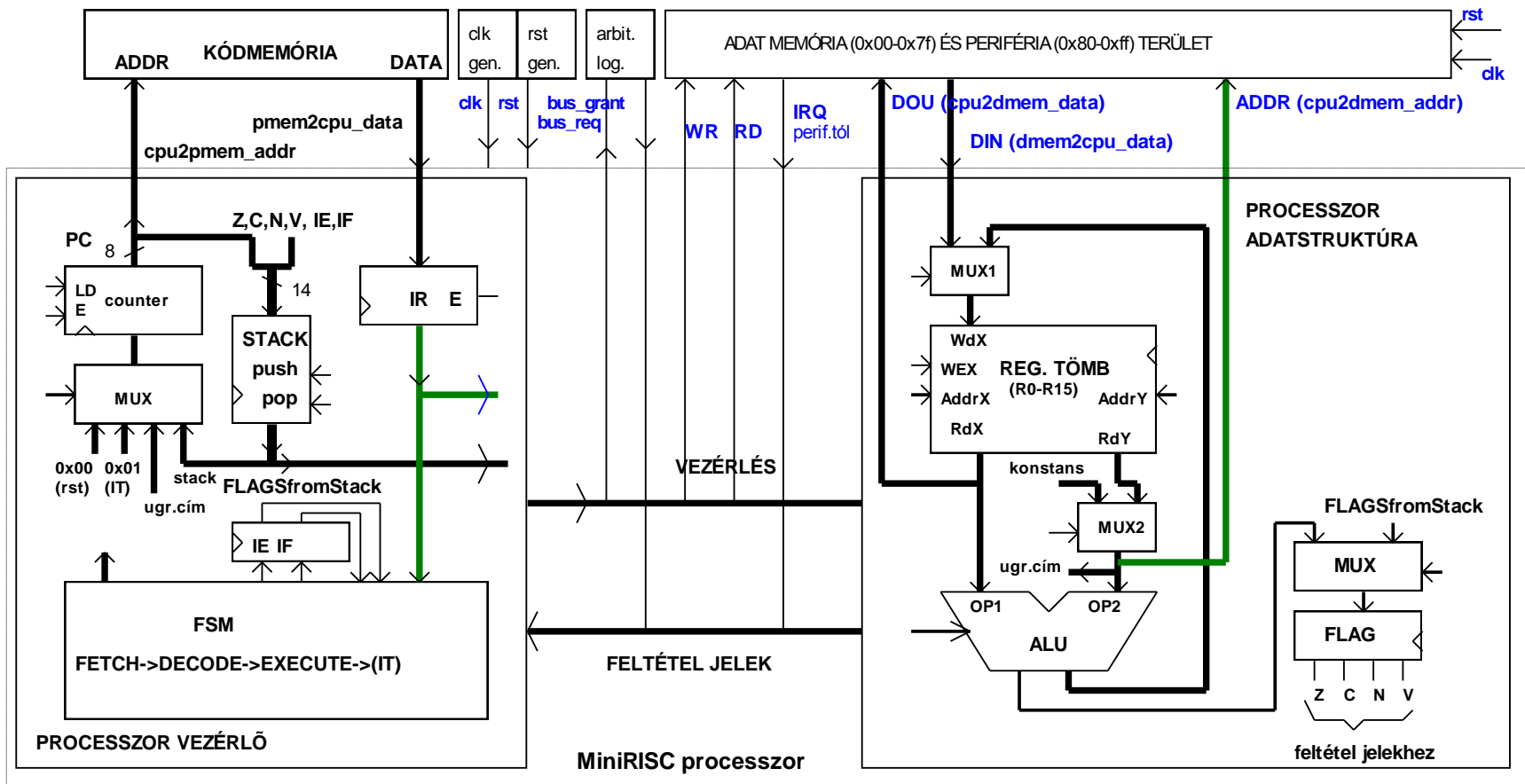


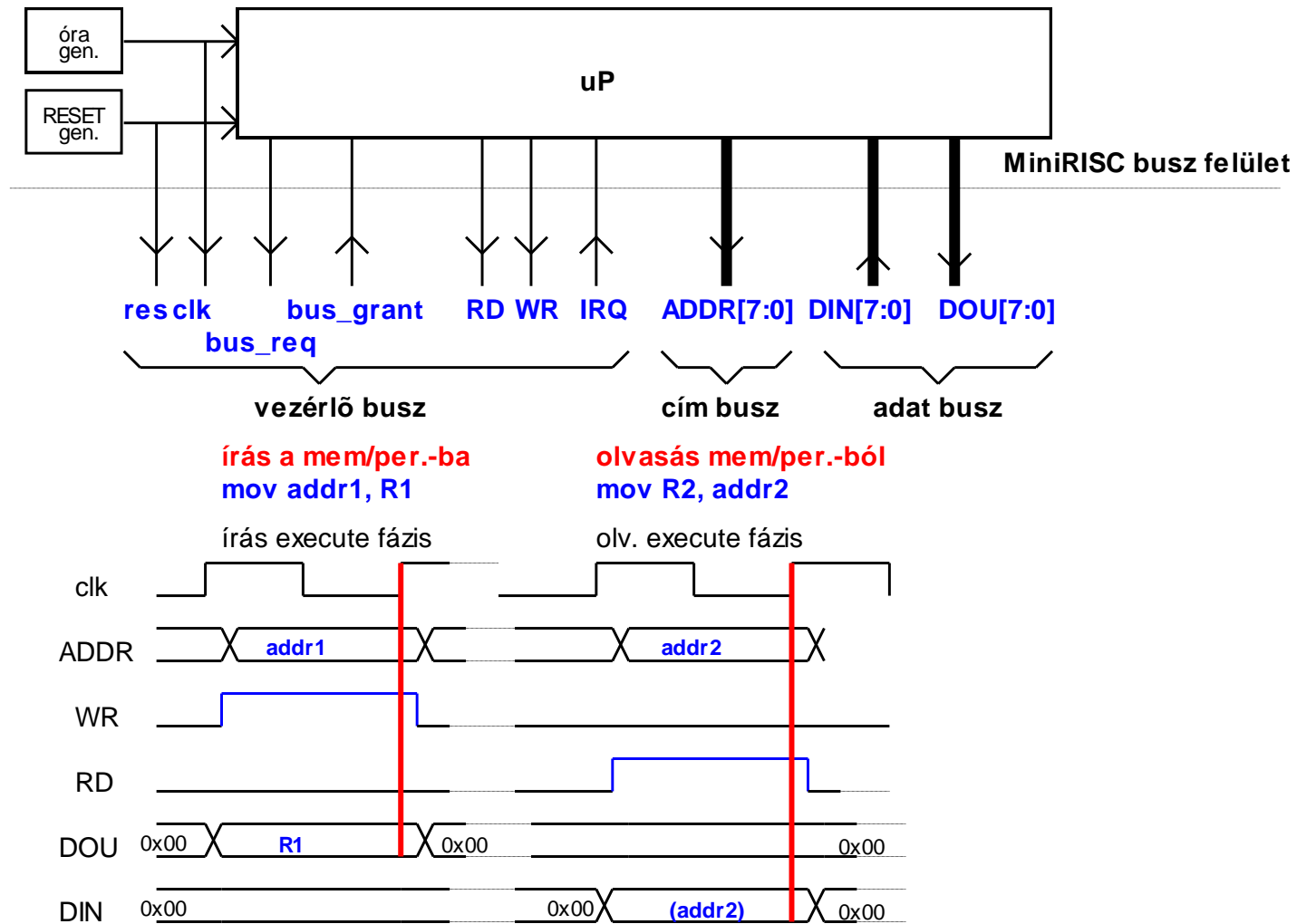
**Periféria illesztési példák  
áramkörön belüli buszra  
(kiegészítés az előadás vázlatához)**

