

Αρχιτεκτονική Υπολογιστή - Διαλέξη 15

FO: Προσάρτη του παρόντος (γρήγορη Δεδομένα και η μνήμη)

WO: Γρήγορη αναζήτηση (γρήγορη στη μνήμη)

$\{MAR_0, MDR_0\} \rightarrow$ Αρχική κατάσταση | Μνήμη Έκδοσης

$\{MAR_1, MDR_1\} \rightarrow$ Δεδομένα

Τι θα γίνει αν στην ίδια κωδικό πρόγραμμα, έχουμε μια ενδιάμεση κατάσταση
προσάρτη Δεδομένων (FO) και μια άλλη, γρήγορη, να γρήγορη
αναζήτηση στη μνήμη (WO)

1) Είναι καθυστερημένη: Η πιο γρήγορη θα λαμβάνει την αλληλ.
π.χ. η αναζήτηση Δεδομένων προηγείται της Εγγραφής

2) Να διαχωριστούν ~~αυτά~~ δύο για γρήγορα Στοιχεία και Καταχωρητές

1) Μνήμη Έκδοσης (MAR_0, MDR_0) και 2 Στοιχεία (Stack, Δεδομένα)

2) Μνήμη Δεδομένων (MAR_1, MDR_1) \rightarrow Προσάρτη Δεδομένων στη μνήμη,
2 Στοιχεία (Stack, Δεδομένα).

Εξαρτάται το πρώτο FO

$(MAR_0, MDR_0) \rightarrow$ Εγγραφή Δεδομένων στη μνήμη

2 Στοιχεία (Stack, Δεδομένα). Εξαρτάται το WO

6 Registers
6 Στοιχεία

Προγράμματα

LDA 500

RLI ACC (Αποσπρι αλίκης 1 bit) *

STA 501

Εμφάνιση

Φόρτωση απροσχηματισμένης θέσης μνήμης 500 στο ACC

Ρολ/γρσ εν 2

Αποθήκευση στο θέση μνήμης 501

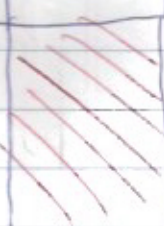


3 στάδια + 5 στάδια pipeline - 1 = 7 κύκλοι ρολογιού

	t0	t1	t2	t3	t4	t5	t6	t7
LDA 500	IF	DI	FO	EX	WU	-	-	
RLI ACC		IF	DI	FO	EX	WU	-	
STA 501			IF	DI	FO	EX	WU	

Το πρόβλημα IF/DI ανιχνεύεται όταν η αντιστάθμιση του σταδίου IF, είναι
 να γίνει. Συντίθεται στο DI

DI/FO

FO/EX → EX: Τα απροσχηματισμένα της θέσης μνήμης 500 από τον MVR₀

	t ₀	t ₁	t ₂	t ₃	t ₄	t ₅	t ₆
LDA 500	MAR ← PC Z ← PC+1 PC ← Z IF	IR ← 1F/D1 (true, opcode) MAR ← PC Z ← PC+1 PC ← Z IF	MAR ← D1/F0 (Address) MVR ₀ ← M[MAR ₀] FO	ACC ← F0/EX	—		
RLI ACC		MAR ← PC Z ← PC+1 PC ← Z IF	IR ← 1F/D1 (true, opcode) —	EX	Z ← RL ₁ (ACC) ACC ← Z EX/ALU	—	
STA 501			MAR ← PC Z ← PC+1 MVR ← M[MAR] PC ← Z IF	IR ← 1F/D1 (true, opcode) —	—	MVR ₁ ← MVR F0/EX EX	MAR ₁ ← EX/W0 (Address) M[MAR ₁] ← MVR ₁ IF

(*) MVR₁ ← F0/EX

○ MVR₁ as address to store in main. Its address in ACC unique for EX/W0

(*) 2 ○ unique and to unique 1F/D1 through to EX/W0

SWAP (11-12 series)
CISC

RISC version

LDA M		ACC ← [M]
MOV R0, ACC		R0 ← [M]
LDA M+1		ACC ← [M+1]
STA M		[M] ← [M+1]
MOV M+1, R0		[M+1] ← R0, [M+1] ← [M]

O, [] given. The description
The other part

109 instructions produced

	t_0	t_1	t_2	t_3	t_4	t_5	t_6
LDA M	MAR ← PC Z ← PC + 1 MVR ← M[MAR] PC ← Z IF	IR ← IR / DI (new, opcode) DI	MAR ← DI / FO (address) MVR ← M[MAR] FO	ACC ← FO / EX (MVR) EX	—	WO	///
MVU R ₀ ACC	MAR ← PC Z ← PC + 1 MVR ← M[MAR] PC ← Z IF	MAR ← PC Z ← PC + 1 MVR ← M[MAR] PC ← Z IF	IR ← IR / DI (new) DI	—	R ₀ ← FO / EX (ACC) EX	—	///
LDA M+1			AVG	AVG	MAR ← DI / FO (new) MVR ← M[MAR] FO	ACC ← FO / EX (MVR) EX	— VO
STA M			AVG	AVG	AVG	—	MVR ₁ ← FO / EX (ACC) EX
MVU M+1 R ₀					AVG	AVG	— FO

t_7	t_8	t_9
////	////	/ / / / /
////	////	/ / / / /
////	/ (/ / /	/ / / / /
$MAR_1 \leftarrow EX/WD(Addr_2)$ $M(MAR_1) \leftarrow MDR_1$ $M(MAR_1) \leftarrow MDR_1$	////	/ / / / /
DELAY	$MDR_1 \leftarrow R_6$ EX	$MAR_1 \leftarrow EX/WD(Addr_1)$ $M(MAR_1) \leftarrow MDR_1$

DELAY: για να γίνει ο R_6 τα αποτελέσματα του EX πάλι είναι να τα πάρει στο MDR_1 ο οποίος είναι αναμενόμενο. Αφού είναι αναμενόμενο, τότε είναι. Example PIPELINE STALL