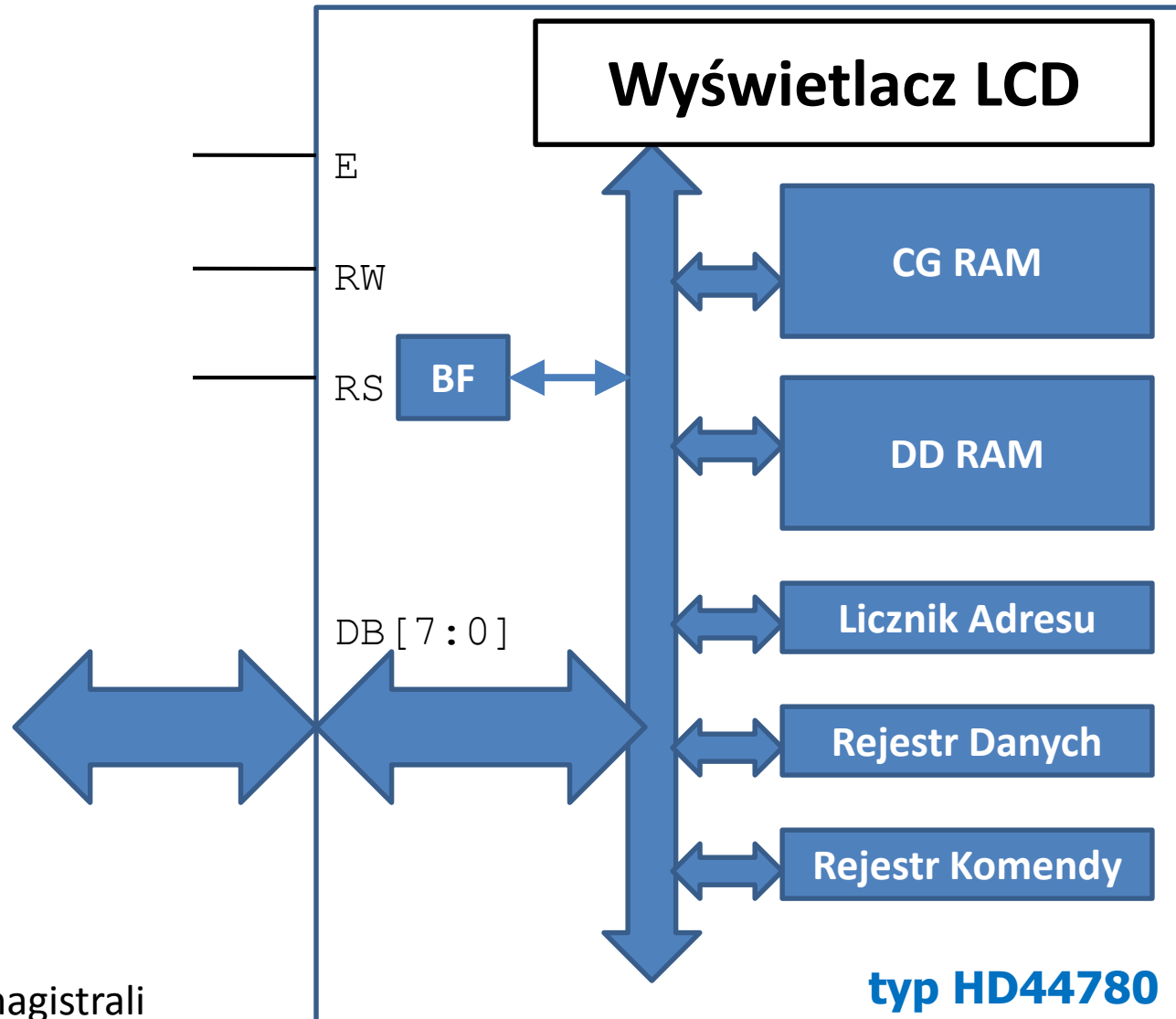


# **Sterowanie wyświetlacza LCD typu HD44780**

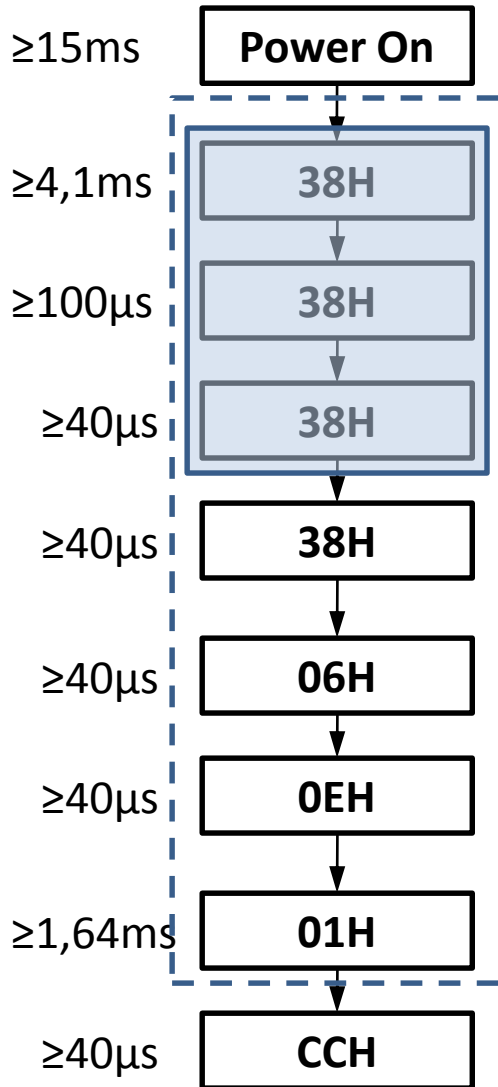
*Laboratorium Języków Opisu Sprzętu AGH WFIS*

