COMP.2030 LAB

1. Each byte sequence below indicates

address: Y86 encoding

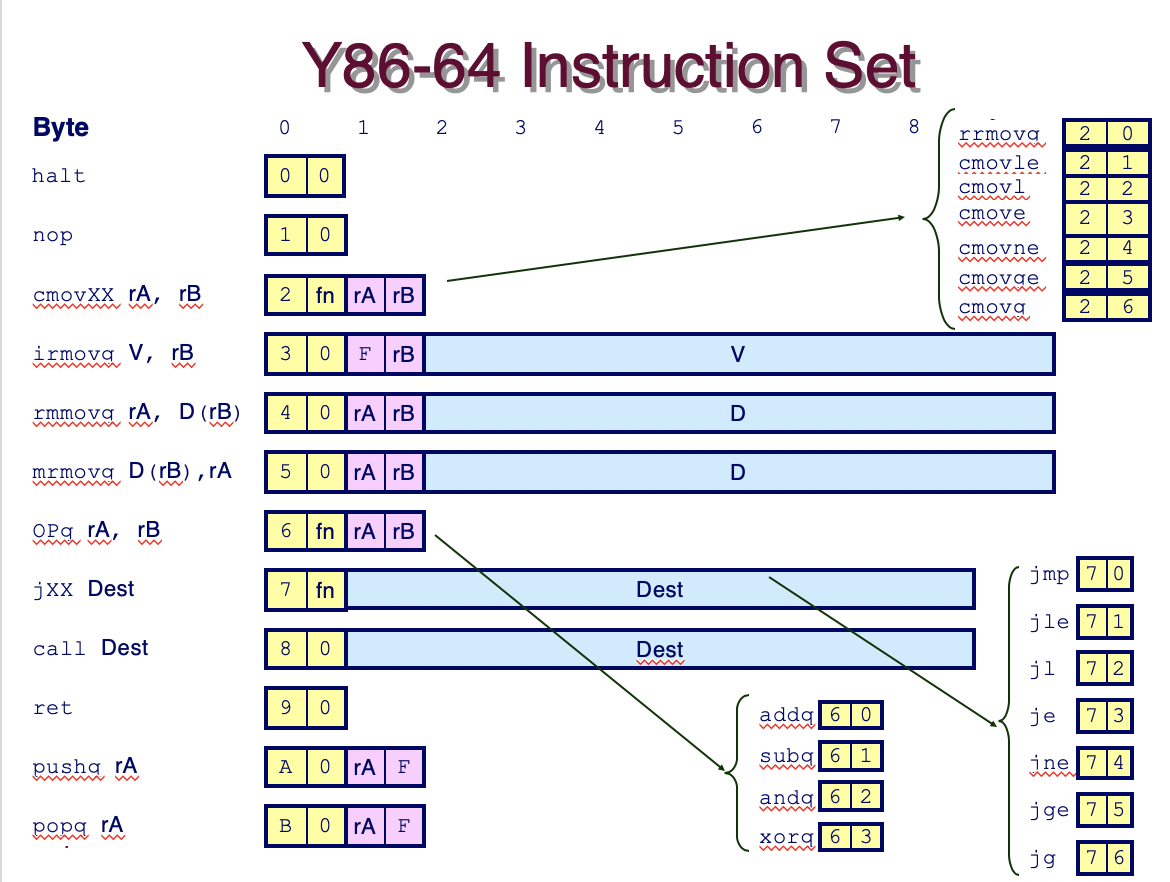
For each byte sequence below, determine the Y86 instruction sequence it encodes.

0x100: 30f3fcffffffffffffff4063000800000000000000

irmovq $-4, %rbx

rmmovq $rsi, 0x800(%rbx)

halt



0x200: 50540700000000000000b01f

mrmovq 7(%rsp),%rbp

popq %rcx

0x300: 611373000200000000000000

subq %rcx, %rbx

je 0x200

halt

0x400: 6362a00f

xorq %rsi, %rdx

pushq %eax

1. Y86 machine is a subset of x86 with the instruction set shown above right. Write byte encoding of the following Y86 instruction sequence, starting at address 0x100.

irmovq $15, %rbx

rrmovq %rbx, %rcx

loop: rmmovq %rcx, -3(%rbx)

addq %rbx, %rcx

jmp loop

1. Diagram, engineering drawing

   Description automatically generatedSuppose that the 1-bit ALU in Y86 is constructed as on the right, except that OR gate is replaced by an XOR gate so that it can perform add, sub, and and xor operations.

Also shown are three control signals: Ainvert, Binvert, and Operation.

The 2-bit Operation controls the multiplexer to select the appropriate input to the output of the ALU. The two bits are determined by the Y86 instruction iCode of 6 for OP instructions and iFun code.

Namely, a separate combinational logic circuit takes two bits (ifun1 and ifun0) for four OP instructions (0, 1, 2, and 3 for add, sub, and, and xor), and generates appropriate values for Operation. Write the truth table for the combinational logic with inputs of ifun1 and ifun0 and the output tied to Operation.

Operation bits as selector of the multiplexer determines which outputs from AND, XOR, and ADD circuits are to be passed to the output of the ALU.

ifun1 ifun0 Operation

-----------------------------------

0 0 2 (adder)

0 1 2 (adder)

0 0 0 (and)

0 0 1 (xor)