

STAM Center

ASU Engineering

BRISC-V Simulator
Adaptive and Secure Computing Systems Lab • Boston University

A Browser-based RISC-V Simulator

Adaptive and Secure Computing Systems (ASCS) Laboratory

1

STAM Center

ASU Engineering

BRISC-V Simulator

- BRISC-V Simulator let's you:
 - Run RISC-V assembly code in the browser
 - Debug hand-written assembly
 - Run code until completion or until it hits a breakpoint
 - Step through execution, instruction-by-instruction
 - View the state of memory and registers at each step
 - View how each instruction is constructed – opcodes, registers, immediate values etc.

<https://ascslab.org/research/briscv/simulator/simulator.html>

2

STAM Center

ASU Engineering

BRISC-V Simulator

ASCS LABORATORY

C Source Code Pane
Not interesting for this class
Let's you compile C to RISC-V assembly

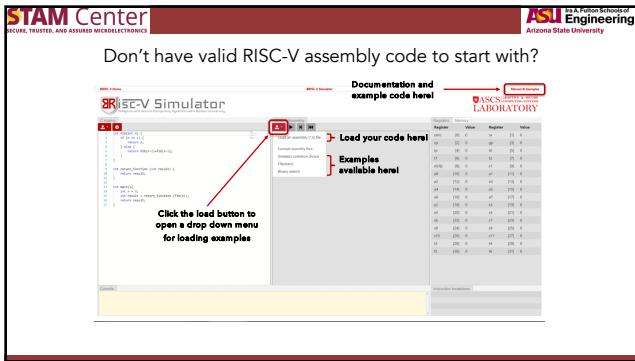
Console Pane
Shows messages from the compiler and the simulator
Also used for system calls – let's you input and print values

RISC-V Assembly Pane
Let's you run assembly step-by-step

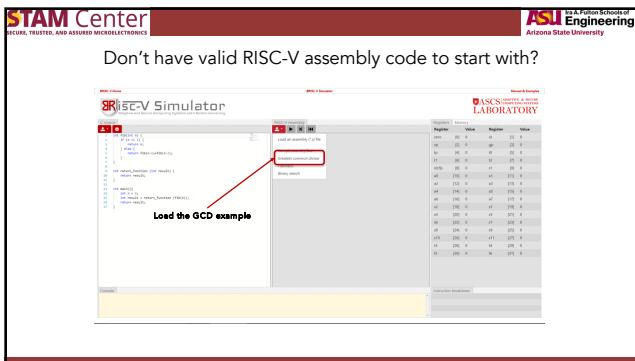
Register & Memory Pane
Shows the state of registers and memory

Instruction Breakdown Pane

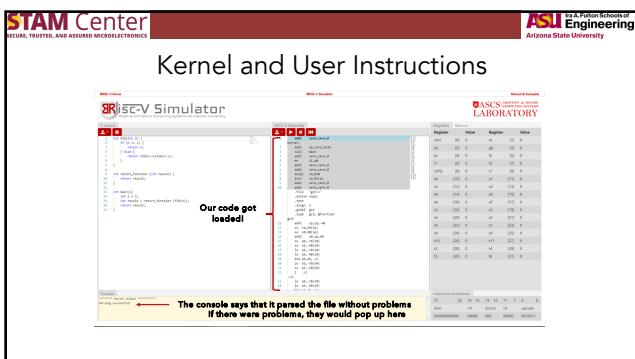
3



4



5



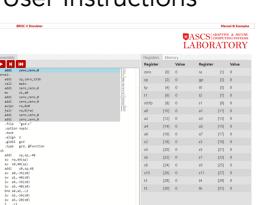
6

Kernel and User Instructions



Grey instructions are kernel instructions

They setup some registers like the stack pointer, and jump to label "main"



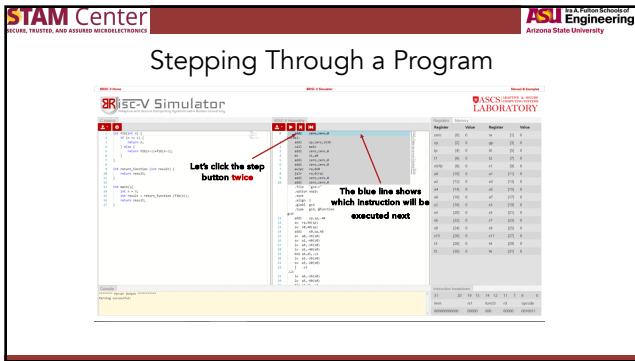
Grey instructions are kernel instructions

They setup some registers like the stack pointer, and jump to label "main"