# Matrycowy moduł LCD



**BF** – Busy Flag –  
znacznik zajętości magistrali

# Incjalizacja modułu LCD



Reset Sequence

Function set

0	0	1	DL	N	F	X	X
---	---	---	----	---	---	---	---

Entry mode

0	0	0	0	0	1	ID	S
---	---	---	---	---	---	----	---

Display on/off

0	0	0	0	1	D	C	B
---	---	---	---	---	---	---	---

Display clear

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

Set DD RAM address

1	A6	A5	A4	A3	A2	A1	A0
---	----	----	----	----	----	----	----

DL – 8 bit (1), 4 bit(0)  
 N – 2 lines(1), 1 line(0)  
 F – 5\*7 dots(0)

ID – increment(1),  
 decrement(0)  
 S – shift on(1), off(0)

D – display on(1), off(0)  
 C – cursor on(1), off(0)  
 B – blink on(1), off(0)

**CCH** oznacza ostatnie cztery widoczne pozycje w drugiej linii.

# Przebiegi czasowe

## Proces inicjalizacji - komendy:

reset sequence  
&  
function set

entry  
mode

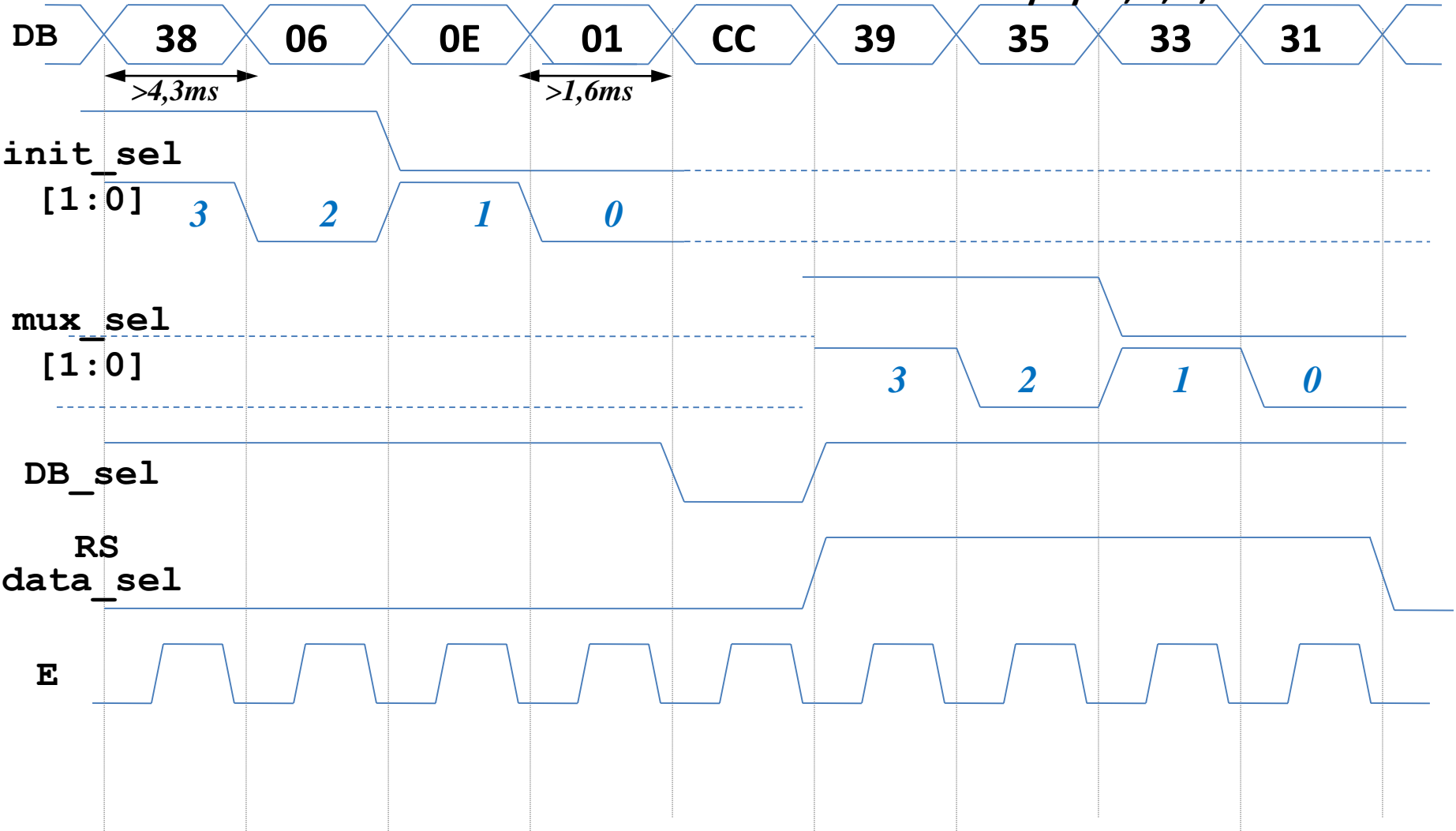
display  
on

display  
clear

Adres  
DD RAM

Na wyświetlaczu pojawią się

cyfry: 9, 5, 3, 1



# Parametryzacja projektu

Liczbę dekad licznika sparametryzować na definicji modułu szczytowego:

```
#(parameter ndec = 4; n =clogb2(ndec)
```

Gdzie `n` jest liczba bitów licznika danych o nazwie `mux_sel`.

Liczbę wysyłanych w inicjacji komend określić za pomocą dyrektywy kompilacji:

```
`define ncmd 4 ,
```

a następnie w module szczytowym ustalić liczbę bitów licznika komend:

```
localparam nc = clogb2(`ncmd);
```

