



BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
VILLAMOSMÉRNÖKI ÉS INFORMATIKAI KAR
MÉRÉSTECHNIKA ÉS INFORMÁCIÓS RENDSZEREK TANSZÉK

Mikrorendszerek tervezése

Szoftverfejlesztés

Fehér Béla
Raikovich Tamás



Asztali vs. beágyazott SW fejlesztés

Asztali rendszerek

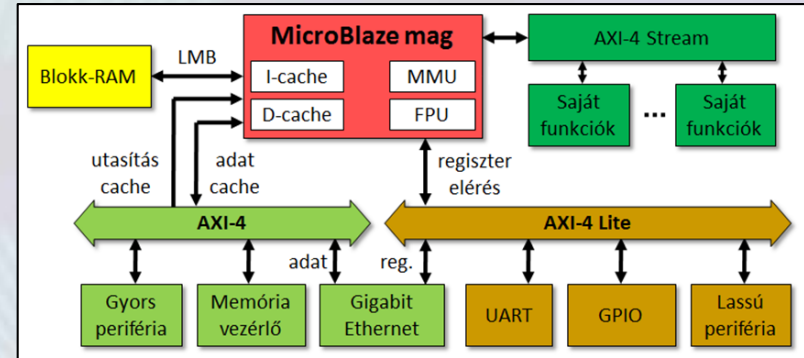
- Fejlesztés, hibakeresés és futtatás ugyanazon a gépen
- Az OS akkor tölti be a programot a memóriába, ha a felhasználó ezt kéri
- **Címek feloldása**
 - Az alkalmazás betöltésekor
 - A betöltő az OS része

Beágyazott rendszerek

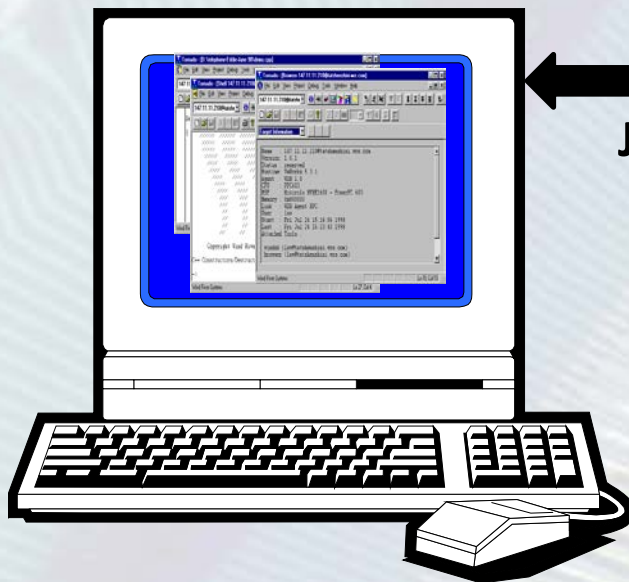
- Fejlesztés: a host gépen
- Futtatás: a célrendszeren
- **Egy futtatható állomány**
 - ELF fájl
 - Bootloader, alkalmazás, ISR, operációs rendszer
 - Címek feloldása linkeléskor
- **Futtatható kód letöltése a célrendszerre**
 - JTAG, Ethernet, soros port
 - Flash programozó

Asztali vs. beágyazott SW fejlesztés

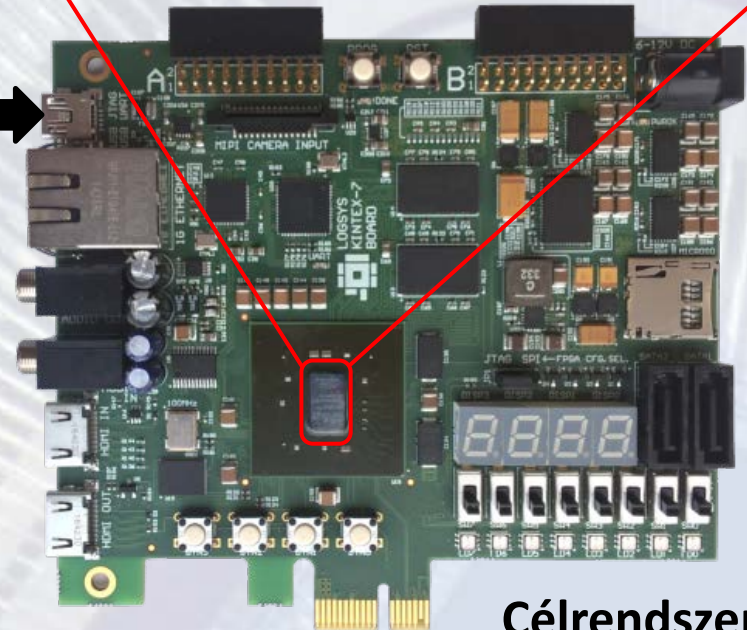
- A fejlesztés külön számítógépen (host) történik, az alkalmazást le kell tölteni a célrendszerre
- A keresztfordító a host gépen fut



Host számítógép



JTAG, UART



Célrendszer

Asztali vs. Beágyazott SW fejlesztés

Különféle problémák

- Minden terv esetén egyedi a hardver
- Megbízhatóság
- Valós idejű válasz megkövetelése
 - RTOS \leftrightarrow normál OS
- Kis méretű, kompakt kód
- Magasszintű nyelvek (C/C++) \leftrightarrow assembly

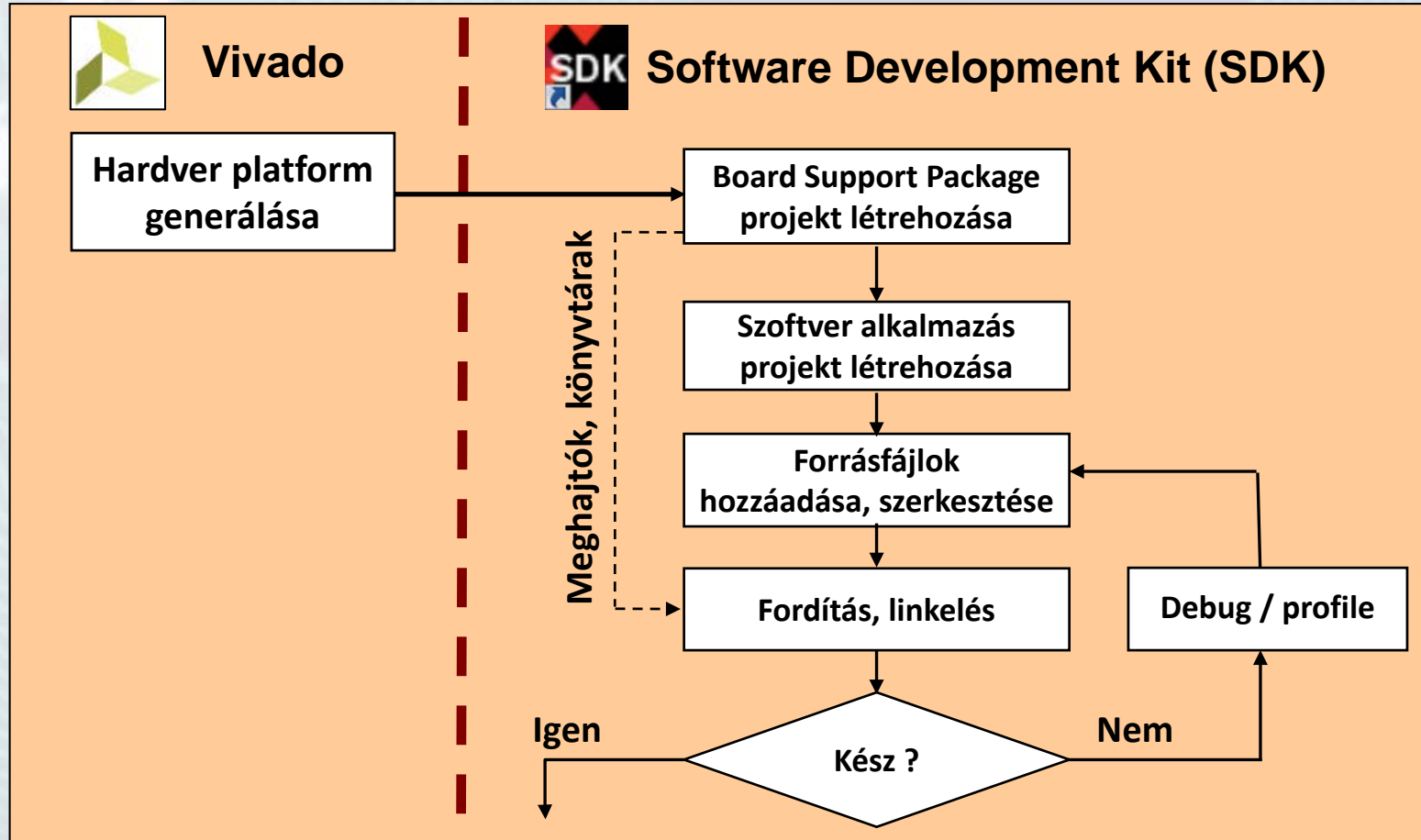
Software Development Kit (SDK)

