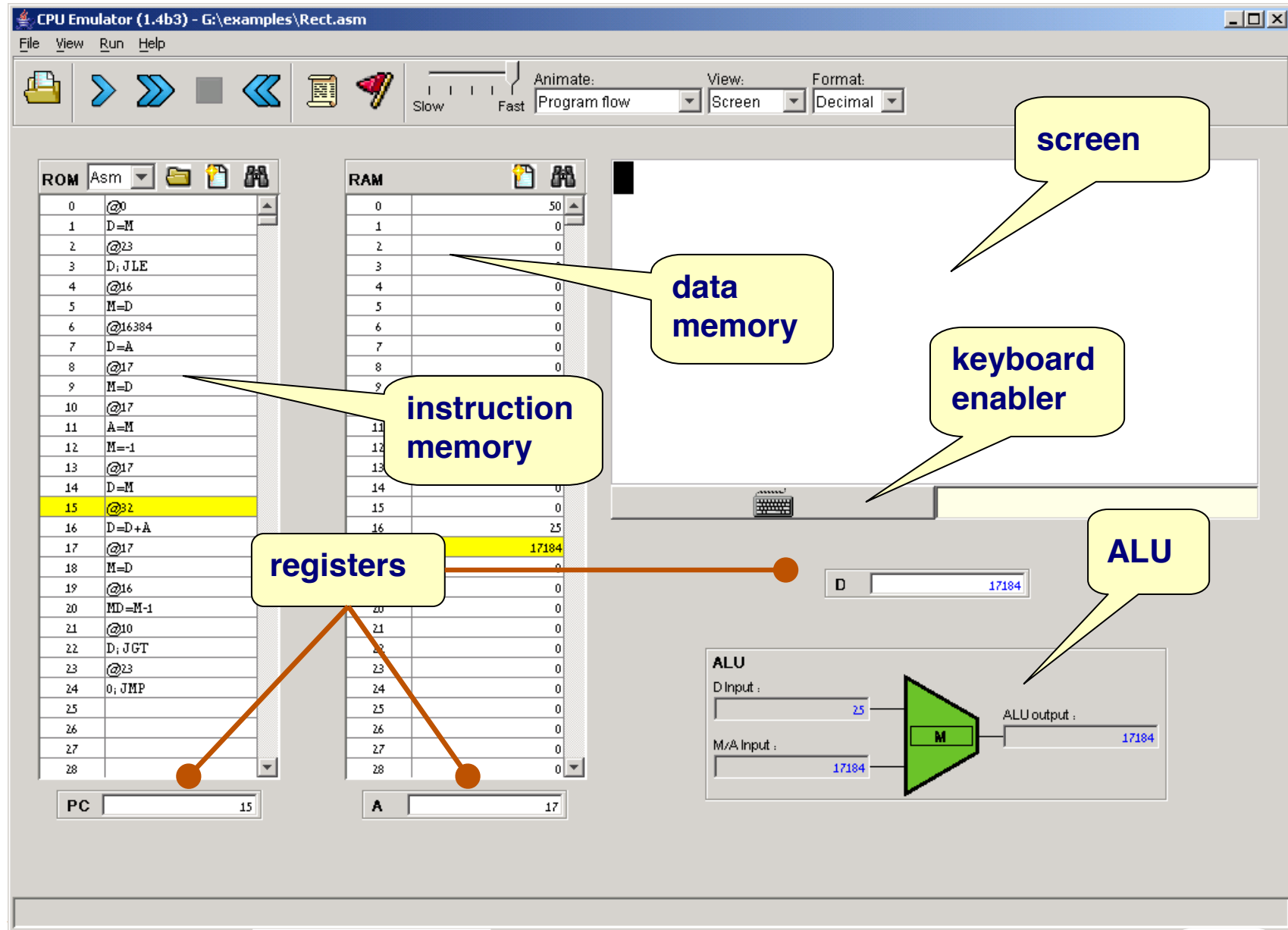


The Hack Computer Platform



Instruction memory

The screenshot displays the CPU Emulator (1.4b3) interface. The main window shows the instruction memory (ROM) and RAM. The instruction memory is currently in symbolic notation (Asm). The instruction at address 15, `@32`, is highlighted in yellow. The program counter (PC) is set to 15. The ALU output is 17184.

Annotations:

- The loaded code can be viewed either in binary, or in symbolic notation (present view)
- Instruction memory (32K): Holds a machine language program
- Next instruction is highlighted
- Program counter (PC) (16-bit): Selects the next instruction.

