EASY ABEL DE DATA I/O

ABEL, de la société américaine DATA I/O, est un standard dans le domaine du développement pour composants programmables. DATA I/O, représentée en France par MB Electronique, est en premier lieu un fabricant de programmateurs, tous composants programmables confondus. L'expansion des composants programmables – et leurs performances accrues – conduit inévitablement à une offre de plus en plus pointue concernant les logiciels de développement. EZ-ABEL constitue un point de passage obligatoire.

Un peu d'histoire

Les réseaux logiques programmables (PLD pour Programmable Logical Devices) ont été introduits il y a une quinzaine d'années, permettant à l'époque le remplacement de quelques circuits TTL traditionnels 74XX. Il s'agissait de composants à boîtiers 20 ou 24 Broches.
Depuis, des composants à plus forte intégration sont apparus comme les CPLDs (pour Complex PLD), à plus grand nombre de broches (40 à 84) intégrant eux-mêmes plusieurs PLDs.
Enfin dans le haut de gamme nous retrouvons des composants à très forte intégration et à grand nombre de broches appelés FPGA (comme Field Programmable Gate Array), rivalisant avec les ASICs bas de gamme en terme de caractéristiques, mais réduisant les coûts et la durée de développement.
Tous ces composants apportent à l'utilisateur une grande souplesse d'exploitation, car la fonction interne est définie par le concepteur, un gain de place sur la carte de circuit imprimé, un abaissement de la consommation, etc.
Les prix de ces composants sont tels qu'ils remplacent avantageusement les circuits TTL, en contrepartie l'utilisateur doit disposer d'outils de développement et de programmation.
La programmation de ces composants est de plus simplifiée car le contenu du PLD peut être programmé et reprogrammé avec des signaux électriques de niveaux TTL, certains étant même programmables in-situ c'est-à-dire lorsqu'ils sont soudés sur la carte ; reste à l'utilisateur à disposer d'outils de saisie suffisamment performants et abordables pour bénéficier pleinement de tous les avantages.

Méthodologies de description

Nombre d'entre nous utilisent une description par schéma fonctionnel pour l'utilisation de composants discrets et en particulier analogiques. Mais des méthodes mieux adaptées aux PLD existent, par une description comportementale ; nous allons les passer en revue :

● Équations :

Utilisant les opérateurs traditionnels booléens ET, OU, NON, l'utilisateur peut exprimer aisément son application ; à l'aide d'opérateurs relationnels (égalité, inférieur à, supérieur à) un confort supplémentaire est offert. De plus les opérateurs arithmétiques + et - sont également supportés, autorisant la description d'incrément/décrément. Il faut aussi souligner que les signaux peuvent être groupés sous une appellation unique, pour une manipulation simple.

Ex : Sortie = Entrée & !Raz;
Où sortie sera vrai lorsque entrée sera vrai ET raz sera faux.

● Tables de vérité :

Très adaptée à la logique de décodage, c'est entre autres la méthode de description la plus utilisée dans les «data books» des fabricants de circuits.

```
[ !Entree, !Raz ] -> !Sortie
[   0    ,   0  ] -> 0 ;
[   0    ,   1  ] -> 1;
[   1    ,   0  ] -> 0 ;
[   1    ,   1  ] -> 0 ;
```

Bien qu'elle soit très employée, cette approche ne permet pas la description de machine d'états.

Machines d'états :
Permet une description algorithmique d'automates ou autres logiques séquentielles.
Le principe est basé sur l'utilisation de diagrammes de machines à états finis, pour lesquelles il existe plusieurs méthodes de représentation, les plus connues étant les machines de Mealy et de Moore. Le principe de base étant de définir des états de l'automate pendant lesquels certaines variables de sortie sont positionnées ; le passage d'un état à un autre s'effectue sous condition plus ou moins complexe de combinaisons d'entrées, cette transition sera effectuée de manière synchrone à l'horloge.

Il existe maintenant des possibilités d'approche par saisie GRAPHIQUE pour ces automates, voir l'exemple (3) décrit plus loin, méthode très proche de la manière dont se comporte l'application et rencontrant de plus en plus de succès.
DATA I/O, leader de solutions pour les composants programmables, à été le premier (1983) à introduire un outil universel indépendant des fabricants de composants, et supportant ces différentes approches.

