



Architecture

Chapitre 3

Décodage des Instructions



Plan

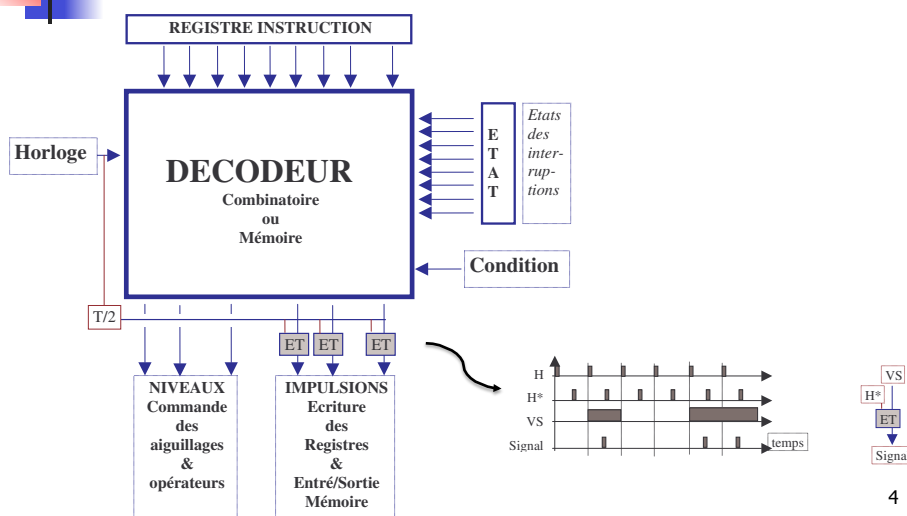
- Principe
- Décodeur « câblé »
- Décodeur « microprogrammé »
- Simulateur

Principe du décodage

- C'est l'opération qui consiste à produire les signaux nécessaires à la réalisation de chacune des instructions. Elle est effectuée par le Décodeur ou Séquenceur d'instruction.

3

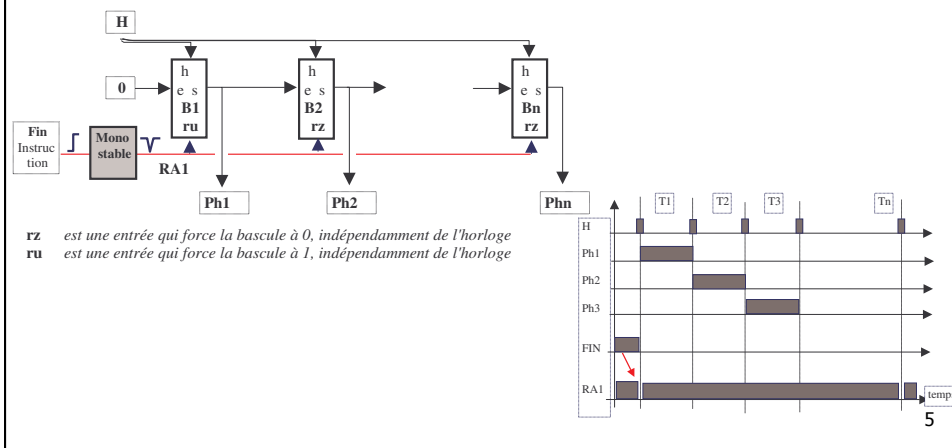
Schéma de Principe



4

Décodeur « Câblé »

Marquage du temps : Compteur de Phase



Pour chaque signal, on cherche pour quelle instruction, et à quelle date (Phi) il doit être vrai. (Phi est vrai pendant la ième période d'horloge).

| ABCD | CodeOp |
|--------|----------|
| 0000 | NoOp |
| 1000 | Chargt A |
| 0100 | Rangt A |
| 1100 | BrIncond |
| etc... | |

| EF | ModeAdr |
|--------|----------|
| 00 | Immédiat |
| 10 | Direct |
| 01 | Indirect |
| 11 | Indexé |
| etc... | |

