PRCMP PL03
Operações lógicas e bit-a-bit

May 2023

1 Boolean algebra

Questions

1. Determine the truth table for each of the following logic sentences.
   (a) $Z = X \cdot Y$
   (b) $Z = X + Y$
   (c) $Z = (X \cdot Y) + X$
2. Assuming that $\overline{A}.B = 0$, $A + B = 1$ and $\overline{A} = B$ are given, what are the logic values of A and B?
   Hint: Build a truth table for the first two logical statements.
3. Consider the following statements,
   A1 – The weather is awful if it rains and it's cold.
   A2 – The weather is poor if it rains or is cold.
   A3 – The weather is acceptable if it rains but not cold, or vice versa.
   A4 – The weather is fine if it doesn't rain and it's not cold.
   A5 – The weather is dry if it doesn't rain.
   and these two propositional variables:
   R – It's raining.
   C – It's cold.
   (a) Represent the five statements in the algebraic form, using the two propositional variables.
   (b) Write the truth table for each of the logic sentences.
   (c) What is the opposite of $A2$?
4. Determine the truth table for each of these logical functions.
   (a) $F = \overline{X} + Y$
   (b) $F = X + \overline{(X + Z)}$
   (c) $F = (X + W) \cdot (X + Y)$
   (d) $F = \overline{\overline{(X + Y)} + Z}$
   (e) $F = \left[\overline{X} \cdot (Y + Z)\right] + (X \cdot Y)$
5. Prove by using truth tables that the following equivalence relations hold.
   (a) $\overline{X + Y} = \overline{X} \cdot \overline{Y}$
   (b) $\overline{X \cdot Y} = \overline{X} + \overline{Y}$

(c) $\overline{X \cdot Y \cdot Z} = \overline{X} + \overline{Y} + \overline{Z}$

(d) $X \cdot (Y + Z) = (X \cdot Y) + (X \cdot Z)$

(e) $X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$

(f) $(\overline{X} \cdot Y) + (\overline{Y} \cdot Z) + (X \cdot \overline{Z}) = (X \cdot \overline{Y}) + (Y \cdot \overline{Z}) + (\overline{X} \cdot Z)$

2 Bitwise logic operations

From this section on, we will use a popular notation (in the computer literature) for decimal, binary, octal and hexadecimal numbers.

Decimal: A decimal number is written without a prefix. *E.g.* `31` meaning *thirty one*.

Binary: A binary number uses the `0b` prefix. *E.g.* `0b1001` meaning $1001_{(2)}$.

Hexadecimal: An hexadecimal number is written with the `0x` prefix. *E.g.* `0xF04` meaning $F04_{(16)}$.

Octal: An octal number uses the `0` prefix. *E.g.* `0644` meaning $644_{(8)}$.

Questions

6. Determine the result for each of the bitwise operations.
   (a) `0b10110010 AND 0b11100111`
   (b) `0x55AA AND 0xAA55`
   (c) `0b10011010 OR 0b11001100`
   (d) `NOT 0x105A5A`
   (e) `0732 OR 0b101101101`
   (f) `0365 XOR 0222`
7. Assume we have a 16-bit processor operating 16-bit words. What is the result of each of the following operations?
   (a) `NOT 0`
   (b) `24 OR 59`
   (c) `0x4F1 AND 0x5287`
   (d) `0655 XOR 037`
8. It is required to flip (*i.e.* to invert) the values of bits 1, 3 and 15 in the 16-bit word `0x73A5`.

   Hint: Remember that the least significant bit (the rightmost bit) is in order 0 *(zero); the orders of bits grow from the rightmost to the leftmost bit (also known as the most significant bit).*
   (a) Which bitwise operation and bit mask would you select for this task?
   (b) Determine the result of the operation.
9. Assume that in expression *n op m*

   n – is an arbitrary unsigned 8-bit number,

   op – is one bitwise operation, and

   m – is an 8-bit mask.

   select *op* and *m* such that the result of *n op m*...
   (a) ... is the remainder of the integer division of *n* by 4.
   (b) ... is *one* if and only if *n* is an odd number.

(c) ... is *zero* if and only if *n* is equal to 5.

(d) ... is *zero* if and only if *n* is less than 32.

(e) ... flips the values of the most and least significant bits.

(f) ... is *not zero* if *n* has either of two lowest bits set.

(g) ... sets the lowest 4 bits of *n*.

(h) ... clears all odd bits in *n*.

(i) ... sets all even bits in *n*.

