School of Computer Science

COMP SCI 2000 Computer Systems Lecture 2

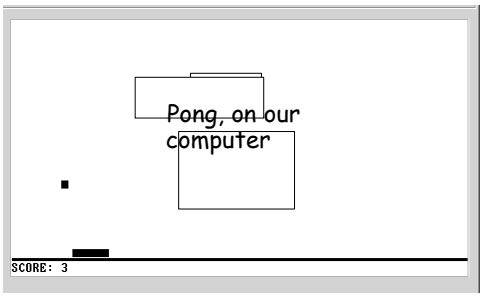
adelaide.edu.au

seek LIGHT

Applications

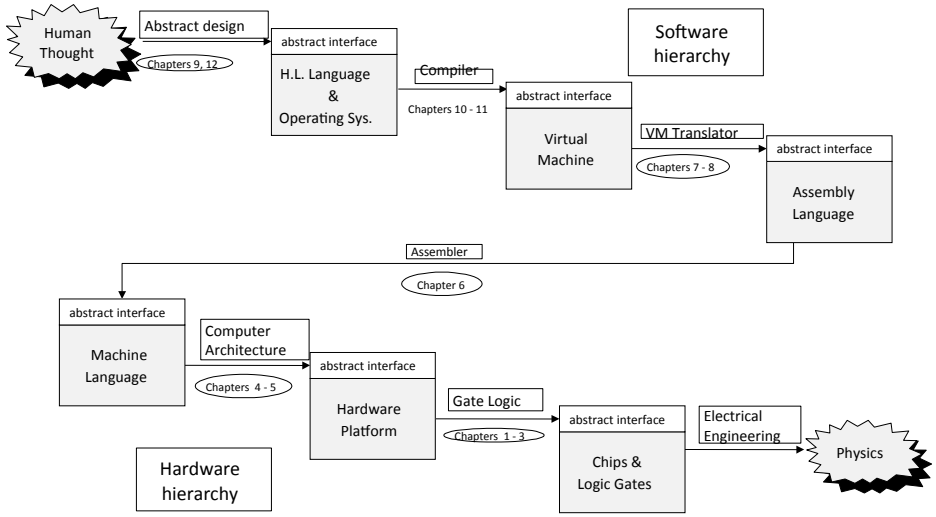
Pong, 1985

Pong, 2011



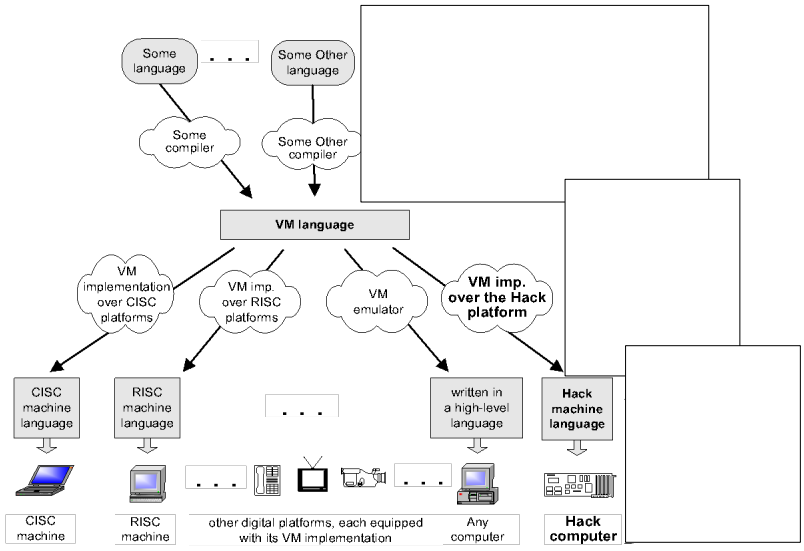
Pong, on our computer

The whole system

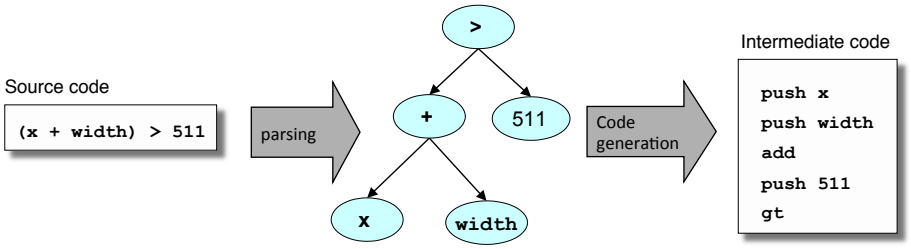


(Abstraction–implementation paradigm)

