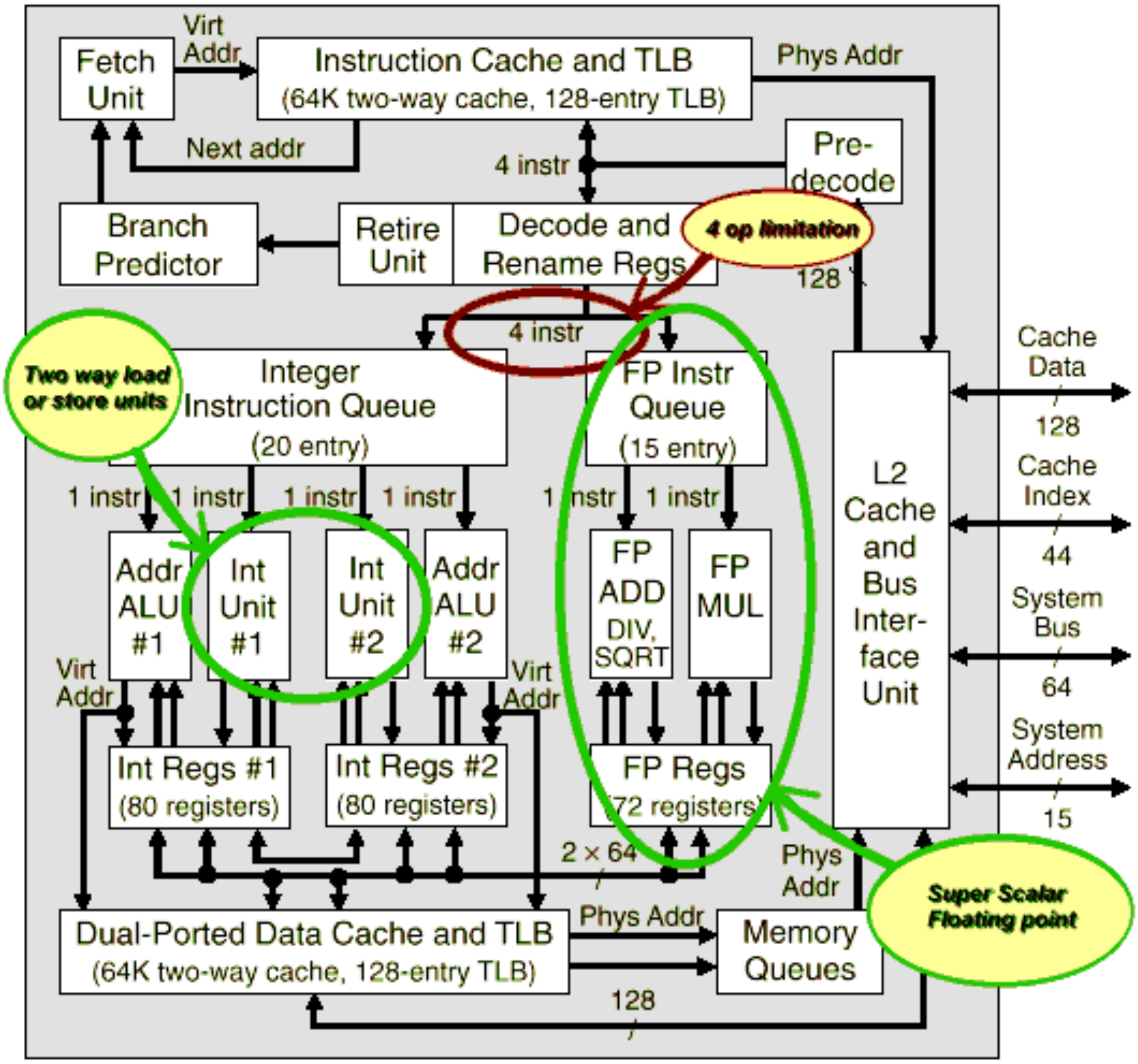
ILP Project

Architectural modifications in the DEC Alpha



The Compaq Alpha 21264

Architectural decisions in the OwensCorp Alpha

Congratulations! You've just been hired by OwensCorp as chief architect for a re-launch of the Alpha processor. OwensCorps has purchased the rights to the Alpha ISA and architecture, and after years of neglect, OwensCorp believes it can make a come back. They intend to launch four new processors next year, each targeting a different market.

1)
Your marketing team explains that with modern processor frequencies more or less stable, OwensCorp needs a new metric with which to mislead the people. They've settled on CPI. They believe they can market a low cost processor lowest cost processor with a CPI one. So, your first task is to build the lowest cost processor you can, with a CPI of at least one. This task is regardless of frequency, which marketing intends to retrain the public to think is immaterial.

2)
Your second set of customers aren't so gullible, they are however, penny pinchers. Marketing believes a significant share of the hobbyist market purchases machines based solely on performance per cost. You must design a processor that maximizes this metric.

3)
A global ISP has expressed interest in seeing some performance numbers for your processor line. Since any devices they purchase will run 24/7365 for years to come they are very concerned about power. They make purchased decisions based on performance per power and are not too concerned about device cost. The operational cost due to power is where they need to save. Design something to impress them.

4)
Marketing thinks it can sell a large amount number of processors targeted at the mobile PC market. Performance again is king here, but power is also a concern. They ask that you keep power under 40w, while maximizing performance.

Design space:

You have been given a base architecture with a simple pipeline. You can make changes to the following aspects of the architecture for each processor.