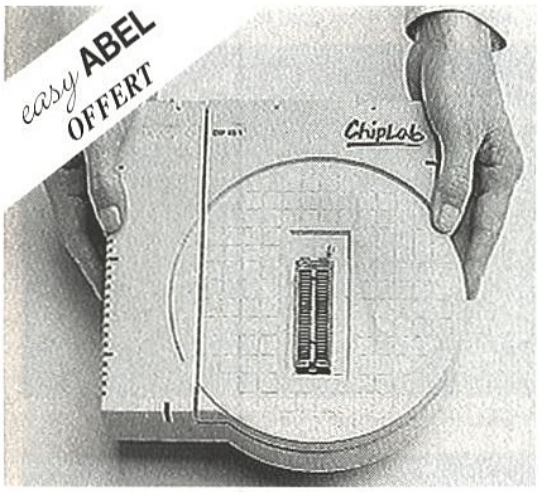
Remarque importante :
Les opérateurs booléens (notamment la fonction OU et la fonction ET) sont différents des opérateurs utilisés traditionnellement, car : le '+' pour la fonction OU (# en Abel-HDL) et le'*' pour la fonction ET (& en Abel-HDL) sont utilisés dans le langage pour leur fonctionnalité arithmétique (l'addition et la multiplication !), d'autres langages évolués proposent les mêmes caractères pour ces opérateurs.
Afin de mieux comprendre la puissance de description de ce langage, nous allons visualiser trois exemples.

● Trois exemples :

1) Un module de décodage d'adresses pour microprocesseur :
Considérons trois champs d'adresses à décoder sur un bus de microprocesseur (RAM, ROM et I/O).
L'espace RAM sera compris entre l'adresse 0 et l'adresse 40000h (Hexadécimal).
L'espace ROM sera compris entre l'adresse 80000h et l'adresse A0000h, valeurs en Hexadécimal.
Les entrées-sorties seront comprises entre F00000h et F00008h (Hexadécimal).
Les signaux de sortie seront actifs à l'état bas, pour être compatibles avec les signaux CE des boîtiers à sélectionner.
Le source équivalent pour cet exemple ressemblera à ceci :

```
module decodeur
title 'Exemple de module de decodage pour microprocesseur'

    A0..A23                     pin ; « Bus d adresse
    RAM,ROM,IO                      pin ; « Sorties

    H,L,X  = 1,0,.X.;                   « Affectations d equivalences
    BUS_addresses = [A0..A23];

equations

    !RAM   = (BUS_addresses <  ^h40000);

    !ROM   = (BUS_addresses >= ^h80000) & (BUS_addresses <  ^hA0000);

    !IO    = (BUS_Addresses >= ^hF00000)& (BUS_addresses <= ^hF00008);


test_vectors
        (BUS_addresses -> [ ROM,RAM, IO])
            ^h0000  -> [  H,  L,  H ];
            ^h4000  -> [  H,  L,  H ];
            ^h8000  -> [  H,  L,  H ];
            ^h40000 -> [  H,  H,  H ];
            ^h70000 -> [  H,  H,  H ];
            ^h80000 -> [  L,  H,  H ];
            ^hC0000 -> [  L,  H,  H ];
            ^hF0000 -> [  H,  H,  L ];
            ^hF0008 -> [  H,  H,  L ];

end
```


OFFREZ-VOUS LA TECHNOLOGIE DATA I/O POUR LE PRIX D'UNE CARTE PC



CHIPLAB™ est votre programmateur personnel pour vos utilisations d'études et de maintenance. CHIPLAB™ programme tous les composants nécessaires à l'aboutissement de vos projets.

- Il permet de programmer le plus grand nombre de composants (PLD, PROM, EPROM et Microcontrôleur, jusqu'à 48 broches) pour un coût très bas.
- Rapidité et simplicité d'utilisation grâce à son environnement Window.
- Sécurité, qualité et fiabilité DATA I/O.
- Mise à jour par modem.
- **Existe en deux versions :**
  * **Chiplab Memory Micro : 7995,00 F**
  * **Chiplab Logic Plus : 12850,00 F**

Pour toute commande ou pour plus d'informations, n'hésitez pas à contacter nos distributeurs agréés :

FARNELL E.C.	16 74 65 94 66
FRANÇAISE D'INSTRUMENTATION	16 25 78 15 55
JOD INSTRUMENTATION	39 56 00 95
LG ELECTRONIC	48 61 95 28
OMNITECH-SERTRONIQUE	16 43 86 74 74
RADIOSPARES COMPOSANTS	16 44 84 72 72

ChipLab

MB ELECTRONIQUE

Easy ABEL : logiciel de conception de PLD
Chiplab™ et Data I/O sont des marques déposées de Data I/O Corporation

606, rue Fourny - ZI de Buc - BP 31 - 78533 BUC CEDEX
Tél. 39 56 81 31 - Fax : 39 56 53 44

On remarquera le choix de la polarité du signal par l'opérateur «!», l'utilisation d'opérateurs relationnels «<, > <=,>= » et la déclaration des valeurs directement en hexadécimal (^h), rendant ce fichier source extrêmement simple à décrire et à comprendre.

