



## ISE makró (saját alkatrész) készítése

1. Makró (saját alkatrész) hozzáadása meglévő projekthez .....	2
1.1. Kapcsolási rajz alapú makró készítése .....	2
1.2. Kapcsolási rajz alapú saját makró javítása .....	4
1.3. VHDL alapú makró hozzáadása.....	7
1.4. VHDL alapú makró ellenőrzése .....	9
1.5. VHDL alapú makró hozzáadásánál keletkezett hiba javítása .....	10
1.6. Kapcsolási rajz alapú makrók hozzáadása .....	12
2. A Digitális Technika II laborgyakorlatok alatt használt makrók leírásai .....	13
2.1. Clk_divide .....	14
2.2. Idozito.....	15
2.3. Prellm_oro .....	17
2.4. Bin_to_7_segment_x4.....	18
2.5. Bin_to_7_segment_x8.....	19
2.6. hex_7seg.....	20
2.7. mux8_4bit.....	21
3. Makrók használata .....	22
3.1. Multiplex vezérlésű hétszegmenses kijelző modulok használata .....	22
3.1. Kapcsolási rajz alapú hétszegmenses dekóder használata .....	23



## 1. Makró (saját alkatrész) hozzáadása meglévő projekthez

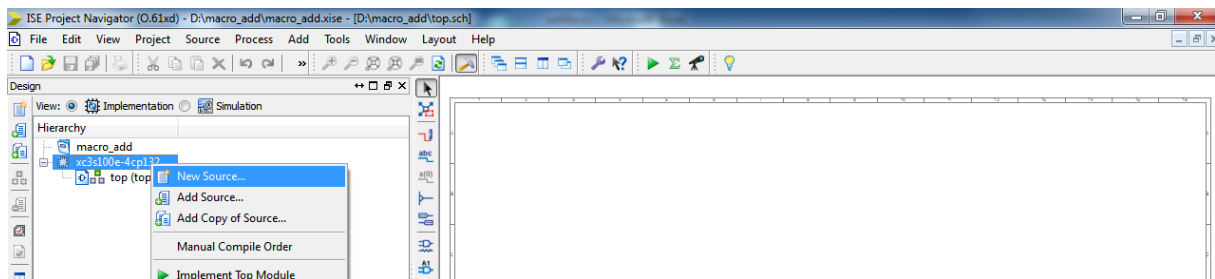
Minden az ISE kapcsolási rajz szerkesztőjében használt makróhoz két fájl tartozik. Az egyik egy szimbólumfájl (\*.sym), ami leírja a makró kapcsolási rajzon való megjelenését. A másik a működést leíró fájl, ami lehet kapcsolási rajz (\*.sch), vagy HDL (\*.vhd) alapú.

A meglévő makrókat mindig egy már létező projekthez lehet hozzáadni. A makró hozzáadása előtt, győződjünk meg róla, hogy a projekt tartalmaz-e **top modul** szintű kapcsolási rajzot.

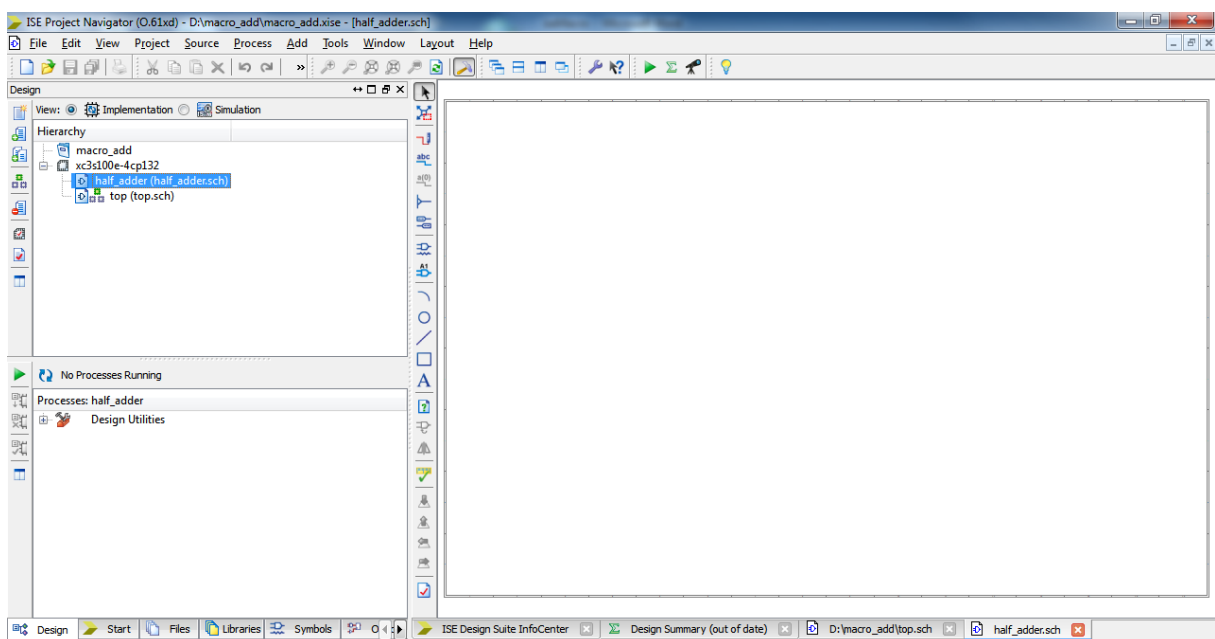
A példában a külső makrók forrása a „d:\macros\” és a projektünk a „d:\macro\_add” könyvtárban található.

### 1.1. Kapcsolási rajz alapú makró készítése

Első lépésként készítsük el a makró működését leíró kapcsolási rajzot (half\_adder.sch). Ha a készítendő makrónk a későbbiekben egy nagyobb kapcsolási rajz része lesz, célszerű létrehozni ennek üres **top modul** szintű kapcsolási rajzát.

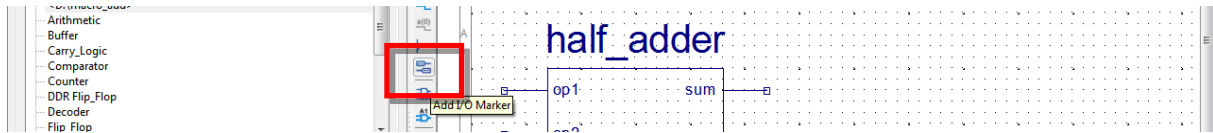


Nyissuk meg szerkesztésre a kapcsolási rajzot.

