

A• OBJECTIFS

- Réaliser les différentes étapes de la programmation d'un PLD.
- Vérifier expérimentalement la validité des programmations.

B• DOCUMENTS NÉCESSAIRES

- Manuel d'utilisation de ViewLogic (Programmation d'un circuit logique prog.).
- Cours 'Circuits logiques programmables'.

C• MATÉRIEL UTILISÉ

- Ordinateur équipé de ViewLogic.
- Maquette pédagogique 22V10 et programmeur.

D• PRÉSENTATION**I• RAPPELS SUR LES CIRCUITS LOGIQUES PROGRAMMABLES**

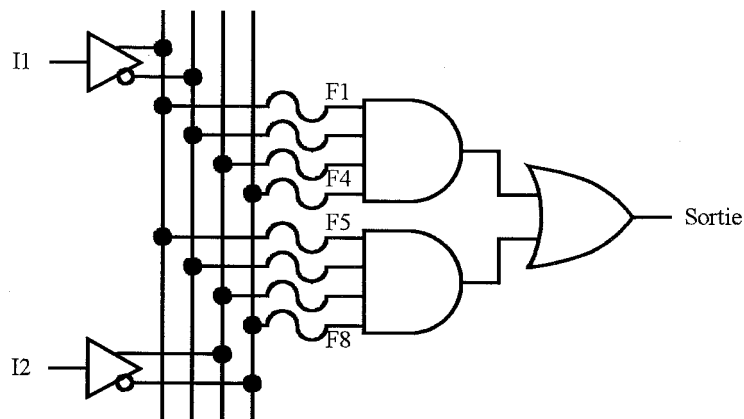
Les circuits logiques programmables disposent au moins de deux ensembles fonctionnels :

- Un ensemble d'opérateurs ET (matrice ET) recevant les entrées et leurs compléments.
- Un ensemble d'opérateurs OU (matrice OU) qui réalise une somme de produits et fournit les sorties.

Selon la complexité des circuits programmables utilisés une ou l'autre ou chacune des matrices pourra être programmée.

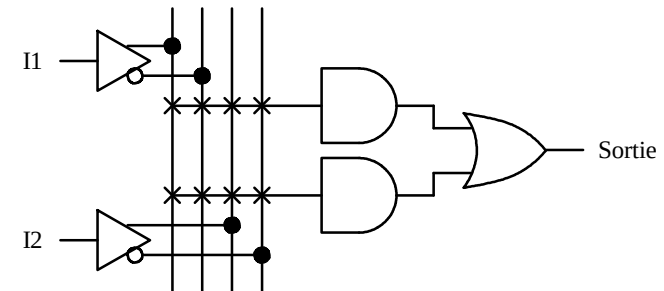
II• STRUCTURE DE BASE

La figure suivante présente un exemple simplifié de la structure interne de base d'un circuit PAL (matrice ET programmable) :

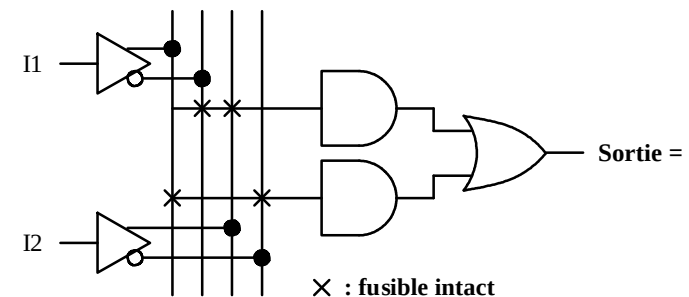


Les matrices programmables utilisent des "fusibles" placés en série entre les entrées et les opérateurs logiques. Dans les structures reprogrammables ces "fusibles" sont remplacés par des transistors MOS.

La représentation donnée dans l'exemple n'est pas utilisée en pratique, car elle occupe trop de place et cela conduirait à des schémas illisibles. Une représentation simplifiée (mais équivalente) a été adoptée pour représenter la structure interne des circuits logiques programmables :



A partir de la représentation simplifiée, il est possible d'établir l'équation de la sortie de la structure programmée :

**III• PRÉSENTATION DU GAL 22 V 10 (UTILISÉ EN TP)****1• PRÉSENTATION**

Le GAL 22V10 est implanté dans un boîtier DIL 24 broches ou PLCC 28 broches. Sa structure interne est donnée à la page suivante.

