

System House Expectations to Flexible Processors

Jari A Parviainen, Specialist,
NMP Communications Baseband RFA
Tampere: 25.11.2002

Contents:

- DSPs and terminal technology trends
- Terminal challenges
- Processor generators and their utilization
- Networking
- Summary: According Keutzer

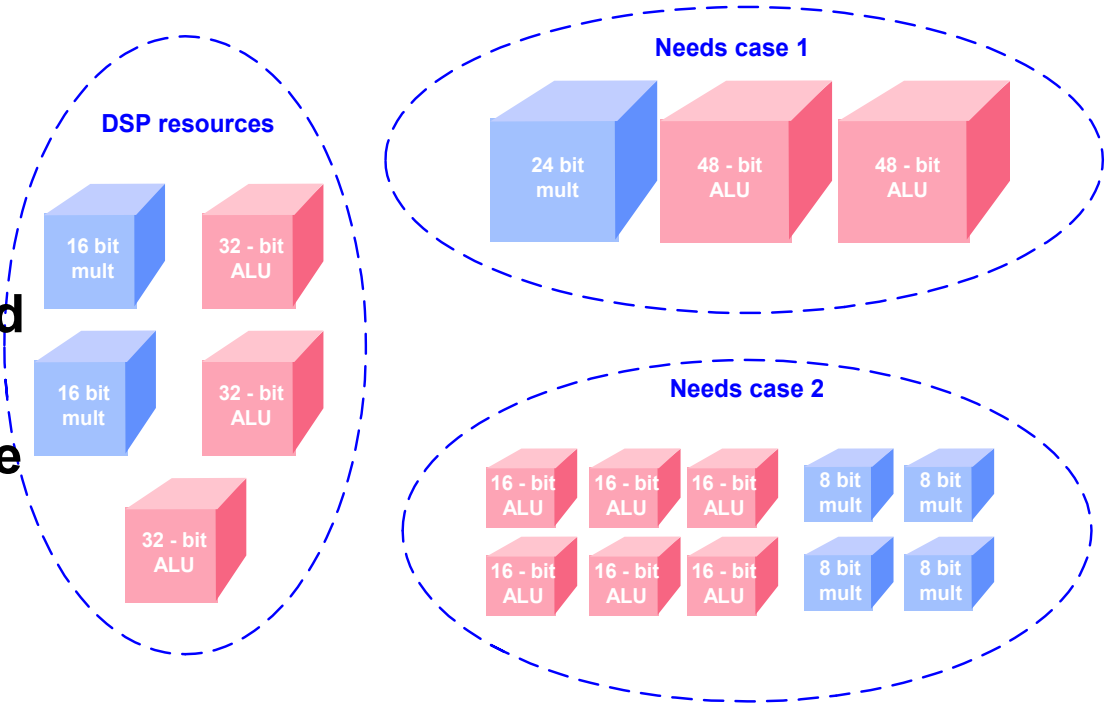
Generic DSPs:

- Algorithm >> << Architecture
- Not possible to modify
 - Resolution
 - Memory interface
 - Execution units

==>> A lots of more cycles needed than planned at the beginning.

- The processing needs for future applications not necessarily known in DSP specification time.

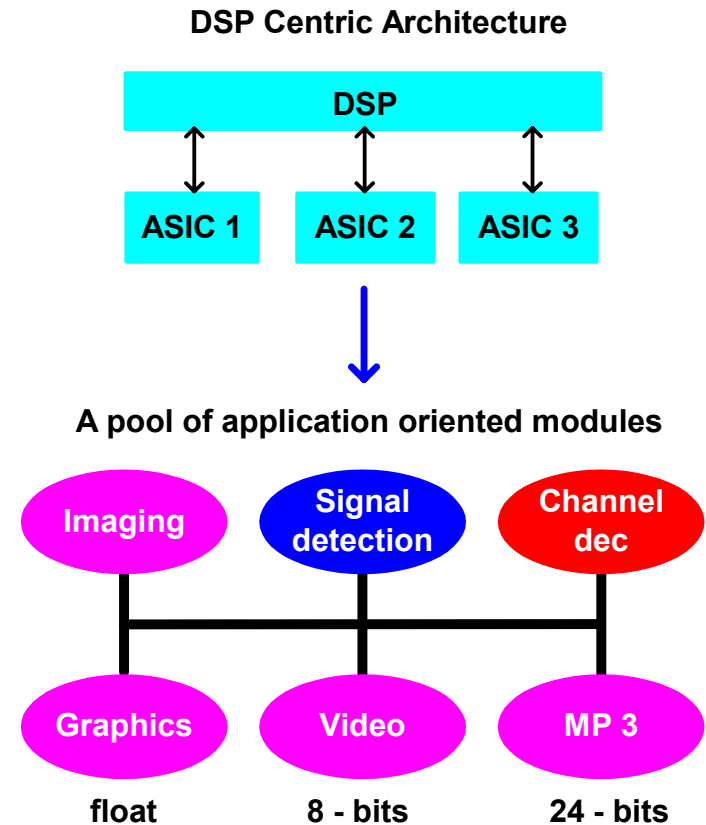
- Processing requirements difficult to estimate:
 - Mixed algorithm & cntrl code
 - Irregularity in algoritms:
 - Sorting
 - Division



