Extraction d'instructions

Plan

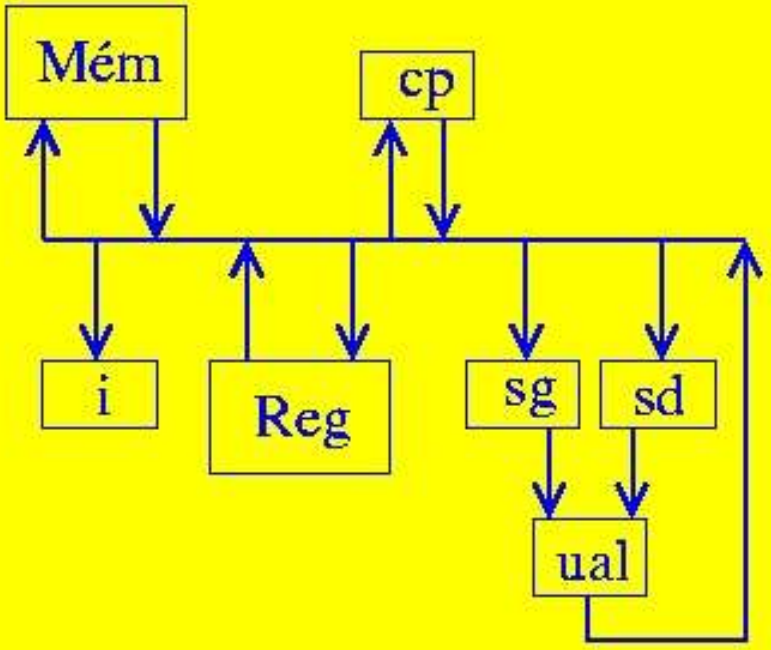
Extraire instruction par instruction dans une machine microprogrammée

Extraire une instruction par cycle dans une machine pipelinée

Extraire une ligne de cache par cycle dans une machine spéculative

Extraire une ligne de cache de trace par cycle dans une machine spéculative

Extraire instruction par instruction dans une machine microprogrammée



Cycle = accès à la mémoire

Une instruction = un microprogramme

Le microprogramme contrôle le bus

Phases:

- Extraction (i:= Mém[cp])
- Lecture source gauche (sg:= Mém/Reg)
- Lecture source droite (sd:= Mém/Reg)
- Calcul (Mém/Reg:= sg op sd)
- Cp suivant (cp:= cp+1/sg op sd/Mém)

Extraire instruction par instruction dans une machine microprogrammée

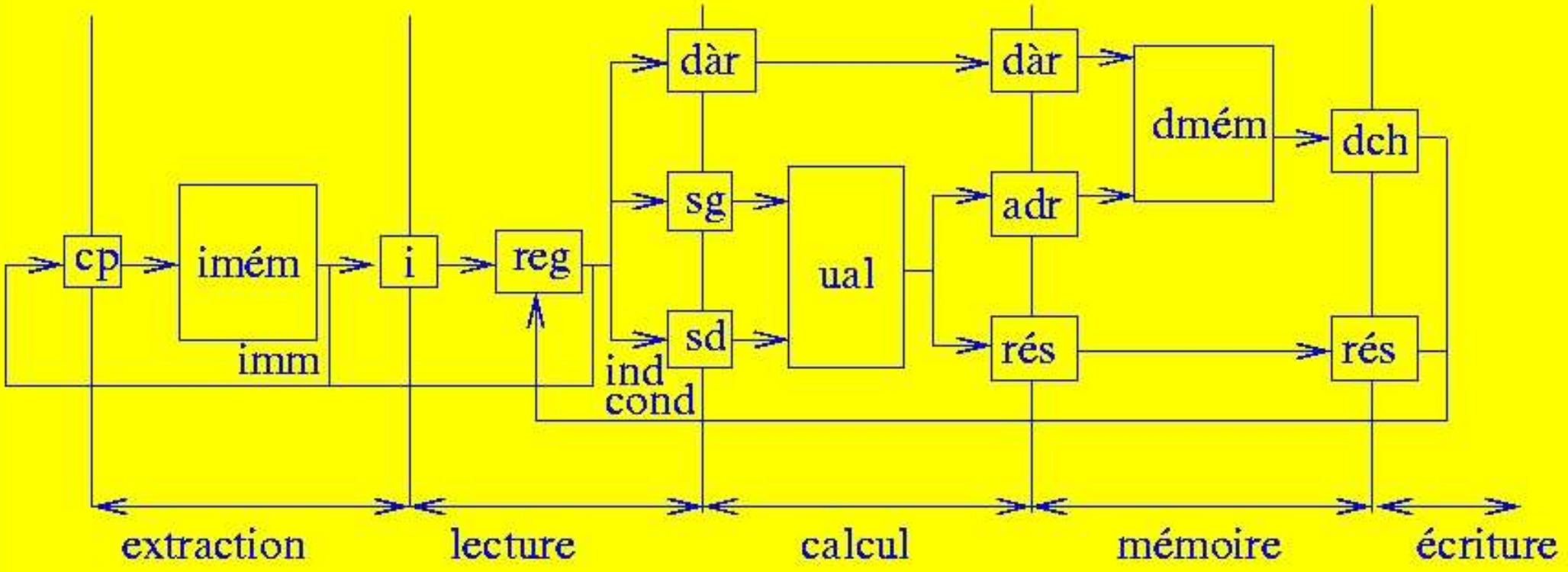
Avantages

Matériel simple: bus partagé, contrôlé par microprogramme
ISA riche (CISC)

Inconvénient:

Lent (plusieurs cycles par instruction)

Extraire une instruction par cycle dans une machine pipelinée



Extraire une instruction par cycle dans une machine pipelinée

Impact sur l'ISA:

Uniformisation du format des instructions: 2 sources, 1 destination

Architecture chargement/rangement

Délai pour les sauts conditionnels et indirects

Jeu d'instruction réduit (RISC)

(exemple: if (x==y) ... traduit par):

```
R1=(Rx==Ry)
BZ R1, étiquette
...
```

Impact sur la microarchitecture:

Uniformisation de la durée des instructions (longueur du pipeline)

Séparation des mémoires de donnée et d'instruction

Extraire une instruction par cycle dans une machine pipelinée

extraction a BZ R1,e	lecture R1		
	extraction a+1 (délai)		
		extraction e	
			extraction e+1

cp=a

cp=a+1
R1=0

cp=e

cp=e+1

Extraire une instruction par cycle dans une machine pipelinée

Cycle = accès à la mémoire

Une instruction = 5 cycles

L'instruction contrôle les bus formant le pipeline

5 Phases:

- Extraction (i:= Mém[cp])
- Lecture sources (sg:= Reg; sd:= Reg)
- Calcul (Rés/Adr:= sg op sd)
- [Accès mémoire (Mém[Adr]:= dàr ou dch:=Mém[Adr])]
- Ecriture résultat (Reg:= Rés/dch)

Extraire une instruction par cycle dans une machine pipelinée

Avantages:

CPI = 1 + aléas

Matériel simple

Inconvénient:

Toutes les opérations ont la même latence (profondeur du pipeline)

Extraire une instruction par cycle dans une machine pipelinée

Traitement des sauts:

saut conditionnel (si g = 0 vers e)

lecture de g et test de nullité et calcul de cp+e
envoi de cp+e et de la condition

saut avec lien (appel f)

calcul de cp+1 et envoi de f
sauvegarde de cp+1 (r31:= cp+1)

retour

restauration dans cp (cp:= r31)

Extraire une ligne de cache par cycle dans une machine spéculative

Pour obtenir un CPI < 1 il faut:

extraire plus d'une instruction par cycle (superscalaire)

Contraintes des machines superscalaires:

plusieurs pipelines de longueurs différentes
dépendances structurelles => files d'attente
très peu d'ILP (dépendances de contrôle et de données)
délai variable pour le calcul des sauts

Pour exploiter plus d'ILP il faut:

permettre une exécution en désordre
prédire les sauts

La lecture en cache L1 prend 2 cycles:

prédire le cp suivant sans examen des instructions extraites

Extraire une ligne de cache par cycle dans une machine spéculative

En examinant les instructions extraites, on calcule cpsv:

pas de saut dans la ligne à partir de cp: cpsv = ligne suivante
le premier saut s à partir de cp est inconditionnel immédiat: cpsv = k(s)
le premier saut s à partir de cp est conditionnel et prédit pris: cpsv = k(s)
le premier saut s1 à partir de cp est conditionnel et prédit non pris: cpsv = s2
(s2 est le second saut dans la même ligne ou à défaut le début de ligne suivante)
le premier saut s1 à partir de cp est un retour: cpsv = pile[sp]
le premier saut s1 à partir de cp est indirect: prédire cpsv

Sans examen des instructions à extraire, on prédit cpsv:

on conserve des couples (saut, cible) dans un cache: le BTB
le BTB doit être accédé en un cycle soit actuellement un cache de 8KO

Extraire une ligne de cache par cycle dans une machine spéculative

Pour un cache d'instruction (lignes de 2^n instructions):

un bloc de base (BB) est un groupe d'instructions en séquence tel que:

- le BB figure tout entier dans une ligne de cache
- le BB ne comporte qu'au plus un saut
- si le BB comprend un saut, celui–ci en est la dernière instruction

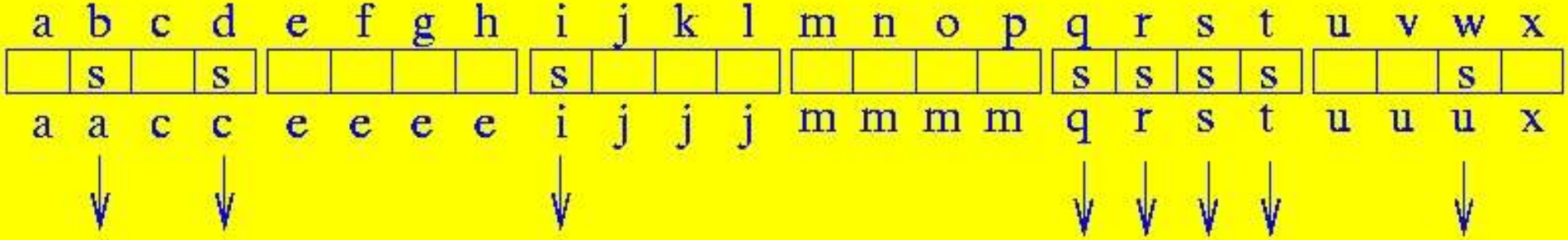
un BB compte entre 1 et 2^n instructions

Le BTB est un cache de lignes de 2^n couples (saut, cible):

un saut d'adresse a entre en colonne (a mod 2^n)
le BTB peut fournir en parallèle les cibles de tous les sauts d'une ligne de cache

Un BTB de 256 lignes de 4 couples occupe 7Koctets

Extraire une ligne de cache par cycle dans une machine spéculative

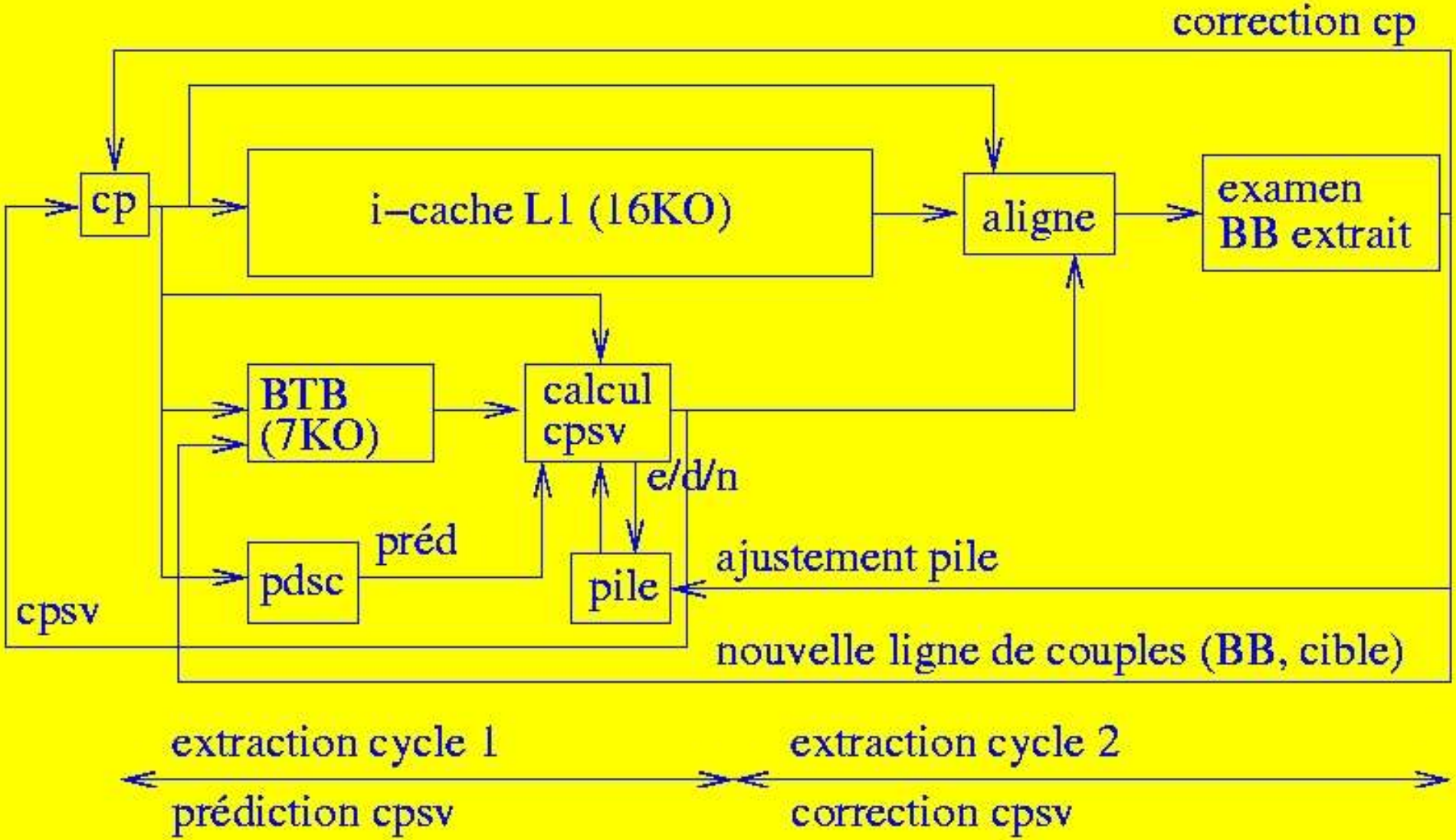


BTB mis à jour (BTB[saut]:= étiquette(saut), cible(saut), type(saut))