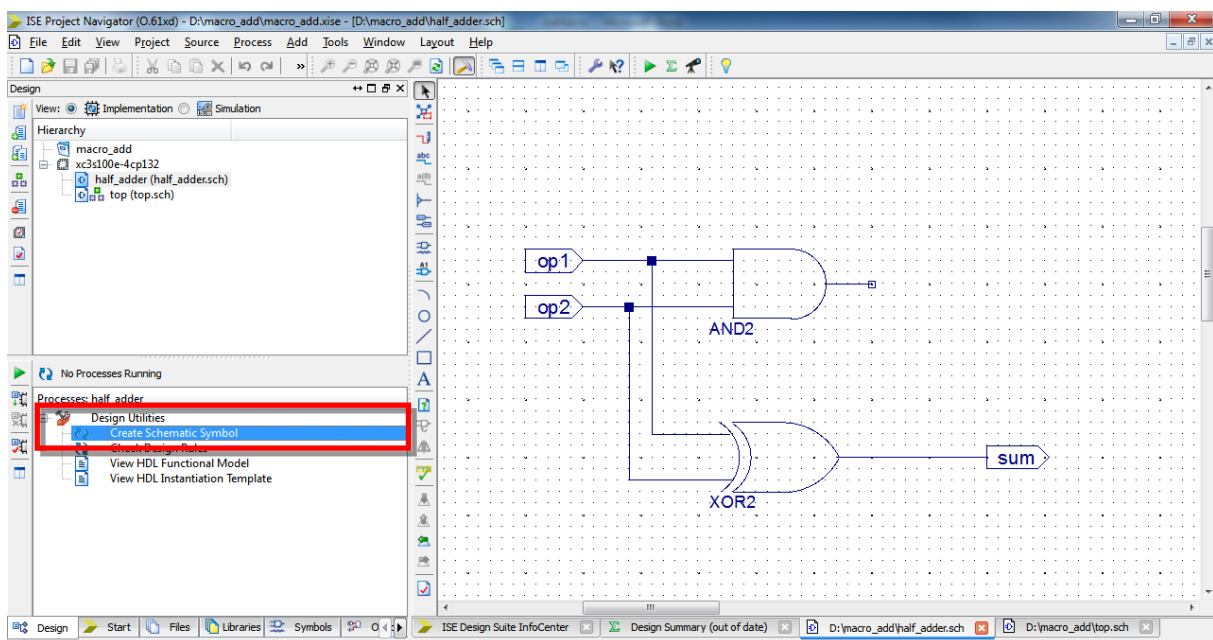




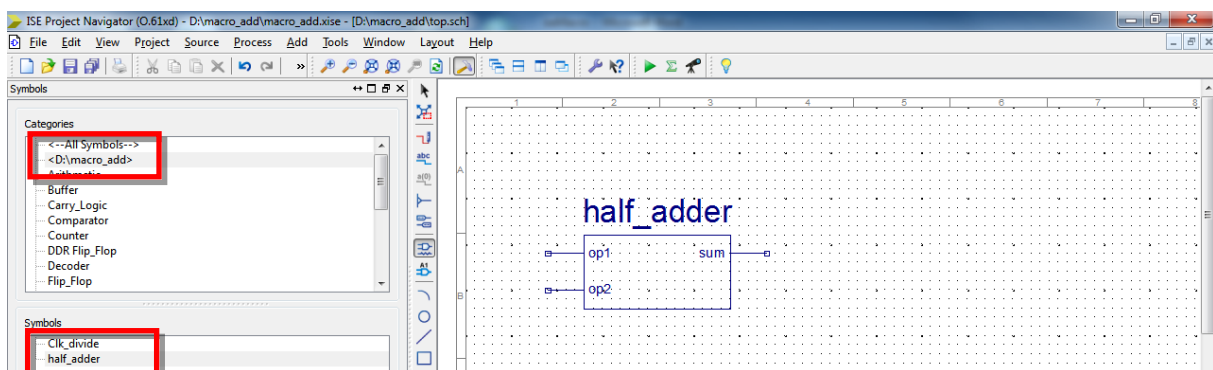
A működést leíró kapcsolási rajz elkészítése után (fél összeadó áramkör), meg kell adni a makró ki és bemeneti portjait. Ezt az „ADD” I/O Marker” gombbal lehet megtenni. Nevezzük el az I/O markerekhez kapcsolódó vezetékeket, ezek lesznek a makró ki- és bemeneti port nevek.



A kapcsolási rajzot kijelölve, megjelenik a „Create Schematic Symbol” opció. Ezt a műveletet futtatva elkészül a makró (létrejön egy sym kiterjesztésű fájl a kapcsolási rajz mellett).

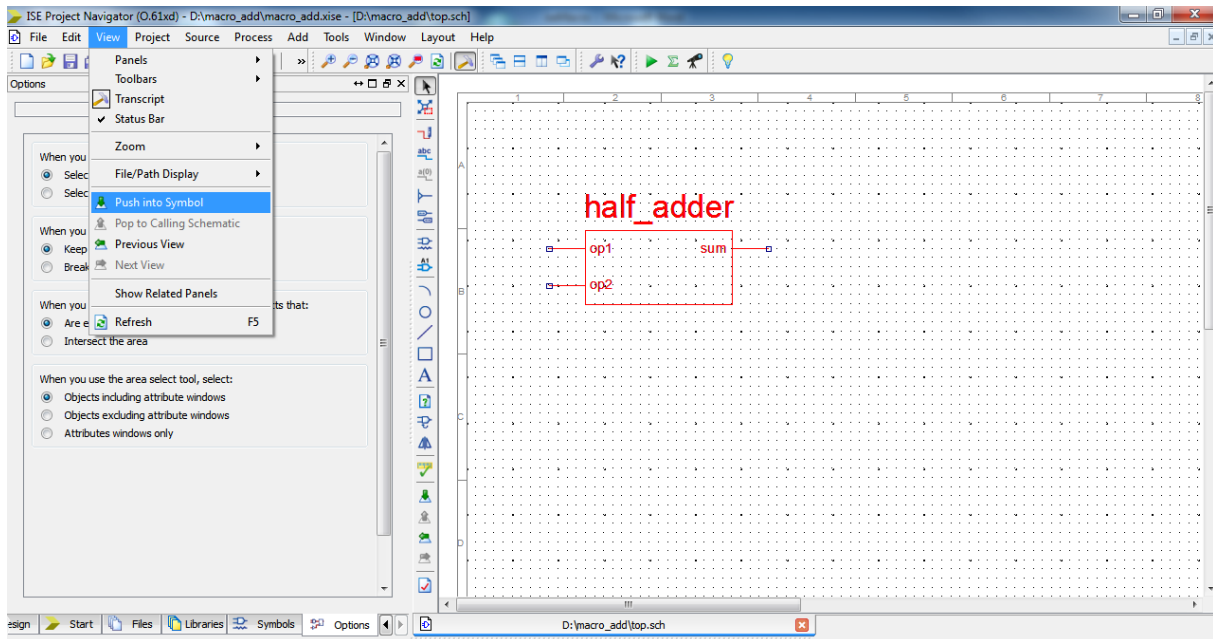


A szimbólumok között ezután már megtalálható a makró.





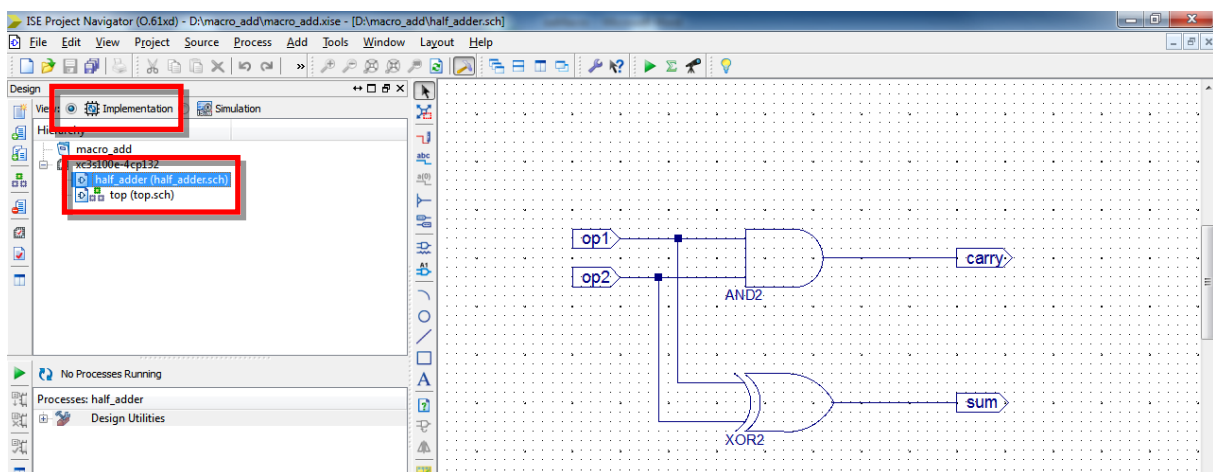
A makró tartalmát a „View” menü „Push into Symbol” gombjával lehet megtekinteni. Előtte szükséges kijelölni a makró, a kapcsolási rajzon!



## 1.2. Kapcsolási rajz alapú saját makró javítása

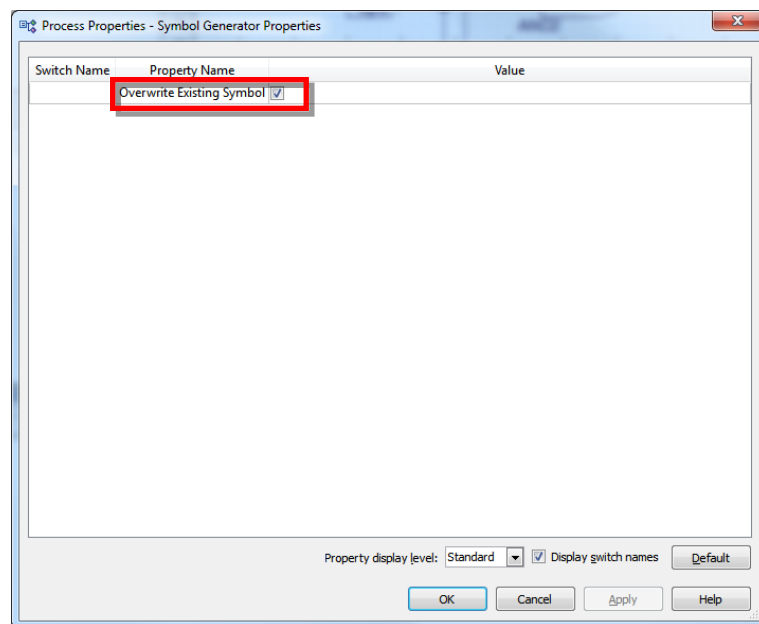
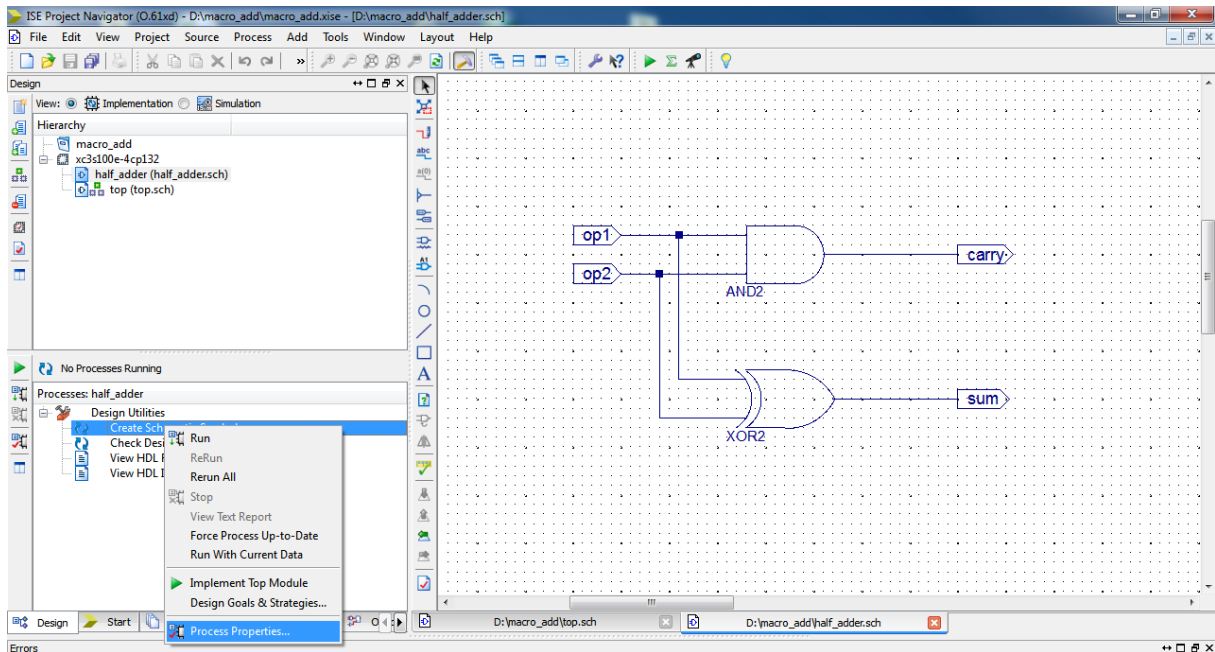
Ha a változtatás nem érinti a makró ki- és bemeneti portjait, akkor a „Push into Symbol” paranccsal megnyitva lehetőség van a makró módosítására. A változásokat minden esetben el kell menteni.

