

**WYDZIAŁ INFORMATYKI, ELEKTRONIKI I TELEKOMUNIKACJI**

KATEDRA ELEKTRONIKI

**PRACA DYPLOMOWA MAGISTERSKA**

Scalony system sterowania windą

ASIC for lift control

Autor: Mateusz Dyrdół

Kierunek studiów: Elektronika i Telekomunikacja

Opiekun pracy: prof. dr hab. inż. Andrzej Kos

Kraków, 2019

**OŚWIADCZENIE**

Uprzedzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz.631 z późn. zm.):„ Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.”, a także uprzedzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.) „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchybiające godności studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej „sądem koleżeńskim”, oświadczam, że niniejszą pracę dyplomową wykonałem osobiście i samodzielnie i że nie korzystałem ze źródeł innych niż wymienione w pracy.

……………………………….*podpis dyplomanta*

**Spis treści**

[Wstęp 5](#_Toc11943353)

[Cel pracy 6](#_Toc11943354)

[Rozdział 1 Technologia dźwigów osobowych 7](#_Toc11943355)

[1.1 Pierwsze systemy dźwigowe 7](#_Toc11943356)

[1.2 Strategie sterowania windą 8](#_Toc11943357)

[1.2.1 Strategia zbiorowej kontroli 8](#_Toc11943358)

[1.2.2 Strategia strefowa 10](#_Toc11943359)

[1.2.3 Strategia oparta na wyszukiwaniu 12](#_Toc11943360)

[1.3 Cechy standardowej windy 13](#_Toc11943361)

[1.3.1 Zależności czasowe 15](#_Toc11943362)

[1.3.2 Interfejs użytkownika w standardowej windzie 16](#_Toc11943363)

[1.4 Układy elektroniczne sterujące windą 17](#_Toc11943364)

[1.5 Przepisy prawne i normy 18](#_Toc11943365)

[1.6 System przyszłości 20](#_Toc11943366)

[1.6.1 Algorytm wysyłki docelowej 20](#_Toc11943367)

[1.6.2 Kierunki rozwoju technologii 23](#_Toc11943368)

[Rozdział 2 Opis algorytmu w języku Verilog 25](#_Toc11943369)

[2.1 Tworzenie modułów z wykorzystaniem języka Verilog 25](#_Toc11943370)

[2.2 Rozwój oprogramowania w oparciu o cechy użytkowe 26](#_Toc11943371)

[2.3 Architektura systemu 28](#_Toc11943372)

[2.3.1 Moduł pełnego piętra 28](#_Toc11943373)

[2.3.2 Moduł piętra przejściowego 28](#_Toc11943374)

[2.3.3 Interfejs użytkownika 29](#_Toc11943375)

[2.3.4 Parametry czasowe i bezpieczeństwo 30](#_Toc11943376)

[2.4 Metody weryfikacji kodu 30](#_Toc11943377)

[2.4.1 Przypadki testowe 31](#_Toc11943378)

[2.4.2 Moduł testowy 31](#_Toc11943379)

[2.4.3 Wnioski/podsumowanie 31](#_Toc11943380)

[2.5 Modułowosc i rozbudowa kodu??? If not earlier 32](#_Toc11943381)

[Rozdział 3 Projektowanie układu scalonego 33](#_Toc11943382)

[3.1 Generacja 33](#_Toc11943383)

[3.2 Synteza 34](#_Toc11943384)

[3.3 Tworzenie layoutu 35](#_Toc11943385)

[3.4 Bezpośrednia synteza cyfrowa 35](#_Toc11943386)

[3.5 Generacja sygnałów 35](#_Toc11943387)

[3.5.1 Zegar referencyjny 35](#_Toc11943388)

[3.5.2 Akumulator fazy 35](#_Toc11943389)

[3.5.3 Tablica z próbkami 37](#_Toc11943390)

[3.5.4 Mikser 37](#_Toc11943391)

[3.6 Interfejs użytkownika 38](#_Toc11943392)

[3.7 Dźwięk w programie Audacity 38](#_Toc11943393)

[Podsumowanie i wnioski 40](#_Toc11943394)

[Streszczenie 41](#_Toc11943395)

[Summary 42](#_Toc11943396)

[Spis stosowanych skrótów 43](#_Toc11943397)

[Słowa kluczowe 44](#_Toc11943398)

[Keywords 45](#_Toc11943399)

[Bibliografia 46](#_Toc11943400)

[Dodatek A. Schemat połączenia MIDI z portem szeregowym UART 48](#_Toc11943401)

[Dodatek B. Spis zawartości dołączonej płyty CD 49](#_Toc11943402)

[Spis ilustracji 50](#_Toc11943403)

[Spis tabel 51](#_Toc11943404)

# Wstęp

W tej nowoczesnej erze windy stały się integralną częścią dowolny kompleks komercyjny lub publiczny. Ułatwia to szybciej ruch osób i bagażu między piętrami. Winda system sterowania jest jednym z najważniejszych aspektów moduł sterujący elektroniki, który jest używany w motoryzacji podanie. Zwykle windy są zaprojektowane dla określonych celów budynek z uwzględnieniem głównych czynników, takich jakwysokość budynku, liczba osób podróżujących do każdego piętro i przewidywane okresy wysokiego zużycia. Winda System został zaprojektowany z różnymi strategiami sterowania. To implementacja oparta jest na FPGA, który może być wykorzystany dobudynek o dowolnej liczbie pięter, z określonymi wejściamii wyjścia. Ten kontroler można zaimplementować dlawinda o wymaganej liczbie pięter po prostuzmiana zmiennej sterującej w kodzie HDL. Takie podejście jestw oparciu o algorytm, który zmniejsza ilośćwymagane obliczenia, koncentrując się tylko na odpowiednich zasadachpoprawia to wydajność grupy windysystem. Sterownik windy został opracowany przy użyciu VerilogHDL

# Cel pracy

Praca magisterska ma na celu przedstawienie algorytmu sterowania windą i jego realizację jako układ scalony. Opis behawioralny układu cyfrowego został w języku opisu sprzętu Verilog. Sterownik bazuje na automacie skończonym, gdzie poszczególne stany są instrukcjami wykonywanymi przez windę.

Podstawowym kwestią do rozważenia w pierwszym etapie pracy jest wybranie optymalnego rozwiązania ruchu dźwigu. Główną inspiracją do opisania algorytmu jest obserwacja istniejących rozwiązań oraz usprawnienie ich działania. System musi być przyjazny dla użytkownika, sterowanie będzie odbywać się przyciskami wewnątrz i na zewnątrz windy. Kolejnym ważnym aspektem przy projektowaniu systemu sterowania winda jest energooszczędność. Szacuje się ze w niskich budynkach mieszkalnych winda jeździ sporadycznie - poniżej **5% czasu dobowego**[[1]](#footnote-1). Sterownik

# Technologia dźwigów osobowych

## Pierwsze systemy dźwigowe

Prymitywne windy były używane już w III wieku pne i były obsługiwane przez ludzi, zwierzęta lub koła wodne. W 1743 r. zbudowano dla króla Ludwika XV przeciwwagę, napędzaną przez człowieka, osobistą windę, łączącą jego mieszkanie w Wersalu z mieszkaniem jego kochanki, Madame de Chateauroux, której kwatera znajdowała się piętro wyżej od króla Ludwika. Pierwsze nowoczesne windy pasażerskie nie więcej niż 150 lat temu. Windy parowe i hydrauliczne zostały już wprowadzone przez 1852, kiedy Elisha Otis stworzył jedno z najważniejszych wynalazków wind, sprzęgło, które uniemożliwił upadek windy. Następnie w 1857 r. Zainstalowano pierwszą windę pasażerską w sklepie E. Haughwout & Company, Nowy Jork. Rozwój technologii windy był bardzo szybki z powodu potrzeb firm przemysłowych.

Pierwsze windy były obsługiwane przez proste urządzenia mechaniczne, takie jak sterowanie „liną ręczną” (Strakosch 1967). Pasażer mógł zadzwonić do windy, naciskając linę po obu stronach kabiny. Ponieważ szyby nie były całkowicie zamknięte, eksploatacja wind była dość niebezpieczna. Prymitywna forma sterowanie windą w jednym samochodzie opierało się na sterowanym elektrycznie przełączniku samochodu (Barney et al.1985).

Nowoczesne windy zostały opracowane w XIX wieku. Te surowe dźwigi powoli ewoluowały od napędzanej parą do mocy hydraulicznej. Pierwsze hydrauliczne dźwigi zostały zaprojektowane z wykorzystaniem ciśnienia wody jako źródła energii. Technologia silnika i metody sterowania ewoluowały szybko, a elektryczność szybko stała się akceptowanym źródłem energii. Pierwsza elektryczna winda została zbudowana przez niemieckiego wynalazcę Wenera Von Siemensa w 1880 roku. W dzisiejszych czasach nowoczesne budynki komercyjne zwykle mają wiele wind z ujednoliconym systemem sterowania. Ze względu na przeznaczenie windy dzielimy na:

* dźwigi osobowe napędzane hydraulicznie bądź elektrycznie, które służą do przenoszenia osób na różne poziomy budynków. Zwykle przemieszczają się one dość szybko i sztywno w pozycji pionowej,
* dźwigi osobowo-towarowe to z kolei urządzenie, które nadaje się zarówno do przewozu osób, jak i towaru,
* dźwigi szpitalne, zwykle przeznaczony jest do przewozu noszy, a także aparatury szpitalnej oraz pacjentów danego oddziału, czy też personelu medycznego,
* dźwigi budowlane przeznaczone do przewozu grupy robotników oraz materiałów budowlanych, które zwykle przewożone są na wysokie elewacje.

