

Architecture des ordinateurs

Cpu et instructions

Florent Devin



Ecole Internationale des Sciences du Traitement de
l'Information

Instructions

Introduction

Introduction

- Correspondance directe
 - Jeu d'instruction d'une machine
 - Sa programmation
 - Conception de sa CPU
- Concevoir une CPU : concevoir le jeu d'instructions.

Généralités

Définitions

- Jeu d'instruction : ensemble des instructions d'une CPU
- Une machine : jeu d'instructions et CPU spécifique.
- Informations requises pour une instruction
 - Code opération : opération à exécuter, appelé opcode
 - Référence aux opérandes sources (adresses)
 - Référence à l'opérande résultat
 - la référence à la prochaine instruction

Généralités

Représentation de l'instruction

- Une instruction : une séquence de bits
- Différents formats d'instruction selon le nombre de parties réservées aux opérandes.
- Un jeu d'instructions : plusieurs formats d'instruction
- Extraction des champs par la CPU depuis IR

Généralités

Représentation de l'instruction

- Association d'une séquence de bit : mnémonique
- Représentation binaire fixe de l'opcode
- Possibilité de programmer en employant le langage machine
- ADD R, Y : ajouter la valeur contenue à l'adresse Y au contenu du registre R.

Généralités

Construction d'un jeu d'instruction

- Suffisamment expressif pour coder toute instruction d'un langage de haut niveau.
- Une instruction d'un langage de haut niveau : plusieurs instructions du langage machine.
- Il n'y a pas de consensus sur la conception d'un jeu d'instructions.

Généralités

Construction d'un jeu d'instruction

- Les aspect fondamentaux à prendre en compte sont
 - les opérations : combien en faut il, quelle doit être leur complexité ?
 - les types de données : quels sont les types de données supportés par les opérations ?
 - les registres : combien en faut il et comment sont ils utilisés ?
 - l'adressage : quelles sont les différentes façons de faire référence aux données ?
 - le format : quelle (s) longueur(s) d'instruction adopter, quelle sera le nombre et la taille de champs ?

Caractéristiques

Format d'instruction

- Une opération : référence
 - des données sources
 - une destination
- Références : adresses des données.
- Format d'une instruction : nombre de champs utilisés
- En théorie, il en faudrait quatre
 - deux opérandes
 - un résultat
 - la prochaine instruction

Caractéristiques

Format d'instruction

- En pratique
 - Instruction à trois adresses peu courants, car trop long
 - Instruction à deux adresses : source et destination confondue
 - Instructions à une adresse : emploient un registre CPU, l'accumulateur, pour garder une opérande et le résultat ;
 - Instructions sans adresse : utilisation d'une pile

Caractéristiques

Format d'instruction

- Choix du nombre d'adresse : compromis
 - moins il y a d'adresses, plus les instructions sont courtes, moins la CPU est complexe ;
 - mais les instructions seront plus nombreuses, donc les programmes plus lents à exécuter.
- Disposer de plusieurs adresses : utilisation de plusieurs registres
- Actuellement un jeu d'instructions : formats 2 et 3 adresses

Caractéristiques

Relation entre mémoire et registres

- Registres : éléments interne de mémorisation rapide
- Permet de minimiser les accès à la mémoire
- Accélère l'exécution des instructions

Caractéristiques

Catégories d'instructions

- transfert de données
 - Emplacement des opérandes source et destination
 - Taille des données à transférer
 - Manière d'accéder à ces données (le mode d'adressage)
- transfert de contrôle : changement explicite du compteur ordinal
- les opérations arithmétiques
- les opérations logiques
- E/S : transfert de données avec les E/S

Caractéristiques

MMX et autres

- En 1996, Intel crée les instructions MMX
- 57 nouvelles instructions dédiées aux opérations multimédia
- Une même opération sur plusieurs éléments à la fois (principe SIMD).
- Trois nouveaux types de données chacun de 64 bits découpés en champs contenant chacun un entier
 - packed bytes (8 octets)
 - packed word (4 mots)
 - packed doubleword (2 doubles mots)

Structure d'une instruction

Remarques

Dans la conception d'un jeu d'instruction, un paramètre important est le nombre de registres et leur utilisation.

