



HAL
open science

Sur la vérification des systèmes digitaux

André Verdillon

► **To cite this version:**

André Verdillon. Sur la vérification des systèmes digitaux. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG; Université Joseph-Fourier - Grenoble I, 1977. tel-00287585

HAL Id: tel-00287585

<https://theses.hal.science/tel-00287585>

Submitted on 12 Jun 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE

présentée à

Université Scientifique et Médicale de Grenoble
Institut National Polytechnique de Grenoble

pour obtenir le grade de

DOCTEUR D'ETAT
Es SCIENCES
Informatique

André VERDILLON



SUR LA VERIFICATION DES SYSTEMES DIGITAUX.



Thèse soutenue le 13 septembre 1977 devant la Commission d'Examen :

Président : J. KUNTZMANN

Examineurs : P. AZEMA

R. DAVID

E.J. Mc CLUSKEY

J.C. RAULT

M. SAKAROVITCH

G. SAUCIER

THESE

présentée à

Université Scientifique et Médicale de Grenoble
Institut National Polytechnique de Grenoble

pour obtenir le grade de

DOCTEUR D'ETAT
Es SCIENCES
Informatique

André VERDILLON



SUR LA VERIFICATION DES SYSTEMES DIGITAUX.



Thèse soutenue le 13 septembre 1977 devant la Commission d'Examen :

Président : J. KUNTZMANN

Examineurs : P. AZEMA

R. DAVID

E.J. Mc CLUSKEY

J.C. RAULT

M. SAKAROVITCH

G. SAUCIER

UNIVERSITE SCIENTIFIQUE
ET MEDICALE DE GRENOBLE

Monsieur Gabriel CAU : Président
Monsieur Pierre JULLIEN : Vice Président

MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.

PROFESSEURS TITULAIRES

MM.	AMBLARD Pierre	Clinique de dermatologie
	ARNAUD Paul	Chimie
	ARVIEU Robert	I.S.N.
	AUBERT Guy	Physique
	AYANT Yves	Physique approfondie
Mme.	BARBIER Marie-Jeanne	Electrochimie
MM.	BARBIER Jean-Claude	Physique expérimentale
	BARBIER Reynold	Géologie appliquée
	BARJON Robert	Physique nucléaire
	BARNOU Fernand	Biosynthèse de la cellulose
	BARRA Jean-René	Statistiques
	BARRIE Joseph	Clinique chirurgicale
	BEAUDOING André	Clinique de pédiatrie et puériculture
	BELORIZKY Elie	Physique
	BERNARD Alain	Mathématiques pures
Mme.	BERTRANDIAS Françoise	Mathématiques pures
MM.	BERTRANDIAS Jean-Paul	Mathématiques pures
	BEZEZ Henri	Pathologie chirurgicale
	BLAMBERT Maurice	Mathématiques pures
	BOLLIET Louis	Informatique (IUT B)
	BONNET Jean-Louis	Clinique ophtalmologique
	BONNET-EYMARD Joseph	Clinique gastro-entérologique
Mme.	BONNIER Marie-Jeanne	Chimie générale
MM.	BOUCHERLE André	Chimie et toxicologie
	BOUCHEZ Robert	Physique nucléaire
	BOUSSARD Jean-Claude	Mathématiques appliquées
	BOUTET DE MONVEL Louis	Mathématiques pures
	BRAVARD Yves	Géographie
	CABANEL Guy	Clinique rhumatologique et hydrologique
	CALAS François	Anatomie
	CARLIER Georges	Biologie végétale
	CARRAZ Gilbert	Biologie animale et pharmacodynamie
	CAU Gabriel	Médecine légale et toxicologie
	CAUQUIS Georges	Chimie organique
	CHABAUTY Claude	Mathématiques pures
	CHARACHON Robert	Clinique oto-rhino-laryngologique
	CHATEAU Robert	Clinique de neurologie
	CHIBON Pierre	Biologie animale
	COEUR André	Pharmacie chimique et chimie analytique
	CONTAMIN Robert	Clinique gynécologique
	COUDERC Pierre	Anatomie pathologique

Mme.	DEBELMAS Anne-Marie	Matière médicale
MM.	DEBELMAS Jacques	Géologie générale
	DEGRANGE Charles	Zoologie
	DELORMAS Pierre	Pneumophtisiologie
	DEPORTES Charles	Chimie minérale
	DESRE Pierre	Métallurgie
	DESSAUX Georges	Physiologie animale
	DODU Jacques	Mécanique appliquée (IUT I)
	DOLIQUE Jean-Michel	Physique des plasmas
	DREYFUS Bernard	Thermodynamique
	DUCROS Pierre	Cristallographie
	GAGNAIRE Didier	Chimie physique
	GALVANI Octave	Mathématiques pures
	GASTINEL Noël	Analyse numérique
	GAVEND Michel	Pharmacologie
	GEINDRE Michel	Electroradiologie
	GERBER Robert	Mathématiques pures
	GERMAIN Jean-Pierre	Mécanique
	GIRAUD Pierre	Géologie
	JANIN Bernard	Géographie
	KAHANE André	Physique générale
	KOSZUL Jean-Louis	Mathématiques pures
	KLEIN Joseph	Mathématiques pures
	KRAVITCHENKO Julien	Mécanique
	KUNTZMANN Jean	Mathématiques appliquées
	LACAZE Albert	Thermodynamique
	LACHARME Jean	Biologie végétale
Mme.	LAJZEROWICZ Janine	Physique
MM.	LAJZEROWICZ Joseph	Physique
	LATREILLE René	Chirurgie générale
	LATURAZE Jean	Biochimie pharmaceutique
	LAURENT Pierre-Jean	Mathématiques Appliquées
	LEDRU Jean	Clinique médicale B
	LE ROY Philippe	Mécanique (IUT I)
	LLIBOUTRY Louis	Géophysique
	LOISEAUX Pierre	Sciences nucléaires
	LONGEQUEUE Jean-Pierre	Physique nucléaire
	LOUP Jean	Géographie
Melle	LUTZ Elisabeth	Mathématiques pures
MM.	MALINAS Yves	Clinique obstétricale
	MARTIN-NOEL Pierre	Clinique cardiologique
	MAZARE Yves	Clinique médicale A
	MICHEL Robert	Minéralogie et pétrographie
	MICOUD Max	Clinique maladies infectieuses
	MOURIQUAND Claude	Histologie
	MOUSSA André	Chimie nucléaire
	NOZIERES Philippe	Spectrométrie physique
	OZENDA Paul	Botanique
	PAYAN Jean-Jacques	Mathématiques pures
	PEBAY-PEYROULA Jean-Claude	Physique
	PERRET Jean	Semeiologie médicale (Neurologie)
	RASSAT André	Chimie systématique
	RENARD Michel	Thermodynamique
	REVOL Michel	Urologie
	RINALDI Renaud	Physique
	DE ROUGEMONT Jacques	Neuro-chirurgie
	SEIGNEURIN Raymond	Microbiologie et Hygiène
	SENGEL Philippe	Zoologie
	SIBILLE Robert	Construction mécanique (IUT I)

MM.	SOUTIF Michel	Physique générale
	TANCHE Maurice	Physiologie
	TRAYNARD Philippe	Chimie générale
	VAILLANT François	Zoologie
	VALENTIN Jacques	Physique nucléaire
	VAUQUOIS Bernard	Calcul électronique
Mme.	VERAIN Alice	Pharmacie galénique
MM.	VERAIN André	Physique
	VEYRET Paul	Géographie
	VIGNAIS Pierre	Biochimie médicale

PROFESSEURS ASSOCIES

MM.	CRABBE Pierre	CERMO
	DEMBICKI Eugéniuz	Mécanique
	JOHNSON Thomas	Mathématiques appliquées
	PENNEY Thomas	Physique

PROFESSEURS SANS CHAIRE

Melle	AGNIUS-DELORD Claudine	Physique pharmaceutique
	ALARY Josette	Chimie analytique
MM.	AMBROISE-THOMAS Pierre	Parasitologie
	ARMAND Gilbert	Géographie
	BENZAKEN Claude	Mathématiques appliquées
	BIAREZ Jean-Pierre	Mécanique
	BILLET Jean	Géographie
	BOUCHET Yves	Anatomie
	BRUGEL Lucien	Energétique (IUT I)
	BUISSON René	Physique (IUT I)
	BUTEL Jean	Orthopédie
	COHEN ADDAD Pierre	Spectrométrie physique
	COLOMB Maurice	Biochimie
	CONTE René	Physique (IUT I)
	DELOBEL Claude	M.I.A.G.
	DEPASSEL Roger	Mécanique des fluides
	FONTAINE Jean-Marc	Mathématiques pures
	GAUTRON René	Chimie
	GIDON Paul	Géologie et minéralogie
	GLENAT René	Chimie organique
	GROULADE Joseph	Biologie médicale
	HACQUES Gérard	Calcul numérique
	HOLLARD Daniel	Hématologie
	HUGONOT Robert	Hygiène et médecine préventive
	IDELMAN Simon	Physiologie animale
	JOLY Jean-René	Mathématiques pures
	JULLIEN Pierre	Mathématiques appliquées
Mme.	KAHANE Josette	Physique
MM.	KRAKOWIACK Sacha	Mathématiques appliquées
	KUHN Gérard	Physique (IUT I)
	LUU DUC Cuong	Chimie organique
	MAYNARD Roger	Physique du solide
Mme.	MINIER Colette	Physique (IUT I)
MM.	PELMONT Jean	Biochimie
	PERRIAUX Jean-Jacques	Géologie et minéralogie
	PFISTER Jean-Claude	Physique du solide
Melle	PIERY Yvette	Physiologie animale

MM.	RAYNAUD Hervé	M.I.A.G.
	REBECQ Jacques	Biologie (CUS)
	REYMOND Jean-Charles	Chirurgie générale
	RICHARD Lucien	Biologie végétale
Mme.	RINAUDO Marguerite	Chimie macromoléculaire
MM.	ROBERT André	Chimie papetière
	SARRAZIN Roger	Anatomie et chirurgie
	SARROT-REYNAULD Jean	Géologie
	SIROT Louis	Chirurgie générale
Mme.	SOUTIF Jeanne	Physique générale
MM.	STIEGLITZ Paul	Anesthésiologie
	VIALON Pierre	Géologie
	VAN CUTSEM Bernard	Mathématiques appliquées

MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

MM.	ARMAND Yves	Chimie (IUT I)
	BACHELOT Yvan	Endocrinologie
	BARGE Michel	Neuro-chirurgie
	BEGUIN Claude	Chimie organique
Mme	BERIEL Hélène	Pharmacodynamie
MM.	BOST Michel	Pédiatrie
	BOUCHARLAT Jacques	Psychiatrie adultes
Mme.	BOUCHE Liane	Mathématiques (CUS)
MM.	BRODEAU François	Mathématiques (IUT B) (Personne étrangère habilitée à être directeur de thèse)
	CHAMBAZ Edmond	Biochimie médicale
	CHAMPETIER Jean	Anatomie et organogénèse
	CHARDON Michel	Géographie
	CHERADAME Hervé	Chimie papetière
	CHIAVERINA Jean	Biologie appliquée (EFP)
	CONTAMIN Charles	Chirurgie thoracique et cardio-vasculaire
	CORDONNIER Daniel	Néphrologie
	COULOMB Max	Radiologie
	CROUZET Guy	Radiologie
	CYROT Michel	Physique du solide
	DENIS Bernard	Cardiologie
	DOUCE Roland	Physiologie végétale
	DUSSAUD René	Mathématiques (CUS)
Mme.	ETERRADOSSI Jacqueline	Physiologie
MM.	FAURE Jacques	Médecine légale
	FAURE Gilbert	Urologie
	GAUTIER Robert	Chirurgie générale
	GIDON Maurice	Géologie
	GROS Yves	Physique (IUT I)
	GUIGNIER Michel	Thérapeutique
	GUITTON Jacques	Chimie
	HICTER Pierre	Chimie
	JALBERT Pierre	Histologie
	JULIEN-LAVILLAVROY Claude	O.R.L.
	KOLODIE Lucien	Hématologie
	LE NOC Pierre	Bactériologie-virologie
	MACHE Régis	Physiologie végétale
	MAGNIN Robert	Hygiène et médecine préventive
	MALLION Jean-Michel	Médecine du travail
	MARECHAL Jean	Mécanique (IUT I)
	MARTIN-BOUYER Michel	Chimie (CUS)
	MICHOULIER Jean	Physique (IUT I)

MM.	NEGRE Robert	Mécanique (IUT I)
	NEMOZ Alain	Thermodynamique
	NOUGARET Marcel	Automatique (IUT I)
	PARAMELLE Bernard	Pneumologie
	PECCOUD François	Analyse (IUT B) (Personnalité étrangère habilité à être directeur de thèse)
	PEFFEN René	Métallurgie (IUT I)
	PERRIER Guy	Géophysique-Glaciologie
	PHELIP Xavier	Rhumatologie
	RACHAIL Michel	Médecine interne
	RACINET Claude	Gynécologie et obstétrique
	RAMBAUD André	Hygiène et hydrologie (Pharmacie)
	RAMBAUD Pierre	Pédiatrie
	RAPHAEL Bernard	Stomatologie
Mme.	RENAUDET Jacqueline	Bactériologie (Pharmacie)
MM.	ROBERT Jean-Bernard	Chimie physique
	Romier Guy	Mathématiques (IUT B) (Personnalité étrangère habilité à être directeur de thèse)
	SCHAERER René	Cancérologie
	SHOM Jean-Claude	Chimie générale
	STOEBNER Pierre	Anatomie pathologie
	VROUSOS Constantin	Radiologie

MAITRES DE CONFERENCES ASSOCIES

MM.	DEVINE Roderick	Spectro physique
	HODGES Christopher	Transition de phases

Fait à SAINT MARTIN D'HERES, NOVEMBRE 1976.

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Monsieur Philippe TRAYNARD : Président

Monsieur Pierre-Jean LAURENT : Vice Président

PROFESSEURS TITULAIRES

MM.	BENOIT Jean	Radioélectricité
	BESSON Jean	Electrochimie
	BLOCH Daniel	Physique du solide
	BONNETAIN Lucien	Chimie minérale
	BONNIER Etienne	Electrochimie et électrometallurgie
	BOUDOURIS Georges	Radioélectricité
	BRISSENEAU Pierre	Physique du solide
	BUYLE-BODIN Maurice	Electronique
	COUMES André	Radioélectricité
	DURAND Francis	Métallurgie
	FELICI Noël	Electrostatique
	FUULARD Claude	Automatique
	LESPINARD Georges	Mécanique
	MOREAU René	Mécanique
	PARIAUD Jean-Charles	Chimie-Physique
	PAUTHENET René	Physique du solide
	PERRET René	Servomécanismes
	POLOUJADOFF Michel	Electrotechnique
	SILBER Robert	Mécanique des fluides

PROFESSEUR ASSOCIE

M.	ROUXEL Roland	Automatique
----	---------------	-------------

PROFESSEURS SANS CHAIRE

MM.	BLIMAN Samuel	Electronique
	BOUVARD Maurice	Génie mécanique
	COHEN Joseph	Electrotechnique
	LACOUME Jean-Louis	Géophysique
	LANCIA Roland	Electronique
	ROBERT François	Analyse Numérique
	VEILLON Gérard	Informatique fondamentale et appliquée
	ZADWORNÝ François	Electronique

MAITRES DE CONFERENCES

MM.	ANCEAU François	Mathématiques appliquées
	CHARTIER Germain	Electronique
	GUYOT Pierre	Chimie minérale
	IVANES Marcel	Electrotechnique
	JOUBERT Jean-Claude	Physique du solide
	MORET Roger	Electrotechnique nucléaire
	PIERRARD Jean-Marie	Mécanique
	SABONNADIÈRE Jean-Claude	Informatique fondamentale et appliquée
Mme.	SAUCIER Gabrièle	Informatique fondamentale et appliquée

MAITRE DE CONFERENCES ASSOCIE

M.	LANDAU Ioan	Automatique
----	-------------	-------------

CHERCHEURS DU C.N.R.S. (Directeur et Maîtres de Recherche)

MM.	FRUCHART Robert	Directeur de Recherche
	ANSARA Ibrahim	Maître de Recherche
	CARRE René	Maître de Recherche
	DRIOLE Jean	Maître de Recherche
	MATHIEU Jean-Claude	Maître de Recherche
	MUNIER Jacques	Maître de Recherche

Avant propos

Je voudrais exprimer toute ma reconnaissance à

Monsieur le Professeur J. KUNTZMANN qui a bien voulu m'accueillir dans son laboratoire. Je lui suis reconnaissant de m'avoir initié à la recherche et je suis particulièrement sensible à l'honneur qu'il me fait en acceptant de présider le jury.

Je remercie très vivement

Monsieur P. AZEMA, Maître de Recherche au CNRS,

Monsieur R. DAVID, Chargé de Recherche au CNRS,

Monsieur E.J. Mc CLUSKEY, Professeur à l'Université de Stanford (USA),

Monsieur J.C. RAULT, Chef du Service Recherche et Développement,
Direction de l'Informatique, Thomson C.S.F.

Monsieur M. SAKAROVITCH, Maître de Conférence à l'USMG,

Madame G. SAUCIER, Maître de Conférence à l'ENSIMAG, pour l'animation
cordiale de l'équipe "Conception et Sécurité des Systèmes Logiques",

qui ont bien voulu accepter de faire partie du jury.

Je remercie également mes camarades de l'équipe "Conception et Sécurité des Systèmes Logiques", en particulier Mlle TURCAT et Mr. RAINARD.

Je tiens aussi à remercier celles et ceux qui ont travaillé à la frappe et au tirage de cette thèse.

CHAPITRE I : VERIFICATION, TOUR D'HORIZON

INTRODUCTION

I - Méthode de vérification déterministe

I - 1. Vérification fonctionnelle

I - 2. Vérification analytique

I - 2.1. Théorie des pannes

I - 2.2. Modèles de circuits

I - 2.3. Génération de test par sensibilisation de chemin

I - 2.4. Génération algébrique de tests

II - Méthodes probabilistes

II - 1. Génération aléatoire de test

II - 2. Test de pannes aléatoires

CHAPITRE II : TEST OPTIMAL DES SYSTEMES SEQUENTIELS

INTRODUCTION

I - Représentation des machines normales et fausses

I - 1. Définitions

I - 2. Table de test des transitions

II - Simulation des fautes

II - 1. Simulation générale

II - 2. Simulation pour la détection

II - 3. Simulation pour la localisation

III - Procédures pour séquences de détection optimales

III - 1. Définition, d-dominance

III - 2. Procédures

IV - Procédures pour séquences de localisation optimales

IV - 1. Définition, l-dominance

IV - 2. Procédures

V - Extensions : pannes dans l'initialisation et machines asynchrones

V - 1. Pannes affectant l'initialisation

V - 2. Circuits asynchrones

CONCLUSION

CHAPITRE III : TEST DE RESEAUX REGULIERS

INTRODUCTION

I - Symétrie, automorphisme et test

I - 1. Réseau de portes

I - 2. Réseau de modules

II - Itération et test

II - 1. Itération simple

II - 2. Itération complexe

CONCLUSION

CHAPITRE IV : TEST DES PANNES MULTIPLES

I - Simulation des pannes

I - 1. Rappels sur la simulation des pannes simples

I - 2. Représentation des pannes multiples

I - 3. Simulation des pannes multiples

II - Test des pannes multiples

II - 1. Relations entre pannes multiples

II - 2. Méthode de test

III - Simulation des pannes multiples dans le séquentiel

INTRODUCTION

Le sujet de ce travail est la vérification des systèmes logiques. En effet, un système logique peut ne pas correspondre à ses spécifications pour de multiples raisons : erreurs de conception, défauts de fabrication, pannes dues à l'usure...

La vérification est particulièrement cruciale pour les systèmes en temps réel (contrôle de processus, systèmes embarqués, téléphones...) et pour les systèmes devant fonctionner dans un environnement sévère (température, vibrations...).

Au niveau composant, les nouvelles technologies très intégrées (LSI, ULSI) rendent la vérification difficile par rapport aux anciennes technologies du fait de :

- la faible accessibilité (faible rapport "pin/bit"),
- la proximité des éléments qui sont nés de nouvelles sources de pannes (couplages..).

Ainsi, il est à prévoir que l'accroissement de l'intégration augmentera le poids du paramètre vérification pour les composants.

Les pannes qui sont considérées ici obéissent à certaines hypothèses : elles sont permanentes et peuvent être modélisées logiquement. Un modèle courant de pannes est le collage à 0 (ou à 1) d'une connexion : du fait du défaut (diode claquée, transistor défectueux..) la connexion collée a toujours la valeur logique 0 (respectivement 1). Le problème de la vérification est alors de générer des tests dont l'application au système permette de décider s'il y a une panne (détection) et éventuellement la localiser (diagnostic). Bien que les erreurs de conception ne soient pas considérées ici, l'expérience a montré que le test des pannes (type collage) permettait de détecter aussi de nombreuses erreurs de conception.

Le premier chapitre est un tour d'horizon qui montre en particulier l'importance du modèle de panne et du modèle du circuit dans la génération de test.

Le deuxième chapitre traite de la génération de test optimum (respectivement pour la détection et le diagnostic). La méthode proposée améliore les résultats précédents sur la détection, tant sur la représentation des données que sur les algorithmes, et surtout permet de localiser les pannes.

Le troisième chapitre étudie les réseaux réguliers : symétriques et itératifs. Les propriétés de symétrie permettent de réduire le nombre de pannes à considérer pour la génération du test, facilitant d'autant celui-ci. Une caractérisation des réseaux itératifs ayant un test indépendant de la longueur, est donnée. L'étude d'itération plus complexe débouche sur une étude comparative du test de différents tests d'additionneurs.

Le quatrième chapitre concerne la génération de test des pannes multiples, c'est une extension de travaux précédents (thèse de 3e cycle). Utilisant une simulation des pannes multiples, la méthode proposée étudie comment le test de pannes simples peut être utilisé pour les pannes multiples.

Le cinquième chapitre est basé sur l'étude d'un système complexe : le multienregistreur du centre E10 (Centre de Commutation Téléphonique) du CNET. Dans un système d'une telle complexité il est impossible d'envisager le test du système comme un seul bloc. Il faut le partitionner et avoir un ordonnancement de test des parties. L'influence des différentes parties les unes sur les autres doit être prise en compte lors du test : c'est le test sous contraintes. Enfin, l'étude du coût de différentes politiques de réparation après diagnostic est étudiée.

Notons que l'intérêt de ce thème se concrétise par la parution de 2 livres :

H.Y. CHANG : "Fault Diagnosis of Digital Systems", Wiley New York 1970.

A.D. FRIEDMAN : "Fault Detection in Digital Circuits", Electrical, Eng. Series, Englewood Cliffs, N.Y. Prentice Hall, 1971;

une bibliographie sur le sujet ^{qui} regroupe plus de 1000 articles.

J.C. RAULT : "Bibliographie sur la Détection et la Localisation des défauts dans les Circuits Logiques", Rapport Thomson CSF;

et la création en 1971 d'un Congrès Annuel IEEE sur la vérification et la fiabilité : "Fault Tolerant Computing".

CHAPITRE I

TOUR D'HORIZON

INTRODUCTION

Tout d'abord une constatation : une vérification absolue d'un système est impossible à réaliser. En effet, si on ne connaît pas de bornes du nombre d'états du système faux, il n'y a aucune séquence finie qui permet de répondre à la question : le système a-t-il une panne ou non ? Mais même si l'on peut borner le nombre d'états des machines fausses, on peut très bien, pratiquement, ne pas pouvoir répondre à la question ci-dessus du fait des temps de calcul astronomiques que cela entraînerait.

La vérification ne peut être que relative à un modèle qui englobe le comportement sain (sans panne) du système et les comportements faux. Il est à noter que le système avec panne peut être de nature différente du système sans panne. Le nombre de boucles de réaction, donc d'états, peut augmenter du fait de court-circuits entre connexions (ainsi un système combinatoire peut devenir séquentiel). Un système synchrone peut se dégrader en système asynchrone si les pannes affectent les éléments de synchronisation. Un système déterministe peut engendrer des systèmes en pannes non déterministes du fait de la création de courses critiques ou d'aléas.

Le modèle de panne à considérer est particulièrement important.