Registre Instruction

Format des Instructions
ABCD EF xxxxxxxxxxxx
 CodeOp MA suite

| Instruction _ABCDEF | NoOP 0000 00 | ChargtA - AdrImm 1000 00 | ChargtA - AdrDir 1000 10 | etc... etc... |
|------------------------|------------------|-----------------------------|-----------------------------|------------------|
| PH1 | COB1,XS,eRAM | COB1,XS,eRAM | COB1,XS,eRAM | etc... |
| PH2 | sM | sM | sM | etc... |
| PH3 | REB1,XS,eRI | REB1,XS,eRI | REB1,XS,eRI | etc... |
| PH4 | COB1,XP1,eCO,Fin | RIB1,XS,EA | RIB1,XS,eRAM | |
| PH5 | | COB1,XP1,eCO,Fin | sM | |
| PH6 | | | REB1,XS,eA | |
| PH7 | | | COB1,XP1,eCO,Fin | |
| etc.... | | | | |

Equations logiques des signaux

$$COB1 = PH1 + PH4.A\bar{B}\bar{C}\bar{D}\bar{E}\bar{F} + PH5.A\bar{B}\bar{C}\bar{D}\bar{E}\bar{F} + PH7.A\bar{B}\bar{C}\bar{D}\bar{E}\bar{F} + \text{etc.....}$$

$$eRAM = PH1 + PH4.A\bar{B}\bar{C}\bar{D}\bar{E}\bar{F} + \text{etc.....}$$

etc...



Décodeur « Câblé »

- Avantages
 - Rapidité car créé spécifiquement

- Inconvénients
 - « Difficulté » de mise au point
 - Flexibilité

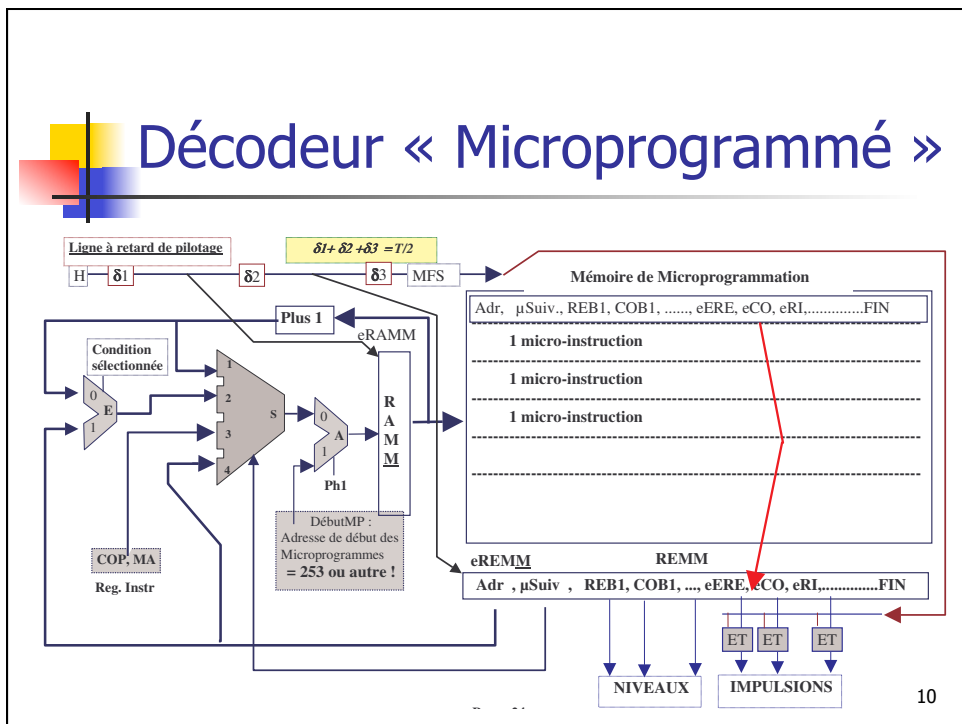
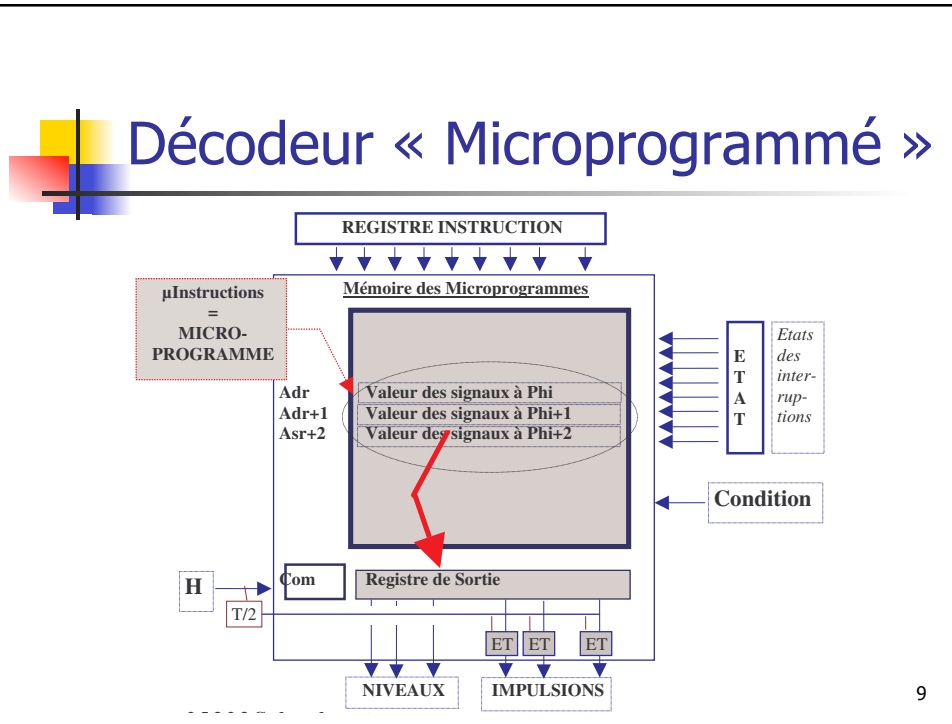
7



