

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI  
I TECHNIK INFORMATYCJNYCH



Instytut Mikroelektroniki i Optoelektroniki

# Praca dyplomowa inżynierska

na kierunku Elektronika, Informatyka i Telekomunikacja  
w specjalności Elektronika i inżynieria komputerowa

Optymalizacja nieoptymalnych rozwiązań w celu uzyskania dyplomu

Adam Kowalski

Numer albumu 123456789

promotor

prof. dr hab. Iks Igrekowski

WARSZAWA 2017



## Streszczenie

**Tytuł:** Optymalizacja nieoptymalnych rozwiązań w celu uzyskania dyplomu

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum eget congue neque. Nunc consectetur ipsum quis urna molestie ultricies. Nam eleifend tempor lorem, a convallis ligula blandit id. Integer aliquet, lorem a porta congue, tortor mi auctor neque, a gravida urna sem vitae nibh. Mauris et risus sit amet tortor blandit lacinia. In a mollis nulla. Vestibulum sodales ex quis enim condimentum, non porta nulla euismod. Suspendisse pulvinar ante non lacus rutrum ultrices.

In a fringilla justo. Mauris convallis orci sed ligula efficitur tincidunt. Morbi vestibulum, enim non fringilla aliquam, ex nisi mattis nisi, in tincidunt arcu nulla non est. Sed lacinia nunc eget placerat vehicula. Donec a viverra massa, ut tempor diam. Fusce venenatis erat et viverra commodo. Curabitur nisi dui, dignissim id fermentum vel, consequat ut est. Duis iaculis est non nibh commodo scelerisque. Donec vitae dolor massa.

Nulla efficitur vulputate est, ut molestie libero. Vivamus facilisis, velit ultricies facilisis cursus, justo dolor hendrerit lectus, eleifend convallis velit nunc vel ante. Duis rhoncus tempor nunc, tempus semper lacus ullamcorper at. Vestibulum imperdiet, eros quis tempus luctus, elit mauris aliquam turpis, non iaculis purus turpis iaculis quam. Aenean eu ante in eros ultricies tincidunt. Sed ut dignissim diam, a dignissim quam. Phasellus accumsan lacus dolor, id faucibus justo ullamcorper ut.

Fusce vestibulum arcu ut urna commodo tincidunt. Ut non porta leo, dignissim tristique leo. Maecenas facilisis, ligula ac vehicula tincidunt, erat risus tempor arcu, vel ornare dui lectus eget libero. Duis mollis felis libero. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nulla ut mauris scelerisque, consequat nulla id, consectetur sapien. Vestibulum commodo, sem eu fermentum varius, justo ante tincidunt dolor, sed egestas ante erat in ligula. Proin sagittis magna odio, eget luctus tellus aliquam at. In eget purus non leo egestas pharetra.

Ut ut nisl nec ipsum eleifend sodales nec nec ligula. In blandit, tellus id semper mattis, velit nunc consequat odio, et finibus nisl nisl eu nisi. Proin molestie euismod elit, fermentum tincidunt ex fringilla ac. Vivamus sed metus lectus. Vestibulum vestibulum massa dapibus, imperdiet odio sit amet, ornare mi. Cras id nisl ut urna finibus elementum in et enim. Morbi rutrum nunc a mauris sodales scelerisque. Vestibulum magna metus, facilisis a felis sed, dictum efficitur lectus. Sed ante tellus, egestas ac ex in, luctus elementum quam. Suspendisse lacus magna, hendrerit in tempor non, pellentesque ac felis. Aliquam id condimentum leo. Vestibulum consectetur dapibus viverra.

**Słowa kluczowe:** *Android, tablet, edytor schematów.*



## Abstract

**Title:** Optimization of non-optimal solutions in order to get a diploma

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum eget congue neque. Nunc consectetur ipsum quis urna molestie ultricies. Nam eleifend tempor lorem, a convallis ligula blandit id. Integer aliquet, lorem a porta congue, tortor mi auctor neque, a gravida urna sem vitae nibh. Mauris et risus sit amet tortor blandit lacinia. In a mollis nulla. Vestibulum sodales ex quis enim condimentum, non porta nulla euismod. Suspendisse pulvinar ante non lacus rutrum ultrices.

In a fringilla justo. Mauris convallis orci sed ligula efficitur tincidunt. Morbi vestibulum, enim non fringilla aliquam, ex nisi mattis nisi, in tincidunt arcu nulla non est. Sed lacinia nunc eget placerat vehicula. Donec a viverra massa, ut tempor diam. Fusce venenatis erat et viverra commodo. Curabitur nisi dui, dignissim id fermentum vel, consequat ut est. Duis iaculis est non nibh commodo scelerisque. Donec vitae dolor massa.

Nulla efficitur vulputate est, ut molestie libero. Vivamus facilisis, velit ultricies facilisis cursus, justo dolor hendrerit lectus, eleifend convallis velit nunc vel ante. Duis rhoncus tempor nunc, tempus semper lacus ullamcorper at. Vestibulum imperdiet, eros quis tempus luctus, elit mauris aliquam turpis, non iaculis purus turpis iaculis quam. Aenean eu ante in eros ultricies tincidunt. Sed ut dignissim diam, a dignissim quam. Phasellus accumsan lacus dolor, id faucibus justo ullamcorper ut.

Fusce vestibulum arcu ut urna commodo tincidunt. Ut non porta leo, dignissim tristique leo. Maecenas facilisis, ligula ac vehicula tincidunt, erat risus tempor arcu, vel ornare dui lectus eget libero. Duis mollis felis libero. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nulla ut mauris scelerisque, consequat nulla id, consectetur sapien. Vestibulum commodo, sem eu fermentum varius, justo ante tincidunt dolor, sed egestas ante erat in ligula. Proin sagittis magna odio, eget luctus tellus aliquam at. In eget purus non leo egestas pharetra.

Ut ut nisl nec ipsum eleifend sodales nec nec ligula. In blandit, tellus id semper mattis, velit nunc consequat odio, et finibus nisl nisl eu nisi. Proin molestie euismod elit, fermentum tincidunt ex fringilla ac. Vivamus sed metus lectus. Vestibulum vestibulum massa dapibus, imperdiet odio sit amet, ornare mi. Cras id nisl ut urna finibus elementum in et enim. Morbi rutrum nunc a mauris sodales scelerisque. Vestibulum magna metus, facilisis a felis sed, dictum efficitur lectus. Sed ante tellus, egestas ac ex in, luctus elementum quam. Suspendisse lacus magna, hendrerit in tempor non, pellentesque ac felis. Aliquam id condimentum leo. Vestibulum consectetur dapibus viverra.

