Assignment 4

Assignment 4

Workshop – Hack CPU

What you will be implementing:
Pt.1: Data Memory Chip (Memory.hdl)
Pt.2: CPU Chip (CPU.hdl)
Pt.3: Computer Chip (Computer.hdl)

What you need to know:

- HACK Computer Architecture [L07]
- More details can be sourced from Ch 5 of the Textbook.

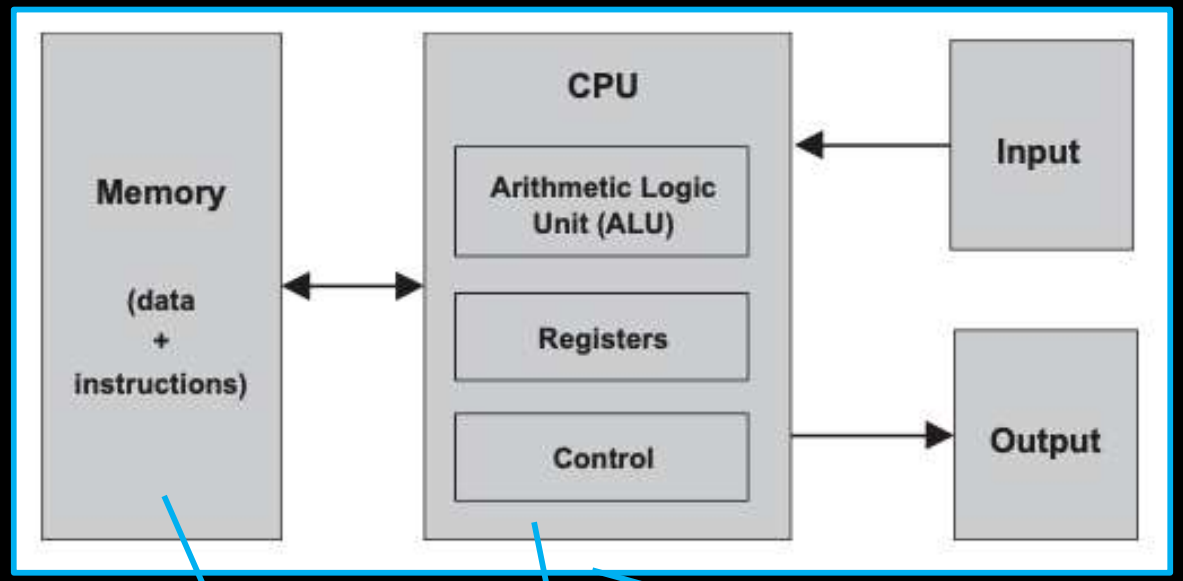
Assignment 4 details found at:
Source 1: CS MyUni -> Modules -> Assignment 4 (Assignments section)
Source 2: CS MyUni -> Modules -> Workshop – HACK CPU (Week 6 section)

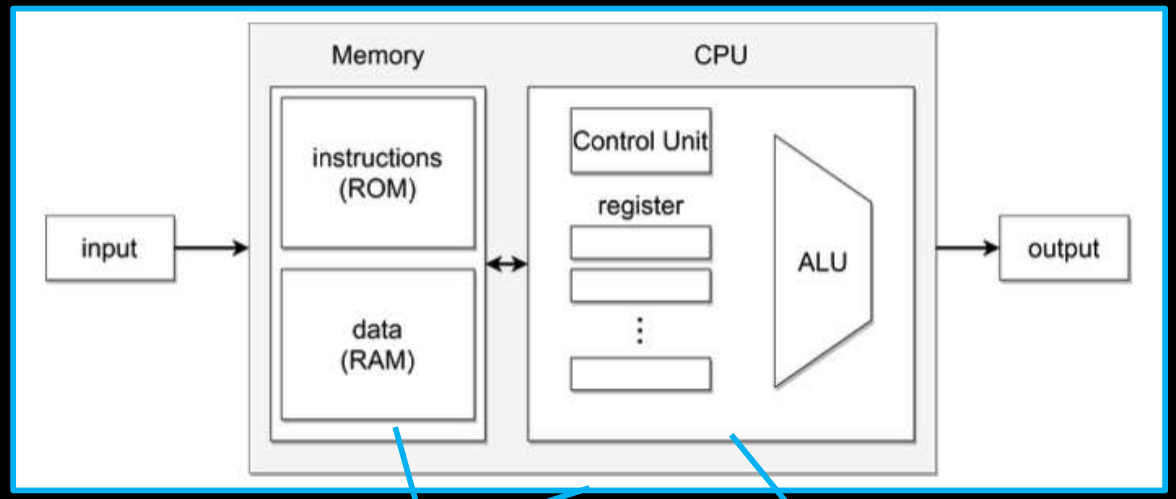
It is advised to also read and understand the /* */ comment sections in the HDL files as they express the chip logic.

