```
        .section __TEXT,__text,regular,pure_instructions
        .macosx_version_min 10, 13
        .globl  _popcount_1_data        ## -- Begin function
popcount_1_data
        .p2align 4, 0x90
_popcount_1_data:                       ## @popcount_1_data
        .cfi_startproc
## BB#0:
        pushq   %rbp
Lcfi0:
        .cfi_def_cfa_offset 16
Lcfi1:
        .cfi_offset %rbp, -16
        movq    %rsp, %rbp
Lcfi2:
        .cfi_def_cfa_register %rbp
        pushq   %r14
        pushq   %rbx
Lcfi3:
        .cfi_offset %rbx, -32
Lcfi4:
        .cfi_offset %r14, -24
        movl    $64, %ecx
        xorl    %eax, %eax
        movl    $257, %r8d              ## imm = 0x101
        movl    $258, %r9d              ## imm = 0x102
        movl    $259, %r10d             ## imm = 0x103
        movl    $260, %r11d             ## imm = 0x104
        movl    $261, %r14d             ## imm = 0x105
        movl    $262, %esi              ## imm = 0x106
        movl    $263, %ebx              ## imm = 0x107
        .p2align 4, 0x90
LBB0_1:                                 ## =>This Inner Loop Header:
Depth=1
        movl    %edi, %edx
        andl    $1, %edx
        addl    %edx, %eax
        bextrl  %r8d, %edi, %edx
        addl    %eax, %edx
        bextrl  %r9d, %edi, %eax
        addl    %edx, %eax
        bextrl  %r10d, %edi, %edx
        addl    %eax, %edx
        bextrl  %r11d, %edi, %eax
        addl    %edx, %eax
        bextrl  %r14d, %edi, %edx
        addl    %eax, %edx
        bextrl  %esi, %edi, %eax
        addl    %edx, %eax
        bextrl  %ebx, %edi, %edx
```

```
        addq    %rdx, %rax
        shrq    $8, %rdi
        addl    $-8, %ecx
        jne     LBB0_1
## BB#2:                                ## kill: %EAX<def> %EAX<kill>
%RAX<kill>
        popq    %rbx
        popq    %r14
        popq    %rbp
        retq
        .cfi_endproc
                                        ## -- End function
        .section __TEXT,__const
        .p2align 5                      ## -- Begin function
popcount_1_control
LCPI1_0:
        .quad   6                       ## 0x6
        .quad   5                       ## 0x5
        .quad   4                       ## 0x4
        .quad   3                       ## 0x3
LCPI1_2:
        .quad   10                      ## 0xa
        .quad   9                       ## 0x9
        .quad   8                       ## 0x8
        .quad   7                       ## 0x7
LCPI1_3:
        .quad   14                      ## 0xe
        .quad   13                      ## 0xd
        .quad   12                      ## 0xc
        .quad   11                      ## 0xb
LCPI1_4:
        .quad   18                      ## 0x12
        .quad   17                      ## 0x11
        .quad   16                      ## 0x10
        .quad   15                      ## 0xf
LCPI1_5:
        .quad   22                      ## 0x16
        .quad   21                      ## 0x15
        .quad   20                      ## 0x14
        .quad   19                      ## 0x13
LCPI1_6:
        .quad   26                      ## 0x1a
        .quad   25                      ## 0x19
        .quad   24                      ## 0x18
        .quad   23                      ## 0x17
LCPI1_7:
        .quad   30                      ## 0x1e
        .quad   29                      ## 0x1d
        .quad   28                      ## 0x1c
```

```
        .quad   27                      ## 0x1b
LCPI1_8:
        .quad   42                      ## 0x2a
        .quad   41                      ## 0x29
        .quad   40                      ## 0x28
        .quad   39                      ## 0x27
LCPI1_9:
        .quad   46                      ## 0x2e
        .quad   45                      ## 0x2d
        .quad   44                      ## 0x2c
        .quad   43                      ## 0x2b
LCPI1_10:
