

WOJSKOWA AKADEMIA TECHNICZNA
im. Jarosława Dąbrowskiego

WYDZIAŁ ELEKTRONIKI



ROZPRAWA DOKTORSKA

mgr Maciej GURSKI

*„Wysokorozdzielczy system pomiaru odcinka czasu
wykorzystujący wielosegmentowe linie opóźniające
o kontrolowanych charakterystykach”*

Promotor: prof. dr hab. inż. Ryszard Szplet

Promotor pomocniczy: dr Robert Frankowski

WARSZAWA 2024

Podziękowania

Pragnę złożyć serdeczne podziękowania wszystkim, dzięki którym powstała ta praca szczególnie:

Panu profesorowi dr. hab. inż. Ryszardowi Szpletowi za nieocenioną pomoc udzieloną w trakcie przygotowywania rozprawy doktorskiej, cierpliwość, wyrozumiałość i cenne wskazówki.

Oraz dr. Robertowi Frankowskiemu za cenne wskazówki i liczne dyskusje oraz poświęcony czas.

Słowa podziękowania należą się także mojej Rodzinie, a szczególnie żonie Annie za motywację i cierpliwość.

*Pamięci
prof. dr. hab. inż. Marka Zielińskiego*

Spis treści

Wykaz ważniejszych skrótów i oznaczeń.....	4
1. Wstęp.....	6
2. Analiza stanu wiedzy.....	9
2.1 Cyfrowe linie opóźniające.....	9
2.1.1 Wielosegmentowe linie opóźniające.....	10
2.1.2 Niezależne wielokrotne linie opóźniające.....	12
2.1.3 Różnicowa linia opóźniająca.....	16
2.1.4 Ekwiwalentne linie opóźniające.....	16
2.2 Metody precyzyjnego pomiaru odcinka czasu.....	17
2.2.1 Metoda licznikowa.....	18
2.2.2 Metoda noniusza elektronicznego z dwoma startowalnymi oscylatorami (chronotron). 18	
2.2.3 Metoda sukcesywnej aproksymacji.....	19
2.2.4 Metoda skracania impulsu.....	20
2.2.5 Metoda kodowania wielokrotnego.....	21
2.2.6 Metoda przetwarzania z użyciem linii opóźniającej z odczepami.....	22
2.2.7 Metody przetwarzania wielostopniowego.....	24
3. Cel i tezy rozprawy.....	27
4. System pomiaru odcinka czasu o kontrolowanych charakterystykach.....	28
4.1 Platforma sprzętowa.....	28
4.1.1 Karta prototypowa KCU105.....	28
4.1.2 Układ programowalny Kintex UltraScale.....	30
4.2 Projekt systemu pomiarowego.....	32
4.2.1 Mikroprocesor MicroBlaze.....	34
4.2.2 Magistrala AXI.....	37
4.2.3 Wielosegmentowe linie opóźniające.....	38
4.2.4 Moduł przetwarzania i akwizycji danych.....	43
4.2.5 Matryca kolejności segmentów.....	47
4.2.6 Liczniki okresów.....	47
4.2.7 Moduły pomocnicze.....	48
4.3 Procedura sortowania segmentów.....	48
4.3.1 Metody minimalizacji błędu bąbelkowego.....	48
4.3.2 Przebieg procesu sortowania segmentów.....	53
5. Algorytm doboru wielokrotnych linii opóźniających.....	58
5.1 Kryteria wyboru wielokrotnych linii opóźniających.....	60
5.1.1 System z jednym kanałem pomiarowym.....	64
5.1.2 System wielokanałowy.....	65
5.1.3 System z rekonfigurowalnymi kanałami pomiarowymi.....	67
5.2 Modyfikacja charakterystyki wielosegmentowej linii opóźniającej.....	71
5.2 Implementacja algorytmu.....	73
5.3 Wyniki badań testowych.....	76
5.3.1 System jednokanałowy.....	80
5.3.2 System wielokanałowy.....	82
5.3.3 System z rekonfigurowalnymi kanałami pomiarowymi.....	85
6. Weryfikacja eksperymentalna algorytmu.....	87
7. Podsumowanie.....	110
Literatura.....	113

Wykaz ważniejszych skrótów i oznaczeń

ADC	– (ang. Analog to Digital Converter) – przetwornik analogowo-cyfrowy
ASIC	– (ang. Application Specific Integrated Circuit) – scalony układ specjalizowany
AXI	– (ang. Advanced eXtensible Interface) – magistrala zgodna z magistralami używanymi w mikroprocesorach ARM
BRAM	– (ang. Block RAM) – blok funkcjonalny pamięci RAM w układzie FPGA
CARRY	– (ang. carry chain logic) – elementy logiki (szybkich) przeniesień arytmetycznych
CLB	– (ang. Configurable Logic Block) – konfigurowalny blok logiczny układu FPGA
CMT	– (ang. Clock Management Tile) – blok funkcjonalny do zarządzania sygnałem zegarowym w układzie FPGA
CPLD	– (ang. Complex Programmable Logic Device) – złożone programowalne układy logiczne
DDR SDRAM	– (ang. Double Data Rate Synchronous Dynamic Random-Access Memory) – rodzaj synchronicznej pamięci dynamicznej RAM, w której dane transmitowane są zarówno wraz ze zboczem narastającym, jak i opadającym sygnału zegarowego
DNL	– (ang. Differential Nonlinearity) – błąd nieliniowości różnicowej
DSPLL	– (ang. Digital Signal Processing Phase Locked Loop) – technika syntezy częstotliwości łącząca technikę PLL z cyfrowymi filtrami
EDA	– (ang. Electronic Design Automation) – kategoria narzędzi programowych do projektowania systemów elektronicznych, takich jak układy scalone i obwody drukowane
INL	– (ang. Integrated Nonlinearity) – błąd nieliniowości sumacyjnej
FMC	– (ang. FPGA Mezzanine Card) – interfejs umożliwiający podłączenie różnych modułów i kart rozszerzeń do płyty z układem FPGA w celu uzyskania dodatkowej funkcjonalności
FPGA	– (ang. Field-Programmable Gate Array) – programowalna matryca bramkowa
GTH	– (ang. Gigabit Transceiver type H) nadajnikoodbiorniki w układach UltraScale, umożliwiające transmisję szeregową z szybkością 16,3 Gb/s
GWLO	– grzebieniowa wielosegmentowa linia opóźniająca
LUT	– (ang. Look-Up Table) – tablica wartości funkcji logicznych w układach FPGA
MicroBlaze	– mikroprocesor wirtualny typu RISC, opracowany przez firmę AMD/Xilinx, implementowany w układach programowalnych tej firmy
MMCM	– (ang. Mixed-Mode Clock Manager) – blok funkcjonalny układu FPGA służący do generowania wielu zegarów ze zdefiniowanymi zależnościami fazowymi i częstotliwościowymi
MMU	– (ang. Memory Management Unit) – jednostka zarządzania pamięcią
PLL	– (ang. Phase Locked Loop) – pętla synchronizacji fazowej
pASIC	– seria układów scalonych, programowalnych jednorazowo firmy QuickLogic
RISC	– (ang. Reduced Instruction Set Computing) – mikroprocesor o zredukowanej liczbie instrukcji
RMS	– (ang. Root Mean Square) – średnia kwadratowa
SFP	– (ang. Small-form Factor Pluggable) – gniazda dla modułów komunikacyjnych światłowodowych lub kabla miedzianego

TDC	– (ang. Time-to-Digital Converter) – przetwornik czasowo-cyfrowy
TOF	– (and. Time of Flight) – pomiar czasu przelotu
TOFMS	– (and. Time of Flight mass spectrometry) – spektrometria masowa czasu przelotu
VHDL	– (ang. Very high speed integrated circuits Hardware Description Language) – język opisu sprzętu
WLO	– wielosegmentowa linia opóźniająca
ZWLO	– złożona wielosegmentowa linia opóźniająca
<i>clk</i>	– zegar referencyjny
<i>f</i>	– częstotliwość
<i>HIT</i>	– impuls reprezentujący zdarzenie
<i>Lk</i>	– liczba kombinacji
<i>T</i>	– czas
<i>T_{clk}</i>	– okres zegara referencyjnego
<i>T_m</i>	– mierzony odcinek czasu
<i>T_n</i>	– odcinek czasu mierzony metodą licznikową
<i>q</i>	– rozdzielczość, przedział kwantyzacji
<i>q_{eqv}</i>	– rozdzielczość ekwiwalentna
<i>σ</i>	– odchylenie standardowe

1. Wstęp

Obserwowany w ostatnich dziesięcioleciach dynamiczny rozwój techniki oraz metod badawczych spowodował między innymi wzrost zainteresowania precyzyjnym pomiarem odcinka czasu upływającego pomiędzy zdarzeniami fizycznymi, a urządzenia mierzące tenże odcinek znajdują coraz powszechniejsze zastosowanie w różnych obszarach nauki i techniki. Przykładowo w badaniach w zakresie fizyki jądrowej towarzyszą wykrywaniu i identyfikacji cząstek, badaniu struktury materii poprzez rozpraszanie cząstek (CERN) [1] oraz do zliczania fotonów [2, 3]. Stosowane są w spektrometrii masowej czasu przelotu (TOFMS) [4], służącej do określenia składu izotopowego pierwiastków oraz w badaniach czasu życia stanów wzbudzonych atomów. Przykładami kolejnych zastosowań są określanie pozycji w systemach nawigacyjnych [5] i kwantowa transmisja danych [6]. Wśród powszechnych już zastosowań precyzyjnych pomiarów odcinka czasu wymienić należy pomiary odległości za pomocą odbitej wiązki ultradźwięków (TOF), fal radiowych albo światła lasera [7]. W przemyśle są to również pomiary intensywności przepływu cieczy i gazów realizowane z użyciem przepływomierzy ultradźwiękowych [8]. Z kolei w medycynie układy pomiaru odcinka czasu stosowane są w pozytonowej emisyjnej tomografii komputerowej (PET) [9]. Inne spektakularne zastosowanie to ultra szybkie kamery o bardzo wysokich częstotliwościach rejestracji klatek rzędu $10^{12}/s$, które umożliwiają zarejestrowanie poruszającego się pojedynczego fotonu [10, 11] i odtworzenie jego ruchu w zwolnionym tempie. Pomiary odcinka czasu od wielu lat mają zastosowania również w telekomunikacji, głównie w pomiarach fluktuacji fazowych sygnałów zegarowych i sygnałów danych [12], co ma istotne znaczenie w świetle coraz większych szybkości transmisji danych.

Od urządzeń pomiaru odcinka czasu wymaga się jednocześnie coraz wyższej precyzji (rzędu pojedynczych pikosekund) i stosunkowo szerokiego zakresu pomiarowego (rzędu kilkudziesięciu sekund i więcej). System pomiaru odcinka czasu powinien charakteryzować się również krótkim czasem martwym, a układ pomiarowy powinien akceptować impulsy o dużej intensywności. Jest to szczególnie ważne w systemach pomiarowych stosowanych w eksperymentach fizycznych. Niektóre zastosowania wymagają przy tym wielu kanałów pomiarowych. Dostępne komercyjnie urządzenia często nie spełniają wymagań odnośnie precyzji i zakresu pomiarowego lub posiadają ograniczoną liczbę kanałów pomiarowych. Dlatego większość urządzeń do precyzyjnego pomiaru odcinka czasu opracowywana jest na potrzeby konkretnego eksperymentu badawczego lub zastosowania.

Precyzyjne systemy pomiaru odcinka czasu posiadają najczęściej budowę modułową. Do ich realizacji stosuje się przeważnie układy specjalizowane lub programowalne wykonane w najnowszych technologiach mikroelektronicznych. Zapewniają one elementy i bloki funkcjonalne charakteryzujące się krótkimi czasami propagacji, co jest istotne z punktu widzenia rozdzielczości projektowanego systemu. Urządzenie do pomiaru odcinka czasu może być zaprojektowane i wykonane jako układ specjalizowany ASIC (Application-Specific Integrated Circuit), czyli układ scalony wykonany w pełnym cyklu projektowo-wytwórczym (*full-custom*). Prototypowanie w tej technologii gwarantuje dużą kontrolę nad opracowywanym układem, począwszy od rozmiarów tranzystorów, co wpływa na gęstość upakowania, poprzez rozmieszczenie elementów na podłożu, co silnie określa opóźnienia w rozprowadzaniu sygnałów. Jednakże odbywa się to kosztem wydłużonego czasu

prototypowania i weryfikacji. Pierwsze przetworniki czasowo-cyfrowe w tej technice zostały wykonane w technologii CMOS o rozmiarze charakterystycznym $2,5 \mu\text{m}$ [13].

Alternatywą do układów ASIC techniką realizacji scalonych urządzeń do pomiaru odcinka czasu jest zastosowanie układów programowalnych, w szczególności matryc bramkowych FPGA (Field-Programmable Gate Array). Pierwszy scalony licznik czasu i częstotliwości z użyciem układu FPGA opracowano w Wojskowej Akademii Technicznej w 1997 [14]. W liczniku tym uzyskano rozdzielczości 200 ps i niepewność pomiarową nie gorszą niż 150 ps. Zastosowane układy pASIC firmy Quicklogic nie wymagają wstępnej rekonfiguracji i są gotowe do użycia po załączeniu napięcia zasilania. Wadą układów pASIC jest możliwość jedynie jednokrotnego programowania, co znacznie komplikuje i podraża proces prototypowania.

Na przestrzeni ostatnich 30-stu lat w układach pomiaru odcinka czasu dokonał się znaczący postęp. Przede wszystkim układy te zwiększyły rozdzielczość pomiaru z nanosekund do pikosekund. Rozwój ten zawdzięczają nowszym technologiom produkcji układów scalonych, w tym nowej technologii tranzystorów 3D FinFET [15, 16, 17, 18, 19]. Rozwój technologii mikroelektronicznych zapewniając coraz szybsze działanie elementów umożliwia poprawę parametrów systemów pomiaru odcinka czasu. Przykładowo linie opóźniające zaimplementowane w układach FPGA FLEX10KE firmy Altera, wytwarzanych w latach 1998 – 2002 w technologii $0,22 \mu\text{m}$, osiągały rozdzielczość rzędu $0,5 \text{ ns}$ [20]. Natomiast linie opóźniające zaimplementowane w nowszych układach FPGA UltraScale firmy AMD/Xilinx, wykonywanych w technologii 20 nm , posiadają rozdzielczość rzędu pojedynczych pikosekund [21]. Firma TSMC (Taiwan Semiconductor Manufacturing Company), produkująca układy scalone dla firmy Xilinx, wytwarza obecnie układy scalone w technologii 5 nm (układy System On Chip do telefonów komórkowych i kart graficznych). Prognozowane jest przejście na technologię 2 nm w ciągu najbliższych kilku lat. Z coraz mniejszymi rozmiarami tranzystorów wiąże się zmniejszenie pojemności pasożytniczych, prowadzące do skracania czasów propagacji sygnałów i skutkujące projektami układów i systemów o coraz większej rozdzielczości i szybkości działania. Kolejnymi pozytywnymi efektami miniaturyzacji są też zmniejszenie mocy strat oraz ograniczenie efektu metastabilności.

W systemach pomiaru odcinka czasu, które są implementowane w układach FPGA, stosuje się na ogół przetworniki czasowo-cyfrowe oparte na wielosegmentowych liniach opóźniających, wykorzystujących łańcuchy szybkich przeniesień arytmetycznych. Układy FPGA Kintex UltraScale (AMD/Xilinx) posiadają duże zasoby logiczne i funkcjonalne oraz pracują z wysokimi częstotliwościami, co umożliwia budowę wielosegmentowych linii opóźniających (WLO) o rozdzielczości pojedynczych pikosekund. Niestety, charakterystyki przetworników opartych na WLO wykazują silnie nieliniowości. Ponadto, zaobserwowano istotny wpływ błędu bąbelkowego, co stanowi dodatkową trudność w procesie dekodowania fazy zegara wzorcowego. Dalsza poprawa parametrów systemu pomiaru odcinka czasu jest możliwa między innymi poprzez zastosowanie optymalnie dobranych zwielokrotnionych wielosegmentowych linii opóźniających (ZWLO). Zmienia to zasadniczo podejście w procesie projektowania ZWLO w tych strukturach.

W niniejszej pracy przedstawiono projekt systemu pomiaru odcinka czasu zbudowanego z użyciem wielu wielosegmentowych linii opóźniających. Układ pomiarowy wykorzystuje interpolacyjną metodę Nutta z 16 przetwornikami jednostopniowymi (flash) opartymi na WLO. System został zaimplementowany w układzie programowalnym Kintex UltraScale firmy AMD/Xilinx.

W celu uzyskania możliwie wysokich parametrów metrologicznych opracowano dedykowany algorytm wyboru WLO spośród dostępnego nadmiarowego zbioru linii zaimplementowanych w strukturze FPGA.

Rozprawa jest podzielona na siedem rozdziałów. W rozdziale drugim dokonano przeglądu metod i technik precyzyjnego pomiaru odcinka czasu oraz opisano architekturę wielosegmentowych linii opóźniających.

Cel i tezy rozprawy zawarte są w rozdziale trzecim. Przedstawiono w nim także założenia projektowe i zadania badawcze, których realizacja jest niezbędna do osiągnięcia celu rozprawy.

W rozdziale czwartym opisano budowę i działanie opracowanego systemu pomiaru odcinka czasu zaimplementowanego w układzie programowalnym Kintex UltraScale. Ponadto opisano procedury sortowania segmentów wielosegmentowych linii opóźniających.

W rozdziale piątym zawarto zaproponowane kryteria oraz algorytm selekcji wielosegmentowych linii opóźniających, wybieranych do systemu pomiaru odcinka czasu w celu uzyskania parametrów systemu spełniających przyjęte założenia.

Rozdział szósty zawiera wyniki badań eksperymentalnych opracowanego systemu pomiaru odcinka czasu potwierdzające jego prawidłowe działanie i poprawność przyjętej koncepcji.

Rozdział siódmy stanowi podsumowanie rozprawy.

2. Analiza stanu wiedzy

Produkcja programowalnych układów scalonych rozpoczęła się w latach 70-tych ubiegłego wieku. Największy obecnie producent tych układów firma Xilinx (aktualnie AMD/Xilinx), swoje pierwsze programowalne matryce bramkowe FPGA (Field-Programmable Gate Array) wprowadziła na rynek w 1985 roku. W 1992 roku firma Lattice Semiconductor rozpoczęła produkcję programowalnych w systemie układów CPLD (Complex Programmable Logic Device) isp. Od 1997 nastąpił gwałtowny rozwój rynku układów programowalnych spowodowany rozwojem języków opisu sprzętu (VHDL i Verilog) oraz oprogramowania EDA (Electronic Design Automation). Jedną z cech charakterystycznych układów programowalnych jest regularne rozmieszczenie bloków logicznych, co predestynuje te układy do budowy wielosegmentowych linii opóźniających (WLO). W typowych rozwiązaniach linie opóźniające są stosowane w procesie przetwarzania czasowo-cyfrowego a ich własności decydują o parametrach metrologicznych przetwornika. Dodatkowo wbudowane bloki funkcjonalne są przydatne w procesie akwizycji i przetwarzania danych [22]. Najnowsze rozwiązania układowe w matrycach FPGA powodują, że są one obecnie wyraźnie wygodniejszą alternatywą, względem układów ASIC, z powodu znacznie niższych nakładów finansowych i czasowych, koniecznych do opracowania układu przetwornika.

Do pomiarów stosunkowo krótkich odcinków czasu stosuje się typowo metody oparte na użyciu wielosegmentowych linii opóźniających o różnych konfiguracjach, opisanych w dalszej części niniejszego rozdziału. W rozdziale tym opisane są także metody pozwalające na zwiększenie zakresu pomiarowego przy zachowaniu wysokiej rozdzielczości oraz przedstawiona jest analiza błędów związanych z pomiarem odcinka czasu.

2.1 Cyfrowe linie opóźniające

W wielu pomiarach fizycznych ważne jest precyzyjne określenie czasu zajścia zdarzeń o charakterze losowym oraz ich wzajemnych relacji [23]. Już w latach dwudziestych ubiegłego wieku stosowano do tego celu ideę linii opóźniającej [24] i efekt niezerowego czasu propagacji prądu lub drgań (dźwięku) w różnych ośrodkach. Najczęściej do eksperymentów stosowano odpowiednio nawinięte i podłączone cewki, które symulowały linię długą [25]. Dla większych opóźnień stosowano ultradźwięki i różne ośrodki propagacji, takie jak kwarc i rtęć. Początkowo linie opóźniające służyły do przesyłania sygnałów analogowych. Stosowane były w radarach (do pamiętania stałych obiektów naziemnych), do dekodowania koloru w systemach telewizji kolorowej, a także w układach oscyloskopów analogowych w celu odpowiedniego wyzwolenia oscyloskopu, aby właściwie zobrazować przebieg sygnału. Stosowane były również jako elementy pamięciowe tj. rejestry szeregowo dla pierwszych komputerów. W dobie powszechnego stosowania układów cyfrowych, zaczęto linie opóźniające budować z elementów logiki cyfrowej, które były łatwiejsze w wykonaniu i mniej problematyczne w użyciu (mniejsze wymiary, waga, wrażliwość parametrów na warunki zewnętrzne). Elementy cyfrowej linii opóźniającej, będące z reguły układami aktywnymi, w sposób „naturalny” replikują sygnał i dlatego sygnały w takich liniach nie wymagają regeneratorów. Zatem, teoretycznie, linie budowane w układach cyfrowych mogą być dowolnie długie. Poniżej opisane są cyfrowe linie opóźniające implementowane w aktualnie produkowanych układach FPGA.

2.1.1 Wielosegmentowe linie opóźniające

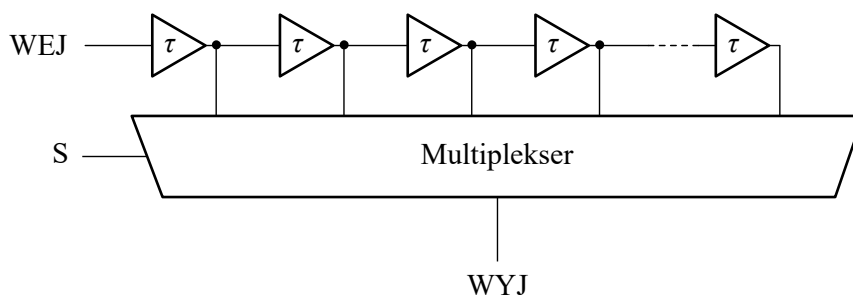
Cyfrowe linie opóźniające służą do dyskretnego opóźniania sygnału adekwatnie do czasów propagacji wprowadzanych przez zasoby logiczne układu, wykorzystane w procesie projektowania linii. Stosowane zasoby logiczne obejmują elementy techniki cyfrowej. Mogą to być bufory, inwertery, bramki cyfrowe lub ich logiczne odpowiedniki (np. LUT, Look-Up Table w układach FPGA), a także odpowiednio połączone zatrzaski.

W przypadku cyfrowych linii opóźniających najczęściej wykorzystuje się szeregowo połączone bramki cyfrowe [26, 27, 28]. Typowo czasy propagacji bramek logicznych określa się dla dwóch przypadków, a mianowicie jako czas przejścia ze stanu wysokiego do niskiego (tp_{HL}) oraz jako czas przejścia ze stanu niskiego do wysokiego (tp_{LH}). Czasy te na ogół się różnią, więc stosuje się ich wartość średnią [27]:

$$t_p = \frac{tp_{HL} + tp_{LH}}{2} \quad (1)$$

Dla przykładu w układach scalonych CMOS wykonanych w technologii 180 nm czas ten dla inwertera wynosi 26 ps [29], a dla technologii 45 nm jest równy 19 ps [30]. Dla aktywnych elementów linii opóźniającej (inwerterów) w układach scalonych ASIC, czas propagacji zależy od użytej technologii oraz od zamysłu projektanta, który może teoretycznie dowolnie zmieniać rozmiary użytych tranzystorów oraz stosować pojemności obciążające. Przykładowo dla technologii 45 nm czas propagacji zmieni się z 19 ps na 33 ps po zmianie szerokość kanału tranzystora z 400 nm na 100 nm [30], zaś wtrącenie na wyjściu bramek dodatkowej pojemności o wartości 15 fF zmieni ten czas odpowiednio na 120 ps dla tranzystora o szerokości bramki 400 nm oraz 471 ps dla tranzystora o szerokości bramki 100 nm.

Linie opóźniające zbudowane są zazwyczaj z jednakowych elementów elektronicznych połączonych kaskadowo. Jeśli w łańcuchu każdy element opóźniający ma wyprowadzenie poza linię to taką linię nazywa się linią wielosegmentową z odczepami (Tapped Delay Line). Jeśli natomiast linia posiada tylko jedno wyjście, a wartość opóźnienia można zmieniać, to jest ona linią programowalną. Przykładowy schemat linii programowalnej o regulowanym opóźnieniu pokazany jest na rys. 1.



Rys. 1. Schemat linii opóźniającej o regulowanym opóźnieniu

Linie opóźniające w układach programowalnych na ogół buduje się z elementów typu bufory, bramki logiczne, zatrzaski, a w bardziej złożonych matrycach programowalnych FPGA dodatko-

wo w tym celu można użyć elementów wbudowanych takich jak tablice LUT, łańcuchy szybkich przeniesień arytmetycznych oraz ścieżki do rozprowadzania sygnałów zegarowych lub ścieżki połączeniowe pomiędzy blokami logicznymi CLB [14].

W niektórych układach programowalnych znajdują się również wbudowane linie opóźniające, lecz mają one ograniczoną funkcjonalność. Występują jako części składowe bloków funkcjonalnych, np. układów generacji zegara, a także regulacji i wyrównywania opóźnień w portach wejścia/wyjścia do obsługi zewnętrznej pamięci DDR [16]. W układach firmy Xilinx (obecnie AMD) takim komponentem jest przykładowo IDELAYE3, tj. programowalna linia opóźniająca sygnał wejściowy w 512 krokach, o zakresie 1,25 ns (dla układów serii Kintex UltraScale) [15].

Przykłady implementacji linii opóźniających w układach CPLD opisano w pracy [31], w której m.in. w układzie ispLSI2128 firmy Lattice Semiconductor zrealizowano 10 bitową linię opóźniającą o rozdzielczości 200 ps. Układy CPLD są mniej elastyczne niż układy FPGA z uwagi na globalną matrycę połączeń, która wymusza realizację połączeń o większych długościach, co negatywnie wpływa na rozdzielczość linii opóźniających. Zazwyczaj oprogramowanie, które służy do projektowania układów CPLD nie zapewnia również takiej elastyczności projektowej jak oprogramowanie dla matryc FPGA, które umożliwia dowolną alokację elementów implementowanego układu i stosunkowo swobodne prowadzenie ścieżek połączeniowych.

Zastosowanie układów FPGA, w porównaniu z układami ASIC, wiąże się z pewnymi ograniczeniami, spowodowanymi matrycową budową tych pierwszych. Na przykład w przypadku użycia łańcucha przeniesień arytmetycznych w układach firmy Xilinx możliwa jest budowa wielosegmentowej linii opóźniającej jedynie w ściśle określony sposób. Konfiguracja łańcucha CARRY jest bowiem niemodyfikowalna. W celu osiągnięcia minimalnego opóźnienia w takiej linii, konieczne jest bezpośrednie połączenie kolejnych komponentów łańcucha CARRY, tak aby wyjście jednego bezpośrednio łączyło się z wejściem następnego. Jednakże kierunek propagacji sygnału w takiej linii możliwy jest tylko w stronę zgodną z rosnącą numeracją bloków CLB, co jest zgodne z kierunkiem połączeń buforów. Oznacza to, że nie jest możliwe zbudowanie linii opóźniającej opartej tylko na łańcuchu CARRY, w taki sposób, aby kolejno podłączone segmenty miały odczepy w kierunku malejącego indeksu (w kierunku umownego dołu układu). Podobnie niemożliwe jest skonstruowanie „poziomej” linii opóźniającej bazującej na łańcuchu CARRY.

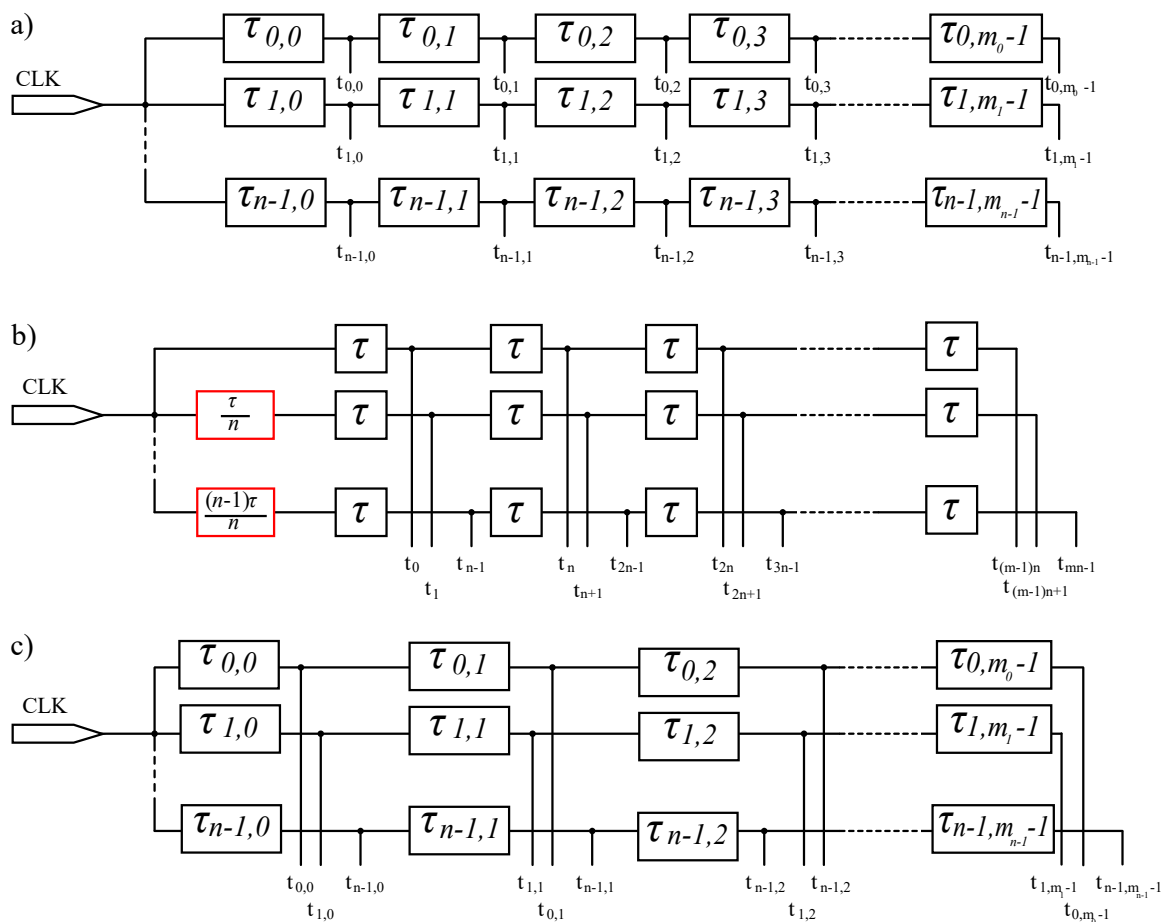
Teoretycznie jest możliwe dowolne połączenie poszczególnych elementów wykorzystanych w linii opóźniającej, jednak wiąże się to z użyciem dodatkowych połączeń pomiędzy elementami/komórkami, zamiast dedykowanych najkrótszych (najszybszych), co powoduje znaczne zwiększenie opóźnień międzyelementowych. W efekcie rozdzielczość takiej linii opóźniającej jest wielokrotnie mniejsza. Jednak wykorzystując do budowy WLO inne elementy FPGA, np. tablice LUT, poprzez zmianę kolejnych elementów w układzie można manipulować opóźnieniami poszczególnych segmentów celem zwiększenia liniowości [32, 33].

Wraz ze wzrostem długości WLO pogarsza się jej liniowość, a w konsekwencji precyzja przetwarzania, dlatego wskazane jest skracanie długości linii [34]. Dodatkowo w większych układach FPGA znaczenie ma budowa matrycowa tych układów i podział na domeny zegarowe, co wiąże się z niemodyfikowalnym rozprowadzeniem sygnałów zegarowych i właściwymi im opóźnieniami.

Linie opóźniające zaimplementowane w układzie FPGA wykorzystują wybrane elementy logiczne o zdefiniowanych czasach propagacji charakteryzujące się losowymi rozrzutami parametrów czasowych. W celu poprawy rozdzielczości oraz liniowości linii opóźniających stosuje się architektury przedstawione w kolejnych podrozdziałach pracy.

2.1.2 Niezależne wielokrotne linie opóźniające

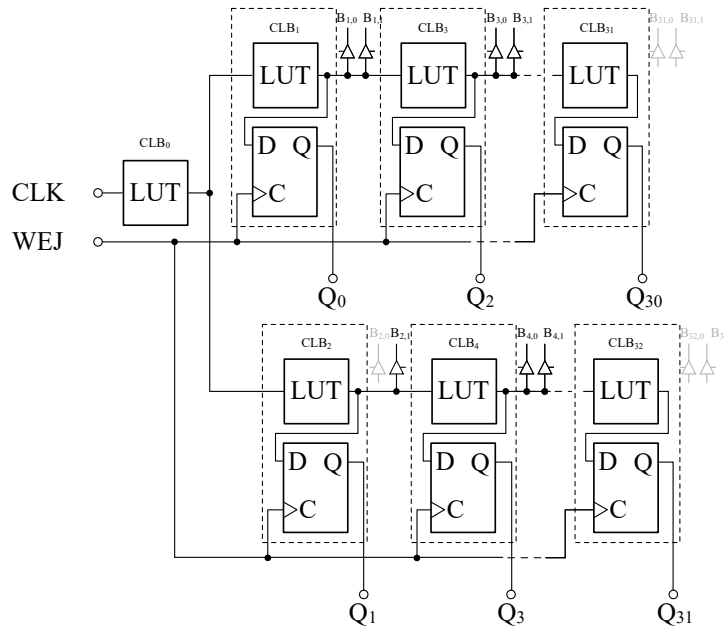
Celem zwiększenia rozdzielczości pojedynczych linii opóźniających budowanych w układach programowalnych stosuje się konstrukcje niezależnych wielokrotnych linii opóźniających WLO [35]. Przykładowe układy takich linii pokazane są na rys. 2 [21, 37].



Rys. 2. Przykładowe konfiguracje WLO w systemie: a) wiele niezależnych linii opóźniających, b) idealne linie opóźniające z dodanymi opóźnieniami wejściowymi w celu zwiększenia rozdzielczości, c) równoległe linie opóźniające o nieidealnych charakterystykach, użyte do budowy jednej linii opóźniającej o korzystniejszej charakterystyce poprzez wybór kolejności odczepów

W przypadku zastosowania samego uśrednienia wyników odczytanych z wielu WLO w celu poprawy jakości przetwarzania, odchylenie standardowe pomiaru maleje wraz z pierwiastkiem z liczby użytych WLO. W artykule [36] opisano zastosowanie w układzie ASIC 128 par WLO (osobno dla impulsu START i STOP) w celu uśrednienia wyniku pojedynczego pomiaru. W efekcie stosując li-

nie opóźniające o 64 odczepach i średniej rozdzielczości 33 ps (przy $\sigma_{\text{rms}}=42,0$ ps dla pary WLO START oraz STOP), uzyskano precyzję pomiaru $\sigma_{\text{rms}}=3,0$ ps.



Rys. 3. Schemat podwójnej WLO zaimplementowanej w układzie Virtex XCV300 [37]

Linie opóźniające implementowane w układzie FPGA Virtex XCV300 firmy Xilinx bazujące na tablicach LUT, zapewniały rozdzielczość wielosegmentowych linii opóźniającej w zakresie od 200 ps do 1 ns [37]. Jednakże użycie w tym układzie łańcucha przeniesień arytmetycznych umożliwiało uzyskanie linii opóźniających o rozdzielczości lepszej niż 50 ps [38]. Z kolei linię o rozdzielczości 500 ps uzyskano w wyniku wirtualnego „złożenia” dwóch linii o rozdzielczości 1 ns (rys. 3). Wykorzystano różnicę w opóźnieniu sygnałów wychodzących z tablicy LUT, mieszczącej się w bloku CLB_0 i sygnałów na wejściach LUT bloków CLB_1 i CLB_2 (rys. 3). Położenie bloku CLB_0 zostało ustalone eksperymentalnie tak, aby względne opóźnienie tych sygnałów wynosiło połowę opóźnienia jednej WLO. W celu zlinearyzowania charakterystyki WLO, podłączano od jednego do kilku buforów trójstanowych $B_{i,0}$, $B_{i,1}$ [39] do wyjścia informacyjnego CLB, co powodowało zwiększenie stałej czasowej danego wyjścia i zwiększanie czasu narastania zbocza sygnału wyjściowego, a w efekcie zmianę charakterystyki całej WLO na bardziej liniową. Podobny efekt uzyskuje się w przypadku linii poczwórnych o jeszcze większej rozdzielczości. W ten sposób w układzie FPGA Virtex XCV300 zaimplementowano linię o rozdzielczości 250 ps [37].

W strukturze układu FPGA Virtex względnie łatwo można zaprojektować linię opóźniającą o stosunkowo liniowej charakterystyce lecz o relatywnie dużym opóźnieniu pojedynczego segmentu. Zastosowanie WLO umożliwia utworzenie systemu o wysokiej rozdzielczości. W tym celu należy zaimplementować m linii o $n-1$ segmentach i opóźnieniu τ pojedynczego segmentu. Wejścia i -tej linii należy podłączyć przez opóźnienie τ_s według wzoru (2):

$$\tau_s = \frac{i}{m} \tau, \quad (2)$$

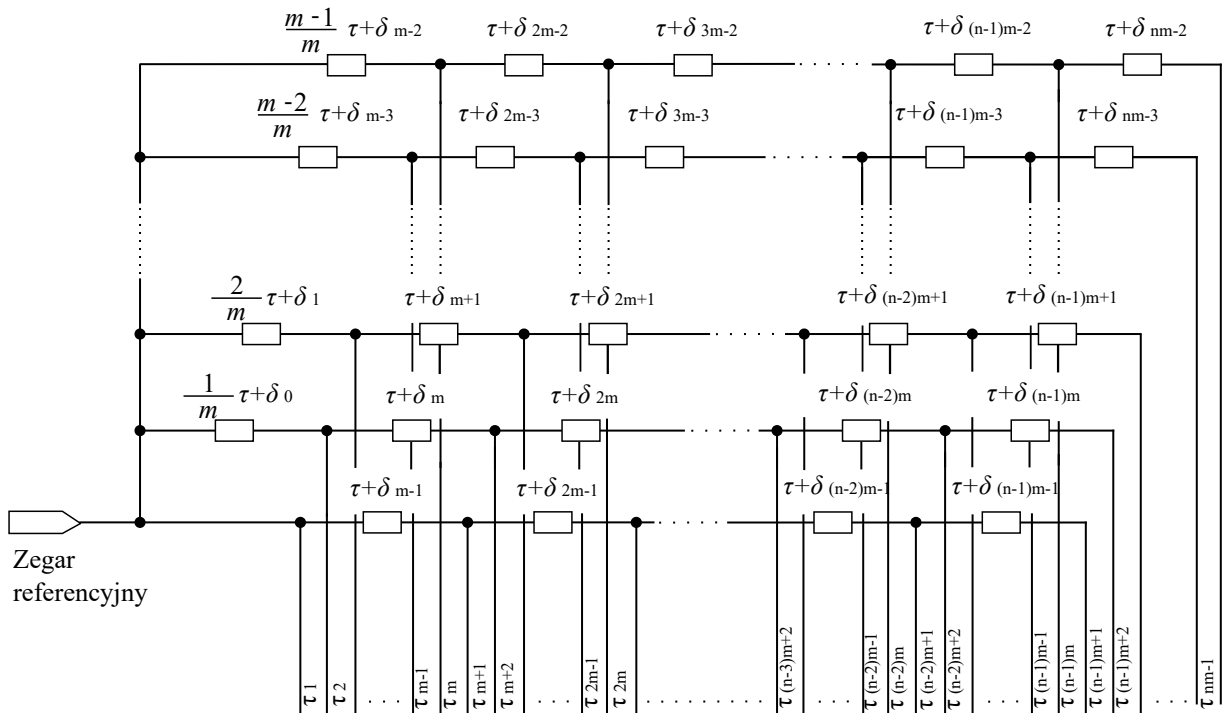
gdzie : $i \in \langle 0, m-1 \rangle$.

Uzyskuje się wówczas linię opóźniającą o czasie propagacji pojedynczego segmentu równym τ/m . Oznaczając odchylenie opóźnienia poszczególnych segmentów od wartości zadanych przez δ_i , gdzie $i \in \langle 0, nm-2 \rangle$, można przyjąć, że $\tau + \delta_i \in \langle m-1, nm-2 \rangle$ jest opóźnieniem segmentu $[(i+1-m)/m]$ (gdzie $[\]$ jest operatorem zwracającym część całkowitą), który znajduje się w linii o numerze n_i :

$$n_i = (i+1) \bmod m. \quad (3)$$

Wielokrotna linia opóźniająca jest charakteryzowana opóźnieniem wirtualnego pojedynczego i -tego segmentu, co w odniesieniu do rys. 4 [37] można zapisać następująco:

$$\tau_i = \frac{\tau}{m} \sum_{j=0}^{\lfloor \frac{i+m-2}{m} \rfloor + \lfloor \frac{i+m-1}{m} \rfloor} (-1)^j \delta_{(i-1-j \bmod 2) \bmod m + \lfloor \frac{j}{2} \rfloor m} \quad (4)$$



Rys. 4. Uogólniony model wielokrotnej WLO [37]

Dla podwójnej linii opóźniającej wzór (4) przyjmie postać:

$$\tau_i = \frac{\tau}{2} + \sum_{j=0}^{i-1} \delta_j (-1)^{j+i+1} \quad (5)$$

dla $i \in \langle 1, 2n-1 \rangle$.

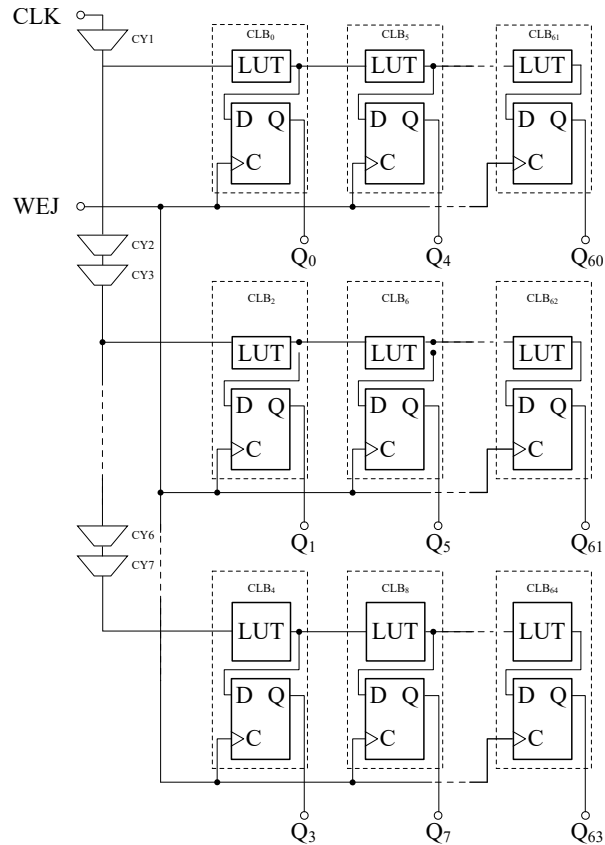
W sytuacji, gdy odchylenia δ_i , dla $i \in \langle 0, mn-2 \rangle$ z pojedynczego segmentu są porównywalne, wtedy trudno jest skonstruować wielokrotną linię opóźniającą. Wówczas należy zastosować metodę korekcji nieliniowości poszczególnych linii opóźniających, wchodzących w skład linii wielokrotnej.

Ważnym problemem w czasie implementacji wielokrotnej linii opóźniającej w układzie FPGA, jest skonstruowanie wysokorozdzielczej linii, która ma na celu rozproszczenie sygnału zegarowego do pozostałych linii składowych. Dla $m=2$ wystarczy wykorzystać dodatkowy blok CLB w układzie FPGA, który należy jednak odpowiednio ulokować. Dla większej liczby linii m , znalezienie takich

bloków nie jest proste i lepszą metodą jest zastosowanie elementów łańcucha przeniesień arytmetycznych znajdujących się w każdym bloku CLB. Zostało to pokazane na rys. 5. Dla poczwórnej wielosegmentowej linii opóźniającej, opóźnienie pojedynczego segmentu linii dane jest wzorem [37]:

$$\tau_i = \frac{\tau}{4} + \sum_{j=0}^{\left\lfloor \frac{i+2}{4} \right\rfloor + \left\lfloor \frac{i+3}{4} \right\rfloor - 1} (-1)^j \delta_{(i-1-j \bmod 2) \bmod 4 + 4 \left\lfloor \frac{j}{2} \right\rfloor}, \quad (6)$$

gdzie i oznacza numer wirtualnego segmentu oraz $i \in \langle 1, 4n-1 \rangle$.

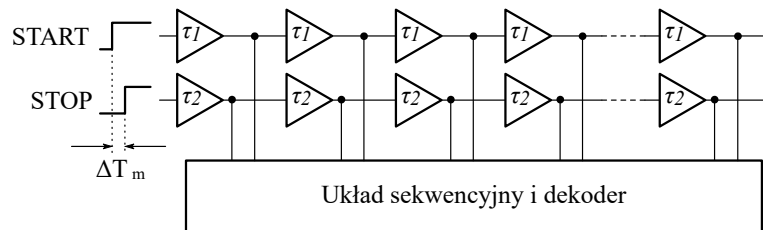


Rys. 5. Schemat poczwórnej WLO zaimplementowanej w układzie Virtex XCV300 [37]

W przypadku układów FPGA pojedyncze opóźnienie elementu łańcucha przeniesień arytmetycznych wynosi od kilku do nawet kilkuset ps. Parametry czasowe tego komponentu wykazują słabą zależność temperaturową w porównaniu do innych bloków logicznych. Dlatego elementy te stosuje się coraz częściej przy konstruowaniu WLO. W pracy [40] opisano linię opóźniającą wykonaną z użyciem łańcucha CARRY, która charakteryzowała się dużą nieliniowością. Segmenty o większym opóźnieniu przeplatają się w niej z segmentami o mniejszym opóźnieniu, a ich opóźnienia mają wartości od 30 ps do 110 ps. Duży rozrzut opóźnień elementów łańcucha przeniesień arytmetycznych stanowią poważną wadę WLO budowanych z użyciem łańcucha. Właściwość tę można jednak wykorzystać przy implementacji linii różnicowej.

2.1.3 Różnicowa linia opóźniająca

Różnicowa linia opóźniająca budowana jest z równoległych łańcuchów elementów cyfrowych, których opóźnienia jednostkowe są rzędu nanosekund. Natomiast różnica tych opóźnień wynosi od kilkuset do kilkudziesięciu pikosekund [14, 41, 42]. Interpolator zbudowany z użyciem różnicowej linii opóźniającej pokazany na rys. 6 umożliwia zwiększenie rozdzielczości interpolacji i uproszczenie układu pomiarowego.



Rys. 6. Schemat interpolatora z różnicową linią opóźniającą

W celu zapewnienia poprawności działania różnicowej linii opóźniającej musi być spełniona relacja $\tau_1 > \tau_2$. Wówczas zakres pomiarowy układu wynosi:

$$T_{max} = n(\tau_1 - \tau_2) = n \cdot \Delta\tau, \quad (7)$$

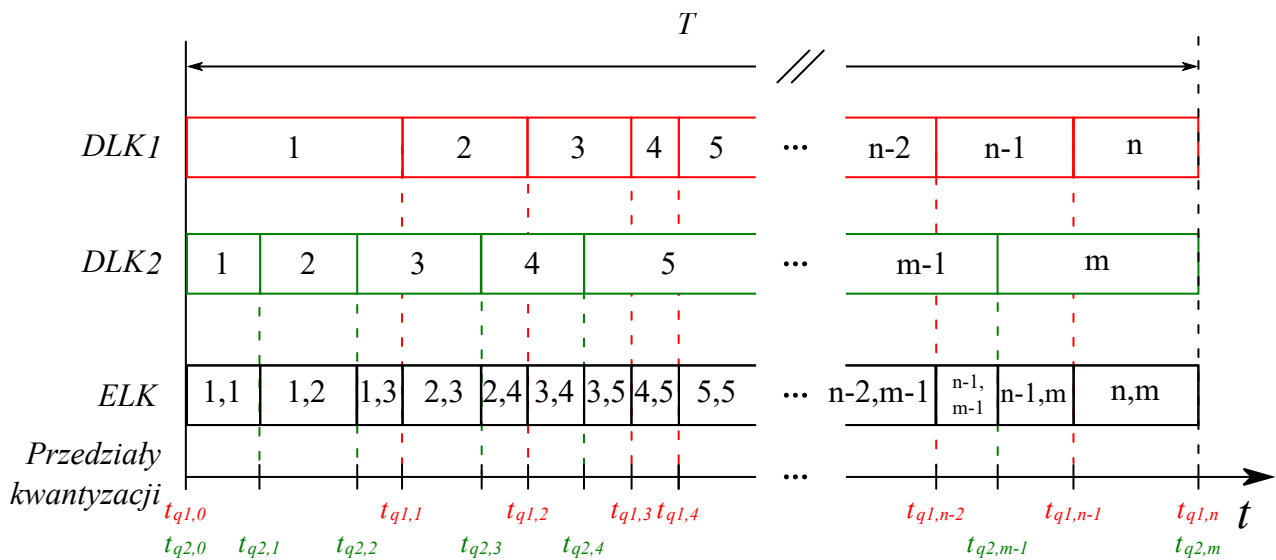
gdzie n oznacza liczbę elementów linii opóźniającej. Czas interpolacji jest obliczany ze wzoru:

$$T_{inter} = \frac{\Delta T_m}{(\tau_1 - \tau_2)} \tau_2. \quad (8)$$

Ze wzoru (8) wynika również, że jeśli układ pomiarowy posiada dużą rozdzielczość, tj. małą różnicę opóźnień $\Delta\tau$ oraz stosunkowo duże wartości τ_1 i τ_2 , to czas interpolacji silnie zależy od wartości mierzonego odcinka czasu. Wadą tego typu rozwiązań jest mały zakres pomiarowy, zależny od długości linii. Problem ten można wyeliminować, stosując metodę noniusza elektronicznego.

2.1.4 Ekwiwalentne linie opóźniające

Metoda ekwiwalentnej linii kodującej (ELK) została opracowana w Wojskowej Akademii Technicznej [43]. Metoda polega na użyciu przynajmniej dwóch WLO o różnych charakterystykach. Jest to typowa sytuacja w układach FPGA, z uwagi na różne opóźnienia dróg rozproszonych sygnałów, rozrzutów produkcyjnych, itp. Biorąc pod uwagę szerokości przedziałów kwantowania, a także ich względne przesunięcia, można stwierdzić, które kanały z różnych WLO „pokrywają się czasowo” całkowicie lub częściowo. W ten sposób można wyznaczyć wirtualne kanały o znacznie mniejszych szerokościach niż kanały z pierwotnych WLO, a ich liczba wynosi $n+m-1$, gdzie n i m to liczby kanałów odpowiednio pierwszej i drugiej WLO (rys. 7). Zastosowanie metody ELK w układzie Spartan-6 dla 16 WLO umożliwiło uzyskanie wysokiej rozdzielczości równej 1,2 ps [43].



Rys. 7. Zasada utworzenia ekwiwalentnej linii kodującej

2.2 Metody precyzyjnego pomiaru odcinka czasu

Z uwagi na zastosowaną krotność przetwarzania, metody precyzyjnego pomiaru odcinka czasu dzieli się na jednostopniowe i wielostopniowe [22]. Ze względu na użyte do ich budowy elementy elektroniczne i zasadę działania, realizowane są one w wersji analogowej i cyfrowej.

W grupie metod analogowych wyróżniamy metody:

- przetwarzania czas-napięcie, która polega na liniowym ładowaniu kondensatora w czasie mierzonego odcinka czasu, a następnie pomiarze napięcia na kondensatorze za pomocą przetwornika ADC,
- analogowej ekspansji czasu, polegającej na liniowym ładowaniu kondensatora i po wystąpieniu impulsu STOP, jego rozładowaniu prądem α razy mniejszym od ładującego, a czas ładowania wraz z α razy dłuższym czasem rozładowania kondensatora mierzony jest za pomocą metody licznikowej.

W grupie metod cyfrowych wyróżniamy metody:

- licznikową,
- noniuszową z dwoma startowalnymi oscylatorami,
- sukcesywnej aproksymacji,
- skracania impulsu,
- kodowania wielokrotnego,
- opartą na linii z opóźniającej z odczepami,
- z wykorzystaniem noniusza elektronicznego,
- stempli czasu,
- z użyciem zegara wielofazowego.