Ensure all chips from Assignments 1, 2, & 3 are working!!!

Assignment 4

Von Neumann System Architecture





Pt.1: Memory.hdl

Pt.2: CPU.hdl

Pt.3: Computer.hdl

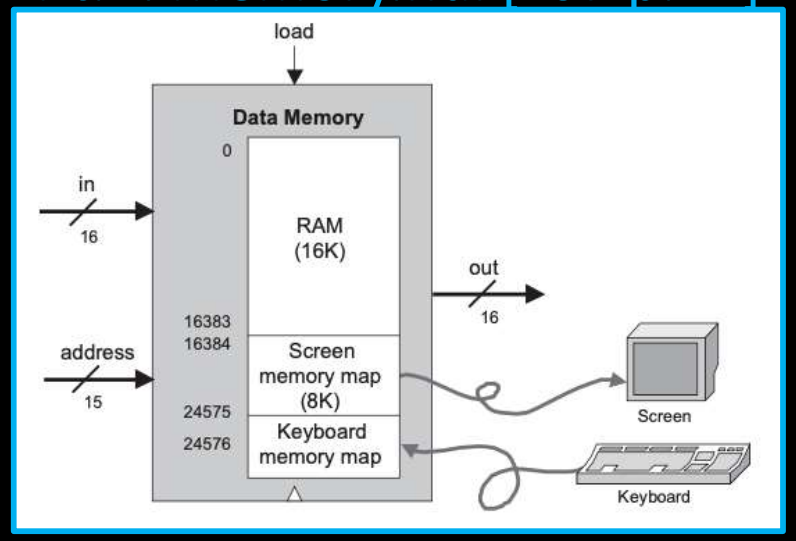
Pt.1: Memory.hdl

Pt.2: CPU.hdl

Assignment 4

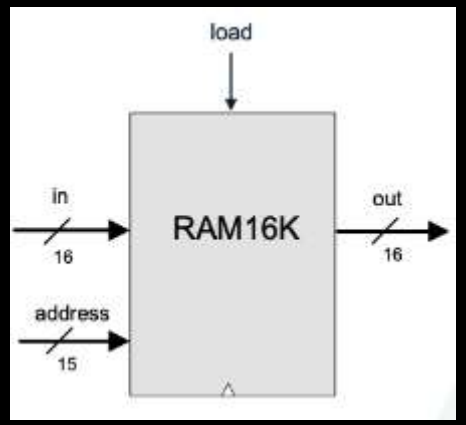
Part 1: Data Memory

Pt.1: Memory.hdl [L07 p.11]



=

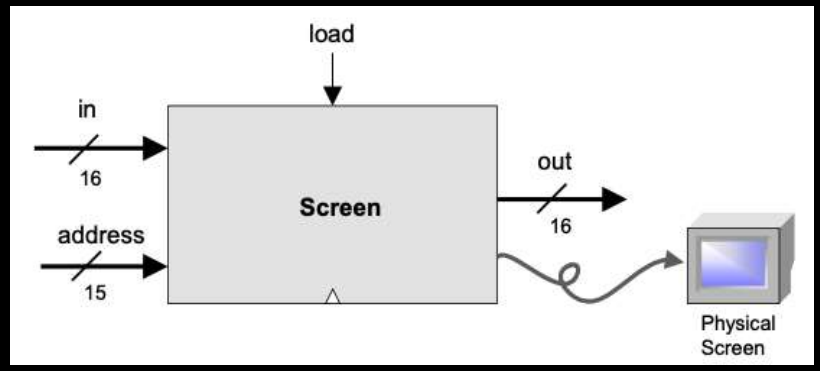
Data Memory [L07 p.12]



From Assignment 3.
MyUni assignment page description hints that more than 16K may be required.