        .quad   58                      ## 0x3a
        .quad   57                      ## 0x39
        .quad   56                      ## 0x38
        .quad   55                      ## 0x37
LCPI1_11:
        .quad   62                      ## 0x3e
        .quad   61                      ## 0x3d
        .quad   60                      ## 0x3c
        .quad   59                      ## 0x3b
LCPI1_12:
        .quad   34                      ## 0x22
        .quad   33                      ## 0x21
        .quad   32                      ## 0x20
        .quad   31                      ## 0x1f
LCPI1_13:
        .quad   38                      ## 0x26
        .quad   37                      ## 0x25
        .quad   36                      ## 0x24
        .quad   35                      ## 0x23
LCPI1_14:
        .quad   50                      ## 0x32
        .quad   49                      ## 0x31
        .quad   48                      ## 0x30
        .quad   47                      ## 0x2f
LCPI1_15:
        .quad   54                      ## 0x36
        .quad   53                      ## 0x35
        .quad   52                      ## 0x34
        .quad   51                      ## 0x33
        .section __TEXT,__literal4,4byte_literals
        .p2align 2
LCPI1_1:
        .long   1                       ## 0x1
        .section __TEXT,__text,regular,pure_instructions
        .globl  _popcount_1_control
        .p2align 4, 0x90
_popcount_1_control:                    ## @popcount_1_control
        .cfi_startproc
```

```
## BB#0:
        pushq   %rbp
Lcfi5:
        .cfi_def_cfa_offset 16
Lcfi6:
        .cfi_offset %rbp, -16
        movq    %rsp, %rbp
Lcfi7:
        .cfi_def_cfa_register %rbp
        movl    %edi, %ecx
        andl    $1, %ecx
        movl    $257, %eax              ## imm = 0x101
        bextrl  %eax, %edi, %r8d
        movl    $258, %eax              ## imm = 0x102
        bextrl  %eax, %edi, %r9d
        vmovq   %rdi, %xmm0
        vpbroadcastq    %xmm0, %ymm3
        vpsrlvq  LCPI1_0(%rip), %ymm3, %ymm0
        vpshufd  $232, %ymm0, %ymm0      ## ymm0 =
ymm0[0,2,2,3,4,6,6,7]
        vpermq  $232, %ymm0, %ymm0      ## ymm0 = ymm0[0,2,2,3]
        movl    LCPI1_1(%rip), %eax
        vmovd   %eax, %xmm1
        vpbroadcastd    %xmm1, %xmm2
        vpand   %xmm2, %xmm0, %xmm0
        vpsrlvq  LCPI1_2(%rip), %ymm3, %ymm2
        vpsrlvq  LCPI1_3(%rip), %ymm3, %ymm4
        vpshufd  $232, %ymm4, %ymm4      ## ymm4 =
ymm4[0,2,2,3,4,6,6,7]
        vpermq  $232, %ymm4, %ymm4      ## ymm4 = ymm4[0,2,2,3]
        vpshufd  $232, %ymm2, %ymm2      ## ymm2 =
ymm2[0,2,2,3,4,6,6,7]
        vpermq  $232, %ymm2, %ymm2      ## ymm2 = ymm2[0,2,2,3]
        vinserti128     $1, %xmm2, %ymm4, %ymm4
        vpbroadcastd    %xmm1, %ymm2
        vpand   %ymm2, %ymm4, %ymm1
        vpsrlvq  LCPI1_4(%rip), %ymm3, %ymm4
        vpsrlvq  LCPI1_5(%rip), %ymm3, %ymm5
        vpsrlvq  LCPI1_6(%rip), %ymm3, %ymm6
        vpsrlvq  LCPI1_7(%rip), %ymm3, %ymm7
        vpshufd  $232, %ymm7, %ymm7      ## ymm7 =
ymm7[0,2,2,3,4,6,6,7]
        vpermq  $232, %ymm7, %ymm7      ## ymm7 = ymm7[0,2,2,3]
        vpshufd  $232, %ymm6, %ymm6      ## ymm6 =
ymm6[0,2,2,3,4,6,6,7]
        vpermq  $232, %ymm6, %ymm6      ## ymm6 = ymm6[0,2,2,3]
        vinserti128     $1, %xmm6, %ymm7, %ymm6
        vpshufd  $232, %ymm5, %ymm5      ## ymm5 =
ymm5[0,2,2,3,4,6,6,7]
        vpermq  $232, %ymm5, %ymm5      ## ymm5 = ymm5[0,2,2,3]
```

```
        vpshufd  $232, %ymm4, %ymm4      ## ymm4 =
ymm4[0,2,2,3,4,6,6,7]
        vpermq   $232, %ymm4, %ymm4      ## ymm4 = ymm4[0,2,2,3]
        vinserti128      $1, %xmm4, %ymm5, %ymm4
        vpand    %ymm2, %ymm4, %ymm4
        vpand    %ymm2, %ymm6, %ymm5
        vpsrlvq  LCPI1_8(%rip), %ymm3, %ymm6
        vpsrlvq  LCPI1_9(%rip), %ymm3, %ymm7
        vpsrlvq  LCPI1_10(%rip), %ymm3, %ymm8