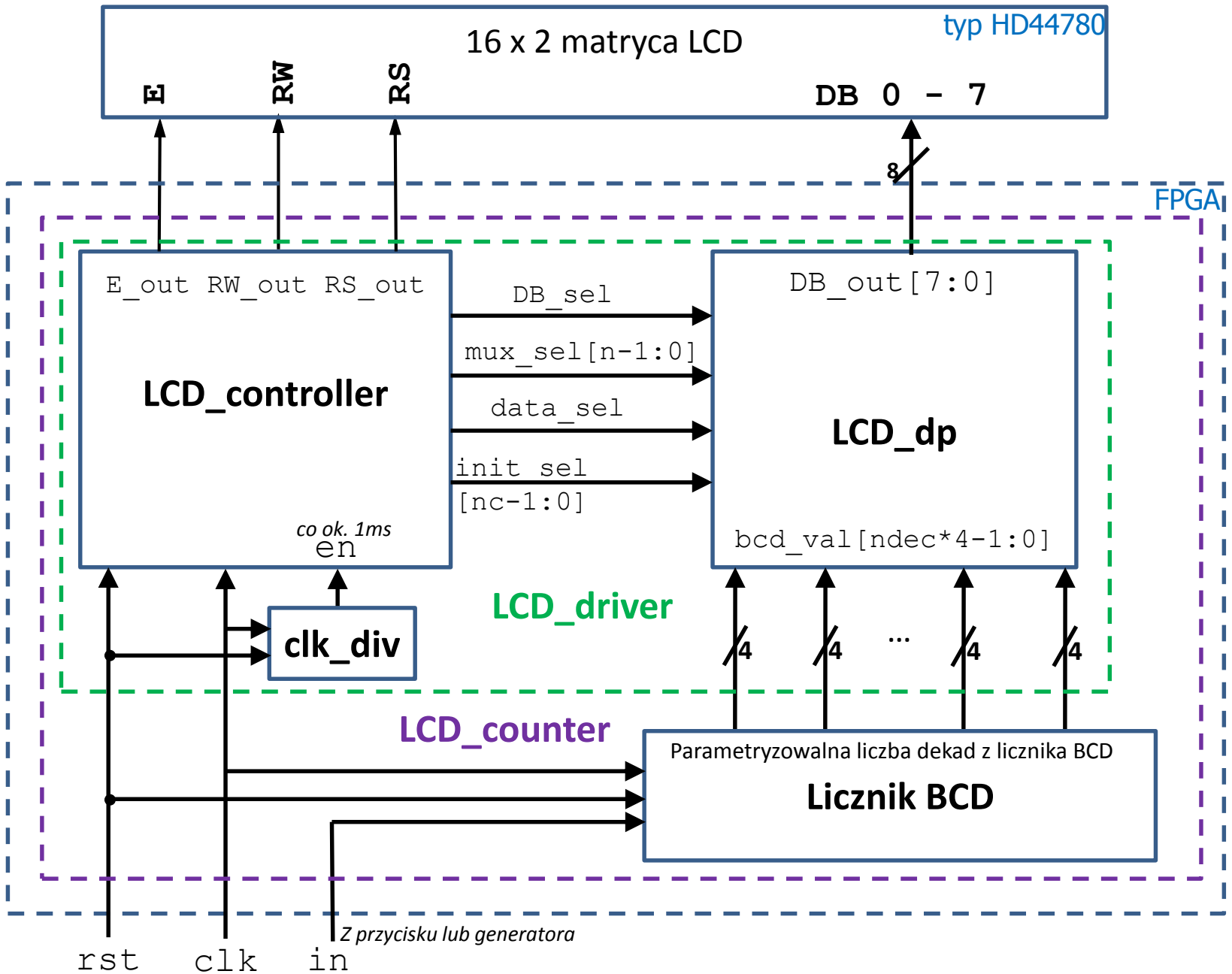
## **Uwaga:**

W wersji wstępnej projektu zrezygnować z parametryzacji ustalając ręcznie rozmiary liczników:

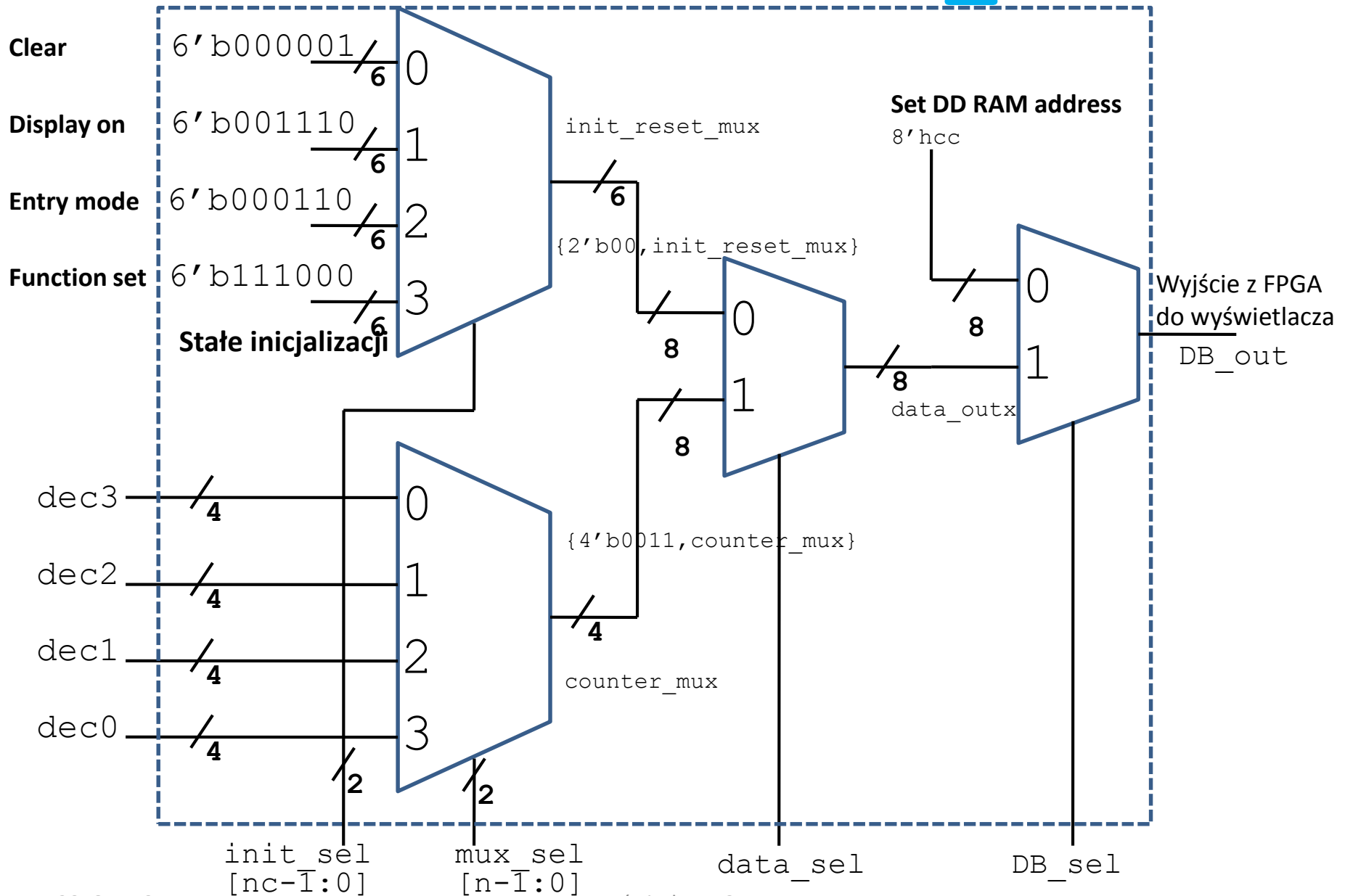
```
[1:0] mux_sel
```

```
[1:0] init_sel
```

i parametr `size` dla licznika BCD.



