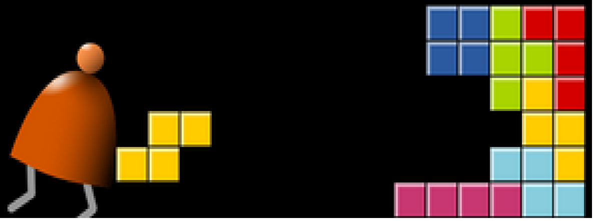
From Nand to Tetris

building a Modern Computer from First Principles



Nand to Tetris, Part I: Hardware

Slide deck for the “Part I: Hardware” chapter of the book

The Elements of Computing Systems (2nd edition)

By Noam Nisan and Shimon Schocken

MIT Press

Audience

students

developers

How computers work?

How they are built?

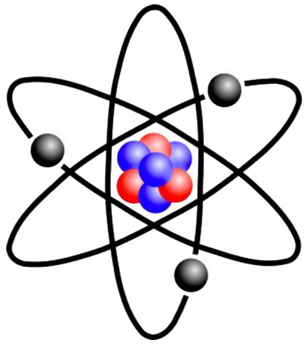
How to become a better thinker / builder?

BANG

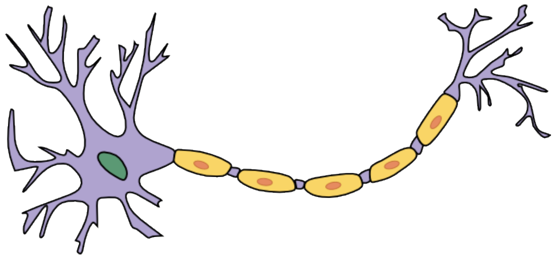
... 1001010100101101010
010010100101001010101
110011010001010010010
010011110010001000111
1111011010100101101 ...

Bits

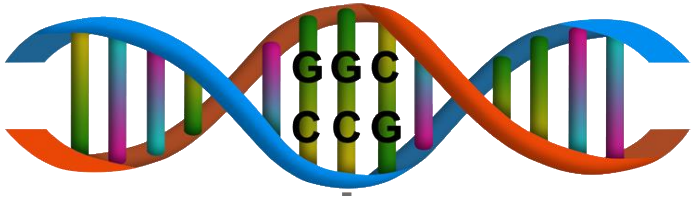
✓



Atoms



Neurons



Genes

Hello World

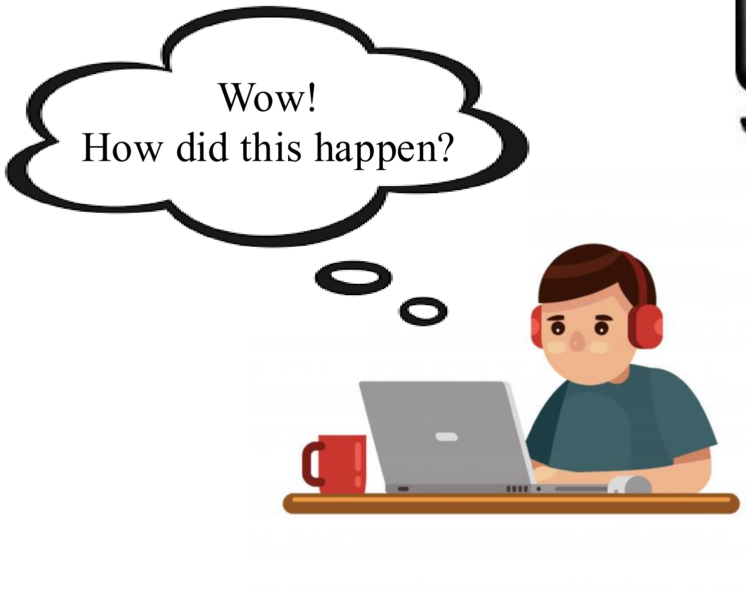
Java / Python

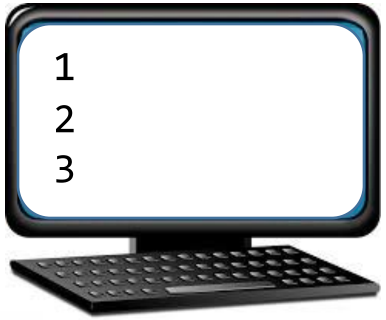
```
// Prints some numbers
i = 1
while (i < 4) {
   print(i);
   i = i + 1;
}
...
```

Hello World

Java / Python

```
// Prints some numbers
i = 1
while (i < 4) {
   print(i);
   i = i + 1;
}
...
```





```
1
2
3
```

Hello, World Below

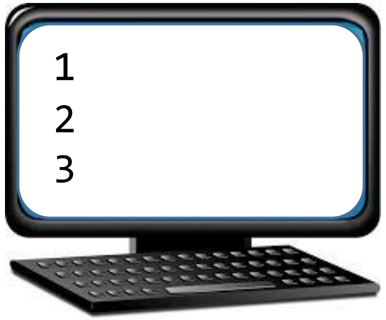
Java / Python

```
// Prints some numbers
i = 1
while (i < 4) {
   print(i);
   i = i + 1;
}
...
```

translate

Binary code

```
0000000000010000
1110111111001000
0000000000010001
1110101010001000
0000000000010000
1111110000010000
0000000000000000
1111010011010000
0000000000010010
1110001100000001
0000000000010000
1111110000010000
...
```



```
1
2
3
```

Hello, World Below

Java / Python

```
// Prints some numbers
i = 1
while (i < 4) {
   print(i);
   i = i + 1;
}
...
```

translate

Binary code

```
0000000000010000
1110111111001000
0000000000010001
1110101010001000
0000000000010000
1111110000010000
0000000000000000
1111010011010000
0000000000010010
1110001100000001
0000000000010000
1111110000010000
...
```

execute

```
1
2
3
```

Hello, World Below

Java / Python

```
// Prints some numbers
i = 1
while (i < 4) {
   print(i);
   i = i + 1;
}
...
```

translate

Binary code

```
0000000000010000
1110111111001000
0000000000010001
1110101010001000
0000000000010000
1111110000010000
0000000000000000
1111010011010000
0000000000010010
1110001100000001
0000000000010000
1111110000010000
...
```

execute

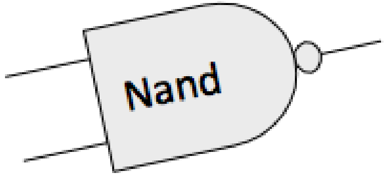
Software questions

- Loop?
- Program?
- Translation?
- …

Hardware questions

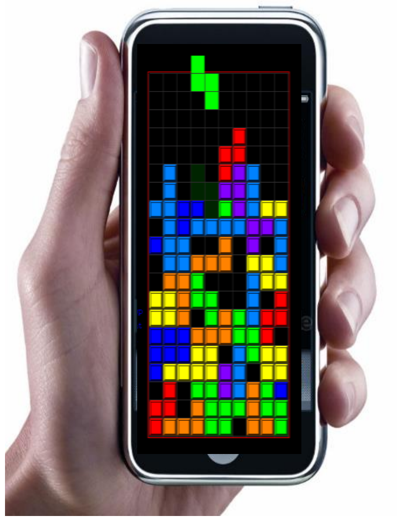
- Binary code?
- Computer?
- Screen?
- …

