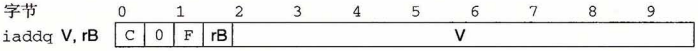

该指令将常数值 V 与寄存器 rB 相加。

使用 iaddq 指令重写图 4-6 的 Y86-64 sum 函数。在之前的代码中，我们用寄存器%r8和%r9来保存常数值。现在，我们完全可以避免使用这些寄存器。

```
                          | # Execution begins at address 0
0x000:                    |   .pos 0
0x000: 30f40002000000000000 |   irmovq stack, %rsp      # Set up stack pointer
0x00a: 803800000000000000 |   call main               # Execute main program
0x013: 00                 |   halt                    # Terminate program
                          |
                          | # Array of 4 elements
0x018:                    |   .align 8
0x018:                    | array:
0x018: 0d000d000d000000   |   .quad 0x000d000d000d
0x020: c000c000c0000000   |   .quad 0x00c000c000c0
0x028: 000b000b000b0000   |   .quad 0x0b000b000b00
0x030: 00a000a000a00000   |   .quad 0xa000a000a000
                          |
0x038:                    | main:
0x038: 30f71800000000000000 |   irmovq array,%rdi
0x042: 30f60400000000000000 |   irmovq $4,%rsi
0x04c: 805600000000000000 |   call sum                # sum(array, 4)
0x055: 90                 |   ret
                          |
                          | # long sum(long *start, long count)
                          | # start in %rdi, count in %rsi
0x056:                    | sum:
0x056: 30f80800000000000000 |   irmovq $8,%r8         # Constant 8
0x060: 30f90100000000000000 |   irmovq $1,%r9         # Constant 1
0x06a: 6300               |   xorq %rax,%rax          # sum = 0
0x06c: 6266               |   andq %rsi,%rsi          # Set CC
0x06e: 707000000000000000 |   jmp     test            # Goto test
0x077:                    | loop:
0x077: 50a70000000000000000 |   mrmovq (%rdi),%r10    # Get *start
0x081: 60a0               |   addq %r10,%rax          # Add to sum
0x083: 6087               |   addq %r8,%rdi           # start++
0x085: 6196               |   subq %r9,%rsi           # count--.  Set CC
0x087:                    | test:
0x087: 747000000000000000 |   jne    loop           # Stop when 0
0x090: 90                 |   ret                     # Return
                          |
                          | # Stack starts here and grows to lower addresses
0x200:                    |   .pos 0x200
0x200:                    | stack:
```

图 4-8　YAS 汇编器的输出。每一行包含一个十六进制的地址，以及字节数在 1～10 之间的目标代码