- More demanding algorithms to HW.
- Which is the best DSP for the purpose?

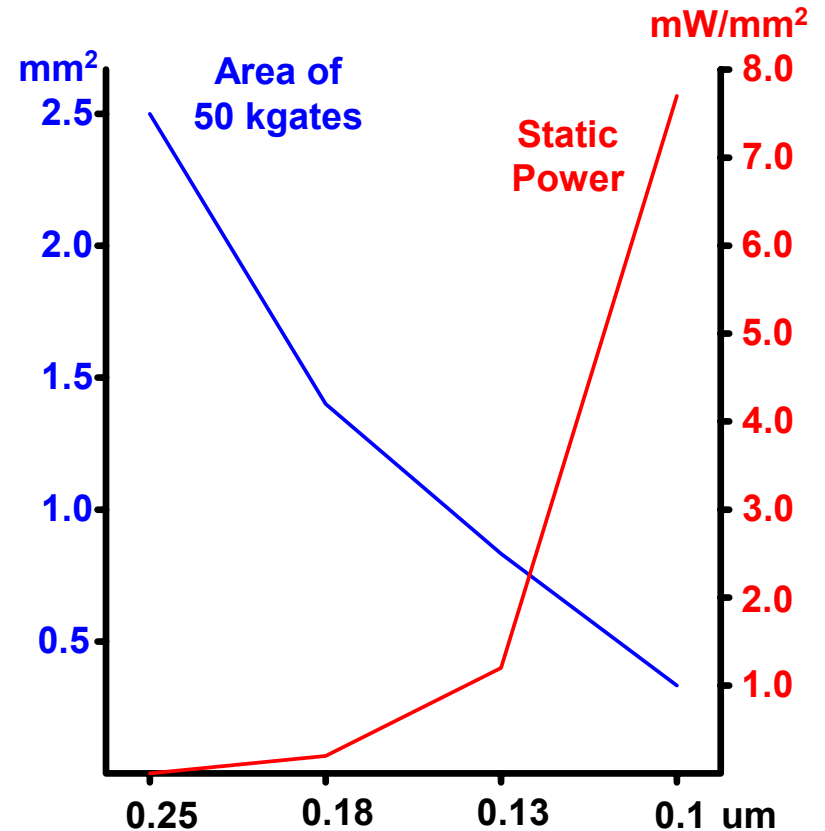
Architecture development trends:

- According Keutzer future SoCs contain 1000 pcs of 50 - 100 kgates functional modules.
- Trend visible in terminal side at some extent. Not enough processing power in DSPs, so ASICs are needed.
- A pool of processing elements connected by a communication network.
- Those individual processing elements could be application oriented processors.