Nand to Tetris

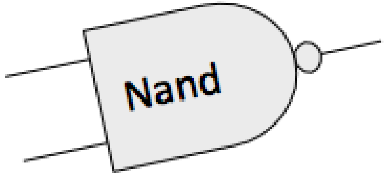


a	b	Nand
0	0	1
0	1	1
1	0	1
1	1	0

building a modern computer system from first principles



Nand to Tetris



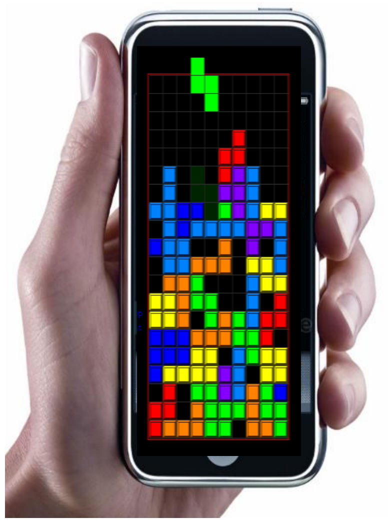
a	b	Nand
0	0	1
0	1	1
1	0	1
1	1	0

hardware platform

Projects 1-6

software hierarchy

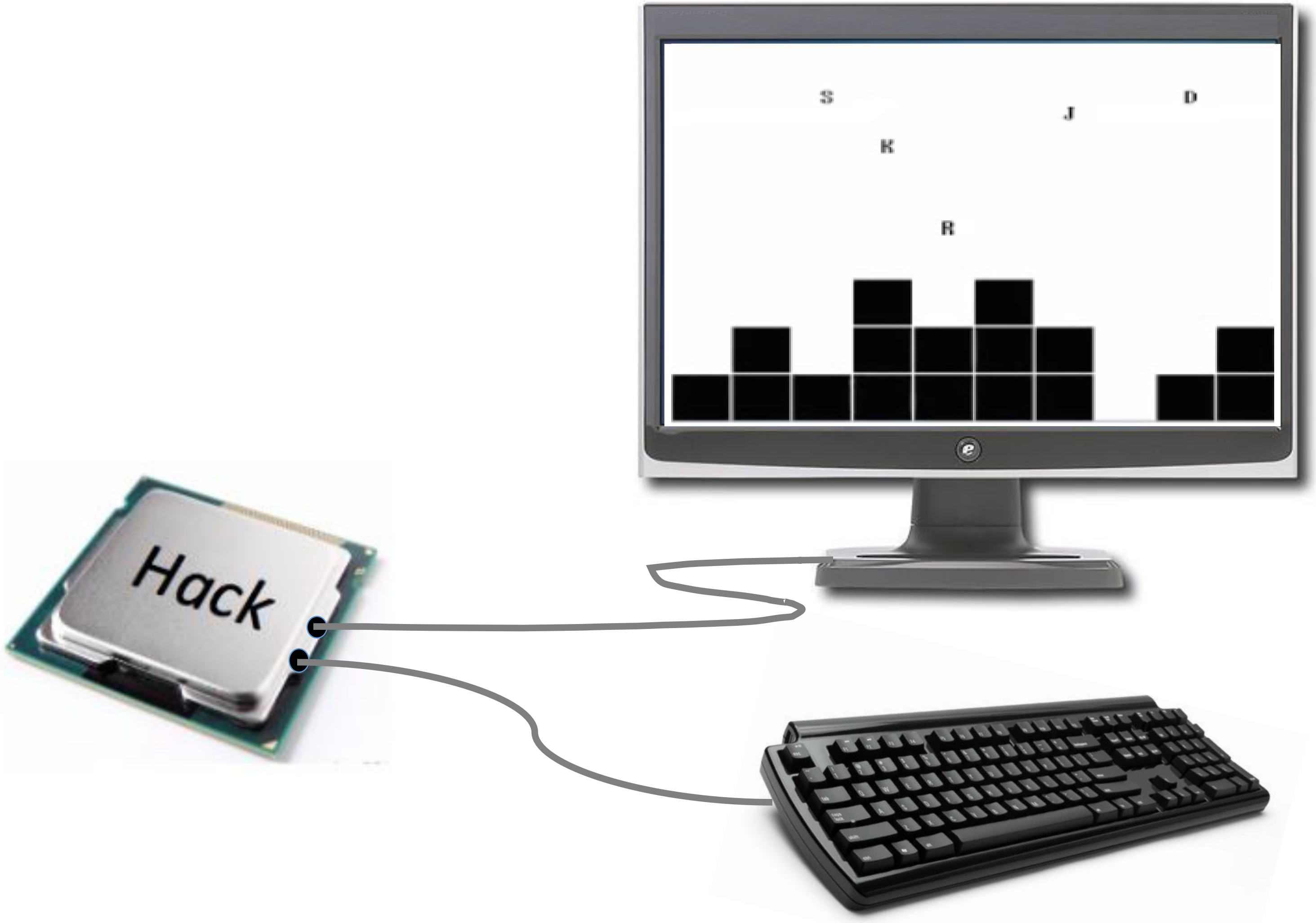
Projects 7-12



Using HDL and a hardware simulator

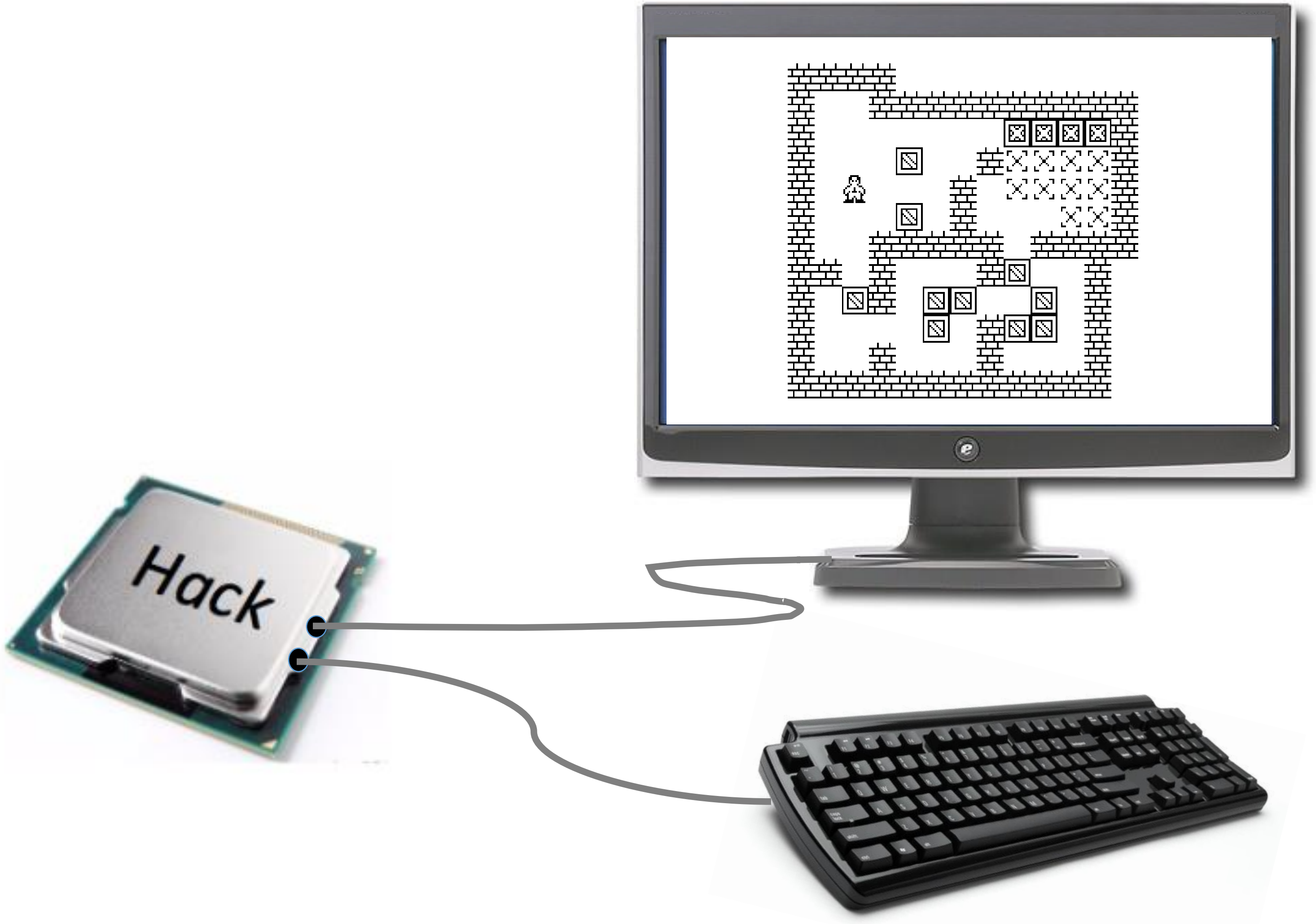
