



RTOS et gestion matérielle de la mémoire



Conception de systèmes embarqués complexes

Cours de A. Fraboulet et T. Risset

Exposé de Pierre PARREND (pierre.parrend@insa-lyon.fr)
www.rzo.free.fr/master.html



Sommaire

Contexte

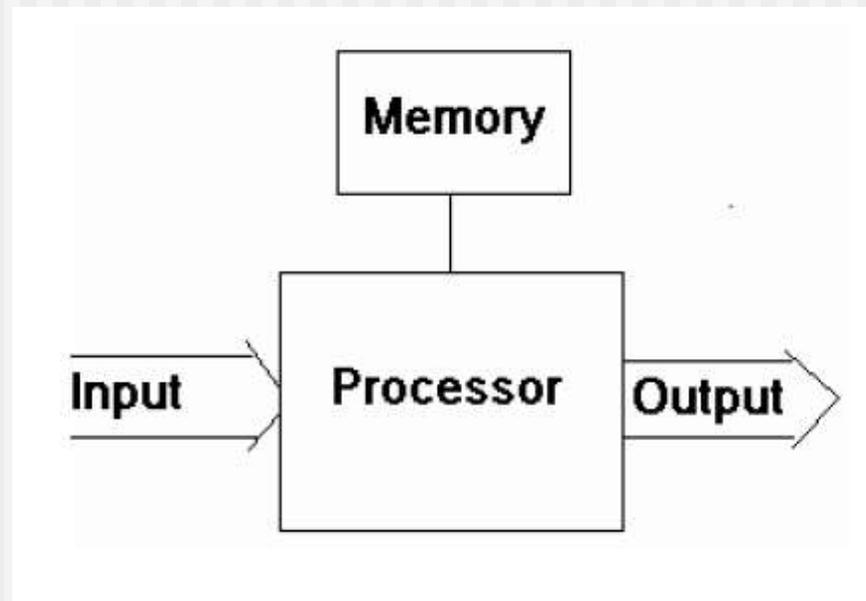
Article

Critique et perspective



Contexte [Sagar02]

Palier 1 : les systèmes embarqués





Contexte

Palier 1 bis : les Systèmes d'Exploitation (SE)

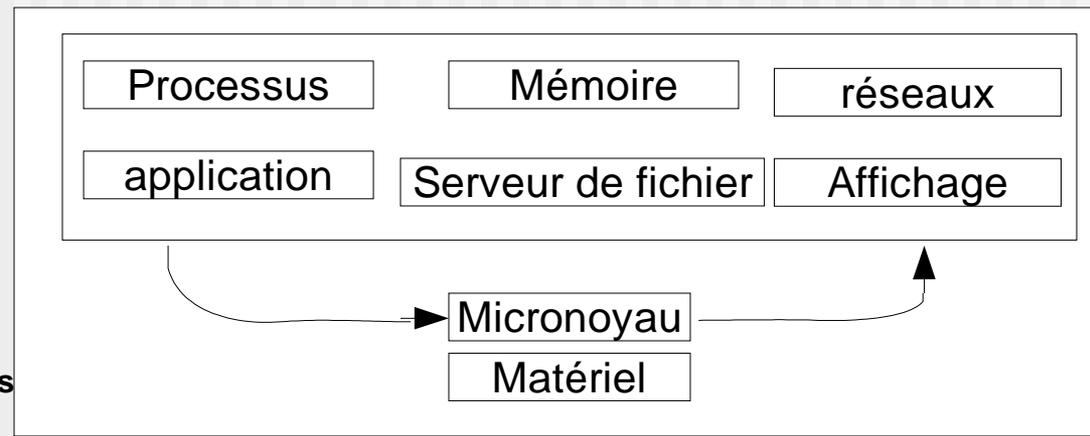
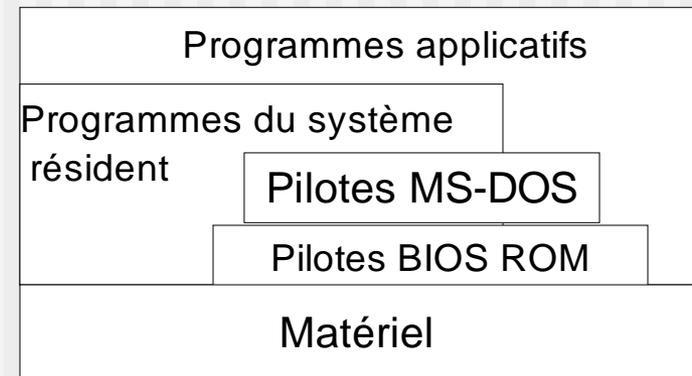
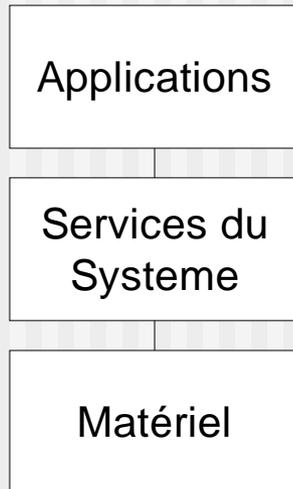
- Mono-utilisateur, mono-tâche (Palm-OS)
- Mono-utilisateur, multi-tâche (Win98, MacOS)
- Multi-utilisateur (Unix, Windows-NT)
- Systèmes d'exploitation temps réel (RTOS)