Structure d'une instruction

Mode d'adressage

- Adressage immédiat
 - Opérande : donnée par l'instruction
 - Pas de référence à la mémoire
 - Taille de l'opérande limitée
- Adressage direct
 - Opérande : adresse mémoire donnée par l'instruction
 - Taille de l'espace adressable : limitée

Structure d'une instruction

Mode d'adressage

- l'adressage registre
 - Opérande : donnée par l'adresse du registre
 - Pas de référence à la mémoire
 - taille de l'espace adressable est limitée
- l'adressage indirect par mémoire
 - Adresse de l'opérande : adresse mémoire donnée dans l'instruction
 - Implique plusieurs références à la mémoire

Structure d'une instruction

Mode d'adressage

- l'adressage indirect par registre
 - adresse de l'opérande : contenu d'un registre dont l'adresse est donnée dans l'instruction
- adressage déplacement
 - l'adresse de l'opérande est donnée en deux parties, une base dans un registre (peut être le Program Counter) et un déplacement (à ajouter à la base) dans l'instruction. Flexible, mais complexe.

Structure d'une instruction

Remarques

- Modes les plus utilisés : adressage immédiat, indirect par registre et déplacement
- Toutes les machines disposent de plusieurs modes d'adressage.

Structure d'une instruction

Mode d'adressage pour les jeux d'instructions

Deux manières d'implanter le mode d'adressage

- par l'utilisation d'un ou plusieurs bits indiquant le mode employé
 - Appelés spécificateur d'adresse
 - Utile lorsqu'on souhaite disposer d'un grand nombre de mode d'adressage
 - Peut donner lieu à une taille d'instruction variable
- par l'utilisation d'opcodes différents
 - ce qui permet de conserver une taille d'instruction fixe
 - Réalisation matériel plus simple

Structure d'une instruction

Format d'instruction

- Le format d'une instruction définit les champs de l'instruction (opcode et adresses) et l'adressage employé.

Structure d'une instruction

Taille d'instruction

- Jeu d'instructions complexe : programmation compacte
- Jeu d'instruction complexe : longueur d'instruction importante
- longueur de l'instruction : multiple de la taille du bus de données.

Quelques exemples

Alpha

- Le processeur Alpha de DEC contient
 - 32 registres (R0 à R31) de 64 bits pour la manipulation des entiers.
 - 32 registres (F0 à F31) de 64 bits pour la manipulation des flottants.
 - Les registres R31 et F31 sont à lecture seule et contiennent la représentation de 0.
- Les instructions ont une taille fixe de 32 bits.

Quelques exemples

Alpha

- Quatre formats d'instruction possibles
 - les instructions spécifiques au système d'exploitation
 - opcode sur 6 bits + un nombre sur 26 bits ;
 - les instructions de branchement
 - opcode sur 6 bits + adresse registre sur 5 bits + déplacement sur 21 bits ;

Quelques exemples

Alpha

- Quatre formats d'instruction possibles
 - les instructions de transfert de données
 - opcode sur 6 bits + adresse registre sur 5 bits + adresse registre sur 5 bits + déplacement sur 16 bits
 - les instructions utilisées pour les opérations de calcul entier ou flottant
 - opcode sur 6 bits + adresse registre sur 5 bits + adresse registre sur 5 bits + extension de l'opcode sur 11 bits + adresse registre sur 5 bits

Quelques exemples

SPARC

- Le processeur SPARC de SUN possède un grand nombre de registres (peut être supérieur à 520).
- Néanmoins seuls 32 registres sont visibles simultanément, divisés en 4 groupes de registres spécifiques.
- Les instructions ont une taille fixe de 32 bits.
- Le format est codé sur 2 bits.

Quelques exemples

SPARC

- 3 format d'instructions différents sont utilisés
 - le premier ne correspond qu'à une seule opération de branchement, l'appel de sous-programme. Cette instruction ne comprend donc pas d'opcode mais simplement un déplacement sur 30 bits ;
 - le deuxième a un opcode sur 3 bits. Cet opcode distingue 2 types d'instruction
 - les instructions de branchement : une condition de saut sur 4 bits (comparée au contenu d'un registre de condition), un déplacement sur 22 bits
 - une instruction permettant de charger une donnée dans les poids fort d'un registres : adresse registre sur 5 bits + 22 bits de donnée immédiate ;

Quelques exemples

SPARC

- le troisième format correspond à toutes les autres opérations
 - une adresse registre de destination sur 5 bits, une adresse registre source sur 5 bits, une extension de l'opcode sur 6 bits. Selon cette extension
 - une extension du code opération pour les opérations flottante sur 9 bits + une adresse registre source sur 5 bits
 - une donnée immédiate sur 13 bits
 - un espace d'adressage mémoire sur 8 bits + une adresse registre source sur 5 bits.

