

Simulation du fonctionnement interne d'une jonction P-N à l'aide d'un micro-ordinateur

Nous analysons une partie d'un programme qui permet de simuler sur l'écran d'un micro-ordinateur, le fonctionnement interne d'une jonction P-N. Nous montrons, à l'aide de cet exemple, que :

- certains micro-ordinateurs actuels permettent des études jusqu'alors impossibles ou difficiles avec des élèves,
- de nouveaux outils pédagogiques sont à la disposition du professeur de physique.

Nous n'insistons pas ici sur les théories de la jonction P-N, de nombreux articles ayant été publiés sur ce sujet.

I. ANALYSE DU DEROULEMENT DE QUELQUES SEQUENCES DU PROGRAMME « JONCTION P-N ».

1. Séquence 1.

Présentation de la constitution interne d'une jonction P-N. Le programme met en place successivement dans la zone de simulation délimitée par un cadre dans la moitié supérieure de l'écran, les différentes particules chargées existant dans une jonction P-N (photographies nos 1 et 2).

On distingue ainsi :

en zone N : les particules fixes : ions donneurs positifs,

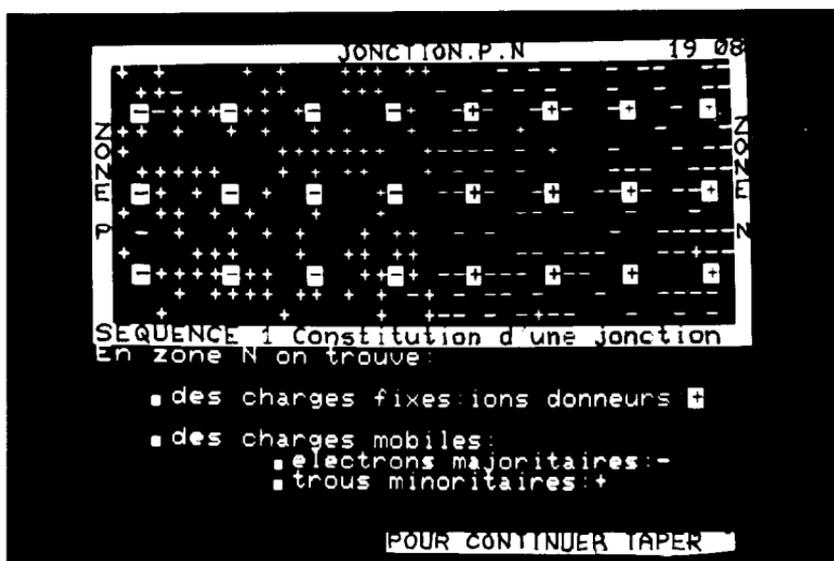
les particules mobiles { électrons majoritaires,
trous minoritaires ;

en zone P : les particules fixes : ions accepteurs négatifs,

les particules mobiles { trous majoritaires,
électrons minoritaires.

Remarquons que :

- les positions des particules mobiles sont définies aléatoirement,
- il est possible de modifier les concentrations des charges,



Photographies nos 1 et 2 :

Séquence 1 : deux étapes de la présentation de la constitution interne d'une jonction P-N. Remarquez, en haut à droite, le temps écoulé depuis le lancement du programme.

— il est possible de réaliser une version du programme sans texte, la zone de simulation occupant tout l'écran, pour une utilisation par le professeur, dans un cours traditionnel. Les photographies présentées montrent une version dans laquelle la mise en place d'un type de particule est accompagnée du commentaire correspondant.

2. Séquence 2.

Création, diffusion, recombinaison.

a) Création d'une paire électron-trou : le programme met en place à deux positions voisines libres, définies aléatoirement, un électron et un trou, avec un effet de clignotement pour attirer l'attention. Le commentaire signale des causes énergétiques susceptibles de provoquer cette création.

b) Diffusion d'une particule mobile (photographie n° 3) : un trou ou un électron, choisi au hasard, puis distingué des autres par une visualisation différente, est déplacé aléatoirement dans la jonction.



Photographie n° 3 :

Diffusion d'une particule mobile ; la flèche précise la position de la particule en train de se déplacer (dans ce cas, il s'agit d'un trou distingué des autres par une visualisation particulière : ■).

c) Recombinaison : la particule qui diffuse, finit par rencontrer une charge mobile de signe contraire. L'attention de l'élève est alors attirée par un commentaire, accompagné du cli-gnotement des deux particules concernées qui sont finalement effacées de l'écran.

d) Remarques :

- cette séquence permet d'introduire, sans calcul ni explications compliquées, les notions de durée de vie, de longueur de diffusion et de courant de diffusion,
- on peut vérifier statistiquement que les recombinaisons se font fréquemment au voisinage de la jonction,
- l'effet dynamique obtenu dans cette séquence ne peut absolument pas être rendu par un livre ou une photographie ; il est comparable à celui d'un dessin animé, l'utilisation du hasard permettant en plus de ne jamais avoir le même déroulement. Il y a là une dimension pédagogique nouvelle.

