

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki
Podstawy Teletransmisji

Koder
Ethernet IEEE 802.3 100BASE-TX

Bartosz Siwiaszczyk
Piotr Kupczyk
Aleksander Sawka

Rok akademicki 2023/2024
Kierunek Teleinformatyka
Semestr 4
Seksja 9

Spis treści

1	Wstęp	3
1.1	Opis tematyki oraz celu projektu	3
1.2	Wybór elementu systemu teletransmisyjnego	3
1.3	Zawężenie symulacji	3
2	Założenia projektowe	4
2.1	Sposób kodowania	4
2.2	Parametry	4
2.2.1	Parametry fizyczne	4
2.2.2	Parametry czasowe	4
2.3	Szybkość transmisji	4
3	Realizacja i implementacja projektu	5
3.1	Opis metody badawczej	5
3.2	Narzędzia i techniki badawcze	5
4	Realizacja schematu kodera przy użyciu elementów idealnych	6
4.1	Wstęp	6
4.2	Analiza logiczna	6
4.2.1	Tablica prawdy dla kodu 4B5B	6
4.2.2	Siatki Karnaugh	7
4.2.3	Uproszczony schemat kodera firmy Actel	8
4.2.4	Opis kodera 4B5B przy użyciu PALASM2	8
4.3	Realizacja przy użyciu środowiska LTSpice	9
4.3.1	Wstęp	9
4.3.2	Realizacja schematu kodera w symulatorze LTSpice	9
4.3.3	Koder bitu Q1	10
4.3.4	Koder bitu Q2	10
4.3.5	Koder bitu Q3	11
4.3.6	Koder bitu Q4	11
4.4	Symulacja działania kodera	12
4.4.1	Wstęp	12
4.4.2	Zadany sygnał wejściowy	12
4.4.3	Sygnał wyjściowy	13
4.5	Analiza uzyskanych wyników	14
4.5.1	Wstęp	14
4.5.2	Wnioski	14
5	Realizacja schematu kodera przy użyciu elementów rzeczywistych	15
5.1	Wstęp	15
5.2	Skrócona specyfikacja techniczna elementów	15
5.3	Realizacja w środowisku LTSpice	16

5.3.1	Wstęp	16
5.3.2	Realizacja koderów	16
5.3.3	Koder bitu Q1	17
5.3.4	Koder bitu Q2	17
5.3.5	Koder bitu Q3	18
5.3.6	Koder bitu Q4	18
5.4	Symulacja działania	19
5.4.1	Wstęp	19
5.4.2	Zadany sygnał wejściowy	19
5.4.3	Sygnał wyjściowy	20
5.5	Analiza uzyskanych wyników	21
5.5.1	Wstęp	21
5.5.2	Parametry układu	21
5.5.3	Wnioski	21
5.5.4	Artefakty sygnału	22
6	Podsumowanie	23
6.1	Elementy idealne	23
6.2	Elementy rzeczywiste	23
6.3	Całokształt	23
7	Wnioski końcowe	24

1 Wstęp

1.1 Opis tematyki oraz celu projektu

Projekt dotyczy kompleksowej analizy wybranego elementu systemu teletransmisyjnego w standardzie IEEE 802.3 100BASE-TX [1], który ma kluczowe znaczenie dla efektywności transmisji danych w sieciach lokalnych. Celem projektu jest nie tylko dokładne zrozumienie i opisanie mechanizmów działania wybranego komponentu, ale także przeprowadzenie serii testów oraz eksperymentów, które pozwolą ocenić jego wydajność i niezawodność w rzeczywistych warunkach eksploatacji. Przewiduje się, że wyniki tych badań pozwolą na głębsze zrozumienie procesów teletransmisyjnych i będą mogły posłużyć do dalszego rozwoju technologii sieciowych. [4].

1.2 Wybór elementu systemu teletransmisyjnego

Dla celów naszego projektu wybrano koder w standardzie IEEE 802.3 100BASE-TX [1]. Urządzenie to odgrywa fundamentalną rolę w przetwarzaniu i przygotowywaniu danych do transmisji poprzez medium transmisyjne, jakim jest kabel miedziany. Koder wykorzystuje zaawansowane techniki kodowania, takie jak NRZ (Non-Return to Zero) i modulacja 4B5B [3], które są kluczowe dla efektywnego i bezbłędnego przesyłania danych. Ponadto, integralnymi elementami systemu są również konwersja NRZI (Non-Return to Zero Inverted), która zwiększa odporność sygnału na zakłócenia, oraz scrambler, służący do randomizacji przesyłanego strumienia danych w celu zredukowania problemów związanych z długotrwałymi pasmami jednolitych bitów i minimalizacji skutków interferencji [5].

1.3 Zawężenie symulacji

Symulacja koncentruje się na technice kodowania 4B5B, co pozwoli na szczegółową analizę tego mechanizmu kodowania oraz jego wpływu na ogólną wydajność systemu transmisyjnego [5].

2 Założenia projektowe

2.1 Sposób kodowania

- **4B5B:** Każde 4 bity danych są przekształcane na 5-bitowy symbol, co pozwala na zredukowanie ciągów długich zer i jedynek, co jest kluczowe dla synchronizacji i efektywności transmisji. Stosuje się 32 symbole kodowe dla 16 możliwych 4-bitowych kombinacji, a także kody kontrolne i specjalne [3].

2.2 Parametry

2.2.1 Parametry fizyczne

- **Typ kabla:** UTP kategoria 5
- **Impedancja:** 100 Ohmów [1]

2.2.2 Parametry czasowe

- **Czas bitowy:** 16 ns na bit
- **Czas symbolu:** 8 ns na symbol 5-bitowy 4B5B
- **Maksymalna długość ciągu bitów o tej samej wartości:** ograniczona do 5, co ułatwia utrzymanie synchronizacji zegarów
- **Minimalny odstęp czasowy pomiędzy ramkami:** 9.6 mikrosekund [1]

2.3 Szybkość transmisji

- **Nominalna szybkość transmisji:** 100 Mbps
- **Częstotliwość sygnału:** Zakres częstotliwości używany w transmisji wynosi od 0 do 125 MHz [1]

3 Realizacja i implementacja projektu

3.1 Opis metody badawczej

Podczas realizacji projektu kodera 4B5B, kluczowym elementem był dostęp do tablicy prawdy kodera, udostępnionej przez firmę Actel w ich nocie aplikacyjnej [5]. Na tej podstawie, zastosowanie metody siatek Karnaugh umożliwiło efektywne rozpisanie równań logicznych, które stanowią podstawę działania kodera.

Wykorzystując te równania, opracowano układ logiczny zdolny do kodowania 4-bitowych słów danych w 5-bitowe słowa kodowe. Projektowanie tego układu rozpoczęto od analizy równań uzyskanych za pomocą metody Karnaugh, co pozwoliło na optymalizację liczby bramek logicznych oraz ich układu.

Finalny projekt układu logicznego zrealizowano po weryfikacji poprawności działania kodera przed jego fizyczną implementacją. Dzięki temu procesowi, możliwe było dokładne dostosowanie parametrów układu do specyfikacji wymaganych zgodnie ze standardem IEEE 802.3 [1].

3.2 Narzędzia i techniki badawcze

W ramach naszego projektu do badania oraz optymalizacji działania kodera 4B5B zastosowano różnorodne narzędzia co pozwoliło na kompleksowe podejście do jego analizy. Szczególnie użyteczne okazały się:

- **Symulacje komputerowe:** Użyto oprogramowania LTSpice, które jest zaawansowanym narzędziem do symulacji układów elektronicznych, pozwalającym na szczegółową analizę działania projektowanego kodera w różnorodnych warunkach. LTSpice umożliwił precyzyjne modelowanie zarówno analogowych i cyfrowych składników naszego kodera, oferując możliwość przetestowania różnych scenariuszy. Symulacje pozwoliły na identyfikację i rozwiązanie potencjalnych problemów w działaniu układu.
- **Porównanie wyników z referencyjną implementacją:** Aby zapewnić, że nasz koder spełnia wymagane standardy, przeprowadzono serię porównań jego działania z wynikami uzyskanymi z referencyjnych implementacji kodera 4B5B. Dokładne porównanie parametrów takich jak szybkość przesyłu danych, skuteczność kodowania, a także odporność na błędy pozwoliło na zweryfikowanie poprawności projektu.

