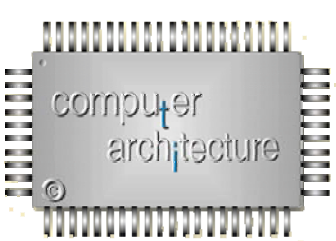


the HT core

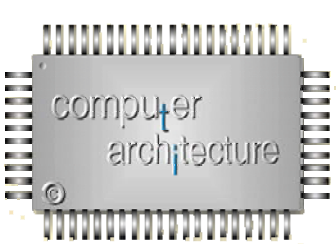
a reference HT device interface

A Versatile, Low Latency HyperTransport Core

David Slogsnat

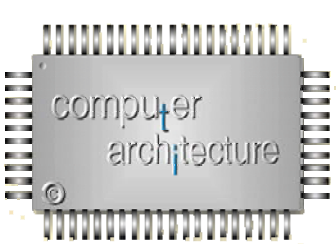
Computer Architecture Group
University of Mannheim

UNIVERSITÄT MANNHEIM



Outline

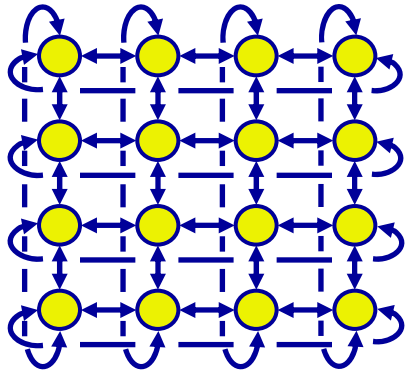
- Introduction
  - Motivation
  - Short introduction to HyperTransport
- The Xilinx FPGA implementation of the HT Core
- HT Core status
- Other targets besides Xilinx
- First benchmark results
- Conclusion



- The talk presents the design of a generic HyperTransport (HT) core. It is specially optimized to achieve a very low latency. The core has been verified in system using the rapid prototyping methodology with FPGAs. This exhaustive verification and the generic design allows the mapping to both ASICs and FPGAs. The implementation described in this talk supports a link width of 16bit, as is used in Opteron based systems. On a Xilinx Virtex4FX60, the core supports a link frequency of up to 400MHz DDR and offers a maximum bidirectional bandwidth of 3.6 GB/s. The in-system verification has been performed using a custom FPGA board that has been plugged into a HyperTransport Extension Connector (HTX) of a standard Opteron based mainboard. HTX slots in Opteron based mainboards allow a very high-bandwidth, low latency communication, as the HTX device is directly connected to one of the HyperTransport links of the processor. The HT core in combination with the HTX board is an ideal base for prototyping systems and FPGA coprocessors. The HT core is available as open source.

Why HyperTransport?

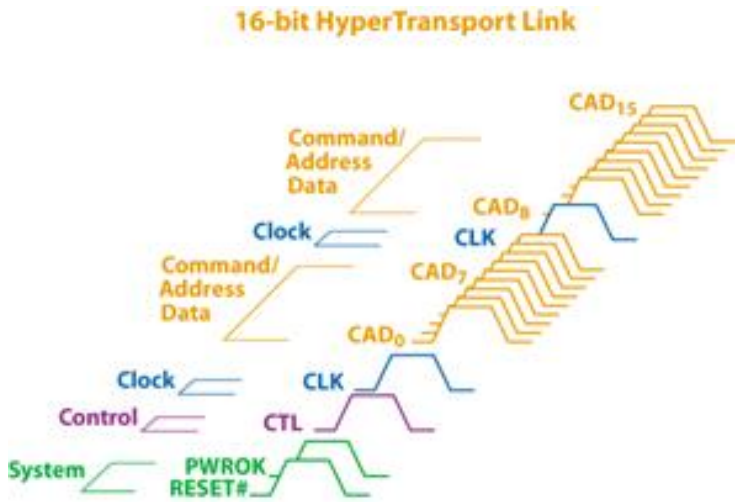
- Research on networks and interfaces in High Performance Computing (HPC)
- Problem: find host interface for prototype



