Trend: lots of false, similar questions very

C SC 230 Computer Architecture and Assembly Language
April 2000 Exam Sample Solutions

0000 1111
1111 0001

1. (12 marks) Circle the correct answer for each of the following:

The 8-bit two's complement representation of -15_{10} is 11110001_2.	**True**	False
Two's complement representation has different representations for +0 and –0.	True	**False**
In two's complement addition, overflow can only occur when adding two positive numbers.	True	**False**
Single bit parity allows for the detection and correction of single bit errors.	True	**False**
On the 6811, the external address and data buses are asynchronous.	True	**False**
The data bus to main memory must have the same number of bits as the word size.	True	**False**
A processor must have a stack pointer register in order to support a jump to subroutine instruction.	True	**False**
Program memory parameter passing can never be used if the program code is to be stored in ROM.	True	**False**
On the 6811, on-processor memory can share addresses with off-processor IO devices.	True	**False**
Unless explicitly forbidden by the user program, nested maskable interrupts are allowed on the 6811.	True	**False**
Polling should never be used if an interrupt can be used instead.	True	**False**
The PowerPC uses purely hardware supported branch prediction.	True	**False**

2. (10 marks) Explain how any **two** of the PowerPC, Pentium II/III and Crusoe processors handle the out of order execution of instructions.

(5 marks each for any two)

PowerPC: Instructions are held in an instruction execution pool. The hardware is capable of dispatching any of the bottom four instructions in the queue to the appropriate execution unit. Hence the out of order execution is at the instruction level where all instructions have fixed size. The hardware ensures that the reordering of the instructions will not affect the result i.e. it checks for dependencies.

Pentium: Instructions are decomposed by the hardware into micro-ops which are placed in an instruction pool. Micro-ops can be dispatched from the pool to execution units out of order. A hardware retirement unit is used to piece back the micro-ops in order to ensure the result is the same as if the original instructions were executed in order.

Crusoe: (not covered Fall 2001) The code morphing software decomposes X86 (Intel) instructions into fixed length instructions called atoms. The software packs these atoms into very long word instructions (molecules). The packing can be out of order. Molecules are

executed in order with atoms executed in parallel. The hardware uses special holding registers to ensure the results reflect execution of the original X86 instructions.

3. (5 marks) Number the following steps from 1 to 5 in the order they are performed in processing an interrupt on the 6811 using the interrupt jump table technique

correct answer ⇓				
1	1	1	1	recognize the interrupt event and set the event flag
3	2	3	2	load the PC with the value from the appropriate interrupt vector
5	5	5	5	execute the first instruction of the interrupt handling routine
2	3	4	4	push the processor registers onto the stack
4	4	2	3	execute the appropriate jump instruction in the jump table
5	4	3	2	marks (the list shows the common responses)

4. (3 marks) What does it mean to say a processor is *superscalar* and what is the most fundamental property of a superscalar processor design?

A processor is superscalar if it can execute more than one instruction per processor clock cycle. (1 mark)

A superscalar design must have multiple execution units so that instructions can be executed in parallel. (2 marks)

5. (6 marks) What are the four basic stages in an instruction pipeline?

Fetch and decode instruction
Fetch operands (1 marl each)
Execute instruction
Store results

Why does a pipeline help improve instruction throughput on a processor?

A pipeline allows several instructions each at different stages of execution to be executed in parallel. (2 marks)

6. (4 marks) For the single precision IEEE floating point representation for 13.5_{10}

1101.1
.011 Mantissa
sign = 0
exp = 130
0.8

```
            LDD  TCNT,X          / START TIMING
            ADDD #SLICE
            STD  TOC2,X
            LDAA #OC2
            STAA TFLG1,X
            CLR  TCOUNT
            CLR  TCOUNT+1
            BRA  DONE
IC12E       BCLR TMSK1,X OC2     / 2ND EDGE OF PAIR STOP
TIMING
            LDD  TCOUNT
            CPD  #MAX
            BGT  DONE
            BSET PORTA,X $40     / WITH TIME SO TURN LED ON
            BRA  DONE
IC1FE       BCLR PORTA,X $40     / TURN LED OFF
DONE
            RTI

OC2ISR      LDX  #REGBASE
            LDAA #OC2            / CLEAR FLAG
            STAA TFLG1,X
            LDX  TCOUNT          / INC TIME COUNT
            INX
            STX  TCOUNT
            RTI
```