Using a programming language + supplied specs and test programs

Nand to Tetris (example programs)



(Ido Adler)

Nand to Tetris (example programs)



(Golan Parashi)

Nand to Tetris (example programs)

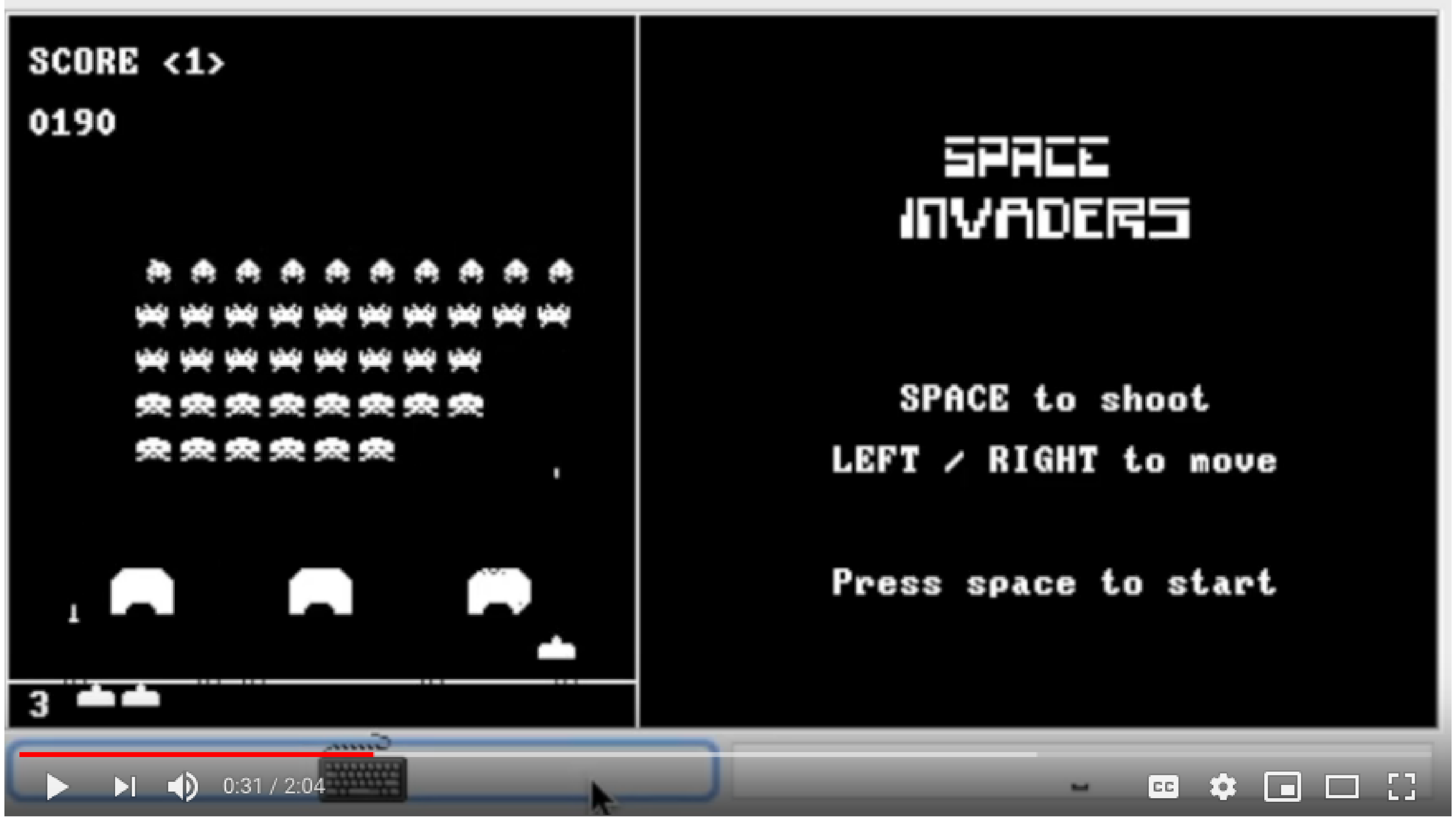
(Gavin Stewart)

Nand to Tetris (example programs)

Hack
RAWR!
Roast All Winged Rapscalions
Press "SPACE" to begin

(Kyle Schulz)

Nand to Tetris (example programs)



