Conception et systèmes embarqués complexes

Master 2004

Antoine Fraboulet, Tanguy Risset
antoine.fraboulet@insa-lyon.fr, tanguy.risset@ens-lyon.fr
Lab CITI, INSA de Lyon, Lab LIP, ENS de Lyon

- Processeurs embarqués
- Introduction
- Architecture des processeurs
- Différents types de processeurs embarqués
- Compilation pour processeurs embarqués
- Exemple de l'appareil photo numérique
- Conclusion

Processeurs embarqués

Part de marché

- Processeurs embarqués

Introduction
- Part de marché
- Contradiction ?
- Variété des processeurs embarqués

Architecture des processeurs

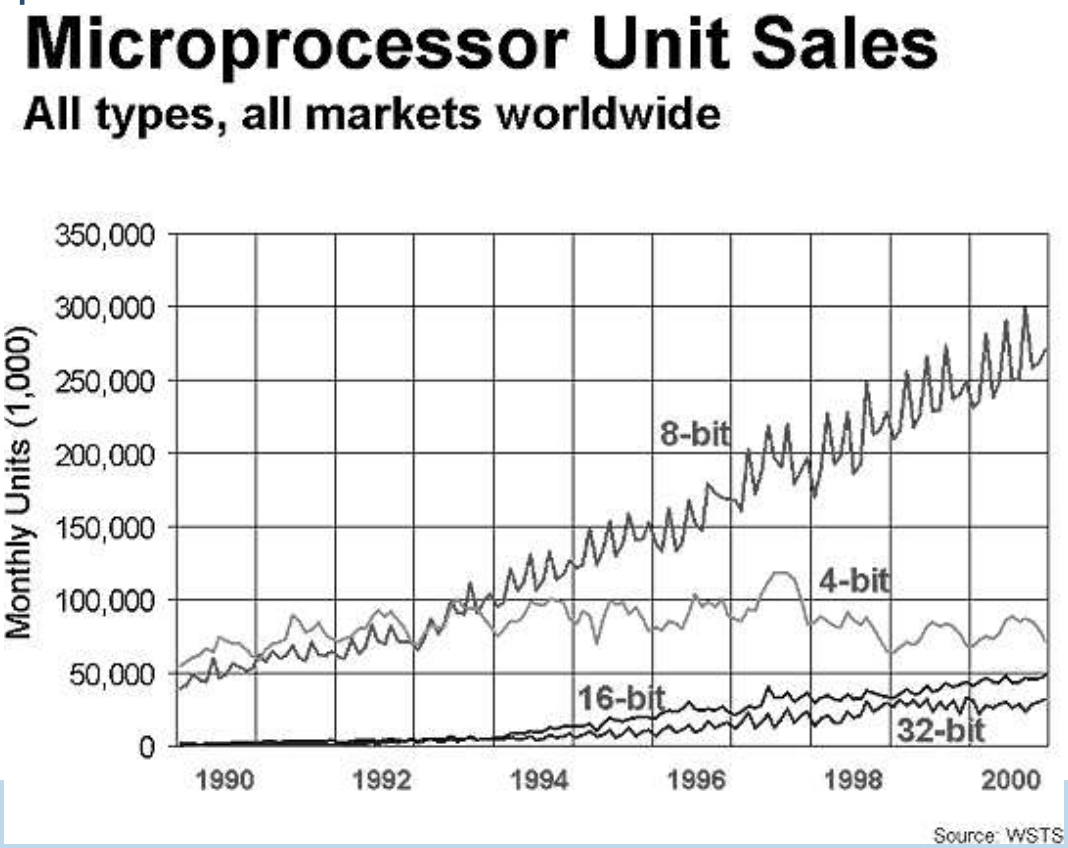
Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- Quel est le le microprocesseur le plus vendu ?
  - Réponse classique: "Le Pentium: 92% du marché"
- Faux!.....
  - En fait les Pentium ne représentent que 2% des microprocesseurs vendus dans le monde.



Contradiction ?

- Processeurs embarqués

Introduction
- Part de marché
- Contradiction ?
- Variété des processeurs embarqués

Architecture des processeurs

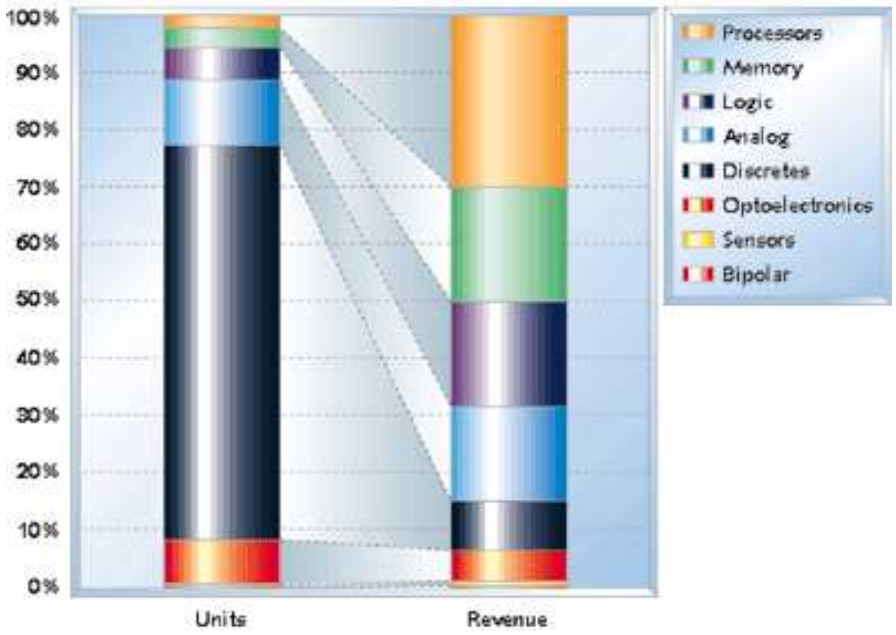
Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- Alors d'ou vient la position d'Intel (16% du marché des semi-conducteurs) ?
- processeurs: 2% du silicium, 30% des revenus



Et au sein des processeurs

- Processeurs embarqués

Introduction
- Part de marché
- Contradiction ?
- Variété des processeurs embarqués

Architecture des processeurs

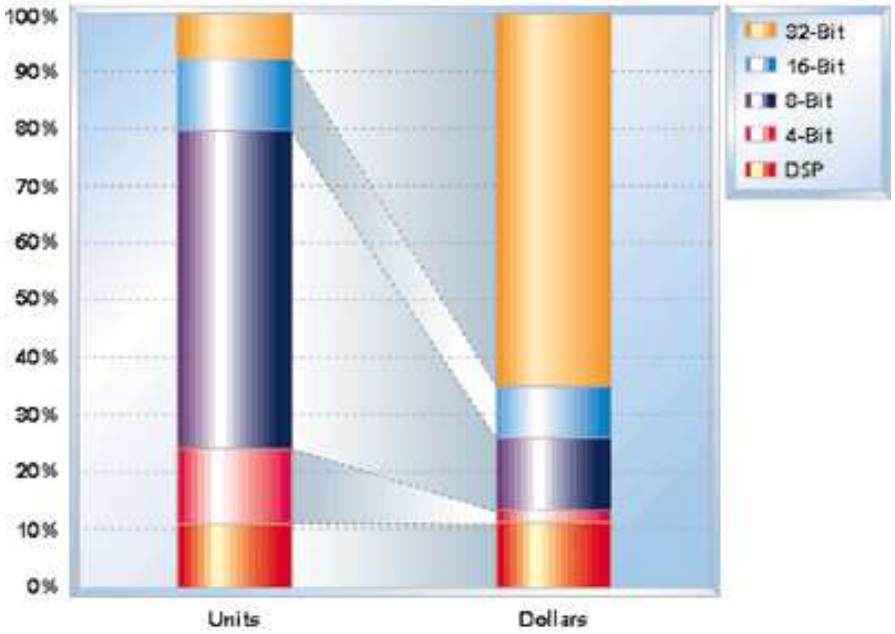
Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- 3 milliards de processeurs 8 bits vendus par an (8051, 6805 etc.)
- 32 bits (Pentium, Athlon, mais aussi PowerPC, 68000, MIPS, ARM etc.)
- La plupart (98%) sont embarqués (3 fois plus d'ARM vendus que de Pentium)



Variété des processeurs embarqués

- Processeurs embarqués

Introduction
- Part de marché
- Contradiction ?
- Variété des processeurs embarqués

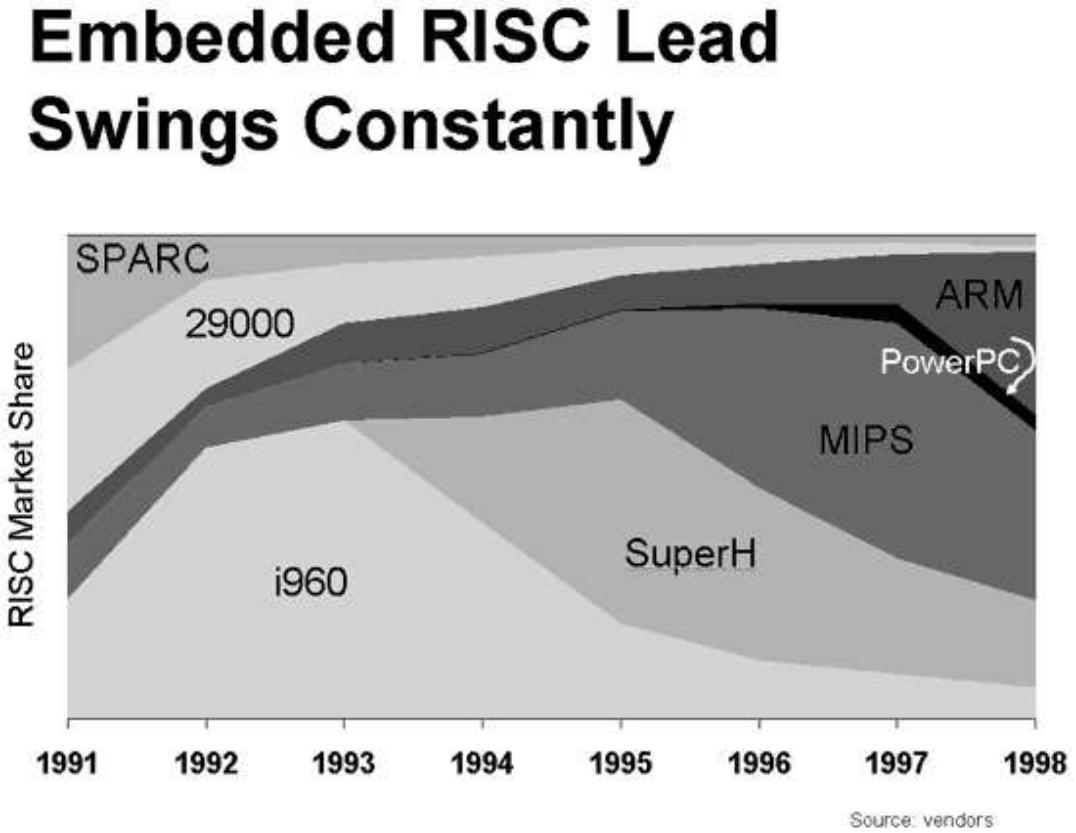
Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion



- Les applications sont plus variées que pour les ordinateurs
- Beaucoup de processeurs embarqués sont des processeurs de bureau qui n'ont pas percés (MIPS, 68K, SPARC, ARM, PowerPC)

- Processeurs embarqués

Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

Architecture des processeurs

Architecture "Von Neuman" ou "Princeton"

- Processeurs embarqués

Introduction

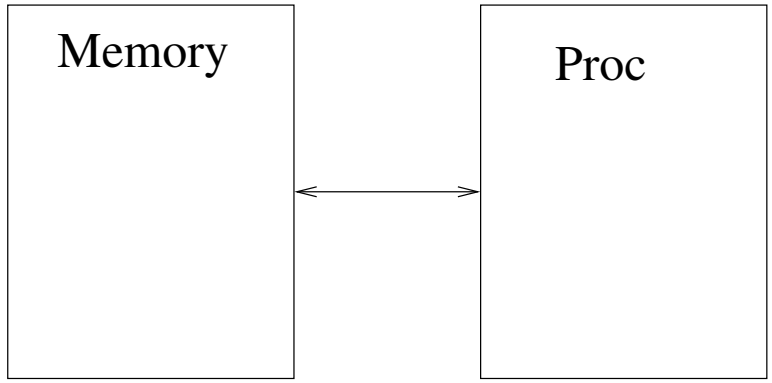
Architecture des processeurs

- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- La mémoire contient les données et les instructions
- L'unité centrale (CPU) charge les instructions depuis la mémoire.
- Un ensemble de registres aide le CPU:
  - Compteur d'instructions (Program counter: PC),
  - Registre d'instruction (Instruction register: IR)
  - Pointeur de pile (stack pointer: SP)
  - Registres à usage général (Accumulateur: A)



Architecture Harvard

- Processeurs embarqués

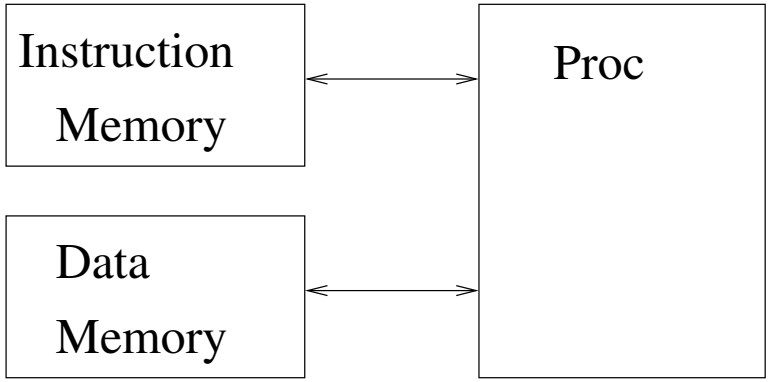
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Données et instructions dans des mémoires séparées
- Autorise deux accès simultanés à la mémoire.
- Utilisé pour la plupart des DSP
  - meilleure bande passante
  - Performances plus prédictibles



Le jeu d'instruction

- Processeurs embarqués

Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Le *jeu d'instruction* (Instruction Set Architecture: ISA) a une importance capitale
  - Il détermine les instructions élémentaires exécutées par le CPU.
  - C'est un équilibre entre la complexité matérielle du CPU et la facilité d'exprimer les actions requises
  - On le représente de manière symbolique (ex: ARM, code sur 32 bits):

```
        LDR r0,[r8] ; commentaire
lab:    ADD r4,r0,r1 ;
```

- Deux classes de jeux d'instructions:
  - CISC: Complex Instruction Set Computer
  - RISC: Reduce Instruction Set Computer

CISC: Complex Instruction Set Computer

- Processeurs embarqués

Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Une instruction peut designer plusieurs opérations élémentaires.
  Ex: un load, une opération arithmétique et un store,
  Ex: calculer une interpolation linéaire de plusieurs valeurs en mémoire.
- Accélération par des mécanismes matériels complexes
- Grandes variation de taille et de temps d'exécution pour les instructions
- Résulte en un code compact mais complexe à générer.
- Vax, Motorola 68000, Intel x86/Pentium

Exemple: instructions de l'ISA du Pentium

- Processeurs embarqués

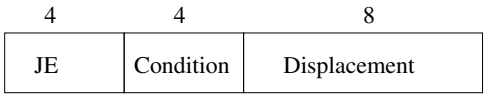
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

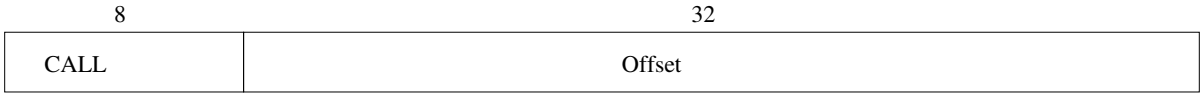
Différents types de processeurs embarqués

Compilation pour processeurs

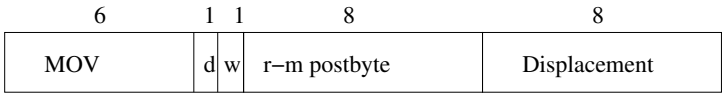
JE EIP + displacement



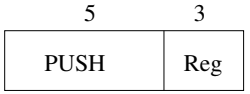
Call



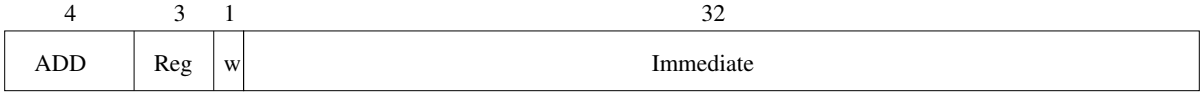
Mov $EBX, [EDI+displacement]



Push ESI



Add $EAX, Immediate



Test $EDX, Immediate

RISC: Reduced Instruction Set Computer

- Processeurs embarqués

Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Petites instructions simples, toutes de même taille, ayant toutes (presque) le même temps d'exécution
- Pas d'instruction complexe
- Accélération en pipelinant l'exécution (entre 3 et 7 étages de pipeline pour une instruction) $\Rightarrow$ augmentation de la vitesse d'horloge
- Code plus simple à générer, mais moins compact
- Tous les microprocesseurs modernes utilisent ce paradigme: SPARC, MIPS, ARM, PowerPC, etc.

Exemple: instructions de l'ISA du MIPS

- Processeurs embarqués

Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

I TYPE (Immediate)

	5	5	
op	rs	rt	Immediate

J TYPE (Jump)

6	26
op	Immediate

R TYPE (Register)

op	rs	rt	rd	sa	funct

- I-Type:
  LW rt, offset(base)
- J-Type:
  JUMP target
- R-Type:
  ADD rd,rt,rs

Le CPU

- Processeurs embarqués

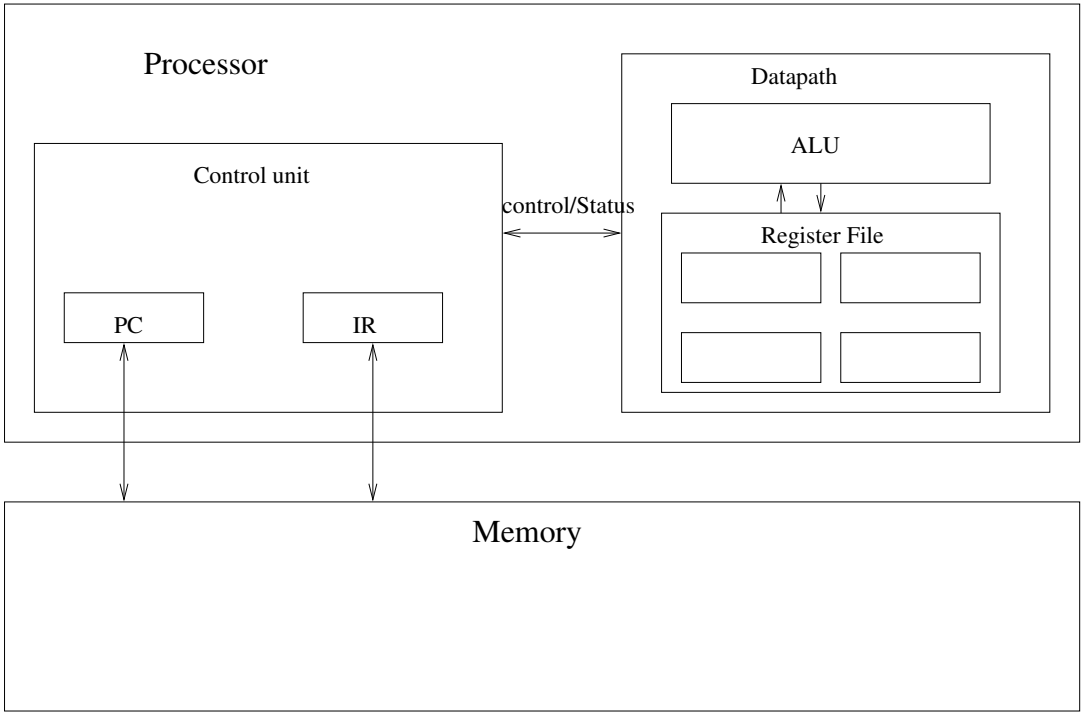
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- L'unité de contrôle configure le chemin de donnée suivant l'instruction à exécuter.
- L'exécution d'une instruction est décomposée en plusieurs phases d'un cycle.



