



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

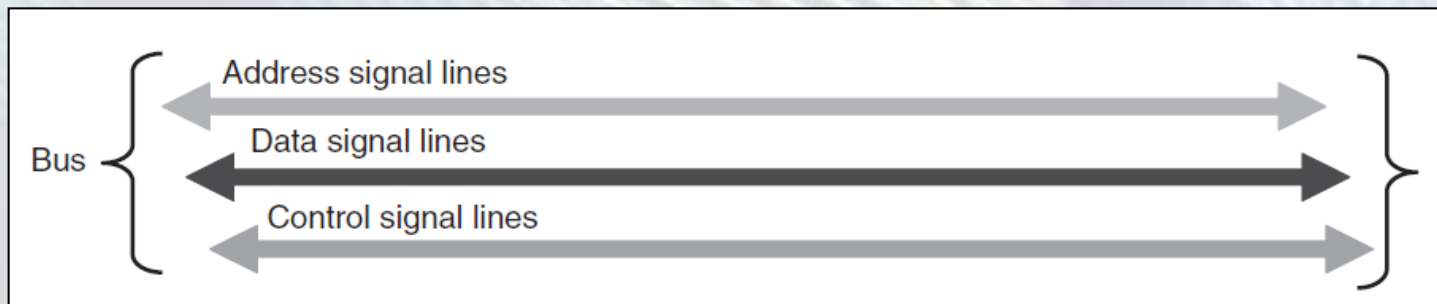
Digitális technika

VIMIAA01

Fehér Béla
BME MIT

A mikroprocesszoros busz

- **A busz részei:**
 - **Címbusz ADDR[n:0]**
 - **Adatbusz DATA[m:0],**
belső busznál külön **D_IN[m:0], D_OUT[m:0]**
 - **Vezérlő busz (sok egyedi jel összessége):**
 - **Rendszerjelek: CLK, RST,**
 - **Arbitrációs jelek: BUSREQ, BUSACK,**
 - **Írányvezérlő jelek: READ, WRITE,**
 - **Átvitelvezérlő jelek: FRAME, TS, TACK, AS, DS,**
 - **Megszakítás vezérlő jelek: IRQi, IACK,**

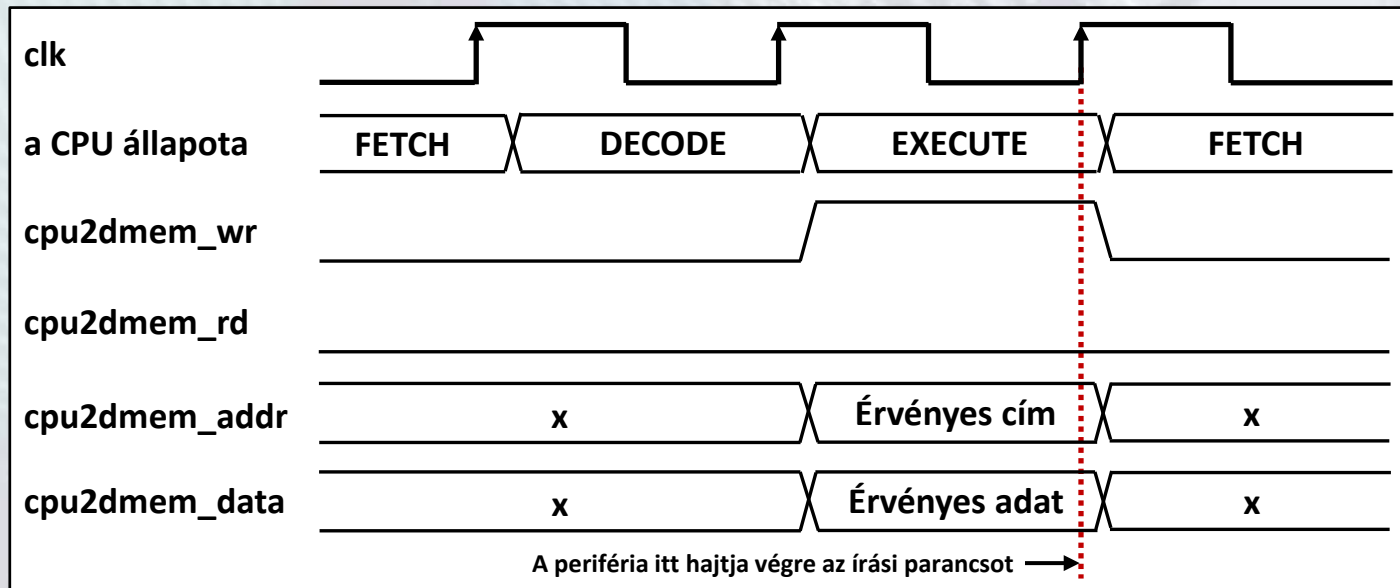


MiniRISC processzor – Interfészek

(Adatmemória interfész – Írási buszciklus)

Az adatmemória interfész írási (write) ciklusa

- Az írási ciklust a végrehajtási (execute) fázisban 1 órajelciklus ideig aktív **cpu2dmem_wr** jel jelzi
- Az írási ciklus alatt a **cpu2dmem_addr** címbuszson a cím stabil
- Az írási ciklus alatt a **cpu2dmem_data** írási adatbuszon az adat stabil, melyet a kiválasztott periféria az órajel következő felfutó élére mintavételez



MiniRISC processzor, 2014.11.11.