ROM	Asm
0	@0
1	D=M
2	@23
3	D; JLE
4	@16
5	M=D
6	@16384
7	D=A
8	@17
9	M=D
10	@17
11	A=M
12	M=-1
13	@17
14	D=M
15	@32
16	D=D+A
17	@17
18	M=D
19	@16
20	MD=M-1
21	@10
22	D; JGT
23	@23
24	0; JMP
25	
26	
27	
28	

RAM	
0	
1	
2	
3	
4	
5	
6	0
7	0
8	
9	
10	
11	
12	
13	
14	
15	0
16	25
17	17184
18	0
19	
20	
21	
22	
23	0
24	0
25	0
26	0
27	0
28	

PC: 15

ALU: D Input: 25, M/A Input: M, ALU output: 17184

Data memory (RAM)

The screenshot shows the CPU Emulator (1.4b3) interface. The title bar reads "CPU Emulator (1.4b3) - G:\examples\Rect.asm". The menu bar includes File, View, Run, and Help. The toolbar contains icons for file operations, execution (single step, break, step over, step into), and speed control (Slow, Fast). The "Animate" dropdown is set to "Program flow", "View" is "Screen", and "Format" is "Decimal".

On the left, the "ROM" window shows assembly instructions. Instruction 15, "@32", is highlighted in yellow. Below the ROM window, the "PC" (Program Counter) register is shown with the value 15. A red arrow points from the highlighted instruction 15 to the PC register.

In the center, the "RAM" window shows memory addresses and their values. Address 17 is highlighted in yellow and contains the value 17184. Below the RAM window, the "A" (Address) register is shown with the value 17. A red arrow points from the highlighted address 17 to the A register.

Two yellow callout boxes provide additional information:

- Data memory (32K RAM), used for:**
 - General-purpose data storage (variables, arrays, objects, etc.)
 - Screen memory map
 - Keyboard memory map
- Address (A) register, used to:**
 - Select the current RAM location

OR

 - Set the Program Counter (PC) for jumps (relevant only if the current instruction includes a jump directive).

Registers

The screenshot shows the CPU Emulator (1.4b3) interface. The ROM window displays assembly code, with line 15 highlighted: `@32`. The RAM window shows memory addresses, with address 17 highlighted and containing the value 17184. The PC register is 15, and the A register is 17. The ALU window shows the D Input as 25 and the M/A Input as 17184, resulting in an ALU output of 17184. Annotations include a yellow box for registers D, A, and M, an orange box for M (=RAM[A]), and another orange box for the D register.

Registers (all 16-bit):

- **D**: Data register
- **A**: Address register
- **M**: Stands for the memory register whose address is the current value of the Address register

M (=RAM[A])

D

ALU

D Input : 25

M/A Input : 17184

ALU output : 17184

Arithmetic/Logic Unit

The screenshot shows a CPU Emulator window titled "CPU Emulator (1.4b3) - G:\examples\Rect.asm". The interface includes a menu bar (File, View, Run, Help), a toolbar with icons for file operations and execution, and a status bar. The main area is divided into three panels: ROM, RAM, and ALU.

ROM Panel: Displays assembly instructions. The current instruction is highlighted in red at address 14: `D=M`. The instruction at address 15 is highlighted in yellow: `@32`.

RAM Panel: Displays memory contents. The value at address 17 is highlighted in yellow: 17184. The value at address 16 is 25.

ALU Panel: Shows the ALU operation. The D Input is 25, and the M/A Input is 17184. The ALU output is 17184.