**Benesóczky Zoltán  
(2017)**

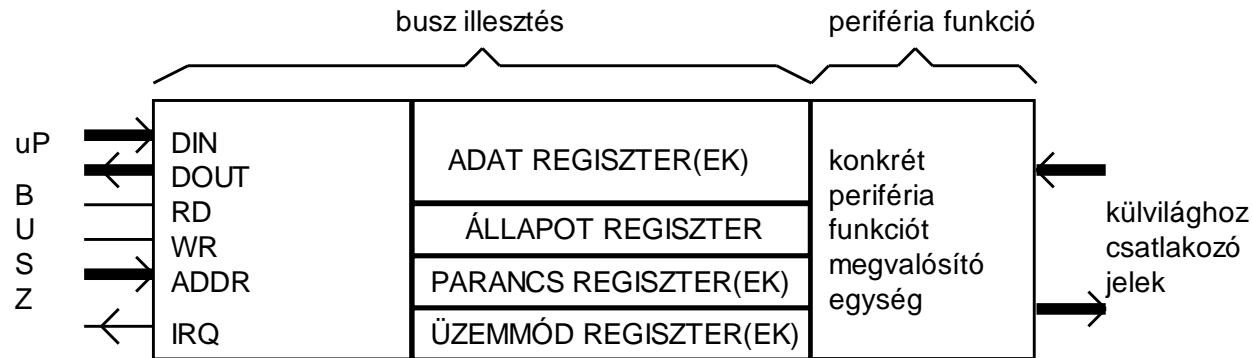
## Periféria illesztés mikroprocesszoros buszra (áramkörön belüli szinkron busz esetén)



Áramkörön belüli busz (pl. FPGA-ban megvalósított uP és perifériák esetén) a busz felület



## A perifériák tipikus programozói felülete



**Üzem mód regiszter:** A periféria működési módjának beállítására szolgál. Egy-egy periféria egy adott feladatcsoport megoldását teszi lehetővé konfigurálható üzemmódok segítségével (pl. egy timer egységgel időzítési, számlálási, impulzus generálási feladatok oldhatók meg).

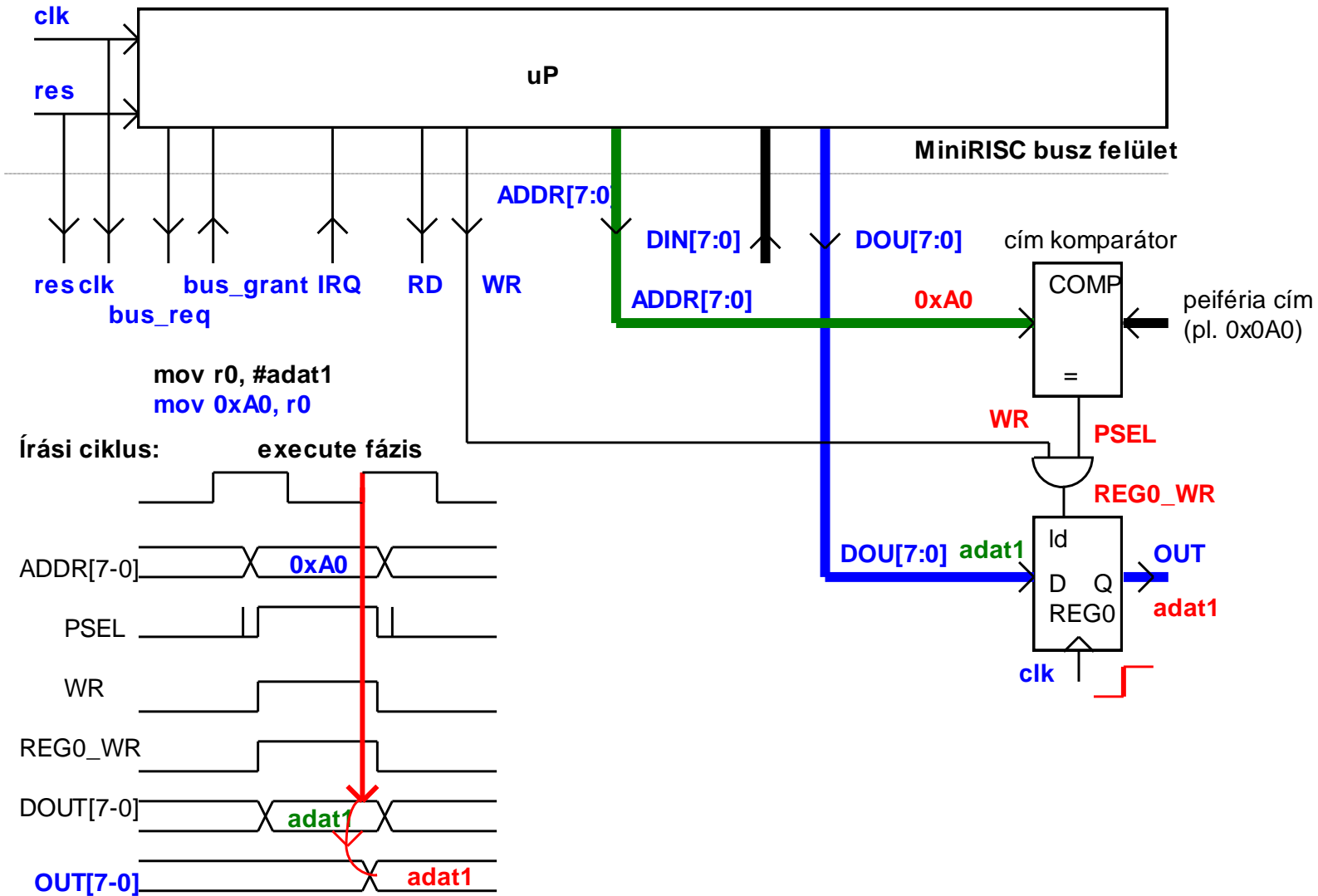
**Parancs regiszter:** A periféria valamely működését lehet vele kezdeményezni. Pl. A/D konverzió indítása.

