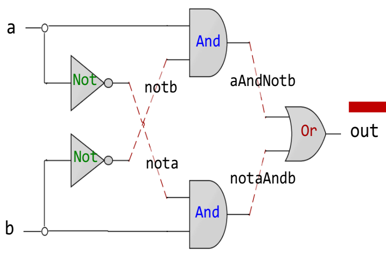
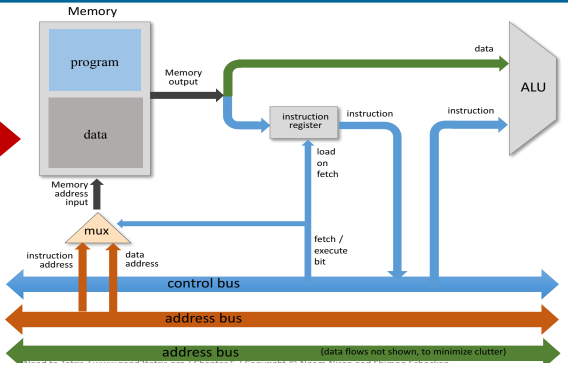
Computer Organization & Assembly Language Programming



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



```
@R1
D=M
@temp
M=D
```

```
0000000000000001
1111110000010000
0000000000010000
1110001100001000
```

Lecture # 03

HDL for Combinational Circuits - II

```
#include<stdio.h>
#include<stdlib.h>
int main(){
  printf("Learning is fun with Arif\n");
  exit(0);
}
```

```
global main
SECTION .data
   msg: db "Learning is fun with Arif", 0Ah, 0h
   len_msg: equ $ - msg
SECTION .text
   main:
      mov rax,1
      mov rdi,1
      mov rsi,msg
      mov rdx,len_msg
      syscall
      mov rax,60
      mov rdi,0
      syscall
```

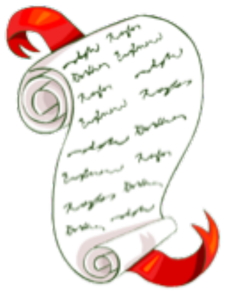
```
0:  b8 01 00 00 00
5:  bf 01 00 00 00
a:  48 be 00 00 00 00 00
11: 00 00 00
14: ba 1b 00 00 00
19: 0f 05
1b: b8 3c 00 00 00
20: bf 00 00 00 00
25: 0f 05
```

Slides of first half of the course are adapted from:
https://www.nand2tetris.org
Download s/w tools required for first half of the course from the following link:
https://drive.google.com/file/d/0B9c0BdDJz6XpZUh3X2dPR1o0MUE/view

Instructor: Muhammad Arif Butt, Ph.D.

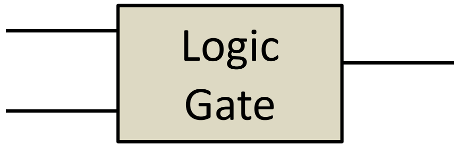
Today's Agenda

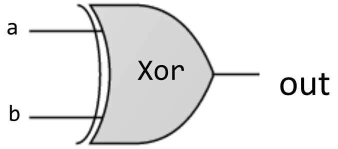
- Design Xor Chip using And, Or and Not gate chips
- Perform the Interactive Testing of Xor Chip
- Class Quiz
- What is Script Based Testing?
- Perform script based testing of Xor chip
- Players Involved in H/W Construction Project

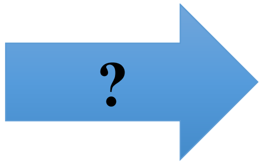


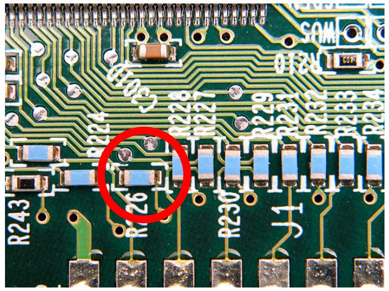
Designing Xor Chip

Designing and Building Xor Chip





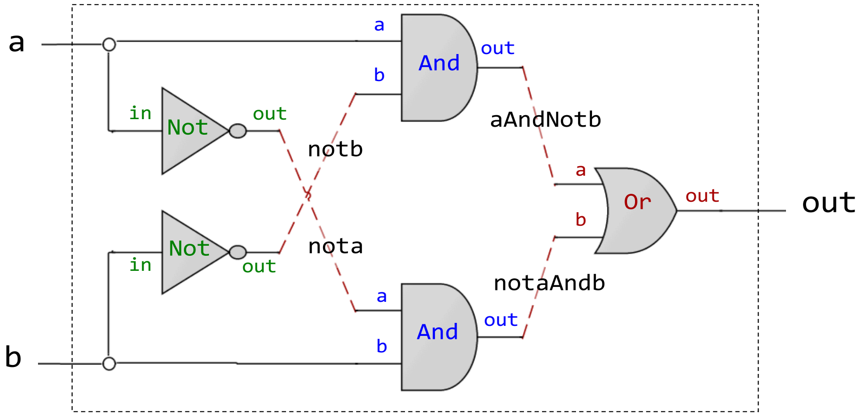




Output is 1 if one, and only one, of its inputs, is 1

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

out(a,b) = a'b + ab'