- Il peut être indépendant de la réalisation, c'est le cas de l'hypothèse de non augmentation (ou augmentation bornée) du nombre des états.
- Il peut dépendre de la décomposition structurelle du système; on peut supposer que les 2 fonctions, transition et sortie, ne peuvent pas être en panne simultanément si ces fonctions sont réalisées séparément.
- Le système de panne peut même être dépendant de l'implantation si on considère les courts-circuits entre connexions.
- Le modèle peut dépendre du type de vérification que l'on fait : si en maintenance, l'hypothèse de la panne simple est valide, par contre en fin de fabrication l'hypothèse de la panne multiple est plus probable notamment pour ce qui est des techniques à haute intégration.

La présence de redondance, c'est-à-dire l'existence de pannes qui ne peuvent pas être testées et la présence de pannes très difficiles à tester parce que difficile à manifester (certains collages dans un additionneur 32 bits demandent, en moyenne, 2^{32} additions avant d'être manifestés) font que souvent on ne cherche pas à tester toutes les pannes. Il existe un compromis entre le coût du test d'une panne et le coût de sa non détection.

On peut distinguer, dans la vérification, deux aspects : la détection des pannes et leur diagnostic ou localisation. Notons que la localisation n'a pas besoin d'être parfaite : la résolution du diagnostic doit se faire à l'unité de remplacement près, cela peut être, par exemple, un processeur, une carte ou un composant.

La vérification est déterministe si l'on cherche à avoir une réponse ou/non aux questions : "le système a-t-il une panne?", "la panne est-elle située dans un tel module?".

Si l'on cherche simplement une réponse statistique, par exemple, parce que les pannes sont intermittentes et de nature aléatoire, on parle alors de test probabiliste.

On peut vérifier trois types de vérification :

- La vérification fonctionnelle. On se donne alors une description fonctionnelle du système (table de vérité d'une fonction booléenne, tableau d'état d'un automate, jeu d'instructions d'un ordinateur..), c'est-à-dire si on considère le système comme une boîte noire dont on a seulement la description de la fonction qu'elle réalise.
- La vérification analytique. On se donne alors une description fine, au niveau de la porte logique.
- La vérification structurelle. On tient alors compte lors de la vérification du découpage d'un système en sous-système (de façon hiérarchique ou coopératif) ainsi que des propriétés particulières des systèmes (symétrie, répétitivité, partie opérative/partie contrôle...).

I - METHODE DE VERIFICATION DETERMINISTE

I - 1. VERIFICATION FONCTIONNELLE

Dans ce qui suit, on suppose aucune connaissance de la réalisation matérielle, et on ne s'intéresse donc pas à la localisation.

a) Systeme combinatoire

On suppose donnée une description de la fonction booléenne, par exemple, sous forme de table de vérité ou d'expression booléenne. On suppose aussi que les pannes conservent au système son caractère combinatoire. Soit n le nombre d'entrées, il y a 2^{2^n} fonctions fausses possibles, cependant il suffit de contrôler les 2^n possibilités d'entrée pour déterminer si la fonction est sans panne ou non. Remarquons que par cette méthode le test d'une addition de 32 bits demande 2^{64} vecteurs ce qui rend cette méthode inutilisable pratiquement, si ce n'est pour les très petits circuits.

b) Systeme séquentiel

Soit une description du système sous forme de tableau (ou graphe) d'états. Il n'existe pas, pour le moment, de méthode générale de génération de séquences de test (on parle aussi de séquences d'identification) si ce n'est une méthode analysant toutes les séquences pour déterminer celles qui identifient l'automate.

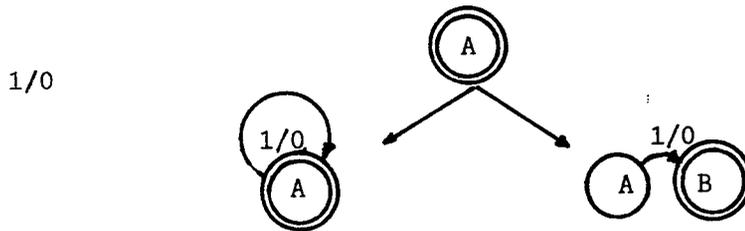
Pour analyser une séquence afin de savoir si elle identifie l'automate, on construit un arbre où chaque noeud correspond à un automate incomplet qui est compatible avec la séquence d'entrée/sortie de l'automate exact.

EXEMPLE

Soit la séquence d'entrée/sortie à analyser :

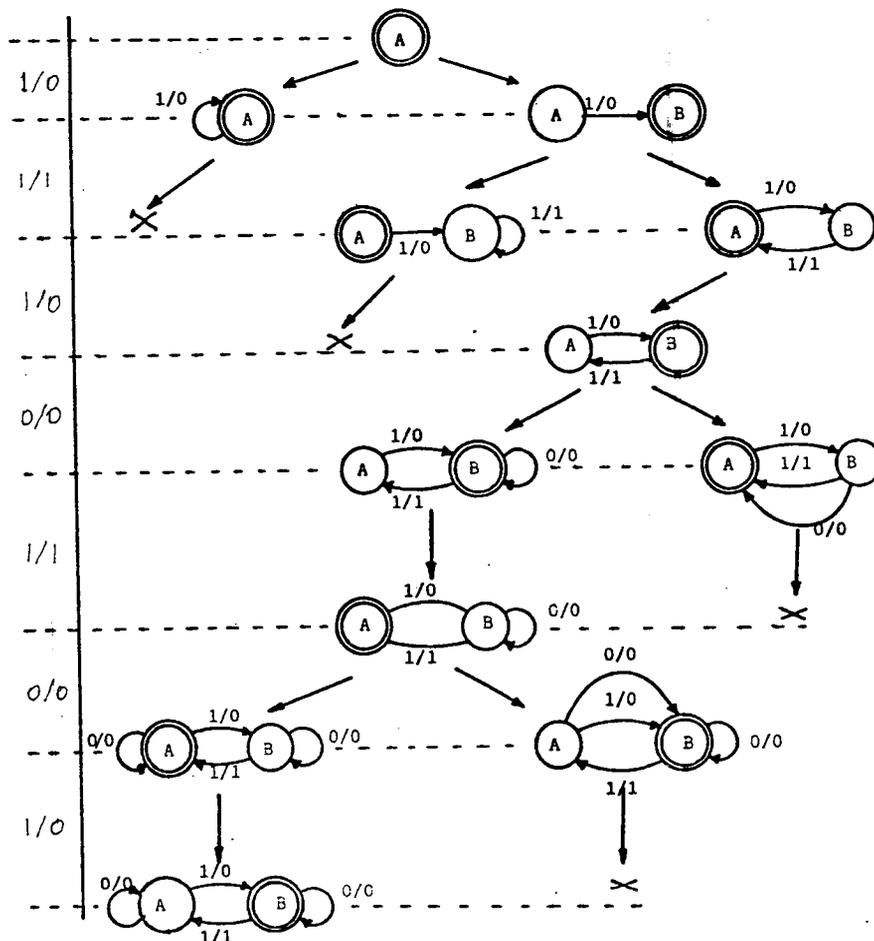
$\left. \begin{array}{l} \text{entrées} \\ \text{sorties} \end{array} \right\}$	$\left\{ \begin{array}{cccccc} 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{array} \right\}$	on suppose que l'automate a au plus
		2 états.

Au départ on suppose que l'automate est dans un état appelé A. La séquence $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ soit laisse l'automate en A, soit l'amène dans un nouvel état : B. Ceci peut se schématiser par l'arbre suivant :

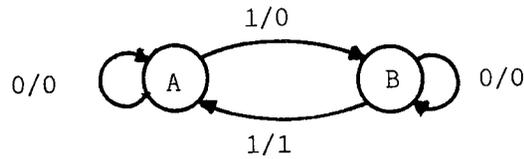


Mais si l'on reste dans l'état A et que l'on a comme entrée 1, la sortie doit rester à 0, or la séquence d'entrée/sortie considérée est $\begin{Bmatrix} 1 & 1 \\ 0 & 1 \end{Bmatrix}$ d'où une impossibilité (qui est notée X).

On peut ainsi construire de proche en proche l'arbre des choix et si en fin de séquence il ne reste plus qu'un choix (ou des choix équivalents) alors la séquence identifie l'automate.



La séquence précédente identifie donc l'automate :



Notons que cette analyse est très laborieuse, en particulier le nombre de branchement dans l'arbre est égal, le plus souvent, au nombre d'états de l'automate.

Si le problème de l'identification n'a pas, en général, de solution (si ce n'est la méthode exhaustive précédente), on sait par contre résoudre le problème quand l'automate a une séquence de distinction ("distinguishing sequence"). Une séquence de distinction est telle que la réponse caractérise de façon unique l'état initial (avant application de la séquence).

L'identification consiste en 3 phases :

Phase 1 : Séquence de positionnement ("homing sequence") éventuellement adaptative, qui positionne la machine dans un état connu.

Phase 2 : Comptage des états par une séquence qui passe par tous les états en appliquant chaque fois la séquence de distinction.

Phase 3 : Vérification des transitions : après chaque transition, l'application de la séquence de distinction permet de s'assurer de l'état final et donc de vérifier la transition.

Malheureusement, toutes les machines ne possèdent pas de séquences de distinction.

I - 2. VERIFICATION ANALYTIQUE

I - 2.1. Théorie des pannes

a) Dans les circuits combinatoires

Comme le processus de génération de vecteur de test est coûteux on cherche à le simplifier. Un des moyens de simplification est la connaissance des structures qui existent sur l'ensemble des pannes. On est, en particulier, amené à réduire le nombre de pannes à considérer.

Soit R un réseau élaborant la fonction f, X étant le vecteur d'entrée du réseau. Si α est une panne du réseau, la fonction fautive correspondant à la panne α est notée f_α .

Pour les problèmes de localisation on peut définir :

Définition

La panne β cache la panne α si :

$$[f_\alpha(X) \neq f(X)] \Rightarrow [f_\beta(X) = f_\alpha(X)] \quad \circ$$

Cette définition signifie que si l'on veut distinguer la panne α de β , le test de α ne sert à rien.

Cette relation est un préordre, la relation d'isovalence associée est :

Définition

Les pannes α et β sont indiscernables si $f_\alpha = f_\beta$ \circ

La résolution de la localisation ne peut se faire qu'au classes d'indiscernabilité près. Le problème de la localisation peut être simplifié en ne considérant que les classes d'indiscernabilité.

Soit T_α l'ensemble des vecteurs d'entrée détectant α ($T_\alpha = \{X \mid f_\alpha(X) \neq f(X)\}$).

Pour les problèmes de détection on peut définir :

Définition

La panne α est moins testable^{*} que β si $T_\alpha \subset T_\beta$ \circ

On peut voir que si α est moins testable que β alors si α est testé, β l'est aussi.

* On dit aussi β domine α .

Cette relation est un préordre et la relation d'isovalence associé est :

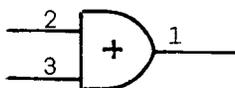
Définition

Les pannes α et β sont isotestables si $T_\alpha = T_\beta$ \circ

La relation "moins testable" induit un ordre sur les classes d'isotestabilité et pour tester toutes les pannes il suffit de tester un représentant de chaque classe d'isotestabilité minimale (sans considérer les pannes indétectables).

EXEMPLE

Considérons les collages aux entrées/sorties d'une porte OU .



il suffit de détecter les 3 collages : $1_1, 2_0, 3_0$ (i_j signifie i collé à j) pour détecter les 6 collages.

Si on veut localiser les pannes on a 4 classes d'indiscernabilité :

$\{1_1, 2_1, 3_1\}, \{2_0\}, \{3_0\}, \{1_0\}$ Δ

Par la suite on ne considère qu'un seul type de collage sur les entrées d'une porte : les collages à 0 pour les portes OU et les collages à 1 pour les portes ET.

Les réseaux arborescents sont particulièrement simples à tester du fait des résultats suivants :

THEOREME

Pour détecter les collages simples dans un réseau arborescent il suffit de détecter les collages simples sur les entrées ainsi que sur les sorties des opérateurs primaires (dont toutes les entrées primaires) \square

COROLLAIRE

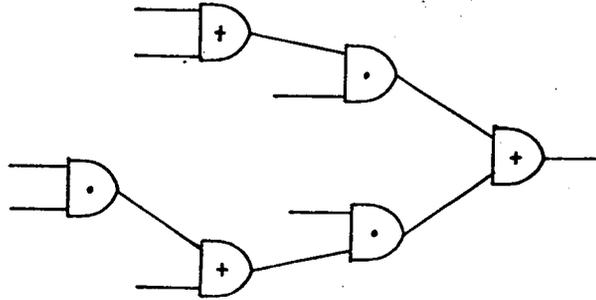
Pour détecter tous les collages simples d'un réseau arborescent ayant n entrées primaires et p opérations primaires, il faut tester au plus $n+p$ collages \square

Ce corollaire permet de comparer la longueur d'un test obtenu par méthode analytique dont le cardinal est inférieur à $2n$ alors que le test fonctionnel (exhaustif) est de 2^n .

REMARQUE

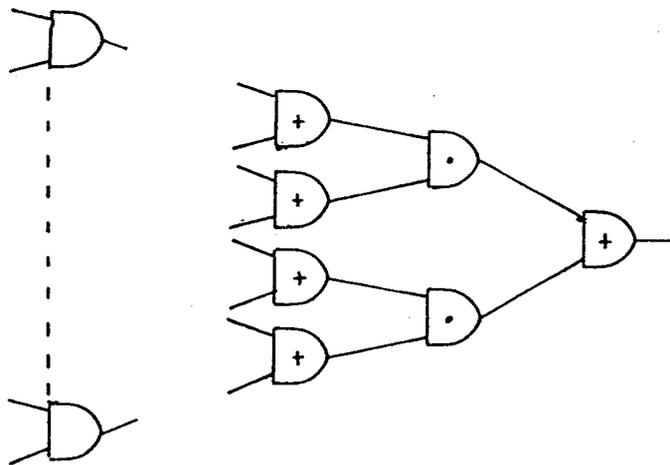
Les résultats précédents se généralisent facilement aux pseudo-arborescences qui sont des réseaux arborescents où la même variable d'entrée peut s'appliquer à plusieurs connexions d'entrée.

EXEMPLE 1. : Soit l'arborescence suivante :



Ce réseau a 13 connexions donc 26 collages simples. Il y a 7 entrées et 2 opérations primaires donc le test de 9 pannes suffit à tester tous les collages Δ

EXEMPLE 2. : Soit l'arborescence dichotomique, à n couches d'opérateurs, décrite par la figure ci-dessous.



- le nombre de connexions est $2^{n+1}-1$
- le nombre de collages simples est $2(2^{n+1}-1)$
- le nombre de classes d'isotestabilité est 2^{n+1}
- le nombre de classes d'isotestabilité minimale est 2^n+2^{n-1} Δ

Dans un réseau avec sortance multiple il n'est pas facile de construire exactement les classes d'indiscernabilité car cela suppose de calculer les fonctions fausses des pannes. Au lieu de calculer exactement les classes d'indiscernabilité on peut les approcher.

DEFINITION

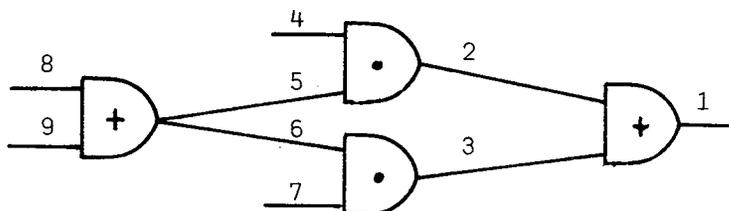
Soit α un collage (éventuellement multiple), le réseau actif de α est formé de l'ensemble des connexions c du réseau R telles qu'il existe un chemin joignant une entrée à une sortie en passant par c , sans passer par α ○

DEFINITION

Les pannes α et β sont R-équivalentes si les réseaux actifs de α et β sont les mêmes ○

Il est facile de voir que la R-équivalence implique l'indiscernabilité mais que le contraire n'est pas vrai en général. Les classes de R-équivalence sont simples à construire car elles ne font intervenir que les propriétés graphiques et que l'on n'a pas besoin d'évaluer de fonction.

EXEMPLE : Soit le réseau ci-dessous :



Il y a $3^9 - 1$ collages multiples. Le nombre de R-classes est 40 et le nombre de classes d'indiscernabilité est 28 △

REMARQUE

Dans les réseaux arborescents les R-classes sont les classes d'indiscernabilité.

La réduction du nombre de pannes à considérer est particulièrement importante pour les pannes multiples. En effet, s'il y a N connexions il y a $3^N - 1$ collages multiples possibles. Pour le réseau précédent il y a 2^{n+1}

connexions donc $3^{2^{n+1}} - 1$ collages multiples mais le nombre de classes d'indiscernabilité est de 2^{2^n} .

Dans le test des pannes multiples on essaye de se ramener au cas des pannes simples, pour cela on définit :

DEFINITION

Soit t un test de la panne α , on dit que β masque α pour t si t n'est pas un test pour la panne multiple $\alpha\beta$. ○

DEFINITION

Soit t un test de la panne α on dit que α détecte toutes les pannes la contenant si pour toute panne β , t détecte $\alpha\beta$ (c.a.d. aucune panne ne masque α point). ○

La première définition décrit quand le test des pannes simples n'est pas suffisant au test des pannes multiples et la deuxième définition décrit quand le test des pannes simples teste aussi les pannes multiples.

On peut montrer que :

THEOREME

Dans une arborescence, pour toute panne simple il existe un test pour lequel la panne simple détecte toutes les pannes multiples le contenant. □

COROLLAIRE

Dans une arborescence, avec n entrées primaires et p opérateurs primaires, il suffit de $n+p$ tests pour détecter tous les collages multiples. □

Dans le cas d'une (pseudo) arborescence 2 couches, on a le résultat :

THEOREME

Tout test des pannes simples est aussi un test des pannes multiples. □

b) Dans les circuits séquentiels

Dans les circuits séquentiels se pose la question de l'état initial. Si on peut faire l'hypothèse d'un système de "remise à zéro", celui-ci peut lui aussi être en panne. On peut supposer pour simplifier le problème que les pannes n'affectent pas la remise à 0.

On peut reprendre des définitions similaires au cas combinatoire, mais maintenant l'ensemble des séquences possibles est infini (Cf. Chapitre II) alors que dans le cas combinatoire l'ensemble des vecteurs était fini.

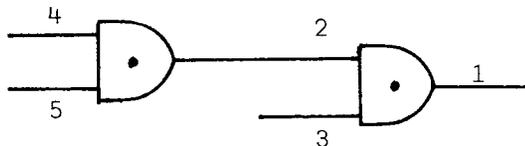
Une approche intéressante est de regarder si les relations définies sur les fonctions combinatoires (en coupant les boucles) peuvent être étendues à la machine séquentielle. Le seul résultat positif que l'on ait est que : si 2 pannes α et β sont indiscernables du point de vue combinatoire, elles le sont aussi du point de vue séquentiel. Notons que 2 pannes peuvent être indiscernables du point de vue séquentiel tout en étant discernables du point de vue combinatoire car 2 machines peuvent être équivalentes sans être identiques. Par contre, les relations "cache" "moins testable" et "isotestable" ne peuvent pas être étendues directement.

I - 2.2. Modèles de circuits

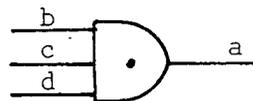
Afin de simplifier le test on peut effectuer sur le réseau des transformations.

a) Réduction d'une cascade de ET (ou bien de OU)

Le réseau

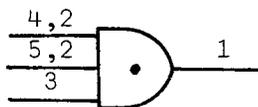


se réduit en :



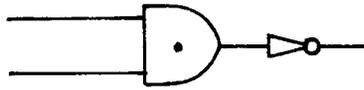
Notons que le collage simple 2 correspond au collage double b, c; si l'on veut conserver les collages initiaux on peut les reporter sur le réseau réduit en les répliquant Δ si nécessaire.

L'exemple ci-dessus donne :

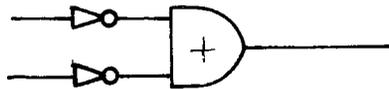


b) Application des lois de De Morgan

Le réseau :



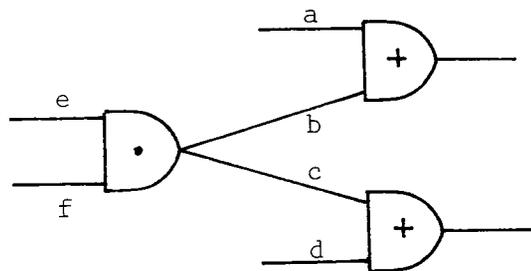
est équivalent au réseau



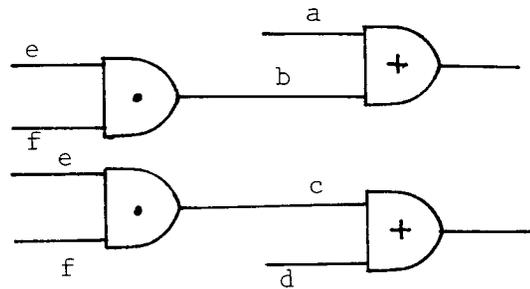
c) Transformation d'une sortance ("fan-out") multiple en arborescence

Ici encore une panne simple du réseau initial peut donner une panne multiple dans le réseau arborescent.

Le réseau :

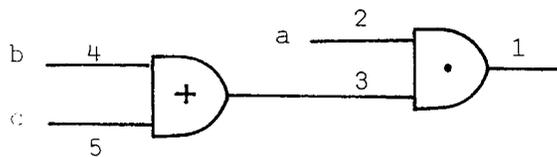


est équivalent au réseau :

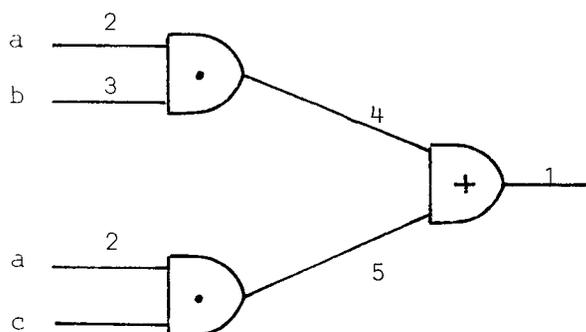


d) Application de la distributivité

Du fait que $a(b+c) = ab+ac$ on peut en déduire que le réseau :



est équivalent au réseau :



Les transformations a, b, d permettent de transformer tout réseau (pseudo) arborescent en un (pseudo) réseau (arborescent) 2 couches.