W interpolacyjnych metodach wielostopniowych w pierwszym stopniu interpolacji stosuje się na ogół zegar wielofazowy [35]. Systemy pomiaru odcinka czasu wykorzystujące interpolatory wielostopniowe charakteryzują się dużą rozdzielczością i szerokim zakresem pomiarowym, co okupione jest zazwyczaj długimi czasami martwym i przetwarzania.

Układy służące do precyzyjnego pomiaru odcinka czasu mogą w ogólności być zrealizowane z użyciem układów analogowych, cyfrowych lub mieszanych. W rozdziale niniejszym opisano cyfrowe metody i układy do pomiaru odcinka czasu, implementowane w układach programowalnych FPGA.

2.2.1 Metoda licznikowa

Metoda licznikowa jest najstarszą metodą pomiaru odcinka czasu. Polega ona na zliczaniu okresów zegara referencyjnego w czasie trwania mierzonego odcinka czasu. Rozdzielczość takiego pomiaru jest równa okresowi zegara referencyjnego. I tak, aby uzyskać rozdzielczość 1 ns w pomiarze jednokrotnym, należy użyć generatora o częstotliwości 1 GHz. Maksymalna częstotliwość użytego sygnału zegarowego jest jednak ograniczona możliwościami współczesnej technologii mikroelektronicznej, co limituje osiągnięte w ten sposób rozdzielczości.

2.2.2 Metoda noniusza elektronicznego z dwoma startowalnymi oscylatorami (chronotron)

Chronotron został opracowany w latach pięćdziesiątych ubiegłego wieku [44]. Impulsy START i STOP, oznaczające odpowiednio początek i koniec mierzonego odcinka czasu, wprowadzane są w dwie pętle zrealizowane jako linie opóźniające. Linie opóźniające posiadają zbliżone opóźnienia różniące się o $\Delta\tau$, i dobrane w taki sposób, aby pętla dla impulsu START miała opóźnienie większe. Liczba cykli impulsu STOP w pętli jest zliczana do momentu wykrycia koincydencji obydwu impulsów. Wartość mierzonego odcinka czasu jest obliczana według zależności:

$$\Delta T_m = n \cdot \Delta\tau, \quad (9)$$

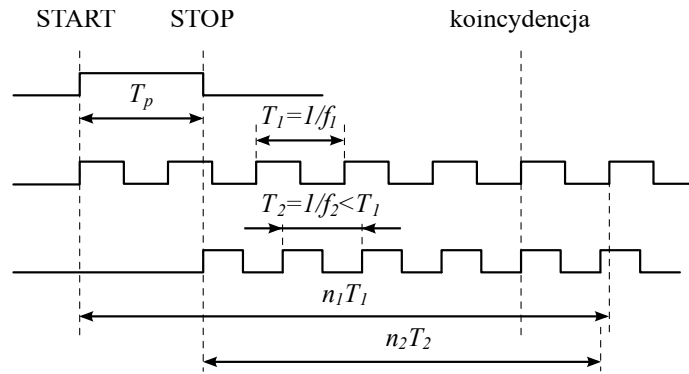
gdzie n jest liczbą cykli impulsów w pętlach.

Zastosowanie analogowych linii opóźniających o opóźnieniu 1 μ s pozwoliło osiągnąć rozdzielczość 2 ns i zakres pomiaru od 0 do 1 μ s [45]. Inną odmianę układu chronotyonu zaprezentowano w [46]. W opisywanym systemie użyto dwóch startowalnych oscylatorów o różniących się nieznacznie częstotliwościach f_1 i f_2 , przy czym $f_1 < f_2$. Pomiar polega na wystartowaniu pierwszego oscylatora na początku mierzonego odcinka czasu (rys. 8), a następnie drugiego oscylatora wraz z końcem odcinka czasu. Okresy zliczane są w osobnych licznikach dla każdego oscylatora. Rozdzielczość takiego systemu wynosi:

$$q = T_1 - T_2 = 1/f_1 - 1/f_2, \quad (10)$$

natomiast wynik pomiaru jest obliczany z następującej zależności:

$$T_m = (n_1 - 1)T_1 - (n_2 - 1)T_2 = (n_1 - n_2)T_1 + (n_2 - 1)q. \quad (11)$$

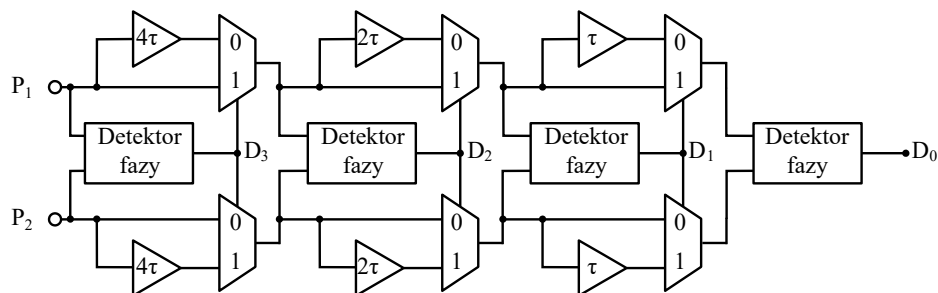


Rys. 8. Przebiegi w układzie chronotronu ze startowalnymi oscylatorami

Układ noniusza z dwoma startowalnymi oscylatorami można zaimplementować w strukturze układów programowalnych [48, 49] lub wykonać w układzie ASIC [47]. Oscylatory te zbudowane są w oparciu o aktywne elementy opóźniające. Dodatkowo w układach ASIC możliwe jest dodanie stabilizacji opóźnienia (częstotliwości) poprzez budowę specjalnych inwerterów o regulowanym opóźnieniu [50]. Wadą metody jest długi czas martwy.

2.2.3 Metoda sukcesywnej aproksymacji

Działanie przetwornika TDC opartego na metodzie sukcesywnej aproksymacji jest analogiczne do działania przetworników ADC o tej samej nazwie [51], a różnica polega na zmianie domeny przetwarzania z amplitudy na czas. Schemat przykładowego przetwornika 4 bitowego przedstawiono na rys. 9.



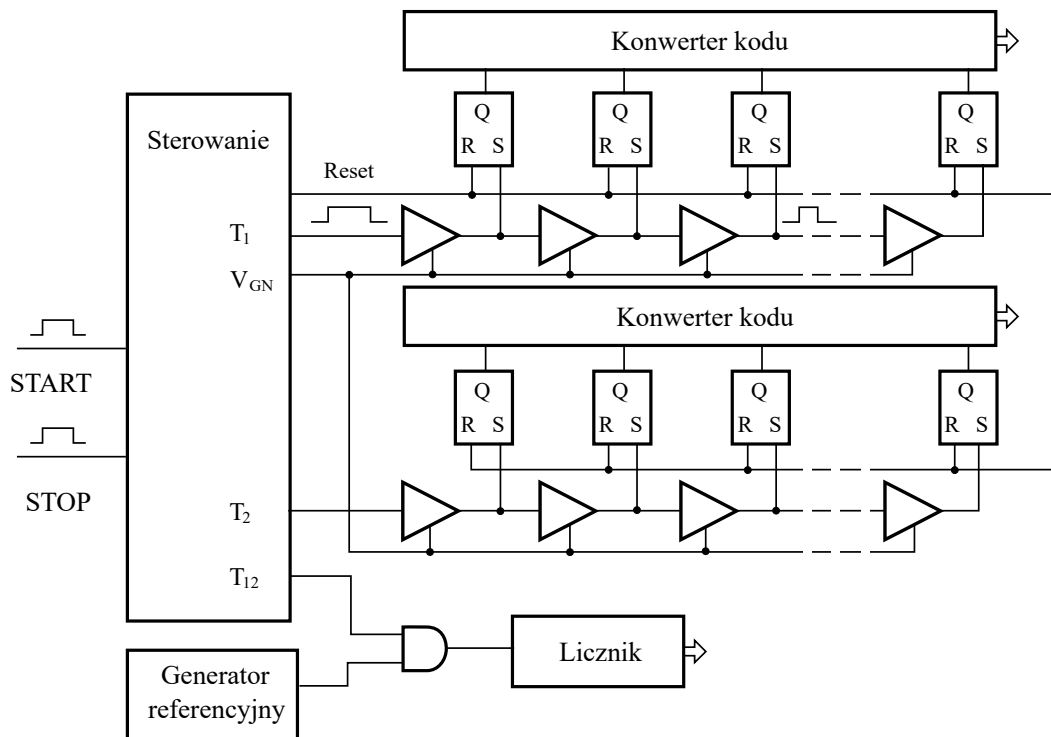
Rys. 9. Schemat blokowy przetwornika TDC czasowo-cyfrowego opartego na metodzie sukcesywnej aproksymacji

Głównymi elementami przetwornika są detektory fazy i tory opóźniające. Liczba detektorów fazy jest równa liczbie bitów n przetwornika. Podczas przetwarzania od najstarszego bitu, detektor fazy określa, który z impulsów wejściowych P_1 lub P_2 , oznaczających odpowiednio początek i koniec mierzonego odcinka czasu, wystąpił wcześniej na jego wejściu i który będzie opóźniony o wartość $2^{n-1} \cdot \tau$, względem późniejszego, transmitowanego dalej bez opóźnienia. Wynik przetwarzania odczytywany jest z wyjść detektorów fazy. W zmodyfikowanej wersji tego układu zamiast łańcucha dekoderek fazy oraz linii opóźniających o opóźnieniach jednostkowych $2^{n-1} \cdot \tau$, stosuje się jeden deko-

der oraz dwie identyczne pętle opóźniające o regulowanym opóźnieniu. Krążące impulsy są sukcesywnie opóźniane i porównywane w detektorze fazy. Wynik przekazywany jest do rejestru przesuwanego. Przetwornik oparty na tym rozwiązaniu w technologii CMOS 0,35 μm ma rozdzielczość 1,2 ps i niepewność pomiarową 3,2 ps w zakresie do 312 μs [52].

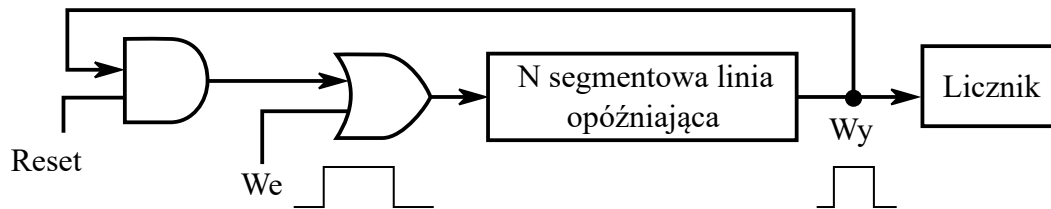
2.2.4 Metoda skracania impulsu

W metodzie skracania impulsu wykorzystuje się różne czasy propagacji zbczy narastającego oraz opadającego impulsów ($tp_{HL} > tp_{LH}$) w układzie scalonym [53]. Mierzony odcinek czasu odpowiada szerokości impulsu wprowadzanego na wejście łańcucha buforów, w których następuje skracanie impulsu, aż do jego całkowitego zaniknięcia. Różnica czasów propagacji odpowiada rozdzielczości przetwornika $q = tp_{HL} - tp_{LH}$, a długość odcinka czasu obliczana jest jako iloczyn wartości rozdzielczości i liczby buforów, po których impuls zaniknął. W pierwszej implementacji metody w technologii CMOS 1,2 μm uzyskano rozdzielczość 780 ps i zakres pomiarowy 10 μs [54]. Przykładową konstrukcję takiego systemu pokazano na rys. 10.



Rys. 10. Schemat układu opartego na metodzie skracania impulsu [53]

Ze względu na dużą nieliniowość przetwarzania oraz istotny wpływ temperatury i dryftu napięcia zasilania na czasy propagacji łańcucha buforów, precyzja opisanego TDC wynosiła zaledwie 3 ns. Zastosowanie innej konstrukcji buforów oraz dodanie automatycznej regulacji parametrów tych buforów umożliwiło osiągnięcie rozdzielczości 100 ps [55]. Niestety długi łańcuch buforów w tym ostatnim rozwiązaniu skutkuje dużym błędem nieliniowości. W kolejnych modyfikacjach zaproponowano skrócenie łańcucha buforów i jego zapętlenie [56]. Tego typu rozwiązanie schematycznie przedstawiono na rys. 11.

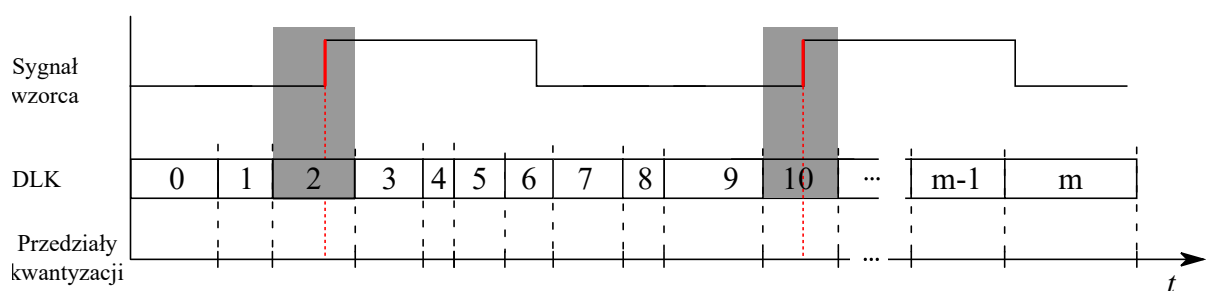


Rys. 11. Schemat układu pomiaru odcinka czasu opartego na metodzie cyklicznego skracania impulsu [56]

W szczególności pętla wymaga tylko jednego elementu, w którym występuje skracanie impulsu, dzięki czemu proces ten można łatwiej kontrolować. Impuls krążący w pętli skracany jest w każdym cyklu o taki sam kwant czasu q i krąży aż do całkowitego zaniku. Wynik pomiaru jest iloczynem liczby obiegów w pętli oraz wartości rozdzielczości. Metoda ta umożliwia osiągnięcie wysokiej rozdzielczości, jednak dla dużego zakresu pomiarowego okupione jest to dużą liczbą cykli obiegu pętli, co wydłuża czas przetwarzania. W celu zwiększenia częstotliwości pomiarów zaproponowano przetwarzanie wielostopniowe, dzięki czemu w układzie FPGA uzyskano rozdzielczość 42 ps i niepewność pomiarową poniżej 56 ps dla zakresu pomiarowego 11,7 ns [57]. W kolejnej modyfikacji metody na wejścia układu wprowadzane są dwa krótkie impulsy zamiast jednego mierzzonego. Ma to na celu eliminację niejednakowych czasów propagacji przeciwnych zboczy impulsu wprowadzanego do pętli [58]. Pierwsza pętla o rozdzielczości rzędu setek pikosekund służy do pomiaru zgrubnego, a druga o rozdzielczości dziesiątek pikosekund do ewaluacji dokładniejszej. Rozdzielczość drugiej pętli jest zarazem ostateczną rozdzielczością systemu pomiarowego. W ten sposób w układzie Spartan 3 firmy Xilinx uzyskano rozdzielczość 42 ps i niepewność pomiaru poniżej 56 ps w zakresie pomiarowym 11,5 ns [58].

2.2.5 Metoda kodowania wielokrotnego

Idea kodowania wielokrotnego jest pokazana na rys. 12 [59, 60].



Rys. 12. Metoda kodowania wielokrotnego z detekcją zbocza narastającego

Metoda polega na wprowadzeniu do wielosegmentowej linii opóźniającej (WLO) nie jednego aktywnego zbocza, lecz przebiegu o określonej liczbie zboczy, nazywanego sygnałem wzorca. Sygnał musi spełniać dodatkowe warunki. Czasy pomiędzy zboczami wzorca nie mogą być zbyt krótkie, aby nie doszło do zanikania tego sygnału ze względu na różne czasy propagacji zboczy narastającego i opadającego w linii opóźniającej. Jednakże czasy pomiędzy zboczami nie mogą być też zbyt długie, gdyż z ich zwiększeniem zachodzi konieczność wydłużania linii opóźniającej. Istota metody

polega na tym, że impuls START inicjuje generację sygnału wzorca, który propaguje się w linii opóźniającej.

W chwili pojawienia się impulsu STOP przebieg wzorca zostaje zapamiętany w rejestrze. W przypadku, gdy linia opóźniająca posiada różne opóźnienia na poszczególnych odczepach, a zbocze sygnału wzorca zostanie zarejestrowane w przedziale o dużej szerokości, to następne zbocza zarejestrowane w innym przedziale o mniejszej szerokości pozwoli uszczegółowić wartość kodowanego odcinka czasu, równocześnie zwiększając liniowość przetwarzania. Możliwe jest osiągnięcie rozdzielczości poniżej średniego opóźnienia linii, a krotność tej poprawy zależy od liczby zastosowanych zboczy wzorca.

Dekoder może reagować zarówno na jedno jak i na obydwa zbocza sygnałów wzorca (narażające i opadające), co podwaja liczbę zboczy branych pod uwagę w procesie dekodowania [61]. W pracy [62] opisano przetwornik zaimplementowany w układzie Kintex UltraScale, wykorzystujący metodę kodowania wielokrotnego dla dwóch zboczy oraz dla dwóch impulsów uzyskując rozdzielczość 1,23 ps i niepewność pomiarową o równą 3,6 ps.

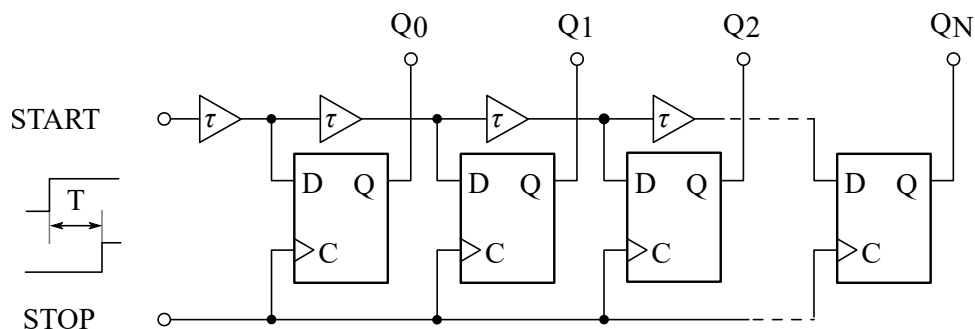
Za dalszą modyfikację dwóch wyżej wymienionych metod można uznać ich połączenie [63], umożliwiające uzyskanie przy użyciu wzorca o 6 zboczach w 3 niezależnych liniach kodujących zaimplementowanych w układzie Spartan-6 firmy Xilinx, rozdzielczości subpikosekundowej (902 fs).

2.2.6 Metoda przetwarzania z użyciem linii opóźniającej z odczepami

W metodzie tej wykorzystuje się wielosegmentowe linie opóźniające uzupełnione o skojarzony z nimi rejestr do bezpośredniego pomiaru odcinka czasu pomiędzy dwoma sygnałami reprezentującymi początek i koniec tego odcinka.

Przetwornik z linią opóźniająca i opóźnianym sygnałem START

W przypadku przetwornika jednostopniowego z opóźnianym sygnałem START, sygnał ten propaguje się w dyskretnej linii opóźniającej z odczepami, do których dołączone są przerzutniki D (rys. 13).



Rys. 13. Schemat budowy przetwornika z linią opóźniająca i opóźnianym sygnałem START

W momencie pojawienia się aktywnego zbocza sygnału STOP, stan linii opóźniającej jest zapamiętywany w przerzutnikach typu D. Wynik konwersji jest określony w kodzie termometrycznym.

W celu zapewnienia poprawności procesu przetwarzania sygnał START nie może być impulsem. Zakres pomiarowy związany jest z całkowitym czasem opóźnienia linii kodującej. Średnia rozdzielczość przetwornika jest obliczana jako iloraz całkowitego czasu opóźnienia i liczby segmentów opóźniających linii.

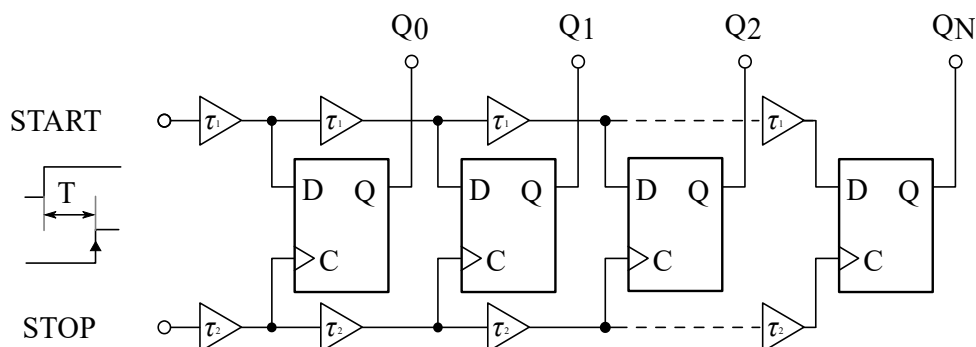
Formą modyfikacji powyższego układu jest użycie sygnału STOP w postaci zegara referencyjnego. Układ taki systematycznie zapamiętuje stany na wejściach przerzutników D, co skutkuje ciągłym próbkowaniem propagacji sygnału START [64, 65].

Przetwornik z linią opóźniającą i opóźnianym sygnałem STOP

Znana jest modyfikacja poprzedniego przetwornika polegająca na opóźnieniu w WLO sygnału STOP zamiast START. Natomiast sygnał START doprowadzony jednocześnie do wejść informacyjnych wszystkich przerzutników D jest próbkowany, a przerzutniki zatraskiwane są kolejno wraz z pojawieniem się zbocza sygnału STOP na odczepach linii opóźniającej. Podobnie jak w układzie poprzednim również tutaj występuje czas martwy, równy całkowitemu opóźnieniu linii opóźniającej.

Przetwornik z dwiema liniami opóźniającymi i opóźnianymi sygnałami START i STOP

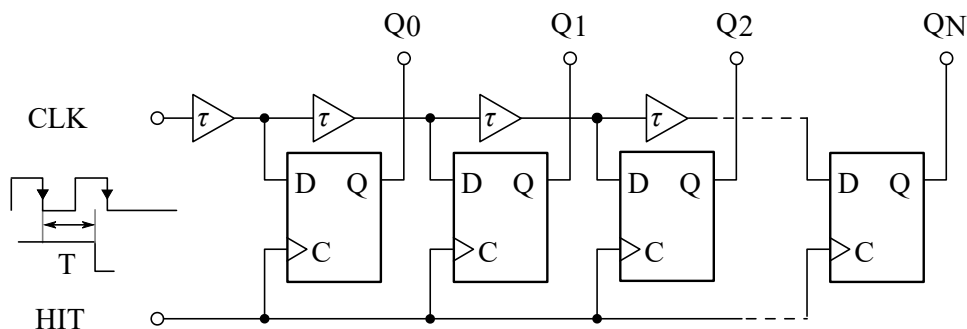
Układ nazywany różnicowym, zawiera dwie WLO, które charakteryzują się różnymi opóźnieniami pojedynczych segmentów τ_1 oraz τ_2 , gdzie $\tau_1 > \tau_2$ (rys. 14). Przykłady takiego rozwiązania opisane są w publikacjach [41] i [66]. W pierwszej z nich rozdzielczość systemu pomiarowego wynosiła 200 ps, a w jego ulepszonej wersji zrealizowanej w układzie pASIC2 firmy QuickLogic rozdzielczość ma wartość 100 ps. Pojedyncza różnicowa komórka charakteryzuje się opóźnieniem równym różnicy opóźnień jednostkowych linii $q = \tau_1 - \tau_2$. W przetworniku impuls STOP propaguje się szybciej w swojej linii opóźniającej i niejako dogania impuls START. Zaletą rozwiązania z dwiema liniami opóźniającymi jest zwiększenie rozdzielczości, a niekiedy także liniowości, przy odpowiednio dobranych opóźnieniach, kosztem zwiększenia czasu konwersji, zwłaszcza gdy łańcuchy linii opóźniających mają znaczne długości. Przetwornik taki daje się implementować w układzie FPGA z wykorzystaniem odpowiednio dobranych ścieżek połączeniowych oraz właściwym rozłożeniem elementów w strukturze układu programowalnego [33].



Rys. 14. Różnicowa linia opóźniająca

Przetwornik z linią opóźniającą i z opóźnionym sygnałem CLK

W układzie tym na wejście linii opóźniającej podaje się sygnał zegara referencyjnego CLK, a na wejście HIT odseparowane czasowo impulsy (rys. 15).



Rys. 15. Schemat przetwornika z linią opóźniającą i z opóźnionym sygnałem CLK

W momencie pojawienia się impulsu na wejściu HIT, w rejestrze zbudowanym z przerzutników D zapamiętany zostaje przebieg sygnału zegara CLK, czyli jego faza względem impulsu HIT. Pewnym problemem w tego typu przetwornikach jest zdekodowanie fazy zegara, ponieważ w jego przebiegu występuje na ogół kilka przejść ze stanu wysokiego na niski i odwrotnie. Długość linii opóźniającej i opóźnienia jej segmentów powinny być powiązane z częstotliwością zegara referencyjnego CLK i linia opóźniająca powinna mieć możliwość zarejestrowania przynajmniej jednego całego okresu zegara, jeśli dekodery dekodują tylko jeden typ zbocza. Istnieją rozwiązania, gdzie zamiast całego okresu rejestrowana jest dokładnie połowa okresu zegara [65], jednak dekodery muszą dekodować nie jedno, ale dwa typy zbocza. W takim przypadku mogą pojawić się dodatkowe problemy, wynikające z różnych opóźnień zboczy narastającego i opadającego [58]. Zaletą układu jest znikomy czas martwy. Możliwe jest ponadto rejestrowanie impulsów HIT z częstotliwością wyższą niż częstotliwość zegara referencyjnego.

2.2.7 Metody przetwarzania wielostopniowego

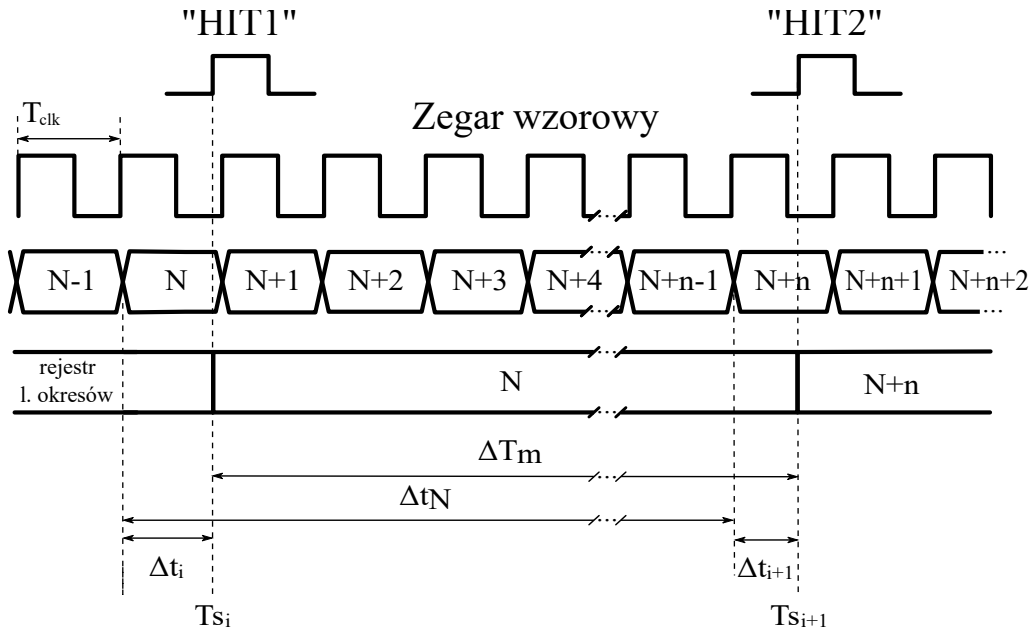
Opisane poprzednio metody charakteryzują się krótkim zakresem pomiarowym rzędu, co najwyżej mikrosekund. Ze względu na występującą często potrzebę pomiaru dłuższych odcinków czasu, zostały opracowane systemy, w których zakres pomiarowy jest znacząco wydłużony. Najstarszą metodą opracowaną w 1968 roku przez Ronalda Nutta jest metoda interpolacyjna [67], polegająca na wykonaniu pomiaru etapowo. Pierwszy etap to pomiar zgrubny, gdzie mierzony jest odcinek czasu z użyciem licznika taktowanego zegarem referencyjnym, zliczającym jego okresy. Drugi etap to pomiar dokładny, za pomocą precyzyjnych przetworników czasowo-cyfrowych w zakresie pojedynczego okresu sygnału zegarowego. Pierwsze układy pomiarowe tego typu o rozdzielczościach pikosekundowych opisane zostały w [68, 69]. Zastosowano w nich metody mieszane, tj. zarówno cyfrową, jak i analogową (wewnątrzokresową ekspansję czasu).

Istnieje kilka rodzajów tego typu rozwiązań. Do interpolacji wewnątrzokresowej wykorzystuje się na przykład przetwornik amplituda-cyfra [70] zaimplementowany w strukturze układu

ASIC [71], ale także w pełni cyfrowe przetworniki czas-cyfra [72], opisane dokładniej w dalszej części rozdziału.

Metoda stempki czasowych

W metodzie stempki czasu pomiar oparty jest na metodzie Nutta i składa się z trzech etapów (rys. 16) [67].



Rys. 16. Zasada pomiaru odcinka czasu metodą Nutta z użyciem stempki czasowych

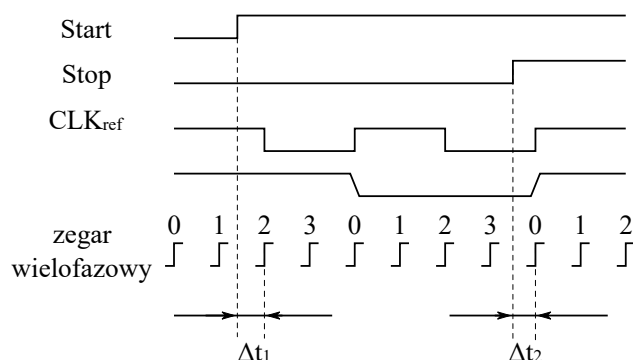
Zegar referencyjny, doprowadzany jest w WLO z rejestrze oraz do licznika okresów zegara. Sygnał HIT1 powoduje zapamiętanie w rejestrze fazy zegara referencyjnego oraz zapamiętanie stanu licznika okresów. Pojawienie się sygnału HIT2, powoduje kolejne zapamiętanie stanu licznika okresów $N+n$ oraz fazy zegara referencyjnego. W celu otrzymania wyniku pomiaru odcinka czasu ΔT_m należy zdekodować fazę sygnału zegarowego. Stempel czasu T_{s_i} zawiera informację o liczbie pełnych okresów zegara oraz opóźnieniu zdekodowanym na podstawie informacji przechowywanej w rejestrze. Wartość odcinka czasu ΔT_m (wzór 12) obliczana jest jako różnica wartości stempki czasowych. System pomiarowy wykorzystujący stempki czasu charakteryzuje się zdolnością do rejestrowania wielu zdarzeń HIT, co umożliwia pracę w trybie wielokanałowym lub MULTISTOP. Ma to istotne znaczenie na przykład w przypadku eksperymentów fizycznych (zliczanie fotonów, PET, spektrometry masowe TOF, itp.).

$$\Delta T_m = T_{s_{i+1}} - T_{s_i} = (N+n)T_{clk} + \Delta t_{i+1} - (NT_{clk} + \Delta t_i) = nT_{clk} + \Delta t_{i+1} - \Delta t_i \quad (12)$$

Metoda z zegarem wielofazowym

Nowoczesne układy scalone charakteryzują się coraz krótszymi czasami propagacji. Na przykład elementy łańcucha szybkich przeniesień arytmetycznych w nowych układach (Xilinx UltraScale, UltraScale+) mają opóźnienia rzędu pojedynczych pikosekund. Jeśli implementowana linia opóźniająca w układzie FPGA będzie działać z zegarem referencyjnym o niskiej częstotliwości, musi

składać się z dużej liczby segmentów (wzór 7). Może się więc okazać, że projektowana linia opóźniająca nie zmieści się w strukturze, ze względu na zbyt dużą liczbę potrzebnych do zrealizowania elementów lub z uwagi na jego topologię. Równocześnie nie będzie można zwiększyć częstotliwości referencyjnej, ze względu na ograniczoną maksymalną dopuszczalną częstotliwość pracy układu FPGA. Z reguły dobiera się częstotliwość zegara tak, aby jego okres był krótszy niż sumaryczne opóźnienie WLO. Drugim ograniczeniem liczby segmentów linii opóźniającej jest zjawisko rozmycia czasowego impulsów w tych liniach [42]. Z tego względu opracowano układy pomiarowe, w których linie opóźniające mają krótszy łańcuch o sumarycznym opóźnieniu mniejszym niż okres zegara referencyjnego, przy zachowaniu niskiej częstotliwości zegara wzorcowego. W układzie takim wykorzystuje się zegar wielofazowy, a sam pomiar realizowany jest w trzech etapach [73]. Zasada pomiaru jest pokazana na rys. 17.



Rys. 17. Zasada pomiaru z użyciem zegara wielofazowego

Zegar wielofazowy, utworzony z zegara referencyjnego, jest z nim ściśle skorelowany fazowo. W momencie pojawienia się impulsu START system dokonuje detekcji impulsu START względem sygnału zegara wielofazowego i wskazuje odpowiedni przedział fazowy tego zegara. Jest to pierwszy etap interpolacji. Kolejny etap odbywa się w interpolatorze wykorzystującym WLO. W układzie są dwa interpolatory drugiego stopnia. Pierwszy służy do określenia czasu Δt_1 dla impulsu START, a drugi do określenia czasu Δt_2 impulsu STOP. Zastosowanie zegara wielofazowego umożliwia użycie WLO zbudowanych z mniejszej liczby segmentów, a także użycie zegara referencyjnego o mniejszej częstotliwości.

3. Cel i tezy rozprawy

Wzrost rozdzielczości pomiaru odcinka czasu w systemach bazujących na interpolatorach realizowanych z użyciem wielosegmentowych linii opóźniających (WLO) może być osiągnięty poprzez zwiększenie liczby zastosowanych linii. Z uwagi na fakt, że charakterystyki WLO obciążone są rozrzutem losowym, wzrost rozdzielczości nie jest proporcjonalny do liczby użytych linii i stosunkowo szybko maleje nawet przy względnie małej ich liczbie (np. 16). Celem pracy jest wykazanie, że poprzez zastosowanie właściwego algorytmu wyboru linii możliwe jest uzyskiwanie wzrostu rozdzielczości dla większej ich liczby (nawet dla 128). Działanie wspomnianego algorytmu wyboru zasada się na odpowiedniej selekcji linii, której głównym kryterium jest możliwie największy wzrost rozdzielczości systemu pomiarowego. Innymi słowy do systemu pomiarowego zostają włączone jedynie te linie, które poprawiają parametry metrologiczne, w szczególności rozdzielczość, rozbudowywanego systemu.

Teza rozprawy:

Poprzez zastosowanie odpowiedniego algorytmu wyboru linii opóźniających do budowy WLO możliwe jest istotne zwiększenie rozdzielczości i liniowości systemu precyzyjnego pomiaru odcinka czasu.

Wyniki pracy znajdą zastosowanie w precyzyjnych systemach pomiaru odcinka czasu oraz we wszystkich systemach i przyrządach, opartych na użyciu linii opóźniających o rozdzielczościach pikosekundowych, których rozdzielczości są ograniczone przez technologię ich wytwarzania.

W tym kontekście rozprawa będzie zawierać:

- Analizę, projekt i badania wysokorozdzielczego systemu pomiaru odcinka czasu, opartego na wielu niezależnych WLO implementowanych w układzie FPGA.
- Opracowanie algorytmu automatycznego uzyskiwania WLO o korzystniejszej charakterystyce w aspekcie rozdzielczości i liniowości systemu.
- Opracowanie metody określającej w sposób ilościowy przydatność konkretnej linii do użycia w celu zwiększenia rozdzielczości systemu oraz weryfikację eksperymentalną tej metody.
- Opracowanie i weryfikację metody minimalizacji wpływu błędu bąbelkowego na wynik pomiaru.
- Opracowanie oprogramowania do kalibracji systemu i obliczania wartości stempli czasu.
- Przeprowadzenie badań potwierdzających tezę rozprawy.

4. System pomiaru odcinka czasu o kontrolowanych charakterystykach

Systemy pomiaru odcinka czasu o pikosekundowych rozdzielczościach są urządzeniami wykorzystywanymi głównie w laboratoriach badawczych. Systemy laboratoryjne budowane są obecnie przede wszystkim z użyciem układów programowalnych, ze względu na stosunkowo niskie koszty jednostkowe produkcji systemu pomiarowego oraz łatwość dostosowania do konkretnego badania i używanej aparatury badawczej. Najkorzystniejszym rozwiązaniem w kontekście uzyskiwania możliwie wysokich parametrów metrologicznych jest zaprojektowanie od podstaw systemu w układzie specjalizowanym ASIC. Jednakże z uwagi na wysoki koszt wyprodukowania układu scalonego nie zawsze jest to rozwiązanie opłacalne. Ponadto, proces projektowo-wytwórczy układów ASIC jest długotrwały, a modyfikacje lub usuwanie ewentualnych błędów w projekcie są praktycznie niemożliwe bez powtórnego wykonania układu scalonego. To powoduje, że obecnie w systemach pomiaru odcinka czasu dominują rozwiązania wykorzystujące układy FPGA, które zapewniają dużą ilość zasobów logicznych oraz sporą elastyczność w procesie projektowania. Zatem na potrzeby badań opisanych w niniejszej rozprawie opracowano system pomiarowy z zastosowaniem układu programowalnego FPGA firmy Xilinx. Dla uzyskania wysokiej rozdzielczości projektowanego systemu pomiarowego postanowiono użyć jednego z najnowszych, a jednocześnie przystępnych cenowo, układów programowalnych serii Kintex UltraScale. Przyspieszenie procesu projektowania uzyskano stosując gotową kartę prototypową KCU105 firmy Xilinx.

4.1 Platforma sprzętowa

W celu przeprowadzenia badań eksperymentalnych opracowano dedykowany system pomiaru odcinka czasu zaimplementowany na karcie prototypowej KCU105 (Xilinx) [17] [74], wyposażonej w układ programowalny serii Kintex UltraScale.

4.1.1 Karta prototypowa KCU105

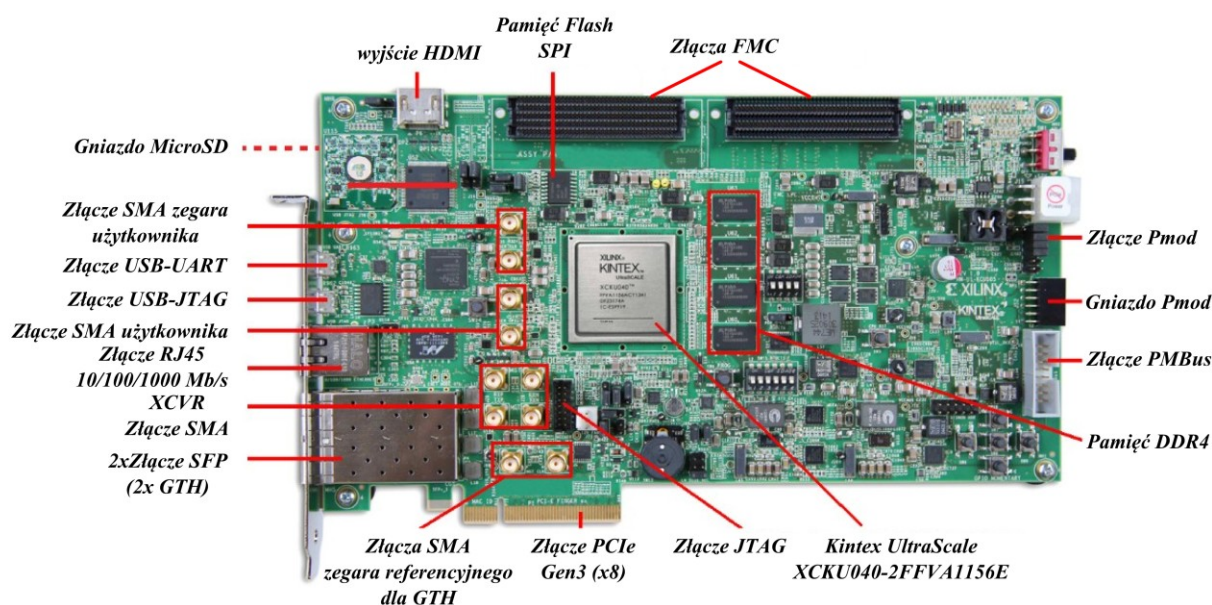
Płyta prototypowa KCU105 wykonana jest w formie karty do komputera PC ze złączem PCI Express (PCIe), ale dzięki zewnętrznemu zasilaniu i odpowiednim interfejsom przystosowana jest również do pracy samodzielnej, tj. poza komputerem.

Karta prototypowa KCU105 zawiera:

- układ FPGA typu XCKU040-2FFVA1156E z serii Kintex UltraScale,
- układ FPGA typu XC7Z010 z serii Xilinx Zynq,
- pamięć DDR4,
- dwa złącza micro USB – jedno służące jako UART-USB oraz drugie umożliwiające programowanie z użyciem interfejsu JTAG,
- trzy generatory – jeden systemowy i dwa programowalne przez użytkownika,

- dodatkowe elementy, takie jak np. przyciski, przełączniki, przełącznik obrotowo-impulsowy,
- zestaw złączy SMA umożliwiających doprowadzenie zewnętrznych sygnałów zegarowych w standardzie LVDT oraz wejściowych sygnałów użytkownika,
- interfejsy dodatkowe, takie jak: Ethernet RJ45, PMOD, SFP i FMC,
- pamięci nieulotne oraz gniazdo pamięci micro SD,
- wyjście Video HDMI,
- układy zasilania.

Widok karty z zaznaczonymi komponentami i złączami pokazany jest na rys. 18.



Rys. 18. Karta prototypowa KCU105 firmy Xilinx

Układ XC7Z010 Zynq-7000 umożliwia nadzorowanie działania karty KCU105 i do jego najistotniejszych funkcji należy zaliczyć: monitorowanie napięcia zasilania, programowanie układu Kintex UltraScale (emulacja programatora) i zadawanie nastaw dla programowalnych generatorów. Układ ten objęty został odrębnym łańcuchem JTAG, co uniemożliwia bezpośrednio programowanie go poprzez złącze USB-JTAG. Karta posiada również złącze PCI Express trzeciej generacji x8, dwa złącza FMC (FPGA Mezzanine Card) pozwalające na transfer danych z prędkościami do 10 Gb/s, zgodne ze standardem ANSI/VITA 57.1 [75] oraz istotne z punktu widzenia niniejszej rozprawy złącza SMA. Złącza te bowiem umożliwiają doprowadzenie m.in. zewnętrznych różnicowych sygnałów zegarowych do układu Kintex UltraScale. Na karcie znajdują się także gniazda na złącza SFP+ (ang. Small-form Factor Pluggable) dedykowane dla modułów komunikacyjnych, umożliwiające podłączenie kabli światłowodowych (jedno i wielomodowych) lub miedzianych (UTP). Moduły te osiągają prędkości transmisji do 16 Gb/s, posiadają własną pamięć EEPROM i zapewniają możliwość realizacji diagnostyki połączenia. Kartę prototypową wyposażono także w układy pamięci RAM typu DDR4 SDRAM o pojemności 2 GB oraz szybkości pracy 1200 MHz/2400 MT/s (Milionów Transferów/s). W skład tej pamięci wchodzi cztery układy scalone o pojemności 4 Gb

(256 Mb x 16) firmy Micron Technology. Komunikację z kartą, oprócz złącza PCIe, zapewniają także złącza UART-USB i Ethernet 10/100/1000 Mb.

Źródłami sygnałów zegarowych dostępnymi na karcie prototypowej KCU105 możliwymi do wykorzystania w projektowanym systemie odcinka czasu są:

- układ generatora Si5335A-B02436-GM firmy Silicon Labs, który wytwarza sygnały zegarowe o częstotliwościach 33,33 MHz, 90 MHz, 125 MHz i 300 MHz (zegar systemowy) i rozmyciu czasowym zboczy 0,7 ps RMS (ang. jitter). Do generacji powyższych częstotliwości używany jest rezonator kwarcowy firmy Epson o częstotliwości 25 MHz i stabilności częstotliwości 50 ppm oraz stabilności temperaturowej 30 ppm w zakresie od -20°C do 70°C. Sygnał zegara systemowego doprowadzany jest do układu FPGA za pomocą linii różnicowej LVDS. Sygnał o częstotliwości 33,33 MHz jest bezpośrednio doprowadzony do wejść układu Zynq,
- układ Si5328B, będący mnożnikiem częstotliwości korzystającym z technologii DSPLL (ang. Digital Signal Processing Phase Locked Loop) o zakresie częstotliwości wyjściowej 8 kHz – 808 MHz i ultra niskiej wartości jittera (0,3 ps RMS w zakresie 12 kHz – 20 MHz dla częstotliwości wyjściowej 125 MHz lub 156,25 MHz przy częstotliwości wejściowej 25 MHz). Sygnał z powyższego układu doprowadzany jest do multiplexera Si53340 i stanowi jedno z możliwych źródeł sygnału użytkownika. Układ Si5328B jest również odpowiedzialny za generację zegara do transmisji w standardzie Ethernet i taktowania GTH w modułach SFP+,
- programowalny generator Si570, dostarczający sygnał zegara użytkownika o częstotliwości z zakresu 10 MHz – 810 MHz i niskiej wartości jittera (0,4 ps RMS w zakresie 12 kHz – 20 MHz dla częstotliwości wyjściowej ≥ 500 MHz) w standardzie LVDS. Sygnał ten doprowadzony jest do układu multiplexera Si53340,
- złącza SMA stosowane do doprowadzania zewnętrznego zegara użytkownika za pomocą linii różnicowych LVDS oraz jako wejścia sygnału pomiarowego.

Układ Kintex UltraScale zawiera 20 nadajnikoodbiorników (ang. transceiver) GTH, które wykorzystywane są m. in. do transmisji PCIe (8 nadajnikoodbiorników) oraz w złączu rozszerzeń FMC (8+1). Jeden nadajnikoodbiornik podłączony jest do dedykowanych złączy SMA, a dwa pozostałe do gniazd SFP/SFP+.

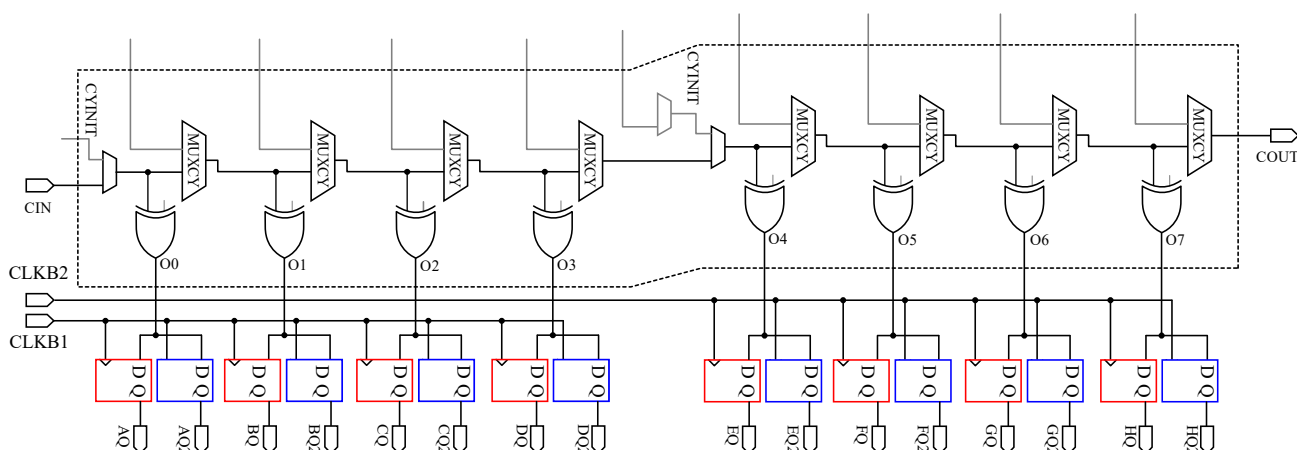
4.1.2 Układ programowalny Kintex UltraScale

Układ programowalny Kintex UltraScale należy do drugiej generacji układów Kintex i został wykonany w technologii CMOS o rozmiarze charakterystycznym 20 nm. Jądro układu zasilane jest napięciem 0,95 V. Układ charakteryzuje się dużą liczbą bloków funkcjonalnych takich jak bloki CLB, pamięci blokowe BRAM oraz procesory sygnałowe DSP48.

Zastosowany w karcie KCU105 układ Kintex UltraScale w wersji XCKU040-2FFVA1156E zawiera między innymi:

- 30300 bloków CLB, obejmujących 242400 tablic LUT oraz 484800 przerzutników,
- 1920 bloków DSP48,
- 21,1 Mb pamięci (600 bloków BRAM o pojemności 1k x 36 bit, mających możliwość podziału na pół, co w sumie daje 1200 BRAM o pojemności 512 x 36 bit), 7 Mb pamięci rozproszonej,
- 8 bloków CMT do zarządzania sygnałem zegarowym,
- 520 wyprowadzeń wejścia/wyjścia,
- system monitorowania odpowiadający za śledzenie temperatury i napięć zasilania,
- 20 nadajnikoodbiorników GTH o maksymalnej szybkości transmisji 16,3 Gb/s.

Architektura układów UltraScale została zaprojektowana do przetwarzania potokowego (ang. pipelining) [18, 19]. Tablice LUT (ang. Look-Up Table) są 6-wejściowe oraz 2-wyjściowe. W jednym CLB znajduje się 16 przerzutników oraz 8 tablic LUT (rys. 19).



Rys. 19. Uproszczony schemat połączeń CARRY8 z przerzutnikami w pojedynczym CLB

Łańcuch przeniesień arytmetycznych w jednym CLB posiada 8 wyprowadzeń. Blok funkcjonalny CMT (ang. Clock Management Tile) zawiera jeden blok MMCM (ang. Mixed-Mode Clock Manager) oraz dwa układy PLL (ang. Phase Locked Loop). Całość służy do generowania wielu sygnałów zegarowych o zdefiniowanych zależnościach fazowych i częstotliwościowych.

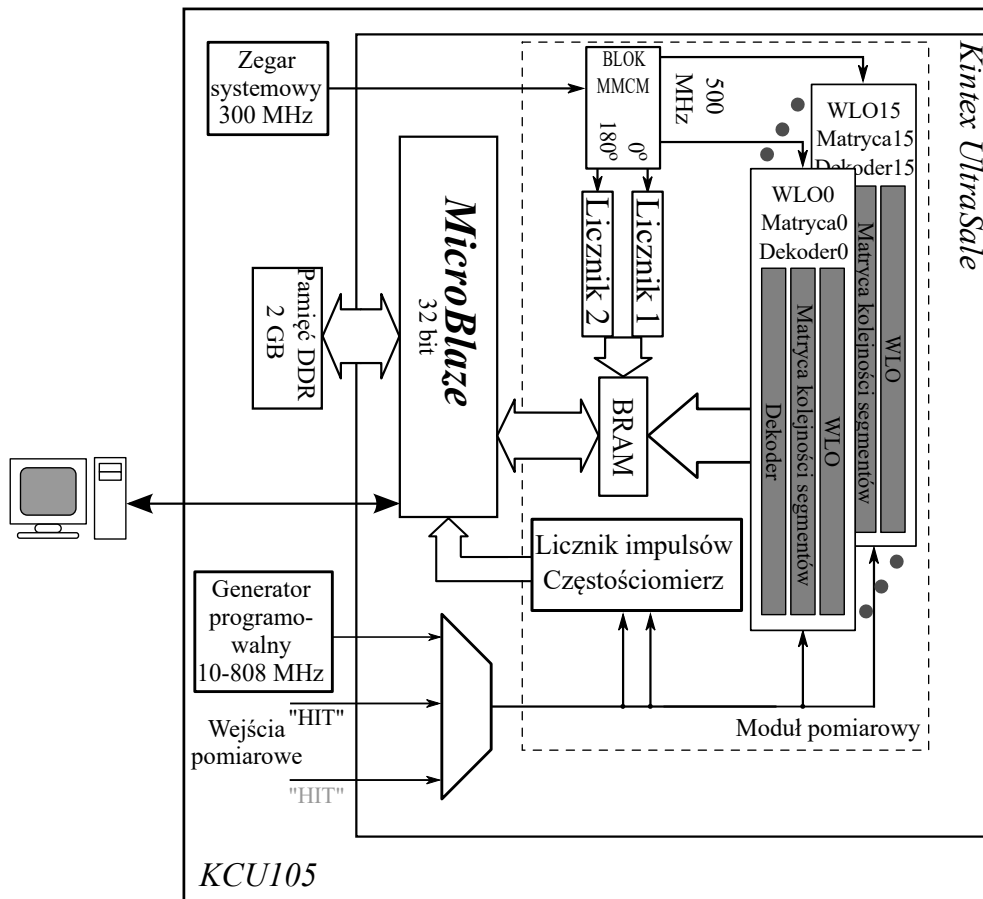
W układzie Kintex UltraScale został zaimplementowany system pomiaru odcinka czasu działający w oparciu o metodę Nutta. Proces przetwarzania realizowany jest przy użyciu WLO, które zostały zbudowane z użyciem łańcuchów przeniesień arytmetycznych. Do ich implementacji wykorzystano komponent CARRY8, który znajduje się w każdym bloku CLB. W układzie Kintex UltraScale element ten ma 8 wyjść, którymi wyprowadzono sygnały z kolejnych segmentów łańcucha CARRY. Uproszczony schemat połączeń CARRY8 z przerzutnikami pokazano na rys. 19. Każdy odczep CARRY8 doprowadzony jest do dwóch przerzutników. Nie istnieje możliwość innego połączenia z pozostałymi przerzutnikami bezpośrednio wewnątrz bloku CLB. Można jedynie wykorzystać dedykowaną parę przerzutników podłączoną odpowiednio do każdego odczepu CARRY8.

