

```

; mul 8x8 16 bit result for when you can't afford big tables
; by djmips
;
; inputs are mul1 and mul2 and should be zero page.
; A should be zero entering but if you want it will factor in as 1/2 A added
to the result.
;
; output is 16 bit in A : mul1 (A is high byte)
;
; length = 65 bytes
; total cycles worst case = 113
; total cycles best case = 97
; avg = 105
; inner loop credits supercat

```

MUL:

```

    dec mul2    ;5 ; decrement mul2 because we will be adding with carry
set for speed (an extra one)
    ror mul1    ;5 \
    bcc b1     ;2/3 \ Best case 8 Worst case 10
    adc mul2    ;3 /
b1: ror        ;2 \
    ror mul1    ;5 \
    bcc b2     ;2/3 / Best case 10 Worst case 12
    adc mul2    ;3 /
b2: ror
    ror mul1
    bcc b3
    adc mul2    ; 10 or 12
b3: ror
    ror mul1
    bcc b4
    adc mul2    ; 10 or 12
b4: ror
    ror mul1
    bcc b5
    adc mul2    ; 10 or 12
b5: ror
    ror mul1
    bcc b6
    adc mul2    ; 10 or 12
b6: ror
    ror mul1
    bcc b7
    adc mul2    ; 10 or 12
b7: ror
    ror mul1
    bcc b8
    adc mul2    ; 10 or 12
b8: ror        ; 2
    ror mul1    ; 5

```

```
inc mul2 ; 5
rts
```

From: <https://codebase64.org/> - Codebase 64 wiki

Permanent link: https://codebase64.org/doku.php?id=base:8bit_multiplication_16bit_product_fast_no_tables&rev=1580677768

Last update: **2020-02-02 22:09**

