

ÉCOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT D'ÉLECTRONIQUE

THESE

MAGISTER

Présentée par Madame TOUHAMI Rachida née MAHDI

INGÉNIEUR E.N.P.

SUJET

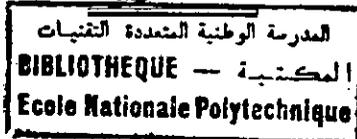
**ÉTUDE ET RÉALISATION D'UN PROGRAMME
D'ANALYSE TRANSITOIRE DES CIRCUITS
INTEGRES A M.O.S. : P.A.T.M.O.S.**

Soutenue le 25 Décembre 1989 devant le Jury composé de :

Mr A. CHEKIMA	Professeur	E.N.P.	Président
Mr B. DERRAS	Maître Assistant	E.N.P.	Examineur
Mr H. FARAH	Maître de conférences	E.N.P.	Examineur
Mr J. GORALSKI	Maître de conférences	E.N.P.	Examineur
Mr M. HADDADI	Chargé de cours	E.N.P.	Examineur
Mr R. GHEFFAR	Responsable à l'ENIE		Invité
Mr A. MERAGHNI	Maître de conférences		Rapporteur

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT D'ELECTRONIQUE



THESE

MAGISTER

Présentée par Madame TOUHAMI Rachida née MAHDI

INGENIEUR E.N.P.

SUJET

ETUDE ET REALISATION D'UN PROGRAMME
D'ANALYSE TRANSITOIRE DES CIRCUITS
INTEGRES A M.O.S. : P.A.T.M.O.S.

Soutenu le 25 Décembre 1989 devant le Jury composé de :

Mr A. CHEKIMA	Professeur	E.N.P.	Président
Mr B. DERRAS	Maître Assistant	E.N.P.	Examineur
Mr H. FARAH	Maître de conférences	E.N.P.	Examineur
Mr J. GORALSKI	Maître de conférences	E.N.P.	Examineur
Mr M. HADDADI	Chargé de cours	E.N.P.	Examineur
Mr R. GHEFFAR	Responsable à l'ENIE		Invité
Mr A. MERAGHNI	Maître de conférences		Rapporteur

A V A N T P R O P O S

Ce travail a été effectué au laboratoire des Télécommunications du Département d'Électronique de l'École Nationale Polytechnique d'Alger .

Je tiens à remercier vivement monsieur E. KARAKHANIAN, professeur pour ses précieux conseils.

Que monsieur A. MERAGHNI, maître de conférences, à l'ENES de Kouba, trouve ici la marque de ma gratitude pour l'aide qu'il m'a apporté .

Je remercie monsieur le professeur A. CHEKIMA, président du conseil scientifique au Département d'Electronique de l'E.N.P.A. d'avoir accepté la présidence de ce jury .

Je remercie également messieurs :

- B. DERRAS Maitre assistant
- H. FARAH Maitre de conférences
- J. GORALSKI Maitre de conférences

à l'E.N.P.A. d'avoir bien voulu accepter de siéger à ce jury

Je remercie également monsieur R. GHEFFAR, directeur à l'E.N.I.E. de Blida, d'avoir accepté l'invitation .

Que monsieur D. BERKANI, chef de Département d'Electronique trouve ici l'expression de ma profonde gratitude pour avoir mis à ma disposition tous les moyens nécessaires.

T A B L E D E S M A T I E R E S

CHAPITRES	PAGES
Table des matières	1
Glossaire	6
Introduction Générale	9
 Chapitre I	 11
 MODELISATION DU TRANSISTOR MOS 	
I/ Introduction	11
II/ Différentes représentations d'un modèle	12
III/ Les modèles élaborés	14
III.1/ Principe	14
III.2/ Les modèles mathématiques et physiques	15
IV/ Modèles linéaires-numériques du transistor MOS sans effet de capacités de structure	17
IV.1/ La méthode de Newton pour une fonction à une seule variable	17
a/ Principe	18
b/ Interprétation graphique	19
c/ Application au transistor MOS	19
e/ Modèle électrique associé	20
IV.2/ La méthode de Newton pour une fonction à plusieurs variables	20
a/ Principe	20
b/ Application au transistor MOS	21
V/ Modèle linéaire-numérique du transistor MOS avec effet des capacités entre électrodes	23
V.1/ Modèle linéaire-numérique du transistor MOS avec les capacités C_{gd} et C_{gs} linéaires (deuxième approximation)	24

V.2/	Modèle linéaire-numérique du transistor MOS avec les capacités Cgd et Cgs non linéaires (troisième approximation)	27
VI/	Vérification expérimentale du modèle numérique-linéaire	28
VI.1/	Méthode de détermination des paramètres en régime de saturation	29
VI.1.1/	Détermination de V_t	29
VI.1.2/	Détermination de n	30
VI.1.3/	Détermination de K_e	30
VI.2/	Application au transistor MOS 2N4351	30
VI.3/	Evaluation de la précision du modèle numérique-linéaire	31
VII/	Conclusion	32
Chapitre II		34

ANALYSE ET FORMULATION DES EQUATIONS

I/	Introduction	34
II/	Analyse et formulation des équations	34
II.1/	Analyse des circuits	36
II.1.1/	Analyse aux noeuds	36
II.1.2/	Analyse aux mailles	37
II.1.3/	Analyse des sections	37
II.2/	Technique de formulation	37
II.2.1/	Par les variables d'état	37
II.2.1.1/	Théorie	37
II.2.1.1.1/	Les circuits linéaires	38
II.2.1.1.2/	Les circuits non linéaires	37

II.2.1.2/ Application aux circuits intégrés à MOS	40
II.2.1.2.1/ Formulation des équations d'état d'un C.I à MOS avec le modèle non linéaire du transistor MOS	40
II.2.1.2.2/ Formulation des équations d'état d'un C.I à MOS avec le modèle numérique-linéaire du transistor MOS	44
II.2.2/ Formulation des équations par la théorie des graphes	46
II.2.2.1/ Théorie	46
II.2.2.1.1/ Les circuits resistifs linéaires	47
II.2.2.1.2/ Les circuits non linéaires	50
II.2.2.2/ Application aux C.I à MOS	51
III/ Les algorithmes d'intégration numérique	54
III.1/ Algorithme d'ADAMS-BASHFORTH	54
III.2/ Algorithme d'ADAMS-MOULTON	56
III.3/ Choix optimum de l'ordre et des dimensions du pas h	58
III.4/ Régions de stabilité des algorithmes d'intégration numérique	59
IV/ Conclusion	62
Chapitre III	64

ELABORATION D'UN ALGORITHME D'ANALYSE TEMPORELLE
 DES CIRCUITS INTEGRES à MOS

I/ Introduction	64
II/ Elaboration d'un algorithme de formulation d'un système d'équations linéaires d'analyse des C.I à MOS (sans capacités entre électrodes)	65
II.1/ Principe de base	65
II.2/ La méthode d'analyse	65

II.3/ La méthode d'intégration 67

II.4/ Formulation d'un système d'équations 68

II.4.1/ Les éléments de description topologique
 du C.I à MOS 68

II.4.1.1/ Les matrices topologiques 69

II.4.1.2/ Les vecteurs topologiques 72

II.4.2/ Elaboration d'un algorithme de
 formulation automatique des équations
 linéaires d'analyse 74

II.4.2.1/ Formulation de la matrice conductance G_c 76

II.4.2.2/ Formulation du vecteur courant J 79

III/ Modification de l'algorithme de formulation 80

IV/ Etude numérique des méthodes de résolution
 d'un système d'équations linéaires 85

IV.1/ Les méthodes directes 85

IV.2/ Les méthodes itératives 86

IV.3/ Les méthodes d'optimisation 89

V/ Conclusion 91

Chapitre IV 92

PATMOS : PROGRAMME D'ANALYSE DES CIRCUITS
 INTEGRES A MOS ET SES APPLICATIONS

I/ Introduction 92

II/ Les modèles du transistor MOS 93

III/ Stratégie du choix optimum des dimensions
 du pas d'intégration h 95

IV/ Organisation des données d'entrées-matrices
 et vecteurs topologiques 98

V/ Algorithme-organigramme d'analyse des C.I à MOS 101

VI/ Applications du programme PATMOS 103

UII/ Vérification expérimentale du programme PATMOS . . . 109

Chapitre V 111

Conclusion générale

Annexe 114

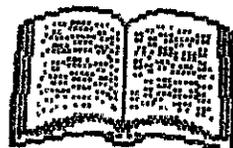
Bibliographie 118

G L O S S A I R E

- I_d : courant drain (A)
 I_s : courant source (A)
 I_g : courant gate (A)
 V_d : tension drain (V)
 V_s : tension source (V)
 V_g : tension gate (V)
 V_{ds} : différence de potentiel entre le drain et la source (V)
 V_{gd} : différence de potentiel entre le gate et le drain (V)
 V_{gs} : différence de potentiel entre le gate et la source (V)
 V_t : tension de seuil (V)
 V_e : tension effective (V)
 K_e : conductivité spécifique effective ($\mu A/V^2$)
 K : conductivité spécifique ($\mu A/V^2$)
 η : effet du substrat
 t_m : temps de montée (s)
 t_d : temps de descente (s)
 t_{op} : durée d'une opération (s)
 N_n : nombre de noeuds dans un circuit intégré à MOS
 N_t : nombre de transistors MOS dans un circuit intégré
 N_a : nombre d'opération d'addition
 N_m : nombre d'opération de multiplication
 N_d : nombre d'opération de division
 N_p : nombre de paramètres du modèle physique utilisé
 n : indice pour la discrétisation temporelle
 m, p : indice pour les itérations de Newton

- C_{gs} : capacité entre le gate et la source (F)
 C_{gd} : capacité entre le gate et le drain (F)
 C_P : capacité parasite (F)
 τ : pas d'intégration (s)
 G : électrode gate
 D : électrode drain
 S : électrode source
 sb : électrode substrat
 W : la largeur du canal (m)
 L : la longueur du canal (m)
 U_{gsb} : différence de potentiel entre la source et le substrat
 EM : espace memoire
 b : nombre de branches dans un graphe
 N : nombre de noeuds dans un graphe
 L : matrice triangulaire supérieure
 U : matrice triangulaire inférieure
 μ : mobilité des porteurs de charge
 C_e : capacité du SiO₂ par unité de surface (F/cm²)
 ϵ_s : constante diélectrique du SiO₂
 ϵ_i : constante diélectrique de l'isolateur
 N' : concentration du dopage (substrat)
 q : charge de l'électron (C)
 d : épaisseur du SiO₂ (m)
 L_{eff} : la longueur effective du canal (m)
 U_{db} : la différence de potentiel entre le drain et substrat (U)
 ϕ_f : potentiel du niveau de Fermi (U)
 ψ : potentiel de la surface du canal (U)

Introduction Generale



I N T R O D U C T I O N

Le développement des simulateurs des circuits intégrés est en perpétuel croissance. Ils assistent les concepteurs dans le choix des paramètres physico-géométriques pour l'obtention d'un régime temporel optimal. Ils utilisent soit la technique transitoire, soit la technique génération du chronogramme.

Ces simulateurs /1/-/4/ ont été élaborés au cours des deux dernières décennies et d'autres versions continuent pas moins de se développer.

Leur but est de répondre aux problèmes des circuits d'une certaine classe. Ils sont soit à niveau logique, soit à niveau électrique :

Les simulateurs MOTIS /5/, MOTIS-C /6/, et MOSTAP /7/ utilisent la technique logique. Elle est basée sur le principe de la macromodélisation, ou décomposition du circuit intégré en sous-circuit de base (ex: inverseurs, portes NAND, portes NOR, ...).

Cette approche /8/-/9/ est rapide mais non universelle car certains circuits intégrés (ex: mémoires, buffers,) ne sont pas décomposables en élément de base macromodèle. L'analyse de leur performance électrique est complexe car il n'existe pas de lien direct entre le régime transitoire et les composants de base (transistors, capacités).

Contrairement à la simulation à niveau logique, celle à niveau électrique permet un lien direct entre la réponse d'un point du circuit et les composants qui lui sont connectés. Ce qui rendent les simulateurs à niveau électrique plus précis.

Dans cette étude, nous proposons un Programme universel d'Analyse

Transitoire non linéaires des circuits intégrés à MOS : PATMOS .

Ce logiciel conversationnel simule les circuits à partir uniquement d'une description topologique. Cette dernière est un ensemble de transistors MOS (type N, type P ou les deux types) et les capacités parasites représentées entre le noeud et la masse, c'est un simulateur statique et dynamique.

Avec le progrès de la technologie moderne de miniaturisation, le transistor MOS a été modélisé par plusieurs approches mathématiques. Ces modèles retenus pour PATMOS (1^{ère} étape) sont simples et précis (erreur relative de 5% à 30%).

Les problèmes d'analyse sont résolus en deux pas :

- Le premier consiste en la formulation des équations différentielles non linéaires utilisant les lois de KIRCHHOFF et les caractéristiques des éléments électriques.

- Le second, à résoudre ces équations par des techniques analytiques (circuits à faibles dimensions) ou numériques (circuits larges).

Jusqu'à présent tous les logiciels d'analyse /10/-/11/ forment un système d'équations différentielles non linéaires. Leur résolution est donnée par les algorithmes Backward-Euler /21/, Newton Raphson /22/ ou de relaxation GS/GJ /24/. Le choix de ces approches est un compromis entre la convergence, la stabilité, la vitesse de calcul et l'espace mémoire .

En tenant compte de ce compromis, PATMOS est basé sur :

- Le modèle numérique linéaire du transistor MOS (première et deuxième approximation).
- L'algorithme de Backward-Euler.
- La formulation automatique des équations linéaires.

- La méthode de résolution des équations par le gradient conjugué.

Ceci nous mène à établir notre plan de travail comme suit :

Dans un premier chapitre, nous présenterons la conception du modèle numérique du transistor M.O.S qui comprend deux approximations (transistor simple, transistor avec C_{gs} et C_{gd}) ainsi qu'une étude sur les modèles mathématiques et physiques .

Au deuxième chapitre, le choix de l'algorithme d'analyse résulte d'un compromis entre la convergence, le temps de calcul et l'espace mémoire. Nous étudierons principalement les méthodes d'analyse, de formulation et d'intégration pour l'élaboration d'un programme universel avec temps de réponse et espace mémoire optimaux.

Au troisième chapitre : Les méthodes de formulation des équations par la théorie des graphes et la variable d'état, présentent des contraintes pour l'élaboration d'un logiciel universel. Nous avons jugé utile d'élaborer un algorithme de formulation automatique des équations d'analyse pour le transistor M.O.S simple, le transistor avec capacités entre électrodes. Une étude numérique sur les méthodes de résolution des équations linéaires nous a permis l'amélioration de ce programme.

Au quatrième chapitre, l'optimisation a été faite à partir :

- de l'élaboration des logiciels de conversion topologie / données de description,
- du choix d'un calcul automatique du pas d'intégration, permettant de diminuer le temps d'exécution du PATMOS.

Les vérifications expérimentales nous ont permis d'évaluer la précision avec laquelle le circuit intégré est simulé.

Chapitre 1



MODELISATION DU TRANSISTOR MOS

1.2 Introduction

Le dispositif semi-conducteur est caractérisé par une série de paramètres physiques. A partir de cette série, la modélisation élabore un certain nombre d'équations : représentation analytique des caractéristiques. Cependant, le modèle permet au concepteur de comprendre comment le transistor opère d'une part, de comprendre le mécanisme interne d'autre part.

Avec le progrès de la technologie, la réduction des dimensions géométriques a attiré l'attention sur deux points importants pour la conception des circuits intégrés à M.O.S. Le premier point, la théorie du transistor M.O.S. est insuffisante pour modéliser un composant miniaturisé. Le second, il n'existe pas de méthode effective et économique à part le programme d'analyse pour la conception des C.I. à M.O.S qui nécessite un modèle simple et précis.

Tous les modèles (mathématiques ou physiques) se répartissent en deux groupes /12/ :

-Les modèles complexes et précis pour la conception du transistor M.O.S. miniaturisé

-Les modèles simples et peu précis élaborés pour les programmes d'analyse des C.I. à M.O.S de première étape.

L'élaboration d'un modèle est basée sur les points suivants :

1) Le développement des équations modélisants le dispositif doit satisfaire les critères de précision et de complexité, sachant que le temps de calcul augmente avec la complexité du modèle.

2) Les modèles du transistor M.O.S. qui relient les paramètres physiques doivent être courants et accessibles par la conception.

3) L'intégration du modèle de simulation doit se faire de façon efficace.

Il faut souligner que la modélisation du transistor M.O.S basé sur des phénomènes physiques est complexe et; la structure du transistor M.O.S change avec sa miniaturisation, le modèle doit être réenvisagé pour chaque structure.

Dans ce présent chapitre, nous traitons le problème de modélisation du transistor M.O.S pour le programme d'analyse des circuits intégrés numériques à M.O.S.. Il existe plusieurs façon de représenter le modèle selon l'objet de son utilisation, ces différentes représentations sont introduites afin de choisir celle(s) qui est la plus convenable pour le programme d'analyse.

Notre programme nécessite un modèle basé sur le principe de fonctionnement du transistor M.O.S., nous présentons quelques modèles élaborés afin de construire une bibliothèque de modèles pour le programme d'analyse.

La mise en oeuvre d'un algorithme d'analyse des C.I. exige un modèle numérique-linéaire du transistor M.O.S, utilisant indifféremment l'un ou l'autre des modèles de la bibliothèque. Des modèles numériques-linéaires avec et sans effets de capacités entre les électrodes sont développés.

II/2 Différentes représentations d'un modèle

Selon le but de modélisation, le transistor M.O.S possède

quatre représentations :

- 1) Modèle sous forme de tableau
- 2) Modèle sous forme de courbes
- 3) Modèle sous forme d'équations
- 4) Modèle sous forme de circuit électrique

La représentation sous forme de tableau est utilisée essentiellement dans certains programmes d'analyse [2], pour diminuer le temps de calcul. Cette représentation est limitée, car elle exige un espace mémoire important pour la technologie LSI, pour chaque transistor MOS il faut donner son modèle sous forme d'un tableau de valeurs. Pour palier à cet inconvénient, nous supposons que les N transistors d'un circuit intégré sont représentés par P modèles avec $P \ll N$, afin de réduire l'espace mémoire. Cependant la miniaturisation du transistor n'est pas limitée, la structure change, les P modèles ne sont plus valables, et cette représentation ne peut être utilisée dans les années à venir.

La représentation sous forme de courbes est intéressante pour extraire les paramètres physiques d'un transistor. Comme cette représentation est souvent obtenue à partir des mesures expérimentales, alors elle permet de déterminer l'exactitude d'un modèle théorique.

La troisième représentation est couramment utilisée. Elle permet d'observer l'influence d'un paramètre physique sur le mécanisme du dispositif. Pour les programmes, il est plus simple d'écrire des équations que d'introduire les données (tableau de valeurs). La dernière est utilisée pour l'analyse d'un circuit basé sur la théorie des graphes. Si le nombre de composants est élevé, alors cette représentation devient encombrante.

Pour les différentes représentations, le passage de l'une à l'autre est réversible. Les représentations adoptées par les programmes d'analyse sont : modèle sous forme d'équations et celui du modèle électrique.

III/ Les modèles élaborés

III.1/ Principe

Pour la compréhension du mécanisme du transistor MOS, les premiers chercheurs ont élaboré à partir des paramètres physiques et de la résolution de l'équation de Poisson, des modèles physiques et mathématiques unidimensionnels. Les paramètres empiriques sont utilisés dans le modèle mathématique, par contre celui physique est développé à partir des phénomènes physiques.

Le progrès de la technologie, a entraîné une diminution graduelle des dimensions du transistor MOS pour lequel la théorie unidimensionnelle est insuffisante. Cependant la modélisation du transistor MOS miniaturisé constitue un thème de recherche pour ces dernières décennies.

Pour choisir un modèle, plusieurs considérations sont désirées, parmi lesquelles, nous citons :

- 1) La procédure d'extraction des paramètres doit être simple
- 2) Une certaine précision du modèle, pour une géométrie donnée
- 3) Une certaine simplicité du modèle afin que la simulation ne soit pas lente.

Le programme d'analyse à élaborer possède une bibliothèque de modèles. Ceci permet à l'utilisateur de changer le modèle mathématique ou physique sans changer la structure du programme. Alors on propose quelques modèles du transistor MOS nécessaires pour le programme d'analyse.

III.2 Les modèles mathématiques et physiques

Un nombre important de modèles du transistor MOS ont été proposé au cours des deux dernières décennies. Notre but est d'élaborer un programme d'analyse, alors nous considérons uniquement les modèles du transistor MOS conçus pour la CAO des C.I. à MOS. Ces modèles sont établis avec simplicité et précision.

Parmi les modèles élaborés; le modèle de Sah /13/ unidimensionnel, basé sur le principe de fonctionnement du transistor MOS donne les caractéristiques $I_d=f(V_d)$ en négligeant l'effet du substrat cependant il tient compte de l'effet des capacités entre électrodes. Sa précision relative est de 30%. Ce dernier peut être amélioré en supposant que : $V_t=f(V_{sb}, W, L')$ et $K=h(V_{gs}, V_{ds})$.

En 1966, Sah et Pao ont proposé un modèle simple de 25%, tenant compte de l'effet du substrat /14/.

La variation de la mobilité en fonction du champs électrique à travers la surface SiO_2 a été prise en compte en 1969 par Frohman-Benchkowsky /15/, avec une précision de 15%. Ce modèle a été utilisé par le programme d'analyse basé sur la macromodélisation. Une analyse unidimensionnelle a été utilisée pour trouver les limites inférieure et supérieure du courant de drain en tenant compte des régions de déplétion de la source et du drain et de la vitesse de saturation des porteurs de charges. Ceci constitue le modèle du transistor MOS de faibles dimensions élaboré par B.Hoeneisein et CA.Mead /17/. G.Merckel, J.Borel et NE.Cupcea /18/ proposent en 1972 un modèle complet du transistor MOS pouvant être simplifié avec une précision désirée (5%). Ce modèle tient compte de l'effet du champs électrique (normale ou

parallèle à la surface) sur la mobilité, avant la saturation. L'effet de la mobilité des porteurs sur la charge d'espace drain-canal est présenté en utilisant l'approximation bidimensionnelle /6/. Il est intégré dans le programme d'analyse des C.I. à MOS IMAG II, et des résultats satisfaisants sont obtenus avec ce modèle. G.Merckel /19/ propose un autre modèle du transistor miniaturisé (1-2 μ m) tenant compte de la variation de la tension de seuil V_t en fonction des dimensions du canal.

Un modèle de 1% à 3% /20/ a été présenté par Marvin H.White et F. Van de Wiele pour décrire les caractéristiques $I_d=f(V_d)$, les résultats de ce modèles était acceptables pour les dimensions considérées. Une procédure de modélisation, de mesure et d'extraction des paramètres a été proposée pour le programme d'analyse. Vu que la miniaturisation du transistor ne s'arrête pas, la structure change avec les dimensions. Ceci conduit à un nombre de modèles non limité.

Les modèles ne se limitent pas à ceux précédemment présentés, en effet l'objet de notre recherche est de connaître une bibliothèque de modèles ayant une structure simple pour la simulation. L'étude bibliographique de ces différents modèles nous a permis de présenter un synoptique (Fig. 1.1) qui donne l'objet d'utilisation de chaque modèle. Les modèles complexes sont utilisés essentiellement dans la CAO du transistor MOS. Vu que le programme d'analyse à réaliser est une simulation de première étape on se contentera d'utiliser le modèle de Sah-Pao pour toutes les applications, ce dernier peut être amélioré en considérant V_t comme une fonction de la tension source-substrat, de la largeur et de la longueur du canal.

IV/ Modèle linéaire-numérique du transistor MOS sans effet de capacités de structure

D'après la (Fig.1.1), entre l'étage programme d'analyse et les étages modèles, il existe un bloc intermédiaire : modèle numérique.

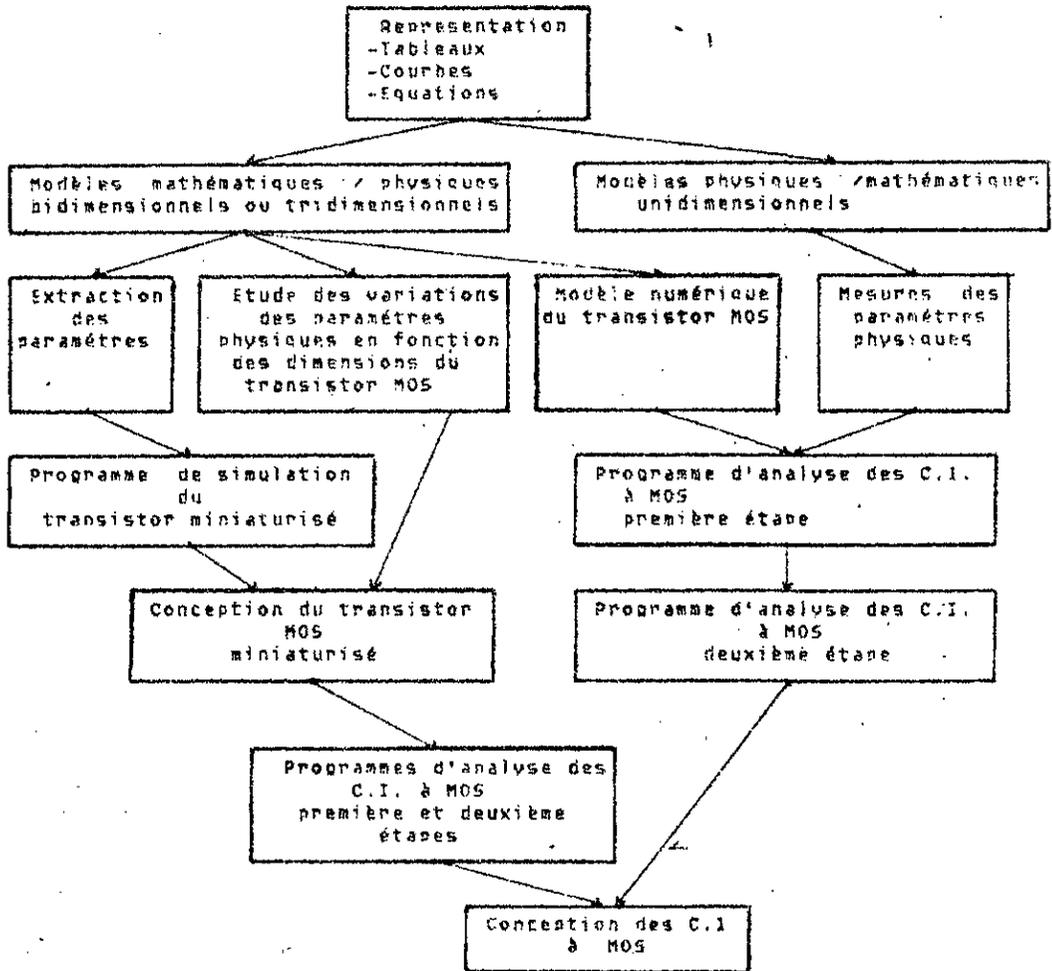


Fig. 1.1 Représentation et utilisation des différents modèles du transistor MOS

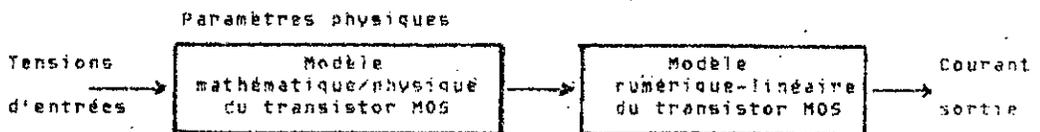


Fig. 1.2 Représentation du modèle global

Pour élaborer un programme universel nous sommes contraints de réaliser un modèle numérique-linéaire du transistor MOS.

L'étude des modèles physiques du transistor MOS, permet de constater que dans tous les cas, le courant I_{ds} est donné par une fonction non linéaire. Un algorithme est appliqué à cette fonction pour la linéariser. L'ensemble modèle numérique et modèle physique constituent un modèle global pour la simulation du transistor MOS (Fig. 1.2).

Le modèle numérique-linéaire doit contenir uniquement des grandeurs électriques (courant, tension et conductance) et être indépendant des paramètres physiques.

Dans l'élaboration du modèle numérique-linéaire du transistor MOS, nous utilisons le principe de la méthode de Newton [21], qui est simple, stable et converge quadratiquement.

IV.1/ La méthode de Newton pour une fonction à une seule variable

a) Principe

Soit f une fonction continue et continument dérivable :

$$y = f(x) \quad \text{où } f : \mathbb{R} \rightarrow \mathbb{R}$$

f est une fonction non linéaire. En appliquant le développement de TAYLOR au voisinage de x^0 nous aurons :

$$f(x) = f(x^0) + \left(\frac{df}{dx}\right)_{x^0} (x-x^0) + \frac{1}{2!} \left(\frac{d^2f}{dx^2}\right)_{x^0} (x-x^0)^2 + \dots + \frac{1}{n!} \left(\frac{d^n f}{dx^n}\right)_{x^0} (x-x^0)^n + \dots \quad (1.1)$$

Pour linéariser on se limitera au premier ordre :

$$f(x) = f(x^0) + \left(\frac{df}{dx}\right)_{x^0} (x-x^0) + 0(x-x^0)^2 \quad (1.2)$$

pour $x=x^1$ on a :

$$f(x^1) = f(x^0) + \left(\frac{df}{dx}\right)_{x^0} (x^1 - x^0) \quad (1.3)$$

$$f(x_1) = y_1 \quad \text{et} \quad f(x_0) = y_0$$

$$y_1 = y_0 + \left. \left(\frac{df}{dx} \right) \right|_{x_0} (x_1 - x_0) \quad (1.4)$$

pour la $(m+1)$ itération nous avons :

$$y_{m+1} = y_m + \left. \left(\frac{df}{dx} \right) \right|_{x_m} (x_{m+1} - x_m) \quad (1.5)$$

b) Interprétation graphique

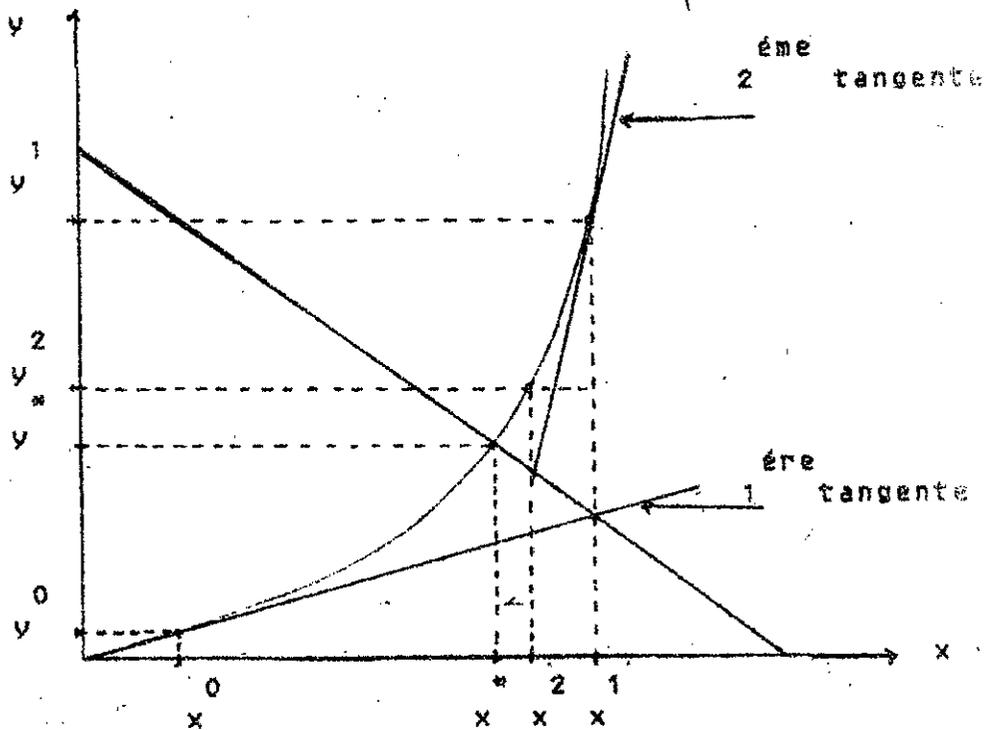


Fig. 1.3 Interprétation graphique de la méthode de Newton

c) Application au transistor MOS

Nous supposons que le courant I_d depend uniquement de V_d , $V_g = 0$ et $V_s = \text{conste}$, V_{sb} est la tension de référence (masse), nous avons :

$$I_d = f(V_d)$$

La formule de Newton donne :

$$I_{d,m+1} = I_{d,m} + \left. \left(\frac{dI_d}{dV_d} \right) \right|_{V_{d,m}} (V_{d,m+1} - V_{d,m}) \quad (1.6)$$

d) Modèle électrique

Le modèle du transistor est :
$$I_d = I_d^{m+1} + (dI_d/dV_d) \begin{matrix} m+1 & m \\ (V_d - V_d) \\ m \\ V_d \end{matrix}$$

Si on pose $dI_d/dV_d = G_d^m$, le modèle numérique-linéaire à une seule

variable est :

$$(I) \quad \begin{cases} I_d = I_d^{m+1} + G_d^m V_d^m - G_d^m V_d^m \\ I_s = I_d^{m+1} \\ I_g = 0 \end{cases}$$

Le modèle électrique associé au système d'équation est :

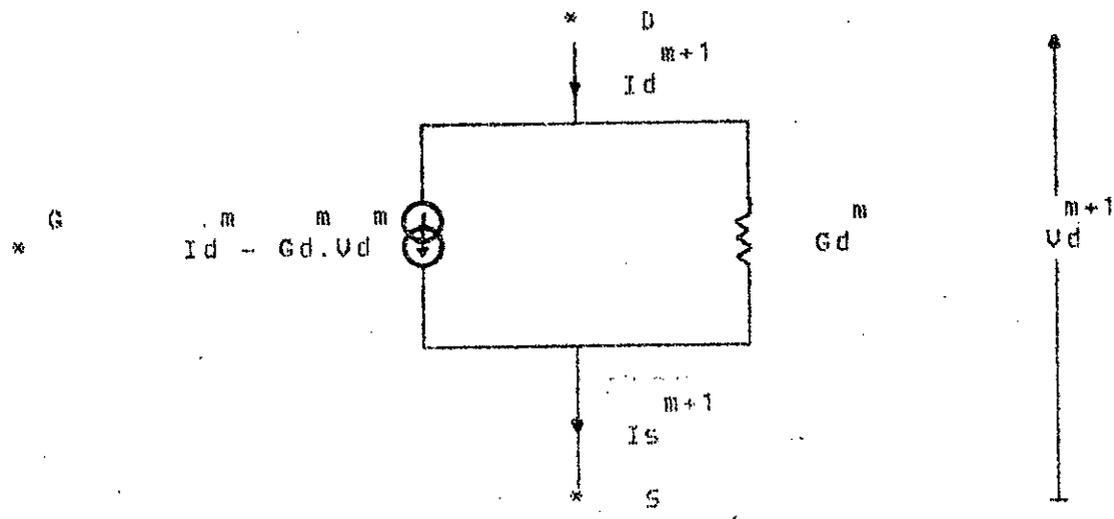


Fig.1.14 Modèle électrique associé

IV.2/ Méthode de Newton pour une fonction à plusieurs variables

a) Principes

Soit une fonction f de plusieurs variables : $y = f(x_1, x_2, x_3, \dots, x_p)$

L'équation différentielle de f :

$$dy = \left(\frac{\partial f}{\partial x_1}\right) dx_1 + \left(\frac{\partial f}{\partial x_2}\right) dx_2 + \dots + \left(\frac{\partial f}{\partial x_p}\right) dx_p \quad (1.1)$$

Si nous faisons l'approximation de $d..$ par $\Delta..$ nous obtiendrons :

$$\Delta y = \left(\frac{\partial f}{\partial x_1}\right) \begin{matrix} m \\ \Delta x_1 \\ m \\ x_1 \end{matrix} + \left(\frac{\partial f}{\partial x_2}\right) \begin{matrix} m \\ \Delta x_2 \\ m \\ x_2 \end{matrix} + \dots + \left(\frac{\partial f}{\partial x_p}\right) \begin{matrix} m \\ \Delta x_p \\ m \\ x_p \end{matrix} \quad (1.2)$$

avec $\Delta^m y = y^{m+1} - y^m$ et $\Delta^m x = x^{m+1} - x^m$ pour $i=1,2,\dots,p$

nous remplaçons les par leurs expressions et nous aurons :

$$y^{m+1} - y^m = \left(\frac{\partial f}{\partial x} \right)_1 \left(x^{m+1} - x^m \right) + \left(\frac{\partial f}{\partial x} \right)_2 \left(x^{m+1} - x^m \right) + \dots + \left(\frac{\partial f}{\partial x} \right)_p \left(x^{m+1} - x^m \right) \quad (1.9)$$

alors la valeur de y est donnée par :

$$y^{m+1} = y^m + \left(\frac{\partial f}{\partial x} \right)_1 \left(x^{m+1} - x^m \right) + \left(\frac{\partial f}{\partial x} \right)_2 \left(x^{m+1} - x^m \right) + \dots + \left(\frac{\partial f}{\partial x} \right)_p \left(x^{m+1} - x^m \right) \quad (1.10)$$

L'itération finale m_f est donnée par la condition (test d'arrêt)

suivante :

$$| y^{m_f+1} - y^{m_f} | < \epsilon$$

L'écriture vectorielle donne la forme suivante :

$$y^{m+1} = y^m + \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_p} \right] \begin{bmatrix} x^{m+1} - x^m \\ x^{m+1} - x^m \\ \vdots \\ x^{m+1} - x^m \end{bmatrix} \quad (1.11)$$

$$y^{m+1} = y^m + \text{grad } f(x) \Delta^m x \quad \text{où } x = (x_1, x_2, \dots, x_p)$$

et $\text{grad } f(x) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_p} \right]$

b) application au transistor MOS

Le courant Drain-Source est une fonction non linéaire des tensions V_d, V_s, V_g et V_{sb} :

$$I_d = f(V_d, V_s, V_g, V_{sb})$$

Nous considérons V_{sb} comme tension de référence (masse).