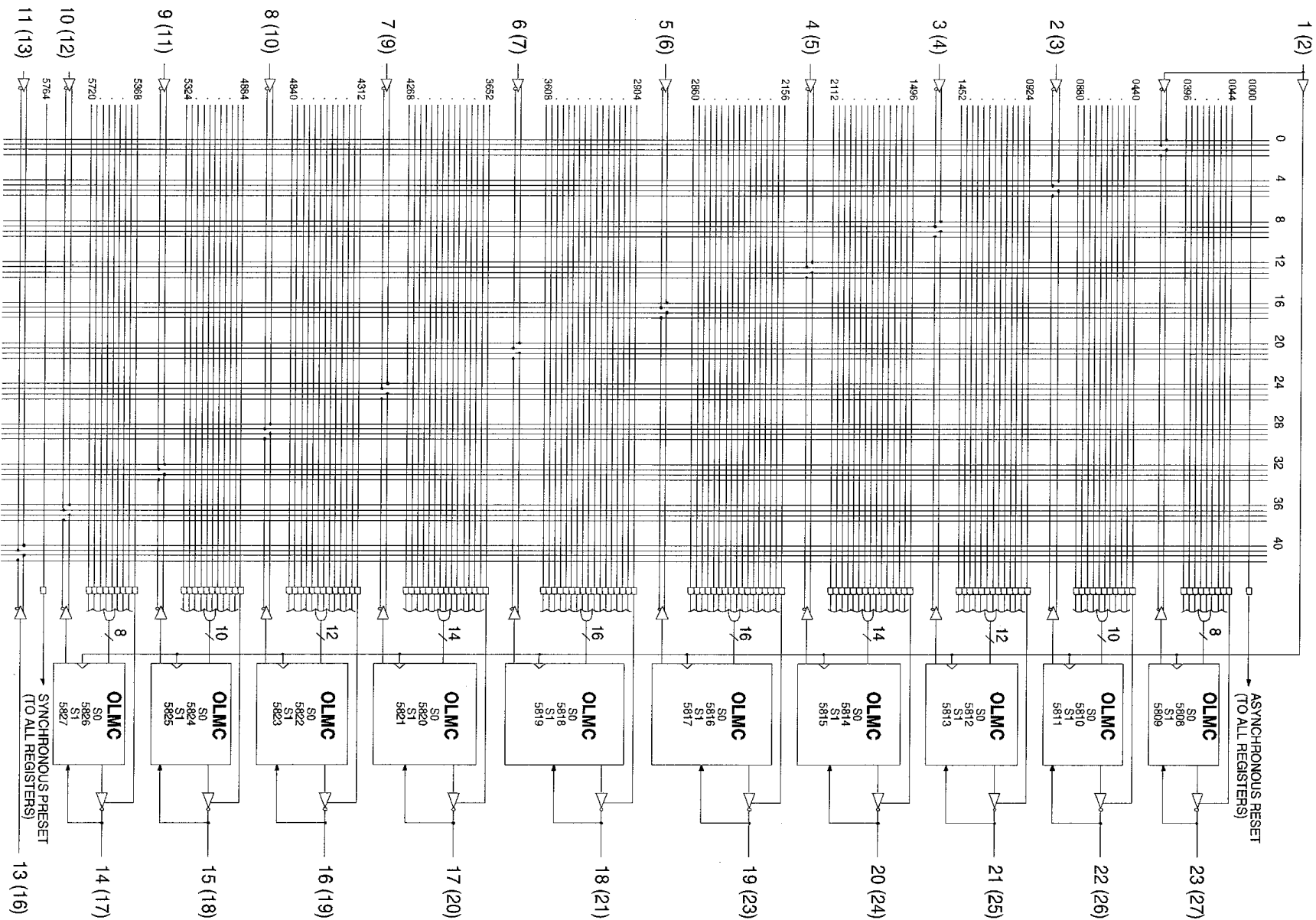
Ce circuit est doté de 12 entrées dont une à double fonction et de 10 sorties. On remarque que ces 10 sorties sont issues de cellules nommées OLMC (Output Logic Macro Cell). L'analyse de la structure interne (paragraphe suivant) montre que ces sorties sont réinjectables comme entrées.

