Author : Irtaza Hyder

Date : 2026-01-30

Module : RISC-V Arch Test

Section : RISC-V Arch Test

Task Name : Task 0

Description: Answer the following questions:

1. What are these flags used by toolchain "-march=rv32g -mabi=ilp32 -nostdlib -T link.ld test.S -o test.elf"

Flags	Description
-march=rv32g	Specifies the machine architecture and extensions. Rv32g means we are using a general purpose 32bit risc-v machine. The g extension is short for IMAFDZicsr_Zifencei (integer multiplication and division, atomicity, single precision floating, double precision floating point, csr, and fence extensions)
-mabi=ilp32	Specifies the calling conventions. The ilp32 flag means that integer, long and pointer datatypes use 32 bits.
-nostdlib	Linker flag to not use the standard system startup files and or libraries when linking.
-T link.ld	Use link.ld as the linker script.
-o test.elf	Store the built elf program as test.elf.

2. What are these flags used by spike "--isa=rv32i -l --log-commit"

Flags	Description
--isa=rv32i	Specifies the risc-v ISA to be used. In this case we are using risc-v 32 bit base integer ISA.
-l	Generate log of execution.
--log-commit	Generate log of commit info, recording registers and memory transactions.

3. What is the purpose of .tohost and .fromhost symbols?

The .tohost section is a memory section in the linker file which is used to communicate between host and guest.

.tohost: Specific memory location where the guest program writes data that it wants the simulator (the host) to process.

.fromhost: This is the memory location where the host machine places data for the guest program to read.

4. What does the .pushsection directive do?

It pushes the current section and its subsections to the stack, and switches it with the context of the new section. In our example, we are loading the .tohost section, and making it “aw” (writeable section), and designating it as @progbits (data section)

5. What is the purpose of the begin_signature symbol?

It is a data section delimiter which specifies the memory location from where the C output from Spike is located.

6. What does the trap_vector label do?

It states the location of the interrupt vector table. Everytime, an interrupt is called, the control flow dictates that the IVT table would be used to find the location of the interrupt handler routine.

7. What is stored in the mcause CSR register?

When a trap is taken into M mode, the mcause register is written with the code indicating the event that caused the trap. The MSB is 1 if the trap is due to an interrupt, otherwise 0. The rest of the bits indicate the contain the exception code.

8. What is the mepc register and what does it contain?

mepc register contains the virtual address of the instruction that was interrupted or encountered an exception. It is written when a trap is taken into M mode.

9. What does the _start label do and why does it jump to main?

The start label is the entry point for the program execution. In the given program we have explicitly written the instruction `j main` in the _start label, which causes the PC to jump to the address of main.

10. What happens in the test_fail handler and why?

The program goes into an infinite loop. It is to prevent the Spike simulator from crashing, and for us to debug what may have gone wrong.

11. What value gets written to tohost at the end of a passing test?

The integer `1` is written.

12. Why does write_tohost enter an infinite loop?

To prevent spike to exit after writing the value to the host.