We acknowledge and pay our respects to the Kaurna people,
the traditional custodians whose ancestral lands we gather on.

We acknowledge the deep feelings of attachment and relationship of the Kaurna people to country and we respect and value their past, present and ongoing connection to the land and cultural beliefs.

THE UNIVERSITY of ADELAIDE

Computer Systems

Lecture 01: Course Admin & Introduction

What is this course?

This course is about the fundamentals principles of how computers work, and how hardware and software components interact to create a functioning system.

- This course is based on the Nand2Tetris curriculum, and the book "Elements of Computing Systems" by Nisan & Schocken
- You will build the core components of a complete computer system, layer by layer, starting with basic logic gates and ending with a compiler.
- Although this course uses the Nand2Tetris curriculum, many of our assessments and areas of focus differ from that course.

THE UNIVERSITY of ADELAIDE

Learning Outcomes

- Understand how the different layers and parts of modern computer systems interact.
- Design the core hardware of a computer from basic components.
- Understand how computers represent programs and data.
- Understand how a computer executes a program.
- Write assembler and machine code
- Understand the translation process from higher level representations into machine language.
- Understand the basics of how computers interact with I/O devices.

THE UNIVERSITY of ADELAIDE

Should I be in this course?

- ✓ I'm a 2nd-year Computer Science or Engineering Student and have successfully completed
  - COMP SCI 1102 Object Oriented Programming
- ✓ I'm a 2nd-year IT Student and have successfully completed both of
  - COMP SCI 1015 Intro to Applied Programming
  - COMP SCI 1013 Intro to Computer Systems, Networks & Security
- ✓ I'm a Postgraduate student
  - Stuying an approved program
    - GCertCompSc, GDipCompSc, MComp&Innov, GCertCybSec , GDipCybSec, MCybSec
  - And having successfully completed a foundations course
    - COMP SCI 7202, COMP SCI 7208, COMP SCI 7211, COMP SCI 7103

Yes; Welcome!

THE UNIVERSITY of ADELAIDE

Should I be in this course?

- ✗ I'm a first-year student
- ✗ I haven’t passed the pre-requisite courses
  - COMP SCI 1102 OOP
    or both of
  - COMP SCI 1015 IAP/IPIT **AND** COMP SCI 1013 ICSNS
    or any one of
  - COMP SCI 7202/7208, COMP SCI 7211, COMP SCI 7103 PG Foundations course
- ✗ I'm a Postgraduate student studying a non-approved program
  - Master of Computer Science
    or
  - Any PG program that does not have this course listed as an approved elective.

No!

This course has an exceptionally high fail rate;
You'll get more out of this course if you are properly equipped to begin with

THE UNIVERSITY of ADELAIDE

Who we are

Mr Yang Su

Main Lecturer

Mr Ian Knight

Course Coordinator & Lecturer

Contacting us

If you need to contact either of us directly, use the dedicated email address for this course:

cs2000cc@adelaide.edu.au

THE UNIVERSITY of ADELAIDE

Class Structure

Lectures

- Come prepared
- Tuesday's lecture will focus on recapping key concepts and answering questions
- Thursday's lecture will focus on exercises and examples. This will be an interactive session. We will also look at key points covered in each week's quiz during Thursday's session.

Workshops

- Start week 2
- Guided activities and time to work on assignments
- Remote students have Zoom workshops.
- **During weeks 4 and 8 there are prac exams during your workshop time.**

THE UNIVERSITY of ADELAIDE

Discussion Board

We're using **Piazza**!

- Allows the whole class to collaborate on answers
- Has categories/tags to make searching easier & notifies for similar questions.

Discussion ettiquette

- ***Before Asking***
  - Check the relevant cateogry/tag in Piazza; has this question already been asked/answered? Saves everyone time!
- ***When asking/answering questions***
  - Avoid posting large/complete sections of code (even if errors)
- ***When posting anonymously***
  - Don't use Anonymous to everyone if you want one of us to be able to check your work (use Anonymous to Classmates instead)
  - Remember to be polite/civil. We can ban users of anonymous posts.

THE UNIVERSITY of ADELAIDE

Assessment

Quiz 5%	Prac Assignments 36%	Prac Exams 14%	Exam 45% (HURDLE)

Quizzes:

- Due weekly at the start of Thursday's Lecture
- Start in Week 2.