SCORE <1>

0190

3

SPACE INVADERS

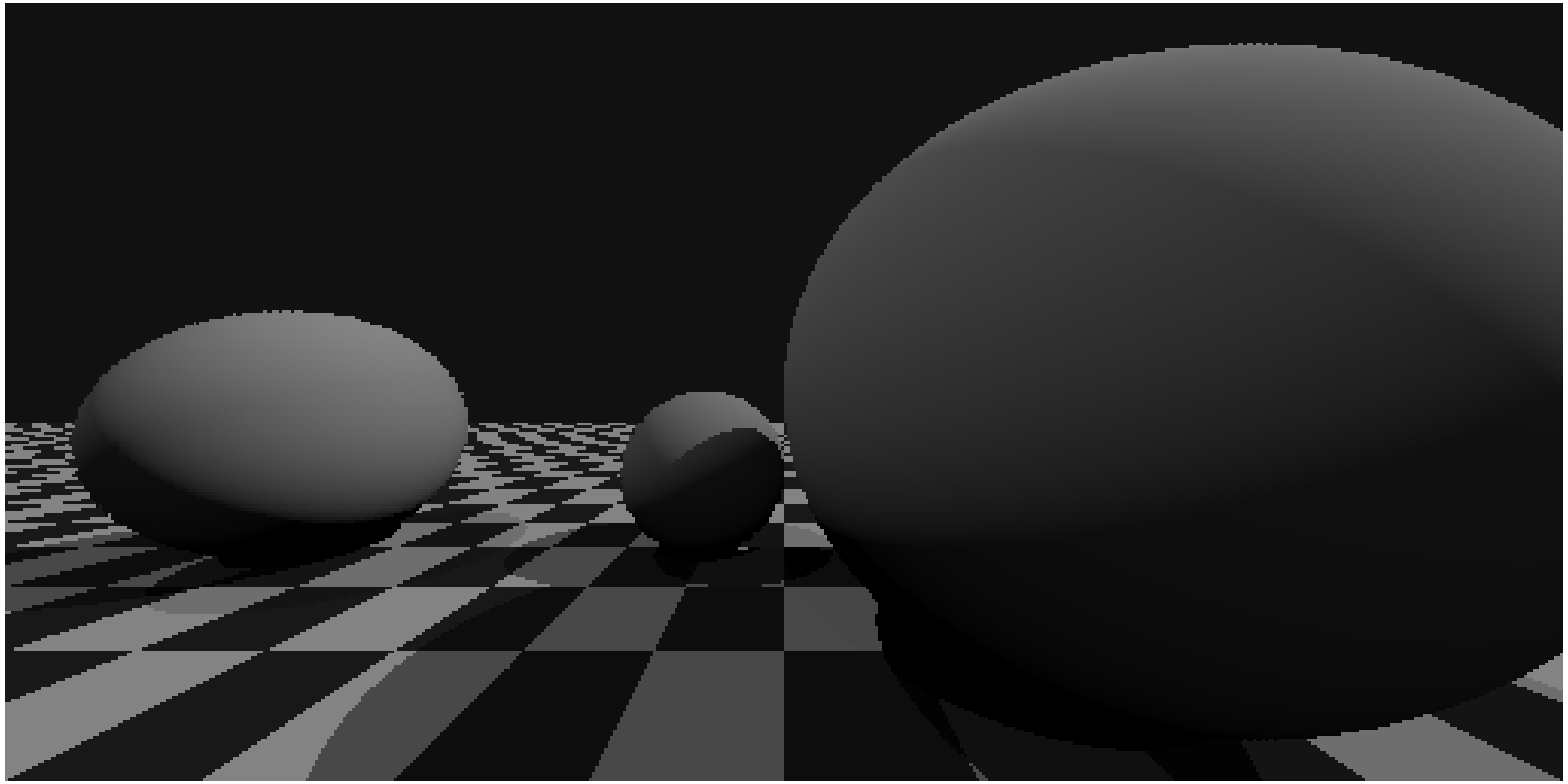
SPACE to shoot

LEFT / RIGHT to move

Press space to start

(James Connel)

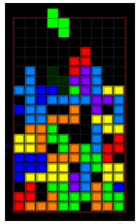
Nand to Tetris (example programs)



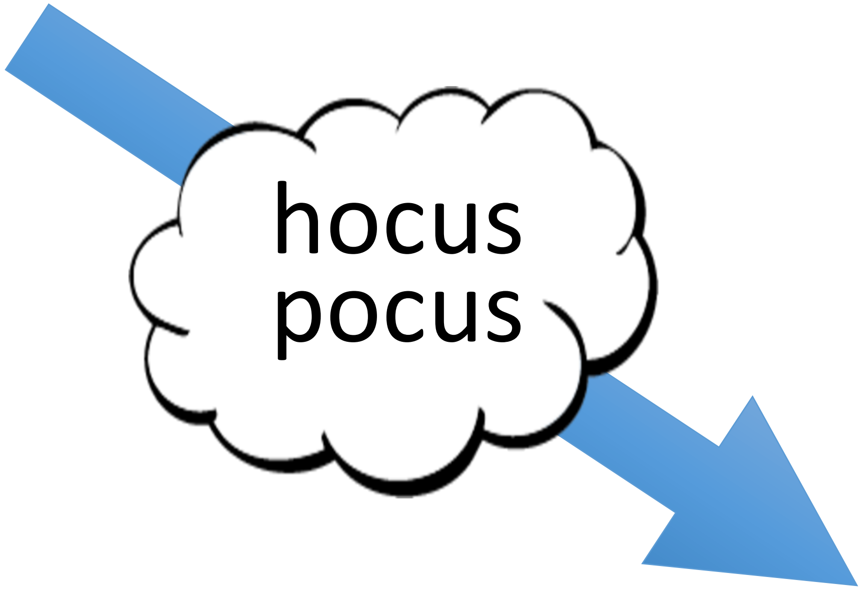
(Alex Quach)