+

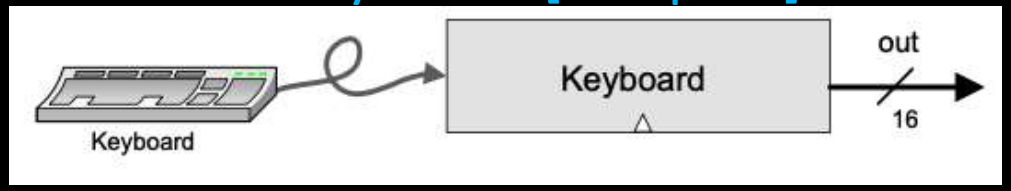
Screen [L07 pp.13-14]



Already implemented.
More details on MyUni pages.
Lecture 7 demonstrates Hardware Simulator GUI testing.

+

Keyboard [L07 p.15]

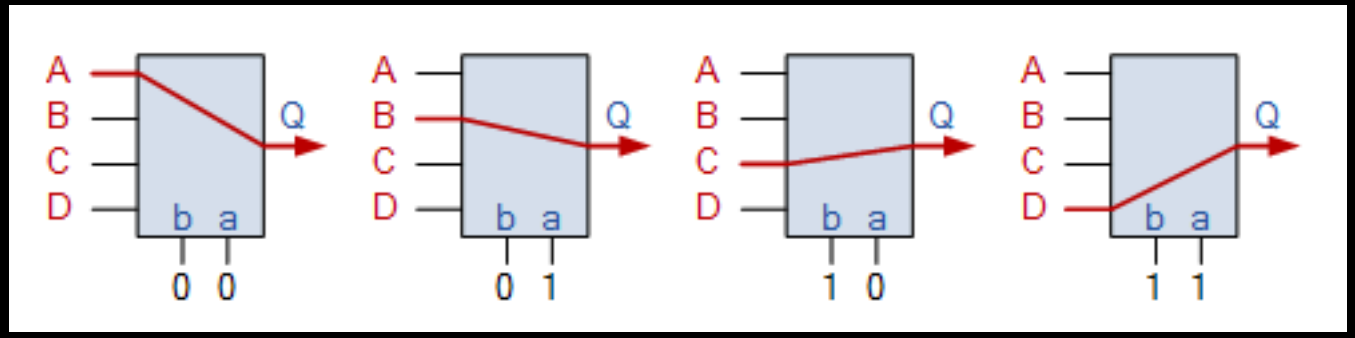


Already implemented. More details on MyUni pages.
Lecture 7 demonstrates Hardware Simulator GUI testing.

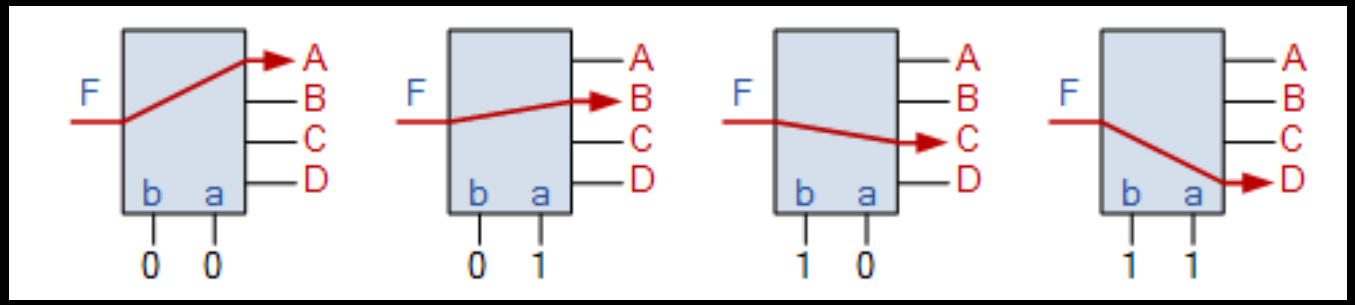
+

Additional chips that provide selection logic.

Mux Input Line Selection



DMux Output Line Selection



The terminology used to “store” the select input (sel) is another way of expressing that sel is used to determine what input is let through (for Mux) or which output is used (for DMux).

So, we would need implementations to use sel and NOT(sel).

