

Analyse temporelle des circuits intégrés digitaux CMOS,
pour les technologies profondément submicroniques

Grégoire AVOT

17 mars 2003

Résumé

Dans le flot de conception des systèmes intégrés microélectroniques digitaux à très grande densité d'intégration, la phase de vérification temporelle devient une étape de validation incontournable. Cette vérification est réalisée grâce à deux grandes familles de logiciels : les analyseurs temporels et les simulateurs électriques. Les premiers sont moins précis que les seconds, mais permettent l'analyse de circuits de plusieurs millions de transistors. Ils apportent en efficacité ce qu'ils perdent en précision.

Avec les technologies profondément submicroniques, des phénomènes électriques auparavant négligeables deviennent aujourd'hui significatifs, voire dominants. Ces phénomènes sont les résistances d'interconnexion et les capacités de couplage. Ce deuxième phénomène rend les simulateurs électriques définitivement inutilisables car ils sont conçus pour donner des résultats exacts à des *stimuli* d'entrée. Or, le temps de propagation de chaque porte dépend du comportement dynamique et électrique des portes situées dans son voisinage. Pour chaque porte, il peut exister des milliers de configurations, il devient donc impossible de déterminer des *stimuli* permettant de prévoir de façon fiable des temps de propagation réalistes. Pour résoudre ce problème, l'utilisation des analyseurs temporels est donc la seule solution, même pour des circuits de quelques milliers de transistors seulement.

Dans cette thèse, j'ai modifié le logiciel d'analyse temporelle HiTas développé au LIP6 afin d'introduire, dans le calcul des délais, les effets des résistances et des capacités de couplage. Afin de ne pas obtenir des résultats exagérément pessimistes, la nature dynamique de l'effet de couplage est prise en compte grâce à une analyse de stabilité, et la définition de l'observabilité d'une agression sur la stabilité. Le calcul des délais élémentaires en prenant en compte la stabilité des signaux est réalisé par un second outil que j'ai développé intégralement, Ctk, qui utilise les résultats de HiTas et de l'analyseur de stabilité STB de la société Avertec.

Mot Clés :

- Analyse temporelle.
- Analyse de stabilité.
- Capacités de couplage.
- Résistance d'interconnexion.
- Bruit de commutation.

Abstract

During the design flow of large deep submicronic integrated microelectronics digital systems, timing verification stage become an important part of validation. This stage is realised with two family of software : timing analyser and electrical simulator. First are less accurate than seconds, but they are enable to analyse chip that contains millions of transistors. They bring in effectiveness what they lose in precision.

With deep submicronic technologies, electrical effects previously negligible, are becoming significant, indeed dominant. These effects are interconnect resistances and coupling capacitances. With this second phenomenon, electrical simulator become definitively unusable. Indeed, they produces exacts results from a set of predefined input patterns. Now, each gate delay depend of dynamic and electrical behaviour of other gate placed in its physical proximity. For each gates, many thousand of different events can occur, so, it becomes impossible to determine input pattern that gives real worst case delay for the circuit. To solve this problem, only the uses of timing analyser become realistic, even for small circuit that contain only a few hundred of transistors.

In this thesis, I have modified existing timing analyser HiTas, made in the LIP6 laboratory, to introduce influence of interconnect resistances and capacitive coupling effects. To avoid too pessimistic results, dynamic behaviour of coupling is taken into account with a stability analysis, and we also introduce the effect of an observable aggression on stability. Elementary gate delay evaluation with stability analysis is realized with a second tool, called Ctk, that use results from HiTas and STB, the timing analyser from spin-off company Avertec.

Keyword :

- Timing analysis.
- Stability analysis.

- Coupling capacitance.
- Interconnect resistance.
- Switching noise.

Remerciements

Je tiens tout d'abord à remercier chaleureusement Mme Marie-Minerve Louërat, chargée de recherche CNRS au LIP6, d'avoir dirigé cette thèse. Sa disponibilité, ses qualités scientifiques et humaines et son optimisme ont permis de voir ce travail de recherche et ce manuscrit aboutir.

Je remercie le professeur Alain Greiner pour m'avoir accueilli dans le département ASIM du laboratoire LIP6, où il a su faire régner une ambiance de travail des plus agréables dans des conditions qui auraient pu être difficiles sans sa volonté de voir rester ce laboratoire tel qu'il est depuis plus de dix ans. Le cotoiement des équipes de recherche matériel et logiciel, ainsi que les enseignements, m'ont permis de trouver auprès de chacun des compétences, des critiques ou des conseils très enrichissants.

Je remercie Messieurs Etienne Sicard et Christian Piguet d'avoir bien voulu juger cette thèse en acceptant d'en être les rapporteurs, ainsi que Messieurs Alain Debreil et Michel Robert de l'intérêt qu'ils ont porté à ce travail en acceptant de l'examiner.

Bien sûr, je ne saurais oublier Karim Dioury, ancien thésard et ami proche, pour son aide précieuse apportée depuis mes débuts dans ce laboratoire durant mon stage de Maîtrise et jusqu'à aujourd'hui.

Ensuite, je tiens à remercier tous mes amis du laboratoire, thésards ou permanents, pour m'avoir accompagné durant ces longues années. De discussions scientifiques au laboratoire aux propos plus détendus autour d'un verre - ou inversement -, tous ont contribué d'une façon ou d'une autre à voir mon travail aboutir.

Enfin, je ne saurais que trop remercier ma femme Roselyne pour le soutien qu'elle m'a apporté tout au long de ces années.

"A tous ceux qui m'ont demandé un jour :
Et alors ? Ta thèse?"

Table des matières

1	Introduction	25
1.1	La validation des circuits intégrés	25
1.2	Description d'un circuit pour la vérification	27
1.3	L'analyse temporelle au LIP6	27
1.3.1	L'abstraction fonctionnelle	28
1.3.2	Le calcul des temps de propagation des portes	28
1.3.3	Le calcul des chemins critiques	29
1.3.4	L'analyse de stabilité	30
1.3.5	L'évolution de l'analyseur temporel du LIP6	30
1.3.6	La valorisation industrielle	30
1.4	Caractéristiques de HiTas	31
1.4.1	Le fichier d'entrée	31
1.4.2	La phase de désassemblage	31
1.4.3	Le calcul des délais élémentaires	32
1.4.4	Le graphe de causalité	32
1.4.5	Le graphe des chemins critiques	33
1.4.6	Limites à l'utilisation de HiTas	35
1.5	Les capacités de couplage	35
1.6	Plan de la thèse	36
2	Problématique des effets submicroniques.	37
2.1	Environnement de caractérisation	37
2.1.1	Technologie et chaîne de CAO	37
2.1.2	Éléments parasites	38
2.1.3	L'inverseur élémentaire	39
2.2	Influence de la résistance des fils d'interconnexion.	40
2.2.1	Expérimentation.	40

2.2.2	Calcul des délais de porte et de ligne.	41
2.3	Influence des capacités de couplage.	42
2.3.1	Introduction.	42
2.3.2	L'influence temporelle d'un agresseur.	43
2.3.3	L'influence temporelle de plusieurs agresseurs.	49
2.3.4	Propagation d'une transition à travers plusieurs portes.	54
2.4	Bruit de couplage capacitif	57
2.5	Influence combinée des résistances et fils d'interconnexion.	58
2.5.1	Effets temporels	58
2.5.2	Influence sur le bruit de commutation	62
2.5.3	Conclusion	62
2.6	Contraintes sur les volumes de données	63
2.6.1	La représentation des circuits en mémoire	63
2.6.2	L'approche hiérarchique	65
2.7	Conclusion	66
3	Vérification post layout - Etat de l'Art	67
3.1	Extraction des circuits	67
3.1.1	Introduction	67
3.1.2	Extraction de circuits dans le cadre de l'analyse temporelle.	67
3.1.3	Les formats de fichiers	68
3.1.4	Les logiciels disponibles	69
3.2	Les analyseurs temporels	72
3.2.1	Solution proposée par Mentor Graphics : Mach TA	72
3.2.2	Solution proposée par Cadence : Celtic	73
3.2.3	Solution proposée par Synopsys : PrimeTime SI et Pathmill	74
3.2.4	Solution proposée par Sequence : ShowTime	74
3.2.5	Conclusion	75
3.3	Les solutions publiées	75
3.3.1	Les modèles de calcul	75
3.3.2	Les méthodologies	77
3.4	Le calcul des délais élémentaires dans HiTas	78
3.4.1	Les temps de propagation dans les portes	78
3.4.2	Les temps de propagation dans les lignes RC	84
3.5	Conclusion	87

4	Modélisation des temps de propagation	89
4.1	Evolution du modèle de charge de porte dans HiTas	89
4.1.1	Problème	89
4.1.2	Calcul de la charge équivalente à un réseau d'interconnexion en sortie d'une porte.	90
4.1.3	Application dans HiTas	92
4.1.4	Validation de l'approche	94
4.1.5	Conclusion	100
4.2	Prise en compte du couplage	100
4.2.1	Présentation générale	100
4.2.2	L'effet Miller.	101
4.3	Introduction de l'effet Miller dans le calcul du délai d'Elmore.	104
4.4	Modèle de calcul des temps de propagation dans les portes en présence de dia- phonie entre interconnexions résistives	106
4.4.1	Calcul de la charge équivalente de la porte en présence de diaphonie . . .	106
4.4.2	Influence du bruit de commutation sur les temps de propagation des portes	108
4.4.3	Calcul du bruit de commutation	110
4.4.4	Correction sur les temps de propagation	113
4.5	Evaluation des délais extrêmes avec interconnexions résistives diaphonantes . .	114
4.5.1	Le graphe de causalité	114
4.5.2	Utilisation des modèles linéaires	117
4.6	Validation	119
4.6.1	Effet du modèle de charge au deuxième ordre	119
4.6.2	Couplage de valeurs typiques	119
4.6.3	Cas d'un couplage fort de deux agresseurs	121
4.7	Conclusion	125
5	Analyse temporelle avec RC et diaphonie	127
5.1	La nécessité d'une approche pire cas	127
5.1.1	Analyseurs temporels statiques et simulateurs électriques	127
5.1.2	L'introduction des couplages capacitifs dans l'analyse temporelle.	128
5.1.3	Approche basée sur l'analyse de stabilité.	129
5.1.4	Approche itérative.	129
5.2	Les intervalles d'instabilité	130
5.2.1	Définition	130
5.2.2	Détermination de la stabilité d'un sommet du graphe temporel.	133

5.3	Influence des agressions sur la stabilité.	135
5.3.1	Influence des agressions.	135
5.3.2	Les agressions observables.	136
5.3.3	Modification conditionnelle des arcs.	136
5.3.4	Influence d'une agression observable	138
5.3.5	Méthode retenue pour la détection des agressions observables	139
5.4	Approche mono intervalle et multi intervalle.	141
5.5	La boucle de relaxation.	142
5.6	Limites de cette approche.	146
5.6.1	Problème de convergence.	146
5.6.2	La propagation d'une erreur de calcul de délai.	147
5.6.3	Les transitions parasites.	147
5.6.4	Critique du critère d'observabilité	148
5.7	Différence avec les méthodes d'élimination des agressions	151
5.8	Conclusion	152
6	Implémentation : Le logiciel Ctk.	153
6.1	Notations et conventions.	153
6.2	Présentation de Ctk.	155
6.3	Analyseurs temporels disponibles.	155
6.4	Cahier des charges	157
6.5	Influence de la diaphonie sur le graphe de causalité.	157
6.5.1	Calcul de la variation des délais des arcs de type porte et de type RC.	157
6.5.2	Charge au deuxième ordre.	158
6.5.3	Informations de couplage.	158
6.6	Structures de données CTK	160
6.6.1	Contexte.	160
6.6.2	Présentation des structures	160
6.6.3	Réévaluation des délais correspondant aux portes.	162
6.7	L'analyseur de stabilité STB de la société Avertec.	162
6.7.1	Présentation.	162
6.7.2	Conventions et structures de données utilisées par STB	163
6.7.3	Type d'analyse de stabilité requise.	164
6.7.4	Post-traitement à effectuer sur les structures de données de STB	166
6.7.5	Utilisation des informations de STB dans CTK	167
6.8	Architecture et implémentation	168

6.8.1	Introduction	168
6.8.2	Le flot de données.	168
6.8.3	Modifications apportées à HiTas	169
6.8.4	Le programme CTK	171
6.9	Conclusion.	173
7	Résultats	175
7.1	Taille de la vue CTK	176
7.2	Convergence	176
7.3	Observations réalisées sur ces trois circuits	177
7.3.1	Nombre d'agresseurs	177
7.3.2	Importance des agressions	178
7.4	Influence temporelle	180
7.5	Détermination de la charge apparente au deuxième ordre	181
7.6	Conclusion	181
8	Conclusion et perspectives	183
A	Structures de données CTK	187
A.1	La structure CTK qui encapsule la vue TTV :	187
A.2	La structure CTK qui encapsule les signaux de la vue TTV :	187
A.3	La structure CTK représentant un agresseur :	188
A.4	La structure CTK représentant la modification d'un délai d'Elmore :	188
A.5	La structure CTK contenant les informations électriques nécessaires pour calculer les modifications des temps de propagation sur les portes :	189
B	Quelques fonctions de la bibliothèque CTK	191
B.1	La fonction <i>ctk_calclload</i>	191
B.1.1	Arguments :	191
B.1.2	Description :	191
B.2	La fonction <i>ctk_calcslope</i>	191
B.2.1	Description :	192
B.3	La fonction <i>ctk_calcrdelay</i>	192
B.3.1	Description :	192
B.4	La fonction <i>ctk_calcdelay</i>	193
B.4.1	Arguments :	193
B.4.2	Description :	193

C	Formats de fichier	195
C.1	Le fichier ctk	195
C.2	Le fichier car	196
D	Glossaire	199
E	Références en ligne	201
F	Détails des calculs	203
F.1	Le délai d'Elmore	203
F.2	Calcul des n premiers moments sur les nœuds d'un arbre RC	206
F.3	Variation des paramètres y_i lorsqu'une capacité change	208
F.3.1	La capacité	208
F.3.2	La résistance	209
G	Liste des publications	213

Table des figures

1.1	Terminologie employée dans ce manuscrit.	28
1.2	Schéma en porte avec réseaux RC et graphe temporel correspondant.	32
1.3	Différents type de graphes dans l'analyse temporelle.	34
2.1	Définitions des dimensions d'un rectangle constituant une interconnexion.	38
2.2	Valeurs nominales des capacités.	39
2.3	Trois inverseurs reliés par une ligne de longueur variable.	40
2.4	Temps de propagation. Abscisse : longueur de la ligne en μm . Ordonnée : temps de propagation en secondes à partir du nœud 1.	40
2.5	Circuits équivalents utilisés par HiTas pour le calcul des délais élémentaires de la figure 2.3.	41
2.6	Temps de propagation. Abscisse : longueur de la ligne en μm . Ordonnée : temps de propagation en secondes à partir du nœud 1.	42
2.7	Bruit de commutation. Abscisse : Temps (secondes). Ordonnées : Tension (volts).	43
2.8	Transition d'un signal suivant l'activité d'un agresseur. Abscisse : temps (secondes), Ordonnée : Tension (volts)	44
2.9	Influence temporelle d'un agresseur sur un signal victime.	44
2.10	Temps de propagations des portes 1 et 2 de la figure 2.9 dans le cas où agresseur et victime commutent dans le même sens. Abscisse : Δt (en secondes), Ordonnée : Temps de propagation (en secondes).	45
2.11	Temps de propagations des portes 1 et 2 de la figure 2.9 dans le cas où agresseur et victime commutent dans des sens opposés. Abscisse : Δt (en secondes), Ordonnée : Temps de propagation (en secondes).	46
2.12	Temps de propagation de la porte 1 relevés pour différentes valeurs de capacité Cl. Abscisse : Δt (en secondes), Ordonnées : Temps de propagation (en secondes).	47
2.13	Temps de propagation de la porte 3 de la figure 2.9 relevés pour différentes valeurs de capacité Cl. Abscisse : Δt (en secondes), Ordonnées : Temps de propagation (en secondes).	47

2.14	Temps de propagation de la victime présentant une discontinuité lorsque l'instant de commutation de l'agresseur varie. Abscisse : Δt (en secondes), Ordonnée : Temps de propagation (en secondes)	48
2.15	Forme des signaux lorsque l'agresseur est décalé de $\Delta t = 38pS$ par rapport à la victime. Abscisse : Temps (en secondes), Ordonnée : Tension (volts)	48
2.16	Signal vic agressé par deux signaux.	51
2.17	Temps de propagation maximum de la porte 1 (fig 2.16) en fonction des instants de commutation relatifs de deux agresseurs. Abscisses : Δt entre l'agresseur et la victime (en picosecondes), Ordonnée : Temps de propagation (en secondes).	52
2.18	Temps de propagation maximum de la porte 2 (fig 2.16)en fonction des instants de commutation relatifs de deux agresseurs. Abscisses : Δt entre l'agresseur et la victime (en picosecondes), Ordonnée : Temps de propagation (en secondes).	53
2.19	Erreur sur un délai élémentaire dans une chaîne.	55
2.20	Circuit simulé	56
2.21	Temps de propagation entre X et E en fonction de la variation de l'instant de commutation de l'entrée X (ΔE). Abscisse : ΔE (secondes), Ordonnée : temps de propagation (secondes)	56
2.22	Bruit de couplage capacitif. En trait continu : le signal sigvic, et en pointillé, le signal sigagr. Abscisse : temps (secondes), Ordonnée : tension (volts).	57
2.23	Influence temporelle d'un agresseur sur un signal victime.	59
2.24	Temps de propagation avec et sans résistances de deux lignes couplées. Abscisse : Δt (secondes), Ordonnée : Temps de propagation (secondes).	59
2.25	Etude de l'influence de la position d'une capacité de couplage.	60
2.26	Temps de propagation entre le nœud 1 et le nœud 2 de la figure 2.25 pour différentes positions du couplage avec prise en compte des effets résistifs. Abscisse : Δt (secondes), Ordonnée : Temps de propagation (secondes).	60
2.27	Temps de propagation entre le nœud 1 et le nœud 4 de la figure 2.25 pour différentes positions du couplage avec prise en compte des effets résistifs. Abscisse : Δt (secondes), Ordonnée : Temps de propagation (secondes).	61
2.28	Bruit de couplage capacitif sur le nœud 2 avec prise en compte des effets résistifs. Abscisse : Temps (Secondes), Ordonnée : Tension (Volts).	62
2.29	Bruit de couplage capacitif sur le nœud 4 avec prise en compte des effets résistifs. Abscisse : Temps (Secondes), Ordonnée : Tension (Volts).	63
3.1	Notations utilisées pour les transistors MOS dans un inverseur.	79

3.2	Modèle équivalent de porte utilisé dans HiTas : Cas d'une transition montante sur l'entrée.	82
4.1	Principe de l'algorithme du calcul récursif de la charge équivalente	90
4.2	Les deux admittances possibles dans le cas des arbres RC	91
4.3	Les charges équivalentes à un développement limité aux degrés 1, 2 et 3	91
4.4	Notations utilisées dans le cas de la charge au deuxième ordre.	92
4.5	Générateur de courant constant chargé par la ligne complète (1) et par la cellule en π équivalente à la ligne (2)	95
4.6	Temps de commutation du générateur de courant constant, en prenant comme charge la ligne complète et la charge en π du montage 4.5. Ordonnée : Temps (S), Abscisse : Longueur de la ligne (μm)	96
4.7	Temps de propagation d'une porte chargée par une cellule en π , calculée par le simulateur algébrique et par la relation 4.36. Ordonnée : Temps (pS), Abscisse : Longueur de la ligne (μm).	98
4.8	Réseau RC en forme d'arbre binaire	98
4.9	Comparaison des temps de propagation entre le simulateur algébrique et l'équation 4.36. Abscisse : front F 20%-80% (pS), Ordonnée : temps de propagation (pS).	99
4.10	Diffusion de l'effet des couplages capacitifs	101
4.11	Définition de l'effet Miller	102
4.12	Délai d'Elmore. Exemple.	105
4.13	Prise en compte du bruit de commutation dans l'évaluation des temps de propagation minimum et maximum. Abscisse : temps (secondes), Ordonnée : tension (volts)	109
4.14	Modèle électrique utilisé pour déterminer approximativement le bruit de commutation	110
4.15	Résistance équivalente d'une porte lorsque sa sortie est soumise à un bruit. Abscisse : Hauteur du bruit (en volts), Ordonnée : Résistance équivalente (en Ω)	111
4.16	Résistance équivalente du transistor en fonction de L pour différentes valeurs de W . Abscisse : L en mètres, Ordonnée : Résistance en Ω	112
4.17	Résistance équivalente du transistor en fonction de W pour différentes valeurs de L . Abscisse : W en mètres, Ordonnée : Résistance en Ω	112
4.18	Schéma équivalent pour le calcul du bruit produit par un agresseur.	113
4.19	Calcul de la correction à partir de la hauteur du bruit et de la pente du signal	114
4.20	Algorithme pour déterminer le temps de propagation des portes d'un circuit.	116
4.21	Exemple de circuits couplés	117

4.22	Temps de propagation d'une porte chargée par une ligne résistive. Abscisse : longueur de la ligne en μm . Ordonnée : Temps de propagation de la porte (secondes).	120
4.23	Exemple d'un circuit couplé possédant des interconnexions résistives.	121
4.24	Signal victime de $200\mu m$ agressé par deux signaux, chacun sur $100\mu m$	121
5.1	Stabilité des signaux	132
5.2	Equivalence entre l'instabilité logique d'un signal et ses formes analogiques possibles.	132
5.3	Propagation entre 2 sommets.	133
5.4	Origine de plusieurs intervalles d'instabilité	134
5.5	Exemple de fusion des intervalles d'instabilité.	134
5.6	Influence d'un changement de temps de propagation entre A et B sur la stabilité d'une série de sommets.	135
5.7	Agression non observable.	137
5.8	Modification partielle des intervalles d'instabilité.	138
5.9	Condition de propagation d'une agression observable.	139
5.10	Plusieurs agresseurs correspondant à plusieurs contexte pour la victime.	139
5.11	Stabilité de deux signaux, et la dernière transition possible sur ces signaux.	140
5.12	Effet d'une agression dans un contexte multi intervalle et mono intervalle.	142
5.13	Exemple de couplages capacitifs bouclés.	143
5.14	Algorithme complet.	145
5.15	Problème de convergence lors de la fusion de deux intervalles d'instabilité.	146
5.16	Exemple de faux chemin statique, vrai chemin dynamique.	148
5.17	Implication du critère d'observabilité sur un exemple simple.	149
5.18	Chronogrammes correspondant aux sommets A et B de la figure 5.17	149
5.19	Application du critère d'observabilité lorsqu'on procède par élimination des agressions.	151
6.1	Utilisation de CTK	156
6.2	Effet d'une capacité de couplage sur le graphe temporel.	159
6.3	Exemple d'une représentation des agressions dans une vue CTK	161
6.4	(a) Portes reliées par un réseau RC, (b) Graphe de causalité correspondant	164
6.5	Intervalles d'instabilité. Situation correspondant à la figure 6.4 où la détection d'une agression est problématique. (a) Stabilité sur les sommets. (b) Stabilité sur les arcs.	165

6.6	Deux intervalles d'instabilité sur le sommet en sortie d'une porte peuvent former un seul intervalle d'instabilité sur un sommet RC.	165
6.7	Fil de routage laissé en l'air dont l'extrémité ne correspond à aucun sommet du graphe temporel.	166
6.8	Plusieurs agresseurs correspondant à plusieurs contexte pour la victime.	167
6.9	Flot de données pour l'analyse temporelle avec prise en compte de la diaphonie.	169
6.10	Vue CNS construite sur une représentation MBK	170
F.1	Définition des délais.	204
F.2	Calcul des moments sur les nœuds d'un arbre RC	206

Liste des tableaux

2.1	Valeurs des capacités à la masse et de couplage typiques.	39
2.2	Valeurs du temps de propagation maximum d'une porte en fonction du nombre d'agresseurs, la totalité des capacités de couplage des agresseurs étant fixé à la valeur constante de $10fF$	54
2.3	Résultat de l'extraction du circuit Noé.	64
2.4	Comparaison en nombre d'éléments résultant de diverses extractions pour les circuits de BULL.	65
4.1	Comparaison des temps de propagation entre le simulateur algébrique et l'équation 4.36 dans le cas d'un arbre binaire ($F = 20pS$).	99
4.2	Les quatre configurations correspondant à l'effet Miller	103
4.3	Comparaison des temps de propagation extrêmes entre l'entrée et différents nœuds du circuit de la figure 4.23 évalués par Eldo et Ctk en fonction des agressions. Les délais nominaux sont évalués avec un état stable des agresseurs.	122
4.4	Détail des temps de propagation de la figure 4.23 évalués par Ctk. Le premier terme des sommes correspond au temps de traversée de la porte et le second au délai d'Elmore	123
4.5	Comparaison des temps de propagation extrêmes entre Vic et V du circuit de la figure 4.24, évalués par Ctk et par Eldo, en fonction des agressions.	124
5.1	Nombre de chemins arrivant sur les sommets de stabilité	131
6.1	Principales bibliothèques utilisées.	154
7.1	Taille de la vue CTK.	176
7.2	Nombre d'itérations dans Ctk.	177
7.3	Nombre d'agresseurs actifs.	178
7.4	Agressions observables les plus sévères.	179

7.5	Agressions les plus sévères en considérant que tous les agresseurs sont observables.	180
7.6	Modifications les plus importantes (en pS) des délais élémentaires.	180
7.7	Modification des chemins critiques (en pS).	180
7.8	Nombre d'évaluation de charge au 1er ordre.	181

Chapitre 1

Introduction

Nous allons premièrement préciser la place et la finalité de l'analyse temporelle dans le flot de conception des circuits intégrés dans le paragraphe 1.1. Puis, nous verrons comment sont décrits ces circuits, et la méthode de conception hiérarchique utilisée pour les concevoir dans le paragraphe 1.2. Nous présenterons ensuite dans le paragraphe 1.3 les principales étapes de l'analyse temporelle, ainsi qu'un historique de la recherche menée au LIP6 sur ce sujet. Ces recherches ont été concrétisées dans la création d'un logiciel, HiTas. Après une rapide présentation dans le paragraphe 1.4 des données manipulées par HiTas et de ses principales caractéristiques, nous expliquerons la nécessité, vue du concepteur de circuit intégrés, de prendre en compte les effets indésirables des capacités de couplage dans le paragraphe 1.5. Nous terminerons ce chapitre d'introduction par une présentation du plan de cette thèse dans le paragraphe 1.6.

1.1 La validation des circuits intégrés

Tout au long du processus de conception d'un circuit intégré microélectronique, les étapes de validation avant sa fabrication occupent une place très importante, que ce soit en temps ou en ressources humaines. Les types de validation sont très diversifiés, ils interviennent à différents stades de la conception et n'ont pas la même finalité. Tous ont en commun de détecter dès que possible les problèmes qui se posent, et d'éviter de fabriquer inutilement des circuits défectueux. En effet, la fabrication de la puce est très coûteuse au niveau des jeux de masque (de l'ordre d'un million d'Euro en 90nm), et prend quelques mois, ce qui peut être très préjudiciable dans un secteur très réactif. De plus, les mesures réalisées sur la puce sont généralement plus difficiles à mettre en œuvre et toujours plus coûteuses que les vérifications avant fabrication.

Il est possible de déterminer au moins trois grandes familles de validation :

1. La première famille est la validation du comportement logique du circuit. Ces validations se font généralement de façon hiérarchique, en validant la fonctionnalité des éléments les plus simples jusqu'au circuit dans son intégralité. Ces opérations sont réalisées par des outils de simulation logique ou logico-temporelle. Ces derniers donnent en plus du comportement logique du circuit une estimation du temps d'exécution des opérations simulées. Lorsque cette étape est passée avec succès, les concepteurs peuvent alors effectuer les opérations de synthèse, de placement et de routage. Bien sûr, chacune de ces étapes doit être validée.
2. La deuxième famille de vérification concerne le respect des contraintes temporelles du circuit, afin de déterminer de façon précise la vitesse de fonctionnement maximale de celui-ci. Cette analyse temporelle se réalise à partir du plan définitif du circuit, de manière à prendre en compte de façon réaliste les éléments parasites liés aux interconnexions. En plus de la vitesse de fonctionnement maximale du circuit, cette analyse permet à l'utilisateur de vérifier ou d'obtenir de nombreuses autres informations, comme les contraintes de pré-positionnement et de maintien à l'entrée des points mémorisants (*setup time* et *hold time*), et la détection des chaînes courtes. Les analyseurs temporels que nous étudions s'appliquent aux circuits numériques synchrones. C'est un sous-ensemble de ces outils que j'ai étudié dans mon travail de recherche.
3. Enfin, il faut vérifier les contraintes physiques de ce circuit. L'utilisation des technologies submicroniques implique la prise en compte de nombreux phénomènes parasites, dont il faut s'assurer qu'ils ne perturbent pas le fonctionnement du circuit. Parmi ces phénomènes, nous pouvons citer :
 - La distribution de l'alimentation électrique et des horloges (le processeur Pentium 4 absorbe un courant de près de 60 Ampères et fonctionne en interne à 2Ghz),
 - La consommation électrique (le processeur Pentium 4 doit dissiper 75 Watt et maintenir sa température en dessous de 76 degrés),
 - L'immunité des signaux aux bruits de commutation,
 - Les variations significatives de performance des transistors sur une même puce,
 - La consommation statique des transistors,
 - Le *Soft Error Rate* : Il s'agit du changement d'état des latches sous l'influence d'une particule alpha ou autre.

1.2 Description d'un circuit pour la vérification

Un circuit intégré numérique en technologie CMOS est composé de **transistors**, regroupés en portes, connectés entre eux par des **signaux**. Le terme signal désigne une connexion au sens large. J'utilise les termes **interconnexion** ou **réseau RC** pour décrire la représentation physique d'un signal. En vue de la simulation électrique, une interconnexion est décrite par un circuit électrique, composé de :

- **nœuds** électriques,
- **résistances** entre ces nœuds,
- **capacités à la masse** : une extrémité de la capacité est sur un nœud du signal, et l'autre est considérée branchée de façon permanente à la masse,
- **capacités de couplage** : les deux extrémités de la capacité sont branchées sur des nœuds pouvant appartenir à des signaux différents. Ces capacités sont aussi appelées capacités flottantes.

Sur les signaux électriques, certains nœuds ont un rôle particulier : ils définissent un point de connexion vers l'extérieur du circuit. Ils sont appelés **connecteurs**. Ces termes sont représentés sur la figure 1.1.

Les transistors sont regroupés par petit nombre, de deux à une dizaine, afin de réaliser une fonction logique de base : ET, OU, NON, élément mémorisant, etc. Ces ensembles de transistors sont appelés **portes**. Dans le cas général, une porte a n entrées, et m sorties. Il est possible de décrire un circuit uniquement en terme de portes, qui sont alors vues comme des "boîtes noires", connectées entre elles par des signaux. Il est possible de regrouper plusieurs portes en **bloc** fonctionnel plus ou moins complexe. Ces blocs sont également reliés entre eux par des signaux. Et plusieurs blocs peuvent être regroupés en un seul bloc, et ce, récursivement, jusqu'à décrire le circuit complètement. C'est la description **hiérarchique** d'un circuit.

1.3 L'analyse temporelle au LIP6

L'étude des outils d'analyse temporelle a commencé au LIP6 à la fin des années 80. Ce type d'outil fait intervenir plusieurs domaines de compétences, qui correspondent aux quatre phases de l'analyse temporelle statique : L'abstraction fonctionnelle du circuit, le calcul du temps de traversée de chaque porte élémentaire, le calcul des chemins critiques et l'analyse de stabilité. Nous donnons maintenant un brève description de chacune de ces phases en indiquant quel a été l'apport du LIP6.

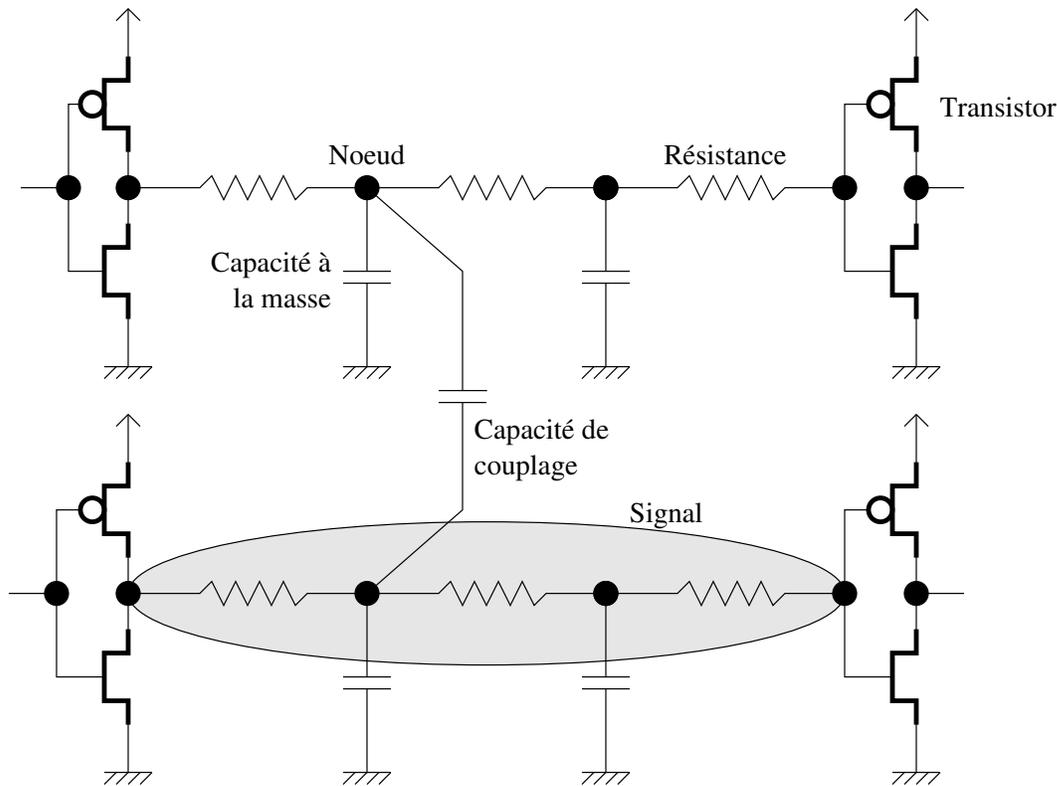


FIG. 1.1 – Terminologie employée dans ce manuscrit.

1.3.1 L'abstraction fonctionnelle

Lorsque le circuit, dont l'analyse temporelle doit être réalisée, est représenté sous la forme de transistors interconnectés entre eux sans information de fonctionnalité, la première opération à réaliser est la reconstruction des portes élémentaires, ainsi que leur fonctionnalité. Cette opération porte le nom d'abstraction fonctionnelle. Deux thèses ont été soutenues au LIP6 sur ce sujet. La première, soutenue par Marc Laurentin en 1992 dans [Lau92], a conduit à la création de l'outil Desb. Les idées développées dans cet outil ont ensuite été reprises et améliorées par Anthony Lester dans le cadre de sa thèse soutenue en 1998 dans [Les98]. Il a ainsi développé l'outil Yagle [LBG98].

1.3.2 Le calcul des temps de propagation des portes

La seconde étape est le calcul des délais élémentaires entre les entrées et les sorties de chaque porte. L'essentiel de ce travail a été réalisé par Amjad Hajjar en 1992 dans [Haj92]. Il a établi un modèle de courant dans les transistors qui lui permet de calculer les temps de traversée des portes les plus complexes. C'est le modèle Mos Canal Court : MCC. Pour faire face

à la complexité des circuits, certaines hypothèses simplificatrices ont dû être faites. Le temps de propagation entre une entrée et la sortie d'une porte dépend de ses caractéristiques intrinsèques (structure de la porte, taille des transistors,...) et de son environnement immédiat : Le front de l'entrée est représenté par une fonction mathématique, la charge de sortie est représentée par une capacité idéale reliée à la masse. Ces hypothèses ne sont bien sûr pas faites au hasard : seuls les phénomènes physiques les plus influents sont pris en compte. Ceux moins significatifs qui sont omis entrent dans la marge d'erreur que peut commettre un analyseur temporel. Toute la difficulté est de répartir ces phénomènes en deux catégories : ceux qui sont influents et ceux qui ne le sont pas. Or, ces choix doivent être réévalués à chaque changement de technologie. En effet, au fur et à mesure que les technologies évoluent, de nouveaux phénomènes doivent être pris en compte, ce qui augmente la complexité de l'analyse temporelle. Heureusement, l'évolution parallèle du matériel et des algorithmes permet de compenser cette augmentation de complexité. L'objectif est donc d'ajuster la marge d'erreur des analyseurs temporels pour ne pas augmenter la complexité de l'analyse temporelle plus rapidement que les moyens dont nous disposons à une époque donnée. Amjad Hajjar a pris en compte certains phénomènes peu significatifs avec une approche pire cas. Il s'agit de l'état des autres entrées de la porte, les transitions partielles sur les entrées,... D'autres phénomènes ont été négligés, comme les résistances dans les interconnexions entre les portes.

1.3.3 Le calcul des chemins critiques

Dans les circuits synchrones, les informations se propagent d'un élément mémorisant vers d'autres éléments mémorisants au travers de fonctions logiques constituées par des portes élémentaires. C'est l'horloge du circuit qui déclenche cette propagation. Le rôle des éléments mémorisants est de restaurer le synchronisme du circuit. Les points mémorisants, les connecteurs, les contrôles de bus sont appelés points de synchronisation. L'ensemble des portes par lesquelles se propage une information est appelé un chemin. Entre deux points de synchronisation, il peut bien sûr y avoir plusieurs chemins possibles. Le temps de propagation d'un chemin s'obtient en effectuant la somme des délais élémentaires de chaque porte sur ce chemin. Entre deux points de synchronisation, les chemins qui ont le temps le plus court et le plus long sont appelés chemins critiques. Ce problème d'apparence simple cache une difficulté liée au nombre de chemins critiques dans un circuit, puisqu'il peut évoluer dans le pire cas comme le carré du nombre de points de synchronisation. Cette augmentation du nombre de chemins rend l'étude de certains circuits, pourtant peu complexes, très coûteuse. Dans sa thèse soutenue en 1998 [Dio98], Karim Dioury a résolu ce problème en remarquant que, dans un circuit, il existe une notion de localité des chemins correspondant à la hiérarchie de conception.

1.3.4 L'analyse de stabilité

La seule vitesse de fonctionnement du circuit, déterminée par la valeur de la chaîne critique la plus longue du circuit, est une information qui est incomplète. En effet, il est nécessaire de connaître la stabilité des signaux aux points de synchronisation par rapport au front d'horloge. Ces deux valeurs sont des contraintes liées à la porte, et il est indispensable de les respecter. L'outil d'analyse de stabilité effectue ces calculs et ces vérifications. Une thèse a été soutenue sur ce sujet au LIP6 par Messaoud Chiheb, en 1992, dans [Chi92]. Un prototype d'outil, STA, avait été développé alors. Il est aujourd'hui inutilisable.

1.3.5 L'évolution de l'analyseur temporel du LIP6

Amjad Hajjar avait développé la première version d'un logiciel d'analyse temporelle, basé sur le modèle MCC, et l'avait appelé TAS, qui est l'acronyme de Timing Analyser with Slope. Ce logiciel utilisait l'abstracteur fonctionnel Desb, et une représentation de chemins à plat. Cette version a été largement améliorée par Karim Dioury, qui a introduit l'utilisation de l'abstracteur fonctionnel Yagle, ainsi que plusieurs innovations, dont la représentation des temps de propagation dans un circuit par une vue temporelle multi-niveau, la résolution du problème de croissance quadratique des chemins par un algorithme de factorisation, ainsi que la prise en compte du temps de propagation dans les résistances d'interconnexions. J'ai participé à l'introduction de cette dernière caractéristique lors de mon stage de DEA. Ce logiciel s'appelle maintenant HiTas, pour Hierarchical TAS. Les améliorations apportées à ce logiciel ont été réalisées en partenariat avec la société Bull, qui utilise aujourd'hui cet outil de manière intensive.

1.3.6 La valorisation industrielle

Karim Dioury et Anthony Lester sont les deux fondateurs la société Averttec, qui commercialise HiTas et Yagle. Ils ont apporté des améliorations considérables à ces deux outils, afin de s'adapter aux contraintes industrielles. Ils ont également développé un nouvel outil d'analyse de stabilité, STB. Averttec existe depuis 4 ans, et emploie aujourd'hui 12 personnes. Leur technologie est actuellement utilisée par une dizaine d'entreprises, et des versions académiques de leurs outils sont également disponibles. Une collaboration étroite existe aujourd'hui entre le LIP6 et Averttec.

1.4 Caractéristiques de HiTas

Les caractéristiques présentées ici concernent l'état de HiTas quand Karim Dioury et Anthony Lester ont quitté le laboratoire en 1999.

1.4.1 Le fichier d'entrée

Ce logiciel utilise les structures de données de la chaîne de CAO Alliance [PBSG01] (voir le chapitre 6.1 pour la définition des bibliothèques et structures de données). Les éléments supportés sont les transistors, les résistances d'interconnexions flottantes et les capacités (à la masse ou flottantes). Le format d'entrée est le format Spice, ou tout autre format compatible avec Spice, en particulier le format SPF. Dans ce fichier, les interconnexions sont décrites comme un ensemble de résistances et de capacités désordonnées. C'est une bibliothèque d'Alliance qui reconstruit les signaux, et construit la structure de données Alliance.

J'ai développé durant mon stage de DEA, puis au cours de ma thèse, tout le support dans Alliance des résistances d'interconnexions et des capacités de couplage, ainsi que l'analyseur lexical et syntaxique (parser) au format Spice. A la requête de Bull, j'ai également ajouté dans Alliance la possibilité de lire des fichiers au travers d'un filtre. Cela permet de lire directement des fichiers compressés, avec le logiciel gzip par exemple. Cette fonctionnalité, d'apparence assez anodine, se révèle extrêmement efficace en terme de gain d'espace disque car les fichiers Spice se compressent avec un gain en moyenne égale à 5.

1.4.2 La phase de désassemblage

Aucune information additionnelle pour le désassemblage n'est nécessaire, grâce à l'utilisation de Yagle, intégré à HiTas. Les structures reconnues sont très variées : portes CMOS, les bleeders, portes MOS incluant des transistors de passage, les latches, etc. Yagle fait partie de la catégorie des abstrauteurs fonctionnels hybrides : Les structures sont reconnaissables de manière automatique et générique, mais il est également possible de procéder à une construction des portes par reconnaissance hiérarchique. Pour cela, le concepteur fournit la description des structures à reconnaître avec le langage de description de règles GENIUS [Les98], proche du VHDL. La bibliothèque Yagle produit à la fin de son exécution une vue en **cônes**. Un cône peut être vu comme une porte à n entrées et une sortie.

1.4.3 Le calcul des délais élémentaires

Le calcul des temps de propagation se fait en utilisant la vue en cône. Dans cette vue, les transistors sont orientés et regroupés en porte. Le calcul des temps de propagation de chaque porte se fait en utilisant le modèle de courant MCC dans les transistors. Ce modèle sera présenté un peu plus loin, dans le chapitre 3.4.1. Il a deux avantages. Tout d'abord, il est possible de l'appliquer pour toutes les structures de porte reconnues par Yagle, et il offre une bonne précision pour un coût CPU très minime. La seule intervention du concepteur est de fournir sur les fronts sur les connecteurs d'entrée, et la charge sur les connecteurs de sortie.

1.4.4 Le graphe de causalité

Considérons le graphe causalité créé par HiTas représenté sur la figure 1.2. Les sommets du

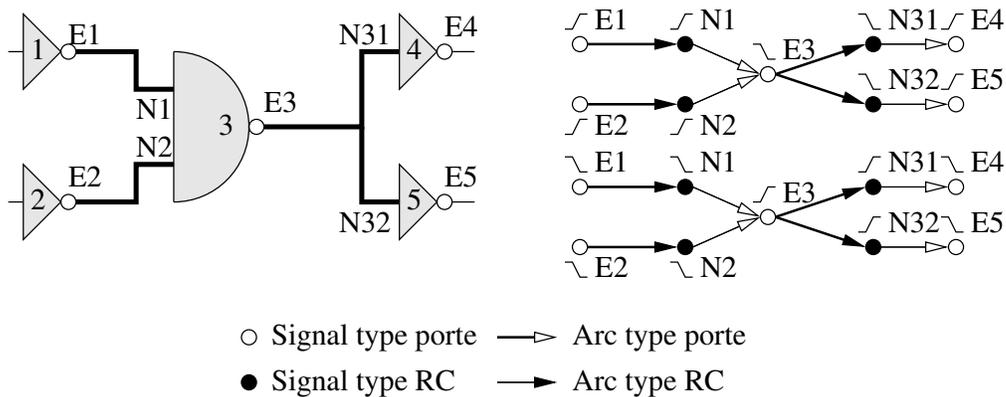


FIG. 1.2 – Schéma en porte avec réseaux RC et graphe temporel correspondant.

graphe temporel sont des événements (transition montante ou descendante) sur des signaux. Un sommet du graphe de causalité est appelé un "signal", mais il faut comprendre "événement sur l'entrée ou la sortie d'une porte". Les arcs représentent des relations de cause à effet directes entre ces sommets. Ces relations correspondent à deux types de temps de propagation : les temps de traversée d'une porte d'une part, et les temps de propagation des réseaux RC d'autre part. Chaque arc possède 4 valeurs : les temps de propagation minimum et maximum, ainsi que les fronts minimum et maximum à l'extrémité de cet arc. Nous distinguons également deux types de sommet : les sommets correspondant à la sortie d'une porte et à l'entrée d'un réseau RC, et les sommets correspondant à l'extrémité d'un réseau RC et à l'entrée d'une porte. Ce graphe de causalité est représenté par une structure appelée **TTV** dans HiTas. Parmi les sommets de ce graphe, certains correspondent à des points mémorisants. L'écriture d'une valeur se fait sous le contrôle d'un signal de commande. Les arcs dont l'extrémité terminale est

un point mémorisant comportent également une référence au sommet du signal de commande. Une propagation à travers un arc de ce type n'est possible que si le sommet de commande qui le contrôle est actif. Cette activation se fait sur front (registre) ou sur état (latch).

1.4.5 Le graphe des chemins critiques

A partir du graphe de causalité, on détermine les temps de propagation minimum et maximum entre chaque point de synchronisation. Cette information doit être calculée pour pouvoir effectuer l'analyse de stabilité. Il est possible de représenter cette information également sous la forme d'un graphe : c'est le graphe des chemins critiques. Les sommets de ce graphe sont les événements sur les points de synchronisation, et les arcs sont les événements possibles entre eux. Le nombre d'arc de ce graphe peut croître comme le carré du nombre de points de synchronisation. En effet, si un circuit comporte n points de synchronisation, et que chacun d'eux peut transmettre une information au cours d'un cycle d'horloge aux n points de synchronisation, il y a n^2 temps minimum et maximum. Ce graphe comporte donc moins de sommets que le graphe de causalité, mais peut contenir beaucoup plus d'arcs. Dans de nombreux cas, le graphe des chemins critiques occupe plus de place en mémoire que le graphe de causalité, et moins d'information. Le seul avantage est que la récupération d'un temps entre deux sommets est immédiate. Sur l'exemple représenté sur la figure 1.3, le graphe de causalité comporte 17 sommets et 20 arcs, et le graphe des chemins critiques comporte 8 sommets et 16 arcs. En supposant que l'objet sommet occupe la même place que l'objet arc, le premier graphe comporte 37 objet, et le second 24. Dans ce cas particulièrement simple, le graphe des chemins critiques est plus compact que le graphe de causalité.

Dans HiTas, deux solutions sont apportées pour résoudre le problème du volume de données du graphe des chemins critiques. La première est assez originale, puisqu'elle consiste paradoxalement à ne pas construire ces temps, mais de permettre leur calcul de manière très rapide à partir du graphe de causalité. Pour cela, on ne conserve dans le graphe de causalité que certains sommets : Les sommets correspondant aux points de synchronisation, et certains sommets particuliers, appelés sommets de factorisation. Ce nouveau graphe est appelé graphe de causalité réduit. Par rapport au graphe des chemins critiques, ce graphe de causalité réduit contient un peu plus de sommets, et beaucoup moins d'arcs. Cela est illustré sur la figure 1.3, où le sommet de factorisation choisi est grisé. Le graphe réduit comporte 17 objets. C'est la représentation la plus compacte.

La seconde solution est indépendante de la première, bien que les deux puissent être utili-

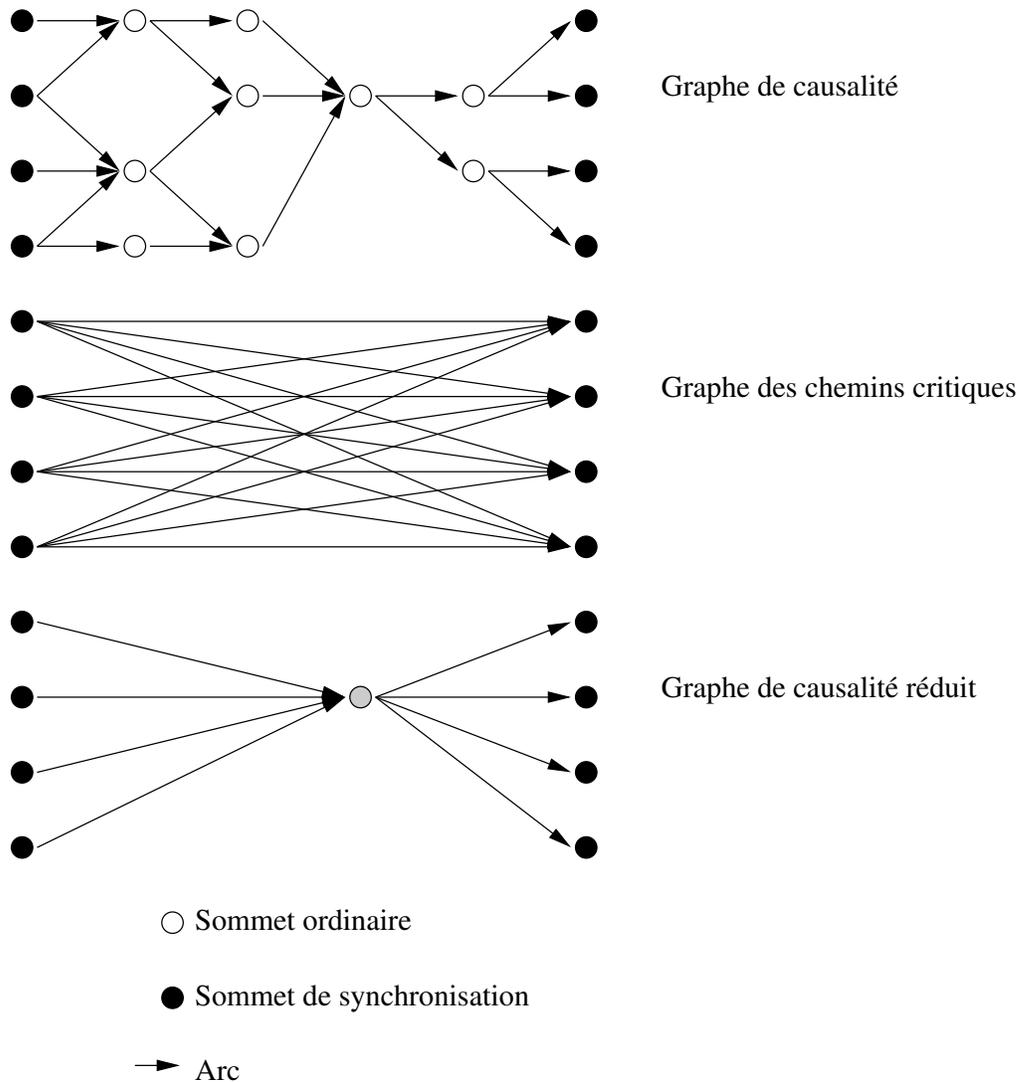


FIG. 1.3 – Différents type de graphes dans l'analyse temporelle.

sées simultanément. Dans les circuits complexes, les blocs sont interconnectés entre eux par un nombre réduit de signaux. Il existe donc une certaine notion de localité qui peut être exploitée : un chemin particulier n'est concerné que par une petite partie du graphe de causalité. Le graphe de causalité est partitionné suivant le découpage hiérarchique utilisé pour la conception du circuit. La vue temporelle d'un bloc devient une instance dans l'arbre d'instanciation de la vue temporelle du circuit. Lors de la recherche d'un chemin, seuls les blocs concernés par celle-ci sont chargés en mémoire au travers d'un mécanisme de mémoire cache. Par rapport à l'utilisation du *swap* (fichier d'échange) du système d'exploitation, il n'y a pas d'écriture sur disque lorsque la description d'un bloc est éliminée de la mémoire.

1.4.6 Limites à l'utilisation de HiTas

Le modèle de courant dans le transistor MCC a été développé à une époque où la technologie $1\mu m$ paraissait lointaine. Aujourd'hui, la technologie $0,12\mu m$ se généralise. Le contexte d'utilisation de HiTas ayant évolué, certains aspects de l'analyse temporelle doivent être revus. Actuellement, deux facteurs risquent de dégrader la précision des résultats de HiTas dans les prochaines années : tout d'abord, l'effet de masquage de la charge par la résistance de la ligne que nous allons voir dans le chapitre 2.2.1 page 40 n'est pas pris en compte. Ensuite, et cela est plus particulièrement l'objet de ma thèse, HiTas ne tient pas compte de l'influence des capacités de couplage. Une grande partie de ma thèse a donc été de proposer des méthodes pour améliorer ce dernier point. HiTas s'est montré capable d'analyser avec succès de nombreux circuits, dont un circuit conçu par Bull et contenant plus de 24 millions de transistors [DLD⁺00].

1.5 Les capacités de couplage

Avec l'évolution technologique, un phénomène jusqu'à présent peu influent est en train de devenir non négligeable : il s'agit du couplage capacitif qui existe entre les fils d'interconnexion. C'est un phénomène difficile à prendre en compte car nous verrons qu'il dépend à la fois des positions relatives des fils, et de l'activité électrique sur ceux-ci. En principe, ce phénomène est pris en compte par les routeurs qui espacent les fils afin de diminuer cette influence. Il demeure néanmoins utile d'avoir une estimation de la variation de fréquence de fonctionnement des circuits due à ce phénomène. Il appartient ensuite au concepteur de déterminer si cette variation est acceptable ou non. L'essentiel de ma thèse a été de réaliser un prototype permettant de prendre en compte ce phénomène, sans négliger les résistances d'interconnexion.

Ce sujet peut se décomposer en deux parties : d'une part les modèles électriques développés afin d'évaluer les délais élémentaires des portes et des interconnexions, et d'autre part les algorithmes mis en œuvre pour utiliser ces modèles. Comme dans beaucoup de cas, et avec l'expérience de ma contribution au développement de HiTas, seule la création d'un prototype fonctionnel permet de s'assurer que certains choix se révèlent bons ou moins bons. C'est pourquoi je me suis attaché à réaliser un prototype logiciel mettant en œuvre les méthodes présentées.

Ce travail a été réalisé avec la collaboration des sociétés Avertec et Bull. La première m'a fourni les dernières versions des outils HiTas, Yagle et STB, et la seconde m'a fourni des exem-

ples de circuits réalisés en technologie $0,12\mu m$, ainsi que son expertise sur le sujet.

1.6 Plan de la thèse

Dans le chapitre 2, nous allons présenter l'ensemble des problèmes posés aujourd'hui par les technologies profondément submicroniques. Nous allons passer en revue l'influence des résistances d'interconnexion et des capacités de couplages, puis des deux effets combinés. Dans le chapitre 3, nous allons détailler les caractéristiques de HiTas. Nous étudierons aussi les différentes solutions proposées par la littérature et par les outils industriels pour réaliser l'analyse temporelle des circuits VLSI en présence de diaphonie. Le chapitre 4 présentera les différents modèles de calculs des délais élémentaires que j'ai développés. Les différents algorithmes utilisés seront ensuite présentés dans le chapitre 5, et nous étudierons en détail les outils développés dans le chapitre 6. Nous donnerons les résultats obtenus grâce à l'utilisation de cet outil sur quelques exemples de circuits industriels dans le chapitre 7. Enfin, nous tirerons les conclusions de ce travail dans le chapitre 8.

Chapitre 2

Problématique des effets submicroniques.

Dans ce chapitre, nous allons passer en revue les effets liés aux deux principaux phénomènes parasites qui existent actuellement dans les circuits utilisant des technologies profondément submicroniques : Les résistances d'interconnexion et les capacités de couplage entre signaux. L'influence de l'effet résistif des interconnexions longues dans les "system on chip" est étudié depuis quelques années. Nous allons donc rappeler brièvement l'influence de ce phénomène parasite dans le paragraphe 2.2, en nous intéressant particulièrement à la façon dont il est pris en compte dans HiTas. L'effet de couplage capacitif n'est devenu significatif que pour les technologies les plus récentes. Nous allons donc étudier son influence plus en détail, d'un point de vue temporel dans le paragraphe 2.3 et d'un point de vue fiabilité dans le paragraphe 2.4. Ensuite, nous étudierons les effets combinés de ces deux phénomènes dans le paragraphe 2.5. Finalement, nous verrons quelles contraintes cela induit sur l'extraction, et plus généralement sur la méthodologie utilisée par l'analyse temporelle dans le paragraphe 2.6. Beaucoup de ces points sont décrits dans [Sic98].

2.1 Environnement de caractérisation

2.1.1 Technologie et chaîne de CAO

Dans ce chapitre et les suivants, nous utiliserons pour les simulations les paramètres technologiques $0,25\mu m$ d'un fondeur industriel. Nous disposons pour cette technologie des modèles des transistors CMOS, et des paramètres des interconnexions, tels que la dimension des fils, la valeur des résistances parasites et des capacités de couplage entre fils. A partir de ces paramètres, nous avons créé les fichiers de configuration pour l'analyseur temporel HiTas ([Dio98]),

et l'extracteur Space [vGvdM97]. A partir de la description comportementale haut niveau d'un circuit, nous disposons de toute la chaîne de CAO pour en obtenir une représentation électrique équivalente [PBSG01]. La référence des mesures est le simulateur électrique Eldo, version v5.3.1.1. [Men00]

2.1.2 Éléments parasites

Les phénomènes dont nous parlons dans ce manuscrit sont par nature des phénomènes distribués. La résistance d'un rectangle de longueur L , de largeur W et de hauteur T (voir la figure 2.1) est définie avec la relation :

$$R = \frac{\rho \cdot L}{W \cdot T} \tag{2.1}$$

tandis que la capacité de ce rectangle est définie par des relations plus complexes, la plus

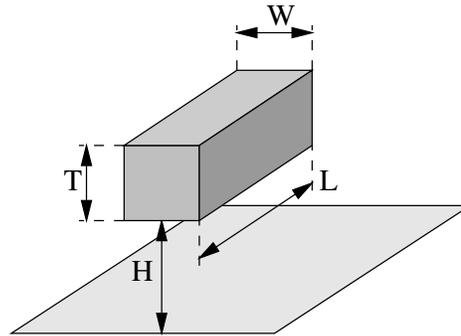


FIG. 2.1 – Définitions des dimensions d'un rectangle constituant une interconnexion.

courante étant donnée dans [ST83] :

$$\frac{C_l}{L \cdot \epsilon_{ox}} = 1.15 \cdot \frac{W}{H} + 2.80 \cdot \left(\frac{T}{H}\right)^{0.222} \tag{2.2}$$

Où H est la distance par rapport au plan de masse de ce rectangle. Le problème est que cette résistance et cette capacité sont réparties sur toute la longueur du rectangle. Si les simulateurs électriques sont capables de représenter ce phénomène distribué, ce n'est pas le cas des analyseurs temporels, ni des extracteurs. Un compromis entre l'efficacité et la précision nous conduit à représenter les lignes par un ensemble de n rectangles élémentaires faisant chacun un micron de long, ce résultat étant obtenu de manière empirique à l'aide de simulations électriques.

Pour nos simulations, nous utiliserons souvent comme référence une ligne du premier niveau de métal. Une ligne de largeur minimale possède une résistance de $0,625\Omega/\mu m$. Les capacités à la masse et de couplage dépendent de l'environnement électrique. Nous avons résumé

dans le tableau 2.1 les valeurs typiques dans le cas de lignes parallèles le plus proche possible, dont les configurations sont données dans la figure 2.2. Sauf indication contraire, ces valeurs sont utilisées dans la suite de ce document.

