

Bonjour

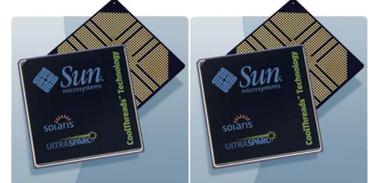
# Fenêtres de Registres SPARC

- Problème
- Fenêtres de registres
- Avantages
- Cas extrêmes
- Autres usages
- Allocation globale
- Performances
- Questions?
- Références

Savaş Ali TOKMEN

M2P Génie Informatique

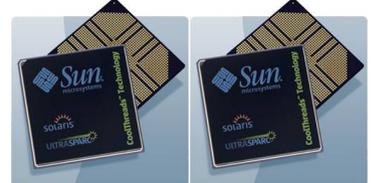
UFR IMA



# Problème

- **Problème**
- Fenêtres de registres
- Avantages
- Cas extrêmes
- Autres usages
- Allocation globale
- Performances
- Questions?
- Références

- Registres utilisés pour stocker les données d'une fonction
  - Rapide
- **Problème 1: l'appel de fonction est lent**
  - Sauvegarder registres
  - Appeler
  - Restaurer registres
- **Problème 2: impossible d'augmenter le nombre de registres**



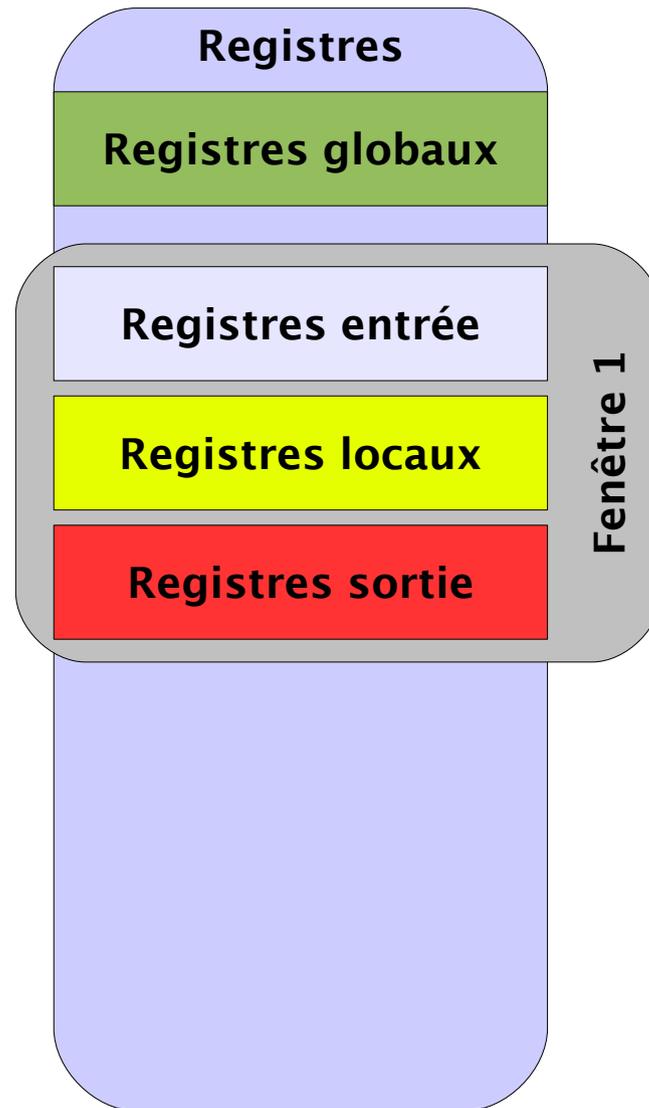
# Fenêtre de registres

- Problème
- Fenêtres de registres
  - Bases
  - Usage
- Avantages
- Cas extrêmes
- Autres usages
- Allocation globale
- Performances
- Questions?
- Références

- Quatre groupes de registres:
  - Registres globaux
  - Registres « entrée »
  - Registres « sortie »
  - Registres locaux
- Registres entrée, sortie et locaux spécifiques à chaque fonction appelée