3. Séquence 3.

Zone de transition.



Photographie n° 4 :

Visualisation de la zone de transition.

Les phénomènes de création, diffusion, recombinaison sont visualisés jusqu'à l'obtention d'un régime stationnaire. Les recombinaisons se produisant plus fréquemment au voisinage de la jonction, les concentrations en charges mobiles diminuent dans cette zone. Le champ électrique interne progressivement pris en compte, intervient dans ce processus. Au bout de quelques minutes (en langage BASIC) apparaît nettement la zone de transition (photographie n° 4).

Remarques.

a) Le point délicat pour le concepteur du programme est le calcul de l'effet du champ électrique interne. Plusieurs solutions sont envisageables :

- faire un bilan complet des charges interagissant avec la particule considérée et déterminer le champ agissant sur cette particule. Cette procédure, trop longue en langage évolué a été abandonnée pour l'instant, mais est envisageable en langage assembleur,
- évaluer ce champ (ou le déplacement qu'il provoque) à l'aide d'une fonction définie empiriquement dépendant par exemple de la distance à la jonction, du nombre des recombinaisons... Les fonctions les plus simples sont celles qui marchent le mieux, le hasard fait le reste.

b) La modification possible des concentrations initiales permet de visualiser éventuellement la dissymétrie de la zone de transition en cas de dopage inégal des zones N et P, à condition que la procédure de calcul du champ prenne en compte cette dissymétrie (application possible aux transitions à effet de champ, MOS...).

4. Autres séquences et prolongements.

Il est possible de visualiser et d'étudier le comportement dans la zone de transition des différents types de porteurs en polarisation inverse.

Le cas de la polarisation directe soulève des difficultés non encore résolues, liées au temps de calcul en langage évolué.

Nous envisageons de poursuivre notre étude afin de montrer l'effet Zener et l'effet d'avalanche qui ne posent pas, *a priori*, de problèmes techniques particuliers.

II. ETUDE DU SOUS-PROGRAMME DE DIFFUSION D'UNE PARTICULE.

1. Organisation de l'écran de visualisation : nous développons ici le cas de l'ordinateur CBM 3016 que nous utilisons. Cette étude est transposable très facilement à un certain nombre de machines

(exemple : TRS 80, APPLE, X1...). Sur d'autres machines, cela est plus difficile (parce que la documentation est très incomplète) ou impossible (à cause de la conception différente de ces machines).

L'écran comprend L lignes, numérotées de \emptyset à $L-1$ et C colonnes numérotées de \emptyset à $C-1$.

L'unité centrale (microprocesseur) considère chaque position d'impression comme une *case mémoire*, repérée par un numéro d'ordre appelé *adresse* de cette case mémoire.

Les adresses des cases mémoires de l'écran sont comprises entre AO et AF (fig. 1).

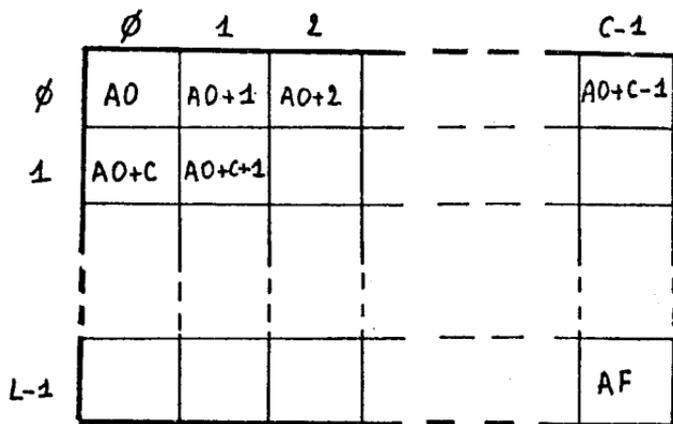


Fig. 1

$$\text{CBM 3016} : L = 25 \quad C = 40 \quad L \times C = 1000$$

$$AO = 32768 \quad AF = 33767$$

$$\text{X1} : L = 25 \quad C = 80 \quad L \times C = 2000$$

$$AO = 49152 \quad AF = 51151.$$

La case mémoire correspondant à la ligne y et à la colonne x a pour adresse :

$$AO + x + C \times y.$$

Le contenu de chaque case mémoire se présente sous les deux aspects :

- pour l'utilisateur : la position d'impression est acquise par une lettre, un chiffre, un signe, un caractère graphique...,
- pour l'unité centrale, le contenu de la case mémoire est représentable par un nombre généralement compris entre 0 et 255 (nombre de possibilités : $256 = 2^8$, codage possible en binaire à l'aide d'un nombre de 8 chiffres).

