ILP Project

April 13, 2009

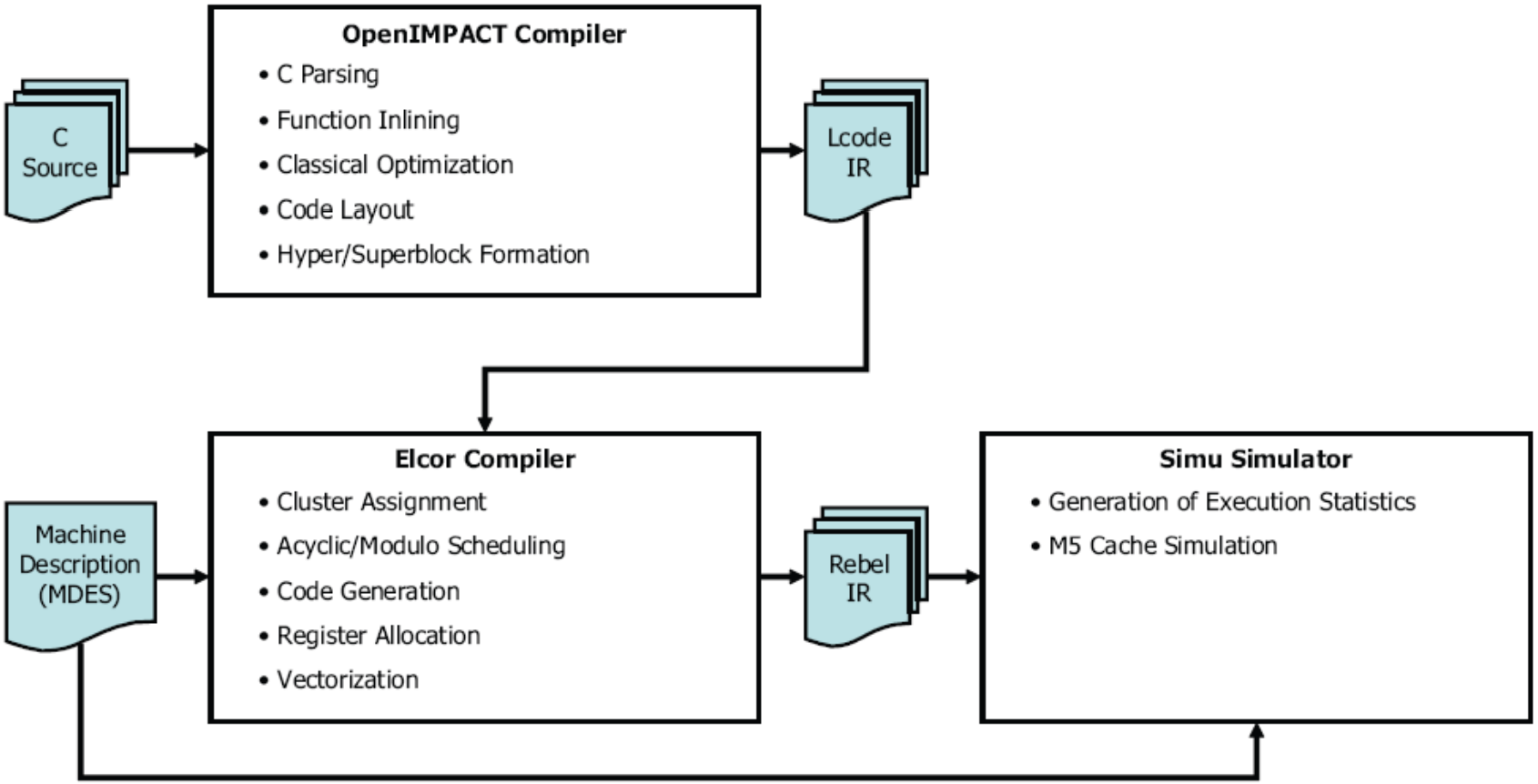
Introduction

- Suppose your company has been making the standard processor with one functional unit, which can satisfy general-computing requirements for a long time.
  - integer, floating point, memory, and branch (one for each)
- Now to expand the market share, your company wants to better appeal to the needs of the customers.
  - General-purpose users who want good performance for their money
  - Scientific users who want loads of computation power
- Please write a report to a technical manager who needs to make design decisions
  - Suggest the best mix of functional units for the 2 processors

Project Overview

- Part 1
  - How processor architecture affects benchmark performance
    - Explore the effects of varying functional units
- Part 2
  - Design 2 processors:
    - General purpose
    - Scientific purpose

Trimaran Organization



Steps in Part 1

- First logon and run the Trimaran setup script
  - *source /afs/ece/users/jowens/eec171/trimaran/scripts/envrc*
- Create a working directory in /tmp
- Copy the MDES file into your working directory
  - hpl_pd_elcore_std.hmdes2
- Modify the MDES file as desired
- Compile the MDES file
  - *hc hpl_pd_elcor_std.hmdes2*
- Run a benchmark
  - *tcc -bench mm_dyn -M/tmp/wdir/hpl_pd_elcor_std.lmdes2*
- Use the Sumstat command to review statistics obtained during the run

Modifying MDES file

- Modify num_clusters
  - $def !num_clusters 1
- Modify functional units as needed
  - // Per-cluster Functional Units
  - $def !integer_units 1
  - $def !float_units 1
  - $def !memory_units 1
  - $def !branch_units 1
- Modify float unit latency as needed
  - $def !float_multiply_latency 3
  - $def !float_divide_latency 8

Benchmarks

- mm_double: matrix multiply
  - both float and int
- fft: fast fourier transform
  - float and int
- mm_dyn: matrix multiply
  - int
- mpeg2dec: decode a MPEG2 file
  - int and float
- epic: experimental image compression
  - float and int
- pegwitenc: encryption code
  - int

Part2

- Design two processors
  - General purpose (Performance/$)
  - Scientific purpose (Performance)
- Design Constraints:
  - Up to 12 total functional units (int, float, memory, branch)
  - Incremental cost for functional units
    - 4 units are included in the $50 base price
    - Unit 5 = $10, unit 6 = $20, unit 7 = $40, unit 8 = $70, unit 9 = $110, unit 10 = $160, unit 11 = $220, unit 12 = $290
    - Floating point units
      - Slow floating point units (adds 1 cycle to mult and div latency) = saves $5 per
      - Fast floating point units (subs 1 cycle from mult and div latency) = costs $5 per

Turn In

- 2 files will be submitted via SmartSite.
  - A written report(.PDF).
    - No more than 3 pages (2 pages prefer)
  - A zip file
    - For each processor you designed, submit the MDES file and the Sumstat output for each benchmark.
    - Submit a text file named “README” that describes each of the MDES and stats files.
- **DUE DATE: Wednesday 4/22 at 5PM.**