

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

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

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

```

                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

```

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

- | | |
|--|---|
| a) LDR R0, N
LDR R1, POINTER
LDR R2, [R1, R0, LSL #2] | c) LDR R0, N
LDR R1, POINTER
LDR R2, [R1, R0, LSL #3] |
| b) LDR R1, POINTER
MOV R4, #7
ADD R3, R1, R4, LSR #2
LDR R2, [R3] | d) MOV R0, #16
LDR R1, POINTER
LDR R2, [R1, R0, LSL #2] |

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

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 in order for the branch to be taken?

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

b) (6 points) Using the ARM register conventions, which of these parameters ("numbers", "count", or both of them) should be moved into v-registers?

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

lastUnsortedIndex	maxIndex	temp

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?

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.

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)

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
```