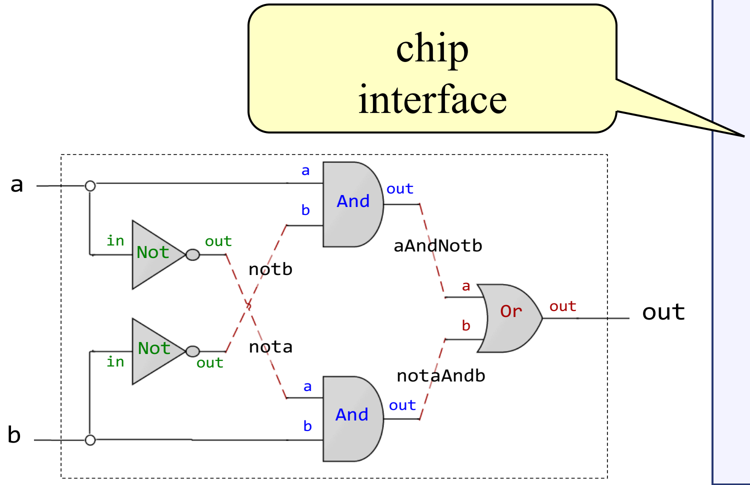
<u>Design Process:</u>

- From truth table derive the simplified Boolean Function
- Design the gate architecture
- Specify the architecture in HDL
- Test the chip in a hardware simulator
- Optimize the design
- Realize the optimized design in silicon

```
CHIP Xor {
    IN a, b;
    OUT out;
    PARTS:
// Chip Implementation
}
```

Instructor: Muhammad Arif Butt, Ph.D.

Chip Interface

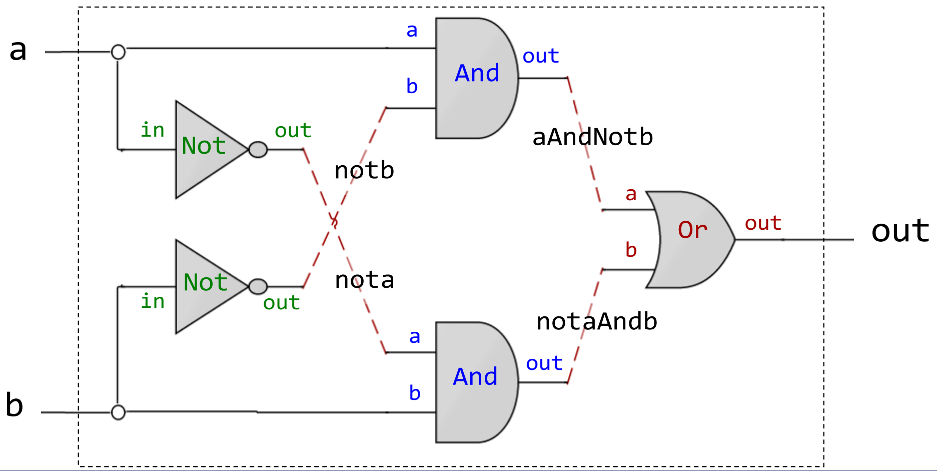


```
/** Exclusive-or gate. out = a xor b */
CHIP Xor {
    IN a, b;
    OUT out;

    // Implementation missing.
}
```

Chip Interface:

- Chip interface is typically supplied by the chip architect; similar to an API, or a contract, which contains:
  - ➢ Name of the chip
  - ➢ Names of its input and output pins
  - ➢ Documentation of the intended chip operation

HDL for Xor Chip



Xor.hdl

```
/** Xor gate: out = (a And Not(b)) Or (Not(a) And b)) */
 CHIP Xor {
    IN a, b;
    OUT out;
    PARTS:
       Not(in=a, out=nota);
       Not(in=b, out=notb);
       And(a=a, b=notb, out=aAndNotb);
       And(a=nota, b=b, out=notaAndb);
       Or(a=aAndNotb, b=notaAndb, out=out);
```

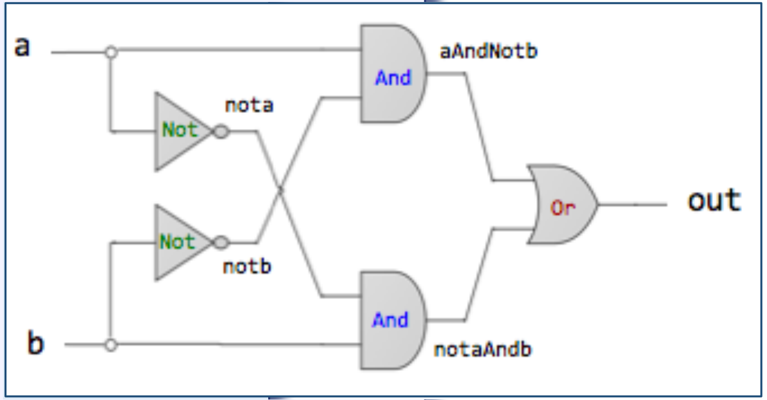
Chip Interface

Chip Implementation

Other Xor implementations are possible!

HDL Some Comments

```
/** Xor gate: out = (a And Not(b)) Or (Not(a) And b)) */


CHIP Xor {
 IN a, b;
 OUT out;
 PARTS:
  Not (in=a, out=nota);
  Not (in=b, out=notb);
  And (a=a, b=notb, out=aAndNotb);
  And (a=nota, b=b, out=notaAndb);
  Or (a=aAndNotb, b=notaAndb, out=out);
```



- HDL is a functional/declarative language
- The order of HDL statements is insignificant
- Before using a chip part, you must know its interface. For example: `Not(in= ,out=)`, `And(a= ,b= ,out= )`, `Or(a= ,b= ,out= )`