A modern compilation model



Compilation

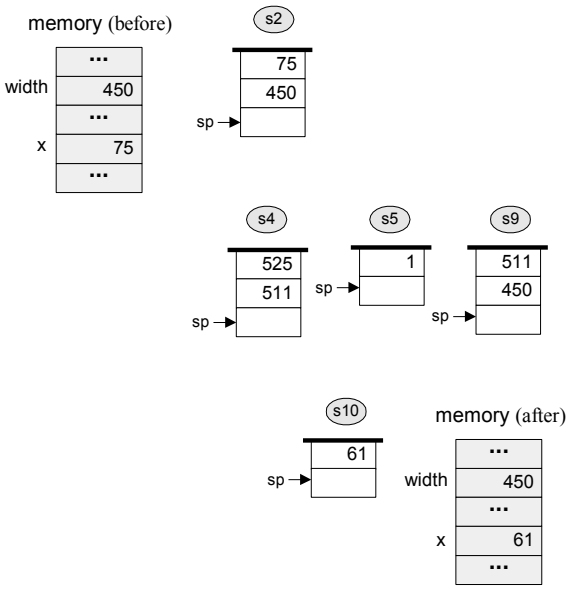


Abstraction Syntax Analysis Parse Tree Semantic Synthesis

Inside a virtual machine

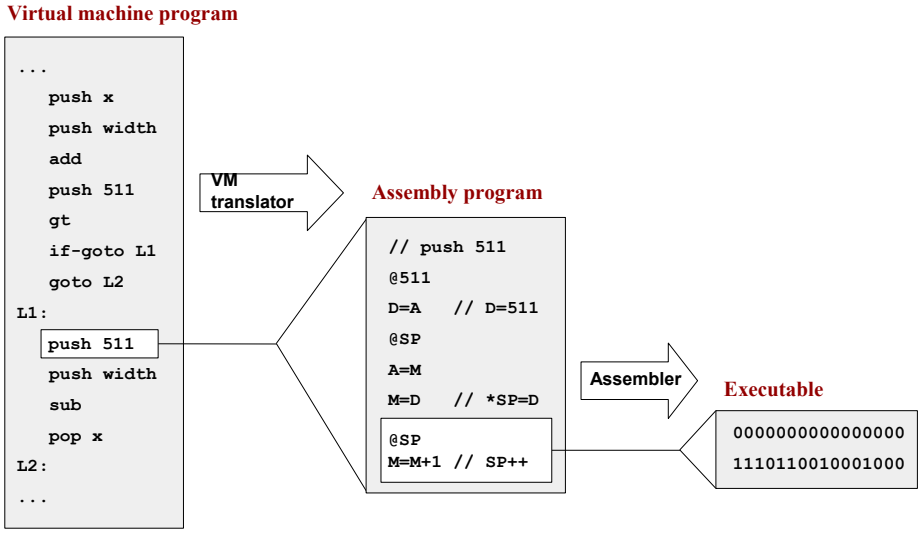
```
if ((x+width)>511) {
   let x=511-width;
}
```

```
// VM implementation
   push x        // s1
   push width    // s2
   add           // s3
   push 511      // s4
   gt            // s5
   if-goto L1    // s6
   goto L2       // s7
L1:
   push 511      // s8
   push width    // s9
   sub           // s10
   pop x         // s11
L2:
...
```