(v1.3)

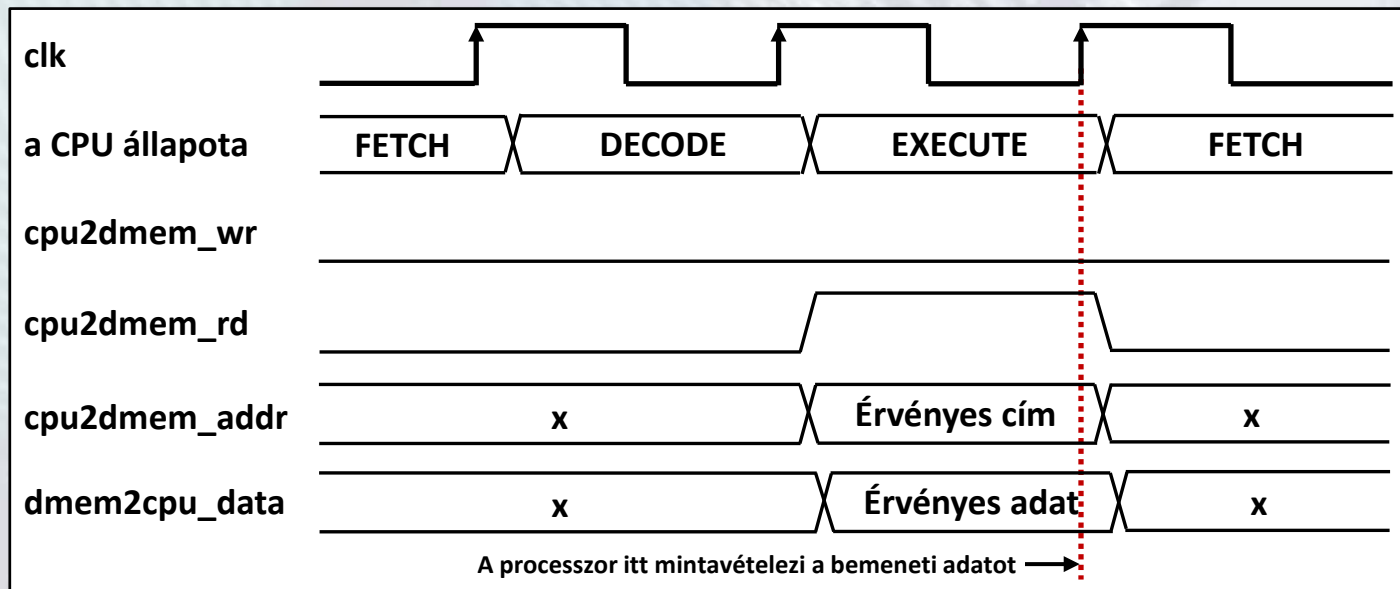
FPGA labor

MiniRISC processzor – Interfészek

(Adatmemória interfész – Olvasási buszciklus)

Az adatmemória interfész olvasási (read) ciklusa

- Az olvasási ciklust a végrehajtási (execute) fázisban 1 órajelciklus ideig aktív **cpu2dmem_rd** jel jelzi
- Az olvasási ciklus alatt a **cpu2dmem_addr** címbuszon a cím stabil
- Az olvasási ciklus alatt a kiválasztott periféria a **dmem2cpu_data** olvasási adatbuszra kapuzza az adatot, a többi periféria ezalatt inaktív nullával hajtja meg az adatkimenetét

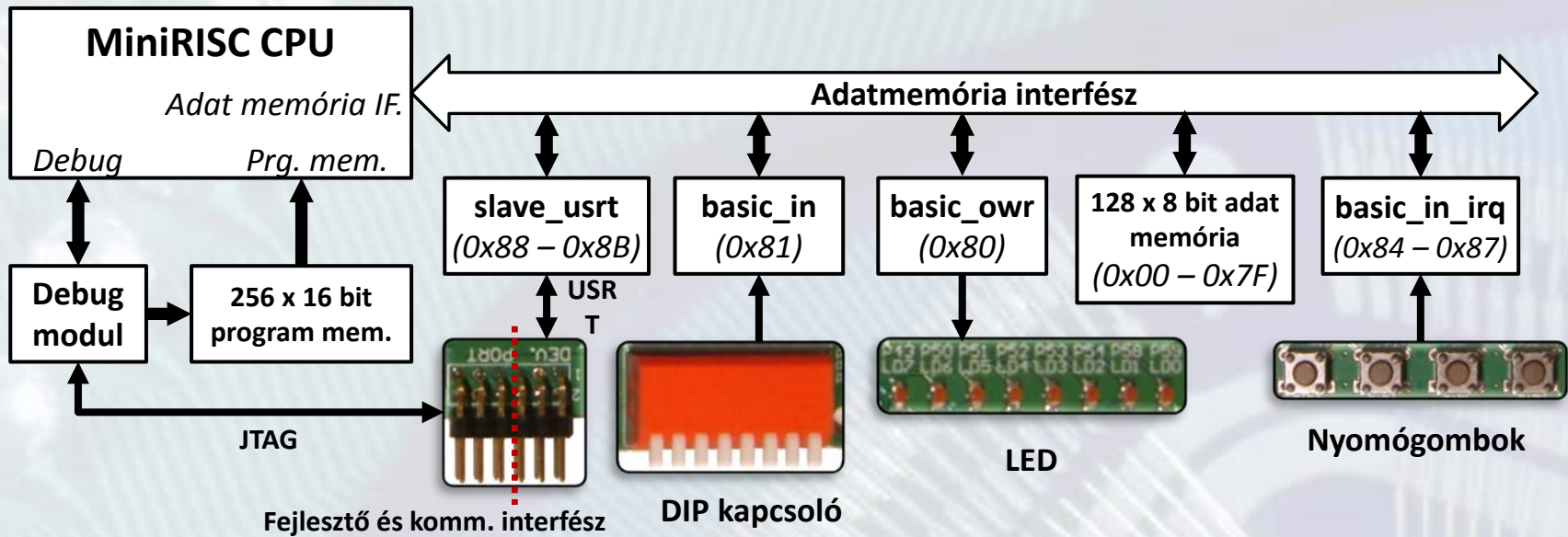


MiniRISC processzor, 2014.11.11.

