Lecture #7: Performance

Kunle Olukotun
Stanford EE183
Jan 29, 2003

Lab Stuff

- General Lab #2 Questions?

Performance

- In many applications, performance is the name of the game
- As long as process technology improves performance is gained by waiting and doing *nothing*…
  - We (the logic design community) benefit directly from the efforts of the Silicon process technology folks
  - 15% per year
- Need architecture improvements to use more transistors
  - 60% per year
- We want 85% improvement per year

Parallelism is the Key

- The main arrow in our quiver is ***parallelism***
  - Do more work in a given unit of wall clock time
- Parallelism is what hardware is all about
  - Focus on RTL
- Two methods
  - Replication
    - Add more units
  - Pipelining
    - Use the existing units more efficiently
    - Increase clock frequency
- Both require parallelism in the application

Replication

- Instead of having only one unit calculate each fractal position, bisect the screen and have two units calculating fractal values
- Linear Speedup does not often happen
  - Does using two units improve the performance by two for an application? *Rarely*
  - Amdahls Law

Sources of Inefficiency in Replication

- The algorithm cannot usually be perfectly partitioned
  - Leverage Symmetry
    - Partition horizontally or vertically for fractal?
  - Load imbalance
    - Do all points have the same number of iterations?
- Conflicts arise from shared resource accesses
  - Single BRAM write port in design so it must be shared
    - Solve the problem by delaying the access of one unit
      - That is, the unit is stalled until the resource become free
  - What is the duty cycle of each unit's access?
    - So not an issue for the fractals

Multi-Cycle Paths

- Why did we insert registers?
- Why not just let the datapath take N clock cycles?
- ***Just say no!!!!***
  - Risk versus Reward
  - Tool chains do not support this
    - Same arguments really as register retiming
      - Is this just inertia? I personally don't think so…

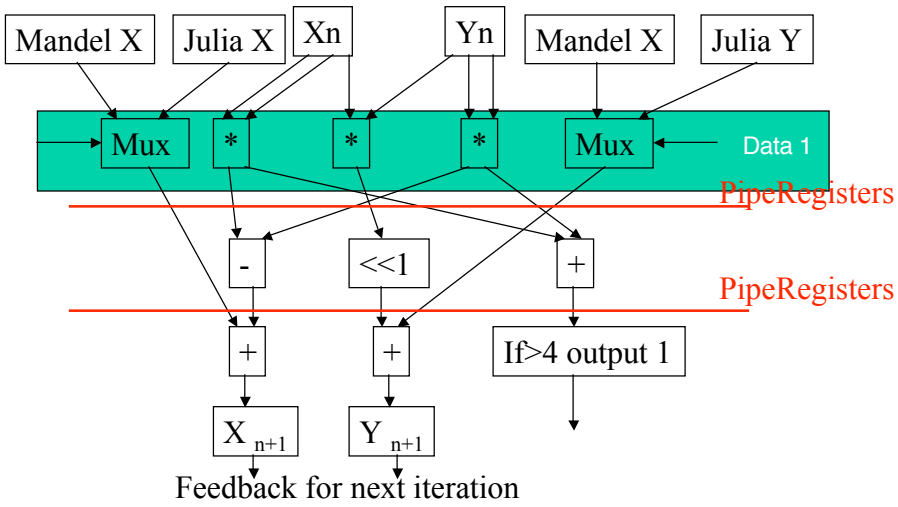
Pipelining??

- Inserting registers did not decrease the wall clock time?!?!?!
  - Simply made our design fully synchronous
  - Was not pipelining
- What is the utilization of the various stages?

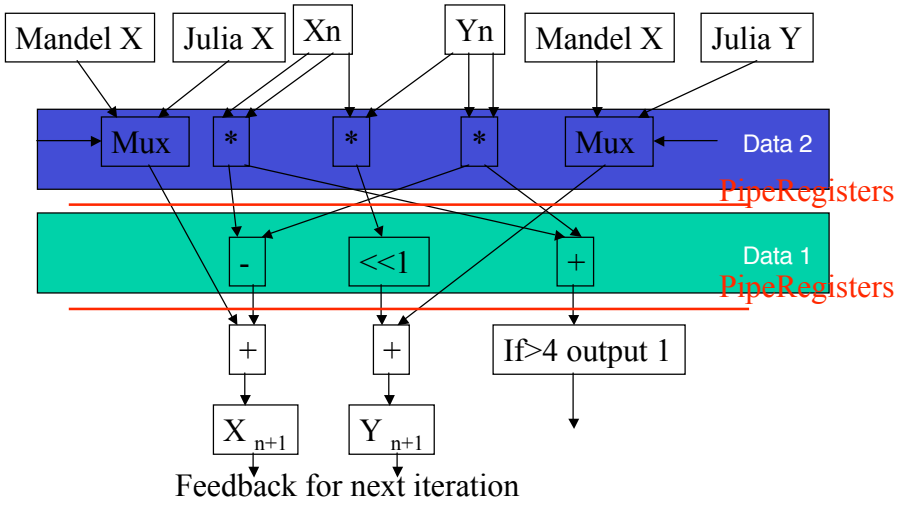
Pipelining!!