Xilinx Software Development Kit (SDK)

- Java/Eclipse alapú alkalmazás fejlesztői környezet
- Szolgáltatásokban gazdag C/C++ kód szerkesztő és fordító környezet
- Projekt menedzsment
- Automatikus konfigurációs fájl és makefile generálás
- Hiba navigálás
- Jól használható környezet a beágyazott rendszerek hibakereséséhez (debug)
- Verziókövetés

Software Development Kit (SDK)

A fejlesztés folyamata



Software Development Kit (SDK)

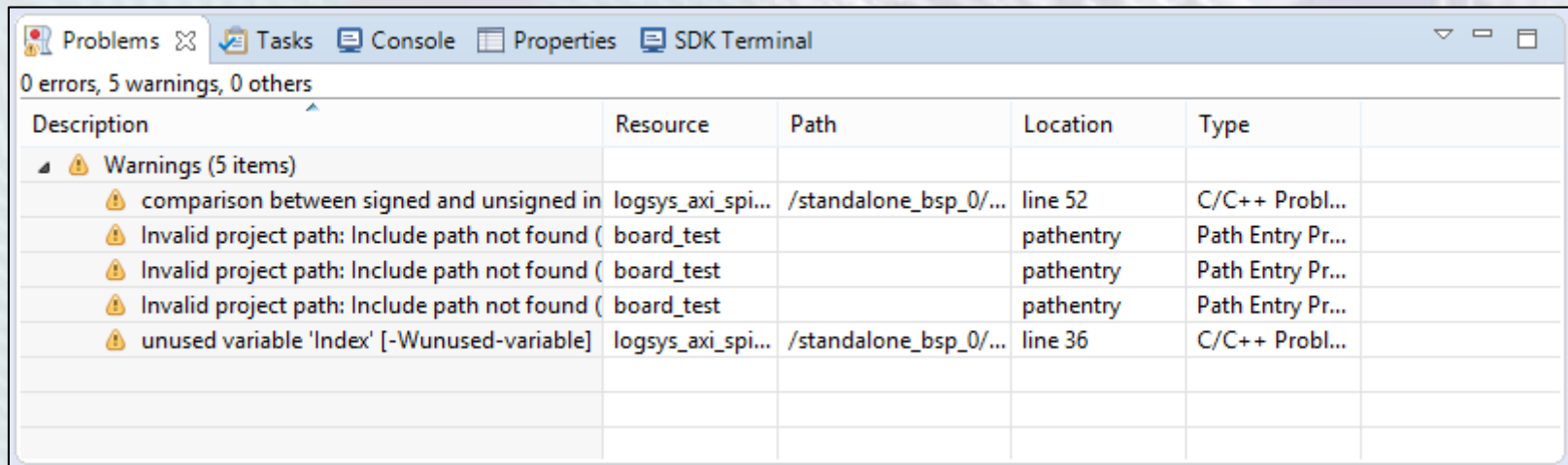
Munkaterületek és perspektívák

- **Munkaterület (workspace)**
 - Itt tárolódnak
 - A projektekkel kapcsolatos beállítások
 - Egyéb belső adatok
 - A felhasználók számára transzparens
 - A projekt fájlokat tartalmazó könyvtárak is itt vannak
- **Nézetek (views) és szerkesztők (editors)**
 - Alapelemek a felhasználói felületen
- **Perspektívák (perspectives)**
 - Azonos funkcióhoz kapcsolódó nézetek gyűjteménye
 - A nézetek szabadon elrendezhetők a perspektíván belül

Software Development Kit (SDK)

Nézetek (views)

- **Eclipse Platform nézetek:**
 - Navigátor, feladatok (tasks), problémák, stb.
- **Debug nézetek:** verem, változók, regiszterek, stb.
- **C/C++ nézetek:** projekt, vázlat (outline), stb.



The screenshot shows the Eclipse IDE's Problems view. The title bar includes 'Problems', 'Tasks', 'Console', 'Properties', and 'SDK Terminal'. The main area displays '0 errors, 5 warnings, 0 others'. A table lists the warnings:

| Description | Resource | Path | Location | Type |
|--|-------------------|-----------------------|-----------|------------------|
| Warnings (5 items) | | | | |
| comparison between signed and unsigned in | logsys_axi_spi... | /standalone_bsp_0/... | line 52 | C/C++ Probl... |
| Invalid project path: Include path not found (| board_test | | pathentry | Path Entry Pr... |
| Invalid project path: Include path not found (| board_test | | pathentry | Path Entry Pr... |
| Invalid project path: Include path not found (| board_test | | pathentry | Path Entry Pr... |
| unused variable 'Index' [-Wunused-variable] | logsys_axi_spi... | /standalone_bsp_0/... | line 36 | C/C++ Probl... |

Software Development Kit (SDK)

C/C++ perspektíva

C/C++ szerkesztő:
- A forráskód szerkesztése

C/C++ projekt nézet:
- A projekt elemeinek listája
- Ikonok az azonosításhoz

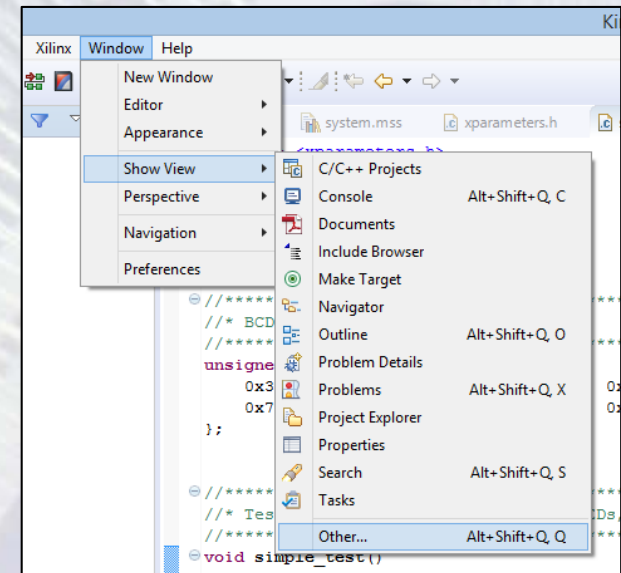
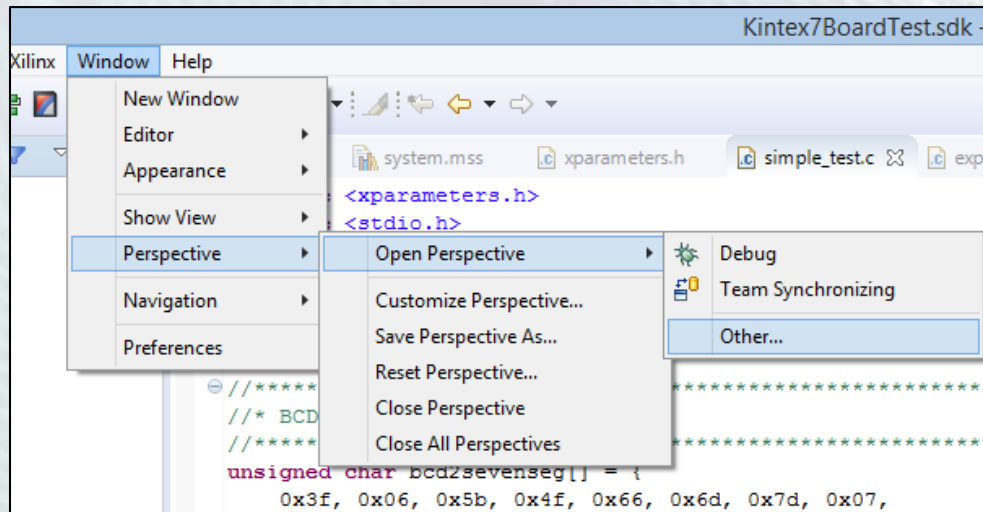
Kód vázlat (outline) nézet:
- A forráskód elemeinek listája
- Ikonok az azonosításhoz

Problems, Console és Properties nézetek:
- A fejlesztési folyamattal kapcsolatos információk

Software Development Kit (SDK)

Perspektívák és nézetek

- Perspektívák megnyitása
 - *Window* menü → *Perspective* → *Open Perspective*
- Nézetek megnyitása
 - *Window* menü → *Show View*
 - Ha a kiválasztott nézet már látható, akkor előtérbe kerül



