



[**Open book, pink book, handouts, and notes**] Show all of your work clearly in the space provided or on the additional pages at the end of the exam. If the additional pages are used, clearly identify to which exam question it is related. Be sure to **read each problem carefully**. You should answer all 4 questions, and you may wish to answer the bonus question if you have time. Note that the exam is double sided.

1. (10 points) The instruction `oraa 0,x` requires four clock cycles to complete whereas the instruction `orab 72,y` requires five clock cycles. Why?

2. (10 points) Indicate the contents of the stack (and the memory locations associated with the contents) after the following operations.

```
lds  #0xF000
ldd  #0x30F2
ldx  #0x1234
psha
pshb
pshx
pulb
pshx
psha
puly
```



3. (20 points) To the right of each statement, indicate the value contained by each of the condition code registers. Assume that the A accumulator initially contains 0x00. For partial credit, show your work.

Instruction	S	X	H	I	N	Z	V	C
tap								
ldaa #0x8B								
adda #0xAA								
ldab #0								
decb								
clra								
comb								
des								



4. In this multi-part problem you will write a program that adds `count` 24-bit numbers and stores the result as a 32-bit number in a 4 byte memory location.

(a) (10 points) Write the appropriate assembler directives to:

- store the following 24-bit numbers in memory: `0x03F12D`, `0x055FFC`, ..., `0x1784D6` with a label `num` pointing to the most significant byte of the first number;
- reserve space for the 32-bit result; and
- create a label `ans` that points to the most significant byte of the 32-bit result.



(b) (15 points) Write a subroutine which adds a 24-bit number to a 32-bit number and stores the result in the location of the 32-bit number. Assume that the X index register points to the most significant byte of the 24-bit number and that the Y register points to the most significant byte of the 32-bit number.



(c) (15 points) Write the body of the main program that will use the subroutine in part (b) to add `count` 24-bit numbers and store the result in four bytes of memory beginning at `ans`. (Note: `count` is a label that will be defined in part (d).)



(d) (10 points) Write the source (`exam1.s`) file for the above code where `count=36`. Be sure to include all necessary assembler directives and initialization instructions not included in parts (a) – (c). For convenience, you may signify the contents of your answers to the previous parts by the part letter with a box around it.



(e) (10 points) Write the link (`exam1.lnk`) file for the source file in part (d) such that the data is stored beginning at `0xD000` and the program is stored beginning at `0xC000`.



bonus (10 points) Rewrite the subroutine in **4 (b)** so that it can add together arbitrarily large (up to 256 bytes long) numbers. Assume that the number of bytes required for the number is stored in the A accumulator and that the result to be stored consists of an additional byte.



Additional work area for any problem. Clearly identify to which problem the work on this page is related.



Additional work area for any problem. Clearly identify to which problem the work on this page is related.