Házi feladat dokumentáció (IMSc)

Feladatkiírás:

Belső memóriában 32 karakteres gyűrűs tároló készítése betételi és kivételi lehetőséggel. Író és olvasó pointer mutatja a következő szabad hely ill. kivehető karakter címét. CY=1 jelzi, ha a tároló betelt. A program két szubrutint tartalmazzon: puffer írása és puffer olvasása. A puffer globális adatterületként állandóan létezik, induláskor üres. Bemenet mindkét rutinnak: tároló kezdőcíme a memóriában (mutató), író és olvasó pointer aktuális állapota, beteendő karakter (írás esetén). Kimenet: kivett karakter (olvasás), pointerek aktuális állapota, CY.

Megvalósítás:

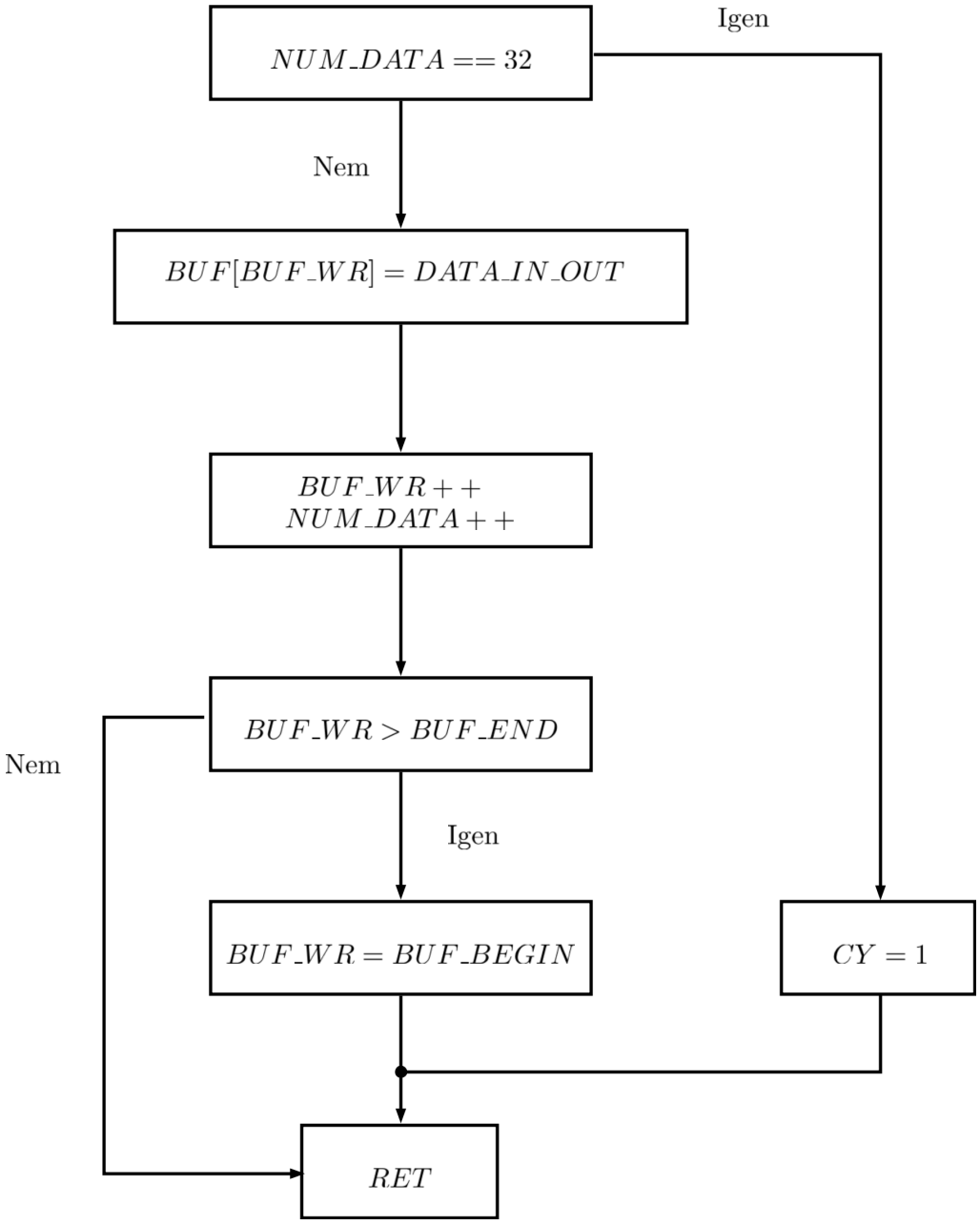
A feladathoz a felhasznált konstans értékek és regiszterek a felhasználó által a program elején található szimbólumokkal személyre szabhatók, így a szubrutin más programban is felhasználható (a konstansok elnevezése, illetve a hívás során módosított regiszterek részletesen megtalálhatók a .asm fájlban található dokumentációban, röviden összefoglalva: BUF_BEGIN, BUF_END a buffer kezdő-, illetve végcímét tartalmazza, BUF_LEN a tároló hossza (azért volt erre külön szükség, mivel a szimulátor nem támogatja a szimbólumokon történő műveletvégzést más szimbólumok definiálása során) BUF_RD, BUF_WR a gyűrűs tároló olvasására, illetve írására szolgáló pointereket tárolja, DATA_IN_OUT az írandó/olvasott karaktert tartalmazza, NUM_DATA a bufferben aktuálisan található byteok számát, míg az IS_EMPTY flag a feladat kiterjesztéseként az üres bufferről ad visszajelzést.