Configuration	C_m , en $fF/\mu m$	C_C , en $fF/\mu m$
Une ligne isolée	0,0915	-
Deux lignes parallèles	0,0555	0,0927
Trois lignes parallèles, ligne centrale	0,0195	0,1854

TAB. 2.1 – Valeurs des capacités à la masse et de couplage typiques.

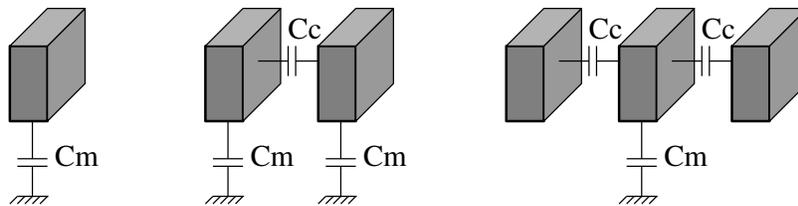


FIG. 2.2 – Valeurs nominales des capacités.

On constate d’après ces valeurs que la part de capacité de couplage C_C peut dépasser les 90% de la part de la capacité totale $C_m + 2.C_C$ sur une interconnexion. Bien sûr, il s’agit d’un cas extrême qui ne prend pas en compte la capacité d’entrée des portes, et qui suppose que les fils sont à une distance minimum, ce qui n’arrive que sur des très courtes distances dans les cellules précaractérisées. L’information à retenir est que la part de capacité diaphonante sur une interconnexion n’est plus négligeable.

2.1.3 L’inverseur élémentaire

L’inverseur standard utilisé pour les simulations qui suivent est extrait à partir de la bibliothèque symbolique de la chaîne de CAO Alliance. Les dimensions du transistor N sont $L = 0,25\mu m$ et $W = 1,25\mu m$, et les dimensions du transistor P sont $L = 0,25\mu m$ et $W = 2,50\mu m$. Ces dimensions ne correspondent pas à un équilibrage temporel, qui est obtenu pour un transistor P de $W = 3,25\mu m$. La capacité d’entrée équivalente de cet inverseur se situe autour de $10fF$.

2.2 Influence de la résistance des fils d'interconnexion.

2.2.1 Expérimentation.

Considérons le montage de la figure 2.3. Trois inverseurs sont reliés par une ligne de lon-

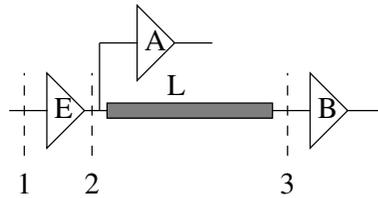


FIG. 2.3 – Trois inverseurs reliés par une ligne de longueur variable.

gueur variable L .

Nous simulons ce circuit sous Eldo et relevons les temps de propagation respectivement entre le nœud 1 et le nœud 2 et entre le nœud 1 et le nœud 3 sur les deux courbes "Simulation charge ligne RC" de la figure 2.4. Nous constatons que le temps de traversée de la porte E -

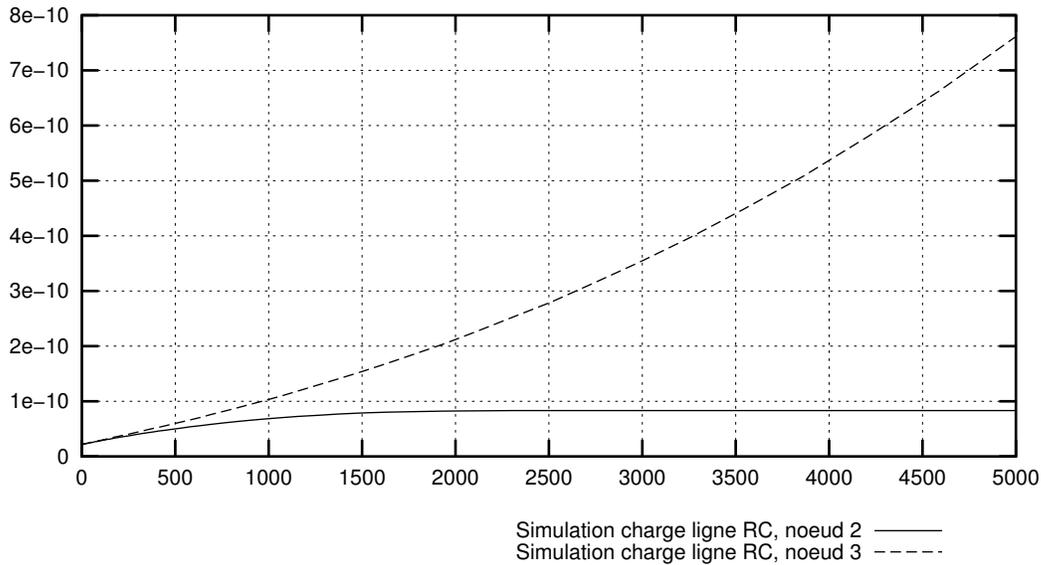


FIG. 2.4 – Temps de propagation. Abscisse : longueur de la ligne en μm . Ordonnée : temps de propagation en secondes à partir du nœud 1.

entre les nœuds 1 et 2 - atteint, pour une longueur de ligne d'environ $1mm$ une valeur constante d'environ $80pS$. Ce temps de traversée n'augmente plus avec la longueur de la ligne. Cela correspond à "l'effet de masquage" de la capacité de ligne par sa propre résistance. Par contre,

le temps de propagation entre le nœud 1 et le nœud 3 croît avec la longueur de la ligne. Ce temps de propagation est donc dû essentiellement au temps de propagation dans la ligne.

2.2.2 Calcul des délais de porte et de ligne.

Dans le cadre de l'analyse temporelle, les modèles de porte sont des modèles relativement simples comparés aux simulateurs électriques. Le temps de propagation d'une porte est déterminé à partir de quelques éléments : La taille des transistors de la porte, les fronts d'entrée et la charge de sortie, qui peut être soit une simple capacité, soit un montage équivalent relativement simple. De même, le temps de propagation dans une ligne est calculé à partir du front de sortie de l'émetteur et des charges équivalentes en entrée des portes. Nous avons représenté sur le circuit de gauche de la figure 2.5 le circuit utilisé lorsque la charge équivalente vue par la porte est une simple capacité C , comme cela est le cas pour le modèle de transistor MCC ([Haj92]) utilisé dans HiTas. Sur le circuit de droite, nous avons représenté le calcul du délai dans la ligne RC. Le front d'entrée est le front calculé en sortie de la porte et les portes de charge ont été remplacées par leur capacité équivalente. Dans HiTas, le délai dans la ligne est le délai

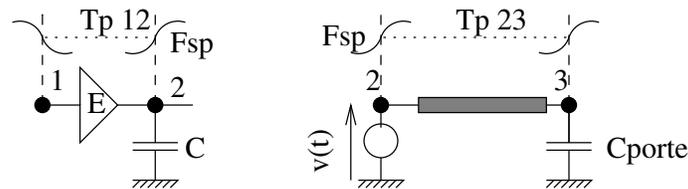


FIG. 2.5 – Circuits équivalents utilisés par HiTas pour le calcul des délais élémentaires de la figure 2.3.

d'Elmore. L'introduction de ce découplage entre le calcul des temps de propagation et le calcul des fronts n'a pour but que de réduire de manière drastique la quantité de calcul à effectuer par l'analyseur temporel.

Le premier problème que nous aurons à résoudre avec l'utilisation du modèle MCC est le calcul de la charge équivalente vue par la porte qui commute. En effet, le fait que la charge équivalente vue par cette porte soit une capacité n'implique pas que la valeur de cette capacité soit la capacité totale sur le réseau : nous négligerions dans ce cas l'effet de masquage des capacités par les résistances. C'est ce que nous constatons avec la courbe "Simulation charge 1er ordre, nœud 2" sur la figure 2.6. Dans cette simulation, nous avons remplacé la ligne de longueur L par une capacité représentant la capacité totale du réseau. Nous obtenons avec cette approche une surestimation très nette du temps de propagation à travers la porte. Ce

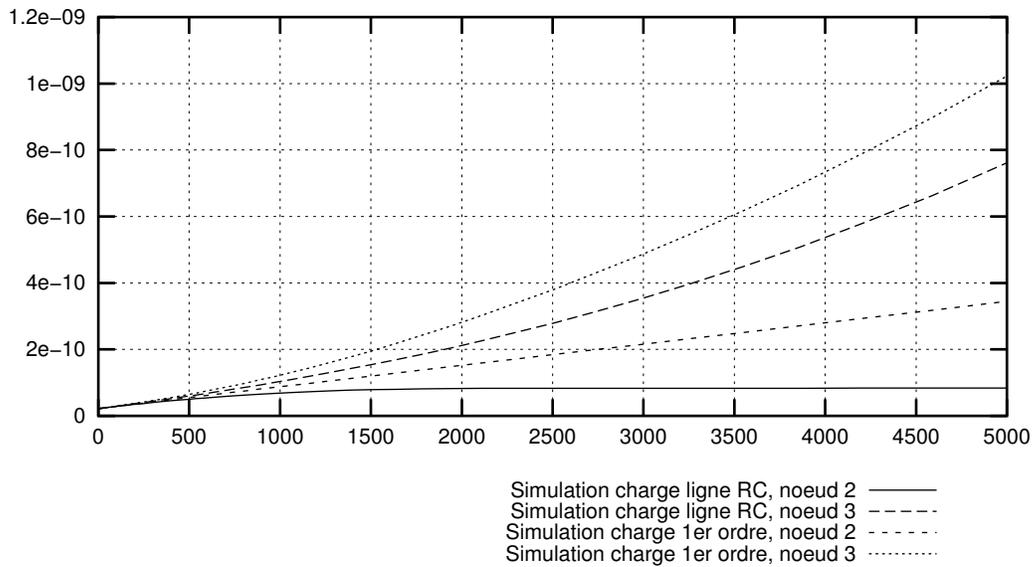


FIG. 2.6 – Temps de propagation. Abscisse : longueur de la ligne en μm . Ordonnée : temps de propagation en secondes à partir du nœud 1.

n'est pas en utilisant une estimation aussi précise qu'Eldo pour l'interconnexion que l'on peut résoudre ce problème, comme le montre la courbe "Simulation charge 1er ordre, nœud 3".

En conclusion, nous observons que la résistance des interconnexions a deux effets temporels : elle diminue le temps de traversée de la porte en masquant les capacités au bout des interconnexions, et elle fait apparaître un temps de propagation dans les interconnexions.

2.3 Influence des capacités de couplage.

2.3.1 Introduction.

Commençons par donner quelques définitions. Une capacité de couplage, encore appelée capacité de diaphonie, est une capacité équivalente qui représente le couplage électromagnétique entre deux conducteurs. Lorsque nous étudions ce phénomène, nous étudions l'influence d'un ou plusieurs conducteurs appelés agresseurs sur le conducteur appelé victime. Selon que le signal victime est stable ou instable, l'influence des effets de couplage agit suivant deux effets. Lorsque la victime est stable, et qu'un ou plusieurs de ces agresseurs commutent, on observe une variation de tension parasite sur la victime. Ce que nous observons sur la figure 2.7. Cette variation de tension est appelée bruit de commutation, sous-entendu dû aux agresseurs. Lorsque la victime commute en même temps qu'un ou plusieurs de ces agresseurs, la transition de la victime est modifiée, et par conséquent le temps de propagation effectif de la porte. C'est

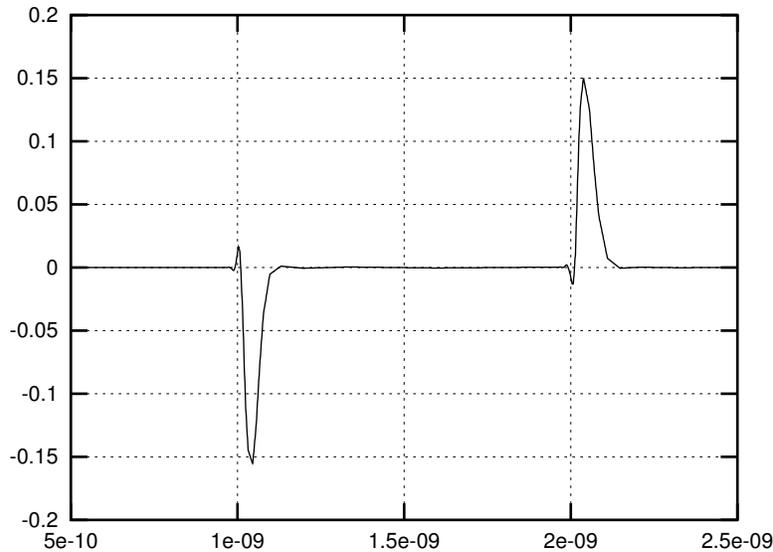


FIG. 2.7 – Bruit de commutation. Abscisse : Temps (secondes). Ordonnées : Tension (volts).

l'effet temporel, que nous cherchons à prendre en compte lors de l'analyse temporelle. Nous nous intéresserons également au bruit de commutation car nous allons voir qu'il est un intermédiaire de calcul dans l'évaluation des délais. Dans cette partie, nous ne ferons que mettre en évidence ces phénomènes, nous verrons ultérieurement les hypothèses et les modèles de calcul permettant de les prendre en compte dans l'estimation des délais.

2.3.2 L'influence temporelle d'un agresseur.

Dans le cadre de l'analyse temporelle, l'influence des capacités de couplage se manifeste par la modification du temps de propagation des portes selon l'activité des agresseurs. Nous observons cet effet sur la figure 2.8, où nous avons relevé la transition d'un signal possédant une capacité de couplage avec un agresseur dans trois cas : l'agresseur est stable, l'agresseur commute dans le même sens que le signal observé, et l'agresseur commute dans le sens opposé au signal observé.

Considérons le montage de la figure 2.9. Les deux signaux $sigvic$ et $sigagr$ représentent deux lignes de $100\mu m$ placées à distance minimale. Nous avons pris en compte la capacité de masse équivalente, ainsi que la capacité de couplage, mais négligé les résistances de ligne sur ces signaux. La capacité C_I représente la charge d'entrée équivalente d'un inverseur. Une transition sur le nœud $invic$ est appliquée à l'instant $t = 0$, et, décalée d'un temps Δt compris entre $-200pS$ et $+200pS$, une transition est appliquée sur le nœud $inagr$. Nous relevons le temps de

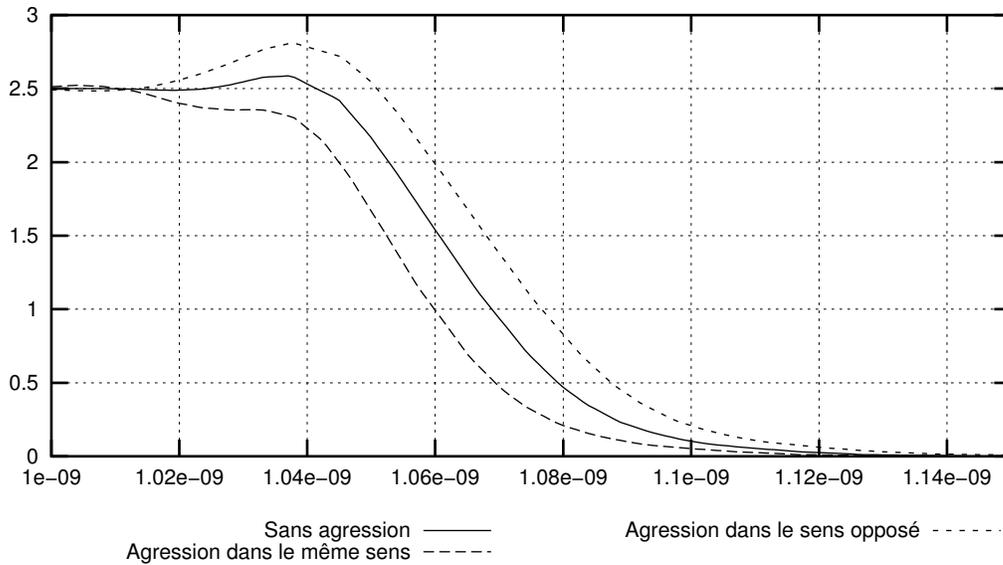


FIG. 2.8 – Transition d’un signal suivant l’activité d’un agresseur. Abscisse : temps (secondes), Ordonnée : Tension (volts)

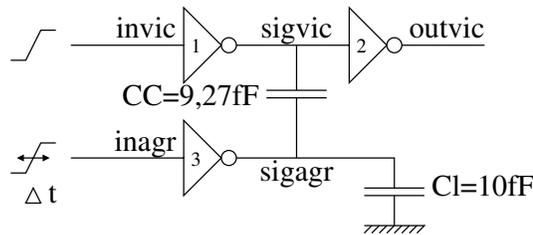


FIG. 2.9 – Influence temporelle d’un agresseur sur un signal victime.

propagation entre *invic* et *sigvic*, et le temps de propagation entre *sigvic* et *outvic* en fonction de l’écart Δt , lorsque les transitions ont lieu dans le même sens sur la figure 2.10 (*invic* et *inagr* montant), et dans des sens opposés sur la figure 2.11 (*invic* descendant et *inagr* montant).

Ces relevés apportent plusieurs enseignements. Tout d’abord, nous observons une variation importante du temps de propagation de la porte 1 de la figure 2.9, et une variation moins sensible sur la porte 2. Ces variations ne sont dues qu’au décalage de l’instant de commutation de l’agresseur. Ensuite, selon le sens des transitions de la victime et de l’agresseur, les délais évoluent dans des sens différents. Lorsque le signal victime et son agresseur commutent dans le même sens, nous observons une diminution du temps de traversée des portes (ici -35%), alors qu’une augmentation du temps de traversée des portes est observée (ici +45%) lorsque

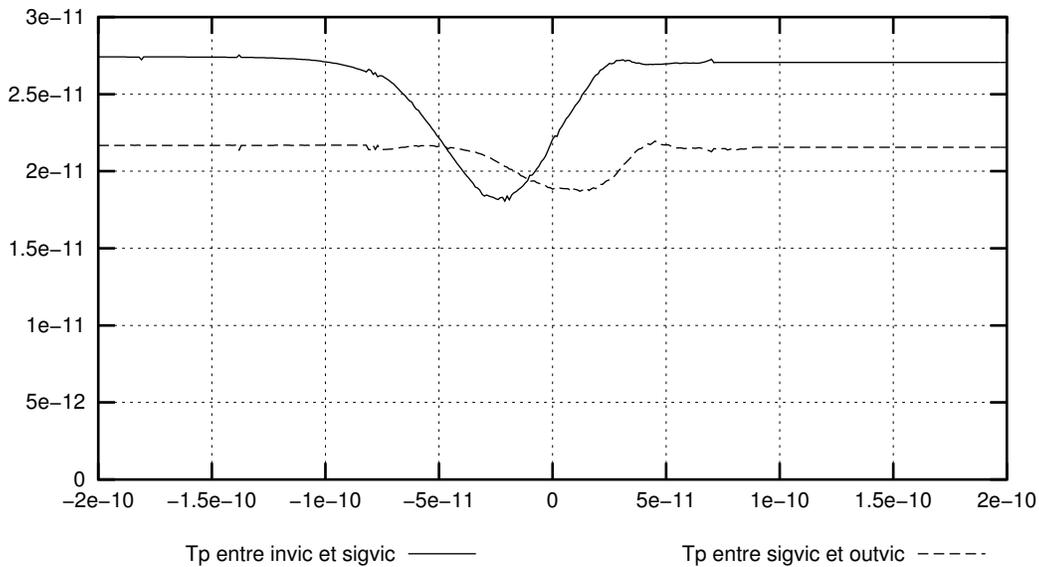


FIG. 2.10 – Temps de propagations des portes 1 et 2 de la figure 2.9 dans le cas où agresseur et victime commutent dans le même sens. Abscisse : Δt (en secondes), Ordonnée : Temps de propagation (en secondes).

les signaux commutent dans le sens opposé. Dans la suite, nous appellerons le premier cas une agression *meilleur cas* (modification du temps de propagation minimum), et le second une agression *pire cas* (modification du temps de propagation maximum).

Ensuite, nous constatons que les agressions qui conduisent aux délais extrêmes, que ce soit en pire cas ou en meilleur cas, ne se produisent pas lorsque les signaux commutent exactement au même instant ($\Delta t = 0$), mais lorsqu'ils sont légèrement décalés. Dans le cadre de l'analyse temporelle statique, nous verrons que ce n'est pas le temps exact qui nous importe, mais les valeurs extrêmes prises par les temps de propagation. Le critère approximatif disant que les conditions d'agression les pires se produisent lorsque les deux signaux commutent au même instant n'est donc pas tout à fait exact. La conclusion de cette première expérience est que *les instants relatifs de commutation entre les deux signaux ont une influence significative sur les temps de propagation des signaux.*

Observons maintenant les mêmes signaux lorsqu'il y a un décalage de puissance et de charge entre les deux émetteurs (portes 1 et 3 figure 2.9). Pour réaliser cette étude, nous faisons varier la capacité C_l de $10fF$ (capacité équivalente d'entrée d'une porte) à $500fF$ (capacité d'entrée de 50 portes), et nous ajoutons deux inverseurs en parallèle à l'inverseur 3. Ainsi, le signal

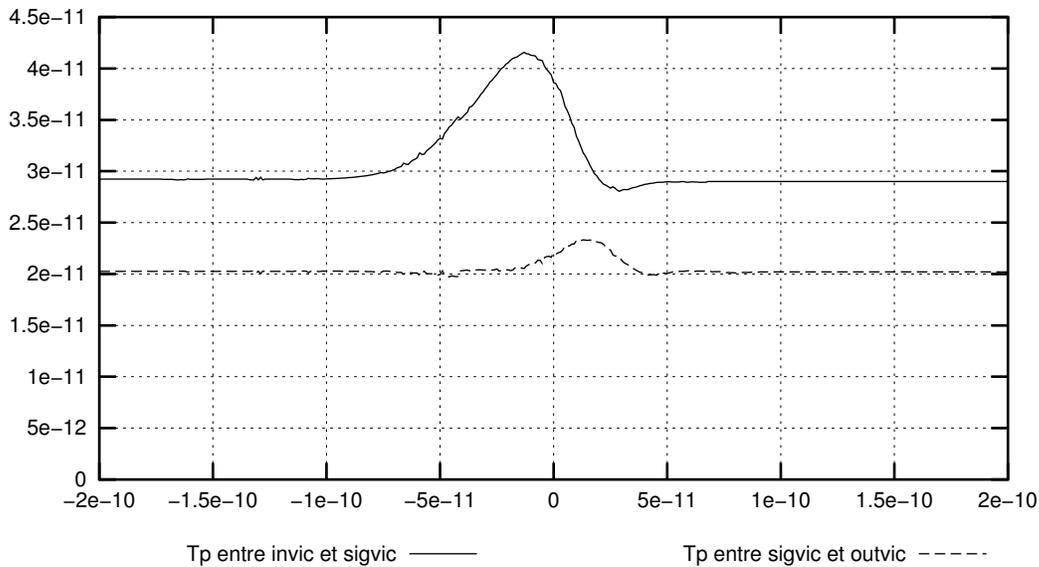


FIG. 2.11 – Temps de propagations des portes 1 et 2 de la figure 2.9 dans le cas où agresseur et victime commutent dans des sens opposés. Abscisse : Δt (en secondes), Ordonnée : Temps de propagation (en secondes).

agresseur est commandé par un inverseur apparemment trois fois plus gros. Nous relevons les temps de propagation entre invic et sigvic sur la figure 2.12 en fonction de Δt , et ce, pour différentes valeurs de capacité. Nous constatons que plus l'agresseur est chargé, moins son influence se fait sentir sur la victime. Nous savons que la pente des fronts dépend du rapport entre la puissance de la porte et sa charge. Plus la charge est importante, plus la transition est lente. L'influence de la capacité de couplage dépend donc de la valeur relative des fronts. Bien sûr, ce phénomène est aussi observable sur l'agresseur. Nous avons relevé le temps de propagation de la porte 3 sur la figure 2.13. Ces temps varient de façon moins marquée. Pour les faibles valeurs de C_l ($10fF$), cela est dû au fait que la porte 3 est bien plus "puissante" que la porte 1, et pour les fortes valeurs de C_l , cela est dû au fait que la part relative de la capacité de couplage est faible.

Toutes ces courbes ont en commun d'avoir des temps de propagation qui varient de manière continue par rapport à l'instant de transition de l'agresseur. Pourtant, il est possible d'observer, lorsque le couplage est très fort, des variations discontinues des temps de propagation, comme cela est le cas sur la figure 2.14. Ce phénomène se produit lorsque, durant la transition d'un signal victime, un agresseur a suffisamment d'importance pour modifier localement le sens de la transition de la victime, comme nous l'avons simulé sur la figure 2.15. L'origine de la dis-

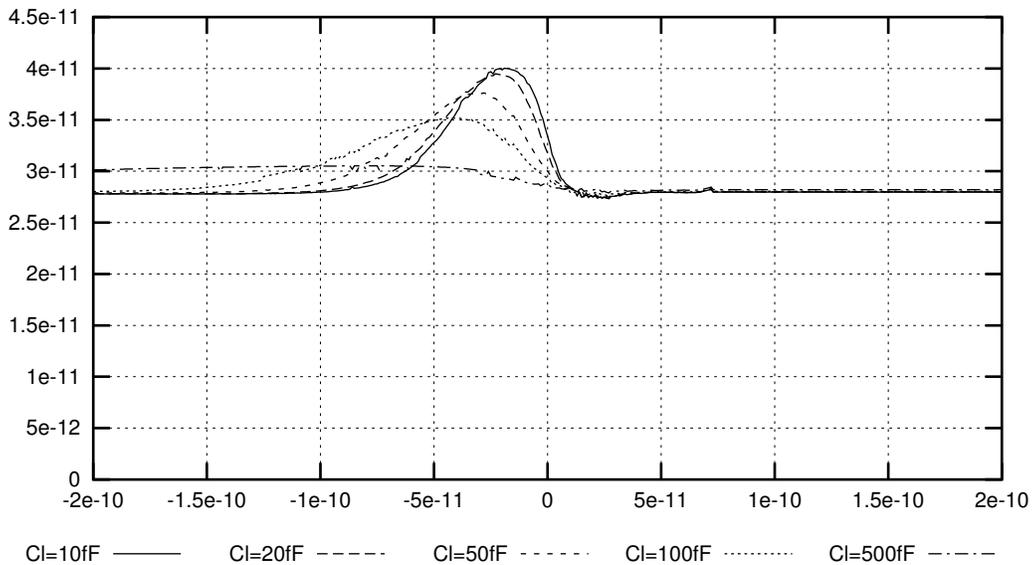


FIG. 2.12 – Temps de propagation de la porte 1 relevés pour différentes valeurs de capacité Cl. Abscisse : Δt (en secondes), Ordonnées : Temps de propagation (en secondes).

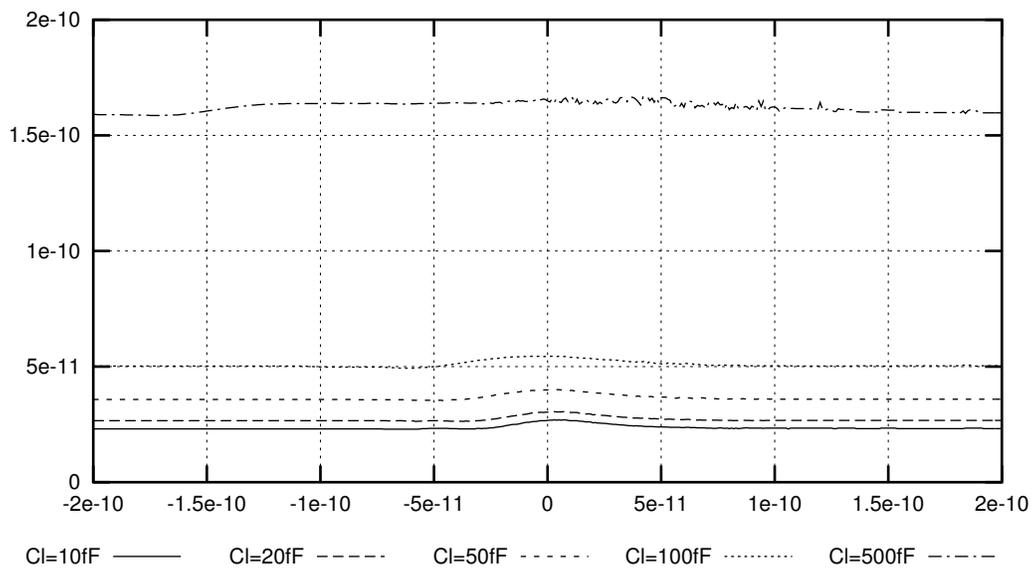


FIG. 2.13 – Temps de propagation de la porte 3 de la figure 2.9 relevés pour différentes valeurs de capacité Cl. Abscisse : Δt (en secondes), Ordonnées : Temps de propagation (en secondes).

continuité de temps de propagation vient du fait que l'on mesure les temps de propagation à l'instant de passage du signal à $v_{dd}/2$. Que l'on prenne le premier ou le dernier instant de passage à cette tension de seuil ne change pas le problème. Dans notre cas, nous prenons tou-

jours le *premier instant*. En pratique, ce phénomène se produit rarement car les couplages ne

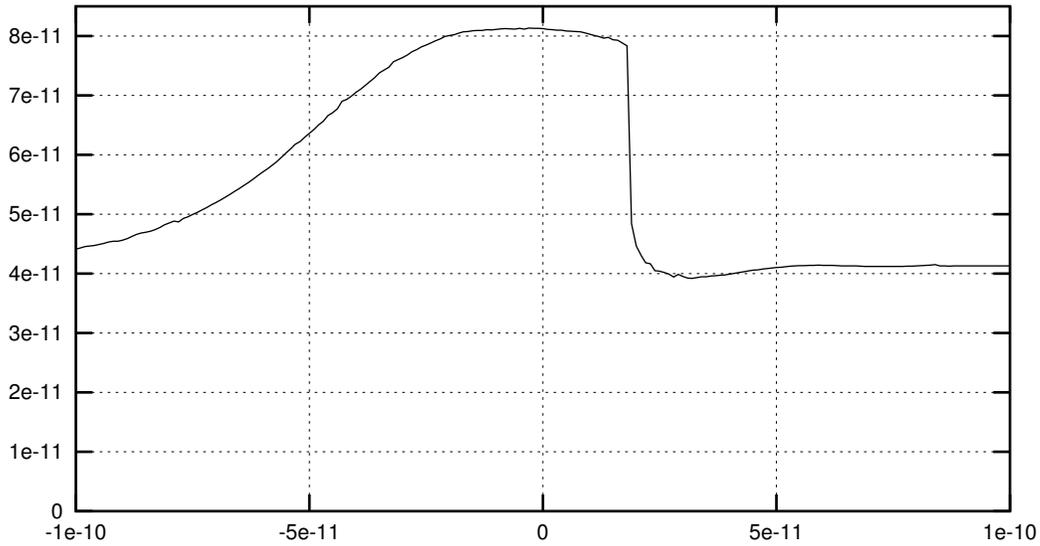


FIG. 2.14 – Temps de propagation de la victime présentant une discontinuité lorsque l’instant de commutation de l’agresseur varie. Abscisse : Δt (en secondes), Ordonnée : Temps de propagation (en secondes)

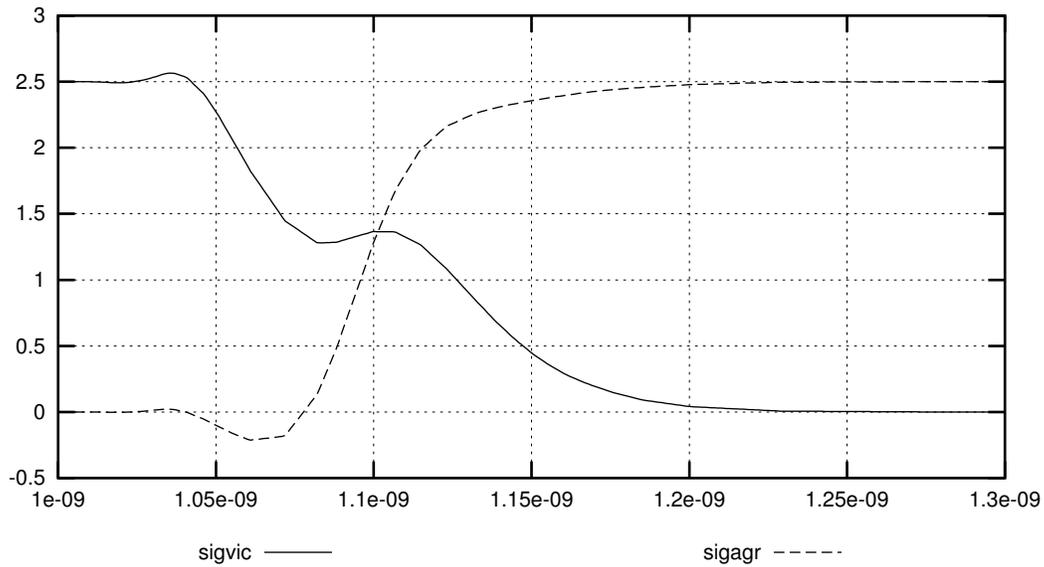


FIG. 2.15 – Forme des signaux lorsque l’agresseur est décalé de $\Delta t = 38pS$ par rapport à la victime. Abscisse : Temps (en secondes), Ordonnée : Tension (volts)

sont pas assez forts. Pour réaliser ces mesures nous avons repris le montage de la figure 2.9, en augmentant la capacité de couplage à $30fF$, et en multipliant par 3 la taille de la porte 3. Néanmoins, cette simulation nous montre que la détermination exacte du temps de propagation n'est pas simple lorsque la transition du signal n'est pas monotone. La détermination des temps de propagation extrêmes ne pose pas ce problème. *Nous avons ici une première raison de déterminer les valeurs extrêmes plutôt que les valeurs intermédiaires.*

Enfin, notons que l'influence des agresseurs se fait aussi sentir lorsqu'ils sont stables, car les capacités de couplage n'agissent pas comme de simples capacités reliées directement à la masse, mais à travers une résistance équivalente à la résistance linéaire des transistors (paragraphe 4.4.3). Cela explique que, selon l'état haut ou bas de l'agresseur statique, le temps de propagation de la victime peut être très légèrement différent.

2.3.3 L'influence temporelle de plusieurs agresseurs.

Méthode de simulation.

Lorsqu'on cherche à prendre en compte l'effet de deux ou plusieurs agresseurs sur un signal, il devient difficile d'obtenir par simulation les temps de propagation minimum ou maximum, puisqu'il s'agit de déterminer l'ensemble des n valeurs Δt correspondant aux n agresseurs. Dans la pratique nous rencontrerons de nombreux cas où, parmi les n agresseurs d'un signal, nous savons qu'un certain nombre d'entre eux ne commuteront jamais en même temps que le signal victime.

Définition 1

Nous appelons une configuration d'agression pire (meilleur) cas l'ensemble des n valeurs Δt des n agresseurs actifs qui permet d'obtenir le temps de propagation maximum (minimum) sur le signal victime.

Définition 2

Nos désignerons par temps de propagation extrême le temps de propagation minimum ou maximum d'une porte, sous l'influence d'un effet de couplage.

Par exemple, sur la figure 2.10 où il n'y a qu'un seul agresseur, la valeur Δt correspondant au temps de propagation minimum entre $invic$ et $sigvic$ vaut $-30ps$. Ces Δt sont évidemment tous différents, selon la puissance de chaque agresseur, sa charge, son front d'entrée... Le problème vient du fait que le Δt d'un agresseur agissant seul sur le signal victime et donnant le temps de propagation extrême dans ces conditions aura une autre valeur lorsqu'on voudra obtenir le temps de propagation extrême avec plusieurs agresseurs.

Dans le cas d'un seul agresseur, obtenir la courbe de l'évolution du temps de propagation en fonction du décalage avec cet agresseur (figures 2.10, 2.11, 2.12, 2.13 et 2.14) nécessite la simulation d'une transition complète autant de fois (N) que nous voulons de points sur la courbe. A partir de cette courbe, nous obtenons le Δt qui nous a donné le temps extrême. Lorsque nous avons deux agresseurs, il faut réaliser N^2 simulations. Pour n agresseurs, il faut N^n simulations. Cela est beaucoup trop coûteux.

Pour obtenir par simulation le temps de propagation maximum d'une porte dont le signal de charge est agressé par n agresseurs, nous procédons par approximations successives. Nous faisons en sorte qu'il n'y ait initialement aucune agression. Cela revient à positionner le Δt de chaque agresseur à la valeur $-\infty$. Nous traçons la courbe, au moyen de N simulations, de la variation du temps de propagation en fonction du Δt du premier agresseur. Nous relevons la valeur Δt qui nous a permis d'obtenir la valeur extrême. Ensuite, nous réalisons la même opération en faisant varier le Δt du second agresseur, en conservant la valeur Δt obtenue pour le premier agresseur. Nous réalisons cette opération successivement pour tous les agresseurs, en conservant les Δt précédemment calculés. Nous avons donc réalisé $n \times N$ simulations pour obtenir le Δt de chaque agresseur. Nous itérons cet algorithme jusqu'à ce que le temps de propagation extrême recherché n'augmente plus. Il est généralement nécessaire d'itérer cet algorithme deux ou trois fois, pour obtenir au final l'ensemble des Δt donnant le temps de propagation extrême recherché. Cette solution nécessite donc $3 \times n \times N$ simulations. La détermination du temps de propagation minimum est identique.

La principale inquiétude que nous avons lors de l'utilisation de cette approche est le risque de tomber sur un maximum ou un minimum local. Nous n'avons cependant pas rencontré de telles situations. Cela est dû au fait que l'influence d'un agresseur sur le temps de propagation se traduit par une courbe présentant un seul maximum (minimum), et notre approche revient à centrer ces maximums (minimums). Nous n'aurons donc toujours *a priori* qu'un seul extremum.

Conventions

Dans l'analyse temporelle statique, nous aurons très souvent le cas où un signal possède n agresseurs, mais seulement m sont susceptibles de commuter en même temps que le signal victime. Par exemple, un signal qui diaphone avec un signal fixé de manière permanente à "0" (signal de *reset*, de *test*, ...) ne sera jamais perturbé par celui-ci. De même si nous avons la certitude que deux signaux ne peuvent pas commuter en même temps. Nous utiliserons par la

suite les termes suivants :

Définition 3

Un agresseur **actif** est un signal qui **est susceptible** d'influencer le comportement temporel d'un signal victime par l'intermédiaire d'une capacité de couplage. Un agresseur qui ne peut **jamais** influencer le comportement temporel d'un signal victime est appelé un agresseur **statique**.

Définition 4

Nous appellerons un **contexte** d'agression d'un signal la répartition de ses agresseurs en deux catégories, les agresseurs actifs et les agresseurs statiques.

Cas de deux agresseurs

Considérons le montage de la figure 2.16. Dans ce montage, le signal vic représente un fil de $100\mu m$ de long, agressé sur $50\mu m$ par le signal Ag1, et sur $50\mu m$ par le signal Ag2. Le signal Ag1 est contrôlé par une porte trois fois plus grosse que le signal victime. Les temps de propa-

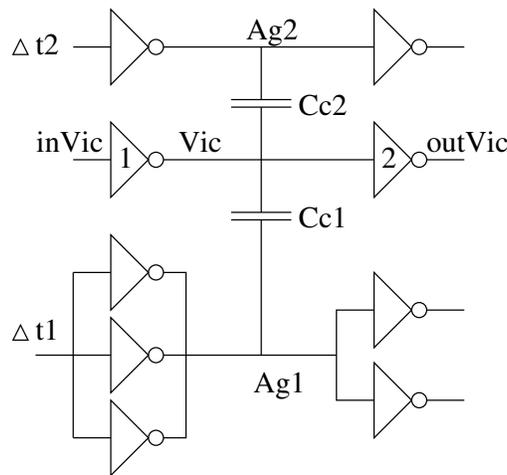


FIG. 2.16 – Signal vic agressé par deux signaux.

gation entre inVic et Vic et entre Vic et outVic dépendent donc de deux paramètres : Δt_1 et Δt_2 . Nous avons représenté sur la figure 2.17 le temps de propagation de la porte 1, et sur la figure 2.18 le temps de propagation de la porte 2. Pour la porte 1, nous relevons un temps de propagation maximum de $55ps$, qui est atteint lorsque Δt_1 vaut $-10ps$ et Δt_2 vaut $-10ps$, alors que le temps nominal, c'est à dire sans l'influence des couplages, valait $31ps$. Pour la porte 2, nous relevons un temps de propagation maximum de $36ps$ pour $\Delta t_1 = +10ps$ et $\Delta t_2 = +20ps$ par rapport à un temps nominal de $23ps$. Dans le premier cas, nous avons eu une variation de

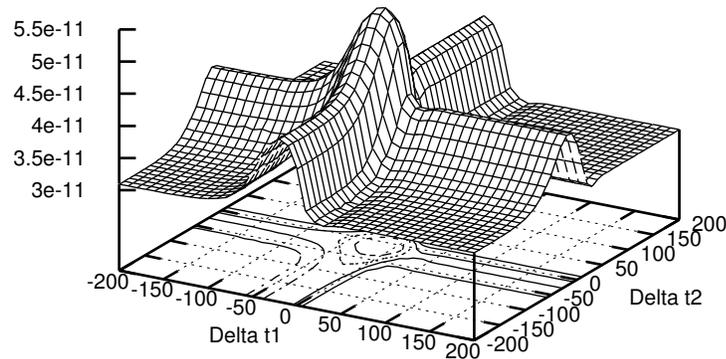


FIG. 2.17 – Temps de propagation maximum de la porte 1 (fig 2.16) en fonction des instants de commutation relatifs de deux agresseurs. Abscisses : Δt entre l’agresseur et la victime (en picosecondes), Ordonnée : Temps de propagation (en secondes).

temps de propagation d’un facteur 1,8, et pour le second cas 1,6. Les temps de propagation relevés lors de la commutation simultanée des agresseurs ($\Delta t = 0$) valent respectivement $52ps$ et $26ps$.

Pour obtenir les figures 2.17 et 2.18, nous avons fait varier les paramètres Δt_1 et Δt_2 entre $-200ps$ et $+200ps$ par pas de $10ps$. Cela représente 1600 simulations complètes. En procédant par approximations successives, nous obtenons le même résultat en seulement 3 itérations, ce qui représente 240 simulations, mais seulement pour le calcul d’un des deux temps de propagation. En effet, les conditions d’agression qui donnent la valeur extrême de temps de propagation de la porte 1 ($\Delta t_1 = -10ps$ et $\Delta t_2 = -10ps$) ne sont pas les mêmes que celles qui donnent la valeur extrême de temps de propagation sur la porte 2 ($\Delta t_1 = +10ps$ et $\Delta t_2 = +20ps$). Au final, nous avons donc effectué réellement 480 simulations par la méthode de l’approximation successive pour obtenir le même résultat.

Cas général de plusieurs agresseurs

La généralisation de ces phénomènes à n agresseurs ne change pas la nature des observations réalisées dans le cas de deux agresseurs. La seule information supplémentaire que nous pouvons apporter est qu’à longueur de ligne égale, et à valeur totale de couplage constante sur

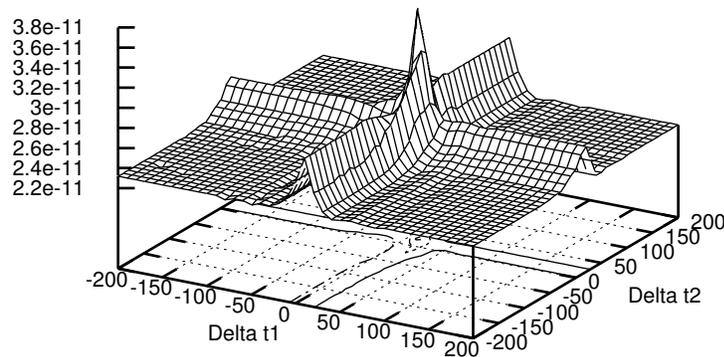


FIG. 2.18 – Temps de propagation maximum de la porte 2 (fig 2.16) en fonction des instants de commutation relatifs de deux agresseurs. Abscisses : Δt entre l'agresseur et la victime (en picosecondes), Ordonnée : Temps de propagation (en secondes).

le signal victime, plus le nombre d'agresseurs est grand, plus l'effet du couplage sera grand. Reprenons encore une fois le circuit 2.9. La valeur de la capacité de couplage entre les deux signaux vaut approximativement $10fF$. Si nous avons un second agresseur, chaque agresseur n'aura plus que $5fF$ de capacité de couplage, et aura donc individuellement moins d'influence. Par contre, une agression simultanée des deux agresseurs aura plus d'influence. Nous avons relevé dans le tableau 2.2 les temps de propagation de la porte 1 pour un nombre variable d'agresseurs.

Nous observons une légère augmentation du temps de propagation maximum (de l'ordre de 9% dans ce cas), alors qu'individuellement chaque agresseur a de moins en moins d'influence. Pour souligner une dernière fois les difficultés à obtenir les temps de propagation, dans le cas de 10 agresseurs, si nous avons simulé tous les cas possibles des Δt entre $-200pS$ et $+200pS$ par pas de $10pS$, cela aurait représenté $40^{10} \approx 10^{16}$ simulations. Par la méthode des approximations successives, nous n'avons effectué que 1200 simulations pour obtenir le même résultat.

L'obtention des temps de propagation extrêmes sous la contrainte de plusieurs agresseurs est une opération qui se révèle finalement assez délicate. Nous avons également vu qu'en cas de forts couplages, l'obtention du temps de propagation (telle que nous le définissons par

Nombre d'agresseurs	Capacité de couplage par agresseur	Temps de propagation sous l'influence d'un seul agresseur	Temps de propagation sous l'influence de tous les agresseurs
1	$10fF$	$36.2pS$	$36.2pS$
2	$5fF$	$32.8pS$	$38.7pS$
3	$3.33fF$	$31.4pS$	$39.3pS$
5	$2fF$	$30.2pS$	$39.7pS$
7	$1,43fF$	$29.6pS$	$39.9pS$
10	$1fF$	$29.2pS$	$40.0pS$

TAB. 2.2 – Valeurs du temps de propagation maximum d'une porte en fonction du nombre d'agresseurs, la totalité des capacités de couplage des agresseurs étant fixé à la valeur constante de $10fF$.

l'instant de passage du signal à $v_{dd}/2$) peut poser également des problèmes de définition. Nous allons maintenant étudier la propagation d'une transition à travers une série de portes en présence d'agresseurs, et montrer qu'elle pose un problème de précision.

2.3.4 Propagation d'une transition à travers plusieurs portes.

Dans l'analyse du comportement temporel d'un circuit, le calcul d'un délai entre deux points du circuit consiste à calculer une somme de délais élémentaires. Dès lors, il faut s'interroger sur les conséquences d'une erreur sur l'un de ces délais élémentaires par rapport au délai total recherché. La cause de cette erreur peut avoir plusieurs origines. Elle peut être due à des hypothèses trop simplificatrices sur les modèles ou à une mauvaise précision des calculs. Il serait faux de considérer le simulateur électrique, quel qu'il soit, comme étant "le cas idéal où il n'y a pas d'erreur" car le problème se situe avant. Nous ne cherchons pas à étudier le comportement de circuits idéaux, mais de circuits bien réels. Or, ceux-ci subissent des variations de processus. Deux circuits ne seront donc jamais identiques. Pour réaliser une analyse temporelle, qui s'effectue avant que le circuit ne soit fabriqué, nous ne disposons que du dessin des masques pour obtenir une représentation électrique équivalente du circuit. Cette différence entre la représentation idéale d'un circuit et la réalité a donc un impact sur la précision. Ensuite, la phase d'extraction ne peut se faire sur de gros circuits qu'au prix de simplifications (voir la section 3.1) qui vont forcément avoir une répercussion que l'on souhaite la plus faible possible au niveau du comportement temporel. Ensuite, le simulateur électrique qui permet de

caractériser précisément des petites portions de circuit est sensible à la précision limitée des ordinateurs. Finalement, l'analyseur temporel qui a la contrainte de gérer de gros volumes de données, doit recourir à des simplifications, qui vont introduire des erreurs de calcul. Nous voyons donc qu'entre la réalité du circuit fondu, et les résultats donnés par l'analyseur temporel, les causes d'erreurs potentielles sont nombreuses.

Sur la figure 2.19, nous supposons qu'une erreur est commise lors du calcul du second délai élémentaire lors du calcul du temps de propagation de la chaîne.

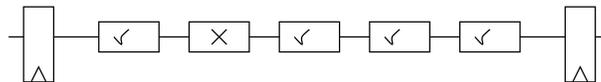


FIG. 2.19 – Erreur sur un délai élémentaire dans une chaîne.

Une erreur sur une porte modifie sensiblement son temps de propagation et son front de sortie. Cette seconde erreur se traduira par une légère variation du temps de propagation sur les portes de charges. L'erreur initiale est très vite atténuée. Ces erreurs commises interviendront directement dans l'évaluation du temps de propagation de la chaîne complète, sans réduction, ni amplification. On peut dire qu'une erreur commise sur un délai élémentaire se transmet avec un coefficient 1 jusqu'à la fin de la chaîne. Ceci est dû au fait que l'on peut calculer les délais élémentaires isolément les uns des autres. Au final, l'erreur est à ramener sur la longueur totale de la chaîne.

Lorsque nous voulons prendre en compte les capacités de couplage, le calcul du délai élémentaire de chaque porte fait intervenir l'activité sur les autres signaux. L'erreur commise sur la seconde porte a pour effet de décaler l'instant de toutes les transitions sur les signaux en aval de cette porte. Or, le délai de chaque porte dépend grandement de l'activité relative entre signaux. Comme l'instant de transition de chaque porte est modifié, le temps de propagation de chaque porte est modifié. Ce phénomène s'amplifie de portes en portes. Par exemple, prenons la simulation du schéma représenté sur la figure 2.20. Les 4 lignes couplées font $100\mu m$ de long chacune. Supposons que sur le montage réel, les instants de commutation des entrées $T1$, $T2$, $T3$ et $T4$ valent respectivement $+5pS$, $+25pS$, $+46pS$ et $+66pS$ par rapport à un instant donné $T0$. Nous représentons une erreur de calcul en amont du signal in par un changement de son instant de commutation d'une valeur ΔE par rapport à $T0$. Observons la variation du temps de propagation entre X et E en fonction de ΔE , sur la figure 2.21.

Nous constatons qu'une diminution de ΔE de $25pS$, qui représente une erreur antérieure,

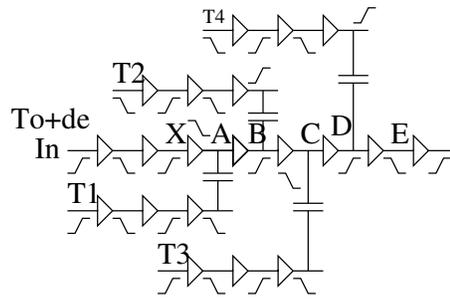


FIG. 2.20 – Circuit simulé

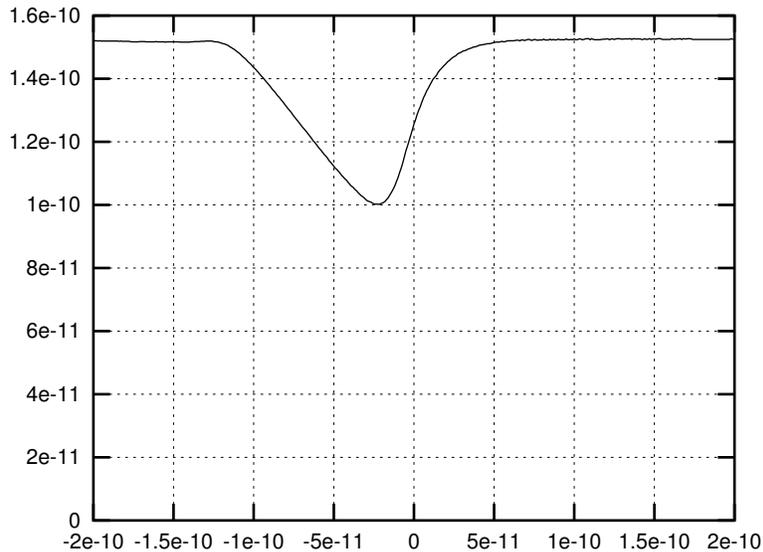


FIG. 2.21 – Temps de propagation entre X et E en fonction de la variation de l’instant de commutation de l’entrée X (ΔE). Abscisse : ΔE (secondes), Ordonnée : temps de propagation (secondes)

suffit à faire passer le temps de propagation entre X et E de $126pS$ à $100pS$. Nous avons donc sur cet exemple une erreur supplémentaire de $26pS$ sur les $25pS$ initiales. Nous avons donc fait plus que doubler l’erreur initiale. Bien sûr, nous avons prévu dans cet exemple d’avoir un phénomène tout à fait significatif. Dans la pratique, ce phénomène est moins important, mais il n’est pas négligeable.

Les effets de couplage capacitifs entraînent donc une très grande sensibilité aux erreurs, ayant comme conséquence que les instants de transition ne sont plus connus précisément. C’est la seconde raison pour laquelle nous préférons nous intéresser aux valeurs extrêmes prises par les temps de propagation, plutôt qu’aux valeurs ponctuelles.

2.4 Bruit de couplage capacitif

Le but ici n'est pas d'étudier en détail les problèmes liés au bruit de commutation, mais de les mentionner. Il s'agit d'une activité de recherche à part entière, sur laquelle une thèse a été soutenue dans notre laboratoire [Iip02].

Reprenons le circuit de la figure 2.9. Cette fois-ci, le signal `sigvic` est stable à la valeur 0, et le signal `sigagr` commute, comme le montre la figure 2.22. La variation de tension qui survient

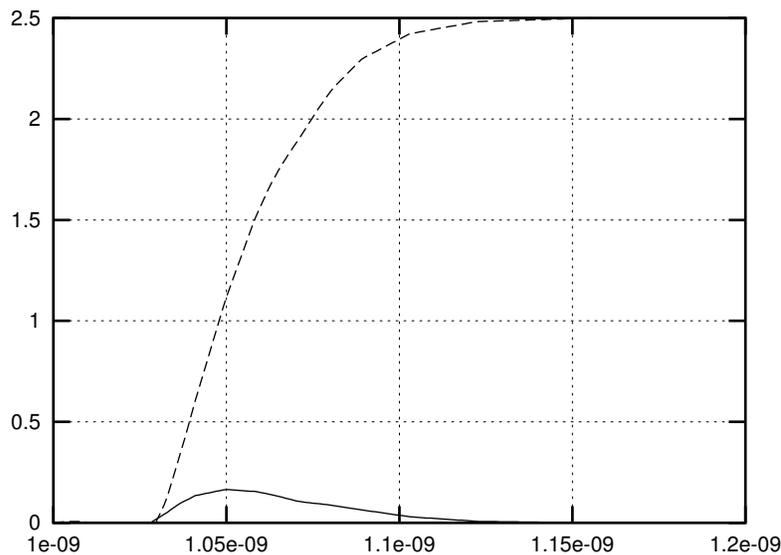


FIG. 2.22 – Bruit de couplage capacitif. En trait continu : le signal `sigvic`, et en pointillé, le signal `sigagr`. Abscisse : temps (secondes), Ordonnée : tension (volts).

sur le signal victime normalement stable lorsqu'un ou plusieurs de ses agresseurs commutent est appelé *bruit de commutation*. La hauteur du pic est liée à la porte qui contrôle le signal victime, les portes qui contrôlent les agresseurs, et à la valeur de la capacité de couplage entre ces signaux. Ce pic de tension parasite peut devenir un réel problème selon la nature du signal affecté ou la hauteur qu'il atteint. Dans le cas général, c'est à dire pour les portes duales CMOS et lorsque la hauteur du bruit est sensiblement inférieure au seuil de commutation de la porte, cela n'introduit qu'une légère surconsommation électrique. Soulignons à cette occasion que la consommation des circuits est aujourd'hui problématique, puisqu'un circuit comme le Pentium 4 d'Intel à 2Ghz consomme plus de 75 Watt en 1,8v ([Int01]). Lorsque ce pic de tension atteint la tension de seuil de la porte, des transitions parasites peuvent se propager. Ces transitions parasites qui se propagent peuvent à leur tour engendrer de nouvelles agressions qui pourront

à leur tour créer de nouvelles transitions parasites. Lorsque ces phénomènes se produisent en boucle, le circuit devient inutilisable. De même, lorsque ce phénomène transitoire est capturé par une bascule, il se traduit par un erreur permanente, et le circuit devient également inutilisable. Pour les signaux haute impédance, ce problème est plus grave car le signal conserve la valeur du bruit. Si plusieurs agressions individuellement très faibles surviennent successivement au cours d'un cycle, la variation totale de tension peut alors être très importante.

Il existe donc un risque de dysfonctionnement fonctionnel du circuit qui n'est plus négligeable pour les technologies submicroniques. Ce critère doit être intégré dans les outils de routage, en espaçant les fils sur les signaux sensibles. Ensuite, des outils de vérification calculant la hauteur maximum que peut atteindre ce bruit sur chaque signal deviennent indispensables.

2.5 Influence combinée des résistances et fils d'interconnexion.

2.5.1 Effets temporels

Dans les interconnexions des circuits submicroniques, les résistances de lignes et les capacités de couplage entre signaux sont évidemment deux phénomènes présents simultanément, et distribués. Nous avons vu dans le chapitre 2.1.2 que ces éléments distribués étaient discrétisés lors de la phase d'extraction. Entre deux signaux, il peut y avoir alors plusieurs capacités de couplage. Nous allons porter notre attention sur deux principaux points : l'influence des résistances sur deux lignes couplées sur toute leur longueur, dans le cas de deux fils parallèles. Il s'agit de *l'effet distribué*. Et l'influence des positions ponctuelles des capacités de couplage, dans le cas de fils qui se croisent. Il s'agit de *l'effet local*.

Effet distribué

Nous remplaçons les deux fils de $100\mu m$ de long du circuit représenté sur la figure 2.9 par deux fils de $1000\mu m$ de long en métal de premier niveau sur la figure 2.23, afin d'observer l'influence des résistances. Ces deux fils de $1000\mu m$ de long sont découpés en 1000 paires de segments de $1\mu m$ de long, incluant la résistance et les capacités de couplage, comme cela a été précisé dans la section 2.1.2. Chaque ligne a une résistance totale de 625Ω . Nous relevons les temps de propagation obtenus avec et sans prise en compte des résistances sur la figure 2.24, lorsque les signaux commutent en sens contraire, avec un écart Δt .

Nous constatons dans ce cas que les résistances ont une influence qui dépend de la porte considérée. Dans le cas de la porte 1, cet effet est une diminution (approximativement constante

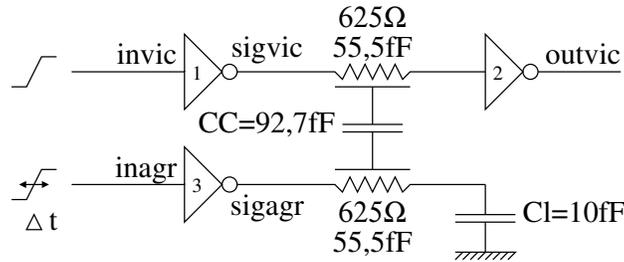


FIG. 2.23 – Influence temporelle d’un agresseur sur un signal victime.

