



Akademia Górniczo-  
Hutnicza  
w Krakowie  
Katedra Elektroniki  
WIET

---



## **Laboratorium TM2**

### **Ćwiczenie 7a**

#### **Port I<sup>2</sup>C i wyświetlacz LCD - dodatek**

Autor: Mariusz Sokołowski

wer. 26.11.2020

# 1. WSTĘP

---

## 1.1.CEL

Celem ćwiczenia jest:

- ✚ zapoznanie studenta z techniką poprawnej inicjalizacji i obsługi portu I<sup>2</sup>C,
- ✚ zapoznanie studenta z zasadami współpracy portu szeregowego z portem równoległym, na przykładzie połączenia portu I<sup>2</sup>C z ośmiobitowym ekspanderem, realizującym jednocześnie pseudo dwukierunkową szynę danych, jak i szynę sterującą,
- ✚ poznanie metody projektowania i programowania polskich liter w generatorze znaków, wyświetlacza LCD.

## 1.2.WYMAGANIA

Sprzętowe:

- komputer klasy PC, spełniający wymagania sprzętowe aplikacji KEIL v5,
- zestaw FRDM-KL05Z

Programowe:

- system operacyjny Windows 7 lub wyższy,
- środowisko Keil / uVision 5 MDK-ARM

Doświadczenie:

- podstawowa umiejętność obsługi komputera klasy IBM-PC,
- podstawowa znajomość systemów operacyjnych rodziny Windows,
- podstawowa znajomość zagadnień z Techniki Mikroprocesorowej 1
- podstawowa znajomość zagadnień z Techniki Cyfrowej
- poprzednie tematy laboratorium Techniki Mikroprocesorowej 2

Literatura:

- KL05 Sub-Family Reference Manual, Freescale Semiconductor
- Kinetis L Peripheral Module Quick Reference, Freescale Semiconductor
- PCF8574 - Remote 8-bit I/O expander for I2C-bus, Data Sheet, PHILIPS
- HD44780U (LCD-II) – Dot Matrix Liquid Crystal Display Controller/Driver, Data Sheet, Hitachi

## 2. MODUŁ PORTU SZEREGOWEGO I<sup>2</sup>C

---

Układ MKL05Z32VLC4, będący centralnym mikrokontrolerem zestawu FRDM-KL05Z, posiada w swoich zasobach sprzętowych, jeden port szeregowy I<sup>2</sup>C – I2C0.

Niniejszy port spełnia warunki specyfikacji, zdefiniowane dla magistrali I<sup>2</sup>C.

Aby przygotować układ I2C0 do pracy, należy wykonać następujące czynności:

- ❖ Sprawdzić, jaką wartość ma stała CLOCK\_SETUP, w zbiorze *system\_MKL05Z4.c*. Informacja ta będzie miała wpływ na ustawienia rejestrów, odpowiedzialnych za szybkość transmisji. W zależności od stałej CLOCK\_SETUP, parametry podstawowych sygnałów zegarowych mają następujące wartości:
  - CLOCK\_SETUP=0 (wartość domyślna):
    - zegar referencyjny dla modułu MCG - 32768Hz,
    - Core clock - 41943040Hz,
    - BusClock - 20971520Hz.
  - CLOCK\_SETUP=1:
    - zegar referencyjny dla modułu MCG - 32768Hz,
    - Core clock - 47972352Hz,
    - BusClock - 23986176Hz.
  - CLOCK\_SETUP=2:
    - zegar referencyjny dla modułu MCG – 4MHz,
    - Core clock – 4MHz,
    - BusClock – 2MHz.
- ❖ dołączyć sygnał taktujący do modułu I2C0, w rejestrze SIM\_SCGC4 [I2C0=1],
- ❖ dołączyć sygnał taktujący do odpowiedniego portu, którego końcówki realizują funkcje SCL (PTB3) i SDA (PTB4), w rejestrze SIM\_SCGC5[PORTB=1],
- ❖ ustawić odpowiednią funkcję dla wykorzystywanych końcówek portu, w rejestrze PORTB\_PCRx[MUX=2]. PTB3 – zegar SCL, PTB4 – dane SDA,
- ❖ w rejestrze I2C0\_F, ustawić wartości pól MULT i ICR tak, aby dla danej wartości częstotliwości BusClock zrealizować założoną częstotliwość transferu BR. Wartość ICR wybieramy z tabeli Tab. 1, po uprzednim wyliczeniu wartości „SCL divider”, wg poniższego wzoru:

$$SCL\ divider = \frac{BusClock}{mul \cdot BR} [Hz]$$

Wartość pola MULT jest związana z czynnikiem *mul*, poniższą zależnością:

mul=1     MULT=00

mul=2     MULT=01

mul=4     MULT=10

- ❖ w przypadku wykorzystywania przerwań, odblokować przerwania od portu I2C0, w rejestrze I2C0\_C1[IICIE=1].

W tym momencie układ I2C0 jest gotowy do pracy.

ICR (hex)	SCL divider	SDA hold value	SCL hold (start) value	SCL hold (stop) value		ICR (hex)	SCL divider (clocks)	SDA hold (clocks)	SCL hold (start) value	SCL hold (stop) value
00	20	7	6	11		20	160	17	78	81
01	22	7	7	12		21	192	17	94	97
02	24	8	8	13		22	224	33	110	113
03	26	8	9	14		23	256	33	126	129
04	28	9	10	15		24	288	49	142	145
05	30	9	11	16		25	320	49	158	161
06	34	10	13	18		26	384	65	190	193
07	40	10	16	21		27	480	65	238	241
08	28	7	10	15		28	320	33	158	161
09	32	7	12	17		29	384	33	190	193
0A	36	9	14	19		2A	448	65	222	225
0B	40	9	16	21		2B	512	65	254	257
0C	44	11	18	23		2C	576	97	286	289
0D	48	11	20	25		2D	640	97	318	321
0E	56	13	24	29		2E	768	129	382	385
0F	68	13	30	35		2F	960	129	478	481
10	48	9	18	25		30	640	65	318	321
11	56	9	22	29		31	768	65	382	385
12	64	13	26	33		32	896	129	446	449
13	72	13	30	37		33	1024	129	510	513
14	80	17	34	41		34	1152	193	574	577
15	88	17	38	45		35	1280	193	638	641
16	104	21	46	53		36	1536	257	766	769
17	128	21	58	65		37	1920	257	958	961
18	80	9	38	41		38	1280	129	638	641
19	96	9	46	49		39	1536	129	766	769
1A	112	17	54	57		3A	1792	257	894	897
1B	128	17	62	65		3B	2048	257	1022	1025
1C	144	25	70	73		3C	2304	385	1150	1153
1D	160	25	78	81		3D	2560	385	1278	1281
1E	192	33	94	97		3E	3072	513	1534	1537
1F	240	33	118	121		3F	3840	513	1918	1921