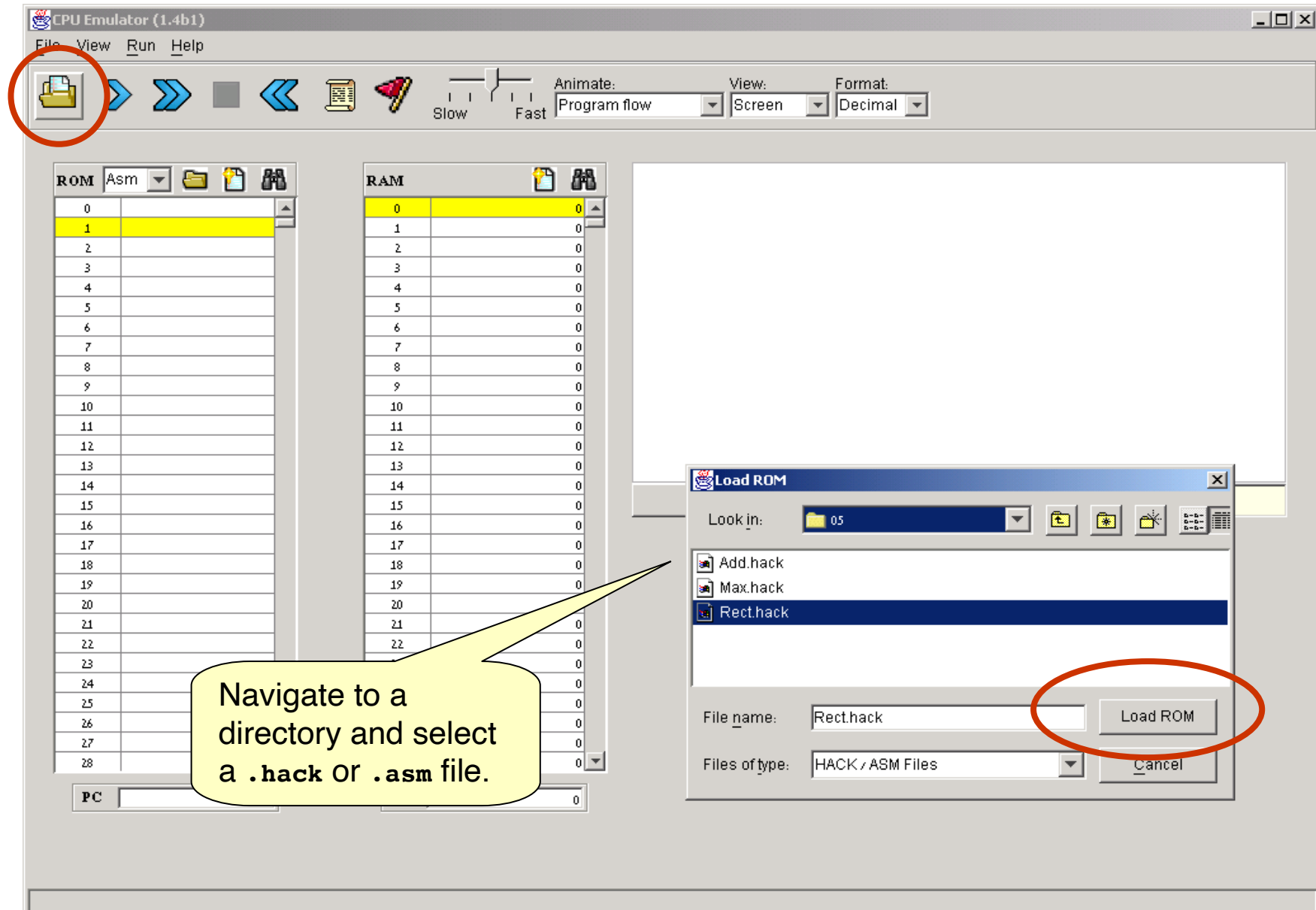
Annotations:

- Current instruction:** Points to the instruction at address 14: `D=M`.
- M (=RAM[A]):** Points to the value at address 17: 17184.
- D:** Points to the D Input of the ALU: 25.
- A:** Points to the A register value: 17.