22 : nombre d'entrées.

V : structure de sortie (bascule D).

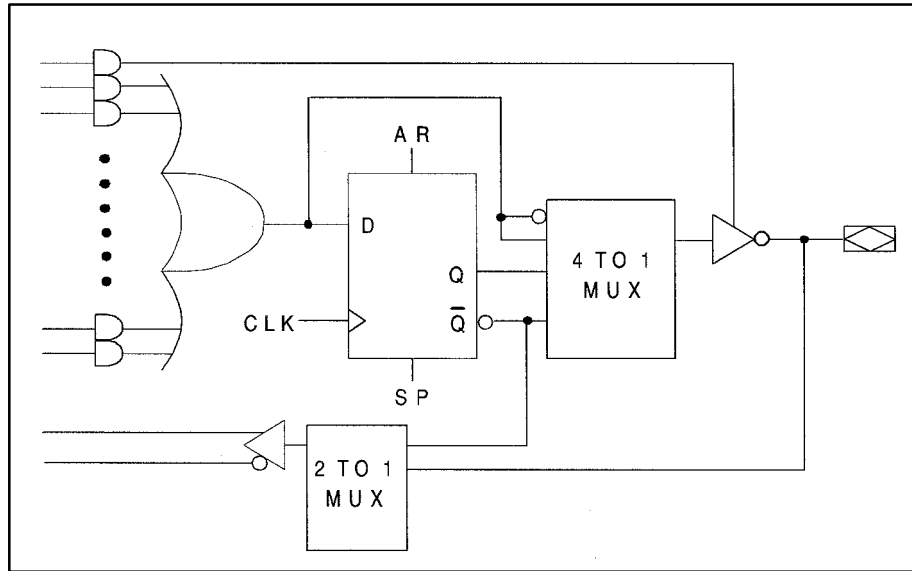
10 : nombre de sorties.

DIP (PLCC) Package Pinouts

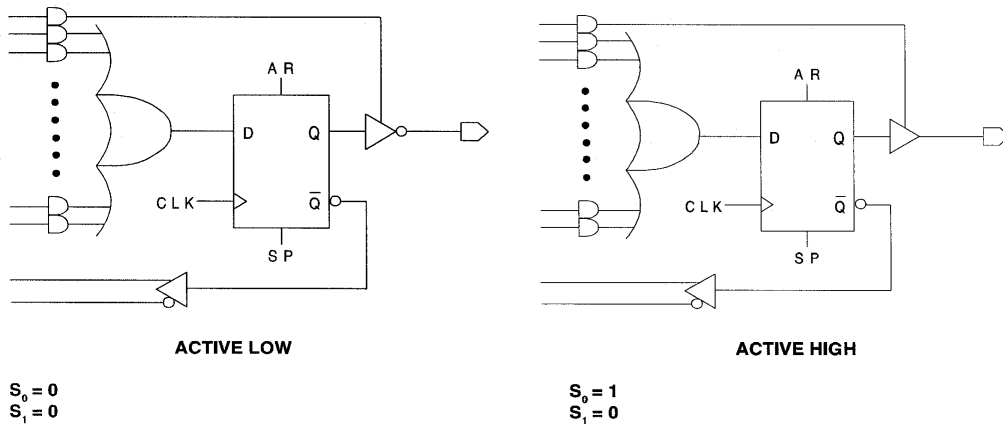
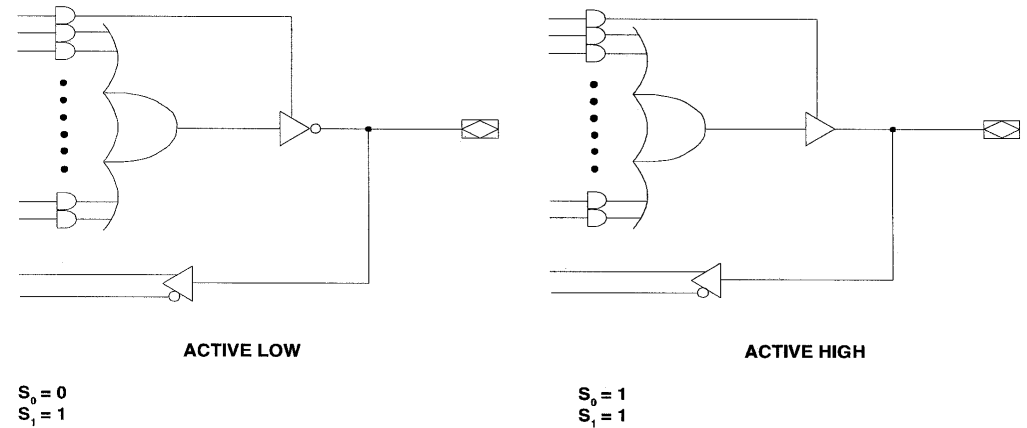