En appliquant l'algorithme de Newton à $f(V_d, V_s, V_g)$ on obtient :

$$I_d = I_d + \left(\frac{\partial f}{\partial V_d} \right) (V_d - V_d) + \left(\frac{\partial f}{\partial V_s} \right) (V_s - V_s) + \left(\frac{\partial f}{\partial V_g} \right) (V_g - V_g) \quad (1.12)$$

où bien

$$I_d^{m+1} = I_d^m + \text{grad} f(V_d, V_s, V_g) \cdot \begin{bmatrix} V_d - V_d^{m+1} \\ V_s - V_s^{m+1} \\ V_g - V_g^{m+1} \end{bmatrix} \quad (1.13)$$

Nous posons :

$$\begin{aligned} G_d^m &= (\partial f / \partial V_d) \Big|_{V_d, V_s, V_g} \\ G_s^m &= (\partial f / \partial V_s) \Big|_{V_d, V_s, V_g} \\ G_g^m &= (\partial f / \partial V_g) \Big|_{V_d, V_s, V_g} \end{aligned}$$

où G_d^m , G_s^m et G_g^m représentent les conductances du modèle du transistor MOS.

Le transistor n'est pas caractérisé uniquement par le courant I_d mais par le système d'équations suivant :

$$\begin{cases} I_d = I_d^{m+1} + (\partial f / \partial V_d) (V_d - V_d^{m+1}) + (\partial f / \partial V_s) (V_s - V_s^{m+1}) + (\partial f / \partial V_g) (V_g - V_g^{m+1}) \\ I_s = I_s^{m+1} \\ I_g = I_g^{m+1} = 0 \end{cases} \quad (II)$$

En remplaçant les dérivées partielles par les conductances, le système devient :

$$\begin{cases} I_d = I_d^m + G_g^m (V_g - V_g^{m+1}) + G_d^m (V_d - V_d^{m+1}) + G_s^m (V_s - V_s^{m+1}) \\ I_s = I_s^m + G_g^m (V_g - V_g^{m+1}) + G_d^m (V_d - V_d^{m+1}) + G_s^m (V_s - V_s^{m+1}) \\ I_g = I_g^m = 0 \end{cases} \quad (III)$$

On transforme (en ajoutant et en retranchant des tensions) les équations (1.14) et (1.15), le système devient :

$$\begin{cases} I_d = I_d^m + G_g^m (V_{gd} - V_{gd}^{m+1}) + G_s^m (V_{sd} - V_{sd}^{m+1}) + (G_s^m + G_d^m + G_g^m) (V_d - V_d^{m+1}) \\ I_s = I_s^m + G_g^m (V_{gs} - V_{gs}^{m+1}) + G_s^m (V_{sd} - V_{sd}^{m+1}) + G_g^m (V_s - V_s^{m+1}) + (G_s^m + G_d^m) (V_d - V_d^{m+1}) \\ I_g = I_g^m = 0 \end{cases} \quad (IV)$$

A partir de ce système d'équation, on élabore le circuit électrique associé au modèle numérique du transistor MOS (première approximation Fig. 1.5). Il est composé de 6 sources de courant dépendantes, une source de tension dépendante et 9 conductances.

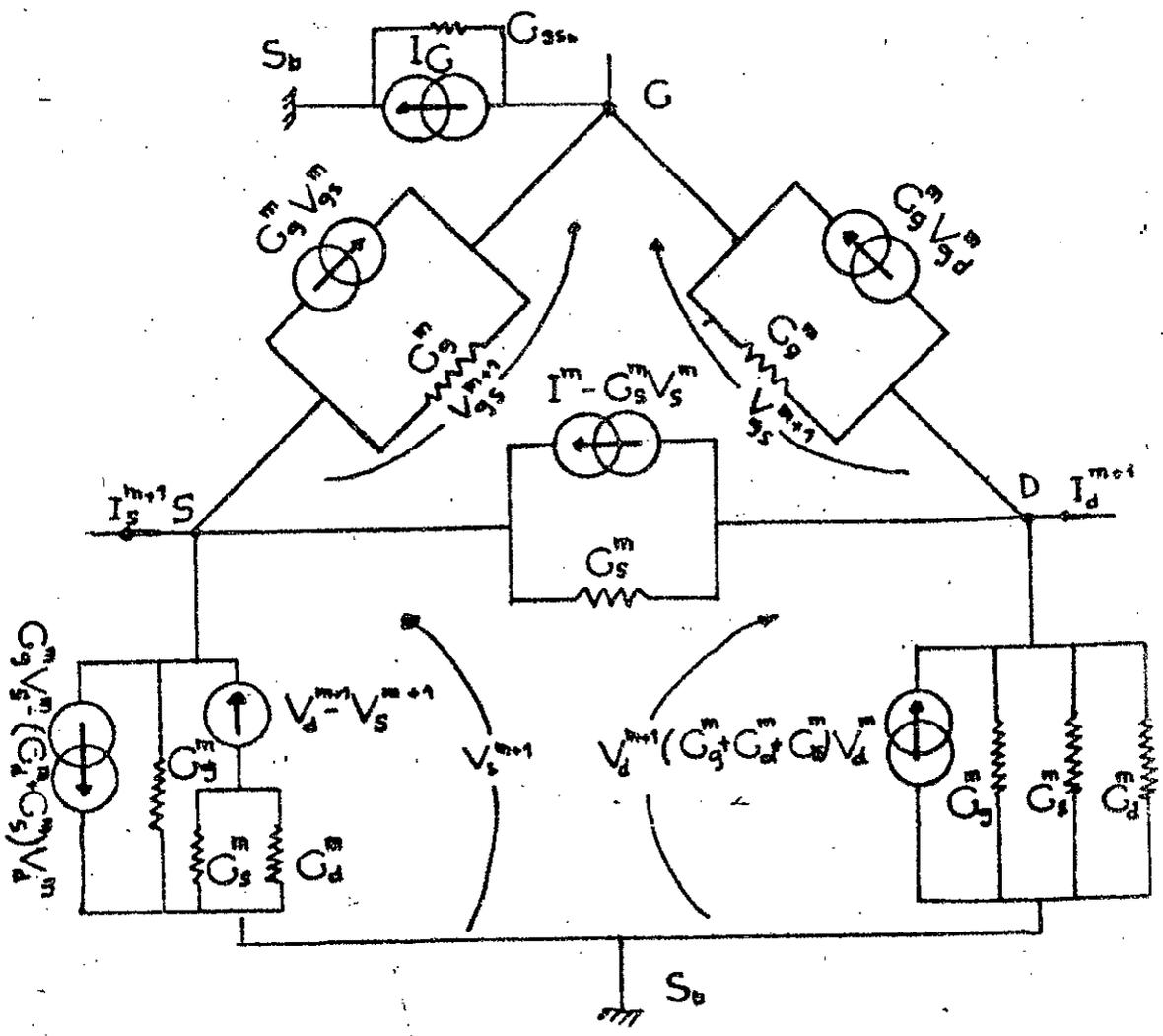


Fig. 1.5 Modèle électrique associé au modèle numérique linéaire du transistor MOS (premier approche)

V2. Modèle linéaire-numérique du transistor MOS avec les capacités entre électrodes C_{gs} et C_{gd}

Dans le modèle précédent, nous avons négligé l'effet de capacités entre électrodes, des capacités de chevauchement, et des capacités de jonction et de liaisons. Il est justifié par

le fait que dans un C.I les capacités des noeuds (parasites) sont plus grandes.

Dans certains C.I. à MOS, les capacités entre électrodes sont comparables aux capacités parasites (noeud-masse), et le modèle ainsi établi est insuffisant pour représenter correctement le transistor dans un programme de simulation d'où la nécessité d'établir un modèle numérique plus complet tenant compte de l'effet des capacités entre électrodes, ce qui conduit à considérer deux cas :

- capacités entre électrodes sont linéaires.
- capacités entre électrodes sont non linéaires.

Chaque cas donne un modèle numérique-linéaire modifié.

U.1/ Modèle numérique-linéaire du transistor MOS avec les capacités C_{gs} et C_{gd} linéaires (deuxième approximation)

Les courants des capacités linéaires sont donnés par:

$$I_{gd} = C_{gd} \cdot dV_{gd}/dt \quad (1.16)$$

$$I_{gs} = C_{gs} \cdot dV_{gs}/dt \quad (1.17)$$

Or que le modèle du transistor (sans effet de capacités) est numérique, alors ces équations ne peuvent être résolues que par un des algorithmes d'intégration numérique /22/.

Pour une équation de la forme $\dot{x} = f(x)$, son intégration par Backward-Euler est :

$$x^{n+1} = x^n + h \cdot f(x^{n+1})$$

où h représente le pas d'intégration et x^{n+1} la valeur de x à l'instant t^{n+1} . En appliquant cette intégration à (1.16) et

(1.17) on obtient :

$$V_{gd}^{n+1} = V_{gd}^n + h \cdot f(V_{gd}^{n+1}) \quad (1.18)$$

avec,

$$f(V_{gd}^{n+1}) = I_{gd}^{n+1} \cdot (h / C_{gd})$$

d'où

$$V_{gd}^{n+1} = V_{gd}^n + (h / C_{gd}) \cdot I_{gd}^{n+1} \quad (1.19)$$

à partir de cette équation on donne le courant :

$$I_{gd} = (C_{gd}/h) \cdot V_{gd}^{n+1} - (C_{gd}/h) \cdot V_{gd}^n \quad (1.20)$$

on peut mettre cette équation sous la forme suivante :

$$I_{gd} - (C_{gd}/h) \cdot V_{gd}^{n+1} + (C_{gd}/h) \cdot V_{gd}^n = 0 \quad (1.21)$$

cette équation $\sum I_{ij}=0$, (loi de KIRCHHOFF) détermine la représentation électrique associée au modèle de la capacité C_{gd} (voir Fig.1.6.a). En utilisant la même procédure, nous déterminons le modèle de la capacité C_{gs} (voir Fig. 1.6.b).

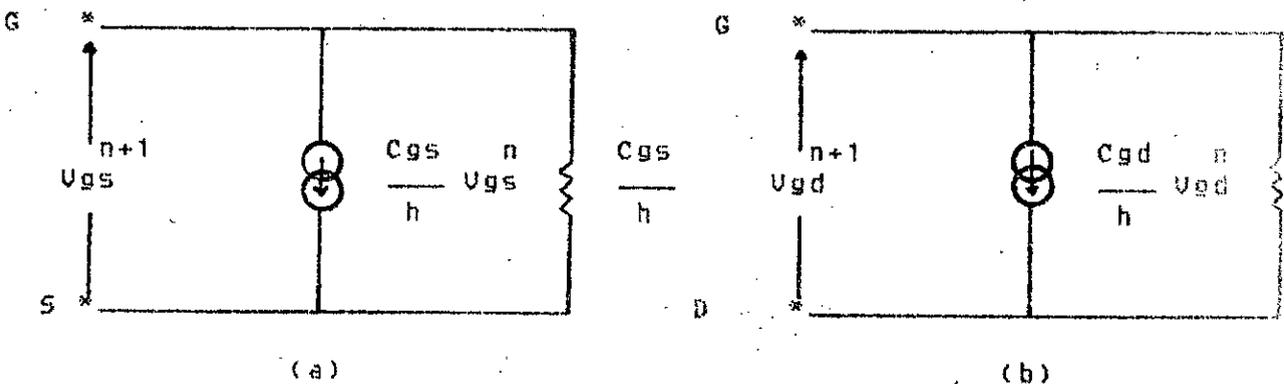


Fig. 1.6 Modèle des capacités C_{gd} et C_{gs}

Les capacités entre électrodes sont considérées comme externes au transistor MOS (Fig. 1.7.a) de façon à conserver le modèle déjà établi au paragraphe 1.4.

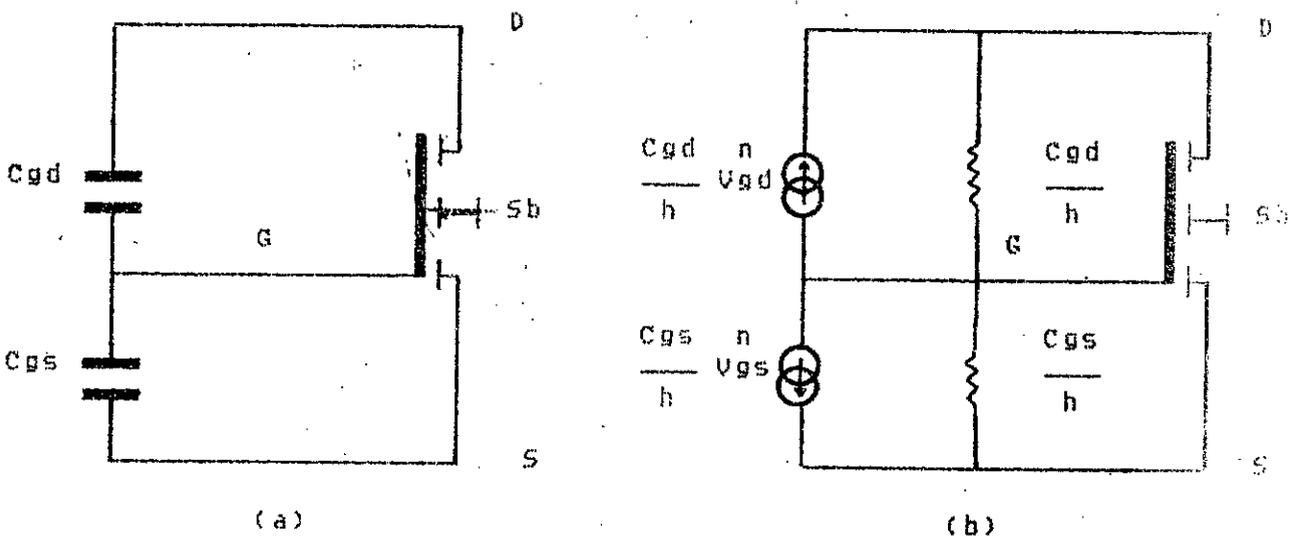


Fig. 1.7 Transistor avec capacités entre électrodes.

A partir des modèles de C_{gd} , C_{gs} et du modèle numérique linéaire du transistor MOS, nous réalisons un modèle numérique linéaire du transistor MOS avec effets de capacités entre électrodes.

Algorithme du modèle :

$$I_d^{m+1,n+1} = I_d^{m,n+1} + G_g^{m,n+1} (V_{gd}^{m+1,n+1} - V_{gd}^{m,n+1}) + G_s^{m,n+1} (V_{sd}^{m+1,n+1} - V_{sd}^{m,n+1}) + (G_s^{m,n+1} + G_d^{m,n+1} + G_g^{m,n+1}) (V_d^{m+1,n+1} - V_d^{m,n+1})$$

$$I_s^{m+1,n+1} = I_s^{m,n+1} + G_g^{m,n+1} (V_{gs}^{m+1,n+1} - V_{gs}^{m,n+1}) + G_s^{m,n+1} (V_{sd}^{m+1,n+1} - V_{sd}^{m,n+1}) + G_g^{m,n+1} (V_s^{m+1,n+1} - V_s^{m,n+1}) + (G_d^{m,n+1} + G_s^{m,n+1}) (V_d^{m+1,n+1} - V_d^{m,n+1})$$

$$I_g^{m+1,n+1} = 0$$

$$I_{gd}^{m+1,n+1} = (C_{gd}/h) \cdot V_{gd}^{m+1,n+1} - (C_{gd}/h) \cdot V_{gd}^{m,n}$$

$$I_{gs}^{m+1,n+1} = (C_{gs}/h) \cdot V_{gs}^{m+1,n+1} - (C_{gs}/h) \cdot V_{gs}^{m,n} \quad \text{avec} \quad I_d^{m+1,n} = I_d^{0,n+1}$$

Ainsi nous obtenons un circuit électrique (Fig.1.8) composé de 16 éléments comme celui de la Fig. 1.5 sauf que les termes des sources de courants et ceux des conductances pour les liaisons G-S et G-D sont modifiés, le reste des éléments est inchangé. Ces modifications sont:

- liaison G-S :

- la conductance est : $G_{gs}^{m,n+1} = G_g^{m,n+1} + C_{gs}/h$ au lieu de $G_{gs} = G_g$

- la source de courant est : $(G_g + C_{gs}/h) \cdot V_{gs}^{m,n+1}$ au lieu de $G_g \cdot V_{gs}$

- liaison G-D :

- la conductance est : $G_{gd}^{m,n+1} = G_g^{m,n+1} + C_{gs}/h$ au lieu de $G_{gd} = G_g$

- la source de courant est $(G_g + C_{gd}/h) \cdot V_{gd}^{m,n+1}$ au lieu de $G_g \cdot V_{gd}$

Le reste des liaisons sont inchangées.

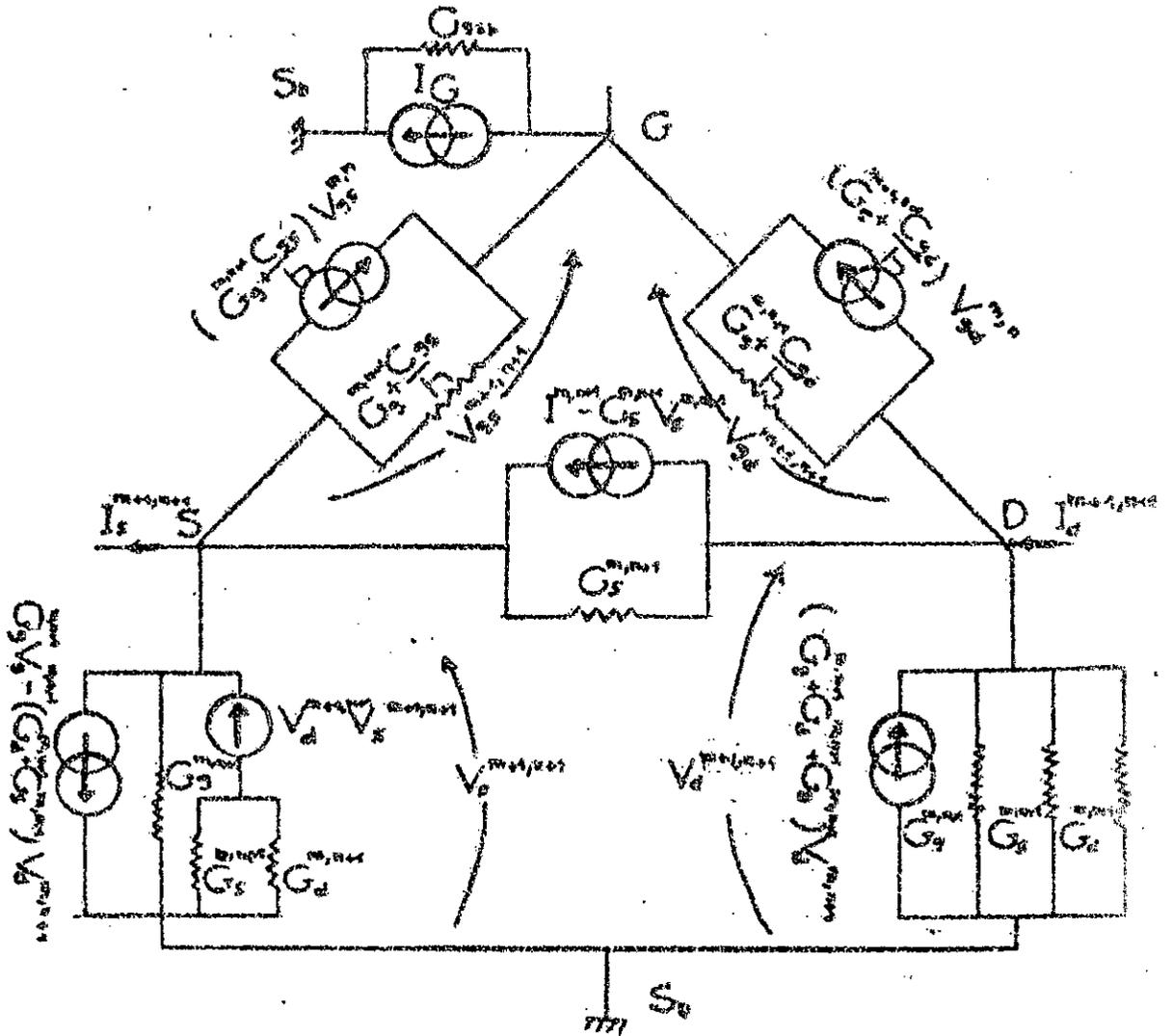


Fig. 1.8: Représentation électrique du modèle numérique associé au transistor MOS tenant compte des capacités C_{gs} et C_{gd} (deuxième approche)

1.2/ Modèle numérique-linéaire du transistor MOS avec les capacités entre électrodes C_{gs} et C_{gd} non linéaires

En réalité les capacités C_{gs} et C_{gd} sont non linéaires et leurs modèles changent avec la structure du transistor MOS. Nous considérons les équations non linéaires suivantes :

$$C_{gs} = g(V_s, V_d, V_g)$$

$$C_{gd} = h(V_s, V_d, V_g)$$

Pour simuler les capacités nous donnons les équations différentielles suivantes :

$$dC_{gs} = (\partial g / \partial V_s) dV_s + (\partial g / \partial V_d) dV_d + (\partial g / \partial V_g) dV_g \quad (1.1)$$

$$dC_{gd} = (\partial h / \partial V_s) dV_s + (\partial h / \partial V_d) dV_d + (\partial h / \partial V_g) dV_g \quad (1.2)$$

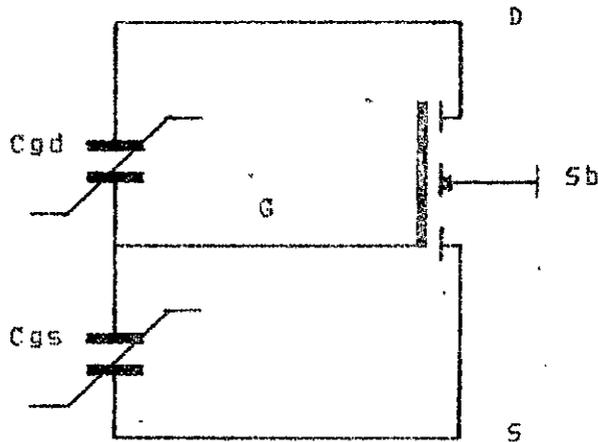


Fig. 1.2 Représentation du transistor MOS avec les capacités Cgs et Cgd non linéaires

En appliquant l'approche de Newton, nous obtenons les équations numériques :

$$C_{gs}^{p+1} = C_{gs}^p + (\partial g / \partial V_s) \begin{vmatrix} p+1 & p \\ (V_s - V_s) + (\partial g / \partial V_d) \\ P & P & P \\ V_d, V_g, V_s \end{vmatrix} + (\partial g / \partial V_g) \begin{vmatrix} p+1 & p \\ (V_d - V_d) + (\partial g / \partial V_g) \\ P & P & P \\ V_d, V_g, V_s \end{vmatrix} + (\partial g / \partial V_g) \begin{vmatrix} p+1 & p \\ (V_g - V_g) \\ P & P & P \\ V_d, V_g, V_s \end{vmatrix} \quad (1)$$

$$C_{gd}^{p+1} = C_{gd}^p + (\partial h / \partial V_s) \begin{vmatrix} p+1 & p \\ (V_s - V_s) + (\partial h / \partial V_d) \\ P & P & P \\ V_d, V_g, V_s \end{vmatrix} + (\partial h / \partial V_d) \begin{vmatrix} p+1 & p \\ (V_d - V_d) + (\partial h / \partial V_g) \\ P & P & P \\ V_d, V_g, V_s \end{vmatrix} + (\partial h / \partial V_g) \begin{vmatrix} p+1 & p \\ (V_g - V_g) \\ P & P & P \\ V_d, V_g, V_s \end{vmatrix}$$

pour $p=0, 1, 2, \dots, p_f$

Nous arrêtons le calcul si nous avons : $\left| C^{p_f+1} - C^{p_f} \right| < \epsilon$

Les capacités C_{gs} et C_{gd} sont des capacités linéaires auxquelles nous appliquons les algorithmes d'intégration numériques, pour Backward-Euler :

$$I_{gs}^{n+1} = \frac{C_{gs}^{p_f+1}}{h} (V_{gs}^{n+1} - V_{gs}^n) \quad (1.26)$$

$$I_{gd}^{n+1} = \frac{C_{gd}^{p_f+1}}{h} (V_{gd}^{n+1} - V_{gd}^n) \quad (1.27)$$

Donc une fois les capacités linéarisées, nous appliquons le principe du paragraphe U.1. Ce qui conduit à la même représentation électrique du modèle du transistor MOS modifié (3^{ème} approche), avec les capacités $C_{gs}^{p_f+1}$ et $C_{gd}^{p_f+1}$ linéarisées

Ce dernier modèle n'est pas utilisable car il nécessite un temps de calcul plus important.

VI/ Vérification expérimentale du modèle numérique-linéaire

Nous considérons le modèle physique de Sah-Pao [14], pour le régime non bloqué $V_e > V_s$:

- Non saturation $V_e > V_d$: $I_d = K_e (2 V_e - V_s - V_d) (V_d - V_s)$
- Saturation $V_e < V_s$: $I_d = K_e (V_e - V_s)^2$

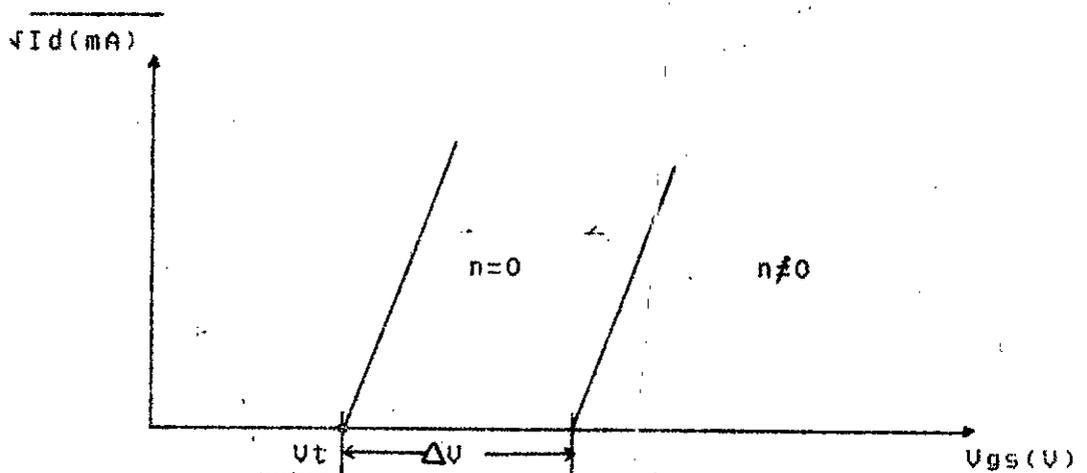
Avec $V_e = (V_g - V_t) / (1 + \eta)$ et $K_e = K (1 + \eta)$ où $K = W/L'$

D'après ces deux équations les paramètres à déterminer sont : V_t , K_e et η .