Contexte

Palier 1 bis : les SE - architecture





Contexte

Palier 1 bis : les SE - rôle

- Gestion du processeur
- Gestion de la mémoire et du stockage
- Gestion des périphériques
- Interface de programmation (API)
- Interface utilisateur





Contexte

Palier 2 : les SE pour systèmes embarqués

- Complexité croissante (SoC)
- Respect des contraintes temps réel
- Conservation d'un temps de développement raisonnable
- Utilisation des ressources disponibles

SE à micro-noyau mieux adaptés





Contexte [LPW97]

Palier 3 : les Systèmes sur puce (SoC)

Ou : Systèmes sur Silicium (SoS)

- + 20.10e6 transistors : HDTV, caméscope, radio logicielle sur une seule puce
- Plusieurs CPU
- Unités fonctionnelles spécifiques (DSP ...)
- RAM embarquée en grande quantité





Contexte [LPW97]

Palier 4 : Evolution des SoC

RTOS + importants

Chers : doivent être fiables, performants

Recherche

- Optimisation de l'ordonnancement
- Co-design

Industrie

- Optimisation de la mémoire
- Temps de changement de contexte





Contexte [LPW97]

Palier 4 : Evolution des SoC

Problèmes

- RTOS et synthèse de niveau système
- RTOS et applications embarquées
- Débuggage (complexité croissante du matériel et logiciel)
- Flexibilité, scalabilité
- Intégration de DRAM et processeur
- Augmentation du volume de données traitées





Contexte [LPW97]

Palier 5 : Gestion de la mémoire langage

- RTSJ, prise en charge de l'allocation mémoire à la compilation [KW04]

Systeme

- Sémaphores [ZS97]

Algorithmie

- Optimisation de l'ordonnancement [MRC04]

Matériel





Sommaire

Contexte

Article

Gestion matérielle de la mémoire dans un RTOS

Hardware Support for Real-Time Embedded Multiprocessor System-on-a-Chip Memory Management, Mohamed Shalan, Vincent J. Mooney

CODES'02. May 6-8, 2002

Critique et perspective



L'Article

Les Auteurs

Mohamed Shalan

Vincent J. Mooney

Georgia Institute of Technology





L'Article

Palier 6 : Travaux antérieurs

Allocation statique

- Avec synthèse de hiérarchie de mémoire [Wuytack99]
- Pb si évolution des contraintes

Allocation dynamique

- Simple et rapide [Knowlton65]
- Sans fragmentation [Puttkamer75]





L'Article

Palier 6 : Travaux antérieurs

Accélération de la mémoire

- Implémentation matérielle des fonctions C (malloc, free, realloc) [SLC99]
- Accélérateur de gestion dynamique de la mémoire [Chang99]
- Détection des blocs libres d'une taille donnée [Cam99]

Introduction de déterminisme

- Elimination de la fragmentation, temps d'exécution constant [CG96]



L'Article

Palier 7 : la solution

Allocation dynamique

Déterminisme

Vitesse



System on-a-Chip Dynamic Memory Management Unit (SoCDMMU)