Software Development Kit (SDK)

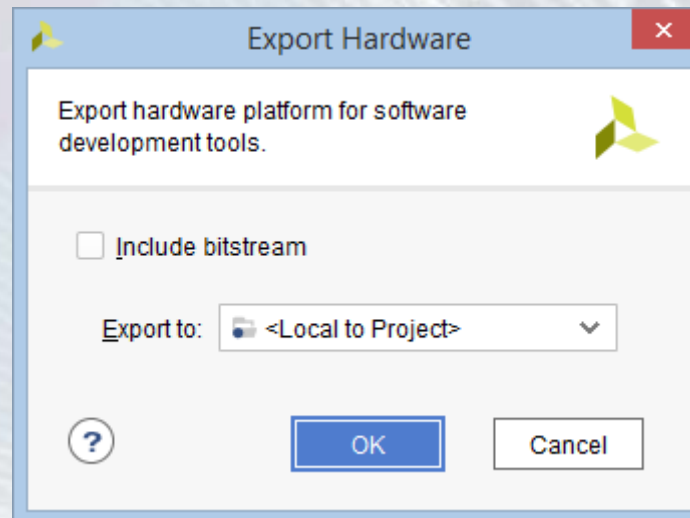
Forráskód szerkesztő

- Szintaxis kiemelés, zárójelek illesztése, content assist
- Refactoring, billentyűparancsok

```
simple_test.c
36 //Initialize the timers.
37 timer_set(&fast_timer, 5);
38 timer_set(&slow_timer, 250);
39
40 for (;;)
41 {
42     //Update the LEDs.
43     gpio_value = led_pattern;
44     if (XGpio_ReadReg(XPAR_AXI_GPIO_SW_BTN_BASEADDR, XGPIO_DATA2_OFFSET) != 0)
45     {
46         gpio_value = XGpio_ReadReg(XPAR_AXI_GPIO_SW_BTN_BASEADDR, XGPIO_DATA_OFFSET) & 0xff;
47     }
48     XGpio_WriteReg(XPAR_AXI_GPIO_LED_DISP_BASEADDR, XGPIO_DATA_OFFSET, gpio_value);
49
50     //Update the 7-segment display.
51     gpio_value = (1 << (8 + digit_select));
52     if (digit_select == current_digit)
53     {
54         if (display_value > 9)
55             gpio_value |= 0x80;
56         else
57             gpio_value |= (unsigned long)bcd2sevenseg[display_value];
58     }
59     XGpio_WriteReg(XPAR_AXI_GPIO_LED_DISP_BASEADDR, XGPIO_DATA2_OFFSET, gpio_value);
60 }
```

Hardver platform projekt

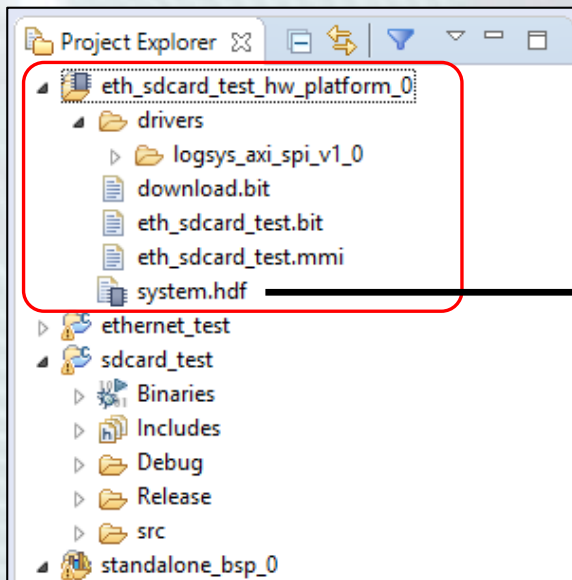
- **A hardver specifikáció exportálása az SDK-ba**
 - Vivado: File menü → Export → Export Hardware...
 - Exportálás helye
 - BIT fájljal vagy anélkül
- **Software Development Kit (SDK) elindítása**
 - Vivado: File menü → Launch SDK



Hardver platform projekt

Hardver platform: a hardver terv leírását tartalmazza

- A hardver tervet leíró fájl: *system.hdf*
- A Blokk-RAM-ok helyét tároló fájl: **.mmi*
- FPGA konfigurációs fájlok: **.bit*
- A saját perifériák meghajtói: *drivers* könyvtár



eth_sdcard_test_hw_platform_0 Hardware Platform Specification

Design Information

Target FPGA Device: 7k70t
Part: xc7k70tffbg676-1
Created With: Vivado 2017.1
Created On: Fri Jul 14 00:35:31 2017

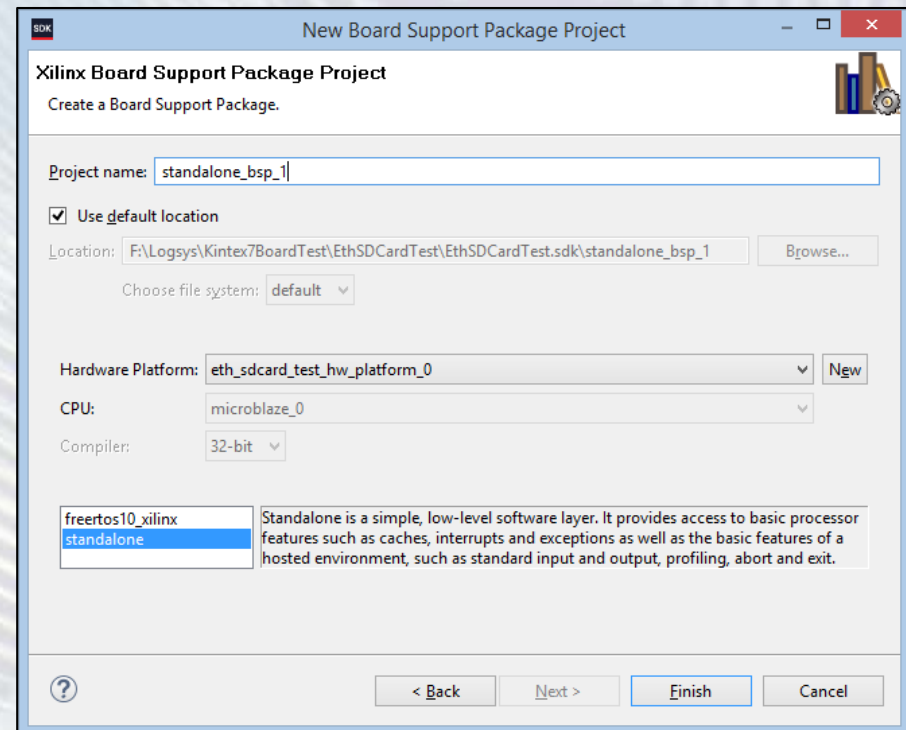
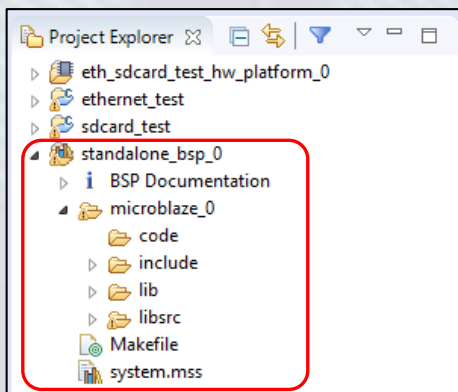
Address Map for processor microblaze_0

| Cell | Base Addr | High Addr | Slave I/f | Mem/Reg |
|----------------------------------|------------|------------|-----------|----------|
| microblaze_0_axi_intc | 0x40000000 | 0x4000ffff | s_axi | REGISTER |
| logsys_axi_spi_0 | 0x44000000 | 0x4400ffff | S00_AXI | REGISTER |
| axi_ethernetlite_0 | 0x43000000 | 0x4300ffff | S_AXI | REGISTER |
| axi_gpio_0 | 0x45000000 | 0x4500ffff | S_AXI | REGISTER |
| axi_uartlite_0 | 0x42000000 | 0x4200ffff | S_AXI | REGISTER |
| microblaze_0_local_memory_dlm... | 0x00000000 | 0x0000ffff | SLMB | MEMORY |
| axi_timer_0 | 0x41000000 | 0x4100ffff | S_AXI | REGISTER |