Układ Kintex UltraScale posiada dwie grupy wbudowanych linii opóźniających. Pierwsza grupa linii znajduje się w blokach wejściowo-wyjściowych. Linie te służą do wyrównania opóźnień sygnałów przy transferze danych pomiędzy układem Kintex UltraScale a zewnętrzną pamięcią typu DDR. Druga grupa linii opóźniających używana jest do syntezy częstotliwości przez bloki MMCM/DLL. Projektant nie ma jednak bezpośrednio dostępu do wyprowadzeń tych linii opóźniających i nie można ich zastosować np. do budowy przetworników czasowo-cyfrowych.

Układy Kintex UltraScale posiadają wiele dodatkowych komponentów wbudowanych, ale nie będą one szczegółowo opisywane w niniejszej pracy z uwagi na ich drugorzędne znaczenie w kontekście działania zaprojektowanego systemu pomiarowego. Dokładniejsze informacje uzyskać można w dokumentacji i materiałach producenta układu tj. firmy Xilinx/AMD [16, 17].

4.2 Projekt systemu pomiarowego

Schemat blokowy opracowanego konfigurowalnego systemu pomiaru odcinka czasu jest pokazany na rys. 20.

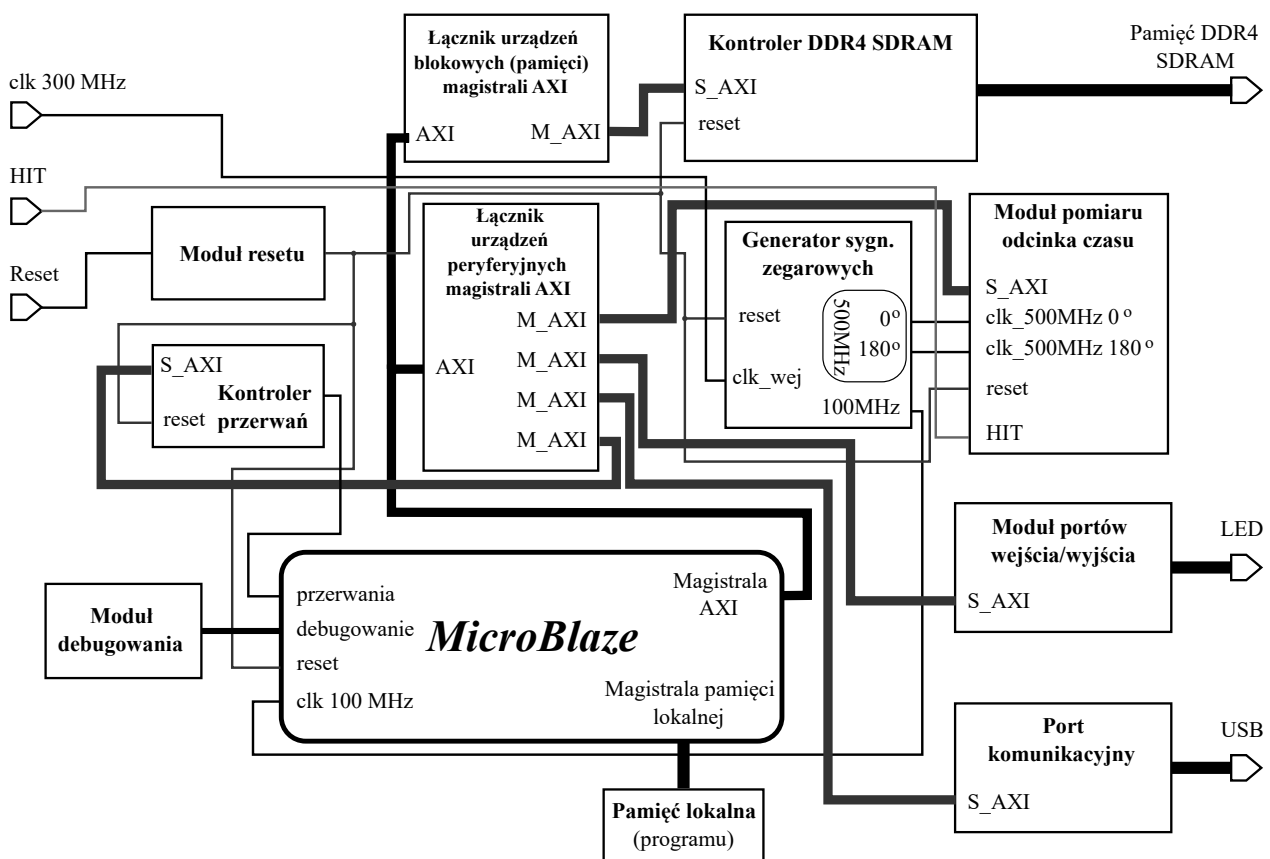


Rys. 20. Schemat opracowanego systemu pomiarowego

W systemie precyzyjnego pomiaru odcinka czasu wykorzystana została metoda Nutta. Zatem zgrubny pomiar odcinków czasu realizowany jest poprzez zliczanie okresów sygnału referencyjnego. W tym celu zastosowane zostały dwa liczniki (Licznik1 i Licznik2), na wypadek gdyby sygnał

wejściowy pojawił się w momencie inkrementacji jednego z liczników, co mogłoby spowodować w nim stany metastabilne i w efekcie błędne zliczanie. Natomiast dokładny pomiar odcinków czasu w zakresie pojedynczego okresu sygnału zegarowego wykonywany jest z użyciem przetworników opartych na WLO. Zegarowy sygnał referencyjny o częstotliwości 500 MHz, stanowiący podstawę czasu w systemie, podawany jest na wejścia WLO, a zdarzenia HIT pojawiające się na wejściach pomiarowych powodują zapamiętanie aktualnych stanów linii w rejestrach oraz wartości liczników w celu dalszego przetwarzania. W systemie wykorzystano szesnaście niezależnych WLO (od WLO0 do WLO15) o 480 odczepach każda. Dane pomiarowe przechowywane są w blokach wbudowanej pamięci BRAM. Po wypełnieniu pamięci BRAM, dane te są przetwarzane i magazynowane w zewnętrznej pamięci DDR. Całością systemu zarządza programowy mikroprocesor MicroBlaze. Po wykonaniu serii pomiarowej, dane pomiarowe z pamięci DDR wysyłane są do komputera PC. Położenie WLO w układzie może być zmienione, co wpływa na parametry metrologiczne linii, a docelowo na parametry całego systemu pomiarowego. Do kalibracji systemu pomiarowego używany jest generator programowalny z karty KCU105.

Bardziej szczegółowy schemat blokowy systemu opracowany w środowisku projektowym Vivado firmy Xilinx jest pokazany na rys. 21.



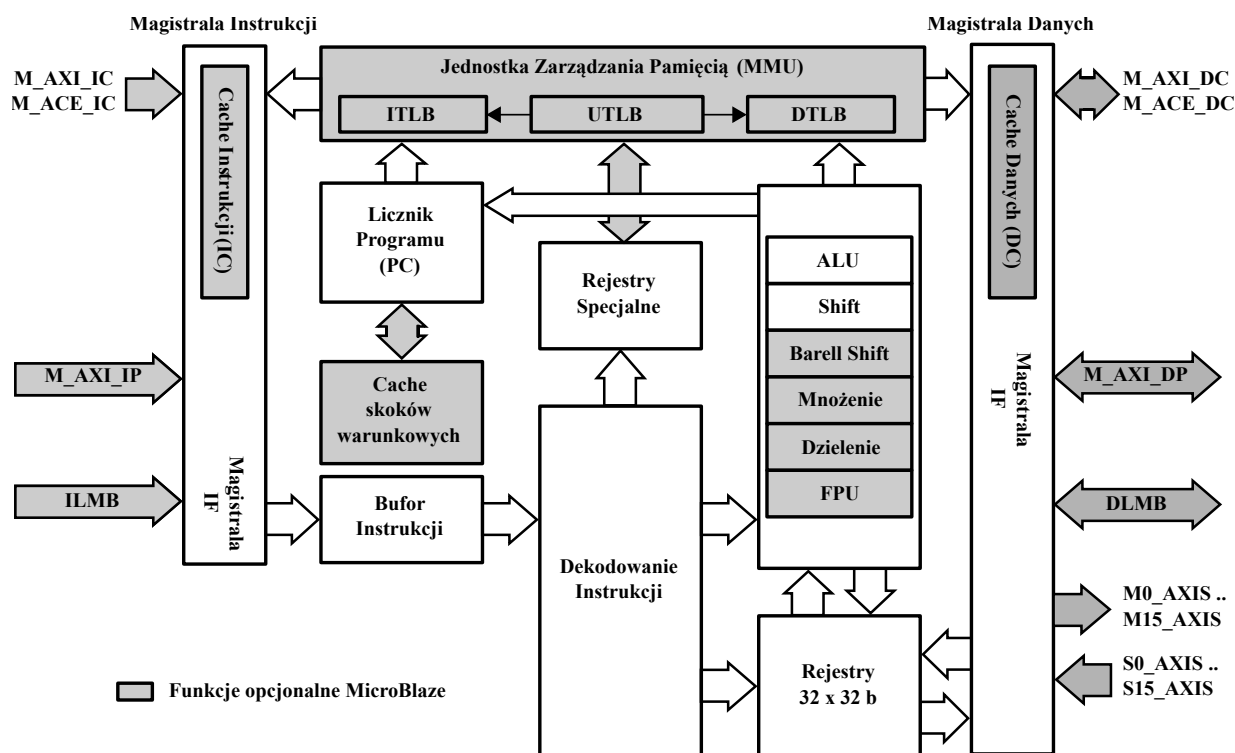
Rys. 21. Schemat kompletnego systemu pomiaru odcinka czasu w środowisku projektowym Vivado

System pomiarowy zaimplementowany w układzie Kintex UltraScale posiada budowę modułową. Dzięki temu może występować w wielu wariantach, dopasowanych do możliwości układu scalonego pod względem ilości zasobów logicznych. System pomiarowy bezpośrednio może być zaimplementowany w układzie z rodziny UltraScale zarówno w większych układach Virtex UltraScale jak

i mniejszych Zynq UltraScale+. Po odpowiednich modyfikacjach może być zaimplementowany w innych układach FPGA firmy Xilinx, z uwzględnieniem posiadania przez układ odpowiedniej ilości zasobów logicznych. System pomiarowy został zaprojektowany przy wykorzystaniu oprogramowania Vivado, które pozwala na utworzenie systemu mikroprocesorowego z modułami peryferyjnymi, a także na dodanie modułów (ang. IP Core) utworzonych przez projektanta. Kod modułu pomiaru odcinka czasu został napisany z użyciem języka VHDL i skompilowany jako IPCore, który może być podłączony do magistrali mikroprocesora MicroBlaze. Po podłączeniu do magistrali AXI, mikroprocesor w swojej przestrzeni adresowej zyskuje dostęp do rejestrów wejścia/wyjścia komponentu. Budowa modułów opracowanego systemu pomiaru odcinka czasu opisana jest w kolejnych rozdziałach rozprawy.

4.2.1 Mikroprocesor MicroBlaze

W systemie pomiarowym został użyty wirtualny mikroprocesor (mikroprocesor programowy) MicroBlaze w wersji 10.0 [77], którego schemat jest pokazany na rys. 22.



Rys. 22. Schemat blokowy mikroprocesora MicroBlaze

Jest to mikroprocesor opisany w języku HDL, jednak projektant nie ma bezpośredniego dostępu do kodu źródłowego¹. Dostępny jest jako komponent w formie czarnej skrzynki (ang. Black-Box), który może być konfigurowany.

¹ Firma Xilinx udostępniła kody źródłowe mikroprocesora MicroBlaze na specjalnych warunkach, np. dla projektów lub urządzeń do zastosowań wojskowych.

Firma Xilinx, która opracowała mikroprocesor MicroBlaze dostarcza różne jego warianty w zależności od typu układu FPGA i jego zasobów. W niniejszym rozdziale zwięźle opisany jest mikroprocesor MicroBlaze w wersji implementowanej w układzie Kintex UltraScale. Jego zasadnicze cechy to:

- 32 bitowy mikroprocesor, z 32 bitowym słowem rozkazowym z trzema operandami i dwoma trybami adresowania,
- 32 rejestry ogólnego przeznaczenia,
- 32 bitowa szyna adresowa z możliwością rozszerzenia do 64 bitów,
- przetwarzanie potokowe.

Mikroprocesor przed implementacją można konfigurować na wiele sposobów. Konfiguracja w programie Vivado [76] może odbywać się za pomocą parametrów predefiniowanych, takich jak:

- maksymalna wydajność osiągnięta typowo kosztem liczby wykorzystanych komponentów układu FPGA,
- maksymalna częstotliwość taktowania mikroprocesora,
- oszczędność wykorzystanych komponentów osiągnięta typowo kosztem wydajności,
- balans oznaczający kompromis pomiędzy wydajnością a liczbą wykorzystanych komponentów układu FPGA,
- optymalizacja architektury mikroprocesora pod kątem użycia docelowego systemu operacyjnego Linux, dostępnego w opcjach Linux z MMU (kontroler pamięci) lub Linux w wersji minimalnej (bez MMU).

Możliwy jest także dodatkowy szczegółowy wybór opcji sprzętowych, takich jak:

- włączenie/wyłączenie kontrolera pamięci MMU, który dokonuje translacji adresów wirtualnych na fizyczne, zapewnia ochronę wybranych obszarów pamięci, zapewnia obsługę pamięci podręcznej (ang. Cache) według mechanizmu znanego z systemów komputerowych (np. PC),
- włączenie/wyłączenie obsługi wyjątków (np. błąd dzielenia przez zero, błąd procesora),
- możliwość implementacji sprzętowej obliczeń zmiennoprzecinkowych, jak również dodatkowo mnożenia/dzielenia liczb całkowitych ze znakiem,
- użycie dodatkowych rejestrów przesuwanych (używane np. w szyfrowaniu i obliczaniu sum kontrolnych),
- włączenie odporności na błędy sprzętowe poprzez aktywowanie sumy kontrolnej pamięci RAM mikroprocesora oraz dodatkowe sumy kontrolne na magistrali lokalnej (LMB), w przypadku błędu generowany jest odpowiedni wyjątek,
- ustawienie obsługi pamięci podręcznej (Cache) dla pamięci danych i pamięci rozkazów, zalecane przy współpracy z zewnętrzną pamięcią dynamiczną RAM,
- debugowanie oraz sprzętowe pomiary wydajności,
- określenie sposobu zerowania (zbozca/poziom) oraz wartości poziomu („1” lub „0”) lub rodzaju zbozca (opadające/narastające).

Mikroprocesor MicroBlaze obsługuje zarówno format reprezentacji danych Big Endian (zapis od najstarszego bajtu) jak i Little Endian (od najmłodszego bajtu) [77]. Wszystkie parametry mikroprocesora są przekazywane po implementacji do kompilatora języka wysokiego poziomu (C, C++). W przypadku gotowego kodu źródłowego w języku wysokiego poziomu wystarczy go tylko na nowo skompilować, aby wykorzystać dodatkowe podzespoły sprzętowe. Kompilator dokonuje odpowiednich zmian w kodzie binarnym. Jeśli do pewnych zastosowań okaże się, że wydajność jest zbyt niska, można włączyć dodatkowe podzespoły mikroprocesora, co zwykle odbywa się kosztem zwiększenia zużycia koniecznych zasobów układu FPGA oraz wydłużenia czasu syntezy i implementacji. Możliwe jest zaimplementowanie więcej niż jednego mikroprocesora MicroBlaze w systemie. Oprogramowanie Vivado/SDK wspiera wieloprocessorowość, a warunkiem jej realizacji jest wystarczająca ilość wolnych zasobów logicznych w układzie FPGA.

W tabeli 1 pokazany jest wynik z raportu po implementacji procesora MicroBlaze w układzie Kintex UltraScale XCKU040. Liczba użytych istotnych zasobów układu nie przekroczyła 1%.

Tabela 1. Stopień wykorzystania zasobów układu Kintex UltraScale XCKU040 po zaimplementowaniu jednego mikroprocesora MicroBlaze syntezywanego w opcji wysokiej wydajności kosztem zwiększonego zużycia zasobów układu

Utilization - Post-Implementation			
Resource	Utilization	Available	Utilization %
LUT	1181	242400	0.49
LUTRAM	135	112800	0.12
FF	965	484800	0.20
BRAM	16	600	2.67
IO	3	520	0.58
BUFG	3	480	0.63
MMCM	1	10	10.00

Po zaimplementowaniu dwóch mikroprocesorów MicroBlaze (tabela 2) liczba użytych zasobów wzrosła wprawdzie dwukrotnie, jednak nadal zużycie zasobów jest mniejsze niż 1% (tablice LUT i przerzutniki).

Tabela 2. Stopień wykorzystania zasobów układu Kintex UltraScale XCKU040 po zaimplementowaniu dwóch mikroprocesorów MicroBlaze syntezywanego w opcji wysokiej wydajności kosztem zwiększonego zużycia zasobów układu

Utilization - Post-Implementation			
Resource	Utilization	Available	Utilization %
LUT	2263	242400	0.93
LUTRAM	262	112800	0.23
FF	1796	484800	0.37
BRAM	32	600	5.33
IO	3	520	0.58
BUFG	3	480	0.63
MMCM	1	10	10.00

Należy podkreślić, że w prezentowanym przypadku, każdy mikroprocesor posiadał 64 kB pamięci lokalnej (16 k x 32 b). Obydwa przypadki (tabela 1, tabela 2) były syntezywane w opcji wysokiej

wydajności, czyli kosztem zasobów układu. Dla porównania dokonano implementacji dwóch procesorów MicroBlaze (tabela 3), ale dla opcji oszczędzania zasobów.

Tabela 3. Stopień wykorzystania zasobów układu Kintex UltraScale XCKU040 po zaimplementowaniu dwóch mikroprocesorów MicroBlaze z opcją oszczędzania zasobów układu programowalnego

Utilization - Post-Implementation			
Resource	Utilization	Available	Utilization %
LUT	1699	242400	0.70
LUTRAM	308	112800	0.27
FF	1188	484800	0.25
BRAM	32	600	5.33
IO	3	520	0.58
BUFG	3	480	0.63
MMCM	1	10	10.00

W tym przypadku zmniejszyło się wykorzystanie tablic LUT o 33% oraz przerzutników (FF) o 51%, jednak kosztem zwiększenia pamięci rozproszonej (LUTRAM) o 17%.

Zastosowanie dwóch procesorów daje możliwość lepszego dostosowania systemu do potrzeb użytkowników lub konkretnych zastosowań. Możliwe jest na przykład przydzielenie jednego mikroprocesora do kontroli i sterowania procesem pomiarowym, a drugiego do przetwarzania i transmisji danych pomiarowych. Ta opcja pozwala na ciągłą realizację procesu pomiarowego nawet dla dużej intensywności rejestrowanych impulsów. W opisanych dalej badaniach eksperymentalnych wykorzystano system z pojedynczym procesorem.

4.2.2 Magistrala AXI

Magistrala AXI (ang. Advanced eXtensible Interface) jest zgodna z magistralami używanymi w mikroprocesorach ARM. Po raz pierwszy użyto jej w 1996 roku. Magistrala ta została wprowadzona w oprogramowaniu Xilinx ISE 12.3 i jest używana przez mikroprocesor MicroBlaze w wersji dla układów Kintex UltraScale. Obecnie stosowana jest 4 generacja AXI (AXI-4 AMBA 4.0), która powstała w 2010 roku [77].

W przypadku mikroprocesora MicroBlaze magistrala ta umożliwia dodawanie komponentów i urządzeń peryferyjnych takich jak wejścia/wyjścia, obliczeniowych, FFT (ang. Fast Fourier Transform) i działających na blokach danych. Komponenty te są dostępne w repozytorium gotowych bloków IP. Możliwe jest również dodawanie własnych komponentów, utworzonych z innych gotowych bloków dostarczonych z programem Vivado przez firmę Xilinx lub napisanych przez użytkownika w językach HDL. W prezentowanej pracy utworzono blok systemu pomiarowego w postaci modułu w języku VHDL.

4.2.3 Wielosegmentowe linie opóźniające

Wielosegmentowa linia opóźniająca została zaimplementowana w systemie pomiarowym w języku VHDL [78] w sposób strukturalny. Na listingu 1 zaprezentowany jest kod dla WLO implementowanej w układzie FPGA.

Listing 1. Opis wielosegmentowej linii opóźniającej w języku VHDL

```
1: -----
2: library IEEE;
3: use IEEE.STD_LOGIC_1164.ALL;
4: use IEEE.STD_LOGIC_ARITH.ALL;
5: use IEEE.STD_LOGIC_UNSIGNED.ALL;
6: use IEEE.numeric_std.all;
7: library unisim;
8: use unisim.vcomponents.all;
9:
10: entity wlo is
11: generic (
12: dl_linii: integer := 480
13: );
14: port (
15:     clk_wlo      : in std_logic;
16:     hit          : in std_logic;
17:     wlo_out      : out std_logic_vector(dl_linii-1 downto 0);
18:     wlo_out2     : out std_logic_vector(dl_linii-1 downto 0)
19: );
20:
21: -- opis
22: -- clk_wlo   - wejście zegara referencyjnego
23: -- hit      - wejście sygnału zatrzymującego
24: -- wlo_out  - wyjście z przerzutników nieparzystych
25: -- wlo_out2 - wyjście z przerzutników parzystych
26:
27: end wlo;
28:
29: architecture Behavioral of wlo is
30:
31: signal D_wy      : std_logic_vector(dl_linii+2 downto 0);
32: signal D_wy2    : std_logic_vector(dl_linii+2 downto 0);
33: signal muxwy    : std_logic_vector(dl_linii+4 downto 0);
34: signal xmuxwy   : std_logic_vector(dl_linii+4 downto 0);
35:
36: attribute KEEP : string;
37: attribute keep of D_wy : signal is "true";
38: attribute keep of D_wy2: signal is "true";
39: attribute keep of muxwy: signal is "true";
40:
41: begin
42:
43: WLOC:for i in 0 to dl_linii-1 generate
44:     FD : FDCE
45:     generic map (INIT => '0')
```

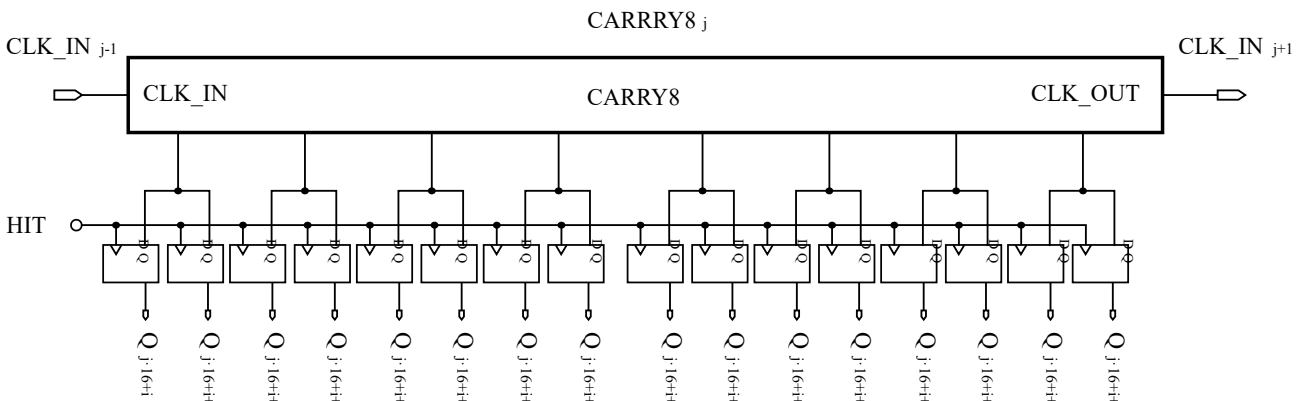
```

46: port map (D => muxwy(i), C => hit, CE => '1', CLR => '0', Q => D_wy(i));
47: FD2:FDCE
48: generic map (INIT => '0')
49: port map (D => xmuxwy(i), C => hit, CE => '1', CLR => '0', Q => D_wy2(i));
50:
51: C_WEJ:if (i=0) generate
52:     muxcy_wej : CARRY8
53:     generic map (
54:         CARRY_TYPE => "SINGLE_CY8"
55:     )
56:     port map (
57:         CO           => muxwy (7 downto 0),
58:         O            => xmuxwy(7 downto 0),
59:         CI           => clk_wlo,
60:         CI_TOP       => '1',
61:         DI           => "00000000",
62:         S            => "11111111"
63:     );
64:
65: end generate;
66:
67: C8:if ((i mod 8=0) and (i>0) and i<dl_linii ) generate
68:     muxcy : CARRY8
69:     generic map (
70:         CARRY_TYPE => "SINGLE_CY8"
71:     )
72:     port map (
73:         CO           => muxwy (i+7 downto i),
74:         O            => xmuxwy(i+7 downto i),
75:         CI           => muxwy (i-1),
76:         CI_TOP       => '1',
77:         DI           => "00000000",
78:         S            => "11111111"
79:     );
80: end generate;
81:
82: end generate;
83:
84: wlo_out <= D_wy(dl_linii-1 downto 0);
85: wlo_out2<= D_wy2(dl_linii-1 downto 0);
86:
87: end Behavioral;
88:
89: -----

```

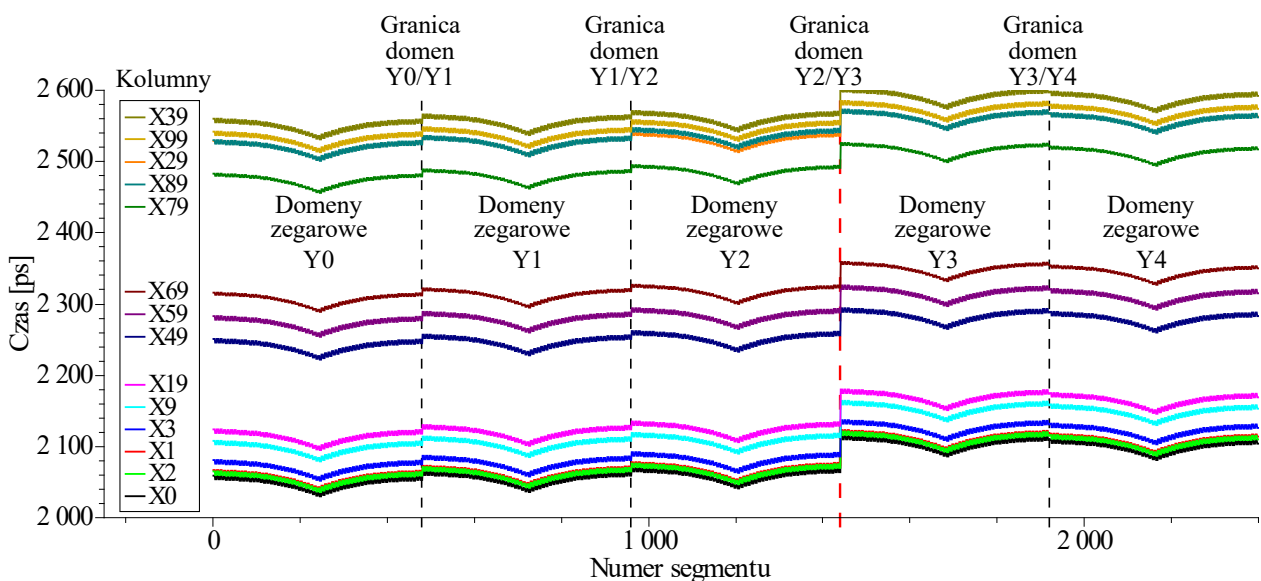
Opracowując projekt WLO skorzystano z zasobów biblioteki producenta firmy Xilinx dla układu Kintex UltraScale [79]. W liniach 36–39 został dodany atrybut *keep*, aby narzędzia do syntezy i implementacji nie dokonały optymalizacji użytych sygnałów, poprzez ich usunięcie. Następnie w pętli tworzona jest WLO w oparciu o komponent CARRY8 i przerzutniki typu D (rys. 23). W liniach 51–63 w łańcuchach przeniesień arytmetycznych wprowadzany jest sygnał zegara referencyjnego (linia 59, początek WLO). W dalszych CLB, wejście następnego CARRY8 podłączone jest do

wyjścia poprzedniego CARRY8 (linie 67–79). W wyniku syntezy tego kodu została utworzona WLO o 480 odczepach (segmentach), do których w projektowanym przetworniku dołączone będzie maksymalnie $480 \cdot 2 = 960$ przerzutników D. Linia zajmuje $480/8 = 60$ bloków CLB.



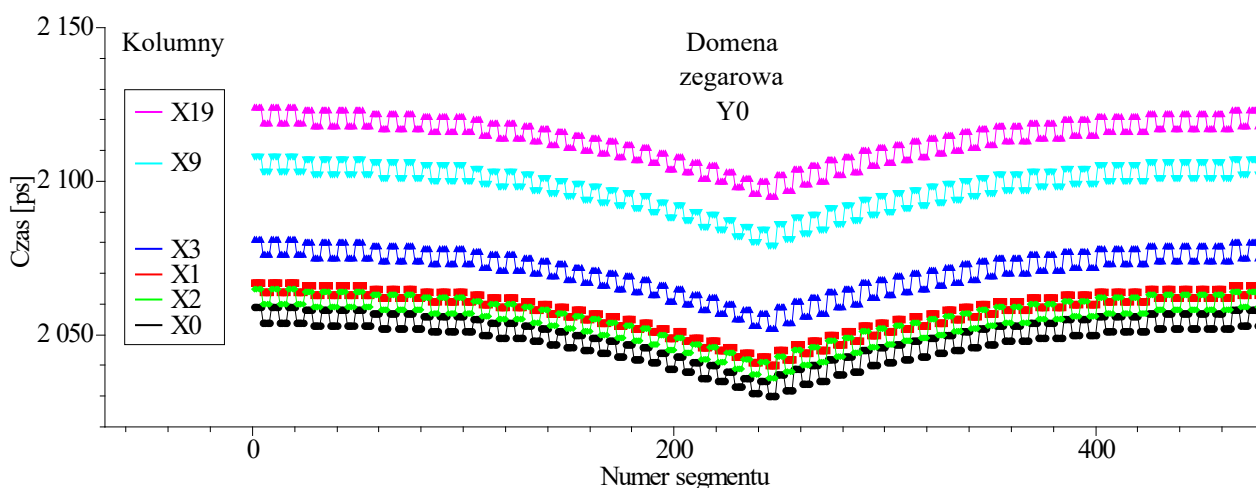
Rys. 23. Fragment schematu WLO opartej na elementach łańcucha przeniesi arytmetycznych (ang. carry chain) CARRY8 układu Kintex UltraScale

Dokładnie tyle CLB znajduje się w pojedynczej kolumnie w jednej domenie zegarowej. Wszystkie 960 przerzutników reaguje na to samo zbocze sygnału HIT, który jest rozprowadzony w sposób bezpośredni, tj. bez dodatkowych łączników i buforów zegarowych, jak ma to miejsce w przypadku transmisji sygnału HIT przez granicę domen zegarowych (widoczne „skoki” opóźnień na rys. 24).



Rys. 24. Opóźnienia sygnału rozprowadzanego do wejść zegarowych przerzutników w WLO o długości całej kolumny układu Kintex UltraScale (2400 odczepów). Na wykresie przedstawiono wybrane kolumny. Wartości opóźnień określone zostały w środowisku Vivado. Zauważalny znaczny „skok” opóźnień na granicy domen Y2 i Y3 zaznaczony jest czerwoną linią przerywaną.

Zaimplementowanie całej linii opóźniającej w jednej domenie zapewnia charakterystykę WLO najbardziej zbliżoną do liniowej i eliminuje z WLO odczepy o bardzo dużym opóźnieniu (rys. 25).

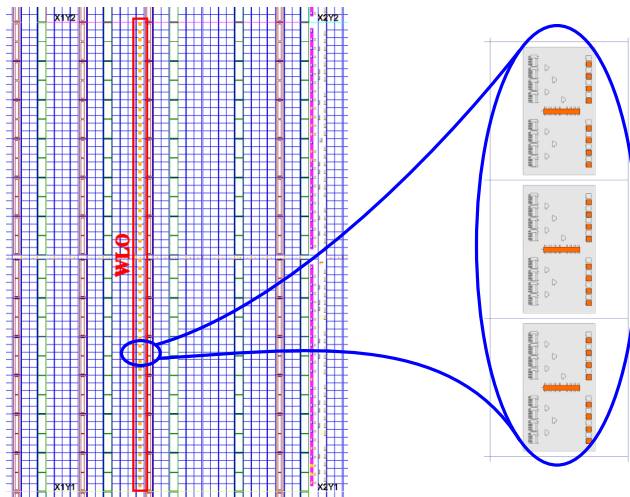


Rys. 25. Opóźnienia sygnału rozprowadzanego do wejść zegarowych przerzutników w WLO o długości jednej domeny zegarowej układu Kintex UltraScale (480 odczepów) dla domeny Y0. Na wykresie przedstawiono wybrane kolumny. Widoczne „schołki” spowodowane są różnicą czasów propagacji ścieżek sygnału wewnątrz jednego CLB. Wartości opóźnień określone zostały w środowisku projektowym Vivado.

Szczególnie duży skok opóźnień jest zauważalny na granicy domen zegarowych Y2 oraz Y3 (rys. 24). Implementacja WLO na granicy tych dwóch domen będzie skutkowałą uzyskaniem przedziałów kwantowania przetwornika o znacznych szerokościach, co spowoduje duże nieliniowości charakterystyk przetwarzania. Z tego powodu, w przypadku konieczności implementacji WLO o długości wykraczającej poza obszar jednej domeny zegarowej, należy unikać dla badanego układu Kintex UltraScale przejścia przez granicę domen Y2/Y3.

Rozmieszczenie w układzie FPGA komórek WLO zbudowanej z użyciem łańcucha przeniesień arytmetycznych zapewnione jest dzięki specyfikacji lokalizacji komórek w pliku ograniczeń użytkownika (.XDC). W pliku tym oprócz przypisania wyprowadzeń i określenia częstotliwości sygnałów zegarowych, projektant ma możliwość zgrubnego ustalenia potencjalnego miejsca implementacji projektowanych modułów, poprzez ograniczenie obszaru implementacji. Możliwe jest także wprowadzenie ograniczeń dla pojedynczych komponentów tworzących moduł pod warunkiem spełnienia wymaganych kryteriów czasowych. Ograniczeniem procesu kompilacji może być również na przykład niewystarczająca liczba wolnych linii rozprowadzenia zegara, buforów zegara lub matryc połączeniowych w danym obszarze. Brak spełnienia powyższych uwarunkowań może skutkować przerwaniem procedury implementacji i koniecznością modyfikacji topologii projektu. W takim przypadku implementacja jest możliwa za pomocą specjalnych dyrektyw ograniczających restrykcje w procesie implementacji [81]. Stosowny komunikat pojawia się w czasie implementacji w programie Vivado wraz z instrukcją dotyczącą sposobu ewentualnego zignorowania ostrzeżeń i jednoczesnego skrócenia czasu implementacji. W przeciwnym przypadku program Vivado będzie nieustannie próbował rozmieścić i połączyć komponenty w celu minimalizacji liczby ostrzeżeń dotyczących naruszeń sugerowanych zależności czasowych.

Dla szeregowo połączonych elementów łańcucha przeniesień arytmetycznych w WLO wystarczające jest wskazanie lokacji pierwszego elementu CARRY8, a oprogramowanie samo rozmieści pozostałe elementy WLO zgodnie z ich rosnącymi indeksami (rys. 26).



Rys. 26. Wycinek projektu topologicznego (layoutu) w programie Vivado obrazujący rozmieszczenie WLO zajmującej 60 kolejnych CLB. Przykładowe rozmieszczenie WLO pokazane jest za pomocą elementów w kolorze pomarańczowym (czerwona ramka). Początkowe elementy WLO znajdują się w SLICE X33Y60 a końcowe w SLICE X33Y119, czyli cała linia znajduje się w obszarze jednej domeny zegarowej XIY1. Na powiększeniu widoczne trzy bloki CLB z zaznaczonymi na pomarańczowo CARRY8 i przerzutnikami.

Projektant, oprócz konfiguracji procesu implementacji w pliku .XDC, może wybrać obszar implementacji używając oprogramowania Vivado, które po syntezie umożliwia rozmieszczenie komponentów za pomocą edytora graficznego. Po edycji rozmieszczenia komponentów program Vivado modyfikuje plik .XDC. Fragment pliku .XDC z opisem dwóch WLO (LINIA[0] oraz LINIA[1]) pokazano w listingu 2.

Listing 2. Fragment pliku ograniczeń wskazanych przez użytkownika (.XDC) ustalający położenie pierwszych bloków łańcucha CARRY8 dwóch linii opóźniających w układzie Kintex UltraScale XCKU040

```
set_property BEL CARRY8 [get_cells {config_mb_i/wlo_sys_1/U0/wlo_sys_v2_0_S00_AXI_inst/SYSTEM/LINIA[0].L/WLOC[0].C_WEJ.muxcy_wej}]
set_property LOC SLICE_X33Y60 [get_cells {config_mb_i/wlo_sys_1/U0/wlo_sys_v2_0_S00_AXI_inst/SYSTEM/LINIA[0].L/WLOC[0].C_WEJ.muxcy_wej}]
set_property BEL CARRY8 [get_cells {config_mb_i/wlo_sys_1/U0/wlo_sys_v2_0_S00_AXI_inst/SYSTEM/LINIA[1].L/WLOC[0].C_WEJ.muxcy_wej}]
set_property LOC SLICE_X35Y60 [get_cells {config_mb_i/wlo_sys_1/U0/wlo_sys_v2_0_S00_AXI_inst/SYSTEM/LINIA[1].L/WLOC[0].C_WEJ.muxcy_wej}]
```

Widniejąca na początku powyższego listingu klauzula **set_property BEL CARRY8** umożliwia wybranie komponentu CARRY8. W następnej linii zapis **set_property LOC SLICE_X33Y60** określa docelowe położenie komponentu. Jeśli cała WLO ma się zmieścić w jednej domenie zegarowej (domena posiada 60 CLB/SLICE), rozmieszczanie należy rozpocząć od początku domeny zegarowej (kolejne wartości pozycji pionowej: Y0, Y60, Y120, Y180, Y240).

Bez wskazań zawartych w pliku ograniczeń użytkownika (.XDC) oprogramowanie zoptymalizuje położenie implementowanych WLO, ze względu na przykład minimalizację opóźnień rozprzodzenia sygnałów i rozlokuje je w układzie w „przypadkowych” kolumnach, w szczególności

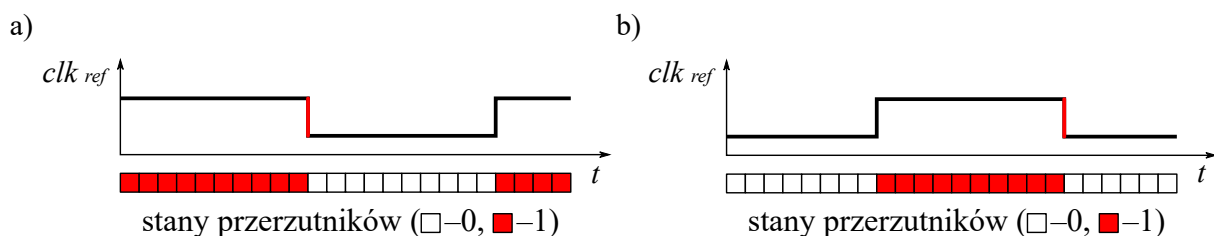
ści w dwóch różnych domenach zegarowych. W wyniku czego powstałe WLO będą posiadały odczepy o dużych opóźnieniach (widoczne skoki opóźnień w rys. 24).

Horyzontalne położenie WLO może przyjmować dowolną wartość X w zakresie 0-100, pod warunkiem, że w przypadku implementacji systemu z wieloma WLO, możliwe będzie doprowadzenie wszystkich sygnałów zegarowych. Z praktyki wynika, że w przypadku rozmieszczenia wertykalnego, najlepiej jest rozmieszczać WLO tak, aby zaczynały się w tej samej pozycji Y (w celu zminimalizowania różnic opóźnień w rozprowadzeniu sygnałów zegara referencyjnego). Z uwagi na dużą czasochłonność procesu rozlokowania komórek WLO napisany został skrypt w języku Python, zapewniający automatyczną generację odpowiednich wpisów w pliku .XDC.

4.2.4 Moduł przetwarzania i akwizycji danych

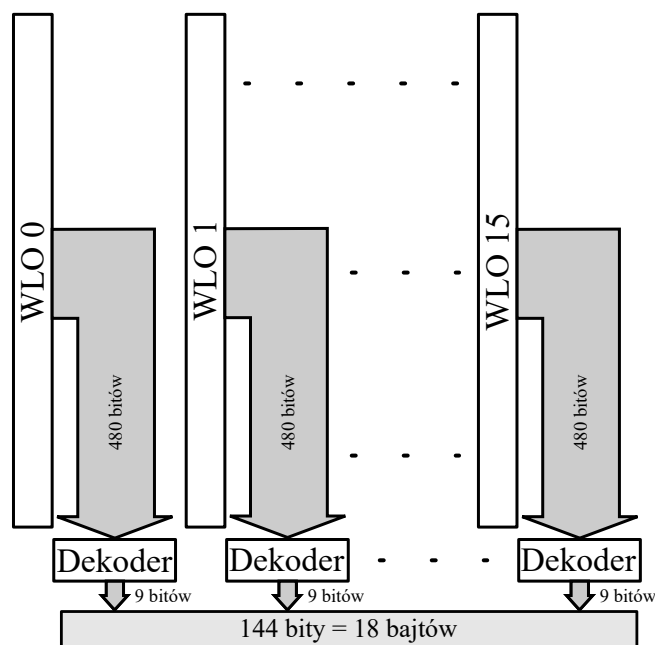
W zaprojektowanym systemie pomiaru odcinka czasu w szczególności, w przetworniku czasowo-cyfrowym, w momencie pojawienia się sygnału mierzonego (HIT) zapamiętywana jest faza zegara wzorcowego. Faza zegara w postaci stanów przerzutników D może być zapamiętana w trybie surowym (RAW) z pominięciem dekodery fazy za pomocą dodatkowego modułu, jak również zapamiętana po zdekodowaniu.

Dekodowanie fazy polega na sprawdzeniu, do którego przerzutnika w WLO zdążył dopropagować się stan wysoki sygnału zegara zanim pojawił się sygnał HIT. Od tego bowiem miejsca identyfikowane są przerzutniki z pamiętanym stanem niskim. Pierwszy przerzutnik ze stanem niskim, wskazuje numer przedziału kwantowania odpowiadający fazie zegara. Na tej podstawie po uwzględnieniu charakterystyki przetwarzania możliwe jest uzyskanie informacji na temat opóźnienia mierzonego względem zegara wzorcowego (rys. 27).



Rys. 27. Dekodowanie fazy sygnału zegarowego poprzez detekcję zbocza opadającego, gdy po pamiętanych stanach wysokich pojawiają się stany niskie (a) oraz gdy po początkowych stanach niskich pojawiają się stany wysokie i ponownie stany niskie (b).

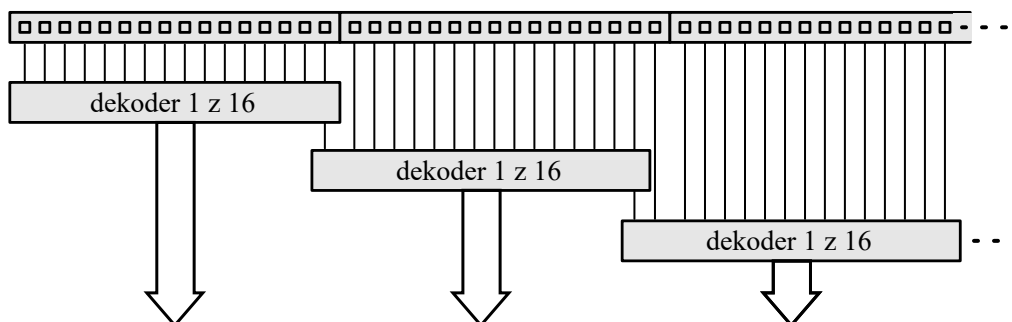
Istotną funkcją dekodera jest ograniczenie liczby danych, które należy przechować w pamięci, a następnie wysłać do komputera w celu ich przetworzenia. Dla systemu zbudowanego z 16 WLO (o 480 odczepach dla każdej WLO) jest to 960 bajtów surowych danych przypadających na pojedyncze pojawienie się sygnału wejściowego HIT. Do zapisania 480 różnych wartości wymagane jest 9 bitowe słowo dwójkowe. Przy zastosowaniu dekodera, który dokona detekcji fazy na jedną z 480 wartości, liczba danych do zapamiętania i transferu zmniejszy się znacząco do $16 \cdot 9 \text{ bitów} = 144 \text{ bity} / 8 = 18 \text{ bajtów}$ co stanowi ponad 50-krotne zmniejszenie ilości danych w stosunku do 960 bajtów na jeden pomiar sygnału HIT (rys. 28).



Rys. 28. Redukcja ilości transmitowanych danych poprzez zastosowaniu konwerterów kodu termometrycznego na binarny

Dekodowanie fazy zegara w sposób sprzętowy może odbywać się na kilka sposobów:

- Pierwsze podejście to zastosowanie na każdą WLO jednego dekodera, który wykrywa przejście w ciągu binarnym stanów logicznych z „1” na „0”. Jest to jednak rozwiązanie mało efektywne w kontekście zaangażowanych zasobów logicznych zwłaszcza dla dużej liczby bitów (rzędu kilkuset).
- Bardziej efektywnym podejściem jest zastosowanie dekodowania z wykorzystaniem rejestru szeregowego. Rozwiązanie to wymaga mniej zasobów, jednak dekodowanie realizowane jest w wielu cyklach przetwarzania (do wykrycia właściwej zmiany przebiegu), co z kolei jest istotnym ograniczeniem w systemach o dużej intensywności impulsów wejściowych.
- Kolejnym rozwiązaniem jest zastosowanie dekodery o prostej budowie, np. dekodujących z kodu „1 z 16” o 4 bitach wyjściowych lub dekodujących z kodu „1 z 32” o 5 bitach wyjściowych, podłączonych równolegle do danej WLO. W celu eliminacji fałszywych detekcji zbocza, wymagane jest częściowe pokrycie zakresów dekodowania (rys. 29), czyli praca na tzw. „zakładkę”.



Rys. 29. Dekodowanie fazy sygnałów jednej WLO za pomocą wielu mniejszych dekodery (dla przykładu dekodery „1 z 16”)

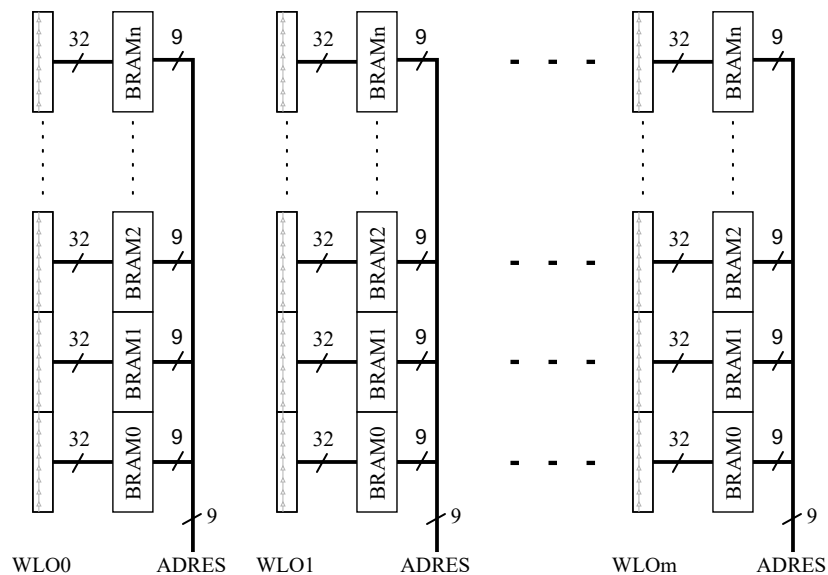
Drugim typem dekodera możliwym do zastosowania jest dekodery programowy. Nie wymaga on dodatkowych zasobów układu scalonego, z wyjątkiem potrzebnej pamięci na przechowywanie surowych danych pomiarowych. Dane te mogą zostać zdekodowane przez mikroprocesor po serii pomiarowej. Jednakże wiąże się to z dłuższym czasem przetwarzania, ale stwarza też możliwość użycia dedykowanych filtrów w celu eliminacji błędów bąbelkowych.

Trzeci sposób realizacji dekodowania fazy sygnału to hybryda dwóch poprzednich rozwiązań. Przykładowo można zastosować dekodery „16 na 4” lub „32 na 5”, podłączone w sposób równoległy do WLO, a następnie zdekodować właściwą fazę z użyciem mikroprocesora.

W celu zmniejszenia objętości danych, niezależnie od wybranego wariantu sprzętowego, układ dekodera powinien zostać zaimplementowany pomiędzy przerzutnikami WLO a pamięcią danych pomiarowych.

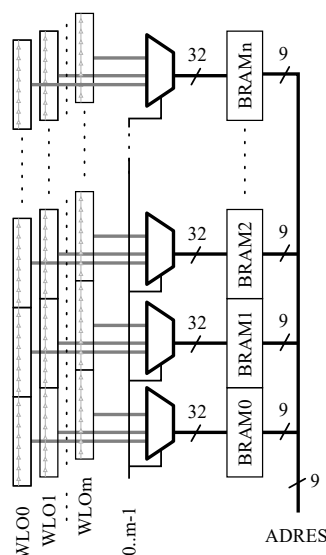
Układ bez dekoderek (tryb RAW) umożliwia pobranie w celu dalszego przetworzenia bezpośrednich danych z przerzutników D w WLO. Przetwarzanie danych obejmuje przede wszystkim procedurę sortowania przerzutników (będzie opisana w rozdziale 4.3.2) oraz ustalenie przesunięć czasowych pomiędzy liniami opóźniającymi zaimplementowanymi w systemie pomiarowym. Po ustaleniu właściwej kolejności (posortowaniu) i właściwym podłączeniu odczepów WLO do dekoderek, moduł trybu RAW można usunąć w celu zaoszczędzenia zaangażowanych zasobów układu oraz obniżenia poboru energii. Wymaga to jednak ponownej syntezy i implementacji całego systemu.

Układ bez dekoderek można zrealizować w dwóch wariantach. Pierwszy wariant jest implementowalny, jeśli układ docelowy posiada dużo wolnych zasobów, w szczególności bloków pamięci BRAM. Wówczas można podłączyć równolegle wejścia danych pamięci BRAM do wyjść każdej WLO (rys. 30).



