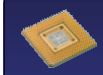
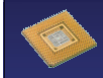


La saga HPS-Metaflow

Les véritables ancêtres du Pentium 4

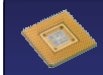


Préambule

Depuis longtemps, je me pose la question de l'origine des machines du type Pentium Pro....4, AMD K5....9 et d'autres qu'il est commun d'appeler: "superscalaires désynchronisés".

A moins d'élargir outrageusement le sens du mot "superscalaire", il me semble que ces machines n'ont que peu à voir avec la définition initiale de superscalaire qui correspondait à des multi-pipelines.

Il me semble qu'il s'agit d'une nouvelle classe de machines. Un examen plus approfondi de la gamme Pentium Pro, II, III, 4 m'a montré qu'il s'agit de machines Data-Flow.

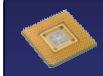


Préambule (2)

La clé de l'énigme m'a été fournie par la lecture de la préface de l'ouvrage "*The Pentium Chronicles*" de Robert P. Colwell.

Cette préface est écrite par Wen-Mei Hwu qui a joué un rôle central dans cette histoire.

The Pentium Chronicles, Robert P. Colwell, Wiley-Interscience, 2006



Préambule (3)

Il semble que cette histoire se soit jouée presque simultanément sur deux scènes qui ont fait mine de s'ignorer. L'enjeu était de trouver une façon d'accélérer les processeurs séquentiels au delà du pipe-line et du superscalaire (classique) en conservant la compatibilité binaire.

- Le projet HPS à Berkeley, puis à Illinois et finalement chez Intel a donné naissance la série des Pentium Pro, …4.

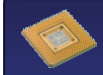
  Cette voie oubliée a montré que la solution se trouve dans des machines Data Flow

- L'évolution des RISC à Berkeley et à Stanford, qui a résolu le même problème par une voie pragmatique basée sur une amélioration des RISC superscalaires en utilisant l'algorithme de Tomasulo.

  Cette voie a engendré la société Metaflow. Elle a bénéficié d'une large publicité et a envahi le monde académique. Elle a donné naissance à toutes les autres machines de ce type: AMD K5-9, MIPS 10000, PPC 604,…..

machines HPS-Metaflow

LIP 6 labo SOC



Préambule (4)

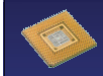
L'évolution des idées a pris un certain temps, surtout pour HPS qui était beaucoup plus ambitieux sur le plan conceptuel.

L'autre voie, que nous appellerons Metaflow a été plus rapide car elle est restée très pragmatique. Bien que partie plus tard, elle a pu offrir un modèle d'exécution exploitable avant HPS.

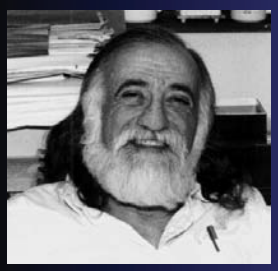
Le système Américain, privilégiant les résultats sur les concepts, semble avoir retenu Metaflow comme vainqueur de cette compétition….

machines HPS-Metaflow

LIP 6 labo SOC



Les acteurs principaux

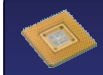


Yale Patt

Wen-Mei Hwu

Val Popescu

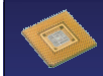
Et beaucoup d'autres: Andy Glew, M. Johnson, D. A. Patterson, …..



HPS acte 1: Le projet Aquarius

- Projet Aquarius:
  - Lancé en 1983 à l'Université de Berkeley par Yale Patt
  - Projet de "5ème génération" (d'inspiration japonaise!)
  - Objectif: réaliser une machine "intelligente" de grande puissance (Machine Prolog massivement parallèle)

The Aquarius Project, A. Despain, Y.Patt ....., COMPCON84, San Francisco, Feb 1984



HPS acte 2: Le sous-projet HPS

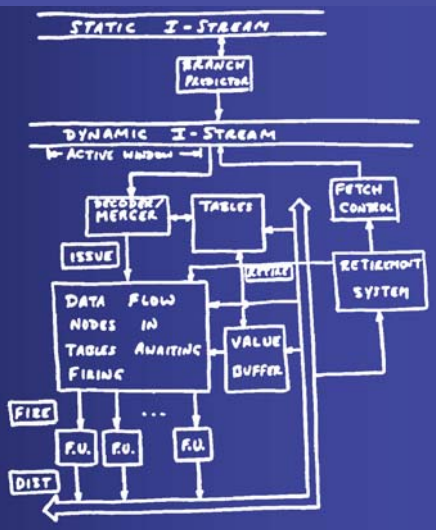
- Lancement d'un sous-projet de processeur performant (HPS pour High Performance Substrate)
- Constitue la thèse de Wen-Mei Hwu
- Subventionné par DEC
- S'oriente vers l'exécution Data-Flow de code séquentiel
- Exemple d'application: version HPS du Vax (CISC!)

HPS, a new Microarchitecture: Rationale and Introduction, Y.Patt, W. M. Hwu, Micro-18, Asilomar, Dec 1985

Run-Time Generation of HPS Microinstructions From a Vax Instruction Stream, Y. Patt,.... Micro-19, New York, Oct 1986

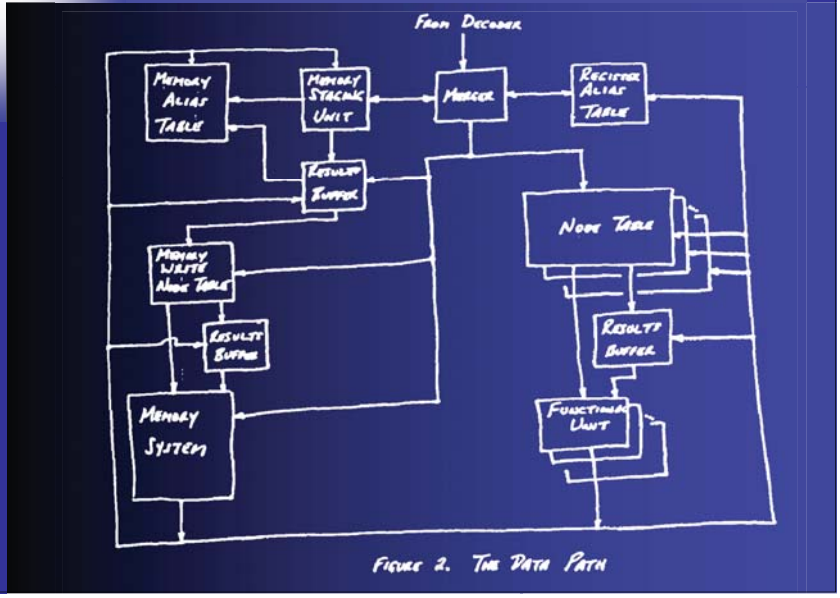
Structure d'une machine HPS

- La transformation data-flow se fait sur le flux d'instructions à exécuter (plus de branchements!)
- Une table des nœuds du graphe data-flow contient la partie du programme en exécution

