in the following blanks.

rd	shamt 5 bits	6 bits

linker object file { relocation information 標明在程式載入Memory時，有哪些指令及數據與absolute address有關
symbol table ⇒ match (label names) to (Memory address)

Computer Organization & Assembly Language

Midterm Exam – 2007/11/14

Dept. of Engineering Science, National Cheng Kung University

1. [16 points] 4 byte，1 byte，8 bits

32 (1) How many bits are in a word? 32 bits.

32 (2) How many bits are used to represent an "unsigned int" in C?

64 (3) How many bits are used to represent a "double" in C?

ISA (4) One of the most important examples of an abstraction is the interface between hardware and the lowest-level software. What is this abstraction called?

Instruction Set Architecture

(5) Which computer components constitute the processor? What are the tasks of these components?

control + Datapath
(5)② { control ⇒ decode
Datapath ⇒ execute

(6) What are the two most important parts in an object file that contain linking information to be used by the linker?

{ relocation ⇒ identify instructions data words LK absolute address
symbol table ⇒ match (label names) to (memory address)

relocation information
symbol table

2. [12 points] Assume that you are in a company that will market a certain IC chip. The fixed costs, including R&D, fabrication and equipments, and so on, add up to $500,000. The cost per wafer is $6,000, and each wafer can be diced into 1500 dies. The die yield is 50%. Finally, the dies are packaged and tested, with a cost of $10 per chip. The test yield is 90%; only those that pass the test will be sold to customers. If the retail price is 40% more than the cost (excluding the fixed costs), at least how many chips have to be sold to break even?

$8 = \frac{6000}{1500 \times 50\%}$ = cost per die

chip 中有 X dies ⇒ 8X+10

3. [8 points] Execute the following MIPS code fragments, showing the changes that occur in the register file and in memory. You only need to show the changes.

(1)

```
addi   $19, $0, 0x20
lw     $17, 0x04($19)
add    $20, $19, $16
sw     $20, 0x08($19)
```

Registers	BEFORE	AFTER	Memory	BEFORE	AFTER
$16	0x10	×	0x20	0x22	×
$17	0x14		0x24	0x30	×
$18	0x16	×	0x28	0x40	
$19	0x28		0x2C	0x50	×
$20	0x1234		0x30	0x60	×

(2) $1 = $0 + 0x20

```
addi   $1, $0, 0x20
lw     $2, 0x04($3)
add    $0, $3, $1
bne    $0, $1, loop
```

Registers	BEFORE	AFTER	Memory	BEFORE	AFTER
$0	0x00	×	0x20	0x10	×
$1	0x14		0x24	0x30	×
$2	0x16		0x28	0x40	×
$3	0x28	×	0x2C	0x50	×
$4	0x1234	×	0x30	0x60	×

(3) Is the branch (i.e., the last line of (2)) taken? (answer YES or NO)

$s1 $t1 $t2 : h $t4=0

4. [16 points] Compile the following C program into MIPS instructions. Assume that the usage of registers is specified as (sum: $s0, i: $t0).

```
int function_x (int* a, int h){
    int sum = 0;
    int i;

    if(h < 0)     return 0;
    else{
        for(i=0; i<h; i++)
            sum += a[i];
    }
    return sum;
}
```

```
sll $s1, $t0, 2        // $s1 = 4*i
add $t1, $t1, $s1      // $t1 = a[i]
add $s0, $zero, $zero  // sum = 0
slt $t3, $t2, 0
add $t4, 0, 0 / beq $t3, 1, $t4
bne $t3, 1, Else
jr $s0.
Else:
Loop: slt $t5, $t0, $t2
add $s0, $s0, $s1 / addi $t0, $t0, $t3
```

5. [4 points] What are the differences between the following two MIPS instructions: `j` and `jal`?

6. [8 points] What kind of instructions is supported by PC-relative addressing? Explain in detail how PC-relative addressing works.

7. [20 points] Given a 32-bit bit pattern: 00110010 01010001 00101010 01010010

(1) What is the corresponding hexdecimal representation if it is an integer?

(2) What is the corresponding string if it is an ASCII string?

(3) What is the corresponding instruction if it is an MIPS instruction?

(4) Assume that $\$t0=\$t1=\$t2=\$t3=23_{ten}$ and $\$s0=\$s1=\$s2=\$s3=51_{ten}$. If the instruction decoded in (3) is then executed, which register is updated? What is the new value (in decimal) for this register?

(5) Can you explain the "stored-program" concept using the results above?

32 16 8 4 2 1
1 1 0 0 1 1

8. [4 points] Add $3.63_{ten} \times 10^4$ to $6.87_{ten} \times 10^3$, assuming that you have only three significant digits, first with guard and round digits and then without them.

9. [8 points] Show the IEEE 754 binary representation for the floating-point number 20.375_{ten} and $-5/6_{ten}$ in single precision, respectively.

128 64 32 16 8 4 2 1
1 0 1 0 0 0 1 1

10. [4 points] When using the IEEE 754 format, what is the meaning of *overflow* and *underflow*, respectively?

0.83
×2
1.66 1
×2
1.32 1
×2
0.64 0
×2
1.28 1
×2
0.56 0