(v1.3)

FPGA labor

Az egyszerűsített MiniRISC rendszer



Címtartomány	Méret	Periféria	Funkció
0x00 – 0x7F	128 byte	adatmemória	128 x 8 bit memória
0x80	1 byte	basic_owr	LED periféria
0x81	1 byte	basic_in	DIP kapcsoló periféria
0x84 – 0x87	4 byte	basic_in_irq	Nyomógomb periféria
0x88 – 0x8B	4 byte	slave_usrt	Soros USRT kommunikáció
0x90 – 0xFF	112 byte		További perifériák

MiniRISC processzor – Címterkép

- A MiniRISC processzor teljes címtartománya: 0x00 – 0xFF
 - 0x00 – 0x7F RAM memória
 - 0x80 – 0xFF Perifériák
- Szabályok:
 - Minden eszköz címezése két részből áll: **BÁZIS_CÍM** + **REGISZTER_CÍM**
 - A **BÁZIS_CÍM** mindig egy 2^N méretű címtartomány elejére mutat, azonosítja az eszközt! (Eszköz kiválasztás!)
 - Nem kell címaritmetika a regisztercím meghatározásához!

Címtartomány	A7	A6	A5	A4	A3	A2	A1	A0	Eszköz
0x00	0	0	0	0	0	0	0	0	RAM
.....	x	x	x	x	x	x	x	x	
0x7F	1	1	1	1	1	1	1	1	
0x80	1	0	0	0	0	0	0	0	LD
0x81	1	0	0	0	0	0	0	1	SW
0x82	1	0	0	0	0	0	1	0	TIMER
0x83								1	
0x84	1	0	0	0	0	1	0	0	BT_sel
0x85							0	1	
0x86							1	0	
0x87								1	
0x88	1	0	0	0	1	0	0	0	USRT
0x89							0	1	
0x8A							1	0	
0x8B							1	1	
0x8C	1	0	0	0	1	1	0	0	DMA
0x8D							0	1	
0x8E							1	0	
0x8F							1	1	
0x90	1	0	0	1	0	0	0	0	DISP
.....					0	x	x	x	
0x9B					1	0	0	0	
0xA0	1	0	1	0	0	0	0	0	GPIO_A
0xA1							0	1	
0xA2							1	0	
0xA4	1	0	1	0	0	1	0	0	GPIO_C
0xA5							0	1	
0xA6							1	0	
0xA8	1	0	1	0	1	0	0	0	GPIO_B
0xA9							0	1	
0xAA							1	0	
0xAC	1	0	1	0	1	1	0	0	GPIO_D
0xAD							0	1	
0xAE							1	0	
0xB0	1	0	1	1	0	0	0	0	VGA
.....						0	x	x	
0xB5						1	0	1	
0xBB	1	0	1	1	0	0	0	0	PS/2
0xB9								1	
0xBA									Tartalék
.....									
0xFF									

MiniRISC processzor – Címterkép

- A memória/periféria címdekódolás ezért két lépésből áll:
 - **BÁZIS_CÍM** felismerése komparátorral, vagy közvetlen logikával. Mindig csak a **BÁZIS_CÍM** aktív címvonalait dekódoljuk, tehát 7-6-5-4 bitet
 - A **REGISZTER_CÍM** felismerése és az egyedi vezérlőjelek (WR_x, RD_x) előállításuk dekóderrel (vagy közvetlen) logikával történik. Az engedélyezést a **BÁZIS_CÍM** kiválasztó jele (psel) és a busz WR és RD vezérlőjele kapuzza, azaz érvényesíti!

Címteremtartomány	A7	A6	A5	A4	A3	A2	A1	A0	Eszköz
0x00	0	0	0	0	0	0	0	0	RAM
.....	x	x	x	x	x	x	x	x	
0x7F	1	1	1	1	1	1	1	1	
0x80	1	0	0	0	0	0	0	0	LD
0x81	1	0	0	0	0	0	0	1	SW
0x82	1	0	0	0	0	0	1	0	TIMER
0x83								1	
0x84	1	0	0	0	0	1	0	0	BT_sel
0x85							0	1	
0x86							1	0	
0x87							1	1	
0x88	1	0	0	0	1	0	0	0	USRT
0x89							0	1	
0x8A							1	0	
0x8B							1	1	
0x8C	1	0	0	0	1	1	0	0	DMA
0x8D							0	1	
0x8E							1	0	
0x8F							1	1	
0x90	1	0	0	1	0	0	0	0	DISP
.....					0	x	x	x	
0x9B					1	0	0	0	
0xA0	1	0	1	0	0	0	0	0	GPIO_A
0xA1							0	1	
0xA2							1	0	
0xA4	1	0	1	0	0	1	0	0	GPIO_C
0xA5							0	1	
0xA6							1	0	
0xA8	1	0	1	0	1	0	0	0	GPIO_B
0xA9							0	1	
0xAA							1	0	
0xAC	1	0	1	0	1	1	0	0	GPIO_D
0xAD							0	1	
0xAE							1	0	
0xB0	1	0	1	1	0	0	0	0	VGA
.....						0	x	x	
0xB5						1	0	1	
0xBB	1	0	1	1	0	0	0	0	PS/2
0xB9								1	
0xBA									Tartalék
.....									
0xFF									

MiniRISC processzor – Memóriaillesztés

(Memória címtartomány dekódolása)

