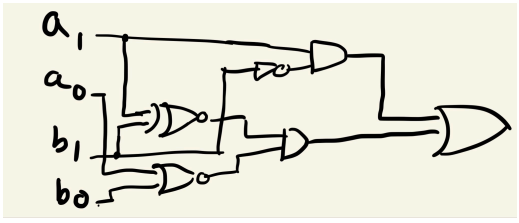
Allen Boyce, Ahmad Aldousari

(Double check 1 and 2)

1.
    a.
        i. X := A or (B and C)
        ii. Y := not-A or C
    b.
        i. X := ((A nand A) nand ((B nand C) nand (B nand C)))
        ii. Y := ((A nand A) nand (A nand A)) nand (C nand C)
2.
    a. 

3.
    a. AND 0xAAAAAAAA
    b. OR 0x00000007
    c. AND 0x00000007
    d. AND 0xAAAAAAAA
    e. MOV 0xFFFFFFFF
    f. XOR 0xCOOOOOOO
    g. AND 0xFFFFFFF8
4.

a.	C0000003		JMP	start
b.	00000000	num:	0	
c.	000000FF	lim:	255	
d.	00000001	start:	LOAD	num
e.	30000008		WRITE	00000008
f.	40000001		ADD	00000001
g.	10000001		STORE	num
h.	50000002		SUB	lim
i.	D0000009		JZ	end
j.	C0000003		JMP	start
k.	C0000009	end:	JMP	end

5. C0000003 00000000 000000FF 00000001 30000008 40000001 10000001 50000002 D0000009 C0000003 C0000009
6.

```
        JMP     start
n1:     0
n2:     0
start:  READ    00000100
        STORE   n1
        READ    00000100
        STORE   n2
gcd:    LOAD    n1
        JZ      write
        MOD     n2
        STORE   n1
        JMP     gcd
write:  LOAD    n2
        WRITE   00000200
end:    JMP     end
```

7.

```
        temp    0
        STORE   temp
        LOAD    000030AA
        STORE   000030AA
        LOAD    temp
```

8.

```
JLZ         skipto
JMP         0837BBE1
skipto:
```

9.

a. r8 and r9 swap values
b. When XOR a,b, a takes on a kind of inversion of b.