- Motivation to choose HT:
  - "beauty of simplicity" compared to PCI-X/PCIe leads to increased ease of use in design & verification
  - protocol better suited for latency sensitive applications
  - Direct Connect Architecture for low-latency path to AMD Opteron processors
- Reasons for an own implementation:
  - is also basis for coherent HT developments
  - few HT cores were available on market

HT Link

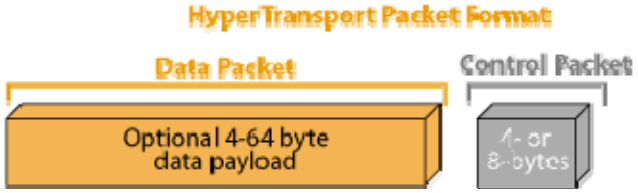
- Parallel links with dedicated DDR source clocks
  - No serializer/deserializer
  - No 8b/10b coding
  - No clock recovery
- Variable link width and frequency
  - links have to be initialized
- In Opteron systems:
  - Link widths of 8 and 16 bit
  - HT200 – HT1000
- Bidirectional link bandwidths:
  - HT200, 8 bit: 0.8 Gbyte/s
  - HT400, 16 bit: 3.2 Gbyte/s
  - HT1000, 16 bit: 8 Gbyte/s



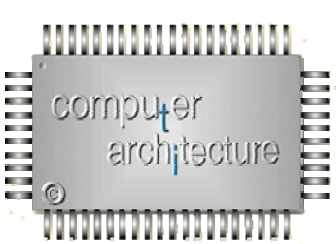
http://www.hypertransport.org

HT Packets

- All data is transferred in chunks of 32 bit
- Packet based protocol
  - Split phase transactions
  - Credit based flow control
  - Control packet interleaving
- Virtual Channels to avoid deadlocks:
  - Posted requests
  - Non-posted requests
  - Responses