2• STRUCTURE INTERNE D'UNE OLMC

L'OLMC comporte principalement une bascule D et deux multiplexeurs. C'est une structure programmable.



Les figures données ci-après présentent les principales configurations possibles de l'OLMC.

3• REGISTERED MODE (BASCULE D UTILISÉE)4• COMBINATORIAL MODE (MODE COMBINATOIRE, BASCULE INUTILISÉE)

Dans ce mode, les sorties peuvent être réinjectées dans la matrice ET. Ces broches peuvent aussi être utilisées directement comme des entrées.

IV• PROGRAMMATION DU GAL 22V10

Pratiquement on ne programme jamais directement ni les OLMC ni les matrices. On utilise pour cela un logiciel (IntelliFlow de ViewLogic) qui connaît la structure du composant et qui y réalise l'implantation (fitting) du problème qu'on lui soumet. Néanmoins la connaissance du composant permet d'évaluer avec efficacité la faisabilité d'un projet.

L'analyse de la structure interne du GAL 22V10 met en évidence certaines contraintes d'utilisation et donc de programmation de certaines broches (en mode registered) :

- Les bascules ont une horloge commune (synchrone), broche 1.
- Les bascules sont dotées d'un RESET (mise à 0 des sorties Q) asynchrone et d'un PRESET (mise à 1) synchrone. **La programmation du RESET est obligatoire.**

La programmation d'un circuit logique programmable peut s'effectuer soit :

- A partir du schéma à implanter.
- A partir des équations ou de la table de vérité dans un fichier de description utilisant le langage ABEL.

1• ÉTAPES DE LA PROGRAMMATION

La présentation des différentes opérations ainsi que l'utilisation du logiciel de programmation des circuits logiques est donnée dans le fascicule View Logic (page 40 et suivantes). Elle comporte les étapes suivantes (par ordre chronologique) :

- La configuration du projet (**librairies indispensables : DIO, DSTD et BUILTIN**).
- L'écriture et la saisie du fichier ABEL (sous IntelliFlow).
- Le choix du circuit programmé (22V10).
- La création du fichier de programmation du circuit (JEDEC).
- La création du modèle de simulation.
- La validation du fichier par simulation logique (pour s'assurer que le fichier descriptif permettra une programmation correcte du circuit. Les circuits ne sont pas programmables plus d'une centaine de fois).
- La programmation du circuit (à l'aide d'un programmeur).
- Le test du composant programmé à l'aide des maquettes pédagogiques 22V10.

2• EXEMPLE DE FICHIER ABEL : DSPPROG.ABL

Module DSPPROG // _____

Title 'alarme de phares' // _____

Declarations // _____

//Entrées Commentaire (facultatif)

C,L,P pin 2, 3, 4; // _____

// _____

// _____

//Sorties Commentaire (facultatif)

