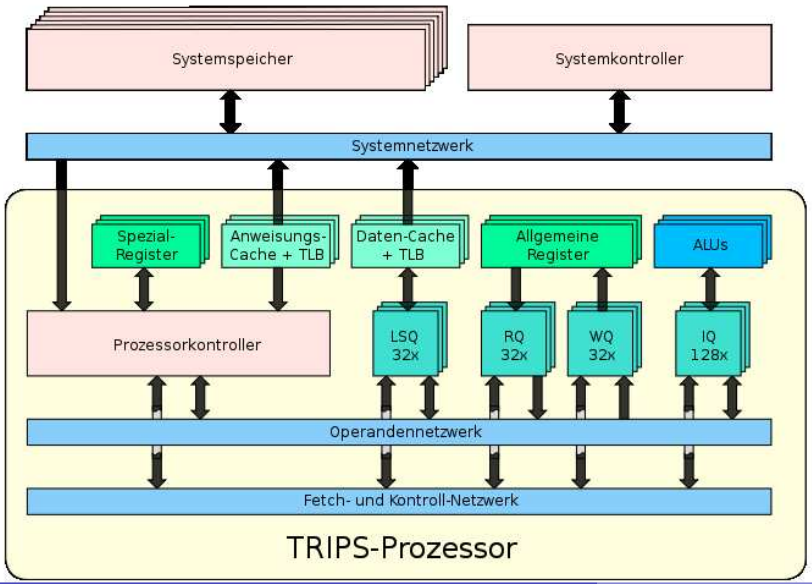
RISC-basierte Features

- Menge von arithmetisch-logischen Einheiten (ALU), Caches und Registern
- ALUs: Ganzzahl und Gleitkomma-Rechenoperationen
- getrennte Caches für Daten und Instruktionen
- mehrere Übersetzungspuffer (TLB), welche virtuelle auf physische Adressen abbilden
- Registerunterteilung in allgemeine und spezielle Register (Special Function Register, SFC):
  - allgemeine Register für beliebige Daten oder Adressen
  - Spezialregister zur Konfiguration und Kontrolle des Prozessorstatus

Aufbau der TRIPS Architektur II

TRIPS-eigene Konzepte

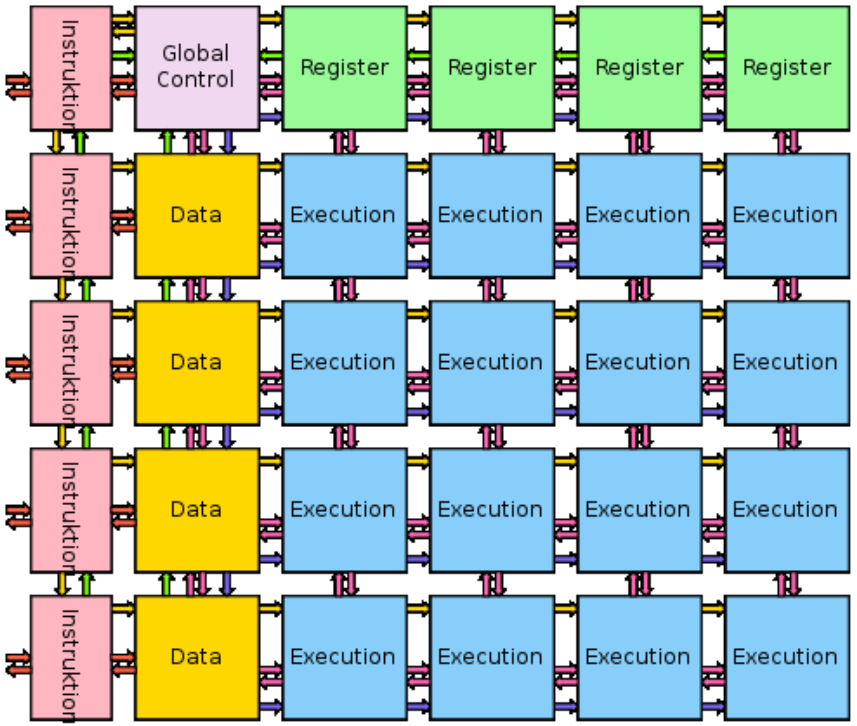
- Definition einer Menge interner Queues als Teil des Befehlssatzes und des Datenflussmodells
- Dies ermöglicht es eine Serie von Anweisungen als Block auszuführen, anstatt immer nur einzelne Befehle:
- Rest des TRIPS besteht aus einem systemweiten Netzwerk, welches die einzelnen Rechenblöcke miteinander verbindet
- Zugriffe von Prozessoren, die über dieses Netzwerk auf einen gemeinsamen Speicher zugreifen wollen, werden von einem Systemkontroller gesteuert.



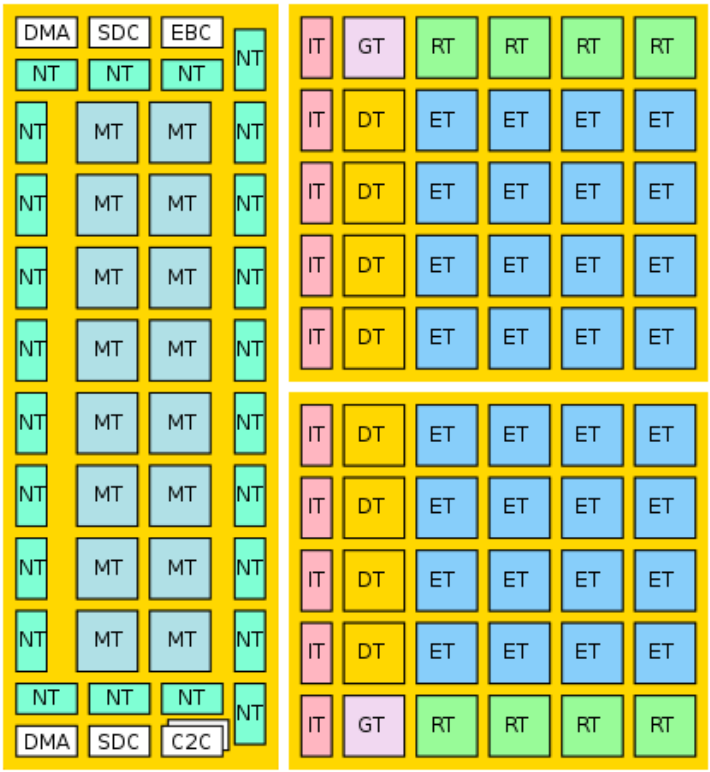
- Anweisungs-Queue (Instruktion-Queue, IQ) verarbeitet bis zu 128 Instruktionen gleichzeitig
- Lese-Queue (Read-Queue, RQ) puffert 32 Lesezugriffe auf allgemeine Register
- Schreib-Queue (Write-Queue, WQ) puffert bis zu 32 schreibende Zugriffe auf allgemeine Register
- zusätzliche Laden- und Halten-Queue (Load & Store Queue, LSQ) puffert 32 Speicherzugriffe