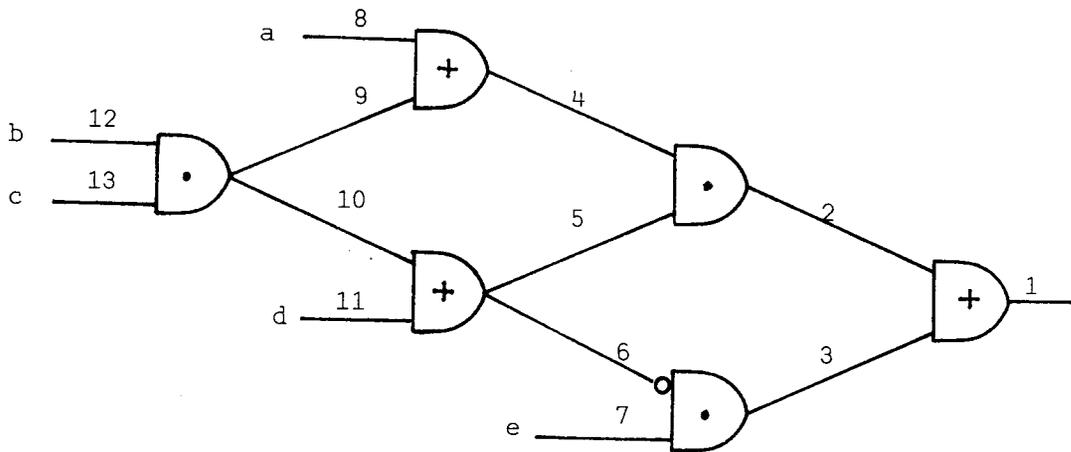
Comme la transformation c permet de ramener tout réseau à un réseau (pseudo) arborescent on en déduit :

Propriété

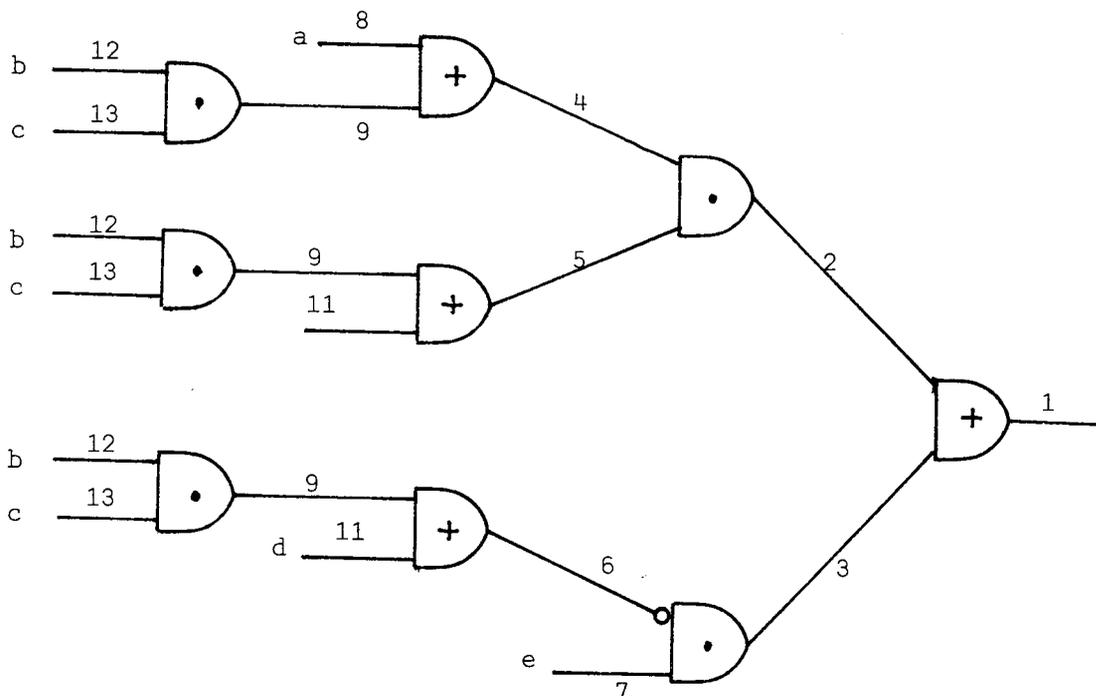
Par application des transformations a, b, c, d on peut transformer tout réseau en un réseau 2 couches équivalent.

EXEMPLE

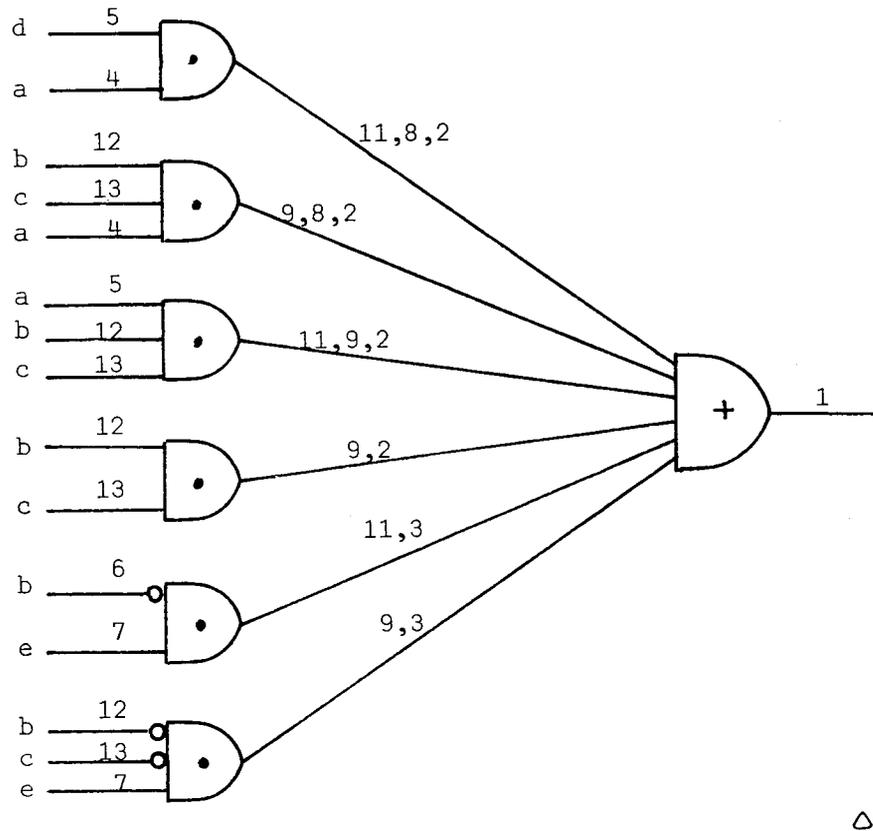
Soit le réseau :



il se transforme en un réseau (pseudo) arborescent :



Ce qui donne enfin le réseau 2 couches :



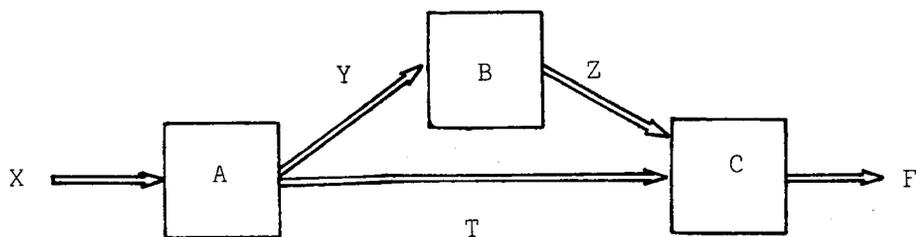
I - 2.3. Génération de test par sensibilisation de chemin

a) Cas du combinatoire

L'idée générale est de distinguer dans la génération de test 3 phases :

1. La manifestation de la panne
2. La propagation de la panne
3. La consistance

Soit un système constitué de 3 blocs interconnectés comme suit :



Supposons que la panne α soit dans le module B.

1. Manifester la panne α c'est trouver une entrée Y_0 telle que

$$Z_1 = Z(Y_0) \neq Z_\alpha(Y_0) = Z_2$$
2. Propager la panne, c.a.d. trouver T_0 tel que

$$F(T_0, Z_1) \neq F(T_0, Z_2)$$
3. La consistance c'est rechercher une entrée du système compatible avec la manifestation et la propagation de α : trouver Y_0 tel que

$$A(Y_0) = (Y_0, T_0).$$

EXEMPLE

Dans le cas de réseau de portes élémentaires avec comme pannes les collages, les règles de sensibilisation de chemin sont très simples : le long du chemin de propagation les portes OU doivent avoir les entrées à 0 et les portes ET les entrées à 1 Δ

La mise en oeuvre de la sensibilisation de chemin peut se faire soit par la différence booléennes (ou dérivée booléenne) soit par le D-algorithme. Il faut noter que les algorithmes de génération de test doivent considérer les chemins multiples car dans certains circuits il existe des pannes qui ne sont détectables que sur des chemins multiples et pas sur des chemins simples.

Le D-algorithme [ROTH] utilise une algèbre à 5 valeurs 0, 1, x, D, \bar{D} ; D signifiant que dans le circuit juste la valeur est 1 et que dans le circuit en panne la valeur est 0 (\bar{D} est l'inverse), la valeur x signifie indéterminée. L'algorithme consiste en une affectation des valeurs 0, 1, x, D, \bar{D} aux connexions du réseau qui soit cohérente avec les opérateurs et qui crée un chemin (ou un ensemble de chemins) de valeur D (ou \bar{D}) entre la panne et la sortie. Il a été montré récemment* qu'une algèbre à 9 valeurs était plus efficace, les 9 valeurs étant le carré de l'ensemble {0, 1, x} ce qui permet l'étude simultanée du comportement des circuits bons et en panne.

La différence booléenne [YAU] est une formalisation de la notion de propagation. Soit $f(x_1, x_2, \dots, x_n)$ une fonction. La dérivée booléenne de f par rapport à x_i est :

$$\Delta_{x_i} f = f(x_1, x_2, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \oplus f(x_1, x_2, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$$

*[LAKEAS], [MUTH], [CAILLAT]

Cette fonction indique dans quelles conditions une variation de la variable x_i se traduit par une variation de la sortie.

L'ensemble des tests de la panne x_i collée à 0 est donc donné par l'ensemble des vecteurs x tels que $x_i \Delta f(x) = 1$. Si x est collée à 1 les tests sont décrits par $\bar{x}_i \Delta f(x) = 1$.

La différence booléenne par rapport à une variable interne se définit de la même façon.

Si tous les chemins joignant la ligne i à la sortie passent par la ligne j , alors on peut décomposer la dérivée booléenne de f par rapport à i comme le produit de la dérivée booléenne de f par rapport à j , et la dérivée booléenne de j par rapport à i :

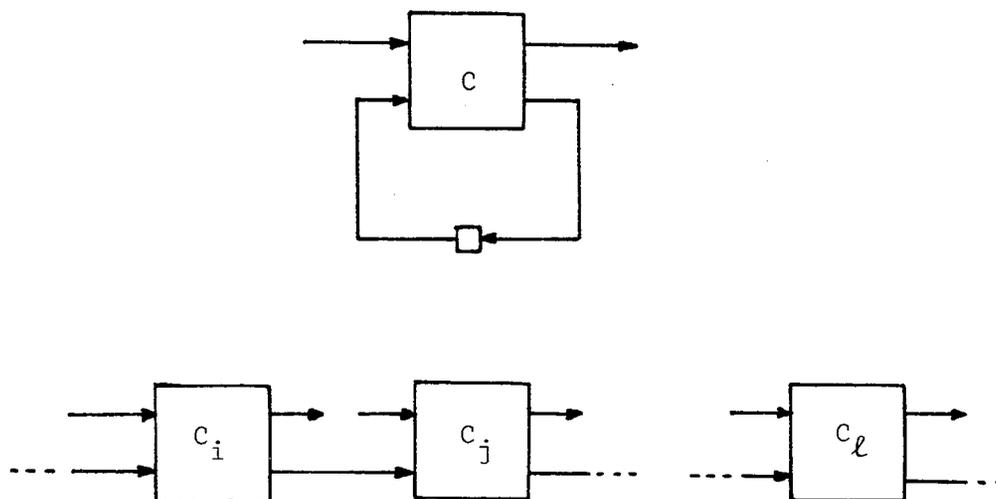
$$\Delta f = \Delta f \Delta j$$

$$i \quad j \quad i$$

Cette décomposition de la dérivée booléenne permet un calcul récursif des différentes dérivées booléennes donc des tests. Dans le cas de *sortance* multiple il faut, soit définir la dérivée booléenne multiple, qui est très lourde, soit se ramener à la définition.

b) Cas séquentiel

Si l'on a localisé les boucles de retard, alors un circuit séquentiel est analogue à un circuit itératif infini :



Ainsi, les méthodes de propagation peuvent être étendues au cas séquentiel [PUTZOLU]. Ce qui est nouveau c'est la question de l'initialisation et la transformation de pannes simples du système séquentiel en pannes multiples dans le réseau combinatoire.

L'analogie précédente donne une idée de l'accroissement de complexité entre le test du système combinatoire (boucles coupées) et celui du système séquentiel.

I - 2.4. Génération algébrique de tests

On peut décrire un réseau par une expression parenthésée qui est isomorphe à la représentation en arbre d'un réseau [CLEGG].

EXEMPLE

Au réseau précédent correspond l'expression :

$$E = (((b_{12} c_{13})_9 + a_8)_4 ((b_{12} c_{13})_9 + d_{11})_5)_2 + (((b_{12} c_{13})_9 + d_{11})_6 e_7)_3)_1 \Delta$$

On peut développer l'expression parenthésée en utilisant les propriétés booléennes en considérant la variable comme différente du fait de son indice. On obtient alors une somme de produits.

L'exemple précédent donne :

$$E = (a_4 d_5)_{2,8,11} + (a_4 b_{12} c_{13})_{2,8,5} + (a_5 b_{12} c_{13})_{2,9,11} \\ + (b_{12} c_{13})_{2,9} + (\bar{b}_6 e_7)_{3,11} + (\bar{b}_{12} \bar{c}_{13} e_7)_{3,9}$$

L'influence des pannes est la suivante : les pannes en indice des variables ont pour effet de remplacer celles-ci par 1 et les pannes en indice des parenthèses ont pour effet de remplacer tout le monôme par 0.

Ainsi l'expression correspondante à la panne $4_1 7_1$ est :

$$E'_{4_1 7_1} = (d_5)_{2,8,11} + (b_{12} c_{13})_{2,8,9} + (a_5 b_{12} c_{13})_{2,9,11} + (b_{12} c_{13})_{2,9} \\ + (\bar{b}_6)_{3,11} + (\bar{b}_{12} \bar{c}_{13})_{3,9}$$

Notons l'isomorphisme total entre cette expression et l'arbre 2 couches obtenu précédemment.

La fonction fautive correspondante est obtenue en enlevant les indices.

On obtient pour la panne $4_1 7_1$:

$$f_{4_1 7_1} = d + bc + abc + bc + \bar{b} + \bar{b}c = d + bc + \bar{b}$$

La fonction juste étant $f = ad + abc + bc + \bar{b}e + \bar{b}ce = ad + bc + \bar{b}e$, on en déduit l'ensemble de test de $4_1 7_1$:

$$T_{4_1 7_1} = f \oplus f_{4_1 7_1} = (ad + bc + \bar{b}e) \oplus (d + bc + \bar{b})$$

$$T_{4_1 7_1} = \overline{abe} + \overline{abcd} + \overline{aced} + \overline{bde}$$

On sait que tout test détectant toutes les pannes simples dans un arbre 2 couches, détecte aussi les pannes multiples; on peut donc appliquer ce résultat ici en testant l'absence de chaque monôme ainsi que sa croissance.

II - METHODES PROBABILISTES

II - 1. GENERATION ALEATOIRE DE TEST

On a vu que la génération algorithmique de test, pour tester une panne donnée, est très coûteuse. D'où l'idée d'utiliser des tests générés de façon aléatoire. Il se pose alors la question : comment mesurer la qualité de tels tests ? Ceci peut être fait de 2 façons :

1) Pour chaque test on calcule, par simulation, la liste des pannes détectées. Il faut donc une étude fine du système.

2) On cherche à priori quelle est la taille du test aléatoire nécessaire à tester un pourcentage donné des défauts. On peut ici se contenter d'une description grossière du système comme par exemple la description de l'automate sans avoir besoin de la description du réseau de portes.

Les mesures expérimentales montrent qu'en cas de génération aléatoire pure on obtient des tests qui, au début, détectent un nombre important de pannes, mais qui au bout d'un certain temps plafonnent. Ainsi ont été proposées des méthodes mixtes (aléatoires/algorithmiques) qui commencent par une génération aléatoire de test qui est suivie (quand la première phase devient inefficace) par une génération algorithmique des tests.

II - 2. TESTS DES PANNES ALEATOIRES

Les pannes traitées par les méthodes déterministes sont permanentes. Or il apparaît que de nombreuses pannes sont de nature intermittente. Le test de telles pannes ne peut se faire que de façon probabiliste.

La nature aléatoire des pannes peut avoir deux causes : soit que le défaut engendrant la panne est de nature aléatoire (Cf. bruit), soit que les circonstances de manifestation de la panne dépendent de questions temporelles que l'on ne maîtrise pas (Cf. système de synchronisation dans un multiprocesseur ou plus simplement création de courses critiques dans les systèmes asynchrones).

BIBLIOGRAPHIE

- F.C. CLEGG : "Use of spoof's in the analysis of faulty logic network", IEEE Trans. on Comp. March 1973.
- D.B. ARMSTRONG : "On finding a nearly minimal set of fault detection tests for combinational logic nets". IEEE Trans. on Comp. Feb. 1966.
- I. KOHAVI, Z. KOHAVI : "Detection of multiple faults in combinational logic networks". IEEE Trans. on Comp. June 1972.
- J.P. ROTH, W.C. BOURICIUS, P.R. SCHNEIDER : "Programmed algorithms to compute test to detect and distinguish between failures in logic circuits" IEEE. Trans. Electron. Comp. Oct. 1967.
- G.R. PUTZOLU, J.P. ROTH : "An heuristic algorithm for the testing of asynchronous circuits" IEEE Trans. Comp., June 1971
- E.J. Mc CLUSKEY, F.W. CLEGG : "Fault equivalence in combinational networks", IEEE Trans. Comp. November 1971.
- P. PIGNAL : "Une méthode de test automatique pour les ensembles logiques", L'Onde Electrique, Novembre-Décembre 1968.
- S.S. YAU, Y.S. TANG : "An efficient algorithm for generating complete tests sets for combinational logic circuits, IEEE Trans. Comp. Noverber 1971.
- R. DAVID, G. BLANCHET : "Sur la détection des pannes dans les circuits combinatoires par les séquences d'entrée aléatoires"
- J. CAILLAT : "Contribution au test des circuits intégrés logiques", Thèse de 3e cycle , Université de Grenoble, 1976.
- J.P. RICHARD : "Contribution à la détection des pannes dans les circuits logiques", Thèse de Docteur Ingénieur, Toulouse 1971.
- M. COURVOISIER : "Contribution à la détection et à la localisation des pannes dans les systèmes séquentiels", Thèse de 3e cycle, Université Paul Sabatier, Toulouse 1971.
- A. VERDILLON : "Pannes dans les réseaux acycliques", Thèse de 3e cycle, Université de Grenoble, 1972.

B.B. AKERS "Logic System for fault Test Generation", IEEE Trans. on Comp. juin 76.

P. MUTH "Nine-Valued Circuit Model for Test Generation "IEEE Trans. on Comp. juin 76.

C H A P I T R E I I

TEST OPTIMAL DES SYSTEMES SEQUENTIELS

- I - REPRESENTATION DES MACHINES NORMALES ET FAUSSES
 - I - 1. Définitions
 - I - 2. Table de test des transitions.

- II - SIMULATION DES FAUTES
 - II - 1. Simulation générale.
 - II - 2. Simulation pour la détection.
 - II - 3. Simulation pour la localisation.

- III- PROCEDURES POUR SEQUENCES DE DETECTION OPTIMALES
 - III- 1. Définition, d-dominance.
 - III- 2. Procédures.

- IV - PROCEDURES POUR SEQUENCES DE LOCALISATION OPTIMALES
 - IV - 1. Définition, ℓ -dominance.
 - IV - 2. Procédures.

- V - EXTENSIONS, PANNES DANS L'INITIALISATION et MACHINES ASYNCHRONES
 - V - 1. Pannes affectant l'initialisation.
 - V - 2. Circuits asynchrones.

INTRODUCTION

Le test de circuits séquentiels est un important problème à la fois pratique et théorique. En pratique, ce problème est souvent résolu en coupant les boucles de façon que le circuit devienne acyclique. Mais le coût d'une telle logique supplémentaire est quelquefois trop important spécialement quand un processus complet est implémenté sur un composant. L'extension de procédures pour circuits combinatoires par exemple D-algorithme [1] aux circuits séquentiels, conduit à de gros programmes qui en général ne détectent pas toutes les pannes. La méthode de Poage [2] résoud le problème en théorie mais en pratique cette méthode est limitée aux circuits avec petits ensembles d'états, et petits ensembles de pannes.

Suivant l'approche de Poage et Mc Cluskey, nous développons de nouvelles procédures utilisant une représentation très compacte du comportement des circuits normal et faux : la table de test. Les pannes considérées ici conservent au circuit son caractère synchrone et n'affectent pas l'initialisation.

Des méthodes de simulation sont présentées pour les problèmes de détection et de localisation. Utilisant la simulation, deux procédures sont décrites qui donnent des séquences de test optimum. La première fait une extension de la table de test en un arbre, à chaque étape l'ensemble des pannes et l'ensemble des séquences sont réduits en utilisant les relations de dominance parmi les pannes et parmi les séquences d'entrées. La seconde procédure a deux étapes. La première étape construit un graphe de détection dont les arcs sont des séquences d'entrées et correspondent au comportement périodique des machines normales et fausses. Dans la deuxième étape, le graphe de détection est étendu en un arbre comme pour la première procédure.

Afin de résoudre le problème de diagnostic, le concept

d'expression de localisation est défini. L'expression de localisation permet d'avoir des procédures analogues au cas de la détection sans avoir besoin de construire les C_2^n produits de machines fausses habituellement utilisées (n est le nombre de pannes).

Enfin, des exemples montrent comment des procédures analogues peuvent être développées pour des pannes qui affectent la remise en état initial et les circuits asynchrones.

I - REPRESENTATION DES MACHINES NORMALES ET FAUSSES

I - 1. Définitions

- Soit $M = (I, O, Q, \delta, \lambda)$ une machine séquentielle synchrone où I, O, Q sont respectivement les ensembles d'entrée, de sortie et d'état;

$\delta : I \times Q \rightarrow Q$ est la fonction de l'état suivant
 $\lambda : I \times Q \rightarrow Q$ la fonction de sortie.

La machine est supposée avoir un état initial q_0

- Soit $F = \{f_1, f_2, \dots, f_m\}$ un ensemble de pannes qui affectent la machine M . Les restrictions imposées sur les pannes sont les suivantes : elles sont permanentes, n'affectent pas l'état initial et les machines fausses restent synchrones. On tient compte des pannes qui augmentent le nombre des états en définissant l'ensemble Q des états tel que $Q = 2^{n+n'}$ où n est le nombre de lignes de réaction de la machine juste M , et n' est le nombre de nouvelles lignes de réaction à cause de certaines pannes. La panne f transforme la machine normale M en une machine fausse $M_f = (I, O, Q, \delta_f, \lambda_f)$.

δ_f et λ_f sont les fonctions de transitions fausses du fait de la panne f , La panne nulle f_1 est un élément de F , elle correspond à la machine normale : $M_{f_1} = M$.

Les comportements des machines fausses (et normales) peuvent être décrits par la machine produit $M_F = (I, O^m, Q^m, \delta_F, \lambda_F)$

$$\text{où } Q^m = \overbrace{Q \times Q \times \dots \times Q}^{m \text{ fois}}$$

$\delta_F : I \times Q^m \rightarrow Q^m$ est défini par :

$$\delta_F(i, [q_1, \dots, q_m]) = [\delta_{f_1}(i, q_1), \dots, \delta_{f_m}(i, q_m)]$$

avec $i \in I, q_j \in Q$

$\lambda_F : I \times Q^m \rightarrow O^m$ est défini par :

$$\lambda_F(i, [q_1, \dots, q_m]) = [\lambda_{f_1}(i, q_1), \dots, \lambda_{f_m}(i, q_m)]$$

avec $i \in I, q_j \in Q$.

I - 2. Table de test des transitions

La machine produit M_F décrit en parallèle le comportement de chaque machine fautive (et normale), mais la complexité de cette machine croît très vite avec le nombre de pannes.

En fait le comportement des machines fautes peut être décrit par deux fonctions beaucoup plus simples : la fonction test de transition et la fonction test de sortie.

Définition

La fonction test de transition, $\Delta : I \times Q \rightarrow Q^m$ est définie par : $\Delta(i, q) = [\delta_{f_1}(i, q), \delta_{f_2}(i, q), \dots, \delta_{f_m}(i, q)]$ avec $i \in I$ et $q \in Q \circ$

Définition

La fonction test de sortie, $\Lambda : I \times Q \rightarrow O^m$ est définie par $\Lambda(i, q) = [\lambda_{f_1}(i, q), \lambda_{f_2}(i, q), \dots, \lambda_{f_m}(i, q)]$ avec $i \in I$ et $q \in Q \circ$

Au lieu de représenter les fonctions Δ et Λ comme vecteurs elles peuvent être représentées par une notation indicée. Par exemple, si l'ensemble des pannes est $F = \{o, a, b, c\}$ (o est la panne nulle) et que $\Delta(i, q) = (A, A, A, B)$ avec q, A, B états de Q , la notation indicée est : $A_{oab} B_c$. La notation indicée peut être simplifiée en ne marquant pas les pannes qui correspondent au comportement normal. L'exemple précédent donne donc : $\Delta(i, q) = AB_c$. Les fonctions Δ et Λ représentées par la notation indicée simplifiée forment la table de test.