# Ścieżka danych LCD\_dp

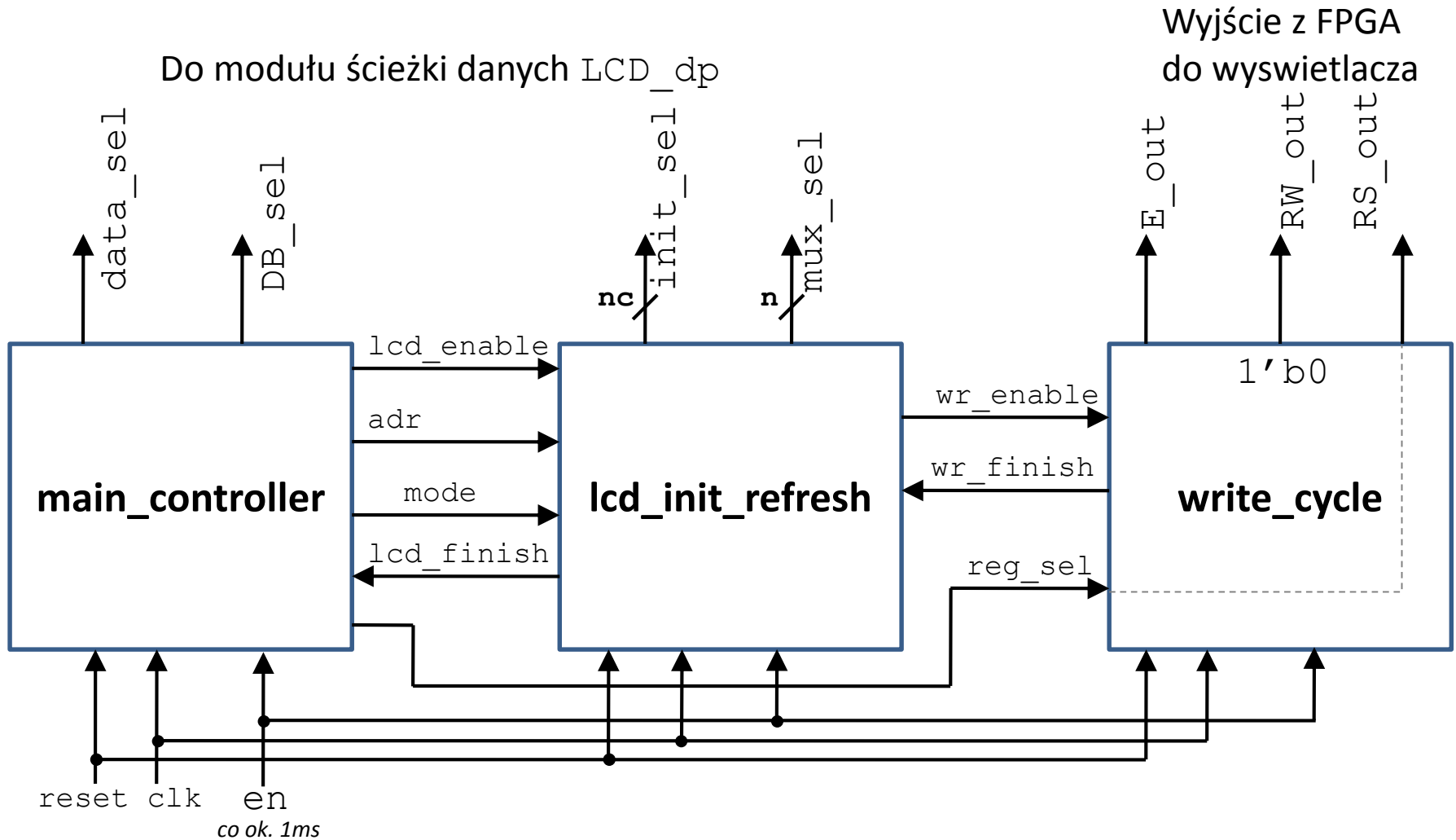


Przydatne makra dla modułu lcd\_dp:

```
`define      Clear          6'b000001
`define      DisplayOn     6'b001110
`define      EntryMode     6'b000110
`define      FunctionSet   6'b111000
`define      InitAddr      8'hcc
```

