

The program counter, also known as the instruction pointer or simply PC, is a fundamental component of a computer's central processing unit (CPU). It is a special register that keeps track of the memory address of the next instruction to be executed in a program. The Hardware Description Language (HDL) code presented here implements a 16-bit counter chip with load and reset control bits. The behavior of the chip is defined by the state transitions and control conditions outlined below.

- **Reset Condition:** If the reset input is set to 1, the output of the chip is immediately set to 0.
- **Load Condition:** If the load input is set to 1, the output of the chip is loaded with the value present at the input.
- **Increment Condition:** If the inc input is set to 1, the output of the chip is incremented by 1.
- **Default Condition:** If none of the above conditions apply, the output of the chip remains the same as its previous output.

There were totally 3 types of components used;

- One 16-bit register
  - **Functionality:** Stores the current value of the counter.
  - **Load:** Set to true when the load control bit is 1.
  - **Outputs:** Connected to the output of the chip.
- One 16-bit incrementor
  - **Functionality:** Increments the value of the register when the inc control bit is 1.
  - **Input:** Connected to the output of the register.
  - **Output:** Connected to one of the inputs of a 16-bit multiplexer.
- Three 16-bit multiplexer
  - **Functionality:** Used to select the appropriate value to be output by the chip.
  - **Control Inputs:** Determine the selection based on control bits (load, inc, reset).
  - **Connections:**
    - **First Multiplexer:** Selects between the output of the incrementor and the output of the register, based on the inc control bit.
    - **Second Multiplexer:** Selects between the input of the chip and the output of the first multiplexer, based on the load control bit.
    - **Third Multiplexer:** Selects between 0 and the output of the second multiplexer, based on the reset control bit. The output is connected to the input of the register, completing the feedback loop.

In conclusion, the 16-bit counter chip HDL code provides a versatile and configurable counter with load and reset functionalities. The combination of a register, incrementor, and multiplexers allows for efficient control and manipulation of the counter's state.

The screenshot shows the Hardware Simulator (2.5) interface. The title bar indicates the file path: D:\Downloads\nand2tetris\nand2tetris\projects\03\PC\PC.hdl. The main window is divided into several sections:

- File View Run Help**: A menu bar at the top.
- Simulation Controls**: A toolbar with icons for simulation (play, pause, stop, reset), a clock icon, and buttons for 'Slow' and 'Fast' simulation speeds. There are also dropdowns for 'Animate' (Program flow), 'Format' (Decimal), and 'View' (Script).
- Chip Name**: A text field showing 'PC (Clocked)' and a 'Time' field showing '15'.
- Input pins**: A table with columns 'Name' and 'Value'.
 

Name	Value
in[16]	22222
load	0
inc	0
reset	1
- Output pins**: A table with columns 'Name' and 'Value'.
 

Name	Value
out[16]	0
- HDL**: A text area containing the HDL code for the PC chip. The code includes comments and logic for a 16-bit counter with reset, load, and increment functionality.
 

```
// 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/03/PC/PC.hdl
/**
 * A 16-bit counter with reset,
 * load, and increment.
 * If reset is 1, the output is set to 0.
 * If load is 1, the output is set to the value of in[16].
 * If inc is 1, the output is incremented by 1.
 * Otherwise, the output remains the same.
 */
CHIP PC {
    IN in[16], load, inc, reset;
    OUT out[16];
    LOCAL regOut[16], incOut[16], loadMuxOut[16], incMuxOut[16];
    LOCAL resetMuxOut[16];

    resetMuxOut[16] = 0;
    regOut[16] = 0;
    incOut[16] = 1;
    loadMuxOut[16] = 0;
    incMuxOut[16] = 0;

    set in 22222,
    set reset 1,
    set load 1,
    tick,
    output;

    set load 0,
    set inc 1,
    tick,
    output;

    set in 22222,
    set reset 1,
    set inc 0,
    tick,
    output;

    tick,
    output;
```
- Internal pins**: A table with columns 'Name' and 'Value'.
 

Name	Value
resetMuxOut[16]	0
regOut[16]	0
incOut[16]	1
loadMuxOut[16]	0
incMuxOut[16]	0

At the bottom of the window, a status bar reads: "End of script - Comparison ended successfully".