My LaTeX Template

pastglory

December 27, 2020
v1.0

目录

摘 要

这个仓库主要保存我的 LaTeX 模版，用于各种文档的书写，目前实现的功能还较少，有待日后在使用中不断优化。

此模版目前设置了页边距、行距、各标题段距等距离，代码块使用Courier New字体，关键字蓝色高亮，注释使用灰色斜体。另外对于数学公式，调整了其编号方式，使之与章节编号关联。除上述内容以外，大部分与 LaTeX 中的article类无异。

Happy TeXing!

第一章 简介

你好，LaTeX! 这个仓库主要保存我的 LaTeX 模版，用于各种文档的书写。为了实现自由扩展的需求，绝大部分格式上的改动都放在 cls 文件中，并且所有文章内容都放在 src 文件夹下，main.tex只用于整理，作为顶层。

一直以来都习惯于使用他人提供的 TeX 模版，然而经常遇到一些细节上的改动想法却难以实现时，通常会选择妥协。直到最近要写的文档较多，突然想起了 LaTeX 这个老朋友，使用过很多次却没能好好研究它，于是下定决心从零开始，通过调整自己所需格式的方式学习研究。所以便有了这个模版。

模版中主要包含了对摘要格式、目录、多级标题、引用格式、参考文献、图片插入、代码插入、数学公式等内容的测试。为了测试参考文献格式是否正确，使用一篇稀疏运算加速的论文[1] 以及一篇老化预测的论文[2] 作为参考文献样例。

第二章 测试

本章主要用于测试该模版。

2.1 结构

图1为 FPGA 基本单元结构图。

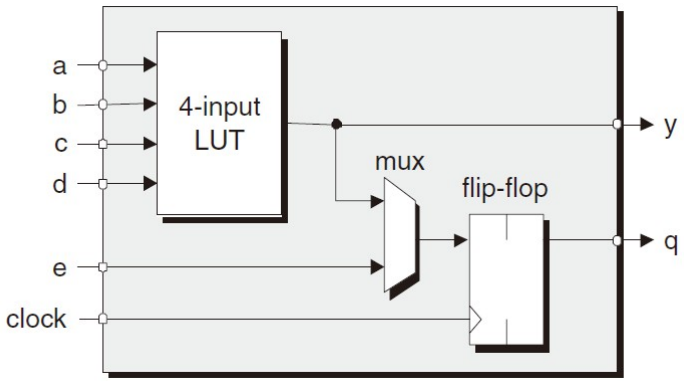

图 1: Slice of FPGA

FPGA 的基本单元是 Slice，主要包括 LUT、MUX 及触发器，其中每个 Slice 的大小和 FPGA 型号有关。

2.2 数学

这里有一个数学公式

$$\int_1^2 x dx = \frac{3}{2} \tag{2-1}$$

2.2.1 数学公式说明

这是一个简单的积分.

2.3 代码

推荐的 verilog 代码风格如下所示，但目前空格对齐的方式还没完全理解清楚。

```
module keyboard(
    input        clk,
    input        rst_n,
    input [3 : 0] col,
    output [3 : 0] key_pulse
);

reg [11 : 0] treg0; // tmp reg
reg [11 : 0] treg1; // tmp reg
reg [11 : 0] treg2; // tmp reg
reg [11 : 0] treg3; // tmp reg
wire [11 : 0] treg0_nxt = treg0 + 1'b1; // tmp reg
```

```
   next val
wire [11 : 0] treg1_nxt = treg1 + 1'b1; // tmp reg
   next val
wire [11 : 0] treg2_nxt = treg2 + 1'b1; // tmp reg
   next val
wire [11 : 0] treg3_nxt = treg3 + 1'b1; // tmp reg
   next val

// when tregx = 12'hffe and tregx_nxt = 12'hfff
// key_pulse will be 1
// then, tregx will be 12'hfff
// when tregx == 12'hfff, it will keep until
   keyboard not pushed
always @ (posedge clk or negedge rst_n) begin
   if (~rst_n) begin
      treg0 <= 12'b0;
      treg1 <= 12'b0;
      treg2 <= 12'b0;
      treg3 <= 12'b0;
   end
   else begin
      if (col[0]) begin
         if (treg0 != 12'hfff)
            treg0 <= treg0_nxt;
      end
      else begin
         treg0 <= 12'b0;
      end

      if (col[1]) begin
         if (treg1 != 12'hfff)
            treg1 <= treg1_nxt;
      end
```

```
      else begin
         treg1 <= 12'b0;
      end

      if (col[2]) begin
         if (treg2 != 12'hfff)
            treg2 <= treg2_nxt;
      end
      else begin
         treg2 <= 12'b0;
      end

      if (col[3]) begin
         if (treg3 != 12'hfff)
            treg3 <= treg3_nxt;
      end
      else begin
         treg3 <= 12'b0;
      end
   end
end

assign key_pulse[3] = (treg3 != 12'hfff) & (
   treg3_nxt == 12'hfff);
assign key_pulse[2] = (treg2 != 12'hfff) & (
   treg2_nxt == 12'hfff);
assign key_pulse[1] = (treg1 != 12'hfff) & (
   treg1_nxt == 12'hfff);
assign key_pulse[0] = (treg0 != 12'hfff) & (
   treg0_nxt == 12'hfff);

endmodule
```

第三章 总结

由于时间关系，除了上述内容之外，还有一些感兴趣的内容暂时没能实现，其中包括使用Tikz作图，插入图片及代码的姿势研究等。

参考文献

[1] Z. Zhang, H. Wang, S. Han, and W. J. Dally, “Sparch: Efficient architecture for sparse matrix multiplication,” in *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2020, pp. 261–274.

[2] M. Sadi, G. K. Contreras, J. Chen, L. Winemberg, and M. Tehranipoor, “Design of reliable socs with bist hardware and machine learning,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 11, pp. 3237–3250, 2017.