

(ここに適切な報告書のタイトルを記入のこと)

(version 4.4)

学生番号: 094255xx

提出者: 渡邊 誠也

提出日: 2015 年 7 月 xx 日

締切日: 2015 年 7 月 xx 日

概要

(ここに報告書の概要を簡潔にまとめる．本文中にある文章をピックアップしつなげてまとめることで作成する．特に「おわりに」にある文章が活用できる場合が多い．)

本稿では，情報工学実験 (ハードウェア実験) の報告書の書き方について説明する．報告書執筆における基本的な注意点について説明する．さらに，報告書の作成と提出について簡単に述べる．この文書の \LaTeX ソースを読むことで， \LaTeX を使った報告書の執筆についてスキルを得ることを期待する．

1 はじめに

報告書の最初の節 (ここでは「はじめに」というタイトルとしている) では，実験の背景と目的，および，本報告書の構成について述べる．報告書の構成は，各自の判断で分かりやすい構成にすればよい．このサンプルでは，1 ～ 4 節の構成をとっているが，報告事項によって適宜変更する．各節では小節 (subsection) を設けるなどして，わかり易い構成をとって欲しい．

以下に，かなり省略してはいるが「はじめに」の記述例を示す．節 (section) や小節 (subsection) 等の参照には，`\ref{ラベル名}` といった記述を用いる．ラベルの定義は，本ドキュメントのソースの見ればわかるだろう．

【記述例】

本実験の目的は，..... である．本報告書では，..... について報告する．情報工学実験テキスト [2] の第 1 章から第 3 章の..... を報告する．

本報告書の構成は次のとおりである．まず 2 にて..... について述べる．3 では，..... する．最後に 4 で，本報告のまとめと今後の課題，..... を述べる．

2 報告書執筆における注意点

ここでは，執筆上の注意点について述べる．

2.1 報告内容に関する事項

実験レポートの報告内容に関する注意事項は次のとおりである。

1. 自分で文章を組み立てること。テキストや文献の文章を、一言一句をそのままコピー&ペーストしたのでは、モラル的に問題があるばかりか、自分のためにもまったく意味がない。

自分自身が記述してある内容をよく理解した上で自分の言葉で表現するように努めて欲しい。その際、事実を述べているのか、自分の意見や考えなのか、他人の考えや意見なのかを明確に区別して書くように心がけて欲しい。

ただし、ここで言う「自分の言葉」というのは、日常、会話をする際に使っている言葉ではなく、報告書の文章として適切な言葉のことである。口語体の文章は、報告書に用いる文章としては不適切である点に注意して欲しい。(例:「～だといいなぁと思う」は不適切な表現である)

2. 他人が読んで理解できるようにすること！報告書を読むのは担当教員だけだろうから、このことは担当教員も知っているだろうからこの部分だけ述べることにしよう」といった考えで報告書を書かないこと。第三者が報告書のみで内容を把握できるように書くように努めて欲しい。例えば「課題 x を行なった」とある場合「課題 x 」はどういった内容なのかを簡潔にまとめて報告書に記載しておくべきである。

3. 結果報告だけの内容にならないこと。得られた結果に対しては、その結果が妥当な結果なのかを考察するべきである。また自分で見出した問題点や、その問題点を解決するまでに考えたこと等を述べるべきである。

例えば「課題 x を行ない、正しく動いた」といった記述だけでは、不十分である。上の 2. とも関連するが「課題 x 」は何なのか「正しい」とは何をもって正しいと言えるのかをきちんと正確に記述する必要がある。

また「課題 y では、図 z に示す SFL 記述を作成した。(おわり)」といった記述も不十分である。課題の意図を理解し、その SFL 記述を作成するまでに考えたこと、SFL 記述の説明等を行なうべきである。逆に、やみくもにすべてを説明していたのでは、焦点が定まらず分かりにくくなるがあるので注意が必要である。その図や説明で主張したいことを伝えることができるように工夫して欲しい。

2.2 体裁に関すること

その他の注意事項(体裁に関すること)としては、次の事項が挙げられる。

1. 「～です」「～します」といったですます調で書かない。
2. 句読点は「，」「，」を用いる¹。~/emacs ファイルに次の式を記述をしておけば、デフォルトの句読点が「，」および「，」となる。

