

Memória interleave

1. feladat

50ns hozzáférési idejű memóriák esetén legalább **hányszoros interleave** megoldást kell alkalmazni, ha 12ns átlagos elérési időt szeretnénk megvalósítani? **Használható-e** a módszer direkt leképzés esetén?

2. feladat

Egy interleaving **megvalósításnál** az Op. memória hozzáférési ideje **46ns**, mérete **1024Mbyte**, a **cache** memória egy sora 256 bites, kapacitása: **32Kbit**, hozzáférési ideje **6ns**.

Minimum hány db (hányas interleave szükséges), **és mekkora** méretű, **milyen szervezésű** memória áramkörből kell létrehozni az operatív memóriát, hogy egy blokk behozatalnál a látszólagos memóriahozzáférési idő **6ns** legyen?

Kapacitás problémák kezelése - Indexelt leképezés

3. feladat

Indexelt leképzés esetén a logikai cím **16** bites, amelyből a legmagasabb helyértékű **4** bit az index. A fizikai memória mérete **1Mbyte**.

Számítsa ki mekkora az indexregiszter-tömb mérete, ha vezérlésre **2** bitet alkalmaz.

Adott egy **1000H kezdőcímre** címfolytonosan lefordított **12KB** méretű program. A fizikai memóriában az **512KB..520KB** (80000H-81FFFH) és az utolsó **4KB** tartományban (FF000H-tól) van szabad memóriahely.

Írja fel az indexregisztertömb –**a program futásakor felhasznált**– regisztereinek sorszámát, hexadecimális tartalmát, ha a legfelső helyértéken alkalmazott vezérlő bitek értéke **01**!

Mit kell biztosítani, és hogyan, ha a regisztertömböt író és a megszakítás kezelő programok logikailag és fizikailag az első 4kB-on helyezkednek el?

4. feladat

Egy 16 bites logikai és egyben fizikai címmel rendelkező rendszerben (pl.: 8085) indexelt leképezésű (index regiszter tömböt alkalmazó) memóriaszervezéssel 1Mbyte-ra kell bővíteni a fizikai memória méretét. Az index regiszter-tömb 32 regisztert tartalmaz. Vezérlésre (adminisztrációra) 3 bitet használunk).

Hány bites az offset (eltolás)?

Hány bites a regiszter tömb egy regisztere?

Hány blokkot (lapot) tartalmaz a teljes memória?

Maximálisan hány blokk (lap) lehet egyszerre aktív?

5. feladat

Egy indexelt leképezést alkalmazó számítógép címsínje **16 bites**, ebből a **3 MSB** az index. Az operatív memória **256 kByte** méretű.

Milyen az indexregiszter tábla szervezése (szó x bit) ha vezérlési célokra **2 bit** szükséges?

Ha egy 1980h byte hosszú programrészt a hexa 8080h kezdőcímre fordítunk le, **hányas indexű** indexregiszter(ek) tartalmát kell beállítani a program futtatásához?

Milyen (hexa) értéket kell ebbe az indexregiszterbe írni, ha a fizikai memóriában 2A000h-2BFFFh tartományban van üres hely a program számára és a vezérlő bitek zérus értékűek?

Processzorok teljesítményének növelése

A gyakorlatra való felkészülés során ajánlott az alábbi témakörök átismétlése az előadás vázlatok alapján: szinkron- és aszinkron ütemezésű pipeline működése, Amdahl törvénye, paraméter átadás szubrutin hívás során.

6. feladat

Rajzolja fel egy három elemi műveletvégzőt tartalmazó, az eltérő ciklusszámú és végrehajtási idejű utasítások esetén alkalmazható, puffertárolót tartalmazó **ASAP** ütemezésű feldolgozó szalag (Pipe-line) blokkvázlatát. **Rajzolja be** az ütemezéshez használt jeleket is.

Milyen utasításrendszernél lehet célszerű az ASAP ütemezés?

Milyen jellegű túlterhelést képes kezelni a puffertároló?

Szinkron ütemezéssel összehasonlítva melyik ütemezésnél valószínűbb a **nagyobb lappangási idő növekedés?**

7. feladat

Egy **szinkronütemezésű** feldolgozó szalagot alkalmazó processzor három elemi műveletvégzőt tartalmaz. Az első elemi műveletvégző végrehajtási ideje 50ns, a második 30ns a harmadik 50ns a részeredmény áttöltéshez szükséges időt elhanyagoljuk.

Hány ns alatt hajtódna végre egy utasítás pipe line alkalmazása nélkül?