Amennyiben a változás a makró megjelenésére is kihat, például új portot kell hozzáadni, törölni, vagy port nevet kell változtatni, akkor a következőképpen kell eljárni. Első lépésként meg kell nyitni a makró tartalmazó sch fájlt. Ezt a „Push into Symbol” paranccsal, vagy a „Design” ablakban a kapcsolási rajzra kétszer kattintva lehet megtenni. Végezzük el a változtatásokat a makró fájlban. A példában a régi rajzot kiegészítettük egy új porttal.



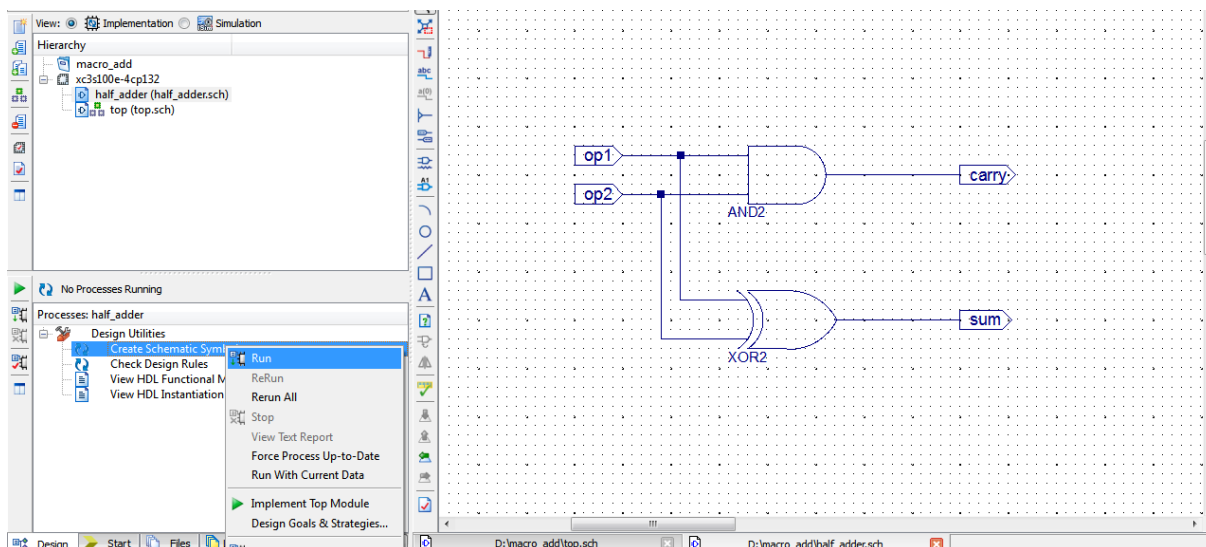


A módosítások mentése után a „Create Schemetic Symbol” parancs beállításáiban engedélyezni kell a létező szimbólum felülírását.

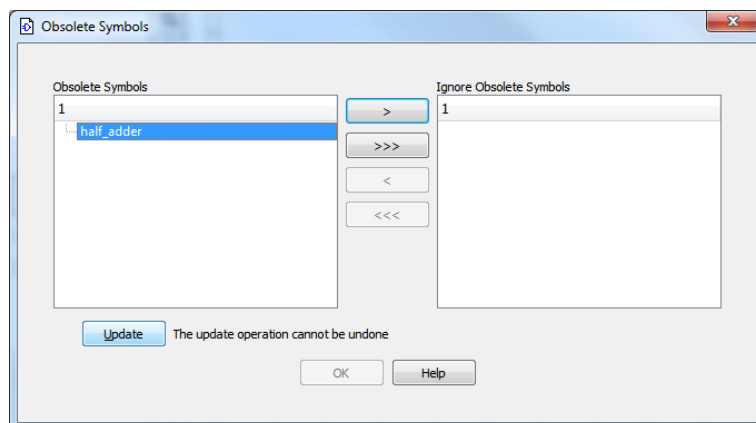




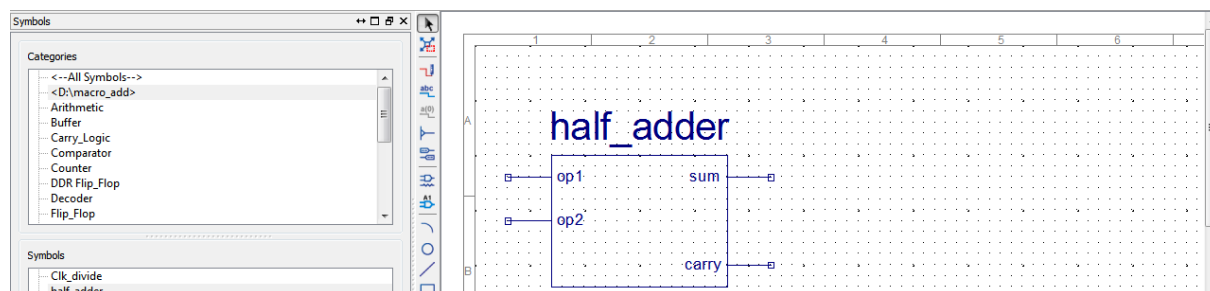
A „Run” parancs futtatásával módosul a makró megjelenését leíró szimbólum fájl.



Ha valamelyik kapcsolási rajzban használunk egy makrót, majd a makró megjelenésére kiható változtatást hajtunk végre, a kapcsolási rajz szerkesztőben „Obsolete Symbols” ablak jelenik meg. Az „Update” paranccsal frissíthetjük a kijelölt objektumra vonatkozó módosításokat.



**Ha a makró újrafordítása után nem kínálja fel a program a makró szimbólum frissítését, és a kapcsolási rajzban nem is változik meg a makró megjelenése, akkor újra kell indítani az ISE fejlesztőkörnyezetet (előtte mindent mentünk el!). Az újraindulás után már a módosított szimbólumnak kell megjelennie.**





### 1.3.VHDL alapú makró hozzáadása

Másoljuk át a makró fájlokat a projektkönyvtárba. A makró megjelenését és működését egy azonos nevű \*.sym és egy \*.vhd kiterjesztésű fájl írja le.

Name	Ext	Size	Date	Attr
[.]	<DIR>		2011.10.13 13:52	—
Clk_divide	sym	1 618	2011.08.09 17:06	-a-
Clk_divide	vhd	2 170	2011.08.09 16:57	-a-

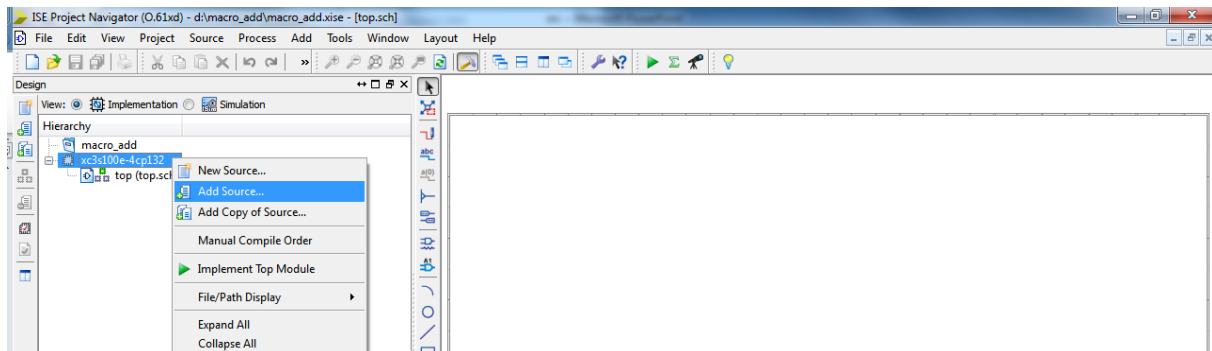
  

Name	Ext	Size	Date	Attr
[.]	<DIR>		2011.10.13 13:52	—
[_xmsgs]	<DIR>		2011.10.13 13:50	—
[iseconfig]	<DIR>		2011.10.13 13:50	—
macro_add	xise	1 741	2011.10.13 13:50	-a-