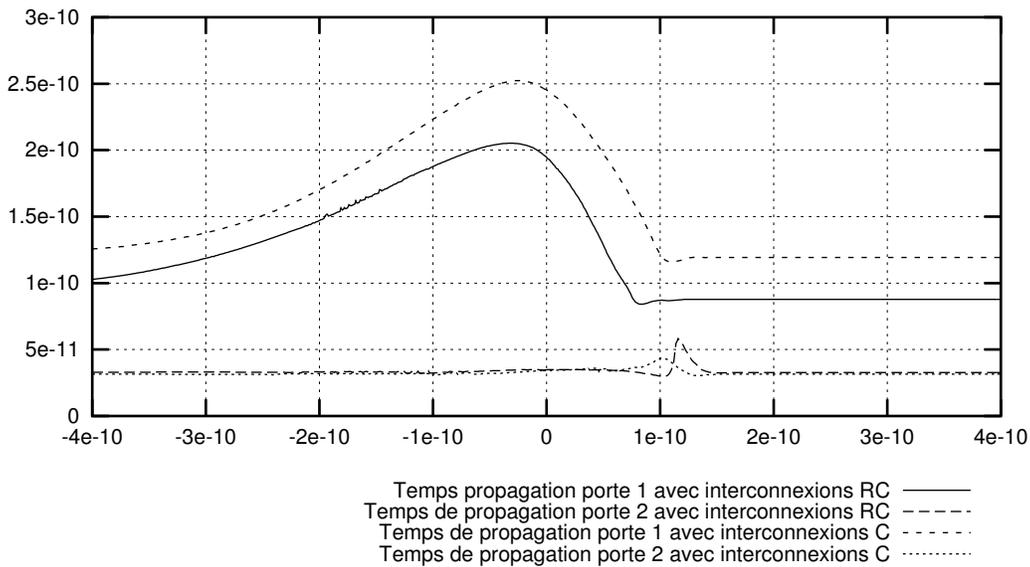


FIG. 2.24 – Temps de propagation avec et sans résistances de deux lignes couplées. Abscisse : Δt (secondes), Ordonnée : Temps de propagation (secondes).

par rapport à Δt) du temps de propagation. Nous retrouvons le phénomène de masquage de la charge de la ligne par sa propre résistance, vue dans le chapitre 2.2.1. La valeur Δt pour laquelle la valeur maximum est atteinte n’a pas changé. Pour la seconde porte, cet effet est plus complexe. Lorsqu’il n’y a pas de transitions simultanées, la résistance de ligne a une influence extrêmement faible. Par contre, lorsque les transitions ont lieu en même temps, nous observons que la prise en compte des résistances des fils se traduit par une augmentation de la valeur maximum du temps de propagation, et du changement de la valeur Δt où se produit ce maximum.

Effet local

Considérons le circuit de la figure 2.25. La longueur de chaque fil est égale à $500\mu m$, dont on considère que $100\mu m$ sont couplés entre les deux lignes. Nous étudions ici l'effet de la position

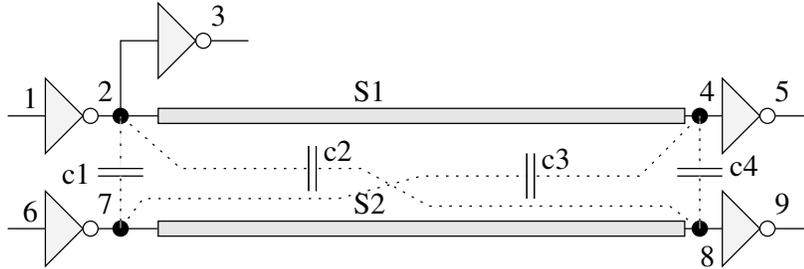


FIG. 2.25 – Etude de l'influence de la position d'une capacité de couplage.

du couplage, entre le début ou la fin de chacun des fils, ce qui nous donne 4 possibilités. Nous avons tracé sur la figure 2.26 le temps de propagation entre le nœud 1 et le nœud 2, pour les quatre positions du couplage. Nous avons refait cette mesure sur la figure 2.27 pour le temps de propagation entre les nœuds 1 et 4. Nous constatons sur ces deux exemples que la posi-

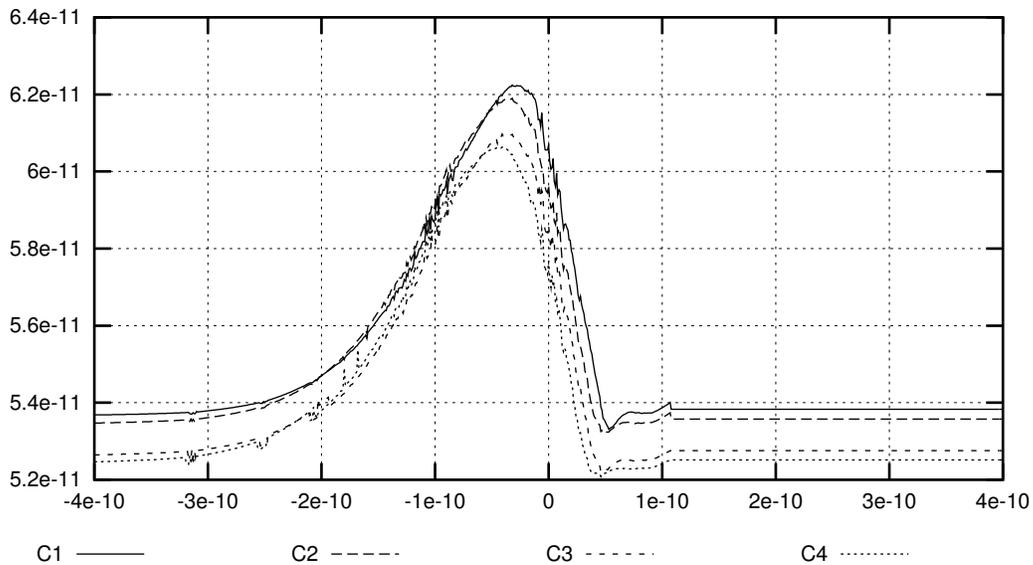


FIG. 2.26 – Temps de propagation entre le nœud 1 et le nœud 2 de la figure 2.25 pour différentes positions du couplage avec prise en compte des effets résistifs. Abscisse : Δt (secondes), Ordonnée : Temps de propagation (secondes).

tion des capacités de couplage joue un rôle non négligeable sur les temps de propagation (ici,

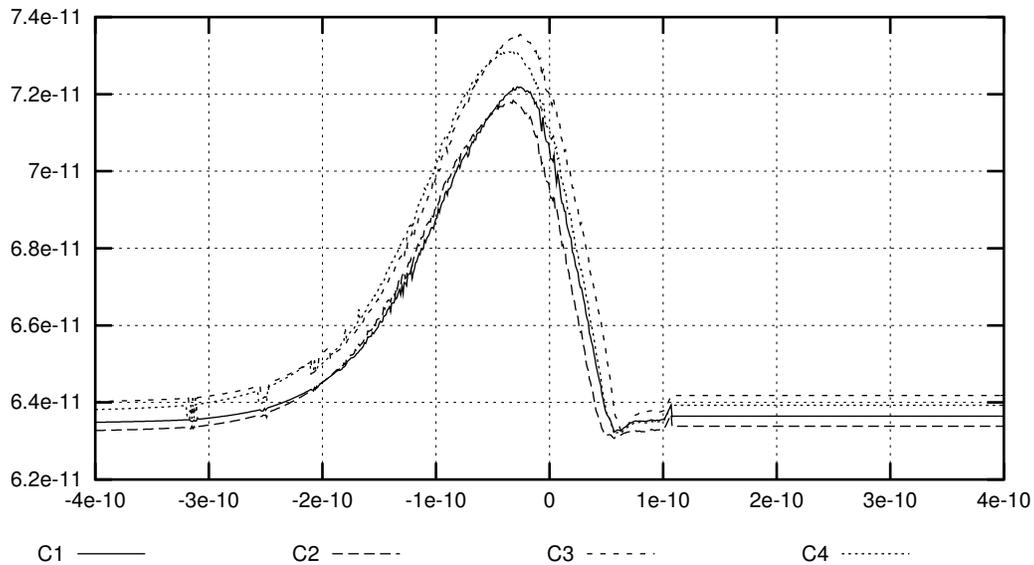


FIG. 2.27 – Temps de propagation entre le nœud 1 et le nœud 4 de la figure 2.25 pour différentes positions du couplage avec prise en compte des effets résistifs. Abscisse : Δt (secondes), Ordonnée : Temps de propagation (secondes).

jusqu'à 4%). De plus, nous observons que la valeur de Δt qui donne le temps de propagation extrême n'est pas identique selon le nœud observé. Cela veut dire qu'il y a **sur un signal victime autant de configurations d'agresseurs qu'il y a de nœuds sur lesquels calculer les temps de propagation.**

Intéressons nous au temps de propagation entre le nœud 1 et le nœud 2. Ce temps est plus grand lorsque les couplages sur la victime sont proches du nœud observé (cas de C1 et C2), que lorsqu'ils sont en fin de ligne (cas de C3 et C4). Dans ces deux configurations, le temps de propagation est plus grand lorsque le couplage est proche de l'émetteur du signal agresseur (cas de C1 et C3). Ces résultats se retrouvent sur le temps de propagation entre le nœud 1 et le nœud 4. Les temps de propagation maximum sont obtenus lorsque le couplage est proche du nœud observé (cas C3 et C4), et moins important lorsque le couplage a lieu plus en amont (cas C1 et C2). Et encore une fois, pour ces deux configurations, le temps de propagation le plus grand a lieu lorsque le couplage sur l'agresseur est proche de l'émetteur (cas C3 et C1).

En résumé, les conditions d'agression sur les temps de propagation minimum ou maximum sont différentes selon le nœud considéré sur le réseau RC. L'influence est d'autant plus significative que le couplage actif a lieu près du nœud observé, et que le couplage de l'agresseur est

proche de son émetteur. La position du couplage sur le signal victime a, dans le cas observé ici, plus d'importance que la position du couplage sur le signal agresseur.

2.5.2 Influence sur le bruit de commutation

Les résistances d'interconnexion ont une influence sur la valeur maximum du bruit de diaphonie. Nous avons tracé sur les figures 2.28 et 2.29 les bruits dus au couplage capacitif du circuit de la figure 2.25. Dans cette simulation, le signal S1 est stable, et le signal S2 a une transition montante. Les tensions sur la figure 2.28 sont mesurées sur le nœud 2 et celles de la figure 2.29 sont mesurées sur le nœud 4. Nous constatons que le nœud 4 est bien plus sensible au

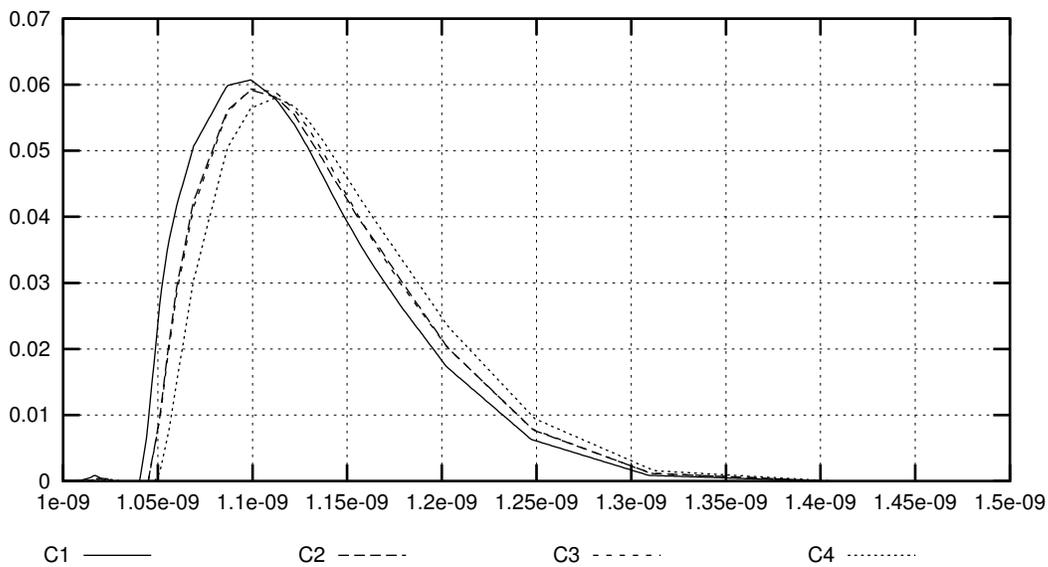


FIG. 2.28 – Bruit de couplage capacitif sur le nœud 2 avec prise en compte des effets résistifs. Abscisse : Temps (Secondes), Ordonnée : Tension (Volts).

bruit que le nœud 2, du fait de la résistance de la ligne. Dans ce cas, cela se traduit par une augmentation de la hauteur du bruit de près d'un facteur 2. Sur le nœud 4, nous constatons que la position du couplage sur le signal victime a plus d'importance (cas C3 et C4) que la position du couplage sur le signal agresseur (cas C1 et C2).

2.5.3 Conclusion

Nous avons mis en évidence avec ces quelques simulations que les résistances d'interconnexion modifient sensiblement l'influence des capacités de couplage. La position de ces dernières a plus d'influence sur le signal victime que sur le signal agresseur, que ce soit d'un point

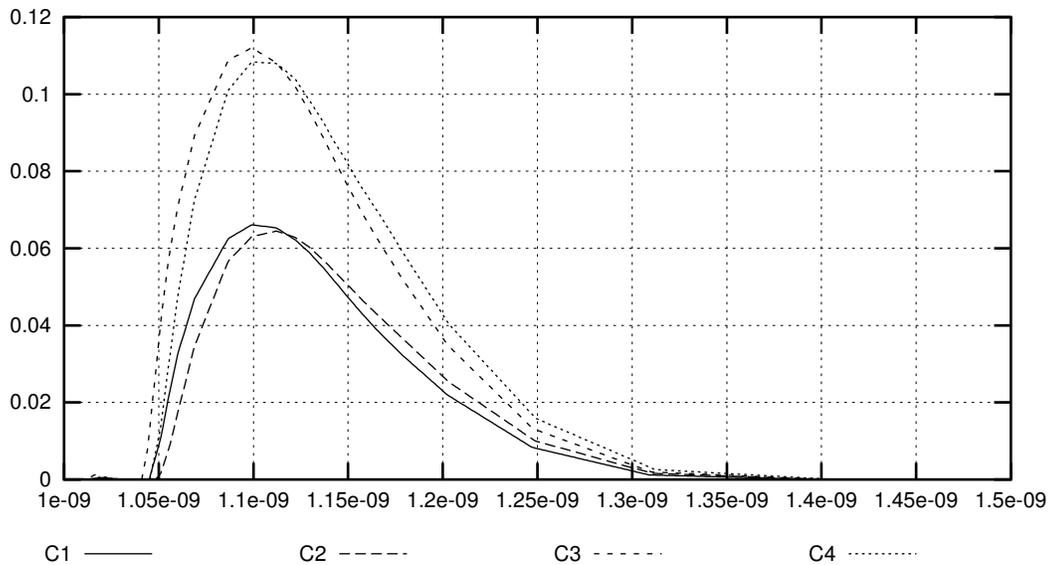


FIG. 2.29 – Bruit de couplage capacitif sur le nœud 4 avec prise en compte des effets résistifs. Abscisse : Temps (Secondes), Ordonnée : Tension (Volts).

de vue temps ou d'un point de vue bruit. Nous avons également montré qu'il y a autant de configurations d'agresseurs donnant un temps de propagation extrême qu'il y a de nœuds à observer.

2.6 Contraintes sur les volumes de données

2.6.1 La représentation des circuits en mémoire

Les outils d'analyse temporelle pour les technologies profondément submicroniques de circuits constitués de plusieurs millions de transistors doivent faire face à un volume de données très important. Ces données représentent, pour la plus grande partie, des informations parasites de routage. L'information concernant les parties "actives" du circuit occupe alors en mémoire une place mineure.

Considérons à titre d'exemple le circuit Noé [Des98]. Ce circuit comporte près de 1,3 millions (1 286 807) de transistors, en 3 niveaux de routage. Les informations extraites par Lynx, donc prenant en compte uniquement les résistances d'interconnexions, sont reportées dans le tableau 2.3.

Les tailles occupées en mémoire correspondent à la vue dans la structure de donnée Alliance. Elle occupe près de 1,2Go. Il est intéressant de constater que la place occupée par les signaux, qui comprennent les résistances et les capacités d'interconnexion, occupe 82% de la

Eléments	Nombre d'éléments extraits avec Lynx	Taille en mémoire (octets)
Transistors	1 286 807	226 478 032
Signaux	648 494	1 024 728 432
Résistances	12 199 148	

TAB. 2.3 – Résultat de l'extraction du circuit Noé.

mémoire. Les éléments actifs (les transistors) n'occupent donc finalement que 18% de la mémoire. De plus, il faut ajouter à ces chiffres les noms des signaux, et plusieurs autres informations. L'occupation en mémoire est donc sensiblement plus importante, puisqu'on dépasse alors 1,5Go de mémoire pour représenter ce circuit. Aujourd'hui, avec des machines basées sur une architecture 32 bits, il n'est pas possible de stocker en mémoire un circuit qui aurait plus de 3 millions de transistors. Un circuit (processeur, processeur graphique, ...) de taille un peu conséquente atteint facilement les 30 millions de transistors. L'utilisation de machines basées sur une architecture 64 bits ne permet pas de contourner cette limite, car les structures de données Alliance comportent plus de 50% de pointeurs. Pour représenter un circuit de 30 millions de transistors sur une machine ayant une architecture 64bits, il faudrait qu'elle possède au moins 45Go de mémoire. Ce chiffre est énorme.

Bien sûr, ces valeurs nécessaires pour stocker en mémoire le circuit ne reflètent pas la consommation mémoire réellement utilisée pour réaliser l'analyse temporelle : celle-ci est en effet considérablement plus importante. L'abstraction fonctionnelle ajoute de nombreuses informations à la description du circuit, lors de la construction des portes et de leur fonctionnalité. Ensuite, le calcul des délais consomme également un peu de mémoire, et le graphe de causalité reste un objet assez volumineux. En pratique, avec 4 Go, il est donc impossible de réaliser l'analyse temporelle d'un circuit comportant plus de quelques centaines de milliers de transistors. Sur ce même circuit, l'extracteur Space fournit 5,4 millions (5 435 720) de capacités de couplage.

Considérons également les résultats obtenus par un extracteur commercial largement utilisé : XCalibre [Men97]. Les blocs dont les résultats d'extraction sont reportés dans le tableau 2.4 proviennent de la société Bull. Deux extractions ont été réalisées : la première en ne prenant en compte que les résistances d'interconnexion, et la seconde en ne prenant en compte que les capacités de couplage.

En conclusion, l'augmentation conjointe du nombre de transistors et du nombre d'informations parasites extraites par transistor est aujourd'hui telle que les approches à plat pour

Bloc	Nombre de transistors	Extraction		Extraction
		Avec Résistances #R	#C masse	Avec Capacités de Couplage #C de diaphonie
blk2	3 296	42 230	34 474	15 507
blk5	12 586	121 406	99 383	63 241
dnp3	11 554	125 600	109 322	77 100

TAB. 2.4 – Comparaison en nombre d’éléments résultant de diverses extractions pour les circuits de BULL.

l’analyse temporelle ne sont pas utilisables. De plus, les capacités diaphonantes ne permettent plus une réduction des réseaux d’interconnexion pendant l’extraction.

2.6.2 L’approche hiérarchique

Revenons au cas où seules les résistances seraient prises en compte. Pour résoudre le problème du volume des données, on utilise la hiérarchie de conception : l’analyse temporelle est tout d’abord réalisée sur les blocs placés et routés au niveau le plus bas de la hiérarchie. Lorsqu’on instancie ce bloc dans un bloc de niveau supérieur, seules les portes connectées sur l’extérieur du bloc de base voient leur temps de propagation changer. Les portes enfouies dans le bloc ont leur temps de propagation qui reste constant. Cette observation fondamentale permet de pouvoir construire le graphe de causalité de ce bloc de base, et de pouvoir le réutiliser lorsque ce bloc sera instancié. Seuls les délais des portes externes devront être recalculés. En pratique, il y en a rarement plus de quelques centaines. De plus, cette approche est récursive. C’est cette solution qu’intègre HiTas. Les avantages sont alors immédiats et nombreux. Premièrement, l’abstraction fonctionnelle, dont la complexité augmente plus vite que le nombre de transistors, devient plus rapide car on ne traite à chaque fois qu’un petit nombre de transistors. Ensuite, le volume des données en mémoire reste très limité. Par exemple, le circuit Noé a été analysé sur une station ne possédant que 256 Mo de mémoire. Finalement, en remarquant que la recherche des chemins critiques, problème fondamentalement à plat, ne fait intervenir à chaque fois qu’un petit nombre de blocs, il est possible en utilisant un mécanisme de mémoire cache de contourner cette difficulté. Tous ces aspects sont présentés dans [Dio98].

L’hypothèse fondamentale qui a permis cette solution repose sur deux hypothèses :

1. Les délais élémentaires internes aux blocs sont figés. C’est une caractéristique particuliè-

rement liée à la technologie C-MOS.

2. La construction des chemins critiques arrivant ou partant d'un point mémoire ne fait intervenir qu'un petit nombre de blocs.

Lors de la prise en compte des capacités de couplage, le premier point n'est plus vérifié. Lorsqu'un bloc voit passer au-dessus de lui (routage overcell), ou même parfois à travers lui dans le cas de l'utilisation d'une transparence, un fil de routage externe à ce bloc, celui-ci va diaphoner avec les signaux appartenant au bloc. Les conséquences sont nombreuses. Premièrement, lorsqu'un bloc est instancié plusieurs fois, il doit être extrait autant de fois, en prenant en compte son contexte d'instanciation. Ensuite, les délais élémentaires à l'intérieur du bloc ne seront plus constants, car ils vont dépendre de l'activité de ce signal de routage externe. Cela veut dire que même si la structure du graphe de causalité reste la même sur ces blocs de base, il faut être en mesure de pouvoir réévaluer les valeurs des arcs. Or, cette évaluation ne peut se faire qu'en ayant les caractéristiques de tous les fils de routage passant au-dessus des blocs. De même, lorsqu'un fil connecte deux blocs, le temps de propagation entre l'émetteur et le récepteur dépend de l'activité des autres blocs traversés. Pour déterminer un temps de propagation *interne* à un bloc, il faut donc être en mesure de connaître les caractéristiques électriques de tous les agresseurs, internes ou externes à ce bloc.

En conclusion, il n'y a donc plus de découplage d'information entre la description du circuit et le graphe temporel, et il n'y a plus de notion de localité pour déterminer les délais élémentaires. Par contre, la phase de désassemblage et la recherche des chemins critiques à partir du graphe de causalité ne sont pas modifiées par la prise en compte ou non des capacités de couplage.

2.7 Conclusion

La principale conclusion que nous pouvons tirer de cette étude est que la détermination exacte du temps de propagation d'une porte n'a plus de signification dans l'analyse temporelle, trop de configuration d'agression étant à prendre en compte pour chaque signal, et trop d'erreurs pouvant intervenir en différents endroits. Seules les approches *pires cas*, c'est à dire dans lesquelles nous déterminons les temps de propagation minimum et maximum, sont possibles. Nous verrons cependant par la suite que la détermination de ces temps de propagation extrêmes reste difficile à obtenir avec précision.

Chapitre 3

Vérification post layout - Etat de l'Art

La méthode d'analyse temporelle du circuit dépend étroitement de la façon dont ce dernier est extrait. Nous allons présenter dans le paragraphe 3.1 les solutions existantes pour réaliser cette opération, puis, nous présenterons les outils industriels d'analyse temporelle dans le paragraphe 3.2. Nous verrons ensuite dans le paragraphe 3.3 les différentes solutions publiées pour réaliser le calcul des délais élémentaires en présence des couplages capacitifs. Enfin nous présenterons en détail le calcul des délais élémentaires tels qu'ils sont réalisés dans HiTas dans le paragraphe 3.4.

3.1 Extraction des circuits

3.1.1 Introduction

La vérification temporelle d'un circuit intégré s'effectue à partir de la description électrique de ce circuit. Cette description est obtenue à l'aide d'un outil appelé **extracteur**. L'étape d'extraction ne peut être étudiée séparément de l'analyse temporelle car de cette première dépend la précision des résultats, et surtout la méthode utilisée par la suite.

3.1.2 Extraction de circuits dans le cadre de l'analyse temporelle.

Pour réaliser une analyse temporelle au niveau transistor, les informations principales requises sont les caractéristiques des transistors, les connexions entre les transistors et les capacités de routage. Le problème de l'analyse temporelle se trouve grandement simplifié si l'on n'extrait que les éléments parasites qui ont une réelle influence sur le comportement temporel du circuit. C'est pourquoi les informations parasites extraites ont évolué avec les différentes technologies.

Jusqu'aux technologies $0,35\mu m$, il était possible de limiter l'extraction des interconnexions à de simples capacités de routage. La complexité de l'information parasite extraite était alors mineure, puisque chaque signal ne comportait au plus qu'une valeur de capacité parasite. En considérant que le nombre de signaux dans un circuit est égal au nombre de transistors, on constate que la quantité d'information pour représenter ces capacités de routage est faible par rapport à la description complète du circuit.

Pour les technologies comprises entre $0,35\mu m$ et $0,18\mu m$, les phénomènes résistifs des signaux ne peuvent plus être ignorés pour une analyse temporelle. Cette nouvelle contrainte a fait apparaître un problème au niveau des extracteurs : gérer la masse des informations extraites. En effet, chaque signal sera extrait sous la forme de plusieurs dizaines de résistances et de capacités. Cela entraîne une augmentation de plusieurs ordres de grandeurs des informations parasites extraites, qui occupent maintenant plus de place en mémoire que les éléments actifs. Les logiciels d'extraction ont donc dû recourir à de nombreuses techniques pour réduire ce volume d'informations. La plus simple consiste à n'extraire les résistances d'interconnexion que sur les signaux où elles ont une réelle influence au niveau temporel. Une autre solution consiste à représenter chaque signal sous une forme réduite, en ne conservant par exemple qu'une fonction de transfert. Cette solution, généralement complexe, implique plus que l'application d'un algorithme de réduction de réseau : il faut également un algorithme qui détermine le sens de propagation dans le signal. Si ce point ne pose pas de problèmes dans le cas de cellules précaractérisées où le sens des connecteurs peut être spécifié, cela est beaucoup plus complexe dans le cas de circuits optimisés au niveau transistor.

A partir des technologies $0,18\mu m$, les capacités de couplage entre signaux commencent à avoir une influence au niveau temporel. Le caractère dynamique de leur influence ne permet plus l'utilisation directe des algorithmes de réduction des réseaux d'interconnexions parasites évoqués précédemment. En supposant que l'on trouve une représentation réduite permettant de conserver l'information de couplage entre signaux, le volume des informations existerait toujours car il faudrait conserver une information de couplage entre chaque paire de signaux voisins. De plus, le phénomène de couplage étant indépendant de la hiérarchie de conception du circuit, des problèmes aux frontières des instances se posent également.

3.1.3 Les formats de fichiers

Le rôle d'un extracteur est de déterminer une **représentation** électrique d'un circuit à partir du dessin de ses masques. Cette description électrique contient des éléments actifs tels que des

transistors, des diodes, des instances ou des cellules précaractérisées, et des éléments passifs tels que des résistances, des capacités, des inductances. Ces éléments passifs peuvent être présentés sous des formes équivalentes plus compactes, généralement des filtres d'ordre limité. Un problème important dans la représentation des capacités de diaphonie est leur description entre deux niveaux de hiérarchie, comme une capacité ayant un nœud dans une figure A, et l'autre nœud dans une figure B qui instancie la figure A.

Il existe plusieurs formats de fichiers possibles pour ces descriptions. Parmi les plus utilisés, nous trouvons :

- Le plus commun est le format Spice [Met90] [Men00], qui contient soit des cellules précaractérisées, soit des transistors, et la description des interconnexions. Ce format a cependant l'inconvénient d'être désordonné, puisque y sont décrits uniquement des éléments discrets, sans notion de topologie, ni de fonctionnalité. Il offre cependant une très grande souplesse d'utilisation, et est devenu LE standard.
- La description SPF [Cad93] pour *Standard Parasitic Format* est une alternative au format Spice proposée par Cadence. Elle est conçue pour être utilisée au niveau cellules précaractérisées, ou supérieur. La description SPF se décline en deux formats de fichier : le format DSPF, pour *Detailed SPF*, et le format RSPF, pour *Reduced SPF*. Ces formats de fichier sont compatibles avec le simulateur électrique Spice. Le premier contient une description au format Spice, dans laquelle le détail en éléments passifs de chaque interconnexion est regroupé. Le second représente chaque interconnexion par un modèle réduit à quelques éléments discrets déterminés par RICE [CNT91]. Cette description n'est pas conçue pour représenter les capacités de diaphonie.
- La description au format SPEF, spécifié par la norme IEEE 1481. Ce format de fichier n'est pas compatible avec le simulateur Spice, mais offre de nombreuses caractéristiques intéressantes : représentation des informations parasites sous la forme de résistances, capacités et inductances, représentation des capacités de couplage, possibilité d'avoir dans un même fichier les valeurs minimums, typiques et maximums des éléments, utilisation de la hiérarchie, simplification des interconnexions sous forme de filtres.

3.1.4 Les logiciels disponibles

L'extraction des circuits est actuellement un problème important, car il y a de nombreuses contraintes à prendre en compte pour obtenir une extraction adaptée à un problème donné. Ces contraintes dépendent de la finalité de l'extraction, et de la technologie visée. Les algorithmes utilisés pour déterminer les caractéristiques des transistors, les résistances d'interconnexion, les capacités de couplage ou toute autre information parasite sont globalement bien maîtrisés.

Le problème est leur utilisation pour des circuits de grande taille, car les temps de calcul et la consommation mémoire peuvent devenir extraordinairement grands. Ces outils ont donc recours à plusieurs techniques pour fournir des résultats pertinents dans des conditions réalistes. Ces techniques - on devrait plutôt parler d'*approximations* - font actuellement l'objet de nombreuses recherches au vu du nombre de publications sur ce sujet [Ber00].

Il existe plusieurs logiciels d'extraction, commerciaux ou académiques. Sans prétendre être exhaustif, voici les principales caractéristiques relatives au traitement des interconnexions de plusieurs extracteurs disponibles actuellement.

XCalibre, de Mentor Graphics

D'après les caractéristiques techniques disponibles [Men97], ce logiciel permet une extraction hiérarchique des circuits, avec extraction des résistances d'interconnexion et des capacités entre signaux. Ce logiciel est conçu - d'après Mentor Graphics - pour extraire des circuits de plus de 100 millions de transistors. Pour déterminer les capacités entre signaux, XCalibre détermine, parmi des structures représentatives du circuit donné, des équations de complexité arbitraire (l'utilisateur détermine la précision requise) à partir d'une résolution précise en 3 dimensions dans une phase appelée phase de calibrage. Une fois ces équations déterminées, le logiciel applique pour chaque structure du circuit la structure la plus similaire trouvée pendant la phase de calibrage et applique la ou les équations correspondantes. Une autre particularité de ce logiciel est de ne pas se fier uniquement à la hiérarchie du circuit pour déterminer les éléments parasites, car ceux-ci peuvent dépendre de leurs voisins. XCalibre transmet donc en interne des informations pour calculer les éléments parasites aux frontières des instances. Finalement, XCalibre intègre également un gestionnaire de base de données des informations extraites pour gérer efficacement les énormes volumes de données produits. XCalibre produit des fichiers au format Spice, SPF, SPEF et SDF.

Assura Parasitic Extractor, de Cadence

Cet extracteur [Cad01a] peut fonctionner au niveau transistor ou au niveau cellule. Il permet l'extraction des résistances d'interconnexion et des capacités entre signaux. Ce logiciel utilise une bibliothèque interne de structures prédéfinies pour déterminer les capacités de couplage entre signaux, et offre un calcul de résistances pour des formes d'interconnexions arbitraires. Ce logiciel utilise un mécanisme de compression des données sur disque à la volée. Il offre également la possibilité de réduire les réseaux RC extraits. Assura produit des fichiers au format Spice, DSPF, RSPF et SPEF.

Dracula RCX, de Cadence

Dracula [Cad] n'est pas un extracteur à proprement parler, puisqu'il s'agit d'un outil de vérification physique au niveau circuit. Il possède cependant un module, le module Dracula RCX, qui permet l'extraction des résistances parasites de routage au format DSPF.

Arcadia, de Synopsys

Cet extracteur [Syn01a] permet l'extraction des résistances parasites sur l'intégralité du circuit, avec un niveau de précision sélectionnable. L'extraction des capacités de couplage est possible sur quelques signaux, particulièrement les signaux critiques. Les formats de fichier en sortie sont DSPF, SPEF et Spice.

Star-RCXT, de Avant!

D'après les caractéristiques disponibles sur cet extracteur [Ava00], il permet l'extraction de circuits de 5 000 000 de transistors en moins de 5 heures, avec une précision de l'ordre de 5%, sur une station possédant 100Mo de mémoire. Je n'ai pas d'information sur la façon dont ces performances ont été obtenus. L'algorithme utilisé pour arriver à ce résultat s'appelle ScanBand. Star-RCXT extrait les circuits au niveau transistor, ainsi que les résistances d'interconnexions et les capacités de couplage. Ces dernières sont extraites de manière exhaustive sans filtrage, puis modélisées sous la forme d'un filtre réduit. Là non plus, je n'ai aucune information sur l'algorithme utilisé pour réduire les réseaux d'interconnexions couplés. Les formats de fichiers en sortie cet extracteur sont le format Spice, DSPF et SPEF.

ExtractionStage, de Sequence

On retrouve dans cet extracteur [Seq01a] toutes les principales caractéristiques d'un outil adapté aux technologies profondément submicronique, avec en plus l'extraction des inductances. Dans certaines conditions, ces dernières peuvent avoir une influence sur les signaux rapides. Cet extracteur annonce une bonne précision, due à un système de reconnaissance de formes des interconnexions. Chaque interconnexion est analysée région par région, jusqu'à 4 000 par millimètre de fil. Ensuite, chacune de ces régions est comparée avec une bibliothèque contenant plus de 50 000 cas précisément caractérisés, en résistance, capacité et inductance. Cette approche est semblable à l'approche utilisée par Xcalibre, sauf que la phase de calibrage a été réalisée une fois pour toutes par l'éditeur du logiciel. Cet extracteur fourni des fichiers au format DSPF, SPEF et Spice.

Space, de l'université de Delft

Space [BvGvdM95] [SvdMvG⁺95] est un extracteur disponible gratuitement pour les universités, que nous utilisons dans notre laboratoire. Il permet l'extraction des circuits de taille significative (la documentation cite plus d'un million de transistors par heures) avec résistances d'interconnexion et capacités de couplage. Les réseaux RC peuvent être réduits en spécifiant une fréquence de coupure à respecter. Il est également possible d'extraire les résistances dans le substrat. Ce logiciel reconnaît les transistors MOS et Bipolaire, a un temps de calcul et une occupation mémoire linéaire avec la surface du circuit. Il produit des fichiers au format Spice, SLS, SPF et SPEF. Informations disponibles sur <http://cas.et.tudelft.nl/space>.

Lynx, de l'université Pierre et Marie Curie

Lynx [PBSG01] [BBGP94] est un extracteur originalement conçu pour vérifier la fonctionnalité des circuits numériques. Il fait partie de la chaîne de CAO Alliance (<http://www-asim.lip6.fr/alliance/doc/overview/overview.html>). Au cours de mon année de DEA au laboratoire ASIM, j'y ai apporté de nombreuses modifications pour permettre l'extraction des résistances d'interconnexion. Lynx permet une extraction à plat au niveau transistor ou au niveau cellule, ou à n'importe quel niveau hiérarchique. Les principales limitations de cet outil sont qu'il n'accepte comme format d'entrée qu'une vue symbolique, propre au flot de conception de la chaîne de CAO Alliance, qu'il n'extrait pas la capacité de couplage et qu'il extrait sans réduction systématiquement tous les réseaux RC. Le format de sortie est le format Spice. Ce logiciel est actuellement très utilisé par notre équipe, à des fins d'enseignement ou pour des projets de recherche, ainsi que par de nombreuses universités à travers le monde. Informations disponibles sur <http://www-asim.lip6.fr/alliance> [PBSG01].

3.2 Les analyseurs temporels

Les caractéristiques des outils industriels présentés dans ce paragraphe sont extraites des spécifications fournies par les éditeurs sur leurs sites web respectifs, ainsi que quelques publications disponibles les concernant.

3.2.1 Solution proposée par Mentor Graphics : Mach TA

Dans la publication [Pay98], les ingénieurs de Mentor Graphics font la constatation que la prise en compte des effets diaphonants entraîne une rupture dans l'augmentation de complexité de l'analyse temporelle. Ils considèrent donc - à juste titre - que la précision des résultats

n'est pas le seul critère qui doit être retenu : la façon dont ils sont obtenus devient aussi un critère essentiel, sinon déterminant. De leur analyse, deux points ressortent clairement :

- Seule l'analyse statique permet de déterminer les chemins critiques du circuit. Cette approche permet d'étudier les circuits de grande taille, mais en contrepartie la précision obtenue n'est pas très bonne.
- Les approches dynamiques sont capables d'analyser des circuits jusqu'à un million de transistors, avec une précision assez bonne par rapport aux simulateurs électriques puisqu'ils prennent en compte beaucoup d'effets parasites (capacités de couplages). Ces approches ont comme inconvénient de s'appuyer des stimuli d'entrée, fournis par le concepteur.

Le choix de Mentor Graphics a été de développer un analyseur temporel dynamique, basé sur la simulation électrique, pour prendre en compte les effets parasites. Ils ont pour cela développé plusieurs techniques de simulations innovantes, basées sur le partitionnement du circuit. Cette famille de logiciel s'appelle Mach, et offre plusieurs outils : Mach TA est l'analyseur temporel dynamique. Il permet de simuler (le concepteur doit fournir des stimuli d'entrée), avec une précision comparable aux simulateurs électriques, des circuits jusqu'à 10 millions de transistors avec 2 Go de mémoire. Un module optionnel à ce premier logiciel, Mach PA, permet l'analyse de la consommation du circuit. La solution proposée par Mentor Graphics permet donc, à partir des stimuli du circuit, d'obtenir des temps de propagation avec précision. En revanche, elle n'offre pas de moyen systématique d'obtenir la chaîne la plus longue en fonction des agressions qui peuvent survenir dans le circuit. C'est le concepteur qui doit s'acquitter de cette tâche, aidé éventuellement par un analyseur statique. Il dispose pour cela de commandes interactives et de plusieurs niveaux de vue offerts par Mach TA. Mach TA lit les fichiers au format Spice, et les stimuli d'entrée peuvent être au format VHDL ou Verilog.

La réponse de Mentor Graphics pour prendre en compte les effets de couplage dans le comportement temporel du circuit passe donc par une simulation. Cette façon de procéder n'est pas systématique car il se peut qu'une configuration des entrées provoquant une erreur ne soit pas simulée. Cette approche ne permet donc pas de fournir des résultats fiables.

3.2.2 Solution proposée par Cadence : Celtic

L'outil Celtic [Cad01b] [Cad01c] est un outil d'analyse statique fonctionnant par itérations. La première phase de cet outil est de déterminer dans le circuit quels sont les signaux qui ne sont pas vulnérables aux effets de couplage. Cette étape se fait en calculant de manière systématique le bruit de couplage sur les signaux. Lorsque la hauteur du bruit est inférieure à un certain seuil, l'outil considère que le temps propagation sur ce signal est insensible au bruit. La

seconde étape concerne les signaux qui sont sensibles au bruit. Celtic détermine les temps de propagation minimums et maximums de façon systématique sur ces signaux. Puis, à partir de cet état, il élimine les agressions qui ne peuvent pas se produire, afin de ne pas tomber dans une approche trop pessimiste. La première information est de s'assurer qu'une agression entre deux signaux est possible, c'est à dire qu'un agresseur peut effectivement commuter en même temps qu'une victime. Ensuite, lorsqu'un signal est routé au-dessus d'un bus, et que les agressions sont toutes possibles, Celtic peut dans certains cas ne pas les considérer toutes actives. Ce choix repose sur l'hypothèse statistique que le cas où tous ces agresseurs commuteront en même temps n'est pas possible. Une fois que toutes les agressions qui ne peuvent pas se produire en même temps ont été éliminées, l'algorithme est réitéré afin d'éliminer de nouvelles agressions.

Contrairement au simulateur de Mentor, Celtic ne fonctionne qu'au niveau cellule. Cette spécification est néanmoins peu gênante, car peu de circuits possédant plusieurs millions de portes sont conçus au niveau transistor. Les fichiers de cellule lus sont le DSPF, Spice ou Verilog. La description des interconnexions lues sont le DSPF ou le SPEF.

3.2.3 Solution proposée par Synopsys : PrimeTime SI et Pathmill

Les outils de Synopsys PrimeTime SI [Syn01b], qui est une évolution de PrimeTime [Syn97], et PathMill [FTWCE00] sont comparables à leur concurrent de Cadence. La même approche itérative est utilisée. Dans un premier temps, les agressions pires cas sont prises en compte systématiquement, puis éliminées de manière itérative en fonction de certains critères [Syn01c]. On retrouve donc dans cet outil la même volonté de ne pas être exagérément pessimiste. Le comportement temporel d'un agresseur sur sa victime est pris en compte, ainsi que les corrélations logiques qui existent entre les signaux. La brochure précise curieusement que l'effet de plusieurs agresseurs peut être pris en compte. Cela est évidemment le cas le plus courant. Une fonctionnalité intéressante est la possibilité laissée à l'utilisateur de pouvoir mettre en place un filtrage, qui permet de réduire de façon considérable la complexité de l'analyse. PrimeTime SI fonctionne au niveau porte, et PathMill au niveau transistor.

3.2.4 Solution proposée par Sequence : ShowTime

L'analyseur temporel statique de la société Sequence [Seq01b] résout le problème des capacités de couplage en utilisant le même algorithme itératif que Cadence et Synopsys. Cet outil a deux finalités : l'analyse temporelle et le calcul du bruit sur les signaux. La précision annoncée

des résultats semble très réaliste puisqu'elle est de 10% pour les délais, et 15% pour la hauteur du bruit. Ces chiffres prennent cependant en compte un phénomène que les deux outils précédents ignorent : l'effet des inductances, qui intervient dans les puces comportant des parties fonctionnant dans le domaine des radio fréquences, ou qui comportent de long fils routés en cuivre. Cet outil fonctionne au niveau porte.

3.2.5 Conclusion

Les solutions proposées par Synopsys, Cadence et Sequence sont toutes trois très proches. Cette convergence des solutions est évidente une fois qu'on a bien pris la connaissance du problème. Le point important remarqué par les concepteurs de ces trois outils est que, bien que les capacités de couplages puissent engendrer de fortes variations de délais localement, leur influence globale reste malgré tout assez limitée. Le rôle des outils d'analyse temporelle est donc d'évaluer cette influence, mais sans tomber dans l'excès. L'approche utilisée dans ces 3 outils est donc de prendre en compte, pour chaque signal victime, l'activité simultanée possible de ses agresseurs. Ensuite, chaque éditeur a tenté de préciser l'analyse en prenant en compte d'autres facteurs, comme les corrélations entre signaux ou des considérations probabilistes. *Aucun de ces trois outils ne fonctionne au niveau transistor.* Le seul qui tente une approche à ce niveau est Mach TA, qui demeure néanmoins limité pour l'analyse de SOC de plusieurs millions de transistors.

Nous verrons que beaucoup de solutions présentées dans ce paragraphe sont étudiées et mises en œuvre dans cette thèse.

3.3 Les solutions publiées

Le nombre de publications sur le sujet a commencé à devenir important depuis l'année 1998, en même temps que l'influence des capacités de couplage a commencé à devenir préoccupante. Nous allons passer en revue quelques unes des solutions publiées.

3.3.1 Les modèles de calcul

Dans [SDM99], une solution est proposée pour ne pas prendre en compte systématiquement les pires conditions d'agression lorsque deux signaux résistifs présentent entre eux une capacité de couplage distribuée dans le calcul du temps de propagation d'une porte. En s'appuyant sur la figure 2.10, où la valeur de ΔT donnant le pire temps de propagation vaut à peu près $-15pS$, les auteurs déterminent si, compte tenu des conditions en entrée du circuit,

cette valeur de ΔT peut effectivement être atteinte. Dans certains cas, on trouvera par exemple que les valeurs possibles de ΔT entre deux signaux sont de $0ps$ et plus. Le temps de propagation maximum possible n'est donc plus $42ps$, mais $38pS$. Cette méthode affine donc les valeurs maximums des temps de propagation. Malheureusement, elle souffre de trois inconvénients majeurs dans le cadre de l'analyse temporelle. Le premier est qu'il faut réaliser une simulation électrique, exactement du même type que celle réalisée pour obtenir la figure 2.10, de deux lignes couplées de longueur "normale", avec toutes les combinaisons possibles d'émetteur. Si une bibliothèque de cellule dispose de 500 modèles différents, il faut réaliser 250 000 simulations électriques. Ensuite, pour prendre en compte l'effet de n agresseurs, les influences sont ajoutées. L'effet du couplage de n agresseurs n'étant pas linéaire, la précision des résultats s'en ressent. Enfin, la position des couplages n'est pas prise en compte sur les réseaux d'interconnexions. Ce modèle n'est donc pas directement utilisable dans l'analyse temporelle.

Un modèle de calcul permettant de prendre en compte les couplages de manière simple et très efficace a été présenté dans [CAKK00b]. Nous le détaillerons dans le paragraphe 4.2.

Le modèle présenté dans [CAKK00a] est une méthode pour déterminer l'alignement des agresseurs provoquant le pire temps de propagation d'un signal, ainsi qu'un algorithme pour l'analyse temporelle utilisant ce modèle. Les auteurs font l'hypothèse que la tension en sortie d'une porte en présence d'un couplage actif est égale à la somme de la tension en sortie de la porte sans prendre en compte ce couplage, à laquelle on ajoute le bruit de commutation due à l'agresseur si le signal victime était stable. Avec cette hypothèse de linéarité, il est alors immédiat de trouver à partir de la forme du signal obtenue l'alignement des agresseurs qui provoque le dernier passage de la tension du signal à $V_{dd}/2$, avec une solution proche de celle présentée dans 2.3.3. Pour utiliser cette méthode de calcul dans l'analyse temporelle, les auteurs proposent un algorithme événementiel (*event-driven*). Lorsqu'un couplage entre un signal victime et un ou plusieurs agresseurs est observé, grâce aux intervalles d'instabilité, la méthode de détermination des agressions pire cas est appliquée, et la nouvelle forme de signal est propagée en aval du circuit. Pour cela, deux types d'événements sont gérés : les événements de couplage (*coupling-event*) qui se produisent lorsqu'une nouvelle agression est détectée, et les événements de propagation (*driving-event*) qui se produisent lorsque l'entrée d'une porte est modifiée. Pour assurer la convergence de cette approche, plusieurs techniques sont également proposées.

Bien que cette approche présente une certaine originalité et de bonnes perspectives pour une utilisation de type industrielle, nous n'avons pas retenu les deux aspects développés ici. Concernant le modèle de détermination des agresseurs pire cas, il ne s'agit que d'une étape dans le calcul des délais, puisqu'il faut pouvoir l'appliquer aux réseaux d'interconnexions ré-

sistives : A chaque nœud d'un réseau RC, il y a une configuration d'agression donnant l'agression la pire, avec un impact sur le temps de calcul nécessaire pour réaliser cette opération. De plus, la linéarité supposée des effets est une hypothèse assez forte. Concernant l'approche événementielle, il reste à prouver que, comparée aux autres solutions présentées ici utilisant l'analyse de stabilité de manière intensive, elle soit effectivement plus rapide. Notons que la solution présentée ici revient à détecter progressivement les agressions. En conclusion, cette approche est très différente de celle que nous souhaitons utiliser, et beaucoup d'études doivent encore être faites sur celle-ci pour savoir si il est possible de l'appliquer sur des circuit VLSI avec des interconnexions résistives.

Le modèle proposé dans [SBO⁺01] perfectionne sensiblement la méthode du calcul de la tension en sortie d'une porte lorsque, pour déterminer son temps de propagation, on additionne le bruit de ses agresseurs à sa tension de sortie. Pour cela, la porte est modélisée de façon très ordinaire par un générateur de Thévenin, dont la source de tension est une rampe. Pour prendre en compte de façon précise l'effet des agresseurs, la valeur de la résistance du générateur est recalculée, à l'aide de quelques simulation très simples et d'une approche itérative. Ce modèle est certainement celui qui présente le meilleur rapport temps de calcul et précision. Malheureusement, cette approche utilisant le générateur de Thevenin est très différente de celle utilisé dans HiTas, et n'est pas directement exploitable dans notre cas. De plus, cette approche nécessite tout de même plusieurs simulations pour déterminer la forme du bruit de chaque agresseur sur le signal victime, et ne prend ici comme charge équivalente qu'une capacité à la masse. Ces raisons, et surtout celle concernant la compatibilité avec les modèles de HiTas, nous ont conduit à ne pas retenir ce modèle.

3.3.2 Les méthodologies

Dans [FTWCE00], les auteurs présentent certaines des méthodes utilisées dans l'outil Path-Mill de Synopsys, afin d'évaluer les délais en prenant en compte les résistances et les capacités d'interconnexion. Les capacités de couplage sont ramenées à la masse avec un facteur 2 pour la détermination des temps de propagation maximum, et 0 pour la détermination des temps de propagation minimum. Ensuite, les agressions qui ne peuvent avoir lieu, à cause de relations logiques entre signaux ou d'effets temporels, sont éliminées sur les chaînes critiques du circuit. Comme seules les chaînes critiques sont évaluées, des modèles plus précis -et donc plus coûteux en temps de calcul- peuvent être utilisés. Une idée intéressante est le calcul de la réponse temporelle en prenant en compte pour chaque signal de la chaîne longue les configurations d'agression pires cas, déterminées en calculant la forme du bruit sur la victime provoqué par chaque agresseur pris individuellement. Ces réponses sont combinées entre elles afin de déterminer la réponse exacte du signal. Cette méthode permet de faire face aux transitions non

monotones, comme représenté sur la figure 2.15.

La solution proposée dans [RLB00] est originale, car elle propose de déterminer l'activité des agresseurs en même temps que la phase de recherche des chemins les plus longs, donc en une seule passe. Pour cela, les auteurs utilisent le fait que la recherche des chemins les plus longs revient à déterminer sur chaque nœud du graphe de causalité le dernier instant de commutation possible. Pendant cette recherche, lorsqu'un délai doit être calculé, trois cas se présentent pour chaque agresseur. Soit cet agresseur a déjà été calculé (a), et son dernier instant de commutation ne risque pas d'influencer la transition du signal victime. Soit il a déjà été calculé (b), et il risque d'influencer la transition. Soit il n'a pas encore été calculé (c). Dans ces deux derniers cas, un modèle de délai prenant en compte cet agresseur actif est utilisé. Pour résoudre le problème du cas (c), les auteurs proposent de réaliser l'opération en plusieurs itérations. Lors de la première itération, aucune agression n'est prise en compte. A la fin de cette itération, tous les nœuds du graphe de causalité ont été calculés, et le dernier instant de commutation est donc connu pour tous les nœuds. Une nouvelle itération est faite, dans laquelle seuls les cas (a) et (b) peuvent se produire pour les agressions. A la fin de cette itération, certains délais ont donc été modifiés. L'algorithme est donc réitéré plusieurs fois jusqu'à ce qu'il n'y ait plus de nouvelles agressions. Cette solution de détection des agressions permet d'obtenir une borne supérieure dans les temps de propagation élémentaires, sans être trop pessimiste. Nous allons voir par la suite que c'est cette solution que nous avons préféré retenir. Néanmoins, telle qu'elle est présentée ici, cette approche souffre de deux limitations. La première est qu'elle ne s'adapte qu'aux circuits à registre. Les latches, qui possèdent un état passant sur la moitié de la période d'horloge, ne permettent pas d'appliquer cet algorithme. La seconde est qu'elle ne permet pas le calcul des délais minimums et la détection des chaînes courtes. Or ce problème est plus important que celui des chaînes longues, car il est indépendant de la période de l'horloge.

Enfin, il existe quelques articles, dont [ADBTP01], qui proposent des solutions pour éliminer les agressions en prenant en compte l'aspect fonctionnel des agresseurs. Dans le cadre de cette thèse, nous ne nous sommes pas intéressés à ce domaine.

3.4 Le calcul des délais élémentaires dans HiTas

3.4.1 Les temps de propagation dans les portes

Introduction

Nous avons vu que les analyseurs temporels sont répartis en deux grandes familles : ceux qui travaillent au niveau porte, et ceux qui travaillent au niveau transistor. Nous nous intéressons à cette seconde famille, dans le cas particulier mais néanmoins très courant des transistors

MOS. Le calcul des temps de propagation dans les portes reconstruites pendant la phase d'abstraction fonctionnelle passe par un modèle de courant dans les transistors. Dans HiTas, c'est le modèle MCC.

Notations

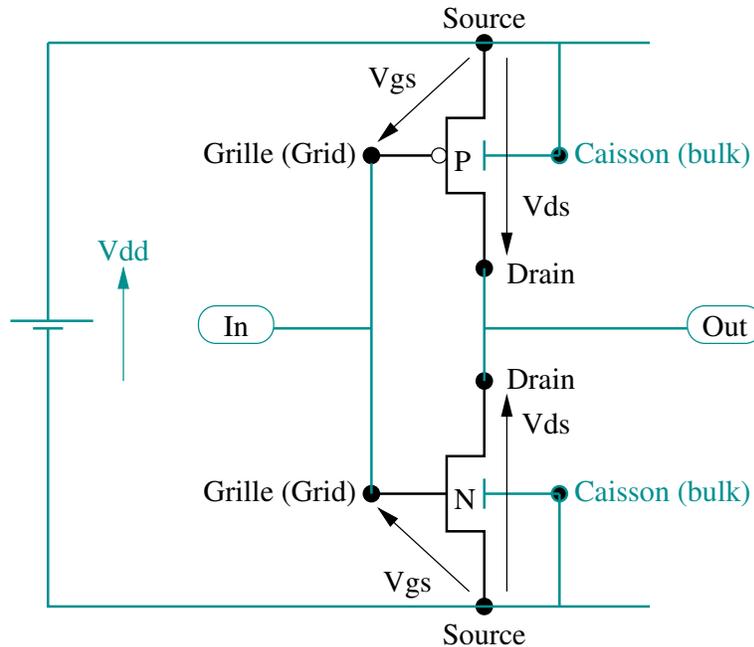


FIG. 3.1 – Notations utilisées pour les transistors MOS dans un inverseur.

Le transistor est couramment modélisé dans les simulateurs électriques par quatre connecteurs : le drain, la grille, la source et le caisson. Dans les circuits numériques, le caisson des transistors est toujours connecté à une alimentation. La première simplification possible est donc de ne pas considérer cette entrée. Le modèle MCC considère donc le transistor comme une entité à trois connecteurs : grille, drain et source. La tension de seuil V_{t_n} du transistor N est positive, et la tension de seuil V_{t_p} du transistor P est par convention négative. Nous appellerons V_{ds} la tension entre le connecteur Drain et le connecteur Source, et V_{gs} la tension entre le connecteur Grille et le connecteur Source. Physiquement, il n'y a aucune différence entre le drain et la source d'un transistor. Par convention, nous appellerons Drain le connecteur qui a la plus grande valeur de tension dans le cas du transistor N, et la plus faible valeur dans le cas du transistor P. Ces conventions sont rappelées dans la figure 3.1, où j'ai représenté un inverseur C-MOS et sa polarisation. Dans le cas des portes C-MOS, la tension d'entrée pour le transistor

est la tension V_{gs_n} , et la tension de sortie est la tension V_{ds_n} .

Les 3 régimes de courant du modèle MCC [Haj92]

Dans l'analyse temporelle, la principale valeur qui nous intéresse est la valeur du courant entre le drain et la source I_{ds} en fonction de la différence de potentiel entre la grille et la source V_{gs} . Comme pour de nombreux modèles, trois états sont définis :

- **L'état bloqué** : Dans ce cas, le transistor ne permet aucun passage du courant entre le Drain et la Source. C'est le régime statique correspondant au cas où l'entrée est stable à V_{dd} ou à V_{ss} . La condition pour que le transistor soit dans cet état est : $V_{gs} < V_t$.
- **L'état linéaire** : Dans ce régime, le courant I_{ds} est proportionnel à la tension V_{ds} , le transistor se comporte exactement comme une résistance. C'est la raison pour laquelle on l'appelle parfois le régime ohmique. Le courant vaut alors :

$$I_{ds} = \frac{I_{sat}}{V_{sat}} V_{ds} \quad (3.1)$$

dès lors que $V_{ds} < V_{sat}$ et $V_{gs} > V_t$.

- **L'état saturé** : Le régime saturé. Dans ce cas, le courant I_{ds} est constant quelle que soit la tension V_{ds} , et vaut I_{sat} . Cela est le cas lorsque $V_{ds} > V_{sat}$ et $V_{gs} > V_t$.

La valeur V_{sat} vaut :

$$V_{sat} = k(V_{gs} - V_t) \quad (3.2)$$

où k représente un paramètre dépendant de la technologie du transistor. On constate que le seuil de saturation du transistor dépend directement de la tension d'entrée V_{gs} . La valeur I_{sat} représente le courant maximum dans le transistor. Elle dépend également des paramètres physiques du transistor, mais aussi de la tension d'entrée V_{gs} , et vaut

$$I_{sat} = \frac{W}{L} \frac{A(V_{gs} - V_t)^2}{1 + B(V_{gs} - V_t)} \quad (3.3)$$

Où W est la longueur effective de la grille, L la largeur effective de la grille, les paramètres A et B sont des paramètres technologiques.

Les paramètres technologiques

Le modèle de courant dans un transistor nécessite donc l'utilisation de quatre paramètres pour le transistor N, et quatre paramètres pour le transistor P. A titre d'exemple, voici les valeurs obtenues pour une technologie industrielle en $0,25\mu m$ pour le transistor N :

$$\begin{aligned}
 A_n &= 1,225 \cdot 10^{-4} & A \cdot V^{-2} \\
 B_n &= 2,280 \cdot 10^{-1} & V^{-1} \\
 k_n &= 0,412 \\
 Vt_n &= 0,39 & V
 \end{aligned}$$

Ces paramètres sont déterminés à l'aide de simulations réalisées avec un simulateur électrique, à partir de la description des transistors de la technologie à caractériser. Comme seul le résultat de la simulation est utilisé, n'importe quel modèle électrique de simulation peut être utilisé. Un outil réalise de manière automatique cette caractérisation.

Le modèle de front

Nous avons vu dans le chapitre 1.3.2 que le front d'entrée des transistors était simplifié et représenté par une fonction mathématique. Dans le modèle MCC, c'est la tangente hyperbolique qui a été choisie :

$$V(t) = Vt + (Vmax - Vt) \tanh\left(\frac{t}{F}\right) \quad (3.4)$$

où $Vmax$ est la valeur maximum de la tension. Celle ci vaut généralement la valeur de la tension d'alimentation du circuit. La valeur F correspond à la "pente" du front. Elle correspond à la durée de montée du signal entre 20% et 80% de son excursion.

Le temps de propagation d'une porte

Le modèle de transistor MCC est donc simplement la définition de 3 régimes de courant dans le transistor, et d'un modèle de front, correspondant aux équations 3.1 à 3.4. Dans HiTas, ce modèle est utilisé de façon très simplifiée afin d'obtenir des performances (temps CPU) acceptables. Le calcul du temps de propagation d'une porte repose sur la commutation d'un inverseur équivalent. Dans le cas d'une transition montante de l'entrée, on calcule le courant du transistor N, dont l'entrée commute et la sortie est chargée par une capacité équivalente C_{eq} , comme cela est représenté sur la figure 3.2.

Lorsque l'entrée de ce transistor commute avec une entrée définie suivant l'équation 3.4, et en faisant l'hypothèse que le transistor est toujours saturé, on peut calculer que le courant instantané absorbé par le transistor à partir des équations 3.1 et 3.3 est :

$$I(t) = I_{max} \frac{1 + B(Vmax - Vt)}{1 + B(Vmax - Vt) \tanh\left(\frac{t}{F}\right)} \left(\tanh\left(\frac{t}{F}\right)\right)^2 \quad (3.5)$$

Avec :

$$I_{max} = \frac{W}{L} \frac{A(Vmax - Vt)}{1 + B(Vmax - Vt)} \quad (3.6)$$

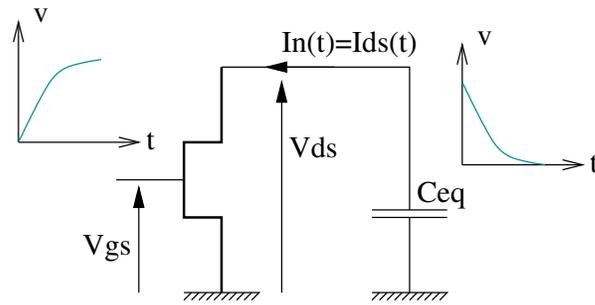


FIG. 3.2 – Modèle équivalent de porte utilisé dans HiTas : Cas d'une transition montante sur l'entrée.