**Keywords:** *Android, tablet, schematic editor.*



Tę kartkę należy zastąpić oświadczeniem o autorstwie pracy.





# Spis treści

<b>Spis treści</b>	<b>1</b>
<b>1 Korzystanie z szablonu</b>	<b>3</b>
1.1 Szablon . . . . .	3
1.2 Kompilowanie tekstu . . . . .	3
1.3 Zawartość szablonu . . . . .	4
1.4 Struktura głównego dokumentu . . . . .	4
<b>2 Wskazówki i przykłady</b>	<b>7</b>
2.1 Formatowanie tekstu . . . . .	7
2.2 Przypisy . . . . .	8
2.3 Odsyłacze . . . . .	9
2.4 Tabele . . . . .	9
2.5 Cytowanie kodu źródłowego . . . . .	10
2.6 Ilustracje . . . . .	11
2.7 Składanie wzorów . . . . .	12
2.8 Ułamki piętrowe . . . . .	13
2.9 Justowanie wzorów . . . . .	13
2.10 Ingerowanie w skład dokumentu . . . . .	14
2.11 Spis literatury . . . . .	15
<b>Bibliografia</b>	<b>17</b>
<b>Wykaz symboli i skrótów</b>	<b>19</b>
<b>Spis rysunków</b>	<b>21</b>
<b>Spis tabel</b>	<b>23</b>
<b>Spis załączników</b>	<b>25</b>



# 1 Korzystanie z szablonu

*Szablon jest zgodny z wymaganiami dla prac dyplomowych określonymi w Zarządzeniu nr 57/2016 Rektora Politechniki Warszawskiej z dnia 15 grudnia 2016r.*

*Ze względu na dwustronny skład pracy, w wielu miejscach pojawiają się puste strony. Wyniki to z tego, że kolejne rozdziały rozpoczynają się od strony nieparzystej.*

## 1.1 Szablon

Niniejszy dokument jest propozycją składu pracy inżynierskiej lub magisterskiej wykorzystującego system  $\text{\LaTeX}$  ([1,6,7]). Dokument został pomyślany jako szablon, który przy minimalnym wysiłku można wypełnić właściwą treścią. Korzystanie z szablonu nie wymaga biegłości w  $\text{\LaTeX}$ -u. Wystarczy wykorzystać przykłady typowych elementów występujących w tekście pracy dyplomowej: wyliczeń, wzorów, tabel, ilustracji, itp.

Szablon pracy pełni dwie funkcje. Po pierwsze, zawiera wszystkie niezbędne definicje (zgromadzone w głównym pliku szablonu) oraz sformatowaną stronę tytułową i strony streszczeń. Po drugie, tekst źródłowy dokumentu jest zbiorem przykładów, które można skopiować i przystosować do własnych potrzeb.

Warto tu wspomnieć o prawidłowej wymowie i odmianie nazwy  $\text{\LaTeX}$ . Jest ona skrótem od [Leslie] Lamport  $\text{\TeX}$ , który jest zbiorem makr stanowiących nadbudowę dla systemu składu  $\text{\TeX}$ . Twórcą  $\text{\TeX}$ -a jest prof. Donald E. Knuth, amerykański matematyk i informatyk pracujący na Uniwersytecie Stanforda. W nazwie występuje litery T, E i X, które w istocie reprezentują greckie: tau, epsilon oraz chi ( $\text{\TeX}$ ). Są to początkowe znaki greckiego słowa τέχνη (technē), od którego pochodzi wyraz *technika*. Tak więc poprawne formy to: "latech", "latecha", "latechowi", "latecha", "latechem", itd.

Z oczywistych względów niniejsze opracowanie ten nie jest kursem  $\text{\LaTeX}$ -a. W celu pogłębienia wiedzy można skorzystać z materiałów dostępnych w Internecie, np.:

- <http://www.latex-kurs.x25.pl/>,
- <http://en.wikibooks.org/wiki/LaTeX>,
- <http://en.wikipedia.org/wiki/LaTeX>,
- <http://tex.stackexchange.com>,
- <http://www.tug.org/>.

## 1.2 Kompilowanie tekstu

Szablon ten powstał przy założeniu, że do kompilacji plików źródłowych używany będzie program `lualatex` – jeden z kompilatorów dostępnych w standardowej dystrybucji  $\text{\LaTeX}$ -a. W przypadku stosowania innego kompilatora (np. `pdflatex`) należy dokonać szeregu istotnych zmian w deklaracjach pakietów i definicjach makr co wymaga dobrej znajomości  $\text{\LaTeX}$ -a i, co nie mniej istotne, zajmie cenny czas. Tak więc znacznie rozsądniej jest przed przystąpieniem do pracy skonfigurować środowisko edycyjne (np. *TeXStudio*) tak, aby domyślnym kompilatorem był właśnie `lualatex`.

## 1.3 Zawartość szablonu

Pliki wykorzystywane w dokumencie zgromadzone w czterech podkatalogach: *bibliografia*, *fonts*, *grafika* i *tex*. Pierwszy z nich zawiera bazę bibliograficzną (*spis.bib*) oraz definicje formatu bibliografii (*plainurl.bst*). Drugi podkatalog zawiera stosowany na stronie tytułowej font *Belgrano* (licencja *SIL Open Font License*) a trzeci jest magazynem plików graficznych. Natomiast w podkatalogu *tex* znajdują się pliki źródłowe:

- *praca\_dplomowa2.tex* – główny plik szablonu,
- *glossary.tex* – plik zawierający definicje stosowanych w pracy symboli i skrótów,
- *title.tex* – plik strony tytułowej,
- *abstract.tex* – plik zawierający streszczenia,
- *chapter1.tex*, *chapter2.tex* i *chapter3.tex* – pliki zawierające teksty źródłowe rozdziałów,
- *appendixA.tex* i *appendixB.tex* – przykładowe pliki dodatków.

Liczba rozdziałów w konkretnej pracy może być oczywiście inna niż w tym szablonie i w takim przypadku należy dopasować liczbę plików *chapterX.tex*.

