CSM 61C and CALL Fall 2019

RISC-V Instruction Formats

Mentoring 4: October 7, 2019

1 Instruction Format Design

Prof. Wawrzynek decides to design a new ISA for his ternary neural network accelerator. He only needs to perform 7 different operations with his ISA: `xor`, `add`, `lw`, `sw`, `lui`, `addi`, and `blt`. He decides that each instruction should be 17 bits wide, as he likes the number 17. There are no `funct7` or `funct3` fields in this new ISA.

1.1 What is the minimum number of bits required for the opcode field?

$\lceil \log_2 7 \rceil = 3$

Binary encoding, which requires least number of bits, is used here. In order to represent 7 operations, we need at least $\lceil \log_2 7 \rceil = 3$ bits.

1.2 Suppose Prof. Wawrzynek decides to make the `opcode` field 6 bits. If we would like to support instructions with 3 register fields, what is the maximum number of registers we could address?

The instruction is 17 bit wide, 6 bits are used for `opcode`, we have 11 bits left for register indexing. Given we need 3 register fields, we can have $\lfloor \frac{11}{3} \rfloor = 3$ bits per register field which means we could address 8 registers.

1.3 Given that the `opcode` field is 6 bits wide and each register field is 2 bits wide in the 17 bit instruction, answer the following questions:

(a) Using the assumptions stated in the above description, how many bits are left for the immediate field for the instruction `blt` (Assume it takes `opcode`, `rs1`, rs2, and `imm` as inputs)?

$17 - 6 - 2 - 2 = 7$
`blt` has 1 `opcode` field (-6), 2 register fields $(-2 - 2)$, we can use the rest $17 - 6 - 2 - 2 = 7$ bits for expressing jump offset.

(b) Let n be your answer in part (a). Suppose that `blt`'s branch immediate is in units of instructions (i.e. an immediate of value 1 means branching 1 instruction away). What is the maximum number of bytes a `blt` instruction can jump forward from the current `pc` using these assumptions? Write your answer in terms of n.

$(2^{n-1} - 1) * 17$
In 2's complement, the range of an n-bit number is $[-2^{n-1}, 2^{n-1} - 1]$. jumping forward means that the offset is positive. With n-bit 2's complement offset, we can jump forward $2^{n-1} - 1$ instructions, which is $(2^{n-1} - 1) * 17$ bits since each instruction is 17 bits wide.

(c) Using the assumptions stated in the description, what is the most negative immediate that could be used in the `addi` instruction (Assume it takes `opcode`, `rs1`, `rd`, and `imm` as inputs)?

-64
First, calculate the bit width of the immediate field, which is $17 - 6 - 2 - 2 = 7$ bits. The range of a 7-bit number in 2's complement is $[-2^6, 2^6 - 1]$ Thus, the most negative immediate is $-2^6 = -64$.

(d) For LUI, we need `opcode`, `rd`, and `imm` as inputs. Using the assumptions stated in the description, how many bits can we use for the immediate value?

$17 - 6 - 2 = 9$
Given the `opcode` is 6 bits wide, register is 2 bits wide, we can use the rest of the bits for immediate. The width of immediate is therefore $17 - 6 - 2 = 9$.

2 RISC-V Behavior

2.1 For parts a and b, answer either Caller, Callee or Neither as applicable. The Caller is the function passing these values to a new call, the Callee is the function being called (ie. if we invoke a function doStuff() from main(), doStuff() is the Callee and main() is the Caller)

(a) Whose responsibility is it to save the return address (ra) in a function call?

Caller

(b) Whose responsibility is it to save the temporary registers (t0-t6)? What about the saved registers (s0-s11)?

Temp: Caller Saved: Callee

(c) You're given a new RISC-V instruction set where instructions are 12 bits long instead of 32 bits. Each rs and rd field uses 2 bits. How many registers does this new format support?

In regular RISC-V, we can represent 32 registers because our 5-bits for each rd/rs field allow us 2^5 possibilities. Here, we have only 2 bits for each rd/rs field and so we can represent $2^2 = 4$ registers

(d) If our opcode and funct3 are now two bits each, what is the largest immediate our I-Type instruction format can support? How far can we now branch (in bytes)?

For the I-Type: opcode + funct code + 2(reg field) = 2 + 2 + 2(2) = 8 bits already in use. This leaves 4 bits for our immediate with which we can represent from -8 to 7. For the B-type: opcode + funct code + 2(reg field) = 8 bits in use. Then there are 4 bits for our immediate. Recall that this immediate is then multiplied by 2 to determine how many bytes we are branching by, so anywhere from -16 to 14 bytes.

(e) Draw the format of the immediate instruction in the following box. Label each field and the number of bits each field holds.

Immediate (4) | r1 (2) | funct (2) | rd (2) | opcode (2)

3 CALL

3.1 The following are some multiple choice questions about CALL. Clearly circle the correct answer:

(a) A system program that combines separately compiled modules of a program into a form suitable for execution is _____.

A. Assembler
B. Loader
C. Linker
D. None of the Above

C

(b) At which point will all the machine code bits be determined for a *la* instruction?

A. C code
B. Assembly code
C. Object code
D. Executable

C

(c) At the end of the compiling stage, the symbol table contains the _____ of each symbol.

A. relative address
B. absolute address
C. the stack segment beginning address
D. the global segment beginning address

A

(d) beq and bne instructions produce _____ and they _____.

A. PC-relative addressing, never relocate
B. PC-relative addressing, always relocate
C. Absolute addressing, never relocate
D. Absolute addressing, always relocate

A

(e) jal and jalr instructions add symbols and _____ to _____.

A. instruction addresses, the symbol table
B. symbol addresses, the symbol table
C. instruction addresses, the relocation table
D. symbol addresses, the relocation table

C