



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

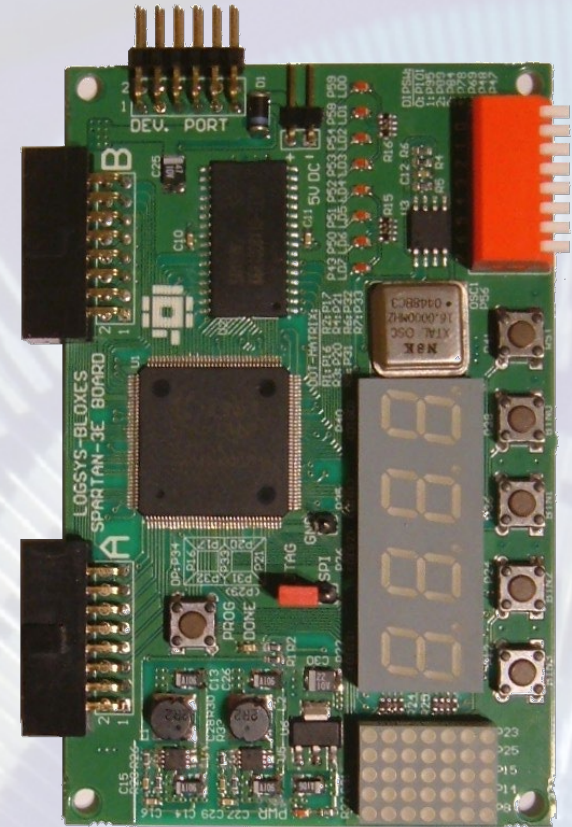
Digitális technika (VIMIAA03)

