

Hobbi Elektronika



		y	
		0	1
x	0	0	0
	1	0	1

		y	
		0	1
x	0	0	1
	1	1	1

		y	
		0	1
x	0	0	1
	1	0	1

		y	
		0	1
x	0	0	1
	1	1	0

Figure 1. Truth tables

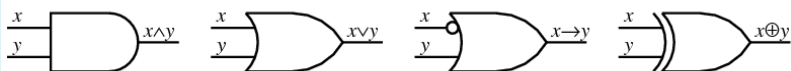


Figure 2. Logic gates

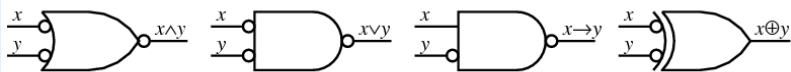


Figure 3. De Morgan equivalents



Figure 4. Venn diagrams



A digitális elektronika alapjai:
Újrakonfigurálható logikai eszközök



Programozható logikai eszközök

Programozható logikai áramkörök (*Programmable Logic Devices*) a kombinációs logikai hálózatok és sorrendi hálózatok tervezésére használhatjuk.

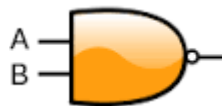
A hagyományos logikai hálózatok dedikált összeköttetésekkel, illetve kötött funkcióval (kimeneti függvény) rendelkeznek. A programozható logikai eszközökben pontosan ezek változtathatók:

- ❖ A felhasználó által *egyszer programozható*/konfigurálható logikai eszközök (*OTP: One Time Programmable*), amelynél a gyártás során nem definiált funkció egyszer még megváltoztatható (ilyenek pl. a korai PAL, PLA eszközök)
- ❖ *Többször, akár tetszőleges módon* programozható (újrakonfigurálható) logikai eszközök (ilyenek pl. a korábbi GAL, vagy a mai modern **CPLD** és **FPGA** eszközök)

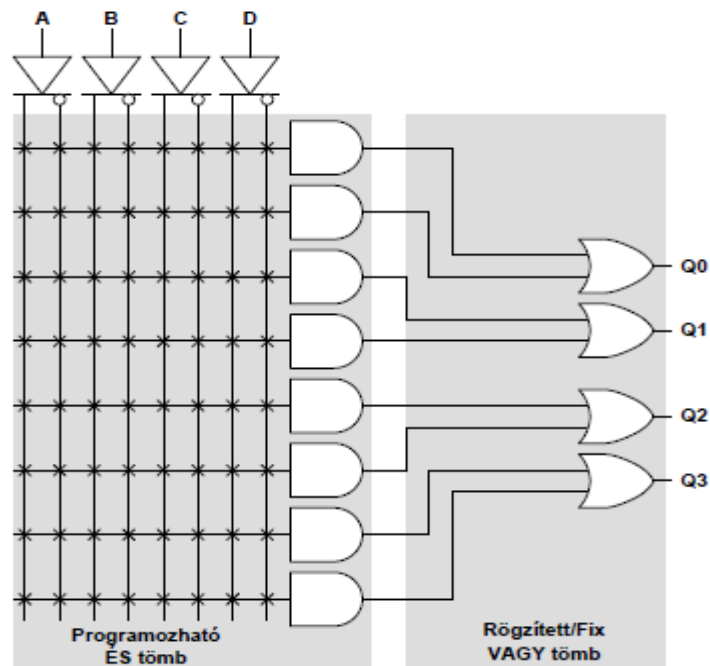
CPLD – Complex Programmable Logical Devices