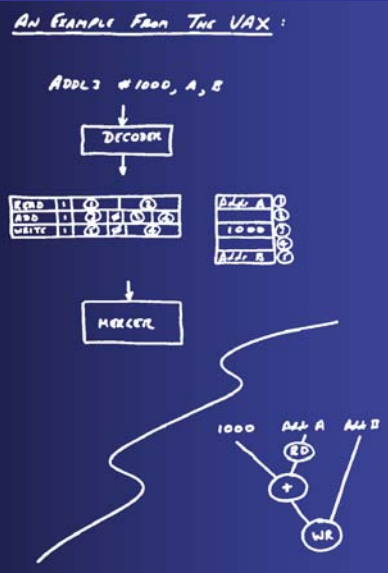


dessins de: HPS, a new Microarchitecture: Rationale and Introduction,

Structure d'une machine HPS (2)



Vax HPS



HPS acte 3: L'échec

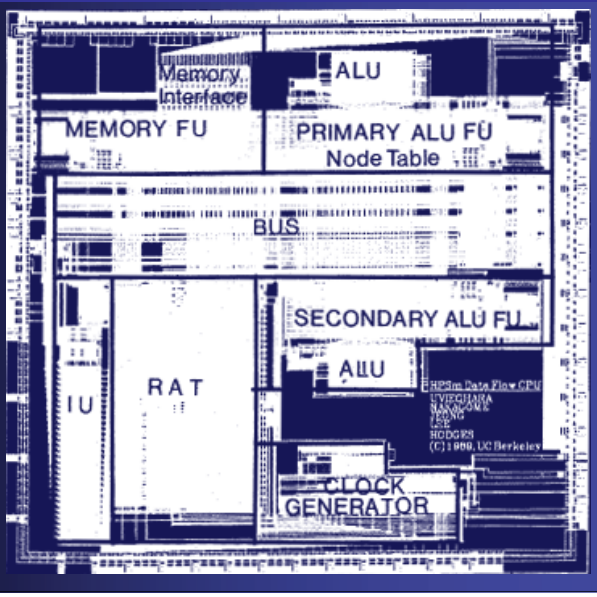
- En 1987:
  - Wen-Mei Hwu soutient sa thèse
  - DEC désapprouve le projet jugé beaucoup trop complexe, et arrête son soutient financier
  - (La communauté des architectes de Berkeley semblerait avoir désapprouvé le projet)
  - Le projet HPS est arrêté
  - Y. Patt et W.M. Hwu quittent l'Université de Berkeley
- Toutefois, un circuit prototype HPSm est réalisé à Berkeley en 1990

machines HPS- Metaflow

LIP 6 labo SOC

Circuit HPSm 1990

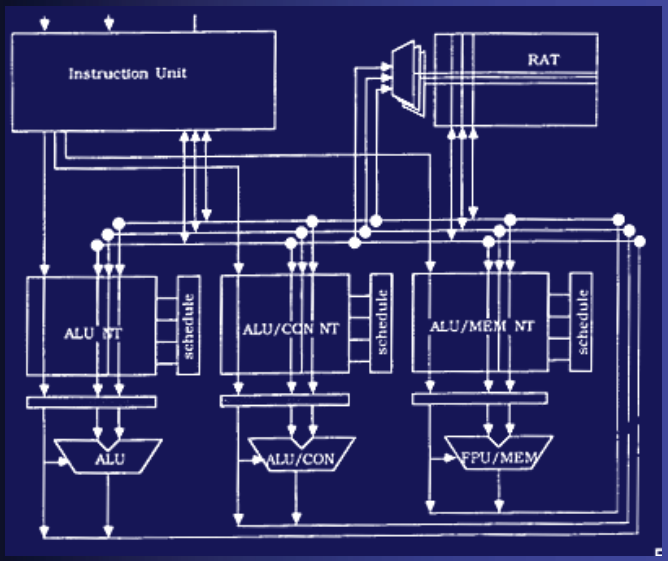
Dessins de:
An Experimental Single-Chip Data Flow CPU
G.A. Uvieghara,
W. Hwu,
Y. Nakagome,
D.K. Jeong,
D. Lee,
D.A. Hodges,
Y. Patt
1990 Symposium on VLSI Circuits



machines HPS- Metaflow

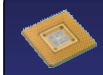
LIP 6 labo SOC

Circuit HPSm 1990



machines HPS-Metaflow

LIP 6 labo SOC

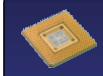


HPS acte 4: La traversée du désert

- Y. Patt devient Prof. à l'Université du Michigan
- W. M. Hwu devient Prof. à l'Université d'Illinois
- Tous deux enseignent l'architecture et montent des projets de recherche en architecture
  - HPS pour Y. Patt
  - Impact pour W. M. Hwu
- Un étudiant de W. M. Hwu, Andy Glew invente le RAT (Register Alias Table)

machines HPS-Metaflow

LIP 6 labo SOC

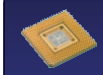


Metaflow acte 1

- Val Popescu propose, dès 1985, une approche pour l'exécution désynchronisée des RISC superscalaires.
  - Il dépose des brevets(*) sur ces idées.
  - Fonde l'entreprise Metaflow en 1988
  - Publie un article (**) dans IEEE Micro en juin 1991 (dans lequel il cite néanmoins un papier HPS)
  - Propose un nouveau vocabulaire pour ses idées, baptisées superscalaire désynchronisées, et placées dans un contexte d'évolution des RISC.

* brevets: 5 708 841(1998), 5 627 983(1997), 5 625 837(1997), 5 592 636(97), 5 561 776(1996), 5 487 156(1996)

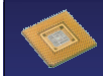
** *The Metaflow Architecture*,Val Popescu, Merle Schultz, John Spracklen, Gary Gibson, Bruce Lightner, David Isaman, IEEE Micro, june 1991



Metaflow acte 2 et fin

- J. L Hennessy et D. A. Patterson diffusent largement le point de vue de V. Popescu dans leur ouvrage *Computer Architecture, A Quantitative Approach*, considéré comme la Bible du domaine. Vocabulaire repris par toute la communauté scientifique.
- Metaflow lance des projets de SPARC désynchronisés multi-chips appelés Lightning (1989) puis Thunder (1995) (ce dernier en liaison avec Hyundaï).
- Metaflow est finalement racheté (en 1997) par STMicroelectronics pour réaliser des cœurs de PC monolithiques d'architecture standard.
- V. Popescu quitte Metaflow en 1999.