Le pipeline RISC: exemple du MIPS

- Processeurs embarqués

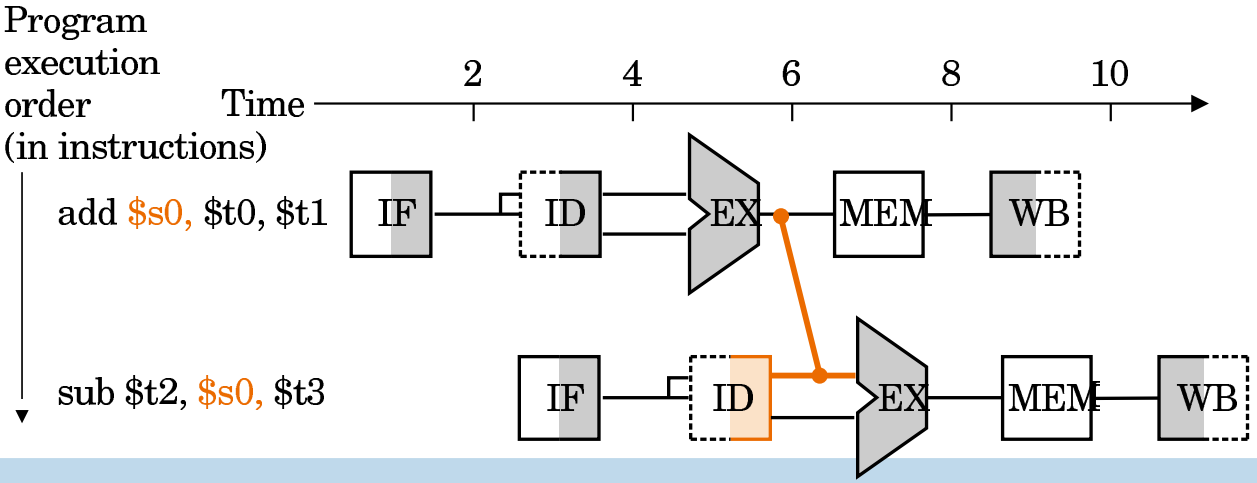
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Le pipeline dépend de l'architecture, pour le MIPS:
  - Instruction Fetch (IF, Fetch): charge l'instruction dans l'IR
  - Instruction Decode (ID, Decode): décode l'instruction et met en place le contrôle du chemin de donnée
  - Execute (Ex): exécute le calcul dans le chemin de donnée.
  - Memory access (Mem): accède la mémoire
  - Write Back (WB): écrit dans le banc de registre



Le pipeline RISC: exemple du MIPS

- Processeurs embarqués

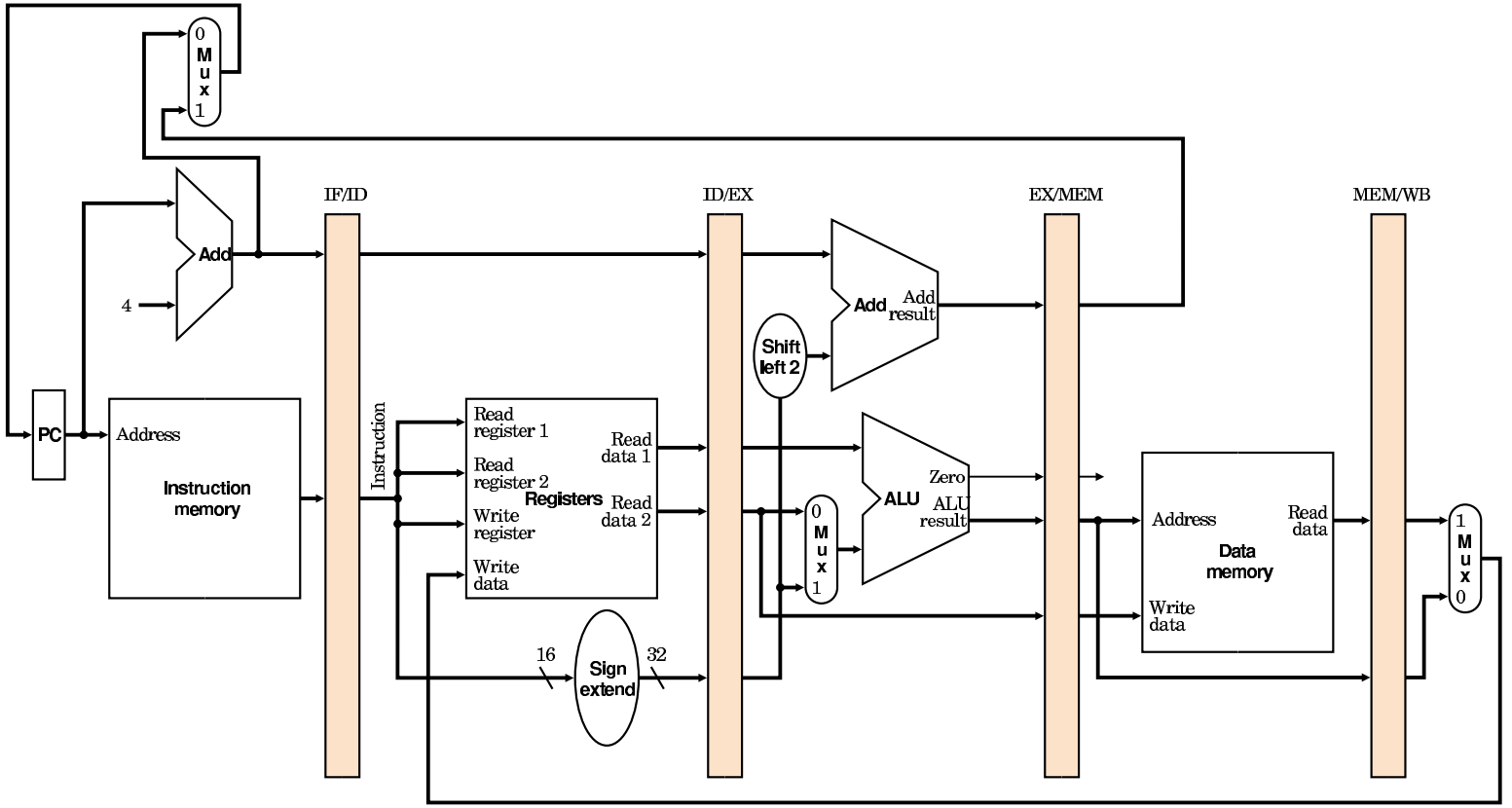
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Physiquement, l'architecture du processeur est organisée en calculs combinatoires pour chaque étape de pipeline, séparés par des registres.



Le pipeline RISC: exemple du MIPS

- Processeurs embarqués

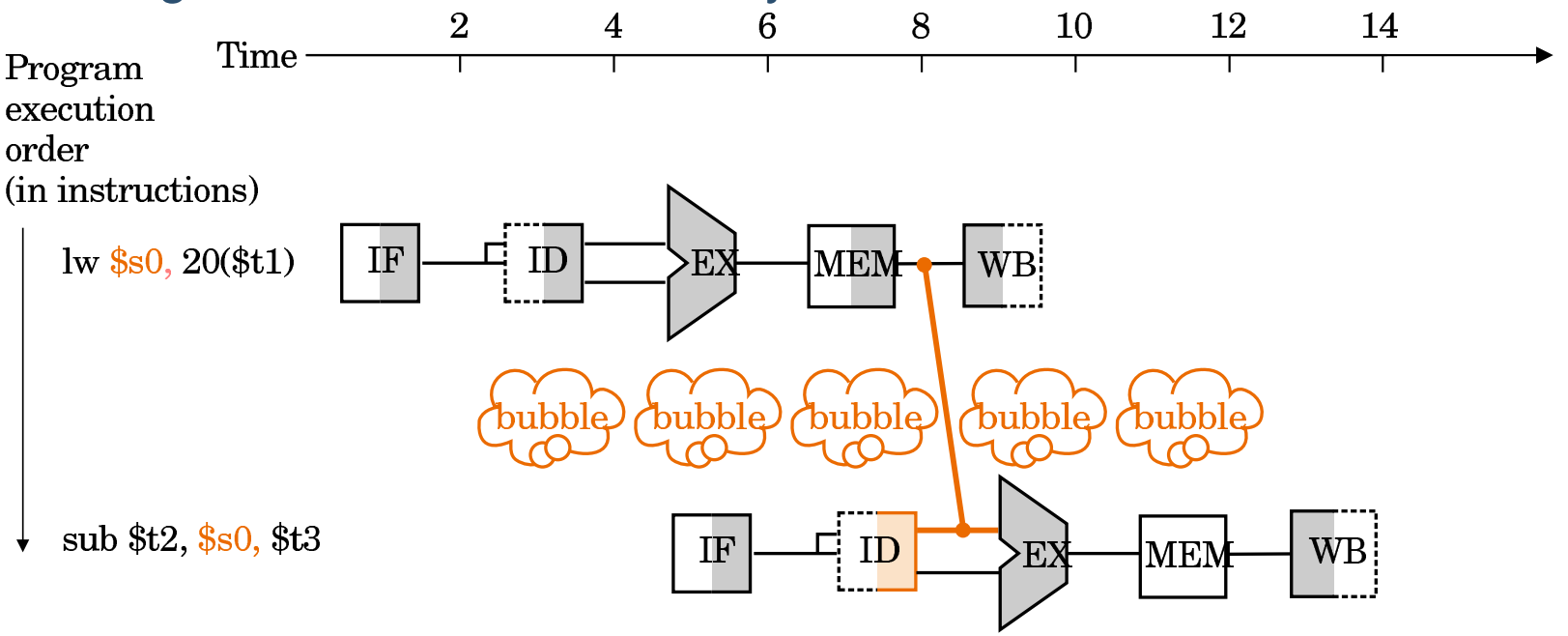
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Lorsque l'instruction suivante ne peut pas être exécutée tout de suite, cela crée une "bulle".
- Par exemple une addition utilisant un registre qui vient d'être chargé doit être retardé d'un cycle.



Exemple d'exécution sans pipeline

- Processeurs embarqués

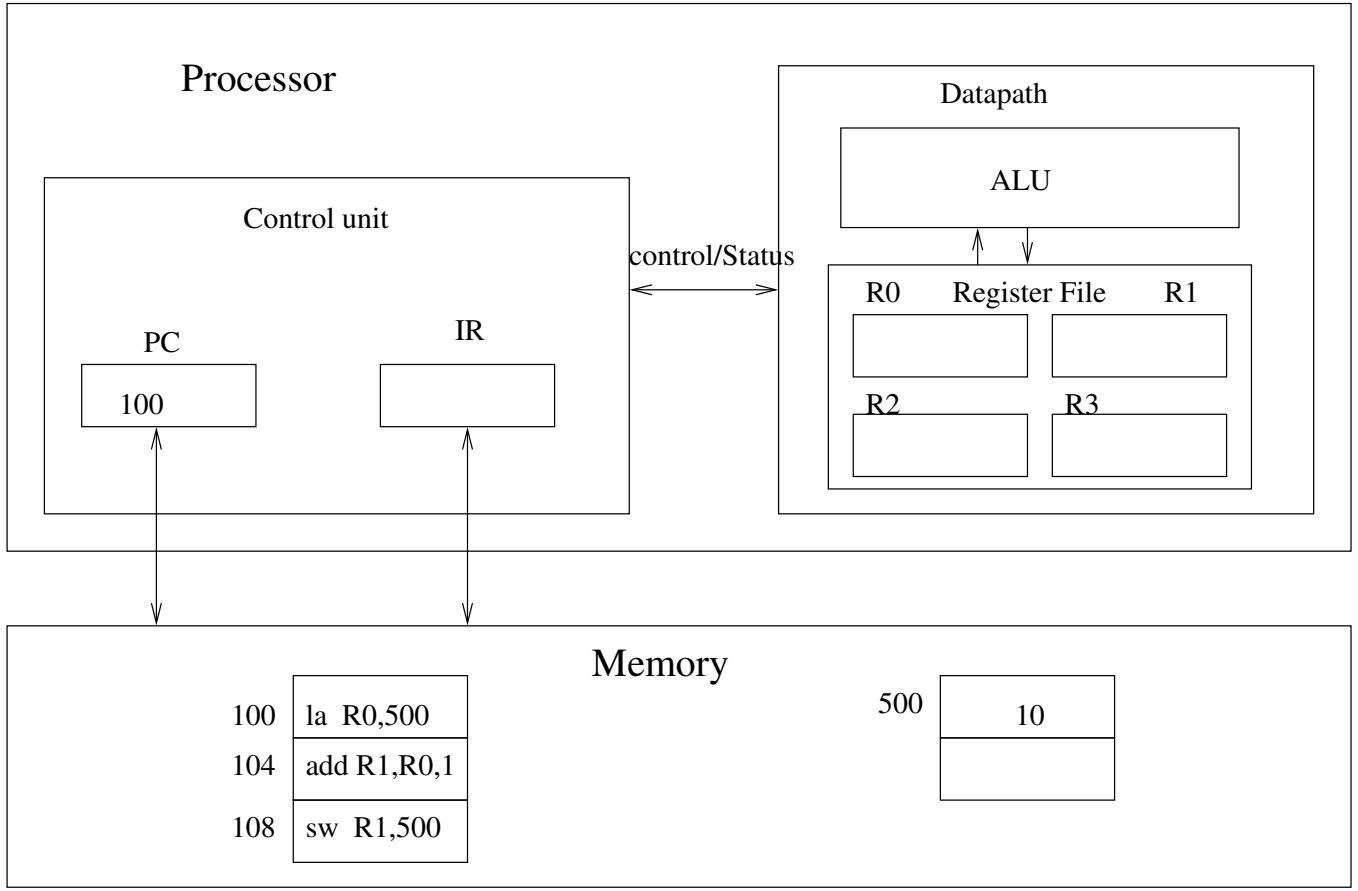
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Avant le début de l'exécution de l'instruction à l'adresse 100



cycle 1

- Processeurs embarqués

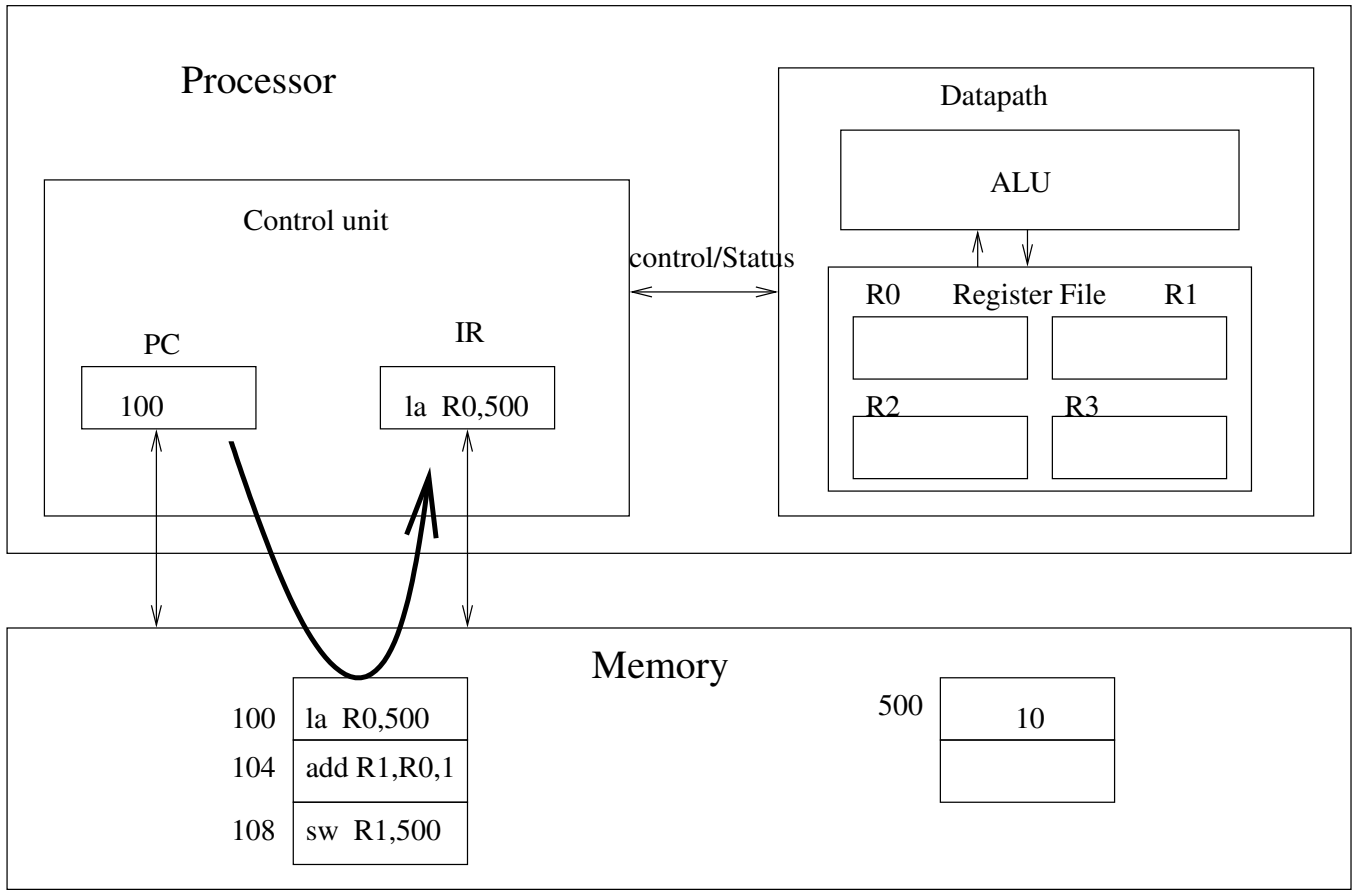
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

