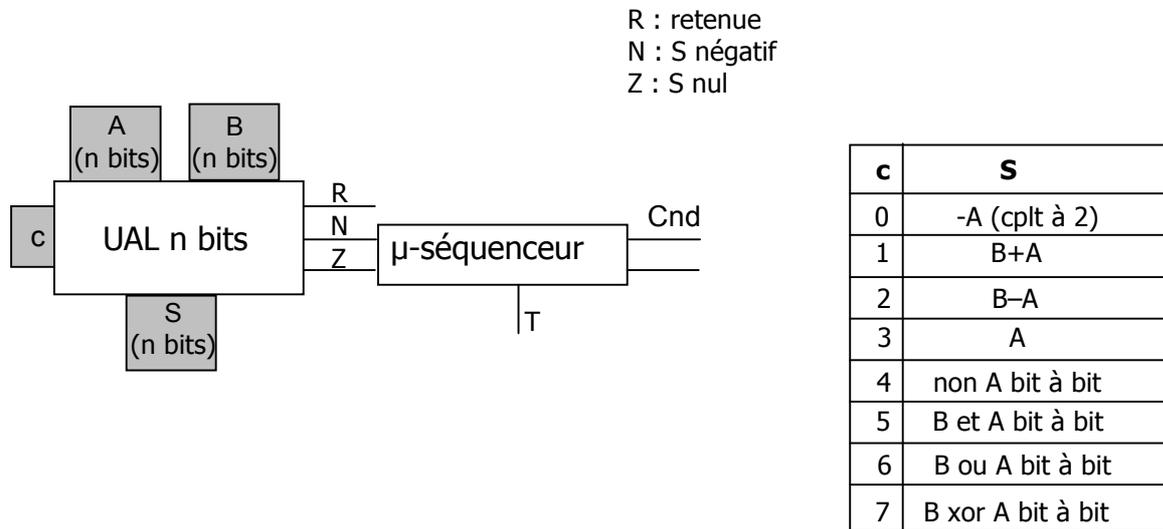


## TD n°1

### Exercice 1 : Pilotage de l'UAL et du micro-séquenceur

Soient l'UAL et le  $\mu$ -séquenceur suivants :



On dispose par ailleurs de 2 registres  $n$  bits  $a$  et  $b$ .

- 1.1. Comment doit-on piloter l'UAL et le micro-séquenceur (connexions (A,B)  $\leftarrow$  (a,b), valeurs de  $c$ , Cnd) pour que la valeur de vérité T corresponde aux tests suivants ?
  - 1.1.1.  $a == b$
  - 1.1.2.  $a < b$
  - 1.1.3.  $a > b$
  - 1.1.4. Vrai systématiquement
  - 1.1.5. Faux systématiquement
  
- 1.2. Les tests suivants sont-ils réalisables directement ? Sinon, que suggérez-vous ?
  - 1.2.1.  $a != b$
  - 1.2.2.  $a >= b$
  - 1.2.3.  $a <= b$

### Exercice 2 : Langage de micro-assemblage (LMA)

2.1. Syntaxe  
Cf. feuille annexe.

2.2. Valeurs binaires des micro-instructions

2.2.1. Exemples

Donner la valeur en binaire des  $\mu$ -instructions suivantes :

- 2.2.1.1.  $A = RI \ \& \ 1$
- 2.2.1.2.  $RES = (OP1 + RDO) \ll 1$  ; if N goto 12
- 2.2.1.3.  $RD$  ;
- 2.2.1.4.  $RDO = (RAD = PP) + RDO$  ; WR
- 2.2.1.5.  $RDO = F = F \ \& \ A$  ; if Z goto 10 ; WR

### 2.2.2. Généralisation

Donner, pour chaque forme d'élément syntaxique du langage, la valeur de chaque case de la micro-instruction impliquée dans la mise en œuvre de cet élément.

		32	31	30	29	28	27	26	25	24	23	22	21	20	19	...	16	15	...	12	11	...	8	7	6	...	1	0			
		A	MS	U			D	SA	SO	WR	RD	EC	Adresse C			Adresse A			Adresse B			Adresse de saut de μ-inst									
<mémoire>	RD																														
	WR																														
	absent																														
<saut>	goto <μi>																														
	if N goto <μi>																														
	if Z goto <μi>																														
	absent																														
RDO =	présent																														
	absent																														
<reg> =	présent																														
	absent																														
<calcul>	<a>																														
	<b> + <a>																														
	<b> - <a>																														
	<a>																														
	~<a>																														
	<b> & <a>																														
	<b>   <a>																														
	<b> ^ <a>																														
	absent																														
	<décalage>	<< 1																													
>> 1																															
>>= 1																															
absent																															
<b>	RAD = <reg>																														
	<reg>																														
	absent																														
<a>	<reg>																														
	RDO																														
	absent																														

### 2.2.3. Vérification

A l'aide de la table construite en 2.2.2, redonner les valeurs binaires des micro-instructions du 2.2.1, puis comparer les résultats.

### 2.3. Petits exercices

2.3.1. Ecrire en LMA l'opération suivante :  $A = (C \text{ ET } D) * 2$  – avec ET : l'opérateur ET logique et \* : l'opérateur de multiplication (cf. cours).

2.3.2. Ecrire en LMA :  $Y = X + 10$  (cf. cours)