4 Realizacja schematu kodera przy użyciu elementów idealnych

4.1 Wstęp

- Zastosowanie metody siatek Karnaugh [rys. 2] umożliwiło efektywne zbudowanie układu logicznego opartego na wcześniej wydedukowanych równaniach logicznych. W fazie implementacji w symulatorze LTSpice napotkano jednak na wyzwania dotyczące właściwego sposobu implementacji mechanizmu ładowania strumienia danych binarnych do symulowanego układu. Po rozważeniu różnych metod, ostatecznie zdecydowano się na wykorzystanie źródeł wymuszeń, których parametry określono w pliku tekstowym. To podejście pozwoliło na precyzyjne kontrolowanie danych wejściowych oraz ich synchronizację z działaniem układu.
- Po starannym doborze parametrów, danych oraz odpowiednich komponentów, skonstruowany w symulatorze układ generuje wyniki zgodne z oczekiwaniami. Wykorzystanie ogólnodostępnych zasobów oraz aplikowanie operacji logicznych pozwoliło na uproszczenie architektury kodera do bardziej modułowej formy, dzięki czemu poszczególne bloki funkcjonalne były łatwiejsze w analizie i modyfikacji.
- W rezultacie, zrealizowany układ skutecznie konwertuje 4-bity informacji na 5-bitowe paczki danych, co jest zgodne z założeniami przedstawionymi w tabeli prawdy [rys. 1].

4.2 Analiza logiczna

4.2.1 Tablica prawdy dla kodu 4B5B

Symbol	5-bit Code Group (in Convergence sublayer)	4-bit Binary Code Group (in MAC)	Interpretation/Function
0	11110	0000	Data character: 0H
1	01001	0001	Data character: 1H
2	10100	0010	Data character: 2H
3	10101	0011	Data character: 3H
4	01010	0100	Data character: 4H
5	01011	0101	Data character: 5H
6	01110	0110	Data character: 6H
7	01111	0111	Data character: 7H
8	10010	1000	Data character: 8H
9	10011	1001	Data character: 9H
A	10110	1010	Data character: AH
B	10111	1011	Data character: BH
C	11010	1100	Data character: CH
D	11011	1101	Data character: DH
E	11100	1110	Data character: EH
F	11101	1111	Data character: FH
I	11111	--	Idle character transmitted between packets
J	11000	--	First control character in start-of-packet delimiter
K	10001	--	Second control character in start-of-packet delimiter
T	01101	--	First control character in end-of-packet delimiter
R	00111	--	Second control character in end-of-packet delimiter
V	00000	--	Invalid character
V	00001	--	Invalid character
V	00010	--	Invalid character
V	00011	--	Invalid character
V	00100	--	Invalid character
V	00101	--	Invalid character
V	00110	--	Invalid character
V	01000	--	Invalid character
V	01100	--	Invalid character
V	10000	--	Invalid character
V	11001	--	Invalid character

Rysunek 1: Kodowane symbole kodu 4B5B

4.2.2 Siatki Karnaugh

Q1 \ Q0	00	01	11	10
Q4 \ Q3				
00	11110	01001	10101	10100
01	01010	01011	01111	01110
11	11010	11011	11101	11100
10	10010	10011	10111	10110

$$Q4 = D2' * D0' + D2' * D1 + D3$$

D1 \ D0	00	01	11	10
D4 \ D3				
00	1	0	1	1
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

$$Q3 = D3' * D1' + D2$$

D1 \ D0	00	01	11	10
D4 \ D3				
00	1	1	0	0
01	1	1	1	1
11	1	1	1	1
10	0	0	0	0

$$Q2 = D3' * D2' * D0' + D1$$

D1 \ D0	00	01	11	10
D4 \ D3				
00	1	0	1	1
01	0	0	1	1
11	0	0	1	1
10	0	0	1	1

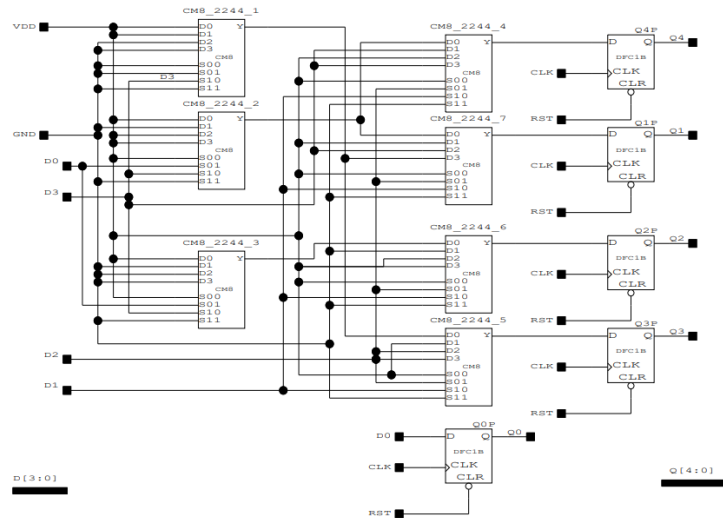
$$Q1 = D1' * D0' + D3' * D2 + D3 * D2' + D2 * D1'$$

D1 \ D0	00	01	11	10
D4 \ D3				
00	1	0	0	01
01	1	1	1	1
11	1	1	0	0
10	1	1	1	1

$$Q0 = D1$$

Rysunek 2: Tablica prawdy przetłumaczona na siatki Karnaugh

4.2.3 Uproszczony schemat kodera firmy Actel



Rysunek 3: Uproszczony schemat kodera przy użyciu widoku logiki FPGA

4.2.4 Opis kodera 4B5B przy użyciu PALASM2

```
;Encoder for 4B to 5B
;Used in 100 Mbit Ethernet application
CHIP 4b5b generic
clk rst d3 d2 d1 d0 q4 q3 q2 q1 q0

EQUATIONS

q4 := d3 + (/d2 * d1) + (/d2 * /d0)
q3 := d2 + (/d3 * /d1)
q2 := d1 + (/d3 * /d2 * /d0)
q1 := (/d1 * /d0) + (d3 :+: d2) + (d2 * /d1)
q0 := d0

q4.clkf = clk
q3.clkf = clk
q2.clkf = clk
q1.clkf = clk
q0.clkf = clk

q4.rstf = /rst
q3.rstf = /rst
q2.rstf = /rst
q1.rstf = /rst
q0.rstf = /rst
```