## 1.4 Struktura głównego dokumentu

W dokumencie głównym wczytywane są (poleceniem `\input`) pozostałe pliki źródłowe. Są tu także zdefiniowane niezbędne parametry oraz ustawione opcje wpływające na proces składania tekstu. Jedną z takich opcji jest `\setmainfont{}` określająca font jakim ma być złożony tekst:

```
%%% Konfiguracja fontu
%%%\setmainfont{TeX Gyre Pagella}
\setmainfont{FreeSerif}
```

Jeśli instrukcja `\setmainfont{}` zostanie usunięta to zastosowany będzie standardowy font *Latin Modern Roman*. Można też uaktywnić instrukcję zmieniającą domyślny font na *TeX Gyre Pagella*.

### 1.4.1 Strona tytułowa

Pliku definiującego wygląd strony tytułowej nie trzeba samodzielnie edytować. Można to zrobić, ale wygodniej jest zdefiniować w pliku głównym makra, które są w nim wywoływane:

- `\newcommand\rok{ }`,
- `\newcommand\rodzajpracy{ }`,
- `\newcommand\kierunek{ }`,
- `\newcommand\specjalnosc{ }`,
- `\newcommand\tytulpl{ }`,
- `\newcommand\tytulen{ }`,
- `\newcommand\autor{ }`,
- `\newcommand\nralbumu{ }`,
- `\newcommand\promotor{ }`,

jak to pokazano poniżej:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Parametry strony tytułowej %%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\newcommand\rok{2017}
\newcommand\rodzajpracy{Praca dyplomowa \\\vskip6pt inżynierska}
\newcommand\rodzajpracy{Praca dyplomowa \\\vskip6pt magisterska}
\newcommand\kierunek{Elektronika, Informatyka i Telekomunikacja}
\newcommand\specjalnosc{Elektronika i inżynieria komputerowa}
\newcommand\tytulpl{Optymalizacja nieoptymalnych rozwiązań w celu uzyskania dyplomu}
\newcommand\tytulen{Optimization of non-optimal solutions in order to get a-diploma}
\newcommand\autor{Adam Kowalski}
\newcommand\nralbumu{123456789}
\newcommand\promotor{prof. dr hab. Iks Igrekowski}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

### 1.4.2 Definicje symboli i skrótów

Nowy format pracy dyplomowej zawiera wykaz symboli i skrótów. W celu jego wygenerowania należy zaktualizować plik *glossary.tex*.

### 1.4.3 Streszczenia

Plik zawierający streszczenia (w języku polskim i w języku angielskim) można edytować samodzielnie. Oprócz samych streszczeń należy zmodyfikować obie listy słów kluczowych.

### 1.4.4 Rozdziały

Zalecane jest umieszczenie tekstu źródłowego każdego z rozdziałów w oddzielnym pliku. W przypadku innej liczby rozdziałów niż w niniejszym dokumencie należy odpowiednio zmodyfikować polecenia wciągające te pliki w dokumencie głównym:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% wczytanie plików zawierających kolejne rozdziały pracy
\input {./chapter1.tex}
\input {./chapter2.tex}
\input {./chapter3.tex}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

### 1.4.5 Ilustracje

Zakłada się, że pliki graficzne zawierające ilustracje są umieszczone w katalogu *grafika*.

### 1.4.6 Bibliografia

Zakłada się, że baza bibliograficzna znajduje się w pliku o nazwie *spis.bib* a sam plik w podkatalogu *bibliografia* co znajduje swoje odzwierciedlenie w poleceniu:

```
\bibliography{./bibliografia/spis}
```



## 2 Wskazówki i przykłady

W rozdziale tym zamieszczono szereg wskazówek i przykładów formatowania w  $\LaTeX$ -u typowych elementów pracy dyplomowej. Tekst źródłowy przykładów można skopiować i przystosować do własnych potrzeb.

### 2.1 Formatowanie tekstu

#### 2.1.1 Font, style, itd.

Font, stopień pisma tekstu i tytułów rozdziałów, wielkość wcięć i odstępów, itd. ustala sam  $\LaTeX$  na podstawie definicji zawartych w szablonie dokumentu. Użytkownik nie musi niczego definiować samodzielnie.

#### 2.1.2 Akapity

Akapit nie wymaga specjalnego formatowania. Tekst należy wpisywać bez wymuszania przejścia do nowego wiersza czyli tzw. ręcznego wstawiania znaków powrotu karetki. Liczba odstępów pomiędzy wyrazami, tzw. *spacji*, większa niż jeden jest interpretowana jak jeden odstęp.

W celu rozpoczęcia nowego akapitu należy pozostawić wiersz odstępu. Wszystkie parametry tekstu takie jak wcięcia akapitowe, odstępy między wierszami i między akapitami ustawia sam  $\LaTeX$ .

Należy pamiętać, aby nie pozostawiać na końcu wiersza samotnego spójnika (np.: "i", "a", "z"). W tym celu pomiędzy spójnik i następny wyraz należy wstawić tzw. "niełamliwy" odstęp. W  $\LaTeX$ -u uzyskuje się taki odstęp przez umieszczenie pomiędzy wyrazami znaku "~" zamiast zwykłego odstępu, np. "w~tekście".

#### 2.1.3 Wstawianie specjalnych symboli

Obecnie, dzięki pakietom językowym, nie ma potrzeby stosowania tzw. *escape codes* w celu uzyskania znaków narodowych (np. `\k{a}` dla uzyskania  $\text{ą}$ ). Wystarczy zapisać tekst w pliku z kodowaniem UTF-8 oraz użyć odpowiedniego pakietu, np:

```
\usepackage{polyglossia}
\setdefaultlanguage[]{}{polish}
```

Często zachodzi potrzeba wstawienia symbolu, np. litery z greckiego alfabetu. Rozwiązanie tego problemu zależy od tego w jakim elemencie tekstu chcemy uzyskać taki symbol.

Jeśli symbol występuje we wzorze to rozwiązaniem jest zastosowanie trybu matematycznego (szczegóły w rozdziale 2.7), w którym pożądaný symbol wstawiamy za pomocą specjalnego kodu, np.: `\mu` –  $\mu$ , `\pi` –  $\pi$ .

Jeśli symbol jest nazwą jednostki to wtedy można "awaryjnie" zastosować tryb matematyczny. Powstaje formuła osadzona w tekście, np.  $3\ \mu\text{m}$ . Jednak znacznie lepiej jest zastosować specjalne polecenia z pakietu `siunitx` uzyskując napis  $3\ \mu\text{m}$ . Szczegóły w rozdziale 2.1.6.

