Marc Daumas
Chargé de recherches au CNRS
Laboratoire de l'Informatique du Parallélisme
http://www.ens-lyon.fr/~daumas/

Arénaire

Projet commun

ENS de Lyon

CNRS

INRIA

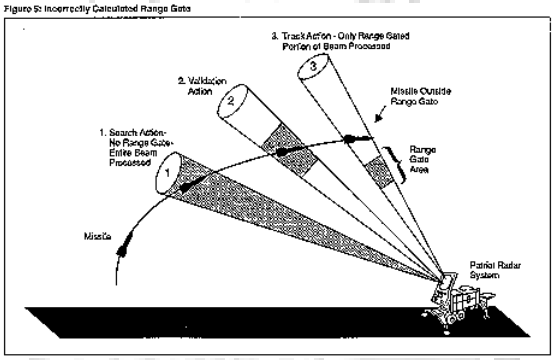
Détails des implantations de la norme de calcul sur les nombres à virgule flottante

Processeur SPARC V9 (et Compaq Alpha)
Processeurs compatibles x86 (IA 32)
Extensions multimédia MMX
Nouveautés de l'IA 64

Quelques échecs retentissants

Missile Patriot

28 morts le 25 février 1991 à Dahran, Arabie Saoudite

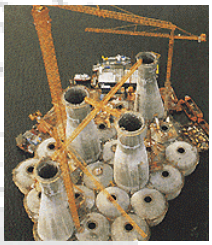


http://www.fas.org/spp/starwars/gao/im92026.htm

Quelques échecs retentissants

Plateforme pétrolière Sleipner A

M$ 700 engloutis le 23 août 1991 à Stavanger, Norvège



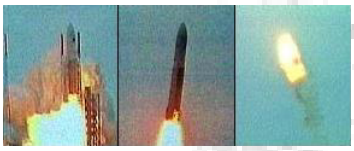
http://www.math.psu.edu/dna/disasters/sleipner.html

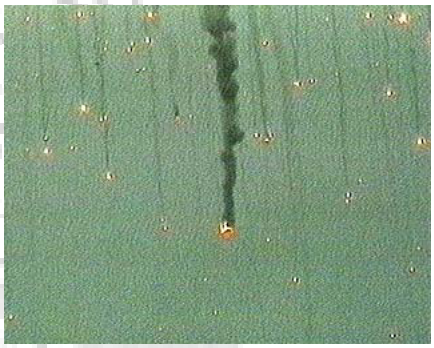
Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Quelques échecs retentissants

Explosion du vol 501 d'Ariane

Destruction automatique le 4 juin 1996 à Kourou, Guyane





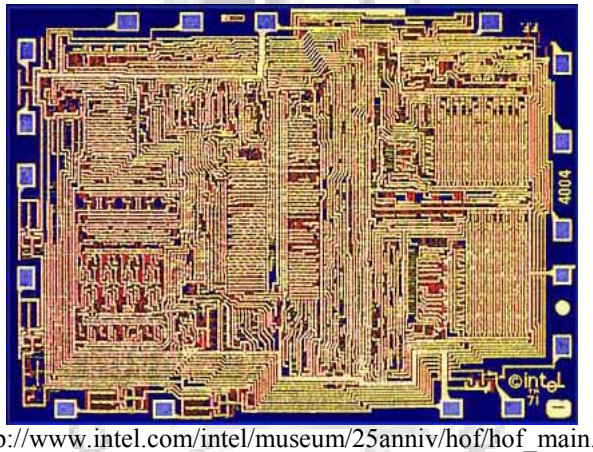
http://www.cnn.com/WORLD/9606/04/rocket.explode/

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Quelques architectures de processeurs

Intel 4004 - (1971) - Calculateur Busicom

108 kHz, 0.06 MIPS, 4 bits, 2300 transistors, 10 µm, mémoire 640 octets



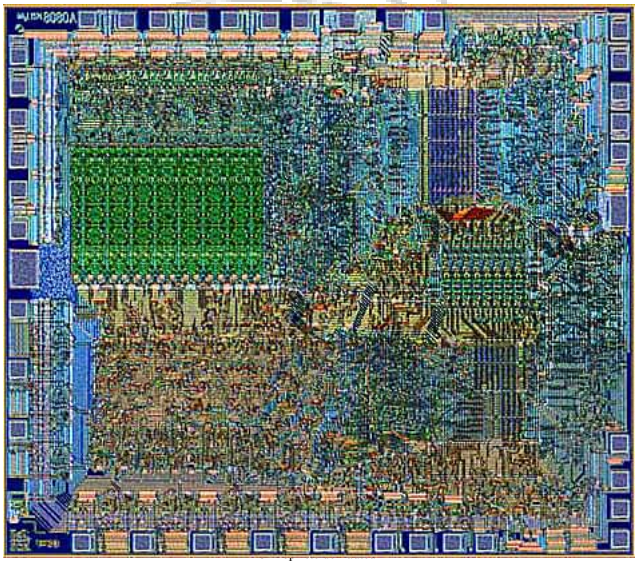
http://www.intel.com/intel/museum/25anniv/hof/hof_main.htm