Board Support Package projekt

Board Support Package (BSP) létrehozása

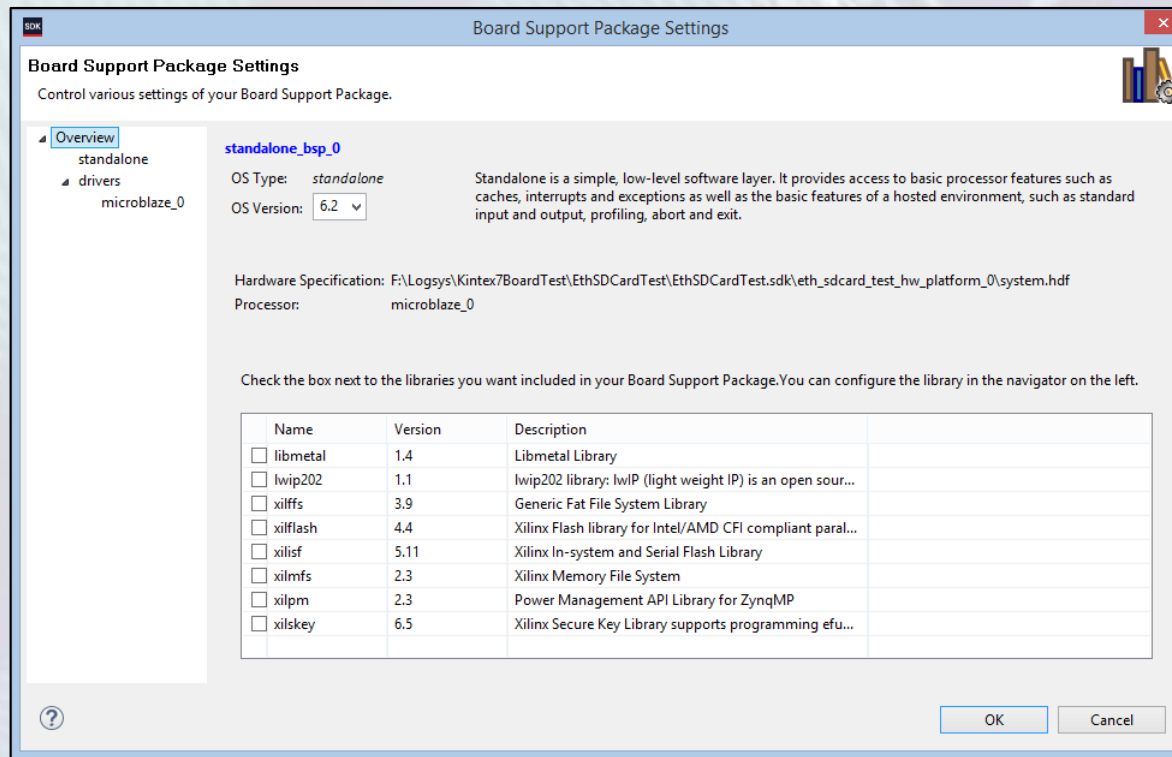
- Eszközmeghajtók, szoftver könyvtárak, dokumentációk
- *File* menü → *New* → *Board Support Package*
- Projekt nézet: jobb klikk → *New* → *Project...* → *Board Support Package*
- Megadható beállítások
 - Projekt elnevezése
 - Projekt helye
 - Cél hardver platform és CPU
 - OS (standalone vagy FreeRTOS)
- Konfigurációs fájl: `system.mss`



Board Support Package projekt

Board Support Package (BSP) beállítások

- Jobb klikk a BSP projekten → *Board Support Package Settings*
- Szoftver könyvtárak kiválasztása
- Operációs rendszer és eszközmeghajtó beállítások



Board Support Package projekt

Eszközmeghajtó programok

- Rétegzett architektúra
- 2. réteg: RTOS alkalmazási réteg
- 1. réteg: Magas szintű eszközmeghajtók
 - Teljes funkcionalitás
 - Többféle processzor és operációs rendszer támogatott
- 0. réteg: Alacsony szintű eszközmeghajtók

2. réteg: RTOS

1. réteg: magas szintű meghajtók

0. réteg: alacsony szintű meghajtók

Board Support Package projekt

0. réteg: alacsony szintű eszközmeghajtó programok

- Makrók és függvények, melyek lehetővé teszik a kisméretű rendszerek megvalósítását
- Tulajdonságok:
 - Kis memóriaigény
 - Egyáltalán nincs vagy csak kevés hibaellenőrzés
 - Csak az eszköz alapfunkcióit támogatják
 - Nem támogatják az eszköz konfigurációs paramétereit
 - Több eszközpéldány támogatása: báziscímek megadásával
 - Csak lekérdezéses I/O
 - Blokkoló hívások
 - Header fájlok végződése “_l” (például xuartlite_l.h)

Board Support Package projekt

1. réteg: magas szintű eszközmeghajtó programok

- Makrók és függvények, melyek lehetővé teszik az eszközök minden tulajdonságainak kihasználását
- Tulajdonságok:
 - Absztrakt API, amely leválasztja a programozói interfészt a hardver rendszer változásairól
 - Támogatják az eszköz konfigurációs paramétereit
 - Több eszközpéldány támogatása
 - Lekérdezéses és megszakításos I/O
 - Nem blokkoló hívások a komplex alkalmazások támogatásához
 - Nagy memóriaigény lehetséges
 - Tipikusan pufferelt adatátvitel a bájtos adatátvitel helyett
 - Header fájlok végződése nem “_l” (például xuartlite.h)

Board Support Package projekt

- **Uartlite magas szintű meghajtó**

- XStatus XUartLite_Initialize(XUartLite *InstancePtr, Xuint16 DeviceId)
- void XUartLite_ResetFifos(XUartLite *InstancePtr)
- unsigned int XUartLite_Send(XUartLite *InstancePtr, Xuint8 *DataBufferPtr, unsigned int NumBytes)
- unsigned int XUartLite_Recv(XUartLite *InstancePtr, Xuint8 *DataBufferPtr, unsigned int NumBytes)
- Xboolean XUartLite_IsSending(XUartLite *InstancePtr)
- void XUartLite_GetStats(XUartLite *InstancePtr, XUartLite_Stats *StatsPtr)
- void XUartLite_ClearStats(XUartLite *InstancePtr)
- XStatus XUartLite_SelfTest(XUartLite *InstancePtr)
- void XUartLite_EnableInterrupt(XUartLite *InstancePtr)
- void XUartLite_DisableInterrupt(XUartLite *InstancePtr)
- void XUartLite_SetRecvHandler(XUartLite *InstancePtr, XUartLite_Handler FuncPtr, void *CallBackRef)
- void XUartLite_SetSendHandler(XUartLite *InstancePtr, XUartLite_Handler FuncPtr, void *CallBackRef)
- void XUartLite_InterruptHandler(XUartLite *InstancePtr)

- **Uartlite alacsony szintű meghajtó**

- void XUartLite_SendByte(Xuint32 BaseAddress, Xuint8 Data)
- Xuint8 XUartLite_RecvByte(Xuint32 BaseAddress)

Board Support Package projekt

Szoftver könyvtárak

- Matematikai könyvtár (*libm*)
- Standard C könyvtár (*libc*)
 - A könyvtár függvényei automatikusan rendelkezésre állnak
- Xilinx C nyelvű meghajtók és könyvtárak (*libxil*)
 - FAT fájlrendszer: *xilffs*
 - Memória fájlrendszer: *xilmfs*
 - TCP/IP hálózati kommunikáció: *lwip202*
 - Párhuzamos flash memória támogatás: *xilflash*
 - Soros flash memória támogatás: *xilisf*
 - Stb.