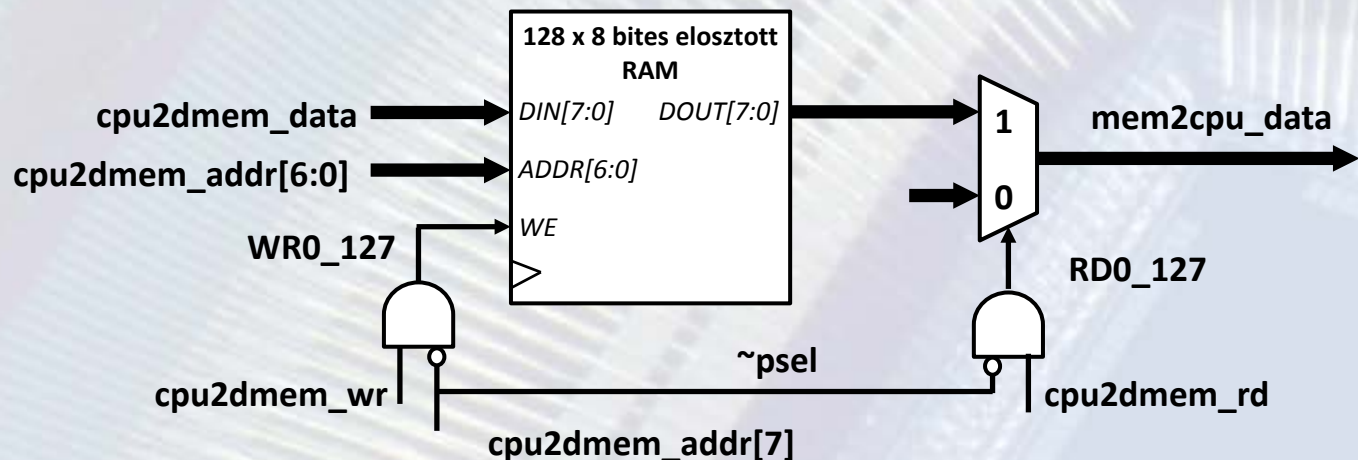
Feladat: 128 x 8 bites memória illesztése a processzoros rendszerhez

A memóriának 7 címbitje van ($2^7 = 128$)

- Kijelölt címtartomány: **00000000 (0x00) – 01111111 (0x7F)**

A címdekóder logika megvalósítása

- A címbusz alsó 7 bitje közvetlenül kapcsolódik a memóriához
- A legfelső címbitet (MSb) használjuk a báziscím dekódoláshoz
 - Ha ez 0, akkor a memória van kiválasztva, ha nem, akkor nincs memória művelet



MiniRISC processzor – Memóriaillesztés

(Megvalósítás Verilog nyelven)

Feladat: 128 x 8 bites memória illesztése a processzoros rendszerhez

```
wire      memsel    = ~cpu2dmem_addr[7];      //A memória kiválasztó jele.
wire [6:0] mem_addr = cpu2dmem_addr[6:0];     //A memória címbusza.
wire      mem_wr    = memsel & cpu2dmem_wr;   //A memória írás engedélyező jele.
wire      mem_rd    = memsel & cpu2dmem_rd;   //A memória olvasás engedélyező jele.

//128 x 8 bites elosztott RAM deklarálása.
(* ram_style = "distributed" *)
reg [7:0] mem [127:0]

//Az elosztott RAM írási portja.
//Az elosztott RAM írása szinkron művelet.
always @(posedge clk)
    if (mem_wr)
        mem[mem_addr] <= cpu2dmem_data;

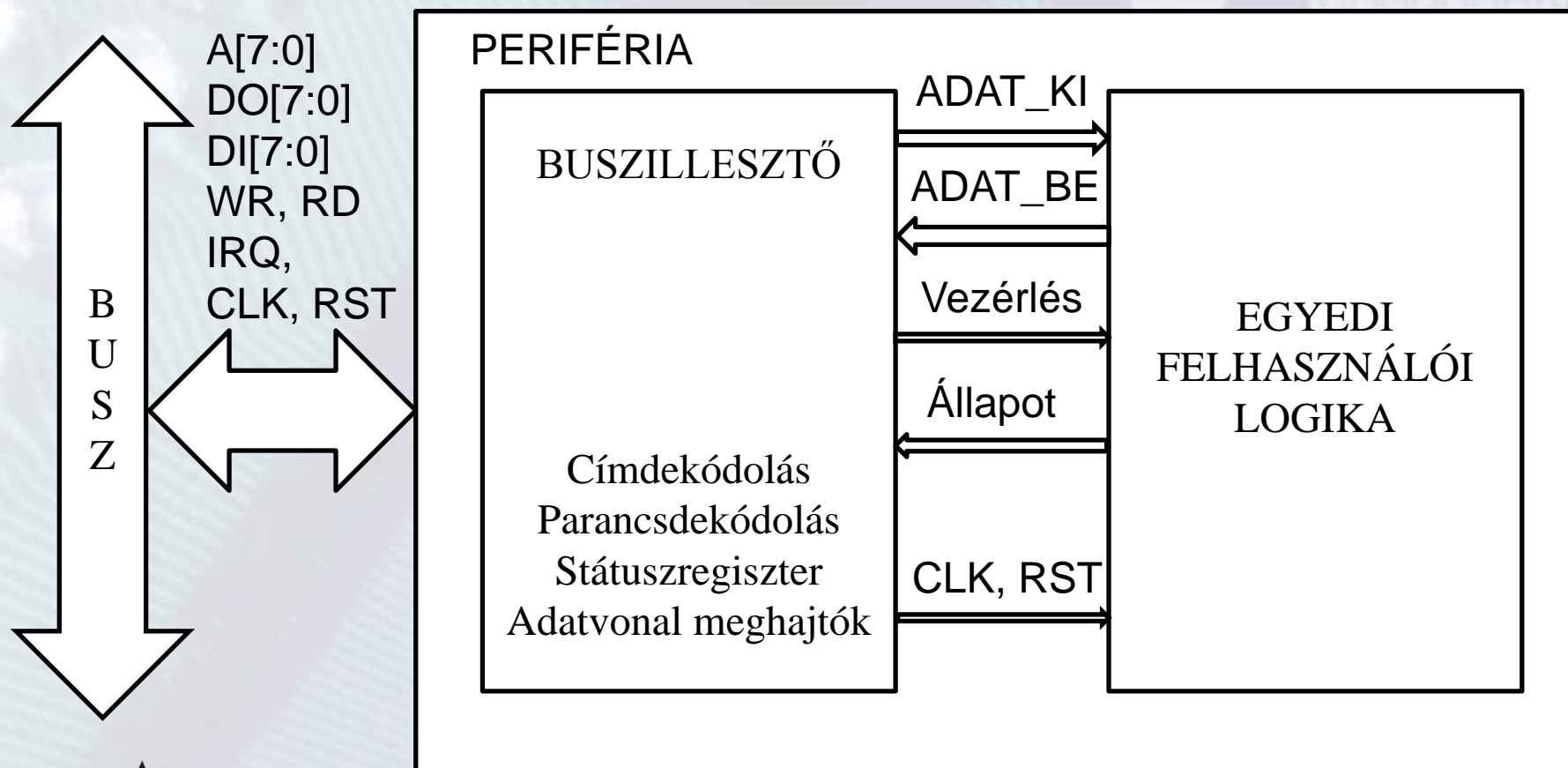
//Az elosztott RAM olvasási portja és az olvasási adatbusz meghajtása
//(ezt kell VAGY kapuval rákötni a processzor olvasási adatbuszára).
//Az elosztott RAM olvasása aszinkron művelet.
wire [7:0] mem2cpu_data = (mem_rd) ? mem[mem_addr] : 8'd0;
```