Terminal challenges 1:

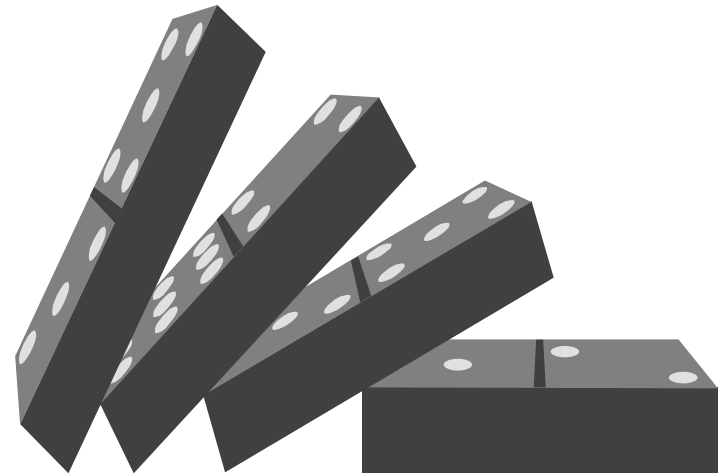
- As the future process geometries shrink, the power dissipation and especially leakage power will become a bigger problem.
- Dynamic power increasing
 - More wires, more capacitance
 - Increasing clock frequencies
- Need for mechanisms, how power dissipation can be reduced.
 - Gated clocks
 - DVS



Impact of small process geometries on microarchitectures in systems on a chip. D. Sylvester, K Keutzer. IEEE 2001

Terminal challenges 2:

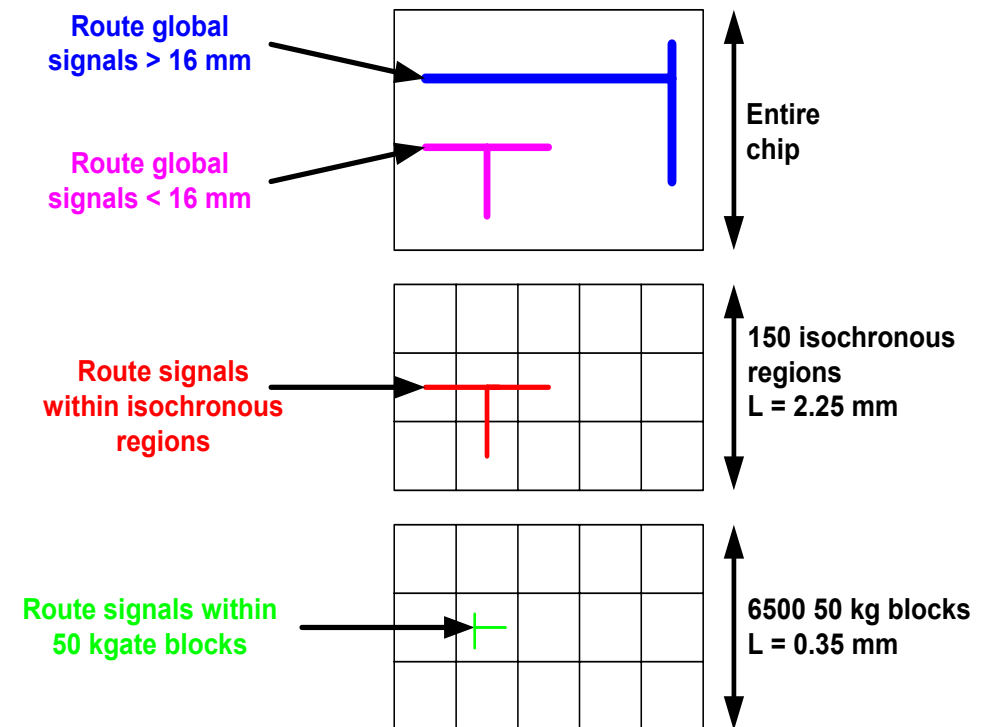
- **A need for a platform:**
 - **Communication framework**
 - **A method for designing modules**
- **Need for a systematic way to design architectures:**
 - **Incremental additions possible**
 - **One modification should not cause a cardcastle effect.**
 - **The modules should be independent.**
- **Other issues:**
 - **Area between logic and memory**
 - **The applicability of ROM**
- **How to implement control and dataflow?**



Typical System Design?