Rys. 30. Podłączenie pamięci BRAM bezpośrednio do wyjść WLO w trybie RAW z pominięciem dekoderek

Wielkość wymaganych zasobów wynika z liczby WLO w systemie pomiarowym oraz liczby segmentów każdej WLO. Każdy blok pamięci BRAM użyty w układzie posiada pojemność 512 słów o długości 32 bitów. Liczbę wymaganych bloków BRAM można obliczyć dzieląc liczbę odczepów WLO przez 32 (długość pojedynczego słowa w BRAM) i mnożąc przez liczbę użytych WLO (liczba odczepów WLO/32 x liczba WLO). Uzyskaną liczbę należy zaokrąglić w górę do wartości całkowitej. Przykładowo, jeśli zaimplementowana WLO posiada 480 odczepów, to dla jednej WLO potrzebnych jest 15 bloków BRAM. Dla systemu zbudowanego z 16 WLO wymagane jest zatem użycie 240 bloków BRAM. Zastosowany w pracy układ Kintex UltraScale posiada 600 bloków pamięci (BRAM36), które mogą być podzielone na 2 mniejsze (BRAM18), co daje w sumie wolnych nawet 1200 bloków BRAM o pojemności 512 słów 32 bitowych. W wersji oszczędniejszej podłączenie do pamięci odbywa się za pomocą multipleksersa (rys. 31). Wystarczy wówczas użyć tyle



Rys. 31. Podłączenie pamięci BRAM w trybie RAW z pominięciem dekodery
- wersja z multipleksowaniem

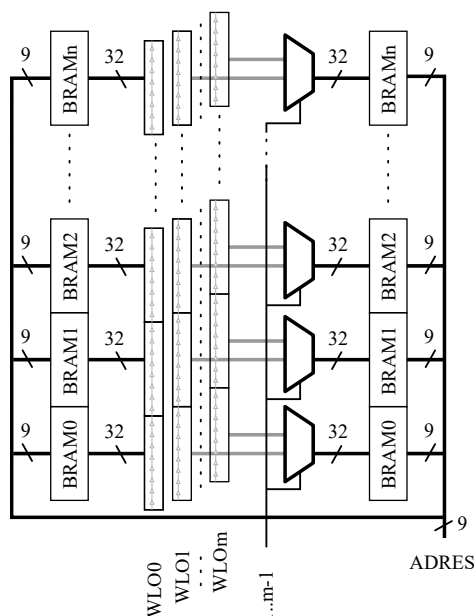
bloków BRAM ile potrzebnych jest dla jednej WLO, po czym za pomocą multipleksersa dołączać kolejne WLO, z których będą pobierane dane. Wadą tego rozwiązania jest wydłużony czas akwizycji. Dostępność nadmiarowej liczby bloków pamięci BRAM umożliwia przeznaczenie dwóch zestawów pamięci na dane RAW. Wówczas przy wsparciu dwóch multipleksersów, każda z pamięci może obsługiwać swoją połowę WLO.

Innym rozwiązaniem jest wykorzystanie dwóch zestawów BRAM oraz jednego multipleksersa. Jeden zestaw BRAM jest bezpośrednio podłączony do WLO traktowanej jako referencyjna do określenia przesunięć czasowych pomiędzy poszczególnymi WLO (np. WLO 0). Natomiast drugi zestaw BRAM podłączany jest poprzez multipleksers do pozostałych WLO (1..N) (rys 32).

Sposób gromadzenia danych z pominięciem dekodery umożliwia pracę z maksymalną prędkością, która ograniczona jest tylko maksymalną częstotliwością bloków BRAM. W zastosowanym układzie FPGA XCKU040-2FFVA1156E częstotliwość taktowania pamięci BRAM wynosi 585 MHz w trybie zapisu bez sumy kontrolnej [17].

Największą zaletą wariantu RAW osobnego dla każdej WLO jest fakt, że przebiegi na wyjściach wszystkich WLO są zapamiętane w jednym momencie w chwili pojawienia się impulsu HIT

i na tej podstawie, niemalże bezzwłocznie, można określić przesunięcia czasowe (offsety) pomiędzy poszczególnymi WLO.



Rys. 32. Podłączenie pamięci BRAM w trybie RAW z pominięciem dekodery - wersja z jednym multiplexerem i dwoma kolumnami BRAM

4.2.5 Matryca kolejności segmentów

Zadaniem matrycy kolejności segmentów (rys. 20) jest podłączenie we właściwej kolejności odczepów wszystkich WLO do ich dekodery fazy po procedurze sortowania (opisanej w rozdziale 4.3). Matryca jest komponentem opisanym w języku VHDL, który zawiera tylko informację o połączeniach wyjść przerzutników WLO z wejściami dekodery i pamięci w trybie RAW. Plik ten generowany jest automatycznie przez oprogramowanie do sortowania segmentów. Przed procedurą sortowania kolejność połączeń wejść i wyjść matrycy pozostaje tożsama. Proces sortowania modyfikuje strukturę matrycy zgodnie z uzyskaną kolejnością segmentów. W celu dokonania połączeń posortowanych segmentów WLO do dekodery w FPGA, należy przeprowadzić syntezę i implementację układu.

4.2.6 Liczniki okresów

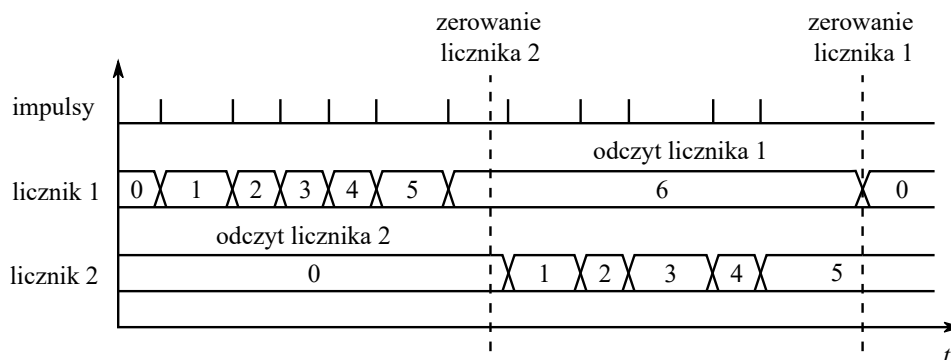
Sygnal zegara podstawy czasu w zaprojektowanym systemie ma częstotliwość 500 MHz. Liczniki (na rys. 20 Licznik1 oraz Licznik2) okresów zegarowych taktowane są dwoma sygnałami wzorcowymi przesuniętymi w fazie o 180°. Liczniki te są identyczne i zbudowane z komponentów DSP48 układu Kintex, skonfigurowanych tak, aby przy zboczu narastającym sygnału zegarowego inkrementowały swoją zawartość o 1. Z bloku DSP48 odczytywane są pierwsze 32 bity rejestru skonfigurowanego jako licznik. W momencie pojawienia się zdarzenia wejściowego HIT, zawartość tych liczników jest zapamiętywana w pamięciach BRAM. Zastosowano dwa liczniki na wypadek, gdyby sygnał wejściowy pojawił się w momencie inkrementacji jednego z nich, co mogłoby spowodować pojawienie się w nim stanów metastabilnych i w efekcie błędne zliczanie. Decyzja o tym, który

licznik zawiera poprawne dane podejmowana jest na podstawie fazy zegara referencyjnego zdekodowanej za pomocą przetwornika.

4.2.7 Moduły pomocnicze

System pomiarowy został wyposażony w dodatkowe moduły pomiarowe, takie jak niezależny licznik impulsów wejściowych oraz częstotściomierz. Służą one do wstępnego określenia parametrów mierzonych odcinków czasu. Oba moduły pomiarowe działają niezależnie i w sposób ciągły, również przed pomiarem właściwych odcinków czasu.

W celu określenia liczby impulsów wejściowych, system pomiarowy został wyposażony w 32 bitowy licznik, który zlicza impulsy wejściowe od momentu zezwolenia na pomiar aż do resetu systemu. Drugim komponentem jest częstotściomierz. Pomiar częstotliwości odbywa się w bramce 1 sekundowej. Częstotściomierz zawiera dwa liczniki 30 bitowe, które działają przeciwnie, tzn. gdy jeden dokonuje pomiaru częstotliwości, z drugiego można odczytać zmierzoną wartość częstotliwości, po czym jest zerowany (rys. 33).



Rys. 33. Zasada działania częstotściomierza z dwoma licznikami

Ma to na celu eliminację błędów, które mogą pojawić się w czasie odczytu zawartości. Maksymalna częstotliwość wejściowa dla licznika impulsów, a zarazem częstotściomierza, wynosi 800 MHz.

4.3 Procedura sortowania segmentów

W przebiegach zegara wzorcowego zarejestrowanych w liniach opóźniających w pobliżu zboczy sygnału zegara, zauważalne są błędy bąbelkowe. Błędy te są wynikiem nierównomierności opóźnień ścieżek rozprowadzających sygnały zegarowe do przerzutników. W celu minimalizacji tych błędów, stosuje się procedurę sortowania, która ma na celu ustalenie odpowiedniej kolejności segmentów linii opóźniającej.

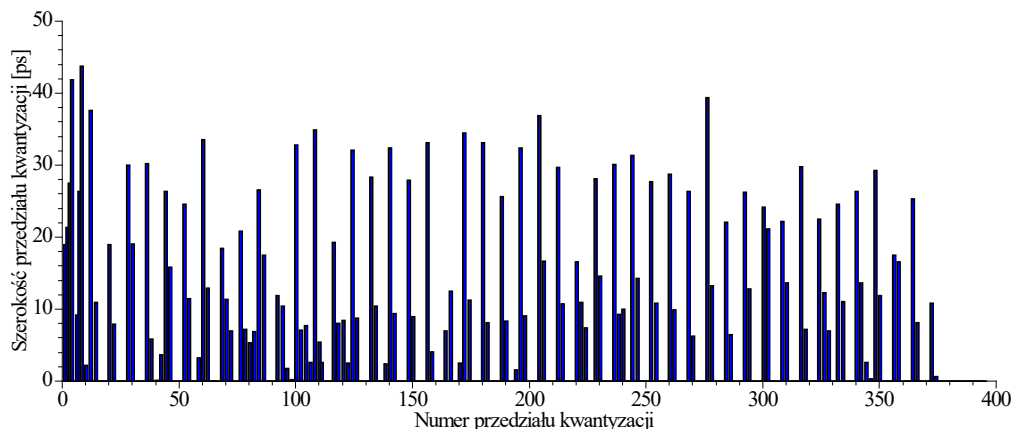
4.3.1 Metody minimalizacji błędu bąbelkowego

W procesie pomiarowym opracowany przetwornik rejestruje przebiegi okresowe zegara wzorcowego tworzące ciągi zer i jedynek. Aby zdekodować fazę zegara stosuje się dekodery. Dekoder identy-

fikuje pierwsze pojawienie się sekwencji „10” lub „01” w zależności od wykrywanego zbocza, i na tej podstawie określa, jaka była faza zegara referencyjnego w momencie pojawienia się sygnału wejściowego (HIT).

W zarejestrowanych przebiegach zegara referencyjnego w WLO implementowanych z użyciem łańcuchów przeniesień arytmetycznych stosunkowo często pojawiają się błędy bąbelkowe. Błąd bąbelkowy odnosi się do niepożądanych sekwencji bitów występujących w wyniku przetwarzania sygnału przez przetworniki błyskawiczne (ang. flash). Źródłem tych błędów są rozrzuty produkcyjne układów scalonych, szumy, zakłócenia, a także nierównomierności opóźnień ścieżek doprowadzających sygnały zegarowe do przerzutników. W efekcie występowania wymienionych czynników może się zdarzyć, że w wyniku przetwarzania zamiast jednolitego ciągu jedynek i następującego po nim ciągu zer na granicy zmian wartości z „1” na „0” (i analogicznie „1” na „0”), pojawiają się tzw. bąbelki, czyli sekwencje typu „101” lub „1001” (jak również przypadki „010” i „0110” itd.). Błędy bąbelkowe dzielimy ze względu na rząd [82], który oznacza błędną liczbę bitów „bąbelka”. Błędy bąbelkowe pierwszego rzędu oznaczają pojedyncze '0' pomiędzy serią bitów '1' lub odwrotnie. Przykładowo ciąg „00001011” ma błąd bąbelkowy pierwszego rzędu (na trzecim bicie od prawej), a w ciągu „00100111” wystąpił błąd bąbelkowy drugiego rzędu. Sposobów eliminacji tego typu błędów jest kilka i zależą one od użytego dekodera.

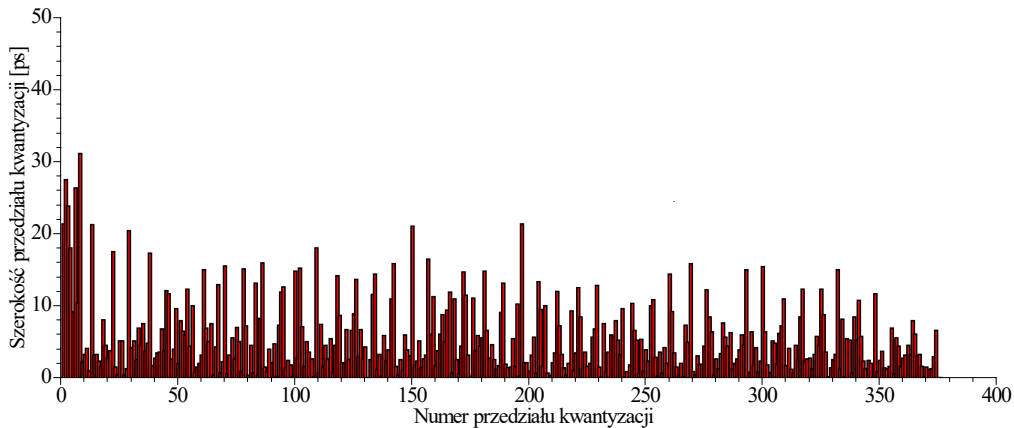
W czasie weryfikacji działania wysokorozdzielczej linii opóźniającej, błąd bąbelkowy będzie widoczny w postaci przedziałów kwantowania bez zarejestrowanych zliczeń, przy dekodowaniu dekodorem. Na rys. 34 przedstawiona jest charakterystyka WLO z nieposortowanymi odczepami. Natomiast na rys. 35 przedstawiona jest charakterystyka tej samej WLO po wykonaniu sortowania odczepów.



Rys. 34. Charakterystyka WLO z nieposortowanymi odczepami. Błędy bąbelkowe objawiają się brakiem zliczeń w niektórych przedziałach kwantyzacji (np. w przedziałach 8, 10, 12, 14–18, 20, 22–26, 28, 30–34 itd.)

Widoczne są różnice w przedziałach kwantowania, ponieważ w nieposortowanej WLO z powodu błędów bąbelkowych dochodzi do błędnego dekodowania i w efekcie fałszywego zliczenia (45 przedziałów o szerokościach powyżej 20 ps) co prowadzi do braku zliczeń w innych przedziałach (250 przedziałów bez zliczeń). Skutkuje to małą liczbą przedziałów, które mają niezerową szerokość (122). Dla posortowanej WLO, przedziałów które mają niezerową liczbę zliczeń jest 375, jednocześnie nie dochodzi do fałszywych zliczeń i w tym przypadku przedziałów o szerokościach po-

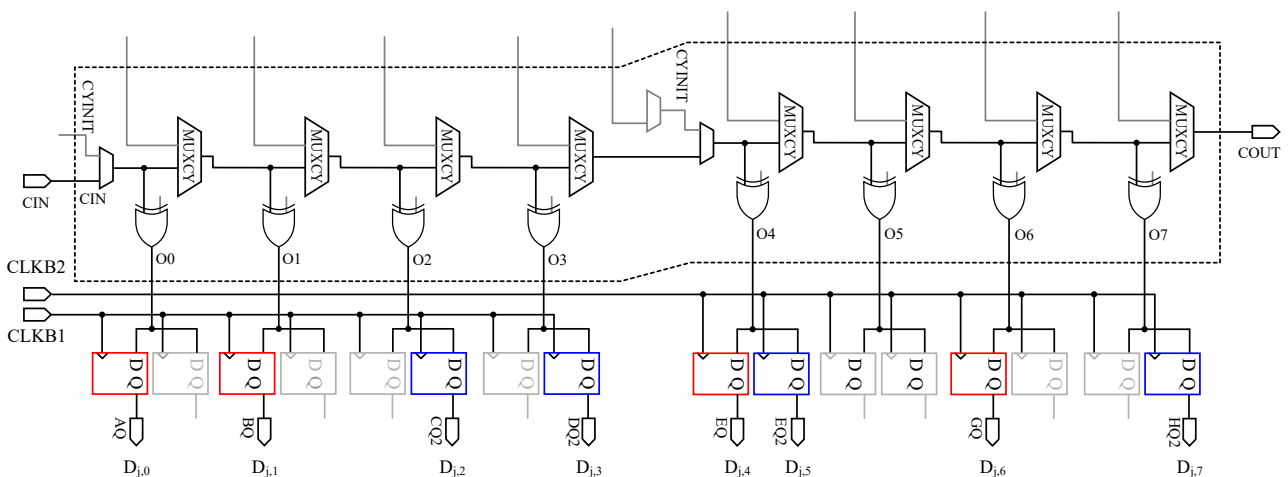
wyżej 20 ps jest znacząco mniej (9), co wpływa na bardziej liniową charakterystykę. Dla nieposortowanych segmentów WLO rozdzielczość ekwiwalentna q_{eqv} wynosi 25,882 ps, natomiast q_{eqv} dla segmentów posortowanych jest równa 12,261 ps, co daje poprawę o 52,6%.



Rys. 35. Charakterystyka WLO z rys. 34 z posortowanymi segmentami

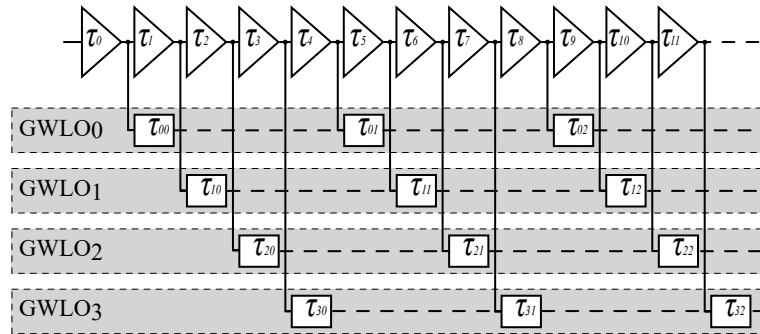
W przypadku występowania błędów bąbelkowych, istnieje możliwość ich redukcji, co na ogół skutkuje zwiększeniem rozdzielczości układu pomiarowego. Eliminację błędu bąbelkowego można bowiem osiągnąć np. poprzez uwzględnienie tylko co n -tych odczepów linii opóźniającej lub wybór takich odczepów, które nie generują błędów bąbelkowych. Metoda ta może mieć zastosowanie dla linii opóźniającej o niskiej rozdzielczości.

W przypadku WLO implementowanej w układzie Kintex UltraScale (dla 480 segmentów), istnieje możliwość wyboru przerzutnika spośród dostępnej pary przerzutników, które są połączone z tym samym odczepem łańcucha przeniesień arytmetycznych CARRY8. Można też pominąć dany odczep w parze przerzutników zachowując liczbę użytych przerzutników, co przedstawiono na rys. 36.



Rys. 36. Fragment WLO zbudowanej z wybranych przerzutników, zaznaczonych kolorami czerwonym i niebieskim dla układu Kintex UltraScale w jednym CLB, w celu eliminacji błędu bąbelkowego. Kolorem czerwonym zaznaczono przerzutniki, które wybrane są automatycznie w procesie implementacji, a kolorem niebieskim przerzutniki wybrane w celu eliminacji błędu bąbelkowego

Kolejnym sposobem eliminacji błędu bąbelkowego jest zastosowanie dekodera grzebienio-
wego [62]. Zasada jego działania polega na wybraniu co n -tego segmentu, aby w wynikowych grze-
bieniowych WLO (GWLO) nigdy nie wystąpił błąd bąbelkowy. Dobór n wynika z maksymalnego
rzędu występującego błędu bąbelkowego, tj. $n >$ rzędu błędu bąbelkowego (rys. 37).



Rys. 37. Zasada tworzenia grzebieniowych WLO. W wyniku wyboru co czwartego segmentu
powstały cztery GWLO w celu eliminacji błędu bąbelkowego trzeciego rzędu

Przyjmując, że dana WLO składa się z m segmentów, a n oznacza liczbę utworzonych grzebienio-
wych wielosegmentowych linii opóźniających (GWLO), to nowo powstała $GWLO_k$ będzie posiada-
ła m/n segmentów. Segmenty w k -tej GWLO będą składać się z segmentów WLO wybranych zgo-
dnie ze wzorem (13):

$$GWLO_k(i) = WLO(i \cdot n + k) \quad (13)$$

gdzie:

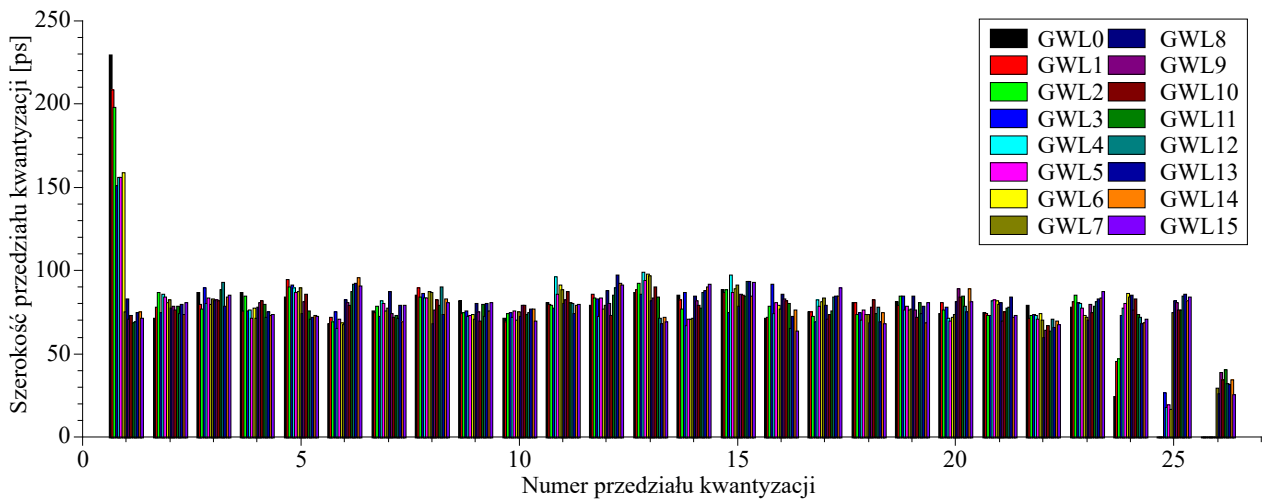
i - oznacza nowy segment grzebieniowej WLO i zawiera się w przedziale $\langle 0, (m/n) - 1 \rangle$, k zawiera
się w przedziale $\langle 0, n - 1 \rangle$ i oznacza numer kolejnej GWLO. Parametry najlepiej dobrać tak, aby ilo-
raz m/n był wartością całkowitą oraz aby n było większe niż rząd błędu bąbelkowego występujący
w WLO. Uzyska się wtedy n GWLO o jednakowej liczbie segmentów (m/n). W każdej GWLO za-
rejestrowany będzie przebieg bez widocznego błędu bąbelkowego, ponieważ w GWLO następuje
wydłużenie interwałów czasu na jej kolejnych odczepach.

W efekcie badań eksperymentalnych, przyjęto, że w opracowanym układzie dla WLO
o $m=480$ segmentów optymalna wartość n wynosi 16. Jest to wartość, dla której nigdy nie wystąpił
błąd bąbelkowy dla nieposortowanych segmentów WLO. Jednocześnie dla takiej wartości n wszyst-
kie powstałe GWLO będą posiadały jednakową liczbę segmentów. W wyniku takiego działania
uzyskano 16 GWLO o 30 segmentach. W celu zdekodowania fazy zegara wystarczy n dekodery
priorytetowych w kodzie „1 z 30”. Dodatkowo taki zestaw dekodery będzie zajmował mniej zasob-
ów, niż jeden dekoder priorytetowy, dekodujący całą WLO o m segmentach. Ponadto dzięki pro-
stocie układu zwiększa się szybkość przetwarzania.

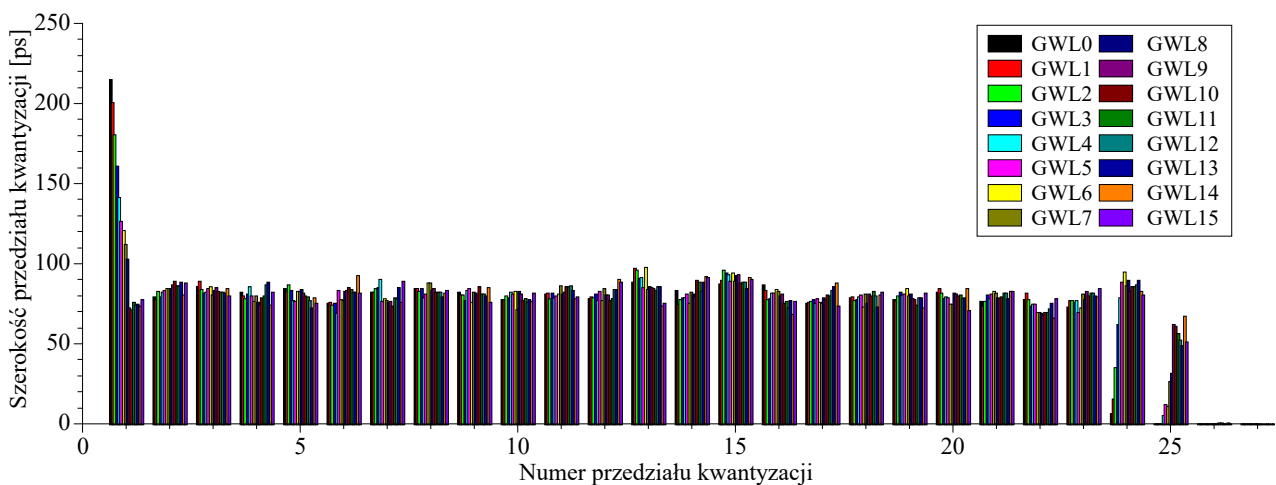
Dla porównania, dekoder, priorytetowy zaimplementowany w układzie Kintex UltraScale
dla 480 segmentów składał się z 864 LUT, natomiast po użyciu 16 dekodery „1 z 30”, taki zestaw
dekodery zajmował 588 LUT. Wadą dekodowania grzebieniowego jest zwiększone zapotrzebowanie
na pamięć do przechowywania danych pomiarowych. Przy jednym dekoderze „1 z 480” wynik

jest 9 bitowy, dla 16 dekodowników „1 z 30” potrzeba $16 \times 5 = 80$ bitów wyniku. Przykładowe charakterystyki szerokości kanałów dla GWLO zamieszczono na rys. 38 oraz rys. 39.

Możliwe jest połączenie obu opisanych poprzednio metod bazujących na sortowaniu segmentów oraz zastosowaniu dekodera grzebieniowego. Dzięki temu powstałe GWLO mają bardziej równomierne charakterystyki (rys. 39), a ich liczba podziału n może zmaleć, ze względu na znacznie mniejsze rzędy błędów bąbelkowych.



Rys. 38. Charakterystyka szerokości kanałów szesnastu GWLO dla nieposortowanej bazowej WLO. Kolor słupka oznacza przynależność do danej GWLO



Rys. 39. Charakterystyka szerokości kanałów szesnastu GWLO dla posortowanej bazowej WLO wykazuje zauważalne wygładzenie w porównaniu z charakterystyką na rys. 38

W wyniku porównania rysunków 38 i 39, dostrzec można drobne różnice w otrzymanych szerokościach przedziałów kwantowania. W porównaniu do nieposortowanej bazowej WLO (rys. 38) w wyniku sortowania zmniejszyły się dysproporcje pomiędzy maksymalnymi szerokościami przedziałów kwantowania, zwłaszcza w początkowych przedziałach q_i . Jednak poprawa ta jest znacznie mniej zauważalna w porównaniu do dekodowania stanów nieposortowanej i posortowanej WLO (rysunki 34 i 35) bez użycia GWLO. Zastosowanie GWLO eliminuje więc konieczność sortowania odczepów WLO. Upraszcza to znacząco proces produkcji seryjnej urządzeń opartych na

układach FPGA wykorzystujących WLO do pomiarów czasowych. Nie wymaga on bowiem przeprowadzania ponownego procesu implementacji matrycy połączeń dla każdego egzemplarza układu FPGA w celu modyfikacji połączeń pomiędzy WLO a dekoderm fazy zegara. Powyższa procedura jest szczególnie ważna z uwagi na wynikające z rozrzutów technologicznych różnice w charakterystykach WLO implementowanych w różnych egzemplarzach tej samej serii układów FPGA.

4.3.2 Przebieg procesu sortowania segmentów

Innym sposobem eliminacji błędów bąbelkowych jest zastosowanie procedury sortowania odczepów linii opóźniającej. Ma to szczególne znaczenie dla wysokorozdzielczych linii opóźniających zbudowanych z użyciem łańcucha szybkich przeniesień arytmetycznych, gdzie występowanie błędów bąbelkowych jest praktycznie zasadą a nie wyjątkiem (rys. 34).

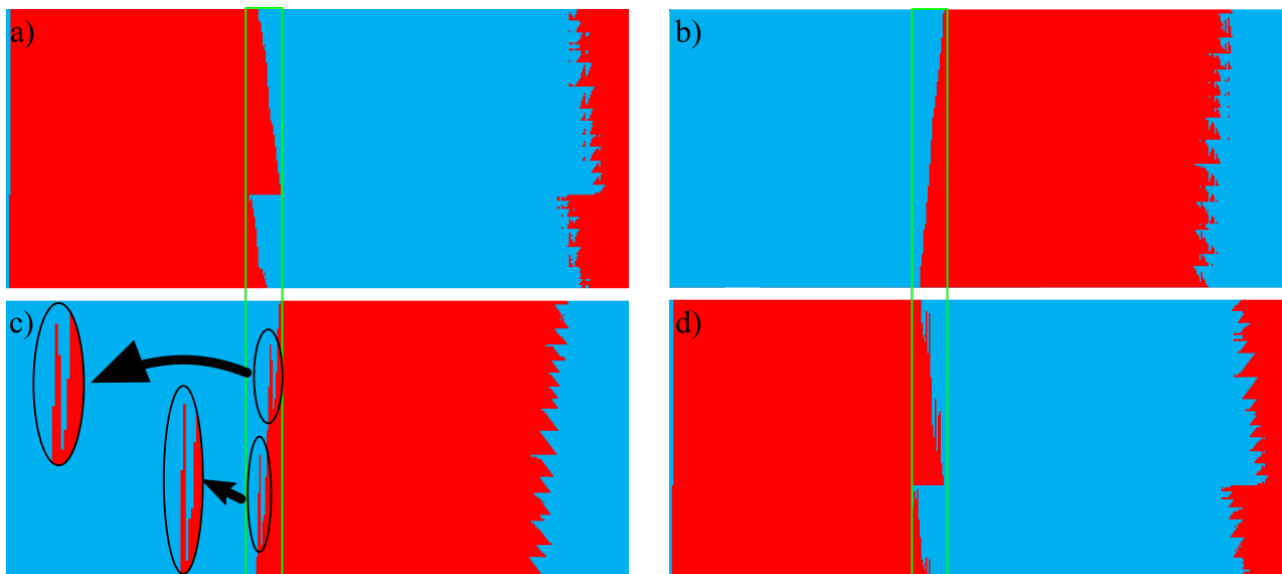
Kalibracja WLO i sortowanie jej odczepów w większości przypadków polega na doprowadzeniu do badanej linii opóźniającej sygnałów START i STOP z precyzyjnego generatora odcinków czasu lub sygnału HIT opóźnionego względem zegara referencyjnego [84, 85].

W badanym systemie pomiaru odcinka czasu utworzono autorską procedurę sortowania segmentów. Zawiera ona dwa etapy. W pierwszym etapie wszystkie przerzutniki linii opóźniającej podłączone są bezpośrednio do pamięci (tryb RAW), a do wejścia pomiarowego systemu (HIT) doprowadzony jest sygnał z generatora, który nie jest skorelowany z zegarem systemowym. W tym celu można wykorzystać programowalny generator na karcie KCU105. W trybie akwizycji danych RAW pomiary rejestrowane są z pominięciem dekodów. Następnie inicjowany jest tryb rejestracji impulsów z generatora, a przebiegi zegara wzorcowego zarejestrowane w przerzutnikach zapamiętywane są w pamięci blokowej BRAM. Po zapełnieniu pamięci BRAM jej zawartość jest kopiowana do zewnętrznej pamięci DDR w celu późniejszego przetworzenia w komputerze PC. Po zarejestrowaniu pomiarów w liczbie rzędu 10^6 - 10^8 , dane transmitowane są do komputera PC (po kompresji), gdzie autorskie oprogramowanie wykonuje procedurę sortowania. W wyniku działania programu uzyskana jest zmiana numeracji wyprowadzeń linii opóźniającej. Taką linię opóźniającą można następnie podłączyć do dekodera za pomocą zmodyfikowanej matrycy połączeń.

W cyfrowych liniach opóźniających zbocza narastające i opadające sygnałów propagują się z różnymi opóźnieniami [62, 80]. Jest to wyraźnie zauważalne w liniach opóźniających opartych na łańcuchu przeniesień arytmetycznych zaimplementowanych w układzie Kintex UltraScale. W przypadku dekodowania obydwu zboczy, należałoby zbudować dwa dekodery i procedurę sortowania przeprowadzić dwukrotnie odpowiednio dla rodzaju zbocza. Albowiem posortowanie komórek dla jednego zbocza generować będzie błędy bąbelkowe dla przeciwnego typu zbocza (rys. 40).

Na rysunku 40 widoczny jest fragment zarejestrowanych przebiegów zegara wzorcowego. W wierszach znajdują się przebiegi po procedurze ich sortowania oraz po usunięciu zduplikowanych przebiegów. W kolumnach znajdują się kolejne stany przerzutników po procedurze ich sortowania ze względu na zbocze. Na rysunkach znajdują się przebiegi po procedurze sortowania segmentów odpowiednio dla zbocza opadającego (rys. 40a) i narastającego (rys. 40b). W powiększeniu (rys. 40c) widoczne są powstałe błędy bąbelkowe w wyniku zarejestrowania zbocza narastającego przez segmenty posortowane dla zbocza opadającego. Analogicznie błędy bąbelkowe widoczne są

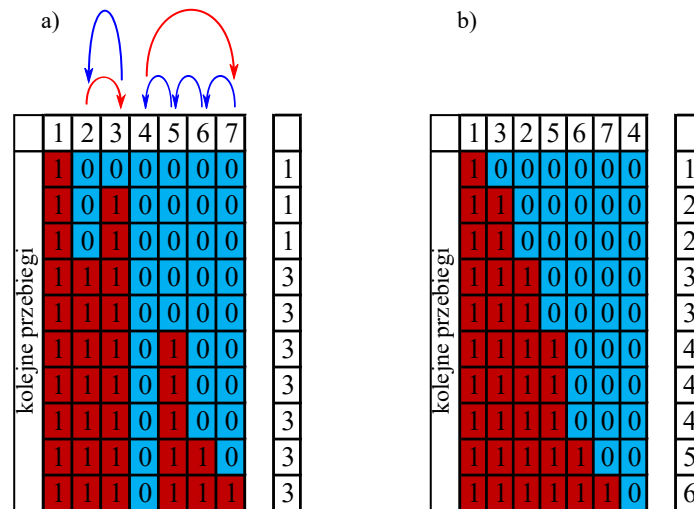
dla przypadku gdzie zarejestrowano zbocze opadające przez segmenty posortowane dla zbocza narastającego (rys. 40d). Wpływ na to mają różnice w szybkości propagacji zboczy, jak również charakterystyki przerzutników implementowanych w układach scalonych.



Rys. 40. Przebiegi zegara po procedurze sortowania segmentów WLO ze względu na zbocze: a) zbocze opadające zarejestrowane dla posortowanych segmentów dla zbocza opadającego, b) zbocze narastające zarejestrowane dla posortowanych segmentów dla zbocza narastającego, c) widoczny błąd bąbelkowy wynikający z zarejestrowania zbocza narastającego dla segmentów posortowanych dla zbocza opadającego, d) błąd bąbelkowy wynikający z zarejestrowania zbocza opadającego dla segmentów posortowanych dla zbocza opadającego. Kolor czerwony oznacza stan wysoki „1”, niebieski – stan niski „0”. Widoczne ząbki są wynikiem nierównomiernego wypełnienia przebiegu zegara.

Po zgromadzeniu w rejestrach informacji o przebiegach zegara wzorcowego, zapisaniu ich w pamięci oraz przesłaniu do komputera następuje drugi etap procedury sortowania, a mianowicie właściwe jej sortowanie odczepów. Procedurę można przeprowadzić manualnie lub automatycznie z użyciem dedykowanego programu. Na początku sortowania ręcznego dane binarne zamieniane są na linie tekstu w kodach ascii, przy czym jedna linia tekstu zawiera ciąg 480 zer i jedynek odpowiadający jednemu stanowi przerzutników dla jednego sygnału HIT. Następnie linie tekstu są sortowane. Przebieg zaczynający się od „1000...” zostanie umiejscowiony wyżej niż przebieg „1100...” itd. wskutek sortowania tekstu. Dla zbocza narastającego zegara wzorcowego wyszukiwane są przebiegi, w których ciąg stanów logicznych „1” przechodzi w ciąg stanów „0”. Wówczas otrzymuje się w wierszach kolejno przebiegi, które mają minimalnie różniące się między sobą fazy zegara systemowego lub identyczne fazy zegara, zarejestrowane w WLO. Następnie tekst jest dzielony na kolumny o szerokości jednego znaku i importowany do arkusza kalkulacyjnego. Dodatkowo w kolumnach arkusza znajdują się numery odczepów/segmentów WLO w kolejności narastającej. Cała idea tej metody sortowania polega na założeniu, że sygnał zegara wzorcowego zmienia swój stan co najwyżej kilka razy w całej długości WLO i jego wypełnienie zbliżone jest do 50% (bez uwzględnienia charakterystyki danej WLO). Każde odstępstwo od tej reguły, szczególnie w pobliżu zbocza rzeczywistego sygnału zegara oznacza, że segment jest na niewłaściwej pozycji. Jeśli porówna się wiele przebiegów, można zauważyć błąd bąbelkowy w danej kolumnie lub kolumnach. W sytuacji, gdy

zamieni się odpowiednie kolumny miejscami lub przesunie je, następuje eliminacja błędów bąbelkowych (rys. 41). Na rysunku 41a pokazany jest fragment przebiegów w WLO przed procedurą sortowania.



Rys. 41. Idea procedury sortowania segmentów. Fragmenty przebiegów w WLO dla nieposortowanych segmentów (a) oraz fragmenty przebiegów po sortowaniu (b). Z prawej strony podane są wartości zdekodowane przez dekoder priorytetowy, a strzałki u góry określają prawidłową kolejność zamiany segmentu

W kolumnach nr 2 oraz nr 4 widoczne są błędy bąbelkowe. Aby usunąć błąd bąbelkowy występujący w kolumnie nr 2 wystarczy zamienić miejscami kolumnę nr 2 z kolumną nr 3. Natomiast aby usunąć błąd bąbelkowy z kolumny nr 4 należy przesunąć ją na koniec w miejsce kolumny nr 7, a pozostałe kolumny od nr 5 do nr 7 przesunąć w lewo o jedną kolumnę. W wyniku operacji przesunięć i zamian otrzymano przebieg jak na rysunku 41b. W górnym wierszu pokazana jest kolejność segmentów po procedurze sortowania.

W czasie procedury sortowania identyfikuje się również segmenty metastabilne. Segmentów takich nie można jednoznacznie przypisać do jednego konkretnego miejsca w linii, gdyż w zależności od procesu metastabilnego mogą reprezentować pozycję N , albo pozycję $N \pm 1$. W takim przypadku program sprawdza częstość wystąpień pozycji segmentu (przykładowo N i $N+1$) i dokonuje takiego wyboru pozycji, aby otrzymać dwa sąsiednie segmenty o możliwie zbliżonych szerokościach (zamiast znacznie różniących się). Skutkuje to tym, że cała linia opóźniająca charakteryzuje się mniejszą wartością rozdzielczości ekwiwalentnej (rys. 42).

Odczyt z takiego segmentu może być pominięty, aby nie wprowadzał błędów. Jednak na ogół nie jest to czynione, aby nie pogarszać rozdzielczości. Na linię złożoną z 480 segmentów, przy aktywnych około 390 segmentach, należałoby bowiem usunąć od kilku do kilkunastu z nich (w zależności od położenia WLO w strukturze układu).

	1	2	3	4	5	6	7	8		
kolejne przebiegi	1	1	0	0	0	0	0	0		2
	1	1	0	0	0	0	0	0		2
	1	1	0	1	0	0	0	0		4
	1	1	0	1	0	0	0	0		4
	1	1	1	0	0	0	0	0		3
	1	1	1	1	0	0	0	0		4
	1	1	1	1	1	0	0	0		5
	1	1	1	1	1	0	0	0		5
	1	1	1	1	1	1	0	0		6
	1	1	1	1	1	1	1	0		7

	1	2	4	3	5	6	7	8		
kolejne przebiegi	1	1	0	0	0	0	0	0		2
	1	1	0	0	0	0	0	0		2
	1	1	1	0	0	0	0	0		3
	1	1	1	0	0	0	0	0		3
	1	1	0	1	0	0	0	0		4
	1	1	1	1	0	0	0	0		4
	1	1	1	1	1	0	0	0		5
	1	1	1	1	1	0	0	0		5
	1	1	1	1	1	1	0	0		6
	1	1	1	1	1	1	1	0		7

Rys. 42. Idea procedury sortowania segmentów metastabilnych. Fragmenty przebiegów posortowanych przed zamianą (a) oraz po zamianie kolejności segmentu 4 z 3 w celu zmniejszenia wpływu błędów bąbelkowego. Z prawej strony podane są wartości zdekodowane przez dekodery priorytetowy.

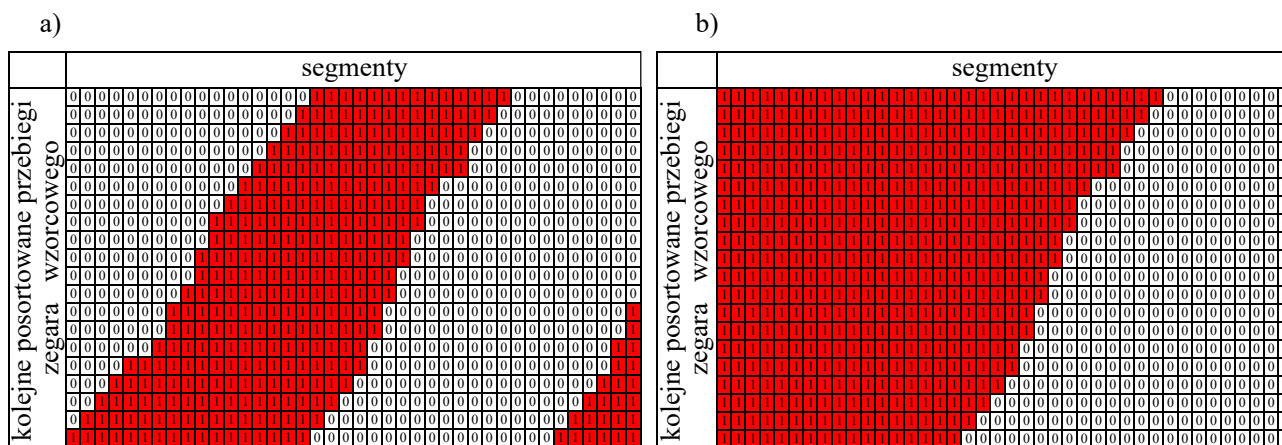
Zasady zamiany kolumn/segmentów w procedurze sortowania są następujące:

- należy zamieniać kolumny najbliższe, znajdujące się obok siebie,
- należy sortować w odniesieniu do jednego zbrocza,
- w sytuacji, gdy segment pasuje do dwóch miejsc (co generalnie jest bardzo rzadkie), należy przyjąć zasadę, że segment o niższym numerze jest umieszczany przed segmentem o wyższym numerze lub wybrać taką pozycję segmentu, aby minimalizować występowanie błędów bąbelkowego,
- jeśli błędy bąbelkowe występują na dwóch pozycjach, należy wybierać taką konfigurację segmentów, która umożliwi jak największą liczbę zliczeń dla nowo powstałego przedziału kwantowania, a segment zawierający błąd bąbelkowy zostanie przyporządkowany do poprzedniego segmentu przez dekodery (rys. 42).

W celu ułatwienia manualnego sortowania segmentów, można zmodyfikować otrzymane przebiegi przed właściwą procedurą sortowania. Dla zbrocza narastającego zegara wzorcowego (czyli przejścia z "0" na "1"), można w zerojedynkowym odwzorowaniu przebiegów (rys. 43) zamienić wszystkie początkowe zera na jedynki. Po wystąpieniu przejścia z "1" na "0" od momentu wystąpienia samych zer, zamienia się z kolei pozostałe jedynki na zera.

Do przeprowadzenia manualnej procedury sortowania wskazane jest użyć arkusza kalkulacyjnego, w którym wszystkie segmenty i bity przebiegu umieszczone są w kolumnach.

W celu automatyzacji procesu sortowania opracowany został skrypt w języku Python, znacząco przyspieszający sortowanie, które w zależności od wydajności komputera i liczby danych do przetworzenia trwa od kilku do kilkunastu sekund. Jest to istotne skrócenie w porównaniu do metody manualnej, w której sortowanie może trwać (w zależności od użytego arkusza kalkulacyjnego, liczby segmentów, liczby danych oraz wprawy użytkownika), co najmniej kilkanaście minut dla jednej WLO.



Rys. 43. Procedura wypełniania przebiegów. Przebieg „surowy” z wieloma zboczami (a) oraz ten sam przebieg po wypełnieniu z jednym zboczem (b).

Dodatkowo opracowany skrypt wyszukuje segmenty metastabilne (potencjalnie pasujące do dwóch sąsiednich miejsc) i wybiera odpowiednie ich położenie. Kryterium dopasowania jest najrzadsze występowanie błędu bąbelkowego danego segmentu poprzez sprawdzenie częstości wystąpień tych błędów na podstawie wcześniej zarejestrowanych przebiegów zegara. Na końcu działania algorytmu generowany jest plik kolejności segmentów oraz macierz połączeń w języku VHDL. Kod macierzy może być wykorzystany w projekcie systemu, a system zsyntetyzowany i zaimplementowany w układzie FPGA. Jeśli użyty jest dekodery programowy plik właściwej kolejności segmentów może być zaimportowany do kodu programu obsługi systemu dla mikroprocesora.

W celu skróceniu czasu transmisji danych do komputera dane gromadzone w pamięci zewnętrznej DDR systemu podlegają kompresji. W związku z tym została utworzona tablica programowa, która posiada dwa pola. Pierwsze pole to zarejestrowany przebieg, a drugie pole to liczba wystąpień tego przebiegu w danych zgromadzonych w pamięci DDR. Dzięki czemu powstaje rodzaj histogramu przebiegów. Pole przebieg posiada 60 bajtów długości, a liczba wystąpień jest 4 bajtowa. Rozmiar tablicy dany jest następującym wzorem:

$$\text{Długość tablicy} = (N + 1) \cdot 64 \text{ Bajty} , \quad (14)$$

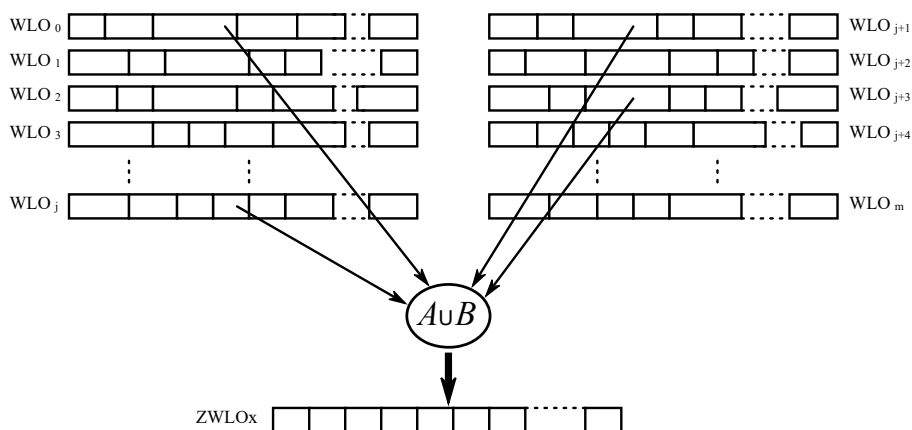
gdzie $(N+1)$ oznacza liczbę niepowtarzalnych przebiegów zarejestrowanych w tablicy. Jeśli dane gromadzone są dla wielu WLO, liczba takich tablic jest równa liczbie WLO w systemie pomiarowym.

W programie mikroprocesora MicroBlaze istnieje procedura sprawdzająca czy w tablicy przebiegów (pamięci danych) jest już zarejestrowany przebieg (dla danej WLO) i w takim przypadku inkrementowane jest pole liczby wystąpień przebiegu. W przeciwnej sytuacji, nowy przebieg dodawany jest na końcu tablicy i zwiększany jest rozmiar tej tablicy. Oszczędność w magazynowaniu danych zależy od liczby zgromadzonych przebiegów i wynosi od kilku do kilkudziesięciu razy w stosunku do rejestracji bez zmniejszenia objętości w trybie RAW. Wadą opisanego rozwiązania jest wydłużony czas gromadzenia danych oraz utrata informacji o zależnościach przebiegów pomiędzy poszczególnymi WLO (przesunięcia fazy), jeśli tryb RAW jest stosowany dla wszystkich 16 WLO jednocześnie.

5. Algorytm doboru wielokrotnych linii opóźniających

Środowiska programistyczne do projektowania z użyciem układów programowalnych FPGA umożliwiają, w pewnym zakresie, wybór miejsca implementacji w układzie FPGA docelowego układu cyfrowego lub jego wybranych podzespołów. Z uwagi na wynik realizowanych równań logicznych rozlokowanie elementów układu nie ma wpływu na wynik operacji, wpływa jednak na opóźnienia w rozprowadzeniu sygnałów, co może wpływać na przykład na wartość maksymalnej częstotliwości pracy implementowanego układu cyfrowego. Aby uzyskać odpowiednie parametry częstotliwościowo-czasowe oprogramowanie implementujące planuje najkorzystniejsze rozlokowanie komponentów w układzie FPGA z uwzględnieniem wytycznych projektanta, takich jak na przykład możliwie małe użycie zasobów, wysoka częstotliwość pracy, albo wręcz przeciwnie, rozproszony obszar implementacji w celu równomiernego nagrzewania się pracującego układu kosztem zmniejszenia częstotliwości. Ponadto parametry użytych elementów układu FPGA mogą mieć różne wartości ze względu na rozrzuty technologiczne, strukturę układu programowalnego, sposób prowadzenia metalizacji, itp. Jak wcześniej wspomniano, zaimplementowane WLO bazująca na komponentach w szczególności elementach łańcucha przeniesień arytmetycznych, w sposób niestandardowy. W niniejszej pracy zaimplementowano WLO w różnych miejscach układu, co znacząco wpływa na charakterystyki WLO. Głównie oddziałuje to na nieliniowości i czas propagacji zegara wzorcowego w danej WLO, co przejawia się różną liczbą aktywnych kanałów czasowych danej WLO, tożsamy z przedziałami kwantowania.

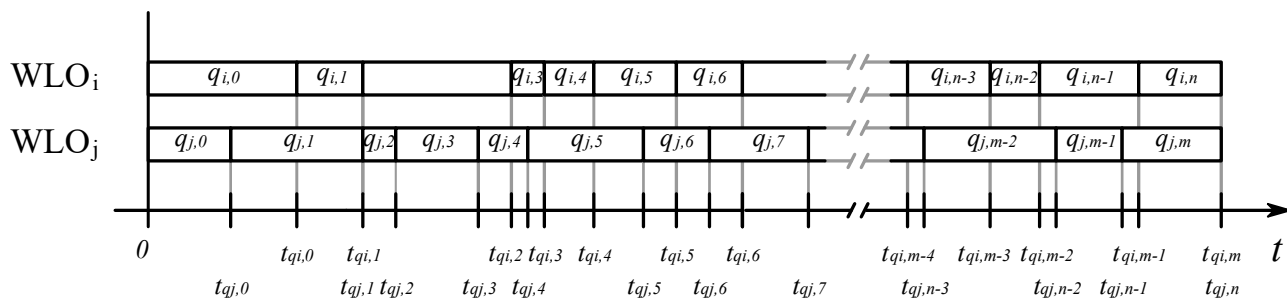
W systemie z wieloma WLO, z posiadanego zbioru W WLO o określonych charakterystykach, można dokonać operacji wirtualnego połączenia wybranych WLO w jedną złożoną WLO (ZWLO) w celu zwiększenia rozdzielczości i poprawy liniowości (rys. 44).



