Project Proposal: On-Chip Support for ILP, DLP, and TLP in an Imagine-Like Stream Processor

James Bonanno, Suzanne Rivoire, Rex Petersen
Tuesday, 7 May 2002

Abstract

Our project attempts to determine the optimal ratio of hardware support devoted to instruction level parallelism (ILP), data level parallelism (DLP), and thread level parallelism (TLP) in an Imagine-like stream processor. We examine the performance of some typical media applications while varying the configuration of computational units within clusters (targets ILP), the number of clusters (targets DLP), and the number of independent sets of clusters (targets TLP). We develop a cost model based on silicon area, and examine the cycle counts of the benchmarks for various constant cost configurations. We also determine how the optimal ratios change with the cost budget.

1 Introduction

The best approach to exploiting parallelism in a stream processor is an open issue. The Stanford Imagine processor is designed to exploit DLP and ILP, the least expensive but also least general types of parallelism. On the other hand, the MIT Raw processor aggressively exploits TLP, at the cost of increased overhead and complexity. While a comparison of these two processors yields some insight into the advantages and disadvantages of exploiting either type of parallelism, the many other differences between the processors prevent any definite conclusion from being reached. Therefore, we propose to focus on the Imagine processor and to vary its design to exploit the three types of parallelism to differing degrees so as to draw some real conclusions about the ideal aspect ratio of the three.

Figure 1 shows the axes of parallelism as described in class.

2 Sample Applications

We will use sample applications from the MediaBench benchmark suite. We are analyzing the characteristics of these applications, and will choose them according to their suitability to the stream model, allowing exploration along the three aforementioned axes of parallelism. We hope to be able to implement parts of three or four of these applications.

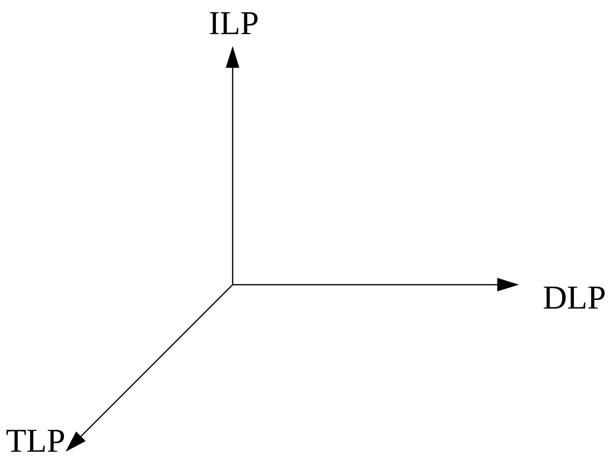

Figure 1: Axes of Parallelism

3 Cost Model

Possible cost models could be based on power, delay, and area. We will use a cost model based solely on area, taking advantage of existing models. The key point in our experiemnts is examining the performance of various configurations while keeping a cost metric constant, whatever that cost metric may be.

4 Targeting Parallelism

To target DLP, ILP, or TLP, we will have to modify the Imagine architecture; to accomplish anything in four weeks, we should try to make modifications that are already supported by the simulation tools. We plan to increase or decrease DLP by changing the number of arithmetic clusters, which the simulator supports with some caveats. We will target ILP by modifying the types of functional unit within a cluster, something that is feasible with the current tools. Finally, while we would like to model on-chip TLP, we may have to use separate instantiations of Imagine as our model instead. We are currently thinking over the advantages and disadvantages of this approach.

Our first task will be to stretch each application along each axis, to find out what the limiting factor in exploiting each type of parallelism might be. We anticipate that a careful examination of schedules and use of profiling tools will give us hints as to the bottlenecks for each type of parallelism.

Once we have performed this study and have a feel for the limitations of each type of parallelism, we will seek to determine the best configurations at a constant cost. Our figure of merit for the configurations is the cycle count of our benchmark programs. Cycle count should correspond reasonably well to execution time.

A major aspect of our project is to determine the mechanisms for supporting TLP in an imagine-like processor, that is support both space and time multiplexing of kernels.

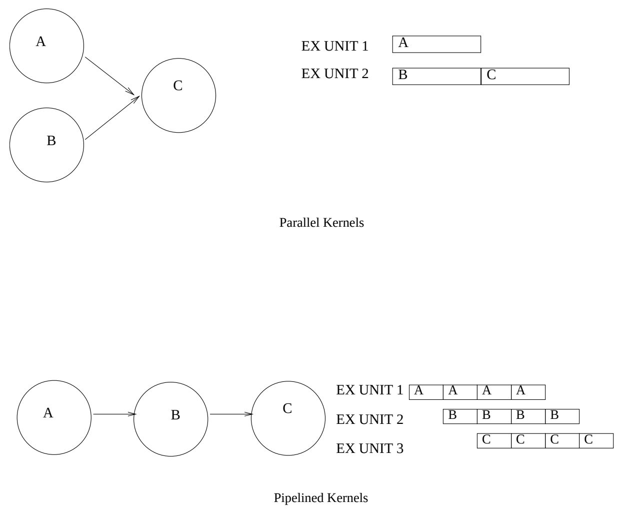

Figure 2: Uses of TLP

The current Imagine system allows support for TLP by using a network of imagine processors, however, we are considering the direct support of TLP on a single chip by placing several identical execution units on the die. These units can be used to

1. run the same kernel with different data

2. run completely independent kernels

3. pipeline the execution of kernels with sequential dependencies

The first of these is a simple use of TLP resourses for DLP. The second use is applicable for several parallel kernels which feed a common subsequent kernel. And the final approach is applicable for any sequential chain of kernels. These approaches are illustrated in figure 2.

In any of these cases, additional compiler and hardware support will be necessary to map the kernels to the execution units, and to enforce dependencies (synchronization). Another aspect that we have realized is that space-multiplexing in a hybrid space/time multiplexed stream processor can be considerably different from that in architectures like RAW which synchronizes every data transfer between kernels.

Communication among execution units can be done in one of three ways:

1. regular network structure (like RAW's grid)
2. independent SRF's (effectively TLP supported by a network of current Imagine processors)
3. single or clustered SRF shared among execution units

The third option seems to provide the greatest benefit in exploiting producer-consumer locality, however, it may not scale well with the number of execution units, and may greatly increase the cost (area) without providing acceptable returns in terms of performance.

We will choose an execution model of TLP, and will simulate it as part of our project, however, we will likely not be able to explore all of the possible aspects involved in supporting TLP.

5 Project Goals

We have two goals in this project. The first is to gain generalized insight into the relationship of DLP, TLP, and ILP for media programs. We will discover the limitations and bottlenecks of increasing support for parallelism along each of these three axes in isolation and in combination.

Our second goal is to make a specific recommendation for the ideal configuration of Imagine (for a given area) along these three axes. We would like to ascertain the best aspect ratio of DLP, TLP, and ILP.

6 Project Timeline

Week 1, 5/06 - 5/10
5/07: proposal due
Select and begin coding media applications
Develop equations for cost model
Develop execution models to adapt simulation tools for increased DLP, ILP, and TLP

Week 2, 5/13 - 5/17
5/14: project update
Finish coding applications
Start experiments

Week 3, 5/20 - 5/24
5/23: project update
Continue experiments
Analyze results

Week 4, 5/27 - 5/31
Finish collecting data, if necessary
Prepare report and presentation

Week 5, 6/03 - 6/06
6/04, 6/06: presentations
6/06 : report due