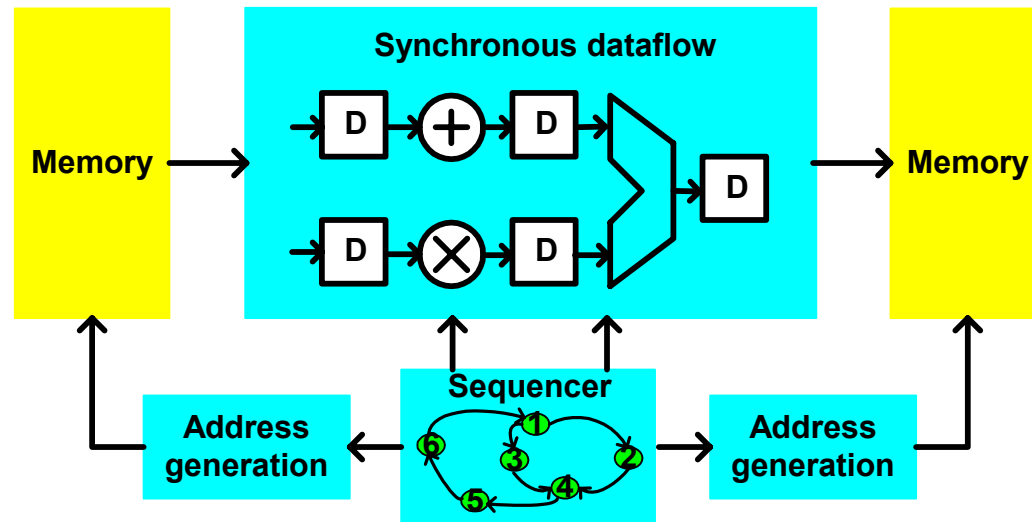
Terminal challenges 3:

- The idea of hierarchical routing.
 - Regions of 50 - 100 kgate modules
 - Routing to connect the adjacent regions.
- Routing & Power dissipation
- Distance to memory longer than in local region, increased capacitance, more power.
- Thus from power dissipation perspective it would be wise to try to reduce the amount of memory accesses.
- Pipelining principle, data processed in local registers instead of distant memory.
- Algorithm design issue.



Impact of small process geometries on microarchitectures in systems on a chip. D. Sylvester, K Keutzer. IEEE 2001

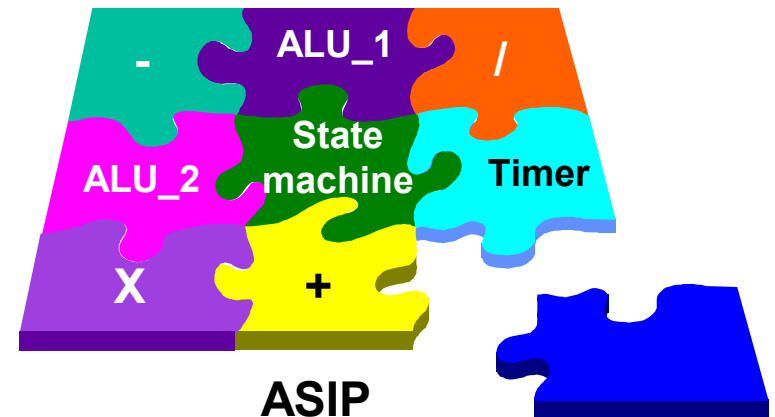
Traditional design vs processor generators:



- **Traditional way:**
 - Memories & Address generators
 - Datapath
 - A sequencer for controlling the datapath processing.
- **What is the added value of manual optimization at system level?**
- **Processor generators:**
 - Automated architecture design.
 - User specifies the proc elements.
 - Sequencer & Address generators.
- **User programs the sequencer.**

Processor generator utilization:

- SW designers have their view, how things should be implemented.
- HW designers have that also.
- Designer using processor generator should understand a bit of both worlds:
 - Predicated instructions, flags
 - SW pipelining
 - The use of timers
 - Peripherals
 - Where to locate state machines? SW or configurable HW
- Processor generator helps the designer to make more flexible architectures.
- Crosstechnology expertise is required.
- Processor design methodology needed.