Interactive Chip Testing Demo

Demo

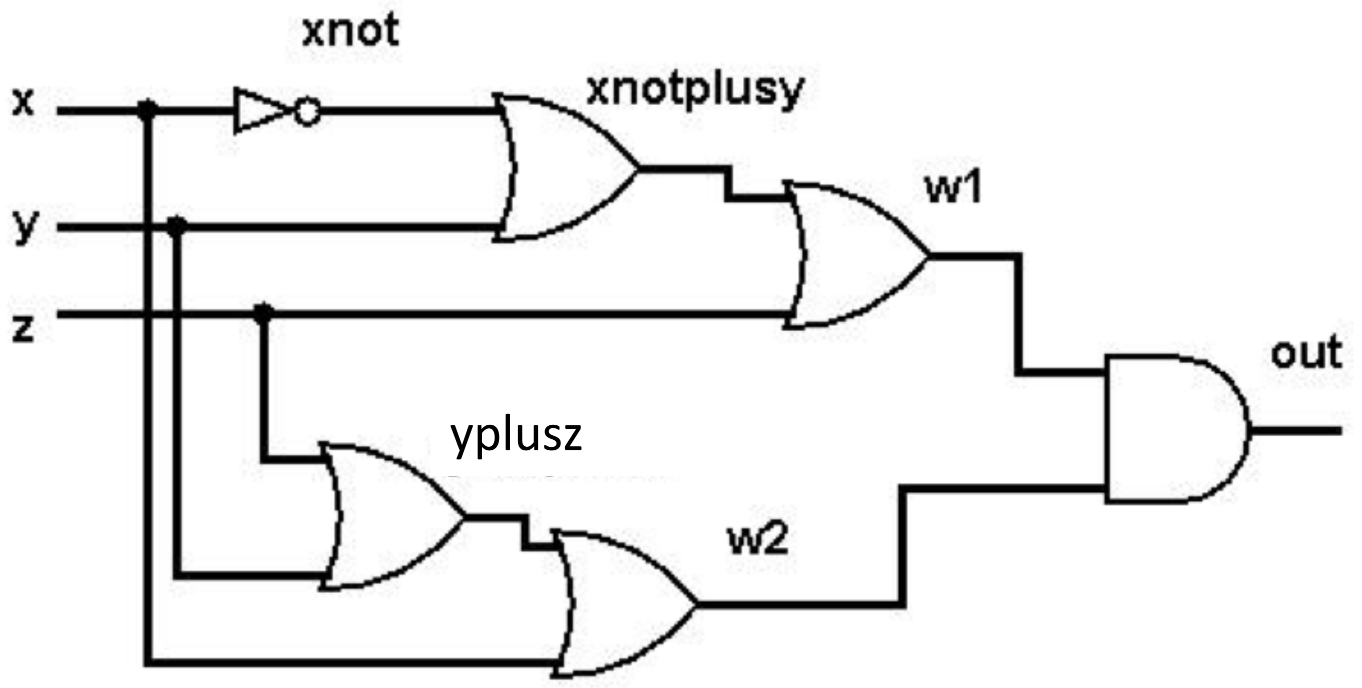
Hardware Simulator

Interactive Testing

`03/Xor.hdl`

Class Quiz

Class Quiz



chip interface

```
/** Class Quiz */
CHIP Quiz {
   IN x, y, z;
   OUT out;
   PARTS:
    //Write chip implementation code
}
```

```
Not(in= ,out=)
And(a= ,b= ,out= )
Or(a= ,b= ,out= )
```

What is Script Based Chip Testing

Script-based Simulation

Simulation Options:

- Interactive
- Script Based: A test script is a series of commands to the simulator
    - With/without output file
    - With/without compare file

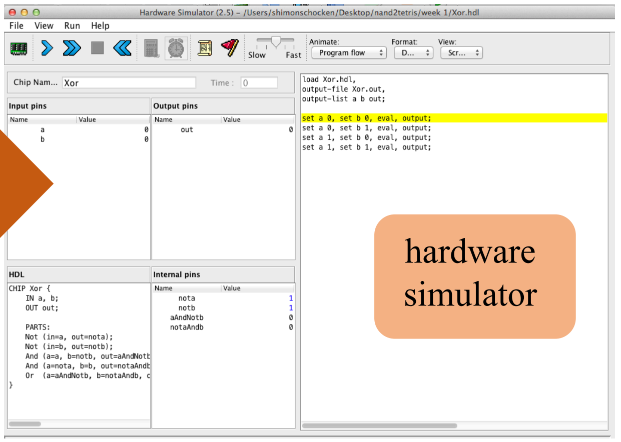
Xor.hdl

```
CHIP Xor {
  IN a, b;
  OUT out;
  PARTS:
    Not(in=a, out=nota);
    Not(in=b, out=notb);
    And(a=a, b=notb, out=aAndNotb);
    And(a=nota, b=b, out=notaAndb);
    Or(a=aAndNotb, b=notaAndb, out=out);
```

Xor.tst

```
load Xor.hdl;
set a 0, set b 0, eval;
set a 0, set b 1, eval;
set a 1, set b 0, eval;
set a 1, set b 1, eval;
```

simulate



Name	Value
a	0
b	0

Name	Value
out	0

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

    PARTS:
    Not (in=a, out=nota);
    Not (in=b, out=notb);
    And (a=a, b=notb, out=aAndNott
    And (a=nota, b=b, out=notaAndt
    Or  (a=aAndNotb, b=notaAndb, c
}
```

Name	Value
nota	1
notb	1
aAndNotb	0
notaAndb	0

```
load Xor.hdl,
output-file Xor.out,
output-list a b out;

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;
```

hardware simulator

Script-base Simulation with an Output File

Xor.hdl

Tested chip

```
CHIP Xor {
  IN a, b;
  OUT out;
  PARTS:
    Not(in=a, out=nota);
    Not(in=b, out=notb);
    And(a=a, b=notb, out=aAndNotb);
    And(a=nota, b=b, out=notaAndb);
    Or(a=aAndNotb, b=notaAndb, out=out);
```

Xor.tst

test script

```
Load Xor.hdl,
output-file Xor.out,
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;
```

Xor.out

```
|a  | b | out|
|0  | 0 | 0  |
|0  | 1 | 1  |
|1  | 0 | 1  |
|1  | 1 | 0  |
```

output File, created by the test script as a side-effect of the simulation process

The logic of a typical test script

- Initialize by loading an HDL file
- Can create an empty output file
- List the names of the pins whose values will be written to the output file
- **Set-eval-output** and repeat

Script-base Simulation with Compare File

Xor.hdl

Tested chip

