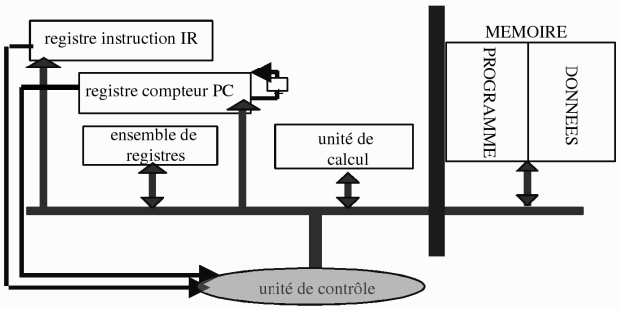
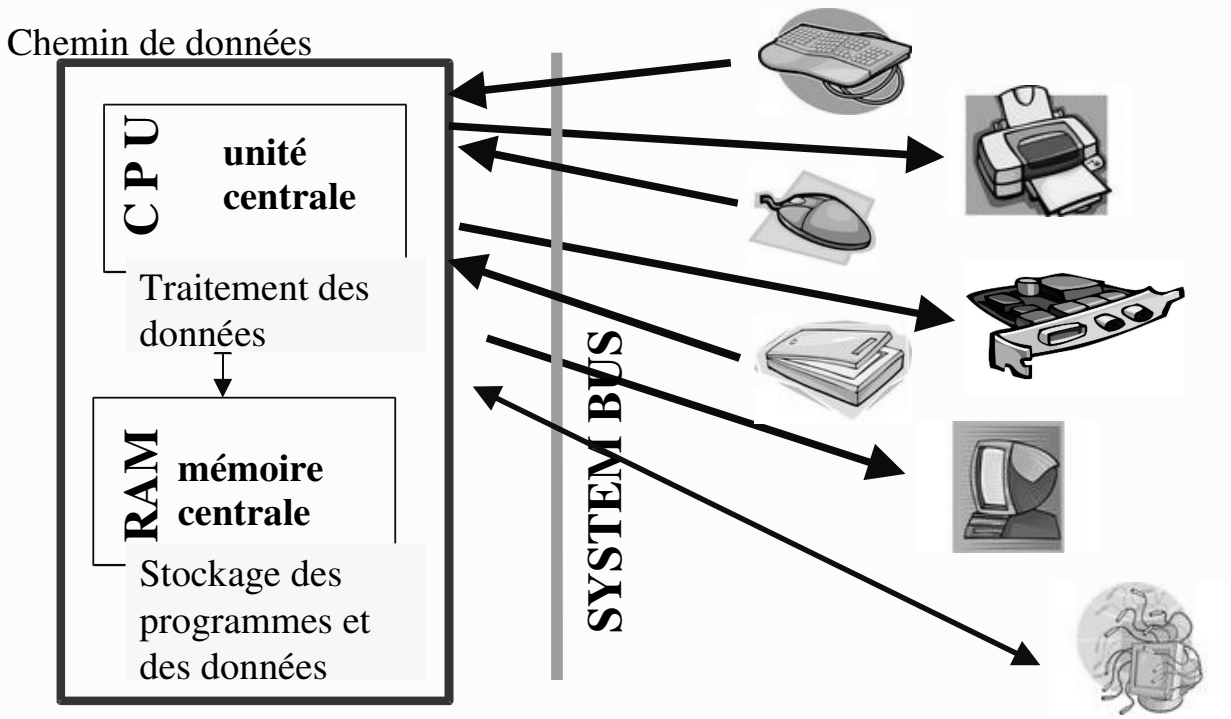
Chemin de données



Les composants qui permettent le stockage, le transfert et la transformation des données.