Chargement de l'instruction



cycle 2

- Processeurs embarqués

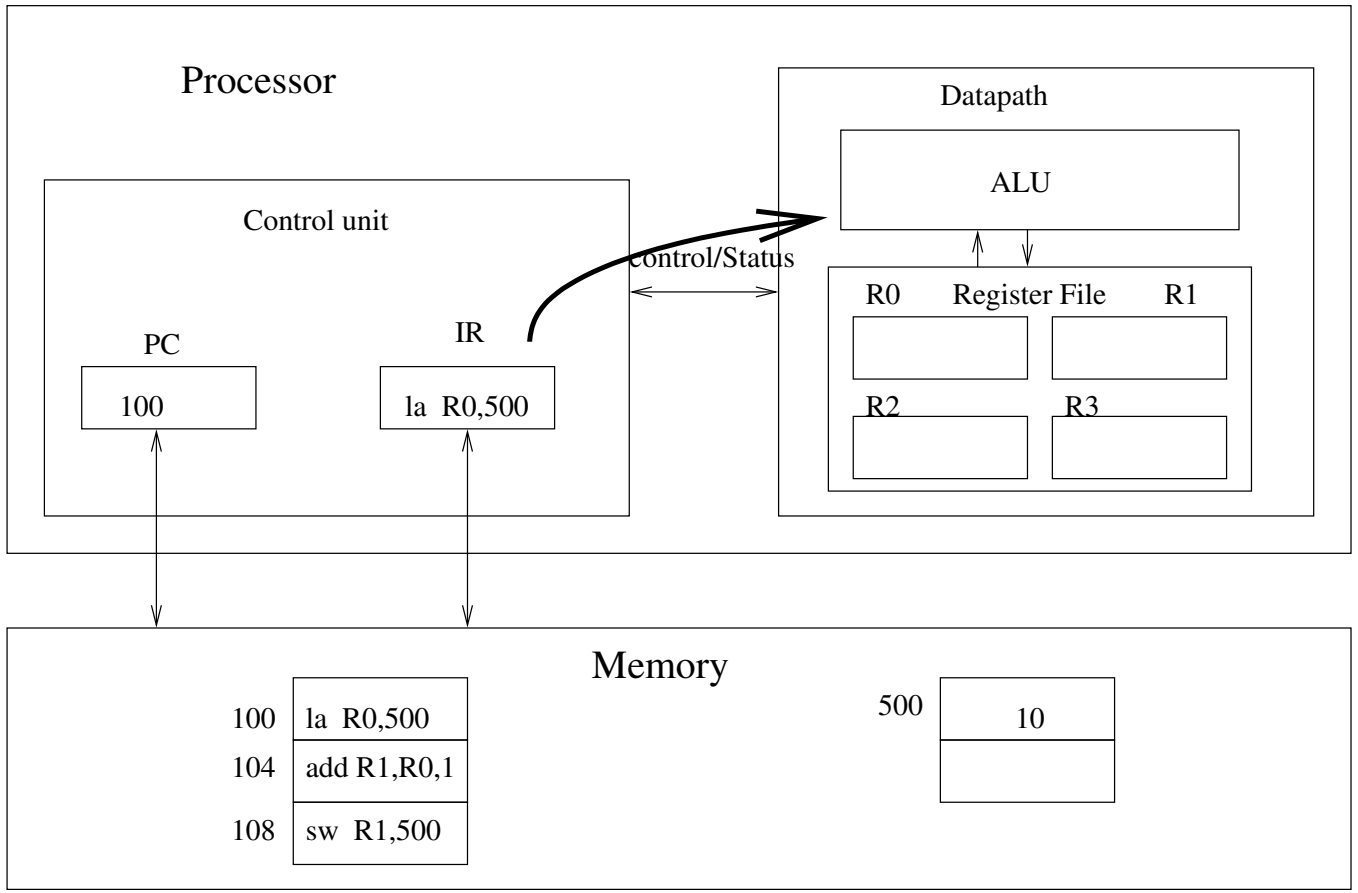
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

Décodage de l'instruction



cycle 3

- Processeurs embarqués

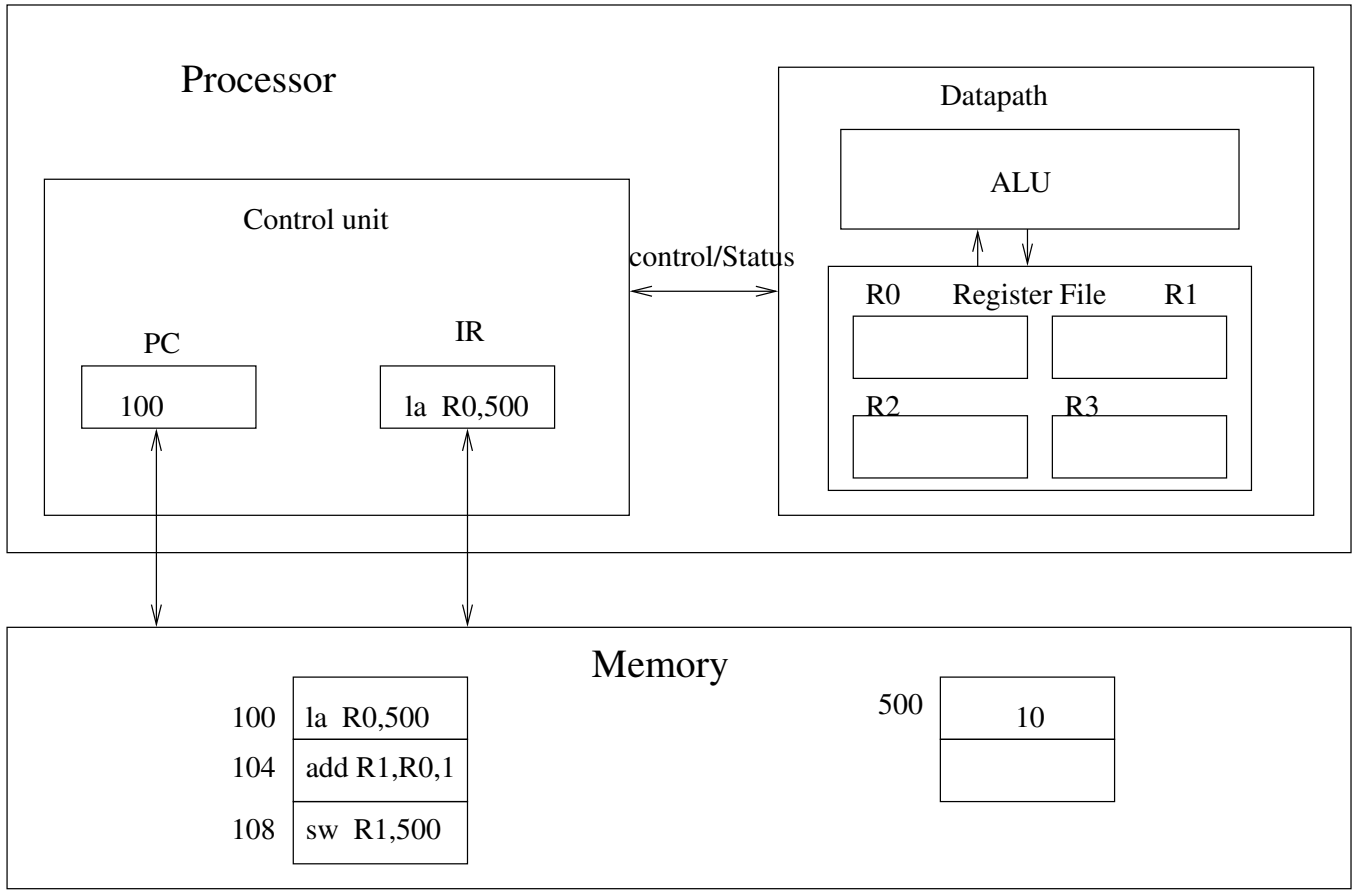
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

Exécution (rien pour load)



cycle 4

- Processeurs embarqués

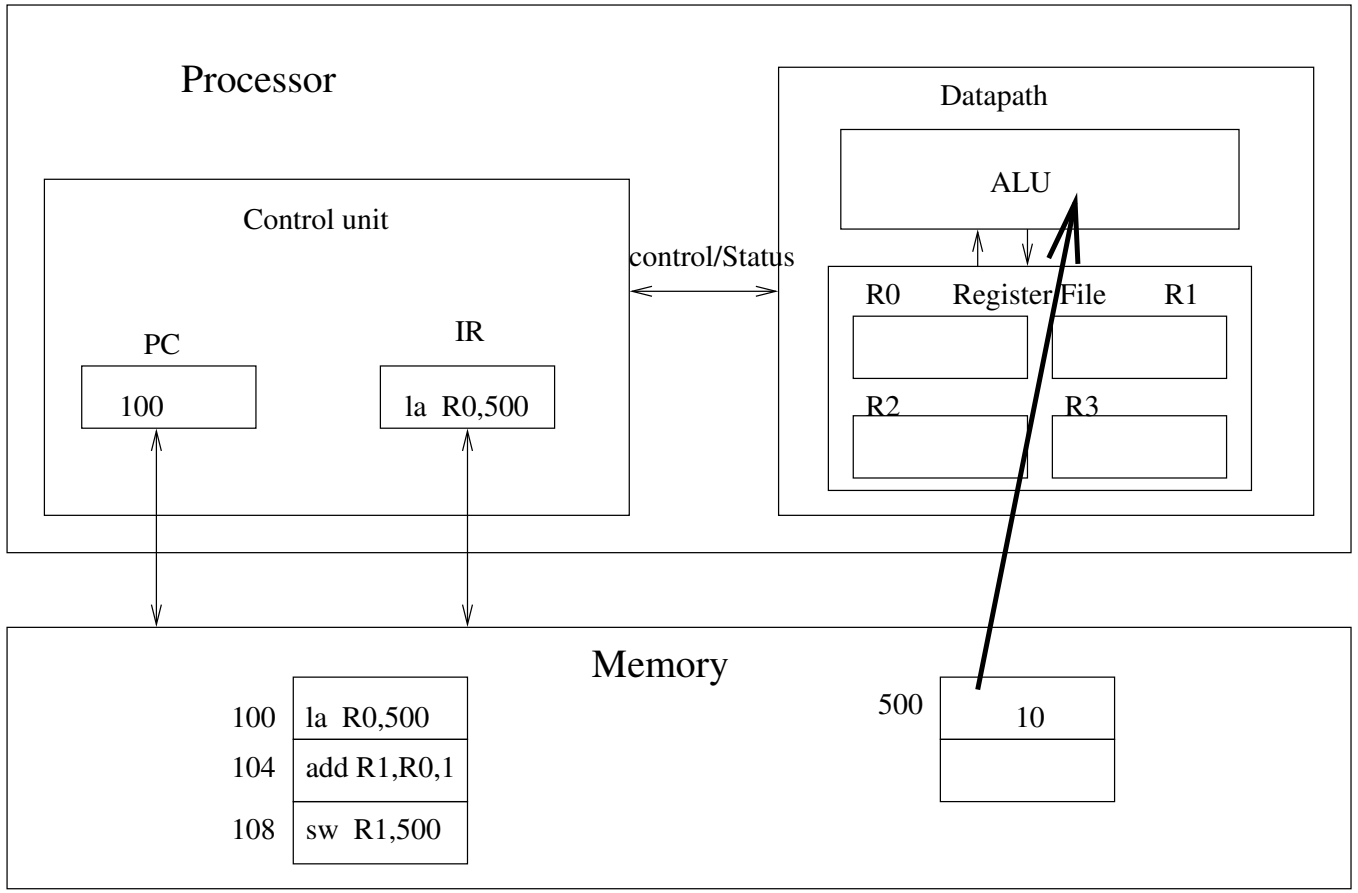
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

Accès mémoire



cycle 5

- Processeurs embarqués

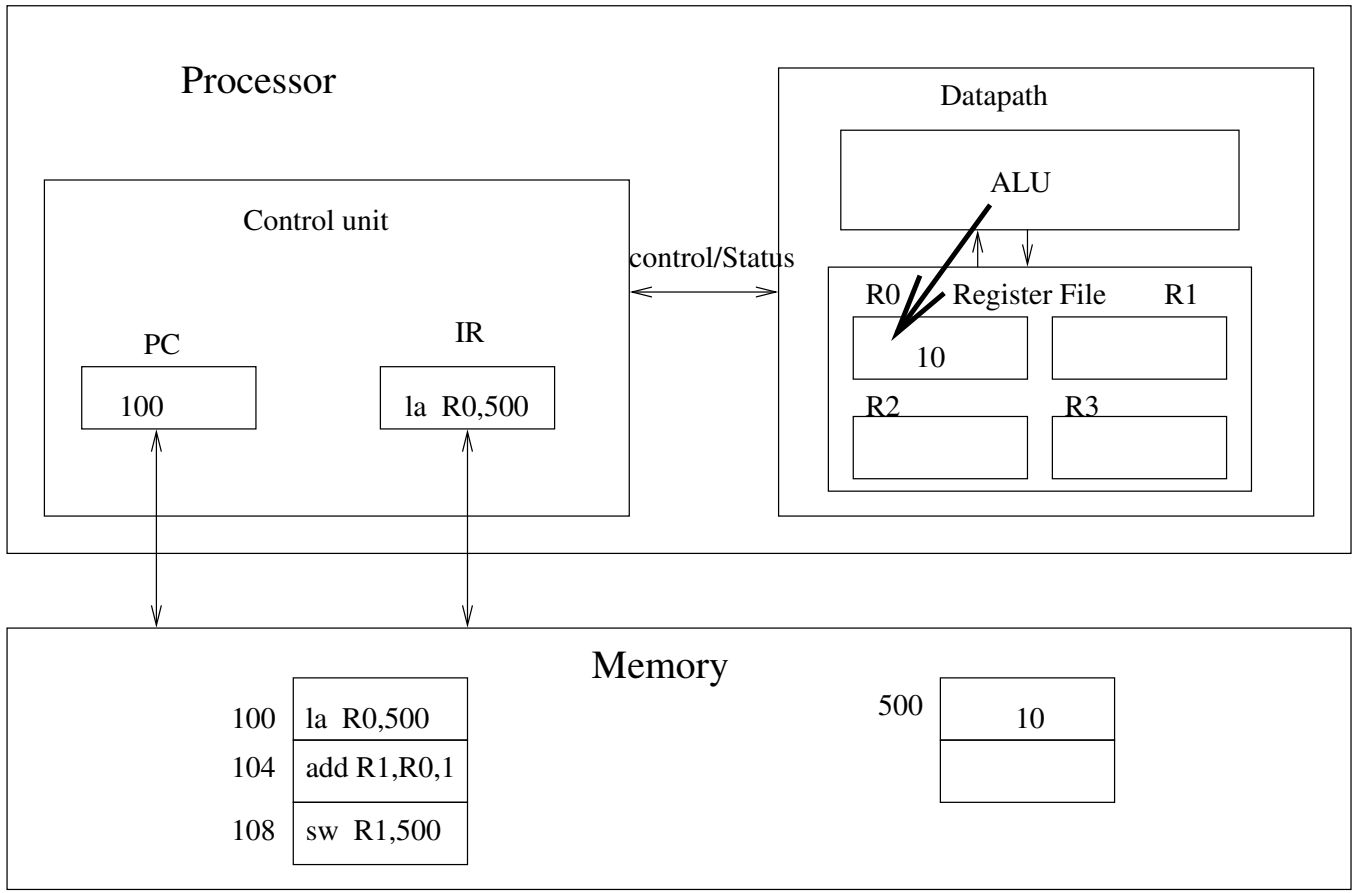
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

Write Back



Bilan architecture pipelinée

- Exécution non pipelinée:
  - 5 cycles pour exécuter une instruction
  - $\Rightarrow$ 15 cycles pour 3 instructions.

- Processeurs embarqués

Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

Exemple d'exécution avec pipeline

- Processeurs embarqués

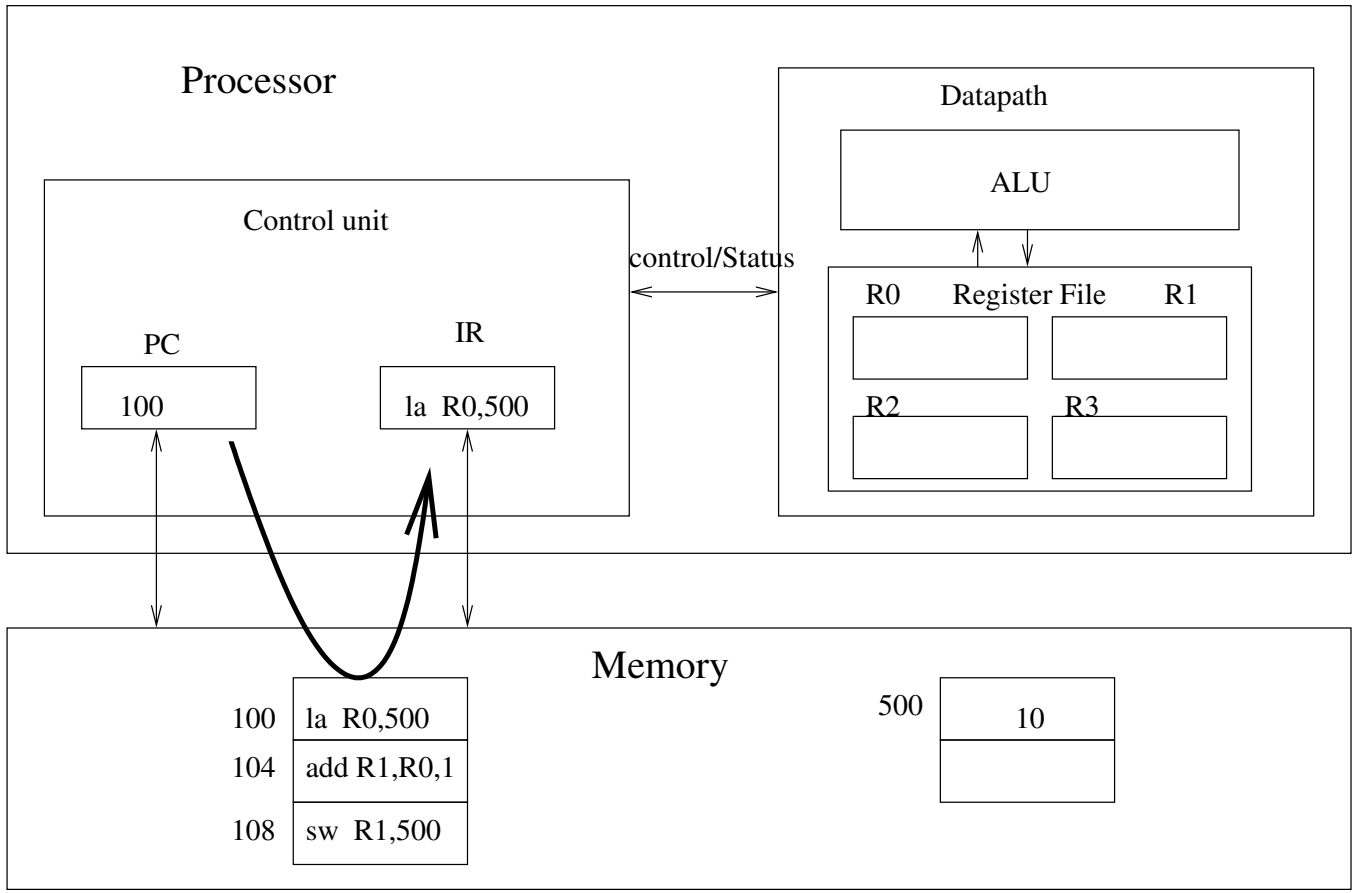
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