Les vecteurs de test, non obligatoires (quoique fortement recommandés !) permettent de s'assurer, pour l'utilisateur, de la bonne compréhension de la syntaxe, voire de la sémantique d'Abel-HDL.

2) Un compteur par 2691

```
module  compteur

title'Un exemple de compteur simple';
     Horloge,Raz pin;
     q0..q3              pin;    « Ou l'on déclare de q0 à q3
     compte = [q3..q0];« Ou l'on regroupe plusieurs signaux ensemble
equations
     compte.clk = Horloge;
     compte := ( compte +1) & ( compte < 2691 ) & ! Raz

test_vectors
          ([Horloge      ,Raz  ] -> compte)
           [  C      , 0   ] - > 0;
           [  C      , 1   ] - > 1;
           [  C      , 1   ] - > 2;
           [  C      , 1   ] - > 3;
           [  C      , 0   ] - > 0;
           [  C      , 1   ] - > 1;
           [  P      , 1   ] - > 2689;
           [  C      , 1   ] - > 2690;
           [  C      , 1   ] - > 2691;
           [  C      , 1   ] - > 0;
end
```

Ici, on remarquera la facilité de description de fonction de comptage/décomptage.

Remarquez pour les vecteurs de test l'utilisation pour l'horloge de mnémoniques particuliers :

C (clock) correspondant à une transition 0,1,0 et P (preload) correspondant à un préchargement d'une valeur.

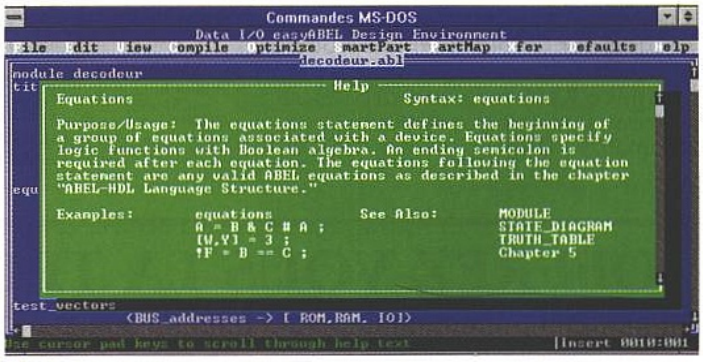
La simplicité de description offerte par le langage permet la lecture directe sans autre explication !

3) Un automate de gestion automatique d'enregistrement pour magnétophone

Ce fichier source a été obtenu automatiquement à partir de la description graphique en tête de chapitre, grâce à l'outil State-CAD de DATA I/O. Cette méthode particulièrement rapide permet une relecture immédiate par n'importe quelle personne, sans commentaires de la part du concepteur. En sortie, l'outil STATE-CAD est à même de générer directement un source compilable en langage Abel-HDL

Lorque l'application est saisie, le logiciel ABEL prendra en compte automatiquement les différentes phases de vérification et d'optimisation nécessaires pour une description efficace du projet.

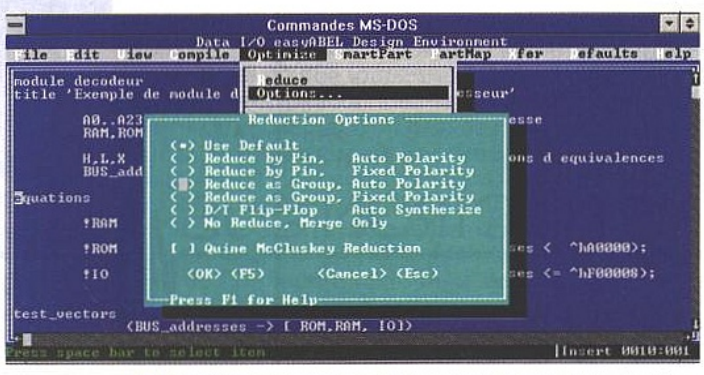
L'environnement de développement d'easy Abel avec son éditeur intégré.



