计算机组成原理实验报告

姓名： 吴迪 学号：11810716

1. 实验目的

Learn the difference between exception and interruption,practice on the trap,eret,mfc0,mtc0 instructions.Learn the kernal spcace and special register such as epc,cause and vaddress in Coprocess 0 ,Practice to rewrite the exception handler.

1. 实验内容

1. Implementing a Simple arithmetic calculator for 31-bits Unsigned Numbers:

1)Read the operators and operands in the main method, call the function to implement the arithmetic calculator

2)If the above abnormal situation does not occur, output the result of the operation.

3)In following situation a exception will be triggered:

1>the value of input is out of the range of 31-bits unsigned number

2>the addition overflow

3>the multiplication result exceeds the width of a word

4>the divisor is zero

The exception handler do the following things:

1>stop the program running

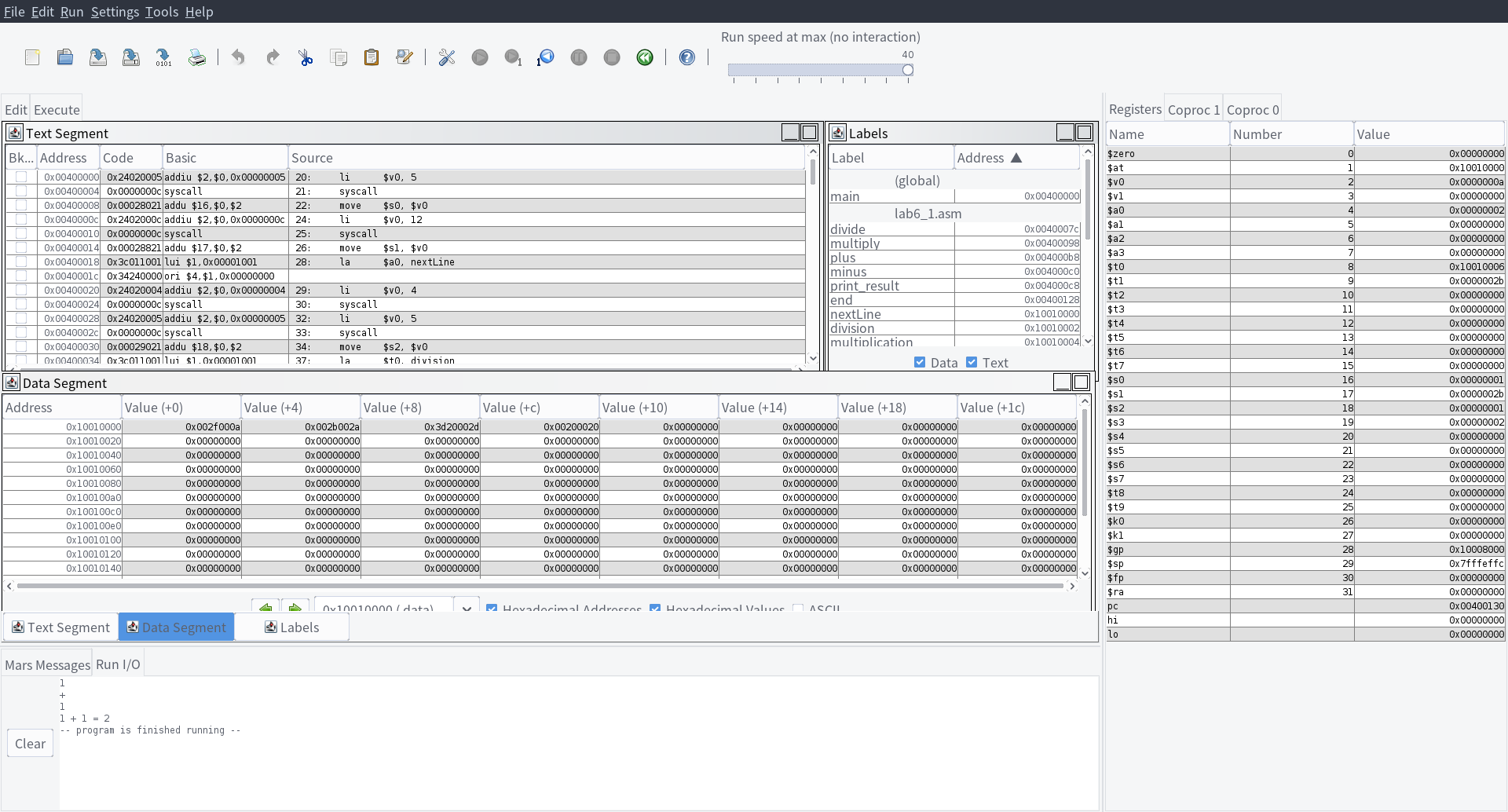
2>output prompt information, including “Runtime exception at 0x\_(the address of the instruction which triggered the exception)”, and the cause of the exception (illegal input , arithmetic overflow , divide by zero)

3>exit the program.

