7 GAIA: ARIKETAK

Liburuan proposatutako ariketak

7.1. Jauzi-agindu bakarra erabili dugu diseinatu dugun prozesadorean: `beq`, jauzi baldin 0. Antzeko beste jauzi batzuk erabili ohi dira prozesadoreetan. Esaterako, BIRD konputagailuan beste hauek ere erabiltzen dira:

```
jmp etiketa          ; pc := pc + despl
bne ri1, etiketa     ; baldin(ri1 ≠ 0) pc := pc + despl
bls ri1, etiketa     ; baldin(ri1 < 0) pc := pc + despl
```

Zabal ezazu kontrol-algoritmoa hiru jauzi horiek exekutatu ahal izateko. Aldatu behar da prozesu-unitatea jauzi horiek exekutatu ahal izateko?

7.2. 8000H helbidetik aurrera memorian gorde den programa zati hau exekutatzen da BIRD konputagailuan:

8000H → 0861H / 1000H / 3063H / 0001H / 1461H / 1000H

Zein agindu exekutatzen dira? Nola aldatuko dira erregistro-multzoko erregistroak eta memoria? Zenbat ziklo beharko dira kode zati hori exekutatzeko?

7.3. `beq` agindua exekutatzen denean BIRD konputagailuan, hiru aldiz alda daiteke PC erregistroaren balioa. Esan zein unetan eta zertarako egiten diren aldaketa horiek.

7.4. Zergatik erabili behar izan dugu multiplexore bat EMko @i2 helbide-sarreran?

7.5. R_e1, R_e2 eta R_ual laneko erregistroak erabiltzea prozesu-unitatean erosoa da, baina, prozesadore guztietan ohikoak badira ere, ez dira guztiz beharrezkoak BIRD prozesadorean. Adierazi nola exekuta zitekeen `add` agindua laneko erregistroak erabili gabe. Zenbat ziklo beharko lirateke?

7.6. PCi erregistro laguntzailea erabili dugu exekutatzen ari den aginduaren helbidea gordetzeko, PC erregistroaren balioa aldatzen delako aginduaren bilaketa fasean, hurrengo agindua erakusteko. Hala, desplazamendu bat gehitu behar denean jauzi bat egiteko, PCi laneko erregistroaren edukia erabiliko dugu, hor gorde baitugu jauzi-aginduaren helbidea.
Proposatu beste soluzio bat problema horretarako, baina erregistro laguntzailea erabili gabe.

7.7. Programa zati hau exekutatzen du BIRD prozesadoreak:

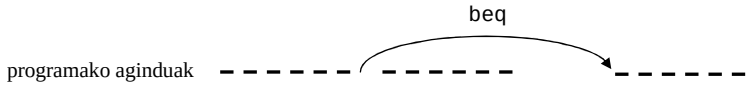
```
baldin (X = Y)
   orduan   BEKT := Y
   bestela  BEKT := Y + 1
```

```
        ...
        ld   r2, X
        ld   r3, Y
        sub  r4, r3, r2
        beq  r4, etik1
        addi r3, r3, #1
etik1:  stx  r3, BEKT[r0]
        ...
```

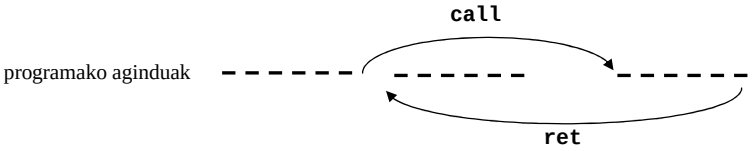
(a) Lehen aginduaren memoria-helbidea 4500H da. Adierazi, zikloz ziklo, PC erregistroak hartuko dituen balio guztiak programa exekutatzean, bi kasu hauetan: (1) X = 4 eta Y = 6; eta (2) X = –2 eta Y = –2.