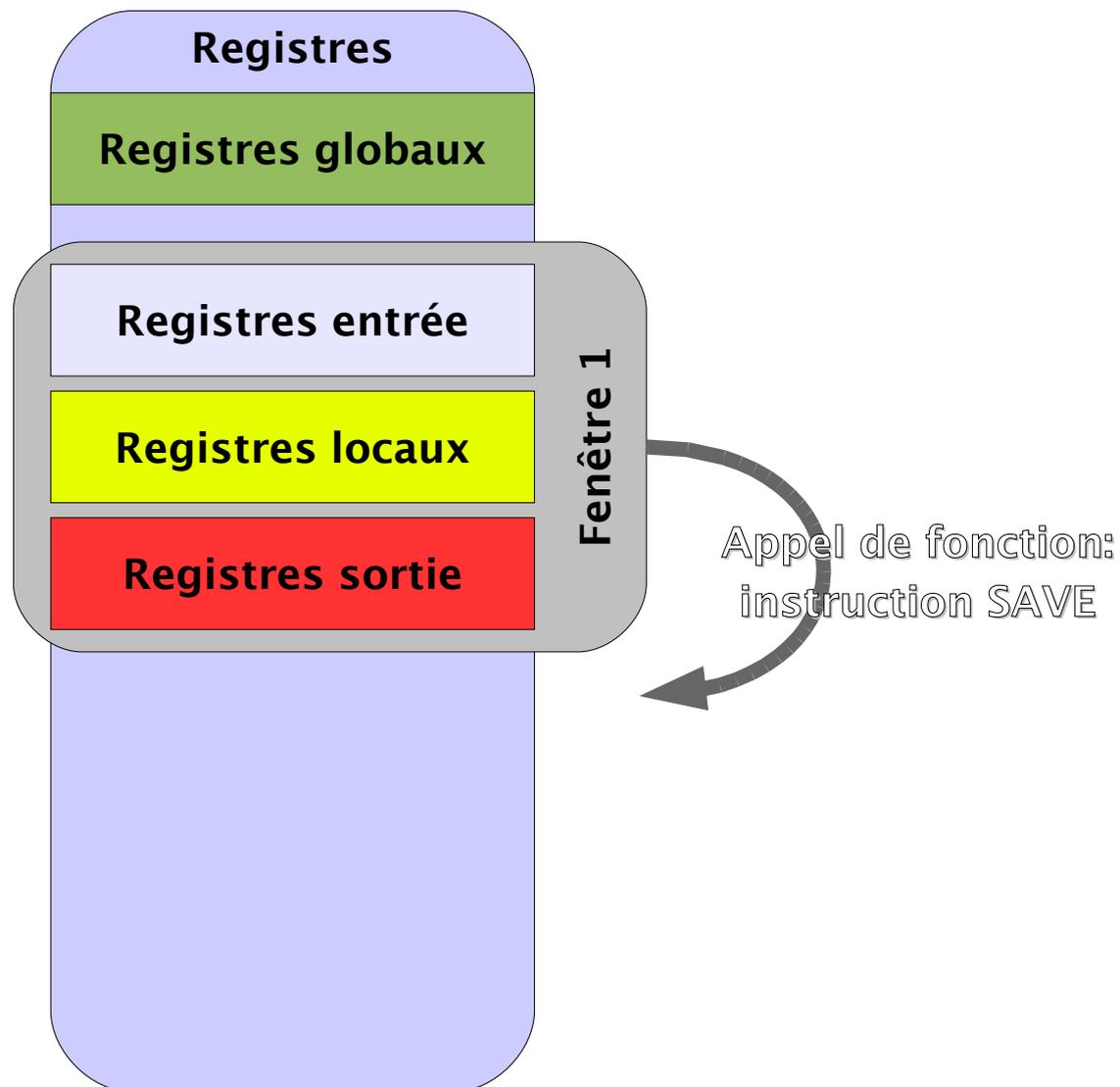
# Fenêtre de registres

- Problème
- Fenêtres de registres
  - Bases
  - Usage
- Avantages
- Cas extrêmes
- Autres usages
- Allocation globale
- Performances
- Questions?
- Références