#### 2.1.4 Wyróżnianie tekstu

W celu wyróżnienia fragmentu tekstu można zastosować *specjalną odmianę kroju czcionki zwaną italikiem*, **wytluszczenie** lub podkreślenie. Należy pamiętać, żeby nie nadużywać tej formy ekspresji. Zaleca się wybranie jednej z tych metod wyróżniania tekstu i konsekwentne jej stosowanie w całej pracy. Wytluszczenia są często zarezerwowane dla oznaczania wektorów, macierzy, itd. Dlatego zwykle wyróżnienie tekstu najlepiej zrealizować *przez zastosowanie italików*.

### 2.1.5 Wyliczenia

Wyliczenia, zwane też listami, tworzymy korzystając z otoczeń (ang. *environments*).  $\LaTeX$  formatuje listy całkowicie automatycznie. Najczęściej stosuje się dwa rodzaje list:

- *nieuporządkowaną*, uzyskiwaną przez użycie otoczenia `itemize`,
- *uporządkowaną*, uzyskiwaną przez użycie otoczenia `enumerate`.

Podczas składania list należy pamiętać o stosowaniu poprawnej interpunkcji. Poniżej dwa przykłady wyliczeń:

- Nieuporządkowana poziom pierwszy,
  - nieuporządkowana poziom drugi,
    - nieuporządkowana poziom trzeci.

1. Uporządkowana poziom pierwszy,
  - a) uporządkowana poziom drugi,
    - i. uporządkowana poziom trzeci.

### 2.1.6 Liczby

Liczby i miana należy sformatować zgodnie z zasadami typografii obowiązującymi w języku polskim oraz zgodnie z normami SI. Częstym błędem jest podawanie wartości liczbowych w taki sposób w jaki wyświetlają je symulatory, np. 3.3V zamiast 3,3 V. Należy zwrócić uwagę na stosowanie właściwego znaku oddzielającego część całkowitą od ułamkowej oraz na oddzielanie miana od samej liczby (najprostszym sposobem jest wstawienie pomiędzy liczbę i symbol tzw. niełamliwej spacji czyli tyldy).

Problem formatowania liczb można rozwiązać w bardziej ogólny sposób używając pakietu `siunitx`. Pakiet ten zawiera szereg poleceń, od prostych służących do formatowania liczb:  $123 \times 10^{-5}$ ,  $100^\circ$ ,  $1050^\circ\text{C}$ ,  $1\ \mu\text{m}$  aż po bardziej złożone służące do justowania liczb w tabelach.

Tabela 2.1. Przykład wykorzystania pakietu `siunitx` do justowania zawartości pól tabeli.

Foo	Normal	Bold
foo1	111	111
foo2	222.2	222.2
foo3	3.33	3.33
foo4	4	4
foo5	5.5	5.5

### 2.1.7 Adresy internetowe

Adresy internetowe formatujemy za pomocą polecenia `\url{adres}`, np. `http://www.tug.org/`.

## 2.2 Przypisy

Jeśli trzeba utworzyć przypis to w  $\LaTeX$ -u<sup>1</sup> korzystamy z instrukcji `\footnote{przypis}`.

---

<sup>1</sup> $\LaTeX$  - (od [Leslie] Lamport  $\TeX$ ) oprogramowanie do zautomatyzowanego składu tekstu.



## 2.3 Odsyłacze

W tekście pracy dyplomowej często umieszcza się odniesienia do ilustracji, tabel, wzorów lub podrozdziałów.  $\LaTeX$  sam zajmuje się generacją numeracji tych elementów. Tak więc na etapie tworzenia tekstu konkretne indeksy nie są jeszcze znane. Dlatego w tekście wstawia się odsyłacze do etykiet a same etykiety umieszcza w definicjach elementów, do których zamierzamy się odnosić.

Etykietę tworzy się za pomocą polecenia `\label{etykieta}`. Warto przemyśleć system tworzenia nazw etykiet tak, aby łatwo kojarzyły się z typem obiektu. Przykładowo, etykieta ilustracji zawiera prefiks *fig*: (np. `\label{fig:fdota}`) a etykieta wzoru prefiks *Eq*: (np. `\label{Eq:ulamek}`).

Odsyłacz powstaje poprzez umieszczenie polecenia `\ref{etykieta}` (np. `\ref{fig:fdota}`). A oto przykłady zastosowania odsyłacza do rysunku: Rys. 2.1, wzoru 2.8 i rozdziału 2.7.

## 2.4 Tabele

Tabele składamy korzystając z otoczenia `tabular`. Podstawowe zasady składania tabel są proste i ich zastosowanie w praktyce nie powinno sprawiać trudności. Jednak w przypadkach bardziej złożonych struktur (np. łączenie pól, kolumn, wierszy) czy też tabel wykraczających poza jedną stronę może się okazać, że konieczne będzie zastosowanie specjalnych pakietów  $\LaTeX$ -a. Przedstawienie zaawansowanych metod składania tabel wykracza poza zakres tego dokumentu. Zainteresowanych odsyłam do dokumentacji. Zawsze warto skorzystać ze wsparcia jakiego w tym zakresie udzielają wyspecjalizowane narzędzia na przykład wspomniany już w rozdziale ?? program *TeXstudio*.

Tabela 2.2. Przykładowa tabela – zawartość pól wyśrodkowana.

	Nagłówek 1	Nagłówek 2	Nagłówek 3	Nagłówek 4
Etykieta wiersza	bb bb bbbb bbbb	cc	dd	ee
Etykieta wiersza	bb	cc	dd	ee
Etykieta wiersza	bb	cc	dd	ee
Etykieta wiersza	bb	cc	dd	ee ee eeee eeeee

Tabela 2.3. Przykładowa tabela – wyrównywanie zawartości pól do lewego lub prawego marginesu.

Nagłówek 1	Nagłówek 2	Nagłówek 3
bb bb bbbb bbbb	dd	ee
cc	dd	ee ee eeee eeeee

Tabela 2.4. Przykład tabeli z diagonalnym podziałem komórki.