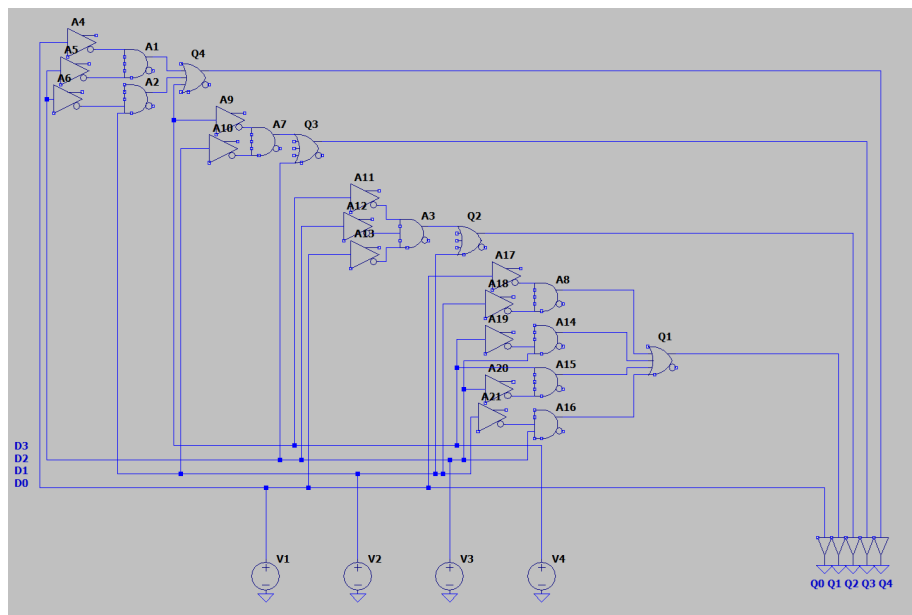
Rysunek 4: Schemat kodera 4B5B w języku PALASM2

4.3 Realizacja przy użyciu środowiska LTSpice

4.3.1 Wstęp

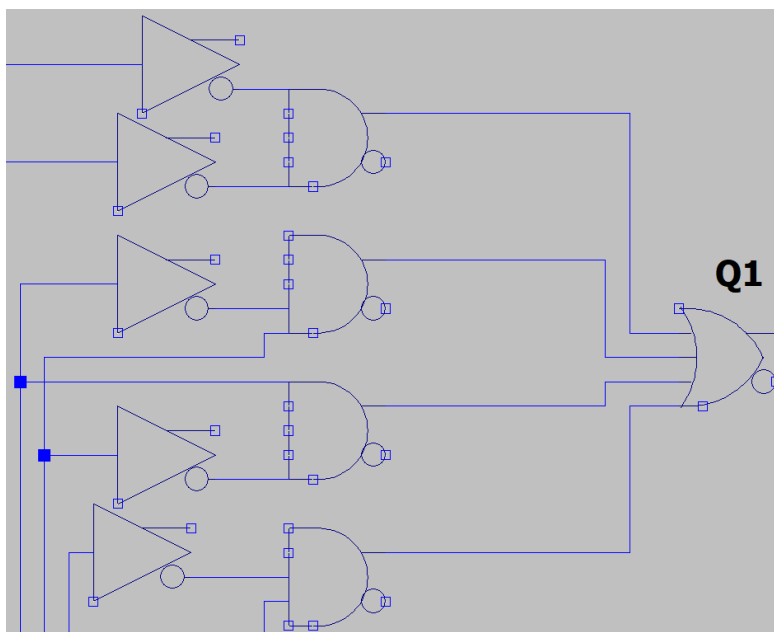
- Na podstawie równań logicznych, uzyskanych dzięki zastosowaniu metody siatek Karnaugh, zbudowano układ składający się z czterech modułów kodujących, oznaczonych jako Q1, Q2, Q3 i Q4 [rys. 5]. Każdy z tych modułów odpowiedzialny jest za kodowanie poszczególnych symboli wyjściowych na podstawie bitów wejściowych [rys. 6-9].
- Specyfikacja bitu Q0 jest bezpośrednio powiązana z bitem wejściowym D1, co wynika z zasady działania kodu 4B5B, gdzie bit Q0 replikuje wartość bitu D1. Taka zależność ma kluczowe znaczenie dla zachowania odpowiedniej struktury danych wyjściowych, która jest wymagana przez standard kodowania.
- Dodatkowo, implementacja układu uwzględnia cztery źródła napięcia, oznaczone jako V1, V2, V3 oraz V4. Każde z tych źródeł napięcia zostało szczegółowo zdefiniowane i opisane, a ich role polegają na dostarczeniu odpowiedniego strumienia bitowego, który jest niezbędny dla poprawnego działania poszczególnych bloków kodujących.

4.3.2 Realizacja schematu kodera w symulatorze LTSpice



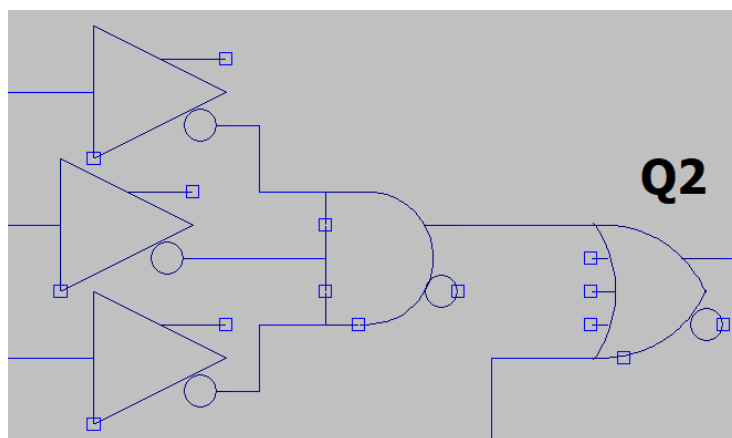
Rysunek 5: Realizacja schematu kodera w symulatorze LTSpice

4.3.3 Koder bitu Q1



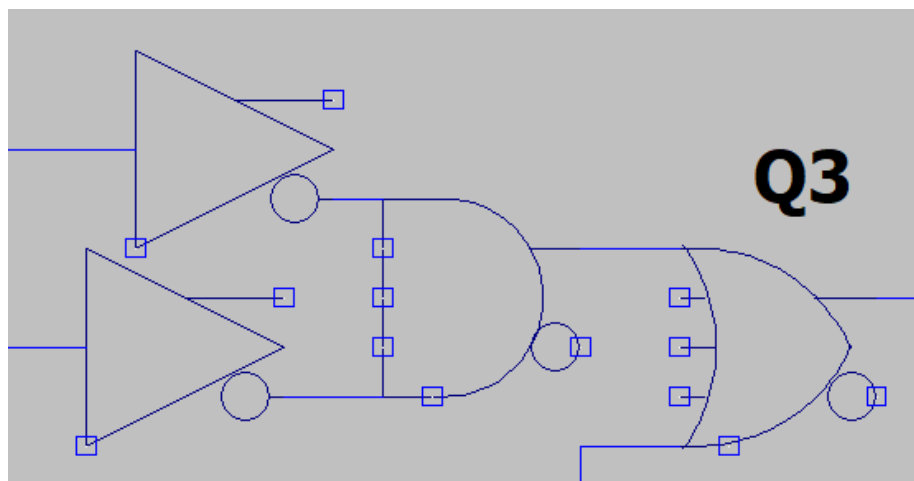
Rysunek 6: Schemat częściowy kodera bitu Q1 w środowisku LTSpice

4.3.4 Koder bitu Q2



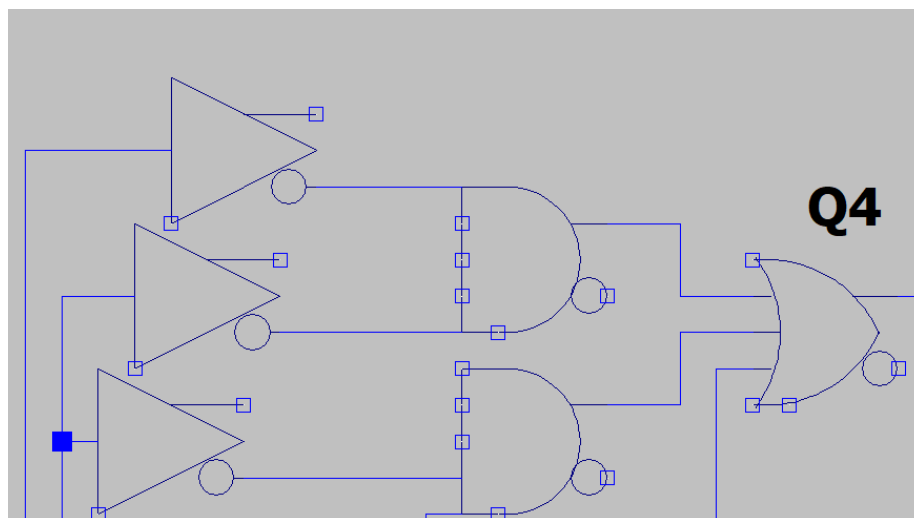
Rysunek 7: Schemat częściowy kodera bitu Q2 w środowisku LTSpice