Tab. 1

### 3. TRANSFER MIĘDZY „MASTER” A „SLAVE”

Ponieważ cała instrukcja będzie się odnosiła do wyświetlacza LCD, podłączonego do mikrokontrolera za pośrednictwem 8-bitowego ekspandera portu I<sup>2</sup>C, dlatego omówione zostaną transfery tylko jednobajtowych danych. Również, ze względu na fakt, że ekspander jest pojedynczym rejestrem, nie będą omawiane przypadki adresacji rejestrowej.

Definiując makra lub funkcje, realizujące poszczególne etapy transferu, można, jak z klocków, „poskładać” każdy rodzaj transferu. Poniżej przedstawiono definicje w/w makr (w bibliotece *i2c.c* makra są zastąpione funkcjami).

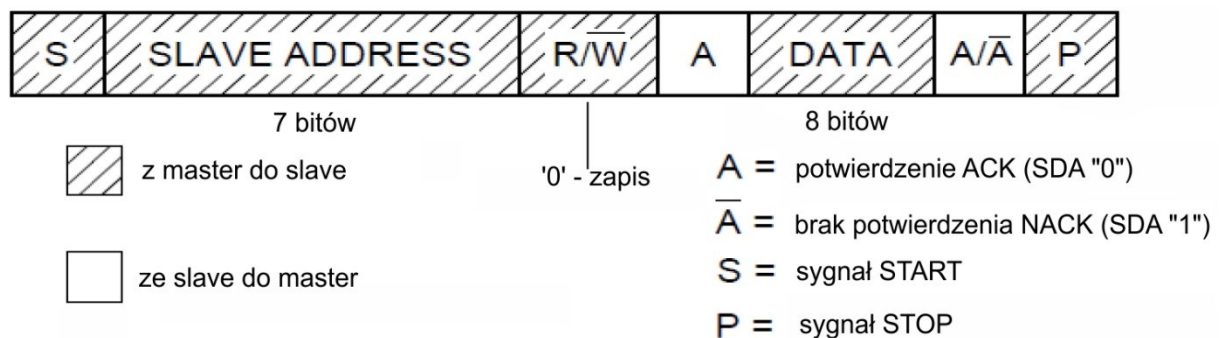
```

#define I2C_ENABLE I2C0->C1 |= I2C_C1_IICEN_MASK; – aktywowanie portu I2C0
#define I2C_DISABLE I2C0->C1 &= (~I2C_C1_IICEN_MASK); – deaktywowanie portu I2C0
#define I2C_M_START I2C0->C1 |= I2C_C1_MST_MASK; – sygnał START
#define I2C_M_STOP I2C0->C1 &= ~I2C_C1_MST_MASK; – sygnał STOP
#define I2C_TRAN I2C0->C1 |= I2C_C1_TX_MASK; – transfer z „master” do „slave”
#define I2C_REC I2C0->C1 &= ~I2C_C1_TX_MASK; – transfer ze „slave” do „master”
#define I2C_SEND(data) I2C0->D=data; – wysłanie danej z „master” do „slave”
#define I2C_READ I2C0->D; – odczyt danej ze „slave” do „master”
#define I2C_WAIT while((I2C0->S & I2C_S_IICIF_MASK)==0){ \
    I2C0->S |= I2C_S_IICIF_MASK; – oczekiwanie na koniec transferu i potwierdzenie
#define NACK I2C0->C1 |= I2C_C1_TXAK_MASK; – wyślij NACK po odczycie (od „master”)
#define ACK I2C0->C1 &= ~I2C_C1_TXAK_MASK; – wyślij ACK po odczycie (od „master”)

```

## 4. ZAPIS I<sup>2</sup>C

Na rysunku Rys. 1 jest pokazany, w uproszczonej formie, protokół zapisu danej, do urządzenia typu „slave” przez urządzenie typu „master”.



Rys. 1 Zapis portu I<sup>2</sup>C

Wykorzystując wspomniane wcześniej „klocki”, graficzny zapis z rysunku Rys. 1 można zapisać w postaci poniższego podprogramu:

```

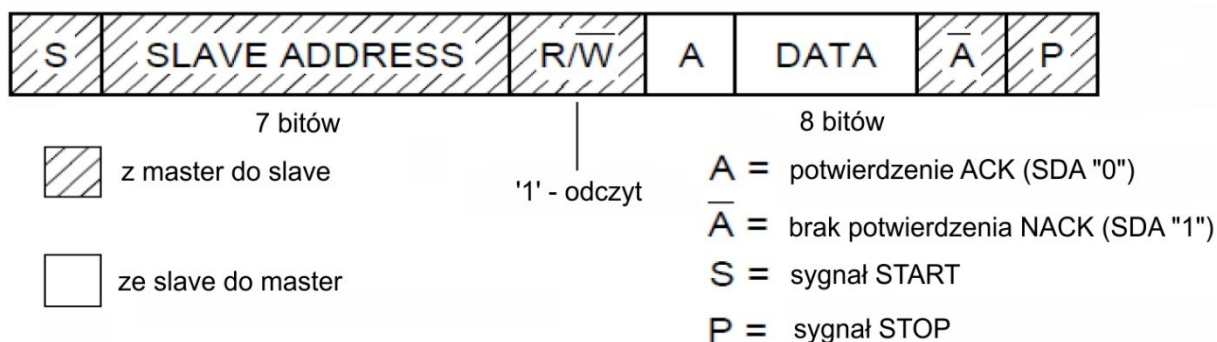
I2C_ENABLE
I2C_TRANS
I2C_M_START
I2C_SEND(address<<1) – najmłodszy bit równy „0”, oznaczający zapis
I2C_WAIT
I2C_SEND(data)
I2C_WAIT
I2C_M_STOP
I2C_DISABLE

```

Powyższy fragment programu wykonuje transfer pojedynczego bajta „data”, z urządzenia typu „master” do urządzenia typu „slave”, o adresie „address”.

