



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Imię i nazwisko studenta: Cezary Wieczorkowski

Nr albumu: 188584

Poziom kształcenia: studia pierwszego stopnia

Forma studiów: stacjonarne

Kierunek studiów: Elektronika i telekomunikacja

Profil: Komputerowe systemy elektroniczne

PRACA DYPLOMOWA INŻYNIERSKA

Tytuł pracy w języku polskim: Projekt i budowa laboratoryjnego stanowiska sterującego szyną danych

Tytuł pracy w języku angielskim: Design and construction of a laboratory data bus control station

Opiekun pracy: dr inż. Kamil Stawiarski

Streszczenie

Celem projektu było zaprojektowanie oraz uruchomienie zestawu laboratoryjnego umożliwiającego realizację ćwiczeń z sterowania szyną danych w laboratorium techniki cyfrowej. W ramach projektu dokonano analizy istniejącego stanowiska pod kątem możliwości rozbudowy oraz poprawy funkcjonalności. Następnie zaprojektowano oraz zbudowano prototyp nowego stanowiska uwzględniając wnioski z analizy poprzedniego.

Słowa kluczowe: Technika cyfrowa, stanowisko laboratoryjne, emulacja, szyna danych, ALU

Dziedzina nauki i techniki, zgodne z wymogami OECD: nauki inżynierijno-techniczne: automatyka, elektronika, elektrotechnika i technologie kosmiczne

Abstract

The goal of this project was to design and construct a laboratory set that enables the performance of exercises on data bus control in a digital electronics laboratory. As part of the project, an analysis of the existing station was carried out in terms of the possibility of expansion and improvement of functionality. Then, a prototype of the new station was designed and built, taking into account the conclusions from the analysis of the previous one.

Keywords: Digital electronics, laboratory set, emulation, data bus, ALU

Spis treści

Wykaz ważniejszych oznaczeń i skrótów	6
1 Wstęp i cel pracy	7
2 Analiza istniejącego stanowiska	8
2.1 Zasada działania stanowiska	8
2.1.1 Opis stanowiska	8
2.1.2 Elementy składowe stanowiska	9
2.1.3 Interfejs użytkownika	11
2.1.4 Tryby pracy stanowiska	12
2.2 Krytyczna ocena stanowiska	13
2.2.1 Założenia funkcjonalne	13
2.2.2 Interfejs użytkownika	13
2.2.3 Platforma sprzętowa	13
3 Koncepcja nowego stanowiska	15
3.1 Założenia projektowe	15
3.2 Projekt sprzętowy	15
3.2.1 Wybór mikrokontrolera	15
3.2.2 Elementy interfejsu użytkownika	15
3.2.3 Dodatkowe peryferia	16
3.2.4 Schemat blokowy sprzętowej części zestawu	16
3.3 Architektura oprogramowania	17
3.3.1 Struktura programu	17
3.3.2 Emulacja elementów układu szyny danych	18
3.3.3 Obsługa peryferiów sprzętowych	18
3.3.4 Obsługa interfejsu użytkownika	19
4 Implementacja stanowiska	21
4.1 Część sprzętowa	21
4.2 Część programowa	21
5 Testy stanowiska	22
6 Podsumowanie	23
Bibliografia	24
Spis rysunków	25
Spis tabel	26

Wykaz ważniejszych oznaczeń i skrótów

ALU	(ang. arithmetic logic unit) jednostka arytmetyczno-logiczna
RAM	(ang. random access memory) pamięć o dostępie swobodnym
LED	(ang. light emitting diode) dioda elektroluminescencyjna
LCD	(ang. liquid crystal display) wyświetlacz ciekłokrystaliczny
GPIO	(ang. general-purpose input/output) uniwersalne wejście/wyjście
I/O	(ang. input/output) wejście/wyjście
I2C	(ang. inter integrated circuit) szeregowy synchroniczny interfejs komunikacyjny
SDK	(ang. software development kit) zestaw narzędzi do rozwoju oprogramowania
PCI	(ang. peripheral component interconnect) interfejs komunikacyjny

1 Wstęp i cel pracy

Elektronika cyfrowa obecna jest w prawie każdym nowoczesnym urządzeniu. Jest to jedna z ważniejszych dziedzin elektroniki która umożliwia rozwój nowoczesnych technologii które leżą u podstaw współczesnego społeczeństwa. W procesie projektowania układów cyfrowych zawsze zachodzi potrzeba wymiany danych pomiędzy różnymi podzespołami układu. Jednym z sposobów realizacji przepływu danych jest zastosowanie szyny danych. Jest to zespół linii (przewodów) służących do przesyłania danych pomiędzy nadajnikami oraz odbiornikami do niej podłączonymi. Do szyny danych można podłączyć wiele nadajników oraz odbiorników danych jednak tylko jeden nadajnik może wysyłać dane w danym momencie, gdyż podłączenie do wspólnej linii kilku nadajników może spowodować ich uszkodzenie. Zapobiega się temu stosując na wyjściach nadajników bufor tryjstanowe. Układy kontrolne sterują stanem tych buforów tak aby tylko wyjście jednego nadajnika było aktywne w danym momencie.

Celem pracy jest budowa stanowiska laboratoryjnego do realizacji ćwiczeń z sterowaniem szyną danych w laboratorium techniki cyfrowej. To ćwiczenie ma za zadanie zapoznać studentów z sposobami realizacji komunikacji między różnymi układami cyfrowymi przy pomocy wspólnej magistrali danych. Rozwiązania tego typu są powszechnie stosowane w nowoczesnych układach cyfrowych takich jak mikrokontrolery oraz mikroprocesory w celu wymiany danych pomiędzy różnymi podzespołami danego układu. W oparciu o szynę danych działa również wiele interfejsów znajdujących się w komputerach osobistych takich jak PCI czy też magistrala pamięci systemowej. Zrozumienie mechanizmów działania systemu opartego o magistralę danych pomoże studentom w zrozumieniu zasad działania bardziej skomplikowanych układów cyfrowych.

