Konputagailuen Arkitektura I

Kontrol-unitatea 2 (ebazpena): RET agindua

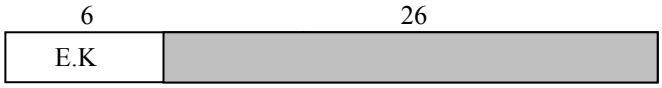
BIRD prozesadorearen agindu-multzoan beste agindu bat gehitu nahi dugu:

```
ret
```

Agindu horrek pilaren tontorrean gordeta dagoen balioa `PC` erregistroari esleitzen dio eta `r31` erregistroaren edukia dekrementatzen du.

```
PC := MEM[r31],
r31 := r31 – 1
```

Bere formatua honako hau da:



Idatz ezazu agindua exekutatzeko behar den mikroprogramaren zatia.
Prozesu-unitateko egituran aldaketarik egin beharko balitz, azaldu ezazu argi eta garbi. Beste aginduei dagokien kontrol-atalean, aldaketarik gertatuko al da?

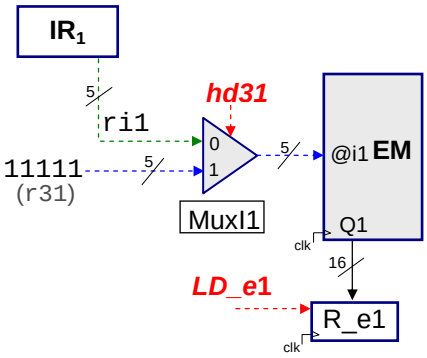
Ebazpena

Batetik, mikroprograma idatzi beharko dugu, eta mikroagindu bakoitzean aktibatu beharreko kontrol-seinaleak adierazi; bestetik, prozesu-unitatean aldaketak egin beharko direnez, zehaztu beharko ditugu.
Mikroprogramari dagokionez, hainbat aukera dauden arren, hurrengoa proposatzen dugu:

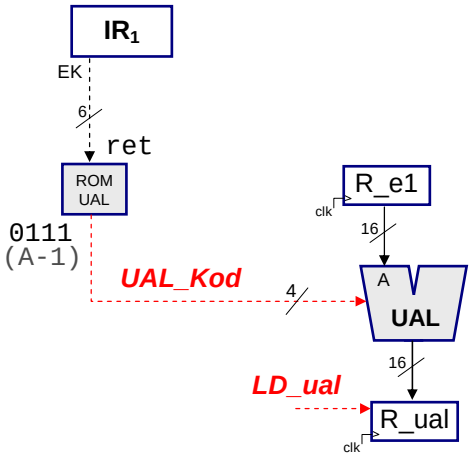
```
ret1: R_e1 := EM[r31];
ret2: R_ual := R_e1 – 1; PC := MEM[R_e1];
ret3: EM[r31] := R_ual; goto 0;
```

Hau da, lehenik pilaren tontorra (`r31` erregistroak gordetzen duen helbidea) eskuratu behar da dekrementatzeko. Beraz, deskodeketa-fasearen ondorengo lehen mikroaginduak, *ret1* (`R_e1 := EM[r31]`), `r31`an dagoena irakurtzen du, balioa `R_e1` erregistrora kopiatzeko. `r31` erregistroa inplizituki erabiltzen denez agindu honetan (ez baita ageri aginduaren formatuan), erregistro-multzoko `@i1` sarrerari adierazi beharko zaio 31 erregistroa dela irakurri beharrekoa. `@i1` sarreran `IR1` erregistrotik datozen `ri1` izeneko 5 bitak jadanik konektatuta daudenez, BIRD makinaren prozesu-unitatea aldatu beharko da `r31` irakurri ahal izateko. Zein izango da hori egiteko behar den aldaketa? Multiplexore bat (MuxI1) erantsi beharko dugu erregistro-multzoko `@i1` sarreran. Multiplexore horren sarrerak bost bitekoak

dira: IR1 erregistrotik datozen ri1 bitak, edota bost bateko (31 balioa bitarrez adierazteko, 11111). Multiplexorearen sarreretako bat aukeratzeko, kontrol-seinale berri bat beharko da, demagun hd31 izenekoa. ret agindua denean, bost batekoen segida aukeratu beharko da, adibidez, multiplexorearen 1 sarreran egongo dira bit horiek, eta bestelako aginduetan 0 sarreratik datozen ri1 bitak aukeratuko dira. Beraz, ret1 mikroagindua exekutatu ahal izateko, aldaketa hauek egin behar dira prozesu unitatean:

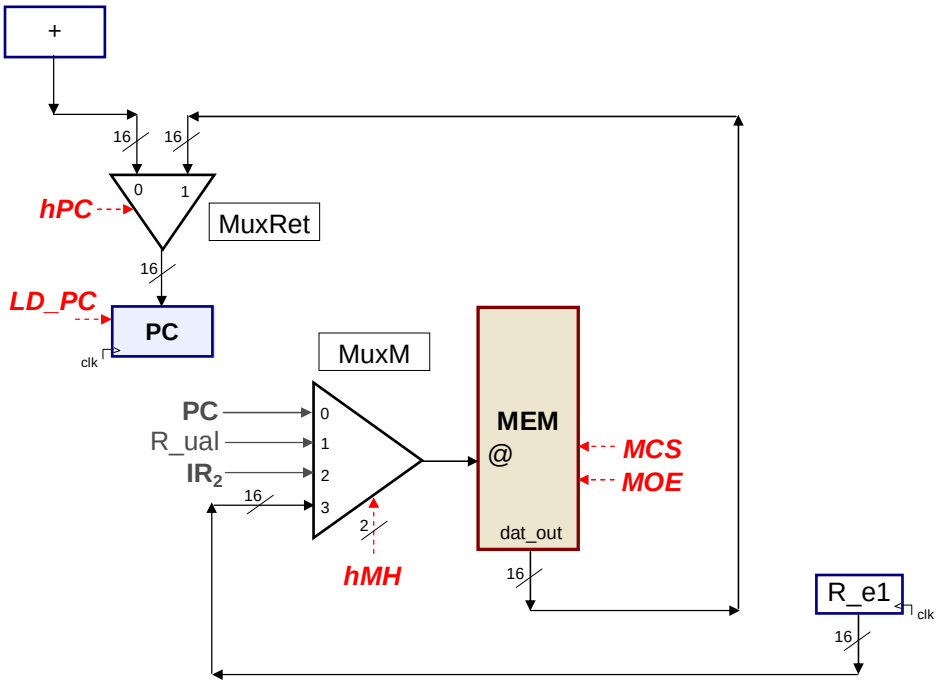


Eta ret1 mikroagindua exekutatu ahal izateko, kontrol-seinale hauek aktibatu behar dira: hd31 eta LD_e1.
Horren ondoren, ret2 mikroagindua (R_ual := R_e1 - 1; PC := MEM[R_e1]) exekutatu behar da; hala, behin kopiatuta R_e1 erregistroan r31 erregistroaren balioa, dekrementatu behar da, UALa erabiliz. UALari ROM_UAL izeneko memoriatik etorriko zaio dekrementatzeko kodea (ret aginduaren eragiketa-kodetik abiatuta) eta emaitza R_ual erregistroan gordeko da, LD_ual kontrol-seinalea aktibatuz.
Hortaz, ret2 mikroagindua exekutatu ahal izateko, prozesu-unitatean ez da aldaketa berezirik egin behar, behar diren osagai guztiak jadanik daudelako prozesu-unitatean, baina ROM_UAL memoriaren barruan, ret aginduaren eragiketa-kodeak adierazitako posizioan, (A-1) eragiketari dagokion eragiketa-kodea (0111) idatzi behar da. Modu horretan, ret agindua datorrenean, UALak jakingo du dekrementatu behar duela R_e1 erregistroan dagoena.

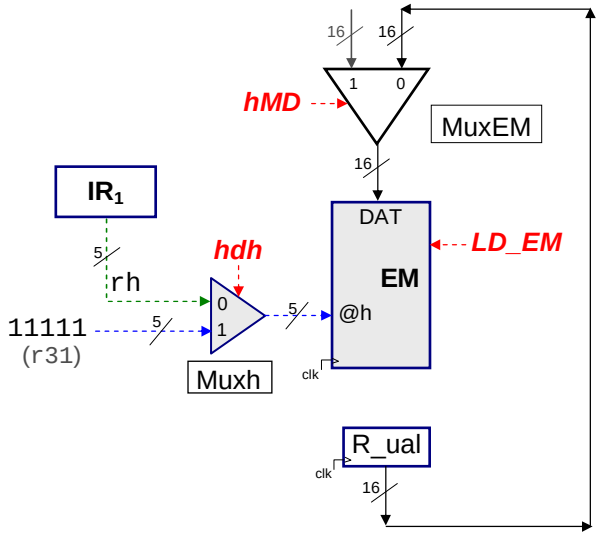


Bestalde, ret2 mikroaginduan PC erregistroaren eguneraketa ere (PC := MEM[R_e1]) jarri dugu, nahiz eta posible izan ret3 mikroaginduan egitea. Prozesu-unitateko konexioak ikusita, agerikoa da ezinezkoa dela eguneraketa hori egitea aldaketarik egin gabe, hau da, PCan ezin