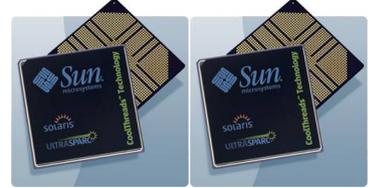


# Fenêtre de registres

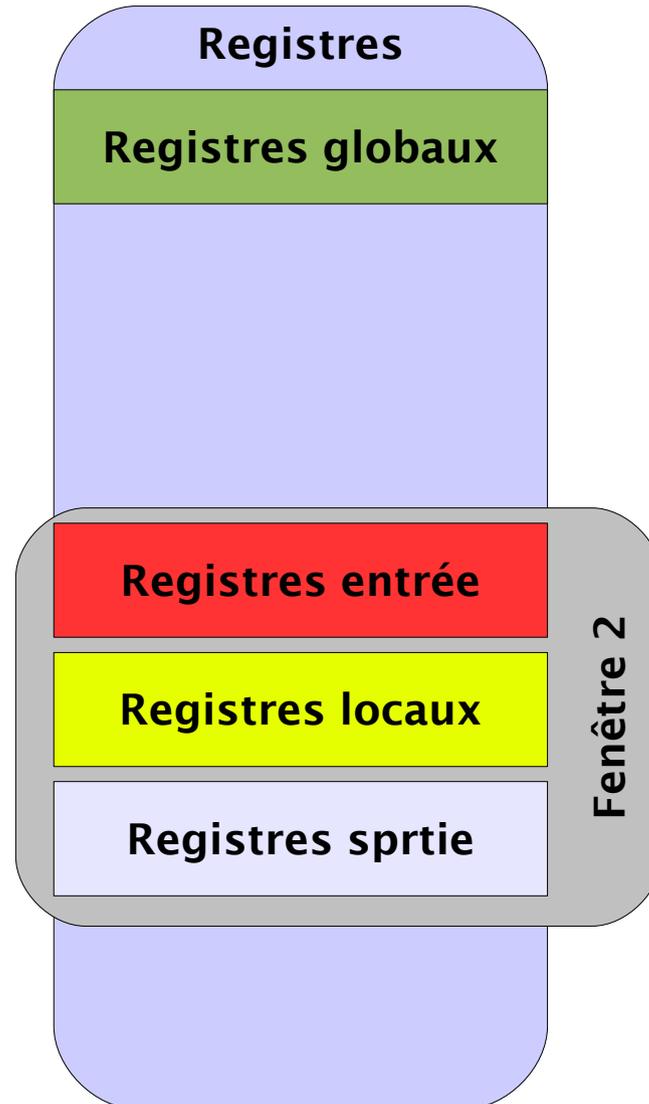
- Problème
- Fenêtres de registres
  - Bases
  - Usage
- Avantages
- Cas extrêmes
- Autres usages
- Allocation globale
- Performances
- Questions?
- Références