(b) Bete ezazu taula bat aurreko (1) kasurako (7.5. ariketan egin den moduan), programaren exekuzioaren ondorioz erregistroen eta gainerako gailuen edukiak egoeraz egoera nola aldatzen diren agerian uzteko. Hartu kontuan helbide hauek: @X = 1000H; @Y = 1001H; @BEKT = 1500H

7.8. Bi motatako jauziak bereizten dira konputagailu guztietan: itzulera gabekoak eta itzuleradunak. Lehenengoak sinpleak dira: jauzi egiten da programaren beste toki batera, hango aginduak exekutatzeko, eta hortik aurrera jarraituko da programaren exekuzioa. BIRD prozesadoreko `beq` jauzia mota horretako jauzi bat da.



Bigarrenak, aldiz, konplexuagoak dira: jauzia egiteaz gain, itzulera-helbidea gordetzen dute, eskuarki erregistro berezi batean. Horrela, agindu batzuk exekutatu eta gero, programa gauza da jauzia egin den puntura itzultzeko, hango aginduen exekuzioarekin jarraitzeko. Jauzi horiek erabili behar dira, esaterako, azpiprogramak edo prozedurak exekutatzeko.



Bi agindu erabili ohi dira horretarako: `call`, jauzia egiteko, eta `ret`, itzultzeko. Hartu kontuan bi agindu berrien definizio hauek:

```
call etiketa   → pc := pc + despl; EM[r31] := pc;
ret            → pc := EM[r31];
```

Definizioetan ageri denez, erregistro-multzoko `r31` erregistroan gordetzen da itzulera-helbidea; hortik hartzen du `ret` aginduak jauzia egin den puntura itzultzeko helbidea. Beraz, erregistroa erreserbatu egin behar da eragiketa horretarako.
Gehitu BIRD prozesadoreari `call` eta `ret` aginduak. Analizatu agindu horiek exekutatzeko prozesu-unitatean behar diren gailuak eta loturak, eta aldatu edo emendatu behar duzun guztia. Horrekin batera, sortu kontrol-algoritmoko adar berriak bi aginduen exekuzioa ondo kontrolatzeko.

7.9. Aginduen eragiketa-kodea deskodetzen ari den bitartean, ohikoa da, hainbat konputagailutan, eragigaiak irakurtzea eta laneko erregistroetan uztea (hori dela eta, `DESK` eta `Ir` faseak `D/Ir` izeneko fase batean biltzen dira).
Aldatu BIRD konputagailuaren kontrol-unitatea bi eragiketa horiek batera egin ahal izateko.
Zein da lortuko den abantaila nagusia eragiketak horrela eginez gero?

Liburuan ebatzitako ariketak

7.1. `ld r7, A` agindua exekutatu behar da BIRD konputagailuan. Egin ezazu aginduaren exekuzio osoa (bilaketatik hasita) erakusten duen kronograma bat. Azal itzazu kronograman aginduaren exekuzioarekin zerikusia duten kontrol-seinale guztiak eta gailu guztien edukia.
Hartu hasiera-datu hauek kontuan: aginduaren memoria-helbidea, 4000H; `A` aldagaiaren helbidea eta edukia, 1000H eta 6, hurrenez hurren.

7.2. `stx r4, A[r5]` agindua exekutatu behar da BIRD konputagailuan. Egin ezazu aginduaren exekuzioa (deskodeketa fasetik hasita) erakusten duen kronograma bat. Azal itzazu kronograman aginduaren exekuzioarekin zerikusia duten kontrol-seinale guztiak eta gailu guztien edukia.
Hartu kontuan datu hauek: aginduaren memoria-helbidea, 4000H; `A` aldagaiaren helbidea, 1000H; `r4` = 20 (0014H); eta `r5` = 15 (000FH).