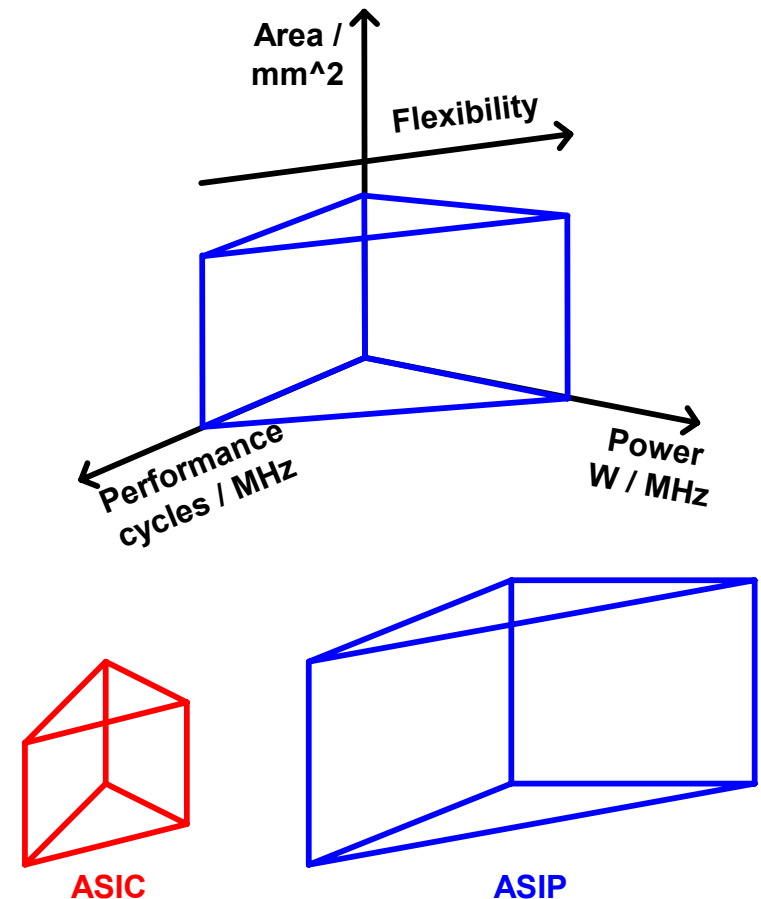


What is inside processor?



Architecture comparison:

- Typical comparison based on:
 - Performance (cycles/MHz)
 - Power dissipation (mW/MHz)
 - Silicon area (mm²)
- Flexibility axis, that covers the angle between performance and power axes.
- ASIC is certainly the most optimized realization method.
- What is the cost penalty of flexibility?
- Where flexibility is needed in the system for cost efficient realization?



LISA vs TTA

LISA: Language for Instr. Set Arch

- Investigated at ISS RWTH Aachen.
- <http://www.iss.rwth-aachen.de/Projekte/Tools/LISA/lisa.html>
- Higher level language to describe processor functionality.
- Designer must have processor design expertise.
 - How to design a processor?
 - Which parts / modules to include?
 - RISC, SIMD?
- Manual architecture exploration
 - Time consuming process
- C compiler under research
- Compiled simulation approach reduces the verification times.

TTA:

- Single instruction type: data move
- Automated architecture exploration
 - Performance
 - Silicon area
 - Power dissipation
- There is no need for the user to be processor design specialist but application specialist.
- C compiler and scheduler in research phase.