**Sokszor az üzemmód és parancs regisztert összevonják és vezérlő regiszternek nevezik.**

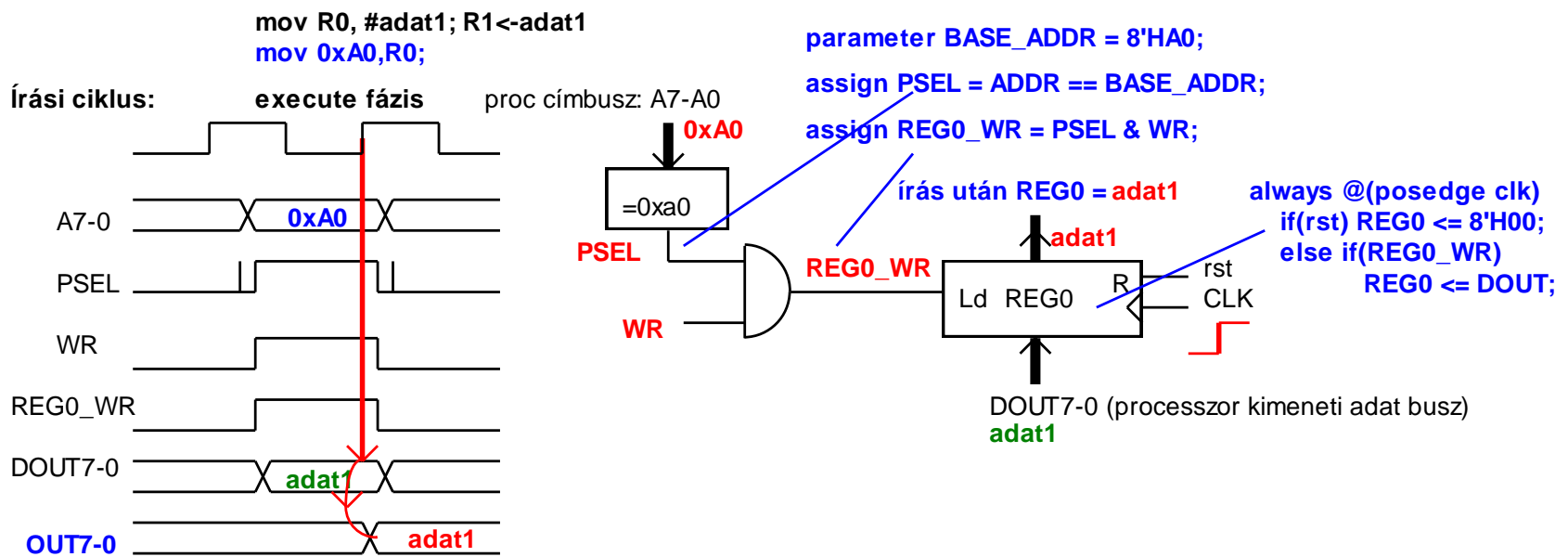
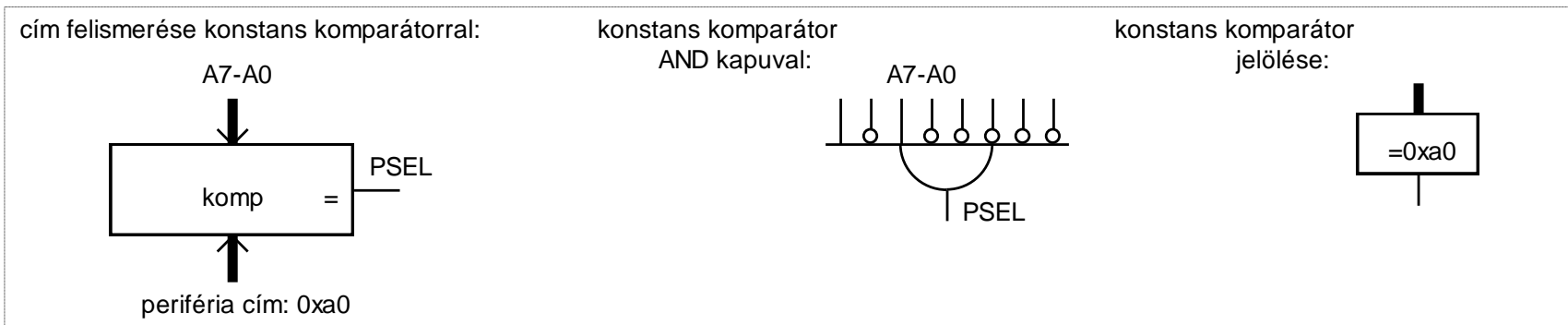
**Állapot (státus) regiszter:** A periféria állapotáról ad információkat. Az állapotregiszter bitjei ún. megszakítást is kérhetnek, ha az engedélyezett.

**Adat regiszter:** Ide kell beírni itt lehet kiolvasni az adatot (nem feltétlenül ugyanazt, amit beírtunk).

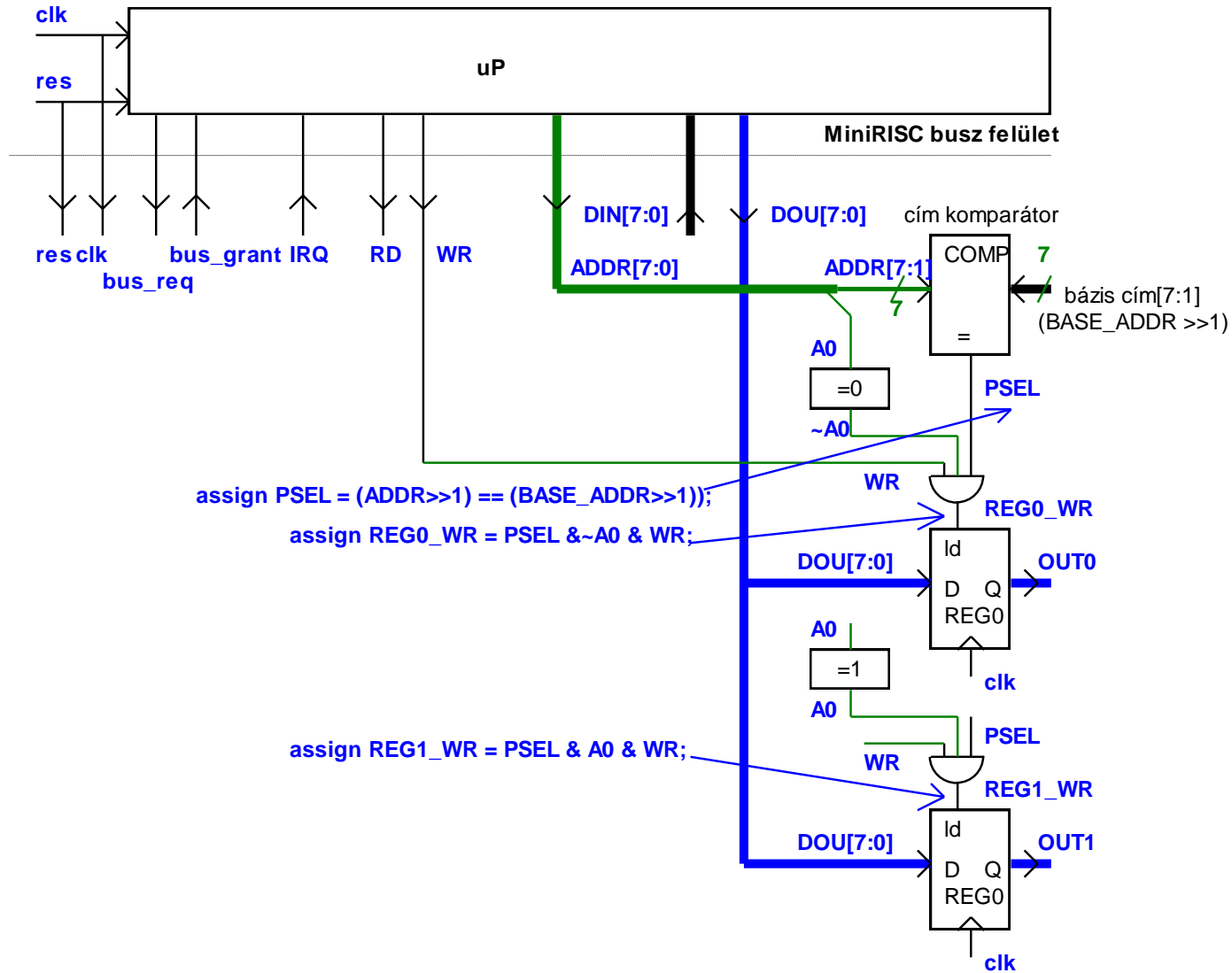
**Kimeneti port regiszter illesztése Mini RISC processzorhoz 1 kimeneti regiszter esetén (a periféria 1 címet foglal)**



