

1. Fichier de Positions

Fichier de Positions

Le fichier de positions décrit de manière structurée (donc hiérarchique) un composant. Il contient des informations sur les interfaces du modèle, les blocs internes ainsi que leur mapping.

Afin d'aider l'outil de visualisation, il indique sur quel face les interfaces d'un composant doivent être placées. Les ports doivent être dessinés dans l'ordre horaire suivant leur position dans le fichier.

Voici un exemple de fichier de position

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<!-- comment -->

<description>

  <comment>
    Ce composant décrit un prédicteur de branchement à deux niveaux
  </comment>

  <entity name="two_level_branch_predictor"
    type="Two_Level_Branch_Predictor"
    schema="1" >

    <port name="clock"
      localisation="west"
      direction="in">
      Port d'horloge
    </port>
    <port name="predict"
      localisation="west"
      direction="in">
      Interface de prediction : demande de la direction pour une adresse
    </port>
    <port name="branch_complete"
      localisation="east"
      direction="in">
      Interface de mise à jour des branchement
    </port>

  </entity>

  <architecture>

    <component name="branch_history_table"
      type="Branch_History_Table"
      size_x="10"
      pos_x="50"
      size_y="10"
      pos_y="10" />

    <port_map name="clock"
      component="two_level_branch_predictor"
      port="clock" />
    <port_map name="predict"
      component="two_level_branch_predictor"
      port="predict"/>
    <port_map name="branch_complete"
      component="two_level_branch_predictor"
      port="branch_complete" />
```

```

</component>

<component name="pattern_history_table"
           type="Pattern_History_Table"
           size_x="10"
           pos_x="50"
           size_y="10"
           pos_y="90" />

<port_map name="clock"
          component="two_level_branch_predictor"
          port="clock" />
<port_map name="predict"
          component="two_level_branch_predictor"
          port="predict"/>
<port_map name="branch_complete"
          component="two_level_branch_predictor"
          port="branch_complete" />

</component>

<component name="two_level_branch_predictor_glue"
           type="Two_Level_Branch_Predictor_Glue"
           size_x="10"
           pos_x="50"
           size_y="10"
           pos_y="50" />

<port_map name="predict"
          component="two_level_branch_predictor"
          port="predict" />
<port_map name="branch_complete"
          component="two_level_branch_predictor"
          port="branch_complete" />

</component>

</architecture>

<description>

```

- **balise description :** (*obligatoire et unique*) Balise englobante
- **balise comment :** (*obligatoire et unique*) Contient un descriptif du composant décrit.
- **balise entity :** (*obligatoire et unique*) Définition du composant courant
 - ◆ **attribut name :** nom d'instance du composant
 - ◆ **attribut type :** nom du composant
 - ◆ **attribut schema :** figure pour représenté le composant (ovale (purement combinatoire), rectangle avec un triangle (purement séquentielle), rectangle (mixte), fifo (composant fifo).
- **balise port :** (*au moins 1 occurrence*) Interface du composant. Une interface est un groupe de port cohérent entre eux. On peut par exemple cité l'interface Read et l'interface Write du composant fifo. Cette balise contient un descriptif de l'interface.
 - ◆ **attribut name :** Nom de l'interface
 - ◆ **attribut localisation :** Localisation de l'interface sur les bords du composant. Cet attribut peut prendre les valeurs *North*, *South*, *East* et *West*. L'ordre des interfaces à une incidence. Ces dernières sont ordonnée suivant le sens horaire sur la face concerné. C'est à dire que sur la face Nord les interfaces seront placés de gauche droite, face ouest de haut en bas, face sud de droite à gauche et enfin face ouest de bas en haut.
 - ◆ **attribut direction :** Peut prendre deux valeurs : *in* ou *out*. Indique la direction du signal *val* dans l'interface. En d'autre mot, est ce que le composant est l'émetteur ou le récepteur sur cette interface.
- **balise architecture :** (*0 ou 1*) Si le composant contient d'autre, alors l'entité contient cette balise

- **balise component** : (*nombre quelconque d'occurrences*) Sous composant incluse dans le composant courant.
 - ◆ **attribut name** : nom de l'instance du sous composant
 - ◆ **attribut type** : nom du sous composant
 - ◆ **attribut pos_x** : abscisse centre du composant
 - ◆ **attribut pos_y** : coordonnée du centre du composant
 - ◆ **attribut size_x** : dépendant de la figure
 - ◆ **attribut size_y** : dépendant de la figure
- **singleton port_map** : (*au moins 1 occurrence*) mapping entre les interfaces du sous composant et des autres sous composant ou du composant englobant.
 - ◆ **attribut name** : nom de l'interface
 - ◆ **attribut component** : nom du composant avec lequel cet interface est inter-connectée
 - ◆ **attribut port** : nom de l'interface avec lequel cet interface est inter-connectée