Két darab 16 bites előjeles szám szorzása

Nyitrai Bence

A feladat megfogalmazása:

Belső memóriában lévő két darab 16 bites előjeles szám szorzása, túlcsordulás figyelése. Az eredmény is 16 bites előjeles szám legyen, a túlcsordulás ennek figyelembevételével állítandó. Bemenet: a két operandus és az eredmény kezdőcímei (mutatók). Kimenet: eredmény (a kapott címen), OV

Az implementált algoritmus:

A feladatot megvalósító algoritmus 3 fő részre osztható:

1. bemeneti kettes komplemens operandusok előjel vizsgálata, konverziója és a végeredmény előjelének meghatározása
2. szorzás elvégzése az előjel nélkülire konvertált számokon
3. a végeredmény előjele alapján konverzió negatívra, ha szükséges

1.

Először kimaszkoljuk az adott operandus MSB-jét, így ha az MSB 1-es volt, akkor tudjuk, hogy át kell konvertálni a kettes komplemens számot előjel nélkülire, mivel a szorzást előjel nélküli számokkal fogjuk elvégezni, emellett beállítjuk PSW_F0 vagy F1-et 1-esre, ezzel jelezve a szám negatív voltát.

A konverziót úgy végezzük, hogy megnegáljuk a számot, aztán hozzáadunk 1-et.
A végeredmény előjelét PSW_F0 és F1 összehasonlításából kapjuk meg. Ha azonosak, akkor pozitív, különböző előjelek esetén negatív, F0-át állítjuk a végső eredmény előjelére.

2.

A számok 16 bites volta lévén a szorzást több lépésben tudjuk csak elvégezni. Az algoritmus lényege, hogy a különböző helyiértékű bájtokat összeszorozzuk, majd helyiérték helyesen eltolva összeadjuk őket.

```
        12 34 * AB CD
        29 A4           = 34 * CD
     0E 6A 00           = 12 * CD
     22 BC 00           = 34 * AB
  0C 06 00 00           = 12 * AB
 ------------
  0C 37 4F A4
```

A szorzást úgy végezzük el, hogy először a két operandus alsó bájtját szorozzuk össze (itt a példában 34*CD), az így kapott 16 bites szám alsó bájtja lesz a végeredmény alsó bájtja (A4). Következőnek az első operandus felső bájtját szorozzuk, a második operandus alsó bájtjával (12*CD). A feladat specifikációjából kiderül, hogy a végeredményt csak 16 biten fogjuk ábrázolni, különben túlcsordulunk, így a szorzás végeredményének a felső bájtjának nullának kell lennie, ha el szeretnénk kerülni a túlcsordulást (jelen esetben nem fogjuk, mivel a felső bájt 0E). Hasonló módon elvégezzük az első operandus alsó bájtjának és a második operandus felső bájtjának szorzatát (34*AB).