Chargement de l'instruction load



cycle 2

- Processeurs embarqués

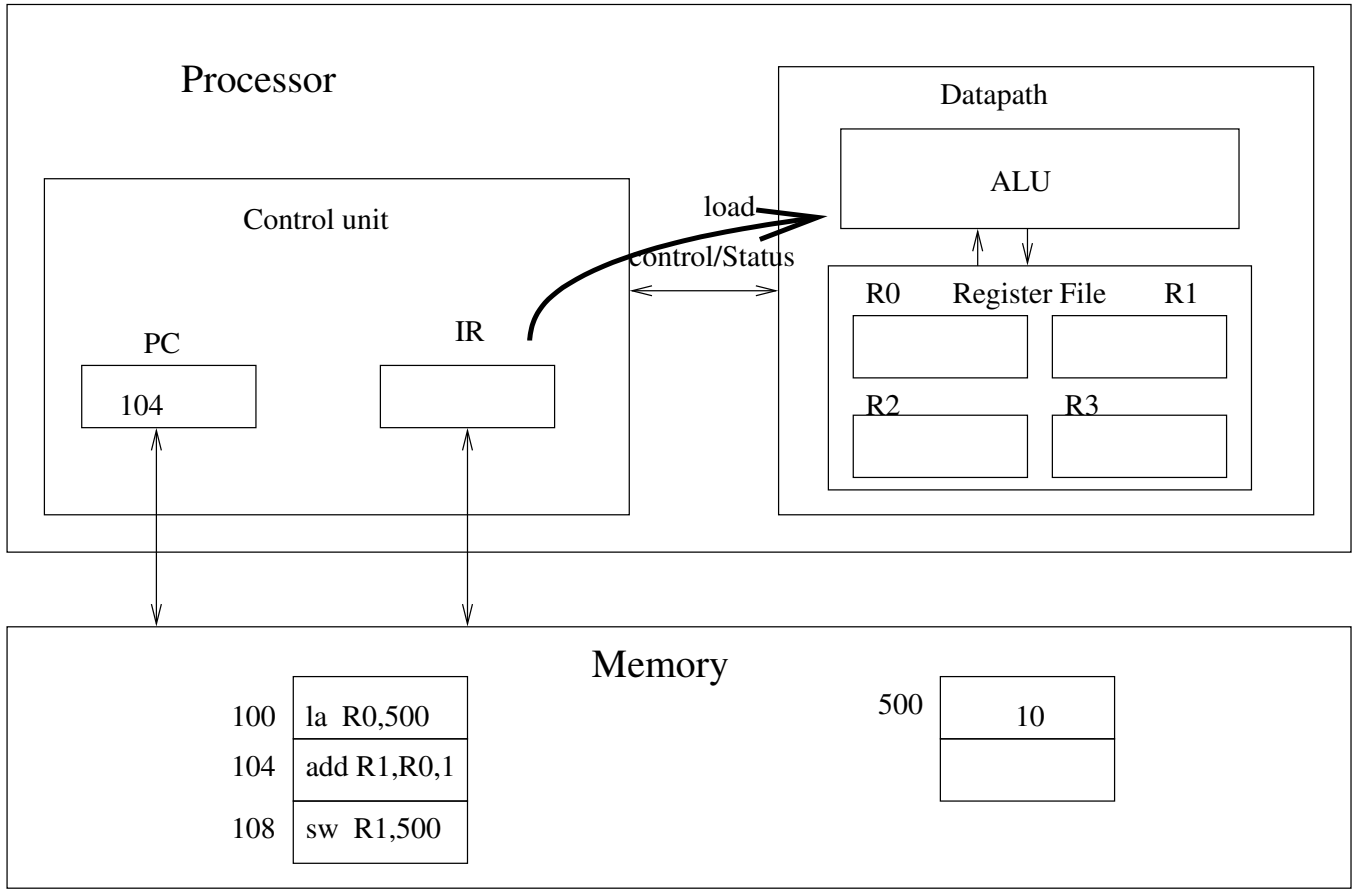
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Décodage de l'instruction load
- et Chargement de Rien (bulle car l'instruction suivante retardée)



cycle 3

- Processeurs embarqués

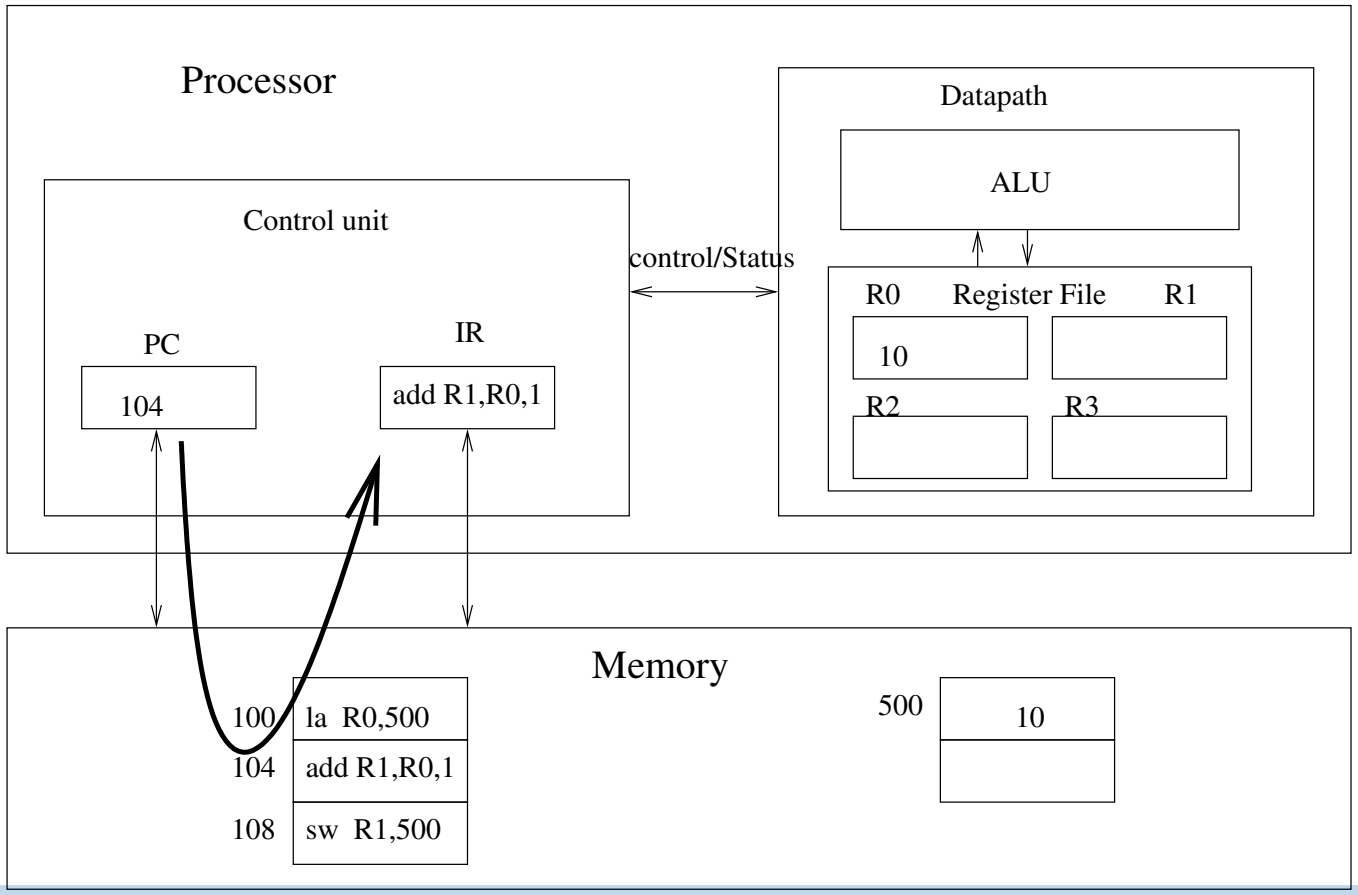
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Exécution de l'instruction load (rien)
- Décodage de rien
- Chargement de l'instruction add



cycle 4

- Processeurs embarqués

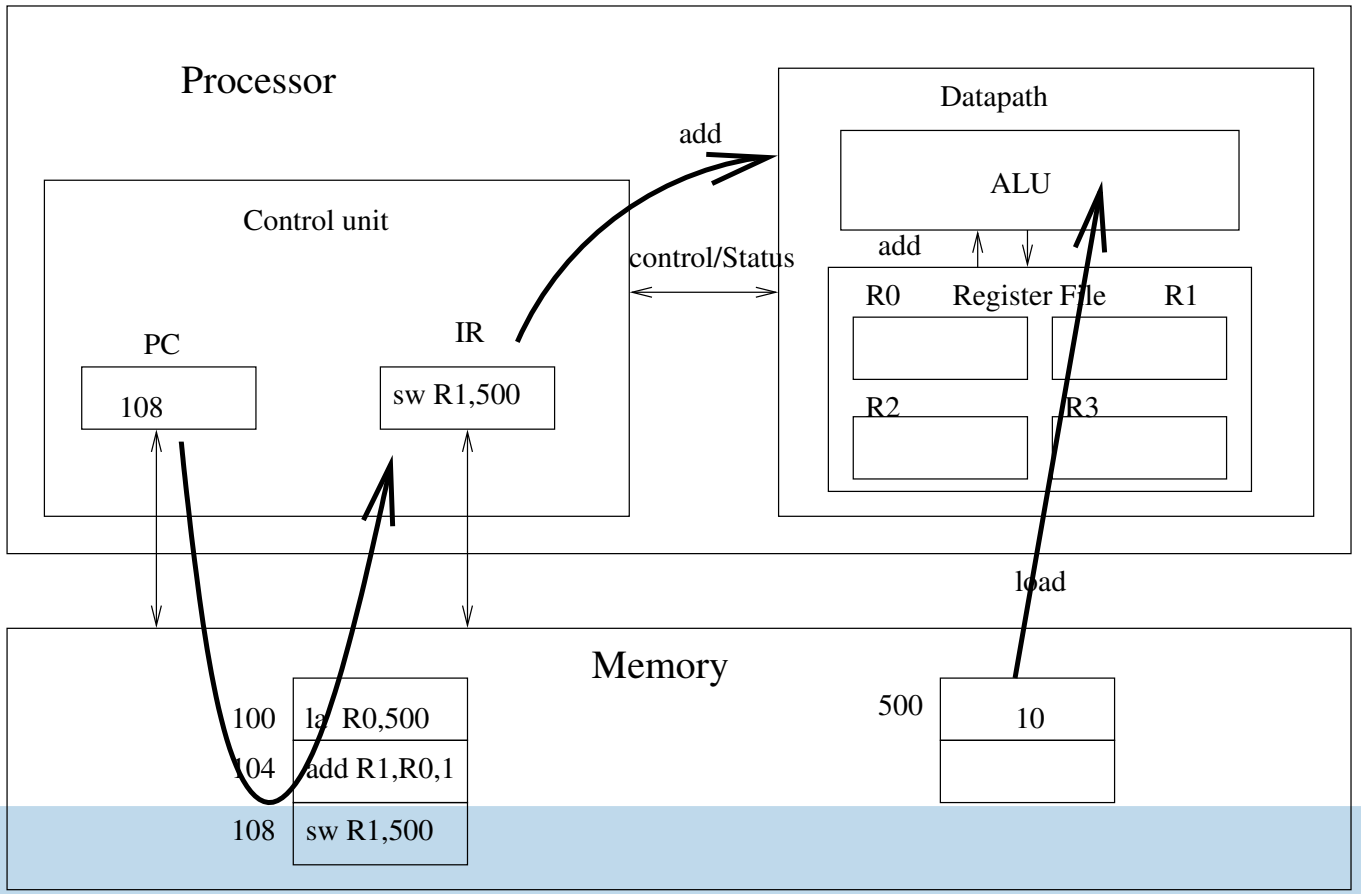
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Accès mémoire de l'instruction load
- Exécution de rien
- Décodage de l'instruction add
- Chargement de l'instruction store



cycle 5

- Processeurs embarqués

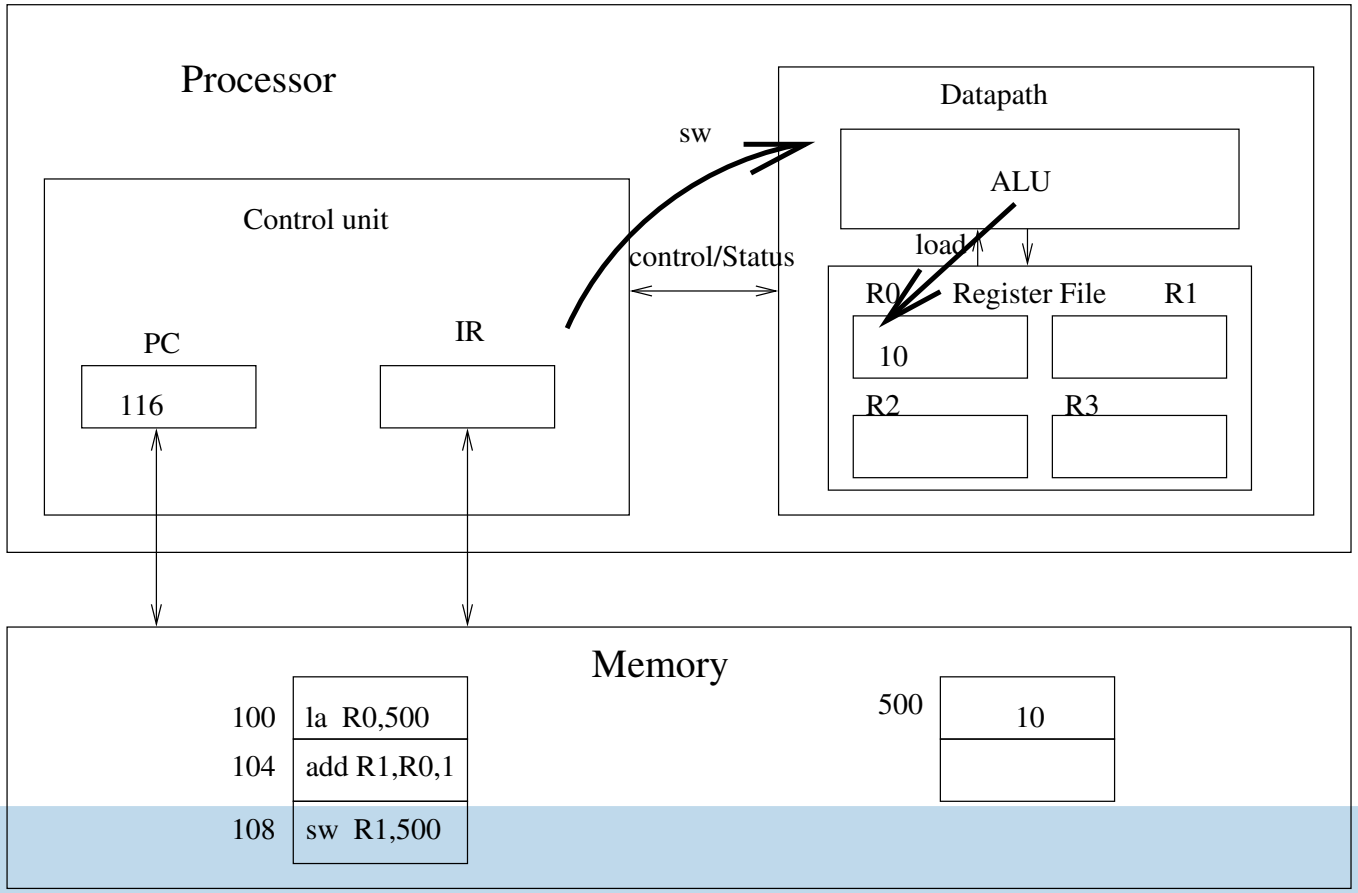
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Write Back de l'instruction load
- Accès mémoire de rien
- Exécution de l'instruction add (bypass)
- Décodage de l'instruction store



cycle 6

- Processeurs embarqués

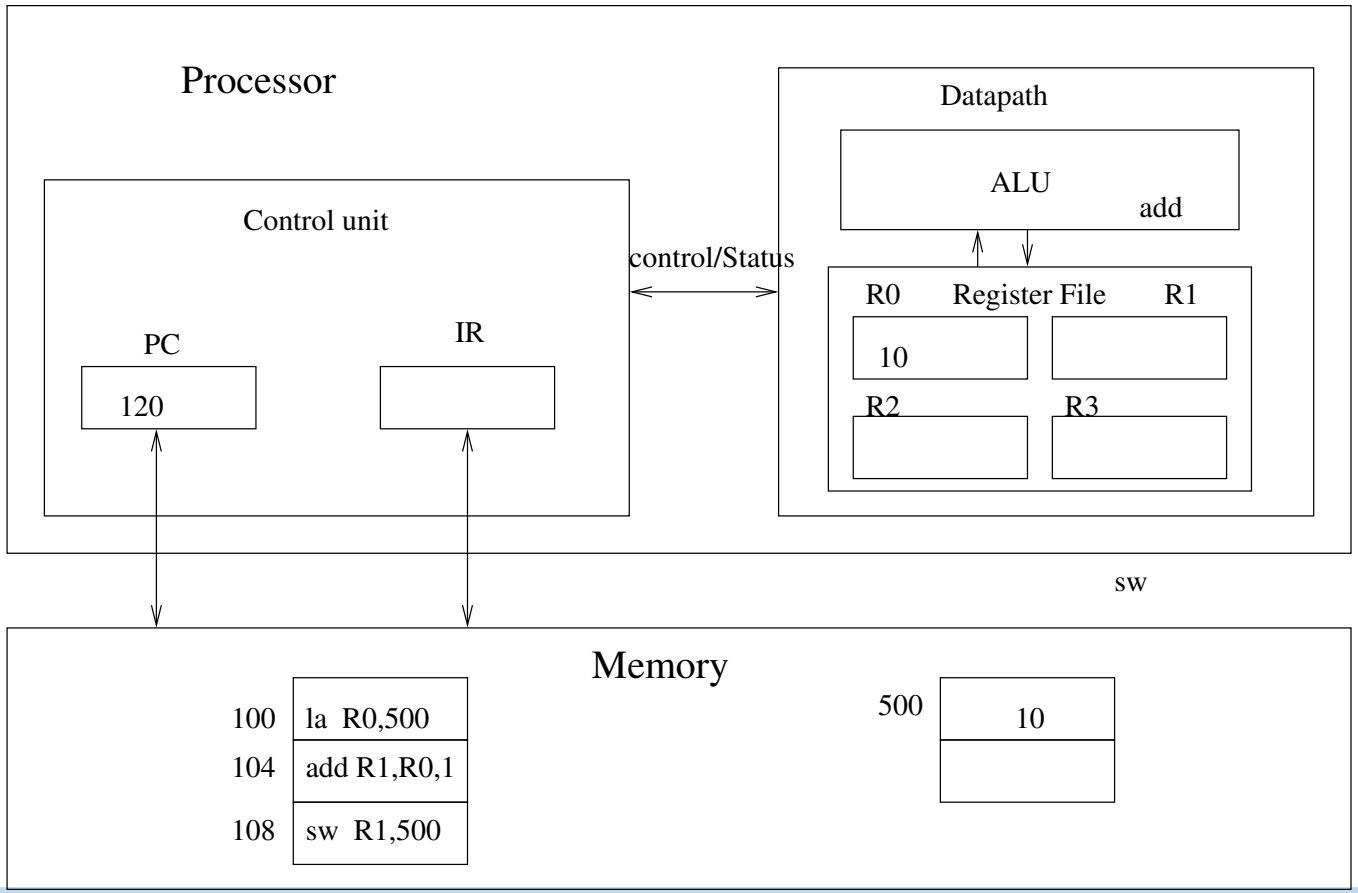
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Write Back de rien
- Accès mémoire de l'instruction add (rien)
- Exécution de l'instruction store (rien)