3 Bitwise shifts

Processors allow us to move a word in a register by arbitrary amounts of bits, by bitwise shift operations. On these operations, the bits moved out from the register are lost. However, the positions that become vacant need to be filled with fresh bits.

The three operations we are going to cover are:

- logical left shift (`SLL` – shift left logical)
- logical right shift (`SRL` – shift right logical)
- arithmetic right shift (`SRA` – shift right arithmetic)

The logical shifts always insert zeroes in the vacant positions. For example, let us shift the 8-bit word `0b00010011` one bit to the left:

$$\texttt{0b00010011 SLL } 1 = \texttt{0b0001001}\mathbf{1}\texttt{0}$$

The processor shifts in one new *zero* on the right; meanwhile, it shifts out the most significant bit on the left.

In the following example, we shift the 8-bit word `0b10010010` 2 bits to the right.

$$\texttt{0b10010010 SRL } 2 = \texttt{0b}\mathbf{00}\texttt{100100}$$

The processor shifts in two *zeroes* on the left of the word and shifts out the two least significant bits.

The arithmetic shift right inserts copies of the most significant bit, thus maintaining the sign of the number (two's complement). In the following two examples, we shift two numbers – the first is positive, and the second is negative) two bits to the right.

$$\texttt{0b00010011 SRA } 2 = \texttt{0b}\mathbf{00}\texttt{000100}$$

$$\texttt{0b10010010 SRA } 2 = \texttt{0b}\mathbf{11}\texttt{100100}$$

In the first case, the processor shifts in two *zeroes*, and the result is positive. In the second case, the processor shifts in two *ones*, such that the result is also negative.

Questions

10. Which shift operations produce each of the following transformations on these 8-bit words?

    (a) `0b00001001` → `0b00010010`

    (b) `0b00000101` → `0b01010000`

    (c) `0b00001101` → `0b00000110`

    (d) `0b00001000` → `0b00000001`

    (e) 8 → 2

(f) `0x10` $\rightarrow$ `0x02`

(g) $4 \rightarrow 8$

11. Consider the 8-bit number 0x10.

    (a) What is its value in decimal?

    (b) Determine the result of `0x10 SLL 1`. What is the relation between the result and the original word?

    (c) Determine the result of `0x10 SRL 1`. What is the relation between the result and the original word?

12. Consider the two's complement 8-bit number 0xE8.

    (a) What is its value in decimal?

    (b) Determine the result of the arithmetic right shift `0xE8 SRA 2`. What is the relation between the result and the original word?

    (c) This time, apply a logical right shift `0xE8 SRL 2`. Can you find any relation between the result and the original word?

    (d) Let us apply the left shift operation to our negative number `0xE8 SLL 2`. What is the relation between the result and the original word?

    (e) Finally, let us apply a wider left shift `0xE8 SLL 3`. What is the reason for the result not being as expected?

13. Use bitwise shifts to compute the following operations.

    (a) `0b00010110` $\times 2$

    (b) `0b00001010` $\times 4$

    (c) `0b00000110` $\times 8$

    (d) `0b11100101` $\div 4$

    (e) `0b00100110` $\div 8$

    (f) `0b11001100` $\div 32$

14. Multiplications and divisions are known to be time-consuming operations. How could it take up less computing resources to:

    (a) ... multiply a given number by 16?

    (b) ... divide a given number by 1024?

    (c) ... multiply a given number by 0.5?

    (d) ... multiply a given number by 10_8?

    (e) ... calculate the result of $X - X \div 2$, where X is an arbitrary integer.

4 Solutions

1. (a)

   (b)

   (c)

2.

3. (a)

   (b)

   (c)

4. (a)

   (b)

   (c)

   (d)

   (e)

5. (a)

   (b)

   (c)

   (d)

   (e)

   (f)

6. (a)

   (b)

   (c)

   (d)

   (e)

   (f)

7. (a)

   (b)

   (c)

   (d)

8. (a)

   (b)

9. (a)

   (b)

   (c)

   (d)

   (e)

   (f) `n AND 0x03`

   (g) `n OR 0x0F`

(h) `n AND 0x55`

(i) `n OR 0x55`

10. (a) `SLL 1`

    (b) `SLL 4`

    (c)

    (d)

    (e)

    (f)

    (g)

11. (a)

    (b)

    (c)

12. (a)

    (b)

    (c)

    (d)

    (e)

13. (a)

    (b)

    (c)

    (d)

    (e)

    (f)

14. (a)

    (b)

    (c)

    (d)

    (e)