Module 13 Example Set 1

1. Shown below is an algorithm for determining the greatest common divisor (GCD) of two integers A and B each of which is greater than zero:

```
Set X = A
Set Y = B
While(X ≠ Y)
{
   if X>Y   set X=X-Y
  else if X<Y set  Y=Y-X
}
GCD = X
```

Show how a relatively small number of ARM conditional execution instructions can implement this algorithm.

Answer: Let R2 contain one number and R3 contain the other.

```
LOOP  CMP     R2,R3       ; compare number to set condition codes
      SUBGT   R2,R2,R3    ; if R2>R3,  R2 = R2 – R3
      SUBLT   R3,R3,R2    ; else if R3>R2,  R3 = R3 – R2
      BNE     LOOP        ; repeat loop until  R2 = R3
```

The CMP instruction sets the condition codes based on which register contains a larger number. SUBGT only executes if the condition codes set by CMP indicate that R2>R3. SUBLT only executes if the condition codes set by CMP indicate that R3>R2. BNE does not branch if the condition codes indicate that R3 = R2, in which case the branch back to LOOP is not taken and the GCD is contained in the two registers R3 and R2.

2. What are some of the advantages of the banked registers available on ARM processors compared to other RISC processors that do not employ banked registers?
The banked registers reduce the need to save and store registers to and from the stack, thus speeding up the processing of exceptions.

3. How does the process of identifying the proper exception handling routine differ on the ARM processor from that used on the MIPS processor?
On the MIPS processor all exceptions cause a transfer to the same exception vector address. Code that resides at that address must examine the cause register to determine the appropriate handling routine to call for the type of exception. On the ARM processor, each exception type is vectored to a different location in low memory. Each location contains an instruction that branches to the corresponding exception handling routine.

4. The ARM processor, unlike the MIPS processor, supports a PC-relative addressing mode for memory operands. What is a major disadvantage of this addressing mode?
One disadvantage is that the memory operands must be located in the same area as the program code and must be close enough to referenced using an 8-bit displacement. Using instead an addressing mode such as register indirect or base register mode would allow the operand to be located anywhere within the 32-bit address space.

5. What effect is produced by the following ARM instruction?

```
SUBS   PC,R14_irq,#4
```

When the result register is the PC, the S suffix causes the processor to copy the contents of SPSR_irq into CPSR. Also the instruction subtracts 4 from R14_irq and places the result into the PC causing control to be returned to the instruction that was about to execute when the interrupt occurred.

6. How does the process of enabling interrupts on the MIPS and ARM processors differ?
To enable interrupts, both processors have to be running in privileged mode.
On the MIPS, the status register (CP0 register 12) can be read and written using the instructions mfc0 and mtc0. Setting the LSB of CP0 $12 enables interrupts.
With the ARM processor running in priviledge mode, special Move instructions, MRS and MSR, can be used to transfer the contents of the processor status register to or from a CPU general purpose register.

```
MRS    R2,CPSR        ;  copies CPSR into R2

MSR    SPSR,R3        ; copies R3 into SPSR_mode
```

After status register contents have been loaded into a CPU register, logic instructions can be used to manipulate individual bits.

7. What is the purpose of the ARM instruction ADR?
This instruction loads into a register the address of operands that can be referenced using PC-relative addressing. That is, operands whose address can be generated adding or subtracting an 8-bit unsigned displacement (possibly shifted) to or from the PC. E.g. ADR R2,DATA1 places the PC-relative address of DATA1 into R2.