Quelques exemples

Pentium II

- Le pentium II comprend 9 modes d'adressage, pouvant utiliser un registre ségment, un registre de base et un registre d'index.
- Le pentium comprend une grande variété de formats, l'opcode étant le seul élément récurrent.
- Le mode d'adressage est donnée par l'opcode.

Quelques exemples

Pentium II

- L'instruction comporte
 - un préfixe de 0, 1, 2, 3 ou 4 octet,
 - un opcode de 1 ou 2 octets,
 - un spécificateur d'adresse de 0, 1 ou 2 octets,
 - un déplacement de 0, 1, 2 ou 4 octets,
 - un champ pour l'adressage immédiat de 0, 1, 2 ou 4 octets

Conclusion

Conclusion

- On oppose traditionnellement
 - CISC (Complex Instruction Set Computer), les machines possédant un jeu d'instruction complexe, en nombre d'instruction, de mode d'adressages, de registres, etc... (comme le pentium)
 - RISC (Reduced Instruction Set Computer), les machines possédant un jeu d'instruction réduit (comme le SPARC ou l'ALPHA) et donc une architecture correspondant à ce jeu d'instruction réduit.

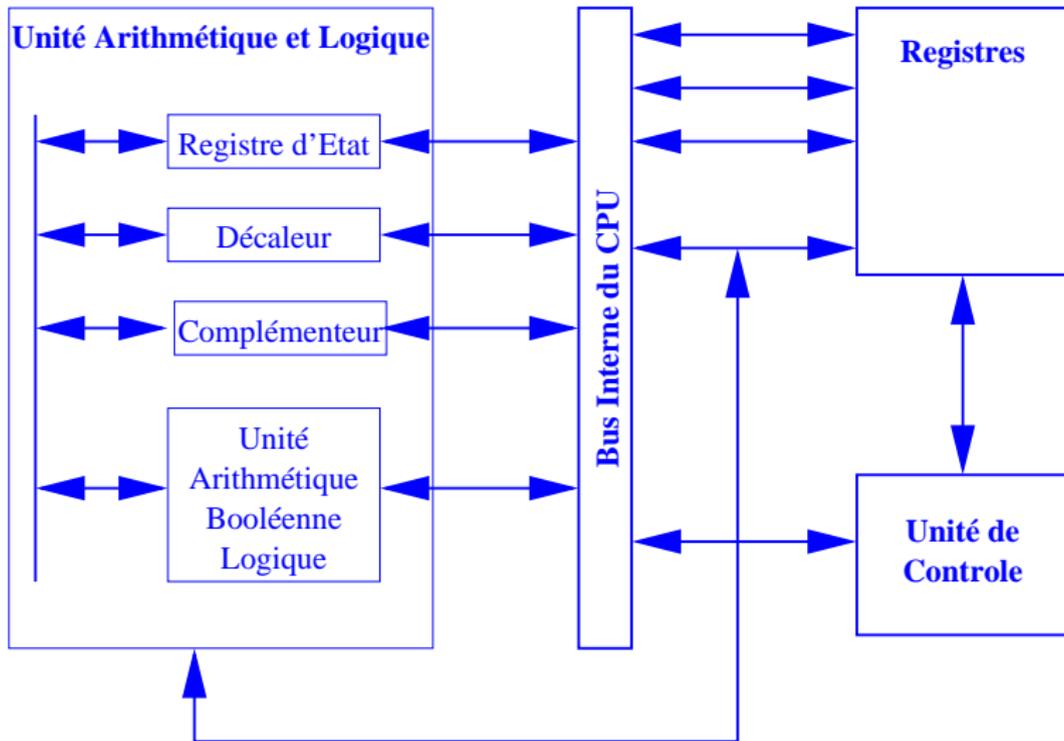
Cpu

Organisation de la CPU

Organisation du processeur

- La CPU possède
 - une ALU pour le traitement des données ;
 - une unité de commande qui contrôle le mouvement des données et des instructions, ainsi que les opérations de l'ALU ;
 - des registres spécifiques pour stocker temporairement les données et les instructions ;
 - un bus interne pour interconnecter ces différentes composants.

