Αρχιτεκτονική Υπολογιστών

Γιώργος Δημητρίου

Ενότητα 6η:

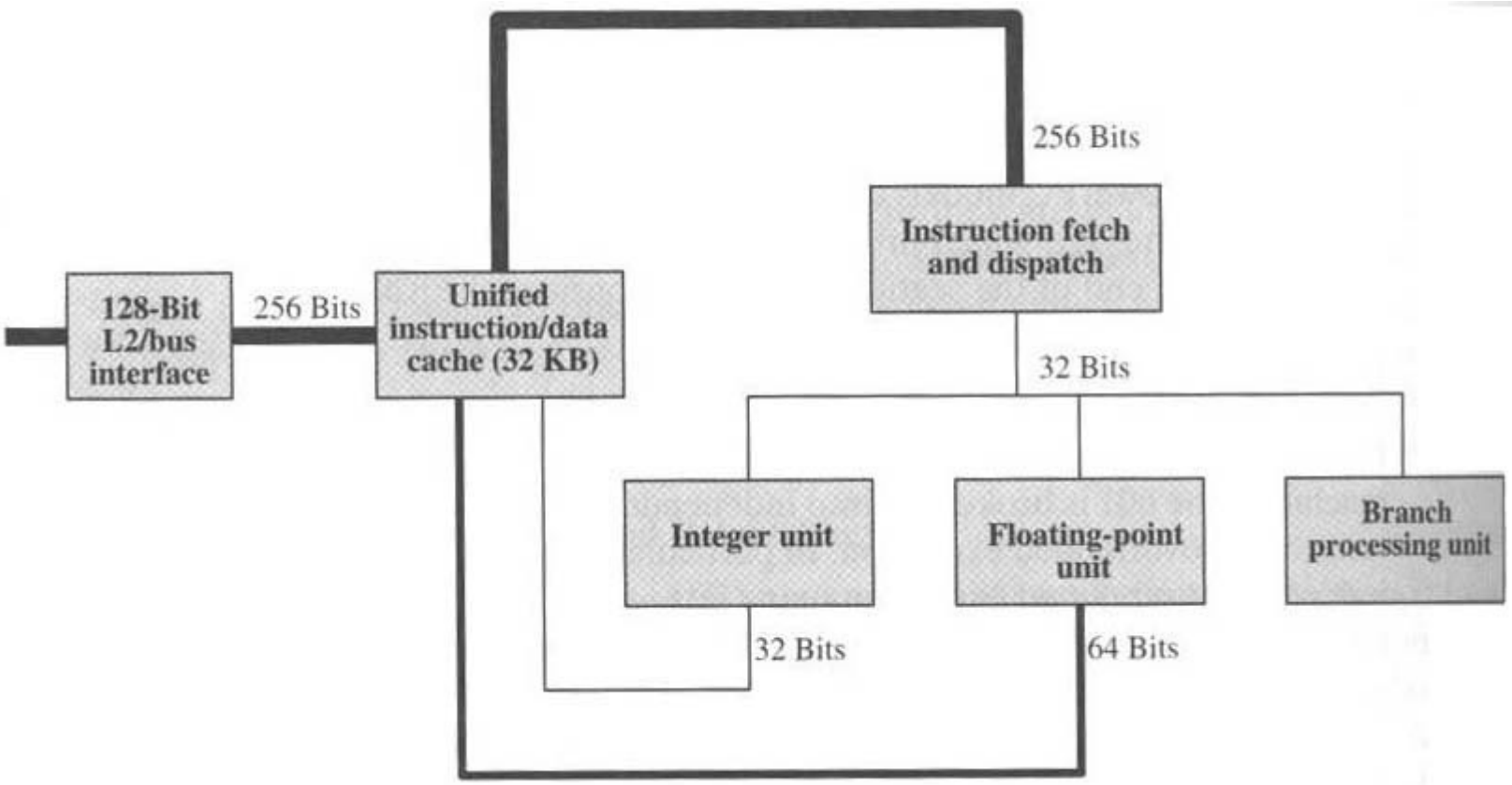
Επεξεργαστές με Δυναμική Δρομολόγηση Εντολών (PowerPC, Intel)

Πανεπιστήμιο Θεσσαλίας - Τμήμα Πληροφορικής

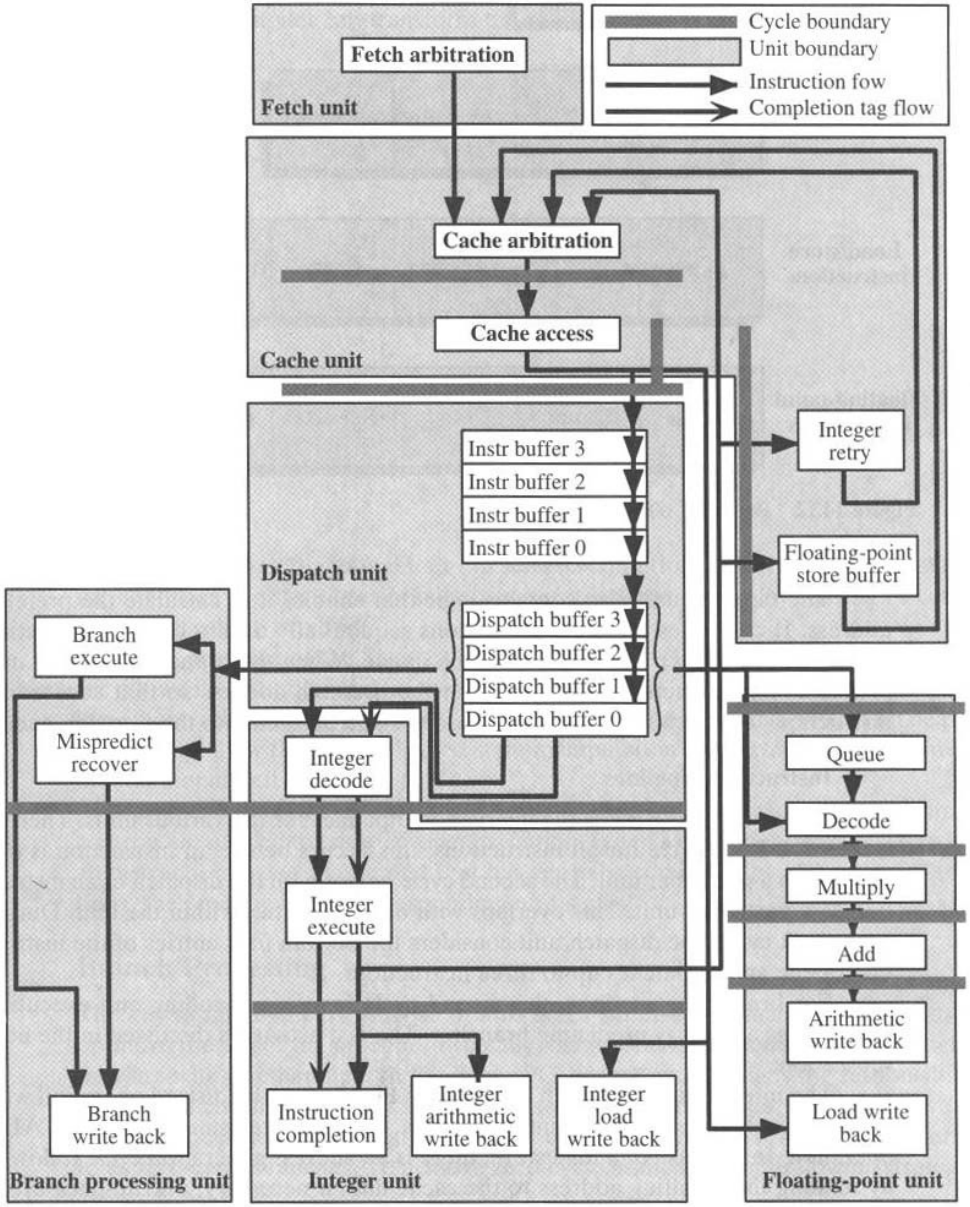
Αρχιτεκτονική PowerPC 601-620

- RISC
- Υπερβαθμωτή βαθμού 3-5
- Παράθυρο εντολών προς έκδοση
- Χωρίς/Με μετονομασία καταχωρητών
- Χωρίς/Με δυναμική πρόβλεψη διακλαδώσεων
- Κοινή/Διαχωρισμένη κρυφή μνήμη
- 1992-1997

