

训练任务：LCD背景

Logic System and Processors

理查德·苏斯塔，捷克技术大学电气工程系，布拉格2025年9月17日 第2.0版

目录

作业要求	2
VHDL代码要求	2
面向硬件设计	4
附录：图示目录	5

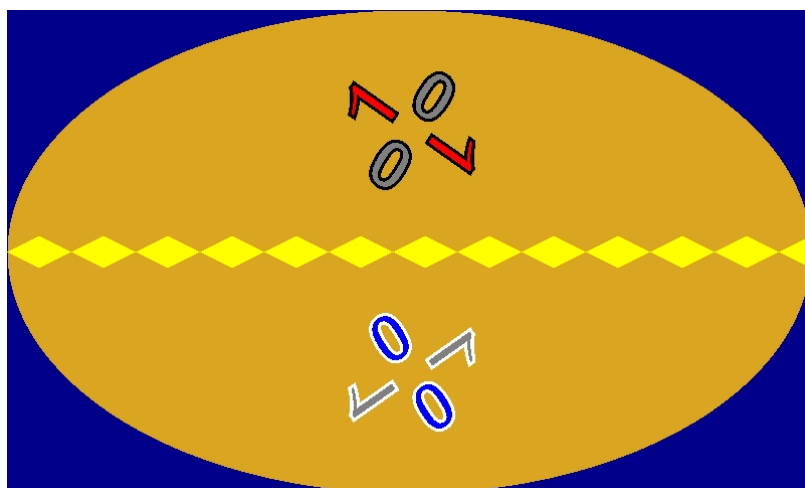


图1 - LCD控制面板背景示例

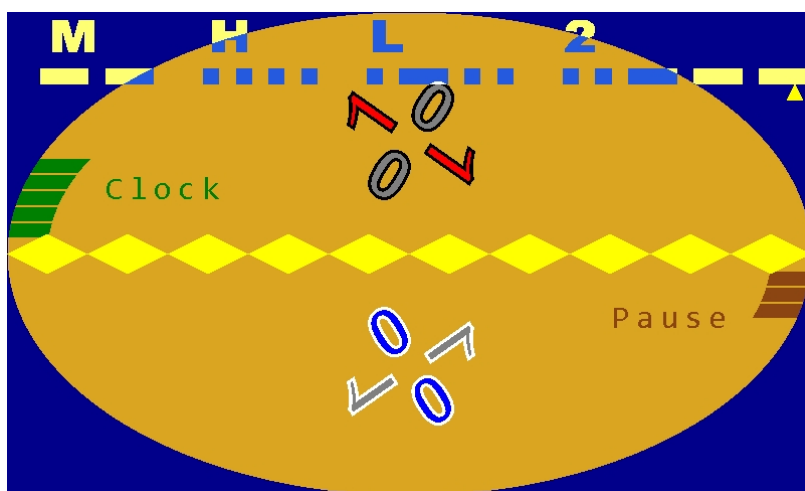


图2 - 摩尔斯信标控制面板中背景的应用

作业要求

实施缘由：

经理Freecoolin ([详见https://www.urbandictionary.com/define.php?term=freecoolin](https://www.urbandictionary.com/define.php?term=freecoolin)) 对饼图和优质表格进行了长期深入研究。历经无数个不眠之夜，他最终认定城市配送无人机无法依赖易受干扰的弱信号GPS系统。他下令采用**类似VOR**导航信标的设备——这种信标至今仍在航空领域使用。

VOR识别信号以摩尔斯电码形式反复传输。发射机控制面板需要背景图像，但FPGA存储图像需降低内存消耗。通过逻辑设计，我们能实现比JPEG或PNG更高效的智能压缩。

