Az utasítás végrehajtása 3 fázisra bontható. Ezt a CPU-ban levő vezérlő (szinkron sorrendi hálózatként megvalósított állapotgép) valósítja meg, a processzor többi részének megfelelő vezérlőjeleket adva.

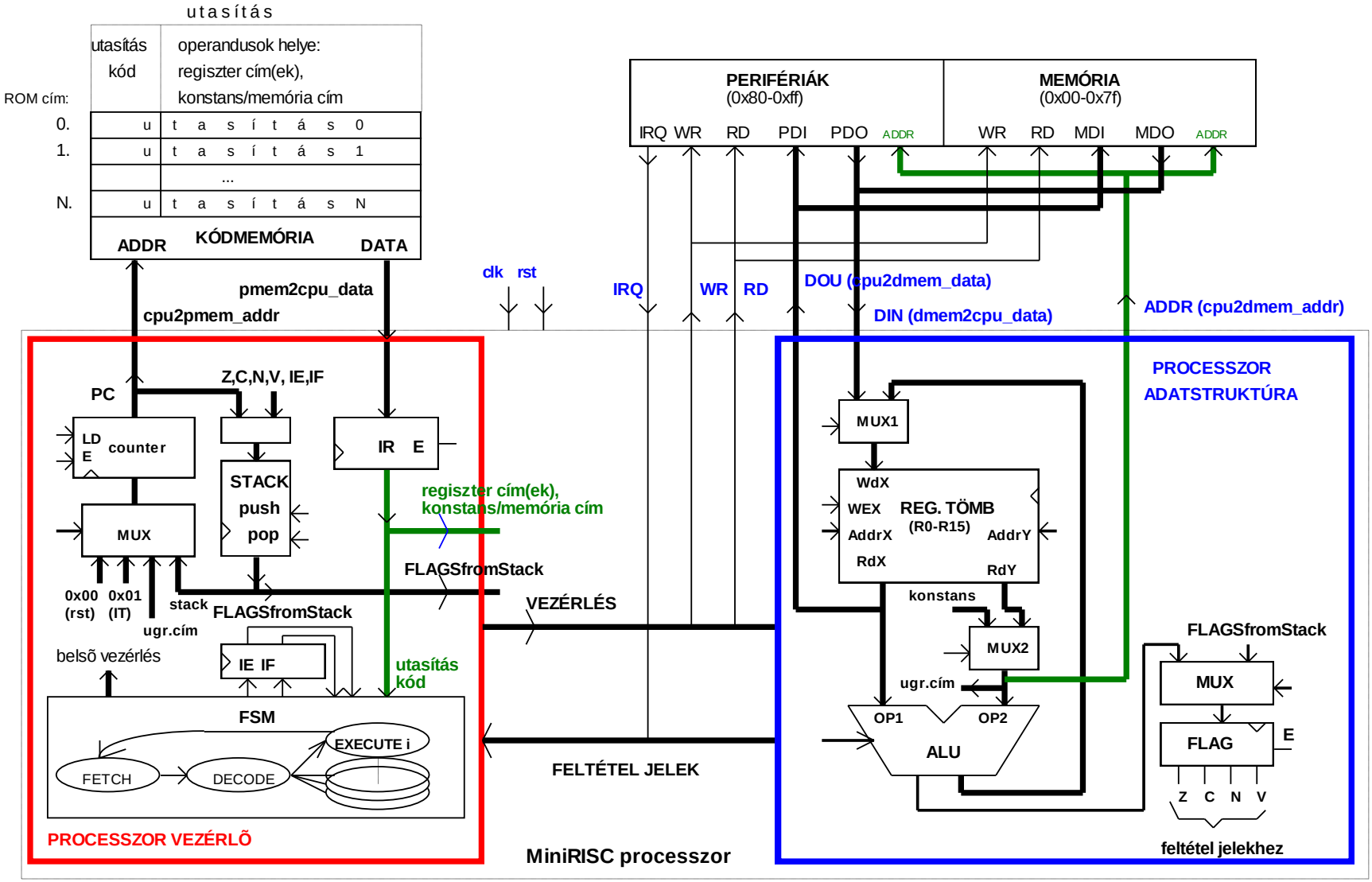
- ***FETCH***: Utasítás beolvasása az utasítás regiszterbe. A FETCH ciklus alatt a vezérlő engedélyezi az utasítás regiszterbe írást. Maga az írás a vezérlő FETCH állapota alatti órajelre történik meg. Ugyanerre az órajelre a vezérlő átlép a DECODE állapotba és a PC-t (a kódmemóriát címző számlálót) növeli, hogy az következő utasítás várható címére mutasson. (Az ugró és szubrutin hívó utasítások ezt felülírják az EXECUTE ciklusban.)

- ***DECODE***: Az utasítás dekódolása. Annak eldöntése, hogy melyik utasításról van szó. Ezt az utasítás kód néhány bitje kódolja. Ezektől függően a vezérlő a DECODE állapot alatt jövő órajelre az aktuális utasítás végrehajtásához szükséges vezérlő jeleket előállító *valamelyik* EXECUTE állapotba lép, a sok közül (ezt itt EXECUTE i-vel jelöljük). Ilyenből annyi van, ahány egymástól eltérő (a vezérlő által előállítandó) vezérlő jelet igénylő utasítás van. (A vezérlő jelek egy része közvetlenül az utasításregiszterből jön. Pl. az operandusok címe.)

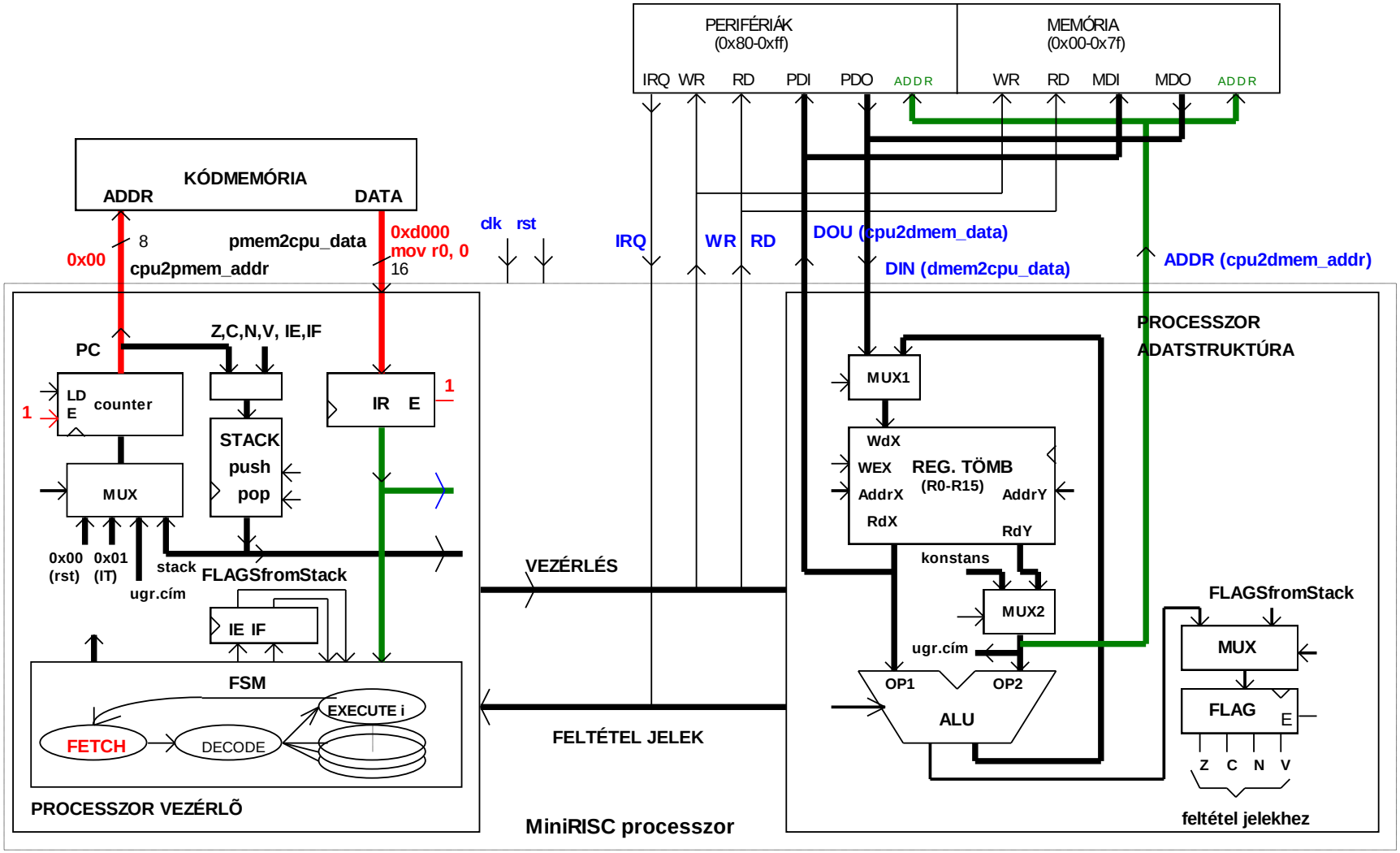
- ***EXECUTE***: Az utasítás végrehajtása. A vezérlő az EXECUTE i állapotban kiadja az aktuális utasítás végrehajtásához szükséges vezérlőjeleket. Az utasítás végrehjatása az EXECUTE i alatt érkező órajelre történik meg. Tehát amikor a vezérlő az EXECUTE i állapotból átlép a következő állapotba. (Ez általában a következő utasítást beolvasó FETCH állapot, ez alól kivétel, ha az utasítás alatt a vezérlő interruptot észlel.)