7.3. `mul r9, r6, r9` agindua exekutatu behar da BIRD konputagailuan. Egin ezazu aginduaren exekuzioa (deskodeketa fasetik hasita) erakusten duen kronograma bat. Azal itzazu kronograman aginduaren exekuzioarekin zerikusia duten kontrol-seinale guztiak eta gailu guztien edukia.
Hartu kontuan hasiera-datu hauek: `r9` = 6 (0006H) eta `r6` = –4 (FFFCH).

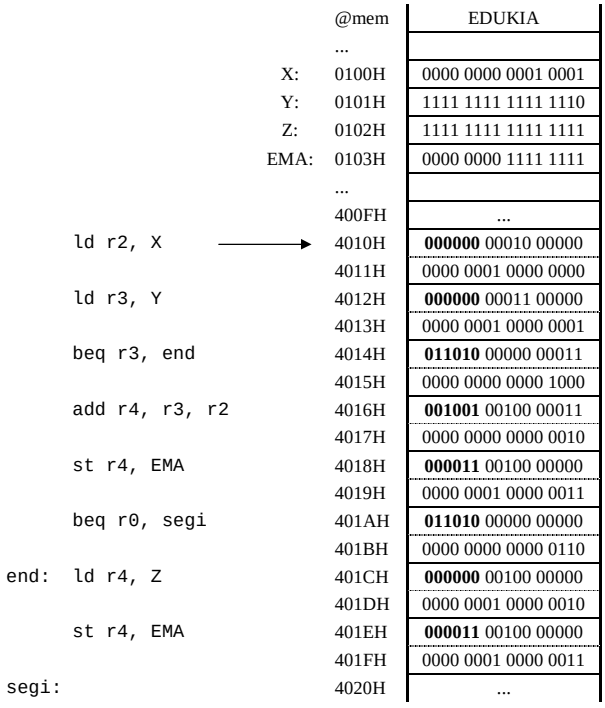
7.4. `beq r2, segi` agindua exekutatu behar da BIRD konputagailuan. Egin ezazu aginduaren exekuzioa (deskodeketa fasetik hasita) erakusten duen kronograma bat. Azal itzazu kronograman aginduaren exekuzioarekin zerikusia duten kontrol-seinale guztiak eta gailu guztien edukia.
Hartu hasiera-datu hauek kontuan: aginduaren memoria-helbidea, 2516H; `r2` = 0; `segi` etiketari dagokion desplazamendua, –8 (FFF8H).

7.5. Programen exekuzioa hobeto ulertzeko, kode zati hau aztertu nahi dugu:

```
  ...
        ld   r2, ALD          ; irakurri ALD aldagaia eta utzi r2 erregistroan
        beq  r2, zero         ; r2 = 0 bada, jauzi egin "zero" etiketa duen agindura
        beq  r0, segi         ; jauzi egin "segi" etiketa duen agindura (r0 = 0)
 zero:  mul  r2, r3, r3       ; bestela, egin biderketa
        st   r2, ALD          ; gorde emaitza ALD aldagaian
 segi:  ...
```

Adieraz itzazu, zikloz ziklo, taula batean, BIRD prozesadorearen erregistroen eta gailu nagusien edukia programa zati hori exekutatzen denean. Hartu kontuan datu hauek: `ld` aginduaren memoria-helbidea, 2500H; `ALD` aldagaiaren helbidea eta balioa, 0100H eta 0; `r3` erregistroaren edukia, 4.

7.6. Irudian, programa zati bat eta dagokion memoria zatia ageri dira.



Programa hori kontuan hartuz, erantzun honako galdera hauei:

(a) Zenbat erloju-ziklo behar dira programa exekutatzeko?

(b) BIRD prozesadorearen kontrol-unitatea `Id_OP` egoeran dago. Zer balio dute une horretan PC, R_e1, R_e2 eta R_ual erregistroek? Eta zer balio dago unitate aritmetiko/logikoaren irteeran? Eta erregistro-multzoko datu- eta helbide-sarreretan?