S pin 23 istype 'com'; // _____

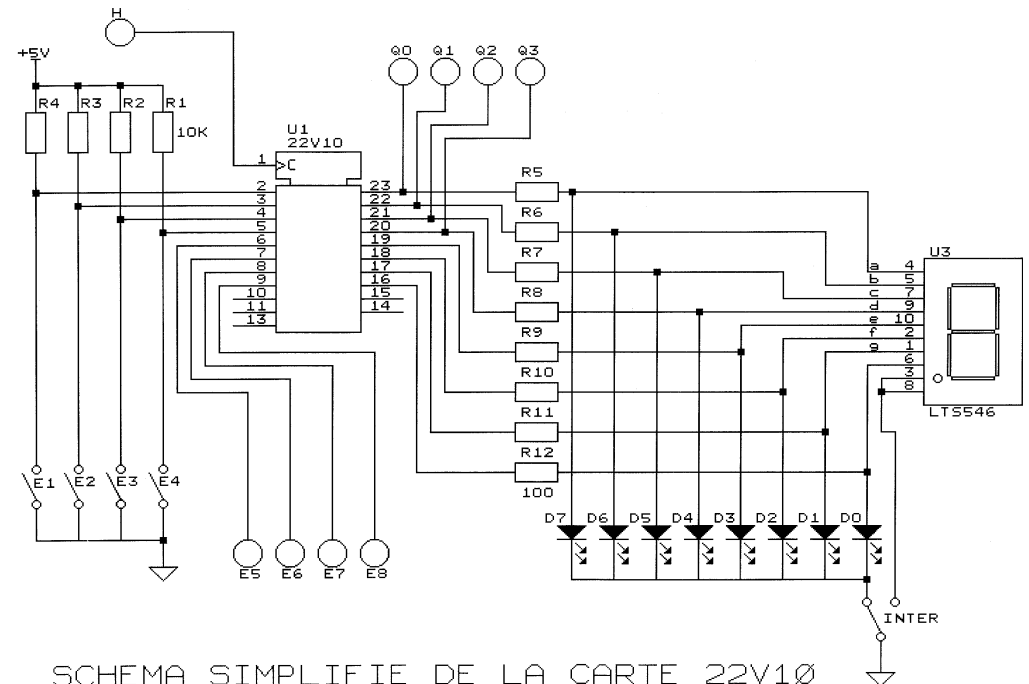
Equations // _____

S = !C & L & P; // _____

End DSPPROG _____

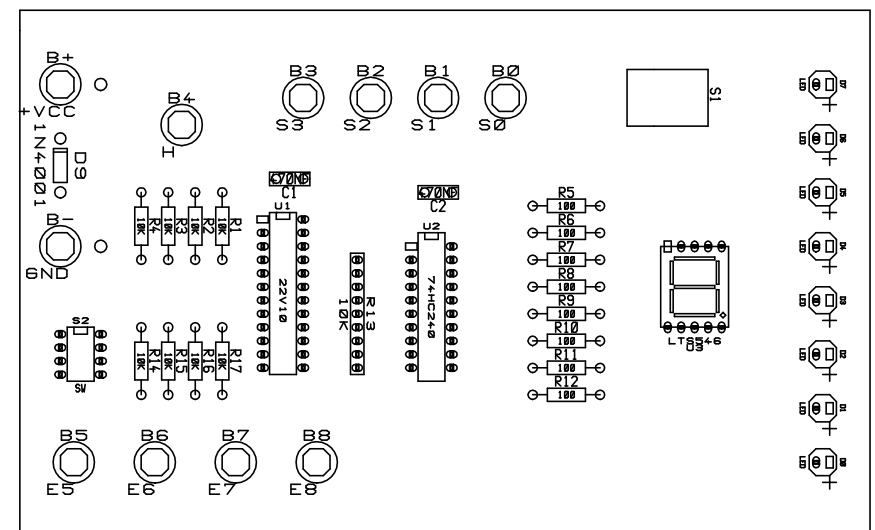
3• SCHÉMA STRUCTUREL DE LA MAQUETTE PÉDAGOGIQUE

On donne ci-contre le schéma structurel de la maquette utilisée pour les TP :



SCHEMA SIMPLIFIE DE LA CARTE 22V10

4• SCHÉMA D'IMPLANTATION



E• TRAVAIL DEMANDÉ

I• ÉTAPES DE LA PROGRAMMATION D'UN 22V10 (PAGE 35 ET SUITE)

Ce travail d'initiation à la programmation porte sur le dispositif de surveillance de phares. On souhaite implanter cette fonction dans un PALCE 22V10H-25PC/4 de AMD.

1• CONFIGURATION DU PROJET (PAGE 36)

Ouvrir **votre** projet de travail. Les librairies suivantes doivent impérativement être présentes ou ajoutées :

- F:\VENDOR\DIO\DIO (DIO)
- F:\LOGIQUE\BUILTIN (BUILTIN)
- F:\VENDOR\DIO\DSTD (DSTD)

2• ÉCRITURE DU FICHIER ABEL : INTELLIFLOW (PAGE 39)

Lancer IntelliFlow.