EXEMPLE : Soit $F = \{o, a, b, c\}$ l'ensemble des pannes. La machine normale M_o et les machines fautes M_a, M_c , sont décrites par les tables suivantes :

Q \ I	0	1
A	A,1	B,0
B	A,0	B,1

M_o

Q \ I	0	1
A	A,1	B,0
B	B,0	A,1

M_a

Q \ I	0	1
A	A,1	A,0
B	A,0	A,1

M_b

Q \ I	0	1
A	B,0	B,0
B	A,0	A,1

M_c

La table test est :

Q \ I	0	1
A	$AB_c, 10_c$	$BA_b, 0$
B	$AB_a, 0$	$BA_{abc}, 1$

L'interprétation de $\Delta(o, A) = AB_c$ et $(0, A) = 10_c$ est la suivante : si les machines normales et fausses sont dans l'état A et si l'entrée est 0 alors l'état suivant sera A pour toutes les machines dont la panne correspondante n'est pas c et pour la machine avec faute c l'état suivant sera B. La sortie est 1 sauf pour la machine avec panne c qui est détectée par la sortie 0. Notons que la table de test peut être directement obtenue par simulation des listes de fautes (Cf. Chap.IV), ou par la construction d'un ensemble de test de chaque panne Δ

II - SIMULATION DES PANNES

La table de test est utilisée afin de simuler les machines normales et avec pannes, pour une séquence d'entrée spécifique. Des règles spéciales de simplification sont utilisées suivant que la simulation a trait avec le problème de détection ou celui de localisation.

II - 1. Simulation générale

Le théorème suivant décrit comment utiliser les fonctions Δ et Λ afin de simuler les fonctions δ_F et λ_F .

La restriction de F à $F' \subseteq F$ induit une restriction de Δ à $\Delta^{F'}$.

Par exemple, si $F = \{0, a, b, c, \}$, $F' = \{0, b\}$, $F'' = \{a, c\}$ et $\Delta(1, B) = BA_{ab}$, alors $\Delta^{F'}(1, B) = BA_b$ et $\Delta^{F''}(1, B) = B_c A_a$ (0 indique la panne nulle : $M_0 = M$).

Soit q_F un état produit ($q_F \in Q^m$), sa représentation indicée est $q_F = (q_F^1, q_F^2, \dots, q_F^p)$ avec $q^i \in Q$ et $F_i \subseteq F$. F_i est l'ensemble des pannes qui mettent la machine dans l'état q_i .

THEOREME

$$\delta_F(x, q_F) = \bigcup_{i=1}^p \Delta_{F_i}(x, q^i) \text{ et}$$

$$\lambda_f(x, q_F) = \bigcup_{i=1}^p \Lambda_{F_i}(x, q^i) \quad (x \in I, q^i \in Q)$$

Dem : Soit $F_k = \{f_{k_j}\} j = 1, \dots, |F_k|$ $f_{k_j} \in F$

$$\text{alors } \delta_F(x, q_F) = \bigcup_k [\bigcup_j \delta_{f_{k_j}}(x, q^k)] \quad (1) \text{ avec}$$

$$k = 1, 2, \dots, p \text{ et } j = 1, 2, \dots, |F_i|.$$

$$\text{Mais, } \delta_f = \Delta_f \text{ entraine que } \delta_{F_k}(x, q^k) = \bigcup_j \delta_{f_{k_j}}(x, q^k) \quad (2).$$

La substitution de (2) dans (1) donne $\delta_F(x, q_F) = \bigcup_k \Delta_{F_k}(x, q^k)$ \square

La preuve pour la fonction de sortie est la même.

L'application itérative du théorème précédent donne une procédure qui décrit le comportement normal et faux des différentes machines.

EXEMPLE Considérons la table de test donnée précédemment et supposons que l'état initial est A. Si la séquence d'entrée est 0110, le comportement des machines normales et fausses est le suivant :

I	Q^m	O^m	
0	AB_c	10_c	
1	BA_{bc}	01_c	
1	BA_{ab}	10_{bc}	
0	A	01_{ab}	Δ

II - 2. Simulation pour le problème de détection

Pour le problème de détection, la procédure basée sur le théorème peut être simplifiée. Quand une panne f est détectée, il n'est plus nécessaire de la simuler et les fonctions et sont alors simplifiées en Δ_{F-f} et Λ_{F-f} .

EXEMPLE : La précédente simulation se simplifie en :

I	Q^m	O^m	détection	
0	A	10_c	c	
1	BA_b	0		
1	BA_a	10_b	b	
0	A	01_a	a	Δ

II - 3. Simulation pour le problème de localisation

Comme précédemment, quand une panne est localisée, il n'est plus nécessaire de la simuler. La différence avec la

détection est que, à une certaine étape de la simulation, certaines pannes peuvent être détectées mais sans qu'elles puissent être distinguées. Supposons que la simulation d'une séquence de deux entrées i_1 et i_2 ($i_j \in I$) donne les sorties suivantes 12_a 3_{bc} et 21_b . L'interprétation est la suivante : si la panne b est présente la séquence de sortie est (3, 1), la panne c donne elle la séquence de sortie (3,2) et la panne a donne (2, 2) donc les pannes a, b, c sont localisées.

Une expression de localisation dans F peut être représentée par un sous-ensemble de 2^F . Par exemple, l'expression de localisation associée à la sortie 12_a 3_{bc} est $\{\{a\}, \{b,c\}\}$ et peut être dénotée $a + bc$, ce qui signifie que la panne a est localisée, que les pannes b et c sont détectées mais sont indistinguables (pour le moment).

Propriété : Si la simulation d'une séquence d'entrées s_1 donne une expression de localisation ℓ_1 et positionne la machine dans l'état q_F , si la simulation d'une séquence s_2 (avec état initial q_F) donne une expression de localisation ℓ_2 , alors l'expression de localisation de $s_1 s_2$ est $\ell_1 + \ell_2$.

Règle de simplification : si $x \wedge y = \emptyset$ alors $xz = x+y+z$

Exemple : Si la séquence de sortie est 12_{ab} , 31_{bc} alors l'expression de localisation est $ab + bc$. Mais les séquences de sortie pour les machines M_a , M_b , M_c sont respectivement (2,3), (2,1), (1,1) dont a, b, c sont localisés et $ab + bc = a + b + c$. Δ

Définition : Une expression de localisation réduite est une expression de localisation pour laquelle la règle de simplification ne peut plus être appliquée. \circ

Propriété : Une expression de localisation a une expression de localisation réduite qui est unique.

Démonstration En fait, une expression de localisation réduite correspond à une partition sur l'ensemble des

pannes. La règle de simplification consiste à faire l'intersection de 2 partitions d'où l'unicité.

EXEMPLE : Considérons la table de test donnée en paragraphe 1.2. Si l'état initial est A et la séquence d'entrée 11101, la simulation est alors :

I	Q^m	O^m	expression de localisation
1	BA_b	0	
1	BA_{ac}	10_b	b
1	B	10_{ac}	b+ac
0	AB_a	0	b+ac
1	B	01_a	b+ac+a = a+b+c

III - PROCEDURES D'OBTENTION DE SEQUENCES DE DETECTION OPTIMALE

III - 1. Définitions, propriétés de d-dominance

Etendons les fonctions de transition et de sortie (δ et λ) au monoïde des séquences.

Soit I^* et O^* les ensembles de séquences d'entrées et de sortie. La fonction de transition et la fonction de sortie peuvent être étendues à I^* . Par exemple $\delta^* : I^* \times Q \rightarrow Q$ est définie par $\delta^*(si, q) = \delta(i, \delta^*(s, q))$ et $\delta^*(i, q) = \delta(i, q)$ avec $i \in I^*$, $i \in I$ et $q \in Q$; q_0 étant l'état initial $\delta^*(s, q_0)$ est noté $\delta^*(s)$. La fonction $\delta_{F'}^*$ ($F' \subseteq F$) est définie de façon similaire $\delta_{F'}^*(si, q) = \delta_{F'}(i, \delta_{F'}^*(s, q))$ et $\delta_{F'}^*(i, q) = \delta_{F'}(i, q)$.

La fonction de sortie λ est étendue à $\bar{\lambda} : I^* \times Q \rightarrow O^*$ qui est définie par $\bar{\lambda}(si, q) = \bar{\lambda}(s, q) \cdot \lambda(i, \delta^*(s, q))$ et $\bar{\lambda}(i, q) = \lambda(i, q)$ avec $s \in I^*$, $i \in I$ et $q \in Q$. $\bar{\lambda}(s, q_0)$ est dénoté $\bar{\lambda}(s)$.

La fonction de sortie pour toutes les machines $\lambda_{F'}$, est étendue de façon similaire à : $\bar{\lambda}_{F'}$.

Soit F' un sous-ensemble de pannes ($F' \subset F$) et s une séquence d'entrées ($s \in I^*$).

Définition : La séquence d'entrées s détecte F' si $\bar{\lambda}(s) \neq \bar{\lambda}_f(s)$ pour toute panne f de F' .

Définition : L'ensemble des pannes détectées par s est noté D_s ($D_s = \{f \in F' \mid \bar{\lambda}(s) \neq \bar{\lambda}_f(s)\}$).

Définition : Soit S et S' 2 ensembles de séquences; S' est un préfixe de S (noté $S' \leq S$) si pour toute séquence s de S il existe une séquence s' de S' et s'' de I^* (et si $\forall s' \in S' \exists \Delta \in S$ et $\Delta'' \in I^*$) telles que $s = s' \Delta''$.

Définition : S' est un strict préfixe de S (noté $S' \leftarrow S$) si S' est un préfixe de S et $S \neq S'$.

Définition : La séquence s détecte de façon minimale l'ensemble de pannes F' si s détecte F' et si tout strict préfixe s' de s ne détecte pas F' .

Définition : La séquence s détecte de façon optimale l'ensemble des pannes F' si s détecte F' et toute séquence s' détectant F' est plus longue que s ou de la même longueur.

L'ensemble des séquences qui détectent F' est dénoté $S_{F'}$.

L'ensemble des séquences qui détectent F' de façon minimale est noté $S_{F'}^m$.

Les relations de dominance entre pannes et entre séquences peuvent être définies d'une part pour la détection et d'autre part pour la localisation.

Définissons dans ce chapitre la d-dominance (dominance pour la détection).

Définition : La panne f d-domine f' (dénoté $f \succ f'$) si l'ensemble de détection de f' est inclus dans celui de f ($S_{f'} \subseteq S_f$).

Définition : La panne f est détectable si l'ensemble de détection de f n'est pas vide ($S_f \neq \emptyset$).

La relation de d-dominance est un pré-ordre sur F (réflexive, transitive) qui induit une relation d'équivalence : la d-équivalence.

Définition : Si l'ensemble de détection de f est le même que celui de f' alors f et f' sont d-équivalents.

La relation de d-dominance induit un ordre partiel sur les classes de d-équivalence.

Propriétés : 1) $S_{f'} \cap I^* = S_{f'}$
 2) $S_{f'} \cap I^* = S_{f'}$

Démonstration : 1) Si $s \in S_{f'}$, alors $\forall s' \in I^*, s$ s' détecte aussi f' donc $s \in S_{f'}$.
 2) $S_{f'} \cap I^* \subseteq S_{f'}$, car si $s \in S_{f'} \cap I^*$, $s \in S_{f'}$, ($s' \in I^*$)
 $\forall s \in S_{f'} \cap I^*, \exists s' \in S_{f'}$ et $s'' \in I^* : s = s' s''$ donc
 $S_{f'} \cap I^* \subseteq S_{f'}$.

THEOREME : Une panne f d-domine f' si et seulement si $S_{f'}$ est un préfixe de S_f ; f d-domine strictement f' si et seulement si $S_{f'}$ est un strict préfixe de S_f .

Démonstration : $f \succ f' \iff S_{f'} \subseteq S_f$, $S_f \subset S_{f'} \iff S_f \prec S_{f'}$.

THEOREME : Soit S préfixe de S_F , soit S' l'ensemble des séquences de S tronquées du dernier élément ($S' = \{s' \mid \exists s \in S, i \in I, s' = s[1:i]\}$). Si f est détectée par toute séquence de S et qu'il existe f' détectée par aucune séquence de S' alors : $S_F = S_{F-f}$

Démonstration : Soit s une séquence détectant toutes les pannes ($s \in S_F$) alors s détecte f avant, ou en même temps que f' .

Ce théorème permet de réduire l'ensemble des pannes.

Ce concept de d -dominance peut être utilisé pour la séquence.

Définition : Une séquence s_1 d -domine s_2 (notée $s_1 \leq s_2$) si $D_{s_1} \supseteq D_{s_2}$ et $|s_1| \leq |s_2|$ et $\delta_{F_1}^*(s_1) = \delta_{F_1}^*(s_2)$ avec $F_1 = F - D_{s_1}$

Définition : Une séquence s_1 d -domine strictement s_2 (noté $s_1 < s_2$) si s_1 d -domine s_2 et $|s_1| < |s_2|$.

Définition : Deux séquences s_1 et s_2 sont d -équivalentes si $D_{s_1} = D_{s_2}$, $|s_1| = |s_2|$ et $\delta_{F_1}^*(s_1) = \delta_{F_1}^*(s_2)$.

Notons que dans ces définitions de d -dominance on tient compte de l'état des machines fausses.

EXEMPLE : Si on simule les 2 séquences $s_2 = 100$ et $s_1 = 110$ de l'automate, on obtient comme état final : pour s_1 , A avec les pannes détectées abc ; pour s_2 , AB_c avec les pannes détectées ab ; donc s_1 d -domine s_2 .

Soit : s, s_1, s_2 des séquences.

- THEOREME
- 1) Si s_1 d-domine s_2 alors s_1s d-domine s_2s . Si s_1 et s_2 sont d-équivalent, alors s_1s et s_2s sont d-équivalent.
 - 2) Si s_1 d-domine s_2 , $|s_1| = |s_2|$ et s_2s détecte de façon optimale D , alors s_1s détecte F de façon optimale.
 - 3) Si s_1 d-domine strictement s_2 alors s_2s ne détecte pas F de façon optimale.

Démonstration : 1) $D_{s_1s} = D_{s_1} \cup D'$ ($D' \cap D_{s_1} = \emptyset$) $f \in D' \Rightarrow f \in F - D_{s_2}$,
 alors $\delta_f^*(s_1) = \delta_f^*(s_2)$ et $\bar{\lambda}_f(s_1s) \neq \bar{\lambda}(s_1s)$ implique
 $\bar{\lambda}_f(s_2s) \neq \bar{\lambda}(s_2s)$.

- 2) Si s_2s ne détecte pas de façon optimale F , il existe $s' (|s'| < |s|)$ telle que s_2s' détecte F . Dans ce cas, s_1s' détecte F , $|s_1s'| < |s_1s|$ et s_1s ne peut pas être optimal.
- 3) Si s_2s détecte F alors s_1s détecte F (Cf.1)) et s_2s n'est pas optimal parce que $|s_1s| < |s_2s|$.

Ce théorème démontre que pour le test il suffit d'utiliser des séquences dominantes.

III - 2. Procédure pour obtenir des séquences de détection optimales -

Afin de construire des séquences de détection optimales, un arbre des successeurs est construit. Une branche de l'arbre correspond à une transition de la machine M . La marque d'un noeud n est $\rho(n) = (q_n, D_n)$. D_n est l'ensemble de pannes détectées quand le noeud n est atteint. Soit F' le sous-ensemble des pannes non détectées en n ($F' = F - D_n$), q_n est l'état du produit de machines M_F .

Il y a bijection entre les noeuds de l'arbre des successeurs et les séquences d'entrée. Notons s_n la séquence d'entrées qui correspondent au noeud n et n_s le noeud qui correspond à la séquence s . Il est facile de vérifier que $D_{s_n} = D_{s_n}$, $q_n = \delta_{F'}^*(s_n)$ et $|n| : |s_n|$.

Définitions et résultats pour les séquences induisent des définitions et résultats similaires pour les noeuds de l'arbre des successeurs. Un noeud n détecte F' si $F' \subseteq D_n$; il correspond à s_n qui détecte F' . Soit N un ensemble de noeuds de l'arbre des successeurs. S_N dénote l'ensemble correspondant de séquences d'entrées; si S est un ensemble de séquences, N_S est l'ensemble correspondant de noeuds.

N' est un prédécesseur de N , dénoté $N' \leq N$ si $S_{N'} \leq S_N$.

Si $S_{N'} < S_N$, est un prédécesseur strict de N dénoté $N' < N$.

Un noeud n détecte F' de façon minimale si s_n détecte F' de façon minimale, si s_n détecte F' de façon optimale alors n détecte F' de façon optimale.

L'ensemble des noeuds qui détectent F' est noté $F_{F'}$, l'ensemble des noeuds qui détectent de façon minimale F' et N'_F .

Propriété : $N_{F'} = N_{SF'}$ et $N'_F = N'_{SF'}$

Propriété : $f \geq f'$ ($f > f'$) si et seulement si $N_f \supseteq N_{f'}$ ($N_f \supset N_{f'}$).

THEOREME : $f \geq f' \iff N_f \leq N_{f'}$, et
 $f > f' \iff N_f < N_{f'}$.

Démonstration : $f \geq f' \iff S_f \leq S_{f'} \iff N_f \leq N_{f'}$.

THEOREME : Soit $N \leq N_F$, soit N' l'ensemble des noeuds prédécesseurs de N . Si $f \in \bigcap_{n \in N} D_n$ et $\exists f' \notin \bigcup_{n \in N'} D_n$ alors:

$$N_F = N_F - f$$

Définition : Le noeud n_1 d-domine le noeud n_2 (noté $n_1 \succcurlyeq n_2$) si la séquence correspondant à n_1 d-domine celle correspondant à n_2 ($s_{n_1} \succcurlyeq s_{n_2}$)

On note $n_1 > n_2$ si le noeud n_1 d-domine strictement n_2 .

THEOREME : [1] soit $n_i = n_{s_i}$ et $n'_i = n_{s_i s}$ ($i \in \{0,1\}$).

- 1) Si $n_1 \succcurlyeq n_2$ et n_1, n_2 prédécesseur respectivement de n'_1 et n'_2 alors $n'_1 \succcurlyeq n'_2$.
- 2) Si $n_1 \succcurlyeq n_2$ $|n_1| = |n_2|$ et n'_2 détecte de façon optimale F' alors n'_1 détecte F de façon optimale.
- 3) Si $n_1 \succcurlyeq n_2$ alors n'_2 ne détecte pas F de façon optimale.

Le théorème précédent est utilisé pour simplifier la construction de l'arbre successeur en enlevant des noeuds qui ne peuvent pas conduire à un test optimum. Le corollaire précédent est utilisé pour réduire l'ensemble des pannes F .

Notons qu'une simplification des noeuds peut entraîner une simplification des pannes et vice versa. De la même façon que dans le problème de couverture classique, une simplification des lignes peut entraîner une simplification de colonne et vice versa.

Un arbre successeur est construit en faisant une expansion des noeuds terminaux. Les marques (état produit, pannes détectées) des nouveaux noeuds sont obtenues par la procédure de simulation décrite en section 2.2. S'il existe 2

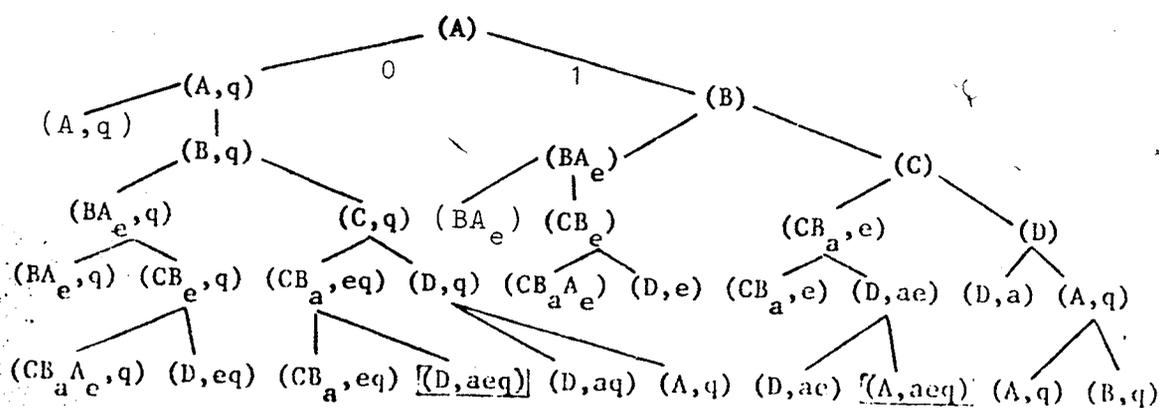
noeuds n et n' tels que $n > n'$, alors n est enlevé de l'arbre ($n \geq n'$ si une seule solution optimale est recherchée). A chaque pas, l'ensemble des noeuds terminaux est un prédécesseur de N_F , par suite l'ensemble des pannes sûrement détectées par $N(\bigcap_{n \in N_n} D_n)$ est calculé afin de réduire l'ensemble des pannes F . La procédure est arrêtée quand aucun noeud ne peut être ajouté et l'arbre qui reste décrit les séquences optimales de test.

EXEMPLE : Considérons l'exemple donné par Poage et Mc.Cluskey [1].

Q \ I	0	1
A	A ,01 _q	B ,0
B	BA _e ,0	C ,0
C	CB _a ,01 _e	D ,1
D	D ,10 _a	A ,01 _{dg}

Supposons que A soit l'état initial.

La construction de l'arbre donne :



Les 2 séquences optimales sont $s_1 = (01101)$ et $s_2 = (11011)$.

Le noeud marqué $(CB_a A_e)$ (4ème ligne) n'est pas

car il existe un noeud (CB_a, e) (sur la même ligne) donc $(CB_a Ae) \leq (CB_a, e)$.

Procédure 2. Dans cette procédure on construit d'abord un graphe de détection réduit qui correspond au comportement périodique des machines justes et fausses, pour le problème des séquences de détection optimales.

Etape 1 Un arbre des successeurs est construit mais les règles de simplification sont différentes de celles de la procédure 1. Un noeud n_1 est inactif si une des 2 propriétés suivantes est vraie.

$$P_1 : D_{n_1} = F$$

$$P_2 : \text{il y a un autre noeud } n_2 \text{ tel que } D_{n_1} \supseteq D_{n_2} \text{ et } q_{n_1} = q_{n_2} \text{ pour les pannes de } F - D_{n_1}$$

Les seuls noeuds qui sont étendus sont les noeuds actifs (pas inactifs).

Soit N l'ensemble des noeuds actifs et N' les noeuds inactifs.

Si $f' \in \bigcap_{n \in N} D_n$ et s'il existe f (détectable) qui vérifie $f \notin \bigcup_{n \in N' \cup N} D_n$ alors $\overset{0}{N}_f < \overset{0}{N}_f$ donc $S_F = S_F - f'$ et f' peut être enlevé de F .

L'arbre de détection peut être représenté par un graphe : le graphe de détection. Chaque noeud du graphe est un noeud de l'arbre, les branches sont marquées avec l'entrée correspondant à la transition ainsi que l'ensemble de pannes détectées par cette transition. Si une transition détecte une panne le prochain noeud est appelé noeud de détection.

A partir du graphe de détection on peut construire

le graphe réduit de détection dont les noeuds sont les noeuds de détection et le noeud initial du graphe de détection. Les branches entre deux noeuds sont les séquences minimales entre les noeuds correspondants du graphe de détection.

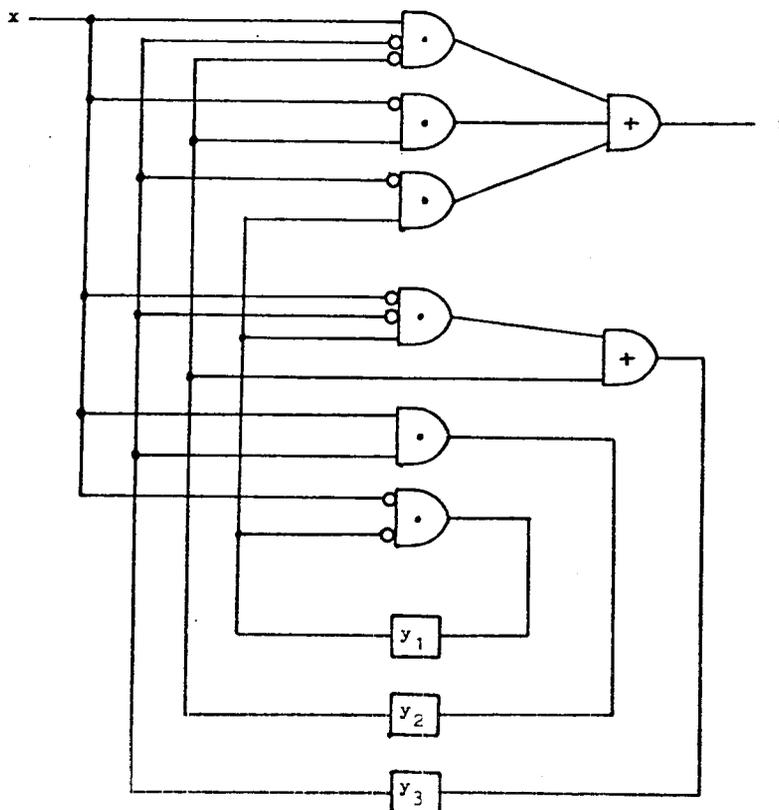
Etape 2

Le graphe de détection réduit est étendu à un arbre de successeur. La différence entre cette étape et la procédure 1 est que maintenant les branches correspondent à des séquences. Les règles de simplification de la procédure 1 sont utilisées. Les noeuds terminaux donnent les séquences de détection optimales.

