

# Example of SSA form optimization

## Conditional constant propagation and aggressive optimization

---

### Source program in C

```
/* appel_19_4.c */
/* Appel, A.: Modern compiler implementation in Java, 2nd ed., 2002
   Fig. 19.4 */

#include <stdio.h>

int main ()
{
    int i;
    int j;
    int k;

    i = 1;
    j = 1;
    k = 0;

    while (k < 100) {
        if (j < 20) {
            j = i;
            k = k + 1;
        } else {
            j = k;
            k = k + 2;
        }
    }

    printf("%d\n",j);
}
```

---

### Optimized code after conditional constant propagation

Actually an assembly language program is generated, but for the reader's convenience, we show an equivalent C program generated using our "lir2c" option.

```
// C file generated by LIR-to-C after conditional constant propagation
/* Module Name = "appel_19_4.c" */

static char string_6[4] = { 37, 100, 10, 0};

void main(void) {
    int i_1_;
    int j_2_;
```

```

    int k_3_;
    int returnvalue_4_;
    int functionvalue_5_;
    int i_1__0;
    int i_1__1;
    int j_2__0;
    int j_2__1;
    int k_3__0;
    int k_3__1;
    int k_3__2;
    int j_2__2;
    int functionvalue_5__0;
    int functionvalue_5__1;
    int k_3__3;
    int k_3__4;
    int _ssaI32;
    int _ssaI32_0;

_L1:
goto _L2;

_L2:
_ssaI32_0 = ((int)( 0));
goto _L3;

_L3:
if ((_ssaI32_0 < 100)) { goto _L5;} else { goto _L8;}

_L5:
_ssaI32_0 = ((int)((((int)(_ssaI32_0 + 1))));
goto _L3;

_L8:
functionvalue_5__1 = printf((unsigned char *)&(string_6), 1);
goto _L9;

_L9:
return;
}

```

This is equivalent to the following C program in usual style.

```

void main(void) {
    int k;
    k = 0;
    while (k < 100) {
        k = k + 1;
    }
    printf("%d\n", 1);
}

```