Real-Time Imaging and Control

Practice

MSCV/ESIREM

Antoine Lavault
antoine.lavault@u-bourgogne.fr

Basics 1 - Boolean Logic

Problem 1

Remark 1. *Reminder on hexadecimal and two's complement:*

- *A 4-bit word can be converted to hexadecimal easily.*
- *Two's complement on n bits: if $m < 0$ and $|m|$ can be expressed on $n-1$ bits, m can be encoded as the unsigned integer $2^n - m$. This is useful to get the negative of a known positive number m*
- *When dealing with negative binary numbers encoded in two's complement, we can interpret it as a sign-bit concatenated to a $(n-1)$ bit word representing $2^{n-1} - |m^\star|$ where $m^\star$ is the number $|m|$ with the MSB removed. Ex :* $1000 = -2^{4-1} + 0b000 = -8 + 0; 1010 = -2^{4-1} + 0b010 = -6$

1. Give decimal values of those unsigned integers: 1001 1011, 0011 1100, and 0101 0101.
2. Same question but with signed integers in two's complement.
3. Give the binary values of the decimal -100, 83, and -29 (in two's complement).

1. $0x9b = 155$, $0x3c = 60$,$0x55 = 85$
2. -101, the other two do not change
3. $0b1001.1100$, $0b0101.0011$, $0b1110.0011$

Problem 2

This exercise will familiarize you with handling IEEE 754 floating point numbers. This exercise will assume a single precision float (i.e., 32-bit long).

1. Write the value of the lowest normal number (in absolute value).
2. Write the value of the lowest denormal (or subnormal) number (in absolute value).
3. Write the value of the highest normal number (in absolute value).
4. Given the bit string $0x40500000$, what is the encoded float value and its decimal value?
5. Given the bit string $0x80600000$, what is the encoded float value and its decimal value? How does it compare to the result of question 2?
6. Ariane flight V88/501: The rocket misbehaved with a too-sharp attack angle due to a conversion error from a 64-bit float (double precision) to a 16-bit integer in the inertial navigation system, causing it to explode with its payload. What kind of error happened to make such an event occur?

1. Exponent = $0x01$, Mantissa, = $"0x000000$, Sign = whatever.
   We get $2^{1-127} * 1.0 \approx 1.1754943508222875 08e - 38$.
2. Exponent = $0x00$, Mantissa, = $"0x000001$, Sign = whatever. We get $2^{0-127} * (2^{-22}) = 2^{-126} * 2^{-23} \approx 1.4e - 45$.
3. Exponent = 0xfe, Mantissa = $0x7fffff$. We then have $2^{254-127} * (1 + \underbrace{\frac{0x7fffff}{2^{23}}}_{1-2^{-23}=1.99999988079}) = 3.402823466e38$.
4. The bit string is first converted to binary: 0100 0000 0101 0000 0000 0000 0000 0000, then reorganized to match the single-precision format : 0 10000000 10100000000000000000000.

   The sign bit is 0, the exponent is $2^7 = 128$, and the mantissa is $2^{-23}(2^22 + 2^20) = 1/2 + 1/8 = 5/8$. The number in decimal is $2^{128-127} * (1 + 5/8) = 2 * 1.625 = 3.25$
5. Same idea here. The number is, once "formatted," 1 00000000 11000000000000000000000. It is a negative (sign bit 1), denormal (exponent 0) number.

   Its value is then: $-2^{-127} * (1 + 1/2) = 8.8162076e - 39$.
6. 16-bit integers encode way fewer numbers than double-precision floats. What happened is that the conversion caused an overflow, which then threw an exception in the system, and everything went ballistic.

Problem 3

Remark 2 (Dual form of a logic expression)**.** *We can define the dual of an expression e as the expression built with all the bits negated and sums and products inverted.*
Example: $x\bar{y} + \bar{x}z \rightarrow (\bar{x} + y)(x + \bar{z})$
Notably, if two expressions, e and f, are equal, their duals are also equal.

Simplify each expression with algebraic manipulation. when applicable, $f(a, b, c) = a + b + c$.