9. gyakorlat és laboratórium

Raikovich Tamás
BME MIT

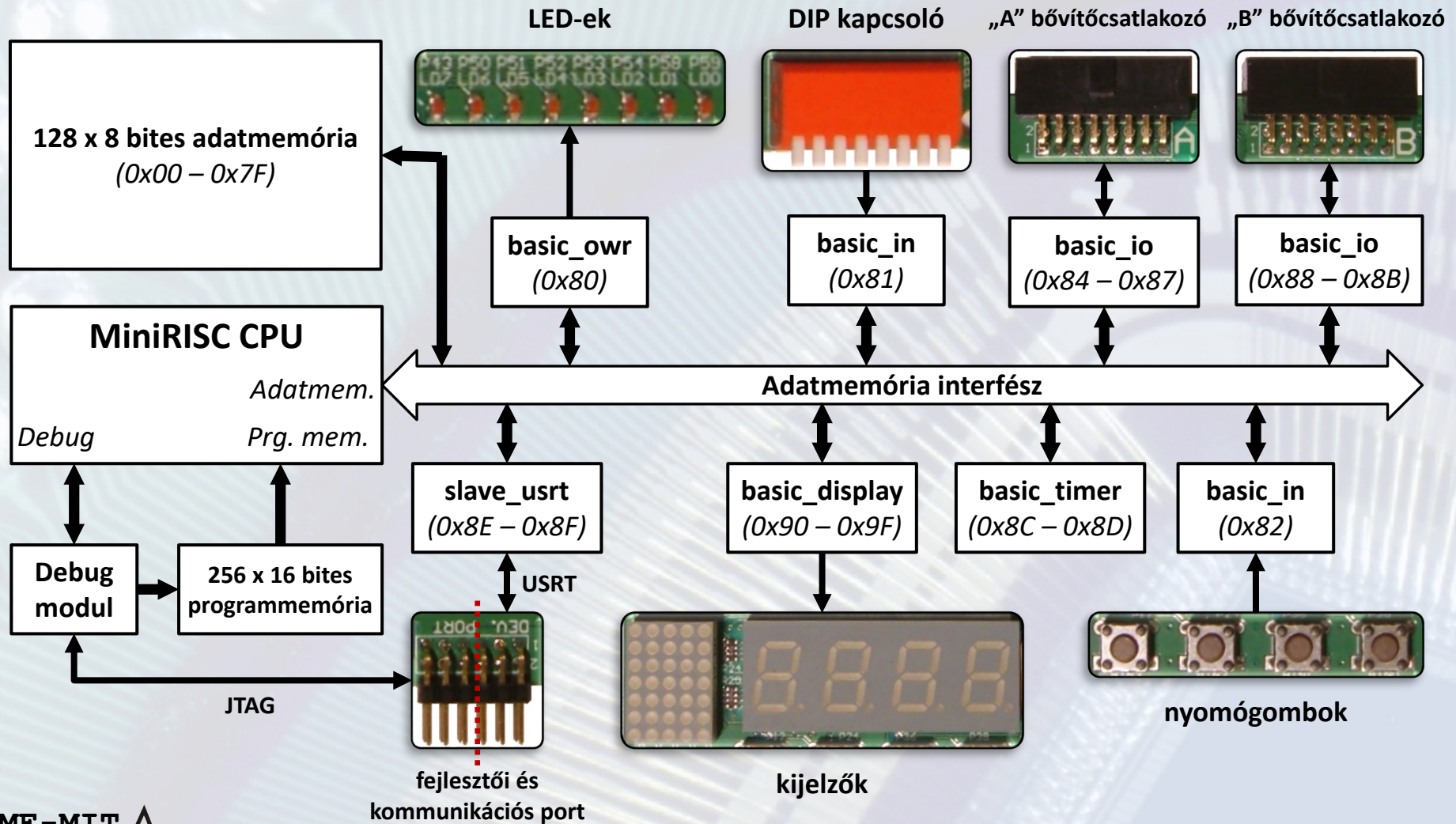
Bevezetés: MiniRISC processzor

- 8 bites vezérlőegység egyszerű alkalmazásokhoz
- Jól illeszkedik a LOGSYS Spartan-3E FPGA kártya komplexitásához
- Egyszerű felépítés, kis erőforrásigény
- Harvard architektúra
 - 256 x 16 bites programmemória
 - 256 x 8 bites adatmemória
- Egyszerű RISC jellegű utasításkészlet
 - Load/store architektúra
 - Műveletvégzés csak regisztereken
 - 16 x 8 bites belső regisztertömb



Bevezetés: MiniRISC mintarendszer

(Egyszerűsített blokkvázlat)



Bevezetés: MiniRISC assembler

- A LOGSYS Spartan-3E FPGA kártyához készült MiniRISC mintarendszere történő programfejlesztéshez egy grafikus fejlesztői környezet áll rendelkezésre (MiniRISC IDE)
- Az assembly nyelven megírt programok lefordítása a LOGSYS MiniRISC assemblerrel lehetséges
- Az assembler a processzorhoz illesztett képességű
 - Egyszerű utasítás értelmezés
 - Abszolút kódot generál (nincs linker)
 - Nincs makrózás
 - Nincs cím aritmetika
 - Nincs feltételes fordítás
 - Hiba esetén az első hibánál hibaüzenettel leáll

Bevezetés: MiniRISC assembler

- Forrásfájl: egyszerű szövegfájl .s kiterjesztéssel
- A feldolgozás soronként történik: 1 sorban 1 utasítás
- Az assembly forráskód sorok formátuma
LABEL: MNEMONIC OP1{, OP2} ; Comment
 - ***LABEL*** Azonosító, értéke az azt követő utasítás címe
 - ***MNEMONIC*** A végrehajtandó műveletre utaló mnemonik, pl. ***ADD*** – összeadás, ***JMP*** – ugrás, stb.
 - ***OP1{, OP2}*** A művelet operandusai, ***OP2*** nincs mindig (***OP1*** sincs az ***RTS***, ***RTI***, ***STI*** és ***CLI*** utasítások esetén)
 - ***; Comment*** A ';' karakter a megjegyzés kezdete, melyet az assembler figyelmen kívül hagy
- Javaslat
 - Minden utasításhoz írjunk megjegyzéseket
 - A formázáshoz használjuk a TAB karaktert