FPGA – Field Programmable Gate Array

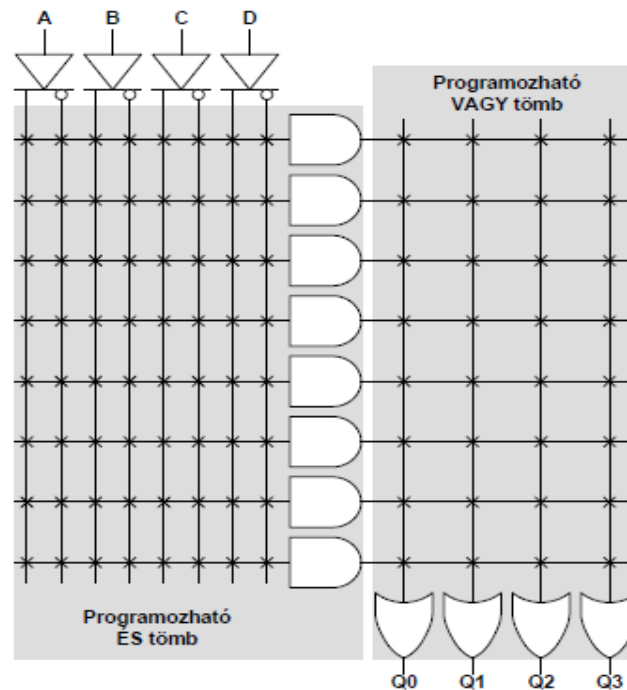
PAL, PLA, CPLD



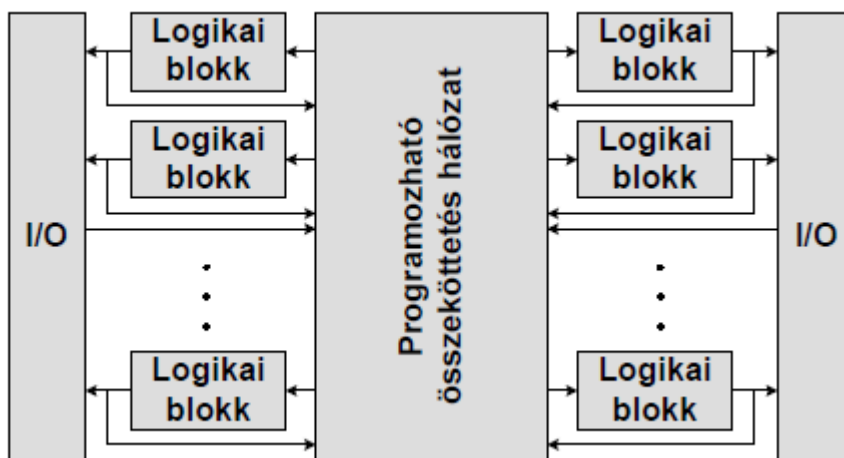
PAL

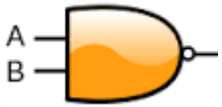


PLA



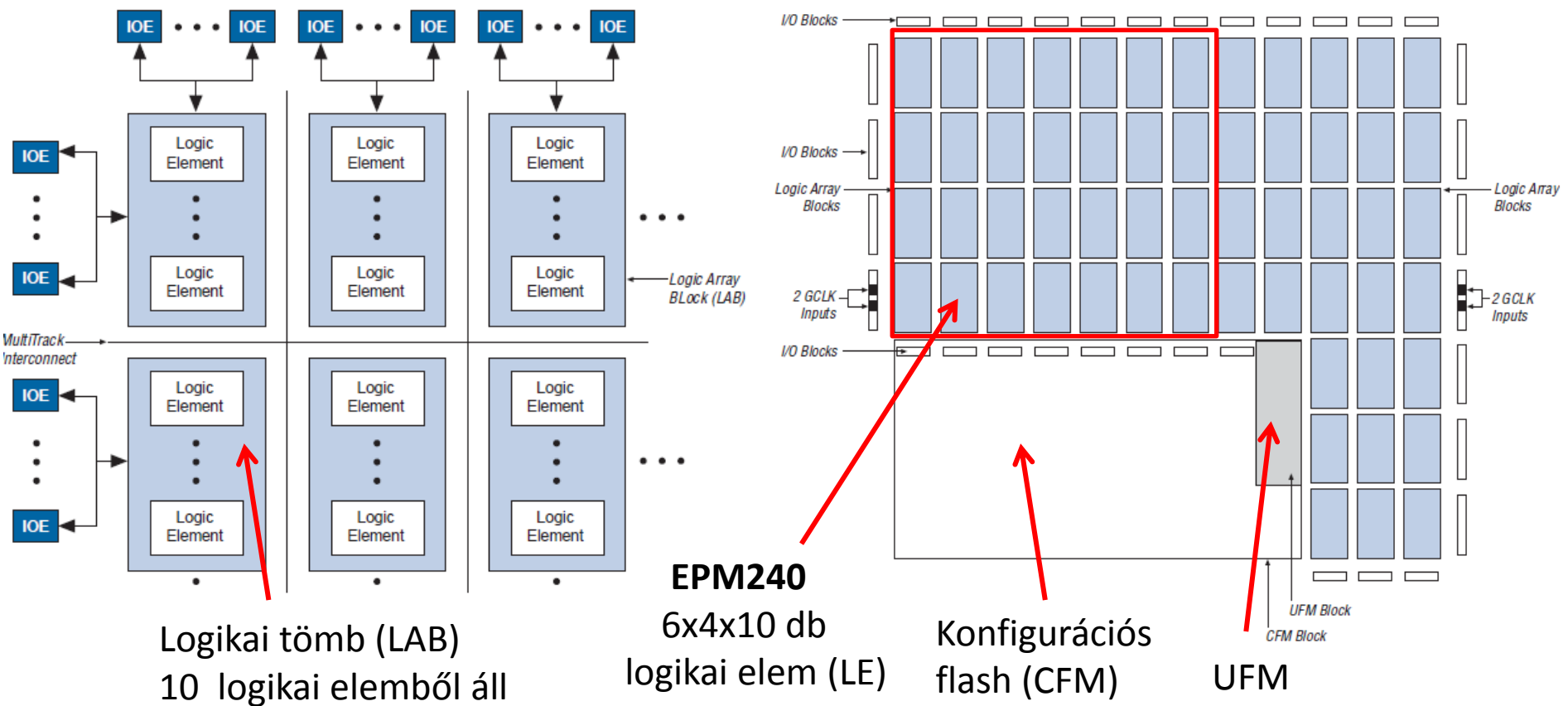
CPLD

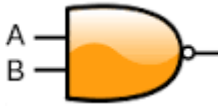




ALTERA MAX II CPLD család

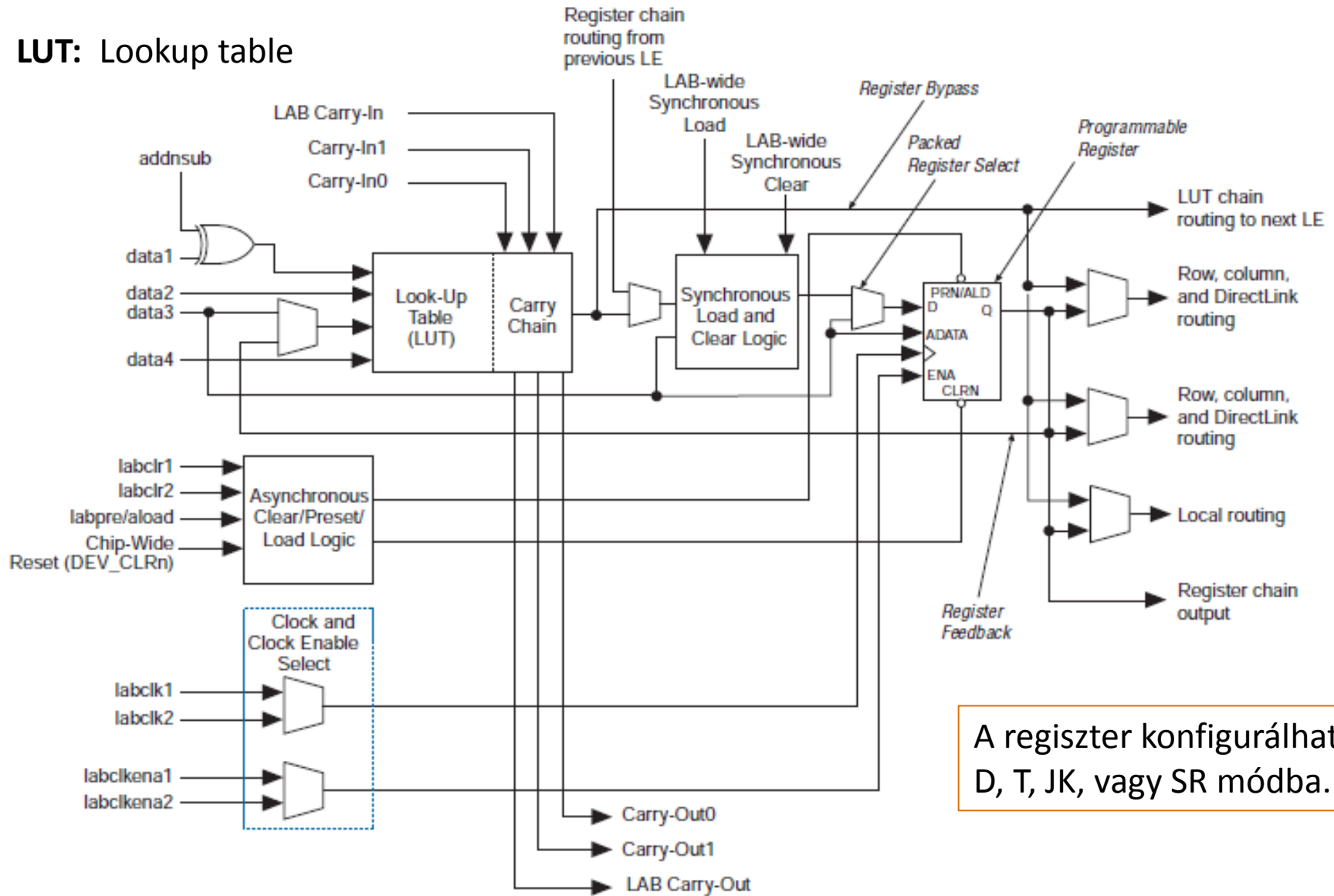
Az **EPM240C100T5N** CPLD IC az **ALTERA MAX II** termékcsaládba tartozik. Főbb jellemzői: 240 db. logikai egység, kis fogyasztású, olcsó CPLD, 50 MHz órajel, max. 3,3 V tápfesz és jelszint, busz-barát ki- és bemenetek, Schmitt-triggeres bemenet, JTAG újrakonfigurálhatóság (tipikusan 100-szor), 8kbit felhasználói FLASH memória (UFM).



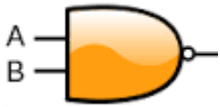


Egy logikai elem szerkezete

LUT: Lookup table



A regiszter konfigurálható D, T, JK, vagy SR módba.



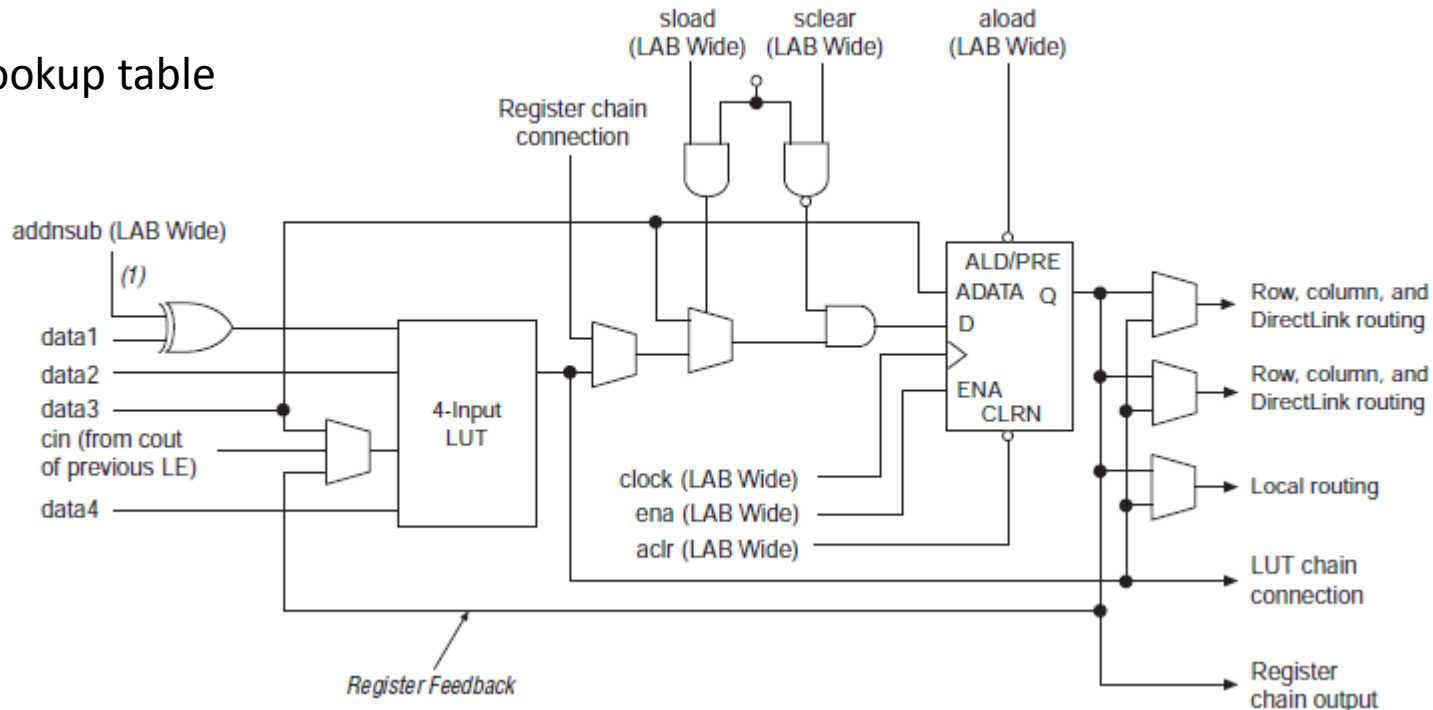
A logikai elemek üzemmódjai

Az ALTERA MAX II logikai elemei kétféle módban működhetnek:

- ❖ Normál mód (általános logikai alkalmazásokhoz, kombinációs hálózatokhoz)
- ❖ Dinamikus aritmetikai mód (összeadó, számláló, akkumulátor, komparátor, stb)