1. $a + 0$
2. $\bar{a} \cdot 0$
3. $a + \bar{a}$
4. $a + a$
5. $a + ab$
6. $a + \bar{a}b$
7. $a \cdot (\bar{a} + b)$
8. $ab + \bar{a}b$
9. $(\bar{a} + \bar{b}) \cdot (\bar{a} + b)$
10. $a \cdot (a + b + c + \cdots)$
11. $f(a, b, ab)$
12. $f(a, b, \bar{a}\bar{b})$
13. $f(a, b, \overline{ab})$
14. $a + a\bar{a}$
15. $ab + a\bar{b}$
16. $\bar{a} + \bar{a}b$
17. $(d + \bar{a} + b + \bar{c})b$
18. $(a + \bar{b})(a + b)$
19. $d + (d + da)$
20. $a(a + ab)$
21. $\overline{(\bar{a} + \bar{a})}$
22. $\overline{(a + \bar{a})}$
23. $d + d\bar{a}bc$
24. $\bar{d}\overline{(dabc)}$
25. $ac + \bar{a}b + bc$
26. $(a + c)(\bar{a} + b)(c + b)$
27. $\bar{a} + \bar{b} + ab\bar{c}$

1. $a + 0 = a$
2. $\bar{a} \cdot 0 = 0$
3. $a + \bar{a} = 1$
4. $a + a = a$
5. $a + ab = a$
6. $a + \bar{a}b = (a + \bar{a})(a + b) = a + b$ (use the dual). An alternative is $a = \bar{a}b = a(1 + b) + \bar{a}b = a + (a + \bar{a})b = a + b$
7. $a \cdot (\bar{a} + b) = ab$
8. $ab + \bar{a}b = b$
9. $(\bar{a} + \bar{b}) \cdot (\bar{a} + b) = \bar{a}$
10. $a \cdot (a + b + c + \cdots) = a$
11. $f(a, b, ab) = a + b$
12. $f(a, b, \bar{a}\bar{b}) = 1$
13. $f(a, b, \overline{ab}) = 1$
14. $a + a\bar{a} = a$
15. $ab + a\bar{b} = a$
16. $\bar{a} + \bar{a}b = \bar{a}$
17. $(d + \bar{a} + b + \bar{c})b = b$
18. $(a + \bar{b})(a + b) = a$
19. $d + (d + da) = d$
20. $a(a + ab) = a$
21. $\overline{(\bar{a} + \bar{a})} = a$
22. $\overline{(a + \bar{a})} = 0$
23. $d + d\bar{a}bc = d$
24. $\bar{d}\overline{(dabc)} = \bar{d}$
25. $ac + \bar{a}b + bc = ac + \bar{a}b$
26. $(a + c)(\bar{a} + b)(c + b) = (a + c)(\bar{a} + b)$ with the dual of the consensus theorem
27. $\bar{a} + \bar{b} + ab\bar{c} = \bar{a} + \bar{b} + \bar{c}$ (use the dual).

For number 27, the direct method would be $\bar a + \bar b + ab\bar c = \overline{ab}(1 + \bar c) + ab\bar c = \overline{ab} + (ab + \overline{ab})\bar c = \bar a + \bar b + \bar c$

Problem 4

Use De Morgan laws to simplify the following :

1. $\overline{\overline{(\bar a + c)} \cdot \overline{(b + c)}}$
2. $\overline{ab\bar c}$
3. $\overline{b + \bar c} \cdot \overline{c + \bar a} \cdot \overline{\bar a + \bar b}$

1. $\overline{\overline{(\bar a + c)} \cdot \overline{(b + c)}} = \overline{\overline{(\bar a + c)}} + \overline{\overline{(b + c)}} = \bar a + b + c$
2. $\bar a + \bar b + c$
3. 0

Problem 5

Remark 3. *In a Karnaugh map, there are two possible groupings :*

- *One with the 1, called "Minimum Sum of Products" (MSP)*
- *One with the 0, called "Minimum Product of Sums"(MPS)*

Note: In the slides, the notation was MSB in the rows and LSB in the columns. The other notation (with MSB and LSB transposed) will be used in the following Karnaugh maps.

c \ ab	00	01	11	10
0	0	X	X	1
1	1	1	1	X

1. In the Karnaugh map above, find the MSP. A careful choice for the Do Not Care ("X") values is advised.
2. Same question with the MPS.
3. Are the equations equal?