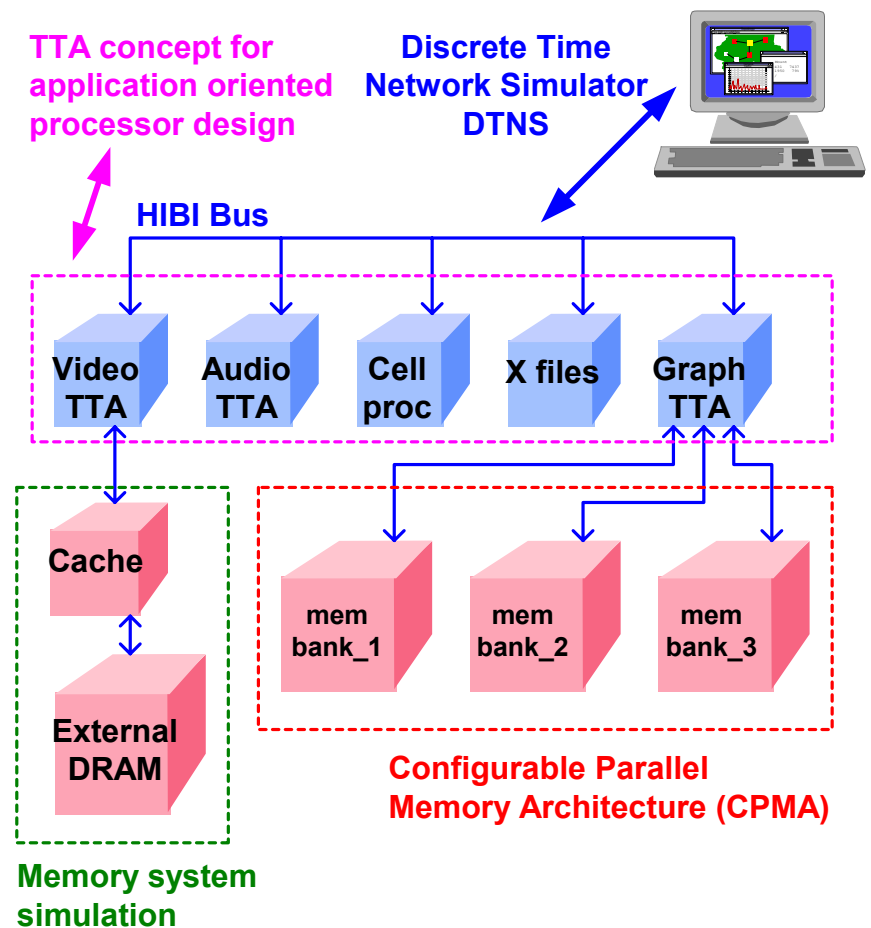
Summary:

- Automated exploration does not exist in current commercial systems.
- Processor design expertise not needed with TTA.