VI.1/ Méthode de détermination des paramètres en régime de saturation

$$\sqrt{I_d} = \sqrt{K} (V_{gs} - V_{gd}) \quad \text{si } \eta = 0 \quad (1.28)$$

$$\sqrt{I_d} = \sqrt{K_e} (V_{gs} - V_t - \eta V_s) (1 + \eta) \quad \text{si } \eta \neq 0 \quad (1.29)$$



Courbe 1.1 : Représentation de $\sqrt{I_d} = f(V_{gs})$ pour $\eta = 0$ et $\eta \neq 0$

VI.1.1 / Détermination de V_t

On considère deux points $(V_{g'} , \sqrt{I'})$ et $(V_{g''} , \sqrt{I''})$ en régime de saturation :

$$V_{e'} = \sqrt{I'} / K_e = (V_{g'} - V_t) / (1 + \eta) \quad (1.30)$$

$$V_{e''} = \sqrt{I''} / K_e = (V_{g''} - V_t) / (1 + \eta) \quad (1.31)$$

Les équations (1.30) et (1.31) on a :
$$V_t = \frac{U_{g''} - U_{g'} \sqrt{I''/I'}}{1 - \sqrt{I''/I'}}$$

VI.1.2/ Détermination de η

Pour $V_{s_sb} = 0$ on a $\Delta V = U_{g''} - U_{g'}$ à $I=0$ (1.32)

A l'aide des équations (1.28), (1.29) et (1.32) on a :

$$\eta = \frac{\Delta V}{V_{s_sb}}$$

VI.1.3/ Détermination de la conductivité spécifique effective K_e
 $\text{tg}(\alpha)$ = pente de la droite, représente la conductivité spécifique K , alors on a : $K_e = (1 + \eta) * \text{tg}^2 \alpha$

VI.2 / Application au transistor MOS 2N4351

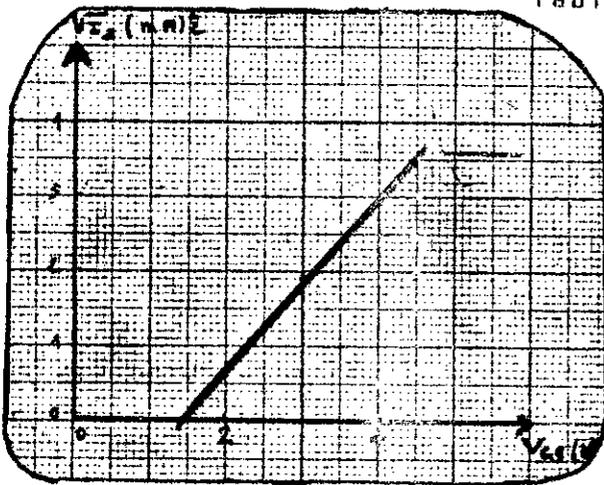
Nous avons deux tableaux de mesures :

$V_{GS}(V)$	4	6	8
$I(mA)$	0.4	3.8	8.7
$\sqrt{I(mA)}$	0.63	1.95	2.25

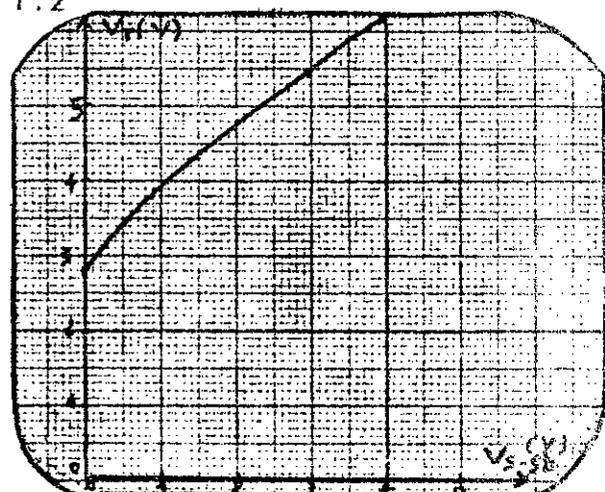
Tableau 1.1

$V_t (V)$	2.8	3.9	4.8	5.4	6.2
$V_{s_sb} (V)$	0	1	2	3	4

Tableau 1.2



Courbe 1.2 :
représentation de $\sqrt{I} = f(V_{GS})$



Courbe 1.3 :
représentation de $V_t = h(V_{s_sb})$

Ainsi nous avons déterminé : $V_t \approx 3V$; $\eta \approx 0.85$; $K_e \approx 1342 \mu A/V$

VI.3/ Evaluation de la précision

Nous déterminerons analytiquement, par simulation numérique et expérimentalement les temps de commutation d'un simple montage

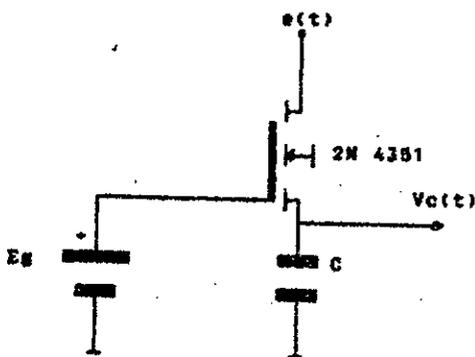


Fig.1.1.10a : Montage SWIVERT

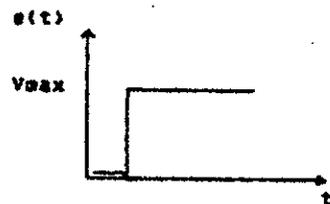


Fig.1.1.10b : Signal d'entrée

a/ Méthode analytique

La charge du condensateur C :

$$I = C \, dV/dt \quad \text{et} \quad dt = (C / (K_e(V_e - V_s))) \, dV \quad \text{à } t=0 \quad V_c = 0 \, V$$

$$t_m = t_{90\%} - t_{10\%}$$

$$t_m = (90 \, C) / (9 \, K_e \, V_e)$$

La décharge du condensateur C :

$$t = C \, \ln((2V_e - V) / V) / (2K_e \, V_e)$$

$$t_d = t_{10\%} - t_{90\%}$$

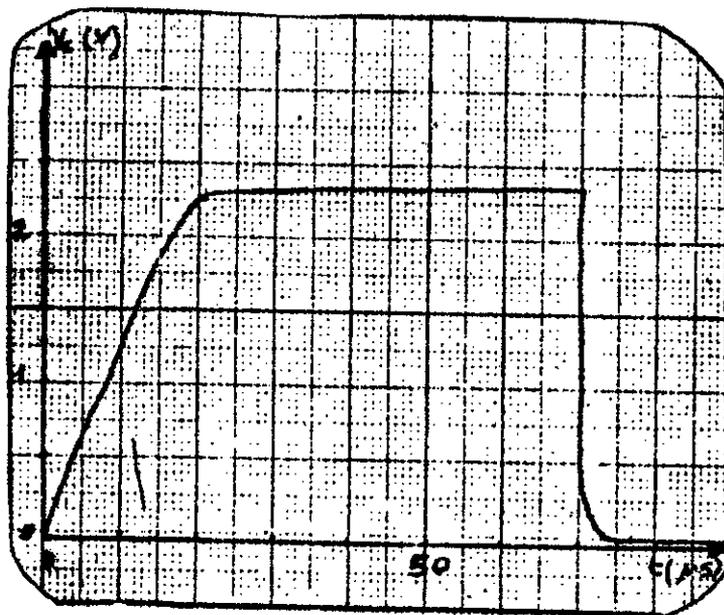
$$t_d = 2.74 \, C / (2K_e \, V_e)$$

Nous regroupons les résultats dans le tableau suivant :

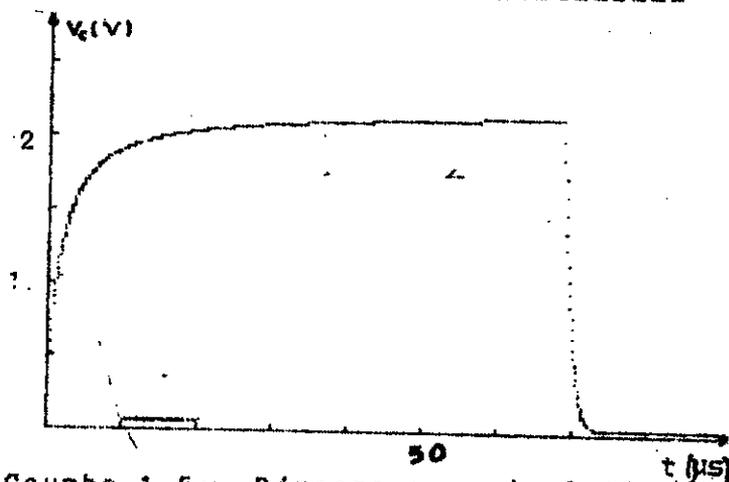
Temps de commutation	Analytique	Méthode Numérique	Expérimentale
t_m (μs)	14.13	14.30	12
t_d (μs)	2.18	2.19	2.6

Tableau 1.3

La réponse expérimentale et celle par simulation (méthode numérique) du montage suiveur sont données par les courbes 1.4 et 1.5 , ci-dessous.



Courbe 1.4 : Réponse expérimentale



Courbe 1.5 : Réponse par simulation

Ces résultats permettent de conclure que l'erreur relative maximale du modèle numérique-linéaire de première étape est de 30% en utilisant le modèle de Sah-Pao qui est de 25% près /14/ ère. Il faut souligner que le modèle est une approximation de 1^{ère} ordre d'une part, et l'analyse expérimentale tient compte de

tous les phénomènes physiques, d'autre part.

VII/Conclusion

Pour la conception, l'efficacité et l'économie d'un programme sont assurées par la simplicité et la précision du modèle.

Un modèle simple, précis est suffisant pour l'algorithme d'analyse des C.I. à MOS. Il reste valable pour n'importe quel modèle physique ou mathématique. Pour cela nous proposons un certain nombre de modèles physiques simple et peu précis, afin que le programme à réaliser soit universel d'une part et utilisable pour la 1^{ère} étape d'analyse des CI à MOS, d'autre part. Pour que le programme puisse utiliser les modèles physiques proposés, nous avons élaboré un modèle numérique-

linéaire du transistor MOS :

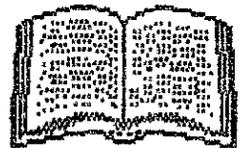
- sans capacités Cgs et Cgd
- avec capacités Cgs et Cgd linéaires
- avec capacités Cgs et Cgd non linéaires.

Or que le programme à réaliser est utilisé pour la 1^{ère} étape d'analyse, la précision du modèle numérique n'est pas importante par rapport à la simplicité.

Par conséquent on utilisera pour le programme d'analyse, le modèle numérique-linéaire simple (sans capacités Cgs et Cgd). Le modèle physique envisagé pour les tests du programme est celui de Sah-Pao.

Une fois le modèle numérique a été réalisé et le modèle physique a été choisi, il faut formuler les équations pour l'analyse des CI à MOS. Pour une meilleur élaboration, les méthodes de formulation automatiques ont été appliquées pour les CI à MOS afin de tirer des conclusions sur l'utilisation de ces techniques, ceci constitue l'objet du chapitre suivant.

Chapitre 2



Chapitre II

ANALYSE ET FORMULATION DES EQUATIONS

I.2 Introduction

Dans l'analyse temporelle, nous différencions la technique transitoire, qui a son origine dans l'analyse du circuit et la technique de génération du chronogramme qui est en fait la simulation logique.

On se limitera à l'analyse transitoire qui n'est pas simple, sa procédure générale pour les circuits linéaire et non linéaire est d'évaluer l'état du circuit à un instant donné. Le temps d'évaluation du régime transitoire est proportionnel aux nombres de points à déterminer, en d'autres termes, le temps de calcul est un multiple de la plus grande constante de temps du circuit à analyser (2 à 3* t_{max}).

Vue l'importance et la complexité de cette analyse, pour la mise en forme d'un système d'équations, nous choisissons les considérations suivantes :

- Méthode d'analyse (structure de la topologie)
- Procédure de formulation
- Algorithme d'intégration numérique
- Organisation des équations

Comme la complexité des circuits électroniques augmente, nous devons utiliser des techniques plus puissantes pour analyser ces circuits, spécialement les méthodes qui peuvent être adoptées par l'ordinateur. La méthode par laquelle les équations sont formulées est une importante clé pour l'analyse assistée par ordinateur. Elle affecte le temps de calcul, l'effort de programmation, le stockage nécessaire et la vitesse d'exécution du programme. La Fig. 2.1 illustre les différentes étapes de

formulation des équations pour les circuits linéaires et non linéaires. Leurs réponses sont obtenues à l'aide des algorithmes de résolution. La réalisation d'un algorithme d'analyse transitoire doit avoir une telle structure. Selon le domaine d'analyse les techniques les plus convenables seront choisies. Les applications aux circuits intégrés à MOS des méthodes : technique des variables d'état et théorie des graphes sont présentées. Comme dans ce type de circuit, on a des équations différentielles de 1^{er} ordre, alors une étude des différentes méthodes d'intégration numériques est présentée.

Etage de formulation des équations

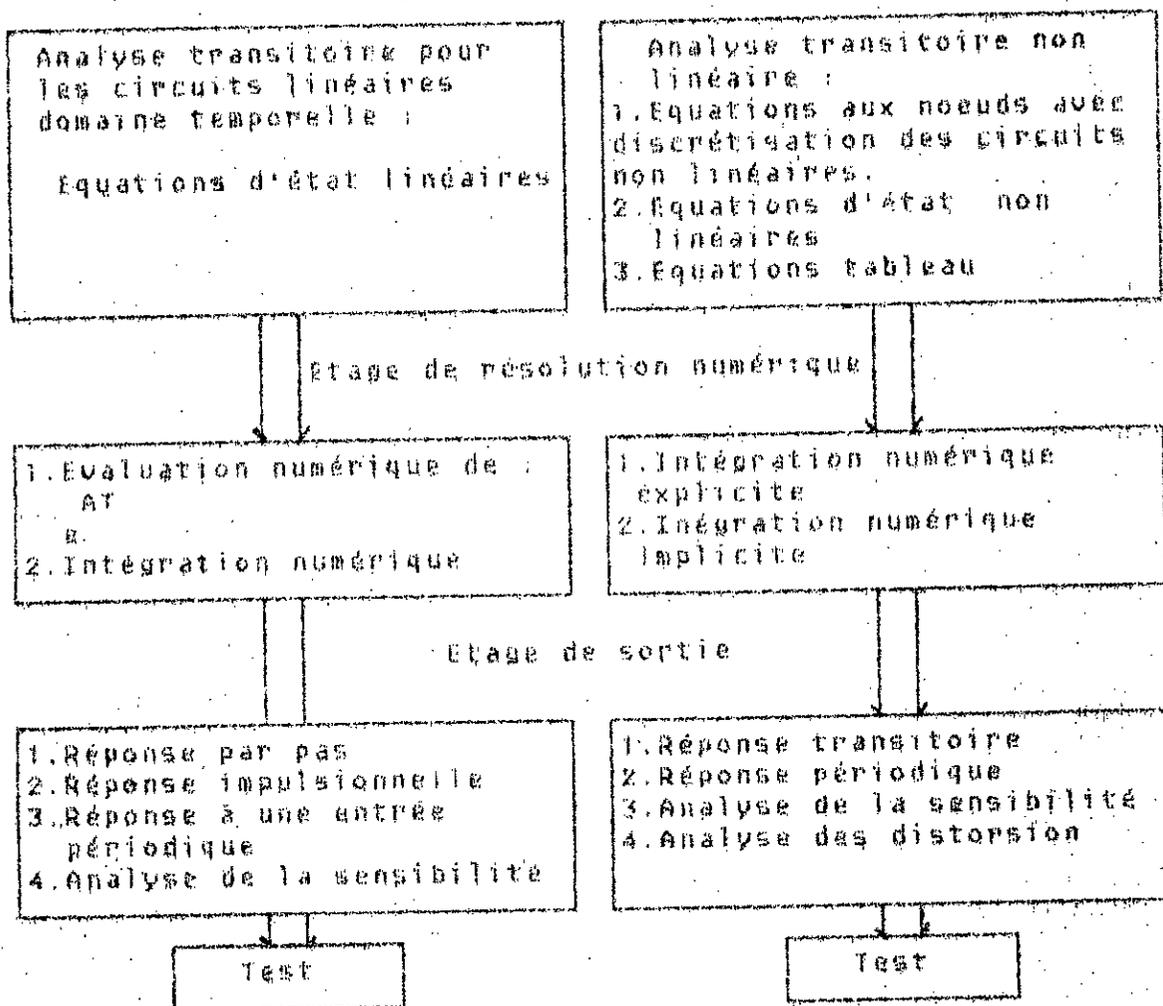


Fig. 2.1 Etages d'analyse des circuits

II.2 Analyse et formulation des équations

Tous les programmes d'analyse commencent à partir du même point: description des éléments du circuit et de leurs interconnexions. Ils sont convertis par le programme en une série d'équations, cette transformation est réalisée par :

- Analyse du circuit (noeud, maille, section, ...).
- Formulation des équations (soit par les variables d'état soit par la théorie des graphes).

II.1.4 Analyse des circuits

Un circuit électrique peut être analysé par :

- La loi des noeuds $\sum I_j = 0$
- La loi des mailles $\sum U_j = 0$
- D'autres lois faisant la combinaison des deux premières /22/ telle que la loi de section etc...

II.1.1.1 Analyse aux noeuds

L'analyse de tous les noeuds donne une forme compacte :

$$A \cdot I = 0$$

où I est un vecteur courant et A est une matrice de description, ses éléments sont donnés par :

$a_{ij} = 1$ si la branche j est reliée au noeud i et le courant est entrant

$a_{ij} = -1$ si la branche j est reliée au noeud j et le courant est sortant

$a_{ij} = 0$ si la branche j n'est pas reliée au noeud i

II.1.1.2 Analyse aux mailles

En appliquant la loi $\sum v_k = 0$ à toutes les mailles, on obtient la forme matricielle suivante :

$$B \cdot V = 0$$

où B la matrice de $N \times N$, si on dispose de N mailles et V le vecteur tension de dimension $N \times 1$.

Les coefficients de B sont définis de la façon suivante :

$b_{ij} = 1$ si la branche j est dans la boucle i et sa direction est la même que celle de la boucle

$b_{ij} = -1$ si la branche j est dans la boucle i et sa direction est en sens inverse de celle de la boucle

$b_{ij} = 0$ si la branche j n'appartient pas à la boucle i

II.1.1.2 Analyse des sections

L'analyse aux noeuds est basée sur la somme algébrique de tous les courants de n'importe quel noeud est nulle. Une forme plus générale, la somme algébrique des courants à travers une section est nulle. Sa forme matricielle est :

$$D \cdot J = 0$$

où D est de dimension $N \times N$ et J est de dimension $N \times 1$, si N représente le nombre de sections dans le circuit.

Les coefficients de la matrice D sont définis de la façon suivante:

$d_{ij} = 1$ si la branche j est dans la section i et de même direction

$d_{ij} = -1$ si la branche j est dans la section i et de direction opposée

$d_{ij} = 0$ si la branche j n'est pas dans la section i

II.2 Techniques de formulation

II.2.1 Par les variables d'état

II.2.1.1 Théorie

Cette méthode est utilisée essentiellement pour les cas suivants:

- Réponses temporelles quand les signaux d'excitations ont des formes arbitraires.

- Localisation des pôles et des zéros de quelques fonctions de transfert du circuit.

La méthode des variables d'état donne une série d'équations

ET

différentielles de 1^{er} ordre avec un minimum d'inconnues.

II.2.1.1.1 Les circuits linéaires

Pour les circuits linéaires, l'approche de la variable d'état est caractérisée par deux équations de la forme :

$$\dot{x}(t) = A \cdot x(t) + B \cdot u(t) \quad (2.1)$$

$$y(t) = C \cdot x(t) + D \cdot u(t) \quad (2.2)$$

où

$u = m \times 1$ vecteur représentant les m sources d'entrées.

$y = p \times 1$ vecteur représentant les p sorties (tensions et/ou courants)

$x = n \times 1$ vecteur consistant de la série des variables auxiliaires indépendantes

A, B, C et D sont des matrices réelles constantes.

L'équation (2.1) peut être résolue soit analytiquement, soit numériquement.

a) Solution analytique

soit l'équation :

$$\dot{x}(t) = A \cdot x(t) + B \cdot u(t)$$

si $u=0$ la solution est :

$$x(t) = x(0) \cdot \exp(A \cdot t) \quad (2.3)$$

si $u \neq 0$ la solution est :

$$x(t) = f(t) \cdot \exp(A \cdot t) \quad (2.4)$$

où $f(t)$ est à déterminer :

$$\dot{x}(t) = A \cdot f(t) \cdot \exp(A \cdot t) + f(t) \cdot \exp(A \cdot t)$$

$$\dot{x}(t) = A \cdot x + f(t) \cdot \exp(A \cdot t)$$

comme

$$\dot{x}(t) = A \cdot x(t) + B \cdot u(t)$$

on a

$$B \cdot u(t) = \exp(A \cdot t) \cdot f(t)$$

d'où

$$f(t) = \int_0^t \exp(A \cdot \tau) \cdot B \cdot u(\tau) \, d\tau \quad (2.5)$$

b) Solution numérique :

L'équation (2.1) peut être résolue par l'un des algorithmes d'intégration numériques existants :

$$\dot{x}(t) = A x(t) + B u(t) \quad \text{où} \quad \dot{x}(t) = f(x)$$

La solution numérique est élaborée par :

- Développement de Taylor

$$x_{n+1} = x_n + h \cdot f'(x_n, t_n) + (h^2/2!) \cdot f''(x_n, t_n) + \dots + (h^{p-1}/(p-1)!) \cdot f^{(p-1)}(x_n, t_n) + \dots$$

Le développement de Taylor peut être mis sous la forme suivante :

$$x_{n+1} = x_n + h \cdot T_p(x_n, t_n, h) \tag{2.6}$$

où $T_p = f'(x_n, t_n) + (h/2!) \cdot f''(x_n, t_n) + \dots + (h^{p-1}/(p-1)!) \cdot f^{(p-1)}(x_n, t_n)$

Si $p=1$ nous avons : $x_{n+1} = x_n + h \cdot f'(x_n, t_n)$

- Algorithme de Runge-Kutta

Dans le développement de Taylor nous devons calculer des dérivées partielles qui nous sont un inconvénient majeur. La méthode de Runge-Kutta remplace $T_p(x_n, t_n, h)$ par une autre fonction $K(x_n, t_n, h)$, qui ne nécessite pas de calcul de dérivées partielles.

Si $p=2$ nous avons : $x_{n+1} = x_n + h \cdot K(x_n, t_n, h)$

où $K(x_n, t_n, h)$ est exprimé par l'équation suivante :

$$K(x_n, t_n, h) = (1-a) \cdot f(x_n, t_n) + a \cdot f(x_n + (h/2) \cdot a, t_n + (h/2) \cdot a) + a^2 \cdot f(x_n + h \cdot a, t_n + h \cdot a)$$

Les coefficients a_1 et a_2 sont à déterminer.

1.2.1.1.2 Circuits non linéaires

L'analyse des circuits non linéaires, nous conduit à un système d'équations différentielles de 1^{er} ordre sous la forme suivante :

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, x_3, \dots, x_n, t) \\ \dot{x}_2 &= f_2(x_1, x_2, x_3, \dots, x_n, t) \\ &\dots \\ \dot{x}_n &= f_n(x_1, x_2, x_3, \dots, x_n, t) \end{aligned}$$

Notation vectorielle :

$$\dot{x}(t) = f(x, t) \quad (2.7)$$

où $f(x, t)$ est une fonction vectorielle non linéaire, donc il est difficile de déterminer la forme standard de (2.7)

$$\dot{x} = A \cdot x + B \cdot u$$

La non existence des équations d'état pour un circuit non linéaire est souvent le résultat du choix des modèles du circuit qui rapp. le circuit physique. Un algorithme général a été développé pour la formulation des équations d'état d'une grande classe de circuits nonlinéaires sans sources dépendantes. Néanmoins l'équation (2.7) peut être résolue à l'aide des algorithmes numériques.

II.2.1.2 Applications aux C.I à MOS

Les C.I à MOS sont des circuits non linéaires comportant des transistors MOS non linéaires et des capacités linéaires. Pour les systèmes, la solution peut être toujours exprimée analytiquement à l'aide de la matrice de transition e^{At} , mais la solution du circuit non linéaire est obtenue soit par méthode graphique, soit par la méthode numérique. La méthode graphique ne peut être appliquée aux circuits d'ordre supérieur à 2. Par contre, les méthodes numériques d'analyse sont valables pour les circuits non linéaires avec avec n'importe quel degré de complexité.

II.2.1.2.1 Formulation des équations d'état d'un C.I à MOS avec le modèle non linéaire du transistor MOS

Pour formuler les équations d'états, on utilise une configuration connue : inverseur à plusieurs étages.

Nous supposons que le circuit contient N_n nœuds et N_t transistors.

a) Modèle du transistor sans effet de capacités série

Pour élaborer les équations d'état, nous considérons la loi

$$I = I_0$$

pour le circuit intégré donné par la Fig. 2.2

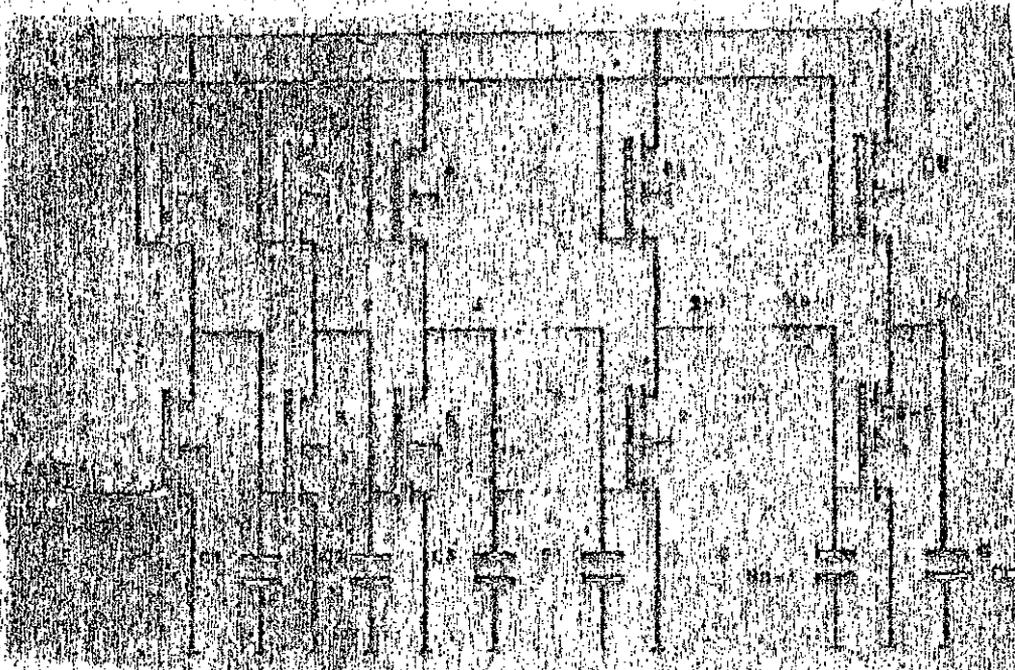


FIG. 2.2. TRANSDUCTION À NO ÉTATS

Au nœud 1
 $I_1 - I_2 + I_3 = 0$
 $f_1(v_1, u_1) - f_2(v_2, u_2) + C_1 \frac{dv_1}{dt} = 0$

Au nœud 2
 $I_3 - I_4 + I_5 = 0$
 $f_3(v_3, u_3) - f_4(v_4, u_4) + C_2 \frac{dv_2}{dt} = 0$

Au nœud i
 $I_p - I_{p+1} + I_{p+2} = 0$
 $f_p(v_p, u_p) - f_{p+1}(v_{p+1}, u_{p+1}) + C_i \frac{dv_i}{dt} = 0$

Au nœud N
 $I_{N-1} - I_N + I_{N+1} = 0$
 $f_{N-1}(v_{N-1}, u_{N-1}) - f_N(v_N, u_N) + C_N \frac{dv_N}{dt} = 0$

Les fonctions non linéaires f_j sont données par l'un des modèles physiques discutés dans le chapitre précédent, pour le modèle de Sah-Pao [14], nous avons :

$$f_j(v) = 0 \quad \text{si} \quad v_{gs} - v_{ds} < v_{th}$$

$$f_j(v) = k_e \left[v_{gs} - v_{th} - v_{ds} \right] \left[v_{ds} - v_{th} \right] \quad \text{si} \quad v_{ds} > v_{th}$$

$$f_j(v) = k_e (v_{gs} - v_{th})^2 \quad \text{si} \quad v_{ds} > v_{gs} - v_{th}$$

D'après l'équation (2.9), nous remarquons que ce procédé de formulation donne une équation qui dépend du modèle du transistor MOS, son algorithme se résume de la façon suivante :

1. donner la matrice de description de la topologie du C.I. : $G(N_n, N_t)$
2. déterminer le régime de chaque transistor et calculer les f_j
3. calculer la matrice C
4. résoudre l'équation (2.9) par l'un des algorithmes d'intégration numérique. Pour le cas de Backward-Euler nous avons :

$$v^{n+1} = v^n + h C^{-1} G F(v^n) \quad (2.10)$$

5. enfin, résolution de l'équation (2.10) par l'un des algorithmes de résolution des équations non linéaires

2. Modèle du transistor MOS avec effet de capacités entre électrodes

En utilisant ce modèle et le C.I de la fig.2.3, la matrice C est plus une matrice diagonale, mais une matrice de la forme :

$$C = [c_{ij}]$$

où les c_{ij} dépendent de la capacité parasites et des capacités entre électrodes :

$$c_{ij} = C + \sum_k (C_{gd} + C_{gs}) \quad \text{si } i=j$$

$$c_{ij} = - \sum_k (C_{gd} + C_{gs}) \quad \text{si } i \neq j$$

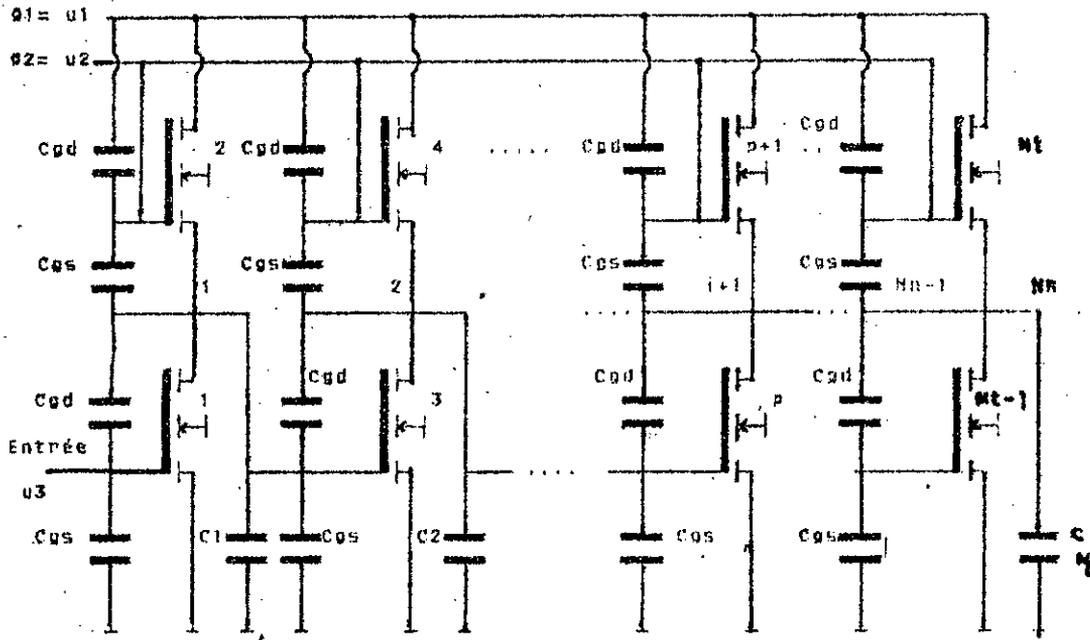


Fig. 2.3 Inverseur à N_n étages avec effet de capacités

C_i représente la capacité parasite au noeud considéré (i).

Les réponses du C.I sont données par la résolution de l'équation (2.9)

II.2.1.2.2 Formulation des équations d'états des CI à MOS pour le modèle du transistor numérique-linéaire

Le modèle numérique-linéaire du transistor MOS présenté dans le chapitre I est :

$$I_d = I_{d0} + G_g (V_g - V_{g0}) + G_d (V_d - V_{d0}) + G_s (V_s - V_{s0})$$

Nous remplaçons $F(v)$ par I_d dans l'équation (2.9), nous avons :

$$v = C_i G^{-1} I_d$$

$$v = C_i G (I_d + G_g (V_g - V_{g0}) + G_d (V_d - V_{d0}) + G_s (V_s - V_{s0}))$$

I_d : vecteur courant à la m itération

V_g : vecteur tension gate à la $(m+1)$ itération (de même pour V_d et V_s)

$$\begin{bmatrix} m+1 \\ Id1 \\ m+1 \\ Id2 \\ \vdots \\ m+1 \\ IdNt \end{bmatrix} = \begin{bmatrix} m \\ Id1 \\ m \\ Id2 \\ \vdots \\ m \\ IdNt \end{bmatrix} + \begin{matrix} m & m \\ +[Gg1, Gg2, \dots] \end{matrix} \begin{bmatrix} m+1 & m \\ Ug1 & -Ug1 \\ m+1 & m \\ Ug2 & -Ug2 \\ \vdots & \vdots \\ m+1 & m \\ UgNt & -UgNt \end{bmatrix} + \begin{matrix} m & m & m \\ +[Gd1, Gd2, \dots, GdNt] \end{matrix} \begin{bmatrix} m+1 & m \\ Ud1 & -Ud1 \\ m+1 & m \\ Ud2 & -Ud2 \\ \vdots & \vdots \\ m+1 & m \\ UdNt & -UdNt \end{bmatrix} \\
 + \begin{matrix} m & m \\ +[Gs1, Gs2, \dots] \end{matrix} \begin{bmatrix} m+1 & m \\ Vs1 & -Vs1 \\ m+1 & m \\ Vs2 & -Vs2 \\ \vdots & \vdots \\ m+1 & m \\ VsNt & -VsNt \end{bmatrix}$$

sous la forme compacte :

$$\dot{V} = C \cdot G \cdot [Gg \cdot Ug + Gd \cdot Ud + Gs \cdot Vs] + G \cdot [Id - Gg \cdot Ug - Gd \cdot Ud - Gs \cdot Vs]$$

Le premier terme peut se mettre sous la forme : $A \cdot \dot{V}$, mais le 2^{ème} terme représente les sources de courant dependantes. La présence du 2^{ème} terme engendre une forme : $\dot{V} = A \cdot \dot{V} + G \cdot V$, c'est une forme non standard, alors nous ne pouvons pas utiliser les techniques de la variable d'état. Pour surmonter ce problème, une étude du processus de réduction a été utilisée afin d'obtenir les équations d'état, mais ce processus exige une condition nécessaire et suffisante pour un tel circuit à posséder une solution. Ces conditions sont liées à la topologie du circuit et aux valeurs de ses éléments /35/-/36/ :

- restriction topologique (fixer l'ordre maximum de complexité du circuit)
- processus d'élimination
- choix des variables d'état

Les problèmes des variables d'état peuvent être surmontés, mais le processus devient de plus en plus complexe et ne garantie pas la solution.

