California Institute of Technology
Department of Computer Science
Computer Architecture

CS184a, Fall 2000 Assignment 5: Programmable Compute Tuesday, October 24

Due: Monday, October 30, 10:30AM

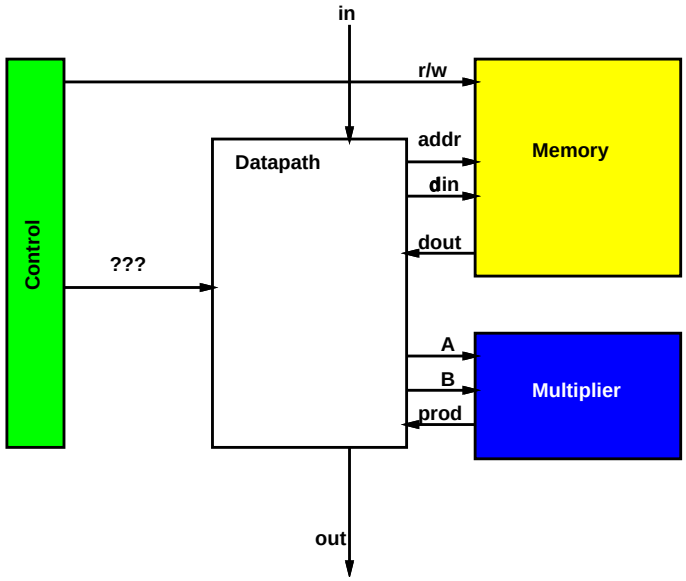
None of this should be tricky. The exercise is intended to give you a feel for different programmable compute media. I believe each of the components should be pretty easy and straightforward to implement.

Functions: Consider implementing a 5-TAP FIR using a single multiplier and a single memory.

```
D[next_data_pos]=in;
tmp_out=0;
tmp_data_ptr=next+data_ptr;
for i = 0 to 4
  tmp_out+=C[i]*D[tmp_data_ptr];
  tmp_data_ptr--;
out=tmp_out;
next_data_pos++;
// for simplicity we ignore the
//  details of buffer wrap here
// also assume all data values are
//  the same width
//  (incl. mpy results and accumulated output)
```

Let's break this into 4 pieces:

1. multiplier
2. non-multiplier datapath (accumulator and address registers, maybe muxes—I expect this to be pretty small and simple)
3. control (sequencing operations that control datapath, muxing, memory r/w...I'm assuming you'll integrate the `for` loop control here (hence no feedback from datapath to control).)
4. memory (assume synchronous behavior for memory)



Programmable Compute Blocks: Consider the following programmable media:

1. array of 4-LUTs
   - with optional FF output as shown in class
   - with bit-level interconnection network
2. array of 4b ALUs
   - all 181 functions + arithmetic shift left and right
   - with optional registered output
   - carry can be set to zero or one or it can be wired to the adjacent ALU
   - interconnect paths are nibble-wide
3. PLA
   - all outputs registered

Implement: Show how to implement the multiplier, datapath, and control in each of these media.

1. Multiplier – showing a 6×6→12 multiplier implementation is sufficient.
2. Datapath – you do **not** have to implement the datapath in the PLA. You may assume 16b datapaths for this portion.
3. Control – you may choose to use the datapath from either your 4-LUT or 4b-ALU case as the reference unit for your PLA control implementation.

Notes:

- You need to show decomposition of larger operations into the basic components – you may use hierarchy so you can reuse common subpieces.
- You may write a logic equation of (up to) 4 inputs to represent the logic in a 4-LUT.
- You may write a suitable logic equation from the ALU's function set to represent ALU functionality.

Analysis:

- Logic Resource Utilization – how many basic building blocks does each design use? (4-LUTs?, 4b-ALUs?, in the PLA case give ins, outs, and product terms)
- Programmable Delay – for the 4-LUT and 4b-ALU case, how many hops through programmable interconnect in the critical path for each design?
- How many instruction bits? (for the 4-LUT and 4b-ALU case, separate out interconnect and non-interconnect; for this assignment assume that you need $\lceil \log_2(N) \rceil$ bits per logical input to specify interconnect routing (where N is the total number of logical resources used).)

Summarize your analysis results in a table.