

A laborgyakorlat célja egy System Generator alapú rendszer tervezése és hardver szoftver ko-szimuláció bemutatása. A System Generator alapú környezet alkalmazása egyszerűsíti a hardver tervezést és tesztelést. Míg a VHDL alapú tervezés alapos hardveres ismerteket igényel, a HLS technikák és egyben a System Generator alapú környezet alkalmazásával minimális hardver ismeretek mellett is könnyedén megvalósítható egy alkalmazás.

A System Generator környezetben speciális Simulink tömbök alkalmazásával tervezhető meg egy áramkör. Az eszköz alkalmazása könnyíti a fixpontos és lebegőpontos aritmetika alkalmazását, nagymértékben egyszerűsítve algoritmusok megvalósítását. Az eszköz lehetővé teszi Matlab programkód, C, C++ vagy akár HDL nyelven (VHDL, Verilog) megvalósított moduloknak a tervbe való integrálását.

A megvalósított terv első fázisban szimuláció során a Simulink környezetben, majd hardver ko-szimulációs üzemmódban tesztelhető.





A terv elkészítése előtt egy pár fontosabb beállítás és megjegyzés

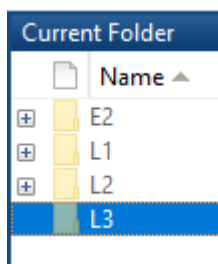
- Ahhoz, hogy a környezet felismerje majd az alkalmazandó fejlesztőlapot, a Matlab startup.m inicializáló állományába (%C:\Users\felhasznalónev\Documents\MATLAB\startup.m) be kell másolni a következő két utasítást és ellenőrizni kell, hogy helyes-e a két elérési útvonal.

```
addpath([getenv('XILINX_VIVADO')] '/scripts/sysgen/matlab');
```

```
xilinx.environment.setBoardFileRepos({'C:/Xilinx/Vivado/2016.2/data/boards'});
```

- A környezet indításához a munkaasztalon kell a System Generator 2016 linkre kattintani, amely majd indítani fogja a Matlab környezetet. Egyes változatokban a Simulink automatikusan elindul a System Generator indításával, a 2015a és 2015b verziók esetében a Simulinket kézzel kell indítani.
- A Simulink indítását követően létre kell hozni a D:\diak\felhasználónév\L3 könyvtárat majd létre kell hozni egy új simulink modellt (New Modell). A modellt azonnal le kell menteni a

megjelölt könyvtárba, például pwm névvel     .



- Az elérési útvonal nem tartalmazhat szünetet.

Bevezető

A létrehozott és lementett Simulink modellbe be kell illeszteni az áramkört megvalósító alegységeket valamint a konfigurációs eszközöket. Minden egyes alegységnek (alrendszernek) tartalmaznia kell egy System Generator konfigurációs eszközt. A System Generator eszköz segítségével lehet meghatározni az alkalmazott FPGA vagy SoC áramkör típusát, működési frekvenciáját, azt, hogy az alrendszernek melyik jelen szolgáltatjuk majd az órajelet.

A hardver ko-szimuláció (vegyes szimuláció) lényege, hogy egy algoritmust több célhardveren futtatunk. Az algoritmus egyik része a gazda számítógépen a Simulink környezetben fog végrehajtódni, míg egy másik része az FPGA áramkörben, pontosabban a Gateway In és Gateway Out közötti részek az FPGA áramkörben, míg a Gateway In előtti részek és a Gateway Out utáni részek a Simulink környezetben. Az alább szemléltetett példában a VHDL hardver leíró nyelven tervezett PWM generátort teszteljük hardver ko-szimulációs üzemmódban. Jelen feladat esetében a Simulink környezetet a bemeneti adatok meghatározására, valamint az eredmény megjelenítésére alkalmazzuk.

