# Customized SIMD Unit Synthesis for System on Programmable Chip - A Foundation for HW/SW Partitioning with Vectorization

M.O.Cheema O.Hammami

Embedded System Design Group
ENSTA, Paris, France
{cheema|hammami}@ensta.fr

# Extensible Processors and Instruction Set Extensions (ASIPs)

Infineon technologies

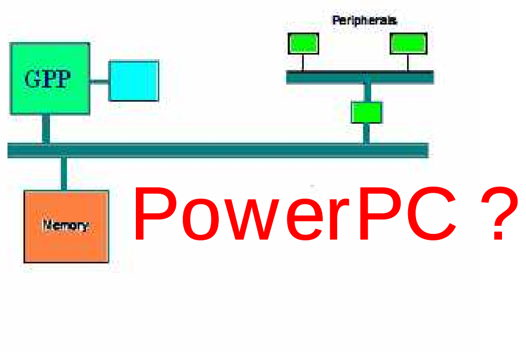
Carmel20xx

tensilica

Xtensa

ST

ST200




MIPS TECHNOLOGIES

CorExtend

ARC

ALTERA

Nios II

Strong emergence of various commercial platforms: <u>none based on PowerPC</u>

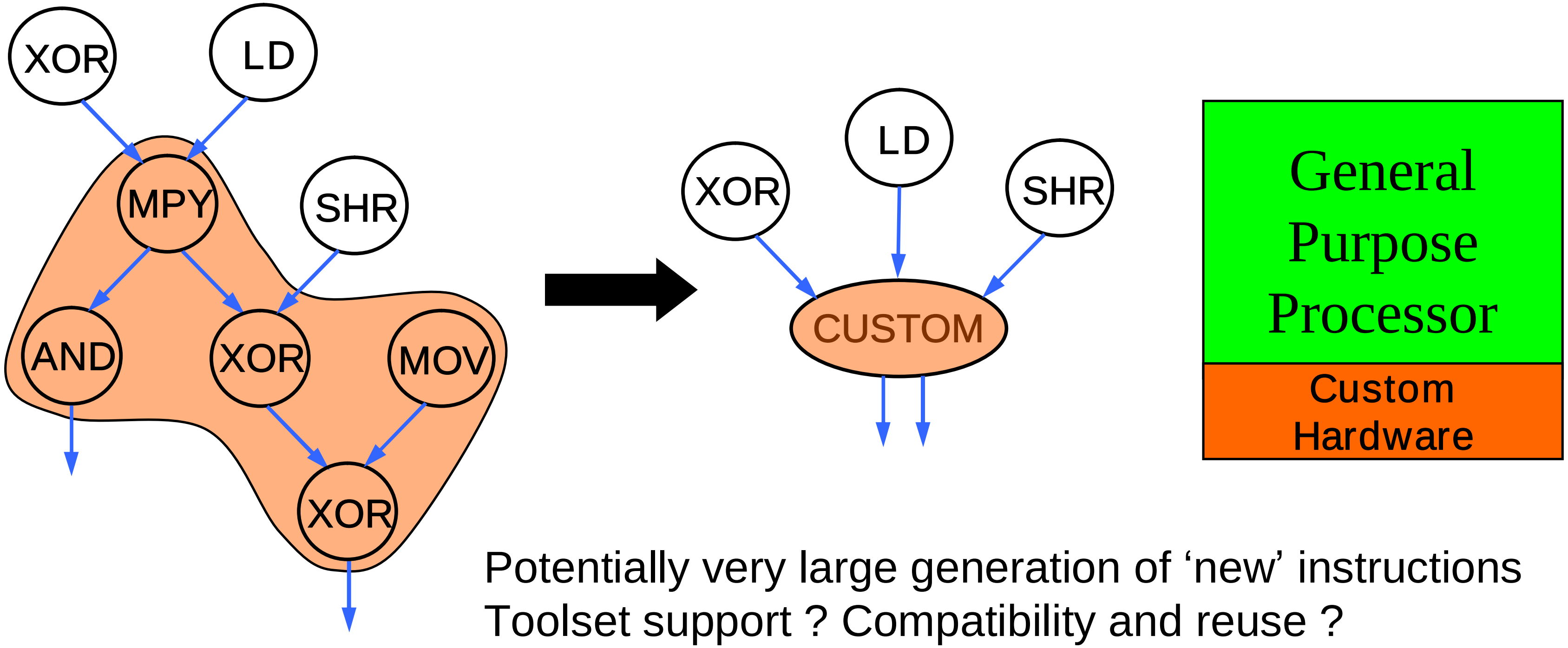
Academia: customized instructions extensive research last decade (recently e.g. Brisk and al.2005, Ienne and al 2003, Jha and al. 2003)

# SIMD Instruction Set Extensions and SIMD Synthesis

- GPP + SIMD extensions:
  - Emergence of multimedia workloads triggered SIMD extensions in GPP: AMD's 3DNow! , Motorola/IBM's AltiVec, Intel's SSE/SSE2/SSE3
  - Important software development environment: compiler with auto-vectorizing support (e.g. loop vectorization, intrinsics ) , libraries (e.g. intel IPP ), tools support
- SIMD Synthesis : Customization of SIMD Units (e.g. [Togawa and al. 2003], Tensilica Xpres Compiler 2004)
- Specialization of existing processors vs design of complete ASIP avoid complexity of complete toolset development
- Standard SIMD instruction set: fixed number of inputs and outputs
- How can we reuse GPP efforts in SIMD for embedded processors ?