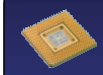
Computer Architecture, A Quantitative Approach, J. L Hennessy & D. A. Patterson, Morgan Kaufmann Publishers Inc., 1990, 1996



Deux visions de l'architecture

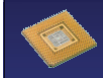
- En 1990, deux visions de l'architecture pour l'exécution parallélisée s'affrontaient:
  - La vision HPS, plus conceptuelle (et plus élégante), basée sur une exécution data-flow, mais pas encore complètement purifiée.
  - La vision Metaflow, plus pragmatique, basée sur une extension des pipe-lines et une généralisation de l'algorithme de réservation des registres de Tomasulo. Cette technique, soutenue par Patterson, était au point et brevetée. Elle était présentée comme une simple extension des superscalaires et orientée RISC.

An Efficient Algorithm for Exploiting Multiple Arithmetic Units, R.M. Tomasulo, IBM Journal of Research and Development jan. 1967, pp. 25-33



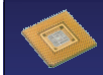
HPS acte 6: La chance

- Vers 1990 Intel songe au successeur du P5-Pentium (superscalaire à 2 voies)
  - Embauche Robert C. Colwell pour diriger ce projet (objectif: P5 x 2)
  - Une arrière pensée de VLIW
    - début des accords Intel – HP
    - R. C. Colwell vient de chez Multiflow
  - Y. Patt présente en 1990 les concepts HPS à Intel au cours d'un séminaire de deux jours
  - R. C. Colwell embauche une équipe d'architectes, dont A. Glew
  - L'équipe décide de faire un X86 HPS
- Intel termine la mise au point de l'approche HPS
- Intel achète des licences des brevets Metaflow
- Le P6-Pentium pro devient un X86 HPS



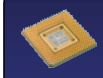
HPS acte 7: Le succès

- Le P6 (Pentium pro) est présenté début 1995 (commercialisé en nov 1995)
- Tous ses successeurs: PII, PIII, P4 sont des machines HPS
- Nexgen a été directement influencé par Y. Patt et par Metaflow. Le Nx586 (1994) puis le Nx686 sont des machines Metaflow
- AMD achète Nexgen. Le Nx686 devient le K6 (1997), et continue à évoluer jusqu'au K9
- Les autres constructeurs adoptent progressivement les principes Metaflow
- Intel reste très discret sur la structure d'exécution des Pentium pro-4
- Y. Patt et W. M. Hwu reçoivent des prix (ACM, IEEE,....) pour leurs travaux en architecture. Dont les prestigieux prix Eckert-Mauchy et Maurice Wilkes
- Y. Patt est maintenant prof à l'Université du Texas

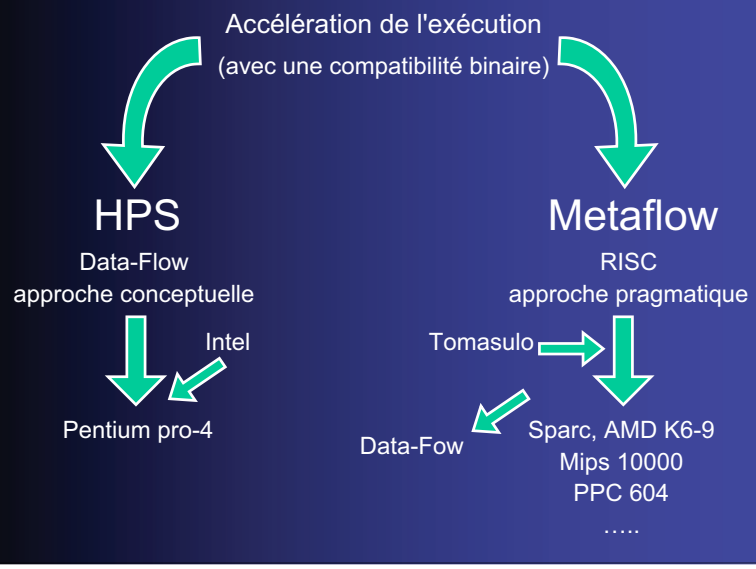


Moralité

- Les processeurs HPS et Metaflow constituent réellement une nouvelle famille de processeurs.
- L'exécution data-flow permet d'extraire automatiquement presque la totalité du parallélisme d'un programme séquentiel.
- HPS semble avoir apporté Une vision conceptuelle élégante (l'exécution Data-Flow de code séquentiel)
- Metaflow semble avoir apporté une solution technique efficace à l'amélioration des RISC superscalaires
- Par sa popularité, Metaflow a masqué l'aspect conceptuel et l'élégance de HPS. Ces machines sont actuellement toujours présentées comme des superscalaires par les tenants des RISC (du "Tomasulo amélioré")
- Il semble que les acteurs principaux de cette aventure (sauf V. Popescu) ne se sont pas immédiatement aperçus de l'importance de leur découverte...

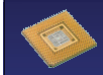


Résumé



machines HPS-Metaflow

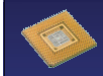
LIP 6 labo SOC



Anatomie des machines HPS-Metaflow

machines HPS-Metaflow

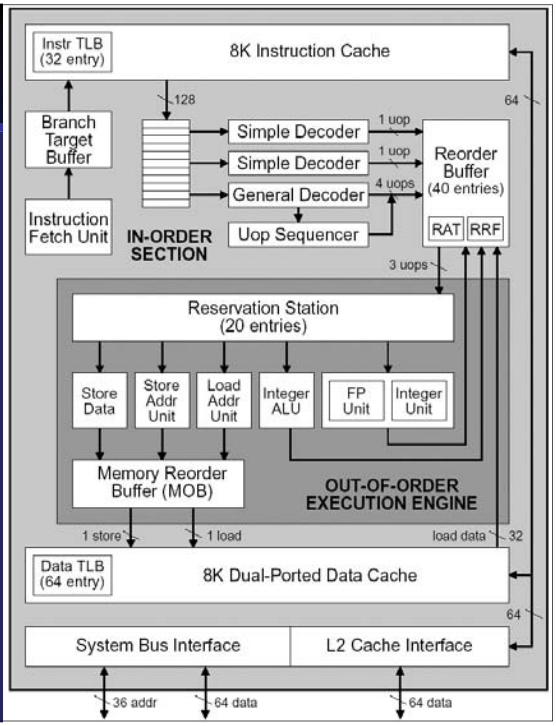
LIP 6 labo SOC



P6

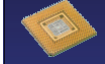
- 1,5 inst / cycle
- 1,4 fois la perf arch du P5
- 1,6 fois la complexité du P5

de: Intel P6 Uses Decoupled superscalar Design, L. Gwennap, Microprocessor Report, Vol 9 n°2 Feb 16 1995