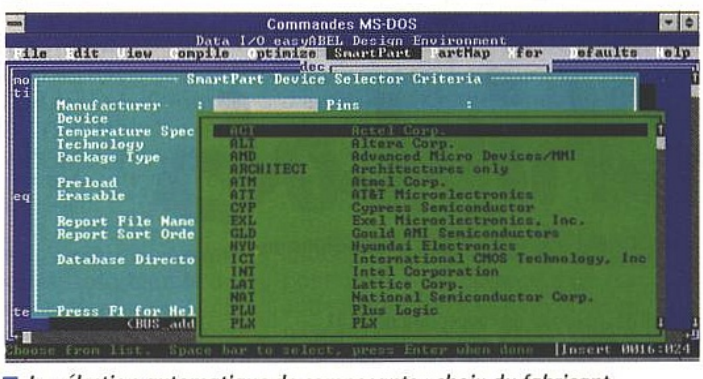
Un des nombreux menus d'aide : ici aide contextuelle relative au mot clé «équations» pointé par le curseur dans la fenêtre d'édition.

Un des nombreux menus d'aide : les opérateurs.

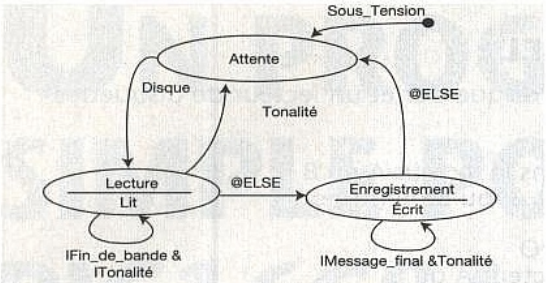
Sélection du composant à utiliser parmi la liste des candidats.



Synthèse logique : choix de l'algorithme d'optimisation.



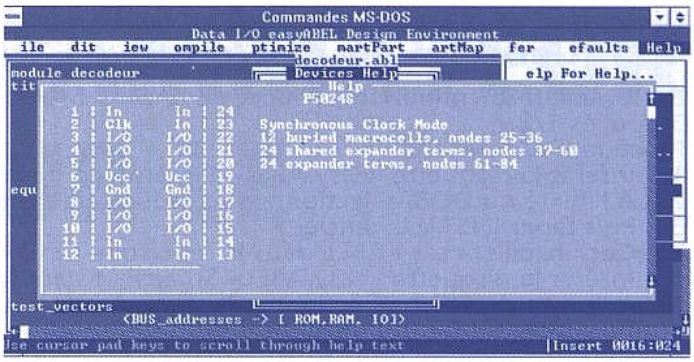
La sélection automatique de composants : choix du fabricant.



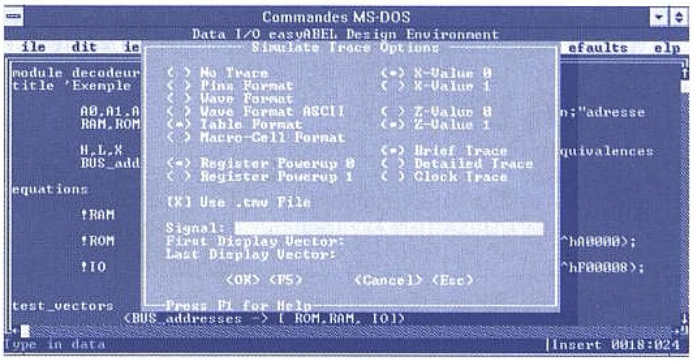
Description graphique.

Automate de gestion automatique d'enregistrement.