II.2.2 Formulation des équations par la théorie des graphes

II.2.2.1 Théorie

Les circuits composés obéissent aux 3 lois de bases KVL, KCL et les lois des éléments (caractéristiques). Les 2 premières lois sont des contraintes algébriques sur la tension et le courant de branche. La topologie du circuit, les propriétés du circuit et les lois de bases conduisent à une technique mathématique puissante pour l'analyse du circuit : la théorie des graphes. Les circuits non linéaires peuvent être modélisés comme une interconnection des éléments à 2 bornes avec des caractéristiques spécifiées.

Une description complète du modèle doit contenir les informations suivantes :

- 1) Comment les branches sont connectées
- 2) Les directions de référence pour les courants et les tensions de branches
- 3) Les caractéristiques des branches

Pour tracer le graphe associé (G_a) au circuit donné, on doit satisfaire (1) et (2), en respectant la règle suivante : remplacer chaque élément à 2 bornes par un segment appelé branche ayant la même direction (voir Fig. 2.4)

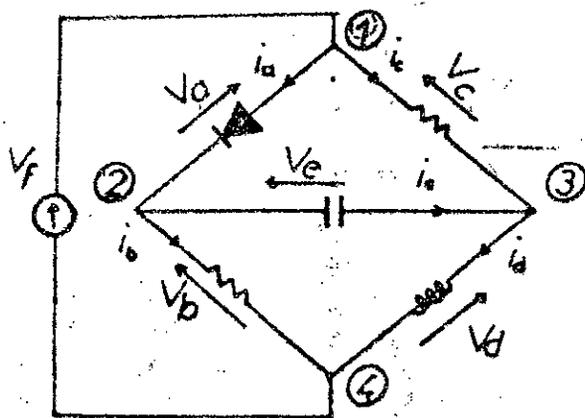


Fig.2.4.a : Circuit

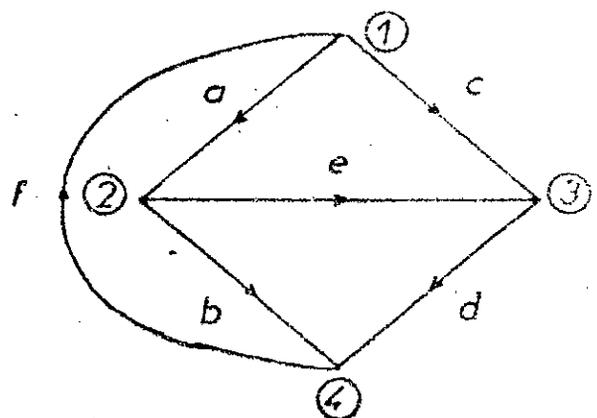


Fig.2.4.b : Graphe associé G_a

Le graphe orienté qui décrit complètement les connexions et les directions de référence des branches du circuit, n'a pas une forme souhaitée pour le stockage dans un ordinateur. Un autre procédé doit être élaboré, l'information contenue dans le graphe Ga peut être complètement stockée dans une matrice incidente noeuds-branches. Pour un graphe orienté de N noeuds et b branches, nous définissons la matrice incidente A de dimension Nxb : $A = [a_{ij}]$

$a_{ij} = 1$ si la branche j est reliée au noeud i et la fleche est orientée vers le noeud.

$a_{ij} = -1$ si la branche j est reliée au noeud i et la fleche est orientée en sens inverse.

$a_{ij} = 0$ si la branche j n'est pas reliée au noeud i.

Le graphe orientée de la Fig. 2.4 donne une matrice de la forme :

		branches					
		noeuds					
		a	b	c	d	e	f
$A =$	1	1	0	1	0	0	-1
	2	-1	1	0	0	1	0
	3	0	0	-1	1	-1	0
	4	0	-1	0	-1	0	1

Les courants de branches sont représentés par un vecteur colonne $i(t)$ de dimensions bxi. La loi KCL appliquée à tous les noeuds donne la forme matricielle suivante: $A \cdot i = 0$

II.2.2.1.1 Formulation des équations pour les circuits résistifs linéaires

Nous considérons trois vecteurs de tensions : v_j représente la d.d.p aux bornes de l'élément, E_j la tension d'alimentation et V_j est la d.d.p de la branche considérée.

$$V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_b \end{bmatrix} \quad v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_b \end{bmatrix} \quad E = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_b \end{bmatrix}$$

nous posons : $V = v - E$ et $I = i - J$

E et J représentent vecteur tension d'alimentation et vecteur courant d'alimentation. La loi KCL appliquée à tous les noeuds nous donne :

$$A \cdot I = 0$$

$$A \cdot (i - J) = 0 \Rightarrow A \cdot i = A \cdot J$$

Chaque branche b est caractérisée par :

- si b est une résistance linéaire nous avons : $i = (1/R) \cdot v$
- si b est une source de courant dépendante nous avons : $i = G \cdot v$

Les lois de Kirchhoff nous conduisent à :

$$\begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_b \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1b} \\ y_{21} & y_{22} & \dots & y_{2b} \\ \vdots & \vdots & \dots & \vdots \\ y_{b1} & y_{b2} & \dots & y_{bb} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_b \end{bmatrix}$$

où

$$y_{ka} = \begin{cases} 0 & \text{si } a = k \\ 1/R_k & \text{si } a \neq k \end{cases} \quad \begin{matrix} \text{branche } k \text{ linéaire} \\ \text{résistive} \end{matrix}$$

$$y_{ka} = \begin{cases} 0 & \text{si } a = j \\ G & \text{si } a \neq j \end{cases} \quad \begin{matrix} \text{source de tension} \\ \text{contrôlée dépendante} \\ \text{de la tension } v_j \end{matrix}$$

alors nous avons :

$$i = Y_b \cdot v$$

Y_b : matrice admittance branche de dimensions $b \times b$

$$A \cdot i = A \cdot J$$

A : matrice de description topologique $n \times b$

nous remplaçons i par $Y_b \cdot v$ dans la 2^{ème} équation :

$$A \cdot Y_b \cdot v = A \cdot J \quad \text{avec } i = Y_b \cdot v$$

où $V = U + E$

$$A.Yb.(U + E) = A.J$$

$$A.Yb.V + A.Yb.E = A.J$$

$$A.Yb.V = A.[J - Yb.E]$$

comme

$$V = A^t . v_n \quad \text{où} \quad v_n : \text{tension noeud-masse}$$

$$A.Yb.A^t . v_n = A.[J - Yb.E]$$

$$(A.Yb.A^t) . v_n = A.[J - Yb.E]$$

$$Y_n . v_n = J_n$$

où

$$Y_n = A.Yb.A^t$$

Y_n : matrice admittance aux noeuds

et

$$J_n = A.[J - Yb.E]$$

J_n : vecteur source de courant aux noeuds

d'où v_n est donné par :

$$v_n = Y_n^{-1} . J_n$$

Une fois v_n est calculé, on calcul $V = A^t . v_n$ et $v = U + E$

La théorie des graphes permet de formuler un système d'équations linéaires ($Y_n . v_n = J_n$).

L'algorithme de formulation des équations linéaires se résume de la façon suivante :

1. Déterminer le graphe G_a
2. déterminer les matrices A et A^t
3. déterminer Y_b
4. calculer $Y_n = A.Yb.A^t$
5. calculer $J_n = A.[J - Yb.E]$
6. résolution du système $Y_n . v_n = J_n$

Nombre d'opérations :
 Multiplications :

$$b = 3.N^2.b + b$$

N représente le nombre de noeuds dans le circuit électrique et
 b le nombre de ses branches .

2.2.1.2 Formulation des équations pour les circuits non linéaires

1. Formulation des équations non linéaires à l'aide des formules topologiques
2. Résolution de ces équations par des techniques numériques appropriées.

Comme dans le cas linéaire, on considère la branche composée et son graphe (Fig.2.5).

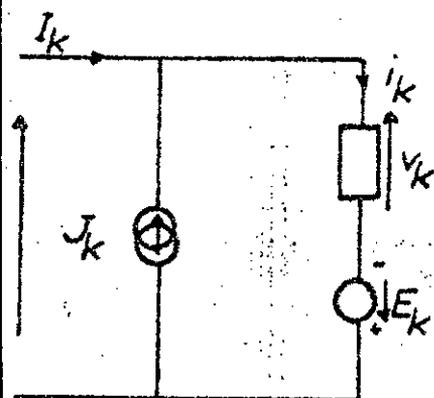


Fig.2.5.a : Circuit

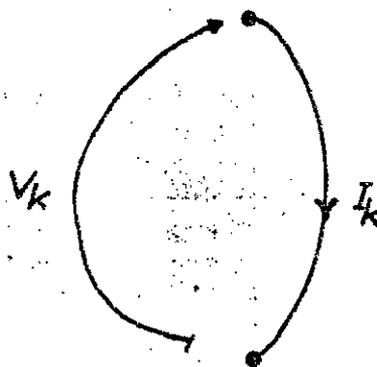


Fig.2.5.b : Sa représentation

avec : $V_k = v_k - E_k$ et $I_k = i_k - J_k$
 la source de courant est caractérisée par :

$$i_k = G_k \cdot v_k \quad (2.11)$$

un élément non linéaire est caractérisé par :

$$i_k = g_k(v_k) \quad (2.12)$$

Les équations (2.11) et (2.12) peuvent être combinées en une forme plus compacte :

$$\begin{bmatrix} i_1 \\ i_1 \\ \vdots \\ i_b \end{bmatrix} = \begin{bmatrix} g_1 (v_a) \\ g_2 (v_b) \\ \vdots \\ g_b (v_f) \end{bmatrix}$$

où v_a, v_b, \dots, v_f représentent les tensions v_1, v_2, \dots, v_b .
 Nous utilisons les mêmes équations que dans le cas linéaire :

$$\begin{aligned} A \cdot i &= A \cdot J \\ A \cdot g(v) &= A \cdot J \end{aligned}$$

comme $v = U + E$ nous avons :

$$A \cdot g(U + E) = A \cdot J$$

$$U = A \cdot v$$

$$A \cdot g(A \cdot v + E) = A \cdot J \quad (2.13)$$

L'équation (2.13) représente un système d'équations aux noeuds non linéaires qui peut être écrite sous la forme :

$$f(v) = 0 \quad (2.14)$$

avec $f(v) = A \cdot g(A \cdot v + E) - A \cdot J$

La résolution de l'équation (2.14) se fait à l'aide des algorithmes numériques .

L'algorithme point fixe : résolution de l'équation de la forme $F(v) = v$. Pour ce procédé il faut donner (2.14) sous forme $F(v) = v$, cette transformation n'est pas toujours évidente. Pour cela on utilise un autre algorithme de Newton-Raphson :

$$v^{m+1} = v^m - [A \cdot (\partial g(A \cdot v^m + E) / \partial v)]^{-1} \cdot [A \cdot g(A \cdot v^m + E) - A \cdot J]$$

II.2.2.2 Formulation des équations par la théorie des graphes pour l'analyse du C.I à MOS

Le modèle du transistor MOS est un modèle numérique-linéaire, c'est un ensemble d'éléments : conductances et sources de courants dépendantes. Pour les différentes configurations (modèles avec et sans effets de capacités) on peut donner le

graphe associé afin de déterminer la matrice incidente A. Pour le modèle de la Fig. 1.5 nous avons le graphe suivant :

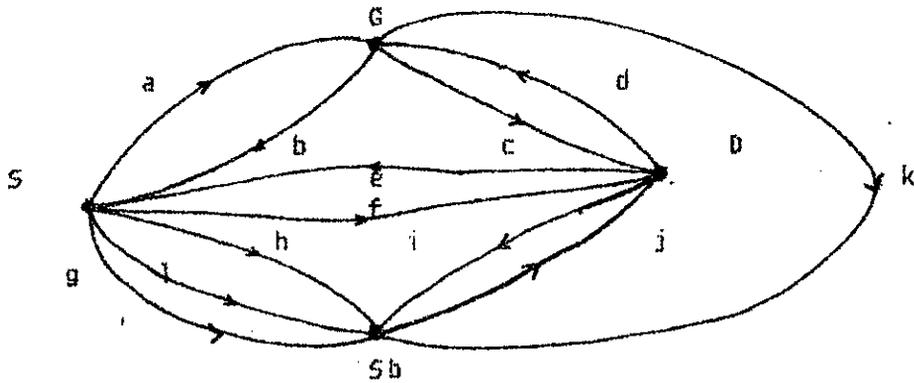


Fig. 2.6 Graphe associé au modèle numérique du transistor

La matrice A du graphe est :

$$A = \begin{matrix} & \begin{matrix} a & b & c & d & e & f & g & h & i & j & k & l \end{matrix} \\ \begin{matrix} S \\ D \\ G \\ Sb \end{matrix} & \begin{bmatrix} -1 & 1 & 0 & 0 & 1 & -1 & -1 & -1 & 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & -1 & 1 & 1 \end{bmatrix} \end{matrix}$$

si nous prenons le cas du circuit intégré à MOS suivant :

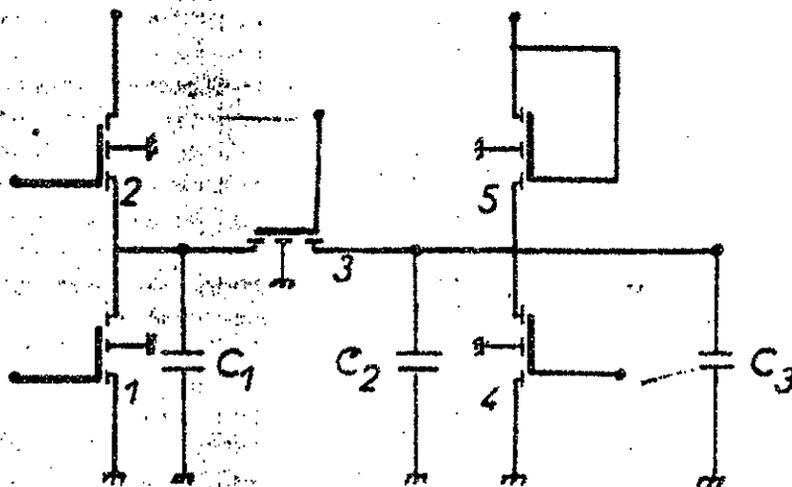


Fig. 2.7 Circuit intégré à MOS quelconque

nous aurons le graphe associé suivante :

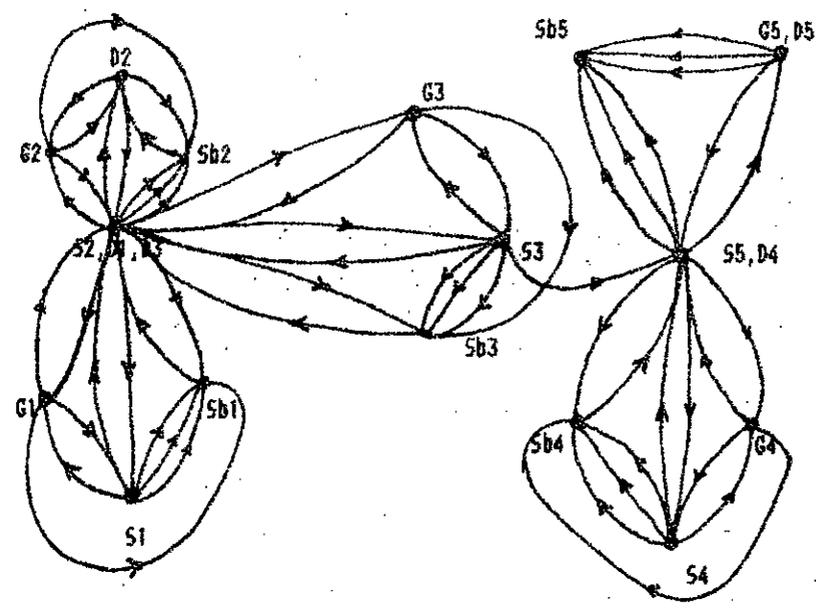


Fig. 2.8 Graphe associé au circuit intégré à MOS quelconque

Plus le nombre de transistors augmente, plus le graphe associé au CI devient complexe et illisible, par conséquent il est difficile de déterminer la matrice A. Cette limitation peut être surmontée, en élaborant un algorithme qui utilise le graphe associé du transistor MOS et la topologie du CI à analyser pour donner la matrice A (Fig. 2.9)

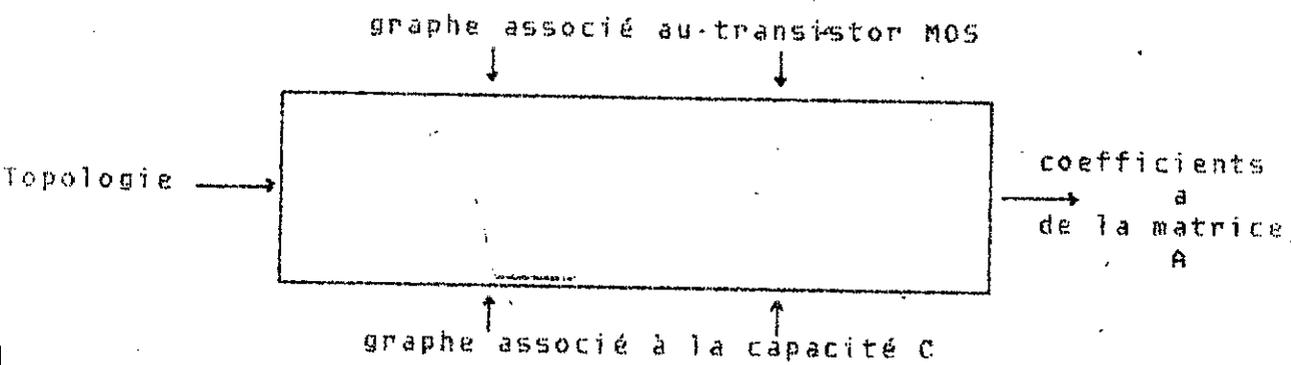


Fig. 2.9 Conversion Topologie/Matrice A

Cependant, la mise en oeuvre d'un programme universel d'analyse des CI à MOS nécessite un programme de conversion Topologie/Graphe (ou matrice A).

Pour un CI de N_n noeuds et N_t transistors, comme le modèle du transistor possède 12 branches alors le CI aura au total :

$(12.N_t + N_n)$ branches et $(3.N_n + 2)$ noeuds. Le nombre d'opérations nécessaire est :

Nombre de multiplications

$$N_m = 3.(3.N_n+2)^2.(12.N_t+N_n)+(12.N_t+N_n)$$

Nombre d'additions

$$N_a = 3.(3.N_n + 2)^2.(12.N_t + N_n - 1) + (12.N_t + N_n - 1)$$

Pour N_n et N_t grands nous avons : $N_m = 632.N_n^2.N_t + 36.N_n$ (2.15)

D'après l'équation (2.15), N_m est proportionnel à N_n^3 et le temps de calcul augmente avec N_n .

L'espace mémoire nécessaire

$E_m = (2.N.b + 2.Nb^2 + 3.Nb)$ espace mémoire d'une variable

avec $N = 3.N_n + 2$ et $b = 12.N_t + N_n$

III Les algorithmes d'intégration numérique

Afin de résoudre le problème de la valeur initiale, l'algorithme se met sous la forme :

$$x^{n+1} = \sum_{i=0}^p a_i x^{n-i} + h \sum_{i=-1}^p b_i f(x^{n-i}, t) \quad (2.16)$$

Les $(2p+3)$ coefficients $(a_0, a_1, \dots, a_p, b_{-1}, b_0, \dots, b_p)$ sont choisis telle que la solution $x(t)$ est un polynôme de degré k , alors l'équation (2.16) donne une solution exacte : $x^{n+1} = x^{n+1}(t)$

Selon le choix des coefficients a_i et b_i on a différents types d'algorithmes :

III.1 Algorithme d'ADAMS-BASHFORTH

L'algorithme d'ADAMS-BASHFORTH du k ^{ème} ordre est un algorithme explicite obtenu en posant :

$$p=k-1, a_1 = a_2 = \dots = a_{k-1} = 0 \text{ et } b_{-1} = 0$$

En remplaçant dans l'équation (2.16) on obtient :

$$x_0^{n+1} = a_0 x_0^n + h [b_0 f(x_0^n, t_0) + b_1 f(x_0^{n-1}, t_0) + \dots + b_{k-1} f(x_0^{n-k+1}, t_0)]$$

Pour résoudre le problème de la valeur initiale, des contraintes ont été établies. Elles servent à déterminer les coefficients a_i, b_i de l'équation (2.16).

Les 2 contraintes pour les algorithmes multi-pas d'ordre k sont :

$$1) \sum_{i=0}^p a_i = 1 \quad j=1, 2, \dots, k \quad (2.17)$$

$$2) \sum_{i=1}^p (-i)^j a_i + j \sum_{i=-1}^p (-i)^{j-1} b_i = 1 \quad (2.18)$$

en utilisant la contrainte 2, l'équation (2.16) devient :

$$x_0^{n+1} = x_0^n + h \sum_{i=0}^{k-1} \beta_i(k) f(x_0^{n-i}, t_0) \quad (2.19)$$

pour déterminer $\beta_0(k), \beta_1(k), \dots, \beta_{k-1}(k)$, nous substituons $a=1$ et

$$b_i = \beta_i(k)$$

l'équation (2.18) devient :

$$\sum_{i=0}^{k-1} (-i)^j \beta_i(k) = 1/j \quad j=1, 2, \dots, k \quad (2.20)$$

l'équation (2.20) représente un système d'équations linéaires avec les inconnues $\beta_0(k), \beta_1(k), \dots, \beta_{k-1}(k)$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & \dots & 1 \\ 0 & -1 & -2 & -3 & \dots & \dots & \dots & -(k-1) \\ 0 & 1 & 4 & 9 & \dots & \dots & \dots & -(k-1) \\ 0 & -1 & -8 & -27 & \dots & \dots & \dots & -(k-1) \\ \dots & \dots \\ 0 & (-1)^{k-1} & (-2)^{k-1} & (-3)^{k-1} & \dots & \dots & \dots & -(k-1) \end{bmatrix} \times \begin{bmatrix} \beta_0(k) \\ \beta_1(k) \\ \beta_2(k) \\ \beta_3(k) \\ \dots \\ \beta_{k-1}(k) \end{bmatrix} = \begin{bmatrix} 1 \\ 1/2 \\ 1/3 \\ 1/4 \\ \dots \\ 1/k \end{bmatrix}$$

Pour illustrer ceci on prend 2 cas

1 cas $k=1$

on a $\beta_0(1) = 1$, l'équation (2.19) devient:

$$x_{n+1} = x_n + h \cdot f(x_n, t_n) \quad \text{la formule de Forward-Euler}$$

2 cas $k=2$

Le système d'équation se réduit à :

$$\begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \beta_0(2) \\ \beta_1(2) \end{bmatrix} = \begin{bmatrix} 1 \\ 1/2 \end{bmatrix}$$

La résolution du système d'équations donne : $\beta_0(2) = 3/2$ et $\beta_1(2) = -1/2$

En remplaçant $\beta_0(2)$ et $\beta_1(2)$ dans l'équation (2.19) on a :

$$x_{n+1} = x_n + h \left[(3/2) f(x_n, t_n) - (1/2) f(x_{n-1}, t_{n-1}) \right]$$

pour $k=1$ l'erreur de troncature est :

$$\epsilon_t = (1/2) f''(\tau) \cdot h^2 \quad t_n < \tau < t_{n+1}$$

III.2 Algorithme d'ADAMS-MOULTON :

L'algorithme d'ADAMS-MOULTON est un algorithme implicite multi-pas est obtenu en remplaçant $p=k-2$, $a_1 = a_2 = \dots = a_{k-2} = 0$ dans l'équation (2.16), on a :

$$x_{n+1} = a_0 x_n + h \left[b_{-1} f(x_n, t_n) + b_0 f(x_n, t_n) + \dots + b_{k-2} f(x_{n-k+1}, t_{n-k+1}) \right]$$

Les $(k+1)$ coefficients sont obtenus en utilisant les équations (2.17) et (2.18) pour $a_i = 1$ et $b_i = \mu_i(k)$, nous avons l'équation suivante:

$$x_{n+1} = x_n + h \sum_{i=-1}^{k-2} \mu_i(k) f(x_{n-i}, t_{n-i}) \quad (2.21)$$

$$\sum_{i=-1}^{k-2} (-i)^{j-1} \mu_i(k) = 1/j \quad j=1, 2, \dots, k \quad (2.22)$$

où $x^{(k)}(t) = dx^{(k)}/dt^k$ pour $t = \tau$.

Pour $k=2$, nous avons l'algorithme trapézoïdal. Pour ces algorithmes d'ordre k , l'erreur de troncature diminue lorsque l'ordre augmente car elle est de la forme :

$$\epsilon_T = \frac{C}{k} x^{(k+1)}(\tau) h^{k+1} \quad (2.23)$$

$x^{(k+1)}(\tau)$: la dérivée $(k+1)$ ème de x pour $t = \tau$

C : constante qui dépend de l'ordre k

III.3 Choix optimum de l'ordre et des dimensions du pas

L'algorithme d'intégration numérique d'un certain ordre est choisi et est fixé durant le processus d'intégration. Sous cette hypothèse, la dimension du pas pour chaque intervalle de temps peut être optimisée en choisissant la valeur la plus large possible de h pour laquelle ϵ_T est limitée ($\epsilon < \epsilon_{max}$), et pour laquelle l'algorithme est stable. D'après les formules des algorithmes, le temps de calcul augmente avec l'ordre k . Par conséquent, il est intéressant de connaître la variation $\epsilon = f(k)$ pour différentes valeurs de k . L'erreur de troncature est donnée par la formule (2.23).

On définit l'erreur maximale par unité 100% de temps par :

$$e_{max} = \epsilon_{max} / T$$

T : intervalle de temps, dans lequel on cherche la solution

ϵ_{max} : erreur maximale admissible sur un intervalle de temps T

L'erreur maximale admissible par unité de pas est alors donnée par $h.e_{max}$

L'erreur de troncature est alors donnée par : $h.e_{max} = \epsilon_T$

d'où

$$e_{\max} = T / h = \left| C_k \cdot x^{(k+1)} \right| \cdot h^{k+1} / h$$

$$e_{\max} = \left| C_k \cdot x^{(k+1)} \right| \cdot h^k \quad (2.24)$$

L'équation (2.24) donne une famille de courbes représentant e_{\max} en fonction de h (Fig.2.7)

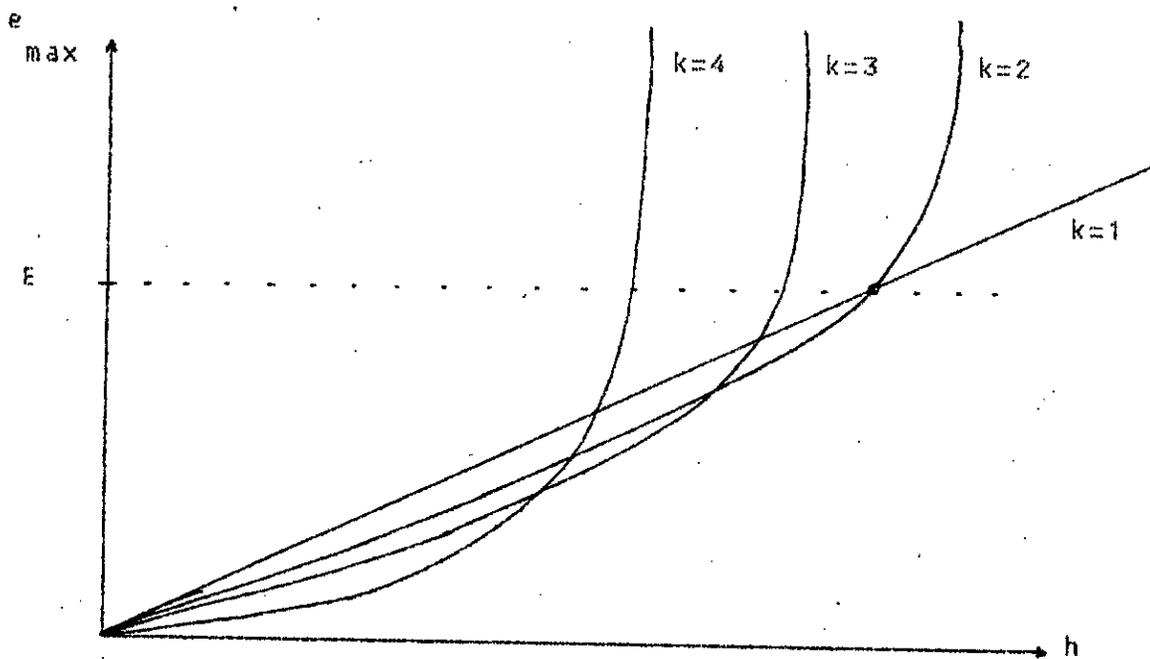


Fig.2.7 : Représentation de l'erreur maximale en fonction de h

D'après cette famille de courbes /22/, pour $e_{\max} > E$, l'algorithme d'ordre 1 donne un plus grande plage de h .

II.4 Régions de stabilité des algorithmes d'intégration numériques

Une méthode de détermination de la région de stabilité absolue a été développée par L.O CHUA /22/ qui donne l'équation suivante :

$$\sigma(\theta) = \frac{-e^{j(p+1)\theta} + a_0 e^{jp\theta} + a_1 e^{j(p-1)\theta} + \dots + a_p}{b_{-1} e^{-j(p+1)\theta} + b_0 e^{-jp\theta} + b_1 e^{-j(p-1)\theta} + \dots + b_p} \quad (2.25)$$

a) Algorithme d'ADAMS-BASHFORTH

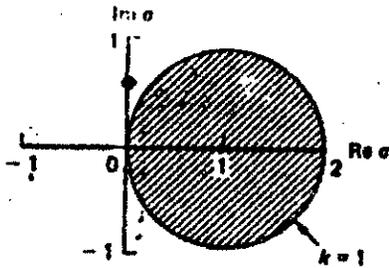
- Ordre 1 : Algorithme Forward-Euler

$$x^{n+1} = x^n + h \cdot f(x^n, t^n)$$

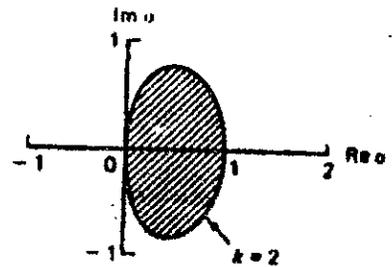
Les coefficients non nuls sont $a_0 = 1$ et $b_0 = 1$, en substituant ces coefficients dans (2.25) avec $p = k-1 = 0$, nous obtenons :

$$\sigma(\theta) = 1 - e^{j\theta} \quad 0 \leq \theta \leq 2\pi$$

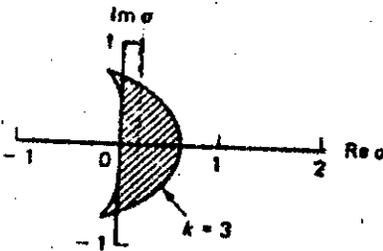
$\sigma(\theta)$ représente un cercle de centre : $\sigma = 1 + j0$, dans le plan complexe σ (Fig. 2.8.a). Même chose pour $k=2, 3, 4$, (voir Fig. 2.8b, c, d).



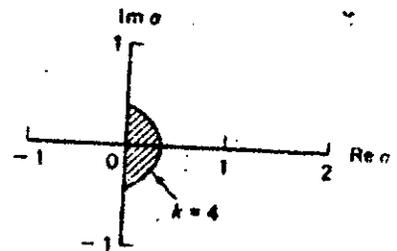
a : $k=1$



b : $k=2$



c : $k=3$



d : $k=4$

Fig. 2.8 : Région de stabilité

b) Algorithmes d'ADAMS-MOULION

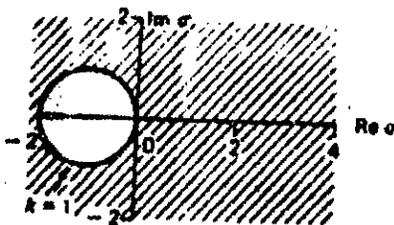
- Algorithme de Backward-Euler

$$x_{k+1} = x_k + h \cdot f(x_{k+1}, t_k)$$

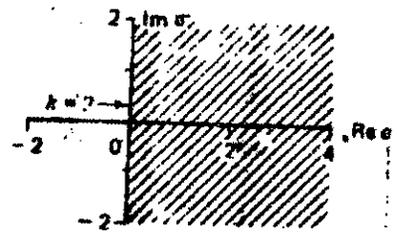
Les coefficients non nuls sont $a = 1$ et $b = -1$ en substituant ces coefficients dans l'équation (2.25) et en posant $p = k-2 = -1$, nous obtenons :

$$\sigma(\theta) = -1 + e^{-j\theta} \quad 0 \leq \theta \leq 2\pi$$

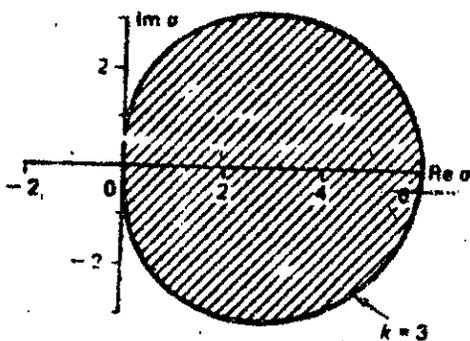
$\sigma(\theta)$ est un cercle de rayon unitaire et de centre : $\sigma = -1 + j0$ dans le plan complexe σ (Fig.2.9). On pose $z = e^{j\theta}$ et $P(z)$ le polynôme défini pour le critère de Root .