        vpsrlvq  LCPI1_11(%rip), %ymm3, %ymm9
        vpsrlvq  LCPI1_14(%rip), %ymm3, %ymm10
        vpsrlvq  LCPI1_15(%rip), %ymm3, %ymm11
        vpshufd  $232, %ymm11, %ymm11    ## ymm11 =
ymm11[0,2,2,3,4,6,6,7]
        vpermq   $232, %ymm11, %ymm11    ## ymm11 = ymm11[0,2,2,3]
        vpshufd  $232, %ymm10, %ymm10    ## ymm10 =
ymm10[0,2,2,3,4,6,6,7]
        vpermq   $232, %ymm10, %ymm10    ## ymm10 = ymm10[0,2,2,3]
        vinserti128      $1, %xmm10, %ymm11, %ymm10
        vpsrlvq  LCPI1_12(%rip), %ymm3, %ymm11
        vpsrlvq  LCPI1_13(%rip), %ymm3, %ymm3
        vpshufd  $232, %ymm3, %ymm3      ## ymm3 =
ymm3[0,2,2,3,4,6,6,7]
        vpermq   $232, %ymm3, %ymm3      ## ymm3 = ymm3[0,2,2,3]
        vpshufd  $232, %ymm11, %ymm11    ## ymm11 =
ymm11[0,2,2,3,4,6,6,7]
        vpermq   $232, %ymm11, %ymm11    ## ymm11 = ymm11[0,2,2,3]
        vinserti128      $1, %xmm11, %ymm3, %ymm3
        vpshufd  $232, %ymm9, %ymm9      ## ymm9 =
ymm9[0,2,2,3,4,6,6,7]
        vpermq   $232, %ymm9, %ymm9      ## ymm9 = ymm9[0,2,2,3]
        vpshufd  $232, %ymm8, %ymm8      ## ymm8 =
ymm8[0,2,2,3,4,6,6,7]
        vpermq   $232, %ymm8, %ymm8      ## ymm8 = ymm8[0,2,2,3]
        vinserti128      $1, %xmm8, %ymm9, %ymm8
        vpshufd  $232, %ymm7, %ymm7      ## ymm7 =
ymm7[0,2,2,3,4,6,6,7]
        vpermq   $232, %ymm7, %ymm7      ## ymm7 = ymm7[0,2,2,3]
        vpshufd  $232, %ymm6, %ymm6      ## ymm6 =
ymm6[0,2,2,3,4,6,6,7]
        vpermq   $232, %ymm6, %ymm6      ## ymm6 = ymm6[0,2,2,3]
        vinserti128      $1, %xmm6, %ymm7, %ymm6
        vpand    %ymm2, %ymm6, %ymm6
        vpand    %ymm2, %ymm8, %ymm7
        vpaddd   %ymm6, %ymm7, %ymm6
        vpand    %ymm2, %ymm3, %ymm3
        vpand    %ymm2, %ymm10, %ymm2
        vpaddd   %ymm3, %ymm2, %ymm2
        vpaddd   %ymm2, %ymm6, %ymm2
        vextracti128     $1, %ymm2, %xmm3
```

```
        vpaddd  %ymm3, %ymm2, %ymm2
        vpshufd $78, %xmm2, %xmm3       ## xmm3 = xmm2[2,3,0,1]
        vpaddd  %ymm3, %ymm2, %ymm2
        vphaddd %ymm2, %ymm2, %ymm2
        vpaddd  %ymm4, %ymm5, %ymm3
        vextracti128    $1, %ymm3, %xmm4
        vpaddd  %ymm4, %ymm3, %ymm3
        vpshufd $78, %xmm3, %xmm4       ## xmm4 = xmm3[2,3,0,1]
        vpaddd  %ymm4, %ymm3, %ymm3
        vphaddd %ymm3, %ymm3, %ymm3
        vmovd   %xmm2, %eax
        vextracti128    $1, %ymm1, %xmm2
        vpaddd  %ymm2, %ymm1, %ymm1
        vpshufd $78, %xmm1, %xmm2       ## xmm2 = xmm1[2,3,0,1]
        vpaddd  %ymm2, %ymm1, %ymm1
        vphaddd %ymm1, %ymm1, %ymm1
        vmovd   %xmm3, %edx
        addl    %eax, %edx
        vmovd   %xmm1, %esi
        vpshufd $78, %xmm0, %xmm1       ## xmm1 = xmm0[2,3,0,1]
        vpaddd  %xmm1, %xmm0, %xmm0
        vphaddd %xmm0, %xmm0, %xmm0
        addl    %edx, %esi
        vmovd   %xmm0, %eax
        addl    %esi, %eax
        addl    %r9d, %eax
        addl    %r8d, %eax
        addl    %ecx, %eax
        shrq    $63, %rdi
        addl    %edi, %eax
        popq    %rbp
        vzeroupper
        retq
        .cfi_endproc
                                        ## -- End function
        .globl  _popcount_4_data        ## -- Begin function
popcount_4_data
        .p2align 4, 0x90
_popcount_4_data:                       ## @popcount_4_data
        .cfi_startproc
## BB#0:
        pushq   %rbp
Lcfi8:
        .cfi_def_cfa_offset 16
Lcfi9:
        .cfi_offset %rbp, -16
        movq    %rsp, %rbp
Lcfi10:
        .cfi_def_cfa_register %rbp
        movl    %edi, %edx
```

```
andl    $15, %edx
leaq    _pop4(%rip), %rcx
movq    %rdi, %rax
shrq    $2, %rax
andl    $60, %eax
movl    (%rax,%rcx), %eax
addl    (%rcx,%rdx,4), %eax
movq    %rdi, %rdx
shrq    $6, %rdx
andl    $60, %edx
addl    (%rdx,%rcx), %eax
movq    %rdi, %rdx
shrq    $10, %rdx
andl    $60, %edx
addl    (%rdx,%rcx), %eax
movq    %rdi, %rdx
shrq    $14, %rdx
andl    $60, %edx