1. 实验步骤（阐述代码思路或操作步骤）
2. First get the data, then match the operand and do the operation. Next print out the result then exist.
3. When any exception happens, we jump to the exception handler. Then we get the corresponding error message related to the exception code, then print out. After all, exit the program.
4. 实验结果（截图并配以适当的文字说明）
5. Normal operations
   1. Addition

Input: 1 + 1

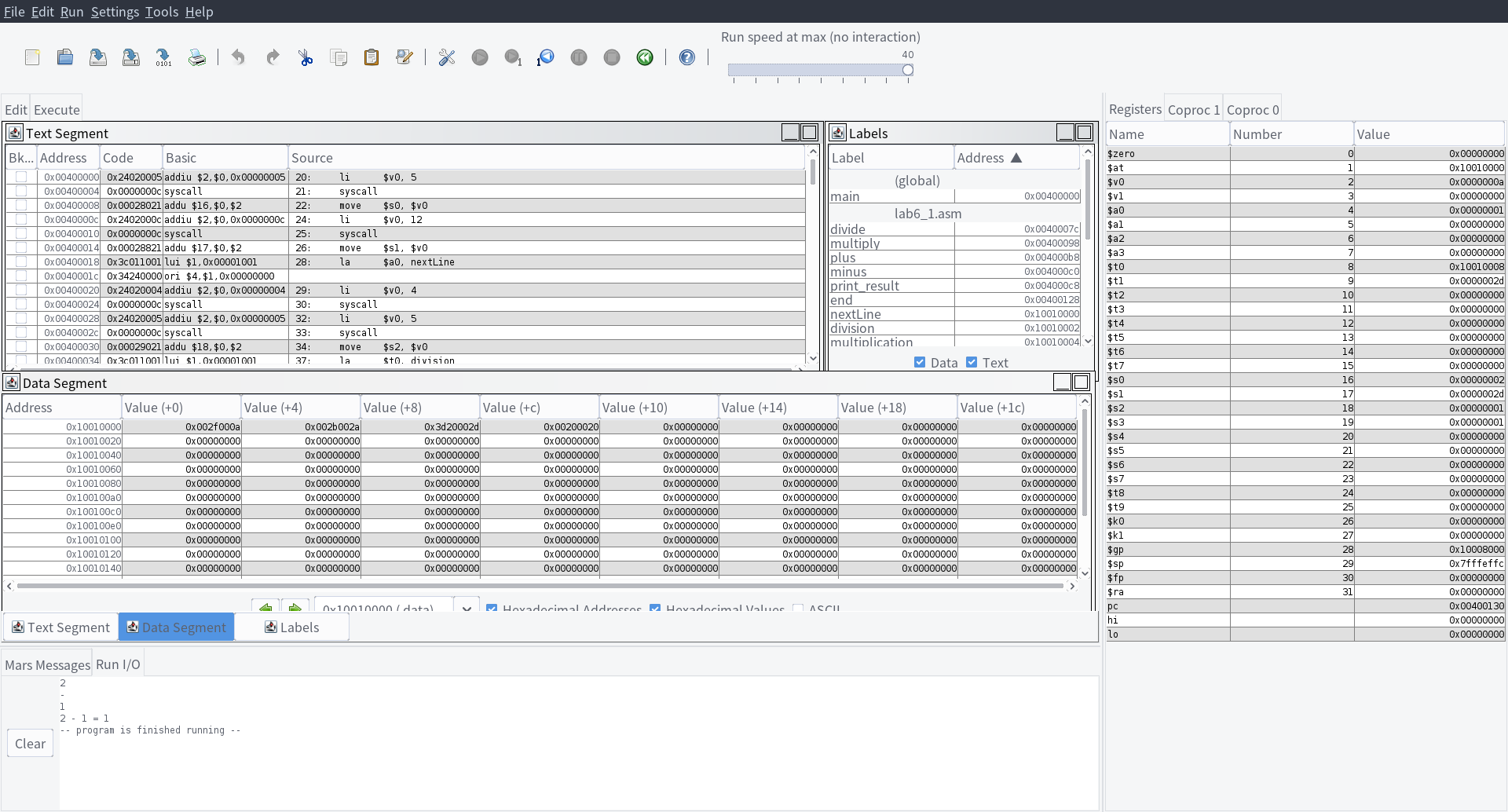
Output: 1 + 1 = 2



* 1. Subtraction

Input: 2 - 1

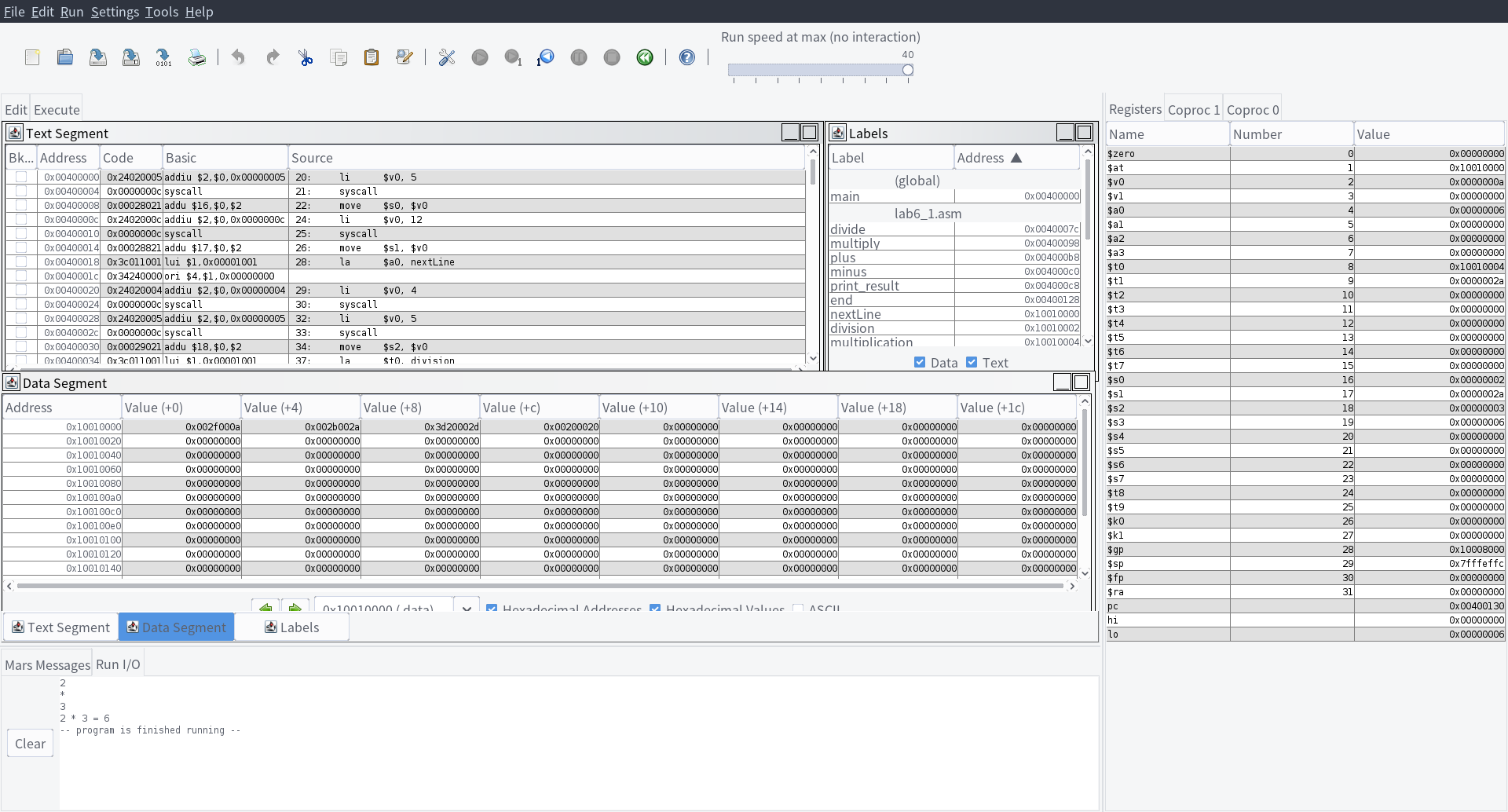
Output: 2 - 1 = 1



* 1. Multiplication

Input: 2 \* 3