2 Analiza istniejącego stanowiska

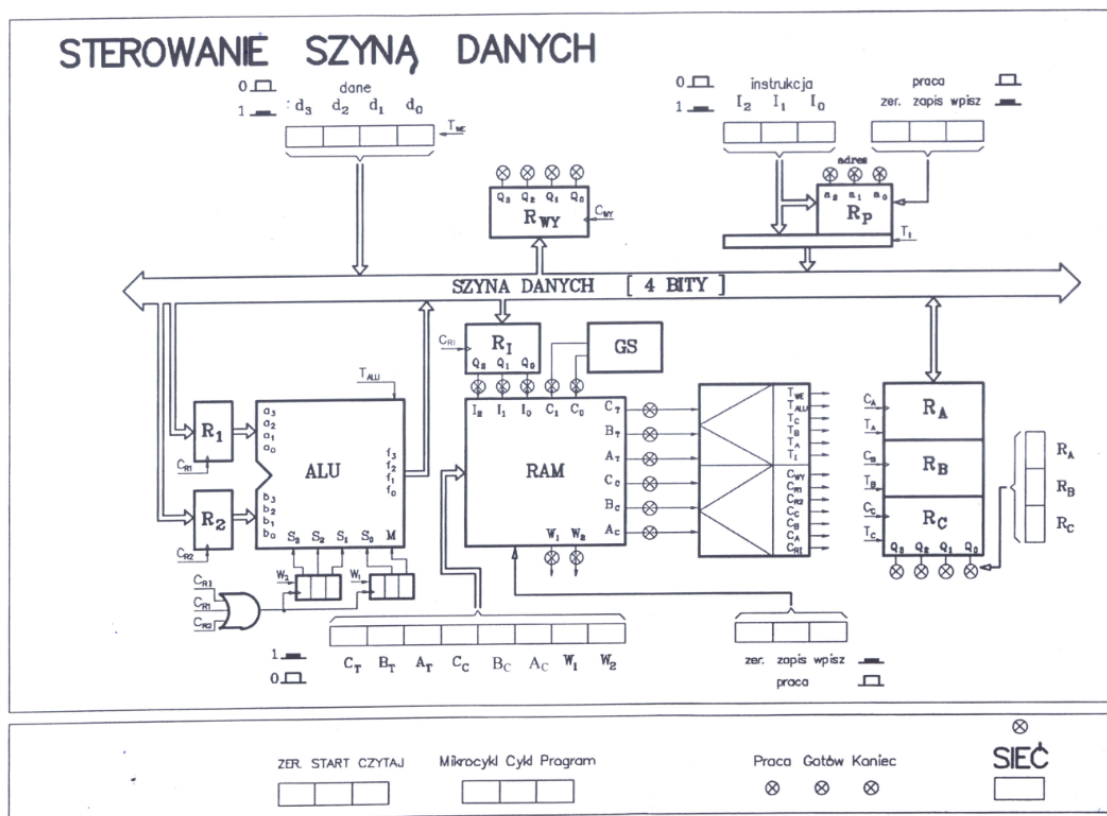
Poniższy rozdział został napisany na podstawie instrukcji obsługi zestawu [1] .

2.1 Zasada działania stanowiska

2.1.1 Opis stanowiska

Zestaw laboratoryjny modeluje system komunikacji oparty na 4 bitowej szynie danych. W skład zestawu wchodzi urządzenia nadawcze i odbiorcze podłączone do szyny danych, pamięć RAM, logika sterująca procesem wymiany danych oraz elementy interfejsu użytkownika. Na rysunku 2.1 można zobaczyć płytę czołową zestawu która przedstawia jego schemat blokowy oraz interfejs użytkownika. Z pomocą interfejsu użytkownika można wprowadzić do pamięci RAM instrukcje które zostaną przez zestaw wykonane. Instrukcje te mogą realizować operacje arytmetyczne lub logiczne, zapisywać dane do rejestrów lub pobierać dane od użytkownika. Pozwala to na realizację prostych programów których efektem może być na przykład obliczanie kolejnych elementów ciągu fibonacciego czy wykonanie zadanej operacji matematycznej na liczbach podanych przez użytkownika.

Stanowisko laboratoryjne w pierwotnej wersji było zrealizowane w oparciu o układy logiczne z serii 74 połączone w sposób pokazany na płycie czołowej zestawu. Obecna wersja zestawu została zrealizowana w ramach pracy magisterskiej [2] i jest ona oparta o mikrokontroler Atmega 32 oraz Atmega 8 emulujące część fizycznych elementów zestawu. Mikrokontrolery te są wspomagane przez szereg układów takich jak buforory oraz przerzutniki. W dalszej części tego rozdziału zostało opisane działanie pierwotnej wersji zestawu.



Rysunek 2.1: Płyta czołowa zestawu

2.1.2 Elementy składowe stanowiska

Pamięć RAM

Pamięć ram w zestawie jest zorganizowana jako 32 słowa po 8 bitów.

C_T	B_T	A_T	C_C	B_C	A_C	W_1	W_2
-------	-------	-------	-------	-------	-------	-------	-------

Tabela 1: Symbole przypisane poszczególnym bitom słowa w pamięci RAM

Zawartość słów w pamięci RAM kontrolują pracę zestawu:

- bity C_T, B_T, A_T - wybór nadajnika na szynie
- bity C_C, B_C, A_C - wybór odbiornika na szynie
- bity W_1, W_2 - wybór operacji ALU

Pamięć RAM zawiera instrukcje pracy zestawu. W celu wykonania kolejnych instrukcji należy zwiększyć o 1 obecny adres pamięci. Wykonanie jednej instrukcji z pamięci RAM nazywamy mikrocyklem.

Wejścia adresowe pamięci są połączone z rejestrem Rp (trzy najstarsze bity) oraz z generatorem Gs (dwa najmłodsze bity). Trzy najstarsze bity wyjścia danych pamięci połączone są z wejściami adresowymi dekodera 1 natomiast kolejne trzy z wejściami adresowymi dekodera 2. Dwa najmłodsze bity wyjścia danych pamięci połączone są wejściami informacyjnymi rejestrów przesuwanych instrukcji ALU.

Generator GS

Generator GS jest licznikiem modulo 4. Jego wyjścia połączone są z dwoma najmłodszymi wejściami adresowymi pamięci RAM. Podczas pracy generator wygeneruje kolejno wartości dwóch najstarszych bitów adresu od 00 do 11. W zależności od trybu pracy zestawu generowane są kolejne adresy pamięci do zapisu lub odczytu i wykonani jako instrukcje w mikrocyklu.