Both are equal to $a + c$

Problem 6

Find the logic equations described by the Karnaugh maps. Empty cells mean 0.
It is highly suggested to use colored pencils to circle the groupings.

1.

cd \ ab	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11		1	1	
10		1	1	

2.

cd \ ab	00	01	11	10
00	1			1
01		1	1	
11		1	1	
10	1			1

3.

cd \ ab	00	01	11	10
00			1	
01	1		1	1
11	1	1	1	1
10			1	

4.

cd \ ab	00	01	11	10
00		1		1
01	1		1	1
11		1		1
10	1	1	1	1

1. $b + \bar{c}$
2. $bd + \bar{b}\bar{d}$
3. $cd + \bar{b}d + ab$
4. $a\bar{b} + c\bar{d} + \bar{a}bc + a\bar{c}d + \bar{a}b\bar{d} + \bar{b}\bar{c}d$

Problem 7

Simplify the following expressions with a Karnaugh map. It is suggested to start with a truth table first.

1. $(a + \bar{b} + \bar{c}) \cdot (\bar{c} + (a + b + d) \cdot (\bar{a} + \bar{b} + \bar{d}))$
2. $(\bar{c} + ab)(\bar{c} + (a + \bar{d})(b + \bar{d}))(\bar{c} + (a + \bar{b})(b + \bar{d}))$

3. $\bar{w}y + w\bar{x}y + \bar{w}x\bar{z}$

1. $\bar{c} + \bar{b}d + a\bar{d}$
2. $ab + (\bar{b}\bar{c}\bar{d})$
3. $\bar{w}x\bar{z} + \bar{x}y + \bar{w}y$

Problem 8

Four people can access a safe: Fabio, Joaquin, Raphaël, and David.
Since they do not have the same role, some rules have been set :

- Fabio can open the safe if Joaquin or Raphaël are present
- The others can open the safe if two other persons are present.

What is the boolean equation for the safe? The binary variables used as inputs are "x is present."
$a(b + c) + bad + cbd + dbc$

Problem 9

Remark 4. *The NAND operator is functionally complete, i.e. $\vee$, $\wedge$, and $\neg$ can only be expressed with the NAND operator.*

1. Write the truth table of the NAND operator. As a reminder, a NAND $b = \overline{a \cdot b}$.
2. Write the operator $\vee$, $\wedge$ and $\neg$ in terms of NAND.
3. Propose a logic gate architecture implementing the OR, AND, and NOT gates in terms of NAND gates.

If we write NAND as $\uparrow$, we have :

- $\neg a = a \uparrow a$
- $a \wedge b = \neg(a \uparrow b) = (a \uparrow b) \uparrow (a \uparrow b)$
- $a \vee b = \bar{a} \uparrow \bar{b}$

Problem 10

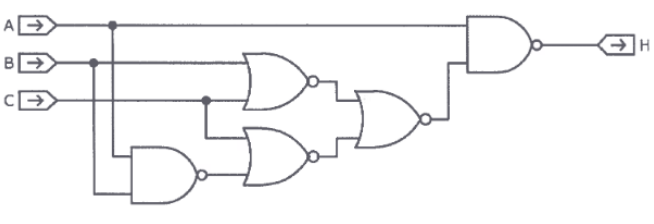

Figure 1: An arrangement of logic gates

1. Derive the truth table of the circuit shown in figure 1.
2. Derive a simplified boolean expression from the truth table.

   By any means necessary.

$\bar{a} + \bar{c}$

Problem 11

A drug dealer wrote what is shown in figure 2 before getting gunned down in a shooting. The police need the help of MSCV and ESIREM students to understand what was written.

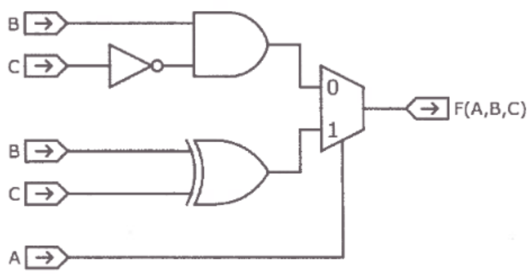

Figure 2: Unknown gibberish.

1. Derive the truth table of the circuit shown in figure 2
2. Derive a simplified boolean expression from the truth table. By any means necessary.

$a\bar{b}c + b\bar{c}$