BOISE STATE UNIVERSITY

MAPLD 2004

Florida State University

A Methodology for System-on-a-Programmable-Chip Resources Utilization

Sin Ming Loo[1], Simon Y. Foo[2]

[1]{*smloo@boisestate.edu*}
Electrical and Computer Engineering Department
Boise State University
Boise, Idaho 83725

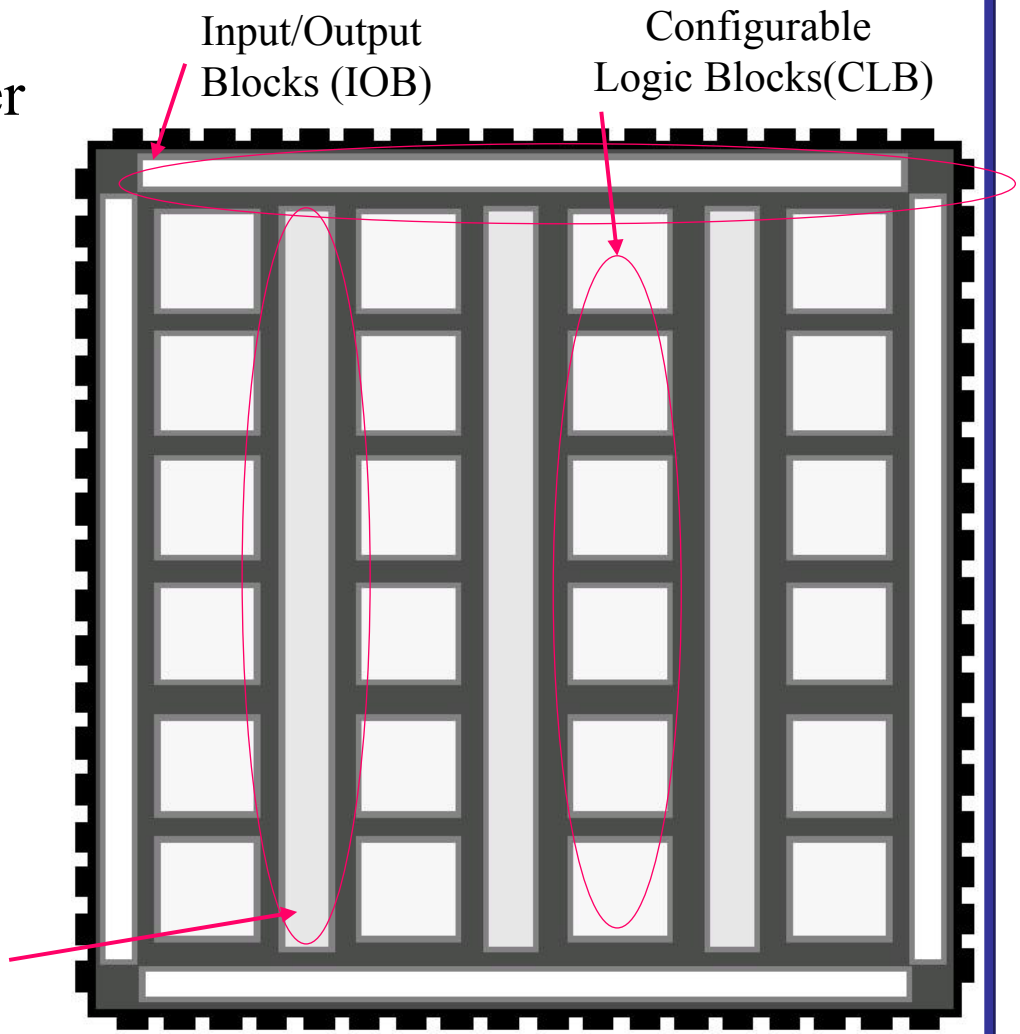
[2]{*foo@eng.fsu.edu*}
Electrical and Computer Engineering Department
Florida State University
Tallahassee, Florida 32310

Agenda

- Reconfigurable Hardware
- Reconfigurable System Design Model
- Search Techniques
- Problems Encountered and Lessons Learned
- On-going Research
- Summary

Reconfigurable Hardware (RH)

- Configured by the user after fabrication
- Provides flexibility formally found only in software based systems
- Eliminates the need to create custom hardware
- Allows specialization and customization to meet the needs of the application



Task Specific Core (TSC)

- Allows the application to be implemented in a more "optimal" fashion from a performance point of view
- Best performance would occur if the application is constructed entirely with TSC
- TSC implementation is most often highly concurrent
  - But, concurrency is not free!
- TSC implementations usually consume enormous amounts of reconfigurable resources
- RH is limited by size
  - Size limitation is unlikely to be solved by improvements in device technology
    - Due to expanding complexity of application problems

Space Time Trade-off

- How to determine an effective balance between performance and resource utilization?
  - which portions must have concurrency?
  - which portions can be implemented sequentially?

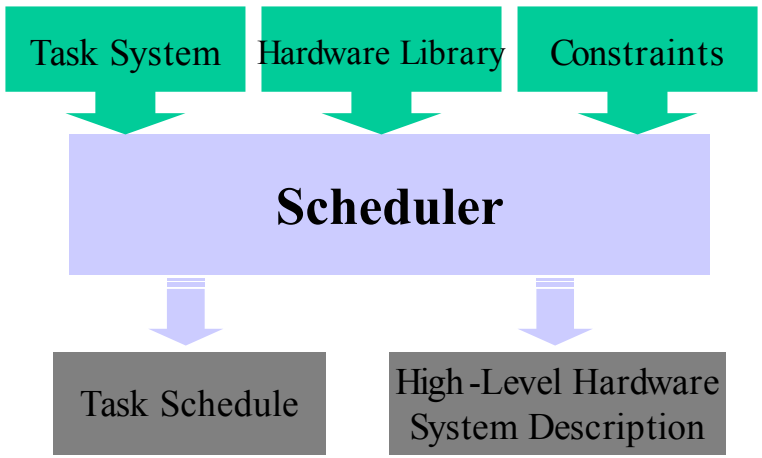
- How much concurrency can be employed without exceeding the available RH resources?