¹例えば、情報処理学会、電子情報通信学会の和文の論文誌では、句読点は「，」と「，」を用いている。

```

\begin{verbatim}
・ 最初の項目
・ 次の項目
・ 3 番目の項目
・ 最後の項目
\end{verbatim}

```

図 1: verbatim 環境による列挙の記述例

```

;; 2 バイト文字の句読点設定
(setq its-hira-period " . ") ;; 「 . 」は全角
(setq its-hira-comma " , ") ;; 「 , 」は全角

```

上記設定をしていない場合に「 . 」および「 , 」をタイプする場合、日本語モードでそれぞれ Z-. (大文字の Z をタイプ後、ピリオド (.) をタイプ)、Z-, (大文字の Z をタイプ後、カンマ (,) をタイプ) を入力する。

3. \TeX の verbatim 環境の乱用は避ける。しばしば、報告書全体を verbatim 環境にて書いている学生が見受けられる。そのような書き方では \TeX で美しい文書が作成できる機能を十分に利用できていないことになる。また、次以降に挙げる箇条書きやフォントの切替え等もできない。
4. 箇条書きや列挙を活用する。文章をだらだらと書いていたのでは分かりにくい場合もある。箇条書き (itemize) や列挙 (enumerate) 等を用いることでわかり易くなる場合には利用するとよい。
5. フォントを適切に選択する。例えば、強調したい単語は、ゴシック体や bold 体にするとか、プログラムリストや結果出力は、タイプライタフェイス (typewriter face) のフォントで書くといったようにフォントを適切に使用するとわかり易い。プログラムコードなどには、タイプライタフェイスのフォント (this uses typewriter face fonts) を用いる。数式は、 \LaTeX の数式モードを利用することで、適切な数式用のフォントを使う。このサンプルでは、適切なフォントを利用している。
6. 図や表には、図番号あるいは表番号とタイトルをつける。
7. 長いリストや結果の記載は、付録を活用する。

itemize 環境の使い方を知らない人がよく書く記述例を図 1 に示す。図 1 に示すコードからは、 \LaTeX により次のように整形される。

- ・ 最初の項目
- ・ 次の項目
- ・ 3 番目の項目
- ・ 最後の項目

図 1 に示した記述は、 \LaTeX では図 2 に示す記述にするべきである。図 2 に示すソースコードからは、 \LaTeX により次のように整形される。

```

\begin{itemize}
  \item 最初の項目
  \item 次の項目
  \item 3 番目の項目
  \item 最後の項目
\end{itemize}

```

図 2: itemize 環境による列挙の記述例

- 最初の項目
- 次の項目
- 3 番目の項目
- 最後の項目

各項目に番号を振りたい場合，itemize 環境の代わりに enumerate 環境を利用するとよい．図 1 に示した記述を enumerate 環境で記述した際には次のように整形される．

1. 最初の項目
2. 次の項目
3. 3 番目の項目
4. 最後の項目

最初から完全な報告書は要求しないが，徐々に改善していくように努力してもらいたい．なお，本ドキュメントの L^AT_EX ソースを公開するので，報告書を作成する際の参考にして頂きたい．この文書自体にはあらわれていないが，ソース内にコメント等を追加してあるので参照されたい．

3 報告書の作成と提出

L^AT_EX で作成した実験報告書は，PDF ファイルに変換して所定の場所へ提出する．提出方法等に関しては，別途，指示があるのでそちらにしたがうこと．

3.1 報告書の作成

報告書は，演習室計算機上の L^AT_EX で作成する．報告内容や執筆上の注意点は，2 で述べたとおりである．エディタ (Emacs 等) で報告書ファイルを記述し，L^AT_EX で整形する．L^AT_EX の使い方については，情報工学実験の範囲ではないのでここでは説明しない．適宜，プログラミング演習で習ったことを復習しておくこと．