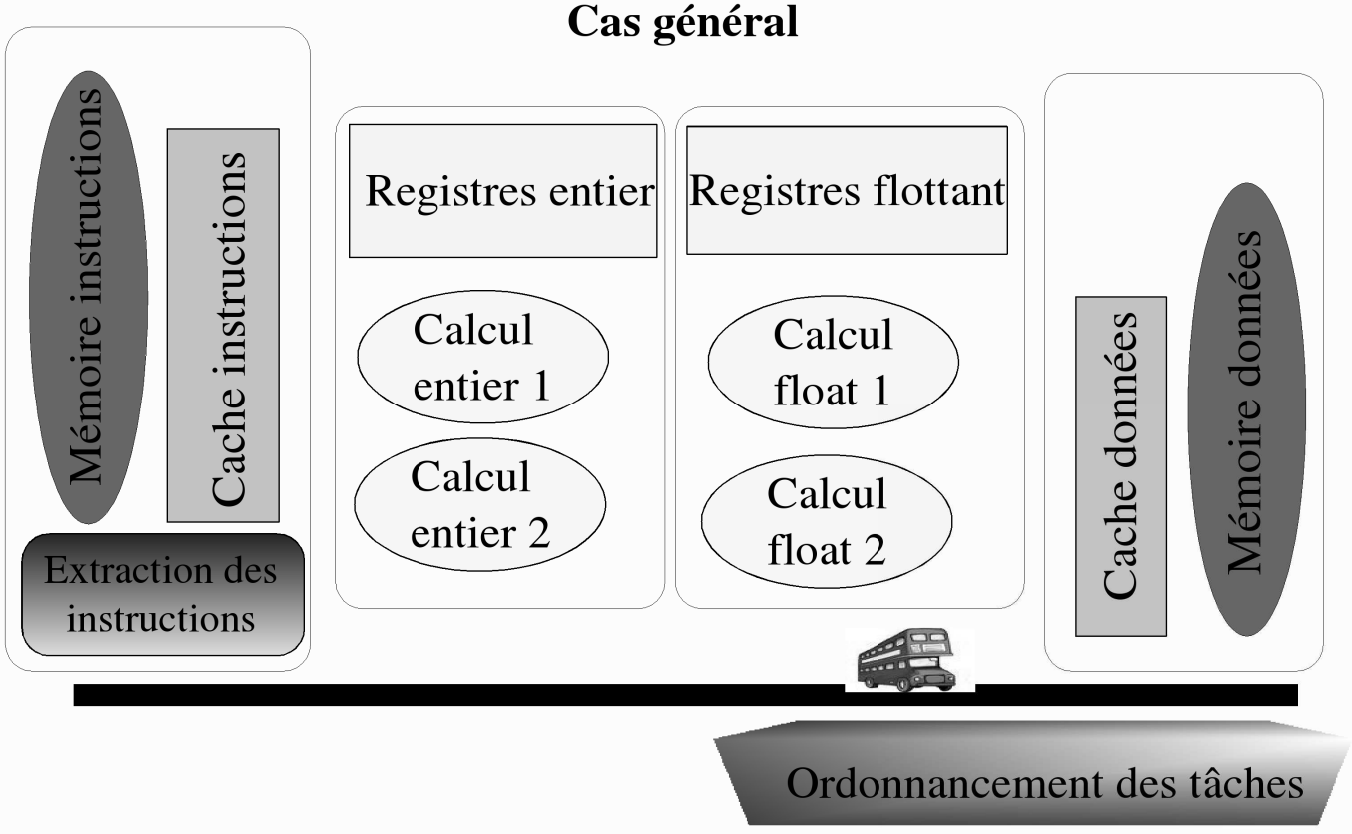
1. Les composants du chemin de données



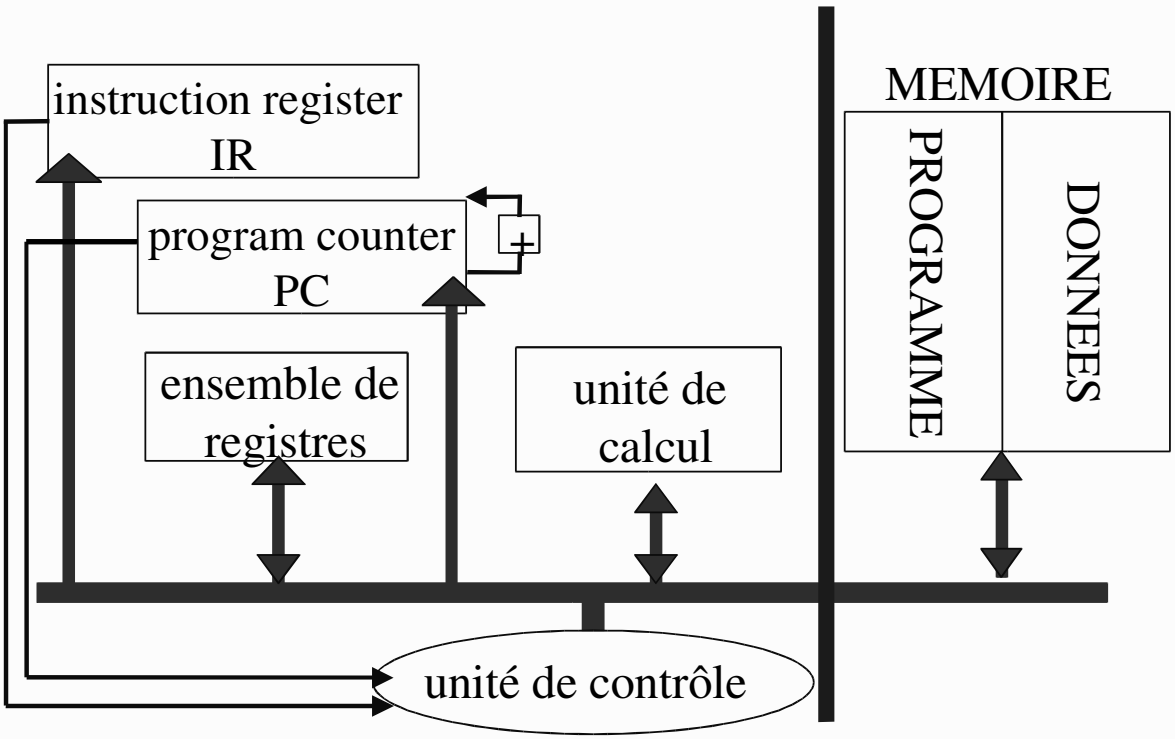
Le chemin de données :

- Les composants qui traitent les données
  - Mémoire centrale : instructions et données des programmes
  - Registres internes :
    - Registres généraux : données temporaires pour les calculs
    - Registres de contrôle et d'état : informations de contrôle nécessaires à l'exécution du programme
  - Unités de calcul
  - Bus
- La façon dont ces composants sont interconnectés

Cas général



Cas étudié



2. Les registres

Les registres sont internes au CPU; ils peuvent être lus/écrits en un cycle processeur (fréquence du CPU)

- Registres généraux : pour les calculs intermédiaires, utilisés comme variables des programmes assembleur
- Registres de contrôle ou d'état : ont un rôle bien spécifique, géré par la partie contrôle

2.1. Registres généraux

Utilisés par le programmeur / compilateur pour stocker des données ou des adresses mémoire

- dans une mémoire statique à accès rapide et petite capacité
- chaque registre peut être lu/écrit en un cycle processeur (fréquence du CPU)
- généraux mais liés à leur type (entier, flottant)

Combien de registres ?

- beaucoup -> moins d'accès mémoire
- MAIS codage du numéro de registre sur plus de bits (-> code binaire plus gros)

  en général entre 16 et 64 registres (16 MC68000, 136 SPARC, 16 ARM)

De quelle taille ?

- Registres d'adresse : taille des adresses mémoire
- Registres de données : taille du bus interne (et entrée des unités de calcul)

2.2. Registres de contrôle et d'état

Registres du chemin de données non modifiables par le programmeur (sauf par instructions spécialisées);
servent à gérer le contrôle de l'exécuton du programme

- PC: *Program Counter*
  adresse de l'instruction à exécuter
- IR: *Instruction Register*
  premier mot de l'instruction : contient la description de l'instruction
- SR: *Status Register*
  code condition: informations sur le résultat des calculs (C retenue, Z résultat nul, N résultat négatif, O overflow, ...)
  + masque d'interruption
- SP: *Stack Pointer*
  adresse de la pile
- registres internes en entrée/sortie des unités de calcul et en entrée/sortie mémoire (stockage intermédiaire de données)