## 5. ODCZYT I<sup>2</sup>C

Na rysunku Rys. 2 jest pokazany, w uproszczonej formie, protokół odczytu danej, z urządzenia typu „slave” przez urządzenie typu „master”.



Rys. 2 Odczyt portu I<sup>2</sup>C

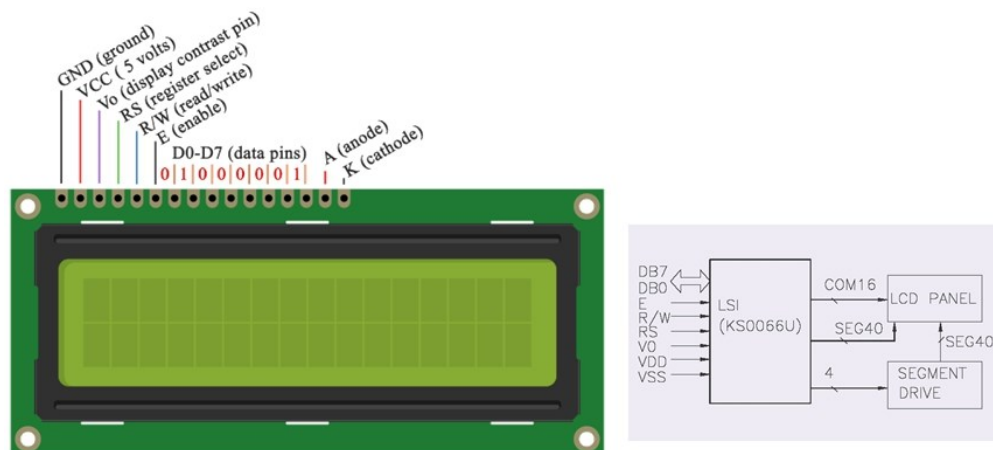
Wykorzystując wspomniane wcześniej „klocki”, graficzny zapis z rysunku Rys. 2 można zapisać w postaci poniższego podprogramu:

```
I2C_ENABLE
I2C_TRANS
I2C_M_START
I2C_SEND((address<<1)|0x1) – najmłodszy bit równy „1”, oznaczający odczyt
I2C_WAIT
I2C_REC
I2C_NACK
data= I2C_READ – „pusty” odczyt, inicjujący transfer danej ze „slave” do „master”
I2C_WAIT
I2C_M_STOP
data= I2C_READ – odczyt danej, bez inicjalizacji następnego transferu (był STOP)
I2C_DISABLE
```

Powyższy fragment programu wykonuje transfer pojedynczego bajta „data”, z urządzenia typu „slave”, o adresie „address”, do urządzenia typu „master”.

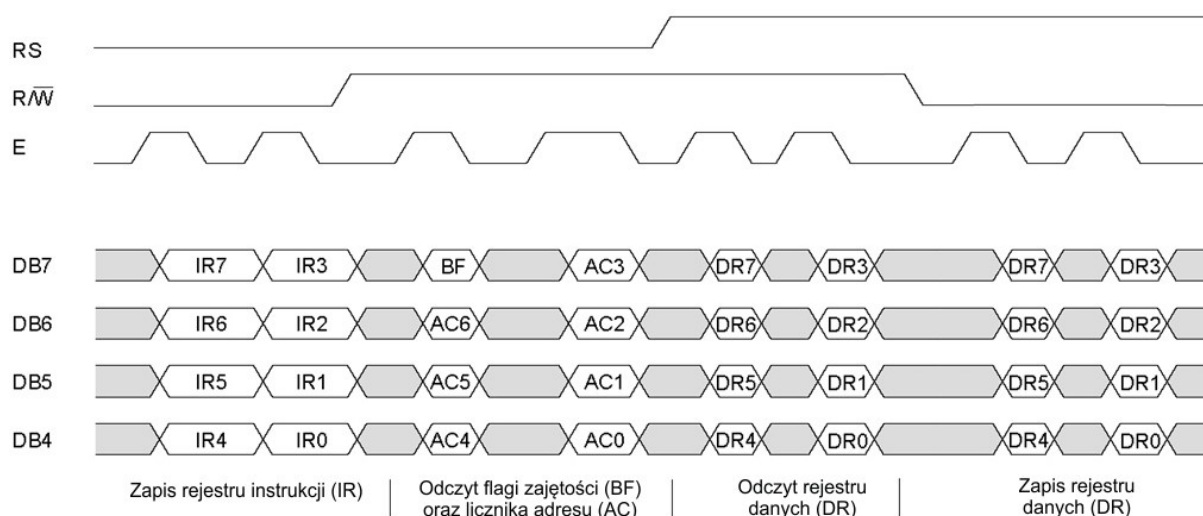
## 6. WYŚWIETLACZ LCD 1602A

Na rysunku Rys. 3 jest pokazany popularny, dwuwierszowy, 16-znakowy, alfanumeryczny wyświetlacz LCD, sterowany układem HD44780U.