Etykieta kolumn \ Etykieta wierszy	Nagłówek 1	Nagłówek 2
Nagłówek 1	Nagłówek 2	Nagłówek 3
AAAAA AAAAA	BBBB BBB	CCCC CCCC
AAAAAA	2	3

Tabela 2.5. Wymuszenie konkretnych szerokości kolumn.

	Nagłówek 1	Nagłówek 2	Nagłówek 3	Nagłówek 4
Etykieta wiersza	bb	cc	dd	Podstawowe zasady składania tabel są proste i ich zastosowanie w praktyce nie powinno sprawiać trudności
Etykieta wiersza	bb	cc cc cccc cccccc	dd	ee

## 2.5 Cytowanie kodu źródłowego

Cytując kod programu czy też modelu HDL najlepiej jest skorzystać z pakietu `listings`, który definiuje otoczenie `\begin{lstlisting}... \end{lstlisting}`. Pakiet ten pozwala na automatyczne wyróżnianie słów kluczowych szeregu języków programowania i opisu sprzętu. Poniżej pokazano przykłady: modelu VHDL (wydruk 2.1), modelu Verilog (wydruk 2.2) oraz programu w języku C++ (wydruk 2.3).

Wydruk 2.1. Przykładowy model VHDL

```

1  entity latch is
  port (sample: in std_logic;
        count : in  std_logic_vector (0 to 7);
        data:   out std_logic_vector (0 to 7));
5  end latch;

  architecture behav of latch is      — zatrask
  begin
    process (sample, count)
10  begin
      if (sample = '1') then
        data <= count;
      end if;
    end process;
15  end behav;

```

Wydruk 2.2. Przykładowy model Verilog

```

1  module add_carry_signed_final (
  input signed [2:0] A,
  input signed [2:0] B,
  input carry_in ,
5  output signed [3:0] Sum);

  assign Sum = A + B + $signed({1'b0, carry_in});

endmodule

```

Wydruk 2.3. Przykładowy program C++

```

1  // 'Hello World!' program

#include <iostream>

5  int main()
  {
    std::cout << "Hello World!" << std::endl;
    return 0;
  }

```

## 2.6 Ilustracje

Ilustracje są pobierane z plików, przy czym preferowany jest format PDF lub PNG.

### 2.6.1 Przygotowanie ilustracji

O ile to możliwe ilustracje powinny być przygotowane w postaci wektorowej, przy pomocy niezależnego oprogramowania i zapisane w plikach o formacie PDF lub SVG (ten drugi wariant wymaga lepszej znajomości  $\text{\LaTeX}$ -a). Zaawansowani użytkownicy mogą wykorzystać pakiety TikZ i PGF (<http://sourceforge.net/projects/pgf/>).

Jeśli ilustracja dostępna jest tylko w wersji rastrowej (np. plik graficzny zawierający przebiegi uzyskane w trakcie symulacji) to należy użyć formatu pliku o bezstratnej kompresji a zastosowana rozdzielczość nie może być mniejsza niż 300 dpi a najlepiej 600 dpi.

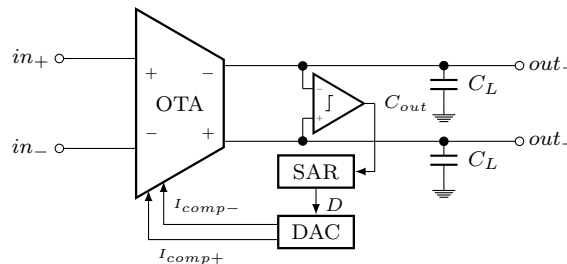
*Uwaga! Tzw. rzuty ekranowe mają rozdzielczość taką jak obraz na ekranie monitora czyli zwykle mniej niż 100 dpi i nie prezentują się dobrze na wydrukach.*

Przygotowując ilustracje trzeba pamiętać, że zastosowany w nich font i stopień pisma musi być zgodny z wyglądem tekstu. Stosujemy ten sam font co w tekście (lub jeśli to niemożliwe, inny o podobnym kroju) a wielkość napisów powinna być nieco mniejsza niż w tekście (jeśli stopień pisma tekstu wynosi 11 pt to ilustracje mogą stosować np. 9 pt). Można też zastosować w ilustracjach font bezszeryfowy (takich jak np. Arial). Niezależnie od tego jaką konwencję przyjmujemy należy *stosować ją konsekwentnie do wszystkich ilustracji*.

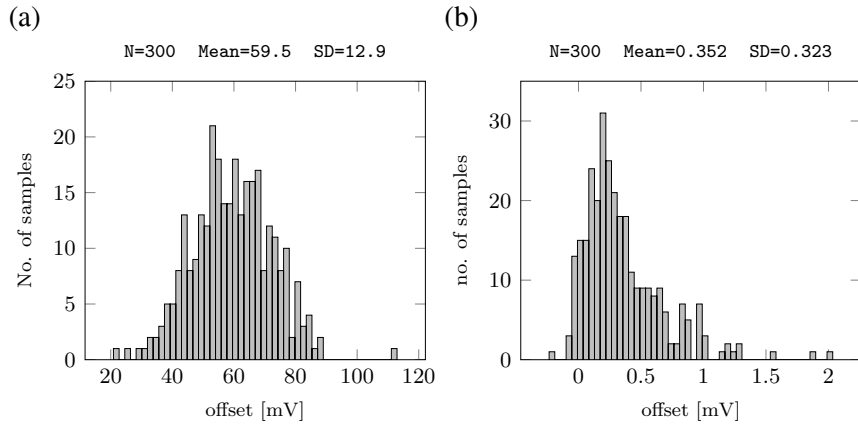
Kolejna kwestia to fizyczny rozmiar ilustracji. Przygotowywana grafika powinna mieć takie fizyczne wymiary, aby w trakcie włączania jej do dokumentu nie dokonywać skalowania. Skalowanie bardzo źle wpływa na czytelność grafiki rastrowej. Ponadto skalowanie zmienia grubość linii i wielkość napisów. Jeśli nie zastosuje się jednakowych zasad to może się okazać, że ilustracje nie prezentują się w jednolity sposób co sprawia fatalne wrażenie.

### 2.6.2 Składanie ilustracji

Ilustracje mogą być zamieszczane pojedynczo (Rys. 2.1) lub, jak pokazano na Rys. 2.2, obok siebie korzystając z otoczenia tabular (Rys. 2.2). Można też zastosować polecenie `\subfloat`.



Rys. 2.1. Schemat blokowy FDOTA z cyfrową kalibracją napięcia niezrównoważenia.