Normál mód: 1 db 4-bemenetű LUT

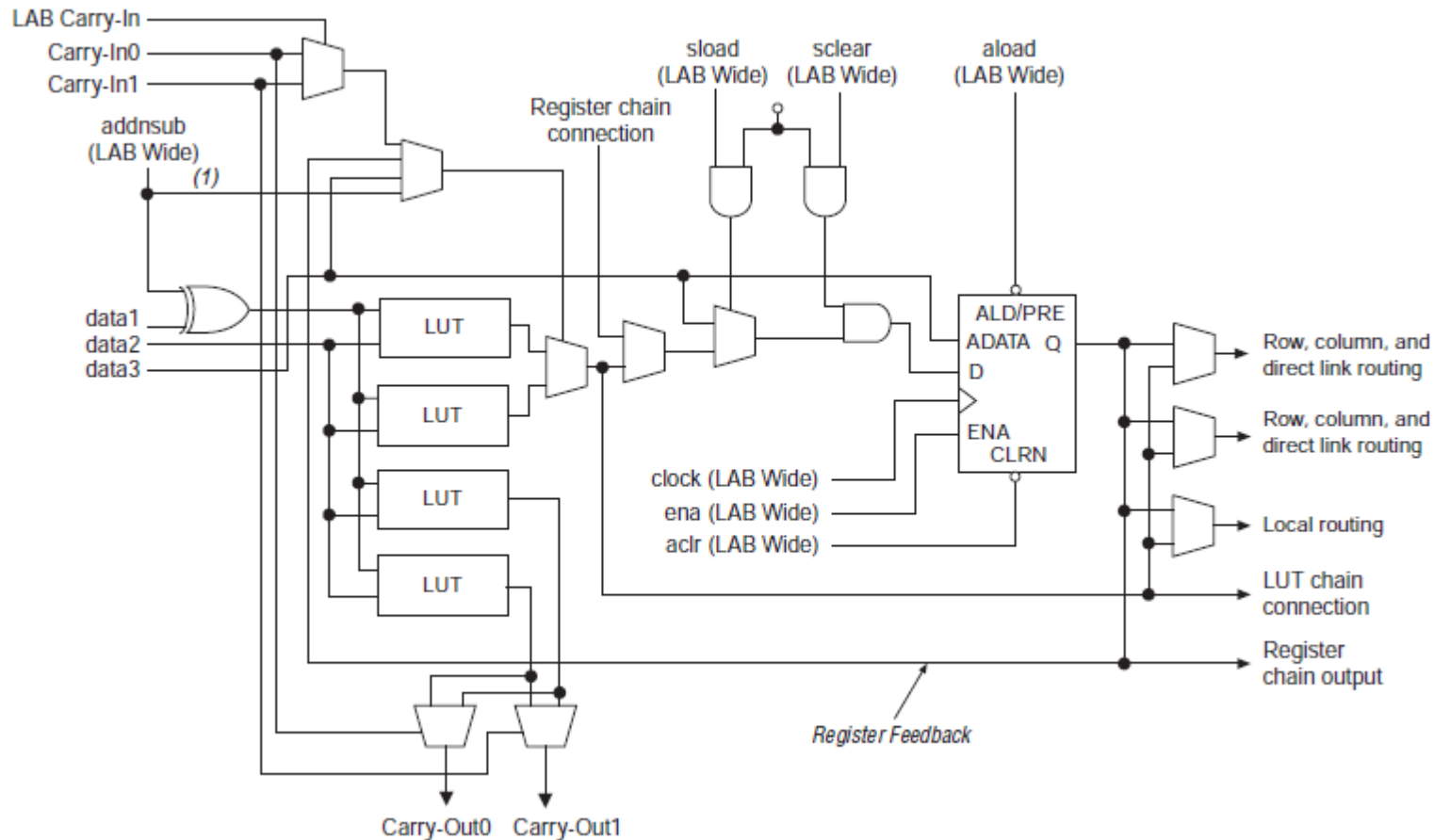
LUT: Lookup table

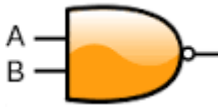




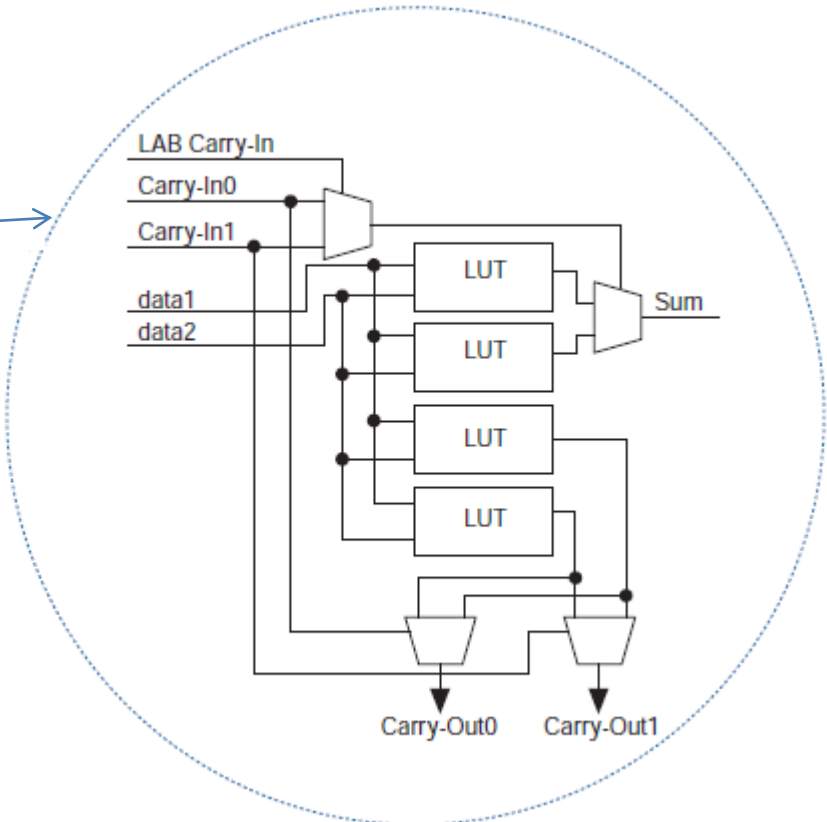
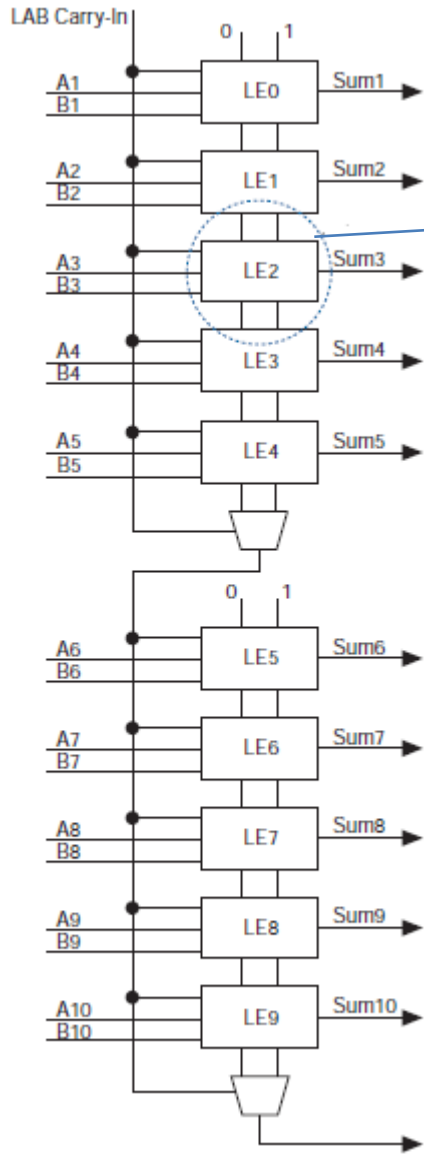
A logikai elemek üzemmódjai

Dinamikus Aritmetikai mód: 4 db 2-bemenetű LUT





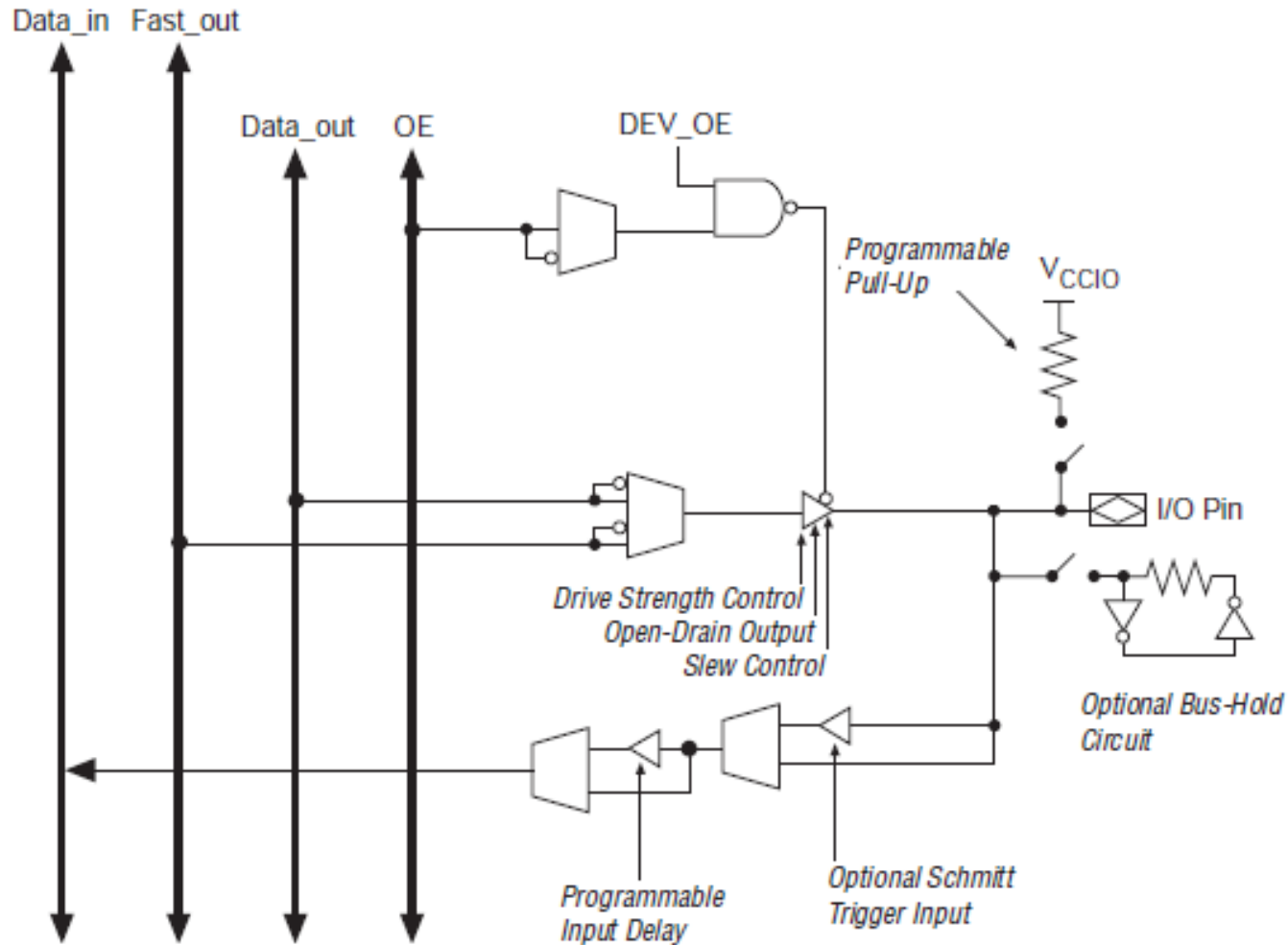
A Carry Select lánc használata



A Carry-select lánc segítségével a gyors átvitelképzés megoldható



Az I/O modulok felépítése





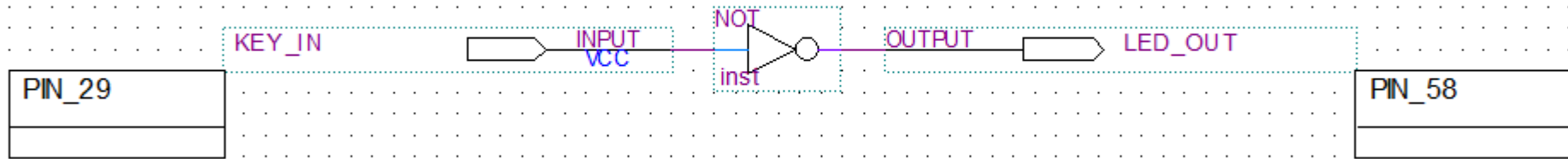
ALTERA QUARTUS Prime (Lite)

Tervezési lehetőségek:

- Kapcsolási rajz
- Verilog HDL
- VHDL

Egyszerű kombinációs logika

Rajz:



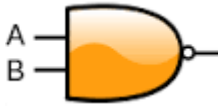
Verilog:

```
module LED_NOT_KEY ( A, F );  
    input KEY_IN;  
    output LED_OUT;  
    assign LED_OUT = ~KEY_IN;  
endmodule
```

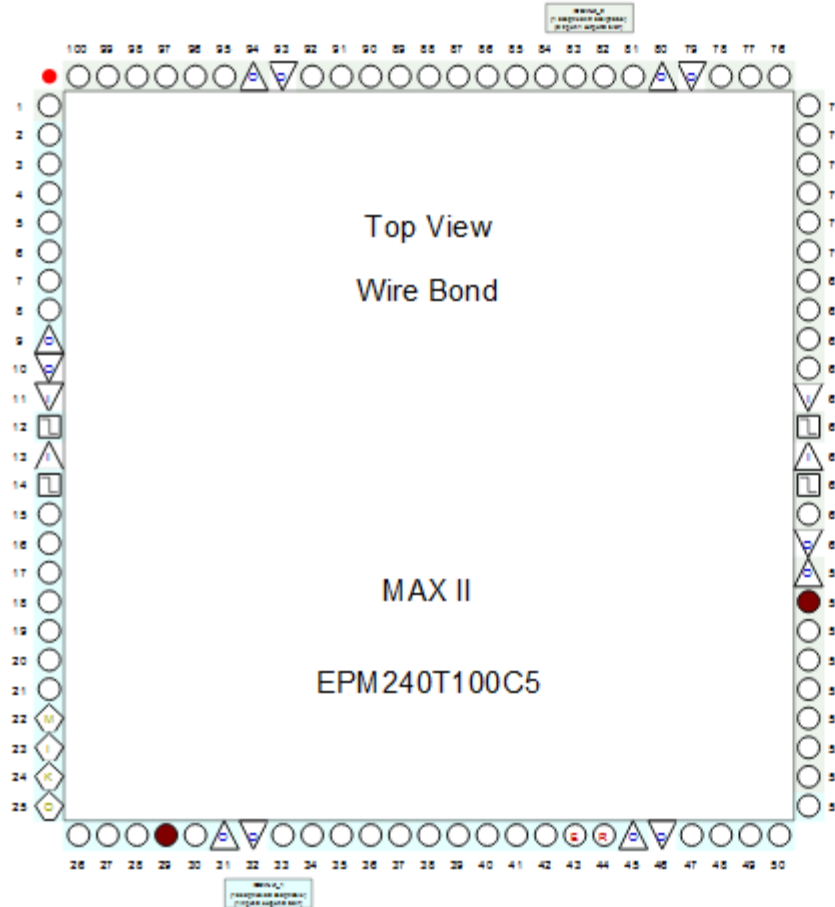
VHDL:

```
entity LED_NOT_KEY is  
    port (  
        key_in : in std_logic ;  
        led_out : out std_logic );  
end LED_NOT_KEY;  
  
architecture key_led_arch of LED_NOT_KEY is  
begin  
    led_out <= not key_in;  
end architecture;
```

Kivezetés tervező



A tervezésben szereplő ki- és bemeneteket az IC konkrét lábaihoz kell rendelni.



