

A feladat kiadása: 2015. 10. 26.

A feladat beadási határideje: 2015. 11.29. 24.00

A HF2 pótlása beadható a pótlási hét végéig, 2015. december 18. 24:00, kölöneljárás díj ellenében!

A feladat beadása elektronikusan, PDF formátumban a <https://hf.mit.bme.hu/> HF beadó portálon.

A feladat beadásához regisztráció szükséges!

Elérhető pontszám: max 15.

A személyes DIGIT kód megtalálható a tárgy honlapján, a Feladatok menüpont alatt megnyitható évfolyamnév sorban!

Digitális technika HF2

Gépi szintű programozás

A Digitális technika HF2 a gép szintű programozás tervezésével foglalkozik. A programozási feladat egy adatvektorban az 1 értékű bitek számának meghatározása (POP_COUNT). Minden hallgató rendelkezik egy egyéni, ún. DIGIT kóddal, amely egy 7 számjegyű számsorozat. Ezt a számsorozatot minden számjegy között kiegészítjük a két számjegy összegének megfelelő hexadecimális számjeggyel és a végén 0-val. Ez a kiegészített sorozat lesz a programozási feladat bemeneti adatvektora. A 14 számjegyű hexadecimális értékeket 2 digitos bájtonként tároljuk az adatmemóriában, a DIGIT_CODE változó által definiált címtől kezdve. Az elkészítendő program feladata az adatvektor elemeinek beolvasásával a benne lévő 1 értékű bitek megszámlálása és az eredmény kiírása bináris formátumban a LED kijelzőre. Ha a minta DIGIT kód az 1234567, akkor a beszűrt értékekkel az alábbi adatokat kapjuk:

1 2 3 4 5 6 7

1 3 2 5 3 7 4 9 5 B 6 D 7 0

A program feladata az 1 értékű bitek számának meghatározása a

{00010011, 00100101, 00110111, 01001001, 01011011, 01101101, 01110000}

56 bites adatvektorban (illetve értelemszerűen a saját DIGIT kód szerinti adatvektorban).

A feladatot 3 különböző algoritmussal kell megoldani. Az elkészült programverziók minőségét a hatékonysággal jellemezzük. A hatékonyság mérőszáma az MEMÓRIA_IGÉNY * VÉGREHAJTÁSI_IDŐ szorzat. A memória igény magába foglalja a PROGRAM és az ADAT memória igényt is, ha az a program végrehajthatóságához tartozik (pl. táblázat). (1 utasítás 16 bit, 1 adatbájt 8 bit.) A bemeneti adatvektor nem számít bele a számításba! A végrehajtási idő a MiniRISC IDE által kijelzett utasításszám a program lefutásának végén.

A 3 részfeladat specifikációja:

HF2_1. Hagyományos algoritmus

A program beolvassa az adatvektor elemeit bájtonként és léptetéssel megszámlálja az egyes értékű bitek számát. Ezeket bájról bájtra összegzi, majd az utolsó bájt feldolgozása után az eredményt kijelzi a LED kijelzőn. A program elkészítése során törekedjen a MiniRISC utasításkészlet tulajdonságainak alapos elemzésére és a legkedvezőbb kódolás használatára.

HF2_2: Táblázatos algoritmus

A program bizonyos részszámításokhoz táblázatos segítséget használ. A részösszegeket tartalmazó táblázatot az adatmemóriában helyezi el, a SUM_LUT címkétől kezdve. A számítás során beolvassa az adatvektor elemeit, ezekből bittartományokat kiválasztva és indexként felhasználva a táblázat hozzátartozó értékét kiolvassa és hozzáadja a teljes összeghez. Az utolsó részösszeget feldolgozva az eredményt kijelzi a LED kijelzőn. A program elkészítése során törekedjen a MiniRISC utasításkészlet tulajdonságainak alapos elemzésére és a legkedvezőbb kódolás használatára.