Implementierung der TRIPS Architektur I

- TRIPS-Prozessoren werden aus einzelnen Kacheln (Tiles) aufgebaut
- jedes Tile erfüllt eine elementare Funktion
- Die einzelnen Tiles werden in eine zweidimensionalen Anordnung (Array) gebracht.
- Man unterscheidet hierbei die folgenden Arten von Tiles:
  - Execution Tiles (ET) enthalten IQ und ALU.
  - Register Tiles (RT) enthalten allgemeine Register, RQ und WQ.
  - Data Tiles (DT) enthalten Daten-Cache, Daten-TLB und LSQ.
  - Instruction Tiles (IT) enthalten Anweisungs-Cache und Anweisungs-TLB.
  - Global Control Tile (GT) enthält die Spezialregister und die Logik des globalen Prozessorkontrollers.



Implementierung der TRIPS Architektur II



- Die meisten Prozessorresourcen werden zusätzlich in Bänke unterteilt und können somit auf mehrere Tiles verteilt sein
- TRIPS-Architektur bildet dadurch eine Grid-Architektur auf Prozessorebene
- Dies erlaubt hohe Taktfrequenzen, hohe Parallelität der Instruktionen und gute Erweiterbarkeit
- Nachteil: Hohe Latenzzeit in Grid-Architekturen, wenn Daten von einem Tile zu einem weit entfernten Tile gebracht werden müssen

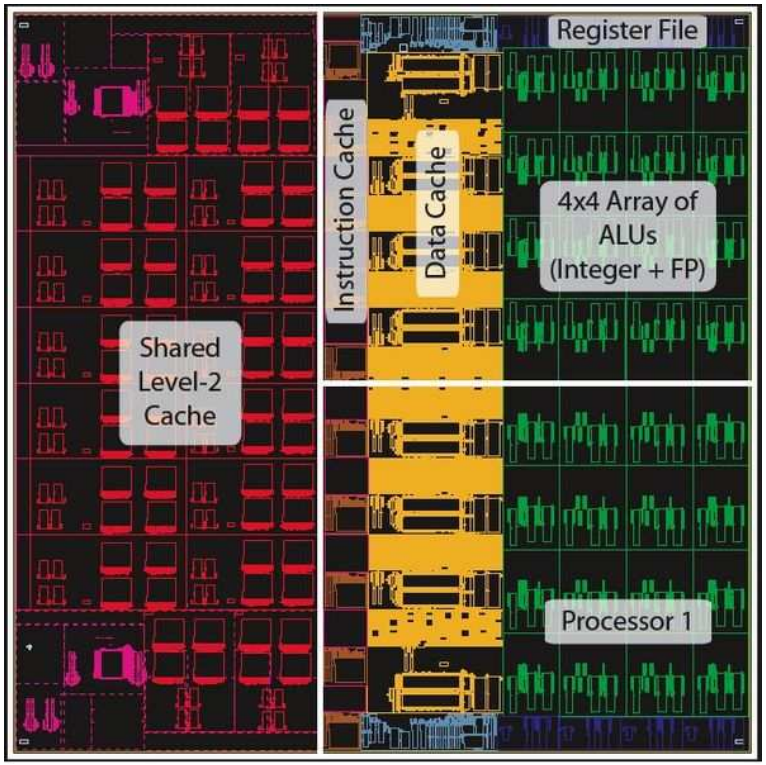
→ Skalierungsprobleme zumindest möglich

- Vorteil der Grid-Architektur: Queues und Register sind durch eine mehrfache Ausführung identischer Tiles mehrfach vorhanden

→ Bearbeitung einer sehr hohen Anzahl an Instruktionen sowie bis zu vier Threads parallel und dadurch gleichzeitig

Prototyp des TRIPS Prozessors I

TRIPS-Prototyp mit zwei TRIPS-Prozessorkerne sowie L2-Cache (Secondary Memory System) mit Schnittstellen zur Peripherie des Mainboards auf dem Prozessor-Die gefertigt



Kürzel	Name	Beschreibung
NT	Network-Tile	Bilden ein Netzwerk, in dem Daten aus und in den Speicher transportiert werden.
MT	Memory-Tile	Bilden den Speicher des L2-Cache, in dem Daten gespeichert werden.
DMA	Direct Memory Access	Kontroller für den Speicherdirektzugriff (Northbridge)
SDC	Static DRAM Controller	Bietet Speicherzugriff auf die SDRAM-Bänke des Arbeitsspeichers
EBC	External Bus Controller	Stellt die Verbindung mit Bussen (Southbridge) her, die sich außerhalb des Prozessors befinden, und kümmert sich um Unterbrechungsanforderungen (*Interrupt Request*, *IRQ*) und externe Busschnittstellen (*External Bus Interface*, *EBI*).
C2C	Chip-to-Chip Connector	Dient dazu, eine direkte Verbindung mit anderen TRIPS-Prozessoren herzustellen. Der C2C ist im TRIPS-Prozessor vierfach vorhanden, um Arrays aus TRIPS-Prozessoren zu bilden und damit Rechencluster aufbauen zu können.