Assignment 4

Part 2: CPU

Pt.2: CPU.hdl [L07 pp.16-19]

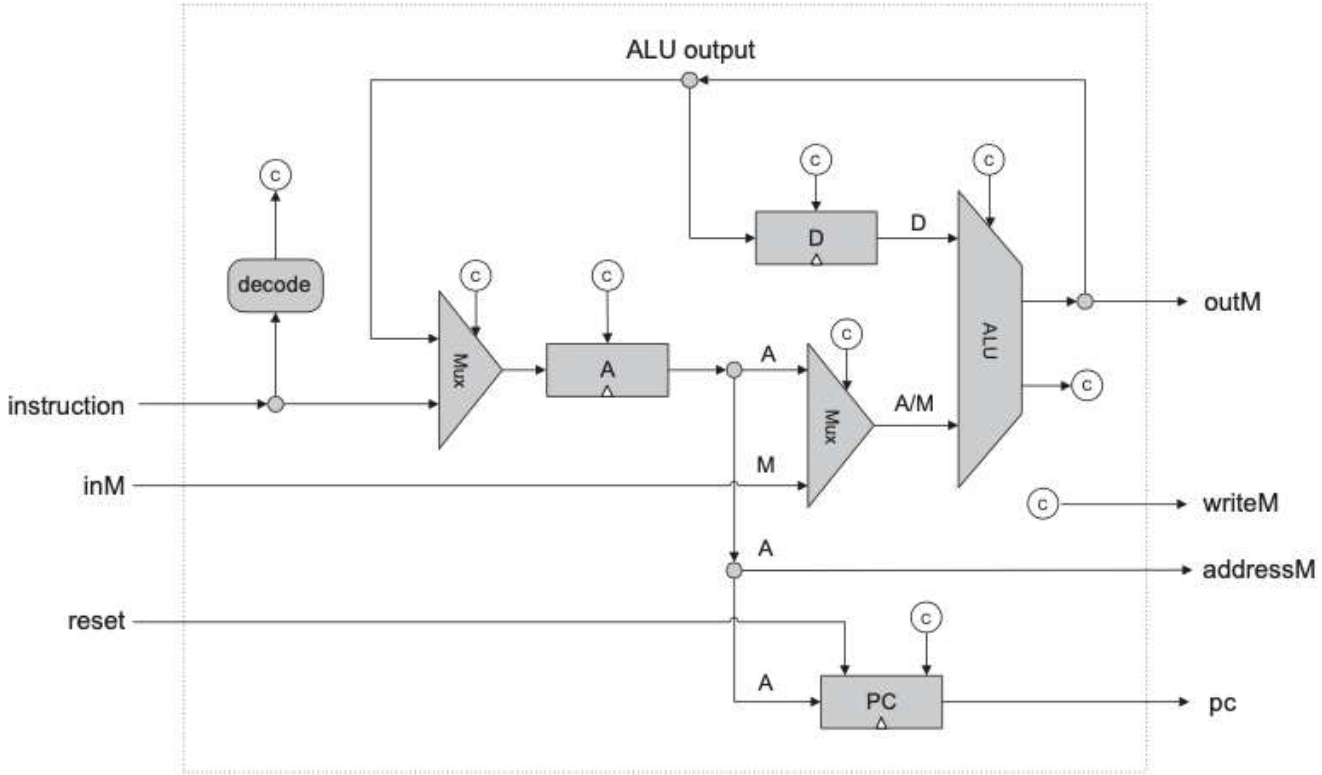
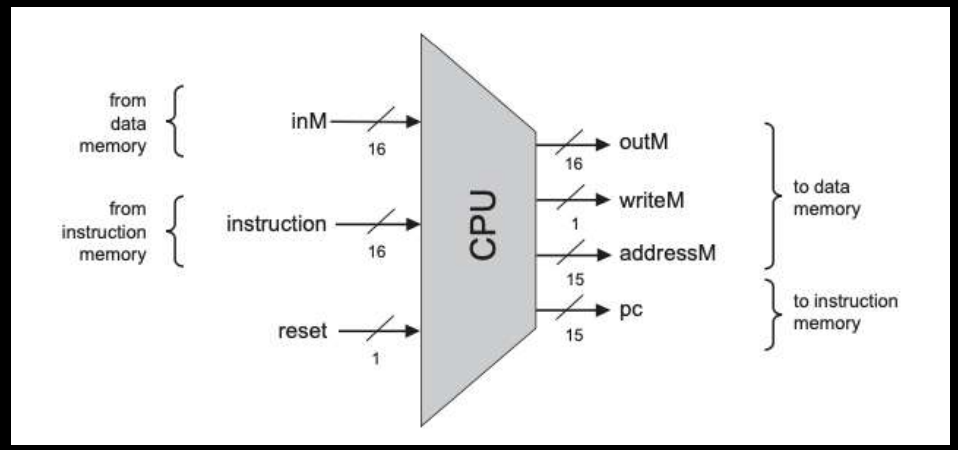


Figure 5.9 Proposed CPU implementation. The diagram shows only *data* and *address paths*, namely, wires that carry data and addresses from one place to another. The diagram does not show the CPU's *control logic*, except for inputs and outputs of control bits, labeled with a circled "c". Thus it should be viewed as an incomplete chip diagram.

