1. **(20%)** It is a common practice to pack multiple patterns into a single word to save space. For example, in the following table, we would like to encode four fields (SPEED, SIZE, RW and DIR) into a byte. As shown in the following table, the first row gives the number of bits in each field. Also in the program list of a include file (under the table), we show coding definitions to be used in a C program.

Encoding bit patterns

4 bits	2 bits	1 bits	1 bits
SPEED	SIZE	RW	DIR

Definitions in a include file for the example

```
/*For SPEED field*/
#define T0 0x00
#define T1 0x10
#define T2 0x20
#define T3 0x30
#define T4 0x40
#define T5 0x50
#define T6 0x60
#define T7 0x70
#define T8 0x80
#define T9 0x90
#define T10 0xa0
#define T11 0xb0
#define T12 0xc0
#define T13 0xd0
#define T14 0xe0
#define T15 0xf0
/*For SIZE field*/
#define S0 0x00
#define S1 0x04
#define S2 0x08
#define S3 0x0c
/*For RW field*/
#define R 0x00
#define W 0x02
/*For DIR field*/
#define IN 0x00
#define OUT 0x01
```

a. **(6pt)** Please write a C procedure to perform the specified encoding. Assume we already define the procedure as `int pack(unsigned char speed, unsigned char size, unsigned char rw, unsigned char dir)`. Note that all parameters are passed into `pack()` with definitions in the include file. For example, we will use `pack(T12, S2, R, OUT)`, to pack SPEED=T12, SIZE=S2, RW=R, and DIR=OUT and to return an `int` with lowest byte storing the encoded pattern. Please also use this example to explain your program.

b. **(7pt)** Please translate the C procedure into a MIPS assembly program. Please assume the four parameters are stored at $a0, $a1, $a2, and $a3. Also assume the byte to be returned is stored in the lowest byte of $v0. Please use the same example in a. to explain your program. Please comment your MIPS program.

c. **(7pt)** Assume we have classified all MIPS instructions into four groups: arithmetic, data transfer, conditional branch, and jump. Each group can have an average CPI of one of the following values: 1.0, 1.2, 1.4 and 1.7 clocks, but no group has the same

(接下頁)

CPI. Please find the optimal CPI for each group such that the program in b. has the best performance.

2. **(10%)** How many addressing modes are used in the following MIPS code? Please select at least one instruction from the assembly code to explain different addressing modes.

Addressing mode examples

```
search:
    sw      $fp,20($sp)
    sw      $gp,16($sp)
    move    $fp,$sp
    sw      $a0,24($fp)
    sw      $a1,28($fp)
    sw      $a2,32($fp)
    sw      $zero,8($fp)
$L2:
    lw      $v0,8($fp)
    lw      $v1,28($fp)
    slt     $v0,$v0,$v1
    bne     $v0,$zero,$L5
    j       $L3
$L5:
    lw      $v0,8($fp)
    move    $v1,$v0
    sll     $v0,$v1,2
    lw      $v1,24($fp)
    addu    $v0,$v0,$v1
                $v1,0($v0)
    lw      $v0,32($fp)
    bne     $v1,$v0,$L4
    lw      $v1,8($fp)
    move    $v0,$v1
    j       $L1
$L4:
    lw      $v0,8($fp)
    addu    $v1,$v0,1
    sw      $v1,8($fp)
    j       $L2
$L3:
    li      $v0,-1
    j       $L1
$L1:
    move    $sp,$fp
    lw      $fp,20($sp)
    addu    $sp,$sp,24
    j       $ra
.end    search
```

3. (10%) Answer the following two yes or no questions about the MIPS assembly language.
   a. Is it true that instruction "*slt* $s1, $s2, $s3" will set $s1 to 1 if $s2 is less than $s3? (5%)
   b. Is it true that the so-called *jump and link* instruction, e.g., *jal 2500*, is mainly to support the return action from a procedure call to its caller function? (5%)

4. (15%) Consider a pipelined CPU design with the 5 stages being (1) *instruction fetch*, (2) *decoding*, (3) *execution*, (4) *memory access*, and (5) *register write back*.

   (a) List all instruction stages in which we need to read or write the register file. (5%)

   (b) What is the minimum number of IO ports required for the register file if the access of the register file takes one full clock cycle? (Note: an IO port can be used for either a *read* or *write* operation.) (5%)

   (c) What is the minimum number of IO ports required for the register file if the access of the register file takes only half a clock cycle? (Note: we will be able to perform two accesses to or from the register files within one clock cycle.) (5%)

5. (10%) Consider the following summary table for execution steps that need to be performed by four major instruction classes: (1) *arithmetic-logic* (or called *R-type*), (2) *memory-reference*, (3) *branch*, and (4) *jump* instructions.

(a) Complete the missing action in entry marked "?" under the column for the memory-reference instructions. (5%)

(b) What is the instruction class of entry marked α? (5%)

Execution Step	R-type	Memory-Reference	α	β
Instruction Fetch	IR = Memory[PC]; PC = PC + 4;			
Instruction Decode	A = Reg[IR[25-21]]; B = Reg [IR[25-21]]; ALUOut = PC + (sign-extend (IR[15-0]) << 2);			
Execution	ALUOut = A op B;	ALUOut = A + sign-extend (IR[15-0]);	PC = PC[31-28] || (IR[25-0] << 2);	If(A==B) then PC = ALUOut;
Memory Access		Load: MDR = (?); Store: omitted;		
Register Write Back	Reg[IR[15-11]] = ALUOut;	Load: Reg [IR[20-16]] = MDR;		

6. (14%) Answer the following questions:

a) Explain (please also draw a diagram) the following methods how can they resolve the multiple simultaneously interrupt requests from the I/O Devices ? (1) Daisy chain, (2) Polling. (6%)

b) Explain the following two modes, (1) cycle stealing and (2) block mode, for DMA controller to transfer the data from I/O device to Memory. Which mode is transparent (unknown) to CPU operation ? why ? (4%)

c) Now suppose that CPU is executing a maximum of 10^6 instructions/sec, An average instruction execution requires five machine cycles, three of which use the memory bus, a memory read/write uses one machine cycle for transferring one word. What is DMA transfer rate (word/sec) for the above two DMA controller modes ? (4%)

7. (15%) Consider a two-level cache-memory scheme that uses demand paging and preemptive allocation policy. The faster memory has a cache capacity of four pages. There are three different replacement policy, FIFO, LRU (lease recently used), and OPT (optimal). Now suppose the page address stream formed by executing a program Q is shown as follows

1 6 4 5 1 4 3 2 1 2 1 4 6 7 4 1 3 1 7

Assume a "hot" start, in which the cache initially has pages 1, 2, 3, and 4 allocated to it.

a) Show the time sequence of the cache-replacement sequence of three different replacement policy, and how many "hits" during the page allocation sequence ?

Time 1 2 3 4 5 6 7 8 9 10 11 12 .....

Page 1 6 4 5 1 4 3 2 1 2 1 4.....

1																		
2																		
3																		
4																		

hit

b) Which of the page-replacement policy FIFO or LRU is more suitable in this case ? Give a short intuitive justification of your answer..

c) What is the stack replacement policy ?

8. (6%) The interrupt breakpoint or the DMA breakpoint is the instance when the CPU responds to the interrupt request (INTR) or DMA request. If the CPU is executing an instruction (several micro-steps or machine cycles are required for executing each instruction in one instruction cycle).

(a) Where the interrupt breakpoint occurs ?

(b) Where the DMA breakpoint occurs ?

(c) After receiving the INTA signal from CPU, how the I/O device identifies itself to CPU ?