

ENSSAT - EII3

TP de Systèmes sur Silicium (SOC)

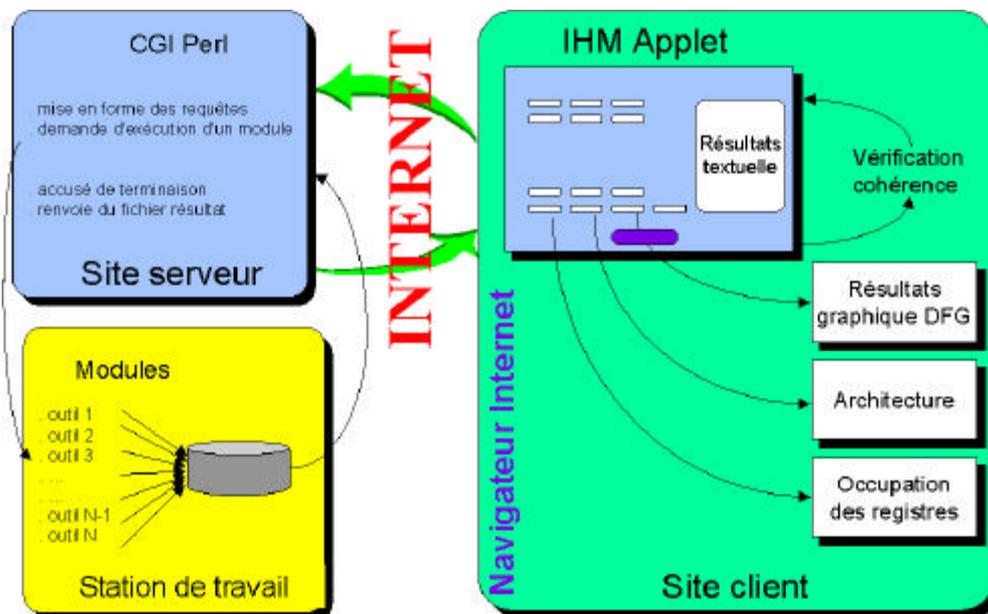
Utilisation de l'outil Breizh Synthesis System

1. Présentation de l'outil

L'équipe R2D2, au travers du serveur web de l'ENSSAT, met à disposition un outil de synthèse de haut-niveau développé au Laboratoire grâce à un accès au travers d'une interface WEB unifiée (Breizh Synthesis System), en permettant une utilisation simplifiée et libre. L'utilisation du langage Java pour l'interface graphique et de CGI pour l'accès aux outils permet de l'utiliser à distance. Nous détaillons ici son principe. L'URL est accessible à partir de <http://r2d2.enssat.fr/bss>.

1.1 Mise en oeuvre et fonctionnement

Cette plate forme est proposée au travers de deux interfaces différentes, mais faisant appel aux mêmes modules de synthèse. La première interface utilise les possibilités offertes par le langage de mise en page html et s'appuie notamment sur les formulaires. La seconde interface est le premier pas vers une refonte complète des outils en langage objet, elle est développée en JAVA sous forme d'une applet et offre donc à ce titre plus de possibilités. Nous avons représenté, sur la figure ci dessous, l'organisation de l'architecture développée autour de l'Internet.



Le site client émet donc des requêtes vers le site serveur. Ces requêtes sont formatées de façon à pouvoir s'interfacer avec les modules de synthèses. Ensuite un appel à l'un de ces modules est réalisé afin d'exécuter la phase de synthèse désirée. Notons que l'exécution du module est réalisée sur une machine autre que celle supportant le serveur ENSSAT.

Le site serveur attend la fin d'exécution du module, puis envoie, à destination du site client, le fichier

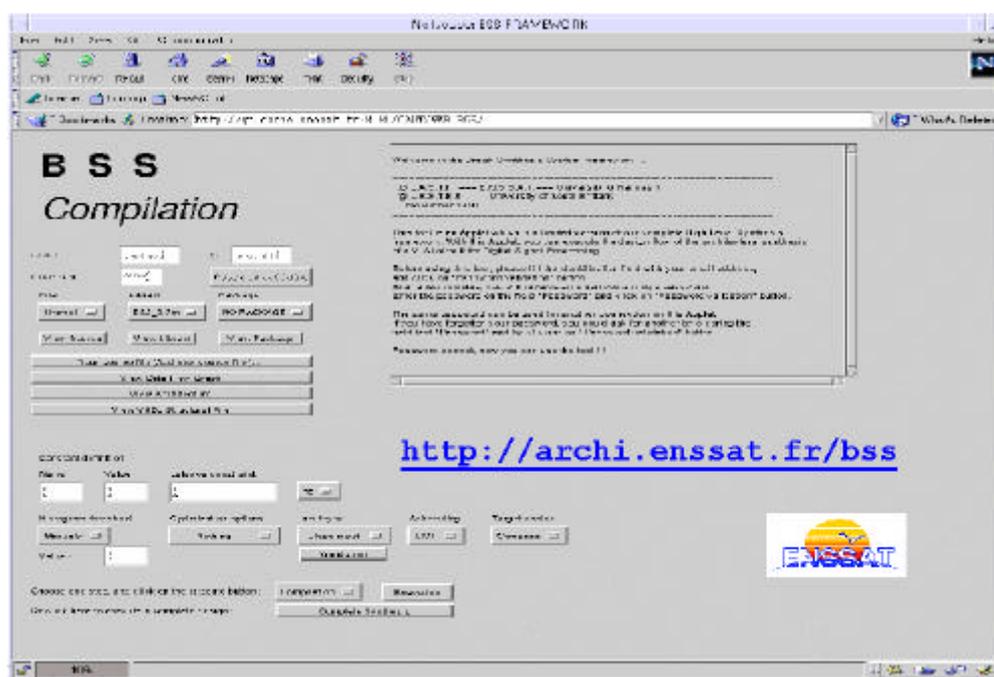
résultat.