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Quelques architectures de processeurs

Intel 8080 - (1974) - Ordinateur Altair

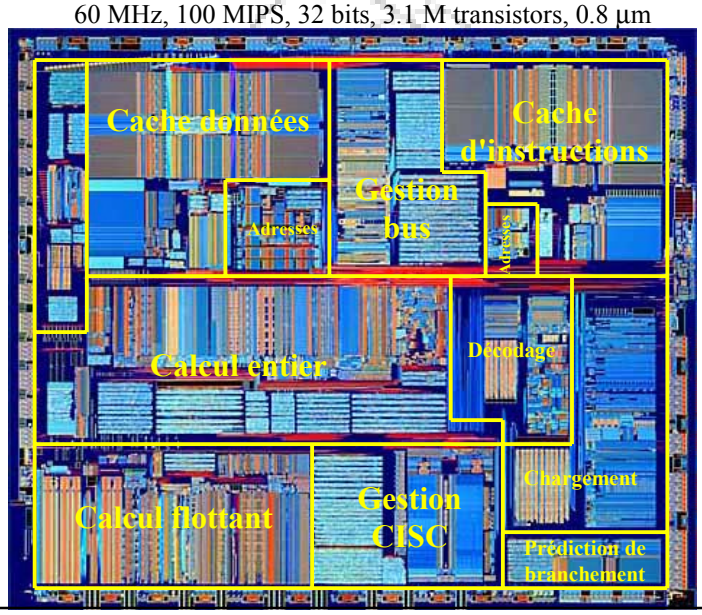
2 MHz, 0.64 MIPS, 8 bits, 6 000 transistors, 6 µm, 64 k octets



Quelques architectures de processeurs

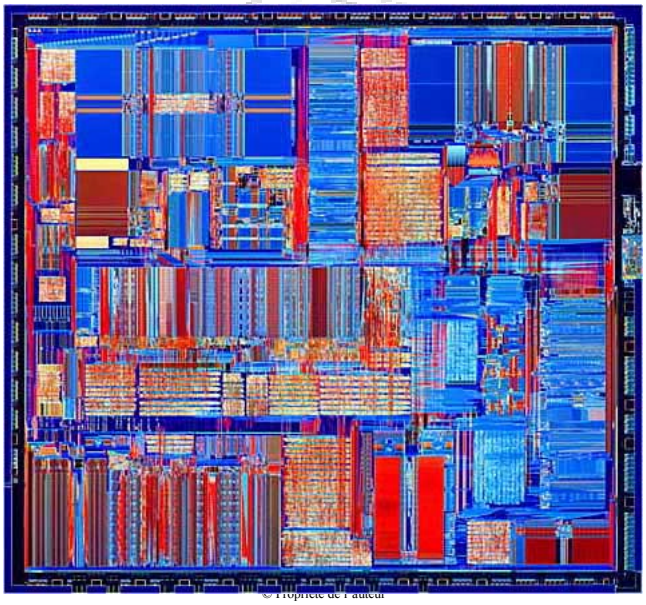
Pentium (1993)