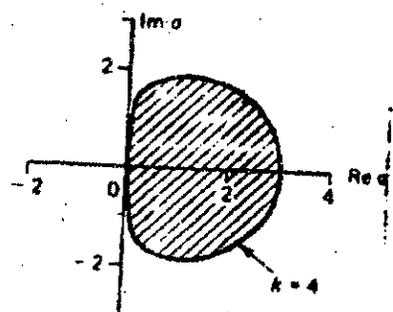
a : $k=1$



b : $k=2$



c : $k=3$



d : $k=4$

Fig.2.2 : Région de stabilité

Les racines de $P(z)$ satisfont à $|z| < 1$, donc il suit que l'algorithme de Backward-Euler est absolument stable à l'extérieur du cercle unitaire centré pour $z = -1 + j0$.

D'après les courbes tracées (Fig.2.8 et Fig.2.9), la région de stabilité absolue diminue lorsque l'ordre k augmente. Les Fig.2.8 a et b montrent que les algorithmes d'ordre 1 et 2 (Backward-Euler et trapézoidale) d'ADAMS-MOULTON sont stables pour un large domaine de variation de h . Le choix de h est limité seulement par la l'erreur maximale e_{max} .

En effet plusieurs programmes de simulation des circuits utilisent l'algorithme de Backward-Euler, qui est considéré comme étant le meilleur pour résoudre le problème de la valeur initiale, il est le plus stable, simple et n'impose aucune condition sur le choix de h .

IV Conclusion

Les algorithmes de formulation des équations (variable d'état et la théorie des graphes) présentent certaines difficultés pour l'élaboration d'un programme universel d'analyse des CI à MOS. La formulation des équations d'état, basée sur le modèle numérique linéaire du transistor MOS, ne donne pas une forme standard, alors la technique des variables d'état ne peut pas être utilisée. Par contre la théorie des graphes donne un graphe associé pour le modèle du transistor MOS. Mais son utilisation pour les CI à MOS, présente des complexités lorsque N_n augmente. Le nombre d'opérations est de l'ordre : $832.N_n^2.N_t$ et l'espace mémoire est de l'ordre : $288.N_t^2 + 4.N_n^2 + 72.N_n.N_t$.

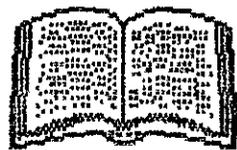
Lorsque N_n et N_t augmentent, cet algorithme devient lent et nécessite un espace mémoire important. Un autre algorithme sera envisagé au chapitre suivant, utilisant directement la topologie du circuit et est basé sur :

1) Modèle numérique - linéaire du transistor MOS

2) La loi $\sum_j I_j = 0$

3) Formule de Backward-Euler pour l'intégration de : $t = 0 \rightarrow \hat{t}$

Chapitre 3



Chapitre III

ELABORATION D'UN ALGORITHME D'ANALYSE TEMPORELLE DES CIRCUITS INTEGRES A MOS

I/ Introduction

L'algorithme d'analyse se subdivise en deux sous algorithmes de bases :

- Algorithme de formulation des équations
- Algorithme de résolution des équations •

Les méthodes de formulation des équations ont été présentées et appliquées aux C.I à MOS dans le chapitre précédent. La théorie des variables d'état présente certaines contraintes, pour le modèle numérique-linéaire du transistor MOS développé au chapitre I. La formulation des équations ne peut donner une forme standard, à cause de la présence des termes des sources de courant dépendantes, la technique de résolution ne s'applique pas aux formes non standards. Une autre approche souvent utilisée pour les circuits électriques : la théorie des graphes, à l'aide de la topologie du circuit, de la représentation électrique du modèle numérique-linéaire et de l'algorithme de formulation, un système d'équations linéaires est formulé.

La théorie des graphes présente deux complexités : mathématique et matérielle. Alors nous sommes amenés à développer un algorithme de formulation automatique des équations d'analyse, basé sur le modèle numérique-linéaire du transistor MOS.

Le principe de base, la technique de formulation ainsi que l'algorithme à développer sont introduits en premier lieu. Une étude numérique des méthodes de résolution est présentée en second lieu.

II/ Elaboration d'un algorithme de formulation d'un système d'équations linéaires d'analyse des C.I à MOS (sans capacités entre électrodes)
1/ Principe de base

Les hypothèses de base sont : la configuration du circuit est quelconque, les sources d'excitation peuvent être continues ou variables (régime statique ou dynamique) et les composants de bases sont : transistor MOS (type P et / ou type N) et les capacités parasites représentées entre le noeud quelconque et la masse.

2/ la méthode d'analyse

L'analyse des circuits linéaires et non linéaires est basée sur deux lois de base : la loi des courants (L.K.C) et la loi des tensions (L.K.V).

a) L.K.C : $\sum_i I_i = 0$ pour un noeud

b) L.K.V : $\sum_i V_i = 0$ pour une maille

et sur la caractéristique des éléments :

a) Eléments linéaires :

résistance : $V = R.I$

capacité : $I = C.dV/dt$

inductance : $V = L.dI/dt$

b) Eléments non linéaires

résistance : $V = R(I).I$

capacité : $I = C(V).dV/dt$

inductance : $V = L(I).dI/dt$

Le programme d'analyse est de première étape, pour sa simplification, nous considérons :

-Le modèle numérique linéaire du transistor MOS :

$$\left\{ \begin{array}{l} I_d = I_d + C_g (V_g - V_g) + C_d (V_d - V_d) + C_s (V_s - V_s) \\ I_s = I_d \\ I_g = 0 \end{array} \right.$$

-La capacité parasite C_p représentée entre un nœud de liaison des transistors et la masse est linéaire : $i = C_p \cdot dV/dt$

Les capacités entre électrodes C_{gd} et C_{gs} sont négligeables devant les capacités parasites.

Le programme d'analyse travaille directement sur la topologie du circuit sans passer à la représentation électrique d'une part et la grandeur électrique choisie est le courant : $I = f(V_g, V_s, V_d)$ d'autre part. Pour l'analyse du circuit, la loi des courants L.K.C. est plus intéressante à utiliser par rapport à la loi des tensions L.K.V.. Pour une topologie quelconque de N_n nœuds et N_t transistors (Fig.3.1) nous avons au nœud K :

$$\sum_{j=1}^{N_t} I_j(t) = 0 \quad (3.1)$$

Cette équation peut être transformée de la façon suivante :

$$\sum_{j=1}^{N_t} I_j(t) + I_c = 0 \quad (3.2)$$

Le premier terme représente la somme algébrique des courants de tous les transistors liés au nœud K , le second représente le courant de charge (ou de décharge) de la capacité C .

L'équation (3.2) devient :

$$\sum_{j=1}^{N_t} I_j(t) + C \cdot dV/dt = 0 \quad (3.3)$$

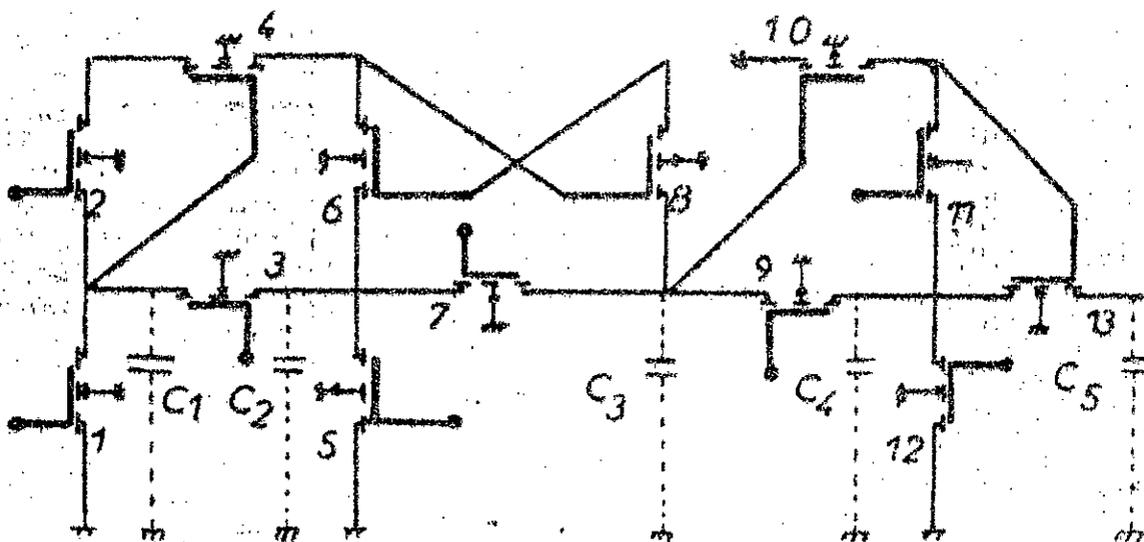


FIG.3.1 Topologie d'un circuit quelconque analysable

3/La méthode d'intégration

L'équation (3.3) est une équation différentielle de 1^{er} ordre, comme $I = f(V_1, V_2, V_3)$ est numérique, il faut utiliser une approche d'intégration numérique, les méthodes retenues sont Backward-Euler et Trapézoidale [22]. Pour illustrer nous appliquons les deux algorithmes pour un circuit simple (Fig.3.2)

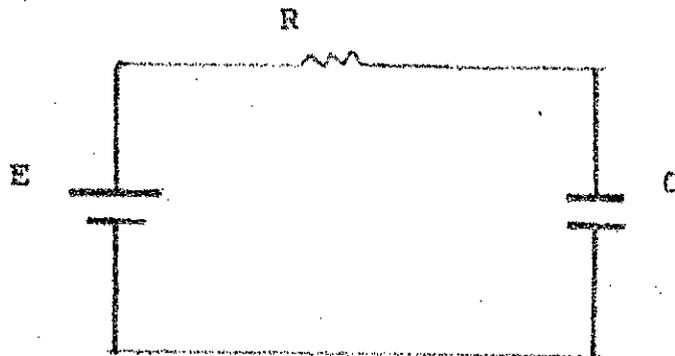


Fig.3.2 Circuit RC

-Intégration par Backward-Euler

Nous considérons le circuit de la Fig.3.2. La solution analytique est :

$$V_c = E(1 - e^{-t/\tau}) \quad \text{avec} \quad \tau = R.C$$

La solution numérique par Backward-Euler est :

$$V^{n+1} = \frac{V^n + (h/\tau)E}{1 + h/\tau} \quad (3.4)$$

-Intégration Trapézoidale

$$V^{n+1} = \frac{(1 - h/2\tau)V^n + (h/\tau)E}{1 + h/2\tau} \quad (3.5)$$

$$h = t_{n+1} - t_n$$

D'après les réponses obtenues (voir annexe courbes 1-8), la méthode trapézoidale est plus précise pour les valeurs de h faibles, par contre elle est instable pour les valeurs de $h > 3.\tau$ contrairement à la méthode de Backward-Euler.

Le programme d'analyse nécessite la simplicité, la rapidité et la stabilité ce qui nous mène à choisir la méthode de Backward-Euler. En intégrant l'équation (3.3) par Backward-Euler nous obtenons :

$$\sum_{j=1}^{N_t} I(j) + \frac{C}{h} (V_{L, n+1} - V_{L, n}) = 0 \quad (3.6)$$

4/ Formulation d'un système d'équations linéaires

La formule (3.6) est l'équation de base pour l'analyse de circuit. Le courant $I(j)$ est donné, pour un noeud quelconque, par le modèle suivant :

$$I_j = I_j + G_{gj} (V_{gj} - V_{gj}) + G_{dj} (V_{dj} - V_{dj}) + G_{sj} (V_{sj} - V_{sj}) \quad (3.7)$$

à l'instant t nous avons :

$$I_j = I_j + G_{gj} (V_{gj} - V_{gj}) + G_{dj} (V_{dj} - V_{dj}) + G_{sj} (V_{sj} - V_{sj}) \quad (3.8)$$

L'équation (3.6), à l'instant t et à la $n+1$ itération, appliquée à tous les noeuds (N_t noeuds) nous conduit à la forme algébrique matricielle suivante :

$$I + \frac{C}{h} V = 0 \quad (3.9)$$

En réalité, les N_t transistors MOS ne sont pas tous présents, à chaque noeud. La configuration du circuit est quelconque et son équation de base est celle de (3.6), pour satisfaire, nous sommes amenés à déterminer certains éléments de description topologique.

4.1/ Les éléments de description topologique du C.I à MOS

Dans cette partie nous allons présenter les matrices et les vecteurs topologiques d'un circuit intégré quelconque. Ces

derniers jouent le rôle de l'identité du circuit intégré à analyser. Pour un C.I quelconque de N_n noeuds et de N_t transistors nous devons connaître :

a/ Le sens des courants de tous les transistors liés à un noeud.

b/ Le terme $(V_x^{m+1} - V_x^m)$ de l'équation (3.7) est nul lorsque l'électrode x est liée à une tension externe (masse, tension d'alimentation, une source d'entrée).

c/ Dans le cas où les électrodes ne sont pas liées à des tensions externes, il faut les identifier à des tensions aux noeuds (les inconnues).

d/ Les différentes liaisons entre noeuds et entre transistors.

4.1.1/ Les matrices topologiques

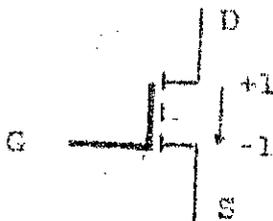
4.1.1.1/ La matrice de description du sens de courant :ND(Nn,Nt)

Pour le transistor MOS le sens du courant est :drain-source

La matrice ND permet d'écrire correctement l'équation (3.6) :

$$\sum_{j=1}^{N_t} ND(L,j) I(j) + C \left(\frac{V^{m+1,n+1}}{L} - \frac{V^{m+1,n}}{L} \right) / h = 0 \quad (3.10)$$

Par commodité nous définissons les éléments $ND(i,j)$ pour $i=1,2,\dots,N_n$ et $j=1,2,\dots,N_t$ de la façon suivante :



$$ND(i,j) = \begin{cases} +1 & \text{si le transistor } j \text{ est lié au noeud } i \text{ et le courant est sortant} \\ -1 & \text{si le transistor } j \text{ est lié au noeud } i \text{ et le courant est rentrant} \\ 0 & \text{si le transistor } j \text{ n'est pas lié au noeud } i \end{cases}$$

En appliquant l'équation (3.10) à tous les noeuds, nous obtenons

la forme matricielle suivante :

$$ND \cdot I + \frac{C}{h} \cdot V = 0 \quad (3.11)$$

ND : matrice nodale de dimensions $Nn \times Nt$

I : vecteur courant de dimension $Nt \times 1$ à l'itération $m+1$ et à l'instant t

C : matrice diagonale capacité de dimension $Nn \times Nn$

V : vecteur tension de dimension $Nn \times 1$ à l'itération $m+1$ et à l'instant t

L'équation (3.11) donne un système d'équations linéaires sous la forme :

$$ND(1,1)I(1) + ND(1,2)I(2) + \dots + ND(1,Nt)I(Nt) + \frac{C(1)}{h}V(1) - \frac{C(1)}{h}V(1) = 0$$

$$ND(2,1)I(1) + ND(2,2)I(2) + \dots + ND(2,Nt)I(Nt) + \frac{C(2)}{h}V(2) - \frac{C(2)}{h}V(2) = 0$$

.....

$$ND(Nn,1)I(1) + ND(Nn,2)I(2) + \dots + ND(Nn,Nt)I(Nt) + \frac{C(Nn)}{h}V(Nn) - \frac{C(Nn)}{h}V(Nn) = 0$$

Le courant $I(p)$ est donné par l'équation (3.8).

Si nous remplaçons les courants $I(p)$ pour $p=1,2,\dots,Nt$ dans le système d'équations précédent, nous constatons que les tensions à l'instant t et à la $m+1$ ème

sont inconnues :

- $V(i)$ pour $i=1,2,\dots,Nn$

- $Vg(j)$ pour $j=1,2,\dots,Nt$

- $Vd(j)$ pour $j=1,2,\dots,Nt$

- $Vs(j)$ pour $j=1,2,\dots,Nt$

Alors ce système de Nn équations possède $(Nn+3.Nt)$ inconnues.

néanmoins nous pouvons toujours réarranger ce système de façon à obtenir N_n inconnues. Les éléments $V_g(j)$, $V_d(j)$ et $V_s(j)$ sont identifiés à $V(i)$ pour $i=1,2,\dots,N_n$ et $j=1,2,\dots,N_t$.

Nous définissons des matrices permettant de connaître si :

a) une électrode S, D ou G d'un transistor j est liée ou non à un noeud i.

b) un noeud i est lié au noeud P par l'intermédiaire d'une électrode S, D ou G d'un transistor j.

4.1.1.2/ Matrices topologiques de description de la liaison transistor-noeud : S(Nn,Nt), D(Nn,Nt) et G(Nn,Nt)

Ces matrices sont définies comme suit :

$$S(i,j) = \begin{cases} 1 & \text{si la source du transistor j est liée au noeud i} \\ 0 & \text{si la source du transistor j n'est pas liée au noeud i} \end{cases}$$

$$D(i,j) = \begin{cases} 1 & \text{si le drain du transistor j est lié au noeud i} \\ 0 & \text{si le drain du transistor j n'est pas lié au noeud i} \end{cases}$$

$$G(i,j) = \begin{cases} 1 & \text{si le gate du transistor j est lié au noeud i} \\ 0 & \text{si le gate du transistor j n'est pas lié au noeud i} \end{cases}$$

D'après la définition de ces matrices nous avons :

$$S(i,j).V(i) = V_s(j) \quad (3.12)$$

$$D(p,j).V(p) = V_d(j) \quad i,p,k=1,2,\dots,N_n \quad (3.13)$$

$$G(k,j).V(k) = V_g(j) \quad j=1,2,\dots,N_t \quad (3.14)$$

L'équation du courant devient :

$$I(j) = I(j) + G_g(j)G(i,j)(V(i) - V(i)) + G_d(j)D(i,j)(V(i) - V(i)) + G_s(j).S(i,j).(V(i) - V(i)) \quad (3.15)$$

4.1.1.3/ Matrices topologiques de description des liaisons entre noeuds

Ces matrices tridimensionnelles indiquent les liaisons entre noeuds de chaque transistor. Le transistor MOS est défini par 3 électrodes, nous définissons les matrices : $LS(Nn, Nn, Nt)$, $LD(Nn, Nn, Nt)$ et $LG(Nn, Nn, Nt)$.

$$LS(p, i, j) = \begin{cases} 0 & \text{il n'existe pas de liaison entre les noeuds } i \text{ et } p \text{ par l'intermédiaire de la source du transistor } j \\ -1 & \text{il existe une liaison entre les noeuds } i \text{ et } p \text{ par l'intermédiaire de la source du transistor } j \end{cases}$$

$$LD(p, i, j) = \begin{cases} 0 & \text{il n'existe pas de liaison entre les noeuds } i \text{ et } p \text{ par l'intermédiaire du drain du transistor } j \\ 1 & \text{il existe une liaison entre les noeuds } i \text{ et } p \text{ par l'intermédiaire du drain du transistor } j \end{cases}$$

$$LG(p, i, j) = \begin{cases} 0 & \text{il n'existe pas de liaison entre le noeud } i \text{ et le noeud } p \text{ par l'intermédiaire du gate du transistor } j \\ 1 & \text{il existe une liaison entre les noeuds } i \text{ et } p \text{ par l'intermédiaire du gate-source du transistor } j \\ -1 & \text{il existe une liaison entre les noeuds } i \text{ et } p \text{ par l'intermédiaire du gate-drain du transistor } j \end{cases}$$

4.1.2/ Vecteurs topologiques

Ils permettent l'identification des liaisons externes pour les différentes électrodes.

$$HC(j) = \begin{cases} 0 & \text{si l'électrode G du transistor } j \text{ est liée à une liaison externe} \\ 1 & \text{sinon} \end{cases}$$

$$HS(j) = \begin{cases} 0 & \text{si l'électrode S du transistor } j \text{ est} \\ & \text{liée à une liaison externe} \\ 1 & \text{sinon} \end{cases}$$

$$HD(j) = \begin{cases} 0 & \text{si l'électrode D du transistor } j \text{ est} \\ & \text{liée à une liaison externe} \\ 1 & \text{sinon} \end{cases}$$

pour $j = 1, 2, \dots, N_T$

Exemples

a/ Vecteurs HG, HS, HD

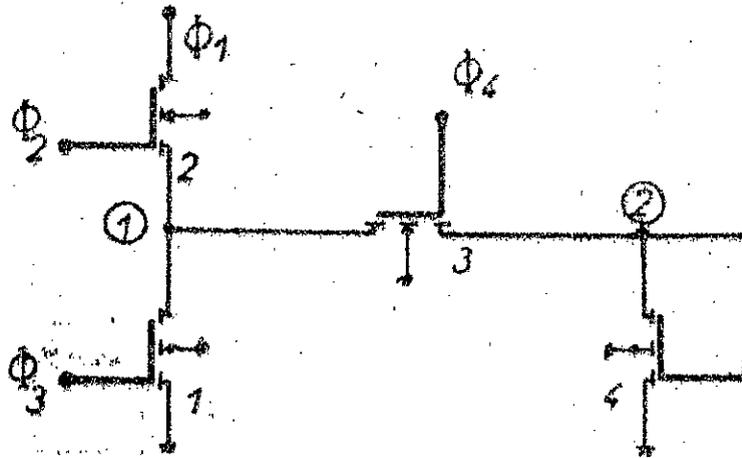


Fig. 3.3 Un circuit de 2x4

$$HG(4) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad HD(4) = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad HS(4) = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

b/ Matrices LG, LD, LS

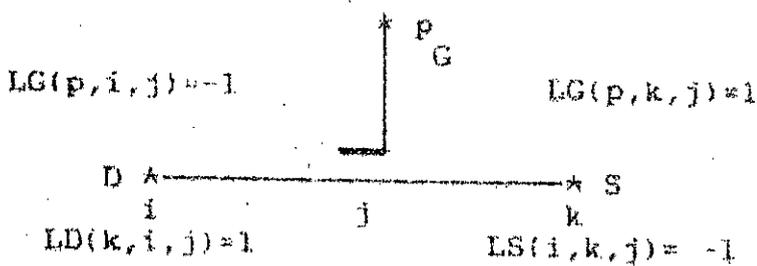


Fig. 3.5 Liaisons entre électrodes d'un transistor

c/ Matrices G, D, S

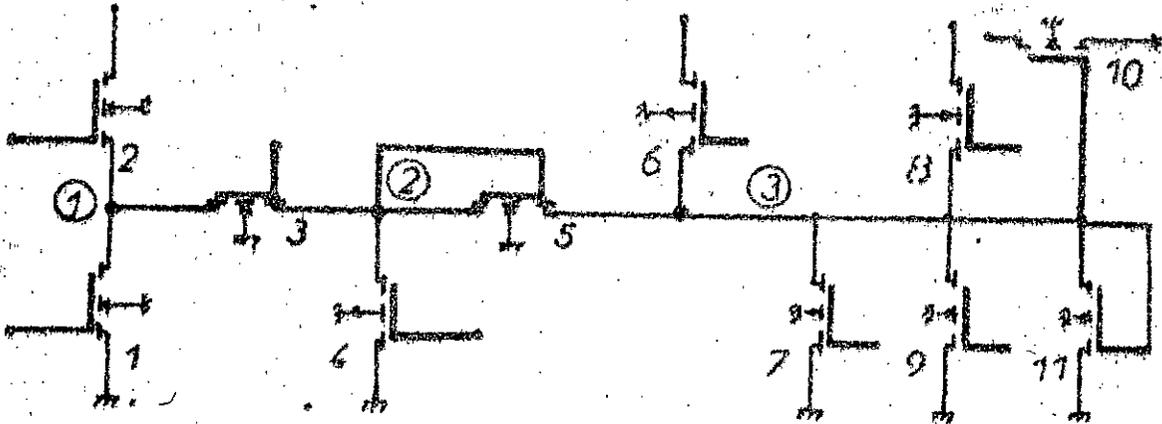


Fig. 3.6 : Un circuit de dimension 3x11

$$S(3,11) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; D(3,11) = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$G(3,11) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

4.2/ Elaboration d'un algorithme de formulation automatique des équations d'analyse

L'idée de base est de ramener le système précédent à un système de la forme :

$$G_c \cdot V = J$$

La détermination de G_c et J constitue l'objet de ce paragraphe.

L'équation matricielle d'un C.I. à MOS est :

$$ND.I + \frac{1}{h} \cdot C \cdot V - \frac{1}{h} \cdot C \cdot V = 0 \quad (3.16)$$

Le vecteur courant $I^{m+1,n+1}$ représente les N_t courants des transistors du C.I :

$$I^{m+1,n+1} = \begin{bmatrix} I^{m+1,n+1} \\ 1 \\ I^{m+1,n+1} \\ 2 \\ \vdots \\ I^{m+1,n+1} \\ N_t \end{bmatrix}$$

Le modèle numérique-linéaire du transistor MOS permet d'écrire :

$$\begin{bmatrix} I^{m+1,n+1} \\ 1 \\ I^{m+1,n+1} \\ 2 \\ \vdots \\ I^{m+1,n+1} \\ N_t \end{bmatrix} = \begin{bmatrix} I^{m,n+1} \\ 1 \\ I^{m,n+1} \\ 2 \\ \vdots \\ I^{m,n+1} \\ N_t \end{bmatrix} + \begin{bmatrix} G_{g1} & & & \\ & G_{g2} & & \\ & & \dots & \\ & & & G_{gN_t} \end{bmatrix} \begin{bmatrix} V_{g1} - V_{g1} \\ 1 \\ I^{m+1,n+1} - I^{m,n+1} \\ V_{g2} - V_{g2} \\ 2 \\ \vdots \\ I^{m+1,n+1} - I^{m,n+1} \\ V_{gN_t} - V_{gN_t} \\ N_t \end{bmatrix}$$

$$\begin{bmatrix} I^{m+1,n+1} \\ 1 \\ I^{m+1,n+1} \\ 2 \\ \vdots \\ I^{m+1,n+1} \\ N_t \end{bmatrix} = \begin{bmatrix} G_{d1} & & & \\ & \dots & & \\ & & G_{dN_t} & \\ & & & \dots \end{bmatrix} \begin{bmatrix} V_{d1} - V_{d1} \\ 1 \\ I^{m+1,n+1} - I^{m,n+1} \\ V_{d2} - V_{d2} \\ 2 \\ \vdots \\ I^{m+1,n+1} - I^{m,n+1} \\ V_{dN_t} - V_{dN_t} \\ N_t \end{bmatrix} + \begin{bmatrix} G_{s1} & & & \\ & \dots & & \\ & & \dots & \\ & & & G_{sN_t} \end{bmatrix} \begin{bmatrix} V_{s1} - V_{s1} \\ 1 \\ I^{m+1,n+1} - I^{m,n+1} \\ V_{s2} - V_{s2} \\ 2 \\ \vdots \\ I^{m+1,n+1} - I^{m,n+1} \\ V_{sN_t} - V_{sN_t} \\ N_t \end{bmatrix} \quad (3.17)$$

Une forme réduite :

$$I_d^{m+1,n+1} = I_d^{m,n+1} + G_g (V_g - V_g) + G_d (V_d - V_d) + G_s (V_s - V_s)$$

$$Id^{m+1,n+1} = (Gg^{m,n+1} \cdot Vg^{m+1,n+1} + Gd^{m,n+1} \cdot Vd^{m+1,n+1} + Gs^{m,n+1} \cdot Vs^{m+1,n+1}) + \\ + (Id^{m,n+1} - Gg^{m,n+1} \cdot Vg^{m+1,n+1} - Gd^{m,n+1} \cdot Vd^{m+1,n+1} - Gs^{m,n+1} \cdot Vs^{m+1,n+1})$$

Dans la première parenthèse nous avons les inconnues par contre la seconde est connue.

4.2.1/ Formulation de Gc

Pour le transistor j nous avons au noeud i :

$$Vg(j)^{m+1,n+1} = G(j,i)^{m+1,n+1} V(i)^{m+1,n+1} \quad (3.18)$$

$$Vd(j)^{m+1,n+1} = D(j,i)^{m+1,n+1} V(i)^{m+1,n+1} \quad (3.19)$$

$$Vs(j)^{m+1,n+1} = S(j,i)^{m+1,n+1} V(i)^{m+1,n+1} \quad (3.20)$$

L'écriture matricielle donne :

$$Vg^{m+1,n+1} = G^{m+1,n+1} V^{m+1,n+1}$$

$$Vd^{m+1,n+1} = D^{m+1,n+1} V^{m+1,n+1}$$

$$Vs^{m+1,n+1} = S^{m+1,n+1} V^{m+1,n+1}$$

or

1 cas i p=i

Pour j quelconque :

$$I^{m+1,n+1} = B_j^{m,n+1} + Gg_j^{m,n+1} Vg_j^{m+1,n+1} + Gs_j^{m,n+1} Vs_j^{m+1,n+1} + Gd_j^{m,n+1} Vd_j^{m+1,n+1} \quad (3.21)$$

$$I^{m+1,n+1} = B_j^{m,n+1} + (Gg(j)^{m,n+1} G(j,i)^{m+1,n+1} + Gs(j)^{m,n+1} S(j,i)^{m+1,n+1} + Gd(j)^{m,n+1} D(j,i)^{m+1,n+1}) V(i)^{m+1,n+1} \quad (3.22)$$

Dans cette équation nous n'avons pas tenu compte des liaisons entre noeuds.

ème

2 cas : p≠i

$$Vg(j)^{m+1,n+1} = LG(i,p,j)^{m+1,n+1} V(p)^{m+1,n+1} \quad (3.23)$$

$$Vd(j)^{m+1,n+1} = LD(i,p,j)^{m+1,n+1} V(p)^{m+1,n+1} \quad (3.24)$$

$$Vs(j)^{m+1,n+1} = LS(i,p,j)^{m+1,n+1} V(p)^{m+1,n+1} \quad (3.25)$$

$$I(j) = B(j) + Gg(j) LG(i,p,j) V(p) + Gd(j) LD(i,p,j) V(p) + Gs(j) LS(i,p,j) V(p)$$

$$I(j) = B(j) + (Gg(j) LG(i,p,j) + Gd(j) LD(i,p,j) + Gs(j) LS(i,p,j)) V(p) \quad (3.26)$$

Maintenant nous allons considerer les deux cas pour l'equation de base :

ème
à la i ligne

$$\sum_{j=1}^{Nt} ND(i,j) I(j) + \frac{C(i)}{h} V(i) - \frac{C(i)}{h} V(i) = 0 \quad (3.27)$$

avec

$$I(j) = B(j) + (Gg(j) G(j,i) + Gs(j) S(j,i) + Gd(j) D(j,i)) V(i) + (Gg(j) LG(i,p,j) + Gs(j) LS(i,p,j) + Gd(j) LD(i,p,j)) V(p)$$

En remplaçant $I(j)$ dans l'equation (3.27) nous avons :

$$\sum_j ND(i,j) (B(j) + (Gg(j) G(j,i) + Gs(j) S(j,i) + Gd(j) D(j,i)) V(i) + (Gg(j) LG(i,p,j) + Gs(j) LS(i,p,j) + Gd(j) LD(i,p,j)) V(p)) + \frac{C(i)}{h} V(i) - \frac{C(i)}{h} V(i) = 0 \quad (3.28)$$

En posant $A = \sum_j ND(i,j) B(j)$ nous avons :

$$\sum_{j=1}^{Nt} ND(i,j) ((Gg(j) LG(i,p,j) + Gs(j) LS(i,p,j) + Gd(j) LD(i,p,j)) V(p)) + ND(i,j) (Gg(j) G(j,i) + Gs(j) S(j,i) + Gd(j) D(j,i)) V(i) + \frac{C(i)}{h} V(i) - \frac{C(i)}{h} V(i) + A = 0 \quad (3.29)$$

Pour simplifier l'équation, nous posons :

$$G(p) = \sum_{j=1}^{m,n+1} ND(i,j) (Gg(j) LG(i,p,j) + Gs(j) LS(i,p,j) + Gd(j) LD(i,p,j))$$

$$G(i) = \sum_{j=1}^{m,n+1} ND(i,j) (Gg(j) G(j,i) + Gs(j) S(j,i) + Gd(j) D(j,i)) + \frac{C(i)}{h}$$

$$J(i) = A + \frac{C(i)}{h} V(i)$$

L'équation (3.27) devient pour le $i^{\text{ème}}$ noeud :

$$Gc(p) V(p) + Gc(i) V(i) = J(i) \quad (3.30)$$

ou bien :

$$Gc(p) V(p) + Gc(i) V(i) + Gc(p) V(p) = J(i)$$

En appliquant l'équation (3.30) pour les Nn noeuds nous obtenons le système suivant :

$$Gc(1,1)V(1) + Gc(1,2)V(2) + \dots + Gc(1,i)V(i) + \dots + Gc(1,Nn)V(Nn) = J(1)$$

$$Gc(2,1)V(1) + Gc(2,2)V(2) + \dots + Gc(2,i)V(i) + \dots + Gc(2,Nn)V(Nn) = J(2)$$

$$Gc(i,1)V(1) + Gc(i,2)V(2) + \dots + Gc(i,i)V(i) + \dots + Gc(i,Nn)V(Nn) = J(i)$$

$$Gc(Nn,1)V(1) + Gc(Nn,2)V(2) + \dots + Gc(Nn,i)V(i) + \dots + Gc(Nn,Nn)V(Nn) = J(Nn)$$

Avec :

$$G_c(i,p) = \begin{cases} \sum_{j=1}^{N_t} ND(i,j) (G_g(j)G(j,i) + G_s(j)S(j,i) + G_d(j)D(j,i)) + \frac{C(i)}{h} & \text{si } p=i \\ \sum_{j=1}^{N_t} ND(i,j) (G_g(j)LG(i,p,j) + G_s(j)LS(i,p,j) + G_d(j)LD(i,p,j)) & \text{si } p \neq i \end{cases} \quad (3.31)$$

Les coefficients de la matrice conductance G_c sont donnés par $G_c(i,p)$. La formule qui donne les éléments diagonaux est différente de celle des non diagonaux, ceci est dû au fait que les coefficients de la diagonale proviennent des éléments du noeud considéré (capacité, conductances des électrodes liées à ce noeud) par contre les non diagonaux proviennent des liaisons entre noeuds, d'où l'absence des termes de capacité.

