Unit 18
Improving Performance
Caching and Pipelining

Review of Software Execution (1)

- Recall:
  1. High level code is compiled to __________ instructions
  2. When executed, CPU uses PC register to know what instruction to __________ next
  3. PC is incremented and used to fetch the next instruction
  4. Registers are used as temp. storage of variables
  5. Variables "live" in _________ & must be updated

Note this init() function initializes a million element array with values 0 to 999999
Next slide shows the execution of the 'init' for loop at several points in time (with some details omitted).

Review of Software Execution (2)

- Time 1: Store
- Time 2: Increment i
- Time 3: Loop & repeat
- Time 4

Some details have been left out (how does sf get initialized, incremented, etc.)

Improving Performance

- We want to improve the performance of our computation
- Question: What are we referring to when we say "performance"?
  - __________
  - __________
  - __________
- We will primarily consider _________ in this discussion
How Do We Measure Speed

- **Fundamental Measurement:** __________
  - Absolute time from __________ to __________
  - To compare two alternative systems (HW + SW) and their performance, start a timer when you begin a task and stop it when the task ends
  - Do this for both systems and compare the resulting times
- We call this the __________ of the system and it works great from the perspective of the __________ task
  - If system A completes the task in 2 seconds and system B requires 3 seconds, then system A is clearly superior
- But when we dig deeper and realize that the single, overall task is likely made of __________ small tasks, we can consider more than just latency

Performance Depends on View Point?!

- What’s faster to get from point A to point B?
  - A 747 Jumbo Airliner
  - An F-22 fighter jet
- If only __________ to get from point A to point B, then the __________
  - This is known as __________ [units of seconds]
  - Time from the start of an operation until it completes
- If __________ to get from point A to point B, the ___ looks much better
  - This is known as __________ [jobs/second]
- The overall execution time (latency) may best be improved by __________ throughput and not the latency of individual tasks

Hardware Techniques

- We can add hardware or reorganize our hardware to improve throughput and latency of individual tasks in an effort to reduce the total latency (time) to finish the overall task
- We will look at two examples:
  - Caching: Improves __________
  - Pipelining: Improves __________
Caching

- **Cache (def.)** – "to store away in hiding or for future use"
- **Primary idea**
  - The _________ you access or use something you expend the _______ amount of time to get it
  - However, store it someplace (i.e. in a cache) you can get it more ________ the next time you need it
  - The next time you need something check if it is in the cache first
    - If it is in the cache, you can get it quickly; else go get it expending the full amount of time (but then _________ it in the cache)
- **Examples:**
  - __________________
  - __________________
  - __________________

Cache Overview

- Remember what register are used for?
  - Quick access to copies of data
  - Only a _______ (32 or 64) so that we can access really quickly
    - Controlled by the _________
- Cache memory is a small-ish, (____bytes to a few _____bytes) "_______" memory usually built onto the processor chip
- Will hold ______________ of the latest data & instructions accessed by the processor
- Managed by the ______
  - ____________ to the software

Cache Operation (1)

- When processor wants data or instructions it always ________ in the cache first
- If it is there, ______ access
- If not, get it from __________
- Memory will also supply __________ data since it is likely to be needed soon
  - Why?
  - Things like ______ & ______ (instructions) are commonly accessed sequentially

Cache Operation (2)

- When processor asks for the data again or for the next data value in the array (or instruction of the code) the cache will likely have it
  - Questions?

Main point: Caching reduces the latency of memory accesses which improves overall program performance.
Memory Hierarchy & Caching

- Use several levels of faster and faster memory to hide _______ of larger levels

Main Memory ~ 100 ns
L2 Cache ~ 10 ns
L1 Cache ~ 1 ns
Registers

Faster
Less
More
Expensive
Slower
More
Expensive
Smaller

Unit of Transfer:
8-64 bytes
Unit of Transfer:
8- to 64- bits

Pipelining

- We'll now look at a hardware technique called **pipelining** to improve ________________
- The key idea is to __________ the processing of multiple "items" (either data or instructions)

Example

- Suppose you are asked to build dedicated hardware to perform some operation on all 100 elements of some arrays
- Suppose the operation \((A[i]+B[i])/4\) takes 10 ns to perform
- How long would it take to process the entire arrays: ______ ns
  - Can we improve?

```
for(i=0; i < 100; i++)
    C[i] = (A[i] + B[i]) / 4;
```

Pipelining Example

- Pipelining refers to insertion of registers to split combinational logic into smaller stages that can be overlapped in time (i.e. create an assembly line)

```
for(i=0; i < 100; i++)
    C[i] = (A[i] + B[i]) / 4;
```

Time for 0th elements to complete: __________
Time between each of the remaining 99 element completing: ______
Total: __________
Define:

\[
speedup = \frac{\text{original time}}{\text{improved time}}
\]

\[
speedup = \frac{1000\text{ns}}{10\text{ns}} = __________
\]

Clock freq: _______
### Need for Registers

- Provides separation between combinational functions
  - Without registers, fast signals could “catch-up” to data values in the next operation stage

### Pipelining Example

- By adding more pipelined stages we can improve throughput
- Have we affected the latency of processing individual elements? ___________
- Questions/Issues?
  - __________ stage delays
  - __________ of registers (Not free to split stages)
  - This limits how much we can split our logic

### Non-Pipelined Processors

- Currently we know our processors execute software 1 instruction at a time
- 3 steps/stages of work for each instruction are:
  - __________
  - __________
  - __________

### Pipelined Processors

- By breaking our processor hardware for instruction execution into stages we can overlap these stages of work
- Latency for a single instruction is the __________
- Overall throughput, and thus total latency, are greatly improved

\[
\text{for}(i=0; i < 100; i++) \\
C[i] = (A[i] + B[i]) / 4;
\]

\[
\text{Time for 0th elements to complete: } \text{000ns} \\
\text{Time between each of the remaining 99 element completing: } \text{257.5ns} \\
\text{Total: } \text{1000ns} = 4x
\]
More and More Stages

- We can break the basic stages of work into substages to get better performance
- **In doing so our clock period goes _____; frequency goes _____**
- All kinds of interesting issues come up though when we overlap instructions and our discussed in future CENG courses

Clock freq. = \( \frac{1}{10\text{ns}} = 100\text{MHz} \)

Clock freq. = \( \frac{1}{\_\_\text{ns}} = \_\_\text{MHz} \)

Summary

- By investing extra hardware we can improve the overall latency of computation
- Measures of performance:
  - **Latency** is start to finish time
  - **Throughput** is tasks completed per unit time (measure of parallelism)
- Caching reduces **latency** by holding data we will use in the future in quickly accessible memory
- Pipelining improves **throughput** by overlapping processing of multiple items (i.e. an assembly line)