1a. The program loads the value 11 into R4, and executes the loop while R4 remains non-negative. In each execution of the loop, R4 is decremented by 3.
1b. 4 times.

2a. The program loads the value x0803 into R1, and executes the loop while R1 is positive. In each execution of the loop, 4 is added to R2. The net effect is multiplying x0803 by x0004.

2b. x200C = #8204.

2c.

//Symbol Name	Page Address
// DONE	3007
// LOOP	3003
// M0	300B
// M1	300C
// RESULT	3009
// ZERO	300A

3a. The loop must terminate when R2 reaches 0. But the instruction "ADD R3,R3,R3" changes the condition code so the branch is not correctly taken.

3b. Switch the instructions in the following manner:

From:

```
LOOP BRz DONE

ADD R2, R2, #-1

ADD R3, R3, R3

BR LOOP
```

to:

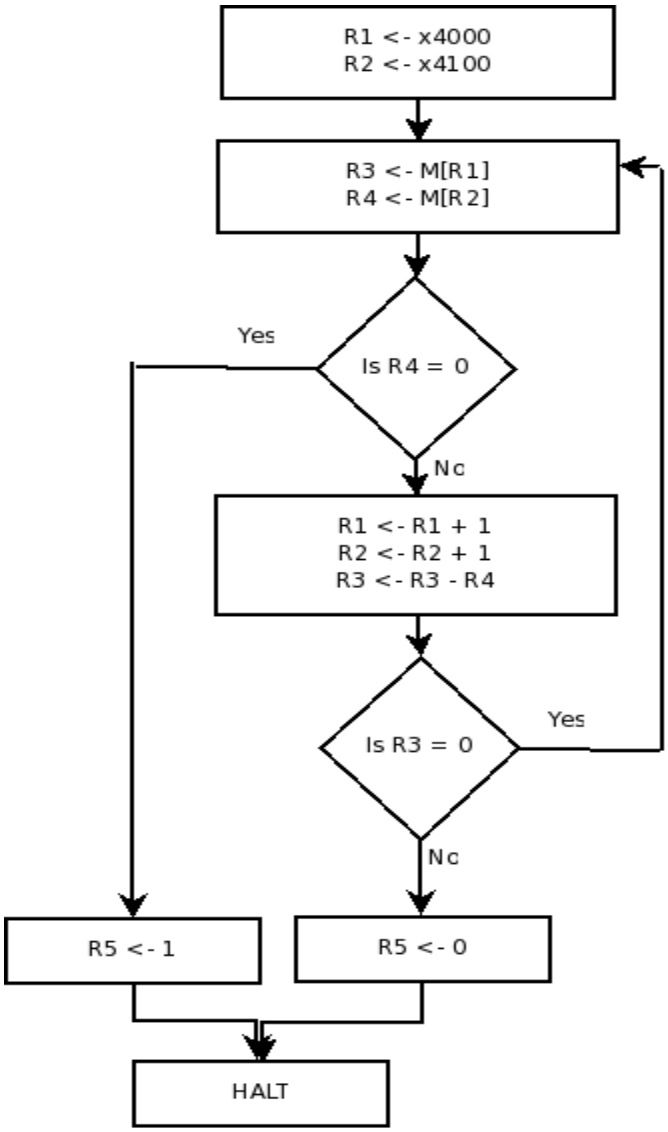
LOOP BRz DONE

ADD R3, R3, R3

ADD R2, R2, #-1

BR LOOP

4a.



