

Apxixenturivni Ynologias - Diastis 6

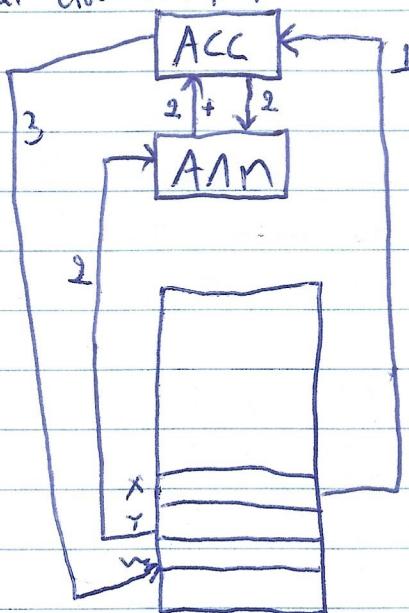
ΣΧΕΔΙΑΣΗ ΣΥΝΟΛΟΥ ΕΝΤΟΛΩΝ

Παράγεται: 1) Enthosi Apxixenturivni, pali tis orous yivri, η enotuvrin

RISC:

$\text{ADD } w \leftarrow x+y : w \leftarrow x+y$ (anifian on neppixipon
tov διεγ. pivri)
μηδενιν
 $x, y, \text{ to anotetwv}\$
on w)

1) Apxixenturivni eukarpi: O epos and tis neppixipou na yagplian
pivs oni evrosi, eton X, neppixipou. Otox eukarpi: O δicte-
pus neppixipou anotetwv on AM. To anotetwv tis neppixipou anotet-
wv na yagplian



1) LOAD X: Ta neppixipon tis
δicte pivri $X \rightarrow \text{Acc}$

2) ADD Y: neppixipon on ACC tu
neppixipon tis δicte pivri
 $\text{Acc} \leftarrow \text{Acc} + y$ (1)

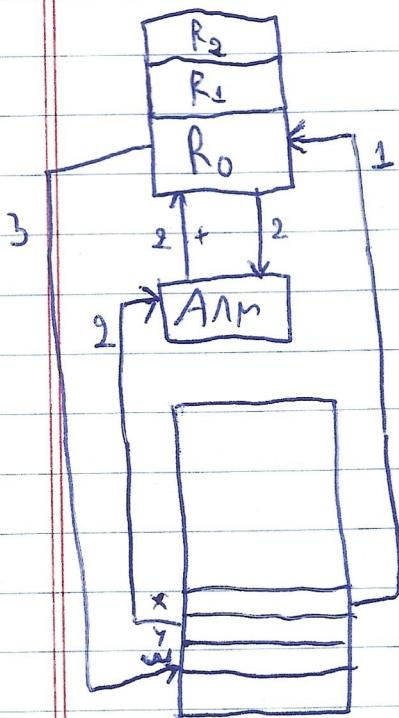
3) STORE W: H tis tu ACC
anotetwvta. on
δicte pivri v

EE odo tis evrosi, o
ACC unotetwv

(*) H neppixipon tis ACC $\leftarrow \text{Acc} + y$

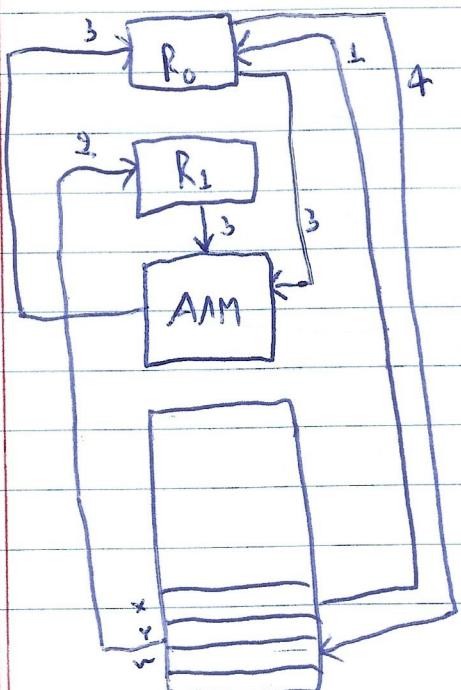
2) Αρχιτεκτονική καταχώρισης-μηδημά: Οι είναι παρόχυτας που πεινει
εις καινούς καταχώριση γενικού ανοιξι R_i και ο άλλος
είναι AΛΜ που μηδεμά (κανεις πρωτότυπη)

πρωτότυπη Αθετήσιμη



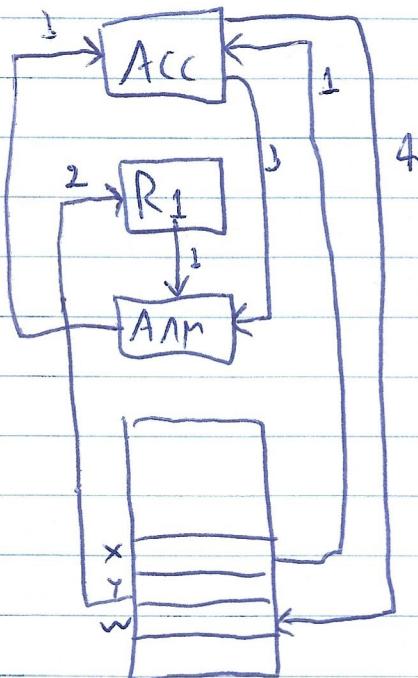
- 1) MOV R₀, X : Μετανίσηση την περιέχουσαν την πρώτη μήδημα X σε R₀
2) ADD R₀, Y : R₀ ← R₀ + Y
To ανατίθεση την πρώτη μήδημα σε R₀ (X+Y)
3) MOV W, R₀ : Μετανίσηση από την R₀ → W