Rys. 44. Idea metody utworzenia ZWLO z wybranych WLO

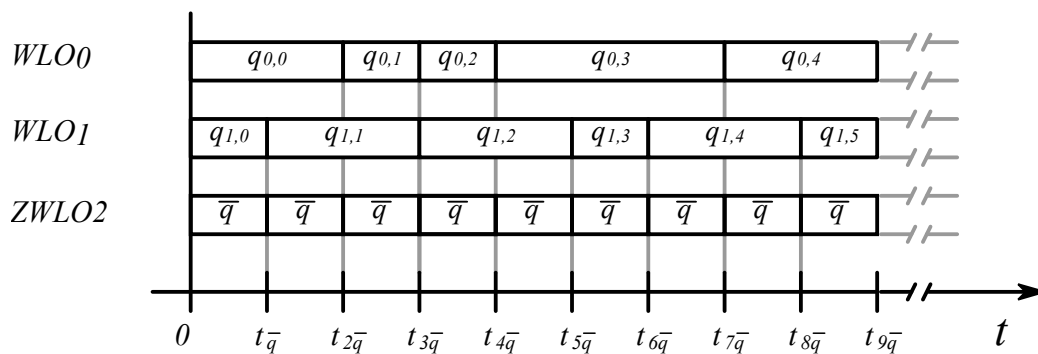
W ten sposób można utworzyć ekwiwalentną linię kodującą lub też, poprzez wybór odpowiednich odczepów ZWLO, podłączyć w odpowiedniej kolejności do dekodatorów w celu zdekodowania fazy zegara wzorcowego. To złożenie może odbyć się fizycznie, w wyniku wyboru w projekcie konkretnych odczepów, lub może być wykonane programowo, na podstawie danych wysłanych z systemu pomiarowego (tak jak np. w ekwiwalentnej linii kodującej ELK rozdział 2.1.4).

Cechą charakterystyczną WLO zbudowanych z użyciem łańcucha przeniesień arytmetycznych (ang. *carry-chain*) są segmenty linii o naprzemiennie dużym i małym opóźnieniu (rys. 35 i rys. 45), które wyraźnie odstają od wartości średniej arytmetycznej opóźnienia dla danej WLO. Dla układu Kintex UltraScale XCKU040 średnia ta wynosi około 5,3 ps. Powoduje to powstanie WLO o charakterystykach silnie nieliniowych. Ideałem byłoby, gdyby do wybranej WLO dobrać inną WLO, która będzie miała dopełniającą charakterystykę, aby w wyniku połączenia obu linii uzyskać możliwie równe przedziały kwantowania.



Rys. 45. Ilustracja szerokości kanałów w WLO implementowanej w FPGA z użyciem komponentów łańcucha przeniesień arytmetycznych

W praktyce uzyskanie linii idealnej, o jednakowych przedziałach kwantowania jest oczywiście niemożliwe. Jednak można podjąć próbę minimalizacji nieliniowości systemu pomiaru odcinka czasu wykorzystującego przetworniki zbudowane z użyciem selekcyonowanych WLO. Dlatego w niniejszej pracy zaproponowano algorytm doboru linii pojedynczej bądź wielu linii w celu polepszenia parametrów przetwornika takich jak rozdzielczość i liniowość. W zaproponowanym algorytmie dąży się do uzyskania możliwie równomiernych przedziałów kwantowania poprzez dobór WLO o uzupełniających się charakterystykach (rys. 46).



Rys. 46. Połączenie dwóch WLO o nieliniowych lecz dopełniających się charakterystykach, w wyniku czego powstaje złożona WLO (ZWLO) o charakterystyce idealnie liniowej

5.1 Kryteria wyboru wielokrotnych linii opóźniających

Dobór WLO może zostać przeprowadzony zarówno w celu spełnienia określonych parametrów metrologicznych systemu, jak i uzyskania zakładanej architektury. Kryteria doboru związane z poprawą parametrów metrologicznych dotyczą:

- liniowości przetwarzania, będącej parametrem szczególnie istotnym w systemach, w których nie uwzględnia się kształtu charakterystyki systemu pomiarowego przy obliczaniu wyniku pomiaru, a zakłada się wartość maksymalną błędu wynikającego z nieliniowości sumacyjnej i za wynik przyjmuje się iloczyn wartości średniej przedziału kwantowania i numeru przedziału kwantowania. W tego typu systemach pomiarowych zakłada się, że różnica w szerokościach przedziałów kwantowania nie powinna przekraczać 10% średniej szerokości przedziału kwantowania,
- rozdzielczości ekwiwalentnej - w przypadku przetworników realizowanych w układach FPGA, charakterystyki kanałów pomiarowych są silnie nieliniowe i jest to bardziej reguła niż wyjątek. Miarą ilustrującą jakość takiego przetwornika, lepiej niż średnia wartość rozdzielczości jest rozdzielczość ekwiwalentna (15):

$$q_{eqv} = \sqrt{\frac{1}{T_0} \sum_i q_i^3} \quad (15)$$

Odpowiada ona wartości rozdzielczości średniej idealnego przetwornika, którego błąd kwantyzacji jest równy błędowi kwantyzacji tego przetwornika dla rzeczywistych przedziałów kwantowania [83].

Przedstawione powyżej kryteria pozwalają określić potencjalne możliwości budowy systemu o zakładanych parametrach metrologicznych. Domyślnym kryterium doboru związanym z poprawą parametrów metrologicznych jest rozdzielczość ekwiwalentna, ale stosunkowo łatwo można osiągnąć poprawę parametrów systemu w zakresie liniowości przetwarzania, a nawet obydwu kryteriów jednocześnie.

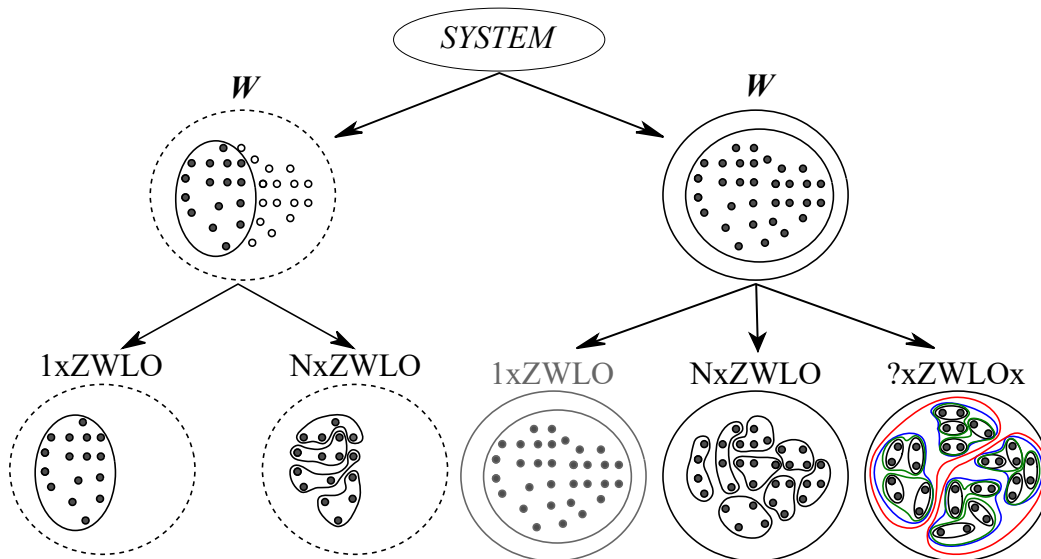
Kryteria związane z projektowaniem architektury systemu pomiarowego pozwalają na wybór jednej z poniższych konfiguracji:

- system z pojedynczym kanałem pomiarowym,
- system wielokanałowy o nieziennej konfiguracji (stała liczba kanałów o stałych parametrach),
- system z rekonfigurowalnymi kanałami pomiarowymi o parametrach zależnych od ich liczby, (takie rozwiązania stosuje się w oscyloskopach cyfrowych, gdzie parametry pomiarowe kanału poprawiane są kosztem liczby kanałów).

Dodatkowym kryterium dla systemów wielokanałowych jest uzyskanie zbliżonych wartości rozdzielczości ekwiwalentnej wszystkich kanałów.

Priorytetyzacja przedstawionych kryteriów prowadzi do różnych konfiguracji projektowanych systemów pomiarowych. Istotnym ograniczeniem w procesie projektowania pozostaje liczba

dostępnych zasobów układu. W zależności od tych zasobów oraz od narzuconych przez projektanta kryteriów, docelowy system pomiarowy może zostać zbudowany na bazie wszystkich lub części zaimplementowanych WLO. Jeśli w systemie pomiarowym algorytm może wykorzystać tylko określoną liczbę z całego zbioru WLO, posiada większą swobodę wyboru, ponieważ część WLO nie zostanie nigdy wykorzystana i docelowo nie jest wówczas implementowana w celu zaoszczędzenia zasobów układu scalonego. W sytuacji gdy wykorzystany ma zostać cały zbiór WLO, szybko maleje liczba wolnych WLO, a w końcowym etapie wszystkie WLO muszą zostać wybrane. Przykładowe architektury systemów możliwe do uzyskania w oparciu o zaproponowane kryteria wyboru pokazano na rys. 47.



Rys. 47. Architektury systemów z dobieranymi WLO. Linia przerywana oznacza system wykorzystujący podzbiór WLO ze dostępnego zbioru W , linia ciągła oznacza system wykorzystujący cały zbiór W . Szare kropki symbolizują wybrane WLO ze zbioru W . System jednokanałowy oznaczony $1xZWLO$, system wielokanałowy nierekonfigurowalny oznaczony $NxZWLO$, system rekonfigurowalny o zmiennych parametrach oznaczony jest $?xZWLOx$.

Na etapie prac przygotowawczych, przed właściwą implementacją systemu pomiarowego, każda z architektur systemu wymaga wcześniejszej identyfikacji parametrów zbioru dostępnych WLO w celu ustalenia przez algorytm wyboru najkorzystniejszego rozwiązania. Liczba możliwych kombinacji na zbiorze dostępnych WLO może być obliczona przy użyciu powszechnie znanych wzorów z kombinatoryki. Niech W oznacza zbiór wszystkich WLO, które możemy wykorzystać w złożonej wielosegmentowej linii opóźniającej (ZWLO). Uzyskana ZWLO powinna charakteryzować się najmniejszą wartością rozdzielczości ekwiwalentnej. Jeśli w zbiorze W znajduje się n WLO, to możliwa liczba kombinacji L_k złożenia każdej WLO z każdą pozostałą, opisana jest poniższym wzorem:

$$L_k = \frac{n(n-1)}{2} . \quad (16)$$

W wyniku działania algorytmu, w szczególności wskutek działania procedury złożenia, uzyskujemy zbiór wyników wartości rozdzielczości ekwiwalentnej dla par WLO wchodzących w skład ZWLO z informacją, które WLO wchodzą w skład ZWLO.

W przypadku systemu pomiarowego z jedną ZWLO, gdzie ze zbioru W o liczności n można wykorzystać k elementów (WLO) liczba możliwych kombinacji Lk wynosi:

$$Lk = \binom{n}{k} = \frac{n!}{(n-k)!k!} . \quad (17)$$

Wyniki przykładowych obliczeń dla systemu jednokanałowego przedstawiono w tabeli 4.

Tabela 4. Przykładowe liczby kombinacji dla jednej ZWLO, gdzie k -liczba WLO wchodzących w skład ZWLO, a n -liczba WLO w zbiorze W linii dostępnych do użycia

$k \backslash n$	128	64	32	16	8	4
32	$1,47 \cdot 10^{30}$	$1,83 \cdot 10^{18}$	1			
16	$9,33 \cdot 10^{19}$	$4,89 \cdot 10^{14}$	$601,1 \cdot 10^6$	1		
8	$1,43 \cdot 10^{12}$	$4,43 \cdot 10^9$	$10,5 \cdot 10^6$	12 870	1	
4	$10,7 \cdot 10^6$	635 376	35 960	1 820	70	1
2	8 128	2 016	496	120	28	6

Zgodnie z wynikami obliczeń zawartymi w tabeli 4, w przypadku systemu jednokanałowego zbudowanego z 8 WLO, stosunek liczby kombinacji wyboru spośród zbioru 32 i 16 linii opóźniających wynosi w przybliżeniu 800. Natomiast dla zbiorów 64 i 16 linii opóźniających wynosi już około 344 tys. dla pozostałych wariantów systemu proces obliczeń staje się bardziej złożony.

Dla systemu wielokanałowego z rekonfigurowalnymi kanałami pomiarowymi, liczba kombinacji Lk wynosi:

$$Lk(M) = \prod_{j=0}^{\log_2 M - 2} \left(\frac{M}{2^j} \right) \left(\frac{M}{2^{j+1}} \right) 2^{j+1} , \quad (18)$$

gdzie M – oznacza liczbę WLO w zbiorze, które będą wykorzystane w systemie pomiarowym. M jest liczbą całkowitą i jest liczby 2:

$$M = 2^n, n \in \mathbb{N}, n > 2 \quad (19)$$

Podstawiając wzór 17 do 18 otrzymujemy:

$$Lk(M) = \prod_{j=0}^{\log_2 M - 2} \frac{\left(\frac{M}{2^j} \right)!}{\left(\frac{M}{2^j} - \frac{M}{2^{j+1}} \right)! \left(\frac{M}{2^{j+1}} \right)!} 2^{j+1} \quad (20)$$

$$Lk(M) = \prod_{j=0}^{\log_2 M - 2} \frac{\left(\frac{M}{2^j}\right)!}{\left(\frac{M 2^{j+1} - M 2^j}{2^j 2^{j+1}}\right)! \left(\frac{M}{2^{j+1}}\right)!} 2^{j+1} \quad (21)$$

$$Lk(M) = \prod_{j=0}^{\log_2 M - 2} \frac{\left(\frac{M}{2^j}\right)!}{\left(\frac{M 2^j}{2^{2^{j+1}}}\right)! \left(\frac{M}{2^{j+1}}\right)!} 2^{j+1} = \prod_{j=0}^{\log_2 M - 2} \frac{\left(\frac{M}{2^j}\right)!}{\left(\frac{M}{2^{j+1}}\right)! \left(\frac{M}{2^{j+1}}\right)!} 2^{j+1} = \prod_{j=0}^{\log_2 M - 2} \frac{\left(\frac{M}{2^j}\right)!}{\left(\left(\frac{M}{2^{j+1}}\right)!\right)^2} 2^j \quad (22)$$

W zapisie uproszczonym:

$$Lk(M) = \prod_{j=0}^{\log_2 M - 2} \frac{2^{j+1} (2^{-j} M)!}{\left((2^{-j-1} M)!\right)^2} \quad (23)$$

$$Lk = \prod_{j=0}^{n-2} \frac{2^{j+1} (2^{n-j})!}{\left((2^{n-j-1})!\right)^2} \quad (24)$$

W tabeli 5 zamieszczono przykładowe wartości liczby możliwych kombinacji Lk w przypadku systemu rekonfigurowalnego dla zbioru WLO o liczebności M .

Tabela 5. Tabela z przykładowymi liczbami kombinacji dla systemu rekonfigurowalnego

M	Lk
128	$\sim 4,8 \cdot 10^{78}$
64	$\sim 1,9 \cdot 10^{38}$
32	$\sim 3,3 \cdot 10^{18}$
16	$\sim 3,5 \cdot 10^8$

Zgodnie z tabelą 5, w przypadku systemu rekonfigurowalnego stosunek liczby kombinacji dla $M=16$ oraz $M=32$ wynosi około 9,5 mld.

Wraz ze wzrostem stopnia złożoności systemu obserwuje się znaczne wydłużenie czasu realizacji algorytmu. Z tego względu założono, że czas znalezienia optymalnych WLO wchodzących w skład systemu pomiarowego powinien być akceptowalnie długi, a mianowicie na poziomie czasu kompilacji i implementacji systemu w układzie FPGA. Sprawdzenie wszystkich kombinacji WLO dalej nazywane metodą bezpośrednią, jest najlepszą metodą, aby uzyskać najkorzystniejszy z możliwych wyników. Jednakże metoda ta jest czasochłonna. Jeśli przyjąć możliwość sprawdzenia tysięcy kombinacji na minutę, to dla przeciętnie małego zbioru możliwych WLO do wykorzystania w systemie, czas obliczeń może sięgać rzędu dni a nawet tygodni (tabele 4 i 5). Dlatego w pracy przeanalizowano i opracowano metody wyszukiwania, które pozwalają znacznie skrócić czas doboru kombinacji WLO, osiągając możliwie najkorzystniejsze wyniki w kontekście przyjętych kryteriów.

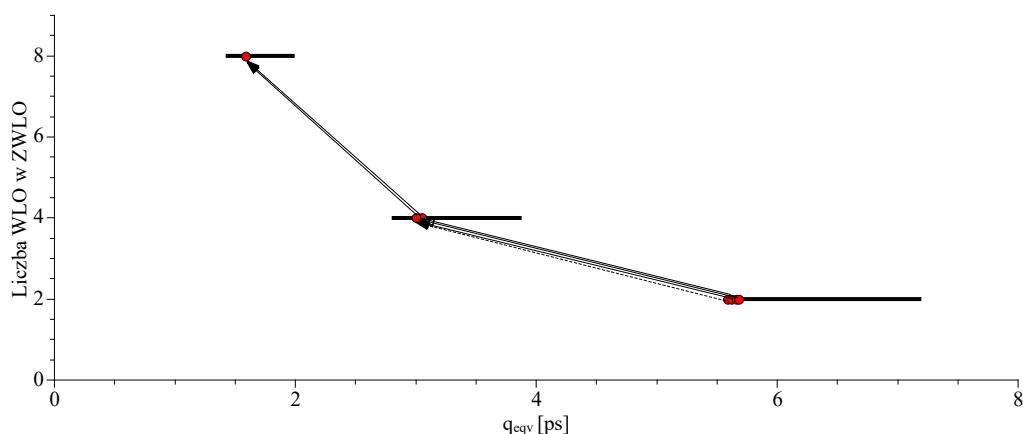
5.1.1 System z jednym kanałem pomiarowym

Priorytetem w budowie systemu pomiarowego jest osiągnięcie optymalnych parametrów metrologicznych. W tym celu, należy uwzględnić różne czynniki, takie jak stopień zaangażowania zasobów układu programowalnego i dostępność wolnych zasobów, stopień skomplikowania dekodery, możliwości połączeń wewnątrz układu, łatwość akwizycji danych oraz intensywność mierzonych impulsów. Dodatkowo, konieczne jest uwzględnienie aspektów energetycznych, takich jak pobór energii (zwłaszcza w przypadku rozwiązań energooszczędnych) lub emisja ciepła (w celu ochrony przed przegrzaniem).

Na początku etapu projektowania określone są wstępnie parametry metrologiczne systemu oraz zasoby niezbędne do implementacji systemu. Na tej podstawie określana jest liczba WLO koniecznych do użycia w docelowym systemie pomiarowym. Wówczas WLO są implementowane w różnych miejscach w układzie FPGA, dla których określa się ich charakterystyki i przesunięcia czasowe pomiędzy poszczególnymi WLO.

Jak pokazano w tabeli 4, liczba kombinacji rośnie w tempie geometrycznym w zależności od liczby WLO wchodzących w skład zbioru W i jest największa gdy docelowy system pomiarowy będzie wykorzystywał połowę WLO ze zbioru W do zbudowania jednej ZWLO. Jeśli w zbiorze W występuje tylko 16 WLO, obliczenia mające na celu sprawdzenie wszystkich możliwych kombinacji zajmą niewielką ilość czasu obliczeniowego. Dla współczesnych komputerów czas ten nie przekracza jednej minuty.

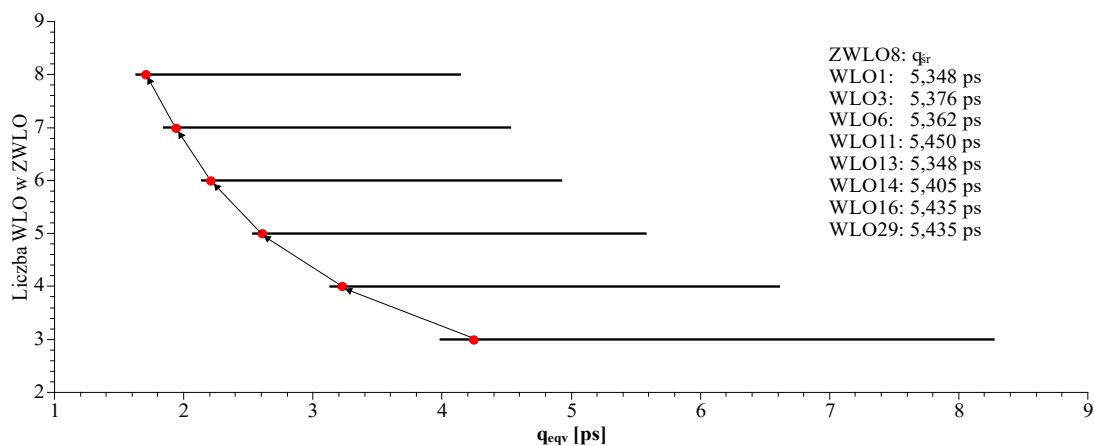
Podczas rozwoju metody optymalizującej proces wyszukiwania, zauważono interesującą właściwość. Wybranie dwóch WLO o najmniejszej wartości rozdzielczości ekwiwalentnej z całego zbioru W , wcale nie zapewnia w wyniku złożenia najlepszych parametrów, czyli najmniejszej wartości rozdzielczości ekwiwalentnej dla ZWLO (rys. 48). Podobnie jeśli wybrana zostanie najlepsza para dwóch ZWLO, to dobieranie do tej pary drugiej najlepszej pary ZWLO również nie zawsze daje najkorzystniejszy wynik.



Rys. 48. Wynik wyboru par najlepszych czyli charakteryzującymi się najmniejszymi wartościami rozdzielczości ekwiwalentnej q_{eqv} . Poziomymi kreskami zaznaczono przedziały wartości rozdzielczości q_{eqv} , a czerwone punkty oznaczają wartości q_{eqv} dla odpowiedniej liczby WLO wchodzących w skład ZWLO

Jak wspomniano wcześniej, korzystniejszą sytuacją jest, gdy charakterystyki dwóch WLO dopełniały się, niż osobno posiadały najlepsze parametry (rys. 46). W związku z tym zaproponowano roz-

wiązanie tego problemu w następującej procedurze. Na początku obliczana jest rozdzielczość q_{eqv} wszystkich kombinacji par WLO $\binom{n}{k}$ dla $k=2$. Kombinacji jest stosunkowo niewiele i na przykład dla zbioru WLO o liczności $n=32$ jest ich 496. Z tej grupy wybierana jest kombinacja par najlepiej spełniająca kryteria. Następnie określana jest maksymalna liczba WLO, które mają wchodzić w skład ZWLO (np. połowę zbioru W). Do wybranej pary wchodzącej w skład ZWLO dodawane są poszczególne WLO spośród jeszcze nieużytych i obliczane są parametry tak złożonej trójki ZWLO3, a ich kombinacja oraz i wartość q_{eqv} są zapamiętywane. Ze zbioru wszystkich możliwych ZWLO3 wybiera się ZWLO3 spełniające przyjęte kryterium (najmniejsza wartość q_{eqv}). Po czym ponownie do ZWLO3 dobierana jest kolejna nieużyta dotychczas WLO, obliczane są parametry ZWLO4, a kombinacja WLO oraz wartość q_{eqv} są zapamiętywane. Dalej postępowanie jest analogiczne. Czynność powtarzana jest aż do osiągnięcia zadanej liczby WLO w ZWLO. Przykładowy wynik działania tej metody jest pokazany na rys. 49.



Rys. 49. Wynik działania metody dobierania WLO do pary o najkorzystniejszych parametrach dla pojedynczego kanału pomiarowego

W przypadku konieczności wykorzystania wszystkich linii ze zbioru W należy dokonać bezpośredniego złożenia wszystkich WLO w jedną ZWLO. W rezultacie otrzymuje się ZWLO o najlepszej możliwej rozdzielczości ekwiwalentnej dla rozważanego zbioru W . Obliczenia tego typu mogą zostać wykonane na samym początku procesu projektowania w celu określenia granicznych parametrów rozdzielczości ekwiwalentnej.

5.1.2 System wielokanałowy

Rozważmy przypadek, gdy ze zbioru W WLO należy utworzyć system pomiarowy o nieziennej konfiguracji. Podejście do rozwiązania problemu jest podobne do opisywanego poprzednio. Jednak w tym przypadku należy wziąć pod uwagę, że tworzone kanały pomiarowe powinny mieć, w miarę możliwości, zbliżone parametry metrologiczne. Jeśli zdefiniowanym kryterium doboru linii jest rozdzielczość ekwiwalentna, to funkcja celu metody dobierania WLO określona jest wzorem poniżej:

$$\Delta q_{eqv,x} = \max(Q_{eqv}(ZWLO_x)) - \min(Q_{eqv}(ZWLO_x)) < \epsilon_x, \quad (25)$$

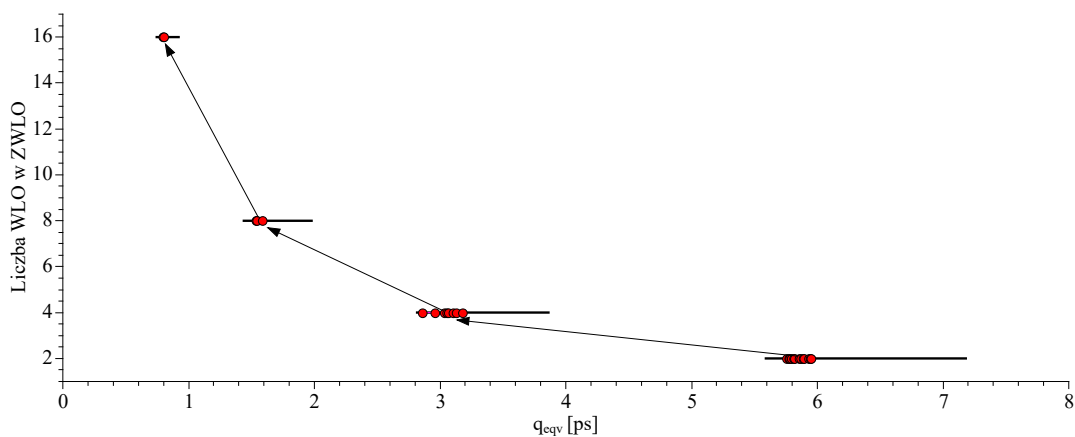
gdzie:

x – liczba WLO wchodzących w skład każdej ZWLO,

Q_{eqv} – zbiór obliczonych wartości q_{eqv} dla zbioru ZWLO $_x$,

ε_x – jest najmniejszą znaną różnicą w zbiorze obliczonych wartości q_{eqv} dla całego zbioru ZWLO $_x$. Wartość $\varepsilon_x > 0$, jest wstępnie określana i jest też na bieżąco aktualizowana, w celu umożliwienia algorytmowi dokonania optymalnego wyboru.

W tym przypadku należy dążyć do uzyskania jednocześnie możliwie małej wartości rozdzielczości ekwiwalentnej ZWLO poszczególnych kanałów i małej wartości ε_x pomiędzy tymi kanałami (rys. 50).



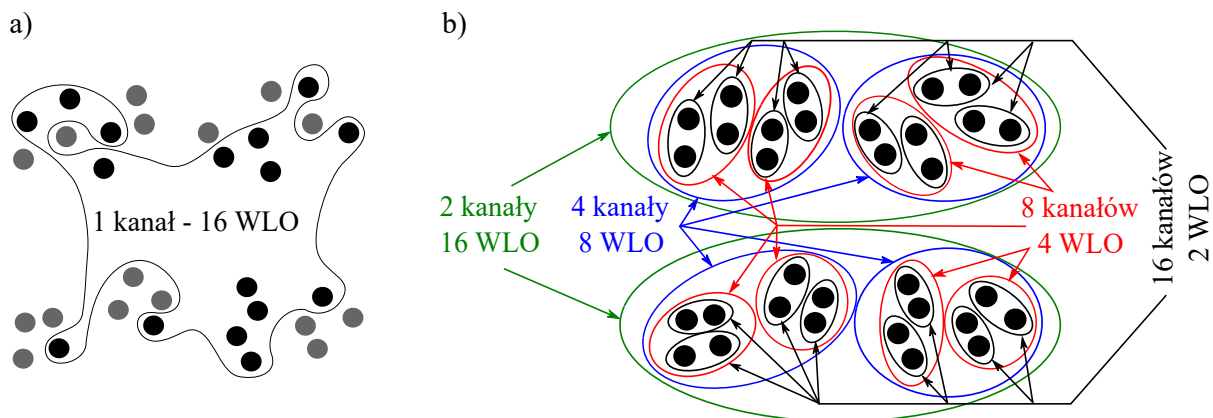
Rys. 50. Wynik realizacji algorytmu w trybie równowagi wszystkich kanałów. Czerwone punkty oznaczają wartości rozdzielczości q_{eqv} dla wybranych ZWLO, które są kompromisem pomiędzy najmniejszymi wartościami q_{eqv} a rozrzutem q_{eqv} dla wszystkich kanałów

Trudność w tym wariancie dla metody bezpośredniej wynika z konieczności przechowywania tablicy wyników zawierającej najmniejsze wartości rozdzielczości ekwiwalentnej dla ZWLO oraz odpowiednio dane o kombinacjach WLO wchodzących w skład tej ZWLO. Priorytetem dla tej metody jest równomierność znalezionych wartości q_{eqv} dla ZWLO. Najkorzystniejszym rozwiązaniem byłoby przechowywanie wszystkich kombinacji WLO oraz wyników q_{eqv} dla tych kombinacji. Dla systemu w konfiguracji 2 kanałów, po 2 WLO w każdym kanale, ten warunek może być z łatwością spełniony, a obliczenia trwają krótko $< 0,1$ sek. Jednak w miarę wzrostu liczby wyników, zgodnie z permutacją (wzór 21), niemożliwe jest przechowywanie wszystkich wyników w pamięci komputera. Dodatkowo część pamięci komputera zarezerwowana jest dla procesu iteracyjnego generującego kombinacje WLO bez powtórzeń. Problem ten został rozwiązany poprzez utworzenie tablicy wyników q_{eqv} i kombinacji WLO o stałym ustalonym wymiarze. Do tablicy tej na bieżąco dopisywane są nowe wyniki, a zawartość tablicy jest przeszukiwana i usuwane są wpisy z największymi wartościami q_{eqv} . Jednak operacja ta spowalnia obliczenia. Dlatego, aby nie przeszukiwać tablicy z każdym nowym wynikiem, robione jest to co pewną liczbę iteracji, zależną od liczby kanałów i liczby WLO wchodzących w skład ZWLO.

Przy wykorzystaniu metody doboru do pary lub par, algorytm zaczyna od sprawdzenia kombinacji wszystkich par WLO ze sobą. Może się zdarzyć, że w grupie najlepszych par wystąpi ta sama WLO kilka razy. Dlatego kryterium wyboru w tym przypadku musi zostać zmienione, tak aby każda WLO ze zbioru W występowała tylko jeden raz w danym kanale pomiarowym w całym docelowym systemie. Dalej metoda postępowania jest analogiczna jak przy pojedynczym kanale. Należy zauważyć, że jeśli w systemie jest N kanałów pomiarowych, liczba wolnych do wykorzystania WLO znacznie szybciej maleje po każdej pętli algorytmu.

5.1.3 System z rekonfigurowalnymi kanałami pomiarowymi

W przypadku systemu z rekonfigurowalnymi kanałami pomiarowymi o zmiennych parametrach, gdzie system może mieć różną liczbę WLO wchodzących w skład kanału pomiarowego założono, że WLO użyta w danym kanale nie zmienia już przynależności pomiędzy kanałami pomiarowymi w czasie zmiany liczby kanałów pomiarowych (rys. 51).

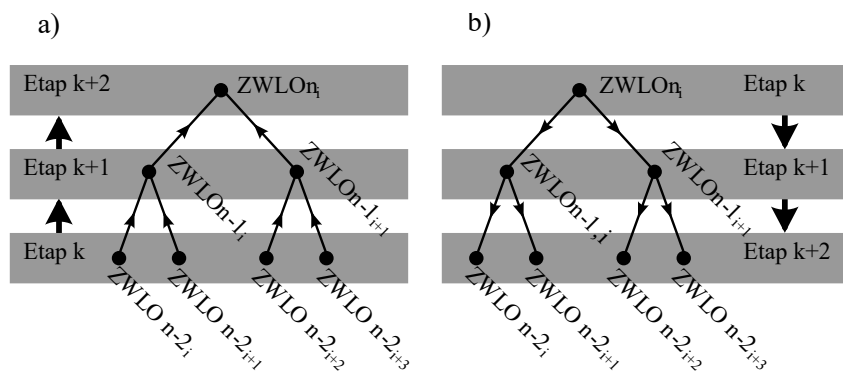


Rys. 51. Różnica w algorytmie doboru WLO pomiędzy systemem z 1 kanałem pomiarowym, a systemem rekonfigurowalnym. Praktycznie dowolny dobór WLO dla jednego kanału (a) oraz dobór WLO w rekonfigurowalnym systemie wielokanałowym (b). Czarne punkty oznaczają wybrane WLO wchodzące w skład ZWLO w systemie. W pętlach zaznaczono WLO wchodzące w skład ZWLO. Kolorem czarnym ZWLO2 dla systemu 16 kanałowego, czerwonym ZWLO4 dla systemu 8 kanałowego, niebieskim ZWLO8 dla systemu 4 kanałowego i zielonym dla ZWLO16 dla systemu dwukanałowego.

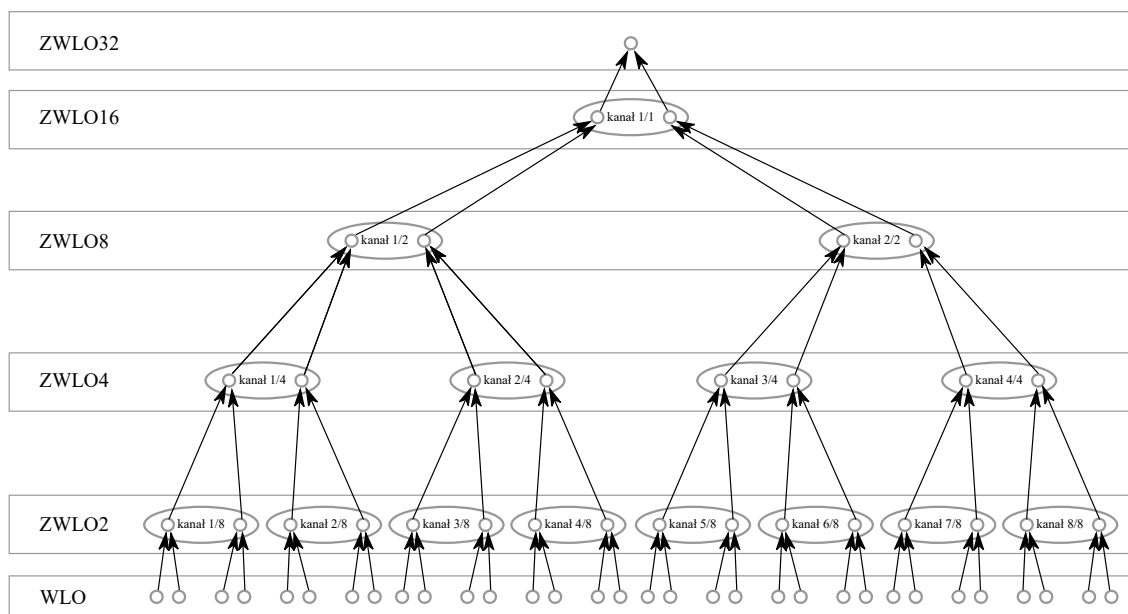
Dla przykładu rozpatrzmy system, posiadający możliwości konfiguracji, umożliwiające pomiar za pomocą 16-8-4-2-1 kanałów, zbudowanych z 32 WLO ze zbioru W . Jeśli system skonfigurowany jest do pomiaru za pomocą jednego kanału ($N = 1$), wszystkie WLO będą użyte w jednej ZWLO32 do budowy tego kanału. Natomiast jeśli system będzie dokonywał pomiaru 16 kanałami ($N=16$), składać się one będą z 16 ZWLO2. Podobnie jak w systemie wielokanałowym nierekonfigurowalnym, należy zadbać, aby rozdzielczości ekwiwalentne kanałów pomiarowych miały jak najbardziej zbliżone wartości (wzór 25) oraz aby WLO były tak dobrane, by przy rekonfiguracji (dla $N = 8, 4, 2$) uzyskiwać również możliwie zbliżone parametry $\Delta q_{eqv,2} < \epsilon_2$, $\Delta q_{eqv,4} < \epsilon_4$, $\Delta q_{eqv,8} < \epsilon_8$. Samo wyszukiwanie właściwych kombinacji WLO w systemie rekonfigurowalnym może następować na trzy sposoby, a mianowicie:

- a) w górę (korony drzewa) (rys. 52a),

- b) w dół (pnia drzewa) (rys. 52b),
- c) całościowo (kompletne drzewo) (rys. 53).



Rys. 52. Kierunki wyszukiwania w systemie rekonfigurowalnym dla kierunku algorytmu a) w górę (korony drzewa), b) w dół (pnia drzewa)



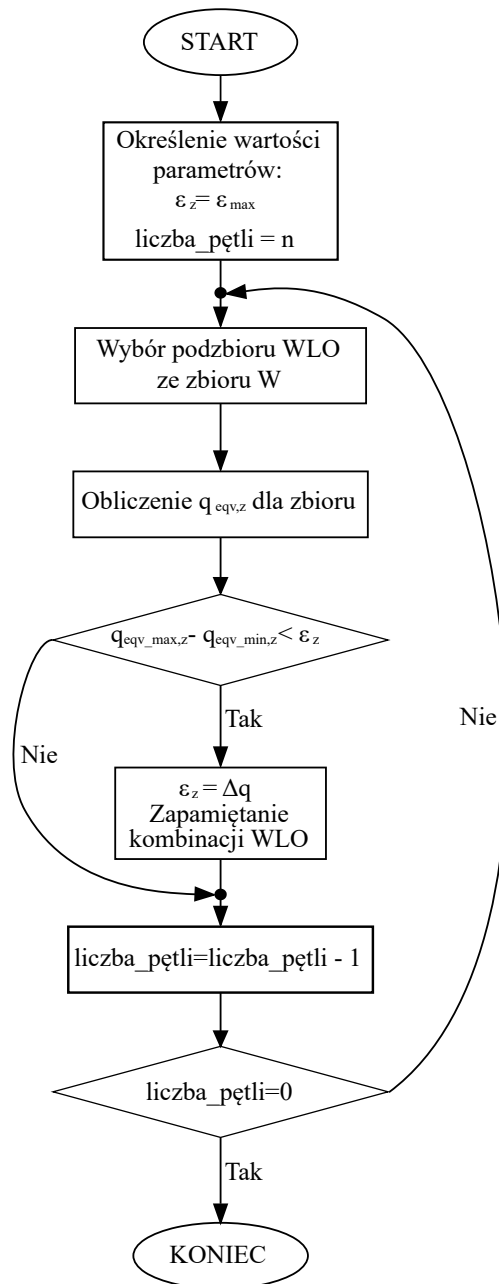
Rys. 53. Diagram ilustrujący zasadę przynależności WLO do kanałów w algorytmie wyszukiwania kompletnego drzewa

Etapy na rys. 52. oznaczają wyszukanie kombinacji spełniających warunek (25), a zwrot strzałki oznacza kierunek w jakim algorytm przechodzi do kolejnych etapów.

Ze względu na stosunkowo dużą wartość czasu obliczeń, wynikającą z dużej liczby kombinacji do przeanalizowania, zastosowano metodę Monte Carlo. Metoda ta została opracowana w latach 40 XX wieku [86] w trakcie rozwoju broni jądrowej. Służyła wówczas do obliczeń oddziaływania wysokoenergetycznych cząstek z jądrami atomów. Współcześnie wykorzystuje się ją w symulacjach fizycznych, matematycznych i ekonomicznych. Nie jest to jedna konkretna metoda lecz raczej klasa metod. Istotą metody Monte Carlo jest przeprowadzenie obliczeń na losowych danych wejściowych. Metoda ta została zaadoptowana na potrzeby algorytmu wyszukiwania WLO.

Wyszukiwanie w kierunku korony drzewa

Algorytm, którego strukturę pokazano na rys. 54, realizowany jest od dołu do góry, czyli w górę korony drzewa.



Rys. 54. Diagram przepływu dla algorytmu wyboru w górę korony drzewa

Działanie algorytmu można opisać w następujący sposób. Ze zbioru W wybierane są w sposób losowy niepowtarzalne pary WLO, z których konstruuje się ZWLO2 w ustalonej liczbie zależnej od przyjętej liczby kanałów. Po złożeniu WLO w ZWLO2 obliczane są wartości rozdzielczości ekwiwalentnych par ZWLO2 i jeśli w tak powstałym zbiorze $\Delta q_{\text{eqv},2} < \varepsilon_2$, algorytm przechodzi do następnego etapu. Z dwóch par ZWLO2 tworzone są następnie ZWLO4 i sprawdzane czy rozdzielczości ekwiwalentne tak powstałych czwórek ZWLO4 w tym zbiorze są zbliżone. Jeśli nie jest spełniony warunek, $\Delta q_{\text{eqv},4} < \varepsilon_4$, algorytm wycofuje się do poprzedniego etapu, w którym są na nowo wybierane

pary ZWLO2. Wartość parametru ε_z na początku algorytmu wyboru ma określoną wartość ε_{max} , która obliczana jest z pierwszego złożenia i obliczenia $\Delta q_{eqv,z}$. Wartość ε_z aktualizowana jest w momencie, gdy $\Delta q_{eqv,z} < \varepsilon_z$.

Wyszukiwanie w kierunku pnia drzewa

W kolejnym analizowanym przypadku algorytm realizowany jest od góry do dołu, czyli w dół drzewa, co zilustrowano na (rys. 52 b). Z dostępnego zbioru W wyodrębniane są w dwa równoliczne podzbiory WLO i obliczana jest rozdzielczość ekwiwalentną dla obu grup. Jeśli $\Delta q_{eqv,z} < \varepsilon_z$, to algorytm przechodzi do drugiego etapu. Podzbiory znów dzielimy na pół, tworząc cztery podzbiory i ponownie sprawdzana jest wartość $\Delta q_{eqv,z}$. Jeśli w tym przypadku $\Delta q_{eqv,z} < \varepsilon_z$, algorytm przechodzi analogicznie do następnego etapu. W przeciwnym wypadku podgrupy wybierane są na nowo. Wybór odbywa się w sposób losowy. Jeśli algorytm natrafi na problem polegający na braku możliwości dobrania WLO, tak aby dla wszystkich kanałów osiągnąć zbliżone parametry, potrafi wycofać się o jeden etap, wylosować na nowo podgrupy i powtórnie rozpocząć wyszukiwanie kombinacji WLO, poruszając się w dół.

Wyszukiwanie kompletnego drzewa

Algorytmu wyszukiwania kompletnego drzewa polega na tym, że od razu tworzone jest całe drzewo przynależności WLO do kanałów. Znaczenie mają zarówno pary, jak i ich umiejscowienie w strukturze drzewa (rys. 53). Wybór miejsca w strukturze drzewa odbywa się metodą losową.

Tabela 6. Porównanie wyników działania algorytmu wyszukiwania dla przykładowego systemu rekonfigurowalnego z 32 WLO. Czasy wykonania 10^5 iteracji.

		Liczba kanałów pomiarowych/liczba WLO w kanale				
wyszukiwanie		2/16	4/8	8/4	16/2	Czas [s]
całościowe (kompletnego drzewa)	$q_{eqv\ \acute{s}r}$ [ps]	0,987	1,983	3,901	7,530	175
	$\Delta q_{eqv\ \acute{s}r}$ [ps]	0,021	0,120	0,357	0,836	
w dół (pnia drzewa)	$q_{eqv\ \acute{s}r}$ [ps]	0,977	1,957	3,776	7,034	254
	$\Delta q_{eqv\ \acute{s}r}$ [ps]	0,012	0,126	0,447	0,712	
w górę (korony drzewa)	$q_{eqv\ \acute{s}r}$ [ps]	1,011	1,930	3,810	7,214	280
	$\Delta q_{eqv\ \acute{s}r}$ [ps]	0,030	0,090	0,196	0,538	

W tak wybranym drzewie przynależności sprawdzane ZWLO_z na danym poziomie drzewa (dla odpowiedniego wariantu liczby kanałów i wchodzących w ich skład liczby ZWLO) i jeśli $\Delta q_{eqv,z} < \varepsilon_z$, sprawdzany jest poziom w strukturze drzewa wyżej. Jeśli rozdzielczości ekwiwalentne kanałów mają zbyt duże rozbieżności $\Delta q_{eqv,z} > \varepsilon_z$ wówczas obliczenia są przerywane i całe drzewo konstruowane jest od nowa. Właściwy wybór kierunku wyszukiwania wpływa na lepsze parametry (średnia wartość rozdzielczości ekwiwalentnej kanałów i różnica rozdzielczości ekwiwalentnej kanałów) dla etapu, od którego zaczyna się proces wyszukiwania (tabela 6).

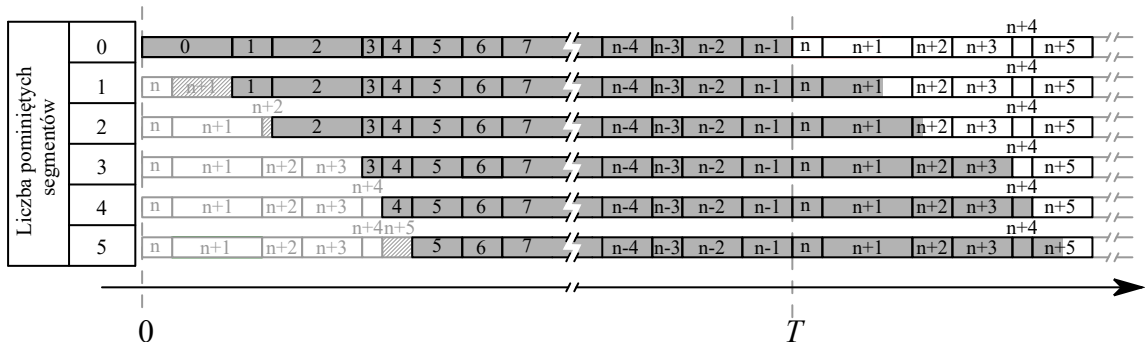
Wartości q_{eqv} oraz różnice Δq_{eqv} będą najmniejsze od poziomu drzewa, z którego zaczynają się obliczenia (lub całości dla algorytmu wyszukiwania kompletnego drzewa), ponieważ algorytm nie jest ograniczony do wcześniej wybranych kombinacji WLO.

W przypadku wyszukiwania w górę, algorytm przechodzi do realizacji kolejnego etapu dopiero, jeśli $q_{eqv,2}$ dla wszystkich ZWLO2 będzie się mieścić w ustalonych granicach ($\Delta q_{eqv,2} < \varepsilon_2$), co skutkuje mniejszymi rozrzutami q_{eqv} dla tych ZWLO2. Na dalszych etapach algorytm jest ograniczony możliwością wyboru jedynie do tych kombinacji ZWLO2. Odwrotnie jest w przypadku wyszukiwania w dół, gdzie dla zbioru 32 WLO algorytm wyszukuje dwa podzbiory ZWLO16. W tym przypadku $\Delta q_{eqv,16}$ dla ZWLO16 będą mieć mniejsze wartości niż w przypadku wyszukiwania w górę, ponieważ algorytm nie jest ograniczony wcześniej wybranymi kombinacjami WLO wchodzącymi w skład ZWLO16. Wyszukiwanie całościowe zachowuje balans pomiędzy dwiema poprzednimi metodami bez faworyzowania najniższego lub najwyższego poziomu. Zauważalna jest w tym przypadku różnica w wydajności, gdyż metoda wyszukiwania całościowego losuje tylko jeden raz całe drzewo, a sprawdzając wartość $\Delta q_{eqv,x}$ może na wczesnym etapie odrzucić taką kombinację drzewa, w której $\Delta q_{eqv,x}$ przekracza zadaną wartość ε_x .

W zależności od wariantu metody i początkowego punktu wyszukiwania, osiągnane są mniejsze różnice Δq_{eqv} oraz wartości q_{eqv} dla danego poziomu ZWLO. Różnica w czasie obliczeń między metodami "w górę" i "w dół" wynika z liczby przeprowadzanych obliczeń. Rozpoczynając od dolnego poziomu konieczne jest sprawdzenie wszystkich 16 ZWLO2, co prowadzi do większej liczby wywołań procedury "złożenia" wielu WLO w ZWLO oraz sprawdzenia, czy dana kombinacja spełnia warunek $\Delta q_{eqv,x} < \varepsilon_x$. Pomimo to, algorytm osiąga lepsze wyniki dla dolnych ZWLO2. Wariant „całościowy” wykonuje się w krótszym czasie, ponieważ po wylosowaniu całego drzewa wartości są sprawdzane dla 16 ZWLO2, a jeśli $\Delta q_{eqv,2} > \varepsilon_2$, to całe drzewo jest ponownie losowane bez próby wybrania nowej kombinacji 16 ZWLO2. Dopiero jeśli wartość Δq_{eqv} mieści się w ustalonym zakresie, są wykonywane obliczenia dla kolejnego poziomu. Należy podkreślić, że jest to metoda losowa, dlatego wyniki mogą się różnić od siebie przy wielokrotnym uruchamianiu algorytmu.

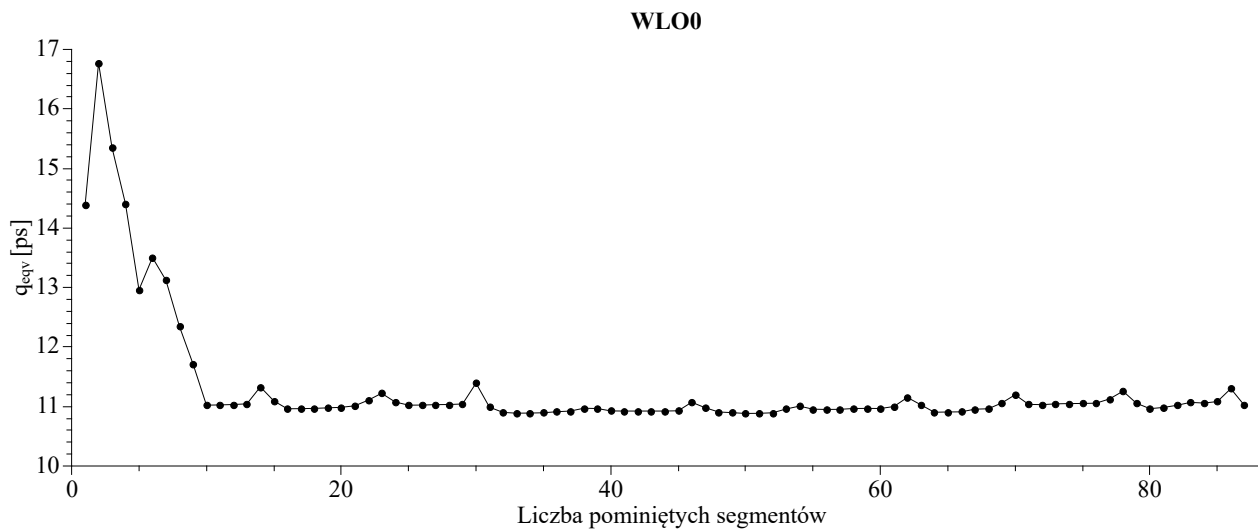
5.2 Modyfikacja charakterystyki wielosegmentowej linii opóźniającej

Stopień skomplikowania obliczeń podczas realizowania algorytmu doboru rośnie w zależności zarówno od skomplikowania docelowego systemu pomiarowego, jak i dostępnej liczby WLO w zbiorze W . W przypadku użytego układu FPGA Kintex UltraScale XCKU040 liczba dostępnych miejsc do zaimplementowania WLO wynosi w przybliżeniu 500. Jeśli jednak uwzględnić to, że implementowana jest cała WLO o długości 480 segmentów, a przebieg zegara o częstotliwości 500 MHz (2 ns) obejmuje nie więcej niż 400 segmentów ($2 \text{ ns}/\tau$), można modyfikować same WLO, poprzez ignorowanie odczytów z początkowych segmentów linii. W efekcie otrzymuje się zmodyfikowaną WLO, która ma również zmienioną charakterystykę przetwarzania. W tak zmodyfikowanej WLO większość przedziałów kwantowania będzie pokrywać się z WLO sprzed modyfikacji, gdyż nie zmieniają się przesunięcia czasowe, ale powstaną nowe kanały (na końcu WLO), które mogą zmienić (poprawić lub pogorszyć) charakterystykę WLO. Zasadę modyfikacji charakterystyki przedstawiono na rys. 55.



Rys. 55. Idea modyfikacji charakterystyki WLO poprzez pomijanie segmentów początkowych. Koloriem ciemno szarym zaznaczono pokrycie całego okresu zegara wzorcowego dla przetwornika jednostopniowego z sygnałem zegarowym doprowadzonym do WLO

Zmiany rozdzielczości ekwiwalentnej dla wybranej WLO w sytuacji pominięcia początkowych od 1 do 87 segmentów przedstawiono na rys. 56.