Board Support Package projekt

xparameters.h header fájl

- A rendszerben lévő hardver egységek paramétereit tárolja
- Elnevezési konvenció: *XPAR_[periféria_név]_[paraméter_név]*

```
/* Definitions for driver GPIO */
#define XPAR_XGPIO_NUM_INSTANCES 3

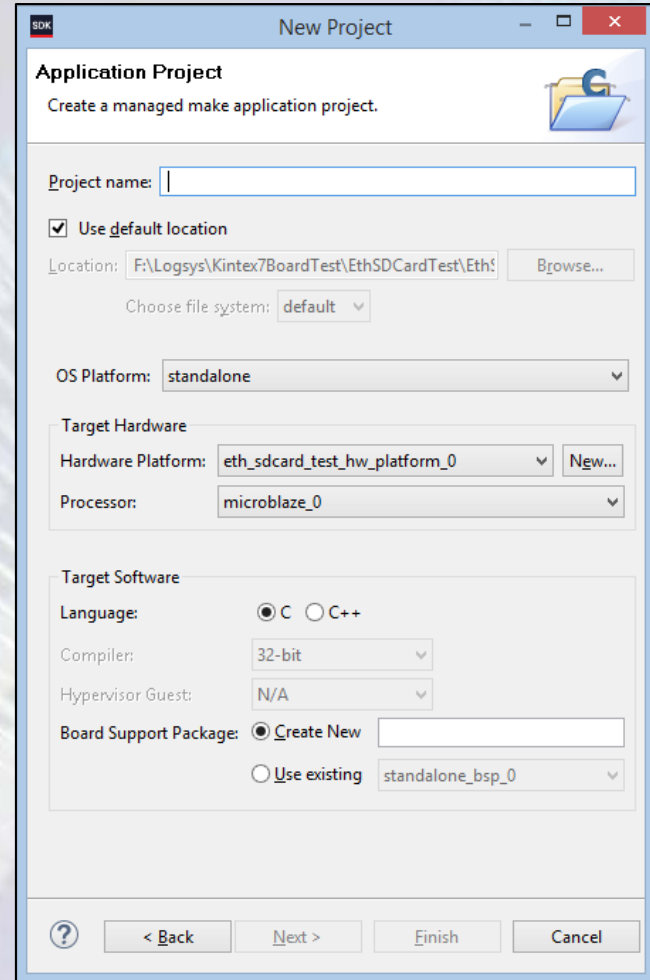
/* Definitions for peripheral AXI_GPIO_EXP_STAT */
#define XPAR_AXI_GPIO_EXP_STAT_BASEADDR 0x46000000
#define XPAR_AXI_GPIO_EXP_STAT_HIGHADDR 0x4600FFFF
#define XPAR_AXI_GPIO_EXP_STAT_DEVICE_ID 0
#define XPAR_AXI_GPIO_EXP_STAT_INTERRUPT_PRESENT 1
#define XPAR_AXI_GPIO_EXP_STAT_IS_DUAL 1

/* Definitions for peripheral AXI_GPIO_LED_DISP */
#define XPAR_AXI_GPIO_LED_DISP_BASEADDR 0x44000000
#define XPAR_AXI_GPIO_LED_DISP_HIGHADDR 0x4400FFFF
#define XPAR_AXI_GPIO_LED_DISP_DEVICE_ID 1
#define XPAR_AXI_GPIO_LED_DISP_INTERRUPT_PRESENT 0
#define XPAR_AXI_GPIO_LED_DISP_IS_DUAL 1
```

Szoftver alkalmazás projekt

Szoftver alkalmazás projekt létrehozása

- **Application Project**
 - *File* menü → *New*
 - *Projekt nézet*: jobb klikk → *New* → *Project*
- **Megadható beállítások**
 - Projekt neve és helye
 - Operációs rendszer
 - Cél hardver: HW platform, processzor
 - Cél szoftver: nyelv (C, C++), BSP választása
- **Minta alkalmazások: *Next* gomb**
 - Üres alkalmazás projekt
 - Memória teszt, periféria teszt
 - Bootloader
 - Hálózati alkalmazások (lwip)
 - Stb.



The screenshot shows the 'New Project' dialog box in the SDK. The dialog is titled 'Application Project' and contains the following fields and options:

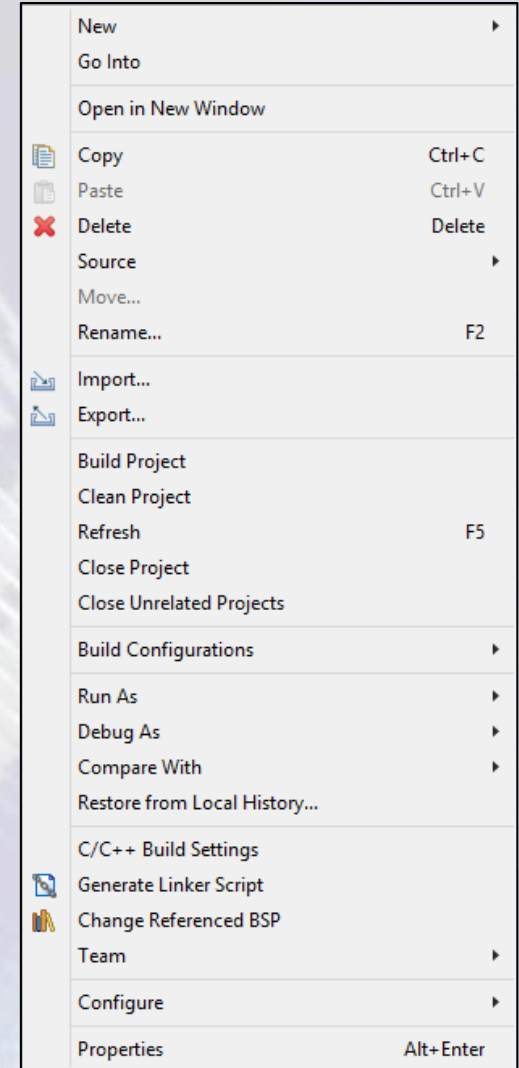
- Project name:** A text input field.
- Use default location**
- Location:** A text input field containing 'F:\Logsys\Kintex7BoardTest\EthSDCardTest\Eth...' and a 'Browse...' button.
- Choose file system:** A dropdown menu set to 'default'.
- OS Platform:** A dropdown menu set to 'standalone'.
- Target Hardware**
 - Hardware Platform:** A dropdown menu set to 'eth_sdcard_test_hw_platform_0' and a 'New...' button.
 - Processor:** A dropdown menu set to 'microblaze_0'.
- Target Software**
 - Language:** Radio buttons for 'C' (selected) and 'C++'.
 - Compiler:** A dropdown menu set to '32-bit'.
 - Hypervisor Guest:** A dropdown menu set to 'N/A'.
 - Board Support Package:** Radio buttons for 'Create New' (selected) and 'Use existing'. The 'Use existing' option has a dropdown menu set to 'standalone_bsp_0'.

At the bottom of the dialog, there are navigation buttons: '?', '< Back', 'Next >', 'Finish', and 'Cancel'.

Szoftver alkalmazás projekt

Projekt nézet: jobb klikk → projekt felugró menü

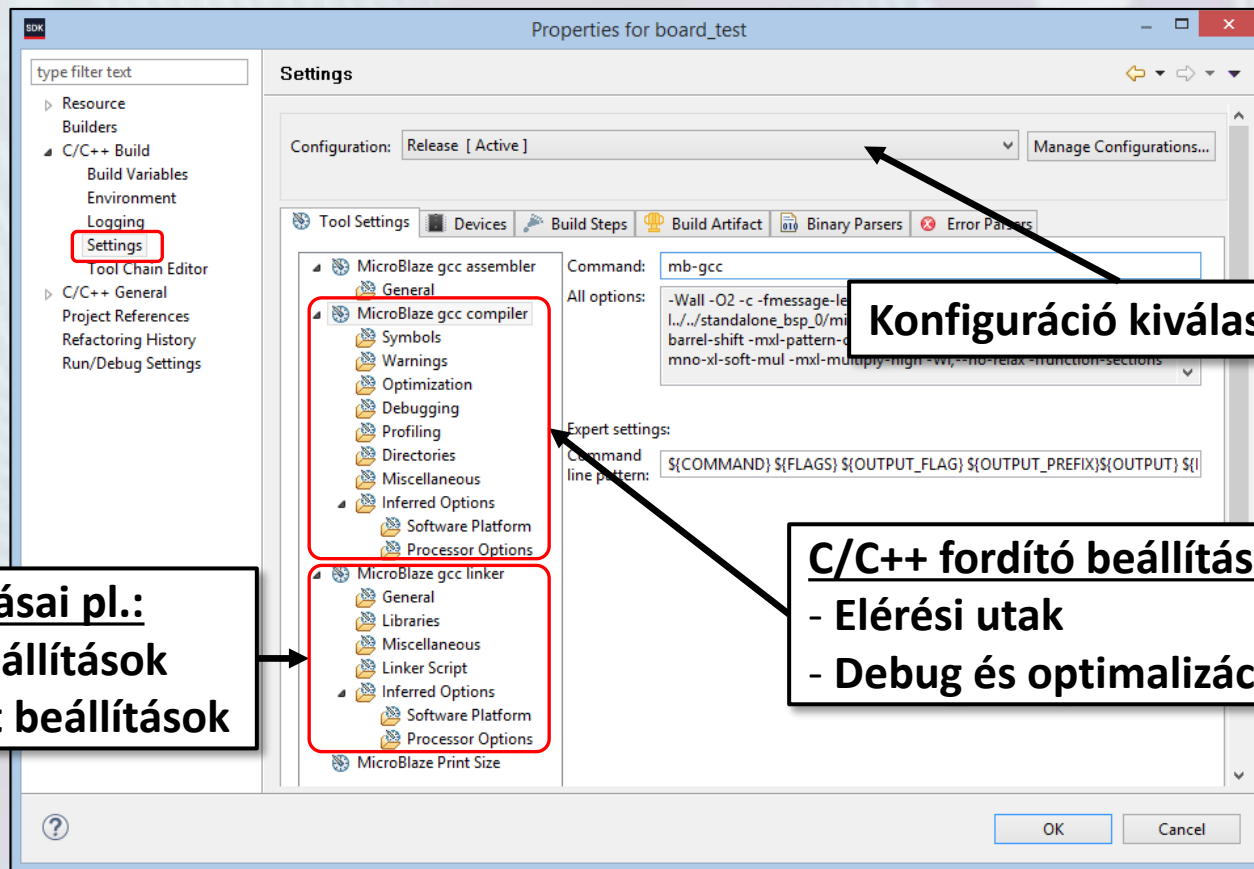
- Új forrásfájl hozzáadása
 - New → Source File / Header File
- A projekt lefordítása
 - Build Project / Rebuild Project
- Másolás, beillesztés, törlés, átnevezés
- Importálás
 - Tömörített fájlból, fájlrendszerből, stb.
- Exportálás
 - Tömörített fájlba, fájlrendszerbe, stb.
- Konfiguráció kiválasztás: *debug, release, egyéni*
- Linker script generálása
- Alkalmazás futtatása
- Alkalmazás debuggolása
- A kiválasztott elem tulajdonságai