# Fenêtre de registres

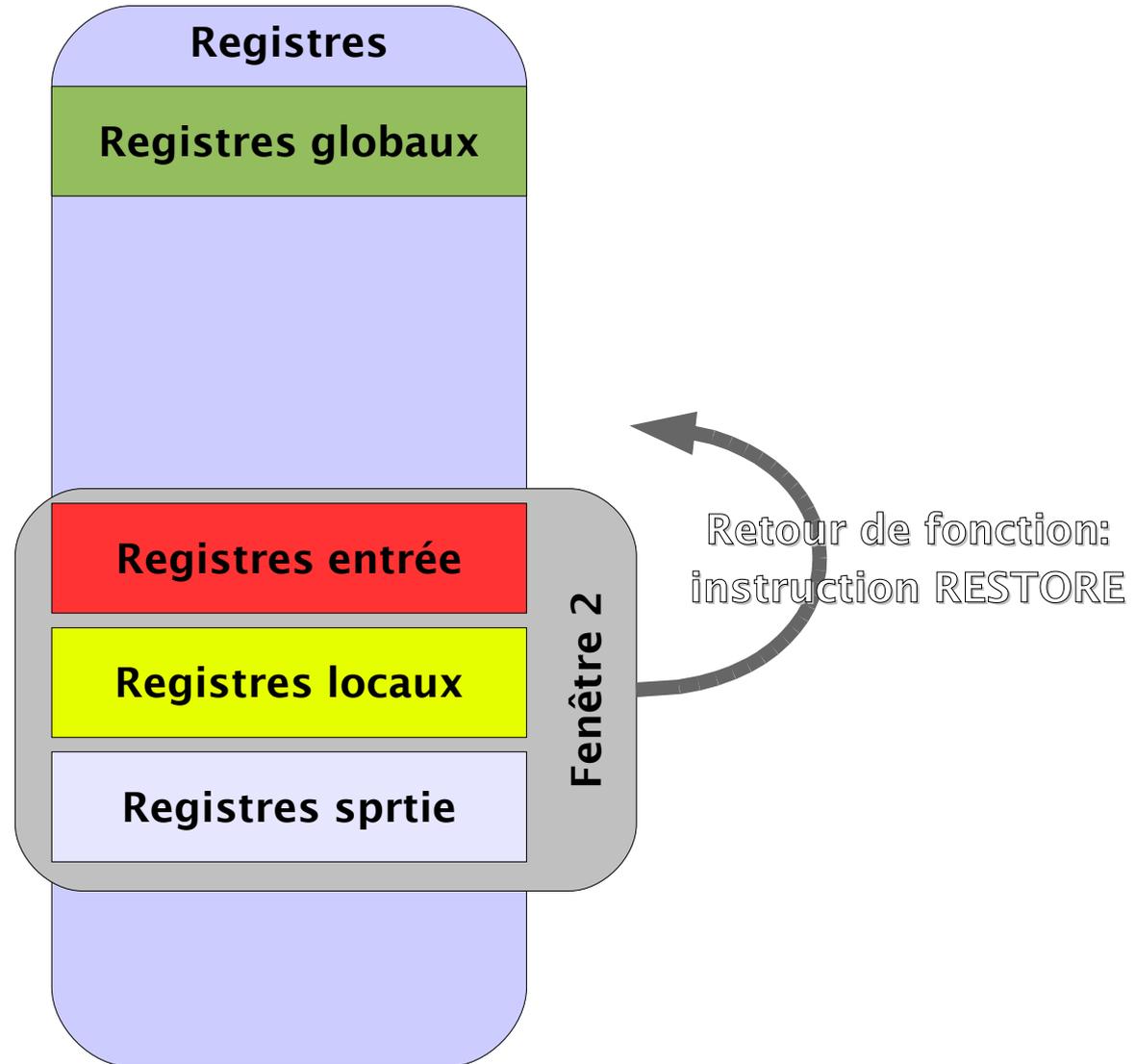


- Problème
- Fenêtres de registres
  - Bases
  - Usage
- Avantages
- Cas extrêmes
- Autres usages
- Allocation globale
- Performances
- Questions?
- Références