Niezależnie od przeznaczenia, najważniejszą kwestią jest sposób sterowania dźwigiem. Od systemu korbowego sterowanego przez dźwigowego aż po systemy sterowane dyspozycjami, windy stały się szybsze, bardziej zaawansowane i niezawodne w poruszaniu się między piętrami.

Współczesne systemy sterowania windą są głównie w oparciu o logikę przekaźnikową, układy PLC lub mikrokontrolery. Windę można uznać jako złożony system reaktywny, który wymaga równoległego przetwarzanie zdarzeń z wieloma wejściami i wyjściami.

## Strategie sterowania windą

Przez lata opracowano wiele strategii sterowania winda. Pierwsze z nich były to bardzo proste strategie a dziś powszechne są algorytmy wykorzystujące sztuczną inteligencję i uczenie maszynowe w celu poprawienia ruchów wind. W wysokich budynkach z dużym przepływem pasażerów, zwłaszcza w biurach, strategia dla system wind ma wielkie znaczenie. Głównym zadaniem jest zminimalizowanie czasu oczekiwania pasażerów oraz czas dostawy na piętro, poprawiając tym samym komfort i wydajność systemu. Istnieje wiele różnych wind w wielu odmianach, które nie są używane tylko do przewozu ludzi czy towarów. Są takie, które mogą być wywoływane i kontrolowane tylko przez operatora, przesuwające się poziomo lub w wielu kierunkach. Strategia powinna zoptymalizować ważne i unikalne dla budynku cechy, które są określone dla każdego z nich w projekcie. Powoduje to dużą liczbę strategii, której liczba rośnie wraz z rozwojem technologii.

### Strategia zbiorowej kontroli

Do jednej z najbardziej popularnych algorytmów należy strategia zbiorowej kontroli. Ten rodzaj systemu jest uznawany za standardowy algorytm kontroli windy. Polega na tym, że winda jedzie w jednym, ustalonym kierunku zabierając pasażerów jadących w tym kierunku. Kiedy po drodze nie ma więcej żądań w kierunku jazdy windy, następuje zmiana kierunku. W przeciwnym razie wózek zatrzymuje się i przechodzi w stan bezczynności, do czasu, gdy upuści ją ostatni pasażer [5]. Jedyną wadą tej strategii jest zjawisko zwane *grupowaniem,* gdzie kilka wózków odbiera to samo żądanie z piętra i przyjeżdża w podobnym czasie, zwiększając tym samym zarówno czas oczekiwania dla pozostałych pasażerów w systemie, jak i odległość podróży wózka windy.

Popularność tej strategii spowodowała użycie jej w innej technologii. Podobnie jest w sterowaniu ramieniem dysku twardego. Dostęp do danych na dysków odbywa się w ruchu uporządkowanym. Nadejście nowego żądania, gdy napęd jest w stanie bezczynności, zapoczątkowuje ruch ramienia w kierunku cylindra, w którym przechowywane są dane, zarówno w, jak i na zewnątrz. Dodatkowe polecenia są obsługiwane tylko w bieżącym kierunku ruchu ramienia, aż ramię osiągnie krawędź dysku. Kierunek ramienia odwraca się, a żądania pozostające w przeciwnym kierunku są obsługiwane. Cykl jest powtarzany, aż wszystkie aktywnych polecenia są wykonane. Jest to opis algorytmu SCAN nazywanego również *algorytmem windy.*

**

*Rysunek 1‑1 Ruch głowicy dysku twardego zgodny z algorytmem SCAN*

***Źródło: [7]***

Dysk twardy znacząco różni się od dźwigu osobowego. Ważne jest by zestaw tych instrukcji odnieść do działania projektowanego algorytmu. Najważniejszą cechą jest utrzymanie kierunku.

### Strategia strefowa

Pierwsze opisy strategii strefowej została opisana przez Georga Strakoscha i Roberta Caporale w pracy *The Vertical Transportation Handbook* z 1983 roku. Sterowanie strefą jest alternatywą dla sterowania zbiorowego i jest preferowane w budynkach z systemem wielu wind. W przypadku korzystania ze strategii strefowej budynek powinien zostać rozdzielony strefy o rozmiarach zależnych od ruchu każdej strefy. Każda winda ma swoje miejsce stopu i tylko odbiera pasażerów z pięter wewnątrz strefy. W przypadku strategii strefy winda ignoruje wszystkie żądania nawet poza jej strefą podczas podróży. Niektóre windy mogą mieć te same piętra w strefach w zależności od przepływu pasażerów i wykorzystywać do poprawy podstawowego założenia strategii [4]. Wózek zatrzymuje się w strefie, gdy jest bezczynna.

Ta strategia ma na celu utrzymanie samochodów w oddzieleniu i uniknięciu zjawiska grupowania. System ten jest odpowiedni dużego natężenia ruchu, gdy połączenia hali są rozłożone na wszystkie strony budynek, ale jednocześnie traci dużą elastyczność [6] od samochodów nie mogą się nawzajem pokrywać. Do stworzenia optymalnej strategii, należy wybrać rozkład stref ostrożnie. Podejmując decyzję o podziale budynku należy wziąć kilka zmiennych do rozważenia. Możesz go podzielić w zależności na populacji pięter lub jeśli jest ważne piętro, takie jak piętro kierownicze. Ogólnym pojęciem jest posiadanie tylu stref, ile jest dostępnych szybów windowych [7].



*Rysunek 1‑2 System strefowy w budynku Shanghai Tower*

***Źródło: [8]***

Istniejący budynkiem, w którym działa systemie strefowym jest największy wieżowiec w Chinach – Shanghai Tower. Budynek został podzielony na dziewięć stref, każdy z nich oznaczona innym kolorem na rysunku 1-2. Połączenia między strefami znajdują się w rożnych miejscach piętra. Ogromna ilość 128 pięter jest skomunikowana przez 108 wind. Dodatkowo windy poruszające się w tym budynku osiągają największe prędkości sięgające do 20.5m/s.



*Rysunek 1‑3 Przykład połączenia dwóch stref*

***Źródło: [8]***

System został zrealizowany przez firmę Mitsubishi. Dostęp do hotelu odbywa się przez piąty hol na wysokości 101/102 piętra. Lokalne strefy są obsługiwane przez windy jednopokładowe w całej wieży, a taras widokowy na szczycie wieży jest obsługiwany przez trzy windy wahadłowe o bardzo dużej prędkości, które poruszają się z prędkością 18 metrów na sekundę. Te trzy windy wahadłowe są uzupełnione trzema windami strażackimi, które znacznie zwiększają przepustowość gości na taras widokowy w szczytowych okresach użytkowania. W przypadku pożaru lub innej sytuacji awaryjnej, windy wahadłowe są zaprojektowane do ewakuacji pasażerów ze specjalny pięter schronienia, rozmieszczonych w regularnych odstępach na całej wysokości wieży [9].

### Strategia oparta na wyszukiwaniu

W przeciwieństwie do opisanych powyżej algorytmów, strategia oparta na wyszukiwaniu opiera się na wybraniu windy o najkrótszym czasie oczekiwania. Optymalizacja odbywa się w dwóch trybach zachłannym i nie-zachłannym. Różnica między tymi strategiami polega na tym, że zachłanne strategie wyszukiwania wykonują natychmiastowe przydzielanie połączeń, czyli przypisują połączenie z wózkiem windy po ich pierwszej rejestracji i nigdy nie rozważają ponownie tych zadań. Chciwe algorytmy rezygnują z pewnej miary wydajności ze względu na brak elastyczności, ale także wymagają mniej czasu obliczeniowego. Przeciwny algorytm jest elastyczny i może ponownie ocenić przydziały połączeń w świetle nowych informacji ciągłych z systemu windy. Nie-zachłanne algorytmy odkładają swoje zadania lub rozważają je w świetle zaktualizowanych informacje, które mogą otrzymać w dodatkowych połączeń lub ilości miejsc dla pasażerów. Ten typ algorytmu zajmie więcej czasu, aby zdecydować, które połączenie powinno zostać przypisane co skutkuje zwiększeniem średniego czasu oczekiwania, ale ogólny wynik może być ostatecznie lepszy [5].

Niezależnie od podejścia, system wybiera kabinę, która minimalizuje czas oczekiwania, czas podróży i liczbę pasażerów. System również wybiera współczynniki i szacowanie funkcji. Symulacje przed uruchomienie systemu służą do weryfikacji ich skuteczności. Po każdym zdarzeniu kontroler szuka najlepszego przypisania połączeń do wózka windy. Słabą stroną tego podejścia jest jego wymaganie obliczeniowe [6].

## Cechy standardowej windy

Każdy projektant windy jest głównie zainteresowany niecierpliwością pasażerów podczas oczekiwania oraz czasem podróży. Podczas gdy pasażerowie czekają na piętrze pośrednim, ich niecierpliwość rośnie. W środowisku komercyjnym z reguły pracownicy są mniej tolerancyjni w oczekiwaniu niż ludzie w środowisku mieszkalnym. Badania wskazują, że pasażerowie stają się niecierpliwym po odczekaniu około 30 sekund w budynku handlowym i około 60 sekund w budynku mieszkalnym [11].