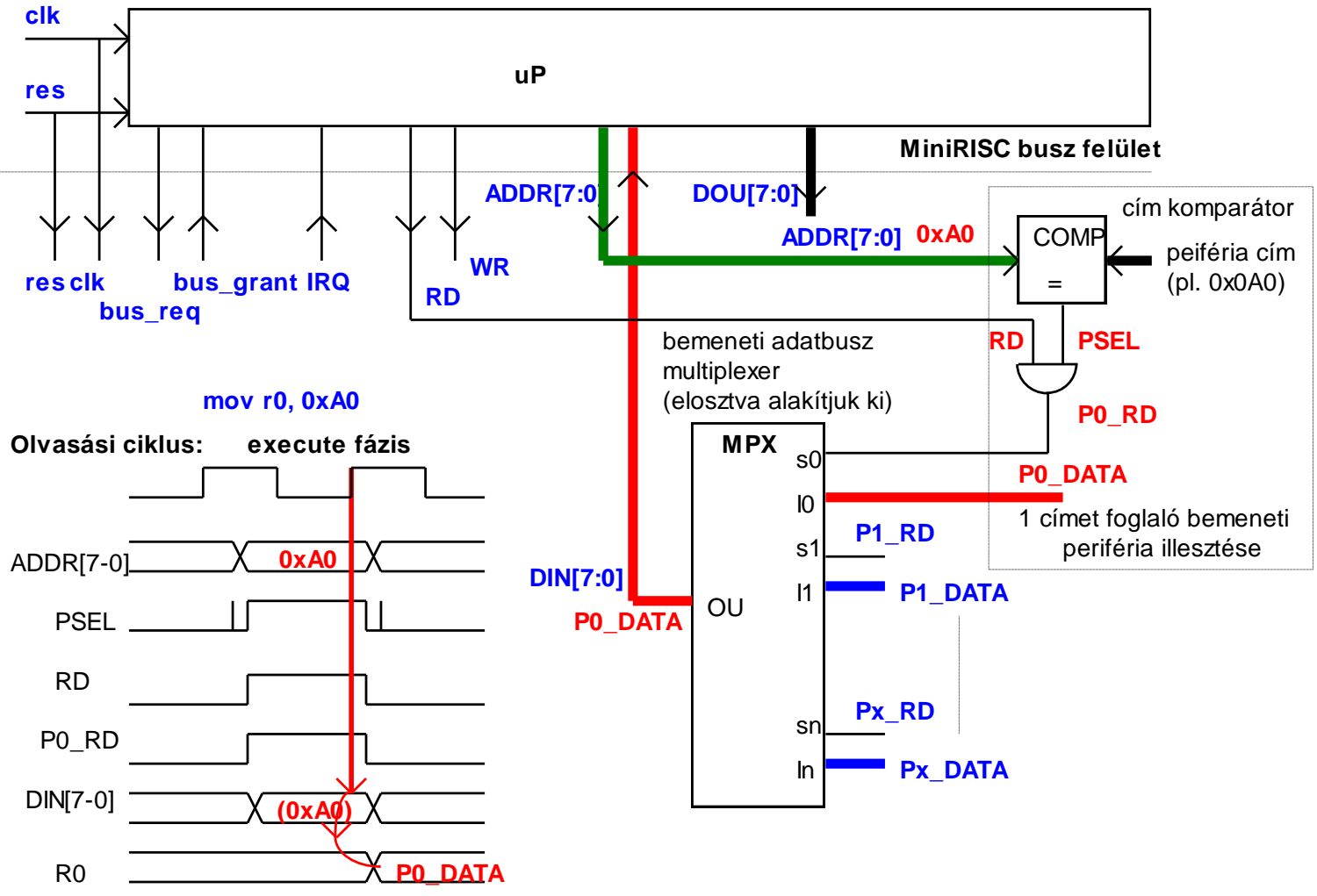
## Egy címmel rendelkező írható periféria illisztése