任务：设计组合逻辑电路，为Freecoolin'信标生成生态背景。请在<https://dcenet.felk.cvut.cz/fpga/>网站选择心仪的背景。

重要提示：

1. 所有LCD背景均包含曲线元素，且曲线始终为**圆形或椭圆形**片段。
2. 每个背景包含两张复杂图像，但内容完全相同。仅可进行旋转、翻转或变色处理。忽略像素形状差异——这些是矢量图形转位图过程中产生的栅格化误差。
3. 通过二进制幂的**除法与模运算**（即除法余数）生成重复形状。

学生马克索·格劳乔的毒舌吐槽：我知道工业级LCD面板没有我神仙iPhone的Super Retina XDR OLED屏幕。但就算最垃圾的玩意儿也能加载PNG图片。为什么非得用逻辑拍着Freecoolin的涂鸦说好话？我们这是在给博物馆供货吗？！

答：iPhone屏幕确实属于不同价位。你也能以33欧元（2025年9月5日Neven报价）买到800x480液晶面板，相当于iPhone替换屏奇迹成本的5%。即便在小处理器上，通过绘制生成图像也比从PNG或JPEG解压更高效——学生Maxo Groucho：哟，现在钻进A帐篷了，这可得老实干活了。

A：我只同意一半。你实际编写的VHDL代码会很简短。唯一可能更具挑战性的部分是理解电路的构建方式。但这正是我们LSP课程的核心内容。我们会找到合适的方法来描述它们在硬件层面的实现方式，无论是基于逻辑运算还是使用存储器：-)

VHDL代码要求

工业设计始终受限于诸多约束条件，这些资源限制是委托方普遍设定的基本条件。以下约束仅适用于构成图示的电路部分。

1. 算术运算仅允许使用**ieee.numeric_std**库。禁止使用已过时且普遍不推荐的旧版库：ieee.std_logic_arith、ieee.std_logic_unsigned、ieee.std_logic_signed。这些库在网络上某些旧版示例中可见，其定义的转换规则不同且不具备可移植性。
2. 项目严禁使用共享**变量**——此类变量仅适用于仿真环境。
3. 每个背景包含两张由ROM生成的复杂图像，除色调差异外完全相同，可能存在镜像翻转或 $N \times 90^\circ$ 旋转。您可能发现的微小像素差异，是矢量图形转换为位图（即**光栅化**）过程中产生的误差。因此可视为图像完全一致。且**两者**不存在任何重叠区域，为节省资源，必须**从同一ROM加载**。
4. 从内存块中，您只能生成这些图像，别无其他：-)
5. 电路中严禁出现**LATCH**和**LOOP**循环——其存在必然表明设计错误。请务必**使用FPGA-LCD工具包**中的Quartus检查工具核对译制项目。

6. 注意：仅VHDL的signed和unsigned类型与C语言的int类型直接对应。
- VHDL整数**数据类型具有自动调整位长的特性，上限为32位，但有时会检测到超过物理需求的位长或发生溢出。此外，该特性还会限制**数值范围**的长度。
7. 请检查RTL查看器中Quartus变量实现的位宽，确认是否存在不必要的位宽扩展导致下游逻辑（如加法器、比较器等）规模膨胀。即便功能正常，教师仍会因效率问题驳回此类方案。
- 建议：**a) 开发阶段保持整数定义无约束。待显示功能实现后，再为定义添加整数范围限制。
b) 计算时避免使用无符号类型，优先采用**整数**类型变量。无符号数可能引发比较错误且不支持减法运算。
8. VHDL自动整数类型没有固定位宽。在VHDL实体中定义输入或输出时若使用整数，必须始终添加范围约束。更专业的设计倾向于在实体中仅使用std_logic类型。
9. 需谨慎处理整数乘法运算。FPGA虽内置硬件乘法器，但自动**整数**类型可能发生溢出。其可靠工作上限约为28位。若需乘以超大常量，建议改用VHDL的有符号或无符号类型进行运算。
10. 在可能的情况下，选择**等于2的幂的**常数进行乘法运算，该运算通过位移实现。或者，优先选择**两个2的幂之和进行乘法运算**（通过两个位移的相加实现）。在电路中，乘以64更容易实现，同时也能轻松处理65（64+1）、66（64+2）、68（64+4）等结果，且无需专用乘法器。硬件乘法器位于FPGA内部；Veek-MT2中的Cyclone IV FPGA拥有266个此类单元，每个宽度为18×18位，可重配置为两个9×9位单元，但它们位于固定位置，这迫使逻辑模块必须靠近它们放置。加法器则可任意布局。因此应赋予编译器更大自由度。