Bevezetés: MiniRISC assembler

- **Assembler direktívák**

- **DEF:** azonosítót rendel konstanshoz

- DEF SW 0x81 ;A DIP kapcsoló periféria címe**

- Ez nem CPU utasítás, hanem az assembler számára definiál egy helyettesítési szabályt, ami a felhasználó számára biztosítja a forráskód jobb olvashatóságát

- **CODE:** a kód szekció kezdetét jelzi

- Az utána lévő forráskód részből generált kód a prg. memóriába kerül

- **DATA:** az adat szekció kezdetét jelzi

- Az utána lévő forráskód részből generált kód az adatmemóriába kerül
 - Az adat szekcióban csak címkék és DB direktívák lehetnek

- **DB:** az adatmemória inicializálása konstansokkal

- DB "MiniRISC.\r\n", 0 ;Nulla végű string**

- Csak az adat szekcióban lehet használni
 - Numerikus, karakter és string konstansok adhatók meg utána
 - Az egyes konstansokat vesszővel kell elválasztani egymástól

Bevezetés: MiniRISC assembler

- **Assembler direktívák**
 - **ORG:** kezdőcím közvetlen előírása
ORG memóriacím
 - Az utána lévő kódrész kezdőcímét állítja be
 - Használható a kód és az adat szekcióban is
- Az assembler által generált LST fájlban ellenőrizhető az utasítások címe és gépi kódja, valamint a fordító által használt azonosítók értelmezése
- Ha a kód hibátlanul lefordult, akkor beépíthető a hardver tervbe, mint memória inicializáló adat (.HEX kimeneti fájlok) vagy letölthető a kész konfigurációra (.SVF kimeneti fájl)

Bevezetés: MiniRISC IDE

Futtatás:

- Szimulátorban
- Hardveren

Forráskód szerkesztő

Adatmemória tartalma

Kijelző vezérlőpanel

GPIO vezérlőpanel

Periféria vezérlőpanel:

- LED-ek, DIP kapcsoló
- Nyomógombok

Assembler konzol

USRT terminál

CPU állapot:

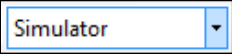
- PC, flag-ek, verem teteje, regiszterek tartalma
- Végrehajtott utasítások száma
- Elfogadott megszakításkérések száma

The image shows a screenshot of the MiniRISC IDE software interface. The main window displays assembly code for a program named 'mul4bitu.s'. The code includes comments in Hungarian and assembly instructions like 'LD 0x80', 'SW 0x81', and 'start: mov r0, SW'. The interface includes a menu bar (Build, Debug, Help), a toolbar with icons for simulation and execution, and a status bar at the bottom showing file length (3927), lines (58), current line (38), column (0), and selection (0|0).






Annotations and labels are placed over the interface:

- Fordítás és letöltés**: Points to the 'Build' and 'Load' icons in the toolbar.
- Végrehajtás vezérlése**: Points to the 'Run', 'Pause', and 'Stop' icons in the toolbar.
- Processor state**: A panel on the right showing PC (04), Stack (00), Registers (r0-r15), Instructions (4), and Interrupts (0).
- LEDs (0x80)**: A row of 8 checkboxes, all currently unchecked.
- Switches (0x81)**: A row of 8 checkboxes, with the 3rd and 6th checked.
- Buttons (0x84)**: A row of 8 checkboxes, all currently unchecked.
- Assembler console**: A panel at the bottom left showing the output of the assembler, including the message 'LOGSYS MiniRISC assembler completed with 0 error(s)'.
- USRT terminál**: A panel at the bottom center, currently empty.
- GPIO vezérlőpanel**: A panel at the bottom right, currently empty.