Szoftver projekt beállítások

A C/C++ fordító és a linker beállításai

- Projekt tulajdonságai → C/C++ Build → Settings



Konfiguráció kiválasztása

Linker beállításai pl.:
- Könyvtár beállítások
- Linker script beállítások

C/C++ fordító beállításai pl.:
- Elérési utak
- Debug és optimalizáció

Szoftver projekt beállítások

Linker script létrehozása

- A tárgykód és a végrehajtható fájl szekciókból áll
 - **.text**: végrehajtható kód
 - **.rodata**: csak olvasható adatok
 - **.sdata2**: kis méretű (max. 7 byte), csak olvasható adatok
 - **.sbss2**: kis méretű, nem inicializált, csak olvasható adatok
 - **.data**: írható/olvasható adatok
 - **.sdata**: kis méretű, írható/olvasható adatok
 - **.sbss**: kis méretű, nem inicializált adatok
 - **.bss**: nem inicializált adatok
 - **.heap**: szekció a dinamikus memória foglaláshoz
 - **.stack**: verem szekció
- A linker script (többek között) azt mondja meg, hogy hol helyezkedjenek el az egyes szekciók a memórián belül

Szoftver projekt beállítások

Linker script létrehozása

- Projekt tulajdonságai → Generate Linker Script... (Basic)

A kimeneti fájl megadása

Project: board_test
Output Script:
7BoardTest\Kintex7BoardTest\Kintex7BoardTest.sdk\board_test\src\lscript.ld Browse

Modify project build settings as follows:
Set generated script on all project build configurations

| Memory | Base Address | Size |
|----------------------------------|--------------|-------|
| ddr3_sdram_memaddr | 0x80000000 | 1 GB |
| microblaze_0_local_memory_ilm... | 0x00000000 | 64 KB |

Fixed Section Assignments

| Section | Assigned Memory | Address |
|---------|-----------------|---------|
|---------|-----------------|---------|

A heap és a verem méretének beállítása

Heap Size: 1 KB
Stack Size: 1 KB

A szekciók hozzárendelése a memóriákhoz

Place Code Sections in: ntr_Mem_microblaze_0_local_memory_dlmbram_if_cntlr_Mem
Place Data Sections in: ntr_Mem_microblaze_0_local_memory_dlmbram_if_cntlr_Mem
Place Heap and Stack in: ntr_Mem_microblaze_0_local_memory_dlmbram_if_cntlr_Mem

Generate Cancel

Szoftver projekt beállítások

Linker script létrehozása

- Projekt tulajdonságai → Generate Linker Script... (Advanced)

A kód szekciók hozzárendelése a memóriákhoz

**Új kód szekció hozzáadása
Meglévő kód szekció törlése**

A kimeneti fájl megadása

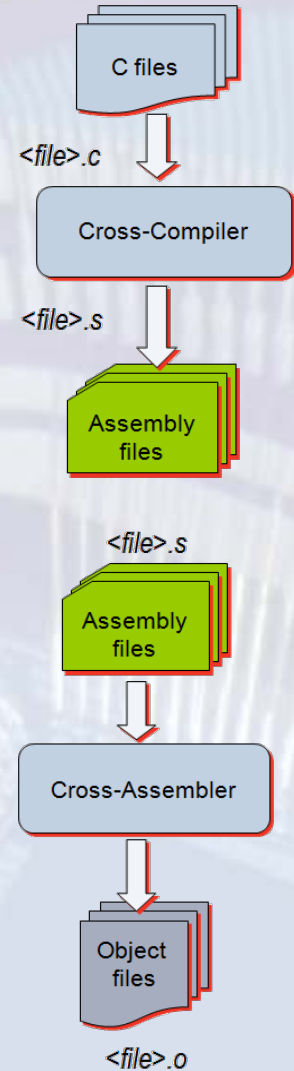
Az adat szekciók hozzárendelése a memóriákhoz

A heap és a verem szekciók beállítása (méret, memória)

Generate **Cancel**

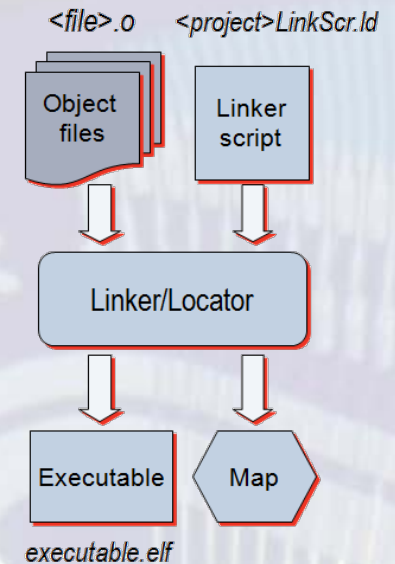
GNU fejlesztőeszközök

- Az SDK a GNU fejlesztőeszközöket használja
 - Parancssoros eszközök
- GNU C/C++ fordító
 - Négy programot futtat le egymás után
 - *Preprocesszor*: feldolgozza a preprocesszor direktívákat és a makrókat (helyettesítés)
 - *Nyelvspecifikus fordító*: assembly forrásfájlt készít a C vagy C++ kódból (.c, .cpp, .cc → .s)
 - *Assembler*: az assembly forrásfájlból tárgykódot (object fájl) készít (.s → .o)
 - *Linker*: a tárgykód fájllokból és a könyvtárakból elkészíti a futtatható fájlt (.o, .a → .elf)



GNU fejlesztőeszközök

- Az assembler és a linker külön is futtatható
- A tárgykód (object) fájl tartalma
 - Az assembler által előállított gépi kód
 - Konstans adatok, külső referenciák
 - Debug információk
- Archiver:
 - Szoftver könyvtár létrehozása tárgykód fájlokból
 - Szoftver könyvtár tartalmának módosítása
 - Tárgykód fájlok kinyerése szoftver könyvtárból
 - Az SDK pl. a BSP könyvtár létrehozásához használja

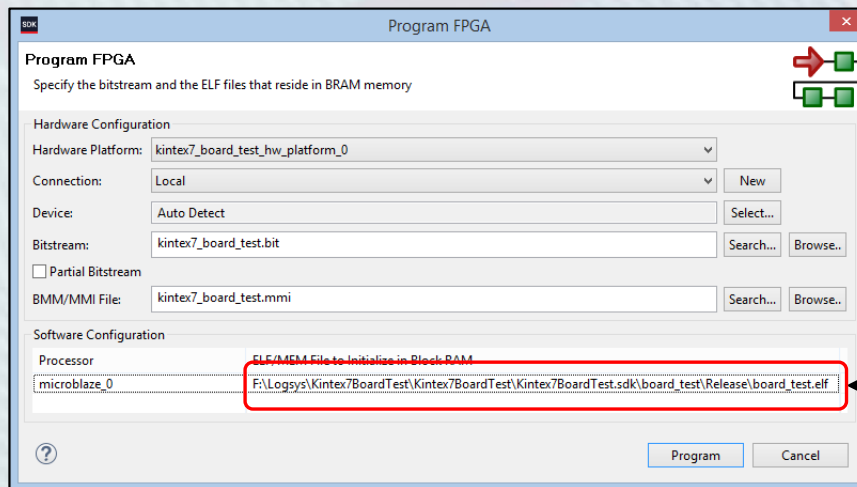


Alkalmazás futtatása, debuggolása

Az FPGA konfigurálása, a lefordított alkalmazás futtatása

- Az FPGA konfigurálása

- *Xilinx* menü → *Program FPGA...* vagy a  gomb
- Kiválasztható a Blokk-RAM-ba kerülő program