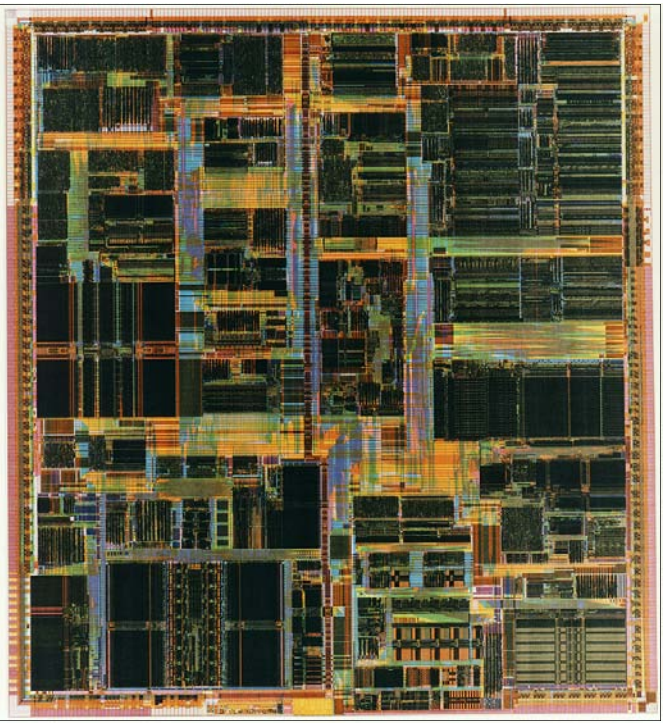
machines HPS- Metaflow

LIP 6 labo SOC



P6

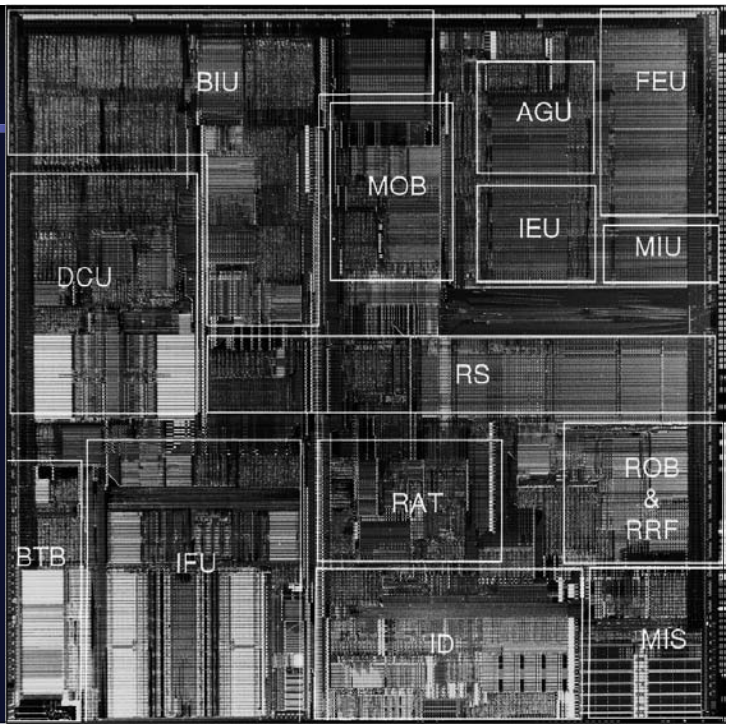
- 5,5 Mtr (cpu + L1)
- Cache inst. 8ko
- Cache don. 8ko
- Bi-CMOS 0,5µ (4 couches métal)
- 2,9 V
- 306 mm^2
- 133 Mhz
- 12W (cpu seul)



machines HPS- Metaflow

LIP 6 labo SOC

P6



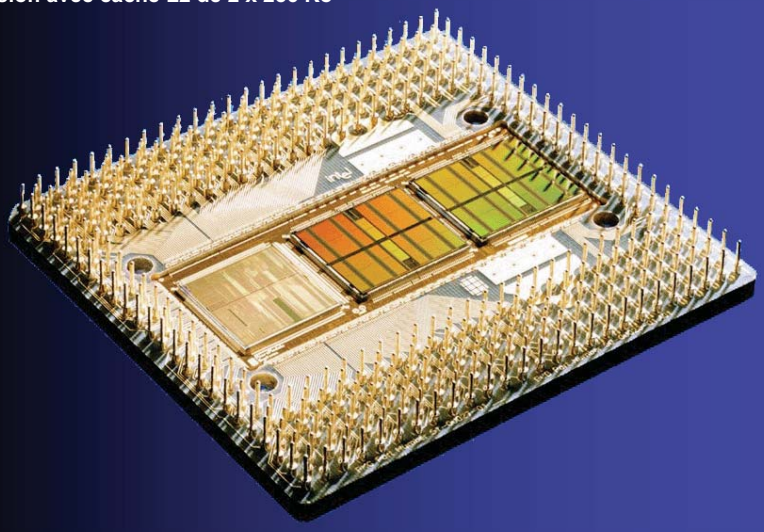
© F. Anceau

machines HPS-Metaflow

LIP 6 labo SOC

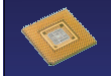
P6 en boîtier

Version avec cache L2 de 2 x 256 Ko



machines HPS-Metaflow

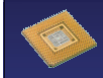
LIP 6 labo SOC

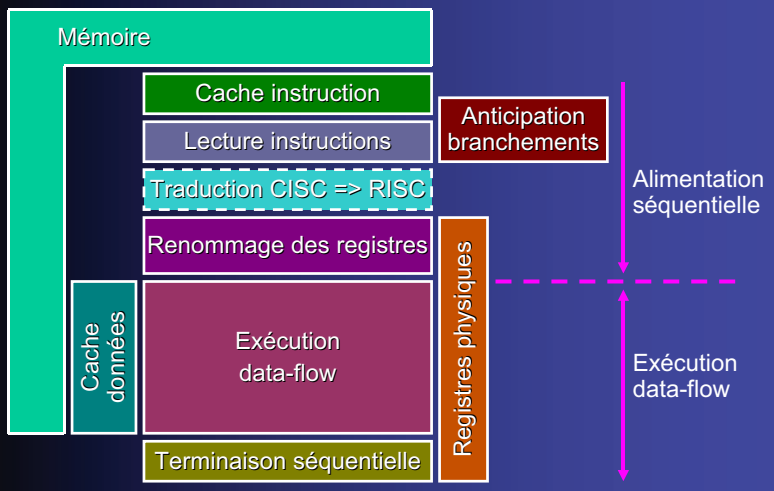


Hypothèses sur le fonctionnement des processeurs HPS

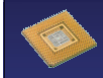
- Le fonctionnement des processeurs Intel Pentium Pro-4 comporte encore beaucoup de mystères.
- J'ai imaginé certains fonctionnements pour correspondre aux bribes d'information disponibles.
- Il peut donc y avoir quelques distances avec le fonctionnement réel de ces machines.
- Cet exposé est donc une recherche de "comment cela pourrait être"
- L'objectif n'est pas de copier ces machines, mais de pouvoir réaliser "facilement" des machines hybrides HPS-Metaflow pour d'autres architectures.