Αρχιτεκτονική PowerPC 601



ΜΕΔ PowerPC 601



Κύκλος Εντολής

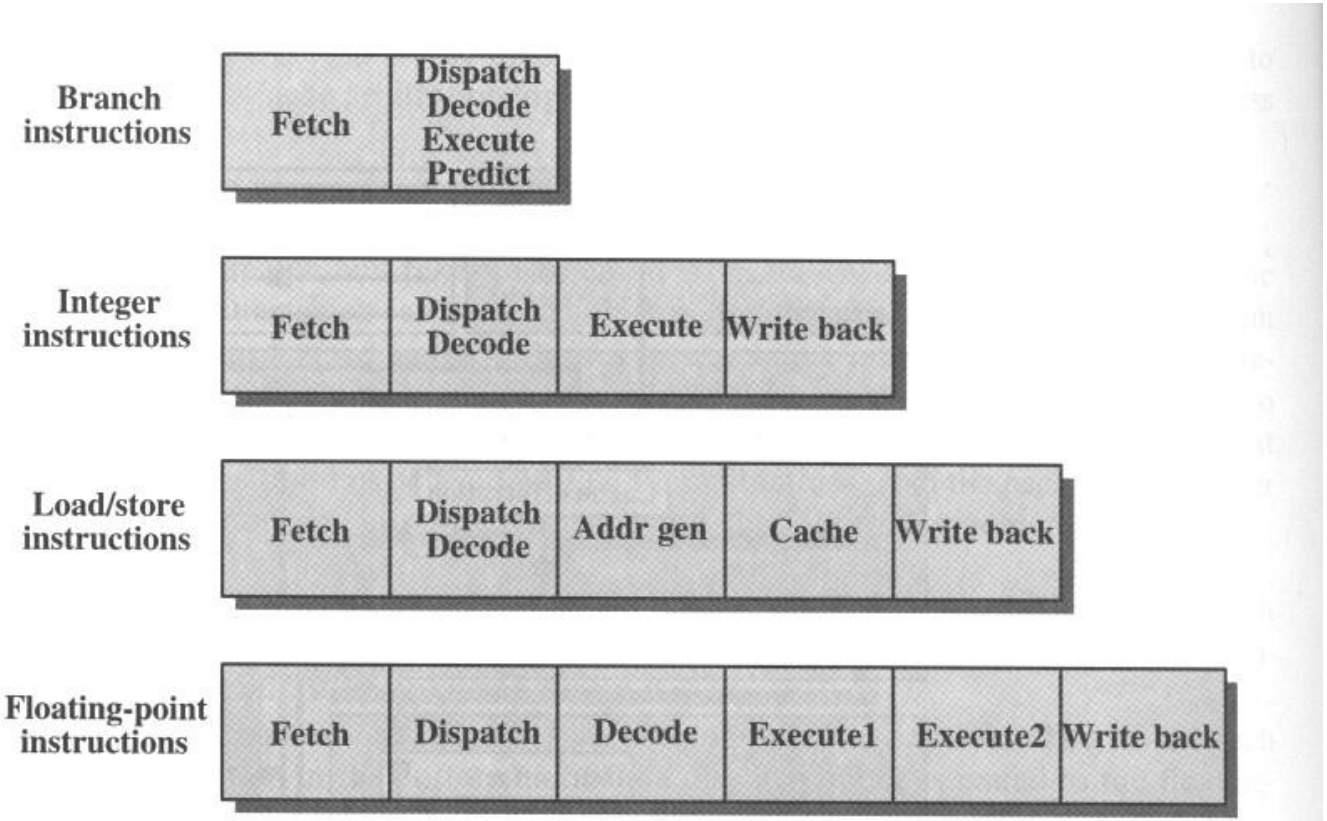


Figure 14.12 PowerPC 601 Pipeline

Περιγραφή ΜΕΔ

- 3 μονάδες εκτέλεσης επιτρέπουν ταυτόχρονη εκτέλεση μέχρι 3 εντολών
- Ουρά ανάθεσης (dispatch queue) 8 εντολών (4 ενεργές για ανάθεση)
- Δυναμική δρομολόγηση εντολών χωρίς σταθμούς δέσμευσης
- Στατική πρόβλεψη διακλαδώσεων
- Ενοποιημένη κρυφή μνήμη με μηχανισμό προτεραιότητας προσπελάσεων

Δρομολόγηση Εντολών

- Από τις 8 εντολές της ουράς, οι 4 παλαιότερες μόνο είναι υποψήφιες για ανάθεση
  - Δυναμική επιλογή έτοιμης εντολής για τις μονάδες διακλαδώσεων και κινητής υποδιαστολής
  - Στατική επιλογή της παλαιότερης εντολής για τη μονάδα σταθερής υποδιαστολής
- Εκτέλεση εκτός σειράς μόνο για διακλαδώσεις και εντολές κινητής υποδιαστολής

Παράδειγμα Κώδικα


```
if (a > 0)
        a = a + b + c + d + e;
else
        a = a - b - c - d - e;
```

(a) C code

```
                                #r1 points to a,
                                #r1+4 points to b,
                                #r1+8 points to c,
                                #r1+12 points to d,
                                #r1+16 points to e.
        lwz     r8=a(r1)        #load a
        lwz     r12=b(r1,4)     #load b
        lwz     r9=c(r1,8)      #load c
        lwz     r10=d(r1,12)    #load d
        lwz     r11=e(r1,16)    #load e
        cmpi    cr0=r8,0        #compare immediate
        bc      ELSE,cr0/gt=false  #branch if bit false
IF:
        add     r12=r8,r12      #add
        add     r12=r12,r9      #add
        add     r12=r12,r10     #add
        add     r4=r12,r11      #add
        stw     a(r1)=r4        #store
        b       OUT             #unconditional branch
ELSE:
        subf    r12=r12,r8      #subtract
        subf    r12=r9,r12      #subtract
        subf    r12=r10,r12     #subtract
        subf    r4=r12,r11      #subtract
        stw     a(r1)=r4        #store
OUT:
```

(b) Assembly code

Figure 14.13 Code Example with Conditional Branch [WEIS94]

Χρονισμός Επιτυχημένης Πρόβλεψης

			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	lwz	r8=a(r1)	F	D	E	C	W										
	lwz	r12=b(r1,4)	F	•	D	E	C	W									
	lwz	r9=c(r1,8)	F	•	•	D	E	C	W								
	lwz	r10=d(r1,12)	F	•	•	•	D	E	C	W							
	lwz	r11=e(r1,16)	F	•	•	•	•	D	E	C	W						
	cmpi	cr0=r8,0	F	•	•	•	•	•	D	E							
	bc	ELSE,cr0/gt=false	F	•	•	•	S										
IF:	add	r12=r8,r12	F	•	•	•	•	•	•	D'	E	W					
	add	r12=r12,r9			F	•	•	•	•	•	D	E	W				
	add	r12=r12,r10			F	•	•	•	•	•	•	D	E	W			
	add	r4=r12,r11									F	•	D	E	W		
	stw	a(r1)=r4									F	•	•	D	E	C	
	b	OUT															
ELSE:	subf	r12=r8,r12															
	subf	r12=r12,r9															
	subf	r12=r12,r10															
	subf	r4=r12,r11															
	stw	a(r1)=r4															
OUT:																	