A chaque extraction d'un BB terminé par un saut,
la cible est mémorisée dans le BTB
(pour un saut conditionnel, la cible est celle du saut pris)
(pour un retour, aucune cible n'est mémorisée; cible en pile matérielle)
(pour un saut indirect, la cible fournira une prédiction)

BTB[(b%1024)/4][1]:= (b/1024, cible(b), type(b))
BTB[(d%1024)/4][3]:= (d/1024, cible(d), type(d))
...
BTB[(w%1024)/4][2]:= (w/1024, cible(w), type(w))

Extraire une ligne de cache par cycle dans une machine spéculative



Extraire une ligne de cache par cycle dans une machine spéculative

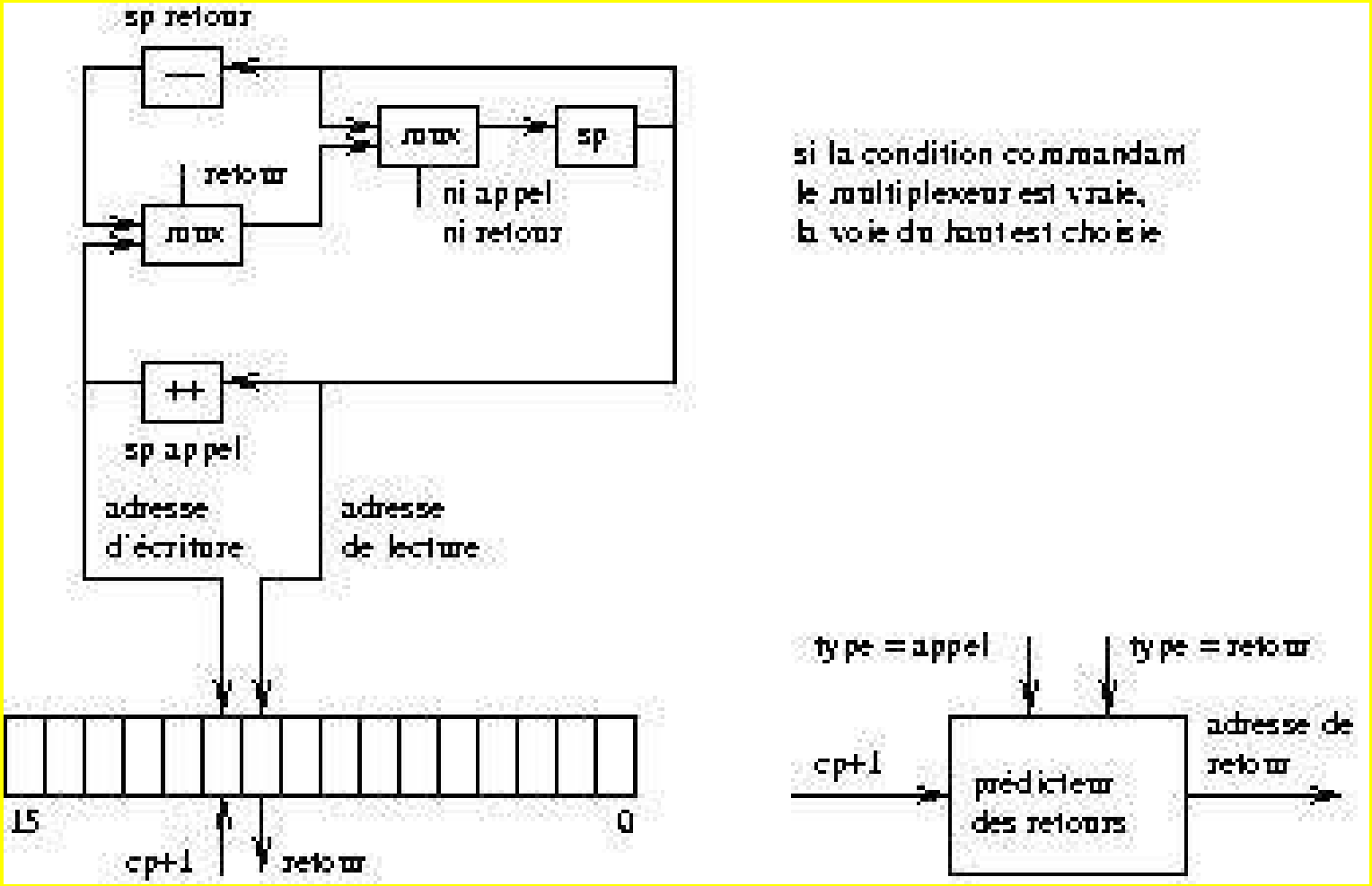
La pile matérielle est mise à jour au premier cycle:

si le BTB indique que le BB extrait est terminé par un retour: dépilement
si le BTB indique que le BB extrait est terminé par un appel: empilement
(on empile cp + longueur(BB))
autrement, la pile est inchangée

La pile matérielle est corrigée au second cycle:

si le BB extrait contient un retour non signalé par le BTB: dépilement
(l'adresse dépilée est envoyée pour corriger le cp)
si le BB extrait contient un appel non signalé par le BTB: empilement
(on empile l'adresse de l'instruction qui suit l'appel en séquence)
(on envoie l'adresse appelée pour corriger le cp)
(pour un appel indirect, la pile est corrigée mais pas le cp)