Chemin de données

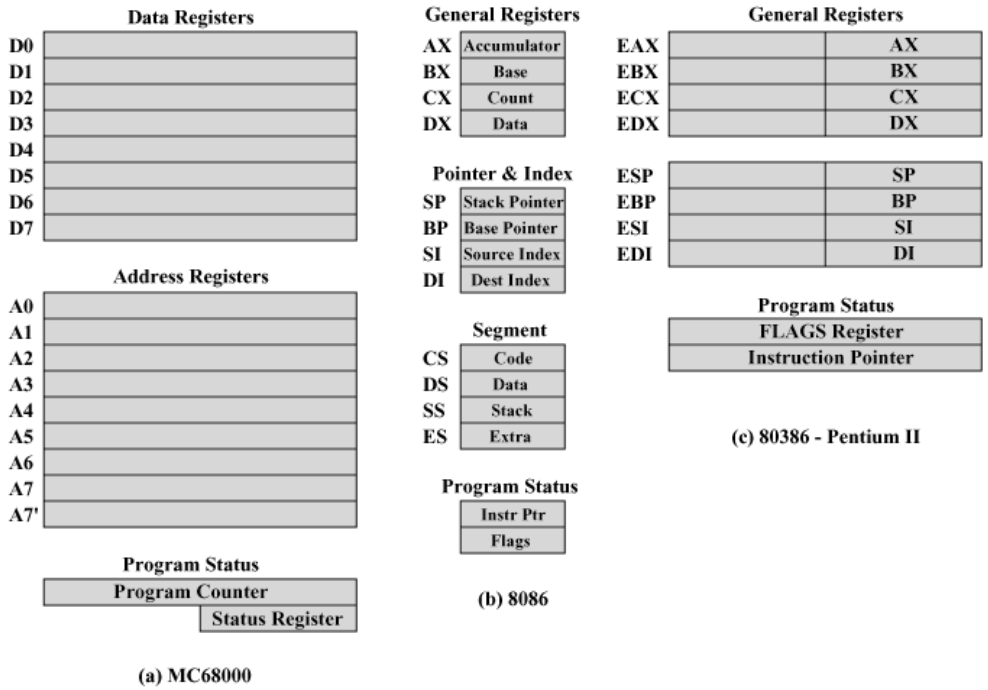


Figure 11.3 Example Microprocessor Register Organizations

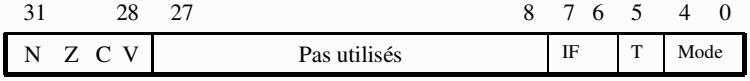
©W. Stallings « Computer Organization and Architecture »

Chemin de données

R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15

- 16 Registres de 32 bits
- R0-R10, R12 : généraux
- R11 (fp) Frame pointer
- R13 (sp) Stack pointer
- R14 (lr) Link register
- R15 (pc) Program counter

- Current Program Status Register

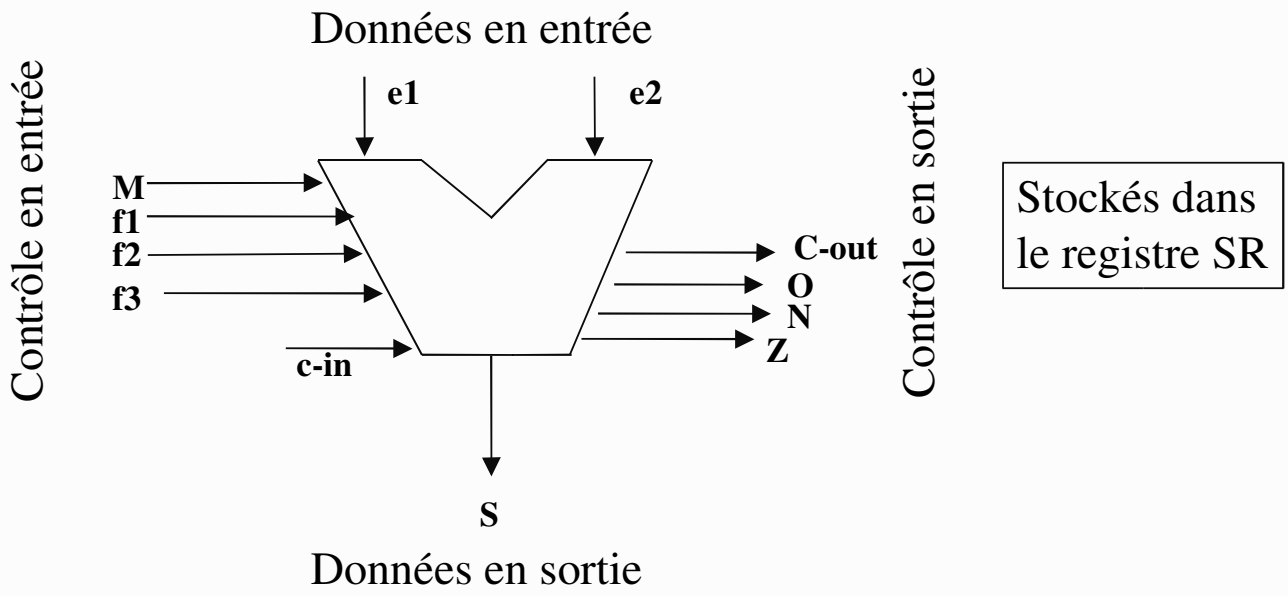


Les registres du processeur ARM © O. Meste « Cours assembleur ARM »