L'Article

Palier 7 : la solution

Gestion hiérarchique de la mémoire

- Niveau 1 : le processeur gère la mémoire qui lui est allouée
- Niveau 2 : SoCDMMU, alloue la mémoire aux processeurs
- Dynamisme

Pagination

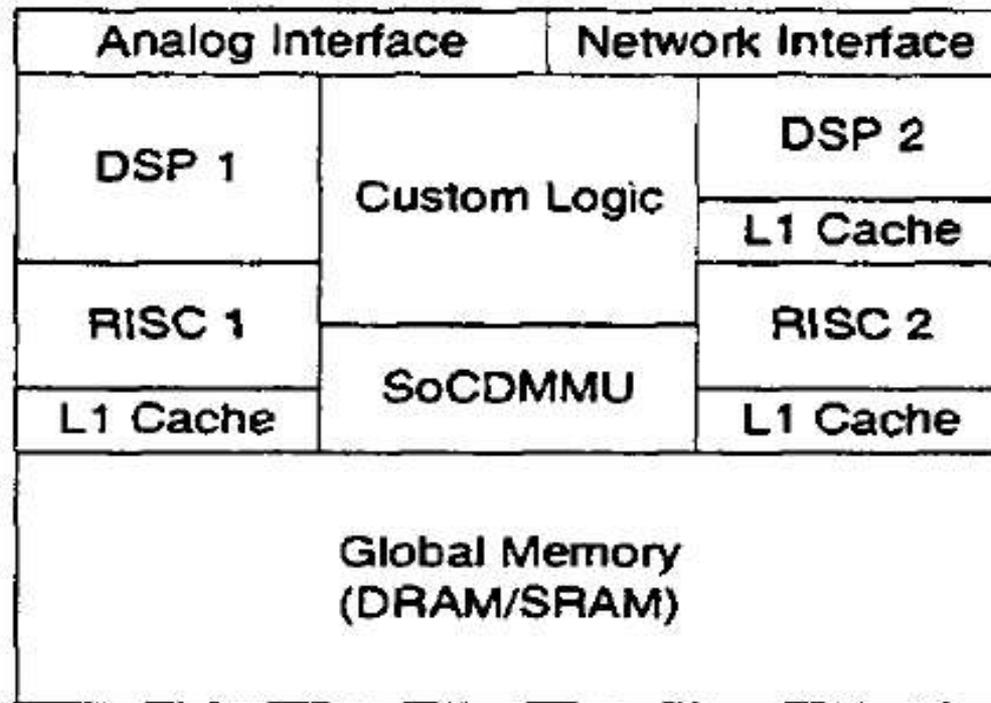
- Déterminisme





L'Article

Palier 7 : la solution

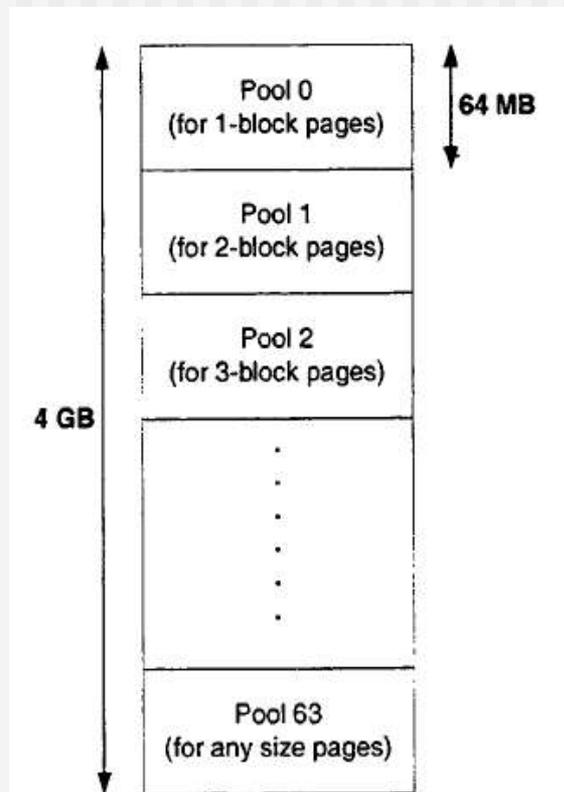




L'Article