Dekoder sygnałów sterujących

Dekodery sygnałów sterujących to układy konwertujące liczby binarne na kod 1 z n. Dekoder 1 wytwarza sygnały sterujące nadajnikami które podawane są na wejścia sterujące przyłączaniem poszczególnych nadajników do szyny danych.

C_T	B_T	A_T	T_I	T_A	T_B	T_C	T_{ALU}	T_{WE}
0	0	0	1	0	0	0	0	0
0	0	1	0	1	0	0	0	0
0	1	0	0	0	1	0	0	0
0	1	1	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	1

Tabela 2: Tablica prawdy dla dekodera 1

Dekoder 2 działa analogicznie do dekodera 1 ale wytwarza sygnały sterujące odbiornikami. Sygnały generowane przez dekodek 2 podawane są na wejścia zapisu poszczególnych odbiorników.

C_C	B_C	A_C	C_{RI}	C_A	C_B	C_C	C_{R1}	C_{R2}	C_{WY}
0	0	0	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0
0	1	1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0
1	0	1	0	0	0	0	0	1	0
1	1	0	0	0	0	0	0	0	1

Tabela 3: Tablica prawdy dla dekodera 2

Jednostka arytmetyczno logiczna ALU

Układ ten realizuje operacje arytmetyczne i logiczne na dwóch liczbach 4 bitowych zapisanych w rejestrach R_1 oraz R_2 . Rodzaj wykonywanej operacji określa pięciobitowe słowo M, S_3, S_2, S_1, S_0 które jest zapisywane w dwóch 3 bitowych rejestrach przesuwnych. Na rysunku 2.1 pokazano sposób połączenia rejestrów przesuwnych w układzie. Na ich wejścia danych podane są sygnały W_1, W_2 z pamięci RAM. Natomiast na wejścia zegarowe podana jest suma logiczna sygnałów C_{RI}, C_{R1}, C_{R2} . Z tych zależności wynika, że podczas projektowania instrukcji z użyciem ALU w celu ustawienia odpowiedniego kodu operacji należy przewidzieć wygenerowanie sygnałów C_{RI}, C_{R1}, C_{R2} . Tabela 4 przedstawia sposób kodowania operacji do wykonania przez ALU.

I_2	I_1	I_0	C_1	C_0	C_T	B_T	A_T	C_C	B_C	A_C	W_1	W_2
X	X	X	0	0							M	S_3
			0	1							S_2	S_1
			1	0							X	S_0
			1	1							X	X

Tabela 4: Zasada kodowania operacji do wykonania przez ALU

Jednostka arytmetyczno logiczna ma możliwość zrealizowania 16 operacji arytmetycznych oraz logicznych:

Rejestry pomocnicze

Rejestry są układami przechowującymi dane. W zestawie występują następujące rejestry:

- R_A, R_B, R_C - są to rejestry przeznaczone do wykorzystania przez użytkownika, mogą one zarówno odbierać jak i nadawać dane na szynę
- R_1, R_2 - przechowują argumenty operacji ALU, mogą tylko odbierać dane z szyny
- R_{wy} - służy do prezentacji wyników operacji wykonywanych przez zestaw laboratoryjny, może tylko odbierać dane z szyny

Lp.	S_3	S_2	S_1	S_0	Operacje arytmetyczne ($M = 0$)	Operacje logiczne ($M = 1$)
1	0	0	0	0	$Y = 0$	$Y = \overline{R_1}$
2	0	0	0	1	$Y = R_1 \div R_2$	$Y = \overline{R_1} \wedge \overline{R_2}$
3	0	0	1	0	$Y = R_1 \times R_2 - 1$	$Y = \overline{R_1} \vee R_2$
4	0	0	1	1	$Y = R_1 + R_2$	$Y = 1$
5	0	1	0	0	$Y = R_1^2 - R_2^2$	$Y = R_1 \oplus R_2$
6	0	1	0	1	$Y = R_1 \times 2$	$Y = \overline{R_1} \iff R_2$
7	0	1	1	0	$Y = \frac{R_1 + R_2}{2}$	$Y = R_1 \implies R_2$
8	0	1	1	1	$Y = R_1 \times R_2$	$Y = R_1 \wedge R_2$
9	1	0	0	0	$Y = R_1 \% R_2$	$Y = \overline{R_1} \implies \overline{R_2}$
10	1	0	0	1	$Y = R_1 \div 2$	$Y = \overline{\overline{R_1}} \vee R_2$
11	1	0	1	0	$Y = (R_1 \times R_2) - (R_1 + R_2)$	$Y = R_1 \vee R_2$
12	1	0	1	1	$Y = R_1 - R_2$	$Y = \overline{R_1} \oplus \overline{R_2}$
13	1	1	0	0	$Y = (R_1 - R_2)^2$	$Y = R_2$
14	1	1	0	1	$Y = R_1$	$Y = \overline{R_1} \vee \overline{R_2}$
15	1	1	1	0	$Y = R_1^2$	$Y = R_1 \iff R_2$
16	1	1	1	1	$Y = R_1 - 1$	$Y = \overline{R_1} \oplus \overline{R_2}$

Tabela 5: Lista operacji realizowanych przez ALU

- R_I - przechowuje trzy najstarsze bity adresu pamięci, może tylko odbierać dane z szyny
- R_p - to zespół ośmiu rejestrów przeznaczonych do przechowywania adresów pamięci RAM pod którymi znajdują się instrukcje do wykonania przez zestaw. Rejestr ten jest wykorzystywany w trybie pracy Program

2.1.3 Interfejs użytkownika

Na rysunku 2.1 poza schematem blokowym zestawu widzimy przyciski oraz diody LED służące do sterowania zestawem i monitorowania jego pracy.

