

## **PROJEKT I OPTIMALIZACJA STRUKTURY LOGICZNEJ DYDAKTYCZNEGO SYSTEMU MIKROPROCESOROWEGO DLA LABORATORIUM PROJEKTOWANIA ZINTEGROWANEGO**

**Sebastian Pawlak**

**Instytut Informatyki i Elektroniki, Uniwersytet Zielonogórski  
65-246 Zielona Góra, ul. Podgórna 50**

*e-mail: S.Pawlak@iie.uz.zgora.pl*

### **STRESZCZENIE**

W artykule opisano projekt i proces optymalizacji struktury logicznej systemu mikroprocesorowego, zaprojektowanego i wykorzystywanego do celów dydaktycznych w laboratorium Instytutu Informatyki i Elektroniki Uniwersytetu Zielonogórskiego.

### **1. WPROWADZENIE**

Najatrakcyjniejszą i najwydajniejszą metodą zdobywania wiedzy jest nauka poprzez doświadczenia praktyczne. Umożliwienie studentom pracy z rzeczywistymi (nie wirtualnymi) modelami powinno znacznie zwiększyć jakość kształcenia.

W tym celu został zaprojektowany system mikroprocesorowy integrujący mikrokontroler jednoukładowy i układ logiki programowalnej CPLD (Complex Programmable Logic Device).

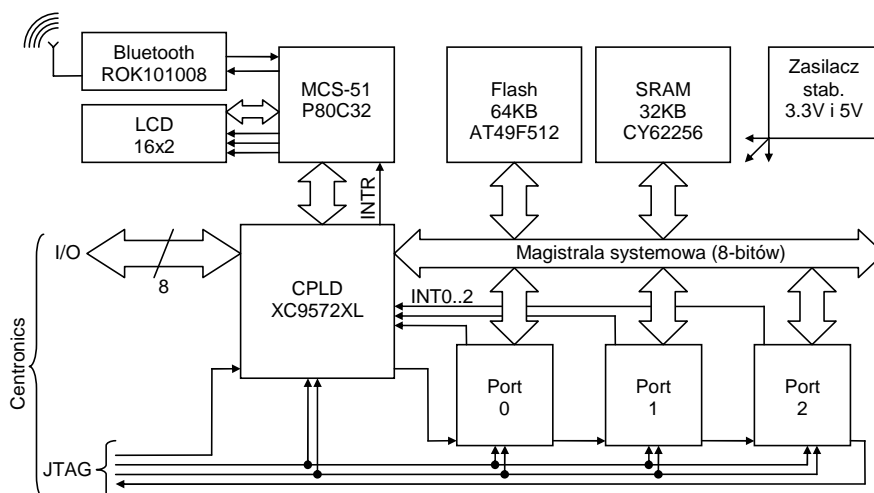
### **2. ARCHITEKTURA SYSTEMU**

System został zaprojektowany jako system modułowy. W podstawowej wersji składa się z mikrokontrolera, układu CPLD, 64KB pamięci programu i 32KB pamięci danych. Płyta główna wyposażona jest w trzy złącza rozszerzeń ogólnego przeznaczenia, podstawkę pod moduł komunikacji bezprzewodowej w standardzie Bluetooth [1] oraz port dla wyświetlacza alfanumerycznego LCD. Schemat blokowy urządzenia przedstawiono na rys. 1.

#### **2.1. Mikrokontroler**

Wybrano układ 80C32 z rodziny Intel MCS-51 [2]. Układ ten jest bardzo popularny i jest jednym z najtańszych na rynku. Zastosowanie zegara 11.0592MHz pozwala na uzyskanie synchronizacji wbudowanego w mikrokontroler portu szeregowego RS232 z wykorzystanym w systemie modułem Bluetooth. Magistralę danych, adresową i sterującą mikrokontrolera

podłączono bezpośrednio do układu CPLD, który pełni rolę mostka do magistrali systemowej. Jeden z portów mikrokontrolera zarezerwowano dla wyświetlacza alfanumerycznego LCD.



Rys. 1. Schemat blokowy systemu mikroprocesorowego

## 2.2. Układ CPLD

Układ CPLD pełni rolę mostka pomiędzy mikrokontrolerem, a magistralą systemową. Konieczne jest zaimplementowanie w nim dekodera adresu, kontrolera przerwań oraz układu sterowania zapisem do pamięci FLASH (w celu rekonfiguracji części programowej).

Zastosowano układ XC9572XL-7TQ100C firmy Xilinx [3].

## 2.3. Pamięć programu

Jako pamięć programu zastosowano układ Atmel AT49F512-55J [4]. Jest to 8-bitowa pamięć FLASH o pojemności 64KB. Układ zasilany jest pojedynczym napięciem 5V i posiada wbudowany generator napięcia programującego.

## 2.4. Pamięć danych

Jako pamięć danych zastosowano układ Cypress CY62256L [5]. Jest to 8-bitowa pamięć SRAM o pojemności 32KB. Pozostałe 32KB z przestrzeni adresowej mikrokontrolera zarezerwowano dla kart rozszerzeń (Port0..2).