Rys. 3 Moduł wyświetlacza LCD wraz ze schematem blokowym

Moduł posiada 8-bitową, dwukierunkową szynę danych DB0 ÷ DB7 oraz, uwzględniając jeszcze dodatkowy bit, sterujący podświetlaniem BL, 4-bitową szynę sterującą. Razem 12 bitów. W systemach takich, jak KL05Z, każda końcówka jest na wagę złota, a tu taka rozrzutność. Z tego też względu, można ograniczyć liczbę końcówek (wyświetlacza) do dwóch, kosztem szybkości. Wyświetlacz, jako wyjściowy interfejs komunikacji z użytkownikiem, nie musi być bardzo szybki. Redukcji dokonamy, stosując 8-bitowy ekspander I<sup>2</sup>C, który będzie pośredniczył pomiędzy portem szeregowym (I<sup>2</sup>C) a równoległym portem układu HD44780U. Układ ten ma możliwość ograniczenia długości swojej szyny danych do 4-ech bitów, co razem z bitami sterującymi daje liczbę 8-miu bitów, czyli dokładnie tyle, ile ma ekspander. Na rysunku Rys. 4 są zamieszczone przebiegi czasowe na magistralach sterownika LCD oraz ich wzajemne zależności (szyna danych 4-bitowa).



Rys. 4 Zależności czasowe między przebiegami na magistralach układu HD44780U



Analizując przebiegi z rysunku Rys. 4, widać, że linia RS zachowuje się jak linia adresująca, raz rejestr statusowo-sterujący, a raz rejestr danych. Linia R/W zachowuje się jak linia sterująca kierunkiem przepływu danych: „0” – zapis, „1” – odczyt. W końcu linia E zachowuje się jak sygnał zapisujący lub odczytujący. Przy zapisie, opadające zbocze „zatrzaskuje” daną w rejestrach lub pamięci układu. Natomiast przy odczycie, stan wysoki otwiera bufory wyjściowe rejestrów lub pamięci danych. W momencie ustawienia linii E w stan „0”, bufory te zamykają się i dana znika z wyjść układu. Obserwacja tego zjawiska będzie pomocna przy konstruowaniu algorytmu odczytu poprzez ekspander.

## 6.1. ROZKAZY STERUJĄCE I ODCZYTUJĄCO-ZAPISUJĄCE

Wszystkie dozwolone operacje (instrukcje), które może wykonać użytkownik, są zebrane w tabeli Tab. 2.

Instruction	Code										Description
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.
Return home	0	0	0	0	0	0	0	0	1	—	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.
Read busy flag & address	0	1	BF	AC	AC	AC	AC	AC	AC	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.
Write data to CG or DDRAM	1	0	Write data								Writes data into DDRAM or CGRAM.
Read data from CG or DDRAM	1	1	Read data								Reads data from DDRAM or CGRAM.

I/D = 1: Increment  
I/D = 0: Decrement  
S = 1: Accompanies display shift  
S/C = 1: Display shift  
S/C = 0: Cursor move  
R/L = 1: Shift to the right  
R/L = 0: Shift to the left  
DL = 1: 8 bits, DL = 0: 4 bits  
N = 1: 2 lines, N = 0: 1 line  
F = 1: 5 × 10 dots, F = 0: 5 × 8 dots  
BF = 1: Internally operating  
BF = 0: Instructions acceptable  
D = 1: Display on  
D = 0: Display off  
C = 1: Cursor on  
C = 0: Cursor off  
B = 1: Cursor position character is blinking  
B = 0: Cursor position character is not blinking

DDRAM: Display data RAM  
CGRAM: Character generator RAM  
ACG: CGRAM address  
ADD: DDRAM address  
(corresponds to cursor address)  
AC: Address counter used for both DD and CGRAM addresses

Tab. 2

Nierozłączną częścią każdej instrukcji jest stan linii RS i R/W. Istnieje tylko jedna instrukcja odczytu, w przestrzeni rejestrów sterujących. Jest nią instrukcja odczytu flagi zajętości (BF) i licznika adresu (AC). Jest to również jedyna instrukcja, którą można wykonywać „bezkarnie”, tzn. nie oglądając się na to, czy sterownik jest zajęty czy nie. Dzieje się tak dlatego, że tylko ta instrukcja dostarcza informacji na temat zajętości sterownika (flaga BF). Wszystkie pozostałe instrukcje mogą być wykonywane, pod warunkiem wcześniejszego sprawdzenia stanu flagi BF. Dopóki jej stan to „1”, nie należy wykonywać żadnej innej operacji.



## 6.2. GENERATOR ZNAKÓW

Generator znaków jest pamięcią, w której przechowywane są wzory bitowe, wszystkich dostępnych do wyświetlenia znaków. Oprócz wbudowanych znaków (pamięć EPROM), istnieje miejsce, typu RAM, dla ośmiu znaków, zdefiniowanych przez użytkownika. Wykorzystywany, w dotychczasowych projektach, wyświetlacz LCD-1602A posiada wbudowany format znaków 5x8 punktów, co oznacza, że na zobrazowanie jednego znaku potrzebnych jest 8 bajtów. Obszar, przeznaczony na znaki użytkownika, nazywa się CGRAM, zawiera 64 bajty i zaczyna się od adresu 00. Tylko trzy młodsze bity kodu znaku adresują CGRAM. Czwarty bit nie bierze udziału w adresacji, a pozostałe bity muszą mieć wartość „0”. Na rysunku Rys. 5 jest pokazany fragment pamięci generatora znaków, z zaznaczonym obszarem na definiowane znaki.

Lower 4 Bits \ Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111
xxxx0000	CG RAM (1)			0	a	P	`	P
xxxx0001	(2)		!	1	A	Q	a	q
xxxx0010	(3)		"	2	B	R	b	r
xxxx0011	(4)		#	3	C	S	c	s
xxxx0100	(5)		\$	4	D	T	d	t
xxxx0101	(6)		%	5	E	U	e	u

Rys. 5. Generator znaków

Na rysunkach Rys. 6 i Rys. 7 jest pokazana zależność pomiędzy kodem znaku a jego adresem w generatorze znaków. Kod znaku, wpisywany do pamięci ekranu DDRAM, jest tak naprawdę starszą częścią adresu, który wskazuje pozycję odpowiedniego znaku w pamięci generatora znaków. Pozostałe bity adresu są generowane przez układ sterujący i, w zależności od stanu starszego półbajta kodu znaku, przybierają następujące wartości:

- półbajt różny od zera – adres generowany tak, jak na rysunku Rys. 6, a znak pobierany jest z wbudowanego generatora znaków typu EPROM,
- półbajt równy zero – adres generowany tak, jak na rysunku Rys. 7, a znak pobierany jest z generatora znaków typu RAM (CGRAM użytkownika). Starsze trzy bity adresu (A5 ÷ A3) to młodsze trzy bity kodu znaku.