Prédiction des retours

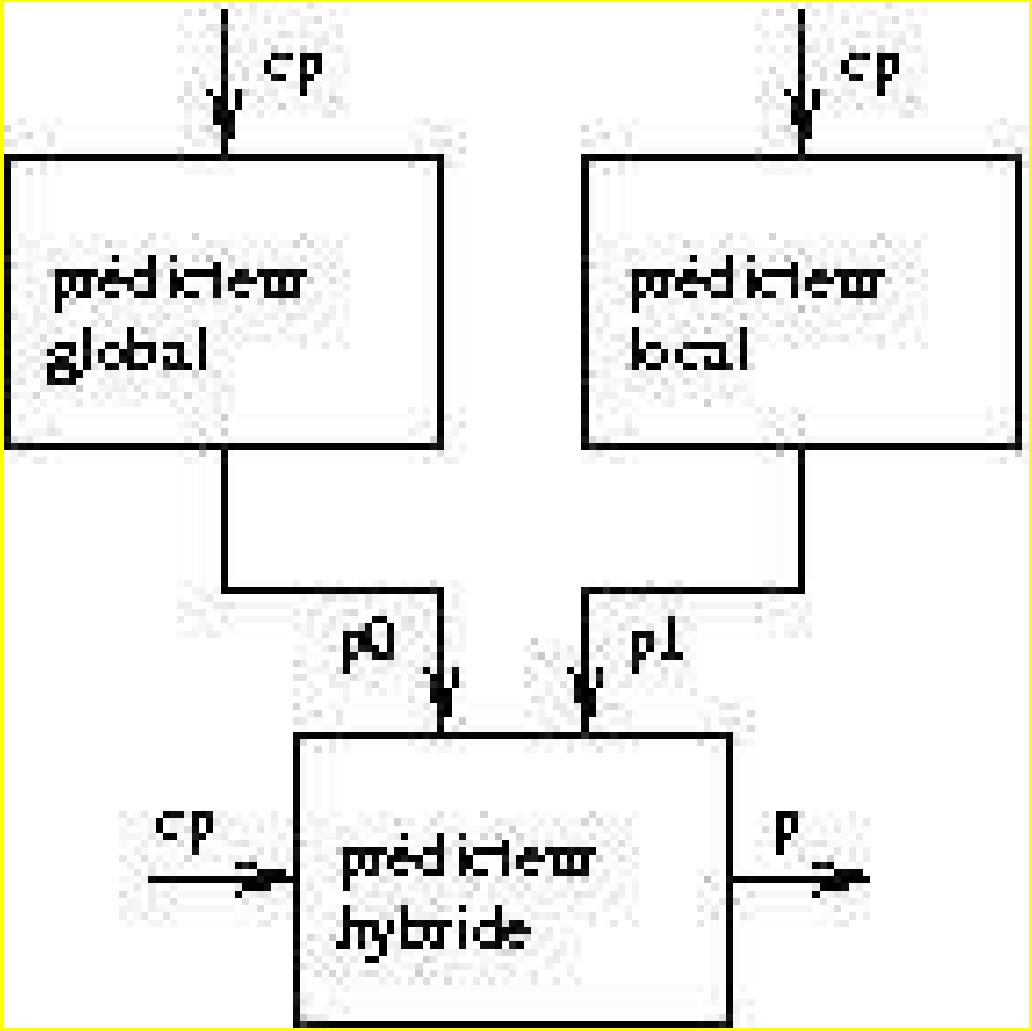


Le prédicteur adresse une pile (mémoire à 2 ports)

On empile à chaque appel, on dépile à chaque retour

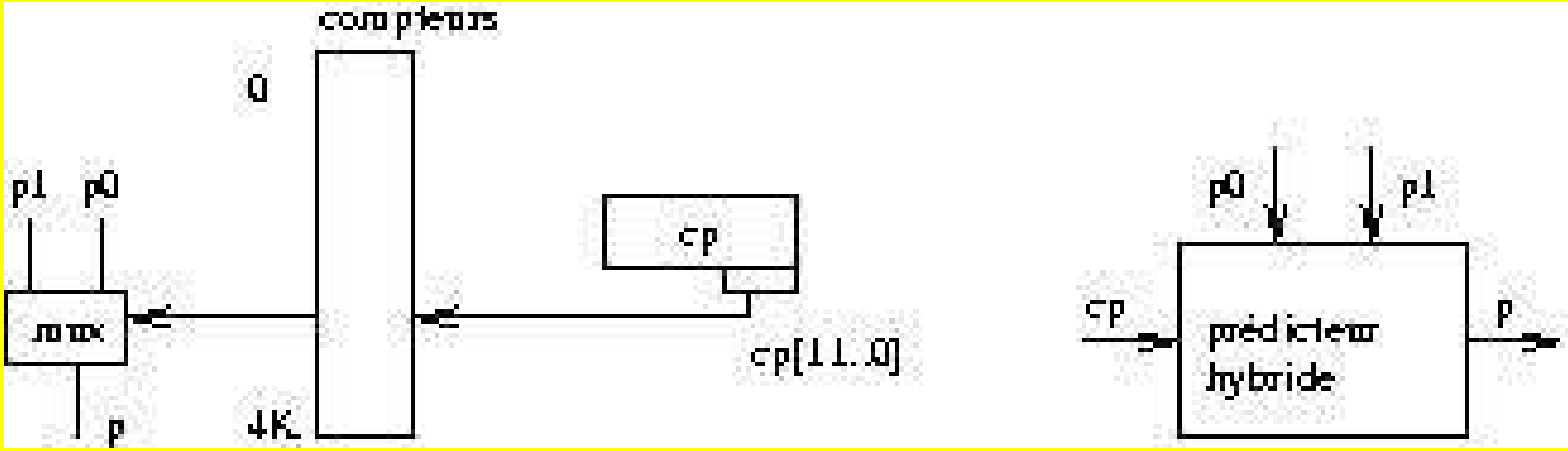
(attention: pas de détection de débordement)

Prédiction de la direction des sauts conditionnels



La direction prédite p est choisie parmi deux prédictions issues de prédicteurs spécialisés et basées sur le cp et sur le comportement antérieur des sauts conditionnels

Prédicteur hybride



Une table de compteurs 2 bits à saturation adressée par la partie basse de cp

(la table est un cache sans étiquette)

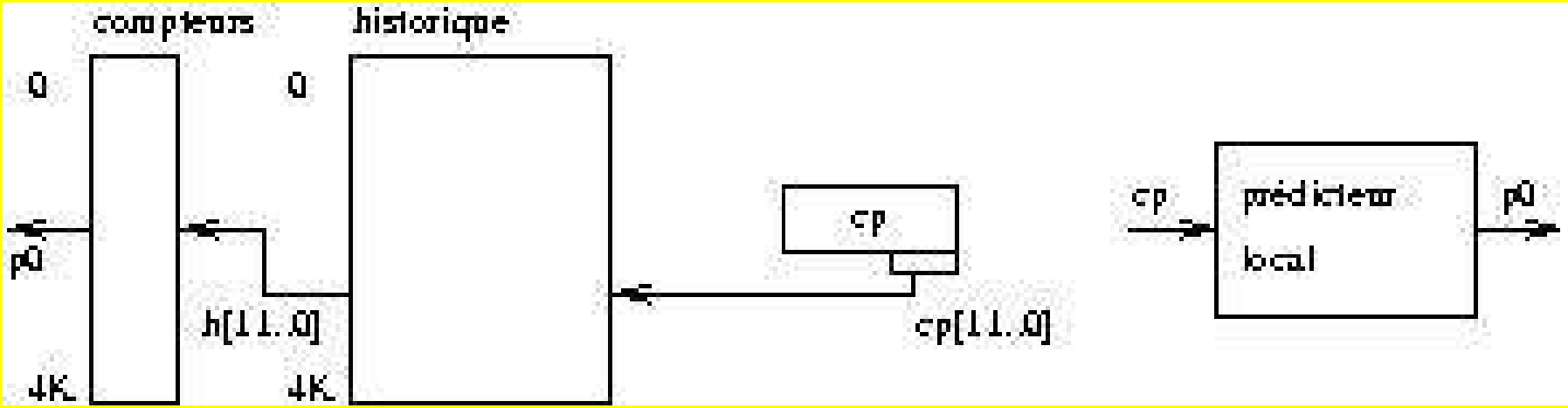
Le bit fort du compteur adressé fixe le choix

compteur++ ⇘ p0 != p1 et p1 correcte

compteur-- ⇘ p0 != p1 et p0 correcte

UP VD

Prédicteur local



L'historique est un mot de 12 bits correspondant aux 12 dernières directions d'un saut