(a) Correct prediction: Branch was not taken

Χρονισμός Αποτυχημένης Πρόβλεψης

			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	lwz	r8=a(r1)	F	D	E	C	W											
	lwz	r12=b(r1,4)	F	•	D	E	C	W										
	lwz	r9=c(r1,8)	F	•	•	D	E	C	W									
	lwz	r10=d(r1,12)	F	•	•	•	D	E	C	W								
	lwz	r11=e(r1,16)	F	•	•	•	•	D	E	C	W							
	cmpi	cr0=r8,0	F	•	•	•	•	•	D	E								
	bc	ELSE,cr0/gt=false	F	•	•	•	S											
IF:	add	r12=r8,r12	F	•	•	•	•	•	•	D'								
	add	r12=r12,r9			F	•	•	•	•	•								
	add	r12=r12,r10			F	•	•	•	•	•								
	add	r4=r12,r11																
	stw	a(r1)=r4																
	b	OUT																
ELSE:	subf	r12=r8,r12									F	D	E	W				
	subf	r12=r12,r9									F	•	D	E	W			
	subf	r12=r12,r10									F	•	•	D	E	W		
	subf	r4=r12,r11									F	•	•	•	D	E	W	
	stw	a(r1)=r4									F	•	•	•	•	D	E	C
OUT:																		

(b) Incorrect prediction: Branch was taken

PowerPC 620

- ΜΕΔ 64 bits
- 6 υπομονάδες εκτέλεσης
- Δυναμική πρόβλεψη διακλαδώσεων με τη βοήθεια BHT 2048 θέσεων
- Σταθμοί δέσμευσης
- Μετονομασία καταχωρητών για υποθετική εκτέλεση βάθους μέχρι 4 διακλαδώσεων

Αρχιτεκτονική PowerPC 750 (1997)

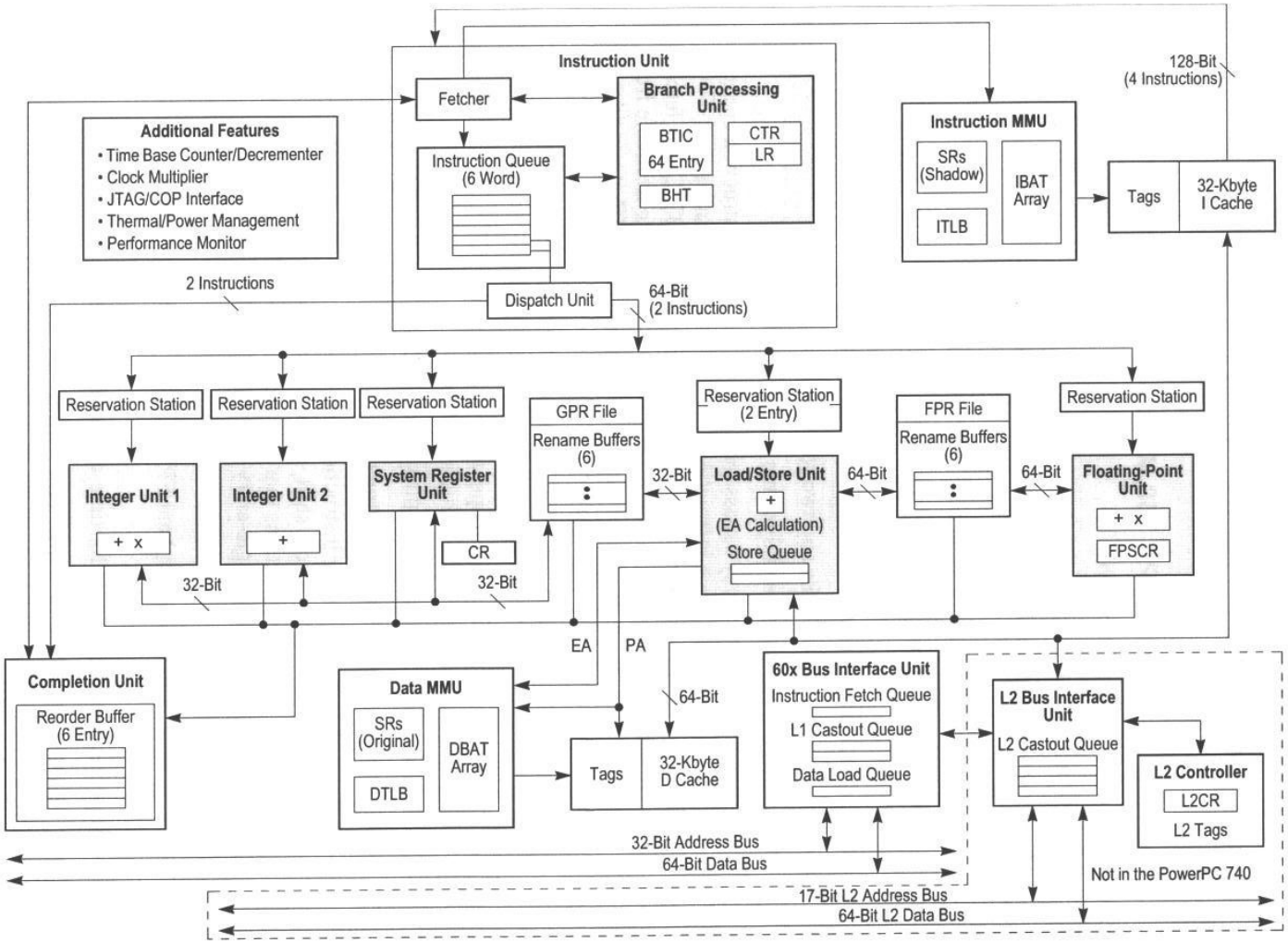
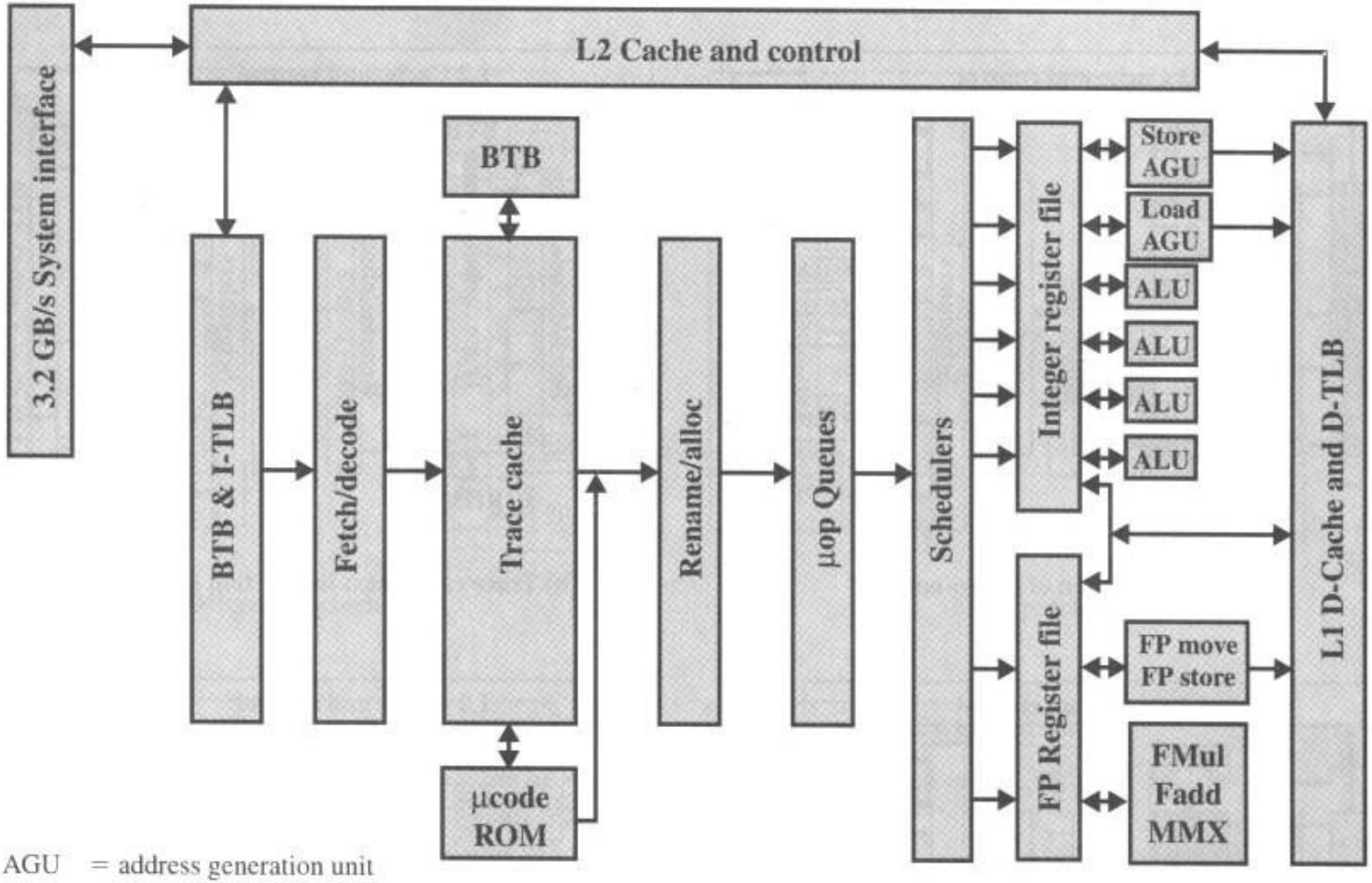