A Matlab és System Generator környezetben a következő típusokat alkalmazhatjuk a különböző jelek, adatok ábrázolására:

- Boole típusú változó (Boolean)
- Fixpontos
 - Előjel nélküli fixpontos aritmetika
UFix_5_3, 5 biten ábrázolt előjel nélküli szám, 2 bit az egész részre és 3 bit a törtrészre
 - Előjeles fixpontos aritmetika
Fix_16_8, kettes komplement, 16 biten van ábrázolva a szám, 8 bit a törtrész

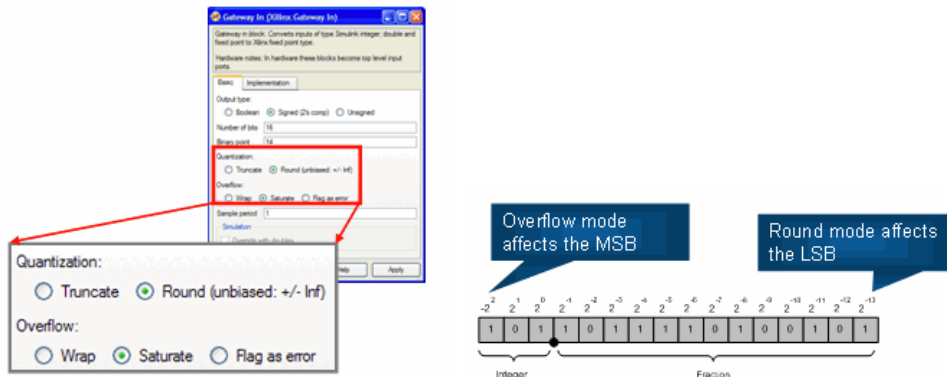
Pontosság

full precision: a modul bemenetének megfelelő pontosságot *alkalmazzuk*

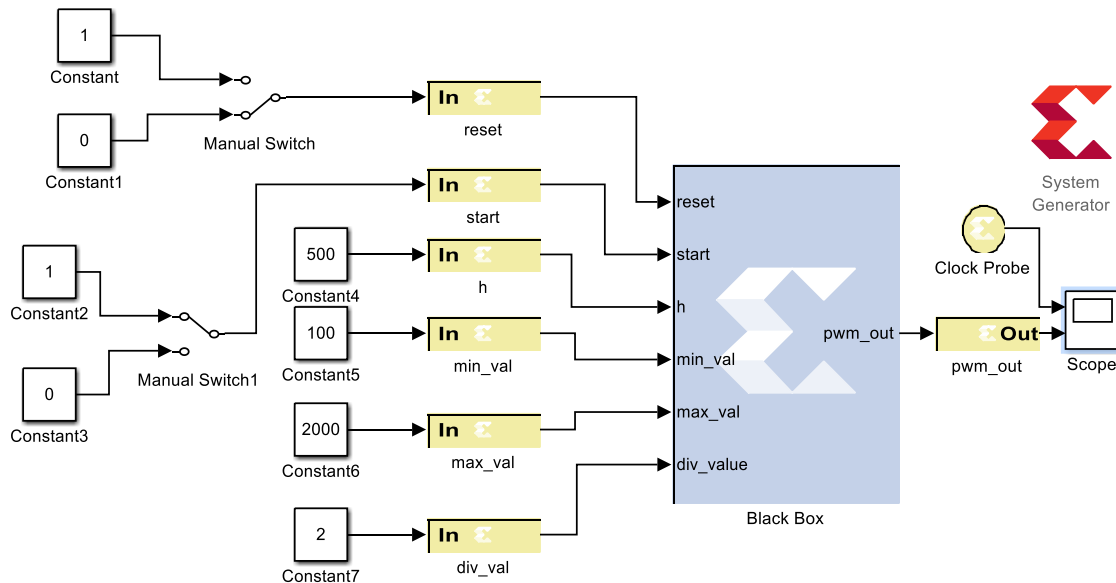
user-specified precision: a felhasználó által meghatározott pontosságot alkalmazzuk. A felhasználó által megválasztott pontosság általában kisebb, mint például a Simulink környezetből átvett változó pontossága, emiatt az alacsony helyértékű és/vagy a magas helyértékű bitekről is le kell mondani. Az alacsony helyértékű bitek esetében kvantálásról, míg a magas helyértékű bitek esetében túlsordulásról beszélünk.

- *Kvantálás: **Truncate, Round***
- *Túlsordulás: **Wrap, Saturate, Flag as error***

A kvantálás megoldható az alacsony helyértékű bitek levágásával (Truncate) vagy kerekítésével. Túlsordulás esetében a (Wrap) opció alkalmazásával egyszerűen elhagyjuk a magasabb helyértékű biteket, míg a Saturate opcióval szaturáljuk a formátumnak megfelelő legnagyobb vagy legkisebb értékre a jelet.



