FAQ sur le travail VHDL

Programmation d'une GAL22V10 et réalisation d'un circuit électronique

Le point crucial de ce travail est le lien étroit existant entre votre code VHDL et les ressources hardware nécessaires à son implémentation dans la GAL. Ainsi, une bonne compréhension de la datasheet de la GAL22V10 est de première importance.

Avoir un schéma général (schéma-bloc, et même le schéma électrique avec cablage pin-à-pin idéalement) du système global est très utile.

Questions sur le code VHDL

Commentez votre code VHDL, sauvegardez les différentes versions dans des fichiers séparés.

a) Message d'erreur type « design does not fit » ou « need more pins » : soit le code tel quel requiert plus de ressources (flip-flop et/ou sortie combinatoire), soit la machine d'états a été encodée de manière « non-optimale selon un critère d'espace ».
Solution : revoyez votre code, tentez de simplifier ou de coder autrement, utilissez d'autres composants externes (compteur, autre GAL, etc…). Vous pouvez forcer l'encodage de la machine d'états à l'aide des commandes suivantes :

```
-- définir les états possibles
type etat12 is (s0,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11);
-- forcer l'encodage dit séquentiel : 0000, 0001, 0010,...
attribute syn_encoding : string;
attribute syn_encoding of etat12 : type is "sequential";
-- definer le registre
signal state : etat12;
-- forcer la préservation de la machine d'états tel que définie -
-- ci-dessus
attribute syn_preserve : boolean;
attribute syn_preserve of state : signal is true;
```

b) Message d'erreur type « too many product terms » : le report file permet de visualiser les équations de sortie qui sont trop complexes pour la GAL (les équations de sortie peuvent contenir entre 8 et 16 termes au maximum).
Solution : simplifiez les équations, i.e. les tests logiques de type OR ! Par exemple, les comparaisons, les opérations arithmétiques (addition, soustraction,…) sont gourmandes en ressources hardware. Utilisez des composants externes pour effectuer ce genre d'opérations ou de la mémoire interne à la GAL pour retenir des résultats intermédiaires.

c) Message d'erreur type « multiple drivers » : un ou plusieurs FF sont déclenchés par plus d'une horloge. Cette erreur survient si un signal/sortie associé à un FF est affecté dans 2 process différents.
   Solution : assurez-vous que lors du reset (synchrone ou asynchrone), tous les FF (signaux/sorties) sont initialisés. De même, veillez à bien séparer la partie séquentielle (machines d'états, mémoires, tous les signaux testés **et** affectés,…) de la partie combinatoire (relations entrées-sorties non synchronisé à l'horloge, il ne peut y avoir de signaux/sorties testées et affectées dans ce genre de process !).

d) Simulation impossible à lancer (erreur type « verilog ») : l'origine de cette erreur est encore inconnue.
   Solution : changer de PC

e) Simulation non-pertinente : les signaux ne réagissent pas comme attendu.
   Solution : vérifiez le résultat de simulation tout en parcourant votre code. Remarquez qu'il n'est pas nécessaire de simuler le code en situation réelle (e.g. cas d'une horloge basse fréquence ~Hz).

Questions sur la réalisation du circuit électronique

Lisez attentivement les datasheet des composants que vous utilisez. Vous y trouverez les attributions des pins, les tables de vérité,… Quant à la datasheet de la GAL, elle contient des informations particulières sur les entrées, les macrocellules de sortie, le reset automatique,…

Il n'y a qu'une seule question qui revient souvent : « Monsieur, ça ne marche pas ! ». Dans l'ordre, veillez à

a) simuler le comportement de la GAL, que celui-ci corresponde bien à vos attentes.
b) vérifier à l'oscilloscope ou au multimètre l'horloge (onde carrée variant entre 0 et 5V), les entrées, l'alimentation (5V) et bien sûr les sorties de chaque composant, si possible séparément.
c) placer des condensateurs de découplage (10 ou 100nF) entre le Vcc et la masse de chaque composant ou d'un ensemble de composants. Ces condensateurs doivent être le plus proche possible du/des composants(s).
d) utiliser une plaque pré-trouée (et des sockets pour remplacer/programmer facilement les composants) au lieu d'une breadboard afin d'éviter de mauvais contacts, des court-circuits,... Souder n'est pas chose aisée pour un débutant, mais avec un peu de temps et de patience, vous finirez par obtenir un circuit propre (**évitez les « plats de spaghetti » !!**) et plus robuste que sur breadboard. Du matériel de soudure est mis votre disposition au laboratoire.
e) forcer les sorties que vous voulez visualiser : les états d'une machine sont souvent mis entre parenthèses sur le schéma d'attribution des broches dans le report file, ce qui ne garantit pas la possibilité de visualiser la valeur binaire. Considérez l'assignation du compilateur (voir report file) et définissez des sorties (un vecteur

de bits généralement) et un process qui attribue à ces nouvelles sorties les valeurs de l'état courant (ceci ne consomme nullement des macrocellules supplémentaires, cette méthode assure que le tri-state buffer de sortie est passant).

Les sorties de la GAL restent aux environs de 2.5V ? il s'agit bien d'une instabilité du circuit qui peut être résolue en touchant à tout et n'importe quoi (changer de composants, changer de fils, tester les IO,…).

Allez trouver l'assistant avant de remplacer du matériel. Il peut vous éviter de casser votre tirelire…