MiniRISC processzor – Perifériaillesztés

- **A perifériaillesztési feladat lépései**
- **A periféria típusa alapján az igények felmérése**
 - Regiszter szám igény és használati mód meghatározása (írható, olvasható)
 - Adat be-/ki, parancs, státusz, üzemmód, stb. regiszterek
 - Esetleg FIFO vagy kisebb memória blokk
- **A báziscím kijelölése, a címtartomány használatának megtervezése**
 - Mindig teljes 2^N méretű címtartomány jelölünk ki, azaz 1-2-4-8-16 méretű regisztercím tartományt foglalunk le
 - Az egyes regiszter címek így a (báziscímhez képest relatív) 0,1,2,3..címeken lesznek!

MiniRISC processzor – Perifériaillesztés

- A perifériaillesztés általános tulajdonságai
 - Minden egység ezeket a részleteket tartalmazza



MiniRISC processzor – Perifériaillesztés

- **A perifériaillesztési általános tulajdonságai**
- **A BUSZ tulajdonságai rendszerfüggőek**
 - A jelek száma, használati módja ezért egyedi lehet
 - Mi a **MiniRISC** egyszerű belső buszával tervezünk
 - Kommunikáció: 8 CÍM, 8 ADAT_KI, 8 ADAT_BE vonal,
 - Vezérlőjelek: CLK, RST, WR, RD, IRQ,
 - Busz master egységeknél: REQ, GRANT,
- **A buszillesztő logika felépítése tehát nem annyira a periféria, mint inkább a busz tulajdonságai által meghatározott**



MiniRISC processzor – Perifériaillesztés

- **A perifériaillesztési általános tulajdonságai**
- **A buszillesztő logika és a felhasználói logika interfésze**
 - 8 ADAT_KI, 8 ADAT_BE, az adatátviteli vonalak
 - Vezérlő jelek a felhasználói logika működtetésére, a regiszterek beírására, kiválasztására
 - Állapotjelek, melyek a periféria működéséről tájékoztatnak
- **Az egyes részáramköröket a továbbiakban elemezzük**

MiniRISC processzor – Perifériaillesztés

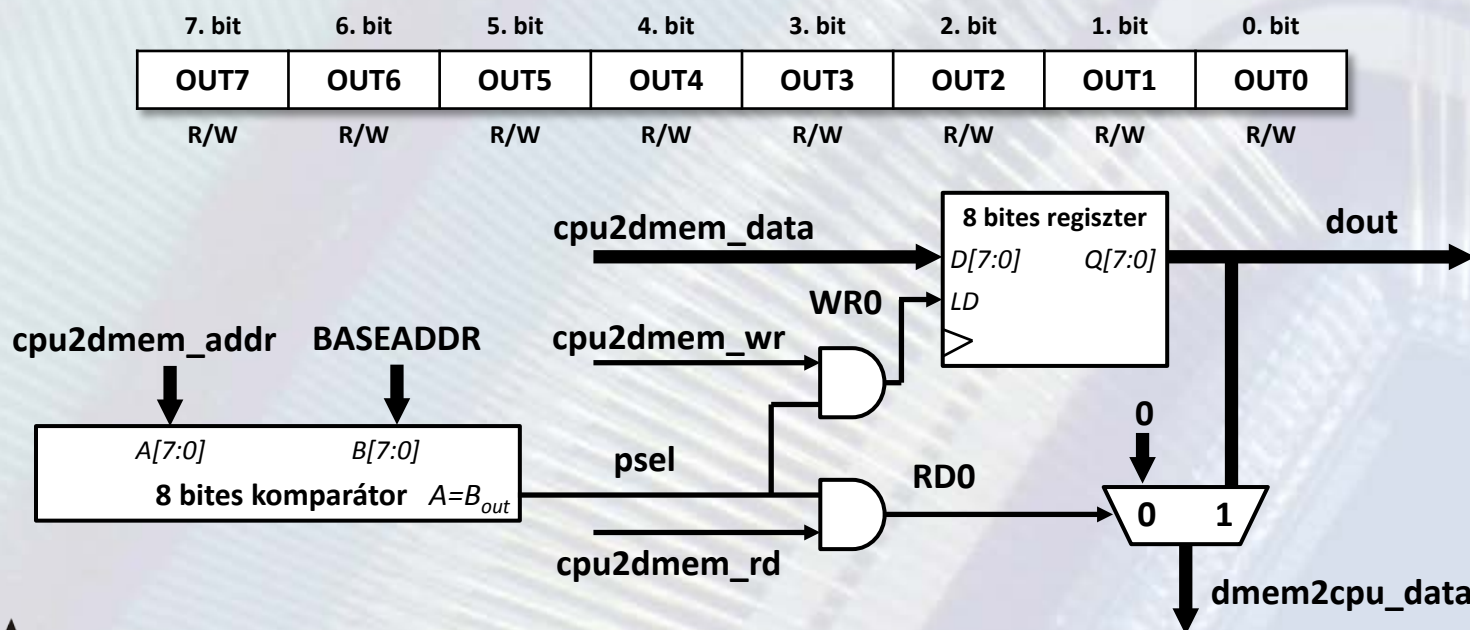
- **A címdekódolás kialakítása (BÁZIS_CÍM felismerése)**
 - $psel = ((cpu2dmem_addr \gg N) == (BASEADDR \gg N))$
 - A címtartomány mérete 2^N byte (1,2,4,8,16..)
- **Parancsdekódolás:**
 - Írás engedélyező jelek
 - $xxx_wr = psel \& cpu2dmem_wr \& (cpu2dmem_addr[N-1:0] == ADDR)$
 - Az egyes perifériaregiszterek beíró/betöltő jelei (WR0, WR1, WR2...)
 - Olvasás engedélyező jelek
 - $xxx_rd = psel \& cpu2dmem_rd \& (cpu2dmem_addr[N-1:0] == ADDR)$
 - Az egyes perifériaregiszterek olvasás engedélyező jelei (RD0, RD1, RD2...)
 - A jeleket a kimeneti elosztott MUX (AND-OR hálózat) vezérlésére, engedélyezésére használjuk: egy időben csak egy kimeneten lehet érvényes adat, a többi periféria kimenetének értéke inaktív nulla lesz
 - Használni kell még, ha az olvasás állapotváltozást okoz (pl. FIFO)