Avec les valeurs des paramètres technologiques vus précédemment, un transistor $W = 1,5\mu m$ et $L = 0,25\mu m$ absorbe un courant $I_{max} = 1,047mA$. Revenons en détail sur cette condition de saturation. Lorsque l'entrée commute, dans le cas d'une transition montante, la capacité est initialement chargée à V_{dd} , et c'est le courant $I(t)$ qui va la décharger. Le courant $I(t)$ peut voir son régime passer de saturé à linéaire au court de cette décharge, suivant la différence entre $V_{sat}(t)$ et $V_{ds}(t)$. La tension V_{sat} dépend des paramètres k et V_t . En 1990, pour la technologie $1\mu m$, Amjad Hajjar avait calculé $k = 0,6$ et $V_t = 0,75V$, pour une tension d'alimentation qui était alors de $5V$. Comme dans beaucoup de cas les cellules sont équilibrées, la tension qui détermine l'instant de commutation de la sortie est définie à $V_{ds} = V_{dd}/2$. Le pire cas pour la saturation, c'est à dire le cas où la décharge de la capacité se fait le plus longtemps possible en régime linéaire, se produit quand l'entrée est un échelon. Dans ce cas, V_{gs} passe directement de $0V$ à $5V$, et conserve cette valeur. $V_{sat}(t)$ vaut alors la valeur constante de $2,55V$. La décharge de la capacité se fera entre $5V$ et $2,55V$ avec le courant I_{sat} , et entre $2,55V$ et $2,50V$ avec le courant correspondant au régime linéaire. L'hypothèse de considérer à l'époque le transistor comme étant toujours saturé était justifiée, et bien que le passage de régime saturé à régime linéaire ne soit pas très nette, l'erreur commise à cette étape n'était pas très grande.

Le même calcul avec les paramètres de la technologie $0,25\mu m$, dont l'alimentation est $2,5V$ nous donne $V_{sat} = 870mV$, à comparer avec la tension de seuil de $1,25V$. C'est à dire que même dans le pire cas, le passage au régime linéaire se fait bien après l'instant de référence de la commutation. Ces chiffres montrent qu'avec les technologies, l'hypothèse que le transistor est saturé se révèle de plus en plus vraie. Cela explique en partie les bons résultats obtenus pour les dernières technologies (jusqu'à $0,18\mu m$) avec cette approche.

La capacité équivalente C_{eq} modélise de nombreux phénomènes : La capacité de routage, la capacité d'entrée des portes de charge, le conflit électrique dû au transistor P, les capacités de couplage grille-drain,... Le calcul des portes complexes s'effectue en calculant un transistor et un signal d'entrée équivalent à la branche qui commute.

L'instant de basculement T_s de la sortie s'obtient à partir du courant donné par l'équation 3.5 simplement en résolvant l'équation différentielle :

$$I(t) = C_{eq} \cdot \left. \frac{dV_{ds}(t)}{dt} \right|_{V_{ds} = \frac{v_{dd}}{2}} \quad (3.7)$$

Cette équation n'étant pas solvable analytiquement, une solution itérative est utilisée pour la résoudre. Le temps de propagation T_p de la porte s'obtient en retranchant l'instant de basculement T_e de l'entrée de T_s . Le front de sortie F_s est calculé à partir de la dérivée a de la tension de sortie à l'instant T_s , et ce, directement à partir de l'équation 3.7. La valeur de F_s est déduite en identifiant ce résultat à la valeur de la dérivée de l'équation 3.4 à l'instant T_s :

$$a = \frac{dV}{dT} = \frac{I(t)}{C_{eq}} = \frac{I_{max}}{C_{eq}} \cdot \frac{(1 + B \cdot (V_{max} - V_t)) \cdot \tanh^2(T_s / F_e)}{1 + B \cdot (V_{max} - V_t) \cdot \tanh(T_s / F_e)} \quad (3.8)$$

$$F_s = \frac{V_{dd} \cdot (3 \cdot V_{dd} - 4 \cdot V_t)}{4 \cdot a \cdot (V_{dd} - V_t)} \quad (3.9)$$

Dans HiTas, le front d'entrée n'est pas pris en compte pour déterminer le front de sortie. Cela revient à considérer que la tension d'entrée est un échelon ($F_e \leftarrow 0$). Le temps de propagation dépend dans HiTas de deux facteurs externes à la porte, le front d'entrée F_e et la valeur de la capacité de charge C_{eq} . Dans le cas très fréquent où $T_s > 2F_e$, HiTas est capable de calculer deux paramètres R et S , qui représentent la sensibilité du temps de propagation de la porte en fonction des valeurs C et F :

$$T_p(C_{eq}, F_e) = R \cdot C_{eq} + S \cdot F_e \quad (3.10)$$

Enfin, soulignons que le modèle de courant dans le transistor MCC peut tout à fait être remplacé par un autre modèle. La démarche (transistor actif saturé, charge capacitive prenant en compte les autres effets, front en tangente hyperbolique) de calcul pour déterminer les temps de propagation ne dépend pas du modèle de transistor. Il est tout à fait possible de conserver

la même approche avec un autre modèle de courant, seules les équations et leurs méthodes de résolution changeront.

3.4.2 Les temps de propagation dans les lignes RC

Le délai d'Elmore

Dans HiTas, les délais à travers les interconnexions sont déterminés en utilisant le délai d'Elmore [Elm48]. L'un des avantages décisif de ce modèle est sa capacité à traiter de grands (> 10000 éléments RC) réseaux dans des temps de calcul particulièrement courts, grâce à la méthode développée dans [RPAH83]. De plus, le calcul de ce délai se prête bien à une implémentation hiérarchique, ce qui est un avantage majeur pour l'analyse de gros circuits. Le détail de ce délai est présenté dans l'annexe F.1.

Le point critiquable de cette approche est sa précision, puisqu'elle se situe autour de 10%. Dans certains cas particuliers, l'erreur peut atteindre plusieurs fois la valeur du délai. Ce cas se produit lorsque la réponse impulsionnelle du réseau RC n'est pas symétrique (voir la figure F.1), principalement en début de ligne. Mais dans ces cas, le délai de ligne est très faible par rapport au délai de porte, une erreur sur cette évaluation n'a donc pratiquement pas de conséquences sur la précision de l'analyse temporelle.

La méthode AWE

Principe : Publiée dans [TPA90], cette méthode généralise le délai d'Elmore à l'ordre n . L'idée de cette approche est de calculer les $q = 2n - 1$ premiers moments de la fonction de transfert sur chacun des nœuds du réseau RC sur lesquels on cherche le temps de propagation, puis, par une approximation de Padé ([PTVF88] chapitre 5.12), d'en déduire la fonction de transfert sur ces nœuds. En décomposant cette équation en pôles et résidus, nous simplifions cette fonction de transfert pour l'inverser dans le domaine temporel.

Calcul des moments dans le cas général : Le calcul des $2n - 1$ premiers moments est basé sur les équations nodales du circuit. Ce circuit peut contenir des résistances, des capacités et des inductances, mais également des générateurs commandés, et un nombre quelconque d'entrées. Il n'y a pas de contrainte de placement de ces éléments. Malheureusement, cette approche passe par une représentation matricielle du circuit. Cette matrice contient N^2 éléments flottants, N étant le nombre de nœuds sur le réseau. Cela pose donc un premier problème : la taille de la matrice, car le nombre de nœuds pour décrire un réseau RC peut dépasser le millier, et même

beaucoup plus. Ensuite, cette matrice est utilisée pour résoudre un système de n équations à n inconnues. L'utilisation de l'algorithme classique de Gauss-Jordan ([PTVF88] chapitre 2.1) pour résoudre cette matrice est donc vouée à l'échec pour deux raisons : le temps de calcul est en O^3 , et l'accumulation des erreurs est très rapide. Des techniques de résolution itératives ([PTVF88] chapitre 2.5) permettent de contourner partiellement le problème des erreurs d'arrondi, mais restent toujours extrêmement coûteuses en calculs.

Calcul des moments dans le cas particulier des arbres RC : Le fait de restreindre les interconnexions analysées aux arbres RC permet d'utiliser une méthode de récurrence et le calcul des moments devient extrêmement rapide et plus précis. Il est présenté dans l'annexe F.2.

Détermination des pôles et résidus : Une fois le calcul des $q = 2n - 1$ premiers moments effectué, nous cherchons une fonction de transfert dont le développement correspond à celui que nous avons obtenu pour le nœud étudié, grâce à l'approximation de Padé :

$$H(s) = \frac{\sum_{i=0}^{n-1} a_i \cdot s^i}{1 + \sum_{i=1}^n b_i \cdot s^i} \approx 1 + \sum_{i=1}^q m_i \cdot s^i \quad (3.11)$$

Le calcul des b_i est assez peu coûteux en temps de calcul puisqu'il suffit de résoudre le système suivant :

$$\begin{bmatrix} m_0 & \dots & m_{n-1} \\ \vdots & \ddots & \vdots \\ m_{n-1} & \dots & m_{2n-2} \end{bmatrix} \begin{bmatrix} b_n \\ \vdots \\ b_1 \end{bmatrix} = - \begin{bmatrix} m_n \\ \vdots \\ m_{2n-1} \end{bmatrix} \quad (3.12)$$

pour un n petit (≤ 6). En effet, dans une analyse temporelle, nous avons rarement besoin d'une représentation à un ordre supérieur à 3. Vu la taille relativement petite de ces matrices, une implémentation de l'algorithme de Gauss-Jordan est tout à fait appropriée car sa convergence est assurée. Nous cherchons maintenant une représentation de $H(s)$ sous la forme d'une somme en pôles et résidus. Il est donc inutile de calculer les coefficients a_i de $H(s)$. Le dénominateur de $H(s)$ est un polynôme de degré n à n racines réelles. Trouver les racines de ce polynôme ne pose donc pas de problème, en utilisant un algorithme de déflation ([PTVF88] chapitre 9.5) combiné avec la méthode de Gauss. Rappelons que l'algorithme de déflation consiste à trouver sur un polynôme de degré n une des racines (à l'aide de l'algorithme de Gauss), puis à calculer un nouveau polynôme de degré $n - 1$ à partir de la racine trouvée et de recommencer jusqu'à ce que le polynôme restant soit d'ordre 1. Une fois que les racines p_i du dénominateur de $H(s)$ sont trouvées, il faut calculer les résidus k_i à l'aide du système :

$$\begin{bmatrix} \frac{1}{p_1} & \frac{1}{p_2} & \cdots & \frac{1}{p_n} \\ \frac{1}{p_1^2} & \frac{1}{p_2^2} & \cdots & \frac{1}{p_n^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{p_1^n} & \frac{1}{p_2^n} & \cdots & \frac{1}{p_n^n} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix} = - \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_n \end{bmatrix} \quad (3.13)$$

Nous obtenons donc la représentation pour le nœud considéré :

$$H(s) = \sum_{i=1}^n \frac{k_i}{s - p_i} \quad (3.14)$$

Réponse dans le domaine temporel : A partir de cette fonction, il est possible d'appliquer n'importe quelle forme d'entrée pour déduire la forme du signal sur n'importe quel nœud de l'arbre RC, et donc d'en déduire le temps de propagation.

Contraintes d'implémentation : Malgré son apparente facilité d'implémentation, cette méthode montre à l'usage quelques limitations. La plus grave est due à l'approximation de Padé, dans l'équation 3.11. Il se peut dans certains cas que les pôles calculés à partir des b_i soient positifs ou complexes. La détection de ce phénomène ne pose pas de problème. Une solution raisonnablement coûteuse consiste à augmenter (ou diminuer) le nombre de pôles demandés. L'exemple suivant montre les temps de propagation, calculés pour différents ordre, d'un signal extrait d'un circuit en technologie $0,18\mu m$:

Ordre 1 1.029262pS
 Ordre 2 3.642427pS
 Ordre 3 3.721567pS
 Ordre 4 Echec
 Ordre 5 3.748838pS

On constate sur cet exemple que seul le calcul à l'ordre 4 pose un problème. Dans les cas courants, les problèmes n'apparaissent qu'à partir de l'ordre 3, cela ne dépendant bien sûr que du réseau considéré. Nous constatons ensuite que le calcul des temps de propagation aux ordre 3 et plus n'apporte pas de gain en précision significatif.

Le second problème est plus subtil : il est dû aux erreurs d'arrondis qui surviennent lors du calcul des moments. En effet, entre le moment d'ordre i et le moment d'ordre $i + 1$, pour un nœud particulier, il est possible d'avoir plus de deux ordres de grandeur d'écart. Par exemple, le premier moment peut valoir -10^3 et le second 10^5 , ces valeurs étant courantes. Des erreurs

de calcul significatives peuvent donc apparaître. Le moyen immédiat pour diminuer l'influence ce phénomène est d'utiliser une représentation des nombres flottants offrant le maximum de précision, par exemple le format `double` du langage C. Le moyen le plus efficace consiste cependant à conserver à peu près le même ordre de grandeur sur ces moments, en normalisant les valeurs des résistances, par exemple par rapport à la valeur du moment d'ordre 1 le plus grand sur le réseau RC. Cette méthode permet d'améliorer très sensiblement la précision pour un surcoût en temps CPU assez faible. Pour dénormaliser la valeur du temps de propagation calculée avec cette approche, il suffit de multiplier le temps obtenu par la valeur choisie pour la normalisation.

Conclusion sur cette méthode : Nous avons donc vu que ce modèle permet d'obtenir plus de précision que l'utilisation du délai d'Elmore, mais au pris d'un coût en temps CPU plus grand. Le logiciel HiTas n'utilise pas pour l'instant d'approximation supérieure à l'ordre 1. Dans le domaine de l'analyse temporelle, il serait judicieux d'utiliser une approximation à l'ordre 3. Au-delà, l'apport en précision n'aurait pas d'utilité pratique.

3.5 Conclusion

Nous avons vu que les solutions disponibles sur le marché sont actuellement assez nombreuses. Toutes sont cependant basées au niveau cellule précaractérisée. Cette limitation rend la prise en compte l'effet de masquage de la capacité des charges par la résistance de l'interconnexion assez délicate. Si pour les technologies actuelles cet effet reste assez limité, il n'en sera peut être pas de même pour les technologies futures. L'approche au niveau transistor de HiTas permet de s'affranchir sans problème de cette contrainte.

La méthode retenue par tous les acteurs industriels est de considérer systématiquement les pires agressions possibles, puis d'éliminer progressivement les agressions qui ne peuvent pas se produire. Cette méthode présente un caractère résolument pire cas. Or, le problème qu'ont un certain nombre d'utilisateurs de ces outils est ce caractère exagérément pessimiste. Nous allons donc tenter de résoudre ce point non pas en proposant des méthodes d'élimination des agressions *a posteriori*, mais en les détectant de manière progressive.

Nous avons également présenté dans ce chapitre le modèle utilisé dans HiTas pour déterminer les temps de propagation élémentaires à travers les portes, et les deux techniques les plus efficaces pour déterminer rapidement et avec précision le temps de propagation dans les interconnexions : le délai d'Elmore et la méthode AWE. Un modèle pour prendre en compte l'effet

de la résistance de l'interconnexion sur le temps de propagation d'une porte sera présenté dans le chapitre suivant. La littérature scientifique comporte de nombreux autres modèles décrivant le calcul de temps de propagation à travers les portes et les interconnexions [Van98]. Nous nous sommes arrêtés ici à ceux que nous avons particulièrement étudiés.

Nous n'avons pas présenté ici de techniques permettant de déterminer de façon immédiate les temps de propagation minimum et maximum des portes et des interconnexions en fonction de leurs agresseurs. Nous voulions trouver une solution à la question : "Comment déterminer sur la porte et sur tous ses récepteurs les temps de propagation minimum et maximum lorsque ses agresseurs actifs sont connus?". Au début de mon travail de thèse, peu de publications étaient disponibles sur le sujet. Aujourd'hui, ce sujet devenant d'actualité, on commence à trouver un certain nombre de solutions sur ce sujet. Cependant, beaucoup d'entre elles sont trop coûteuses en temps CPU, ne prennent pas en compte les résistances d'interconnexions ou ne sont pas adaptables directement à HiTas. C'est la raison pour laquelle nous avons préféré développer un modèle plus adapté à HiTas. Ce modèle est présenté dans le chapitre suivant.

Chapitre 4

Modélisation des temps de propagation

Dans ce chapitre, nous allons présenter un modèle de calcul au niveau transistor qui prenne en compte à la fois les résistances d'interconnexion et les capacités de couplage. Nous commencerons par résoudre le problème du masquage de la charge par la résistance de ligne dans le paragraphe 4.1. Ensuite, nous expliquerons comment nous avons modélisé les effets de couplage dans le paragraphe 4.2. Nous présenterons ensuite la façon dont les délais dans les interconnexions et dans les portes doivent être calculés dans les paragraphes 4.3 et 4.4. Nous verrons ensuite la façon dont ces modèles doivent être utilisés dans le chapitre 4.5. Enfin, nous présenterons les résultats expérimentaux obtenus avec ces modèles dans le paragraphe 4.6, puis nous conclurons ce chapitre.

4.1 Evolution du modèle de charge de porte dans HiTas

4.1.1 Problème

Nous avons vu dans l'équation 3.7 que le modèle de charge utilisé dans HiTas est une capacité pure reliée à la masse. Cette modélisation était en 1992 tout à fait justifiée, car la résistance des fils les plus longs était négligeable. Nous avons vu dans le chapitre 2.2.2 que cette hypothèse est remise en question pour les technologies actuelles. C'est pourquoi HiTas intègre deux types de délais, les délais dus aux portes, et les délais dus aux interconnexions. Cependant, HiTas ne prend pas en compte l'effet de masquage des capacités dû aux interconnexions résistives. La part de la capacité de charge équivalente de la porte (qui comprend de nombreux effets), correspondant aux capacités d'entrée des portes de charge et des interconnexions, est égale à la somme algébrique de celles-ci, ce qui peut conduire dans certains cas à une surestimation des délais [QPP94]. Avant de prendre en compte les effets temporels de couplage, il faut donc améliorer ce point.

4.1.2 Calcul de la charge équivalente à un réseau d'interconnexion en sortie d'une porte.

On trouve dans la littérature [OL89] un modèle extrêmement efficace pour le calcul d'une charge équivalente à n'importe quel ordre d'un arbre RC. Cette approche calcule de manière récursive en partant des récepteurs vers l'émetteur un développement limité de l'admittance de la ligne à un degré quelconque :

$$y(s) = y_1 \cdot s + y_2 \cdot s^2 + y_3 \cdot s^3 + \dots \quad (4.1)$$

Le principe du calcul est de déterminer l'admittance $y_u(s)$ de la charge $y_d(s)$ vue à travers une admittance discrète (R ou C) comme cela est représenté sur la figure 4.1. Dans le cas des arbres

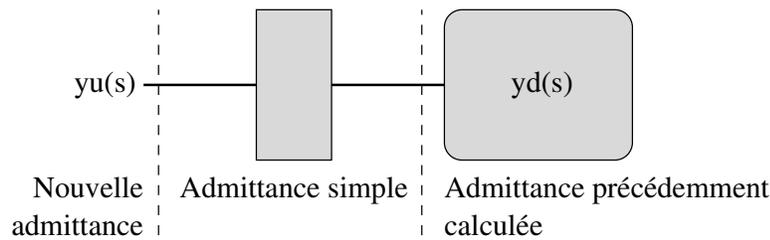


FIG. 4.1 – Principe de l'algorithme du calcul récursif de la charge équivalente

RC, l'admittance simple est soit une capacité reliée à la masse, soit une résistance flottante :

- Cas d'une résistance : Dans ce cas, nous avons une mise en série d'une admittance $y_d(s)$ avec une résistance $R = 1/G$ (voir la figure 4.2). L'admittance $y_u(s)$ vaut :

$$y_u(s) = \frac{G \cdot y_d(s)}{G + y_d(s)} \quad (4.2)$$

En utilisant pour $y_d(s)$ et $y_u(s)$ la définition donnée dans 4.1, et en identifiant les puissances de s , nous trouvons alors très facilement que :

$$y_{u1} = y_{d1} \quad (4.3)$$

$$y_{u2} = y_{d2} - R \cdot y_{d1}^2 \quad (4.4)$$

$$y_{u3} = y_{d3} - 2 \cdot R \cdot y_{d1} \cdot y_{d2} + R^2 \cdot y_{d1}^3 \quad (4.5)$$

- Cas d'une capacité à la masse : Ce cas est plus simple, car il s'agit de la mise en parallèle de l'admittance de la capacité et de $y_d(s)$. Nous obtenons donc immédiatement :

$$y_{u1} = y_{d1} + C \quad (4.6)$$

$$y_{u2} = y_{d2} \quad (4.7)$$

$$y_{u3} = y_{d3} \quad (4.8)$$



FIG. 4.2 – Les deux admittances possibles dans le cas des arbres RC

Dans le cas de plusieurs branches arrivant sur un nœud de l'arbre RC, l'admittance $y_u(s)$ est égale à la somme des admittances des branches $y_d(s)$ en aval de ce nœud. Ces différentes opérations permettent de calculer le développement limité à n'importe quel degré de l'admittance équivalente $y(s)$ vue par une porte. Notons que le terme y_1 est égal à la somme des capacités de ce réseau.

Cette admittance équivalente peut être transformée en une impédance équivalente. Nous avons représenté, sur la figure 4.3, les charges correspondant à l'équation 4.1, des développements limités aux degrés 1, 2 et 3. Les degrés 1 et 2 correspondent à un filtre du premier ordre, et le degré 3 à un filtre du deuxième ordre. En pratique, le degré 2 n'apporte pas d'amélioration significative par rapport au degré 1, et le degré 3 est suffisant pour les technologies actuelles.

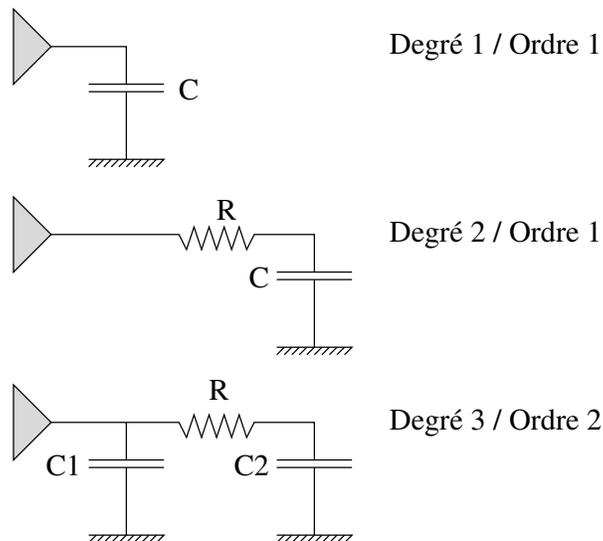


FIG. 4.3 – Les charges équivalentes à un développement limité aux degrés 1, 2 et 3

Le calcul des éléments discrets à partir de l'admittance équivalente est immédiat :

- Pour le degré 1, nous avons $C = y_1$;
- Pour le degré 2, $C = y_1$ et $R = -\frac{y_2}{y_1}$;
- Et pour le degré 3, $C1 = y_1 - \frac{y_2^2}{y_3}$, $C2 = \frac{y_2^2}{y_3}$, et $R = -\frac{y_3}{y_2}$. En particulier, la capacité totale de l'interconnexion est égale à $C1 + C2$.

4.1.3 Application dans HiTas

Le calcul du temps de propagation d'une porte s'effectue à partir du courant absorbé par un transistor en régime saturé et de la charge équivalente, qui est pour l'instant représentée par une capacité. Le passage à l'ordre 2 revient à résoudre le système d'équations différentielles suivant, avec les conventions représentées sur la figure 4.4 :

$$i_1(t) = C_1 \cdot \frac{dv_1(t)}{dt} \quad (4.9)$$

$$i_2(t) = C_2 \cdot \frac{dv_2(t)}{dt} \quad (4.10)$$

$$i_2(t) = \frac{v_1(t) - v_2(t)}{R_\pi} \quad (4.11)$$

$$i(t) = i_1(t) + i_2(t) \quad (4.12)$$

Nous cherchons à déterminer l'instant où $v_1(t)$ atteint $v_{dd}/2$, le courant $i(t)$ étant le courant

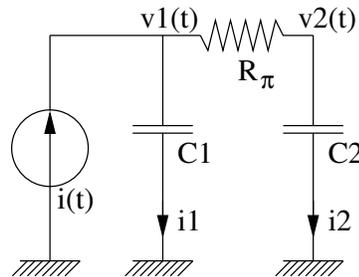


FIG. 4.4 – Notations utilisées dans le cas de la charge au deuxième ordre.

défini par l'équation 3.5. Nous n'allons pas tenter de résoudre ce système pour deux raisons : Il n'y a pas de solution analytique à ce système, et cela nous obligerait à réécrire une grande partie de HiTas. J'ai préféré revenir à la forme approchée du temps de propagation, donné par l'équation :

$$Tp(C, F) = R.C + S.F \quad (4.13)$$

En développant le calcul de l'équation 3.7, nous trouvons que :

$$R = \frac{v_{dd}}{2.I_{max}} \quad (4.14)$$

$$\begin{aligned}
S = \ln(2) \cdot \left(1 + \frac{1}{B \cdot (V_{max} - Vt)}\right) & \quad (4.15) \\
- \frac{\ln(0,5 \cdot (1 + B \cdot (V_{max} - Vt)))}{B \cdot (V_{max} - Vt) \cdot (B \cdot (V_{max} - Vt) - 1)} \\
+ \frac{1}{2} \cdot \ln \left(\frac{(V_{max} - Vt) - 0,5 \cdot Vdd + Vt}{(V_{max} - Vt) + 0,5 \cdot Vdd - Vt} \right)
\end{aligned}$$

Pour la technologie en $0,25\mu m$, les valeurs typiques sont $R = 1139\Omega$ et $S = 0,478$. Le produit $R \cdot C$ correspond donc au temps que met le courant constant I_{max} (équation 3.6) pour charger la capacité entre $0v$ et $vdd/2$. L'idée que j'ai donc retenue est de remplacer ce temps $R \cdot C$ par le temps que met le courant I_{max} pour absorber la charge de la cellule en π . Le système d'équation 4.9 à 4.12 devient :

$$i_1(t) = C_1 \cdot \frac{dv_1(t)}{dt} \quad (4.16)$$

$$i_2(t) = C_2 \cdot \frac{dv_2(t)}{dt} \quad (4.17)$$

$$i_2(t) = \frac{v_1(t) - v_2(t)}{R_\pi} \quad (4.18)$$

$$I_{max} = i_1(t) + i_2(t) \quad (4.19)$$

Ce système se résout en utilisant la transformée de Laplace :

$$I_1(s) = s \cdot C_1 \cdot V_1(s) - C_1 \cdot v_1(0) \quad (4.20)$$

$$I_2(s) = s \cdot C_2 \cdot V_2(s) - C_2 \cdot v_2(0) \quad (4.21)$$

$$I_2(s) = \frac{V_1(s) - V_2(s)}{R_\pi} \quad (4.22)$$

$$\frac{I_{max}}{s} = I_1(s) + I_2(s) \quad (4.23)$$

Initialement, $v_1(0)$ et $v_2(0)$ valent V_i . En égalisant 4.21 et 4.22, nous obtenons :

$$V_2(s) = \frac{V_1(s) + R_\pi \cdot C_2 \cdot V_i}{1 + R_\pi \cdot C_2 \cdot s} \quad (4.24)$$

En regroupant 4.20, 4.21 et 4.24 dans 4.23, nous obtenons :

$$\frac{I_{max}}{s} = s \cdot V_1(s) \cdot \frac{C_1 + C_2 + R_\pi \cdot C_2 \cdot C_1 \cdot s}{1 + R_\pi \cdot C_2 \cdot s} - V_i \cdot \frac{R_\pi \cdot C_1 \cdot C_2 \cdot s + (C_1 + C_2)}{1 + R_\pi \cdot C_2 \cdot s} \quad (4.25)$$

En isolant V_1 :

$$V_1(s) = \frac{V_i}{s} + \frac{I_{max}}{s^2} \cdot \frac{1 + R_\pi \cdot C_2 \cdot s}{C_1 + C_2 + R_\pi \cdot C_1 \cdot C_2 \cdot s} \quad (4.26)$$

La transformée de Laplace inverse nous permet d'obtenir :

$$v_1(t) = V_i + I_{max} \cdot \left(\frac{R_\pi \cdot C_2^2}{(C_1 + C_2)^2} \left(1 - e^{-\frac{C_1 + C_2}{R_\pi \cdot C_1 \cdot C_2} \cdot t}\right) + \frac{1}{C_1 + C_2} \cdot t \right) \quad (4.27)$$

Il nous reste donc à résoudre $v_1(t_f) = v_{dd}/2 = V_f$. Cette équation n'est malheureusement pas solvable analytiquement. Nous pouvons la résoudre par approximations successives en utilisant la méthode de Newton. Pour cela, nous réécrivons l'équation :

$$a \cdot (1 - e^{-c \cdot t_f}) + b \cdot t_f + V_i - V_f = 0 \quad (4.28)$$

Avec

$$a = \frac{R_\pi \cdot C_2^2}{(C_1 + C_2)^2} \cdot I_{max} \quad (4.29)$$

$$b = \frac{I_{max}}{C_1 + C_2} \quad (4.30)$$

$$c = \frac{C_1 + C_2}{R_\pi \cdot C_1 \cdot C_2} \quad (4.31)$$

Nous initialisons t_f à 0. La pente de l'équation 4.28 est donnée par la relation :

$$k = c \cdot a \cdot e^{-c \cdot t_f} + b \quad (4.32)$$

Et son terme constant est donné par :

$$m = a \cdot (1 - e^{-c \cdot t_f}) + b \cdot t_f + V_i - V_f - k \cdot t_f \quad (4.33)$$

L'équation d'itération pour déterminer $t_f^{(i)}$ est donc donnée par :

$$t_f^{(i)} = - \frac{a \cdot (1 - (1 + c \cdot t_f^{(i-1)}) \cdot e^{-c \cdot t_f^{(i-1)}}) + V_i - V_f}{c \cdot a \cdot e^{-c \cdot t_f^{(i-1)}} + b} \quad (4.34)$$

Une fois que t_f est calculé, nous déterminons la capacité équivalente par :

$$C_{eq} = \frac{t_f \cdot 2 \cdot I_{max}}{V_{dd}} \quad (4.35)$$

Nous réécrivons alors symboliquement l'équation 4.13 en remplaçant C par C_{eq} . L'intérêt de cette approche est sa simplicité d'implémentation dans le code existant de HiTas : seul le courant maximum I_{max} et les paramètres R et S sont nécessaires pour déterminer le temps de propagation de la porte chargée par une cellule en π .

4.1.4 Validation de l'approche

Pour valider l'approche précédente, nous allons procéder en trois temps. Tout d'abord, nous allons vérifier que la charge en π correspondant à un réseau RC est une approche qui donne de bons résultats, lorsque l'on considère que les émetteurs sont des générateurs de courant. Ensuite, nous vérifierons que les relations 4.13 et 4.35 donnent des résultats proches du temps de

propagation d'une porte modélisée par un générateur de courant calculé par le modèle MCC, donné par l'équation 3.5, et déchargeant une charge en π . Nous appliquerons cette méthode à deux cas de figure. La comparaison du modèle complet avec un simulateur électrique se fera dans le paragraphe 4.6. Finalement, nous étudierons le domaine de validité de ce modèle.

Limites de la représentation par une charge équivalente en π

Nous venons de voir que le calcul du temps de propagation de la porte passe par la détermination de la tension d'entrée de la ligne, ou du moins de la charge équivalente, lorsque la porte est approximée par un générateur de courant constant. Nous allons voir tout de suite que la première étape qui consiste à remplacer la ligne et les charges (ici une seule charge de $10fF$) par une cellule en π calculée par la méthode décrite dans [OL89] introduit une erreur au niveau temporel lorsque la source est effectivement un générateur de courant constant, pour des interconnexions longues.

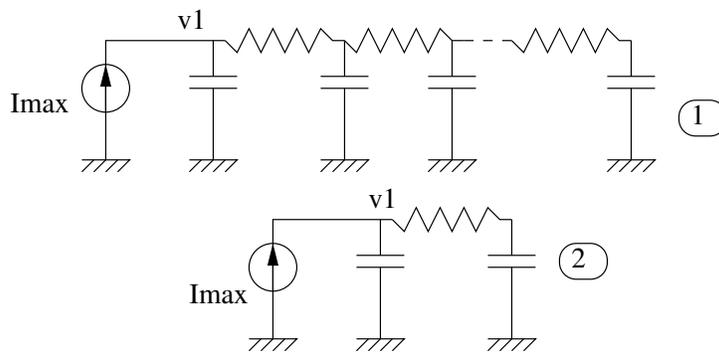


FIG. 4.5 – Générateur de courant constant chargé par la ligne complète (1) et par la cellule en π équivalente à la ligne (2)

Considérons les deux montages de la figure 4.5 que nous simulons sous Eldo. Nous avons tracé sur la figure 4.6 pour la ligne et son équivalent en π le temps que met le premier nœud ($v1$) de la charge à atteindre $v_{dd}/2$ en fonction de la longueur d'une ligne en ALU1, terminée par une capacité de $10fF$ représentant la charge d'entrée d'une porte. La ligne est composée de cellules élémentaires représentant chacune un tronçon de ligne de $1\mu m$.

On constate que lorsque le générateur de courant est chargé par la ligne, le temps de propagation croît de façon monotone jusqu'à une valeur constante lorsque la ligne atteint environ $2500\mu m$. Ensuite, l'augmentation de la longueur de la ligne ne se fait plus sentir au niveau de l'émetteur. Nous observons ici de façon évidente l'effet de masquage des résistances. Dans le cas de la cellule en π , on constate que jusqu'à une longueur de ligne de $2500\mu m$, le temps de

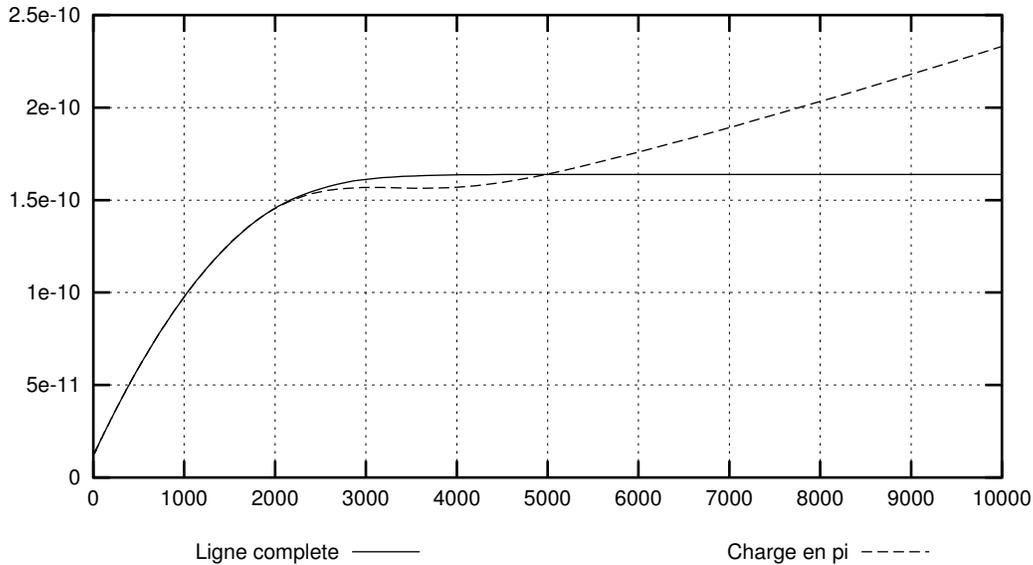


FIG. 4.6 – Temps de commutation du générateur de courant constant, en prenant comme charge la ligne complète et la charge en π du montage 4.5. Ordonnée : Temps (S), Abscisse : Longueur de la ligne (μm)

propagation est exact. Au-delà, bien que la charge augmente, on observe une stagnation -voire une diminution- du temps de propagation jusqu'à $5000\mu m$, puis le temps augmente à nouveau en fonction de la longueur de la ligne. Nous atteignons dans ce cas les limites du modèle, puisque la représentation au deuxième ordre fait apparaître une diminution des temps de propagation, et surtout ne prend pas en compte la limite supérieure du temps de propagation réel, puisque le modèle de la charge en π fait apparaître une pente dans le temps de propagation de la porte pour des lignes de très grande longueur au lieu d'une valeur constante. Heureusement, ces phénomènes apparaissent lorsque le générateur de courant correspond à un inverseur de taille 1, 5/3, à partir d'une longueur de ligne de $2,5mm$ en ALU1 de largeur minimum. Ces cas correspondent à des erreurs de routage, et produisent un délai dans la ligne RC très grand. Le concepteur a donc une indication que son routage est défectueux. La charge au second ordre est donc tout à fait adaptée.

Calcul du temps de propagation de la porte par simulateur MCC

Nous allons maintenant examiner si il est toujours possible de remplacer le temps de propagation obtenu avec un transistor absorbant depuis la charge au second ordre un courant donné

par l'équation 3.3 par la relation approchée donnée par l'équation :

$$Tp(C_{eq}, F) = R.C_{eq} + S.F \quad (4.36)$$

Pour obtenir le temps de propagation d'une porte, nous avons dû recourir à l'utilisation d'un simulateur algébrique dédié. La tension sur les nœuds aux bornes de la résistance de la figure 4.4 est donnée par les équation 4.9 à 4.12 :

$$\begin{aligned} v1_i &= X_1.(i_i + X_2.v1_j + X_3.v2_j) \\ v2_i &= X_4.v1_i + X_5.v2_j \end{aligned} \quad (4.37)$$

Les termes $v1_i$ et $v2_i$ correspondent aux tensions à l'instant t , et les tensions $v1_j$ et $v2_j$ sont les mêmes à l'instant $t - 1$. Le courant i_i est le courant instantané donné par la relation 3.5. Les termes X_n sont des constantes :

$$\begin{aligned} X_1 &= \left(\frac{C_1}{\Delta t} + \frac{C_2}{R.C_2 + \Delta t} \right)^{-1} \\ X_2 &= \frac{C_1}{\Delta t} \\ X_3 &= \frac{1}{\Delta t} \cdot \left(C_2 - \frac{R.C_2^2}{R.C_2 + \Delta t} \right) \\ X_4 &= \frac{\Delta t}{R.C_2 + \Delta t} \\ X_5 &= \frac{R.C_2}{R.C_2 + \Delta t} \end{aligned} \quad (4.38)$$

Ces équations sont obtenues en discrétisant les intégrales par la méthode des rectangles. Si nous voulions être plus précis, la méthode des trapèzes peut apporter un léger gain de précision. Le terme Δt correspond au pas de calcul. Nous déterminons ainsi T_s lorsque $v1_i$ atteint $vdd/2$. Nous obtenons $Tp = T_s - T_e$.

Application

Porte chargée par une ligne de longueur variable : Nous traçons sur la figure 4.7 les temps de propagation obtenus par simulation algébrique et ceux obtenus avec la relation 4.36 dans le cas d'une ligne de longueur variable, pour différentes valeurs de front d'entrée.

Dans le cas où le front est "rapide" $-F = 20pS$, c'est à dire dans la majeure partie des signaux dans les circuits CMOS, les résultats sont excellents : il est impossible de différencier les résultats obtenus par la résolution du système d'équation 4.9 à 4.12 page 92 de la solution approchée par l'équation 4.36. Pour $F = 50pS$, une légère différence apparaît entre les deux. Pour $F = 100pS$, cette différence augmente encore, mais l'écart reste acceptable, surtout si l'on compare ces résultats par rapport au temps que l'on calcule dans HiTas : ils correspondent à la tangente des courbes à l'origine.

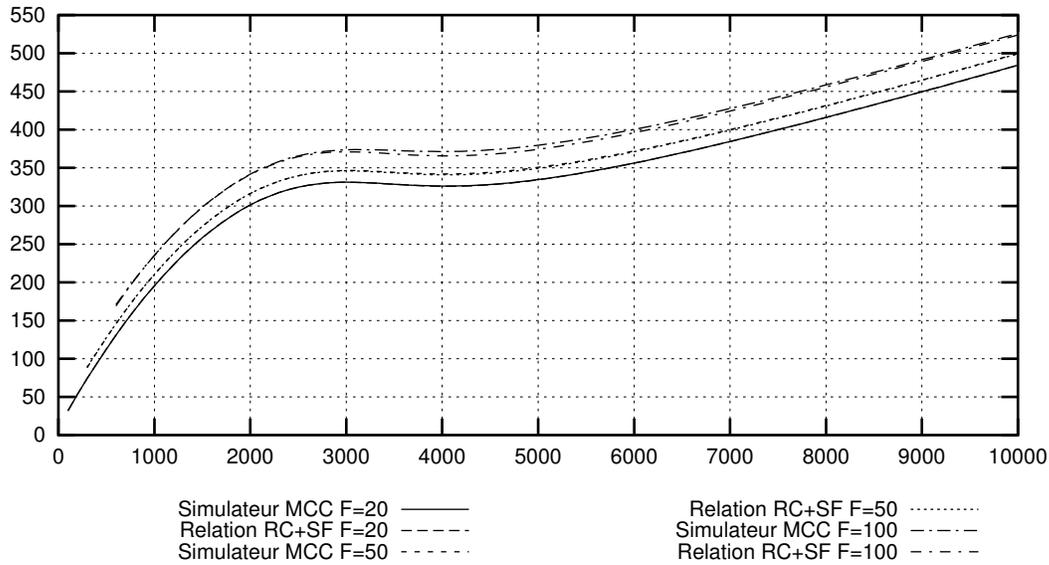


FIG. 4.7 – Temps de propagation d’une porte chargée par une cellule en π , calculée par le simulateur algébrique et par la relation 4.36. Ordonnée : Temps (pS), Abscisse : Longueur de la ligne (μm).

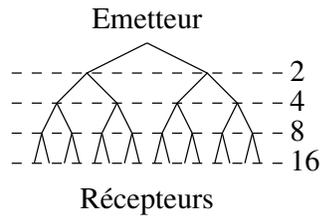


FIG. 4.8 – Réseau RC en forme d’arbre binaire

Porte chargée par un arbre : Considérons maintenant le cas de la figure 4.8 où les réseaux RC forment un arbre binaire. Chaque tronçon de ligne fait $100\mu m$ de long.

Le tableau 4.1 présente une comparaison des temps de propagation de la porte en fonction de la profondeur de l’arbre, en plaçant aux feuilles de l’arbre des charges de $10fF$, estimé par le simulateur algébrique et la relation 4.36.

Nous pouvons constater sur cet exemple que l’approximation de la charge au second ordre donne de bons résultats. Ceci n’est pas surprenant, car ne pas prendre en compte l’effet de masquage a une influence qui se fait plus sentir pour les longues lignes. A longueur de ligne totale donnée, l’arbre binaire nous pose beaucoup moins de problèmes.

Nombre de charges	Longueur totale de ligne (μm)	Temps de propagation simulé algébriquement (pS)	Temps de propagation calculé avec $Tp(C_{eq}, F)$ (pS)
2	200	54.9	55.0
4	600	120	120
8	1400	250	250
16	3000	507	507
32	6200	1021	1021
64	12600	2047	2047

TAB. 4.1 – Comparaison des temps de propagation entre le simulateur algébrique et l'équation 4.36 dans le cas d'un arbre binaire ($F = 20pS$).

Sensibilité du temps de propagation de la porte au front d'entrée

Nous avons tracé sur la figure 4.9 le temps de propagation d'une porte chargée par une cellule en π correspondant à une ligne de $500\mu m$ et de $1000\mu m$ en fonction du front d'entrée F .

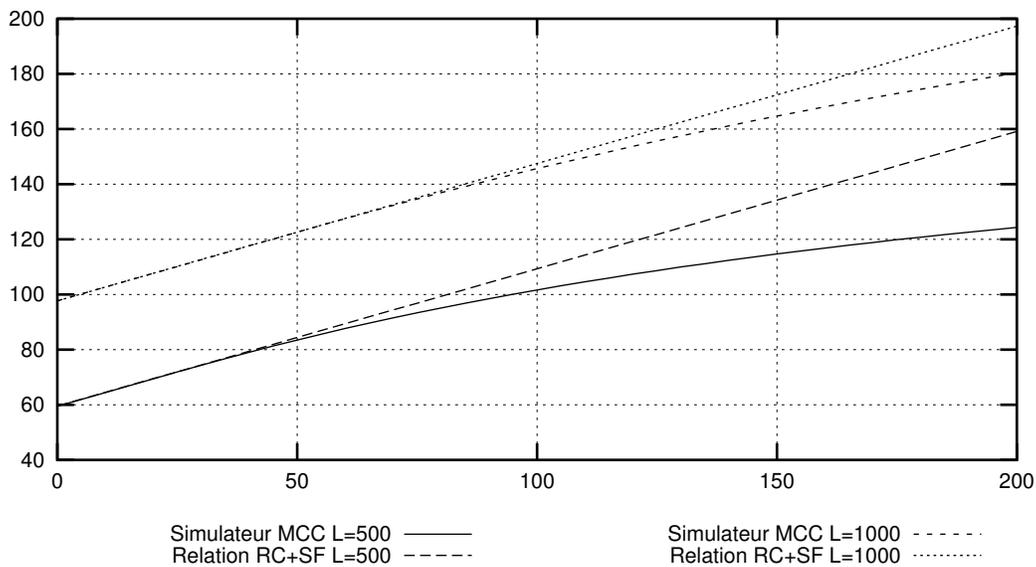


FIG. 4.9 – Comparaison des temps de propagation entre le simulateur algébrique et l'équation 4.36. Abscisse : front F 20%-80% (pS), Ordonnée : temps de propagation (pS).

Pour une longueur de ligne de $500\mu m$, le domaine pour lequel nous avons la relation $T_S >$

$2.F$ couvre les fronts entre 0 et $33,4pS$. Nous voyons sur la figure que dans cet intervalle, et même au-delà, la précision est très bonne. Dans le cas d'une ligne de $1000\mu m$, la relation $T_s > 2.F$ est valable jusqu'à $51,7ps$, ce que nous constatons également.

4.1.5 Conclusion

L'hypothèse de séparer dans le calcul du temps de propagation, d'une part le temps mis pour charger la charge par un courant constant I_{max} , et d'autre part la dépendance avec le front d'entrée reste donc valable dans de nombreux cas lorsque la charge est modélisée une cellule en π . Bien sûr, la solution qui consiste à déterminer systématiquement le temps de la porte avec une approche utilisant le simulateur algébrique est largement préférable en terme de précision dans le contexte d'une approche industrielle, et elle demeure tout à fait acceptable avec la puissance de calcul dont nous disposons actuellement, que ce soit en terme de MFLOP ou de nombre de processeurs. A titre indicatif, le simulateur tel que je l'ai implémenté détermine T_s en moins de $1ms$ (Sparc Entreprise à $450Mhz$) pour la charge au deuxième ordre. Pour le prototype d'analyseur temporel développé pendant ma thèse, je me suis limité au modèle équivalent (équation 4.36) qui offre une précision suffisante pour mettre en œuvre la méthode que je vais proposer dans le chapitre suivant.

Cette représentation de la charge par une cellule en π n'est possible qu'à l'approche au niveau transistor de HiTas. On trouve également dans la littérature quelques travaux sur les modèles de porte CMOS utilisant ces charges [HOT98]. Les outils (y compris les logiciels commerciaux) utilisant des tables pour calculer les délais ne sont pas capables de gérer simplement cette représentation, car la dimension des tables passe de 2 (front, capacité) à 4 éléments (front, $C1$, $C2$, R), ce qui revient à élever le nombre de valeurs de chaque table au carré. Une solution pour ces outils serait de fournir les paramètres d'une équation permettant de déterminer le temps de propagation de la porte.

4.2 Prise en compte du couplage

4.2.1 Présentation générale

L'objectif que nous nous fixons est de déterminer un modèle qui permette de calculer rapidement les temps de propagation minimum et maximum mettant en jeu un signal victime, susceptible d'être agressé par quelques uns de ses agresseurs. Nous avons vu dans la problématique que les simulations électriques qui permettent d'obtenir ces temps sont très coûteuses en temps CPU, et donc inadaptées au problème que nous cherchons à résoudre.

Le premier problème que nous rencontrons est la "diffusion" de cet effet. Considérons le cas de la figure 4.10 qui comporte 3 signaux, S1, S2 et S3, et supposons qu'il existe une capacité de couplage entre les signaux S1 et S2, et entre les signaux S2 et S3. Faisons l'hypothèse que S1

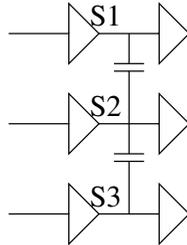


FIG. 4.10 – Diffusion de l'effet des couplages capacitifs

et S3 soient susceptibles de commuter en même temps. Il va bien sûr y avoir une influence de l'un sur l'autre. Tenter de résoudre ce problème est extrêmement complexe : si chaque signal a n agresseurs, cela veut dire qu'il faudra regarder l'état de n^2 autres signaux avant même de procéder au moindre calcul. Cela est beaucoup trop complexe, car la détection des agressions est un travail coûteux en temps CPU (nous allons le constater dans le chapitre 7). De plus, cette influence temporelle reste malgré tout assez faible. Il y a donc tout intérêt à ne considérer que les agresseurs directs pour un signal victime, et ne pas prendre en compte les agresseurs des agresseurs.

Cette première contrainte introduit *a priori* une imprécision dans le calcul des délais extrêmes. Cette contrainte a une influence pour la suite : il devient dès lors inutile de calculer de façon trop précise les délais extrêmes car notre hypothèse de départ introduit déjà une légère imprécision.

4.2.2 L'effet Miller.

Considérons le schéma (a) de la figure 4.11. L'effet Miller est la variation de la valeur apparente de la capacité vue par le premier nœud de la capacité lorsque la tension sur le second nœud varie. La détermination de cet effet est immédiat. Considérons les tensions définies par le chronogramme la figure (b). Les commutations ont lieu dans le même sens. Les tensions $v_1(t)$ et $v_2(t)$ sont des droites entre leur valeur initiale V_i et leur valeur finale V_f . Dans le cas où $T_1 < T_2$, calculons le courant qui circule dans la capacité au cours de la transition de $v_1(t)$:

$$i(t) = C_c \cdot \left(\frac{dv_1(t)}{dt} - \frac{dv_2(t)}{dt} \right) \quad (4.39)$$

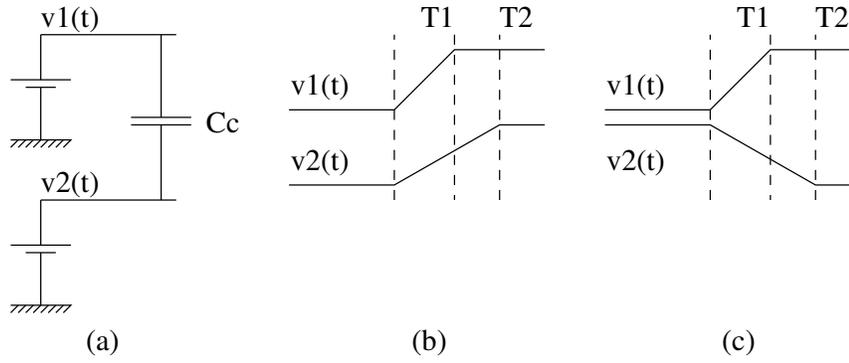


FIG. 4.11 – Définition de l'effet Miller

$$= C_c \cdot \left(\frac{vdd}{T1} - \frac{vdd}{T2} \right) \quad (4.40)$$

$$= C_c \cdot vdd \cdot \frac{T2 - T1}{T1 \cdot T2} \quad (4.41)$$

Calculons d'autre part le courant qui circulerait dans cette même capacité si le second nœud était relié à un potentiel fixe :

$$i(t) = C_{Miller} \cdot \frac{dv_1(t)}{dt} \quad (4.42)$$

$$= C_{Miller} \cdot \frac{vdd}{T1} \quad (4.43)$$

En identifiant les deux courants donnés par les équations 4.41 et 4.43, nous trouvons que :

$$C_{Miller} = C_c \cdot \frac{T2 - T1}{T2} \quad (4.44)$$

Pour le second nœud, le calcul est à peine plus compliqué : Pendant une partie de la transition la tension sur le premier nœud varie, et pendant l'autre partie, elle est constante.

$$\begin{cases} i(t) = C_c \cdot \left(\frac{dv_2(t)}{dt} - \frac{dv_1(t)}{dt} \right) & \text{de } 0 \text{ à } T1 \\ i(t) = C_c \cdot \frac{dv_2(t)}{dt} & \text{de } T1 \text{ à } T2 \end{cases} \quad (4.45)$$

Qui devient :

$$\begin{cases} i(t) = C_c \cdot vdd \cdot \frac{T1 - T2}{T1 \cdot T2} & \text{de } 0 \text{ à } T1 \\ i(t) = C_c \cdot vdd \cdot \frac{1}{T2} & \text{de } T1 \text{ à } T2 \end{cases} \quad (4.46)$$

La charge totale absorbée par la capacité au cours de la transition vaut :

$$Q = C_c \cdot vdd \cdot \frac{T1 - T2}{T1 \cdot T2} \cdot T1 + C_c \cdot vdd \cdot \frac{1}{T2} \cdot (T2 - T1) \quad (4.47)$$

Ce qui conduit à

$$C_{Miller} = 0 \quad (4.48)$$

Cas	Coefficient M
Transitions dans le même sens $F_v < F_a$	$\frac{F_a - F_v}{F_a}$
Transitions dans le même sens $F_v > F_a$	0
Transitions dans des sens opposés $F_v < F_a$	$\frac{F_a + F_v}{F_a}$
Transitions dans des sens opposés $F_v > F_a$	2

TAB. 4.2 – Les quatre configurations correspondant à l'effet Miller

Le même calcul dans le cas des tensions définies sur les chronogrammes (c) de la figure 4.11 donne pour $v_1(t)$:

$$C_{Miller} = C_c \cdot \frac{T1 + T2}{T2} \quad (4.49)$$

Et pour $v_2(t)$:

$$C_{Miller} = 2 \cdot C_c \quad (4.50)$$

L'utilisation de ce modèle dans l'évaluation des délais est intéressante, car elle permet de remplacer les capacités de couplage entre signaux par des capacités équivalentes reliées à la masse :

$$C_{eq} = M \cdot C_c \quad (4.51)$$

Où M correspond à l'une des valeurs du tableau 4.2. Insistons sur le fait que la valeur de la capacité équivalente n'est pas la même sur les deux signaux. La seule information de couplage nécessaire à ce modèle est la valeur du front sur l'agresseur. La différence de front permet de prendre en compte indirectement le rapport de force entre les signaux.

On trouve souvent dans la littérature ce modèle simplifié, qui ne prend en compte que le sens de transition des signaux, avec une approche $0.C - 2.C$. Cette approche simplifiée est un modèle pire cas, car les signaux rapides ($F_v < F_a$) seront évalués avec des capacités équivalentes trop importantes.

L'effet Miller s'appuie sur la variation des tensions issues de générateurs parfaits aux bornes de la capacité. Dans la pratique, la tension sur le signal résulte de l'absorption d'un courant par

le transistor. Plusieurs itérations de cet algorithme sont donc nécessaires afin d'obtenir la bonne valeur de C_{Miller} : Tout d'abord, les fronts en sortie des portes sont calculés en remplaçant les capacités de couplage par des capacité équivalentes $C_{Miller} = C$ à la masse. Puis, à partir des valeurs obtenues pour les fronts, les capacités C_{Miller} sont réévaluées suivant les quatre cas présentés dans le tableau 4.2. Il est donc possible de réévaluer les fronts, puis les capacités, et ainsi de suite, jusqu'à ce que le système converge. Dans la pratique, cette convergence est atteinte dès la seconde itération (stabilisation des fronts à $1ps$ près).

Dans [CAKK00b], un modèle tout à fait équivalent à celui qui vient d'être présenté à été publié. Les auteurs considèrent que la transition qui nous intéresse est celle entre 0 et $V_{dd}/2$ au lieu de 0 et V_{dd} . Cette constatation entraîne une modification des valeurs de l'effet Miller extrêmes $0.C$ et $2.C$ en $-1C$ et $3.C$. N'ayant pris connaissance de cette publication qu'à la rédaction de ce manuscrit, je n'ai pas pris en compte ce modèle, qui pourtant semble répondre tout à fait à notre critère de trouver un modèle simple, robuste et permettant d'obtenir les temps de propagation extrêmes. Même si la modification pour l'intégrer dans nos modèles est extrêmement simple, il faut cependant réaliser quelques tests dans le cas où l'effet Miller vaut -1 afin de vérifier si une "charge négative" peut être obtenue en sortie d'une porte, et si oui comment la traiter. Cependant, le modèle $0.C$ et $2.C$ que nous avons présenté ici fourni une bonne approximation des temps de propagation extrêmes, et nous verrons dans le paragraphe 4.4.2 comment nous réussissons néanmoins à obtenir le temps de propagation extrême d'une porte en introduisant le bruit de commutation.

4.3 Introduction de l'effet Miller dans le calcul du délai d'Elmore.

Nous avons décrit dans le paragraphe 3.4.2 et dans l'annexe F.1 la façon dont est calculé le premier moment de la fonction de transfert dans une interconnexion RC, et comment ce moment correspond au retard introduit par le réseau d'interconnexion. Nous allons maintenant voir quelle facilité l'effet Miller apporte pour prendre en compte la diaphonie dans les interconnexions.

Dans le cas particulier du délai d'Elmore, le temps de propagation entre deux nœuds d'un arbre RC s'écrit sous la forme :

$$Tp = \sum_{i \in \xi} \left(R_i \cdot \sum_{j \in \delta_i} C_j \right) \quad (4.52)$$

Où ξ désigne l'ensemble des nœuds sur les branches de l'arbre RC entre l'émetteur et le récepteur, et δ_i désigne l'ensemble des nœuds en aval du nœud i . Sur l'exemple représenté sur la

figure 4.12, nous avons :

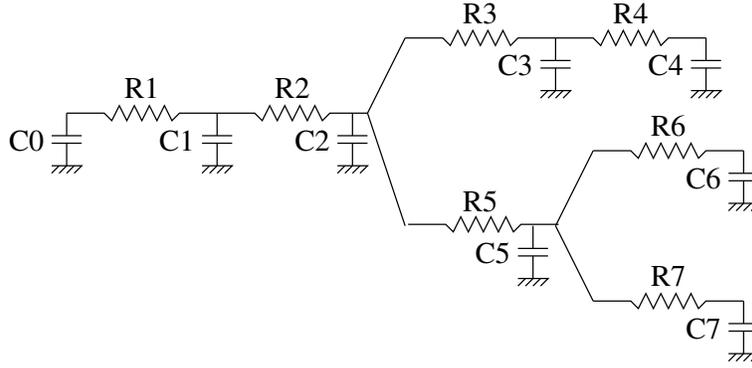


FIG. 4.12 – Délai d’Elmore. Exemple.

$$\begin{aligned}
 T_{p_{1 \rightarrow 6}} &= R_1.(C_1 + C_2 + C_3 + C_4 + C_5 + C_6 + C_7) \\
 &\quad R_2.(C_2 + C_3 + C_4 + C_5 + C_6 + C_7) \\
 &\quad R_5.(C_5 + C_6 + C_7) \\
 &\quad R_6.(C_6)
 \end{aligned} \tag{4.53}$$

L’effet Miller tel que nous l’utilisons revient à changer la valeur des capacités de couplage. Ecrivons cette influence sur le délai d’Elmore :

$$T_{p\Delta} = \sum_{i \in \xi} \left(R_i \cdot \sum_{j \in \delta_i} (C_j + \Delta C_j) \right) \tag{4.54}$$

$$= T_p + \sum_{i \in \xi} \left(R_i \cdot \sum_{j \in \delta_i} \Delta C_j \right) \tag{4.55}$$

$$= T_p + \sum_{j \in \omega} \left(\Delta C_j \cdot \sum_{i \in \psi_j} R_i \right) \tag{4.56}$$

Où ω est l’ensemble des nœuds sur l’arbre RC et ψ_j l’ensemble des nœuds communs à deux branches, la première allant de l’émetteur au nœud j , et la seconde allant de l’émetteur au récepteur. Lorsqu’on augmente la valeur d’une capacité sur un nœud j , le délai d’Elmore *varie* d’une valeur égale au produit de la somme des résistances sur ψ_j et de la *variation* de cette capacité.