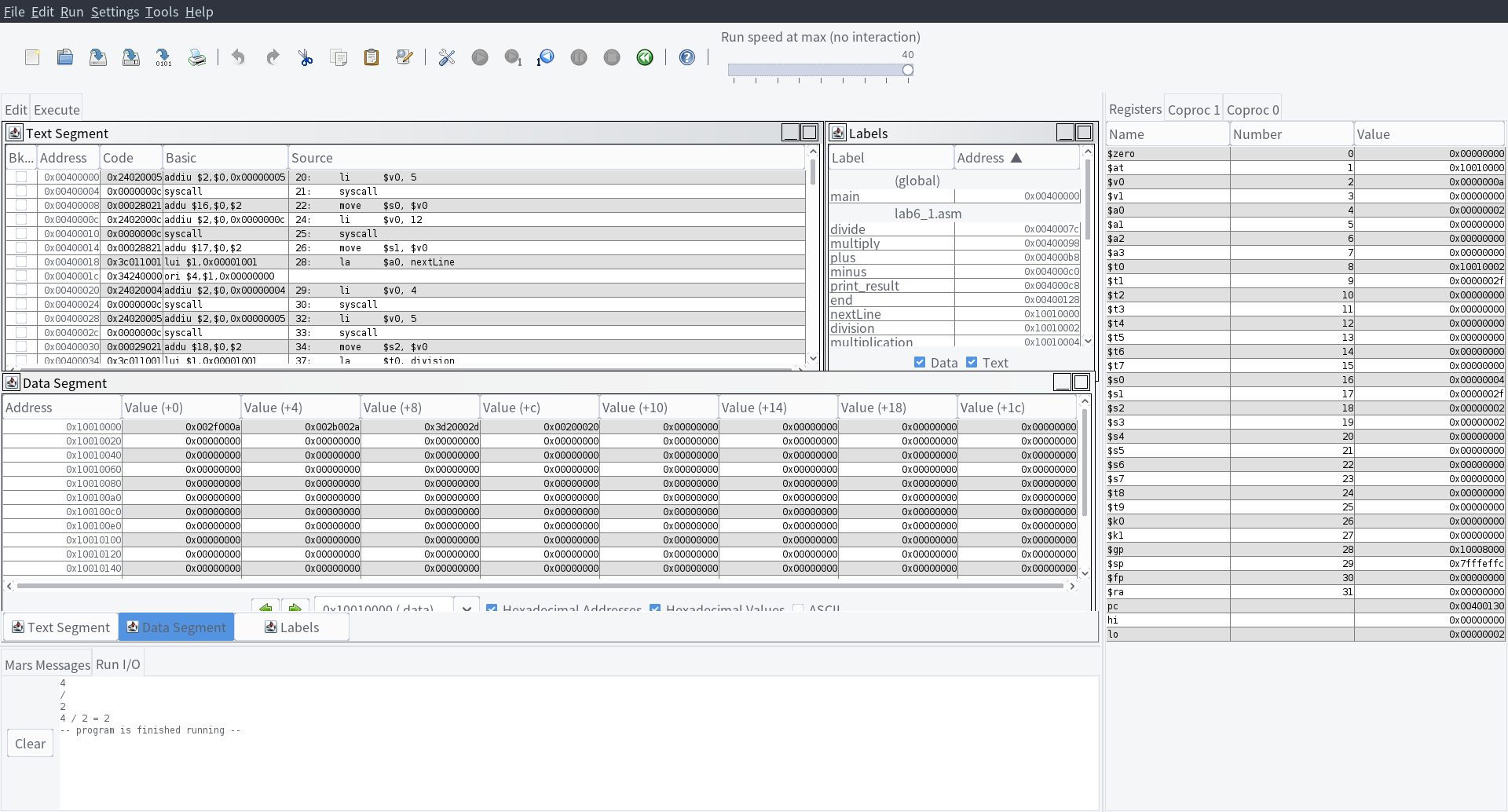
Output: 2 \* 3 = 6



* 1. Division

Input 4 / 2

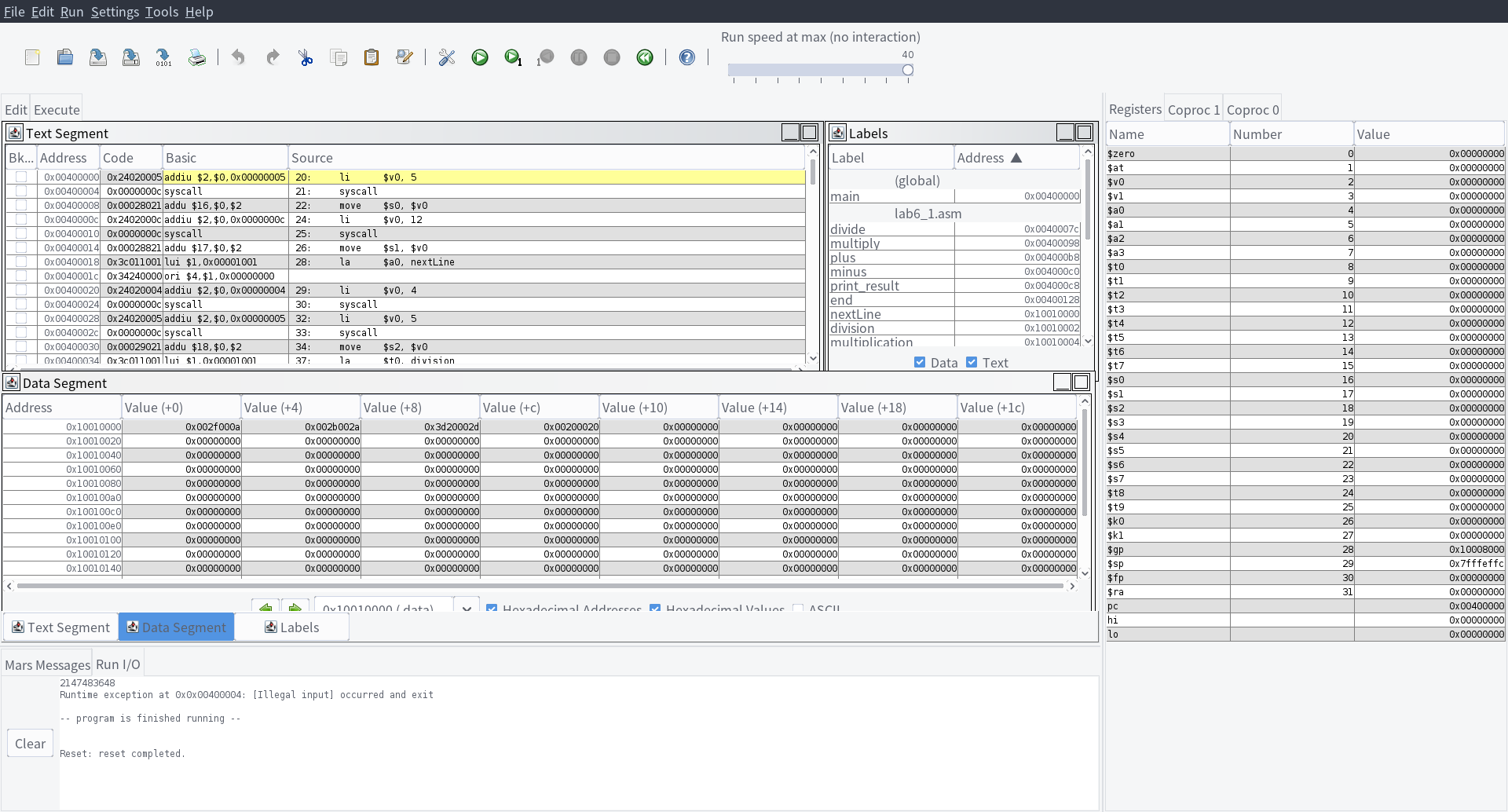
Output 4 / 2 = 2



1. Errors handle