Figure 1-1. PowerPC 750 Microprocessor Block Diagram

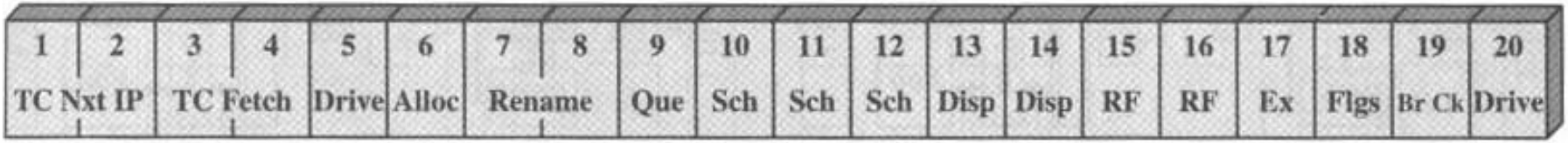
Αρχιτεκτονική Pentium 4 (Netburst, 2000-2006)



AGU = address generation unit
BTB = branch target buffer
D-TLB = data translation lookaside buffer
I-TLB = instruction translation lookaside buffer

Μερική Επικάλυψη Εντολών

- 20 βαθμίδες επικάλυψης (αργότερα 31):



TC Next IP = trace cache next instruction pointer
TC Fetch = trace cache fetch
Alloc = allocate
Rename = register renaming
Que = micro-op queuing
Sch = micro-op scheduling
Disp = Dispatch
RF = register file
Ex = execute
Flgs = flags
Br Ck = branch check

Figure 14.8 Pentium 4 Pipeline

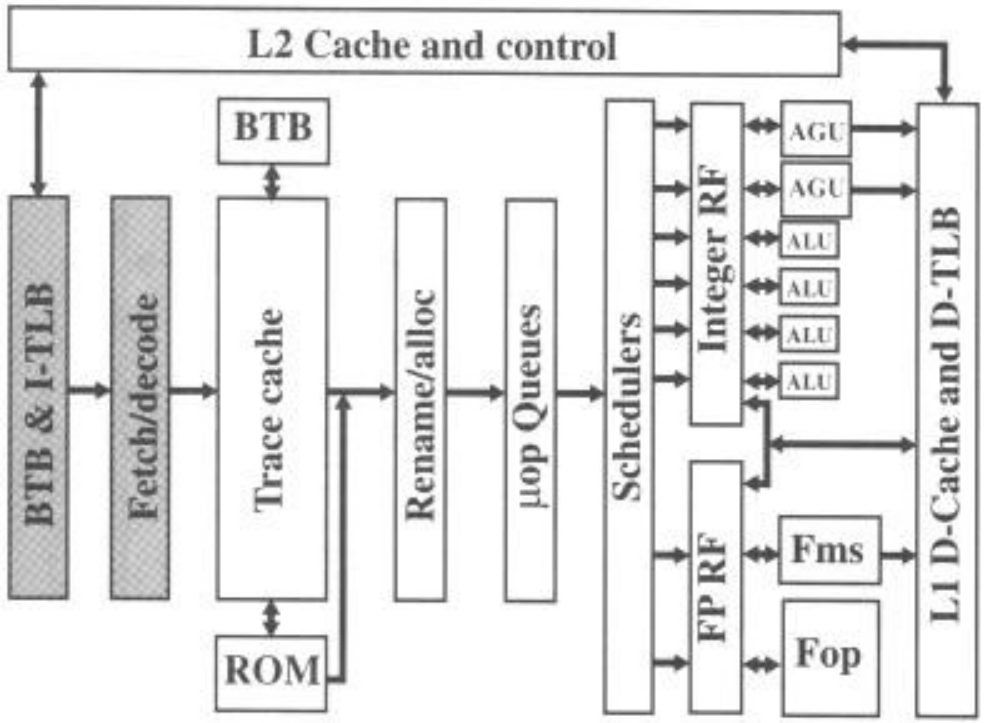
Περιγραφή ΜΕΔ

- Προσκόμιση εντολών IA-32, μετάφραση σε μ-ops, αποθήκευση στην L1 trace cache
- Δυναμική πρόβλεψη διακλαδώσεων: BTB 4-τρόπων συνόλου συσχέτισης, 512 γραμμές, 4 ψηφία ιστορίας ανά διακλάδωση
- Υποστήριξη για έκδοση και εκτέλεση εκτός σειράς, απόσυρση σε σειρά
- Πίνακας επαναδιάταξης (ROB) για 126 μ-ops, μετονομασία 128 καταχωρητών