Kontrola pracy zestawu

Do sterowania pracą zestawu służą następujące przyciski:

- ZER. - przycisk zerowania generatora GS
- START - przycisk rozpoczynający pracę zestawu
- CZYTAJ - przycisk inicjujący odczyt z klawiatury dane i kontynuację pracy zestawu
- Mikrocykl, Cykl, Program - przełączniki wyboru trybu pracy zestawu
- klawiatura dane - klawiatura do wprowadzania danych wejściowych na szynę danych

Zapis i odczyt danych z pamięci RAM

Do obsługi zapisu pamięci RAM służą przyciski:

- zer. - zerowanie adresu przy zapisie i odczycie pamięci RAM

- zapis/praca - wybór pomiędzy zapisem danych do pamięci RAM a pracą zestawu
- wpisz - zapis danych z klawiatury do RAM po jego wciśnięciu adres zwiększa się o 1
- klawiatura $C_T - W_2$ - przyciski do ustawiania wartości bitowej słowa wpisywanego do pamięci RAM

Zapis i odczyt danych z R_P

Zapis oraz odczyt danych z rejestru odbywa się podobnie jak w przypadku pamięci RAM z użyciem odpowiednich przycisków:

- klawiatura dane - klawiatura do wprowadzania danych wejściowych na szynę danych
- klawiatura instrukcja - klawiatura do wprowadzania wartości do rejestru R_P
- zer. - zerowanie adresu przy zapisie i odczycie pamięci R_P
- zapis/praca - wybór pomiędzy zapisem danych do pamięci R_P a pracą zestawu
- wpisz - zapis danych z klawiatury instrukcja do R_P po jego wciśnięciu adres zwiększa się o 1

Monitorowanie pracy zestawu

Na płycie czołowej zestawu znajdują się również diody LED obrazujące wartości

- Adresu pamięci RAM
- Wartości słowa w pamięci RAM
- Wartości słowa w rejestrze R_{WY}
- Wartości słowa w jednym z rejestrów R_A, R_B, R_C wybieranych przyciskami
- Adres rejestru R_P

2.1.4 Tryby pracy stanowiska

Stanowisko posiada trzy tryby pracy:

- Mikrocykl - zestaw po wciśnięciu przycisku START wykona tylko jedną instrukcję z pamięci RAM, zostanie wygenerowana tylko jedna wartość w generatorze GS. Po jej wykonaniu zestaw zatrzyma się i będzie oczekiwał na ponowne wciśnięcie przycisku START.
- Cykl - zestaw po wciśnięciu przycisku START wykona cztery kolejne instrukcje z pamięci RAM. Nastąpią cztery kolejne zmiany wyjść w generatorze GS. Po wykonaniu czterech instrukcji zestaw zatrzyma się i będzie oczekiwał na ponowne wciśnięcie przycisku START.
- Program - po wciśnięciu przycisku START zestaw poda na wyjście rejestru R_P kolejno osiem instrukcji które zostały do niego uprzednio zapisane. Układ wygeneruje kolejne 32 stany generatora GS, zmiana wartości w rejestrze R_P nastąpi co czwartą wartość generatora GS.

Niezależnie od trybu pracy zestawu automatycznie generowane są tylko wartości generatora GS. W celu wykonania kolejnych instrukcji należy manualnie załadować wartość trzech najstarszych bitów adresu pamięci kolejnej instrukcji do rejestru R_I . W tym celu w pierwszej instrukcji każdego bloku czterech instrukcji w pamięci RAM należy umieścić wartość 000000XX. Taka wartość instrukcji odpowiada ustawieniu rejestru R_I jako odbiornika a rejestru R_P jako nadajnika. W przypadku pracy w trybie Mikrocykl oraz Cykl wartość trzech najstarszych bitów adresu pamięci kolejnej instrukcji należy za każdym razem manualnie wprowadzić na klawiaturze instrukcji. Natomiast w trybie Program kolejne osiem wartości rejestru R_P zostanie automatycznie podanych na jego wyjście.

2.2 Krytyczna ocena stanowiska

2.2.1 Założenia funkcjonalne

Zestaw w obecnej formie pozwala na realizację prostych programów obrazujących działanie szyny danych w praktyce. Urządzenia podłączone do szyny współpracują ze sobą w sposób przemysłowy pozwalają na realizowanie prostych operacji na jednostce ALU, zapis danych do rejestrów, pobieranie oraz prezentowanie danych użytkownikowi. Tryby pracy zestawu pozwalają na łatwe zaobserwowanie procesów zachodzących w zestawie podczas wykonywania programu. Tryby cykl i mikrocykl ułatwiają debugowanie programów przechodząc przez każdy cykl lub mikrocykl programu.

2.2.2 Interfejs użytkownika

Interfejs użytkownika zestawu opiera się na przyciskach jako urządzeniach wejściowych oraz diodach LED jako urządzeniach wyjściowych. Wizualizacja stanu poszczególnych rejestrów oraz bitów za pomocą diod LED jest czytelna i intuicyjna. W połączeniu ze schematem blokowym na płycie pozwala to w łatwy sposób zrozumieć jakie działania zestaw wykonuje w danym momencie. Możliwość podglądu adresu pamięci RAM oraz słowa pod nim zapisanego ułatwia znajdowanie błędów w programach uruchamianych na zestawie.

Wprowadzanie danych do zestawu (wartości pamięci RAM, rejestru R_P oraz danych na szynę) odbywa się za pomocą klawiatur. Jest to kilka przycisków gdzie każdy przycisk odpowiada jednemu bitowi wprowadzanego słowa. Wciśnięty przycisk odpowiada wartości 1 a wyciśnięty wartości 0. Wprowadzanie wartości danych binarnych w ten sposób jest intuicyjne i szybkie.

Pozostałe przyciski zestawu służą do kontroli jego pracy oraz kontroli procesów zapisu i odczytu wartości z pamięci RAM oraz rejestru R_P . Jest to sposób sprawdzony i stosunkowo intuicyjny. Głównym ograniczeniem tego podejścia jest trudność w dokonaniu zmian w interfejsie gdyż jego elementy są stałymi elementami płyty czołowej zestawu. Kolejnym ograniczeniem tego podejścia jest ograniczona możliwość sposobów zapisu oraz odczytu pamięci RAM oraz rejestru R_P . Zapis oraz odczyt musi się odbywać w sposób sekwencyjny co znacząco wydłuża proces w przypadku gdy zachodzi potrzeba zmiany lub odczytu wartości znajdujących się w dalszych regionach pamięci lub rejestru.