4.2.2/ Formulation de J

Jusqu'à présent nous n'avons pas considéré le 2^{ème} membre de

$$l'équation : G_c \begin{matrix} m,n+1 \\ V \end{matrix} = \begin{matrix} m,n+1 \\ J \end{matrix}$$

$$\text{avec } J(i) = \frac{C(i)}{h} V(i) - \sum_{j=1}^{N_t} ND(i,j) B(j) \quad i=1,2,\dots,N_n \quad (3.33)$$

$$\text{ou } B(j) = I(j) - G_g(j) V_g(j) - G_s(j) V_s(j) - G_d(j) V_d(j) \quad j=1,2,\dots,N_t \quad (3.34)$$

Dans le modèle du courant les termes $(V_g^{m+1,n+1} - V_g^{m,n+1})$, $(V_s^{m+1,n+1} - V_s^{m,n+1})$ ou $(V_d^{m+1,n+1} - V_d^{m,n+1})$ sont nuls lorsque l'électrode G, S ou D sont liées à des tensions externes. Les vecteurs HG, HS et HD introduits dans le paragraphe (4.1.2), nous permettent de tenir compte de ce type de liaison. Alors nous avons les équations d'identifications suivantes:

$$HG(j) Vg(j) = \begin{cases} Vg(j) & \text{si } G \text{ n'est pas liée à une tension externe} \\ 0 & \text{si } G \text{ est liée à une tension externe} \end{cases}$$

$$HS(j) Vs(j) = \begin{cases} Vs(j) & \text{si } S \text{ n'est pas liée à une tension externe} \\ 0 & \text{si } S \text{ est liée à une tension externe} \end{cases}$$

$$HD(j) Vd(j) = \begin{cases} Vd(j) & \text{si } D \text{ n'est pas liée à une tension externe} \\ 0 & \text{si } D \text{ est liée à une tension externe} \end{cases}$$

Alors dans l'équation (3.34) nous considérons : $HG(j) Vg(j)$

$HS(j) Vs(j)$ et $HD(j) Vd(j)$ au lieu de $Vg(j)$, $Vs(j)$ et $Vd(j)$ seules.

L'équation (3.33) devient :

$$J(i) = \frac{C(i)}{h} V - \sum_{j=1}^{Nl} ND(i, j) (I(j) - Gg(j) HG(j) Vg(j) + Gs(j) HS(j) Vs(j) + Gd(j) HD(j) Vd(j)) \quad i=1, 2, \dots, Nn \quad (3.35)$$

Ainsi les équations (3.31), (3.32) et (3.35) nous permettent de formuler le système : $G V = J$, avec V vecteur tension inconnu

III/ Modification de l'algorithme de formulation des équations linéaires d'analyse des C.I à MOS.

En effet la précision de l'algorithme s'améliore si nous tenons compte des capacités entre électrodes dans le modèle

numérique-linéaire du transistor MOS (voir chapitre I). Les courants des capacités C_{gs} et C_{gd} sont donnés par :

$$I_{gs} = C_{gs} \frac{dV_{gs}}{dt}$$

$$I_{gd} = C_{gd} \frac{dV_{gd}}{dt}$$

Un simple procédé de modification sera utilisé, l'hypothèse de base est de considérer les capacités entre électrodes comme étant des capacités externes au transistor MOS.

Transistor avec capacité entre électrodes = transistor simple + $C_{gs} + C_{gd}$

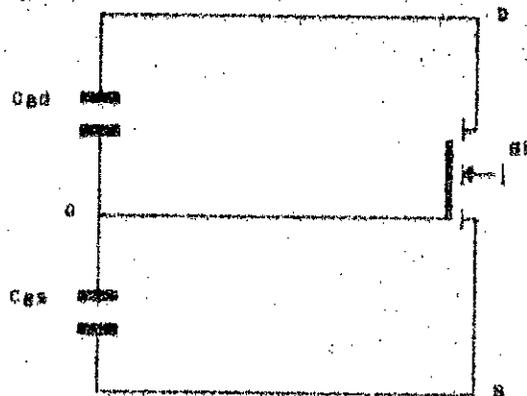


Fig. 3.7 : Représentation du transistor MOS les capacités C_{gd} et C_{gs}

Les équations sont réécrites sous la forme :

$$I_{gs} = C_{gs} \left(\frac{dV_s}{dt} - \frac{dV_g}{dt} \right) \text{ ou } I_{gs} = C_{gs} (\dot{V}_s - \dot{V}_g) \quad (3.36)$$

$$I_{gd} = C_{gd} \left(\frac{dV_d}{dt} - \frac{dV_g}{dt} \right) \text{ ou } I_{gd} = C_{gd} (\dot{V}_d - \dot{V}_g) \quad (3.37)$$

Nous considérons un circuit intégré à MOS (Fig.3.8) quelconque à N_n noeuds , N_t transistors et $(2.N_t + N_n)$ capacités (entre électrodes et parasites).

Au noeud i quelconque nous avons :

$$\sum_{j=1}^{Nt} ND(i,j) I(j) + I_c(i) + \sum_{j=1}^{Nt} (I_{gs}(j) + I_{gd}(j)) = 0 \quad (3.38)$$

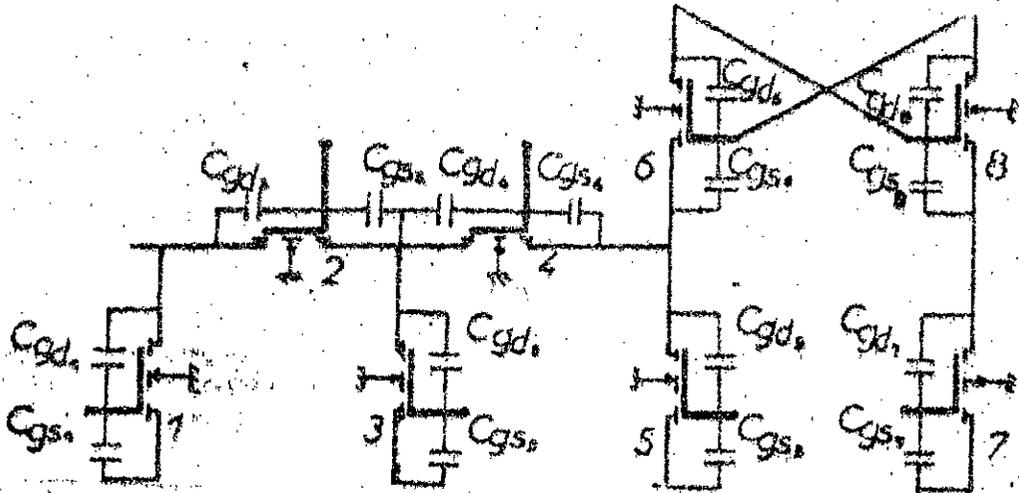


Fig. 3.8 : Topologie quelconque pour les C.I. avec effet des capacités entre électrodes

nous avons déjà établi :

$$\sum_{j=1}^{Nt} ND(i,j) I(j) + I_c(i) = G_c V - J$$

Alors l'équation (3.38) devient :

$$G_c(i,p) V(p) - J(i) + \sum_{j=1}^{Nt} (I_{gs}(j) + I_{gd}(j)) = 0$$

Nous remplaçons les courants I_{gs} et I_{gd} par leurs formules (3.36) et (3.37) :

$$G_c(i,p) V(p) - J(i) + \sum_{j=1}^{Nt} (C_{gs}(j) V_{gs}(j) + C_{gd}(j) V_{gd}(j)) = 0$$

$$G_c(i,p) V(p) - J(i) + \sum_{j=1}^{Nt} ((C_{gs}(j) V_s(j) + C_{gd}(j) V_d(j)) - (C_{gs}(j) + C_{gd}(j)) V_q(j)) = 0 \quad (3.39)$$

Les coefficients $G_c(i,p)$ et $J(i)$ sont donnés par les équations (3.31), (3.32) et (3.35), il reste uniquement à transformer le reste de l'équation (3.39), celle-ci contient les dérivées V_s , V_d et V_g alors nous devons procéder à une intégration numérique (Backward-Euler).

$$V_s = \frac{V_s^{n+1} - V_s^n}{h}$$

$$V_d = \frac{V_d^{n+1} - V_d^n}{h}$$

$$V_g = \frac{V_g^{n+1} - V_g^n}{h}$$

L'équation (3.39) devient à l'instant t_{n+1} et à la $(n+1)^{\text{ème}}$ itération:

$$G_c(i,p)V(p)^{m,n+1} - J(i)^{m,n+1} + \sum_{j=1}^{M_t} \left(\frac{C_{gs}(j)}{h} (V_s^{m+1,n+1} - V_s^{m+1,n}) + \frac{C_{gd}(j)}{h} (V_d^{m+1,n+1} - V_d^{m+1,n}) - \frac{(C_{gs}(j)+C_{gd}(j))}{h} (V_g^{m+1,n+1} - V_g^{m+1,n}) \right) = 0 \quad (3.40)$$

Les tensions-électrodes V_s , V_d , et V_g peuvent être soit des tensions inconnues (parmi les $V(i)$), soit des tensions externes, pour cela nous avons :

pour $p=i$, nous utilisons les équations (3.18), (3.19) et (3.20)

pour $p \neq i$, les équations (3.23), (3.24) et (3.25) sont utilisées.

L'équation (3.40) devient :

$$G_c(i,p)V(p)^{m,n+1} - J(i)^{m,n+1} + \sum_{j=1}^{M_t} \left(\frac{C_{gs}(j)}{h} V_s^{m+1,n+1} + \frac{C_{gd}(j)}{h} V_d^{m+1,n+1} - \frac{(C_{gs}(j)+C_{gd}(j))}{h} V_g^{m+1,n+1} \right) - \left(\frac{C_{gs}(j)+C_{gd}(j)}{h} V_g^{m+1,n} - \frac{C_{gs}(j)}{h} V_s^{m+1,n} - \frac{C_{gd}(j)}{h} V_d^{m+1,n} \right) = 0 \quad (3.41)$$

Nous cherchons à ramener l'équation (3.41) à la forme suivante :

$$G_c(i,p)V(p) - J'(i) + G_c(i,p)V(p) = 0$$

De la même façon que précédemment, nous obtenons :

$$G_c(i,p)V(p) - J'(i) + \sum_{j=1}^{N_t} \left(\frac{C_{gs}(j)}{h} S(j,i) + \frac{C_{gd}(j)}{h} D(j,i) - \left(\frac{C_{gs}(j)+C_{gd}(j)}{h} \right) G(j,i) \right) V(i) + \sum_{j=1}^{N_t} \left(\frac{C_{gs}(j)}{h} LS(i,p,j) + \frac{C_{gd}(j)}{h} LD(i,p,j) - \left(\frac{C_{gs}(j)+C_{gd}(j)}{h} \right) LG(i,p,j) \right) V(p) = 0$$

C'est de la forme : $G_c(i,p)V(p) - J'(i) = 0$ pour $i = 1, 2, \dots, N_n$
avec :

$$G_c'(i,p) = \begin{cases} G_c(i,i) + \sum_{j=1}^{N_t} \left(\frac{C_{gs}(j)}{h} S(j,i) + \frac{C_{gd}(j)}{h} D(j,i) - \left(\frac{C_{gs}(j)}{h} + \frac{C_{gd}(j)}{h} \right) G(j,i) \right) & \text{si } p=i \\ G_c(i,p) + \sum_{j=1}^{N_t} \left(\frac{C_{gs}(j)}{h} LS(i,p,j) + \frac{C_{gd}(j)}{h} LD(i,p,j) - \left(\frac{C_{gs}(j)}{h} + \frac{C_{gd}(j)}{h} \right) LG(i,p,j) \right) & \text{si } p \neq i \end{cases}$$

Les coefficients $G_c(i,p)$ sont donnés par les équations (3.31) et (3.32).

De même pour J' nous avons :

$$J'(i) = J(i) - \sum_{j=1}^{N_t} \left(\frac{C_{gs}(j)}{h} V_s(j) + \frac{C_{gd}(j)}{h} V_d(j) - \left(\frac{C_{gs}(j)+C_{gd}(j)}{h} \right) V_g(j) \right) \quad (3.42)$$

Nous utilisons les équations d'identification du paragraphe 4.2.2 :

$$J'(i) = J(i) - \sum_{j=1}^{N_t} \left(\frac{C_{gs}(j)}{h} HS(j)V_s(j) + \frac{C_{gd}(j)}{h} HD(j)V_d(j) - \left(\frac{C_{gs}(j)+C_{gd}(j)}{h} \right) HG(j)V_g(j) \right)$$

Alors nous sommes amenés à résoudre la nouvelle équation matricielle

pour $V^{m+1, n+1}$ vecteur tension inconnu, à chaque itération m suivante

$$G_c^{m, n+1} V^{m+1, n+1} = J^{m, n+1}$$

IV/ Etude numérique des méthodes de résolution d'un système d'équations linéaires

Les réponses du CI à MOS sont obtenues en résolvant le système d'équations formulé au paragraphes II et III ($G_c V = J$).

Le nombre de composants par circuit est important, la méthode de résolution la plus puissante sera choisie. Ceci nous conduit à une étude numérique des différentes méthodes de résolution.

Suivant les caractéristiques de la matrice G_c , nous avons deux cas :

- 1/ La matrice G_c est de taille réduite et est pleine.
- 2/ La matrice G_c est de grande taille et est éparse.

IV.1/ Les méthodes directes

IV.1.1/ Méthode de Gauss

Elle consiste à transformer le système $G.V = J$ à matrice G quelconque en un système équivalent $G'' V = J''$ ou G'' est une matrice triangulaire supérieure [24].

Le nombre d'opérations est :

$$N_m = N_n(N_n - 1)(2N_n + 5)/6$$

$$N_a = N_m$$

$$N_d = N_n(N_n + 1)/2$$

Pour $N_n > 20$ nous avons $N_m = N_a = N_n^3 / 3$

IV.1.2 /Méthode de Gauss-Jordan

Elle consiste à transformer /24/ le système $Gc V = J$ en un système équivalent $I V = Y$ où I est une matrice identité d'ordre Nn . Le nombre d'opérations est :

$$N_m = N_a = Nn \left(\frac{Nn - 1}{2} \right)$$

$$N_d = Nn(Nn+1)$$

Pour $Nn > 20$ nous avons $N_m = N_a = \frac{Nn^3}{2}$

IV.1.3/ la méthode de décomposition LU

La décomposition de $G = LU$ donne :

$$LY = J \quad \text{et} \quad UV = Y$$

Le nombre d'opérations est de l'ordre $Nn^3/3$.

Les méthodes directes sont surtout employées pour les petits systèmes d'équations.

IV.2/ Les méthodes itératives

Ils existent différentes méthodes itératives permettant de calculer la solution du système linéaire. Pour résoudre le système $G V = J$ /24/, on pose : $G = A' - B'$

$$(A' - B') V = J$$

d'où $V = A'^{-1} B' V + A'^{-1} J$

La méthode itérative associée à l'égalité précédente consiste, à partir d'un vecteur initial V^0 , à générer la suite V^1, V^2, \dots, V^{k+1} , en utilisant l'équation suivante :

$$V = A'^{-1} B' V + A'^{-1} J$$

$$V^{k+1} = \alpha V^k + \beta$$

où $\alpha = A'^{-1} B'$ et $\beta = A'^{-1} J$ $k = 1, 2, \dots$

Le problème qui se pose est de trouver la décomposition sous la

forme $(A' - B')$ de telle façon que A' soit inversible et vérifie $\rho(A' - B') < 1$ avec ρ rayon spectral /24/ (condition de convergence).

IV.2.1/ La méthode de Jacobi

Le principe /25/-/26/ est de décomposer G de la façon suivante:

$$G = D - (L + U) = A' - B'$$

$$D V = (L + U) V + J$$

$$V = D^{-1} (L + U) V + D^{-1} J$$

Algorithme de calcul :

$$V(i) = (J(i) - \sum_{j=1}^{Nn} G(i,j) V(j)) / G(i,i) \quad i \neq j, \quad i=1, \dots, Nn$$

test d'arrêt :

$$|V(i)^{k+1} - V(i)^k| < \epsilon \quad i=1, \dots, Nn$$

$$k = 1, 2, \dots, k_{max}$$

Nombre d'opérations : $Na = Nm = Nn$ par itération

Espace mémoire : elle nécessite de ranger en mémoire G , J , V et V soit $(Nn + 3Nn)$ places mémoires.

La convergence : $\sum_j |G(i,j)| < G(i,i)$

IV.2.2/ La méthode de Gauss-Seidel

La matrice G étant décomposée en : $G = (D - L) - U$ /26/

$$((D - L) - U) V = J$$

$$D V = L V + U V + J$$

soit encore

$$V = (D - L)^{-1} U V + (D - L)^{-1} J$$

Algorithme de calcul est :

$$V(i) = (J(i) - \sum_{j=1}^{i-1} G(i,j) V(j) - \sum_{j=i+1}^{Nn} G(i,j) V(j)) / G(i,i)$$

test d'arrêt :

$$|V^{(i)}(i)^{k+1} - V^{(i)}(i)^k| < \epsilon$$

$i = 1, 2, \dots, Nn$

$k = 1, 2, \dots, kmax$

Nombre d'opérations :

$$Na = Nm = Nn(Nn - 1)$$

Nombre de places mémoires : la méthode Gauss-Seidel nécessite 2 ranger en mémoire G, J et V soit au total $Nn + 2Nn$.

La méthode de Gauss-Seidel converge si G est à diagonale fortement dominante

$$\left(\sum_j \frac{|G(i,j)|}{G(i,i)} < 1 \right)$$

La méthode Gauss-Seidel est une grande amélioration par rapport à celle de Jacobi : Jacobi nécessite $Nn + 3Nn$ places mémoires alors que Gauss-Seidel nécessite $Nn + 2Nn$. Cette dernière est plus rapide et converge plus vite que celle de Jacobi.

IV.2.3/ Méthode de relaxation

Principe

w est une constante scalaire réelle non nulle.

$$V^{(i)}(i)^{k+1} = V^{(i)}(i)^k + w (V^{(i)}(i)^k - V^{(i)}(i)^{k-1})$$

$V^{(i)}(i)^{k+1}$ estimé par Gauss-Seidel

si $w = 1$ méthode Gauss-Seidel

si $w > 1$ méthode de surrelaxation

si $w < 1$ méthode de sousrelaxation

Algorithme de calcul /24/-/26/

$$V^{(i)}(i)^{k+1} = V^{(i)}(i)^k + \frac{w}{G(i,i)} (J(i) - \sum_{j=1}^{i-1} G(i,j) V^{(j)}(j)^{k+1} - \sum_{j=i+1}^{Nn} G(i,j) V^{(j)}(j)^k)$$

$i = 1, 2, \dots, Nn$

test d'arrêt si :

$$|V(i)^{k+1} - V(i)^k| < \epsilon \quad k = 1, 2, \dots, k_{max}$$

Nombre d'opérations par itération : $Nm \times Nn$

Nombre de places mémoires : $Nn + 2Nn$

La convergence est assurée /26/ si $0 < w < 2$

IV.3/ Méthodes d'optimisation

Soit le système linéaire $GV = J$ à matrice G symétrique, définie positive /27/ (si G est quelconque on peut se ramener à un système à matrice symétrique définie positive par prémultiplication par G : $G^t G V = G^t J$).

Nous définissons le résidu $r = J - G V$, $r, J, V \in \mathbb{R}^{Nn}$ et la forme quadratique est :

$$E(V) = \frac{1}{2} (V - V^*)^t G (V - V^*)$$

$$V = G^{-1} (J - r) \quad \text{avec} \quad V^* = G^{-1} J$$

d'où
$$E(V) = \frac{1}{2} r^t G^{-1} r$$

La solution V^* du système $GV = J$ correspond au vecteur minimisant $E(V)$, alors nous pouvons ramener la solution de système à celui de minimisation de la forme quadratique

$$E(V) = \frac{1}{2} r^t G^{-1} r$$

Parmi les méthodes d'optimisation nous présentons celle qui est la plus courante.

Méthode du gradient conjugué

Cette méthode consiste à construire une suite de vecteurs V telle que les résidus $r = J - G V$ forme un système orthogonal ; dans ces conditions $r = 0$ et en théorie la solution est obtenue

en au plus N_n itérations, mais la rapidité de convergence dépend fortement du conditionnement de la matrice /28/.

Principe de base de la méthode du gradient conjugué (GC)

Si $M \in \mathbb{R}^{N_n \times N_n}$, facilement inversible, on définit la suite des vecteurs V par :

$$1/ M z^k = r^k = J - G V^k$$

$$2/ V^{k+1} = V^k + w^k (\alpha^k z^k + V^k - V^k)$$

$$3/ \alpha^k = \frac{ (r^k, r^k) }{ (G M r^k, r^k) }$$

$$4/ w^k = \frac{ w^{k-1} \alpha^{k-1} (r^{k-1}, r^{k-1}) }{ w^{k-1} \alpha^{k-1} (r^{k-1}, r^{k-1}) - \alpha^k (r^k, r^k) }$$

avec $w^0 = 1$, (x,y) représente le produit scalaire de deux vecteurs x et y

Pour les matrices symétriques on ne possède pas encore une méthode aussi générale et aussi puissante que la méthode /28/-/29/ du gradient conjugué. Voici un tableau de comparaison avec $N_n = 100$ sur Var 75/11 /30/ .

	Gauss-Seidel	Surrelaxation	GC
Nbre d'itération	201	87	12
Temps de calcul en centième de seconde	443	192	29

Tableau 4.1

La méthode GC apparait aujourd'hui comme le meilleur compromis entre une méthode directe et une méthode itérative.

V/Conclusion

Nous avons établi deux algorithmes de formulation automatique d'un système d'équations d'analyse des circuits intégrés à MOS. L'algorithme tenant compte des capacités entre électrodes présente un degré de complexité plus élevé que le premier. Pour l'élaboration du PATMOS nous considérons le premier cas (sans capacités entre électrodes). Une étude numérique sur les méthodes de résolution a été faite afin de déterminer la technique la plus puissante pour les valeurs de N_n élevées .

Chapitre 4



Chapitre IV

PAIMOS : PROGRAMME D'ANALYSE DES C.I. MOS ET SES APPLICATIONS

I/ Introduction

Les programmes de simulation des circuits sont des outils importants pour l'analyse des performances électriques des C.I. contenant un grand nombre d'éléments actifs non linéaires. Pour les simulateurs de C.I. à MOS, le point le plus commun et le plus cher en temps de calcul est l'analyse transitoire non linéaire.

Jusqu'à présent nous avons traité les différents points nécessaires afin de réaliser l'algorithme d'analyse sans préoccuper du temps de calcul, de l'espace mémoire nécessaire à l'analyse et du nombre de données d'entrées. Dans l'algorithme de formulation nous avons des matrices topologiques et le pas d'intégration dans le domaine temporel. L'optimisation de ces grandeurs nous conduit à l'amélioration du programme en espace mémoire et en temps de calcul et en nombre de données d'entrées. Alors nous sommes amenés à choisir correctement les matrices topologiques, nécessitant un minimum de données d'une part et pas d'intégration d'autre part.

Les approximations mathématiques ou modèles des caractéristiques électriques sont essentielles dans n'importe quelle procédure d'analyse des circuits d'où la nécessité de créer une bibliothèque de modèles utilisés particulièrement dans l'analyse assistée par ordinateur des C.I.

II/ Les modèles du transistor MOS

Pour des raisons de simplicité, les modèles les plus utilisés par des programmes de simulation sont : le modèle de Sah /13/, le modèle de Sah-Pao /14/, le modèle de Frohman-Bentchkowsky /15/ /16/ le modèle de Shichman-Hodges /34/ et de G.Merckel, J.Ber et Z.Cupcea /18/-/19/.

II.1/ Modèle de Sah

C'est un modèle simple ayant une précision relative de 25% du courant I_{ds} , basé sur le principe de fonctionnement du transistor MOS, mais ne tient pas compte de l'effet du substrat. Les caractéristiques électriques sont données par :

$$\begin{aligned} - \text{si } V_{gs} < V_t & \quad I_{ds} = 0 \\ - \text{si } V_{gs} > V_t & \quad I_{ds} = K(2(V_{gs} - V_t) - V_s - V_d)(V_d - V_s) \quad V_{ds} < V_{gs} - V_t \\ & \quad I_{ds} = K(V_{gs} - V_t - V_s)^2 \quad V_{ds} > V_{gs} - V_t \end{aligned}$$

$$\text{avec } K = K_0 \frac{W}{L} \quad ; \quad K_0 = \frac{\mu C_0}{2}$$

II.2/ Modèle de Sah-Pao

C'est un modèle avec une précision de 20% qui tient compte de l'effet du substrat (η) :

$$\begin{aligned} V_{gs} < V_t & \quad I_{ds} = 0 \\ V_{gs} > V_t & \quad I_{ds} = K_e(2V_e - V_s - V_d)(V_d - V_s) \quad \text{si } V_d < V_e \\ & \quad I_{ds} = K_e(V_e - V_s)^2 \quad \text{si } V_d > V_e \end{aligned}$$

$$K_e = K(1 + \eta) \quad ; \quad V_e = \frac{V_{gs} - V_t}{1 + \eta}$$

II.3/ Modèle de Frohman-Bentchkowsky

C'est un modèle simple, souvent utilisé par les simulateurs de

C.I. . Il est plus précis que les modèles précédents (15%).
 l'équation générale du courant Id est donnée par :

$$I_d = K((V_g - V_{fb} - 2\phi_f - V_d/2)V_d - (V_g - V_{fb} - 2\phi_f - V_s))$$

Ce modèle est donné dans le programme MOSTAP /7/ sous une forme plus simple , nécessitant la génération de trois tables de V_{th} afin d'accélérer le calcul des équations, les caractéristiques sont données par :

$$V_t = g(x)$$

$$I_{ds} = \frac{W}{L_{eff}} h(V_{gs} - \frac{1}{2} V_{ds}) \cdot (f(V_{ds}) - f(V_{db}) + V_{ds} \cdot V_{db})$$

où $g(x)$, $h(x)$ et $f(x)$ sont données par un tableau de valeurs expérimentales.

II.4/Modèle H. Shichman-D. Hodges

Ce modèle est utilisé particulièrement dans l'analyse assistée par ordinateur /34/ :

$$V_t = V_{t0} + K'((-V_{sd} + \Psi)^{\frac{1}{2}} - \Psi^{\frac{1}{2}}) \quad \text{si } V_{sd} > V_{ss}$$

$$V_t = V_{t0} + k'((-V_{ss} + \Psi)^{\frac{1}{2}} - \Psi^{\frac{1}{2}}) \quad \text{si } V_{sd} \leq V_{ss}$$

$$I_d = K(F_1 - F_2)(1 + \beta |V_{gs} - V_{gd}|)$$

où

$$F_1 = \begin{cases} 0 & \text{si } V_{gs} < V_t \\ (V_{gs} - V_t) & \text{si } V_{gs} \geq V_t \end{cases}$$

$$F_2 = \begin{cases} 0 & \text{si } V_{gd} \leq V_t \\ (V_{gd} - V_t) & \text{si } V_{gd} > V_t \end{cases}$$

$$K = \frac{\mu C_o W}{2 L} \quad ; \quad K' = \frac{2 \epsilon_s}{\epsilon_i} \left(\frac{N q d^2}{2 \epsilon_s} \right)^{\frac{1}{2}}$$

II.5/Modèle G. Merckel, J. Borel et N. Cupcea

C'est un modèle simplifié, établi pour la conception assistée par ordinateur /19/. C'est un modèle qui tient compte de la

mobilité entre gate-canal et drain-source. Pour la région avant la saturation ($V_{ds} < V_{dss}$) le courant I_{ds} est donné par :

$$I_{ds} = \frac{M^0}{1 + \theta \cdot V_{g'}} (V_{g'} - V_{ds}/2) \cdot V_{ds} \quad (15\%)$$

$$I_{ds} = \frac{M^0}{1 + \theta \cdot V_{g'}} (V_{g'} - (V_{ds}/2)(1 + \delta)) \quad (16\%)$$

avec δ le terme de correction et $V_{g'} = V_{gs} - V_t$.

Une autre correction faite sur l'équation du courant I_{ds} conduit à une précision de 5% .

III/Strategie du choix optimum des dimensions du pas d'intégration h

La dimension de h , pour chaque pas, peut être optimisée en choisissant la valeur la plus large possible pour laquelle l'erreur de troncature est limitée par une erreur permmissible maximale et pour laquelle l'algorithme reste stable, afin de diminuer le temps de simulation. Il existe deux procédés d'utilisation de h . La dimension de h peut être constante ou variable durant le processus d'intégration. Pour les grands systèmes, le premier procédé (h constante) est lent, par contre le second, est plus rapide, la valeur de h est calculée automatiquement en fonction des valeurs précédentes. Ce procédé est intéressant dans le cas où la réponse d'un circuit reste constante à partir d'un certain temps, car il permet de réduire le nombre d'itérations. Si l'intervalle d'intégration est $(0, T)$ et h le pas d'intégration constant, alors le nombre de points à calculer est :

$$N = \frac{T}{h}$$

où T représente la durée de la réponse.

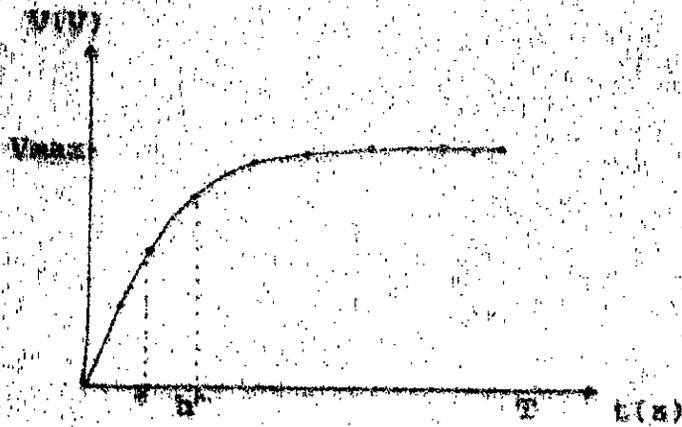


Fig. 4.1a : pas h constant



Fig. 4.1b : pas h variable

Mais si h est variable sur le segment (0,T) et la réponse reste constante dans l'intervalle (tc,T), le nombre de points est :

$$N' = \frac{tc}{h_{moy}} \quad \text{avec } tc \ll (T-tc)$$

h_{moy} : la valeur moyenne de h (estimée)

d'où $N' < N$

III.1/ Calcul automatique du pas d'intégration h pour l'algorithme de Backward-Euler

En général, la nouvelle valeur du pas (h') est donné par :

$$h' = \alpha h$$

avec
$$\alpha = \frac{1}{1.2} \sqrt{\frac{2 \epsilon_{max}}{(1/2) \nabla^2(z_n)}}$$

où
$$\nabla^2(z_n) = h^2 \nabla_n^{(2)}$$

$$\nabla_n^{(2)} = \left. \frac{\partial^2 v}{\partial t^2} \right|_{t=t_n}$$

Cette technique n'est pas pratique, car elle nécessite la dérivée seconde. Donc une autre stratégie plus simple et plus

pratique sera développée .