Dobrą analogią jest porównanie transportu wertykalnego do przepływu ciągłego wody. System ciągłego przepływu transportuje wodę ze zbiornika strumieniem lub wężem, która jest przenoszony do miejsca przeznaczenia. System wsadowy przenosi zmierzone ilości do zbiornika, gdzie się gromadzą się, dopóki kolejna partia nie zostanie przeniesiona, zwykle w wiadrze. Można porównać windę do takiego przenośnika wsadowego. Przybycie ludzi do budynku jest w ciągłym przepływie, a system windy to przenośnik przenoszący ludzi ze zbiornika (lobby) do ich docelowych miejsc. Idealnym rozwiązaniem dla wind jest posiadanie wielu wind do przybliżenia procesu ciągłego przepływu, tak aby lobby (zbiornik) nigdy nie było wypełnione nadwyżką ilości osób, którą przewiezie jedna winda.

Z tych obserwacji można wywnioskować pierwsze wymagane dla dobrej obsługi windy: system musi zapewniać wystarczającą ilości usług windy dla maksymalna stawka przylotu lub odlotu pasażera oczekiwana w szczytowym okresie ruchu. To jest możliwe do osiągnięcia albo przez platformę o wystarczającej powierzchni, aby pomieścić wszystkie osoby czekając na przejażdżkę lub alternatywnie, wystarczającą liczbę mniejszych platform. Alternatywa większej ilości platform jest zazwyczaj preferowana, ze względów bezpieczeństwa. Drugim wymaganiem jest to by zaprojektować system tak, aby zapewnić średni czas oczekiwania poniżej 30 sekund dla budynku komercyjnego i mniej niż 60 sekund w budynkach mieszkalnych.

Kolejnym podobnym przykładem systemu zbliżonego do windy są schody ruchome. Platformy są dostępne przy minimalnym czasie oczekiwania, aby osoba miała natychmiastowy dostęp do schodów. Każda z platform jest wystarczająco duża, aby pomieścić tylko jedną lub dwie osoby, jeśli więcej osób czeka na platformę w tej samej chwili, ktoś musi czekać. Potencjalni pasażerowie nie zniecierpliwiają się, ponieważ widzą schody w ruchu, których czas oczekiwania jest bardzo krótki. W przypadku windy, osoba czekająca na wyższym piętrze może nie być w stanie sprawdzić, czy wózek windy jest w środku wykonywania żądania co może powodować niecierpliwość podczas oczekiwania.

Ponadto, gdy pasażerowie wchodzą na ruchome schodów, wiedzą, że zostaną zawiezieni na następne piętro w stosunkowo krótkim czasie i - poza ekstremalnymi przypadkami w niektórych stacjach metra - widzą koniec schodów. Pasażerowie windy często nie wiedzą, jak długo będą podróżować na wybrane piętro. Jeśli obsługiwane jest wiele pięter w ruchliwym budynku i liczba wind jest ograniczona, osoba może być w windzie przez długi okres czasu. Badania wykazały, że jazda około 100 sekund staje się granicą tolerancji, gdy winda robi kilka przystanków, na których wysiada jedna osoba. Tolerancja wydłuża się do około 150 sekund, jeśli kilka osób jest obsługiwanych na każdym przystanku; przeciętna osoba "czuje się bardziej spokojna”. Ostatecznie, jeżeli monotonia jest zakłócona przez zmieniającą się scenerię (np. widok z wieżowca), pasażer może tolerować jazdę nawet przez 180 sekund. Te czynniki czasowe są przybliżone, ponieważ tolerancja jednostki może różnic się w zależności od celu użycia windy.

### Zależności czasowe

Bazując na poprzednich można zauważyć jak bardzo są ważne parametry czasowe dla systemu windy. Do obliczeń całkowitej podróży windą, praktyczną procedurą jest rozbicie podróży na pojedyncze składniki. Podstawowy modelem jest winda dwuprzystankowa. Przypuśćmy, że istnieje budynek, gdzie ​​dwa kolejne przystanki są oddalone od siebie o 3 metry. Gdy pasażerowie przybywają na i obsługują przycisk wezwania windy, zaczynając. Kiedy opuszczają windę na drugim podeście, ich podróż jest zakończona. Po zarejestrowaniu połączenia winda obsługuje pasażera i godzinę czynniki będą następujące.



*Rysunek 1‑4 Model windy dwuprzystankowej*

***Źródło: [5]***

Odnosząc się do rysunku 1.4, jeśli wózek znajduje się na dolnym podeście, drzwi windy muszą się otworzyć, gdy pasażer naciśnie przycisk wywołania (a). Drzwi otwierają się około dwie sekundy, w zależności od szerokości i rodzaju drzwi. Kolejne dwie sekundy trwa wejście pasażera do środka i naciśnięcie przycisk (b). Drzwi zamykają się (około 3 sekundy), a samochód musi przejechać 3 metry do następnego piętra (około 7,5s)(c). Kolejno drzwi otwierają się, co zajmuje 2 sekundy, a kolejne 2 sekundy pasażer opuszcza windę (d). Całkowity czas spędzony przez tego pasażera wynosi około 18,5 sekund. Zanim inna osoba będzie mogła skorzystać z usługi, musi upłynąć więcej czasu. Drzwi musi zamknąć się ponownie (3s), a wózek windy musi wrócić na początkowe piętro (7,5s). W tym punkcie cykl się powtarza. Całkowity czas podróży w obie strony ma około 30 sekund. Tak więc 30 sekund to przybliżony czas osoby, która właśnie przeoczyła windę na pierwszym. Ten czas się nazywa „interwałem” między obsługą windy na parterze. Jeśli uznamy to jako proces ciągły ze strumieniem pasażerów poruszających się w jednym kierunku, przeciętny pasażer może oczekiwać średnio połowę czasu podróży windą w obie strony. Niektórym pasażerom uda się przybyć tuż przed przyjazdem windy i nie będą musieli czekać, podczas gdy inni mogą przeoczyć windę i ich czas oczekiwania będzie wynosił cały interwał, co skutkuje średnim czasem oczekiwania piętnastu sekund. Jeśli dwie windy są obok siebie, każda z nich obsługuje dwa przystanki i działa zgodnie z opisem (każdy z czasem podróży w obie strony 30 sekund), odstęp będzie wynosił połowę czasu podróży w obie strony, czyli 15 sekund, a średni czas oczekiwania wyniesie około 7,5 sekundy.

Dwustopniowa winda to najprostszy model systemu wind. Oczekiwane jest, że pasażer, który wsiądzie na jednym piętrze wysiądzie na drugim. Czas transferu jest zminimalizowany i nie ma mowy o prawdopodobnych zatrzymaniach, ponieważ jest tylko jeden możliwy przystanek. Planowanie jest proste: jeśli ruch jest dwukierunkowy, jeden wózek powinien znajdować się na górze, a drugi na dole; w ruchu jednokierunkowym windy powinny być skoncentrowane u góry lub u dołu. Obliczenia windy stają się bardziej złożone z trzema lub większa ilością przystanków.

### Interfejs użytkownika w standardowej windzie

W nowoczesnych budynkach ludzie są przyzwyczajeni do przycisku przywołania windy, następnie przyjazdu, po którym otwierają się drzwi, aby system wykonał swoje zadanie. Zazwyczaj pasażerowie zauważą zapaloną lampkę pokazującą kierunek podróży windy i wybierają szyb odpowiadający żądanemu kierunkowi. W bardziej ruchliwych budynkach istnieje duża możliwość zatrzymania dwóch wózków jednocześnie na piętrze. Gdy pasażerowie wsiądą do kabiny windy, oczekuje się, że wybiorą przycisk na docelowe piętro. Niezastosowanie się do tej reguły może spowodować, że winda nie pojedzie w wybrane miejsce. Odmianą powyższych reguł jest wprowadzenie docelowych systemów operacyjnych które są szczegółowo opisane w rozdziale 1.6. Ogólnie dotyczy to grup wind, którym system przydziela kabinę pasażerom, która zabiera ich na docelowe miejsce. Szczególna winda odpowiadając na to wezwanie, pokazuje kierunkową lampkę z odczytem wskazującym na przypisaną windę. Pasażer wchodzi i zostaje zabrany do miejsca przeznaczenia. Od pasażerów wymaga się innych niż poprzednio działań. W środku kabiny może być więcej niż jeden oczekujący pasażer przez co winda może zatrzymywać się na wielu piętrach. Każdy pasażer musi zwracać uwagę i wsiąść do wózka przeznaczonego dla jego celu podróży. Nie ma panelu obsługi w środku windy więc pasażer popełniający błąd musi wysiąść na losowym piętrze i zacząć od nowa. Zaletą systemu jest to, że ma tendencję do zmniejszania liczby zatrzymuje windę i poprawia obsługę. Ta różnica w obsłudze pasażerów odniosła wielki sukces w budynkach biurowych, w których ludzie zapoznają się z podejściem, ale ograniczono użycie tego w hotelach, w których nieznane systemu powodują trudności.

