Spring 2003 Name: Mark F.

Computer Organization Test 2

Question 1. (25 points) Translate the following high-level language code segment to ARM assembly language. Use the registers indicated in the code.

a)
```
for R0 = 0 to 100 by steps of size 10 do
    if (R3 < R0) AND (R2 >= 50) then
        R2 = R2 + R3
    end if
end for
```

```
FOR_INIT   MOV R0, #0
FOR_CMP    CMP R0, #100
           BGT END-FOR

IF         CMP R3, R0
           BLE END_IF
           CMP R2, #50
           BLT END_IF
THEN       ADD R2, R2, R3

END_IF
                ADD R0, R0, #10
                B   FOR_CMP
END_FOR
```

b)
```
while (R8 > 20) do
    if (R8 < 100)  OR (R8 > 200) then
        R7 = R8
        R8 = R8 - 10
    else
        R8 = R8 - R7
    end if
    R7 =  R6 + 4
end while
```

```
WHILE    CMP R8, 20
         BLE END_WHILE

IF       CMP R8, #100
         BLT THEN
         CMP R8, #200
         BLE ELSE
THEN
         MOV R7, R8
         SUB R8, R8, #10

             B END_IF
ELSE
             SUB R8, R8, R7
END_IF
             ADD R7, R6, #4

             B WHILE
END_WHILE
```

Question 2. (10 points) Suppose you have the following data AREA in ARM assembly language:

```
              [0] [1] 2  3  4 [5]                                  [15]
              AREA DATA, READWRITE
ARRAY         DCD  10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25
N             DCD  5
POINTER       DCD  ARRAY
              END
```

We did "ADR R1, ARRAY" this semester.

For each of the following assembly language segments, what value is loaded into register R2?

```
a)  LDR R0, N                        c)  LDR R0, N
    LDR R1, POINTER                      LDR R1, POINTER
    LDR R2, [R1, R0, LSL #2]             LDR R2, [R1, R0, LSL #3]
      15                                   20
                                                     5 * 2^3 = 5x8 = 40 bytes
                                                     index [10]
b)  LDR R1, POINTER                  d)  MOV R0, #16
    MOV R4, #7                           LDR R1, POINTER
    ADD R3, R1, R4, LSR #2               LDR R2, [R1, R0, LSL #2]
    LDR R2, [R3]    0111₂ = 7₁₀            5
      10            0                        16 * 2^2 = 64 bytes
                                             index [16] <- Does not
                                             exist so grabs value of N
```

Question 3. (7 points) For the data AREA in question 2, complete the translation of the following high-level code segment to ARM assembly language.

```
for i = 0  to 14 do                      LDR R1, POINTER
    array[i+1] = array[i]                MOV R0, #0
end for                          FOR     CMP R0, #14
                                         BGT  END_FOR
                                         LDR R2, [R1, R0, LSL #2]
                                         ADD R0, R0, #1
                                         STR R2, [R1, R0, LSL #2]
                                         B  FOR
                                 END_FOR
```

Question 4. (8 points) The ARM Compare instruction "CMP R2, R3" sets the condition codes (N, Z, C, V bits) according to the result of (R2 - R3). For the ARM conditional-branch instruction "BLT LABEL" (branch less than), what must the condition code values be inorder for the branch to be taken?

N = 1

Z = 0

Question 5. Consider the following selection sort subprogram that utilizes a function Max to search for the largest element in the unsorted part of the array.

```
procedure selectionSort(numbers - array of integers, count - integer)
    local integer variables: lastUnsortedIndex, maxIndex, temp

    for lastUnsortedIndex = (count-1) downto 1 do
        maxIndex = Max(numbers, 0, lastUnsortedIndex)
        temp = numbers[lastUnsortedIndex]
        numbers[lastUnsortedIndex] = numbers[maxIndex]
        numbers[maxIndex] = temp
    end for

end selectionSort
```

a) (6 points) Using the ARM register conventions (a1-a4, v1-v6, sp, lr, pc, etc.), what registers would be used to pass each of the following parameters to selectionSort:

base address of "numbers" array	count
a1	a2

b) (6 points) Using the ARM register conventions, which of these parameters ("numbers", "count", or both of them) should be moved into v-registers? numbers a1 → v1
count's value only needed before call to Max.

c) (6 points) Using the ARM register conventions, what registers should be used for each of the local variables:

lastUnsortedIndex	maxIndex	temp
v2	a1	a2

d) (4 points) In addition to the above registers, the value of "numbers[maxIndex]" will need to be stored into a register. Using the ARM register conventions, what register should be used to hold this value? a3

e) (8 points) For the registers indicated above, write the STMFD and LDMFD instructions which would be the first and last instructions in the subprogram selectionSort.

```
STMFD sp!, {v1,v2,lr}
LDMFD sp!, {v1,v2,pc}
```

f) (10 points) For the registers indicated above, write the assemble language code to call the Max function ("maxIndex = Max(numbers, 0, lastUnsortedIndex)"). Include the ARM instructions to setup the parameters to Max and assigning "maxIndex" the value returned. (You do not need to write the Max function code just the code to call it)

```
MOV a1, v1
MOV a2, #0
MOV a3, v2
BL Max      ; returns result in a1
            ; nothing to do since maxIndex chosen to be a1
```

g) (10 points) Using the registers you indicated, write the ARM assembly language statements to perform the statements:

```
temp = numbers[lastUnsortedIndex]
numbers[lastUnsortedIndex] = numbers[maxIndex]
numbers[maxIndex] = temp
```

```
LDR a2, [v1, v2, LSL #2]
LDR a3, [v1, a1, LSL #2]
STR a3, [v1, v2, LSL #2]
STR a2, [v1, a1, LSL #2]
```