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?

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?

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)?

(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.

(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)?

(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?

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?

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

(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?

(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)?

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

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

(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) 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

(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

(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