Prac Assignments:

- First due in Week 2.
- Further details in each assignment.

Prac Exams:

- Run in Weeks 4 & 8
- Further details given in Week 2

Exam:

- 40% Hurdle;
  If you do not achieve at least 40% in the exam, your final grade will be capped at 45F.
- Supp Exam available if final grade in 45-49 range.

Full details are available on MyUni

https://myuni.adelaide.edu.au/courses/72399/pages/assessment-overview-and-grading

THE UNIVERSITY of ADELAIDE

Academic Integrity

All students are required to abide by the University's Academic Integrity Policy

In brief summary:

- Don't copy other peoples' work.
- Don't submit work that is not 100% your own.
- Don't do anything that gives you an unfair advantage over other students.

We do check your work for Academic Integrity issues and will not hesitate to raise/escallate issues we find.

This can result in failing assignments, whole courses, or even being excluded.

THE UNIVERSITY of ADELAIDE

Academic Integrity

… BUT we also have measures to help you keep your work's integrity:

- Explicit integrity rules included in the assignment descriptions to avoid confusion.
- For practical assignments, the ability to select which submission you want us to use.

More details:

- Full details for this course: https://myuni.adelaide.edu.au/courses/72399/pages/academic-integrity
- The University's course on Academic Integrity: https://myuni.adelaide.edu.au/enroll/4PKNJH
- The full policy https://www.adelaide.edu.au/policies/230/

If unsure, speak to your Lecturer

THE UNIVERSITY of ADELAIDE

Extensions

Can I get one (TL;DR)

Weekly Quizzes	✗ **NO**	Because we discuss them in lectures
Prac Assignments	✓ **YES***	Automatically applied **if conditions are met**.
Prac Exams	* **CHECK**	You will need to send us a formal request with appropriate documentation before it can be approved.
Final Exam	* **CHECK**	This assessment is run centrally. You will need to submit a formal request to the **faculty** with appropriate documentation.

We know that life happens. If you need an extension (outside of the automatic extensions in pracs)

- Contact us ASAP.
- The sooner the better.
- If in doubt, ask for one!

Full details of the extension policy are available on MyUni

https://myuni.adelaide.edu.au/courses/72399/pages/assessment-overview-and-grading

THE UNIVERSITY of ADELAIDE

PASS for Computer Systems

PASS (Peer Assisted Study Sessions) are weekly 50-minute study sessions, run by a former high-achieving student of the course, with a casual, welcoming environment.

You can ask any questions you have and clarify confusing concepts while meeting like-minded fellow students.

- Sessions are free
- No need to sign up; drop in as needed!
- Sessions run Mondays 12-1pm and Fridays 11am-12pm, starting Week 2
- Available face-to-face in Hub Central Project Room 3034
  or via Zoom (https://adelaide.zoom.us/j/87135484545 , passcode: 2000)
- There is also a PASS group in MyUni that you can join for additional resources

For more information, see: https://www.adelaide.edu.au/pass/

How to get the most out of this course

Everything makes sense when someone else explains it.

- Ability to read a solution != ability to write a solution
- This is a hands-on course; try all the things!