A Matlab környezetben az adatok *double* (dupla pontosságú lebegőpontos) típusként vannak kódolva, míg az FPGA áramkörben általában egész számként, fixpontos vagy lebegőpontos formátumban. Általában a Matlabból (Simulinkből) az adatoknak az FPGA áramkörre való áramlása során veszítünk a pontosságból. Az átalakítást illetve a visszaalakítást a Gateway In illetve a Gateway Out modulok végzik el.



A terv megvalósításának és konfigurálásának lépései

Fontosabb Xilinx alapú Simulink tömbök

System Generator- az alkalmazott FPGA áramkör paraméterezése, működési frekvencia meghatározása

Gateway In- bemeneti jelek egy egységben. A Gateway In modulok a bemenetek mintavételezését végzik, és a bemeneti adatokat átalakítják a modulban konfigurált típusnak megfelelően, valamint a következő mintavételig megtartják az utolsó mintavételi értéket.

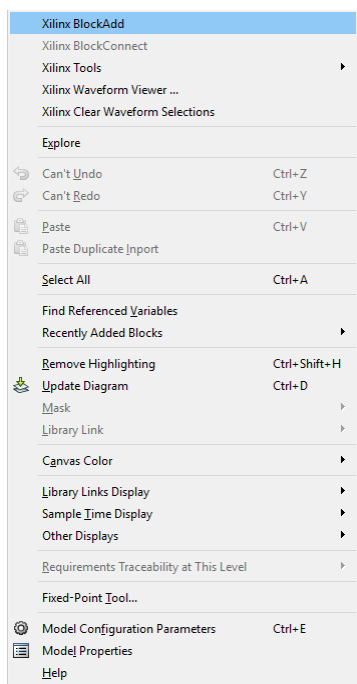
Gateway Out- tömbök az áramkör kimeneteit jelentik, hardver ko-szimuláció során visszaalakítják a Xilinx típusú adatokat a MATLAB-ban alkalmazott double típussá.

BlackBox- VHDL illetve Verilog alapú moduloknak a tervhez való csatolása. A Black Box alkalmazása során létrejön egy konfigurációs .m állomány, amelyben a modul működésével kapcsolatosan több paraméter is beállítható. Többek között meghatározható (lecserélhető) a kimeneti adatok típusa.

Regiszter- a tervbe regisztereknek való integrálását teszi lehetővé. A regiszter esetében nem kell meghatározni a bitek számát, ugyanis ezt automatikusan örökli a bemeneti jel sínszélessége függvényében. A Xilinx tömbök nagy részének esetében alkalmazhatók (validálható) a reset (rst- Provide Synchronous reset port) illetve az órajelet engedélyező (EN-Provide Enable port) bemeneti jelek.

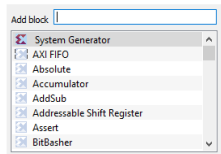
Bitbasher- sínek kezelése, jelek összevonása, illetve jelek sínről való leválasztása

Az alapkönyvtárban megtalálható Xilinx tömbök a *Xilinx BlockAdd* gyorsparancssal (jobb kattintás a modellre, majd kattintás a *Xilinx BlockAdd* almenüpontra) vagy Simulink könyvtár keresőben (*Simulink Library Search*) megtalálható *Xilinx Blockset* illetve *Xilinx Reference Blockset* eszköztárakból érhetők el.

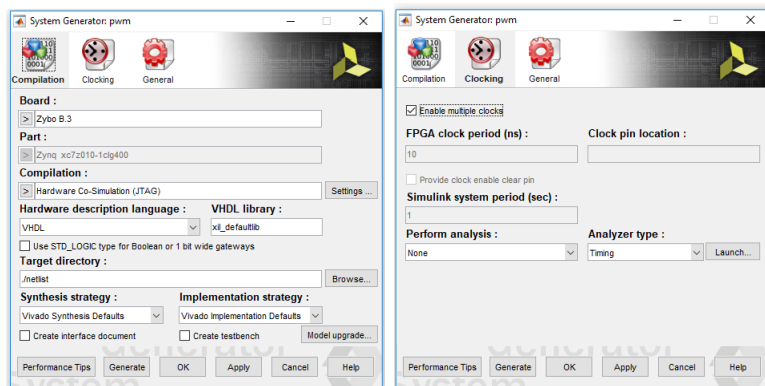


ábra 1 gyorsparancsként elérhető menüpontok.

