Optimask Manual

by

Optixera

Optixera
USA and China

Silicon Valley, California, USA, 2017

2017/05/15

Abstract

Optimask is a revolutionary layout tool for advanced photomask drawing. It is developed after decades of professional experience from elite engineers and researchers in semiconductor and photonics industry companies, as well as scientific institutes and universities. Our team of international developers consist of photonic experts, electronics researchers and computer scientists. Our primary and ultimate goal is to simplify photonics simulation and photomask design as easy as word processing, with increased accuracy, precision, versatility, and portability. We will make photomask drawing a pleasure, and remove the barrier of advanced knowledge, skill-set and training required.

***Index Terms*:** Graphic Data System II (GDSII)

Acknowledgements

We would like to express our sincere gratitude to all people who helped with this product.

Optixera
10/22/2007–May 15, 2017
Revised: May 15, 2017

Contents

List of Tables

List of Figures

List of Acronyms

GDSII Graphic Data System II

Chapter 1

代码规范

良好的软件开发习惯和代码规范可以：便于团体协作开发，有效提高软件开发效率，增强软件代码质量，增强软件运行速度，提高软件的通用性可移植性，降低代码出错机率，去除冗余代码，减少软件维护时间和成本，等等。所以，这里大概总结几条代码规范建议，希望可以使我们的团队代码开发更有效率。

1.1 代码规范

良好的软件开发习惯和代码规范可以：便于团体协作开发，有效提高软件开发效率，增强软件代码质量，增强软件运行速度，提高软件的通用性可移植性，降低代码出错机率，去除冗余代码，减少软件维护时间和成本，等等。所以，这里大概总结几条代码规范建议，希望可以使我们的团队代码开发更有效率。

1. 分类目录：软件代码按功能分类和分子目录，相同或相似功能的子程序文件放在同一目录下。单个程序文件尽量只包含相同目标的子函数。

2. 注释说明：每个文件头，每个子程序头，每个函数头应该有简短的功能目的的说明注释。子函数内的关键部分（结构、算法、界面、变量等等）要有简短的描述。注释不需要长，而是应该言简意赅、简明扼要。

3. 层次结构：代码希望层次分明、缩进整齐、格式统一，以便调试检错和扩展。

4. 命名规范：变量和函数的命名尽量规范化。尽量使用有涵义的简短命名。规范、简短、有涵义、不重复、易区分、易查找、格式统一、尽量归类。同一类型的变量和函数使用类似的变量名字，命名可以加简短相同前缀（比如 file****，layer****，cell****），以便于管理和查找。

5. 精简扼要：冗余代码即刻删除！所有过时代码，如果希望今后继续参考，可以专门归总在某个目录或某个文件中。但是过时和冗余代码一定不要留在最终代码中。

6. 速度效率：代码和算法尽量有效率，注意计算速度。能够用向量和矩阵处尽量用，能够用循环尽量用，减少单个变量和单个操作。任何低效的代码，一经循环调用，对程序的速度影响可以是毁灭性的。

7. 用户体验：用户体验至上！降低软件使用所需要的学习时间，减少软件操作所需要的操作次数。比如设定目标“天下没有难做的版图!”，让入门门槛尽量降低。如果你自己都觉得怎样用不方便，那么用户更加会觉得不方便。有人说过，软件代码设计增加一次 click 会减少 10

8. 检查优化：写完的程序代码一定要检查和优化，你会经常发现原来你可以做得更好，代码可以更规范更有效。尽量把程序当作文章来写，而不是散漫的代码，那么你就自然会该写的写，不该写的不写，该花功夫的地方花功夫，不该花功夫的地方不浪费。

9. 交流沟通：及时和团队成员交流沟通。有任何问题需要团体其他成员注意或者解决，可以点名要求回答。我们可约定在需要检查的地方加入代码注释如下：//CHECK!!

统一思路，我们一定可以做到最好！