Schéma général d'une machine HPS-Metaflow





Traduction dynamique des instructions

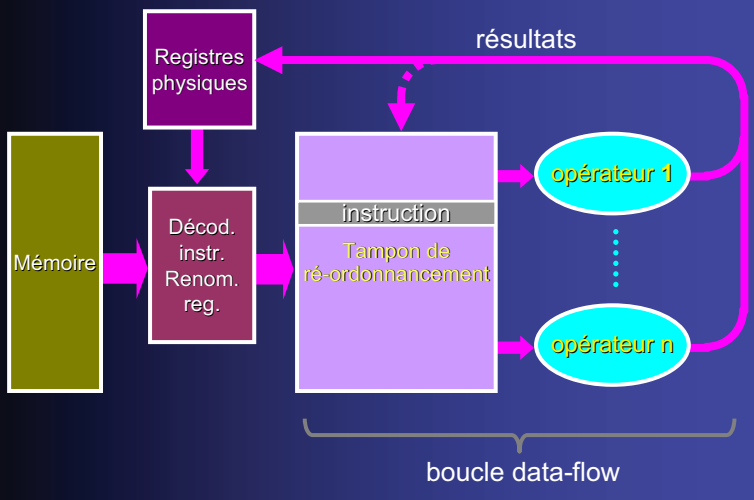


- Toute machine RISC ou CISC peut être réalisée en HPS
- Comme beaucoup de machines modernes, les machines HPS n'exécutent que des instructions RISC
  - format fixe
  - relativement simples
  - accès mémoire séparés
- Pour les machines CISC, Il faut donc traduire les instructions en RISC
- Origine de la traduction CISC ⇨ RISC:
  - Intel 486 (pipe-line CISC) (1989)
  - Nexgen Nx586 (1994)
- Par simple macro-génération (par ROM + séquenceur)
- Peut être très complexe
  - Transmeta Crusoe (traduction logicielle CISC => VLIW)

Séquentiel / data-flow HPS

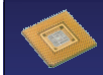
- Les machines HPS traduisent dynamiquement le flux des instructions du programme (lues séquentiellement) en un flux d'instructions data-flow
- La transformation séquentiel ➔ data-flow se fait:
  - Sur le flux d'instructions après l'exécution des branchements
  - Par le renommage des registres. Les instructions deviennent à écriture unique (langage LAU)
- Les instructions data-flow:
  - deviennent indépendantes (lorsqu'elles comportent tous leurs opérandes)
  - elles peuvent donc s'exécuter dans le désordre
- La boucle d'exécution consiste à ramener les résultats comme opérandes des instructions en attente

Boucle data-flow HPS



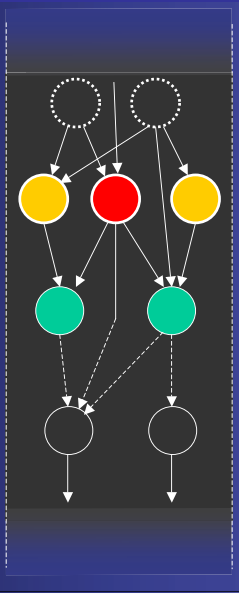
machines HPS-Metaflow

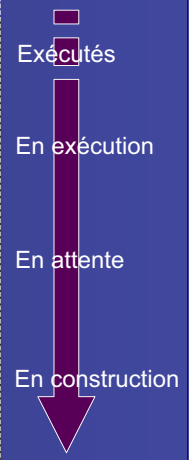
LIP 6 labo SOC



Graphe Data-Flow HPS

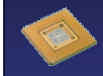
- Le graphe Data-Flow se présente comme une bande infinie avec:
  - une zone de construction de la suite du graphe
  - des nœuds en attente de leurs opérandes
  - des nœuds exécutables (et en exécution)
  - des nœuds exécutés
  - une zone effacée
- Ces zones progressent par le fetch et l'exécution
  - effacement des nœuds
  - conservation des résultats
- Le graphe ne concerne pas les ruptures de séquence





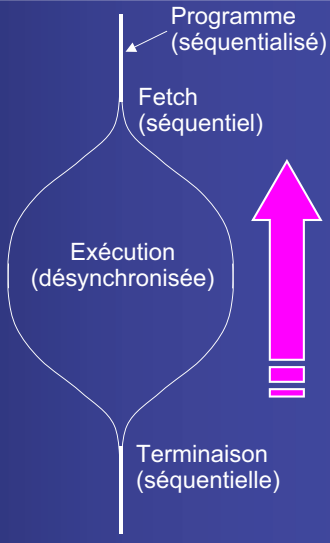
machines HPS-Metaflow

LIP 6 labo SOC



Terminaison des instructions

- L'exécution Data-Flow doit être strictement compatible avec une exécution séquentielle.
- Un mécanisme classique anticipe les branchement et produit un flux continu d'instructions séquentielles, éventuellement spéculatives,
- Le Fetch lit séquentiellement les instructions (par paquets)
- L'exécution Data-Flow exécute le flux d'instructions de manière désynchronisée
- Un mécanisme de terminaison termine les instructions séquentiellement pour restituer les ressources et traiter les IT.
- Les instructions conditionnelles ne sont pas terminables

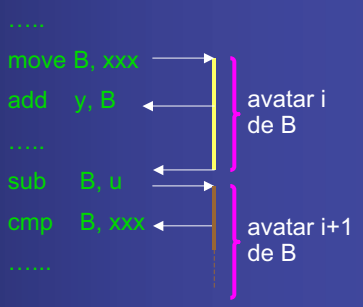


Dualité Pipe-line / HPS

- Pipe-line ⇨ Chaîne de montage
  - Les objets se déplacent devant les postes de travail fixes
  - Les informations propres aux instructions se déplacent dans des registres propres à chaque étape
- Data-Flow HPS ⇨ Hall de montage
  - Les objets sont construits à des emplacements fixes par des intervenants mobiles
  - Les instructions restent à la même place dans le ROB en attendant tous leurs opérandes

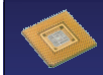
Avatars des registres

- Dans un flux séquentiel d'instructions, un registre architectural possède plusieurs "vies", que nous appellerons "avatars" qui correspondent aux valeurs successives qu'il reçoit.
- Rien n'oblige ces avatars à se dérouler dans le même registre architectural
- L'exécution data-flow consiste à isoler ces avatars



machines HPS-Metaflow

LIP 6 labo SOC

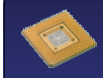


Renommage des registres

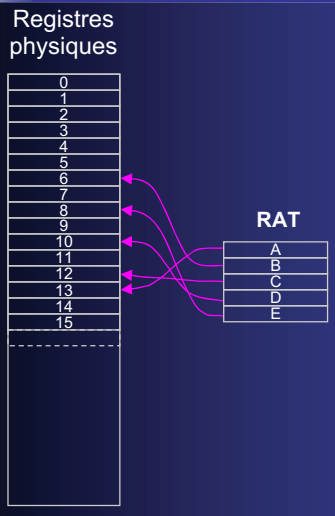
- Transformation du flux d'instructions séquentielles en instructions data-flow
- Transforme l'utilisation de registres réutilisables en registres à assignation unique
  - La machine dispose d'un banc de registres physiques beaucoup plus vaste que le nombre de registres "officiels" (40 registres pour le P6 et <??> pour le Pentium 4)
  - A chaque affectation on choisi un nouveau registre
- Le banc de registres physiques contient les derniers avatars des registres architecturaux
- Le RAT (Register Allias Table) désigne a tout instant l'emplacement des registres "architecturaux"
- Le RAT est sauvegardé au début de toute exécution conditionnelle. Le retour arrière est immédiat par simple rechargement du RAT

machines HPS-Metaflow

LIP 6 labo SOC



Renommage des registres (2)