# Motivation # 1:Traditional Instruction Set Synthesis

- Demanding parts of applications run on special hardware
- New instructions use the special hardware
- Compiler support




Potentially very large generation of 'new' instructions
Toolset support ? Compatibility and reuse ?

# Motivation #2 : SIMD Unit Utilization low

## Report for AltiVec System (G4) Image processing Filters Application

| | Filter v1 | Filter v2 | Filter v3 | Filter v4 |
|---|---|---|---|---|
| Instruction/Cycle | 0.8615 | 0.8483 | 0.6703 | 0.8465 |
| FXU1 Idle Time | 53.28% | 58.44% | 45.46% | 54.09% |
| FXU2 Idle Time | 76.36% | 70.41% | 64.18% | 75.89% |
| FPU Idle Time | 100.00% | 100.00% | 100.00% | 100.00% |
| VAUS Idle Time | 99.27% | 99.32% | 100.00% | 99.25% |
| VAUC Idle Time | 93.90% | 93.23% | 92.83% | 93.77% |
| VAUF Idle Time | 100.00% | 100.00% | 100.00% | 100.00% |
| VPU Idle Time | 100.00% | 91.87% | 100.00% | 90.76% |
| SYS Idle Time | 91.90% | 92.52% | 97.98% | 91.74% |
| LSU Idle Time | 56.01% | 61.16% | 49.73% | 67.42% |
| DL1 Hit Rate | 98.52% | 98.72% | 97.18% | 98.36% |
| IL1 Hit rate | 99.82% | 99.84% | 99.54% | 99.82% |
| Branch Prediction | 93.45% | 93.45% | 94.91% | 93.45% |

# Motivation #3 : SIMD Unit Utilization is Data Size Dependent for same function

Report for AltiVec System (G4) Image processing Filters Application

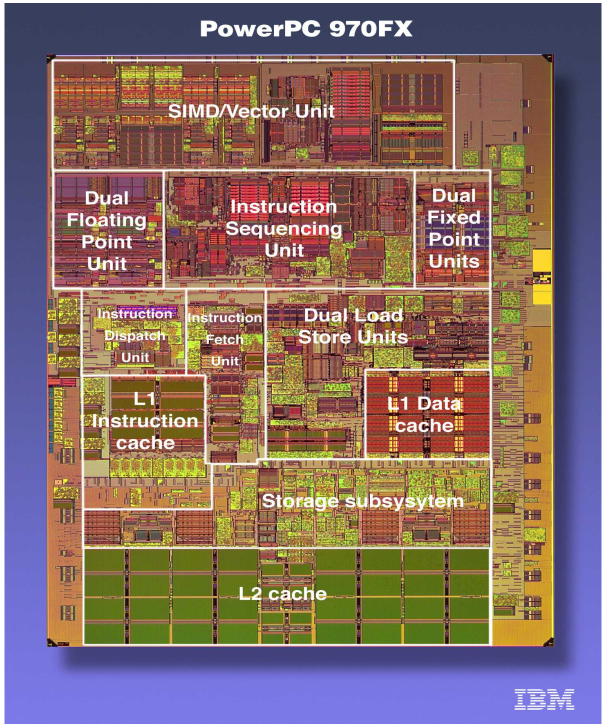
| Image Size | 32x32 | 80x80 | 800x800 | 8000x8000 |
|---|---|---|---|---|
| Instructions/Cycle | 0.7495 | 0.7319 | 0.4591 | 0.3975 |
| FXU1 Idle Time | 31.97% | 34.81% | 84.93% | 95.80% |
| FXU2 Idle Time | 54.03% | 55.95% | 90.27% | 98.49% |
| FPU Idle Time | 100% | 100% | 100% | 100% |
| VAUS Idle Time | 100% | 99.80% | 96.03% | 95.75% |
| VAUC Idle Time | 99.99% | 98.31% | 66.94% | 64.61% |
| VAUF Idle Time | 100% | 100% | 100% | 100% |
| VPU Idle Time | 99.98% | 97.89% | 57.83% | 48.80% |
| SYS Idle Time | 97.13% | 97.26% | 99.46% | 95.07% |
| LSU Idle Time | 67.23% | 68.47% | 87.50% | 90.43% |
| DL1 Hit Rate | 97.37% | 97.06% | 84.71% | 59.08% |
| IL1 Hit rate | 99.38% | 99.40% | 99.84% | 100.00% |
| Branch Prediction | 95.18% | 95.17% | 96.22% | 99.79% |

# Motivation # 4 : Hardware Implementation Cost of SIMD Units

## Report for AltiVec Units

AltiVec Integer Unit implementation:

- Approx. 22000 Xilinx Virtex-4 slices
- 90 % area of a Virtex-4 FX60
- 450 % of Virtex-4 FX-12