A MiniRISC processzor belső felépítésének blokkvázlata

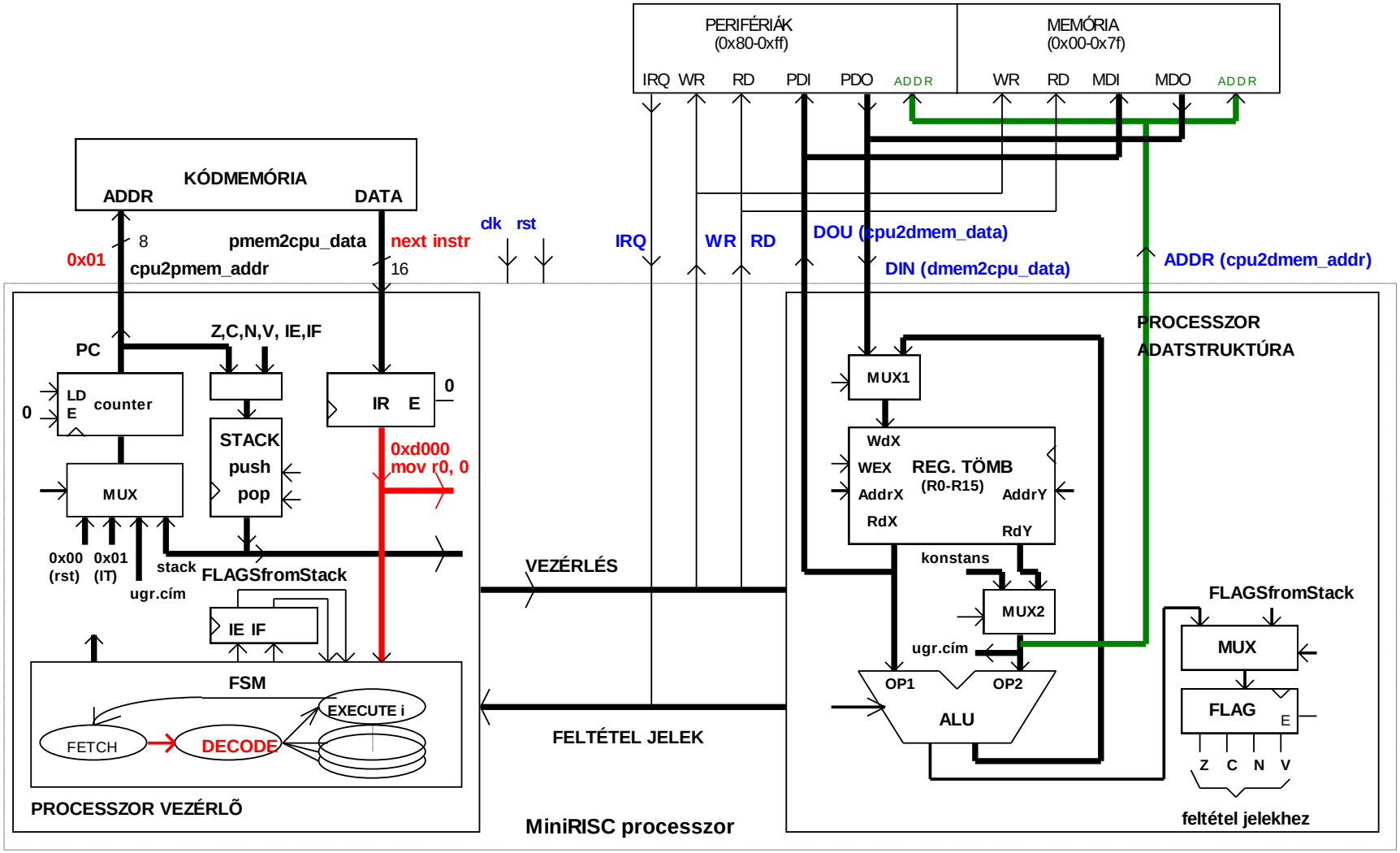
A szaggatott vonallal körülhatárolt rész a MiniRISC processzor (CPU), a többi a kódmemória, adat memória és perifériák.



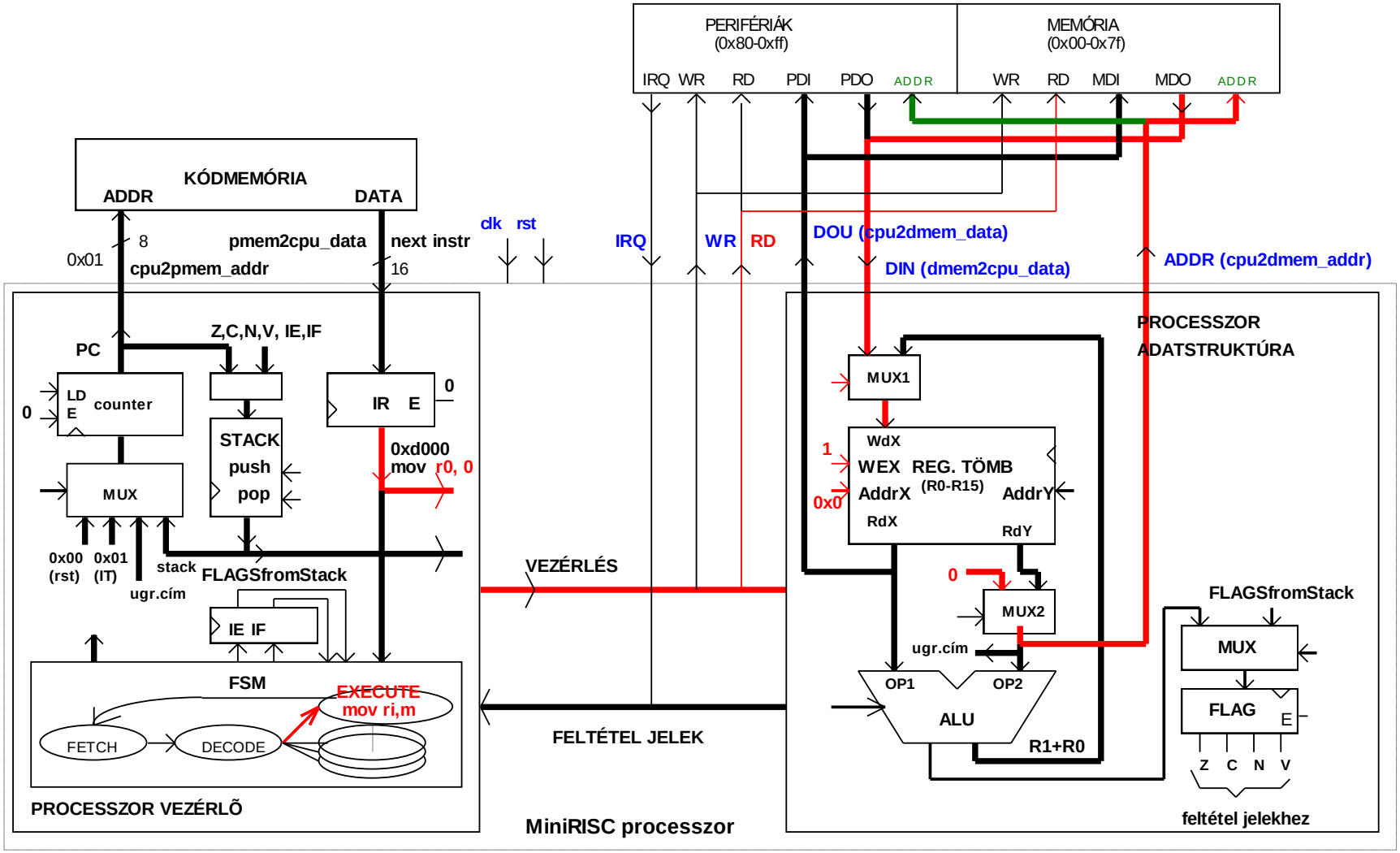
Adatmozgató utasítás, olvasás a memóriából: **FETCH** (uatsítás kód olvasás) ciklus



