Fault Report

Bowen Bai 969899
Zeming Yao 962403

In the machine.adb, there are four kinds of faults, range checking faults, overflow checking faults, dividing by zero faults, and the unused assignment.
The function incPC is the one which causes the range checking faults. This function is used to increase the PC by an Offset value. However, the sum of PC and this Offs value can be out of the range of ProgramCounter type which is from one to max length of a program. Using the SPARK prove, it will give this message:
machine.adb:24:40 medium: range check might fail, in call inlined at machine.adb:146 (e.g. when PC= 65536)
Function DoLdr and DoStr also might cause the range checking faults. The reason for these two faults are similar since the range of an Addr value is from 0 to 65535 however the result of the sum of the value in a register and a Offset value can be negative or larger than the valid range. The messages given by SPARK:
machine.adb:68:33/ 78:33 medium: range check might fail, in call inlined at machine.adb: 125/128 (e.g. when A = 0).
If these faults are triggered, the VM would give errors when running.
The overflow checking faults appear in multiple functions of this VM which are DoAdd, DoSub, DoMul. What's more, the DoLdr and DoStr could also cause this fault besides the range checking one.
For this VM, all data is defined as a DataVal type which used to represents a 32-bit integer. The range of DataVal is from -(2^31) to (2^31-1). All the operations should be in this range. However, in those functions mentioned above, all the operations on data value didn't check whether they are valid before it's done. For example, in the DoAdd function, when adding two values in registers, which are both 2^31-1, the result must be out of the valid range of a 32-bit integer. If this fault is triggered, the overflow would happen because that the maximum of a 32-bit integer cannot hold that result.
The SPARK give messages for that potential fault:
DoAdd:
machine.adb:33:29 medium: overflow check might fail, in call inlined at machine.abd:113 (e.g. when Regs = (others => -1073741825))
DoSub:
machine.adb:42:29 medium: overflow check might fail, in call inlined at machine.abd:116(e.g. when Regs = (1 => 1, others => -2147483648))
DoMul:
machine.adb:51:29 medium: overflow check might fail, in call inlined at machine.abd:119(e.g. when Regs = (1 => 2, others => -2))
DoLdr, DoStr:
machine.adb:68:33 medium: overflow check might fail, in call inlined at machine.abd:125(e.g. when A = 0, Regs = (others => -2147483648))
machine.adb:68:33 medium: overflow check might fail, in call inlined at machine.abd:128(e.g.

when A = 0, Regs = (others => -2147483648))

The dividing by zero fault is caused by the DoDiv function. When this function is trying to do the dividing, it didn’t check whether the value in second register is zero. The VM would terminate with error when this fault is triggered. SPARK gives the message to indicate this potential fault:

machine.adb 60:29 medium: divide by zero might fail, in call inlined at machine.adb:122 (e.g. when Reg = (others => 0))

SPARK also gives some warning of unused assignment. In the ExecuteProgram function, there is a loop to execute each instruction of the program. After an instruction is executed, a ReturnCode will be given to show whether this instruction is executed successfully. Then the VM would increase the PC to execute next instruction. However, the it didn’t check the ReturnCode before next instruction being executed which made the assigned value unused.

The warning message given by SPARK is:

```
machine.adb 34:11 warning: unused assignment, in call inlined at machine.adb: 113
machine.adb 43:11 warning: unused assignment, in call inlined at machine.adb: 116
machine.adb 52:11 warning: unused assignment, in call inlined at machine.adb: 119
machine.adb 61:11 warning: unused assignment, in call inlined at machine.adb: 122
machine.adb 71:11 warning: unused assignment, in call inlined at machine.adb: 125
machine.adb 81:11 warning: unused assignment, in call inlined at machine.adb: 128
machine.adb 89:11 warning: unused assignment, in call inlined at machine.adb: 131
```