addl    (%rdx,%rcx), %eax
movq    %rdi, %rdx
shrq    $18, %rdx
andl    $60, %edx
addl    (%rdx,%rcx), %eax
movq    %rdi, %rdx
shrq    $22, %rdx
andl    $60, %edx
addl    (%rdx,%rcx), %eax
movq    %rdi, %rdx
shrq    $26, %rdx
andl    $60, %edx
addl    (%rdx,%rcx), %eax
movq    %rdi, %rdx
shrq    $30, %rdx
andl    $60, %edx
addl    (%rdx,%rcx), %eax
movq    %rdi, %rdx
shrq    $34, %rdx
andl    $60, %edx
addl    (%rdx,%rcx), %eax
movq    %rdi, %rdx
shrq    $38, %rdx
andl    $60, %edx
addl    (%rdx,%rcx), %eax
movq    %rdi, %rdx
shrq    $42, %rdx
andl    $60, %edx
addl    (%rdx,%rcx), %eax
movq    %rdi, %rdx
shrq    $46, %rdx
andl    $60, %edx
```

```
        addl    (%rdx,%rcx), %eax
        movq    %rdi, %rdx
        shrq    $50, %rdx
        andl    $60, %edx
        addl    (%rdx,%rcx), %eax
        movq    %rdi, %rdx
        shrq    $54, %rdx
        andl    $60, %edx
        addl    (%rdx,%rcx), %eax
        shrq    $60, %rdi
        addl    (%rcx,%rdi,4), %eax
        popq    %rbp
        retq
        .cfi_endproc
                                        ## -- End function
        .globl  _popcount_4_control     ## -- Begin function
popcount_4_control
        .p2align 4, 0x90
_popcount_4_control:                    ## @popcount_4_control
        .cfi_startproc
## BB#0:
        pushq   %rbp
Lcfi11:
        .cfi_def_cfa_offset 16
Lcfi12:
        .cfi_offset %rbp, -16
        movq    %rsp, %rbp
Lcfi13:
        .cfi_def_cfa_register %rbp
        xorl    %eax, %eax
        movl    $-1, %ecx
        leaq    LJTI3_0(%rip), %rsi
        .p2align 4, 0x90
LBB3_1:                                 ## =>This Inner Loop Header:
Depth=1
        movl    %edi, %edx
        andb    $15, %dl
        addb    $-1, %dl
        cmpb    $14, %dl
        ja      LBB3_14
## BB#2:                                ##   in Loop: Header=BB3_1
Depth=1
        movzbl  %dl, %edx
        movslq  (%rsi,%rdx,4), %rdx
        addq    %rsi, %rdx
        jmpq    *%rdx
LBB3_7:                                 ##   in Loop: Header=BB3_1
Depth=1
        addl    $2, %eax
        jmp     LBB3_14
```

```
        .p2align 4, 0x90
LBB3_3:                                 ##   in Loop: Header=BB3_1
Depth=1
        addl    $1, %eax
        jmp     LBB3_14
        .p2align 4, 0x90
LBB3_13:                                ##   in Loop: Header=BB3_1
Depth=1
        addl    $3, %eax
        jmp     LBB3_14
LBB3_9:                                 ##   in Loop: Header=BB3_1
Depth=1
        addl    $4, %eax
        .p2align 4, 0x90
LBB3_14:                                ##   in Loop: Header=BB3_1
Depth=1
        shrq    $4, %rdi
        addl    $1, %ecx
        cmpl    $15, %ecx
        jb      LBB3_1
## BB#15:
        popq    %rbp
        retq
        .cfi_endproc
        .p2align 2, 0x90
        .data_region jt32
L3_0_set_3 = LBB3_3-LJTI3_0
L3_0_set_7 = LBB3_7-LJTI3_0
L3_0_set_13 = LBB3_13-LJTI3_0
L3_0_set_9 = LBB3_9-LJTI3_0
LJTI3_0:
        .long   L3_0_set_3
        .long   L3_0_set_3
        .long   L3_0_set_7
        .long   L3_0_set_3
        .long   L3_0_set_7
        .long   L3_0_set_7
        .long   L3_0_set_13
        .long   L3_0_set_3
        .long   L3_0_set_7
        .long   L3_0_set_7
        .long   L3_0_set_13
        .long   L3_0_set_7
        .long   L3_0_set_13
        .long   L3_0_set_13
        .long   L3_0_set_9
        .end_data_region
                                        ## -- End function
        .globl  _popcount_8_data        ## -- Begin function
popcount_8_data
```

```
        .p2align 4, 0x90
_popcount_8_data:                       ## @popcount_8_data
        .cfi_startproc
## BB#0:
        pushq   %rbp
Lcfi14:
        .cfi_def_cfa_offset 16
Lcfi15:
        .cfi_offset %rbp, -16
        movq    %rsp, %rbp
Lcfi16:
        .cfi_def_cfa_register %rbp
        movq    %rdi, %rcx
        movzbl  %cl, %esi