## Układy elektroniczne sterujące windą

Główną częścią systemu windy nazywana jest kontrolerem. jest urządzenie logiczne, które monitoruje różne funkcje bezpieczeństwa, zapewnia środki do sterowania sygnały i zapewnia wyjścia do sterowania sygnałami i urządzeniami. Pierwsze elektroniczne układy sterowania wind były oparte na logice przekaźnikowej. W nowoczesnych windach logika sterowania jest głównie oparta na sterowniku PLC, prawdopodobnie dlatego, że przekaźniki są droższe niż elementy elektroniczne. Poza tym PLC ułatwia proste połączenia elektryczne i zmniejsza ryzyko awarii. W starszych wariantach wind kierunek samochodu jest kontrolowany przez selektory zamontowane w szybie, po jednym na każdym piętrze. Na nowoczesnych windach może być tylko jeden przełącznik impulsowy, zamontowany na górnej części wózka windy.

Kontroler może mieć różne formy, kształty, rozmiary i może zapewniać funkcję za pomocą przekaźników, dyskretnych elementów logicznych, dedykowanych systemów mikroprocesorowych lub programowalnych sterowników logicznych. Funkcjonalny układ logiczny może znajdować się w skrzynce sterowniczej w maszynowni lub może być zintegrowany z kabiną. Sterowanie ruchem głównie stanowi integralną część systemu lub wysyła sygnały do ​​niezależnego systemu sterowania ruchem. Architektura sterownika jest skończoną maszyna stanu lub wiele maszyn stanów które odczytują dane wejściowe systemu i, w oparciu o aktualny stan systemu, wysyłają wyjścia w ustalonym stanie.

W obecnej chwili sterowniki PLC są powszechnie stosowane w sterowaniu i automatyzacji przemysłu, w którym proste algorytmy są wystarczające. Wraz z rozwojem technologii pojawiły się potrzeby do przetwarzania większej ilości danych a także przyśpieszenie operacji logicznych przez przetwarzanie równoległego. Układy macierzy programowalnych bramek (tzw. FPGA) stanowią lepsze rozwiązanie jako kontroler windy z dodatkowymi zaletami jak konfigurowalność, mniejsze zużycie energii, krótki czas odpowiedzi i elastyczność w rozbudowie projektów. Jedyną wadą tego podejścia jest czas implementacji danego algorytmu, choć ten słaby punkt jest powoli redukowany przez rozwój narzędzi kompilujących kod z języka wysokiego poziomu (Java, Python, Ruby) do języków opisu sprzętu. Biorąc pod uwagę kierunek rozwoju elektroniki oraz wpływ sztucznej inteligencji na nowe systemy, przewidywalne jest przejście układów sterujących do tego typu rozwiązań. Do celów na wysoką skalę produkcji układy FPGA zastępuje się układami o specjalizowanej funkcjonalności (tzw. ASIC). Te układy są znacznie bardziej wydajny niż FPGA ze względu moduły specjalnie projektowane do określonego zadania. Zużycie energii przez układy ASIC można bardzo dokładnie kontrolować i optymalizować. Ponadto w wyspecjalizowanych układach można projektować analogowe peryferia, na przykład nadajnik-odbiornik WiFi, na tym samym waflu krzemowym co rdzenie mikroprocesora. Jest to zaleta, której brakuje FPGA. Ze względu na brak możliwości zmiany funkcjonalności układu wyspecjalizowanego, wersje prototypowe wykonuje się na FPGA aż osiągnie się pełną walidację. Układy scalone w fazie testowania mają możliwość dostosowania konfiguracji do budynku w którym zainstalowano windę. Zminimalizowane układy pobierają mniej energii, wykonują więcej operacji w ciągu sekundy i zajmują znacznie mniejszą powierzchnie niż odpowiadające ich układy PLC. Celem sterowania windą za pomocą układów ASIC jest zwiększenie niezawodności, redukcja wielkości jednostek sterujących windą oraz zmniejszenie poboru energii.

## Przepisy prawne i normy

Instalacja oraz funkcjonowanie dźwigów osobowych w Polsce jest regulowane przez przepisy unijne. Największa zmiana w ostatnim czasie dotycząca norm dźwigowych w Polsce miała miejsce w sierpniu 2017 roku. Obecnie **krajach Unii Europejskiej wszystkie instalowane windy podlegają tym samym dyrektywom i rozporządzeniom.**

**Dyrektywa w sprawie dźwigów 2014/33 / UE zezwala na swobodny obrót dźwigami i elementami bezpieczeństwa do dźwigów na wewnętrznym rynku UE i zapewnia wysoki poziom bezpieczeństwa użytkownikom dźwigów i personelowi obsługi technicznej. To zharmonizowane prawodawstwo UE reguluje projektowanie, produkcję i instalację dźwigów. Dotyczy to głównie instalatorów dźwigów i producentów podzespołów, ale ma również istotne konsekwencje dla właścicieli i użytkowników dźwigów. Obecna dyrektywa jest dostosowany do nowej polityki ram prawnych i ma zastosowanie od 20 kwietnia 2016 r., Zastępując poprzednią dyrektywę 95/16 / WE [12].**

**Bezpieczeństwo istniejących dźwigów (zainstalowanych przed wejściem w życie dyrektywy 95/16 / WE w sprawie dźwigów) podlega przepisom krajowym. Zalecenie dotyczące poprawy bezpieczeństwa istniejących dźwigów zachęca kraje UE do podjęcia wszelkich niezbędnych działań w celu zapewnienia zadowalającego poziomu utrzymania istniejących dźwigów i poprawy bezpieczeństwa tych dźwigów. Zalecenie nie jest prawnie wiążące i jest wdrażane przez kraje UE w świetle sytuacji i przepisów istniejących na szczeblu krajowym [13].**

**W odniesieniu dostępu dla osób niepełnosprawnych do kabin wind, kraje UE zachęca się do podejmowania wszelkich środków krajowych niezbędnych do zapewnienia, że ​​wszystkie poziomy istniejących budynków, jak również te w budowie, są dostępne dla osób niepełnosprawnych, w szczególności dla osób korzystających z wózków inwalidzkich. Zaleca się, aby we wszystkich nowych budynkach zapewnić co najmniej jedną windę dostępną dla osób niepełnosprawnych na wózkach inwalidzkich. Ponadto winda musi spełniać wszystkie wymogi regulacyjne (w zakresie wymiarów, położenia elementów sterujących itp.) [12].**

**Standaryzacja wind i schodów ruchomych nie kończy się na granicach Unii Europejskiej. Rynek wind i schodów ruchomych jest naprawdę globalny, nawet w porównaniu z innymi sektorami przemysłu: ponad 75% wszystkich nowych instalacji rocznych (63% w samych Chinach), region Azji i Pacyfiku dominuje obecnie na globalnym rynku wind i schodów ruchomych i jest ma wzrosnąć jeszcze bardziej w latach 2016-2023. Europejski przemysł dźwigowy ze ściśle zintegrowanymi łańcuchami produkcyjnymi i dostawczymi na całym świecie, ma znaczący udział w globalnym rynku. W tej sytuacji niezwykle ważne jest, aby normy europejskie były otwarte na świat: dzięki uznawaniu norm także poza rynkiem europejskim firmy, które je przyjmują, nie muszą obawiać się, że zostaną odcięte od rozwijających się rynków Azji i Pacyfiku z powodu niezgodności. Pozytywne jest zatem, że większość krajów uznała wartość norm CEN / TC 10, zwłaszcza EN 81-20 / 50 i EN 115-1, i postanowiła wdrożyć je w swoim systemie regulacyjnym [14].**

## System przyszłości

### Algorytm wysyłki docelowej

W dzisiejszych miastach, gdzie dominują wieżowce, windy stały się nieodzownym środkiem transportu w życiu codziennym. Jednak coraz więcej pięter i użytkowników sprawiło, że tradycyjny system windy nie jest w stanie skutecznie rozprowadzać i przenosić użytkowników do miejsca przeznaczenia, co powoduje problemy z wydajnością transportu w budynku, takie jak długi czas oczekiwania w holu windy czy wydłużony czas podróży.

Z tego powodu w większości budynków zastosowano metody mające na celu zwiększenie wydajności transportu budynków, takie jak wdrożenie strefowego systemu windy a nawet zastosowanie dwupoziomowych wind. Wraz z rozwojem technologii doprowadziło to do stworzenia bardziej optymalnego systemu dystrybucji wind, znanego jako „wysyłka docelowa”, który w ostatnich latach stał się bardziej atrakcyjną technologią dla wind, przerywając stosowanie tradycyjnych systemów wind i metod dystrybucji.

Tradycyjne systemy wind mają przyciski wezwania na zewnątrz windy do wywoływania windy. Po tym, jak pasażer wejdzie do windy, wciska przycisk żądanego piętra. System nie może przewidzieć pięter, do których wszyscy pasażerowie muszą się udać, więc pasażerowie są zmuszeni do jazdy na różne piętra. Zwiększa to zatem czas jazdy windy, a nawet powoduje, że wszystkie windy podnoszą się i wracają do głównego piętra w tym samym czasie, bezpośrednio wpływając na czas oczekiwania i wydajność windy. Dzięki docelowemu systemowi wysyłkowemu wyeliminowano tradycyjne przyciski wywołania. Zamiast tego pasażerowie wjeżdżają do miejsca docelowego przez wybranie docelowego piętra przed szybem windy, na urządzeniu takim jak klawiatura dziesiętna lub ekran dotykowy.