Bevezetés: MiniRISC IDE

- A  legördülő menüből kiválasztható, hogy a program
 - A szimulátorban fusson (ez mindig rendelkezésre áll)
 - A csatlakoztatott hardveren fusson (**LDCxxx** letöltőkábel)
 - Választás csak akkor lehetséges, ha program nem fut
- **Szimulátor**
 - Órajelciklus pontossággal szimulálja az FPGA áramkörben megvalósított processzoros rendszert
 - A program végrehajtása itt lassabb, mint a hardveren
 - Kb. 400000 utasítás/s → **kb. 1,2 MHz rendszerórajel frekvencia**
 - A MiniRISC mintarendszerben lévő perifériák nagy része a szimulátorban is elérhető
- **Futtatás hardveren**
 - Ha van FPGA kártya csatlakoztatva a számítógéphez

Bevezetés: MiniRISC IDE

- A program a **Compile** (, F5) gombbal fordítható le
 - A hibaüzenetek a konzolban jelennek meg
 - A hibás programsorok pirossal aláhúzásra kerülnek
- A lefordult program a **Download** (, F6) gombbal tölthető le
 - Hardveren történő futtatás esetén ha még nincs felkonfigurálva az FPGA, akkor először ez történik meg
 - A program futása megáll a 0x00 címen (reset vektor)
- A program végrehajtásának vezérlése
 - **Run** (, F7): a program végrehajtásának folytatása
 - **Break** (, F8): a program futásának felfüggesztése, a következő végrehajtandó utasítást sárga kiemelés jelzi
 - **Step** (, F10): az aktuális utasítás végrehajtása, a program futása megáll a következő utasításnál

Bevezetés: MiniRISC IDE

- A program végrehajtásának vezérlése
 - **Auto step** (): a **Step** parancs automatikus kiadása
 - A gyakoriság a Debug menüben állítható be
 - Az automatikus léptetés a **Break** paranccsal állítható le
 - **Reset** ( , F9): reset jel kiadása a processzoros rendszernek
 - **Stop** (): a program futtatásának befejezése
 - Ezután ismételt programletöltés szükséges
- Töréspont elhelyezése a szerkesztőben a margóra történő kattintással lehetséges
 - A töréspontot piros kör jelzi a margón

```
10      mov    r0, SW      ; A kapcsolók állapota r0-ba kerül.
11  ●     mov    LD, r0    ; A LED-ekre kiírjuk r0 értékét.
12      jmp    start      ; Ugrás a ciklus elejére.
```

- Töréspontra futáskor a program végrehajtása megáll
 - Lényegében egy **Break** parancs kiadása történik meg hardveresen

Bevezetés: MiniRISC IDE

- A processzor állapotának és az adatmemória tartalmának módosítása, illetve a perifériák vezérlése csak felfüggesztett programvégrehajtás (break állapot) esetén lehetséges
- **Processor state panel:** a processzor állapota
 - Programszámláló (PC) értéke
 - Flag-ek (Z, C N, V, IE, IF) értéke (IF csak olv.)
 - Verem tetejének értéke (csak olvasható)
 - Regiszterek értéke
 - A végrehajtott utasítások száma
 - Az elfogadott megszakításkérések száma
- **Alap perifériák vezérlőpanelje**
 - A LED-ek, a kapcsolók és a nyomógombok állapotát jeleníti meg
 - Biztosítja a perifériák regiszterszintű vezérlését

Processor state							
PC	IF	IE	V	N	C	Z	
01	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Stack							
00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0000
Registers							
r0	r1	r2	r3	r4	r5	r6	r7
EE	7D	00	00	00	00	00	00
00	00	00	00	00	00	00	00
r8	r9	r10	r11	r12	r13	r14	r15
Instructions							14286945
Interupts							0

LEDs (0x80)							
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	06
Switches (0x81)							
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EE
Buttons (0x84)							
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	00

