Homework 7

Y86 Assembly

Write down the byte codes of the following y86 instructions

Y86 Instructions	Byte Codes
irmovl $0x20, %ebx	30 f3 20
jmp 0x1314	70 14 13 00 00
call 0x2323	80 23 23 00 00
jne 0x4312	74 12 43 00 00
popl %ebx	b0 3f (or b0 38)
mrmovl 0x30(%edx), ebx	50 32 30 00 00 00

Y86 Byte Code

We now have a piece of Y86 code. Please answer the value of %eax when the function ends, and give a description about what the function has done. %eax = 34 (or 0x22), The function will use 0x30 to sub the sum of a[0], a[1], a[2], a[3].

C-like Code is :

int a[5] = {2, 3, 4, 5, 6};

int b = 4;

fun(a, b);

Y86 Byte Code:

0x042: a05f

0x044: 2045

0x046: 501508000000

0x04c: 50250c000000

0x052: 30f030000000

0x058: 6222

0x05a: 737c000000

0x05f: 506100000000

0x065: 6160

0x067: 30f304000000

0x06d: 6031

0x06f: 30f3ffffffff

0x075: 6032

0x077: 745f000000

0x07c: 2054

0x07e: b05f

0x080: 90

HCL

We have introduced HCL in class to design combinatorial logic circuit. Try to write HCL code for the following circuit: Take A, B, C and D as input, output the min value of them.

E.g. Min(1,2,3,4) = 1.

```
int Min = [
        A <= B && A <= C && A <= D: A;
        B <= A && B <= C && B <= D: B;
        C <= A && C <= B && C <= D: C;
        1                          : D;
];
```

2. We now have an expression: (a, b, c are all integers)

bool out = (a || !a) || (b && c)

Please explain the difference between C expression and logical expression when they execute the expression above.

1. in C a, b and c can be arbitrary integers, and interpret 0 as FALSE while others as TRUE. Logical gates only use 0 and 1
2. C has partial evaluation which means it will not calculate the b&&c because a || !a will always be true.