cycle 7

- Processeurs embarqués

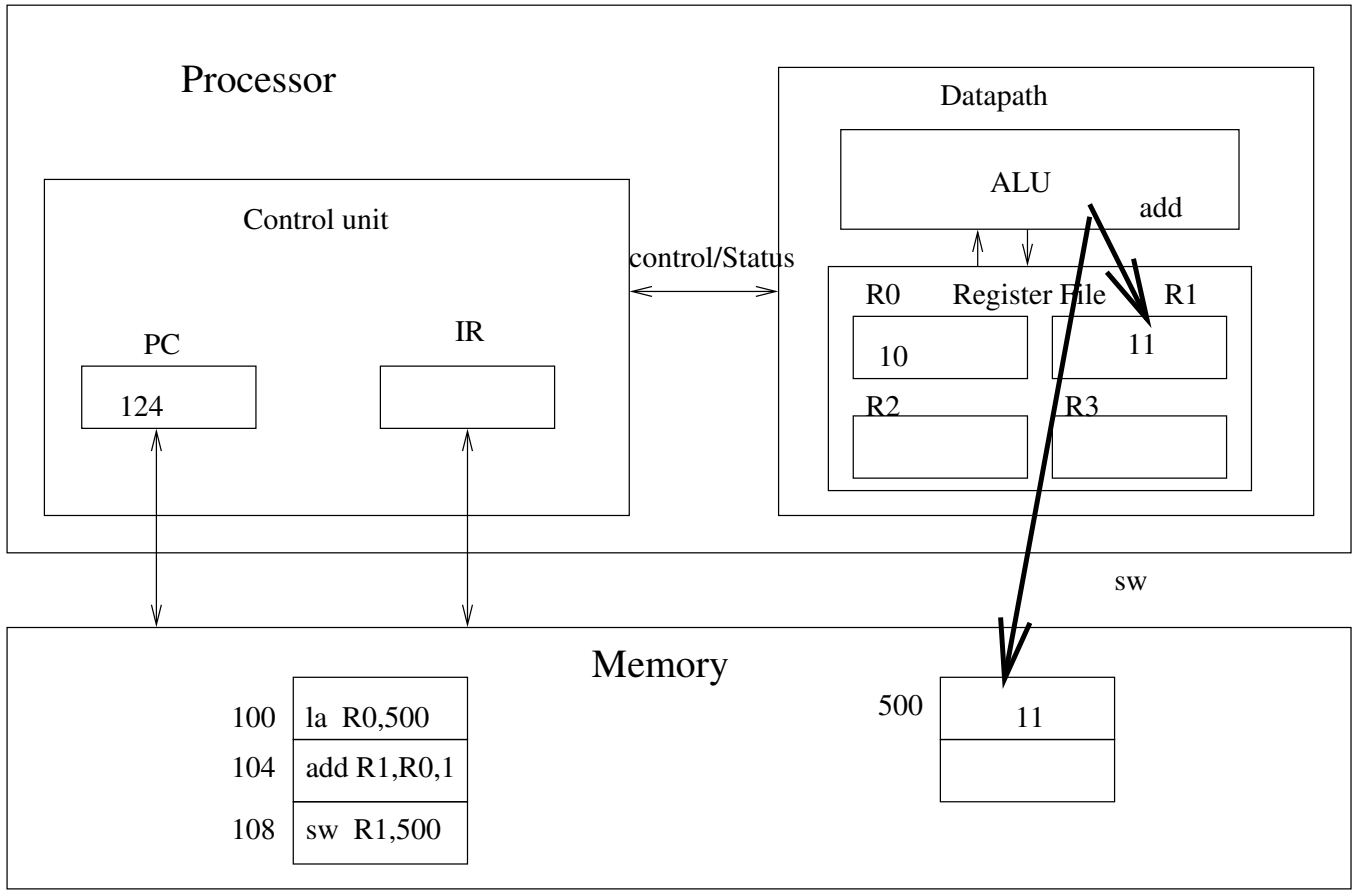
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Write Back de l'instruction add
- Accès mémoire de l'instruction store (bypass)



Bilan architecture non pipelinée

- Processeurs embarqués

Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Exécution non pipelinée:
  - 5 cycles pour exécuter une instruction
  - $\Rightarrow$ 15 cycles pour 3 instructions.
- Exécution pipelinée:
  - 5 cycles pour exécuter une instruction
  - 8 cycles pour 3 instructions.
  - $\Rightarrow$ sans branchement, une instruction par cycle
  - Un branchement (conditionnel ou pas) interrompt le pipeline car il faut attendre de décoder l'adresse de branchement pour charger l'instruction suivante $\Rightarrow$ quelques cycles d'inactivité (pipeline stall)
  - Lors d'un branchement, certain ISA autorisent l'utilisation de ces *delai slots*: une ou deux instructions après le branchement sont exécutées, que le branchement soit pris ou pas (comme si elles étaint écrites avant le branchement).

Parallélisme au sein du processeur

- Processeurs embarqués

Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

Indépendamment du pipeline, Deux paradigmes dominants:

- Super Scalaire
  - Duplication des unités,
  - Répartition au vol des instructions sur les unités disponibles (re-ordonnancement des instructions: *out of order execution*)
  - Exemple: le PowerPC 970 (4 ALU, 2 FPU)
  - Efficace mais complexifie l'unité de contrôle (problème des interruptions)
- Very Large Instruction Word (VLIW)
  - Duplication des unités,
  - L'ordonnancement des instructions est fixé à la compilation (tout se passe comme si les instructions pouvait être regroupe sur 64 bits, 128 bits etc.)
  - Inventé par Josh Fisher (Yale) à partir du trace scheduling
  - Les processeurs VLIW sont tous basés sur les architecures RISC, avec entre 4 et 8 unités.
  - Exemple: TriMedia (Philips), Itanium IA64 (Intel).

Parallélisme au sein du processeur

- Processeurs embarqués

Introduction

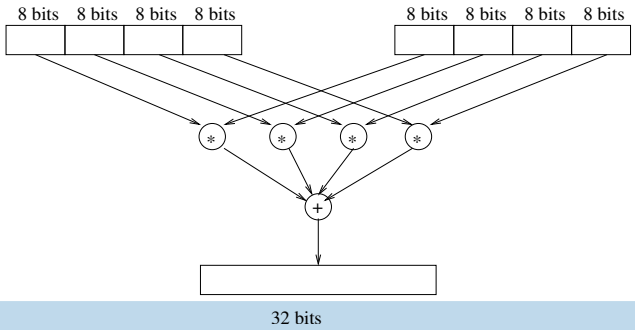
Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

Une autre approche possible: instructions SIMD.

- Modification du data-path pour proposer des opérations parallèles sur 16 ou 8 bits
- Exemple: Sun Visual Instruction Set, Intel Pentium MMX, Philips TriMedia
- Gains importants sur certains traitements mais très peu utilisé en pratique (difficile à inférer par le compilateur)
  - Librairies écrites en assembleur (programmes non portables)
  - Fonction C représentant les instructions assembleurs (*compiler instrisic*)
  - Exemple: instruction `ifir8ii R1, R2, R3` du Trimedia:



Mémoire

- Processeurs embarqués

Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

- Plusieurs technologies pour les mémoires:
  - Mémoires statiques (SRAM): petites, rapides, consommatrices, peu denses (chères).
  - Mémoires dynamiques (DRAM): grandes, lentes, très denses, transactions chères
- De plus en plus de place On-Chip pour la mémoire (dans ce cas elles sont moins efficaces que les chips mémoire).
- Ne pas oublier que le code aussi réside en mémoire
- Tous les systèmes ont des caches pour cacher les temps de latence lors de l'accès à la mémoire, en général plusieurs niveaux de caches: hiérarchie mémoire.

Principe du Cache

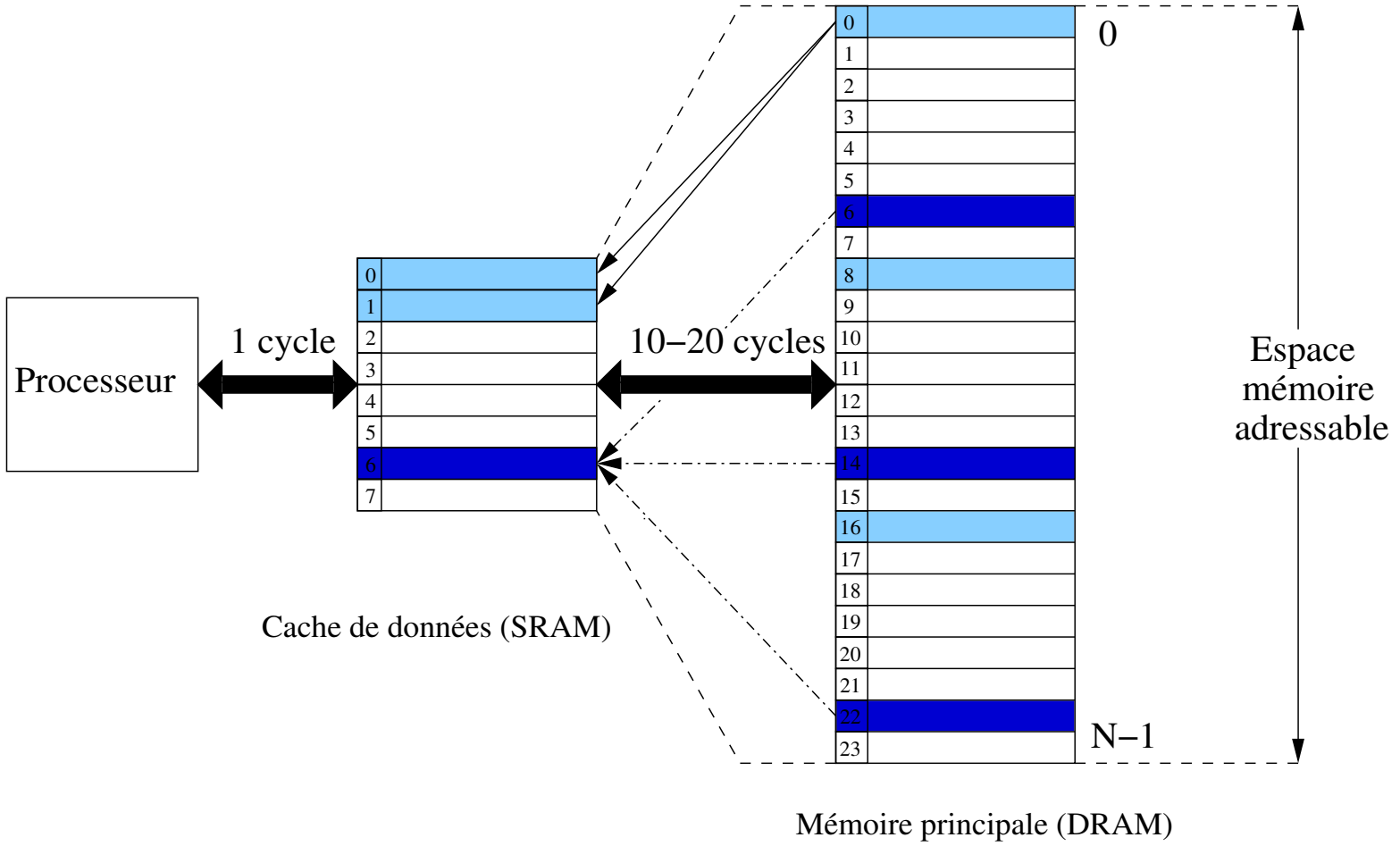
- Processeurs embarqués

Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs



Hiérarchie Mémoire

- Processeurs embarqués

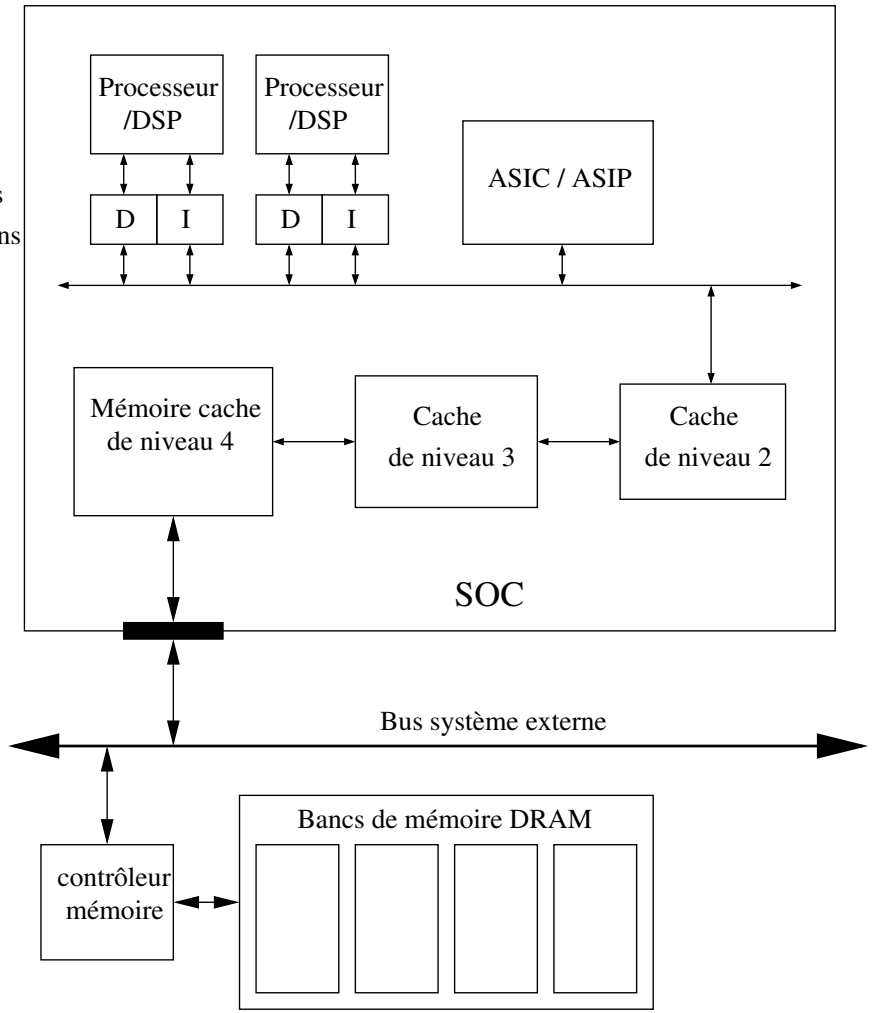
Introduction

Architecture des processeurs
- Architecture "Von Neuman" ou "Princeton"
- Architecture Harvard
- Le jeu d'instruction
- CISC: Complex Instruction Set Computer
- Exemple: instructions de l'ISA du Pentium
- RISC: Reduced Instruction Set Computer
- Exemple: instructions de l'ISA du MIPS
- Le CPU
- Le pipeline RISC: exemple du MIPS
- Exemple d'exécution sans pipeline
- Exemple d'exécution avec pipeline
- Parallélisme au sein du processeur
- Parallélisme au sein du processeur
- Mémoire

Différents types de processeurs embarqués

Compilation pour processeurs

D: cache de données
I: cache d'instructions



- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

Différents types de processeurs embarqués

Différents types de processeurs embarqués

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- Beaucoup de Processeurs à usage général ayant une ou deux générations
- 4, 8, 16 ou 32 bits (taille des mots)
- RISC et CISC
- DSP: Digital Signal Processor
- ASIP: Application Specific Integrated Processor

68000, x86

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- Famille des Motorola 68000
  - Un des plus vieux processeur embarqué (ex Sun, Mac)
  - Architecture CISC
  - ISA propre et les meilleurs outils de développement, beaucoup d'utilisateurs
- Famille des x86
  - Démarre au 8086 (Intel) puis 80286, 386, 486, Pentium, et Athlon (AMD)
  - En processeurs embarqués: 5 fois moins que MIPS, ARM ou 68000.
  - architecture CISC, compatible avec le code du 8086
  - compatibilité mais mauvaises performances

SPARC, 29000 et i960

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués
- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- SPARC
  - Un des premier RISC à avoir été embarqué (pratiquement plus aujourd'hui)
  - SPARC est une architecture brevetée (soft core, Intellectuel Property: IP), plusieurs compagnies fabriquent des SPARC
- 29000 (AMD)
  - Le 29000 a eu beaucoup de succès (imprimante laser Apple) grâce à ces 192 registres
  - AMD a arrêté la production car le développement des outils coûtait trop cher.
- i960 (intel)
  - Le i960 a été le plus vendu des processeurs embarqués au milieu des années 90 (router réseau et HP Laserjet).

MIPS, ARM, SuperH et PowerPC

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- MIPS (microprocessor without interlocked pipeline stages)
  - Originellement pour les stations puissantes (SGI)
  - Puis, marché des consoles de jeux (Nitendo N64)
  - Famille très étendue: du plus gros (MIPS 20Kc, 64 bit) au plus petit (SmartMIPS, 32 bit pour carte à puce)
- ARM (Advanced RISC Machines, ex Acorn)
  - Un des 32 bits embarqués les plus populaires : téléphones portables
  - Faible consommation
  - Le successeur: StrongArm est commercialisé par Intel sous le nom de XScale
- SuperH (ou SH: Hitachi) Utilisé dans les stationsxs Sega et les PDA
- PowerPC autant utilisé en embarqué qu'en ordinateur

Et les autres....

