

1. Feladat Legyen adott az alábbi utasítás sorozat:

```
i1: D1 ← MEM [R0+0]
i2: D1 ← D0 * D1
i3: D4 ← D2 / D1
i4: D2 ← MEM [R1+0]
i5: D3 ← D0 + D2
i6: D4 ← D4 - D3
i7: MEM [R1+8] ← D4
```

- (a) Rajzolja fel az utasítások percedenciagráfját! Az éleket címkézzé fel az egymásrahatások típusának megfelelően (RAW/WAR/WAW)! Ha két utasítás között több egymásrahatás is van, tüntesse fel mindet!
- (b) Szüntesse meg a WAW és WAR adat-egymásrahatásokat regiszter átnevezés segítségével! A regiszter átnevezést végezze el minden utasításra szisztematikusan! A lebegőpontos fizikai regisztereket jelölje T0, T1, T2, ..., az egész típusú fizikai regisztereket U0, U1, U2,

A regiszterleképező tábla tartalma kezdetben:

Logikai	Fizikai regiszter
R0:	U7
R1:	U3
D0:	T6
D1:	T2
D2:	T8
D3:	T1
D4:	T11

Ha új, szabad fizikai regiszterre van szüksége, válassza a táblázatban szereplő legnagyobb sorszámú után következőt!

- (c) Rajzolja fel az utasítások precedenciagráfját a regiszter átnevezés után!

2. Feladat Legyen adott az alábbi utasítássorozat:

```
i1: R2 ← MEM [R0+0]
i2: R3 ← R0 * R2
i3: R8 ← R4 / R3
i4: R5 ← MEM [R1+8]
i5: R6 ← R2 + R5
i6: R9 ← R5 / R6
i7: R10 ← R6 * R9
```

Az utasítássorozatot lefordítjuk egy olyan VLIW processzorra, melyben az alábbi utasítások helyezhetők el egy utasításcsoportban:

- 2 db memóriaművelet (végrehajtási idő: 3 órajel, iterációs idő: 1 órajel)
- 2 db egész aritmetikai művelet, vagy ugró utasítás (végrehajtási idő: 1 órajel)

- (a) Ütemezze a megadott utasítássorozatot a megadott VLIW processzoron! Határozza meg az utasításcsoportok tartalmát, és a csoportok végrehajtásának idejét!
- (b) Mennyivel gyorsabb az így kapott program, mintha hagyományos 1-utas processzoron futtatnánk (pipeline nélkül)?
- (c) Hány utasításcsoportot állít elő a fordító klasszikus, és hányat dinamikus VLIW architektúra esetén?

3. Feladat A processzor a program végrehajtása közben az alábbi sorrendben, az alábbi címeken lévő feltételes ugró utasításokat értékeli ki (a zárójelben "T" jelzi, ha az ugrás ténylegesen bekövetkezett, és "N", ha nem):

- 464 (T), 543 (N), 777 (N), 543 (N), 777 (N), 464 (T), 777 (N), 464 (T), 543 (T)

A processzor egyszerű, 2 bites állapotgépre alapozott dinamikus elágazásbecslést használ, és 4 ugró utasítást tud követni.

(a) Hány bitet foglal a PHT?

(b) Írja fel a PHT tartalmát minden egyes feltételes ugrás végrehajtása után! Jelezze, hogy a becselő eltalálta-e az ugrás kimenetelét! Kezdetben minden állapotgép értéke 1.

4. Feladat A processzor a program végrehajtása közben az alábbi sorrendben, az alábbi címeken lévő feltételes ugró utasításokat értékeli ki (a zárójelben "T" jelzi, ha az ugrás ténylegesen bekövetkezett, és "N", ha nem):

- 464 (T), 543 (N), 777 (N), 543 (N), 777 (N), 464 (T), 777 (N), 464 (T), 543 (T)

A processzor korrelált elágazásbecslést alkalmaz globális, 2 bites állapotgépeket tartalmazó PHT-vel, úgy, hogy az utolsó 2 feltételes ugrás kimenetelét tudja nyomon követni.

(a) Hány bitet foglal a PHT?

(b) Írja fel a PHT tartalmát minden egyes feltételes ugrás végrehajtása után! Jelezze, hogy a becselő eltalálta-e az ugrás kimenetelét! Kezdetben minden állapotgép értéke 1, a globális ugrási előzmény regiszter értéke binárisan 11.

5. Feladat Egy processzorban az ugrási címek becslésére BTB-t használnak. A BTB találati arány (vagyis hogy egy ugrási utasításra egyáltalán tartalmaz becslést) 90%. BTB találat esetén a becslés pontossága 90%. Ha az ugró utasítás nincs a BTB-ben, 4 ciklus, ha benne van, de rossz a becslés, 3 ciklusnyi késleltetést szenved a futó program végrehajtása. A programban a feltételes elágazások aránya 15%. Most feltesszük, hogy semmi más (semmilyen más egymásrahatás) nem lassítja a program futását, és ha nincs ugrás, a pipeline minden ciklusban végezni tud egy utasítás végrehajtásával.

(a) Átlagosan hány ciklusonként végez a pipeline egy utasítás végrehajtásával?

(b) Hogy viszonyul az eredmény a BTB nélküli esethez, amikor minden ugró utasítás egységesen 2 szünet beiktatását vonja maga után?