Rys. 2.2. Rozrzut napięcia niezrównoważenia FDOTA: (a) bez kalibracji, (b) z 9-bitową kalibracją.

## 2.7 Składanie wzorów

Jedną z wielu zalet  $\text{\LaTeX}$ -a, jak przystało na system składu publikacji naukowych, jest tryb matematyczny. W tym podrozdziale pokazano szereg przykładów użycia podstawowych konstrukcji tego trybu.

### 2.7.1 Formuły w osobnych wierszach

Podstawowym sposobem składania wzorów jest umieszczanie ich w osobnym wierszu. Wzory powinny być numerowane, tak jak to widać w poniższy przykładzie:

$$y = \frac{x - 1}{x + 2} \quad (2.1)$$

Automatyczną numerację uzyskujemy stosując otoczenie `\begin{equation}...\end{equation}`. Umieszczenie etykiety umożliwi łatwe odwoływanie się do wzoru w tekście, np. do formuły 2.8.

### 2.7.2 Formuły osadzone w tekście

W najprostszym przypadku wzory mogą być integralną częścią akapitu, np.  $E = mc^2$ . Trzeba jednak pamiętać, że takie rozwiązanie stosujemy tylko do bardzo prostych (pomocniczych) zależności. Taka formuła jest trudna do odnalezienia, nie można uzyskać do niej odsyłacza a jeśli jest bardziej rozbudowana "rozpycha" tekst.

Szczególnie niekorzystnie prezentują się osadzone w tekście ułamki ze względu na zmniejszony stopień pisma, np.  $y = \frac{x-1}{x+2}$ . Jeśli zdecydujemy się temu zaradzić stosując polecenie `dfrac` zamiast `frac`, np.:  $y = \frac{x - 1}{x + 2}$ , to znacząco zaburzona zostanie wielkość odstępów międzywierszowych, przez co tekst prezentuje się w sposób mało profesjonalny.

### 2.7.3 Ułamki z ukośnikiem

Proste ułamki lub miana można zapisać z wykorzystaniem ukośnika, np.:  $1/3$ ,  $1/n$ ,  $V/A$ .

### 2.7.4 Indeksy górne i dolne

Często spotykane są wyrażenia, w których stosowane są indeksy górne i dolne, np.:

$$a_1, a_{i_1}, a^2, a^{b^c}, a^{i_1}, a_i + 1, a_{i+1}, a_2^1, a_1^2 \quad (2.2)$$

$$x_1 + x_2 + \dots + x_n \quad (2.3)$$

### 2.7.5 Nawiasy i ograniczniki

Szereg symboli takich jak nawiasy występuje w kilku odmianach różniących się wielkością. np.:

$$(\dots(\dots(\dots(\dots(\dots$$

Zastosowanie "dużych" symboli poprawia czytelność złożonych formuł, np.:

$$F(x)|_a^b \dots F(x)|_a^b \dots F(x)|_a^b$$
$$[\sum_i a_i]^{1/p} \dots \left[\sum_i a_i\right]^{1/p} \dots \left[\sum_i a_i\right]^{1/p} \dots \left[\sum_i a_i\right]^{1/p}$$

### 2.8 Ułamki piętrowe

Zastosowanie zwykłego polecenia `frac` nie daje dobrych rezultatów:

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4}}}} \quad (2.4)$$

W takiej sytuacji należy zastosować instrukcję `cfrac`:

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4}}}} \quad (2.5)$$

### 2.9 Justowanie wzorów

Jeśli w tekście występuje kolejno kilka formuł to powinny one być wyjustowane względem siebie. Efekt ten uzyskuje się za pomocą otoczenia `\begin{align}...\end{align}`. W formułach umieszczany jest symbol "&", którego położenie wyznacza punkt odniesienia w każdej z nich. Poniżej kilka przykładów wyjustowanych formuł.

$$r^2 = s^2 + t^2, \quad (2.6)$$

$$2u + 1 = v + w^\alpha, \quad (2.7)$$

$$x = \frac{y + z}{\sqrt{s + 2u}}; \quad (2.8)$$

$$h(x) = \int \left( \frac{f(x) + g(x)}{1 + f^2(x)} + \frac{1 + f(x)g(x)}{\sqrt{1 - \sin x}} \right) dx =$$
$$= \int \frac{1 + f(x)}{1 + g(x)} dx - 2 \tan^{-1}(x - 2) \quad (2.9)$$

W niektórych przypadkach samo justowanie nie wystarczy, np. podczas składania wzoru funkcji określonej przedziałami. W takim przypadku korzystamy z otoczenia `\begin{cases}...\end{cases}`.

$$f(x) = \begin{cases} -x^2, & \text{if } x < 0; \\ \alpha + x, & \text{if } 0 \leq x \leq 1; \\ x^2, & \text{otherwise.} \end{cases} \quad (2.10)$$

## 2.10 Ingerowanie w skład dokumentu

L<sup>A</sup>T<sub>E</sub>X stara się rozmieścić tekst i inne elementy takie jak tabele, ilustracje czy wzory tak, aby uzyskać najlepszy efekt. Zdarza się jednak, np. kiedy blisko siebie występuje duża liczba rysunków, że trzeba zaingerować w układ pracy.

### 2.10.1 Kotwiczenie rysunków i tabel

Jedną z najczęściej spotkanych sytuacji jest wymuszenie konkretnej pozycji ilustracji lub tabeli. W tym celu należy definicję obiektu, czyli otoczenie `table` lub `figure`, uzupełnić o opcje wskazujące pożądaną lokalizację obiektu: `\begin{table}[!h]`, `\begin{figure}[!h]`. Litera *h* jest skrótem od *here*. Wykrzyknik nakazuje zaniechanie stosowania standardowych reguł rozmieszczania rysunków i tabel.

### 2.10.2 Wymuszanie złamania strony

Można wymusić złamanie strony w konkretnym miejscu wstawiając polecenie `\pagebreak[n]` (gdzie opcjonalny parametr  $n=1\dots4$ ), które L<sup>A</sup>T<sub>E</sub>X interpretuje jako życzenie by zakończyć w tym miejscu stronę, tym silniejsze im większa jest wartość  $n$ .

### 2.10.3 Wymuszanie złamania wiersza

W celu wymuszenia przejścia do nowego wiersza *bez rozpoczynania nowego akapitu* można zastosować polecenie `\\` lub `\newline`.