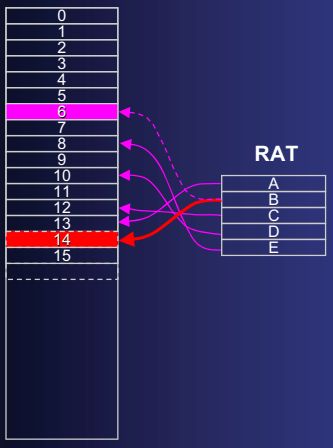


Exemple:
……...
mov B,D
add A,B
……...

machines HPS-Metaflow

LIP 6 labo SOC

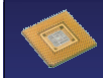
Renommage des registres (3)



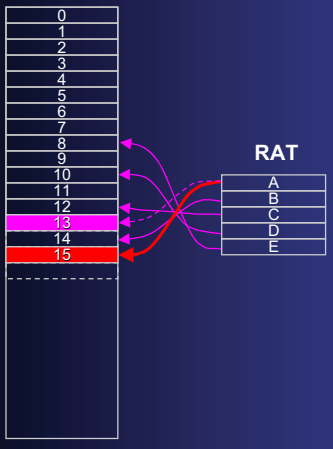
Exemple:
……...
mov B,D => [14] <=[10]
add A,B
……...

machines HPS-Metaflow

LIP 6 labo SOC



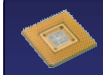
Renommage des registres (4)



Exemple:
……...
mov B,D => [14] <=[10]
add A,B => [15] <=[13] + [14]
……...

machines HPS-Metaflow

LIP 6 labo SOC

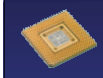


Libération des registres physiques

- La mémoire des registres physiques est gérée de manière circulaire
- La libération des registres physiques nécessite la terminaison des instructions dans l'ordre
- Si une instruction réaffecte le registre architectural R_p du registre physique r_i à r_k
  - A la terminaison de cette instruction, on est sûr que le registre physique r_i ne sera plus utilisé ➔ il sera donc libéré

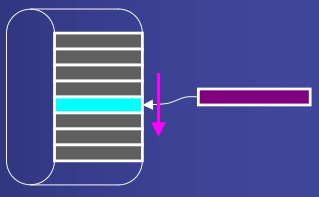
machines HPS-Metaflow

LIP 6 labo SOC



Alternative

- Une mémoire circulaire (un tambour) par registre architectural
- On avance d'une position au décodage de chaque instruction d'affectation
- Un registre Rk est renommé en rk(i) avec i la position courante du tambour k
- La terminaison d'une instruction qui affecte rk(i) libère la position i-1 précédente du tambour
- Peut être mélangé avec la solution précédente pour les registres très fréquemment utilisés (ex PC)
- Permet un nombre très important d'avatars pour ce registre particulier.



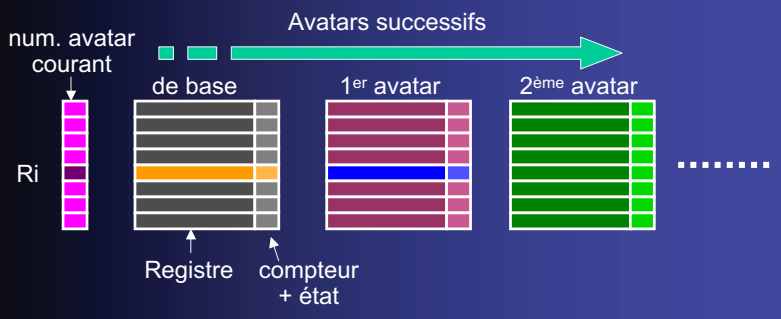
Solution Metaflow (DRIS)

- Pas de renommage explicite des registres!
- Bancs des avatars des registres logiques
  - Banc "normal" de registres.
  - Second (troisième) banc de registres pour les avatars successifs des registres logiques.
- L'utilisation de chaque registre est gérée à l'exécution par un algorithme de Tomasulo.
  - réservation au décodage
  - libération à la lecture
- Pour un nouvel avatar de ce registre (chargement), on commute vers sa version dans le second ou le troisième banc
- Peut être vu comme un renommage implicite <Ri,bj> avec bj numéro de banc
- Le nombre d'avatars d'un registre est très limité.
- Mauvaise utilisation des registres.

machines HPS-Metaflow

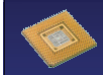
LIP 6 labo SOC

Solution Metaflow (DRIS)



machines HPS-Metaflow

LIP 6 labo SOC



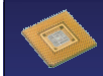
Tomasulo amélioré

- Instruction de lecture du registre Ri de l'avatar j courant:
  - au décodage:
    - Ri est transcodé en R< i, j >
    - compt< i, j > := compt< i, j > + 1
  - à la lecture effective:
    - attente état(R< i, j >)== chargé
    - lire valeur R< i, j >
    - compt < i, j > := compt< i, j > - 1
- Instruction d'écriture du registre Ri
  - au décodage:
    - si ∃ k tel que compt< i, k > = 0 ➔ Ri est transcodé en R< i, k >
      ➔ avatar_courant(Ri) := k
      ➔ état(R< i, k >) := vide
  - à l'écriture effective de R< i, k >:
    - état(R< i, k >) := chargé
- D'autres solutions sont envisageables, tenant compte de la terminaison des instructions

An Efficient Algorithm for Exploiting Multiple Arithmetic Units, R.M. Tomasulo, IBM Journal of Research and Development jan. 1967, pp. 25-33

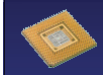
machines HPS-Metaflow

LIP 6 labo SOC



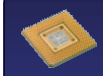
Tampon d'assemblage (ROB)

- Les instructions transcodées sont rangées (dans l'ordre) dans un tampon appelé ROB (ReOrdering Buffer) (de 40 entrées pour le P6)
  - en format long (adresses (et valeurs) des opérandes + leurs indicateurs de disponibilité) (118 bits pour le P6)
- Les valeurs des opérandes proviennent des résultats des opérateurs (reconnaissance de leurs numéro de registre physique)
- Une instruction sera finie lorsque son résultat sera chargé dans le registre physique correspondant. (reconnaissance du numéro de registre physique de son résultat). Elle pourra être terminée (dans l'ordre) (et sa place rendue libre). Le ROB est donc géré en file. Les instructions y sont rangées dans l'ordre séquentiel.
- Les instructions "prêtes" sont envoyées aux opérateurs quelque soit leur ordre
- Soit le RAT contient le PC, soit il est sauvegardé dans chaque instruction dans le ROB