        leaq    _pop8(%rip), %rdx
        movzbl  %ch, %eax  # NOREX
        movl    (%rdx,%rax,4), %eax
        addl    (%rdx,%rsi,4), %eax
        movq    %rcx, %rsi
        shrq    $14, %rsi
        andl    $1020, %esi             ## imm = 0x3FC
        addl    (%rsi,%rdx), %eax
        movq    %rcx, %rsi
        shrq    $22, %rsi
        andl    $1020, %esi             ## imm = 0x3FC
        addl    (%rsi,%rdx), %eax
        movq    %rcx, %rsi
        shrq    $30, %rsi
        andl    $1020, %esi             ## imm = 0x3FC
        addl    (%rsi,%rdx), %eax
        movq    %rcx, %rsi
        shrq    $38, %rsi
        andl    $1020, %esi             ## imm = 0x3FC
        addl    (%rsi,%rdx), %eax
        movq    %rcx, %rsi
        shrq    $46, %rsi
        andl    $1020, %esi             ## imm = 0x3FC
        addl    (%rsi,%rdx), %eax
        shrq    $56, %rcx
        addl    (%rdx,%rcx,4), %eax
        popq    %rbp
        retq
        .cfi_endproc
                                        ## -- End function
        .globl  _popcount_16_data       ## -- Begin function
popcount_16_data
        .p2align 4, 0x90
_popcount_16_data:                      ## @popcount_16_data
        .cfi_startproc
## BB#0:
```

```
        pushq   %rbp
Lcfi17:
        .cfi_def_cfa_offset 16
Lcfi18:
        .cfi_offset %rbp, -16
        movq    %rsp, %rbp
Lcfi19:
        .cfi_def_cfa_register %rbp
        movzwl  %di, %ecx
        leaq    _pop16(%rip), %rdx
        movq    %rdi, %rax
        shrq    $14, %rax
        andl    $262140, %eax           ## imm = 0x3FFFC
        movl    (%rax,%rdx), %eax
        addl    (%rdx,%rcx,4), %eax
        movq    %rdi, %rcx
        shrq    $30, %rcx
        andl    $262140, %ecx           ## imm = 0x3FFFC
        addl    (%rcx,%rdx), %eax
        shrq    $48, %rdi
        addl    (%rdx,%rdi,4), %eax
        popq    %rbp
        retq
        .cfi_endproc
                                        ## -- End function
        .globl  _popcount_kernighan     ## -- Begin function
popcount_kernighan
        .p2align 4, 0x90
_popcount_kernighan:                    ## @popcount_kernighan
        .cfi_startproc
## BB#0:
        pushq   %rbp
Lcfi20:
        .cfi_def_cfa_offset 16
Lcfi21:
        .cfi_offset %rbp, -16
        movq    %rsp, %rbp
Lcfi22:
        .cfi_def_cfa_register %rbp
        xorl    %eax, %eax
        testq   %rdi, %rdi
        je      LBB6_2
        .p2align 4, 0x90
LBB6_1:                                 ## =>This Inner Loop Header:
Depth=1
        movl    %edi, %ecx
        andl    $1, %ecx
        movl    %eax, %eax
        addq    %rcx, %rax
        shrq    %rdi
```

```
        jne     LBB6_1
LBB6_2:                                 ## kill: %EAX<def> %EAX<kill>
%RAX<kill>
        popq    %rbp
        retq
        .cfi_endproc
                                        ## -- End function
        .globl  _popcount64a            ## -- Begin function
popcount64a
        .p2align 4, 0x90
_popcount64a:                           ## @popcount64a
        .cfi_startproc
## BB#0:
        pushq   %rbp
Lcfi23:
        .cfi_def_cfa_offset 16
Lcfi24:
        .cfi_offset %rbp, -16
        movq    %rsp, %rbp
Lcfi25:
        .cfi_def_cfa_register %rbp
        movabsq  $6148914691236517205, %rax ## imm =
0x5555555555555555
        movq    %rdi, %rcx
        andq    %rax, %rcx
        shrq    %rdi
        andq    %rax, %rdi
        addq    %rcx, %rdi
        movabsq  $3689348814741910323, %rax ## imm =
0x3333333333333333
        movq    %rdi, %rcx
        andq    %rax, %rcx
        shrq    $2, %rdi
        andq    %rax, %rdi
        addq    %rcx, %rdi
        movabsq  $506381209866536711, %rax ## imm = 0x707070707070707
        movq    %rdi, %rcx
        andq    %rax, %rcx
        shrq    $4, %rdi
        andq    %rax, %rdi
        addq    %rcx, %rdi
        movabsq  $4222189076152335, %rax ## imm = 0xF000F000F000F