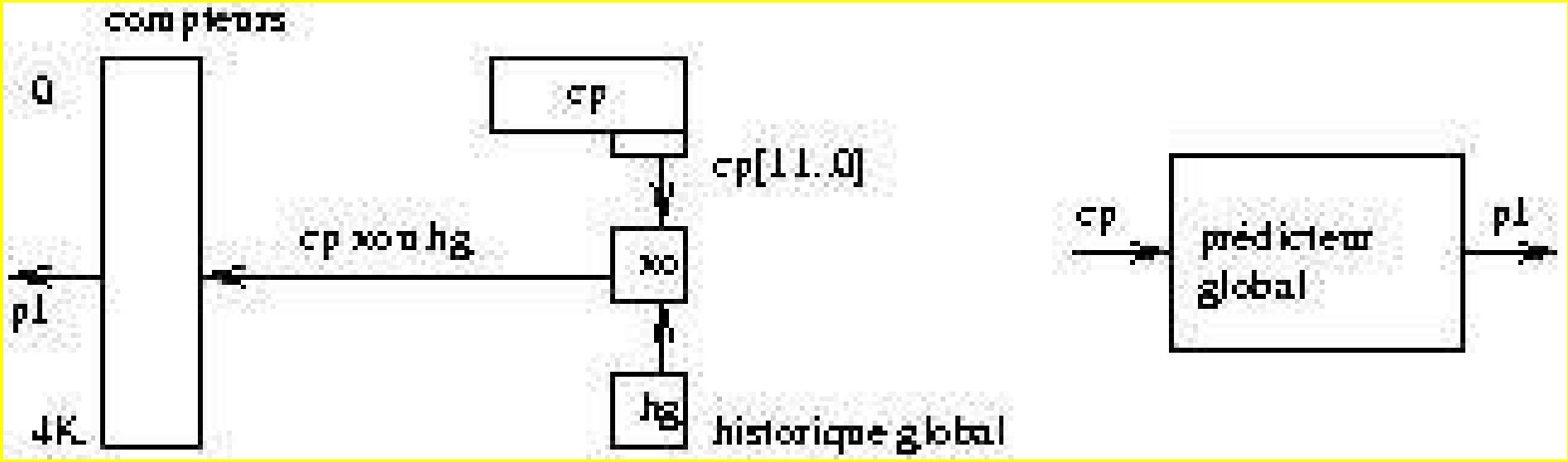
Le motif obtenu adresse un cache de compteurs

Le bit fort issu du cache est la prédiction

compteur++/-- ⇘ saut pris/saut non pris

Le prédicteur local prédit bien les sorties de boucles

Prédicteur global



L'historique global est un mot de 12 bits formé des directions des 12 derniers sauts conditionnels

Le mélange de cp et de l'historique adresse un cache de compteurs deux bits à saturation

Le bit fort du compteur adressé est la direction prédite

compteur++/-- ↘ saut pris/saut non pris

Le prédicteur global prédit bien les sauts corrélés

UP VD

Extraire une ligne de cache de trace par cycle dans une machine spéculative

Dans une ligne de cache de 4i, il y a en général un seul BB

Pour extraire plus d'un BB à la fois, il faut:

soit lire plusieurs ligne à la fois, ce qui implique un cache multi–banc
soit disposer de plusieurs BB dans une ligne, c'est–à–dire un cache de trace

un cache multi–banc ne permet pas de lire deux lignes dans le même banc
(ce qui est le cas quand on doit lire deux fois la même ligne)

Un cache de trace contient des traces

Une trace est une juxtaposition de BB

trois sauts conditionnels au plus (prédicteur multi–saut)
2^n instructions au plus (taille de la ligne de cache)

Extraire une ligne de cache de trace par cycle dans une machine spéculative

Contraintes supplémentaires sur les traces:

un BB (terminé par un saut ou une fin de ligne) ne peut être coupé
(il entre tout entier dans la trace de 2^n instructions ou dans la suivante)

deux traces du cache ne peuvent avoir le même préfixe
(si la trace AB est cachée, la trace AC ne l'est pas)

Constitution des traces:

les traces sont construites à la fin de l'extraction à partir des BB prédits

un BB extrait est ajouté en queue de la trace en cours de construction ssi:
- la nouvelle trace a au plus 2^n instructions
- la nouvelle trace a au plus 3 sauts immédiats

dans le cas contraire, la trace ancienne est écrite dans le cache
et le BB extrait entre dans une nouvelle trace vierge
(sauf s'il se termine par un saut indirect: dans ce cas, il est écarté)

Extraire une ligne de cache de trace par cycle dans une machine spéculative

Constituants d'une trace:

- l'adresse de son BB de départ (index et étiquette dans le cache)
- la direction des trois sauts immédiats de la trace ("pris" pour un saut inconditionnel)
- l'adresse de l'instruction suivante (dernier saut pris)
- l'adresse de l'instruction suivante (dernier saut non pris)
- le nombre de sauts immédiats de la trace
- l'indication que la trace est ou n'est pas terminée par un saut immédiat

Un cache de 64 traces occupe 4,7Koctets

Extraire une ligne de cache de trace par cycle dans une machine spéculative

Fonctionnement de l'extracteur basé sur un cache de trace:

une trace est extraite du cache à l'adresse (cp mod taille)

un prédicteur multi–saut délivre une prédiction de la direction des trois prochains sauts immédiats

ces prédictions sont confrontées aux directions des sauts immédiats de la trace extraite (à concurrence du nombre de tels sauts dans la trace)

l'accès au cache est un succès ssi:

- l'étiquette de la ligne lue est cp/taille
- les directions des saut immédiats coincident avec les prédictions

en cas d'échec, un extracteur standard fournit le BB suivant et le cpsv

en cas de succès, le cpsv est fourni par le cache de trace

Extraire une ligne de cache de trace par cycle dans une machine spéculative

Cas d'échec:

saut indirect ou retour
(les sauts indirects sont prédits par l'extracteur standard)
succès partiel (le cache fournit un préfixe du chemin prédit)
échec d'étiquette (nouveau chemin)

