ערעור רטוב 2 – חלק 1

מגישים: רועי מרום ותואם אלהרר

על מנת לעבור את $runtime\ test$ החלפנו את הלולאה $count\_1\_bit\_loop$ בפונקציה $hemming\_weight$. אשר בודקת כמה ביטים דולקים יש ברגיסטר $\%r9$, בפקודה $popcnt$. (שמנו את החלק אשר החלפנו בהערה בקוד).

הקוד לפני השינוי:

```
# parameters: rdi = unsigned long* codeword ; rsi = unsigned long len #
hamming_weight:
    # calee convention -PROLOGUE #
    pushq %rbp
    movq %rsp, %rbp

    xor %r8, %r8 # r8 is the loop index
    xor %rax, %rax # 1-bit counter
    memory_to_register_loop:
        cmp %rsi, %r8
        je finish
        movq (%rdi, %r8, 8), %r9 # r9 is 8 byte section of codeword
        count_1_bits_loop:
            cmp $0, %r9
            je outer_counter
            shr $1, %r9 # the bit is now in the carry flag
            jnc skip_add
            inc %rax
            skip_add: jmp count_1_bits_loop
        outer_counter:
        inc %r8
        jmp memory_to_register_loop

    finish:
    # EPILOGUE #
    movq %rbp, %rsp
    popq %rbp
    ret
```

הקוד לאחר השינוי:

```
# parameters: rdi = unsigned long* codeword ; rsi = unsigned long len #
hamming_weight:
    # calee convention -PROLOGUE #
    pushq %rbp
    movq %rsp, %rbp

    xor %r8, %r8 # r8 is the loop index
    xor %rax, %rax # 1-bit counter
    memory_to_register_loop:
        cmp %rsi, %r8
        je finish
        movq (%rdi, %r8, 8), %r9 # r9 is 8 byte section of codeword
        popcnt %r9, %rdx # THE CHANGE - using POPCNT to count the number of 1 bits.
        add %rdx, %rax
       # count_1_bits_loop:
        #    cmp $0, %r9
        #    je outer_counter
        #    shr $1, %r9 # the bit is now in the carry flag
        #    jnc skip_add
        #    inc %rax
        #    skip_add: jmp count_1_bits_loop
        outer_counter:
        inc %r8
        jmp memory_to_register_loop

    finish:
    # EPILOGUE #
    movq %rbp, %rsp
    popq %rbp
    ret
```