*Rysunek 1‑5 Wybór piętra w systemie z wysyłką docelową*

***Źródło: []***

System wysyłania miejsc docelowych polega na grupowaniu pasażerów według tego samego miejsca docelowego, gdy tylko znajdą się w miejscu przeznaczenia i przypisuje ich do tych samych wagonów windowych. Kieruje poszczególnymi wagonami windy w grupie, aby obsługiwały tylko określone piętra. Dlatego liczba przystanków jest zmniejszona, a ponieważ winda zatrzymuje się na kilku przystankach, czas podróży jest krótszy niż w konwencjonalnym systemie wind. Zasadą tego systemu jest doprowadzenie pasażerów do miejsca docelowego w możliwie najkrótszym czasie z mniejszym skupiskiem i komfortem.



*Rysunek 1‑6 Przyciski żądań w windzie z wysyłką docelową i klasycznej windzie*

***Źródło: []***

Metodę alokacji można zmienić w zależności od sytuacji. Na przykład, przejście do wyższych, niższych lub sąsiednich pięter zostanie przypisane do tej samej windy, aby uniknąć nieefektywności, takich jak zatrzymanie się na piętrze gdzie nie użytkownik nacisnął zły przycisk kierunku ruchu. Niektóre systemy wind dwupokładowych, w połączeniu z tym systemem, mogą sprawić, że winda będzie bardziej elastyczna, dopasowana do potrzeb budynku.

Windy wyposażone w docelowy system wysyłkowy mają co najmniej dwie klawiatury numeryczne lub ekrany dotykowe w każdym holu, które zastępują tradycyjne przyciski żądań. Jeśli windy mają ekrany dotykowe, zazwyczaj zawierają listę dostępnych pięter obsługiwanych przez windy. Istnieją również tabliczki identyfikacyjne windy zawierające literę do identyfikacji wind (np. A, B, C itd.).



*Rysunek 1‑7 Identyfikacja wind w systemie wysyłkowym*

***Źródło: []***

Pasażerowie są przypisywani do danych szybów i nie mają potrzeby wybierania pięter w środku wagonów. Pozostawiając tylko przyciski otwierania i zamykania drzwi oraz przyciski alarmowe. Przyciski pięter są ukryte za panelem i zwykle nie są używane w normalnych warunkach. Niektóre windy mogą nadal posiadać przyciski podłogowe, ale nie można ich nacisnąć; wskazują tylko piętra na których winda się zatrzymuje. Istnieją również systemy z konfiguracją hybrydową, gdzie przyciski można naciskać. W takim układzie panel sterowania z wysyłką docelową znajduje się tylko na niektórych określonych piętrach a na pozostałych piętrach obowiązuje układ konwencjonalnej windy.

### Kierunki rozwoju technologii

Pomimo tak wyrafinowanych systemów operacyjnych wind, wciąż istnieją ogromne możliwości dla rozwoju i ulepszeń. Producenci wind pracowali ciężko by poprawić wydajność windy zarówno z perspektywy operacyjnej w systemach dyspozytorskich oraz z perspektywy efektywności energetycznej z ulepszoną konstrukcją silnika i napędu. Praktyczne, wydajne przekładnie zębate i przekładnie do zastosowań niskopoziomowych zastępują mniej wydajne konstrukcje hydrauliczne.



*Rysunek 1‑8 Nowa generacja mechanizmów wind GEN2 LIFT firmy OTIS*

*Źródło: []http://www.abpenterprises.com/gen2-premier.html*

Brak maszynowni pozwala architektom na większą elastyczność projektowania. Konstruktorzy korzystają z kontrolowanego, usprawnionego procesu instalacji i minimalnej ingerencji w inne transakcje. Dla właścicieli budynków system Gen2 przekłada się na niższe koszty budowy i znaczny wzrost powierzchni do wynajęcia. Modułowy kontroler, zaprojektowany dla grup składających się z maksymalnie trzech samochodów, zawiera nową generację płytek drukowanych i oprogramowania, aby zapewnić optymalny czas reakcji. Napęd VF z cyfrową pętlą zamkniętą, z technologią sterowania wektorowego, dodatkowo zwiększa wydajność i dokładność, a cyfrowy enkoder prędkości zapewnia prawidłową prędkość i pozycję samochodu. Rezultatem jest system wyjątkowej niezawodności.

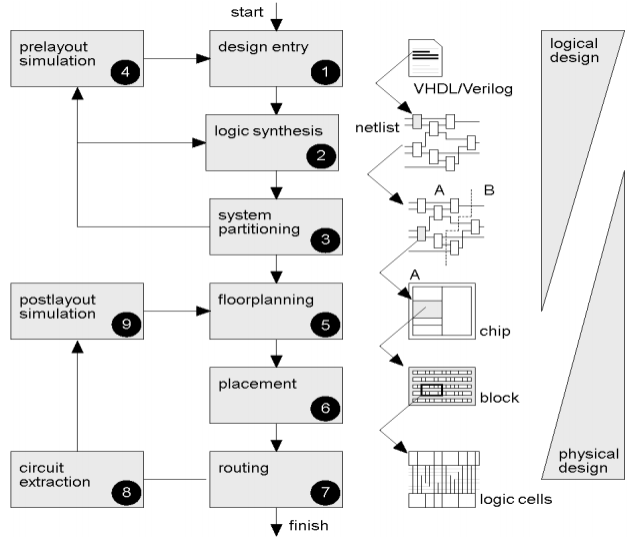
Trend będzie kontynuowany przy użyciu lżejszych materiałów, bardziej wydajnych napędów regeneracyjnych. Ulepszenia w zdalnym monitorowaniu, niezawodności systemu i konserwacji opartej na użytkowaniu będą kontynuowane przez bardziej zaawansowaną diagnostykę kontrolerów oraz wyższy poziom integracji z narzędziami zarządzania utrzymaniem ruchu. Poprawiono poziom niezawodności, co skutkuje mniejszą ilością zbudowanych wind w budynku o określonej powierzchni. Niezawodność będzie miarą tego, czy takie systemy będą praktyczne w budynkach o dużym natężeniu ruchu. Z związku z rozwojem technologii następuje ciągłe ulepszanie interfejsów użytkownika i urządzeń bezpieczeństwa dla wind.

# Opis algorytmu w języku Verilog

## Tworzenie modułów z wykorzystaniem języka Verilog

Język Verilog HDL zawiera funkcje opisujące charakter projektu, przepływ danych, skład strukturalny, opóźnienia i mechanizm generowania przebiegów, w tym aspekty monitorowania odpowiedzi i weryfikacji, wszystkie modelowane przy użyciu jednego języka. Ponadto język zapewnia interfejs języka programowania, dzięki któremu można uzyskać dostęp do elementów wewnętrznych projektu podczas symulacji, w tym do sterowania przebiegiem symulacji. Język nie tylko definiuje składnię, ale także definiuje bardzo jasną semantykę symulacji dla każdego konstruktu językowego. Dlatego modele napisane w tym języku można zweryfikować za pomocą symulatora Verilog. Język dziedziczy wiele symboli operatora i konstruktów z języka programowania C. Verilog HDL zapewnia szeroki zakres możliwości modelowania, z których niektóre są dość trudne do zrozumienia na początku. Jednak podstawowy podzbiór języka jest łatwy do opanowania i użycia. Jest to wystarczające do modelowania większości aplikacji.

Narzedia???//



*Rysunek 2‑1 Schemat projektowania układu ASIC*

*Źródło: []https://www.slideshare.net/yogeshwarank1/asic-design-flow-71285160*

## Rozwój oprogramowania w oparciu o cechy użytkowe

Wybraną metodologia tworzenia oprogramowania jest podejście iteracyjne z nastawieniem na implementacje cech produktu (ang. *Feature-Driven Development)*. Jest to zwinna metoda tworzenia oprogramowania. Ten proces łączy wiele uznanych, najlepszych praktyk w branży programistów w spójną całość. Praktyki te są sterowane z perspektywy funkcjonalności, które są najważniejsze dla klienta. Jego głównym celem jest dostarczanie namacalnego, działającego oprogramowania wielokrotnie w odpowiednim czasie, zgodnie z zasadami stojącymi za manifestem Agile.

Głównym elementem tej metodyki jest wyznaczenie cech produktu do stworzenia i zaplanowaniu ich implementacji w krótkim, określonym czasie. Duża funkcjonalność jest dzielona na mniejsze co pozwala na kontrolę nad funkcjonalnością produktu. Po każdej iteracji nowe cechy są integrowane z dotychczasową wersją oprogramowania i testowane. W przypadku wystąpienia błędu, kod jest analizowany i poprawiany. Produkt jest dostarczany okresowo wraz z nowo wprowadzonymi zmianami. Zaletą tego podejścia jest rozbicie projektu na mniejsze części, które są łatwe do kontrolowania. Podzielenie architektury oprogramowania na małe obszary umożliwia pokrycie w dużym stopniu przez testy modułowe i zapewnienie wysokiej jakości produktu. Kolejną zaletą jest możliwość testowania wraz z rozwojem oprogramowania co nie powoduje przerwy na testowanie po zakończeniu pracy programistów. Klasyczna metoda FDD składa się z pięciu faz:

1. Opracowanie ogólnej architektury produktu
2. Budowanie listy cech produktu
3. Zaplanowanie w czasie implementacji cech
4. Projektowanie każdej z listy planowych cech
5. Budowanie oprogramowania w nastawieniu na cechy

Ostatnie dwa punkty są powtarzane aż do zakończenia implementacji wszystkich zaplanowanych cech. W przypadku rozbudowy produktu, klient może zaproponować nowe funkcje, które zostaną być dołączone w nowych iteracjach. Takie podejście do tworzenia oprogramowania jest bardzo adaptacyjne i ułatwia rozbudowę kodu. Charakterystyczne cechy tej oraz podobnych metod zostały określone jako zwinne (ang. agile)

Przy wyborze metodologii Agile wszystko zależy od wymagań projektu. Na przykład, w przypadku małych projektów, które nie są złożone, można łatwo przejść do programowania ekstremalnego, czyli takiego, gdzie nie wiadomo do końca, co się tak naprawdę robi i jak to prawidłowo zrobić, głównie opartego na obserwacji innych projektów, które odniosły sukces. Metodyki zwinne takie jak Scrum czy FDD są zalecane, jeśli chodzi o projekty oprogramowania, które są bardziej złożone i większe. Odnosząc się do tematu pracy, metodologia z nastawieniem na implementacje cech produktu była słuszna do tego projektu gdyż pierwsze próby opisu kontrolera polegały na cechach modelu obserwowanych systemów. W początkowej fazie kluczowe było określenie cechy windy, jak będzie wyglądać kontrola jazdy wózka bazując na istniejących rozwiązaniach standardowej windy. Te wymagania sprawiły, że mimo małej złożoności projektu i jednoosobowym zespole użycie tej metodologii miało sens i w dużym stopniu pozwoliło na sukcesywną implementacje algorytmu sterowania windą.

