The University of Alabama in Huntsville
Electrical & Computer Engineering Department
CPE/EE 421/521 01
Final Exam
December 13, 2004

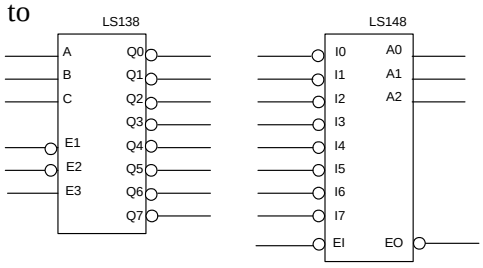
Name: _____

1. (25 points) Write a subroutine in 68000 assembly language to calculate the value of x + x2. The parameter x is a 16-bit value that is passed to the subroutine by value. The result is a 32-bit longword that is to be passed by reference. You must use a stack frame in your subroutine. You must save any working registers on the stack. You should use the minimum number of registers in your subroutine. Draw a memory map of the stack during the execution of your subroutine. **(**

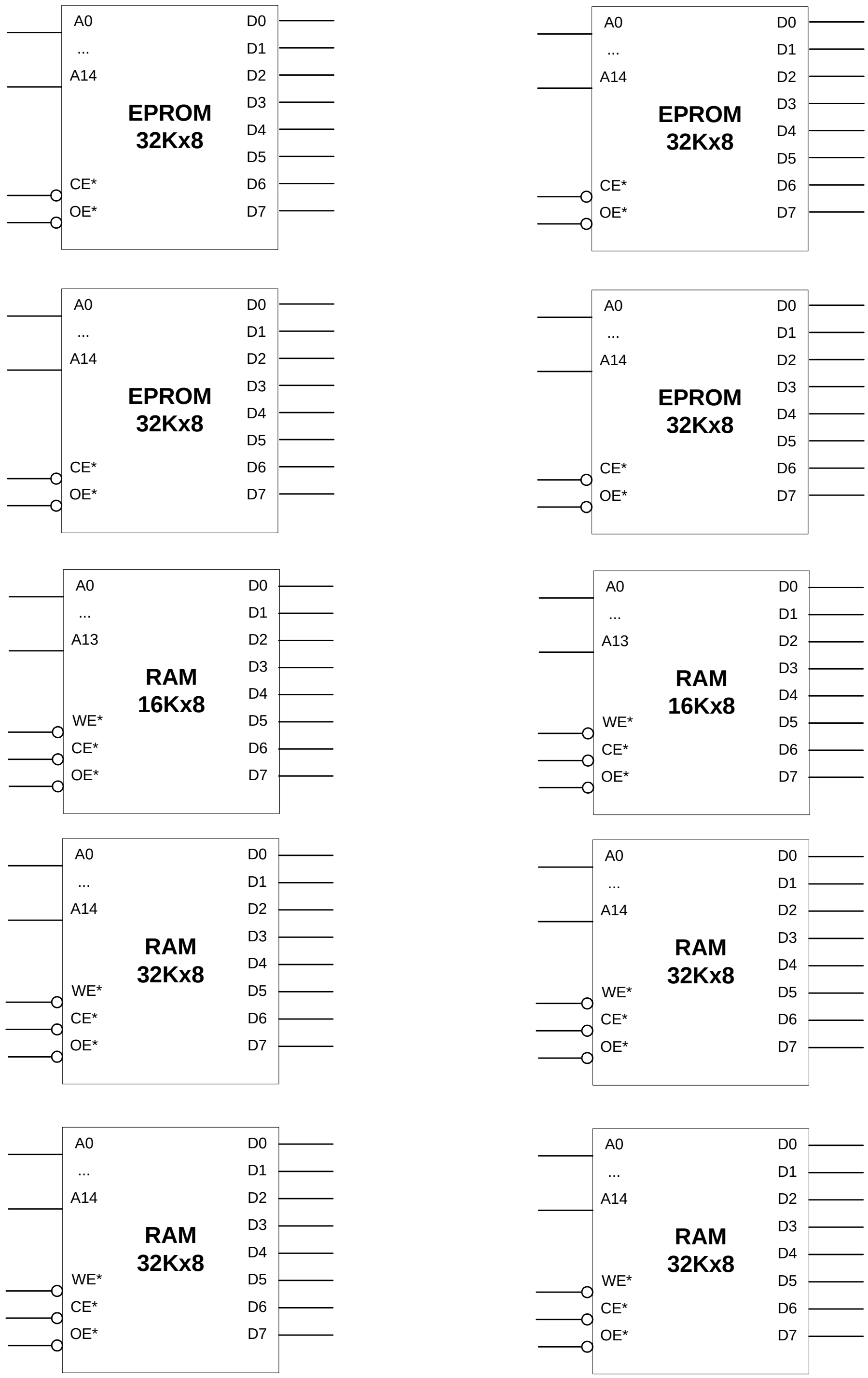
2. (40 points) CPU Hardware Model
   Design a microcomputer system using a MC68000 microprocessor that has the following characteristics.
   - Supervisor program memory of 128KB is implemented using 32K×8 EPROM components, starting at the address $00 0000. 32K×8 EPROM components require 3 wait cycles for read cycles. Control inputs for 32Kx8 EPROM components are CE* and OE*.
   - Supervisor data memory of 32KB is implemented using RAM 16K×8bit components and resides in the consecutive address window. The 16K×8bit RAM components do not require wait cycles. Control inputs for the 16K×8bit RAM components are CE*, OE*, and WE – 0 for write, 1 for read.
   - User program and data memory of 128KB is implemented using 32K×8bit RAM components and resides at the address $06 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 16-byte address windows starting at address $27 8400. Both peripherals require 1 wait cycle for read, write, and IACK cycles. Interrupt mechanism is vectored. PER2 is connected to IRQ5 and PER3 to IRQ1. Control inputs for the peripherals are CE*, IRQ*, IACK,* and WE – 0 for write, 1 for read.