Palier 7 : la solution

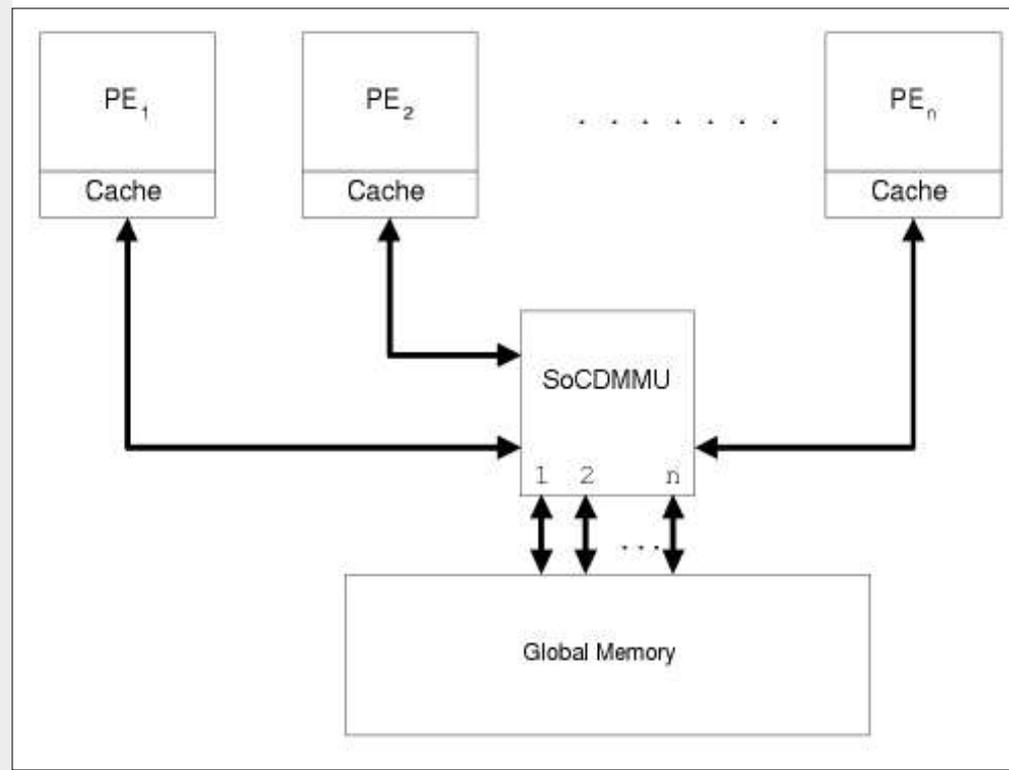
Pagination : RTOS Atalanta [Mooney02]





L'Article

Palier 7 : la solution Hiérarchie de mémoire





L'Article

Palier 7 : la solution

Intégration dans le RTOS

- Le SoCDMMU doit accéder aux processeurs pour faire l'allocation
- L'adresse détermine la taille du bloc
- Extension nécessaire de l'API