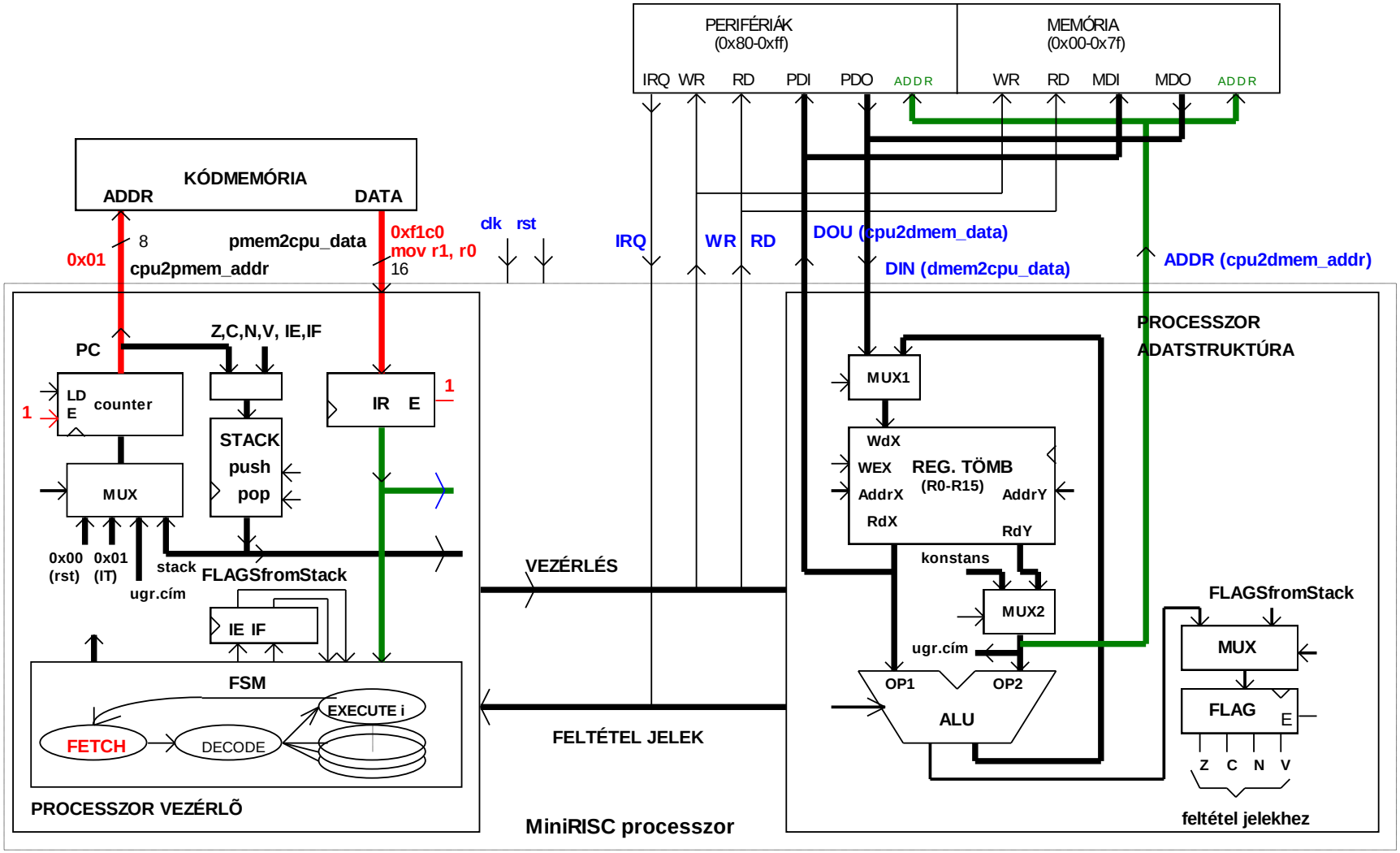
Adatmozgató utasítás, olvasás a memóriából: **DECODE** (utasítás dekódolás) ciklus



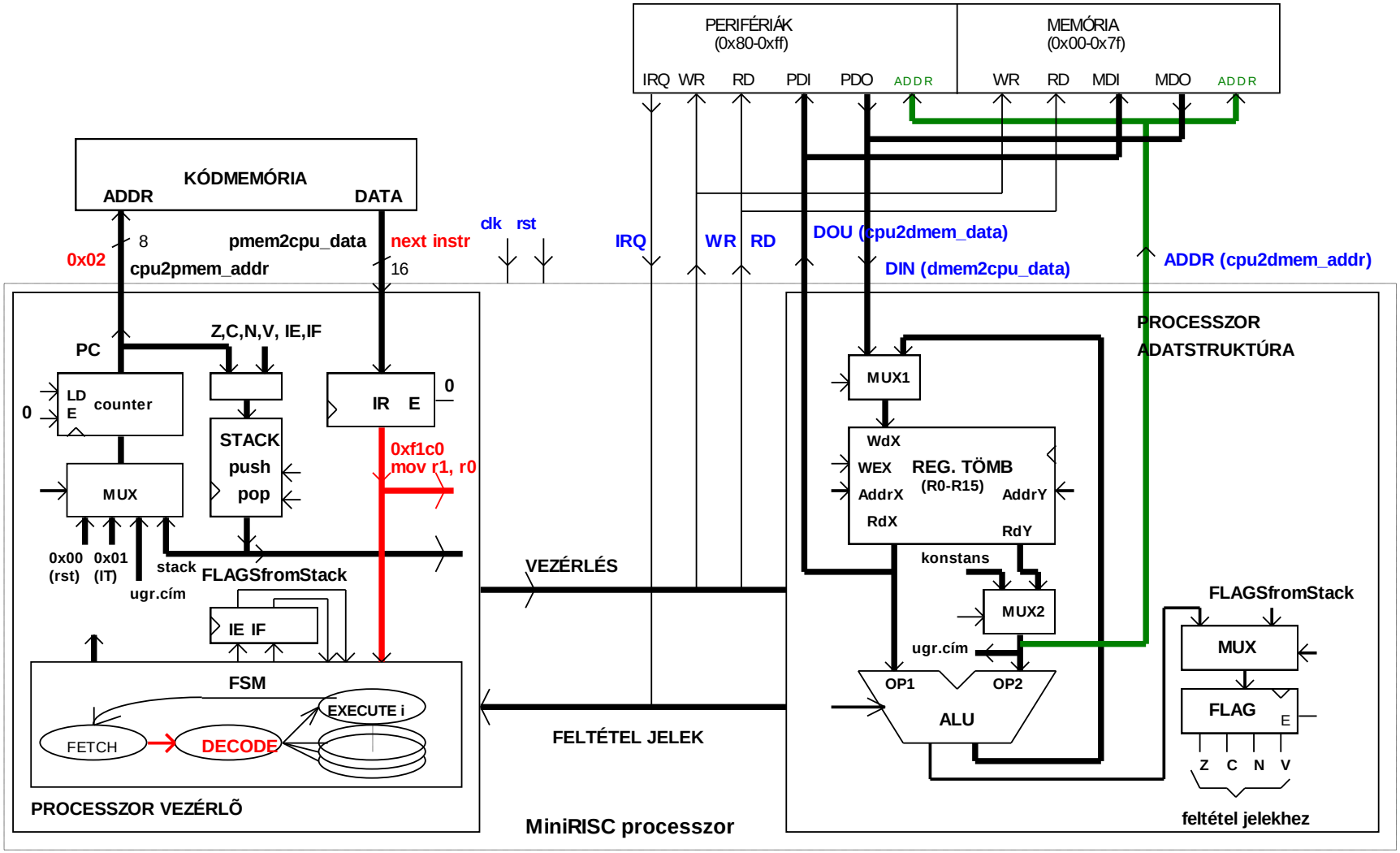
Adatmozgató utasítás, olvasás a memóriából: **EXECUTE** (utasítás végrehajtás) ciklus



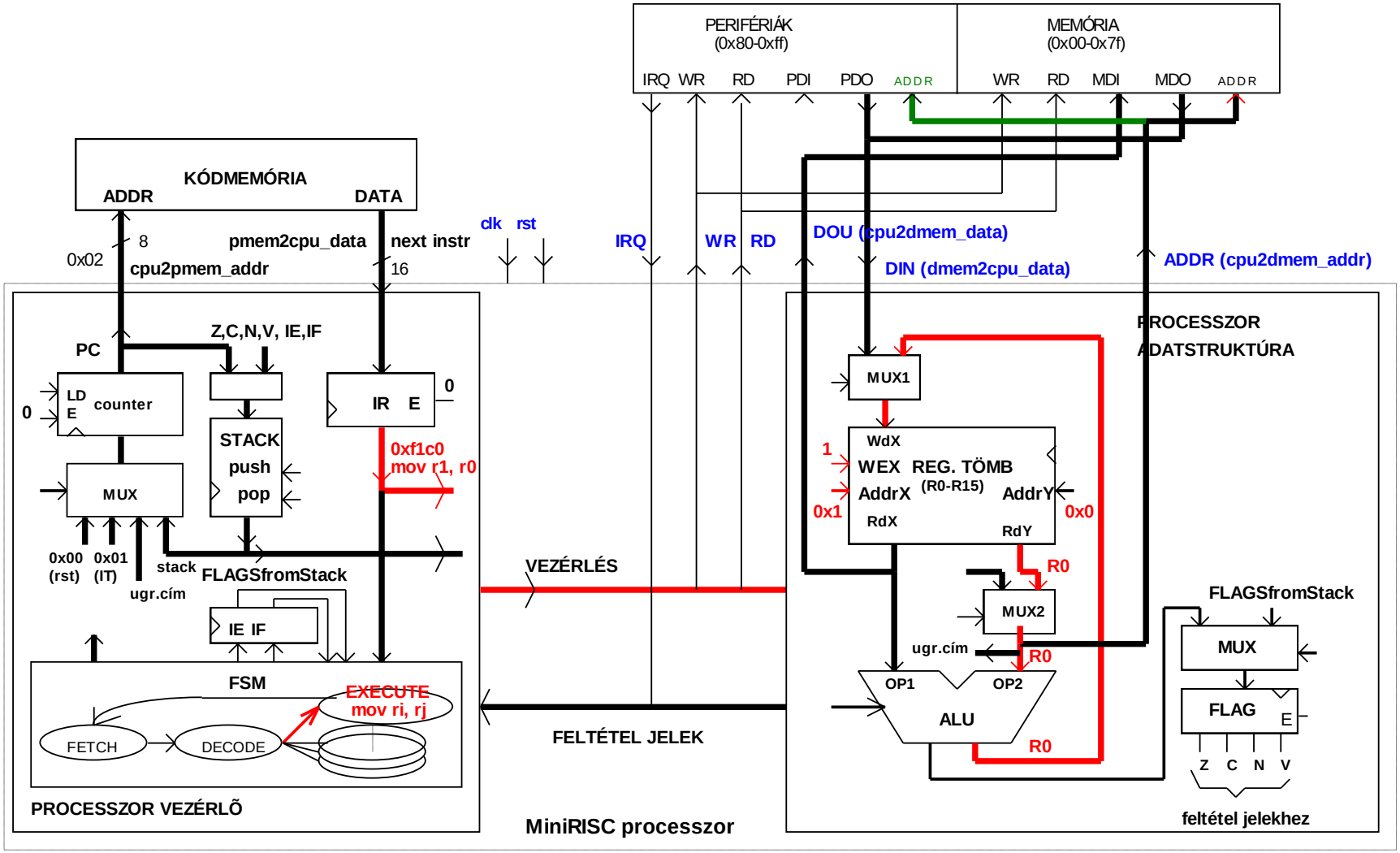
Adatmozgató utasítás, regiszterek között: **FETCH** (utasítás beolvasás) ciklus