**Kimeneti port, regiszter illesztése Mini RISC processzorhoz 2 kimeneti regiszter esetén (a periféria 2 címet foglal)**

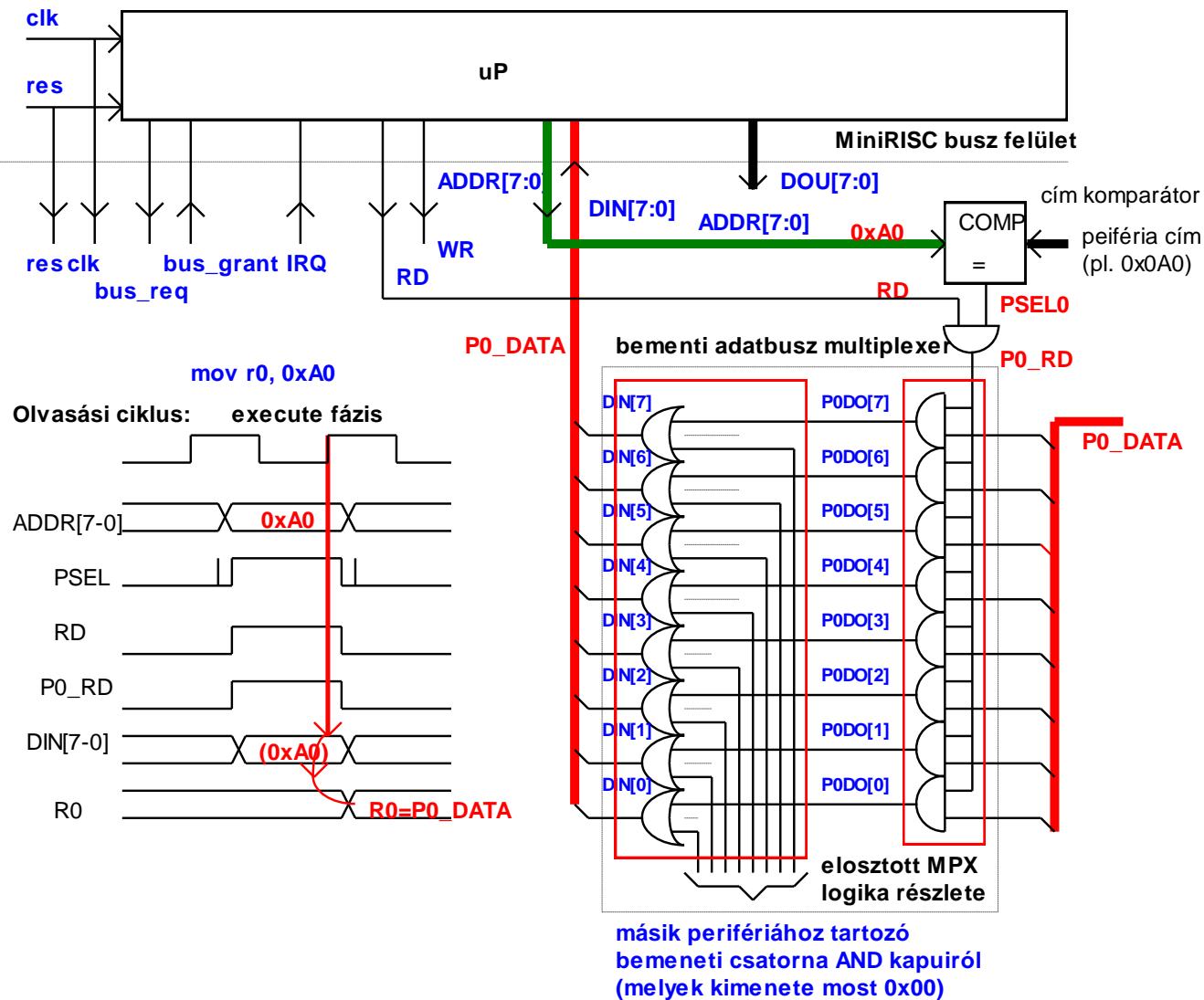


**Bemeneti port illesztése Mini RISC processzorhoz 1 adat forrás esetén (a periféria 1 címet foglal)**





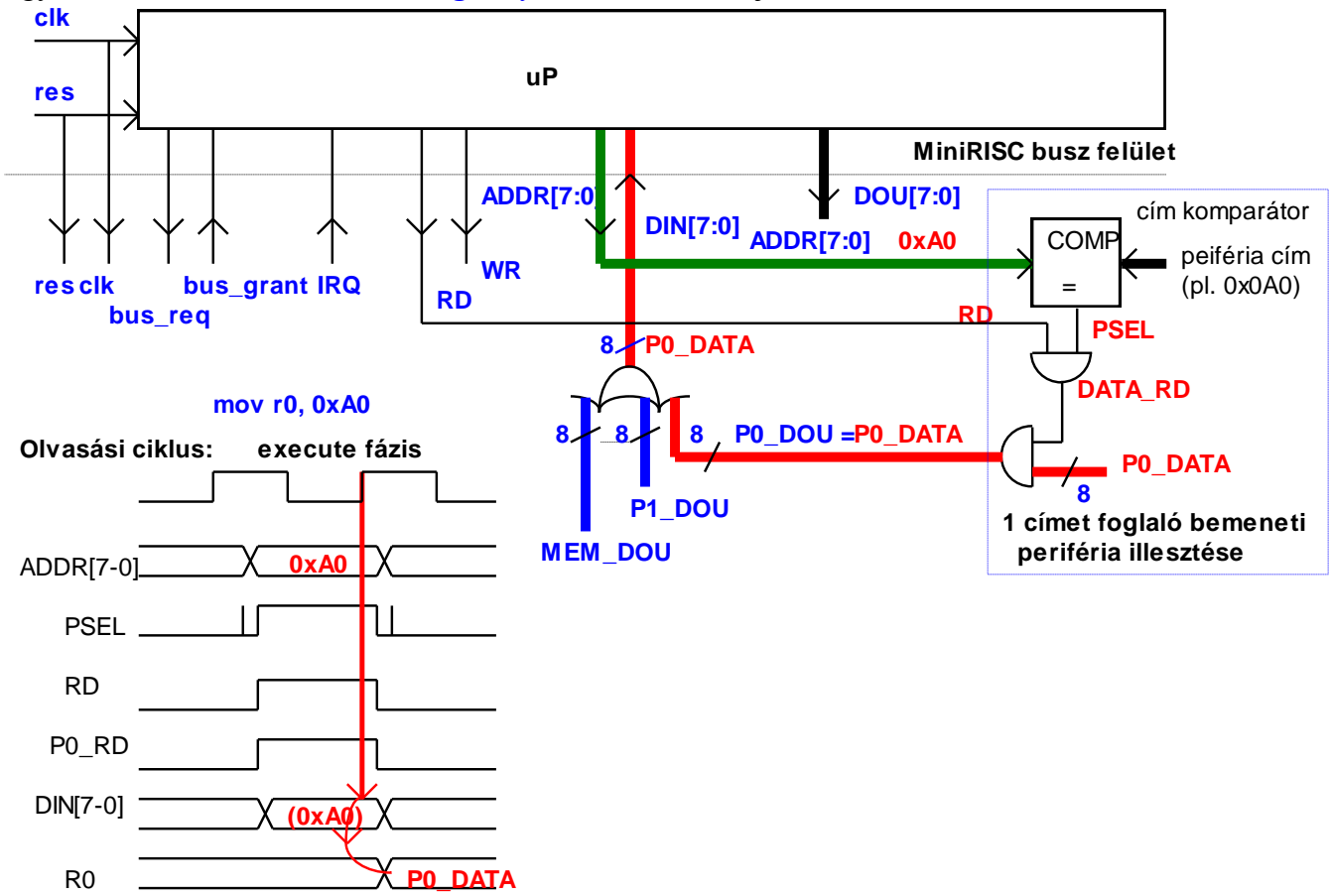
**Bemeneti port illesztése Mini RISC processzorhoz 1 adat forrás esetén (a periféria 1 címet foglal)**



**Elosztott multiplexer:**  
Az adatbusz MPX OR kapuit a processzorhoz, AND kapuit a perifériákhoz rendeljük

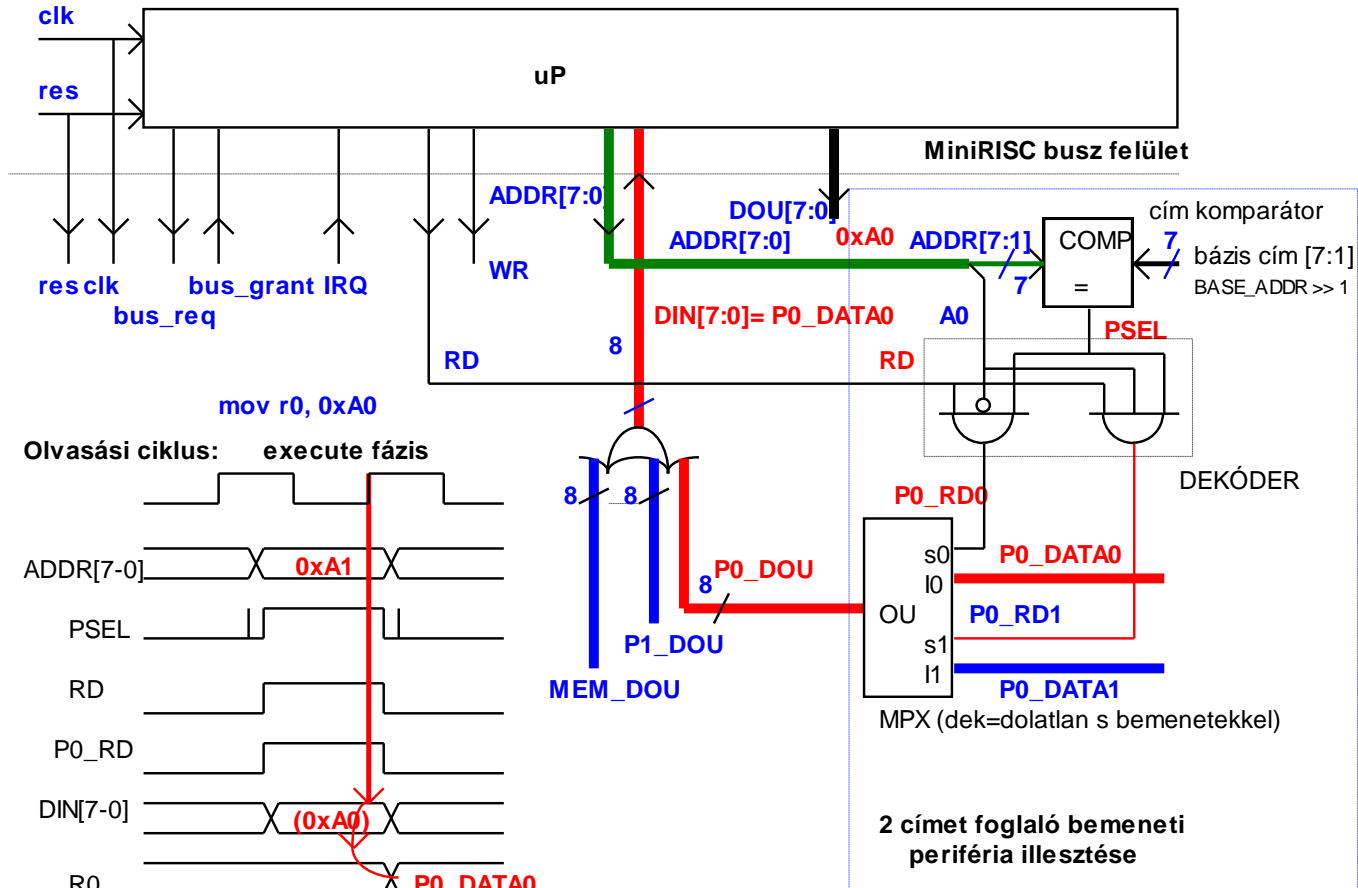
### Bemeneti port illesztése Mini RISC processzorhoz 1 adat forrás esetén (a periféria 1 címet foglal)