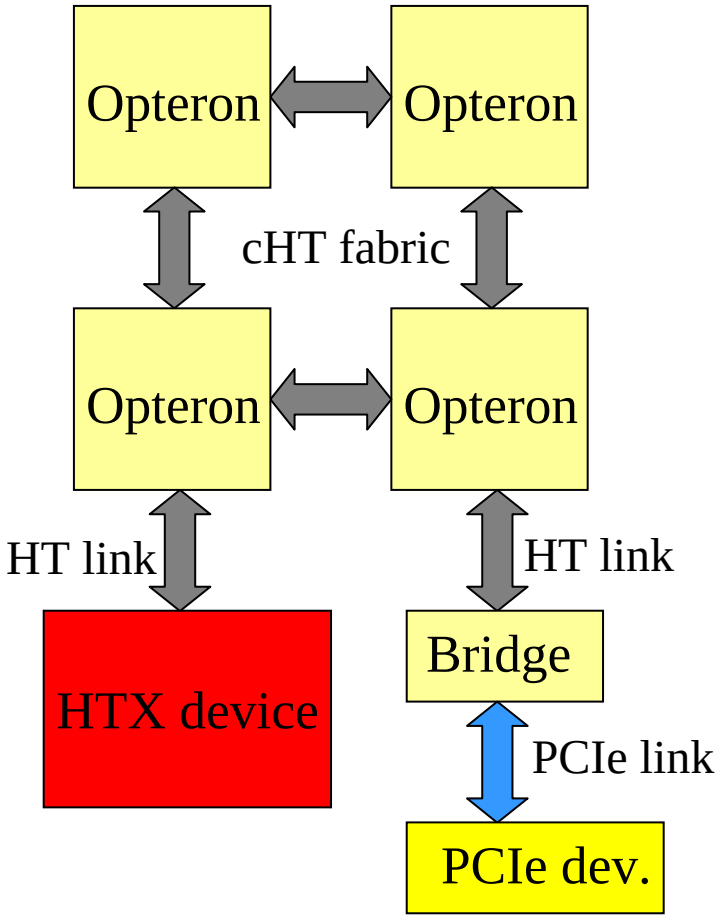
http://www.hypertransport.org



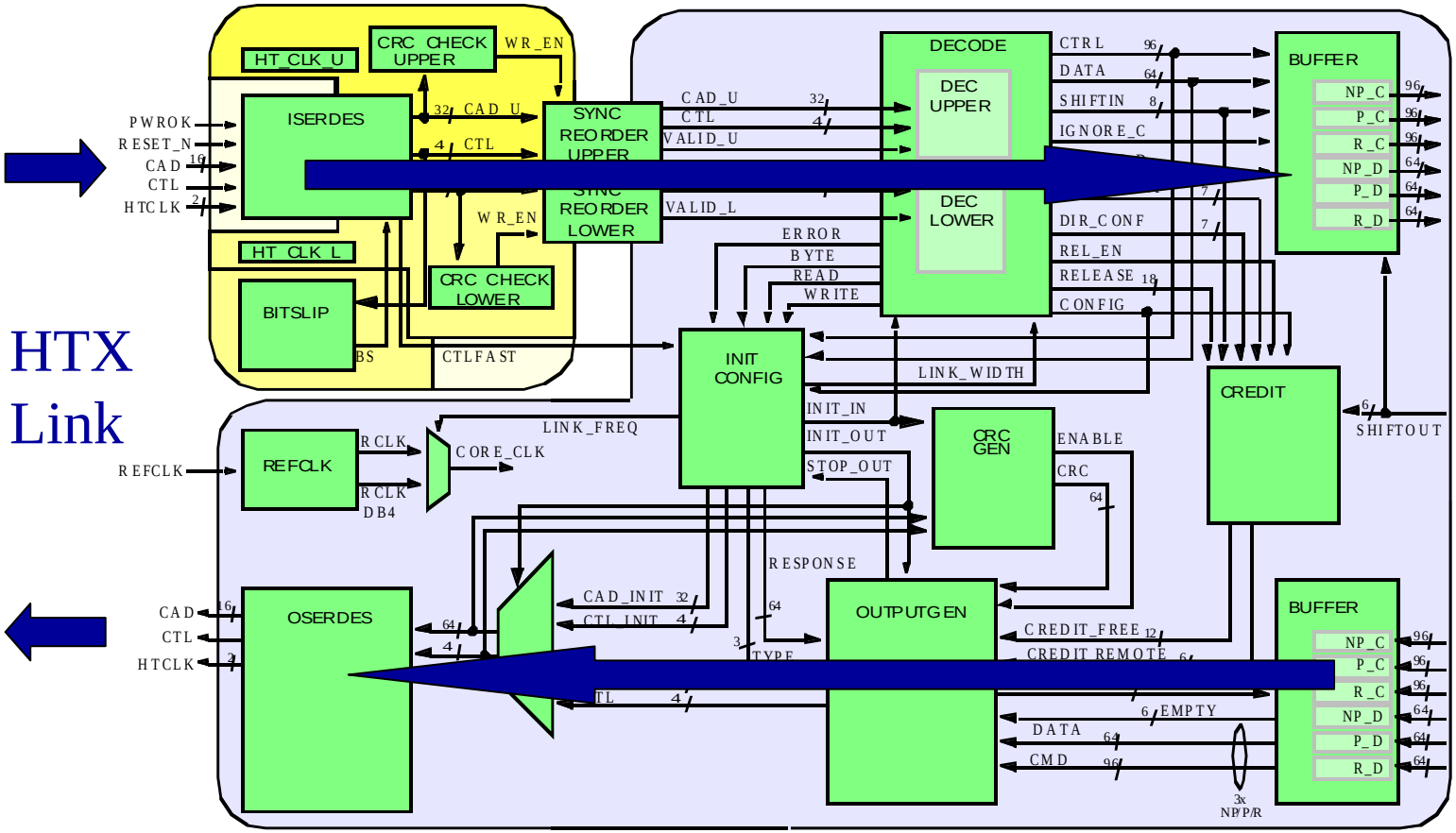
HT Core Overview

HT Core Quick Facts

- Developed for:
  - Xilinx Virtex-4 FX device
  - HT200 and HT400 support
  - 8bit/16bit wide links
  - to achieve lowest resource utilization, use of FPGA hard macro blocks where helpful
  - usage in HTX slots
- When used in AMD Opteron systems, slower link between FPGA and Opteron does not decrease speed of other links