Δρομολόγηση Εντολών

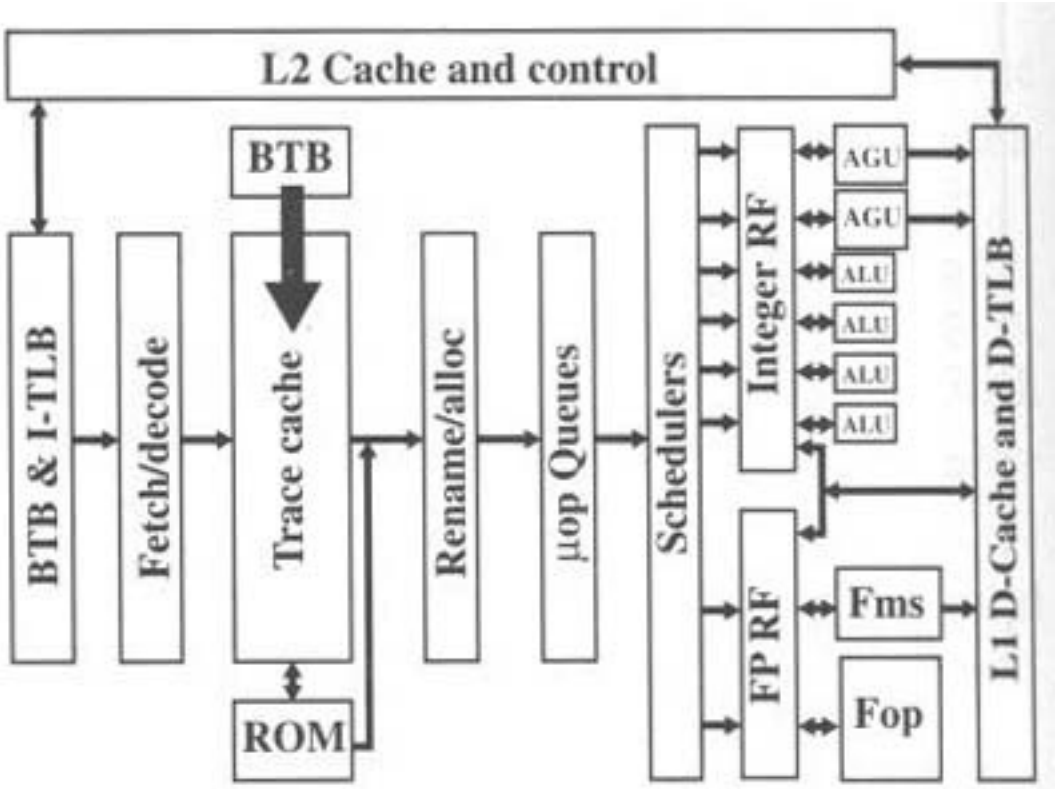
- Εισαγωγή από την trace cache στον ROB για μετονομασία καταχωρητών μέχρι 3 μ-ops / κύκλο μηχανής
- Εισαγωγή σε 2 ουρές αναμονής (εντολές με/χωρίς προσπέλαση μνήμης)
- Έκδοση σε σειρά για κάθε ουρά, εκτός σειράς μεταξύ ουρών και ανάθεση στις μονάδες εκτέλεσης μέχρι 6 μ-ops / κύκλο μηχανής για εκτέλεση εκτός σειράς
- Απόσυρση μέχρι 3 μ-ops / κύκλο μηχανής

Προσκόμιση Εντολών IA-32



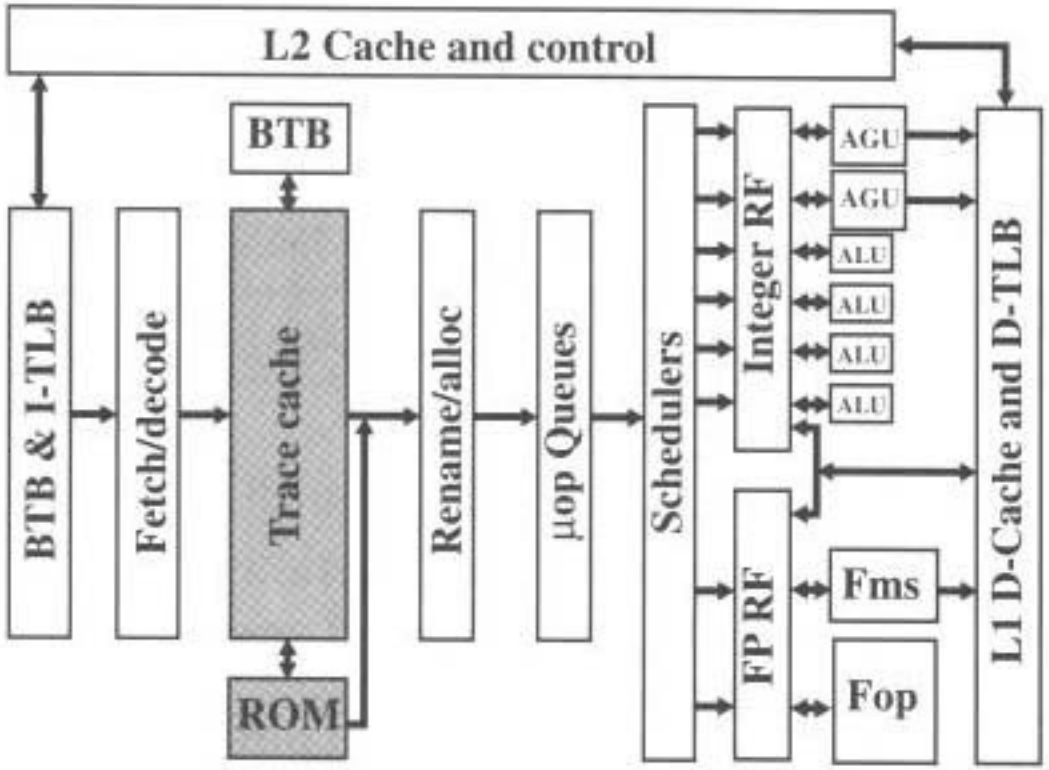
(a) Generation of micro-ops

Προσπέλαση BTB



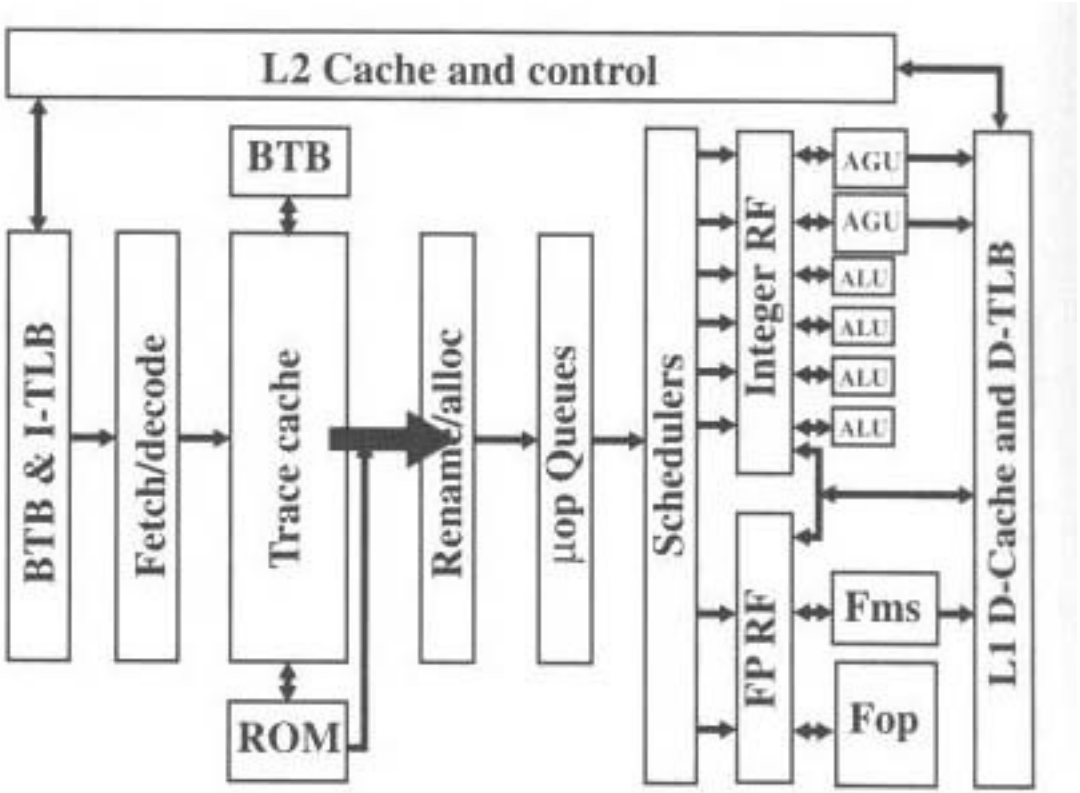
(b) Trace cache next instruction pointer