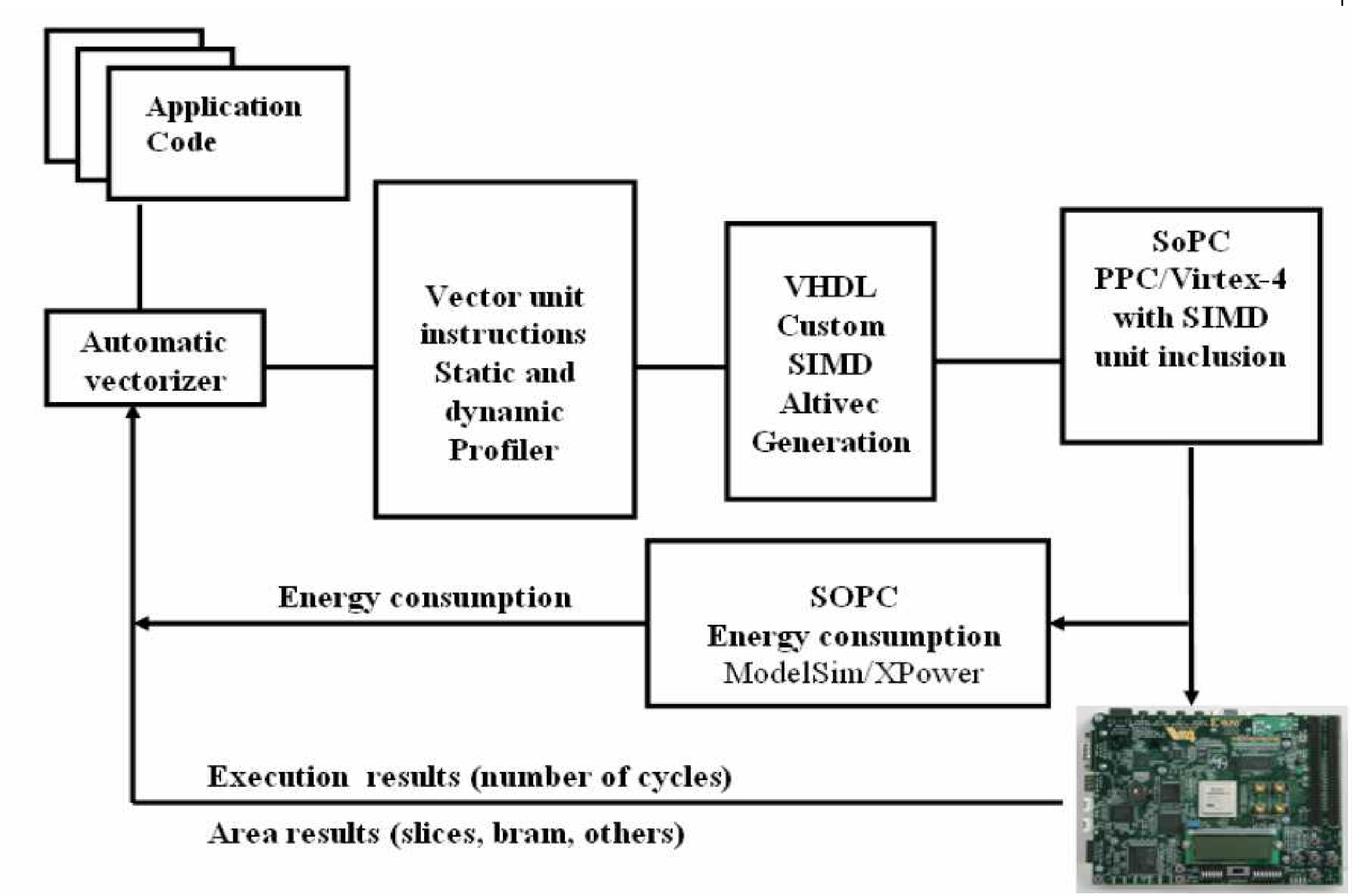



## Our Goals:

Our Goals:

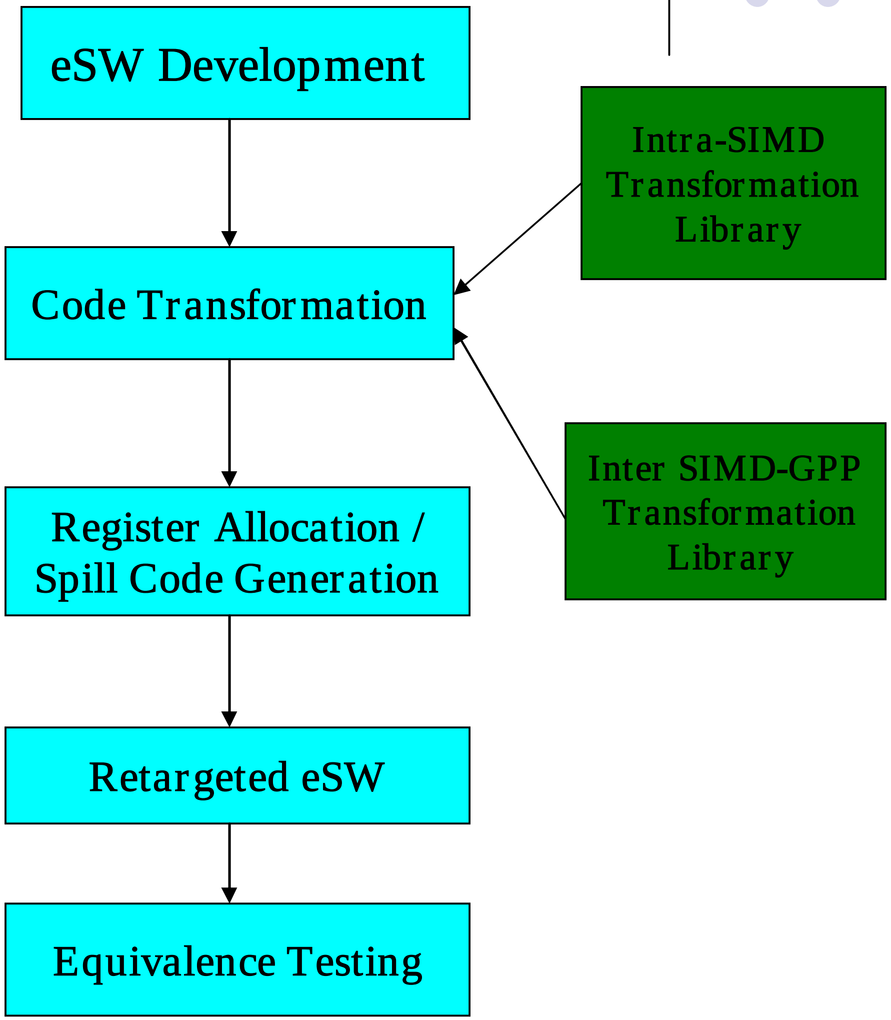
- To propose a methodology for SIMD Instruction Set synthesis reusing Compilers, Tools, IP Libraries, Programming Environment Support, Compatibility
- Efficient Utilization of SIMD Units
- Automatic Synthesis Flow
- Potential for dynamic reconfigurability/Custom SIMD unit per (function/application/data size)

# Proposed System Synthesis Flow



# Vectorization and Software Synthesis

- Embedded Software development approaches
  - General Purpose Processor Instructions
  - Extended Instruction Set (Standard SIMD extension)
- Vectorization: gcc 4.0. and VAST Optimizer
- Our approach
  - Instruction Decomposition
  - Equivalence Classes
    - Intra SIMD Equivalence
    - Inter SIMD-GPP Equivalence

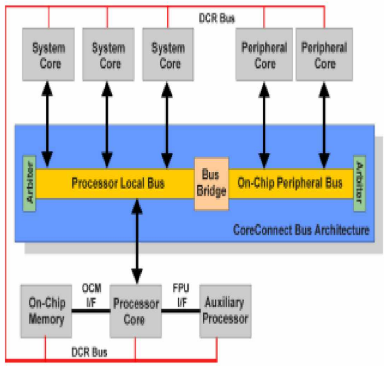


# Dynamic and Static Profiling

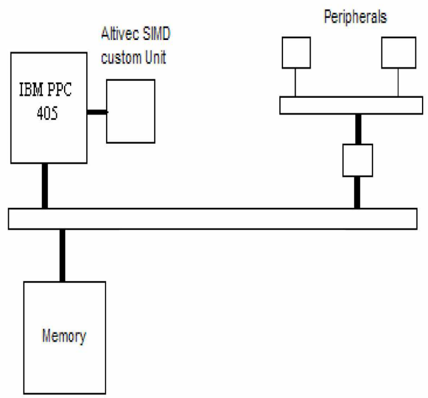
- Profiling Tools: Shark, MONSter, simg4 for AltiVec/PowerPC
- Information about:
  - Instruction Set being used
  - Frequency of instructions being used

# Hardware Generation

- AltiVec Instruction Library: RTL implementation in VHDL
- Automatic Top Level Module Generation
  - Importing Used instructions
  - Commenting out unused logic
- Fixed IBM Coreconnect Communication Architecture

