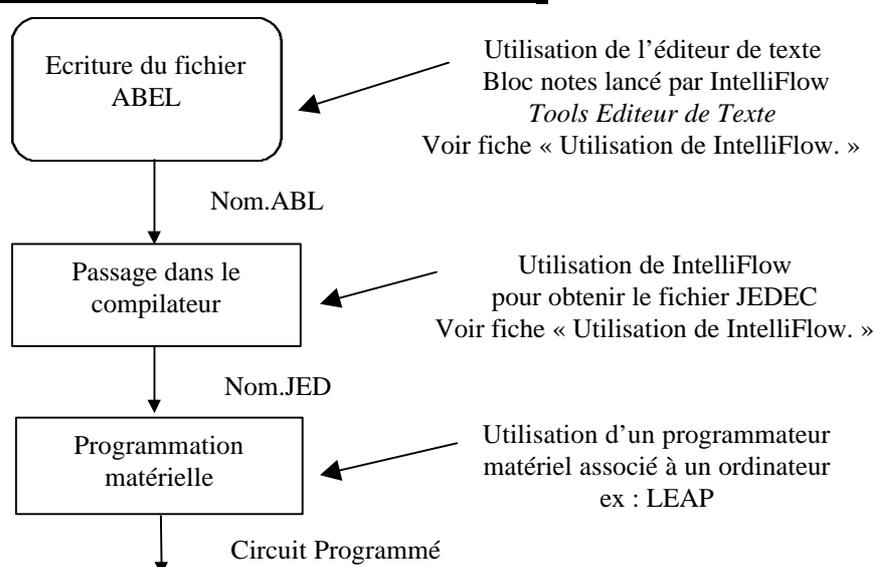


Programmation des PLD avec ABEL (W95)

Cette fiche est consacrée au langage de programmation ABEL (Advanced Boolean Equation Logical) dans le cas de fonctionnements combinatoires et séquentiels.

La programmation d'un circuit logique programmable de type PAL ou GAL nécessite un fichier standardisé au format JEDEC (Nom.JED) qui correspond à la table des fusibles (masque) du composant à programmer. L'obtention de ce fichier JEDEC est assurée par un compilateur (un grand mot pour dire programme !) dont le point d'entrée est un fichier texte écrit dans le langage ABEL présenté dans cette fiche.

1 - Principe de programmation d'un PLD



2 - Syntaxe de base du langage ABEL

2-1 Opérateurs Logiques

Le tableau ci-dessous indique la syntaxe ABEL permettant de décrire des équations logiques utilisant les opérateurs logiques usuels.

Opérateurs	Noms	Syntaxe	Fonction obtenue
!	Non	$C = !A$	\overline{A}
&	Et	$C = A \& B$	$A.B$
#	Ou	$C = A \# B$	$A+B$
\$	Ou Exclusif	$C = A \$ B$	$A \oplus B$
!\$	Non Ou Exclusif	$C = A !\$ B$	$\overline{A \oplus B}$

2-2 Ecrire une équation combinatoire ou séquentielle

=	<p align="center"><i>Affectation combinatoire</i></p> <p align="center">ex : F = A & B</p> <p>La sortie combinatoire F est égale à A et B.</p>
:=	<p align="center"><i>Affectation de registres</i></p> <p align="center">ex : Q1 := A</p> <p>Après action de l'horloge la sortie Q1 recopie l'état logique de la broche d'entrée A.</p>

Dans les parties 3 et 4 les mots et les caractères indiqués ***en gras et en italique*** sont des mots clés syntaxiques permettant au compilateur de reconnaître un fichier ABEL.

3 - Exemple de fichier ABEL combinatoire

4 - Exemple de fichier ABEL séquentiel

```

MODULE   PROG1
TITLE   ' Exemple Combinatoire '

DECLARATIONS
    A, B PIN 2, 4 ;
    F1, F2 PIN 15, 16 ISTYPE 'COM' ;

EQUATIONS
    F1      =      A & B ;
    F2      =      A # B ;

END PROG1
    
```

```

MODULE   PROG2
TITLE   ' Exemple Séquentiel '

DECLARATIONS
    RAZ,H PIN 2,1;
    q3,q2,q1,q0 PIN 14, 15, 16, 17 ISTYPE 'REG';

EQUATIONS
    [q3..q0].CLK=H;
    [q3..q0].ACLR=H;

    q0:=!q0;
    q1:=(q1 & !q0) # (!q1 & q0) ;
    q2:=(q2 & !q0) # (q2 & !q1) # (!q2 & q0 & q1);
    q3:=(q3 & !q0) # (q3 & !q1) # (q3 & !q2) # (!q3 & q0 & q1 & q2);

END PROG2
    
```

Remarques sur les attributs (Dot Extensions) en séquentiel:

- .CLK** ⇒ Entrée d'horloge sur front.
- .AR** ⇒ Reset asynchrone.
- .ASET** ⇒ Set Asynchrone.
- .OE** ⇒ Output Enable.