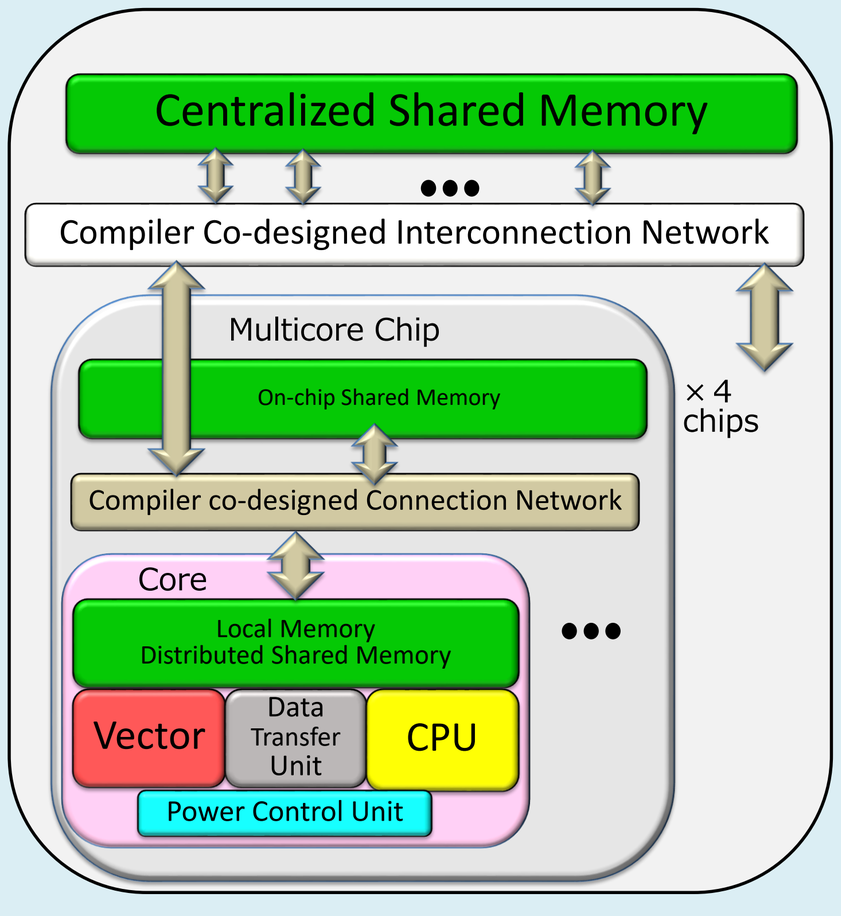
OSCAR Vector Multicore System
Platinum Vector Accelerator on FPGA

Kasahara & Kimura Lab, Waseda University, TOKYO
http://www.kasahara.cs.waseda.ac.jp

- OSCAR Compiler
- Vector Accelerator
- Clang/LLVM

Platinum Multicore Architecture



Compile Flow

OSCAR compiler can generate C code for Accelerator Automatically

Sequential C Programs

OSCAR Compiler (C to Parallel C)
- Multi Grain Parallelization
- Local Memory Management
- Data Transfer Optimization
- Power Management
- Auto Vectorization

Host side code (OSCAR API) / Accelerator side code (Using Intrinsic function)

Accelerator Compiler
- Clang / LLVM based
- Acc. C code to binary

OSCAR API Translator
- Convert architecture independent directives to architecture specific code

Host CPU Compiler
- Nios II compiler

Executable File
- Download to FPGA board using Nios II toolchain

SC18 members

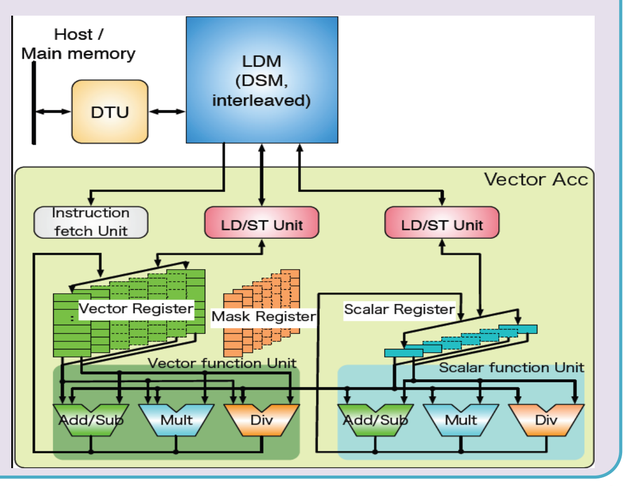
K. Miyamoto
T. Kawata
K. Takahashi
T. Kashimata
B.A. Adhi
Y. Minato
H. Mikami
Prof. T. Kitamura
Prof. K. Kimura
Prof. H. Kasahara

Vector Accelerator / DTU

Features
- Attachable for any CPUs (Intel, ARM, IBM)
- Data driven initiation (sync flags)

Function Units [tentative]
- Vector Function Unit
  - 8 double precision ops/clock
  - 64 characters ops/clock
  - Variable vector register length
  - Chaining LD/ST & Vector pipes
  - Vector Length = 256 element
- Scalar Function Unit

Data Transfer Unit
- Overwrap execution w/ data transfer (Currently using DMA)



FPGA Implementation

Board: C5P development kit (intelFPGA Cyclone V / 301K LE)
Specification:
- 16 single precision ops/cycle
- Local Data Memory Bandwidth 32 byte/clock
- All data located on Local Data Memory
- Local Data Memory size: 32KB

Performance on FPGA

Applications
- Swim (SPEC 95)
  - Size 257x257
  - Iteration = 10 times
  - Evaluation except 1st iteration (initialization)

Hardware
- CPU (NIOS)
  - Compiler: nios2-elf-gcc
  - FPU: Floating Point Hardware 2
  - Cache: 32KB
- 1CPU + 1VA / 2CPU + 2VA
  - No cache
  - Local Memory: 32KB
  - DMA transfer between DDR and Local Memory controlled by OSCAR Compiler

Result
- 3.5 times speed up w/o initialization
- Execution time of CPU part increases w/o cache