4.3.5 Koder bitu Q3



Rysunek 8: Schemat częściowy kodera bitu Q3 w środowisku LTSpice

4.3.6 Koder bitu Q4



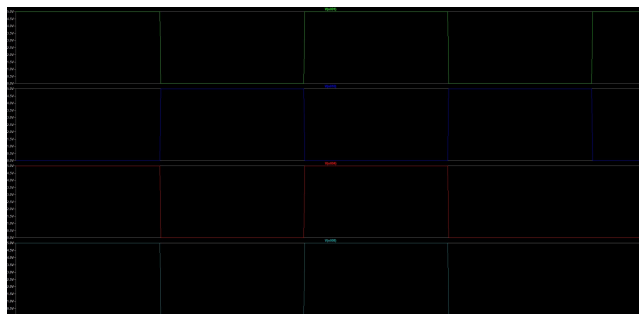
Rysunek 9: Schemat częściowy kodera bitu Q4 w środowisku LTSpice

4.4 Symulacja działania kodera

4.4.1 Wstęp

- W ramach symulacji źródła sygnału pseudolosowego, wykorzystano źródła napięcia V1, V2, V3 oraz V4. Program symulacyjny LTSpice daje możliwość zastosowania źródła napięcia typu PWL, które jest definiowane w pliku tekstowym, co pozwala na odwzorowanie warunków operacyjnych.
- W symulacji, każde z tych źródeł napięcia skonfigurowano do generowania sekwencji napięciowej, która odpowiada sekwencjom sygnałów danych binarnych. Źródło napięcia zmienia stan w określonych momentach czasowych: w chwili 5 ns napięcie jest równe 0 V, natomiast w chwili 5.1 ns wzrasta do 5 V, i tak dalej, aż do chwili 20 ns generując sygnał prostokątny. Taka sekwencja napięciowa jest szczegółowo przedstawiona na [rys. 10] oraz opisana w [tab. 1]. Czas trwania sygnału wybrano tak, aby spełniać założenia specyfikacyjne zegara według normy IEEE 802.3. [1]

4.4.2 Zadany sygnał wejściowy

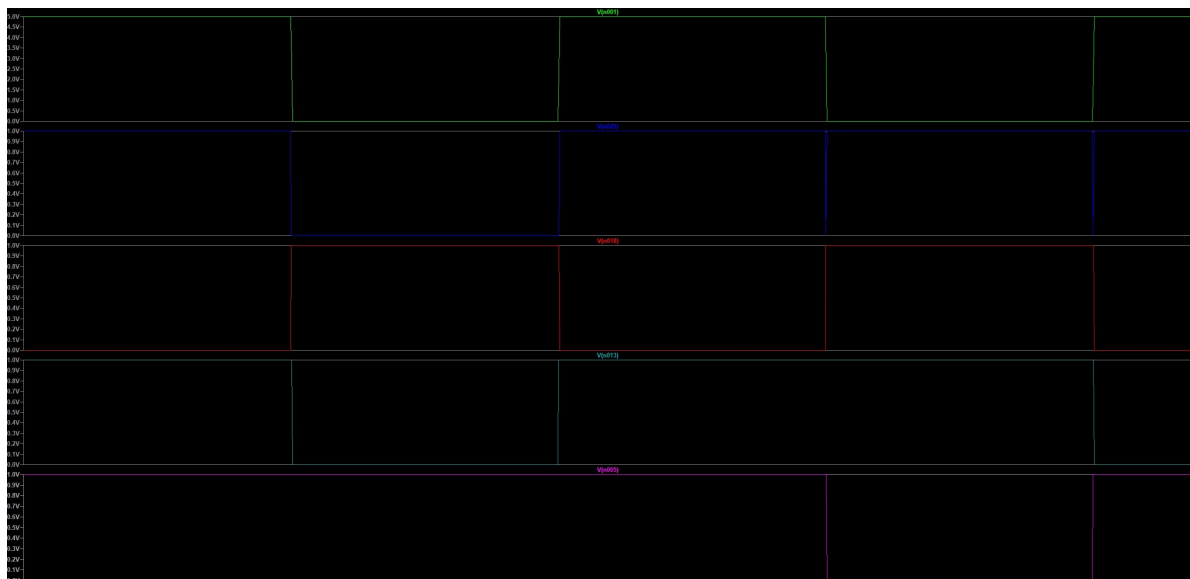


Rysunek 10: Wykresy wejść bitów D0, D1, D2, D3

Tabela 1: Wymuszenia pseudolosowe na źródłach

Wejście D0	Wejście D1	Wejście D2	Wejście D3
5 0	5 5	5 5	5
5.1n 5	5.1n 0	5.1n 0	5.1n 0
10n 5	10n 0	10n 0	10n 0
10.1n 0	10.1n 5	10.1n 5	10.1n 5
20n 0	20n 5	20n 5	20n 5
20.1n 5	20.1n 5	20.1n 0	20.1n 0
Symbol: 1010	Symbol: 0101	Symbol: 1010	Symbol: 0101

4.4.3 Sygnał wyjściowy



Rysunek 11: Wykresy wyjść bitów Q0, Q1, Q2, Q3, Q4

Tabela 2: Zakodowane symbole 4B5B

Lp.	Wyjście Q0	Wyjście Q1	Wyjście Q2	Wyjście Q3	Wyjście Q4
Symbol 1	1	1	0	1	1
Symbol 2	0	0	1	0	1
Symbol 3	1	1	0	1	1
Symbol 4	0	1	1	1	0

4.5 Analiza uzyskanych wyników

4.5.1 Wstęp

W odniesieniu do układu referencyjnego, przebiegi sygnałów wygenerowane przez analizowany koder wykazują identyczność stanów, co umożliwia potwierdzenie poprawności działania układu kodującego. Dokładna analiza stanów, przedstawionych w symulacji za pomocą oprogramowania LTSpice, pozwala na obserwację zgodności wyników uzyskanych (prezentowanych na rysunku 11) z oczekiwanymi rezultatami, które zostały szczegółowo określone w tabeli 2.

4.5.2 Wnioski

- W trakcie projektu zrealizowanego na elementach idealnych nie napotkano znaczących przeszkód, co stanowi dowód na efektywne wykorzystanie środowiska LTSpice do modelowania i analizy układów elektronicznych. Proces ten nie tylko wzmocnił nasze umiejętności w zakresie korzystania z tego narzędzia, ale także nauczył nas efektywnego poszukiwania specjalistycznych informacji, które często były trudno dostępne w dostępnych źródłach internetowych.
- Długotrwała analiza dostępnych materiałów pozwoliła na głębsze zrozumienie problemu oraz wypracowanie odpowiednich rozwiązań. Finalnie, po przeprowadzeniu szeregu testów i weryfikacji, koder został skutecznie zaimplementowany i zademonstrował prawidłowe działanie zgodnie z oczekiwaniami. Wynik ten nie tylko potwierdza skuteczność naszego podejścia projektowego, ale także podkreśla wartość dogłębnej analizy teoretycznej i praktycznej w procesie inżynierskim.