Note

Au lieu de construire un graphe de détection réduit, l'arbre réduit de détection peut être obtenu directement à partir de l'arbre de détection.

EXEMPLE : Soit le circuit suivant :

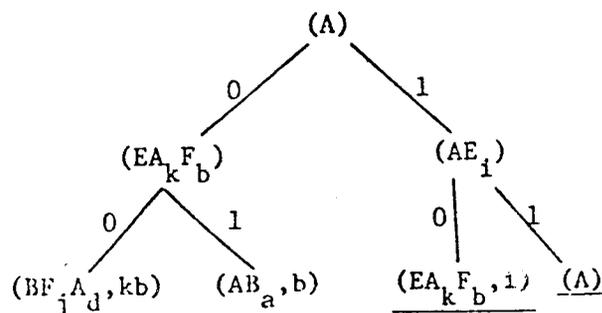


Si l'on considère les collages dans la partie du circuit élaborant la fonction de transition, on obtient la table de test suivante :

Q \ I	0	1
A	$EA_k F_b, 0$	$AE_i, 1$
B	$EA_k H_f, 0$	$DH_i A_h, 0$
C	$FB_k G E_e, 0$	$CG_i B D_e, 1$
D	$FB_k E_e, 0$	$BF_i A_e, 1$
E	$BF_j A_d, 1$	$AB_a, 1$
F	$AE_j D B_c, 0$	$DA_h, 0$
G	$BF_j C A_e, 0$	$CB_h D_e, 1$
H	$BF_j, 1$	$BA_e, 1$

L'ensemble des pannes est $\{a, b, c, d, e, f, g, h, i, j, k\}$. L'état initial est A.

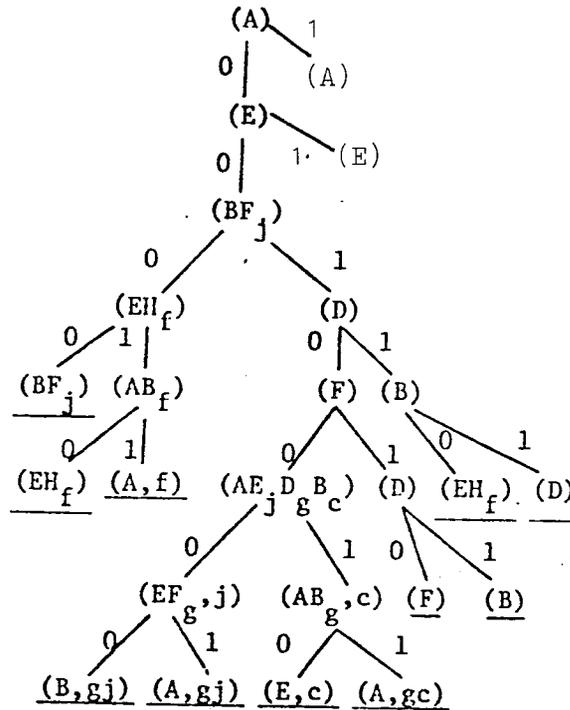
En construisant l'arbre de détection on commence par construire l'arbre suivant :



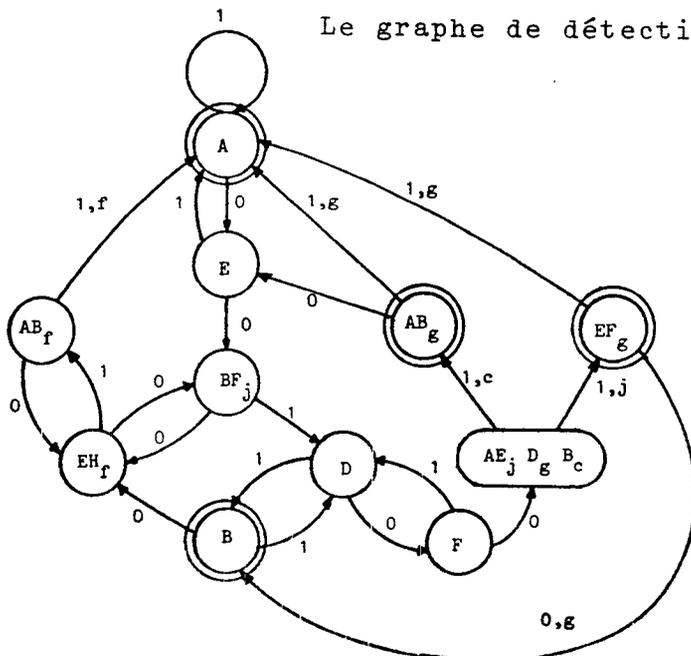
Les noeuds inactifs sont soulignés. A cette étape la panne b peut être le noeud $(EA_k F_b, i)$ (3ème ligne) test inactif à cause de la propriété P_2 et du noeud $(EA_k F_b)$ (2ème ligne).

Notons qu'il n'y a pas de dominance entre les noeuds (AE_k^F, i) et (EA_k^F, i) donc dans la procédure 1 le noeud (EA_k^F, i) aurait été étendu.

La panne b peut être enlevée car pour tester les pannes autres que i on est obligé de passer par un des noeuds actifs de cette étape, et donc de tester la panne b.



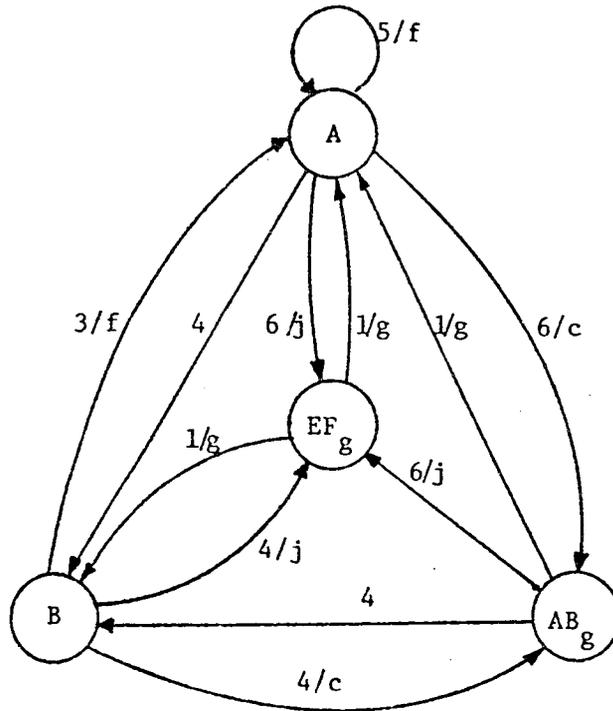
Le graphe de détection est :



le cercle.

Les noeuds de détection sont marqués avec un double cercle

Le graphe de détection réduit est :



Les arcs sont marqués par la longueur de la séquence correspondante et par des pannes détectées. L'arc 6/c entre (A) et (A B_g) correspond à la séquence (A) (Z) (BF_j) (D) (F) (AE_jD_gB_c) (AB_g) sur le graphe de détection.

Une séquence de test optimale est $s = (0010000011000001)$ la longueur de la séquence est 16.

N.B. La complexité de la procédure 2 (ou 1) est difficile à évaluer précisément. En effet si e est le nombre d'entrées du circuit et ℓ la longueur d'une séquence optimale, l'arbre de détection peut contenir dans le pire des cas : e^ℓ noeuds. L'exemple précédent montre que le nombre de noeuds du graphe de détection est 25 qui est de l'ordre de grandeur du nombre d'états de la machine (8) et est très éloigné du pire cas : 2^{16}

Notons que le problème de la génération de séquence

de test optimal se ramène à un problème type "voyageur de commerce" qui est un problème N.P. complet.

L'amélioration de ces procédures par rapport à la méthode de Poage vient d'une part d'une représentation plus compacte de l'information, d'autre part d'un meilleur usage des simplifications, en particulier sur l'ensemble des pannes.

IV - PROCEDURES POUR OBTENIR DES SEQUENCES DE LOCALISATION OPTIMALES

L'approche générale [3] d'extension des procédures de détection à la localisation des pannes consiste à considérer pas seulement tous les produits de la machine sans panne pour chaque machine fausse, mais aussi tous les produits de machines fausses deux par deux. L'ensemble des machines produit est alors un ensemble de C_m^2 machines avec $m = |F|$.

Les procédures présentées ici ne nécessitent pas la construction des C_m^2 machines. En effet, le concept d'expression de localisation conduit à de simples extensions des procédures pour la détection.

IV - 1. Définitions - Propriété de la ℓ -dominance

Soit F' un sous-ensemble de pannes de F .

Définition : Une séquence s localise F' dans F si s détecte F' et $\bar{\lambda}_f(s) \neq \bar{\lambda}_{f'}(s)$ avec $f \in F$, $f' \in F'$ et $f \neq f'$.

Définition : La séquence s localise de façon minimale F' si s localise F' et pour chaque séquence $s' \neq s$ qui localise F' $|s| \leq |s'|$.

Définition : La séquence s localise de façon optimale F' si s localise F' et si chaque strict préfixe de s ne localise pas F' .

L'ensemble des séquences qui localisent F' est noté $\mathcal{S}_{F'}$. L'ensemble des séquences qui localisent F' de façon minimale est dénoté $\mathcal{S}_{F'}$.

Définition : Une panne f ℓ -domine f' (dénoté $f > f'$) si $\mathcal{S}_f \supseteq \mathcal{S}_{f'}$.

Définition : Une panne f est localisable si $\mathcal{S}_f \neq \emptyset$.

Définition : Deux pannes f et f' sont ℓ -équivalentes si $\mathcal{S}_f = \mathcal{S}_{f'}$.

La relation de ℓ -dominance est un préordre sur F elle induit une relation d'équivalence : la ℓ -équivalence est un ordre partiel sur les classes de ℓ -équivalence.

La notion de ℓ -équivalence ne doit pas être confondue avec la relation d'indistinguabilité.

Définition : f et f' sont indistinguables si $\bar{\lambda}_f(s) = \bar{\lambda}_{f'}(s)$ pour tout s de I^* .

THEOREME f ℓ -domine f' si et seulement si $\mathcal{S}_{f'} \leq \mathcal{S}_f$
 f ℓ -domine f' strictement si et seulement si $\mathcal{S}_{f'} < \mathcal{S}_f$.

Définition : Cf. Théorème similaire pour la détection.

Afin de définir la ℓ -dominance pour les séquences l'ensemble des expressions de localisation doit être ordonné.

Soit L et L' deux expressions de localisation, L est plus précis que L' noté $L > L'$ si $L + L' = L$; L est strictement plus précise que L' (noté $L >> L'$) si $L + L' = L$ et $L \neq L'$.

Il est facile de voir que cette relation est un ordre sur l'ensemble des expressions de localisation.

L'ensemble des pannes localisées par s est noté L_s .

L'expression de localisation de s est notée l_s .

Du fait que généralement une séquence peut détecter certaines pannes sans les localiser, on a $L_s \ll L_{s'}$.

THEOREME Soit S préfixe de s_F et S' l'ensemble des séquences de S tronquées du dernier élément. Si f est localisée par toute séquence de S et qu'il existe f' localisée par aucune séquence de S' alors $s_F = s_{F-f}$

Définition : Une séquence s_1 ℓ -domine s_2 (noté $s_1 \succcurlyeq s_2$) si $L_{s_1} \supseteq L_{s_2}$, $|s_1| \leq |s_2|$ et $\delta_{F_1}^*(s_1) = \delta_{F_1}^*(s_2)$ avec $F_1 = F - L_{s_1}$.

Définition : Une séquence s_1 ℓ -domine strictement s_2 (noté $s_1 \succ s_2$) si s_1 ℓ -domine s_2 et $|s_1| < |s_2|$.

Définition : Deux séquences s_1 et s_2 sont ℓ -équivalents si $L_{s_1} = L_{s_2}$, $|s_1| = |s_2|$, et $\delta_{F_1}^*(s_1) = \delta_{F_1}^*(s_2)$

La relation de ℓ -dominance entre séquences est un préordre sur I^* , elle induit la relation d'équivalence : la ℓ -équivalence et un ordre partiel sur les classes de ℓ -équivalence.

- THEOREME
- 1) Si s_1 ℓ -domine s_2 alors $s_1 s$ ℓ -domine $s_2 s$.
 - 2) Si s_1 ℓ -domine s_2 , $|s_1| = |s_2|$ et $s_2 s$ localise F de façon optimale alors $s_1 s$ localise F de façon optimale.
 - 3) Si s_1 ℓ -domine strictement s_2 , alors $s_2 s$ ne localise pas F de façon optimale.

Démonstration : Similaire à la d-dominance.

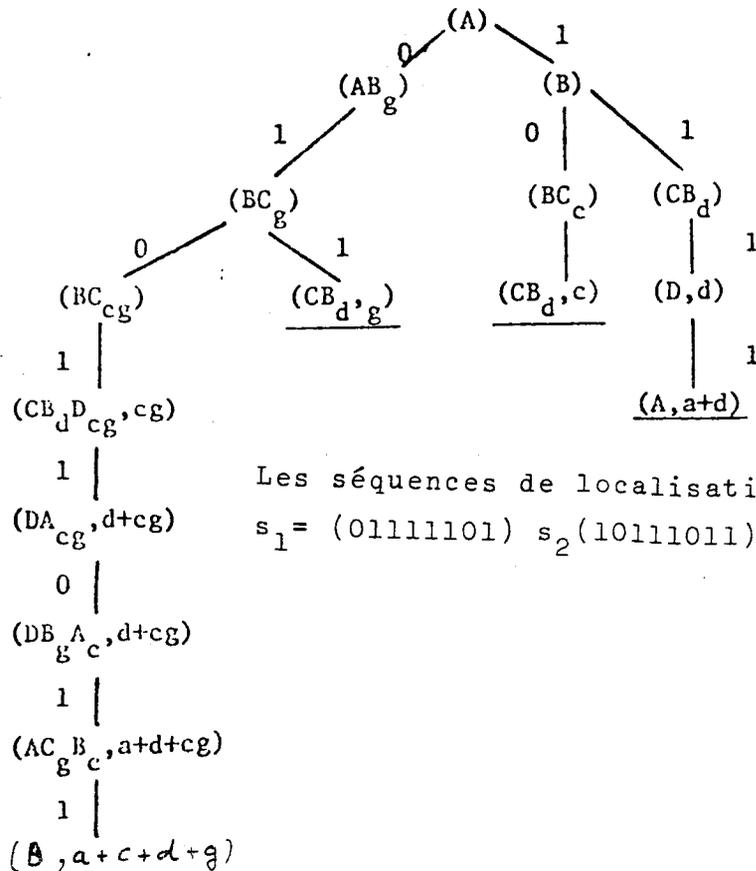
IV - 2. Procédure

Dans la procédure de localisation, comme dans la procédure de détection un arbre successeur est construit. Un noeud n marqué avec (q_n, L_n) ou L_n est l'expression de localisation de n ; soit L_n l'ensemble des pannes localisées par n , q_n est l'état produit de la machine produit $M_{F'}$, avec $F' = F - L_n$. La relation de l -dominance entre séquences peut être interprétée sur l'arbre et les relations $n_1 \geq n_2, > n_1$ définies. La translation des procédures de détection aux procédures de localisation est simple.

EXEMPLE : Considérons la table de test suivante :

Q \ I	0	1
A	$AB_g, 0$	$B, 0$
B	$BC_c, 0$	$CB_d, 0$
C	$C, 0$	$DA_d, 10_d$
D	$D, 1$	$AD_a, 01_a$

L'arbre de localisation réduit est :



Les séquences de localisation optimales sont :
 $s_1 = (01111101)$ $s_2 = (10111011)$ $s_3 = (01011011)$

V - EXTENSIONS : PANNE DANS L'INITIALISATION et MACHINES ASYNCHRONES

V - 1. Traitement des pannes qui affectent l'initialisation :

Dans les chapitres précédents il était fait l'hypothèse que l'état initial de la machine sous test était correcte. En fait, la méthode présentée peut se généraliser au cas où l'état initial n'est pas certain.

Quand l'état initial n'est pas connu, la séquence de sortie dépend en général de l'état initial et n'est pas unique. Dans ce cas, la simulation d'une séquence d'entrée est l'arbre des séquences de sortie de la machine normale. Pour la machine produit le concept de vecteur d'incertitude [4] sur l'état atteint, peut être aisément étendu.

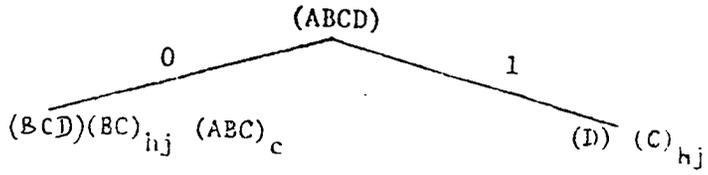
EXEMPLE : Considérons la table de test suivante :

Q \ I	0	1
A	$AB_{hi}, 0$	$BC_d, 0$
B	$AB_{hi}D_e, 0$	$CB_d, 0$
C	$AB_h D_{ef}, 0$	$DC_{hj}, 1$
D	$AB_h D_f, 0$	$DC_{hj}A_c, 0$

L'incertitude initiale est (ABCD). Après l'application de l'entrée 0 la sortie est toujours 0 et l'incertitude devient: (A) (B)_h (AB)_i (AD)_{cf}. L'interprétation est que la machine juste et les machines fausses M_b, M_c, M_d, M_j , sont dans l'état A; la machine fausse M_h est dans l'état B; l'incertitude de M_i est (AB) et l'incertitude de M_c et M_f est (AD).

Si l'on applique l'entrée 1 la machine normale peut

répondre 0 ou 1, on a donc un arbre :



La simulation de la séquence $s = (01101111111)$ est la suivante :

I	Q	0
	(ABCD)	
0	(A) (B) _h (AB) _i (AD) _{ef} , (0)	
1	(B) (C) _{dh} (BC) _i (BD) _{ef} , (0)	
1	(C) (CD) _{ef} , (0) (1) _{dh} (01) _i	
0	(A) (AD) _e (D) _f , (0)	
1	(B) (BD) _e (D) _f , (0)	
1	(C) (B) _d (CD) _e (D) _f , (0)	
1	(D) (C) _j , (1) (0) _{df} (01) _e	
1	(D) (A) _c , (0) (1) _j	
1	(D) (B) _c , (0)	
1	(D) , (0) (1) _c	
1	(D)	

Cet exemple montre qu'à l'incertitude des états il faut ajouter l'incertitude des sorties. Ainsi, la 3ème ligne montre que les pannes i et c peuvent ou ne peuvent pas être détectées suivant l'état initial. Par contre, si l'état initial est A chaque panne est détectée.

Une séquence d'entrée peut conduire à un arbre des

sorties possibles suivant l'état initial ainsi les concepts de séquences optimales de détection ou localisation ne peuvent plus s'appliquer. Vu la complexité du traitement des incertitudes il est préférable de faire cette étude uniquement dans la phase de forçage à un état initial. Le positionnement dans un état initial doit l'être à la fois pour la machine juste et pour les machines fausses. Dans l'exemple précédent une telle séquence de positionnement est $s = 0110111$ qui positionne machines justes et machines fausses dans l'état DC_j .

V - 2. Traitement des circuits asynchrones

Considérons une machine asynchrone (mode fondamental) dont les pannes sont représentées par la table de test suivante :

$Q \backslash x_1 x_2$	00	01	11	10
A	$\textcircled{A} B_c, 0$	$\textcircled{A}, 0$	$\textcircled{A} B_{df}, 1$	$B \textcircled{A}_a, 0$
B	$A \textcircled{B}_{ce}, 0$	$A \textcircled{B}_e, 0$	$B \textcircled{A}_b, 0$	$B \textcircled{A}_{ab}, 0$

La simulation de la séquence d'entrée 10, 11, 01, 11 avec l'état initial A est la suivante :

I	Q	O
	A	
10	$B \textcircled{A}_a$,0
10	$\textcircled{B} \textcircled{A}_a$,0
11	$\textcircled{B} \textcircled{A}_b \textcircled{A}_a$,01 _a
11	$\textcircled{B} \textcircled{A}_{ab}$,01 _{ab}
01	$A \textcircled{B}_e$,0
01	$\textcircled{A} \textcircled{B}_e$,0
11	$\textcircled{A} B_{df} \textcircled{B}_e$,10 _e
11	$\textcircled{A} \textcircled{B}_{def}$,10 _{def}

Pendant la simulation une entrée est changée seulement quand un état stable global (pour machines justes et fausses) est atteint.

On peut, d'autre part, se ramener au cas synchrone en construisant la machine synchrone équivalente. Il est à noter que s'il y a création de cycles (pour des entrées fixes) pour une machine faussi il n'est pas possible d'avoir de machine synchrone équivalente. D'autre part, une étude complète des machines asynchrones nécessite le traitement d'aléas qui peuvent être introduits par des pannes.

CONCLUSION

La difficulté du test des systèmes séquentiels et par rapport au test des systèmes combinatoires vient entre autres du fait que l'ensemble des séquences est infini. Pour simplifier le problème on utilise les propriétés de d-dominance entre pannes et séquences comme dans le cas du combinatoire; mais ici on définit la dominance entre séquences de façon que si s_1 domine s_2 alors $s_1 s$ domine $s_2 s$ ce qui permet, par une étude locale (sur un sous-ensemble² de séquences) d'en déduire une propriété globale (sur toutes les séquences). Ces définitions et les résultats qui en découlent permettent de déduire des algorithmes plus efficaces que les anciens pour la détection et donnent, pour l'utilisation de l'expression de localisation, des généralisations faciles au cas de la localisation.

Enfin, il a été montré que ces notions peuvent aussi être utilisées dans les cas où l'initialisation est défectueuse et dans le cas de réseaux asynchrones.

-:-:-:-:-:-:-:-

BIBLIOGRAPHIE

- 1) J.P. ROTH, W.C. BOURICUS, R. SCHNEIDER : "Programmed Algorithms to compute Tests to Detect and Distinguish Between Failures in Logic Circuits", IEEE Trans. on Comp. Octobre 1967.
- 2) J.F. POAGE : "The Derivation of Optimum Test for Logic Circuits", Ph. D Princeton Univ. 1963.
- 3) Z. KOHAVI : "Switching Automata Theory", Mc Graw Hill Computer Science Series, 1970.

CHAPITRE III

TEST DE RESEAUX REGULIERS

CHAPITRE III : TEST DE RESEAUX REGULIERS

INTRODUCTION

I - Symétrie, automorphisme et test

I - 1. Réseau de portes

I - 2. Réseau de modules

II - Itération et test

II - 1. Itération simple

II - 2. Itération complexe

CONCLUSION

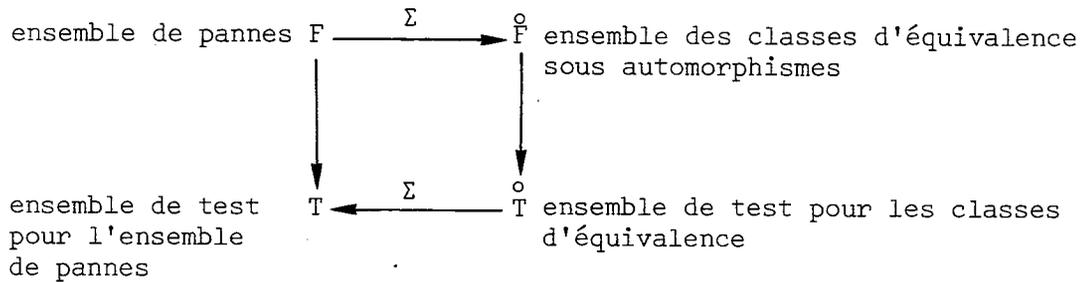
Ce travail a été fait en collaboration avec Mlle C. TURCAT.