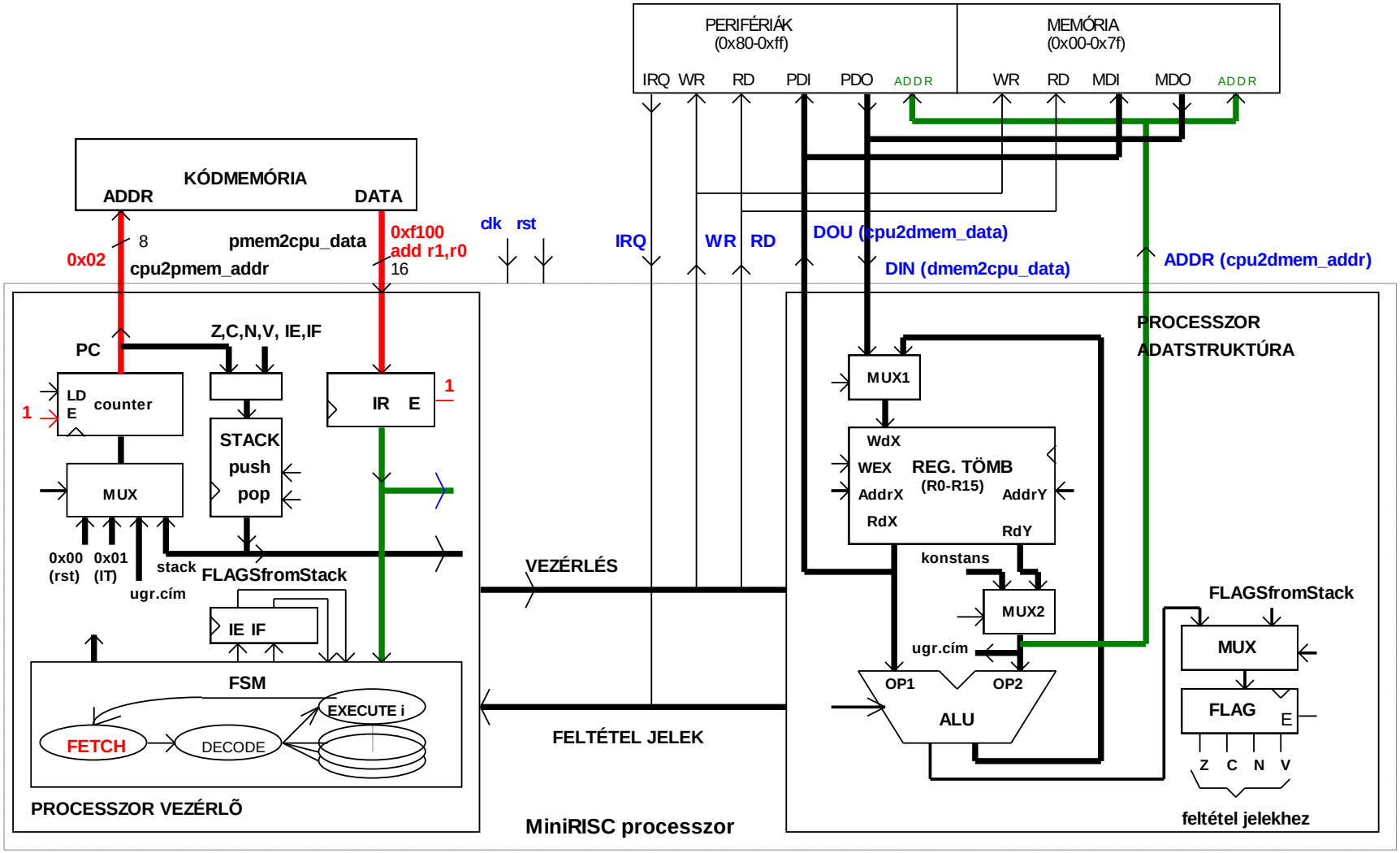
Adatmozgató utasítás, regiszterek között: **DECODE** (utasítás dekódolás) ciklus



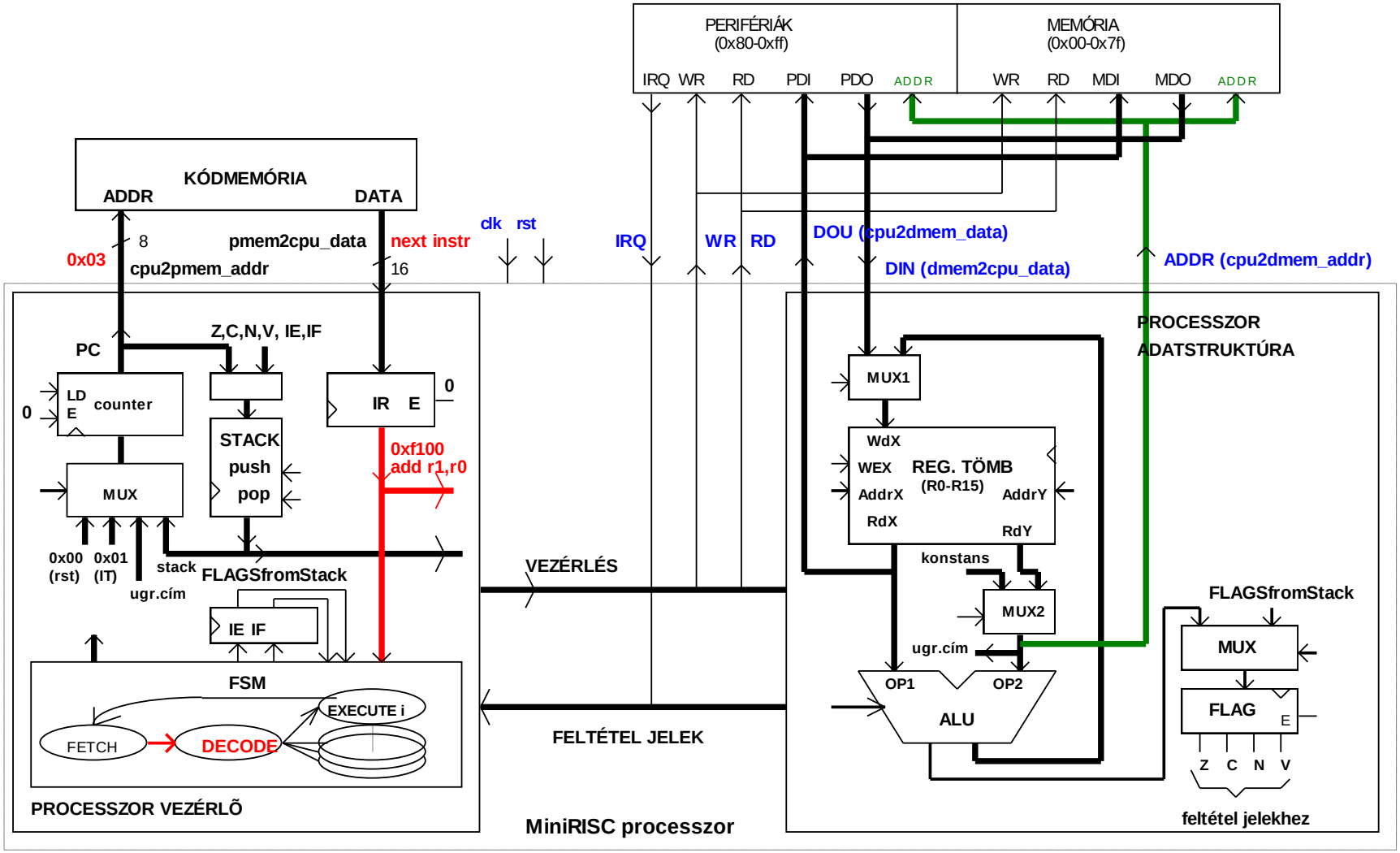
Adatmozgató utasítás, regiszterek között: **EXECUTE** (utasítás végrehajtás) ciklus