III.2/Technique de calcul automatique du pas d'intégration h

Dans cette méthode, nous considérons la réponse donnée par la Fig.4.2, nous subdivisons l'intervalle (0, Vmax) de façon équitable , en M troçons, chaque segment a une dimension de Vmax/M. Nous supposons que la prochaine variation de la tension est Vmax/M, (la variation se fait à pente constante). En connaissant la valeur de h à l'instant tn (h=hn), nous avons :

$$\frac{V_{\max}/M}{h_{n+1}} = \frac{\Delta V_n}{h_n}$$

$$\Delta V_n = V_{n+1} - V_n = V(t_{n+1}) - V(t_n)$$

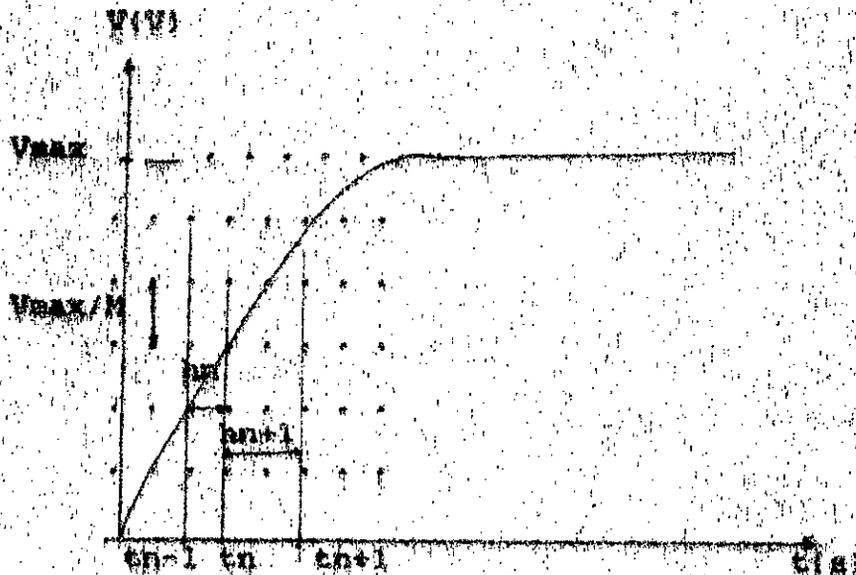


Fig. 4.2 Détermination du pas h_{n+1}

$$h_{n+1} = \frac{V_{\max}/M}{\Delta V_n} h_n$$

V_{\max}/M est fixe , h_{n+1} est proportionnelle à h_n et inversement proportionnelle à ΔV_n .

IV/Organisation des données d'entrées-matrices et vecteurs topologiques

Dans la 1^{ère} phase, la description topologique nécessite la numérotation des noeuds et des transistors du circuit intégré à analyser (voir Fig 4.3).

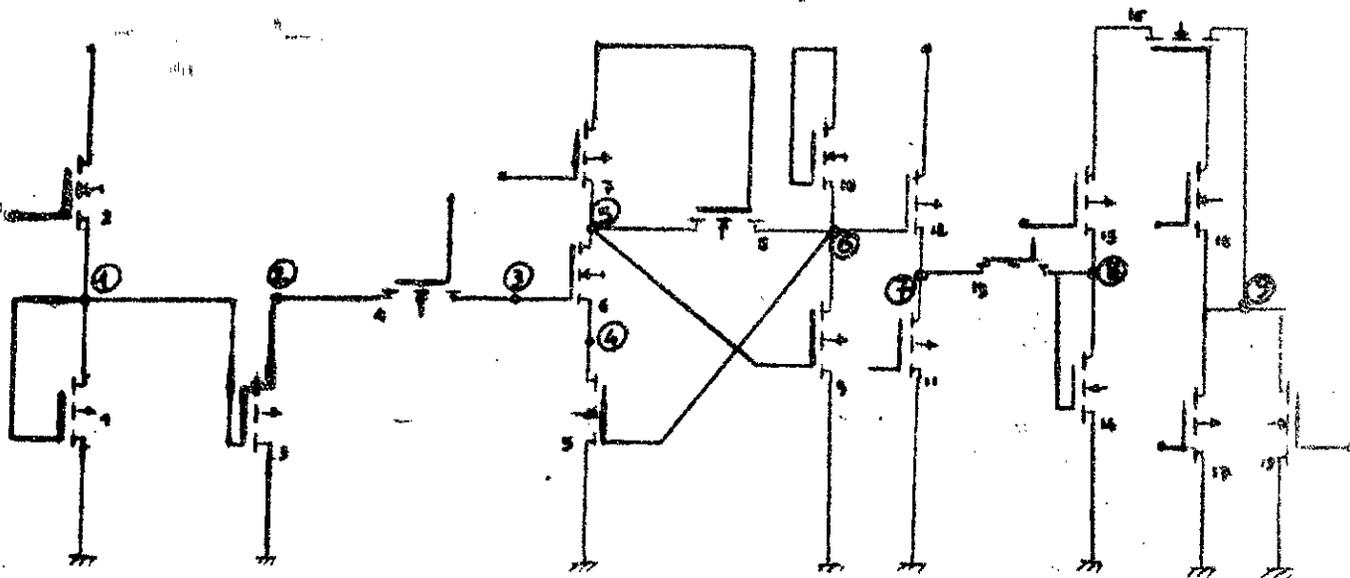


Fig. 4.3 Topologie d'un circuit intégré quelconque

Les capacités parasites sont représentées par des capacités linéaires reliées entre le noeud et la masse. Dans le chapitre précédent, nous avons présenté 7 types de matrices topologiques S,D,G,ND,LS,LD et LG et les vecteurs HS, HD et HG nécessaires à la formulation des équations d'analyse.

IV.1/Matrice noeud ND

Les coefficients de la matrice ND prennent les valeurs: +1,-1 ou 0 selon que le courant drain-source circule du drain vers la source, de la source vers le drain ou il est nul. Pour le CI de la Figure 4.3, nous avons :

IV.3/ Matrices liaisons

Le coefficient $LS(i,p,j)$ (respectivement $LD(i,p,j)$) prend la valeur 1 ou 0 si le noeud i est relié (ou non) au noeud p par l'intermédiaire de la source (respectivement le drain). Vu que ces matrices sont tridimensionnelles, nous donnons uniquement la composante $LS(1,2,j)$ pour $j = 1,2,\dots,19$

$$LS(1,2,19) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad LD(1,2,19) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 0 \end{bmatrix} \quad LG(1,2,19) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 0 \end{bmatrix}$$

Le circuit intégré de 9 noeuds et de 19 transistors nécessite 4674 données d'entrées pour la description topologique et 103 données des transistors MOS (pour un modèle de 7 paramètres). Pour palier à cet inconvénient, nous avons établi des logiciels permettant la conversion topologie/matrices (Fig.4.4).



Fig.4.4

Ces logiciels de conversion minimisent le nombre de données d'entrées. C'est un langage simple sous forme de questionnaire (Fig.4.5), facilitant l'utilisation du FATMOS. Nous présentons les organigrammes de génération des matrices et vecteurs topologiques sous la forme la plus simple pour le cas des matrices S et LG et le vecteur HS (voir Fig.4.6. 4.7)

V/Algorithme-Organigramme d'analyse des CI à MOS

La structure de n'importe quel programme d'analyse comprend une bibliothèque de modèle des éléments actifs et des éléments non linéaires, un bloc de description topologique, formulation des équations, résolution des équations et un test de convergence des algorithmes développés. Ceci est illustré par l'organigramme général de la Fig. 4.8. Les matrices D et G sont créées de la même façon que la matrice S. Les matrices tridimensionnelles LD et LS ont le même organigramme que celui de la Fig.4.7. Même chose pour HD et HG .

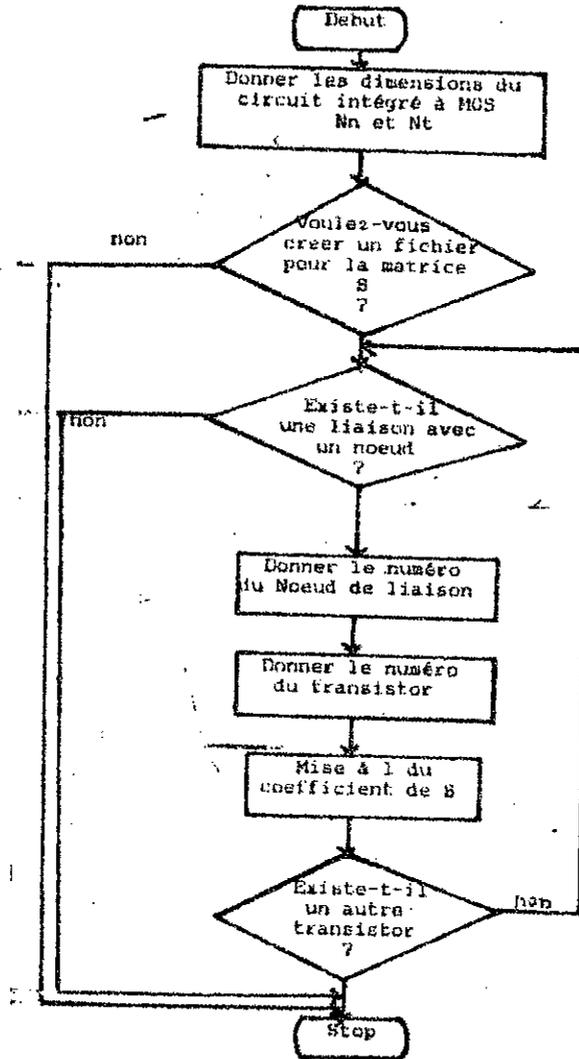


Fig.4.5 Organigramme de génération de la matrice S

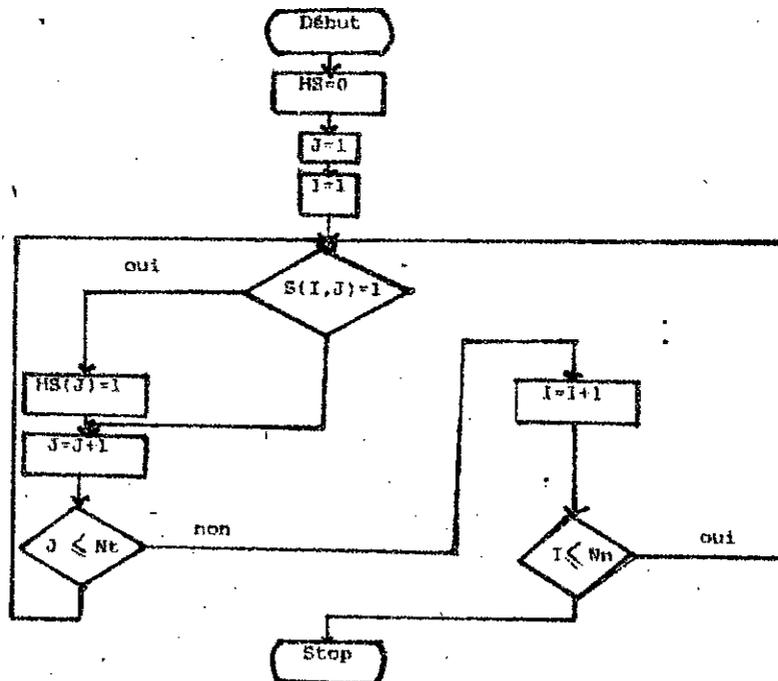


Fig.4.6 Organigramme de génération du vecteur HS

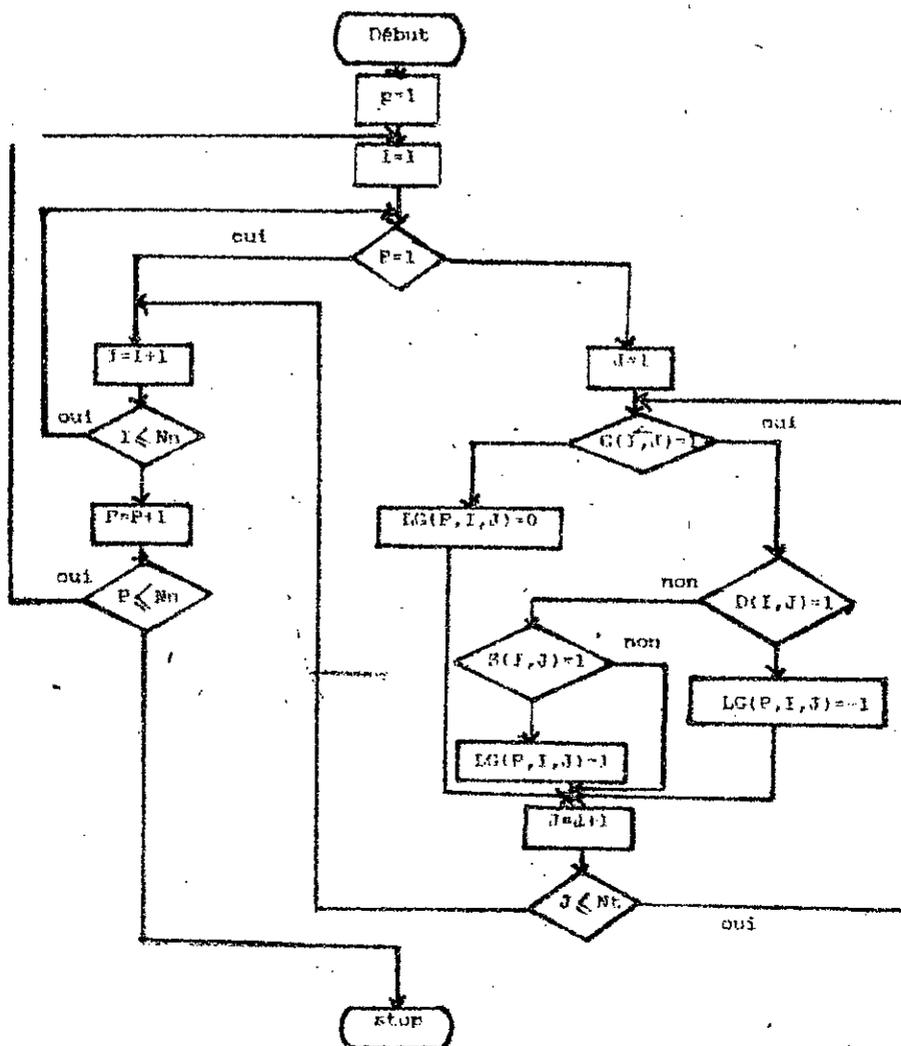


Fig.4.7 Organigramme de génération de la matrice LG

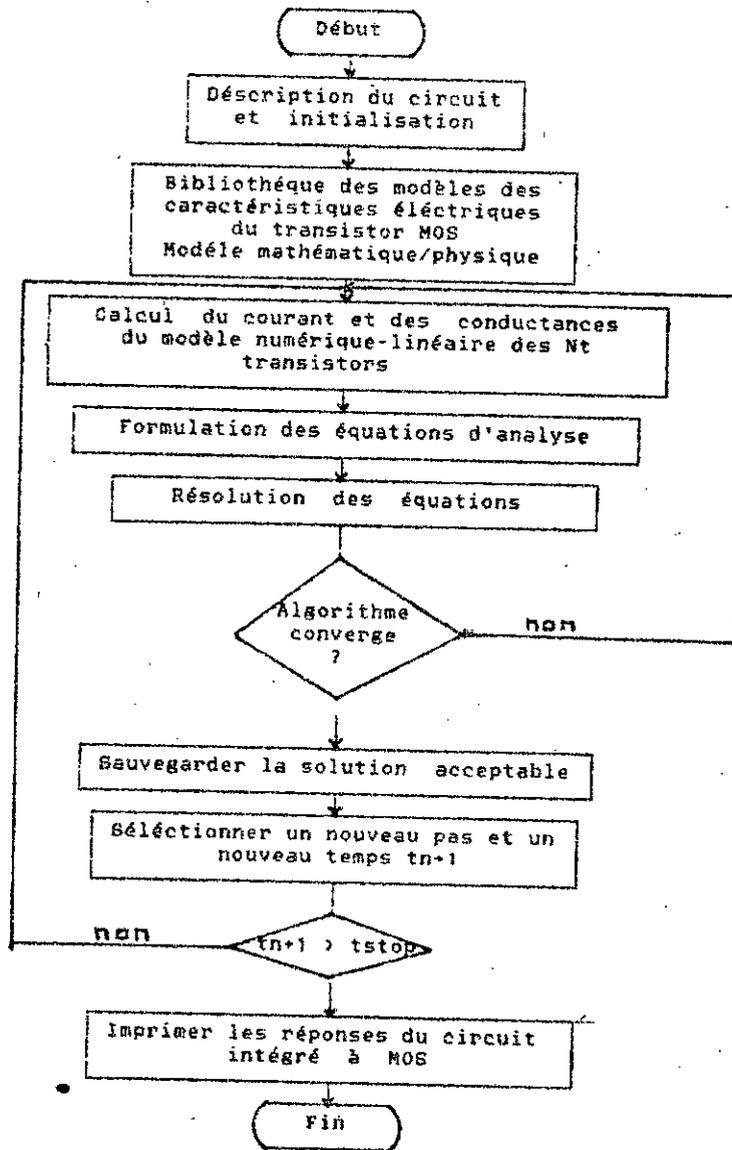


Fig.4.8 Organigramme général du programme PATMOS

VI/ Applications du programme PATMOS

L'utilisation du PATMOS est simple, elle nécessite quelques données d'entrées pour la description topologique et les valeurs des paramètres physiques du modèle choisi. Comme applications

nous considérons quelques circuits intégrés, les réponses obtenues par PATMOS (voir figures 4.9-4.13) sont conformes avec celles données en théorie /31/, /32/.

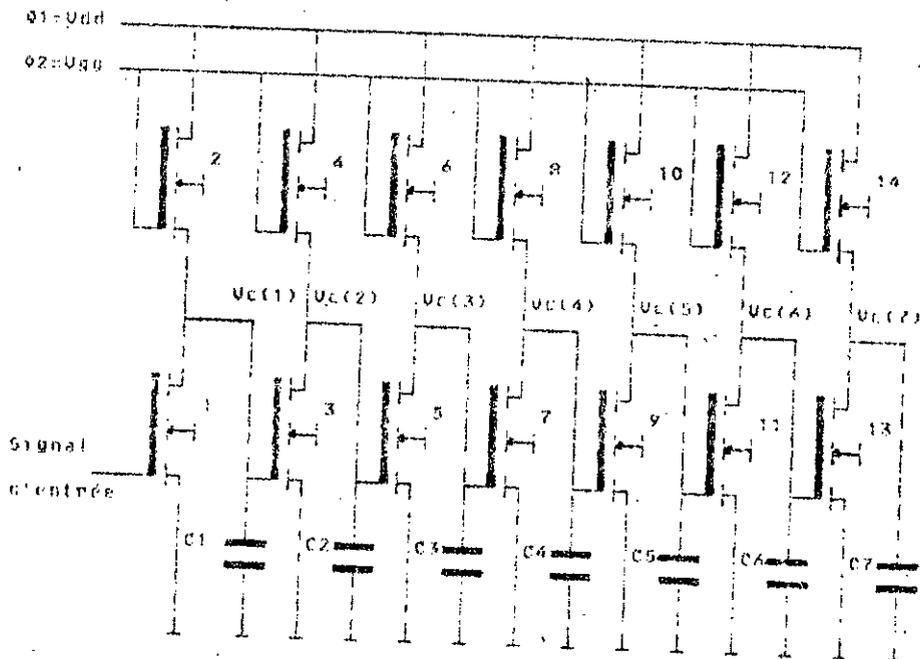


Fig.4.9.a : Topologie de l'inverseur à sept étages

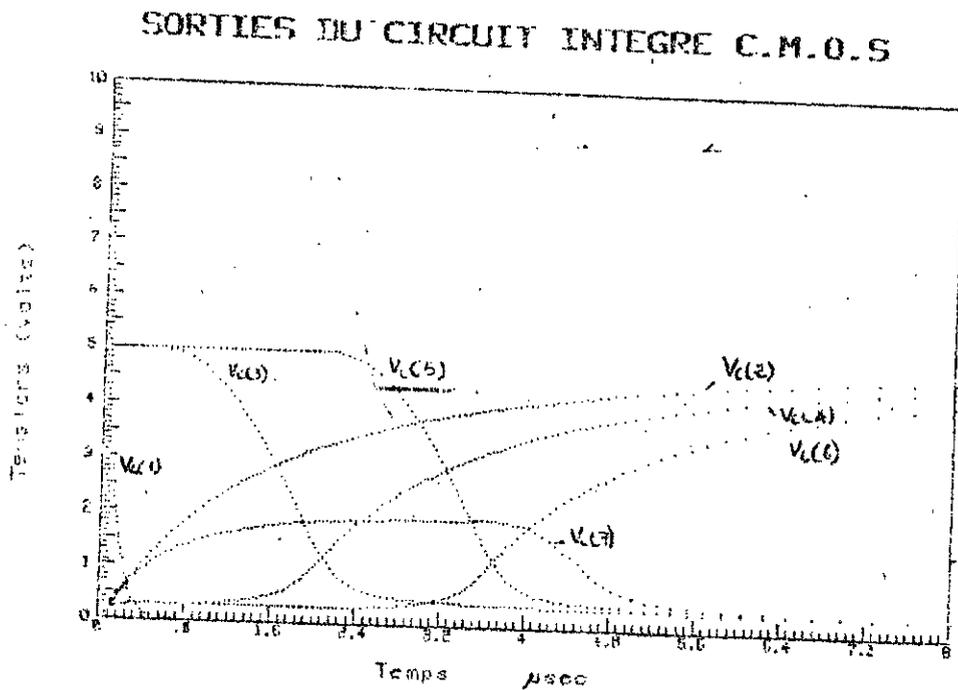


Fig.4.9.b : Les réponses de l'inverseur à sept étages données par PATMOS

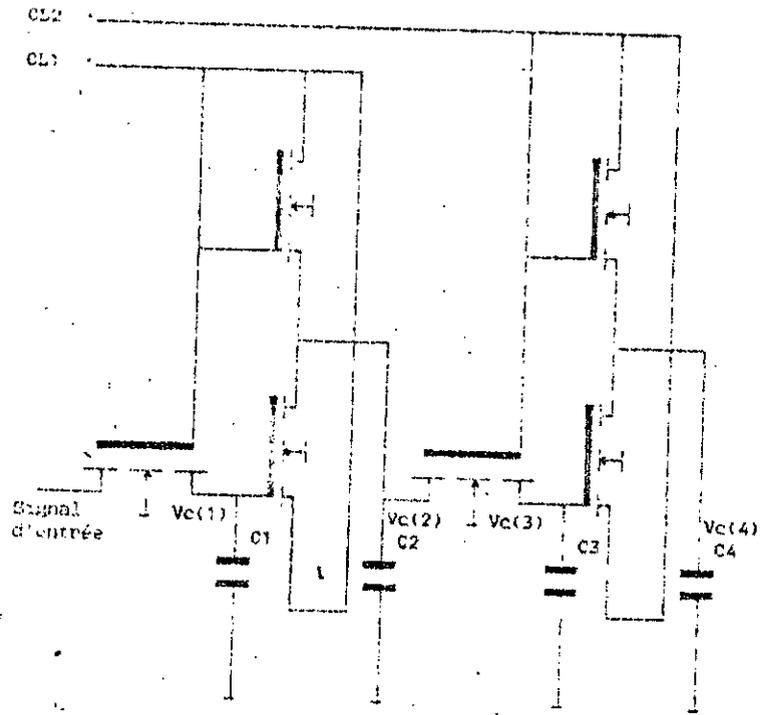


Fig.4.11.a Topologie de la bascule D biphasé sans rapport

SORTIES DU CIRCUIT INTEGRE C.M.O.S

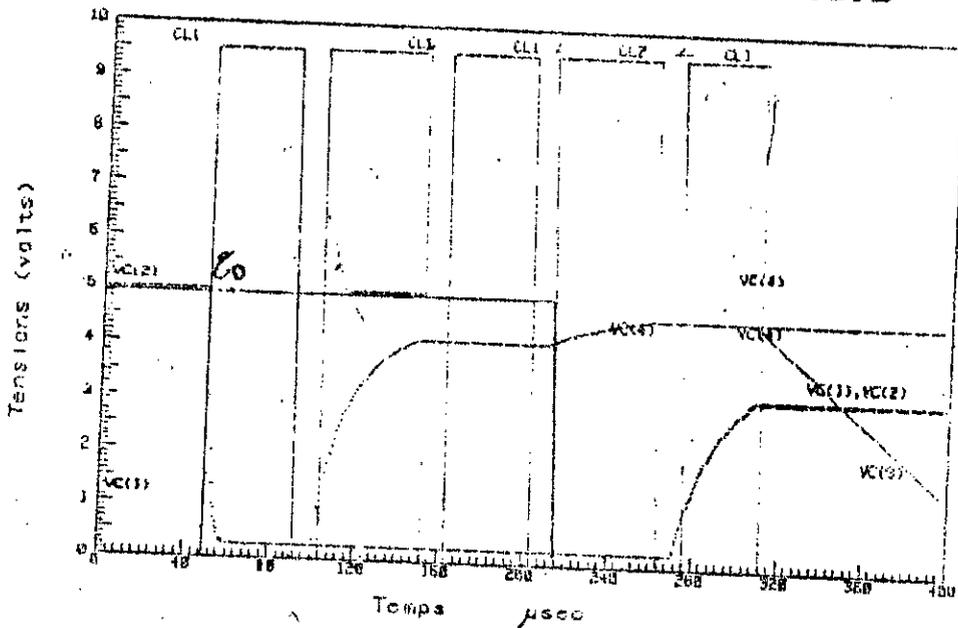


Fig.4.11.b Réponses par PATMOS

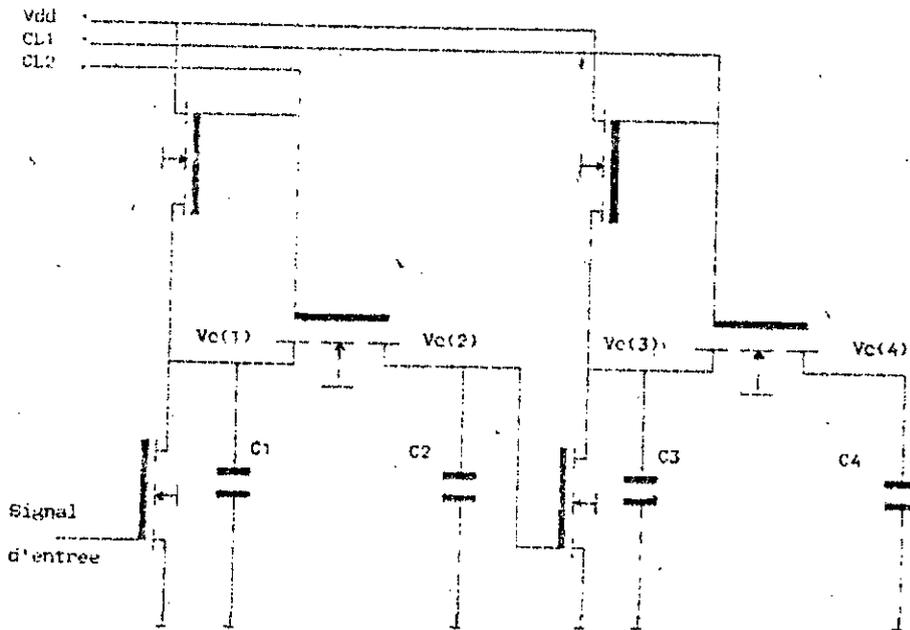


Fig.4.12.a Topologie de la bascule D biphasé avec rapport

SORTIES DE CIRCUIT INTEGRE CMOS

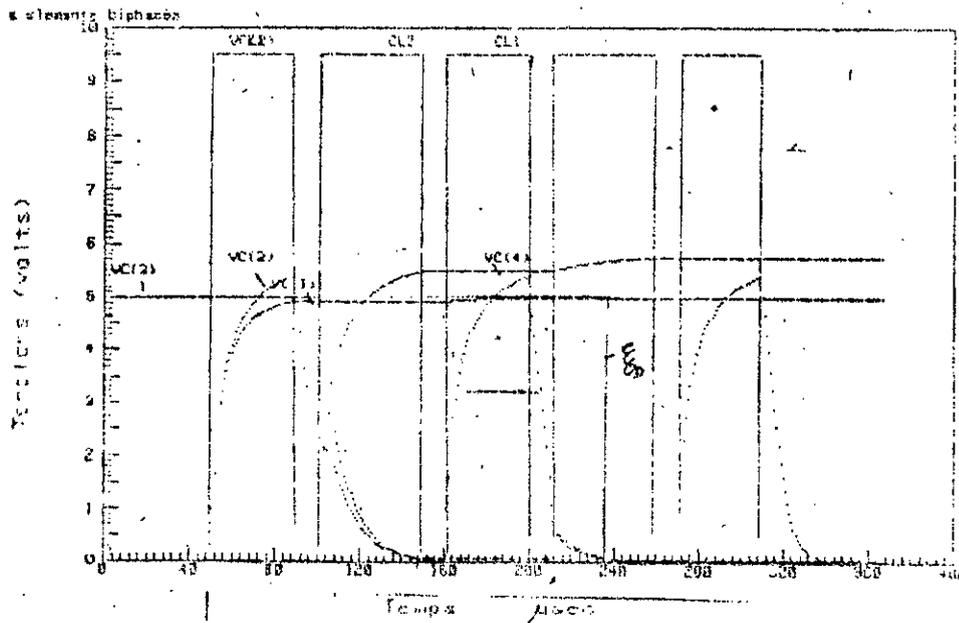


Fig.4.12.b Réponses par PATMOS

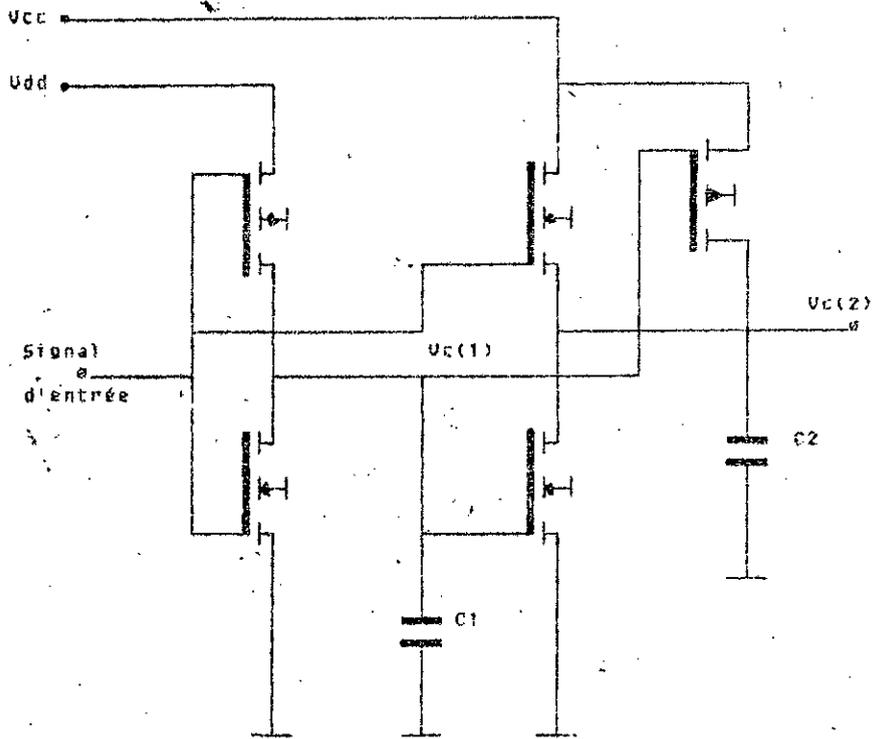


Fig.4.13.a Topologie du buffer CD 4010 (CMOS)

SOURCE IS CIRCUIT INTEGRE C.F.9.8

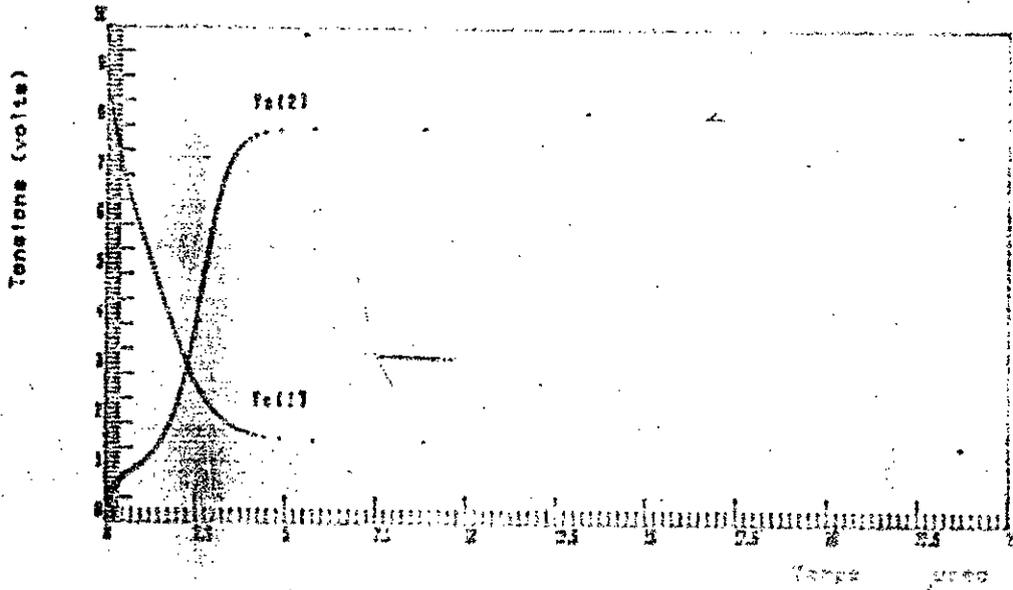


Fig.4.13.b Réponses par PATMOS

VII/ Vérification expérimentale du programme PATMOS

Pour le test expérimental du PATMOS, nous considérons le cas de l'amplificateur de lecture pour cellules mémoires, composé de 5 transistors et de 3 noeuds. Nous étudions ce circuit intégré en régime dynamique avec un signal d'horloge U . Pour le noeud N°3, nous supposons que la capacité $C3=0$. Au cours de l'impulsion d'horloge les capacités $C1$ et $C2$ se chargent, au repos ces dernières se déchargent mais avec des vitesses différentes. Les réponses par simulation (PATMOS) Fig.4.14 sont données avec une précision de 12% par rapport aux réponses expérimentales Fig.4.15. Cette précision représente l'erreur relative maximale des temps de commutation, pour le cas de l'amplificateur de lecture nous avons considéré le temps de descente t_d .