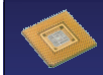
Alternatives

- Les instructions dans le ROB ne contiennent que les adresses des registres physiques (opérandes et résultat), les indicateurs de disponibilité des opérandes et ceux de terminaison de l'instruction.
  - Les opérateurs lisent les registres dans une première étape d'exécution.
- Les instructions prêtes peuvent être lues dans le ROB ou transférées (ou copiées) dans des stations de réservation (files d'attente d'alimentation des opérateurs).
  - Les opérandes sont lus au moment de ce transfert.
- Plus simple ➔ mais temps d'exécution des instructions plus long
  - Cette solution n'est pas forcément pénalisante!



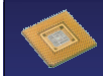
Alternatives (2)

- Certaines machines Metaflow ne possèdent pas de ROB.
  - L'assemblage des instructions avec leurs opérandes se fait directement dans les stations de réservation qui jouent le rôle d'un ROB distribué vers les opérateurs
  - Ces machines peuvent être vues comme des étapes intermédiaires entre les superscalaires de base et ceux désynchronisés



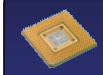
Organes de traitement

- Simples exécuteurs sans aucun mécanisme de gestion de l'exécution (5 sur le P6)
- Travaillent indépendamment
- Prennent les instructions prêtes dans le ROB (un nœud du graphe data-flow)
  - quand il ont fini les précédentes
  - pour alimenter un tampon dit "station de réservation"
- Souvent pipe-lines



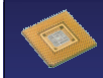
Spécifications du ROB

- Mémoire circulaire (P6: 40 mots de 118 bits)
- Mutli-associative
  - 3 résultats pour 2 opérandes et un résultat
- Calcul automatique du bit prêt
- (5 lectures simultanées pour alimenter les opérateurs)
- (3 écritures simultanées venant du fetch-renommage)



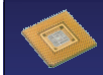
Possibilité de multitraitement

- L'indépendance des instructions exécutables permet de mélanger des flux d'instructions différents dans le ROB
  - des alternatives du programme (exécutions spéculatives)
  - des programmes différents (hyper-threading)
- La puissance de traitement se trouve alors divisée dans toutes ces exécutions



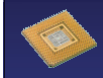
Mise à jour des opérandes

- Les sorties des opérateurs sont envoyées:
  - aux registres physiques destinataires
  - au ROB
    - pour la mise à jour des opérandes (si ils sont dans le ROB)
    - pour le positionnement des indicateurs de présence des opérandes (accès associatifs)
    - pour le positionnement d'un indicateur de fin d'exécution des instructions (par reconnaissance de l'adresse de leurs résultats)



Branchements conditionnels (HPS)

- Estimés au Fetch par un mécanisme d'anticipation de branchement (celui de Y. Patt!) (95%)
- Les instructions lues à la suite sont marquées spéculatives, mais traitées (presque) normalement:
  - Elles ne sont pas terminables
  - Elles ne peuvent pas écrire en mémoire (voir Write-Buffer)



Branchements conditionnels (2)

- L'instruction de branchement conditionnel est aussi rangée dans le ROB en attente des bits du registre d'état. Son exécution déclenche:
  - Cas de l'estimation bonne:
    - Transformation des instructions spéculatives qui la suivent en instructions normales
    - Libération des écritures mémoire (concernées) dans le Write-Buffer
  - Cas de l'estimation erronée:
    - Annulation des instructions spéculatives concernées
    - Annulation des écritures mémoire concernées
    - Rechargement du RAT à partir de sa sauvegarde
    - Lancement du Fetch à l'adresse alternative

Branchements conditionnels (3)

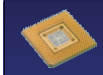
- En cas d'une succession de branchements conditionnels
  - La normalisation des instructions spéculatives doit s'arrêter au prochain branchement conditionnel
  - L'invalidation doit se poursuivre à toutes les instructions non terminées

Opérateur d'accès mémoire

- Le X86 étant un CISC, il effectue de nombreux accès mémoire.
- Les opérations d'accès mémoire doivent être exécutées dans l'ordre séquentiel pour garantir la cohérence de la mémoire.
- Pour permettre les exécutions spéculatives il faut pouvoir "défaire" les accès mémoire
  => Write-buffer d'accès mémoire (non vidé en mémoire en mode spéculatif)

machines HPS-Metaflow

LIP 6 labo SOC

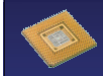


Write-Buffer d'accès mémoire

- Stocke les écritures et les lectures mémoire dans l'ordre avant leur exécution par la mémoire.
- Réservation d'une place dans la file au décodage de l'instruction. L'instruction conserve l'adresse de cette place. Les info seront rangées dans cette place au fur et à mesure de leur disponibilité. Elle sera marquée "exécutable" lorsque toutes les info seront présentes.
- Les lectures interrogent la file avant d'accéder la mémoire
  - Une lecture doit n'interroger que la partie de la file qui lui est antérieure dans l'ordre séquentiel des instructions
  - Si elle trouve l'info dans la file elle se marque comme exécutée
- Ecritures successives à la même adresse:
  - Une écriture mémoire peut invalider une écriture précédente à la même adresse, s'il y a pas d'instructions de lecture, non exécutées, à cette adresse en attente entre les deux

machines HPS-Metaflow

LIP 6 labo SOC

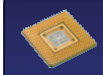


Write Buffer d'accès mémoire (2)

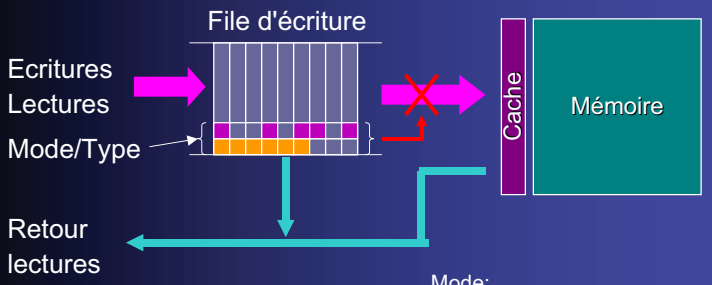
- Les écritures mémoire en mode spéculatif ne sont pas exécutables par la mémoire. Elles restent dans la file tant qu'elles ne sont pas revenues normales.
- A la résolution de la dépendance
  - si la prédiction est correcte: mode spéculatif ➔ normal
  - si la prédiction est incorrecte on efface les écritures spéculatives
- Permet d'exécuter des lectures mémoire en mode spéculatif

machines HPS-Metaflow

LIP 6 labo SOC



Wtrite-Buffer d'accès mémoire



Mode:
- normal
- spéculatif
- vide, terminée ou annulée

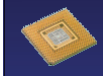
Type:
- lecture
- écriture

Type écriture:
- en attente de la donnée
- prête

Num. inst. dans le ROB
Num. proc

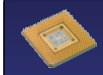
machines HPS-Metaflow

LIP 6 labo SOC



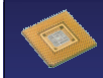
Pb des interruptions

- Il faut distinguer 3 types d'interruptions:
  - les déroutements (erreurs d'exécution)
  - les appels externes
  - les appels systèmes
- Les appels systèmes sont des appels de sous-programme. Ils peuvent être traités comme tels par le transcodage
- Les appels externes sont des appels de processus impromptus. Ils ne sont pas synchronisés avec le code. Ils peuvent être insérés comme des appels système par le fetch



