**SoC** are multi-core heterogeneous computing platforms

- No good tools for programming them efficiently (debugging, instrumentation) exist
- writing parallel programs the old way (by using pthreads) does not scale
- Programming for these platforms is hard and prohibits efficient use of all available compute resources

**Support many-cores and heterogeneous configurations for applications**

- We propose to adopt TaskSuperscalar (TSS) programming models
  - OmpSs, OpenMP 4.0, StarPU ...
  - Break down large programs into a connected mesh of small tasks; running on heterogeneous group of processors, use dataflow dependencies for synchronization (instead of barriers)
  - Out-of-execution of tasks depends on inter-task dependencies

**Task definition, creation and submission**

- Sequential code with pragmas to convey additional information to construct the data-flow graph
- Record dataflow dependencies as tasks are submitted

**Running small tasks**

- Run-time system introduces a fixed, non-negligible overhead for task creation, task submission, task issue and dependency management
- HW support, acceleration, for critical functions, for example task scheduling and dependency resolution
- Manage helper tasks (preload data for accelerators)

---

**Heterogeneous Compute Platform Architecture**

- Dig
- LITTLE
- TiGa

**TiGa Task-Graph-Accelerator**

- Task Graph Accelerator key parameters
  - Estimated die area: \( \leq 0.1 \) mm\(^2\) (TSMC 16FF)
  - Clock speed: \( \geq 800 \) MHz
  - Capacity:
    - 512 active tasks
    - 4096 graph edges (data flow dependencies) on up to 512 variables
  - Design has been implemented and demonstrated on a ZEDBOARD/ZYO (XILINX Zynq) platform

---

**Software Architecture**

**Application Level**

- Compiler Support
  - OmpSs = GNU compiler
  - CLANG = LLVM compiler
- Task Graph Accelerator
- HW Runtime Accelerator

**Hardware Acceleration**

- Tioga Task Graph Accelerator
- OmpSs
- OpenMP
- C/C++ Sequential Code

**DiscoPop**

- Automatically parallelize
- Identify CU tasks (computational Units)

---

**Performance (run on Xilinx Zynq platform)**

- 2 cores = 1.20 GHz; Tioga = 850 MHz

**Kastors Benchmark**

- The KASTORS benchmarks suite was designed to evaluate OpenMP 4.0 task dependencies
- Modified state-of-the-art OpenMP 3.0 benchmarks and data-flow parallel linear algebra kernels to make use of tasks with dependencies
- KASTORS can also be used to evaluate performance of OpenMP implementations of task dependencies compared to global taskwait-based approaches

**Toy Benchmarks**

- Synthetic: 16 benchmark sets
- Different number and type of dataflow dependencies

---

**Literature**


---

**Heterogeneous Compute Platform Architecture**

- Dig
- LITTLE
- TiGa

**Heterogeneous Compute Platform Architecture**

- Dig
- LITTLE
- TiGa

**Software Architecture**

**Application Level**

- Compiler Support
  - OmpSs = GNU compiler
  - CLANG = LLVM compiler
- Task Graph Accelerator
- HW Runtime Accelerator

**Hardware Acceleration**

- Tioga Task Graph Accelerator
- OmpSs
- OpenMP
- C/C++ Sequential Code

**DiscoPop**

- Automatically parallelize
- Identify CU tasks (computational Units)