Hány ns alatt hajtódik végre három utasítás a pipe-line működésekor (a feltöltődés után)?

Mekkora az újraindítási idő a fenti esetben?


Mekkora a szalag sebességnövelő hatása a fenti esetben?

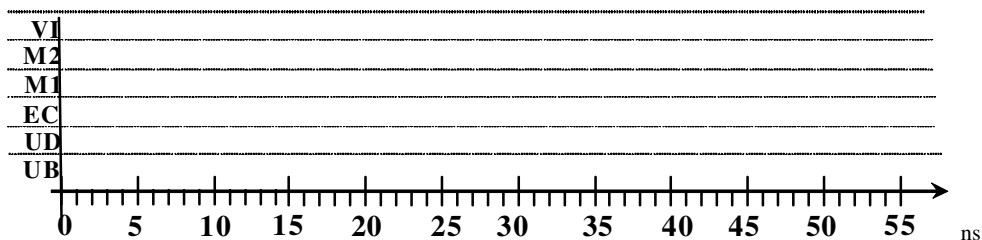
Mekkora az utasításkibocsátás a fenti esetben /mértékegység is/?

Mekkora a szalag hatékonysága?

Milyen módszerrel csökkenthető az elemi műveletvégző egységek időben egyenlőtlen terheléséből adódó probléma?

8. feladat

Rajzolja be az alábbi ábrába az utasítások végrehajtását **55ns**-ig, 1 ns-os felbontásban (pontossággal), **szinkronütemezésű**, hat elemi műveletvégzőt tartalmazó pipe-line esetében, **ha buborék** (üres részművelet) **beszúrással** oldjuk meg azt, hogy a 2. utasítás V_i művelete és az 5. utasítás EC művelete ugyanazt az egységet használná. A műveletvégzők **üres idejét**  jelzéssel jelölje. A **műveletvégzők** működési ideje már tartalmazza a **0,5ns áttöltési időt is**. A műveletvégzők működési idejei a következők: **UB=5ns, UD=4ns, EC=4ns, M1 és M2=5ns, VI=4ns** (Jelölje, hogy a műveletvégző hányadik utasítás melyik elemi műveletét hajtja végre).





Mekkora a fenti rendszer hatékonyságának értéke?

Adja meg az ötödik utasítás lappangási idejét.

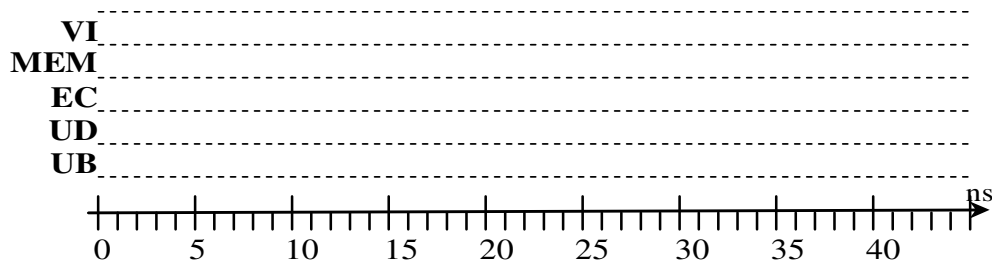
9. feladat

Rajzolja be az alábbi ábrába az utasítások végrehajtását 44ns-ig 1 ns-os felbontásban (pontossággal) **ASAP (aszinkron) ütemezésű** öt elemi műveletvégzőt tartalmazó pipe-line esetében

A műveletvégzők **üres idejét** (várakozás az előző eredményére)  jelzéssel, a **foglaltságot** (következő műveletvégző felszabadulására várás)  jelzéssel jelölje. **A műveletvégzők működési ideje már tartalmazza az áttöltési időt is.**

A műveletvégzők működési idejei a következők: **UB=5ns, UD=4ns, EC=5ns MEM=10ns VI=4ns**

(Jelölje, hogy a műveletvégző hányadik utasítás melyik elemi műveletét hajtja végre).



Adja meg a második utasítás lappangási idejét.

Mekkora az utasítás kibocsajtás (mértékegységgel)?

Hogyan lehetne ezt növelni?