        movq    %rdi, %rcx
        andq    %rax, %rcx
        shrq    $8, %rdi
        andq    %rax, %rdi
        addq    %rcx, %rdi
        movabsq  $133143986207, %rax     ## imm = 0x1F0000001F
        movq    %rdi, %rcx
```

```
        andq    %rax, %rcx
        shrq    $16, %rdi
        andq    %rax, %rdi
        addq    %rcx, %rdi
        movq    %rdi, %rax
        shrq    $32, %rax
        addl    %eax, %edi
        movl    %edi, %eax
        popq    %rbp
        retq
        .cfi_endproc
                                        ## -- End function
        .globl  _popcount64b            ## -- Begin function
popcount64b
        .p2align 4, 0x90
_popcount64b:                           ## @popcount64b
        .cfi_startproc
## BB#0:
        pushq   %rbp
Lcfi26:
        .cfi_def_cfa_offset 16
Lcfi27:
        .cfi_offset %rbp, -16
        movq    %rsp, %rbp
Lcfi28:
        .cfi_def_cfa_register %rbp
        movq    %rdi, %rax
        shrq    %rax
        movabsq  $6148914691236517205, %rcx ## imm =
0x5555555555555555
        andq    %rax, %rcx
        subq    %rcx, %rdi
        movabsq  $3689348814741910323, %rax ## imm =
0x3333333333333333
        movq    %rdi, %rcx
        andq    %rax, %rcx
        shrq    $2, %rdi
        andq    %rax, %rdi
        addq    %rcx, %rdi
        movq    %rdi, %rax
        shrq    $4, %rax
        addq    %rdi, %rax
        movabsq  $1085102592571150095, %rcx ## imm = 0xF0F0F0F0F0F0F0F
        andq    %rax, %rcx
        movq    %rcx, %rax
        shrq    $8, %rax
        addq    %rcx, %rax
        movq    %rax, %rcx
        shrq    $16, %rcx
        addq    %rax, %rcx
```

```
        movq    %rcx, %rax
        shrq    $32, %rax
        addl    %ecx, %eax
        andl    $127, %eax
                                        ## kill: %EAX<def> %EAX<kill>
%RAX<kill>
        popq    %rbp
        retq
        .cfi_endproc
                                        ## -- End function
        .globl  _popcount64c            ## -- Begin function
popcount64c
        .p2align 4, 0x90
_popcount64c:                           ## @popcount64c
        .cfi_startproc
## BB#0:
        pushq   %rbp
Lcfi29:
        .cfi_def_cfa_offset 16
Lcfi30:
        .cfi_offset %rbp, -16
        movq    %rsp, %rbp
Lcfi31:
        .cfi_def_cfa_register %rbp
        movq    %rdi, %rax
        shrq    %rax
        movabsq $6148914691236517205, %rcx ## imm =
0x5555555555555555
        andq    %rax, %rcx
        subq    %rcx, %rdi
        movabsq $3689348814741910323, %rax ## imm =
0x3333333333333333
        movq    %rdi, %rcx
        andq    %rax, %rcx
        shrq    $2, %rdi
        andq    %rax, %rdi
        addq    %rcx, %rdi
        movq    %rdi, %rax
        shrq    $4, %rax
        leaq    (%rax,%rdi), %rax
        movabsq $1085102592571150095, %rcx ## imm = 0xF0F0F0F0F0F0F0F
        andq    %rax, %rcx
        movabsq $72340172838076673, %rax ## imm = 0x101010101010101
        imulq   %rcx, %rax
        shrq    $56, %rax
                                        ## kill: %EAX<def> %EAX<kill>
%RAX<kill>
        popq    %rbp
        retq
        .cfi_endproc
```

```
                                        ## -- End function
        .globl  _popcount64_fast        ## -- Begin function
popcount64_fast
        .p2align 4, 0x90
_popcount64_fast:                       ## @popcount64_fast
        .cfi_startproc
## BB#0:
        pushq   %rbp
Lcfi32:
        .cfi_def_cfa_offset 16
Lcfi33:
        .cfi_offset %rbp, -16
        movq    %rsp, %rbp
Lcfi34:
        .cfi_def_cfa_register %rbp
        popcntq %rdi, %rax
                                        ## kill: %EAX<def> %EAX<kill>
%RAX<kill>
        popq    %rbp
        retq
        .cfi_endproc
                                        ## -- End function
        .section __TEXT,__literal8,8byte_literals