```
CHIP Xor {
  IN a, b;
  OUT out;
  PARTS:
    Not(in=a, out=nota);
    Not(in=b, out=notb);
    And(a=a, b=notb, out=aAndNotb);
    And(a=nota, b=b, out=notaAndb);
    Or(a=aAndNotb, b=notaAndb, out=out);
```

Xor.tst

test script

```
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;
```

Simulation-with-compare-file logic

- If the script specifies a compare file, when each output command is executed, the outputted line is compared to the corresponding line in the compare file
- If the two lines are not the same, the simulator throws a comparison error

Xor.out

```
|  a  |  b  |out|
|  0  |  0  |  0  |
|  0  |  1  |  1  |
|  1  |  0  |  1  |
|  1  |  1  |  0  |
```

Xor.cmp

```
|  a |  b  |out|
|  0  |  0  |  0  |
|  0  |  1  |  1  |
|  1  |  0  |  1  |
|  1  |  1  |  0  |
```

Script Based Chip Testing Demo

Demo

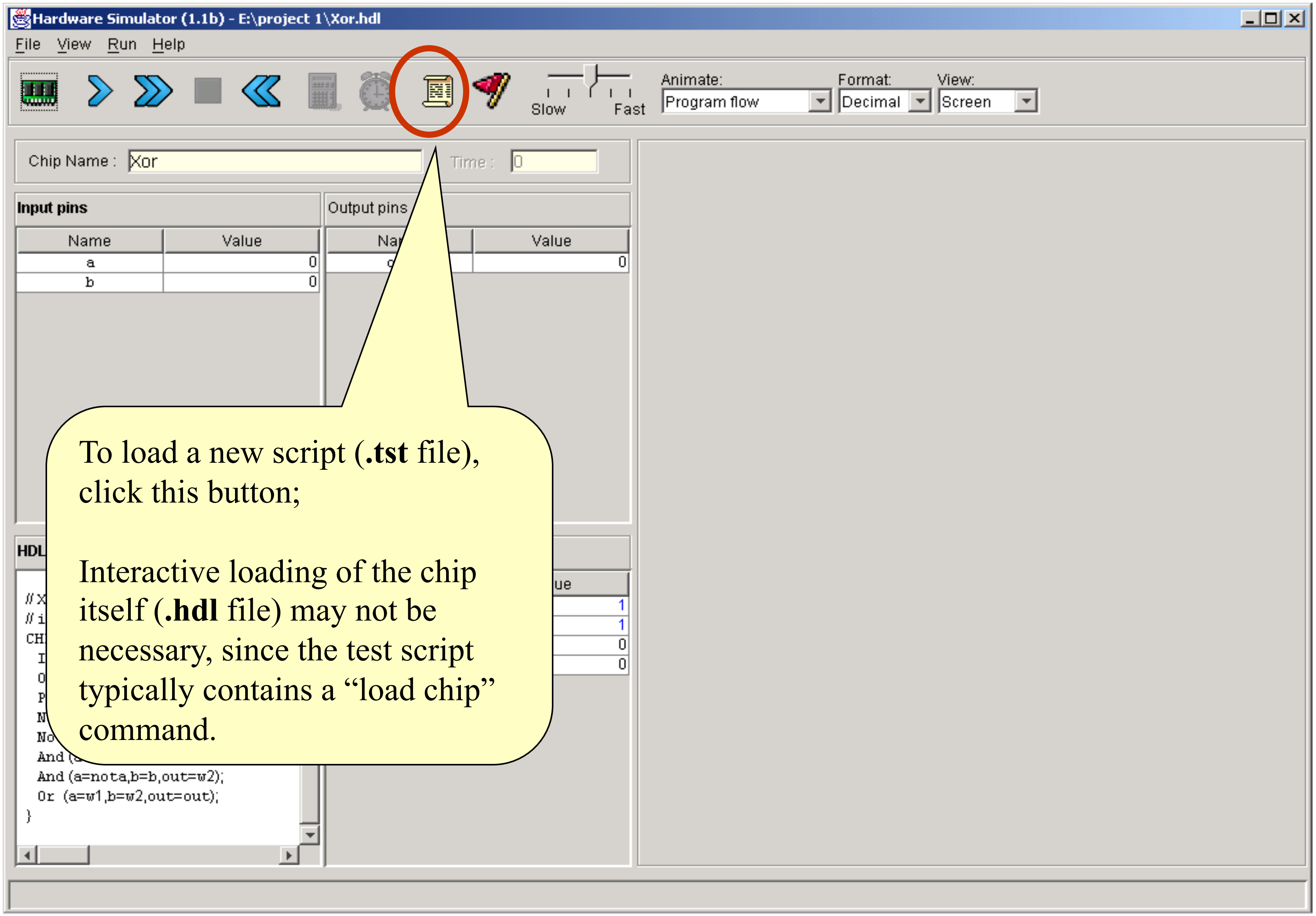
Hardware Simulator

Script Based Testing

`03/Xor.tst`

Script Based Chip Testing on Hardware Simulator

Loading A Script



Hardware Simulator (1.1b) - E:\project 1\Xor.hdl

Animate: Program flow Format: Decimal View: Screen

Chip Name : Xor Time : 0

Input pins

Name	Value
a	0
b	0

Output pins

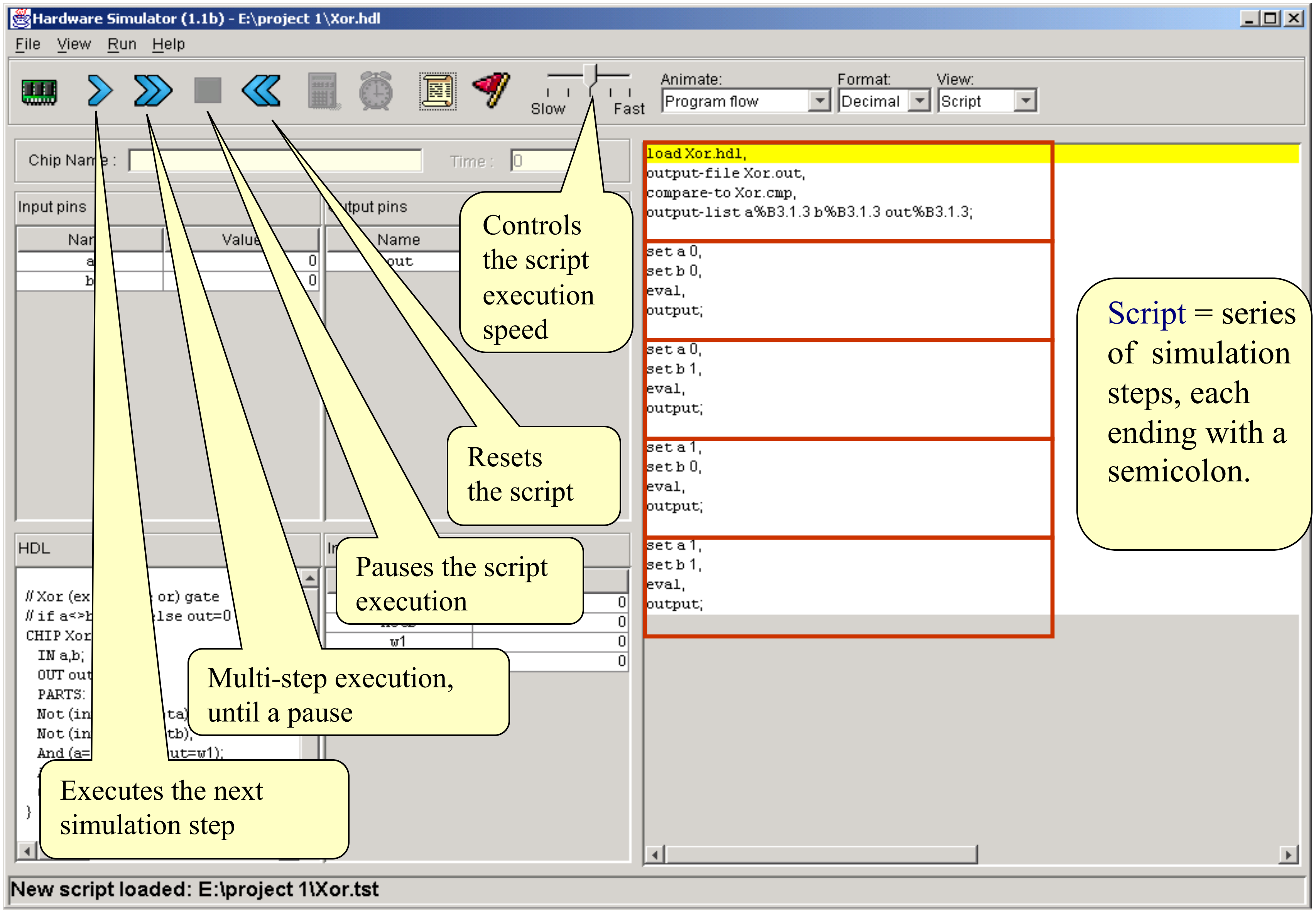
Name	Value
out	0