60 MHz, 100 MIPS, 32 bits, 3.1 M transistors, 0.8 µm



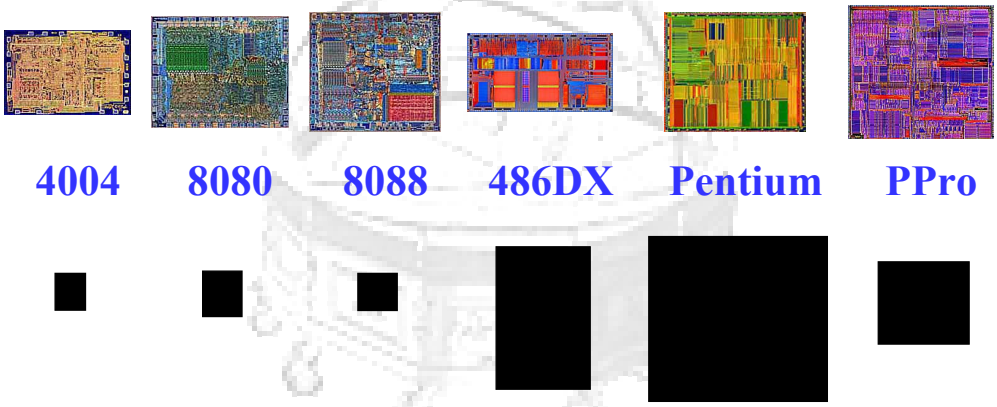
Quelques architectures de processeurs

Pentium (1993) - Niveau métal



Quelques architectures de processeurs

Évolution de la famille INTEL



Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Références bibliographiques

A proposed standard for binary floating point arithmetic

- D. Stevenson *et al.*
- **IEEE Computer**
- Vol. 14, no. 3, pp. 51-62, 1981

An american national standard: IEEE standard for binary floating point arithmetic

- D. Stevenson *et al.*
- **ACM SIGPLAN Notices**
- Vol. 22, no. 2, pp. 9-25, 1987

A proposed radix and word-length independent standard for floating point arithmetic

- W. J. Cody, R. Karpinski, *et al.*
- **IEEE Micro**
- Vol. 4, no. 4, pp. 86-100, 1984

Computer Arithmetic

- D. Goldberg
- **Computer architecture: A quantitative approach**
- pp. A1-A77. Morgan Kaufmann, 1996
- Disponible en français

What every computer scientist should know about floating point arithmetic

- D. Goldberg
- **ACM Computing Surveys**
- Vol. 23, no. 1, pp. 5-47, 1991

Numerical Computation Guide

- SUN Microsystems, 1996

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA
© Propriété de l'auteur

Quelques temps de calcul

Circuit	Cycle (ns)	Latence / pipeline (cycles) $a \pm b$	$a \cdot b$	$a \div b$	$\sqrt{a}$
DEC 21164 Alpha AXP	2.00	4/1	4/1	22-60*	†
Hal Sparc64	6.49	4/1	4/1	8-9/7-8	†
HP PA7200	7.14	2/1	2/1	15	15
HP PA 8000	5.00	3/1	3/1	31	31
IBM RS/6000 Power2	13.99	2/1	2/1	16-19/15-18*	25/24*
Intel Pentium	5.00	3/1	3/2	39	70
Intel Pentium Pro	7.52	3/1	5/2	30*	53*
Mips R8000	13.33	4/1	4/1	20/17	23/20
Mips R10000	3.64	2/1	2/1	18	32
PowerPC 604	5.56	3/1	3/1	31	†
PowerPC 620	7.50	3/1	3/1	18	22
Sun SuperSparc	16.67	3/1	3/1	9/7	12/10
Sun UltraSparc	4.00	3/1	3/1	22	22

« *Division and square root choosing the right implementation* »

Peter Soderquist et Miriam Leeser dans IEEE Micro Juillet-Août 1997

(* valeurs déduites - † fonction non disponible)

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

I - Processeur SPARC V9

Architecture de haut niveau, registres et types de données

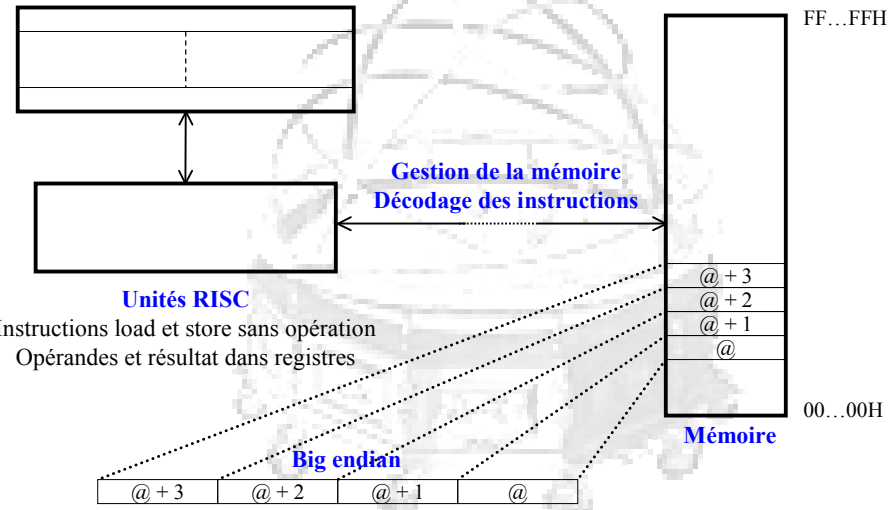
Conformité avec la norme et sources de problèmes