Be active in your learning! Come prepared for discussions and to solve problems

- This requires discipline if you're only watching the recordings

Do the practical assignments

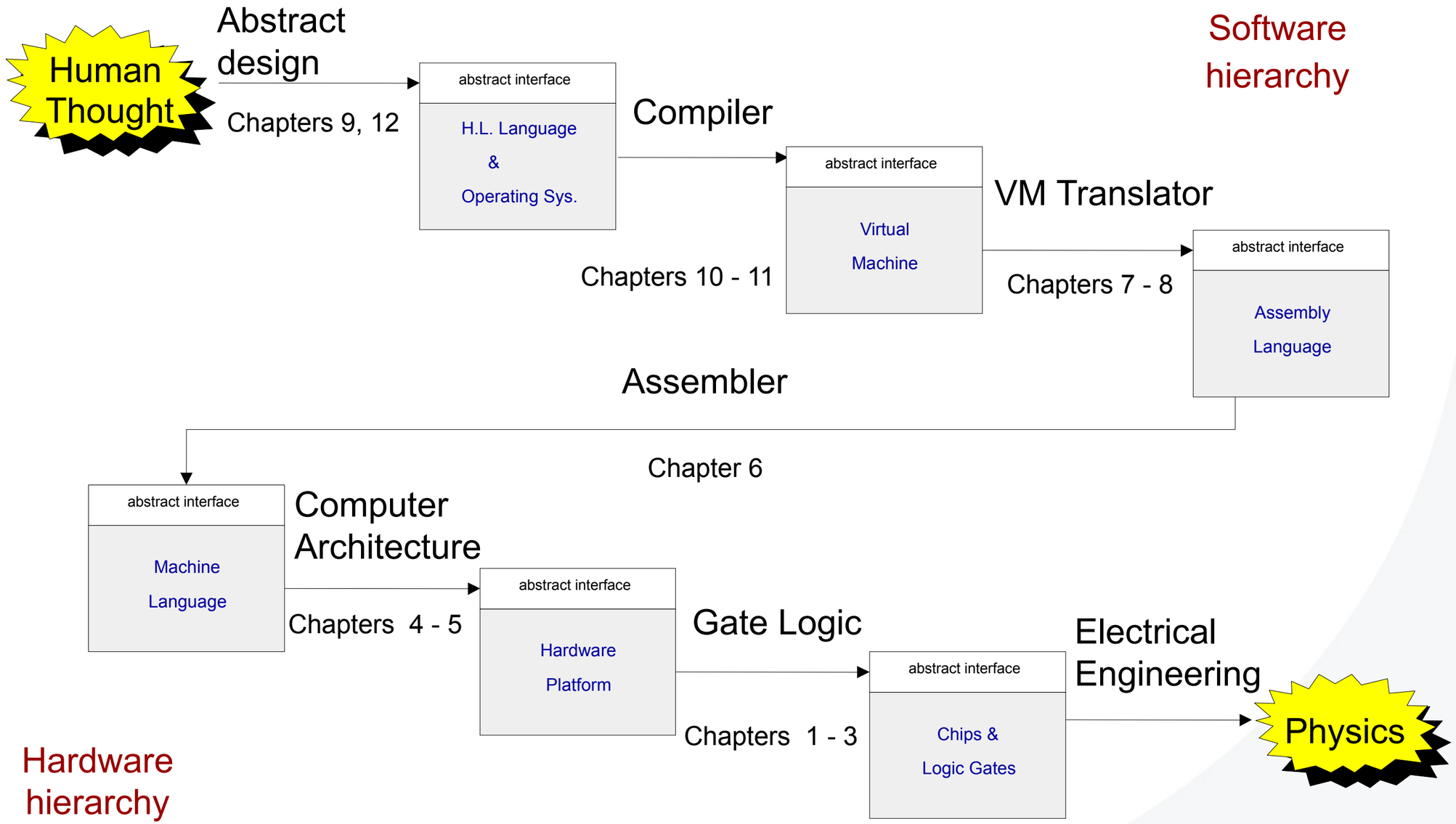
- These will help you practice and learn the concepts needed to pass the exams.

Plan to attend Workshops/Online Discussions

- This is where you will get the most individual feedback and is the best chance to ask questions.
- Join PASS sessions if you need extra help.