(c) Kontrol-unitatea `M_ST` egoeran dago, eta PCaren edukia `401AH` da. Zein da IR_1 eta IR_2 erregistroen edukia une horretan? Eta memoriako helbide-sarrera? Eta memoriako datu-sarrera?

(d) Kontrol-unitatea `Id_PC` egoeran dago. Zein agindu ari da exekutatzen? Zein da PCi laneko erregistroaren edukia? Zer dago PCaren datu-sarreran?

(e) PCaren edukia `4011H` denean, zein egoeratan dago kontrol-unitatea?

7.7. BIRD prozesadorea zabaldu nahi dugu agindu berri bat exekutatu ahal izateko:

```
swap ra,rb      →     ra := rb; rb := ra;
```

Aginduak trukatu behar ditu adierazitako bi erregistroen edukiak. Aginduaren formatua hau da:

Egin ezazu agindu honi dagokion adarra kontrol-algoritmoan (deskodeketa eta gero) eta adierazi prozesu-unitatean egin beharreko aldaketak agindua exekutatu ahal izateko.

7.8. `C` = `B` + `A` bektore-eragiketa egiteko, honako programa hau idatzi dugu (ikus 7.3.2. atala):

```
       movi r1, #0          ; indize-erregistroa hasieratu
       movi r2, #16         ; bektoreen osagai kopurua
segi:  ldx  r3, A[r1]       ; irakurri A bektorearen osagai bat
       ldx  r4, B[r1]       ; irakurri B bektorearen osagai bat
       add  r5, r4, r3      ; batu bi osagaiak (emaitza, r5)
       stx  r5, C[r1]       ; gorde emaitza C bektorearen osagaian
       addi r1, r1, #1      ; inkrementatu r1, bektoreen hurrengo osagaiaren indizea
       subi r2, r2, #1      ; osagai bat gutxiago geratzen da
       beq  r2, buka        ; r2 = 0 bada, eragiketa bukatu da
       beq  r0, segi        ; bestela, segi eragiketarekin (r0 = 0)
buka:  ...
```

Irudiko informazioa erabiliz, idatzi bitarrez nola geratuko den programa BIRD konputagailuaren memorian. Programaren hasiera-helbidea, 1000H; bektoreen hasiera-helbideak, `A` → 0100H; `B` → 0200H; `C` → 0300H.

7.9. 100 osagaiko bi bektoreen biderkadura eskalarra —$BE = \sum (X_i \times Y_i)$—kalkulatzen du programa honek:

```
       ...
       movi r1, #0          ; indize-erregistroa hasieratu
       movi r2, #100        ; bektoreen osagai kopurua
       movi r6, #0          ; batura partziala hasieratu
segi:  ldx  r3, X[r1]       ; irakurri X bektorearen osagai bat
       ldx  r4, Y[r1]       ; irakurri Y bektorearen osagai bat
       mul  r5, r3, r4      ; biderkatu bi osagaiak (r5 := r3 × r4)
       add  r6, r6, r5      ; egin batuketa partziala (r6 := r6 + r5)
       addi r1, r1, #1      ; inkrementatu r1, bektoreen hurrengo osagaiaren indizea
       subi r2, r2, #1      ; osagai bat gutxiago geratzen da
       beq  r2, buka        ; r2 = 0 bada, eragiketa bukatu da (jauzi buka-ra)
       beq  r0, segi        ; bestela, segi eragiketarekin (r0 = 0) (jauzi segi-ra)
buka:  st   r6, BE          ; gorde emaitza memorian
       ...
```

(a) Prozesadorearen erlojua 100 MHz-ekoa da. Kalkula ezazu zenbat denbora beharko den eragiketa exekutatzeko, kontrol-algoritmoa 7.15. irudikoa bada.

(b) Erloju-maiztasuna 250 MHz-etara igo daiteke, baina, hala egiten bada, memoria-eragiketek 3 ziklo behar dute. Zenbat aldiz azkarrago exekutatuko da programa aurreko kasuarekin alderatuta?