A-type (no immediate): 5bits op code, 4bits registers

// 0 < n < 4, can only specify (5 – n) opcodes

// Example: 5bits for op code, 2bits for registers, 2bits for immediate

B-type (opcode, register, immediate): less than 5 - n bits op code, 4 bits registers, n-bits immediate

// 0 < n < 3, can only specify (4 – n) registers

// Example: 5bits for op code, 2bits for registers, 2bits for immediate

C-type (opcode, register, immediate): 5 bits op code, 4 - n bits registers, n-bits immediate

// 0 < x < 4, 0 < y < 3, n = x + y, can only specify (5 – n) opcodes, and (4 – n) registers

// Example: 3bits for op code, 3bits for registers, 3bits for immediate

D-type (opcode, register, immediate): 5 - x bits op code, 4 - y bits registers, n - bits immediate

// no bits for registers can only imply accumulator.

// Example: add #1 🡺 means add 1 to accumulator

E-type (opcode, immediate, no register-only accumulator): Number of bits for opcode and immediate depends on what we need

***PROBLEM***: we can't change the number of bits to represent registers or opcodes.

For example if our ISA uses 8 op codes total: add, shift, load, store add1, shift1, load1, store1, and #17 uses only 4 opcodes: add, shift, load, store. Then in problem #17 we always need to encode our opcodes in 3 bits even if we only need 2bits to represent 4bits. ☹