MiniRISC processzor – Perifériaillesztés

(1. példa – Specifikáció)

Feladat: 8 darab LED illesztése a processzoros rendszerhez.
Kimeneti periféria, de az aktuális állapot legyen visszaolvasható

- Egyszerű, egy 8 bites írható és olvasható regiszter szükséges
- Adatregiszter: BASEADDR + 0x00, 8 bites, írható/olvasható
 - Az OUT_i bit hajtja meg az i -edik LED-et



MiniRISC processzor – Perifériaillesztés

(1. példa – Megvalósítás Verilog nyelven)

Feladat: 8 darab LED illesztése a processzoros rendszerhez

```
module basic_owr #(
    //A periféria báziscíme.
    parameter BASEADDR = 8'hff
) (
    //Órajel és reset.
    input wire      clk,
    input wire      rst,

    //Adatmemória interfész.
    input wire [7:0] cpu2dmem_addr,
    input wire      cpu2dmem_wr,
    input wire      cpu2dmem_rd,
    input wire [7:0] cpu2dmem_data,
    output reg [7:0] dmem2cpu_data,

    //Kimenő adat.
    output reg [7:0] dout
);

//A periféria kiválasztó jele.
wire psel = (cpu2dmem_addr == BASEADDR);
```

```
//Az adatreg. írás engedélyező jele.
wire dreg_wr = psel & cpu2dmem_wr;

//Az adatreg. olvasás engedélyező jele.
wire dreg_rd = psel & cpu2dmem_rd;

//Adatregiszter.
always @(posedge clk)
    if (rst)
        dout <= 8'd0;
    else
        if (dreg_wr)
            dout <= cpu2dmem_data;

//Az olvasási adatbusz meghajtása.
always @(*)
    if (dreg_rd)
        dmem2cpu_data <= dout;
    else
        dmem2cpu_data <= 8'd0;

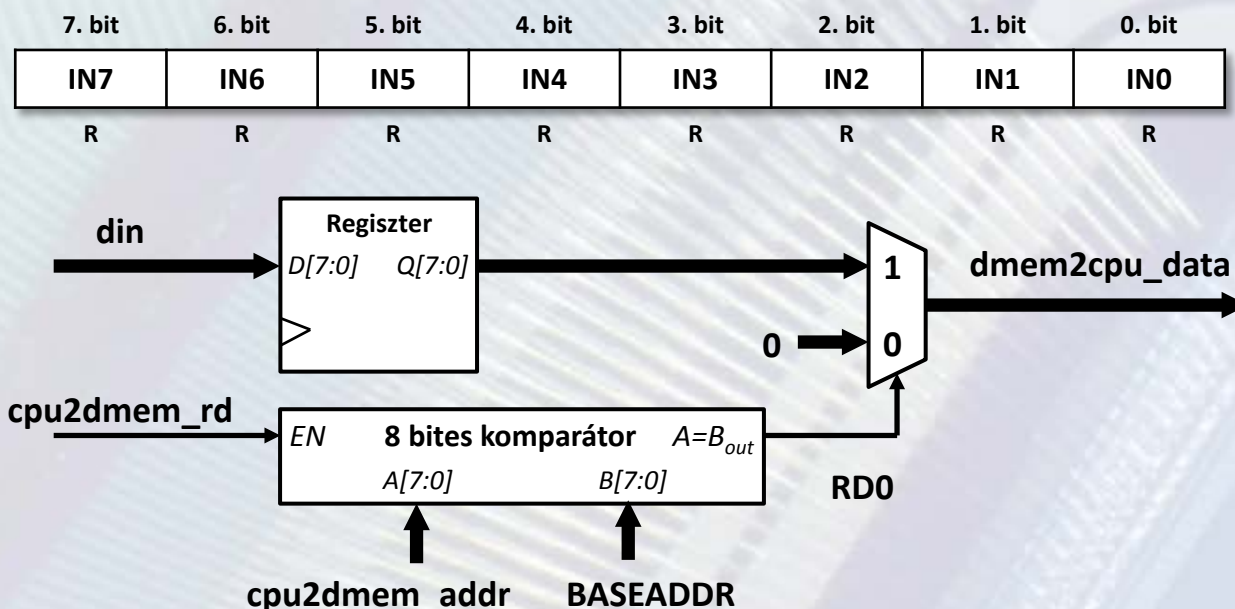
endmodule
```