3. Unités de calcul

Arithmetic and Logic Unit (ALU)

Circuit permettant de réaliser ≠ calculs selon le code fourni par la partie contrôle



Exemple PROCSI :

page 1

Mf1f2f3 = 1101	*addition*
e1 = 0110000000110000	*24624*
e2 = 0011111011110000	*16112*

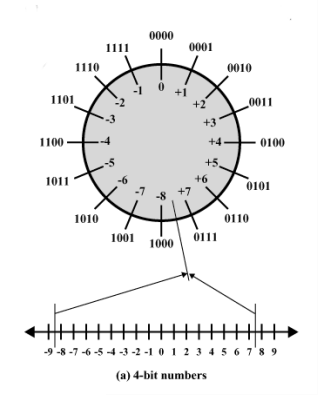
Résultats :

- données : 1001111100100000 - *24800*
- contrôle :
  - C-out = 0 *pas de retenue*
  - O = 1 *addition de 2 nb >0, résultat <0*
  - N = 1 *le résultat est négatif*
  - Z = 0 *le résultat n'est pas nul*

Différentes unités de calcul en fonction du type de données:

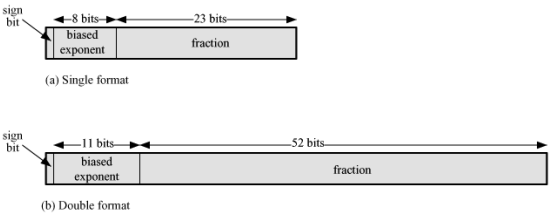
- entiers : codés en complément à 2

$$A = - 2^{n-1} a_{n-1} + \Sigma_{i=0}^{n-2} a_i 2^i$$



(a) 4-bit numbers

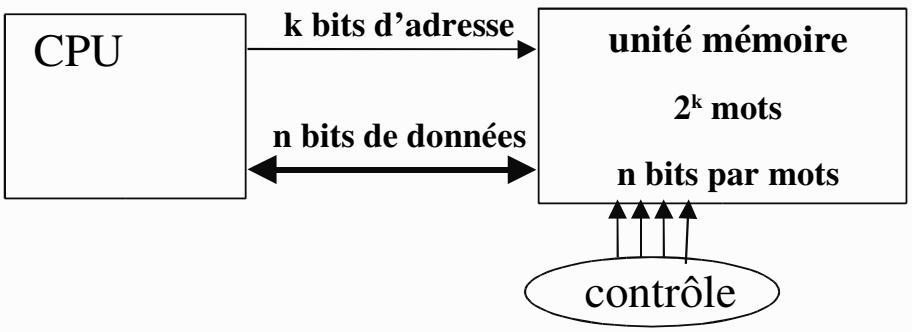
- flottants :



©W. Stallings « Computer Organization and Architecture »

Figure 8.21 IEEE 754 Formats

4. Mémoire centrale



- **Mémoire dynamique à accès lent et grande capacité DRAM**
- **Le temps de cycle mémoire est plus grand que le temps de cycle processeur**
  → **mot mémoire centrale lu / écrit en <u>plusieurs</u> cycles processeur**
  Pentium 4 : CPU ~ 3 Ghz, system bus 400 Mhz

Le processeur et sa mémoire communiquent selon un protocole :

- ***modèle synchrone***
  - **le processeur met sur les broches les signaux pour activer un accès mémoire**
  - **au bout de x tops il sait que la donnée est disponible ou écrite.**
- ***modèle asynchrone (contrôlé)***
  - **le processeur demande un accès mémoire**
  - **la mémoire signale que l'accès est fini**
  - **le processeur scrute les signaux mémoire pour savoir si l'accès mémoire est terminé**

Modèle synchrone

Lecture

- **positionner l'adresse de l'accès**
- **positionner les signaux de contrôle pour activer une lecture**