Nous appellerons *code-écran* la correspondance existant entre un caractère sur l'écran et le nombre qui le représente en mémoire. Ce code diffère selon les machines,

Exemples : sur un micro-ordinateur X1, il coïncide avec le

code international ASCII :

code de « A » : 65

code de « B » : 66

code de « a » : 97

sur CBM :

code de « A » : 1

code de « ▀ » : 105

code de « ■ » : 127.

2. Quelques opérations et fonctions du langage BASIC utilisées dans ce programme (pour plus de détails, on consultera les ouvrages cités) :

- * multiplication,
- + addition,
- soustraction,
- > plus grand que,
- < plus petit que,
- >< différent de,
- = affectation, exemple : $X = 5$, affecte la variable X de la valeur numérique 5 (notée \leftarrow dans l'analyse du programme),
- = égalité (opération logique) dans un test conditionnel,
- ABS(V) valeur absolue de V,
- AND ET logique,
- GOTO aller à (suivi d'un numéro de ligne),
- IF(C1) THEN I si la condition C1 est vraie, alors exécuter l'instruction I,
- INT(V) partie entière de V,
- PEEK(A) code contenu à l'adresse A,
- POKE A,P place le code P à l'adresse A,
- RETURN retour de sous-programme,
- RND(\emptyset) nombre aléatoire compris entre 0 et 1 (1 exclu),
- \emptyset zéro (pour le distinguer de la lettre O qui peut représenter une variable).

3. Liste des variables utilisées, les valeurs constantes sont données dans le cas du micro-ordinateur CBM :

- A adresse écran de la particule avant son déplacement,
- AF adresse écran de la future position de la particule,
- C nombre de colonnes ($C = 40$),
- D adresse écran du début de la zone de visualisation du phénomène ($D = 32809$) : ligne 1 colonne 1 compte tenu du cadre,
- DX éventuel déplacement horizontal,
- ED éventuel terme du déplacement horizontal dû au champ interne,
- EL code écran du signe « - » ($EL = 45$) : représentation d'un électron,
- NR nombre des recombinaisons déjà effectuées, initialement $NR = \emptyset$,
- P code écran en Å,
- P1 code écran en AF,
- R code écran de l'espace ($R = 32$),
- TR code écran de « + » ($TR = 43$) : représentation d'un trou,
- X numéro de la colonne où se trouve la particule.

4. Organisation du sous-programme de diffusion d'une particule.

* *Initialisation* des constantes et des variables (voir le paragraphe précédent).

* *REPETER* :

- | calcul du numéro d'une colonne aléatoire X,
- | calcul d'une position initiale aléatoire A,
- | détermination du contenu de cette position initiale P,

JUSQU'A ce que cette position soit occupée par un trou ou un électron.

* *REPETER* :

- | * calcul du déplacement dû au champ électrique ED,
- | * calcul du déplacement horizontal (terme de diffusion aléatoire et terme dû au champ) DX,
- | * calcul de l'éventuelle future position de la particule AF,
- | * détermination du contenu de cette future position P1,

* CHOIX :

si la future position est libre alors :

* effectuer le déplacement de la particule,
 * $A \leftarrow AF$ } mise à jour de la colonne
 * $A \leftarrow X + DX$ } et de la position,

si la future position n'est pas libre : ne rien faire.

FIN CHOIX

JUSQU'A ce que la future position AF soit occupée par une particule mobile de charge contraire à celle qui diffuse ;

- * effectuer la recombinaison : effacer le contenu de A et de AF,
- * fin du sous-programme de diffusion.

5. Le sous-programme de diffusion : accessible à partir du programme à l'aide de la commande GOSUB (branchement à un sous-programme), il simule la diffusion d'une particule mobile, choisie aléatoirement, jusqu'à sa recombinaison.

```

10000  X = INT((C-1)*RND(Ø)) :
        A = D + X + C * INT(NL * RND(Ø)) :
        P = PEEK(A) : IF P >< TR AND P >< EL THEN 10000,
10010  ED = Ø : IF NR > 50 AND X > 8 AND X < 22 THEN
        ED = 3 * (P-44),
10020  DX = INT(5 * RND(Ø) - 2) + ED :
        AF = A + DX + C * INT(3 * RND(Ø) - 1) :
        P1 = PEEK(AF),
10030  IF P1 = R THEN POKE AF, P : POKE A, R : A = AF :
        X = X + DX : GOTO 10010,
10040  IF ABS(P1 - P) >< 2 THEN 10010,
10050  POKE AF, R : POKE A, R : NR = NR + 1,
10060  RETURN.
```

III. POSSIBILITES INTERACTIVES DANS UN PROGRAMME.

Il est possible de prévoir à la fin de chaque séquence d'un programme d'enseignement des choix possibles, à la disposition de l'utilisateur, par exemple :

- recommencer la séquence,
- recommencer le programme,

- continuer,
- faire un exercice,
- faire des calculs avec l'ordinateur et revenir ensuite au programme,
- obtenir des précisions (notations, unités...),
- revoir les instructions permettant l'utilisation du programme,
- être aidé (cas d'un exercice),
- pause récréative (interlude).