除法与模运算要求

在逻辑设计中，通用除法或模运算（除法后的余数）只能通过模拟手动除法流程实现。这种方法效率低下且笨拙，因为它会导致冗长的减法器 and 比较器级联，消耗大量逻辑资源和时间。在设计中，我们力求避免采用低效实现的运算，因此必须省略通用除法和模运算。建议定期[使用FPGA-LCD工具包中的](#)Quartus检查工具验证编译后的项目，该工具可检测类似违规操作。这将节省您的时间——您的导师会用它检查作业，若检查器提示错误则会直接否决。

- 部分除法运算可被替代。例如，条件 $x/3 < y/5$ 可改写为 $5*x < 3*y$ 。
- 任何位置均可自由采用2的幂次除法（即2、4、8、16、32等），此类运算可通过位移操作实现。
- 即使除以2的幂次后的余数也会转换为选择低位。
- **其他除法和取模运算呢？**除法总能通过乘法近似实现；另参见《FPGA逻辑电路》第6.3.3至6.3.5章（第108至112页）。像素坐标x和y的递增除法也可用计数器实现，相关内容将在课堂讲授。

注：即使高速计算算法也应避免除法运算。即便在大型处理器上，除法效率仍不理想。上述限制不适用于编译时评估！因此常量初始化可使用任意运算，包括实数运算。测试台同样不进行综合而是通过仿真实现。在仅用于模拟的模块中，可自由使用任何运算。

请为硬件设计！

背景图片由逻辑条件生成，其中写入了你所熟悉的线性代数方程。**请务必选择适合硬件的图案！**

使用FPGA-LCD工具包中的LCD几何尺规。这些尺规类似于广为人知的Geodebra工具，但已适配LCD坐标系，并输出整数结果——这正是电路设计所需。

调整曲线与直线的方程参数。方向盘图标可启动自动优化功能，该功能会在直线或椭圆附近搜索更优的实现方案。

例如，297/800直线的斜率难以实现，因两个数值均无法截断。而斜率300/800=3/8更易处理，该直线与297/800仅相差0.22角度，可通过最大公约数(gcd)实现截断。随后，我们用方程中的较小数值进行乘法运算，这显著减少了所需的逻辑单元数量。