DVI ファイルを PDF ファイルへ変換する際には，dvipdfmx コマンドを用いる．L^AT_EX で生成した DVI ファイル report.dvi を report.pdf というファイル名の PDF ファイルに変換する例を

```
nobuya@edu006[304]% platex report.tex
This is pTeX, Version p2.1.11, based on TeX, Version 3.14159 (EUC) (Web2C 7.3.1)(report.tex
pLaTeX2e <2000/11/03>+0 (based on LaTeX2e <2001/06/01> patch level 0)
(/usr/share/texmf/ptex/platex/base/jarticle.cls
Document Class: jarticle 1999/05/18 v1.1q Standard pLaTeX class
(/usr/share/texmf/ptex/platex/base/jsize10.clo))
(/usr/share/texmf/ptex/platex/base/ascmac.sty
(/usr/share/texmf/ptex/platex/base/tascmac.sty))
(/usr/share/texmf/tex/latex/graphics/graphicx.sty
(/usr/share/texmf/tex/latex/graphics/keyval.sty)
(/usr/share/texmf/tex/latex/graphics/graphics.sty
(/usr/share/texmf/tex/latex/graphics/trig.sty)
(/usr/share/texmf/tex/latex/config/graphics.cfg)
(/usr/share/texmf/tex/latex/graphics/dvips.def))) (report.aux) [1] [2] [3]
[4] (alu4.sfl.tex) [5] [6] <count4_gtkwave.eps> [7] (report.aux) )
Output written on report.dvi (7 pages, 23196 bytes).
Transcript written on report.log.
nobuya@edu006[305]%
```

図 3: DVI ファイルの作成例

図 3 に示す．作成した PDF ファイルがプレビューア（xpdf，acroread 等）で読めるかを提出する前に確認することを忘れないで欲しい．

4 おわりに

最後の節では，本報告書のまとめを述べる．どういったことを報告したのか，目的は達成できたか，どういう成果があったか，どういったことが分かったのか，不十分な点や課題事項は何かを簡潔にまとめる．例えば，実験テキスト [2] の章末にあるチェック項目の事項が達成できたかを示せばよいだろう．

なお，この「おわりに」に書かれていることは，基本的に「おわりに」に至るまでの本文中に書かれていることであるべきである．つまり，ここには本文中で書かれていない新たなことを書くことは避けるべきである．

【記述例】

本報告書では，..... について報告した．本実験テーマの目的である (1)，(2) ... \cdots (n) ... は，それぞれ達成できた．また，..... により，..... であることがわかった．今後の課題としては がある．

参考文献

- [1] 著者名，文献のタイトル，(もしあればページ)，出版社，出版年．
- [2] 渡邊誠也，ハードウェア記述言語を用いたマイクロプロセッサの設計，情報工学実験テキスト，岡山大学工学部情報工学科 (2003)

付録

報告書本体部分に掲載するとわかりにくくなるものは、付録に掲載する。例えば、出力結果全体や SFL 記述のリストなどである。本文中に掲載したほうが、分かりやすい場合は本文中に掲載すべきであり、その場合でも掲載は必要最小限になるように努めるべきである。

A 4ビット ALU の SFL 記述

```
1  /* (alu4.sfl) */
2  %i 'add4.h'
3  %i 'alu4_func.def'
4
5  module alu4 {
6      input  a<4>, b<4>; /* input data */
7      input  func<3>;    /* function */
8      output out<4>;     /* output data */
9      instrin enable;
10
11      add4 adder;
12
13      instruct enable alt {
14          func == THAFUNC: out = a;
15          func == THBFUNC: out = b;
16          func == ANDFUNC: out = a & b;
17          func == ORFUNC:  out = a | b;
18          func == XORFUNC: out = a @ b;
19          func == NOTFUNC: out = ^a;
20          func == ADDFUNC: out = adder.enable(a, b, 0b0).sum;
21          func == SUBFUNC: out = adder.enable(a, ^b, 0b1).sum;
22      }
23  }
24  /* End of file (alu4.sfl) */
```

図 4: 4ビット ALU の SFL 記述

B 表の例

表の例を表 1 と表 2 に示す²。

表を参照する際には、単に表を掲載するだけではなく、「表 1 に論理合成で得られた諸量をまとめた結果を示す」といった文章にて、参照している「表」が何を示しているのかを説明した後に、その詳細に関する説明が必要である。

²表の見出し（表題）は、表の上に置く。一方、図の見出しは図の下に置く。

表 1: 論理合成で得られた諸量のまとめ

モジュール	最大遅延 (ns)	最大動作周波数 (MHz)	ゲート数	実装面積 ($1000\mu\text{m}^2$)	消費電力 ($\mu\text{W}/\text{MHz}$)	最大動作周波数で 動作時の消費電力
4 ビット加算器 4 ビット桁上げ先見加算器 8 ビット桁上げ先見加算器 16 ビット桁上げ先見加算器 4 ビット ALU 8 ビットシフタ 4 ビットカウンタ 10 進カウンタ 4 ビットアップダウンカウンタ						