Décodeur « Microprogrammé »

- Principe
 - Une mémoire contient, pour chaque instruction et pour chaque phase de l'instruction, un mot dans lequel est codé les signaux à appliquer
 - Un nouveau mot est lu à chaque période d'horloge et sa sortie sur les lignes produit les signaux

8



Exemple de microprogramme

LOAD A, Immédiat, RA COP, MA = 1

Ph1
 Ph2 Cycle de recherche de l'instruction
 Ph3
 Ph4 RIB,1 XS, eA
 Ph5 COB, XP1, eCO, FIN

JUMP Direct, RA COP, MA = 2

Ph1
 Ph2 Cycle de recherche de l'instruction
 Ph3
 Ph4 RIB1, XS, eCO, FIN

JUMPC Direct, RA COP, MA = 3

Ph1
 Ph2 Cycle de recherche de l'instruction
 Ph3

Si la Condition sélectionnée est VRAIE
 Ph4 RIB1, XS, eCO, FIN

Si la Condition sélectionnée est FAUSSE
 Ph4 COB1, XS, eCO, FIN

11

Hypothèse: le jeu d'instruction a 65 instructions seulement, sinon la zone de début serait plus longue...

| | Adresse | Séquence | μS | FIN | REB1 | RIB1 | COB1 | XS | XP1 | ... | ... | eRA | eCO | eRI | eRAM | sM | ... | Commentaire |
|---|-------------------------------------|----------|----|-----|------|------|------|----|-----|-----|-----|-----|-----|-----|------|----|-----|---|
| Début de Zone de Début des Instructions | 1 | 4 | 66 | | | 1 | 1 | | | | | 1 | | | | | | Chargement A - Immédiat - RIB1, XS, eRA |
| | 2 | x | x | 1 | 1 | 1 | | | | | | 1 | | | | | | Branch Incond Direct - RIB1, XS, eCO, FIN |
| | 3 | 4 | 67 | | | | | | | | | | | | | | | Branch Cond Direct - Saut Obligatoire |
| | ... | | | | | | | | | | | | | | | | | |
| | ... | | | | | | | | | | | | | | | | | |
| Fin de Zone de Début des Instructions | 64 | | | | | | | | | | | | | | | | | |
| Début Zone de suite des instructions | 65 | | | | | | | | | | | | | | | | | |
| | 66 | x | x | 1 | | 1 | 1 | | | | | 1 | | | | | | COB1, XP1, eCO, FIN |
| | 67 | 2 | 69 | | | | | | | | | | | | | | | saut conditionnel |
| | 68 | x | x | 1 | | 1 | 1 | | | | | 1 | | | | | | COB1, XP1, eCO, FIN |
| | 69 | x | x | 1 | | 1 | 1 | | | | | 1 | | | | | | RIB1, XS, eCO, FIN |
| 70 | | | | | | | | | | | | | | | | | | |
| Fin Zone de suite des instructions | 252 | | | | | | | | | | | | | | | | | |
| | Fetch | 253 | 1 | x | | | 1 | 1 | | | | | | 1 | | | | COB1, XS, eRAM |
| | Cycle de recherche de l'instruction | 254 | 1 | x | | | | | | | | | | | | 1 | | sM |
| | | 255 | 3 | x | | 1 | | 1 | | | | | 1 | | | | | REB1, XS, eRI |