INTRODUCTION

Les grands systèmes logiques sont conçus et réalisés en utilisant les régularités structurelles des données et des transformations qu'elles subissent. L'information peut être représentée de façon symétrique (vecteurs ...) ou de façon récurrente (listes) ; de telles données seront traitées avec des fonctions symétriques ou récurrentes. Aux concepts fonctionnels de symétrie et de récurrence correspondent, dans la réalisation, ceux d'isomorphisme et d'itération dans les réseaux. Ces notions, intégrant et l'aspect fonctionnel et l'implémentation, simplifient la recherche d'un test et conduisent à une expression fonctionnelle du test.

I - SYMETRIE, AUTOMORPHISME ET TEST

Nous montrons ici comment les symétries qui conduisent aux automorphismes de réseaux (c'est-à-dire isomorphisme du réseau sur lui-même) simplifient le test. Les automorphismes préservent la structure du réseau et donc les relations classiques entre pannes et vecteurs d'entrée sont préservées. Par suite, l'ensemble des pannes peut être réduit de façon originale. Les exemples traités montrent un facteur de réduction de 4 à 14 suivant le réseau. Il doit aussi être noté que plus le nombre d'automorphismes est grand, plus la réduction est importante. Quand la réduction de l'ensemble des pannes par la relation d'automorphisme est faite, des méthodes classiques sont utilisées pour générer un ensemble de test réduit. Ensuite un ensemble de test complet est généré en étendant le précédent ensemble de test par automorphisme. Soit F l'ensemble des pannes, $\overset{\circ}{F}$ l'ensemble réduit de pannes, $\overset{\circ}{T}$ un ensemble de test de $\overset{\circ}{F}$ et T un ensemble de test de F . La méthode peut être visualisée par le diagramme commutatif suivant, où Σ dénote l'ensemble des automorphismes.



Le problème de la recherche des automorphismes dépasse l'objet de cette étude.

Le lecteur intéressé peut trouver un tel matériel, par exemple dans Turcat [4] ou Saucier [3].

Le premier paragraphe traite des réseaux de portes et donne des méthodes pour obtenir des tests optimaux ou pseudo-optimaux pour les circuits combinatoires. Le deuxième paragraphe étudie les réseaux de modules. La définition d'automorphisme est étendue et conduit à une approche multi-niveau pour tester les grands réseaux symétriques.

I - 1. RESEAUX DE PORTES

Dans ce paragraphe, nous considérons qu'un réseau est fait de portes ET, OU, NON, NI, Disjonction.

I - 1.1. Réseaux acycliques avec entrées indépendantes

Considérons un réseau acyclique N de portes ET, OU, NI, NON, Disjonction. Soit c une connexion de N , la panne c collée à k ($k \in \{0, 1\}$) est notée c_k . Rappelons que si α est une panne et f la fonction réalisée par le réseau N , alors f_α dénote la fonction fautive réalisée par le réseau avec la panne α . Si $f(t) \neq f_\alpha(t)$, alors t est un test de la panne α , l'ensemble de test de la panne α est noté T_α . La fonction de test ζ_α est définie par :

$$\zeta_\alpha(x) = 1 \iff f(x) \neq f_\alpha(x) \text{ (avec } x \text{ et } t \text{ vecteurs d'entrée).}$$

Afin de définir les automorphismes de réseau, un graphe marqué peut être associé au réseau R : l'ensemble des sommets correspondent aux portes, aux entrées et aux sorties du réseau alors que les arcs correspondent aux connexions (chaque type de porte a une marque différente).

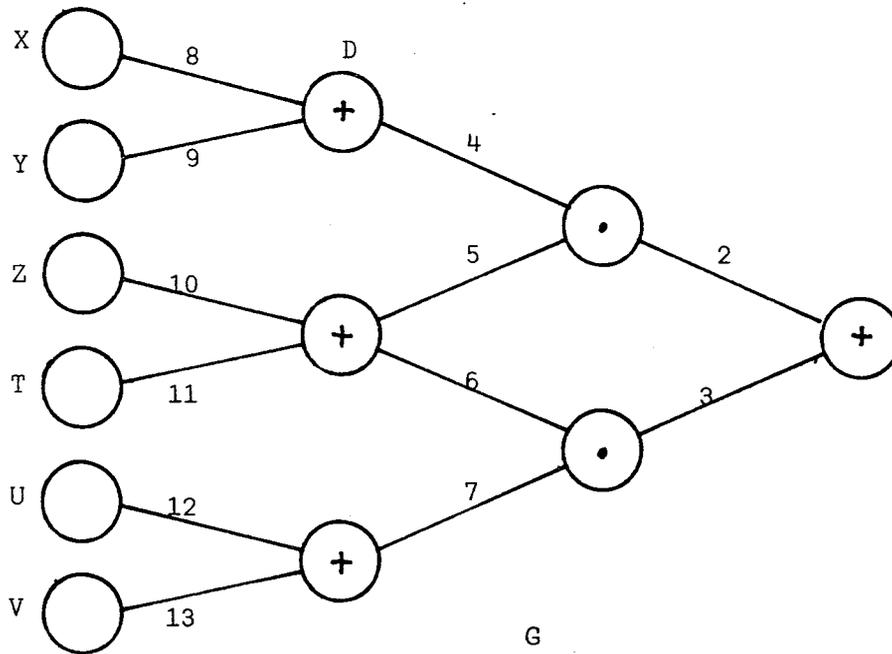
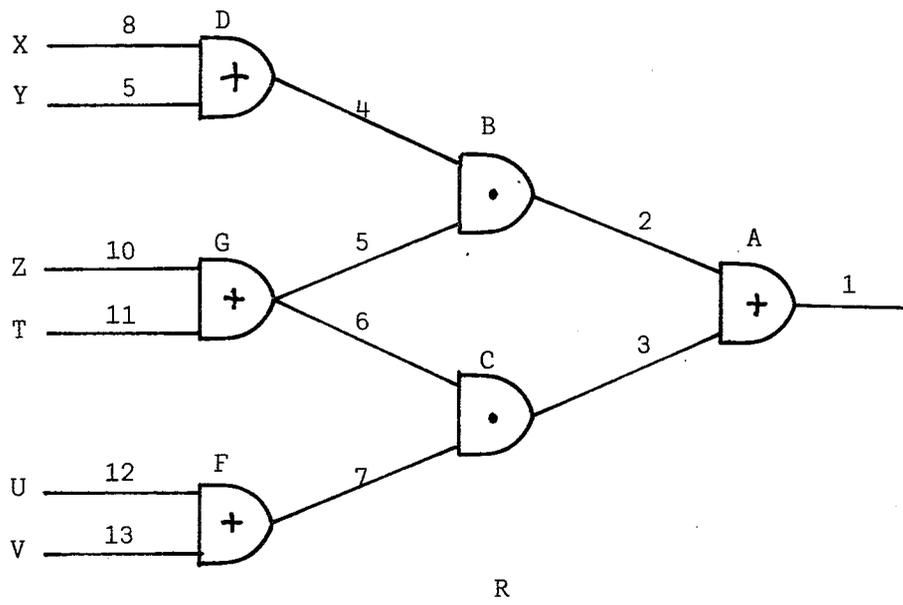


Figure 1 - Un réseau R et son graphe associé G

Soit $G = (S, A)$ un graphe où S est l'ensemble des sommets et A l'ensemble des arcs.

Définition

Un automorphisme σ de G est une bijection de l'ensemble des sommets S sur lui-même qui induit une bijection de l'ensemble des arcs (notée aussi σ) telle que si $(s_i, s_j) \in A$ alors $\sigma(s_i, s_j) = (\sigma(s_i), \sigma(s_j)) \in A$ et qui respecte les marques (s_i, s_j sont des sommets) \circ

Si σ est un automorphisme du graphe associé au réseau N , il définit un automorphisme de réseau noté aussi σ . L'ensemble des automorphismes est noté $\Sigma = \{\sigma_i\}$.

EXEMPLE

Les automorphismes du réseau N de la figure 1 sont (Turcat [1]) :

- 8 transpositions échangeant seulement des connexions d'entrée :

$$TR = \{\tau_0 = \text{id} ; \tau_1 = (8 \leftrightarrow 9) ; \tau_2 = (10 \leftrightarrow 11) ; \tau_3 = (12 \leftrightarrow 13) ;$$

$$\tau_4 = \tau_1 \circ \tau_2 ; \tau_5 = \tau_2 \circ \tau_3 ; \tau_6 = \tau_3 \circ \tau_1 ; \tau_7 = \tau_1 \circ \tau_3\}$$

- 8 automorphismes obtenus en composant TR avec la bijection entre connexions :

$$\bar{\sigma} = \{(1)(2 \leftrightarrow 3)(4 \leftrightarrow 7)(5 \leftrightarrow 6)(\{8,9\} \leftrightarrow \{12,13\})(\{10,11\})\} \quad \Delta$$

Définition

Soit σ un automorphisme du réseau N et soit f une fonction booléenne des entrées et connexions de N , alors la fonction automorphe σf est définie par :

$$\sigma f(x_1, x_2, \dots, x_n, c^1, c^2, \dots, c^p) = f(\sigma(x_1), \dots, \sigma(x_n), \sigma(c^1), \dots, \sigma(c^p))$$

(x_1, x_2, \dots, x_n sont les entrées et c^1, c^2, \dots, c^p les connexions) \circ

Définition

La panne multiple $\beta_{i_1}^1 \beta_{i_2}^2 \dots \beta_{i_q}^q$ est automorphe à la panne multiple $\alpha_{k_1}^1 \alpha_{k_2}^2 \dots \alpha_{k_q}^q$ si :

- il existe un automorphisme σ de N tel que $\sigma(\alpha^j) = \beta^j$
- les connexions α^j et β^j sont collées à la même valeur $(\sigma(i_j) = k_j)$ \circ

EXEMPLE

Pour le réseau décrit dans la figure 1, considérons l'automorphisme σ

$$\sigma = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ 1 & 3 & 2 & 7 & 6 & 5 & 4 & 12 & 13 & 10 & 11 & 8 & 9 \end{bmatrix}$$

La fonction automorphe de la fonction fautive $f_{2_o} = (C+D)(E+F)$ est :

$$\sigma f_{2_o} = f_{3_o} = (A+B)(C+D).$$

La panne multiple $2_o 12_o$ est automorphe à $3_o 8_o$ parce que

$$\sigma(2_o 12_o) = 3_o 8_o \quad \Delta$$

I - 1.2. Automorphisme et test

Ce paragraphe montre que si deux pannes sont automorphes, alors leur fonction de test (et ensemble de test) sont aussi automorphes par le même automorphe. Par suite, on ne cherche les vecteurs de test que pour les pannes qui ne sont pas automorphes et un test complet peut être obtenu en étendant les vecteurs de test obtenus.

THEOREME

La relation, "les pannes α et β sont automorphes" est une relation d'équivalence.

DEMONSTRATION

L'ensemble des automorphismes est un groupe, donc $\alpha = \sigma(\beta) \Rightarrow \sigma^{-1}(\alpha) = \beta$ (σ^{-1} est l'inverse de σ), donc la relation est symétrique.

$\alpha = \sigma_0(\alpha)$ σ_0 est l'identité, donc la relation est réflexive.

$\alpha = \sigma_1(\beta)$ et $\beta = \sigma_2(\gamma) \Rightarrow \alpha = \sigma_1\sigma_2(\gamma)$, donc la relation est transitive.

L'ensemble des classes d'automorphisme de l'ensemble de pannes F est noté $\overset{\circ}{F}$ \square

EXEMPLE

Les classes d'automorphisme pour les collages simples du réseau décrit figure 1 sont :

$$\{2,3\} \{4,7\} \{5,6\} \{10,11\} \{8,9,12,13\} \quad \Delta$$

LEMME

Si les pannes α et β sont automorphes par l'automorphisme σ , alors les fonctions de test \mathcal{C}_α et \mathcal{C}_β sont automorphes par σ .

DEMONSTRATION

Si $f_\alpha(\sigma^{-1}x) \neq f(\sigma^{-1}x)$ alors $\sigma f_\alpha(\sigma^{-1}x) \neq \sigma f(\sigma^{-1}x)$, donc $f_\beta(x) \neq f(x)$. De la même façon, si $f_\alpha(\sigma^{-1}x) = f(\sigma^{-1}x)$ alors $f_\beta(x) = f(x)$ \square

THEOREME

Si $\overset{\circ}{T} = \bigcup_i t_i$ est un test de $\overset{\circ}{F}$ alors $T = \bigcup_{i,j} \sigma_j(t_i)$ est un test de F .

DEMONSTRATION

Soit α une panne de F alors il existe β dans $\overset{\circ}{F}$ et σ_j dans Σ tels que $\alpha = \sigma_j(\beta)$. Si $\overset{\circ}{T}$ est un test de $\overset{\circ}{F}$, alors il existe $t_i \in \overset{\circ}{T}$ qui détecte β , donc à cause du lemme, $\sigma_j(t_i)$ détecte α .

Rappelons que l'ensemble de pannes F_1 domine l'ensemble de panne F_2 (noté $F_1 \geq F_2$) si t détecte F_2 implique t détecte F_1 .

Soit F' dominée par F et $(\overset{\circ}{F}')$ dominée par $\overset{\circ}{F}$. Soit $\Sigma(T) = \bigcup_{i,j} \sigma_j(t_i)$ avec $t_i \in T, \sigma_j \in \Sigma$ Δ

COROLLAIRE

- Si T' est un test de $\overset{\circ}{F}'$, alors $\Sigma(\overset{\circ}{T}')$ est un test de F
- si $(\overset{\circ}{T})'$ est un test de $(\overset{\circ}{F})'$, alors $\Sigma(\overset{\circ}{T}')$ est un test de F \square

EXEMPLE

Le réseau de la figure 1 a $3^{13}-1$ collages multiples. Si l'on considère seulement les pannes sur les entrées et fan-out (Bossen and Hong [1]) $3^8-1 = 6560$ pannes sont obtenues ; utilisant le concept d'écoulement (Verdillon [5]), un ensemble de 126 pannes est obtenu et le nombre de classes d'automorphisme est 34.

$$\begin{aligned} \overset{\circ}{F}' = \{ & (2) (2.6) (2.6.12) (2.7) (2.7.10) (2.10) (2.10.12) \\ & (2.12) (4) (4.6) (4.6.10) (4.6.10.12) (4.6.12) (4.7) \\ & (4.7.10) (4.10) (4.10.12) (4.12) (5) (5.6) (5.6.8) \\ & (5.6.8.12) (5.8) (5.8.10) (5.8.10.12) (5.8.12) (5.10) \\ & (5.10.12) (5.12) (8) (8.10) (8.10.12) (8.12) (10) \} \end{aligned}$$

En utilisant des résultats donnés dans Verdillon [5], les classes d'automorphisme dominantes sont $(\overset{\circ}{F})' = \{4,5,8,10\}$.

Les fonctions de test pour 4,5,8,10 sont :

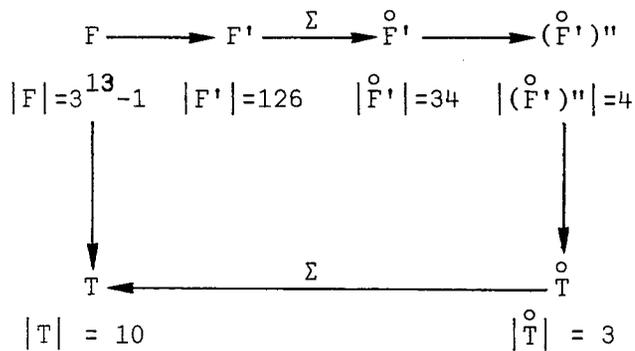
$$\begin{aligned} \mathcal{T}_4 &= \bar{A} \bar{B} (C+D) \bar{E} \bar{F} \\ \mathcal{T}_5 &= (A+B) \bar{C} \bar{D} \\ \mathcal{T}_8 &= A \bar{B} (C+D) \bar{E} \bar{F} \\ \mathcal{T}_{10} &= C \bar{D} (A+B+E+F) \end{aligned}$$

Le test $\overset{\circ}{T} = \{\bar{A}\bar{B}C\bar{D}\bar{E}\bar{F}, \bar{A}B\bar{C}\bar{D}E\bar{F}, \bar{A}\bar{B}C\bar{D}E\bar{F}\}$ détecte les pannes 4, 5, 8 et 10.

Les tests automorphes sont :

$$\begin{aligned} \Sigma(\bar{A}\bar{B}C\bar{D}\bar{E}\bar{F}) &= \{\bar{A}\bar{B}C\bar{D}\bar{E}\bar{F}\} \\ \Sigma(\bar{A}B\bar{C}\bar{D}E\bar{F}) &= \{\bar{A}B\bar{C}\bar{D}E\bar{F}\} \\ \Sigma(\bar{A}\bar{B}C\bar{D}E\bar{F}) &= \{\bar{A}\bar{B}C\bar{D}E\bar{F}, \bar{A}B\bar{C}\bar{D}E\bar{F}, \bar{A}\bar{B}C\bar{D}E\bar{F} \\ & \quad \bar{A}\bar{B}C\bar{D}\bar{E}\bar{F}, \bar{A}B\bar{C}\bar{D}\bar{E}\bar{F}, \bar{A}\bar{B}C\bar{D}\bar{E}\bar{F} \\ & \quad \bar{A}\bar{B}C\bar{D}\bar{E}\bar{F}\} \end{aligned}$$

$T = \Sigma(\overset{\circ}{T})$ détecte toutes les pannes multiples et $|T| = 10$. Le diagramme suivant illustre les étapes de cet exemple :



△

I - 1.3. Test optimal et pseudo-optimal

Test Optimal

Le précédent théorème ne garantit pas un test optimal car en général l'ensemble Σ de tous les automorphismes n'est pas nécessaire pour décrire une classe d'automorphisme. Par suite si t est un vecteur de test d'une faute représentant la classe, il n'est plus nécessaire de générer tout $\Sigma(t)$ pour avoir un ensemble de test qui détecte tout élément dans la classe.

DEFINITION

Soit α une panne. Un ensemble d'automorphisme Σ_α est un ensemble de générateur de la classe d'automorphisme $\overset{\circ}{\alpha}$ si $\Sigma_\alpha(\alpha) = \overset{\circ}{\alpha} \quad \circ$

EXEMPLE

Dans le réseau de la figure 1, la classe d'automorphisme de 8 est $\overset{\circ}{8} = \{8,9,12,13\}$ et un ensemble de générateur est

$$\Sigma_8 = \{\tau_0, \tau_1, \tau_0\bar{\sigma}, \tau_1\bar{\sigma}\} \quad \Delta$$

La procédure pour obtenir un test optimal est la suivante :

Etape 1 : Trouver $(\overset{\circ}{F})'$ dominée par $\overset{\circ}{F}$

Etape 2 : Trouver T_α et Σ_α pour $\alpha \in (\overset{\circ}{F})'$

Etape 3 : Trouver une couverture optimale de $\bigcup_\alpha \Sigma_\alpha(\alpha)$ avec $\bigcup_\alpha \Sigma_\alpha(T_\alpha)$

EXEMPLE

$$\Sigma_4 = \{\tau_0, \tau_0\bar{\sigma}\} = \Sigma_5$$

$$\Sigma_8 = \{\tau_0, \tau_1, \tau_0\bar{\sigma}, \tau_1\bar{\sigma}\}$$

$$\Sigma_{10} = \{\tau_0, \tau_2\}$$

D'où

$$\Sigma_4 \mathcal{C}_4 = \{\bar{A}\bar{B}(C+D)\bar{E}\bar{F}\}$$

$$\Sigma_5 \mathcal{C}_5 = \{(A+B)\bar{C}\bar{D}, (E+F)\bar{C}\bar{D}\}$$

$$\Sigma_8 \mathcal{C}_8 = \{\bar{A}\bar{B}(C+D)\bar{E}\bar{F}, \bar{A}\bar{B}(C+D)\bar{E}\bar{F}, \bar{A}\bar{B}(C+D)\bar{E}\bar{F}, \bar{A}\bar{B}(C+D)\bar{E}\bar{F}\}$$

$$\Sigma_{10} \mathcal{C}_{10} = \{\bar{C}\bar{D}(A+B+E+F), \bar{C}\bar{D}(A+B+E+F)\}$$

Un test optimal est :

$$T_1 = \{\bar{A}\bar{B}\bar{C}\bar{D}\bar{E}\bar{F}, \bar{A}\bar{B}\bar{C}\bar{D}\bar{E}\bar{F}, \bar{A}\bar{B}\bar{C}\bar{D}\bar{E}\bar{F}, \bar{A}\bar{B}\bar{C}\bar{D}\bar{E}\bar{F}, \bar{A}\bar{B}\bar{C}\bar{D}\bar{E}\bar{F}, \bar{A}\bar{B}\bar{C}\bar{D}\bar{E}\bar{F}\}, |T_1| = 6$$

Test pseudo-optimal

Dans la procédure pour obtenir un test optimal, les automorphismes n'ont pas été utilisés pour simplifier le problème de couverture. Si la couverture est faite sur l'ensemble des classes, avant d'étendre le test avec les automorphismes, un test pseudo-optimal est atteint. Afin de définir la méthode, étendons la définition d'ensemble des générateurs à un ensemble de classes.

DEFINITION

Soit t un ensemble de pannes $\alpha_1, \alpha_2, \dots, \alpha_n$ un ensemble de générateurs des classes $\overset{\circ}{\alpha}_1, \overset{\circ}{\alpha}_2, \dots, \overset{\circ}{\alpha}_n$ est $\Sigma_t = \Sigma_{\alpha_1} \cup \Sigma_{\alpha_2} \cup \dots \cup \Sigma_{\alpha_n} \quad \circ$

PROCEDURE

Etapes I et II comme précédemment

Etape III, trouver une couverture optimale de $(\overset{\circ}{F})'$ avec $\cup T_{\alpha}$. Soit $T' = \{t_i\}$ une telle couverture, alors un test est $T = \cup_i \Sigma_{t_i}(t_i)$.

