

USRT soros kommunikáció a MiniRISC processzoron

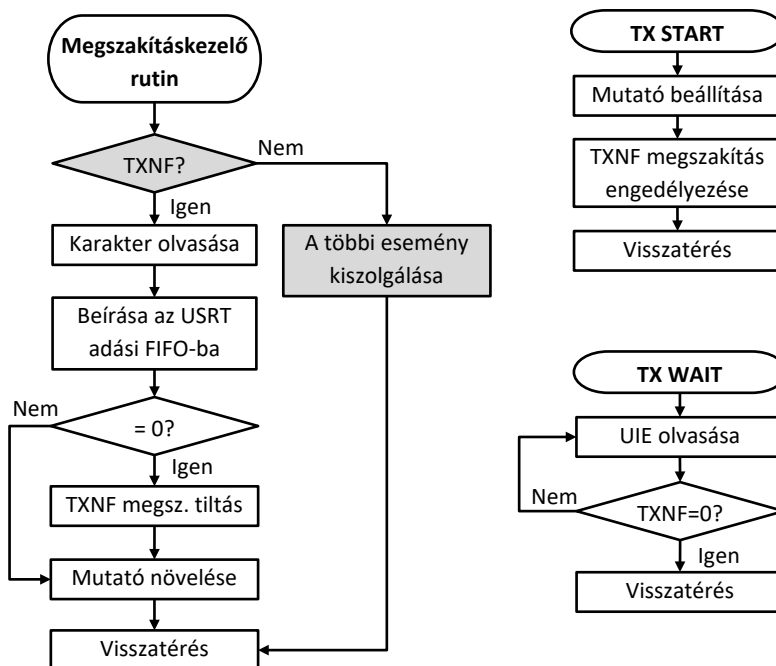
F1. Készítsünk egy programot a MiniRISC processzoros rendszerre, amely az USRT perifériát megszakításosan kezelve elküld egy, az adatmemóriában tárolt 0 végű stringet a soros porton.

- Az adatátvitelhez milyen paraméter(ek) átadására van szükség?
- Hogy tudjuk elindítani az adatátvitelt?
- Hogy tudjuk ellenőrizni, hogy az adatátvitel befejeződött?

Újabb adatot akkor tudunk küldeni az USRT periférián keresztül, ha annak az adási FIFO-ja nincs tele. Tehát az adatküldés megszakításos kezeléséhez az „adási FIFO nincs tele” (TXNF) megszakítást kell engedélyezni. Az ide tartozó megszakításkezelő rutin beolvassa az aktuális karaktert az adatmemóriából, beírja az adási FIFO-ba és a mutatót a következő karakterre állítja. Amennyiben 0 értékű bajtot olvastunk be, akkor végeztünk az adatátvitellel, tehát az „adási FIFO nincs tele” (TXNF) megszakítást le kell tiltanunk. A megszakításkezelő rutin elején a megszakítási esemény vizsgálata csak akkor szükséges, ha több megszakításkérést is használunk. A megszakításkezelő rutin számára két regisztert kell fenntartani, az egyik a munkaregiszter, a másik az indirekt címezéshez használt mutató.

Mivel a string 0 végkaraktere egyértelműen jelzi annak végét, így az adatátvitel indítása előtt a string kezdőcímét kell csak beállítani. Ezután az adatátvitel az „adási FIFO nincs tele” (TXNF) megszakítás engedélyezésével indítható el.

Az adatátvitel befejeződése az „adási FIFO nincs tele” (TXNF) megszakítás tiltott állapotának vizsgálatával ellenőrizhető.



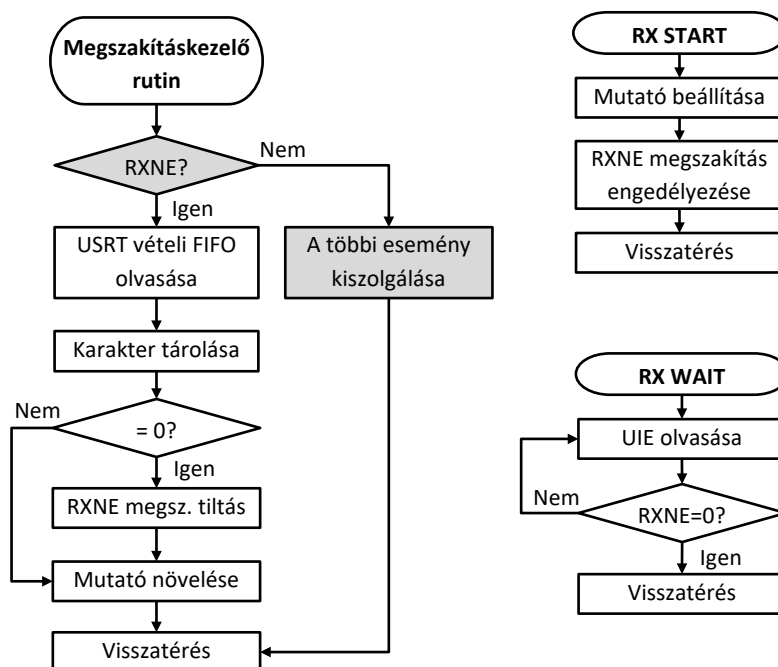
F2. Készítsünk egy programot a MiniRISC processzoros rendszerre, amely az USRT perifériát megszakításosan kezelve a soros portról beolvasson egy 0 végű stringet és ezt az adatmemóriába az *str_rx_buf* címtől kezdve tárolja. Az egyszerűség kedvéért feltesszük, hogy a vett stringek olyan méretűek, hogy biztosan elférnek a rendelkezésre álló vételi pufferben.