A Xilinx BlockAdd menüponttal Xilinx eszközöknek/elemeknek a modellhez való csatolása válik lehetővé. A listából ki kell választani a modult, amelyet a tervbe szeretnénk illeszteni



System Generator- az FPGA áramkör típusának meghatározása

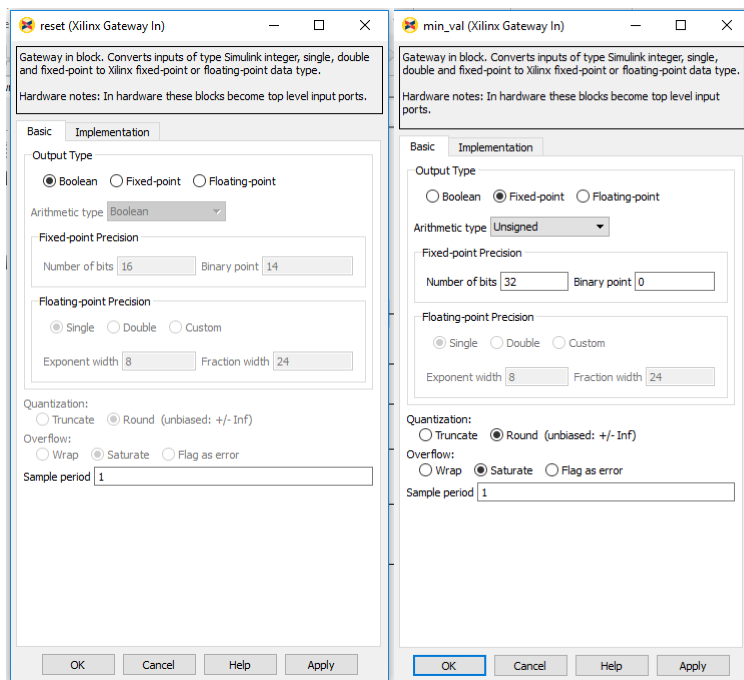


ábra 2 A System Generator eszköztárban elérhető konfigurációs beállítások

A System Generator tömb keretében a következő három konfigurációs ablak érhető el:

Compilation: az alkalmazott fejlesztő lap (vagy FPGA) típusának kiválasztása, az alkalmazás típusa, az alkalmazott hardver leíró nyelv (VHDL) vagy Verilog, alkalmazott VHDL könyvtár, szintézisre kiválasztott stratégia (Vivado Synthesis Defaults), implementációra kiválasztott stratégia (Vivado Implementation Defaults), automatikus dokumentáció generálás (Create interface document), automatikus tesztpad (Create testbench)

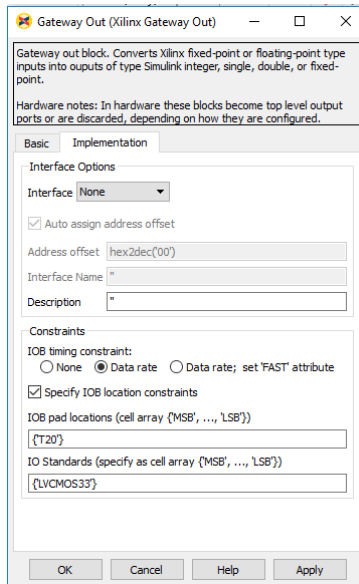
Clocking: Enable multiple clock- több órajeles rendszer esetében, az FPGA áramkör működési frekvenciájának megfelelő órajel periódusa (FPGA clock period), a modul órajelének meghatározása (Clock pin location), Simulinkben az FPGA órajelnek megfelelő periódus