Deux exceptions difficiles : *Not a Number* et infiniment petit

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Architecture de haut niveau

Registres numérotés
Pas d'attribution prédéfinie

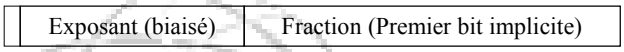


Unités RISC
Instructions load et store sans opération
Opérandes et résultat dans registres

Big endian

Registres et types de données

Trois types de données définis



– Simple précision 1 - 8 - 23
– Double précision 1 - 11 - 52
– Quadruple précision 1 - 15 - 113
Disponible également sur Power PC

Segmentation des registres

Espace unique de registres pour réaliser jusqu'à
- 32 registres en simple précision
- 32 registres en double précision
- 16 registres en quadruple précision

Conformité avec la norme IEEE

Formats simple et double précision

Compatible avec la norme IEEE

Coopération entre le matériel et le logiciel

- Implantés par le système
  - Gestion et apparition des nombres dénormalisés
  - Opérations complexes (racine carrée)
  - Opération au format quad
- Exceptions
  - `unfinished`
  - `unimplemented`

Bien positionner d'éventuelles options

Paranoia (W. Kahan)

- R. Karpinski, **PARANOIA: a floating-point benchmark**, *Byte*, 1985.

Etendre les tests

- Nombres particuliers - N. L. Schryer, **A test of computer's floating-point arithmetic unit**, Technical report 89, AT&T Bell Laboratories, 1981.
- Propriétés mathématiques - M. Parks, **Number theoretic test generation for directed rounding**, in *Proceedings of the 14th Symposium on Computer Arithmetic*, (Adelaide, Australia), pp. 241-248, 1999.

UCB Test (W. Kahan)

- http://netlib.org/fp/ucbtest.tgz

Résultats de UCB Test - Paranoia (1)

```
 ucbtest START in … at line 339 for double
 1000000 tests on 1 regions with 53 sig. bits
…
Diag. resumes after milestone Number 0 Page: 1
…
Diag. resumes after milestone Number 1 Page: 2
…
Diag. resumes after milestone Number 2 Page: 3
The program attempts to discriminate among
   FLAWs, like lack of a sticky bit,
   Serious DEFECTs, like lack of a guard dig., and
   FAILUREs, like 2+2 == 5 .
Failures may confound subsequent diagnoses.
…

Diag. resumes after milestone Number 3 Page: 4
Program is now RUNNING tests on small integers:
-1, 0, 1/2, 1, 2, 3, 4, 5, … , 32 & 240 are O.K.

Searching for Radix and Precision.
Radix = 2.000000.
Closest rel. sep. found is U1 = 1.1102230e-16.

Recalculating radix and precision.
Confirms closest relative separation U1.
Radix confirmed.
The num. of sig. dig. of the Radix is 53.000000.
```

```
Diag. resumes after milestone Number 30 Page: 5
Sub. appears to be normalized, as it should be.
Checking for guard digit in *, /, and -.
    *, /, and - appear to have guard dig.,
    as they should.

Diag. resumes after milestone Number 40 Page: 6
Checking rounding on
  multiply, divide and add/subtract.
Multiplication appears to round correctly.
Division appears to round correctly.
Addition/Subtraction appears to round correctly.
Checking for sticky bit.
Sticky bit apparently used correctly.

Does Multiplication commute?
  Testing on 20 random pairs.
  No failures found in 20 integer pairs.

Running test of square root(x).
Testing if sqrt(X * X) == X for 20 Integers X.
Test for sqrt monotonicity.
sqrt has passed a test for Monotonicity.
Testing whether sqrt is rounded or chopped.
Square root appears to be correctly rounded.
```

Résultats de UCB Test - Paranoia (2)

```
Diag. resumes after milestone Number 90 Page: 7
Testing powers Z^i for small Integers Z and i.
... no discrepancies found.

Seeking Underflow thresholds UfThold and E0.
Smallest strictly positive number found is
  E0 = 4.94066e-324 .
Since comparison denies Z = 0,
  evaluating (Z + Z) / Z should be safe.
What the machine gets for (Z + Z) / Z is
  2.00000000000000000e+00 .
This is O.K., provided
  Over/Underflow has NOT just been signaled.
Underflow is gradual; it incurs
  Absolute Error = (roundoff in UfThold) < E0.
test double rounding in gradual underflow E0
4.94066e-324 y4-E0 0 y 6.77626e-21
  1.01644e-20 1.37153e+303 4.94066e-324
The Underflow threshold is
  2.22507385850720188e-308,  below which
calculation may suffer larger
  Relative error than merely roundoff.
Since underflow occurs below the threshold
UfThold =
  (2.00000000000000000e+00)
    ^ (-1.02200000000000000e+03)
only underflow should afflict the expression
  (2.00000000000000000e+00)
    ^ (-1.02200000000000000e+03);
actually calculating yields:
0.00000000000000000e+00 .
This computed value is O.K.

Testing X^((X + 1) / (X - 1))
  vs. exp(2) = 7.38905609893065218e+00 as X -> 1.
Accuracy seems adequate.
Testing powers Z^Q at four nearly extreme values.
 ... no discrepancies found.
```