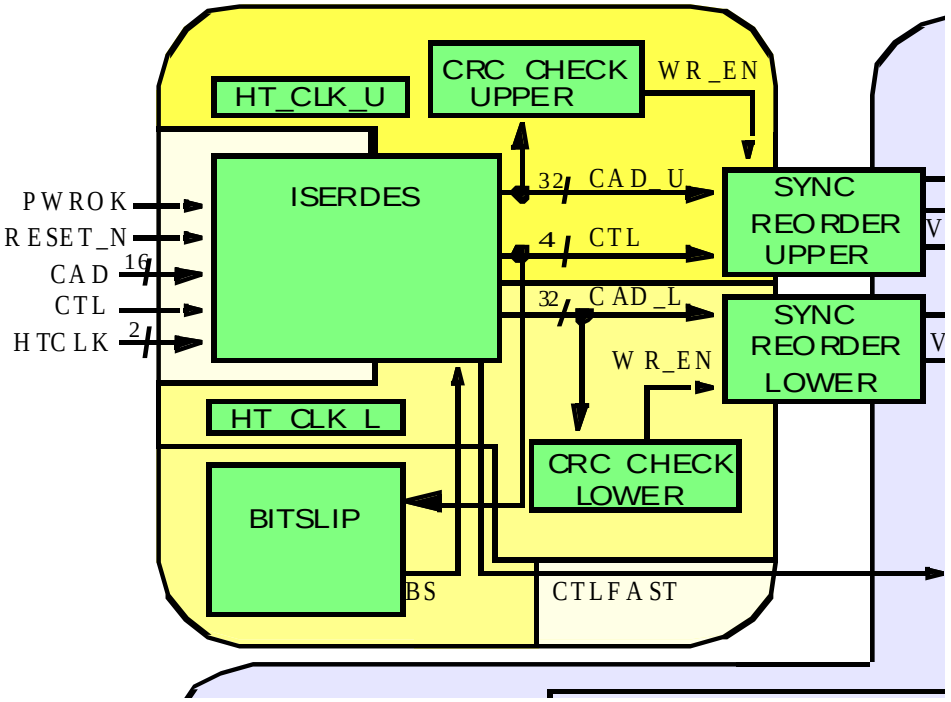


HT core


HTX Link
User Application
HT_CLK_U
CRC CHECK UPPER
WR_EN
PWROK
RESET_N
CAD
CTL
HTCLK
ISERDES
32 CAD_U
4 CTL
SYNC REORDER UPPER
SYNC REORDER LOWER
W R_EN
HT_CLK_L
CRC CHECK LOWER
BITSLIP
BS
CTLFAST
CAD_U
CTL
VALID_U
VALID_L
DECODE
DEC UPPER
DEC LOWER
CTRL
DATA
SHIFTIN
IGNORE_C
DIR_CONF
REL_EN
RELEASE
CONFIG
ERROR
BYTE
READ
WRITE
BUFFER
NP_C
P_C
R_C
NP_D
P_D
R_D
INIT CONFIG
LINK_WIDTH
LINK_FREQ
INIT_IN
INIT_OUT
STOP_OUT
CRC GEN
ENABLE
CRC
CREDIT
SHIFTOUT
REFCLK
RCLK
CORE_CLK
RCLK DB4
RESPONSE
OSERDES
CAD_INIT
CTL_INIT
TYPE
OUTPUTGEN
CREDIT_FREE
CREDIT REMOTE
EMPTY
DATA
CMD
3x NPP/R