Considérons maintenant le cas -très courant- où entre deux réseaux RC existent plusieurs capacités de couplage. Nous allons voir par la suite dans le paragraphe 4.5.2 que grâce à quelques raisonnements en pire cas, il est possible de considérer l’effet Miller M comme identique pour toutes les capacités de couplage entre deux réseaux RC. Chaque capacité entre deux

réseaux RC va donc être modifiée suivant :

$$\Delta C_j = C_j \cdot (M - 1) \quad (4.57)$$

où M correspond à l'une des valeurs décrites dans le tableau 4.2. L'équation 4.56 peut donc se réécrire :

$$T_p \Delta = T_p + \Delta T_p \quad (4.58)$$

$$= T_p + \sum_{j \in \omega_X} \left(C_j \cdot (M - 1) \cdot \sum_{i \in \psi_j} R_i \right) \quad (4.59)$$

$$(4.60)$$

Où ω_X est l'ensemble des nœuds sur lesquelles il existe une capacité entre le réseau RC courant et le réseau RC agresseur. Nous pouvons donc écrire :

$$\Delta T_p = (M - 1) \cdot K_x \quad (4.61)$$

où K_x vaut :

$$K_x = \sum_{j \in \omega_X} \left(C_j \cdot \sum_{i \in \psi_j} R_i \right) \quad (4.62)$$

Le résultat intéressant est que pour calculer la variation d'un délai d'Elmore dû à un effet diaphonique entre deux signaux, un seul paramètre constant, K_x , doit être mémorisé pour effectuer la correction. Lorsqu'un réseau RC possède n récepteurs et m signaux agresseurs, il faut donc mémoriser $n \times m$ valeurs. En pratique, n est très souvent faible, le nombre de variables à mémoriser pour réévaluer les délais d'Elmore est donc linéaire avec le nombre d'agresseurs.

4.4 Modèle de calcul des temps de propagation dans les portes en présence de diaphonie entre interconnexions résistives

4.4.1 Calcul de la charge équivalente de la porte en présence de diaphonie

On cherche à calculer le temps de propagation d'une porte lorsque son signal de sortie est soumis à une agression diaphonique. Lorsque la charge de la porte peut être considérée comme une capacité égale à la somme des capacités sur le réseau RC, ce problème est trivial : la variation de la charge apparente de la porte est égale à la variation des capacités de diaphonie par effet Miller. Nous allons étendre cette approche au cas où la charge de la porte est modélisée par une charge en π vue au paragraphe 4.1.

Nous cherchons à calculer la variation des paramètres y_1 , y_2 et y_3 définis dans le paragraphe 4.1 lorsqu'une ou plusieurs capacités varient sur le réseau. Pendant le parcours sur le réseau RC pour déterminer les y_i , il est possible de calculer également quelle sera la variation de ces paramètres lorsqu'une seule des capacités sur le réseau RC varie. Le détail de ce calcul est donné dans l'annexe F.3. La variation d'une capacité sur le réseau RC entraîne la modification des paramètres y_1 , y_2 et y_3 suivant les équations :

$$\begin{aligned}\Delta y_1 &= \Delta C & (4.63) \\ \Delta y_2 &= \alpha_2 \cdot \Delta C + \beta_2 \cdot \frac{\Delta C^2}{2} \\ \Delta y_3 &= \alpha_3 \cdot \Delta C + \beta_3 \cdot \frac{\Delta C^2}{2} + \gamma_3 \cdot \frac{\Delta C^3}{6}\end{aligned}$$

Lorsqu'une seule des capacités varie, il est donc possible de calculer immédiatement la variation de la charge apparente de la porte. Les paramètres à mémoriser sont 5 nombres flottants : α_2 , β_2 , α_3 , β_3 et γ_3 .

Malheureusement, ces équations ne sont pas linéaires en C . Cela signifie que les effets ne sont pas additifs si plusieurs capacités changent de valeur. La solution la plus précise serait donc de disposer du réseau RC complet, avec les capacités éventuellement modifiées, et de réévaluer la charge apparente à chaque fois qu'une capacité change. Néanmoins, nous avons réalisé quelques simulations sur des réseaux RC couplés, et observé ce qui se passe si nous linéarisons le système d'équations précédent, qui devient alors :

$$\begin{aligned}\Delta y_1 &= \sum_{i \in \omega} \Delta C_i & (4.64) \\ \Delta y_2 &= \sum_{i \in \omega} \alpha_{2i} \cdot \Delta C_i + \sum_{i \in \omega} \beta_{2i} \cdot \frac{\Delta C_i^2}{2} \\ \Delta y_3 &= \sum_{i \in \omega} \alpha_{3i} \cdot \Delta C_i + \sum_{i \in \omega} \beta_{3i} \cdot \frac{\Delta C_i^2}{2} + \sum_{i \in \omega} \gamma_{3i} \cdot \frac{\Delta C_i^3}{6}\end{aligned}$$

Où ω représente l'ensemble des capacités qui varient. En considérant les mêmes hypothèses que pour le délai d'Elmore, à savoir qu'entre deux signaux nous considérerons l'effet Miller constant, nous avons :

$$\begin{aligned}\Delta y_1 &= M \cdot x_{11} & (4.65) \\ \Delta y_2 &= M \cdot x_{21} + M^2 \cdot x_{22} \\ \Delta y_3 &= M \cdot x_{31} + M^2 \cdot x_{32} + M^3 \cdot x_{33}\end{aligned}$$

Avec

$$\begin{aligned}
 x_{11} &= \sum_{i \in \omega_X} C_i \\
 x_{21} &= \sum_{i \in \omega_X} \alpha_{2i} \cdot C_i & x_{22} &= \frac{\sum_{i \in \omega_X} \beta_{2i} \cdot C_i^2}{2} \\
 x_{31} &= \sum_{i \in \omega_X} \alpha_{3i} \cdot C_i & x_{32} &= \frac{\sum_{i \in \omega_X} \beta_{3i} \cdot C_i^2}{2} & x_{33} &= \frac{\sum_{i \in \omega_X} \gamma_{3i} \cdot C_i^3}{6}
 \end{aligned} \tag{4.66}$$

Etant donné le signal de sortie de la porte, pour chacun de ses agresseurs, il suffit donc de cinq paramètres pour réévaluer instantanément la charge de la porte au deuxième ordre. Lorsque plusieurs agresseurs sont actifs, il n'est en principe pas juste d'additionner ces effets. Dans de nombreux cas (voir paragraphe 4.6.2), cette approche fonctionne cependant assez bien. Dans une minorité d'autres cas (voir paragraphe 4.6.3), lorsque les couplages sont trop forts, la différence entre les y_i directement calculés à partir du réseau RC et les y_i calculés par les relations approchées devient significative, et au-delà d'une certaine limite les valeurs y_i ne correspondent plus à une charge au second ordre : les valeurs obtenues pour les capacités et la résistance peuvent être négatives. Dans ce dernier cas, nous utiliserons alors le modèle du premier ordre, à partir de y_1 qui est toujours calculé de manière exacte. Lorsque ce modèle du premier ordre est utilisé, nous obtenons les mêmes résultats que HiTas.

4.4.2 Influence du bruit de commutation sur les temps de propagation des portes

Lorsque nous calculons la charge apparente de la porte dans un contexte diaphonant, nous en déduisons un nouveau temps de propagation de la porte et un nouveau front de sortie (équation 3.10). Or, la seule modification de la charge apparente ne permet pas d'obtenir les temps de propagation extrêmes en présence de couplage [DTP97]. Nous allons donc tenter de répondre simplement à ce problème.

Nous avons constaté qu'il est possible de tenir compte d'un phénomène supplémentaire dans l'évaluation des délais : l'augmentation ou la diminution de la dynamique du signal. Ce phénomène est très simple à observer. Nous avons vu dans le chapitre 2.4 les aspects liés au bruit dû à la commutation d'un agresseur sur le signal victime fixe. Considérons la figure 2.22. Supposons qu'une transition du signal victime se produise à l'instant $t = 1,05ns$, la tension de départ de ce signal n'aurait pas été de $0V$, mais de l'ordre de $0,2V$. Par rapport à $1,25v$, cela représente une diminution de la dynamique de plus de 15%. La réduction de cette dynamique a un impact sur le temps de propagation de la porte.

Sur la figure 4.13, nous avons simulé la sortie d'un inverseur agressé (fortement) par un autre signal dans la configuration donnant le temps de propagation le plus petit ("signal réel"

sur la figure 4.13). Dans le meilleurs cas, l'effet Miller vaut 0. Nous avons superposé sur ce graphe la sortie de la porte lorsque les capacités de couplages sur ce signal sont simplement retirées ("signal Miller" sur la figure 4.13). Nous constatons qu'une différence significative existe entre ces deux courbes. Suivant le principe précédemment expliqué, nous avons superposé sur le graphe le bruit de commutation qu'aurait provoqué dans ces conditions l'agresseur sur la sortie de la porte si celle-ci n'avait pas commuté ("signal bruit" sur la figure 4.13). Finalement,

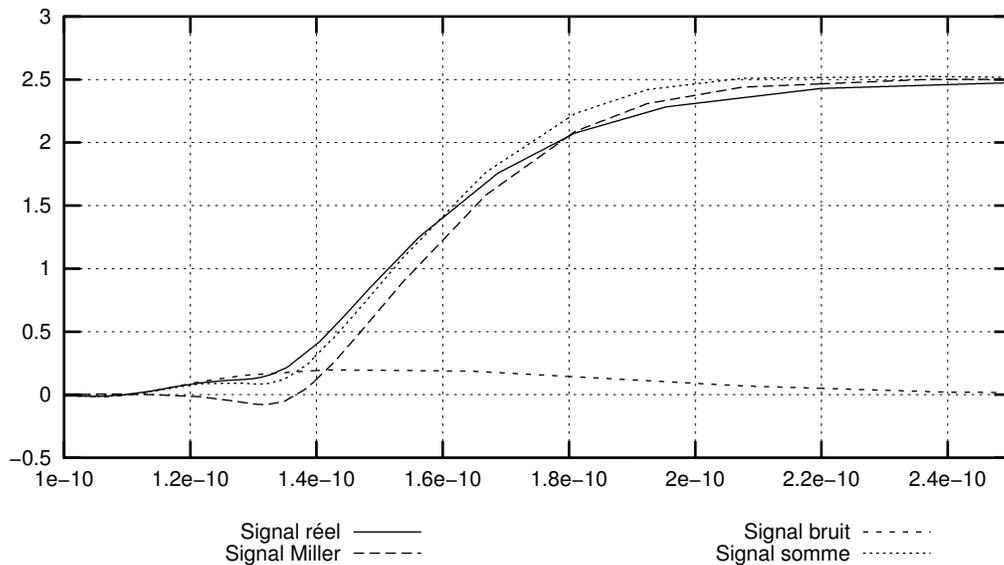


FIG. 4.13 – Prise en compte du bruit de commutation dans l'évaluation des temps de propagation minimum et maximum. Abscisse : temps (secondes), Ordonnée : tension (volts)

nous avons construit le signal somme en ajoutant le signal de sortie lorsque les capacités de couplage sont retirées et le signal de bruit ("signal somme" = "signal miller" + "signal bruit" sur la figure 4.13). Nous constatons que le signal somme et le signal réel sont très proches. Cette méthode permet de déterminer de façon plus précise les temps de propagation extrêmes de la porte.

Dans un circuit réel, nous avons plusieurs agressions à prendre en compte sur chaque victime. Chaque agresseur aura une influence qui lui sera propre, en hauteur de bruit et en durée d'influence. Pour résoudre ce problème, nous raisonnons en pire cas en faisant l'hypothèse que le bruit reste à sa valeur maximum durant toute la commutation.

4.4.3 Calcul du bruit de commutation

Lorsqu'on se place dans le cas des réseaux RC, il est bien plus difficile d'appliquer ce raisonnement, car la hauteur du bruit dépend de la position des nœuds agressés. Il est inutile de chercher à résoudre ce problème précisément dans le cadre de l'analyse temporelle car l'approche consistant à prendre en compte le changement de la dynamique des signaux est une approximation assez grossière. Il s'agit en réalité d'un ensemble de phénomènes liés, mais que pour des raisons de complexité nous isolons.

Nous avons donc été amenés à négliger les résistances d'interconnexion dans le calcul du bruit de commutation, et nous proposons ici un modèle de calcul présentant un bon compromis précision/vitesse. Le modèle électrique équivalent est représenté sur la figure 4.14. Nous rem-

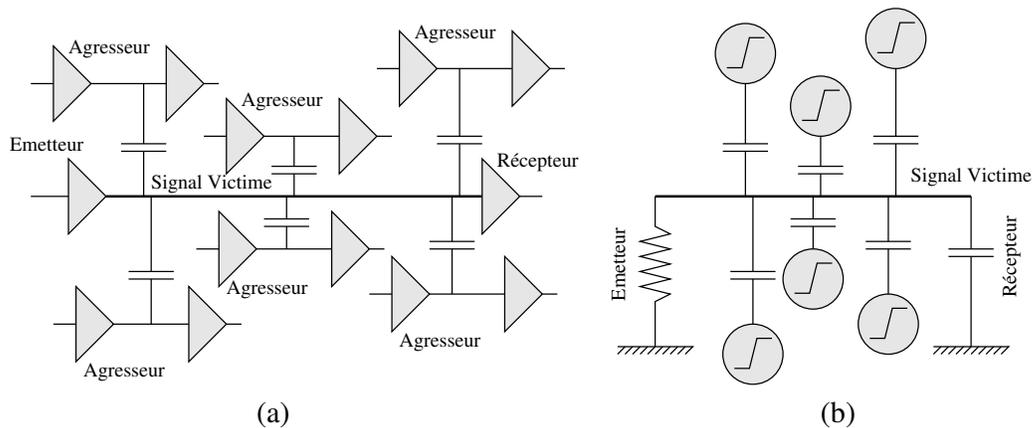


FIG. 4.14 – Modèle électrique utilisé pour déterminer approximativement le bruit de commutation

plaçons les transistors de la branche la plus résistive par leur résistance équivalente. Il s'agit là bien sûr d'une approximation car la résistance des transistors de la porte émettrice n'est pas constante selon la hauteur du bruit, comme le montre la mesure reportée sur la figure 4.15. Pour mesurer cette résistance, nous avons fixé la tension d'entrée d'un inverseur à V_{dd} afin d'avoir un 0 sur la sortie. Nous avons ajouté sur la sortie de la porte une source de tension très faible représentant un bruit extérieur, et le rapport entre cette tension et le courant absorbé par la porte nous donne la résistance équivalente de la porte. Nous constatons que cette résistance varie linéairement avec la tension. Nous n'allons pas non plus essayer de modéliser précisément ce phénomène pour les raisons précédemment expliquées. Dans la plupart des cas, le bruit est très faible. Nous prendrons donc arbitrairement comme valeur de résistance la valeur correspondant à 1/10 de la dynamique, soit $125mV$.

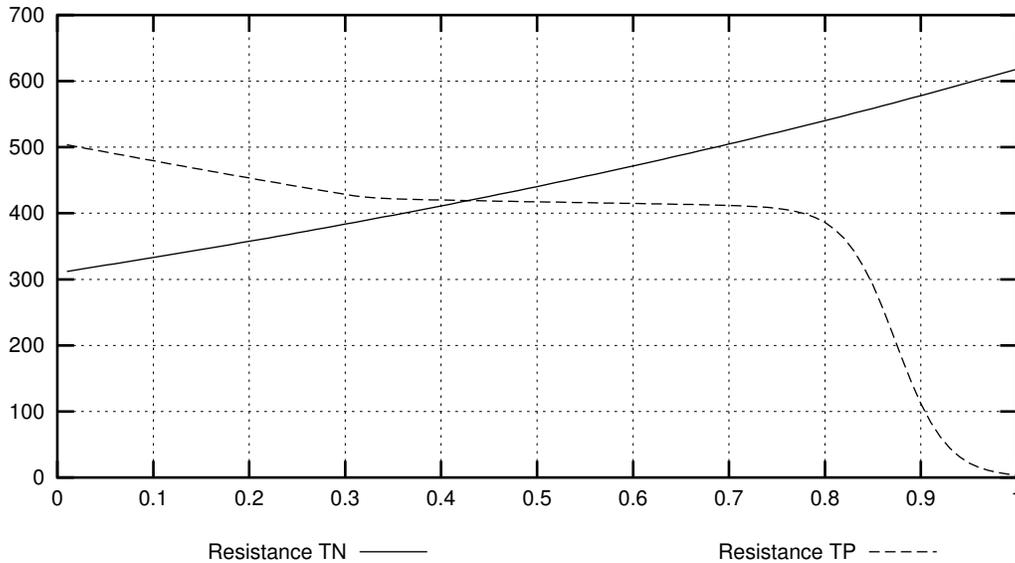


FIG. 4.15 – Résistance équivalente d’une porte lorsque sa sortie est soumise à un bruit. Abscisse : Hauteur du bruit (en volts), Ordonnée : Résistance équivalente (en Ω)

Reste le problème de calculer cette résistance de façon systématique. Nous avons procédé de manière empirique. Nous avons relevé par simulation électrique, sur la figure 4.16, la variation de la résistance en petits signaux d’un transistor lorsque nous faisons varier la longueur L de la grille, pour différentes largeurs W . Réciproquement, nous avons relevé sur la figure 4.17 cette résistance lorsque nous faisons varier la largeur W de la grille, pour différentes longueurs L de grille. A partir de ces courbes, nous constatons que cette résistance peut être représentée par la relation :

$$r = \frac{L.a + b}{W.c + 1} \quad (4.67)$$

Les paramètres a , b et c étant des paramètres liés à la technologie. Pour la technologie en $0,25\mu m$ que nous utilisons, les valeurs de a , b et c sont :

$$\begin{aligned} a_n &= -3.093e + 10 & a_p &= -1.056e + 11 \\ b_n &= +1.932e + 03 & b_p &= +1.169e + 04 \\ c_n &= -1.473e + 07 & c_p &= -1.269e + 07 \end{aligned} \quad (4.68)$$

Une fois calculée la résistance en petits signaux de la porte, il faut calculer le bruit à partir du schéma de la figure 4.14. Pour cela, nous calculons le bruit élémentaire provoqué par chacun des agresseurs, en considérant tous les autres statiques. Le bruit total sera égal à la somme des bruits élémentaires. Chaque bruit élémentaire se calcule à l’aide du schéma équivalent de la

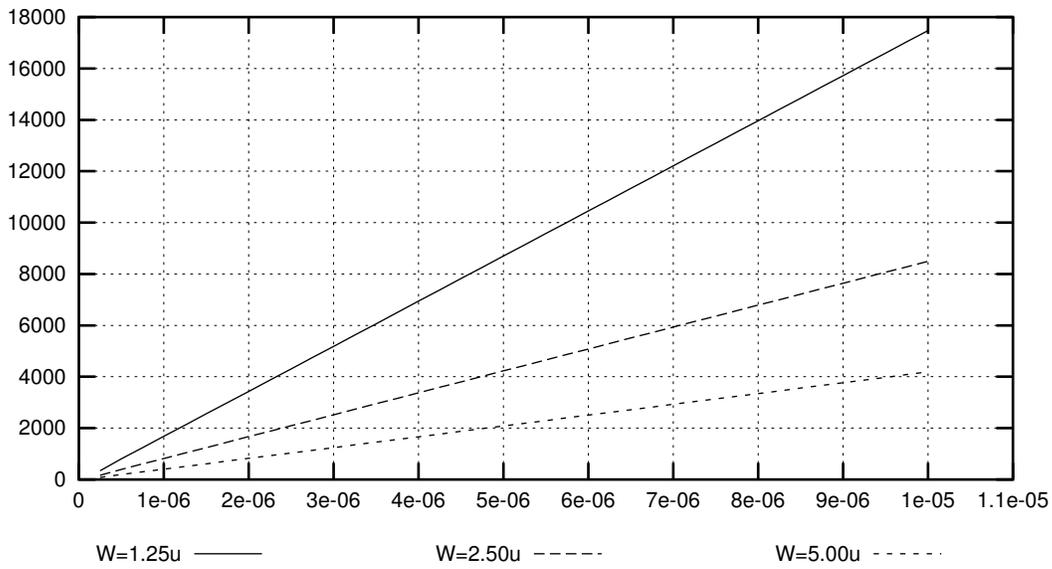


FIG. 4.16 – Résistance équivalente du transistor en fonction de L pour différentes valeurs de W . Abscisse : L en mètres, Ordonnée : Résistance en Ω

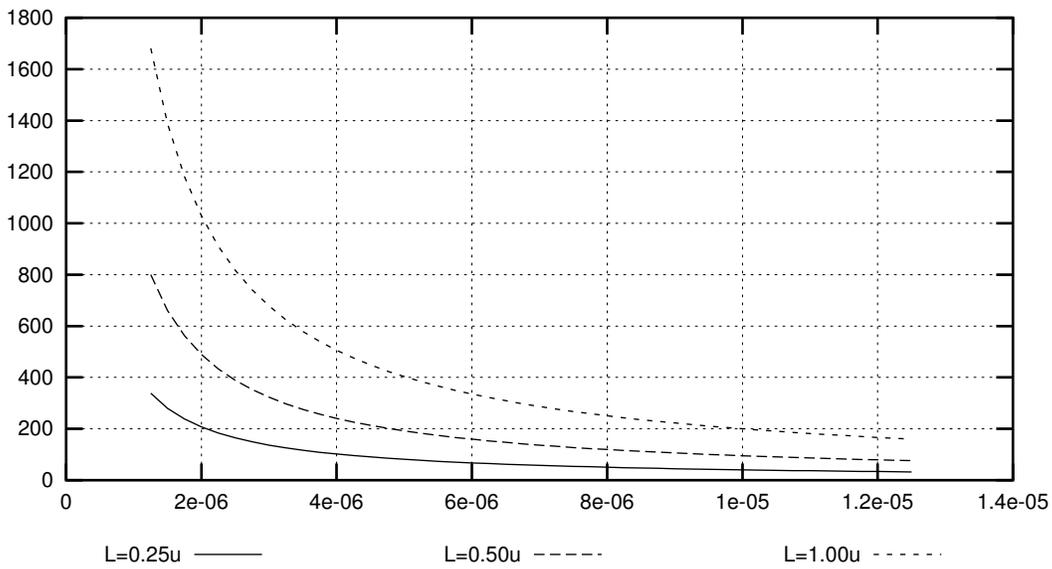


FIG. 4.17 – Résistance équivalente du transistor en fonction de W pour différentes valeurs de L . Abscisse : W en mètres, Ordonnée : Résistance en Ω

figure 4.18. La résistance R_v représente la résistance de la victime, la capacité C_c la somme algébrique des capacités entre la victime et l'agresseur considéré, la capacité C_m la capacité de charge de la victime à la masse, qui inclut la capacité de routage et les capacités de charge, et

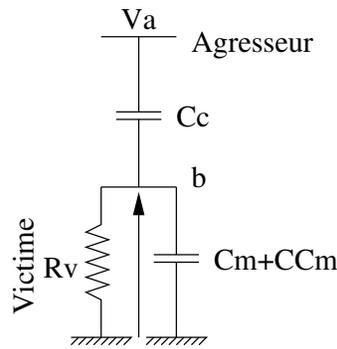


FIG. 4.18 – Schéma équivalent pour le calcul du bruit produit par un agresseur.

finalement la capacité CC_m représente l'ensemble des capacités de couplage entre les autres agresseurs.

Lorsque le front sur l'agresseur est considéré comme une droite passant de V_0 à V_1 en un temps T , la tension aux bornes de la résistance - le bruit - se calcule avec la relation :

$$v(t) = (V_1 - V_0) \frac{R_v \cdot C_c}{T} \cdot \left(\left(1 - e^{-\frac{t}{R_v \cdot (C_c + CC_m + C_m)}} \right) \cdot U(t) - \left(1 - e^{-\frac{t-T}{R_v \cdot (C_c + CC_m + C_m)}} \right) \cdot U(t - T) \right) \quad (4.69)$$

Où la fonction $U(t)$ est la fonction échelon, qui vaut 0 si $t < 0$ et 1 si $t \geq 0$. En pratique, cette équation n'est pas utile car le bruit maximum est atteint lorsque $t = T$. Dans ce cas la valeur du bruit s'obtient directement à partir de l'équation :

$$bruit = (V_1 - V_0) \cdot \frac{R_v \cdot C_c}{T} \cdot \left(1 - e^{-\frac{T}{R_v \cdot (C_c + CC_m + C_m)}} \right) \quad (4.70)$$

Nous avons admis dans la partie 3.4.1 que le front d'un signal est proche d'une tangente hyperbolique. Nous assimilerons ici cette transition à une droite, ce qui conduira à une légère surestimation du bruit. HiTas fournit la valeur de F_a , qui représente le temps de propagation du signal entre 20% et 80% de sa dynamique. On en déduit :

$$T = \frac{F_a}{0,6} \quad (4.71)$$

La hauteur du bruit est donnée par :

$$bruit = 0,6 \cdot V_{dd} \cdot \frac{R_v \cdot C_c}{F_a} \cdot \left(1 - e^{-\frac{F_a}{0,6 \cdot R_v \cdot (C_c + CC_m + C_m)}} \right) \quad (4.72)$$

4.4.4 Correction sur les temps de propagation

Une fois le bruit total calculé, et considéré comme constant, le signal de sortie de la porte est translaté vers le haut, comme cela est montré sur la figure 4.19. L'avance ou le retard qui en

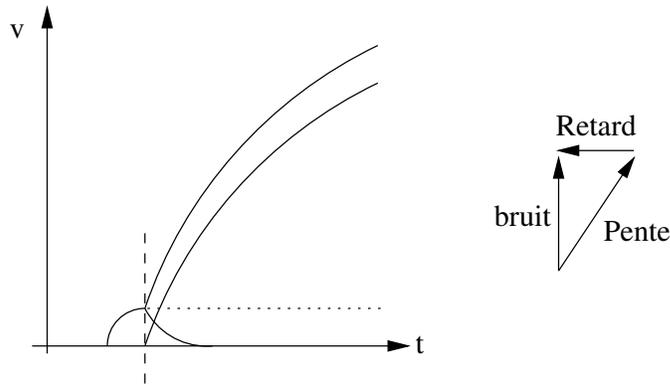


FIG. 4.19 – Calcul de la correction à partir de la hauteur du bruit et de la pente du signal

résulte se calcule simplement à partir de la pente du signal victime :

$$\Delta T_p = \frac{\text{bruit} \cdot F_v}{0,6 \cdot V_{dd}} \quad (4.73)$$

Ces opérations sont réalisées dans quatre contextes d'agression :

- Signal victime montant, calcul de Tp_{max} . Seuls les agresseurs dont la transition est descendante sont pris en compte pour le calcul du bruit. La correction est positive.
- Signal victime montant, calcul de Tp_{min} . Seuls les agresseurs dont la transition est montante sont pris en compte pour le calcul du bruit. La correction est négative.
- Signal victime descendant, calcul de Tp_{max} . Seuls les agresseurs dont la transition est montante sont pris en compte pour le calcul du bruit. La correction est positive.
- Signal victime descendant, calcul de Tp_{min} . Seuls les agresseurs dont la transition est descendante sont pris en compte pour le calcul du bruit. La correction est négative.

4.5 Evaluation des délais extrêmes avec interconnexions résistives diaphonantes

4.5.1 Le graphe de causalité

Nous cherchons à construire le graphe de causalité (paragraphe 1.4.4) d'un circuit comportant des résistances d'interconnexion et des capacités de couplage. Nous ne pouvons pas attribuer une valeur correcte aux délais élémentaires avec une seule analyse temporelle statique car ces délais dépendent d'un contexte diaphonant. En effet, dans le paragraphe 4.2, nous avons montré que le couplage entre interconnexions pouvait être modélisé par une variation de

capacité équivalente, et dans les paragraphes 4.3 et 4.4 nous avons présenté les conséquences de cette variation sur le calcul des délais élémentaires des interconnexions et des portes. En supposant que nous connaissions pour chaque signal la liste des agresseurs qui sont actifs ou non, l'évaluation des délais élémentaires, à l'aide des équations présentées dans les paragraphes 4.3 et 4.4, permet de calculer les fronts en sortie des portes. Les fronts sur les agresseurs actifs modifient à nouveau les capacités équivalentes, et les délais doivent être réévalués. Voyons ici comment les modèles des délais élémentaires des paragraphes 4.2 à 4.4 peuvent être utilisés pour résoudre ce problème couplé. L'idée est de prendre en compte le phénomène dynamique de la diaphonie en itérant l'estimation des fronts résultant d'une analyse de stabilité classique.

Première itération

Le temps de propagation d'une porte dépend de sa charge et du front de l'entrée qui commute, ces deux informations devant être déterminées en fonction du contexte diaphonant. La première itération se fait en supposant qu'il n'y a pas de couplage actif entre les signaux. Nous remplaçons chaque capacité de couplage par deux capacités équivalentes de même valeur reliées à la masse. Nous pouvons de cette façon calculer la charge apparente sur chaque porte. La détermination du front d'entrée se fait en deux temps. Tout d'abord, nous calculons un front de manière sommaire à l'entrée de chaque porte. A partir de ce front et de la charge apparente de la porte, nous sommes en mesure de calculer un front de sortie plus réaliste en sortie de chaque porte, et de le transmettre à l'entrée portes de charge à travers les interconnexions. A la fin de cette première itération, nous avons donc déterminé les fronts sur tous les nœuds du circuit. Notons que les temps de propagation n'ont pas été calculés.

Itérations suivantes

A partir des valeurs des fronts sur les nœuds, il est possible en utilisant les relations présentées dans le tableau 4.2 de calculer les valeurs apparentes des capacités de couplage. Nous en déduisons les charges équivalentes sur chaque porte. Ces nouvelles charges équivalentes vont entraîner une variation des fronts de sortie des portes, qui vont avoir deux impacts : la modification des valeurs apparentes des capacités de couplage et la modification des fronts d'entrée des portes. Les fronts sont donc réévalués, et ainsi de suite jusqu'à ce que l'ensemble converge. Nous considérons que le système a convergé dès lors que les fronts ne varient plus de plus d'une pico-seconde (unité de calcul de HiTas).

Calcul des délais

A partir de la valeur définitive des fronts, nous pouvons évaluer le temps de propagation des portes ainsi que dans les interconnexions. Finalement la correction des délais due au bruit est appliquée.

Algorithme

L'algorithme que je viens de dérouler est présenté sur la figure 4.20.

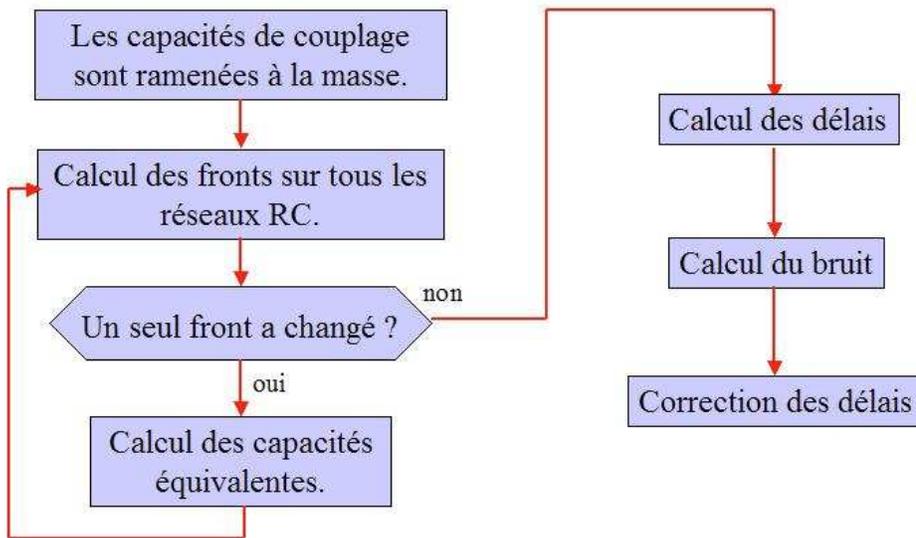


FIG. 4.20 – Algorithme pour déterminer le temps de propagation des portes d'un circuit.

Conclusion

Cette approche a le mérite de fournir la meilleure précision que permettent les modèles présentés dans ce chapitre. Néanmoins, la charge de calcul et la consommation mémoire sont importantes. J'ai donc tenté de contourner ces difficultés en utilisant les relations approchées présentées dans les équation 4.64 à 4.66 page 107.

4.5.2 Utilisation des modèles linéaires

Hypothèses sur les fronts

Dans de nombreux cas, les extracteurs fournissent, entre deux signaux, plusieurs capacités de couplage placées en différents nœuds des réseaux RC, comme cela est le cas sur l'exemple de la figure 4.21. Et d'une manière générale, chaque nœud comporte une ou plusieurs capacité de couplage vers d'autres signaux. Nous savons également que plus un agresseur a un front relativement raide, plus il aura d'effet sur le comportement temporel de la victime. Afin de diminuer le nombre de fronts à calculer, nous majorons l'effet d'un agresseur sur une victime en faisant deux choix :

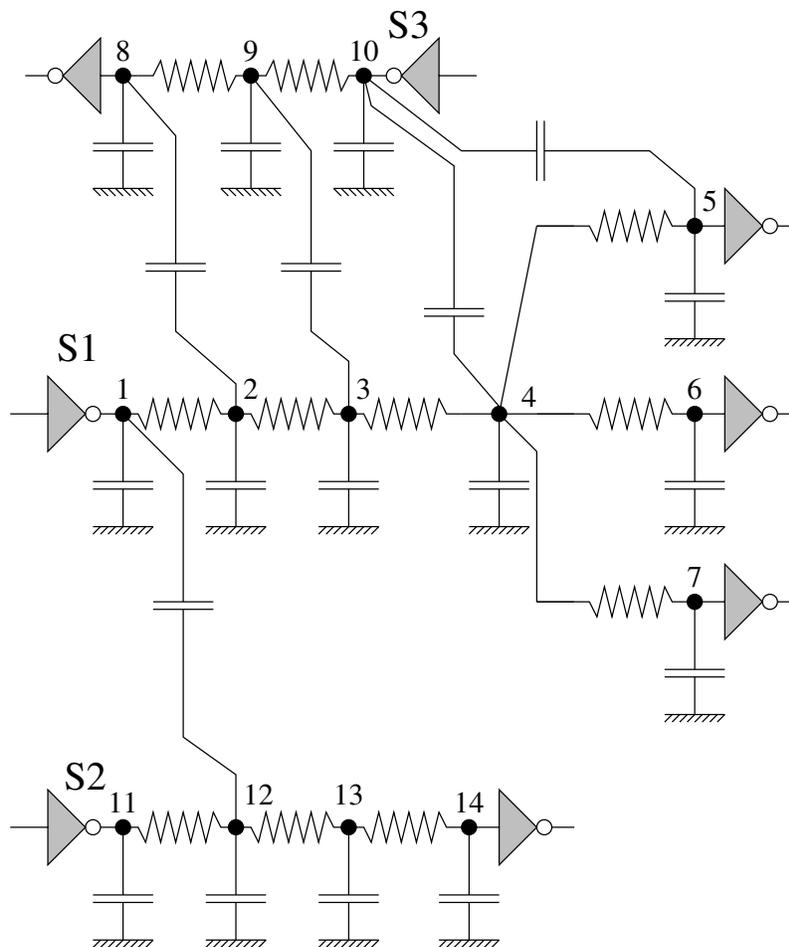


FIG. 4.21 – Exemple de circuits couplés

- Le front de la victime est le moins rapide possible. On choisit le minimum des fronts à l'entrée des portes de charge.

- Le front de l’agresseur est le plus rapide possible. On considère le front directement en sortie de la porte qui contrôle le signal agresseur.

Cela revient à considérer l’effet Miller identique sur toutes les capacités de couplage entre deux signaux donnés. Nous diminuons ainsi très sensiblement le nombre de front à déterminer, puisque nous passons de plusieurs dizaines (voir milliers) de calcul par signal à quelques unités. Ce chiffre est indépendant des modèles de délais utilisés pour les portes ou les interconnexions.

Conséquence sur l’évaluation des charges des portes

Pour l’évaluation des temps de propagation des portes, nous modélisons leur charge par une cellule en π . Avec l’hypothèse que l’effet Miller est identique sur toutes les capacités de couplage entre deux signaux, nous pouvons appliquer la relation 4.66 pour déterminer la variation de l’admittance vue par la porte. Le gain est alors très sensible pour deux points :

- En mémoire, il n’est plus nécessaire de conserver la description du réseau RC puisque le calcul des 5 paramètres permettant de calculer la variation de la charge apparente n’est effectué qu’une seule fois.
- En temps de calcul, le gain est également important car la détermination de la charge apparente est linéaire en fonction du nombre de nœuds sur le signal, tandis que grâce à la relation 4.66, elle devient linéaire en fonction du nombre d’agresseurs.

La charge en π étant ainsi modélisée, la capacité équivalente C_{eq} est déterminée par la relation 4.35, puis le délai de la porte est évalué avec la relation $R.C_{eq} + S.F$. Le front de sortie est calculée avec la relation :

$$F = F_0 + R.\Delta C_{eq} \quad (4.74)$$

Nous considérons que la variation du front de sortie ne dépend pas de la variation du front d’entrée. Ce point, ainsi que le calcul de cette variation, sont discutables et améliorables. Nous verrons dans le chapitre suivant que l’hypothèse de considérer la variation du front de sortie indépendante de la variation du front d’entrée apporte une simplification très importante à la méthode retenue.

Conséquence sur l’évaluation des temps dans les interconnexions

Dans HiTas, l’évaluation des délais dans les interconnexions se fait en utilisant un modèle de charge au premier ordre : le délai d’Elmore. Si, dans le cas où les capacités de couplage ne sont pas prises en compte la méthode AWE apporte un gain de précision, je considère que dans le cas où elles sont prises en compte l’apport en précision devient superflu. Nous essayons de

déterminer des temps de propagation extrêmes selon un critère rudimentaire : un agresseur est actif ou non. Indépendamment de la précision des modèles utilisés, le contexte dans lequel nous appliquons ces modèles est basé sur une détermination pire cas. De même, le remplacement des capacités de couplage par des capacités équivalentes reliées à la masse, l'effet de bruit et les autres simplifications entraînent forcément une perte de précision. L'utilisation de l'algorithme AWE calculera de façon précise des délais sur des réseaux modélisés de façon approximative. De plus, cet algorithme est un peu plus coûteux en temps de calcul. L'utilisation du délai d'Elmore offre un bon compromis entre précision et vitesse.

Lorsque l'effet Miller est considéré comme identique sur toutes les capacités entre deux signaux, nous avons vu que l'effet d'un agresseur modifie un délai d'Elmore avec la relation 4.61. Là encore, la description en mémoire du réseau RC n'est plus nécessaire car le coefficient K_x défini par la relation 4.62 est déterminé une fois pour toute.

4.6 Validation

Nous allons maintenant montrer quelques résultats obtenus avec ces modèles. Ces modèles sont implémentés dans l'outil Ctk, qui sera présenté dans le chapitre 6. Ctk calcule les temps de propagation des portes avec une charge au second ordre en utilisant les relations 4.1 page 89, 4.35 et 4.13. Lorsque les agresseurs actifs sont connus, Ctk calcule les délais extrêmes avec les relations 4.61, 4.66 et 4.74, et en prenant en compte le bruit de commutation avec les relations 4.72 et 4.73.

Les délais de référence ont été calculés avec Eldo, et lorsque tous les agresseurs sont muets, nous avons comparé les délais avec HiTas.

4.6.1 Effet du modèle de charge au deuxième ordre

Nous avons tracé sur la figure 4.22 le temps de propagation d'une porte chargée par une ligne de longueur variable et par une porte placée à l'extrémité de la ligne, sans prendre en compte les effets diaphoniques. Le temps de propagation de la porte est calculé par Eldo, HiTas et Ctk. Comme prévu, l'erreur est très largement atténuée dans Ctk par rapport à HiTas grâce à l'introduction du modèle de charge au second ordre.

4.6.2 Couplage de valeurs typiques

Considérons maintenant le réseau représenté sur la figure 4.23. Les interconnexions représentées en trait épais correspondent à des fils en métal 2 de $1000\mu m$ de long, et les couplages

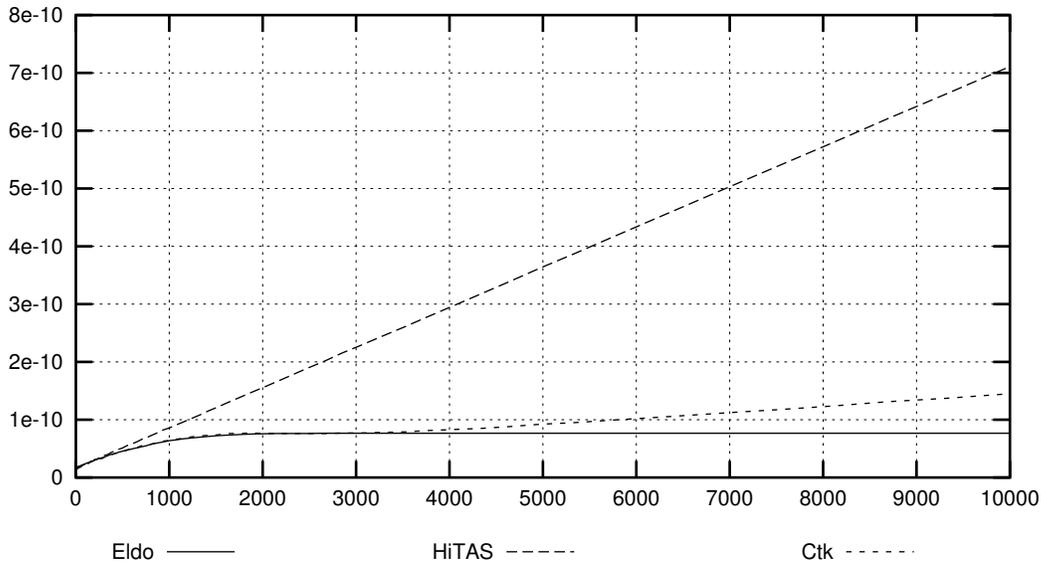


FIG. 4.22 – Temps de propagation d’une porte chargée par une ligne résistive. Abscisse : longueur de la ligne en μm . Ordonnée : Temps de propagation de la porte (secondes).

correspondent à des fils en métal 1 de $100\mu m$ les plus proches possibles. Nous avons considéré plusieurs cas de figure. Tout d’abord, en ne considérant qu’un seul agresseur actif (Ag1 ou Ag2), puis, en considérant plusieurs agresseurs actifs. Les simulations sous Eldo de ce circuit pour déterminer les temps de propagation extrêmes prennent à peu près 8 heures, pour une soixantaine de transistors, et pour chaque configuration d’agresseur. Ce chiffre rappelle que l’usage des simulateurs électriques n’est plus adapté. Le temps d’exécution de Ctk se situe autour d’une seconde.

Comme nous le constatons sur le tableau 4.3, le modèle proposé implémenté dans Ctk donne des résultats très satisfaisants par rapport à Eldo, puisque l’erreur maximum commise n’est que de 4,5%. L’erreur commise de 6,3% sur le délai minimum du nœud 4 lorsque les agresseurs Ag1, Ag2 et Ag3 sont actifs signifie que le délai obtenu par le modèle Ctk est inférieur au délai obtenu par Eldo, ce qui n’est pas un problème dans le cas d’une approche pire cas, tant que l’ordre de grandeur est respecté. Ces résultats sont satisfaisants car ils montrent que le modèle utilisé permet de bien cerner les variations des délais dus à la diaphonie, grâce à la variation de charge apparente et à la modification des délais d’Elmore.

Sur cet exemple, la vue Ctk est très compacte, il y a $4 \times 5 = 20$ nombres flottants pour calculer la variation de charge apparente de la porte en fonction des agressions, et $4 \times 4 \times 1 = 16$

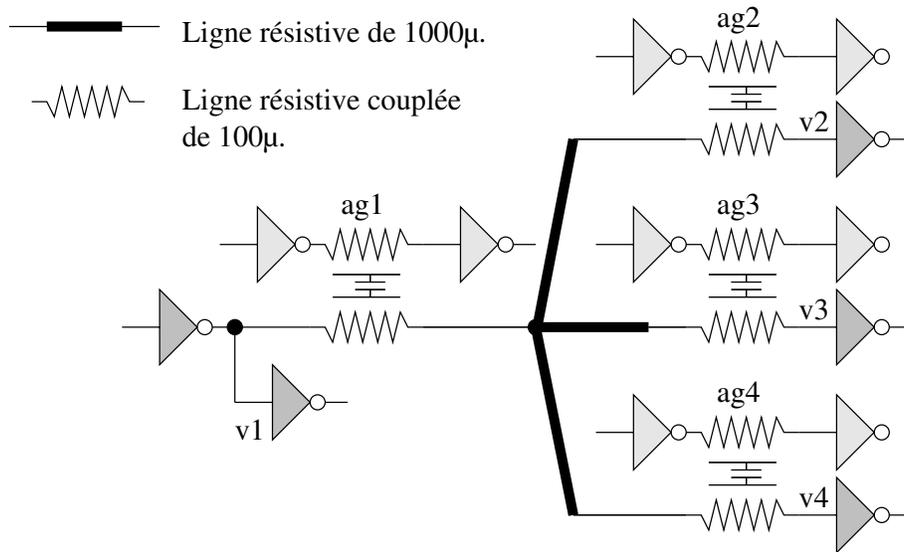


FIG. 4.23 – Exemple d’un circuit couplé possédant des interconnexions résistives.

nombres flottants (il y a quatre agresseurs et chaque agresseur modifie les quatre temps de propagation des émetteurs vers les récepteurs) pour calculer les variations des délais d’Elmore. Le détail de la répartition des temps de propagation calculés par Ctk entre le délai de porte et le délai d’Elmore est donné sur le tableau 4.4.

4.6.3 Cas d’un couplage fort de deux agresseurs

Une porte est chargée par une autre porte à travers un fil de 200μm en métal de premier niveau. Ce fil est agressé par deux signaux, chacun sur 100μm. La figure 4.24 représente le schéma électrique de cette configuration. Nous avons considéré le cas du couplage le plus fort,

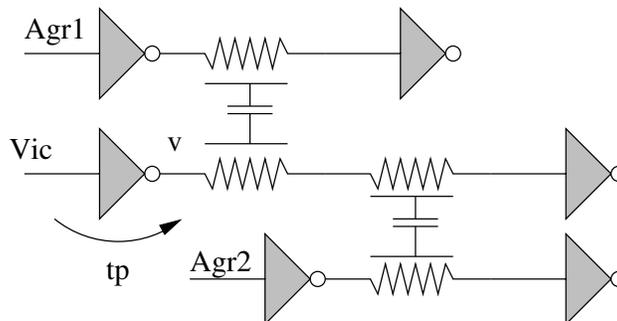


FIG. 4.24 – Signal victime de 200μm agressé par deux signaux, chacun sur 100μm.

Agresseurs	Nœud V1		Nœud V2		Nœud V3		Nœud V4	
	Eldo	Ctk	Eldo	Ctk	Eldo	Ctk	Eldo	Ctk
	Délais nominaux							
	273	268 (-1,8%)	304	295 (-3,0%)	304	295 (-3,0%)	304	295 (-3,0%)
	Délais maximum							
Ag1	290	286 (-1,4%)	320	313 (-2,2%)	320	313 (-2,2%)	320	313 (-2,2%)
Ag2	290	285 (-1,7%)	331	313 (-5,4%)	320	312 (-2,5%)	320	312 (-2,5%)
Ag1 et Ag2	305	303 (-0,7%)	347	331 (-4,6%)	336	330 (-1,8%)	336	330 (-1,8%)
Ag1 à Ag3	323	321 (-0,6%)	364	349 (-4,1%)	365	349 (-4,4%)	353	348 (-1,4%)
Ag1 à Ag4	339	337 (-0,6%)	382	365 (-4,5%)	382	365 (-4,5%)	382	365 (-4,5%)
	Délais minimum							
Ag1	257	250 (-2,7%)	288	276 (-4,2%)	288	276 (-4,2%)	288	276 (-4,2%)
Ag2	257	250 (-2,7%)	281	275 (-2,1%)	287	276 (-3,8%)	287	276 (-3,8%)
Ag1 et Ag2	241	233 (-3,3%)	265	257 (-3,0%)	272	258 (-5,1%)	272	258 (-5,1%)
Ag1 à Ag3	226	216 (-4,4%)	250	239 (-4,4%)	250	239 (-4,4%)	256	240 (-6,3%)
Ag1 à Ag4	211	199 (-5,7%)	235	221 (-6,0%)	235	221 (-6,0%)	235	221 (-6,0%)

TAB. 4.3 – Comparaison des temps de propagation extrêmes entre l'entrée et différents nœuds du circuit de la figure 4.23 évalués par Eldo et Ctk en fonction des agressions. Les délais nominaux sont évalués avec un état stable des agresseurs.

Agresseurs	Nœud V1	Nœud V2	Nœud V3	Nœud V4
	Délais nominaux			
	268	268+27=295	268+27=295	268+27=295
	Délais maximums			
Ag1	286	286+27=313	286+26=313	286+26=313
Ag2	285	285+28=313	285+27=312	285+27=312
Ag1 et Ag2	303	303+28=331	303+27=330	303+27=330
Ag1 à Ag3	321	321+28=349	321+28=349	321+27=348
Ag1 à Ag4	337	337+28=365	337+28=365	337+28=365
	Délais minimums			
Ag1	250	250+26=276	250+26=276	250+26=276
Ag2	250	250+25=275	250+26=276	250+26=276
Ag1 et Ag2	233	233+24=257	233+25=258	233+25=258
Ag1 à Ag3	216	216+23=239	216+23=239	216+24=240
Ag1 à Ag4	199	199+22=221	199+22=221	199+22=221

TAB. 4.4 – Détail des temps de propagation de la figure 4.23 évalués par Ctk. Le premier terme des sommes correspond au temps de traversée de la porte et le second au délai d’Elmore

puisque la répartition de la capacité de charge de la porte victime est :

- Capacité due à la porte de charge : $\approx 10fF$
- Capacité de routage par rapport à la masse : $\approx 11fF$
- Capacité de couplage : $\approx 18,3fF$

Temps de propagation sans agresseur actif

Le modèle de charge apparente que nous avons utilisé et le modèle de capacité d’entrée de porte utilisé par HiTas nous permet d’obtenir une charge apparente sur le nœud v qui vaut :

$$R = 73\Omega \quad (4.75)$$

$$C1 = 10,5fF \quad (4.76)$$

$$C2 = 30.7fF \quad (4.77)$$

La capacité équivalente C_{eq} utilisée pour le modèle MCC passe de $41,2fF$ à $39fF$, soit une diminution d’environ 5% de la capacité équivalente de charge. Dans cet exemple, sur une ligne aussi courte, l’effet de masquage est donc peu significatif.

Configuration	Délai	Eldo	Ctk	Erreur
Agresseur 1 actif	Tp max	46.3	44.0	-5.0%
	Tp min	24.8	29.0	+17.0%
Agresseur 2 actif	Tp max	46.3	45.0	-2.8%
	Tp min	24.7	28.0	+13.4%
Agresseurs 1 et 2 actifs	Tp max	58.6	53.0	-9.5%
	Tp min	16.2	20.0	+23.5%

TAB. 4.5 – Comparaison des temps de propagation extrêmes entre Vic et V du circuit de la figure 4.24, évalués par Ctk et par Eldo, en fonction des agressions.

La simulation sous Eldo du schéma représenté sur la figure 4.24 nous permet d'obtenir un temps de propagation de $35,8ps$ pour la porte, contre $37ps$ pour Ctk, soit un écart d'un peu plus de 3%.

Temps de propagation avec les agresseurs actifs

Nous relevons les temps de propagation, avec un ou deux agresseurs actifs, dans le tableau 4.5. Dans cette simulation, la linéarisation de la charge apparente au deuxième ordre n'a pas fonctionné. Le modèle du premier ordre a donc été utilisé. Nous savons que ce modèle conduit à une légère surestimation des temps de propagation. Nous en payons donc le prix dans ces résultats sur l'évaluation des temps de propagation minimum. La correction apportée au temps de propagation de la porte due au bruit est de $-3.3ps$ pour le temps de propagation minimum et de $+3.0ps$ pour le temps de propagation maximum. Le signal victime étant plus lent que ces agresseurs, l'effet Miller vaut dans ce cas 2 pour le calcul du délai maximum et 0 pour le délai minimum.

Les résultats montrent sur cet exemple que le modèle que nous avons utilisé ne donne pas de très bons résultats par rapport à Eldo. Néanmoins, lorsqu'on rapporte l'erreur commise par rapport au délai nominal (de $36ps$ sans agressions) l'erreur commise est proportionnellement beaucoup moins importante. De plus, les temps de propagation extrêmes dépendent d'une configuration des agresseurs bien particulière qui arrive rarement.

4.7 Conclusion

Le modèle que nous avons présenté dans ce chapitre permet de déterminer avec un temps de calcul très faible les temps de propagation extrêmes d'une porte et des réseaux RC à partir de la liste des agresseurs actifs. Les paramètres permettant ce calcul sont facilement déterminés, et ce calcul n'a lieu qu'une seule fois. Dans la plupart des cas, ce modèle donne de bons résultats en terme de précision. Nous allons maintenant pouvoir étudier la méthodologie utilisée dans l'analyse temporelle, et l'utilisation concrète de ces modèles, dans le chapitre suivant.

Chapitre 5

Analyse temporelle avec RC et diaphonie

Nous avons présenté dans le chapitre précédent comment évaluer les délais élémentaires en tenant compte des agresseurs actifs. Nous allons maintenant présenter dans ce chapitre la solution retenue pour déterminer le contexte d'agression (définition 4) de chaque signal. Dans le paragraphe 5.1, nous allons aborder la nécessité d'une approche pire cas, et l'information apportée par l'analyse de stabilité pour résoudre ce problème. Dans le paragraphe 5.2, nous présenterons l'information élémentaire de l'analyse de stabilité, qui est l'intervalle d'instabilité. Dans le paragraphe 5.3, nous observerons la modification qu'introduit la prise en compte d'une agression sur les intervalles d'instabilité, et nous introduirons la notion d'agression observable, et dans le paragraphe 5.4, nous étudierons les réponses apportées par les analyses de stabilité mono-intervalle et multi-intervalles. A partir de ces éléments, nous présenterons dans le chapitre 5.5 la boucle de relaxation, combinant analyse temporelle et analyse de stabilité, afin d'obtenir les délais élémentaires en ne prenant en compte que les agressions observables. Cette approche comporte quelques limitations qui seront étudiées dans le paragraphe 5.6. Enfin, nous présenterons les conclusions de ce chapitre dans le paragraphe 5.8.

5.1 La nécessité d'une approche pire cas

5.1.1 Analyseurs temporels statiques et simulateurs électriques

Jusqu'à ces dernières années, dans les circuits intégrés CMOS, les temps de propagation des portes et des interconnexions ne dépendaient que de quelques éléments constants, tels que les dimensions des transistors, les résistances des fils et des capacités. Les seules dimensions dynamiques qui intervenaient dans ces évaluations étaient la vitesse (la pente) des fronts des

signaux, l'état des autres entrées de la porte, et les transitions partielles (*glitches*). L'influence de ce dernier phénomène étant assez faible, il était possible de le négliger et de calculer sur chaque porte les délais des transitions "normales". L'intérêt de cette approche était de pouvoir traiter chaque porte indépendamment des autres, en ayant déterminé une fois pour toutes un contexte "pire cas" pour chaque transition possible d'un signal.

L'affirmation selon laquelle les analyseurs temporels ont une précision comparable à celle des simulateurs électriques cache cependant un problème complexe : pour simuler un chemin dans un circuit, il est nécessaire de trouver la (ou les) configuration des entrées du circuit qui va permettre la propagation du signal. Lorsqu'une telle configuration n'existe pas, on parle de "faux chemin statique". Lorsque plusieurs configurations existent, elles donnent généralement des temps de propagation très proches.

L'avantage décisif de l'analyseur temporel statique par rapport au simulateur électrique est qu'il ne nécessite pas de vecteurs de test. Dans la pratique, les simulateurs temporels sont utiles pour caractériser précisément les circuits contenant un petit nombre de chemins. Cette notion de "petit nombre de chemins" étant bien sûr à réévaluer d'année en année, avec l'évolution du matériel et du logiciel. Dès que le nombre de chemins augmente, les simulateurs électriques ne sont plus utilisables, seuls les analyseurs temporels statiques le sont.

5.1.2 L'introduction des couplages capacitifs dans l'analyse temporelle.

Dans le cas où l'on cherche à prendre en compte les effets de couplage capacitif, la situation se complique très sensiblement. En effet, une nouvelle dimension dynamique est introduite dans le calcul de délais : il s'agit de l'état des autres signaux dans le voisinage (au sens physique) du chemin étudié. Le nombre de ces signaux peut être extrêmement important (plusieurs milliers sur la totalité d'un chemin), et l'activité sur chacun de ces signaux peut être très variée. Le simulateur qui était la référence devient alors inutilisable pour déterminer les pires conditions de propagation, car le nombre de paramètres à faire varier devient très rapidement ingérable. Nous en avons fait l'expérience dans le chapitre traitant des modèles électriques. Même si on fait l'effort de tester le circuit par simulation électrique (ce qui est au demeurant impossible dès que le circuit dépasse quelques milliers de transistors) avec de nombreux scénarios pour s'assurer que les contraintes temporelles seront respectées dans un grand nombre de cas, il n'y a aucune certitude que ce circuit une fois fondu et mis en place chez le client fonctionne correctement. Il suffit qu'un jour, une seule fois, un contexte électrique non simulé se produise pour qu'un chemin ait un temps de propagation supérieur au temps de cycle du circuit, avec comme conséquence qu'un registre échantillonne une valeur fautive. De plus, cette

erreur sera certainement non reproductible.

Lorsqu'un concepteur est face à un circuit présentant des couplages importants, la solution de diminuer la vitesse de fonctionnement du circuit ne résout que le problème des chaînes longues, pas celui des chaînes courtes introduites par les effets capacitifs. Il faut donc pouvoir être en mesure d'évaluer de manière fiable ce phénomène. On dispose de la valeur nominale des chemins en ne prenant pas en compte les effets de couplage, et on souhaite pouvoir disposer d'un encadrement pire cas de cette valeur en fonction du contexte diaphonant. Nous avons présenté dans le chapitre précédent des modèles d'évaluation des délais pire cas que nous avons mis en place. Une des difficultés de cette analyse est que, considérer systématiquement tous les agresseurs comme étant actifs conduit à des surestimations des délais, non représentatives de ce qui se passera en réalité. Le problème est donc choisir parmi les agresseurs d'un signal lesquels peuvent réellement influencer les temps de propagation dans le circuit. Une fois que nous saurons quels sont les signaux qui ont un comportement agressif effectif, nous pourrons calculer leur influence.

5.1.3 Approche basée sur l'analyse de stabilité.

Nous avons vu qu'un agresseur a un effet temporel dès lors qu'il commute en même temps que le signal victime. Pour un signal, il faudrait donc connaître les instants de commutation possibles, ainsi que ceux de ses agresseurs. Bien évidemment, ces instants sont beaucoup trop nombreux sur chaque signal pour les considérer, et l'ensemble des combinaisons de ces instants atteint un nombre astronomique. Une autre approche consiste à raisonner par rapport aux intervalles d'instabilité, et à considérer que si un intervalle d'instabilité d'un agresseur coïncide avec celui de sa victime, alors il faut envisager dans ce cas les pires conditions d'agression entre ces deux signaux. C'est la raison pour laquelle nous nous sommes intéressés essentiellement aux agressions pire cas dans le chapitre traitant des modèles électriques.

5.1.4 Approche itérative.

L'approche que nous avons retenue consiste à procéder par itérations. Nous avons vu dans le chapitre traitant de l'état de l'art que cette approche est utilisée dans tous les logiciels d'analyse temporelle qui prennent en compte les effets de couplage. Dans la phase d'initialisation, les délais sont calculés sans prendre en compte les effets de couplage. Nous obtenons donc les délais nominaux. Puis, la boucle d'itération commence par une analyse de stabilité pour calculer les intervalles d'instabilité des signaux. Pour chaque signal, il existe un ou plusieurs agresseurs potentiels. Les délais sont donc recalculés en prenant en compte ces couplages. Bien

sûr, cela va modifier à nouveau les intervalles d'instabilité, il faudra donc à nouveau réévaluer les agressions potentielles, puis à nouveau recalculer les délais et ainsi de suite. Tout ceci est présenté en détail dans la suite de ce chapitre (paragraphe 5.5).