ábra 3A reset bemenet esetében a Gateway In modul konfigurálása

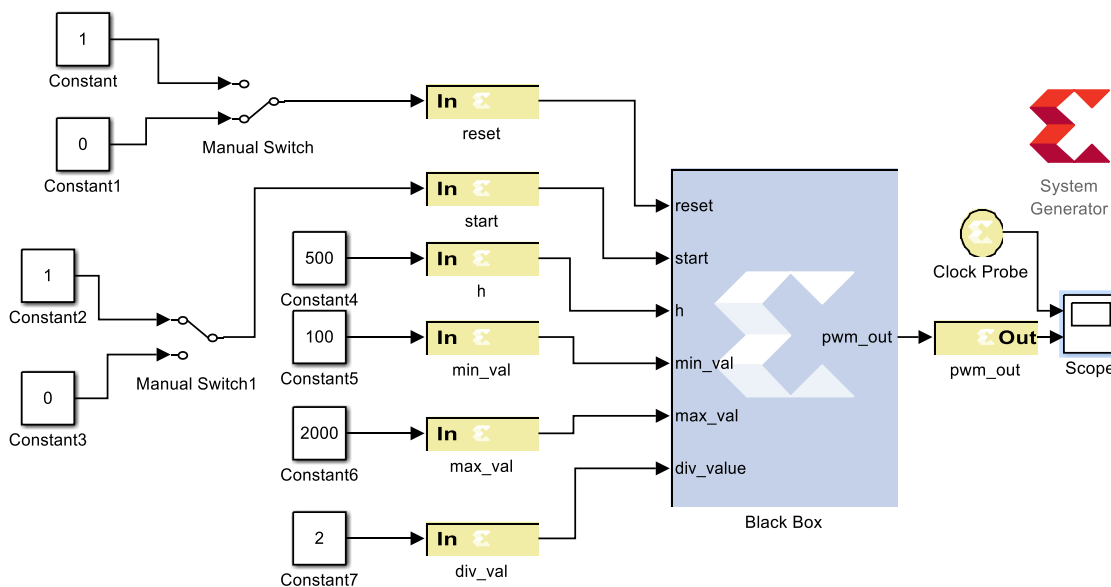
Pmod JA (XADC)	Pmod JB (Hi-Speed)	Pmod JC (Hi-Speed)	Pmod JD (Hi-Speed)	Pmod JE (Std.)	Pmod JF (MIO)
JA1: N15	JB1: T20	JC1: V15	JD1: T14	JE1: V12	JF1: MIO-13
JA2: L14	JB2: U20	JC2: W15	JD2: T15	JE2: W16	JF2: MIO-10
JA3: K16	JB3: V20	JC3: T11	JD3: P14	JE3: J15	JF3: MIO-11
JA4: K14	JB4: W20	JC4: T10	JD4: R14	JE4: H15	JF4: MIO-12
JA7: N16	JB7: Y18	JC7: W14	JD7: U14	JE7: V13	JF7: MIO-0
JA8: L15	JB8: Y19	JC8: Y14	JD8: U15	JE8: U17	JF8: MIO-9
JA9: J16	JB9: W18	JC9: T12	JD9: V17	JE9: T17	JF9: MIO-14
JA10: J14	JB10: W19	JC10: U12	JD10: V18	JE10: Y17	JF10: MIO-15

ábra 4 A Zybo fejlesztőlapon a Pmod típusú kimeneteknek megfelelő FPGA kivezetések



ábra 5 A `pwm_out` kimenet esetében meghatároztuk, hogy a jelet nem a Simulink modulba szeretnénk visszavezetni, hanem a T20-as jelölésű FPGA lábon kivezetni.

A terv megvalósításához az alábbi rajznak megfelelő modellt próbáljuk megrajzolni. A PWM vezérlő VHDL forráskódja (`pwm_ultra.vhd`) elérhető a hálózaton az L3 könyvtárban. A VHDL forrásállományt másoljuk a `D:\diak\felhasznalonev\UKDA\L3\VHDL` könyvtárba.



ábra 6 PWM jelgenerátor System Generator alapú megvalósítása. A VHDL-ben megvalósított PWM egységet BlackBox modulként kell a tervbe illeszteni.