CS view of EE

abstract concept



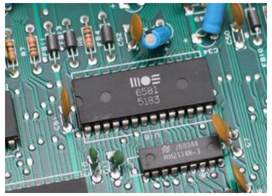
CS



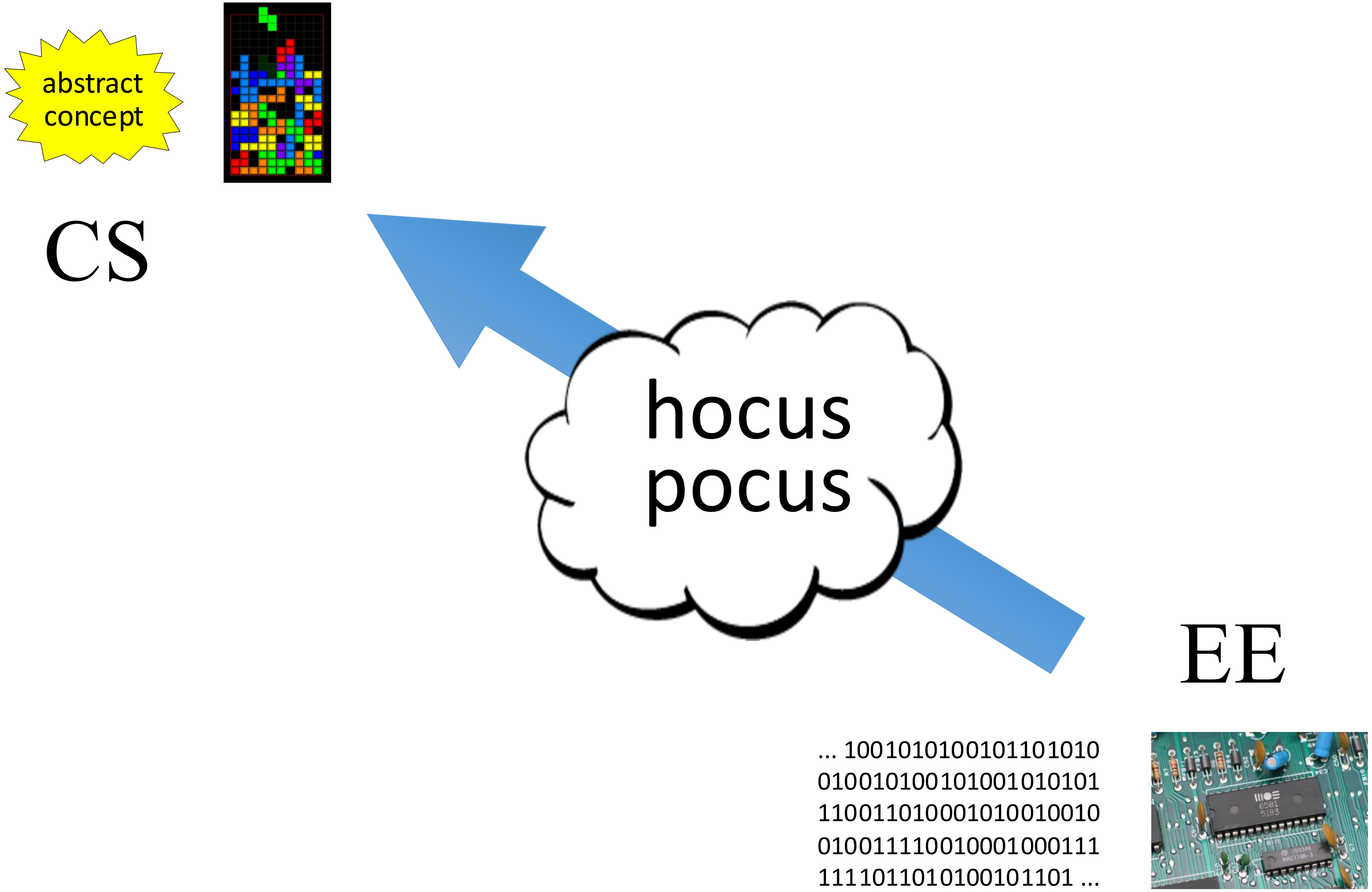
hocus pocus

EE

... 1001010100101101010
010010100101001010101
110011010001010010010
010011110010001000111
1111011010100101101 ...



EE view of CS



abstract concept

CS

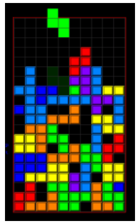
hocus pocus

EE

```
... 1001010100101101010
010010100101001010101
110011010001010010010
010011110010001000111
1111011010100101101 ...
```

Nand to Tetris

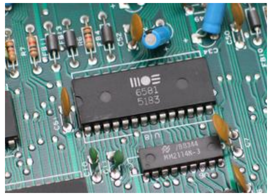
abstract concept



CS

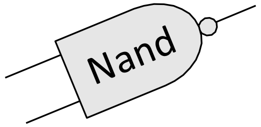
EE

... 1001010100101101010
010010100101001010101
110011010001010010010
010011110010001000111
1111011010100101101 ...