2.2.3 Platforma sprzętowa

W obecnej formie zestaw jest oparty o dwa mikrokontrolery Atmega 32 oraz Atmega 8. Układy te są odpowiedzialne za emulację części zestawu Pierwszy układ pełni rolę pamięci RAM, jednostki ALU,

rejestrów R_I , R_P oraz generatora GS. Podejście emulacyjne jest po części wymuszone ograniczoną dostępnością w handlu dedykowanych układów scalonych takich jak jednostki ALU ze względu na ich znikomą popularność nowych projektach. Drugi układ pełni rolę dekodera sygnałów sterujących. Pozostałe rejestry znajdujące się w zestawie zrealizowane zostały za pomocą układów LS74125 - cztery przerzutniki typu D. Komponenty zestawu wymieniają między sobą daną przy pomocy fizycznej szyny danych, wymusza to zastosowanie układów buforów trójstanowych na wyjściach nadajników.

Obecne rozwiązanie sprzętowe powstało z myślą o zachowaniu kompatybilności z płytą czołową z pierwszej części zestawu [2]. Dużym problemem tego zestawu jest brak elastyczności w rozbudowie oraz naprawie błędów. Duża część elementów zestawu została zrealizowana na fizycznych układach scalonych co znacząco utrudnia modyfikacje projektu. Rozbicie komponentów emulowanych pomiędzy dwa mikrokontrolery również nie jest optymalnym rozwiązaniem gdyż komplikuje to proces rozwoju oprogramowania oraz naprawy potencjalnych błędów. Dodatkową wadą obecnej platformy sprzętowej jest trudność modyfikacji interfejsu użytkownika spowodowana ciasną integracją elementów płyty czołowej ze sposobem działania zestawu.

3 Koncepcja nowego stanowiska

3.1 Założenia projektowe

Po przeanalizowaniu poprzedniego zestawu laboratoryjnego wypracowano następujące założenia dla projektu nowego zestawu:

- Zestaw musi zachować bloki funkcjonalne oraz zasadę działania poprzedniego zestawu.
- Zestaw ma zachować tryby pracy: cykl, mikrocykl oraz program.
- Zestaw musi emulować komponenty rzeczywistego układu szyny danych.
- Zestaw musi umożliwiać łatwą naprawę błędów w jego funkcjonowaniu oraz rozbudowę o nowe funkcjonalności.
- Zestaw musi zachować przejrzysty sposób wizualizacji stanu poszczególnych komponentów.
- Zestaw musi posiadać uniwersalny interfejs użytkownika pozwalający na łatwą obsługę zestawu oraz późniejsze jego modyfikacje.

3.2 Projekt sprzętowy

3.2.1 Wybór mikrokontrolera

Ze względu na wybrane podejście symulacji programowej działania układu szyny danych, mikrokontroler jest centralnym elementem budowanego układu. W celu zapewnienia możliwości dalszej rozbudowy oraz wprowadzania poprawek do układu musi być to nowoczesna i łatwo dostępna platforma. Istniejąca wersja zestawu jest oparta o mikrokontroler Atmega 8 oraz Atmega 32. Są to urządzenia pracujące z częstotliwością 16 MHz oraz posiadające odpowiednio 1 Kb oraz 2 Kb pamięci RAM [3] [4]. Większość mikrokontrolerów dostępnych obecnie na rynku oferuje znacząco większe możliwości.

Jako mikrokontroler do budowy zestawu wybrano układ RP2040 firmy Raspberry Pi [5] obecny na płytce rozwojowej Raspberry Pi Pico. Jest to układ oparty o architekturę ARM Cortex M0+, posiada 264 Kb pamięci RAM oraz pracuje z częstotliwością 133 MHz. Układ posiada duży zapas zasobów sprzętowych co pozwoli na szerokie możliwości rozbudowy zestawu w przyszłości. Dodatkowo dostępność układu na płytce rozwojowej uprości proces budowy sprzętowej części zestawu gdyż płytka ta zawiera wszystkie elementy niezbędne do pracy mikrokontrolera.

3.2.2 Elementy interfejsu użytkownika

Interfejs użytkownika zestawu powinien umożliwiać łatwą obsługę zestawu oraz umożliwiać łatwe wprowadzenie zmian w jego strukturze. Takie wymagania najlepiej spełni interfejs oparty o wyświetlacz wyświetlający menu z opcjami do wyboru oraz interfejs do nawigacji pomiędzy nimi. Jako wyświetlacz zdecydowano się zastosować wyświetlacz znakowy LCD 20x4. Jest to wyświetlacz oparty o popularny układ sterujący HD44780 [6]. Dzięki prostocie obsługi oraz długiej obecności na rynku tego typu wyświetlacze są wspierane przez bardzo szeroką gamę oprogramowania.

W celu nawigacji po menu zdecydowano się zastosować enkoder kwadraturowy. Jest to element generujący impulsy na podstawie ruchu obrotowego. Pozwala on na stwierdzenie kierunku oraz prędkości obrotu. Tego typu urządzenia stosowane są w układach sterowania pracą silników lub jako elementy wejściowe interfejsu użytkownika. Enkodery stosowane jako element nawigacyjny

często posiadają wbudowany przycisk [7] dzięki czemu obracając enkoderem można wybierać opcje menu a naciskając go można potwierdzać wybór.

Dodatkowo w celu zapewnienia wizualizacji stanu elementów podczas pracy zestawu zdecydowano się ja w oryginale zastosować diody LED do wizualizacji stanu adresów oraz słowa pamięci RAM, stanu rejestru R_{WY} oraz wybranego rejestru (R_A , R_B , R_C).

Jako elementy do wprowadzania danych binarnych zdecydowano się zachować przełączniki jak w oryginalnym zestawie. Dla uproszczenia obsługi zestawu dane do pamięci RAM oraz rejestru R_P będą wprowadzane za pomocą jednego zestawu ośmiu przełączników natomiast dane użytkownika na szybę będą wprowadzana za pomocą osobnych czterech przełączników.