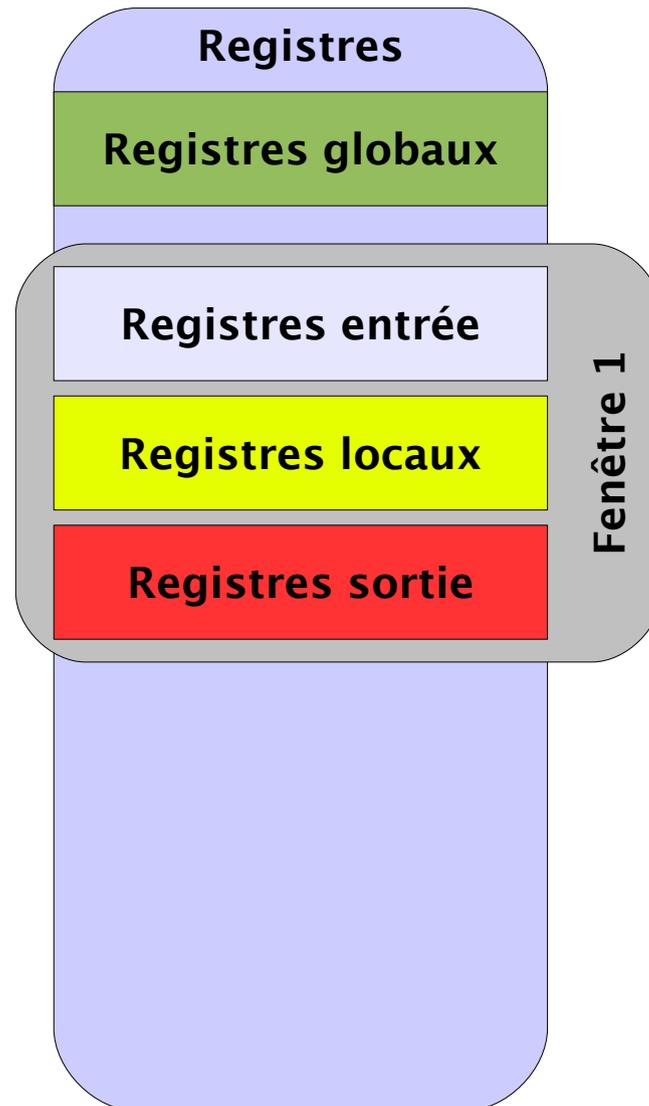
# Fenêtre de registres

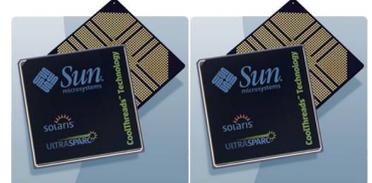
- Problème
- Fenêtres de registres
  - Bases
  - Usage
- Avantages
- Cas extrêmes
- Autres usages
- Allocation globale
- Performances
- Questions?
- Références



# Fenêtre de registres

- Problème
- Fenêtres de registres
  - Bases
  - Usage
- Avantages
- Cas extrêmes
- Autres usages
- Allocation globale
- Performances
- Questions?
- Références



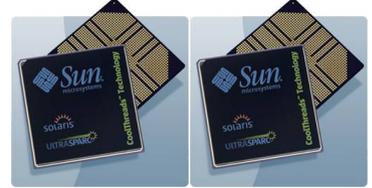


# Avantages

- Problème
- Fenêtres de registres
- **Avantages**
- Cas extrêmes
- Autres usages
- Allocation globale
- Performances
- Questions?
- Références

- Plus rapide
  - Appel direct de fonction, sans aucune interaction avec la pile
- Moins gourmand
  - Fenêtres combiné avec un fetch « intelligent » des instructions => 53% d'économie en puissance consommée!
- Extensible
  - Le nombre de registres globaux, locaux, entrée et sortie reste identique (8), on change le nombre de fenêtres (2 à 32)

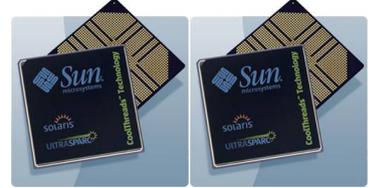
# Cas extrêmes



- Problème
- Fenêtres de registres
- Avantages
- **Cas extrêmes**
- Autres usages
- Allocation globale
- Performances
- Questions?
- Références

- Overflow et underflow
  - Overflow: tous les fenêtres sont occupés et on appelle SAVE
  - Underflow: tous les fenêtres sont vides et on appelle RESTORE
- Génération de TRAP
  - Géré par le système d'exploitation
  - Interaction avec une pile
- Problème du déterminisme
  - Surtout dans un cas multi-threadé

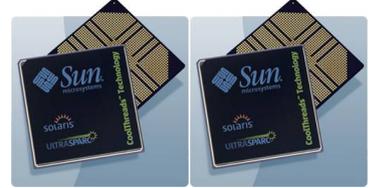
# Autres usages



- Problème
- Fenêtres de registres
- Avantages
- Cas extrêmes
- **Autres usages**
- Allocation globale
- Performances
- Questions?
- Références

- Motivations
  - Un bon RTOS ne promet pas rapidité mais prédictibilité
  - Les fenêtres créent une certaine indépendance
- Une fenêtre par thread
  - Temps de commutation plus courte
  - Performances accrues tout en restant déterministe sur le temps maximal d'exécution

# Allocation globale



- Problème
- Fenêtres de registres
- Avantages
- Cas extrêmes
- Autres usages
- **Allocation globale**
- Performances
- Questions?
- Références

- Toutes les registres sont disponibles à tout moment à tout programme
- Compilateur (optimiseur) s'occupe de l'allocation des registres
- Problème NP-complet (coloriage de graphe)
- Intel Pentium / Itanium / ...
- **Pas extensible**

