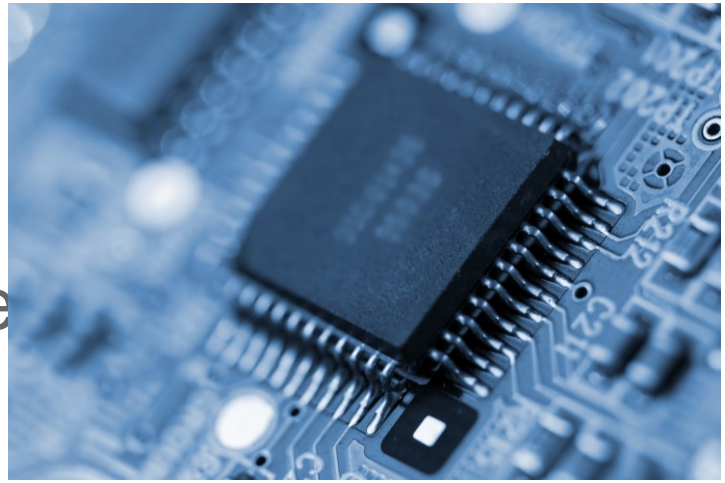
UNIVERSITY OF LEEDS

Computer Processors

Hardware Description Language & Computer Aided Design



Lecture 4

CAD

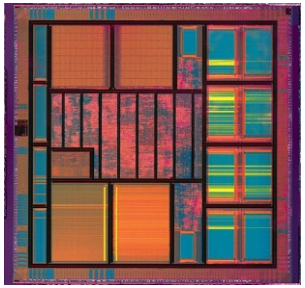
UNIVERSITY OF LEEDS

- 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

UNIVERSITY OF LEEDS

- 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

UNIVERSITY OF LEEDS

- 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;

    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);
}
```

Header

Parts

Defining a chip in HDL

UNIVERSITY OF LEEDS

- 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

UNIVERSITY OF LEEDS

- 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

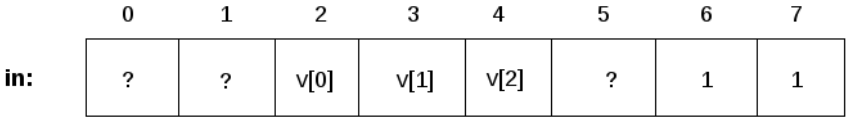
Part pin

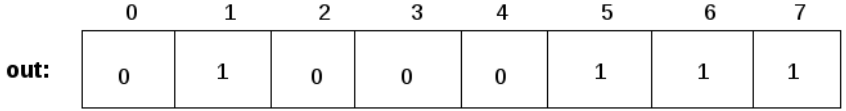
Internal pins

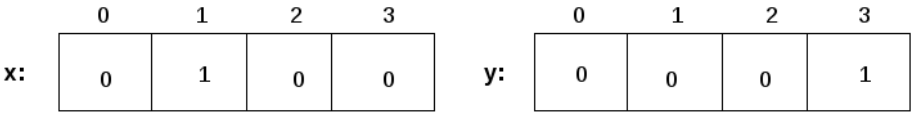
Defining a chip in HDL

UNIVERSITY OF LEEDS

- Buses are a collection of pins
- Buses can be used as input, output or interna 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)
```