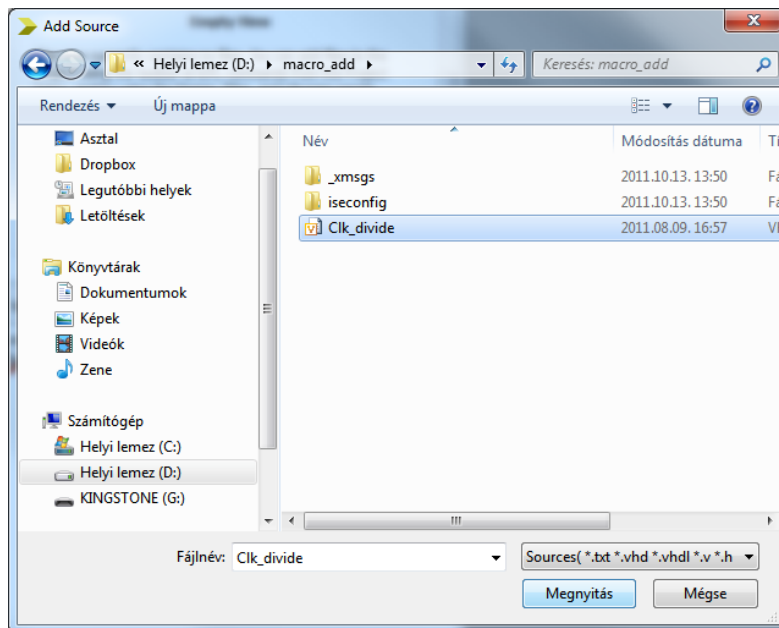
Name	Ext	Size	Date	Attr
[.]	<DIR>		2011.10.13 13:53	—
[_xmsgs]	<DIR>		2011.10.13 13:50	—
[iseconfig]	<DIR>		2011.10.13 13:50	—
Clk_divide	sym	1 618	2011.08.09 17:06	-a-
Clk_divide	vhd	2 170	2011.08.09 16:57	-a-
macro_add	xise	1 741	2011.10.13 13:50	-a-

A \*.vhd fájlt az „Add Source” paranccsal a projekthez kell adni. Fontos, hogy a fájlok másolása után hajtsuk végre ezt a lépés, mert az ISE ellenkező esetben a \*.vhd fájlt felülírhatja egy template fájljal.

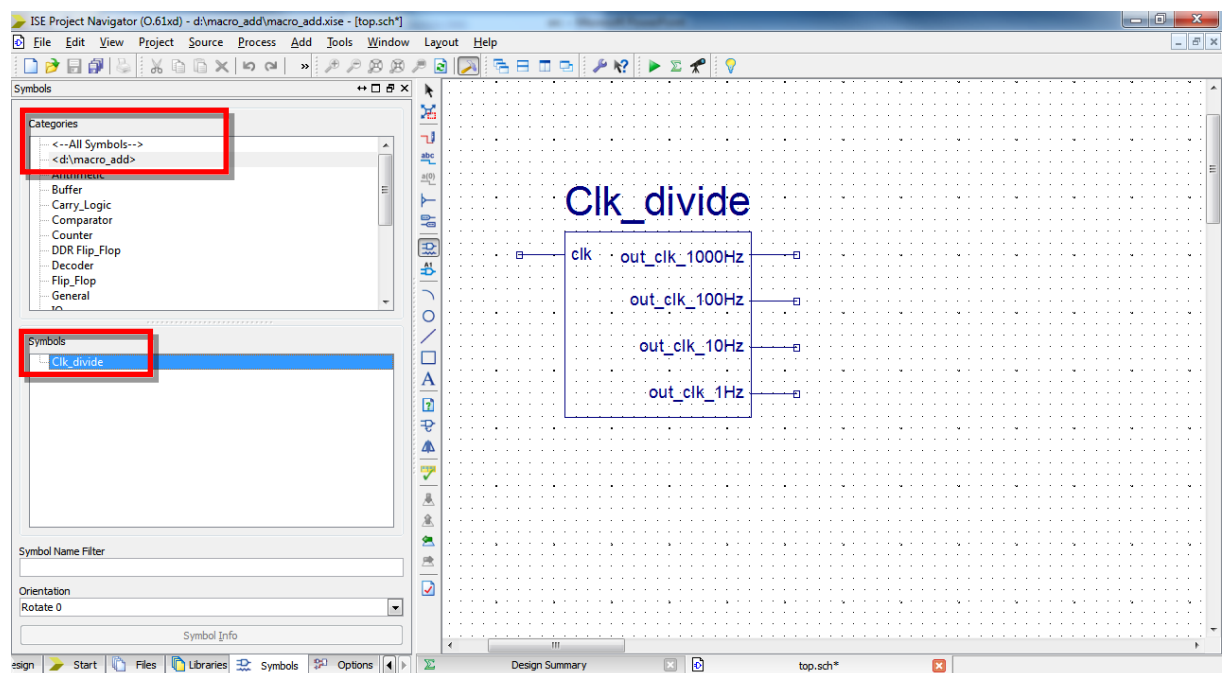




A vhd fájl megkeresése. Mivel korábban átmásoltuk a projektkönyvtárunkban kell lennie.



Végül a szimbólumok között megjelenik az új makró.

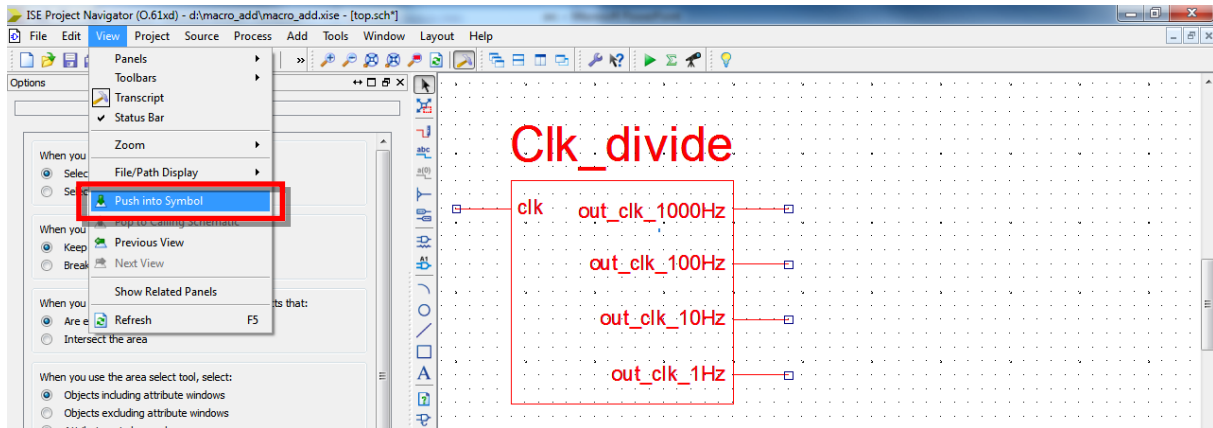




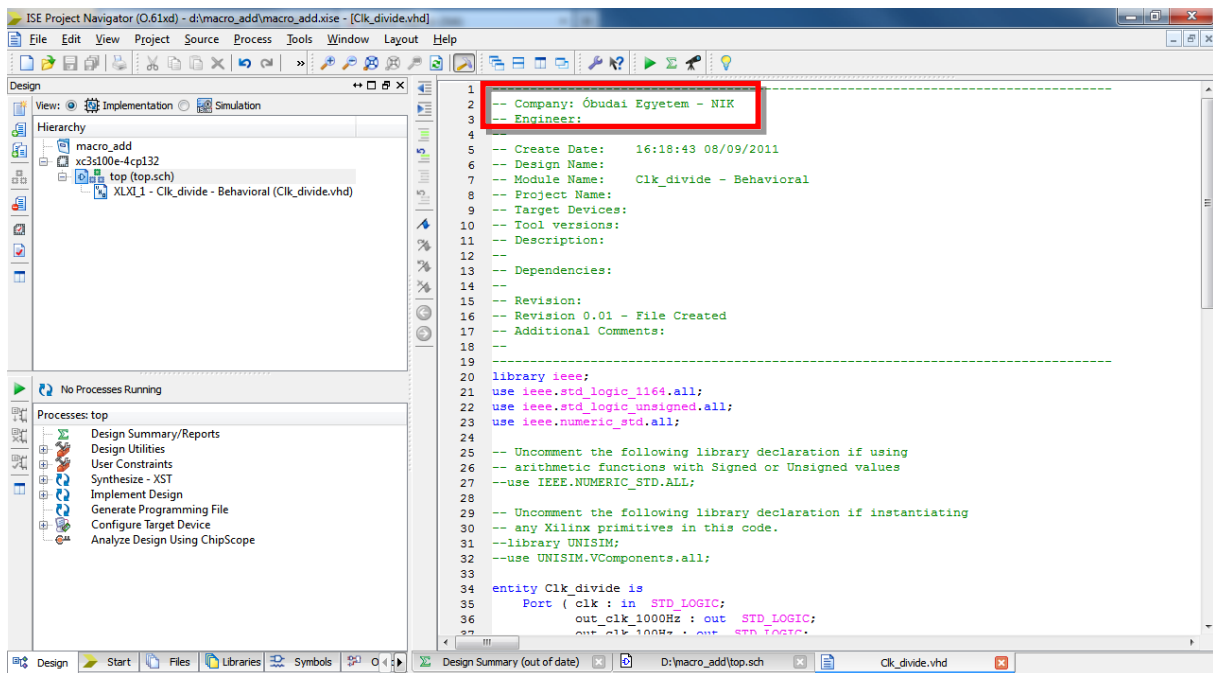


## 1.4.VHDL alapú makró ellenőrzése

Lehetőség van a projekthez hozzáadott makró tartalmának megtekintésére. A makró szimbólum kijelölése után a „View” menü „Push into Symbol” gombjára kattintva megjelenik a működést leíró fájl.