Certains choix peuvent même être disponibles à chaque instant :

- figer le programme tant qu'une condition n'est pas réalisée,
- arrêter le programme (pause momentanée ou arrêt définitif).

On peut envisager différentes utilisations pour ces choix :

1. le professeur qui utilise le programme dans son cours en démonstration ou simulation doit rester libre de sa méthode pédagogique et de ses commentaires : il doit donc avoir accès à toutes les possibilités,
2. lors d'une « séquence de formation informatisée » (S.F.I.), l'élève étant seul face à la machine, le concepteur de didacticiels peut décider d'interdire certains choix pour éviter des recours systématiques :
 - aux séquences d'aide,
 - à l'interlude,
 - en général aux solutions de facilité.

IV. INTERETS PEDAGOGIQUES DES PROGRAMMES DE SIMULATION EN COURS DE PHYSIQUE.

1. Les programmes de simulation peuvent parfaitement s'intégrer à un cours traditionnel, sans en modifier la conception, il s'agit simplement d'utiliser un nouvel appareil. La limitation due actuellement à la taille des écrans peut être résolue soit en travaillant avec des groupes restreints, soit en utilisant un ou des écrans T.V. répéteurs de grand format.

2. La simulation ne peut en aucun cas remplacer l'expérience, surtout celle qui est faite par les élèves. Il ne faut pas tenter une prothèse informatique lorsque la méthode traditionnelle (expérimentale) est meilleure. Il n'est pas question de remplacer l'écran d'un oscilloscope par celui d'un ordinateur.

3. Certains phénomènes ne sont accessibles que par simulation pour des causes diverses :

- durée trop longue ou trop brève,
- dimensions du système trop grandes ou trop petites,
- dangers éventuels en cas de manipulation de certains produits.

Pour compléter cet inventaire, on peut dire que, chaque fois que le professeur utilise une représentation d'un système physique, au tableau noir, sous forme de tirages, ou à l'aide d'un rétroprojecteur, il pourrait se servir d'un micro-ordinateur, avec les avantages suivants :

- dynamisme possible de l'image,
- disponibilité d'un outil de calcul puissant.

4. L'utilisation de ce type de programmes suppose, de la part du professeur, une bonne connaissance des méthodes informatiques, pour pouvoir répondre aux questions des élèves, mais aussi pour être libre de choisir sa pédagogie.

V. CONCLUSION.

La micro-informatique apporte aux enseignants de toutes disciplines des possibilités pédagogiques nouvelles. Nous avons évoqué, en ce qui concerne le professeur de physique, la simulation d'un phénomène inaccessible expérimentalement, à travers l'exemple de la jonction P-N. D'autres points tels que :

- instrumentation,
- saisie et traitement de données expérimentales,
- commandes de machines diverses,

mériterait des études approfondies, car ils sont susceptibles de modifier et d'améliorer notre enseignement.

Il importe donc que les professeurs de physique ressentent la nécessité d'une formation aux méthodes informatiques et que les pouvoirs publics leur en donnent la possibilité.

J.-B. MARANINCHI, P. FAVRE-NICOLIN,
(*Lycée Vaucanson - Grenoble*).

BIBLIOGRAPHIE

Méthodes informatiques :

- « *Algorithmique et structures de données* », R. MAHL et J.-C. BOUSSARD (laboratoire d'informatique de l'université de Nice).
- « *Introduction à la programmation systématique* », N. WIRTH (Masson).

Langage BASIC :

- « *Lire BASIC* », de F.-M. BLONDEL (C.E.D.I.C. - Nathan).

Utilisation de l'informatique dans l'enseignement :

- Revue « *Education et informatique* » (C.E.D.I.C. - Nathan).
- *Bulletin de l'E.P.I.* (Association Enseignement public et Informatique).
- « *L'informatique au lycée* » : support de formation à l'utilisation de l'ordinateur dans les classes (ministère de l'éducation, direction des lycées).
- « *L'informatisation de la société* », S. NORA et A. MINC (Seuil).
- « *Informatique et pédagogie* », de M. LUMBROSO dans « *l'Ecole et la nation* », n° 298.
- « *La révolution télématique* », Science et Vie, n° 128.

Micro-ordinateurs CBM :

- « *La découverte du PET* », de D.-J. DAVID (éditions du PSI).
 - « *The PET revealed* » (Nick Hampshire).
-