1- Consider a 1-address CPU that has a memory unit with 128K words of 32 bits each. An instruction is stored in one word of memory. The instruction format is divided into four fields: opcode field, 2-bit addressing mode field that specify direct, indirect, indirect with pre-decrement, or indirect with post-increment addressing mode, a register field that specifies one of 32 registers, and an address field. For your information, given an address

- Direct addressing is where the operand is located in $M[address]$. 
- Indirect addressing is where the operand is located in $M[M[address]]$.
- Indirect addressing with pre-decrement is where the operand is located in $M[M[address]-1]$.
- Indirect addressing with post-increment is where the operand is located in $M[M[address]+1]$.

(a) What is the maximum number of opcodes that can be incorporated into the CPU? How many bits are in the opcode field, the register field, and the address field? Draw the instruction format and indicate the number of bits in each field.
(b) How many bits are in the registers PC, MAR, MDR, IR, and AC?

2- Consider the internal structure of a pseudo-CPU discussed in class. Suppose the pseudo-CPU can be used to implement the hypothetical instructions given below (Actually, these are instructions from PDP-8, which is the first commercially successful minicomputer from the 60’s). Give the sequence of microoperations required to implement the Fetch and Execute cycle for each of the instructions shown below. Your solution should result in minimum number of microoperations. Assume PC is currently pointing to the instruction and only PC and AC have the capability to increment itself.

(a) Logical AND: $AND \ Y \ ; \ AC \leftarrow AC \land M(Y)$
(b) Increment and skip if zero: $ISZ \ Y \ ; \ M(Y) \leftarrow M(Y) + 1, \ If(M(Y)+1=0) \ Then \ PC \leftarrow PC +1$
(c) Deposit and clear the accumulator: $DCA \ Y \ ; \ M(Y) \leftarrow AC, \ AC \leftarrow 0$
(d) Jump to subroutine: $JMS \ Y \ ; \ M(Y) \leftarrow PC, \ PC \leftarrow Y + 1$

3- Consider the following hypothetical 1-address assembly instruction called “Store Accumulator Indirect with Pre-Decrement” of the form

$STA -(x) \ ; \ M(x) \leftarrow M(x)-1, \ M(M(x)) \leftarrow AC$

Suppose we want to implement this instruction on the pseudo-CPU discussed in class augmented with a temporary register TEMP. An instruction consists of 16 bits: A 4-bit operation code and a 12-bit address. All operands are 16 bits. PC and MAR each contain 12 bits. AC, MDR, and TEMP each contain 16 bits, and IR is 4 bits. Give the sequence of microoperations required to implement the Execute cycles for the above $STA -(x)$ instruction. Your solution should result in minimum number of microoperations. Assume PC is currently pointing to the STA instruction and only PC and AC have the capability to increment/decrement itself. Fetch cycle is given below
4. Based on the initial register and data memory contents shown below (represented in hexadecimal), show how these contents are modified (in hexadecimal), including N, Z, and C bits of the Status register (SR), after executing each of the following AVR assembly instructions. Instructions are unrelated.

(i) MOV R1, R28
(ii) LD R4, Y+
(iii) LDI R4, 33
(iv) MUL R2, R3
(v) ROL R3

[20 pts]

5. Consider the following code written in AVR assembly. Explain in words what the program accomplishes when it is executed. That is, explain what it does, how it does it, and how many times it does it. What is the value of location CTR when the execution completes?

```
.ORG 0x000F
LDI XH, high(CTR)
LDI XL, low(CTR)
LDI R31, 0xf0
CLR R5
LOOP:
CLC
ROL R31
BRCC SKIP
INC R5
SKIP:
CPI R31, 0x00
BRNE LOOP
ST X, R5
DONE:
JMP DONE
.DSEG
CTR: .BYTE 1
```

[20 pts]