Προσπέλαση Trace Cache



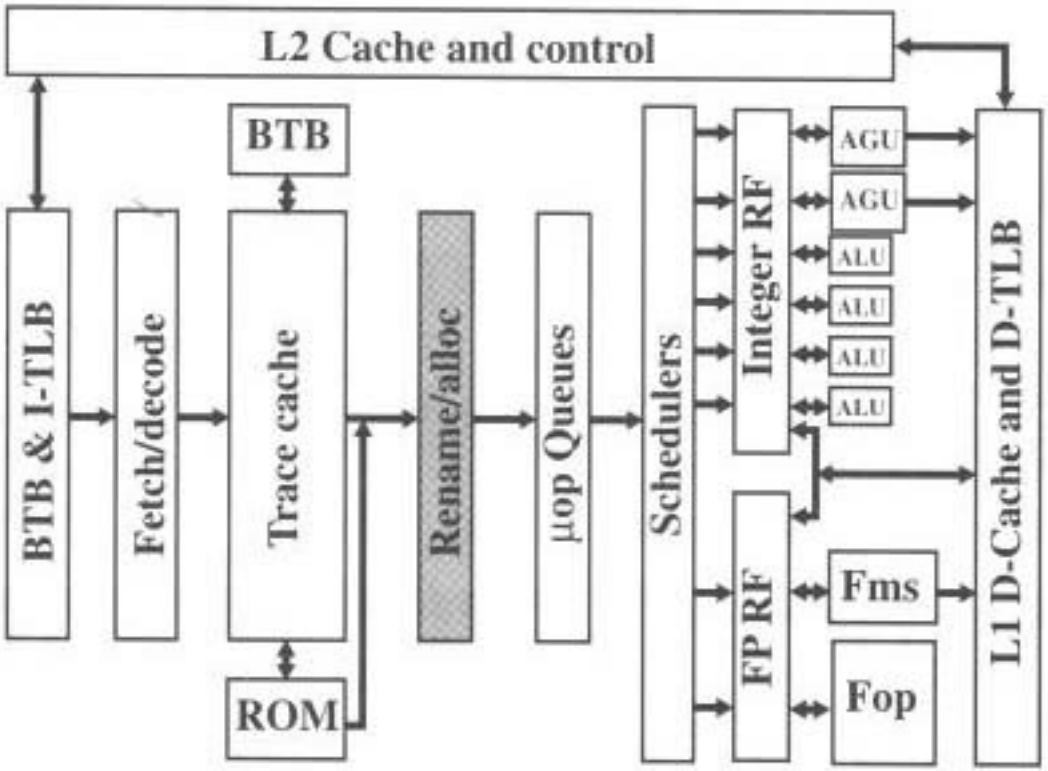
(c) Trace cache fetch

Προσκόμιση μ.Εντολών



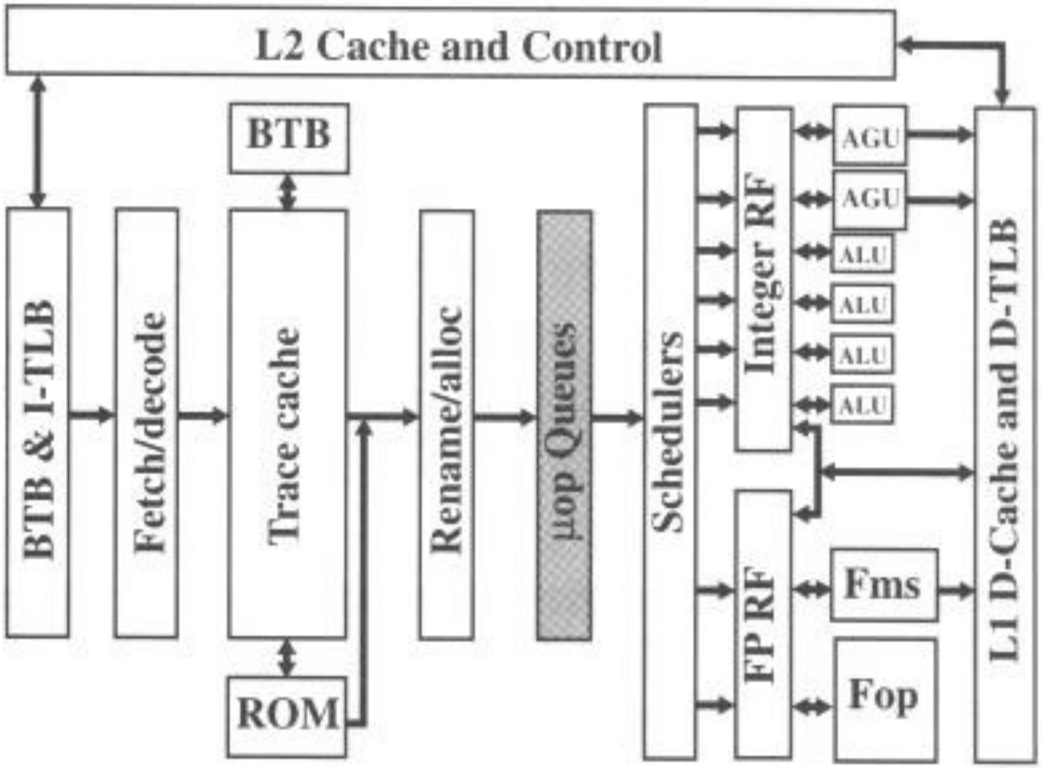
(d) Drive

Μετονομασία Καταχωρητών



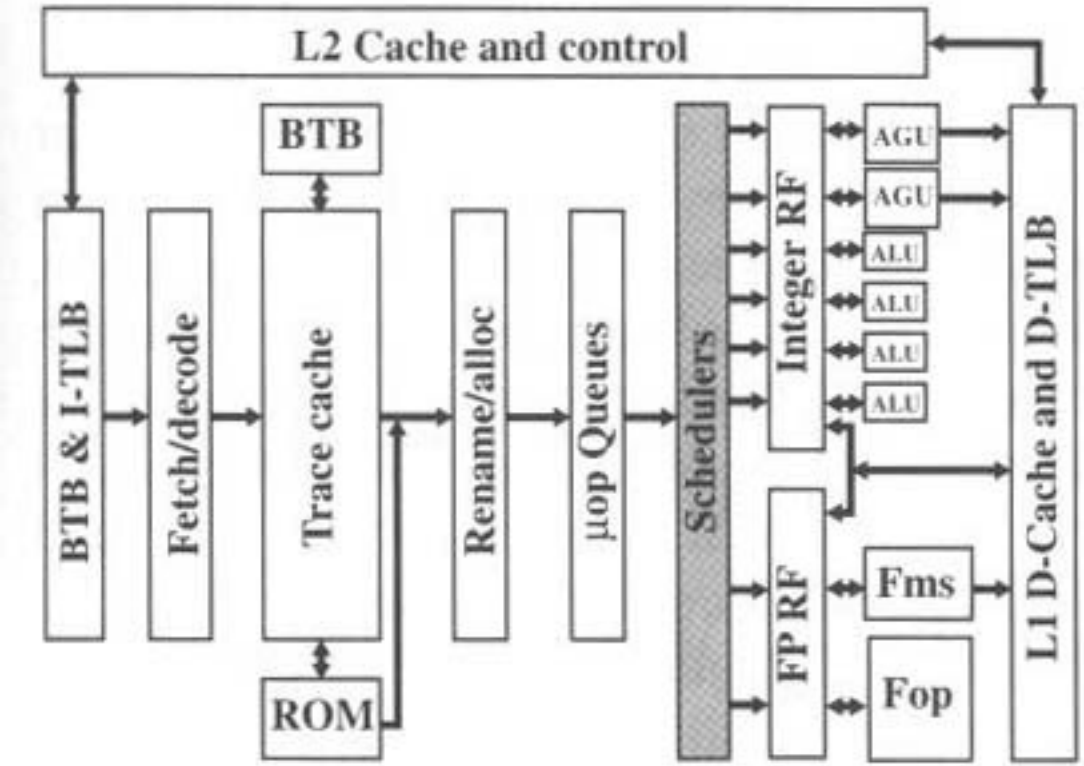
(e) Allocate; Register renaming

Εισαγωγή στις Ουρές



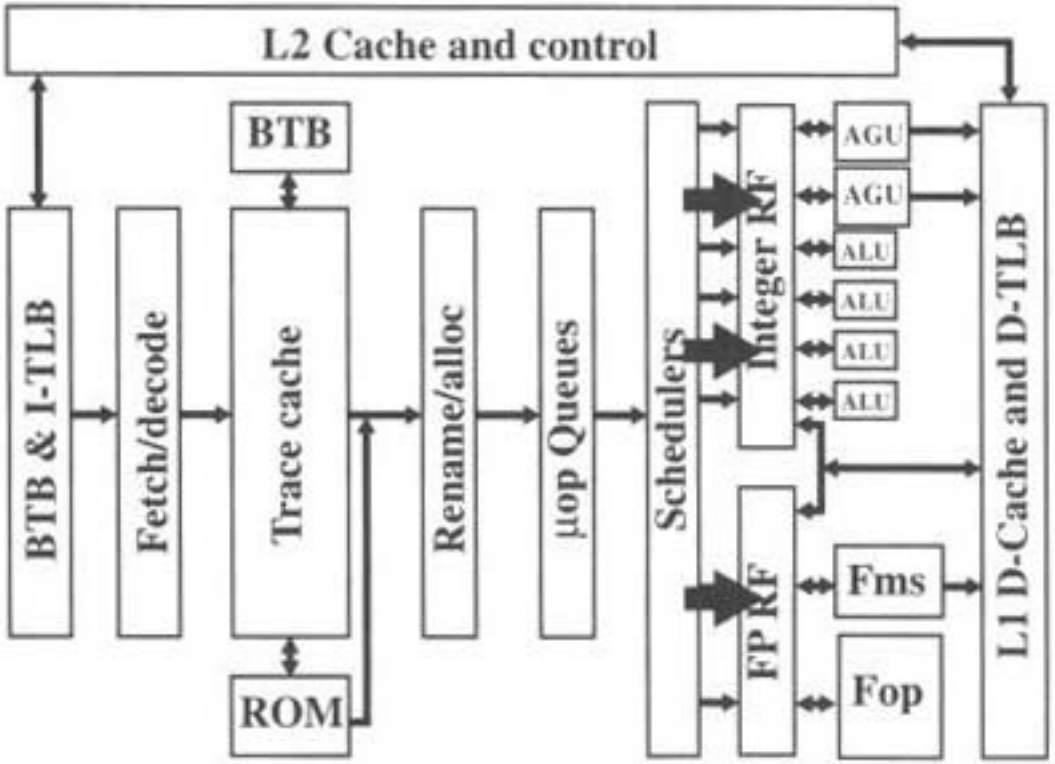
(f) Micro-op queuing

Δρομολόγηση



