

Name:

Student ID:

Computer Organization, Spring 2024  
Midterm (120pt)

1. [10pt] Environmental monitoring tracks real-time physical conditions for a specific purpose, such as wildlife tracking, disaster monitoring, and structural health monitoring. Usually, sensor nodes are deployed in the field for monitoring and transmitting the collected environmental data to a central controller so as to reflect the real-time status of the target environment. In order to fulfill the demand for a longer lifetime and low maintenance overhead, these sensors are often powered by ambient energies (e.g., solar, wind, and radio-frequency) without batteries. They are also known as networked intermittent systems, owing to the intermittent execution behaviors caused by the unstable energy sources. Micro-processors on such systems may face power failure during operations. Which kind(s) of memory is (are) improper for such systems to keep long-lasting data because these data may be lost when power failure events occur.
- a) register
  - b) cache
  - c) magnetic tape
  - d) DRAM
  - e) optical disc
  - f) SRAM
  - g) Flash

ABDF

答錯一個選項扣三分

2. [15pt] Endianness indicates how the data is interpreted in memory. According to the RFC 1700 standard, the most significant byte of the data transferred by a network interface is stored in memory first. However, unlike the network interface using the fixed byte order (or endianness), the byte order is not standardized across different processors. The POSIX API (i.e., **htonl** function) is developed to convert the data produced by different processors (known as host byte order) into the network byte order, facilitating the conversion of 32-bit unsigned values from host to network byte order. Use only bitwise operations to implement macro HTONL.

```
#include <stdint.h>
#if HOST_BYTE_ORDER == BIG_ENDIAN

#define HTONL(n) /* Q2.1: Write your code here. */

#else

#define HTONL(n) /* Q2.2: Write your code here. */

#endif

uint32_t htonl(uint32_t n) {
    return HTONL(n);
}
```

```

#if BYTE_ORDER == BIG_ENDIAN
#define HTONL(n) (n)
#else
#define HTONL(n) \
    (((uint32_t)(n) & 0xFF) << 24)    |\
    (((uint32_t)(n) & 0xFF00) << 8)    |\
    (((uint32_t)(n) & 0xFF0000) >> 8)  |\
    (((uint32_t)(n) & 0xFF000000) >> 24))
#endif

```

寫成 inline function 形式（有宣告變數）也給分，但是不能使用 for loop  
寫 assembly 不給分。

3. [40pt] The **strcpy** function copies the string pointed by *src*, including the terminating null byte ('\0'), to the buffer pointed by *dest*. The strings may not overlap, and the destination string *dest* must be large enough to receive the copy. Beware of buffer overruns! The **strncpy** function is similar, except that at most *n* bytes of *src* are copied. Note that if the length of *src* is less than *n*, **strncpy** writes additional null bytes to *dest* to ensure that a total of *n* bytes are written to *dest*. A possible C implementation of the **strncpy** function is listed below, which is followed by its corresponding RISC-V assembly code.

3.1 [30pt] Please fill out the following blanks of B01 ~ B10.

3.2 [10pt] What are the machine codes of the instructions at Line 11 and Line 13, respectively? Please write down your answers with hexadecimal digits.

每格三分

|     |                    |                     |
|-----|--------------------|---------------------|
| B01 | epilogue or _loop2 |                     |
| B02 | α1                 | Line 11: 0x00030303 |
| B03 | t1                 |                     |
| B04 | t1                 | Line 13: 0x005583B3 |
| B05 | α1                 |                     |
| B06 | _loop1             |                     |
| B07 | α2                 |                     |
| B08 | α0                 |                     |
| B09 | t1                 |                     |
| B10 | _loop2             |                     |

4. [25pt] The encodings of IEEE 754 floating-point numbers can represent normal numbers and unusual numbers (special events). For example, when the exponent is between 1 to 254, a normal floating-point number can be calculated by the values in the sign bit, exponent bits, and fraction bits. Please answer the following questions regarding IEEE 754 single precision floating-point numbers.

4.1. [13pt]: The exponent is biased by -127. What is the benefit of this biasing design? Why are all-0s and all-1s reserved in the exponent? What are purposes of these values?

方便排序(6%)，保留供特殊值使用(7%)

4.2. [4pt] What is the special event that represents the result of invalid operations, such as 0/0 or subtracting infinity from infinity? What is the bit pattern for the exponent and fraction bits for this special event?

NaN, exponent all 1, but fraction not 0.

全對才給分。

4.3 [8pt] Please fill out the table below and provide a brief explanation.

| Description                           | Value      | hexadecimal presentation |
|---------------------------------------|------------|--------------------------|
| Smallest positive denormalized number | $2^{-149}$ | 0x00000001               |
| Smallest positive normalized number   | $2^{-126}$ | 0x00800000               |

5. [15pt] Overclocking is the practice of increasing a processor's clock rate, running it at a higher speed than it was originally designed for. Please answer the following questions related to operating frequencies of processors.

Given three processors, A, B and C, all of which runs the same ISA. The CPI of A, B and C is 1.5, 1.75 and 2.0, respectively. Meanwhile, the cycle time of A and B is 300 ps and 350 ps. Processor C is running at a default clock rate, whereas the performance of A and C is identical.

5.1 [5pt] What is the default clock rate of processor C ?

4.44Ghz

5.2 [10pt] After 10% overclocking on processor C, is processor C faster than B? How much is C faster/slower than B?

Yes, ~1.49X

只有答對 faster 給 2 分，取近似值後答案不是 1.49 但算式合理也給分

(Friendly reminder:  $1\text{ s} = 10^{12}\text{ ps}$ )

6. [15pt] Please answer the True/False questions. Explanation is required when the answer is false.

6.1. [3pt] Atomic instructions aim for preventing data races, which do not occur in single-core processor system.

False keyword: concurrency, data race.

6.2. [3pt] The same algorithm written in assembly language will always be faster than the program written in C.

False keyword: modern compiler, optimization.

6.3. [3pt] As the cost of a division/multiplication operation is notably higher than corresponding bitwise operations, modern compilers tend to replace multiplications and divisions with logical left shifts and right shifts.

False keyword: arithmetic right shift or power of two

6.4. [3pt] System software sits between application and hardware, including operating system, compiler, loader, assembler, etc.

True

6.5. [3pt] RISC-V is a reduced instruction set computer ISA. This indicates fixed-length instructions of RISC-V processors.

本題開放，RVC extension 不在課程範圍內，寫 True 也給 3 分，但寫 False 不附理由只給 1 分

Keyword: mixed of 16-, 32-, 48- or 64- bits instructions.