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Résultats de UCB Test - Paranoia (3)

```
Diagnosis resumes after milestone Number 160          Page: 8
Searching for Overflow threshold:
This may generate an error.
Can `Z = -Y' overflow?
Trying it on Y = -Infinity .
Seems O.K.
Overflow threshold is V  = 1.79769313486231571e+308 .
Overflow saturates at V0 = Infinity .
No Overflow should be signaled for V * 1 = 1.79769313486231571e+308
                       nor for V / 1 = 1.79769313486231571e+308 .
Any overflow signal separating this * from the one
above is a DEFECT.

Diagnosis resumes after milestone Number 190          Page: 9
What message and/or values does Division by Zero produce?
    Trying to compute 1 / 0 produces ...  Infinity .
    Trying to compute 0 / 0 produces ...  NaN .

Diagnosis resumes after milestone Number 220          Page: 10
No failures, defects nor flaws have been discovered.
Rounding appears to conform to the proposed IEEE standard P754.
 ucbtest UCBPASS in … at line 2021 for double
```

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Absence des nombres dénormalisés

Programme itération

```
double x, y, tmp;

cin >> x >> y;
if (y == 0)
  cerr << "Données erronées"
cout << x / (y * 1.5) << endl;
```

Programme attentif

```
double a, b;

if (a != b)
  x = (a + b) / (a - b);
else
  x = 1;
```

Passage à l'arithmétique non standard Sparc

- `y` non nul dénormalisée
- `y * 1.5` trop petit remplacé par 0
- Le programme divise par 0

Programme sans les nombres dénormalisés

- `a` et `b` sont différents
- `a - b` trop petit remplacé par 0
- Le programme divise par 0

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Fonctions élémentaires

Implantation logicielle

- Bibliothèque de bonne qualité

Sources de problème

- Erreur d'approximation
- Réduction d'argument
- Calcul très proche d 'un point

Propriétés supplémentaires

- Monotone
- Domaine de définition

Fonction cosinus en double précision

- 6250 valeurs dans 16 régions dans [0, 7)
- Fonction symétrique et monotone
- Erreur en ulp sur 53 bits

Solaris
[-0.805, 0.819]

Exemple

- Y = arctan (X)
- X devient très grand
- Y tend vers l'arrondi au plus proche de Π / 2

Y est plus grand que Π / 2
tan (Y) < 0

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Résultats de UCB Test – BeEF (1)

```
ucbtest START … at line 448 for double
6250 tests on 16 regions with 53 significant bits

NME  =  Negative Maximum Error observed in ULPs
PME  =  Positive Maximum Error observed in ULPs
NMC  =  Non-monotonicity count
{SYM}=  Non-symmetry count if nonzero

               [      From     ,       to     )  N.M.E.  P.M.E. NMC {SYM}

SIN(X)      ALL [      0.0000000,      7.0000000)  -0.826    0.851 0
ucbtest UCBPASS in SIN(X) at line 442 for generic
```

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Résultats de UCB Test – BeEF (2)

```
ucbtest START in … at line 73 for double
 1000000 tests on 1 regions with 53 significant bits
 PIRATS: computes  PI = A/B+C and R = PI-P.
 J  1 A                22 / B                7 + C  -1.26448926734961872e-03
 J  1 R   0.00000000000000000e+00 +-   1.965411139e-18
 J  2 A               333 / B              106 + C   8.32196275290875219e-05
 J  2 R   0.00000000000000000e+00 +-   2.032631625e-19
 J  3 A               355 / B              113 + C  -2.66764189062422295e-07
 J  3 R   0.00000000000000000e+00 +-   8.885032345e-22
 J  4 A            103993 / B            33102 + C   5.77890634390381943e-10
 J  4 R   1.03397576569128459e-25 +-   2.438032454e-24
 J  5 A            104348 / B            33215 + C  -3.31627806246072593e-10
 J  5 R  -5.16987882845642297e-26 +-   1.693631800e-24
 J  6 A            208341 / B            66317 + C   1.22356532942188593e-10
 J  6 R   0.00000000000000000e+00 +-   7.335524165e-25
 J  7 A            312689 / B            99532 + C  -2.91433849348569077e-11
 J  7 R   1.29246970711410574e-26 +-   2.006050732e-25
 J  8 A            833719 / B           265381 + C   8.71546725822407103e-12
 J  8 R  -6.46234853557052871e-27 +-   6.773278694e-26
 J 25 A    139755218526789 / B    44485467702853 + C  -1.61109530158630018e-28
 J 25 R    4.484155085839414 63e-44 +-   3.684670703e-42
 J 26 A    428224593349304 / B   136308121570117 + C   3.80544972802866803e-30
 J 26 R    1.40129846432481707e-45 +-   9.041281521e-44
 J 27 A  5706674932067741 / B  1816491048114374 + C  -2.33281989961436183e-31
 J 27 R    4.37905770101505335e-47 +-   5.749689810e-45
 J 28 A  6134899525417045 / B  1952799169684491 + C   4.86271497755635769e-32
 J 28 R  -1.64214663788064500e-47 +-   1.241700570e-45
 ucbtest UCBPASS in ... at line 129 for double
```

