ICS Homework Week 2

September 17, 2019

1 Number Conversion

Fill the table below:

Binary	Decimal	Hexadecimal
11111100011	2019	0x7E3
1010101001100100	43620	0xAA64
110111001100101	28261	0x6E65

2 Binary Operations

2.1 Basic Operations

Give two 8-bits wide numbers A=0x57 and B=0x62. Calculate the values of the following expressions.

1. A & B = 0x42
3. A | B = 0x77
5. A ^ B = 0x35
6. (A & B) | (A ^ B) = 0x77

2.2 Endianness Conversion

In computing, endianness refers to the order of bytes within a binary representation of a number. A **big-endian** ordering places the most significant byte first and the least significant byte last, while a **little-endian** ordering does the opposite. For example, a 32-bits unsigned number 0x12345678 would be [0x12, 0x34, 0x56, 0x78] in a **big-endian** ordering but [0x78, 0x56, 0x34, 0x12] in a **little-endian** ordering.

You are required to design a macro **DATA_LE32(data)**, which can convert a 32-bit big-endian number (e.g. 0x12345678) into a 32-bit little-endian number (e.g. 0x78563412).

```
1 #define DATA_LE32(data)                          \
2       ((((data) & 0x000000ff) << 24)   |     \
3        (((data) & 0x0000ff00) << 8)    |     \
4        (((data) & 0x00ff0000) >> 8)    |     \
5        (((data) & 0xff000000) >> 24))
```