5 Realizacja schematu kodera przy użyciu elementów rzeczywistych

5.1 Wstęp

- Realizacja schematu kodera z wykorzystaniem elementów rzeczywistych przebiegła z napotkaniem pewne wyzwań związanych z dostępnością tych komponentów w środowisku LTSpice. Standardowa biblioteka LTSpice nie zawierała wszystkich potrzebnych elementów rzeczywistych, co skłoniło nas do poszukiwania zewnętrznych źródeł. Ostatecznie, niezbędne komponenty udało się uzyskać dzięki plikom dostępnym na amatorskiej encyklopedii wiedzy o środowisku LTSpice. [6] [7]
- Po pobraniu odpowiednich plików, przystąpiono do modyfikacji pierwotnego schematu, który opierał się na elementach idealnych. Zamieniono te elementy na ich rzeczywiste odpowiedniki, co wymagało dostosowania parametrów układu. Zmiana ta pozwoliła na osiągnięcie realistycznej symulacji, co jest kluczowe dla weryfikacji działania kodera w realnych warunkach. Pomimo początkowych trudności, adaptacja układu do warunków zbliżonych do rzeczywistych przebiegła poprawnie. [5]

5.2 Skrócona specyfikacja techniczna elementów

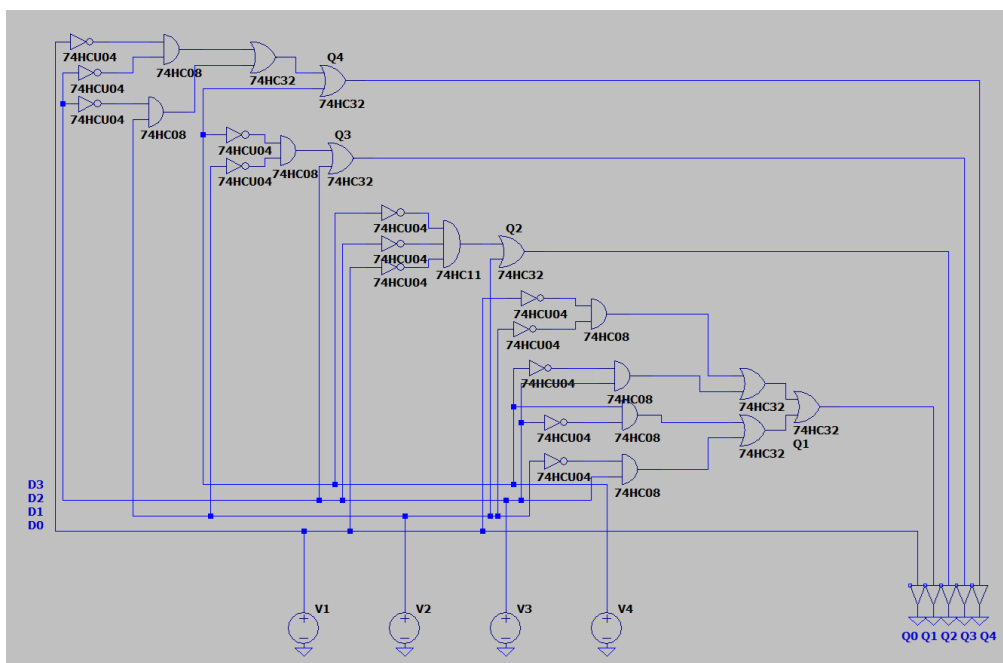
- **Inwertery NOT 74HC04**
Czas narastania i opadania sygnału: Typowo 6 ns przy obciążeniu 50 pF. [8]
- **Bramki AND 74HC08**
Czas narastania i opadania sygnału: Około 8 ns przy obciążeniu 50 pF. [9]
- **Bramki OR 74HC32**
Czas narastania i opadania sygnału: Osiąga do 7 ns przy obciążeniu 50 pF. [10]
- **Bramka AND 74HC11**
Czas narastania i opadania sygnału: Zazwyczaj wynosi około 9 ns przy obciążeniu 50 pF. [11]

5.3 Realizacja w środowisku LTSpice

5.3.1 Wstęp

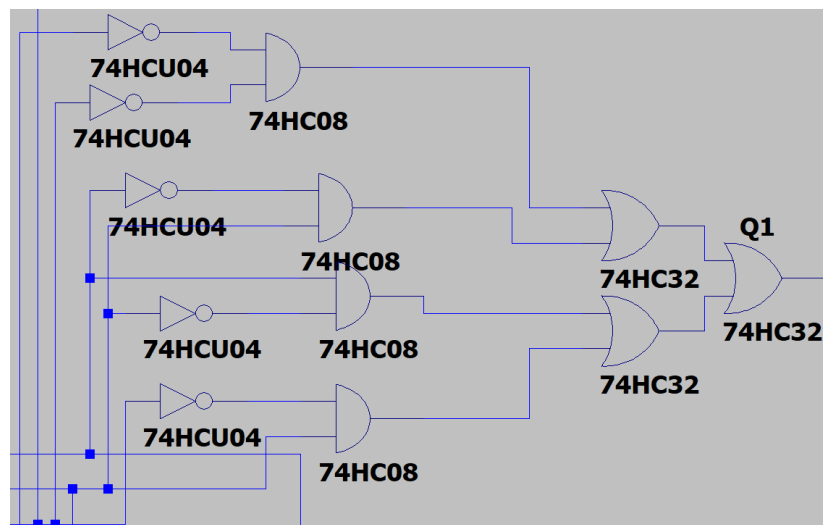
- Realizacja schematu kodera na elementach rzeczywistych wiązała się z koniecznością zamiany elementów idealnych, standardowo dostępnych w bibliotece LTSpice, na ich rzeczywiste odpowiedniki. Elementy te to 74HC04, 74HC08, 74HC32, i 74HC11. [8] [9] [10] [11] Modele tych nie znaleziono w standardowych bibliotekach LTSpice, niezbędne komponenty udało się uzyskać dzięki plikom dostępnym na amatorskiej encyklopedii wiedzy o środowisku LTSpice. [6] [7]
- Przeprowadzono proces integracji tych modeli z istniejącym schematem kodera. Wymagało to modyfikacji parametrów komponentów w schemacie oraz dostosowania połączeń, aby uwzględnić rzeczywiste charakterystyki czasowe i fizyczne komponentów. Dzięki tym zmianom, udało się osiągnąć wiarygodne rezultaty symulacji, co pozwoliło na ocenę działania kodera w warunkach, występują podczas rzeczywistego użytkowania.

5.3.2 Realizacja kodera



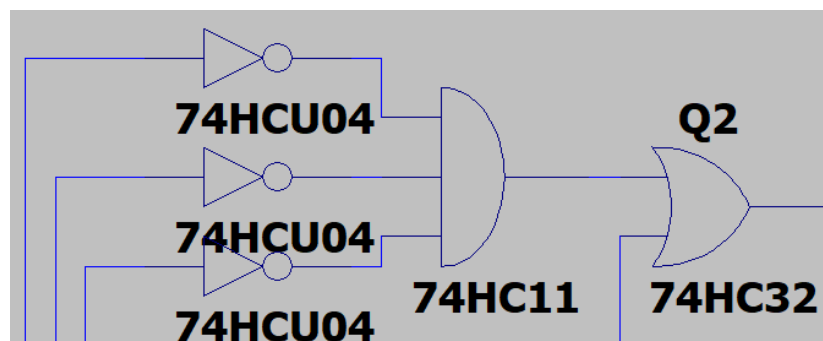
Rysunek 12: Realizacja schematu kodera w środowisku LTSpice