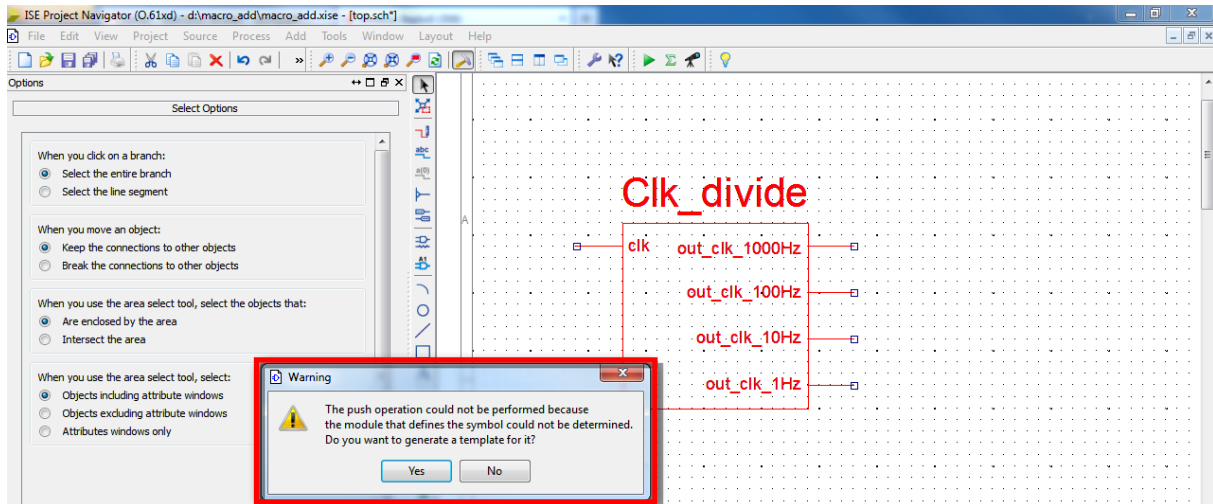


A Clk\_divide makró esetében a következő fájl látható. A fájl 2. sorában a „-- Company: Óbudai Egyetem - NIK” szövegnek kell megjelennie, a labormérésen FTP-ről letöltött makróknál. Amennyiben ez jelenik meg a makró hozzáadása a projekthez sikeres volt.

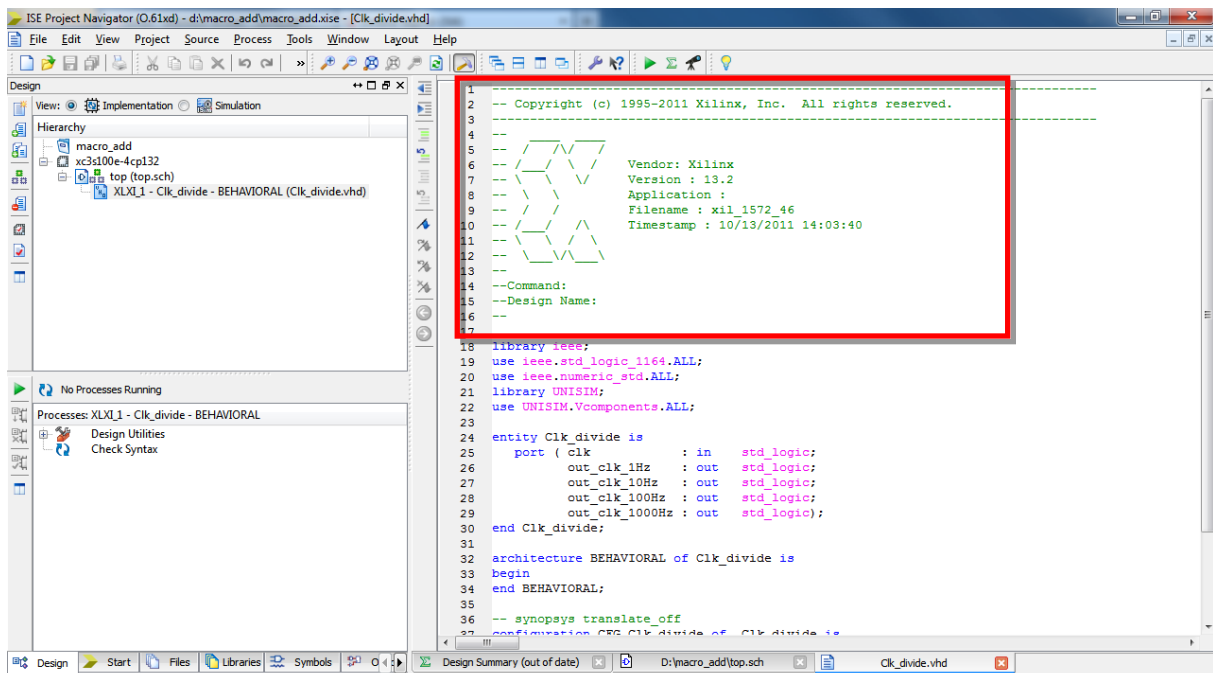


## 1.5.VHDL alapú makró hozzáadásánál keletkezett hiba javítása

Amennyiben a következő képen látható üzenet jelenik meg a „Push into Symbol”-ra kattintva, akkor a \*.vhd fájlt nem, vagy rosszul adtuk hozzá a projekthez. A felugró ablak megkérdezi, hogy készítsen-e a \*.sym fájlhoz egy template (vázlat) vhd fájlt. Itt válasszuk a „No” gombot. Majd az 1.3-as (VHDL alapú makró hozzáadása) fejezetben ismertetett módon adjuk hozzá a \*.vhd fájlt a projekthez újra.

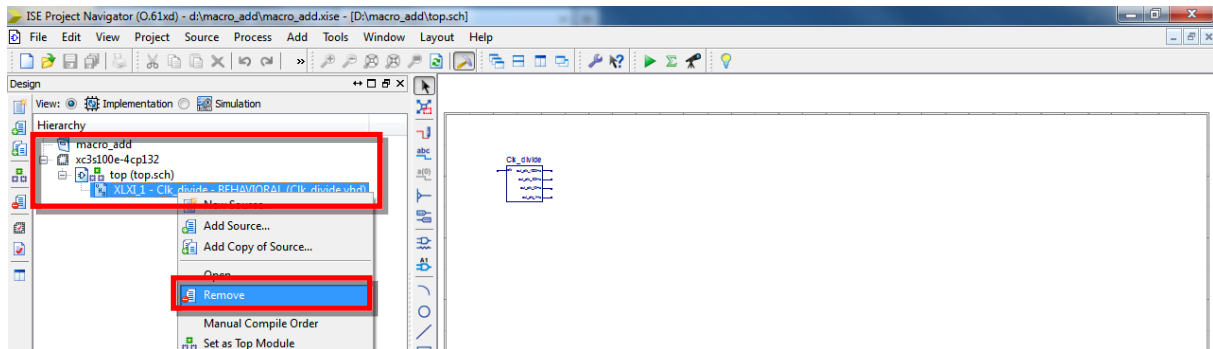


Ha a VHDL szerkesztőben a Vendor név alatt az Óbudai Egyetem helyett a Xilinx felirat jelenik meg, akkor az ISE egy template fájlt rendelt a makróhoz. Ilyenkor a makró működését leíró fájlt nem adtuk hozzá a projekthez.