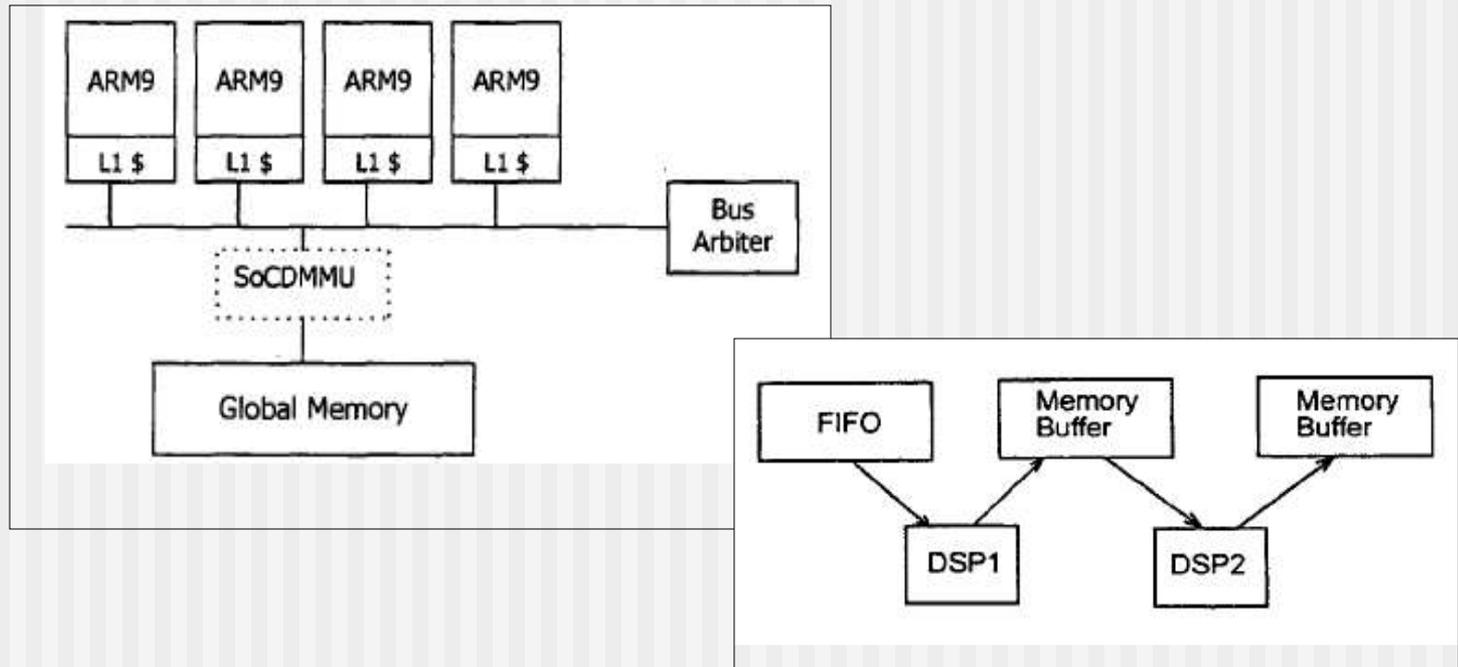


Nom de la fonction	Description
asc_partition_create	Création d'une partition par le SoCDMMU
asc_partition_delete	Destruction d'une partition par le SoCDMMU
asc_memory_find	Trouver un place dans l'espace d'adressage du processeur



L'Article

Palier 7 : la solution Validation





L'Article

Palier 7 : la solution

Validation

- SoCDMMU vs. implémentation matérielle des fonctions d'allocation C
- Gain de 440 %



	SoCDMMU	Fonctions C	Gain
Allocation	28 Cycles	106 Cycles	3,78 x
Libération	14 Cycles	83 Cycles	5,9 x
Experience	280 Cycles	1240 Cycles	4,4 x

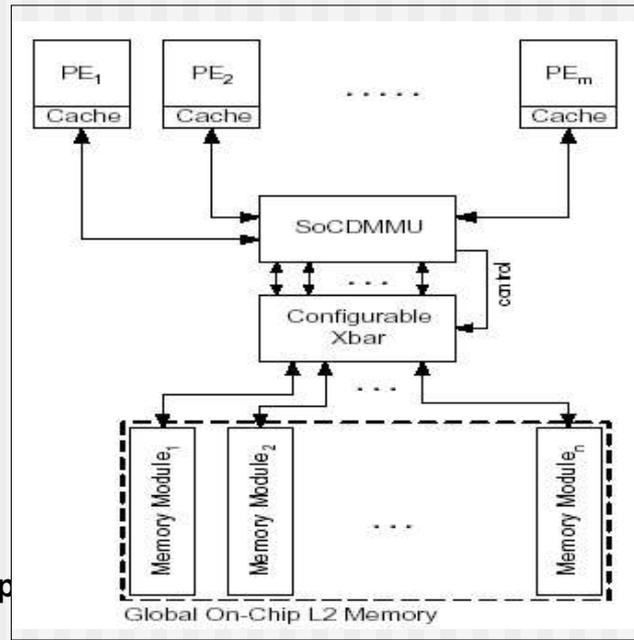
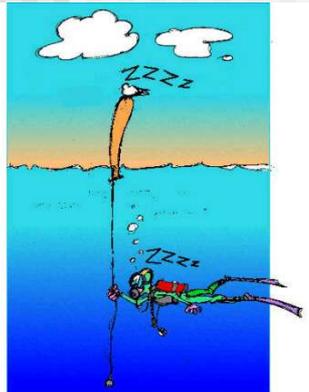


L'Article

Et ensuite ...