THE UNIVERSITY of ADELAIDE

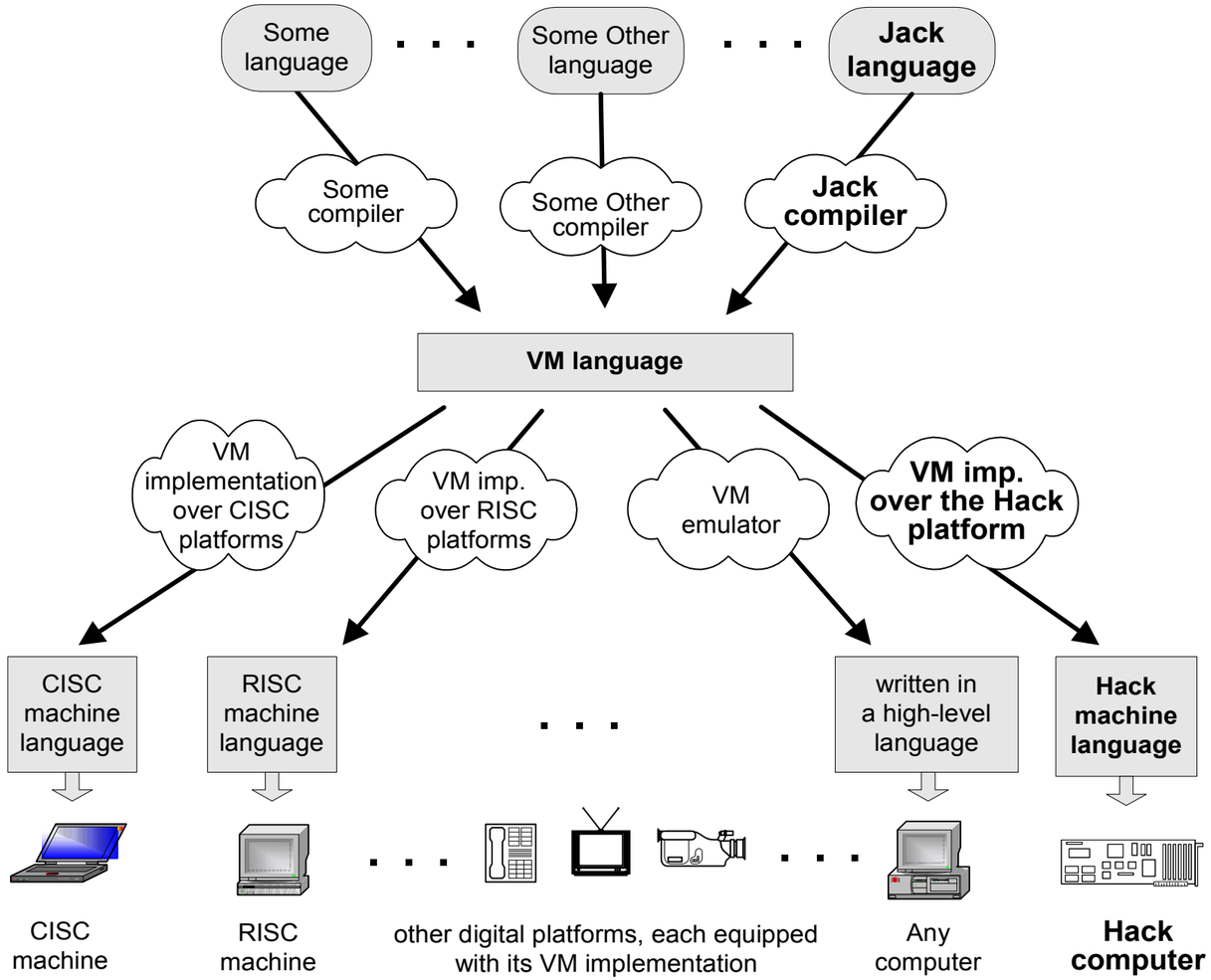
The whole system



(Abstraction–implementation paradigm)

THE UNIVERSITY of ADELAIDE

A modern compilation model

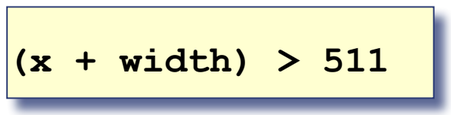


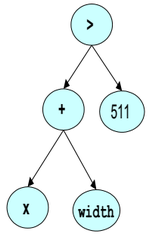
THE UNIVERSITY of ADELAIDE

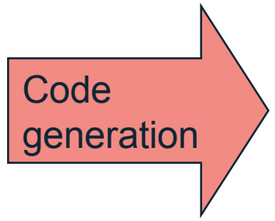
Compilation

Source code

```
(x + width) > 511
```







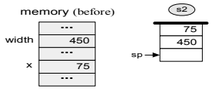
Intermediate code

```
push x
push width
add
push 511
gt
```

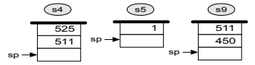
Abstraction Syntax Analysis Parse Tree Semantic Synthesis

