

Quiz 6

$$\begin{array}{r} 0110 \rightarrow 1001 \\ + \quad 1 \\ \hline 1010 \end{array}$$

CENG 351: Computer Architecture I Midterm Exam Study Topics

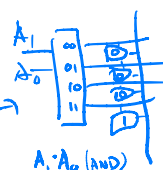
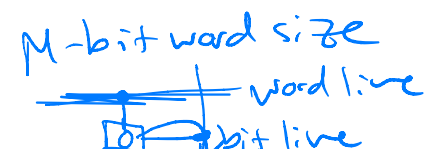
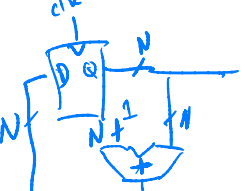
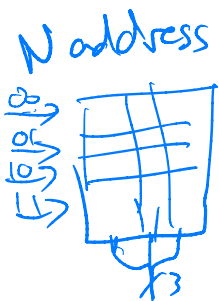
1. Building Blocks

- 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
 - i. Understand ALU control → dedicated XOR → AND
- e. Shifters and Rotators
 - i. Logical vs arithmetic shift → copy MSB or right shift F_2 inverts
 - ii. Rotator vs shifter
- f. Counters
 - i. N-bit counter design

left = $\times 2^N$
right = $\div 2^N$

2. Arrays

- a. Memory Arrays
 - i. 2^N word x M bit size
 - ii. Bit cell organization
 - 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
 - iii. Register files → SRAM: cross-coupled bit cell / faster / more space on chip
- c. Logic with Memory Arrays
 - i. Lookup table → multiplexed memory



3. Instructions

- a. Assembly Language
 - i. Instruction Set → instr. format & definition
 - ii. MIPS Register Set → defined/reg-numbers
 - iii. Byte-addressable memory
- b. Machine Language
 - 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 → bitfields do not line up w/ hex boundaries
- c. Instructions Programming
 - i. If → branch (beq/bne) not followed by a jump
 - ii. If/else → beq/bne followed by a jump forward
 - iii. While loops
 - iv. For loops
 - v. Addressing modes → how we refer to operands

assembly tests opposite condition of high-level code

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 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