1 Introduction

A running theme through many of these presentation has been the idea of reuseable code. The idea being that once you code something up in a way that you like, say for example your preamble, you can reuse it the next time you're writing a report. But this is code, so we want to simplify this as much as possible. In fact, we don't ever want to copy and paste, we want it to be set up so that if something goes wrong and we fix some issue that fix propagates out everywhere. So how do we do that? LaTeX has a variety of commands and packages that facilitate this sort of thing at different levels as we explore these, keep in mind a couple of different ideas:

- Reusing material from your report in your presentation

- Modularizing documents; especially big documents

- Reusing general code

2 LaTeX Macro Expansion

3 A (very) Brief Pathing Primer

When setting up a modular document, often times its desireable to have it use relative paths `../../your-file.tex` instead of absolute ones `/home/your-user-name/your/path/to/file.tex`. Figure 1 shows a sample of the directory for this report.

4 Handling Paths Yourself

My preferred solution to the pathing issue is to define some variables and simply define the relative distance to the top level in each file independently. This way, no matter which file or subfile I'm using, the correct path is being followed, and only one variable needs to be set for a file.

```
1  \providecommand{\toplevel}{../..}
2  \providecommand{\importPath}{\toplevel/Shared/Imports}
3  \providecommand{\assetPath}{\toplevel/Assets}
4  \providecommand{\sharedPath}{\toplevel/Shared}
5
6  \documentclass[hidelinks, float=false, crop=false]{standalone}
7
8  \input{\importPath/preamble}
9  \input{\importPath/glossary}
10 \input{\importPath/symbols}
```

Listing 1: Manually solving the pathing issue

5 input Command

`\input{you-file.tex}` is the basic form of importing. Requires no external packages, but also doesn't do anything fancy. Grabs the file and drops it in-place.
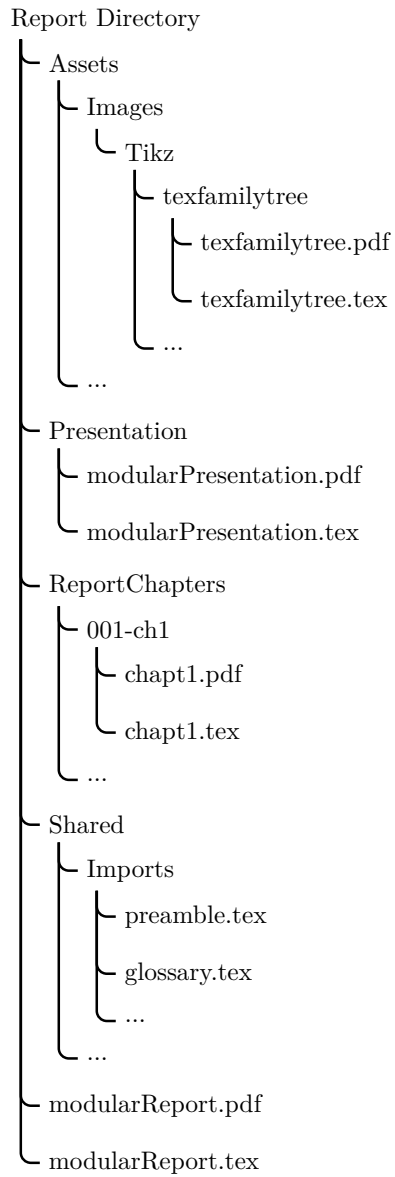
```
Report Directory
└─ Assets
   └─ Images
      └─ Tikz
         └─ texfamilytree
            ├─ texfamilytree.pdf
            └─ texfamilytree.tex
         └─ ...
   └─ ...
└─ Presentation
   ├─ modularPresentation.pdf
   └─ modularPresentation.tex
└─ ReportChapters
   └─ 001-ch1
      ├─ chapt1.pdf
      └─ chapt1.tex
   └─ ...
└─ Shared
   └─ Imports
      ├─ preamble.tex
      ├─ glossary.tex
      └─ ...
   └─ ...
├─ modularReport.pdf
└─ modularReport.tex
```

Figure 1: The directory structure for this report (as an example of a modular structure)

- Reusing your preambles (without copying and pasting)

- Reusing glossaries

6 import Package

Can aid with pathing, because it separated out the file from its path and so can find files imported from subimports following their source directory.