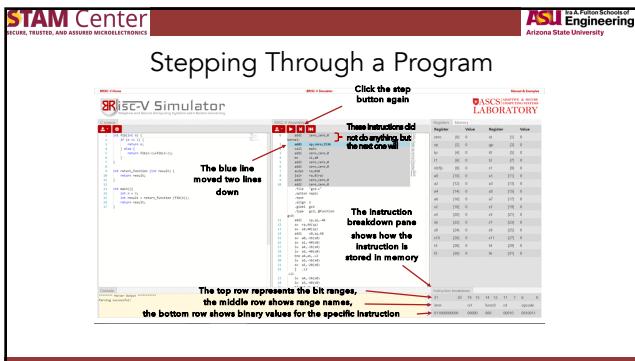
7

8

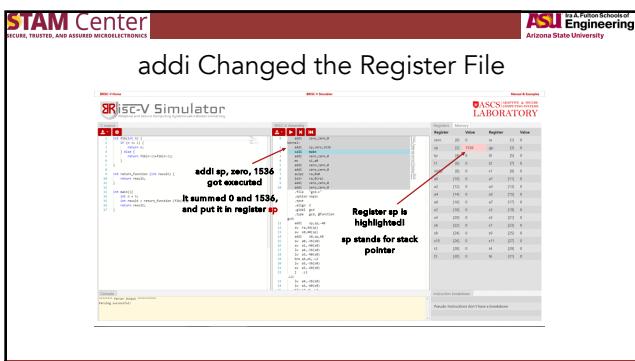
9



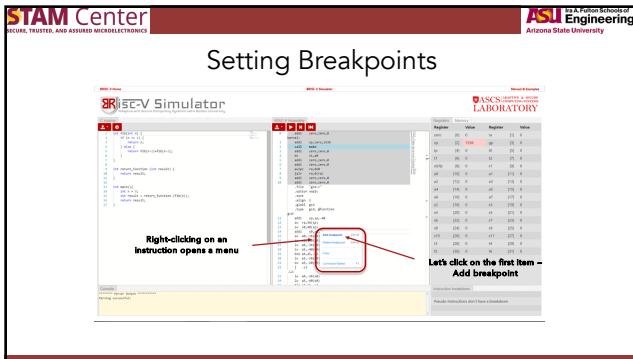
10



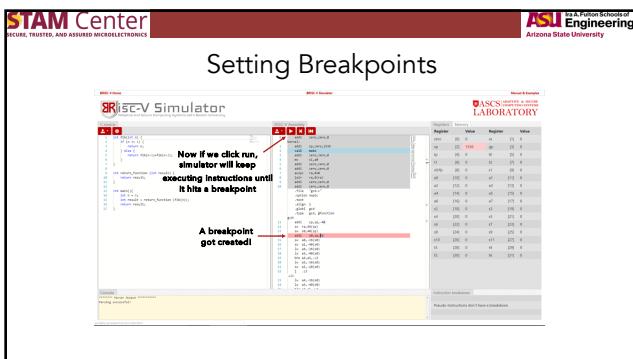
11



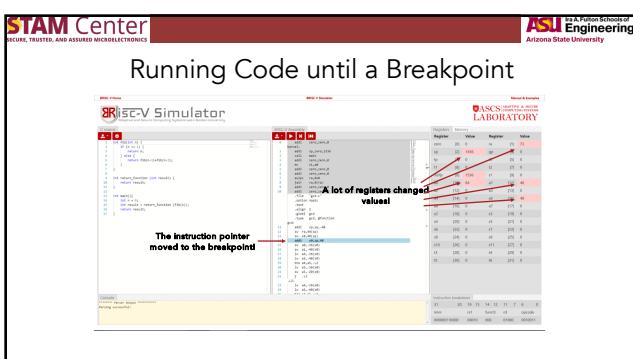
12



13



14



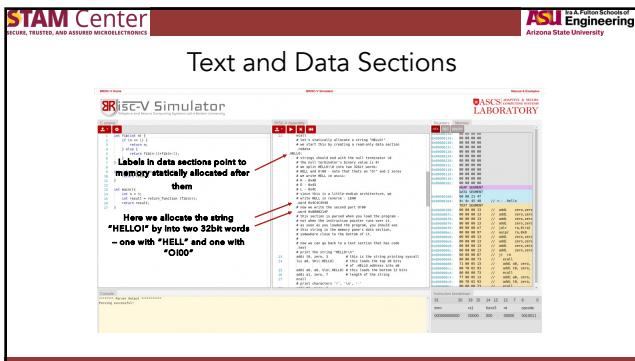
15

Text and Data Sections



The screenshot shows the RISC-V Simulator interface. On the left, the assembly code for a program is displayed, including sections for .data and .rodata. A red arrow points from the text "data and .rodata sections allow us to set some memory before the program is run" to the .data section in the assembly code. On the right, a memory dump window shows the initial memory state with labels for .text, .data, and .rodata sections. A red arrow points from the text ".text sections are used for code. Programs are in the text section by default" to the .text section in the assembly code.

16



17

The screenshot shows the STAM Center SBCV Simulator interface. On the left, the assembly code for a C program is displayed:main() { char str[10]; str[0] = 'H'; str[1] = 'E'; str[2] = 'L'; str[3] = 'L'; str[4] = 'O'; str[5] = '\0'; printf("%s", str);}On the right, the memory dump panel shows the allocated memory for the string 'HELLO'. A red arrow points from the text in the center of the slide to the memory dump panel, highlighting the pointer variable `_HELLO` and the string data.

18

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Arizona State University
Engineering

System Calls

- We also provide some simple system calls
- System calls are used for functionalities provided by the operating system
 - Think file systems, IO, etc.
- In RISC-V, system calls look something like:
 - Put the type of system call you want in register t0
 - More about that on the next slide
 - Put any arguments you may have in a0 and a1
 - Call instruction ECALL
 - If the system call has return values, they will be in a0
- **To really get familiar with syscalls, try running the example syscall file in the simulator**

19

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Arizona State University
Engineering

Supported Syscalls

Syscall	Syscall ID (put this in t0)	Description
Print integer	1	Print integer value in a0 to console
Print char	2	Print ascii value in a0 to console
Print string	3	Print string with address in a0 and length in a1 to console
Read integer	4	Read integer from console into a0
Read char	5	Read character from console into a0 as an ascii value
Read string	6	Read string of length given in a1 from console and store it at address in a0
SBRK	7	Dynamically allocate the amount of bytes specified in a0. The pointer to the beginning of the newly allocated memory will be stored in a0. The value in a0 can be negative, if you want to deallocate some memory!

20

STAM Center
SECURE, TRUSTED, AND ASSURED MICROELECTRONICS

ASU Arizona State University
Engineering

RISCV Simulator

That's All Folks!
Now open a text editor,
write some assembly,
and try to run it!



21