A Blokk-RAM-ba
kerülő program
kiválasztása

- A lefordított szoftver alkalmazás letöltése és futtatása

- *Run* menü vagy projekt felugró menü → *Run As* → *Launch on Hardware (System Debugger)* vagy a  gomb

Alkalmazás futtatása, debuggolása

A szoftver alkalmazás debuggolása

- **Az alkalmazás lefordítása: debug konfigurációval**
 - Az optimalizáció le van tiltva
 - A debug szimbólumok generálása engedélyezett
- **A debugger elindítása**
 - *Run* menü vagy projekt felugró menü → *Debug As* →
→ *Launch on Hardware (System Debugger)*
 - A  gombra kattintva

Alkalmazás futtatása, debuggolása

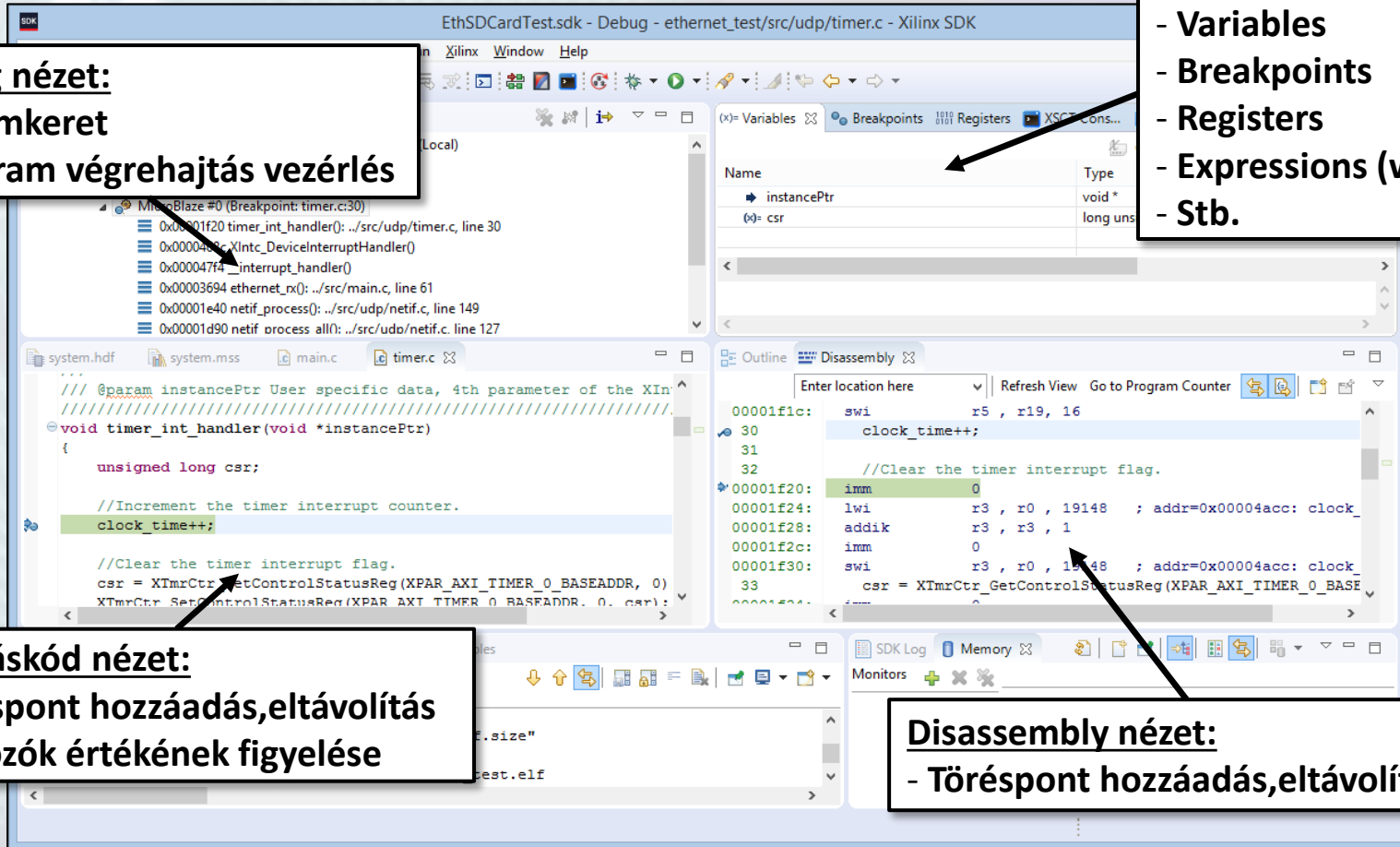
A debug perspektíva

Debug nézet:

- Veremkeret
- Program végrehajtás vezérlés

Egyéb nézetek pl.:

- Variables
- Breakpoints
- Registers
- Expressions (watch)
- Stb.



C forráskód nézet:

- Töréspont hozzáadás, eltávolítás
- Változók értékének figyelése

Disassembly nézet:

- Töréspont hozzáadás, eltávolítás

Alkalmazás futtatása, debuggolása

A debugger funkciói

- **Töréspontok elhelyezése és eltávolítása:**
 - Dupla kattintás a sor száma mellett a szürke területen

```
32 //Clear the timer interrupt flag.  
33 csr = XTmrCtr_GetControlStatusReg(XPAR_AXI_TIMER_0_BASEADDR, 0);  
34 XTmrCtr_SetControlStatusReg(XPAR_AXI_TIMER_0_BASEADDR, 0, csr);
```

- **A program végrehajtás vezérlése**
 - **Resume:** a program futásának folytatása
 - **Suspend:** a program futásának megállítása
 - **Step Into:** az aktuális forráskód sor végrehajtása
 - Függvényhívás esetén belép a függvénybe
 - **Step Over:** az aktuális forráskód sor végrehajtása
 - Függvényhívás esetén lefut a függvény, nem lép be a függvénybe
 - **Step Return:** a futás leáll a függvényből való kilépéskor
 - **Run to Line:** futtatás a kijelölt forráskód sorig

Alkalmazás futtatása, debuggolása

A debugger funkciói: nézetek

- **Forráskód nézet (C, disassembly)**
 - Töréspontok elhelyezése, eltávolítása
 - Változók értékeinek megtekintése
 - Vigyük az egérkurzort a változó fölé
- **Debug nézet**
 - Veremkeret
 - A program végrehajtás vezérlése
- **Variables: a lokális változók listája, értékeik módosítása**
- **Breakpoints: töréspontok engedélyezése, tiltása**
- **Registers: a CPU regiszterek listája, értékeik módosítása**
- **Expressions: kifejezések értékének figyelése (watch)**
- **Memory: memóriatartalom megjelenítése, módosítása**

Példák

Nagyon egyszerű szoftver alkalmazás (1):

- Két 8 bites GPIO periféria: LED-ek, kapcsolók
- A kapcsolók állapotát megjelenítjük a LED-eken
- A perifériákat közvetlenül kezeljük, nem használjuk az eszközmeghajtó programokat
- GPIO regiszterkészlet: a periféria adatlapjában vagy *system.hdf* → *IP blocks* csoport → *Registers* link

| Address Space Offset ⁽³⁾ | Register Name | Access Type | Default Value | Description |
|-------------------------------------|-----------------------|----------------------|---------------|--|
| 0x0000 | GPIO_DATA | R/W | 0x0 | Channel 1 AXI GPIO Data Register. |
| 0x0004 | GPIO_TRI | R/W | 0x0 | Channel 1 AXI GPIO 3-state Control Register. |
| 0x0008 | GPIO2_DATA | R/W | 0x0 | Channel 2 AXI GPIO Data Register. |
| 0x000C | GPIO2_TRI | R/W | 0x0 | Channel 2 AXI GPIO 3-state Control. |
| 0x011C | GIER ⁽¹⁾ | R/W | 0x0 | Global Interrupt Enable Register. |
| 0x0128 | IP IER ⁽¹⁾ | R/W | 0x0 | IP Interrupt Enable Register (IP IER). |
| 0x0120 | IP ISR ⁽¹⁾ | R/TOW ⁽²⁾ | 0x0 | IP Interrupt Status Register. |

Példák

Nagyon egyszerű szoftver alkalmazás (1):

- I/O makrók: az *xio.h* fájlban vannak definiálva
- Memória írás
 - 8 bit: ***XIo_Out8(OutputPtr, Value)***
 - 16 bit: ***XIo_Out16(OutputPtr, Value)***
 - 32 bit: ***XIo_Out32(OutputPtr, Value)***
- Memória olvasás
 - 8 bit: ***XIo_In8(InputPtr)***
 - 16 bit: ***XIo_In16(InputPtr)***
 - 32 bit: ***XIo_In32(InputPtr)***