7 standalone Package

The major limitation of the import and input commands is that if they're being use with files you'd like to compile separately (for testing, or for any other reason) having multiple preambles will create break your files. The standalone package is a sophisticated tool that solves this by (optionally) ignoring the preambles of any files you're pulling in (or by combining preambles). This brings us closer to our modularized document because it means TiKz figures can be written separately, compiled, tested and all the rest without having to compile your entire document every time. As an added bonus, it introduces the `\includestandalone[\width=0.5\textwidth]{your-tikz-image}` which as shown, allows you to scale TiKz figures as though the were regular images. Very handy. It also allows you to break your report into multiple chapters.

- Breaking your document into modular parts

- TikZ Images

7.1 A Useful Additional Command

When working with a modularized document, it is often desireable to compile it with glossaries or bibliographies added at the end. Of course you can't just add the commands `\printbibliography` because then you would have duplicate bibliographies printed in the super file. My preferred approach to solving this is through the use of boolean flags. Listing 2, gives the code to be added to the preamble that creates a command to conditionally input a bibliography if the boolean flag `standaloneFlag` is set to `true`.

```
1   % For using the standalone package
2   \newboolean{standaloneFlag}
3   \setboolean{standaloneFlag}{true}
4   % Command to conditionally typeset a bibliography.
5   \newcommand{\standaloneBib}{%%
6   \ifthenelse{\boolean{standaloneFlag}}%%
7   {\printbibliography[heading=bibintoc]
8           \printglossary[type=symbols]
9           \printglossary[type=acronymtype]
10  \printglossary[type=main]}{}}
```

Listing 2: Conditionally typeset bibliography and glossaries

In the subfiles, now you can issue the command `\standaloneBib` to have the bibliography and glossaries printed if you in the subfile. In the super file then, issue the command `\setboolean{standaloneFlag}{false}` after loading the preamble, to have all instances of `standaloneBib` blocked, and then use the normal commands to have the bibliography and glossaries typset where you want them.

8   catchfilebetweentags Package

8.1   catchfilebetweentags

This handy package allows you to store a bunch of different things (say for example, equations) and pull them in to your document. This is really nice because it makes your document's code more easily read. It also makes it that you can reuse the equations in another place, say for example an associated presentations.

You can use it for anything from equations,

$$\Delta E_B \equiv \sum E_D(Reactants) - \sum E_D(Products) \tag{1}$$
$$\Delta E_B = (E_D(^2\mathrm{H}) + E_D(^2\mathrm{H})) - (E_D(^3\mathrm{He}) + E_D(\mathrm{n}^0))$$
$$\Delta E_B = (13.135\mathrm{MeV} + 13.135\mathrm{MeV}) - (14.931\mathrm{MeV} + 8.071\mathrm{MeV})$$
$$\Delta E_B = 3.27\mathrm{MeV}$$

$$\Delta = \begin{cases} \delta, & \text{for } A \text{ and } N \text{ both even} \\ 0, & \text{for } A + N \text{ odd} \\ -\delta, & \text{for } A \text{ and } N \text{ both odd} \end{cases} \tag{2}$$

$$E_B(Z, N) = \alpha_1 A - \alpha_2 A^{2/3} - \alpha_3 \frac{Z(Z-1)}{A^{1/3}} - \alpha_4 \frac{(N-Z)^2}{A} + \Delta \tag{3}$$