EPROM Address												Data								
A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	LSB								
												O4	O3	O2	O1	O0				
<div>01100010</div>												0	0	0	0	1	0	0	0	0
												0	0	0	1	1	0	0	0	0
												0	0	1	0	1	0	1	1	0
												0	0	1	1	1	1	0	0	1
												0	1	0	0	1	0	0	0	1
												0	1	0	1	1	0	0	0	1
												0	1	1	0	1	1	1	1	0
												0	1	1	1	0	0	0	0	0
<div></div>												1	0	0	0	0	0	0	0	0
												1	0	0	1	0	0	0	0	0
												1	0	1	0	0	0	0	0	0
												1	0	1	1	0	0	0	0	0
												1	1	0	0	0	0	0	0	0
												1	1	0	1	0	0	0	0	0
												1	1	1	0	0	0	0	0	0
												1	1	1	1	0	0	0	0	0

pozycja kursora

Rys. 6. Zależność pomiędzy kodem znaku a jego adresem – pamięć EPROM

Character Codes (DDRAM data)	CGRAM Address	Character Patterns (CGRAM data)
7 6 5 4 3 2 1 0 High A5 A4 A3 Low	5 4 3 2 1 0 A5 A4 A3 A2 A1 A0 High Low	7 6 5 4 3 2 1 0 High Low
0 0 0 0 0 * 0 0 0	0 0 0	<div> <div> * * * </div> <div> ↑ </div> <div> 1 1 1 1 0 </div> <div> 1 0 0 0 1 </div> <div> 1 0 0 0 1 </div> <div> 1 1 1 1 0 </div> <div> 1 0 1 0 0 </div> <div> 1 0 0 1 0 </div> <div> 1 0 0 0 1 </div> <div> * * * </div> <div> ↓ </div> </div>
0 0 0 0 0 * 0 0 1	0 0 1	<div> * * * </div> <div> ↑ </div> <div> 1 0 0 0 1 </div> <div> 0 1 0 1 0 </div> <div> 1 1 1 1 1 </div> <div> 0 0 1 0 0 </div> <div> 1 1 1 1 1 </div> <div> 0 0 1 0 0 </div> <div> 0 0 1 0 0 </div> <div> * * * </div> <div> ↓ </div>
0 0 0 0 0 * 1 1 1	1 1 1	<div> * * * </div> <div> ↑ </div> <div> 1 0 0 </div> <div> 1 0 1 </div> <div> 1 1 0 </div> <div> 1 1 1 </div> <div> * * * </div> <div> ↓ </div>

Rys. 7. Zależność pomiędzy kodem znaku a jego adresem – pamięć CGRAM

Jak wspomniano na początku podrozdziału, użytkownik może zdefiniować do ośmiu swoich znaków, wg własnego projektu. Poniżej przedstawiono etapy projektowania i programowania takich znaków, na przykładzie małych, polskich liter: *ą, ć, ę, ł, ń, ó, ś, ż* i *ź*. Ponieważ w pamięci

CGRAM jest miejsce tylko na osiem znaków, przykład pomija znak „ż”, mimo, że zaprojektowano jego wzór.

1. Na początek należy zaprojektować wzór bitowy wszystkich znaków. Każdy znak składa się z ośmiu linijek, po pięć punktów każda. Poniższe tabelki, z rysunku Rys. 8, pokazują odpowiednie wzory, wraz z odpowiadającymi im wartościami liczbowymi.

<table><tr><th>D4</th><th>D3</th><th>D2</th><th>D1</th><th>D0</th></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>	D4	D3	D2	D1	D0																																				<table><tr><th>D4</th><th>D3</th><th>D2</th><th>D1</th><th>D0</th></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>	D4	D3	D2	D1	D0																																				<table><tr><th>D4</th><th>D3</th><th>D2</th><th>D1</th><th>D0</th></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>	D4	D3	D2	D1	D0																																			
D4	D3	D2	D1	D0																																																																																																																						
D4	D3	D2	D1	D0																																																																																																																						
D4	D3	D2	D1	D0																																																																																																																						
0x00	0x02	0x00																																																																																																																								
0x00	0x04	0x00																																																																																																																								
0x0e	0x0e	0x0e																																																																																																																								
0x01	0x10	0x11																																																																																																																								
0x0f	0x10	0x1f																																																																																																																								
0x11	0x11	0x10																																																																																																																								
0x0f	0x0e	0x0e																																																																																																																								
0x01	0x00	0x02																																																																																																																								

ą

D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																													0x0c	0x02	0x02
0x04	0x04	0x04																																																																																																																																																																					
0x06	0x16	0x0e																																																																																																																																																																					
0x04	0x19	0x11																																																																																																																																																																					
0x0c	0x11	0x11																																																																																																																																																																					
0x04	0x11	0x11																																																																																																																																																																					
0x0e	0x11	0x0e																																																																																																																																																																					
0x00	0x00	0x00																																																																																																																																																																					

ć

D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																													0x02	0x04	0x02
0x04	0x00	0x04																																																																																																																																																																					
0x0e	0x1f	0x1f																																																																																																																																																																					
0x10	0x02	0x02																																																																																																																																																																					
0x0e	0x04	0x04																																																																																																																																																																					
0x01	0x08	0x08																																																																																																																																																																					
0x1e	0x1f	0x1f																																																																																																																																																																					
0x00	0x00	0x00																																																																																																																																																																					

ę

	D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																											
0x02	0x04	0x02																																																																																																																																																																		
0x04	0x00	0x04																																																																																																																																																																		
0x0e	0x1f	0x1f																																																																																																																																																																		
0x10	0x02	0x02																																																																																																																																																																		
0x0e	0x04	0x04																																																																																																																																																																		
0x01	0x08	0x08																																																																																																																																																																		
0x1e	0x1f	0x1f																																																																																																																																																																		
0x00	0x00	0x00																																																																																																																																																																		