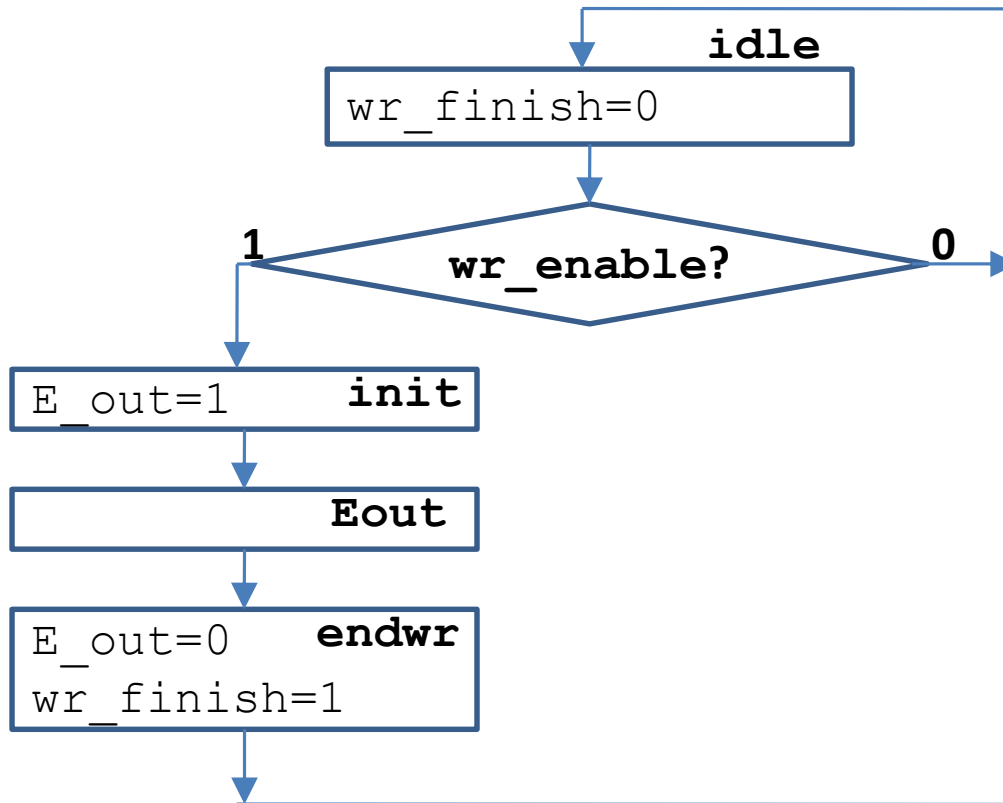


# Sterowanie LCD\_controller

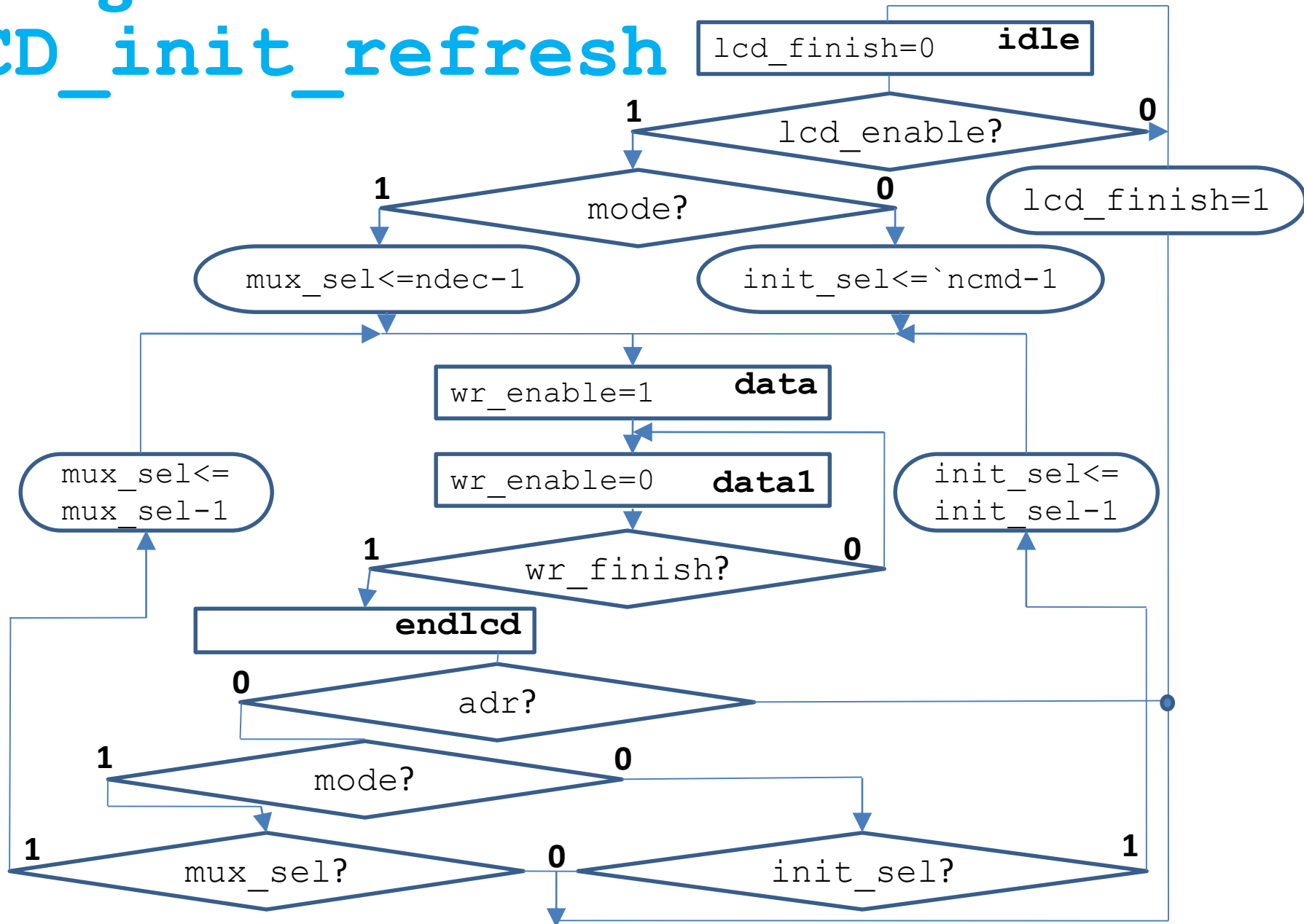