3.2.3 Dodatkowe peryferia

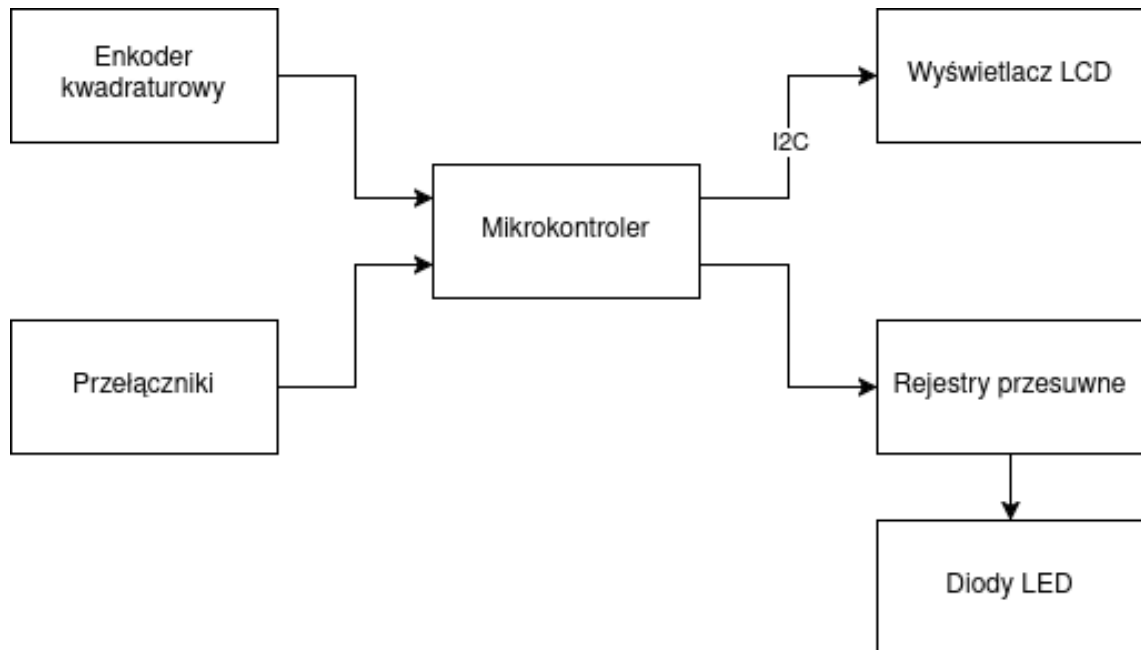
Sumując liczbę końcówek GPIO mikrokontrolera potrzebną do obsługi wybranych peryferiów otrzymujemy 43. Mikrokontroler RP2040 posiada 30 końcówek GPIO [5] co oznacza, że nie jest możliwe bezpośrednie podłączenie wszystkich peryferiów. W celu obsługi wszystkich peryferiów konieczne jest zastosowanie dodatkowych układów scalonych rozszerzających ilość dostępnych końcówek GPIO.

Jednym z prostszych tego typu układów jest rejestr przesuwany. Jest to układ który posiada wejście zegarowe oraz wejście danych. Po każdym zboczu zegara dane z wejścia danych są przesuwane do kolejnej komórki rejestru. Niektóre rejestry przesuwne posiadają wyjście danych z ostatniej komórki rejestru. Dzięki temu można połączyć kilka rejestrów kaskadowo. Jednym z takich układów jest 74HC595 [8]. Jest to 8 bitowy rejestr przesuwany z wyjściem równoległym. Rejestry te zastosowano do sterowania diodami LED. Użyto trzech układów ze względu na konieczność wysterowania 22 diod.

W celu redukcji ilości końcówek GPIO potrzebnych do obsługi wyświetlacza LCD zdecydowano się zastosować ekspander I/O PCF8574 [9]. Jest to układ który pozwala na sterowanie 8 bitowym portem wejścia/wyjścia za pomocą interfejsu I2C. Dzięki temu obsługa wyświetlacza zajmuje tylko dwie końcówki GPIO potrzebne do obsługi interfejsu I2C.

3.2.4 Schemat blokowy sprzętowej części zestawu

Na rysunku 3.1 przedstawiono schemat blokowy sprzętowej części zestawu.



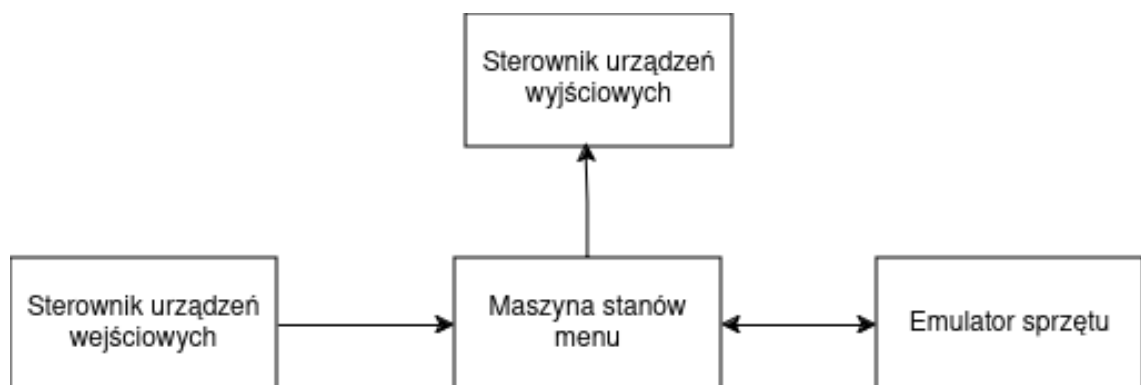
Rysunek 3.1: Schemat blokowy sprzętowej części zestawu

3.3 Architektura oprogramowania

3.3.1 Struktura programu

Wybrany mikrokontroler RP2040 posiada dostarczane przez producenta SDK dla języka C/C++. W skład zestawu wchodzi szereg bibliotek pozwalających na obsługę peryferiów mikrokontrolera. Są to biblioteki niskopoziomowe o niewielkim stopniu abstrakcji. Znaczącym ograniczeniem SDK jest niewielka ilość bibliotek do obsługi urządzeń dołączanych do mikrokontrolera. W związku z tym do napisania oprogramowania zestawu zdecydowano się wykorzystać framework Arduino [10]. Jest to popularne środowisko programistyczne wspierające wiele mikrokontrolerów oparte o język C++. Framework ten dostarcza wysokopoziomowe biblioteki do obsługi peryferiów mikrokontrolera. Ze względu na popularność Arduino framework dostępnych jest wiele bibliotek z nim kompatybilnych napisanych przez społeczność.

