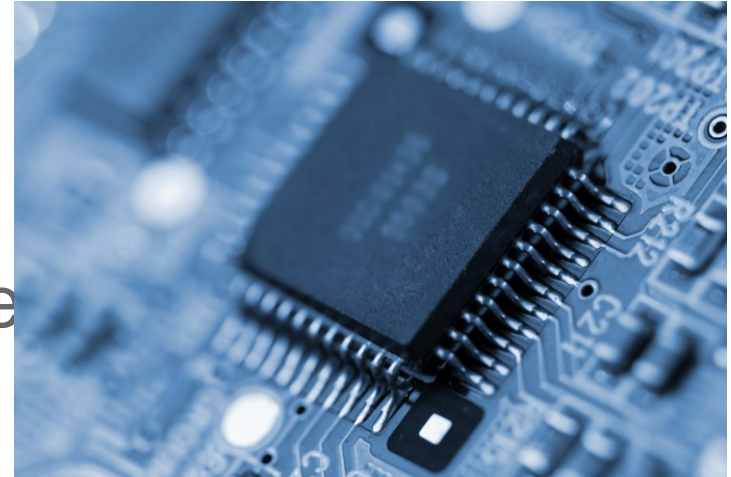




# Computer Processors

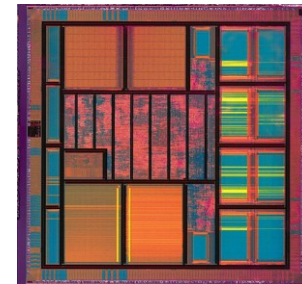
Hardware Description Language  
& Computer Aided Design



- Computer Aided Design (CAD) refers to the process of design and testing of systems using computers
- CAD may automate some of the design and testing process
- Helps to ensure that the design conforms to a specification
- Encourages reuse of common components
- Often supports simulations

# Hardware Description Language

- Hardware Description Language (HDL) is a language used to specify the internal wiring of a chip
- The specified chip can then be tested against a range of tests to ensure compliance to a specification
- HDL specification can be simulated
- HDL can be 'blown' onto an Field Programmable Gate Array (FPGA) or dedicate silicon dies



# Hardware Description Language

- A HDL definition defines the internal wiring of a chip and consists of 2 parts
  - A *header* section - defining the interface of the chip
  - A *parts* section - defining the names and internal topology

```
/**  
 * Xor gate.  
 */
```

```
CHIP Xor {  
    IN a, b;  
    OUT out;
```

Header

PARTS:

```
And (a=a, b=b, out=x);  
Or (a=a, b=b, out=y);  
Not (in=x, out=notx);  
And (a=y, b=notx, out=out);
```

Parts

```
}
```

# Defining a chip in HDL

- Each chip is defined in a separate text file
- A chip called Xxx is defined in a file called Xxx.hdl (case sensitive!!)
- Keywords are capitalised
- Identifier name may be any sequence of letters and digits not starting with a digit
- Whitespace has no meaning
- Lines end with semicolons
- Comments are provided in a variety of way

```
// Comments to end of line
/* Comment until closing */
/** API documentation */
```

```
/**
 * Xor gate.
 */
```

```
CHIP Xor {

    IN  a, b;
    OUT out;

    PARTS:
    And (a=a, b=b, out=x);
    Or  (a=a, b=b, out=y);
    Not (in=x, out=notx);
    And (a=y, b=notx, out=out);

}
```

Xor.hdl

Identifiers

Keywords

Comments

# Defining a chip in HDL

- Internal chip parts
  - parts pin name = chips pin name
- Internal pins
  - Visible only within the chip
  - No need to define them, they are generated when required
  - Fan-in 1 / Fan-out unlimited
- true / false are reserved constants

```
/**  
 * Xor gate.  
 */  
  
CHIP Xor {  
  
    IN a, b;  
    OUT out;  
  
    PARTS:  
    And (a=a, b=b, out=x);  
    Or (a=a, b=b, out=y);  
    Not (in=x, out=notx);  
    And (a=y, b=notx, out=out);  
  
}
```

Xor.hdl

Part pin

Internal pins

# Defining a chip in HDL

- Buses are a collection of pins
- Buses can be used as input, output or internal pins
- Buses are indexed (similar to arrays in C)

```
CHIP Foo {  
  IN in[8];  
  OUT out[8];  
  
  // irrelevant internals of foo  
}
```

```
Foo(in[2..4]=v, in[6..7]=true, out[0..3]=x,  
out[2..5]=y)
```

