Department of Electrical and Computer Engineering
University of Alabama in Huntsville

EE/CPE 421/521 – Microcomputers
Final Exam

Instructor: Dr. Aleksandar Milenkovic

Date: Thursday, April 29, 2004

Place: EB 240

Time: 6:30 PM – 9:00 PM

Note: Work should be performed systematically and neatly. This exam is closed books and closed neighbour(s). Allowable items include exam, pencils, straight edge, calculator, and materials distributed by the instructor. Best wishes.

Question	Points	Score
1	30	
2	25	
3	50	
Sum	105	

Please print in capitals:

Last name:_____

First name: _____

1. (30 points) M68000 Assembly language and C

A. (6 points) Finish the compilation of this C program by adding **one** assembly instruction per C source lines *13 and *14. If the initial value of A6 is $1000, fill in the new values of the register that has changed (A2 or A3) after the execution of each assembly instruction. Integer data type (**int**) is 2 bytes long, while **long** occupies 4 bytes.

```
*11     register int *P_x = &x;
        LEA.L -2(A6),A2                    A2 =

*12     register long *P_y = &y;
        LEA.L -8(A6),A3                    A3 =

*13     P_x += 3;
                                           A2 =

*14     P_y -= 5;
                                           A3 =
```

B. (10 points) Parameters can be passed to a subroutine by *value* or by *reference*. Explain the meaning of the expressions "*passing parameters by **value***" and "*passing parameters by **reference***". Discuss their relative merits.

C. (10 points) Illustrate your answer using a simple example given below.
Show the result of compilation for **pqr(a, &b)** in the main program (fill in the box A). Show the content of the stack at the beginning of the subroutine execution. Show how parameters are accessed from within the subroutine (fill in the box B).

```
void pqr(int a, int * b) {
  int x, y;
  x = a;
  *b = x;
}
void main (void){
  int a = 22; int b = 33;

 pqr(a, &b);
}
```

```
*1    void pqr(int a, int * b) {
                                    SECTION           S_pqr,,"code"
                                    * Parameter a is at 8(A6)
                                    * Parameter b is at 10(A6)
                                    * Variable x is at -2(A6)
                                    * Variable y is at -4(A6)
                                    XDEF        _pqr
                                    _pqr
                               LINK        A6,#-4
                                    __P1        EQU         $000004
*2      int x, y;
*3      x = a;
                                                                    Box B
*4      *b = x;


*5    }
                               UNLK        A6
                               RTS

                          * Function size = 22
```

```
*6
*7    void main (void){
                                        XREF        __main
                                        * Variable a is at -2(A6)
                                        * Variable b is at -4(A6)
                                        XDEF        _main
                                        _main
                                LINK        A6,#-4
                                __P2        EQU         $00001a
*8     int a = 22; int b = 33;
                                MOVE        #22,-2(A6)
                                MOVE        #33,-4(A6)
*9     pqr(a, &b);
                                                                    Box A
*10   }
                                UNLK        A6
                                RTS
                                _dgroup             data
                            END
```

D. (4 points) Populate the memory map to illustrate the effect of the following 68K assembly language directives. Use only HEX NUMBERS and enter each byte separately. For unknown values (memory locations) use "?".

```
        ORG   $400
LX1     DC.W  $1234
Y1      DC.L  1
PQ      DS.B  4
ZZ      DC.L  Y1
```

	15-8	7-0
$00 0400		
$00 0402		
$00 0404		
$00 0406		
$00 0408		
$00 040A		
$00 040C		
$00 040E		

2. (25 points) Bus cycles, performance, power

A. (5 points) How many (if any) wait states must be inserted for the following CPU and memory parameters:

- M68000 CPU operates at 12.5 MHz,
- data setup time t_{DICL} is 10ns,
- clock low to address valid t_{CLAV} is 55ns,
- and the memory access time t_{AA} is 450ns.

B. (10 points) For the given assembly language program executed on an 8MHz Motorola 68000 processor and given number of clock cycles needed for each instruction (last column) find the total execution time [s], CPI (cycles per instruction), and MIPS (Million of Instructions per second).

```
_main           MOVEQ.L     #7,D4           4
                MOVEQ.L     #1,D0           4
                CLR.L       D1              6
L20001          MOVE.L      D4,D7           4
                EXT.L       D7              4
                ADD.L       D7,D1           8
                MULS.W      D7,D0           69
                SUBQ.L      #1,D4           8
                CMPI.B      #1,D4           8
                BGT.B       L20001          10/8 not taken
                RTS                         16
```

Find the total execution time. (For multiplication, use the worst-case execution time)

Calculate the average CPI (number of clocks per instructions).

Calculate the MIPS rate.

C. (10 points) We are considering energy efficiency of a computational task for two configurations of a wireless intelligent sensor. The computation is performed 5 times per second (system frequency is 5Hz) and one execution takes 124,800 processor clock cycles.
Configuration C1: processor clock frequency is 8MHz, power supply is 3.3V;
Configuration C2: processor clock frequency is 5MHz, power supply is 2.4V.
Ignoring energy consumed when the processor is inactive which configuration is more energy efficient and how many times (E_{C1}/E_{C2}=?)?

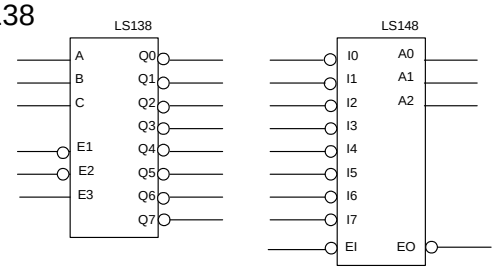
3. (50 points) CPU Hardware Model

Design microcomputer system with MC68000 microprocessor with the following characteristics.

- Supervisor program memory of 128KB is implemented using 32K×8 EPROM components, starting at the address $00 0000. 32K×8 EPROM components require 2 wait cycles for read cycles. Control inputs for 32Kx8 EPROM components are CE* and OE*.
- Supervisor data memory of 32KB is implemented using RAM 8K×8bit components and resides in the consecutive address window. RAM 8K×8bit components do not require wait cycles. Control inputs for RAM 8K×8bit components are CE*, OE*, and WE – 0 for write, 1 for read.
- User program and data memory of 64KB is implemented using 16K×8bit RAM components and resides at the address $10 0000. RAM 16K×8bit components do not require wait cycles. Control inputs for RAM 16K×8bit components are CE*, OE*, and WE – 0 for write, 1 for read.
- Two 8-bit peripherals (PER2 and PER3) using 32-byte address windows starting at address $20 0000. Both peripherals require 1 wait cycle for read, write, and IACK cycles. Interrupt mechanism is vectored. PER2 is connected to IRQ2 and PER3 to IRQ3. Control inputs for the peripherals are CE*, IRQ*, IACK,* and WE – 0 for write, 1 for read.

Your solution should include: processor with all relevant lines, memory subsystem, i/o subsystem, corresponding interrupt interface, and decoding logic.

You may use AND, OR, NOT, NAND, NOR, XOR logic gates, LS138 (3 to 8 decoder), LS148 (priority encoder) or any other standard digital circuits.



Function Code Output			Processor Cycle Type
FC2	FC1	FC0	
0	0	0	Undefined, reserved
0	0	1	User data
0	1	0	User program
0	1	1	Undefined, reserved
1	0	0	Undefined, reserved
1	0	1	Supervisor data
1	1	0	Supervisor program
1	1	1	CPU space (interrupt acknowledge)

Bonus question:

Add necessary logic that will allow the microcomputer system to recover from an irregular bus cycle (DTACK* is not generated).