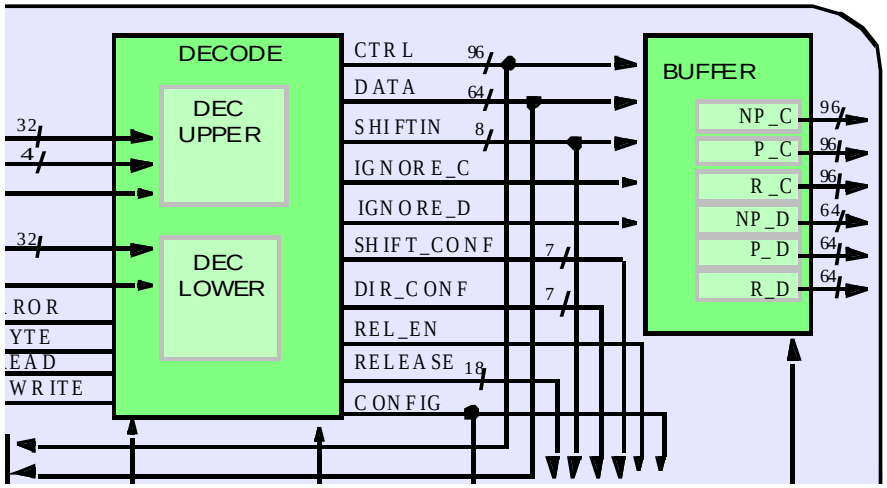
Physical Link Input

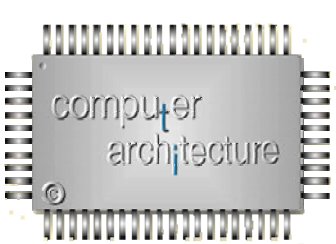
- Separate clock regions for every 8-bit CAD word
- Data rate of 800Mbit/s per LVDS pin
- Each region has 400 MHz link and internal ISERDES clocks
- Parallelization degree of 4 leads to internal 200MHz SDR signals
- Alignment to 32-bit boundaries
- CRC check and removal
- Synchronization to core clock region in small FIFOs



Decode

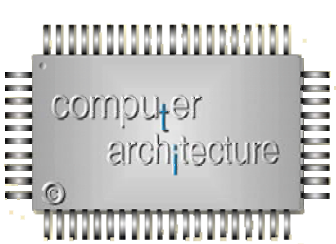
- Assemble control packets and:
  - Put packets destined to device into the queues of the application interface
  - Attach data packets to control packets
  - Decode credit information
  - Treat “unwanted” packets
- Parallelism vs. clock frequency:
  - Parallelism of 2 required, leads to good scalability of core clock frequency
  - More parallelism not feasible
- 2- stage pipeline





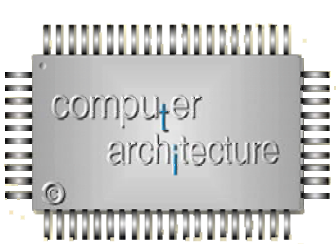
Application Interface

- Dedicated data and control packet queues for every Virtual Channel
  - Control packets have a 96 bit format very similar to the format of 96 bit HT packets
  - Data stored in 64 bit chunks
  - Queues sized to hold up to 32 packets each
- Applications can directly access these queues using a valid-stop synchronization mechanism



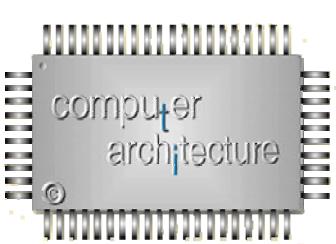
Output Path

- Output arbiter selects with priority between:
  - CRC
  - Configuration responses
  - VC queues from application
  - NOP packets with credit information
- 4x serialization similar to input path