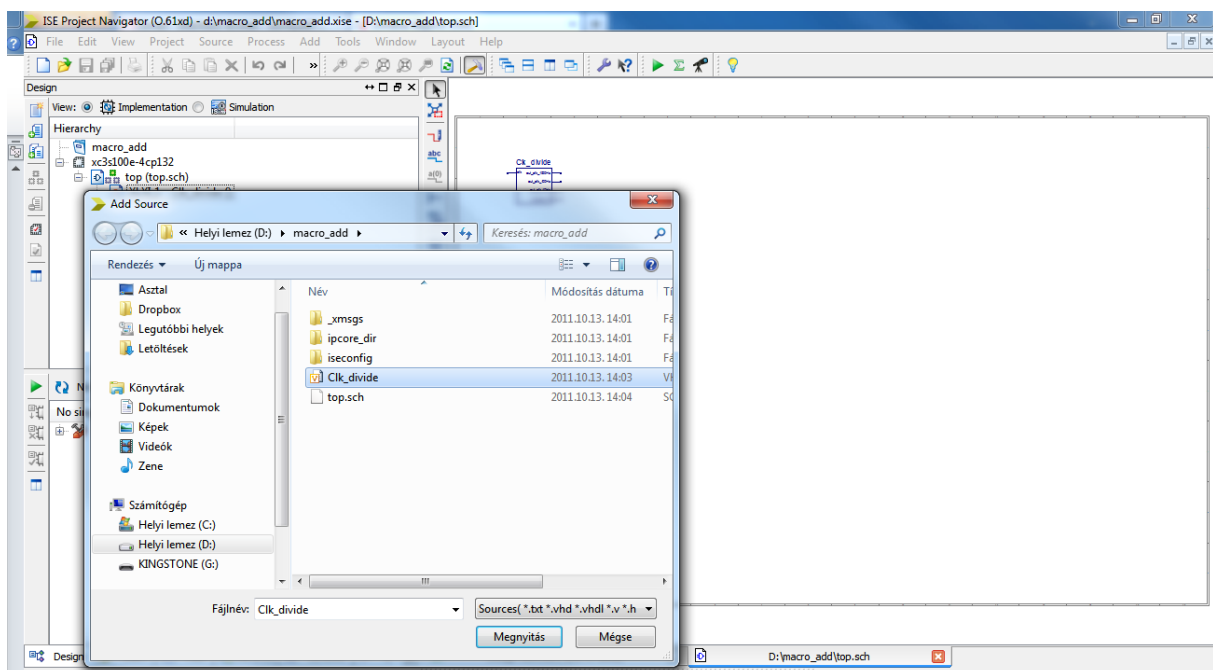
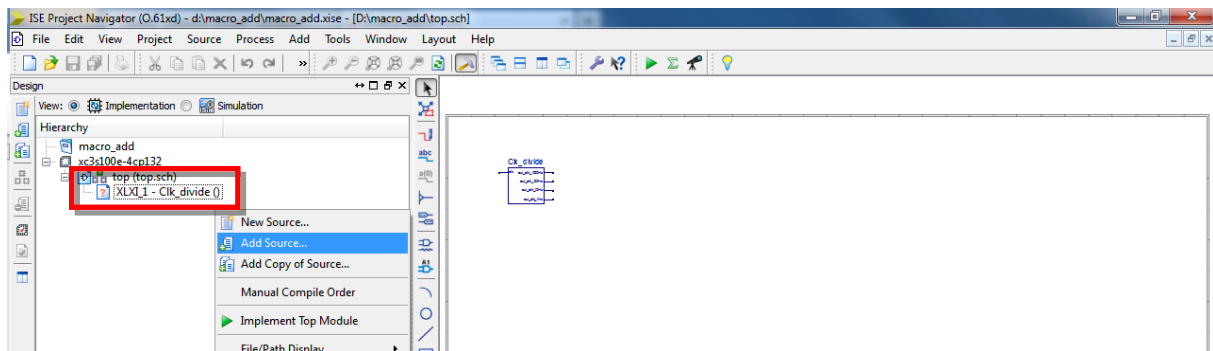




Ennek javítása a következő: először törölni kell a makróhoz tartozó leírófájlt a projektből.



Majd a projektkönyvtárban lévő vhd-fájlt felül kell írni az eredeti fájlal. Lásd 1.3-as (VHDL alapú makró hozzáadása) fejezet. Végül az új vhd fájlt az „Add source” gombbal hozzá kell adni a projekthez. A hozzáadás előtt a sym fájl már létezik, így a projektben megjelenik a makró, de a leírófájl hiányában az ikonján egy kérdőjel látható, vagyis nem találja a fájlt.





## 1.6. Kapcsolási rajz alapú makrók hozzáadása

A kapcsolási rajz alapú makró hozzáadása hasonlóan történik, mint a VHDL alapú makróké. A különbség, hogy a működést leíró fájl \*.sch kiterjesztésű.

Első lépés az sch, és a sym fájl átmásolása a projektkönyvtárba, majd a sch fájlt az „Add Source” paranccsal a projekthez kell adni.

Ebben az esetben is kell egy meglévő projekt, ami célszerűen már tartalmazza a top modult.



## 2. A Digitális Technika II laborgyakorlatok alatt használt makrók leírásai

Az ISE saját szimbólum és makró leírásai az alábbi fájlokban találhatóak meg:

- Kapcsolási rajz alapú makrók:

*Xilinx\13.2\ISE\_DS\ISE\doc\usenglish\isehelp\spartan3e\_scm.pdf*

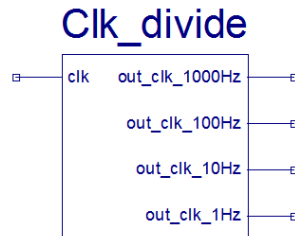
- VHDL alapú makrók:

*Xilinx\13.2\ISE\_DS\ISE\doc\usenglish\isehelp\spartan3e\_vhd.pdf*



## 2.1.Clk\_divide

Makró: Rendszer órajel leosztó áramkör. 1000Hz, 100Hz, 10Hz, 1Hz –es kimenettel.



### Leírás

Rendszer órajelet leosztó áramkör. Az **mclk** 50 MHz-es rendszer órajelet 1KHz, 100Hz, 10Hz és 1Hz-es frekvenciákra osztja le.

### Logikai táblázat

Bemenetek	Kimenetek			
clk	out_clk_1000Hz	out_clk_100Hz	out_clk_10Hz	out_clk_1Hz
50 Mhz	1Khz	100Hz	10Hz	1Hz

A Basys kártya használatakor a clk bemenetre az **mclk** rendszer órajelet kell kötni. Az **mclk** port az ucf-fájlban szerepel, a portra egy 50 Mhz-es oszcillátor kapcsolódik. Ezt a bemeneti portot IBUFG bemeneti bufferrel kell ellátni!



## 2.2. Idozito

Modul: programozható órajel osztó



### Leírás

Rendszer órajelet leosztó áramkör. Az **mclk** 50 MHz-es rendszer órajelet kívülről programozható frekvenciára osztja le. Az „**ido(31:0)**” bemenetre kötött 32-bites konstans (N) határozza meg a kiadott jel frekvenciáját. Az „**ido\_van**” kimeneten a beállított frekvenciának megfelelően periódikusan megjelenik egy rövid ideig (1 órajel ciklus ideig) tartó impulzus. A Demó panelen használt 50 MHz-es órajelnél ennek az impulzusnak a hossza 20 ns.

Az „**rst**” bemenet aktív '1' szintű aszinkron törlés jel.

### Kimenő jel periódus ideje

$T = 20 * (N+1)$  ns (50MHz-es órajel esetén az órajel periódus ideje = 20 ns)

Pld. , ha az „**ido\_van**” kimeneten **T = 1s**-os periodikus jelet (1 Hz-es frekvenciájú) akarunk beállítani, akkor:

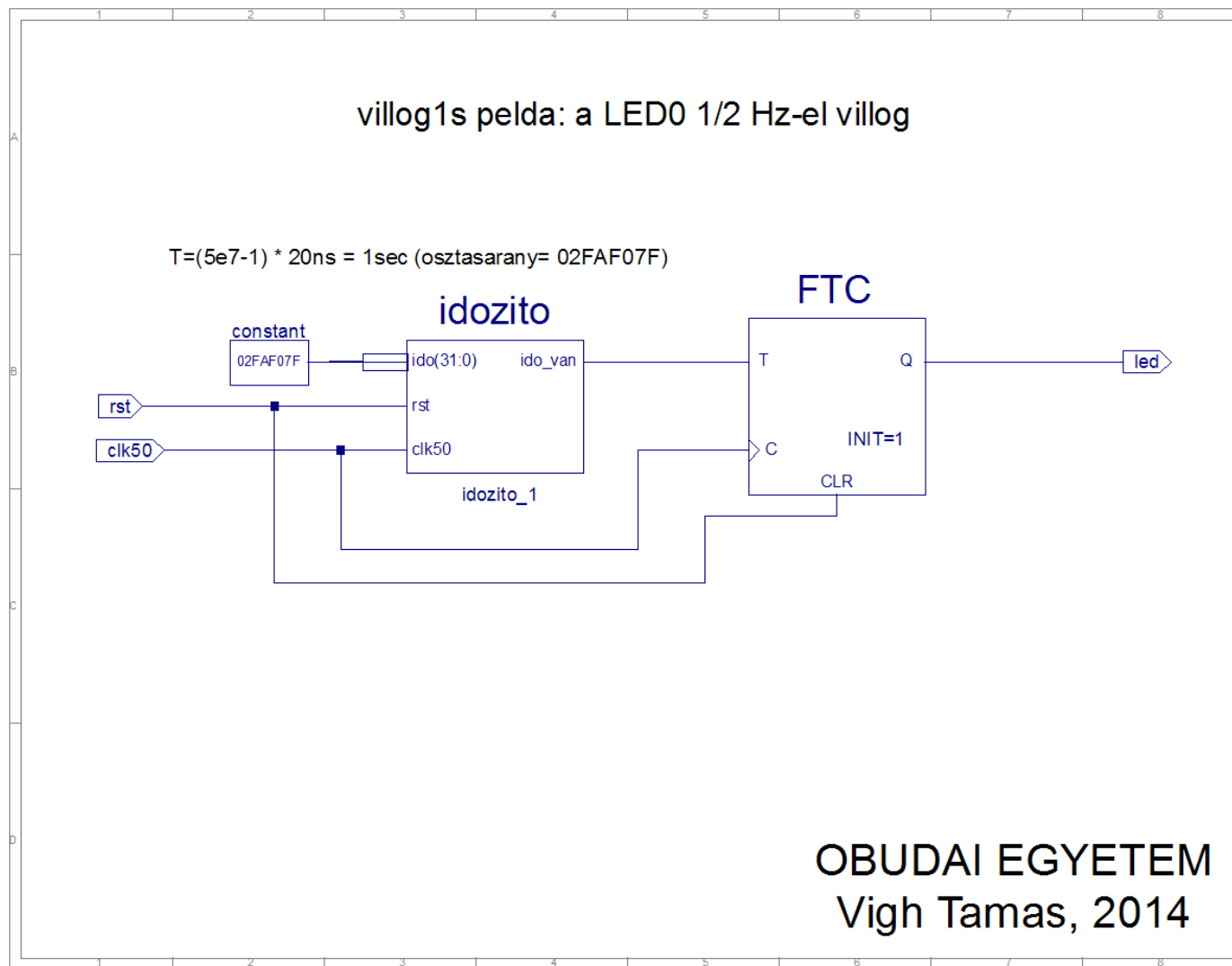
$$N = (T/20ns) - 1 = 5e7 - 1 = (02FAF07F)_H$$

