|  |
| --- |
| HES-SO |
| Projet C++ |
| Protocole JSON |

|  |
| --- |
| Gilles Mottiez  01/05/2018 |

Ce document décrit l’implémentation des fichier .json qui représentent un circuit de logique numérique :

1. Chaque fichier commence avec l’attribut « name »,
2. Vient ensuite un tableau de portes logiques « gates », qui répertorie toutes les portes du circuit
3. Une porte logique est définie comme telle :
   1. Attribut « id » : chaque porte logique possède un id différent qui permet de la distinguer. Les id sont tels que :
      1. Porte OR : id = OR1, O2, …
      2. Porte AND : id = AND1, AND2, …
      3. Porte NOT : id = NOT1, NOT2, …
   2. Attribut « level » : définit l’enchainement logique des portes :
      1. Une porte de level 0 possède les entrées du système
      2. Une porte de level maximum possède la ou les sorties du système
   3. Tableau « pins » : contient les connecteurs de la porte :
      1. Chaque connecteur a un attribut label tel que : « id » + Numéro du connecteur. Un label de pin input 1 pour l’id OR1 est : « OR1\_I1 »
      2. Un attribut « connected » qui contient le label du connecteur en face
   4. À la fin du tableau se trouve la pin de sortie !!!
4. Les signaux logiques d’entrées sont dénommés comme suit : LOG\_LOW pour un 0 logique, LOG\_HIGH pour un 1 logique
5. Les signaux de sortie des portes logiques sont définis tel que : « id » + « U ». Un bloc avec l’id « OR1 » possède un signal de sortie de label « OR1\_U »

Exemple d’implémentation :

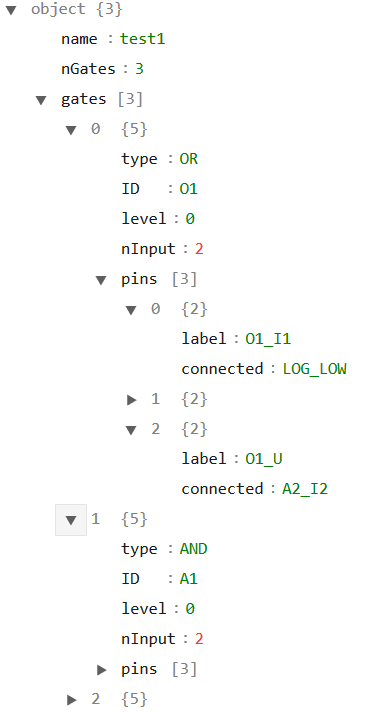


Figure 1 : code JSON

Est la représentation JSON du circuit logique suivant :

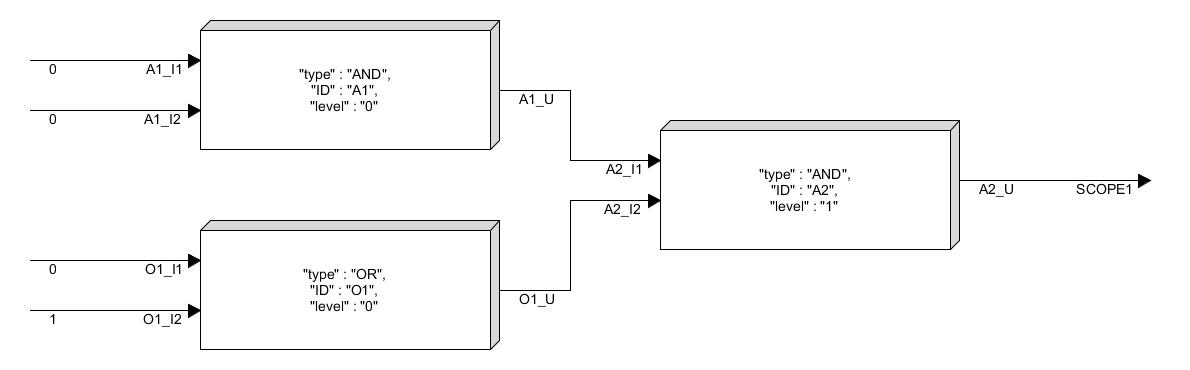


Figure 2 : circuit logique