5.2 Les intervalles d'instabilité

5.2.1 Définition

Les temps de propagation élémentaires dans le circuit sont une information basique. Pris individuellement, ils n'apportent aux concepteurs du circuit qu'une vision extrêmement locale du comportement temporel du circuit. Au niveau global, le circuit doit répondre à des contraintes sur les signaux de stabilité, qui sont les connecteurs et les points mémorisants. L'information utile porte donc sur les instants de transition possible (au moins le premier et le dernier) des signaux de stabilité. Dans n'importe quel circuit, même pour les simples, les instants de transitions possibles sur un signal sont extrêmement nombreux, beaucoup trop pour pouvoir être représentés dans les ordinateurs. Nous avons reporté dans le tableau 5.1 le nombre de chemins possibles arrivant sur les signaux de stabilité d'un seul des blocs d'un circuit conçu par Bull. Chaque chemin a un temps de propagation qui lui est propre. La colonne "Cône" est l'index (l'identifiant) du cône. La colonne "Chemins" est le nombre de chemins qui arrivent sur ce cône, et la dernière colonne "Profondeur" est le nombre de signaux maximum traversés dans le graphe de causalité. Supposons que le signal à l'entrée du cône 33 ait une capacité de couplage avec l'entrée du cône 32. Sur ce seul signal, il y a donc 96446×93813 possibilités de couplage différentes, soit plus de neuf milliards. Bien sûr, il ne s'agit que d'une borne infinitésimale par rapport à l'ensemble de ces combinaisons, car nous avons supposé qu'il n'y a qu'une seule transition possible sur ces signaux au cours d'un cycle d'horloge. De plus, il ne s'agit que du dernier signal d'un chemin, en réalité, il faut faire ce calcul pour tous les signaux composant un chemin.

La solution est donc de ne pas considérer les transitions possibles une à une, mais de considérer des durées qui regroupent potentiellement plusieurs transitions. Ces durées sont appelées intervalles d'instabilité.

Définition 5

*La stabilité d'un sommet du graphe temporel est définie sur une période d'horloge par un ensemble de n intervalles temporels disjoints. Ces intervalles sont définis par deux variables, u et d , avec par définition $d < u$. Le signal est **absolument** stable à l'extérieur de ces intervalles, et **potentiellement** instable à l'intérieur de ces intervalles.*

Nous utiliserons dans la suite de ce manuscrit la représentation de la figure 5.1 pour représen-

Cone	Chemins	Profondeur
33	96446	258
32	93813	254
31	109140	266
30	106023	262
29	102668	258
28	99075	254
27	113409	266
26	104852	258
25	100167	254
24	109266	262
23	84915	254
22	59423	262
21	58956	258
20	58387	254
19	75880	266
18	75031	262
17	74046	258
16	72925	254
15	89832	266
14	88363	262
13	86724	258
12	59788	266
11	41845	254
10	101100	266
9	19283	262
8	23360	266
7	23555	262
6	23716	258
5	23843	254
4	42100	266
3	42083	262
2	41998	258
1	98875	262

TAB. 5.1 – Nombre de chemins arrivant sur les sommets de stabilité

ter visuellement la stabilité des signaux. Les abscisses représentent le temps, et les ordonnées

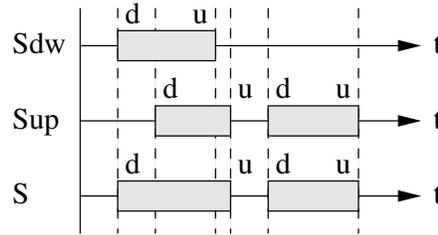


FIG. 5.1 – Stabilité des signaux

la stabilité sur les signaux au cours du temps. Les intervalles d'instabilité sont représentés par des rectangles gris. La stabilité peut être calculée pour chaque signal ou pour chaque sommet du graphe de causalité. Dans le second cas, la stabilité est calculée pour chaque type de transition (sommets Sup et Sdw sur la figure 5.1). Alors que pour le premier cas, cette stabilité est calculée pour n'importe quelle transition, montante ou descendante. Elle correspond alors à l'union des intervalles d'instabilité des deux sommets qui correspondent au signal (signal S sur la figure 5.1). La variable d signifie que la stabilité du sommet est "down", tandis que u signifie que la stabilité est "up". C'est une convention de nommage habituelle dans notre laboratoire.

Ces paramètres de stabilité sont calculés pour des transitions, c'est à dire lorsque le signal passe par $v_{dd}/2$. Cette définition nous posera quelques problèmes par la suite, car le signal, bien que logiquement stable à 0 ou à 1 en dehors de l'intervalle $(u; d)$, a encore la moitié de la dynamique à parcourir. Nous avons représenté ce cas sur l'exemple de la figure 5.2. En

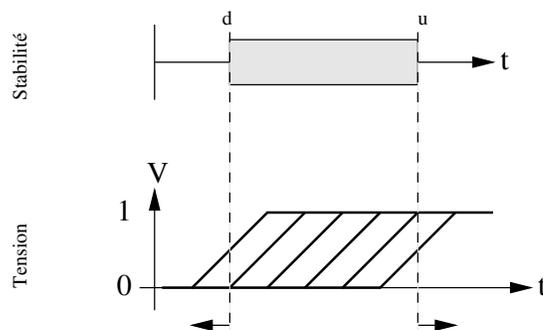


FIG. 5.2 – Equivalence entre l'instabilité logique d'un signal et ses formes analogiques possibles.

considérant que les fronts sont des segments de droite, on peut définir une stabilité électrique qui correspond à :

- Premier instant d'instabilité : d auquel on retire la valeur de la moitié de la durée du front

- maximum ;
- Dernier instant d'instabilité : u auquel on ajoute la valeur de la moitié de la durée du front maximum.

5.2.2 Détermination de la stabilité d'un sommet du graphe temporel.

Cette information se calcule avec un outil appelé analyseur de stabilité. L'instabilité d'un sommet se calcule à partir de l'instabilité des sommets précédents (voir l'exemple sur la figure 5.3) et des délais sur les arcs, avec les relations :

$$D_j = D_i + T_{p_{min}} \quad (5.1)$$

$$U_j = U_i + T_{p_{max}} \quad (5.2)$$

Ces équations fondamentales se déterminent par le raisonnement suivant : nous connaissons

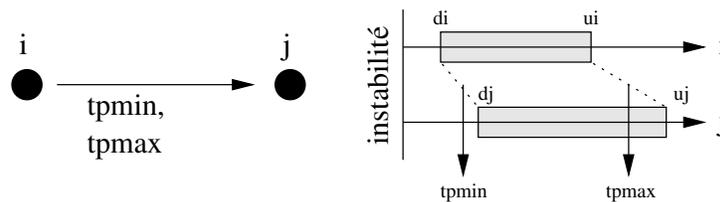


FIG. 5.3 – Propagation entre 2 sommets.

le premier instant où le sommet i peut avoir une transition. C'est l'instant D_i . Sur le sommet suivant j , le premier instant où une transition est possible correspond au cas où la transition sur le sommet i s'est propagée la plus rapidement possible, donc avec le temps $T_{p_{min}}$. Avec le même raisonnement, nous connaissons le dernier instant où le sommet i peut avoir une transition U_i , la dernière transition possible sur le sommet j se produit avec un décalage correspondant au temps de propagation maximum $T_{p_{max}}$.

L'apparition de plusieurs intervalles d'instabilité vient des convergences de signaux qui traversent des blocs ayant des temps de propagation très différents. Cela est le cas sur l'exemple de la figure 5.4 où nous avons supposé que le temps de propagation maximum du bloc 1 était très inférieur au temps de propagation minimum du bloc 2.

Les analyseurs de stabilité ne font pas tous de l'analyse multi-intervalle. Dans le cas général, un sommet possède plusieurs sommets en entrée. Les intervalles d'instabilité résultant sur ce sommet correspondent alors à l'union des intervalles d'instabilité calculés individuellement pour chaque entrée. Il peut alors se produire des chevauchements des intervalles d'instabilité.

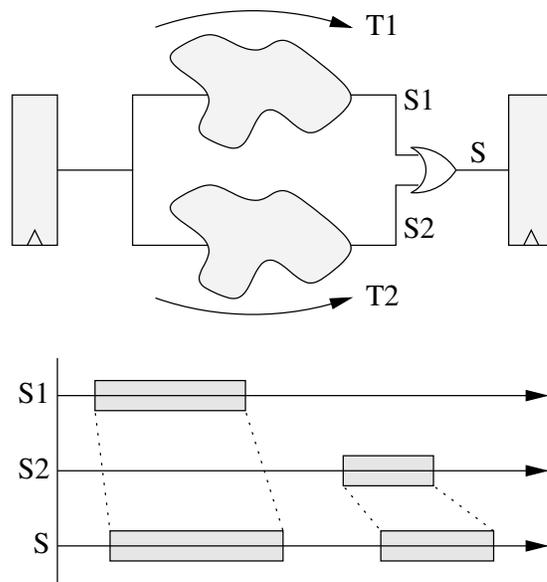


FIG. 5.4 – Origine de plusieurs intervalles d’instabilité

Nous avons représenté sur la figure 5.5 un exemple de recouvrement des intervalles d’instabilité : deux intervalles d’instabilité sur un sommet se regroupent en un seul sur le sommet suivant à cause de l’écart entre les temps de propagation minimum et maximum (zone 1 sur l’intervalle d’instabilité de C), et d’autre part des intervalles d’instabilité provenant de deux sommets différents se regroupant en un seul sur le sommet suivant(zone 2).

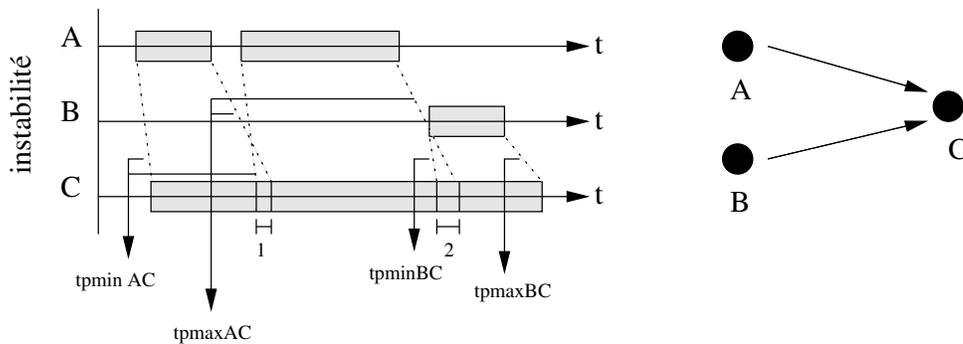


FIG. 5.5 – Exemple de fusion des intervalles d’instabilité.

Le calcul des intervalles d’instabilité sur le graphe temporel est une propagation dans le sens causal du graphe, avec un "parcours en largeur d’abord" [GM79] entre les sommets de stabilité. Cet algorithme est assez simple dans le cas de registre, et l’est moins dans le cas de

latches car les intervalles d'instabilité "traversent" les latches sur leur état transparent. Sans entrer dans les détails, plusieurs itérations (passage à travers les latches) sont nécessaires pour réaliser cette analyse jusqu'à obtenir la convergence. Cette convergence dépend des conditions initiales. D'autres problèmes se posent également, lorsqu'on a plusieurs phases ou plusieurs domaines d'horloge. Une thèse a été soutenue sur ce sujet dans notre laboratoire il y a une dizaine d'années par Messaoud Chiheb dans [Chi92]. Quel que soit l'algorithme utilisé, seules comptent pour nous les stabilités relatives des signaux, indépendamment de leur phase d'horloge.

5.3 Influence des agressions sur la stabilité.

5.3.1 Influence des agressions.

Nous savons que l'effet des couplages capacitifs a deux effets sur un arc :

1. Une diminution du temps de propagation minimum ;
2. Une augmentation du temps de propagation maximum.

Lorsqu'on modifie les caractéristiques d'un arc, cela se traduit par une **augmentation** de la **durée des intervalles d'instabilité** des sommets. Cette modification peut se répercuter sur

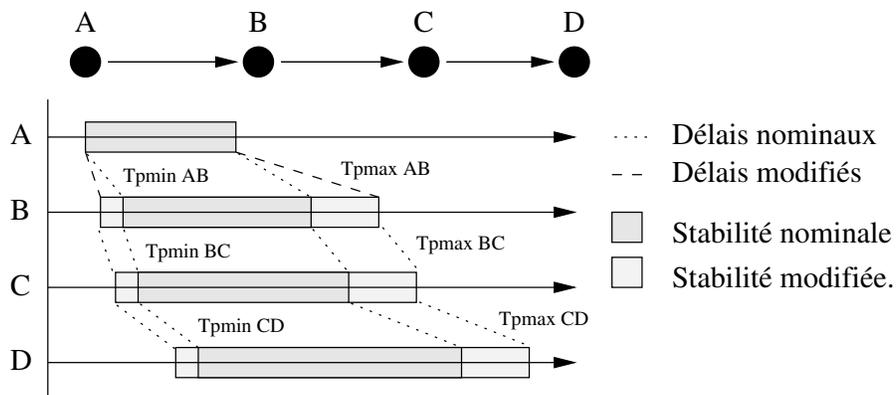


FIG. 5.6 – Influence d'un changement de temps de propagation entre A et B sur la stabilité d'une série de sommets.

tous les sommets en aval du graphe. Sur la figure 5.6, nous avons représenté la stabilité de 4 sommets consécutifs dans un graphe temporel, et représenté la stabilité résultante d'un changement de temps de propagation sur le premier arc. Bien évidemment, l'augmentation de la largeur des intervalles d'instabilité peut conduire à de nouveaux recouvrements d'intervalles. Si cette agression se propage jusqu'aux sommets de stabilité, elle entraîne une modification des

caractéristiques temporelles du circuit.

5.3.2 Les agressions observables.

Notre approche pour déterminer l'influence temporelle des effets de couplage capacitif se base sur l'analyse de stabilité. Le calcul des intervalles d'instabilité est construit à partir du graphe de causalité, en utilisant les deux seules valeurs dont nous disposons sur les arcs : le temps de propagation minimum et maximum. Nous allons voir que dans certains cas les agressions entre signaux ne sont pas suffisantes pour modifier la stabilité des signaux. La stabilité étant une conséquence des temps de propagation élémentaires, il ne faudra donc pas systématiquement modifier les temps de propagation sur les arcs, même si une agression a physiquement lieu. Nous introduisons donc la notion d'agression observable :

Définition 6

Une agression observable est une agression qui modifie réellement la stabilité du sommet sur lequel s'est produit l'agression.

Si on ne prend pas en compte cette notion d'agression observable, et qu'à chaque agression on modifie systématiquement les délais sur les arcs, on prend un risque important de surestimer l'influence des capacités de couplage au niveau temporel.

5.3.3 Modification conditionnelle des arcs.

Voyons maintenant quand prendre en compte la modification des délais sur les arcs lorsqu'une agression a lieu. Sur la figure 5.7, nous présentons un signal victime V agressé par le signal A . Le signal électrique A correspond à la sortie de la première porte, $A1$, à l'entrée de la seconde porte, $A2$, et au réseau RC décrivant l'interconnexion, représenté par R . Dans le graphe temporel, le signal électrique A est donc représenté par deux signaux temporels, $A1$ et $A2$. Nous avons la même description pour le signal V . L'agression de V par A va modifier les temps de propagation entre V_{in} et V_{out} , cette modification des délais ne se produisant que lorsqu'il y a une possibilité de commutation simultanée des deux signaux V et A . En dehors de cette possibilité, ce sont les temps nominaux qu'il faut prendre en compte. Sur notre exemple, même en prenant en compte les délais calculés avec les effets de couplage (traits en tiret) autour des intervalles d'instabilités communs entre A et V , l'intervalle d'instabilité sur V (en sombre) résultant reste toujours contenu dans celui d'origine de V . Si dans cet exemple nous avons modifié effectivement la valeur des temps de propagation des arcs entre V_{in} et V_{out} , nous aurions pris en compte une influence temporelle qui n'influence pas la stabilité, et donc trop pessimiste.

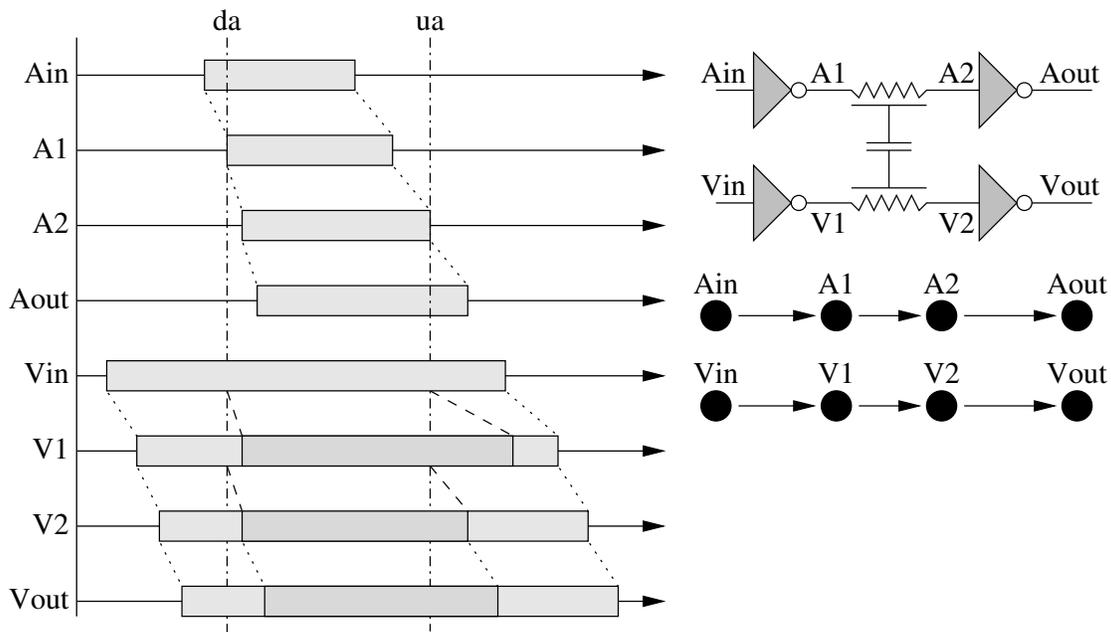


FIG. 5.7 – Agression non observable.

Cela revient à dire que l'agression de V par A n'est pas observable. Nous énonçons donc deux propriétés :

Propriété 1

Considérons la figure 5.7. Les fonctions $u(X)$ et $d(X)$ représentent les variables de stabilité d'un nœud X , et $Tpm_{X \rightarrow Y}(Z)$ et $TpM_{X \rightarrow Y}(Z)$ les temps de propagation minimum et maximum entre les nœuds X et Y en prenant en compte l'agresseur Z .

L'agresseur A est considéré comme un agresseur **observable** "meilleurs cas" pour V si l'une des deux relations suivantes est vérifiée :

$$\begin{cases} d(A_1) + Tpm_{V_{in} \rightarrow V_1}(A) < d(V_1) \\ d(A_1) + Tpm_{V_1 \rightarrow V_2}(A) < d(V_2) \end{cases} \quad (5.3)$$

L'agresseur A est considéré comme un agresseur **observable** "pire cas" pour V si l'une des deux relation suivante est vérifiée :

$$\begin{cases} u(A_2) + TpM_{V_{in} \rightarrow V_1}(A) > u(V_1) \\ u(A_2) + TpM_{V_1 \rightarrow V_2}(A) > u(V_2) \end{cases} \quad (5.4)$$

Cette notion d'observabilité n'est pas symétrique, comme cela est le cas sur notre exemple. L'intervalle d'instabilité de A est complètement inclus dans celui de V , il doit donc être modifié *via* les délais des arcs passant par le sommet A .

Propriété 2

L'observabilité d'une agression n'est pas symétrique entre deux signaux : Si S_1 est un agresseur observable sur S_2 , cela n'implique pas que S_2 soit un agresseur observable sur S_1 .

Reprenons l'exemple précédent avec les intervalles de la figure 5.8. Dans ce cas, nous

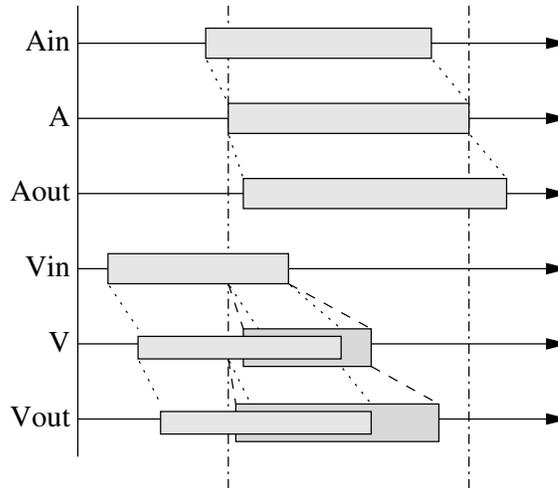


FIG. 5.8 – Modification partielle des intervalles d'instabilité.

constatons que le début de l'intervalle d'instabilité ne change pas, et que seule la fin change. Cela revient à dire que seul le Tp_{max} de l'arc doit être recalculé. Il est bien sûr possible d'avoir la situation où seul Tp_{min} doit être changé. Dans les agressions correspondant à un "pire cas", lorsque les intervalles d'instabilité de deux sommets commutant en sens opposés se recouvrent, l'intervalle d'instabilité s'agrandit éventuellement par la droite (augmentation de u), correspondant à une modification des délais maximum. Réciproquement, dans les agressions correspondant à un "meilleur cas", lorsque les intervalles d'instabilité de deux sommets commutant dans le même sens se recouvrent, l'intervalle d'instabilité s'agrandit par la gauche (diminution de d), correspondant à une modification des délais minimum. D'où la propriété :

Propriété 3

L'observabilité d'une agression peut être partielle, c'est à dire ne modifier qu'un seul des temps de propagation minimum ou maximum.

5.3.4 Influence d'une agression observable

Considérons les intervalles d'instabilité de trois sommets représentés sur la figure 5.9, et supposons qu'en amont, un effet de couplage ait élargi l'intervalle d'instabilité du signal A, suivant les mécanismes présentés précédemment. Nous voyons sur notre exemple que, face à la largeur de l'intervalle d'instabilité du sommet B et des conditions de propagation sur les arcs

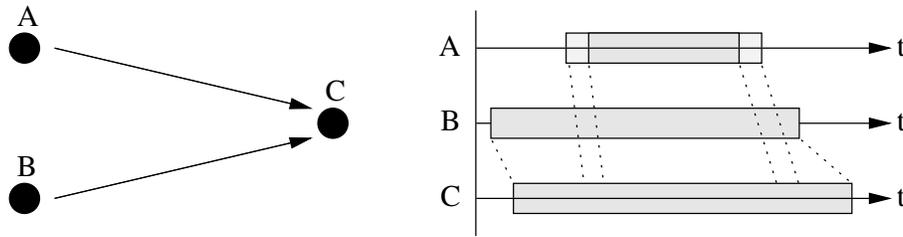


FIG. 5.9 – Condition de propagation d’une agression observable.

AC et BC, l’intervalle d’instabilité de C n’est pas modifié. La stabilité de tous les sommets en aval de C ne sera pas donc modifiée non plus. Cet exemple montre que dans de nombreux cas, il est possible d’avoir des agressions qui modifient des temps de propagation sur certains arcs du graphe de causalité, mais que ces modifications n’ont pas de répercussion temporelle sur le circuit. Ce phénomène est implicitement pris en compte, lors de l’analyse de stabilité. Nous ne risquons pas de cette façon de prendre en compte des agressions qui n’ont physiquement aucune influence temporelle sur les sommets de stabilité.

5.3.5 Méthode retenue pour la détection des agressions observables

La solution qui vient immédiatement à l’esprit pour choisir quelles sont les agressions observables est de calculer les délais minimum et maximum afin de constater leurs effets potentiels sur la stabilité des signaux. Cela risque d’être excessivement coûteux en terme de puissance de calcul requise, car en règle générale il n’y a pas qu’un agresseur, mais plusieurs, et il y a au cours d’un intervalle d’instabilité d’un signal victime plusieurs configurations d’agresseurs, comme cela est représenté sur la figure 5.10. Ici, ces deux agresseurs définissent trois

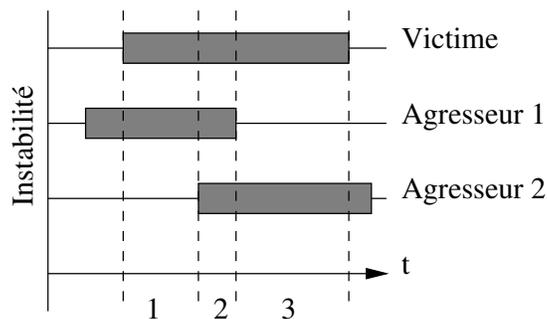


FIG. 5.10 – Plusieurs agresseurs correspondant à plusieurs contexte pour la victime.

contextes d’agression : Agresseur 1 seul, Agresseurs 1 et 2 en même temps, et Agresseur 2 seul. Nous préférons plutôt considérer que si la première ou la dernière transition possible du signal

victime n'est pas influencée par un agresseur, alors il n'y a pas d'agression observable. Il reste à préciser ce que nous voulons entendre avec le terme "influence". Nous considérons qu'une transition sur un signal victime (un front) est influencée par une transition sur un agresseur dès que ces transitions se recouvrent temporellement. Considérons les définitions données sur

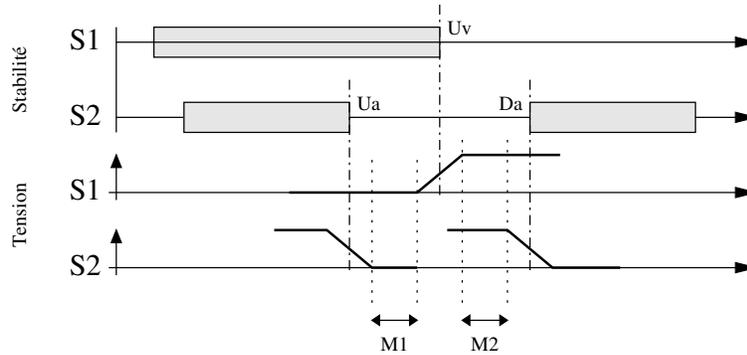


FIG. 5.11 – Stabilité de deux signaux, et la dernière transition possible sur ces signaux.

la figure 5.11. Nous avons représenté la stabilité de deux sommets S1 et S2 dans un contexte d'agression "pire cas" en considérant que S1 est la victime. L'agresseur S2 a ici deux intervalles d'instabilité, de part et d'autre de la fin de l'instabilité Uv de S1. Nous avons également tracé en dessous de ces intervalles la dernière transition possible de S1, et la dernière et première transition de S2. Nous appelons F_v le front le plus long sur S1, et F_a le front le plus long sur S2. Le cas représenté ici est celui où la dernière transition montante de S1 n'est pas influencée par l'agresseur S2, ni avant, ni après. Il faut donc considérer l'agresseur S2 comme étant un agresseur non observable pour Tp_{max} . Ce raisonnement est applicable au début de l'intervalle d'instabilité D de S1, pour déterminer si S2 est, ou n'est pas, un agresseur observable pour Tp_{min} .

Avec cette approche, il devient aisé de définir des conditions simples pour déterminer si un agresseur est observable ou non. La condition pour qu'un agresseur soit observable pour U (modification de Tp_{max}) est donnée par la vérification de l'une des deux inégalités :

$$\begin{cases} Ua + \frac{F_a}{2} > Uv - \frac{F_v}{2} \\ Da - \frac{F_a}{2} < Uv + \frac{F_v}{2} \end{cases} \quad (5.5)$$

De même, la condition pour qu'un agresseur soit observable pour D (modification de Tp_{min}) est donnée par :

$$\begin{cases} Ua + \frac{F_a}{2} > Dv - \frac{F_v}{2} \\ Da - \frac{F_a}{2} < Dv + \frac{F_v}{2} \end{cases} \quad (5.6)$$

Généralisons ces équations au cas où nous aurions plusieurs configurations d'agresseurs. Il faut déterminer quels sont les agresseurs qui sont observables et ceux qui ne le sont pas. Sur la figure 5.10, nous avons le cas où deux agresseurs déterminent trois contextes différents. Pour déterminer l'observabilité des agresseurs sur la borne u , nous parcourons chaque contexte d'agression (sur le dessin, les contextes de 1 à 3), de gauche à droite. Dans chacun des contextes, nous déterminons le front F_a le plus long parmi les agresseurs de ce contexte, et nous effectuons le test d'observabilité précédent. Si le test est positif, nous supposons que tous les agresseurs de ce contexte sont des agresseurs observables, ainsi que tous les agresseurs des contextes jusqu'à la borne u de l'agresseur. Nous procédons de la même façon pour la borne d de la victime, en parcourant les contextes d'agression de droite à gauche.

5.4 Approche mono intervalle et multi intervalle.

Jusqu'ici, nous avons présenté les agressions observables qui modifient la stabilité du circuit, et les critères qui permettent de les sélectionner. Nous avons présenté plusieurs cas où, au cours d'une même période d'horloge, un signal a plusieurs intervalles d'instabilité possibles. Nous allons présenter un comparatif des deux analyses de stabilité, sachant que c'est du ressort de l'utilisateur de choisir laquelle est la plus adaptée à ses besoins.

L'approche multi-intervalle permet une étude plus fine du comportement du circuit, en particulier lorsqu'on s'intéresse au calcul des bruits de commutation. Malheureusement, nous sommes toujours contraints au fait que, quel que soit le nombre d'intervalles d'instabilité, il n'y a toujours que deux valeurs de temps de propagation sur les arcs du graphe de causalité. Or, comme cela est représenté sur la figure 5.12, il suffit qu'un agresseur A1 soit observable sur un seul des intervalles de V pour changer la taille de tous les intervalles de V, y compris le dernier instant d'instabilité, alors que c'est souvent l'information la plus pertinente. Dans l'approche mono-intervalle, ce problème ne se pose pas car l'agresseur A1 n'est plus un agresseur observable pour V. Pour le signal V, il paraît donc judicieux de retenir l'approche mono-intervalle. Toujours sur le même exemple, considérons le cas du signal A2. Du point de vue du signal V, il n'est pas agresseur lorsqu'on considère l'approche multi-intervalle, et c'est un agresseur non observable dans le cas de l'approche mono-intervalle. Dans ce cas, il n'y a pas d'avantages à prendre l'une ou l'autre des deux approches. Par contre, lorsqu'on considère le signal A2 comme une victime agressée par V, dans l'approche multi-intervalle, on constate que V ne peut pas agresser A2. Dans l'approche mono-intervalle, V est un agresseur observable pour A2.

Non venons donc sur cet exemple de voir trois cas différents où les approches mono-

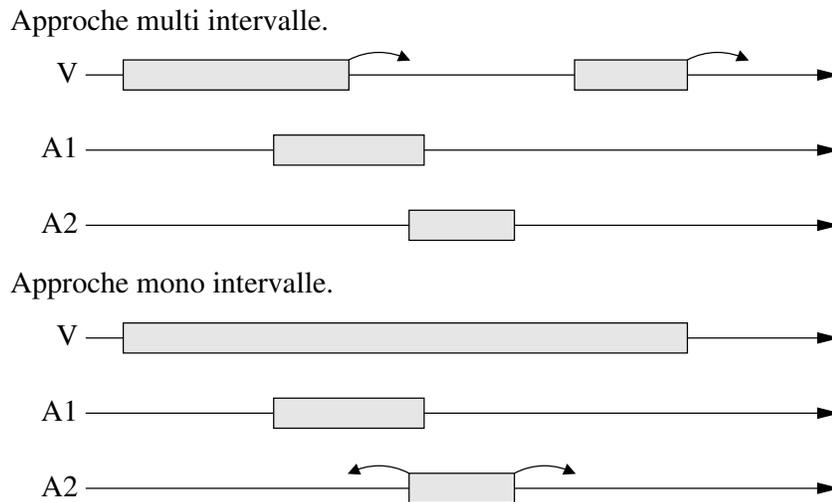


FIG. 5.12 – Effet d’une agression dans un contexte multi intervalle et mono intervalle.

intervalle et multi-intervalle ont à la fois des avantages et des inconvénients. Il est difficile de concilier les avantages des deux en faisant autrement que de mémoriser plusieurs valeurs de temps de propagation sur les arcs du graphe de causalité. Nous n’avons pas retenu cette solution car elle remet beaucoup trop en cause le logiciel existant, et sa mise au point risque d’être très délicate. Nous avons donc choisi de continuer sur l’approche multi-intervalle, sachant que l’approche mono-intervalle est un cas particulier de cette première. Nous verrons dans le chapitre présentant le prototype que le concepteur a le choix au niveau de l’analyseur de stabilité entre les deux approches.

5.5 La boucle de relaxation.

Le principal problème méthodologique posé par la nécessité de prendre en compte les capacités diaphonantes est l’apparition de dépendances dont le sens n’est plus le sens causal. Une dépendance dans le sens causal, telle que les fronts, est facile à prendre en compte avec un parcours du graphe de causalité en "largeur d’abord". Dans le cas des couplages capacitifs, cette dépendance n’est plus causale, le système est complètement bouclé. Sur l’exemple de la figure 5.13, nous avons représenté une situation simplifiée, bien que courante. Les conditions de propagation sur le signal 2 dépendent de l’activité sur le signal 6. Pour calculer cette dernière, il faut connaître l’activité sur le signal 5. Or, les conditions de propagation de celle-ci dépendent du signal 3, qui nécessite de connaître l’activité sur le signal 2. Nous sommes là dans une situation d’inter blocage. Dans les situations autrement plus complexes que nous rencontrons dans les circuits intégrés contenant des millions de portes, et dont les interconnexions

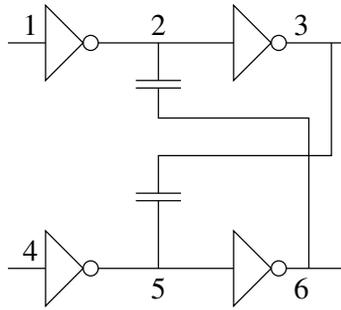


FIG. 5.13 – Exemple de couplages capacitifs bouclés.

comportent un grand nombre de capacités de couplage, il n’y a pas d’autre solution que de procéder par approximations successives, et de considérer chaque signal indépendamment. Nous profitons alors du découplage introduit dans ce but dans nos modèles électriques présentés dans les précédents chapitres. L’approche se basant sur les intervalles d’instabilité et considérant les agressions en pire cas apporte une solution satisfaisante. La solution retenue consiste à considérer que tous les agresseurs sont initialement inactifs, puis de prendre en compte progressivement les agressions actives qui peuvent se produire. Il s’agit donc d’une approche par relaxation. Nous avons présenté brièvement cette solution au début de ce chapitre dans le paragraphe 5.1.4. Nous développons ici plus en détail cet algorithme :

1. Nous commençons par réaliser une analyse temporelle classique pour déterminer les temps de propagation nominaux. Cette analyse suppose que toutes les agressions sont non observables. Dans cette phase, nous construisons le graphe de causalité détaillé : chaque sommet du graphe correspond à une transition d’un signal (sommet porte et sommet RC), et les arcs contiennent les délais entre sommets (Tp_{min} et Tp_{max}).
2. Pour détecter les agressions observables sur chaque type de transition, nous réalisons l’analyse de stabilité. Les agresseurs pouvant modifier de manière observable la borne u d’une victime sont ceux qui commutent dans le sens inverse de la victime et qui répondent au critère de l’équation 5.5. De même, les agresseurs pouvant modifier de manière observable la borne d d’une victime sont ceux qui commutent dans le même sens de la victime et qui répondent au critère de l’équation 5.6. A la fin de cette phase, chaque sommet connaît la liste de ses agresseurs observables.
3. Ensuite, nous calculons la charge équivalente en sortie des portes. Ce calcul nécessite plusieurs itérations, car nous avons vu que la charge apparente dépend de l’effet Miller, que l’effet Miller dépend des fronts, et finalement que les fronts dépendent de la charge

apparente :

- (a) Calcul de l'effet Miller pour chaque paire victime-agresseur en utilisant les relations résumées dans le tableau 4.2,
- (b) Calcul de la charge apparente en sortie de chaque porte, en utilisant les relations 4.66 et 4.66 page 107. Cette charge est convertie en une capacité équivalente avec la relation 4.35.
- (c) Calcul du front en sortie de chaque porte,
- (d) Calcul des fronts aux extrémités des réseaux RC.

Cette série de quatre opérations est à itérer jusqu'à ce que les fronts ne varient plus significativement.

4. A partir de la charge équivalente et des fronts d'entrée des portes, il est alors possible de calculer les temps de propagation des portes et des réseaux RC.

A la fin de cette série d'opérations, nous avons donc pour chaque signal les intervalles d'instabilité avec prise en compte des effets de couplage capacitif. Cependant, le changement d'un seul des arcs de la figure peut suffire à modifier de nombreux intervalles d'instabilité (voir la figure 5.6). Cette modification des intervalles d'instabilité peut suffire, dans notre approche pire cas, à transformer des agressions non observables en agressions observables (voir la figure 5.11 page 140). C'est la raison pour laquelle il faut recommencer un ou plusieurs cycles à partir de l'étape 2 jusqu'à ne plus détecter de nouvelles interactions. Cet algorithme est représenté sur la figure 5.14.

Cette façon de procéder va à l'opposé des logiciels d'analyse temporelle issus du monde industriel, puisque nous avons vu que ceux-ci remplacent les deux premières phases par :

1. Réalisation d'une analyse temporelle en pire cas, en prenant systématiquement en compte les agressions.
2. A partir d'une analyse de stabilité, il est procédé à l'élimination des agressions qui ne peuvent pas se produire simultanément.

Cette façon de procéder est plus prudente, mais très pessimiste : le système résolu dans ces conditions peut très bien se stabiliser sur un état qui peut ne pas représenter des agressions réelles.

Intéressons nous maintenant au calcul de la charge équivalente de la porte. Nous avons vu que celle ci ne dépendait que des fronts calculés aux étapes 3.c et 3.d. Cela est tout à fait normal car notre approche est une approche pire cas, c'est à dire qu'à partir du moment où un agresseur est considéré comme observable, son instant précis de commutation n'intervient pas. Le seul

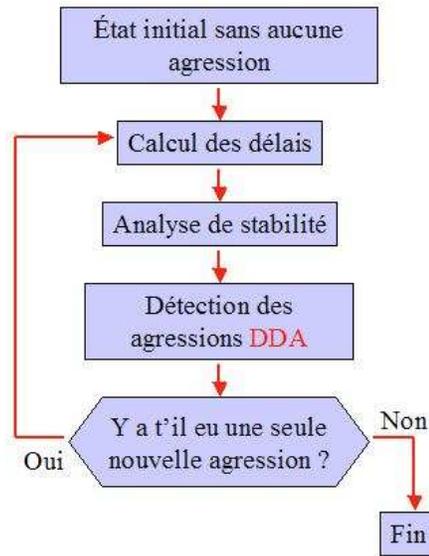


FIG. 5.14 – Algorithme complet.

paramètre qui le caractérise est alors son front. Compte tenu de la dépendance entre l'effet Miller, les fronts et la charge équivalente, plusieurs itérations des étapes 3.a à 3.d sont nécessaires pour obtenir une convergence, c'est à dire que les trois valeurs calculées (charge équivalente, front en sortie de la porte et front aux extrémités des réseaux RC) pour tous les signaux soient stables d'une itération à l'autre. Dans la pratique, nous avons rencontré plusieurs cas où l'état de ce système oscillait de façon périodique entre plusieurs valeurs. Cela peut être dû à la représentation des valeurs sous la forme d'entier, donc une représentation discrète d'éléments continus. En pratique, l'origine de ce phénomène n'est pas importante. Nous avons donc fixé un nombre maximum d'itérations pour réaliser ce calcul, et nous retenons pour ces 3 valeurs le pire cas des deux dernières itérations.

Dans le calcul de la charge équivalente, il reste un dernier problème que nous n'avons pas abordé, à savoir la dépendance du front de sortie d'une porte par rapport à son front d'entrée. Lorsque cette dépendance est prise en compte, la modification d'un seul front en amont du graphe de causalité se répercute jusqu'aux sommets de stabilité situés en aval. Il faut réévaluer les fronts en réalisant un parcours en "largeur d'abord" sur le graphe de causalité. Dans le cas où la dépendance des fronts entre l'entrée et la sortie des portes ne serait pas prise en compte,

l'ordre d'évaluation n'est pas important. Lorsque la propagation des fronts est prise en compte, elle ne diminue seulement que la vitesse de convergence du système, mais apporte un léger gain sur la précision des résultats obtenus.

5.6 Limites de cette approche.

5.6.1 Problème de convergence.

Nous ne modifions donc le graphe de causalité que si une agression est observable, c'est à dire si elle modifie la stabilité des sommets. Cela pose quelques problèmes de convergence, que nous illustrons sur un exemple. Sur la figure 5.15, nous avons représenté la stabilité de deux

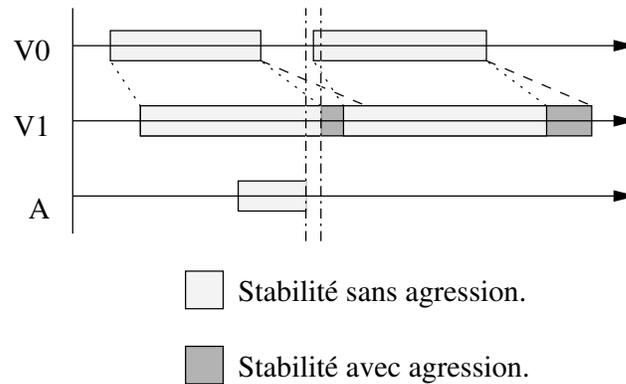


FIG. 5.15 – Problème de convergence lors de la fusion de deux intervalles d'instabilité.

sommets consécutifs V0 et V1 dans le graphe de causalité. V0 possède deux intervalles d'instabilité disjoints, qui sont transmis en deux intervalles disjoints sur V1 lorsqu'on calcule les temps de propagation sans agression observable. Lors de la première itération, l'agresseur A remplit les critères pour être un agresseur observable pire cas sur V1. Le temps de propagation maximum entre V0 et V1 va donc être augmenté, et les deux intervalles initialement disjoints sur V1 vont fusionner suivant les règles de construction des intervalles d'instabilité présentées dans le chapitre 5.2.2. A l'itération suivante, on va détecter que, compte tenu de l'unique intervalle d'instabilité de V1, A n'est plus un agresseur observable pour V1. La tentation est alors grande de modifier à nouveau le temps de propagation maximum entre V0 et V1 pour lui rendre sa valeur initiale. Et de recommencer ainsi indéfiniment.

Ce problème est particulier à l'approche multi-intervalle. Le problème plus général est qu'un agresseur peut être initialement observable, et qu'une fois la stabilité recalculée avec

les nouveaux délais, l'agresseur peut ne plus être observable. Pour contourner ce problème, nous énonçons la propriété suivante :

Propriété 4

La décision de considérer un agresseur observable est irrévocable.

Cette propriété permet en plus de pouvoir sélectionner un contexte d'agression systématiquement pire cas pour les nombreux signaux qui sont faiblement agressés, réalisant un filtre réduisant la complexité lors de la détection des agresseurs.

5.6.2 La propagation d'une erreur de calcul de délai.

Nous avons vu dans le chapitre 2.3.4 les conséquences d'une évaluation incorrecte d'un délai élémentaire sur le temps de propagation des chemins contenant ce délai. Ce problème n'est toujours pas résolu lorsque nous travaillons avec des intervalles d'instabilité car les limites $(u; d)$ de ceux-ci correspondent aux temps de propagation extrêmes à travers les portes. Ils peuvent donc être -et même seront- entachés d'erreurs. Par contre, comme nous avons supposé qu'à chaque fois les agressions "pire cas" se produisent, les délais élémentaires sont soit légèrement trop optimistes, soit légèrement trop pessimistes. Les intervalles d'instabilité sont donc probablement plus larges que ce qu'ils seront en réalité, ce qui devrait limiter l'effet d'une erreur d'évaluation d'un délai élémentaire.

A ce jour, et malgré les différents essais que nous avons fait, il faut admettre que nous manquons de recul pour évaluer ce phénomène d'accumulation d'erreurs, et la façon dont il est atténué par l'approche pire cas ou meilleurs cas. Si nous constatons un problème à ce niveau, il est toujours possible de calculer les intervalles d'instabilité en prenant systématiquement une marge de quelques pour cent sur les délais élémentaires.

5.6.3 Les transitions parasites.

L'un des principaux reproches faits aux analyseurs temporels est l'obtention de faux chemins statiques, dont nous avons brièvement parlé au début de ce chapitre. Un faux chemin statique est un chemin qui ne peut être observé logiquement. Sur l'exemple de la figure 5.16, la valeur logique du signal out est toujours 0 car les deux entrées du ET sont toujours opposées. Tous les chemins qui passent par ces portes sont donc des faux chemins statiques. La détection de ce genre de situation est simple dans ce cas, mais dans beaucoup d'autres situations, la dépendance des données peut être infiniment plus complexe.

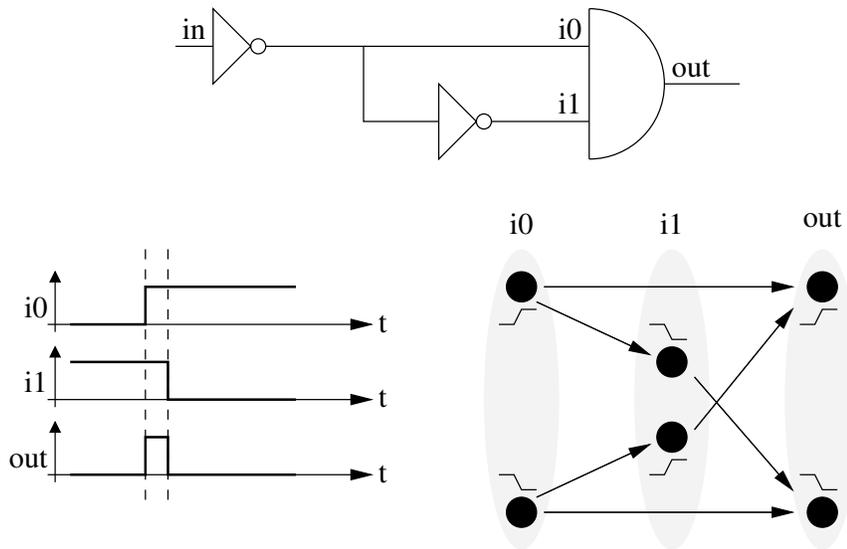


FIG. 5.16 – Exemple de faux chemin statique, vrai chemin dynamique.

Pourtant, lorsqu'on regarde les valeurs électriques des signaux, on s'aperçoit que le retard introduit par l'inverseur entre les signaux $i0$ et $i1$ est suffisant pour avoir simultanément les deux entrées de la porte ET à 1, et donc de faire basculer la sortie pendant un court moment. Il s'agit donc d'un vrai chemin dynamique. Nous n'allons pas aborder ici plus en détail le problème des faux chemins dans l'analyse temporelle des circuits, mais seulement leurs conséquences dans l'analyse temporelle avec prise en compte des effets de couplage.

En pratique, le fait que dans HiTas nous ne supprimons pas les faux chemins statiques ne pose pas de problèmes. Lorsque nous étudions les effets de couplage capacitifs, ce n'est pas le comportement logique des signaux qui nous importe, mais le comportement dynamique. Le fait que le graphe de causalité soit construit de façon systématique est donc un avantage, puisque nous pouvons toujours détecter les agressions.

5.6.4 Critique du critère d'observabilité

Considérons l'exemple de la figure 5.17. Nous supposons qu'il y a quatre chemins possibles, de 1 vers 2 et 4, et de 3 vers 2 et 4. Tous ces chemins passent par les signaux A et B. Sur les chronogrammes de la figure 5.18, nous avons représenté la stabilité calculée pour les sommets A et B, ainsi que la stabilité d'un signal agresseur du signal B. Nous avons également représenté la stabilité du sommet B, due aux événements qui viennent du registre 1 et du registre 3. Le signal B, suivant le critère d'observabilité, n'est pas agressé par le signal agr. Cela implique que nous

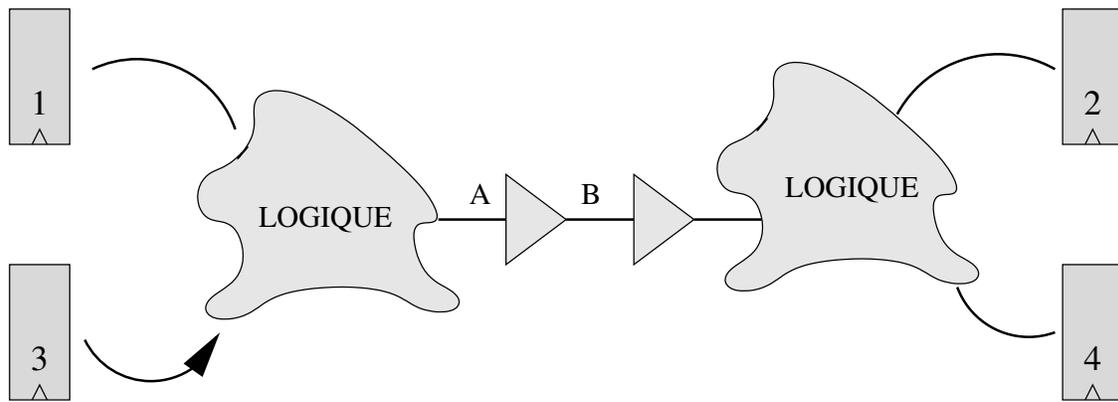


FIG. 5.17 – Implication du critère d’observabilité sur un exemple simple.

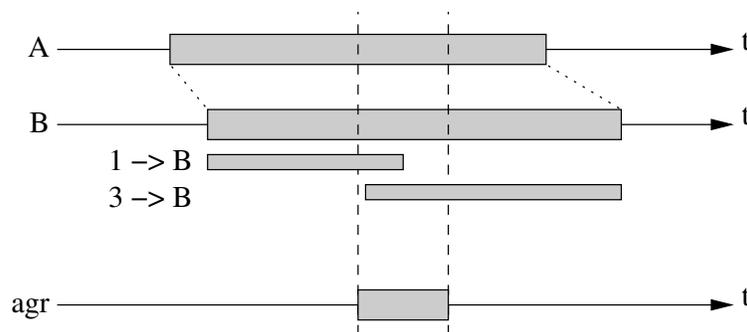


FIG. 5.18 – Chronogrammes correspondant aux sommets A et B de la figure 5.17

ne modifierons pas les temps de propagation minimum et maximum entre A et B. Considérons maintenant la stabilité des événements provenant du registre 1. Le critère d’observabilité appliqué à cet intervalle donne comme résultat que le temps de propagation maximum de A vers B doit être modifié. De même, le temps de propagation minimum de A vers B doit être modifié lorsqu’on considère les événements provenant du registre 3.

Les événements qui proviennent du registre 3 sont ceux qui détermineront, parmi ces quatre chemins, les temps de propagation les plus longs de ce circuit arrivant sur les registres 2 et 4. Lorsque nous prenons en compte la stabilité sur le signal B due aux événements du registre 3 seul, le temps de propagation maximum entre A et B n’est pas modifié, car il n’y a pas d’agression observable dans ce cas. De même, les événements qui proviennent du registre 1 sont ceux qui détermineront le temps de propagation le plus court du circuit. La stabilité calculée sur le signal B due à ce registre n’entraîne pas non plus une agression observable. La détermination des chemins critiques du circuit est donc conforme à ce que nous attendions avec l’introduction

du critère d'observabilité, même lorsque nous regardons en détail l'origine de la stabilité d'un sommet.

Le problème concerne le calcul de la chaîne courte en provenance du registre 3 et le calcul de la chaîne longue en provenance du registre 1 une fois que l'on a caractérisé tous les arcs de la figure avec les agressions observables. Ces chaînes auront des valeurs fausses : en ne modifiant pas le temps de propagation minimum entre A et B, la chaîne courte en provenance du registre 3 aura une valeur plus grande que celle que l'on risque d'obtenir. De même, en ne modifiant pas le temps de propagation maximum entre A et B, la chaîne longue en provenance du registre 1 aura une valeur plus petite que celle que l'on risque d'obtenir.

Pour calculer les délais des chemins partant d'un registre particulier, en y appliquant le critère d'observabilité, il faudrait calculer la stabilité des signaux sur le chemin en fonction du registre émetteur. Ceci est bien sûr trop contraignant, puisqu'il faudrait le réaliser autant de fois qu'il y a de registres dans le circuit. Nous conservons donc le critère d'observabilité, en sachant qu'il permet d'obtenir les temps de propagation critiques du circuit au niveau des récepteurs, mais qu'il ne permet pas de garantir les autres délais. La décision de conserver ou non la notion d'agression observable n'est pas contraignante au niveau d'une implémentation logicielle, cela peut même être une simple option paramétrable. Enfin, cet effet ne se propage pas : ce n'est pas parce qu'on n'a pas modifié la stabilité du signal B pour un chemin particulier que les intervalles d'instabilité des signaux en avals seront faussés : ils sont toujours calculés par rapport au pire cas.

En conclusion, l'application du critère d'observabilité permet d'obtenir des résultats réalistes pour les chemins extrêmes, et peut être faux pour les autres. Lorsqu'on ne prend pas en compte ce critère, pratiquement tous les chemins sont trop pessimistes, car dans le graphe les intervalles d'instabilité grandiront bien plus rapidement.

L'application du critère d'observabilité a également un impact sur l'évaluation des temps de propagation des agresseurs. Reprenons la figure 5.7, où nous avons considéré que l'agresseur A n'était pas observable pour la victime V. Dans ce cas, le front sur la victime est évalué sans prendre en compte l'effet Miller sur la capacité de couplage. Lorsqu'on souhaite ensuite calculer le temps de propagation de la porte contrôlant le signal A, l'effet Miller est pris en compte sur la capacité de couplage, puisque V est un agresseur observable sur A. Or, le front sur le signal V vient d'être calculé sans agression. Pour le calcul du temps de propagation sur le signal A, ce front n'est pas correct. Pour résoudre ce problème, il faut définir pour chaque signal

un front qui sera utilisé pour déterminer l'effet Miller lorsque ce signal est considéré comme un agresseur. En théorie, il faudrait calculer autant de fronts qu'il y a de contextes d'agression (voir la figure 6.8). Ce nombre pouvant être assez grand, il est possible de raisonner en pire cas en calculant un front pour la détermination de l'effet Miller en considérant que l'ensemble des agresseurs du signal sont observables. Ce point n'a pas été pris en compte dans le prototype logiciel.

5.7 Différence avec les méthodes d'élimination des agressions

Nous avons vu dans le chapitre concernant l'état de l'art que de nombreuses approches pour déterminer les temps de propagation partent de conditions d'agression systématiques, et éliminent celles qui ne peuvent pas se produire. Voyons quelles sont les différences entre l'approche d'élimination des agressions (EDA), et l'approche de détection des agressions (DDA).

Dans les méthodes EDA, le critère d'observabilité est plus difficile à mettre en œuvre. Considérons la stabilité initiale des deux signaux S1 et S2, calculée initialement en supposant les pires contraintes d'agression possibles, représentés en clair sur la figure 5.19. Lors du test d'agres-

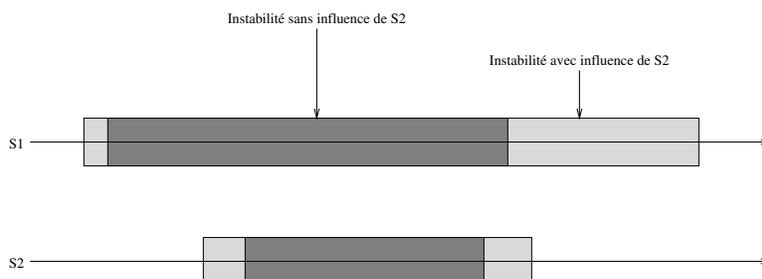


FIG. 5.19 – Application du critère d'observabilité lorsqu'on procède par élimination des agressions.

sion observable, supposons que S2 ne soit plus considéré comme un agresseur observable pour S1. Dans ce cas, l'intervalle d'instabilité de S1 se réduit, représenté en sombre. Si l'on réapplique le critère d'observabilité, S1 peut redevenir victime d'une agression observable de S2. Comme il faut conserver les agressions réellement constatées, il faut annuler la décision précédemment prise de considérer S2 comme non observable, et marquer qu'il ne faut plus appliquer le critère d'observabilité sur cette paire de signaux, au moins jusqu'à ce que la stabilité de S1 ou S2 change. Cela implique un mécanisme assez lourd à mettre en œuvre, et dont la convergence risquera d'être lente.

Les méthodes EDA ont l'inconvénient sur les méthodes DDA d'être intrinsèquement plus pessimistes : le système peut se bloquer dans une situation où l'on détecte des fausses agressions en boucle. C'est pourquoi de nombreuses solutions sont proposées pour éliminer ces agressions, que se soient des méthodes statistiques ([Cad01b], [Cad01c]) ou des relations logiques entre signaux ([FTWCE00], [Syn01c]). Bien sûr, ces méthodes d'élimination sont directement applicables dans les méthodes DDA. Les EDA permettent d'utiliser des modèles de calcul de délai plus précis car il est possible de ne s'intéresser qu'aux chaînes longues lors des itérations : elles sont connues à tout instant, et les autres chemins sont évalués avec un modèle plus simple et légèrement pessimiste. Le temps de calcul et la consommation en mémoire s'en trouvent alors grandement réduits. Dans les méthodes DDA, il faut attendre la dernière détection d'agression pour déterminer de manière sûre les chaînes longues et les chaînes courtes.

5.8 Conclusion

Nous avons vu dans ce chapitre que l'analyse de stabilité est un élément fondamental dans la détermination des temps de propagation extrêmes. Cette analyse nous permet de sélectionner parmi les n agresseurs d'un signal les m agresseurs réellement actifs, et grâce au modèle de calcul des délais, nous sommes maintenant en mesure de pouvoir déterminer des temps de propagation dans des conditions réalistes. Ce dernier point a nécessité l'introduction de la notion d'agression observable. Nous avons également cerné les différences entre une analyse temporelle mono-intervalle et multi-intervalle, ainsi que l'effet des faux chemins.

Afin de s'assurer que les hypothèses que nous avons faites sont applicables, la meilleure solution est la réalisation d'un outil mettant en œuvre les résultats que nous avons obtenus dans ces deux derniers chapitres. Cet outil est présenté dans le chapitre suivant.

Chapitre 6

Implémentation : Le logiciel Ctk.

Nous allons maintenant présenter une réalisation utilisant les algorithmes développés dans les chapitres précédents. Ce prototype porte le nom de Ctk, et permet l'analyse temporelle de circuits en prenant en compte la diaphonie au niveau bloc. Pour réaliser un prototype, nous avons besoin d'au moins trois fonctionnalités : une fonctionnalité pour construire un graphe temporel à partir d'un fichier au format Spice, une fonctionnalité pour réaliser l'analyse de stabilité, et une fonctionnalité pour évaluer les délais en fonction des couplages. Nous allons étudier les deux premiers qui existent déjà dans les paragraphes 6.3 et 6.7, et nous présenterons ensuite le calcul des délais et l'intégration de l'ensemble. Dans le paragraphe 6.4, nous donnerons les spécifications de notre outil. Nous verrons ensuite les modifications que la diaphonie apporte sur le graphe de causalité dans le paragraphe 6.5, puis nous en déduirons la structure de données appropriée dans le paragraphe 6.6. Nous présenterons ensuite l'architecture du logiciel dans le paragraphe 6.8, puis nous présenterons les conclusions sur le développement de l'outil.