```
  And (a=nota,b=b,out=w2);
  Or  (a=w1,b=w2,out=out);
}
```

To load a new script (**.tst** file), click this button;

Interactive loading of the chip itself (**.hdl** file) may not be necessary, since the test script typically contains a “load chip” command.

Script Controls



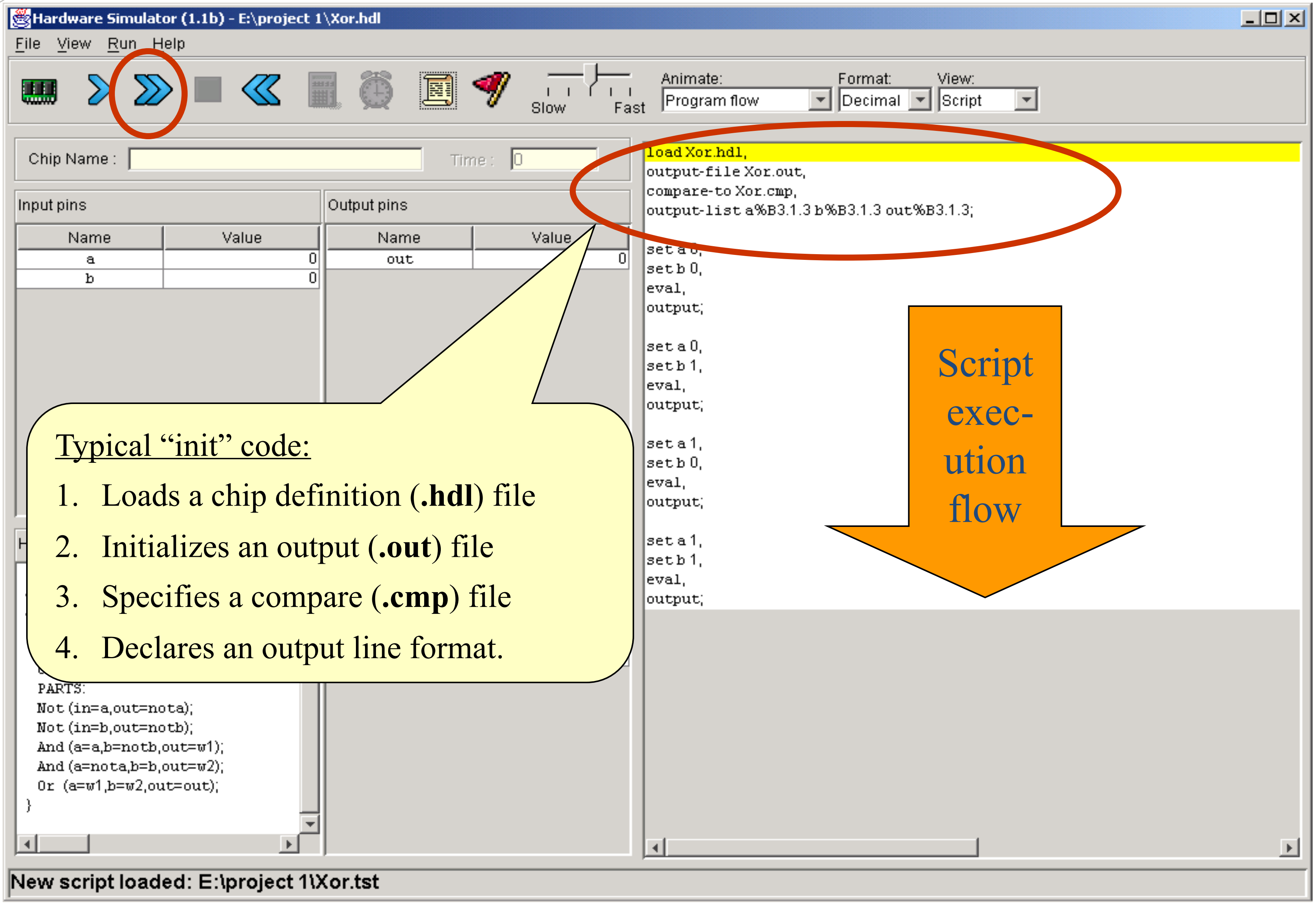
Hardware Simulator (1.1b) - E:\project 1\Xor.hdl
File View Run Help
Slow Fast
Animate: Program flow
Format: Decimal
View: Script
Chip Name :
Time : 0
Input pins
Output pins
HDL
Controls the script execution speed
Resets the script
Pauses the script execution
Multi-step execution, until a pause
Executes the next simulation step
Script = series of simulation steps, each ending with a semicolon.

Name	Value
a	0
b	0

