Lab RISC-V to ARM - ALU

CMPUT 229

About this lab

- Translate RISC-V Arithmetic Logic Unit (ALU) instructions into ARM ALU instructions.
- Control instructions will be translated in the next lab.

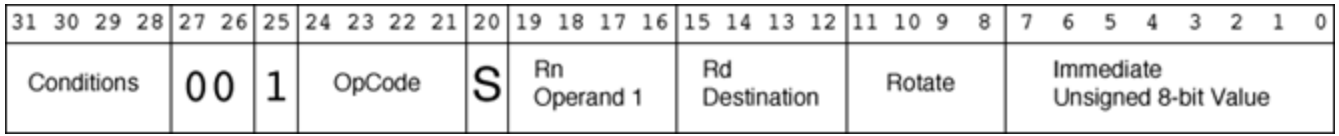
What is ARM?

- ARM is a RISC architecture.
- A Reduced Instruction Set Computer, or RISC (/rɪsk/), is a computer with a small, highly optimized set of instructions, rather than the more specialized set often found in other types of architecture, such as in a complex instruction set computer (CISC).
  - From https://en.wikipedia.org/wiki/Reduced_instruction_set_computer
- With over 180 billion ARM chips produced, as of 2021, ARM is the most widely used instruction set architecture (ISA) and the ISA produced in the largest quantity.
  - From https://en.wikipedia.org/wiki/ARM_architecture

Your tasks in this lab

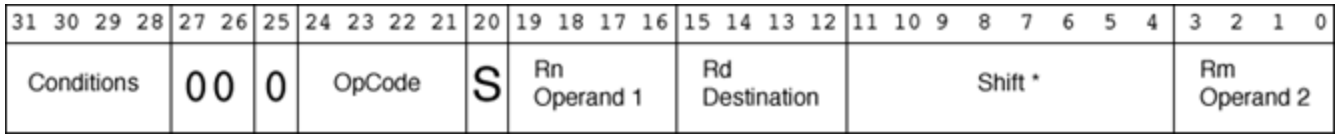
- Create a binary translator that turns a small subset of RISC-V instructions into ARM instructions.
- Implement a function to convert RISC-V I-Type instruction immediates into the immediate format required by ARM instructions.
- Implement a function for translating RISC-V registers into ARM registers as specified.

ARM Data-Processing Immediate Format



- Bits 27-25 and the **OpCode** combine to uniquely identify the instruction.
- **Rn** is the first register operand, similar to the RISC-V source register, and **Rd** is the destination register.
- The **Rotate** bits are used with the 8-bit **Immediate** field in order to obtain a 32-bit immediate.
- The **Conditions** and **S** bits aren't relevant until the next lab.

ARM Data-Processing Register Format



- The **`Shift`** field is used for shift instructions and allows for certain instructions to be combined with a shift.

- **`Rm`** is the second register operand, to the RISC-V target register.

Instruction Translation

- Input is RISC-V binary terminated by the sentinel word `0xFFFFFFFF`.
- Each word in the input represents a single RISC-V instruction.
- Parse each RISC-V instruction to find out which ARM instruction it should be translated into.
- Conver RISC-V registers and immediate fields (if applicable) into an appropriate format.
- Combine everything to obtain the translated instruction.

Register Translation

- The ARM architecture exposes 16 registers by default, one of which is the PC.
- This lab only translates 15 non-PC registers using the following mapping:

t0 (x5)	R0
t1 (x6)	R1
t2 (x7)	R2
s0 (x8)	R3
s1 (x9)	R4

s2 (x18)	R5
s3 (x19)	R6
s4 (x20)	R7
s5 (x21)	R8
s6 (x22)	R9

a0 (x9)	R10
a1 (x10)	R11
a2 (x11)	R12
sp (x2)	R13
ra (x1)	R14

Register Translation

- Write a function that translates a RISC-V register into an appropriate ARM register.
  - The RISC-V register is denoted by the number following the `x` in `x0`, `x1`, etc.
  - The ARM register is denoted by the number following the `R`.

Immediate Rotation

- In RISC-V, there is a single immediate field containing the value of the immediate.
- The ARM data-processing immediate format contains two fields of bits that determine the value of the immediate:
  - immediate field
  - rotation-bit field

Immediate Rotation

- The value of the immediate is obtained by shifting the immediate to the right by the value of the rotation field multiplied by 2.

- The following is an example of the above process and also appears as a GIF and a PDF in this lab's description.

Consider this ARM **ADD** instruction. The rotate bits, which are highlighted, are first multiplied by two.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	0	1	1	1	0	1	1	1

The instruction is equivalent to **`ADD r1, r1, 0x00000770`**

The rotate field (highlighted below), when multiplied by two, is equal to 28.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0

Now, to get the 32 bit immediate, the 8 bit immediate field from the instruction must be rotated 28 bits to the right with wraparound.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1

Now, to get the 32 bit immediate, the 8 bit immediate field from the instruction must be rotated 28 bits to the right with wraparound.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1

This 32 bit immediate will now be used for the **`ADD`** instruction in the first slide. The immediate is equal to **`0x00000770`**.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	0

Immediate Rotation

- This format does not exist in RISC-V.
- Write a function that converts a standard RISC-V immediate into the ARM data-processing immediate format.

Tips

- Use the provided test cases.
- Create your own tests for various edge cases.
- Start with the functions that translate RSIC-V registers and compute ARM rotate and immediate bits from RISC-V immediate fields.
- Follow the function's specification.
  - These functions are used in the next lab.

University of Alberta Code of Student Behavior

http://www.governance.ualberta.ca/en/CodesofConductandResidenceCommunityStandards/CodeofStudentBehaviour.aspx

30.3.2(1) Plagiarism

No Student shall submit the words, ideas, images or data of another person as the Student's own in any academic writing, essay, thesis, project, assignment, presentation or poster in a course or program of study.

30.3.2(2) Cheating

30.3.2(2) d No Student shall submit in any course or program of study, without the written approval of the course Instructor, all or a substantial portion of any academic writing, essay, thesis, research report, project, assignment, presentation or poster for which credit has previously been obtained by the Student or which has been or is being submitted by the Student in another course or program of study in the University or elsewhere

Overview

31 30 29 28	27 26	25	24 23 22 21	20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4 3 2 1 0
Conditions	00	1	OpCode	S	Rn Operand 1	Rd Destination	Rotate	Immediate Unsigned 8-bit Value

31 30 29 28	27 26	25	24 23 22 21	20	19 18 17 16	15 14 13 12	11 10 9 8 7 6 5 4	3 2 1 0
Conditions	00	0	OpCode	S	Rn Operand 1	Rd Destination	Shift *	Rm Operand 2

Now, to get the 32 bit immediate, the 8 bit immediate field from the instruction must be rotated 28 bits to the right with wraparound.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	0