LCS: A Hardware Design Example

Greg Gibeling

9/24/2009

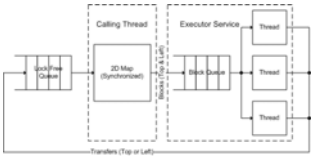
Problem & Algorithm

- The Problem: LCS
  - Longest common subsequence (discontiguous)
  - The heart of "diff"
  - In this example we compute length only
  - $O(N^2)$ Complexity, O(N) Parallelism, O(N) Space
- OPL Comp Ptrn: Dynamic Programming

```
function LCS:(X:String, Y:String, I:Integer(32) = X.Length-1,
J:Integer(32) = Y.Length-1)->Integer(32) {
        if ((I == 0) || (J == 0)) return 0;
        if (X[I-1] == Y[J-1]) return LCS(X, Y, I - 1, J - 1) + 1;
        return Integer.Max(LCS(X, Y, I, J - 1), LCS(X, Y, I - 1, J))
};
```

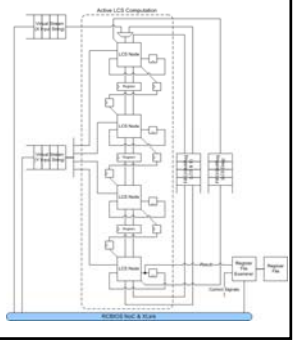
Parallel Software

- Parallel Software Impl
  - OPL Structural Pattern: Master/Worker
  - Blocking creates coarse parallelism
  - Master/worker allows load balancing
  - Rendezvous in a 2D hashmap

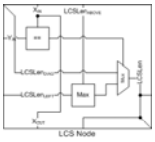


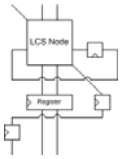
Hardware (1)

- Hardware Impl
  - "LCS Accelerator"
  - OPL Struct Ptrn: Pipeline
  - HPL: In Order Pipeline
  - Systolic Array
    - No load balancing or scheduling
  - Perfect Scaling
- Infrastructure
  - Virtual Stream (FIFO) Inputs
  - Control/Output registers



Hardware (2)





- Application Patterns
  - Pipeline
  - Dynamic Programming
- Machine Org
  - Systolic (LCS Core)
  - Heterogeneous (CPU + Acc/FPGA)
- PMS Layer
  - In Order Pipeline
  - FIFO/ShiftRegister Memory
- Infrastructure (RCBIOS)
  - Self-Timed (VS Inputs)
  - Multi-Stage Network
  - XLink Comm. Channels
  - Drill down into NoC, XLink, etc…