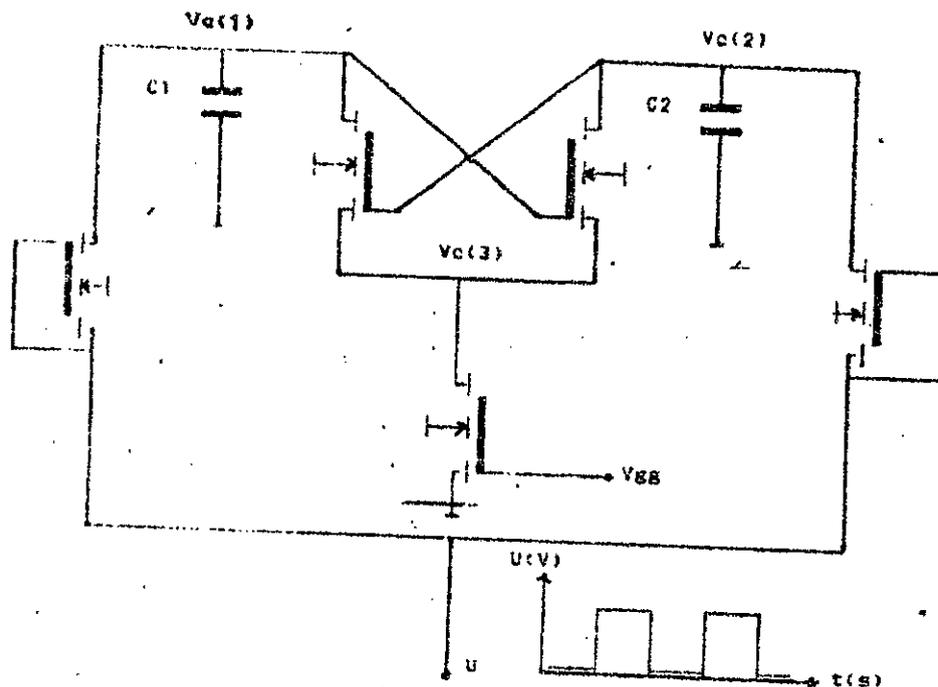


Fig.4.14.a Topologie de l'amplificateur de lecture

SORTIES DU CIRCUIT INTEGRE C.M.O.S

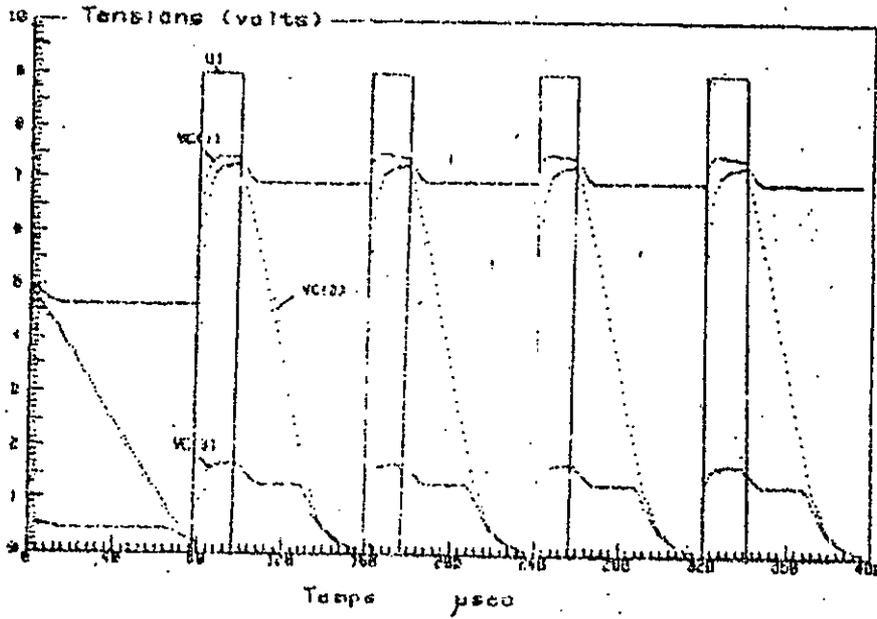


Fig.4.14.b Réponses par simulation (PATMOS)

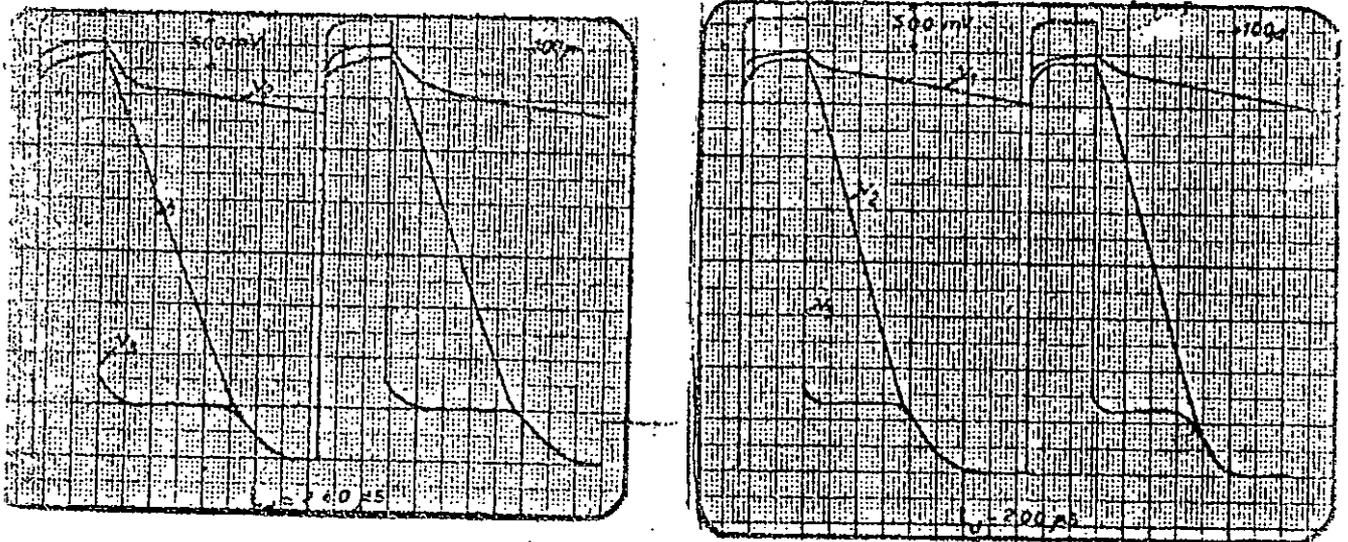
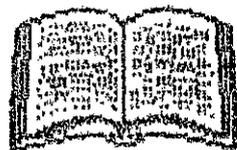


Fig.4.15 Réponses expérimentales

I

Conclusion
Generale



CONCLUSION

Les programmes d'analyse pour la conception des circuits intégrés à MOS (metal-oxide-semiconducteur) occupent une place de choix dans le développement de la microélectronique. Leurs avantages indiscutables sont bien connus : le temps, l'économie et la facilité de travail. Dans l'élaboration de ce programme les problèmes les plus importants sont incontestablement le temps d'exécution et l'espace mémoire. Tenant compte de ces paramètres, nous avons été guidé par trois critères principaux :

- La convergence
- La stabilité
- La simplicité

Dans notre travail, l'étude sur les modèles mathématiques et physiques du transistor MOS, nous a permis d'établir une classification et d'en déduire que quelque soit les dimensions 1D, 2D ou 3D, le transistor MOS est caractérisé par une fonction non linéaire. Le programme élaboré est basé sur un algorithme à niveau électrique-universel. Ceci nous a amené à réaliser un modèle numérique-linéaire, ainsi que sa représentation électrique pour le cas du transistor simple (sans effet de capacité) et le cas du transistor MOS avec capacités entre électrodes. Ce modèle a été développé en utilisant l'algorithme de Newton qui présente une convergence quadratique et une meilleure stabilité. Il est indépendant du modèle physique ou mathématique.

L'application des méthodes de formulation existantes (variable d'état, théorie des graphes) pour l'analyse des circuits intégrés à MOS, nous a permis de déceler leurs limites.

- Non obtention de la forme standard pour les variables d'état
- Espace mémoire et temps de calcul très élevés pour la théorie des graphes

De ce fait nous avons réalisé un algorithme universel de formulation automatique d'un système d'équations d'analyse, grâce à l'élaboration d'un modèle numérique-linéaire du transistor MOS et du modèle numérique-linéaire de la capacité parasite. Nous avons développé ce dernier à partir de l'algorithme d'intégration numérique : Backward-Euler, simple et présente la plus grande région de stabilité.

Le programme d'analyse transitoire des circuits intégrés à MOS : P.A.T.M.O.S. basé sur l'algorithme élaboré est un programme de première étape, permet de simuler n'importe quelle configuration topologique d'une part, permet de générer le chronogramme des CI digitaux d'autre part. Son application peut être élargie au CI à MOS analogiques, car il est à niveau électrique. Il peut utiliser n'importe quel modèle mathématique ou physique.

Pour son optimisation en temps de calcul, nous avons établi une technique pratique de calcul automatique du pas d'intégration et une meilleure organisation des données d'entrées afin de réduire leur nombre et l'espace mémoire.

Les caractéristiques et limitations du PATMOS sont :

-Grandeurs de mesure sont : nombre de noeuds (N_n) et le nombre de transistors (N_t)

-Nombre de données d'entrées : $N_p.N_t + N_n$ (N_p est le nombre de paramètres par modèle)

-Espace mémoire : $3.N_n.N_t.(N_n+1)$

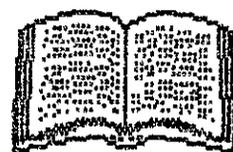
-Temps de calcul par itération : $3.(N_n \cdot N_t).t$
op

-Convergence est assurée après deux ou trois itérations en moyenne

Les données de sorties sont des courbes de réponses du CI, obtenues en même temps et qui sont en nombre de N_n . L'échelle temporelle est déterminées automatiquement par PATMOS.

A l'aide des tests expérimentaux, nous avons évalué pour le cas de l'amplificateur de lecture, la précision relative du PATMOS qui est de l'ordre 12% en utilisant le modèle de Sah-Pao. Sa précision dépend en grande partie du modèle mathématique ou physique utilisé. Ce programme peut être amélioré si nous considérons le transistor avec effet des capacités entre électrodes. Ceci constitue l'objet d'un autre thème de recherche qui donne naissance à une autre version du PATMOS.

Annexe



ANNEXE

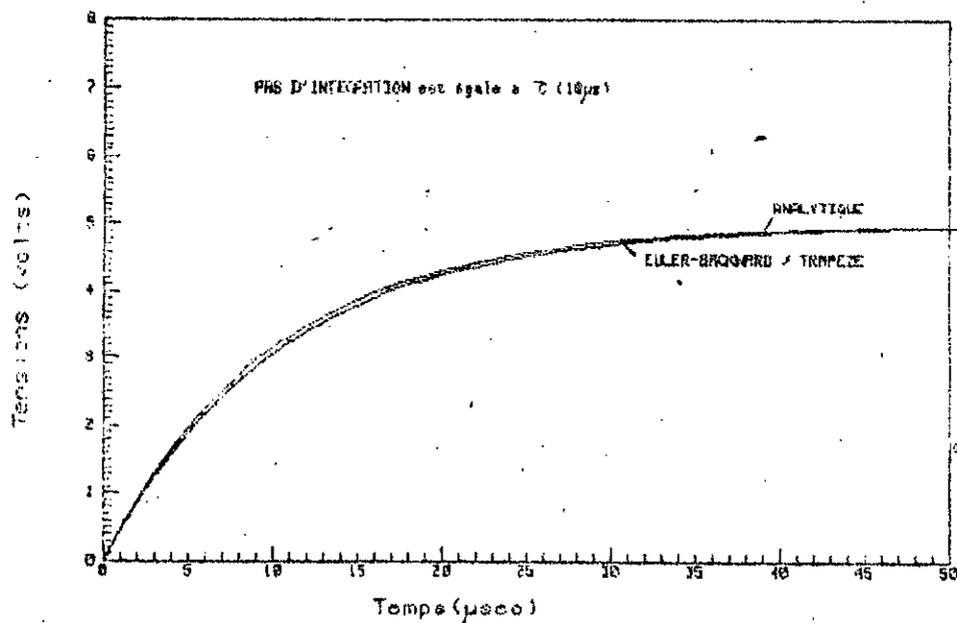


Fig.1 Réponse V_c en fonction du temps pour $h = c/10$

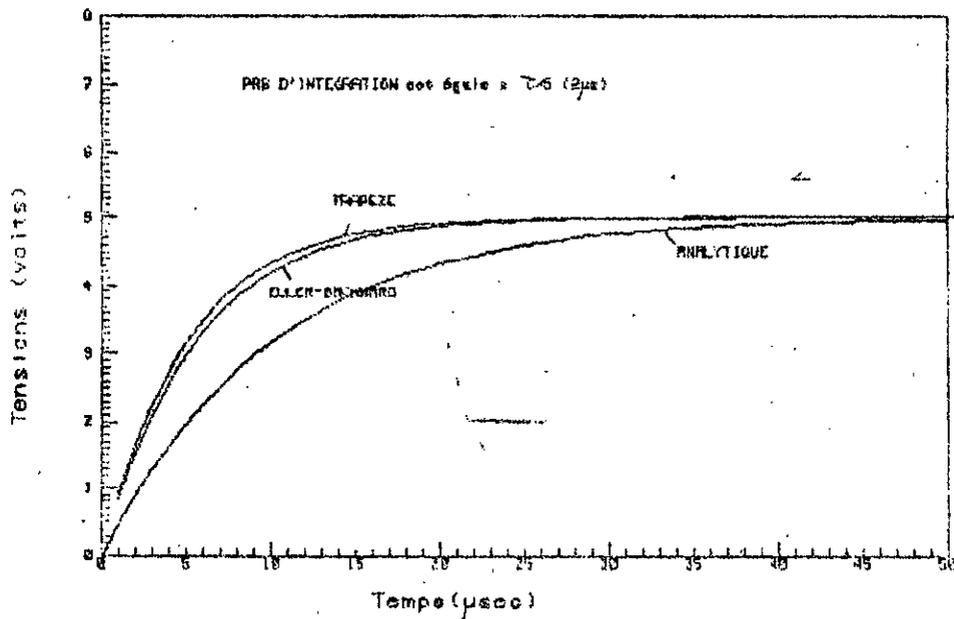


Fig.2 Réponse V_c en fonction du temps pour $h = c/5$

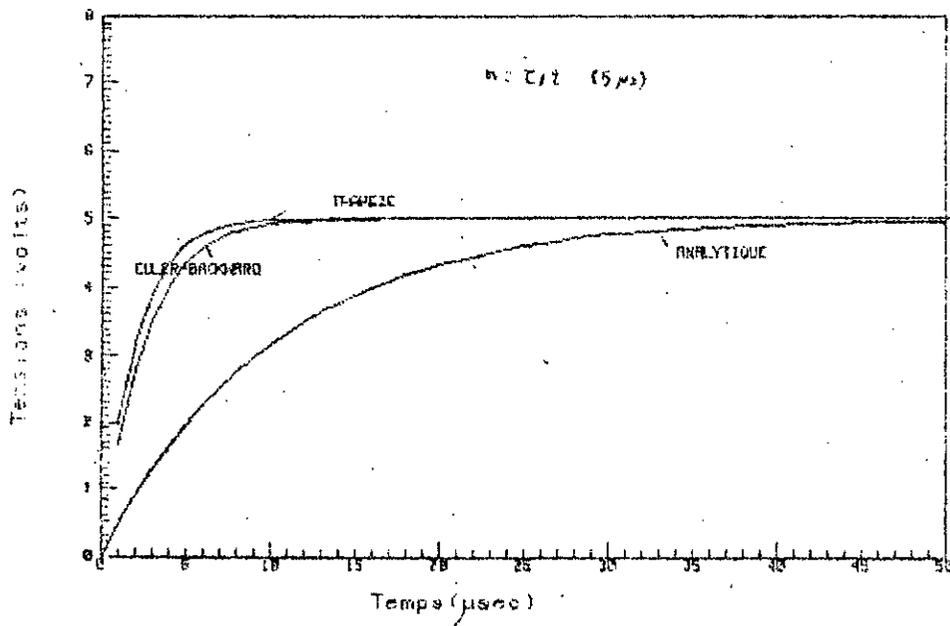


Fig.3 Réponse V_c en fonction du temps pour $h = c/2$

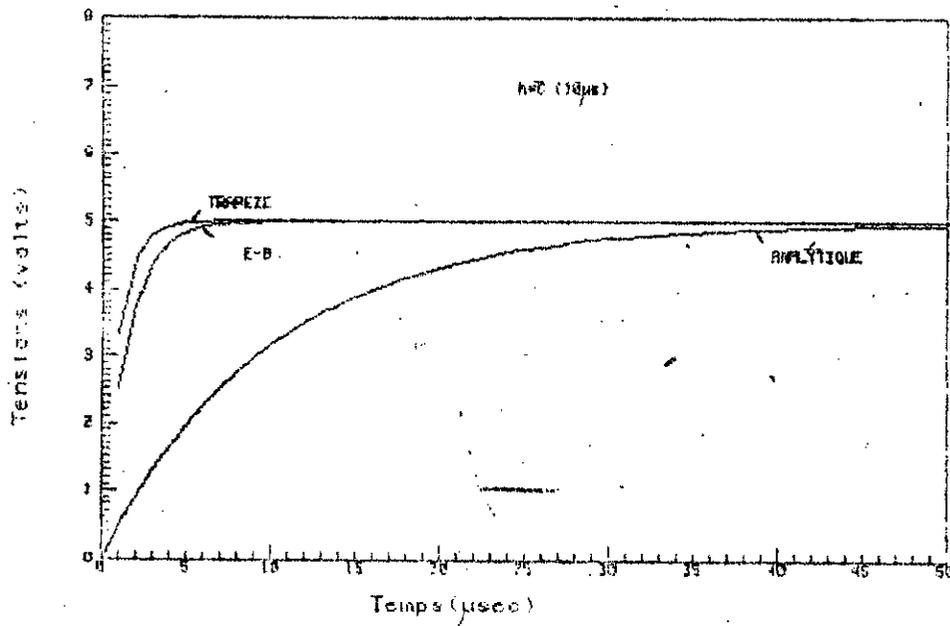


Fig.4 Réponse V_c en fonction du temps pour $h = c$

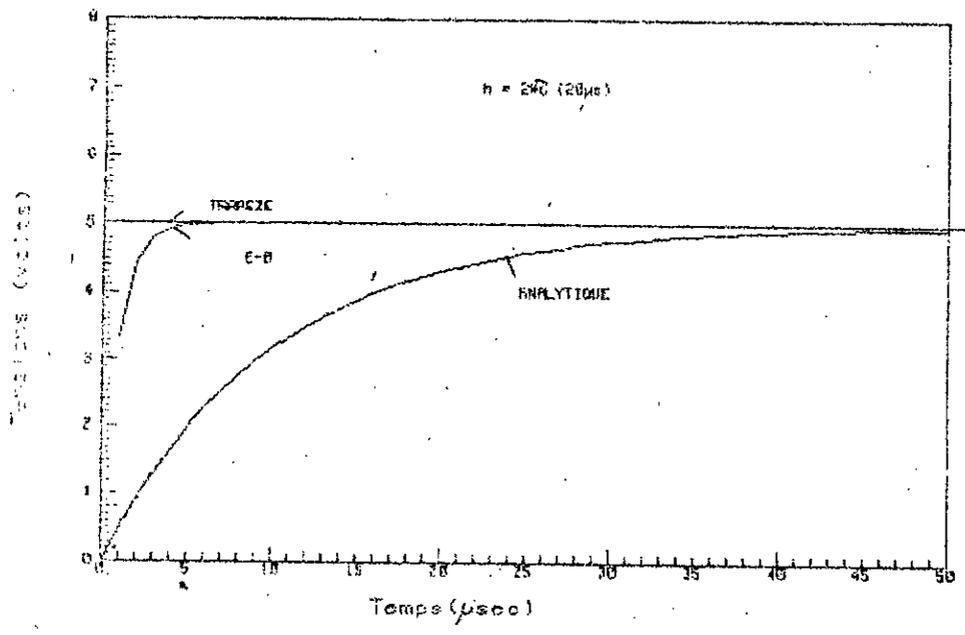


Fig.5 Réponse V_c en fonction du temps pour $h = 2.c$

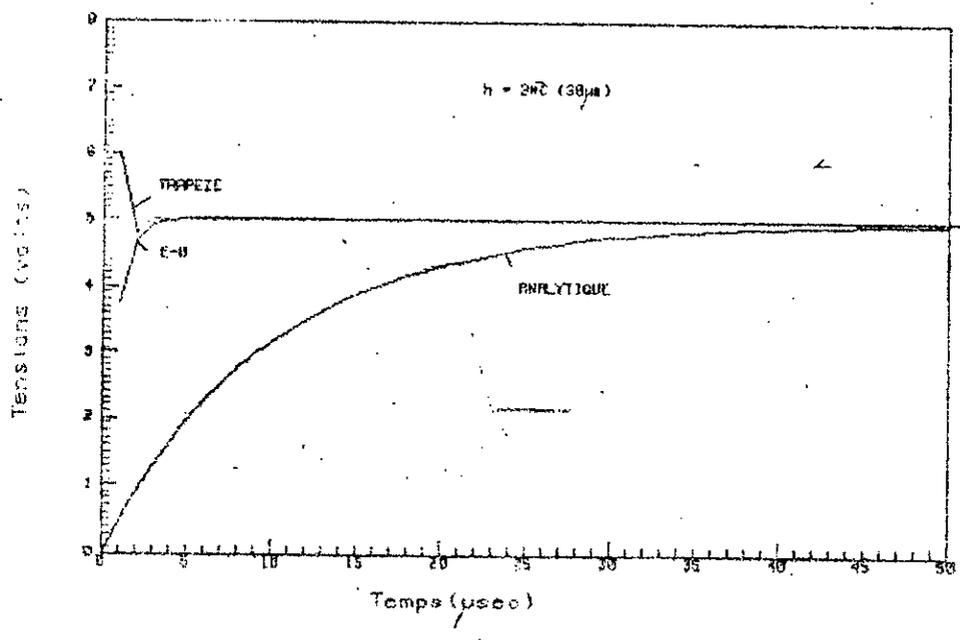


Fig.6 Réponse V_c en fonction du temps pour $h = 3.c$

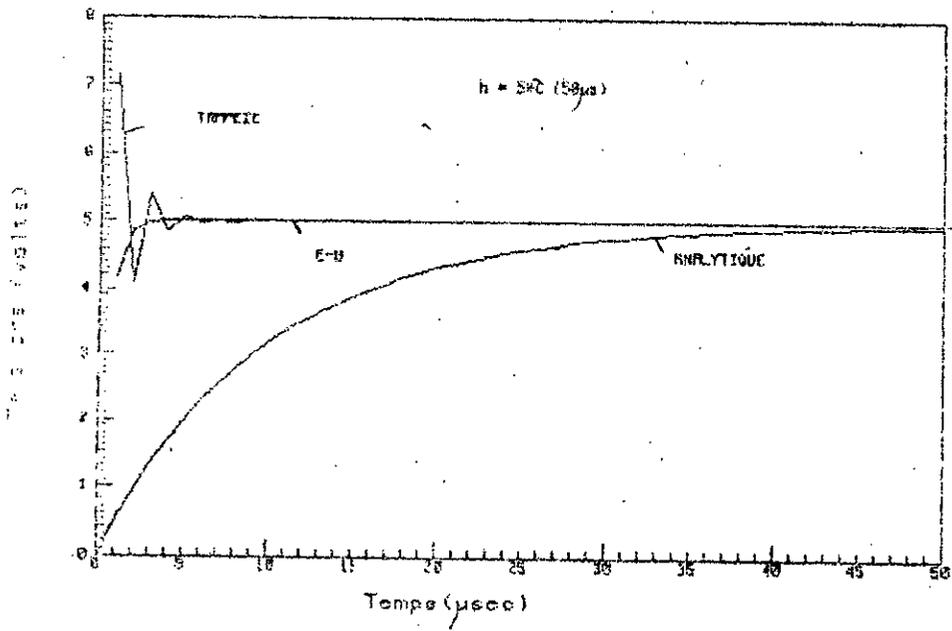


Fig.7 Réponse Vc en fonction du temps pour $h = 5.c$

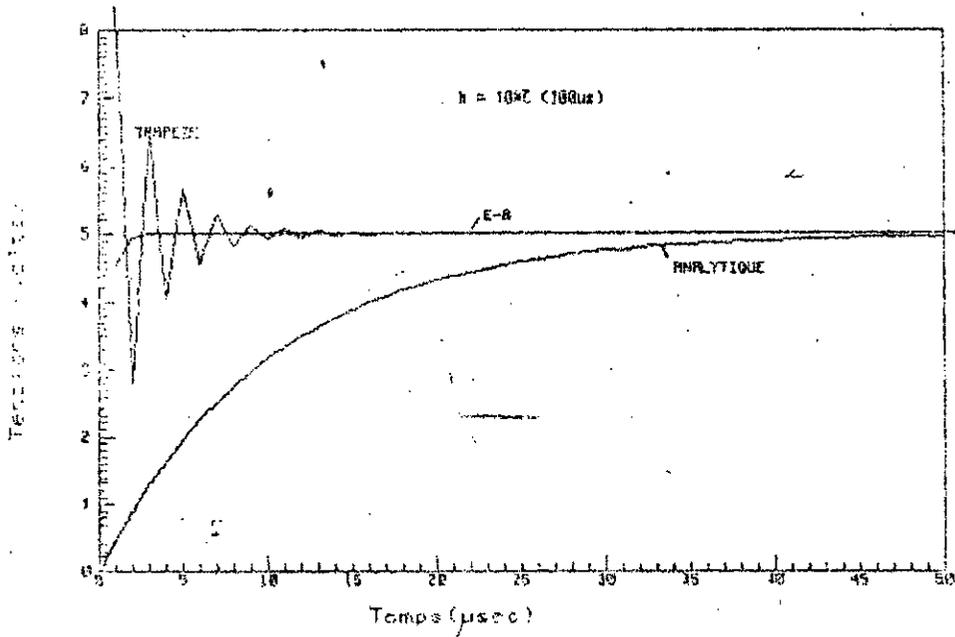


Fig.8 Réponse Vc en fonction du temps pour $h = 10.c$

B I B L I O G R A P H I E

- 1/ W. NAGEL , " SPICE2 , A Computer Program to Simulate Semiconductor Circuits , " Univ. of California, Berkeley , Memo No ERL-M520, May 1975 .
- 2/ KANJI HIRABAYASHI and JOE WATANABE , " MATIS-Macromodel Timing Simulator for Large Scale Integrated MOS Circuits " , 3rd USA-JAPAN Computer Conference , pp 457-461, 1978.
- 3/ JERRY R. PATTERSON , " SCEPTRE : Automatic Digital Computer Program for Network Transient Analysis " , Proc. IEEE COMPUTER , pp 78-82, 1967 .
- 4/ M. JACOLIN , " Les programmes "IMAG I" et "IMAG II" , L'ONDE ELECTRIQUE , Vol. 49, fasc.1 pp 20-27 , janvier 1969 .
- 5/ B.R. CHAWIA , A. K. GUMMEL and P. KOZAK , " MOTIS-an MOS Timing Simulation " , IEEE Trans. Circuits and Systems , Vol. CAS-22 , pp901-909 , Dec 1975 .
- 6/ S. P. FAN , M. Y. HSUEH , A. R. NEWTON and D. O. PEDERSON , " MOTIS-C : A New Circuit Simulator for MOS LSI Circuits " , in Proc. IEEE Int. Symp Circuits Syst. Apr. 1977
- 7/ NORIO TANABE , HIROSHI NAKAMURA and KENJI KAWAKITA , " MOSTAP: An MOS Circuit Simulator for LSI Circuits " , In Proc. IEEE Int. Symp. Circuits and Systems , pp 1035-1038, april 1980 ,
- 8/ RANDAL E. BRYANT , " An Algorithm for MOS Logic Simulation " , LAMBDA Fourth Quarter , pp46-53, 1980 .
- 9/ M.Y. HSUEH, A. R. NEWTON and D. O. PEDERSON, " The Development of Macromodels for MOS Timing Simulators " ,
- 10/ E. LELARASMEE and A. I. SANGIOVANNI-VINCENTELLI , " Some New Results On Wave Form Relaxation Algorithms for The Simulation of integrated Circuits " , In Proc. , IEEE Int., Large Scale Syst Symp. pp 371-376, 1982.
- 11/ A. R. NEWTON , A. I. SANGIVANNI-VINCENTELLI , " Relaxation Based Electrical Simulation " , IEEE Trans. ON Electron Devices , Vol. ED-30, N°9, pp1184-1207, Sept. 83
- 12/ WALTER L. ENGL , HEINZ K. DIRKS and BERND MEINERZHAGEN , " Device Modeling " , Proceedings of the IEEE , Vol. 71 , N°1 , pp 10-32, January 1983 .

- 13/ C.T. SAH " Characteristics of the Metal-Oxide-Semiconductor transistors " IEEE Trans. On Electron Devices pp 324-345, July 1964
- 14/ H.C. PAO and C.T. SAH " Effects of diffusion current on characteristics of Metal-Oxide (Insulator)-Semiconductor transistors " Solid-State Electronics Pergamon Press, pp 927-937, 1966 Vol.9,
- 15/ D. FROHMAN-BENTCHKOWSKY and A.S GROVE " Conductance of MOS transistors in saturation " IEEE Trans. On Electron Devices ,Vol. ED-16 N°1 pp 108-113, January 1969 .
- 16/ D. FROHMAN-BENTCHKOWSKY and LESLIE VADASZ , "Computer-Aided Design and characterisation of digital MOS Integrated Circuits " IEEE Journal of Solid-State Circuits Vol. SC-4 , N°2 , pp 57-64 , April 1969 .
- 17/ B. HOENEISEN AND C.A MEAD, " Current-Voltage characteristics of small size MOS transistors " , IEEE Trans. Electron Devices pp382-383, DE-19 , 1972 .
- 18/ G. MERCKEL , J. BOREL and N. Z. CUPCEA , " An Accurate Large-Signal MOS transistor Model for use in Computer-Aided Design " , IEEE Trans. On Electron Devices , Vol. ED-19 , N°5 , pp681-690, May 1972 .
- 19/ G. MERCKEL , " A Simple Model of the Threshold Voltage of Short and Narrow Channel MOSFETs " , Solid-State Electronics Vol. 23 , pp1207-1213, Pergamon Press Ltd. , 1980 .
- 20/ MARVIN H. WHITE, F. Van de Wiele and J. Lambot, "High-accuracy MOS Models for computer-aided design " , IEEE Trans. Electron Devices, Vol. ED-27, N°5 , pp 899-906 , 1980.
- 21/ D.A. CALAHAN, " Computer Aided Network Design " , Mc.HILL, 1975 .
- 22/ LEON CHUA , " Computer Aided Analysis Network " , Mc.HILL, 1978 .
- 23/ E. S. KUH and R. A. ROHRER , " The State-Variable Approach to Network Analysis " , Proceedings of the IEEE , pp 672-686 .
- 24/ M. BOUMAHATH, A. GOURDIN , " Méthodes Numériques - Théorie et Applications- " , OPU , 1983 .
- 25/ F. G. GUSTAVSON , M. LINIGER , and R. WILLOURHBY , " Symbolic Generation of an optimal Crout Algorithm for Sparse Systems of Linear Equations " , Journal of

The Association for Computing Machinery
Vol. 17, N°1, pp87-109, January 1970.

- 26/ GENE H-GOLUB, GERARD A-MEURANT, "Resolution numerique des grands systemes lineaires". Edition EYROLLES.
- 27/ M. R. HESTENES and E. STIEFEL, "Methode of conjugate gradients for solving linear systems", Journal of Research of the National, Bureau of Standards Vol.49, N°6, pp409-437, Dec.1952.
- 28/ DAVID S.KERSHAW, "The Incomplete Cholesky-Conjugate Gradient Method for the iterative solution of systems of linear Equations", Journal of Computational Physics 26, pp43-65, 1978.
- 29/ J. K. REID, "On the Method of Conjugate Gradients for the Solution of large Sparse Systems of Linear Equations", SIAM J. Number Anal Vol.12, N°4, pp 231-254, September 1975.
- 30/ P.M. DEW, K.R. JAMES, "Introduction to numerical computation in Pascal", (LONDON); Mac Millan, 1983.
- 31/ A ALEXENKO, I. CHAGOURINE, "Technique de Synthese des Circuits Integres", Editions MIR MOSCOU.
- 32/ LILEM, "Circuits Integres MOS et CMOS - Principe et applications", Editions Radio.
- 33/ J. MAVOR, "Conception et Technologie des Circuits Integres A Transistors MOS".
- 34/HAROLD SHICNMAN AND DAVID A. HODGES, "Modeling and Simulation of Insulated-Gate, field-effect transistor switching circuits", IEEE Journal of Solid-State Circuits, vol.SC-3, N°3, pp285-289 September 1968.
- 35/PARKER, S.R. AND V.T BARMES, "Existence of numerical solution and the order of linear circuits with depend sources", IEEE Trans. Circuit Theory, Vol. CT-18, pp368-377, May 1971.
- 36/PURSLow, E. J. "Solvability and Analysis of linear Active Networks by use of the state equations", IEEE Trans. Circuit Theory, Vol. CT-17 pp 469-475, Nov 1970.