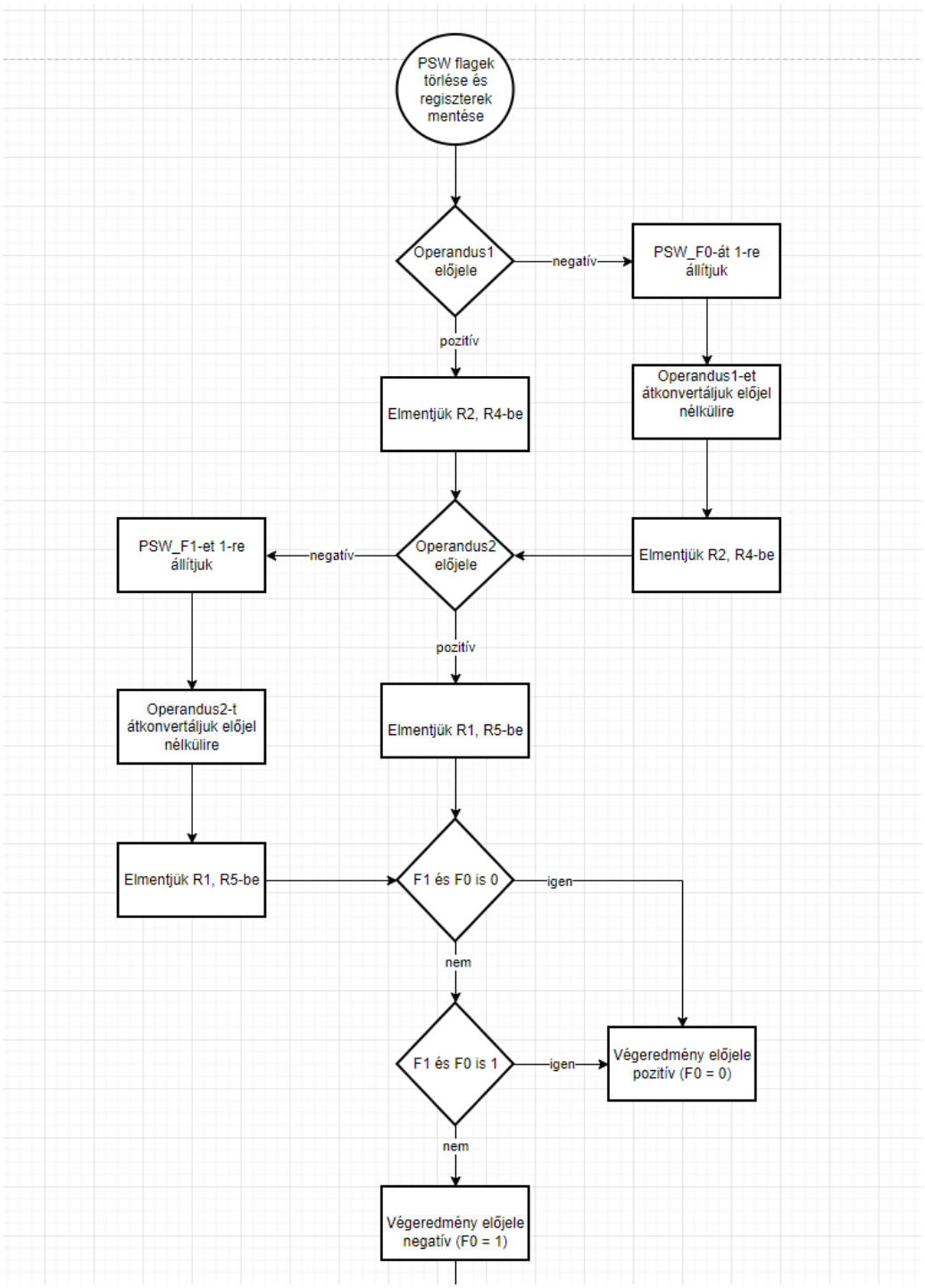
```
29 + 6A + BC =
          14F
```

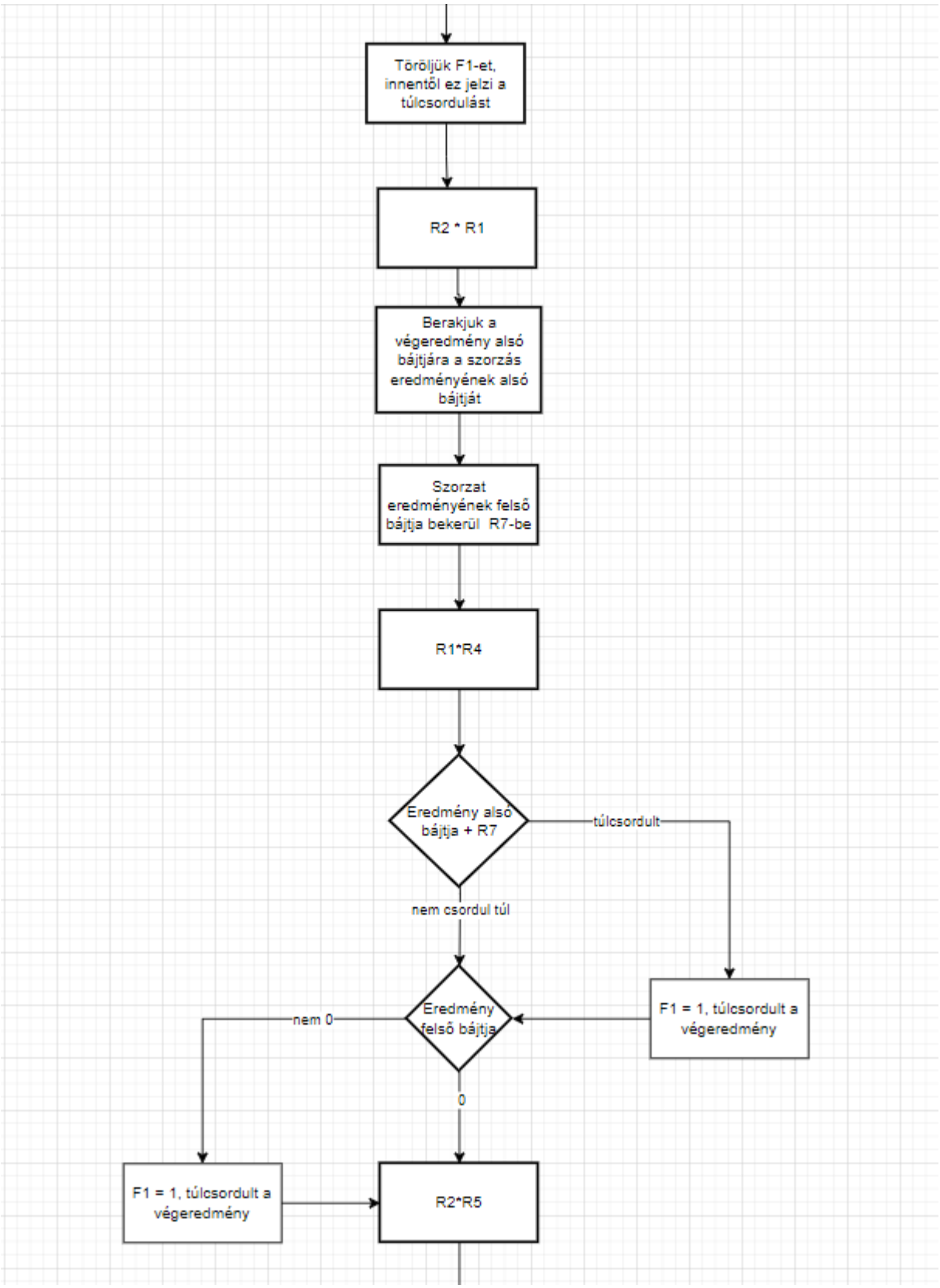
Következő lépésként kiszámoljuk a végeredmény felső bájtját, ezt összeadással fogjuk megtenni. az először elvégzett szorzás eredményének felső bájtja összeadva, a később végzett szorzások alsó bájtjaival (itt látszik, hogy az eredmény túl fog csordulni, mivel 14F nem fér el egy bájton).
Az operandusok felső bájtjainak szorzása leegyszerűsödik egy 0-val történő összehasonlításra mivel, ha a szorzat eredménye nem 0, akkor túlcsordulás történt.
A szorzás elvégzése után a végeredményt beírjuk a memóriába a kapott címre.