Symbol	Pin Type
○	User I/O
●	User assigned I/O
●	Filter assigned I/O
●	Unbonded pad
●	Reserved pin
E	DEV_OE
R	DEV_CLR
L	CLK_n
I	TDI
K	TCK
M	TMS
T	TDO
△	VCCINT
△	VCCIO

Node Name	Direction	Location	I/O Bank	Filter Location	I/O Standard	Reserved	Current Strength
in KEY_IN	Input	PIN_29	1	PIN_29	3.3-V LVTTTL		16mA (default)
out LED_OUT	Output	PIN_58	2	PIN_58	3.3-V LVTTTL		16mA (default)
<<new node>>							



C-M240 fejlesztői kártya

Gyártó: Shenzhen 21EDA Electronic Technology

CPLD: Altera MAX II EPM240T100C5N

Órajel: 50 MHz

VDD: 3,3 V

Perifériák: 8-digit kijelző, 8 LED, 4 +1 nyomógomb, 1 csipogó





10_key_digital_tube_display

```
module key_led(clk_50M,key,segment,digit);
input clk_50M; // system clock 50MHz input from pin 17 input.
input [3:0] key; // key1 key2 key3 key4 is the value of the input key code
output [3:0] digit; // digital control bit selection
output [7:0] segment; // digital tube segment code ABCDEFGH
reg [7:0] segment;
reg [3:0] digit;
reg [3:0] key_temp; // set a register

always@ (posedge clk_50M)
begin
key_temp<=key; // assign the value of the key to the register
case (key_temp)
4'b1110:segment<=8'b1111_1001; // segment code // KEY1 press to show 1
4'b1101:segment<=8'b1010_0100; // segment code // KEY1 press to show 2
4'b1011:segment<=8'b1011_0000; // segment code // KEY1 press to show 3
4'b0111:segment<=8'b1001_1001; // segment code // KEY1 press to show 4
endcase
end

always@ (posedge clk_50M)
begin
case(key_temp) // bit selected signal
4'b0111:digit<=4'b0111; // press KEY4 4th digit display
4'b1011:digit<=4'b1011; // press KEY3 3th digit display
4'b1101:digit<=4'b1101; // press KEY2 2th digit display
4'b1110:digit<=4'b1110; // press KEY1 1th digit display
endcase
end
endmodule
```

Ez a mintaprojekt
a lenyomott gomb
sorszámát írja ki a
7-szegmenses
kijelzőkön



15_fdiv_LED: LED villogtatás

Ebben a Verilog mintapéldában az 50 MHz-es órajelet 1 Hz-re osztjuk le, s egy LED-et villogtatunk vele.

```
module ledwater (clk_50M, led_out);
input clk_50M;           // system clock input 50M input pin17
                        // means every second to change 50,000,000 Hz

output led_out;        // flash every second

reg [24:0] count;      // Counter counts up to 25,000,000
reg div_clk;          // Frequency divider output (1 Hz)
reg led_out;          // LED port status

// Divide the counter. Get a second frequency
always @ (posedge clk_50M)
begin
if (count == 25000000)
begin
// clock changes 50,000,000 HZ every second
// We now use the count counter to add it to 25,000,000HZ
div_clk <= ~ div_clk; // here we get a 0.5 second change signal.
// so a cycle is 1Hz is one second.
count <= 0;           // clear the counter.
end
else
count <= count + 1;  // counter from the increase.
led_out <= div_clk;  // use the divided frequency to blink a LED (1 Hz)
// In the LED lights above the show.

end
endmodule
```



Cypress PSOC mikrovezérlők

PSOC = Programozható System-on-chip
(CPLD + mikrovezérlő)

AZ évek során több változatot fejlesztettek ki:

PSoC 1: M8C (8-bites)

PSoC 3: 8051 (8-bites)

PSoC 4: ARM Cortex-M0 (32-bites) - BLE változatban is!

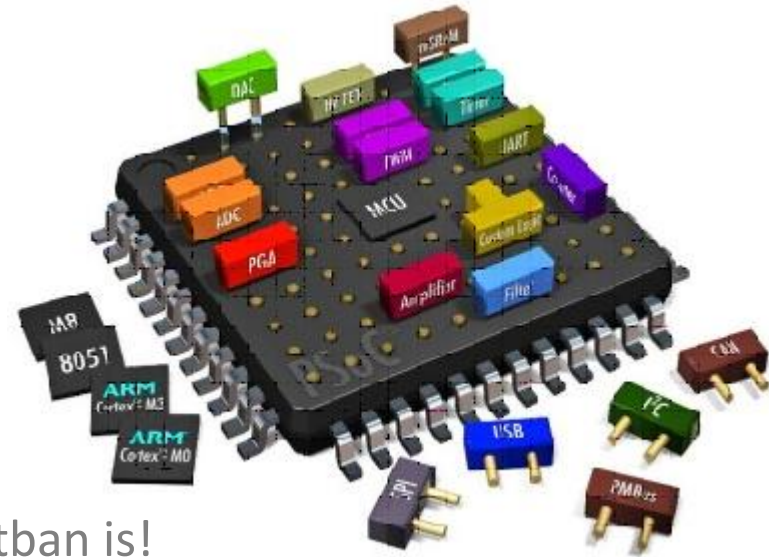
PSoC 5LP: ARM Cortex-M3 (32-bites)

PSOC 6: ARM Cortex M4F + M0+ + BLE (32 bites)

Fejlesztői környezet:

PSOC Designer – csak PSOC 1-hez

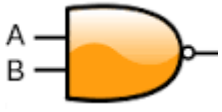
PSOC Creator – PSOC 3, 4, 5, 6-hoz



A számtalan fix funkciós periféria mellett az univerzális digitális blokkokból **egyedi perifériákat**, vagy **kiegészítő logikai áramköröket** is kialakíthatunk.

Bennünket most mindez elsősorban a CPLD miatt érdekel...

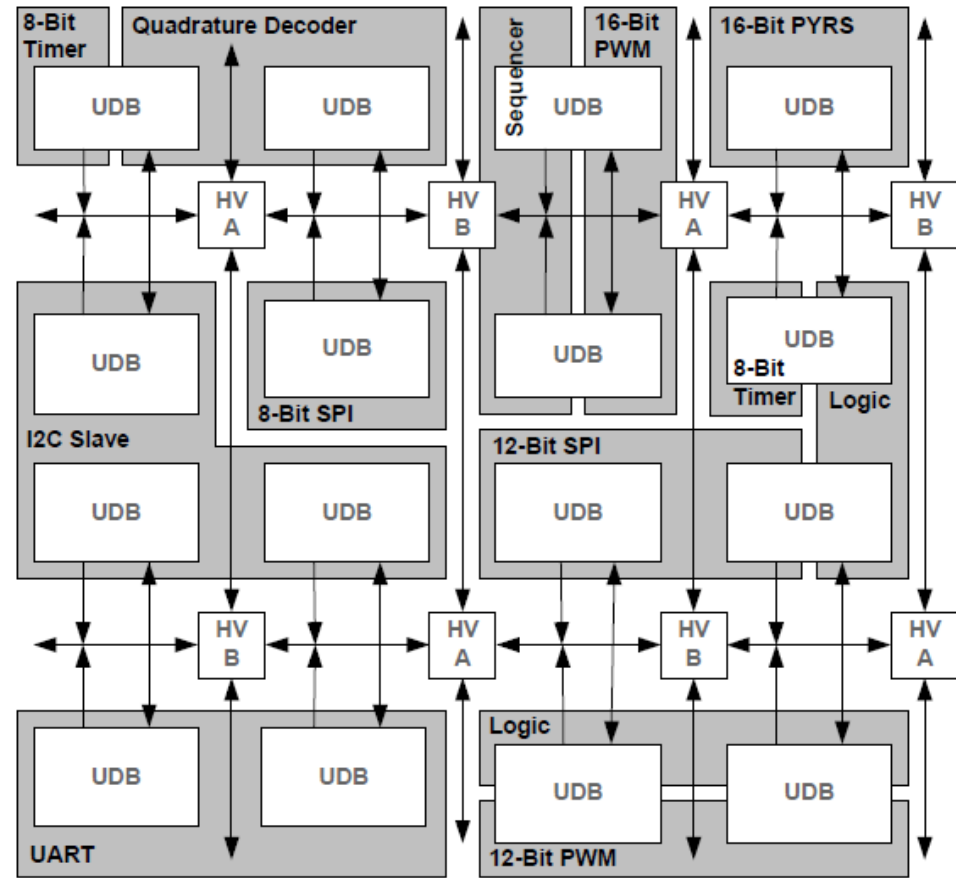
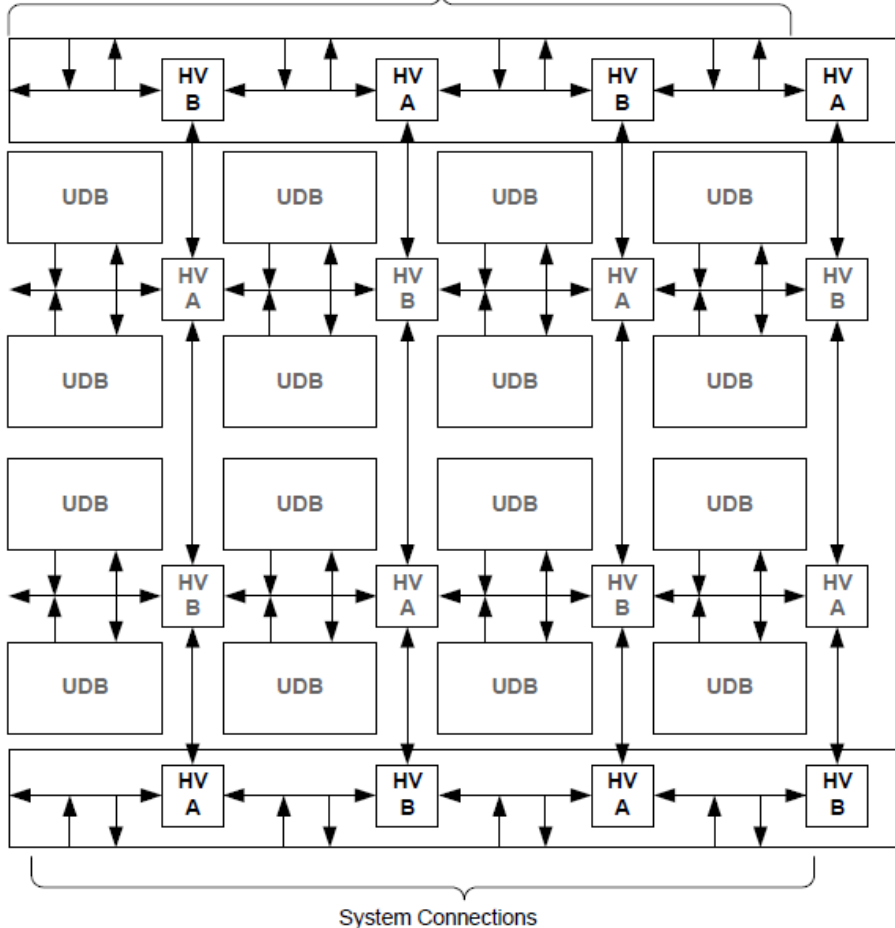
A PSOC4 és PSOC5 mikrovezérlők 3,3 V-os és 5 V-os tápfeszültségen is működhetnek!