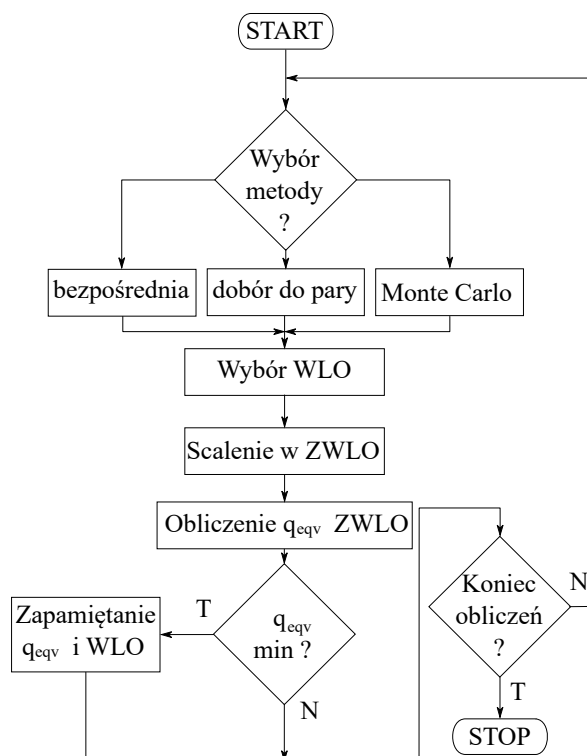


Rys. 56. Przykład zmiany rozdzielczości ekwiwalentnej q_{eqv} w przypadku pominięcia początkowych przedziałów kwantowania dla wybranej WLO

Widoczna jest istotna zmiana wartości q_{eqv} szczególnie, jeśli początkowe przedziały kwantowania WLO, mają wartości znacznie większe od wartości średniej $\bar{\tau}$ dla danej WLO. Pominięcie kanału o dużej wartości τ_i powoduje również, że na końcu pojawia się kilka przedziałów kwantowania, których suma czasów trwania musi pokrywać się z wartością τ_i , co wpływa na zmniejszenie wartości rozdzielczości ekwiwalentnej. W wyniku pominięcia 10 segmentów WLO z rys. 56, wartość rozdzielczości ekwiwalentnej uległa zmianie z 16,8 ps na 11 ps, czyli poprawa wyniosła około 35%.

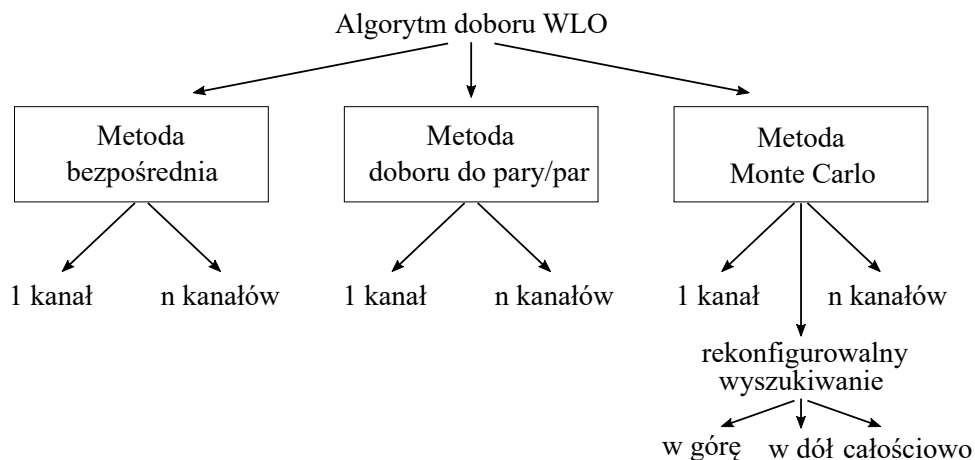
5.2 Implementacja algorytmu

W oparciu o zaproponowane w rozdziale 5.1 kryteria konfiguracji systemu pomiarowego opracowano algorytm wyboru WLO (rys. 57).



Rys. 57. Schemat blokowy Algorytmu wyboru WLO

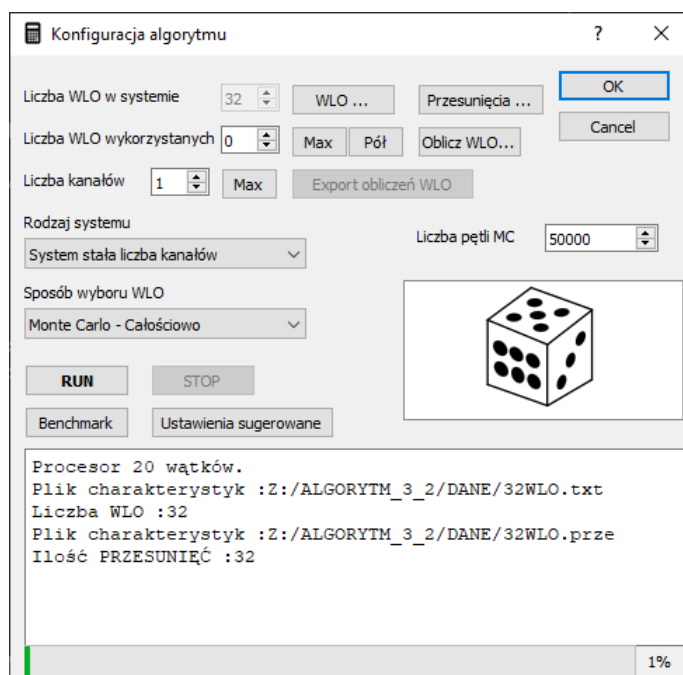
Umożliwia on wybór jednej z trzech dostępnych metod wyboru: bezpośredniej, doboru do pary i Monte Carlo (rys. 58). Wynikiem działania algorytmu są informacje o parametrach metrologicznych wybranego wariantu systemu wraz ze strukturą doboru WLO określoną na bazie wczytanych przez użytkownika charakterystyk przetwarzania.



Rys. 58. Schemat algorytmu doboru WLO z podziałem na metody i ich warianty

Po uruchomieniu programu użytkownik wybiera pliki z charakterystykami wcześniej zaimplementowanych WLO i przesunięciami czasowymi pomiędzy WLO. Następnie określa parametry systemu a mianowicie liczbę użytych WLO w docelowym systemie, liczbę kanałów, sposób wyszukiwania, konfigurację systemu pomiarowego. Na tym etapie może uruchomić wyszukiwanie kombinacji WLO o najmniejszej rozdzielczości ekwiwalentnej lub sprawdzić orientacyjny czas wyszukiwania dla metody bezpośredniej. W metodzie bezpośredniej algorytm wyszukuje wszystkie możliwe kombinacje WLO, odpowiednio dokonuje złożenia w ZWLO, a następnie oblicza wartość rozdzielczości ekwiwalentnej q_{eqv} . W kolejnym kroku zapamiętywana jest wartość minimalna q_{eqv} łącznie z bieżącą kombinacją WLO. W metodzie losowej, algorytm losuje kombinacje ze zbioru WLO i analogicznie do metody bezpośredniej wyszukuje minimalną wartość q_{eqv} . Dla metody doboru do pary, algorytm postępuje inaczej. Na samym początku obliczane są wszystkie kombinacje par ZWLO2 i w zależności od liczby kanałów i wybierana jest, liczba par o najmniejszej wartości rozdzielczości ekwiwalentnej q_{eqv} . W kolejnym kroku algorytm dobiera do wybranych par pozostałe wolne WLO. Dalsze obliczenia są analogiczne jak dla dwóch wyżej wymienionych metod.

Algorytm wyboru WLO został zaimplementowany z użyciem języka **Python** w wersji 3.x [87, 88]. W celu przyspieszenia obliczeń została użyta biblioteka obliczeń numerycznych **numpy**. W programie zastosowano także bibliotekę **itertools**, której celem jest generowanie permutacji z numerami WLO ze zbioru W . Dodatkowo, w celu utworzenia interfejsu graficznego (rys. 59) wykorzystano bibliotekę **PyQt**.



Rys. 59. Widok interfejsu użytkownika programu z zaimplementowanym algorytmem doboru WLO

Program przed rozpoczęciem wyszukiwania właściwych WLO adekwatnie do parametrów docelowego systemu, umożliwia wykonanie testu (benchmarku), obliczającego orientacyjny czas wyszukiwania właściwych kombinacji WLO wchodzących w skład ZWLO (systemu pomiarowego), a także

sugeruje odpowiednie wartości opcji wyszukiwania (ze względu na czas obliczeń i wymagania dostępnej pamięci RAM komputera) (tabela 7).

Tabela 7. Sugerowane przez algorytm metody wyszukiwania w zależności od wymaganej konfiguracji systemu i czasu obliczeń, T – wartość czasu obliczeń wyznaczona z benchmarku, T_0 – czas ustalony przez użytkownika, np. 1 godzina

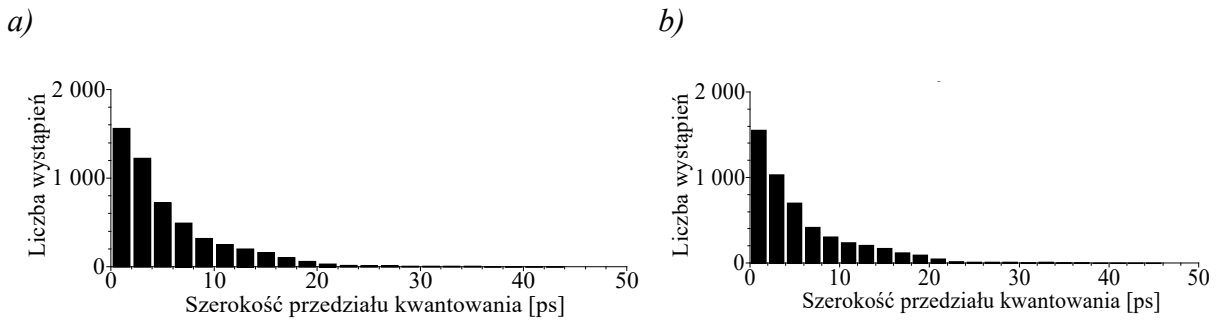
Konfiguracja sytemu	Warunki dodatkowe		Sugerowana metoda algorytmu
	Dostępna wymagana pamięć RAM	Czas obliczeń	
Jeden kanał	TAK	$T < T_0$	Bezpośrednia
	NIE	$T \approx T_0$	Dobór do pary
	NIE	$T \gg T_0$	Monte Carlo
Wiele kanałów	TAK	$T < T_0$	Bezpośrednia
	NIE	$T \approx T_0$	Dobór do par
	NIE	$T \gg T_0$	Monte Carlo
System rekonfigurowalny	–	–	Monte Carlo całościowo

Głównymi zadaniami obliczeniowymi algorytmu wyboru WLO pokazanego na rys. 57 są:

- wybranie odpowiedniego podzbioru WLO do utworzenia ZWLO,
- wykonanie operacji scalenia WLO, podczas której na podstawie wybranych WLO utworzona zostaje docelowa ZWLO,
- obliczenie rozdzielczości ekwiwalentnej nowo powstałej ZWLO.

W celu przyspieszenia operacji obliczeniowych dane wejściowe oraz pośrednie struktury danych w czasie obliczeń są reprezentowane zmiennymi typu całkowitego, co również zabezpiecza przed błędami zaokrągleń. Dodatkowo taki typ danych umożliwia wstępne przygotowanie tablic wartości potęg sześciąt dla wszystkich możliwych wystąpień wartości liczbowych szerokości przedziałów kwantowania, co znacznie zwiększa szybkość obliczenia wartości rozdzielczości ekwiwalentnej. Na końcu procesu obliczeń wartość rozdzielczości ekwiwalentnej zamieniana jest na format zmiennoprzecinkowy.

W celu istotnego skrócenia czasu realizacji algorytmu wyboru WLO, na podstawie wcześniej zebranych charakterystyk WLO, opracowano symulator, który wytwarza w znacznie krótszym czasie WLO o charakterystykach zbliżonych do tych jakie można zmierzyć w rzeczywistym układzie FPGA (rys. 60).



Rys. 60. Histogramy szerokości kanałów czasowych (tożsamy z przedziałami kwantowania) WLO rzeczywistych (a) oraz WLO symulowanych (b).

Na podstawie zmierzonych rzeczywistych charakterystyk WLO, zaimplementowanych w układzie Kintex UltraScale z użyciem komórek łańcucha szybkich przeniesień arytmetycznych, określono rozkład statystyczny szerokości kanałów. Na podstawie tego rozkładu wylosowano szerokości przedziałów kwantowania i dokonano przypadkowej (o rozkładzie równomiernym) zamiany ich kolejności, aby zachować charakterystyczny dla rzeczywistych WLO przeplot przedziałów o wąskich i szerokich przedziałach czasowych. Otrzymano w ten sposób szereg zbiorów W , których użyto następnie do opracowania i weryfikacji działania algorytmu wyboru WLO. Wykreślone histogramy przedziałów kwantowania dla WLO rzeczywistych oraz symulowanych przedstawiono na rys. 60. Duża zgodność uzyskanych wyników pozwoliła na znaczne skrócenie procesu opracowania algorytmu doboru WLO. Dalsza weryfikacja algorytmu została przeprowadzona w oparciu o otrzymane wyniki pomiarów z użyciem rzeczywistych danych pomiarowych z przetwornika zaimplementowanego w układzie FPGA Kintex UltraScale.

5.3 Wyniki badań testowych

W pomiarach naukowych, na przykład na potrzeby fizyki wysokich energii, zachodzi potrzeba zastosowania układu pomiarowego zbudowanego w oparciu o układ FPGA do wymagań eksperymentu. Docelowo w takich systemach pomiarowych najważniejsze są parametry metrologiczne, jednak w początkowym etapie należy również uwzględnić czas projektowy. Dlatego ważne jest określenie wydajności algorytmu wyboru linii w zależności od wybranej metody doboru i parametrów zadanych przez projektanta systemu pomiarowego.

W celu zweryfikowania skuteczności działania opracowanych algorytmów obliczenia zostały przeprowadzone na komputerach o parametrach wy specyfikowanych w tabeli 8. Do badań użyto komputerów o konfiguracjach różniących się ilością pamięci RAM oraz wydajnością procesorów. Komputery nie posiadają kart graficznych, które mogłyby posłużyć do obliczeń. Testowanie algorytmu w odmiennych warunkach, z wykorzystaniem komputerów o zróżnicowanych zasobach, umożliwiło ocenę jego skuteczności i wydajności. Przeprowadzone badania miały na celu potwierdzenie zgodności wyników benchmarku z obliczeniami algorytmu wyszukiwania oraz określenie wydajności tych obliczeń (tabela 9).

Tabela 8. Parametry komputerów zastosowanych do weryfikacji eksperymentu. W nawiasach podana jest częstotliwość taktowania w trybie turbo dla pojedynczego rdzenia

Oznaczenie procesora	Częstotliwość taktowania	Liczba rdzeni/wątków	Pamięć operacyjna komputera
I7-2670QM	2,2 GHz (3,1 GHz)	4/8	8 GB
I7-2600	3,6 GHz (3,8 GHz)	4/8	16 GB
I9-10850K	3,6 GHz (5,2 GHz)	10/20	128 GB

Komputery pracowały pod kontrolą systemu operacyjnego Windows 10 w wersji 64 bitowej. Zainstalowany był język Python 3.10 w wersji 64 bitowej z odpowiednimi bibliotekami.

Użyty w testach benchmark wykonuje obliczenia w pętli i sprawdza wydajność obliczeń na sekundę dla wszystkich etapów pośrednich. Dla N użytych w systemie ze zbioru W wykonuje kolejne obliczenia dla n WLO, symulując procedurę wyszukania linii:

$$n \in W, \text{ dla } n = 2, 3, \dots, N \quad (26)$$

i wyświetla w podsumowaniu całkowity przewidywany czas wszystkich etapów.

Używany w porównawczych testach wydajności komputer z procesorem I7-2670QM jest komputerem przenośnym (laptop), natomiast dwa pozostałe są komputerami stacjonarnymi. W rozwiązaniach mobilnych główny nacisk położony jest na niską wagę oraz oszczędność zużywanej energii przy zadowalającej wydajności systemu komputerowego. Wiąże się to z mniejszymi częstotliwościami taktowania procesora, brakiem lub zubożonymi komponentami (np. mniejsza pamięć cache procesora, wolniejsze taktowanie magistral danych, wolniejsze pamięci itd.). dodatkowo chłodzenie procesora jest znacznie słabsze w porównaniu z komputerami stacjonarnymi. Dlatego wydajność komputera z procesorem I7-2670QM jest mniejsza niż wynikałoby to ze stosunku częstotliwości taktowania procesorów porównywanych komputerów.

Tabela 9. Porównanie wydajności obliczeniowej komputerów dla metody bezpośredniej algorytmu wyboru WLO. Wyniki benchmarku przeprowadzonego dla systemu jednokanałowego. Czas jest sumą czasu obliczeń dla konfiguracji od 2 WLO do liczby N WLO w ZWLO. W oznacza liczbę wszystkich WLO do wykorzystania

Parametry systemu „N” z „W”	Komputer		
	I7-2670QM	I7-2600	I9-10850K
	Przewidywany czas obliczeń		
8 z 32	3 godz	51 min	43 min
16 z 32	15 dni	10 dni	8 dni
32 z 32	31 dni	25 dni	12 dni

Przeprowadzono test długości czasu obliczeń dla metody bezpośredniej za pomocą benchmarku dla określonych zbiorów danych zawierających różne liczby WLO w zbiorze W . Tabela 10 zawiera wyniki dla komputera z procesorem I9-10850K.

Tabela 10. Prognozowany czas obliczeń dla komputera z procesorem I9-10850K dla metody bezpośredniej algorytmu. Czas jest sumą czasu obliczeń dla konfiguracji od 2 WLO do liczby WLO w ZWLO

Lp.	Liczba WLO w ZWLO	Liczba wszystkich WLO do wykorzystania	Czas obliczeń
1	8	16	20 s
2	16	16	40 s
3	4	32	17 s
4	8	32	43 min
5	16	32	8 dni
6	32	32	12 dni

Prognozowany czas obliczany jest dla wszystkich kolejnych etapów realizacji algorytmu. Zauważalne jest wyraźne wydłużenie czasu obliczeń dla dwukrotnie zwiększonej liczby WLO w zbiorze W do użycia. Dla zbioru W zawierającego 16 WLO (Tabela 10, wiersze 1 i 2), czas obliczeń jest znikomy w porównaniu do czasu obliczeń dla zbioru z 32 WLO (Tabela 10, wiersze 5 i 6). Również zapotrzebowanie na pamięć RAM dla procesu obliczeniowego jest niewielkie. Wynika z tego, że obecnie dowolny współczesny komputer będący w stanie uruchomić 64 bitowy system operacyjny (Windows) umożliwi przeprowadzenie obliczeń metodą bezpośrednią dla zbioru 16 WLO.

Tabela 11. Wynik działania benchmarku dla komputerów z procesorami I7-2600 oraz I9-10850K dla metody bezpośredniej

n z 32 WLO	Czas obliczeń [s]		Liczba obliczeń/s	
	I7-2600	I9-10850K	I7-2600	I9-10850K
2	0,041	0,020	12055,1	24981,4
3	0,471	0,243	10532,3	20407,8
4	4,246	2,229	8468,2	16129,3
5	27,371	15,305	7357,2	13157,3
6	147,624	84,276	6138,5	10752,7
7	625,691	366,870	5379,4	9174,5
8	2176,280	1283,039	4833,2	8198,0
9	6419,506	3871,337	4369,3	7245,2
10	16110,664	9741,150	4004,3	6622,7
11	35441,539	21676,233	3640,5	5952,3
12	68150,257	41772,039	3313,2	5405,4
13	113712,796	69474,488	3054,8	5000,0
14	167446,082	103715,733	2815,4	4545,5
15	213728,283	132947,612	2646,9	4255,2
16	242095,311	149666,481	2482,8	4016,1

n z 32 WLO	Czas obliczeń [s]		Liczba obliczeń/s	
	I7-2600	I9-10850K	I7-2600	I9-10850K
17	242556,335	150481,048	2332,3	3759,4
18	216264,454	134830,588	2179,9	3496,5
19	167132,648	104559,329	2078,4	3322,3
20	114800,824	71125,780	1966,8	3174,6
21	68601,687	42577,976	1880,8	3030,3
22	36234,994	22708,539	1780,4	2840,9
23	16455,233	10321,872	1704,6	2717,4
24	6531,085	4091,623	1610,5	2570,7
25	2176,455	1363,175	1546,5	2469,1
26	622,744	386,038	1455,2	2347,4
27	143,942	89,813	1399,0	2242,2
28	26,717	16,865	1346,0	2132,2
29	3,827	2,371	1296,1	2092,0
30	0,398	0,246	1247,6	2016,1
31	0,027	0,016	1206,8	1949,3
32*	0,001	0,001	1154,1*	1872,7*

* – wystarczy jedno obliczenie ZWLO32. Pozycja została dodana w celu zobrazowania wydajności obliczeniowej.

W tabeli 11 podano szczegółowe wyniki benchmarku dla metody bezpośredniej, dla kolejnych liczb WLO w docelowej ZWLO, dla jednego kanału pomiarowego. Komputer z procesorem I9 już od 9 WLO do 24 WLO będzie wyszukiwał najlepszą kombinację WLO w czasie dłuższym niż 1 godzinę. Przewidywany czas wszystkich obliczeń dla komputera z procesorem I7-2600 to 25 dni, a dla komputera z procesorem I9-10850K przewidywany czas obliczeń wynosi 12 dni.

Test wydajności polega na wykonaniu operacji złożenia wybranej liczby WLO do ZWLO oraz obliczeniu rozdzielczości ekwiwalentnej złożonej ZWLO. Benchmark bazuje na przybliżonych wynikach wykonania symulowanych obliczeń w pętli, w krótkim przedziale czasu dla wczytanego zbioru danych WLO. Nowoczesne procesory umożliwiają wzrost wydajności poprzez zwiększenie szybkości taktowania (jeśli obciążenie procesora na to pozwala), a także obniżenie wydajności poprzez zmniejszenie szybkości taktowania do wartości nominalnej (jeśli nastąpi wzrost temperatury procesora) lub nawet poniżej częstotliwości nominalnej w celu ochrony przed przegrzaniem. Sam proces zmiany wydajności często trwa dłużej niż chwilowe testy wydajności, co skutkuje pewnymi rozrzutami wyników. Zjawisko to jest zauważalne w przypadku wielokrotnego powtórzenia tych samych testów wydajności w krótkich odstępach czasu. Dodatkowo należy uwzględnić procesy systemu operacyjnego. Benchmark ma na celu pokazanie orientacyjnego czasu potrzebnego do znalezienia odpowiedniej konfiguracji ZWLO metodą bezpośrednią. Wyniki mogą się różnić w zależności od danych wejściowych, tj. zastosowanego zbioru WLO (np. WLO o innych liczbach segmentów niż badane).

W opracowanym programie użyto bibliotekę **itertools**, która zapewnia zestaw narzędzi do efektywnego przetwarzania iteracji oraz generowania kombinacji. Biblioteka **itertools**, dostępna w języku Python, pozwala na szybkie wygenerowanie wszystkich kombinacji danych z określonego zbioru danych, co znacznie przyspiesza działanie programu, ale angażuje więcej pamięci w procesie obliczeniowym. Zostało to przedstawione w tabeli 12. Wartości zapotrzebowania na zasoby pamięci dotyczą tylko samego procesu obliczeniowego bez uwzględnienia pamięci wymaganej przez system operacyjny.

Tabela 12. Zapotrzebowanie na zasoby pamięci RAM dla procesu obliczeniowego dla metody bezpośredniej dla zbioru 32 WLO

Liczba WLO	5	6	7	8	9	10	11	12	13	14	15	16
Pamięć [GB]	<0,1	0,1	0,4	1,2	3,2	8,4	16,8	32,5	49,9	75,6	90,7	108,4
Liczba WLO	17	18	19	20	21	22	23	24	25	26	27	28
Pamięć [GB]	98,3	90,8	65,8	47,1	27,0	14,1	6,3	2,6	0,9	0,3	0,1	<0,1

Jak pokazano w tabeli 12 zapotrzebowanie na pamięć RAM jest znaczące i zależy od liczby możliwych kombinacji (wzór 17) oraz liczby WLO. W przypadku 16 WLO ze zbioru 32 WLO zapotrzebowanie na pamięć RAM jest największe (108,4 GB). Dla 15 WLO oraz 17 WLO liczba kombinacji jest taka sama, jednak dla 17 WLO wygenerowane kombinacje są dłuższe o 2 elementy. Sytuacja jest analogiczna dla następnych liczb WLO (14 oraz 18, 13 oraz 19, itd.). Jeśli przyjąć, że przeciętne komputery posiadają obecnie 8÷16 GB pamięci RAM, to możliwe są obliczenia metodą bezpośrednią dla komputera z pamięcią 8 GB dla konfiguracji systemu pomiarowego od 1 do 9 WLO z 32 WLO, a potem powyżej 23 WLO. Dla komputera z 16 GB będą to konfiguracje odpowiednio od 1 do 10 z 32 WLO, a następnie powyżej 22 WLO. Niestety, do obliczeń użyciem biblioteki **itertools** nie może być wykorzystana pamięć wirtualna, jaką zapewnia system operacyjny. Dla liczby WLO poniżej 7 i powyżej 26 zużycie pamięci nie przekracza 0,1 GB pamięci RAM.

5.3.1 System jednokanałowy

Wydajność algorytmu w metodzie bezpośredniej

Dla sprawdzenia wydajności metody bezpośredniej określono szczegółowe wyniki czasu wyszukiwania ZWLO tą metodą dla kolejnych liczb WLO w docelowej ZWLO dla jednego kanału pomiarowego (Tabela 13). Komputer z procesorem I9 już od 9 WLO do 24 WLO będzie wyszukiwał najlepszą kombinację WLO w czasie dłuższym niż 1 godzinę, a od 14 do 19 WLO wyszukiwanie potrwa dłużej niż jedną dobę. Wyniki te są zbliżone do wyników benchmarku z tabeli 11 dla kolumny komputera z procesorem I9-10850K. Wynik czasu dla ZWLO zawierającej 16 WLO wyróżniono kolorem szarym, ponieważ dotyczy opracowanego systemu pomiarowego (16 WLO z 32 WLO).

Tabela 13. Czasy wyszukiwania ZWLO ze zbioru 32 WLO metodą bezpośrednią algorytmu z użyciem komputera z procesorem I9-10850K

liczba WLO	Czas obliczeń [s]		liczba WLO	Czas obliczeń [s]	
	[s]	dni:godz:min:sek		[s]	dni:godz:min:sek
1	0,001	00:00:00:00,001	17	147767,833	01:17:02:47,833
2	0,035	00:00:00:00,035	18	134695,250	01:13:24:55,250
3	0,318	00:00:00:00,318	19	103719,673	01:04:48:39,673
4	2,542	00:00:00:02,542	20	72248,148	00:20:04:08,148
5	16,256	00:00:00:16,256	21	43134,491	00:11:58:54,491
6	87,821	00:00:01:27,821	22	22891,171	00:06:21:31,171
7	365,658	00:00:06:05,658	23	10332,310	00:02:52:12,310
8	1329,971	00:00:22:09,971	24	4138,608	00:01:08:58,608
9	3849,558	00:01:04:09,558	25	1354,957	00:00:22:34,957
10	10311,812	00:02:51:51,812	26	384,242	00:00:06:24,242
11	22293,877	00:06:11:33,877	27	88,878	00:00:01:28,878
12	43166,76	00:11:59:26,760	28	15,022	00:00:00:15,022
13	71413,766	00:19:50:13,766	29	2,145	00:00:00:02,145
14	104167,567	01:04:56:07,567	30	0,225	00:00:00:00,225
15	129916,741	01:12:05:16,741	31	0,018	00:00:00:00,018
16	151239,874	01:18:00:39,874	32	0,001	00:00:00:00,001

Metoda dobierania do najlepszej pary

W poniżej opisanym eksperymencie wykorzystano metodę doboru do pary, dla systemu z jednym kanałem pomiarowym. W tabeli 14 wartość Δq_{eqv} jest różnicą pomiędzy znalezioną wartością rozdzielczości ekwiwalentnej metodą bezpośrednią a metodą doboru do pary, dla zadanych parametrów. Wartość ta nie przekracza w najgorszym przypadku 7% w stosunku do q_{eqv} znalezionej metodą bezpośrednią. Natomiast różnice czasu wyszukiwania metodą doboru do pary są niewspółmiernie krótkie w porównaniu do czasów metody bezpośredniej. I tak na przykład dla 16 WLO czas wyszukiwania jest $2 \cdot 10^6$ razy krótszy i wynosi 0,074 s.

Tabela 14. Wyniki porównania wartości q_{eqv} oraz czasu obliczeń dla metody bezpośredniej algorytmu oraz metody doboru do pary, ze zbioru 32 WLO, z użyciem komputera z procesorem I9-10850K

Liczba WLO	q_{eqv} [ps] bezpośrednio	q_{eqv} [ps] dobieranie	Δq_{eqv} [ps]	T [s] obliczeń m. bezpośrednia	T [s] obliczeń m. doboru do pary
2	6,3293	6,3293	0,0000	0,035	0,050
3	3,9897	4,2429	0,2532	0,318	0,028
4	3,1367	3,2273	0,0906	2,542	0,030
5	2,5377	2,6064	0,0687	16,256	0,034
6	2,1428	2,2088	0,0660	87,821	0,037

Liczba WLO	q_{eqv} [ps] bezpośrednio	q_{eqv} [ps] dobieranie	Δq_{eqv} [ps]	T [s] obliczeń m. bezpośrednia	T [s] obliczeń m. doboru do pary
7	1,8471	1,9373	0,0902	365,658	0,039
8	1,6351	1,7060	0,0709	1329,971	0,042
16	0,8701	0,8991	0,0290	151239,874	0,074

Metoda Monte Carlo

W poniższym eksperymencie wykorzystano metodę Monte Carlo, dla systemu z jednym kanałem pomiarowym. W tabeli 15 przedstawiono wartości minimalne q_{eqv} uzyskane metodą Monte Carlo i metodą bezpośrednią dla wybranych liczb WLO w ZWLO, szczególnie dla wartości których znalezienie zajmuje powyżej doby (15÷17). Wartości q_{eqv} pokazane w tabeli 15 nie przekraczają w najgorszym przypadku 2% w stosunku do wartości q_{eqv} określonej metodą bezpośrednią. Na przykład dla ZWLO z 16 WLO określono metodą Monte Carlo wartość q_{eqv} wynosila 0,885 ps, a określona metodą bezpośrednią 0,870 ps. Natomiast różnice czasu wyszukiwania metodą doboru do pary są, podobnie jak w metodzie dobierania do pary, niewspółmiernie krótkie w porównaniu do czasów w metodzie bezpośredniej.

Tabela 15. Czasy obliczeń z użyciem komputera z procesorem I9-10850K dla metody Monte Carlo, dla 10^5 iteracji i dla wybranych liczb WLO

Liczba WLO	q_{eqv} [ps]	q_{eqv_min} [ps]*	Czas obliczeń [s]
8	1,669	1,635	21,60
15	0,943	0,934	38,77
16	0,885	0,870	41,37
17	0,843	0,837	43,41

* - wartości minimalne q_{eqv} dla pojedynczej ZWLO określone metodą bezpośrednią

Metoda doboru do pary jest metodą błyskawiczną w stosunku do metody bezpośredniej, jednak wartości q_{eqv} będą różniły się od wartości znalezionych metodą bezpośrednią. Metoda Monte Carlo jest wolniejsza niż metoda doboru do pary, jednak w szczególności może dostarczyć kombinacji o mniejszej wartości q_{eqv} podobnie jak metoda bezpośrednia.

5.3.2 System wielokanałowy

Dla systemu wielokanałowego jako priorytet zakłada się uzyskanie możliwie zbliżonych parametrów każdego z kanałów. Natomiast same wartości parametrów metrologicznych, choć istotne, są drugorzędne. Poniżej przedstawiono wydajność algorytmu w zależności od wybranej metody doboru linii w systemie wielokanałowym.

Metoda bezpośrednia

W opisanym poniżej eksperymencie sprawdzono wydajność algorytmu doboru metodą bezpośrednią, dla systemów wielokanałowego, składającego się z odpowiednio dobranych WLO w wielu ZWLO. Tabela 16 zawiera czasy obliczeń dla wariantu systemu dwukanałowego, w którym każdy kanał zawierał od 2 do 8 WLO. Wartości q_{eqv_min} oraz q_{eqv_max} są to odpowiednio minimalne i maksymalne wartości rozdzielczości ekwiwalentnej q_{eqv} określonych ZWLO.

Tabela 16. Czasy trwania obliczeń z użyciem komputera z procesorem I9-10850K dla metody bezpośredniej algorytmu i 2 kanałów pomiarowych

Liczby Kanałów/WLO	q_{eqv_min} [ps] kanałów	q_{eqv_max} [ps] kanałów	Δq_{eqv} [ps]	q_{eqv_min} [ps]* ogólnie	Czas obliczeń [s]
2/2	6,9687	6,9687	$2,77 \cdot 10^{-6}$	6,3293	0,035
2/3	5,1366	5,1366	$2,26 \cdot 10^{-7}$	3,9897	1,255
2/4	3,5103	3,5103	$2,54 \cdot 10^{-8}$	3,1367	3,521
2/6	2,2911	2,2911	0,00	2,1428	90,543
2/7	1,9330	1,9330	$0,03 \cdot 10^{-5}$	1,8471	386,012
2/8	1,6404	1,6819	$4,15 \cdot 10^{-2}$	1,6351	1363,518

* - wartości minimalne q_{eqv} dla pojedynczej ZWLO określone metodą bezpośrednią

Zauważalne są spadki wydajności wraz ze zwiększaniem liczby kombinacji WLO koniecznych do weryfikacji (wzór 17). Dla systemu o konfiguracji 2 kanałów, po 2 WLO w każdym kanale, liczba kombinacji jest niewielka i można zapisać w tablicy wszystkie wyniki q_{eqv} z kombinacjami WLO. Dlatego obliczenia były krótkie i trwały 0,035 s. Dla większej liczby WLO w ZWLO wyniki zapisywane są w tablicy, która przechowuje ustaloną liczbę wyników o najmniejszych wartościach q_{eqv} wraz z odpowiadającymi im kombinacjami WLO. Tablica ta jest następnie sortowana ze względu na wartości q_{eqv} a największe wartości q_{eqv} wraz z odpowiadającymi im kombinacjami WLO są usuwane. Aby nie aktualizować całej tablicy z każdym nowym wynikiem q_{eqv} uaktualnienie wykonywane jest co 1000 iteracji pętli. Wartość ta jest kompromisem pomiędzy wydajnością a zapotrzebowaniem na pamięć RAM komputera. Analogiczne postępowanie stosowane jest dla większej liczby kanałów pomiarowych. Wyniki wydajności określone dla systemu 3 kanałowego pokazano w tabeli 17. Analogicznie jak dla systemu dwukanałowego wartości q_{eqv_min} oraz q_{eqv_max} są to odpowiednio minimalne i maksymalne wartości rozdzielczości ekwiwalentnej q_{eqv} określonych ZWLO.

Tabela 17. Czasy trwania obliczeń wykonanych z użyciem komputera z procesorem I9-10850K dla metody bezpośredniej algorytmu dla 3 kanałów

Kanały/WLO	q_{eqv_min} [ps] kanałów	q_{eqv_max} [ps] kanałów	Δq_{eqv} [ps]	q_{eqv_min} [ps]* ogólnie	Czas obliczeń [s]
3/2	7,0041	7,0046	$5,4000 \cdot 10^{-4}$	6,3293	0,038
3/4	3,5871	3,5872	$7,9756 \cdot 10^{-7}$	3,1367	4,140
3/6	2,1624	2,2398	$7,7450 \cdot 10^{-2}$	2,1428	86,060

* - wartości minimalne q_{eqv} dla pojedynczej ZWLO określone metodą bezpośrednią

Tabela 18 zawiera wyniki badań wydajności dla systemu 4 kanałowego.

Tabela 18. Czasy trwania obliczeń wykonanych z użyciem komputera z procesorem I9-10850K dla metody bezpośredniej algorytmu dla 4 kanałów

Kanały/WLO	q_{eqv_min} [ps] kanałów	q_{eqv_max} [ps] kanałów	Δq_{eqv} [ps]	q_{eqv_min} [ps]* ogólnie	Czas obliczeń [s]
4/2	7,0041	7,0056	$1,4197 \cdot 10^{-3}$	6,3293	0,040
4/3	4,8691	4,8692	$5,3176 \cdot 10^{-5}$	3,9897	1,899
4/4	3,5933	3,5933	$4,7274 \cdot 10^{-6}$	3,1367	5,611
4/5	2,6801	2,6806	$0,5740 \cdot 10^{-3}$	2,5377	16,428
4/6	3,4541	3,4552	$1,0113 \cdot 10^{-2}$	2,1428	3,421

* - wartości minimalne q_{eqv} dla pojedynczej ZWLO określone metodą bezpośrednią

W tabeli 18 zauważalna jest anomalia czasu obliczeń dla konfiguracji 4 kanałów po 6 WLO. Czas ten jest znacznie krótszy niż dla konfiguracji 4 kanałów po 5 WLO. Powodem skrócenia czasu obliczeń jest wartość q_{eqv} znacząco odbiegająca od wartości minimalnej q_{eqv} i algorytm łatwiej mógł znaleźć takie kombinacje WLO w ZWLO, aby już wykorzystane WLO w danym kanale pomiarowym nie pojawiały się w kolejnych kanałach pomiarowych. Kombinacje ZWLO o najmniejszych wartościach q_{eqv} w większości przypadków wykorzystywały wspólne WLO, co dla systemu wielokanałowego jest niedopuszczalne. Zauważalna jest tu również zależność wzrostu wartości rozdzielczości ekwiwalentnych ZWLO, związana z liczbą wykorzystanych WLO w docelowym systemie pomiarowym, która skutkuje zmniejszeniem liczby możliwych kombinacji w miarę wykorzystania WLO ze zbioru W . Przykładowo dla ZWLO4 w systemie 1 kanałowym wartość rozdzielczości ekwiwalentna q_{eqv} wynosi 3,1367 ps, w systemie 2 kanałowym q_{eqv} wynosi 3,5103 ps (2/4 tabela 16), w systemie 3 kanałowym ma ona wartość 3,5871 ps (3/4 tabela 17), zaś w systemie 4 kanałowym jest równa 3,5933 ps (4/4 tabela 18).

Metoda dobierania do najlepszych par

Wybrane wyniki badań zastosowania metody doboru do najlepszych par pokazane są w tabeli 19.

Tabela 19. Czasy trwania obliczeń określone dla komputera z procesorem I9-10850K dla metody doboru do pary dla wielu kanałów

Liczby Kana- łów/WLO	q_{eqv_sr} [ps] kanałów	Δq_{eqv} [ps] kanałów	q_{eqv_min} [ps]*	Czas obliczeń [s]
2/3	5,135	$0,113 \cdot 10^{-3}$	3,989	0,12
2/4	3,921	$1,532 \cdot 10^{-3}$	3,137	0,14
2/8	1,952	$0,188 \cdot 10^{-3}$	1,635	0,16
2/16	1,028	0,043	0,870	0,25
4/8	1,961	0,056	1,635	3,81
5/6	2,649	0,040	2,143	47,78
6/5	3,004	0,094	2,538	541,76

* - wartości minimalne q_{eqv} dla pojedynczej ZWLO określone metodą bezpośrednią

Wartość średnia rozdzielczości ekwiwalentnej q_{eqv_sr} jest średnią wartością rozdzielczości ekwiwalentnej q_{eqv} wszystkich ZWLO w danej konfiguracji systemu, zaś Δq_{eqv} jest różnicą pomiędzy maksymalnymi i minimalnymi wartościami q_{eqv} ZWLO. Zauważalny jest spadek wydajności algorytmu wraz ze zwiększeniem liczby kanałów pomiarowych, co objawia się coraz dłuższymi czasami obliczeń. Spowodowane jest to zapewnieniem przez algorytm dobierania do najlepszych par zbliżonych wartości q_{eqv} wszystkich kanałów przy jednoczesnym wyczerpywaniu się wolnych WLO. Jednak czasochłonność obliczeń przyrasta wolniej, niż dla poprzednich algorytmów (kolumna czasu obliczeń, wiersze 1–5 tabeli 19).

Metoda Monte Carlo

Wybrane wartości czasów wyszukiwania kombinacji WLO w ZWLO za pomocą metody Monte Carlo dla konfiguracji wielokanałowej pokazane są w tabeli 20. Ze względu na losowy charakter metody, uzyskane wartości q_{eqv} mogą się różnić pomiędzy kolejnymi wywołaniami metody.

Tabela 20. Czasy trwania obliczeń określone dla komputera z procesorem I9-10850K dla metody Monte Carlo dla 10^5 iteracji

Liczby kanałów/WLO	q_{eqv_sr} [ps] kanałów	Δq_{eqv_sr} [ps] kanałów	q_{eqv_min} [ps]*	Czas obliczeń [s]
16/2	7,310	0,663	6,329	42,80
8/4	3,692	0,109	3,137	31,18
4/8	1,995	0,004	1,635	26,62
2/16	0,983	$9,092 \cdot 10^{-8}$	0,870	25,51

* - wartości minimalne q_{eqv} dla pojedynczej ZWLO określone metodą bezpośrednią

5.3.3 System z rekonfigurowalnymi kanałami pomiarowymi

Liczba wszystkich możliwych kombinacji dla system rekonfigurowalnego dla zbioru W 32 WLO to ok. $3,3 \cdot 10^{18}$ (tabela 5), co jest wielkim wyzwaniem obliczeniowym. Konieczne jest bowiem sprawdzenie i zapamiętanie wszystkich możliwych kombinacji WLO, przynależnej do kanałów pomiarowych oraz wartości rozdzielczości ekwiwalentnych q_{eqv} , a następnie wybranie najlepszych z punktu widzenia parametrów metrologicznych. Dlatego dla tej konfiguracji systemu pomiarowego zrezygnowano z prób metodą bezpośrednią i skupiono się na metodzie Monte Carlo. Należy zauważyć, jak wcześniej wspomniano, że przykładowo wybór z najlepszych kombinacji ZWLO2 i złożenie ich w ZWLO4 wcale nie gwarantuje dobrych parametrów tychże i następnych, czyli ZWLO4, ZWLO8 itd. Podobnie jest na wyższych poziomach drzewa.

Metoda – Monte Carlo

Przykładowe wyniki działania algorytmu dla rekonfigurowalnego systemu w wariacie wyszukiwania metodą Monte Carlo pokazano w tabeli 21. System rekonfigurowalny oznacza zmienną liczbę kanałów oraz zmienną liczbę WLO w kanale. W wyniku działania tego wariantu algorytmu wyboru WLO ze względu na losowość metody Monte Carlo, poszczególne wyniki wartości q_{eqv_sr} między sesjami mogą różnić. Przykładowo w trzech kolejnych sesjach dla samej tylko konfiguracji 2 kana-

łów z ZWLO16 uzyskano wartości średnie rozdzielczości ekwiwalentnej q_{eqv_sr} oraz różnice Δq_{eqv} (w nawiasach) odpowiednio: 0,9597 ps (0,0733 ps), 0,9951 ps (0,0346 s), 0,9773 ps (0,0475 ps).

Tabela 21. Wyniki obliczeń q_{eqv} dla metody Monte Carlo w wariancie całościowym dla systemu o rekonfigurowalnej liczbie kanałów z puli 32 WLO, dla jednej konfiguracji, dla komputera z procesorem I9-10850K

Liczby kanałów/WLO	Sesja 1 – 10^4 iteracji		Sesja 2 – 10^5 iteracji	
	Czas 39,8 s		Czas 397,1 s	
	q_{eqv_sr} [ps]	Δq_{eqv} [ps]	q_{eqv_sr} [ps]	Δq_{eqv} [ps]
2/16	0,9799	0,0074	0,9614	0,0423
4/8	1,9672	0,2001	1,8876	0,0907
8/4	3,8196	0,3963	3,6856	0,1825
16/2	7,3904	0,8213	7,2495	1,0059

W tabeli 21 pokazano wyniki przykładowych dwóch sesji, które różniły się liczbą iteracji. Sesja 1 która wykonywała się 10^4 iteracji trwała 39,8 s oraz sesja 2 wykonywała się 10^5 iteracji której obliczenia trwały 397,1 sekund. Zauważalna jest tu w przybliżeniu liniową zależność czasu od liczby iteracji. Jako wynik obliczeń w sesji uzyskuje się kompletny system rekonfigurowalny o konfiguracji wyspecyfikowanej w pierwszej kolumnie tabeli. Jak można zauważyć, zwiększenie iteracji w sesji 2 spowodowało uzyskanie mniejszych wartości średnich wartości rozdzielczości q_{eqv_sr} dla wszystkich kanałów (większe prawdopodobieństwo uzyskania mniejszych wartości q_{eqv_sr} oraz Δq_{eqv}).

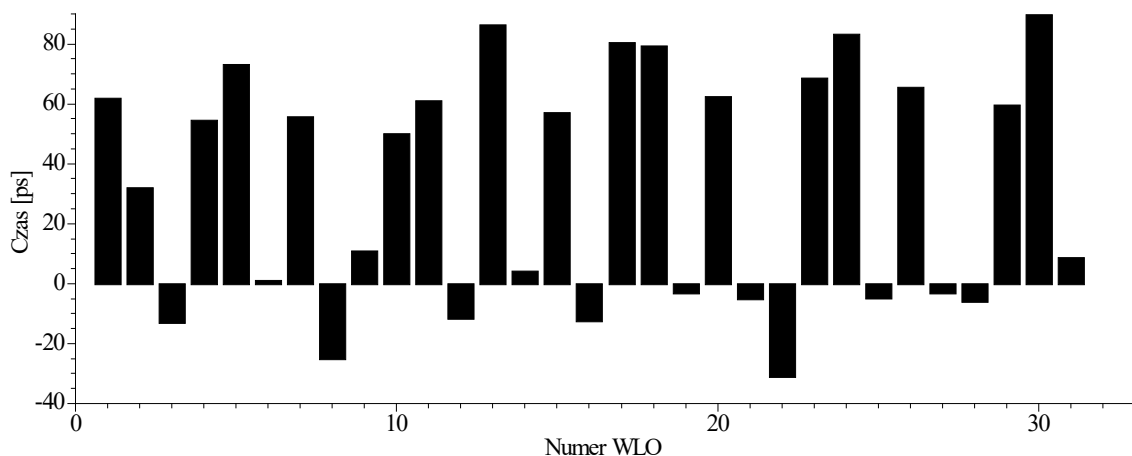
6. Weryfikacja eksperymentalna algorytmu

W celu przeprowadzenia eksperymentalnej weryfikacji algorytmu wyboru WLO zaimplementowano 32 wielosegmentowe linie opóźniające w układzie Kintex UltraScale XCKU040. Wszystkie WLO zostały zaimplementowane w domenie X0Y0, a swoje początki miały we współrzędnej pionowej Y0 i współrzędnej poziomej Xn, gdzie n jest numerem WLO (tabela 22). Odczepy wielosegmentowych linii opóźniających zostały posortowane, a następnie obliczono wartości parametrów q_{sr} , q_{eqv} , INL_{max} dla WLO (tabela 23). Karta KCU105 z układem Kintex UltraScale w czasie pomiarów znajdowała się w komorze klimatycznej zapewniającej stałą temperaturę $20 \pm 0,5^\circ\text{C}$.

Tabela 22. Lokalizacja WLO zaimplementowanych w układzie Kintex UltraScale

WLO	Lokalizacja	WLO	Lokalizacja	WLO	Lokalizacja	WLO	Lokalizacja
0	X0Y0:Y59	8	X8Y0:Y59	16	X16Y0:Y59	24	X24Y0:Y59
1	X1Y0:Y59	9	X9Y0:Y59	17	X17Y0:Y59	25	X25Y0:Y59
2	X2Y0:Y59	10	X10Y0:Y59	18	X18Y0:Y59	26	X26Y0:Y59
3	X3Y0:Y59	11	X11Y0:Y59	19	X19Y0:Y59	27	X27Y0:Y59
4	X4Y0:Y59	12	X12Y0:Y59	20	X20Y0:Y59	28	X28Y0:Y59
5	X5Y0:Y59	13	X13Y0:Y59	21	X21Y0:Y59	29	X29Y0:Y59
6	X6Y0:Y59	14	X14Y0:Y59	22	X22Y0:Y59	30	X30Y0:Y59
7	X7Y0:Y59	15	X15Y0:Y59	23	X23Y0:Y59	31	X31Y0:Y59

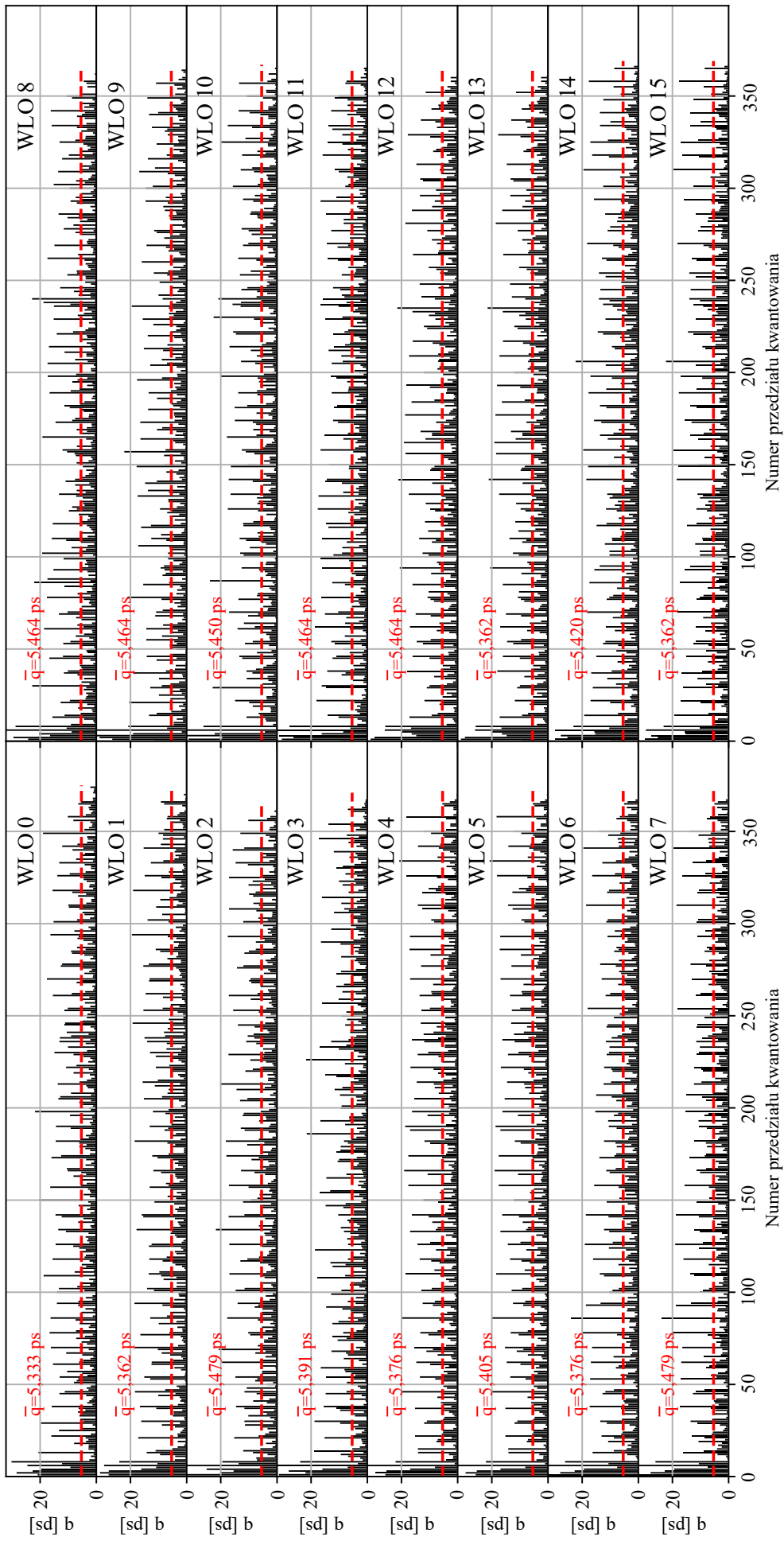
Uzyskane w wyniku badań eksperymentalnych parametry WLO zaimplementowanych w obszarze X0Y0 badanego układu Kintex UltraScale przedstawione są w tabeli 23. Wszystkie wielosegmentowe linie opóźniające ponumerowane są od 0 do 31. Natomiast na rysunkach 62 i 63 pokazano charakterystyki 32 badanych WLO. Przesunięcia czasowe kolejnych WLO względem WLO0 zaprezentowane są na rys. 61.