- A vételhez az elindítás előtt milyen paraméter(ek) átadására van szükség?
- Hogy tudjuk elindítani a vételt?
- Hogy tudjuk ellenőrizni, hogy a vétel befejeződött?

Újabb adatot akkor tudunk beolvasni az USRT perifériáról, ha annak a vételi FIFO-ja nem üres. Tehát a vétel megszakításos kezeléséhez a „vételi FIFO nem üres” (RXNE) megszakítást kell engedélyezni. Az ide tartozó megszakításkezelő rutin beolvassa a karaktert az USRT perifériáról, beírja az adatmemóriában lévő vételi pufferbe és a mutatót a következő karakterre állítja. Amennyiben 0 értékű bájtot olvastunk be, akkor végeztünk az adatátvitellel, tehát a „vételi FIFO nem üres” (RXNE) megszakítást le kell tiltanunk. A megszakításkezelő rutin elején a megszakítási esemény vizsgálata csak akkor szükséges, ha több megszakításkérést is használunk. A megszakításkezelő rutin számára két regisztert kell fenntartani, az egyik a munkaregiszter, a másik az indirekt címezéshez használt mutató.

Mivel a string 0 végkarakterre egyértelműen jelzi annak végét, így az adatátvitel indítása előtt a vételi puffer kezdőcímét kell csak beállítani. Ezután az adatátvitel a „vételi FIFO nem üres” (RXNE) megszakítás engedélyezésével indítható el.

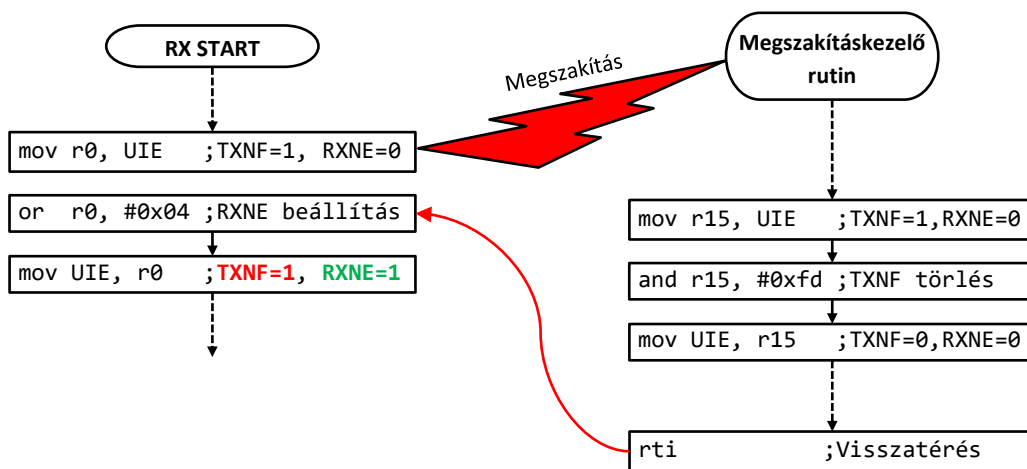
Az adatátvitel befejeződése a „vételi FIFO nem üres” (RXNE) megszakítás tiltott állapotának vizsgálatával ellenőrizhető.



F3. Tegyük fel, hogy a fenti USRT adást és vételt egy programban szeretnénk használni. Milyen probléma adódhat ebből és milyen megoldást tudnánk erre adni?

Ebben az esetben a fent említett két megszakítást egymástól függetlenül kell kezelnünk a programban. A megszakításkezelő rutinban fontos, hogy az esemény vizsgálata előtt a FIFO státuszt maszkoljuk az USRT periféria megszakítás engedélyező regiszterének (UIE) értékével, így nem fogunk kiszorgálni hamis eseményt.

A megszakítások független kezelése miatt a megszakítások engedélyezése és tiltása csak bitműveletekkel (törlés, beállítás) végezhető el. A MiniRISC processzor esetén ezek a műveletek nem atomi műveletek, három utasítás végrehajtása szükséges (adatmemória olvasás, bit törlés (AND) vagy bit beállítás (OR), adatmemória írás). Probléma akkor keletkezik, ha egy ilyen nem atomi műveletet megszakít egy olyan megszakítás kiszorgálás, amely során módosul az USRT megszakítás engedélyező regiszter értéke.



Az ilyen nem megfelelő szinkronizációból adódó hibák teljesen véletlenszerűen jönnek elő, így észlelésük és javításuk nagyon nehéz. A program írásakor már törekedni kell az ilyen hibák elkerülésére. Az egyik lehetséges megoldás, ha a nem atomi műveletek időtartamára letiltjuk a megszakításokat a processzoron. A másik lehetséges megoldás esetén a perifériában alakítunk ki olyan regisztereket, amelyek segítségével a megszakítás engedélyező regiszter bitjei 1 beírásával beállíthatók, illetve törölhetők.