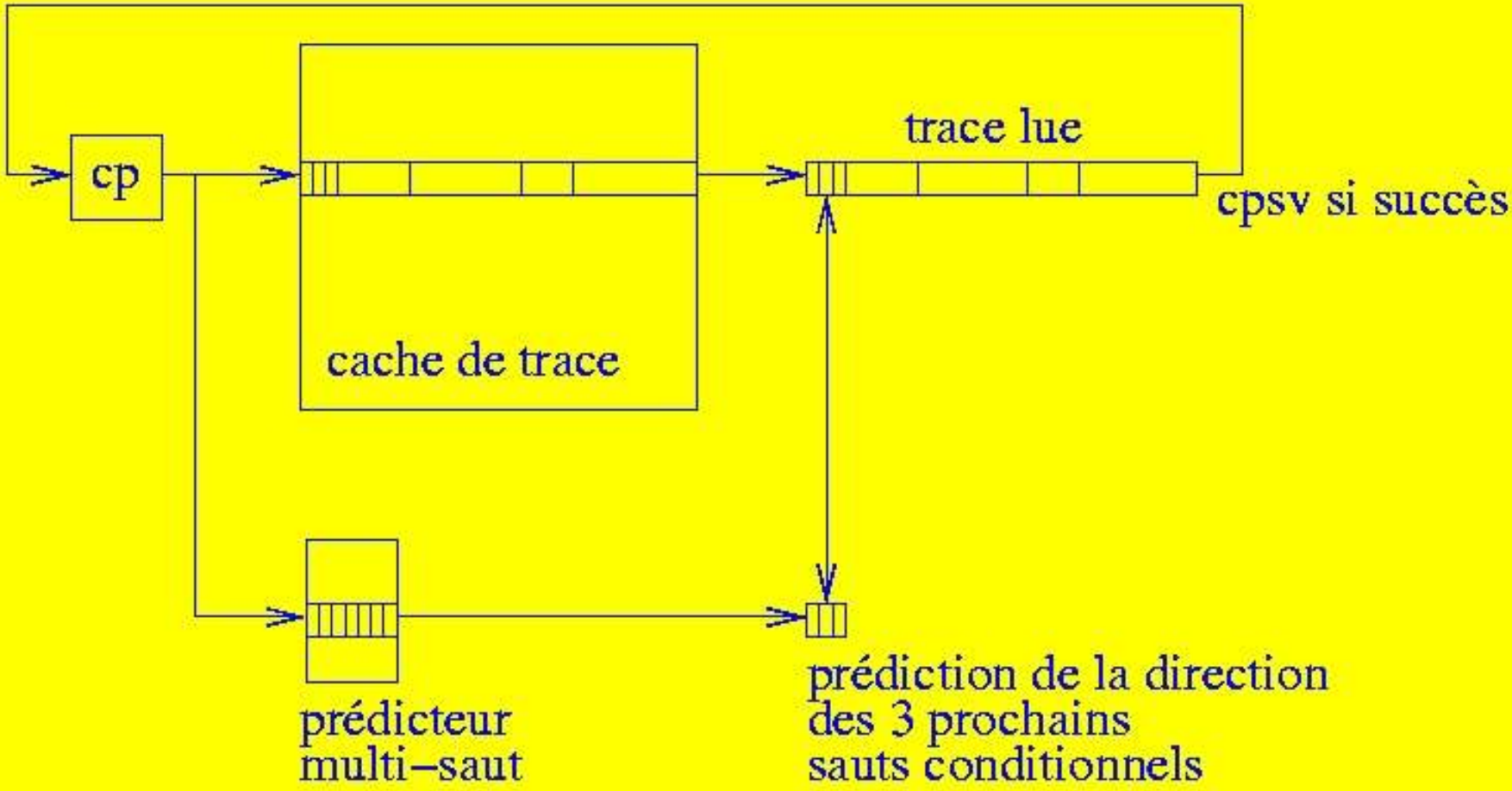
Alternatives:

Associativité des adresses
Associativité des traces (AB et AC dans le cache)
Succès partiel (le cache doit conserver les cibles des sauts intermédiaires)

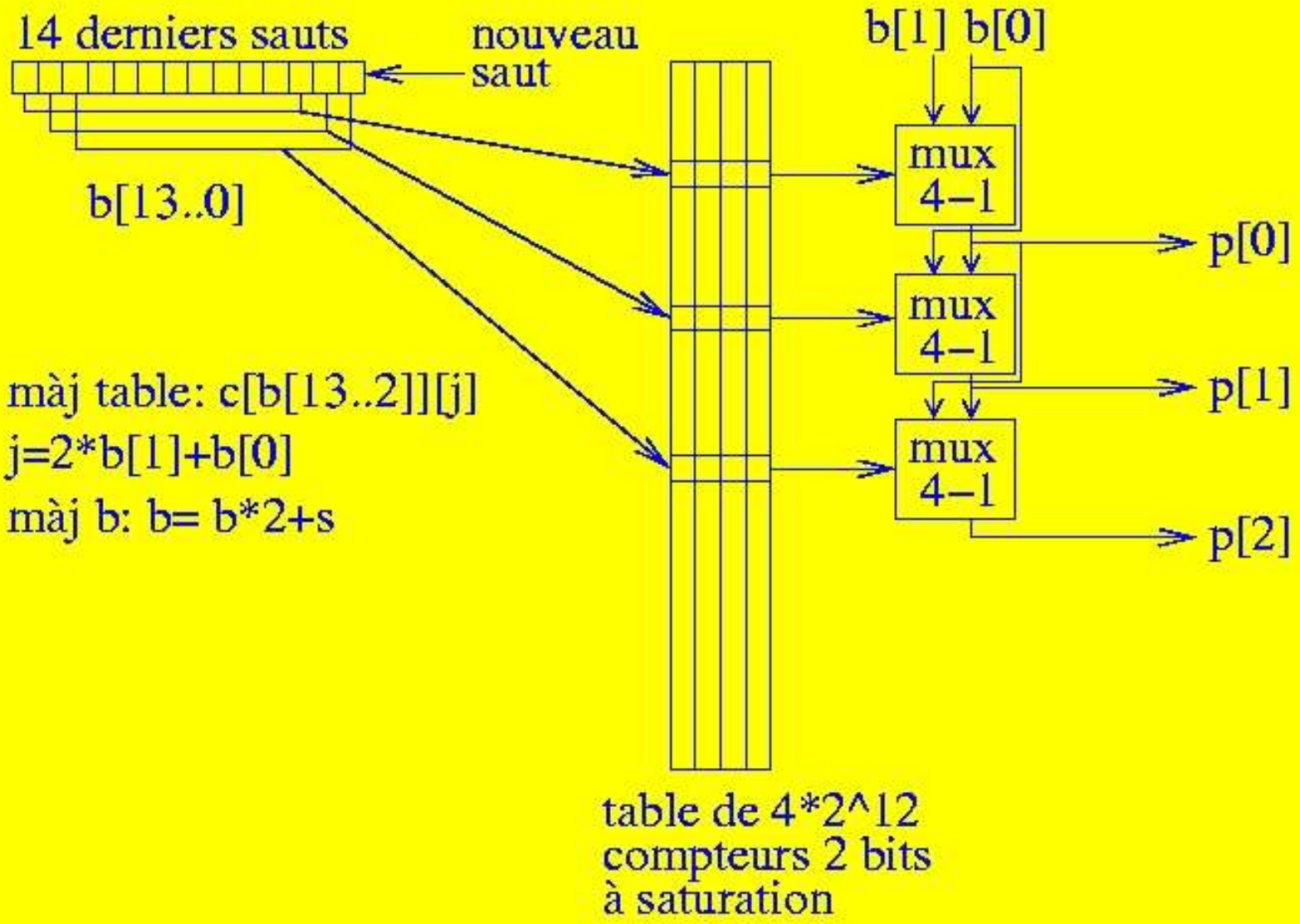
Le cache de trace est incompatible avec le BTB et la pile:

on ne dispose pas des adresses des BB internes à une trace
on ne connaît pas le nombre d'appels internes à la trace