SoCDMMU et co-design [SH03]

- Génération dynamique pour M processeurs et N éléments mémoire



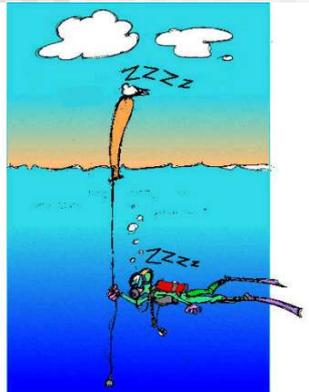


L'Article

Et ensuite ...

SoCDMMU et co-design [SH03]

- CrossBar pour le lien entre les processeurs et la mémoire
- Nécessaire si les bus sont insuffisants (>8 proc.)
- Premiers travaux du genre (selon l'auteur)





Sommaire

Contexte

Article

Critique et perspective



Critiques positives

Gain qualitatif et quantitatif

- Performances bonnes
- Déterminisme et dynamisme
- Scalabilité

Partie d'un projet complet

- Aboutit à une technique exploitable

Destiné à être réutilisé (si confirmation de la faisabilité)



Critiques négatives

Problème de l'absence de transparence
(modification de l'API)

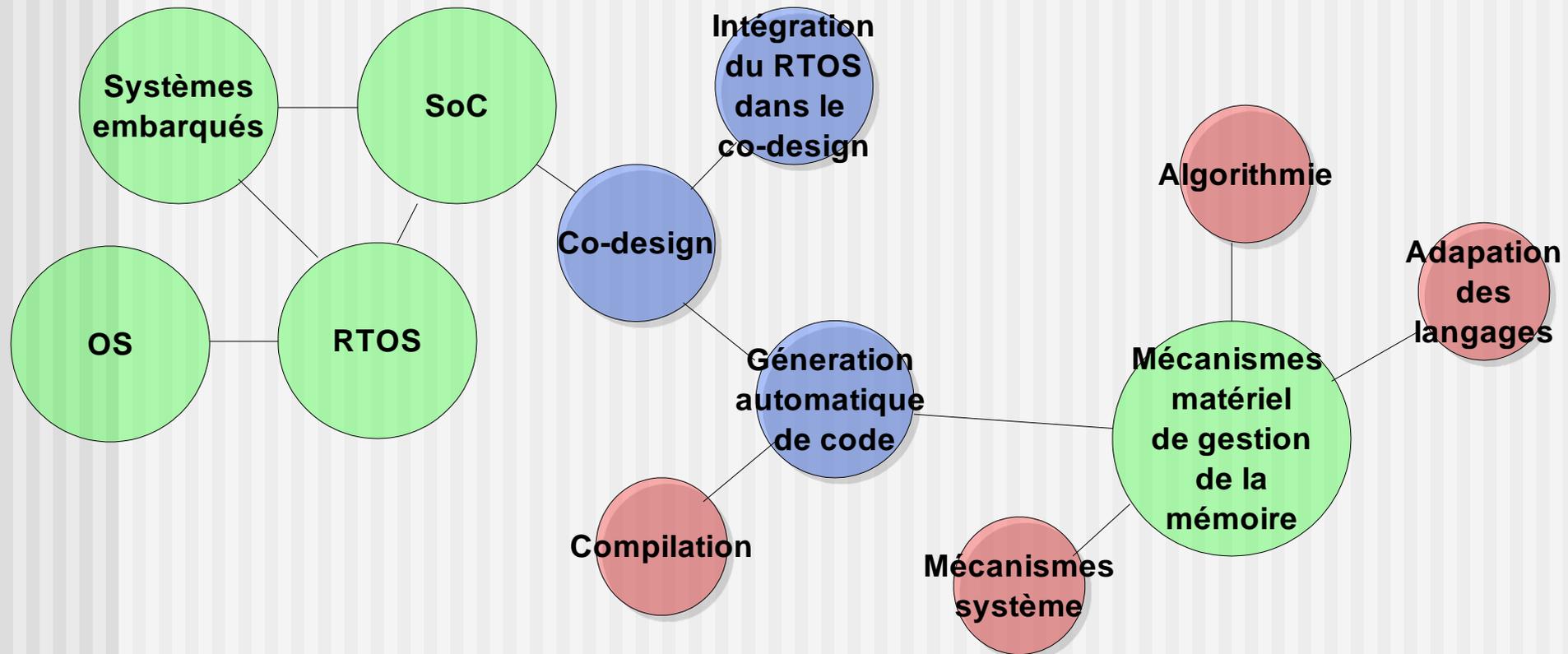
- Difficile à intégrer dans des systèmes existants ?