Ordering

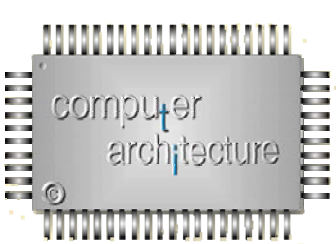
- HT ordering rules allow posted packets to block packets from other VCs under some circumstances
- Current implementation allows access to non-posted or response packets only if the posted queue is empty
- More efficient implementation exists as behavioral model, implementation pending



HT Core Status and First Results

HT Core Status

- Stable and verified environment:
  - HT Core with HT 200MHz (400MT/s), 8bit wide
  - Iwill DK8-HTX mainboard with LinuxBIOS, UoM HTX reference board
- Ongoing work:
  - Support for HT400 and 16bit wide link, resulting in a bidirectional bandwidth of 3.2GByte/s
  - Verification of
    - 2 additional mainboards
    - 2 Altera-based platforms



First Benchmark Results

- Benchmark platform:
  - HT core with HT200 and 8 bit wide links
  - Iwill DK8-HTX mainboard with two Opteron 246 processors and Linux 2.6.18
- CPU write accesses using write combining result in a bandwidth of up to 350MByte/s (theoretical maximum is 400MByte/s)
- CPU read accesses result in
  - a latency of 30 HT core cycles, i.e. 300ns, for 32bit reads
  - Larger reads executed sequentially as 32bit reads
  - As a result, read bandwidth is only 12,5 MByte/s

Transfer Size	Bandwidth
4 Byte	95 MB/s
8 Byte	189 MB/s
16 Byte	253 MB/s
32 Byte	303 MB/s
64 Byte	337 MB/s

Write bandwidth

Read Latency

Transfer Size	Latency
4 Byte	320 ns
8 Byte	620 ns
64 Byte	4900 ns

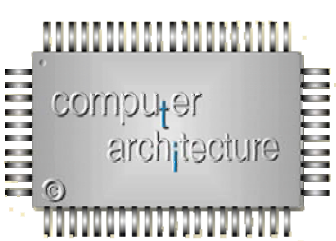
Xilinx Implementation Results

Resource utilization on Virtex-4 FX60:

Ressource	8 bit link		16 bit link	
Logic Slices	2,699	10%	4,123	16%
FIFO16/ RAMB16s	30	12%	30	12%
DCM_ADVs	3	25%	4	33%
ISERDESs	10	1%	19	2%
OSERDES	9	2%	17	2%

Hardware latency:

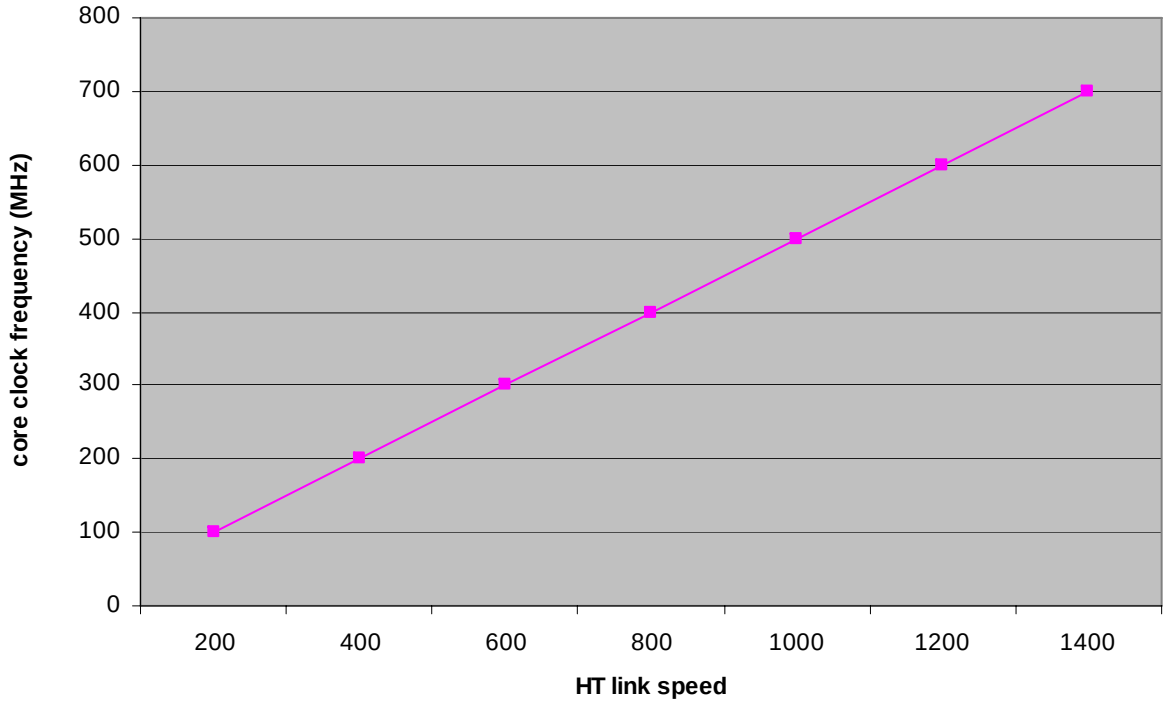
Direction	Clock Cycles	Delay@ HT200	Delay@ HT400	Delay@ HT500
In	11	55ns	27.5ns	22ns
Out	7	35ns	17.5ns	14ns