10. feladat

Egy CISC utasításrendszer utasításainak **80%-a** megvalósítható **egyszerűbb RISC** utasításként. Ezek végrehajtására az eredeti idő fele is elegendő. A maradék 20% utasításokat az így megvalósított utasításokból átlagosan 3 felhasználásával helyettesítik. Amdahl módosított törvényével erősítse meg, hogy a feladat célszerűen végrehajtható, sőt még gyorsítást is elérünk. **Adja meg az elérhető gyorsítás kiszámításának menetét és az értékét is 2 tizedes pontossággal.**

Megoldások

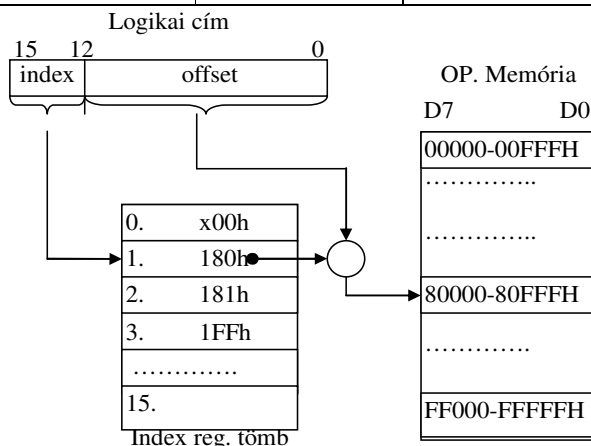
1. $50/12 = 4.12 \rightarrow 5x$ átlapolás kell
 Igen használható, mert az interleave alkalmazása független a cache belsejében alkalmazott leképezéstől.

2. Memória áramkörök száma: $\left\lceil \frac{46}{6} \right\rceil = \lceil 7,66 \rceil = 8$ db

Memória áramkörök mérete: $1GB/8 = 8$ darab **128MB**
 szóhossza: 256 bites sormérethez 8db memória esetén $256/8=32$ bites szóhossz kell memóriánként

3. Mivel 4 bites index \rightarrow 16 db regisztert tud kiválasztani
A címzés offset részének bitszáma: $16-4 = 12$ bit
 Mivel a fizikai memória címzéséhez 20 bitre van szükség, ezért az indexregiszterben $20-12 = 8$ címbitet kell tárolni. Ehhez jön hozzá a két vezérlő bit.
Egy regiszterbejegyzés bitszáma: $8+2 = 10$ bit
Az indexregisztertömb mérete (regiszter x bit): 16×10 bit

Regiszter sorszám	Bin. érték	Hexadecimális érték
1	01 1000 0000	180
2	01 1000 0001	181
3	01 1111 1111	1FF

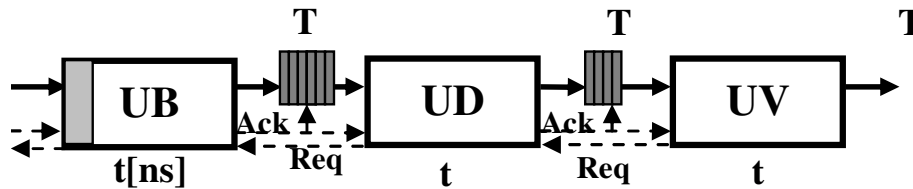


A 0. sorszámú regiszterbe xx 0000 0000 értéket kell beírni. A későbbi működés során pedig tiltani kell ezen indexregiszter bejegyzés módosítását.

4. **Az offset rész bitszáma:** $16-5 = 11$ bit
A regiszter tömb egy regiszterének bitszáma: $20-11+3 = 12$ bit
A memóriában tárolt blokkok száma: $2^{(20-11)} = 512$
Az egyszerre aktív lapok száma: 32 (mivel ennyi regiszter van, ennyi különböző lap lehet egyszerre aktív)

5. **A regiszter tömb szervezése:** 8×7 bit
A programhoz tartozó regiszter index: 4
A programhoz tartozó regiszter értéke: 15h (00 10101)

6. ASAP ütemezésű aszinkron pipeline

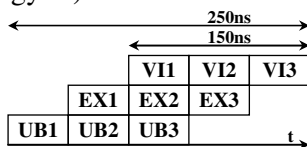


CISC utasításrendszernél célszerű alkalmazni az eltérő végrehajtási idők miatt. A pufferekkel átmeneti túlterhelést lehet kezelni. A lappangási időt nézve az szinkron ütemezésnél nagyobb lehet, mert szinkron esetben a léptetés idejét mindig a leglassabb fokozathoz méretezzük.