Il n'est pas possible de créer un nouveau fichier ABEL, on peut seulement ouvrir un fichier déjà existant.

Ouvrir le fichier DSPPROG.ABL. Compléter le fichier selon l'exemple donné dans à la page 4. Sauvergarder votre travail, puis fermer le bloc-notes.

3• CHOIX DU CIRCUIT : INTELLIFLOW (PAGE 40)

- ☐ **Source + Properties**
- ☐ Onglet **Device**
- ☐ Dans la rubrique **Design Target**, sélectionner PLD

Cliquer en suite sur **Run Devsel ...**

Le choix du circuit se fait sur sélection dans une liste après la définition de certains critères :

- Manufacturer (fabricant) : **Advanced Micro Devices (AMD)**
- Device (type de circuit) : **ANY**
- Température (gamme de température) : **Commercial**
- Technology (technologie) : **CMOS**
- Package (type de boîtier) : **DIP**
- Preload (programmable sur site) : **ANY**
- Erasable (mode d'effacement) : **EE** (effaçable électriquement).

Les autres critères de sélection ne sont pas modifiés.

Cliquer ensuite sur OK et attendre. Dans l'onglet **PLD Device Datasheet**, où apparaissent tous les composants répondant aux critères définis, sélectionner le **PALCE 22V10H-25PC/4**.

4• CRÉATION DU FICHIER JEDEC : INTELLIFLOW (PAGE 41)

Pour créer le fichier JEDEC (DSPPROG.JED) qui servira à la programmation du composant :

- ☐ Double cliquer sur la ligne **Implement and create JEDEC FILE**.

Attendre, puis vérifier que la case est cochée par une marque verte.

Pour créer le modèle de simulation qui servira à la simulation logique du composant :

- ☐ Double cliquer sur la ligne **Implement and Perform Timing Simulation**.

Attendre, puis vérifier que la case est cochée par une marque verte.

5• SIMULATION LOGIQUE DU COMPOSANT CRÉÉ

On effectue cette simulation pour s'assurer que le fichier de programmation créé est correct avant d'effectuer la programmation réelle du 22V10.

Lancer Viewdraw et ouvrir un nouveau fichier de dessin : PHAREGAL.

Insérer le composant créé (DSPPROG.1) dans le schéma.

Pour pouvoir simuler ce composant il faut impérativement effectuer la modification suivante :

- Pointer le composant et cliquer avec le bouton droit. Sélectionner **Symbol**. Le composant apparaît dans une autre fenêtre.
- Cliquer avec le bouton droit à l'extérieur du composant (mais dans la fenêtre). Sélectionner **Properties**.
- Dans le cadre **Symbol Type**, sélectionner **Module** et cliquer sur OK.
- Enregistrer la modification par : **File + Save**. Quitter : **File + Close**.

Compléter le schéma : placer les fils d'entrées et de sortie sans oublier les labels.

Effectuer la simulation en utilisant le fichier de commande déjà créée pour cette application lors d'un TP précédent. Vérifier le fonctionnement du montage.

Faire vérifier par le professeur.

6• VÉRIFICATION EXPÉRIMENTALE

Programmer un 22V10 avec le fichier JEDEC : DSPPROG.JED

Placer le composant programmé sur la carte pédagogique 22V10.

Attention ! le composant est extrêmement fragile. La tension d'entrée ne doit surtout pas dépasser 5V.

Tester les différentes combinaisons. **Faire vérifier par le professeur.**

II• STORE SOMFY : INTÉGRATION DE FP4 DANS UN 22V10

On souhaite intégrer la fonction combinatoire FP4 (étudiée lors du TD N°4) dans un 22V10. Il est donc indispensable de posséder les documents relatifs à cette étude.

1• ÉQUATIONS DE FONCTIONNEMENT

Rappeler les équations des sorties Monte et Descente en fonction des variables d'entrée S, V, M et D :

2• DESCRIPTION DU FICHIER ABEL

On utilisera comme pour les entrées S, V, M et D les broches 2, 3, 4 et 5.

les sorties Monte et Descente seront placées sur les broches 23 et 22.

En vous aidant de l'exemple précédent et du langage de programmation ABEL, compléter le fichier STORESOM.ABL permettant d'obtenir le fonctionnement souhaité.