10. (25 marks)

```
REGBASE   EQU  $1000   / USE THESE DEFINITIONS
PORTA     EQU  $0      / YOU MAY NOT NEED THEM ALL
TCNT      EQU  $0E     / YOU CAN ADD ANY OTHERS
TIC3      EQU  $14     / YOU NEED
TOC2      EQU  $18
TCTL2     EQU  $21
TMSK1     EQU  $22
TMSK2     EQU  $24
TFLG1     EQU  $23
TFLG2     EQU  $25

IC1       EQU  $04
OC2       EQU  $40
SLICE          EQU  40000
MAX       EQU  250
          ORG  $D000
TCOUNT    RMB  2
EDGEPAR   RMB  1

          ORG  $00E8
          JMP  IC1ISR
          ORG  $00DC
          JMP  OC2ISR

          ORG  $C000
asm("SEI")
          LDX  #REGBASE
          LDAA #IC1
          STAA TMSK1,X
          LDX  #REGBASE
          BSET TCTL2,X $30     / DETECT BOTH EDGES ON IC1
          LDAA #IC1            / CLEAR FLAG
          STAA TFLG1,X
          CLR  EDGEPAR
asm("CLI")

          BRA  *

IC1ISR    LDX  #REGBASE
          LDAA #IC1
          STAA TFLG1,X
          LDAA PORTA,X
          ANDA #IC1
          BEQ  IC1FE
IC1RE     LDAA EDGEPAR         / FLIP EDGE PARITY
          EORA #1
          STAA EDGEPAR
          BEQ  IC12E
IC11E     BSET TMSK1,X OC2     / FIRST EDGE OF PAIR
```

9. (15 marks)

```
; checks a list to see if it is ordered
; returns 0 if not ordered; +1 if ascending; -1 if descending
;
ORDERED
      PSHB          / SAVE REGISTERS
      PSHX
      PSHY
      TSX
      LDAA  #0      / ASSUME UNORDERED
      LDAB  7,X
      LDX   8,X
LOOP  LDY   0,X
      CPY   2,X     / COMPARE ADJACENT ELEMENTS
      BLT   ASCEND
      BEQ   NEXT    / IGNORE PAIR IF EQUAL
      CMPA  #1      / DESCENDING
      BEQ   UN
      LDAA  #-1
      BRA   NEXT
ASCEND
      CMPA  #-1     / ASCENDING
      BEQ   UN
      LDAA  #1
NEXT
      INX           / MOVE DOWN LIST
      INX
      DECB          / DEC COUNT
      CMPB  #1      / CHECK FOR END
      BNE   LOOP
      BRA   DONE
UN    CLRA          / UNORDERED
DONE  PULY          / RESTORE REGISTERS
      PULX
      PULB
      RTS
```

What is the value of the sign bit: __0__(1 mark)

What is the actual value stored for the exponent (in decimal): ___130____(1 mark)_

What is the actual value stored for the mantissa (in binary): __1011___(2 marks)__
(do not show trailing 0's)

7. (10 marks) Explain each of the following terms with respect to cache memory:

associative cache: The memory address is held in a tag (1 mark) and all tags have to be searched to locate the address (1 mark).

direct mapped cache: The cache is divided into slots and each line must be in a specific slot (1 mark). The cache checks one slot only and if the line is not found it is brought in from memory into that slot. (1 mark)

set-associative cache: A combination of set associative and direct mapped. (1 mark) A line can appear in any one of a set of slots (1 mark).

write back: The data is written to the cache and is not written back to main memory until the line of the cache has to be swapped out. (2 marks)

replacement policy (give one example): The policy by which the cache decides which line to swap out to make room for a needed line (1 mark). Example: one of random, FIFO, LRU or access count.

8. (10 marks)

```
; finds sum of numbers from 1 to 99 which are also
integer
; multiples of 3
;
; the answer is in IX


     ORG  $C000
START
     LDX  #0
     LDAB #3 / SMALLEDT ODD NUMBER WHICH IS A MULTIPLE OF
3
LOOP
     ABX
SKIP
     ADDB #6 / MULTIPLES OF 3 WHICH ARE ODD ARE 6 APART
     CMPB #99
     BLS  LOOP
```