Endianness and Struct Packing

Code

#include <stdio.h>

int main() {

union {

unsigned int i;

unsigned char c[4];

} test;

test.i = 0x01020304;

printf("Integer: 0x%x\n", test.i);

printf("Bytes in memory: %02x %02x %02x %02x\n",

test.c[0], test.c[1], test.c[2], test.c[3]);

if (test.c[0] == 0x04) {

printf("Little-endian detected\n");

} else if (test.c[0] == 0x01) {

printf("Big-endian detected\n");

} else {

printf("Unknown endian\n");

}

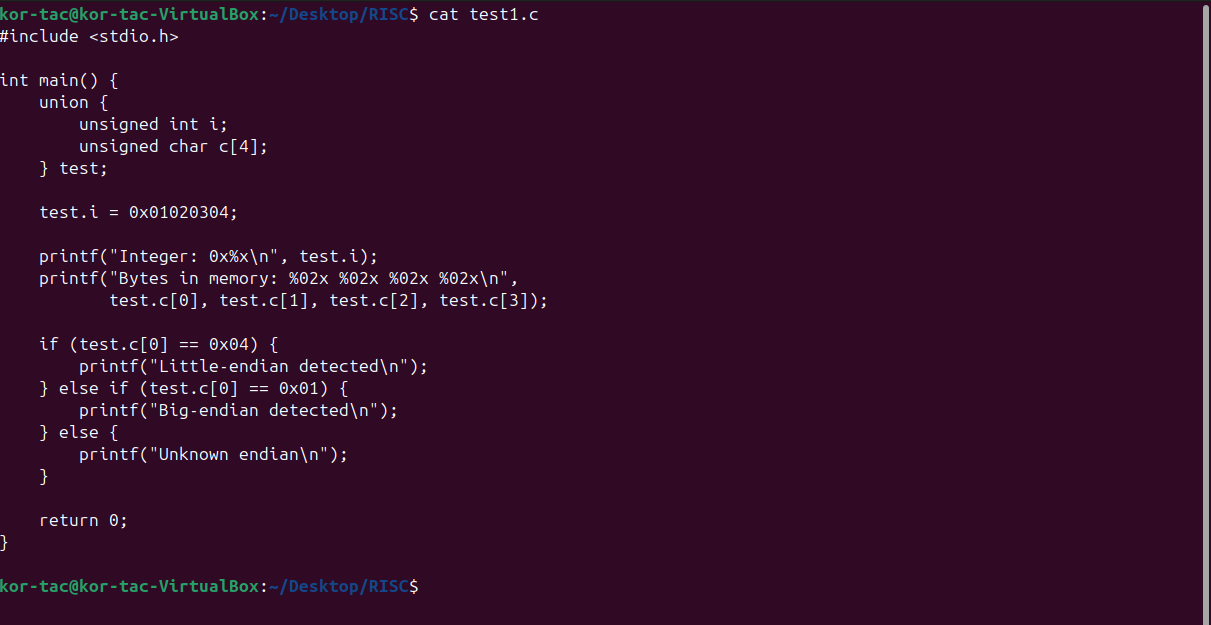
return 0;

}

 he integer 0x01020304 is stored in 4 bytes.

 If the first byte (c[0]) in memory is 0x04, it means least significant byte is stored first → **little-endian**.

 If the first byte is 0x01, then most significant byte is stored first → **big-endian**.



To Compile type the command:

riscv32-unknown-elf-gcc -march=rv32imac -mabi=ilp32 -o endian\_test.elf test1.c



Finally to see the output

spike --isa=rv32imac pk endian\_test