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Not a Number

NaN actif (Signaling NaN)

– Réagir à un problème
– Adapter la réaction à la situation

Jamais généré automatiquement

Gestionnaire

– Générer un SNaN en cas de problème

Adapter le SNaN à l'opération qui l'utilise

NaN passif (Quiet NaN)

– Propager un résultat inconnu
– Invalider une partie du calcul

Réponse automatique à une opération invalide

Pas de gestionnaire

– Pollution de l'état
– Remise à zéro

Gestionnaire

– Définir un résultat par défaut

Adapter le résultat à l'opération qui le crée

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Exception d'infiniment petit

Déclenchement de l'interruption
Résultat de l'opération

Plus petit que le plus petit nombre normalisé

- Résultat exact
- Résultat arrondi

On se rapproche de l'infiniment petit
Il faut s'en éloigner

Les choix doivent être consistants

– Pour toutes les opérations
– Pas pour tous les types

Mise à jour de l'historique
Résultat de l'opération

De plus moins précis que

- Résultat exact
- Résultat arrondi sans limitation de l'exposant

Un infiniment petit a fait perdre de la précision
Faire attention à la qualité du résultat

Non fixé par la norme SPARC

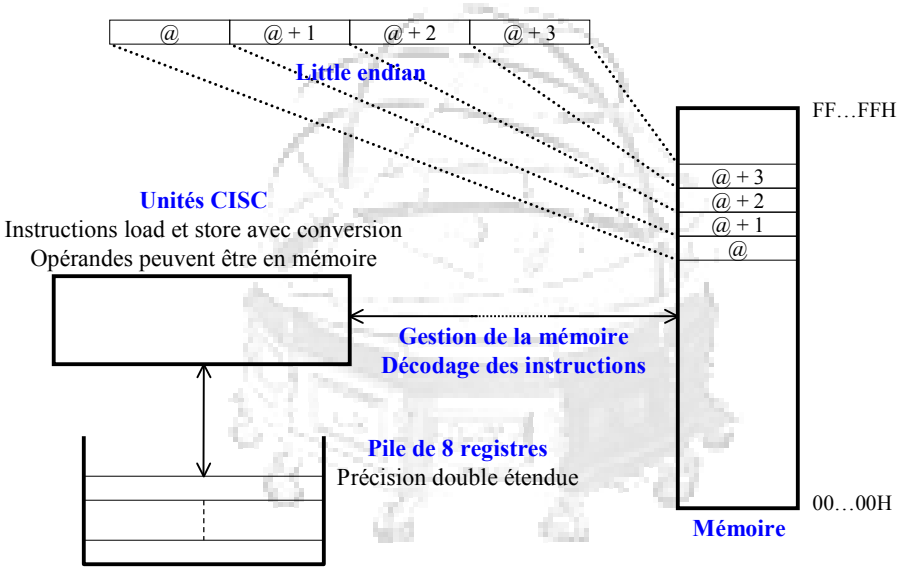
Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

II - Processeurs compatibles x86

Architecture de haut niveau, registres et types de données
Conformité avec la norme et sources de problèmes
Des petites nouveautés qui peuvent être importantes

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

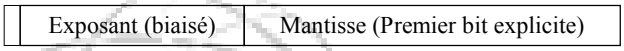
Architecture de haut niveau



Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

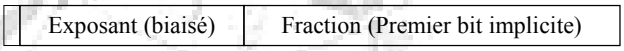
Registres et types de données

Un seul type de donnée interne



– Double étendu 1 - 15 - 63 + 1
Promu à 96 bits avec Unix ABI

Deux types supplémentaires de donnée externe



– Simple précision 1 - 8 - 23
– Double précision 1 - 11 - 52

Gestion de la pile

– Sommet de la pile (Top of the Stack - TOP)
  • Opérandes
  • Résultat
– Dépassement de capacité