to tables,

Table 1: Table of specified parameters and achieved values

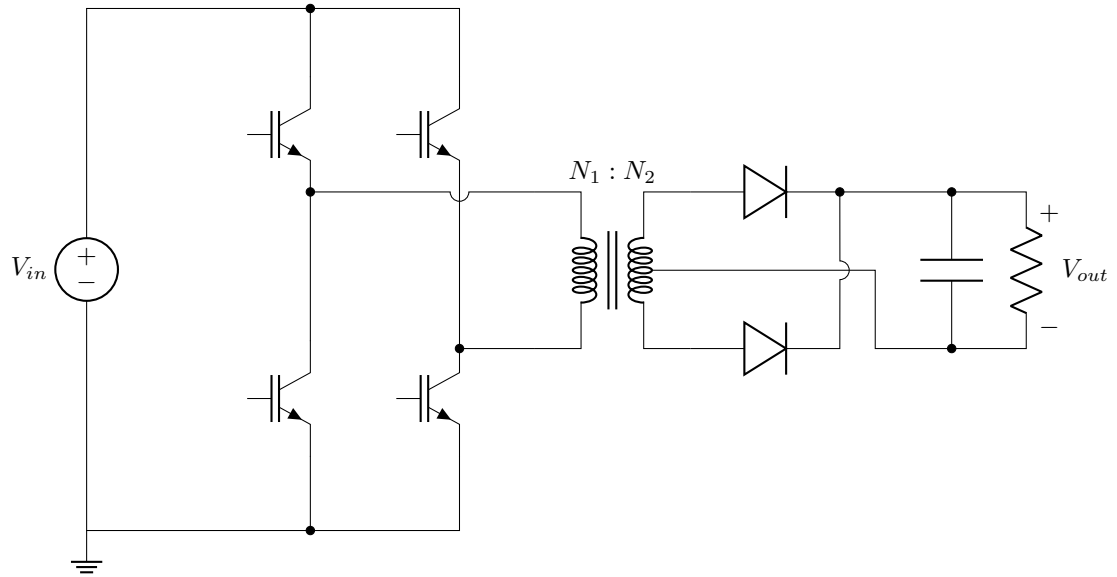| Parameter	Target	Calculated	Simulated
NF_{dsb}	≤ 4	11.17	5.95
$IIP3$	≥ -22	≥ -2.73	-4.98
1 dB Compression	≥ -32	≥ -12.73	-14.2
Gain	≥ 16	≥ -3.26	-4.58
I_{bias}			
I_{buf}			
I_{ref}		1	1
R_D	≤ 10	570	600
V_{lo}	≤ 1		
V_{rf}	≤ 1		

to figures.

Figure 2: An Example of a circuit (an isolated boost converter) done in circuit TikZ
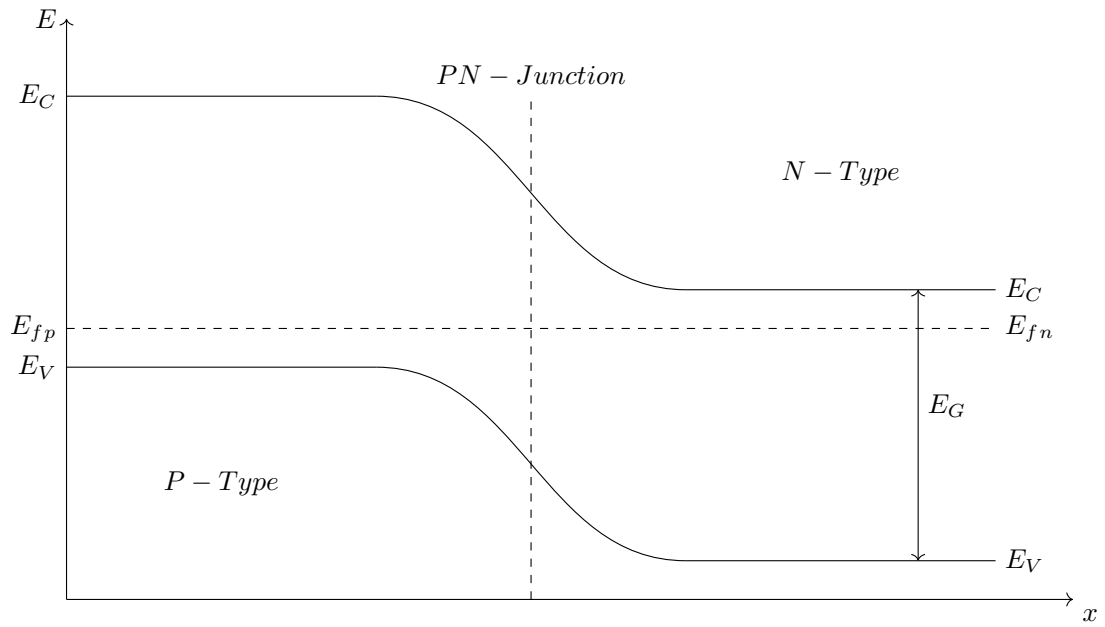