- Insert the next data item into the datapath before the previous one has finished
- Pipe registers keep the computations stage
- If you have N stages in your pipeline, think of it as N different units overlapped in space and offset in time
  - Same as replication but much cheaper
- Keeps hardware as busy as possible
- What is the effect of overlaping execution of different units if units have dependencies?
  - Pipeline data hazards

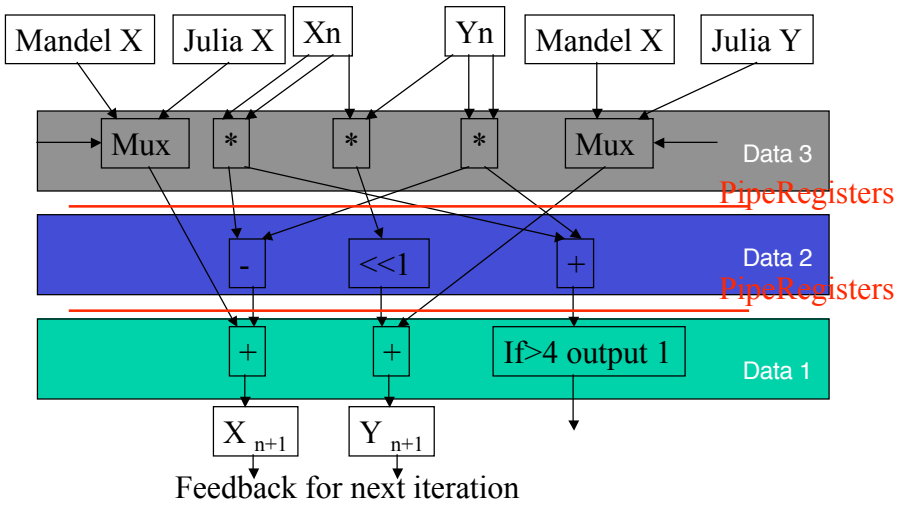
Path of Data for Fractal Datapath



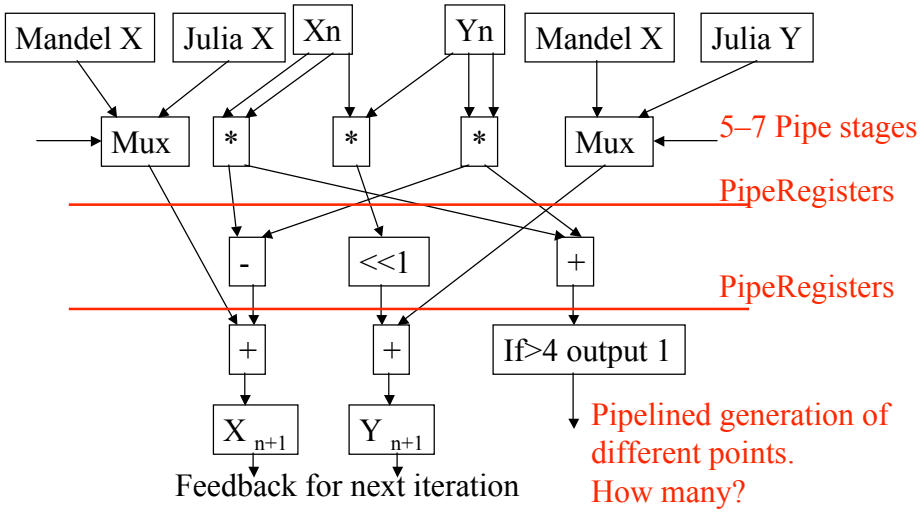
Path of Data for Fractal Datapath



Path of Data for Fractal Datapath



Fractal Generation Datapath



Pipeline Performance Losses

- The datapath rarely can be evenly split
  - Depends on the longest path in any stage
- PipeRegisters add fixed overhead (setup and hold)
- Data "Hazards" can cause stalls/bubbles in the pipeline

What performance is required?

- Replication and Pipelining are not trivial to implement—make sure you need them
- Is either needed for Lab #2?
- How would you tell?
  - Hint: each Julia image takes (64*64*64*7*1/50e6) = 0.036s to create.
  - Is this "real-time" enough for an animation?
- Regardless very important idea
  - Extra points for reuse of a single pipelined multiplier
  - More points for multiple coordinates using three or more pipelined multipliers

How to Pipeline for Lab #2?

- Create the datapath in a single module
  - Use the coregen multiplier
  - ***Register all inputs and outputs***
  - Implement and then run the static timing tool
  - Add a pipe stage
  - Iterate
- Note that the coregen multiplier has several pipeline options

Static Timing Tool

- Must fully implement design first
  - Needs placement and routing info
- Tools → Simulation/Verification → Interactive Timing Analyzer
- Analyze → Against Auto Generated Design Constraints
- Most Critical Paths will be shown
  - You can crossprobe into the Floorplanner
- You should report the critical path for all your desings
  - You should try and optimize that path if you have time