

C SC 230 - Computer Architecture and Assembly Language - Spring 2001

Midterm Test - February 19, 2001

INSTRUCTOR: M. Serra

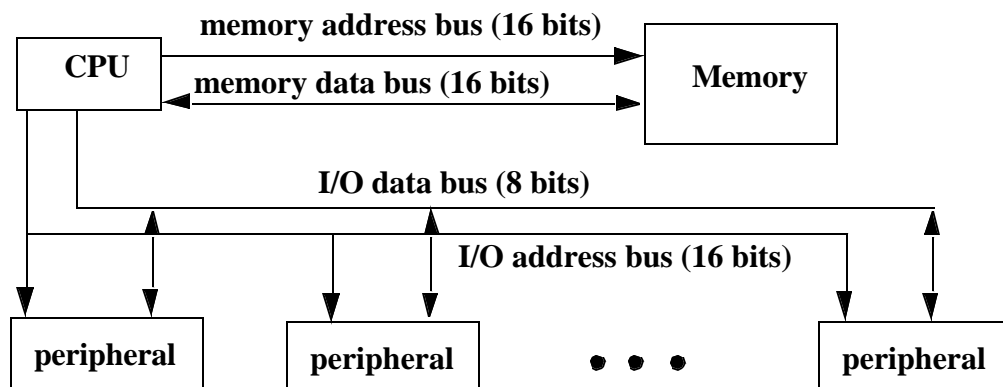
STUDENT NUMBER: (last 5 digits): _____

LAST NAME: _____ **SIGNATURE:** _____

TOTAL MARKS: 98. This is a closed book test. Attempt all questions. Write your answers on the question paper in the spaces provided. Use the reverse side for rough work. Total time allowed is 60 minutes. Use **ONLY** the handout provided for extra notes.

QUESTION 1 [8 marks]. In the picture below, an isolated I/O architecture is shown with the size of the address buses and data buses labelled. An 8-bit data bus implies here a byte-addressable cycle, while a 16-bit data bus implies here a word-addressable cycle.

(a) How many addressable locations are available for memory?	
(b) What is the total memory size in bytes?	
(c) How large is the address space for I/O peripherals in bytes?	
(d) If every peripheral must be allocated an address space of 2^{12} bytes, what is the largest number of peripherals that can be connected?	



QUESTION 2 [10 marks]. (a) Assuming a 2's complement representation, convert the following numbers from decimal to 8-bit binary and to its equivalent hexadecimal:

Decimal	Binary	Hexadecimal
35_{10}		
-36_{10}		

(b) Given the 4-bit hexadecimal number, state its decimal equivalent according to the assumption of its representation listed in each heading:

Hexadecimal	<i>Unsigned</i>	<i>2's complement</i>	<i>Signed magnitude</i>
E_{16}			
4_{16}			

QUESTION 3 [4 marks].

Give one or more 68HC11 instructions to swap the contents of the two bytes labelled P and Q, previously declared as:

P RMB 1
Q RMB 1

QUESTION 4 [6 marks].

Assume accumulator A contains the number -36_{10} and the 68HC11 instruction CMPA #10 has been executed. Which of the following branch instructions will result in the branch being taken towards "target"?

Branch	Yes/No
BEQ target	
BPL target	
BHS target	

QUESTION 5 [22 marks].

(a) [10] Fill in the table below with the appropriate information about the instructions listed

Instruction	Addr. Mode	# bytes in code	# READ cycles	#WRITE cycles	What it does (use appropriate notation)
LDAA 0,X					
STAB \$0123					
TXS					
SUBA #12					

(b) [12] For each instruction, state exactly what happens in the “Fetch/Decode/Execute” cycle by indicating what goes on each bus and when. One instruction is given to you as an example with all the answers so you can see what is expected. You may not need all the spaces provided.

SUBA #12		what happens	Address Bus	Data Bus
	Step1	fetch instruction	content of PC = address of instruction	<i>going:</i> nothing <i>returning:</i> opcode
	Step 2	fetch operand	content of PC = address of operand	<i>going:</i> nothing <i>returning:</i> operand = 12
	Step 3	ALU does subtraction of A - 12, result in A	nothing	nothing
	Step 4	_____	_____	_____
LDAA 0,X		what happens	Address Bus	Data Bus
	Step1			
	Step 2			
	Step 3			
	Step 4			
	Step 5			
STAB \$0123		what happens	Address Bus	Data Bus
	Step1			
	Step 2			
	Step 3			
	Step 4			
	Step 5			

QUESTION 6 [24 marks].

Assume the registers have been initialized as shown in the table below.

Draw the stack frame, enter the value of each stack slot (if it is known), and state the value of the register listed at the end of the given 68HC11 instruction sequence (write in the table).

Also write the precise address of each stack slot on the left end side of the stack picture.

Register	Initial	At End	work space (not marked)
A	\$12		
B	\$20		
X	\$FFFF		
Y	\$F1E2		
D	\$1220		
SP	\$01FF		

	LDS	#\$01FF
	DES	
	DES	
	CLRB	
	LDAA	#10
	PSHA	
	LDAA	#\$C0
	PSHA	
	LDX	#\$D000
	PSHX	
	BSR	SUB1
	
	
SUB1	PSHB	
	PSHY	
	TSX	
	XGDX	
	SUBD	#6
	XGDX	
	TXS	
	

[illegible]

QUESTION 7 [20 marks].

Consider the program below which manipulates three strings in some fashion using the subroutine “mystery”. Parameters are being passed through registers. You have 4 tasks:

(a) [4] place appropriate comments besides the code

(b) [4] state what the subroutine mystery does

(c) [4] state what the program does overall

(d) [12] rewrite using the stack for parameters by rewriting only the necessary lines or inserting some (use the numbering of lines given by the Assembler to be clear). For example, if inserting lines between 13 and 14, list them as 13.1, 13.2, etc.

- 1 ;(a) state what the program does overall
- 2 ;(b) state what the subroutinemystery does
- 3 ;(c) place appropriate comments
- 4 ;(d) rewrite using the stack for parameters

```

5
6      ORG  $0001
7  string1 FCC  "xxxxxxx"      ;string 1
8          FCB  0              ;end string 1
9  string2 FCC  "yyyyyyy"      ;string 2
10         FCB  0              ;end string 2
11  string3 RMB  80
12
13      ORG  $C000
14      LDX  #string1
15      LDY  #string3
16      JSR  mystery
17      LDX  #string2
18      JSR  mystery
19      LDAB #0
20      STAB 0,Y
21      STOP
22
23  mystery
24      PSHX
25      PSHB
26      top  LDAB 0,X
27          BEQ  exit
28          STAB 0,Y
29          INX
30          INY
31          BRA  top
32
33  exit  PULB
34          PULX
35          RTS
36          END

```