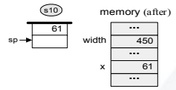
Inside a virtual machine

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



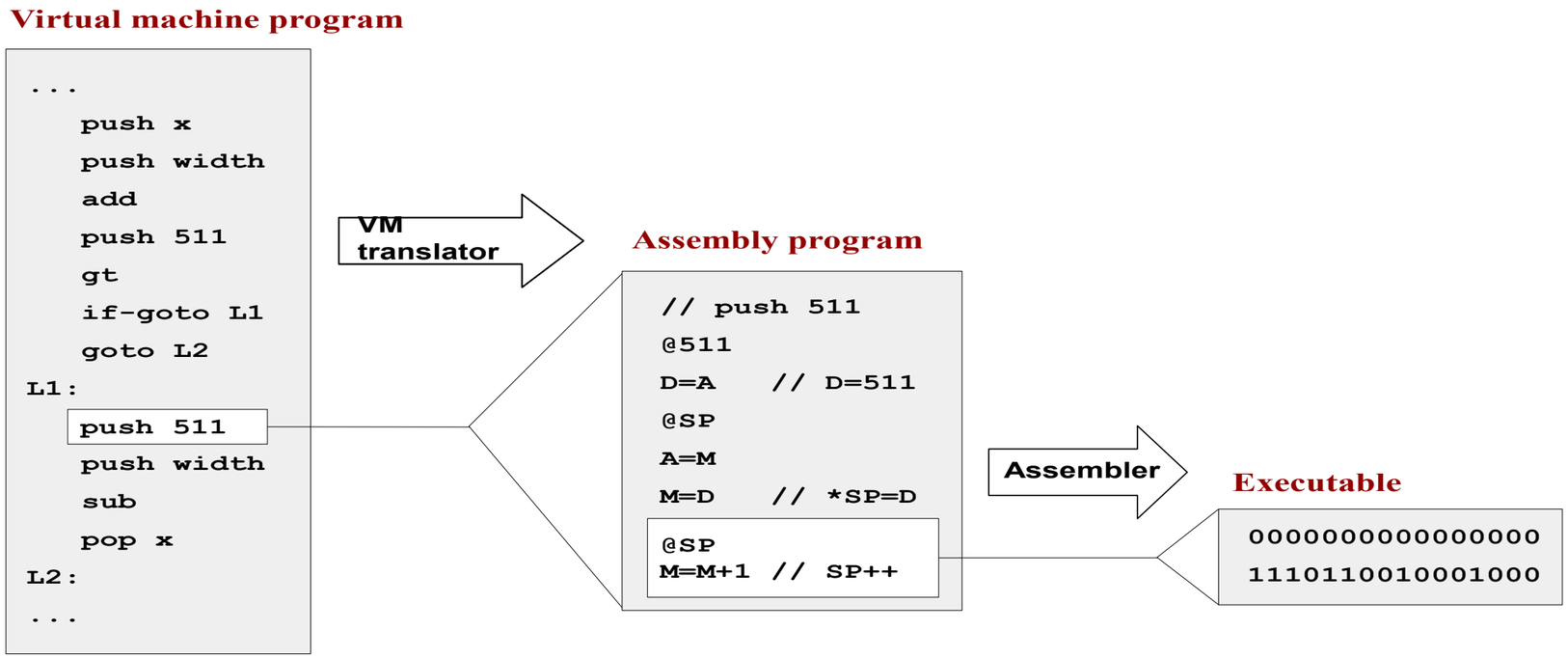

```
// VM implementation
   push x        // s1
   push width    // s2
   add           // s3
   push 511      // s4
   gt            // s5
   if-goto L1    // s6
   goto L2       // s7
L1:
   push 511      // s8
   push width    // s9
   sub           // s10
   pop x         // s11
L2:
...
```





