538 Lecture Notes Week 6

Answers to last week's questions

Remark on Decimal arithmetic and bcd (packed and unpacked) formats

Questions

1. Consider the following program:

```
              XDEF Entry            ; export 'Entry' symbol
              ABSENTRY Entry        ; for absolute assembly: mark
this as application entry point
    INCLUDE 'derivative.inc'

              org $400  ;RAM data section
  counter     dc.b $00, $11, $22, $33
              dc.b $44, $55, $66, $77
              dc.b $88, $99, $aa, $bb
              dc.b $CC, $dd, $ee, $ff

              org   $4000 ;ROM code section
  Entry:
              lds   #$0410
              jsr init
              cli

              ldx #$400
              ldy #$401
```

```
              ldd 1,x+
              std 2,y+

              bra *   ;Infinite loop (awaiting interrupts)


  tofISR:
             bset TFLG2,$80
             inc counter
             rti

  init:
             ldaa #%10000000
             staa TSCR1 ;Enable TCNT by setting bit 7
             staa TFLG2 ;Clear the TOF flag by writing to bit 7
             staa TSCR2 ;Turn timer overflow interrupt on by
setting bit 7
             rts

  ;*************************************************************
  ;*                  Interrupt Vectors                        *
  ;*************************************************************
              org $ffde
              dc.w tofISR      ;Timer TOF Vector

              ORG   $FFFE
              DC.W  Entry      ;Reset Vector
```

Show the memory dump 0x0400-0x040f just before the “rti” instruction is executed. Assume that the interrupt occcurs during the “bra *” instruction.

ANSWER:

```
0400 01 00 11 33 44 55 66 C0 11 00 04 01 04 03 40 12
```

Vocabulary

New terms are in **bold**.

Memory Dump	The standard way to display the contents of a block of memory in hexadecimal.
CPU	The Central Processor Unit, the computer's heart.
Bus	A set of parellel wires of digital signals
Address Bus	It indicates the address involved in a bus operation
Data Bus	The data being transferred during a bus cycle.
Control Bus	Signals to time and define the type of bus cycle.
IDE	"Integrated Development Environment". Includes editors, compilers, assemblers, disassemblers, etc. We use *CodeWarrior* in this course. In your Java course, you used *Netbeans*. *Eclipse* is another IDE you may use in other courses.
Read Bus Cycle	Data transferred to the CPU.
Write Bus Cycle	Data transferred from the CPU to memory or a memory-mapped device.
Idle Bus Cycle	Bus inactive
Assembler	Software to translate from symbolic assembler to machine code
Disassembler	Software to translate from machine code to symbolic assembler.
Assembler directive	A line of assembler code that gives information to the assembler software and does not correspond to a machine instruction.
Program Counter	A register containing the address of the next instruction.
Stack Pointer	A register containing the address of the top of stack.
Condition Code Register	Contains bits indicating various conditions (such as whether the last number added was zero, negative, generated a carry, etc.) Also called the *Status Register* in some machines (such as Intel processors).
Index Register	A register mainly used as a pointer. It's size equals the width of the Address Bus.
Arithmetic Shift	Only applies to a right shift where the sign bit is "shifted in" from the right maintaining the sign of the shifted value.
Indexed Addressing	Accessing the contents of memory whose address is calculated by adding an offset (usually zero) to an index register.
Indirect Indexed Addressing	Using indexed addressing to obtain another pointer in memory and then dereferencing the location it points to.
Overflow (V) bit	Set when the result of an addition/subtraction will not fit in the number

	of bits used.
Effective Address	The address that will be used in indexed addressing. i.e. the index register + offset.
Addressing Mode	The way an operand is specified.
Inherent Addressing	The operand is inherent in the instruction itself.
Immediate Addressing	The operand is part of the instruction (a specific field) and no further memory reference is required.
PC-Relative Addressing	The operand is an offset relative to the current value of the PC.
Subroutine	Similar to a C function. Parameters, if any, can be passed in registers or on the Stack.
Side effect	A change in the CPUs state (such as a register) unrelated to the subroutine's function.
Parameter	An argument passed to a subroutine. They are usually passed in registers or on the stack.
Direction Register	A control register associated with a parallel port that configures individual bits to be inputs or outputs.
Memory Mapped Device	A peripheral device whose internal registers are mapped to specific memory locations.

Bcd ascii