

## Lecture #1

### Introduction & Administration

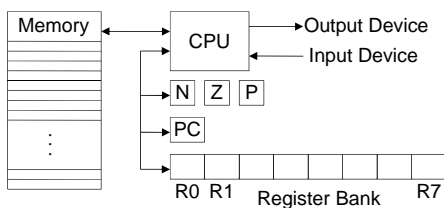
- Predictions and prejudice
- Real themes of 560
- Resources:
  - Class web page (syllabus, slides, handouts)
  - Carmen (discussion groups)
  - Classroom Q&A, office hours, team members

## Lecture #2

## Introducing the Machine

## Our Abstract Machine

- An “abstract” machine
- The world’s simplest architecture!



## Instruction Processing Cycle (IPC)

### Repetition of 6 phases:

1. Fetch
  - Read memory word indicated by PC
  - Increment PC
2. Decode
  - examine instruction to see what must be done
3. Evaluate Address
  - Calculate address of operand (e.g., for LOAD)
4. Fetch Operands
  - Obtain operands needed by instruction
5. Execute
6. Store Result

## Notes on the Instruction Processing Cycle

- Clock ticks control this cycle \*
  - Intel Core2 CPU 2.8 GHz: 2.8 billion ticks/second
  - Light bulb flicker: 60 Hz.
  - Every on/off flicker: ~50 million clock ticks!
- Not all phases needed for all instructions
  - Basic steps: fetch, decode, execute
- Some phases can be concurrent \*
  - E.g., when adding two registers, decode and fetch operands can occur at the same time!
- All modern architectures are pipelined \*
  - IPC for next instruction begun before current instruction has completed!

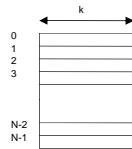
\* No need to account for this in your simulator (lab 1)

## Incrementing the PC

- Note that increment is in the Fetch phase
- So, if a subsequent phase uses the PC, its value is the address of the *next* instruction
- Useful for “branch to subroutine” instructions:
  - Execute: store PC, then modify it
  - Next Fetch is for instruction at new address
  - Stored value is address of instruction *after the branch*

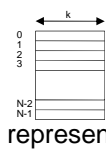
## Memory

- Organized in “cells”
  - i.e. smallest addressable unit
- N cells, addressed 0..N-1
  - each cell consists of k bits
- A “cell” is usually:
  - a byte (k = 8), “byte addressable”
  - a word (k = 16), “word addressable”
  - (but more exotic architectures exist too)

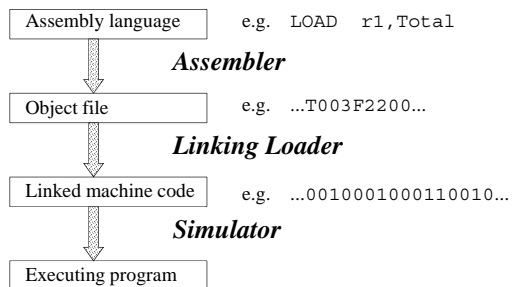


## Questions

- How many different values can be represented in such a cell?
- How many bits are required to represent an address?



## Overview of Labs



## Lab 1 Requirements

- Given a text file (the object file)
  - initial “header record”, followed by a sequence of “text records”, finished with an “end record”
- Develop a simulator that allows the user to:
  - initialize the machine
  - load the program from the object file into the machine’s memory
  - simulate execution (3 modes: trace, quiet, step)
  - optional: other user actions (display memory, modify registers)
- For trace mode:
  - 1) memory configuration & registers after loading (the relevant page)
  - 2) trace of each instruction executed (i.e. memory & registers affected)
  - 3) memory after termination (the relevant page)
- Robustness (so test it *thoroughly*)

## Lab 1 Milestones

- Preliminary Documentation (in class): **Oct 2**
  - Programmer's guide particularly important, but we will look at (and comment on) everything you turn in.
- Mandatory Design Review: **Oct 5/6**
  - sign up for a 25-minute slot
  - everyone in group must be present
- Completed Documentation (in class): **Oct 14**
- Interactive Grading: **Oct 15/16**
  - sign up for a 48 minute slot
  - everyone in group must be present

## Advice for Groups

- Meetings
  - Set up *frequent* and *convenient* times and places
    - IM is useful, but can not replace face-to-face time
  - Keep *minutes* in real-time (designate someone)
  - Have an *agenda* (set it beforehand)
    1. always start by reviewing and *approving previous minutes*
    2. continue with *progress report* on open action items
    3. continue by discussing *new issues, upcoming concerns, etc*
    4. always conclude with *action items*
      - assign responsibility for each action item!
- Rotate responsibilities through quarter
  - Lead for various documents
  - Coding / testing
  - Designated group "manager"?