Rules for energy efficiency in C/C++

Pietro Ghiglio

April 1, 2020

Contents

1 Introduction

These are some rules that I've found in the papers that I have read in the previous months. They mainly refer to the papers [1] and [2]. For each rule I will provide its informal description and eventually a note when needed.

2 Rules

1. **Boolean return**
   Instead than returning the conjunction/disjunction of boolean variables, convert the multiple booleans into an unsigned int with associated flags and return its comparison with 0 or 1.
   Example:

```
bool or4(bool a, bool b, bool c, bool d){
        return a || b || c || d;
}

#define flagA (1u << 0)
#define flagB (1u << 1)
#define flagC (1u << 2)
#define flagD (1u << 3)

bool or4opt(unsigned int mask){
        return (mask & (flagA | flagB | flagC | flagD)) != 0;
}

//usages
or4(1,0,0,0);
or4opt(1000);
```

2. **Passing objects by reference**
   Passing an object or a struct by reference avoids the overhead of creating a complete copy of the argument.
   Example:

```
typedef struct {
        int a;
        int b;
```

```
} my_s;

int f(my_s value){
        return s.a+s.b;
}

int f_opt(my_s* ref){
        return s->a + s->b;
}
```

References

[1] Javier Corral García, Felipe Lemus Prieto, and Miguel Pérez-Toledano. "Analysis of Energy Consumption and Optimization Techniques for Writing Energy-Efficient Code". In: *Electronics* 8 (Oct. 2019), p. 1192. DOI: 10.3390/electronics8101192.

[2] Maurizio Morisio et al. "Definition, implementation and validation of energy code smells: an exploratory study on an embedded system". In: Mar. 2013, pp. 34–39.