  → après x tops d'horloge processeur, la donnée est disponible

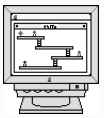
Ecriture

- **positionner l'adresse de l'accès**
- **positionner la donnée à écrire**
- **positionner les signaux de contrôle pour activer une écriture**

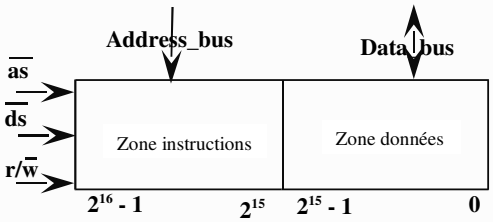
  → après x tops d'horloge processeur, la donnée a été écrite en mémoire

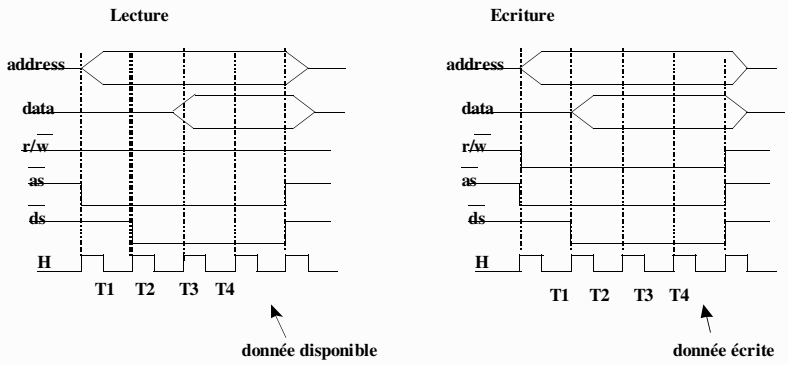
Notation : MEM[ad] le mot de la mémoire MEM à l'adresse ad

Exemple PROCSI



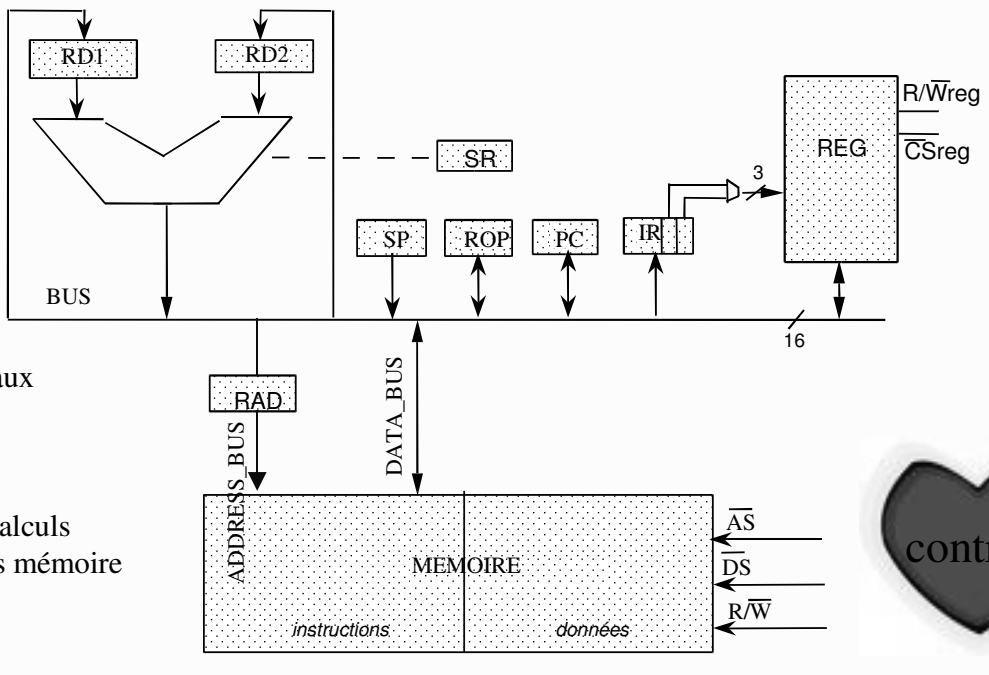
page 2





page 3

5. Le chemin de données étudié : PROCSI



REG : 8 registres généraux

Registres spécialisés :
PC, IR, SR, SP
RD1 et RD2 : pour les calculs
RAD : adresse de l'accès mémoire
ROP : registre opérande