Targets

- Core is portable, since written in Verilog RTL
- Target specific macro blocks:
  - SRAM
  - PLL/DLLs
  - De-/Serializers
  - I/O Cells
- Implementation status:
  - Xilinx Virtex-4: done
  - Altera Stratix II: in progress
  - Lattice: planned
  - ASIC: trial planned

Targets: Scalability

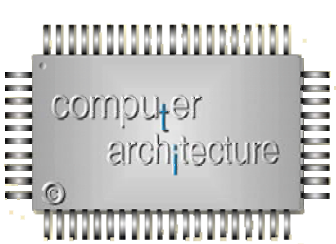


FPGA implementation

ASIC implementation

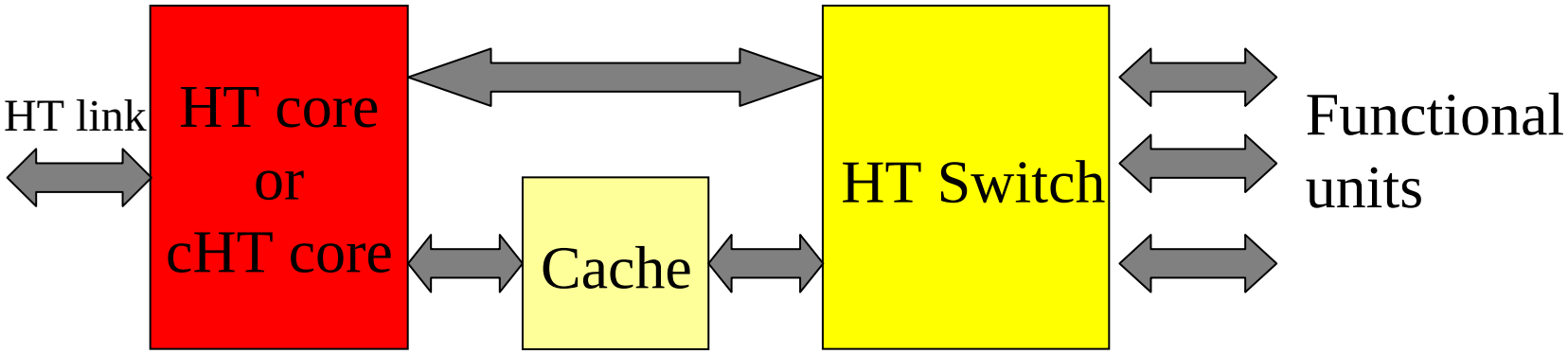
HT core on Xilinx Virtex-4

Current AMD Opteron processors



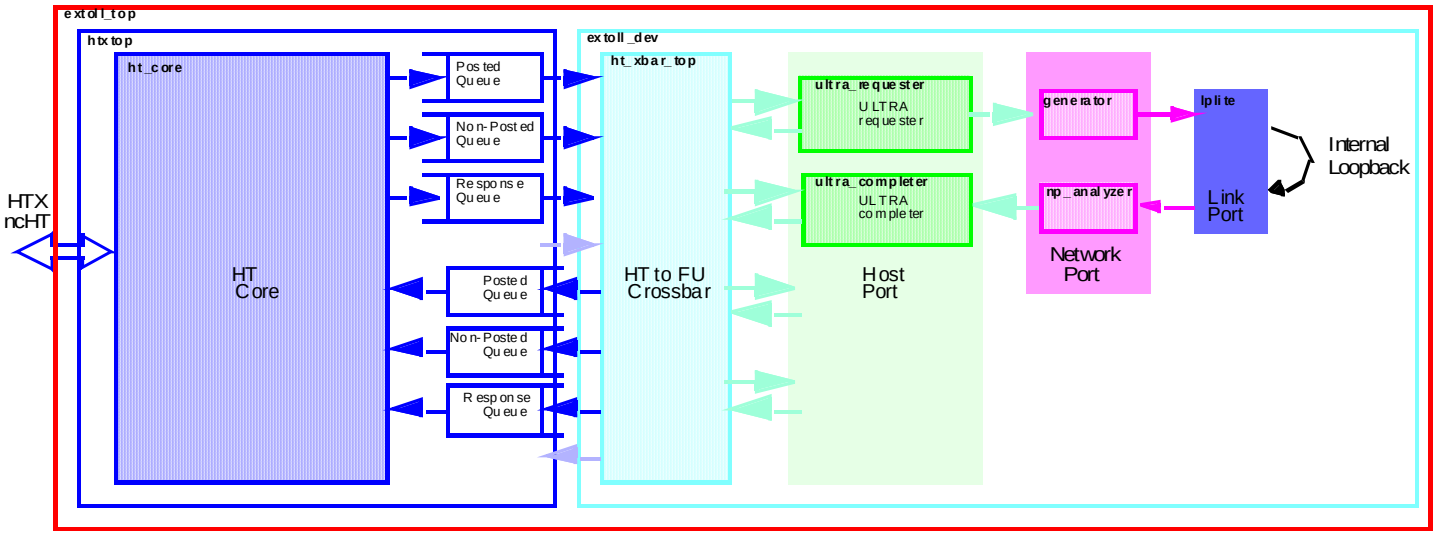
Usage Examples

HT Environment at the CAG



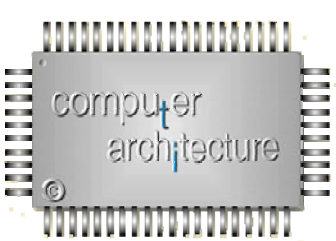
- Environments for both HT and cHT
  - Bus functional model based simulation environments
  - HT switch for use with both cores
  - Multiple devices or functional units on the same link
  - Plans to extend switch to support multiple HT cores, thus forming an HT bridge device

Sample NIC Design



Latencies in clock cycles

Direction	HT core	XBar	ULTRA	Netport	Linkport	overall latency	latency @ 100MHz
In	11	3	15	4	1	34	340 ns
Out	7	3	31	8	5	67	670 ns



Conclusion

- HT best choice for our prototyping environment
- Basis for research on HT and CPU-NIC interfaces together with:
  - AMD
  - HyperTransport Consortium
  - Network of Excellence in HyperTransport
- Continued development thanks to support of AMD
  - Core available under open-source license from CoEHT website
  - Officially supported platform: HTX board and Iwill DK8-HTX mainboard
  - Strong interest to spread core to other platforms

Thank You for Your Attention!

http://www.ra.informatik.uni-mannheim.de/coeht/