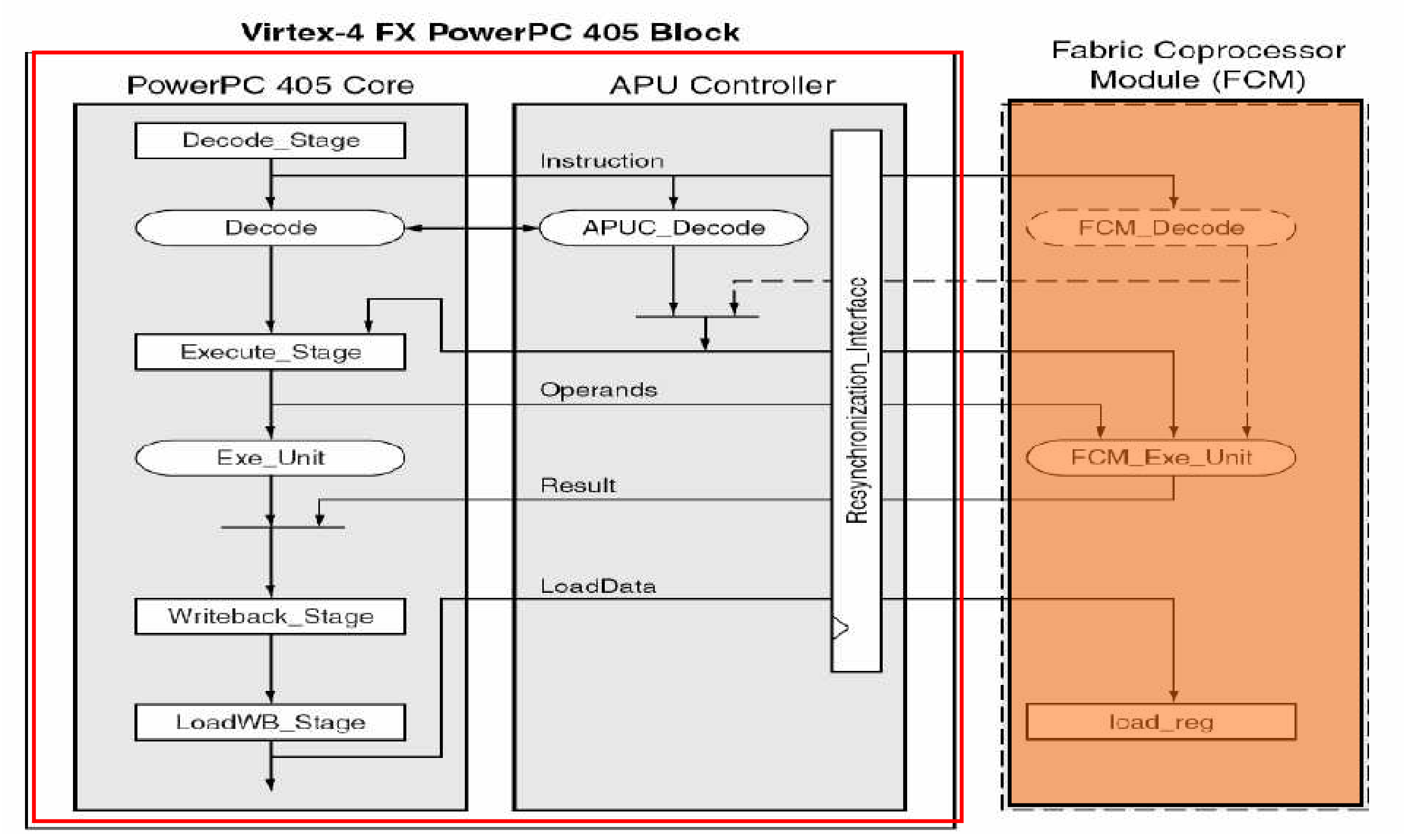


CoreConnect Block Diagram



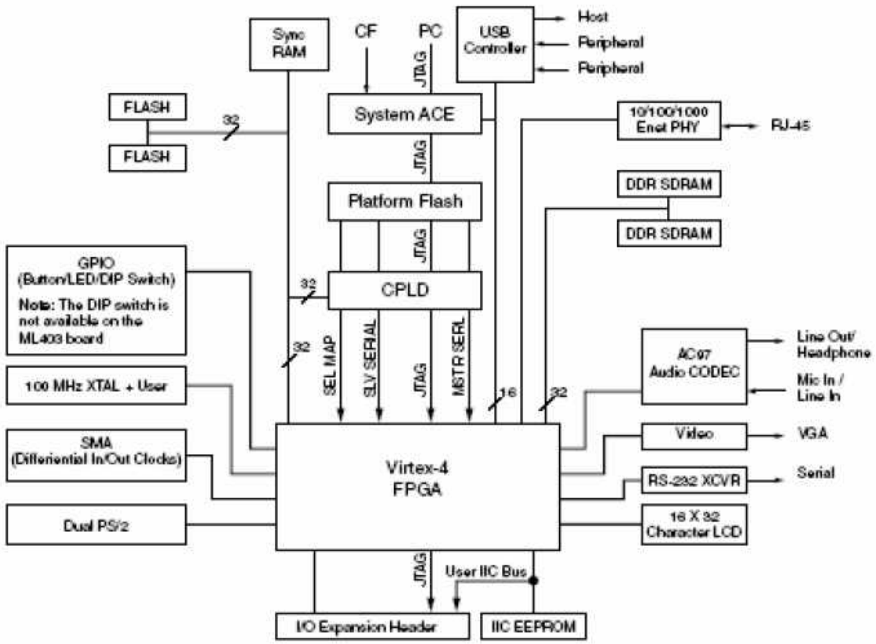
# Case study: PPC 405 hardcore SIMD Extension

- New coprocessor support in newly released Xilinx Virtex 4: APU controller
- Instruction sequencing properties: Blocking / Non-Blocking

