

### **Primjer jednociklusne naredbe ADD R1,R2,R3**

Razina dohvata:

Rastući brid CLOCK-a:

PC → AR

Padajući brid CLOCK-a:

PC+1 → PC

(AR) → DR ; dohvat strojnog koda iz memorije

DR → IR, dekodiranje

dohvat operanada iz registara R1 i R2 i slanje u ALU

Razina izvođenja:

Rastući brid CLOCK-a:

ALU izvodi operaciju zbrajanja

Padajući brid CLOCK-a:

upis zbroja iz ALU u R3

postavljanje zastavica u SR-u

59

### **Primjer jednociklusne naredbe LOADL R1,123**

Razina dohvata:

Rastući brid CLOCK-a:

PC → AR

Padajući brid CLOCK-a:

PC+1 → PC

(AR) → DR ; dohvat strojnog koda iz memorije

DR → IR, dekodiranje

dohvat broja 123 iz strojnog koda i predznačno proširenje

Razina izvođenja:

Rastući brid CLOCK-a:

-

Padajući brid CLOCK-a:

upis predznačno proširenog broja 123 u R1

60

### **Primjer dvociklusne naredbe LOADcc R1,(1000)**

Razina dohvata:

Rastući brid CLOCK-a:

PC → AR

Padajući brid CLOCK-a:

PC+1 → PC

(AR) → DR ; dohvat strojnog koda iz memorije

DR → IR, dekodiranje

dohvat adrese 1000 iz strojnog koda i predznačno proširivanje  
onemogućavanje dohvata sljedeće naredbe

Razina izvođenja:

Rastući brid CLOCK-a:

predznačno proširena adresa → AR

Padajući brid CLOCK-a:

(AR) → DR ; dohvat podatka iz memorije

DR → R1

postavljanje zastavica u SR-u

omogućavanje dohvata sljedeće naredbe

61

### **Primjer dvociklusne naredbe STORE R1,(R2+30)**

---

Razina dohvata:

Rastući brid CLOCK-a:

PC → AR

Padajući brid CLOCK-a:

PC+1 → PC

(AR) → DR ; dohvat strojnog koda iz memorije

DR → IR, dekodiranje

dohvat R2, predznačno priširivanje 30 i njihovo slanje u ALU

onemogućavanje dohvata sljedeće naredbe

Razina izvođenja:

Rastući brid CLOCK-a:

računanje adrese u ALU

adresa → AR

R1 → DR

Padajući brid CLOCK-a:

DR → (AR) ; upis podatka u memoriju

omogućavanje dohvata sljedeće naredbe

62

### **Primjer dvociklusne naredbe PUSH R1**

---

Razina dohvata:

Rastući brid CLOCK-a:

PC → AR

Padajući brid CLOCK-a:

PC+1 → PC

(AR) → DR ; dohvat strojnog koda iz memorije

DR → IR, dekodiranje

dohvat R7 i slanje u ALU

onemogućavanje dohvata sljedeće naredbe

Razina izvođenja:

Rastući brid CLOCK-a:

R7-1 → AR ; računanje nove adrese vrha stoga (u ALU)

R1 → DR

Padajući brid CLOCK-a:

DR → (AR) ; upis podatka na stog

rezultat iz ALU, tj. R7-1 → R7

omogućavanje dohvata sljedeće naredbe

63

### **Primjer dvociklusne naredbe POP R1**

---

Razina dohvata:

...

; kao u naredbu PUSH

Razina izvođenja:

Rastući brid CLOCK-a:

R7-1 ; računanje nove adrese vrha stoga (u ALU)

R7 → AR

Padajući brid CLOCK-a:

(AR) → DR ; čitanje podatka sa stoga

rezultat iz ALU, tj. R7+1 → R7

DR → R1

omogućavanje dohvata sljedeće naredbe

64

### Primjer dvociklusne naredbe JP\_EQ 1000

---

Razina dohvata:

Rastući brid CLOCK-a:

PC → AR

Padajući brid CLOCK-a:

PC+1 → PC

(AR) → DR ; dohvat strojnog koda iz memorije

DR → IR, dekodiranje

ispitivanje istinitosti uvjeta (je li Z=1 ?)

onemogućavanje dohvata sljedeće naredbe

Razina izvođenja:

Rastući brid CLOCK-a:

ako je Z=1: dohvat adrese 1000 iz strojnog koda i predznačno proš.

Padajući brid CLOCK-a:

ako je Z=1: adresa → PC

omogućavanje dohvata sljedeće naredbe

65

### Primjer dvociklusne naredbe JR\_UGE 1000

---

Razina dohvata:

Rastući brid CLOCK-a:

PC → AR

Padajući brid CLOCK-a:

PC+1 → PC

(AR) → DR ; dohvat strojnog koda iz memorije

DR → IR, dekodiranje

ispitivanje istinitosti uvjeta (je li C=0 ?)

onemogućavanje dohvata sljedeće naredbe

Razina izvođenja:

Rastući brid CLOCK-a:

ako je C=0: dohvat pomaka iz strojnog koda i predznačno prošir.

Padajući brid CLOCK-a:

ako je C=0: PC + pomak → PC

omogućavanje dohvata sljedeće naredbe

66

### Primjer dvociklusne naredbe CALL 1000

---

Razina dohvata:

... ; kao za JP i JR

određivanje da je uvjet uvijek istinit

uvijek: dohvat R7 i slanje u ALU

onemogućavanje dohvata sljedeće naredbe

Razina izvođenja:

Rastući brid CLOCK-a:

uvijek: dohvat adrese 1000 iz strojnog koda i predznačno prošir.

uvijek: R7-1 → AR ; računanje nove adrese vrha stoga (ALU)

uvijek: PC → DR

Padajući brid CLOCK-a:

uvijek: adresa → PC

uvijek: DR → (AR) ; spremanje povratne adrese na stog

uvijek: rezultat iz ALU, tj. R7-1 → R7

omogućavanje dohvata sljedeće naredbe

67

## Primjer dvociklusne naredbe RET

---

Razina dohvata:

... ; kao za prethodni CALL

Razina izvođenja:

Rastući brid CLOCK-a:

uvijek:  $R7+1$  ; računanje nove adrese vrha stoga (u ALU)

uvijek:  $R7 \rightarrow AR$

Padajući brid CLOCK-a:

uvijek:  $(AR) \rightarrow DR$  ; dohvat povratne adrese sa stoga

uvijek:  $DR \rightarrow PC$

uvijek: rezultat iz ALU, tj.  $R7+1 \rightarrow R7$

omogućavanje dohvata sljedeće naredbe