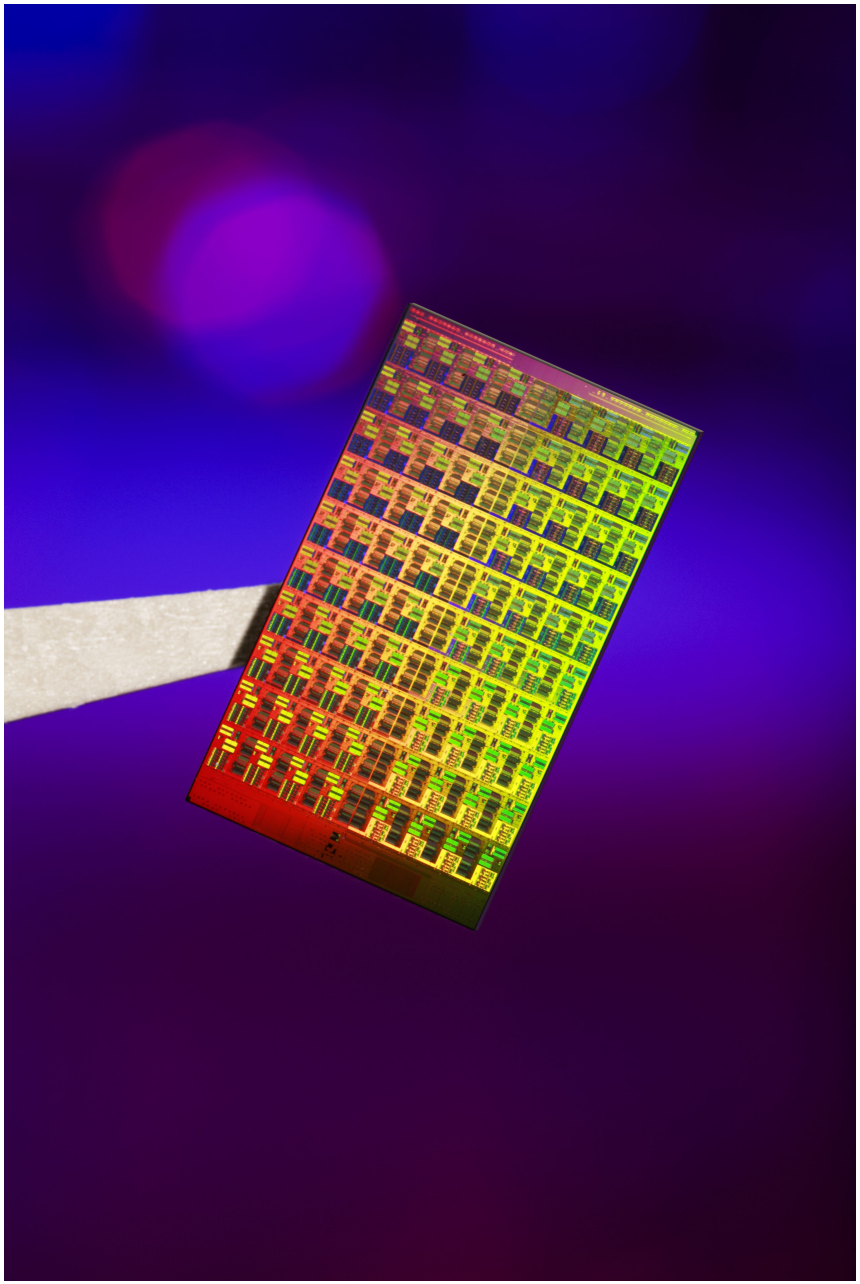
- Prototypfertigung als ASIC in einem 130 nm Prozess mit etwa 170 Millionen Transistoren
- 4-faches Multithreading und kann bis zu 16 Instruktionen je Takt und Prozessor bei einer Taktfrequenz 500 MHz ausfÃ¼hren

→ Spitzenleistung von 16 GOps

Prototyp des TRIPS Prozessors II

Datenflussorientierte Ausführung
Die einzelnen Anweisungsblöcke (bis zu 128 Befehle) werden nicht wie in traditionellen Prozessoren in der Reihenfolge der Anweisungen verarbeitet, sondern in der Reihenfolge des Datenflusses.
Die Abhängigkeiten der Anweisungen voneinander werden direkt in den Anweisungen selbst gespeichert. Eine Anweisung wird ausgeführt sobald alle von der Anweisung benötigten Daten verfügbar sind.

Intel: Tera



Intel TeraFlop Processor (2007)

- 80-core (8x10 array), 100 Mio. Trans, 65nm, 275mm^2
  - 2 FPUs with multiply/accumulate operation (FMAC) in SP with a 9 stage pipeline
  - 3 KB instruction memory (IMEM) = 256 96-bit instructions.
  - 2 KB of data memory (DMEM) = 512 single precision numbers.
  - 10 port (6-read, 4-write) 32 entry register file
- network-on-a-chip
  - A five port router
  - 16 Gigabytes/sec over each port
- performance
  - 4 FLOPS/cycle/core at 4,27 GHz and 1,07V = 1,37 TFLOPs peak
  - 97 Watt, 80 degrees

Intel: Tera

Performance of 4 application kernels

Kernel	Problem sizes and FLOP counts:	Theoretical peak performance projections	Observed cycle counts and single precision TFLOPS @ 4.27 GHz
Stencil on NxM grid	M=2240, N=16 FLOP = N*M*10 = 358400	Even mix of adds and multiplies overlapped with loads and comm. for 95% of peak performance.	1536 cycles per iteration 1 TFLOP
Matrix Multiplication C(N,N) = A(N,M)*B(M,N)	N=80, M=206 FLOP = N*N(2*M-1) = 2630400	Limited by loads from DMEM to 50% of peak performance	22000 cycles 0.51 TFLOPS
Spreadsheet: LxN table of (value,weight) pairs	N = 10 L = 1600 FLOP = 4*L*N-N-L = 62390	Limited by loads from DMEM to 50% peak performance	Two phases: 1st phase 484 cycles; 2nd phase 589 cycles Pipelining over multiple spreadsheets to achieve 0.45 TFLOPS
2D FFT $Y(N,N)=F_N \otimes F_N * X(N,N)$	N=64 FLOP = $8\ N^2 \log_2 N$ = 196608	Limited by integer and floating point operation mix and redundant operations to 17.5% of peak	41846 cycles 0.020 TFLOP