Oprogramowanie zestawu podzielono na kilka modułów odpowiadających za poszczególne funkcjonalności zestawu:



Rysunek 3.2: Schemat blokowy oprogramowania zestawu

3.3.2 Emulacja elementów układu szyny danych

Wybrane podejście symulacji programowej układu szyny danych wymaga zaimplementowania emulacji poszczególnych elementów układu:

- Pamięci RAM
- Jednostki ALU wraz z komponentami pomocniczymi
- Mechanizmy sterujące pracą układu
- Rejestru R_{WY}
- Rejestru R_P
- Rejestru R_A
- Rejestru R_B
- Rejestru R_C
- Rejestru R_I

Rejestry oraz pamięć RAM są strukturami przechowującymi dane. Ich implementacje programowo można rozwiązać stosując zmienne które podobnie jak rejestry przechowują dane. Pamięć RAM oraz R_P przechowują więcej niż jedną wartość dlatego ich reprezentacje w programie muszą być tablicami zmiennych o odpowiedniej długości. Symulacja jednostki arytmetyczno-logicznej wymaga implementacji operacji przez nią wykonywanych 5. Operacje te można zaimplementować z użyciem operatorów logicznych oraz arytmetycznych obecnych w języku C++. Kolejną częścią jednostki ALU jest system dekodowania słowa sterującego wyborem operacji oraz sposób pobierania wartości słowa sterującego z kolejnych instrukcji programu. Implementacja tego elementu wymaga zaimplementowania funkcji dekodującej. Ostatnią częścią konieczną do zaimplementowania jest system sterujący pracą układu. Kontrola pracy w pojedynczym mikrocyklu polega na zdekodowaniu pojedynczej instrukcji oraz wykonania na jej podstawie odpowiedniej operacji. Sterowanie pracą układu w trybach program oraz cykl można oprzeć na maszynie stanów która na podstawie aktualnego stanu układu oraz dokonuje odpowiednich przejść pomiędzy stanami.

3.3.3 Obsługa peryferiów sprzętowych

Oprogramowanie układu musi obsługiwać następujące peryferia:

- Wyświetlacz LCD
- Enkoder kwadraturowy
- Przełączniki
- Układy 74HC595 - diod LED

Przełączniki są podłączone bezpośrednio do końcówek GPIO mikrokontrolera więc ich status można odczytać za pomocą funkcji dostarczonych przez framework Arduino. Diody LED są sterowane za pomocą układów 74HC595. Sterowanie tymi układami wymaga generacji sygnału zegarowego oraz wystawiania kolejnych bitów danych na wejście danych układu zgodnie z taktowaniem zegara.

Zdecydowano się użyć biblioteki implementującej ten protokół w środowisku Arduino framework [11].

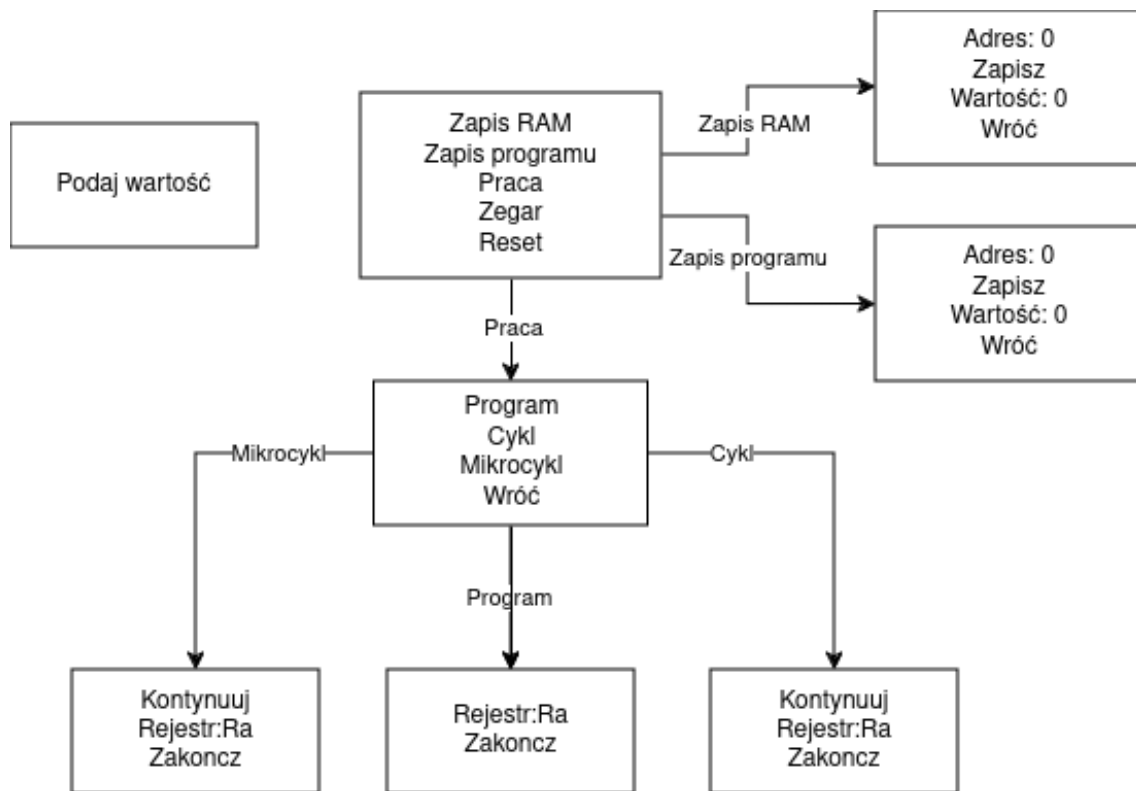
Obsługa wyświetlacza LCD odbywa się poprzez ekspander I/O PCF8574. Komunikacja z ekspanderem odbywa się za pomocą interfejsu I2C. Do obsługi wyświetlacza wykorzystano bibliotekę kompatybilną z frameworkiem Arduino - LiquidCrystal PCF8574 [12]. Enkoder kwadraturowy jest obsługiwany za pomocą biblioteki wybranej do implementacji systemu menu opisanej w dalszej części dokumentu.