1.2 Manipulation de l'outil via l'Internet

Dans l'état actuel, seule la partie synthèse de la partie opérative est disponible. Toutefois, chacun des modules développés autour de cette plate-forme fera l'objet d'une intégration dans notre environnement web.

L'interface est conçue de façon à limiter la saisie des nombreux paramètres disponibles dans l'outil. Afin de faciliter la gestion des fichiers générés par l'outil, une identification est demandée pour pouvoir effectuer toute opération de synthèse. Une identification par email sera un moyen de connaître les utilisateurs et éventuellement de les contacter. **Pour ce TP, vous disposez de comptes tp1 à tp8, avec des mots de passes identiques.**

Les étapes de synthèse (compilation, sélection, ordonnancement, allocation, optimisation, traduction VHDL) sont disponibles en bas de la fenêtre.



L'interface permet, après saisie des paramètres correspondant aux différentes étapes de synthèse, de lancer un certain nombre d'outils et d'obtenir des informations et/ou des résultats sous forme textuelle ou sous forme graphique.

- Résultats textuels : chacune des étapes de synthèse lancée renvoie un compte rendu de synthèse et permet d'obtenir un minimum d'information sur le déroulement de celle-ci : temps de synthèse, fichiers générés, algorithmes utilisés, temps de latence effectif, etc. L'ensemble de ces résultats est fourni dans la fenêtre *Results* en haut à droite de l'applet BSS. De plus, les fichiers source, bibliothèque et paquetage, servant de support à la synthèse, peuvent être visualisés au travers d'une nouvelle fenêtre de type édition de texte (voir la figure a).
- Résultats graphiques : un certain nombre d'informations graphiques sont disponibles tout au long du flot de conception. Il s'agit, par exemple, du graphe flot de données (voir figure b) ou encore des résultats de simulations de l'algorithme sous forme de réponse à une excitation. En phase terminale de conception, c'est à dire avant de synthétiser le circuit à l'aide d'un outil de synthèse logique, des informations concernant la structure et l'occupation de l'architecture peuvent être affichées. Il s'agit entre autres des fenêtres affichant l'architecture proprement dite (voir figure c) ou encore des graphes d'occupation des registres, bus et opérateurs de l'architecture (voir figures d et e).

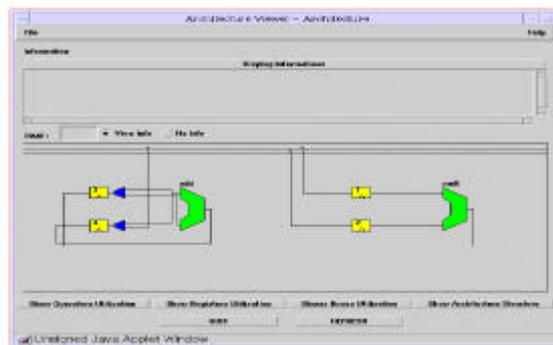
Par limitation des possibilités offertes à l'utilisateur Internet, nous nous assurons d'un masquage partiel de la complexité de l'outil, et ainsi nous simplifions l'accès à ce type d'outil toujours assez lourd à manipuler au travers de leurs nombreuses commandes.

L'architecture de cette solution Internet, bien qu'assez complexe, nous permet de continuer à utiliser notre outil de façon exhaustive en local (c'est à dire au travers de tous ces paramètres), tout en ouvrant ces mêmes modules au monde extérieur. L'insertion des nouveaux modules sera, bien entendu, réalisée au fur et à mesure de leur développement et de leur validation. Ainsi certains modules, notamment un module d'estimation de consommation, devraient venir se greffer aux outils existants dans les mois à venir.