Fontosabb beállítások:

Reset: Boolean típusú bemenet

Star: Boolean típusú bemenet

H: vezérlő jel, jelen esetben 32 bites előjel nélküli szám (UFix_32_0)

Min_val: meghatározza a minimális kitöltést, jelen esetben 32 bites előjel nélküli szám (UFix_32_0)

Max_val: meghatározza a jel periódusát, jelen esetben div_val32 bites előjel nélküli szám (UFix_32_0)

Div_val: előosztó, skálázni lehet a PWM jel periódusát, tíz bites előjel nélküli szám (UFix_10_0)

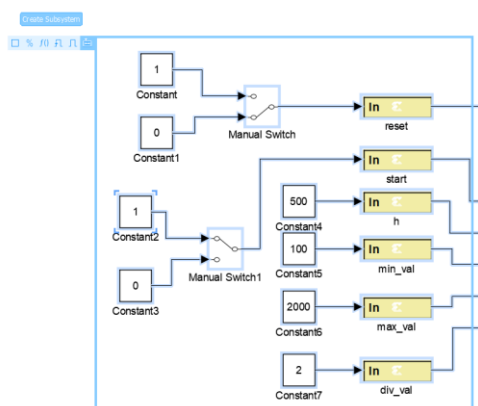
A Simulinkben a beállított bemenetek függvényében lefuttatva a szimulációt, ki lehet értékelni, hogy a jelforma megfelel-e a valóságnak.

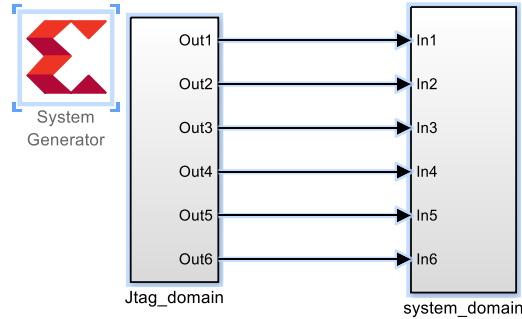
A hardver ko-szimulációs üzemmódban, ahhoz hogy az FPGA részben a PWM modul, az FPGA áramkör órajelén működjön, a tervet át kell alakítani több órajel alapú tervvé, mivel a Gateway In és Gateway Out modulok a Jtag interface órajelén működnek, a PWM modult pedig az FPGA áramkör órajelén szeretnénk működtetni. Korábbi ISE alapú System Generator környezetekben hardver ko-szimuláció során lehetőség volt több üzemmód közül választani. Free running üzemmódban a rendszer az órajelet az FPGA fő órajeltől kapta, Single clock üzemmódban pedig a Jtag interfésztől.

Jelen esetben a Zybo fejlesztőlaphra, ha nem alakítunk ki egy több órajel által vezérelt rendszert, vagyis egy Multiple Clock típusú rendszert, csak a Jtag egység órajelén van lehetőség futtatni az FPGA-ban integrált egységeket (legalábbis az egyetemen csak ebben a változatban sikerült a Zybo lapon létrehozni a valósidejű hardver ko-szimulációt).

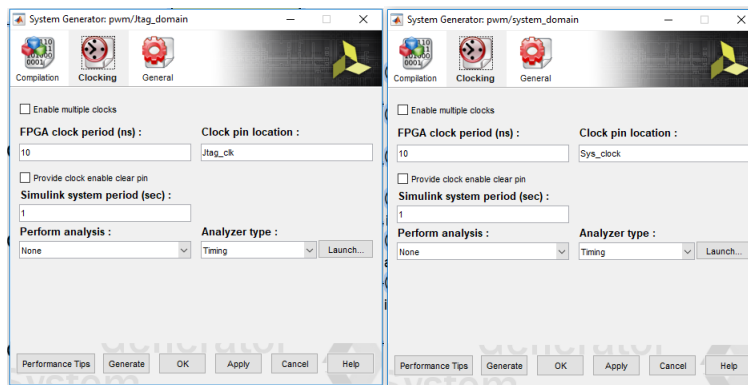
A Multiple Clock típusú rendszer kialakítása egyszerűen megvalósítható: egy részét a tervnek be kell vinni egy alrendszerbe (Simulinkhez kapcsolódó egységek, Gateway In, modulok) és a Xilinx tömböket (vagyis a terv azon részét, amely az FPGA áramkörön kell lefusson) egy másik alegységbe (subsystem típusú Simulink elem).

Kijelölve az alrendszerekhez tartozó elemeket, majd a Create Subsystem utasításra kattintva automatikusan az elemek bekerülnek egy alrendszerbe.