3.3.4 Obsługa interfejsu użytkownika

Interfejs użytkownika powinien udostępniać użytkownikowi następujące funkcje:

- Wybór trybu pracy zestawu
- Zapis oraz odczyt RAM oraz R_P
- Wprowadzanie danych na szynę
- Wizualizacja stanu elementów układu
- Wybór aktualnie wyświetlanego rejestru

Wizualizacja stanów elementów układów jest realizowana za pomocą diod LED, pozostałe funkcjonalności należy zaimplementować w oparciu o wyświetlacz LCD oraz enkoder. Zdecydowano się zastosować interfejs użytkownika w formie menu. Menu jest to element interfejsu użytkownika składający się z szeregu opcji pomiędzy którymi użytkownik może nawigować. Opcje te po wybraniu mogą prowadzić do kolejnych poziomów menu lub wykonywać konkretne akcje. W tym przypadku do menu będzie wyświetlane na wyświetlaczu LCD a nawigacja będzie odbywać się poprzez obrót oraz naciśnięcie enkodera. Na rysunku 3.3 przedstawiono strukturę menu dla projektowanego zestawu.



Rysunek 3.3: Diagram menu interfejsu użytkownika

Pierwszym menu widocznym dla użytkownika jest ogólny ekran nawigacyjny prowadzący do pozostałych opcji. Z jego poziomu można przejść do zapisu/odczytu pamięci RAM oraz R_P , wyboru trybu pracy zestawu oraz zresetować zestaw. Z poziomu ekranu wyboru trybu pracy można rozpocząć pracę zestawu w jednym z trzech dostępnych trybów. Podczas pracy w dowolnym trybie użytkownik ma możliwość natychmiastowego zakończenia pracy zestawu oraz wyboru aktualnie wyświetlanego rejestru. W trybach cykl oraz mikrocykl znajduje się dodatkowa opcjami "Kontynuuj" po wciśnięciu której zestaw wykonuje kolejny cykl/mikrocykl. Ostatni dostępny ekran jest wywoływany w przypadku gdy program działający na zestawie w danej instrukcji czyta dane z klawiatury "Dane". Program jest zatrzymywany a użytkownik jest proszony o wprowadzenie danych. Po zatwierdzeniu przez użytkownika zestaw kontynuuje pracę.

Do programowej implementacji systemu menu zdecydowano się wykorzystać bibliotekę Arduino menu [13]. Jest to biblioteka pozwalająca na implementację hierarchicznego systemu menu w prosty sposób. Posiada ona szereg wbudowanych prymitywów pozwalających na łatwą implementację opcji menu pełniących funkcje takie jak: wywołanie przypisanej funkcji, zmiana wartości zmiennej lub przejście do kolejnego poziomu menu. Dużą zaletą tej biblioteki są wbudowane sterowniki dla wielu urządzeń wejścia oraz wyjścia. Biblioteka bez natywnie wspiera obsługę enkodera kwadraturowego oraz wyświetlacza LCD za pomocą biblioteki LiquidCrystal PCF8574 [12].

4 Implementacja stanowiska

4.1 Część sprzętowa

4.2 Część programowa

5 Testy stanowiska

Opis testów przeprowadzonych na stanowisku. Przykłady oraz analiza użytych programów testowych.

6 Podsumowanie

Podsumowanie rezultatów pracy, wnioski oraz możliwości dalszego rozwoju projektu.

Bibliografia

- [1] Mariusz Szostek. *Sterowanie szyną danych*.
- [2] Mariusz Szostek. “Sterowanie szyną danych”. Prac. mag. Politechnika Gdańska, 2012.
- [3] *Atmega 8*. Rev.2486AA–AVR–02/2013. Atmel Corporation. Lut. 2013. URL: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2486-8-bit-AVR-microcontroller-ATmega8_L_datasheet.pdf.
- [4] *Atmega 32*. 2503Q–AVR–02/11. Atmel Corporation. Lut. 2011. URL: <https://ww1.microchip.com/downloads/en/DeviceDoc/doc2503.pdf>.
- [5] *RP2040*. Ver. eec2b0c-clean. Raspberry Pi Ltd. Paź. 2024. URL: <https://ww1.microchip.com/downloads/en/DeviceDoc/doc2503.pdf>.
- [6] *HD44780 LCD Starter Guide*. URL: https://www-leland.stanford.edu/class/ee281/handouts/lcd_tutorial.pdf (term. wiz. 08.01.2025).
- [7] *Rotary Encoder*. URL: <https://electropeak.com/learn/rotary-encoder-how-it-works-how-to-use-with-arduino/> (term. wiz. 08.01.2025).
- [8] *SN74HC595*. Texas Instruments. Paź. 2021. URL: <https://www.ti.com/lit/ds/symlink/sn74hc595.pdf>.
- [9] *PCF8574*. Texas Instruments. Wrz. 2024. URL: <https://www.ti.com/lit/ds/symlink/pcf8574.pdf>.
- [10] *Arduino framework for RP2040*. URL: <https://github.com/earlephilhower/arduino-pico> (term. wiz. 08.01.2025).
- [11] *Biblioteka Arduino 74HC595*. URL: <https://github.com/Simssso/ShiftRegister74HC595> (term. wiz. 08.01.2025).
- [12] *Biblioteka LCD PCF8574*. URL: https://github.com/mathertel/LiquidCrystal_PCF8574 (term. wiz. 08.01.2025).
- [13] *Biblioteka Arduino menu*. URL: <https://github.com/neu-rah/ArduinoMenu> (term. wiz. 08.01.2025).

Spis rysunków

2.1	Płyta czołowa zestawu	8
3.1	Schemat blokowy sprzętowej części zestawu	17
3.2	Schemat blokowy oprogramowania zestawu	17
3.3	Diagram menu interfejsu użytkownika	20

Spis tabel

1	Symbole przypisane poszczególnym bitom słowa w pamięci RAM	9
2	Tablica prawdy dla dekodera 1	9
3	Tablica prawdy dla dekodera 2	10
4	Zasada kodowania operacji do wykonania przez ALU	10
5	Lista operacji realizowanych przez ALU	11