Egyszerűsített *busz OR* és *közös engedélyezésű busz AND* jelöléssel

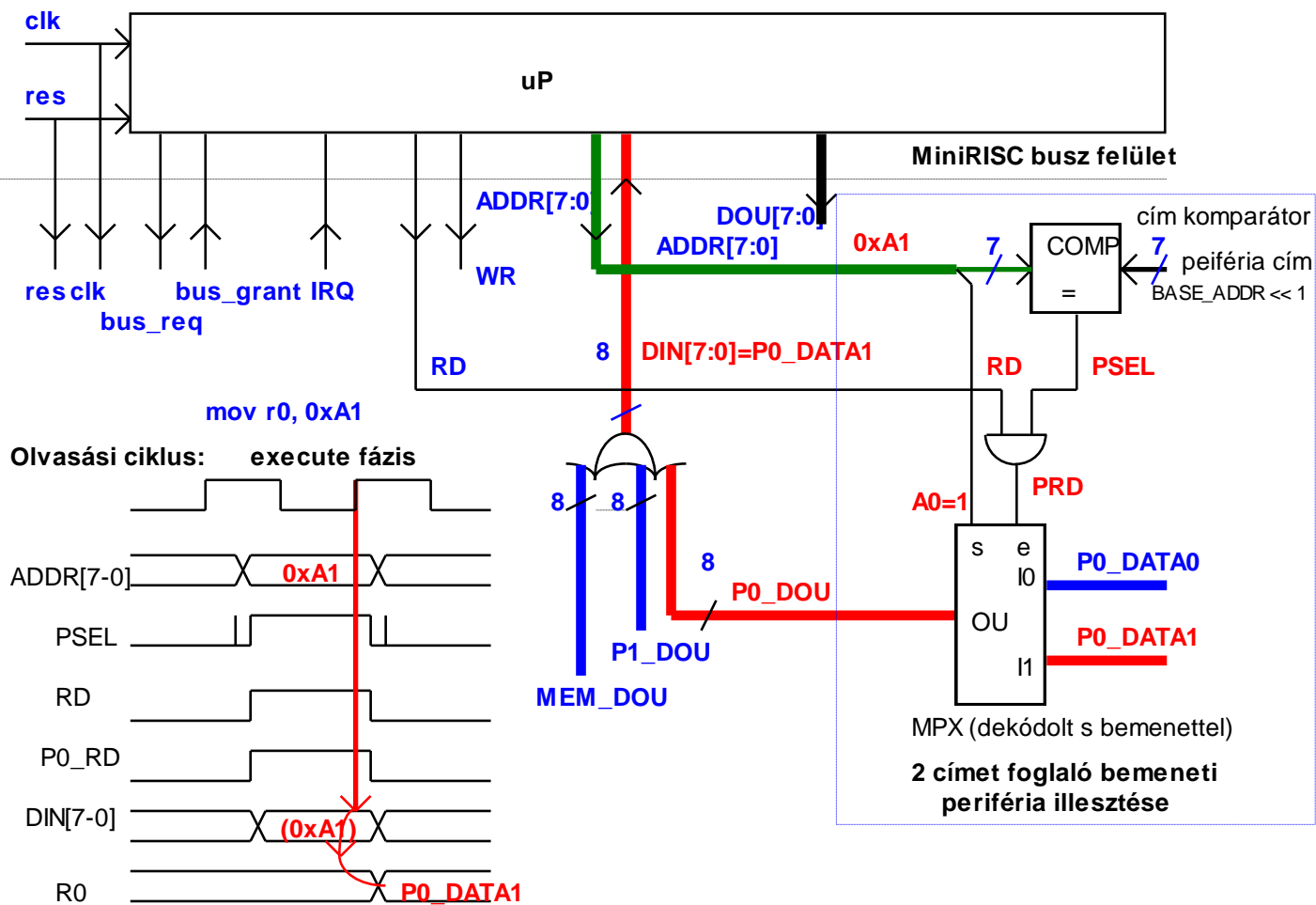


### Bemeneti port illesztése Mini RISC processzorhoz 2 adat forrás esetén (a periféria 2 címet foglal)

- Kódolatlan select-tekkel rendelkező multiplexert használva



**Bemeneti port illesztése Mini RISC processzorhoz 2 adat forrás esetén (a periféria 2 címet foglal)**  
 - Kódolt select-tel rendelkező multiplexert használva



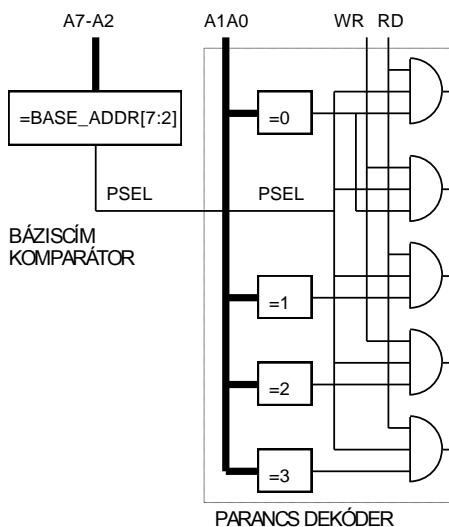
#### 4 cím fogláló periféria illesztése: parancs (RD/WR), státusz (RD), adat ki (WR), adat be (RD)

parameter BASE\_ADDR = 8'H0A0;