EXEMPLE

$$\mathcal{C}_4 = \bar{A}\bar{B}(C+D)\bar{E}\bar{F}$$

$$\mathcal{C}_5 = (A+B)\bar{C}\bar{D}$$

$$\mathcal{C}_{8,10} = \bar{A}\bar{B}\bar{C}\bar{D}\bar{E}\bar{F} \quad (\mathcal{C}_{8,10} = \mathcal{C}_8 \cdot \mathcal{C}_{10})$$

$$\Sigma_{8,10} = \Sigma_8 \cup \Sigma_{10} = \{\tau_0, \tau_1, \tau_0\bar{\sigma}, \tau_1\bar{\sigma}, \tau_2\}$$

$$\Sigma_{8,10} (\mathcal{C}_{8,10}) = \{\overline{ABCDEF}, \overline{ABCDEF}, \overline{ABCDEF}, \overline{ABCDEF}, \overline{ABCDEF}\}$$

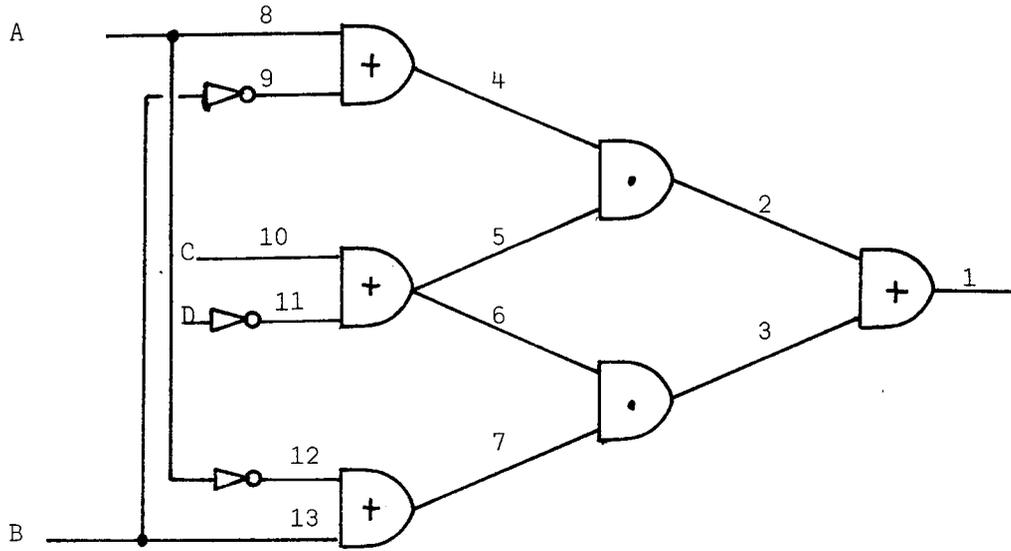
Un test est donc $T_2 = \{\overline{ABCDEF}, \overline{ABCDEF}, \overline{ABCDEF}, \overline{ABCDEF}, \overline{ABCDEF}, \overline{ABCDEF}, \overline{ABCDEF}\}$, $|T_2| = 7$

On obtient ici un test de 7 vecteurs alors que le test optimal en a 6 Δ

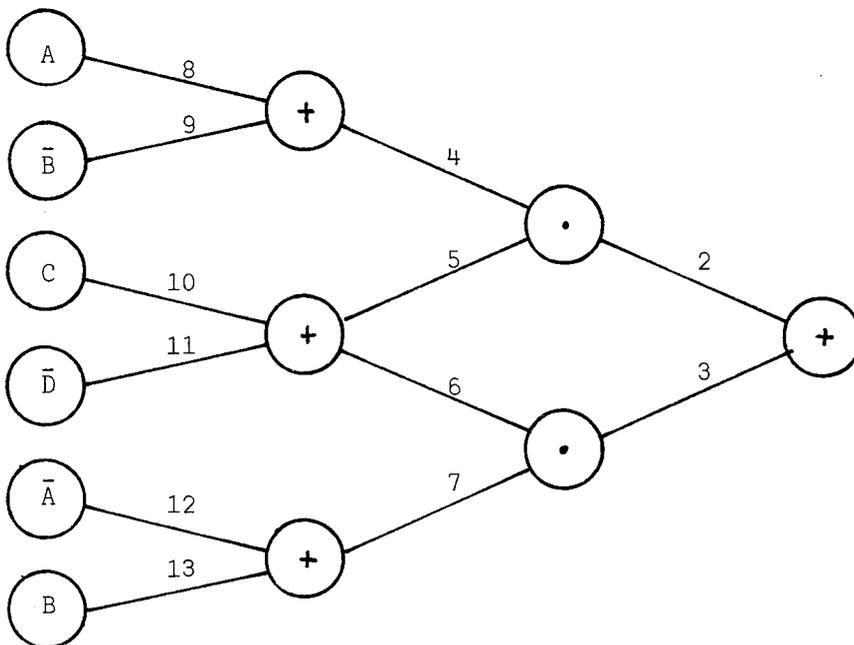
I - 2. RESEAU GENERAL

I - 2.1. Réseau avec entrées dépendantes

Considérons le réseau décrit par la figure 2. C'est le précédent réseau (figure 1) avec des entrées non indépendantes. Avec la précédente définition d'automorphisme, le seul automorphisme est maintenant l'identité. Cependant la fonction réalisée est encore symétrique (invariante par permutation et complémentation des variables). Afin de tenir compte de la complémentation, le graphe associé au réseau est défini d'une nouvelle façon : si une entrée est normale et complétement, alors deux sommets sont utilisés, un pour l'entrée, l'autre pour son complément.



Un réseau R



Le graphe associé de G

Figure 2

DEFINITION

Un automorphisme booléen σ d'un graphe associé est un automorphisme tel que si $\tilde{x}_j = \sigma(\tilde{x}_i)$ alors $\tilde{x}_i = \sigma(\tilde{x}_j)$ (x_i et x_j sont des entrées) \square

Comme précédemment, cette définition induit la définition d'automorphisme booléen pour un réseau. Il est aisé de voir que les précédents résultats sont toujours vrais.

EXEMPLE

Dans le réseau de la figure 2, les automorphismes booléens sont :

σ_0 : identité

σ_1 : permutation (10 \leftrightarrow 11)

σ_2 : permutation {(8 \leftrightarrow 12) (9 \leftrightarrow 13)}

σ_3 : permutation {(8 \leftrightarrow 13) (9 \leftrightarrow 12)}

σ_4 : permutation {(8 \leftrightarrow 9) (12 \leftrightarrow 13)}

$\sigma_5 = \sigma_1 \sigma_2$

$\sigma_6 = \sigma_1 \sigma_3$

$\sigma_7 = \sigma_1 \sigma_4$

Le nombre d'automorphismes est maintenant 8 au lieu de 16 quand les entrées étaient indépendantes \triangle

I - 2.2. Réseau acyclique avec plusieurs sorties

Considérons le réseau de la figure 3 : il n'y a pas d'automorphisme et la fonction (f,g) n'est pas symétrique, mais chaque fonction f et g séparément est symétrique. De la même façon, les deux sous-réseaux N_f et N_g (figure 4), prédécesseurs de f et g, ont des automorphismes.

Sur chacun des sous-réseaux N_f et N_g , les méthodes précédentes peuvent être appliquées.

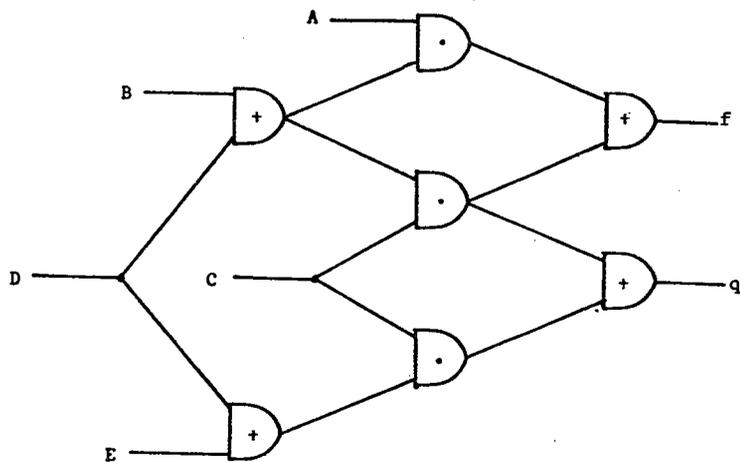
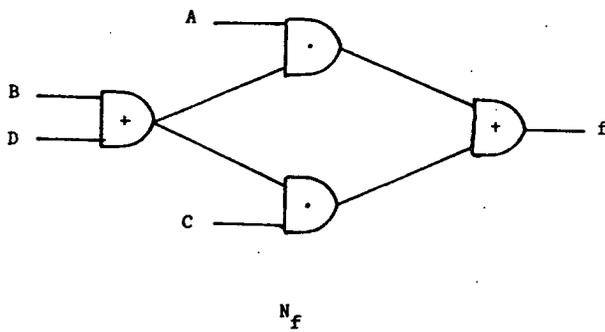
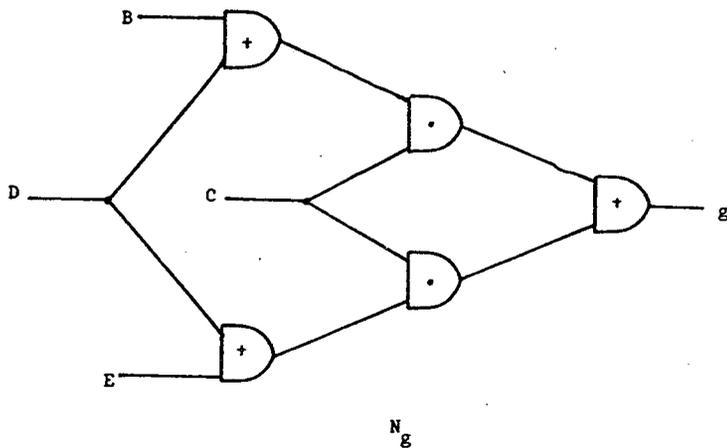


Figure 3



N_f



N_g

Figure 4

I - 2.3. Réseaux cycliques

Il n'y a pas de différence entre automorphismes dans les réseaux cycliques ou acycliques. La différence vient du test qui est maintenant une séquence au lieu d'un ensemble. Si l'on considère des circuits avec initialisation et les pannes qui n'affectent pas l'initialisation, alors les précédentes méthodes peuvent être appliquées pour générer un ensemble de séquences de test automorphes, mais il faut chaque fois réinitialiser le circuit et un test optimal ne peut pas être garanti.

I - 3. RESEAUX DE MODULES

Pour un grand système, la description au niveau de la porte a de sévères limitations du fait de la complexité. Ce paragraphe montre comment le concept d'automorphismes de réseau de portes peut être étendu aux réseaux de modules. Cette extension donne une utilisation multi-niveau qui permet de traiter de grands réseaux symétriques. Les portes sont des modules très particuliers qui sont symétriques, ont une seule sortie et sont combinatoires. Afin d'étendre les résultats précédents à des modules plus compliqués, un automorphisme local de module doit être défini. Etudions d'abord les réseaux de modules combinatoires.

I - 3.1. Réseaux de modules combinatoires

Soit $\sigma = (\sigma_x, \sigma_y)$ une paire de permutations sur les entrées X et les sorties Y d'un module M qui réalise la fonction f . Notons σf la fonction définie par $\sigma f(x) = \sigma_y f(\sigma_x X)$. σ est un automorphisme du module M si $f = \sigma f$.

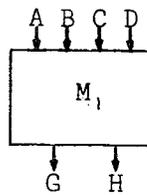
DEFINITION

Un automorphisme σ d'un réseau de modules est un automorphisme de réseau tel que la restriction de σ aux entrées sorties de chaque module soit un automorphisme de ce module \circ

EXEMPLE

Soit M_1 un module qui réalise la fonction

$$\begin{cases} G = A + BC \\ H = D + BC \end{cases}$$



Les automorphismes de module sont

$$\sigma_0^1 = \text{identité}$$

$$\sigma_1^1 = \begin{bmatrix} A & B & C & D & G & H \\ A & C & B & D & G & H \end{bmatrix}$$

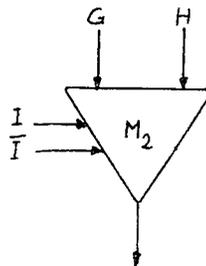
$$\sigma_2^1 = \begin{bmatrix} A & B & C & D & G & H \\ D & B & C & A & H & G \end{bmatrix}$$

$$\sigma_3^1 = \sigma_1^1 \sigma_2^1$$

Considérons le réseau fait du précédent module et d'un multiplexeur M_2 (Figure 6). Les automorphismes du multiplexeur sont :

$$\sigma_0^2 = \text{identité}$$

$$\sigma_1^2 = \begin{bmatrix} G & H & I & \bar{I} \\ H & G & \bar{I} & I \end{bmatrix}$$



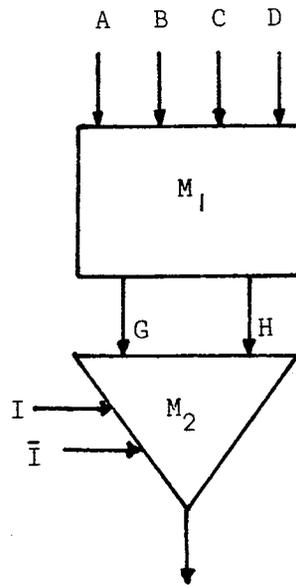


Figure 6

Les automorphismes du réseau global sont :

$$\sigma_0 = \text{identité} = (\sigma_0^1, \sigma_0^2)$$

$$\sigma_1 = \begin{bmatrix} A & B & C & D & G & H & I & \bar{I} \\ A & C & B & D & G & H & I & \bar{I} \end{bmatrix} = (\sigma_1^1, \sigma_0^2)$$

$$\sigma_2 = \begin{bmatrix} A & B & C & D & G & H & I & \bar{I} \\ D & B & C & A & H & G & \bar{I} & I \end{bmatrix} = (\sigma_2^1, \sigma_1^2)$$

$$\sigma_3 = \sigma_1 \sigma_2 = (\sigma_3^1, \sigma_1^2) \quad \Delta$$

I - 3.2. Réseau avec des modules séquentiels

Soit M un module séquentiel (modèle de Mealy) qui réalise f :

$$f = \begin{cases} \lambda : X \times Q \rightarrow Y \\ \mu : X \times Q \rightarrow Q \end{cases}$$

avec X ensemble d'entrées, Y ensemble de sorties et Q ensemble d'états (q_0 est l'état initial).

Soit $\sigma = (\sigma_x, \sigma_q, \sigma_y)$ un triplet de permutations sur les entrées, les états et les sorties.

La fonction ofX est définie par :

$$\begin{cases} \sigma.\lambda : \sigma\lambda(x,q) = \sigma_y \lambda[\sigma_x(x), \sigma_q(q)] \\ \sigma.\mu : \sigma\mu(x,q) = \sigma_q \mu[\sigma_x(x), \sigma_q(q)] \end{cases}$$

DEFINITION

σ est un automorphisme de module séquentiel si

$$\sigma(q_0) = q_0 \text{ et } \sigma f = f \quad \text{O}$$

EXEMPLE

Considérons le module défini par le réseau de portes suivant :

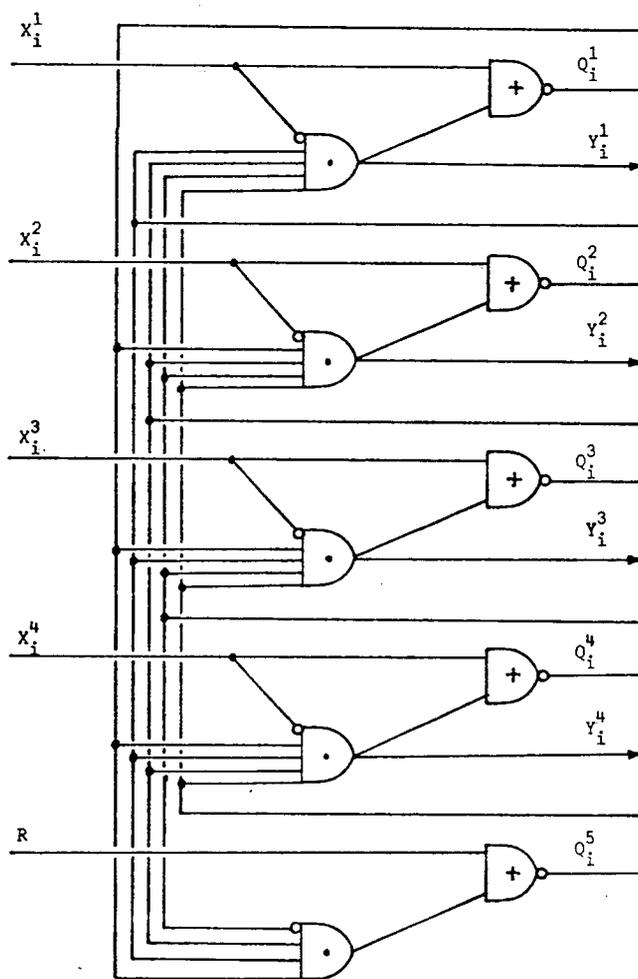


Figure 7

Un automorphisme de module est : $R \begin{matrix} X^1 & X^2 & X^3 & X^4 \\ Y^1 & Y^2 & Y^3 & Y^4 \end{matrix}$
 $R \begin{matrix} X^3 & X^2 & X^1 & X^4 \\ Y^3 & Y^2 & Y^1 & Y^4 \end{matrix} \quad \Delta$

DEFINITION

Un automorphisme de réseau de modules est un automorphisme de réseau tel que sa restriction (σ_x, σ_y) aux entrées et sorties de chaque module M vérifie :

- si M est séquentiel, il existe σ_q tel que $(\sigma_x, \sigma_q, \sigma_y)$ soit un automorphisme du module M
- si M est combinatoire, (σ_x, σ_y) est un automorphisme de M \circ

EXEMPLE

Considérons le circuit décrit par la figure 8 : la description au niveau de la porte est donnée par la figure 9.

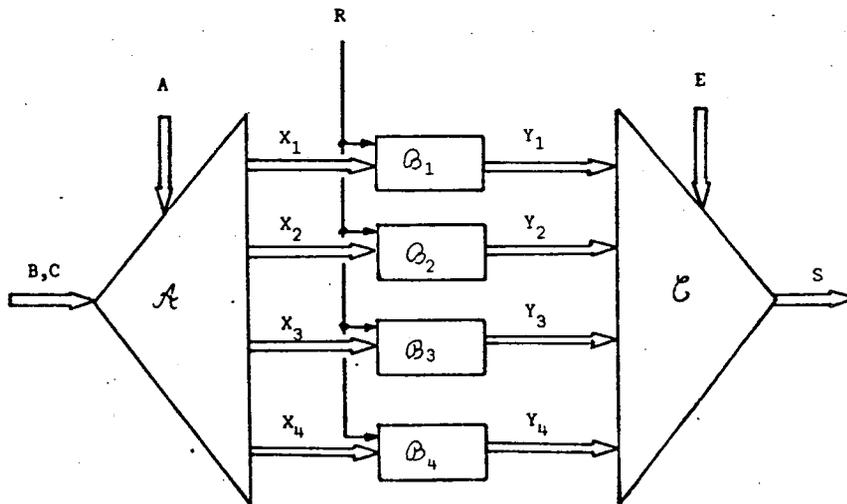


Figure 8

Niveau 1 - \mathcal{A} est un démultiplexeur, \mathcal{C} un multiplexeur, et les \mathcal{B}_i sont des modules séquentiels. A ce niveau, on voit qu'il est suffisant de tester un des \mathcal{B}_i et de déduire les tests des autres par automorphisme.

Niveau 2 - Chaque \mathcal{B}_i décrit par la figure 7 a des automorphismes ; chaque permutation des entrées X_i^j induit une permutation des lignes de réaction Q_i^j et des sorties Y_i^j .

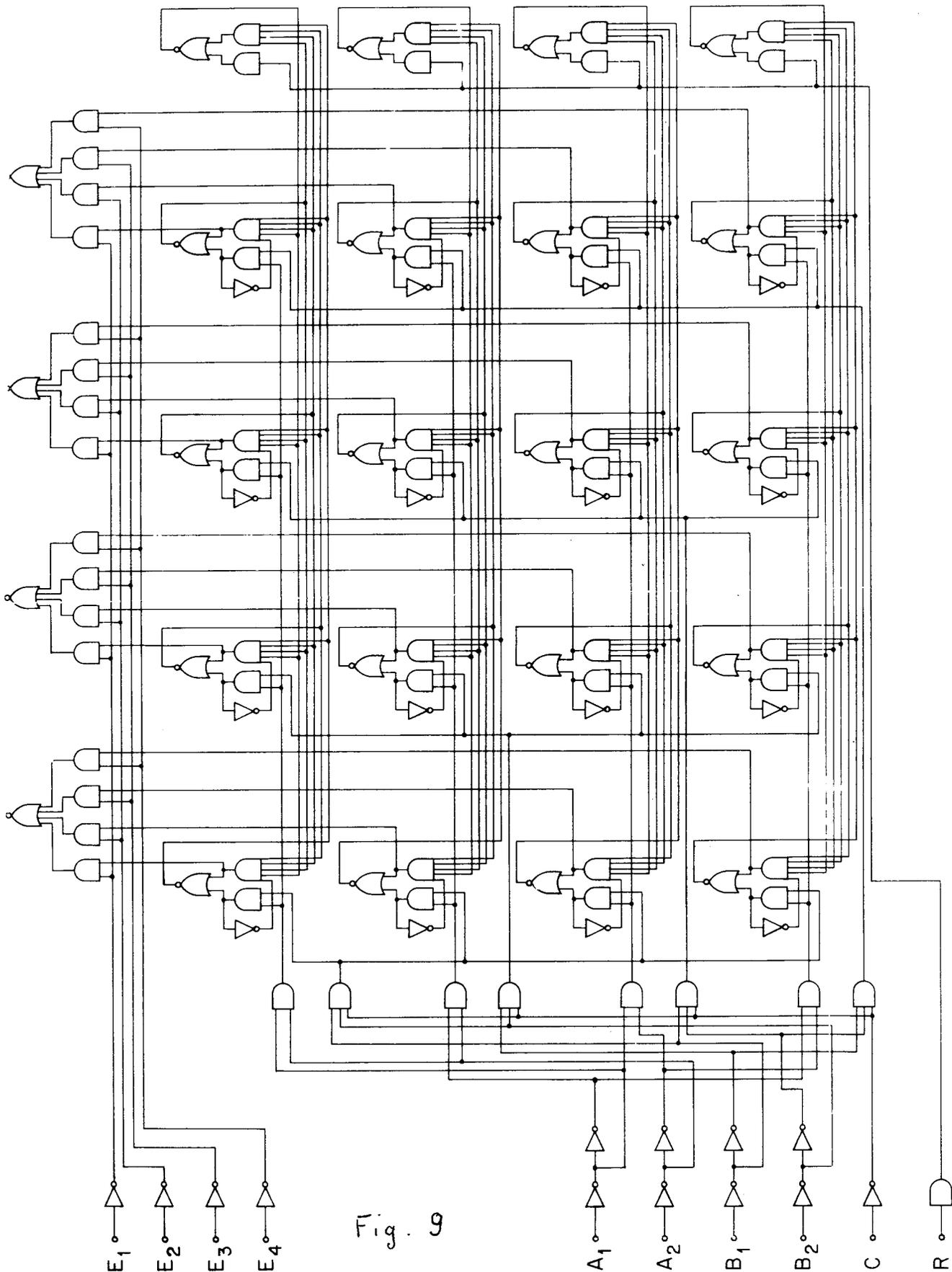


Fig. 59

Par suite, il suffit de tester 10 pannes des sous-modules $\begin{matrix} j \\ i \end{matrix}$. Pour le multiplexeur et les démultiplexeurs, on teste seulement 11 pannes. En résumé, on ne regarde le test que de 21 pannes au lieu de 296 Δ

II - ITERATION ET TEST

Le concept de récurrence conduit à la notion d'itération qui peut se faire soit sur des modules identiques (itération simple), soit sur des modules différents (itération complexe). La méthode de test direct proposée consiste à tester chaque module en observant sa sortie primaire, ce qui donne rapidement un test de tout le réseau en opérant au maximum en parallèle et permet de tester les pannes multiples.

II - 1. ITERATION SIMPLE

On considère un réseau de cellules identiques, chaque cellule étant formée de deux modules : A et B. Le module A a comme entrées les entrées primaires X de la cellule, et les sorties Q (états) de la cellule précédente. Les sorties de A sont entrées de B et de la cellule suivante.

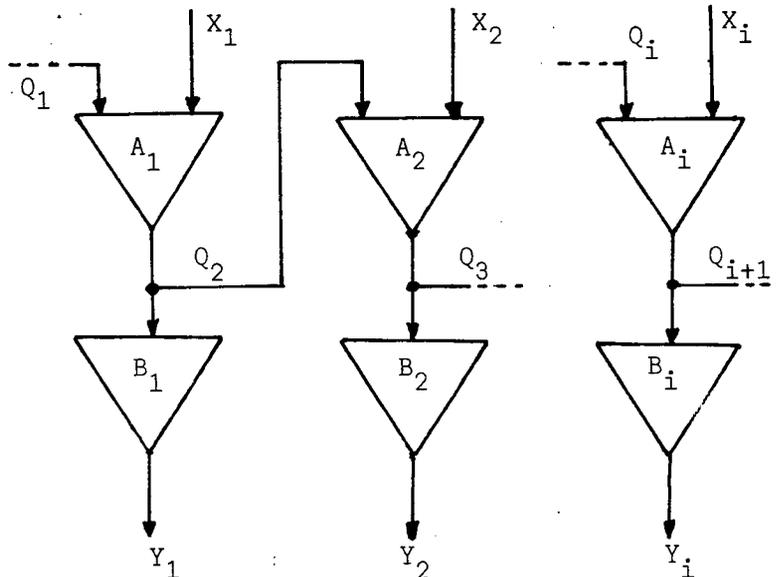


Figure 10

Tout réseau itératif peut se ramener au schéma de la figure 10 : en effet, si une cellule a la forme générale avec trois modules (figure 11), on pose $A'_i = A_i \cup C_{i-1}$ et $B'_i = B_i$ (figure 12). Avec cette transformation, les états de la nouvelle description ne sont plus ceux de l'ancienne. Ceci revient à faire la transformation machine de Mealy, machine de Moore.