### 2.10.4 Zapobieganie wstawianiu wcięcia akapitowego

Czasem (np. w sytuacji cytowania w podrozdziale ??) chcemy wstawić wiersz odstępu i jednocześnie uniknąć rozpoczęcia nowego akapitu. Wtedy na początku bloku tekstu wstawiamy polecenie `\noindent`.

### 2.10.5 Zapobieganiu przenoszeniu do nowego wiersza

L<sup>A</sup>T<sub>E</sub>X składając akapity stosuje automatyczne dzielenie i przenoszenie wyrazów, zgodnie z regułami obowiązującymi w języku, w jakim tworzony jest dokument. Zdarza się niekiedy, że zostaje rozdzielony w ten sposób tekst, którego nie chcemy dzielić, np.: adres internetowy, stała fizyczna, itp. Można zapobiec dzieleniu słów i przenoszeniu do nowego wiersza umieszczając tekst jako argument polecenia `\mbox{...}`. Poniżej przykład takiej sytuacji.

*TeXstudio* jest dostępne nieodpłatnie na wszystkie platformy systemowe na <http://texstudio.sourceforge.net/>.

*TeXstudio* jest dostępne nieodpłatnie na wszystkie platformy systemowe na <http://texstudio.sourceforge.net/>.

### 2.10.6 Znaki specjalne

Zdarza się że, musimy użyć słowa, które zawiera znak specjalny L<sup>A</sup>T<sub>E</sub>X-a. Przykładem takiej sytuacji jest cytowanie nazwy pliku, która zawiera znak podkreślenia "\_", np. w nazwie pliku *praca\_dyplomowa.tex*. Wtedy należy taki znak poprzedzić odwróconym ukośnikiem czy znakiem `\`.

## 2.11 Spis literatury

Spis literatury generowany jest automatycznie, na podstawie umieszczonych w tekście odsyłaczy do rekordów bazy bibliograficznej.

### 2.11.1 Baza bibliograficzna w formacie BibTeX

Baza bibliograficzna w formacie BibTeX jest w istocie plikiem tekstowym (o nazwie zwyczajowo zakończonej przez rozszerzenie .bib) zawierającym rekordy opisujące poszczególne pozycje. Praktycznie wszystkie wydawnictwa dostarczają referencji w formacie BibTeX, tak więc skompletowanie odpowiednio sformatowanej bazy bibliograficznej nie będzie narażać na żadne trudności. Każdy rekord w bazie musi zawierać identyfikujący go unikalny klucz (etykietę). W podanym poniżej przykładzie są to Pastre2006, Pastre2009, Pelgrom1989 i wikibook.

```
@BOOK{Pastre2006,
  author = {Marc Pastre and Maher Kayal},
  title = {Methodology for the Digital Calibration of Analog Circuits and Systems: with
    Case Studies (The Springer International Series in Engineering and Computer Science)},
  year = {2006},
  publisher = {Springer},
  isbn = {1402042523},
  url = {http://www.amazon.com/Methodology-Digital-Calibration-Circuits-Systems/dp/}
}
@INPROCEEDINGS{Pastre2009,
  author = {Pastre, M. and Kayal, M.},
  title = {Methodology for the digital calibration of analog circuits and systems
    using sub-binary radix DACs},
  booktitle = {Mixed Design of Integrated Circuits Systems, 2009. MIXDES '09. MIXDES-16th
    International Conference},
  year = {2009},
  pages = {456-461},
  keywords = {analogue integrated circuits;calibration}
}
@ARTICLE{Pelgrom1989,
  author = {Pelgrom, M. J M and Duinmaijer, Aad C J and Welbers, A.P.G.},
  title = {Matching properties of MOS transistors},
  year = {1989},
  volume = {24},
  number = {5},
  pages = {1433-1439},
  issn = {0018-9200},
  journal = {Solid-State Circuits, IEEE Journal of},
  keywords = {MOS integrated circuits;insulated gate field effect transistors}
}
@WWW{wikibook,
  author = {Wikibooks},
  title = {LaTeX -- Wikibooks{,} The Free Textbook Project},
  year = {2012},
  url = {http://en.wikibooks.org/wiki/LaTeX}
}
```

Bazę bibliograficzną można przygotować przy pomocy dowolnego edytora tekstowego lub specjalnego programu, np. *JabRef* (<http://www.jabref.org/>).

### 2.11.2 Odsyłacze

W celu umieszczenia w tekście odsyłacza do konkretnej pozycji (lub grupy pozycji) należy użyć polecenia `\cite{klucz1,klucz2,...}`. Przykładowo, umieszczenie w jednym miejscu odsyłaczy do dwóch pozycji wygląda następująco: `\cite{Pastre2006,Pelgrom1989}`, co po przetworzeniu da następujący efekt: [4, 5]. W przypadku sekwencji trzech odsyłaczy do kolejnych pozycji

`\cite{Pastre2006,Pastre2009,Pelgrom1989}` efektem będzie [3–5]. W tym drugim przypadku wykryta została sekwencja odsyłaczy do kolejnych pozycji literatury i automatycznie zastąpiono ją przedziałem indeksów.

### 2.11.3 Generowanie spisu literatury

Wygenerowanie spisu literatury wymaga kilku przebiegów kompilatora (np. `pdflatex`, `lualatex`). Wynika to z faktu, że w trakcie przetwarzania tekstu nie są jeszcze znane wszystkie odwołania do pozycji bibliografii.

Po każdym przebiegu kompilatora tworzony jest pomocniczy plik (o rozszerzeniu `.aux`) zawierający klucze cytowanych pozycji. Dopiero na tej podstawie specjalny program `bibtex` może wygenerować zestawienie cytowanych pozycji bibliografii (plik z rozszerzeniem `.bbl`). To zestawienie jest wykorzystywane w kolejnym przebiegu kompilatora do wygenerowania rozdziału *Bibliografia* w miejscu, w którym tekst zawiera polecenie `\bibliography{nazwa_bazy_bibliograficznej}` oraz zastąpienia odsyłaczy identyfikatorami. Postać identyfikatorów zależy od przyjętego stylu formatowania bibliografii. W tym dokumencie zastosowany najprostszy styl czyli oznaczenie kolejnych pozycji spisu literatury kolejnymi liczbami (polecenie `\bibliographystyle{plainurl}`).