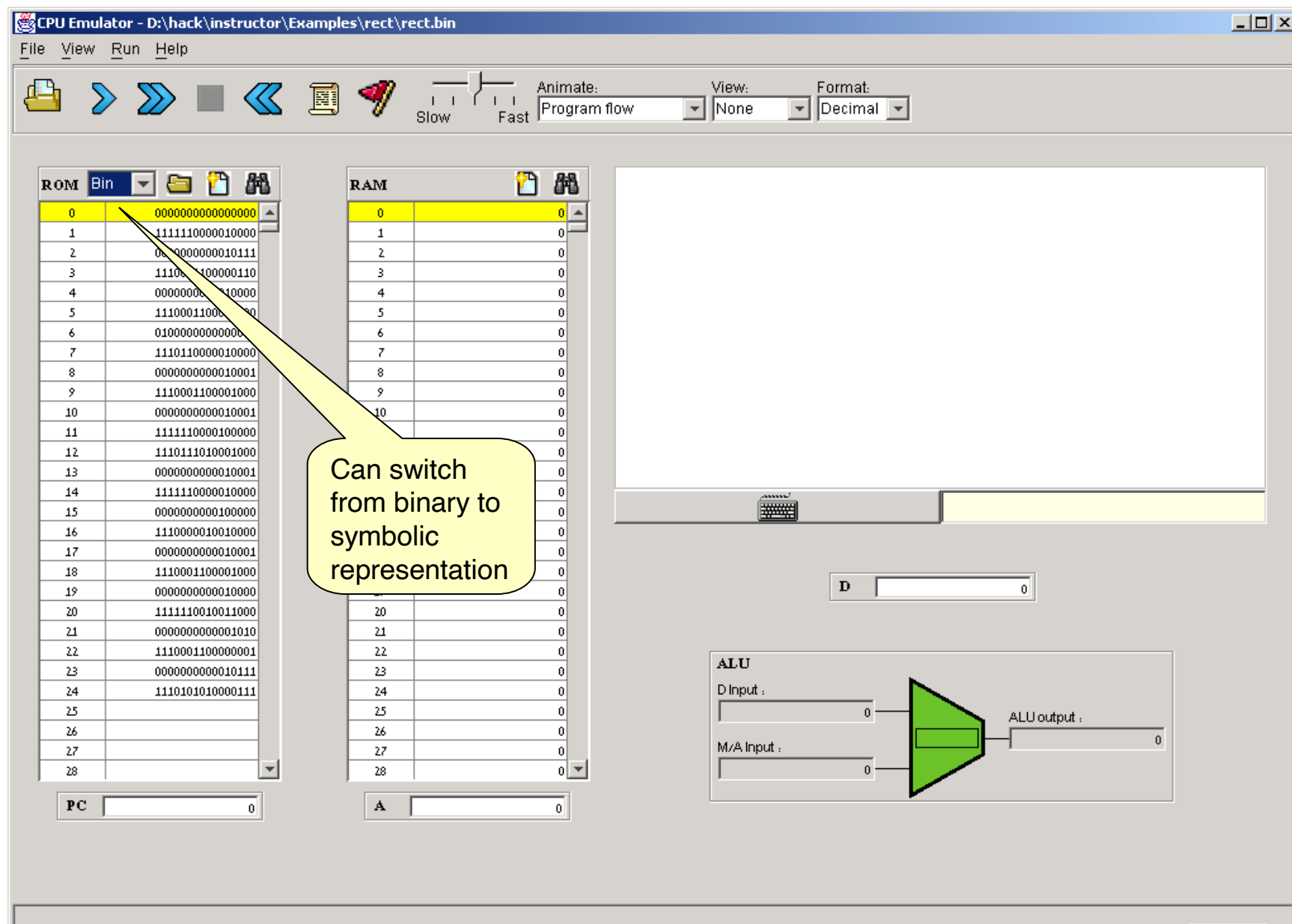
Arithmetic logic unit (ALU)

- The ALU can compute various arithmetic and logical functions (let's call them f) on subsets of the three registers $\{M, A, D\}$
- All ALU instructions are of the form $\{M, A, D\} = f(\{M, A, D\})$ (e.g. $M=M-1$, $MD=D+A$, $A=0$, etc.)
- The ALU operation (LHS destination, function, RHS operands) is specified by the current instruction.

Loading a program



Loading a program



Running a program

The screenshot shows a CPU Emulator window titled "CPU Emulator - D:\hack\instructor\Examples\rect\rect.bin". The interface includes a menu bar (File, View, Run, Help), a toolbar with icons for file operations and execution, and a speed control slider. The main area is divided into several panels: ROM (Asm), RAM, a keyboard input area, and a display area. The ROM panel shows assembly code, and the RAM panel shows memory values. The display area is currently blank. Callouts provide instructions on how to use the emulator.

2. Click the "run" button.

1. Enter a number, say 50.

3. To speed up execution, use the speed control slider

4. Watch here

Program's description: Draws a rectangle at the top left corner of the screen. The rectangle's width is 16 pixels, and its length is determined by the current contents of RAM[0].

Note: There is no need to understand the program's code in order to understand what's going on.

Running a program

2. Click the "run" button.

1. Enter a number, say 50.

3. To speed up execution, use the speed control slider

4. Watch here

Program's description: Draws a rectangle at the top left corner of the screen. The rectangle's width is 16 pixels, and its length is determined by the current contents of RAM[0].

Note: There is no need to understand the program's code in order to understand what's going on.