Quiz 6

0110 → 1001
+ 1
1010

CENG 351: Computer Architecture I
Midterm Exam Study Topics

1. Building Blocks
   a. Adder → part of the ALU / multibit adder
      i. Ripple-carry adder ⇒ different delay based on construction
   b. Subtractors
   c. Comparators → if A−B=0, A=B (use subtractor & zero flag)
   d. ALU → dedicated XNOR → AND
      i. Understand ALU control → Function bits
   e. Shifters and Rotators
      i. Logical vs arithmetic shift → copy MSB or shift in 0's; right shift F_2 inverts $F_{1:0}$ select mux output
      ii. Rotator vs shifter
   f. Counters
      i. N-bit counter design

left = × 2^N
right = ÷ 2^N

2. Arrays
   a. Memory Arrays
      i. 2^N word x M bit size → N address bits; depth/rows; M-bit word size
      ii. Bit cell organization → width/columns; word line; bit line
      iii. Reading/writing bit cells
   b. Memory types
      i. RAM/ROM → RAM is volatile / ROM is nonvolatile
      ii. SRAM vs DRAM → DRAM: capacitor for bit cell / slower / less space on chip; SRAM: cross-coupled bit cell / faster / more space on chip
      iii. Register files → multi ported memory
   c. Logic with Memory Arrays
      i. Lookup table → program mem. array w/ truth table

3. Instructions
   a. Assembly Language
      i. Instruction Set → instr. format & definition
      ii. MIPS Register Set → defined/reg-numbers
      iii. Byte-addressable memory → word addresses increment by 4
   b. Machine Language (translate)
      i. R-type → opcode | rs | rt | rd | shift | funct
      ii. I-type → opcode | rs | rt | imm
      iii. J-type → opcode | addr.
      iv. Fields of instruction: *op, rs, rt, etc.*.
      v. Decoding instructions → hexadecimal → bit fields do not line up w/ hex boundaries
   c. Instructions Programming (assembly tests opposite of high-level code condition)
      i. If → branch (beq/bne) not followed by a jump
      ii. If/else → beq/bne followed by a jump forward
      iii. While loops } beq/bne followed by a jump backward
      iv. For loops }
      v. Addressing modes → how we refer to operands

add rd, rs, rt
rd = rs + rt

4. Single-cycle Processor

1) register 2) immediate
3) memory imm(reg) → base + offset → $reg + imm.
4) PC-relative (branch)
5) pseudo-direct (jump) → 26 bits | 00; borrow from PC+4

immediate = # instructions to branch from PC+4

a. Datapath
    i. Architectural state → PC, reg. file, memories
    ii. Functional blocks
b. Control
    i. Control truth table
    ii. Main Decoder and ALU decoder
c. Adding instructions
    i. Update truth table and adjust datapath if necessary
d. Performance
    i. Performance calculation: real time to execute
    ii. Calculating the length of the clock cycle