Univerzális digitális blokkok

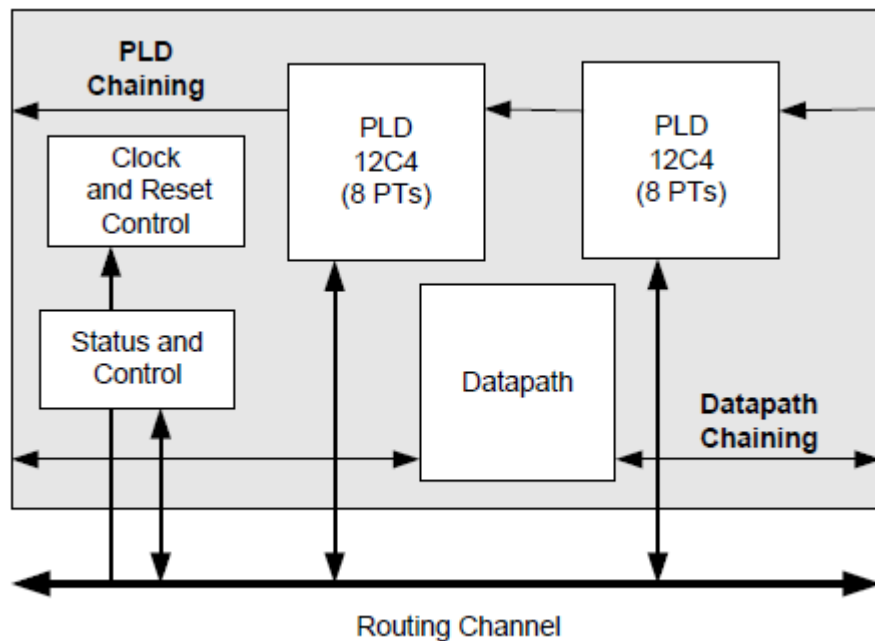
Az Univerzális Digitális Blokkokból a CY8C4245 PSOC 4 mikrovezérlők 4 db-ot, a CY8C5868, illetve CY8C5888 PSOC 5LP mikrovezérlők 24 db-ot tartalmaznak.

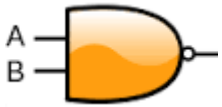
System Connections





Egy UDB felépítése





Egy 12C4 PLD blokk felépítése

PLD blokk jellemzők:

- 12 bemenet
- 8 szorzat tag (product term)
- 4 makrocella kimenet

$$X = (A \& B) \mid (\sim C \& D)$$

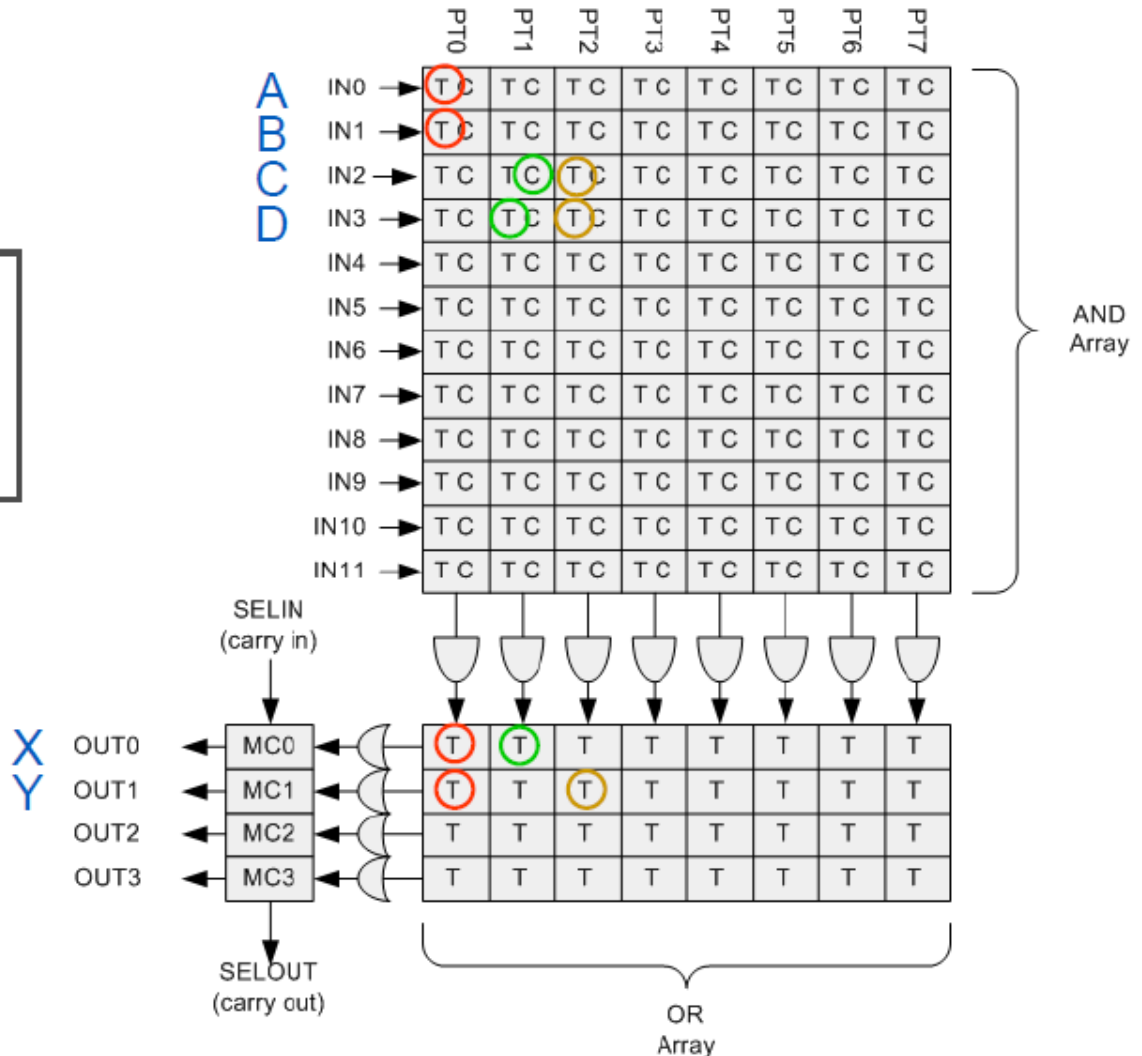
$$Y = (A \& B) \mid (C \& D)$$

TC = True vagy Complement

AND = logikai ÉS kapuk

OR = logika VAGY kapuk

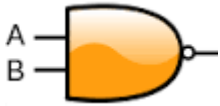
MC = makrocella



X
Y

AND
Array

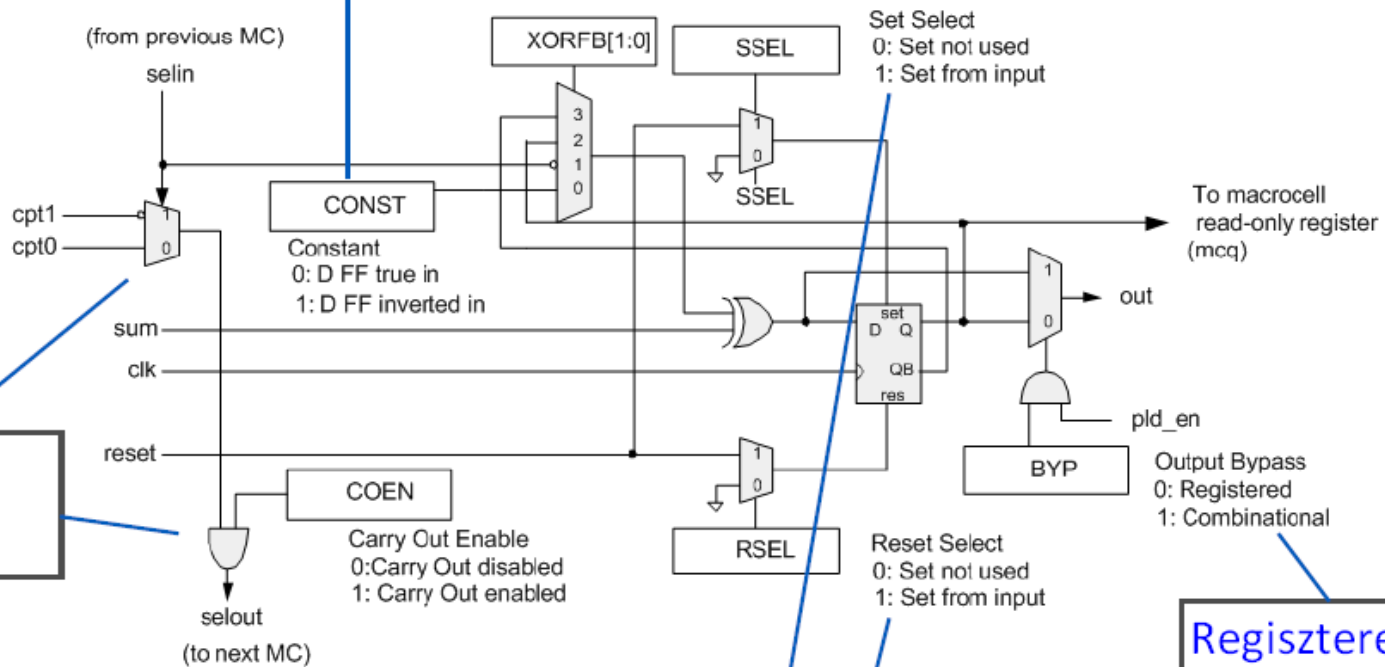
OR
Array



Makrocella felépítése

D vagy T flip-flop
választható invertálás

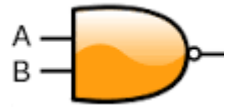
XOR Feedback
00: D FF
01: Arithmetic (Carry)
10: T FF on high
11: T FF on low



