Software Construction
Example CPU Simulator

Dhammika Elkaduwe

Department of Computer Engineering
Faculty of Engineering

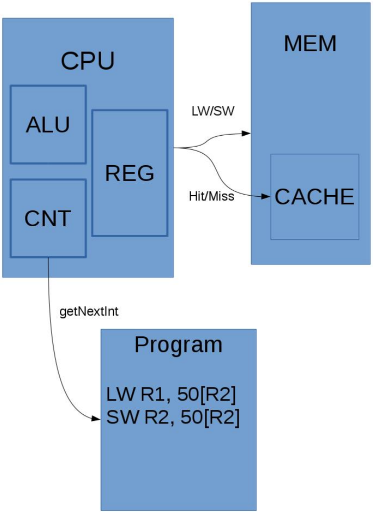
University of Peradeniya

ILOs

How to use the following concepts in a program;

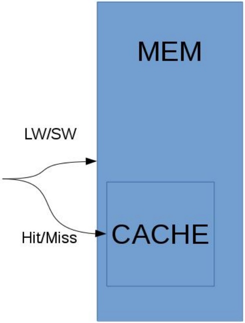
- interfaces
- classes
- abstractions
- testing

Specification



- Three main modules: memory, CPU and program
- Memory has a cache (which can be enabled)
- For simplicity:
  - Instructions are basically strings with *op code* and registers
  - Instructions are not stored in memory
  - memory only support *LW* and *SW* (no *LH, LB* etc)
  - Only few instructions (LW, SW, ADD, SUB, MUL, LI)
  - Only a data memory

Specification: Memory module



- Memory module only store data
- The size is determined at the time creating a machine (example: put things together like setting up a computer)
- Initially all memory locations are stored with zeros
- Trying to access memory address zero should throw an exception

How to start?

begin with defining interfaces!

```java
import java.io.IOException;// IOExceptins

public interface MemInterface {
  int lw(String address) throws IOException;
  void sw(String address, int val) throws IOException;
  int cacheHits();
  int cacheMisses();
};
```

How to start?

begin with defining interfaces!

```
import java.io.IOException;// IOExceptins

public interface MemInterface {
  int lw(String address) throws IOException;
  void sw(String address, int val) throws IOException;
  int cacheHits();
  int cacheMisses();
};
```

Implementation memory module

see RAM.java

```java
public class RAM implements MemInterface {

  int cacheHits, cacheMiss;
  int sizeOfRAM;

  Map<String, Integer> memory;

  public RAM(int sizeInMB) {
    cacheHits = cacheMiss = 0;
    sizeOfRAM = sizeInMB * 1024 * 1024;//size in bytes
    memory = new HashMap<String, Integer>();
}
```

Implementation memory module ...

see RAM.java

```java
public int cacheHits() { return this.cacheHits; }
public int cacheMisses() { return this.cacheMiss; }

private boolean withinMemory(String address) throws
    IOException {
  int addr;
  try {
    addr = Integer.parseInt(address);
  } catch(NumberFormatException e) {
    throw new IOException();
  }

  return (addr > 0) && (addr < this.sizeOfRAM);
}
```

Implementation memory module ...

see RAM.java

```java
public int lw(String address) throws IOException {
  if(!withinMemory(address)) throw new IOException();

  Integer val = memory.get(address);

  if(val == null) return 0;
  else return (int) val;
}

public void sw(String address, int val) throws IOException {
  if(!withinMemory(address)) throw new IOException();

  memory.put(address, new Integer(val));
}
```