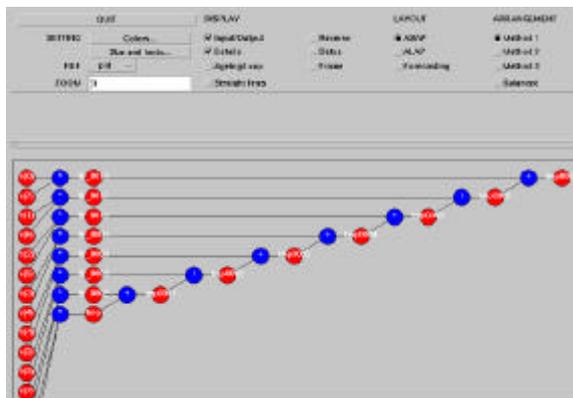
```

Text Viewer (Not editable)
ENTITY fir IS
  GENERIC (lagance : TIME := 2000 ns);
  PORT (x:IN INTEGER, y:OUT INTEGER);
END fir;
ARCHITECTURE fir_comportemental OF fir IS
  CONSTANT N: INTEGER := 8;
  TYPE vecteur IS ARRAY (0 TO N-1) OF INTEGER;
BEGIN
  fir : PROCESS
    VARIABLE i,x:   vecteur;
    VARIABLE temp: INTEGER;
    VARIABLE i:     CONTROL;
  BEGIN
    temp := x(0) * H(N-1);
    FOR i IN 1 TO N-1 LOOP
      temp := temp + x(i) * H(N-i-1);
    END LOOP;
    y(0) := temp;
    FOR i IN N-1 DOWNTO 2 LOOP
      x(i) := x(i-1);
    END LOOP;
    x(1) := x(0);
    WAIT FOR lagance;
  END PROCESS fir;
END fir_comportemental;
    
```

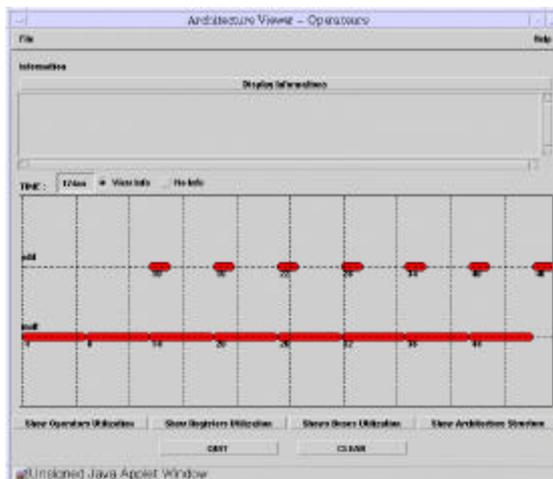
a. Code VHDL source de l'algorithmme



c- Structure de l'architecture synthétisée



b- Exemple d'affichage de graphe flot de données



d- Diagramme d'utilisation des opérateurs



e- Diagramme d'utilisation des registres

1.3 Manipulation de l'outil via UNIX

Il est possible d'utiliser l'outil BSS directement sous unix (c'est d'ailleurs souvent plus rapide). Les étapes de compilation, sélection, ordonnancement, assignation, optimisation et traduction sont accessibles via les commandes suivantes :

```
gcomp fir -l ST_0.25.lib #compilation : creation des fichiers .dfg et .sru
gsel fir #selection : creation du fichier .dfa
gord fir #ordonnancement par liste : creation du fichier .dfs
gass fir #assignation : creation du fichier .archi
gopt fir #optimisation du fichier .archi
grad trad #traduction vhdl : creation du fichier .vhd
```

Des outils graphiques sont disponibles :

```
visugf fir.dfg #visualisation du DFG
visugf fir.dfs #visualisation du DFG ordonnance
visigaut -archi fir.archi #visualisation de l'architecture et du contenu des elements
VisiGaut #visualisation de l'architecture (ecrit en java)
```

1. Travail à réaliser

Vous pouvez ici reprendre l'exemple du fir16 ou utiliser d'autres applications fournies.

- Connectez vous à l'URL puis entrez tp1 dans les champs *Email* et *Password* et validez. Evidemment, chaque binôme n'utilisera pas le même login!
- Visualisez quelques fichiers sources ou bibliothèques de composants. Vous pouvez également définir votre propre fichier source.
- Exécutez la [compilation](#). Visualisez le graphe flot de données en ASAP et ALAP.
- Exécutez la [sélection](#) puis l'[ordonnancement](#). Visualisez le graphe flot de données en *Forecasting* correspondant au résultat de cette étape.
- Exécutez l'[allocation](#) puis l'[optimisation](#). Visualisez l'[architecture](#) conçue.
- [Traduisez](#) l'architecture en VHDL de niveau RT et visualisez ce fichier.

**A partir de quelle valeur de contrainte temporelle va-t-on utiliser 2 multiplieurs ?
Faire la synthèse dans ce cas.**