Manque une validation externe

- Validation des résultats de l'expérience
- Eventuelles restrictions

Mais travail conséquent et intéressant

Thèmes abordés





Perspectives

Aboutissement d'un mécanisme efficient et efficace

- Gain en exploitation (performances)
- Gain en conception (co-design)
- Scalabilité

Appel d'autres améliorations ?

- Synthèse de niveau système
- Algorithmes



Où trouver des infos ?

Conférences

- International Conference on Computer Design (ICCD)
- Real-Time Technology and Applications Symposium
- Euromicro Conference on Real-Time Systems (ECRTS)
- ICCD Workshop on Hardware Support for Objects and Micro architectures for Java
- Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI)



Où trouver des infos ?

Journaux

- Communications ACM
- IEEE Transaction on Computers
- IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



Bibliographie

[Sagar02] Embedded Operating Systems for Real-Time Applications
Sagar P M (02307406) Supervisor: Prof. Vivek Agarwal. 11.2002

[LPW97] Real-Time Operating Systems for Embedded Computing
Yanbing Li , Miodrag Potkonjak, and Wayne Wolf, Proceedings of the
1997 International Conference on Computer Design (ICCD '97)

[KW04] Memory Management Based on Method Invocation in RTSJ, J.
Kwon, A. Wellings, Lecture Notes in Computer Science (3292)
Proceedings of the OTM 2004 Workshops: Workshop on Java
Technologies for Real-Time and Embedded Systems (JTRES), 2004

[ZS97] K. M .Zuberi and K. G. Shin, "An efficient semaphore
implementation scheme for small-memory embedded systems," in Proc.
Real-Time Technology and Applications Symposium, pp. 25-34, June

1997

24/01/2005



Bibliographie

[MRC04] Dynamic storage allocation for real-time embedded systems, 'TLSF: A New Dynamic Memory Allocator for Real-Time Systems', M. Masmano, I. Ripoll, and A. Crespo, ECRTS04

[Wuytack99] S. Wuytack et al., "Memory Management for Embedded Network Applications," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 18, no. 5, pp. 533, May 1999.

[Knowlton65] K.C. Knowlton, A Fast Storage Allocator, Communications ACM, Vol. 8, October 1965, pp. 623-625.

[Puttkamer75] E. V. Puttkamer, A simple hardware buddy system memory allocator, IEEE Transaction on Computers, vol. 24, no. 10, October 1975, pp. 953-957.



Bibliographie

[SLC99] W. Srisa-an, C. D. Lo, and J. M. Chang, "A Hardware Implementation of Realloc Function," Proceedings of WVLSI'99 IEEE Annual Workshop on VILSL April 1999, pp. 106-111.

[Chang99] J. M. Chang et al., Introduction to DMMX (Dynamic Memory Management Extension), Proceedings of ICCD Workshop on Hardware Support for Objects and Micro architectures for Java, October 1999, pp. 11-14.

[CG96] J. M. Chang and E. F. Gehringer, A High-Performance Memory Allocator for Object-Oriented Systems, IEEE Transactions on Computers, vol. 45, no. 3, March 1996, pp. 357-366.

[Cam99] H. Cam et al., A high-performance hardware-efficient memory allocation technique and design, Proceedings of International Conference on Computer Design (ICCD 99), October 1999, pp. 274-276.