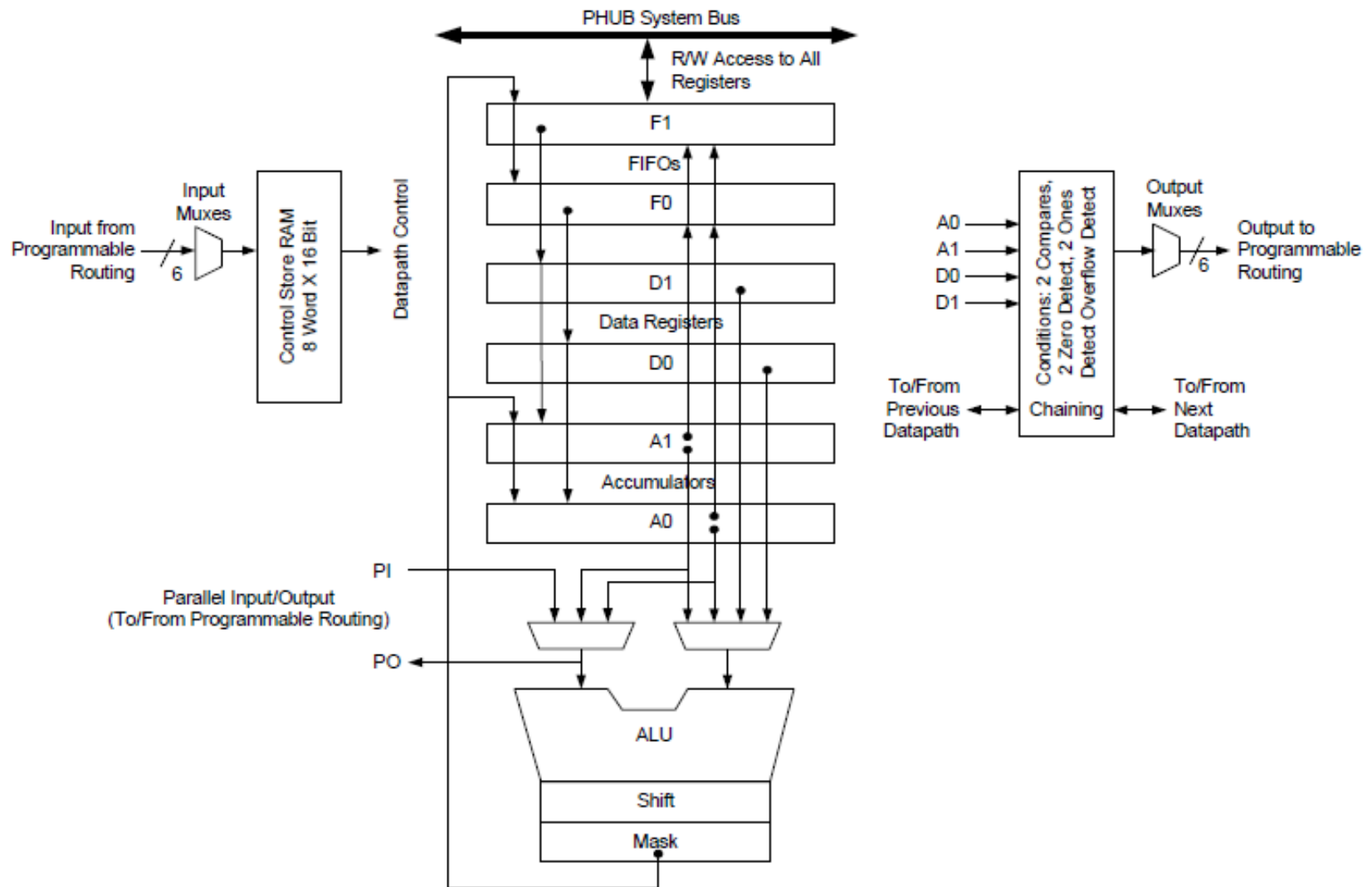
Átvitel lánc
logika

Aszinkron
Set / Reset

Regiszteres vagy
kombinációs



Adatút (Data path)





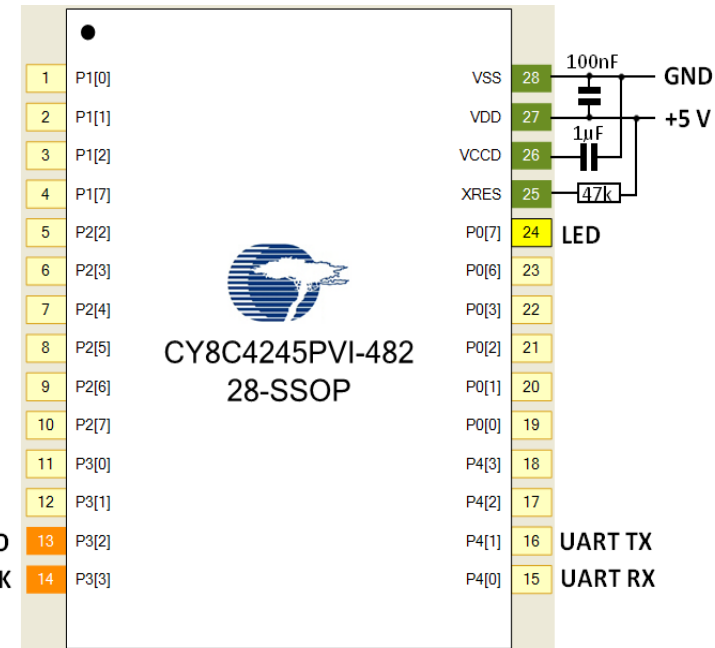
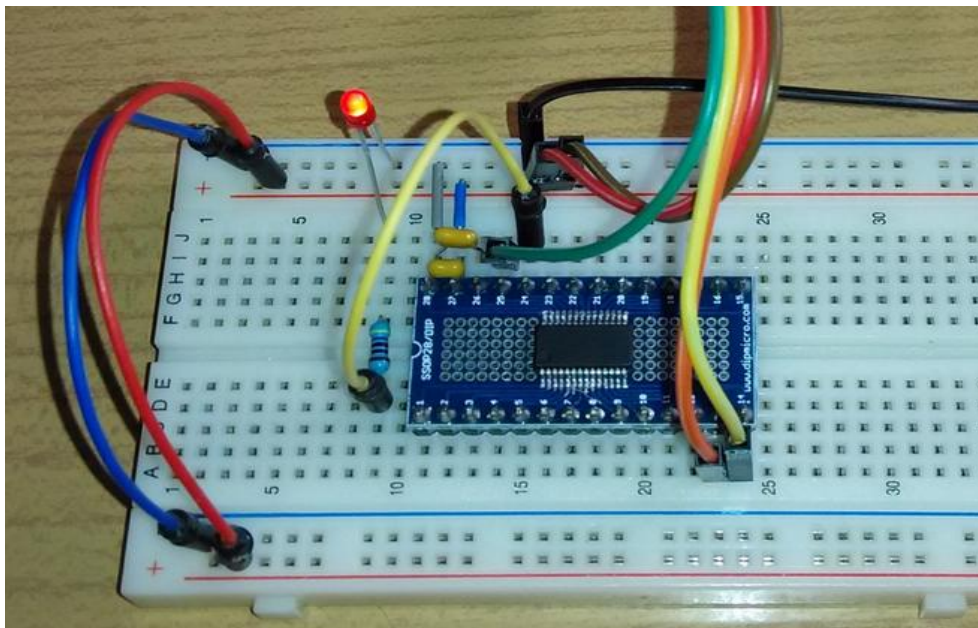
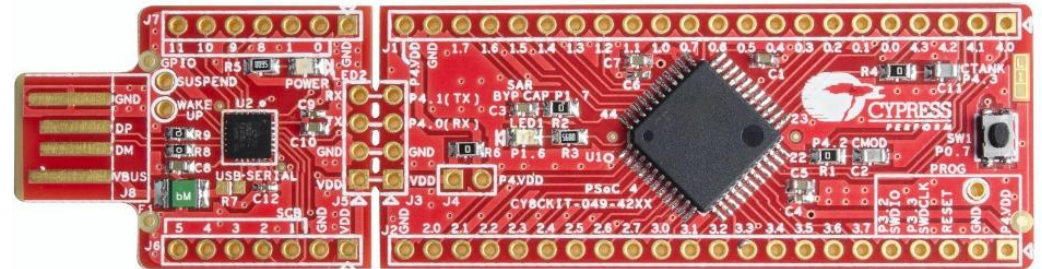
PSOC 4 fejlesztőeszközök