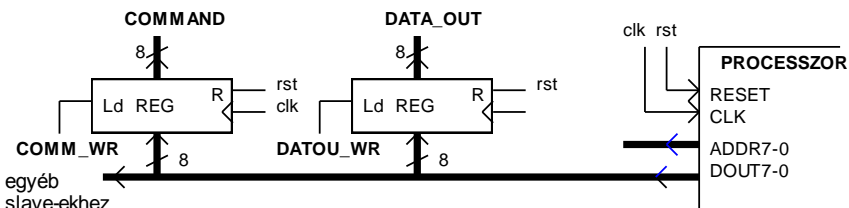
	A7	A6	A5	A4	A3	A2	A1	A0		
0xa0	1	0	1	0	0	0	0	0	W/R	COMAND
0xa1	1	0	1	0	0	0	0	1	R	STATUS
0xa2	1	0	1	0	0	0	1	0	W	DATA_OUT
0xa3	1	0	1	0	0	0	1	1	R	DATA_IN

parameter COMM\_ADDR = 2'b00;  
 parameter STAT\_ADDR = 2'b01;  
 parameter DATOU\_ADDR = 2'b10;  
 parameter DATIN\_ADDR = 2'b11;

assign PSEL = ((ADDR >>2) == (BASE\_ADDR >>2));

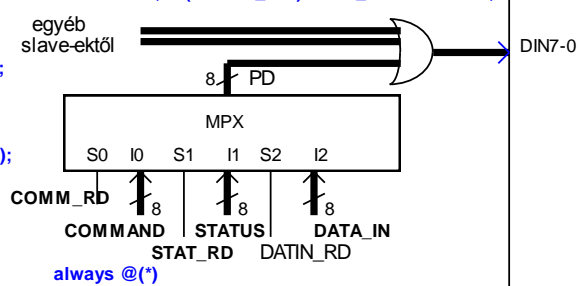


assign COMM\_RD = RD & PSEL & ((A1,A0) == COMM\_ADDR);  
 assign COMM\_WR = WR & PSEL & ((A1,A0) == COMM\_ADDR);  
 assign STAT\_RD = RD & PSEL & ((A1,A0) == STAT\_ADDR);  
 assign DATOU\_WR = WR & PSEL & ((A1,A0) == DATOU\_ADDR);  
 assign DATIN\_RD = RD & PSEL & ((A1,A0) == DATIN\_ADDR);



```
always @(posedge clk)
    if(rst) COMMAND <= 0x00;
else
    if(COMM_WR) COMMAND <= DOUT;

always @(posedge clk)
    if(rst) DATA_OUT <= 0x00;
else
    if(DATOU_WR) DATA_OUT <= DOUT;
```



```
always @(*)
    case {{COMM_RD, STAT_RD, DATIN_RD}}
        3'b100: PD <= COMMAND;
        3'b010: PD <= STATUS;
        3'b001: PD <= DATA_IN;
        default: PD <= 8'h00;
    endcase
```

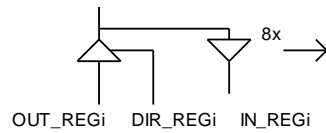
(Ha nincs több periféria, akkor nincs DIN előtt a 8db VAGY kapu, így olvasáskor a periféria közvetlenül DIN-re teszi az adatot a mostani PD helyett. A ZH feladatokban is ez a helyzet.)

# GPIO általános célú be/kimeneti port

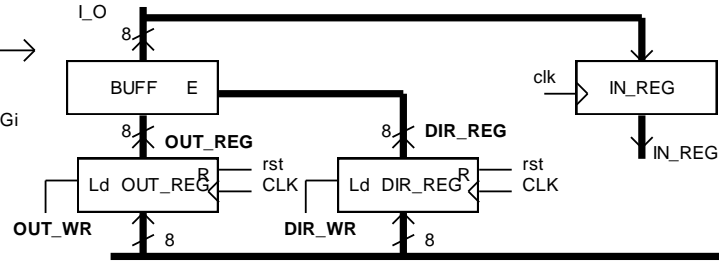
## GPIO A felépítése

parameter BASE\_ADDR = 8'HA0;

A7	A6	A5	A4	A3	A2	A1	A0		
1	0	1	0	0	0	0	0	R/W	OUT_REG
1	0	1	0	0	0	0	1	R	IN_REG
1	0	1	0	0	0	1	0	R/W	DIR_REG



parameter OUT\_ADDR = 2'b00;  
parameter IN\_ADDR = 2'b01;  
parameter DIR\_ADDR = 2'b10;



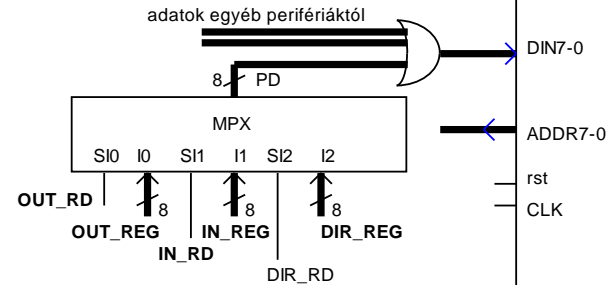
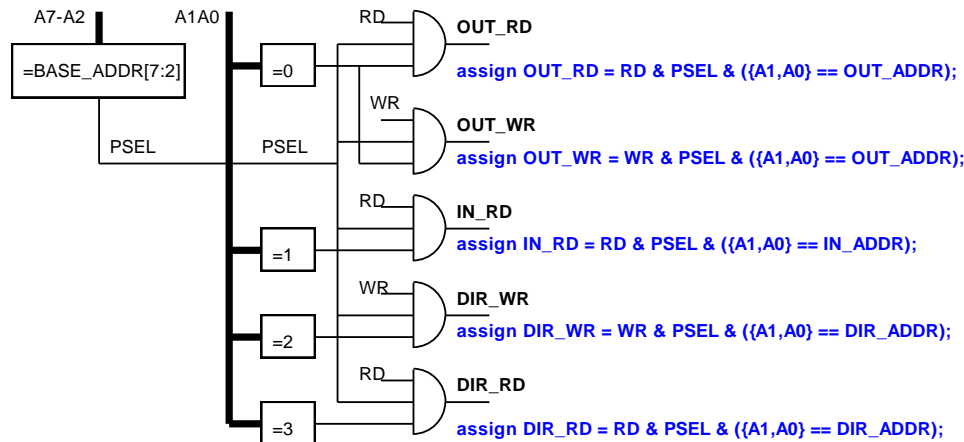
always @ (posedge clk)  
IN\_REG <= I\_O;

always @ (posedge clk)  
if(rst) OUT\_REG <= 0x00;  
else  
if(OUT\_WR) OUT\_REG <= DOUT;

always @ (posedge clk)  
if(rst) DIR\_REG <= 0x00;  
else  
if(DIR\_WR) DIR\_REG <= DOUT;

assign PSEL = (ADDR >> 2) == (BASE\_ADDR >> 2);

