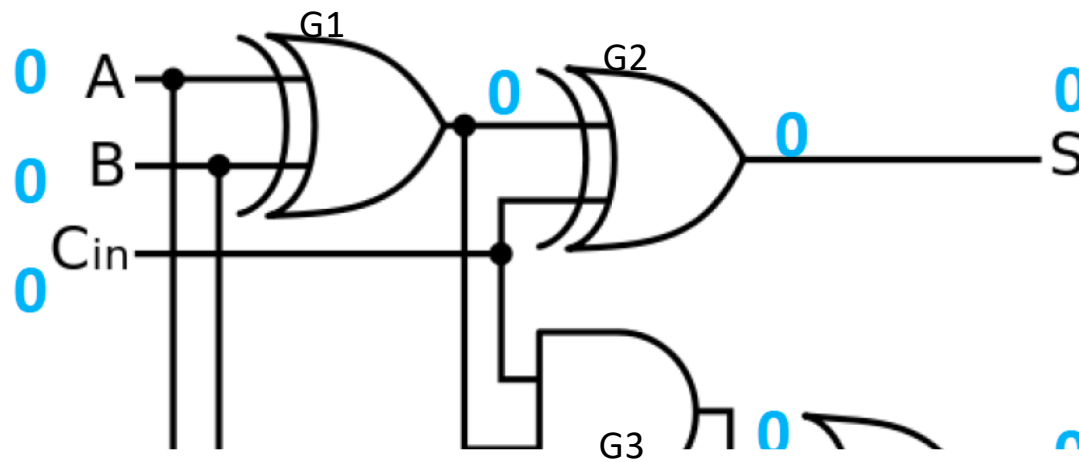


Projet info

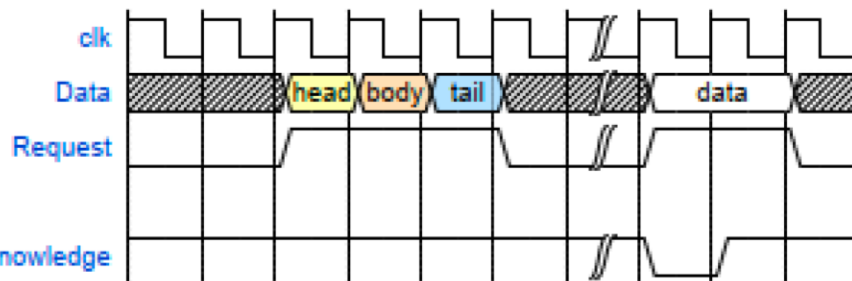
Simulateur de circuit logique



```

1 { signal: [
2   { name: "clk",      wave: "p.....|..." },
3   { name: "Data",     wave: "x.345x|=..x", data: ["head", "body", "tail", "data"]
4   { name: "Request",  wave: "0.1..0|1.0" },
5   {}},
6   { name: "Acknowledge", wave: "1.....|01." }
7 ]}

```



```

Digraph FA{
A[label=« input »];
B[label=« input »];
Cin[label=« input »];
S[label=« output »];
Cout[label=« output »];
G1[label=« xor »];
G2[label=« xor »];
G5[label=« or »];
G3[label=« and »];
G4[label=« and »];

```

```

A->G1->G2->S;
B->G1;
Cin->G2;
G1->G3->G5->Cout;
Cin->G3;
A->G4->G5;
B->GA;
}

```

[Edit Me >>](#)

Organisation:
4 semaines

		s48	S49	S50	S51
lundi	08h00		KMA 1	partiel	KMA
	10h00	KMA	KMA 2		KMA -15
	13h30				
	15h30				
mardi	08h00			partiel	XL- 16
	10h00		XL 3		
	13h30				KMA 17
	15h30				KMA- 18
mercredi	08h00		DA-tutorat?	partiel	XL-DA-20
	10h00			partiel	
	13h30		KMA-XL 5	KMA-DA 9	
	15h30				
jeudi	08h00				DA-21
	10h00				DA-22
	13h30				
	15h30				
vendredi	08h00		KMA-DA 7	KMA-XL 11	XL-DA-24
	10h00				
	13h30			KMA-DA 13	
	15h30				

Organisation

- 3 séances de tutorat
- Des seances de libre service.
- Un rapport à rendre le 4 décembre (ca arrive tres vite)
 - le planning
 - choix de 2 livrables intermédiaires
 - le partage des tâches
 - L'architecture logiciel: algo et protoypte des fonctions

Le format dot

```
digraph graphname {  
    a -> b -> c;  
    b -> d;  
}
```

Le format dot

```
graph graphname {  
    // This attribute applies to the graph itself  
    size="1,1";  
    // The label attribute can be used to change the label of a node  
    a [label="Foo"];  
    // Here, the node shape is changed.  
    b [shape=box];  
    // These edges both have different line properties  
    a -- b -- c [color=blue];  
    b -- d [style=dotted];  
    // [style=invis] hides a node.  
}
```

Création d'un circuit

```
digraph test {  
  
  I1 [label = "INPUT"];  
  I2 [label = "INPUT"];  
  
  GATE [label = "AND2" ];  
  
  O [label = "OUTPUT"];  
  
  I1 -> GATE -> O;  
  I2 -> GATE;  
  
}
```

```
Digraph « mot reserve» L1 C1  
Espace  
Test « eti »  
Espace  
{  
Eol  
Eol  
I1  
Espace  
[
```

Problème: comment lire un fichier dot

- Deux étapes:
 - Découper en lexèmes (en mots)
 - Quels sont les mots valides
 - Quelle structure de donnée pour stocker ses mots
 - Quelles informations stockées pour les mots (le numero de ligne, de colonne)
 - Faire la différence entre les mots clés et les définitions d'étiquettes
 - Vérifier l'ordre des lexèmes (parser)
 - Vérifier l'ordre des mots
 - Renvoyer des messages d'erreurs si necessaire
 - Créer une structure de données pour contenir les informations pertinentes

Le test

- 50% du travail!
- Définir les tests unitaires dès le début, en même temps que le découpage des fonctions
- Automatiser les tests (soit avec des classes prédéfinies, soit en bricolant votre solution)

Comment lire un fichier json

- Même méthode que pour lire un fichier dot
- Définir la structure de donnée pour mémoriser les informations

Le simulateur

- Comment simule t'on un circuit
- Notion de delta cycle
- les circuits combinatoires: vu comme des arbres (pas de cycles)
- Comment gérer les registres

Générer un fichier json

- Fichier résultat qui contient toutes les simulations