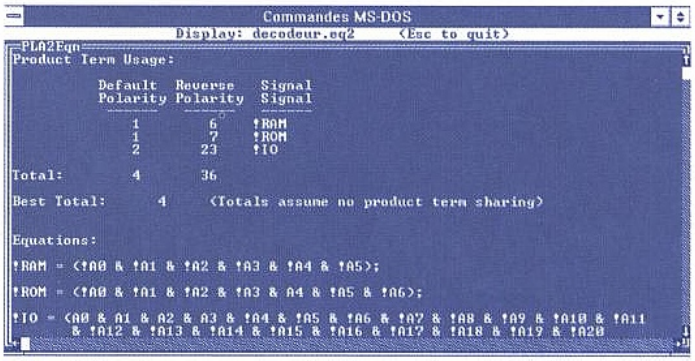
```
MODULE STATE
DECLARATIONS
«Horloge
    CLK                 PIN;
« Variables d' entrées
    Tonalite            PIN;
    Fin_de_bande                PIN;
    Message_final               PIN;
    Disque                      PIN;
    Sous_Tension                PIN;
« Variables de sorties
    Lit             PIN ISTYPE 'reg';
    Ecrit           PIN ISTYPE 'reg';
« Variables d' états
    SV1             PIN ISTYPE 'reg';
    SV0             PIN ISTYPE 'reg';
«Assignation des Horloges
EQUATIONS
    Lit.clk=CLK;
    Ecrit.clk=CLK;
«Assignation du registre d'états
DECLARATIONS
    sreg=[ SV1,SV0];
EQUATIONS
    sreg.clk=CLK;
DECLARATIONS
    Attente             =       [0, 0];
    Enregistrement      =       [0, 1];
    Lecture             =       [1, 0];
EQUATIONS
    WHEN (!(
            (sreg.FB==Attente)
          # (sreg.FB==Enregistrement)
          # (sreg.FB==Lecture)
    ) &
          ( Sous_Tension )) THEN
          [sreg] := [Attente];
state_diagram sreg;
state Attente:
    IF ( !Disque  & !Sous_Tension ) THEN Attente
    IF ( !Sous_Tension  & Disque ) THEN Lecture WITH
          Lit :=1;
    ENDWITH;
    IF ( Sous_Tension ) THEN Attente;
state Enregistrement:
    IF ( !Sous_Tension  & !Message_final  & Tonalite ) THEN Enregistrement WITH
          Ecrit :=1;
    ENDWITH;
    IF ( !Tonalite  & !Sous_Tension #  Message_final  & !Sous_Tension ) THEN
          Attente
    IF ( Sous_Tension ) THEN Attente;
state Lecture:
    IF ( !Sous_Tension  & Tonalite ) THEN Attente
    IF ( !Sous_Tension  & !Fin_de_bande  & !Tonalite ) THEN Lecture WITH
          Lit :=1;
    ENDWITH;
    IF ( !Tonalite  & Fin_de_bande  & !Sous_Tension ) THEN Enregistrement WITH
          Ecrit :=1;
    ENDWITH;
    IF ( Sous_Tension ) THEN Attente;
END
```



Assistance en ligne permanente pour le concepteur : brochage d'un PLD.



La simulation permet de mettre au point rapidement un projet : ici les arguments et autres paramètres.



Des fichiers documents sont générés pour chacune des étapes de la conception : ici les équations bas niveaux.

Le concept d'EZ-ABEL (easyABEL)

Avec easyABEL vous aurez à votre disposition un compilateur Abel-HDL assurant l'optimisation de votre description par synthèse logique, permettant la génération d'un fichier de type JEDEC pour la programmation des PLDs ; le support des architectures est très vaste (+ de 300) couvrant les composants les plus répandus, comme les plus performants. L'outil comprend une base de données avec les références commerciales des fabricants, avec leurs caractéristiques (vitesse, consommation, nombre de broches etc). L'utilisateur peut effectuer une recherche automatique de solution, en la contraignant avec les caractéristiques souhaitées ; l'étape de mise en conformité (Fitter) sera assurée en affectant le brochage le mieux adapté et configurant les macros-cellules internes du PLD.
La technologie «Fitter» développée par DATA I/O, permet à l'utilisateur de s'affranchir au maximum des contraintes technologiques induites par les com-

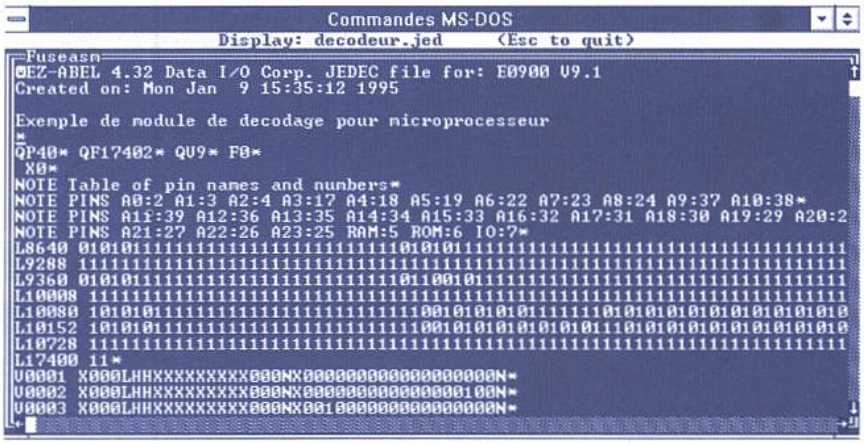
Configuration nécessaire pour EZ-ABEL :