Programming the Intel 80-core network-on-a-chip Terascale Processor, 2008

Intel Larrabee

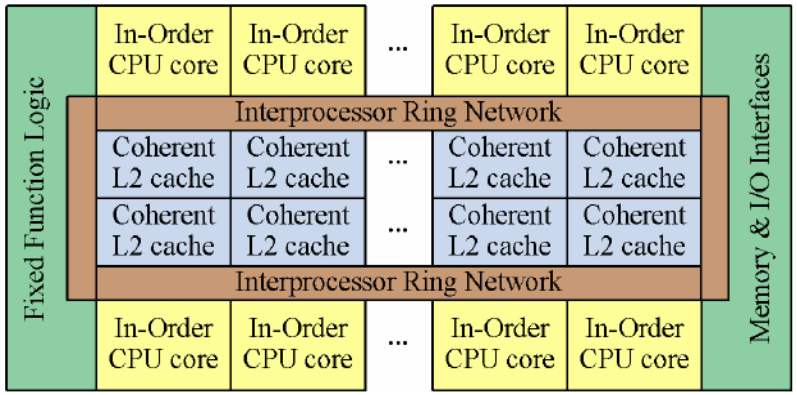
Lessons learned from Programming the 80 . . . Our future work will highlight three major advantages of message passing architectures for many core chips.

1. Managing messages and explicitly distributing data structures **adds considerable complexity** to the software development process. This complexity, however, is balanced by the greater ease of **avoiding race conditions** when all sharing of data is through explicit messages rather than through a shared address space.
2. While a programmer can block data to fit into a cache or use prefetching to prepare a cache in advance of a computation, programmers have little or **no control over eviction of data from a cache**. It is difficult to write software with predictable performance when the state of the cache is so difficult to control. A **NoC chip without cache coherence between cores avoids this problem** altogether.
3. There are software engineering advantages to message passing architectures. Well engineered software is often constructed as modules that can be composed into the final application. Composition is based on isolation of concerns; computing occurs inside a black- box module while sharing of information occurs only through the interfaces to the modules. A shared address space makes it difficult to assure that no unintended sharing is taking place. A **message passing architecture, however, naturally supports a discipline of isolation between modules** since sharing can only occur through the exchange of messages.

Intel Larrabee

Intel Larrabee (2009)

- Many-core Prozessor auf Basis von modifizierten Pentium (P4) Kernen mit zusätzlicher Vektoreinheit
- Einsatz als Graphikprozessor geplant
- ringförmige Kommunikationstopologie
- vollständiger Paradigmenwechsel
- enttäuschende erste reale Kennzahlen
- vorerst offiziell eingestellt



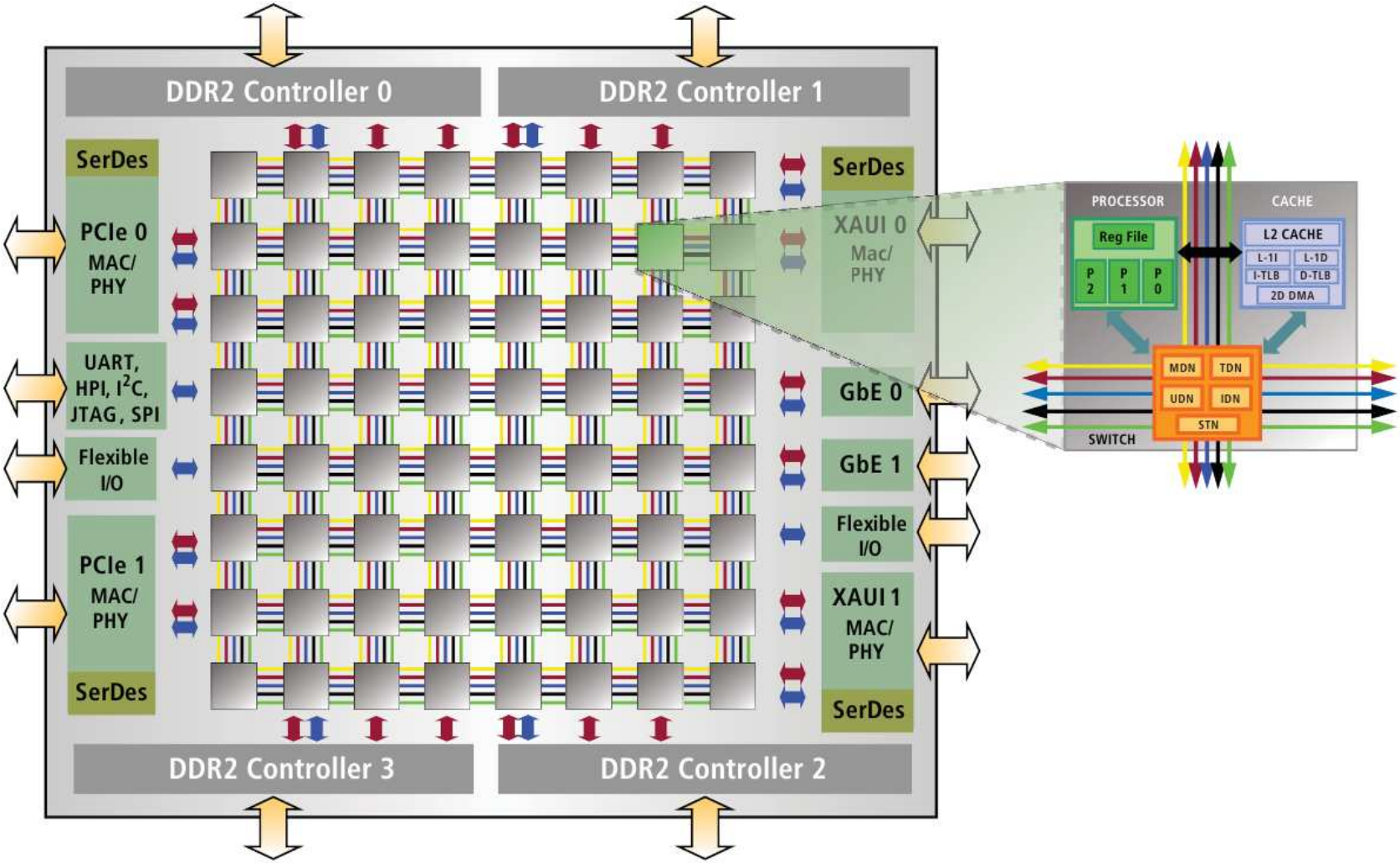
Tilera Tile64 I

Processor Architecture:

- MIMD architecture
- 2D grid of 64 homogeneus, generalpurpose compute elements (tiles)
- Tileras iMesh on-chip network
- 4 DDR2 controllers + IO controllers
- Tiles:
  - Processor
  - L1 & L2 cache
  - non-blocking switch

Tilera Tile64 II

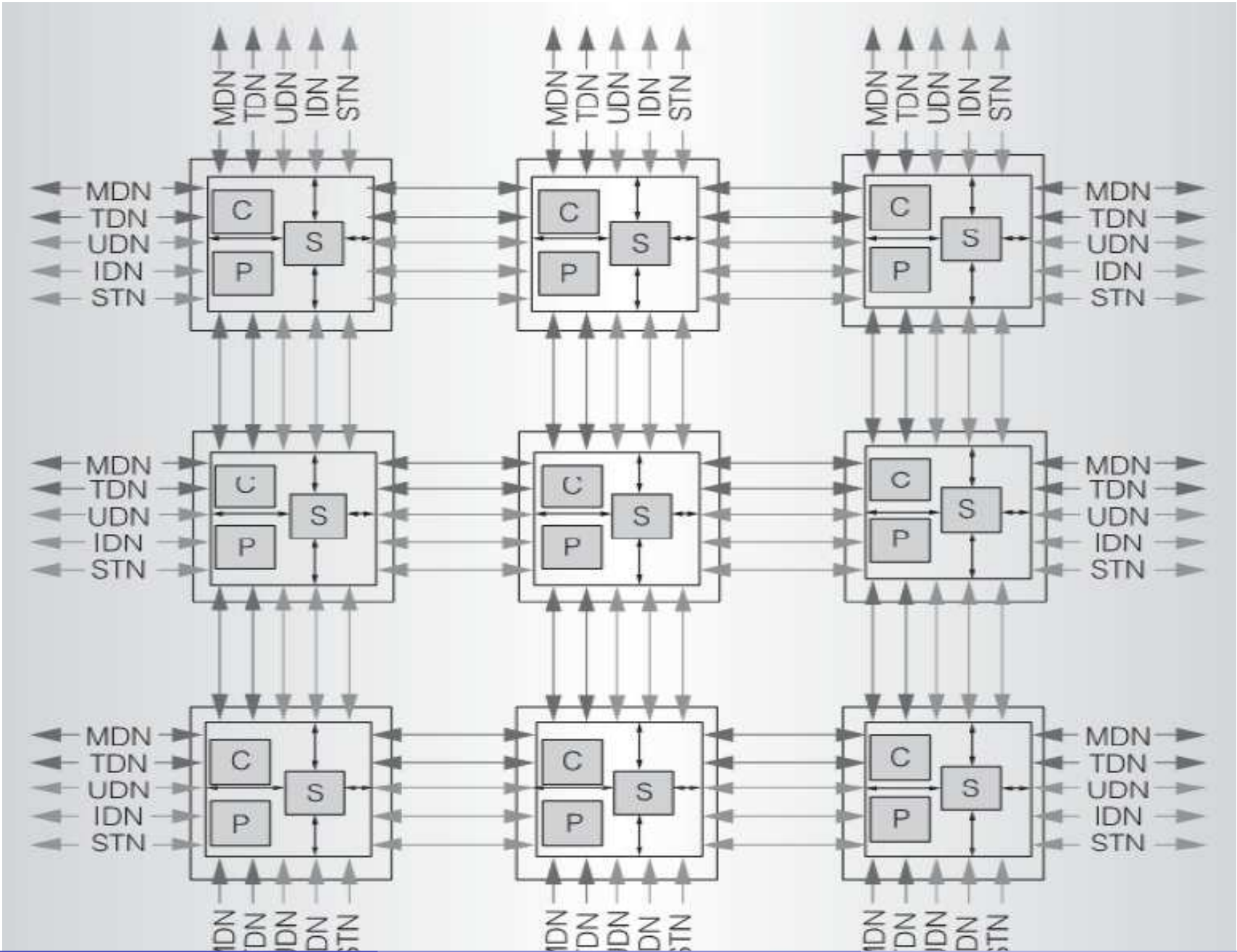
Tile64 floorplan



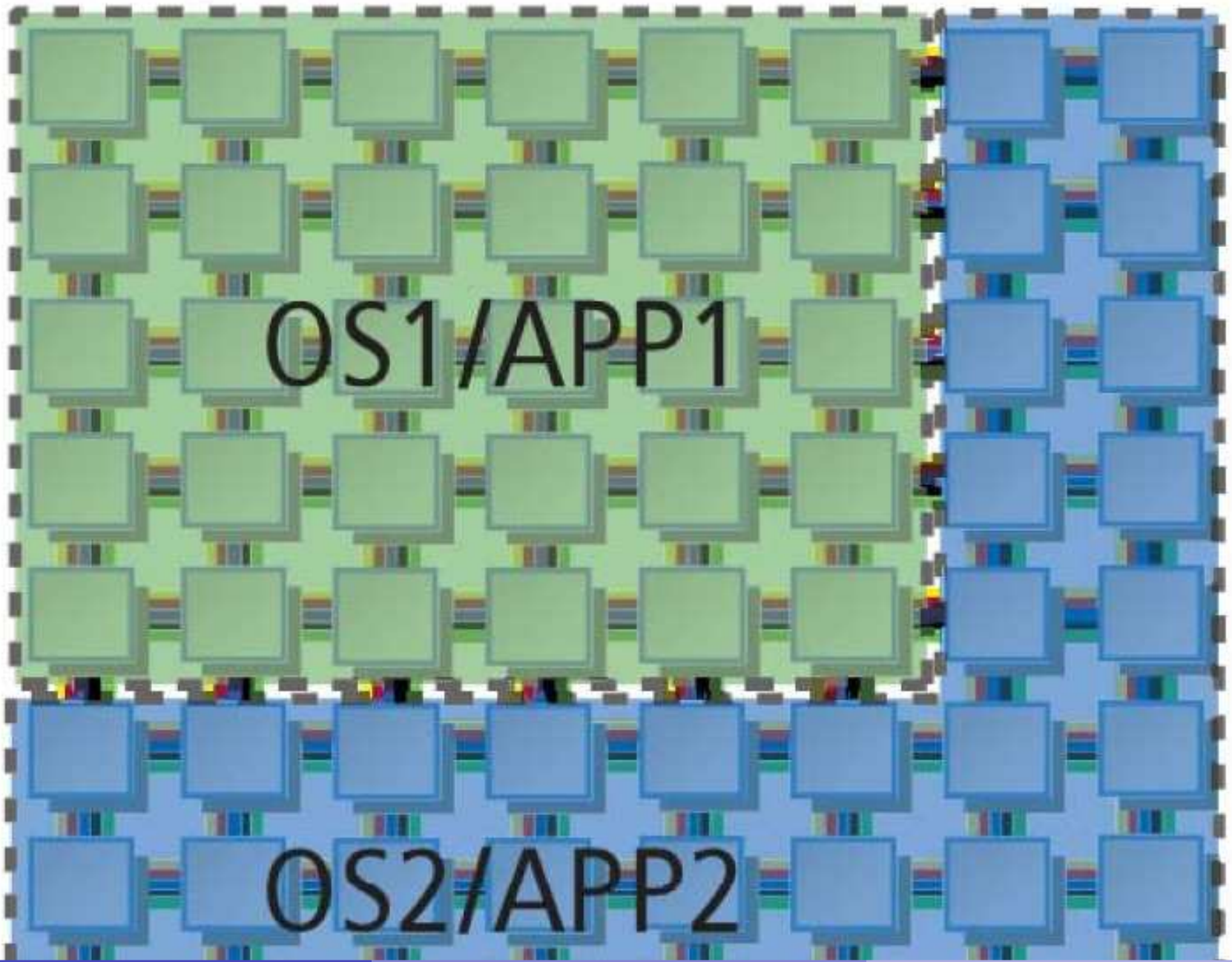
Tilera Tile64 II

	Features	Enables
Massively Scalable Performance	• 8 X 8 grid of identical, general purpose processor cores (tiles) • Three-way VLIW pipeline for instruction level parallelism • 5 Mbytes of on-chip cache • Up to 443 billion operations per second (BOPS) • 31 Tbps of on-chip mesh interconnect enables linear application scaling • Up to 50 Gbps of I/O bandwidth	• 10 Gbps Snort® processing • 20 Gbps nProbe • 16 X 16 SAD at 540 MBlocks/s • H.264 HD video encode: 2+ streams of 720p30
Power Efficiency	• 700MHz - 866MHz operating frequency • 15 - 22W @ 700MHz all cores running full application • Idle Tiles can be put into low-power sleep mode • Power efficient inter-tile communications	• Highest performance per watt • Simple thermal management and power supply design • Lower operating cost
Integrated Solution	• Four DDR2 memory controllers with optional ECC • Two 10GbE XAUI configurable MAC or PHY interfaces • Two 4-lane PCIe interfaces: Root complex or endpoint mode • Two GbE MAC interfaces • Flexible I/O interface	• Reduces BOM cost – standard interfaces included on-chip • Dramatically reduced board real estate • Direct interface to leading L2-L3 switch vendors
Multicore Development Environment	• ANSI standard C / C++ compiler • Advanced profiling and debugging designed for multicore programming • Supports SMP Linux with 2.6 kernel • Tilera Multicore Components (TMC™) libraries for efficient inter-tile communication	• Run off-the-shelf C programs • Reduce debug and optimization time • Faster time to production code • Standard multicore communication mechanisms