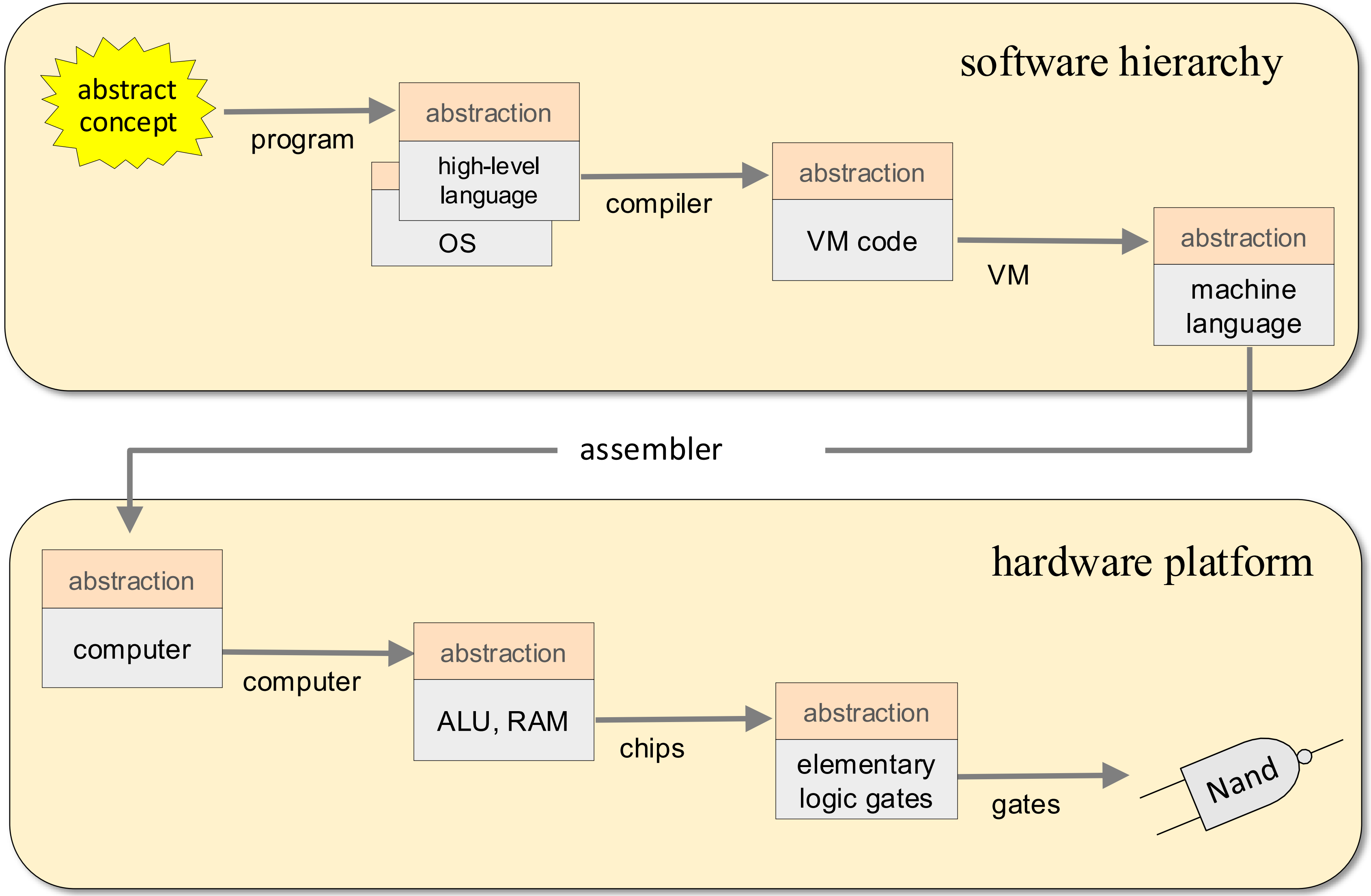


Nand to Tetris

abstract concept



Nand to Tetris Roadmap



software hierarchy

abstract concept

program

abstraction

high-level language

OS

compiler

abstraction

VM code

VM

abstraction

machine language

assembler

hardware platform

abstraction

computer

computer

abstraction

ALU, RAM

chips

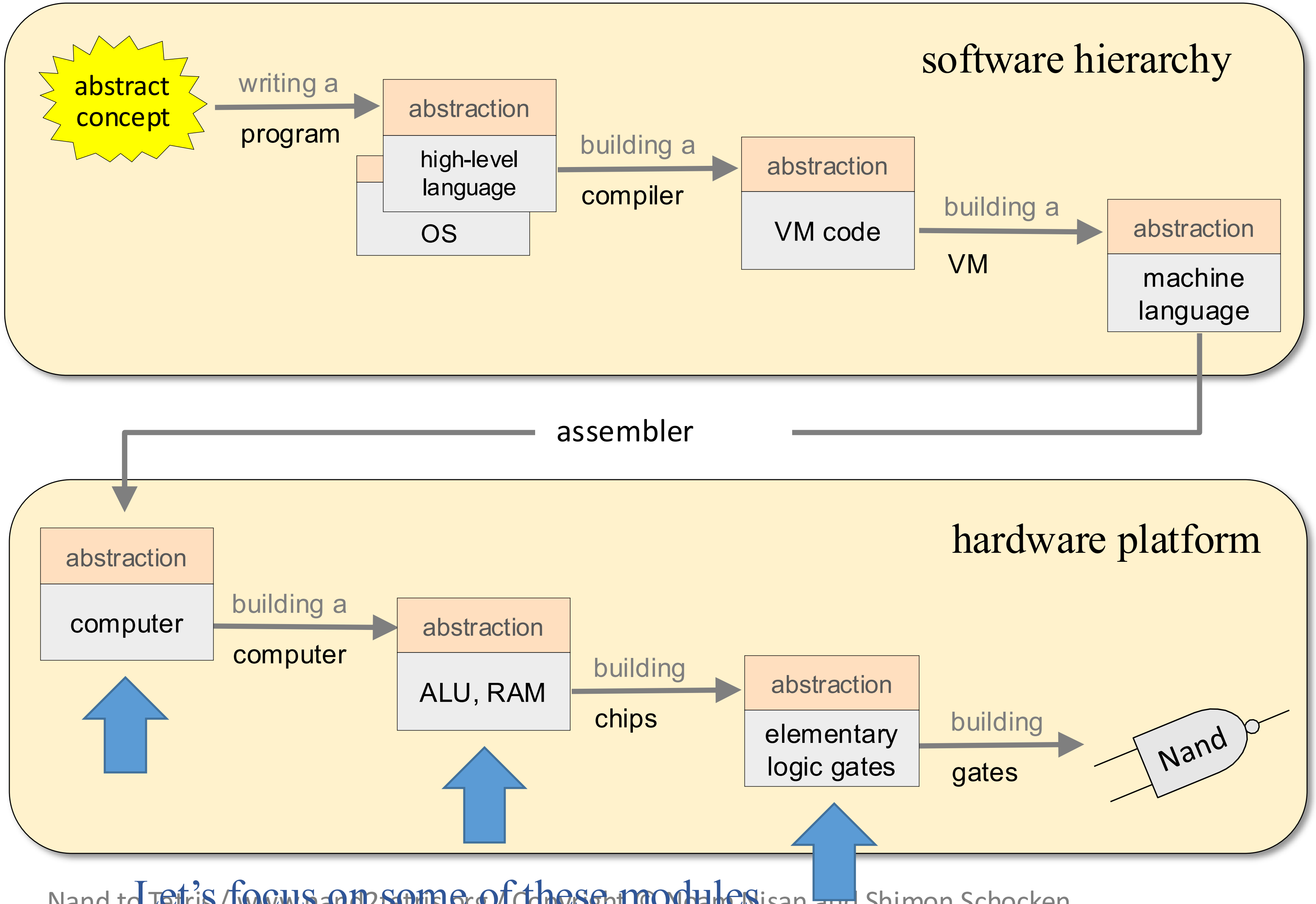
abstraction

elementary logic gates

gates

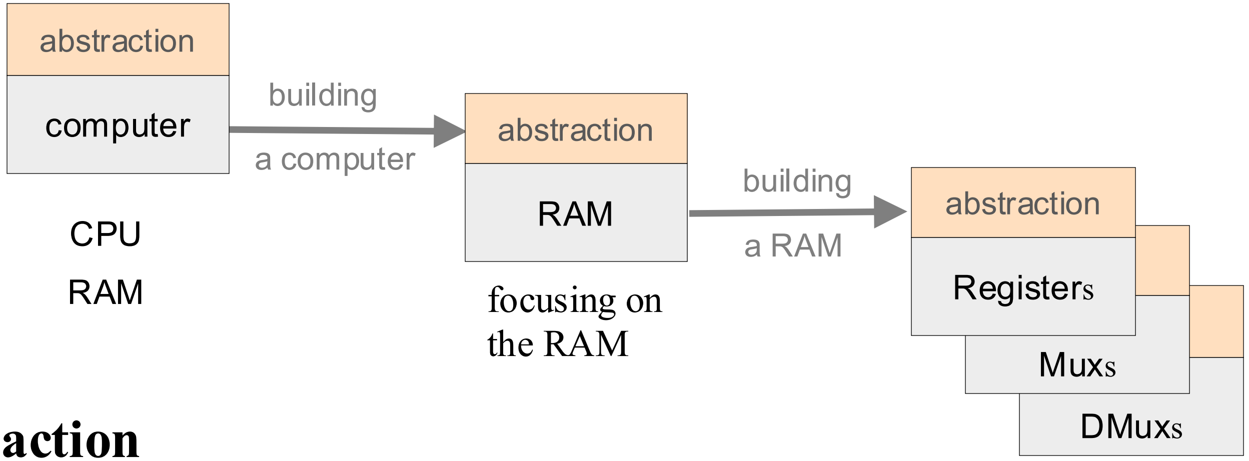
Nand

Abstraction / Implementation



Let’s focus on some of these modules

Abstraction / Implementation (example)



Abstraction

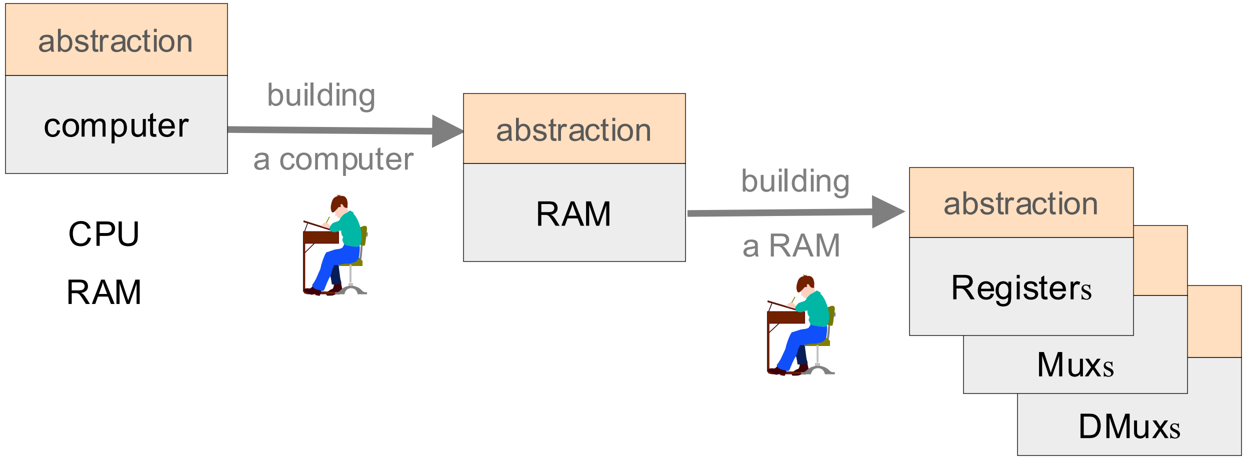
What the module is designed to do,

AKA the module’s *interface*

Implementation

How the module’s functionality is realized

Abstraction / Implementation (example)