ł

D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																													0x02	0x04	0x02
0x04	0x00	0x04																																																																																																																																																																					
0x0e	0x1f	0x1f																																																																																																																																																																					
0x10	0x02	0x02																																																																																																																																																																					
0x0e	0x04	0x04																																																																																																																																																																					
0x01	0x08	0x08																																																																																																																																																																					
0x1e	0x1f	0x1f																																																																																																																																																																					
0x00	0x00	0x00																																																																																																																																																																					

ń

D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																													0x02	0x04	0x02
0x04	0x00	0x04																																																																																																																																																																					
0x0e	0x1f	0x1f																																																																																																																																																																					
0x10	0x02	0x02																																																																																																																																																																					
0x0e	0x04	0x04																																																																																																																																																																					
0x01	0x08	0x08																																																																																																																																																																					
0x1e	0x1f	0x1f																																																																																																																																																																					
0x00	0x00	0x00																																																																																																																																																																					

ó

	D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																											
0x02	0x04	0x02																																																																																																																																																																		
0x04	0x00	0x04																																																																																																																																																																		
0x0e	0x1f	0x1f																																																																																																																																																																		
0x10	0x02	0x02																																																																																																																																																																		
0x0e	0x04	0x04																																																																																																																																																																		
0x01	0x08	0x08																																																																																																																																																																		
0x1e	0x1f	0x1f																																																																																																																																																																		
0x00	0x00	0x00																																																																																																																																																																		

ś

D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																													0x02	0x04	0x02
0x04	0x00	0x04																																																																																																																																																																					
0x0e	0x1f	0x1f																																																																																																																																																																					
0x10	0x02	0x02																																																																																																																																																																					
0x0e	0x04	0x04																																																																																																																																																																					
0x01	0x08	0x08																																																																																																																																																																					
0x1e	0x1f	0x1f																																																																																																																																																																					
0x00	0x00	0x00																																																																																																																																																																					

ź

	D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																													D4	D3	D2	D1	D0		----	----	----	----	----																																											
0x02	0x04	0x02																																																																																																																																																																		
0x04	0x00	0x04																																																																																																																																																																		
0x0e	0x1f	0x1f																																																																																																																																																																		
0x10	0x02	0x02																																																																																																																																																																		
0x0e	0x04	0x04																																																																																																																																																																		
0x01	0x08	0x08																																																																																																																																																																		
0x1e	0x1f	0x1f																																																																																																																																																																		
0x00	0x00	0x00																																																																																																																																																																		

ż

Rys. 8. Wzory bitowe małych, polskich liter

2. Uaktywnić pamięć CGRAM i ustawić licznik adresów (AC) na początek jej przestrzeni adresowej, czyli ACG=00, wykorzystując instrukcję „Set CGRAM address” (patrz tabela Tab. 2).
3. Wpisać, bajt po bajcie, powstałe ciągi liczb (tylko osiem znaków), do pamięci CGRAM, stosując instrukcję „Write data to CG or DDRAM” (patrz tabela Tab. 2). Każdy zapis lub odczyt zwiększa automatycznie zawartość licznika adresów (AC). Jest to domyślny kierunek zmiany zawartości licznika adresów (AC). Kierunek ten można odwrócić, modyfikując odpowiedni bit (I/D), instrukcją „Entry mode set”. Zakładając kolejność, jak na rysunku Rys. 8, pod adresem 00 będzie litera „ą”, pod adresem 01 litera „ć”, itd. Ponieważ czwarty bit kodu znaku nie bierze udziału w adresacji pamięci CGRAM, dlatego, np. kod litery „ą”

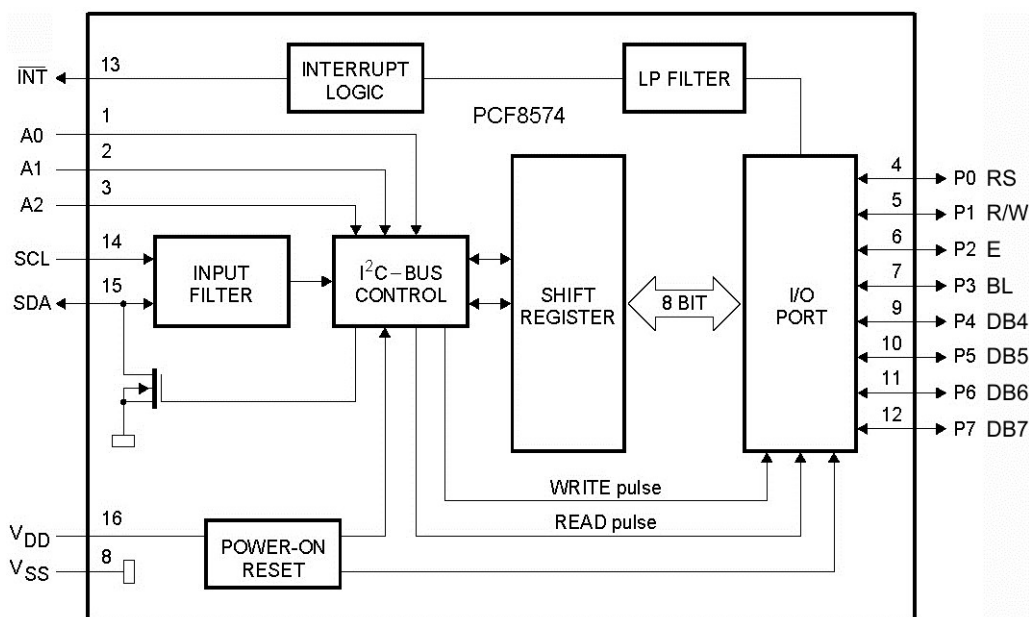
to 00 lub 08. Przed każdym zapisem należy sprawdzić stan flagi BF. Kontynuować można tylko w przypadku BF=0.