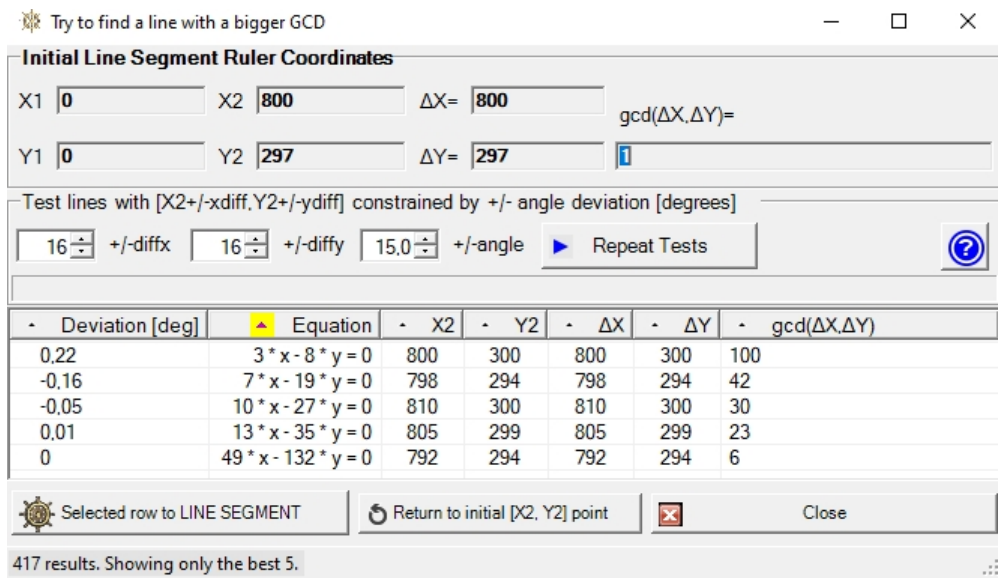


图3 - FPGA-LCD实用工具中LCD几何尺规的直线优化对话框

通过调整几何系数，我们实施举世闻名的“测量误差补偿”，巧妙地将结果乘以启发式专家奇迹般发现的作弊常数进行精炼。欢迎来到科学世界！☺

您大可将LSP网站上的LCD控制面板背景图集视为灵感素材。毕竟Freecoolin只是个普通经理，艺术缪斯可没邀请他参加才艺抽签。

随心调整色彩或形状，尽情设计专属图案吧 :) 您的创造力仅受图像复杂度限制。背景设计需不逊于原始模板，且包含两幅不同方向的图像。如有疑问请咨询导师。确实，过于简单的背景（例如蓝屏上几个圆圈等）将被拒收。

学生马克索·格劳乔的沮丧：可这样弗里库林就能把我们的积分单撕得粉碎了！！！！

A：若Freecoolin有异议，请替他考虑。你的方案必将降低功耗。即便仅减少毫瓦级的耗电量，每年也能节省9瓦时，以亿年为单位计算，相当于免去一座巨型发电厂的电力消耗。Freecoolin经理定会欢呼雀跃地兑现积分！毕竟在下届董事会，他可以自豪地宣称凭借“他的”底层设计，已为未来锁定巨额节省！

在GHDL中翻译和模拟VHDL代码约需7秒，我们仅执行一步操作——通过Visual Studio Code终端运行批处理文件。Quartus虽也支持安装Intel Questa仿真器，但其操作更为复杂。即便Questa偶尔能检测到更多错误，使用GHDL仍快得多也简单得多。☺

注：若将Quartus中调试过的VHDL代码编译后，即可加载至开发板。

但您忽略了任何关于时序定义不完整的编译器警告：

严重警告 (332168)： 以下时钟传输未分配时钟不确定性。为获得更精确的结果，请应用时钟不确定性分配或使用 `derive_clock_uncertainty` 命令...

这是我们BDF方案中为实现更简洁清晰的**顶层实体**所付出的代价:-)

作业结束

学生马克索·格劳乔抗议：结尾竟在最悬念处戛然而止，活像惊悚剧的跳跃式收尾？！你开玩笑吧。这下我们得在极其艰难、令人沮丧的漫长等待中煎熬了！

总会有现成的解决方案，能快速'复制'到不同背景中吧？

答：不同背景下的图像差异性太强。在我的代码中，插入这些图像仅需24行代码。复制它们对其他设计毫无用处。你只会因反复粘贴、重写而陷入混乱.....

在LCDlogic0中编写VHDL图像代码会更直观。将本次作业当作练习，随后即可进入最终任务——届时将提供简易作业。

教程《[LCD背景设计](#)》提供了几何形状模板，精选示例的源代码收录于[LCDbackgrounds_VeekMT2.zip](#)，可作为你创作的基石

。

以上就是全部内容.....

附录：图例表

图1 - LCD控制面板背景示例	1
图2 - 摩尔斯信标控制面板中的背景应用示例	1
图3 - FPGA-LCD实用工具中LCD几何尺的线优化对话框	4