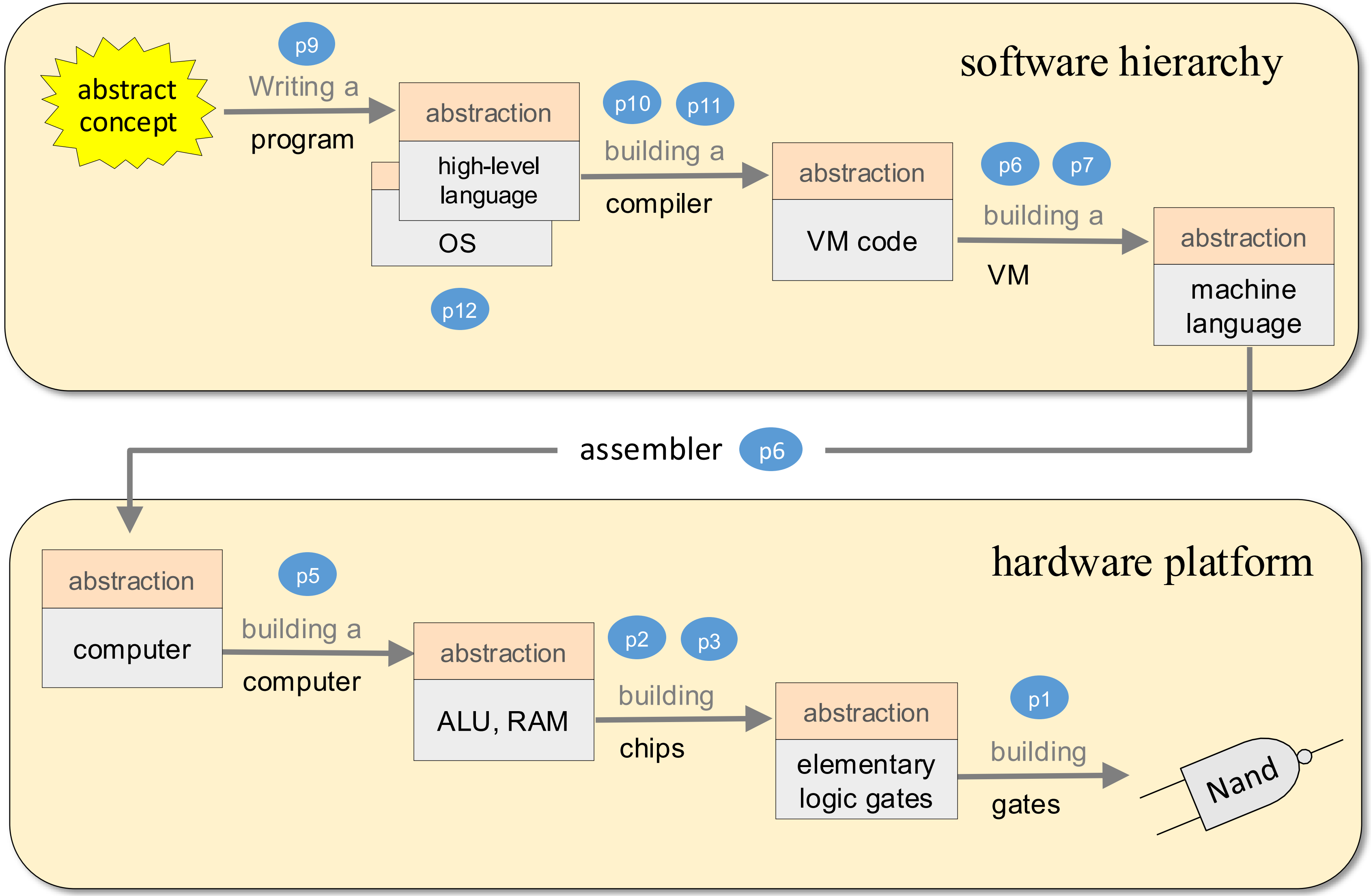
When implementing a module

- Use modules from the level below, as *abstract building blocks*
- Arrange them in a way that delivers the module's functionality

When using a module as a building block:

- Focus on *what* the module is designed to do
- Ignore *how* the module is implemented.