4. Uaktywnić pamięć DDRAM i ustawić licznik adresów (AC) na początek jej przestrzeni adresowej, czyli 00, wykorzystując instrukcję „Set DDRAM address” (patrz tabela Tab. 2). Aby wyświetlić wybraną literę, wystarczy wpisać jej kod do pamięci DDRAM.

W dołączonym do instrukcji pakiecie *Lab\_7a.zip*, w katalogu *PL\_Char*, jest program prezentujący polskie litery na wyświetlaczu LCD. Biblioteka *lcd1602.c* została uzupełniona o funkcję *LCD1602\_PL\_CH()*, ładującą osiem, małych polskich liter do pamięci CGRAM oraz o funkcję *LCD1602\_PrintPL(char \*str)*, pozwalającą wyświetlać te litery w tekście, znajdującym się w tablicy *\*str*. Aby funkcja ta działała poprawnie, należy wprowadzić kodowanie polskich liter w tekście. Wybrano następujący sposób: //a to „ą”, //s to „ś”, //z to „ż”, itd. Aby, np. wyświetlić „/ś” należy wpisać ///*s*.

## 7. 8-BITOWY EKSPANDER I<sup>2</sup>C

Układ PCF8574 jest 8-bitowym ekspanderem magistrali I<sup>2</sup>C, wykorzystywanym do realizacji interfejsu pomiędzy magistralą szeregową, jaką jest I<sup>2</sup>C, a magistralą równoległą, którą, przykładowo posiada układ HD44780U, sterujący popularnymi wyświetlaczami LCD. Na rysunku Rys. 9 Rys. 9 jest przedstawiony schemat blokowy ekspandera, wraz z podłączeniem do linii sterujących wyświetlacza.



Rys. 9 Schemat blokowy ekspandera I<sup>2</sup>C – 8-bitów, wraz z podłączeniem do linii sterujących wyświetlacza

Rejestr przesuwany (shift register) zamienia daną szeregową na równoległą (zapis do I<sup>2</sup>C) lub równoległą na szeregową (odczyt I<sup>2</sup>C). Port P0 ÷ P7 jest pseudo dwukierunkowy, tzn., że aby dana końcówka mogła pracować jako wejście, jej bufor wyjściowy musi być wcześniej

zapisany wartością „1” (żeby wyjście Px nie „walczyło” z sygnałem wejściowym). To jest druga obserwacja, która pozwoli na niezakłócony odczyt danych z wyświetlacza.

Ponieważ do budowy modułów wyświetlaczy, używanych w laboratorium, wykorzystano układy PCF8574 i PCF8574A, które posiadają różne adresy I<sup>2</sup>C, dlatego program inicjujący pracę wyświetlacza ma na początku funkcję rozpoznającą, który z tych układów jest obecny w systemie (*LCD1602\_CheckAddress()* w bibliotece *ldc1602.c*).

Jak wspomniano w rozdziale 6, 4-bitowa szyna danych oraz 4-bitowa szyna sterująca wyświetlacza LCD, zostały podłączone do 8-bitowego portu ekspandera (Rys. 9). W jednym bajcie jest zawarte równocześnie sterowanie, jak i część danej. W jednym bajcie potrafi być zarówno wielkość wejściowa, jak i wyjściowa. Dlatego, tak przy zapisie, jak i odczycie, transfer jest prowadzony „na raty” tak, aby przesyłając daną, generować jednocześnie odpowiednie stany oraz ich zmienność na szynie sterującej (patrz Rys. 4), dla każdego półbajta osobno.

## 7.1. ZAPIS DO WYŚWIETLACZA LCD

Na przykładzie instrukcji „*Set DDRAM address*” (patrz Tab. 2), zostanie zaprezentowany algorytm zapisu licznika adresu (AC), wskazującego na adres (ADD), w pamięci DDRAM (Tab. 3) adresem będzie siódma pozycja, w pierwszej linii (ADD=06) na wyświetlaczu. Każdy wiersz w tabeli Tab. 3 to jeden zapis do I<sup>2</sup>C. Bit BL (podświetlenie) ma wartość, zależną od wymagania. W niniejszym przykładzie ma wartość „1”.

\* - oznacza dowolną wartość

WR – zapis I<sup>2</sup>C

RD – odczyt I<sup>2</sup>C

ADDH – starsza część adresu ADD

ADDL – młodsza część adresu ADD

LCD	DB7	DB6	DB5	DB4	BL	E	R/W	RS	Wyjaśnienie
EKSP	P7	P6	P5	P4	P3	P2	P1	P0	
WR	*	*	*	*	1	0	0	0	Stan wyjściowy: E=0 – nieaktywny, R/W=0 – zapis, RS=0 – rejestr sterujący
WR	1	0	0	0	1	1	0	0	Starsze pół bajta instrukcji (ADDH), E=1 – przygotowanie sygnału zapisu
WR	1	0	0	0	1	0	0	0	Dana bez zmian, E=0 – generacja opadającego sygnału zapisu; koniec zapisu
WR	0	1	1	0	1	1	0	0	Młodsze pół bajta instrukcji (ADDL), E=1 – przygotowanie sygnału zapisu
WR	0	1	1	0	1	0	0	0	Dana bez zmian, E=0 – generacja opadającego sygnału zapisu; koniec zapisu całości

Tab. 3. Zapis rejestru wyświetlacza LCD

Na przykładzie instrukcji „*Write data to CG or DDRAM*” (patrz Tab. 2), zostanie zaprezentowany algorytm zapisu danej do pamięci DDRAM (Tab. 4), w pozycji, ustawionej poprzednią instrukcją (Tab. 3). Przykładową daną będzie kod litery „R” (kod 0x52).

