

ICS Home 1 Solution

February 23, 2023

1 HCL

Please write down the HCL expressions for the following signals.

Example: Show if the two input signals **a** and **b** are equal

```
bool eq = (a&&b) || (!a && !b);
```

1. The HCL expression for a signal **nand**, which is equal to **NAND** of inputs **a** and **b**, the truth table is given, and you should only use **NOT** (!) and **OR** (||) operators.

NAND	0	1
0	1	1
1	1	0

Solution: `bool nand = !a || !b;`

2. The HCL expression for a three-way xor called **XOR3**. **If and only if all the inputs are the same, output will be true.** Each input and output is one-bit wise. The three input signals are **a**, **b** and **c**. Hint: You can use boolean expressions or case expressions.

Solution1: `bool xor3 = (A && B && C) || (!A && !B && !C)`

Solution2:

```
1  xor3 = [  
2    A && B && C : 1;  
3    A ^ B      : 0;  
4    A ^ C      : 0;  
5    B ^ C      : 0;  
6    1          : 1;  
7  ]
```

2 Y86

```

0x000:                                     | .pos 0
0x000:                                     | init:
0x000: 30f400020000000000000000         |   irmovq stack, %rsp
0x00a: _____[1]_____             |   irmovq stack, %rbp
0x014: 801e0000000000000000             |   call main
0x01d: 00                               |   halt
0x01e:                                     | main:
0x01e: 30f700030000000000000000         |   irmovq list, %rdi
0x028: 30f603000000000000000000         |   _____[2]_____
0x032: 803c00000000000000000000         |   call calculate
0x03b: 90                               |   ret
0x03c:                                     | calculate:
0x03c: 6300                             |   _____[3]_____
0x03e: 30f308000000000000000000         |   irmovq $8, %rbx
0x048: 2072                             |   rrmovq %rdi, %rdx
0x04a: 706d00000000000000000000         |   jmp test
0x053:                                     | loop:
0x053: 50120000000000000000000000         |   _____[4]_____
0x05d: 6010                             |   addq %rcx, %rax
0x05f: _____[5]_____             |   andq %rsi, %rax
0x061: 6032                             |   addq %rbx, %rdx
0x063: 50220000000000000000000000         |   mrmovq (%rdx), %rdx
0x06d:                                     | test:
0x06d: 6222                             |   andq %rdx, %rdx
0x06f: 745300000000000000000000         |   jne loop
                                           |
0x200:                                     |   _____[6]_____
0x200:                                     | stack:
                                           |
0x300:                                     | .pos 0x300
0x300:                                     | .align 8
0x300:                                     | list:
0x300: 080000000000000000000000         |   .quad 0x8
0x308: 200300000000000000000000         |   .quad _____[7]_____
0x310: 090000000000000000000000         |   .quad 0x9
0x318: 000000000000000000000000         |   .quad 0x0
0x320: 0a0000000000000000000000         |   .quad 0xa
0x328: 100300000000000000000000         |   .quad 0x310

```

1. Please fill in the blanks within above Y86 binary and assembly code.

Solution:

```

[1]30f500020000000000000000   [2]irmovq $3, %rsi
[3]xorq %rax, %rax             [4]mrmovq (%rdx), %rcx
[5]6260                         [6].pos 0x200
[7]0x320

```

2. Please calculate the value of **%rax** after the program **HALT**.

%rax = 0x3