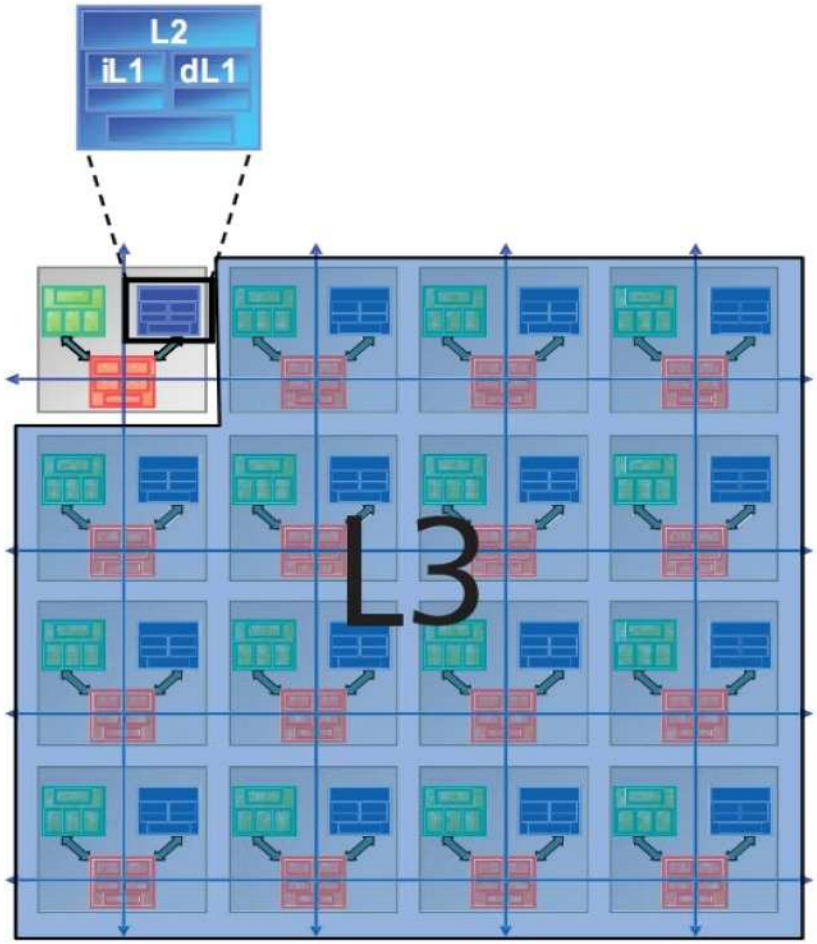
Tilera Tile64 III

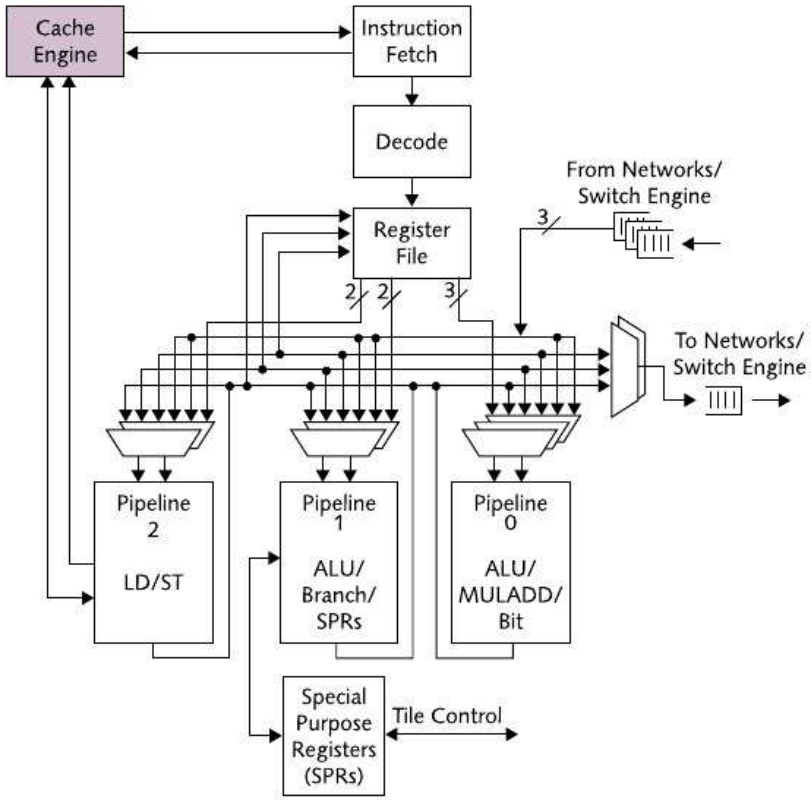


Tilera Tile64 IV



Tilera Tile64 V





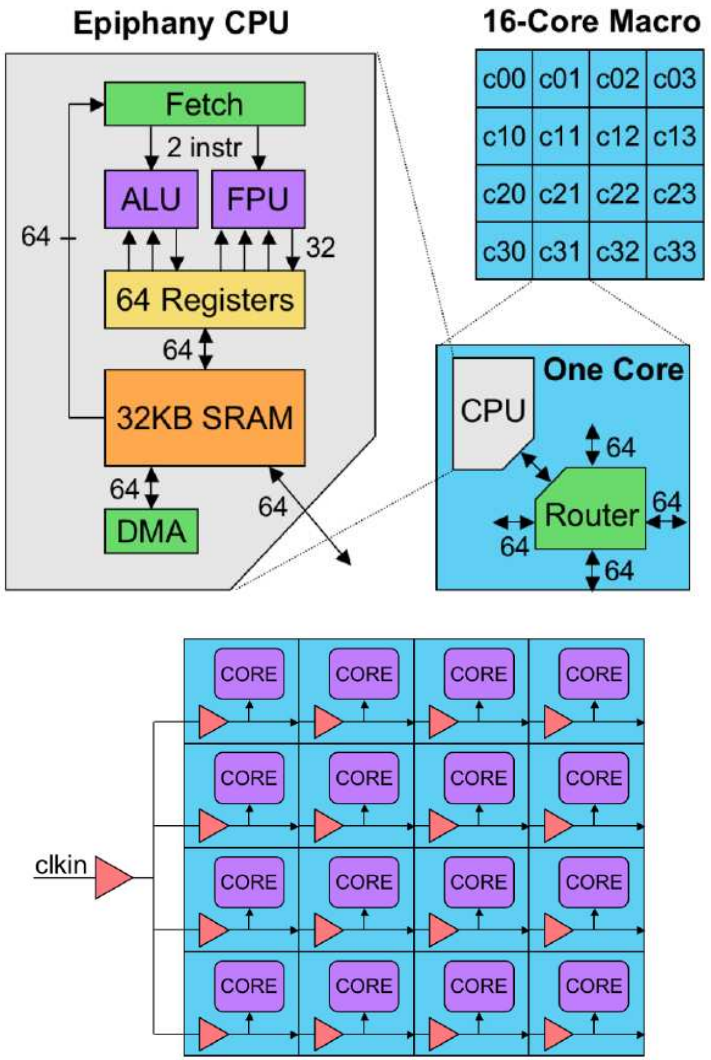
Tilera Tile64 VI

# of Cores	Bus/Mesh	Very Low Traffic	Medium Traffic	High Traffic
4 Cores	Mesh	24	25	29
	Bus	23	28	69
16 Cores	Mesh	26	29	41
	Bus	24	137	>1,000 (Unusable)
64 Cores	Mesh	28	34	58
	Bus	25	>1,000 (Unusable)	>1,000 (Unusable)

Latency in clock cycles

Adapteva Epiphany

- 2D mesh of general-purpose RISC cores hooked up via a high bandwidth, low latency on-chip network
- current implementation has 16 cores with SP FP operations, 4k-core layout with 64 W and 4 TFLOPs peak finished
- 1000 thousand core version is already in the works with DP FP (2011)
- similar to Tilera's manycore chips with focus on floating point power
- processor doesn't have a hardware cache, each core has 32 KB of local memory
- fast on-chip-network allows extremly low-latency data requests
- programming in ANSI-C with message-passing paradigm following MCAPI-Standard not MPI
- currently as co-processor in consumer mobile devices and embedded systems