### Logikai táblázat

interfész jel	irány	funkció
clk50	in	MCLK 50 Mhz-es fő órajel
rst	in	aszinkron reset (aktív '1' szintű)
ido(31:0)	in	N = 32 bites hexa osztásarány
ido_van	out	T = 20 * (N+1) ns periódus idő



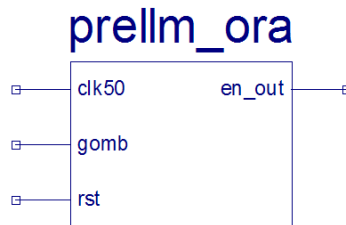
Példa az **Idozito** modul (makró) használatára:





## 2.3.Prellm\_ora

Modul: nyomógomb prellmentesítő és megnyomás (felfutó él) detektor



### Leírás

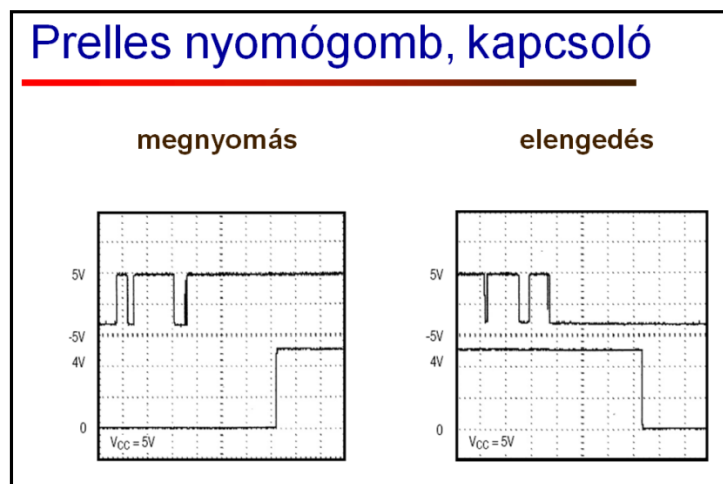
A modul egy nyomógomb (vagy tolókapcsoló) jelét prellmentesíti (zavarmentesíti) és megállapítja a jel felfutó élét. Amikor bekövetkezik a felfutó él a jelen, akkor a modul „en\_out” kimenetén megjelenik egy 1 órajel periódus ideig tartó logikai „1” szintű impulzus.

A modul felhasználására egy példa található a „Lab02\_utmutato.pdf” nevű labor útmutatóban.

### Logikai táblázat

interfész jel	irány	funkció
clk50	in	MCLK 50 Mhz-es fő órajel
rst	in	aszinkron reset (aktív '1' szintű)
gomb	in	nyomógomb (vagy tolókapcsoló), aktív '1'
en_out	out	felfutó él jelzése 20 ns-ig a clk50-nel szinkronizálva

### Prells nyomógomb jele



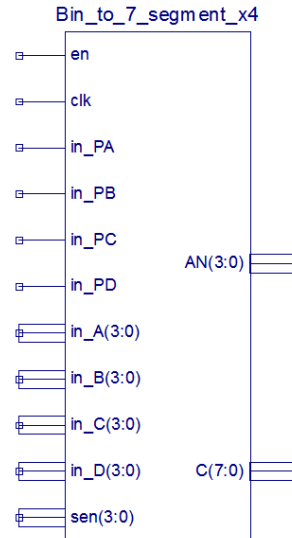
Az ábrákon felül látható a nyomógomb prells jele, alul pedig a prellmentesítés után kapott jel megnyomáskor, ill. elengedéskor.

A prellmentesítő modul a prellmentesítésen kívül még megállapítja a bejövő gombnyomás jel felfutó élét is ('0' – '1' átmenetét).



## 2.4. Bin\_to\_7\_segment\_x4

Makró: Négy idő multiplexált kijelzésű hétszegmenses kijelző meghajtója.



### Leírás

A Basys és Basys 2 FPGA fejlesztőpanelen található négy hétszegmenses kijelző meghajtó blokkja. A kijelzők külön-külön engedélyezhetőek és tilthatóak. Minden kijelző egy négybites adatot kaphat.

### Logikai táblázat

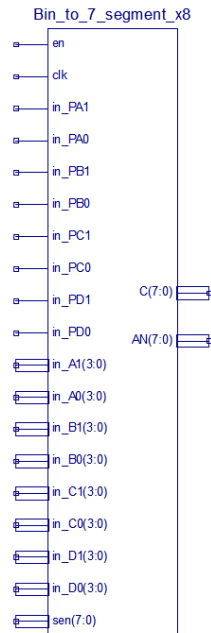
Port	i/o	Busz	Leírás
en	i	-	A modul engedélyező bemenet
clk	i	-	A modul órajel bemenete
in_PA, in_PB, in_PC, in_PD	i	-	Az <b>A...D</b> kijelzők tizedespontjai.
in_A, in_B, in_C, in_D	i	4 bit	Az <b>A...D</b> kijelzők értékei.
sen	i	4 bit	Az <b>A...D</b> kijelzők engedélyező jelei. A sen(0) az <b>A</b> , sen(3) az <b>D</b> kijelző
AN	o	4 bit	Az <b>A...D</b> kijelzők engedélyező jelei. Az AN(0) az <b>A</b> , AN(3) az <b>D</b> kijelző.
C	o	8 bit	Az kijelzők szegmens kimenetei. A C(0) az <b>a</b> , C(6) az <b>g</b> szegmens és C(7) a tizedespont.

Az **en** bemenetre logikai '1' et, a **clk** bemenetre (villódzás mentes megjelenítéshez legalább) 1KHz-es órajelet kell kötni a kijelző működéséhez. Az in\_A ... in\_D bemenetekre a kijelzésre kerülő négybites adatvonalat kell kötni.



## 2.5. Bin\_to\_7\_segment\_x8

Makró: Nyolc idő multiplexált kijelzésű hétszegmenses kijelző meghajtója.



### Leírás

A Basys és Basys 2 FPGA fejlesztőpanelhez készült 8 db hétszegmenses kijelzőt tartalmazó áramkör meghajtó blokkja. A kijelzők külön-külön engedélyezhetőek és tilthatóak. Minden kijelző egy négybites adatot kaphat.

### Logikai táblázat

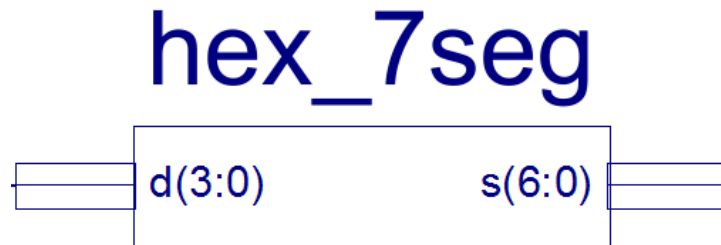
Port	i/o	Busz	Leírás
en	i	-	A modult engedélyező bemenet
clk	i	-	A modul órajelbemenete
in_PA1, in_PA0, in_PB1, in_PB0, in_PC1, in_PC0, in_PD1, in_PD0	i	-	Az <b>A1...D0</b> kijelzők tizedespontjai.
in_A1, in_A0, in_B1, in_B0, in_C1, in_C0, in_D1, in_D0	i	4 bit	Az <b>A1...D0</b> kijelzők értékei.
sen	i	8 bit	Az <b>A1...D0</b> kijelzők engedélyező jelei. A sen(0) az <b>A</b> kijelző.
AN	o	4 bit	Az <b>A1...D0</b> kijelzők engedélyező jelei. Az AN(0) az <b>A</b> kijelző.
C	o	8 bit	Az kijelzők szegmens kimenetei. A C(0) az <b>a</b> , C(6) az <b>g</b> szegmens és C(7) a tizedespont.

Az **en** bemenetre logikai '1' et, a **clk** bemenetre (villódzás mentes megjelenítéshez legalább) 1KHz-es órajelet kell kötni a kijelző működéséhez. Az in\_A1 ... in\_D0 bemenetekre a kijelzésre kerülő négybites adatvonalat kell kötni.



## 2.6.hex\_7seg

Makró: Hétszegmenses kijelző dekóder.



### Leírás

Hétszegmenses kijelző dekóder.