5.3.3 Koder bitu Q1



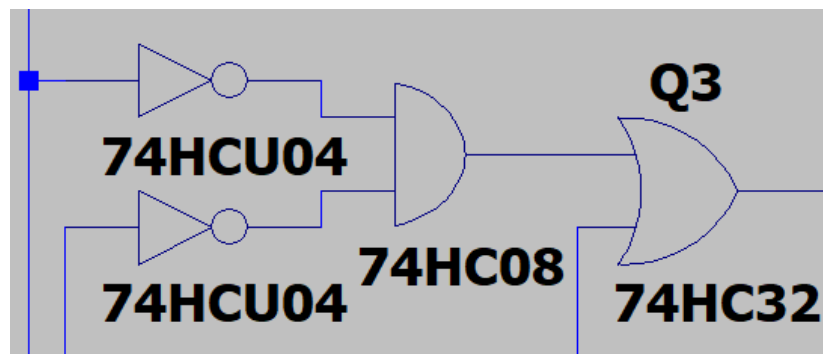
Rysunek 13: Schemat częściowy kodera bitu Q1 w środowisku LTSpice

5.3.4 Koder bitu Q2



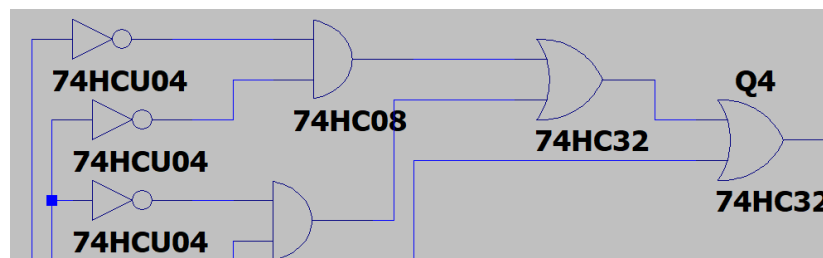
Rysunek 14: Schemat częściowy kodera bitu Q2 w środowisku LTSpice

5.3.5 Koder bitu Q3



Rysunek 15: Schemat częściowy kodera bitu Q3 w środowisku LTSpice

5.3.6 Koder bitu Q4



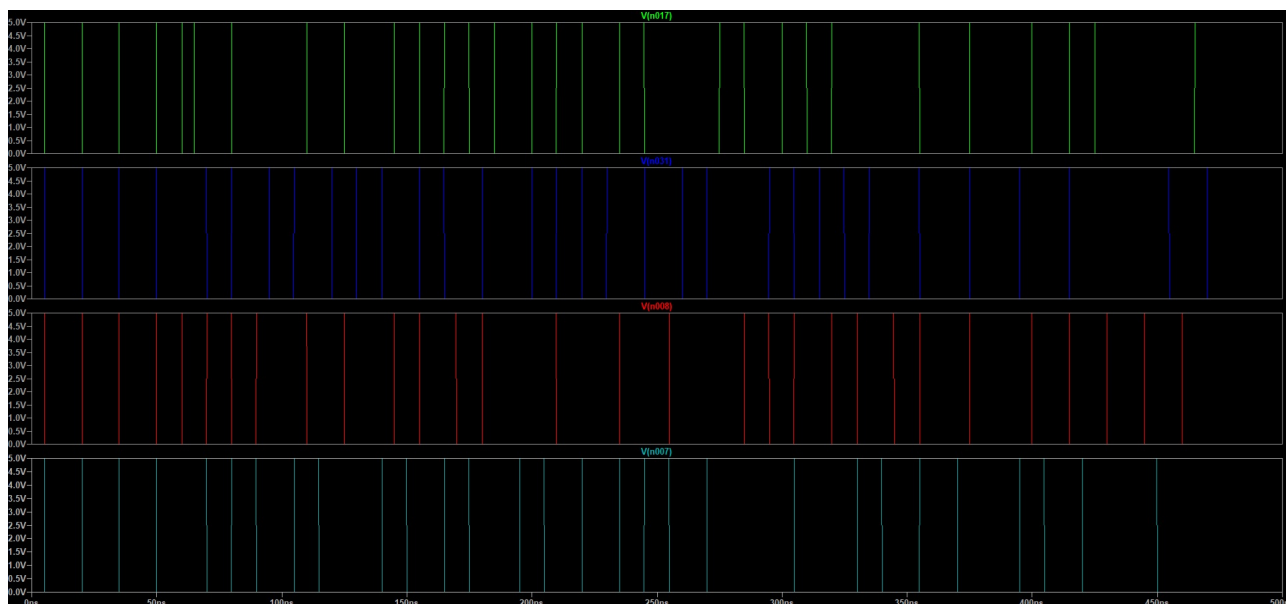
Rysunek 16: Schemat częściowy kodera bitu Q4 w środowisku LTSpice

5.4 Symulacja działania

5.4.1 Wstęp

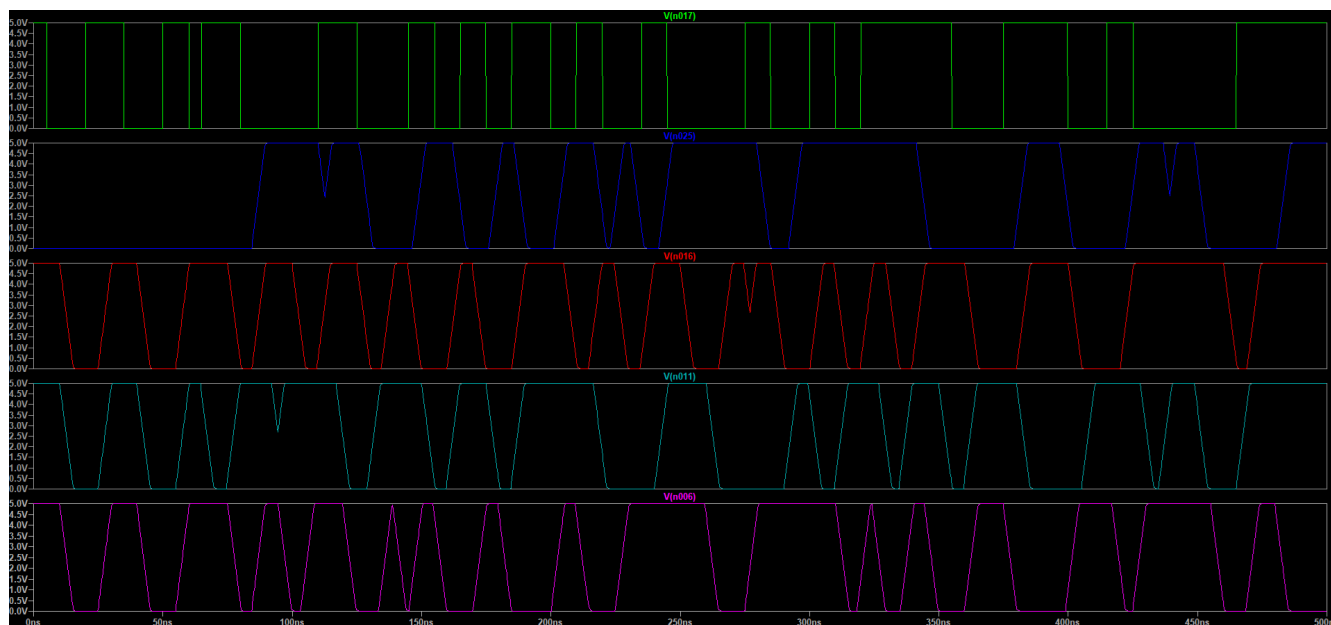
- Analogicznie do przypadku układu na elementach idealnych użyliśmy źródeł pseudolosowych, tutaj źródło sygnału danych zostało przedłużone aby działać do 500ns. Wykorzystywane są źródła napięcia V1, V2, V3 oraz V4. Wycinek definicji działania przedstawiona jest w załączonym pliku tekstowym, co pozwala na precyzyjne odwzorowanie rzeczywistych warunków operacyjnych
- W symulacji, każde z tych źródeł napięcia zostało skonfigurowane do generowania sekwencji napięciowej, która odpowiada sekwencjom zegarowych sygnałów binarnych. Analogicznie jak w układzie na elementach idealnych, źródło napięcia zmienia stan w określonych momentach czasowych: w chwili 5 ns napięcie jest równe 0 V, natomiast w chwili 5.1 ns wzrasta do 5 V, i tak dalej, aż do chwili 500 ns generując sygnał prostokątny. Taka sekwencja napięciowa jest w wycinku przedstawiona na [rys. 17]. Czas trwania sygnału wybrano tak, aby spełniać założenia specyfikacyjne zegara według normy IEEE 802.3. [1]

