Der Assemblercode ohne die Optimierung:

```assembly
multacc(unsigned long*, unsigned long*, unsigned long*, unsigned int):
    cmp     r3, #0
    bxeq    lr
    add     ip, r1, #4
    push    {r4, r5, r6, r7, r8, r9, lr}
    sub     ip, r0, ip
    sub     lr, r3, #1
    cmp     ip, #8
    cmphi   lr, #2
    add     lr, r2, #4
    movhi   ip, #1
    movls   ip, #0
    sub     lr, r0, lr
    cmp     lr, #8
    movls   ip, #0
    andhi   ip, ip, #1
    cmp     ip, #0
    beq     .L8
    lsr     r5, r3, #2
    mov     ip, r0
    mov     r4, r1
    add     r5, r0, r5, lsl #4
    .L9:
    vld1.32 {q10}, [r4]!
    vld1.32 {q9}, [lr]!
    vld1.32 {q8}, [ip]
    vmla.i32 q8, q10, q9
    vst1.32 {q8}, [ip]!
    cmp     ip, r5
    bne     .L9
    bic     ip, r3, #3
    cmp     r3, ip
    popeq   {r4, r5, r6, r7, r8, r9, pc}
    ldr     r4, [r1, ip, lsl #2]
    ldr     r5, [r2, ip, lsl #2]
    ldr     lr, [r0, ip, lsl #2]
    mla     lr, r5, r4, lr
    str     lr, [r0, ip, lsl #2]
    add     lr, ip, #1
    cmp     r3, lr
    lsl     lr, ip, #2
    poplps  {r4, r5, r6, r7, r8, r9, pc}
    add     ip, ip, #2
    cmp     r3, ip
    add     r3, lr, #4
    ldr     r4, [r1, r3]
    ldr     r5, [r2, r3]
    ldr     ip, [r0, r3]
    mla     ip, r5, r4, ip
    str     ip, [r0, r3]
```
popls  {r4, r5, r6, r7, r8, r9, pc}
add   r3, lr, #8
ldr   ip, [r2, r3]
ldr   r1, [r1, r3]
ldr   r2, [r0, r3]
mla   r2, ip, r1, r2
str   r2, [r0, r3]
pop   {r4, r5, r6, r7, r8, r9, pc}

.L8:
cmp   r3, #8
bls   .L15
sub   lr, r3, #9
add   r4, r0, #64
bic   lr, lr, #7
add   r5, r1, #64
add   lr, lr, #8
mov   r6, r2

.L12:
ldr   r8, [r5, #64]
add   ip, ip, #8
ldr   r9, [r6]
cmp   ip, lr
ldr   r7, [r4, #64]
add   r6, r6, #32
pld  [r5]
add   r5, r5, #32
pld  [r4]
add   r4, r4, #32
mla   r7, r9, r8, r7
str   r7, [r4, #96]
ldr   r7, [r4, #92]
ldr   r8, [r5, #92]
ldr   r9, [r6, #28]
mla   r7, r9, r8, r7
str   r7, [r4, #92]
ldr   r7, [r4, #88]
ldr   r8, [r5, #88]
ldr   r9, [r6, #24]
mla   r7, r9, r8, r7
str   r7, [r4, #88]
ldr   r7, [r4, #84]
ldr   r8, [r5, #84]
ldr   r9, [r6, #20]
mla   r7, r9, r8, r7
str   r7, [r4, #84]
ldr   r7, [r4, #80]
ldr   r8, [r5, #80]
ldr   r9, [r6, #16]
mla   r7, r9, r8, r7
str   r7, [r4, #80]
ldr   r7, [r4, #76]
ldr   r8, [r5, #76]
ldr   r9, [r6, #12]
mla   r7, r9, r8, r7
str        r7, [r4, #-76]
ldr        r7, [r4, #-72]
ldr        r8, [r5, #-72]
ldr        r9, [r6, #-8]
mla        r7, r9, r8, r7
str        r7, [r4, #-72]
ldr        r8, [r5, #-68]
ldr        r9, [r6, #-4]
mla        r7, [r4, #-68]
str        r7, [r4, #-68]
bne        .L12

.L11:
    lsl        ip, lr, #2
    add        r0, r0, ip
    add        r1, r1, ip
    add        r2, r2, ip

.L13:
    ldr        ip, [r0]
    add        lr, lr, #1
    ldr        r4, [r1], #4
    cmp        r3, lr
    ldr        r5, [r2], #4
    mla        ip, r5, r4, ip
    str        ip, [r0], #4
    bhi        .L13
    pop        {r4, r5, r6, r7, r8, r9, pc}

.L15:
    mov        lr, ip
    b          .L11