## Architektura systemu

### Moduł pełnego piętra

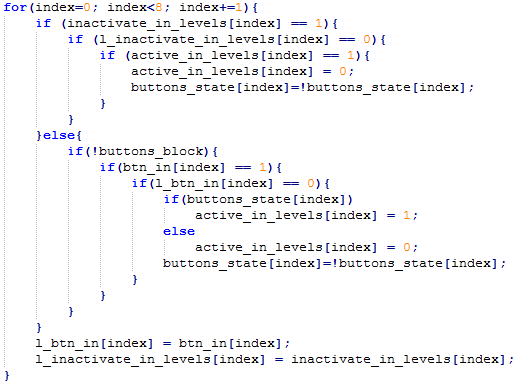
### Moduł piętra przejściowego

### Interfejs użytkownika

Zgodnie z wybrana modelem strategii zbiorowej kontroli wysyłanie żądań do systemu windy odbywa się przez przyciski na zewnątrz szybu oraz wewnątrz kabiny.

Przyciski zewnętrzne

Przyciski wewnetrzne



*Rysunek 2‑2 Kod dla resetowalnych przycisków*

***Źródło: Opracowanie własne***

### Parametry czasowe i bezpieczeństwo

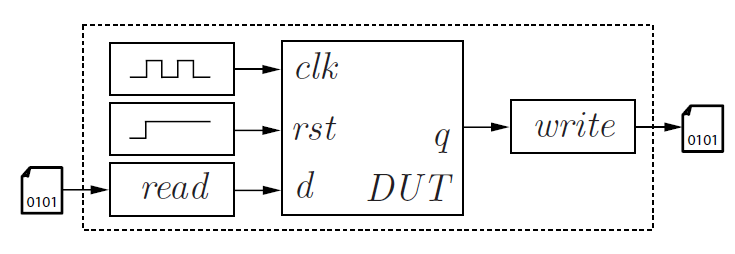
## Weryfikacja kodu

Zgodnie z metodologią tworzenia oprogramowania w oparciu o cechy, równocześnie z fazą implementacji, nowe funkcje są testowane. Przetestowanie jest kluczową sprawą przy projektowaniu układu ASIC. Układy te nie mają możliwości zmiany struktury po fabrykacji, dlatego zapewnienie pokrycia funkcjonalności przez scenariusze testowe jest sprawą kluczową. Najlepszą metodą przy testowaniu algorytmu, którego kod jest znany są testy strukturalne (ang. *white box*). Trzy wykorzystane techniki w testowaniu:

* Pokrycie linii kodu *(ang. statement coverage)* – technika testowania sprawdzająca wszystkie linijki implementowanego kodu, znane również jako pokrycie segmentu. Określa, które fragmenty programu pokrywają się ze sobą, czyli zostały wywołane przez zestaw testowy oraz wskazuje na te, których nie udało się wywołać. *Statement coverage* pozwala w prosty sposób zidentyfikować obszar niewytestowanego kod oraz znaleźć „kod martwy”, który nigdy nie zostanie wykonany.
* Pokrycie gałęzi *(ang. branch coverage)* – w języku programowania jest ściśle związana z instrukcją IF, która posiada dwa „odgałęzienia”: Prawdę i Fałsz (*True and False*). Ta metoda polega na sprawdzeniu, czy każda z gałęzi decyzji po instrukcji IF jest wykonywana co najmniej raz. Generalnie będą występowały dwa warunki, jedna gałąź do sprawdzenia, gdy warunek jest spełniony i pozostała, gdzie warunek jest fałszywy. Jeżeli dla drugiej sytuacji nie istnieje żadna instrukcja do wykonania to sprawdzana jest tylko jedna gałąź. Takie podejście zapewnia, że kod z decyzyjnego punktu widzenia jest poprawnie wykonywany.
* Analiza statyczna kodu (ang. static analysis) - metoda znajdowania błędów programu komputerowego, która jest wykonywana przez badanie linijek kodu bez wykonywania programu. Proces ten powstaje przez zrozumienie struktury kodu i może pomóc w zapewnieniu zgodności ze standardami branżowymi. Zautomatyzowane narzędzia mogą pomóc programistom w przeprowadzeniu analizy statycznej. Proces sprawdzania kodu wyłącznie przez kontrolę wizualną (na przykład przez oglądanie wydruku), bez pomocy automatycznych narzędzi, jest czasami nazywany zrozumieniem programu. Główną zaletą analizy statycznej jest fakt, że może ona ujawniać błędy, które nie pojawiają się, dopóki nie nastąpi wyjątkowy scenariusz po pewnym czasie. Niemniej jednak analiza statyczna jest tylko pierwszym krokiem w kompleksowym systemie kontroli jakości oprogramowania. Po przeprowadzeniu analizy statycznej często przeprowadza się analizę dynamiczną w celu wykrycia subtelnych defektów lub luk. Analiza dynamiczna obejmuje testowanie i ocenę programu opartego na wykonaniu. Analizy statyczne i dynamiczne, rozpatrywane razem, są czasami nazywane testami „na szkle”.

### Przypadki testowe

W poprzednim paragrafie wspomniano jak bardzo ważne podczas pisania kodu projektu ważne jest sprawdzenie jego funkcjonalności. Najbardziej powszechną metodą weryfikacji kodu w języku opisu sprzętu jest utworzenie testbench’a, tj. utworzenie instancji DUT (ang. Device Under Test), wygenerowania wektorów testowych (zestaw wejść) i monitorowanie wyjście, jak pokazano na rysunku 2-3. Wspólne zadania testbench mają generować sygnały zegara i resetowania oraz odczyt / zapis informacji do pliku le. Pisanie wyjścia wartości do le umożliwiają weryfikację wyniku za pomocą np. Matlab.



*Rysunek 2‑3 Przykładowy testbench dla przerzutnika D*

***Źródło: Opracowanie własne***

### Moduł testowy

### Wnioski/podsumowanie

## Modułowosc i rozbudowa kodu??? If not earlier

# Projektowanie układu scalonego

## Generacja

Zestaw uruchomieniowy na płytce MAXimator (Rys. 3-1) jest świetnym rozwiązaniem nie tylko dla poczatkujących elektroników, ale również dla profesjonalistów i hobbystów. Cechuje się nowoczesnym układem firmy Altera, należącym do rodziny MAX10 o dużych zasobach logicznych. Najważniejsze cechy dostępne dla użytkownika to:

* ponad 8000 komórek LE,
* 378kb konfigurowalnej pamięci,
* 1376kb wewnętrznej pamięci Flash dla aplikacji,
* wbudowany 12-bitowy przetwornik A/C 1MSpS,
* wewnętrzny generator sygnału zegarowego,
* 2 wewnętrzne wielokanałowe syntezery PLL.

*Rysunek 3‑1 Wygląd płytki bazowej MAXimator*

Zalety układów MAX10 są bardzo dobrze wyeksponowane w zestawie MAXimator. Budowa płytki konstrukcyjnie przypomina Arduino Uno Rev.3 co pozwala na różnorodne zastosowanie, również w celach edukacyjnych. W skład zestawu MAXimator wchodzą:

* płytka MAXimator z układem 10M08,
* programator zgodny z USB Blaster,
* ekspander z 4-cyfrowym wyświetlaczem LED, analogowym sensorem temperatury, trzema przyciski i dwiema diodami LED RGB.

W odróżnieniu od zestawów Arduino, MAXimator nie ma wbudowanego programatora, lecz korzysta z dołączonego do zestawu USB Blastera (Rys. 3-2), który pozwala na zaprogramowanie dostarczonego układu naszym kodem przez środowisko Quartus Prime. Kolejną atrakcją zestawu jest ekspander (Rys. 3-3) mający szeroką gamę funkcjonalności dla projektantów. Wyposażony w cztero-pozycyjny, siedmio- segmentowy wyświetlacz LED sterowany multipleksowo, dwie diody RGB, czujnik temperatury oraz trzy przyciski: jeden przycisk *RESET* a pozostałe dwa do dowolnego wykorzystania przez użytkownika.

## Synteza

Cały projekt jest przystosowany do współczesnych standardów wraz z funkcją obsługi przez użytkownika. Syntezer składa się z:

* dekodera MIDI,
* sterownika oscylatorów,
* dwóch oscylatorów,
* miksera sygnałów,
* obsługi interfejsu użytkownika.