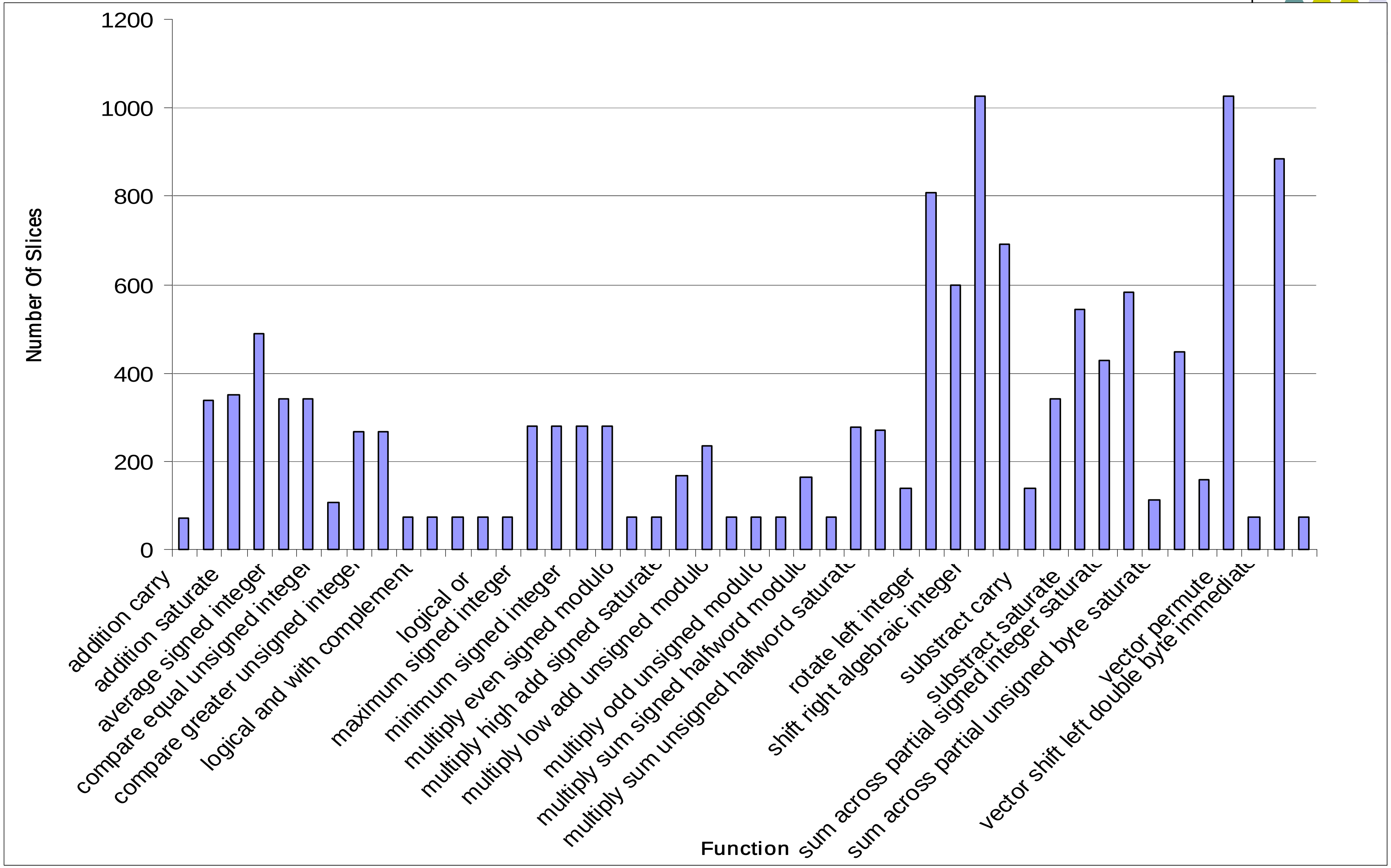


# SOPC Testbench : Xilinx Virtex-4 based ML403 board

- PPC405 hard processor core and SIMD extensions fully synthesized and placed and routed using Xilinx CAD tools: Xilinx EDK7.1 (system design) , ISE 7.1 (VHDL and IP Synthesis, P&R) : area results
- Performance results : Actual  Execution