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués
- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- Plus de 100 processeurs embarqués 32 bits sur le marché
- Les constructeurs de FPGA proposent des soft-processeurs pour configurer les FPGA: Nios (Altera), MicroBlaze (Xilinx)
- Certain processeurs RISC (Crusoe de Transmetta) peuvent exécuter du code CISC (Intel)
  - Principe: recompilation du code à l'exécution (*runtime compilation*)
  - Gain obtenus par un mécanisme de cache, d'optimisation poussée des portion de code répétées (boucle), et grâce au parallélisme de niveau instruction
  - Réduction drastique de la consommation pour des performances équivalentes

Micro-contrôleurs

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- Utilisé pour le contrôle embarqué
  - Censeur, contrôleurs simples
  - Manipule des événements, quelques données mais en faible quantité
  - Exemple: camescope, disque dur, appareil photo numérique, machine à laver, four à micro-onde
- Quelques caractéristiques fréquentes
  - Périphériques présents sur le circuit (timer, convertisseur analogique numérique, interface de communication), accessible directement grâce aux registres
  - Programme et données intégrées au circuit
  - Accès direct du programmeur à de nombreuses broches du circuit
  - Instructions spécialisées pour les manipulation de bits.

DSP: Digital Signal Processing

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- Utilisés pour les applications de traitement du signal
  - Grande quantités de données numérisées, souvent organisées en flux
  - Filtre numérique sur téléphone, TV numérique, synthétiseur de sons
- Relativement proche des GPP, mais quelques caractéristiques en plus:
  - Bande passante élevée (deux bus)
  - Instructions dédiées pour les calculs de traitement du signal: multiplication accumulation,
  - Arithmétique spécifique (mode d'arrondi)
  - Registres dédiés pour certains opérateurs.
  - Constructeurs: Texas Instrument, puis Analog Devices, Motorola

Quelques mécanismes matériels utiles

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- Densité de code:
  - La taille du code est importante pour les codes embarqués car elle influe sur la taille de la mémoire utilisée
  - Un programme C compilé pour SPARC prendra deux fois plus de place en mémoire que le même programme compilé pour le 68030.
  - En général les code RISC sont deux fois moins dense que les codes CISC (ex: instruction `TBLS` du 68300: *table lookup and interpolate*)
  - les options de compilation doivent être utilisée avec précaution.
  - Le code est quelquefois stocké compressé et decompressé au vol par du matériel spécifique.

Quelques mécanismes matériels utiles

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués
- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- Manipulations au niveau bit:
  - Utilisé pour les algorithmes de cryptage mais surtout pour les pilotes de périphériques.
  - La plupart des périphériques indiquent leur état au processeur en mettant un certain bit à 1 dans un certain registre.
  - Un processeur RISC standard doit rapatrier le mot de 32 bit, masquer et tester à 0
  - L'instruction `BTST` (bit test) du Motorola 68000 permet de faire tout cela en une instruction

Quelques mécanismes matériels utiles

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- Données non-alignés
  - De nombreux traitements manipulent des données de taille non-multiple de 32 (paquets TCP/IP, video streams, clés d'encryption, 20 bits, 56 bits)
  - Les processeurs RISC savent uniquement transférer des mots (32 bits) *alignés* (calés sur une adresse multiple de 32 bits).
  - La plupart des architectures CISC (68k, x86) peuvent faire des chargements non alignés

Quelques mécanismes matériels utiles

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués
- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- Gestion spécifique du cache
  - Les caches améliorent les performances mais introduisent du non-déterminisme.
  - Les contraintes spécifiques des systèmes embarqués ont entraîné des mécanismes particuliers pour les cache
  - On peut vouloir bloquer le cache (cache locking): forcer certaines données ou instruction à se charger et rester dans le cache (on parle aussi de mémoire *scratch-pad memory* ou de *software controlled cache*).
  - La plupart des caches utilisent une politique de Write-Back: une donnée modifiée dans le cache n'est pas forcément immédiatement recopiée en memoire. Dans le cas de périphériques mappés en mémoire, il est indispensable de recopier immédiatement (politique *write-through*)

Quelques mots sur la consommation

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- Trois composantes de la consommation d'une porte logique (inverseur)
  - Consommation dynamique : $P_{dyn} = C.V_{CC}^2$
    (C capacité de la porte)
  - Consommation statique : $P_{static} = V_{CC}.I_{leak}$
    (V_{CC}: tension d'alimentation, I_{leak} intensité des courants de fuite)
  - Consommation de court-circuit $P_{cs} = K.\tau.(V_{CC} - 2V_{Th})^3$.
    (K:constante technologique ; V_{Th}:tension seuil ; τ:temps de montée descente du signal)
- Aujourd'hui (2004) $P_{dyn} \gg P_{static} \gg P_{cs}$
- Demain (2006) $P_{dyn} \approx P_{static} \gg P_{cs}$

Consommation d'un circuit CMOS

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- Généralisation naïve en prenant en compte une activité moyenne α (nombre moyen de portes commutant)
  - Consommation dynamique : $P_{dyn} = C.V_{CC}^2.\alpha.f$
    (f: fréquence du circuit)
  - Consommation statique : $P_{static} = V_{CC}.I_{leak}.N.k_{design}$
    (N: nombre de portes, k_{design} constante dépendant du design)
- Cette modélisation est très imprécise pour un circuit dont le comportement n'est pas stationnaire (ex: processeur)

Réduction statique de la consommation

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- Le facteur le plus important est la tension d'alimentation (V_{CC}) d'abord 3.3 V puis 2.5 V. Les versions récentes de Xscale (strong ARM, Intel) et les puces smartCard fonctionnent a 0.65 V
- On peut différentier les tensions en fonction du bloc du chip: 1.5 V pour le processeur, 3.3 pour l'interface du bus et les pattes d'entrée/sortie (ex: Strong ARM de Digital)
- Plus la technologie est intégrée, moins elle consomme (capacité diminuée).
- Fréquence d'horloge peu élevée compensée par le parallélisme
- Complexité réduite des différents composants (moins de registres, architectures RISC)

Réduction dynamique de la consommation

- Processeurs embarqués

Introduction

Architecture des processeurs

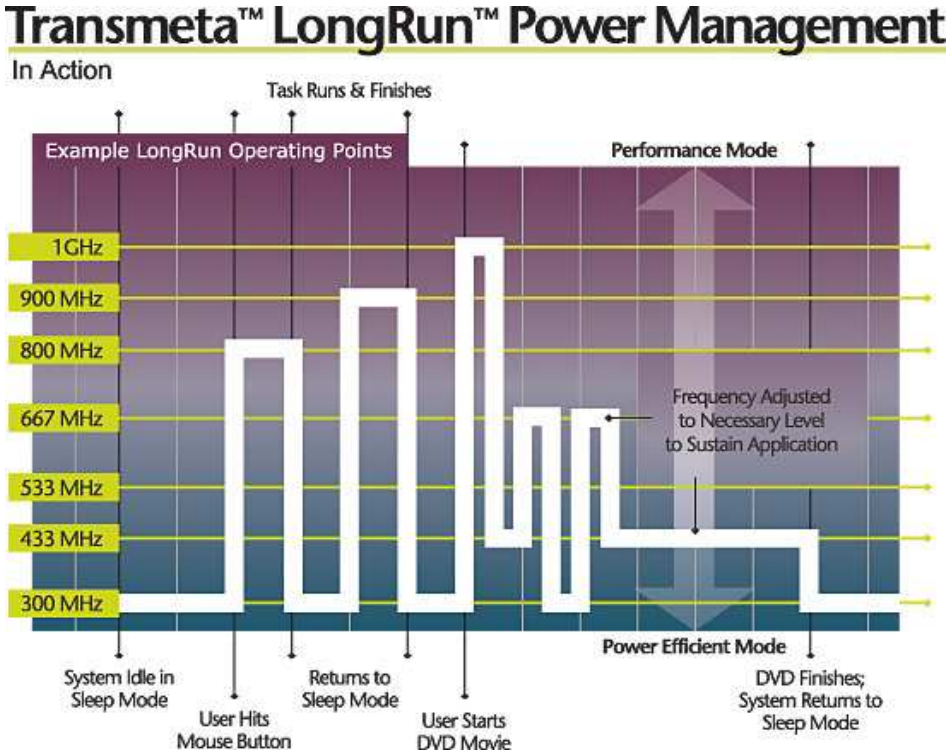
Différents types de processeurs embarqués
- Différents types de processeurs embarqués
- 68000, x86
- SPARC, 29000 et i960
- MIPS, ARM, SuperH et PowerPC
- Et les autres....
- Micro-contrôleurs
- DSP: Digital Signal Processing
- Quelques mécanismes matériels utiles
- Quelques mots sur la consommation

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

Conclusion

- Gestion dynamique de la fréquence d'horloge
- Exemple: processeur Crusoe (Transmetta)
  Suppression de l'horloge sur un bloc (*Dynamic clock gating*)
- Gestion dynamique de l'alimentation (pas encore réalisé)



- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

Compilation pour processeurs embarqués

Compilation: biblio

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Dragon book:
  - Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. "Compilers: Principles, Techniques and Tools." Addison-Wesley, 1988.
  - La bible, un peu dépassé sur certains sujets, mais beaucoup n'ont pas changé depuis.
- Cooper/Torczon:
  - Keith D. Cooper and Linda Torczon. Engineering a Compiler. Morgan-Kaufmann, 2003.
  - récent, survol assez complet.

Compilation: Principes généraux

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Définition:
  - Un compilateur est un programme qui prend en entrée un programme exécutable et produit en sortie un autre programme exécutable.
- Les principes fondamentaux de la compilation sont:
  - Le compilateur doit conserver le sens du programme compilé
  - Le compilateur doit améliorer le code
- Les propriétés importantes d'un compilateur sont:
  1. Code produit efficace (rapidité, mémoire).
  2. Informations retournées en cas d'erreurs, debboging
  3. Rapidité de la compilation
- Pour un système embarqué les points 1 et 3 sont différents

Compilation: Contraintes supplémentaires

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Définition d'un "bon" compilateur
  - Contraintes temps-réel
    - Plutôt résolu par langages dédié (langages synchrone, e.g. Esterel) et des OS spécialisés que par le compilateur.
  - Consommation
    - Organisation mémoire
    - accès mémoire (cache miss)
    - Taille de code
  - Performances
- Temps de compilation peu critique
- Architecture variées: DSP, instructions multi-média etc.
- Relation avec le système d'exploitation peu standardisé
  - Actuellement: petit OS ou pas d'OS
  - Prochainement: OS léger pour multi-threads

Compilation: Le flot général

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

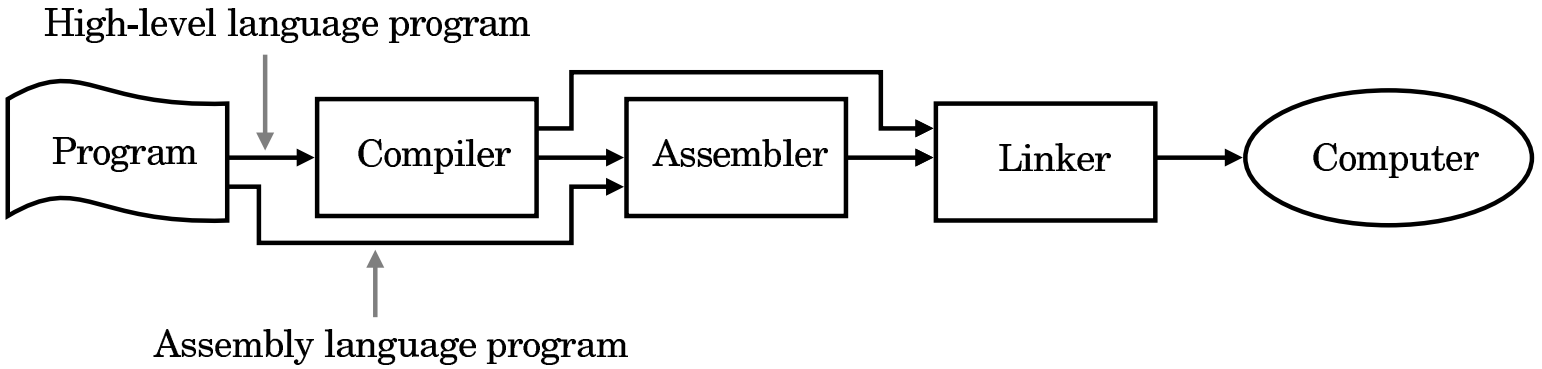
Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Le flot complet de compilation est le suivant:



- La programmation d'un système embarqué nécessite souvent d'écrire explicitement des parties en assembleur (pilotes de périphériques et d'accélérateurs matériels).

Compilation: Le flot détaillé

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

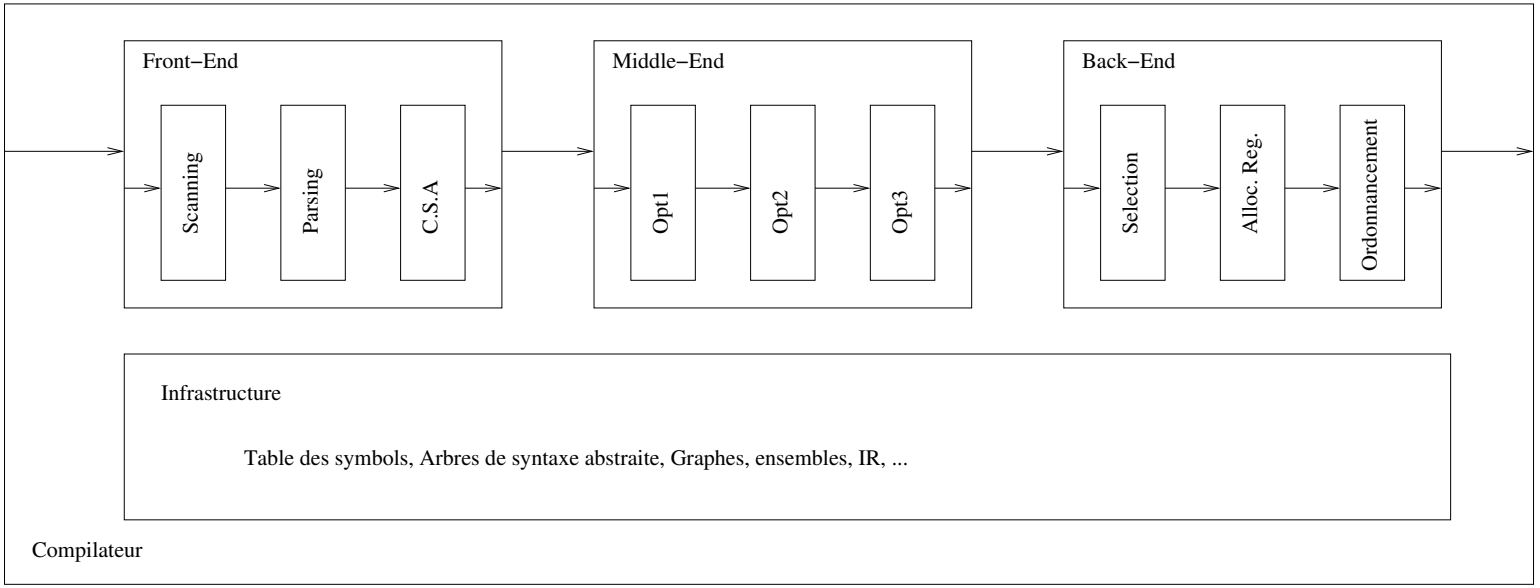
Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- le flot détaillé est le suivant:



Compilation: Représentation intermédiaires (1)

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

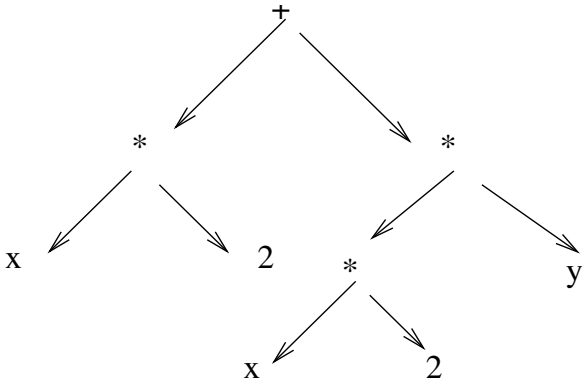
Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

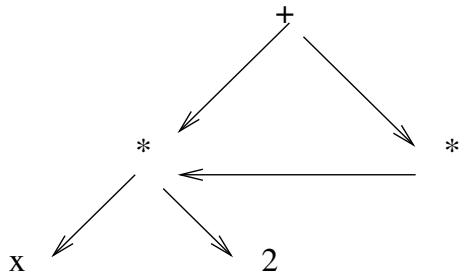
Exemple de l'appareil photo numérique

Conclusion

- Arbre: Abstract Syntax Tree (AST), Graphe acyclique (DAG)
- ex: $x \times 2 + x \times 2 \times y$



AST



DAG

Compilation: Représentation intermédiaires (2)

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

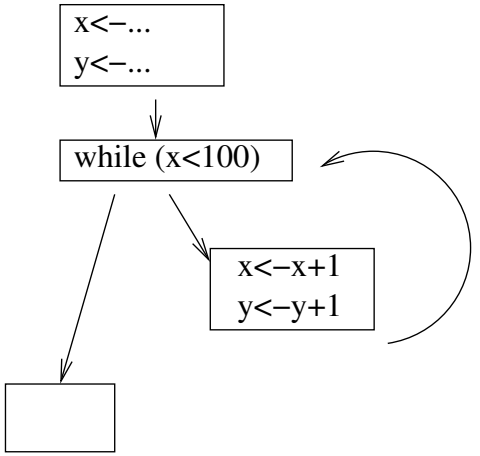
- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- CDFG: Control and Data-flow graph

$x \leftarrow \ldots$
$y \leftarrow \ldots$
while $(x < 100)$
$\quad x \leftarrow x + 1$
$\quad y \leftarrow y + x$
end



- Représentation intermédiaire linéaire (proche de l'assembleur). Pour $x \times 2 + x \times 2 \times y$:

t_1	$\leftarrow$	2	$loadI$	2	$\Rightarrow$	t_1
t_2	$\leftarrow$	x	$loadAI$	$\$fp, @x$	$\Rightarrow$	t_2
t_3	$\leftarrow$	$t_1 \times t_2$	$mult$	t_2, t_1	$\Rightarrow$	t_3
t_4	$\leftarrow$	y	$loadAI$	$\$fp, @y$	$\Rightarrow$	t_4
t_5	$\leftarrow$	$t_4 \times t_3$	$mult$	t_4, t_3	$\Rightarrow$	t_5
t_6	$\leftarrow$	$t_5 + t_3$	add	t_5, t_3	$\Rightarrow$	t_6

Compilation: Représentation intermédiaires (3)