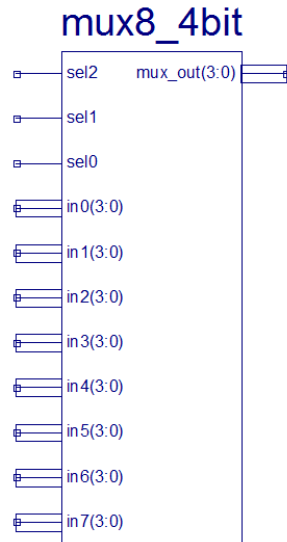
### Logikai táblázat

Port	i/o	Busz	Leírás
d(3:0)	i	4	Dekóder bemenete. (LSB: d(0); MSB: d(3))
s(6:0)	o	7	Szegmens kimenetek: A:s(0)...G:s(6)



## 2.7.mux8\_4bit

Makró: 8-ról 1-re multiplexer, 4 bites adatvonalakkal.



### Leírás

8-ról 1-re multiplexer, 4 bites adatvonalakkal.

### Logikai táblázat

Port	i/o	Busz	Leírás			
sel2, sel1, sel0	i	-	Multiplexer címző bemenete. A sel2 a legnagyobb, a sel0 a legkisebb helyiérték.			
in0, in1 ...in7	i	4 bit	A multiplexer adatbemenetei.			
mux_out	out	4bit	A multiplexer adatkimenete			
			sel2	sel1	sel0	mux_out
			0	0	0	in0
			0	0	1	in1
			0	1	0	in2
			0	1	1	in3
			1	0	0	in4
			1	0	1	in5
			1	1	0	in6
1	1	1	in7			

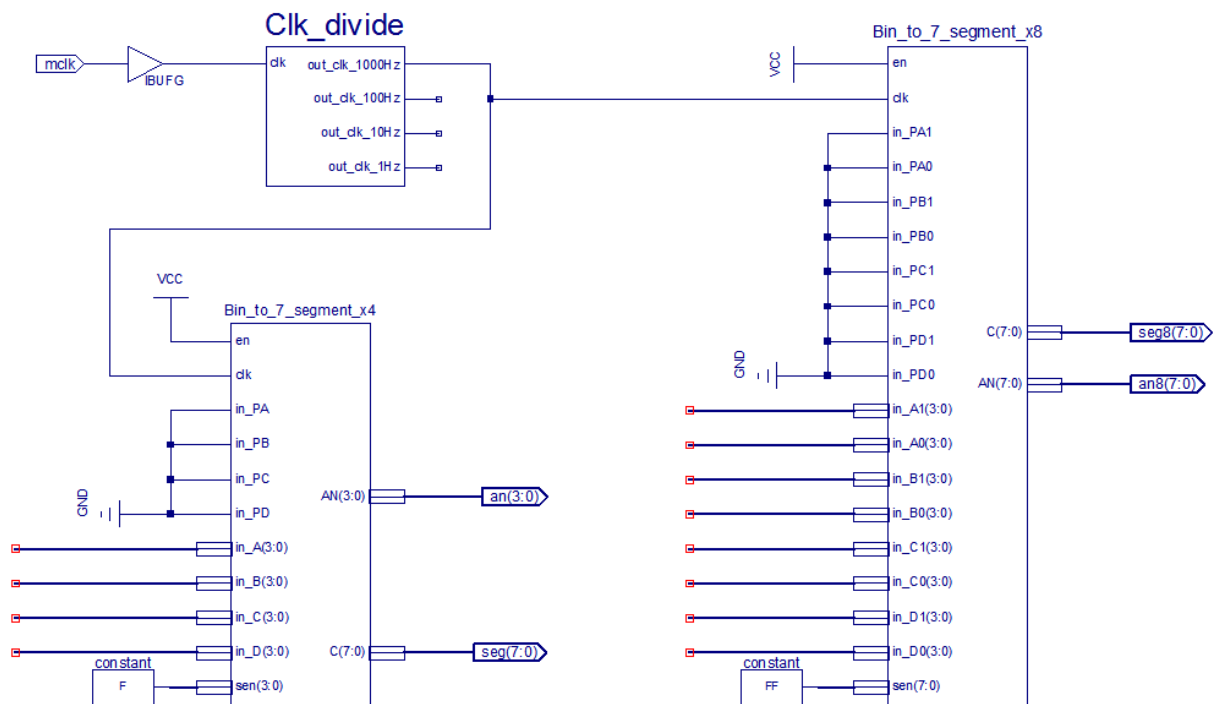


### 3. Makrók használata

#### 3.1. Multiplex vezérlésű hétszegmentes kijelző modulok használata

A kijelzők célszerű bekötése a labormérések feladatainak megoldásához. Az ábrán mind a 12 darab hétszegmentes kijelző engedélyezve van.

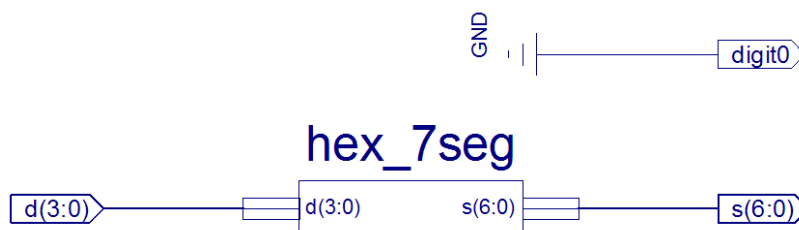
A 12 db **in\_x** bemenetre a kijelzeni kívánt négybites adatvezetéseket kell kötni.





### 3.1.VHDL alapú hétszegmenses dekóder használata.

Egy darab hétszegmenses kijelző használata. A példában a **hex\_7seg** modul (makró) bemenetére kapcsolunk egy négybites hexadecimális számot, amit kijelez az **an0** jelű kijelzőn (ez a jobb szélső) és a tizedespont nem világít.

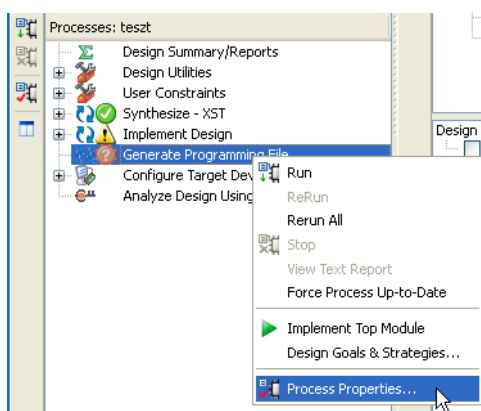


#### Lábkiosztás fájl (.ucf)

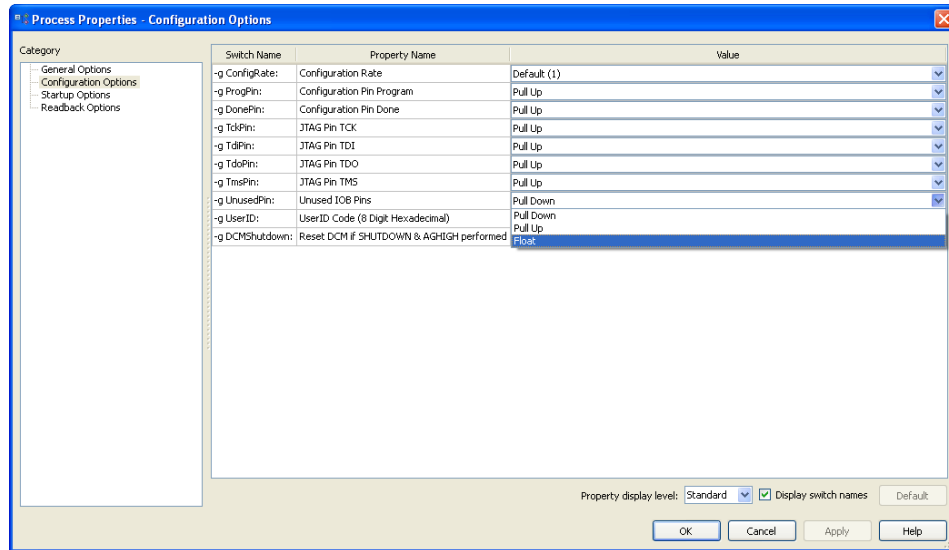
```
2 NET "digit0" LOC = F12; # AN0 , azaz a jobb
3                               # szélső 7-szegm. kij.
4 NET "d[0]" LOC = P11; # SW0
5 NET "d[1]" LOC = L3; # SW1
6 NET "d[2]" LOC = K3; # SW2
7 NET "d[3]" LOC = B4; # SW3
8
9 NET "s[0]" LOC = L14; # A seg
10 NET "s[1]" LOC = H12; # B
11 NET "s[2]" LOC = N14; # C
12 NET "s[3]" LOC = N11; # D
13 NET "s[4]" LOC = P12; # E
14 NET "s[5]" LOC = L13; # F
15 NET "s[6]" LOC = M12; # G
```

#### Konfigurációs paraméterek

**Először beállítunk néhány paramétert:** jobb gombbal kattints a **Processes (műveletek)** ablakban lévő **Generate Programming File** parancson, majd nyisd meg a **Process Properties** ablakot.



A megnyíló **Process Properties** ablakban a **Configuration Options**-ban keresd meg az **Unused IOB Pins** (a nem használt lábak) tulajdonságot és állítsd be **Float** (lebegő) értékűre. Ez a beállítás azt eredményezi, hogy a tervben **nem használt kimeneti perifériák** (ledek, nem használt 7-szegmenses kijelzők stb.) **nem fognak működni**, így nem zavarunk.



A **Startup Options** menüben az **FPGA Start-Up Clock** mezőben válaszd ki a legördülő listából a **JTAG Clock**-ot, amit azért kell beállítani, mert a PC-ről töltjük le a konfigurációt USB-JTAG illesztő interfészen keresztül.