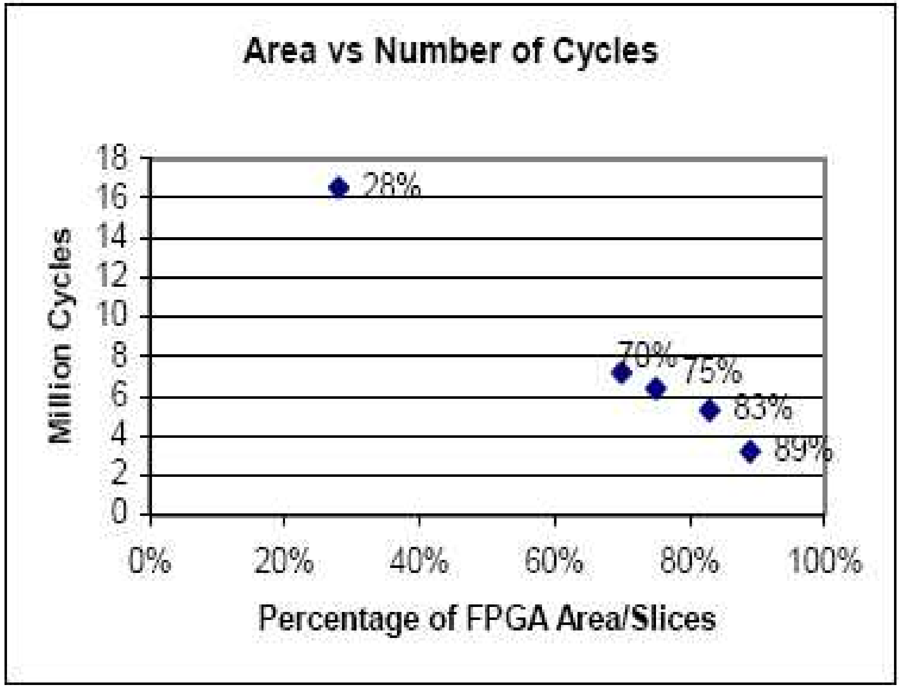



# AltiVec implementation cost



# SIMD Synthesis Results: Matrix Transpose

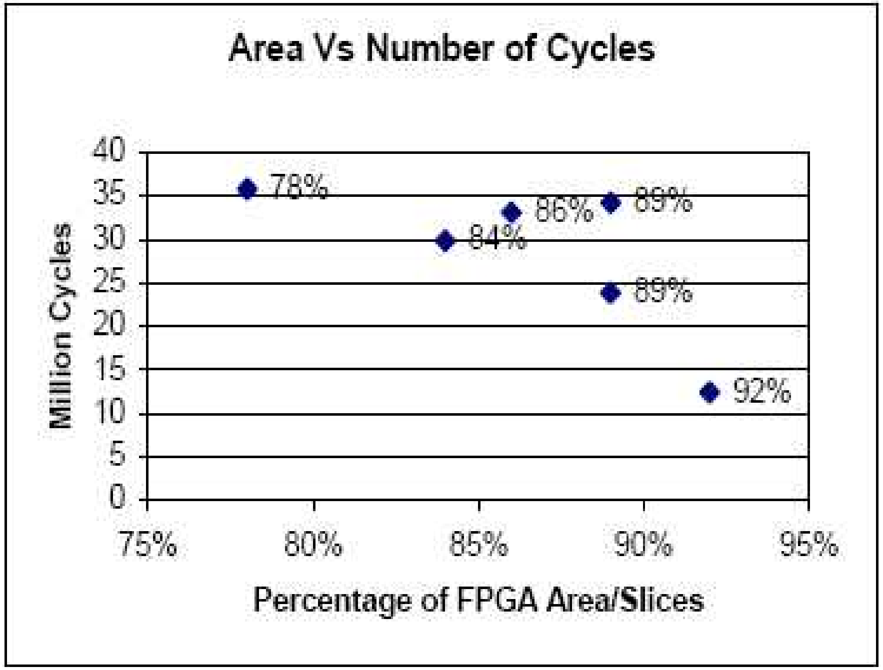
| Config. No. | FPGA Area | Time (cycle) | Speedup over Non-SIMD Code |
|---|---|---|---|
| Config. 1 | 89% | 3 171 944 | 5.2 |
| Config. 2 | 83% | 5 275 383 | 3.1 |
| Config. 3 | 75% | 6 357 824 | 2.6 |
| Config. 4 | 70% | 7 188 232 | 2.3 |
| Config. 5 | 28% | 16 534 108 | 0 |




- Complete SIMD Implementation Statistics
  - Speed up: 5.2
  - Time Cycles 3171944
  - FPGA 450%

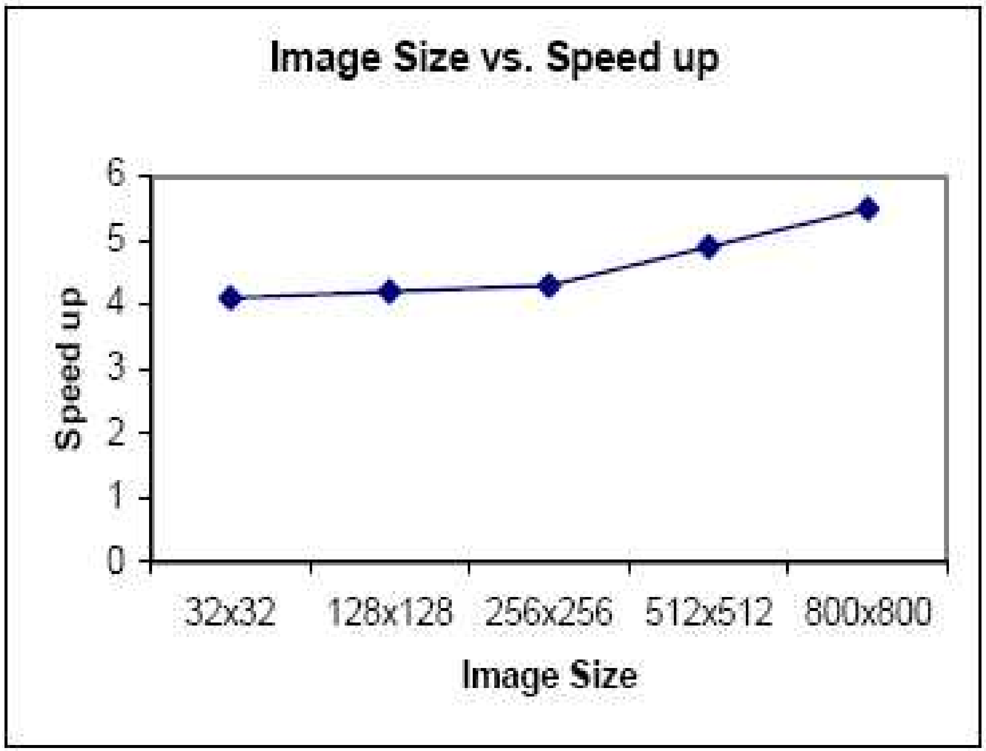
# SIMD Synthesis Results: Average Filter