DOS version 3.1 minimum, 640 ko de RAM, 5 Mo sur le disque dur et un lecteur de disquettes 3 pouces 1/2.
Procédure d'installation :

1. Insérer la disquette «easy-ABEL» Système de base dans le lecteur A ou B
2. A l'invite du DOS, sélectionner la disquette en tapant A: ou B: <entrée>
3. Puis taper INSTALL <entrée>
4. Ceci installera easy-ABEL dans le répertoire C:\DATAIO
5. Insérer la disquette «easy-ABEL» SmartPart dans le lecteur A ou B
6. A l'invite du DOS, sélectionner la disquette A en tapant A: ou B: <entrée>
7. Puis taper INSTALL 2 <entrée>
8. Ceci installera le complément SmartPart easy-ABEL dans le répertoire C:\DATAIO
9. Archiver vos disquettes
10. A l'invite du DOS, taper CD C:\DATAIO <entrée>
11. Taper la commande suivante : DB_BUILD <entrée>
12. Editer le fichier C:\AUTOEXEC.BAT y ajouter ou modifier les lignes suivantes :
    a) à la fin de la ligne PATH ajouter : C:\DATAIO
    b) ajouter la ligne SET ABEL4DEV=C:\DATAIO\LIB4
    c) ajouter la ligne SET ABEL4DB=C:\DATAIO\LIB4\DEVICES
    d) ajouter la ligne SET DB_DICT=C:\DATAIO\LIB4\DBASE
13. Editer le fichier C:\CONFIG.SYS y ajouter ou modifier les lignes suivantes :
    a) ajouter ou modifier la ligne BUFFERS = 20 au minimum
    b) ajouter ou modifier la ligne FILES = 15 au minimum
    c) avant l'utilisation du logiciel réinitialisez le PC.

posants, en évitant la tâche fastidieuse d'adaptation du logiciel au circuit ciblé.
Cette technologie est aussi bien utilisée pour des composants à faible complexité comme les PLDs que pour les composants haut de gamme que sont les CPLDs ou les FPGAs, mais là les étapes de synthèse logique et de placement-routage mettent en jeu des algorithmes plus sophistiqués, utilisés dans d'autres versions du produit, telles qu'ABEL6 ou SYNARIO, capables de gérer ces familles, mais c'est une autre histoire ...

Olivier CASSE
MB Électronique



L'ultime étape : le fichier JEDEC qui servira à programmer le PLD.

NDLR : ABEL est le logiciel indépendant de développement pour PLD le plus utilisé à ce jour.
En partenariat avec MB Électronique, représentant français de DATA I/O, ERP a décidé de vous faire bénéficier d'une offre particulièrement attractive pour acquérir ce logiciel compatible avec la grande majorité des composants programmables provenant de la plupart des fabricants de ce domaine. Pour cela nous vous proposons deux options d'acquisition.

Les modalités d'acquisition d'EZ-ABEL

Vous pouvez acquérir le logiciel EZ-ABEL selon deux options en retournant le coupon ci-dessous à Electronique Radio-Plans.
Option 1 : le logiciel seul sur disquettes en joignant 50 F à l'ordre d'Electronique Radio-Plans.
Option 2 : le logiciel plus le manuel plus l'enregistrement auprès de MB Electronique pour bénéficier du support technique MB Electronique. Dans ce cas, libellez votre chèque à l'ordre de MB Electronique pour un montant de 1180 F (995 F HT).

OFFRE SPÉCIALE EZ-ABEL MB Électronique/ERP

Coupon à retourner avec votre règlement à **Électronique Radio-Plans,**
2 à 12, rue de Bellevue - 75940 Paris Cedex 19
option 1 : logiciel seul, joindre 50 F à l'ordre d'Électronique Radio-Plans par chèque
option 2 : logiciel + manuel + support technique, joindre 1180 F à l'ordre de MB Electronique par chèque

Votre choix : ❑ OPTION 1 ❑ OPTION 2

Nom, prénom :.....	Société :.....
Profession :.....	Domaine d'activités :.....
Adresse :.....	Adresse :.....
Code postal :..... Tél. :.....	Code postal :..... Tél. :.....