void wait(uint16\_t millis)

{

258: cf 93 push r28

25a: df 93 push r29

25c: cd b7 in r28, 0x3d ; 61

25e: de b7 in r29, 0x3e ; 62

260: 2a 97 sbiw r28, 0x0a ; 10

262: 0f b6 in r0, 0x3f ; 63

264: f8 94 cli

266: de bf out 0x3e, r29 ; 62

268: 0f be out 0x3f, r0 ; 63

26a: cd bf out 0x3d, r28 ; 61

26c: 9e 83 std Y+6, r25 ; 0x06

26e: 8d 83 std Y+5, r24 ; 0x05

uint32\_t i=millis/1000\*16000000/34;

270: 8d 81 ldd r24, Y+5 ; 0x05

272: 9e 81 ldd r25, Y+6 ; 0x06

274: 96 95 lsr r25

276: 87 95 ror r24

278: 96 95 lsr r25

27a: 87 95 ror r24

27c: 96 95 lsr r25

27e: 87 95 ror r24

280: 9c 01 movw r18, r24

282: a5 ec ldi r26, 0xC5 ; 197

284: b0 e2 ldi r27, 0x20 ; 32

286: 0e 94 cf 01 call 0x39e ; 0x39e <\_\_umulhisi3>

28a: 92 95 swap r25

28c: 82 95 swap r24

28e: 8f 70 andi r24, 0x0F ; 15

290: 89 27 eor r24, r25

292: 9f 70 andi r25, 0x0F ; 15

294: 89 27 eor r24, r25

296: cc 01 movw r24, r24

298: a0 e0 ldi r26, 0x00 ; 0

29a: b0 e0 ldi r27, 0x00 ; 0

29c: 9c 01 movw r18, r24

29e: ad 01 movw r20, r26

2a0: 60 e0 ldi r22, 0x00 ; 0

2a2: 74 e2 ldi r23, 0x24 ; 36

2a4: 84 ef ldi r24, 0xF4 ; 244

2a6: 90 e0 ldi r25, 0x00 ; 0

2a8: 0e 94 a0 01 call 0x340 ; 0x340 <\_\_mulsi3>

2ac: dc 01 movw r26, r24

2ae: cb 01 movw r24, r22

2b0: 22 e2 ldi r18, 0x22 ; 34

2b2: 30 e0 ldi r19, 0x00 ; 0

2b4: 40 e0 ldi r20, 0x00 ; 0

2b6: 50 e0 ldi r21, 0x00 ; 0

2b8: bc 01 movw r22, r24

2ba: cd 01 movw r24, r26

2bc: 0e 94 b0 01 call 0x360 ; 0x360 <\_\_divmodsi4>

2c0: da 01 movw r26, r20

2c2: c9 01 movw r24, r18

2c4: 89 83 std Y+1, r24 ; 0x01

2c6: 9a 83 std Y+2, r25 ; 0x02

2c8: ab 83 std Y+3, r26 ; 0x03

2ca: bc 83 std Y+4, r27 ; 0x04

for(; i>0; i--){ // question

2cc: 0c c0 rjmp .+24 ; 0x2e6 <wait+0x8e>

asm volatile("nop");

2ce: 00 00 nop

}\*/

void wait(uint16\_t millis)

{

uint32\_t i=millis/1000\*16000000/34;

for(; i>0; i--){ // question

2d0: 89 81 ldd r24, Y+1 ; 0x01

2d2: 9a 81 ldd r25, Y+2 ; 0x02

2d4: ab 81 ldd r26, Y+3 ; 0x03

2d6: bc 81 ldd r27, Y+4 ; 0x04

2d8: 01 97 sbiw r24, 0x01 ; 1

2da: a1 09 sbc r26, r1

2dc: b1 09 sbc r27, r1

2de: 89 83 std Y+1, r24 ; 0x01

2e0: 9a 83 std Y+2, r25 ; 0x02

2e2: ab 83 std Y+3, r26 ; 0x03

2e4: bc 83 std Y+4, r27 ; 0x04

2e6: 89 81 ldd r24, Y+1 ; 0x01

2e8: 9a 81 ldd r25, Y+2 ; 0x02

2ea: ab 81 ldd r26, Y+3 ; 0x03

2ec: bc 81 ldd r27, Y+4 ; 0x04

2ee: 89 2b or r24, r25

2f0: 8a 2b or r24, r26

2f2: 8b 2b or r24, r27

2f4: 61 f7 brne .-40 ; 0x2ce <wait+0x76>

asm volatile("nop");

}

}

2f6: 2a 96 adiw r28, 0x0a ; 10

2f8: 0f b6 in r0, 0x3f ; 63

2fa: f8 94 cli

2fc: de bf out 0x3e, r29 ; 62

2fe: 0f be out 0x3f, r0 ; 63

300: cd bf out 0x3d, r28 ; 61

302: df 91 pop r29

304: cf 91 pop r28

306: 08 95 ret

Cycles

the number of cycles inside the loop : 2+8+2+1+16+2+3+1=10+3+18+4=13+22=35

sbc 1\*2 =2

std 2 \*4 = 8

rjmp 2\*1 =2

nop 1 \*1 =1

ldd 1 or 2 or 3 \*8 = 16

sbiw 2 \*1 =2

or 1\*3 =3

brne 1 if condition is false 2 if condition is true = 1

why in the assembly file, there are two wait functions?

Millis/1000\*16000000 cycles needed.

The number of loop \* the number of cycles per loop = cycles needed.