   1. the value of input is out of the range of 31-bits unsigned number

Input: 2147483648

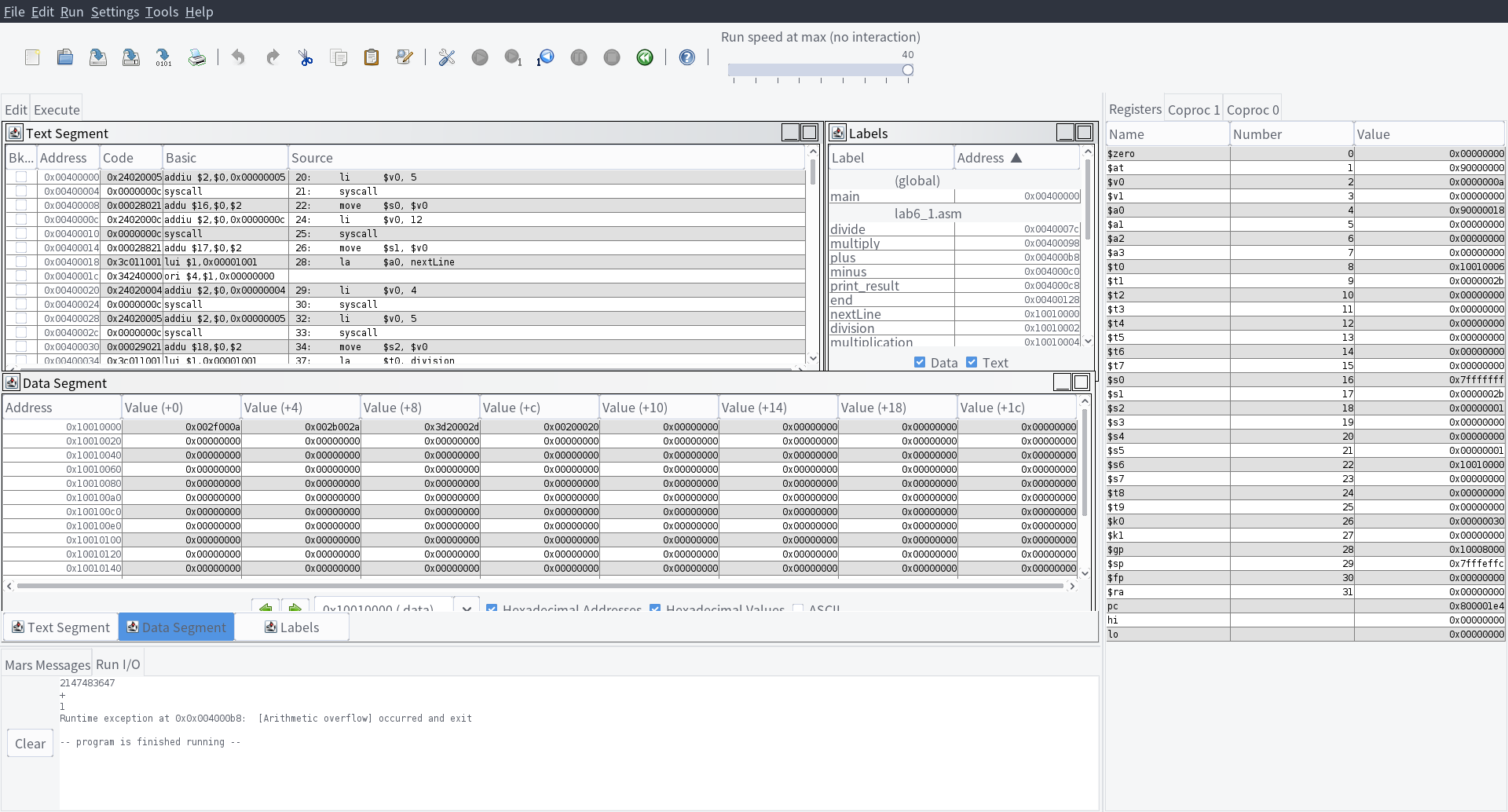
Output: Runtime exception at 0x0x00400004: [Illegal input] occurred and exit