Methodology

- Employs multiple Von Neumann style processing cores + high performance task specific modules
- Allows task executes sequentially on Von Neumann type processing cores
- Employs parallel processing style static scheduling
- Insight gained through this problem can be extrapolated and extended to less deterministic problems

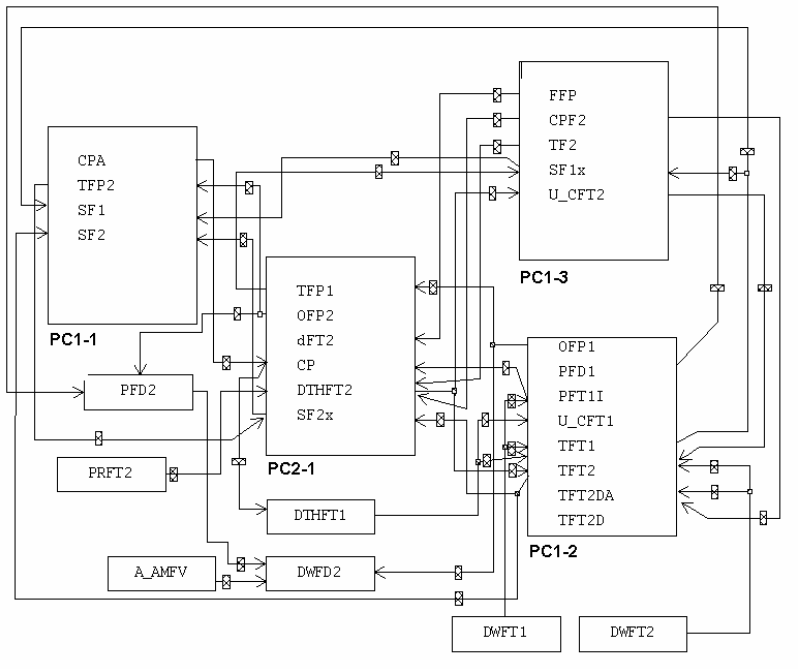
Reconfigurable System Design Model (RSDM)

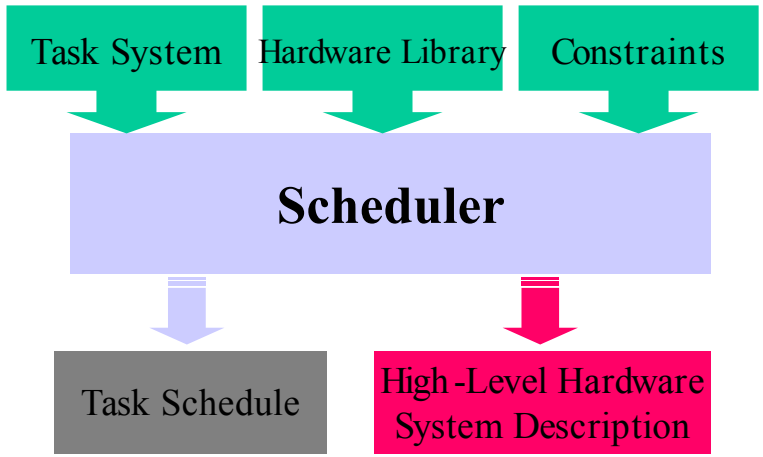
- Apply scheduling theory to a pre-fabricated RH environment
- The heart of this design model is a scheduler



- Input to scheduler:
  - *Hardware Library, Task System, Resource Constraints, Design Constraints*
- Scheduler produces:
  - *Task Execution Schedule, High-Level Hardware System Description*

High-Level Hardware System Description





- Specifies
  - the number and type of functional units used
  - number and type of communication core elements used
  - the necessary system topology

Computational Complexity

- Hard problem!
- To find the optimum solution, it requires $(\#PC + \#TSC)^n$ operations
- For example,
  - A system with 100 tasks
  - Hardware Library has 3 PCs
  - Each task has two TSC implementations
  - Scheduling operation requires 0.5 nsec
  - $3 + 2)^{100} = 7.886 \times 10^{69}$ operations
  - For 0.5 nsec/operation, it will require 1.2507×10^{53} years to find the optimum solution

Search Techniques

- Search Algorithms
  - Random
  - Simulated Annealing
  - Genetic Algorithms (Found to be the most effective technique in finding a ‘good’ solution)
  - Others (depth-first, breadth-first, branch-and-bound, etc.)
- Using synthetic task systems

Problems Encountered & Lessons Learned

- Representative task systems for reconfigurable hardware scheduling (co-synthesis) research?
  - First-cut approach, synthetic task systems
- Within reconfigurable hardware, especially Field-Programmable Gate Arrays, what percentage of resources should be utilized during scheduling
- No applications specific information are used during scheduling

Application Specific Inputs

- Designer knows the design/application the best
- Needs to be able to take in designer's view of tasks' execution location
- Mobility factor
  - Allow user to specify the execution location of a task, in IP processor or as TSC
  - All other tasks are scheduled around the 'fixed tasks'

On-going …

- Collecting resources and performance (timing) data from real-world applications
- Benchmark selected: MiBench, an embedded system benchmark
- Synthetic task systems are being used to compare the scheduling algorithms

Summary

- Presented a Reconfigurable System Design Model (RSDM)
- Collected data is available at http://coen.boisestate.edu/smloo/rsdm