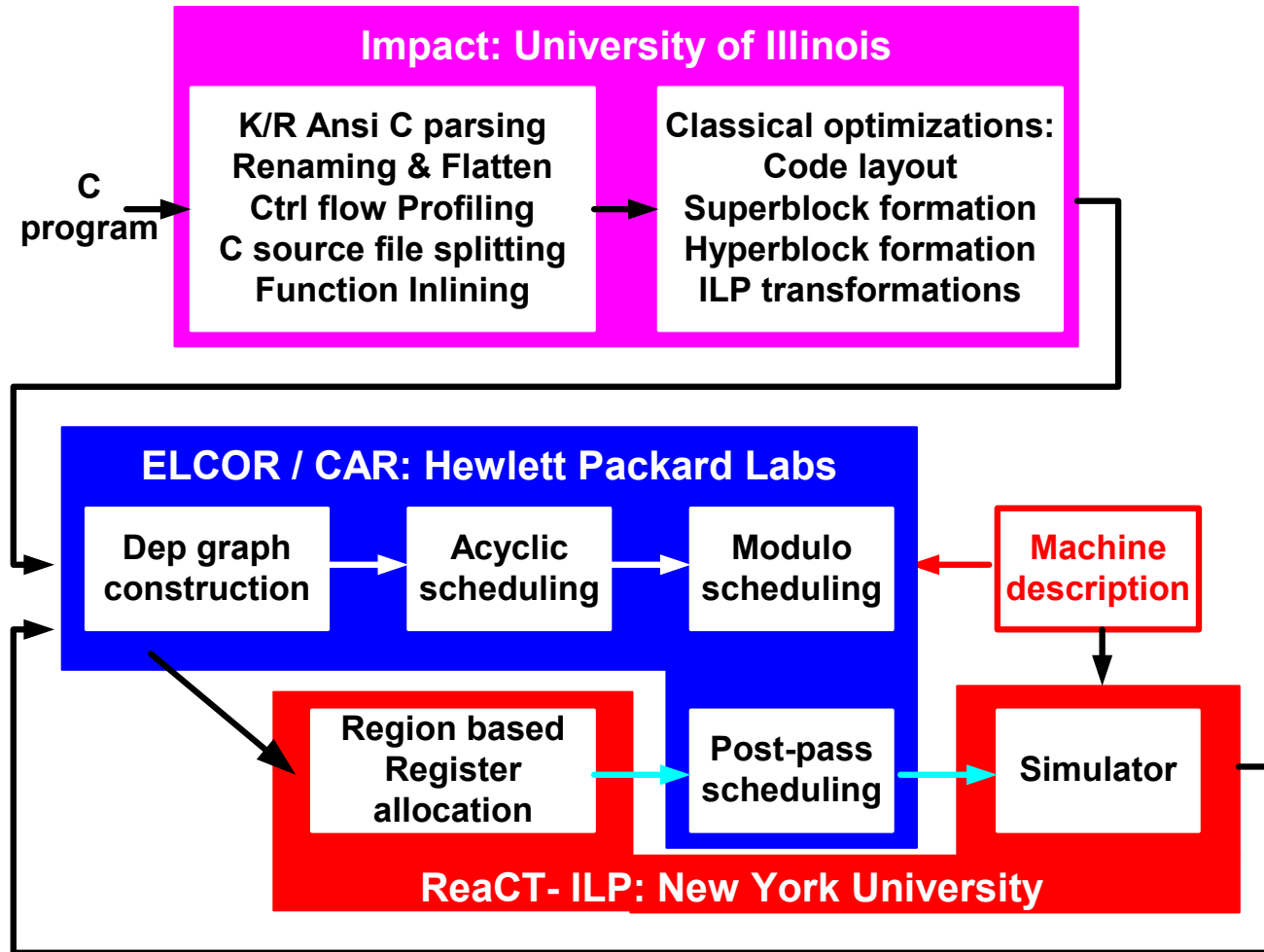
NOKIA

TUT research & TTA

- A lots of potential ideas has been developed in earlier research projects with TUT.
 - HIBI bus system
 - Discrete time network simulator
 - Configurable Parallel Mem Arch
 - Memory System simulation tool
- HIBI a higher level design framework:
 - TTA can be applied for application oriented processor design.
- Valuable thing would be, if the dedicated ideas developed in different projects could be merged someday.
- In ideal situation, it would be nice to integrate some things to environment developed outside TUT.



Trimaran consortium

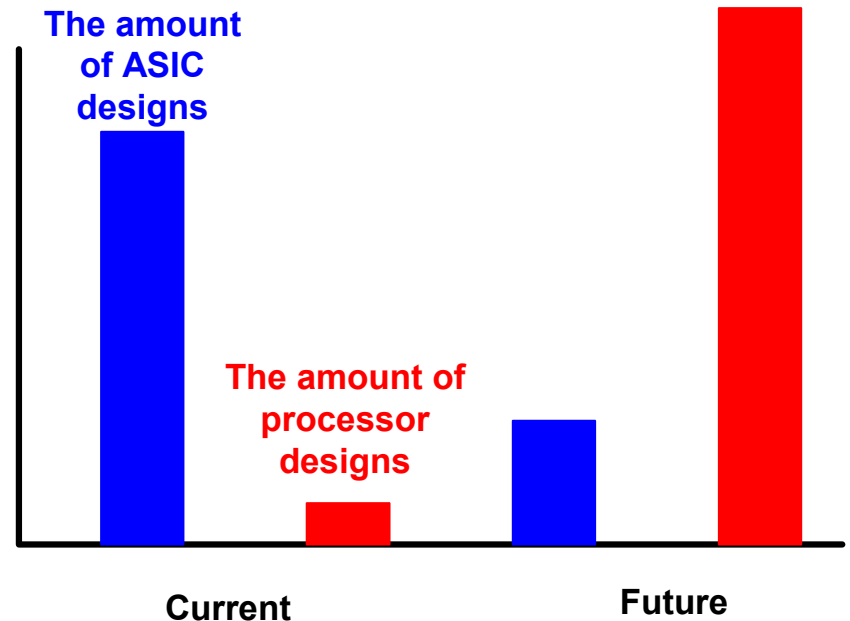


[Http://www.trimaran.org](http://www.trimaran.org)

Something similar as an outcome of FlexDSP?

Summary: According Keutzer

- Bright future for programmable processors. Kurt Keutzer. IEEE Design & Test of computers. Nov-Dec 01.
- Need for programmable platforms.
- EDA corps revenue based on the amount of design seats.
 - Thus static timing analysis,
 - Formal equivalence checking
- Better revenue by mainstream technology
- Need for Design Space Exploration.
 - Discipline for architecture design
 - Benchmarks
 - Automated tool generation from higher level languages.
- The balance in EDA industry is likely to shift towards processor design tools.



The FlexDSP project is the ringside place to follow this paradigm shift.