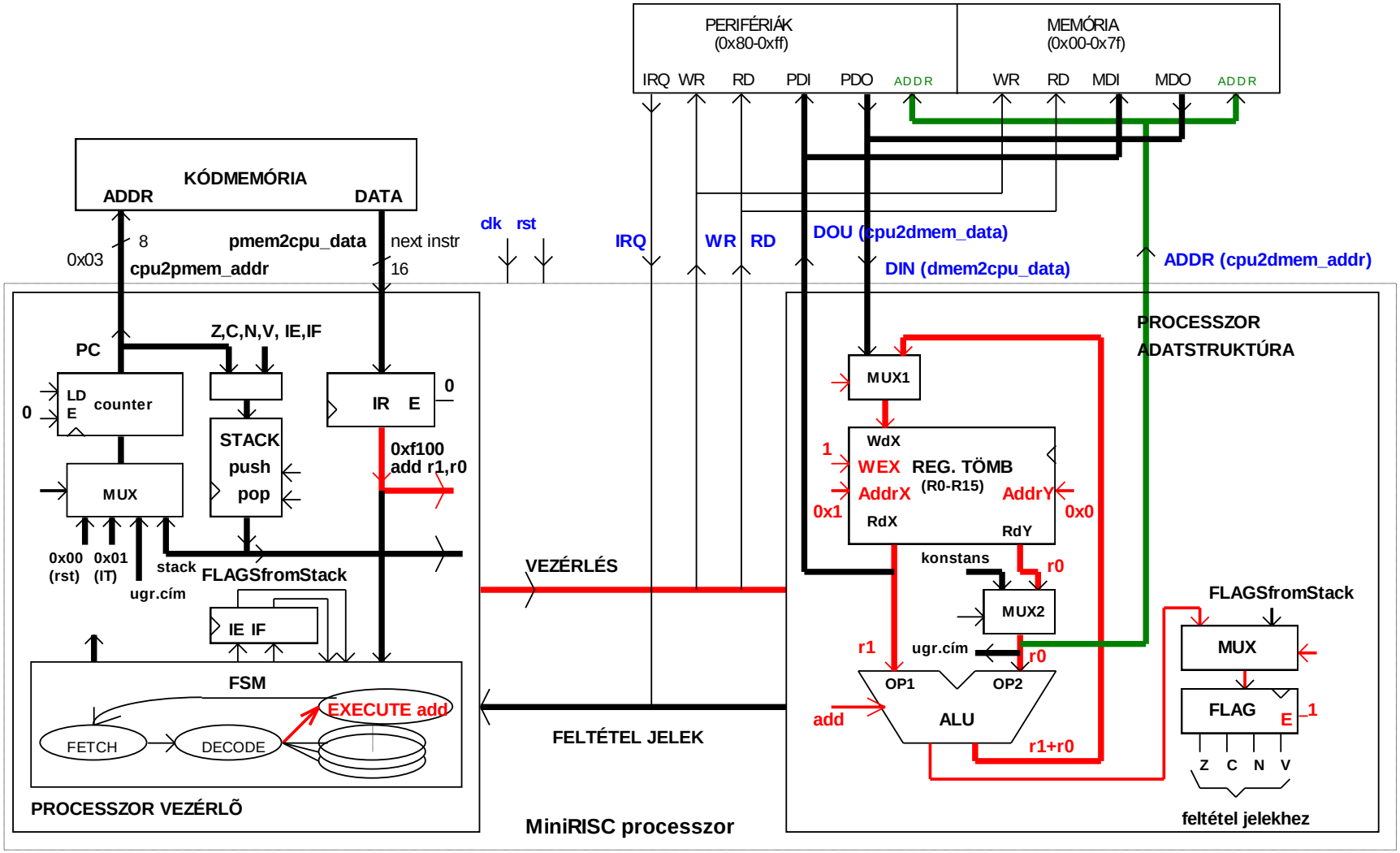
művelet végző utasítás (aritmetikai) összeadás: **FETCH** (utasítás beolvasás) ciklus



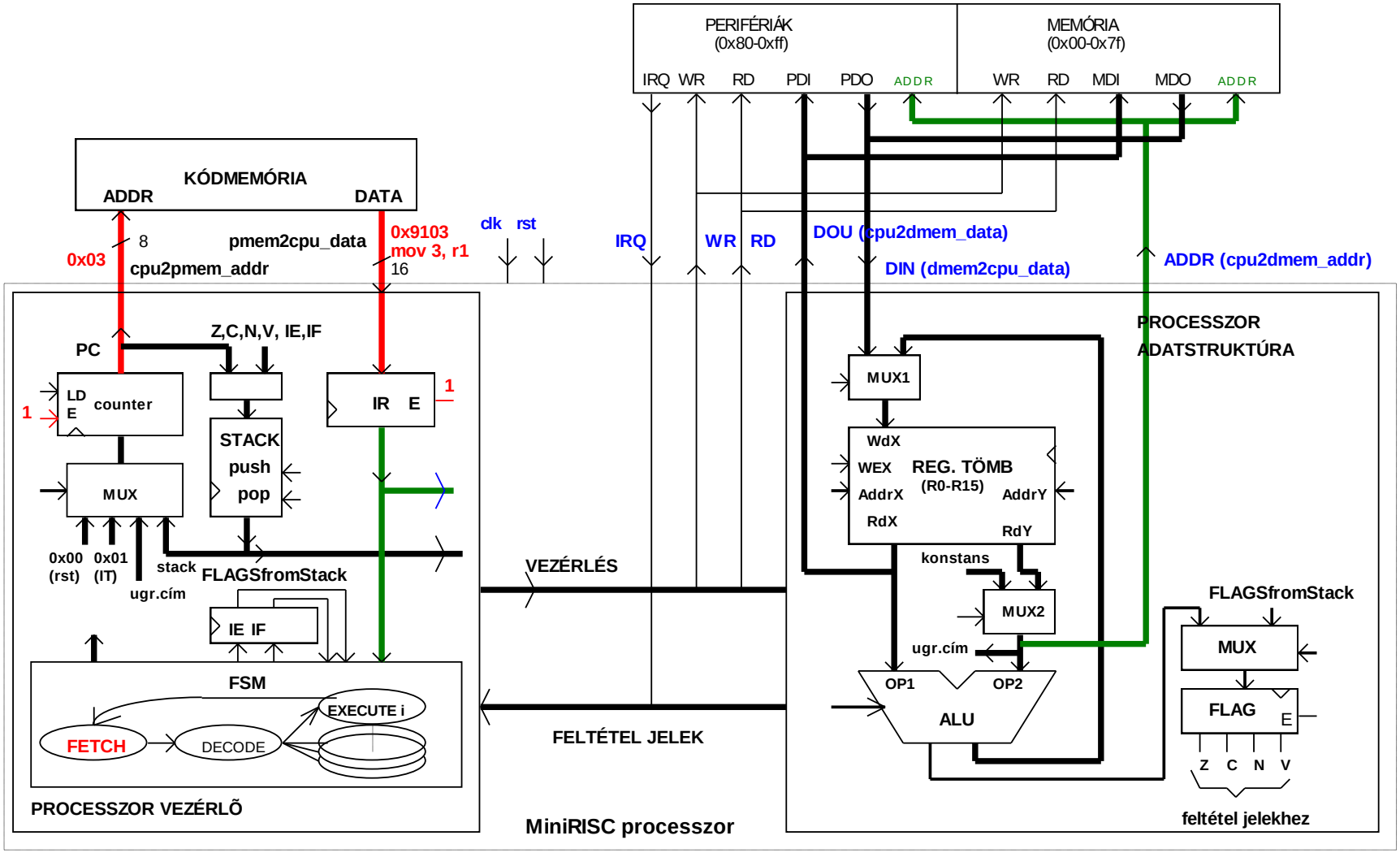
művelet végző utasítás (aritmetikai) összeadás: **DECODE** (utasítás dekódolás) ciklus



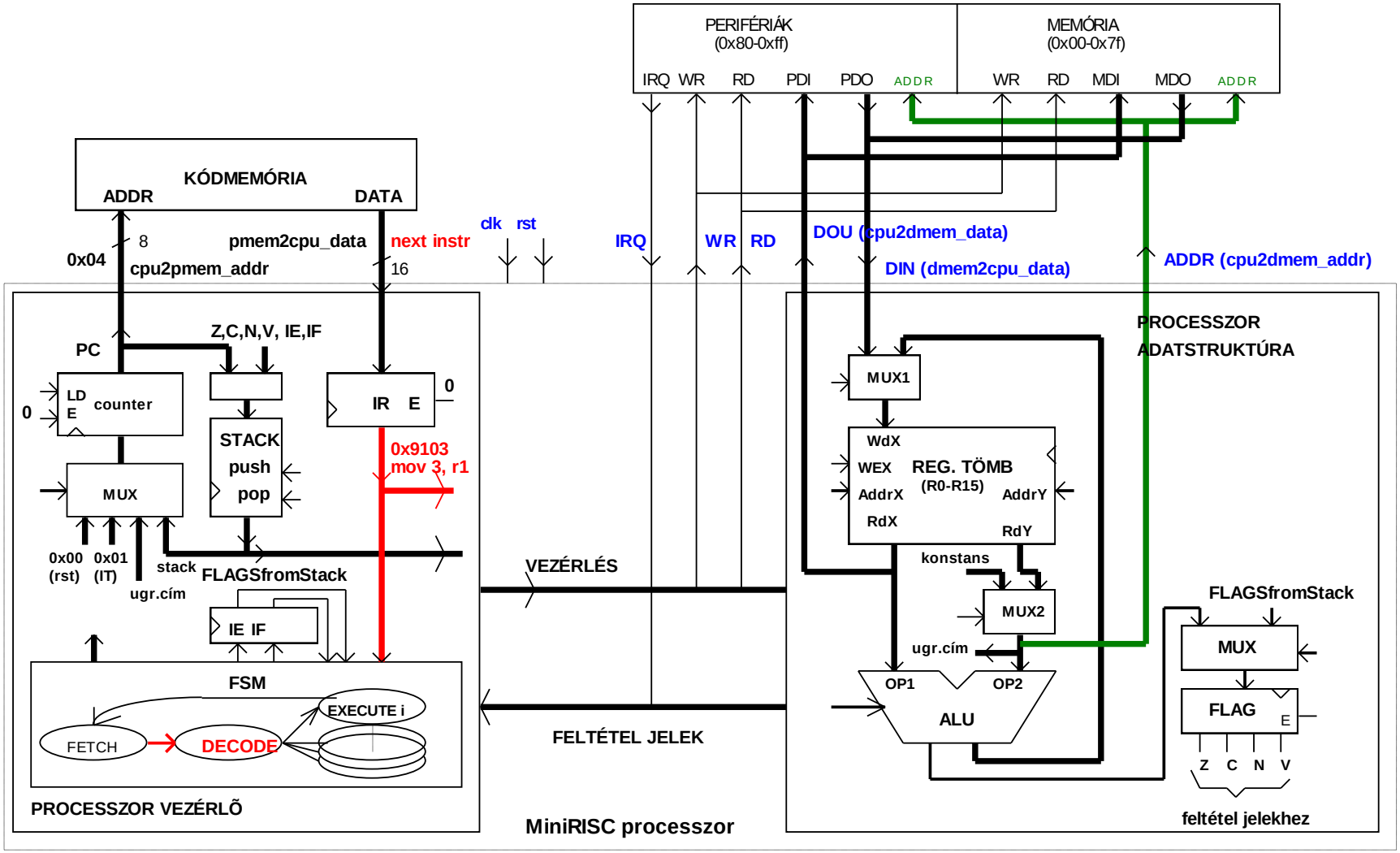
művelet végző utasítás (aritmetikai) összeadás: **EXECUTE** (utasítás végrehajtás) ciklus