6.1 Notations et conventions.

Nous nous référons dans ce chapitre à des programmes, des bibliothèques, et des structures de données. La structure de données est la représentation en mémoire d'un ensemble d'objets ayant un rapport entre eux. Une bibliothèque est un ensemble de services (ou fonctions) qui servent à manipuler une structure de données. Un programme n'est qu'un appel à quelques services offerts par différentes bibliothèques. Un programme bien conçu ne doit contenir aucune intelligence, et tout déléguer aux bibliothèques qu'il utilise. Une structure de données peut très bien se construire par-dessus une ou plusieurs autres structures de données différentes, et donc une bibliothèque peut faire appel à d'autres bibliothèques. Cela est très souvent le cas en pratique.

Bibliothèque	Origine	Préfix des fonctions et structures	Logiciels qui l'utilisent	Bibliothèques appelées directement
TTV	Avertec	ttv_*	HiTas (Avertec), Stb, Ctk	
TTV	Alliance	ttv_*	HiTas (Alliance)	
STB	Avertec	stb_*	Stb, Ctk	TTV
CTK	Alliance	ctk_*	Ctk	TTV
MBK	Avertec	lo*	HiTas (avertec)	
MBK	Alliance	lo*	HiTas (Alliance)	

TAB. 6.1 – Principales bibliothèques utilisées.

L'avantage d'utiliser un tel découpage lors de la conception d'outils est de bénéficier le plus possible de l'existant. De plus, le concepteur n'a à connaître que la description de la structure de données et le mode d'emploi des services offerts par la bibliothèque pour pouvoir l'utiliser. Il n'a pas besoin de savoir comment fonctionne en interne cette bibliothèque, qui peut être très complexe. Dans la chaîne de CAO Alliance, ce concept de bibliothèques est constamment utilisé. Par exemple, lorsqu'on veut créer une structure de données correspondant à un fichier sur disque, il n'y a qu'à effectuer un appel à un service de la bibliothèque Alliance. La bibliothèque Alliance se charge alors de créer cette structure et de faire appel à la bibliothèque spécialisée qui va lire la structure de données en mémoire selon le format de fichier choisi. Dans ce cas particulier, cela nous a permis d'intégrer à Alliance la possibilité de pouvoir décompresser et compresser les fichiers à la volée, sans aucune modifications des bibliothèques et programmes qui font appel à cette fonction de lecture des fichiers.

Nous avons résumé dans le tableau 6.1 la liste des principales bibliothèques dont nous parlerons dans ce chapitre. A titre d'indicatif, le logiciel Ctk utilise les services de 21 bibliothèques, dont 3 sont propres au système d'exploitation.

Pour différencier les logiciels et les bibliothèques dans la suite, nous écrirons en majuscules (ex **CTK**) la bibliothèque et la structure de données correspondante, et en minuscules (ex Ctk) le programme du même nom qui appelle cette bibliothèque. La bibliothèque **MBK** citée ici regroupe l'ensemble des bibliothèques de base de la chaîne de CAO Alliance. Tous les programmes présentés ici utilisent cette base. Parmi les services offerts par **MBK**, citons la ma-

nipulation des vues structurelles en mémoire, la définition de certaines structures communes, les "parser" et les "driver". Dans la liste des bibliothèques, nous avons une colonne "Origine". En effet, depuis la création de la société Avertec, les sources de la chaîne de CAO Alliance et les sources de la société Avertec ont évolué différemment. Cela a pour conséquence que les bibliothèques ne sont plus compatibles au niveau binaire, et parfois au niveau source. Or, nous n'avons pas à notre disposition les sources des bibliothèques en provenance d'Avertec. Comme nous souhaitons utiliser l'analyseur de stabilité d'Avertec, il est important de nous interfacer avec la bibliothèque **STB** d'Avertec. Cette bibliothèque **STB** s'appuie sur un graphe temporel de la bibliothèque **TTV**. Comme la bibliothèque **TTV** d'Avertec est compatible avec la bibliothèque **TTV** d'Alliance, la bibliothèque (**CTK**) qui s'appuie sur la bibliothèque **TTV** d'Alliance pourra aussi être compilée en utilisant les bibliothèques d'Avertec. Cela n'aurait pas été si simple si nous avions utilisé directement la vue structurelle de **MBK** puisqu'il n'y a plus de compatibilité entre les bibliothèques d'Alliance et celles d'Avertec.

6.2 Présentation de Ctk.

Ctk est un programme qui récupère le graphe de causalité créé par HiTas, et en modifie les délais afin qu'ils prennent en compte les effets de couplage. Le contexte d'utilisation de cet outil est représenté sur la figure 6.1. Le terme Ctk correspond aux lettres du mot CrossTalk. Le fichier d'entrée lu par HiTas est au format Spice. Pour le calcul des délais, les informations technologiques sont fournies par un fichier qui contient les paramètres MCC des transistors. HiTas fournit un graphe de causalité dont les arcs contiennent les délais évalués sans agressions. Je l'ai modifié afin qu'il sorte deux autres fichiers contenant les informations électriques nécessaires pour recalculer les délais en fonction des agressions. A partir de ces trois fichiers et de quelques informations de stabilité du circuit (essentiellement la fréquence de l'horloge et la stabilité des entrées), Ctk réalise les itérations comprenant l'analyse de stabilité et le calcul des délais élémentaires. Le fichier de sortie contient les délais corrigés et le détail des agressions.

6.3 Analyseurs temporels disponibles.

Nous avons vu dans le paragraphe 1.4 les principales caractéristiques du logiciel HiTas. Ce logiciel permet la création du graphe de causalité d'un circuit à partir d'un fichier Spice. Nous disposons des sources complètes de ce logiciel à l'état de la création de la société Avertec. Il contient un certain nombre de bugs, mais reste tout à fait utilisable. Il utilise les structures de données de la chaîne de CAO Alliance. Nous l'appellerons par la suite Tas Alc.

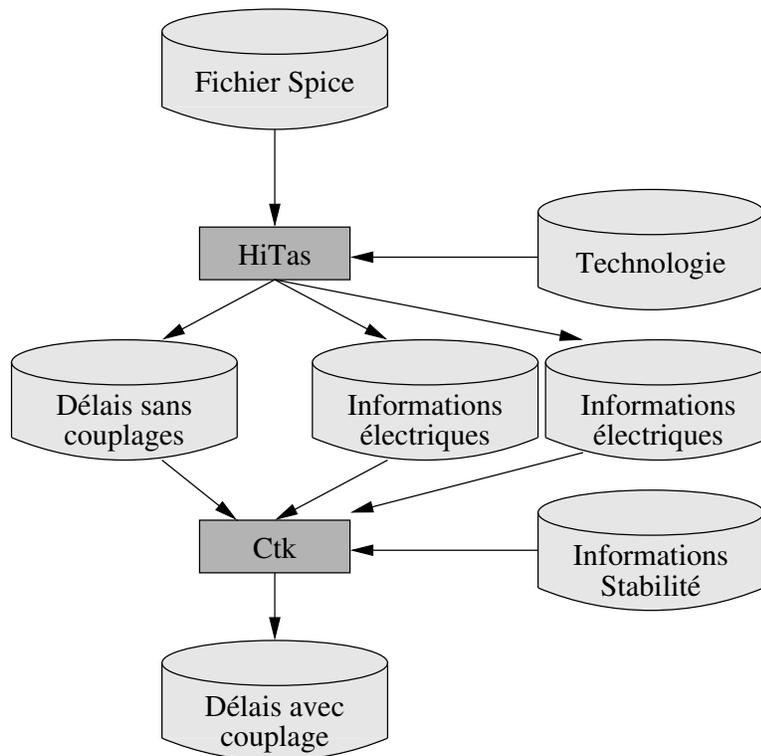


FIG. 6.1 – Utilisation de CTK

La version de HiTas disponible auprès d’Avertéc a connu de très importantes évolutions. Nous ne disposons pas des sources de ce logiciel pour des raisons de confidentialité. Nous avons cependant accès aux bibliothèques qui le constituent. Nous l’appellerons par la suite Tas Avt. Ce logiciel offre une totale compatibilité ascendante avec Tas Alc au niveau des structures de données et des fichiers temporels. Le calcul des temps de traversée des portes s’effectue toujours avec le modèle MCC (avec quelques corrections), et s’évalue toujours en prenant en compte la capacité totale du réseau qu’elle commande.

Les bibliothèques de ces deux versions de HiTas nous permettent donc toutes les deux de construire un graphe de causalité en mémoire, mais aucune n’offre simplement et rapidement la possibilité de calculer plusieurs fois de suite les délais (avec modification de la charge équivalente), car de nombreuses informations électriques sont perdues ou modifiées. Nous allons tout de même utiliser ces analyseurs, en particulier Tas Alc, car cette bibliothèque est le produit de nombreuses années d’études, et prend en compte une très grande diversité de circuiterie logique, le tout ayant été largement validé.

6.4 Cahier des charges

Compte tenu des contraintes vues précédemment, nous avons donc orienté notre choix vers la solution consistant à procéder par correction des délais, et ce, pour des raisons de rapidité de développement.

Notre cahier des charges est donc le suivant :

- Approche à plat. Dans le paragraphe 2.6, nous avons vu que la taille mémoire est un problème qui ne peut pas être ignoré dans le cas d'un logiciel fonctionnel. N'ayant pas abordé ce point dans le cadre de cette thèse, nous allons nous orienter vers une approche à plat pour valider les algorithmes et les concepts présentés ici, tout en se fixant comme objectif de réduire l'encombrement en mémoire, de façon à pouvoir analyser des blocs de taille significative.
- Capacité de charge. Compte tenu de l'importance de l'effet de masquage des capacités de charge sous l'influence des résistances de ligne, nous modéliserons la charge sous la forme d'une capacité calculée par la méthode décrite dans le paragraphe 4.1.
- Calcul des délais sous la contrainte d'agresseurs actifs. La variation de charge équivalente des portes sera calculée avec l'effet Miller. Comme nous procédons par correction des délais, nous utiliserons les relations présentées dans les paragraphes 4.4 et 4.3.
- Prise en compte du bruit. Nous prendrons en compte l'augmentation ou la diminution de la dynamique des signaux sous l'influence du bruit de commutation avec les relations présentées dans le paragraphe 4.4.2.
- Analyse de stabilité. Le calcul des intervalles d'agression entre signaux sera réalisée grâce à la bibliothèque **STB** de la société Avertec. L'avantage d'utiliser cette bibliothèque, plutôt que d'en créer une sous forme forcément plus limitée, est de pouvoir prendre en compte une grande catégorie de circuiterie.

6.5 Influence de la diaphonie sur le graphe de causalité.

6.5.1 Calcul de la variation des délais des arcs de type porte et de type RC.

Pour les arcs correspondant à des délais à travers les portes, HiTas est capable de calculer l'influence de la variation du front d'entrée et de la capacité de sortie sur le temps de propagation de l'arc, au moyen de l'expression :

$$\Delta T_p = T_p - T_{p_0} = R(C_{out} - C_{out_0}) + S(Fin - Fin_0) \quad (6.1)$$

HiTas calcule les paramètres R et S pour les corrections des temps de propagation minimum et maximum. L'influence de la capacité de charge de la porte sur le front de sortie est calculé dans HiTas par la relation :

$$\Delta F = F_{out} - F_{out_0} = R(C_{out} - C_{out_0}) \quad (6.2)$$

On remarque que pour le modèle MCC, la variation du front de sortie d'une porte ne dépend pas du front d'entrée. Les paramètres R et S ne sont normalement calculés que pour les signaux externes, en vue d'une prise en compte de la modification du contexte lors d'une utilisation hiérarchique. De plus, ils ne dépendent que du modèle MCC choisi ici. Si d'autres modèles sont nécessaires, il suffit de changer la caractérisation des arcs de type portes. La seule condition que doit remplir ce modèle sont les dépendances :

$$\begin{aligned} T_p &= t_p(Fin, C_{out}) \\ F_p &= f_p(C_{out}) \end{aligned} \quad (6.3)$$

Ici, le terme C_{out} représente une charge quelconque adaptée au modèle de la porte, et les fonctions $t_p()$ et $f_p()$ ne dépendent que du modèle de la porte. Les délais sur les arcs de type RC dépendent du front sur l'entrée de ce réseau :

$$\begin{aligned} T_{rc} &= t_{rc}(Fin) \\ F_{rc} &= f_{rc}(Fin) \end{aligned} \quad (6.4)$$

Les fonctions $t_{rc}()$ et $f_{rc}()$ ne dépendent que du modèle choisi pour calculer les temps et les fronts dans les interconnexions. Contrairement aux fonctions $t_p()$ et $f_p()$, ces fonctions ne sont pas figées : elles dépendent des agressions par couplage capacitif.

6.5.2 Charge au deuxième ordre.

En sortie de chaque porte, nous voulons représenter la charge équivalente initiale au deuxième ordre (paragraphe 4.1). Il faut donc conserver les paramètres y_1 , y_2 et y_3 . Cette information doit évidemment être mémorisée sur les signaux temporels qui correspondent à la sortie d'une porte.

6.5.3 Informations de couplage.

Ensuite, il faut représenter les informations de couplage capacitif qui existent entre les interconnexions. La question que nous nous sommes posée est de savoir où mettre cette information et comment la représenter. Rappelons que pour chaque agresseur potentiel nous avons cinq coefficients pour modifier la charge au deuxième ordre (équation 4.64), et chaque délai d'Elmore

est modifié avec l'utilisation d'une constante (paragraphe 4.3). Dans l'exemple de la figure 6.2, il faut prendre en compte le fait que la capacité de couplage entre la sortie de la porte 3 et l'en-

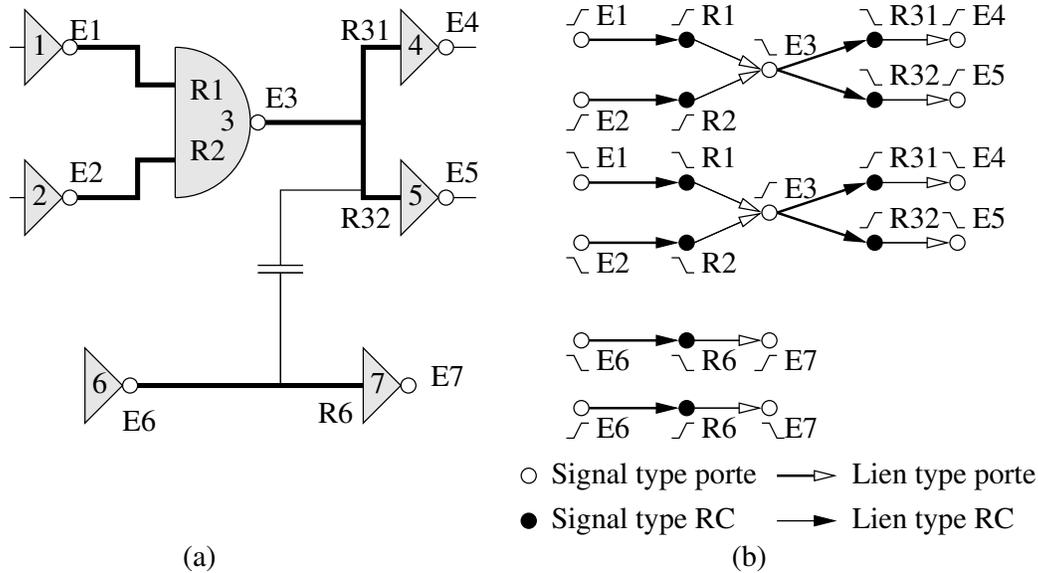


FIG. 6.2 – Effet d'une capacité de couplage sur le graphe temporel.

trée de la porte 5 modifie le temps de propagation et le front de sortie sur les arcs $(E_3;R_{32})$, $(E_3;R_{31})$, $(R_1;E_3)$ et $(R_2;E_3)$. De plus, la modification des fronts sur R_{31} et R_{32} modifie également les temps de propagation sur les arcs $(R_{31};E_4)$ et $(R_{32};E_5)$. L'influence de la diaphonie reste contenue sur ces 6 arcs car dans le modèle MCC, on considère que la variation du front de sortie des portes 4 et 5 ne dépend pas de la variation du front d'entrée sur R_{31} et R_{32} . Nous pouvons insister ici sur le fait que la non-dépendance de la variation du front de sortie de la porte en fonction de son front d'entrée (ce qui est le cas pour MCC) permet de confiner l'effet des couplages capacitifs aux réseaux concernés. Dans le cas contraire, le changement d'un seul front entraînerait la nécessité de recalculer tout le graphe temporel en aval. Cette influence ne se fait cependant pas sentir sur plus de deux ou trois portes. Nous résumons sur cet exemple les 3 effets des capacités de diaphonie :

- Modification des temps de propagation et des fronts de sortie dans les portes par effet de charge apparente : arcs $(R_1;E_3)$ et $(R_2;E_3)$.
- Modification des délais et des fronts dans les réseaux RC : arcs $(E_3;R_{31})$ et $(E_3;R_{32})$. L'effet sur le front est double, puisque le front de sortie de la porte a changé et la capacité de couplage entraîne un changement dans la propagation du front.
- Modification du temps dans les portes à cause du front d'entrée : arcs $(R_{31};E_4)$ et arcs $(R_{32};E_5)$.

Nous choisissons donc de conserver la liste des agresseurs dans les signaux correspondant aux portes E_i . Pour chaque agresseur on trouvera les coefficients de modification de la charge, et une liste de délais RC à modifier, avec leurs constantes.

6.6 Structures de données CTK.

6.6.1 Contexte.

Le programme Ctk et la bibliothèque **CTK** ont été écrits en langage C, et compilé sous Solaris et sous Linux. Ce programme est assez petit puisqu'il contient moins de 4000 lignes de code. La structure de données du logiciel est donnée dans l'annexe A. Nous avons choisi de séparer totalement le graphe de causalité des informations de couplage, en encapsulant le graphe de causalité **TTV** sans modification dans la structure **CTK**, utilisant le principe de réutilisation présenté dans le paragraphe 6.1. Deux raisons nous ont conduits à procéder ainsi. La première est purement formelle : le graphe de causalité **TTV** et les informations de couplage **CTK** n'ont pas la même sémantique. La seconde est pratique. La bibliothèque d'analyse de stabilité **STB** prend en entrée un graphe de causalité au format **TTV**. Comme nous ne disposons pas des sources de la bibliothèque **STB**, il nous est impossible d'ajouter des champs supplémentaires dans les structures **TTV** et de recompiler le tout. C'est la raison pour laquelle avons construit la vue **CTK** "par-dessus" la vue **TTV**.

6.6.2 Présentation des structures

Un exemple graphique de la vue **CTK** est représenté sur la figure 6.3.

La vue principale.

La première structure que nous définissons est **ctkview_list**. Elle contient la liste des signaux sur lesquels nous souhaitons prendre en compte les effets de couplage. Cette structure utilisant les informations d'une structure **TTV**, nous prévoyons un champ pour accéder à cette vue (champ **TTVFIG**).

Le signal.

C'est la structure **ctksig_list**. Elle contient évidemment une référence vers le signal de type porte correspondant dans la vue **TTV** (champs **SIGNAL**). Elle contient les 3 valeurs représentant la charge initiale au deuxième ordre (champs **Yi**), et les valeurs maximums des résistances

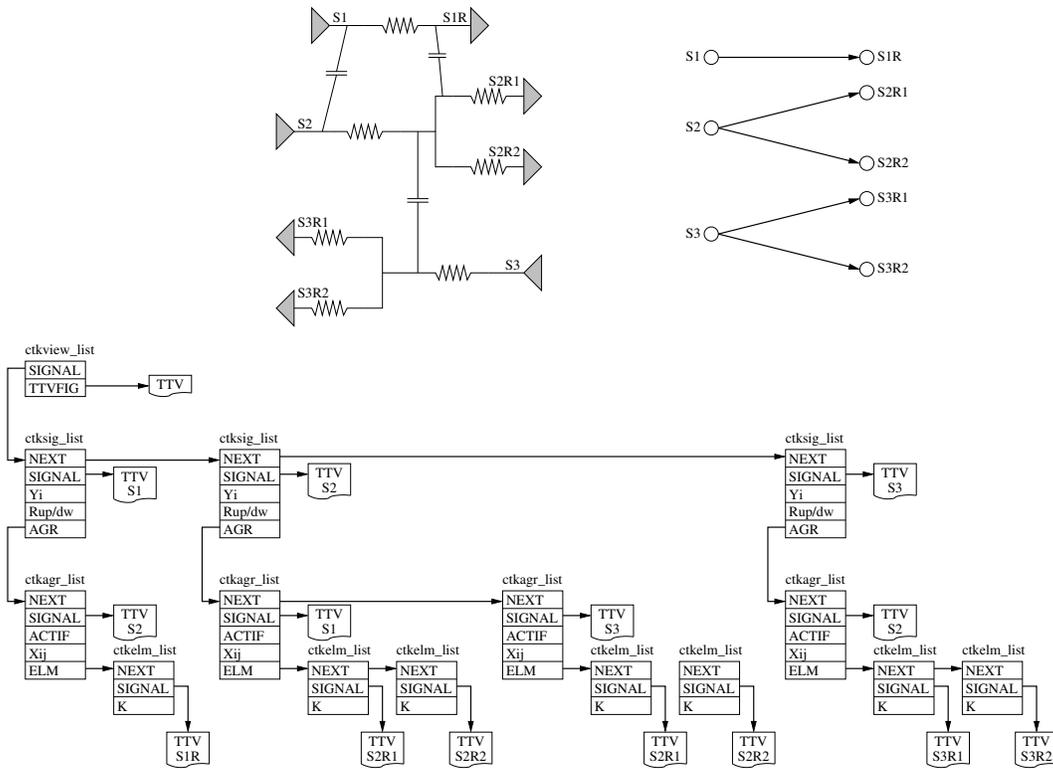


FIG. 6.3 – Exemple d’une représentation des agressions dans une vue CTK.

vers la masse et vers VDD pour le calcul du bruit de commutation (champs **Rup/Rdw**). Cette structure contient aussi la liste des agresseurs potentiels de ce signal (champ **ELM**).

L’agresseur.

C’est la structure **ctkagr_list**. Cette structure contient une référence vers le signal de type porte dans la structure **TTV** qui peut agresser le signal de **ctksig_list** (champs **SIGNAL**). Associé à cet agresseur, on trouve les cinq paramètres pour le calcul de la modification de la charge apparente (champs **Xij**), et la liste des délais RC à modifier (champs **ELM**). Finalement, un ensemble de bits dans le champ **ACTIF** mémorise les quatres agressions possibles : Pour une transition haute du signal **ctksig_list**, un bit indique si le début de la période d’instabilité doit être changé, un autre la fin de cette période, et de même pour une transition basse.

La modification d’un délai d’Elmore.

La structure **ctkelm_list** contient deux champs : le premier est une référence vers le signal de type RC dans la vue **TTV** connecté au signal de type porte du **ctksig_list** (champs **SIGNAL**),

le second est la constante utilisée pour modifier le délai correspondant (champs **K**).

6.6.3 Réévaluation des délais correspondant aux portes.

Cette structure est indépendante de **CTK** et de **TTV**. Elle pourrait rentrer dans une troisième bibliothèque : celle qui réévalue les délais des portes. Nous avons cependant choisi de l'intégrer à **CTK** par facilité de développement. Cette structure est accessible dans le champ **USER** des arcs de type porte. Ce champ, présent dans la plupart des structures de données Alliance, permet d'ajouter dynamiquement des composants à une structure existante. Cette structure s'appelle **ctkcaracline**. Elle contient les valeurs initiales de l'arc, ainsi que les paramètres permettant la réévaluation des délais. Ces valeurs sont les paramètres Tp_0 , R , $Cout_0$, S et Fin_0 de l'équation 6.1, le courant I_{max} lors de la commutation pour déterminer la capacité équivalente (équation 4.35), et le front de sortie initial.

Une structure similaire existe aussi pour les arcs de type RC : elle contient les valeurs initiales des fronts et des temps de propagation.

6.7 L'analyseur de stabilité STB de la société Avertec.

6.7.1 Présentation.

Choix de cet analyseur.

L'analyseur de stabilité **STB** que nous avons utilisé provient de la société Avertec. Cette bibliothèque a pour fonction de vérifier les contraintes de stabilité sur les points mémorisants et les connecteurs d'un bloc, à partir de sa vue temporelle. Nous avons choisi d'utiliser cette bibliothèque pour quatre raisons :

- Cette bibliothèque s'appuie sur les structures de données de l'analyseur temporel HiTas. Cela permet d'effectuer les itérations sans avoir à passer par un programme externe, et donc d'éviter de très coûteuses opérations de lecture et d'écriture sur disque.
- **STB** calcule des intervalles d'instabilité, ce qui est tout à fait adapté à notre problème.
- Il est possible d'obtenir la stabilité sur tous les sommets de la figure temporelle. En outre, un ensemble de fonctions est disponible pour la manipulation des intervalles d'instabilité.
- Il est possible d'analyser les circuits comprenant des points mémorisants sur état ou sur front. De plus, la propagation des commandes est prise en compte.

Enfin, la bibliothèque **STB** a été conçue dès le départ pour être utilisée dans le contexte d'outils de vérification des contraintes électriques submicroniques. La réalisation de l'outil **CTK** a

contribué à la mise au point de l'outil **STB** par l'usage intensif que nous en avons fait.

Caractéristiques et fonctionnement.

L'analyse de stabilité du circuit peut être globale ou détaillée, mono-intervalle ou multi-intervalle. Rappelons la signification de ces 4 termes :

- Analyse globale. Dans ce type d'analyse, les intervalles d'instabilité ne sont calculés que sur les sommets du graphe des chemins critiques.
- Analyse détaillée. Les intervalles d'instabilité sont calculés sur tous les sommets du graphe de causalité.
- Analyse mono-intervalle. On obtient pour chaque sommet un seul intervalle d'instabilité "pire cas".
- Analyse multi-intervalle. Dans ce cas, il est possible pour chaque sommet d'obtenir plusieurs intervalles d'instabilité non recouvrants. Le logiciel gère entièrement les problèmes de chevauchement des intervalles d'instabilité.

Nous avons bien sûr utilisé l'analyse détaillée, et nous souhaitons pouvoir réaliser une analyse mono-intervalle ou multi-intervalle pour observer les éventuelles différences.

Pour fonctionner, **STB** a besoin d'informations qu'il prend dans un fichier. Ces informations sont les valeurs de stabilité sur les connecteurs d'entrée, la description des horloges (qui peut être multi-phases et multi-domaines), et les marges requises sur les sommets de stabilité.

6.7.2 Conventions et structures de données utilisées par STB.

L'intervalle d'instabilité.

L'intervalle d'instabilité de base est décrit par deux nombres, D et U . Un signal devient instable à partir de l'instant D , et redevient stable à l'instant U , avec la convention $D < U$. Comme il peut y avoir plusieurs intervalles d'instabilité pour une phase donnée, ces intervalles peuvent être chaînés. Cette structure s'appelle **stbpair_list**.

La stabilité sur les sommets du graphe de causalité.

Les informations de stabilité sont placées sur les sommets dans une structure **stbnode**. Cette structure contient un tableau dont les éléments sont les listes des intervalles d'instabilité de chaque phase d'horloge sur ce sommet.

6.7.3 Type d'analyse de stabilité requise.

Lorsqu'une vue temporelle contient des arcs correspondant à des temps de propagation à travers des interconnexions résistives, un outil d'analyse de stabilité ayant pour but de vérifier les contraintes de stabilité n'est pas directement utilisable dans un contexte où l'on cherche avant tout à observer les agressions potentielles entre signaux. Considérons le cas de la sortie d'une porte connectée aux entrées de plusieurs autres portes au travers d'un réseau RC (figure 6.4). L'analyseur de stabilité **STB** nous donnera des intervalles d'instabilité sur les sommets E,

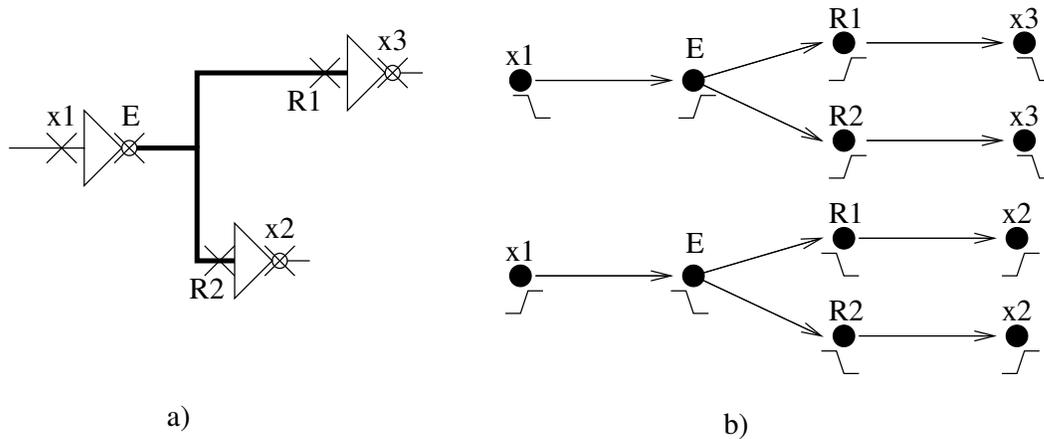


FIG. 6.4 – (a) Portes reliées par un réseau RC, (b) Graphe de causalité correspondant

R1 et R2. Dans la pratique, les intervalles d'instabilité de E, R1 et R2 peuvent ne pas se chevaucher, comme nous l'avons représenté sur la figure 6.5 (a). Le problème apparaît alors de pouvoir être en mesure de détecter un recouvrement des intervalles d'instabilité entre les sommets correspondant au signal victime et son agresseur. Nous venons donc mettre en évidence que la stabilité calculée sur les sommets n'est pas la bonne approche. Il faut calculer une instabilité unique en sortie de chaque porte qui corresponde à l'instabilité de tout le réseau RC qu'elle contrôle (voir la figure 6.5 (b)). Cette instabilité correspond à l'union des intervalles d'instabilités des arcs RC. Nous définissons la stabilité d'un arc RC par un ensemble de n couples $(d; u)$, à l'intérieur desquels une transition est susceptible de se produire n'importe où sur le réseau RC entre l'émetteur et le récepteur correspondant à l'arc. A l'extérieur de ces couples de valeurs, le signal est rigoureusement stable. Malheureusement, la bibliothèque **STB** ne fournit que l'instabilité sur les sommets du graphe de causalité défini dans le paragraphe 1.4.4, et non pas sur les arcs. Cependant, nous savons que l'instabilité d'un sommet correspondant à l'extrémité d'un

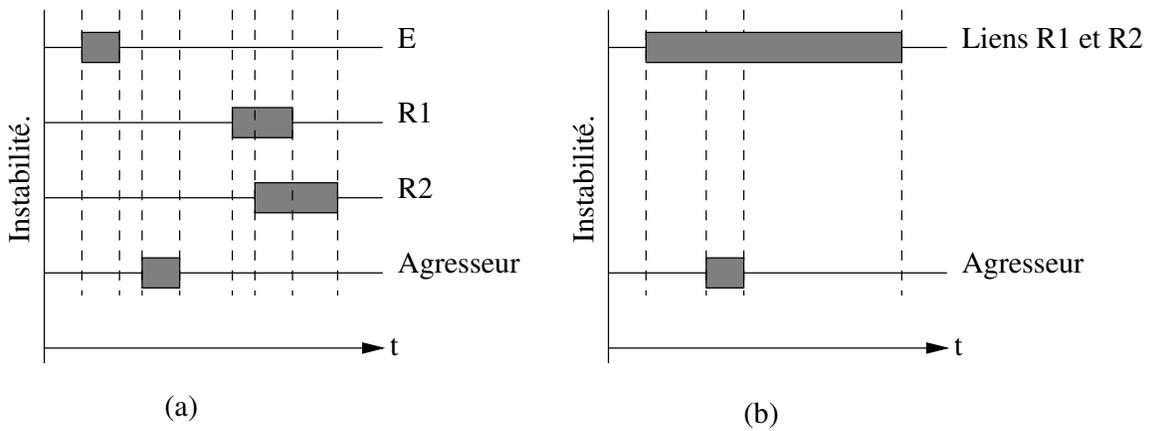


FIG. 6.5 – Intervalles d’instabilité. Situation correspondant à la figure 6.4 où la détection d’une agression est problématique. (a) Stabilité sur les sommets. (b) Stabilité sur les arcs.

réseau RC se calcule à partir de l’instabilité du sommet sur la porte, avec les relations :

$$\begin{aligned}
 D_{rc} &= D_p + T_{p_{min}RC} \\
 U_{rc} &= U_p + T_{p_{max}RC}
 \end{aligned}
 \tag{6.5}$$

L’instabilité d’un arc RC s’obtient en retranchant du début de l’intervalle d’instabilité D , le temps de propagation minimum de cet arc (voir la figure 6.6). La raison de procéder de cette

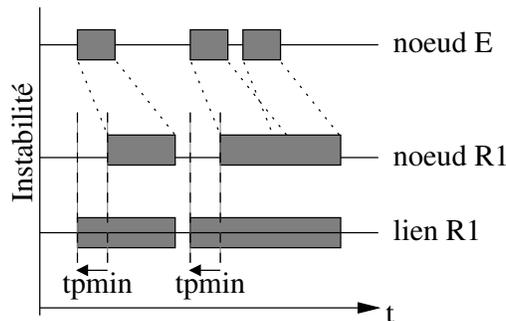


FIG. 6.6 – Deux intervalles d’instabilité sur le sommet en sortie d’une porte peuvent former un seul intervalle d’instabilité sur un sommet RC.

façon plutôt que de considérer que l’instabilité sur l’arc débute en même temps que l’instabilité sur le sommet de la porte E est que l’on souhaite conserver la possibilité d’avoir une approche multi-intervalle. Il est difficile de connaître sur un sommet RC quelles sont les intervalles du sommet de l’émetteur qui lui ont donné naissance, comme cela était le cas sur la figure 6.6.

Plus généralement, ce qui nous intéresse n'est pas de connaître l'instabilité sur les sommets, mais de pouvoir détecter si une transition sur une interconnexion peut avoir un effet sur une autre interconnexion. Nous avons vu que considérer uniquement la stabilité sur les sommets temporels n'était pas suffisante : il faut calculer une stabilité sur le sommet de type porte qui inclut la stabilité des arcs RC. Ce n'est toujours pas suffisant car électriquement, il peut y avoir des parties de l'interconnexion qui ne correspondent à aucun sommet, et qui peuvent commuter bien après la stabilité calculée à partir des sommets dont nous disposons. Cela se produit lorsque le concepteur d'un circuit réutilise un bloc sans connecter toutes les sorties. Certains fils restent alors en l'air, comme par exemple la dernière retenue dans un additionneur *ripple carry* (figure 6.7).

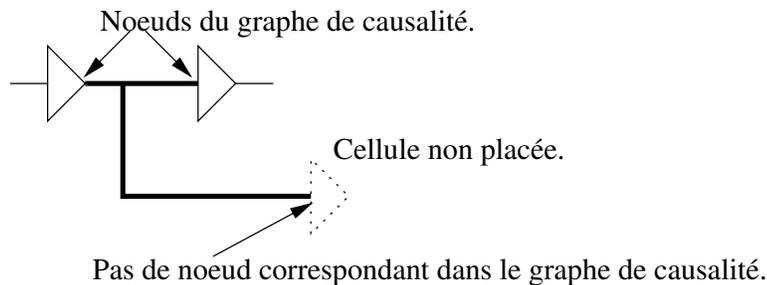


FIG. 6.7 – Fil de routage laissé en l'air dont l'extrémité ne correspond à aucun sommet du graphe temporel.

En conclusion, l'analyse de stabilité ayant pour finalité la vérification du respect des contraintes de stabilité sur les sommets du graphe de causalité est adaptée à la détection des agressions. Les résultats fournis doivent cependant être interprétés, et dans certains cas rares, la détection de certaines agressions risque de ne pas être possible (cas des fils n'aboutissant à aucun sommet).

6.7.4 Post-traitement à effectuer sur les structures de données de STB

Nous cherchons à détecter les agressions qui peuvent survenir entre les interconnexions. Compte tenu des informations fournies par **STB** et de ce que nous venons de voir, une solution est de prendre en compte l'information de la stabilité de la totalité de l'interconnexion en sortie de la porte, et ce, indépendamment des phases d'horloge considérées.

Nous commençons donc par retrancher sur chaque début intervalle d'instabilité D correspondant à un sommet de type RC le temps de propagation minimum de l'arc qui nous a permis

d'arriver dessus. Dans les circuits dual CMOS, cet arc est unique. Cette opération est réalisée en utilisant la fonction de la bibliothèque **STB** `stb_adddelaystbpair()`. Ensuite, nous regroupons les intervalles d'instabilité modifiés des sommets RC sur les sommets de type porte avec la fonction `stb_mergestbpairlist()`, qui gère tous les problèmes de chevauchement qui surviennent. Ces opérations sont réalisées pour chacune des phases.

A ce stade, nous avons calculé la stabilité complète des interconnexions en sortie de chaque porte. Nous n'avons cependant pas pris en compte le problème évoqué précédemment concernant les interconnexions sur lesquelles les sommets de stabilité n'ont pas tous été construits.

6.7.5 Utilisation des informations de STB dans CTK.

La seule information calculée par **STB** utilisée dans **CTK** est la liste des intervalles d'instabilité communs entre un signal victime et une liste d'agresseurs donnée. Cette information est obtenue par la fonction `stb_overlapdev()`, sous la forme d'une liste d'intervalles. Cette fonction prend comme arguments le signal victime et une liste d'agresseurs.

Chaque élément de la liste renvoyée par `stb_overlapdev()` contient une liste de signaux dont les intervalles d'instabilité se recouvrent, ainsi que le début et la fin de chacun de ces intervalles. Ces intervalles sont mémorisés dans une structure **stbgap_list**. Considérons l'exemple de la figure 6.8, sur laquelle nous avons représenté les 3 contextes d'agression d'un signal agressé par deux autres signaux.

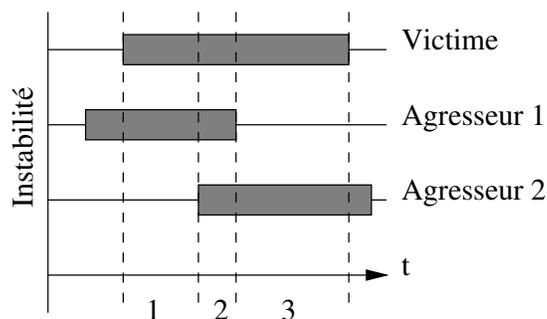


FIG. 6.8 – Plusieurs agresseurs correspondant à plusieurs contexte pour la victime.

Dans la structure **stbgap_list**, les temps sont ramenés dans le même référentiel. C'est à dire qu'on n'a plus différentes phase d'horloge à prendre en compte. Lorsqu'on voudra comparer ces intervalles avec le signal victime, dont les intervalles d'instabilité sont triés sur les diffé-

rentes phases d'horloge, il faudra ramener ces intervalles dans le même référentiel que celui utilisé dans les structures **stbgap_list**. Cela est réalisé à l'aide de la fonction *stb_mergestbpairlist()*, qui gère également les chevauchement d'intervalles.

6.8 Architecture et implémentation

6.8.1 Introduction

Nous présentons ici les principaux aspects de la bibliothèque **CTK**, dont nous avons vu la structure de données dans le chapitre 6.6.2, et du fonctionnement du programme prototype. Nous verrons au fur et à mesure l'utilisation des différents services offerts par la bibliothèque, et la façon dont elle s'intègre dans le flot de données de la vérification temporelle avec prise en compte des effets de couplages capacitifs. La description de quelques-uns de ces services est donné dans l'annexe B.

6.8.2 Le flot de données.

Le calcul des paramètres pour les couplages et les informations de charge au deuxième ordre ne peuvent se faire qu'en disposant des sources complètes de HiTas. C'est la raison pour laquelle la création de ces informations se fait en utilisant l'ensemble des bibliothèques **Tas Alc**. Malheureusement, les bibliothèques en provenance d'Avertec ne sont plus compatibles avec les bibliothèques Alliance, en particulier au niveau de **MBK**. Cela signifie qu'il est impossible d'inclure directement la bibliothèque **STB** dans la version de HiTas que nous utilisons ici. Nous devons donc découper le programme que nous voulons réaliser en deux programmes distincts : Le premier, basé intégralement sur les bibliothèques Alliance, crée le graphe temporel **TTV**, la vue **CTK** et les informations nécessaires pour calculer les variations des délais sur les arcs de type porte. Le second programme, Ctk, basé intégralement sur les bibliothèques Avertec, réalisera la boucle de stabilité. Le transfert des données entre ces deux programmes se fera par fichier, qui sont au nombre de trois. Le premier est la vue temporelle classique **TTV**, dans le fichier portant l'extension **.dtx**. La description de ce format de fichier est décrite dans [Dio98]. Le second est la vue **CTK** qui contient les informations de couplage et de modélisation de charge au deuxième ordre, dans un fichier portant l'extension **.ctk**. Finalement, le troisième contient la caractérisation des arcs de type porte (paramètres *R* et *S*), dans un fichier portant l'extension **.car**. Les descriptions de ces deux derniers fichiers sont données dans les annexes C.1 et C.2 page 195. La figure 6.9 présente le flot de données de ces deux programmes.

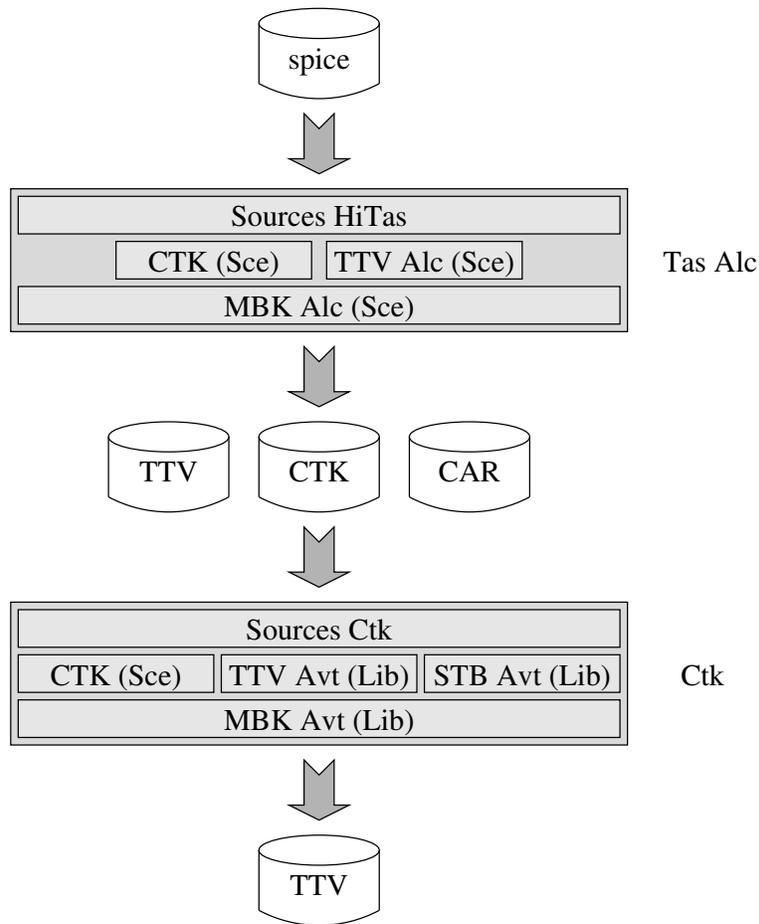


FIG. 6.9 – Flot de données pour l'analyse temporelle avec prise en compte de la diaphonie.

6.8.3 Modifications apportées à HiTas

Les modifications apportées dans les sources des bibliothèques de Tas Alc sont assez limitées :

- Construire la vue **CTK**,
- Forcer la caractérisation ($R, S, I_{max}, R_{up}, R_{down}$) de tous les arcs de type porte,
- Ecrire dans un fichier au format **.car** la caractérisation des arcs de type porte,
- Ecrire dans un fichier au format **.ctk** la vue **CTK**,
- Correction de quelques bugs (évaluation des capacités des signaux, ...).

La construction de la vue **CTK** s'effectue à partir de la description en cônes **CNS** [GLM92] de la vue structurelle **MBK** du circuit, une fois que le graphe de causalité **TTV** a été construit. Des informations temporaires sont placées dans les trois vues **MBK**, **CNS** et **TTV** pour retrouver sur chaque signal l'élément correspondant dans les autres vues. La figure 6.10 représente le

passage de la vue en transistor **MBK** à la vue porte **CNS**.

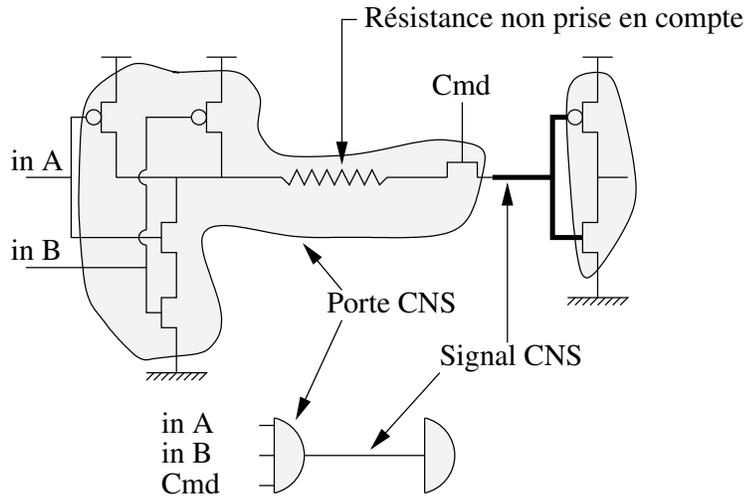


FIG. 6.10 – Vue **CNS** construite sur une représentation **MBK**

Pour créer la vue **CTK**, nous avons fait quelques hypothèses. Tout d’abord, nous avons négligé, en raison de leur influence extrêmement faible, les résistances et les capacités de couplage à l’intérieur des portes qui ne manqueront pas d’apparaître sur les signaux internes lors de l’extraction. Encore qu’il faille garder à l’esprit le problème posé par les transistors de passage comme cela est le cas sur l’exemple précédent. Dans ce type de porte à transistor de passage, il peut parfois arriver que les résistances internes aux portes ne soient pas négligeables, ainsi que des capacités de couplage. Nous n’avons pas résolu ce problème.

A partir des vues **CNS** et **MBK**, la construction de la structure **CTK** se limite à des parcours sur des arbres RC. Pour optimiser à la fois le temps d’exécution et la consommation en mémoire, les parcours se font de manière itérative sur les branches RC, et récursives sur les sommets où l’on trouve plus de trois résistances. La construction de la vue **CTK** n’ajoute pratiquement rien sur le temps total d’exécution de *Tas*.

Nous avons écrit que nous construisons la vue **CTK** une fois que la vue **TTV** était construite. Cela veut dire que dans cette vue **TTV**, la caractérisation temporelle des arcs de type porte s’est fait en prenant comme capacité de charge la capacité totale des interconnexions. La première chose que nous aurons donc à faire dans le programme *Ctk* sera de réévaluer les temps de la vue **TTV** avec une charge apparente au deuxième ordre. Ce choix est justifié par la volonté de ne pas modifier les caractéristiques originelles de l’outil *Tas Alc*.

6.8.4 Le programme CTK

Le calcul des délais avec charge au deuxième ordre

Le programme Ctk commence par lire les trois vues. Tout d'abord la vue **TTV**, puis la vue **CTK**, et finalement la caractérisation des arcs de type porte (fichier **.car**). La première opération à réaliser est la correction des arcs de type porte avec la charge au 3^e ordre. Bien que nous ayons présenté toute la marche à suivre pour la réalisation de ce calcul, il nous manque cependant une information pour le mener à bien : la valeur de la capacité C_{out_0} dans l'équation 6.1. Cette capacité inclut les capacités d'entrée des portes de charge, la capacité de routage, et la capacité de conflit lors de la transition. La valeur exacte de C_{out_0} n'est cependant pas nécessaire, car ce qui nous importe, c'est la variation $C_{out} - C_{out_0}$. Cette variation de capacité apparente ne dépendant dans notre contexte que de la capacité de charge, nous pouvons considérer que C_{out_0} correspond au terme y_1 , et C_{out} à la capacité de charge équivalente au deuxième ordre. Comme nous l'avons vu dans le chapitre 6.5, la modification des fronts en sortie des arcs de type porte a bien sûr une répercussion sur les arcs de type RC, au niveau des délais et des fronts de sortie, qui vont eux-mêmes influencer sur les délais des portes de charge. Cette double influence est prise en compte. Nous avons vu les règles de dépendance par les équations 6.3 et 6.4 page 158 des effets de diaphonies :

$$\begin{aligned}
 Tp &= t_p(Fin, Cout) \\
 Fp &= f_p(Cout) \\
 Trc &= t_{rc}(Fp) \\
 Frc &= f_{rc}(Fp)
 \end{aligned}
 \tag{6.6}$$

L'ordre d'évaluation des 4 équations des arcs est donc fixé par ces règles de dépendance :

1. Calcul des capacités de sortie équivalente C_{out} à partir du courant fourni par la porte et des paramètres y_1 , y_2 et y_3 ,
2. Calcul des fronts de sortie des portes Fp ,
3. Calcul des fronts de sortie Frc et des temps de propagation Trc des réseaux RC,
4. Calcul des temps de propagation des portes Tp , avec les nouveau Fin et $Cout$.

Le séquençement des opérations pour le calcul des délais et des fronts dans le graphe est donc le suivant :

1. Calcul de la capacité équivalente en sortie de chaque porte : fonction *ctk_calclload*
2. Calcul du front de sortie de chaque porte : fonction *ctk_calcslope*
3. Calcul du front et des délais aux extrémités des réseaux RC : fonction *ctk_calcrdelay*

4. Calcul des délais de chaque porte : fonction *ctk_calcdelay*.

Notons que ces 4 fonctions, dont la définition est donnée dans l'annexe B, sont avant tout destinées à calculer des informations dans un contexte diaphonant. Ici, nous nous en servons pour caractériser le graphe temporel en utilisant des charges au deuxième ordre en considérant tous les agresseurs comme inactifs. Ce qui est le cas lors de l'initialisation.

Détection des agressions.

Nous avons vu dans le chapitre 5.3 les conditions pour qu'un ensemble de n agresseurs puisse modifier la stabilité d'un signal, et comment, parmi ces n agresseurs, choisir ceux qui sont observables. Vu la façon dont a été conçue la structure **stbgap_list**, renvoyée par la fonction *stb_overlapdev*, cela ne pose pas de problème. Il faut juste veiller à prendre les bons intervalles lorsqu'il y a plusieurs intervalles d'instabilité (approche multi-intervalle). Il reste cependant un point que nous n'avons pas corrigé : nous avons vu que la fonction *stb_overlapdev* renvoie la liste des agresseurs ayant des intervalles d'instabilité communs avec la victime. Or, nous avons vu dans le chapitre 5.3 qu'une influence peut se faire sentir même si les signaux n'ont pas d'intervalle d'instabilité communs, à cause de la prise en compte des fronts et de la notion d'influence de ces derniers. Ce problème est lié à **STB**, il faudrait prévoir une marge autour des intervalles d'instabilité de la victime dans laquelle les agresseurs seraient pris en compte. Une telle marge vaudrait typiquement :

$$\frac{Fv + Fa}{2} \quad (6.7)$$

Ce phénomène reste cependant peu significatif.

Les itérations de stabilité.

Maintenant que nous disposons de toutes les fonctions de réévaluation des délais et des hypothèses de dépendances du graphe de causalité, la boucle itérative est assez simple à mettre en œuvre. Nous considérons qu'il n'y a initialement aucune agression entre les signaux. Le programme commence par réaliser l'analyse de stabilité. De cette analyse de stabilité, pour chaque signal nous examinons si chacun de ses agresseurs peut avoir un effet observable. Lorsqu'une agression observable est détectée, le *bit* correspondant dans le champ **ACTIF** de l'agresseur est positionné. Pour assurer la convergence de l'algorithme de manière absolue, ce positionnement est définitif. L'utilisation de ce champ est donnée dans l'annexe A.3. Nous disposons donc à cet instant pour chaque signal la liste de ces agresseurs observables. Nous pouvons donc mettre en place les itérations pour déterminer la valeur de l'effet Miller entre chaque paire de signal victime/agresseur. Pour cela, nous effectuons plusieurs fois de suite des appels aux fonctions **STB** :

1. Calcul de la charge apparente sur les signaux : *ctk_calcload*. Cette fonction détermine la charge apparente des portes en déterminant l'effet Miller, à partir des fronts à la sortie des portes pour les agresseurs, et aux extrémités des réseaux RC pour la victime.
2. Calcul des fronts en sortie des portes : *ctk_calcslope*
3. Calcul des fronts aux extrémités des réseaux RC : *ctk_calcrdelay*

Sur les exemples en provenance de la société Bull, en technologie $0,12\mu$, une seule itération est nécessaire pour déterminer les fronts à $1pS$ près. Cette limite est l'unité de base utilisée dans HiTas. La convergence de cette boucle peut ne pas être assurée, et donc osciller indéfiniment entre deux ou plusieurs états. Nous avons donc prévu un nombre maximum d'itérations, et à partir de l'avant dernière itération, les seuls changements de fronts possibles doivent se faire dans le sens conduisant à un meilleur cas pour les fronts minimums, ou un pire cas pour les fronts maximums. C'est la justification du paramètre qui indique si les valeurs doivent être calculées de manière impérative ou de manière conditionnelle dans certaines fonctions de la bibliothèque CTK.

Une fois les charges apparentes et les fronts calculés, le calcul des temps de propagation à travers les portes se fait avec la fonction *ctk_calcdelay*, le calcul des délais à travers les interconnexions ayant été fait en même temps que l'appel à la fonction *ctk_calcrdelay*. A partir de ce nouvel état (délais et front), nous pouvons réitérer à nouveau l'algorithme à partir de l'analyse de stabilité, le critère de fin étant qu'on ne détecte plus de nouvelles agressions observables.

6.9 Conclusion.

Les choix de conception de l'outil ont donc été influencés par deux aspects. Le premier est l'origine des différentes bibliothèques, nous obligeant à séparer la construction du graphe de causalité de la boucle de relaxation, et le second par la volonté de séparer autant que possible les différentes bibliothèques et structures de données. Cet outil implémente tous les principes vus dans les précédents chapitres. Au cours de la réalisation de cet outil, nous avons également vu plus en détail les résultats que doit fournir un analyseur de stabilité afin de s'interfacer avec la détection des agressions. Les résultats et l'expérience de ce programme sont présentés dans le chapitre suivant.

Chapitre 7

Résultats

Nous présentons dans ce chapitre les résultats des algorithmes précédemment décrits. Nous disposons pour cela des trois blocs extraits fournis par Bull dont nous avons parlé dans le chapitre 2.6.1. Ces trois circuits dont je pouvais disposer ont été extraits sous deux formats : Le premier contient des résistances et des capacités à la masse, et le second ne contient que des capacités de couplage entre signaux. Pour générer des fichiers qui contiennent à la fois des résistances d'interconnexion et des capacités de couplage, nous avons écrit un programme qui répartit aléatoirement les capacités de couplage des signaux sur les nœuds de leurs réseaux RC, tout en maintenant constante la capacité totale des réseaux en diminuant la valeur des capacités à la masse. Les résultats obtenus nous permettront donc seulement de valider la faisabilité de la méthode présentée, mais pas d'analyser avec précision ces circuits. Ces trois circuits sont des blocs élémentaires d'un circuit plus important, réalisé en technologie $0,12\mu m$. La société Bull nous a fourni en plus de ces trois circuits le fichier de caractérisation pour HiTas.

Notre approche suppose que nous connaissons la description des horloges, et la stabilité des signaux d'entrée, afin de réaliser les analyses de stabilité dans la bibliothèque STB. Si cette information nous a été fournie par Bull, elle n'est pas directement exploitable avec STB car les conventions entre les outils d'Avertec et ceux de Bull ne sont pas les mêmes. Nous avons donc dû également réaliser un convertisseur pour les différents formats de fichier décrivant la stabilité des signaux externes.

Nous avons vu que pour réaliser l'analyse temporelle d'un circuit avec prise en compte des capacités diaphonantes nous avons dû recourir à l'enchaînement de deux programmes : HiTas, qui crée la structure de données, et Ctk, qui réalise les modifications à apporter sur les délais élémentaires. Dans ce chapitre, nous ne nous intéresserons qu'au programme Ctk.

7.1 Taille de la vue CTK

Nous savons que notre approche à plat limite la taille des circuits que nous pouvons analyser. Néanmoins, la vue CTK est une structure de données assez compacte au regard des résultats obtenus dans le tableau 7.1.

		blk2		blk5		dnp3	
Nb. Transistors		(3 296T)		(12 586T)		(11 554T)	
	Taille (octets)	Nombre d'éléments	Octets	Nombre d'éléments	Octets	Nombre d'éléments	Octets
ctksig	36	790	28 440	3 390	122 040	2 061	74 196
ctkagr	40	8 864	354 560	89 951	3 598 040	111 795	4 471 800
ctkelm	12	25 523	306 276	442 131	5 305 572	439 011	5 268 132
Total			689 276		9 025 652		9 814 128

TAB. 7.1 – Taille de la vue CTK.

Nous constatons sur ces exemples qu'avec cette structure de données, la taille des éléments nécessaire pour déterminer la variation de la charge au deuxième ordre et la variation des délais d'Elmore ne dépasse pas 850 octets par transistor. En pratique, nous avons également à manipuler les informations pour calculer la variation des délais sur les liens de type porte (T_0, R, S, C_0, F_0), le graphe de causalité et l'analyse de stabilité. Lors de l'exécution de Ctk, nous obtenons une consommation mémoire globale de 5,01Mo pour blk2, 20,88Mo pour blk5 et 18,68Mo pour dnp3, soit en moyenne 1 700 Octets par transistor. Par rapport à une limite théorique de 4Go, cette approche ne permet donc pas de dépasser 2 millions de transistors. Il est possible que l'abstraction fonctionnelle d'un bloc de cette taille soit très difficile. La limite d'analyse d'un bloc de cette taille est donc limitée par HiTas plutôt que par Ctk.

7.2 Convergence

Comme le montre le tableau 7.2, la convergence des algorithmes est assez rapide, puisque au plus 7 itérations sont nécessaires pour détecter toutes les agressions possibles. Nous constatons sur ces trois blocs que le nombre d'agressions détectées par itération décroît rapidement. Cette décroissance n'est pas surprenante, mais n'a rien d'obligatoire. Nous avons rencontré sur certains exemples une augmentation du nombre d'agressions détectées entre l'itération 2 et 3. Ceci ne dépend que du circuit étudié.

	blk2	blk5	dnp3
	Nombre d'agressions détectées par itération		
1	2 633	10 364	6 522
2	363	827	4 945
3	51	162	2 017
4	5	35	421
5	0	7	63
6		1	18
7		0	0
	Temps total d'exécution		
	14s	7m48s	22m52s

TAB. 7.2 – Nombre d'itérations dans Ctk.

Les temps d'exécution de Ctk sont très satisfaisants. Ces temps correspondent à l'exécution complète du programme Ctk, qui comprend le temps de chargement des différentes vues, les analyses de stabilité et la correction des délais élémentaires. Nous avons constaté qu'une grande partie du temps d'exécution dans les itérations était la détection des agressions. La correction des délais élémentaires est très rapide une fois que les agresseurs sont connus, et l'analyse de stabilité occupe un temps très court. Bien que pour cette catégorie de programme qui ne requiert pas d'interactivité, le temps d'exécution ne soit pas un critère très significatif, une amélioration à ce niveau peut être envisagée pour ne détecter les agressions et calculer les délais que si la stabilité d'un signal et de ses agresseurs a évolué. En tout cas, ces exemples confirment que notre approche permet de résoudre de manière satisfaisante le problème posé par les capacités de couplage. L'ordinateur utilisé pour réaliser ces mesures est une Sparc Enterprise 480, à 450Mhz. Enfin, dans HiTas, la construction de la vue Ctk n'augmente pas de manière significative le temps d'exécution du programme.

7.3 Observations réalisées sur ces trois circuits

7.3.1 Nombre d'agresseurs

Nous allons maintenant étudier quelles informations nous avons été capables d'obtenir lors de l'exécution de Ctk. Commençons par le nombre d'agresseurs. Nous avons reporté dans le

tableau 7.3 le nombre d'agresseurs total par circuit. Ce chiffre correspond au nombre de **ctkagr** du tableau 7.1. Rappelons qu'entre deux signaux S1 et S2, il peut y avoir n capacités de cou-

	blk2	blk5	dnp3
Agresseurs	8 864	89 951	111 795
Dont observables	5 074 (57,2%)	69 963 (77,8%)	36 697 (32,8%)
Modification T _{pmax}	3 347	46 136	31 105
Modification T _{pmin}	4 226	54 997	34 222

TAB. 7.3 – Nombre d'agresseurs actifs.

plage. Ces n capacités donnent 2 agresseurs, un de S1 pour S2, et l'autre de S2 pour S1. Chaque agresseur peut modifier le temps de propagation minimum et maximum du signal.