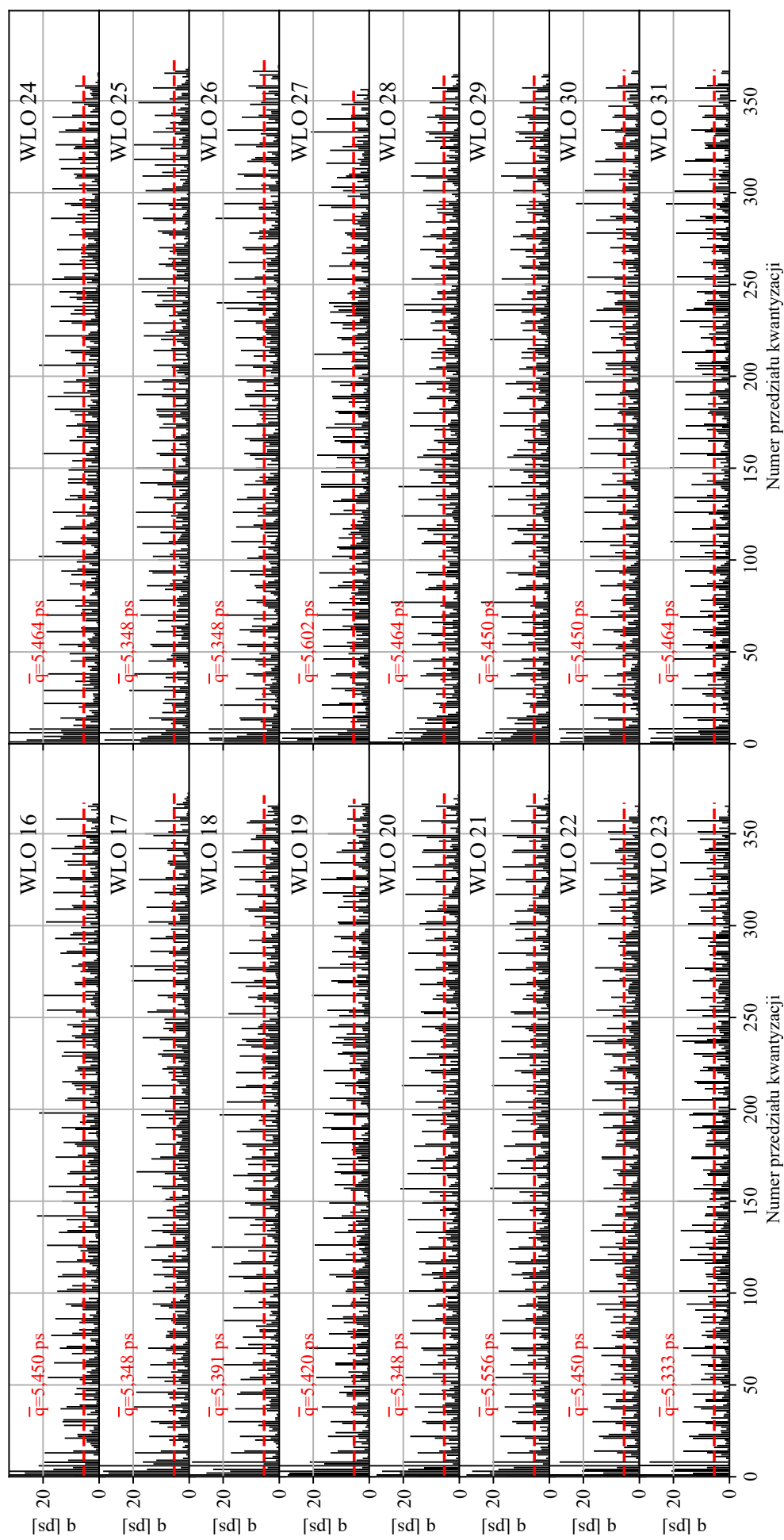
Rys. 61. Przesunięcia czasowe pomiędzy WLO zaimplementowanymi w badanym układzie Kintex UltraScale w kafelku X0Y0 względem WLO0

Tabela 23. Parametry 32 WLO zaimplementowanych w układzie Kintex UltraScale w celu weryfikacji algorytmu doboru WLO. W tabeli zawarto parametry przed i po uwzględnieniu przesunięcia pomiędzy daną WLO a WLO0

nr WLO	przed uwzględnieniem przesunięcia				przesunięcie względem WLO0 [ps]	po uwzględnieniu przesunięcia			
	liczba kanałów	q_{sr} [ps]	q_{eqv} [ps]	INL_{max} [lsb]		liczba kanałów	q_{sr} [ps]	q_{eqv} [ps]	INL_{max} [lsb]
0	375	5,333	12,518	26,793	0,000	375	5,333	12,518	26,793
1	373	5,362	12,863	27,706	61,938	374	5,348	12,851	16,932
2	365	5,479	13,496	25,034	31,972	366	5,464	13,465	15,058
3	371	5,391	12,808	25,316	-13,163	372	5,376	12,418	24,032
4	372	5,376	13,555	24,870	54,633	373	5,362	13,548	13,400
5	370	5,405	13,635	26,907	73,153	371	5,391	13,634	19,483
6	372	5,376	13,132	27,228	1,032	373	5,362	13,132	22,620
7	365	5,479	14,751	25,996	55,567	366	5,464	14,694	14,315
8	366	5,464	13,031	29,674	-25,297	367	5,450	12,827	26,378
9	366	5,464	13,438	25,979	10,839	367	5,450	13,426	19,214
10	367	5,450	14,726	25,815	50,170	368	5,435	14,726	16,239
11	366	5,464	13,632	27,340	60,968	367	5,450	13,630	17,480
12	366	5,464	12,820	26,212	-11,741	365	5,479	12,677	24,893
13	373	5,362	12,568	25,467	86,214	374	5,348	12,552	17,349
14	369	5,420	12,952	26,903	4,295	370	5,405	12,944	23,822
15	373	5,362	12,821	24,296	57,125	374	5,348	12,821	16,889
16	367	5,450	13,434	25,967	-12,710	368	5,435	13,192	24,710
17	374	5,348	13,590	27,323	80,293	375	5,333	13,562	17,107
18	371	5,391	14,058	26,216	79,444	372	5,376	14,045	17,229
19	369	5,420	13,724	26,995	-3,474	370	5,405	13,719	24,186
20	374	5,348	13,391	25,658	62,285	375	5,333	13,352	12,537
21	360	5,556	13,181	25,353	-5,490	361	5,540	13,181	23,896
22	367	5,450	13,192	25,627	-31,405	368	5,435	12,185	20,923
23	375	5,333	12,839	24,040	68,535	376	5,319	12,837	13,653
24	366	5,464	13,559	27,125	83,308	367	5,450	13,495	17,651
25	374	5,348	13,322	27,424	-5,169	375	5,333	13,316	20,296
26	374	5,348	13,038	26,688	65,534	375	5,333	13,035	15,192
27	357	5,602	13,921	26,310	-3,375	358	5,587	13,921	26,346
28	366	5,464	12,508	25,223	-6,289	367	5,450	12,461	24,337
29	367	5,450	13,774	26,818	59,684	368	5,435	13,768	16,524
30	367	5,450	13,313	25,979	89,828	368	5,435	13,306	16,759
31	366	5,464	13,191	26,779	8,702	367	5,450	13,187	22,156



Rys. 62. Charakterystyki pierwszych 16 WLO zaimplementowanych w badanym układzie Kintex UltraScale. Czerwoną linią zaznaczono średnią szerokość przedziału kwantowania \bar{q} .



Rys. 63. Charakterystyki pozostałych 16 WLO zaimplementowanych w badanym układzie Kintex UltraScale. Czerwoną linią zaznaczono średnią szerokość przedziału kwantowania \bar{q} .

Jednymi z głównych miar jakości przetworników czasowo-cyfrowych są błędy nieliniowości różnicowej DNL oraz nieliniowości sumacyjnej INL. Błąd nieliniowości różnicowej DNL opisuje odchylenie wartości szerokości konkretnego przedziału kwantowania od wartości średniej dla danego przetwornika. Wartość ta, w przypadku linii opóźniającej wykonanej w układzie scalonym, zależy od rozrzutów produkcyjnych struktury układu scalonego, połączeń pomiędzy elementami przetwornika, przesłuchami pomiędzy sygnałami, a także szumami zasilania. Wartość błędu nieliniowości różnicowej $DNL(i)$ dla i -tego przedziału kwantowania oblicza się jako różnicę pomiędzy szerokością i -tego przedziału kwantowania i średnią szerokością przedziału, podzieloną przez średnią szerokość przedziału kwantowania :

$$DNL(i)[LSB] = \frac{\tau_i - \bar{\tau}}{\bar{\tau}}, \quad (27)$$

gdzie $\bar{\tau}$ oznacza średnią szerokość przedziału kwantowania, która jest średnią arytmetyczną szerokości przedziałów kwantowania linii opóźniającej o N segmentach, a τ_i jest szerokością przedziału kwantowania i -tego segmentu, $i=1,2,..N$.

Na ogół jako wartość błędu nieliniowości różnicowej podaje się jego wartość maksymalną $DNL(i)$ wyrażoną wzorem:

$$DNL[LSB] = \max_i \left\{ \left| \frac{\tau_i - \bar{\tau}}{\bar{\tau}} \right| \right\}, \quad (28)$$

dla i -tego przedziału kwantowania $i=1,2,..N$.

Błąd INL jest skumulowanym błędem rozrzutów przedziałów kwantowania (DNL). Odchylenia wartości w przedziałach kwantowania sumują się, co skutkuje nieliniowością charakterystyki przetwarzania interpolatora.

$$INL(i)[LSB] = \sum_{j=1}^i DNL(j) \quad (29)$$

Podobnie jak dla błędu DNL, definiuje się maksymalną wartość błędu nieliniowości sumacyjnej INL:

$$INL[LSB] = \max_i \{ INL(i) \}. \quad (30)$$

Wartości DNL oraz INL można również wyrażać w jednostkach czasu. W tym celu należy pomnożyć wartości $DNL[LSB]$ oraz $INL[LSB]$ przez wartość średnią czasu trwania przedziału kwantowania $\bar{\tau}$.

Określono charakterystyki błędów nieliniowości dla analizowanych wcześniej 32 WLO zaimplementowanych w układzie Kintex UltraScale. Na rysunkach 64 i 65 pokazane są charakterystyki DNL dla 32 WLO, a na rysunkach 66 i 67 charakterystyki INL tych linii.

Po uwzględnieniu przesunięć czasowych pomiędzy WLO (rys. 55) zmieniły się charakterystyki badanych linii. Zauważalne są przesunięcia całych charakterystyk na dalsze miejsca, szczególnie widoczne dla początkowych szerokich kanałów. Najbardziej widoczne jest to w zmianach charakterystyk INL, co przedstawiono na rys. 70 oraz na rys. 71. Procedura przesunięcia czasowego

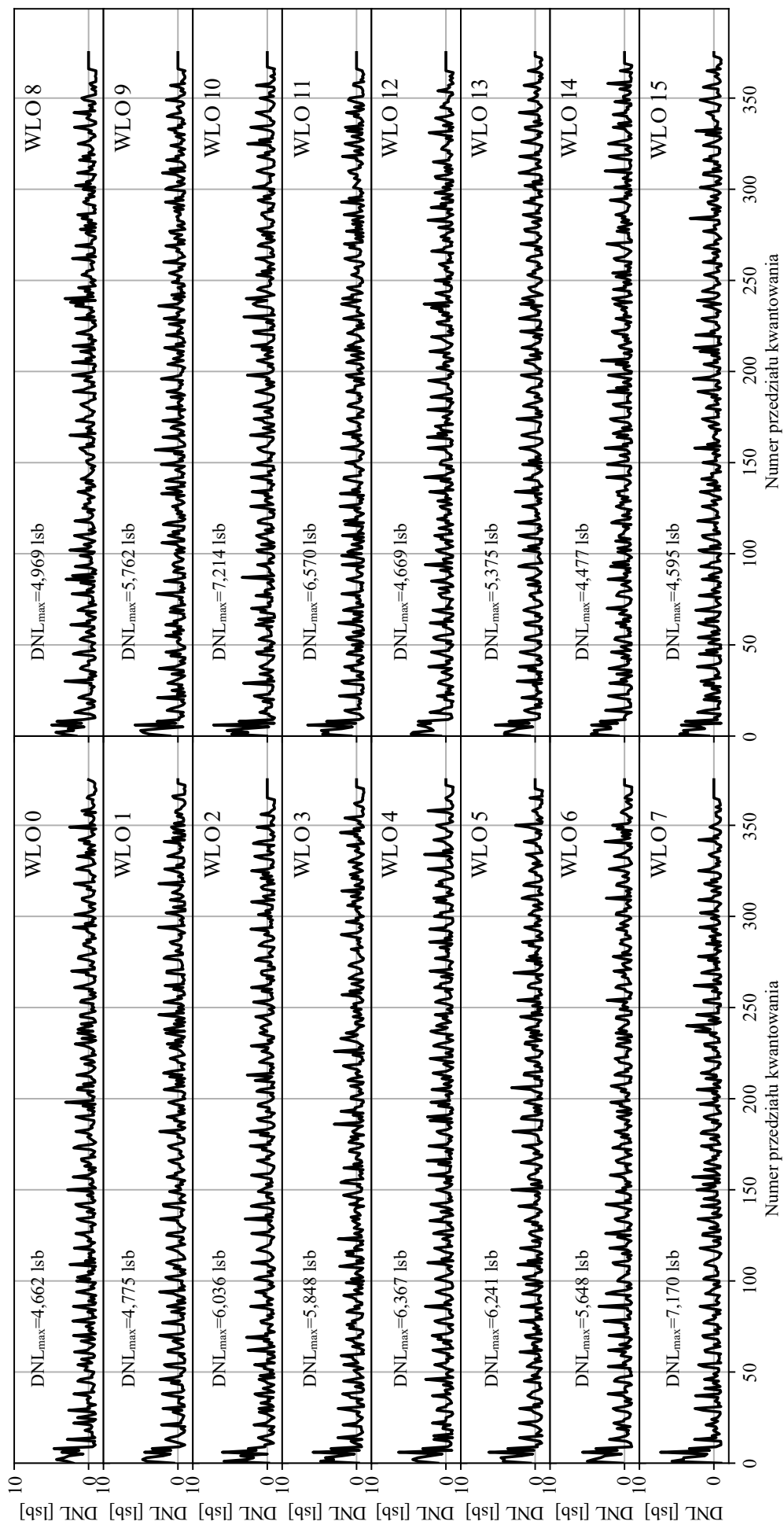
została przeprowadzona względem referencyjnej linii WLO0, dlatego też, charakterystyka i błędy DNL i INL dla tej linii pozostają niezmiennione.

Wielosegmentowe linie opóźniające zbudowane z użyciem łańcuchów przeniesień arytmetycznych zaimplementowane w układzie Kintex UltraScale charakteryzują się dużymi nieliniowościami charakterystyk, co skutkuje znacznymi wartościami rozdzielczości ekwiwalentnej q_{eqv} (około 2,5 krotna wartość \bar{q}). W początkowych przedziałach kwantowania zauważalne są takie przedziały q_i , które wykazują wartości wyraźnie przekraczające wartości średnie. Jest to cecha charakterystyczna układów Kintex UltraScale.

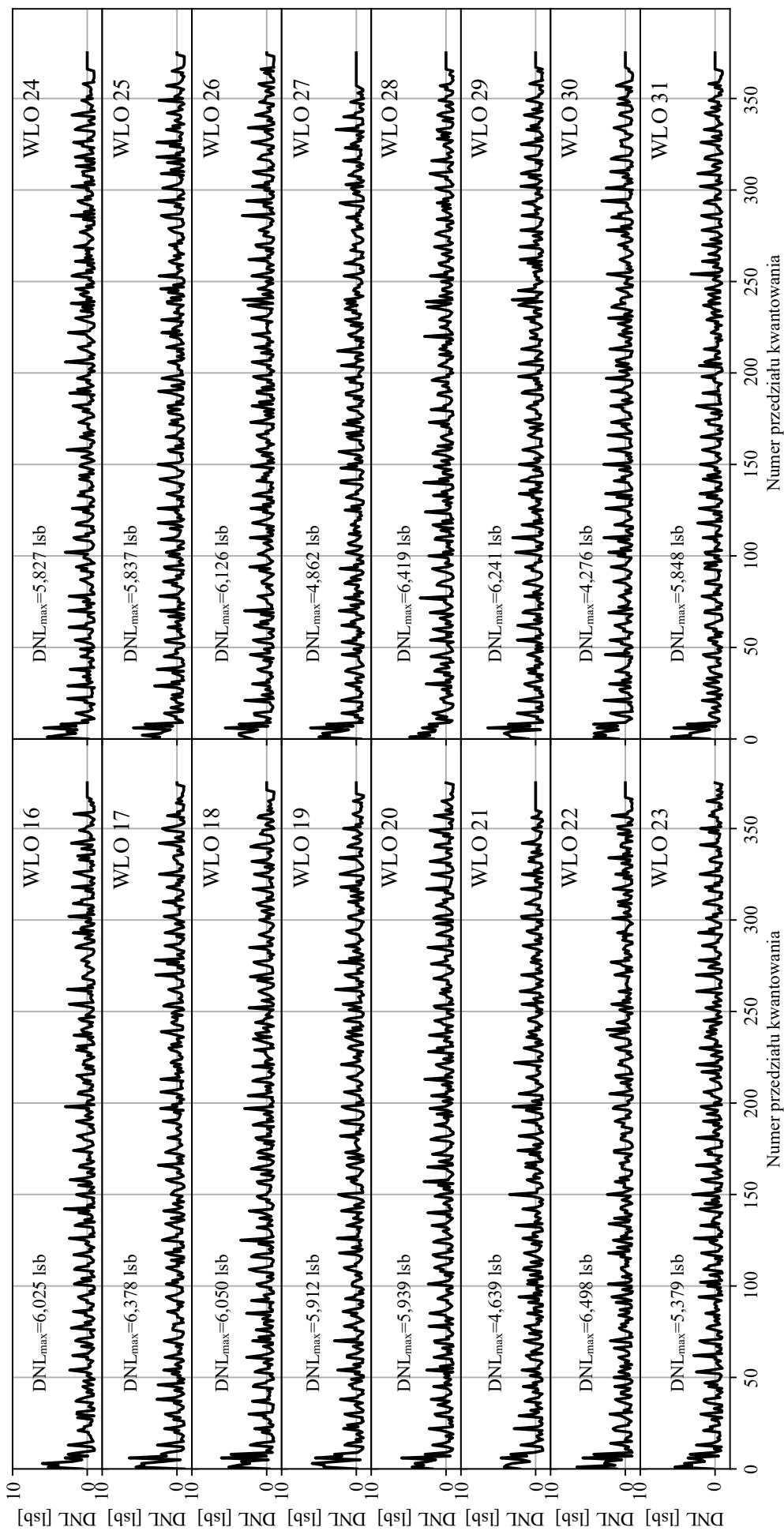
Jak można zauważyć, po porównaniu charakterystyk odpowiednio z rys. 64 i rys. 68 oraz z rys. 65 z rys. 69, uwzględnienie przesunięcia czasowego pomiędzy poszczególnymi liniami opóźniającymi względem WLO0, spowodowało przesunięcie początkowych przedziałów kwantowania w dalsze miejsca (w zależności od wartości przesunięcia czasowego). Skutkuje to wyraźnymi zmianami charakterystyk sumacyjnych, które przed uwzględnieniem przesunięcia czasowego wykazują zbliżony charakter (rysunki 66 i 67), jednak uwzględnienie przesunięć czasowych spowodowało w tym wypadku znaczne różnice w kształtach tych charakterystyk (dla porównania rys. 66 z 70 i odpowiednio rys. 67 z rys 71). Na wykresach tych przesunięciu uległ skok wartości *INL* z początkowych przedziałów kwantowania na dalsze przedziały jak również przesunięciu uległy wykresy w osi rzędnych (na przykład WLO4 na rysunkach 66 i 70), co poskutkowało zmianą wartości INL_{max} .

W tabeli 24 przedstawiono zakresy możliwych wartości q_{eqv} oraz czasy obliczeń na komputerze z procesorem I9-10850. Natomiast zakresy wartości q_{eqv} dla ZWLO przedstawiono na rys. 72. Znalezienie podzbioru 16 wielosegmentowych linii opóźniających ze zbioru 32 WLO, metodą bezpośrednią (sprawdzenie wszystkich możliwych kombinacji) zajęło ok. 42 godziny. Wynika z tego, że metoda ta jest kosztowna obliczeniowo i długotrwała dla zbioru 32 WLO.

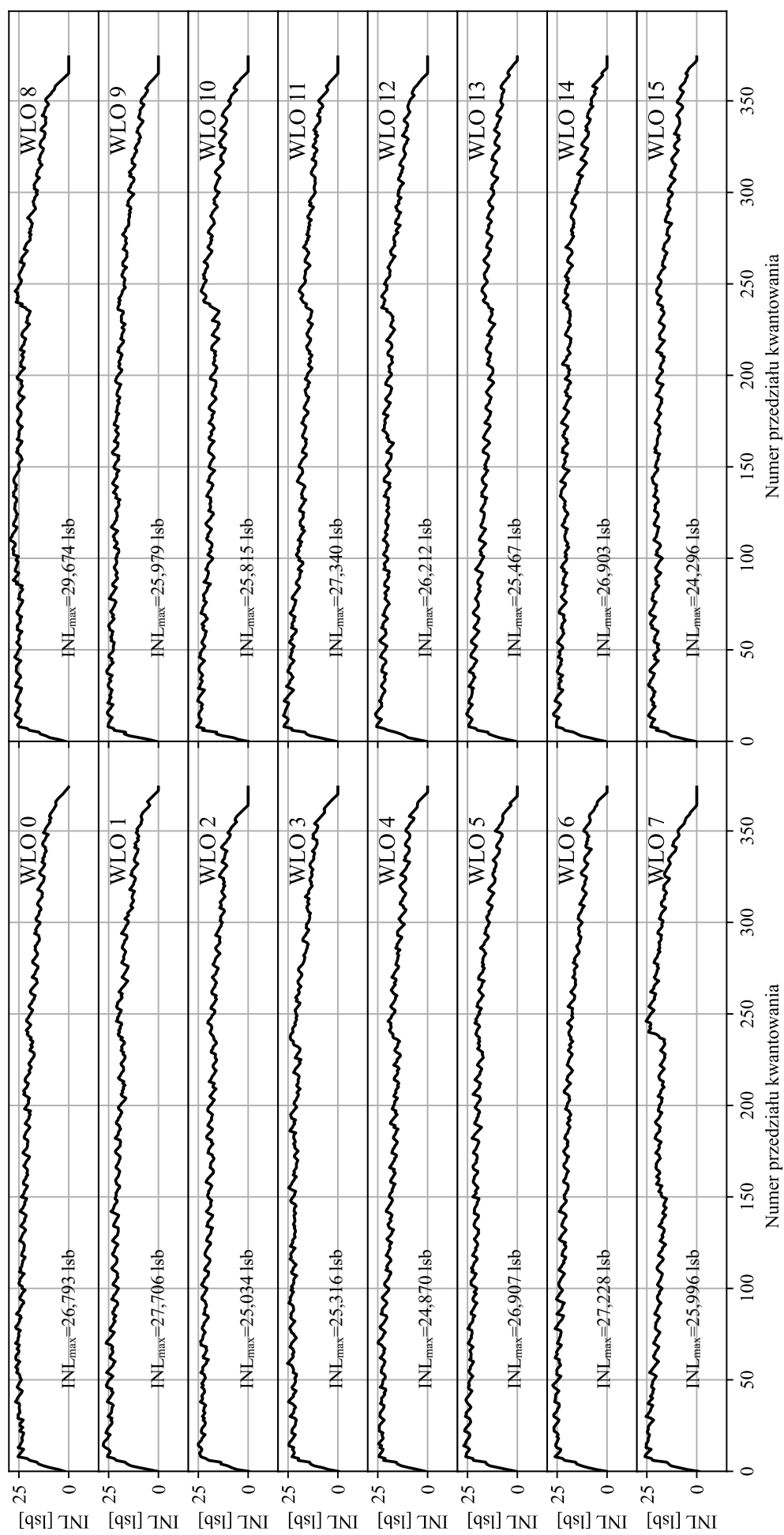
Wartości $q_{eqv,min}$ oraz $q_{eqv,max}$ przedstawione w tabeli 24, są wartościami obliczonymi metodą bezpośrednią poprzez sprawdzenie wszystkich możliwych kombinacji WLO wchodzących w skład jednej ZWLO. Zauważalna jest różnica czasu dla ZWLO złożonej z 17 WLO w porównaniu do ZWLO złożonej z 16 WLO mimo że możliwych kombinacji jest mniej, to jednak obliczenia trwają dłużej, ponieważ ZWLO17, której obliczana jest wartość q_{eqv} , posiada więcej segmentów, niż dla ZWLO16. Zauważalne jest to również na etapie sprawdzania wydajności komputera i algorytmu (benchmark tabela 11). Sumaryczny czas samych obliczeń to 299,7 godziny, czyli około 12,5 dnia. Są to wartości zbliżone do wartości podanych w tabeli 11 (przykładowo dla 16 WLO 41,57 h, wszystkie obliczenia około 299,2 h). Niewielkie odstępstwo wynika z tego, że benchmark nie uwzględnia procesów systemu operacyjnego (np. aktualizacje systemu w tle, obsługę sieci lokalnej) oraz czasu potrzebnego na alokację pamięci RAM komputera, niezbędnej do przeprowadzenia obliczeń.



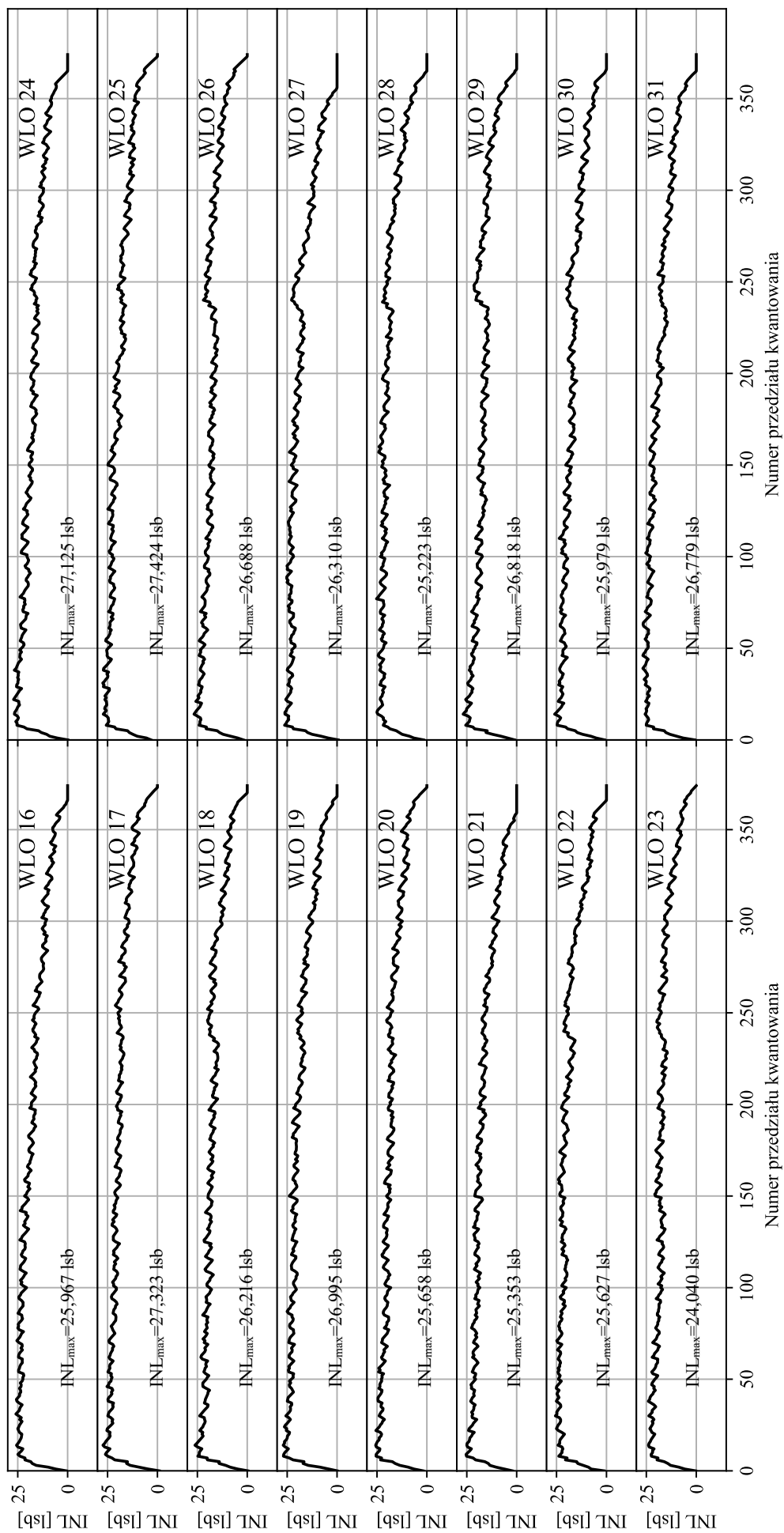
Rys. 64. Charakterystyki DNL[LSB] pierwszych 16 WLO (WLO0 – WLO15) zaimplementowanych w domenie XOY0 układu Kintex UltraScale



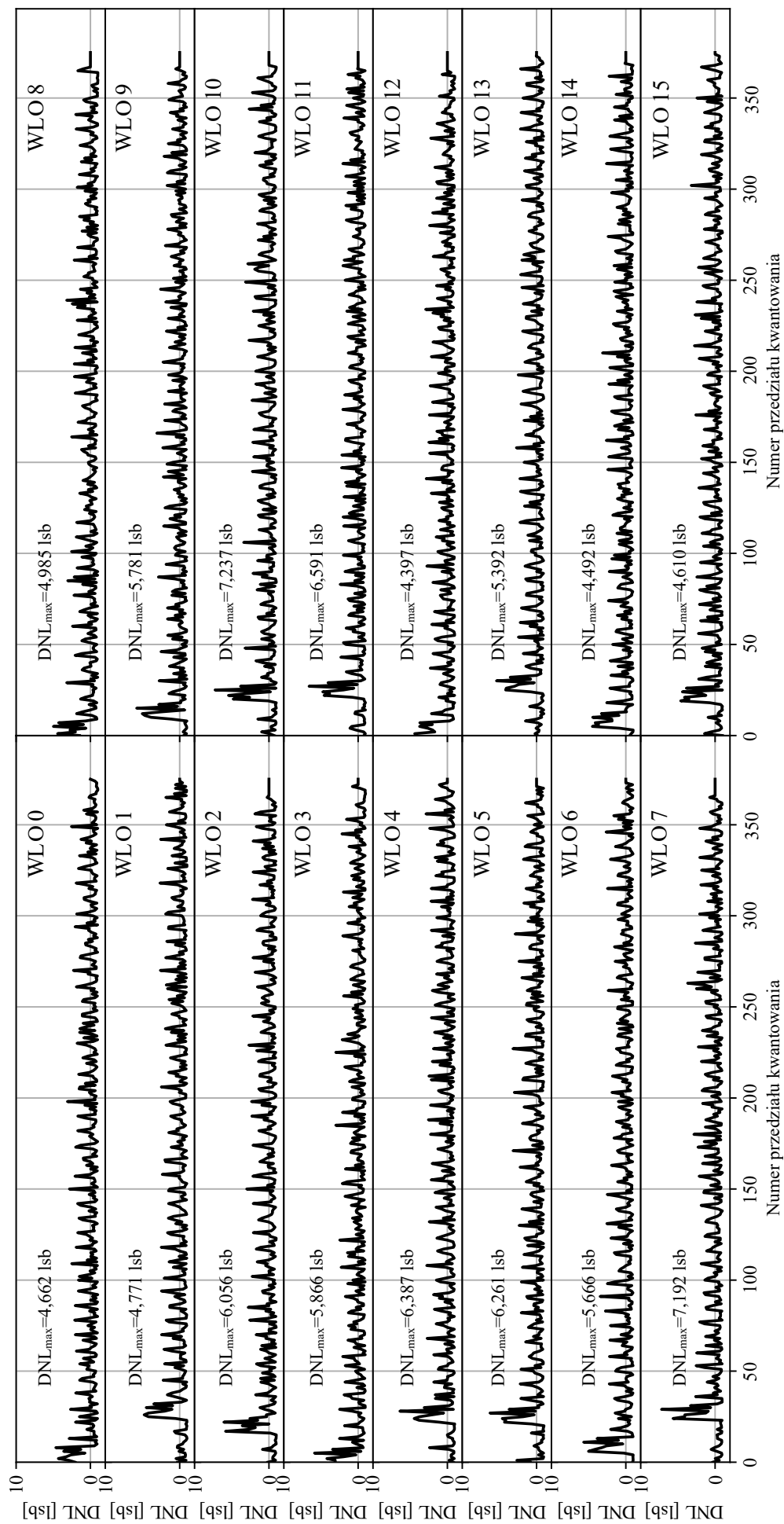
Rys. 65. Charakterystyki DNL[LSB] pozostałych 16 WLO (WLO16 – WLO31) zaimplementowanych w domenie X0Y0 układu Kintex UltraScale



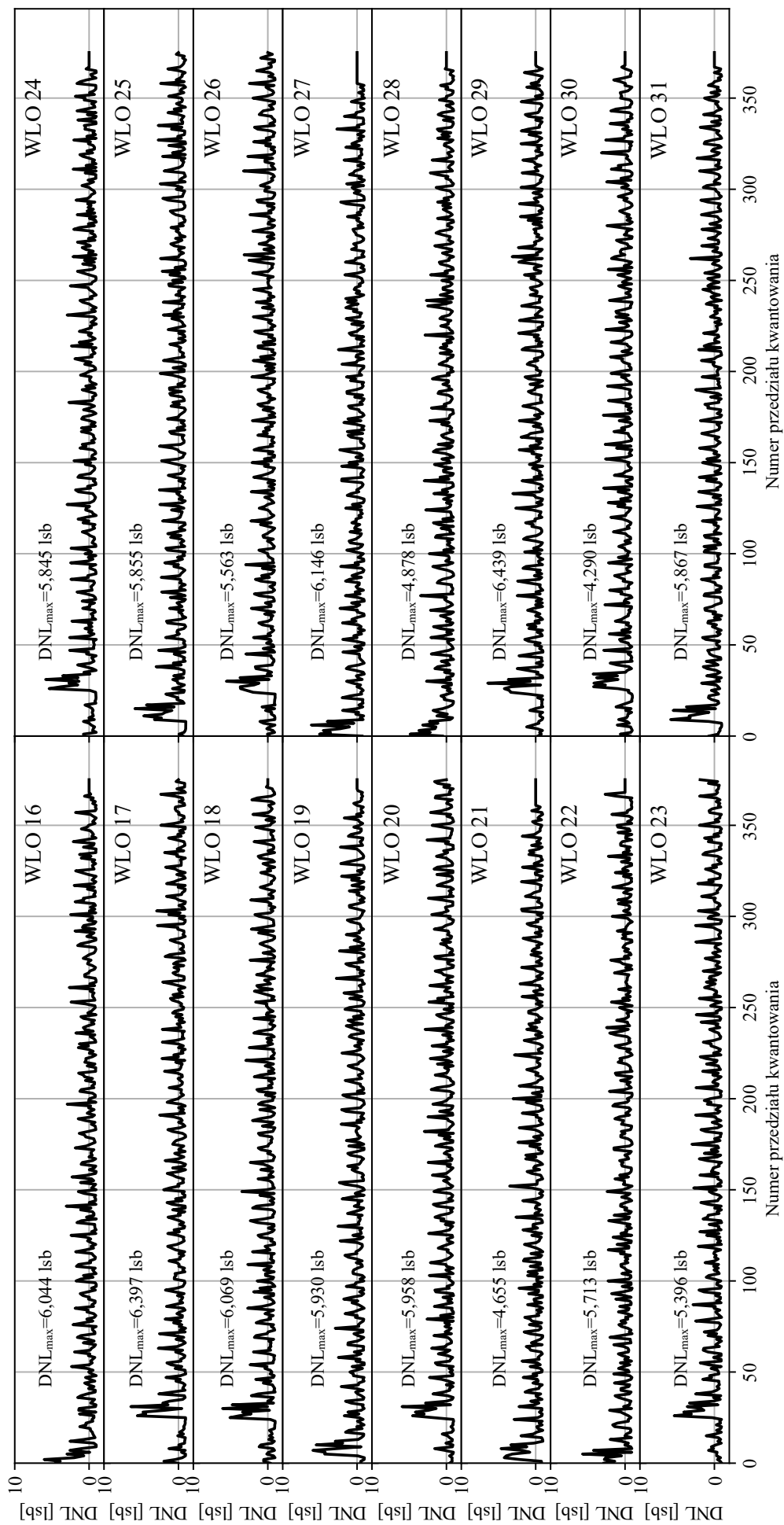
Rys. 66. Charakterystyki INL[LSB] pierwszych 16 WLO (WLO0 – WLO15) zaimplementowanych w domenie X0Y0 układu Kintex UltraScale



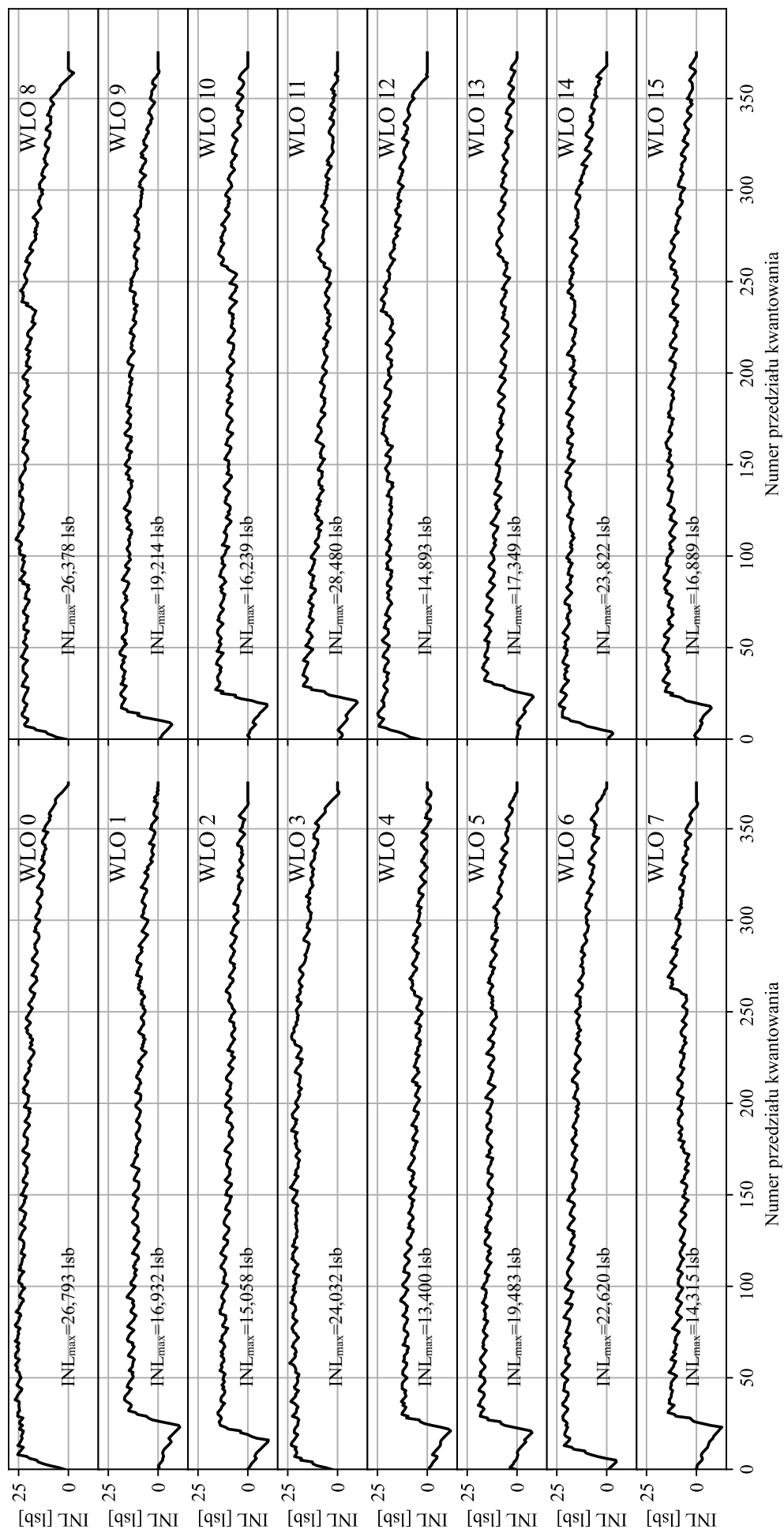
Rys. 67. Charakterystyki INL[LSB] pozostałych 16 WLO (WLO16 – WLO31) zaimplementowanych w domenie XOY0 układu Kintex UltraScale



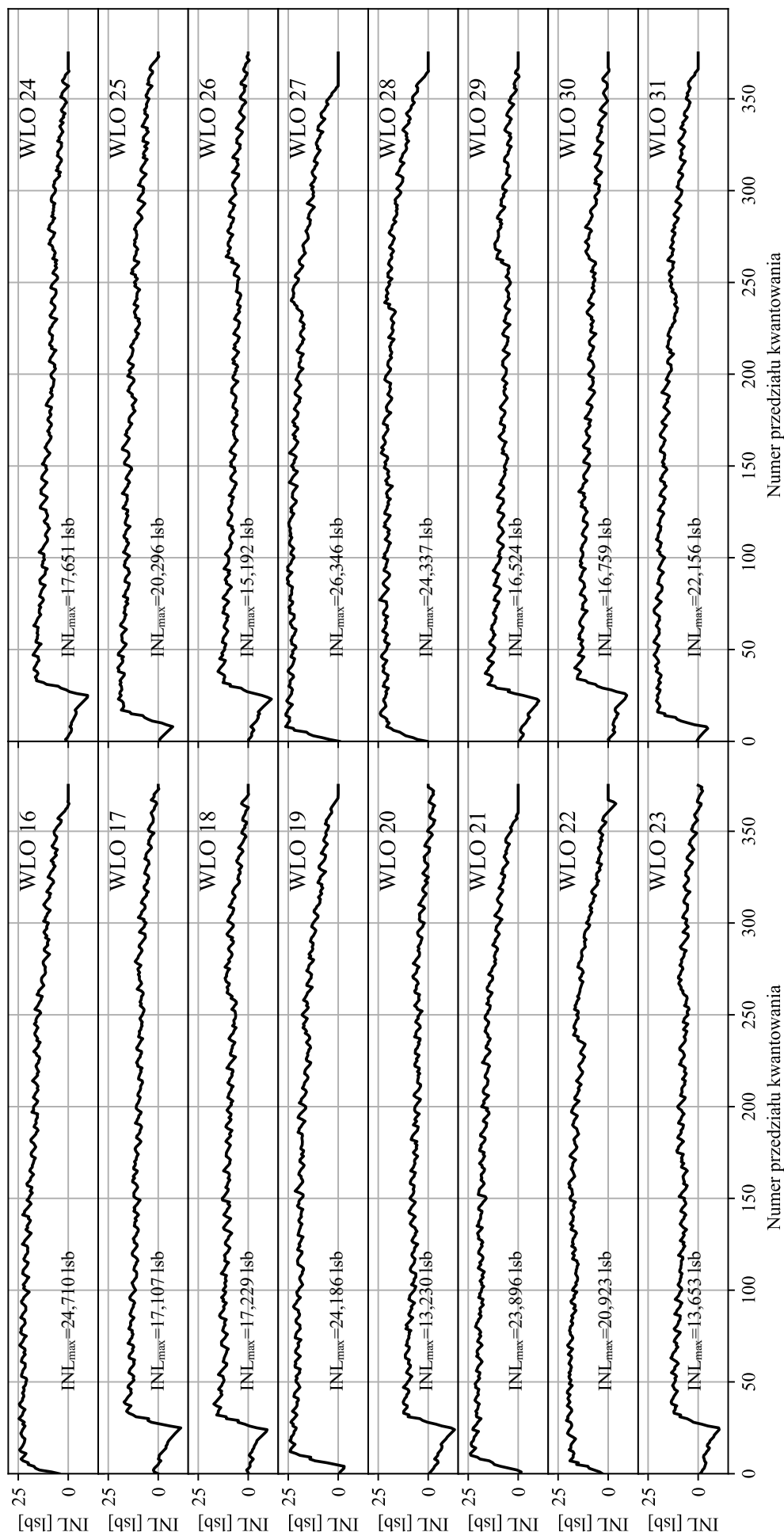
Rys. 68. Charakterystyki DNL[LSB] pierwszych 16 WLO (WLO0 – WLO15) zaimplementowanych w domenie X0Y0 układu Kintex UltraScale po uwzględnieniu przesunięć pomiędzy WLO



Rys. 69. Charakterystyki DNL[LSB] pozostałych 16 WLO (WLO16 – WLO31) zaimplementowanych w domenie X0Y0 układu Kintex UltraScale po uwzględnieniu przesunięcia pomiędzy WLO



Rys. 70. Charakterystyki $INL[LSB]$ pierwszych 16 WLO (WLO0 – WLO15) zaimplementowanych w domenie X0Y0 układu Kintex UltraScale z uwzględnieniem przesunięć pomiędzy WLO



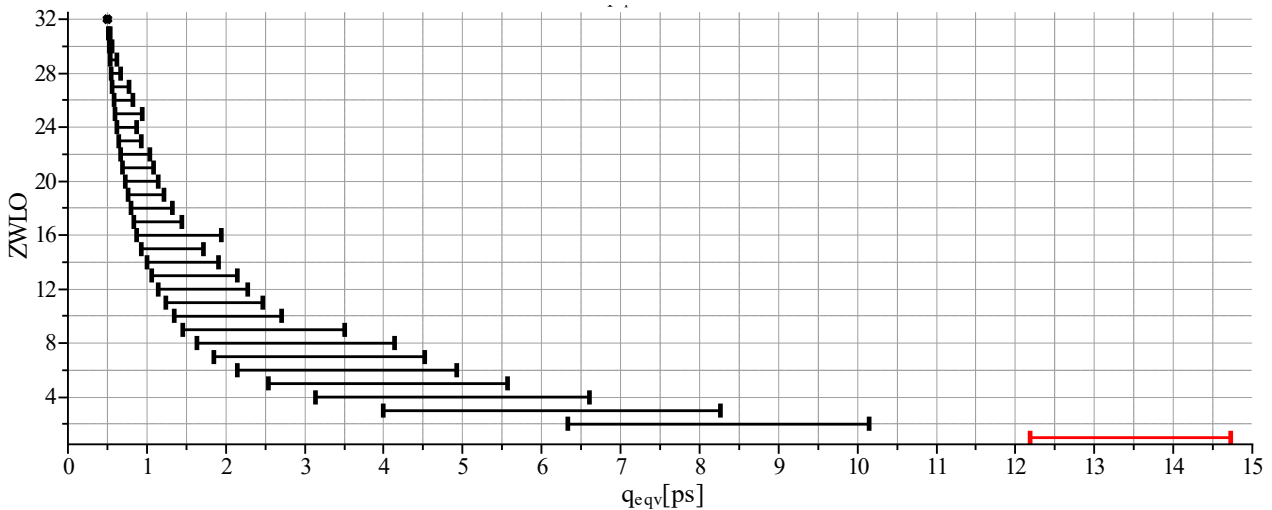
Rys. 71. Charakterystyki INL[LSB] pozostałych 16 WLO (WLO16 – WLO31) zaimplementowanych w domenie XOY0 układu Kintex UltraScale z uwzględnieniem przesunięć pomiędzy WLO

W tabeli 24 zawarto wyniki badań 32 z użyciem metody bezpośredniej rzeczywistych WLO.

Tabela 24. Zakresy wartości q_{eqv} dla różnej liczby WLO wchodzących w skład ZWLO dla 32 WLO badanych metodą bezpośrednią. Obliczenia przeprowadzono na komputerze z procesorem I9-10850K

liczba WLO	$q_{eqv} \min$ [ps]	$q_{eqv} \max$ [ps]	Czas obliczeń [s]	liczba WLO	$q_{eqv} \min$ [ps]	$q_{eqv} \max$ [ps]	Czas obliczeń [s]
1	12,185	14,726	0,001	17	0,837	1,444	147767,833 (41,05 h)
2	6,329	10,152	0,035	18	0,796	1,327	134695,250 (37,42 h)
3	3,990	8,268	0,318	19	0,759	1,214	103719,673 (28,81 h)
4	3,137	6,607	2,542	20	0,725	1,143	72248,148 (20,07 h)
5	2,538	5,574	16,256	21	0,696	1,086	43134,491 (11,98 h)
6	2,143	4,921	87,821	22	0,667	1,034	22891,171 (6,36 h)
7	1,847	4,525	365,658	23	0,644	0,931	10332,310 (2,87 h)
8	1,635	4,137	1329,971 (0,37 h)	24	0,621	0,875	4138,608 (1,15 h)
9	1,460	3,501	3849,558 (1,07 h)	25	0,597	0,937	1354,957 (0,38 h)
10	1,349	2,706	10311,812 (2,86 h)	26	0,579	0,826	384,242
11	1,237	2,466	22293,877 (6,19 h)	27	0,563	0,777	88,878
12	1,139	2,281	43166,76 (11,99 h)	28	0,548	0,667	15,022
13	1,064	2,150	71413,766 (19,84 h)	29	0,535	0,619	2,145
14	0,996	1,913	104167,567 (28,94 h)	30	0,523	0,566	0,225
15	0,934	1,718	129916,741 (36,09 h)	31	0,512	0,540	0,018
16	0,870	1,943	151239,874 (42,01 h)	32	0,502	0,502	0,001

Przedstawiono w niej zakresy wartości $q_{eqv} min$ i $q_{eqv} max$ oraz czasy obliczeń dla wszystkich możliwych kombinacji w zależności od liczby WLO w ZWLO. Zakresy wartości $q_{eqv} min$ oraz $q_{eqv} max$ zostały pokazane na rys. 72.



Rys. 72. Wykres przedstawiający minimalne i maksymalne wartości q_{eqv} dla ZWLO w badanym systemie z 32 WLO. Kolorem czerwonym zaznaczono minimalne i maksymalne wartości q_{eqv} dla pojedynczych WLO

Na osi odciętych naniesiono rozdzielczość ekwiwalentną, a na osi rzędnych znajduje się liczba WLO wchodząca w skład ZWLO. Dla porównania umieszczono również wartości rozdzielczości ekwiwalentnych pojedynczych WLO sprzed operacji scalenia (dla liczby WLO = 1). Końce kresek oznaczają zakresy (wartości minimalne i maksymalne) rozdzielczości ekwiwalentnych dla danej liczby WLO wchodzących w skład ZWLO. Wyniki wskazują, że już samo scalenie dwóch WLO wybranych w sposób przypadkowy (dla badanej kombinacji 32 WLO) zawsze daje istotne polepszenie wartości rozdzielczości ekwiwalentnej. Wartości rozdzielczości ekwiwalentnej q_{eqv} dla pojedynczej WLO znajdują się w przedziale od 12,185 ps do 14,726 ps, a dla ZWLO2 q_{eqv} w zakresie od 6,329 ps do 10,152 ps. Zatem obserwujemy niemal dwukrotną poprawę rozdzielczości przetwornika. Jednak dla ZWLO3 złożonych z 3 WLO wartość rozdzielczości ekwiwalentnej częściowo pokrywa się z wartościami q_{eqv} dla ZWLO2 i wynosi od 3,990 ps do 8,268 ps. W skrajnym przypadku, jeśli WLO wybierane byłyby w sposób przypadkowy, można osiągnąć pogorszenie rozdzielczości ekwiwalentnej w porównaniu do ZWLO2 o najmniejszej wartości q_{eqv} złożonej z 2 WLO. Dla większej liczby WLO wchodzących w skład ZWLO zjawisko pokrywania się wartości q_{eqv} pogłębia się. Wynika z tego, że budując systemy z wieloma WLO należy szczególnie zbadać charakterystyki różnych kombinacji linii, aby nie pogarszać parametrów poprzez przypadkowe dodawanie WLO do ZWLO. Zatem jedynie wyselekcjonowane kombinacje WLO wchodzące w skład ZWLO pozwalają poprawiać parametry wraz z rosnącą liczbą WLO.