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

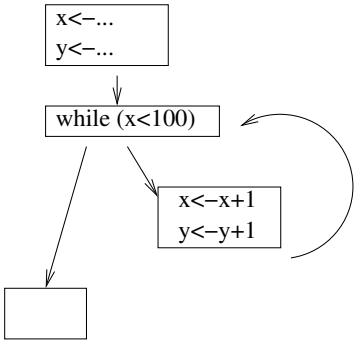
Exemple de l'appareil photo numérique

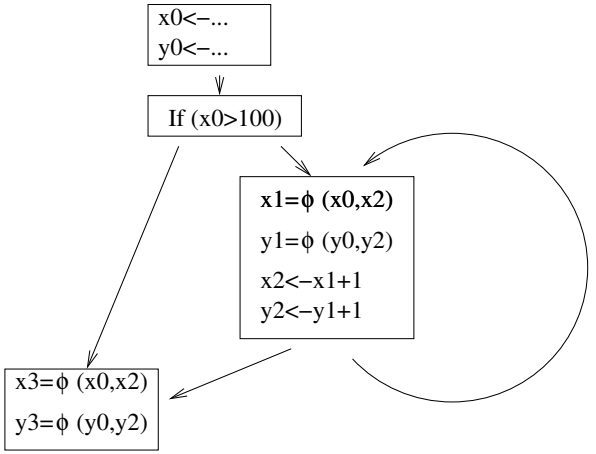
Conclusion

SSA: static single assignement

$x \leftarrow \ldots$
$y \leftarrow \ldots$
while $(x < 100)$
    $x \leftarrow x + 1$
    $y \leftarrow y + x$
end

$x_0 \leftarrow \ldots$
$y_0 \leftarrow \ldots$
if $(x_0 \geq 100)$ goto next
loop: $x_1 \leftarrow \phi(x_0, x_2)$
$y_1 \leftarrow \phi(y_0, y_2)$
$x_2 \leftarrow x_1 + 1$
$y_2 \leftarrow y_1 + x_2$
if $(x_2 < 100)$ goto loop
next: $x_3 \leftarrow \phi(x_0, x_2)$
$y_3 \leftarrow \phi(y_0, y_2)$





Compilation: Le front-end

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Le front-end d'un compilateur pour code embarqué utilise les mêmes techniques que les compilateurs traditionnels (on peut vouloir inclure des partie d'assembleur directement)
- Parsing LR(1): Le parseur est généré à partir de la grammaire du langage.
- Flex et bison: outils GNU

Compilation: Le middle-end

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Certaines phases d'optimisations sont ajoutées, elles peuvent être très calculatoires
- Quelques exemples de transformation indépendantes de la machine:
  - Élimination d'expressions redondantes
  - Élimination de code mort
  - Propagation de constantes

- Processeurs embarqués

Introduction

Architecture des processeurs

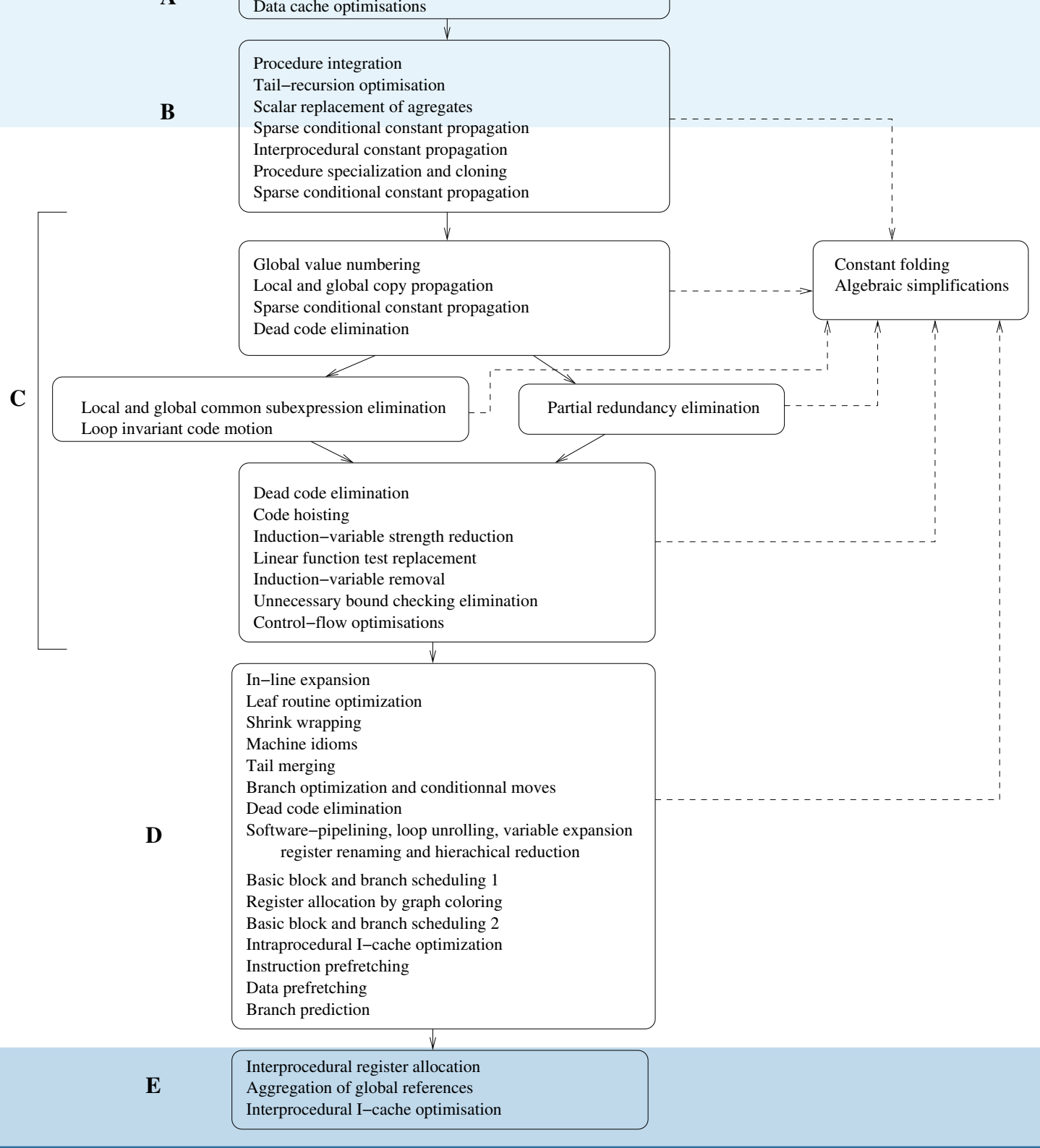
Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion



Compilation: Le back-end

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- La phase de génération de code est dédiée à l'architecture cible. Les techniques de compilation reciblable sont utilisées pour des familles d'architectures.
- Les étapes les plus importantes sont
  - Selection de code
  - Allocation de registre
  - Ordonnancement d'instructions

Exemple de code généré: MIPS

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

`gcc -S fib.c`

```
 int fib (int i) {
  if (i<=1) return(1);
  else return(fib(i-1)+fib(i-2));
}

int main (int argc, char *argv[]) {
  fib(2);
}
```

```
        .align      2
        .globl      main
        .ent        main
main:
        .frame      $fp,24,$ra
        .mask       0xc0000000,-4
        .fmask      0x00000000,0
        subu        $sp,$sp,24      # SP<-SP-24 :AR de 24 octet(6 mots)
        sw          $ra,20($sp)     # stocke adresse retour SP+20
        sw          $fp,16($sp)     # stocke ARP appelant SP+16
        move        $fp,$sp         # ARP <- SP
        sw          $a0,24($fp)     # stocke Arg1 dans la pile (ARP+24)
        sw          $5,28($fp)      # stocke Arg2 dans la pile (ARP+48)
        li          $a0,2           # $a0 <- 2 ($a0: Arg1)
        jal         fib             # jump and link fib($ra<-next instr)
        move        $sp,$fp         # SP <- ARP
        lw          $ra,20($sp)     # $ra <- adresse retour
        lw          $fp,16($sp)     # ARP <- ARP appelant
        addu        $sp,$sp,24      # SP->SP+24
        j           $ra             # jump adresse retour
        .end        main
```

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

gcc -S fib.c

```
 int fib (int i) {
  if (i<=1) return(1);
  else return(fib(i-1)+fib(i-2));
}

int main (int argc, char *argv[]) {
  fib(2);
}
```

```
        .file       1 "fib.c"
        .text
        .align      2
        .globl      fib
        .ent        fib
fib:
        .frame      $fp,40,$ra      # vars= 8, regs= 3/0, args= 16, ex
        .mask       0xc0010000,-8
        .fmask      0x00000000,0
        subu        $sp,$sp,40      # SP <- SP-40 :AR de 40 octet (10 m
        sw          $ra,32($sp)     # stocke adresse retour SP+32
        sw          $fp,28($sp)     # stocke ARP appelant SP+28
        sw          $s0,24($sp)     # sauvegarde registre $s0
        move        $fp,$sp         # ARP <- SP
        sw          $a0,40($fp)     # stocke Arg1 dans la pile (ARP+40)
        lw          $v0,40($fp)     # charge Arg1 dans $v0
        slt         $v0,$v0,2       # $v0 <- 1 si $v0<2 0 sinon
        beq         $v0,$0,$L2      # branch L2 si $v0==0
        li          $v0,1           # $v0 <- 0x1 ($v0 sera le registre
        sw          $v0,16($fp)     # stocke le resultat dans la pile
        j           $L1             # saute à L1
$L2:
        lw          $v0,40($fp)     # charge Arg1 dans $v0
        addu        $v0,$v0,-1      # retranche 1
        move        $a0,$v0         # $a0 <- $v0 ($a0 contient Arg1 pou
        jal         fib              # jump and link fib ($ra<-next ins
        move        $s0,$v0          # $s0 <- $v0 ($v0: res appel fib)
        lw          $v0,40($fp)      # charge Arg1 dans $v0
        addu        $v0,$v0,-2       # retranche 2
        move        $a0,$v0          # $a0 <- $v0 ($a0: contient Arg1 p
        jal         fib              # jump and link fib ($ra<-next ins
        addu        $s0,$s0,$v0      # $s0 <- $s0+$v0 ($v0: res appel f
        sw          $s0,16($fp)      # stocke le resultat dans la pile
$L1:
        lw          $v0,16($fp)     # $v0 <- resultat
        move        $sp,$fp         # SP <- ARP
        lw          $ra,32($sp)     # $ra <- adresse retour
        lw          $fp,28($sp)     # ARP <- ARP appelant
        lw          $s0,24($sp)     # restaure $s0
        addu        $sp,$sp,40      # SP->SP+40
```

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

Quelques exemples d'optimisation de compilation pour processeurs embarqués

Exemple 1: génération d'adresse pour DSP

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Les petits DSP possèdent souvent peu de registres et les calculs se font directement entre un registre par défaut REG et la mémoire.
- Minimiser les accès mémoire augmente beaucoup les performances.
- Pour cela, les assembleurs sont pourvus d'un registre d'adresse AR et de plusieurs mode d'adressage: adressage indirect ($REG \leftarrow Mem[reg_1]$) et indirect indexé ($REG \leftarrow Mem[reg_1 + const_1]$).
- ils possèdent généralement des versions avec auto-incrément (ou auto-décrement) des instructions standard: $load, store$.

Exemple: addressage avec auto incrément

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Auto-incrément à chaque $load, store$ le registre d'adresse AR est incrémenté ou décrémenté de 1 (éventuellement d'une constante c)
- Auto-modify à chaque $load, store$ le registre d'adresse AR est incrémenté ou décrémenté de la valeur d'un registre MR (*modify register*)
- Exemple de jeu d'instruction (assembleur TMS320C25):

$instruction$	$effet$
$LDAR\ AR, val$	$AR \leftarrow val$ //chargement du registre d'adresse AR avec une valeur
$LOAD\ *(AR)$	$REG \leftarrow mem(AR)$ //AR est le registre d'adresse
$LOAD\ *(AR)+$	$REG \leftarrow mem(AR); AR \leftarrow AR + 4$ //version avec auto incrément
$LOAD\ *(AR)-$	$REG \leftarrow mem(AR); AR \leftarrow AR - 4$ //version avec auto decrement
$STOR\ *(AR)$	$mem(AR) \leftarrow REG$ //AR est le registre d'adresse
$STOR\ *(AR)+$	$mem(AR) \leftarrow REG; AR \leftarrow AR + 4$ //version avec auto incrément
$STOR\ *(AR)-$	$mem(AR) \leftarrow REG; AR \leftarrow AR - 4$ //version avec auto decrement
$ADD\ *(AR)+$	$REG \leftarrow REG + mem(AR); AR \leftarrow AR + 4$ //
etc	

Exemple 1: addressage avec auto incrément

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

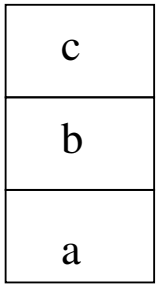
Conclusion

- Avec ce jeu d'instruction, le chargement d'une variable locale d'une procédure se fait en deux instructions:

$$LDAR\ AR, @a$$
$$LOAD\ *(AR)$$

- Mais l'addition de trois variables rangées consécutivement dans la pile: $a + b + c$ peut se faire en quatre instructions:

$$LDAR\ AR, @a$$
$$LOAD\ *(AR)+$$
$$ADD\ *(AR)+$$
$$ADD\ *(AR)$$



Optimisation de l'adressage

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Considérons le code suivant extrait d'une procédure:

$$
\begin{aligned}
c &= a + b \\
f &= d + e \\
a &= a + d \\
c &= d + a \\
b &= d + f + a
\end{aligned}
$$

- Le compilateur choisi d'accéder les variables dans un certain ordre, par exemple celui-ci:

$$a\ b\ c\ d\ e\ f\ a\ d\ a\ d\ a\ c\ d\ f\ a\ b$$

- Quelle est l'arrangement de ces différentes variables qui minimise la taille du code généré (et donc le temps d'exécution)?

Solution

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

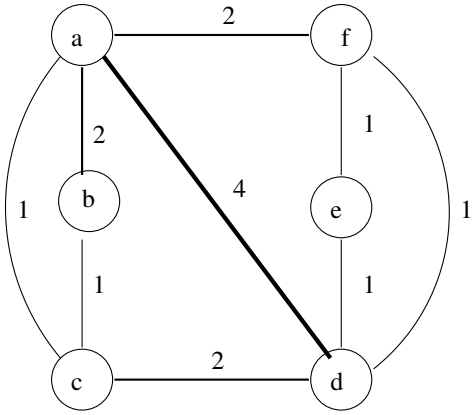
Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- On construit un graphe d'accès dont les noeuds représentent les variables et les poids sur les arcs représentent le nombre de fois ou les variables sont accédées consécutivement dans le code:
- On cherche un chemin hamiltonien dans ce graphe qui maximise le poids (problème NP-complet).
- solution: 10 accès avec la solution suivante:



a
c
d
a
f
e

Unité de génération d'adresse

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

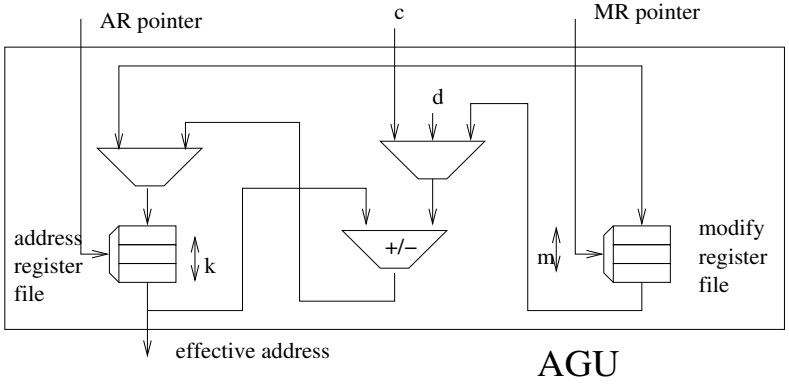
Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Pour les DSP plus gros, le problème est plus compliqué
- Ils possèdent souvent une unité de génération d'adresse hors du data-path (*address generation unit*, AGU) qui possède:
  - k registres d'adresse (AR_0,...,AR_k)
  - m registres de modification (MR_0,...,MR_m)
  - Un décalage cablé borné par une petite constante
  - ainsi que la possibilité d'entrer une constante pour le décalage ou pour modifier les AR ou les MR.



Quelques exemple d'unité de génération d'adresses (Leupers [?])

	TI C25	Motorola 56xxx	ADSP-210x	AMS Gepar
k	8	4	4	8
m	1	4	4	8
r	1	1	0	7

Exemple 2: cas des Registres dédiés

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

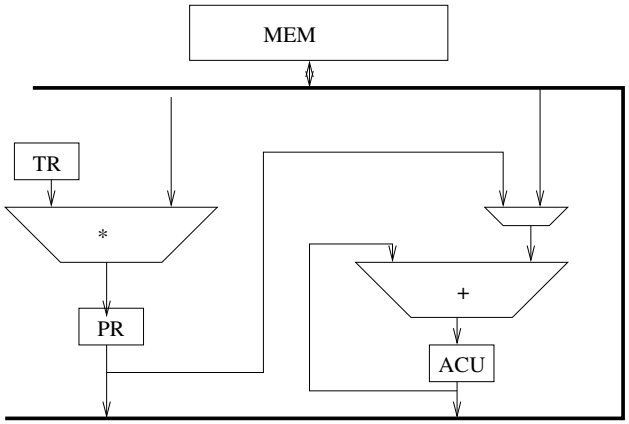
Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- De nombreux DSP ont des registres dédié dans le data-path qui permettent:
  - Un mini-pipeline à l'intérieur du data-path
  - De limiter les lectures/ecritures sur le fichier de registre
  - De réduire la taille du code (les registres sont adressé implicitement)
- exemple: le chemin de donnée et le jeu d'instruction associé du C25 de Texas Instrument.



instruction	effet
lac	ACU ⟵ MEM
addk	ACU⟵ ACCU+Constant
add	ACU⟵ ACCU+MEM
pac	ACU ⟵ PR
apac	ACU⟵ ACCU+PR
mpy	PR⟵ PR*MEM
lt	TR⟵ MEM
sacl	MEM⟵ ACU
spl	MEM⟵PR

Registre dédiés

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Dans les compilateurs standards, il y a une phase de sélection de code suivie d'une phase d'allocation de registres
- Il devient difficile de séparer la phase d'allocation de registres et la phase de sélection de code.
- Trouver le meilleur code pour une séquence d'instructions donnée devient quasiment impossible/
- La solution passe par des algorithmes d'optimisation de type heuristique ou aléatoire (algorithmes génétique, recuit simulé, etc.)