da kargatu memoriatik irakurritako datu bat. Horregatik, memoriako `dat_out` irteeratik `PC` erregistrorako bidea zabaldu behar da. Hori egin ahal izateko, `PC`aren sarreran multiplexore bat jarriko da, 0 sarreran batugailutik datorren balioa konektatuz eta 1 sarreran `dat_out` bidetik datorrena. Multiplexore horretan aukeraketa egiteko `hPC` kontrol-seinale berria erantsi beharko da.
Dena den, `PC`an balio hori kargatzeko memoriatik irakurri behar da `R_e1` erregistroak adierazten duen helbidean. Horretarako, `R_e1`en irteera memoriaren `@` sarrerara konektatu behar da. Sarrera horretan `MuxM` aurkitzen dugu eta 0, 1 eta 2 sarrerak beteak ditu, eta 3 sarrera libre dago. Beraz, 3 sarreran `R_e1`etik datorren balioa erantsi dezakegu, jakin ahal izateko non dagoen `PC`an kopiatu beharreko balioa. Halaber, memoriako `MCS` eta `MOE` kontrol-seinaleak aktibatu behar dira irakurri ahal izateko.
Laburbilduz, *ret2* mikroagindua exekutatu ahal izateko kontrol-seinale hauek aktibatu behar dira: `LD_ual`, `hMH` (‘`11`’ balioa eman behar zaio), `MCS`, `MOE`, `hPC` eta `LD_PC`. Eta aldaketa hauek egin behar dira prozesu unitatean:



ret3 mikroaginduan erregistro-multzoko `r31` erregistroan balio bat gordetzeko, irakurtzean bezala adierazi beharko diogu `@h` sarreran 31 balioa, `IR1`etik datozen 5 bitez gain. Hemen ere, multiplexore bat erantsi beharko da bi sarrerekin (batetik, `IR1`etik datorren `rh` bit segida, eta, bestetik, `11111` bit segida) eta kontrol-seinale berri batekin (demagun `hdh` izenekoa). Aurrekoetan egin dugun bezala, multiplexorearen 0 sarrera izango da prozesu-unitatean dagoen oraingo balioa, eta 1 sarreran erantsi behar izan duguna (`11111` segida, kasu honetan). Orduan, `R_ual` erregistroan dagoen balioa `r31` erregistroan gorde ahal izateko, `R_ual` erregistrotik erregistro-multzoko datu-sarrerara doan lotura jadanik egina dagoenez jatorrizko prozesu-unitatean, `hMD` seinaleak zero balioa izan behar du, eta bakar bakarrik aktibatu behar dira `LD_EM` eta `hdh` seinaleak. Aldaketaren ondoren, hau ere erantsi da prozesu-unitatean:



Sekuentziazioari dagokionez, *ret1* mikroaginduaren ondoren exekutatu beharrekoa beti *ret2* mikroagindua izango da, eta *ret2* mikroaginduaren ondoren, *ret3* mikroagindua izango da beti. Beraz, bi kasu horietan baldintza kodea `00` da (zero konstanteari dagokiona, ez baita jauzirik egin behar sekula), deskodeketako bita `0` da, eta jauzi helbideko bost bitetan berdin zaigu zer jarri, ez baita sekula jauzirik egingo. *ret3* mikroaginduaren kasuan, ordea, ondoren beti `0` mikroagindua exekutatu behar denez (`goto 0`), baldintza kodea `11` da (bateko konstantea, beti jauzia egin behar delako), deskodeketa bita `0` izango da, eta jauzi helbideko bost bitak zeroak izango dira, hurrengo aginduaren lehen bilaketa fasera joan ahal izateko.

Mikroaginduak bitarrean nola geratzen diren jarraian adierazten dugu.

egoera	BK1BK0	Desk	S4S3S2S1S0	mcs	moe	mwr	hMH	ld_IR1	ld_IR2	ld_EM	hMD	hH	ld_e1	ld_e2	hB	ld_ual	ld_PC	ld_PCi	hD	hd31	hPC	hdh
ret1	00	0	xxxxx	0	0	0	xx	0	0	0	x	x	1	0	x	0	0	0	0	1	x	x
ret2	00	0	xxxxx	1	1	0	11	0	0	0	x	x	0	0	x	1	1	0	0	x	1	x
ret3	11	0	00000	0	0	1	xx	0	0	1	0	x	0	0	x	0	0	0	x	x	x	1

Argi dago, kontrol-seinale berriak sartu behar izan ditugunez, erantsi beharreko aldaketak zirela eta, mikroaginduen luzera handitu dela. Beste aginduen exekuzioan, gehitu ditugun kontrol-seinale berriak desaktibatuta egongo dira, ariketan zehar aipatu diren kasuetan salbu.