        .p2align 3               ## -- Begin function main
LCPI11_0:
        .quad   4741671816366391296     ## double 1.0E+9
LCPI11_1:
        .quad   4666723172467343360     ## double 1.0E+4
        .section __TEXT,__text,regular,pure_instructions
        .globl  _main
        .p2align 4, 0x90
_main:                                  ## @main
        .cfi_startproc
## BB#0:
        pushq   %rbp
Lcfi35:
        .cfi_def_cfa_offset 16
Lcfi36:
        .cfi_offset %rbp, -16
        movq    %rsp, %rbp
Lcfi37:
        .cfi_def_cfa_register %rbp
        pushq   %r15
        pushq   %r14
        pushq   %r13
        pushq   %r12
        pushq   %rbx
        subq    $80040, %rsp            ## imm = 0x138A8
Lcfi38:
        .cfi_offset %rbx, -56
```

```
Lcfi39:
        .cfi_offset %r12, -48
Lcfi40:
        .cfi_offset %r13, -40
Lcfi41:
        .cfi_offset %r14, -32
Lcfi42:
        .cfi_offset %r15, -24
        movq    ___stack_chk_guard@GOTPCREL(%rip), %rax
        movq    (%rax), %rax
        movq    %rax, -48(%rbp)
        xorl    %eax, %eax
        leaq    _pop8(%rip), %rcx
        leaq    _pop16(%rip), %rdx
        jmp     LBB11_1
        .p2align 4, 0x90
LBB11_2:                                ##   in Loop: Header=BB11_1
Depth=1
        movl    %esi, (%rcx,%rax,4)
        movl    %esi, (%rdx,%rax,4)
        addq    $1, %rax
LBB11_1:                                ## =>This Inner Loop Header:
Depth=1
        popcntl %eax, %esi
        cmpq    $255, %rax
        jbe     LBB11_2
## BB#3:                                ##   in Loop: Header=BB11_1
Depth=1
        movl    %esi, (%rdx,%rax,4)
        addq    $1, %rax
        cmpq    $65536, %rax            ## imm = 0x10000
        jne     LBB11_1
## BB#4:
        movq    $-1, %rax
        vmovq   %rax, %xmm0
        vpslldq $8, %xmm0, %xmm0        ## xmm0 =
zero,zero,zero,zero,zero,zero,zero,zero,xmm0[0,1,2,3,4,5,6,7]
        vmovdqa %xmm0, -80048(%rbp)
        movl    $1, %r12d
        leaq    L_.str.3(%rip), %r14
        leaq    -80064(%rbp), %r15
        .p2align 4, 0x90
LBB11_5:                                ## =>This Inner Loop Header:
Depth=1
        xorl    %esi, %esi
        xorl    %eax, %eax
        movq    %r14, %rdi
        callq   _open
        movl    %eax, %ebx
        testl   %ebx, %ebx
```

```
        js      LBB11_17
## BB#6:                                ##   in Loop: Header=BB11_5
Depth=1
        movl    $8, %edx
        movl    %ebx, %edi
        movq    %r15, %rsi
        callq   _read
        cmpq    $8, %rax
        jne     LBB11_17
## BB#7:                                ##   in Loop: Header=BB11_5
Depth=1
        movl    %ebx, %edi
        callq   _close
        testl   %eax, %eax
        jne     LBB11_17
## BB#8:                                ##   in Loop: Header=BB11_5
Depth=1
        movq    -80064(%rbp), %rax
        movq    %rax, -80040(%rbp,%r12,8)
        addq    $1, %r12
        cmpq    $9999, %r12             ## imm = 0x270F
        jb      LBB11_5
## BB#9:
        leaq    _popcounts(%rip), %r15
        xorl    %ebx, %ebx
                                        ## implicit-def: %R12D
        .p2align 4, 0x90
LBB11_10:                               ## =>This Loop Header: Depth=1
                                        ##     Child Loop BB11_11
Depth 2
        movl    %r12d, %r14d
        movl    $6, %edi
        leaq    -80064(%rbp), %rsi
        callq   _clock_gettime
        xorl    %r13d, %r13d
        xorl    %r12d, %r12d
        .p2align 4, 0x90
LBB11_11:                               ##   Parent Loop BB11_10
Depth=1
                                        ## =>  This Inner Loop Header:
Depth=2
        movq    -80048(%rbp,%r13,8), %rdi
        callq   *(%r15)
        addl    %eax, %r12d
        addq    $1, %r13
        cmpq    $10000, %r13            ## imm = 0x2710
        jne     LBB11_11
## BB#12:                               ##   in Loop: Header=BB11_10
Depth=1