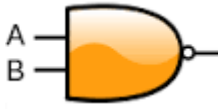
CY8CKIT-042 PSoC® 4 Pioneer Kit



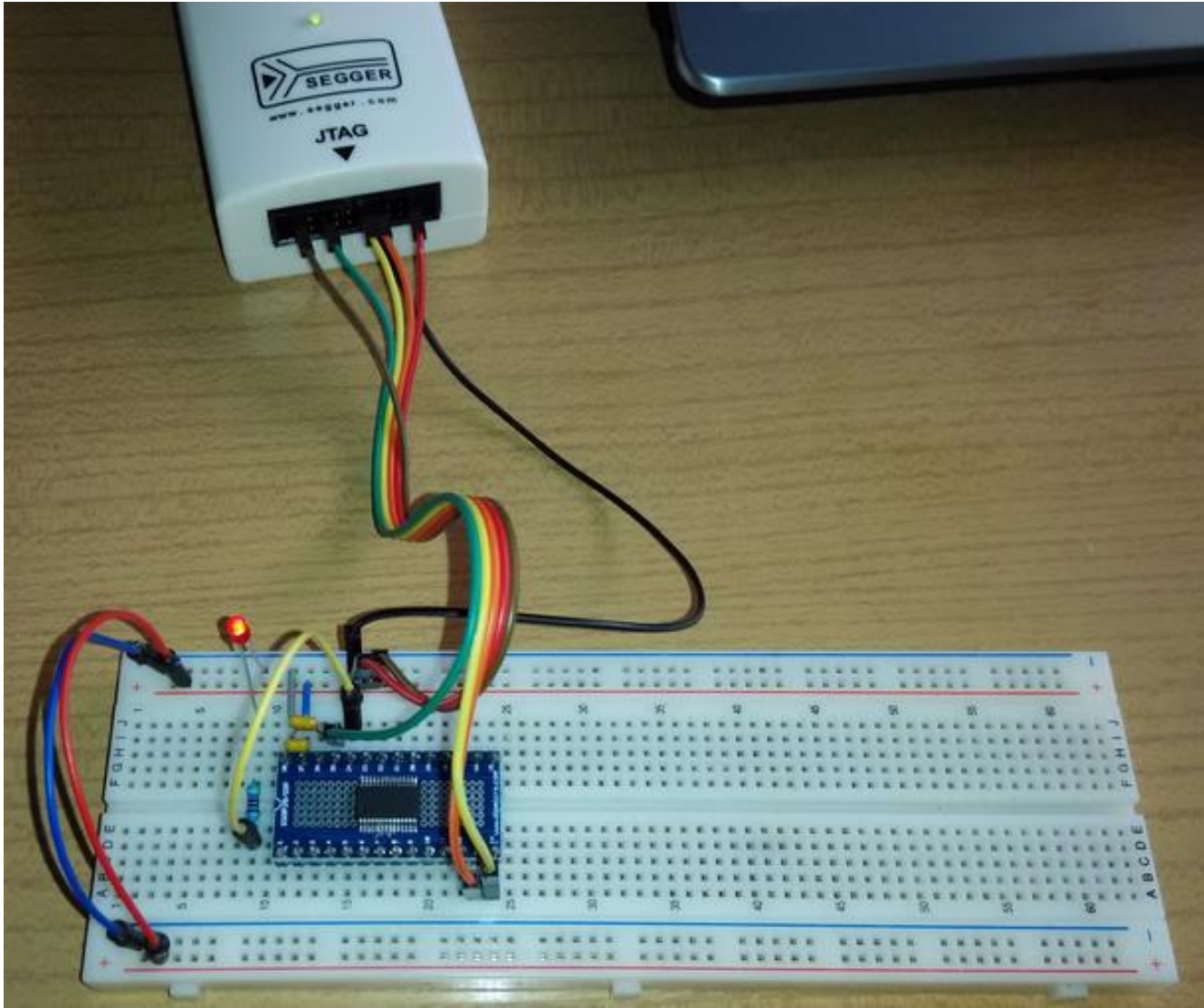
CY8CKIT-049 PSoC® 4 Prototyping Kits

CY8C4245AXI-483 (44 pin TQFP)
48 MHz, 32k/4k, 4 UDB, 12-bit ADC



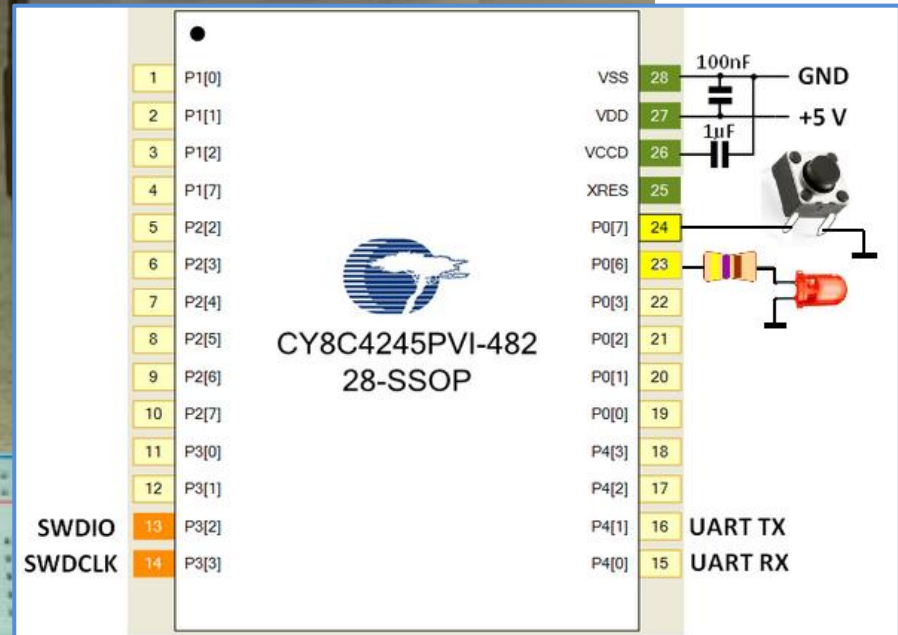
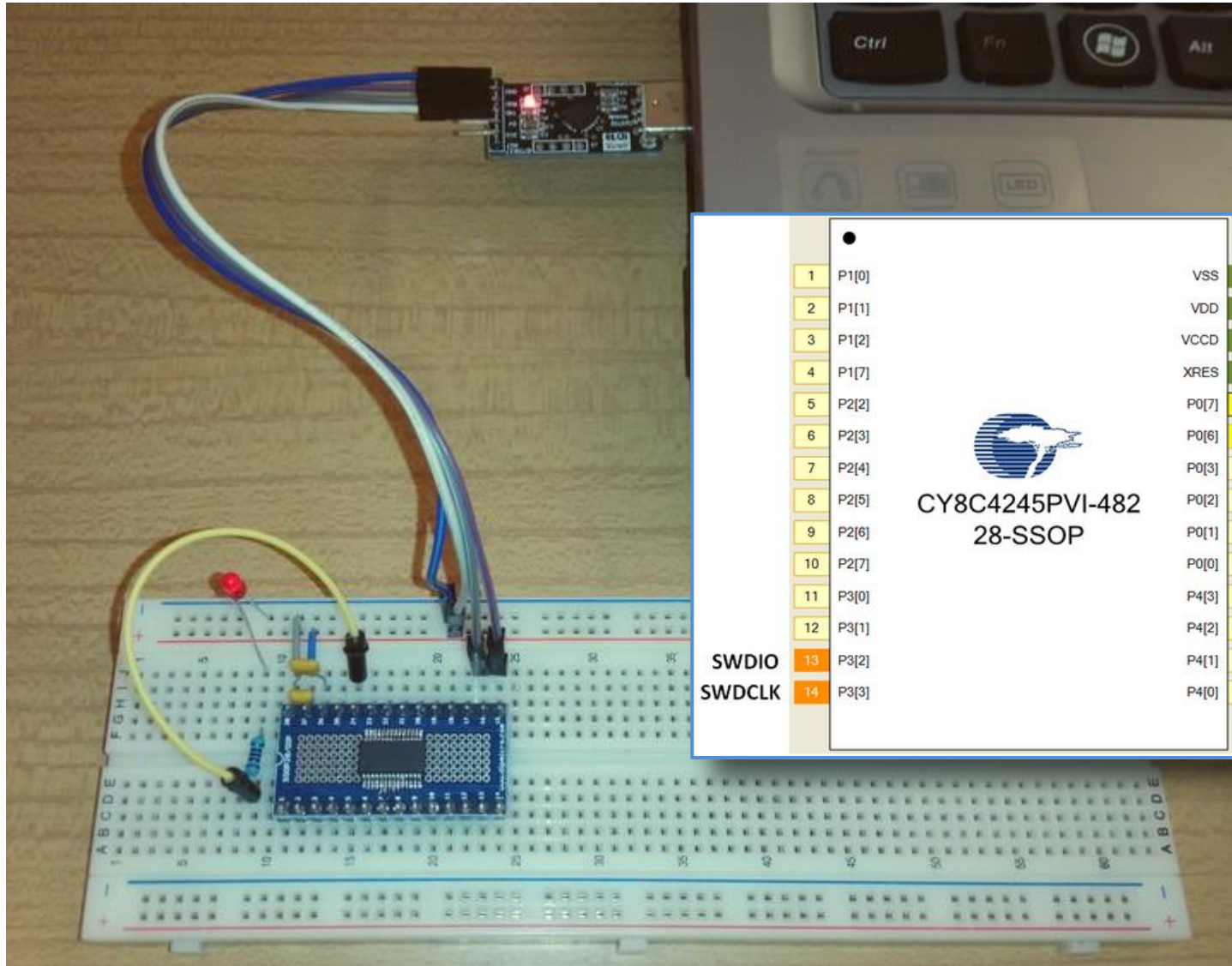


Programletöltés JTAG eszközzel





Programletöltés bootloaderrel





A „lélegző” LED esete két PWM modullal...

Két PWM egységet konfigurálunk, egymáshoz közeli frekvenciával (98 és 98.4 Hz) és 40 % körüli kitöltéssel.

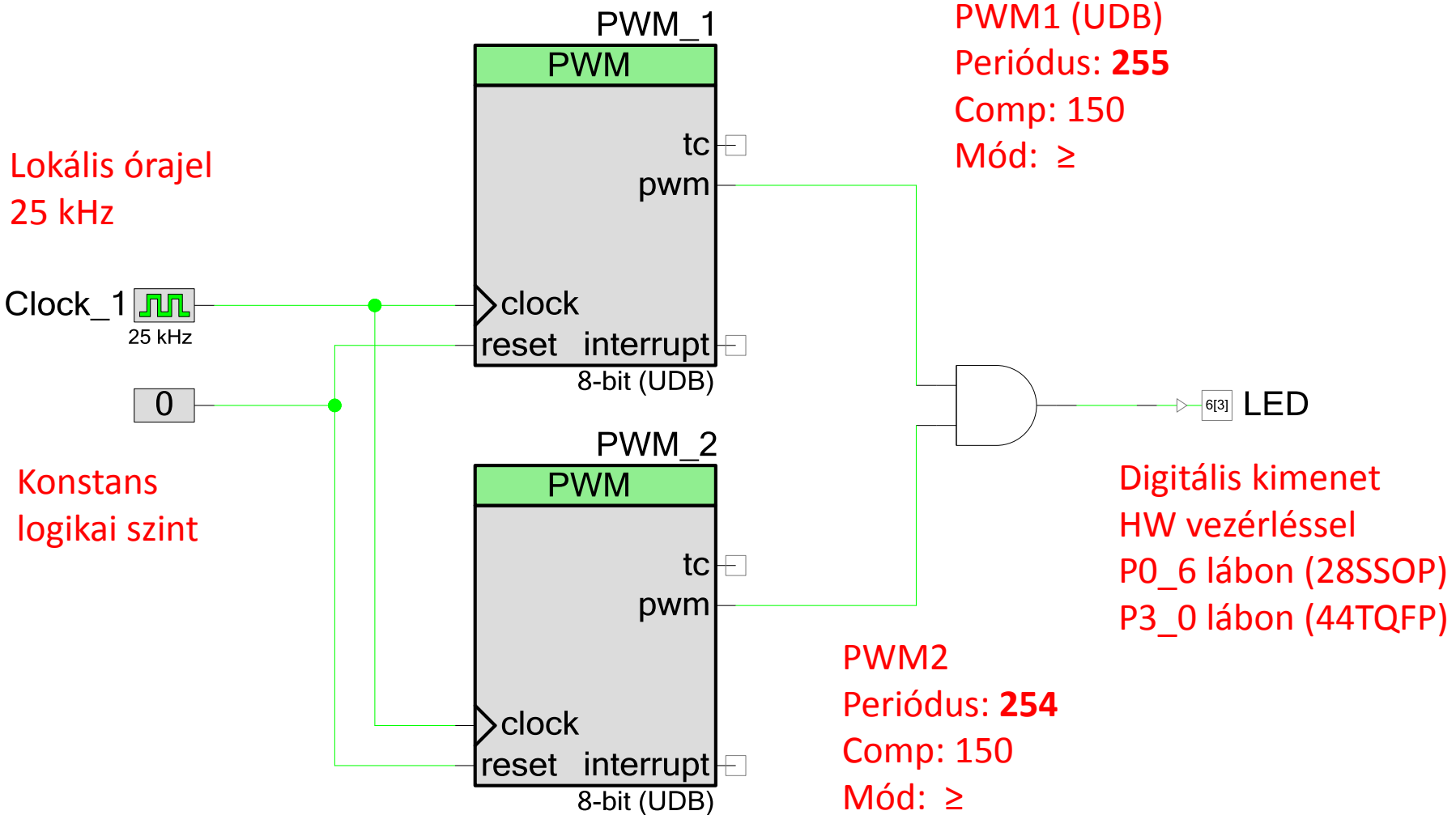
Ha a kimeneteket ÉS kapcsolatba hozzuk, akkor az eredő jel kitöltése az egymáshoz közeli frekvenciák összelebegésének megfelelően periodikusan változik, 0.4 Hz-es frekvenciával.

Ha ezt a jelet vezetjük a LED4-hez kapcsolódó kimenetre, akkor a LED fényereje periodikusan változik.

A LED villogásának elkerülésére a PWM frekvenciát 100 Hz körülire kell megválasztani. 8-bites PWM esetén ehhez 25 kHz-es bemenő órajelet kell definiálni.

