

California Institute of Technology
Department of Computer Science
Computer Architecture

CS184a, Winter 2005

Assignment 7: Retiming

Monday, February 14

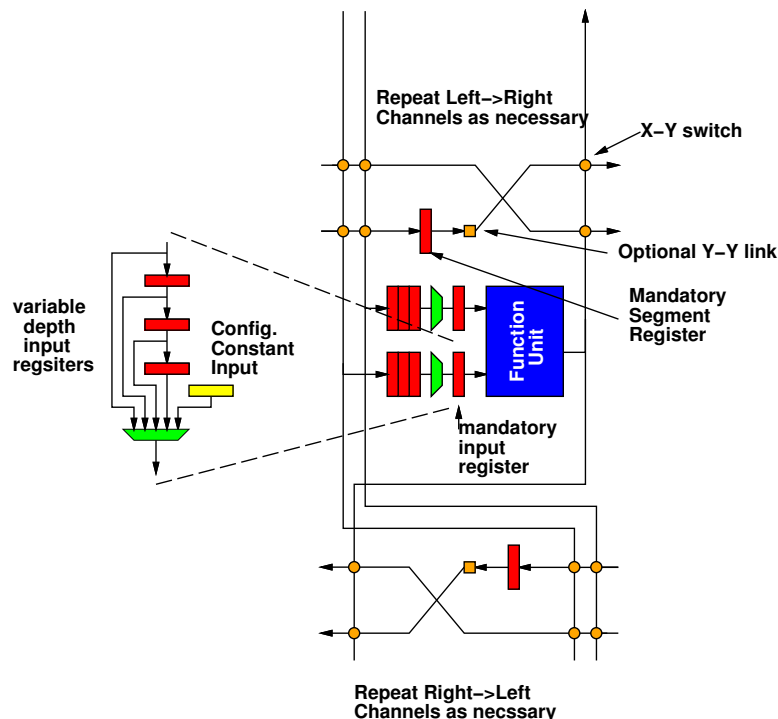
Due: Friday, March 4, 9:00AM

Shown below is a single slice of a one-dimensional datapath architecture. A 7 slice instance of the architecture is shown on the last page. Datapaths are multiple bits wide. The functional units are ALUs which include a multiplier. This datapath has been designed to run at 1GHz. To meet this 1ns cycle time:

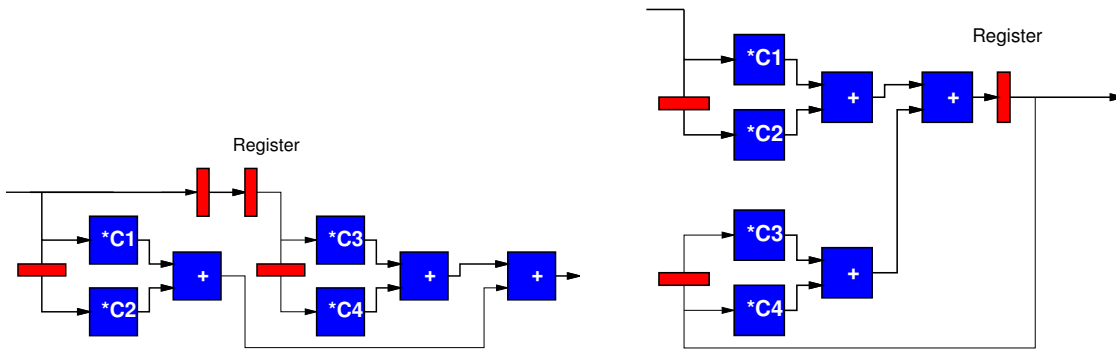
- Every functional unit has a mandatory register on its input (shown as a red rectangle) which is preceded by a variable delay input register.
- The network consists of length-2 lines and there is a **mandatory** register (shown as a red rectangle) at the programmable switch between segments (the funniness with crossover at the bottom maintains the invariant that outputs are driven after the segment register and inputs are consumed before the segment register so that we maintain the length-2 property suggested above).

The orange circles in the interconnect denote X-Y switches. The orange squares denote Y-Y switches.

Your design must respect the mandatory registers in the architecture. Your freedom includes placing the datapath and programming the variable delay input registers.



Consider the two computational graphs shown here:



For each graph:

1. Place the graph onto the 7 cell instance of the architecture. Show the routing. What is the minimum left→right and right→left channel width?
2. Pipeline and retime the placed graph so that it produces a new result on every cycle. Make the design C-slow if necessary. Attempt to minimize C. Report the C.
3. Identify the programming of each cell:
 - Compute function
 - Retiming depth of each input (or configured constant)
 - Input sources
 - Output destination
 - Y-Y link programming
4. Repeat for a time-multiplexed, 4-cell design. Each cell has two instructions which execute in round-robin fashion. Target producing a single result every second cycle.
 - (a) In the spatial design, you could use the variable delay register as a fixed-depth queue which advanced on every cycle and where you always accessed the head. Can you keep this restriction for the time-multiplexed design? If not, draw the modified datapath and identify the new **pinst**. Try to minimize the number and scope of changes necessary.
 - (b) give placement of functions
 - (c) show routing (may share wires in time)
 - (d) give minimum channel widths
 - (e) give necessary maximum depths for input retiming registers
 - (f) identify minimum C for each design
 - (g) give programming for each cell (all timesteps)
5. Repeat for a time-multiplexed, 1-cell design. Aside from the datapath modification above, the output of the cell can feedback to one of the cell inputs such that the datapath can use its own output on the following cycle. The cell has seven instructions which execute in round-robin fashion. Target producing a single result every seventh cycle. Give same information as above.

I expect you might approach the retiming:

- Generate a modified graph which adds delay blocks to model the mandatory retiming in the placed design.
- Determine what it will take to retime the resulting graph so that it is fully pipelined (pipeline, C-slow?). I expect you to be able to perform the retiming algorithm and identify the minimum C-slow C necessary to achieve a cycle delay of one.
- Slide registers to match the existing, mandatory registers and place the balance on the inputs of functional units.
- Read off/summarize the registers per input from your retimed design.