Taka procedura całkowicie uwalnia użytkownika od konieczności samodzielnego wstawiania indeksów oraz bardzo kłopotliwego i podatnego na błędy korygowania tych indeksów po każdej modyfikacji bazy bibliograficznej.

Znacznym ułatwieniem w tej sytuacji jest zastosowanie "inteligentnego" środowiska (np. *TeXStudio*), które samodzielnie wywołuje w poprawnej kolejności wszystkie niezbędne programy w celu wygenerowania aktualnego zestawienia bibliografii.

Jeśli zajdzie potrzeba wymuszenia ponownej generacji bibliografii należy usunąć plik `.bbl` oraz zaktualizować datę modyfikacji bazy bibliograficznej a następnie uruchomić przetwarzanie dokumentu.



# Bibliografia

- [1] TeX Users Group web site. URL: <http://www.tug.org/>.
- [2] PN-83/P-55366 Zasady składania tekstów w języku polskim, 1983.
- [3] M. Pastre and M. Kayal. Methodology for the digital calibration of analog circuits and systems using sub-binary radix dacs. In *Mixed Design of Integrated Circuits Systems, 2009. MIXDES '09. MIXDES-16th International Conference*, pages 456–461, 2009.
- [4] Marc Pastre and Maher Kayal. *Methodology for the Digital Calibration of Analog Circuits and Systems: with Case Studies (The Springer International Series in Engineering and Computer Science)*. Springer, 2006. URL: <http://www.amazon.com/Methodology-Digital-Calibration-Circuits-Systems/dp/1402042523%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D1402042523>.
- [5] M. J M Pelgrom, Aad C J Duinmaijer, and A.P.G. Welbers. Matching properties of mos transistors. *Solid-State Circuits, IEEE Journal of*, 24(5):1433–1439, 1989. doi:10.1109/JSSC.1989.572629.
- [6] Wikibooks. Latex – wikibooks, the free textbook project, 2012. URL: <http://en.wikibooks.org/wiki/LaTeX>.
- [7] Wikipedia. Latex – wikipedia, the free encyclopedia, 2012. URL: <http://en.wikipedia.org/wiki/LaTeX>.



# Wykaz symboli i skrótów

$V_t$	. . . . .	napięcie progowe
$g_m$	. . . . .	transkonduktancja
$C_{ox}$	. . . . .	pojemność jednostkowa bramki
$\mu$	. . . . .	ruchliwość nośników
$W$	. . . . .	szerokość kanału tranzystora
$L$	. . . . .	długość kanału tranzystora
CMOS	. . . .	Complementary MOS
VHDL	. . . .	Very High Speed Integrated Circuits Hardware Description Language
FSM	. . . . .	Finite State Machine
VLSI	. . . . .	Very Large Scale Integration



# Spis rysunków

2.1	Schemat blokowy FDOTA z cyfrową kalibracją napięcia niezrównoważenia. . . . .	11
2.2	Rozrzut napięcia niezrównoważenia FDOTA: (a) bez kalibracji, (b) z 9-bitową kalibracją. . . .	12



# Spis tabel

2.1	Przykład wykorzystania pakietu <code>siunitx</code> do justowania zawartości pól tabeli. . . . .	8
2.2	Przykładowa tabela – zawartość pól wyśrodkowana. . . . .	9
2.3	Przykładowa tabela – wyrównywanie zawartości pól do lewego lub prawego marginesu. . . . .	9
2.4	Przykład tabeli z diagonalnym podziałem komórki. . . . .	9
2.5	Wymuszenie konkretnych szerokości kolumn. . . . .	10





## **Załączniki**



# Spis załączników

<b>A</b>	<b>Tekst źródłowy programu</b>	<b>29</b>
<b>B</b>	<b>Model VHDL projektu</b>	<b>31</b>



# A Tekst źródłowy programu

```
1  library ieee;
   use ieee.std_logic_1164.all;
   use ieee.std_logic_unsigned.all;
   use ieee.std_logic_arith.all;
5
   entity wave is
   port (
       clk, reset : in std_logic;
       t_l, t_h   : in std_logic_vector(31 downto 0);
10  w           : out std_logic);
   end wave;

   architecture rtl of wave is
   type FSM is (IDLE, PH, PL);
15  signal state, next_state: FSM;
   signal timer : std_logic_vector(31 downto 0);
   signal resetc: std_logic;

   begin
20  RS_PROC:
   process (clk, reset)
   begin
       if (reset='0') then
           state <= IDLE;
25  elseif (rising_edge(clk)) then
           state <= next_state;
       end if;
   end process;

30  NS_PROC:
   process (state, t_l, t_h, timer)
   begin
       case state is
35  when idle =>
           if (t_l = 0 or t_h = 0) then
               next_state <= idle;
           else
               next_state <= PH;
40  end if;

           when PH =>
           if (timer < t_h - 1) then
               next_state <= PH;
45  else
               next_state <= PL;
           end if;

           when PL =>
50  if (timer < t_l - 1) then
               next_state <= PL;
           else
               next_state <= PH;
           end if;
55  end case;
   end process;

   DW_PROC:
```

```
60  process (state)
    begin
        if (state = PH) then w <= '1';
        elsif (state = PL) then w <= '0';
        end if;
65  end process;
    end if;
    end process;
    end rtl;
```

# B Model VHDL projektu

```
1  library ieee;
   use ieee.std_logic_1164.all;
   use ieee.std_logic_unsigned.all;
   use ieee.std_logic_arith.all;
5
   entity shift_reg is
     generic(N: integer range 0 to 32 := 8);
     port (
10    clk, reset, load : in  std_logic;
       pos, reg_in      : in  std_logic_vector(N-1 downto 0);
       reg_out          : out std_logic_vector(N-1 downto 0));
     end shift_reg;

   architecture rtl of shift_reg is
15    signal rejestr: std_logic_vector(N-1 downto 0);

     begin
       process (clk, reset, load, reg_in)
         begin
20           if (reset = '0') then
               rejestr <= (others => '0');
             elsif (load = '0') then
               rejestr <= reg_in;
             elsif (rising_edge(clk)) then
25               rejestr (N-1 downto conv_integer(pos)) <= rejestr(N-1-conv_integer(pos) downto 0);
               rejestr (conv_integer(pos) downto 0) <= (others => '0');
             end if;
           end process;

30    reg_out <= rejestr;
     end rtl;
```