Defining a chip in HDL

UNIVERSITY OF LEEDS

So much more is possible with HDL

- Clocked inputs
- Feedback loops
- Built-in chips

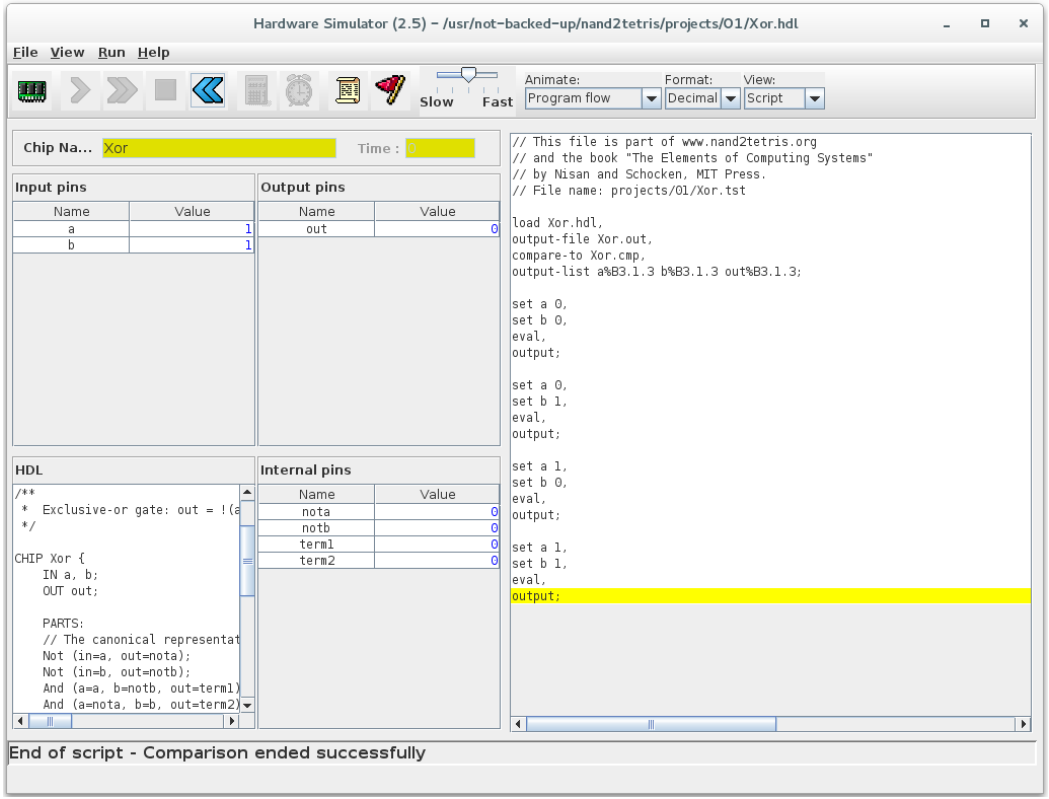
Hardware Simulator

UNIVERSITY OF LEEDS

- Simulate HDL chips
- Automated test scripts

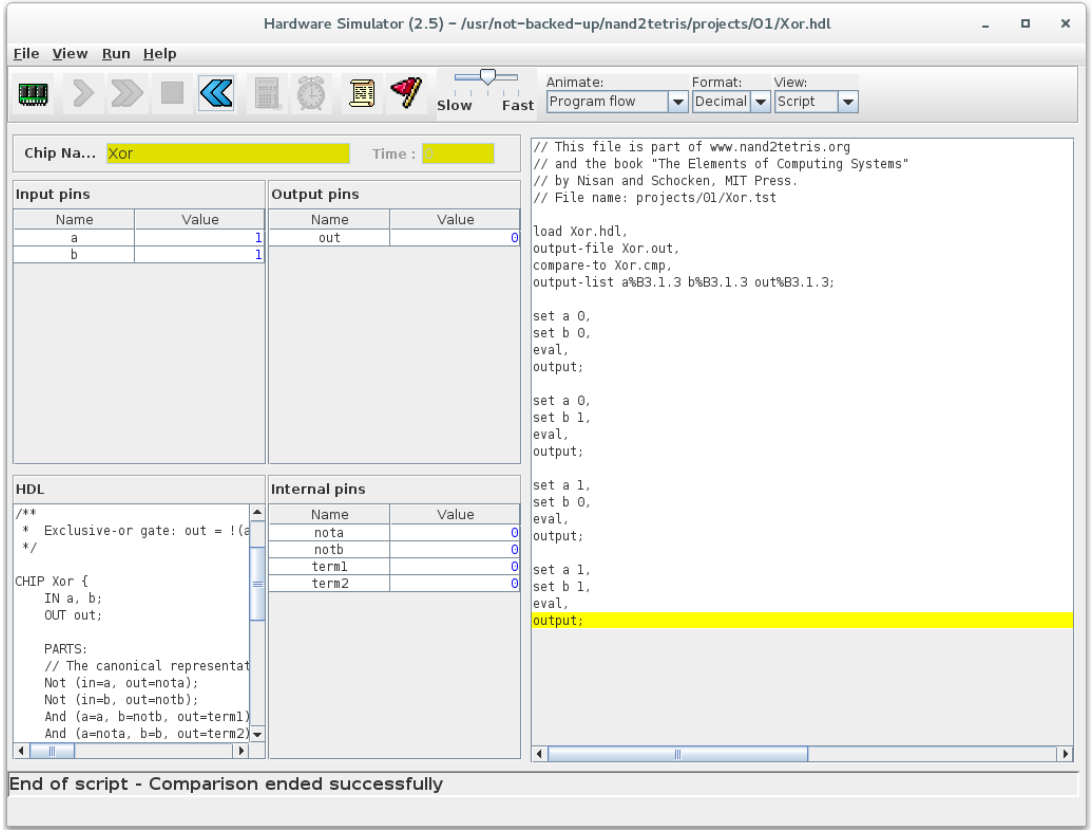
Available for download from Minerva

Watch the Video



Hardware Simulator

UNIVERSITY OF LEEDS



Summary

UNIVERSITY OF LEEDS

- 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