CSC258 Mid-Term Test solutions 2015 fall

1[12] Prove or disprove the binary (boolean) expression

$$(a{\le}b) \land (c{\le}d) \;\le\; (a{\land}c \le b{\land}d)$$

"Prove" means show that its value is always $\top$, and "disprove" means show that its value isn't always $\top$.

§ $\quad (a{\le}b) \land (c{\le}d)$ — using generalization, weaken each conjunct

$\le \quad ((a{\le}b) \lor (c{\le}b)) \land ((a{\le}d) \lor (c{\le}d))$ — use antidistribution in each conjunct

$= \quad (a{\land}c \le b) \land (a{\land}c \le d)$ — use distribution

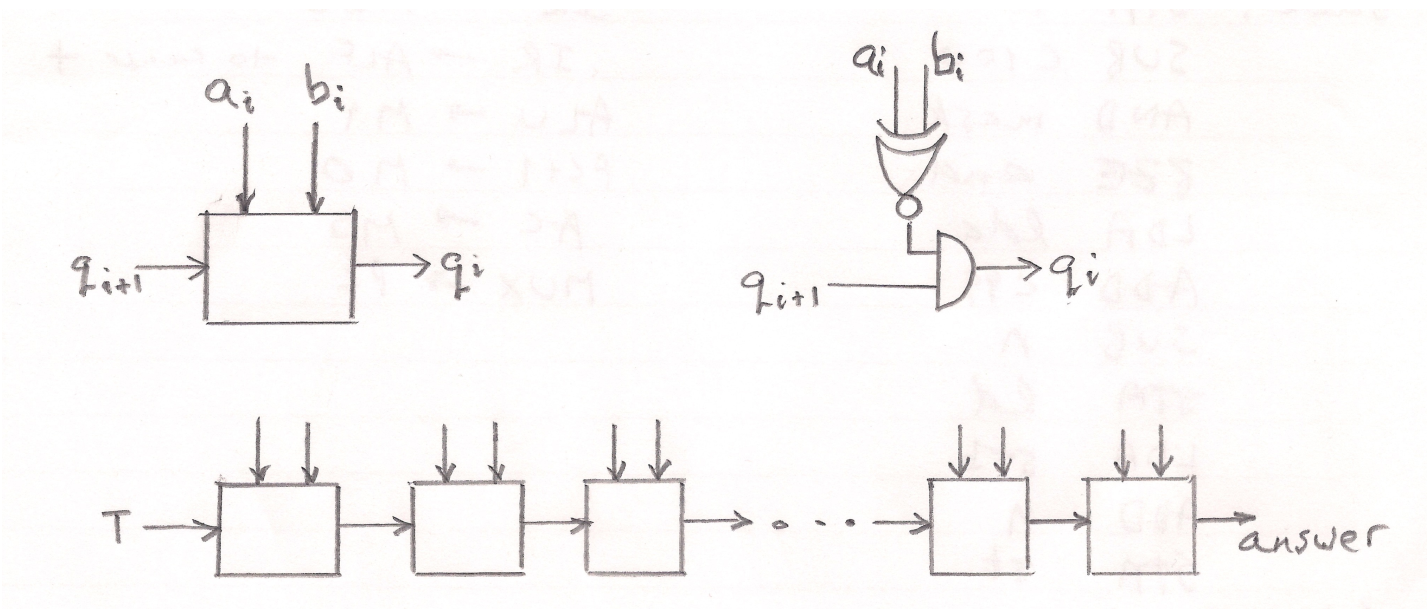
$= \quad (a{\land}c \le b{\land}d)$

Another way to prove it is by value tables.

a	b	c	d	$(a$	$\le$	$b)$	$\land$	$(c$	$\le$	$d)$	$\le$	$(a$	$\land$	c	$\le$	b	$\land$	$d)$
$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
$\top$	$\top$	$\top$	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\top$	$\bot$	$\bot$
$\top$	$\top$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$
$\top$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$
$\top$	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\top$
$\top$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$
$\top$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\top$
$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\top$	$\top$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$
$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$
$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$
$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\top$
$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\bot$
$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\top$
$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$

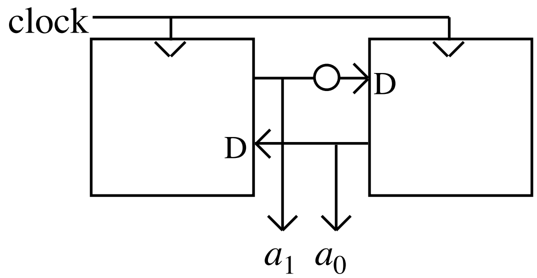
2[12] Design a circuit that compares two n-bit 2's complement numbers for equality. There are $2 \times n$ bits of input $a_{n-1}\, a_{n-2} \ldots a_2\, a_1\, a_0$ and $b_{n-1}\, b_{n-2} \ldots b_2\, b_1\, b_0$ representing two numbers a and b in two's complement representation. There is one bit q of output that's $\top$ just when a and b are equal. You can use any gates with two or fewer inputs.

§ First I define a "one-bit" comparator. It has 3 bits of input and 1 bit of output. There are n of these "one-bit" comparators, and they form a chain. Two of the input bits are a_i and b_i. Input q_{i+1} and output q_i each mean "all previous pairs of bits of a and b were equal". We feed $\top$ in one end of the chain, and the answer comes out the other end.



The chain could just as well go the other way.

3[9] Here is a sequential circuit composed of two D flip-flops and a NOT gate.



What pattern of output numbers occurs as the clock pulses?

§ The pattern is $a_1\,a_0 =$ 00, 01, 11, 10, 00, 01, 11, 10, ... repeating cyclicly. Or if you prefer, a = 0, 1, 3, 2, 0, 1, 3, 2, ... The cycle might start with any of these numbers.

4[12] Express the number $-10^1/_6$ (minus ten and a sixth) in IEEE single-precision floating-point representation.

§ 6 in binary is 110. To find 1/6 divide 1 by 6.

```
          .0 0 1
110 ) 1.0 0 0 ...
        1 1 0
          1 0 0
```

so 1/6 is $.0\overline{01}$

$-10^1/_6$ in binary

= $-1010.0\overline{01}$ normalize

= $-1.01000\overline{01} \times 2^{130-127}$ and 130 in binary is 10000010
so in IEEE floating-point

= 1 10000010 01000101010101010101010
or the final bit could be 1 due to rounding