```
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;
```

New script loaded: E:\project 1\Xor.tst

Running A Script



Name	Value
a	0
b	0

Name	Value
out	0

```
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;
```

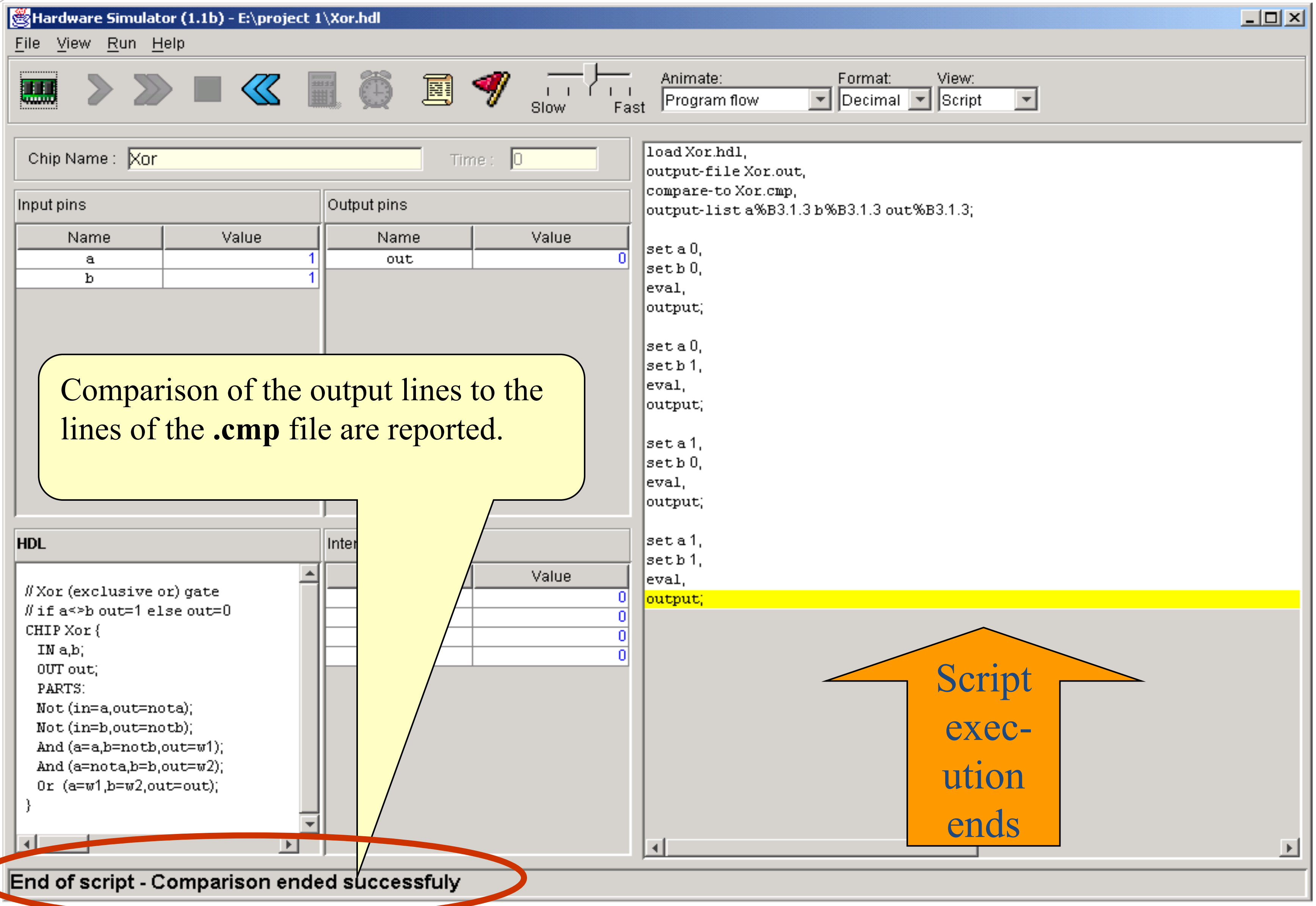
```
PARTS:
Not (in=a,out=nota);
Not (in=b,out=notb);
And (a=a,b=notb,out=w1);
And (a=nota,b=b,out=w2);
Or  (a=w1,b=w2,out=out);
}
```

New script loaded: E:\project 1\Xor.tst

Typical "init" code:

1. Loads a chip definition (**.hdl**) file
2. Initializes an output (**.out**) file
3. Specifies a compare (**.cmp**) file
4. Declares an output line format.

Running A Script



Hardware Simulator (1.1b) - E:\project 1\Xor.hdl

Chip Name : Xor Time : 0

Input pins

Name	Value
a	1
b	1

Output pins

Name	Value
out	0

HDL