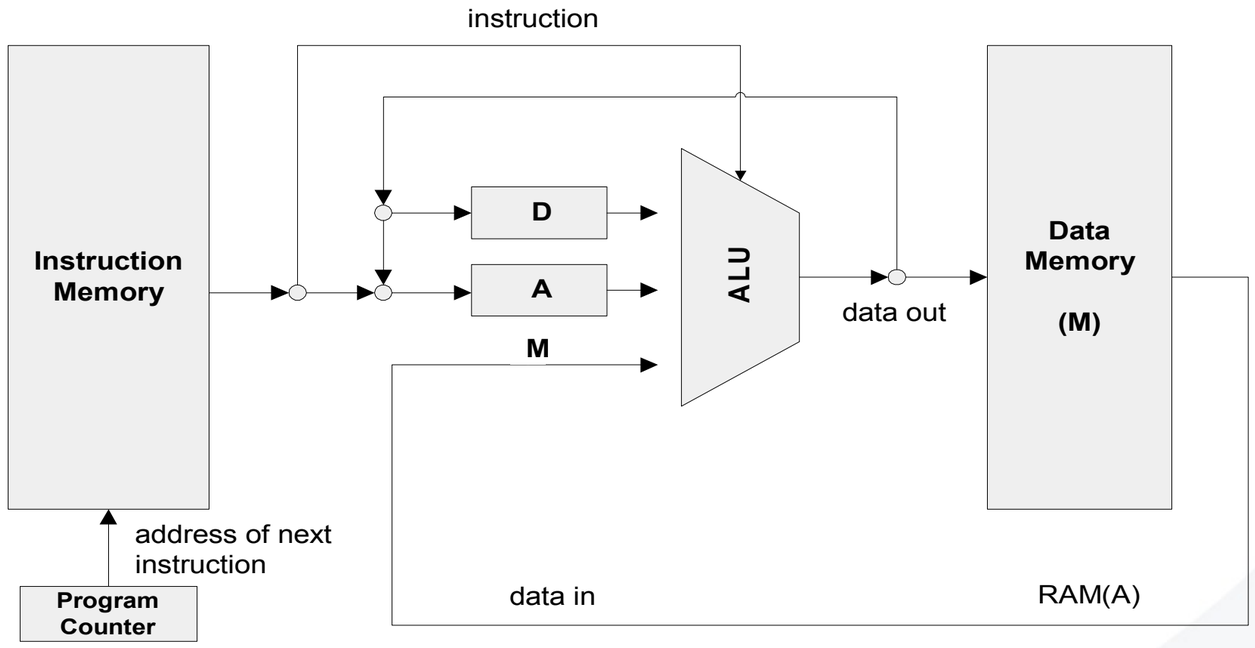
The low-level programming path

For now, ignore all details!