5.4.2 Zadany sygnał wejściowy



Rysunek 17: Wykresy wejść bitów D0, D1, D2, D3

5.4.3 Sygnał wyjściowy



Rysunek 18: Wykresy wyjść bitów Q0, Q1, Q2, Q3, Q4

5.5 Analiza uzyskanych wyników

5.5.1 Wstęp

- W ramach projektu zrealizowano symulację kodera wykorzystując zarówno elementy idealne, jak i rzeczywiste, w środowisku LTSpice. Z analizy wyników symulacji, której efekty są widoczne na załączonym obrazie, można zaobserwować znaczące różnice w zachowaniu sygnałów generowanych przez te dwa rodzaje elementów. Głównym celem symulacji było porównanie dynamiki sygnałów wyjściowych kodera z uwzględnieniem czasów narastania i opadania, które są kluczowe w zastosowaniach rzeczywistych.
- Symulacja na elementach rzeczywistych pokazała wyraźne efekty ograniczeń fizycznych komponentów, takich jak artefakty sygnału, czas narastania i opadania sygnału, co nie występuje w przypadku modeli idealnych. Te różnice mają krytyczne znaczenie dla projektowania układów cyfrowych, które działają w rzeczywistych warunkach, zapewniając dokładność w dostarczaniu danych.

5.5.2 Parametry układu

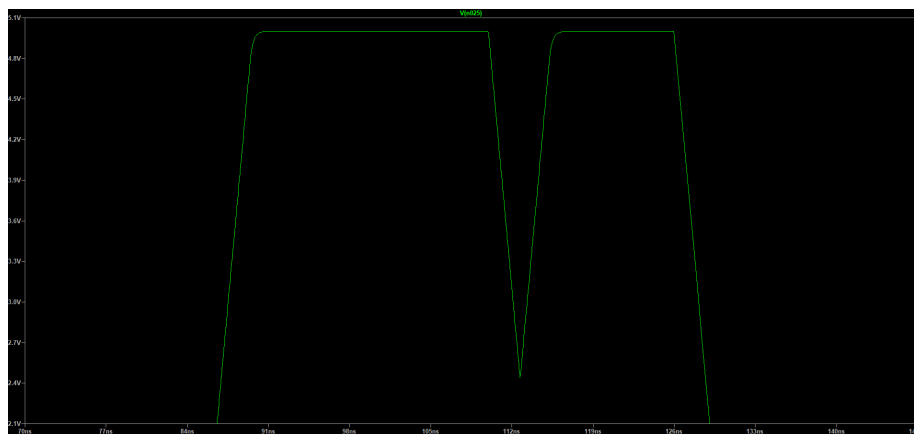
- **Przepustowość przy użyciu elementów idealnych:** $100\text{Mbit/s} \pm 1\%$
- **Czas narastania:** $5\text{ns} \pm 1\%$
- **Czas trwania bitu:** $10\text{ns} \pm 1\%$
- **Przybliżony czas trwania symbolu:** $15\text{ns} \pm 1\%$
- **Zmierzona przepustowość przy użyciu elementów rzeczywistych:** $80\text{Mbit/s} \pm 10\%$

5.5.3 Wnioski

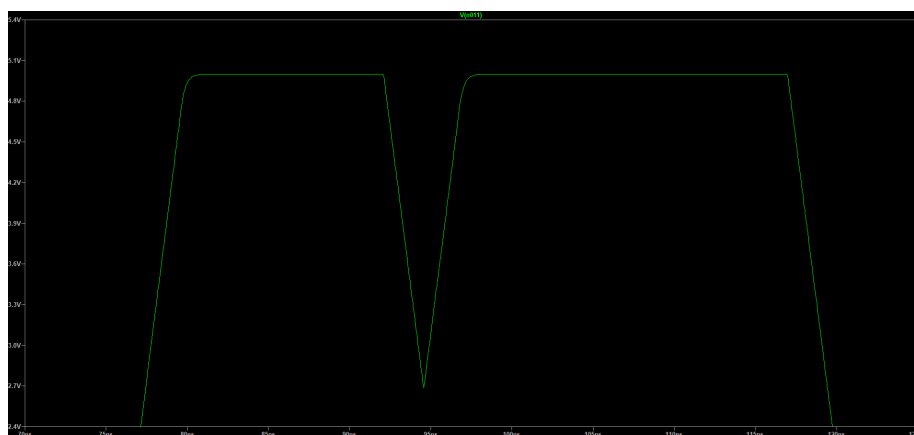
- **Czasy Narastania i Opadania:** Symulacja z elementami rzeczywistymi uwidoczniła czasy narastania i opadania na sygnałach wyjściowych, które są nieobecne w symulacji z elementami idealnymi. Te czasy wpływają na synchronizację sygnału w rzeczywistych aplikacjach cyfrowych, gdzie dokładne czasowanie jest niezbędne. Tworząc układy należy brać pod uwagę te parametry, aby zapewnić niezawodność i efektywność działania układów w różnych warunkach operacyjnych.
- **Optymalizacja:** Realizując układy, należy zwrócić szczególną uwagę na minimalizację ilości elementów, gdyż każdy zbędny, błędnie dobrany lub nieoptymalnie wykorzystany element generuje dodatkowe opóźnienia lub błędy kodowania, co bezpośrednio przekłada się na wyniki końcowe.

- **Artefakty:** Artefakty sygnału wskazują na to, że realne komponenty posiadają ograniczenia w szybkości reakcji na zmiany stanów, co jest kluczowe w szybkich systemach cyfrowych. Szybkość zmiany stanu sygnału wpływa na możliwość prawidłowego odczytu danych przez inne komponenty systemu. W obu przypadkach napięcie spada znacznie poniżej oczekiwanej wartości 5V, osiągając wartości około 2.4V i 2.7V. Tego rodzaju spadki mogą być wynikiem różnych czynników, takich jak opór wewnętrzny elementów lub nieidealne właściwości komponentów, które nie są w stanie utrzymać stabilnego poziomu napięcia pod obciążeniem.

5.5.4 Artefakty sygnału



Rysunek 19: Spadek napięcia z 5V do około 2.4V



Rysunek 20: Spadek napięcia z 5V do około 2.7V

6 Podsumowanie

6.1 Elementy idealne

Symulacja na elementach idealnych dostarcza podstawowych danych na temat teoretycznego funkcjonowania kodera. Ze względu na brak ograniczeń fizycznych, takich jak opóźnienia lub rezystancja wewnętrzna, modele idealne pozwalają na przeprowadzenie symulacji w uproszczonej formie. Wyniki tej symulacji zapewniają czyste i klarowne dane wyjściowe, które zakrzywiają rzeczywiste działanie układu. Są one użyteczne głównie do wstępnych testów logicznych struktury kodera, pozwalając na identyfikację ewentualnych błędów logicznych w projektowaniu bez zakłóceń wynikających z fizycznych właściwości elementów.