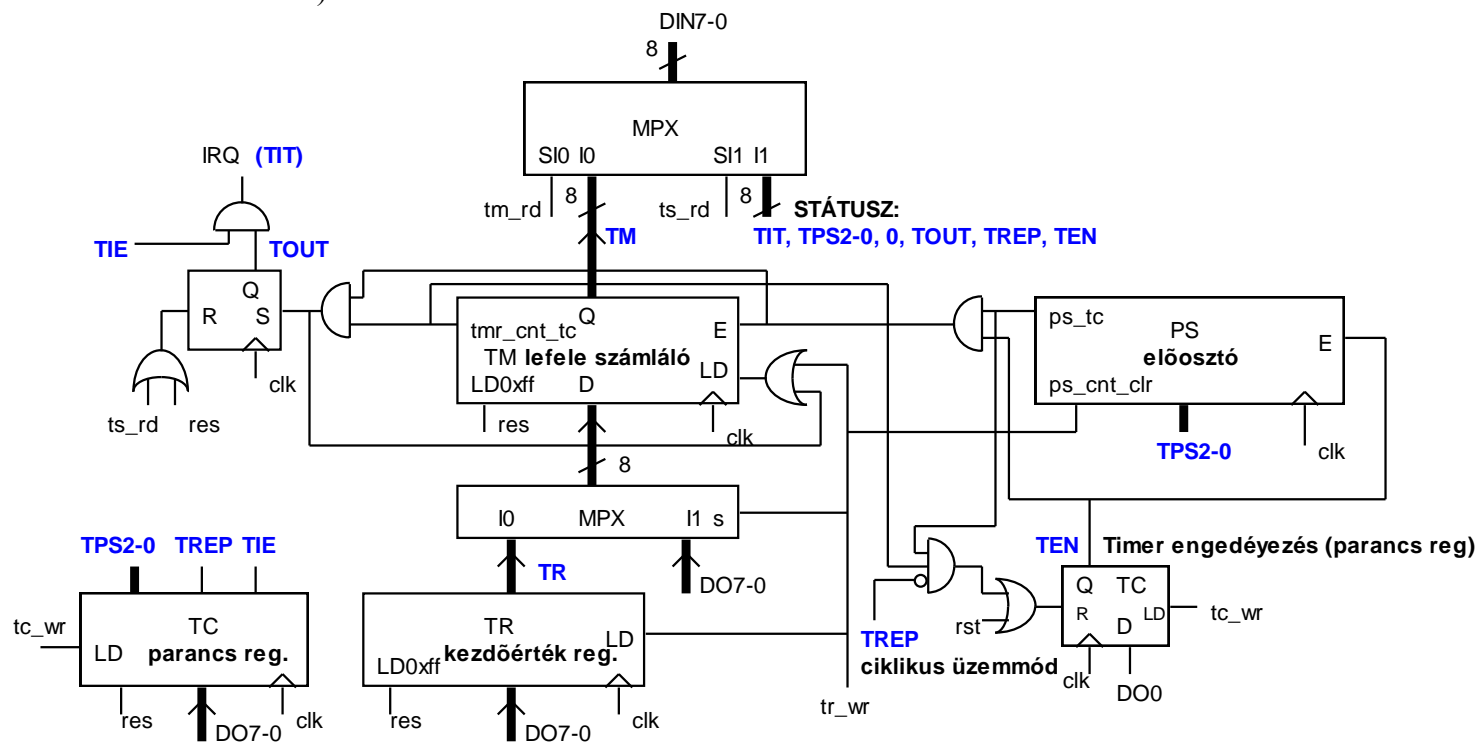
másképp is lehet: assign PSEL = ADDR[7:2] == BASE\_ADDR[7:2];



always @ (\*)  
case ({OUT\_RD,IN\_RD, DIR\_RD})  
3'b100: PD <= OUT\_REG;  
3'b010: PD <= IN\_REG;  
3'b001: PD <= DIR\_REG;  
default: PD <= 8'h00;  
endcase

## TIMER egység blokkvázlata

(cím és parancs dekódolók nélkül)

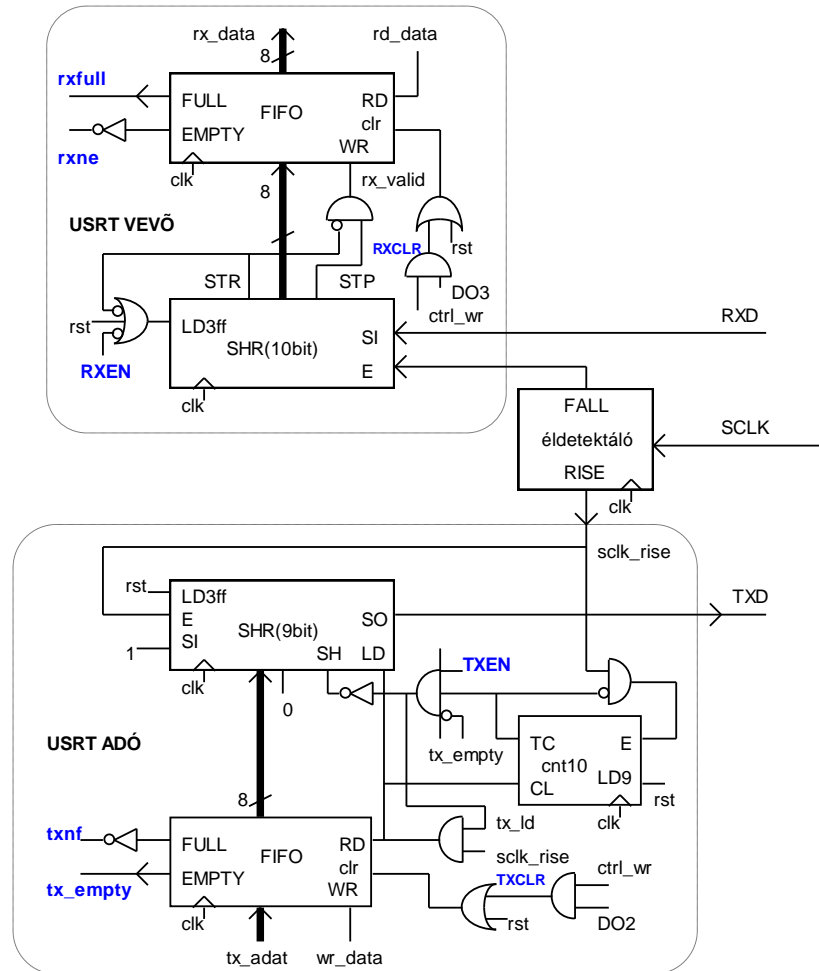


	D7	D6	D5	D4	D3	D2	D1	D0
parancs regiszter TC:	TIE	TPS2	TPS1	TPS0	-		TREP	TEN
státusz regiszter TS:	TIT	TPS2	TPS1	TPS0	0	TOUT	TREP	TEN

TPS2-0:	000	001	010	011	100	101	110	111
előosztás:	1	16	64	256	1024	4096	16384	65536
Periódusidő (frekvencia) ha felk: 16MHz	1/16 us (16MHz)	1 us (1MHz)	4 us (250kHz)	16 us (62500Hz)	64 us (15625Hz)	256 us (3906.25Hz)	1024 us (976.5625Hz)	4096 us (244.141Hz)

## USRT blokkvázlata

(cím és parancs dekódolók nélkül)



	D7	D6	D5	D4	D3	D2	D1	D0
parancs regiszter UC:	-	-	-	-	RXCLR	TXCLR	RXEN	TXEN
státus regiszter US:	0	0	0	0	RXFULL	RXNE	TXNF	TXEMPTY
IT engedélyezés UIE	0	0	0	0	RXFULL	RXNE	TXNF	TXEMPTY



## Két masteres rendszer mintapélda, MiniRISC processzor és DMA vezérlő (DMC) (nem tananyag)