Exemple 3: gestion du cache d'instruction

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

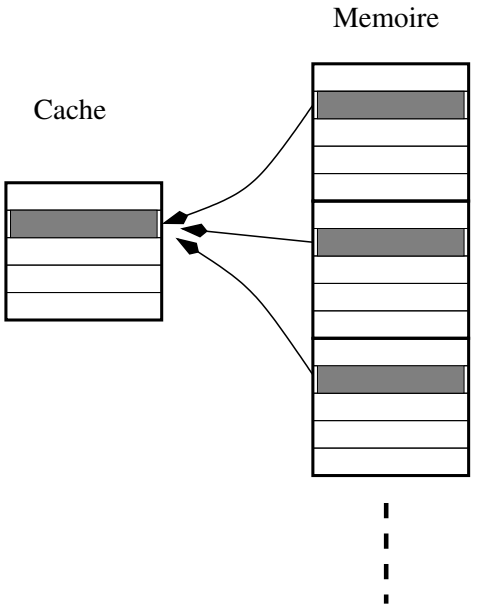
Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Exemple du ST200: DSP VLIW pour telephone portables
- Le cache d'instructions est "direct-mapped"



- Taille du cache d'instructions: 32K
- Taille d'une ligne de cache 64 bytes: 512 lignes.
- Coût d'un "cache-miss": 150 cycles.
- Pour un placement de code aléatoire les performances varient avec un rapport de 3!

Conflits de cache d'instruction

- Processeurs embarqués

Introduction

Architecture des processeurs

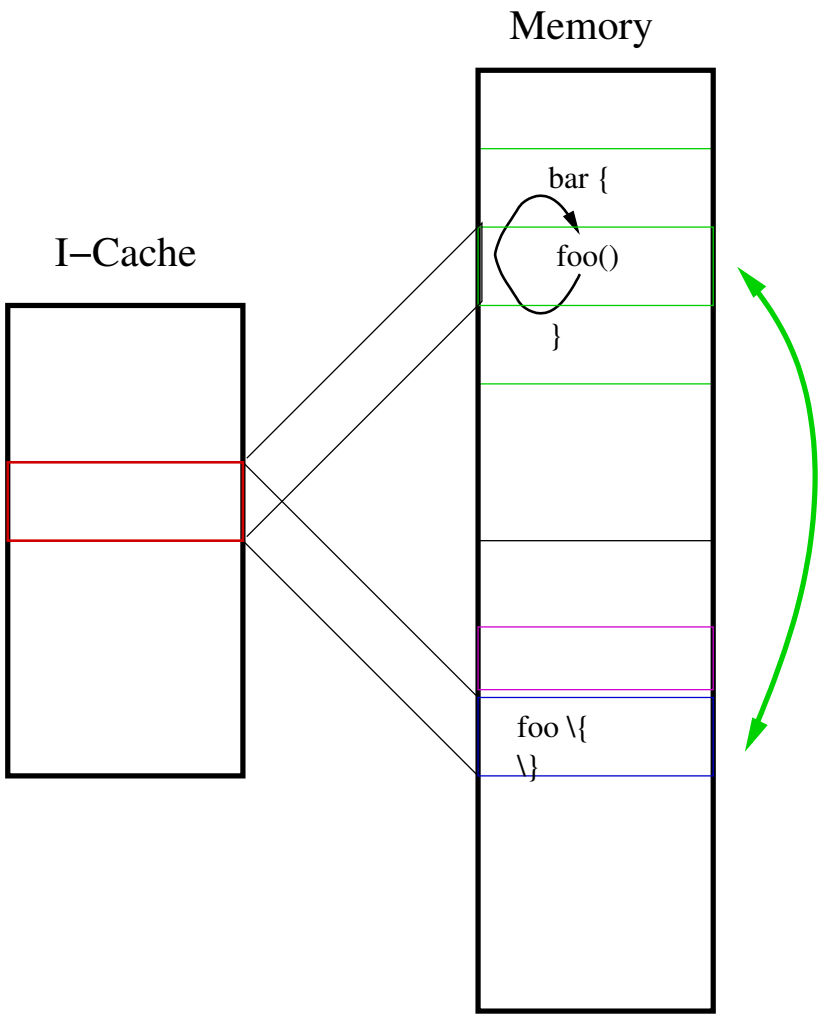
Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion



Pas de Conflits de cache d'instruction

- Processeurs embarqués

Introduction

Architecture des processeurs

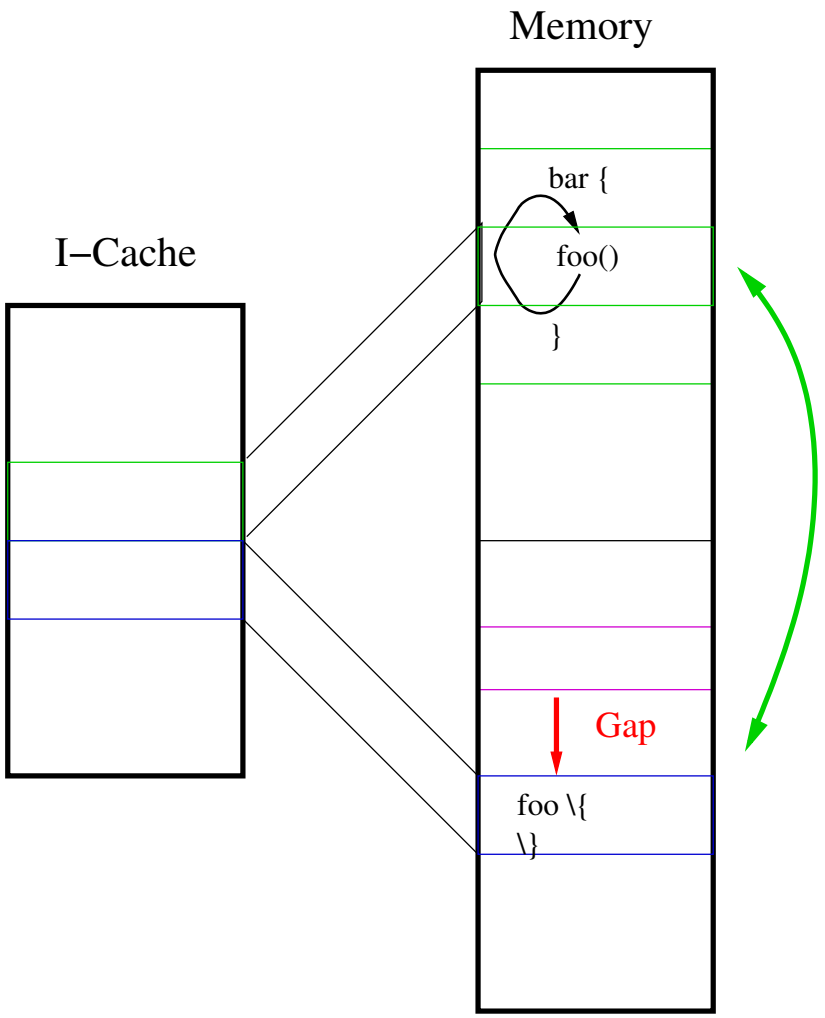
Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion



Optimisation proposée

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Placement explicite des fonctions en mémoire pour limiter les conflits du cache d'instrution (au moment de l'édition de liens)
- Ces conflits dépendent du flot d'exécution, on optimise donc pour une trace données:
- Analyse de la trace: quelles fonctions sont appelées ensembles
- Déduction d'un placement optimal en mémoire des différentes fonctions (pour cette trace là).
- Plusieurs approche existe:
  - Basée sur un graphe d'appel
  - Basée sur un graphe conflit

Utilisation du graphe d'appel

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

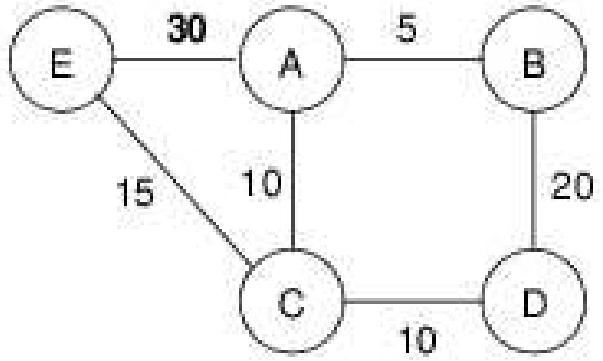
Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

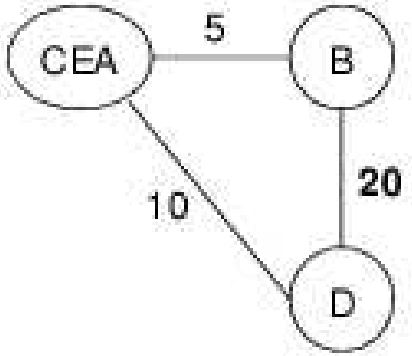
Exemple de l'appareil photo numérique

Conclusion

- Construction d'un graphe où les sommets sont des fonctions et les arêtes *les fréquences d'appel*.
- Deux fonctions souvent appelées doivent avoir une place différente dans le cache. C'est assuré si on les place cote à cote dans la mémoire.
- L'algorithme est glouton sur le graphe: il fusionne les sommets reliés par l'arête de poids max.



(a) Combine E and A; merge edges (E,C) and (A,C) into (EA,C).



(c) Combine B and D; merge edges (B,CEA) and (D,CEA).

Utilisation du graphe des conflits

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Construction d'un graphe (GC) où les sommets sont des fonctions (ou des morceaux de code) et les arêtes sont les conflits constatés sur ne trace particulière
- Il y a conflit entre A et B si la trace contient le pattern suivant: $A \ldots B \ldots A$ ou $B \ldots A \ldots B$.
- Exemple:
  `ABCBABABABA`
  7 conflits entre `A` et `B`, 1 conflit entre `A` et `C` 1 conflit entre `B` et `C`

$\Rightarrow$ On place en priorité `A` et `B` consecutivement en mémoire

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

Quelques manipulation avec GCC

Un compilateur pour l'embarqué: GCC

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués
- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- GCC: Gnu C Compiler ou Gun Compiler Collection
- http://gcc.gnu.org/
- Outil développé par la communauté mondiale, développement rapide.
- De plus en plus utilisé pour le calcul embarqué car il est reciblable.
- Exemple d'utilisation
  - Créer un compilateur pour le Mips sur votre Pentium
  - Insérer une routine d'interruption dans votre programme
  - Répartir les différentes section de votre code dans différentes mémoire.

compilateur, éditeur de liens, binutils

- Processeurs embarqués

Introduction

Architecture des processeurs

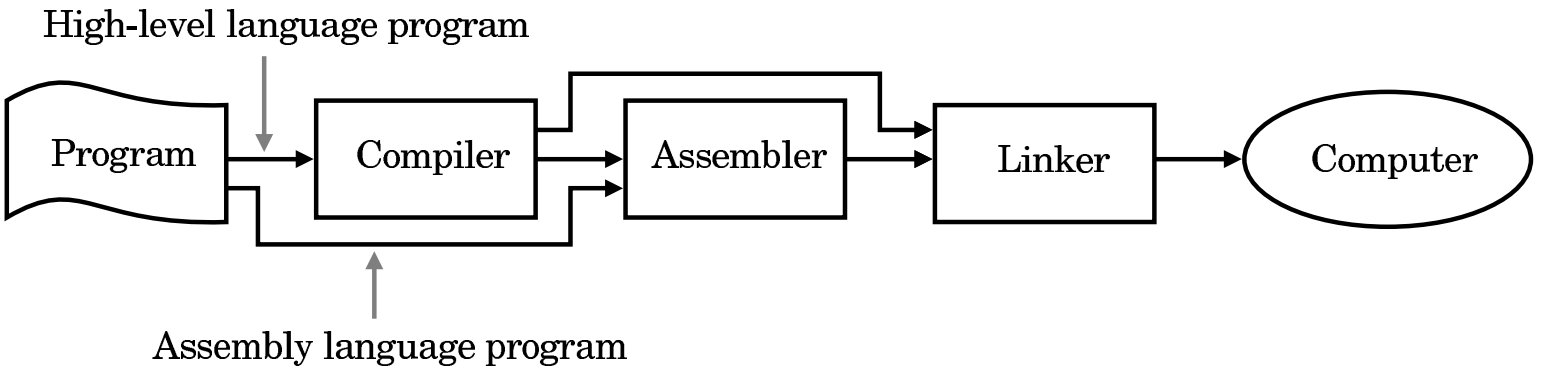
Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion



- gcc: `ftp://ftp.gnu.org/gnu/gcc/gcc-3.3.tar.gz`
- Assembleur, éditeur de lien et utilitaire de manipulation de binaire (`objdump`, etc..) `ftp://ftp.gnu.org/gnu/binutils/binutils-2.9.1.tar.gz`
- Run-time library (`printf`, `malloc`, etc.): `newlib` (ou `glibc`) `ftp://sourceware.cygnus.com/pub/newlib/newlib-1.8.1.tar`

Installation pour le Mips

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- `tar zxf gcc-3.3.tar.gz`
- `tar zxfbinutils-2.9.1.tar.gz`
- ```
  cd binutils-2.9.1
  configure
   -target=mipsel
  make all install
  ```
- Même chose pour `gcc`
- Voir le fichier configure.sub pour les plate-formes cibles possibles
- On peut aussi compiler gcc sur une machine pour l'utiliser sur une autre machine (ou il produira du code pour une troisième machine): GCC est un *Cross-compiler*
  ```

Assembleur dans le code C

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- On peut include directement des instruction assembleur dans le code C: fonction `__asm__`

```
void set_imask_to_6( void )
{
  printf( "switching to interrupt
   mask level 6.\n" );
  __asm__( " andi #0xf8, sr" );
  __asm__( " ori #6, sr" );
  printf( "Interrupt mask level
   is now 6.\n" );
}
```

- Permet d'écrire des pilotes de périphériques, de contrôler la gestion des interruptions sans système d'expoitation

Assembleur dans le code C

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- On peut aussi mettre explicitement des variables dans des registres sans connaître l'allocation de registres faite par le processeur
- Exemple: utilisation de la fonction `fsinx` du 68881:

```
__asm__("fsinx %1,%0" : "=f" (result) : "f" (angle));
```

- %0 et %1 représente le résultat et l'opérande de la fonction qui vont correspondre aux variables `result` et `angle` du programme C
- `"t"` est une directive indiquant à `gcc` qu'il doit utiliser des registres flottants

Contrôler les section du programme

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Le code contient différentes sections. Par exemple avec GCC
  - La section `.text` contient les instructions du programme
  - La section `.data` contient des données statiques etc.
- Le concepteur de logiciel embarqué veut souvent contrôler explicitement la répartition des variables globales dans les sections (à la compilation): pour distinguer les variables des constantes par exemple.
- Il peut vouloir aussi contrôler la répartition des sections dans les composants matériels (à l'édition de lien et au chargement du programme)
- Utilisation de la directive `__attibute__`

```
const int put_this_in_rom
      __attribute__((section("myconst" )));
const int put_this_in_flash
      __attribute__((section("myflash" )));
```

Gestionnaire d'interruption

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Beaucoup de périphériques dédiés communiquent avec le processeur par des interruptions
  - Problèmes (adresse mémoire invalide), erreur de transmission sur le bus
  - Fin de tache pour un accélérateur matériel
  - Détection de données pour un censeurs
- GCC ne peut pas généralement pas directement compiler un gestionnaire d'interruption (retour par `rte`: return from exception)
- on peut contourner ce problème en encapsulant la procédure de gestion de l'interruption

```
/* code C de gestion de l'interruption */
 void isr_C( void ) {
  /* ISR: interupt service routine
    faire quelque chose en C */
  ...
}

/* code assembleur utilisé
   dans le code source */
 __asm__("
  .global _isr
  _isr:
  /* Sauvegarde des registres
   * choisis
   */
  push r0
  push r1
  ...
  /* appel du gestionnaire */
  jsr _isr_C
  /* restauration des registre
   * et retour
   */
  ...
  pop r1
  pop r0
  rte
");
```

Le LD script

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- L'éditeur de liens assemble les différents fichiers objets résultant de la compilation séparée des différents fichiers.
- C'est là que sont résolus les appels à des fonctions entre fichier ou à des fonctions de bibliothèque non fournies par l'utilisateur
- C'est aussi là que sont agencées les différentes sections mémoire dans l'espace d'adressage final

Exemple de LD script

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Spécification de fichier de librairie
- Spécification du format de sortie
- 5 sections nommées

```
/* Une list de fichier à inclure (les autres sont
     spécifiés par la ligne de commande */
INPUT(libc.a libg.a libgcc.a libc.a libgcc.a)

/* Specification du format de sortie
(binaire  :bin}, Intel Hex:
ihex, debug coff-\$target */
  OUTPUT_FORMAT("coff-sh")


/* list of our memory sections */
MEMORY {
 vect : ORIGIN = 0x00000000, LENGTH = 1k
 rom  : ORIGIN  = 0x00000400, LENGTH = 127k
 reset: ORIGIN = 0xBFC00000, LENGTH = 0x00000400
 ram  : ORIGIN = 0x400000, LENGTH = 128k
 cache : ORIGIN = 0xfffff000, LENGTH = 4k
}
```

Exemple de LD script (suite)

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués
- Compilation pour processeurs embarqués
- Compilation: biblio
- Compilation: Principes généraux
- Quelques exemples d'optimisation de compilation pour processeurs embarqués
- Exemple 1: génération d'adresse pour DSP
- Exemple 2: cas des Registres dédiés
- Exemple 3: gestion du cache d'instruction
- Quelques manipulation avec GCC
- Un compilateur pour l'embarqué: GCC

Exemple de l'appareil photo numérique

Conclusion

- Description du placement de chaque section en mémoire
- Création d'un symbole au début de la section (ex: `__text_start`) et à la fin (`__text_end`)
- Placement des parties du code préfixé par la directive `.text` dans la section `rom` de la mémoire.
- Éventuellement insertion de code spécifiquement écrit directement dans la mémoire (`.reset`)

```
SECTIONS {
 /* the interrupt
    vector table */
 .vect :
 {
  __vect_start = .;
  *(.vect);
  __vect_end = .;
 } > vect

 /* code and constants */
 .text :
 {
  __text_start = .;
  *(.text)
  *(.strings)
   __text_end = .;
 } > rom

.reset : {
   ./libhandler.a(.reset)
   } > reset
```

```
 /* uninitialized data */
 .bss :

 {
   __bss_start = . ;
  *(.bss)
  *(COMMON)
   __bss_end = . ;
 } > ram

 /* initialized data */
 .init : AT (__text_end)
 {
  __data_start = .;
  *(.data)
  __data_end = .;
 } > ram

 /* application stack */
 .stack :
 {
   __stack_start = .;
  *(.stack)
   __stack_end = .;
 } > ram
}
```

- Processeurs embarqués

Introduction

Architecture des processeurs

Différents types de processeurs embarqués

Compilation pour processeurs embarqués

Exemple de l'appareil photo numérique

- Exemple de l'appareil photo numérique
- Le point de vue du concepteur
- Charge-coupled device (CCD)
- Discrete Cosine Transform: DCT
- Étape de quantization
- Encodage de Huffman
- Contrainte du système
- Spécification fonctionnelle
- Implémentation 1: Microcontroller seul
- Implémentation 2: Microcontroller et CCDPP

Conclusion

Exemple de l'appareil photo numérique Tiré du cours de Franck Vahid: http://www.cs.ucr.edu/content/esd/