Tabela 25. Wartości q_{eqv} wyznaczone metodą doboru do pary spośród 32 badanych WLO. Wartości $q_{eqv} min$ oraz $q_{eqv} max$ wyznaczone zostały metodą bezpośrednią. Obliczenia przeprowadzono na komputerze z procesorem I9-10850K. Wyróżniono parametry dla kombinacji 16 WLO, dla której czas obliczeń dla metody bezpośredniej był najdłuższy i wynosił ok. 42 godziny

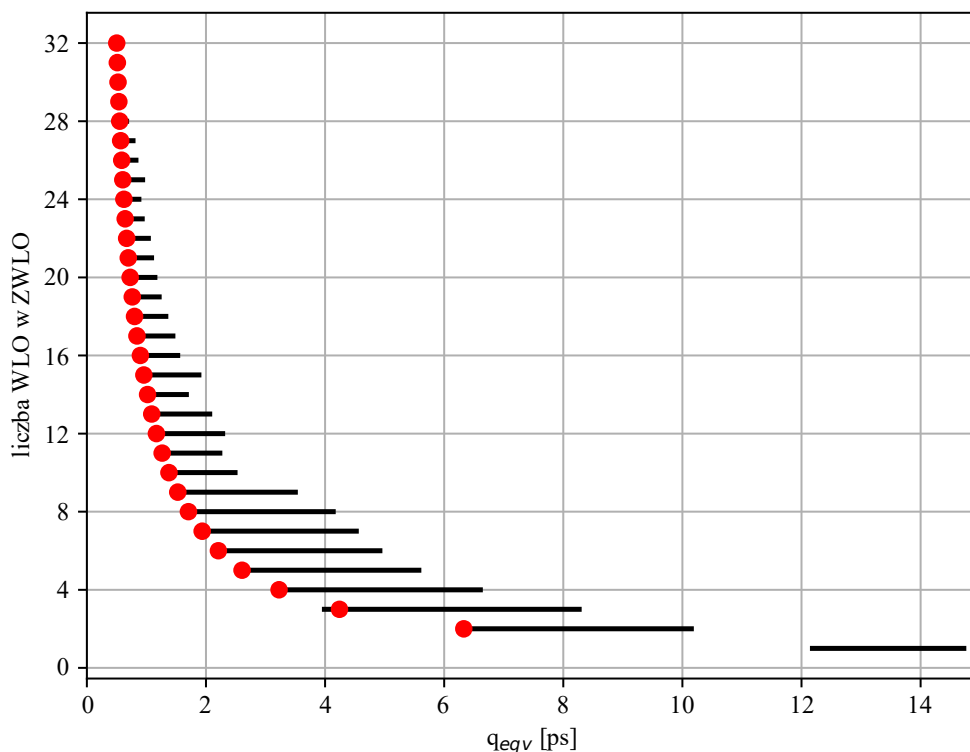
Liczba WLO	q_{eqv} [ps]	$q_{eqv} min$ [ps]	$q_{eqv} max$ [ps]	Czas [s]	Liczba WLO	q_{eqv} [ps]	$q_{eqv} min$ [ps]	$q_{eqv} max$ [ps]	Czas [s]
3	4,243	3,990	8,268	0,028	18	0,801	0,796	1,327	0,087
4	3,227	3,137	6,607	0,029	19	0,763	0,759	1,214	0,091
5	2,606	2,538	5,574	0,031	20	0,728	0,725	1,143	0,097
6	2,209	2,143	4,921	0,036	21	0,696	0,696	1,086	0,102
7	1,937	1,847	4,525	0,038	22	0,669	0,667	1,034	0,119
8	1,706	1,635	4,137	0,043	23	0,644	0,638	1,050	0,111
9	1,526	1,460	3,501	0,046	24	0,623	0,611	1,001	0,114
10	1,379	1,314	3,146	0,051	25	0,603	0,597	0,937	0,118
11	1,265	1,237	2,466	0,054	26	0,585	0,579	0,826	0,122
12	1,169	1,139	2,281	0,059	27	0,569	0,563	0,777	0,124
13	1,089	1,064	2,150	0,064	28	0,552	0,548	0,667	0,130
14	1,020	0,996	1,913	0,068	29	0,538	0,535	0,619	0,131
15	0,957	0,934	1,718	0,073	30	0,524	0,523	0,566	0,133
16	0,899	0,870	1,943	0,078	31	0,512	0,512	0,540	0,134
17	0,842	0,837	1,444	0,082	32	0,502	0,502	0,502	0,136

W metodzie wyboru do pary, czas obliczeń dla dowolnej liczby WLO w ZWLO jest niewspółmiernie mały w stosunku do metody bezpośredniej i dla badanego układu nie przekracza 0,14 s dla ustalonej liczby WLO w ZWLO. Jest to znikomo mały w stosunku do obliczeń trwających nawet 42 godziny dla 16 WLO z badanego zbioru 32 WLO metodą bezpośrednią. Sumaryczny czas obliczeń wszystkich etapów metodą doboru do pary wyniósł 2,5 sekundy w porównaniu do $1,08 \cdot 10^6$ s (12,5 dnia) dla metody bezpośredniej, co daje przyspieszenie obliczeń na poziomie 426000 razy.

Celem porównania wyników q_{eqv} metody przybliżonej, jaką jest metoda doboru do pary, w tabeli 25 zawarto minimalne i maksymalne określone metodą bezpośrednią. Dla zbioru 32 WLO w przypadku przyjęcia kryterium czasu obliczeń, jak i użycia pamięci operacyjnej komputera, metoda ta jest zdecydowanie korzystniejsza w porównaniu do metody bezpośredniej. Jednakże różnice wartości minimalnych rozdzielczości ekwiwalentnych q_{eqv} wyznaczonych metodą doboru do pary są niewielkie i nie przekraczają kilku procent wartości minimalnych q_{eqv} osiągniętych za pomocą metody bezpośredniej dla zadanej liczby WLO w ZWLO. Wyniki analizy w formie graficznej pokazane są na rys 73. Wykres ten pokazuje różnice pomiędzy wartościami wyznaczonymi metodą doboru do pary, a metodą bezpośrednią. Czerwone kropki są wartościami rozdzielczości ekwiwalentnej q_{eqv} określone metodą doboru do pary, zaś czarne linie oznaczają zakresy wartości q_{eqv} metodą bezpo-

średnią. Największa zauważalna różnica wartości rozdzielczości ekwiwalentnej q_{eqv} występuje dla przypadku 3 WLO w ZWLO i wynosi 0,253 ps (6,34%), a w przypadku 16 WLO wynosi 0,029 ps (3,33%).

W tabeli 26 umieszczono wyniki działania algorytmu w czasie wyszukiwania systemu dwukanałowego, który w każdym kanale posiada ZWLO złożone z dwóch WLO. Uzyskane wartości q_{eqv} dla wybranych przez algorytm kombinacji WLO dla wariantów dwóch kanałów niemal pokrywają się (Δq_{eqv} od $2,771 \cdot 10^{-6}$ ps do $2,501 \cdot 10^{-1}$ ps), ale są znacznie większe niż dla przypadku z jednym kanałem pomiarowym. Dla jednego kanału wartość rozdzielczość ekwiwalentnej q_{eqv} wynosi 6,329 ps, a dla wielu kanałów 6,969 ps – 7,126 ps.

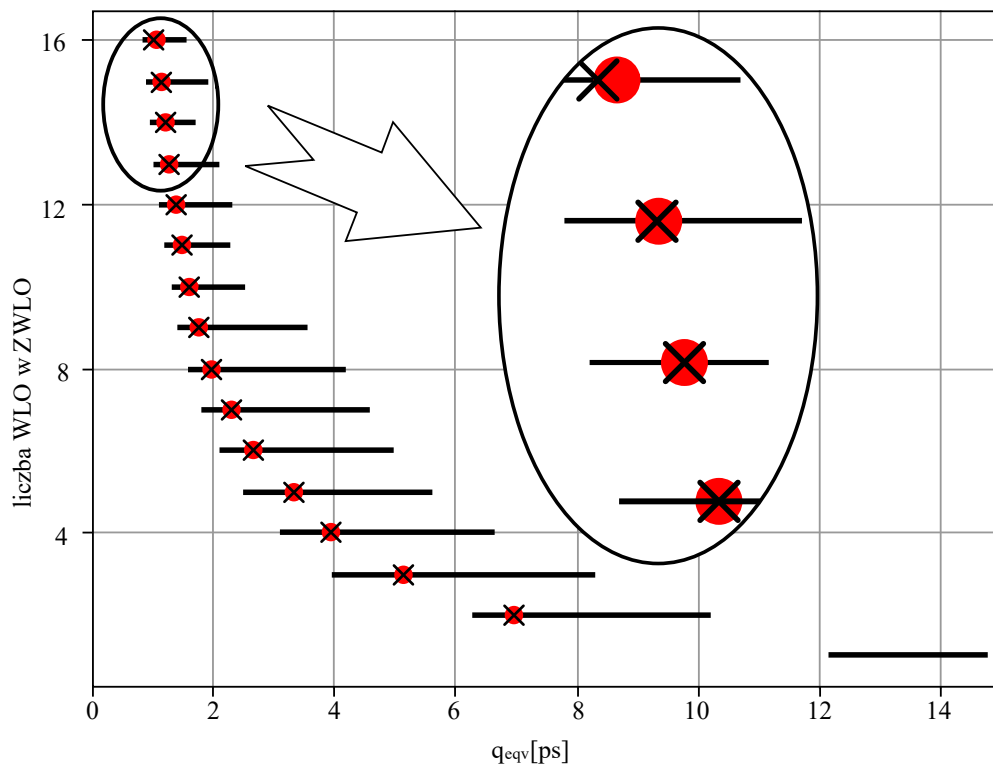


Rys. 73. Rozdzielczość ekwiwalentna q_{eqv} dla ZWLO uzyskana w wyniku działania algorytmu doboru do pary dla systemu 1 kanałowego. Czerwona kropka oznacza wartość q_{eqv} dla najlepszej ZWLO, linia czarna oznacza zakres wartości q_{eqv} dla danej liczby WLO wchodzącej w skład ZWLO (dla 1 WLO – oznacza q_{eqv} WLO)

Wynika to ze sposobu dobierania WLO przez algorytm, który działa tak, aby utworzone docelowo kanały posiadały zbliżone parametry kosztem wartości parametru q_{eqv} . W sytuacji, gdy wyczerpują się wolne WLO w zbiorze W zauważalne są coraz większe różnice q_{eqv} pomiędzy kanałami. I tak dla 3 WLO w ZWLO jest małe i wynosi $0,1132 \cdot 10^{-3}$ ps, dla 15 WLO jest 0,002 ps, ale dla 16 WLO równe jest 0,043 ps co pokazano na powiększeniu na rys. 74.

Tabela 26. Parametry realizacji algorytmu wyszukiwania kanałów ZWLO2 (2 WLO) za pomocą metody bezpośredniej dla systemu wielokanałowego

Liczba kanałów	q_{eqv} średnie [ps]	Δq_{eqv} [ps]	Czas obliczeń [s]	Liczba kanałów	q_{eqv} średnie [ps]	Δq_{eqv} [ps]	Czas obliczeń [s]
1	6,329	-	0,028	9	7,302	$2,038 \cdot 10^{-2}$	0,306
2	6,969	$2,771 \cdot 10^{-6}$	0,050	10	7,301	$2,275 \cdot 10^{-2}$	0,445
3	7,004	$5,400 \cdot 10^{-4}$	0,058	11	7,009	$3,328 \cdot 10^{-2}$	0,478
4	7,005	$1,420 \cdot 10^{-3}$	0,066	12	7,321	$4,720 \cdot 10^{-2}$	0,655
5	7,006	$2,922 \cdot 10^{-3}$	0,076	13	7,321	$6,393 \cdot 10^{-2}$	0,844
6	7,003	$5,149 \cdot 10^{-3}$	0,171	14	7,599	$9,511 \cdot 10^{-2}$	1,013
7	7,004	$6,652 \cdot 10^{-3}$	0,195	15	7,475	$1,535 \cdot 10^{-1}$	1,483
8	7,317	$1,616 \cdot 10^{-2}$	0,295	16	7,126	$2,501 \cdot 10^{-1}$	1,455



Rys. 74. Rozdzielnice ekwiwalentne q_{eqv} dla ZWLO uzyskana w wyniku działania algorytmu doboru do pary dla systemu 2 kanałowego. Czerwone punkty i krzyżyki oznaczają wartości q_{eqv} odpowiednio dla pierwszego i drugiego kanału, linia czarna oznacza zakres wartości q_{eqv} dla danej liczby WLO wchodzącej w skład ZWLO

W tabelach 27-30 przedstawiono wybrane wyniki obliczeń wartości rozdzielnicy ekwiwalentnej q_{eqv} i czasu obliczeń dla algorytmu doboru WLO ze stałą liczbą kanałów pomiarowych przy zadanej liczbie WLO wchodzących w skład ZWLO jednego kanału pomiarowego. W tabelach tych znajdują się kolejno wyniki dla systemu posiadającego odpowiednio 3, 4, 5 oraz 6 WLO w każdym kanale.

Tabela 27. Wyniki doboru linii do ZWLO3 (3 WLO) za pomocą metody bezpośredniej

Liczba kanałów	q_{eqv} średnie [ps]	Δq_{eqv} [ps]	Czas obliczeń [s]	Liczba kanałów	q_{eqv} średnie [ps]	Δq_{eqv} [ps]	Czas obliczeń [s]
1	3,990	-	0,553	6	5,194	$2,396 \cdot 10^{-4}$	5,613
2	5,137	$2,263 \cdot 10^{-7}$	2,347	7	5,194	$5,870 \cdot 10^{-4}$	8,286
3	5,273	$1,901 \cdot 10^{-5}$	2,608	8	5,194	$9,771 \cdot 10^{-4}$	13,961
4	4,869	$5,317 \cdot 10^{-5}$	3,403	9	4,880	$1,872 \cdot 10^{-3}$	31,285
5	5,061	$1,301 \cdot 10^{-4}$	4,305				

Tabela 28. Wyniki doboru linii do ZWLO4 (4 WLO) za pomocą metody bezpośredniej

Liczba kanałów	q_{eqv} średnie [ps]	Δq_{eqv} [ps]	Czas obliczeń [s]	Liczba kanałów	q_{eqv} średnie [ps]	Δq_{eqv} [ps]	Czas obliczeń [s]
1	3,137	-	4,631	5	3,529	$1,143 \cdot 10^{-4}$	20,567
2	3,510	$2,547 \cdot 10^{-8}$	12,458	6	3,322	$2,480 \cdot 10^{-1}$	12,146
3	3,448	$2,048 \cdot 10^{-6}$	13,044	7	3,327	$3,887 \cdot 10^{-1}$	11,431
4	3,480	$3,186 \cdot 10^{-5}$	14,364	8	3,427	$9,960 \cdot 10^{-1}$	11,436

Tabela 29. Wyniki doboru linii do ZWLO5 (5 WLO) za pomocą metody bezpośredniej

Liczba kanałów	q_{eqv} średnie [ps]	Δq_{eqv} [ps]	Czas obliczeń [s]	Liczba kanałów	q_{eqv} średnie [ps]	Δq_{eqv} [ps]	Czas obliczeń [s]
1	2,538	-	15,902	3	2,779	$2,863 \cdot 10^{-6}$	16,811
2	2,746	0,000	17,076	4	2,680	$5,739 \cdot 10^{-4}$	16,429

Tabela 30. Wyniki doboru linii do ZWLO6 (6 WLO) za pomocą metody bezpośredniej

Liczba kanałów	q_{eqv} średnie [ps]	Δq_{eqv} [ps]	Czas obliczeń [s]
1	2,143	-	160,479
2	2,291	0,000	165,382
3	2,205	$7,745 \cdot 10^{-2}$	165,812

Widoczne jest zwiększanie się wartości rozdzielczości ekwiwalentnej q_{eqv} w miarę wykorzystania WLO przy coraz większej liczbie kanałów. Zwiększa się także różnica w wartościach q_{eqv} pomiędzy kanałami. Rośnie także czas obliczeń. Powodem jest coraz większa liczba kombinacji do sprawdzenia. W tabeli 27 zauważalna jest anomalia wartości rozdzielczości ekwiwalentnej średniej q_{eqv} dla systemu 9 kanałowego, która ma wartość 4,880 ps i jest mniejsza niż dla systemu 7 i 8 kanałowego.

Związane jest to z zapewnieniem przez algorytm wyboru jak najmniejszej różnicy w wartościach q_{eqv} dla wszystkich kanałów. W prezentowanym przypadku dla systemu 9 kanałowego Δq_{eqv} wynosi $1,872 \cdot 10^{-3}$ ps i jest większa niż dla systemu 8 kanałowego $9,771 \cdot 10^{-4}$ o 191,6%. Analizując średnie wartości wartości q_{eqv} przykładowo w tabeli 27, zauważyć można, że o ile w systemie uda się wybrać ZWLO zbudowaną z 3 WLO o q_{eqv} poniżej 4 ps (1 kanał), to nie będzie to możliwe dla większej liczby kanałów. Aby zbudować system o większej liczbie kanałów i osiągnąć założone przykładowo $q_{eqv} < 4$ ps na kanał, należy skorzystać z ZWLO4 zbudowanych z 4 WLO.

W tabelach 31–33 przedstawiono wyniki działania algorytmu losowego Monte Carlo dla wariantów: „w górę”, „w dół” oraz „całościowego”.

Tabela 31. Wyniki doboru linii do systemu rekonfigurowalnego z 16 WLO ze zbioru 32 z użyciem metody Monte Carlo „w górę” (korony drzewa), sesja 10^6 iteracji

Kanały/WLO	q_{eqv} średnie [ps]	Δq_{eqv} [ps]	Czas obliczeń [s]
1/16	0,978	-	1162,451
2/8	1,963	0,005	
4/4	3,758	0,127	
8/2	6,959	0,217	

Tabela 32. Wyniki doboru linii do systemu rekonfigurowalnego z 16 WLO ze zbioru 32 z użyciem metody Monte Carlo „w dół” (pnia drzewa), sesja 10^6 iteracji

Kanały/WLO	q_{eqv} średnie [ps]	Δq_{eqv} [ps]	Czas obliczeń [s]
1/16	0,997	-	655,713
2/8	1,902	0,022	
4/4	3,915	0,109	
8/2	7,418	0,248	

Tabela 33. Wyniki doboru linii do systemu rekonfigurowalnego z 16 WLO ze zbioru 32 z użyciem metody Monte Carlo „całościowo”, sesja 10^6 iteracji

Kanały/WLO	q_{eqv} średnie [ps]	Δq_{eqv} [ps]	Czas obliczeń [s]
1/16	1,076	-	338,472
2/8	1,905	0,210	
4/4	3,766	0,095	
8/2	7,763	0,232	

Zawarte w tabelach 31–33 wyniki potwierdzają związek, według którego wartości q_{eqv} są mniejsze wraz z poziomem drzewa, od którego algorytm rozpoczyna wyszukiwanie. Spowodowane jest to tym, że algorytm zaczynając dobór WLO od danego poziomu drzewa może wybrać kombinację WLO o lepszych parametrach metrologicznych i nie jest w tym wyborze ograniczony przez nastę-

ne poziomy. Natomiast przechodząc na następne poziomy algorytm jest już ograniczony poprzednio wybranymi kombinacjami WLO i tylko z tych może dobierać nowe ZWLO. Zachowana jest również zależność czasu wyszukiwania względem benchmarku (w porównaniu z tabelą 6). Najkrócej trwał dobór linii do systemu rekonfigurowalnego w metodzie Monte Carlo „całościowo” (338 s), dłużej w metodzie „w dół” (666 s), a najdłużej w metodzie „w górę” (1162 s). Różnice otrzymanych wartości, wynikają z 10 krotnie większej liczby iteracji w porównaniu do wyników z tabeli 6 (dla 10^5 iteracji metoda „całościowa” 175 s, „w dół” 254 s, „w górę” 280 s). Znaczne różnice w stosunkach tych wartości należy dopatrywać się w tym, że obliczenia na niższych poziomach drzewa trwają dłużej niż na wyższych poziomach (na najniższym poziomie wybranie 16 ZWLO2, obliczenie 16 wartości q_{eqv} , a na najwyższym 2 ZWLO16 i obliczenie dwóch wartości q_{eqv}). Równocześnie algorytm musi spędzić więcej czasu na startowym poziomie, aby uzyskać lepsze wartości q_{eqv} niż dotychczas określone, aby przejść na następny poziom. W przypadku metody całościowej, po wylosowaniu całego drzewa algorytm sprawdza najwyższy poziom, i jeśli nie uzyskuje lepszych parametrów metrologicznych, od razu powtórnie losuje całe drzewo od nowa, bez konieczności doboru WLO i sprawdzania kolejnych poziomów. Równocześnie należy uwzględnić, że metoda Monte Carlo bazuje na losowości wyboru, co może powodować odstępstwa od wyżej wymienionych relacji czasu wyszukiwania.

W celu podsumowania uzyskane parametry systemu jednokanałowego z 16 WLO zawarto w tabeli 34. WLO w systemie zostały wybrane metodą bezpośrednią, co gwarantuje najmniejszą wartość rozdzielczości ekwiwalentnej q_{eqv} , jednak okupione jest to znacznym czasem wyszukiwania (42 h). Żadna inna metoda nie uzyska mniejszej wartości rozdzielczości ekwiwalentnej q_{eqv} .

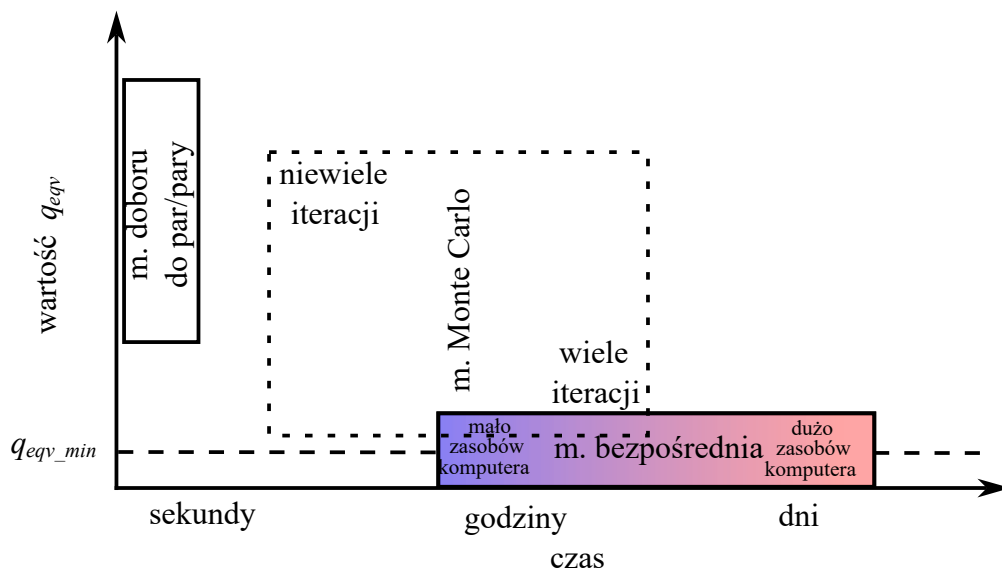
Tabela 34. Parametry systemu jednokanałowego dla kombinacji 16 WLO w ZWLO

Parametr	Minimalne	Maksymalne
\bar{q}	0,359 ps	
q_{eqv}	0,870 ps	1,943 ps
INL_{max}	226,847 ps	316,035 ps
<i>Kombinacja WLO o najmniejszej wartości q_{eqv}</i>	0, 3, 7, 12, 15, 17, 18, 19, 21, 23, 24, 27, 28, 29, 30, 31	
<i>Kombinacja WLO o najmniejszej wartości INL_{max}</i>	2, 3, 4, 5, 7, 10, 13, 17, 18, 20, 23, 25, 26, 29, 30, 31	

Dla systemu wykorzystującego 16 WLO ze zbioru 32 WLO wartość rozdzielczości ekwiwalentnej q_{eqv} mieściła się w przedziale od 0,870 ps (44,8% wartości maksymalnej) do 1,943 ps, a wartość nieliniowości całkowitej INL_{max} zawarta była w zakresie od 226,847 ps do 316,035 ps, dla średniej wartości przedziału kwantowania równej 0,359 ps. W zależności od przyjętego kryterium dla algorytmu, możliwe jest wyszukiwanie kombinacji WLO, aby uzyskać najmniejszą wartość rozdzielczości ekwiwalentnej q_{eqv} , najmniejszą wartość nieliniowości całkowitej INL_{max} lub obu tych wartości jednocześnie. W przypadku systemu jednokanałowego dla kombinacji 16 WLO w ZWLO określono kombinację WLO o następujących numerach : 0, 3, 7, 12, 15, 17, 18, 19, 21, 23, 24, 27, 28, 29, 30,

31 (tabela 34), która ma najmniejszą wartość rozdzielczości ekwiwalentnej. Aby polepszyć liniowość ZWLO dla wybranych WLO z 316,035 ps do 226,847 ps czyli o 28,2% w stosunku do przypadku z największą wartością INL_{max} należy zbudować ZWLO z WLO o następujących numerach: 2, 3, 4, 5, 7, 10, 13, 17, 18, 20, 23, 25, 26, 29, 30, 31.

Na rys. 75 przedstawiono schemat ułatwiający wybór metody doboru w zależności od oczekiwanej wartości rozdzielczości q_{eqv} szukanej kombinacji WLO danej kombinacji, czasu trwania, a także wymaganych zasobów komputera (dla metody bezpośredniej).



Rys. 75. Wykres ułatwiający projektantowi selekcję w zależności od oczekiwanych parametrów, czasu i zasobów komputera

Z przeprowadzonych analiz wynika, że najwyższe rozdzielczości (najmniejsze wartości q_{eqv}) uzyskuje się stosując metodę bezpośrednią doboru linii. Wysokie parametry metrologiczne (q_{eqv} na poziomie pojedynczych pikosekund) są jednakże okupione długimi czasami realizacji algorytmu (dziesiątki godzin).

Jeśli w opracowywanym systemie uzyskanie tak wysokiej rozdzielczości nie jest priorytetowe można zastosować którąś z odmian metody Monte Carlo. Natomiast jeśli decydująca jest duża szybkość realizacji algorytmu, należy użyć metodę doboru do pary.

7. Podsumowanie

W rozprawie opisano opracowany algorytm doboru dyskretnych linii opóźniających w systemie precyzyjnego pomiaru odcinka czasu opartym na użyciu wielokrotnych linii opóźniających. Opisa- no także system pomiarowy zaimplementowany w układzie Kintex UltraScale (AMD/Xilinx) z 16 wielosegmentowymi liniami opóźniającymi. Projekt systemu został utworzony w programie Vivado (Xilinx), a moduły pomiarowe zostały przygotowane w języku VHDL. Badany algorytm doboru wielosegmentowych linii opóźniających został opracowany w języku Python.

Jako oryginalny wkład własny w realizację eksperymentu, będącego podstawą niniejszej rozprawy autor może zaliczyć:

1. Opracowanie w środowisku Vivado (Xilinx) modułu IP Core do pomiaru odcinka czasu z zegarem referencyjnym o częstotliwości taktowania 500 MHz, realizującego pomiary me- todą interpolacyjną z precyzyjnym przetwarzaniem czasowo-cyfrowym realizowanym z użyciem 16 wielosegmentowych linii opóźniających.
2. Implementację kompletnego systemu pomiaru odcinka czasu w układzie Kintex UltraScale (AMD/Xilinx) działającego pod kontrolą wirtualnego mikroprocesora MicroBlaze.
3. Opracowanie i zbadanie metody minimalizacji błędów bąbelkowych w przetwornikach czasowo-cyfrowych opartych na wielosegmentowych liniach opóźniających implementowa- nych w układach FPGA, a także opracowanie metody automatycznego sortowania segmen- tów wielosegmentowej linii opóźniającej. Program sortujący został utworzony w języku Python. Zapewnia wysoką skuteczność poprawy parametrów (przykładowo zmniejszenie wartości rozdzielczości ekwiwalentnej q_{eqv} o 52,6%).
4. Opracowanie i przeprowadzenie kompleksowej analizy metody doboru wielosegmentowych linii opóźniających w systemach pomiaru odcinka czasu w celu osiągnięcia poprawy para- metrów metrologicznych systemu (w szczególności zmniejszenie wartości rozdzielczości ekwiwalentnej) w zależności od wariantów konfiguracji systemów. W systemie jednokana- łowym poprzez dobór WLO metodą bezpośrednią można uzyskać poprawę wartości roz- dzielczości ekwiwalentnej q_{eqv} dla 2 WLO w granicach od 120% do 230% (tabela 24) w sto- sunku do pojedynczej WLO, dla 3 WLO jest to już od 147% do 369% w stosunku do poje- dynczej WLO i 76% do 254% w stosunku do ZWLO2. Dla 16 WLO uzyskuje się poprawę rozdzielczości dla najgorszej kombinacji o 627% , a dla najlepszej kombinacji 1692% w sto- sunku do 1 WLO.
5. Przeprowadzenie pomiarów wydajności zaproponowanego algorytmu w zależności od meto- dy doboru linii, na komputerach o różnych zasobach systemowych oraz zaproponowanie metody wielokrotnie skracającej czas obliczeń dla różnych wariantów doboru linii. Porów- nując różne metody doboru dla systemu 1 kanałowego, metoda doboru do pary (tabela 25) daje gorsze rezultaty od bezpośredniej jedynie o 3,33%, ale jest znacznie szybsza 426000 ra- zy. Metoda Monte Carlo (tabela 15) uzyskała wartość rozdzielczości ekwiwalentnej q_{eqv} gor- szą o 1,7% w stosunku do bezpośredniej, jednak w czasie ponad 500 razy dłuższym od me- tody doboru do pary.

6. Przeprowadzenie badań eksperymentalnych weryfikujących zaproponowane rozwiązanie, a w szczególności identyfikujących poprawę parametrów rzeczywistego układu pomiaru odcinka czasu z 16 wielosegmentowymi liniami opóźniającymi z zastosowaniem różnych metod doboru linii.
7. Zaimplementowanie algorytmu doboru linii w programie i utworzenie graficznego interfejsu użytkownika. Program może być zastosowany w systemach operacyjnych Windows lub Linux.

W systemach pomiaru odcinka czasu implementowanych w układzie FPGA, jedną z metod zwiększenia rozdzielczości jest zwiększenie liczby przetworników czasowo-cyfrowych opartych na wielosegmentowych liniach opóźniających. Charakterystyki tych przetworników są silnie nieliniowe. Opracowany algorytm i przeprowadzone badania wykazują, że w systemach o wielu wielosegmentowych liniach opóźniających nie wystarczy łączyć dowolne wielosegmentowe linie w kanał pomiarowy, lecz należy takie linie opóźniające odpowiednio dobierać w celu istotnego zwiększenia rozdzielczości. Jak wynika z badań rzeczywistego systemu z 16 wielosegmentowymi liniami opóźniającymi możliwe jest uzyskanie rozdzielczości ekwiwalentnej o wartości subpikosekundowej (870 fs) jednakże przypadkowe wybranie linii opóźniających ze zbioru 32 WLO, w najbardziej niekorzystnym przypadku, może spowodować pogorszenie wartości rozdzielczości z 0,870 do 1,943 ps (różnica ponad 200%, tabela 24). Tym samym w systemie z 15 WLO rozpiętość zmian rozdzielczości ekwiwalentnej wynosi od 0,934 ps do 1,718 ps. W bardzo niekorzystnym wariancie, poprzez wybranie systemu o jednej WLO więcej w kanale pomiarowym, można zamiast poprawy uzyskać pogorszenie wartości rozdzielczości ekwiwalentnej. Wiąże się z tym pogorszenie właściwości metrologicznych całego systemu pomiarowego.

Wartość nieliniowości sumacyjnej systemu z 16 wielosegmentowymi liniami opóźniającymi INL_{max} mieściła się w zakresie od 226,847 do 316,035 ps. Dla prezentowanego systemu pomiaru odcinka czasu z 16 WLO, z użyciem opracowanego algorytmu, określono taką kombinację linii, która umożliwia polepszenie liniowości ZWLO dla wybranych WLO nawet o 28,2% w stosunku do najmniej korzystnego przypadku.

Projekt kompletnego systemu pomiarowego, zaimplementowanego w układzie Kintex UltraScale, może być przeniesiony bez zmian do innego układu UltraScale firmy AMD/Xilinx, jeśli zapewnia on wystarczające zasoby logiczne. Sam moduł pomiarowy IP Core może być również zaimplementowany w układach z wbudowanymi mikroprocesorami (np. Zynq), pod warunkiem występowania w nich magistrali AXI. Moduł systemu został utworzony w języku VHDL, co umożliwia względnie łatwą adaptację także do układów FPGA innych producentów.

Przeprowadzona analiza systemów pomiaru odcinka czasu z wielokrotnymi liniami opóźniającymi implementowanymi w układach FPGA oraz analiza charakterystyk wielosegmentowych linii opóźniających, a następnie opracowanie algorytmu doboru wielokrotnych linii opóźniających pozwoliło spełnić postawioną tezę, że: **poprzez zastosowanie odpowiedniego algorytmu wyboru linii opóźniających możliwe jest istotne zwiększenie rozdzielczości i liniowości systemu pomiaru odcinka czasu.** Jak wykazano w zaimplementowanym układzie Kintex UltraScale ze zbioru 32 wielosegmentowych linii opóźniających, poprzez odpowiedni dobór wyselekcjonowanych wielosegmentowych linii opóźniających możliwa jest poprawa rozdzielczości ekwiwalentnej bez potrze-

by zwiększania ilości użytych zasobów sprzętowych. Wysoką skuteczność zaproponowanego algorytmu potwierdzają wyniki przeprowadzonej analizy, a w szczególności wnioski, że aby uzyskać podobną wartość rozdzielczości ekwiwalentnej, bez proponowanego w rozprawie doboru do systemu wielosegmentowych linii opóźniających, system musiałby posiadać w kanale pomiarowym przynajmniej 26 wielosegmentowych linii opóźniających (tabela 24), czyli o 62,5 % więcej niż obecnie.

Literatura

1. E. Pedreschi, B. Angelucci, C. Avanzini, S. Galeotti, G. Lamanna, G. Magazzù, J. Pinzino, R. Piandani, M. Sozzi, F. Spinella, S. Venditti, *A high resolution TDC-based board for a fully digital trigger and data acquisition system in the NA62 experiment at CERN*, IEEE Transactions on Nuclear Science 62(3), 1-6.
2. Q. Yuan, B. Zhang, J. Wu, M. E. Zaghoul, *A high resolution Time-to-Digital Converter on FPGA for Time-Correlated Single Photon Counting*, 2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), 5-8 Aug. 2012, 900-903.
3. G. Acconcia, M. Crotti, S. Antonioli, I. Rech, *High Performance Time-to-Amplitude Converter Array*, 2013 IEEE Nordic-Mediterranean Workshop on Time-to-Digital Converters (NoMe TDC), 3-3 Oct. 2013.
4. W. C. Wiley, I. H. McLaren, *Time-of-Flight Mass Spectrometer with Improved Resolution*, Rev. Sci. Instrum. 26, 1150–1157 (1955).
5. F. Zhang, C. Zhang, X. Jia, Y. Zhang, *Design and Implementation of a Time to Digital Converter Circuit Based on Field-Programmable Gate-Array in Positioning System*, 2014 IEEE International Conference on Electron Devices and Solid-State Circuits, 18-20 June 2014.
6. S. Lin, H. Zhang, J. Wang, D. Yang, K. Cui, J. Wang, *A Generic Reconfigurable System With High-Speed Data Transmission and High-Precision Time Measurements Applied to a Quantum Bit Commitment Experiment*, IEEE Transactions on Nuclear Science, VOL. 62, NO. 3, JUNE 2015, 1027-1032.
7. M. Peca, M. Vacek, V. Michalek, *Time-to-Digit Converter Based on Radiation-Tolerant FPGA*, 2012 European Frequency and Time Forum, 23-27 April 2012, 286-289.
8. L. Mingwe, L. Guosheng, H. Yanguo, *Research on Improving the Accuracy of the Ultrasonic Flow-Meter with Time Difference Method*, 2010 International Conference on Electrical and Control Engineering, 1704-1707.
9. B. K. Swann, B. J. Blalock, L. G. Clonts, D. M. Binkley, J. M. Rochelle, *A 100-ps Time-Resolution CMOS Time-to-Digital Converter for Positron Emission Tomography Imaging Applications*, IEEE Journal of Solid-State Circuits, Vol. 39, NO. 11, November 2004, 1839-1852.
10. K. Morimoto, M. Wu, A. Ardelean, E. Charbon, *Superluminal Motion-Assisted 4-Dimensional Light-in-Flight Imaging*, Phys. Rev. X 11, (011005), 1-7.
11. I. Gyongy, S. W. Hutchings, A. Halimi, M. Tyler, S. Chan, F. Zhu, S. McLaughlin, R. K. Henderson, J. Leach, *High-speed 3D sensing via hybrid-mode imaging and guided upsampling*, Optica, Vol. 7, No. 10, October 2020, 1253-1260.
12. W. Lijuan, Y. Zhendong, *Time-domain Measurements of Frequency Stability*, 2007 8th International Conference on Electronic Measurement and Instruments, 1639-1643.
13. J. Kostamovaara; K. Maatta; T. Rahkonen; R. Rankinen, *ECL and CMOS ASICs for time-to-digital conversion*, Proceedings., Second Annual IEEE ASIC Seminar and Exhibit, 25-28 September 1989, doi: 10.1109/ASIC.1989.123198.
14. J. Kalisz, R. Szplet, J. Pasierbinski, A. Poniecki, *Field-programmable-gate-array-based time-to-digital converter with 200-ps resolution*, 1997, IEEE Trans. on Instr. Vol. 46, pp. 51-55.
15. *UltraScale Architecture Select IO Resources – User Guide UG571 (v1.12)* August 28, 2019, materiały firmy Xilinx.

16. *UltraScale Architecture and Product Data Sheet: Overview* - DS890 (v3.13) July 21, 2020, materiały firmy Xilinx.
17. *Kintex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* - DS892 (v1.18) May 21, 2019, materiały firmy Xilinx.
18. N.Mehta, *Xilinx UltraScale Architecture for High-Performance, Smarter Systems* - WP434 (v1.2) October 29, 2015, materiały firmy Xilinx.
19. N.Mehta, *UltraScale Architecture: Highest Device Utilization, Performance, and Scalability*, WP455 (v1.2) October 29, 2015, materiały firmy Xilinx.
20. R. Frankowski, S. Grzelak, M. Kowalski, D. Chaberski, M. Zieliński, *Time-Interval Measurements, using tapped delay lines implemented in programmable logical Structures*, 12th IMECO TC4, Zagreb, pp. 333 – 336, Croatia 2002.
21. Y. Wang, Q. Cao, C. Liu, *A Multi-chain Merged Tapped Delay Line for High Precision Time-to-Digital Converters in FPGAs*, IEEE Transactions on Circuits and Systems II: Express Briefs, Volume: 65, Issue: 1, January 2018.
22. M. Zieliński, *Układy pomiaru i generacji odcinka czasu*, Wydawnictwo Naukowe Uniwersytetu Mikołaja Kopernika, Toruń 2011.
23. R. Frankowski, M. Kowalski, M. Zieliński, S. Grzelak, *Linie opóźniające implementowane w strukturach programowalnych i ich zastosowanie w systemach pomiarowych*, MWK'2003, T III, str.181-186, 2003.
24. F.A. Hubbard, *System for determining the direction of propagation of wave energy*, 1921, (US 1641432 A).
25. G.L. Tawney, *Electrical time delay line*, 1945, (US 2390563 A).
26. Y. Hagita, K. Ishii, *Circuit technique for improving propagation delay times in CMOS source-coupled logic circuits*, 2012 International Symposium on Intelligent Signal Processing and Communications Systems, Taipei, Taiwan, 4-7 Nov. 2012, pp 615-618.
27. R. D. Lewis, K. W. Awkward, *Methods for determining a maximum operating frequency for TTL gates*, Annual Proceedings on Reliability and Maintainability Symposium, Los Angeles, CA, USA, 1990, pp. 372-377.
28. K. M. Chu, D. L. Pulfrey, *A comparison of CMOS circuit techniques: differential cascode voltage switch logic versus conventional logic*, in IEEE Journal of Solid-State Circuits, vol. 22, no. 4, pp. 528-532, Aug. 1987.
29. A. Sridevi, V. Lakshmiprabha, N. Prabhu, *Design and Analysing the Various Parameters of CMOS Circuit's under Bi-Triggering Method Using Cadence Tools*. Circuits and Systems. January 2016, 07(09) pp. 2622-2632, doi: 10.4236/cs.2016.79227.
30. S. Nikolaidis, *Modeling CMOS Gates Using Equivalent Inverters*, 2015 IEEE 18th International Symposium on Design and Diagnostics of Electronic Circuits & Systems, Belgrade, Serbia, 2015, pp. 119-122, doi: 10.1109/DDECS.2015.20.
31. J. Smyl, J. Zduniak, M. Zieliński, *Wykorzystanie układów programowalnych CPLD w systemach pomiarowych*, KKM 2001, T2 str 159 – 162, Warszawa 2001.
32. D. Chaberski, R. Frankowski, M. Zielinski, L. Zaworski, *Multiple-tapped-delay-line hardware-linearisation technique based on wire load regulation*, 2016, Measurement Vol. 92, pp. 103 – 113.
33. H. Wang, M. Zhang, Q. Yao, *A New Realization of Time-to-Digital Converters Based on FPGA Internal Routing Resources*, IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 60, no. 9, September 2013, 1787-1795.

34. B. I. Abdulrazzaq, I. Abdul Halin, S. Kawahito, R. M. Sidek, S. Shafie N. A. Md. Yunus, *A review on high-resolution CMOS delay lines: towards sub-picosecond jitter performance*, SpringerPlus 5, 434 (2016).
35. M. Zieliński, *Review of single-stage time-interval measurement modules implemented in FPGA devices*, Metrology and Measurement Systems, 2009, Vol. 16, nr 4, pp 641-647.
36. J.-P. Jansson, P. Keränen, S. Jahromi, and J. Kostamovaara, *Enhancing Nutt-Based Time-to-Digital Converter Performance With Internal Systematic Averaging*, IEEE Transactions on Instrumentation and Measurement (Volume: 69 , Issue: 6 , June 2020), pp 3928 – 3935.
37. D. Chaberski, *System pomiaru funkcji intensywności strumienia jonów w spektrometrii masowej czasu przelotu*, praca doktorska (promotor M. Zieliński), WAT, Warszawa, 2006.
38. M. Kowalski, *Wpływ fluktuacji fazowych generatora wzorcowego na dokładność pomiaru odcinka czasu*, praca doktorska (promotor M. Zieliński), Akademia Morska Gdynia, 2009.
39. Xilinx. *Virtex - 2.5 V Field Programmable Gate Arrays Product Specification*, Technical Documentation, 2001.
40. R. Frankowski, *Wielosegmentowe linie opóźniające i ich zastosowanie w precyzyjnych systemach pomiarowo kontrolnych*, praca doktorska (promotor M. Zieliński), PW, Warszawa, 2010.
41. J. Kalisz, R. Szplet, R. Pelka, A. Ponięcki, *Single-chip interpolating time counter with 200-ps resolution and 43-s range*, 1996, Vol. 1 Quality Measurement: The Indispensable Bridge between Theory and Reality (No Measurements? No Science! Joint Conference - 1996: IEEE Instrumentation and Measurement Technology Conference and IMEKO Tec, pp. 676-680 Vol.1.
42. T. Rahkonen, J. Kostamovaara, *The use of stabilized CMOS delay lines in the digitization of short time intervals*, Proc. IEEE Symp. Circuits Syst., Singapore, Vol. 4, pp. (2252 - 2255), 1991.
43. R. Szplet, Z. Jachna, P. Kwiatkowski, K. Różyk, *A 2.9 ps equivalent resolution interpolating time counter based on multiple independent coding lines*, 2013, Measurement Science and Technology (24), pp. 1-15.
44. S. H. Neddermeyer, E. J. Althaus, W. Allison, E. R. Schatz, *The Measurement of Ultra-Short Time Intervals*, The Review Of Scientific Instruments Vol. 18, Number 7, July 1947.
45. P. J. Kindlmann, J. Sunderland, *Phase Stabilized Vernier Chronotron*, 1966, The Review of Scientific Instruments Vol. 37(4), pp. 445-452.
46. H. W. Lefevre, J.T. Russell, *Vernier Chronotron*, 1959, The Review of Scientific Instruments Vol. 30(3), pp. 159-166.
47. L. Vercesi, A. Liscidini, R. Castello, *Two-Dimensions Vernier Time-to-Digital Converter*, IEEE Journal of Solid-State Circuits Vol. 45 , Issue: 8 , Aug. 2010.
48. S. Magadum, M. P. Mattada, H. Guhilot, *Multiphase clock based vernier TDC on FPGA for on-chip temperature measurement application*, 2017 2nd International Conference for Convergence in Technology (I2CT), pp 828-831.
49. M.-C. Lin, G.-R. Tsai, C.-Y. Liu, S.-S. Chu, *FPGA-Based High Area Efficient Time-To-Digital IP Design*, TENCON 2006 - 2006 IEEE Region 10 Conference, Hong Kong, China, pp 1-4.
50. J. Mauricio, L. Freixas, A. Sanuy, S. Gómez, R. Manera, J. Marín, J. M. Pérez, E. Picatoste, P. Rato, D. Sánchez, A. Sanmukh, O. Vela, D. Gascon, *MATRIX16: A 16-Channel Low-Power TDC ASIC with 8 ps Time Resolution*, Electronics 2021, 10, 1816, doi :10.3390/electronics10151816.
51. D. J. Kinniment, O.V. Maevsky, A. Bystrov, G. Russell, A.V. Yakovlev, *On-chip structures for timing measurement and test*, Proc. 8th International Symposium on Asynchronous Circuits and Systems ASYNC 2002, Manchester, 190-197, 8-11.04.2002.

52. A. Mäntyniemi, T. Rahkonen, J. Kostamovaara, *A CMOS time-to-digital converter (TDC) based on a cyclic time domain successive approximation interpolation method*, IEEE Journal of Solid-State Circuits, vol. 44, no. 11, 3067-3078, 2009.
53. T. Rahkonen, J. Kostamovaara, *Pulsewidth measurements using an integrated pulse shrinking delay line*, Proc. IEEE Int. Symp. Circuits and Systems, vol. 1, 1990.
54. E. Raisanen-Ruotsalainen, T. Rahkonen, J. Kostamovaara, *A Low-Power CMOS Time-to-Digital Converter*, IEEE Journal of Solid-State Circuits, Vol. 30, No. 9, 1995, 984 – 990.
55. T.E. Rahkonen; J.T. Kostamovaara, *The use of stabilized CMOS delay lines for the digitization of short time intervals*, IEEE Journal of Solid-State Circuits, Volume: 28, Issue: 8, Aug 1993, 887-894.
56. P. Chen, S. Liu, J. Wu, *Highly accurate cyclic CMOS time-to-digital converter with extremely low power consumption*, Electronics Letters, Vol. 33, No.10, 1997, 858 – 860.
57. R. Szplet, K. Klepacki, *A two-stage time-to-digital converter based on cyclic pulse shrinking*, 2009, 2009 IEEE International Frequency Control Symposium Joint with the 22nd European Frequency and Time forum, pp. 1133-1136.
58. R. Szplet, K. Klepacki, *An FPGA-Integrated Time-to-Digital Converter Based on Two-Stage Pulse Shrinking*, 2010, IEEE Transactions on Instrumentation and Measurement Vol. 59(6), pp. 1663-1670.
59. J. Wu, Z. Shi, *The 10-ps wave union TDC: Improving FPGA TDC resolution beyond its cell delay*, IEEE Nuclear Science Symposium Conference Record NSS/MIC 2008, Drezno, 19-25.10.2008, 3440-3446.
60. R. Szplet, D. Sondej, G. Grzęda, *Interpolating time counter with multi-edge coding*, 2013, 2013 Joint European Frequency and Time Forum International Frequency Control Symposium (EFTF/IFC), pp. 321-324
61. D. Sondej, R. Szymanowski, R. Szplet, *Methods of precise determining the transfer function of picosecond time-to-digital converters*, Metrol. Meas. Syst., Vol. 28 (2021) No. 3, 539–549.
62. W. Xie, H. Chen, D. D. Li, *Efficient Time-to-Digital Converters in 20 nm FPGAs With Wave Union Methods*, IEEE Transactions on Industrial Electronics, VOL. 69, NO. 1, JANUARY 2022, 1021-1031.
63. R. Szplet, D. Sondej, G. Grzęda, *High-Precision Time Digitizer Based on Multiedge Coding in Independent Coding Lines*, 2016, IEEE Transactions on Instrumentation and Measurement Vol. 65(8), pp. 1884-1894.
64. J. Wu, *Several Key Issues on Implementing Delay Line Based TDCs Using FPGAs*, IEEE Transactions on Nuclear Science, VOL. 57, NO. 3, JUNE 2010, 1543-1548.
65. T. Sui, Z. Zhao, S. Xie, Y. Xie, Y. Zhao, Q. Huang, J. Xu, Q. Peng, *A 2.3-ps RMS Resolution Time-to-Digital Converter Implemented in a Low-Cost Cyclone V FPGA*, IEEE Transactions on Instrumentation and Measurement, Vol. 68, No. 10, October 2019, 3647-3660.
66. R. Szplet, J. Kalisz, R. Szymanowski, *Interpolating Time Counter with 100 ps Resolution on a Single FPGA Device*, 2000, IEEE Trans. Instr. & Meas. Vol. 49(4), pp. 879-883.
67. R. Nutt, *Digital Time Intervalometer*, 1968, Review of Scientific Instrument Vol. 39, pp. 1342-1345.
68. B. Turko, *A Picosecond Resolution Time Digitizer for Laser Ranging*, 1978, IEEE Transactions on Nuclear Science Vol. NS-25(1), pp. 75-80.

69. B. Turko, *Multichannel Interval Timer*, 1984, IEEE Transactions on Nuclear Science Vol. NS-31(1), pp. 167-171.
70. D. I. Porat, *Review of subnanosecond time-interval measurements*, IEEE Transactions on Nuclear Science, vol. 20, no. 5, 1973, 36-51.
71. K. Koch, H. Hardel, R. Schulze, E. Badura, *A New TAC-Based Multichannel Front-End Electronics for TOF Experiments With Very High Time Resolution*, IEEE Transactions on Nuclear Science, Vol. 52, No. 3, June 2005, 745-747.
72. R. Szplet, *Time-to-Digital Converters*. In: P. Carbone, S. Kiaei, F. Xu (eds), *Design, Modeling and Testing of Data Converters*, Signals and Communication Technology, Springer, Berlin, Heidelberg, 2013, doi: 10.1007/978-3-642-39655-7_7.
73. J. Jansson, A. Mäntyniemi, J. Kostamovaara, *Synchronization in a Multilevel CMOS Time-to-Digital Converter*, IEEE Transactions on Circuits and Systems Part I: Regular Papers, Vol. 56, No. 8, AUGUST 2009, 1622-1634.
74. *KCU105 Board User Guide - UG917 (v1.10)* February 6, 2019, materiały firmy Xilinx.
75. R. Seelam, *I/O Design Flexibility with the FPGA Mezzanine Card (FMC)*, materiały firmy Xilinx.
76. *Vivado Design Suite Tutorial Implementation - UG986 (v2020.1)* August 12, 2020, materiały firmy Xilinx.
77. *MicroBlaze Processor Reference Guide - UG984 (v2020.1)* June 3, 2020 materiały firmy Xilinx
78. *IEEE Standard VHDL Language Reference Manual*, 2000.
79. *UltraScale Architecture Libraries Guide - UG974 (v2020.1)* June 3, 2020, materiały firmy Xilinx.
80. Y. Wang and C. Liu, *A 3.9 ps time-interval RMS precision Time-to-Digital converter using a dual-sampling method in an ultrascale FPGA*, IEEE Trans. Nucl. Sci., Vol. 63, No. 5, Oct. 2016, 2617–2621.
81. *Vivado Design Suite Tcl Command Reference Guide*, UG835 (v2021.2) October 22, 2021, materiały firmy Xilinx.
82. M. Rahman, K. L. Baishnab, F. A. Talukdar, *A novel ROM architecture for reducing bubble and metastability errors in high speed flash ADCs*, 2010 20th International Conference on Electronics Communications and Computers (CONIELECOMP), Cholula, 2010, pp. 15-19.
83. R. Szymanowski, R. Szplet, P. Kwiatkowski, *Quantization error in precision time counters*, 2015, Measurement Science and Technology Vol. 26(7), pp. art. nr 075002.
84. Y.-H. Chen, *A high resolution FPGA-based merged delay line TDC with nonlinearity calibration*, 2013 IEEE International Symposium on Circuits and Systems (ISCAS), Beijing, China, 2013, pp. 2432-2435, doi: 10.1109/ISCAS.2013.6572370.
85. P. Carra, M. Bertazzoni, M. G. Bisogni, A. Del Guerra, M. Morrocchi, G. Pazzi, G. Sportelli, N. Belcari, *Auto-Calibrating TDC for a SoC-FPGA Data Acquisition System*, 2017 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), Atlanta, GA, USA, 2017, pp. 1-2, doi: 10.1109/NSSMIC.2017.8533060.
86. N. Metropolis, S. Ulam, *The Monte Carlo Method*, Journal of the American Statistical Association. 44 (247), 1949, 335–341.
87. A. Kumar, S. P. Panda, *A Survey: How Python Pitches in IT-World*, 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 2019, pp. 248-251, doi: 10.1109/COMITCon.2019.8862251.

88. Oficjalna dokumentacja języka Pythona 3. Dostęp World Wide Web : <https://docs.python.org/>