Schemat blokowy architektury projektu widnieje na rysunku 3.2.1. Wszystkie komponenty są sterowane narastającym zboczem zegara oraz negatywnym stanem resetu. Weryfikacja poprawnej działalności modułów następuje przez symulator ModelSIM Altera.

*Rysunek 3‑2 Architektura syntezera*

Rozwój architektury odbywał się na równoległym poziomie z testowaniem zaimplementowanych funkcji. W czasie tworzenia cyfrowego syntezatora opartego na DDS jedyną trudność sprawiła implementacja akumulatora fazy i zrozumienie jego działania.

## Tworzenie layoutu

Korzystanie z interfejsu MIDI jest praktycznym zabiegiem dla współczesnych syntezatorów. W implementacji, dekoder MIDI otrzymuje wejściową ramkę i przesyła dwie informacje: start dźwięku od danym tonie i stop dźwięku. Blok dekodera został przedstawiony na rysunku 3-5.

## Bezpośrednia synteza cyfrowa

Metoda bezpośredniej syntezy cyfrowej to technika generacji sygnałów, oparta w całości na cyfrowych metodach syntezy sygnału. Układy DDS stały się alternatywą dla analogowych syntezerów opartych na pętlach sprzężenia fazowego, zapewniając szybkie przestrajanie w szerokim zakresie częstotliwości oraz regulację amplitudy. Generatory korzystające z bezpośredniej syntezy cyfrowej charakteryzują się dokładnością w tworzeniu sygnałów dzięki dostępowi do pamięci, gdzie zawierają się próbki oraz specjalnego mechanizmowi zegarowemu, który różni się od tradycyjnej metody.

*Rysunek 3‑3 Blok bezpośredniej syntezy cyfrowej*

## Generacja sygnałów

*Rysunek 3‑4 Schemat blokowy oscylatora DDS*

### Zegar referencyjny

Zegar referencyjny ma za zadania wytworzenie słowa przestrajającego, aktualizowanie wartość akumulatora fazy oraz kieruje konwersją cyfrowo-analogową. Zegar referencyjny determinuje, kiedy próbka jest przesyłana w przetworniku, ale nie ma

### Akumulator fazy

Sercem każdego syntezatora korzystającego z bezpośredniej syntezy cyfrowej jest akumulator fazy. Ten komponent jest zbiorem słów przestrajających(ang. tuning word), które pozwalają na dokładne wytworzenie częstotliwości sygnału. Każde z tych słów oznacza skok między próbkami. Im większy skok tym więcej wartości jest pomijanych,co skutkuje szybszym pokonaniem koła fazy cyfrowej przedstawiony na rysunku 3-8. Jedno okrążenie wokół koła cyfrowego oznacza jeden okres dla sygnału. Największe

*Rysunek 3‑5 Koło fazy cyfrowej*

Moduł *NOTE\_TO\_PA* przedstawiony na rysunku 3-7 jest odpowiednikiem tablicy przekształcenia numeru danego dźwięku na skok akumulatora fazy. Wartości w tej tablicy są liczone według wzoru 3.5.1.

Do obliczenia częstotliwości generowanego sygnału służy wzór:

fOUT – częstotliwość wyjściowa,

M – słowo przestrajające,

fC - częstotliwość zegara referencyjnego,

n- rozdzielczość akumulatora fazy.

*Rysunek 3‑6 Analiza czterokrotnego pomnożenia kroku*

W celu przybliżenia generacji sygnału przy pomocy koła fazowego niżej przedstawiono graficznie jak wartości na kole fazowym odpowiadają wartościom napięć. Przed zapisywaniem sygnału sinusoidy, powstają próbki a po zakończeniu generacji sygnału cyfrowego, następuje konwertowanie do sygnału analogowego przez przetwornik.

*Rysunek 3‑7 Generacja sinusoidy*

Dodatkowym elementem w akumulatorze fazy jest rejestr fazy. Jest to operacja zmniejszania rozdzielczości akumulatora fazy. Duża rozdzielczość akumulatora fazy jest wymagana do otrzymania mały odstępów częstotliwość na każdy skok(około 0,006Hz na skok dla implementowanego syntezera). Rejestr fazy pozwala na użycie mniejszych tablic niż rozdzielczość akumulatora fazy. Ten zabieg oszczędzą miejsce w pamięci, jego początkiem był fakt nierozwiniętej w takim stopniu technologii do trzymania tak dużej ilości danych. W ten sposób możemy regulować dokładność generowanego przebiegu bez ingerencji w efektywność akumulatora fazy.

### Tablica z próbkami

Implementowany syntezator zawiera cztery różne pamięci tablicowe. Każda z nich zawiera jeden okres przebiegu o danym kształcie, co pozwala na dokładne odtworzenie sygnału przez akumulator fazy. Sygnał z wyjścia rejestru fazy determinuje wybór próbki z tablicy. Tablica składa się z 256 wartości, które zawierają kształt danego przebiegu. Wyjątkiem jest tablica przebiegu prostokątnego, która zawiera tylko dwie wartości – amplitudę stanu niskiego i wysokiego. Wartość wejściowa z rejestru fazy oznacza wybór danej wartości z tablicy próbek a następnie ta wartość próbki jest przesyłana do instancji miksera. Przykładowa generacja sygnału z wykorzystaniem tablicy została przedstawiona na rysunku 3-11.

*Rysunek 3‑8 Generacja sinusoidy z wykorzystaniem tablicy*

Wartości próbek w tablicach są zapisane w formacie 16-bitowych zmiennych ze znakiem. Przedział amplitud został ograniczony w przedziale od -32000 do 32000.

Użycie akumulatora fazy oraz metody tablicowej generuje dokładne przebiegi. Wady i zalety metody tablicowej zostały opisane w punkcie 1.4.1 *Synteza z użyciem tabeli przebiegu fali.*

### Mikser

Połączenie sygnałów z kilku oscylatorów wymaga użycia miksera. Mikser kontroluje wyjściową wartość amplitudy, aby sygnał nie uległ nasyceniu przez duże wartości amplitud kilku sygnałów.

*Rysunek 3‑9 Blok miksera*

Moduł miksera został zbudowany w oparciu o istniejące przykłady. Wszystkie sygnały z oscylatorów są przekazywane a użytkownik korzystając z przycisków zmienia wartość sygnału *WAV\_SELECT*. Dane wyjściowe są kierowane do przetwornika lub interfejsu audio.

## Interfejs użytkownika

Użytkownik ma umożliwiona kontrolę na generowanym sygnałem. Korzystając z przycisków oraz wyświetlacza na ekspanderze można wybrać kształt fali na wyjściu oraz liczbę użytych oscylatorów. Przyciski są wyposażone w debouncer, który zatrzaskuje się po naciśnięciu na jedną sekundę, dzięki czemu mikrokontroler nie ma problemów z odczytywaniem zamiarów użytkownika. Liczba użytych wyświetlaczy została zredukowana do jednego, ze względu na wybór mniej niż dziesięciu funkcji syntezatora. Ewentualna rozbudowa architektury projektu nie przeszkadza w użyciu pozostałych wyświetlaczy.

## Dźwięk w programie Audacity

Z powodu problemów generacji dźwięku interfejsem HDMI, osiągniecie rezultatów osiągnięto przez symulacje, zapis wygenerowanych próbek oraz odtworzenie pliku pcm w darmowym edytorze plików dźwiękowych Audacity.

*Rysunek 3‑10 Sygnały wygenerowane w Audacity*

Generacja sygnału w tym programie odbywa się poprzez importowanie pliku pcm. Końcowe rezultaty potwierdzają poprawne działanie syntezatora. Do wykonania poprawnej generacji należy użyć widoku listu, przeciągnąć wygenerowany sygnał do okna listy i zapisać do pliku.

Generowanie dźwięku odbywa się w trzech krokach:

* zapis danych wyjściowych z symulacji,
* konwersja z formatu hex do ASCII i zapis pliku z rozszerzeniem pcm,
* import danych z powyższymi ustawieniami.

Ustawienia do generowania dźwięku w Audacity:

* kodowanie – bez znaku 16-bit PCM,
* kolejność bajtów – big endian,
* kanały – 1 kanał (mono),
* offset – 0 bajtów,
* ilość importowanych bajtów – 100%,
* częstotliwość próbkowania – 44100 Hz.

Wygenerowane sygnały są eksportowane do formatu wav, który umożliwia odsłuchanie próbki. Otrzymane wyniki są zgodne z założeniami i potwierdzają prawidłowe działanie syntezera. Na tym etapie prace nad syntezatorem zostają zakończone.

# Podsumowanie i wnioski

Implementacja syntezatora została zakończona sukcesem. Niestety jedno z głównych założeń, czyli generacja dźwięku przy użyciu interfejsu HDMI nie zostało spełnione. Liczba prób realizacji tego założenia oraz czas włożony w rozwój projektu pod tym względem pozostawia możliwość ukończenia sterowania interfejsem HDMI w prawdopodobnie krótkim czasie. Potencjał płytki MAXimator nie został całkowicie wykorzystany, co pozostawia chęć do zbudowania syntezatora o większej funkcjonalności.

