Modèle d’execution

Modèle d'exécution

- Modèles d'exécution (n,m)
- n : nombre d'opérandes par instruction
- m : nombre d'opérandes mémoire par instruction
- Les différents modes
  - RISC : (3,0)
    - Instructions de longueur fixe
    - Load et Store : seules instructions mémoire
  - IA-32 : (2,1)
  - Pile (0,0)
    - Tous les opérandes sont accédés via la pile

Modèles d'exécution (3,0)

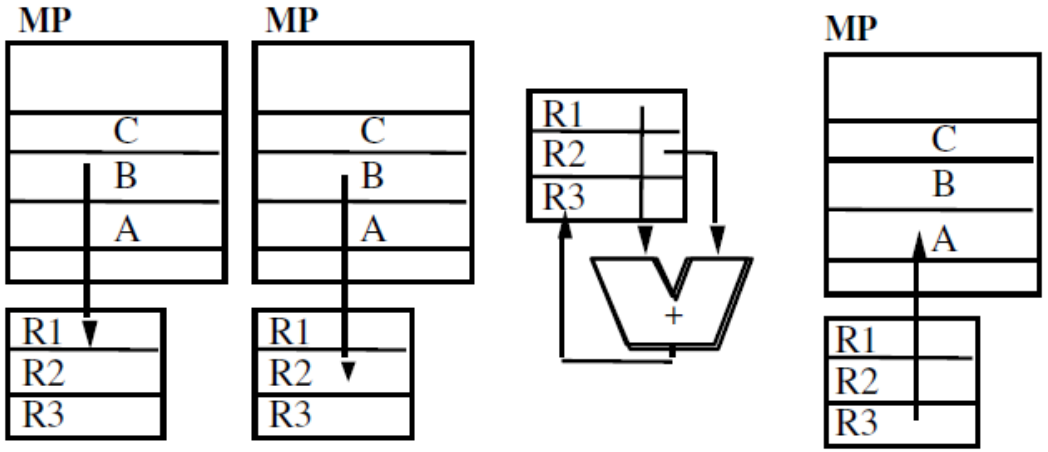
- Registre Registre
  - Aussi appelé "Load/store" architecture
  - 3 opérandes possibles, seulement depuis les registres
  - Opération spécifique "load & store" depuis la mémoire.
- Exemple: C=A+B
  - Load R1,A / Load R2,B / add R3, R1, R2 / store C,R3

Modèles d'exécution (3,0)

- Instruction de longueur fixe
- Seules les instructions load, store accèdent à la mémoire

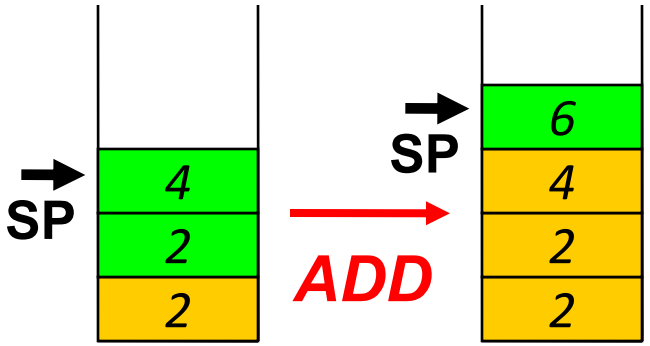
LOAD-STORE (3,0)

Load	R1	@B	
Load	R2	@C	
Add	R3	R2	R1
Store	R3	@A	



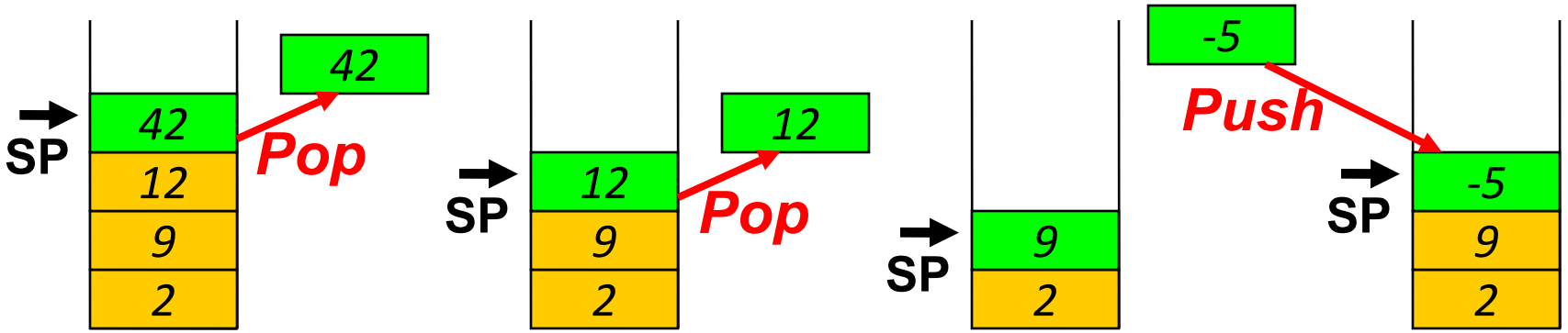
Modèles d'exécution (0,0)

- Pile
  - Toutes les opérandes sont en pile
  - Résultat stocké en pile
  - Problème: la mémoire est lente…
  - Exemple: JVM



Passage par pile

- Données échangées par la pile:
  - Deux opérations: Push et Pop



  - Présent dans toutes les architectures
    - Utilisés pour les adresses et les données
    - Pointeur de pile (registre spécifique)
    - Instructions dédiées

Modèle d'exécution (3,3)

- Mémoire Mémoire
  - Toutes les opérandes en mémoire
  - 3 opérandes par instruction
- Exemple: C=A+B;
  - Add C,A,B
- Avantages/ inconvénients
  - Le plus compact, pas de registres inutilisés
  - Instructions irrégulières format/exécution, mémoire slow
- Exemple: VAX

Modèle d'exécution (2,1)

- Mémoire Registre
  - Un opérande en mémoire, l'autre en registre
  - Exemple: C=A+B
    - Load R1,A; Add R1,B; store C,R1;
  - Avantages/inconvénients
    - Bonne densité
    - Un opérande perdue
  - Intel IA-32, Motorola 68k

Modèle d'exécution (2,1)

REGISTRE-MEMOIRE (2,1)

Load	R1	@B
Add	R1	@C
Store	R1	@A