        movl    $6, %edi
```

```
        leaq    -80080(%rbp), %rsi
        callq   _clock_gettime
        vcvtsi2sdq      -80080(%rbp), %xmm3, %xmm0
        vmovsd  LCPI11_0(%rip), %xmm1   ## xmm1 = mem[0],zero
        vmovapd %xmm1, %xmm2
        vmulsd  %xmm2, %xmm0, %xmm0
        vcvtsi2sdq      -80072(%rbp), %xmm3, %xmm1
        vaddsd  %xmm1, %xmm0, %xmm0
        vcvtsi2sdq      -80064(%rbp), %xmm3, %xmm1
        vmulsd  %xmm2, %xmm1, %xmm1
        vcvtsi2sdq      -80056(%rbp), %xmm3, %xmm2
        vaddsd  %xmm2, %xmm1, %xmm1
        vsubsd  %xmm1, %xmm0, %xmm0
        vdivsd  LCPI11_1(%rip), %xmm0, %xmm0
        movb    $1, %al
        leaq    L_.str(%rip), %rdi
        callq   _printf
        testq   %rbx, %rbx
        je      LBB11_14
## BB#13:                               ##   in Loop: Header=BB11_10
Depth=1
        cmpl    %r14d, %r12d
        jne     LBB11_18
LBB11_14:                               ##   in Loop: Header=BB11_10
Depth=1
        leaq    _popcounts(%rip), %rax
        leaq    (%rax,%rbx,8), %r15
        addq    $8, %r15
        addq    $1, %rbx
        cmpq    $11, %rbx
        jne     LBB11_10
## BB#15:
        movq    ___stack_chk_guard@GOTPCREL(%rip), %rax
        movq    (%rax), %rax
        cmpq    -48(%rbp), %rax
        jne     LBB11_19
## BB#16:
        xorl    %eax, %eax
        addq    $80040, %rsp            ## imm = 0x138A8
        popq    %rbx
        popq    %r12
        popq    %r13
        popq    %r14
        popq    %r15
        popq    %rbp
        retq
LBB11_19:
        callq   ___stack_chk_fail
LBB11_17:
        callq   _abort
```

```
LBB11_18:
        leaq    L___func__.test_popcounts(%rip), %rdi
        leaq    L_.str.1(%rip), %rsi
        leaq    L_.str.2(%rip), %rcx
        movl    $236, %edx
        callq   ___assert_rtn
        .cfi_endproc
                                        ## -- End function
        .section __TEXT,__const
        .p2align 4               ## @pop4
_pop4:
        .long   0                        ## 0x0
        .long   1                        ## 0x1
        .long   1                        ## 0x1
        .long   2                        ## 0x2
        .long   1                        ## 0x1
        .long   2                        ## 0x2
        .long   2                        ## 0x2
        .long   3                        ## 0x3
        .long   1                        ## 0x1
        .long   2                        ## 0x2
        .long   2                        ## 0x2
        .long   3                        ## 0x3
        .long   2                        ## 0x2
        .long   3                        ## 0x3
        .long   3                        ## 0x3
        .long   4                        ## 0x4

.zerofill __DATA,__bss,_pop8,1024,4     ## @pop8
.zerofill __DATA,__bss,_pop16,262144,4  ## @pop16
        .section __DATA,__const
        .p2align 4               ## @popcounts
_popcounts:
        .quad   _popcount_1_data
        .quad   _popcount_1_control
        .quad   _popcount_4_data
        .quad   _popcount_4_control
        .quad   _popcount_8_data
        .quad   _popcount_16_data
        .quad   _popcount_kernighan
        .quad   _popcount64a
        .quad   _popcount64b
        .quad   _popcount64c
        .quad   _popcount64_fast
        .quad   0

        .section __TEXT,__cstring,cstring_literals
L_.str:                                 ## @.str
        .asciz  "%lf ns / op\n"
```

```
L___func__.test_popcounts:              ## @__func__.test_popcounts
        .asciz  "test_popcounts"

L_.str.1:                               ## @.str.1
        .asciz  "popcount.c"

L_.str.2:                               ## @.str.2
        .asciz  "sum == last_sum"

L_.str.3:                               ## @.str.3
        .asciz  "/dev/urandom"


.subsections_via_symbols
```