# Performances

- Problème
- Fenêtres de registres
- Avantages
- Cas extrêmes
- Autres usages
- Allocation globale
- Performances

- Selon [Wall], « Link-time register allocation based on profile frequencies does best across the board. »

Processor	AMD Athlon MP	Intel Itanium	Sun Ultra-II	Sun Ultra-III
Clock Rate	1.6GHz	800MHz	480MHz	1,050MHz
Cache (I/D/L2)	64K/64K/256K	16K/16K/96K	16K/16K	32K/64K
Issue Rate	3 x86 instr	6 issue	4 issue	4 issue
Pipeline Stages	9/11 stages	10 stages	6/9 stages	14/15 stages
Out of Order	72RCPs	None	None	None
Rename Regs	36/36	328 total	None	None
BHT Entries	4K x 2-bit	512 x 2-bit	512 x 2-bit	16K x 2-bit
TLB Entries	780/288	64I/32L1D/96L2D	64I/64D	128I/512D
Memory B/W	2.1GB/s	2.1GB/s	1.9GB/s	4.8GB/s
Package	PGA-462	PAC-418	CLGA-787	FC-LGA 1368
IC Process	0.18x 6M	0.18x 6M	0.29x 6M	0.15x 7M
Die Size	128mm <sup>2</sup>	300mm <sup>2</sup> *	126 mm <sup>2</sup>	210mm <sup>2</sup>
Transistors	37.5 million	25 million	3.8 million	29 million
Est Die Cost	\$41*	\$170*	\$30*	\$72*
Power (Max)	66W	130W	20W*	75W*
Availability	4Q01	2Q01	3Q00	1Q02

Processor	AMD Athlon MP	Intel Itanium	Sun UltraSPARC II	Sun UltraSPARC III
System or Motherboard	EpoX 8KHA+	HP 12000	Sun Enterprs 450	Sun Blade 2050
Clock Rate	1.6GHz	800MHz	480MHz	1,050MHz
External Cache	None	4MB	8MB	8MB
164.gzip	811	332	165	433
175.vpr	423	376	212	460
176.gcc	460	407	232	577
181.mcf	343	402	356	659
186.crafty	942	356	175	558
197.parser	621	296	211	488
252.eon	1280	370	209	527
253.peribmk	982	320	247	540
254.gap	776	256	171	372
255.vortex	1017	459	304	738
256.bzip2	554	334	237	629
300.twolf	534	449	243	570
SPECint_base2000	677	358	225	537
SPECfp_base2000	588	703*	274	701

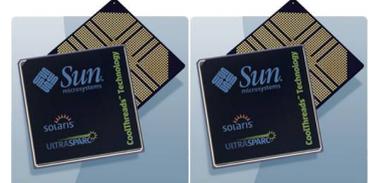
Graphes de Microprocessor Report, Cahners, 2002

# Questions ?

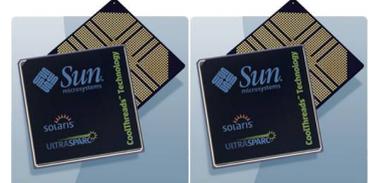
- Problème
- Fenêtres de registres
- Avantages
- Cas extrêmes
- Autres usages
- Allocation globale
- Performances
- **Questions?**
- Références

Questions ?

# Références

- 
- Problème
  - Fenêtres de registres
  - Avantages
  - Cas extrêmes
  - Autres usages
  - Allocation globale
  - Performances
  - Questions?
  - **Références**

- Determinism in RTOS Design for SPARC-like Architectures, University of Alcalá
- Exploiting SPARC Buffer Overflow vulnerabilities, UNF
- Reducing instruction fetch cost, Florida State University, 2005
- Register Renaming, Ecole Polytechnique Fédérale de Lausanne
- Register Renaming, Wikipedia
- Register Windows, Barney Maccabe, 1996
- Register Windows, Nikos Drakos, University of Leeds
- Register Windows, University of New Mexico
- Register Windows, Wikipedia
- Register Windows vs. Register Allocation, David W. Wall, Digital Equipment Corporation
- UltraSparc Unleashes SPARC Performance, Microprocessor Report, 1994



# Merci

# Merci

Documents disponibles sur

<http://scholar.alishomepage.com/Master/RegisterWindows/>