MiniRISC processzor – Perifériaillesztés

(2. példa – Specifikáció)

Feladat: 8 darab kapcsoló illesztése a processzoros rendszerhez

- Egyszerű, egy 8 bites csak olvasható regiszter szükséges, amely folyamatosan mintavételezi a 8 kapcsoló állapotát
- Adatregiszter: **BASEADDR + 0x00**, 8 bites, csak olvasható
 - Az IN_i bit az i -edik kapcsolón beállított értéket veszi fel



MiniRISC processzor – Perifériaillesztés

(2. példa – Megvalósítás Verilog nyelven)

Feladat: 8 darab kapcsoló illesztése a processzoros rendszerhez

```
module basic_in #(
    //A periféria báziscíme.
    parameter BASEADDR = 8'hff
) (
    //Órajel és reset.
    input wire      clk,
    input wire      rst,

    //Adatmemória interfész.
    input wire [7:0] cpu2dmem_addr,
    input wire      cpu2dmem_rd,
    output reg  [7:0] dmem2cpu_data,

    //Bejövő adat.
    input wire [7:0] din
);

//A periféria kiválasztó jele.
wire psel = (cpu2dmem_addr == BASEADDR);
```

```
//Az adatreg. olvasás engedélyező jele.
wire in_reg_rd = psel & cpu2dmem_rd;

//Adatregiszter.
reg [7:0] in_reg;

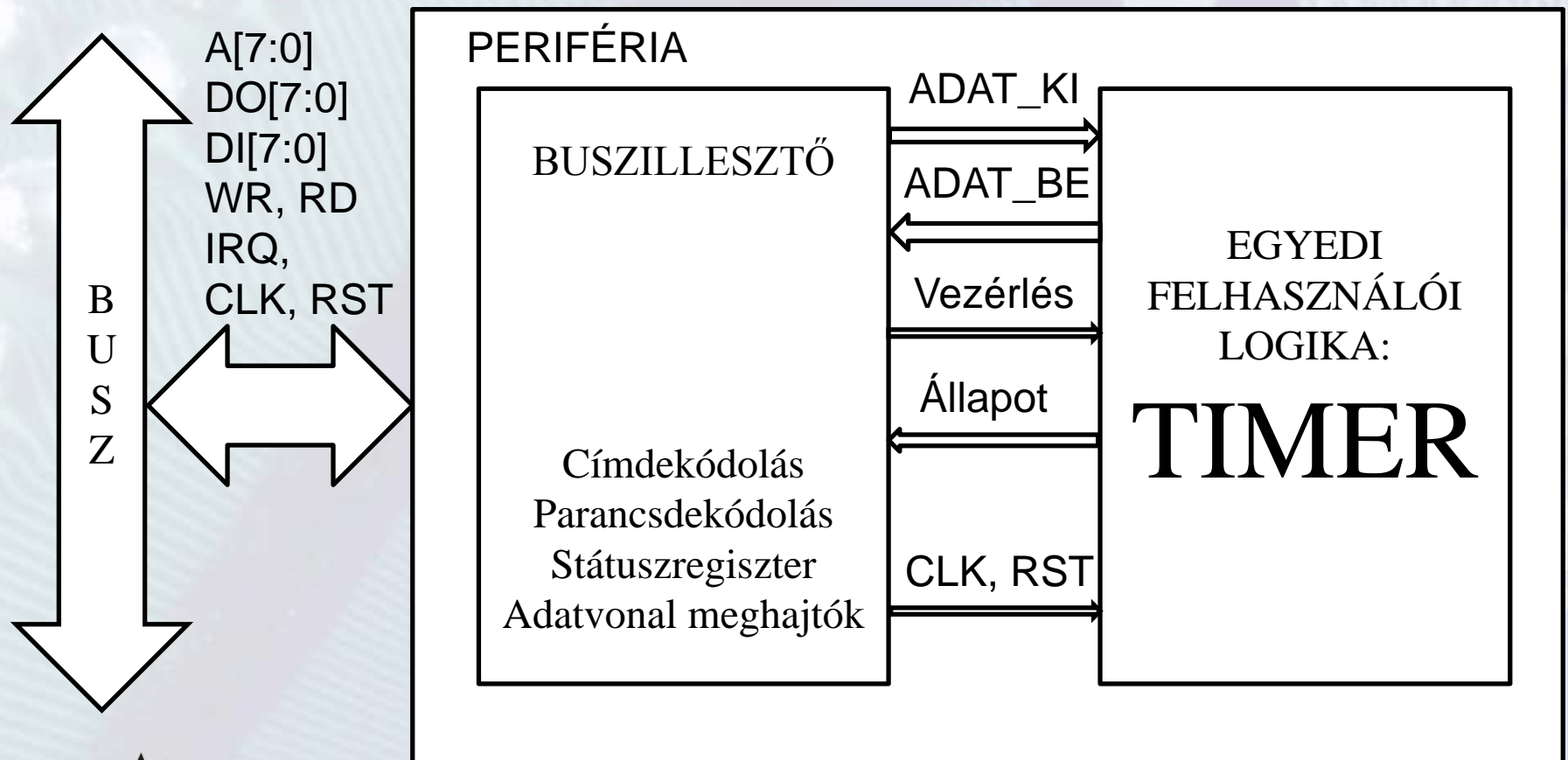
always @(posedge clk)
    if (rst)
        in_reg <= 8'd0;
    else
        in_reg <= din;

//Az olvasási adatbusz meghajtása.
always @(*)
    if (in_reg_rd)
        dmem2cpu_data <= in_reg;
    else
        dmem2cpu_data <= 8'd0;

endmodule
```