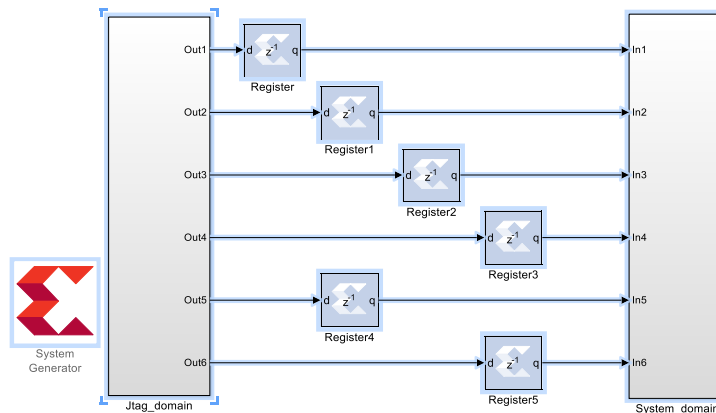




A Jtag_domain-nek megfelelő alegység az órajelet a Jtag_clk jel, míg a system_domain alegységnek az órajelet a sys_clock jel fogja szolgáltatni. A felső szintű System Generator konfigurációs eszközben engedélyezni kell a Multiple Clock típusú rendszert. A System Generator tömböt minden egyes alrendszerbe (jelen esetben Jtag_domain, illetve system_domain), az alrendszerek Single Clock típusú rendszerként működnek.

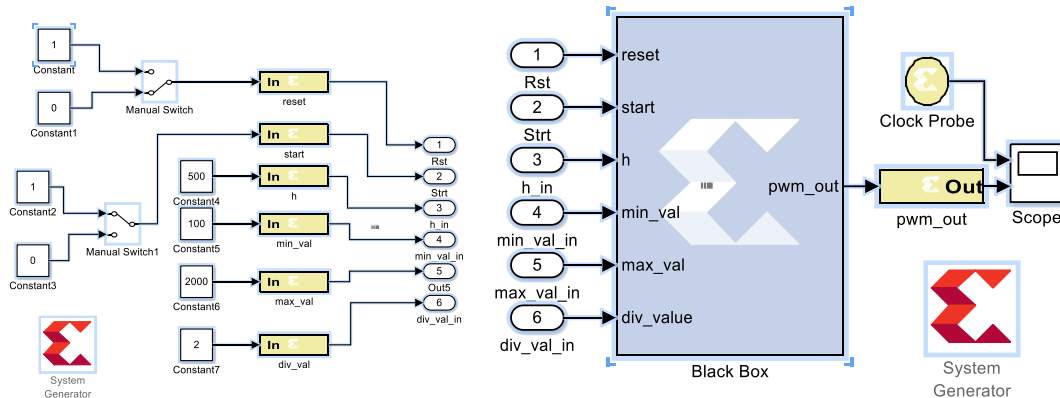


ábra 7+Órajelek konfigurálása a két alrendszerrendszerre

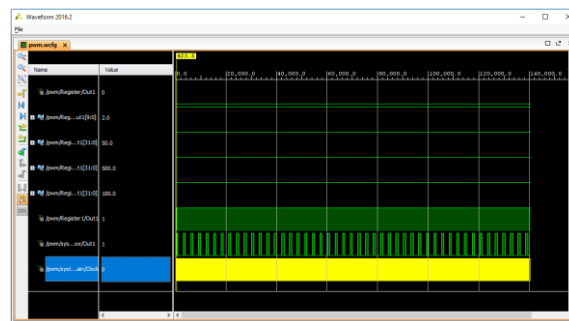


ábra 8 A két alrendszer közé illesztett regiszterek

A két modul közé, mivel a két modul különböző frekvencián működik, regisztereket kell illeszteni mindkét irányba. Jelen esetben csak a Jtag_domain-ből kapcsolódnak a jelek a System_domain irányába.



ábra 9 A Jtag_domain illetve system_domain alrendszereknek megfelelő elemek. Bal oldali ábra Jtag_domain, job oldali ábra System_domain



A Simulink Scope-okon kívül lehetőség van a jelformák megjelenítésére a Xilinx Waveform Viewer eszköznek az alkalmazásával is. Egyszerűen ki kell jelölni a jeleket, amelyeket meg szeretnénk jeleníteni, majd ki kell választani a gyorsmenüből a Xilinx Waveform Viewer parancsot. Xilinx Clear Waveform Selection parancssal pedig visszavonhatjuk a megjelenítést. A jelformáknak ebben a változatban való megjelenítése több lehetőséget nyújt a rendszer tesztelése szempontjából, ugyanis többek között lehetővé teszi a lefutó/felfutó élek keresését.