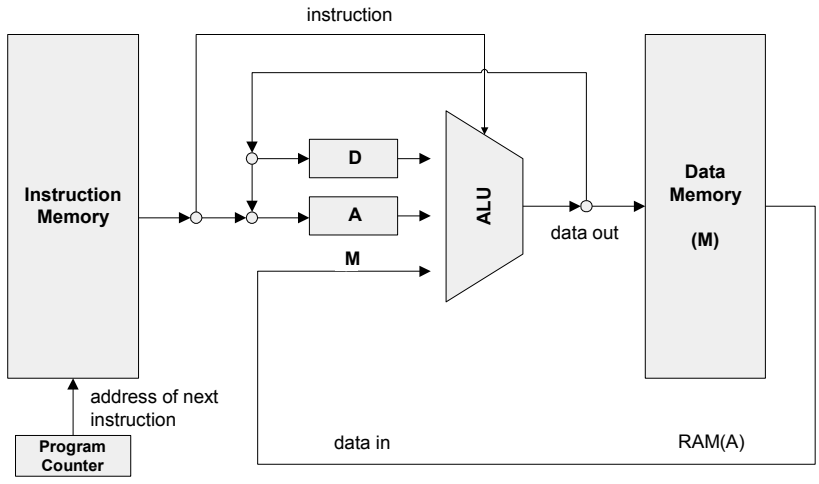


The low-level programming path

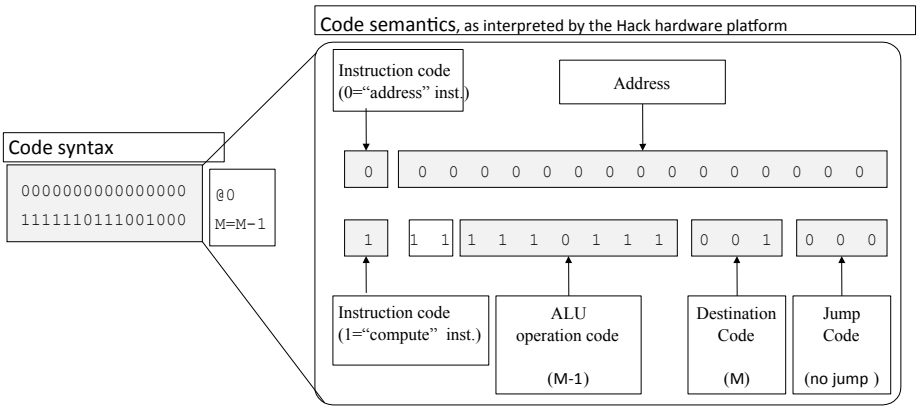
For now, ignore all details!



What do the instructions do?



The code directs elements of the processor in order to achieve results



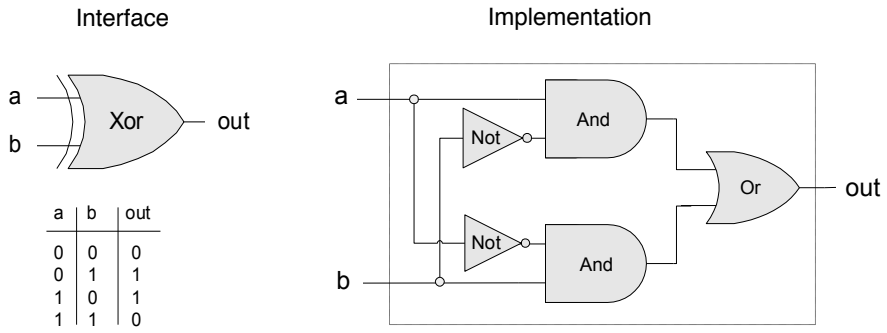
Logic design

- Three types of logic we will be using:
  - Combinational logic – used for the ALU
  - Sequential logic – used for RAM
  - Gate logic – putting it all together to get a computer

What is gate logic?

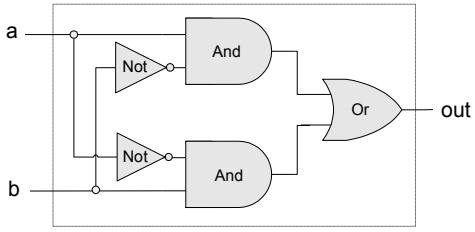
- Our hardware is an inter-connected set of chips.
- Chips are built of simpler chips, down to the simplest structure of all – the elementary logic gate.
- Logic gates are hardware implementations of Boolean functions. This allows us to represent logical statements in computer form.
- Every chip and gate has:
  - An interface: Telling us what it does
  - An implementation: Telling us how it does it.

Example



Building gates

- We won't be building real gates, we'll build them in simulation using a Hardware Description Language (HDL)




```
CHIP Xor {
   IN a,b;
   OUT out;
   PARTS:
   Not(in=a,out=Nota);
   Not(in=b,out=Notb);
   And(a=a,b=Notb,out=w1);
   And(a=Nota,b=b,out=w2);
   Or(a=w1,b=w2,out=out);
}
```


Summary

- We're going to show you how software and hardware work together to build a computer system.
- Over the course, you will build parts of that system and get practice in combinational, sequential and gate logic, as well as learning how high level languages make things happen in real systems.
- You will be building gates in your first assignment so start getting familiar with the HDL.
- Don't forget to finish your Chapter 1 reading.

Next week

- Monday is a public holiday. There is no lecture.
- There is no tutorial next week.
- You should read “Chapter 2” from the forums and keep working on your Assignment 1.
- Any questions? Ask on the forum or right now!