MODULE _____

TITLE _____

DECLARATIONS

EQUATIONS

END _____

3• ÉCRITURE DU FICHIER ABEL

Lancer IntelliFlow. Ouvrir le fichier STORESOM.ABL. Compléter le fichier. Sauvegarder votre travail, puis fermer le bloc-notes.

4• CHOIX DU CIRCUIT : INTELLIFLOW (PAGE 40)

Reprendre les mêmes critères de sélection que ceux utilisés à l'exemple N°1 et sélectionner le **PALCE 22V10H -25PC/4** de AMD.

5• CRÉATION DU FICHIER JEDEC : INTELLIFLOW (PAGE 41)

Créer le fichier JEDEC (STORESOM.JED) ainsi que le modèle de simulation.

6• SIMULATION LOGIQUE DU COMPOSANT CRÉE

Lancer Viewdraw et ouvrir un nouveau fichier de dessin : STOREGAL

Insérer le composant créé (STORESOM.1) dans le schéma.

Penser à changer les propriétés du composant en module (à la place de composite).

Compléter le schéma : placer les fils d'entrées et de sortie sans oublier les labels.

Effectuer la simulation en utilisant le fichier de commande déjà créé pour cette application lors du TP N°4. Vérifier les différentes combinaisons.

7• VÉRIFICATION EXPÉRIMENTALE

Programmer un 22V10 avec le fichier JEDEC : STORESOM.JED

Placer le composant programmé sur la carte pédagogique 22V10.

Tester le fonctionnement du montage. **Faire vérifier par le professeur.**

III• DÉCODEUR BCD / 7 SEGMENTS

On souhaite intégrer la fonction Décodeur BCD/7 segments (TD N°3) dans un 22V10.

1• DESCRIPTION DU FICHIER ABEL

On va utiliser dans le cas présent la description par la **table de vérité**.

On utilisera les entrées E1, E2, E3 et E4 (broches 2, 3, 4 et 5).

les sorties a, b, c, d, e, f et g seront placées aux broches 23 à 17 (dans cet ordre).

Compléter le fichier DECODEUR.ABL permettant d'obtenir le fonctionnement souhaité.

MODULE _____

TITLE 'decodeur BCD 7 segments'

DECLARATIONS

//entrées

E1, E2, E3, E4 pin _____

//sorties

a, b, c, d, e, f, g pin _____

@dcset //pour ne pas tenir compte des combinaisons d'entrée inutilisées

TRUTH_TABLE ([E4, E3, E2, E1] -> [a, b, c, d, e, f, g])

[0, 0, 0, 0] -> [1, 1, 1, 1, 1, 0];

[0, 0, 0, 1] -> [0, 1, 1, 0, 0, 0];

[0, 0, 1, 0] -> _____

[0, 0, 1, 1] -> _____

[0, 1, 0, 0] -> _____

[0, 1, 0, 1] -> _____

[0, 1, 1, 0] -> _____

[0, 1, 1, 1] -> _____

[1, 0, 0, 0] -> _____

[1, 0, 0, 1] -> _____

END _____

2• ÉCRITURE DU FICHIER ABEL

Lancer IntelliFLoW. Ouvrir le fichier DECODEUR.ABL. Compléter le fichier.

Sauvegarder votre travail, puis fermer le bloc-notes.

3• CHOIX DU CIRCUIT ET CRÉATION DU FICHIER JEDEC

Reprendre les mêmes critères et sélectionner le **PALCE 22V10H-25PC/4** de AMD.

Créer le fichier JEDEC (DECODEUR.JED) ainsi que le modèle de simulation.

4• SIMULATION LOGIQUE DU COMPOSANT CRÉE

Lancer Viewdraw et ouvrir un nouveau fichier de dessin : DECODGAL

Insérer le composant créé (DECODEUR.1) dans le schéma.

Penser à changer les propriétés du composant en module (à la place de composite).

Effectuer la simulation du montage : écrire le fichier de commande qui permet d'appliquer les différentes combinaisons d'entrée en utilisant un stepsize de 1 ms.

Vérifier le fonctionnement .

5• VÉRIFICATION EXPÉRIMENTALE

Programmer un 22V10 avec le fichier JEDEC : DECODEUR.JED.

Placer le composant programmé sur la carte pédagogique 22V10.

Tester le fonctionnement du montage. **Faire vérifier par le professeur.**