* 1. the addition overflow

Input: 2147483647 + 1

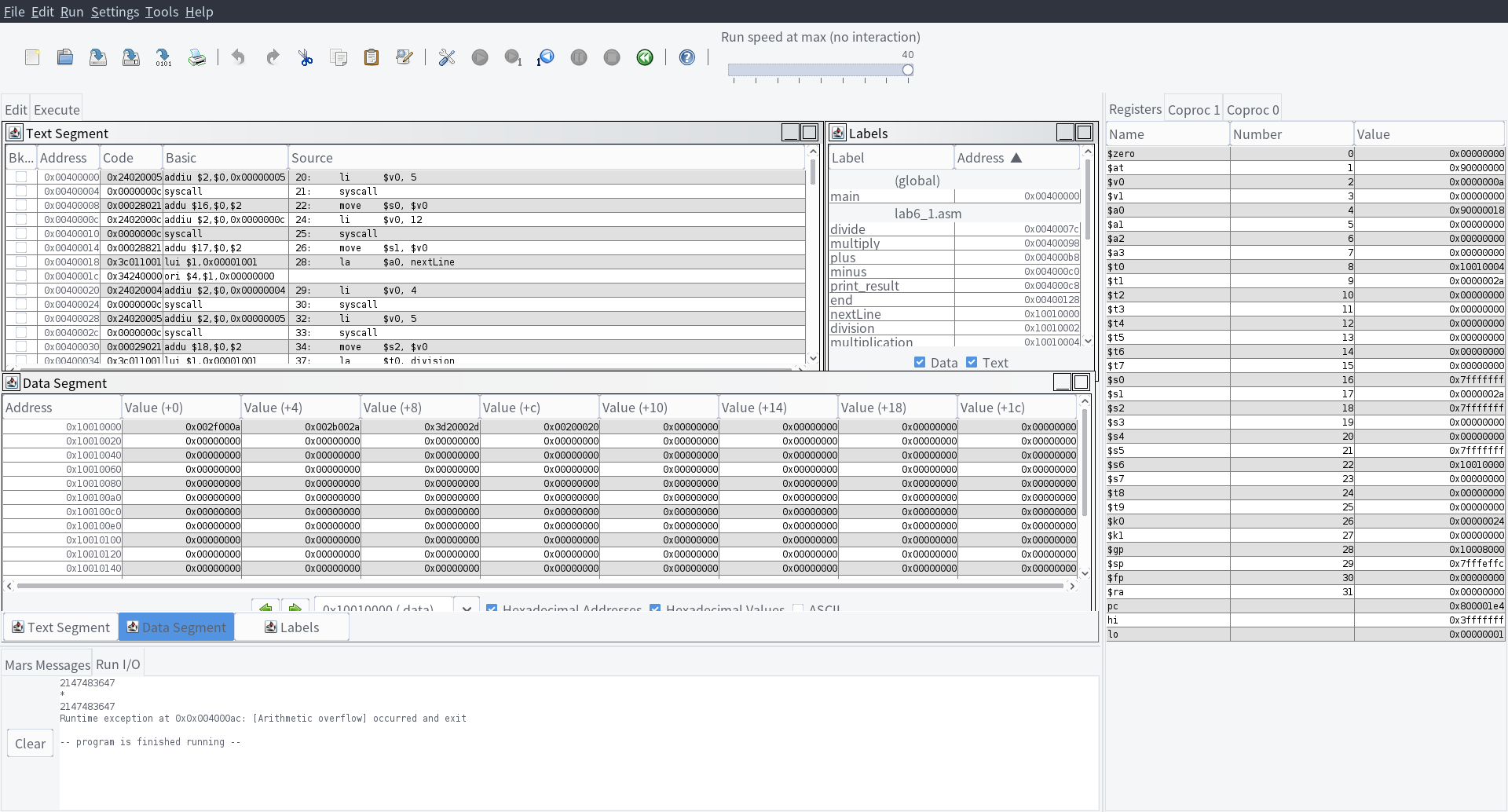
Output: Runtime exception at 0x0x004000b8: [Arithmetic overflow] occurred and exit