12



Décodeur « Microprogrammé »

- Avantages
 - Flexibilité

- Inconvénients
 - Surface
 - Performance

13

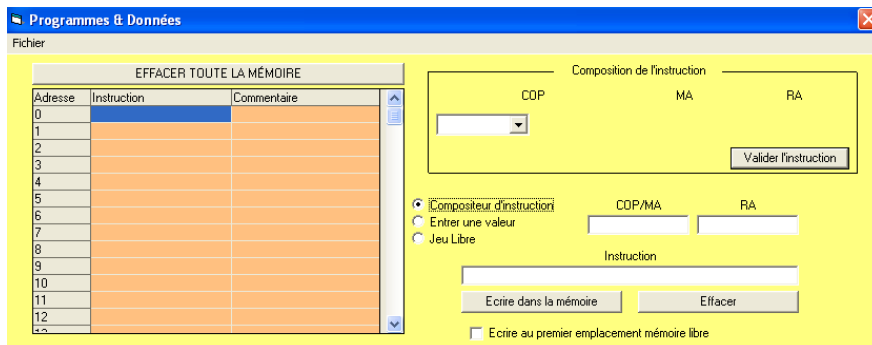


Simulateur

- Emulateur d'architecture
 - Mémoire Programme et donnée avec compositeur d'instruction
 - Unité de traitement : Architecture 3 bus
 - Séquenceur : Décodeur d'instruction microprogrammé

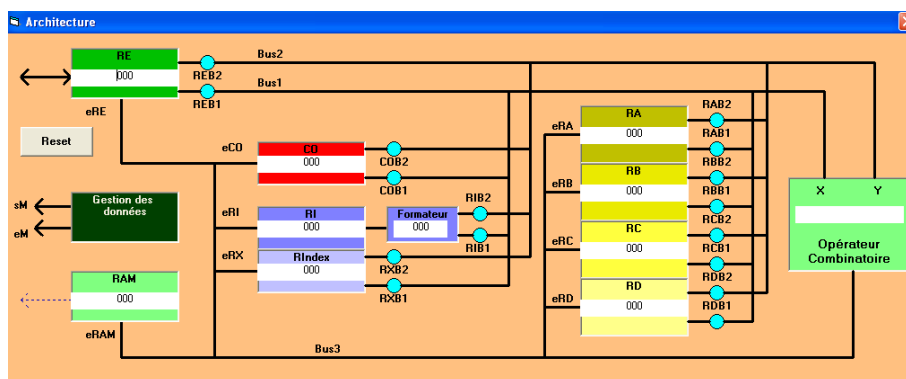
14

Mémoire Programme/Data



15

Unité de Traitement



16

Séquenceur

Séquenceur
Fichier Mode de fonctionnement Complément

Mode de fonctionnement sélectionné
 Sélectionnez un Mode de fonctionnement dans le menu.

Mémoire des Microprogrammes
 En : MicroInstruction :

| Adress | AdiSuiv | SelMS | Cond | FIN | ieM | seM | Stop | COB1 | ROB1 | Rv |
|--------|---------|-------|------|-----|-----|-----|------|------|------|----|
| 0 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ~ | ----- | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ | ~ |

Registre d'Echange de la Mémoire de Microprogrammation

The diagram illustrates the internal logic of the sequencer. It features several registers: Aig3 (containing 000), Aig2 (containing 1, 2, 3, 4), Aig1 (containing 0), and Aig4 (containing 498). Control signals include 'Condition: 0', 'False', 'Plus 1', 'Phase1', and 'Phase'. A 'Registre-Adresse de la mémoire de microprogrammation' is shown with the value 0000. The table on the right lists microinstructions with columns for address, next address, selection, condition, finish, memory enable, stop, and other control bits.