Adapteva Epiphany

Comparison of mobile Processors

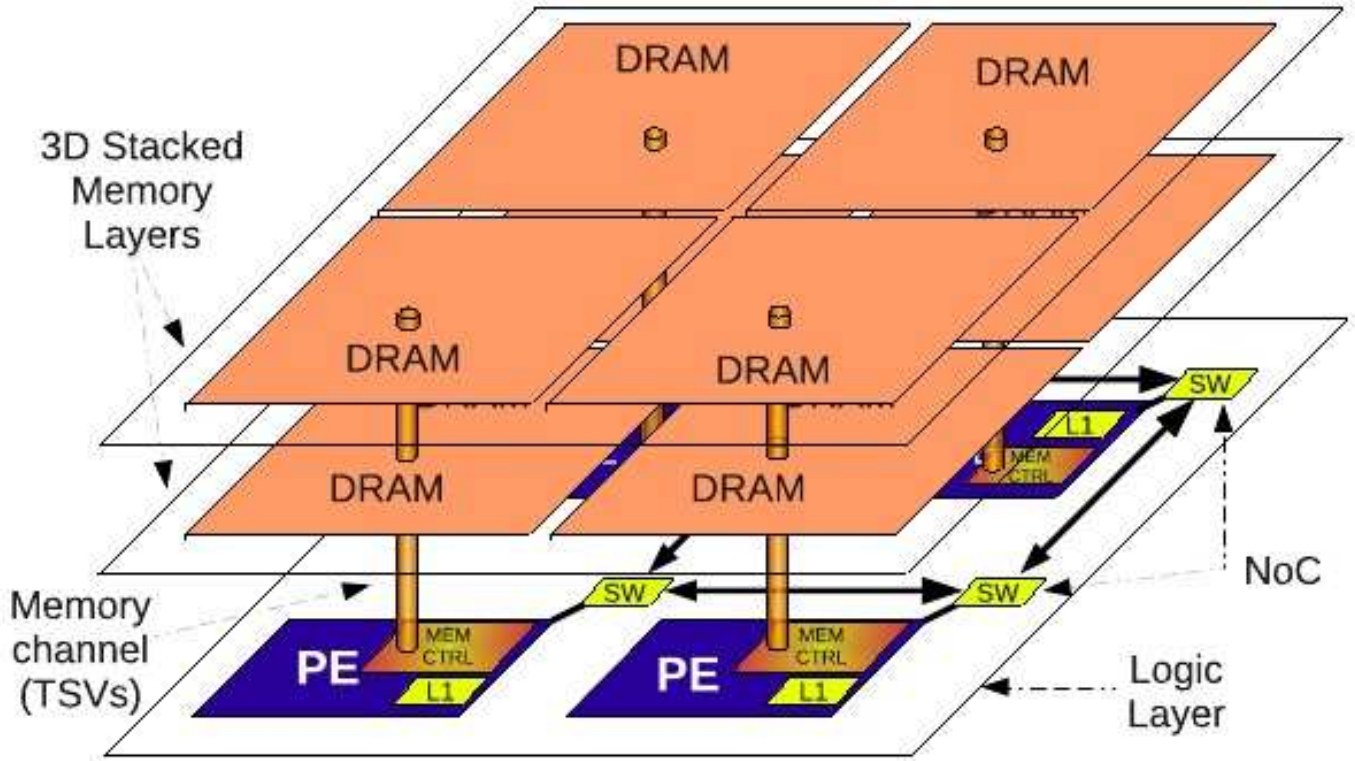
	Adapteva Epiphany*	ARM Cortex-A9*	Vivante GC2000
# of Cores	16 cores	4 cores	4 cores
Total SRAM	512KB SRAM	256KB L1$	152KB SRAM
FP Ops/Cycle	32 FP	16 FP	16 FP
Clock Speed	600MHz	1.2GHz†	1.5GHz†
Peak Mflops	19,200	19,200	24,000
Power (typ)	270mW	1,350mW†	650mW†
Mflops/W	71,000/W	14,000/W	37,000/W
Die Area	2.05mm^2	4.6mm^2†	2.8mm^2†
Mflops/mm^2	9,400/mm^2	4,200/mm^2	8,600/mm^2

NoC: Hiting the memory wall again!

Memory access speed is not the only cause of the memory wall, which is also tied to memory-to-logic interfacing issues. **DRAMs currently are developed using high-density NMOS process** optimized to create high-quality capacitors and low-leakage transistors. On the other hand, logic **chips are manufactured in high-speed CMOS processes** optimized for transistor performance and complex multi-level metalizations. The two processes are not compatible, therefore highly optimized DRAM and logic cannot coexist on the same die1 and they must be interfaced through **off-chip interconnects**. This imposes tight **constraints on the maximum DRAM pin count resulting in a limited-bandwidth** interface between DRAM and Logic chips. A number of countermeasures have been adopted to overcome this problem, such as multiplexing the address in two phase (RAS and CAS) and fast burst-transfer modes to increase per-pin bandwidth. This has lead to an ever-increasing power and signal integrity bottleneck in memory interfaces.

NoC and RAM: Possible Solution Approach?

3D stacking of NoC-processor layers and memory layers



Literatur

Literatur zu Paralleler Rechnerarchitektur

- Hennessy J., Patterson D.: Computer Architecture – A Quantitative Approach, 3. Ausgabe, Morgan Kaufmann, 2003
- Hennessy J., Patterson D.: Computer Architecture – A Quantitative Approach, 4. Ausgabe, Morgan Kaufmann, 2007
- Culler D., Singh J., Gupta A.: Parallel Computer Architecture – A Hardware/Software Approach, Morgan Kaufmann, 1999
- HyperTransport Konsortium: www.hypertransport.org
- Hwang K.: Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw-Hill, 1993
- Grama A., Gupta A., Karypis G., Kumar V.: Introduction to Parallel Computing, 2. Ausgabe, Benjamin/Cummings, 2003