* 1. the multiplication result exceeds the width of a word

Input: 2147483647 \* 2147483648

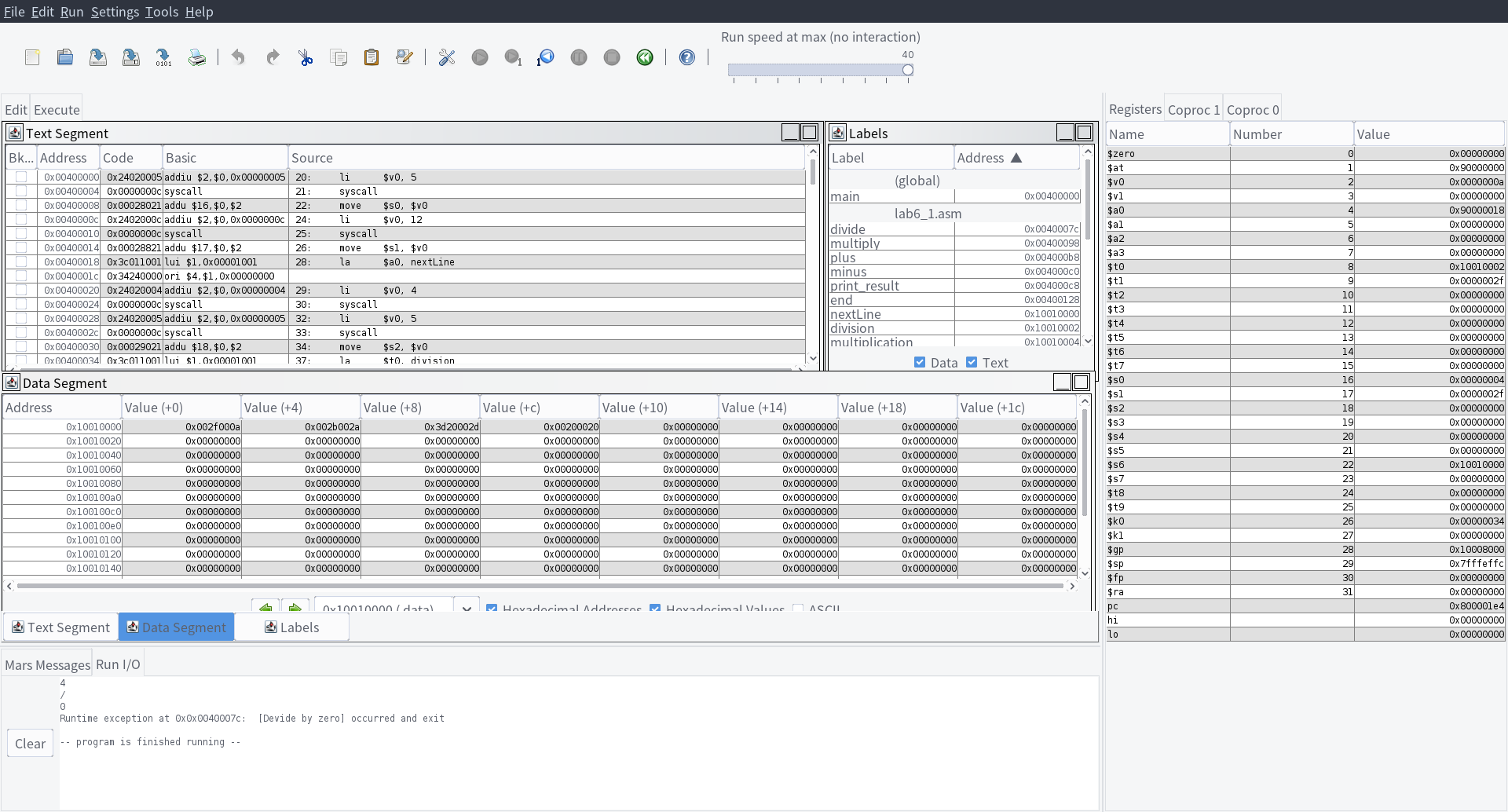
Output: Runtime exception at 0x0x004000ac: [Arithmetic overflow] occurred and exit



* 1. the divisor is zero

Input: 4 / 0

Output: Runtime exception at 0x0x0040007c: [Devide by zero] occurred and exit



1. 实验分析（遇到的问题以及解决方案）
2. Different error may get the same cause code when using the instructions. So we need to test the data and store a different error code in a place other than the $13 in Coproc 0.
3. In the document in mars, it says there is an cause code 15 represents the “divide by zero” error. However, when I try some number divide by zero, the cause code is not 15. In fact, it’s 9, which is the same as the overflow error occurred by addition. So the final exception need an additional bias to separate from other error.
4. 实验小结与体会

The exception handler is a very difficult process. The default exception handler is good enough but may not suitable to our requirements. The document in mars need to be updated.