6.2 Elementy rzeczywiste

Symulacja na elementach rzeczywistych jest znacznie bardziej kompleksowa. Wprowadzenie modeli, które uwzględniają czasy narastania i opadania sygnału, oporność, pojemność oraz inne realne charakterystyki elementów, pozwala na dokładniejsze odwzorowanie rzeczywistego działania układu. Takie symulacje są kluczowe do testowania układu w warunkach, które będą występować w praktycznych zastosowaniach, umożliwiając identyfikację potencjalnych problemów technicznych, takich jak problemy z czasowaniem, interferencje czy błędy wynikające z opóźnień sygnałów. Symulacja ta pozwala również na optymalizację projektu pod kątem wydajności i niezawodności, co jest niezbędne przed finalną realizacją projektu.

6.3 Całokształt

Analiza obu typów symulacji jest niezbędna dla kompleksowego rozwoju i weryfikacji układów elektronicznych. Symulacje na elementach idealnych oferują prostotę, która jest użyteczna w początkowych etapach projektowania, podczas gdy symulacje na elementach rzeczywistych dostarczają głębszego wglądu w działanie układu w realnych warunkach. Połączenie obu podejść pozwala na bardziej efektywne projektowanie, testowanie i implementację układów elektronicznych, co finalnie przekłada się na większą niezawodność i funkcjonalność finalnego produktu.

7 Wnioski końcowe

- **Wyzwania w poszukiwaniu informacji:** Poszukiwanie informacji niezbędnych do realizacji zaawansowanych układów elektronicznych w sieci często nie przynosi oczekiwanych rezultatów. Wiele kluczowych danych na temat zaawansowanych technologii pozostaje niejawnych, a korporacje rzadko udostępniają informacje dotyczące swoich wysokobudżetowych projektów. Taki stan rzeczy znacząco utrudnia dostęp do pełnych, niezbędnych zasobów wiedzy, co stawia dodatkowe bariery w procesie naukowym i inżynierskim.
- **Projektowanie układu na elementach idealnych:** Realizacja projektu z wykorzystaniem elementów idealnych napotkała na przeszkody wynikające z braku dostępu do wystarczających informacji kluczowych do prawidłowego odtworzenia układu. Dopiero uzyskanie dostępu do schematów logicznych, które umożliwiły zastosowanie metody siatek Karnaugh do projektowania, pozwoliło na efektywne skonstruowanie układu idealnego w LTSpice. Ten etap podkreślił znaczenie posiadania konkretnych, szczegółowych danych dla osiągnięcia wiarygodnych wyników symulacji.
- **Implementacja układu z elementami rzeczywistymi:** Realizacja projektu z wykorzystaniem elementów rzeczywistych również napotkała bariery, głównie z powodu braku dostępnych modeli tych elementów w bibliotekach LTSpice. Sytuację udało się rozwiązać poprzez odnalezienie odpowiednich modeli na amatorskiej stronie internetowej, co umożliwiło kontynuację prac nad projektem. Ten przypadek dodatkowo podkreśla, jak kluczowe jest posiadanie dostępu do odpowiednich zasobów informacyjnych dla efektywnej realizacji projektów inżynierskich.
- **Edukacyjne i praktyczne korzyści:** Realizacja tego projektu przyniosła znaczne korzyści edukacyjne i praktyczne. Użytkowanie środowiska LTSpice w znaczny sposób poszerzyło nasze umiejętności praktyczne oraz utrwaliło wiedzę z zakresu techniki cyfrowej. Projekt ten uświadomił również, jak cenne i trudno dostępne są informacje dotyczące zaawansowanych rozwiązań technologicznych oraz projektów w dziedzinie elektroniki.

Literatura

- [1] IEEE Standard Association, "IEEE 802.3-2015 - Standard for Ethernet," IEEE, 2015. [Online]. Adres: <https://ieeexplore.ieee.org/document/7428776> [Dostęp: 18-Kwie-2024].
- [2] A. Opal, "Ethernet," Akademia Górniczo-Hutnicza w Krakowie. [Online]. Adres: <https://home.agh.edu.pl/~opal/sieci/wyklady/4-ethernet.pdf> [Dostęp: 18-Kwie-2024].
- [3] M. Mazzola, L. Cafiero, and M. DeNicolò, "Method and apparatus for multilevel encoding for a local area network," U.S. Patent US 5280500A, Jan. 18, 1994. [Online]. Adres: <https://patents.google.com/patent/US5280500> [Dostęp: 18-Kwie-2024].
- [4] Editorial Team, "100BaseTX: Implementation and Troubleshooting," Network Encyclopedia, Feb. 4, 2024. [Online]. Adres: <https://networkencyclopedia.com/100basetx/> [Dostęp: 18-Kwie-2024].
- [5] Actel Corporation, "Using Actel FPGAs to Implement the 100 Mbit/s Ethernet Standard," April 1996. [Online]. Adres: https://www.microsemi.com/document-portal/doc_view/129803-100mbethernet-an [Dostęp: 09-Maj-2024].
- [6] Współautorzy LTwiki, "Main Page - LTwiki-Wiki for LTspice," Styczeń 17, 2024. [Online]. Adres: https://ltwiki.org/index.php?title=Main_Page [Dostęp: 10-Cze-2024].
- [7] Bordodynov, "Bordodynov's Electronics Web Page" 2024 [Online]. Adres: <http://bordodynov.ltwiki.org/> [Dostęp: 11-Cze-2024].
- [8] Diodes Incorporated, "74HC04 Datasheet," Diodes, [Online]. Adres: <https://www.diodes.com/assets/Datasheets/74HC04.pdf> [Dostęp: 12-Cze-2024].
- [9] Diodes Incorporated, "74HC08 Datasheet," Diodes, [Online]. Adres: <https://www.diodes.com/assets/Datasheets/74HC08.pdf> [Dostęp: 12-Cze-2024].
- [10] Diodes Incorporated, "74HC32 Datasheet," Diodes, [Online]. Adres: <https://www.diodes.com/assets/Datasheets/74HC32.pdf> [Dostęp: 12-Cze-2024].
- [11] Texas Instruments, "CD74HC11 Datasheet," Texas Instruments, [Online]. Adres: <https://www.ti.com/lit/ds/symlink/cd74hc11.pdf> [Dostęp: 12-Cze-2024].

Spis rysunków

1	Kodowane symbole kodu 4B5B	6
2	Tablica prawdy przetłumaczona na siatki Karnaugh	7
3	Uproszczony schemat kodera przy użyciu widoku logiki FPGA	8
4	Schemat kodera 4B5B w języku PALASM2	8
5	Realizacja schematu kodera w symulatorze LTSpice	9
6	Schemat częściowy kodera bitu Q1 w środowisku LTSpice	10
7	Schemat częściowy kodera bitu Q2 w środowisku LTSpice	10
8	Schemat częściowy kodera bitu Q3 w środowisku LTSpice	11
9	Schemat częściowy kodera bitu Q4 w środowisku LTSpice	11
10	Wykresy wejść bitów D0, D1, D2, D3	12
11	Wykresy wyjść bitów Q0, Q1, Q2, Q3, Q4	13
12	Realizacja schematu kodera w środowisku LTSpice	16
13	Schemat częściowy kodera bitu Q1 w środowisku LTSpice	17
14	Schemat częściowy kodera bitu Q2 w środowisku LTSpice	17
15	Schemat częściowy kodera bitu Q3 w środowisku LTSpice	18
16	Schemat częściowy kodera bitu Q4 w środowisku LTSpice	18
17	Wykresy wejść bitów D0, D1, D2, D3	19
18	Wykresy wyjść bitów Q0, Q1, Q2, Q3, Q4	20
19	Spadek napięcia z 5V do około 2.4V	22
20	Spadek napięcia z 5V do około 2.7V	22

Spis tabel

1	Wymuszenia pseudolosowe na źródłach	12
2	Zakodowane symbole 4B5B	13