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Quelques temps de calcul (Latence / Pipeline)

Addition

– Deux nombres flottants 3 / 1
– Avec un entier 7 / 4

– Exécutable en même temps qu'un échange
– Indépendant de la précision de travail

Multiplication

– Deux nombres flottants 3 / 1
– Suivi par une autre multiplication
  Latence pipeline 3 / 2

Arrondi

– Vers un entier 9-20

Opérations micro-codées non triviales

– Division	39
– Racine carrée	70
– Fonctions trigonométriques	59 - 174
– Fonctions hyperboliques	54 - 114

Mot de contrôle de l 'unité

– Lecture	7
– Ecriture	2

Accès à la mémoire

– Lecture en simple ou double	1
– Ecriture en simple ou double	2
– Lecture-écriture en double étendu	3

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Conformité avec la norme IEEE

Format double étendu

Compatible avec la norme IEEE

Matériel uniquement

Fonctions élémentaires implantées

- Bonne qualité
  - AMD, Cyrix
  - Intel depuis le Pentium
- Quelques problèmes persistent

Cosinus : erreur maximum constaté (en ulps)

- Linux [-0.531, 0.536]
- Solaris [-0.998, 1.000]

Formats simple et double précision

Arrondi de la mantisse
(Norme 754 § 4.3)

- Conversion pendant l'écriture en mémoire
- Double arrondi et double exception
  - Après l'opération, double étendu
  - Lors de l'écriture, simple ou double

Détection de l'infiniment petit

- Arrondi pour les formats simple et double
- Exact pour le format double étendu

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Double arrondi

Programme retrouve

```
double ref, tmp;

ref = 169.0 / 170.0;
for (int i = 0; i < 250; i++) {
  tmp = i;
  if (ref == tmp / (tmp + 1))
    break;
}
cout << i << endl;
```

Changement de la précision de l'arrondi

```
ieee_flags ("set", "precision",
            "double", &out);
```

Unité Sparc

- Résultat attendu 169
- Le résultat d'un opération est spécifié entièrement et ne peut pas varier

Unité x86

- Résultat non trouvé
- `ref` stocké en mémoire en précision double
- `tmp / (tmp + 1)` évalué en précision double étendue
- Jamais d'égalité

Arrondi de la mantisse au format double

- Résultat attendu 169 sur unité x86

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Double détection des exceptions

Programme précision

```
double fp = 1024.0;

for (int i = 0;
     fp / 1024.0 != 0.0;
     i++)
  fp /= 2.0;
cout << i << endl;
```

Unité Sparc

– Résultat attendu 1075

Unité x86

– Résultat retardé 1085

– `fp / 1024.0`

- Mantisse arrondie au format double 53 bits
- Exposant calculé au format double étendu 15 bits
- Jamais d'infiniment petit

– Arrêt de la boucle

- `fp /= 2.0` retourne 0.0
- Stocké en précision double
- Exposant sur 11 bits

Mauvais déclenchement des exceptions

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

III Plongeons plus profondément dans les différences

Registres d 'état
Gestion des exceptions
Historique du programme

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Registres d'état

Valeurs numériques et places différentes

– Arrondi actif (`Rounding direction`)
  Plus proche, zéro, excès, défaut

– Comparaison (`Condition codes`)
  Egalité, inférieur, supérieur, sans ordre

– Exceptions masquées / activées
  - `Invalid operation`
  - `Division by zero`
  - `Numeric overflow`
  - `Numeric underflow`
  - `Inexact result`

Unité Sparc

FSR (Floating Point State Register 64b)

– `unfinished_FPop`
– `unimplemented_FPop`

Unité x86

SW (Status Word 16b)

CW (Control Word 16b)

– `Invalid operation` regroupe
  - `Stack overflow or underflow`
  - `Invalid arithmetic operand`
– Exception `denormal operand`
– Conditions étendues
– Précision de l'arrondi (`Precision control`)

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Gestion des exceptions

Système (SIGFPE)

– Attribution d'une zone mémoire
– Sauvegarde de l'état de l'unité flottante
– Appel du gestionnaire d'interruption
– Exception la plus prioritaire

Unité Sparc

– Détection à la fin de l'instruction

Unité x86

– Déclenchement retardé des interruptions
– Début de l'instruction flottante suivante
– Mise à zéro du registre historique