Make sure that the 68000 has access to user program and data space in supervisor mode. (Note: one wait state is ONE CLOCK CYCLE long)

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)


EPROM
32Kx8
A0
...
A14
CE*
OE*
D0
D1
D2
D3
D4
D5
D6
D7
EPROM
32Kx8
A0
...
A14
CE*
OE*
D0
D1
D2
D3
D4
D5
D6
D7
EPROM
32Kx8
A0
...
A14
CE*
OE*
D0
D1
D2
D3
D4
D5
D6
D7
EPROM
32Kx8
A0
...
A14
CE*
OE*
D0
D1
D2
D3
D4
D5
D6
D7
RAM
16Kx8
A0
...
A13
WE*
CE*
OE*
D0
D1
D2
D3
D4
D5
D6
D7
RAM
16Kx8
A0
...
A13
WE*
CE*
OE*
D0
D1
D2
D3
D4
D5
D6
D7
RAM
32Kx8
A0
...
A14
WE*
CE*
OE*
D0
D1
D2
D3
D4
D5
D6
D7
RAM
32Kx8
A0
...
A14
WE*
CE*
OE*
D0
D1
D2
D3
D4
D5
D6
D7
RAM
32Kx8
A0
...
A14
WE*
CE*
OE*
D0
D1
D2
D3
D4
D5
D6
D7
RAM
32Kx8
A0
...
A14
WE*
CE*
OE*
D0
D1
D2
D3
D4
D5
D6
D7

3. (5 points) How many (if any) wait states must be inserted for the following CPU and memory parameters:

- M68000 CPU operates at 10 MHz,
- data setup time t_{DICL} is 10ns,
- clock low to address valid t_{CLAV} is 50ns,
- and the memory access time t_{AA} is 420ns.

4. (15 points) For the given assembly language program executed on a 12.5 MHz Motorola 68000, find the total execution time [s], CPI (cycles per instruction), and MIPS (Million of Instructions per second).

```
_main           MOVEQ.L       #7,D4
                MOVEQ.L       #1,D0
L20             MOVE.L        D4,D7
                MULS.W        D7,D0
                SUBQ.L        #1,D4
                CMPI.B        #15,D4
                BLT.B         L20
                RTS
```

a. (10 points) Find the total execution time. (For multiplication, use the worst-case execution time)

b. (5 points) Calculate the average CPI (number of clocks per instructions).

68000 Registers (all values are hex unless otherwise noted)

D0	01234567	D1	89ABCDEF	D2	0001002D	D3	ABCD7FFF
D4	33449127	D5	AAAAAAAA	D6	ABCD0003	D7	55555555
A0	00007020	A1	00007028	A2	0001000A	A3	00007034
A4	00010020	A5	0000FFFA	A6	00010000	A7	00010080
		XNZVC	b10100			PC	00004000

Main memory (all values are hex unless otherwise noted)

007000	AE	007020	5A	010000	DD	010020	DC
007001	F2	007021	AD	010001	B2	010021	25
007002	32	007022	99	010002	00	010022	15
007003	77	007023	92	010003	15	010023	17
007004	89	007024	79	010004	76	010024	29
007005	90	007025	33	010005	19	010025	39
007006	1A	007026	97	010006	92	010026	49
007007	AE	007027	14	010007	26	010027	2D
007008	EE	007028	79	010008	17	010028	B2
007009	F1	007029	E7	010009	14	010029	62
00700A	F2	00702A	00	01000A	23	01002A	81
00700B	A4	00702B	0A	01000B	E7	01002B	21
00700C	AE	00702C	88	01000C	19	01002C	45
00700D	88	00702D	18	01000D	92	01002D	18
00700E	AA	00702E	82	01000E	19	01002E	31
00700F	E4	00702F	79	01000F	54	01002F	D9
007010	7E	007030	2B	010010	45	010030	AA
007011	8D	007031	17	010011	99	010031	77
007012	9C	007032	46	010012	15	010032	78
007013	C4	007033	9E	010013	43	010033	AE
007014	B2	007034	FC	010014	25	010034	EA
007015	12	007035	FF	010015	76	010035	34
007016	39	007036	77	010016	89	010036	25
007017	90	007037	60	010017	17	010037	17
007018	00	007038	21	010018	81	010038	15
007019	89	007039	42	010019	17	010039	14
00701A	14	00703A	55	01001A	4E	01003A	17
00701B	01	00703B	EA	01001B	72	01003B	F9
00701C	3D	00703C	61	01001C	33	01003C	8A
00701D	77	00703D	81	01001D	23	01003D	0F
00701E	89	00703E	C9	01001E	E1	01003E	F2
00701F	9A	00703F	AA	01001F	CD	01003F	E5

5. (9 points) What is the effect of applying each of the following 68000 instructions assuming the initial conditions shown? Represent modified internal registers, memory locations and condition codes.

(a) (3 points)

```
MOVE.B  (A1)+, D0
CMPI.B  #$5A, D0
```

X	N	Z	V	C

(b) (3 points)

```
CMPM.W  (A1)+, (A3)+
```

X	N	Z	V	C

(c) (3 points) `CLR.L 5(A3,D6.W)`

X	N	Z	V	C

6 (6 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     DS.W    2
Y1      DC.L    1
PQ      DS.B    4
ZZ      DC.L    PQ
```

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