California Institute of Technology
Department of Computer Science
Computer Architecture

CS184a, Winter 2005 — Assignment 3: Instructions — Wednesday, January 19

Due: Monday, January 24, 9:00AM

Everyone should do problems 1–4. You should use a drawing program for datapaths (where appropriate).

1. Consider a simple, sequential, non-branching, programmable datapath with a single one-bit output computational unit. For this problem consider two possible functional units: a 3-input NAND and a 3-input LUT. Now, let us consider implementing a 5-input parity function (XOR5) on each of these programmable datapaths.

   - Draw your datapth for each case.
   - Define the primitive instruction (pinst) for each of the units (what bits are included, what do they do, how many of each).
   - How many instruction bits are required to specify the computation for each instruction in the two cases?
   - How many instruction cycles will it take to implement the parity function in each of these cases? (show μcode to support)
   - How many total operation instruction bits in memory are required to describe this operation in each case?

2. Consider the branching datapath from the previous assignment. Concretely, consider the datapath width to be 16.

   - Identify a useful function (other than multiply or divide) which can be performed from at most two 16b inputs which would take over 24 cycles to compute on this datapath.
   - How many cycles does this operation require? Provide assembly instructions to support.
   - Sketch a revised datapath and instruction encoding that can support a new instruction which allows this operation to be completed in one or a small number of cycles (small can be absolute [say less than 5] or relative [say a factor of 4 less than the original]). The new instruction should not have a longer delay than the existing adder.
   - If the operation still requires multiple instructions, provide the new instruction sequence to compute it.
   - How many total instruction bits have you saved with this addition?

3. Consider the branching datapath from the previous assignment and your multiply routine. Reimplement your multiply for each of the 2 additional cases below. Assume we have 16 addressable registers in each case.

   (a) base case is the existing datapath. *i.e.* 2 source registers and one destination register in each instruction: rdst = rsrc1 op rsrc2 [so, no new coding here, just count and summarize the results from last assignment]

   (b) 2 register instruction: rdst = rsrc op rdst (allow overwrite in single instruction with operation)

   (c) 1 operand/instruction: accum = rsrc op accum (also operations rsrc = accum, accum=0, accum=rsrc; state additional datapath assumptions as necessary)

   Complete the following table based on your results.

Architecture	Total Cycles for 8b mpy	Total bits for 8b mpy	Instr. bits/cycle
3 register			
2 register			
1 register			

4. Let's say you have an old design which is 70% instruction memory, and you've picked an optimized datapath and instruction encoding scheme to reduce the instruction memory size by 35% while keeping other things the same. Assume, for simplicity, technology is continuously improving such that you get a reduction in feature size by a factor of 2 every three years. How many months of technology scaling give the same size reduction as your improved design?

extra Describe other techniques which can be used to decrease issued instruction width and/or total instruction bits to describe a computation.