

Branch Counting

CMPUT 229

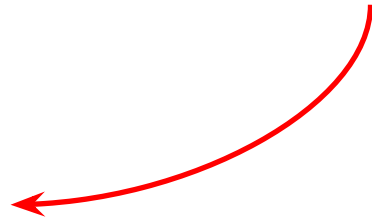
University of Alberta

Consider the program below

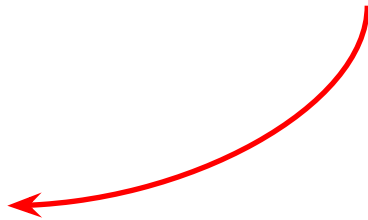
RISC-V Assembly

```
1  foo:
2      bge  t0 t1 target
3      addi t0 t0 1
4      j  foo
5  target:
6      addi t1 t1 1
7      li  t0 0
8      li  t2 10
9      blt  t1 t2 foo
```

Forward Branch



Backward Branch

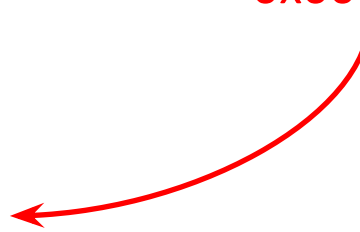


Now consider how branch targets are stored in assembly

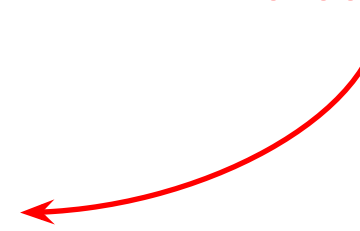
RARS

Address	Code	Basic
0x00400000	0x0062d663	bge x5 x6 0x0000000c
0x00400004	0x00128293	addi x5 x5 1
0x00400008	0xff9ff06f	jal x0 0xffffffff8
0x0040000c	0x00130313	addi x6 x6 1
0x00400010	0x00000293	addi x5 x0 0
0x00400014	0x00a00393	addi x7 x0 10
0x00400018	0xfe7344e3	blt x6 x7 0xffffffffe8

Positive Offset
 $0x00400000 + 0x0000000c$
 $= 0x0040000c$



Negative Offset
 $0x00400018 + 0xffffffffe8$
 $= 0x00400000$



Task

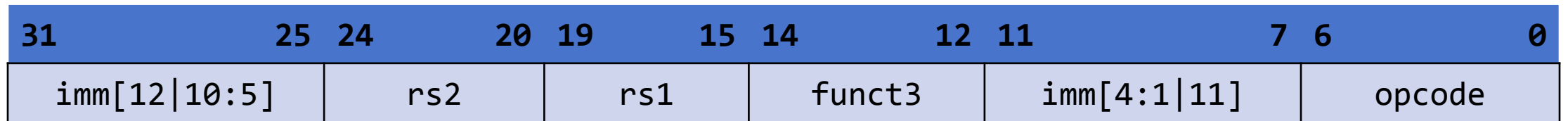
Identify branches by their opcode, and use the offset to determine if the branch goes forwards or backwards.

RISC-V Branch Instructions

beq	Branch Equal
bge	Branch Greater than or Equal
bgeu	Branch Greater than or Equal Unsigned
blt	Branch Less Than
bltu	Branch Less Than Unsigned
bne	Branch Not Equal

SB Type Format

All branch instructions are encoded in the SB Instruction Type Format



The relative offset is encoded in bits 7-11 & 25-31 of the instruction

Assignment

Students must implement the branchCounting function

Instructions Array

- The **instructions array** of a program is an array of RISC-V instructions in binary representation
- The instructions in the array are the instructions in the input program
- The array is terminated by the sentinel value 0xFFFFFFFF

branchCounting

This function is the entry point of student's solution to this lab. It counts forward and backward branches in the program pointed to by a0. Returns the number of forward branches in a0 and the number of backward branches in a1.

Arguments:

a0: Pointer to the instructions array of the input program.

Returns:

a0: Integer number of forward branches in input.

a1: Integer number of backward branches in input.

Input Guarantees

- The instructions array is terminated by the sentinel value 0xFFFFFFFF.
- All instructions in the input file are valid RISC-V instructions.
- No branches will have an offset of zero.
- There are at most 100 instructions in the input file.

Testing Workflow

You can generate input files from any RISC-V .s file satisfying input guarantees. To generate the binary file, test.bin, for the test program, test.s, you can run the following command:

```
rars a dump .text Binary test.bin test.s
```

Run your solution on the input file with

```
rars branchCounting.s pa test.bin nc > lab.out
```