表 2: マイクロプロセッサ p16 の設計状況

モジュール			設計状況
1.	16 ビット加算器	add16	(0) 未着手
2.	16 ビット実行ユニット	exec16	(1) 設計中
3.	レジスタファイル	reg16x8	(2) 動作確認済み
4.	増加器	inc16	(2) 動作確認済み
5.	メモリユニット	memunit	(2) 動作確認済み
6.	p16 トップモジュール	p16	(3) 設計完了
7.	p16 制御ユニット	p16_controller	(3) 設計完了

C 画像の取り込み

画像の取り込みの例を図 5 に示す．図 5 に示す画像は，画面に表示されるウィンドウを取り込んで，EPS ファイルとして保存したものを TeX に貼りつけたものである．画像の取り込み方法は C.1 にて具体的に説明する．

C.1 画像の取り込み方法

以下では，画面に表示されている画像（ウィンドウ）を EPS として保存する方法を説明する．

1. 取り込みたい図を画面に表示させ，ターミナルから gimp とタイプし，gimp を起動させる．
2. 「The GIMP」というタイトルのついたウィンドウの「ファイル」メニューから「取り込み」「画面取り込み...」を選択する．
3. 「画面取り込み」といタイトルのついたウィンドウが表示されるので，そのウィンドウにて適切に設定を行なった後に「了解」ボタンをクリックする．
4. カーソルが十字にかわるので，取り込みたいウィンドウの上でクリックする．すると，取り込まれたウィンドウの画像が表示される．
5. 取り込まれた画像ウィンドウの上にカーソルをあわせ，右クリックを押し「ファイル」メニューの「別名で保存」を選択する．

6. 保存先とファイル名を指定して、「了解」ボタンをクリックする．ファイル名の最後を .eps としておくと，自動的に EPS ファイルとして保存される．
7. 保存された EPS ファイルを報告書の TeX ソースから読み込むことで，報告書に画像（画面）を張り付けることができる．

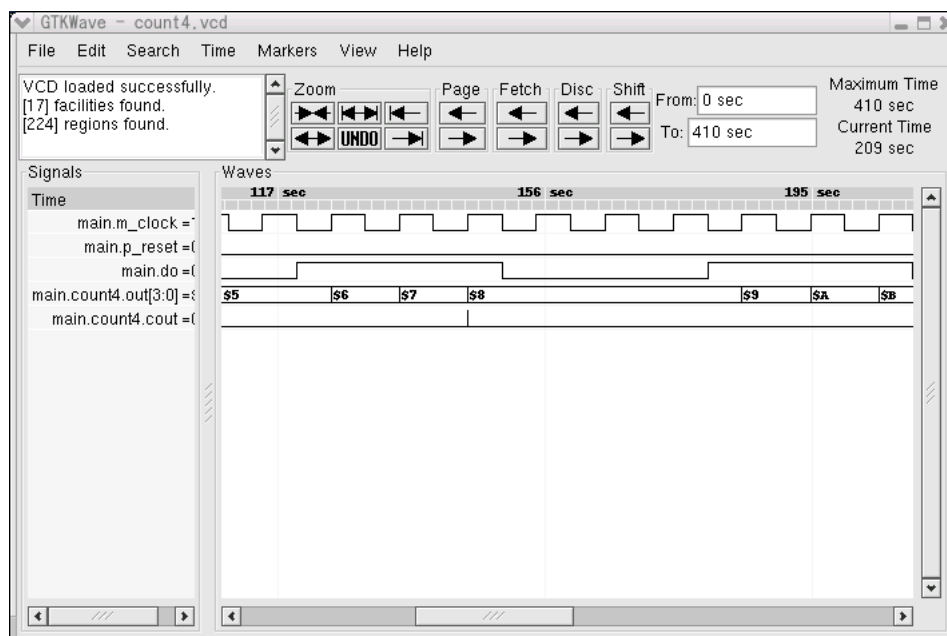


図 5: 4 ビットカウンタをシミュレーションした際の信号波形