# Diagram automatu `write_cycle`

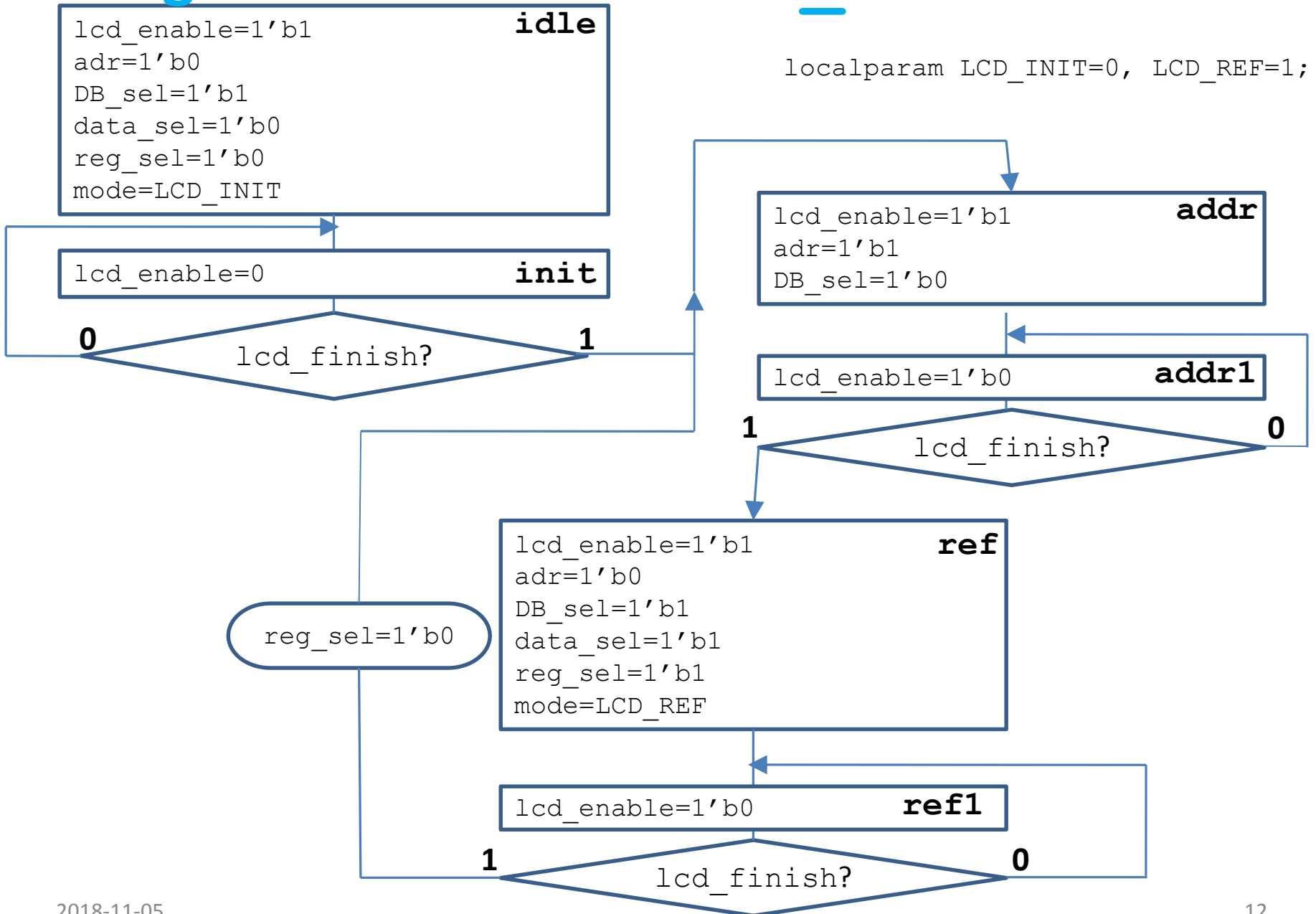
```
assign RS_out=reg_sel;  
assign RW_out=1'b0;
```