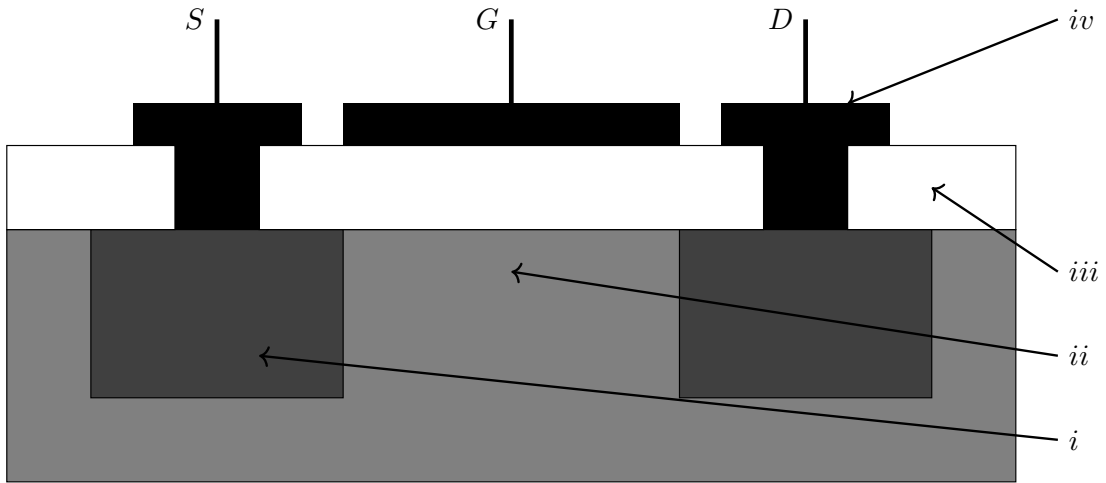Figure 3: A simple Example TikZ showing the band diagram of a PN-junction

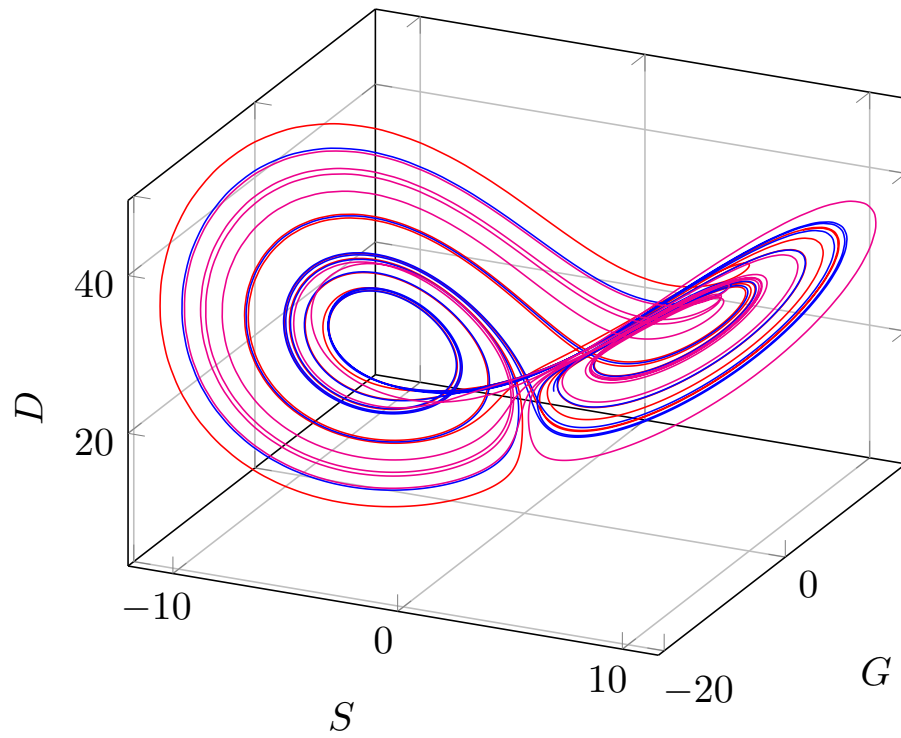Figure 4: A simple TikZ diagram showing a MOSFET

Figure 5: Lorenz Double Scroll Produced in LuaLatex

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x) \tag{4}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y \tag{5}$$

$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z \tag{6}$$

$$\tag{7}$$

Handy!

Symbols