Organisation de la CPU



Organisation de la CPU

Organisation des registres

- Exécution d'une instruction : stockage temporaire de données.
- Deux types de registres
 - visibles par l'utilisateur : optimisation des références à la mémoire
 - registres de contrôle et de statuts

Organisation de la CPU

Registres visibles par l'utilisateur

- Registre utilisateur : registre utilisable via des instructions
- Plusieurs catégories :
 - données : Pas utilisable pour des adresses (taille trop petite)
 - adresses : souvent dévolus à un mode d'adressage particulier
 - conditions (flags) : suite de bits positionnée en fonction du résultat d'une opération ;
 - autres : n'ont pas de fonction spécifique.

Organisation de la CPU

Registres de contrôle et de statuts

- En général, ces registres ne sont pas visibles par l'utilisateur.
- Quatre registres essentiels :
 - le compteur ordinal (PC, pour Program Counter)
 - le registre d'instruction (IR)
 - le registre d'adresse mémoire (MAR)
 - le registre tampon mémoire (MBR) : Fait le lien avec les registres visibles par l'utilisateur.
- Des registres supplémentaires peuvent être intercalés entre l'ALU et les registres utilisateurs/le MBR.

Organisation de la CPU

Registres de contrôle et de statuts

- Registre PSW (Program Status Word)
 - signe : le bit de signe du résultat de la dernière opération arithmétique
 - zéro : à 1 lorsque le résultat est 0
 - retenue : à 1 lorsqu'une opération a générée une retenue
 - égal : à 1 si le résultat d'une comparaison est une égalité
 - débordement : à 1 lorsqu'une opération a provoqué un débordement
 - interruption : indique si le fonctionnement normal peut être interrompu
 - superviseur : indique un mode privilégié
- D'autres spécifiques peuvent être intercalés

Cycle d'une l'instruction

Cycle normal

- recherche de l'instruction (fetch) : lecture de l'instruction depuis la mémoire
- interprétation de l'instruction (decode) : décodage de l'instruction pour déterminer l'action

Cycle d'une l'instruction

execute

- recherche des données (`fetch data`) : lecture de données depuis la mémoire ou le module d'E/S
- traitement des données (`process data`) : opération arithmétiques ou logique
- écriture des données (`write data`) : écriture du résultat dans la mémoire ou le module d'E/S

Cycle d'une l'instruction

Schéma d'exécution

- répéter
 - fetch
 - decode
 - execute

Cycle d'une l'instruction

Flots de données

- La séquence exacte des évènements se produisant durant un cycle dépend de l'architecture de la CPU.
- Considérons le cas générique d'une CPU possédant
 - un registre MAR ;
 - un registre MBR ;
 - un compteur ordinal PC ;
 - un registre d'instruction IR.

RISC

Introduction

- Compilateur : production d'instructions simples
- \Rightarrow Instructions complexes des CISC rarement exploitées

RISC

Caractéristiques

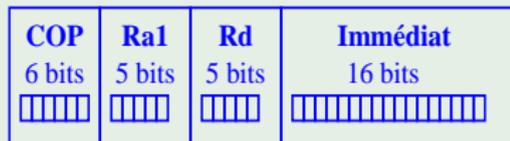
- Vers une instruction par cycle
- Certaines instructions implanter matériellement
- Opérations registre - registre
- Modes d'adressage simples et peu nombreux
- Modes d'adressages complexes : réalisés à partir de ceux-ci
- Formats d'instructions simples et peu nombreux
- Taille d'instruction fixe, possédant des champs fixes.

RISC

Exemple de machine RISC

Considérons la machine RISC suivante.

- Jeu d'instructions et registres
 - La machine utilise 3 formats d'instructions (on ne s'intéresse pas au troisième format correspondant aux instructions de saut).



RISC

Exemple de machine RISC

- Quatre groupes d'instructions
 - accès mémoire : instruction de lecture - écriture ;
 - opération registre - registre : calcul ou échange de données n'impliquant que des registres ;
 - opérations registre - immédiat : calcul impliquant un registre et une donnée immédiate ;
 - branchement.