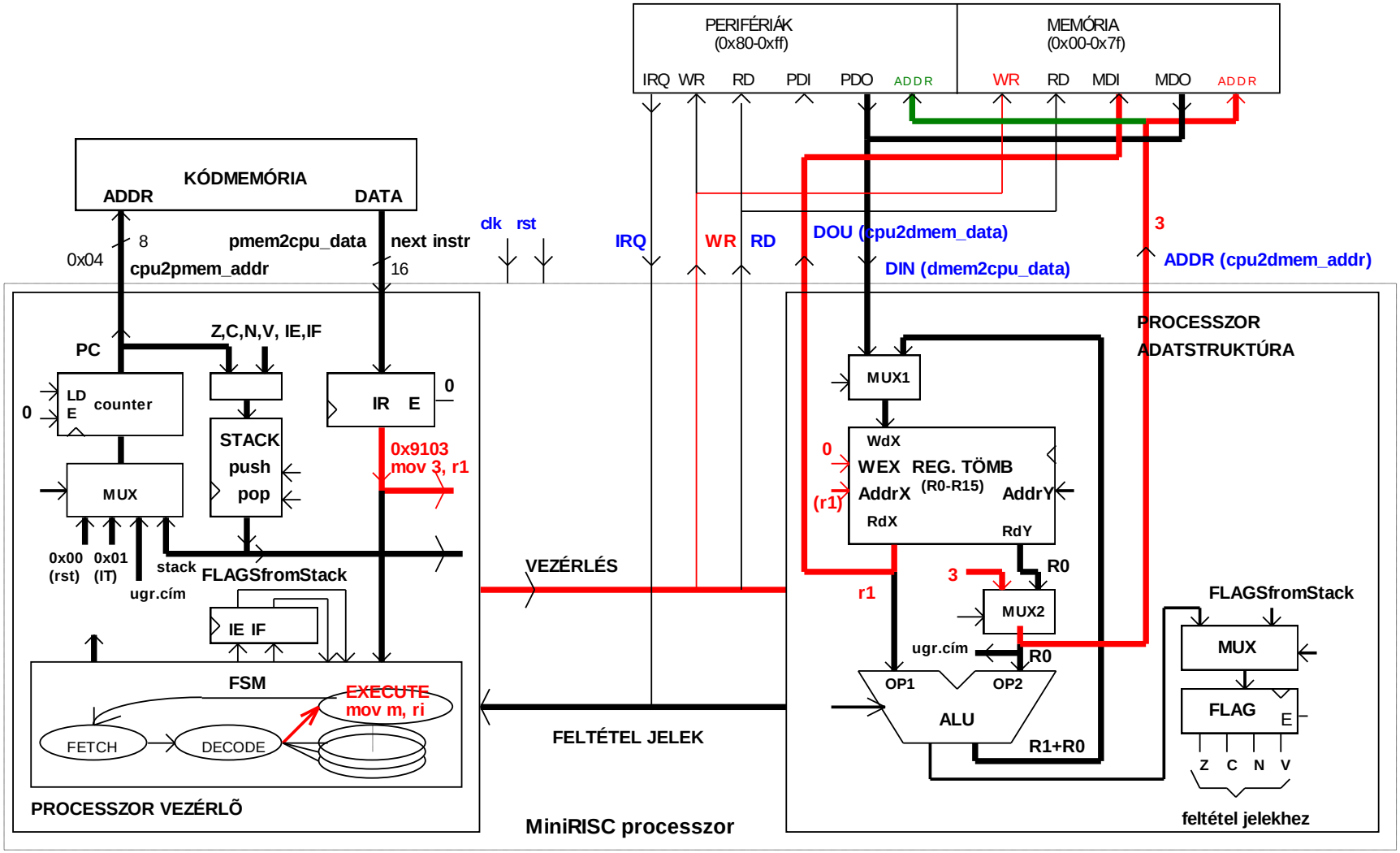
Adatmozgató utasítás, írás a memóriába: **FETCH** (utasítás beolvasás) ciklus



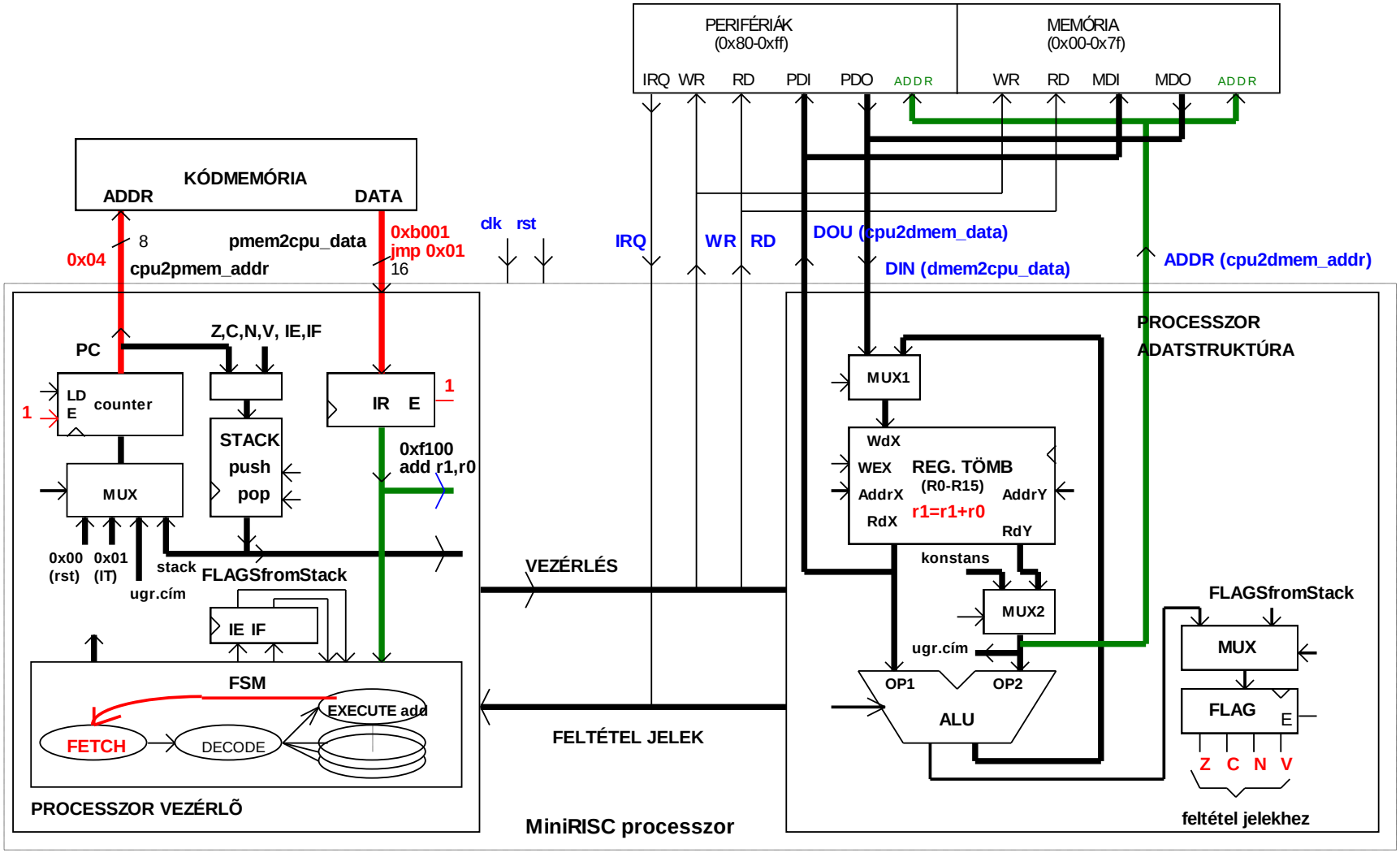
Adatmozgató utasítás, írás a memóriába: **DECODE** (utasítás dekódolás) ciklus