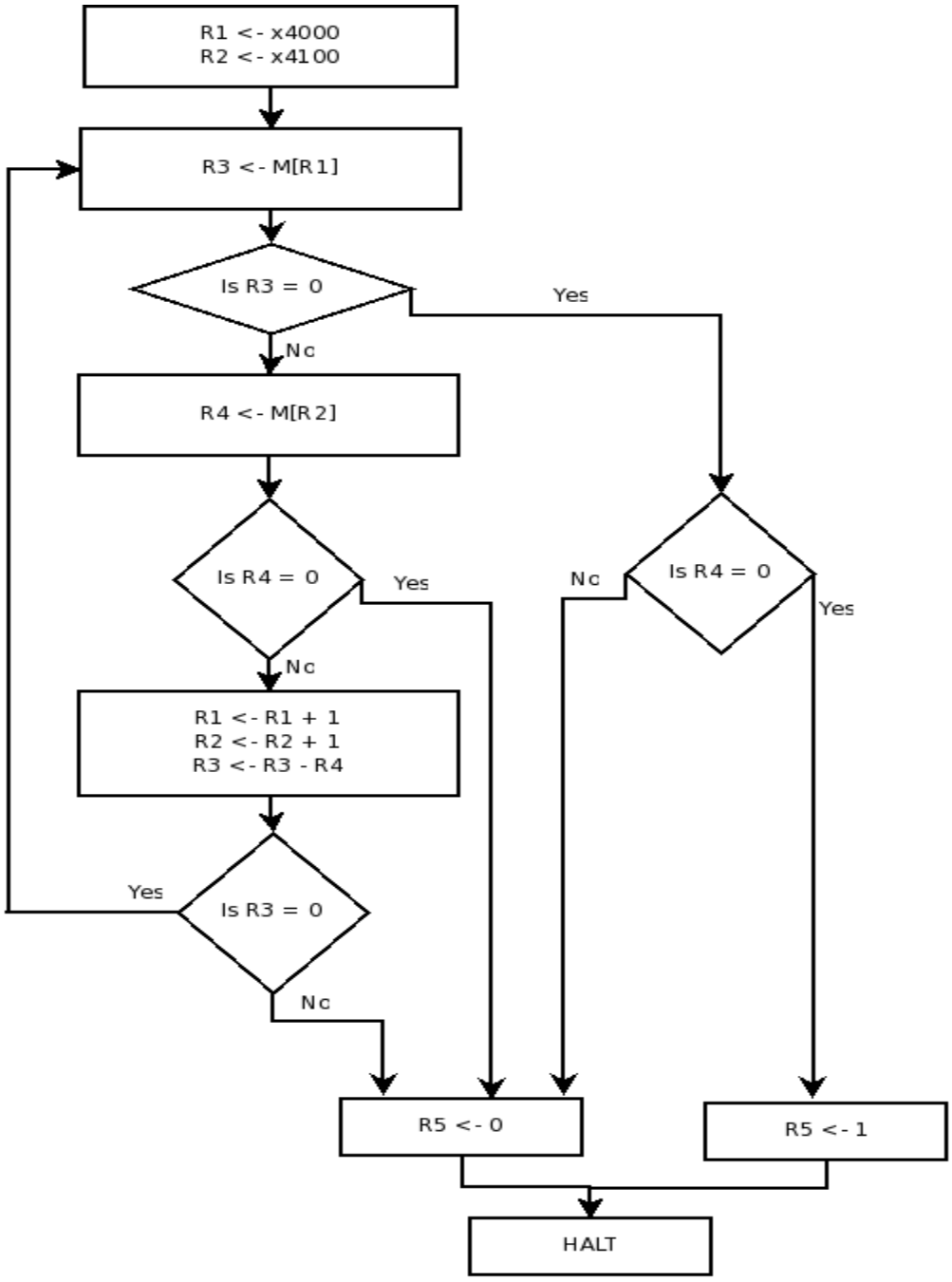
4b.

(a) LDR R3, R1, #0

(b) NOT R4, R4

(c) ADD R4, R4, #1

4c.



```
.ORIG x3000
            LD R1, FIRST
            LD R2, SECOND

LOOP        LDR R3, R1, #0
            BRz S1ZERO
            ; first string not ended
            LDR R4, R2, #0
            BRz DIFF
            ; both strings not ended
            ADD R1, R1, #1
            ADD R2, R2, #1
            NOT R4, R4
            ADD R4, R4, #1
            ADD R3, R3, R4
            BRz LOOP
;this is executed if both strings are different
DIFF        AND R5, R5, #0
            ADD R5, R5, #1
            BRnzp  DONE

;this is executed if both strings are the same
SAME        AND R5, R5, #0
```

```
                BRnzp DONE

;this is executed if first string has ended
S1ZERO      LDR R4, R2, #0
                BRz SAME
                BRnzp DIFF

DONE        TRAP x25
FIRST  .FILL x4000
SECOND .FILL x4100
.END
```

5.

```
.orig x3000
        LDI R1, X
        LDI R2, Y
        NOT R2, R2
        ADD R2, R2, #1
        AND R3, R3, #0

LOOP  ADD R1, R1, R2
        BRn DONE
        ADD R3, R3, #1
        BR  LOOP
```

```
DONE  HALT


X       .fill x3200

Y       .fill x3201

.end
```