Assembly: operations; load/store cont.

COSC 208, Introduction to Computer Systems, 2021-10-08

Announcements

- Exam1 Q5

Warm-up

- Q1: `lsl w9, w9, w10`

- Q2: `and w9, w9, w10`

- Q3: `mul w9, w9, w10`

- Q4: `sdiv w9, w9, w10`

Practice

The following C program (`operands.c`) has been compiled into assembly:

```c
int operandsA(int a) {
    return a;
}
long operandsB(long b) {
    return b;
}
int operandsC(int *c) {
    return *c;
}
long operandsD(long *d) {
    return *d;
}
int main() {
    operandsA(5);
    operandsB(5);
    int x = 5;
    operandsC(&x);
    long y = 5;
    operandsD(&y);
}
```

Q5: *Write the C code equivalent for each line of assembly, treating registers as if they were variable names. The assembly code for the `operandsA` function has already been translated into C code.*

```
00000000000007ec <operandsA>:
    7ec:    d10043ff    sub sp, sp, #0x10   // sp = sp - 0x10
    7f0:    b9000fe0    str w0, [sp, #12]   // *(sp + 12) = w0
    7f4:    b9400fe0    ldr w0, [sp, #12]   // w0 = *(sp + 12)
    7f8:    910043ff    add sp, sp, #0x10   // sp = sp + 0x10
    7fc:    d65f03c0    ret                 // return

0000000000000800 <operandsB>:
    800:    d10043ff    sub sp, sp, #0x10   //
    804:    f90007e0    str x0, [sp, #8]    //
    808:    f94007e0    ldr x0, [sp, #8]    //
    80c:    910043ff    add sp, sp, #0x10   //
    810:    d65f03c0    ret                 //

0000000000000814 <operandsC>:
    814:    d10043ff    sub sp, sp, #0x10   //
    818:    f90007e0    str x0, [sp, #8]    //
    81c:    f94007e0    ldr x0, [sp, #8]    //
    820:    b9400000    ldr w0, [x0]        //
    824:    910043ff    add sp, sp, #0x10   //
    828:    d65f03c0    ret                 //

000000000000082c <operandsD>:
    82c:    d10043ff    sub sp, sp, #0x10   //
    830:    f90007e0    str x0, [sp, #8]    //
    834:    f94007e0    ldr x0, [sp, #8]    //
    838:    f9400000    ldr x0, [x0]        //
    83c:    910043ff    add sp, sp, #0x10   //
    840:    d65f03c0    ret                 //
```

Q6: *How does the assembly code for* `operandsA` *and* `operandsB` *differ? Why?*

Q7: *How does the assembly code for* `operandsB` *and* `operandsD` *differ? Why?*

Q8: *How does the assembly code for* `operandsC` *and* `operandsD` *differ? Why?*