C:\Munka\Tiha\UKDA\L3

C:\Munka\Tiha\UKDA\L3\netlist\hwcosim\pwm.srscs\constrs_1\imports\sysgen

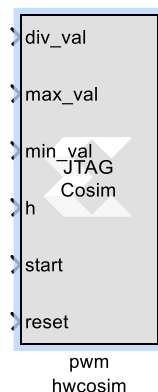
A Multiple Clock típusú rendszer kialakítását követően létre kell hozni a hardver ko-szimuláció elvégzéséhez szükséges Simulink tömböt, a System Generator modulból a *Generate* gombra kattintva.

Minden egyes fordítást követően a System_domain egységnek meg kell határozni, hogy honnan származtatja a Sys_clock órajelet. Ehhez a módosításhoz a fordítás kezdetét követően meg kell keresni a **netlist\hwcosim\pwm_a.srscs\constrs_1\imports\sysgen** könyvtárban található megkötés állományt, valamint a tartalmát az alábbi minta szerint átszerkeszteni.

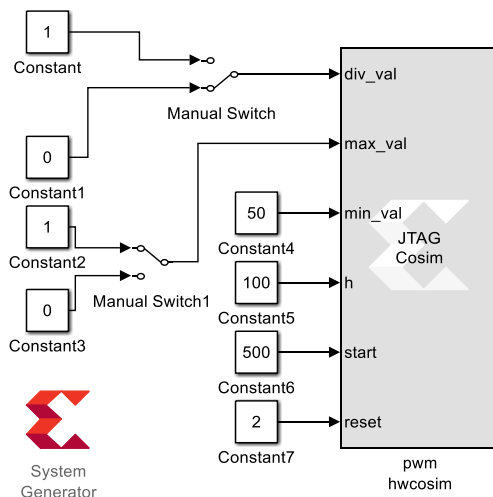
```
set_property -dict { PACKAGE_PIN L16 IOSTANDARD LVCMOS33 } [get_ports { sys_clock }];
```

```
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports { sys_clock }];
```

Ha a terv sikeresen lefordul, létrejön a ko-szimulációhoz szükséges könyvtárelem.



Kimásolva a könyvtárelemet, egy újabb Simulink modellben ki kell alakítani a ko-szimulációs modellt. Ez egyszerűen úgy is megoldható, hogy az eredeti modellből kivágjuk a Xilinx elemeket, és helyettük behelyezzük a létrehozott könyvtárelemet. Ebben az esetben a System Generator modulban a tervet egy órajeles rendszerre kell állítani. A létrehozott könyvtárelem háttérében megtalálható az FPGA áramkör konfigurálásához szükséges bitfolyam, valamint a System Generator beállítása alapján azonosítható az alkalmazandó fejlesztő rendszer.



Összekapcsolva a számítógéppel és lefuttatva a szimulációt, első lépésben felprogramozódik az FPGA áramkör, majd azt követően elindul a szimuláció. Ahhoz, hogy a megvalósított automata működésbe lépjen, a *reset* jelet logikai nullára illetve a *start* jelet logikai egyesre kell állítani. A szimuláció futási ideje alapértelmezetten 10 s, ami az FPGA áramkörben 10 órajelnek felel meg. A szimuláció időtartamát javasolt átállítani *inf* értékre.

A gyakorlati feladat elvégzésnek fontosabb lépései

- Valósítsuk meg a leírások alapján a felvázolt tervet
- Szimulációval ellenőrizzük, hogy a rendszer működése megfelel-e az elvárásoknak
- Alakítsuk ki a Multiple Clock típusú rendszert

- Generáljuk ki a ko-szimulációhoz szükséges könyvtári elemet, figyelve, hogy minden újrafordítás során helyesen beállítottuk a sys_clock órajel származtatását.
- Oszilloszkópot alkalmazva, különböző bemeneti paraméterekre tanulmányozzuk a generált jelformát.
- Hasonlítsuk össze a valós mérések alapján, valamint a szimuláció során elért eredményeket.