RISC

Exemple de machine RISC

- RI : registre d'instruction
- CP : compteur ordinal
- NCP : adresse de l'instruction suivante
- A, B, IMM entrées d'ALU
- SALU : sortie ALU
- COND : registre de condition/statut
- DMC : sortie mémoire de données
- des registres utilisateurs

RISC

Cycle d'instruction

- fetch (LI)
 - $RI \leftarrow \text{mémoire d'instruction (PC)}$
 - $NCP \leftarrow CP+4$
- decode (DI)
 - $A \leftarrow \text{registre}(RI_{25-21})$
 - $B \leftarrow \text{registre}(RI_{20-16})$
 - $IMM \leftarrow RI_{15..0}$

RISC

Cycle d'instruction

- selon l'opcode,
 - accès mémoire
 - $SALU \leftarrow A + IMM$
 - opération registre-registre
 - $SALU \leftarrow A \text{ op } B$
 - opération registre-immédiat
 - $SALU \leftarrow A \text{ op } IMM$
 - branchement
 - $SALU \leftarrow NCP + IMM$
 - $COND \leftarrow A \text{ op } 0$

RISC

Cycle d'instruction

- accès mémoire/branchement (MEM)
 - chargement
 - $DMC \leftarrow \text{mémoire données(SALU)}$
 - $CP \leftarrow NCP$
 - rangement
 - $\text{mémoire données(SALU)} \leftarrow B$
 - $CP \leftarrow NCP$
 - branchement
 - si GOND alors $CP \leftarrow \text{SALU}$ sinon $CP \leftarrow NCP$
 - autres instructions
 - $CP \leftarrow NCP$

RISC

Cycle d'instruction

- écriture du résultat (ER)
 - chargement
 - registre ($R_{120..16}$) \leftarrow DMC
 - opération registre-registre
 - registre ($R_{15..11}$) \leftarrow SALU
 - opération registre-immédiat
 - registre ($R_{20..16}$) \leftarrow SALU

RISC

Cycle d'instruction

- L'exécution d'une instruction prendra de 4 à 5 cycles, les plus rapides étant les instructions de branchement.
- Pour une machine de ce type, on estime qu'elles constituent 12% des instructions exécutés.
- La durée moyenne d'une instruction est donc de $12\% \times 4 + 88\% \times 5 = 4,88$ cycles.

Pipeline

Pipeline à 3 étages

- Amélioration des performances : diminution du temps d'exécution
- Idée : Faire travailler différentes unités fonctionnelles à la chaîne
 - Une instruction : 3 phases indépendantes (fetch, decode, execute)
 - Création de trois unités

Pipeline

Pipeline

- Ainsi, au cycle i
 - fetch de l'instruction i
 - decode l'instruction $i - 1$
 - execute l'instruction $i - 2$

Cycle d'horloge	1	2	3	4	5	6
étage fetch	inst1	inst2	inst3	inst4	inst5	inst6
étage decode		inst1	inst2	inst3	inst4	inst5
étage execute			inst1	inst2	inst3	inst4

Pipeline

Pipeline

- Temps d'exécution d'une instruction : divisé par 3
- Pipeline à 3 étages
- Temps d'exécution de n instructions : $2 + n$ cycles d'horloge.
- Délai de 2 cycles : temps de latence du pipeline.

Pipeline

Aléa structurel

- Apparaît lorsqu'un conflit de ressources ne peut pas être géré par le matériel.
- C'est le cas de la mémoire unique pour instructions et données.
- Dans ce cas, le pipeline est suspendu durant un cycle pour lever l'aléa.

Pipeline

Aléa de données

- Apparaît lors d'une dépendance de données entre les instructions.
- Soit par exemple les deux instructions
 - $r2 \leftarrow \text{mémoire}(120)$
 - $r3 \leftarrow r2 + r1$

Pipeline

Aléa de contrôle

- Concerne les instructions de modification du registre PC
- branchement conditionnel
- Principal facteur de dégradation de performance dans une architecture pipeline.

Pipeline

Solution proposées

- dupliquer l'architecture de pipeline pour traiter les deux cas du branchement (pris ou pas) ;
- précharger l'instruction correspondant à l'adresse de branchement (quitte à ne pas l'utiliser) ;
- se baser sur une prédiction des branchements
- supposer qu'un branchement ne sera jamais/toujours pris ;
- supposer que certains opcodes favorisent le branchement ;
- générer des instruction NOP (No OPeration) après l'instructions de branchement le temps que le branchement puisse se conclure.