

CSE 240A - Principles of Computer Architecture

Homework 1

Q1. Suppose Intel has decided that they would like to port an existing quad-core 45 nm design down to 32 nm. In this question, we examine the implications of this. Let's start by assuming that Intel has decided to do an exact shrink of the design, i.e. that they are going to use a straight optical shrink to shrink all three dimensions of the design in exactly the same proportions. Assume that we are in a leakage-limited scaling scenario, and that the voltage is not being scaled. There are no other changes to the design.

To see the solutions turn the page

1. Given a single transistor:

1.1. how has its capacitance changed? $\rightarrow 1/5$

1.2. how has its speed changed? $\rightarrow 5$

1.3. how has the energy dissipated by switching the transistor changed? $\rightarrow 1/5$

2. Given a single segment of wire:

2.1. how much does the resistance change? $\rightarrow 5$

2.2. how much does the capacitance change? $\rightarrow 1/5$

2.3. how does the RC delay of the wire change? $\rightarrow 1$

2.4. how does the energy dissipated by switching the wire change? $\rightarrow 1/5$

3. Will we be able to clock the design faster, and if so, by how much? $\rightarrow 5$

4. Assuming the same clock frequency, how much does the energy dissipation change? $\rightarrow 1/5$

5. How will the area of the design change? $\rightarrow 1/5^2 \approx 0.04$

from this point on until the last page for the solutions

6. Assuming that Intel uses the remaining area to add new cores:

6.1. how many new cores will there be?

6.2. assuming they run it at the maximum frequency, how much power will it consume?

7. Suppose that in the Intel multicore designs, the design employs a wide, unpipelined bus (i.e. 512 wires running across the length of the chip) to which all of the cores are connected. When additional cores are added, this bus is extended to connect to the new cores. Assuming that all components are optically shrunk as in the previous examples. Consider only RC delay and not the delay of the driving transistors.

7.1. How are the energy and delay of signals on that bus affected? Consider only RC delay and not the delay of the driving transistors.

we could decrease the energy and we're doing a bus architecture. we route outputs to reduce the delay of the bus

7.2. What are the ramifications of this, and what changes can be implemented to address this?

Q2

A. What is the high-level trade-off the "Efficient Embedded Computing" paper is trying to find the balance between? (i.e. what is its claim to being better than both an embedded RISC processor and an ASIC?)
 B. One of the key decisions the ELM folks made is to replace the fastest instruction cache with an instruction register file. Explain what role each of the following has in motivating this decision:

1. Amdahl's Law
2. Dennard Scaling

Q3 Your engineering team tells you that there are four possible enhancements they can make to your CPU. The speedups for each of these enhancements are:

speedup1 = 10
 speedup2 = 15
 speedup3 = 10
 speedup4 = 15

However, they also tell you it is only feasible to use enhancements 1 and 2 together or enhancements 3 and 4 together (no other combinations possible).
 Enhancements 1 and 2 are used 20% and 30% of the time respectively, and only one of these enhancements is in use at any given time.
 Enhancements 3 and 4 may be used 25% and 30% of the time respectively, but 15% of the time they overlap and the speedup of the combined enhancements is only 10.
 Both sets of enhancements would require the same amount of time and cost to implement.

A. What is the overall speedup with each set of enhancements?
 B. You later realize that the speedup during the overlap of enhancements 3 and 4 is actually 15, NOT the 10 that you were told. Does this change the recommendation for which speedup to use?

Q4 Reg-Mem ISA Pipeline

A Reg-Mem ISA allows one of the operands of ALU instructions to come from the data memory. A simple alu operation can have reg-constant, reg-reg and reg-mem addressing modes.
 Eg: sub R2, R3, R4 OR sub R2, R3, 4 OR sub R2, R3, (R4) OR sub R2, R3, (4)
 (Recall that MIPS is a reg-reg ISA)

One way to implement this ISA is to re-order the MIPS pipeline to IF-ID-MEM-EXE-WB, with the hardware resources are identical to the MIPS pipeline. Compared with the MIPS pipeline, both the change of ISA (from Reg-Reg to Reg-Mem) and the change of addressing modes will have an influence on the instruction count.

A. Please compare the Reg-Mem pipeline with Reg-Reg pipeline (modifying the addressing modes and varying the number of instructions) for instruction sequences that are a combination of loads, stores, and/or arithmetic operations. Try different combinations and addressing modes to show the influence on instruction count.

B. From (4A), we see that there are instances where Reg-Mem ISA increases the number of instructions compared to MIPS. To minimize the performance penalty caused by data hazards, we need to introduce forwarding paths in the pipeline. For the Reg-Mem ISA, please identify the forwarding paths and also provide a pair of dependent instructions that would benefit.

C. Assuming the required forwarding paths have been added to the pipeline, draw the pipeline diagram for the following code fragment. Use the style shown in Fig C.37 in P&H:

```
sub r2, r3, (r4)
and r1, r2, r6
ld r2, 0(r4)
add r1, r2, (r3)
sub r4, r5, (r1)
st r4, 0(r10)
```

It is obvious that although instructions between registers that routing values that might cause the memory or are not included to access the memory will remain identical between the two ISAs, however...

MIPS

Reordered ISA

$ld\ r2, 0(r4) \Rightarrow$
 $add\ r5, r2, r6$
 $add\ r3, r3, (r4)$

$ld\ r2, 100(r4) \Rightarrow$
 $add\ r6, r5, 100$
 $ld\ r2, (r6)$