EECS 388: Computer Systems and Assembly Language
Homework 1
Due Feb. 12, 2007
Justify your answers!

Figure 1 shows a part of the memory (both contents and locations).

Contents	Locations
$20	$4000
$50	$4001
$01	$4002
: :	
: :	
$B5	$5000
$CD	$5001

Figure 1.

Problem 1 (10 Points):
Consider the memory shown in Figure 1. Write a program to add the two numbers in $4000 and $4001 and store the results in $5002.

Here, we simply need to add two 8-bit numbers and store the 8-bit result, so we can use accumulator A for this task:

```
ORG     $4100
LDAA    $4000
ADDA    $4001
STAA    $5002
SWI
```

Problem 2 (10 Points):
Consider the memory shown in Figure 1. What is the content in accumulator D after the following lines are executed?

```
LDD     $4001
ADDA    $5000
```

First we load D with the 16-bit value starting at location $4001. At this point, we have

D: $5001

Next we add the 8-bit value at location $5000 to the value contained in A (which is the most significant byte of D, $50):

A: $50 + $B5 = $05 (The real result is $105)

The above result is too large to fit in A, so A will contain $05 and the overflow bit in the CCR will be set.
Consequently, the upper byte of D (which is A) is now $05 but the lower byte has not changed at all, giving us:

D: $0501

Problem 3 (30 points):
Consider the memory shown in Figure 1. What are the contents in memory location $5002 and accumulator A after the following lines are executed?

```
LDAA    $4000
LDD     #$5000
LDAB    $4002
STD     $5002
```

First, we load accumulator A with the 8-bit value at location $4000, which gives us:

A: $20

Next, we load D with the immediate 16-bit value $5000, which now gives us:

D: $5000

But since D is really A and B, this means:

A: $50
B: $00

Next we load accumulator B with the 8-bit value at location $4002, which gives us:

B: $01

So now we have:

A: $50
B: $01
D: $5001

Finally we store the 16-bit value in D to memory location, starting at location $5002, so now memory will look like:

Contents	Locations
$20	$4000
$50	$4001

\$01		\$4002
	: :	
	: :	
\$B5		\$5000
\$CD		\$5001
\$50		\$5002
\$01		\$5003

Therefore, our final answer is the following:

Memory location \$5002: \$50
A: \$50

Problem 4 (25 points):
If A contains \$BB, B contains \$CD and the carry bit in CCR is 1, what are the results of the following instructions? Assume that A, B, and CCR are restored to their original values before each instruction.

a) ASLA

We are told that A contains \$BB at the start, which is:

\$BB: 1011 1011

Next, we perform an arithmetic shift to the left,

Original(hex): \$ B B
Original(bin): 1011 1011
Shifted(bin): 0111 0110
Shifted(hex): \$ 7 6

So the answer is:

A: \$76

b) ASRB

Original(hex): \$ C D
Original(bin): 1100 1101
Shifted(bin): 1110 0110
Shifted(hex): \$ E 6

Final answer:

B: \$E6

c) LSLD

Original(hex): $ B B C D
Original(bin): 1011 1011 1100 1101
Shifted(bin): 0111 0111 1001 1010
Shifted(hex): $ 7 7 9 A

Final answer:

A: $77 **B: $9A**

d) ROLB

Remember that B is the lower byte of D, and that the carry bit is 1 before the rotate:

Original(hex): $ B B C D
Original(bin): 1011 1011 1100 1101
Shifted(bin): 1011 1011 1001 1011
Shifted(hex): $ B B 9 B

The upper byte of D (which is A) is not affected by the rotate.
Final answer:

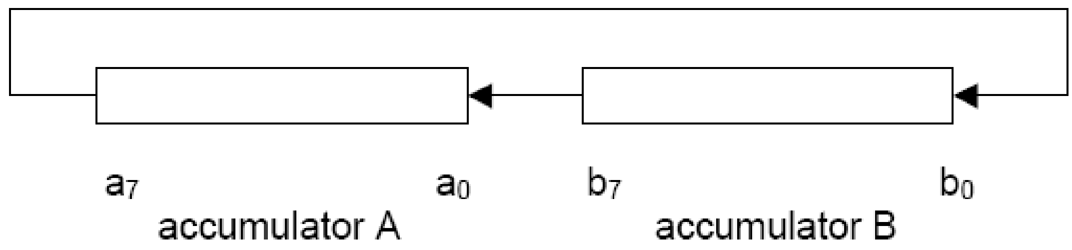
D: $BB9B

Problem 4 (10 Points):
Write a program to add two values $20 and $40 and store the result in memory location $8000.

We are asked to add to immediate 8-bit values and store the 8-bit result in memory.

```
ORG     $4000
LDAA    #$20
ADDA    #$40
STAA    $8000
SWI
```

Problem 5 (15 Points):
Write a program to implement the following 16 bit rotation, i.e., originally, A and B contain $\boldsymbol{a}_7\boldsymbol{a}_6\ldots\boldsymbol{a}_0$ and $\boldsymbol{b}_7\boldsymbol{b}_6\ldots\boldsymbol{b}_0$, respectively. After the 16 bit rotation, the contents in A and B are $\boldsymbol{a}_6\boldsymbol{a}_5\ldots\boldsymbol{a}_0\boldsymbol{b}_7$ and $\boldsymbol{b}_6\boldsymbol{b}_5\ldots\boldsymbol{b}_0\boldsymbol{a}_7$, respectively. (Note: You can only use the data transfer and manipulation instructions in Section 2.4.1).



There are a lot of ways to do this problem; here are a few solutions (we only show the instructions for brevity):

Solution 1:

```
ASRA
ROLA
ROLB
ROLA
```

Solution 2:

```
ASRB
ROLB
ROLA
ROLB
```

Solution 3:

```
STAB    $4000
LSLD
LDAB    $4000
ROLB
```

Solution 4:

```
STAA    $4000
ROLA
ROLB
LDAA    $4000
ROLA
```

Solution 5:

LSLB
ROLA
RORB
ASRB
ROLB
ROLB