3) Αρχιτεκτονική καταχώρισης-καταχώριση: Οι που εκτελούνται σε
μήδημα την περιέχουσαν ~~παραγόντα~~ την σημείωση πρώτη που πεινει
εις καταχώρισης (για την ανάλογη πρωτότυπη να γίνει και
ο ACC)



- 1) MOV R₀, X
2) MOV R₁, Y
3) ADD R₀, R₁ : R₀ ← R₀ + R₁ (R₀ ← X+Y)
4) MOV W, R₀ : To ανατίθεση αντικειμένων
εις διαγόμενη μήδημα W.

4) Approximating division: 0_{2n} or register γ_1 ends up positive
on division



1) LOAD X: putting our Acc

to Register γ_1
big num X

2) MVR R_1, X

3) ADD R_1 : $(ACC \leftarrow ACC + R_1)$

4) STORE W: 0 Acc another time
on big num v.

Answer

~~1~~. $f = \frac{x+y}{2+w}$ Na Subs to approximate or approximating
register-to-register if x,y are same!

1^{os} tries

1) LOAD X ($ACC \leftarrow X$)

2) MVR R_0, Y ($R_0 \leftarrow Y$)

3) ADD R_0 ($ACC \leftarrow ACC + R_0$, $ACC \leftarrow X+Y$)

4) MVR R_1, ACC ($R_1 \leftarrow ACC = X+Y$)

5) LOAD Z ($ACC \leftarrow Z$)

6) MVR R_0, W ($R_0 \leftarrow W$)

7) ADD R_0 ($ACC \leftarrow ACC + R_0 = Z+W$)

8) DIV R_1 : ($ACC \leftarrow ACC / R_1 = \frac{Z+W}{X+Y}$)

9) ~~MVR~~ MVR $R_2, H1$

10) DIV R_2 : ($R_2 \leftarrow \frac{1}{ACC}$)

MVR = move immediate

11) MVR F, R_2

- (Acc ← Z)
- (R₀ ← w)
- (Acc ← Acc + R₀ = Z + w)
- (R₁ = Z + w)
- (Acc ← X)
- (R₀ ← Y)
- (Acc ← Acc + R₀ = X + Y)
- (Acc ← Acc / R₁)
- STORE F

X ← R₁
 Z ← R₀, Y ← R₁
 (Acc ← R₀ + R₁)

- 1) MOV R₀, X
- 2) MOV R₁, Y
- 3) ADD R₁, R₀
 $(R_0 \leftarrow R_0 + R_1 = X + Y)$
- 4) MOV R₂, Z (R₂ ← Z)
- 5) MOV R₃, W (R₃ ← w)
- 6) ADD R₂, R₃
 $(R_2 \leftarrow R_2 + R_3 = Z + w)$

① ~~MOV R₀, X~~
 Shift left ADD/MOV
 2 jumps

① ~~MOV R₁, Y~~
 4 jumps o Acc

② ~~MOV R₂, Z~~

7) DIV R₀, R₂

$$(R_0 \leftarrow R_0 + R_2 = \frac{X+Y}{Z+w})$$

ADD

[OPC ADD01S]

ADD

[OPCODE] ADD | ADD2

8) MOV F, R₀

2^{ος} Λεπτίγιαν: πήγαν την πρόγραμμα σε μνήμη

1) Εντολή ισχίου παραγόντων

OPCODE	AD1	AD2	AD3
--------	-----	-----	-----

CISC

2) Εντολή σύνταξης παραγόντων

OPCODE	AD1	AD2
--------	-----	-----

3) Εντολή ενδιάμεσης παραγόντων

OPCODE	AD1	AD2
--------	-----	-----

: Ενδιάμεση σε μνήμη υποδοχής

4) Εντολή ο παραγόντων (αριθμ.)

OPCODE

Θεωρούμε ότι οι παραγόντες της εντολής προσεγγίζονται σε σειρά στοιχείων

$$F = \left(\frac{X+Y}{Z+W} \right)$$

1^η αριθμητική: Εντολή ισχίου παραγόντων

1) ADD F₀, X, Y (R₀ ← X+Y)

2) ADD R₁, Z, V (R₁ ← Z+W)

3) DIV F, R₀, R₁ (F ← R₀/R₁ = $\frac{X+Y}{Z+W}$)

Σε αρχιτεκτονική Reg - tv - Reg

1) φέρνει τον X στη R₁

2) φέρνει τον Y στη R₂

3) Αριθμητική

Εντολή ADD ισχίου παραγόντων: MDR ← [OPC | R₀ | X | Y]

ΠΡΩΤΑ ΑΡΧΑΙΑΝΤΑ
T₀ → T₂ MAR ← MAR + 1

AD₀ AD₁ AD₂

T₁: ~~Z ← MVR~~

T₂: MAR ← MVR (AV₁)

T₃: MDR ← M(MAR) (MDR ← M(X))

T₄: ACC ← MVR

T₅: MAR ← Z (ADD)

T₆: MVR ← n(MAR) (MVR ← M(X))

T₇: ACC ← MVR + ACC

CISC : COMPLEX
INSTRUCTION
SET
COMPUTER

T₁₀: R₀ ← ACC