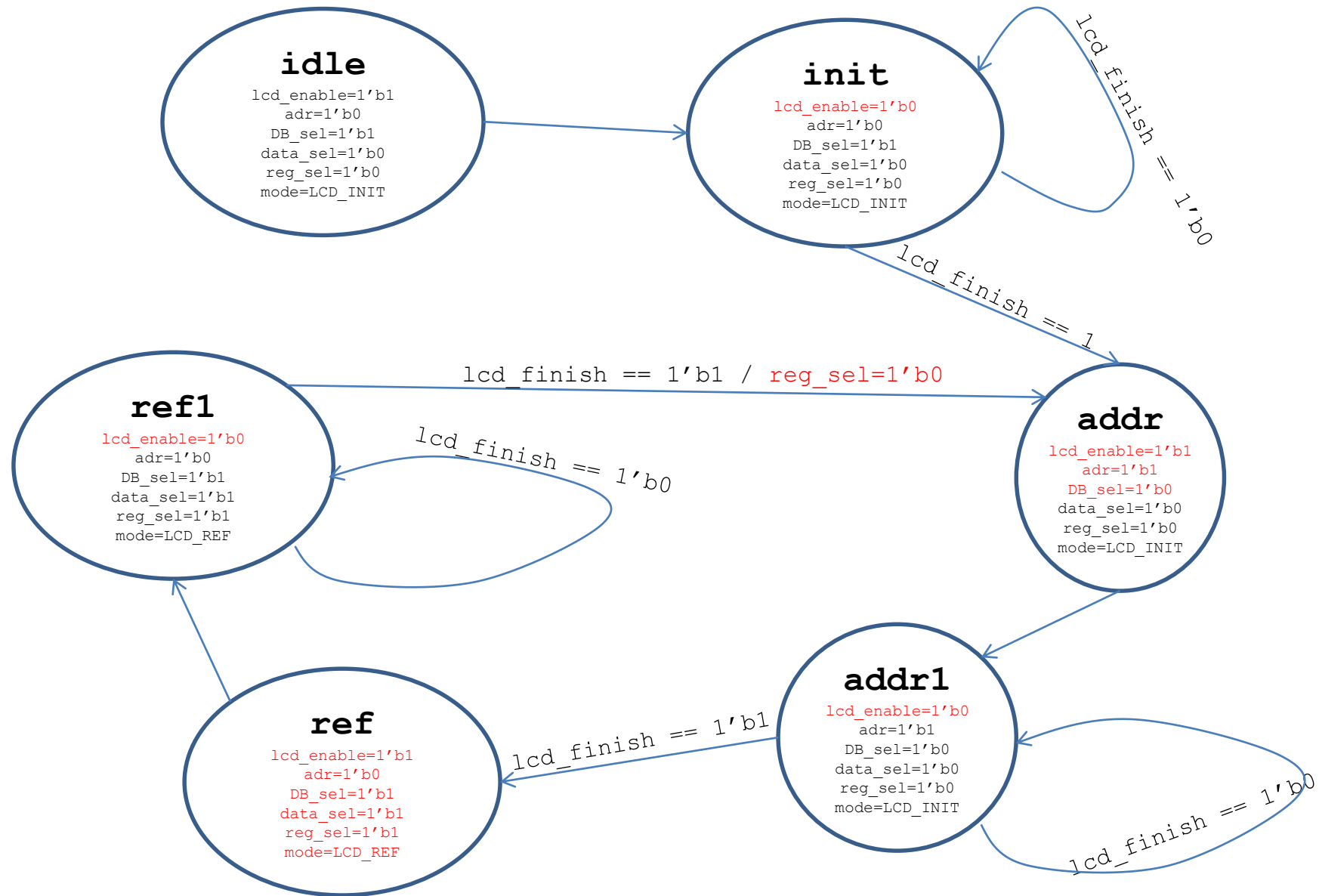
# Diagram automatu LCD\_init\_refresh



# Diagram automatu main\_controller



# Graf automatu main\_controller



# Sygnaty main\_controller

sygnał		znaczenie
lcd_enable	output	Uruchomienie automatu podrzędnego lcd_init_refresh
adr	output	Znacznik trybu adresowania dla automatu podrzędnego lcd_init_refresh
DB_sel	output	Sterowanie wyjściowego multipleksera adres-(dane i komendy) w ścieżce danych lcd_dp
data_sel	output	Sterowanie pośredniego multipleksera dane-komendy w ścieżce danych lcd_dp
reg_sel	output	Sygnał RS dla wyświetlacza służący do rozróżnienia danych i komend
mode	output	Rozróżnienie trybu pracy LCD_INIT=0 (inicjalizacja + adresowanie) od trybu LCD_REF=1 (odświeżanie danych)
lcd_finish	input	Status zakończenia pracy przez automat podrzędny lcd_init_refresh