A feladat megvalósítását két szubrutin végzi, a *write_buf* és a *read_buf*, ezek működése a következő.

A *write_buf* szubrutin meghívásakor először a buffer tele voltát ellenőrzi le, ehhez a BUF_LEN és NUM_DATA különbségét képezzük (ebben a sorrendben nem kerül a CY flag feleslegesen beállításra, amire azért kell ügyelni, mivel ez a program szempontjából jelzésértékkel rendelkezik).

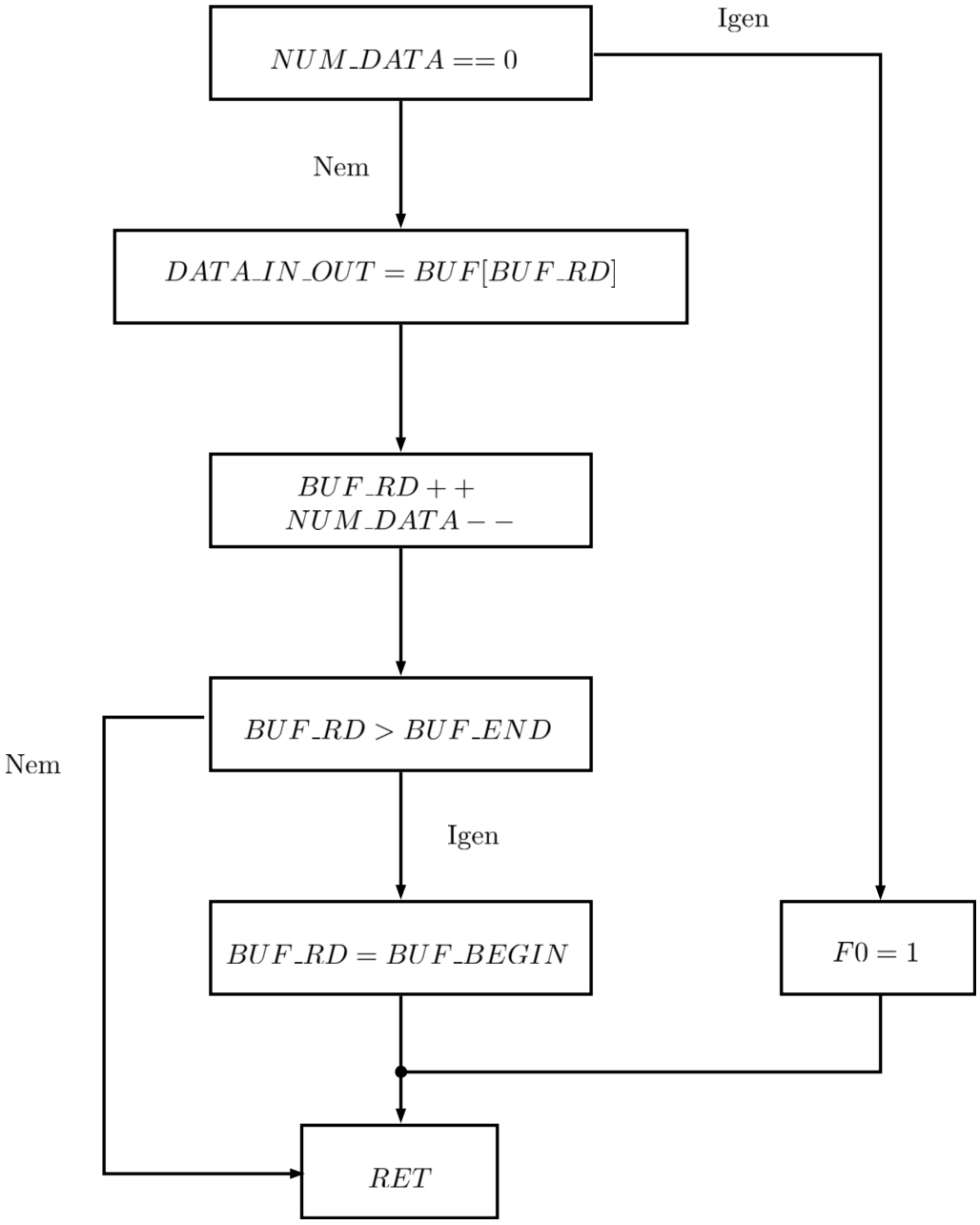
Amennyiben a bufferben van még hely további byte számára, abban az esetben megtörténik az írás, majd az írást végző pointer, illetve a bufferben lévő byteok számának frissítése (előbbi esetében figyelembe véve a cirkularitást is).

A szubrutin visszatérése előtt megtörténik a CY flag beállítása is, abban az esetben, amennyiben a buffer a függvény meghívásakor már tele volt.



A *read_buf* szubrutin a *write_buf* szubrutinnal teljesen analóg módon működik, meghíváskor ellenőrzésre kerül, hogy a buffer üres-e. Ezt követően nem üres buffer esetében DATA_IN_OUT megkapja a beolvasott byte értékét, majd BUF_RD értéke a cirkularitást figyelembe véve megnövelésre kerül, míg NUM_DATA értékét eggyel csökkenti a szubrutin.

Mivel a felhasználó számára nemcsak az az információ a fontos, hogy a buffer tele van-e, hanem az is megkönnyítheti a cirkuláris buffer kezelését, hogyha közvetlen visszajelzést ad az olvasást végző szubrutin az üres bufferről, ez jelen esetben az IS_EMPTY aliasszal ellátott F0 flag segítségével történik, közvetlen a RET utasítást megelőzően.



Alulírott, Reizinger Patrik (W5PDBR), polgári és büntetőjogi felelősségem tudatában kijelentem, hogy a Mikrokontroller alapú rendszerek (VIAUAC06) tárgy keretein belül számomra kitűzött IMSc házi feladatot önállóan, a tárgy során a hallgatók rendelkezésére bocsátott szakirodalom, illetve a http://www.edsim51.com/ oldalon elérhető Intel 8051 szimulátor, illetve dokumentáció felhasználásával, nem megengedett segítséget/segédeszközt fel nem használva oldottam meg.

Budapest, 2017. 12. 02.

Reizinger Patrik