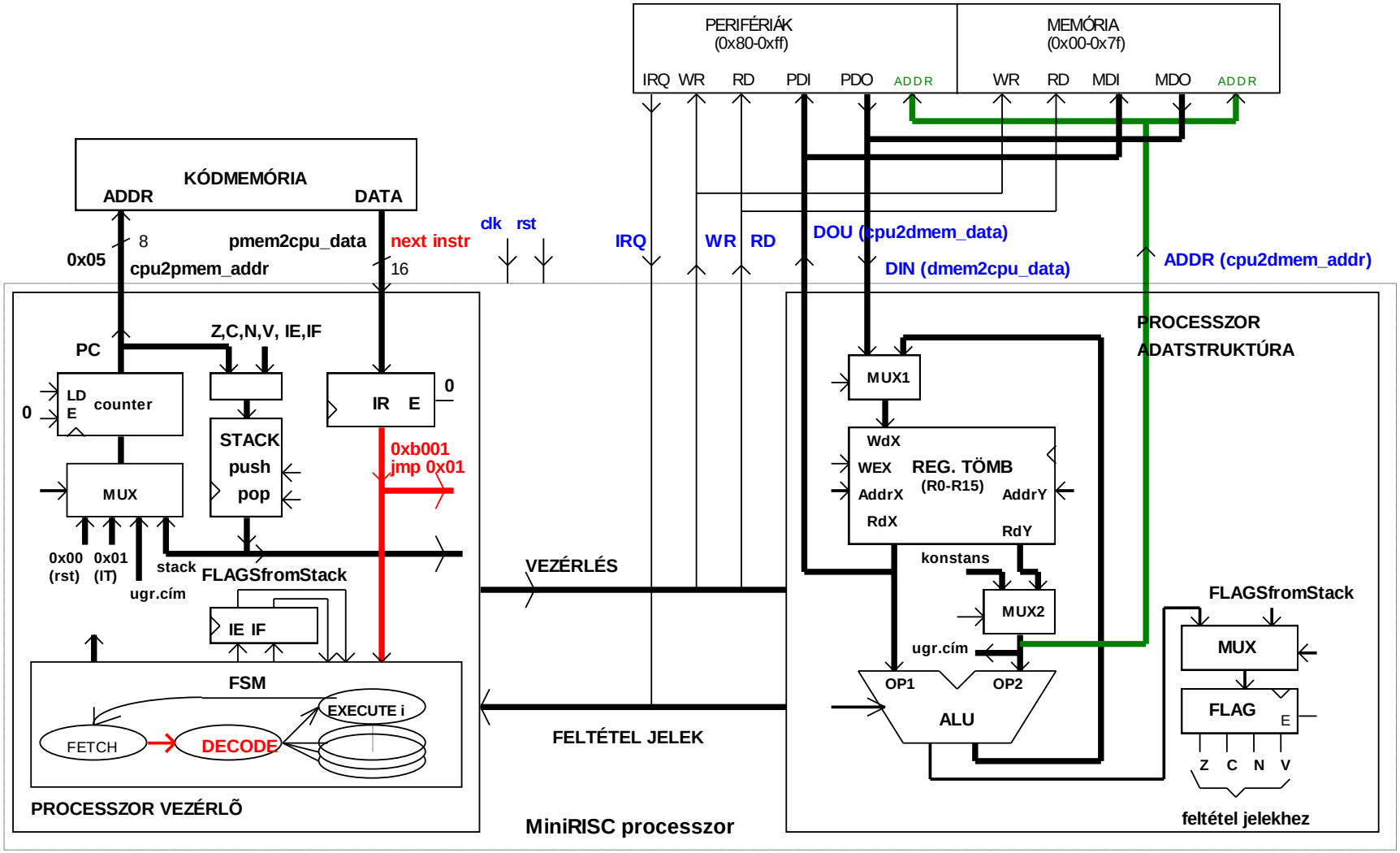
Adatmozgató utasítás, írás a memóriába: **EXECUTE** (utasítás végrehajtás) ciklus



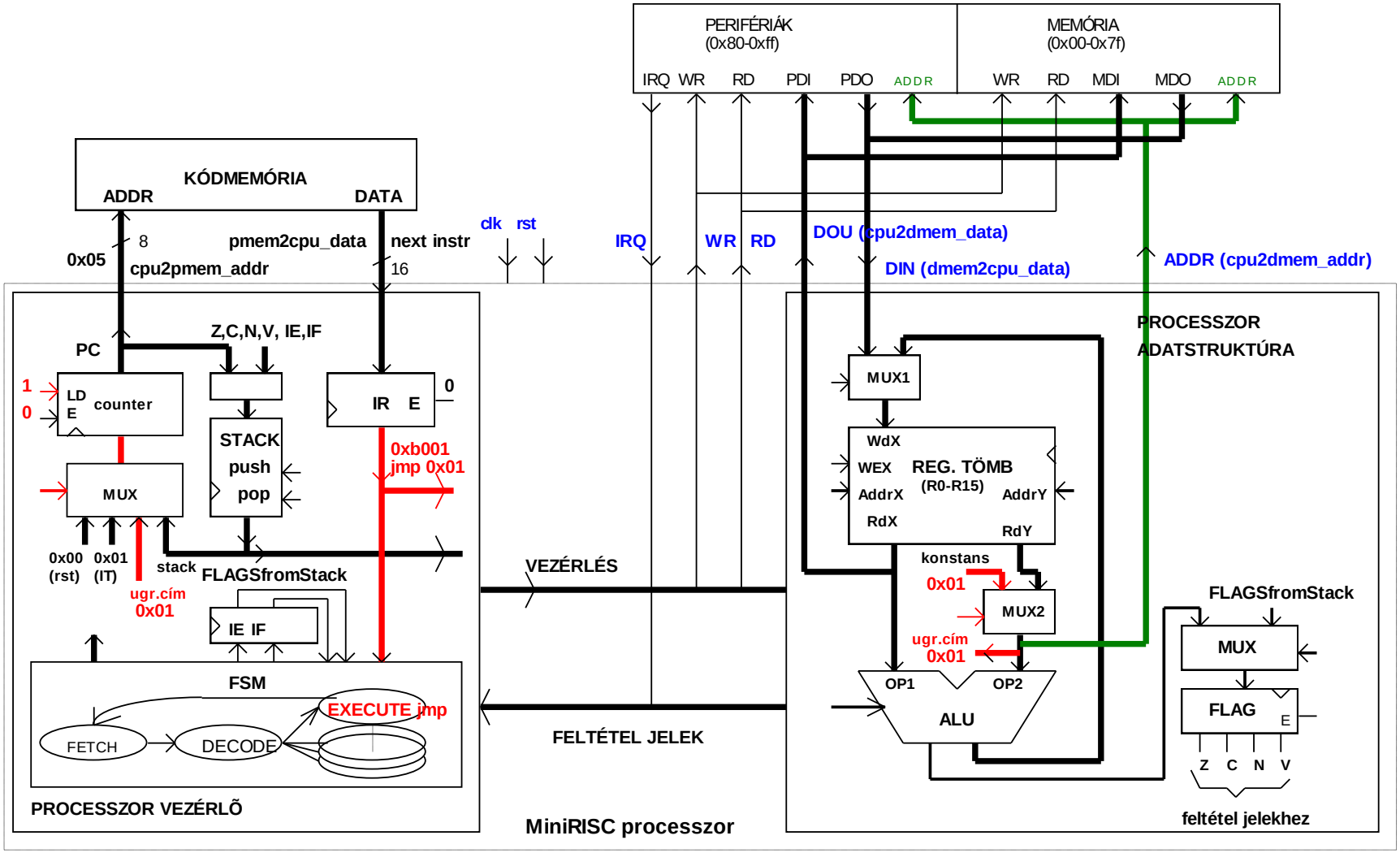
vezérlés átadó utasítás (ugró), direkt ugrás: **FETCH** (utasítás beolvasás) ciklus



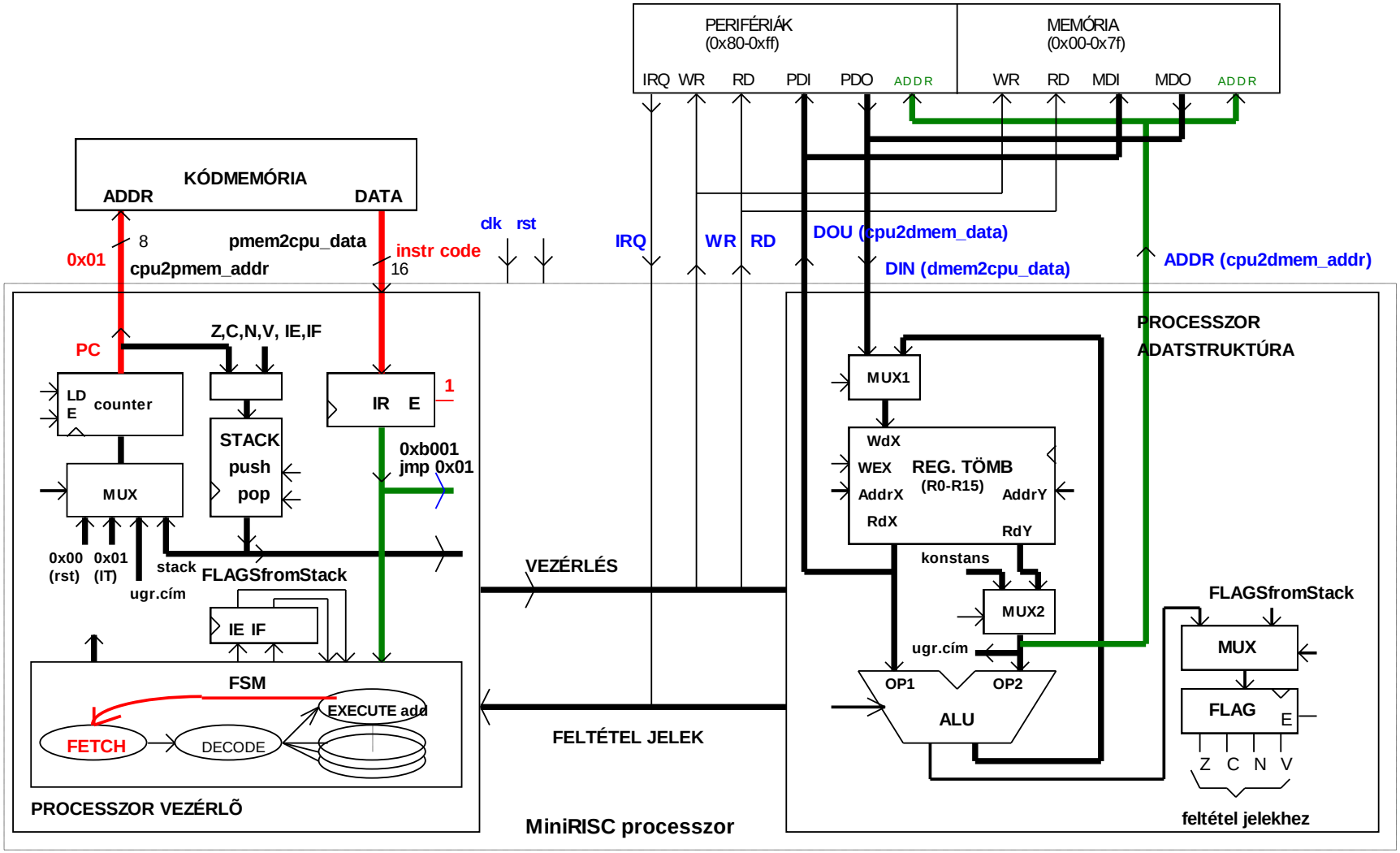
vezérlés átadó utasítás (ugró), direkt ugrás: **DECODE** (utasítás dekódolás) ciklus