Cas des déroutements

- les déroutements se décomposent en deux sous-classes:
  - ceux qui apparaissent au décodage (ex code invalide)
  - ceux qui apparaissent à l'exécution
- Les premiers peuvent être traités comme des appels système. L'instruction fautive sera alors transformée en un appel système au décodage (l'instruction fautive devra aussi être préservée).

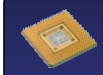


Cas des déroutements à l'exécution

- Les déroutements à l'exécution doivent être traités à la terminaison de l'instruction fautive par une technique voisine de celle des branchements conditionnels:
  - L'invalidation des instructions suivantes chargées dans le ROB et de leurs accès mémoire
  - L'insertion d'un appel impromptu de sous-programme
- Ce mécanisme nécessite de disposer de l'état des registres au moment du décodage de cette instruction:
  - soit par sauvegarde des états du RAT à chaque instruction.
  - soit en ajoutant un banc de registres architecturaux mis à jour à la terminaison des instructions.
    - Ces registres sont chargés dans les prochains registres physiques en cas de déroutement à l'exécution
    - Le RAT pointe dessus
    - L'exécution est relancée sur l'appel impromptu
    - Solution Metaflow

machines HPS-Metaflow

LIP 6 labo SOC

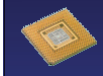


Pour finir.....

- Les approches HPS et Metaflow constituent une nouvelle famille de processeurs
  - 1,5 à 2 fois la puissance architecturale des superscalaires (classiques)
  - 1,5 à X fois leur complexité (sans les caches L2)
- Ces approches sont celles qui permettent actuellement les meilleures performances en respectant la stricte compatibilité binaire avec des processeurs séquentiels
- Elles sont applicables à n'importe quel processeur (RISC ou CISC)

machines HPS-Metaflow

LIP 6 labo SOC



Le "p'tit" dernier....

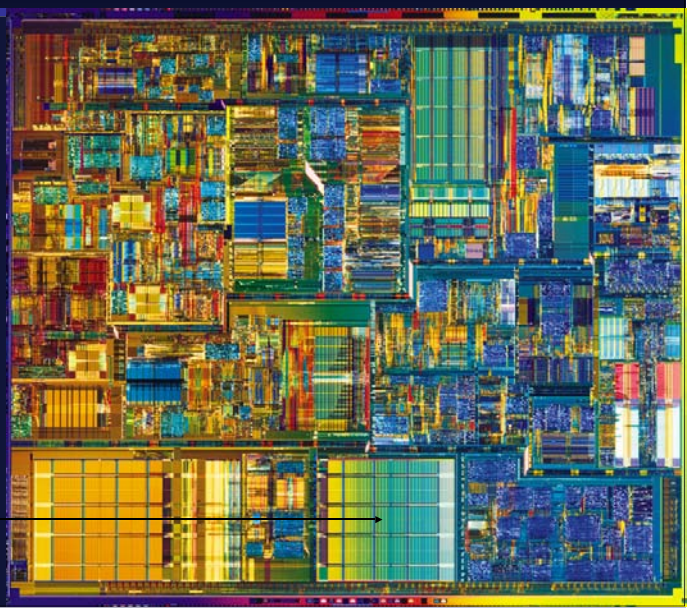
Pentium 4

42 Mtr

techno: 0,18 - 0,06µm

1,5 – 3,4 Ghz

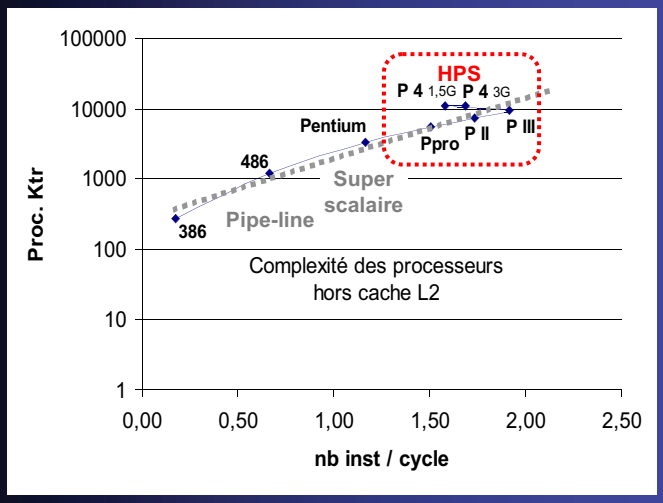
47 zones isochrones



machines HPS-Metaflow

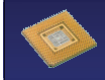
LIP 6 labo SOC

Relation Complexité / puissance architecturale



machines HPS-Metaflow

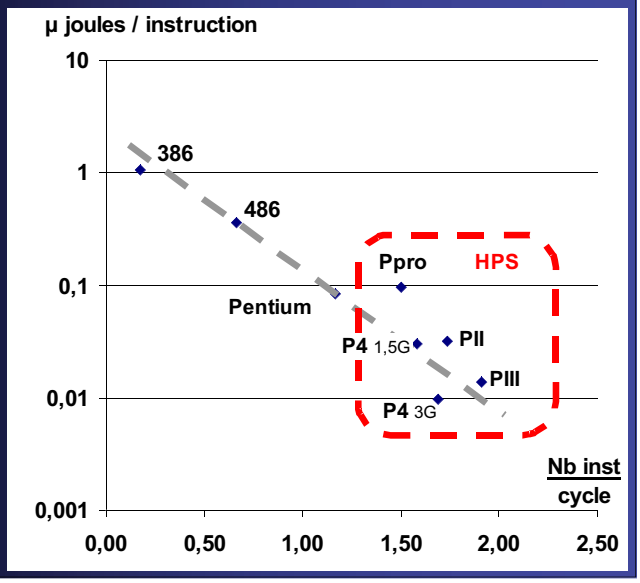
LIP 6 labo SOC



Energie par instruction versus puissance architecturale

Caches L2 inclus

Il faudrait enlever l'effet techno!



© F. Anceau