

Série de TP N° 4

1. Découvrir les différentes connexions des bascules JK et T prédéfinie dans Logisim (**Dossier memory**).

2. Simuler le fonctionnement des compteurs synchrone et asynchrone traités dans la série de TD numéro 7. Chercher leurs équivalents qui sont prédéfinis dans Logisim (**Dossier memory**). Ensuite découvrir les différentes connexions d'un compteur.

3. A l'aide d'un compteur prédéfini dans Logisim, donner le logigramme correspondant à la boucle suivante :

for (unsigned short int i = 0 ; i < 65536 ; i++)

Indiquer le nombre de bits (nombre de bascules JK utilisées), le type de cycle et le modulo de ce compteur. Ajouter un comparateur qui contrôlera l'arrêt du compteur C pour éviter une boucle infinie.

Modifier le logigramme précédent pour simuler la boucle suivante :

for (unsigned short int i = 0 ; i < 100 ; i++)

en faisant fonctionner le compteur à cycle incomplet. Indiquer le modulo de ce compteur.

4. Modifier le logigramme de la question 3 pour simuler la somme, sur 16 bits, de 1 à 99 premiers:

unsigned short int i, S=0 ;

for (i =1 ; i <100 ; i++)

S = S + i ;

Utiliser un additionneur et un registre R de stockage de la somme S.

5. La division binaire peut être effectuée par le principe des soustractions successives.

Exemple : $A / B = 34 / 7$, $Q(\text{quotient}) = 0$, $R(\text{reste}) = 0$

$A - B = 34 - 7 = 27 \Rightarrow Q = 1$, $A - B = 27 - 7 = 20 \Rightarrow Q = 2$, $A - B = 20 - 7 = 13 \Rightarrow Q = 3$,

$A - B = 13 - 7 = 6 \Rightarrow Q = 4$: Arrêt ($A = 6 < B = 7$) $\Rightarrow Q(\text{quotient}) = 4$, le dernier résultat est le $R(\text{reste}) = 6$.

On remarque que $A = 34$ à la 1^{ère} itération, $A = 27$ à la 2^{ème} itération, ..., $A = 6$. Ainsi, pour une nouvelle itération on a $A - B = A$, donc l'arrêt de l'opération sera effectué quand $A < B$

En utilisant ce principe, donner le logigramme qui permet de matérialiser la division binaire sur 8 bits:

A / B , $A > B$ et $B \neq 0$.

Composants à utiliser:

- à $t = 0$, A est stockée préalablement dans un registre R, ensuite R sera utilisé pour stocker de $A - B$,
- Un additionneur pour effectuer « $A - B$ »,
- Un compteur C,
- Un comparateur pour contrôler l'arrêt du compteur C.

A la fin de l'opération le registre R réservé pour A représentera le reste et le compteur C indiquera le quotient. Indiquer le « Data bits » des composants.

Liaisons : une seule horloge CK et un minimum de portes logiques.

6. La suite de Syracuse d'un nombre entier naturel $N > 0$ est définie par récurrence de la manière suivante :

$$u_0 = N \in \mathbb{N}^* \quad \forall n \in \mathbb{N}, u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } u_n \text{ pair} \\ 3u_n + 1 & \text{si } u_n \text{ impair} \end{cases}$$

Exemple : Suite de Syracuse pour $N = 15$

u_0	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}	u_{13}	u_{14}	u_{15}	u_{16}	u_{17}
15	46	23	70	35	106	53	160	80	40	20	10	5	16	8	4	2	1

Traduire sous forme de logigramme l'algorithme de calcul de la suite de Syracuse pour $N = 15$.

Début

Entier $i \leftarrow 0$

Entier $N \leftarrow 15$

Tant que $N \neq 1$ faire

 si N est pair alors $N \leftarrow N/2$

 sinon $N \leftarrow 3*N + 1$

 Fin si

$i \leftarrow i + 1$

Fin Tant que

Fin

- On supposera que la valeur initiale de la variable N est stockée dans un registre.
- La variable entière i sera représentée par un compteur.
- Une seule horloge CK sera utilisée pour tous les composants séquentiels.
- Indiquer les "data bits" de chaque composant.