Parmi ces agresseurs, nous avons donc reporté ceux qui étaient observables, c'est à dire qui ont contribué à modifier un délai T_{pmax} ou T_{pmin}. Nous constatons sur ces trois exemples qu'il peut y avoir une grande variation de la part des agresseurs observables. Cela conforte notre hypothèse de considérer au cas par cas l'influence de chaque agresseur, au lieu d'aborder une approche meilleur cas systématique (qui ne considère aucune influence), ou pire cas systématique (qui considère toutes les influences possibles). Nous avons également reporté le nombre d'agressions qui ont conduit à une modification d'un temps de propagation maximum ou minimum.

7.3.2 Importance des agressions

Nous avons reporté pour chaque bloc dans le tableau 7.4 le signal soumis à l'agression la plus importante sur T_{pmax} et T_{pmin}. Le critère retenu ici pour déterminer les signaux les plus agressés est l'observation de la variation de la capacité de charge équivalente au 1er ordre. Dans ce tableau, nous avons relevé cette variation de charge calculée par Ctk dans les lignes "Variation de charge au 1er ordre". Cette variation de charge est due à l'agression de n agresseurs observables parmi les m agresseurs potentiels que comporte ce signal. Ces valeurs n/m sont reportées dans les lignes "Nombre d'agresseurs observables". Enfin, nous avons reporté la variation de charge que nous aurions obtenue si nous avions fait l'hypothèse sur ce signal que tous ses agresseurs étaient observables dans la ligne "Variation de charge au 1er ordre lorsqu'on considère tous les agresseurs observables". Il est intéressant de comparer cette dernière

	blk2	blk5	dnp3
Agression sur Tpmax			
Nom du signal	DW ndp_rdca_l2_d[37]	DW ndp_zmul_x1[28]_tas_1	UP p1054_t
Variation de charge au 1er ordre.	+18,9%	+31,3%	+36,9%
Nombre d'agresseurs observables	3/5	209/224	74/136
Variation de charge au 1er ordre lorsque on considère tous les agresseurs observables	+19,2%	+35,7%	+65,7%
Agression sur Tpmin			
Nom du signal	UP pks2_170	UP ndp_rppq_l1[21]_tas_1	UP p1174_t
Variation de charge au 1er ordre.	-29,0%	-34,9%	-41,4%
Nombre d'agresseurs observables	13/25	27/53	114/226
Variation de charge au 1er ordre lorsque on considère tous les agresseurs observables	-36,9%	-41,1%	-67,6%

TAB. 7.4 – Agressions observables les plus sévères.

information avec les valeurs relevées dans le tableau 7.5. Dans ce tableau, nous avons extrait les signaux dont la capacité équivalente de charge auraient le plus varié si nous avons considéré tous les agresseurs comme observables. Nous avons comparé cette valeur avec la valeur trouvée par Ctk. Dans ces trois blocs, la notion d'agression observable est flagrante, puisque les signaux dont on aurait pu croire qu'ils étaient sensibles par rapport à la valeur des capacités de couplage ne sont finalement pas trop influencés par leur entourage. De même, le nombre d'agresseurs n'est absolument pas un critère pour déterminer la sensibilité d'un signal. Ceci confirme que la prise en compte de l'activité des agresseurs est une information déterminante pour une estimation réaliste des délais.

	blk2	blk5	dnp3
Nom du signal	DW pks2_l32	DW ndp_radd_l1[31]	DW p1259_t
Variation de charge au 1er ordre lorsque on considère tous les agresseurs observables	+36,9%	+46,0%	+73,3%
Variation de charge au 1er ordre.	00,0%	+14,2%	-29,0%

TAB. 7.5 – Agressions les plus sévères en considérant que tous les agresseurs sont observables.

7.4 Influence temporelle

Nous avons relevé dans le tableau 7.6 les modifications des délais élémentaires les plus significatives en valeur absolue pour ces trois blocs. Nous constatons que des variations impor-

	blk2	blk5	dnp3
Agression sur T _{pmax}			
De	UP nc1_ovp_data_x1[2]_tas_1	UP n1881_tas_1	UP n1548_tas_1
A	DW n235	DW rm01_io_busa_p[8]	DW p1194_t
ΔT	+7	+27	+28
Agression sur T _{pmin}			
De	UP n626	UP rm01_r15a_tas_21	DW nc1_r1tno_1x1n
A	UP n626_tas_18	DW rm01_io_busa_p[10]	DW nc1_r1tno_1x1n_tas_31
ΔT	-12	-37	-142

TAB. 7.6 – Modifications les plus importantes (en pS) des délais élémentaires.

tantes sur certains délais élémentaires se produisent. Les variations sur les délais élémentaires se reportent inégalement sur les chaînes les plus longues, que nous avons reporté dans le tableau 7.7. Sur ces trois circuits, la chaîne la plus longue reste la même, seuls les délais changent.

blk2	blk5	dnp3
+6	+33	+11

TAB. 7.7 – Modification des chemins critiques (en pS).

Par contre, lorsqu'on observe les 10 chaînes les plus longues, l'ordre de celles-ci varie. L'influence des capacités de couplage se fait finalement peu ressentir.

7.5 Détermination de la charge apparente au deuxième ordre

Nous savons que le modèle utilisé dans ce prototype pour évaluer la variation de la charge au deuxième ordre lorsque les agresseurs observables sont connus peut ne pas fonctionner dans certains cas. Nous avons donc compté le nombre de cas pour lesquels un modèle de charge équivalente au 1er ordre a dû être utilisé. Ce résultat est donné dans le tableau 7.8. Nous constatons que dans de nombreux cas cette approche a donné de bons résultats, puis-

blk2	blk5	dnp3
5/6 592	75/25 172	98/23 108
0,08%	0,30%	0,42%

TAB. 7.8 – Nombre d'évaluation de charge au 1er ordre.

qu'elle a fonctionné dans plus de 99% des cas. Néanmoins, dans quelques cas critiques où les couplages sont trop importants, il a fallu recourir à un modèle du premier ordre, comme utilisé actuellement dans HiTas.

7.6 Conclusion

Les résultats obtenus avec ce prototype logiciel nous ont permis de tirer les conclusions suivantes :

- Tout d'abord, le nombre d'agressions décroît très rapidement, et reste limité. Cela nous montre que les solutions basées sur l'analyse de stabilité sont utilisables.
- Ensuite, la méthode de correction des délais élémentaires avec la structure **CTK** permet d'obtenir une bonne précision dans de nombreux cas, et surtout un gain en temps de calcul et en espace mémoire extrêmement important.
- Enfin, le temps de calcul nécessaire pour détecter les agresseurs est dans notre prototype largement plus grand que le temps nécessaire pour recalculer les délais. Ce temps de calcul étant, sur ces trois exemples, inférieur à la seconde pour la totalité d'exécution de **CTK**, il est sans doute envisageable d'utiliser un modèle plus précis.

Bien évidemment, entre le modèle de calcul des délais élémentaire, la méthodologie de détection des agressions observables et l'écriture du programme, de nombreux allers et retours ont

eu lieu, mais n'ont pas remis fondamentalement en cause notre approche initiale. Ils nous ont au contraire permis de l'affiner sensiblement.

Chapitre 8

Conclusion et perspectives

Nous nous étions fixé comme objectif de réaliser l'analyse temporelle de circuits en prenant en compte simultanément les résistances d'interconnexion et les capacités de couplage. Pour cela, nous disposions d'un analyseur temporel, HiTas, et d'un analyseur de stabilité, STB. Au cours de cette étude, nous avons développé :

- Un modèle électrique présentant un compromis satisfaisant en terme de précision, vitesse et consommation mémoire,
- Un ensemble d'algorithmes pour utiliser et interpréter les résultats fournis par un analyseur de stabilité,
- Un ensemble d'algorithmes pour déterminer les agressions réelles, grâce à une méthode itérative,
- Et un prototype logiciel mettant en œuvre tous ces principes.

Concernant les modèles électriques, la grande amélioration dans HiTas a été de représenter la charge par une cellule en π , possibilité que n'offre aujourd'hui aucun outil industriel. La solution de procéder par correction des délais de porte et d'interconnexion est satisfaisante dans les cas que nous avons traités. Deux améliorations sont néanmoins envisageables dans le futur. Tout d'abord recalculer les temps de propagation des portes à l'aide du simulateur algébrique décrit par les équations 4.37. Et ensuite ne pas procéder à la modification de la charge apparente, mais de la recalculer systématiquement à partir de la description complète du réseau RC. Ces deux améliorations permettront d'améliorer légèrement la précision des résultats si cela s'avère utile, au prix d'une consommation mémoire beaucoup plus grande.

Concernant l'aspect méthodologie, la prise en compte du comportement individuel de chaque agresseur est une approche tout à fait réaliste. Le nombre d'itérations est raisonnablement faible, ce qui n'entraîne pas un temps d'exécution intolérable. La solution que nous avons utilisée de détecter progressivement les agressions possibles permet d'introduire le critère d'agres-

sion observable. La détection progressive des agressions et le critère d'observabilité permettent d'obtenir des résultats moins pessimistes que l'approche d'élimination progressive des agressions. Si ces deux approches sont théoriquement très différentes, dans le cas d'une implémentation logicielle elles sont très proches.

Dans les perspectives à l'issue de cette thèse, plusieurs points restent à étudier. Nous avons soulevé les avantages et inconvénients liés aux approches mono et multi-intervalles, dans l'objectif de ne pas exagérer la détection des agressions. La solution idéale serait d'avoir un moyen de déterminer quels sont les signaux pour lesquels une approche mono-intervalle est préférable, et quels sont ceux pour lesquels une approche multi-intervalle est préférable. Ce critère pouvant bien sûr être différent selon l'agresseur considéré d'un signal. Toujours dans l'objectif de ne déterminer que les agressions réelles, il peut être utile d'introduire une notion de fonctionnalité afin de déterminer si deux signaux peuvent commuter ensemble. Ensuite, nous n'avons fait qu'effleurer le problème posé par la taille des circuits à considérer. Le problème est fondamentalement à plat, il n'y a pas d'arrêt de l'influence des couplages aux interfaces, ni sur les signaux connectés à l'extérieur d'un bloc. Seules des approches mettant en place des mécanismes de mémoire cache devraient permettre de résoudre le problème sur des circuits de plusieurs millions de transistors. En supposant que l'on sache réaliser l'analyse temporelle d'un circuit complexe, il faut ensuite mener une réflexion sur la conduite à tenir en cas de non respect des contraintes temporelles d'un circuit à cause de la diaphonie. Faut-il refaire une opération complète de placement et de routage, modifier certaines portes pour les rendre moins sensible aux agressions (et donc rendre plus sensibles ses agresseurs), ou dévier certains fils ? Cette dernière opération, la plus intuitive, nécessite de mémoriser la relation entre le signal physique et sa représentation électrique. Enfin, pour réduire le temps d'exécution du logiciel, des mécanismes de filtrages peuvent être envisagés pour ne traiter la diaphonie que sur les signaux sensibles.

A plus long terme, d'autres phénomènes seront à prendre en compte dans l'analyse temporelle : chute de tension dans les alimentations [BBNH01], couplages par le substrat, couplages inductifs et effets de ligne, le tout sur des circuits de plus en plus gros, puisque le milliard de transistors sur une seule puce est annoncé pour 2010. Comme pour ces dernières décennies, des phénomènes de plus en plus complexes à modéliser apparaissent sur des circuits de plus en plus gros.

Annexes

Annexe A

Structures de données CTK

Nous présentons ici les principales structures de données utilisées dans l'outil CTK.

A.1 La structure CTK qui encapsule la vue TTV :

```
typedef struct ctkview {  
    struct ctksig *SIGNAL;  
    ttvfig_list *TTVFIG;  
} ctkview_list ;
```

- SIGNAL : Liste des signaux sur lesquels on souhaite pouvoir prendre en compte les effets de couplage.
- TTVFIG : La vue temporelle que l'on souhaite modifier.

A.2 La structure CTK qui encapsule les signaux de la vue TTV :

```
typedef struct ctksig {  
    struct ctksig *NEXT ;  
    ttvsig_list *SIGNAL ;  
    float Y1, Y2, Y3 ;  
    float RUP, RDW ;  
    struct ctkagr *AGR;  
} ctksig_list ;
```

- NEXT : Signal suivant.
- SIGNAL : Le signal TTV associé.

- Y1, Y2, Y3 : La charge au deuxième ordre sans couplages actifs.
- RUP, RDW : La résistance en petit signaux vers VDD et VSS de la porte, pour le calcul du bruit.
- AGR : La liste des agresseurs.

A.3 La structure CTK représentant un agresseur :

```
typedef struct ctkagr {
    struct ctkagr *NEXT;
    ttvsig_list  *SIGNAL ;
    char          ACTIF ;
    float         X11 ;
    float         X12, X22 ;
    float         X13, X23, X33 ;
    struct ctkelm *ELM ;
} ctkagr_list ;
```

- NEXT : L'agresseur suivant.
- SIGNAL : Le signal TTV agresseur.
- ACTIF : Détermine le type d'agression selon la transition :
 - CTK_UPWRSTAGR : Cet agresseur agit comme un pire cas (transition descendante) pour l'évènement correspondant à une transition montante du signal victime.
 - CTK_DWWRSTAGR : Cet agresseur agit comme un pire cas (transition montante) pour l'évènement correspondant à une transition descendante du signal victime.
 - CTK_UPBESTAGR : Cet agresseur agit comme un meilleur cas (transition montante) pour l'évènement correspondant à une transition montante du signal victime.
 - CTK_DWBESTAGR : Cet agresseur agit comme un meilleur cas (transition descendante) pour l'évènement correspondant à une transition descendante du signal victime.
- X11, X12, X22, X13, X23, X33 : Les coefficients des équations 4.66.
- ELM : La liste des délais d'Elmore à modifier.

A.4 La structure CTK représentant la modification d'un délai d'Elmore :

```
typedef struct ctkelm
```

```
struct ctkelm *NEXT ;
ttvsig_list  *SIGNAL ;
float        K ;
ctkelm_list ;
```

- NEXT : Le délai d'Elmore à modifier suivant.
- SIGNAL : Le signal TTV à modifier.
- K : La constante de l'équation 4.61.

A.5 La structure CTK contenant les informations électriques nécessaires pour calculer les modifications des temps de propagation sur les portes :

Cette structure ne dépend pas de CTK, mais du modèle MCC. Elle est placée dans le champ USER des `ttvline_list`.

```
typedef struct s_ctkcaracline {

    float  TOMIN;
    float  COMIN;
    float  FOMIN;
    float  RMIN;
    float  SMIN;
    float  IMIN;
    float  RNMIN;
    float  FOMINOUT;

    float  TOMAX;
    float  COMAX;
    float  FOMAX;
    float  RMAX;
    float  SMAX;
    float  IMAX;
    float  RNMAX;
    float  FOMAXOUT;

} ctkcaracline ;
```

Les informations électriques sont déterminées pour permettre le calcul des temps de propagation minimum et maximum.

- T_{0MIN}, T_{0MAX} : Délai initial évalué par Tas.
- C_{0MIN}, C_{0MAX} : Capacité de routage Y1 sans conflit.
- F_{0MIN}, F_{0MAX} : Front d'entrée sur le lien sans conflit.
- R_{MIN}, R_{MAX} : Variation linéaire de T avec C.
- S_{MIN}, S_{MAX} : Variation linéaire de T avec F.
- I_{MIN}, I_{MAX} : Courant constant dans la porte.
- R_{NMIN}, R_{NMAX} : Détermination du front de sortie proportionnel à C.
- F_{0MINOUT}, F_{0MAXOUT} : Front de sortie sur le lien sans conflit.

Annexe B

Quelques fonctions de la bibliothèque CTK

B.1 La fonction *ctk_calcload*

B.1.1 Arguments :

Cette fonction prend deux arguments :

- La vue CTK.
- Un paramètre qui indique si l'on doit prendre en compte le modèle basique 0C/2C pour l'effet Miller ou les relations plus détaillées présentées dans le chapitre 4.2.2.

Ce dernier argument ne sert que pour la mise au point du programme. En usage normal, les relations détaillées sont toujours utilisées.

B.1.2 Description :

Cette fonction calcule la charge apparente au deuxième ordre correspondant à réseau RC agressé par d'autres réseaux, d'après les équations présentées dans le chapitre 4.4. Les charges apparentes, calculées sous la forme y_1 , y_2 et y_3 minimum et maximum, sont placées dans les nœuds de type porte.

B.2 La fonction *ctk_calcslope*

Arguments :

Cette fonction prend deux arguments :

- La vue CTK

- Un paramètre qui indique comment sont déterminés les fronts : de manière impérative ou de manière conditionnelle. La manière impérative signifie que le résultat du calcul du front est systématiquement pris en compte. Dans la manière conditionnelle, le calcul du front n'est pris en compte que si sa valeur va au-delà de la valeur précédente. C'est à dire que lorsqu'on cherche à calculer un front maximum, on ne prend en compte la valeur du front recalculé que si elle est effectivement supérieure au front existant, et réciproquement pour les fronts minimums.

B.2.1 Description :

Calcul les fronts sur tous les arcs de type porte, à partir de la charge équivalente déterminée sur les nœuds de type déterminé au moyen de la fonction *ctk_calclload*. La charge étant fournie sous la forme y_1 , y_2 et y_3 , cette fonction transforme cette charge en calculant soit une cellule en π , soit une simple capacité. Cette seconde éventualité se produit lorsque la linéarisation de la variation de charge présentée dans le chapitre 4.4 n'a pas fonctionné. Dans le cas d'une charge en π , cette charge est ramenée à une simple capacité pour rester compatible avec le modèle MCC, avec la méthode présentée dans le chapitre 4.1.

B.3 La fonction *ctk_calcrdelay*

Arguments :

Cette fonction prend deux arguments :

- La vue CTK
- Un paramètre qui indique comment sont déterminés les temps et les fronts : de manière impérative ou de manière conditionnelle.

B.3.1 Description :

Cette fonction calcule à la fois les temps de propagation et les fronts sur les arcs RC, en prenant en compte les agresseurs dans les mêmes conditions que pour la fonction *ctk_calclload*, et le front d'entrée calculés précédemment par avec la fonction *ctk_calcslope* sur les nœuds de types porte.

B.4 La fonction *ctk_calcdelay*

B.4.1 Arguments :

Cette fonction prend les mêmes arguments que la fonction *ctk_calcslope*.

B.4.2 Description :

Cette fonction calcule le temps de propagation sur tous les arcs de type porte, à partir des fronts d'entrée et de la charge équivalente calculée par la fonction *ctk_calcload*. La charge équivalente est déterminée dans les mêmes conditions que la fonction *ctk_calcslope*.

Annexe C

Formats de fichier

Les fichiers que nous présentons ici sont au format ASCII, c'est à dire au format texte, lisible par n'importe quel éditeur ou traitement de texte. Les fichiers au format texte sont bien plus volumineux que les fichiers au format binaire, et une perte de précision dans la représentation des nombres flottants n'est pas impossible. Concernant la taille des fichiers, nous avons introduit dans la bibliothèque **MBK** la possibilité d'appeler à la volée n'importe quel logiciel de compression et de décompression (gzip), de manière transparente. Concernant la perte de précision des nombres flottants, elle est liée au changement de base entre le format binaire et le format décimal dans les deux sens, à l'écriture du fichier et à sa relecture. Cette perte de précision n'est pas significative dans notre domaine. Par contre, on y gagne en portabilité, car tous les systèmes informatiques n'utilisent pas la même norme pour représenter les nombres flottants ou les nombre entiers (*little endian* ou *big endian*). Le format texte permet d'éviter ce problème.

C.1 Le fichier **ctk**

Ce fichier contient la représentation de la structure de données **CTK** sur disque. Les lignes commençant par un dièse sont ignorées : elles servent de commentaire. L'en tête de ce fichier commence toujours par :

```
CTK 1.0
```

Cette information sert à différencier les éventuelles versions de la structure de données **CTK**. La dernière ligne lue dans le fichier contient :

```
EOF
```

Entre les deux, on trouve la description des signaux. La description d'un signal commence par le mot clé **SIGNAL** suivi du nom du signal, et se termine par une ligne blanche. Nous avons extrait une partie d'un fichier pour l'exemple :

```
SIGNAL nc1_rczau_l1[3]
Y 1.783044e-02 -9.315811e-03 6.043673e-03
R 0.000000e+00 0.000000e+00
AGR nc1_rczau_l1[7]
X 2.721000e-04 -3.926776e-04 -1.290554e-05 4.050727e-04 2.496545e-05 6.121019e-07
ELM nc1_rczau_l1[3]_tas_1
K 4.742939e-02
ELM nc1_rczau_l1[3]_tas_2
K 3.169320e-03
ELM nc1_rczau_l1[3]_tas_3
K 2.192582e-03
AGR n175
X 1.055000e-04 -1.150823e-04 -1.091229e-06 1.071488e-04 1.589832e-06 1.128701e-08
ELM nc1_rczau_l1[3]_tas_1
K 4.829654e-03
ELM nc1_rczau_l1[3]_tas_2
K 1.228825e-03
ELM nc1_rczau_l1[3]_tas_3
K 8.501189e-04
```

Pour ce signal `nc1_rczau_l1[3]`, nous trouvons la charge équivalente au deuxième ordre, dans la ligne commençant par **Y**, sous la forme **Y Y1 Y2 Y3**. Ensuite, nous avons la résistance en petits signaux de cette porte dans la ligne commençant par **R**, sous la forme **R RUP RDOWN**. Ensuite vient la liste de tous les agresseurs. La définition d'un agresseur apparaît sous la forme **AGR**, suivi du nom du signal agresseur. Associé à cet agresseur, on trouve la liste des constantes qui servent à calculer la variation de charge apparente due à l'agresseur sous la forme **X X11 X12 X22 X13 X23 X33**. Ensuite, on trouve la liste des délais RC que cet agresseur va modifier. Chaque délai RC apparaît sous la forme **ELM** suivit du nom du délai RC, et sur la ligne d'après la constante servant à modifier le délai d'Elmore. Cette constante apparaît sous la forme **K**, suivit de la constante.

C.2 Le fichier car

C'est le fichier qui contient la caractérisation des arcs de type porte, pour recalculer les délais lorsqu'ils seront soumis à l'influence d'agresseurs observables. La syntaxe de ce fichier est très simple. Il y a une caractérisation par ligne, la dernière ligne lue du fichier étant EOF.

Le premier élément de la ligne est le type de transition dans la porte : UU, DU, UD ou DD. Les deux suivants sont les nœuds source et destination qui servent à identifier l'arc de type porte. le nœud source est forcément un nœud de type porte, et le nœud destination est forcément un nœud de type porte. Les quatre termes suivant sont RMIN, SMIN, RMAX et SMAX. Nous avons extrait une partie d'un fichier pour l'exemple :

```
UD n46_tas_1 ndp_raln_m[6] 1366 1366 457 457
DU n47_tas_1 ndp_raln_m[5] 2202 2202 396 396
UD n47_tas_1 ndp_raln_m[5] 1366 1366 457 457
DU n35_tas_1 ndp_raln_m[5] 2202 2202 440 447
UD n35_tas_1 ndp_raln_m[5] 1366 1366 460 461
DU n92_tas_1 ndp_raln_m[4] 2202 2202 433 444
UD n92_tas_1 ndp_raln_m[4] 1366 1366 459 460
DU n48_tas_1 ndp_raln_m[4] 2202 2202 396 403
UD n48_tas_1 ndp_raln_m[4] 1366 1366 457 457
DU n61_tas_1 ndp_raln_m[3] 2202 2202 396 396
UD n61_tas_1 ndp_raln_m[3] 1366 1366 457 457
DU n81_tas_1 ndp_raln_m[3] 2202 2202 378 378
UD n81_tas_1 ndp_raln_m[3] 1366 1366 454 454
DU n82_tas_1 ndp_raln_m[2] 2202 2202 378 378
EOF
```


Annexe D

Glossaire

Terme	Signification
Alliance	Chaîne de CAO utilisée au LIP6.
ASCII	American Standard Code for Information Interchange. Norme de codage des caractères textuels.
ASIM	Architecture des Systèmes intégrés et Micro électronique.
AWE	Asymptotic Waveform Evaluation. Algorithme permettant d'obtenir une représentation en pôles et en zéros d'un circuit électrique.
DSPF	Detailed SPF (page 69). Voir SPF.
Bleeder	Porte utilisée pour rétablir le potentiel électrique d'un signal dont le niveau est dégradé.
CMOS	Complementary Metal-Oxide Semiconductor. Technologie de porte utilisant des réseaux de transistors symétriques.
HiTas	Analyseur temporel hiérarchique développé au LIP6.
LIP6	Laboratoire d'Informatique de Paris 6.
MBK	Structure de donnée commune à tous les outils de CAO développé au LIP6.
MCC	MOS Canal Court. Equations décrivant le courant dans un transistor.
RC	Terme générique décrivant une interconnection composée de résistances flottantes par rapport à la masse, des capacités à la masse et des capacités de couplage.
RSPF	Reduced SPF (page 69). Voir SPF.
SDF	Standard Delay Format.

Terme	Signification
SLS	Switch Level Simulator (page 72). Simulateur logico-temporel de l'ensemble de logiciels Space.
SPEF	Standard Parasitics Exchange Format (page 69). Format de fichier normalisé par IEEE1481-1999 clause 9 (http://www.si2.org/ieee1481/spef_pdef).
SPF	Standard Parasitic Format (page 69). Format de description de circuit extrait, proposé par Cadence [Cad93].
TTV	Tas Timing View. Structures de données représentant un graphe de causalité dans HiTas.
Yagle	Abstracteur fonctionnel développé au LIP6.

Annexe E

Références en ligne

Un certain nombre de références citées dans ce manuscrit ne sont maintenant disponibles qu'en ligne. Leur contenu étant susceptible de disparaître ou d'être modifié, je les ai regroupé à l'adresse <http://www-asim.lip6.fr/~gregoire/these>.

Référence	Nom du document
[Int01]	Datasheet P4
[Pay98]	Dynamic timing analysis for system on chip design
[Cad01b]	Celtic : Crosstalk analyzer for cell-based desings
[Cad01c]	Noise - Aware timing Analysis
[Syn97]	Synopsys Prime Time datasheet
[Syn01b]	Prime Time SI : Static Crosstalk Analysis
[Syn01c]	Static Crosstalk Analysis
[Men97]	Interconnect parasitic extraction for deep submicron IC design
[Cad01a]	Assura Parasitic Extractor
[Cad]	Dracula, the physical verification standard
[Syn01a]	Arcadia : RC extraction for timing, power, reliability and signal integrity analysis
[Ava00]	Datasheet : Star RCXT
[Seq01a]	ExtractionStage
[Seq01b]	ShowTime

Annexe F

Détails des calculs

F.1 Le délai d'Elmore

Rappelons brièvement la théorie de ce calcul, en commençant par quelques rappels élémentaires de la théorie du signal et de l'analyse des arbres RC. La réponse indicielle $e(t)$ d'un filtre est égale à l'intégrale de la réponse impulsionnelle $e'(t)$ de ce filtre :

$$e(t) = \int_0^t e'(t).dt \quad (\text{F.1})$$

Une conséquence directe de ce point est, la réponse indicielle valant l'unité au bout d'un temps infini :

$$\int_0^{+\infty} e'(t).dt = 1 \quad (\text{F.2})$$

La fonction de transfert $H(s)$ d'un filtre est égale à la transformée de Laplace de sa réponse impulsionnelle.

$$H(s) = \int_0^{+\infty} e'(t).e^{-s.t}.dt \quad (\text{F.3})$$

Un arbre RC peut s'écrire sous la forme d'une fonction de transfert :

$$H(s) = \frac{1 + a_1.s + a_2.s^2 + \dots + a_n.s^n}{1 + b_1.s + b_2.s^2 + \dots + b_m.s^m} \quad (\text{F.4})$$

Où les termes a_i et b_i sont réels et $m > n$, et les pôles de cette fonction sont réels et négatifs. Considérons les réponses impulsionnelles et indicielles de la figure F.1. Le délai d'Elmore T_D s'exprime à partir de $e'(t)$ avec la relation :

$$e(T_D) = \int_0^{T_D} e'(t).dt = \frac{1}{2} \quad (\text{F.5})$$

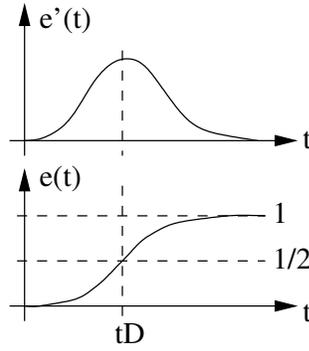


FIG. F.1 – Définition des délais.

Cette approche directe pour déterminer la valeur de T_D n'est pas évidente, puisque nous ne disposons pas de ces réponses. Elmore contourne le problème en supposant que la réponse impulsionnelle est symétrique. Compte tenu de l'équation F.2, il est possible d'appliquer un résultat du domaine des probabilités : Dans le cas d'une répartition modale et symétrique, la médiane et l'espérance mathématique sont identiques. Cela permet donc d'écrire la relation :

$$T_d \approx \int_0^{+\infty} t \cdot e'(t) \cdot dt \quad (\text{F.6})$$

Le passage de la définition du filtre de l'équation F.4 et la définition du délai d'Elmore de l'équation F.6 passe par la décomposition en série de Taylor de l'équation F.3. La décomposition de $e^{-s \cdot t}$ autour de $s = 0$ est :

$$e^{-s \cdot t} = \sum_{n=0}^{+\infty} (-1)^n \cdot \frac{(s \cdot t)^n}{n!} \quad (\text{F.7})$$

En remplaçant ce terme dans l'équation F.3, nous obtenons :

$$H(s) = \int_0^{+\infty} e'(t) \cdot \sum_{n=0}^{+\infty} (-1)^n \cdot \frac{(s \cdot t)^n}{n!} \cdot dt \quad (\text{F.8})$$

Que l'on peut réécrire :

$$H(s) = \sum_{n=0}^{+\infty} \left((-1)^n \cdot \frac{s^n}{n!} \cdot \int_0^{+\infty} e'(t) \cdot t^n \cdot dt \right) \quad (\text{F.9})$$

Finalement, nous obtenons la décomposition de $H(s)$ sous la forme d'une somme de moments :

$$\begin{cases} H(s) = \sum_{n=0}^{+\infty} m_n \cdot s^n \\ m_n = \frac{-1^n}{n!} \cdot \int_0^{+\infty} t^n \cdot e'(t) \cdot dt \end{cases} \quad (\text{F.10})$$

En particulier :

$$m_1 = - \int_0^{+\infty} t.e'(t).dt \quad (\text{F.11})$$

Nous constatons que le premier moment, m_1 , correspond au délai d'Elmore changé de signe. Il ne nous reste qu'à rapprocher ce résultat avec l'équation F.4, en décomposant cette dernière sous la forme d'une série de Taylor, en nous intéressant au premier moment de cette série :

$$\begin{aligned} H(s) &= \sum_{i=0}^{\infty} \frac{1}{i!} H^{(i)}(0) s^i \\ &= 1 + H'(0).s + \sum_{i=2}^{\infty} \frac{1}{i!} H^{(i)}(0) s^i \end{aligned} \quad (\text{F.12})$$

Le calcul de $H'(s)$ pour $s = 0$ est immédiat :

$$H'(s) = \left(\frac{u(p)}{v(p)} \right)' = \frac{u'(p).v(p) - v'(p).u(p)}{v(p)^2} \quad (\text{F.13})$$

Avec

$$\begin{aligned} u(p) &= 1 + \sum_{i=1}^n a_i . s^i \\ u'(p) &= a_1 + \sum_{i=2}^n a_i . s^{i-1} \\ v(p) &= 1 + \sum_{i=1}^m b_i . s^i \\ v'(p) &= b_1 + \sum_{i=2}^m b_i . s^{i-1} \end{aligned} \quad (\text{F.14})$$

Le terme constant de $u'(p).v(p)$ vaut a_1 et le terme constant de $v'(p).u(p)$ vaut b_1 . Nous avons donc $H'(0) = a_1 - b_1$. Le délai d'Elmore correspondant au filtre de l'équation F.4 vaut donc $T_D = b_1 - a_1$. Il est facile de constater que cela correspond bien à un retard, donc un temps de propagation, en partant de la définition de la fonction retard. La transformée de Laplace d'un filtre qui introduit un retard sans déformation d'un signal s'écrit :

$$\begin{aligned} L \{ f(t - T) \} &= \int_0^{\infty} f(t - T).e^{-s.t}.dt \\ &= e^{-s.T}.F(s) \end{aligned} \quad (\text{F.15})$$

La fonction retard dans le domaine temporel se traduit par une multiplication par $e^{-s.T}$ dans le domaine fréquentiel. A partir de la fonction retard :

$$R(s) = e^{-s.T} \quad (\text{F.16})$$

Nous déterminons le retard introduit par cette fonction avec :

$$T = - R'(s) \Big|_{s=0} \quad (\text{F.17})$$

Nous étendons cette notion de retard pour un filtre quelconque $H(p)$ avec la relation :

$$T_D \approx -H'(0) \tag{F.18}$$

En résumé, nous avons rappelé que le délai d’Elmore correspond à un retard, et qu’il vaut $b_1 - a_1$. Nous verrons dans la présentation de l’algorithme AWE comment calculer pour un arbre RC la valeur de ce premier moment.

F.2 Calcul des n premiers moments sur les nœuds d’un arbre RC

Si l’on restreint l’ensemble des interconnexions que l’on souhaite analyser aux arbres RC avec un seul émetteur, le calcul des $2n - 1$ premiers moments devient extrêmement rapide puisqu’il se ramène à appliquer une équation de récurrence sur les nœuds de l’arbre. Considérons le schéma de la figure F.2. A chaque nœud i correspond une résistance R_i en amont de ce

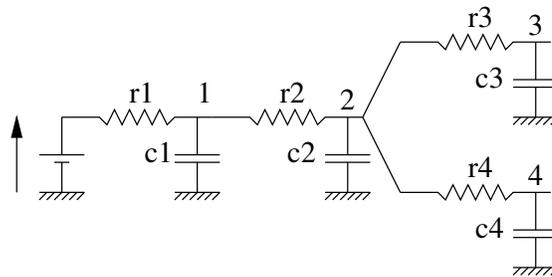


FIG. F.2 – Calcul des moments sur les nœuds d’un arbre RC

nœud et une capacité C_i reliée à la masse. Nous avons vu que la transformée de Laplace de la réponse impulsionnelle sur chaque nœud est la fonction de transfert réalisée par ce filtre. De plus, nous savons d’après l’équation F.10 que cette réponse peut s’écrire sous la forme :

$$H(s) = m_0 + m_1.s + m_2.s^2 + m_3.s^3 + \dots \tag{F.19}$$

La tension sur chaque nœud i s’écrit donc :

$$V_i(s) = m_{0,i} + m_{1,i}.s + m_{2,i}.s^2 + m_{3,i}.s^3 + \dots \tag{F.20}$$

Le terme $m_{n,i}$ désignant le coefficient d’ordre n sur le nœud i . Le courant qui circule dans chaque capacité C_i vaut :

$$i_i(t) = C_i \cdot \frac{dv_i(t)}{dt} \tag{F.21}$$

La transformée de Laplace de ce courant est, les conditions initiales étant nulles :

$$\begin{aligned} I_i(s) &= C_i \cdot s \cdot V_i(s) \\ &= C_i \cdot (m_{0,i} \cdot s + m_{1,i} \cdot s^2 + m_{2,i} \cdot s^3 + \dots) \end{aligned} \quad (\text{F.22})$$

La tension sur chaque nœud du circuit est égale à la tension d'entrée à laquelle on retire la chute de tension dans chaque résistance. C'est là que la caractéristique de l'arbre RC intervient, puisque le courant qui circule dans chaque résistance R_i est égale à la somme des courants dans toutes les capacités en aval du nœud i . Ecrivons cette equation pour un nœud i , en prenant les moments jusqu'à l'ordre q :

$$\sum_{j=0}^q (m_{j,i} \cdot s^j) = 1 - \sum_{k \in \delta_i} \left(R_k \sum_{l \in \zeta_k} \left(C_l \cdot \sum_{m=0}^{q-1} (m_{m,l} \cdot s^{m+1}) \right) \right) \quad (\text{F.23})$$

Où δ_i représente l'ensemble des résistances entre l'émetteur et le nœud i , et ζ_k représente l'ensemble des nœuds en aval du nœud k , nœud k compris. On détermine chaque moment en identifiant les puissances de s :

$$\begin{aligned} 0 : m_{0,i} &= 1 \\ 1 : m_{1,i} &= - \sum_{k \in \delta_i} \left(R_k \sum_{l \in \zeta_k} (C_l \cdot m_{0,l}) \right) \\ 2 : m_{2,i} &= - \sum_{k \in \delta_i} \left(R_k \sum_{l \in \zeta_k} (C_l \cdot m_{1,l}) \right) \\ \dots & \\ n : m_{q,i} &= - \sum_{k \in \delta_i} \left(R_k \sum_{l \in \zeta_k} (C_l \cdot m_{q-1,l}) \right) \end{aligned} \quad (\text{F.24})$$

On constate que sur un arbre RC, le moment d'ordre 0 vaut toujours 1. Avec ces relations, il est possible de calculer rapidement les n moments sur chaque nœud du réseau RC. La complexité de l'algorithme est en O^3 . Il est possible de réduire cette complexité en calculant les moments sur le nœud $i + 1$ à partir du nœud i :

$$\sum_{j=0}^q (m_{j,i+1} \cdot s^j) = 1 - \sum_{k \in \delta_{i+1}} \left(R_k \sum_{l \in \zeta_k} \left(C_l \cdot \sum_{m=0}^{q-1} (m_{m,l} \cdot s^{m+1}) \right) \right) \quad (\text{F.25})$$

Que l'on peut réécrire :

$$\sum_{j=0}^q (m_{j,i+1} \cdot s^j) = 1 - \sum_{k \in \delta_i} \left(R_k \sum_{l \in \zeta_k} \left(C_l \cdot \sum_{m=0}^{q-1} (m_{m,l} \cdot s^{m+1}) \right) \right) - R_{i+1} \sum_{l \in \zeta_{i+1}} \left(C_l \cdot \sum_{m=0}^{q-1} (m_{m,l} \cdot s^{m+1}) \right) \quad (\text{F.26})$$

Soit :

$$\sum_{j=0}^q (m_{j,i+1} \cdot s^j) = \sum_{j=0}^q (m_{j,i} \cdot s^j) - R_{i+1} \sum_{l \in \zeta_{i+1}} \left(C_l \cdot \sum_{m=0}^{q-1} (m_{m,l} \cdot s^{m+1}) \right) \quad (\text{F.27})$$

Les moments du nœud $i + 1$ se calculent alors avec les relations :

$$\begin{aligned}
 0 : m_{0,i+1} &= 1 \\
 1 : m_{1,i+1} &= m_{1,i} - R_{i+1} \sum_{l \in \zeta_{i+1}} C_l \\
 2 : m_{2,i+1} &= m_{2,i} - R_{i+1} \sum_{l \in \zeta_{i+1}} (C_l \cdot m_{1,l}) \\
 &\dots \\
 n : m_{n,i+1} &= m_{n,i} - R_{i+1} \sum_{l \in \zeta_{i+1}} (C_l \cdot m_{n-1,l})
 \end{aligned} \tag{F.28}$$

L'équation de récurrence pour déterminer le n ème moment sur les nœuds d'un arbre RC est donc donnée par :

$$\begin{cases}
 m_{n,i+1} = m_{n,i} - R_{i+1} \sum_{l \in \zeta_{i+1}} (C_l \cdot m_{n-1,l}) \\
 m_{0,i>0} = 1 \\
 m_{n,0} = 0
 \end{cases} \tag{F.29}$$

Le calcul des q moments est donc récursif sur un arbre RC, et possède une complexité en σ^2 par rapport au nombre de nœuds.

F.3 Variation des paramètres y_i lorsqu'une capacité change

Nous voulons être en mesure de construire l'influence de la variation de la charge de la porte au deuxième ordre lorsqu'une capacité sur le réseau change. Sur chaque nœud du réseau RC, nous avons écrit l'admittance :

$$y(s) = y_1 \cdot s + y_2 \cdot s^2 + y_3 \cdot s^3 \tag{F.30}$$

Nous voulons obtenir simplement la valeur de y_1 , y_2 et y_3 en fonction de la variation de cette capacité. La façon la plus efficace est de calculer cette influence de la même façon, et en même temps, que le calcul des y_i . Il faut donc trouver des "fonctions de transfert" pour la résistance et la capacité qui permettent de propager cette influence :

$$\begin{array}{ccc}
 & \text{Admittance simple} & \\
 \Delta y_{u1} & & \Delta y_{d1} \\
 \Delta y_{u2} & \longleftarrow & \Delta y_{d2} \\
 \Delta y_{u3} & & \Delta y_{d3}
 \end{array} \tag{F.31}$$

F.3.1 La capacité

Dans ce premier cas, le résultat est immédiat. Nous savons que :

$$y_{u1} = C + y_{d1} \tag{F.32}$$

$$y_{u2} = y_{d2} \quad (\text{F.33})$$

$$y_{u3} = y_{d3} \quad (\text{F.34})$$

Nous en déduisons immédiatement que

$$\Delta y_{u1} = \Delta y_{d1} \quad (\text{F.35})$$

$$\Delta y_{u2} = \Delta y_{d2} \quad (\text{F.36})$$

$$\Delta y_{u3} = \Delta y_{d3} \quad (\text{F.37})$$

F.3.2 La résistance

Nous approximons les admittances par leur série de Taylor :

$$\Delta y_1 = \Delta C \quad (\text{F.38})$$

$$\Delta y_2 = \alpha_2 \cdot \Delta C + \beta_2 \cdot \frac{\Delta C^2}{2} \quad (\text{F.39})$$

$$\Delta y_3 = \alpha_3 \cdot \Delta C + \beta_3 \cdot \frac{\Delta C^2}{2} + \gamma_3 \cdot \frac{\Delta C^3}{6} \quad (\text{F.40})$$

Calculons Δy_{u2} et Δy_{u3} . Nous avons initialement :

$$y_{u2}(C_0) = y_{d2}(C_0) - R \cdot y_{d1}(C_0)^2 \quad (\text{F.41})$$

$$y_{u3}(C_0) = y_{d3}(C_0) - 2 \cdot R \cdot y_{d1}(C_0) \cdot y_{d2}(C_0) + R^2 \cdot y_{d1}(C_0)^3 \quad (\text{F.42})$$

Le développement en série de Taylor de y_{u2} et y_{u3} autour de C_0 vaut :

$$y_{u2}(C) = y_{u2}(C_0) + (C - C_0) \cdot y_{u2}'(C_0) + \frac{(C - C_0)^2}{2} \cdot y_{u2}''(C_0) \quad (\text{F.43})$$

$$y_{u3}(C) = y_{u3}(C_0) + (C - C_0) \cdot y_{u3}'(C_0) + \frac{(C - C_0)^2}{2} \cdot y_{u3}''(C_0) + \frac{(C - C_0)^3}{6} \cdot y_{u3}'''(C_0) \quad (\text{F.44})$$

Avec $\Delta y_{ui} = y_{ui}(C) - y_{ui}(C_0)$ et $\Delta C = C - C_0$, nous pouvons écrire plus concisément :

$$\Delta y_{u2} = \Delta C \cdot y_{u2}'(C_0) + \frac{\Delta C^2}{2} \cdot y_{u2}''(C_0) \quad (\text{F.45})$$

$$\Delta y_{u3} = \Delta C \cdot y_{u3}'(C_0) + \frac{\Delta C^2}{2} \cdot y_{u3}''(C_0) + \frac{\Delta C^3}{6} \cdot y_{u3}'''(C_0) \quad (\text{F.46})$$

Calculons maintenant les dérivées de y_{u2} et y_{u3} :

$$y_{u2} = y_{d2} - R \cdot y_{d1}^2 \quad (\text{F.47})$$

$$y_{u2}' = y_{d2}' - 2 \cdot R \cdot y_{d1}' \cdot y_{d1} \quad (\text{F.48})$$

$$y''_{u2} = y''_{d2} - 2.R.y'_{d1}{}^2 \quad (\text{F.49})$$

$$y_{u3} = y_{d3} - 2.R.y_{d1}.y_{d2} + R^2.y_{d1}^3 \quad (\text{F.50})$$

$$y'_{u3} = y'_{d3} - 2.R.y'_{d1}.y_{d2} - 2.R.y_{d1}.y'_{d2} + 3.R^2.y'_{d1}.y_{d1}^2 \quad (\text{F.51})$$

$$y''_{u3} = y''_{d3} - 4.R.y'_{d1}.y'_{d2} - 2.R.y_{d1}.y''_{d2} + 6.R^2.y'_{d1}{}^2.y_{d1} \quad (\text{F.52})$$

$$y'''_{u3} = y'''_{d3} - 6.R.y'_{d1}.y'_{d2} + 6.R^2.y'_{d1}{}^3 \quad (\text{F.53})$$

Le calcul des dérivées des y_{di} s'obtient avec :

$$y_{d2}(C) = y_{d2}(C_0) + \alpha_2.(C - C_0) + \beta_2.\frac{(C - C_0)^2}{2} \quad (\text{F.54})$$

$$y'_{d2}(C) = \alpha_2 + \beta_2.(C - C_0) \quad (\text{F.55})$$

$$y''_{d2}(C) = \beta_2 \quad (\text{F.56})$$

$$y_{d3}(C) = y_{d3}(C_0) + \alpha_3.(C - C_0) + \beta_3.\frac{(C - C_0)^2}{2} + \gamma_3.\frac{(C - C_0)^3}{6} \quad (\text{F.57})$$

$$y'_{d3}(C) = \alpha_3 + \beta_3.(C - C_0) + \gamma_3.\frac{(C - C_0)^2}{2} \quad (\text{F.58})$$

$$y''_{d3}(C) = \beta_3 + \gamma_3.(C - C_0) \quad (\text{F.59})$$

$$y'''_{d3}(C) = \gamma_3 \quad (\text{F.60})$$

En $C = C_0$, nous obtenons :

$$y'_{d1}(C_0) = 1 \quad (\text{F.61})$$

$$y'_{d2}(C_0) = \alpha_2 \quad (\text{F.62})$$

$$y''_{d2}(C_0) = \beta_2 \quad (\text{F.63})$$

$$y'_{d3}(C_0) = \alpha_3 \quad (\text{F.64})$$

$$y''_{d3}(C_0) = \beta_3 \quad (\text{F.65})$$

$$y'''_{d3}(C_0) = \gamma_3 \quad (\text{F.66})$$

Ce qui nous permet d'écrire :

$$\Delta y_{u2} = \Delta C.(\alpha_2 - 2.R.y_{d1}) + \frac{\Delta C^2}{2}.(\beta_2 - 2.R) \quad (\text{F.67})$$

$$\begin{aligned} \Delta y_{u3} = \Delta C. & \left(\alpha_3 - 2.R.y_{d2} - 2.R.y_{d1}.\alpha_2 + 3.R^2.y_{d1}^2 \right) \\ & + \frac{\Delta C^2}{2}.(\beta_3 - 4.R.\alpha_2 - 2.R.y_{d1}.\beta_2 + 6.R^2.y_{d1}) \\ & + \frac{\Delta C^3}{6}.(\gamma_3 - 6.R.\beta_2 + 6.R^2) \end{aligned} \quad (\text{F.68})$$

$$(\text{F.69})$$

Nous obtenons donc les équations de récurrence suivantes :

$$\alpha_{u2} = \alpha_{d2} - 2.R.y_{d1} \quad (\text{F.70})$$

$$\beta_{u2} = \beta_{d2} - 2.R \quad (\text{F.71})$$

$$\alpha_{u3} = \alpha_{d3} - 2.R.y_{d2} - 2.R.y_{d1}.\alpha_{d2} + 3.R^2.y_{d1}^2 \quad (\text{F.72})$$

$$\beta_{u3} = \beta_{d3} - 4.R.\alpha_{d2} - 2.R.y_{d1}.\beta_{d2} + 6.R^2.y_{d1} \quad (\text{F.73})$$

$$\gamma_{u3} = \gamma_{d3} - 6.R.\beta_{d2} + 6.R^2 \quad (\text{F.74})$$

A l'aide de ces equation de récurrence, nous obtenons donc 5 paramètres qui permettent de déterminer la variation des y_i , et donc de la charge en π , en fonction de la variation d'une capacité sur le réseau.

Annexe G

Liste des publications

Conférence	Colloque CAO de Circuits Intégrés et Systèmes, GDR 732, Aix-en-Provence, France.
Date	Mai 1999.
Auteurs	Avot Grégoire, Rosset-Louërat Marie-Minerve.
Titre	Influence et prise en compte des capacités de diaphonies dans la conception d'outils d'analyse temporelle pour les technologies profondément submicroniques.
Conférence	Design Automation and Test in Europe Conference User Forum (DATE'2000), Paris, France.
Date	Mars 2000.
Auteurs	Dioury Karim, Lester Anthony, Debreil Alain, Avot Gregoire, Greiner Alain, Rosset-Louërat Marie-Minerve.
Titre	Hierarchical Static Timing Analysis at Bull with HiTas.
Conférence	Mixed Design of Integrated Circuits and Systems (MIXDES'2001), Zakopane, Pologne.
Date	June 2001.
Auteurs	Avot Grégoire, Louërat Marie-Minerve.
Titre	Models for delay estimation taking into account both cross-talk and wire resistance for timing analysis.

Conférence	Design Automation and Test in Europe Conference (DATE'2002), Paris, France.
Date	Mars 2002.
Auteurs	Avot Grégoire, Greiner Alain, Louërat Marie-Minerve, Dioury Karim, Lester Anthony, Debreil Alain.
Titre	Use of MutiPhase Stability Intervals to handle Crosstalk with the Timing Analyzer HiTas.

Bibliographie

- [ADBTP01] Ravishankar Arunachalam, Ronald D. Blanton, and Lawrence T. Pileggi. False coupling interactions in static timing analysis. In *Proceedings of the DAC 2001 conference*, 2001.
- [Ava00] Avant! *Datasheet : Star RCXT*, 2000. Voir annexe page 201.
- [BBGP94] Pirouz Bazargan, Luc Burgun, Alain Greiner, and Frédéric Pétrot. Methodology and development of a complete cad system for digital vlsi design. In *Proceedings of the VIII Simposio Brasileiro de Concepcao de Circuitos Integrados*, pages 11–15, November 1994.
- [BBNH01] Geng Bai, Sudhakar Bobba, and Ibrahim N. Hajj. Static timing analysis including power supply noise effect on propagation delay in vlsi circuits. In *Proceedings of the DAC 2001 conference*, June 2001.
- [Ber00] David Bernard. *Caractérisation et modélisation des capacités parasites dues aux interconnexions en technologie CMOS fortement submicronique*. PhD thesis, Thèse de l'Université Montpellier II, 2000.
- [BvGvdM95] J. Beeftink, A.J. van Gendern, and N.P. van der Meijs. Accurate and efficient layout to circuit extraction for high speed mos and bipolar/bicmos integrated circuits. In *Proc. ICCD Conference*, pages 360–365, October 1995.
- [Cad] Cadence. *Dracula, the physical verification standard*. Voir annexe page 201.
- [Cad93] Cadence Design Systems. *Cadence Standard Parasitic Format (SPF) version C1.3*, September 1993.
- [Cad01a] Cadence. *Assura Parasitic Extractor*, 2001. Voir annexe page 201.
- [Cad01b] Cadence. *Celtic : Crosstalk analyzer for cell-based desings*, 2001. Voir annexe page 201.
- [Cad01c] Cadence. *Noise - Aware timing Analysis*, 2001. Voir annexe page 201.

- [CAKK00a] Pinhong Chen, Desmond A. Kirkpatrick, and Kurt Keutzer. Switching window computation for static timing analysis in presence of crosstalk noise. In *Proceedings of the ICCAD 2000 conference*, 2000.
- [CAKK00b] Pinong Chen, Desmond A. Kirkpatrick, and Kurt Keutzer. Miller factor for gate-level coupling delay calculation. In *Proceedings of the ICCAD 2000 conference*, 2000.
- [Chi92] Messaoud Chiheb. *Analyse de la performance de systemes electroniques complexes composees de circuits VLSI ASIC utilisant des latches level-sensitive*. PhD thesis, Thèse de l'Université Pierre et Marie Curie, 1992.
- [CNTP91] L. Ratzlaff Curtis, Gopal Nanda, and Lawrence T. Pillage. Rice : Rapid interconnect circuit evaluator. In *IEEE Design Automation Conference*, pages 555–560, 1991.
- [Des98] Jean-Lou Desbarbieux. *Conception et réalisation d'un contrôleur réseau programmable pour machine parallèle de type 'grappe de PC'*. PhD thesis, Thèse de l'Université Pierre et Marie Curie, 1998.
- [Dio98] Karim Dioury. *Analyse temporelle hiérarchique des circuits VLSI à très haute densité d'intégration*. PhD thesis, Thèse de l'Université Pierre et Marie Curie, 1998.
- [DLD⁺00] Karim Dioury, Anthony Lester, Alain Debreil, Grégoire Avot, Alain Greiner, and Marie-Minerve Louërat. Hierarchical Static Timing Analysis at Bull with Hi-TAS. In *Proc. of the Design Automation and Test in Europe Date 2000 User Forum*, pages 55–60, March 2000. User Forum Prize.
- [DTP97] Florentin Dartu and Lawrence T. Pileggi. Calculating worst-case delays due to dominant capacitance coupling. In *Annual ACM IEEE Design Automation Conference*, pages 46–51, 1997.
- [Elm48] W.C. Elmore. The transient response of damped linear networks with particular regard to wideband amplifiers. In *Journal of applied physics*, volume 19, January 1948.
- [FTWCE00] Peivand F. Tehrani, Shang Woo Chyou, and Uma Ekambaram. Deep sub-micron static timing analysis in presence of crosstalk. In *Proceedings of the First International Symposium on Quality of Electronic Design*, March 2000.
- [GLM92] Alain Greiner, Marc Laurentin, and Roland Marbot. Desb, a functional abstractor for cmos vlsi circuits. In *Proceedings of the European Design Automation Conference*, pages 22–27, September 1992.
- [GM79] M Gontron and M Minoux. *Graphes et Algorithmes*. Eyrolles, 1979.

- [Haj92] Amjad Hajjar. *Modélisation des temps de propagation et analyse temporelle statique des circuits intégrés CMOS*. PhD thesis, Thèse de l'Université Pierre et Marie Curie, 1992.
- [HOT98] Akio Hirata, Hidetoshi Onodera, and Keikichi Tamaru. Proposal of a timing model for cmos logic gate driving a crc π load. In *ICCAD98*, 1998.
- [Ilp02] Fabrice Ilp. *Analyse du bruit due aux couplages capacitifs dans les circuits intégrés numériques fortement submicroniques*. PhD thesis, Thèse de l'Université Pierre et Marie Curie, 2002.
- [Int01] Intel. *Intel®Pentium®processor in the 478-pin package at 1.50GHz, 1.60GHz, 1.70GHz, 1.80GHz, 1.90GHz, and 2GHz*, August 2001. Voir annexe page 201.
- [Lau92] Marc Laurentin. *Abstraction fonctionnelle de circuits intégrés CMOS*. PhD thesis, Thèse de l'Université Pierre et Marie Curie, 1992.
- [LBG98] Anthony Lester, Pirouz Bazargan, and Alain Greiner. Yagle, a second generation functional abstractor for cmos vlsi circuits. In *Proceedings of the 10th International Conference on Microelectronics (ICM'98)*, pages 265–268, December 1998.
- [Les98] Anthony Lester. *Abstraction fonctionnelle des circuits numériques VLSI avec une méthode formelle basée sur une extraction de réseau de portes*. PhD thesis, Thèse de l'Université Pierre et Marie Curie, 1998.
- [Men97] Mentor Graphics. *Interconnect parasitic extraction for deep submicron IC design*, February 1997. Voir annexe page 201.
- [Men00] Mentor Graphics Corporation. *Eldo User's Manual, Document Number 310101 for Software Version v5.3_1.1*, 2000.
- [Met90] Meta-Software, Inc., 1300 White Oaks Road, Campbell, CA 95008. *HSPICE user's manual*, 1990.
- [OL89] Peter R O'Brien and Thomas L.Savarino. Modeling the driving-point characteristic of resistive interconnect for accurate delay estimation. In *Proceedings of the International Conference on Computer-Aided Design*, 1989.
- [Pay98] Daniel Payne. *Dynamic timing analysis for system-on-chip design*. Mentor Graphics, July 1998. Voir annexe page 201.
- [PBSG01] Marie-Martine Paget, Pirouz Bazargan Sabet, and Alain Greiner. An example of practice based engineering education : the design of a microprogrammed mips processor with the alliance cad system. In *Proceedings of the Computer Aided Learning In Engineering (CALIE'2001)*, pages 47–50, November 2001. <http://www-asim.lip6.fr/alliance/doc/overview/overview.html>.

- [PTVF88] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C : The art of scientific computing*. Cambridge University Press, 2 edition, 1988. version 2.08. Site web : <http://www.nr.com>.
- [QPP94] Jessica Qian, Satyamurthy Pullela, and Lawrence Pillage. Modeling the effective capacitance for the rc interconnect of cmos gates. In *IEEE transactions on computer-aided design of integrated circuits and systems*, volume 13, pages 1526–1535, December 1994.
- [RLB00] Matthias Ringe, Thomas Lindenkreuz, and Erich Barke. Static timing analysis taking crosstalk into account. In *Proceedings of the Design, Automation and Test in Europe.*, March 2000.
- [RPAH83] Jorge Rubinstein, Paul Penfield, and Mark A. Horowitz. Signal delay in rc tree networks. In *IEEE Transaction on computer aided design*, July 1983.
- [SBO⁺01] Supamas Sirichotiyakul, David Blaauw, Chanhee Oh, Rafi Levy, Vladimir Zolotov, and Jingyan Zuo. Driver modeling and alignment for worst case delay noise. In *Proceedings of the ICCAD 2000 conference*, 2001.
- [SDM99] Yasuhiko Sasaki and Giovanni De Micheli. Crosstalk delay analysis using relative window method. In *IEEE Int. Asic SOC Conference 12th*, September 1999.
- [Seq01a] Sequence. *ExtractionStage*, 2001. Voir annexe page 201.
- [Seq01b] Sequence. *ShowTime*, 2001. Voir annexe page 201.
- [Sic98] Etienne Sicard. *Habilitation a diriger des recherches*. PhD thesis, INSA Toulouse, 1998.
- [ST83] T. Sukarai and K. Tamaru. Simple formulas for two and three dimensional capacitances. In *IEEE Transactions on Electron Devices*, pages 183–185, February 1983.
- [SvdMvG⁺95] T. Smedes, N.P. van der Meijs, A.J. van Genderen, P.J.H. Elias, and R.R.J. Vanoppen. Layout extraction of 3d models for interconnect and substrate parasitics. In *Proceedings ESSDERC*, 1995.
- [Syn97] Synopsys. *Synopsys PrimeTime datasheet*, October 1997. Voir annexe page 201.
- [Syn01a] Synopsys. *Arcadia : RC extraction for timing, power, reliability and signal integrity analysis*, October 2001. Voir annexe page 201.
- [Syn01b] Synopsys. *Prime Time SI : Static Crosstalk Analysis*, 2001. Voir annexe page 201.
- [Syn01c] Synopsys. *Static Crosstalk Analysis*, April 2001. Voir annexe page 201.

- [TPA90] Lawrence T. Pillage and Ronald A. Rohrer. Asymptotic waveform evaluation for timing analysis. In *IEEE Transactions on computer aided design*, volume 9, pages 352–366, April 1990.
- [Van98] Eric Vanier. *Caractérisation et optimisation temporelles des interconnexions dans les circuits sub-microniques CMOS*. PhD thesis, Thèse de l'Université Montpellier II, 1998.
- [vGvdM97] A.J. van Gendern and N.P. van der Meijs. *Space 3D capacitance extraction user's manual*. Delft university of technology, 1997.