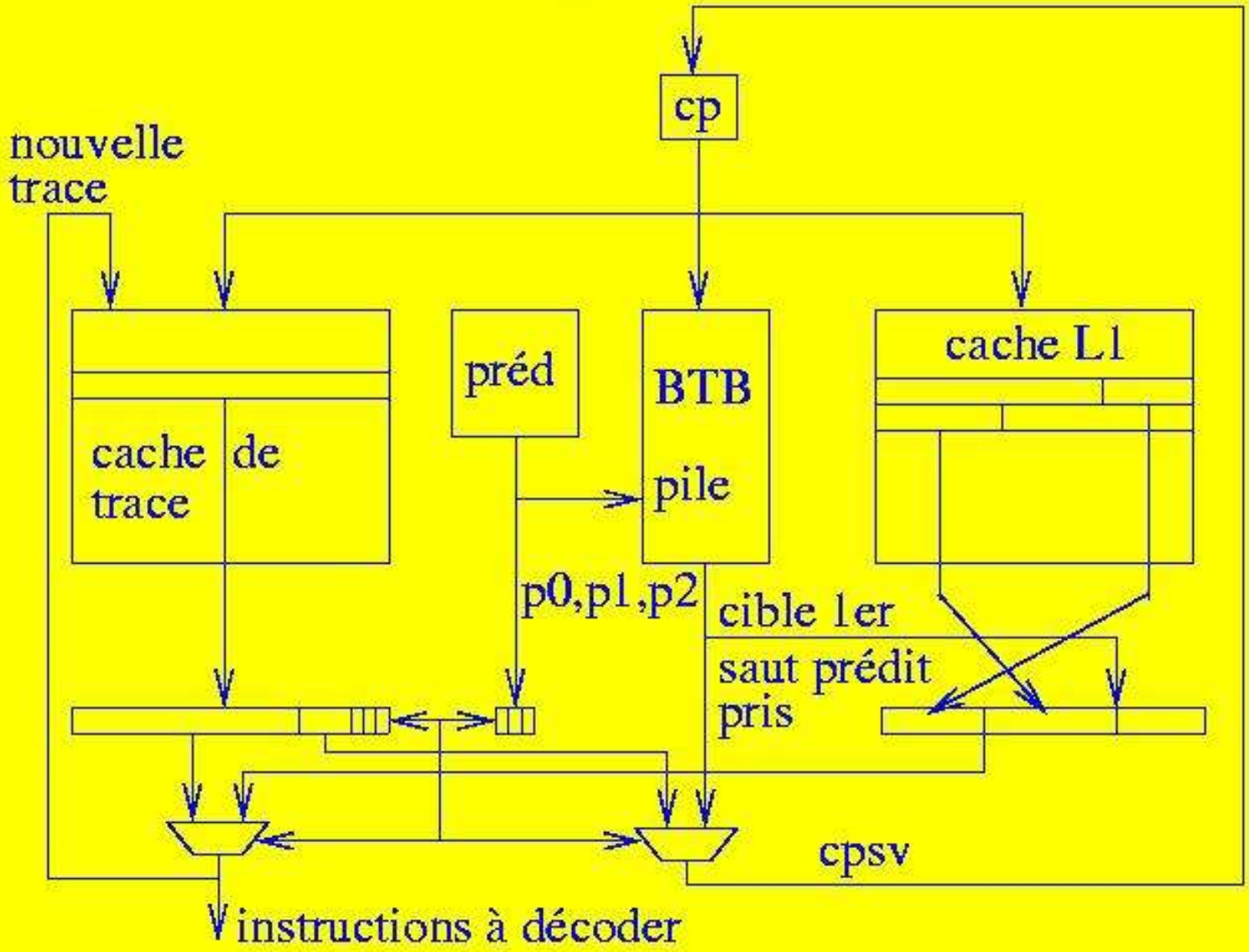
Extraire une ligne de cache de trace par cycle dans une machine spéculative



Extraire une ligne de cache de trace par cycle dans une machine spéculative



Extraire une ligne de cache de trace par cycle dans une machine spéculative



Extraire une ligne de cache de trace par cycle dans une machine spéculative

Avantage:

L'extraction de plusieurs BB en parallèle fournit beaucoup d'instructions au décodeur

Inconvénient:

Le chemin critique passe par la lecture du cache de trace ce qui limite pratiquement sa taille à 8KO (128 lignes)

Améliorations possibles:

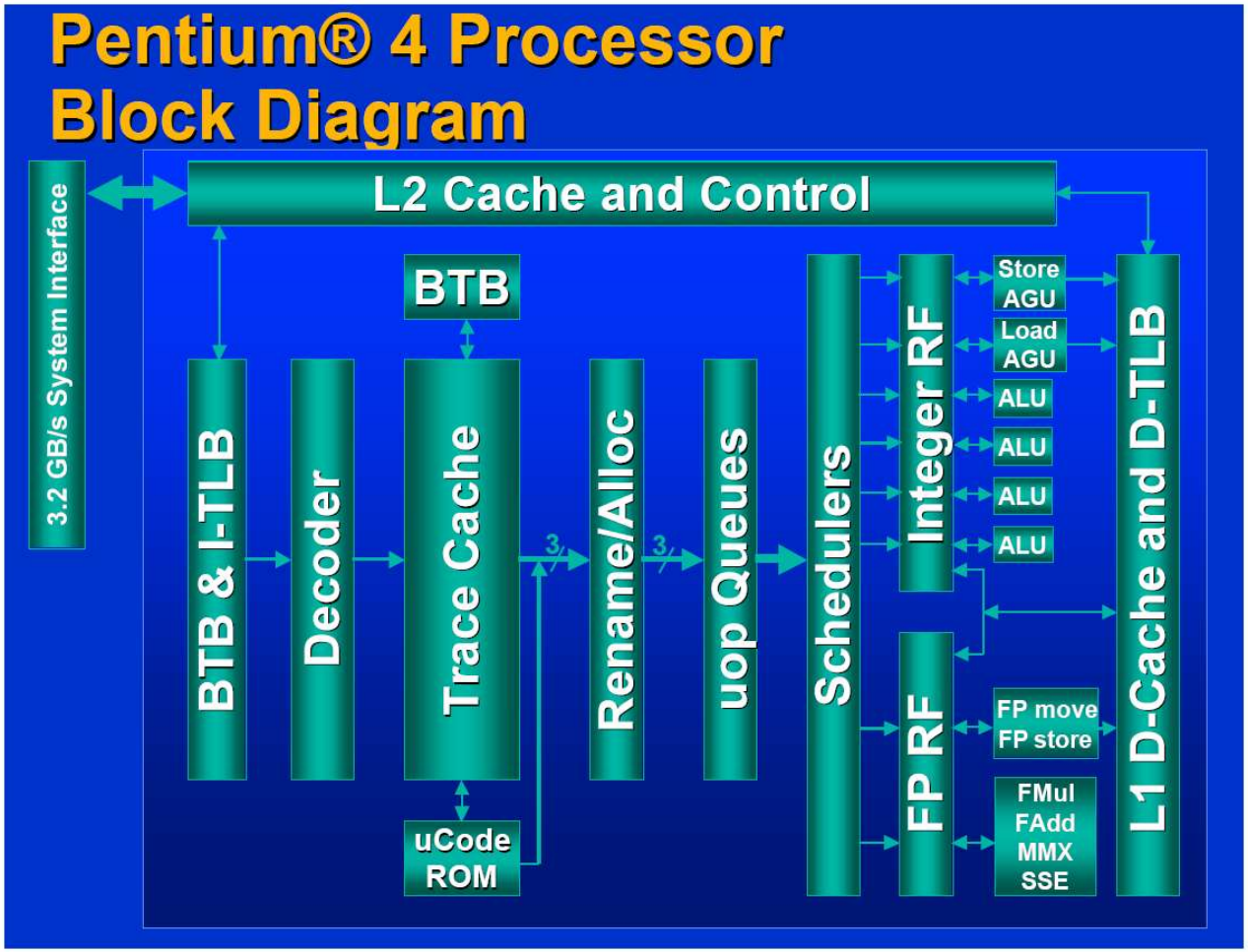
Oter le cache de trace du chemin critique