W projekcie wykorzystano układ FPGA jako generator sygnałów pracujący w trybie bezpośredniej syntezy cyfrowej. Istnieje duża możliwość rozbudowania projektu o większą ilość sygnałów, dodanie filtru lub efektów dźwiękowych, które ze zostały zaimplementowane z uwagi na. Udało się zaobserwować wygenerowane sygnały oraz przetworzyć do formatu .wav oraz je odsłuchać. Na chwilę obecną syntezator jest narzędziem czysto programowym ale już są planowane pracę nad uruchomieniem projektu wraz z interfejsem HDMI lub innym dostępnym kodekiem.

Projekt w układzie FPGA zajmuje:

* 1412 logicznych elementów (18% całości),
* 387 rejestrów,
* 11 z 178 dostępnych pinów,
* jeden moduł PLL z dwóch dostępnych.

# Streszczenie

Tematem pracy jest syntezator dźwięku w układzie FPGA. Projekt został zbudowany w środowisku Intel Quartus Prime z wykorzystaniem języku opisu sprzętu VHDL. Jednym z głównych założeń projektu była generacja dźwięku przez interfejs HDMI. Niestety po wielu próbach, prace nad uruchomieniem HMDI zakończyły się niepowodzeniem ale projekt będzie kontynuowany celem osiągniecia zamierzonych rezultatów.

Syntezator jest instrumentem muzycznym, generatorem sygnałów akustycznych. Dogłębna analiza tematu współczesnych syntezatorów wskazała na użycie interfejsu MIDI do komunikacji między urządzeniem a innymi instrumentami.

Użytkownik obsługuje instrument przy użyciu przycisków znajdujących się na płycie ekspandera dołączonej do zestawu MAXimator. Funkcja syntezatora zostaje wyświetlona na płytce, tak by osoba sterująca miała pełną kontrolę nad urządzeniem.

W pracy została zawarta rozprawa na temat różnic w budowie i funkcjonalności syntezatorów a także nacisk na funkcjonalność i dopasowanie metod syntezy do cyfrowego układu FPGA. Ostatecznie w projekcie wykorzystano metodę syntezy bezpośredniej z wykorzystanie tablic pamięci do generacji przebiegów. Satysfakcjonujące wyniki otrzymano przez symulacje i generacje zapisanych próbek do formatu wav.

# Summary

The subject of the diploma is a sound synthesizer in FPGA. The project was built in the Intel Quartus Prime environment using the VHDL hardware description language. One of the main assumptions of the project was the generation of sound via the HDMI interface. Unfortunately, after many attempts, work on starting the HMDI was unsuccessful. Despite the lack of results, the project will be continued in order to achieve the intended results.  
 Synthesizer is a musical instrument, a generator of acoustic signals. After depth analysis of the topic of modern synthesizers indicated the use of MIDI interface for communication between the device and other instruments.  
 The user operates the instrument using the buttons on the expander board attached to the MAXimator kit. The synthesizer function is displayed on the board so that the control person has full control over the device.  
 The work included a dissertation on the differences in the structure and functionality of synthesizers, as well as the emphasis on functionality and adaptation of synthesis methods to the digital FPGA system. Finally, the project used a direct synthesis method with the use of memory tables for generation of waveforms. Satisfactory results were obtained by simulations and generations of saved samples to the wav format.

# Spis stosowanych skrótów

ASIC – Autoregresja i ruchoma średnia

ADSR – Attack Decay Sustain Release

FDD – Direct Digital Synthesis

FPGA – Field Programmable Gate Array

# Słowa kluczowe

* ASIC,
* syntezator,
* bezpośrednia synteza cyfrowa,
* synteza częstotliwości,
* generacja dźwięku,
* MAXimator,
* Altera MAX10.

# Keywords

* FPGA,
* synthesizer,
* direct digital synthesis,
* frequency synthesis,
* sound producing,
* MAXimator,
* Altera MAX10.

# Bibliografia

1. Antoszkiewicz K., *Generacja i synteza częstotliwości*, Warszawa: Oficyna Wydawnicza Politechniki Warszawskiej, 2015, str. 260-271.
2. Bhasker J., *A VHDL Primer,* Upper Saddle River: Prentice Hall PTR, 1999.
3. Naylor D., Jones S., *VHDL: a logic synthesis approach,* London: Chapman & Hall, 1997.
4. Pellegrino R., *The Electronic Arts of Sound and Light*, New York: Van Nostrand Reinhold Company, 1983.
5. Strakosch The verical transportation nhandbook
6. Elevator control strategies j axelsson,s bernlind
7. Algorytm Scan http://aragorn.pb.bialystok.pl/~wkwedlo/OS1-9.pdf
8. Elevators https://www.slideshare.net/saramesallam/shanghai-tower-elevator-and-escalators
9. Shanghai towr <https://en.wikipedia.org/wiki/Shanghai_Tower>
10. Elevator gruop <https://link.springer.com/content/pdf/10.1023%2FA%3A1007518724497.pdf>
11. Designing Elevator Installations Using Modern Estimates of Passenger Demand <https://www.peters-research.com/index.php?option=com_content&view=article&id=57%3Alift-passenger-traffic-patterns-applications-current-knowledge-and-measurement&catid=3%3Apapers&Itemid=1>
12. Elevators UE <https://ec.europa.eu/growth/sectors/mechanical-engineering/lifts_en>
13. Directive 95/16/EC - Lifts - European standards for existing lifts – references
14. http://www.elca-eu.org/

http://www.worldofarchi.com/2012/08/shanghai-tower-elevator-system-drawings.html

https://vignette.wikia.nocookie.net/elevation/images/a/aa/BigTransit.jpg/revision/latest?cb=20130115104717&format=original

http://www.neii.org/destdispatch.cfm

<https://www.tripadvisor.com.au/LocationPhotoDirectLink-g53449-d294369-i310683750-Cathedral_of_Learning-Pittsburgh_Pennsylvania.html>

<https://elevation.fandom.com/wiki/Destination_dispatch>

<https://www.retrofitmagazine.com/destination-dispatch-system-improves-buildings-elevator-performance-and-efficiency/>

<http://global.ctbuh.org/resources/papers/download/399-elevator-traffic-simulation-procedure.pdf>

https://link.springer.com/content/pdf/10.1023%2FA%3A1007518724497.pdf

# Dodatek A. Schemat połączenia MIDI z portem szeregowym UART



# Dodatek B. Spis zawartości dołączonej płyty CD

All About Direct Digital Synthesis \_ Analog Devices.htm – strona internetowa zawarta w bibliografii na pozycji 8,

Everything You Need to Know About Direct Digital Synthesis.htm – strona internetowa zawarta w bibliografii na pozycji 9,

Mateusz\_Dyrdół\_276528.doc – Tekst pracy zapisany w formacie MS Word,

Mateusz\_Dyrdół\_276528.pdf – Tekst pracy zapisany w formacie rozszerzonym,

synth.qar – Archiwum projektu wygenerowane w środowisku Intel Quartus Prime,

sine.pcm – plik w rozszerzeniu pcm do generacji sygnału sinusoidalnego,

# Spis ilustracji

[*Rysunek 1‑1 Ruch głowicy dysku twardego zgodny z algorytmem SCAN* 9](#_Toc11938932)

[*Rysunek 1‑2 System strefowy w budynku Shanghai Tower* 11](#_Toc11938933)

[*Rysunek 1‑3 Przykład połączenia dwóch stref* 12](#_Toc11938934)

[*Rysunek 1‑4 Model windy dwuprzystankowej* 15](#_Toc11938935)

[*Rysunek 1‑5 Wybór piętra w systemie z wysyłką docelową* 21](#_Toc11938936)

[*Rysunek 1‑6 Przyciski żądań w windzie z wysyłką docelową i klasycznej windzie* 22](#_Toc11938937)

[*Rysunek 1‑7 Identyfikacja wind w systemie wysyłkowym* 23](#_Toc11938938)

[*Rysunek 1‑8 Nowa generacja mechanizmów wind GEN2 LIFT firmy OTIS* 24](#_Toc11938939)

[*Rysunek 2‑1 Schemat projektowania układu ASIC* 26](#_Toc11938940)

[*Rysunek 2‑2 Kod dla resetowalnych przycisków* 29](#_Toc11938941)

[*Rysunek 2‑3* 31](#_Toc11938942)

[*Rysunek 3‑1 Wygląd płytki bazowej MAXimator* 33](#_Toc11938943)

[*Rysunek 3‑2 Architektura syntezera* 34](#_Toc11938944)

[*Rysunek 3‑3 Blok bezpośredniej syntezy cyfrowej* 35](#_Toc11938945)

[*Rysunek 3‑4 Schemat blokowy oscylatora DDS* 35](#_Toc11938946)

[*Rysunek 3‑5 Koło fazy cyfrowej* 36](#_Toc11938947)

[*Rysunek 3‑6 Analiza czterokrotnego pomnożenia kroku* 36](#_Toc11938948)

[*Rysunek 3‑7 Generacja sinusoidy* 36](#_Toc11938949)

[*Rysunek 3‑8 Generacja sinusoidy z wykorzystaniem tablicy* 37](#_Toc11938950)

[*Rysunek 3‑9 Blok miksera* 37](#_Toc11938951)

[*Rysunek 3‑10 Sygnały wygenerowane w Audacity* 38](#_Toc11938952)

# Spis tabel

[***Tabela 1 Przyporządkowane piny do sygnałów* 25**](#_Toc502937456)

[***Tabela 2 Tablica częstotliwości dla piątej oktawy* 27**](#_Toc502937457)

1. Norma energtyczna [↑](#footnote-ref-1)