0	1	2	3	4	5	6	7
?	?	v[0]	v[1]	v[2]	?	1	1

0	1	2	3	4	5	6	7
0	1	0	0	0	1	1	1

0	1	2	3
0	1	0	0

0	1	2	3
0	0	0	1

# Defining a chip in HDL

So much more is possible with HDL

- Clocked inputs
- Feedback loops
- Built-in chips

Appendix A - The Elements of Computer Systems

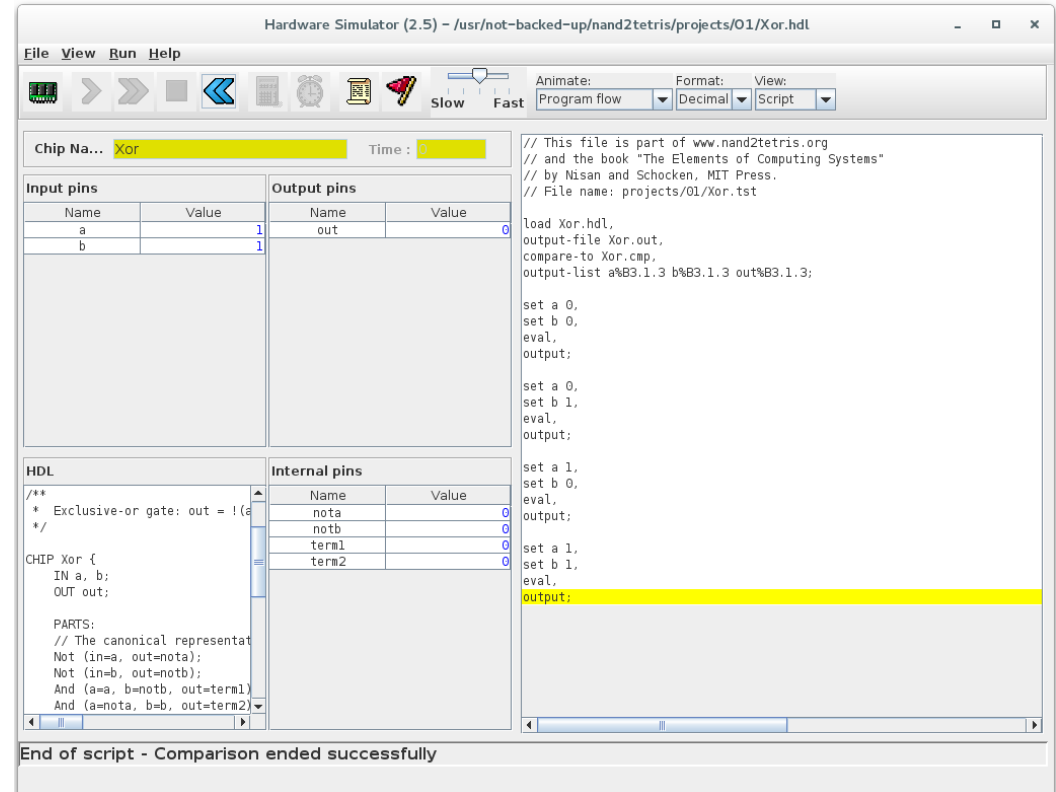


# Hardware Simulator

- Simulate HDL chips
- Automated test scripts

Available for download from Minerva

Watch the Video



# Hardware Simulator

Hardware Simulator (2.5) - /usr/not-backed-up/nand2tetris/projects/01/Xor.hdl

File View Run Help

Animate: Program flow Format: Decimal View: Script

Slow Fast

Chip Na... Xor Time: 0

Input pins		Output pins	
Name	Value	Name	Value
a	1	out	0
b	1		

```
HDL
/**
 * Exclusive-or gate: out = !(a == b)
 */
CHIP Xor {
    IN a, b;
    OUT out;

    PARTS:
    // The canonical representation
    Not (in=a, out=nota);
    Not (in=b, out=notb);
    And (a=a, b=notb, out=term1);
    And (a=nota, b=b, out=term2);
}
```

Internal pins	
Name	Value
nota	0
notb	0
term1	0
term2	0

```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Xor.tst

load Xor.hdl,
output-file Xor.out,
compare-to Xor.cmp,
output-list a%B3.1.3 b%B3.1.3 out%B3.1.3;

set a 0,
set b 0,
eval,
output;

set a 0,
set b 1,
eval,
output;

set a 1,
set b 0,
eval,
output;

set a 1,
set b 1,
eval,
output;
```

End of script - Comparison ended successfully

- Introduced the concept of Computer Aided Design (CAD)
- Introduced Hardware Description Language (HDL)
- Introduced the hardware simulator
- Demonstrated the development of a composite logic gate

Interested in logic design? [Digital Logic Design](#)