Extraire une ligne de cache de trace par cycle dans une machine spéculative

Références:

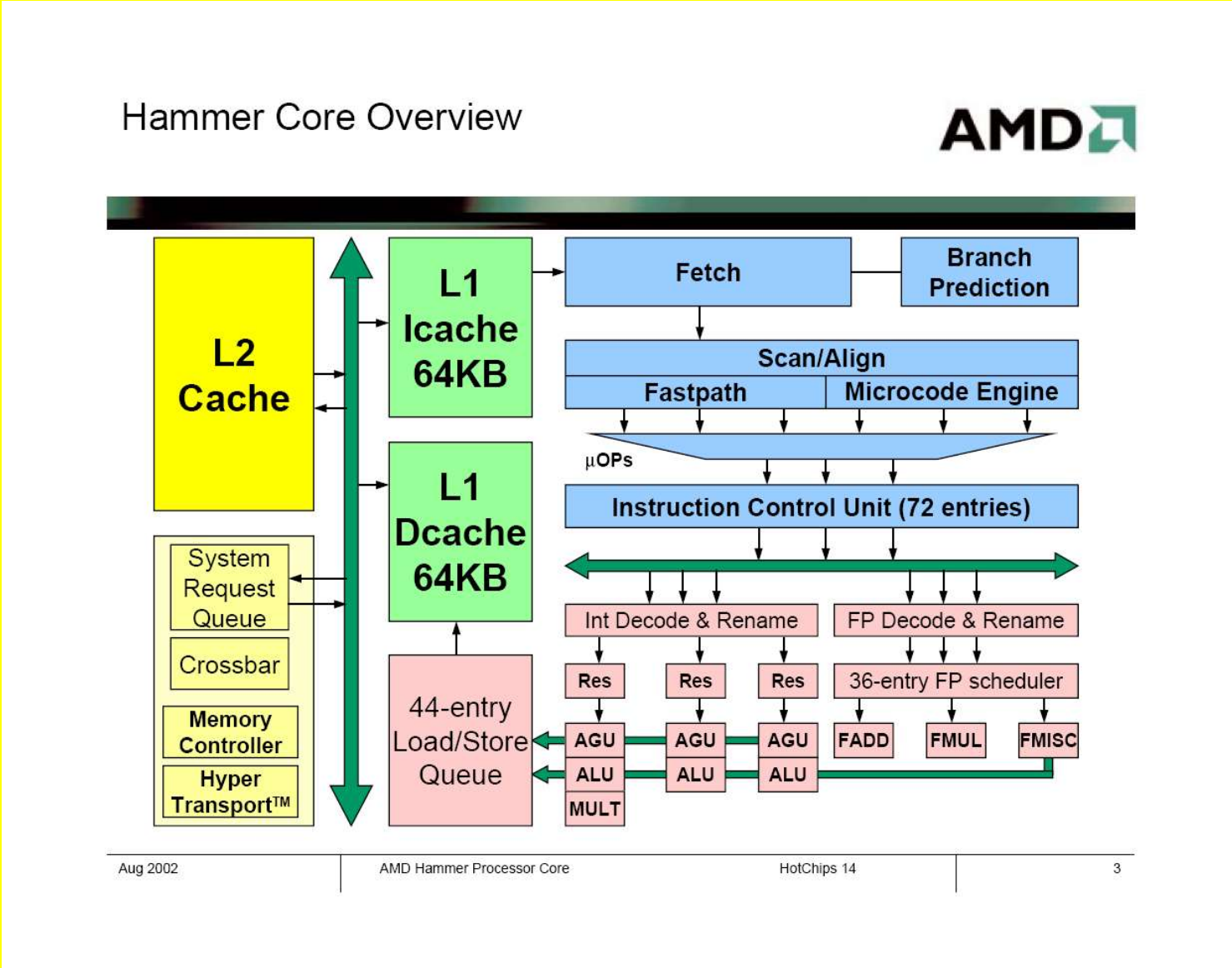
E. Rotenberg, S. Bennett, and J.E. Smith:
Trace cache: a low latency approach to high bandwidth instruction fetching.
In Proceedings of the 29th Annual International Symposium
on Microarchitecture, November 1996.

D. Holmes, S. Patel, and Y. Patt:
Alternative fetch and issue policies for the trace cache fetch mechanism.
In Proceedings of the 30th Annual International Symposium
on Microarchitecture, December 1997.



Le cache de trace du P4 contient des blocs de micro-instructions, mais pas de traces contenant des sauts prédits.

AMD Athlon 64



Conclusion

Pour extraire plus d'instructions, il faut:

- soit une mémoire à un port de lecture contenant des traces,
- soit un banc de registre contenant des blocs de base.

Dans le premier cas, on lit une trace qu'on coupe dès qu'elle s'écarte de la trace prédite ou calculée.
Dans le second cas, on assemble des blocs selon la trace prédite ou calculée.

Les prédicteurs, aussi précis soient-ils, ne fournissent pas une prédiction assez précise du chemin d'extraction: pour un pipeline de profondeur P (latence d'un saut conditionnel) et de largeur L, on emmagasine P*L instructions parmi lesquelles P*L/6 sauts conditionnels prédits (9*4/6, soit 6 sauts aujourd'hui et 15*8/6, soit 20 sauts demain). On a une fausse prédiction toutes les 200 instructions (97% de succès, 16% de sauts), soit 50 cycles pour L=4 et 25 cycles pour L=8.