Dr. Bassem Al-Halabi, S&E362 March 6, 2000

Name: Grade: **/10**

Arithmetic Functions

[4] 1) One method of fault-tolerant computing is to perform a certain operation 3 times on 3 different machines and compare the 3 outputs. If any two are equal, then, their value is assumed to be the correct one and the third is automatically reset to the correct one. Write a complete 68000 assembly language subroutine, CORRECT, which examines three registers, D1, D2, and D3 as inputs. If any two are equal, the third is set to the same value. Otherwise, all three registers are cleared. Make sure no other registers are changed in your subroutine.

[3] 1) Write a 68000 assembly language subroutine, FUNCTION, to implement the following arithmetic function. All numbers are signed and word size. Ignore possible overflow conditions. The results are returned in D0. Make sure the original contents of all registers remain unchanged except for D0.

$$\mathbf{D0} = \frac{(\mathbf{D1} \times \mathbf{D2}) + (\mathbf{D3})^{\mathbf{2}}}{\mathbf{D4} + \mathbf{D5}}$$

[6] 2) Write a 68000 assembly language subroutine, SUM, to implement the following arithmetic summation function. The summation limits are passed to the subroutine via D0 and D1. Both of them are positive words and D1 is greater than D0. The summation results are returned in D2. Ignore possible overflow conditions. All registers must remain unchanged except D2.

$$\mathbf{D2} = \sum_{\mathbf{i}=\mathbf{D0}}^{\mathbf{i}=\mathbf{D1}} \left(\mathbf{4i}^{\mathbf{4}} + \mathbf{2i}^{\mathbf{2}}\right)$$

[4] 1) Write a 68000 assembly language subroutine, SUM, which adds the elements of an array pointed to by A0. The length of the array is passed to the subroutine in D0. The resulting sum is returned in D1.

Logical Functions

[6] 2) Write a 68000 assembly language subroutine, OR-HEX, which performs the bit-wise AND logical operation on the 8 hex digits of register D0. Results are returned in the least significant hex position of D0.

Bit Manipulation

[5] 2) Write a 68000 assembly language subroutine, FLIPBYTES, which flips the 4 bytes of D0 right-side-left. In other words, the 4 bytes of D0 are reordered in sequence so that the least significant byte becomes the most significant one.

[4] 1) Write a 68000 assembly language subroutine, MIRROR, to examine the contents of register D0. If the two word patterns of D0 are mirror images of each other, the subroutine will return 1s on all bits of D0. Otherwise, D0 will be all cleared.

[7] 2) Write a 68000 assembly language subroutine, FLIPBYTE, which flips (left-side-right) the contents of each of the 4 bytes of D0 individually. For example if the contents of D0 is 1123CC34, the results would be 88C4332C.

Dr. Bassem Al-Halabi, S&E362 March 6, 2000

Name: Grade: **/10**

Sequence Detection

[5] **1)** Write a 68000 assembly language subroutine to examine the contents of register D0. If, at any position in the register, there is a sequence of ..1011.., the subroutine will set D1 to all ones, otherwise, D1 will be all zeros. Overlap is allowed.

[5] **1)** Write a 68000 assembly language subroutine, FIND, to examine the contents of register D0. If at any position in D0, there are two 1's separated by exactly two consecutive 0's, the program returns all 1's in D0, otherwise, D0 is all cleared.

[5] **1)** Write a 68000 assembly language subroutine, FIND, to examine the contents of register D0. If at any position in D0, there are two consecutive 1's followed by two consecutive 0's, the program returns all 1's in D0, otherwise, D0 is all cleared.

Data Arrays

[5] **2)** Write a 68000 assembly language subroutine, MIN, which finds the smallest number in an array of ten elements pointed to by A0. The subroutine returns the smallest number in D0.

[7] **3)** Write a 68000 assembly language subroutine, REP_FREE, to examine the elements of a list of positive byte-size numbers stored at location LIST_IN. The list is already sorted in an ascending order and it ends with an asterisk (*). The subroutine will copy the elements from location LIST_IN to location LIST_OUT. While copying, if an element occurs more than once (replicated), then it is copied only one time to LIST_OUT. In short, the subroutine eliminates the replicated elements from LIST_IN and places the results in LIST_OUT.

[8] **2)** Write a 68000 assembly language subroutine, FILTER, to examine the elements of a list of positive word-size numbers stored at location SAMPLE. These numbers are actually the digitized values of a sample of analog waveform. The first word of the list is the desired cut-off level and the second word is the length of the list. The subroutine will scan the whole list and replace all values which are less than the cut-off value by the cut-off value itself.

[8] **1)** Write a 68000 assembly language subroutine, ROTMEM, which rotates the elements (words-size) of an array stored at location pointed to by (A0). The depth of the array is stored as the first element. The amount of rotation is passed to the subroutine in D0. The rotated array is kept in the same place in memory. Make sure no registers are changed.