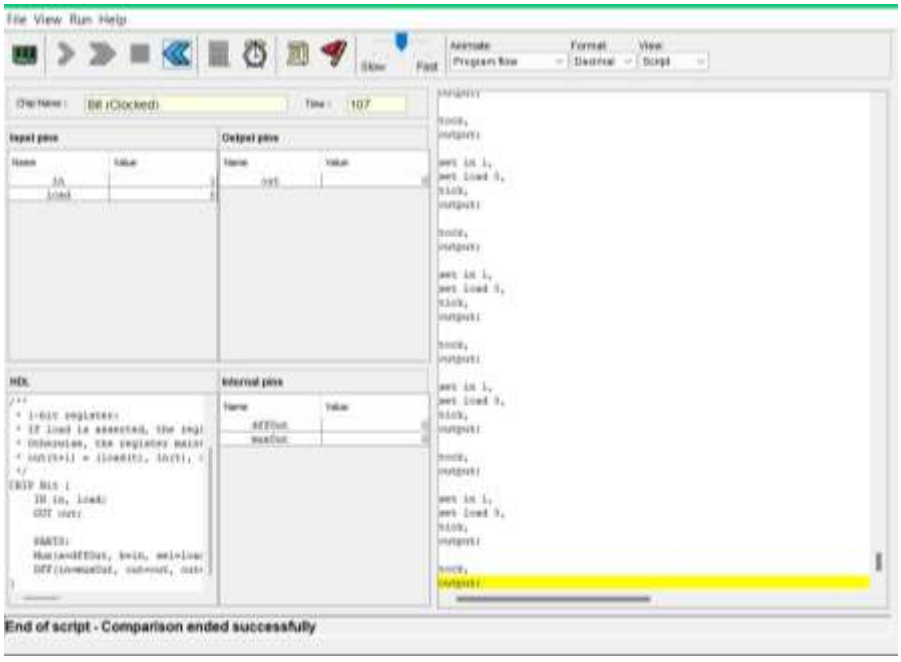
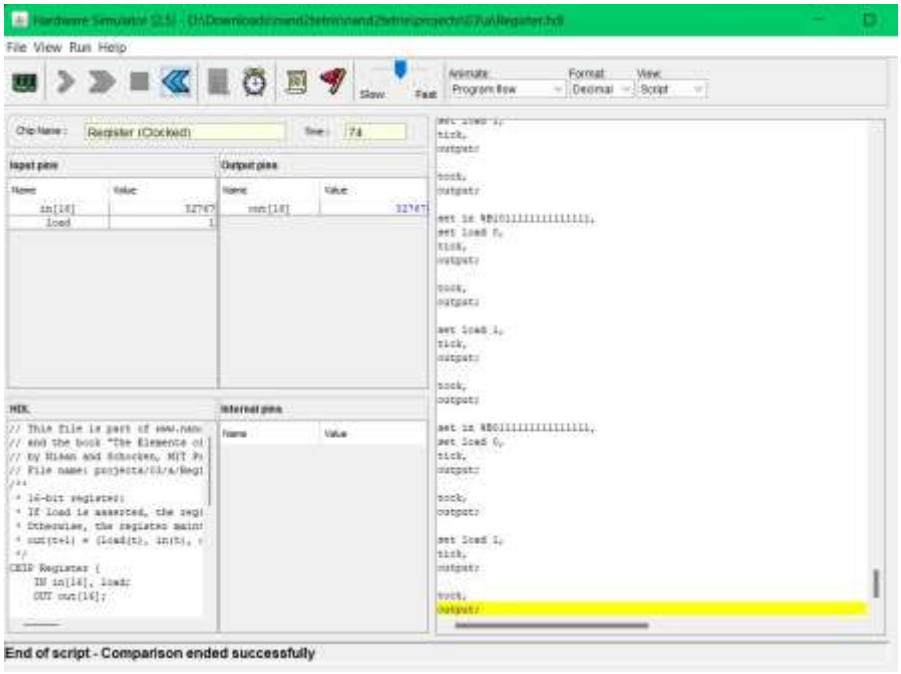
Task 4.1P Implement 16-bit register in HDL

- To build this 16-bit register, I used the Nand Tetris hardware simulator's HDL programming.
- 16 single-bit register arrays are formed to produce this.
- After that, joining them to create the 16-bit register.
- There is a DFF and a Mux in the 1-bit register.
- The chip interface is made up of an output pin that outputs the cell's current state, an input pin that carries a data bit, and a load bit that allows the cell to be written to.
- After that, in order to fully implement the 16-bit register, I had to construct 16 arrays of 1-bit registers and connect them to one another in the 16-bit register HDL file.
- Using bus syntax, the in [0] and out [0] bits are connected, and this connection process is repeated up to the 16th bit. To put it another way, bus syntax links the input bit in [0] to the output bit out [0], and this connection is maintained progressively up to the 16th bit.
- After that, launch the Hardware Simulator and load the test file and 16-bit register HDL file.
- The comparison concluded satisfactorily, demonstrating the successful creation of a 16-bit register.




File View Run Help
Slow
Fast
Animate:
Program flow
Format:
Decimal
View:
Script
Chip name:
Register (Clocked)
Time:
74
Input pins
Name
Value
in[16]
32767
load
1
Output pins
Name
Value
out[16]
32767
HDL
Internal pins
Name
Value
// This file is part of www.nan
// and the book "The Elements of
// by Nisan and Schocken, MIT Pr
// File name: projects/01/a/Regi
/**
* 16-bit register:
* If load is asserted, the regi
* Otherwise, the register main
* out(t+1) = (load(t), in(t), o
*/
CHIP Register {
IN in[16], load;
OUT out[16];
set in %B1111111111111111,
set load 0,
tick,
output;
tock,
output;
set load 1,
tick,
output;
tock,
output;
set in %B0111111111111111,
set load 0,
tick,
output;
tock,
output;
set load 1,
tick,
output;
tock,
output;
End of script - Comparison ended successfully