LCD	DB7	DB6	DB5	DB4	BL	E	R/W	RS	Wyjaśnienie
EKSP	P7	P6	P5	P4	P3	P2	P1	P0	
WR	*	*	*	*	1	0	0	1	Stan wyjściowy: E=0 – nieaktywny, R/W=0 – zapis, RS=1 – przestrzeń danych
WR	0	1	0	1	1	1	0	1	Starsze pół bajta danej (5), E=1 – przygotowanie sygnału zapisu
WR	0	1	0	1	1	0	0	1	Dana bez zmian, E=0 – generacja opadającego sygnału zapisu; koniec zapisu
WR	0	0	1	0	1	1	0	1	Młodsze pół bajta danej (2), E=1 – przygotowanie sygnału zapisu
WR	0	0	1	0	1	0	0	1	Dana bez zmian, E=0 – generacja opadającego sygnału zapisu; koniec zapisu całości

Tab. 4. Zapis danej w pamięci DDRAM wyświetlacza LCD

## 7.2. ODCZYT Z WYŚWIETLACZA LCD

Na przykładzie instrukcji „*Read busy flag & address*” (patrz Tab. 2), zostanie zaprezentowany algorytm odczytu flagi zajętości (BF) i licznika adresu (AC) (Tab. 5), wskazującego na adres (ADD), w pamięci DDRAM, z przykładu zapisu danej do tej pamięci (Tab. 4). W przykładzie tym, dana, w postaci litery „R”, została zapisana pod adres ADD=06. Po zapisie, licznik adresu (AC) automatycznie zwiększył swoją zawartość o 1, czyli na ADD=07. Aby wygenerować zmienność sygnału E, należy wykonać odpowiednią liczbę zapisów do I<sup>2</sup>C. W czasie tych zapisów, część wysyłanej danej, odnosząca się do DB7 ÷ DB4, musi mieć wartość „1”, aby linie te mogły pracować jako wejścia (odczyt).

ACL – młodsza część AC

ACH – starsza część AC

LCD	DB7	DB6	DB5	DB4	BL	E	R/W	RS	Wyjaśnienie
EKSP	P7	P6	P5	P4	P3	P2	P1	P0	
WR	1	1	1	1	1	0	1	0	Stan wyjściowy: E=0 – nieaktywny, R/W=1 – odczyt, RS=0 – rejestr sterujący
WR	1	1	1	1	1	1	1	0	E=1 – otwarcie buforów wyjściowych w LCD, ze starszą częścią danej (BF i ACH - Rys. 4)
RD	BF	0	0	0	1	1	1	0	Odczyt przez I <sup>2</sup> C otwartych buforów LCD, ze starszą częścią danej (BF i ACH)
WR	1	1	1	1	1	0	1	0	E=0 – zamknięcie buforów wyjściowych LCD
WR	1	1	1	1	1	1	1	0	E=1 – otwarcie buforów wyjściowych w LCD, z młodszą częścią danej (ACL - Rys. 4)
RD	0	1	1	1	1	1	1	0	Odczyt przez I <sup>2</sup> C otwartych buforów LCD, z młodszą częścią danej (ACL)
WR	1	1	1	1	1	0	1	0	E=0 – zamknięcie buforów wyjściowych LCD

Tab. 5. Odczyt stanu flagi BF i licznika adresu AC

Na przykładzie instrukcji „*Read data from CG or DDRAM*” (patrz Tab. 2), zostanie zaprezentowany algorytm odczytu danej z pamięci DDRAM (Tab. 6), z pozycji, ustawionej instrukcją zapisującą literę „R” do pamięci DDRAM (Tab. 4). Po zapisie, licznik adresu (AC) automatycznie zwiększył swoją zawartość o 1, czyli na ADD=07. Załóżmy, że pod tym adresem jest cyfra „1” (kod 0x31).

LCD	DB7	DB6	DB5	DB4	BL	E	R/W	RS	Wyjaśnienie
EKSP	P7	P6	P5	P4	P3	P2	P1	P0	
WR	1	1	1	1	1	0	1	1	Stan wyjściowy: E=0 – nieaktywny, R/W=1 – odczyt, RS=0 – przestrzeń danych
WR	1	1	1	1	1	1	1	1	E=1 – otwarcie buforów wyjściowych w LCD, ze starszą częścią danej (Rys. 4)
RD	0	0	1	1	1	1	1	1	Odczyt przez I <sup>2</sup> C otwartych buforów LCD, ze starszą częścią danej (3)
WR	1	1	1	1	1	0	1	1	E=0 – zamknięcie buforów wyjściowych LCD
WR	1	1	1	1	1	1	1	1	E=1 – otwarcie buforów wyjściowych w LCD, z młodszą częścią danej (Rys. 4)
RD	0	0	0	1	1	1	1	1	Odczyt przez I <sup>2</sup> C otwartych buforów LCD, z młodszą częścią danej (1)
WR	1	1	1	1	1	0	1	1	E=0 – zamknięcie buforów wyjściowych LCD

Tab. 6. Odczyt danej z pamięci DDRAM wyświetlacza LCD

### 7.3. OBSERWACJA DZIAŁANIA INSTRUKCJI „*READ BUSY FLAG & ADDRESS*”

---

W dołączonym do instrukcji pakiecie *Lab\_7a.zip*, w katalogu *BF\_Read*, jest program prezentujący działanie instrukcji „*Read busy flag & address*”. Ponieważ trudno jest wyświetlić stan flagi BF, więc program wyświetla stan aktualnej zawartości licznika adresów (AC), odnoszący się do pozycji kursora w pamięci DDRAM, który cyklicznie zmienia swoją pozycję. Bibliotekę *lcd1602.c* uzupełniono o funkcję *LCD1602\_BF\_AC(uint8\_t \*ptr)*, która zwraca wartość flagi BF, a zmiennej *\*ptr* przypisuje odczytaną wartość licznika adresu (AC). Proszę zwrócić uwagę na:

- ✓ wartość adresu początku drugiej linii,
- ✓ ilość znaków w jednej linii (widać tylko pierwszych szesnaście znaków).

Bibliotekę *i2c.c* uzupełniono o funkcję *I2C\_Read(uint8\_t address, uint8\_t \*ptr)*. Funkcja ta czyta daną z urządzenia I<sup>2</sup>C, o adresie *address*, i zwraca błąd, w przypadku niepowodzenia, a zmiennej *\*ptr* przypisuje odczytaną wartość danej.