Bibliographie

[Mooney02] D. Sun, D. M. Blough, and V. J. Mooney, "Atalanta: A New Multiprocessor RTOS Kernel for System-on-a-Chip Applications", Georgia Institute of Technology, Atlanta, Georgia, Technical Report GIT-CC-02-19, 2002

[SH03] M. Shalan, E. Shin and V. Mooney, "DX-Gt: Memory Management and Crossbar Switch Generator for Multiprocessor System-on-a-Chip," Proceedings of the 11th Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI'03), pp. 357-364, April 2003.



Des Questions ?





Contexte

SE pour les systèmes embarqués

Micro-noyaux

- Erreur dans un module : pas de conséquence sur les autres
- Doit pouvoir être distribué
- Redondance
- Chargement (et déchargement) dynamique de modules



Contexte

SE pour les systèmes embarqués

Implémentations

- Inferno (Lucent) : avec sécurité intégrée
- VxWorks, différentes versions
 - Générique,
 - Pour l'automobile,
 - L'électronique gd public,
 - L'industrie
- pSOS Systems
- Lynx Real-Time Systems (LynxOS)



Contexte

Techniques d'ordonnancement

- *Cooperation* : le processus en cours rend la main. Systèmes très simples (filtres numériques avec entrées périodiques)
- *A priorités statiques* : le processus le plus prioritaires s'exécute, les autres attendent (ex : RMS)
- *A priorités dynamiques* : évolution des priorités, en fonction des politiques d'ordonnancement (ex : EDF, Earliest Deadline First)



Contexte

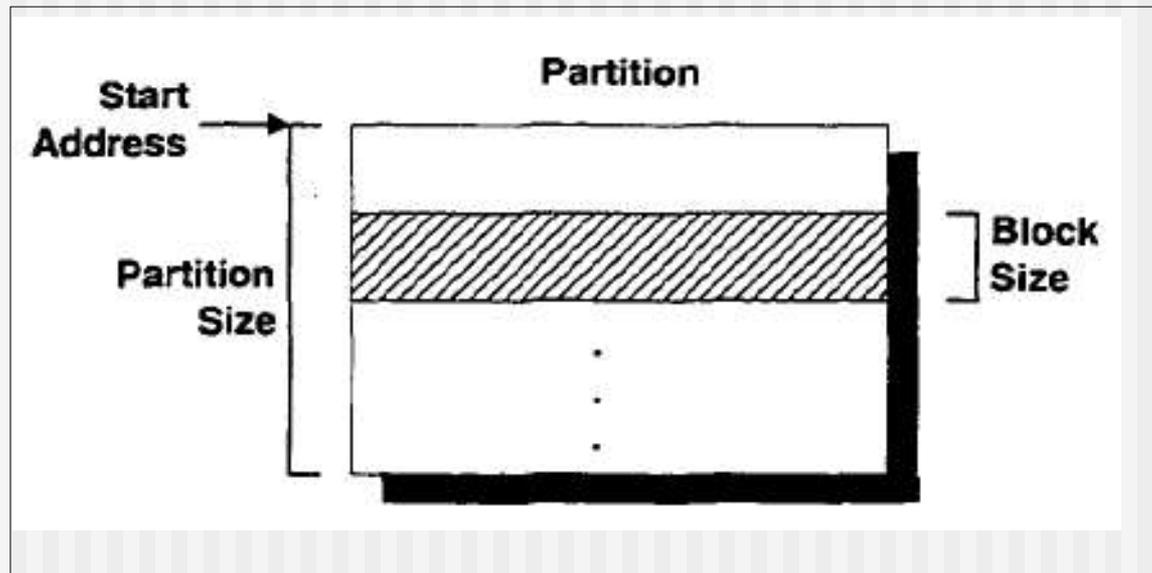
Evolutions récentes des SoC

- Balance de charge en fonction de profils
- Ordonnancement loterie
- Génération dynamique de code (performant en terme de débit, latence, mémoire)
- Mécanismes de sécurité et d'authentification embarqués (ex : Inferno)
- Masque de gigue (semble performant)
- Vérification
- Extension pour CORBA



L'Article

Partitionnement de la mémoire





L'Article

API existante dans Atalanta

Fonction

asc_partition_gain
asc_partition_seek
asc_partition_free
asc_partition_reference

Description

Obtenir un bloc de mémoire d'une partition
Obtenir un bloc de mémoire d'une partition
Libérer un bloc de mémoire
Demande d'information sur une partition