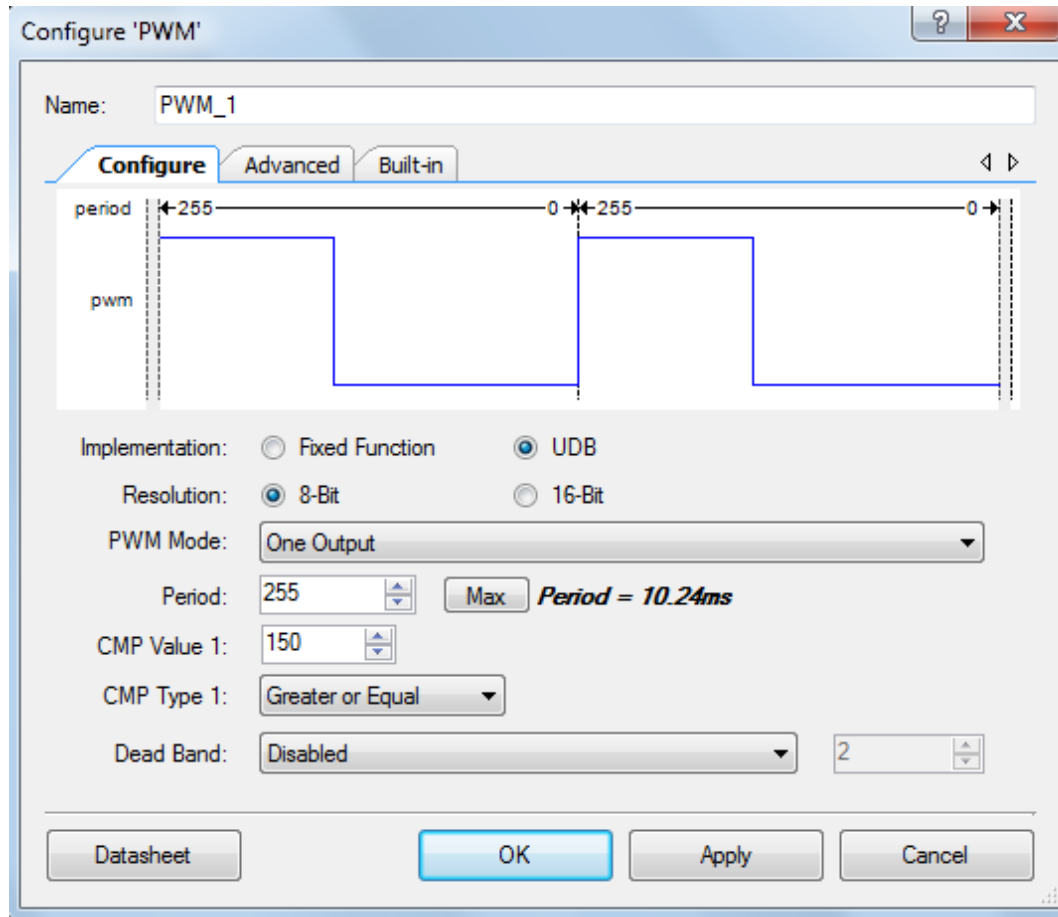


A „lélegző” LED esete két PWM modullal...

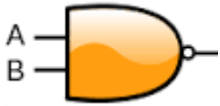


A
B

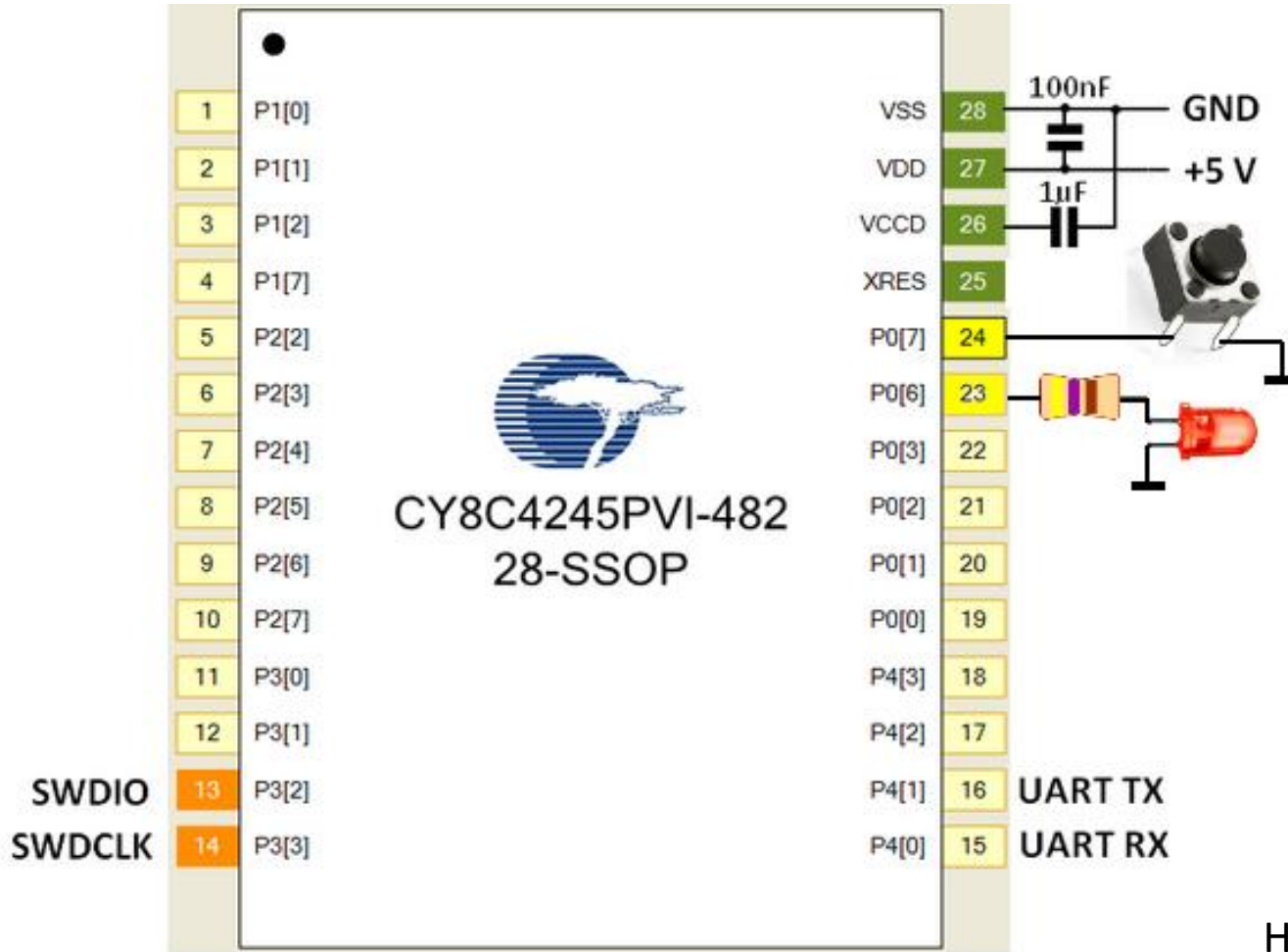
A PWM_1 modul konfigurálása



PWM_2 beállítása hasonlóan történik, de a periódus ennél 254 legyen!



PSoC I/O hozzárendelés



Ha a **Lock** opció nincs bejelölve, szintetizálásnál automatikus láb kiosztás lesz!

Alias	Name /	Port	Pin	Lock
Pin_1	P0[6] SRSS:ext_clk, SCB1:spi_clk		23	<input checked="" type="checkbox"/>



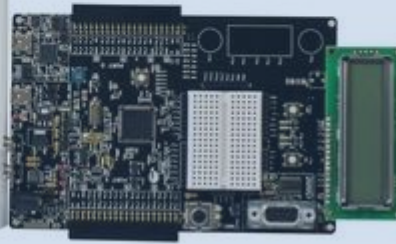
PSOC 5 fejlesztőeszközök



CY8CKIT-050 PSoC® 5LP Development Kit

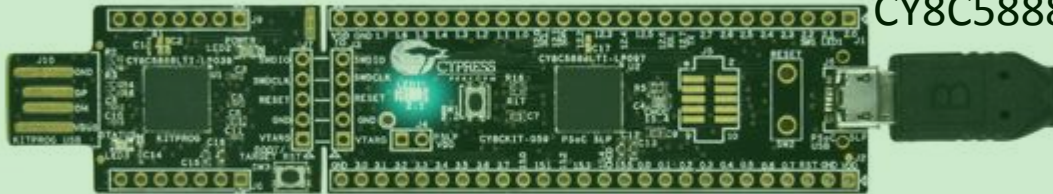
CY8C5868AXI-LP035 (100 pin TQFP)

USB/DMA/12-bit SAR és 20-bit delta-sigma ADC
24 UDB (digitális blokk), 4 analóg blokk

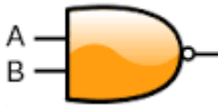


CY8CKIT-059 PSoC® 5LP Prototyping Kit

CY8C5888LTI-LP097 (68 pin QFN)

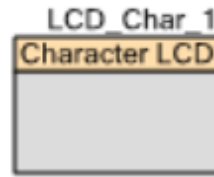


USB/DMA/12-bit SAR és 20-bit delta-sigma ADC
24 UDB (digitális blokk), 4 analóg blokk

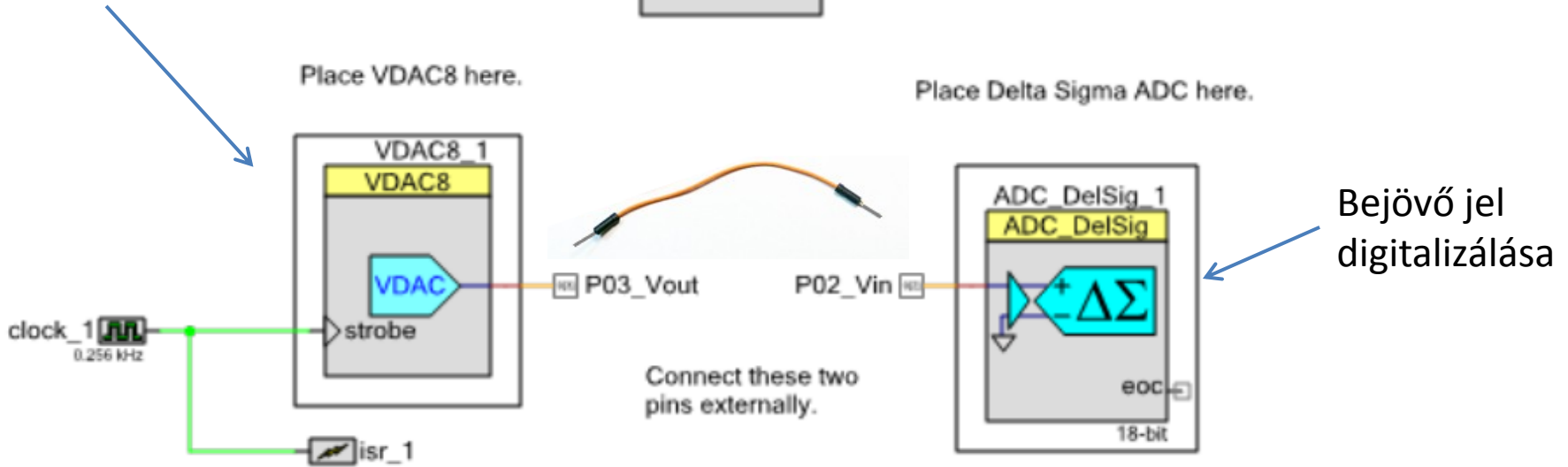


LCD „oszcilloszkóp” projekt

Hullámforma generálás táblázatból, VDAC-kal



Oszlopdiagram formájú megjelenítés



VDAC8_1 Parameters

- * Range -> 0 - 1.024 (4mV/bit)
- * Value -> 100 mV
- * Speed -> Slow Speed
- * Data_Source -> CPU_or_DMA
- * Strobe_Mode -> External

ADC_DelSig_1 Parameters

- * Conversion Mode -> 3 Multi Sample (Turbo)
- * # Configs -> 1
- * Resolution -> 18
- * Conversion Rate -> 128
- * Input Mode -> Single Ended
- * Input Range -> Vssa to 1.024V (0 to Vref)
- * Buffer Gain -> 1
- * Buffer Mode -> Rail-to-Rail
- * Reference -> Internal Vref
- * Clock Source -> Internal



LCD „oszcilloszkóp” projekt

1. Csatlakoztassuk a P2 portra az LCD kijelzőt!
2. Kössük össze a P0_2 és P0_3 kivezetéseket!
3. Programozáshoz dugjuk a J1 csatlakozóba az USB kábelt!
4. A program indítása után ellenőrizzük, hogy az alábbi videón láthatóhoz hasonló képet látunk a kijelzőn!