A programfejlesztés lépései

- Gondoljuk át a problémát és készítsünk vázlatos elképzelést, hogy mit és hogyan szeretnénk megoldani
- Fogalmazzuk meg a megoldást a MiniRISC assembly utasítások szintjén és készítsük el a forráskódot
- A begépelte programot fordítsuk le a Compile (F5) paranccsal
- Az esetleges hibákat javítsuk ki
- A hibátlanul lefordult programból generált SVF fájl letölthető a Download (F6) paranccsal
- Ellenőrizzük a program működését debug módban a MiniRISC IDE szolgáltatásainak segítségével

Bevezető feladatok

- A MiniRISC processzor programozásának gyakorlásához a félév első laborjainak egyszerű Verilog HDL alapú bevezető feladatait ismételjük meg gépi szintű programozással
- A feladatok így ismertek, csak egy kicsit más szemlélet szükséges a realizációhoz
- Sok esetben az új megközelítés sokkal egyszerűbb, sok esetben itt ütközünk nehézségekbe
- Nézzük át a MiniRISC processzor utasításkészletét!
 - Elérhető a MiniRISC IDE Help menüjében
- Sok sikert a gépi szintű programozáshoz!

1. feladat: assembly „Hello World!”

- **Folyamatosan jelenítsük meg a kapcsolók állapotát a LED-eken**
 - Hogy tudjuk elérni a perifériákat?
 - Milyen utasítások szükségesek ehhez?
- **Rajzoljuk fel a program folyamatábráját**
- **Készítsük el a programot a MiniRISC IDE-ben**
- **Próbáljuk ki a program működését az FPGA kártyán**

2. feladat: kettes komplementes képzés

- **Folyamatosan jelenítsük meg a kapcsolón beállított érték kettes komplementesét a LED-eken**
 - Hogy tudunk kettes komplementet képezni?
 - Milyen utasítások szükségesek ehhez?
- **Rajzoljuk fel a program folyamatábráját**
- **Egészítsük ki az előző feladatban elkészített kódot**
- **Próbáljuk ki a program működését az FPGA kártyán**

3. feladat: összeadó

- **Folyamatosan jelenítsük meg a kapcsolón beállított érték alsó és felső 4 bitjének összegét a LED-eken**
 - $SW[3:0] + SW[7:4] \rightarrow LD$
 - Az operandusok előjel nélküli értékek
 - Hogy kell egymáshoz igazítani az operandusokat?
 - Milyen utasítások szükségesek ehhez?
- **Készítsük el a programot a MiniRISC IDE-ben**
- **Próbáljuk ki a program működését az FPGA kártyán**
 - Mit tapasztalunk, ha az operandusokat kettes komplementes kódolásúnak értelmezzük?
 - Milyen módosítás szükséges, hogy az eredmény 8 biten előjel helyesen jelenjen meg a LED-eken?

4. feladat: szorzó

- Folyamatosan jelenítsük meg a kapcsolón beállított érték alsó és felső 4 bitjének szorzatát a LED-eken
 - $SW[3:0] \cdot SW[7:4] \rightarrow LD$
 - Az operandusok előjel nélküli értékek
- A MiniRISC processzornak nincs szorzás utasítása
 - Vegyük alapul az írásbeli szorzást
 - Elemezzük az algoritmus működését

$$\begin{array}{r} \underline{1001} \cdot 0110 \leftarrow \text{szorzandó, szorzó} \\ 0000 = (1001 \ \& \ 0000) \lll 0 \leftarrow \text{1. részszorzat} \\ 10010 = (1001 \ \& \ 1111) \lll 1 \leftarrow \text{2. részszorzat} \\ 100100 = (1001 \ \& \ 1111) \lll 2 \leftarrow \text{3. részszorzat} \\ + \underline{0000000} = (1001 \ \& \ 0000) \lll 3 \leftarrow \text{4. részszorzat} \\ \hline 00110110 \leftarrow \text{szorzat} \end{array}$$

4. feladat: szorzó

- Rajzoljuk fel a program folyamatábráját
- Készítsük el a programot a MiniRISC IDE-ben
- Próbáljuk ki a program működését az FPGA kártyán