From Textbook Ch. 5 p.94

Uses many chips implemented from all previous Assignments.

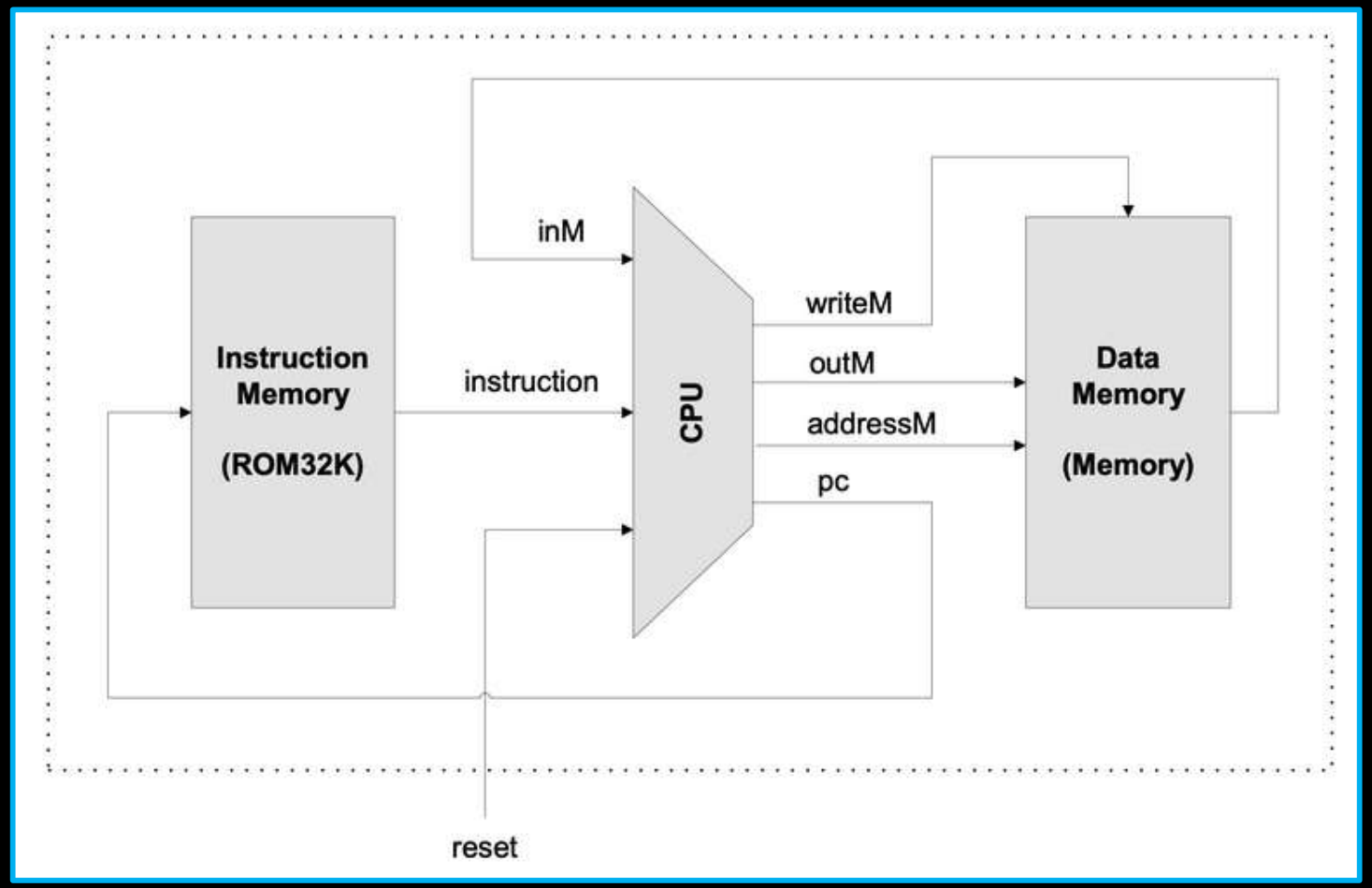
More details on MyUni Assignment pages!