Nand to Tetris Roadmap



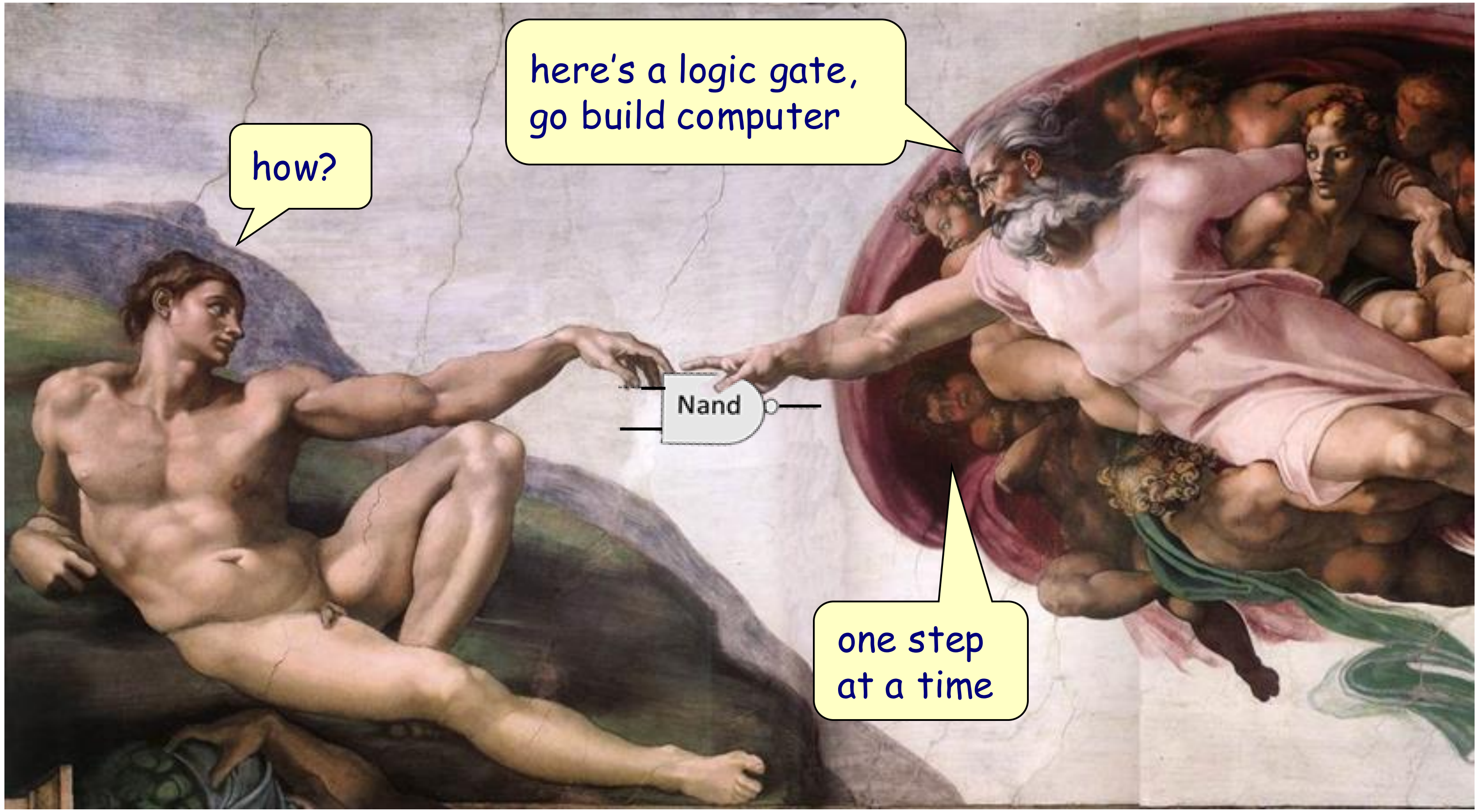
Module

A well-specified sub-system that can be implemented and unit-tested independent of other modules

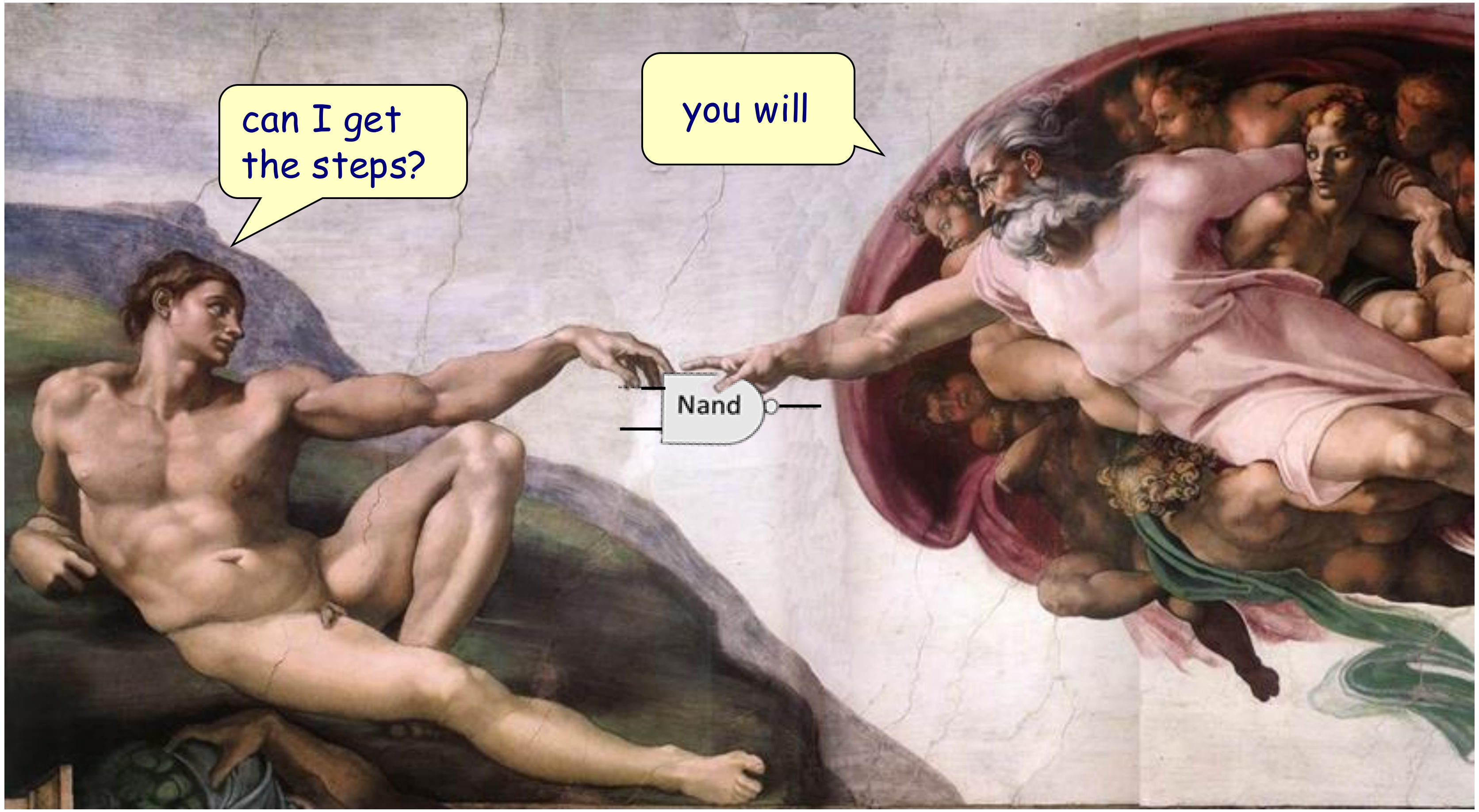
Modular design

Breaking a complex system into a "good" set of modules.

Nand to Tetris: Methodology



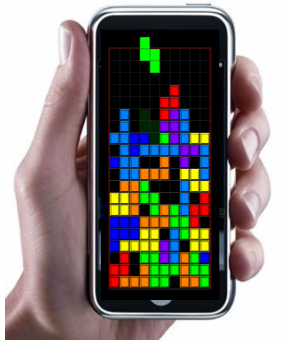
Nand to Tetris: Methodology



Nand to Tetris: Take home lessons

a	b	Nand
0	0	1
0	1	1
1	0	1
1	1	0

Nand to Tetris



Hardware: Logic gates, Boolean arithmetic, multiplexors, flip-flops, registers, RAM units, counters, Hardware Description Language, chip simulation and testing.

Architecture: ALU/CPU design and implementation, addressing modes, memory-mapped I/O, machine code, assembly language programming,

Programming Languages: Object-based design and programming, abstract data types, scoping rules, syntax and semantics, references.

Compilation: Lexical analysis, top-down parsing, symbol tables, pushdown automata, virtual machine, code generation, implementation of arrays and objects.

Data structures and algorithms: Stacks, trees, hash tables, lists, recursion, arithmetic algorithms, geometric algorithms, time / space complexity

Engineering: Abstraction / implementation, modular design, API design and documentation, unit testing, quality assurance, programming at the large.