| Config. No. | FPGA Area | Time (cycle) | Speedup over Non-SIMD Code |
|---|---|---|---|
| Config. 1 | 84% | 29 812 080 | 2 |
| Config. 2 | 86% | 33 087 428 | 1.8 |
| Config. 3 | 89% | 23 853 953 | 2.8 |
| Config. 4 | 89% | 34 237 811 | 1.8 |
| Config. 5 | 92% | 12 388 586 | 4.9 |
| Config. 6 | 78% | 35 770 207 | 1.7 |
| Config. 7 | 28% | 60 810 431 | 0 |



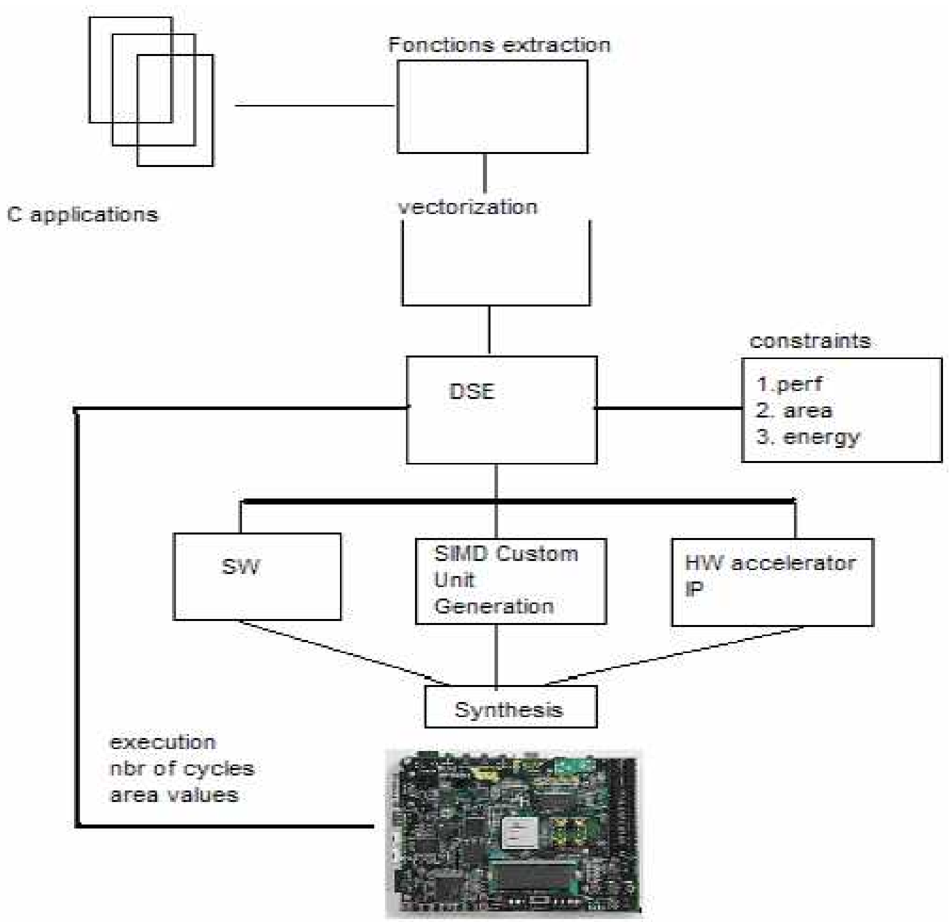

- Complete SIMD Implementation Statistics
  - Speed up: 4.9
  - Time Cycles 12388586
  - FPGA Area: 450 %

# Synthesis Results: Data Dependent Application Behavior




Can we take into account through vectorizer options
Area vs speed up benefit based on program data structures ?

# HW/SW With Vectorization



# Conclusions

- Customized RTL VHDL SIMD Unit Synthesis (Altivec compatible)
- Efficient use of SIMD unit area with designer choice for area-performance tradeoff
- SOPC Implementation on reconfigurable Xilinx Virtex-4
- Xilinx Virtex-4 partial reconfiguration opens up custom SIMD Units per function
- Vectorization technique improvements can have direct impact on SOC area using Instruction Set extensions (not visible in GPP fixed area)
- Hw/Sw partitioning should systematically include custom SIMD design space exploration whenever possible
- Physically aware Hw/Sw partitioning for reconfigurable architectures with partial dynamic reconfiguration
- Future work: extension to MPSOC

# Thank You
# Questions?