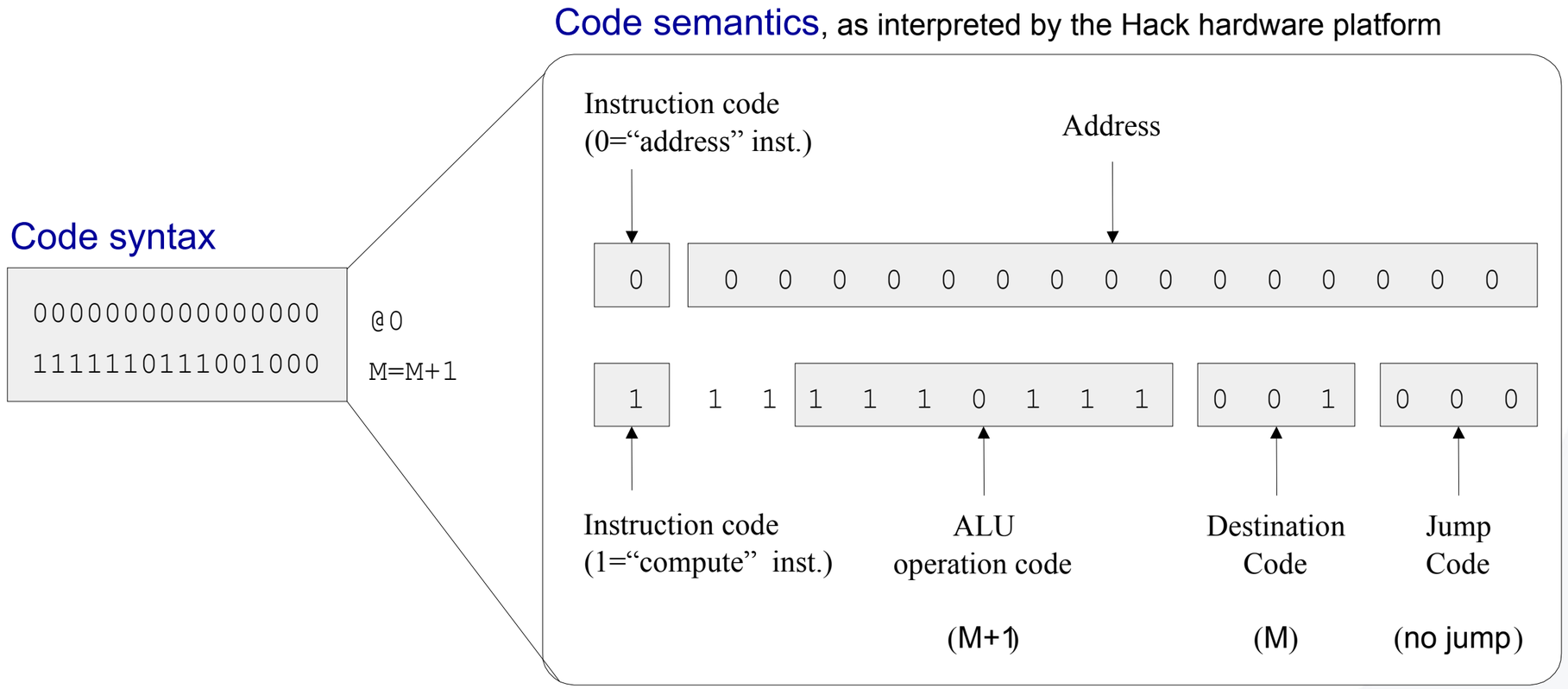


(template)

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 calculation (e.g., ALU)

Sequential logic – used for storage (e.g., RAM)

Gate logic – basic building blocks, putting it all together to get a working 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 (Nand gate in our course).

Logic gates are hardware implementations of Boolean functions. This allows us to represent logical statements in computer form.

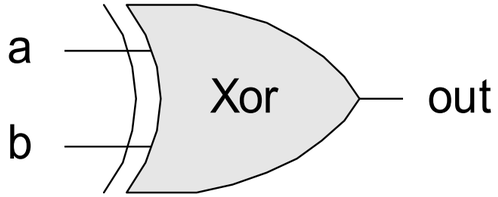
Every chip and gate has:

An interface: Telling us what it does

An implementation: Telling us how it does it.

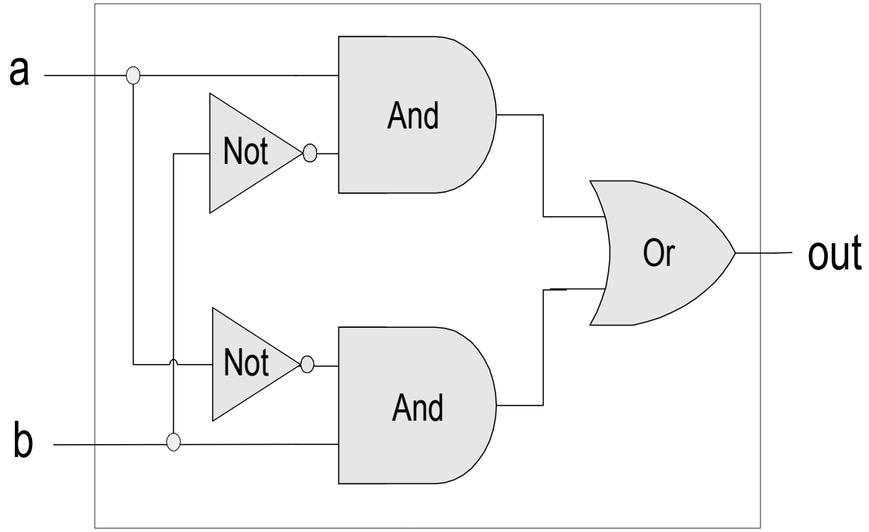
Example

Interface



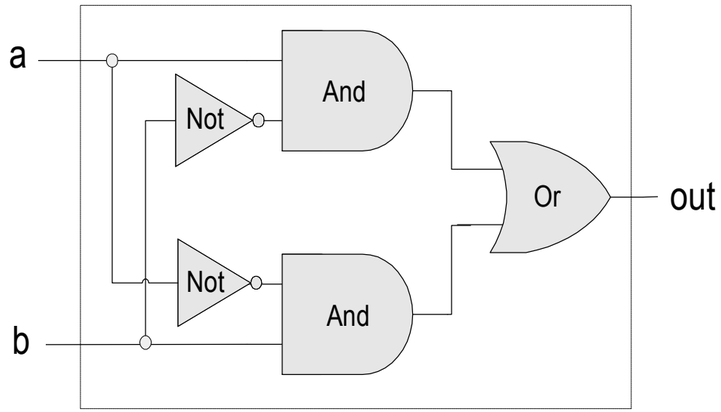
a	b	out
0	0	0
0	1	1
1	0	1
1	1	0

Implementation



Building gates

We won’t be building physic 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.

What's Happening

- Thursday's Lecture: more detail on boolean logic and logic gates.
- Workshops start **week 2**
- Review Chapter 1 of the Text Book (if you haven't already)