```
// Xor (exclusive or) gate
// if a<>b out=1 else out=0
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);
}
```

Inter

	Value
	0
	0
	0
	0

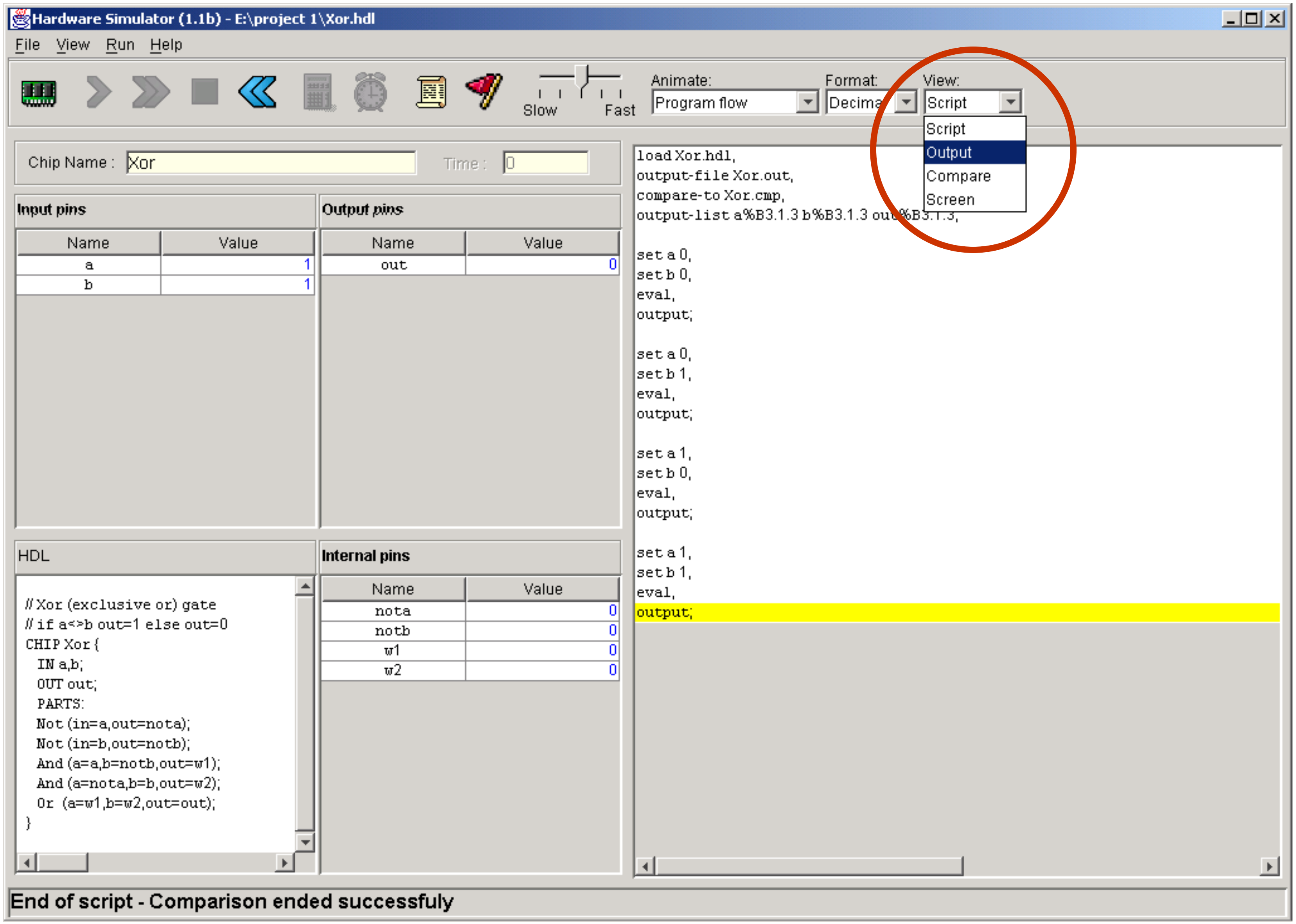
```
load Xor.hdl,
output-file Xor.out,
compare-to Xor.cmp,
output-list a%B3.1.3 b%B3.1.3 out%B3.1.3;

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 successfuly

Comparison of the output lines to the lines of the **.cmp** file are reported.