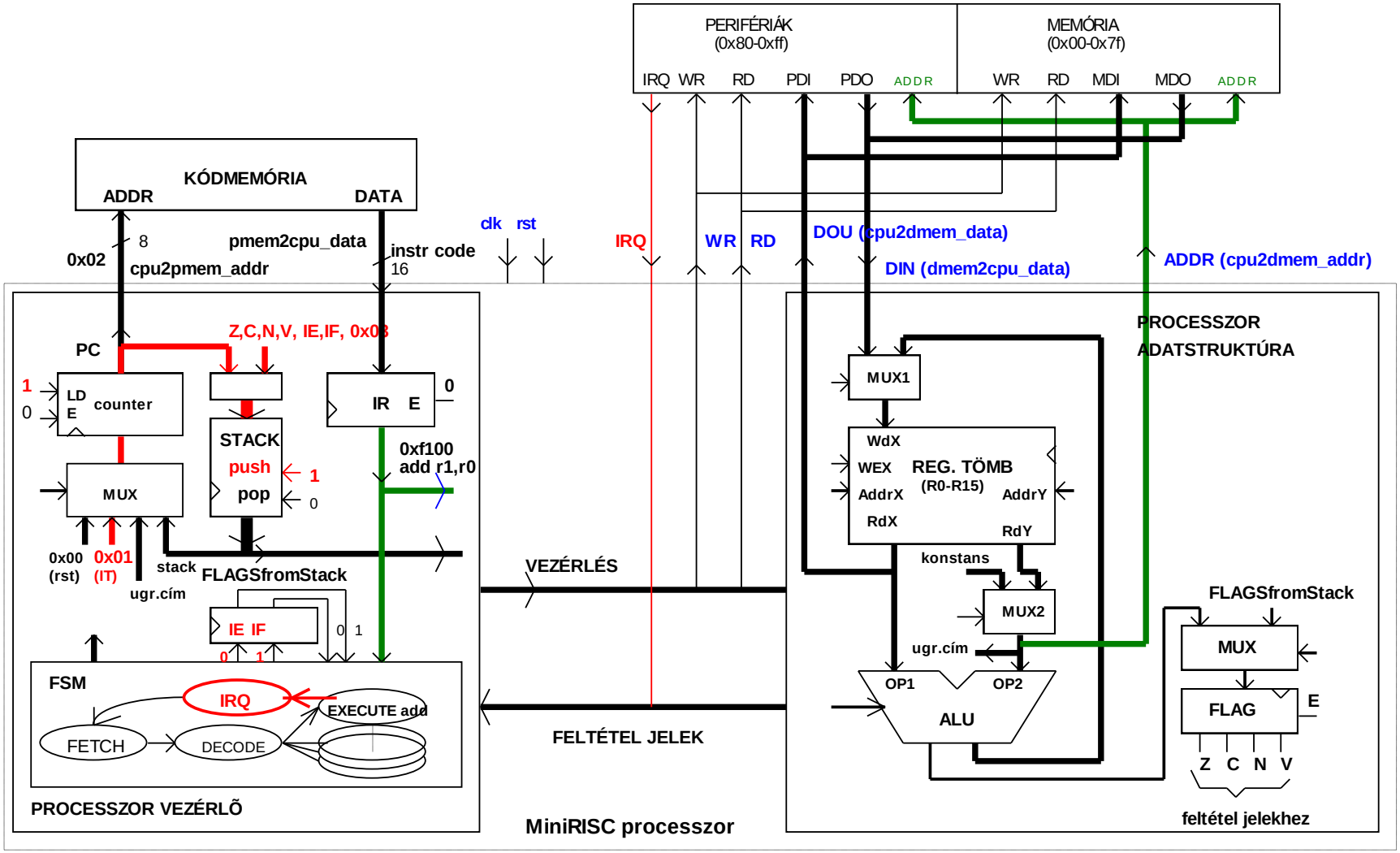
vezérlés átadó utasítás (ugró), direkt ugrás: **EXECUTE** (utasítás végrehajtás) ciklus



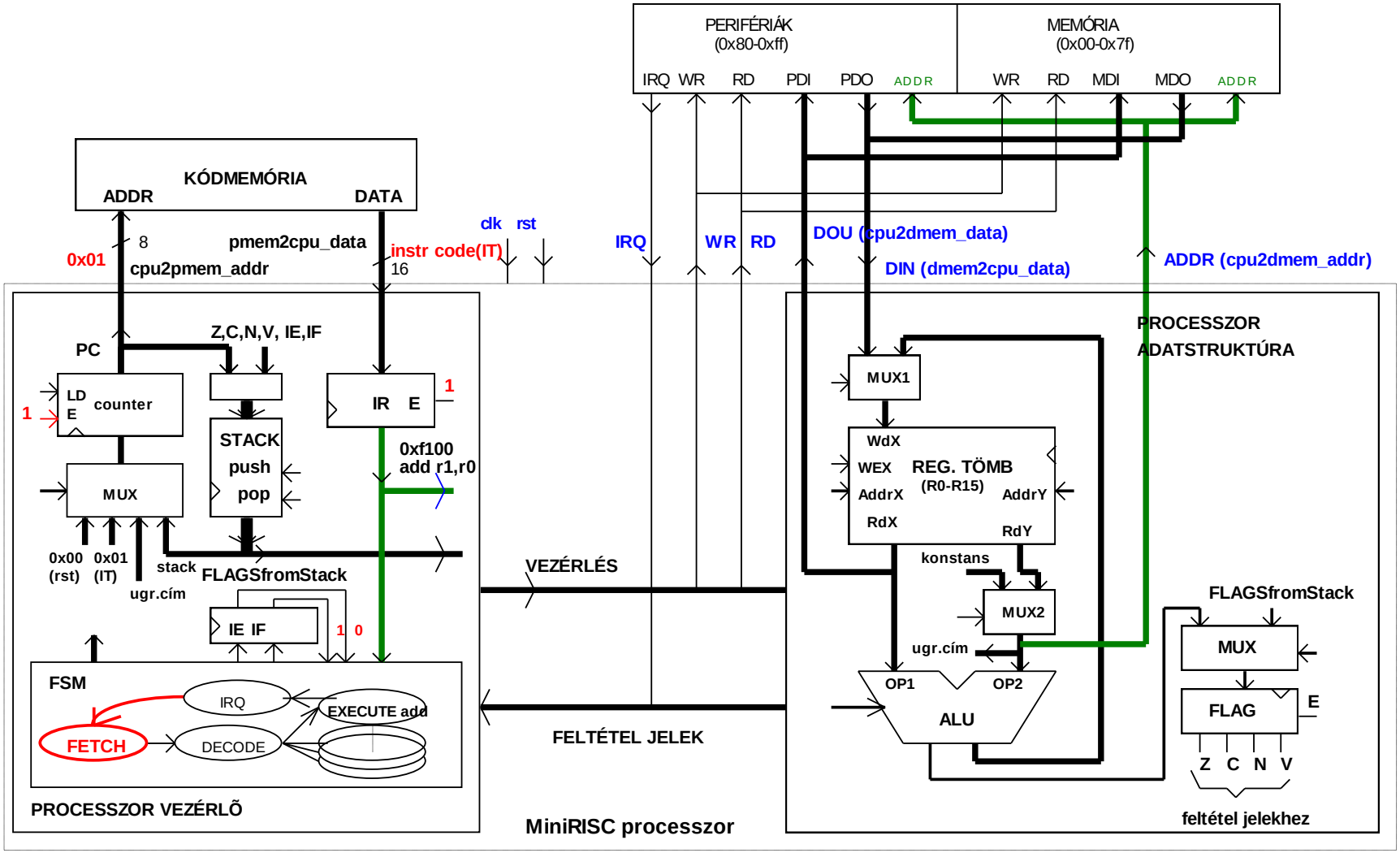
Ugrás végrehajtása uán a PC tartalma (a következő utasítás címe) az ugrási cím



Interrupt kérés hatására (IRQ)**: interrupt elfogadás** ciklus (az FSM IRQ állapotba került)



interrupt elfogadás ciklusból a FETCH-be lépéskor a PC-be az interrupt első utasításának címe kerül (Mini RISC processzornál ez mindig 0x01), a stack-be bekerül a megaszakított utasítást követő utasítás címe, az IT engedélyezés (IE) letiltódik, az IF flag 1 lesz.



Minta program

A memóriában levő 2 adat beolvasása, összeadása, az eredmény kiírása a memóriába.

```
   CODE

   mov  r0, data1        ; REG[0] = DMEM[0]
   mov  r1, data2        ; REG[1] = DMEM[1]
   add   r1, r0          ; REG[1] = REG[1] + REG[0]
   mov  result, r1       ; DMEM[3] = REG[1]
loop:
   jmp loop

   DATA
data1:
   DB 0x20
data2:
   DB 0x14
result:
   DB 0x00
```