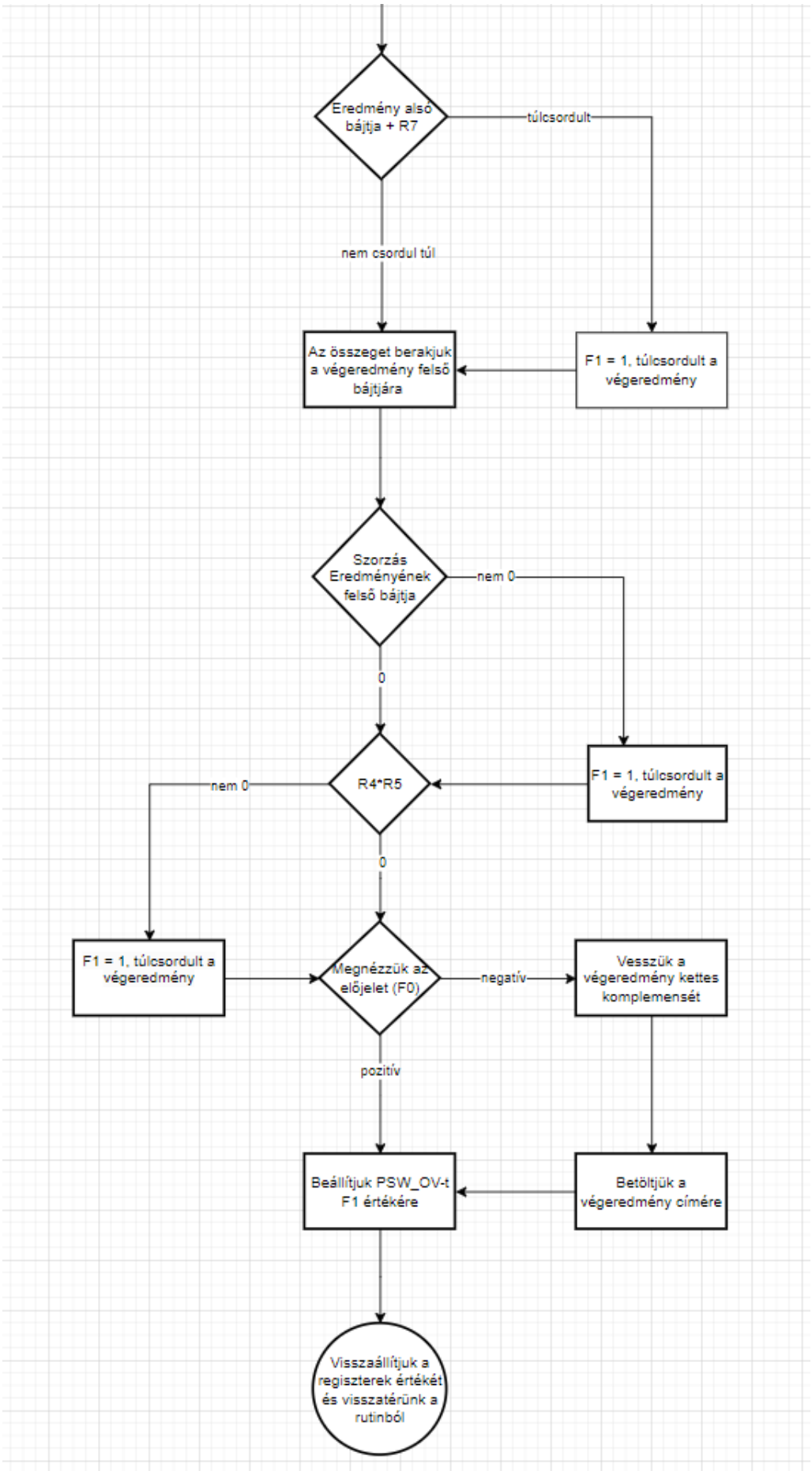
3.

Ha túlcsordulás nélkül kijött az előjel nélküli szorzás eredménye, akkor megvizsgáljuk az 1. pontban beállított előjel bitet. Ha 0, akkor végeztünk. PSW_F0 1-es értéknél megcsináljuk a kettes komplemens konverziót (most a másik irányba), aztán az így kapott számmal felülírjuk a korábban beírt végeredményt.

Folyamatábra:




Töröljük F1-et, innentől ez jelzi a túlcsordulást
R2 * R1
Berakjuk a végeredmény alsó bájtjára a szorzás eredményének alsó bájtját
Szorzat eredményének felső bájtja bekerül R7-be
R1*R4
Eredmény alsó bájtja + R7
túlcsordult
nem csordul túl
Eredmény felső bájtja
nem 0
F1 = 1, túlcsordult a végeredmény
0
F1 = 1, túlcsordult a végeredmény
R2*R5


Eredmény alsó bájtja + R7
túlcsordult
nem csordul túl
Az összeget berakjuk a végeredmény felső bájtjára
F1 = 1, túlcsordult a végeredmény
Szorzás Eredményének felső bájtja
nem 0
0
R4*R5
nem 0
F1 = 1, túlcsordult a végeredmény
0
F1 = 1, túlcsordult a végeredmény
Megnézzük az előjelet (F0)
negatív
Vesszük a végeredmény kettes komplemensét
pozitív
Beállítjuk PSW_OV-t F1 értékére
Betöltjük a végeredmény címére
Visszaállítjuk a regiszterek értékét és visszatérünk a rutinból

Források:

- Mar jegyzet 2.b fejezet
- https://stackoverflow.com/questions/2713972/how-do-i-detect-overflow-while-multiplying-two-2s-complement-integers
- https://pages.cs.wisc.edu/%7Emarkhill/cs354/Fall2008/beyond354/int.mult.html
- https://www.calculator.net/binary-calculator.html?number1=&c2op=x&number2=&calctype=op&x=Calculate