Script exec-ution ends

Viewing Output And Comparing Files



Chip Name : Xor   Time : 0

Input pins

Name	Value
a	1
b	1

Output pins

Name	Value
out	0

HDL

```
// Xor (exclusive or) gate
// if a<>b out=1 else out=0
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);
}
```

Internal pins

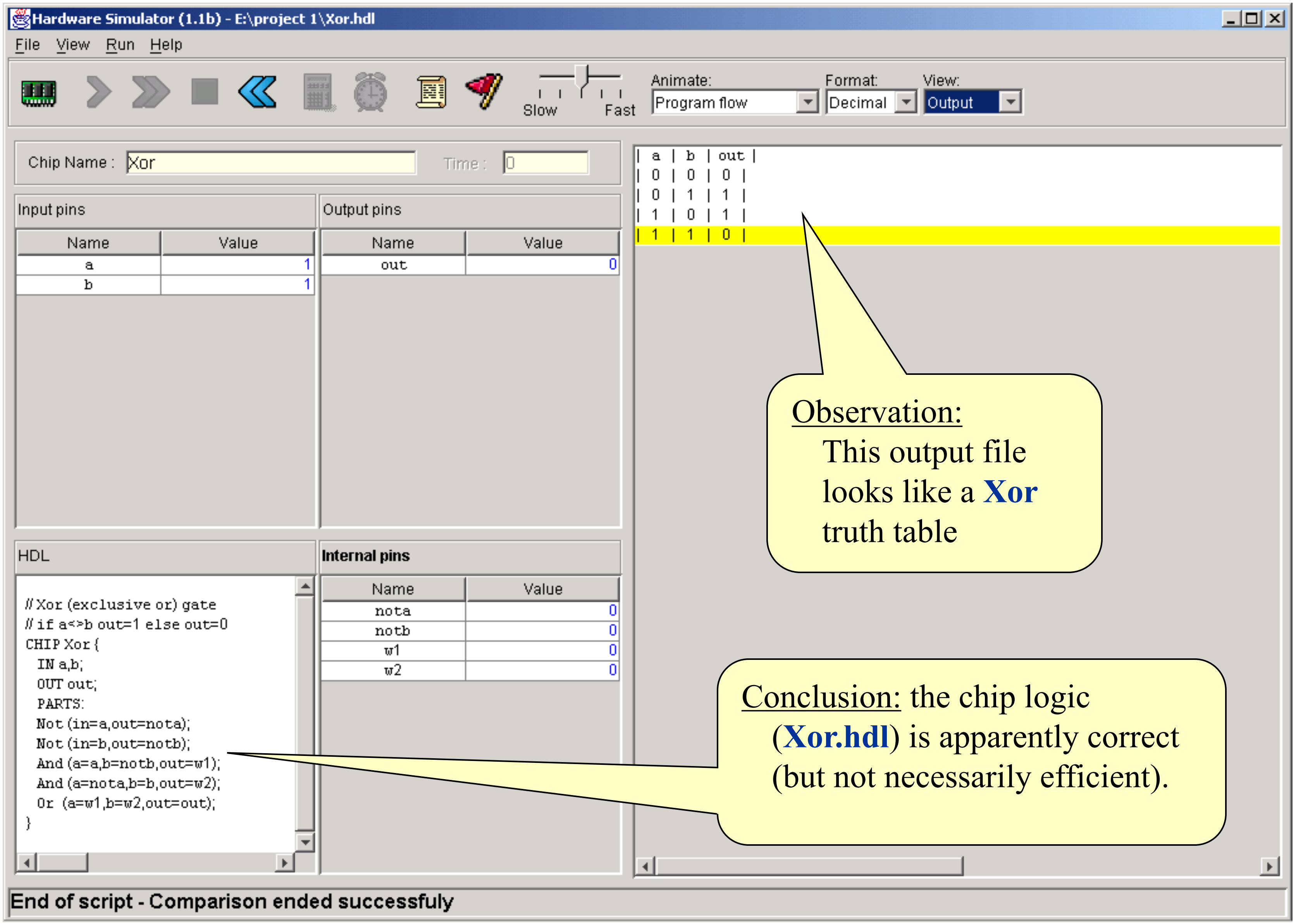
Name	Value
nota	0
notb	0
w1	0
w2	0

```
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 successfuly

Instructor: Muhammad Arif Butt, Ph.D.

Viewing Output And Compare Files



Hardware Simulator (1.1b) - E:\project 1\Xor.hdl

Animate: Program flow
Format: Decimal
View: Output

Chip Name : Xor
Time : 0

Input pins

Name	Value
a	1
b	1

Output pins

Name	Value
out	0

HDL

```
// Xor (exclusive or) gate
// if a<>b out=1 else out=0
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);
}
```

Internal pins

Name	Value
nota	0
notb	0
w1	0
w2	0

```
| a | b | out |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
```

End of script - Comparison ended successfuly

Observation: This output file looks like a **Xor** truth table

Conclusion: the chip logic (**Xor.hdl**) is apparently correct (but not necessarily efficient).

Players Involved in a H/W Construction Project

System Architect:

- Decides which chips are needed, and for each chip the architect creates:
  - ➢ A chip API
  - ➢ A test script
  - ➢ A compare file

Developer:

- The above three files given to the developer provide a convenient specification of
  - ➢ The chip interface (.hdl file)
  - ➢ What the chip is supposed to do (.cmp file)
  - ➢ How to test the chip (.tst file)
- Developer tasks is to implement the chip using these resources

Things To Do

- Perform script based chip testing of the AND, OR, NOT and the XOR chips in the h/w simulator
- No need to write the test script (.tst) and compare files (.cmp). Both these files for all the h/w chips to be designed are available on course bitbucket repository (coal-repo)

https://bitbucket.org/arifpucit/coal-repo/

- Interested students can learn to write test scripts and a tutorial on the Test Description Language is also available on course web site at

http://www.arifbutt.me

Coming to office hours does NOT mean you are academically week!