Példák

Nagyon egyszerű szoftver alkalmazás (1):

```
#include <xio.h>
#include <xparameters.h>

int main()
{
    unsigned long data;

    //A GPIO portok irányának beállítása.
    XIo_Out32(XPAR_LEDS_BASEADDR + 0x04, 0x00);
    XIo_Out32(XPAR_SWITCHES_BASEADDR + 0x04, 0xff);

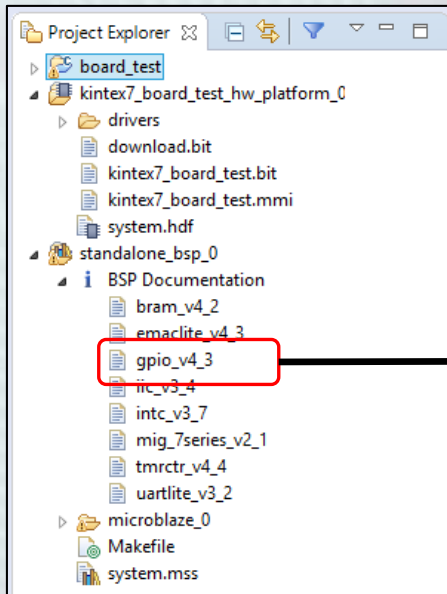
    //Végtelen ciklus.
    while (1)
    {
        data = XIo_In32(XPAR_SWITCHES_BASEADDR + 0x00);
        XIo_Out32(XPAR_LEDS_BASEADDR + 0x00, data);
    }

    return 0;
}
```

Példák

Nagyon egyszerű szoftver alkalmazás (2):

- Az előző példa alacsony szintű meghajtó használatával
- Eszközpéldány azonosítása: báziscímek alapján
- Eszközmeghajtók dokumentációja:
 - *Board Support Package* projekt → *BSP Documentation*



```
#define XGpio_ReadReg ( BaseAddress,  
                        RegOffset  
                    ) XGpio_In32((BaseAddress) + (RegOffset))  
  
#include <xgpio_1.h>  
  
Read a value from a GPIO register.  
  
A 32 bit read is performed. If the GPIO core is implemented in a smaller width, only the least significant data is read from the register. The most significant data will be read as 0.  
  
Parameters  
    BaseAddress is the base address of the GPIO device.  
    RegOffset is the register offset from the base to read from.  
  
Returns  
    Data read from the register.  
  
Note  
    C-style signature: u32 XGpio_ReadReg(u32 BaseAddress, u32 RegOffset)  
  
Referenced by XGpio_DiscreteRead(), XGpio_GetDataDirection(), XGpio_InterruptClear(), XGpio_InterruptDisable(), XGpio_InterruptEnable(), XGpio_InterruptGetEnabled(), and XGpio_InterruptGetStatus().
```

Példák

Nagyon egyszerű szoftver alkalmazás (2):

```
#include <xgpio_1.h>
#include <xparameters.h>

int main()
{
    unsigned long data;

    //A GPIO portok irányának beállítása.
    XGpio_WriteReg(XPAR_LEDS_BASEADDR, XGPIO_TRI_OFFSET, 0x00);
    XGpio_WriteReg(XPAR_SWITCHES_BASEADDR, XGPIO_TRI_OFFSET, 0xff);

    //Végtelen ciklus.
    while (1)
    {
        data = XGpio_GetDataReg(XPAR_SWITCHES_BASEADDR, 1);
        XGpio_SetDataReg(XPAR_LEDS_BASEADDR, 1, data);
    }

    return 0;
}
```

Példák

Nagyon egyszerű szoftver alkalmazás (3):

- Az első példa magas szintű meghajtó használatával
- Eszközpéldány azonosítása: eszköz azonosító alapján
- Hasonlítsuk össze a három példa esetén a kód méreteket

```
#include <xgpio.h>
#include <xparameters.h>

//A GPIO perifériák leírói.
XGpio leds;
XGpio switches;

int main()
{
    unsigned long data;

    //A GPIO leírók inicializálása.
    XGpio_Initialize(&leds, XPAR_LEDS_DEVICE_ID);
    XGpio_Initialize(&switches, XPAR_SWITCHES_DEVICE_ID);
```


Példák

Nagyon egyszerű szoftver alkalmazás (3):

```
//A GPIO portok irányának beállítása.
XGpio_SetDataDirection(&leds, 1, 0x00);
XGpio_SetDataDirection(&switches, 1, 0xff);

//Végtelen ciklus.
while (1)
{
    data = XGpio_DiscreteRead(&switches, 1);
    XGpio_DiscreteWrite(&leds, 1, data);
}

return 0;
}
```

Példák – Megszakítások kezelése

- **Példa: időzítő megszakítások kezelése**
 - Másodperc számláló értékének megjelenítése a LED-eken
- **Lépések**
 - A megszakításkezelő rutin regisztrálása
 - A perifériához tartozik egy megszakítás azonosító érték
 - A megszakítás vezérlő konfigurálása
 - Globális megszakítás engedélyezés
 - A megfelelő megszakítás bemenet engedélyezése
 - Megszakítások engedélyezése a MicroBlaze processzoron
 - Az időzítő (ill. a megszakítást kérő perifériák) konfigurálása
 - A periódusregiszter beállítása
 - A megszakítások engedélyezése
- **A megszakításkezelő rutinban használt globális változók**
 - ***volatile*** módosító: optimalizálás letiltása a változóra
 - Nem atomi műveletek (pl. i++): megszakítások tiltása, művelet végrehajtása, megszakítások engedélyezése (szinkronizáció)

Példák – Megszakítások kezelése

Megszakítások kezelése

```
#include <xtmrctr.h>
#include <xintc_1.h>
#include <xgpio.h>
#include <mb_interface.h>
#include <xparameters.h>

//Globális változók.
XGpio leds;
volatile unsigned long ledValue;

//Megszakításkezelő rutin.
void timer_int_handler(void *instancePtr)
{
    unsigned long csr;

    //Az új érték kiírása a LED-ekre.
    ledValue = (ledValue + 1) & 0xff;
    XGpio_DiscreteWrite(&leds, 1, ledValue);
    //A megszakítás jelzés törlése.
    csr = XTmrCtr_GetControlStatusReg(XPAR_XPS_TIMER_0_BASEADDR, 0);
    XTmrCtr_SetControlStatusReg(XPAR_XPS_TIMER_0_BASEADDR, 0, csr);
}
```

Példák – Megszakítások kezelése

Megszakítások kezelése

```
int main()
{
    //A LED GPIO inicializálása.
    ledValue = 0;
    XGpio_Initialize(&leds, XPAR_LEDS_8BIT_DEVICE_ID);
    XGpio_SetDataDirection(&leds, 1, 0x00);
    XGpio_DiscreteWrite(&leds, 1, ledValue);

    //A megszakításkezelő rutin beállítása.
    XIntc_RegisterHandler(
        XPAR_XPS_INTC_BASEADDR, //INTC báziscíme
        XPAR_XPS_INTC_XPS_TIMER_INTERRUPT_INTR, //Megszakítás azonosító
        (XInterruptHandler) timer_int_handler, //Megszakításkezelő rutin
        NULL //Megsz. kezelő rutin paramétere
    );

    //A megszakítás vezérlő konfigurálása.
    XIntc_MasterEnable(XPAR_XPS_INTC_BASEADDR);
    XIntc_EnableIntr(XPAR_XPS_INTC_BASEADDR, XPAR_XPS_TIMER_INTERRUPT_MASK);

    //A megszakítások engedélyezése a processzoron.
    microblaze_enable_interrupts();
}
```

Példák – Megszakítások kezelése

Megszakítások kezelése

```
//A timer LOAD regiszterének beállítása (megszakítás másodpercenként).
XTmrCtr_SetLoadReg(XPAR_XPS_TIMER_BASEADDR, 0, XPAR_PROC_BUS_0_FREQ_HZ);
//A timer alapállapotba állítása.
XTmrCtr_SetControlStatusReg(
    XPAR_XPS_TIMER_BASEADDR,
    0,
    XTC_CSR_INT_OCCURED_MASK | XTC_CSR_LOAD_MASK
);
//A timer elindítása.
XTmrCtr_SetControlStatusReg(
    XPAR_XPS_TIMER_BASEADDR,
    0,
    XTC_CSR_ENABLE_TMR_MASK | XTC_CSR_ENABLE_INT_MASK |
    XTC_CSR_AUTO_RELOAD_MASK | XTC_CSR_DOWN_COUNT_MASK
);

//Végtelen ciklus.
while (1);

return 0;
}
```