MiniRISC processzor – Perifériaillesztés

- **Összetett perifériatervezési példa: Időzítő (TIMER) periféria a processzoros időzítések támogatására**



MiniRISC processzor – Perifériaillesztés

- **Probléma: A processzorral programozottan tudunk időzítést beállítani, de akkor a processzor nem tud hasznos munkát végezni**
- **A idő múlását számolható utasításciklusok alatt végezhetne hasznos feladatokat is a processzor**
- **Ehhez kell egy párhuzamosan működtethető időmérő egység, ez az időzítő TIMER periféria**
- **Az időzítő perifériák a leggyakoribb perifériák a processzoros rendszerekben**
- **Gyakran több egységet is tartalmaznak, kiegészítve speciális felhasználási lehetőségeket biztosító eszközökkel (komparátorok, külső órajel lehetőség,..)**



MiniRISC processzor – Perifériaillesztés

- **A TIMER periféria tervezett paramétere:**
 - 8 bites időzítő számláló TR, amivel 1 - 256 ütemezési ciklus hosszú időzítés állítható be. A számláló egy tölthető, engedélyezhető, lefelé számláló, ami a lejáratkor, azaz a nulla érték elérésekor jelez. (Időzítés letelt, TIME_OUT).
 - Az időzítő rendelkezik egy előosztóval PS, ami az ütemezési periódust beállítja. A PS a rendszer órajelet programozhatóan leosztja 1-16-64-256-1024-4096-16384-65536 értékkel, azaz a PS érték 2^0 -tól 2^{16} -ig skálázható
 - Relatív időzítés pontosság mindig 0,5% (1/256)!
 - A beállítható teljes időzítési tartomány

$$T = (TR+1) * PS * T_{clk} \quad (62,5ns - 1,048s)$$

MiniRISC processzor – Perifériaillesztés

- **A TIMER periféria tervezett üzemmódjai:**
 - Egyszeres lefutási mód, az időzítés végén jelzéssel és leállással
 - Periodikus újraindulásos mód, az időzítések végén jelzéssel és automatikus újraindítással.
 - Az időzítés végét jelző TIME_OUT jelzés állapota lekérdezéssel kiolvasható, vagy ha engedélyezett, akkor lehetőség van megszakításkérésre is.
 - Az időzítő regiszter TR aktuális értéke időzítés közben bármikor visszaolvasható

MiniRISC processzor – Perifériaillesztés

- Először megtervezzük a programozói felületet
- 4 interfész regiszter, ebből kettő-kettő azonos címen
- Megtehető, mert egyik WR, másik RD
- A Báziscím 0x82, a regisztercímelek 0x82 és 0x83.

Időzítő	TR	kezdőállapot regiszter	0x82	WR	0xFF	Az időzítő számláló kezdeti értéke					
	TM	számláló regiszter	0x82	RD	0xFF	Az időzítő számláló aktuális értéke					
	TC	parancs regiszter	0x83	WR	0x00	TIE	TPS[2:0] - előosztás	-	-	TREP	TEN
	TS	státusz regiszter	0x83	RD	0x00	TIT	TPS[2:0] - előosztás	0	TOUT	TREP	TEN

• Parancs és státuszbitok jelentése

- TEN Engedélyezés (1 engedélyez, 0 leállít)
- TREP Ismétléses mód (1 ismétlés, 0 egyszeri futás)
- TOUT Időzítés lejárt (státusz olvasása törli)
- TPS[2:0] Előosztás mértéke, 8 érték, 1-65536
- TIE Megszakítás engedélyezés

MiniRISC processzor – Perifériaillesztés

- **A szükséges modulok:**
- **Egy 16 bites előosztó:**
 - tölthető lefelé számláló végérték jelzéssel
- **Egy 8 bites számláló:**
 - tölthető lefelé számláló végérték jelzéssel
- **Egy 8 bites regiszter:**
 - Tölthető, törölhető, az időzítés érték tárolásához
- **Parancs regiszter bitek**
 - Az üzemmód vezérlőbitek tárolásához
- **Státuszregiszter bitek**
 - Részben a parancsregiszter bitjei, új bit a TOUT, ami az időzítés lejártának állapotát jelzi
- **A részletek megtekinthetők a MiniRISC Verilog projekt periféria forrásfájlok között**

MiniRISC processzor – Perifériaillesztés

- A TIMER periféria használata:
- A kívánt időzítés felprogramozása
 - Üzem mód, előosztó, nincs megszakítás

```
                                ; Az időzítő beállítása kb. 0,5 s periódusidőre (2 Hz).  
                                ; 16*10^6 Hz / 65536 / 122 = 2 Hz  
mov    r2, #121                ; A számláló regiszter értéke 121 (0x79).  
mov    TM, r2                  ; Beírás a TM időzítő regiszterbe.  
mov    r2, #0x73               ; Az időzítő konfigurálása: 65536-os előosztás,  
mov    TC, r2                  ; ismétléses mód, időzítő engedélyezve (0111_0011).  
mov    r2, TS                  ; Az időzítő esetleges TOUT jelzésének törlése.
```

- A TOUT státuszbit tesztelése

```
tm_loop:                        ; Várakozunk, amíg az időzítő nem jelez.  
  mov    r2, TS                ; Beolvassuk az időzítő státusz regisztert  
  tst    r2, #0x04             ; a TOUT bit vizsgálatához.  
  jz     tm_loop               ; Várakozás, ha a TOUT bit még mindig 0.
```

- Megjegyzés: A tm_loop-ban bármi „hasznosat” is csinálhatnánk, a processzor ráér, van szabad utasítás végrehajtási ideje
- A következő előadáson megismerjük a megszakításos módot!

Digitális technika

11. EA vége