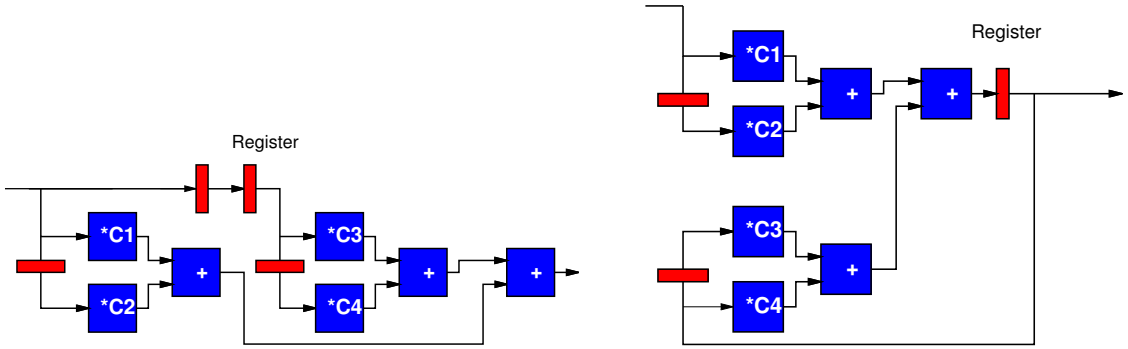
University of Pennsylvania
Department of Electrical and System Engineering
Computer Organization

ESE680-002, Spring 2007 — Assignment 7: Retiming — Wednesday, March 21

Due: Wednesday, March 28, 12:00PM

Consider the two computational graphs shown here:



For each graph:

1. Place the graph onto a 7 cell instance of the architecture shown on the next page. 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 (input delay) of each input or configured constant input value
   - Input sources
   - Output destination
   - Y-Y link programming

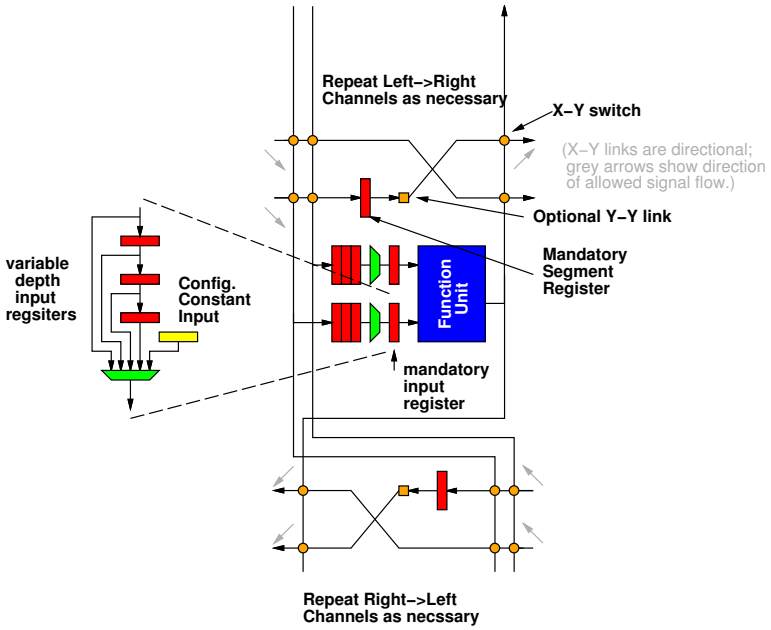
I expect you might approach the retiming:

- 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.

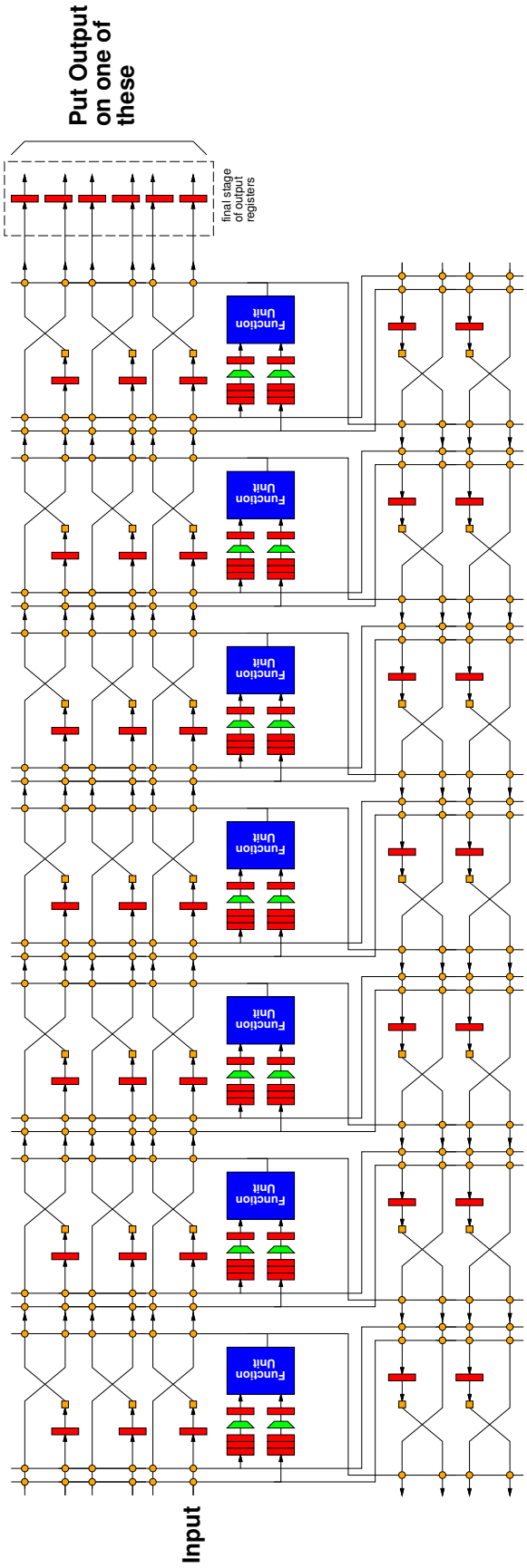
Shown below is a single slice of a one-dimensional datapath architecture. A 7 slice instance of the architecture is shown on the following page. Datapaths are multiple bits wide. The functional units are ALUs which include a multiplier. This datapath has been designed to run at 1GHz. To meet this 1ns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

Your design must respect the mandatory registers in the architecture. That is, you must use them, and your retiming must preserve the correct operation of the original graph—the mapped design should produce the same sequence of outputs, perhaps delayed by a number of pipeline cycles, as the original in response to a sequence of inputs. Your freedom includes placing the datapath and programming the variable delay input registers.



- The orange circles in the interconnect denote X-Y switches. The X-Y switches are directional (as shown). They cannot be used to switch between the horizontal route channels.
- The orange squares denote Y-Y switches.
- Each "functional unit" can be configured to execute a single function.


Put Output on one of these
final stage of output registers
Function Unit
Function Unit
Function Unit
Function Unit
Function Unit
Function Unit
Function Unit
Input