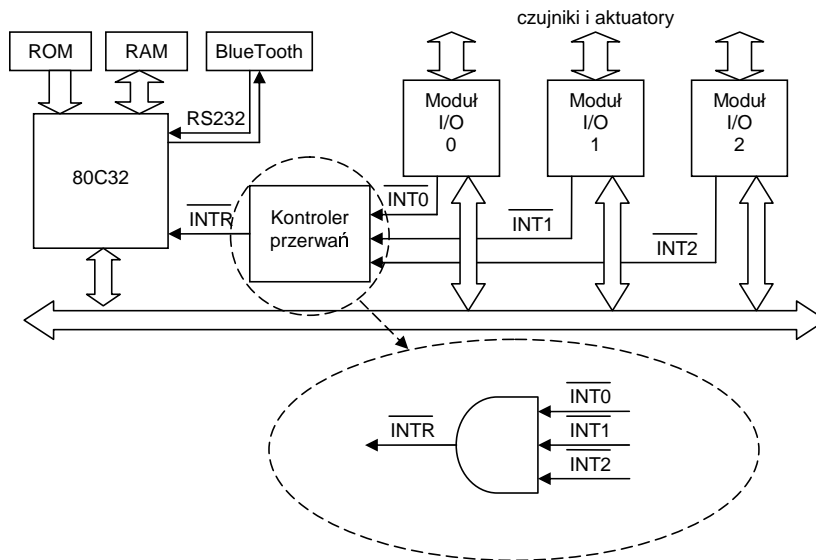
## 2.5. Moduł komunikacji bezprzewodowej Bluetooth

Zastosowano moduł ROK 101 008 firmy Ericsson [6].

## 2.6. Układ zasilania

Układ zasilania został zaprojektowany w sposób umożliwiający zasilanie napięciem stałym lub przemiennym o napięciu 7-24V.

### 3. STRUKTURA LOGICZNA SYSTEMU



Rys. 2. Struktura logiczna systemu

Opisywany system mikroprocesorowy wykorzystywany jest jako jednostka centralna, sterująca elementami wykonawczymi robota. System poprzez szereg czujników, takich jak: czujniki stykowe, optoelektroniczne czujniki przemieszczenia/obrotu oraz czujniki zbliżeniowe czerpie informacje o otoczeniu i po ich analizie podejmuje decyzje, w jaki sposób sterować aktuatorami (np. silnikami krokowymi).

Na rys. 2. przedstawiono strukturę logiczną systemu. Kontroler przerwań realizuje operację iloczynu logicznego trzech sygnałów przerwań z portów rozszerzeń.

### 4. OPTIMALIZACJA STRUKTURY SYSTEMU

#### 4.1. Kryterium optymalizacji

Podstawowym kryterium optymalizacji jest czas reakcji na zgłoszenie przerwania zewnętrznego.

#### 4.2. Zakres optymalizacji

Zakres optymalizacji obejmuje modyfikację struktury kontrolera przerwań i podprogramu jego obsługi.

#### 4.3. Analiza parametrów

Mikrokontroler pracuje z częstotliwością 11,0592MHz. Na cykl maszynowy składa się 12 taktów zegara, co daje czas równy 1,085µs.

Poniżej przedstawiono program obsługi przerw zewnętrznych przed optymalizacją:

```

.org    0000h
ljmp   START
.org    0003h
ljmp   INT0           ;2 cykle
.org    26h
START:
ljmp   START
INT0:
mov    DPTR,#INT0_BASE ;2 cykle
movx   A,@DPTR         ;2 cykle
jz     INT1           ;2 cykle
lcall  INT0_SERVICE   ;2 cykle
reti                   ;2 cykle
INT1:
mov    DPTR,#INT1_BASE ;2 cykle
movx   A,@DPTR         ;2 cykle
jz     INT2           ;2 cykle
lcall  INT1_SERVICE   ;2 cykle
reti                   ;2 cykle
INT2:
mov    DPTR,#INT2_BASE ;2 cykle
movx   A,@DPTR         ;2 cykle
jz     RETURN         ;2 cykle
lcall  INT2_SERVICE   ;2 cykle
RETURN:
reti                   ;2 cykle
.end