HF2_3: Aritmetikai algoritmus

A bináris aritmetika speciális tulajdonságait felhasználva hatékony algoritmust fejleszthetünk. A hatékonysága 8 bites ALU esetén nem nyilvánvaló, de nagyobb adatméret esetén egyértelműen előnyös lehet. Az elv az, hogy két érték összegzése esetén maximum 1 átvitel bit képződik. Tehát, ha a rész bittartományok között van legalább egy-egy bit pozíció „rés”, akkor a sokbites összeadónk egyetlen ciklusban több kisebb bitszámú összeadást is képes elvégezni, azaz párhuzamos műveletvégzést végezhet. Ez az ún. SIMD műveletvégzés (Single Instruction Multiple Data), azaz ugyanazt a műveletet egyszerre több részadaton végezzük el. Az algoritmus a következőképpen működik: A beolvasott adatelemet és másolatát maszkoljuk páros-páratlan bitekre. Ezután a páratlan biteket tartalmazó bájtot jobbra léptetjük. Ekkor mindkét regiszterben csak a páros bitpozíciókban lehet adatbit, a páratlan bitpozíciók értéke biztosan 0. Ha összeadjuk a két értéket, akkor keletkeznek 2 bites részösszegek, amelyek értéke 0, 1 vagy 2, attól függően, hogy az adott páros bitpozícióban sehoegyik adatrészben sem, illetve csak egyik vagy mindkét részben szerepelt 1 értékű adatbit. Ezután a 4 db 2 bites részösszeget tartalmazó eredményt és másolatát újra maszkoljuk, de most már 2 bites mezők szerint. Ahol az alsó 2 bit 0 lett a maszkolás után, azt az adatot kétszer jobbra léptetjük. Ezután a két adatot összeadjuk, az eredmény 2 db 3 bites részösszeg az alsó és a felső 4 biten. A következő lépés talán már mindenki számára magától értetődő. Ha a processzor utasításkészlet képes egy lépésben több bites léptetésre is, akkor egy adat feldolgozása „n” bites adatméret esetén $5 \cdot \log_2(n)$ (COPY, MASK, MASK, SHIFT, ADD) azaz pl. 64 bites processzoron 30 utasításciklus és az alsó 7 bit tartalmazza az eredményt. Az egyes adatelemek részeredményeit természetesen külön összegeznünk kell. Az utolsó részösszeget feldolgozva az eredményt jelezze ki a LED kijelzőn. A program elkészítése során törekedjen a MiniRISC utasításkészlet tulajdonságainak alapos elemzésére és a legkedvezőbb kódolás használatára.

DOKUMENTÁCIÓ

A feladat formai követelményeit legkönnyebben a mintamegoldás stílusát követve lehet betartani.

Az első oldal összefoglalja az azonosító adatokat (Név, NEPTUN, Tankör, Gyak. Kurzus, DIGIT kód).

Nyilatkozat:

A feladatokat önállóan, meg nem engedett segédeszközök használata és mások közvetlen közreműködése nélkül oldottam meg:

.....
aláírás

Az aláírás egy elektronikusan reprodukált kézi aláírásképp legyen. A közvetlen közreműködés tiltása nem az előzetes konzultációs segítség igénybevételének tiltását jelenti, hanem azt az elvárást fogalmazza meg, hogy a feladatot a szerző készíti el. A meg nem engedett segédeszköz olyan esetleg mások által készített általános „generátor” program, ami akár csak a részfeladatok megoldásait is automatikusan előállítja. Célunk a HF kiadásával a tárgy tananyagának elsajátítását segíteni, és ehhez kérjük a korrekt hallgatói közreműködést, mert csak így biztosítható a tervezett cél elérése.

A feladatokat tehát az előírások értelmében önállóan, egyéni munkában kell elkészíteni. Házi feladat elkészíttetése mással a kari szabályozás értelmében minimálisan elégtelen minden érintettnek, részletesen az előírások megtalálhatók: <https://www.vik.bme.hu/szabalyzatok/123.html>.

Az egyes részfeladatok ismertetésénél kérjük egy-két mondatban leírni az előzetes tervezői észrevételeket, a későbbiekben használt megoldások választásának indoklását, és a kisebb részletek ASM folyamatábráját. A dokumentumba kérjük bemásolni a programok forráskódját. A forráskód minden utasítása tartalmazzon megjegyzést, a működés megérthetősége érdekében. Minden részfeladat végén kérünk egy-két mondatos összefoglalót az elvégzett munkáról (vélemény, problémák felsorolása, stb.).

A teljes dokumentáció az előlapot is beleértve ne legyen hosszabb 4 oldalnál, azaz részfeladatonként maximum 1-1- oldalt várunk.

A terveket a laboratóriumi gyakorlatokon is használt MiniRISC IDE fejlesztési környezetben kérjük elkészíteni, akár szimulációs módban, akár valódi hardveren futtatva. A MiniRISC IDE program a tárgy honlapjáról letölthető.

Kérdések, problémák esetén a tárgy előadóival, a gyakorlatvezetőivel, ill. a laboratóriumi mérésvezetőkkel javasolt konzultálni.

A beadási határidő a MIT HF portál szerint mért idő. Ezt fogadjuk el érvényes időnek. A határidő után beadott feladatok is kaphatnak teljes értékű pontszámot, a késedelem negatív következménye pusztán a félév végén kirótt különjárás díj befizetési kötelezettség (1800Ft/feladat).

Budapest, 2015. október 26.

A tárgy oktatói