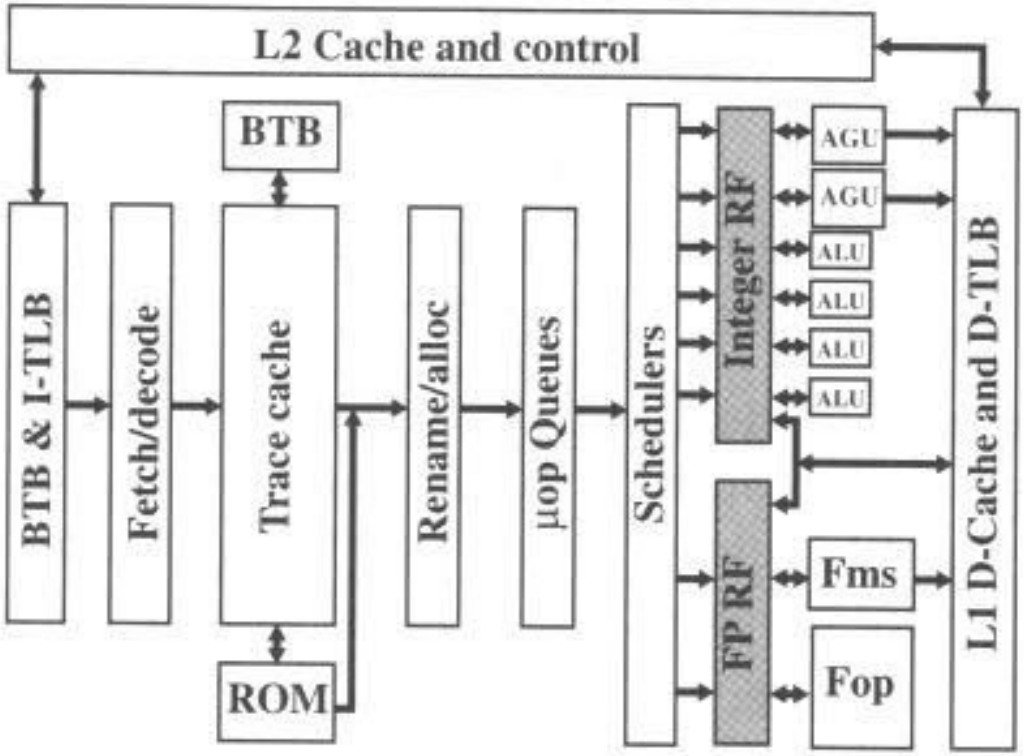
(g) Micro-op scheduling

Ανάθεση στις Μονάδες Εκτέλεσης



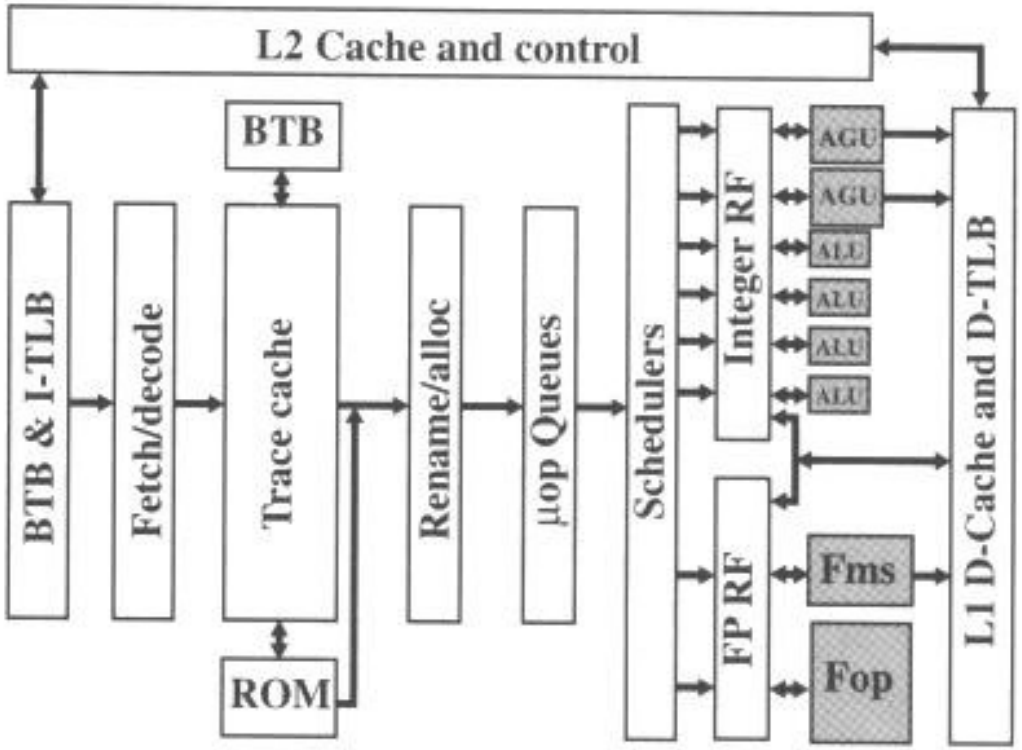
(h) Dispatch

Ανάγνωση Τελούμενων



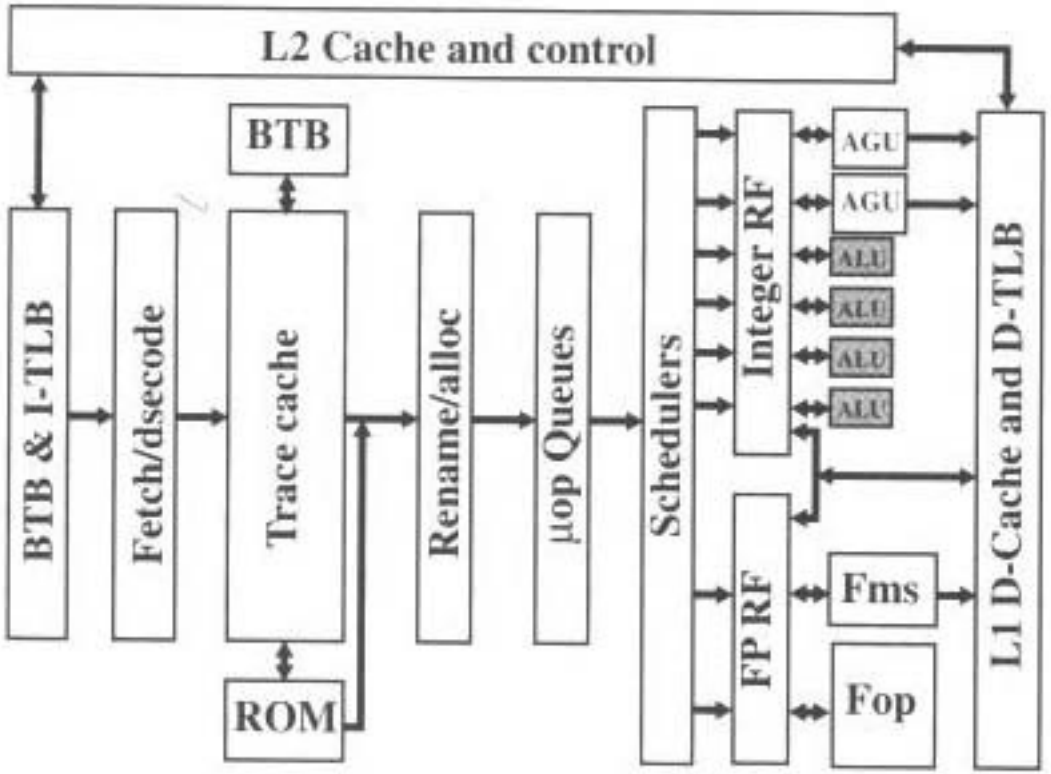
(i) Register file

Εκτέλεση



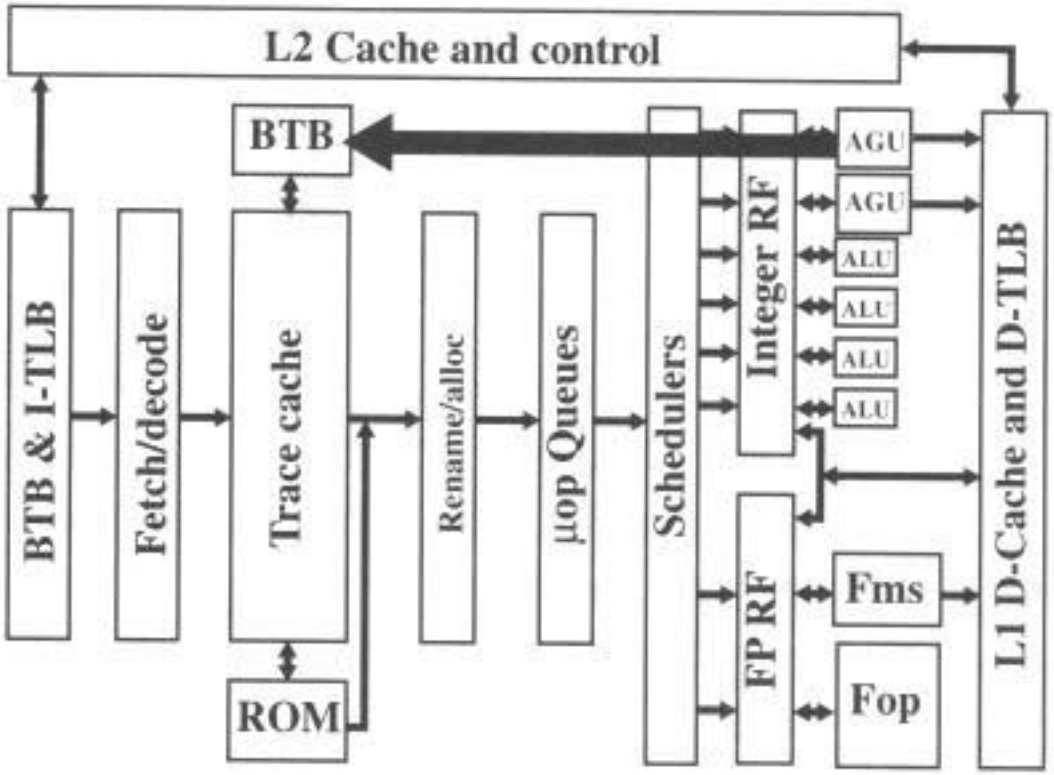
(j) Execute; flags

Έλεγχος Διακλαδώσεων



(k) Branch check

Αποθήκευση στην BTB



(l) Branch check result

Τι έγινε ο Pentium 4;

- Αποτυχία?
  - Μέτρια απόδοση?
  - Υψηλή κατανάλωση ισχύος?
    - Υπερθέρμανση (καλοριφέρ?)
  - Καμία επεκτασιμότητα?
- Επιλογή νέας αρχιτεκτονικής!
  - Στροφή στις αρχιτεκτονικές πολλών πυρήνων (Core, Core 2, Core i3/i5/i7)
    - Κάθε πυρήνας πιο απλός από τον Pentium 4
    - Δυνατότητες απενεργοποίησης πυρήνων

Αρχιτεκτονική i3/i5/i7 (Nehalem)