- In-order Execution or not?
- Branch-Predictor
- Fetch Width
- Issue Width
- Decode Width
- Re-Order buffer size
- Load/Store Queue

- Number of Integer ALUs
- Number of Integer Multiply/Divide Units
- Number of FP ALUs
- Number of FP Multiply/Divide Units
- L1 Cache Ports
- Cache size
- Cache number of sets
- Unified L2 cache or dual L2 Cache

Determining Performance

To judge performance you will run the same four benchmarks you ran in the warm-up exercise. The metric used will be average IPC*Frequency, this will normalize for differences in execution time between programs in the benchmark suite.

Determining Area

The original design you will modify has an area of 300units. The table below contains the area increase associated with each design choice. So, selecting an additional three floating point multiply/divide units will cost you 21 area units, and doubling the L2 Instruction cache capacity will cost you 200.

Determining Cost

The cost of the base processor is $10. Cost increases approximately proportional to area^3. So if your area goes up by a factor of 1.2, your cost goes up a factor of 1.72

Determining Power

In this exercise we will use Area as a proxy for power. Area increases in the pipeline part of the processor will result in an equally proportional power increase form the base model. Increase in the cache portion of the processor costs one fifth that of an area increase in the pipeline part.

The base processor consumes 30watts. So a design with an area increase of 50units in the pipeline and 100unit in the cache would consume 30*(1+50/300 +.2*100/300)= 37

Inputting and Simulating your design

When invoking simplesim-3.0 you can pass a configuration file detailing your design. You can find a file for the base model here:

```
/afs/ece/users/jowens/simplescalar/base171.cfg
```

You can modify the details of the architecture in the file, then invoke simplesim with the command line switch "-config filename".

The table below shows the architectural variation you should consider when designing your processors:

Pipeline Options			
Feature	Options	Additional Area	Frequency Divisor Penalty
Branch Predictor	Not-taken\taken	0	1
	Bimodal Branch Target Buffer	10	1.05
Fetch Width	1,2,4,8	0,1,3,7	1,1.02,1.04,1.06
Issue Width	1,2,4,8	0,3,6,12	1,1.02,1.04,1.06
Decode Width	1,2,4,8	0,8,16,32	1,1.02,1.04,1.06
Reorder Buffer Size	2,4,8,16,32	0,2,4,8,12	+.01 per option
Load/Store Queue	2,4,8,12,16,24,32	0,2,4,6,8,10,12	1
Integer ALUs	1,2,3,4,5,6,7,8	4 per ALU above 1	1, 1.02 <4, 1.04 <8, 1.06=8
Integer Mult/Div	1,2,3,4,5,6,7,8	5 per ALU above 1	1, 1.02 <4, 1.04 <8, 1.06=8
FP ALUs	1,2,3,4,5,6,7,8	5 per ALU above 1	1, 1.02 <4, 1.04 <8, 1.06=8
FP Mule/Div	1,2,3,4,5,6,7,8	7 per ALU above 1	1, 1.02 <4, 1.04 <8, 1.06=8
L1 Cache Ports	2,4,6,8	0,3,7,15	1,1.04,1.08,1.12
Inorder Execution	false, true	0, -15	1, .97
Memory Options		Additional Area	Latency
L1 Inst Cache	8k Direct Mapped	0	1
	16k Direct Mapped	10	2
	32k Direct Mapped	30	2
	8k 2-Way	2	2 (1 at Frequency Penalty above 1.05)
	16k 2-Way	11	2
	32k 2-Way	33	2
L1 Data Cache	8k Direct Mapped	0	1
	16k Direct Mapped	10	2
	32k Direct Mapped	30	2
	8k 2-Way	2	2 (1 at Frequency Penalty above 1.05)
	16k 2-Way	11	2
	32k 2-Way	33	2
	8k 4-Way	4	2 (1 at Frequency Penalty above 1.06)
	16k 4-Way	12	2
	32k 4-Way	36	3
L2 Inst Cache	256K Direct Mapped	0	6
	512k Direct Mapped	200	7

	256K 4-Way	16	7
	512k 4-Way	220	8
L2 Inst Cache	256K Direct Mapped	0	6
	512k Direct Mapped	200	7
	256K 4-Way	16	7
	512k 4-Way	220	8
L2 Unified	512k Direct Mapped	0	7
	512k 4-Way	20	8

Submission:

Submit two files to Canvas.

- A written report per the guidelines below
- A zip file containing the following:
  - A .cfg file for each of the four processors you designed
  - One log file containing for each processor and benchmark combination containing the summary statistics output from sim-outorder

Report Guidelines:

Audience: Your report should target a technical manager who needs to make design decisions. You can assume that your manager understands parallel architectures but doesn't know the details of your evaluation. Explain your design recommendation, then support your recommendation with data and analysis. Describe your approach and how you tackled the problem. Why do you believe your designs are the best possible?

Length: The report, including graphs and tables, should be no more than 3 pages in PDF format with 1 inch margins and no smaller than 10 point type.

Graphs: You are encouraged to include graphs to better illustrate how your experimental results support your conclusions. However, don't feel like you have to make graphs of everything. A single graph or a few graphs that emphasize important points are best.

Summary Table: The first page of your report should include a summary table of the following metrics for each of the four processors you designed:

- Performance
- Power
- Area
- Cost
- Performance/area
- Performance/cost
- CPI

Completeness: Be sure you have thoroughly read the assignment and include all information it asks for.

Getting Started Hints:

By now you should be well on your way to fluency with simplescalar. But if you are behind, try this to get started:

```
cd ~
mkdir ilpProject
cd ilpProject
setenv PATH $PATH\:/afs/ece/users/jowens/simplescalar/simplesim-3.0
mkdir benchmarks
cp /afs/ece/users/jowens/benchmarks/* ./benchmarks/
cd benchmarks
cp /afs/ece/users/jowens/simplescalar/base171.cfg ./
#now modify the config file all you want
sim-outorder -config base171.cfg anagram.alpha words < anagram.in >
myOutputFile
```