From Textbook Ch. 5 p.88

Part 3: Hack Computer

Pt.3: Computer.hdl [L07 p.9]



From Textbook Ch. 5 p.97

ROM32K already implemented.
So can implement as seen on Figure.

More details on MyUni Assignment pages!

Testing

HardwareSimulator.sh GUI

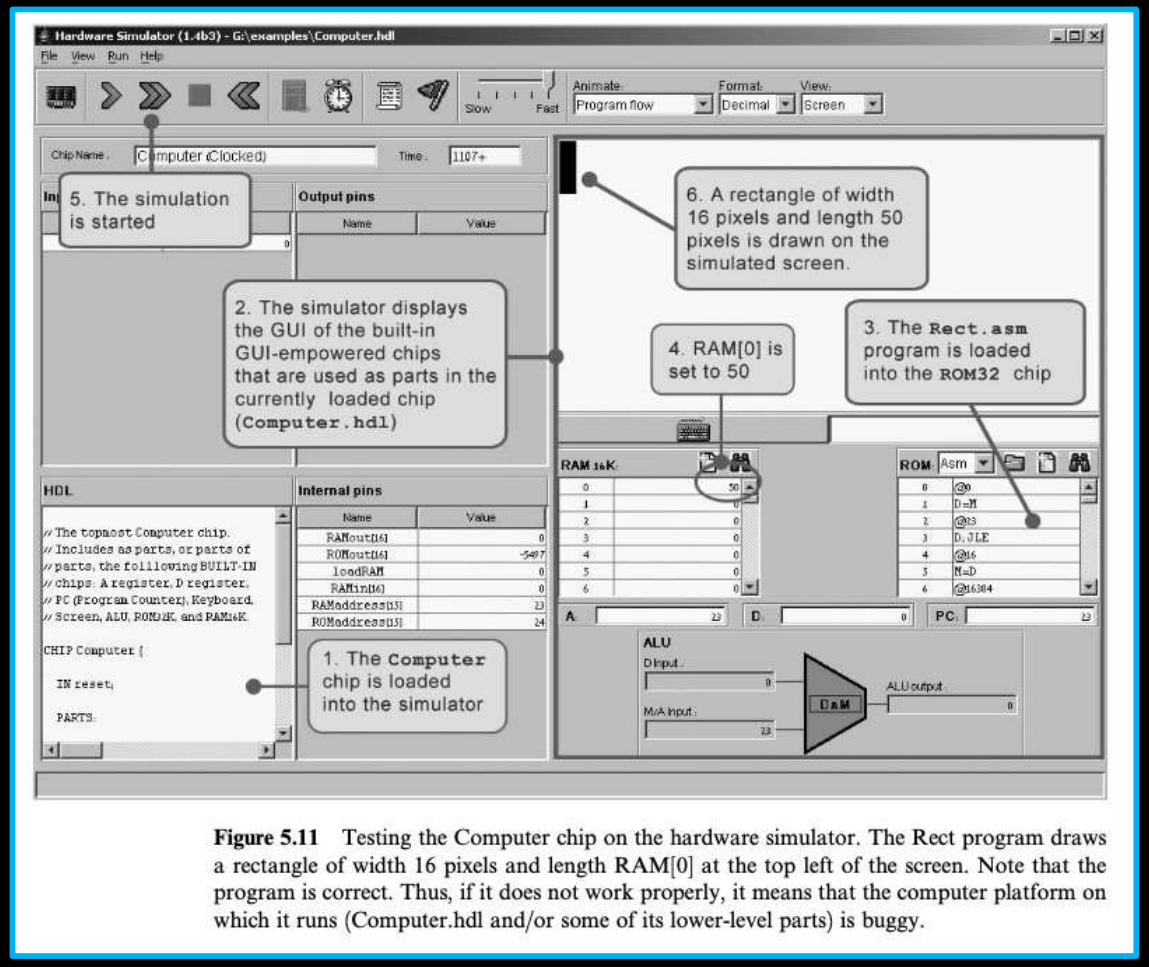
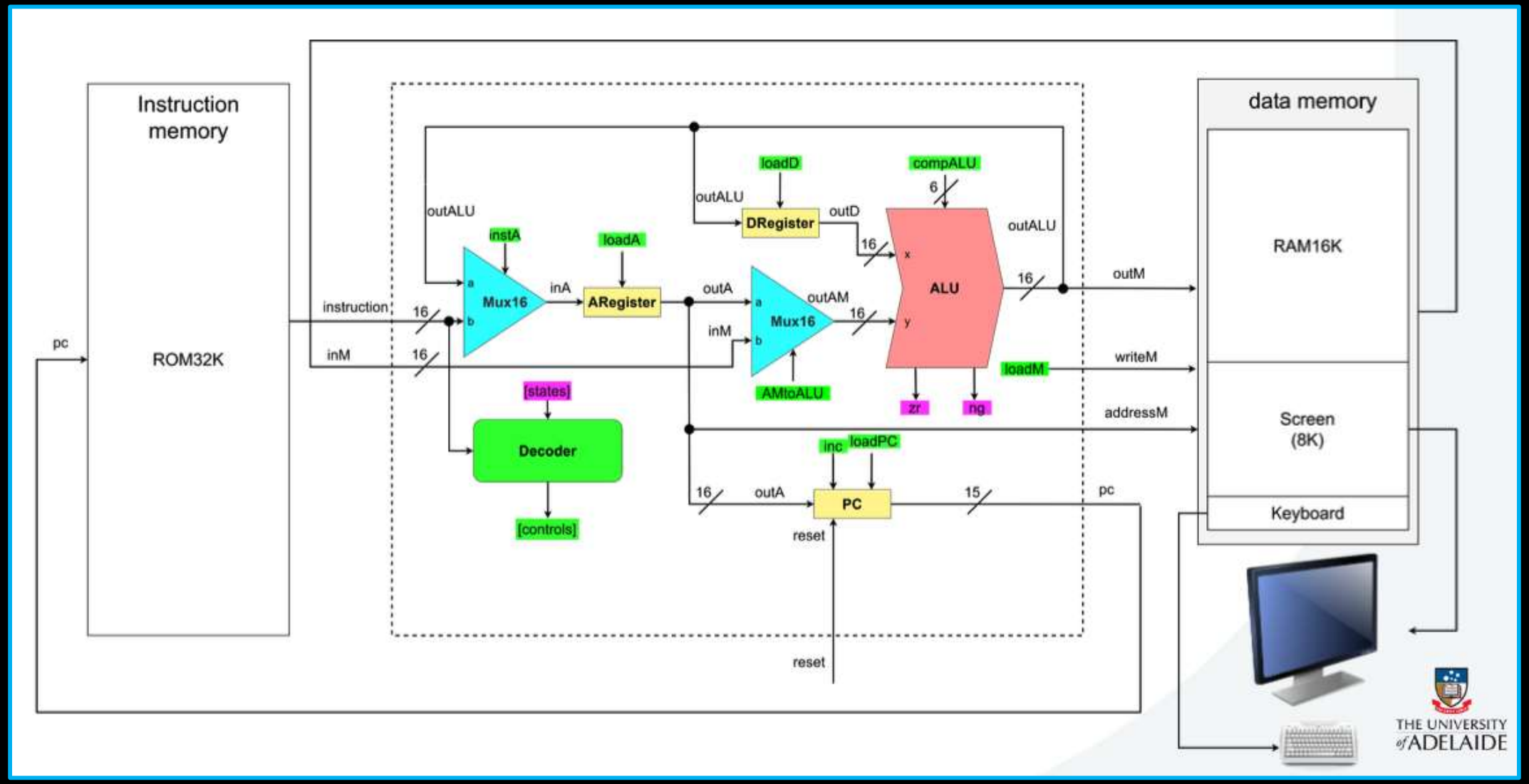


Figure 5.11 Testing the Computer chip on the hardware simulator. The Rect program draws a rectangle of width 16 pixels and length RAM[0] at the top left of the screen. Note that the program is correct. Thus, if it does not work properly, it means that the computer platform on which it runs (Computer.hdl and/or some of its lower-level parts) is buggy.

From Textbook Ch. 5 p.101

Details on what to expect from testing in HardwareSimulator can be found in Textbook Ch.5 & Lecture 7 Notes.

Complete Implementation



Hack Computer Architecture (from Lecture 7 p.4 & Lecture 5).