Processus utilisateur

– Retrouver l'adresse de l'opération fautive
– Décoder l'opération
  - Registres ou adresses des opérandes
  - Registres ou adresse du résultat
– Mise à jour des registre et de la mémoire
– Retour au fonctionnement normal

Apparition d'une exception non activée

– Unité Sparc
  - Problème quand seul le inexact est actif
– Unité x86
  - Transfert vers un gestionnaire par défaut
  - Arrêt du programme

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Pile

Fonctionnement

- Registres alloués depuis le registre 7
- Pointeur vers le somme de la pile TOP
- Insertion dans la pile
  - Décrément de TOP
  - Affectation de la nouvelle valeur
- Retrait de la pile
  - Lecture de la valeur mémorisée
  - Incrément de TOP
- Dépassement de capacité
  - Insertion alors que le TOP = 0
  - Nouvelle valeur de TOP = 7
- Pile vide
  - Retrait alors que TOP = 7
  - Registre 0 marqué vide

Exception

- `Stack overflow or underflow`
- Active `Invalid operation`
- Condition `C1`
- Extension de la pile sur un registre non vide
- Lecture depuis un registre vide

Registre `Tag word`

- Mis à jour avec les opérations sur la pile
- Quatre états
  - Entrée valide
  - Entrée nulle
  - Valeur spéciale
  - Registre vide

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Historique du programme

Unité Sparc

- Deux champs séparés
  - `Current exception (cexc)`
  - `Accrued exception (aexc)`
- Historique modifié sur l'instruction
  - Aucune interruption déclenchée
  - Ajout (`ou` logique) du statut

Unité x86

- Un champ unique
- Ajout (`ou` logique) du statut
- Détection d'une exception non masquée
- Transfert à un code de contrôle unique

Activation d'une exception après son apparition

- Aucune autre interruption déclenchée
- Exception toujours active dans l'historique
- Déclenchement de l'interruption
- Adresse de l'instruction fautive erronée

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

IV - Extensions Multimédia et IA 64

Parallélisme de données (SIMD)
Ce qui va changer avec l'IA 64

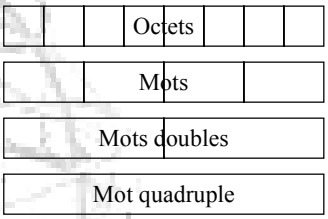
Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Parallélisme de données (SIMD)

MMX – MultiMedia eXtensions

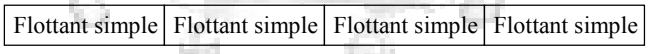
- Formats entiers
- Octets, mots, mots doubles, mot quadruple
- Utilisation des 8 registres flottants
- Arithmétique modulaire ou saturée
  - Saturation signée ou non

Stockage mémoire little endian



SSE – Internet Streaming SIMD Extensions

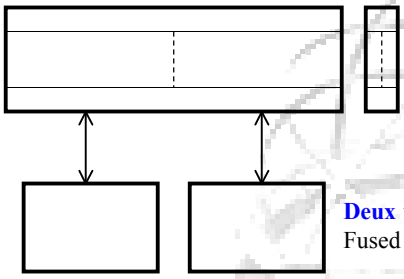
- Nouveaux registres (8) dédiés
- Formats flottant
  - Simple précision (Pentium 3)
  - Double précision sur SSE2 (Pentium 4)
- Gestion logicielle des dénormalisés



Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA
© Propriété de l'auteur

Architecture de haut niveau de l'IA 64

Registres numérotés (128)
Registres statiques (32)



Prédicats (64)
Prédicats statiques (16)

Deux unités flottantes
Fused MAC (5 cycles)

Architecture au choix
Big endian et Little endian

L2 - 9 cycles
L3 - 24 cycles

Unités RISC extrême
Instructions load et store sans opération
Opérandes et résultat dans registres

Disparition de la division, de la racine carrée et des fonctions élémentaires

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA

Encore de nouveaux changements

Nouveau type interne de données étendu

	Exposant (biaisé)	Mantisse (Premier bit explicite)

– Double étendu 1 - 17 - 63 + 1

Architecture 64 bits

Multiplicité des mots de contrôle
(Exécution prédiquée)

Instructions non normalisées IEEE
– Routines logicielles de correction
– Exposant supprime les problèmes d'infini

– Arithmétique d'intervalle efficace
– Contrôle étendu sur l'exécution spéculative

Correction automatique / Contrôle de l'erreur

Risque de tentation de l'utilisateur
Vite et mal

Quadruple précision
Toujours pas implantée

Marc Daumas – CNRS – LIP
Projet ARENAIRE – ENS de Lyon – CNRS – INRIA