```

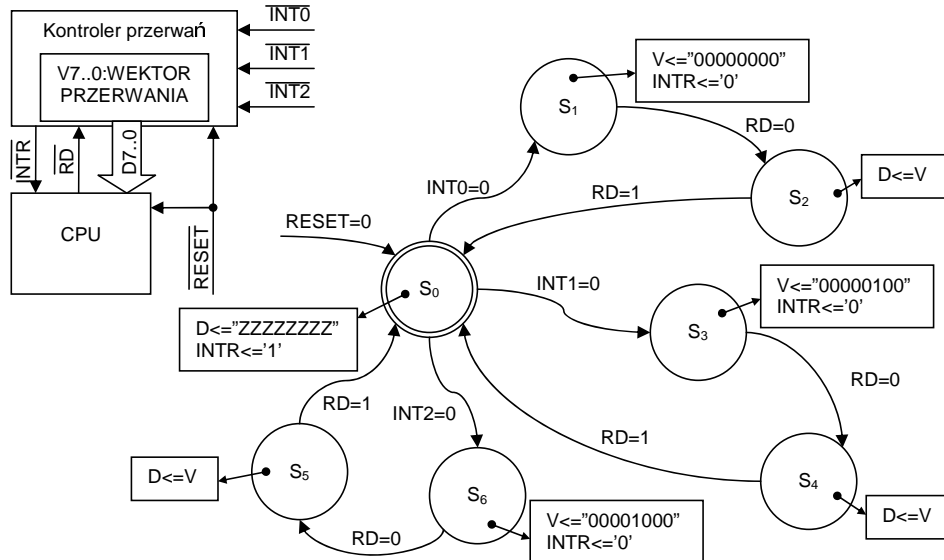
Ponieważ zastosowano bardzo prosty kontroler przerw, konieczne jest przeglądanie statusu portów, w celu określenia, z którego portu nadeszło żądanie przerwania. Czasy reakcji uzależnione są od numeru analizowanego przerwania i narastają dla każdego kolejnego przerwania o 6 cykli maszynowych, czyli 6,51µs.

Czasy reakcji na zgłoszenia poszczególnych przerw przedstawiono w tabeli 1.

Tab.1. Czas reakcji na zgłoszenie przerwania

Nr przerwania	Czas reakcji	
	[cykle]	[µs]
0	10	10,85
1	16	17,36
2	22	23,87

#### 4.4. Propozycja rozwiązania optymalnego



Rys. 3. Automat stanu proponowanego kontrolera przerwania

Zastosowanie kontrolera przerwania przedstawionego na rys. 3. umożliwi zoptymalizowanie podprogramu obsługi żądania przerwania:

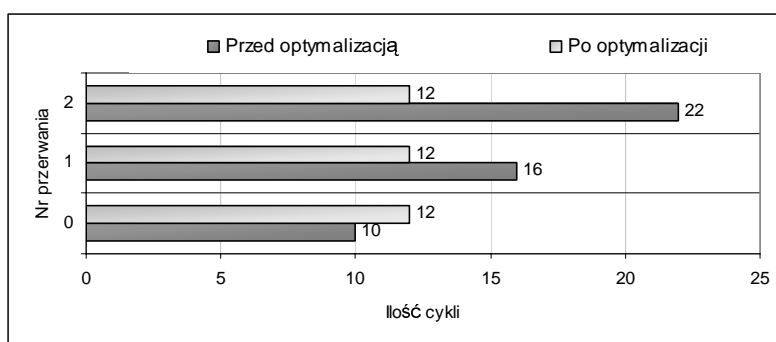
```
.org 0000h
ljmp START
.org 0003h
ljmp INT_EXT ;2 cykle
.org 26h
START: ljmp START
INT_EXT:
mov DPTR,#0000h ;2 cykle
movx A,@DPTR ;2 cykle
mov DPTR,#WYWOLANIA ;2 cykle
jmp @A+DPTR ;2 cykle
WYWOLANIA:
lcall INT0_SERVICE ;2 cykle
reti
lcall INT1_SERVICE ;2 cykle
reti
lcall INT2_SERVICE ;2 cykle
reti
.end
```

W tabeli 2 przedstawiono czasy reakcji na żądanie przerwania, uzyskane po optymalizacji.

Tab.2. Czas reakcji na zgłoszenie przerwania po optymalizacji

Nr przerwania	Czas reakcji		Zmiana
	[cykle]	[ $\mu$ s]	
0	12	13,02	20%
1	12	13,02	-25%
2	12	13,02	-45%

Na rys. 4. przedstawiono wykres porównawczy.



Rys. 4. Porównanie czasów reakcji przed i po optymalizacji

## 5. WNIOSKI

Przeprowadzona optymalizacja doprowadziła w przypadku przerwania INT0 do wydłużenia czasu reakcji o 20%, jednakże średni czas reakcji zmniejszył się o 25%. Dodatkowo osiągnięto jednakowe czasy reakcji na wszystkie przerwy zewnętrzne. Opracowany system znalazł zastosowanie praktyczne w dydaktyce Instytutu. Planuje się dalszą jego modyfikację przez zastosowanie wielopoziomowego, priorytetowego kontrolera przerw i jego optymalizację.

## LITERATURA

- [1] Bluetooth Special Interest Group: *Specification of the Bluetooth System*, Bluetooth Special Interest Group, 2001
- [2] Philips Semiconductors: *80C31/80C32 DataSheet*, Philips Semiconductors, 2000
- [3] Xilinx: *XC9572XL High Performance CPLD*, Xilinx, 2004
- [4] Atmel: *AT49F512 512K (64K x 8) 5-volt Only FLASH Memory*, Atmel Corporation, 2004
- [5] Cypress: *CY62256 256K (32K x 8) Static RAM*, Cypress, 2002
- [6] Ericsson Microelectronics: *ROK 101 008 Bluetooth Module*, Ericsson Microelectronics, 2001