7. Ha nincs pipeline, akkor $T_{ut} = 50+30+50 = 130ns$

Szinkron ütemezés esetén (feltöltődés után) $T_{ut} \rightarrow 3 \times 50ns = 150ns$

Ha a feltöltődést is figyelembe kell vennünk, akkor 3 komplett utasítás végrehajtása (lappangással együtt) 250ns



Az újraindítási idő megegyezik a leglassabb művelet idejével: $T_{ui} = 50ns$.

$$S_u = \dots \frac{T_{ut}}{T_{pl}} = \frac{N \cdot T_{ut}}{N \cdot T_{ui} + (3-1) \cdot T_{ui}} \quad \text{Ha } N \rightarrow \infty \quad S_u \cong \frac{T_{ut}}{T_{ui}} = \frac{130}{50} = 2.6$$

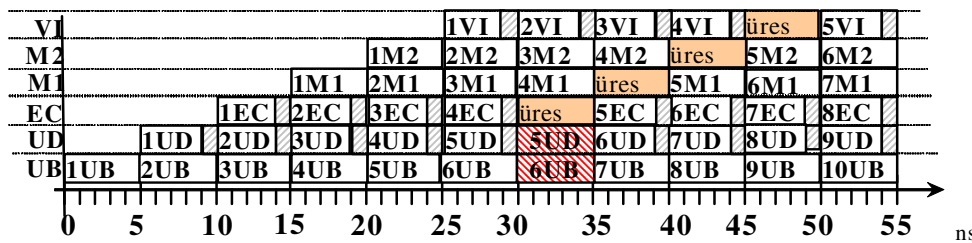
Az utasításkibocsajtás a fenti esetben: $TP = 1 \text{ utasítás} / 50 \text{ ns}$

A szalag hatékonysága:

$$H = \frac{S_u}{EMV_{sz}} = \frac{2.6}{3} = 0.86 \rightarrow 86\%$$

Az elemi műveletvégző egységek időben egyenlőtlen terheléséből adódó problémát puffer tároló alkalmazásával csökkenthetjük. A puffer csak ideiglenes túlterhelés esetén jelent megoldást. Ha lehet, célszerű a nagyobb időt igénylő tevékenységeket kisebb részekre osztani.

8.

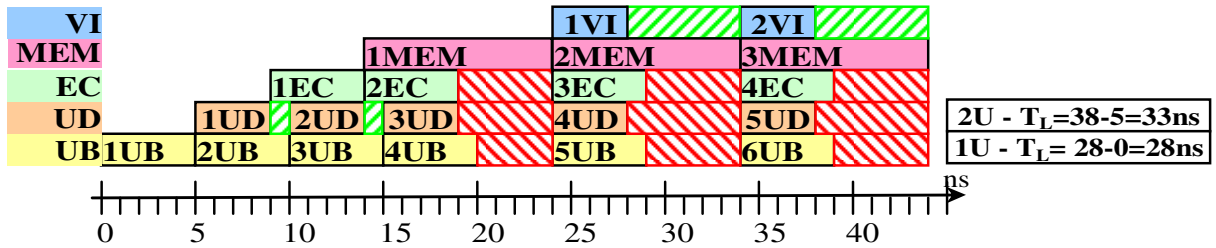


Mekkora a fenti rendszer hatékonysága

$$H_{su} = \frac{S_u}{EMV_{sz}} = \frac{(5ns + 4ns + 4ns + 5ns + 5ns + 4ns) - 6 \cdot 0.5ns}{5ns} = \frac{27ns - 3ns}{5ns} = \frac{24ns}{5ns} = 4.8$$

Az ötödik utasítás lappangási ideje: $T_L = 55-20=35ns$

9.



A második utasítás lappangási ideje: $T_L = 38 - 5 = 33ns$

Az utasítás kibocsájtás: $T_p = 1ut/10ns$

Az utasítás kibocsájtás növelhető például a memória elérés műveletvégzőt két 5ns-os műveletvégzőre bontásával (ha lehetséges)

10. Amdahl törvény a konkrét példára:

$$S_u = \frac{t_{MA}}{t_{MAúj}} = \frac{t_{MA}}{\frac{x \cdot t_{MA}}{n} + \frac{(1-x) \cdot m \cdot t_{MA}}{n}} \rightarrow S_u = \frac{n}{x + m \cdot (1-x)} = \frac{1}{\frac{x}{n} + \frac{m}{n}(1-x)}$$

$n=2, m=3$

$$S_u = \frac{2}{0.8 + 3 \cdot 0.2} = \frac{2}{1.4} = 1.42$$

Egy lehetséges értelmezés:

Az eredeti CISC feldolgozás 2 időegységet igényelt / az új RISC pedig: az esetek 80%-ban 1 időegységet, a maradék 20%-ában pedig 3 időegységet.