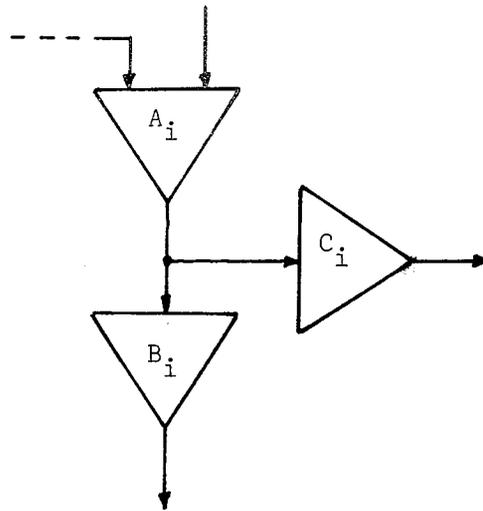


Figure 11

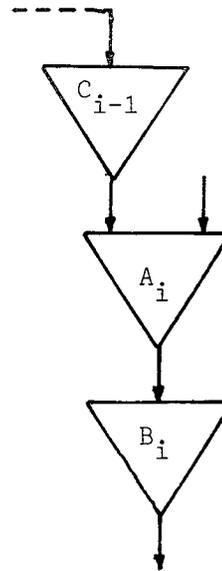


Figure 12

Chaque cellule M_i est caractérisée par les fonctions de transition de sortie suivantes :

$$\begin{aligned} \mu : Q \times X &\rightarrow Q && \text{telles que} && \mu(q_i, x_i) = q_{i+1} \\ \lambda : Q \times X &\rightarrow Y && && \lambda(q_i, x_i) = y_i \end{aligned}$$

DEFINITION

La panne α d'une cellule est directement détectable s'il existe $q \in Q$ et $x \in X$ tels que $\lambda_\alpha(q, x) \neq \lambda(q, x)$ \circ

DEFINITION

Le graphe de test est le graphe de transition entre états où, à un arc représentant la transition (q, x) est associée la marque formée de toutes les pannes directement détectées par cette transition. \circ

THEOREME

S'il existe dans le graphe de test un ensemble de cycles $(C_k)_{k \in K}$ tel que toute panne directement détectable soit une marque d'au moins un arc d'un des cycles, alors le réseau peut être testé avec un nombre de tests

$$n = \sum_{k \in K} |C_k| \text{ (indépendant du nombre de cellules).}$$

De plus, les pannes multiples sont aussi détectées et la première cellule en panne est localisée.

DEMONSTRATION

Supposons le cycle C_1 passant par les états $(q_1, q_2, \dots, q_p, q_1)$ et la suite des entrées (x_1, \dots, x_p) telles que $\mu(q_i, x_i) = q_{i+1}$ et $\mu(q_p, x_p) = q_1$.

Appliquons au réseau le test $t_1 = (q_1, x_1, x_2 \dots x_p, x_1 \dots)$ qui affecte aux entrées primaires des cellules $M_{i \pmod{p}}$ les valeurs x_i . Les pannes directement détectées par la transition $(q_i, x_i)_{i=1, \dots, p}$ le sont sur les modules M_i, M_{p+i}, \dots . Si on applique au réseau les p tests $t_1 = (q_1, x_1 \dots x_p, x_1), t_2 = (q_2, x_2 \dots x_p, x_1, x_2 \dots), \dots$
 $t_p = (q_p, x_p, x_1, x_2 \dots x_{p-1}, x_p, x_1 \dots)$, toutes les pannes figurant dans le cycle C_1 sont testées pour chaque cellule. Tout le réseau est testé en faisant de même avec les cycles C_1, \dots, C_k \square

REMARQUE

Toute panne détectable de B est détectable par cette méthode. De plus, si B réalise une bijection, alors toute panne détectable de A est directement détectable.

Caractérisation des réseaux itératifs où le test direct ne dépend pas du nombre de cellules.

Soit (q_i, x_i) une transition. Nous définissons les propriétés :

$p_1(q_i, x_i) \iff$ il existe un cycle C auquel (q_i, x_i) appartient.

$p_2(q_i, x_i) \iff p_1(q_i, x_i)$ est faux et il existe un cycle C et une chaîne partant d'un sommet de C et aboutissant à q_i .

THEOREME

Soit α une faute détectée par un ensemble de transitions $\{(q_i, x_i)\}_{i \in I}$

- 1) S'il existe i tel que $p_1(q_i, x_i)$ soit vrai, alors le test de α ne dépend pas du nombre n de cellules.
- 2) Si pour tout i , $p_1(q_i, x_i)$ et $p_2(q_i, x_i)$ sont faux, alors le test de α devient indépendant de n pour n supérieur à un nombre $k(\alpha)$.
- 3) Si pour tout i , $p_1(q_i, x_i)$ est faux et s'il existe un j tel que $p_2(q_j, x_j)$ soit vrai, le test de α est indépendant de n \square

EXEMPLE

Soit le réseau suivant transformé d'un exemple décrit dans [2] :

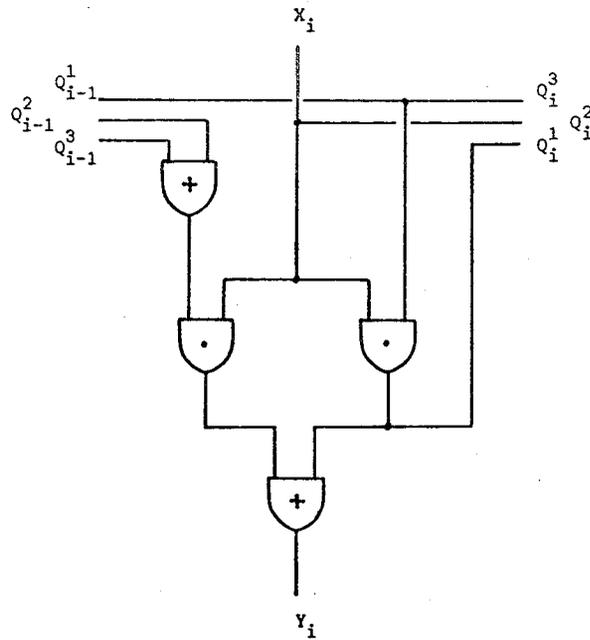


Figure 13

Les transitions sont décrites par les équations booléennes suivantes et par le tableau de transition (où $q = (q_i^1 q_i^2 q_i^3)$ est noté par son équivalent décimal)

q	0	1	2	3	4	5	6	7
X	0	0	0	0	2	2	2	2
1	1	5	5	5	3	7	7	7

$$q_i^1 = x_i \cdot (q_{i-1}^2 + q_{i-1}^3)$$

$$q_i^2 = q_{i-1}^1$$

$$q_i^3 = x_i$$

Le graphe de test est donné par la figure 14 :

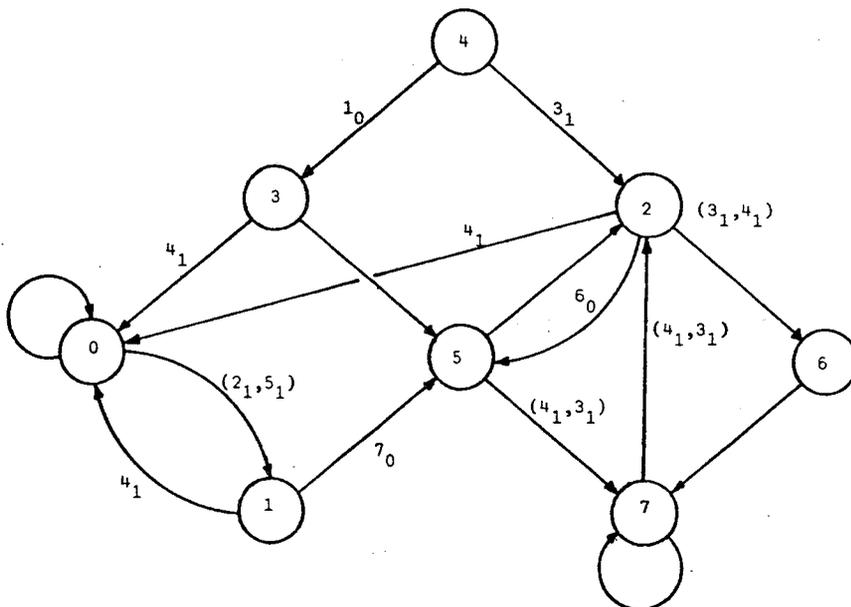


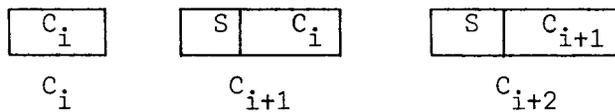
Figure 14

On voit que la panne 1_0 ne pourra être testée que dans le premier module. En considérant les cycles $C_1 = (0,1,5,2)$ pour les pannes $\{2_1, 7_0, 4_1, 3_1, 5_1\}$ et $C_2 = (2,5)$ pour $\{6_0\}$, on détectera tous les collages multiples avec 7 vecteurs de test.

Pour les tests de pannes qui ne modifient pas la structure itérative, il faut trouver un ensemble de cycles couvrant les arcs des composantes fortement connexes, par exemple $\{C'_1 = (2,5,7,7) ; C'_2 = (2,0,1,5) ; C'_3 = (0,0,1)\}$, ce qui donne 11 tests. Pour vérifier les transitions des premiers modules qui ne sont pas sur des composants fortement connexes, il faut rajouter les tests $(4\ 3\ 5\ \emptyset\ \dots)$ $(4\ 3\ 0\ \emptyset\ \dots)$ $(4\ 2\ \emptyset\ \dots)$ $(6,2,\emptyset\dots)$ $(6,7,\emptyset\dots)$.

II - 2. ITERATION COMPLEXE

On considère maintenant un réseau de cellules non identiques C_i constituées de deux modules A_i et B_i interconnectés comme précédemment et réalisant les fonctions f_i et g_i . De plus, il existe une relation de récurrence entre les f_i et les g_i , de telle sorte que C_{i+1} soit construite en ajoutant au réseau réalisant C_i un réseau donné S indépendant de i .



Soit T_{C_i} un test de C_i ; un test de C_{i+1} est $T_{C_{i+1}} = T_{C_i}^S \cup T_S^{C_i}$ (où T_X^Y représente un test de X dans le contexte Y) et est fonction de T_{C_i} . Si on connaît T_{C_0} et la relation de récurrence entre T_{C_i} et $T_{C_{i+1}}$, on peut alors calculer T_{C_i} .

EXEMPLE

Soit le circuit d'additionneur à retenue anticipée de la figure 15 :

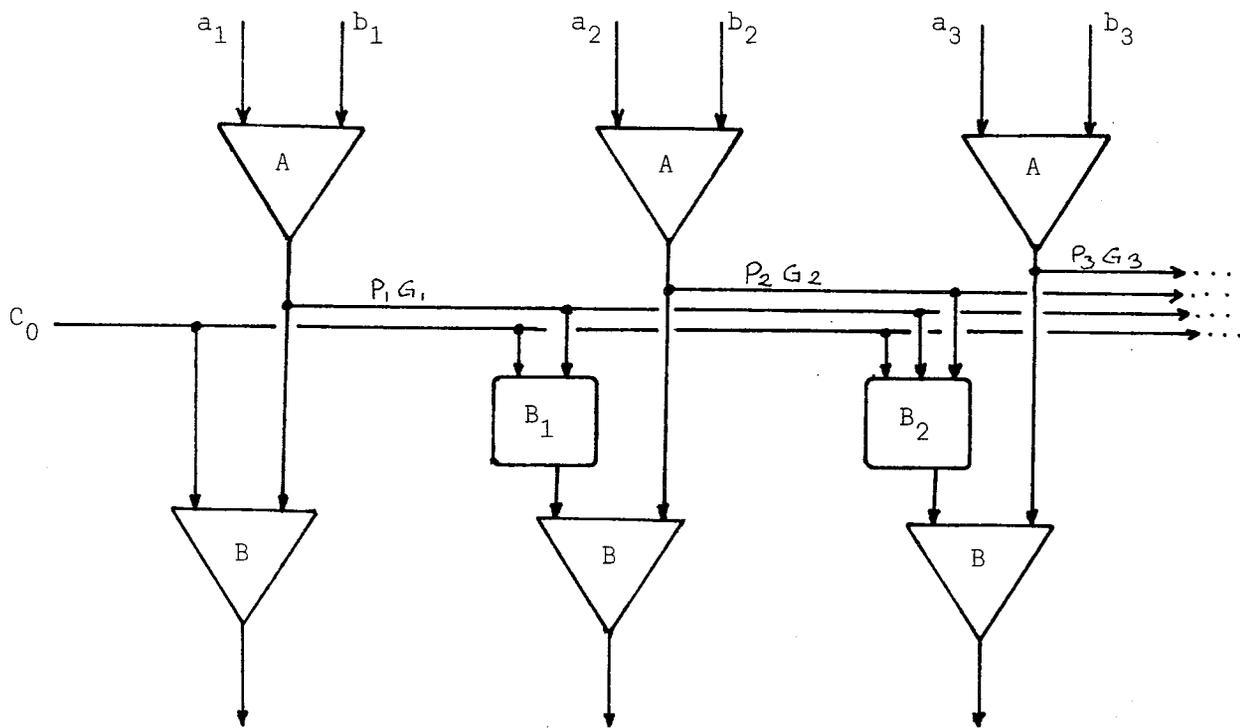


Figure 15

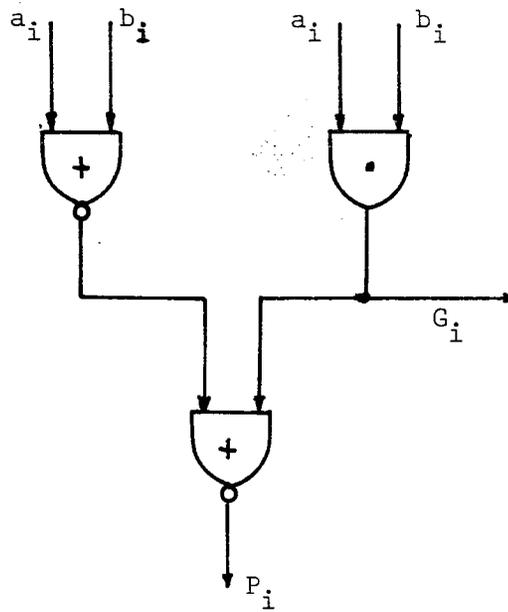


Figure 16

A est fixe et est représenté par la figure 16, ses sorties vers la cellule suivante de rang i sont P_i et G_i qui représentent une propagation ou une génération de retenue. B est fixe et réalise une disjonction ; R_i est représenté à la figure 16 bis.

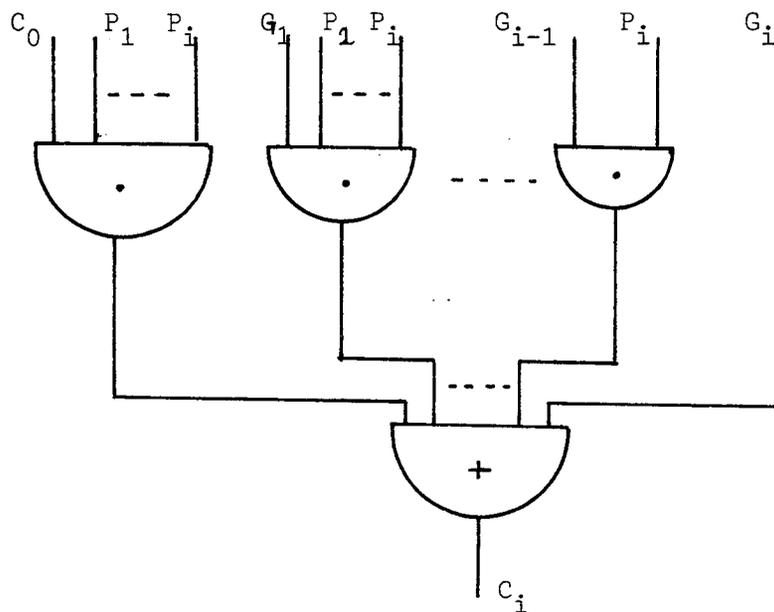


Figure 16 bis

La relation de récurrence entre T_{R_i} et $T_{R_{i-1}}$ est :

$$T_{R_i} = (T_{R_{i-1}}) \cdot P_i \cup_{0 < j < i} (G_j \cdot P_{j+1} \dots P_{i-1} \bar{P}_i \bar{G}_i) \cup G_i. \text{ On peut montrer que } T_{R_{i-1}} \text{ est inclus dans } T_{R_i}.$$

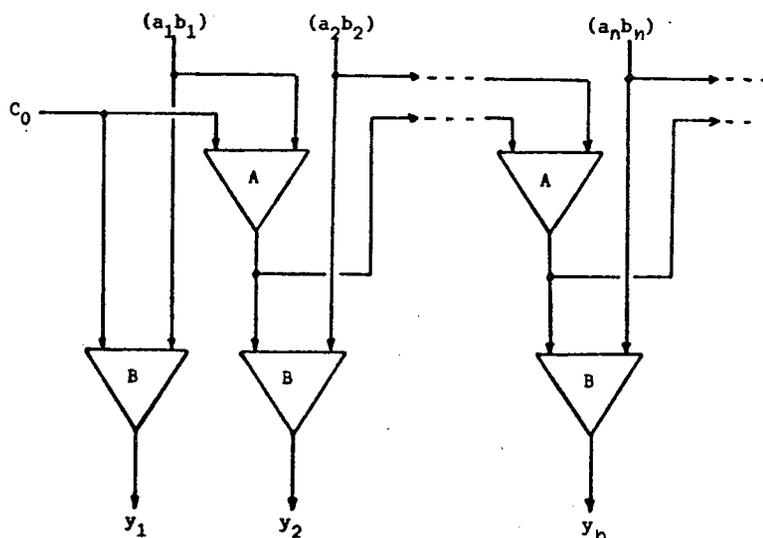
Tests comparés d'additionneurs à n étages

On recherche maintenant à évaluer le nombre de vecteurs de test nécessaires pour un additionneur, en fonction de n et selon le type et le niveau de description.

- 1) Test fonctionnel (boîte noire) : 2^{n+1} vecteurs de test sont nécessaires.
- 2) Additionneur sans retenue anticipée

On peut considérer 2 niveaux de description :

- Niveau 1



Si on ne dispose pas de renseignements sur la façon dont sont réalisés les modules A et B, il faut appliquer un test exhaustif à chaque cellule, soit $2^5 = 32$ vecteurs. Les 8 états possibles (C_{i-1}, a_i, b_i) peuvent être regroupés en 6 car, pour la propagation de la retenue $(C_{i-1}, 1, 0)$, est équivalent à $(C_{i-1}, 0, 1)$.

De même on peut ne considérer que 3 entrées (a_i, b_i) ; nous notons A,P,G les valeurs de ces entrées quand elles absorbent; propagent ou génèrent la retenue de l'étage précédent. Le tableau des états et un ensemble \mathcal{C} de cycles couvrant tous les arcs est donné ci-dessous :

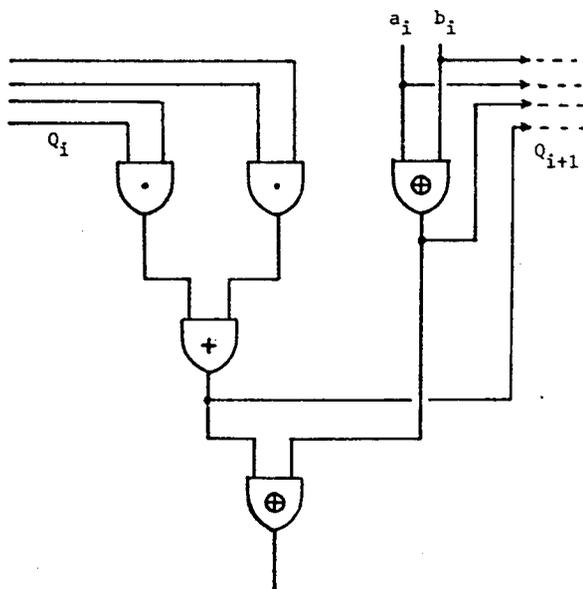
	Q \ X	A	P	G
(0,A) :	0	0	1	2
(0,P) :	1	0	1	2
(0,G) :	2	3	4	5
(1,A) :	3	0	1	2
(1,P) :	4	3	4	5
(1,G) :	5	3	4	5

$$\mathcal{C} = \{(0) ; (1) ; (4) ; (5) ; (0,1) ; (2,3) ; (4,5) ; (0,2,4,3) ; (1,2,5,3)\}$$

D'où les tests : (0AA...) ; (0PP...) ; (1PP...) ; (1GG...) ; (0APAP...) ; (0PAPA...) ; (0GAGA...) ; (1AGAG...) ; (1PGPG...) ; (1GPGP...) ; (0AGPAAGPAA...) ; (0GPAAGPAA...) ; (1PAAGPAAG...) ; (1AAGPAAGP...) ; (0PGGAPGGA...) ; (0GGAPGGAP...) ; (1GAPGGAPG...) ; (1APGGAPGG...).

- Niveau 2

Supposons qu'on connaisse une réalisation (cf. Figure 12) et que l'on teste les collages. Dans ce cas, les 5 cycles $\{(0), (1), (4), (5), (2,3)\}$ recouvrent tous les collages et 8 tests suffisent.



$$T = \{(0PP...) ; (1PP...) ; (1AGA...) ; (0GAG...) ; (0AA...) ; (1GG...)\}$$

Figure 12

3) Additionneur avec retenue anticipée

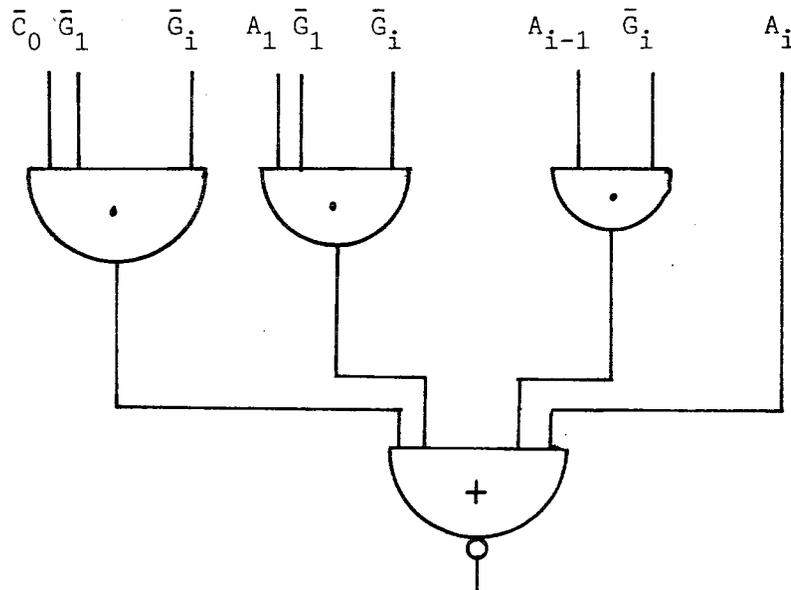
- Niveau 1

On reprend le schéma de la figure 9. Le test de la cellule C_i nécessite $2^3 3^{i-1}$ vecteurs de tests et peut tester en même temps C_1, C_2, \dots, C_{i-1} .

- Niveau 2

Le module calculant la retenue peut avoir plusieurs réalisations. Une première réalisation (a) est celle de la figure 11. Pour tester le module R_i , il faut $\frac{i(i+1)}{2} + i + 2$ tests.

Une deuxième réalisation (b) est donnée ci-dessous :



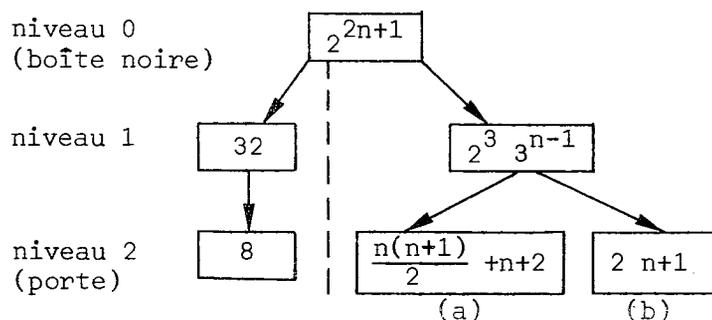
Si $T_{R_{i-1}}$ est le test de R_{i-1} , le test de R_i est :

$$T_{R_i} = (T_{R_{i-1}}) \cdot P_i \cup C_0 A_1 \dots A_{i-1} \bar{G}_i \cup C_0 P_1 \dots P_{i-1} A_i. \text{ Il faut donc } 2i+1 \text{ tests.}$$

Le schéma suivant donne le nombre de vecteurs de tests en fonction du nombre d'étages n et du niveau de description.

Additionneur

sans retenue anticipée avec retenue anticipée



Si, pour trouver un test court d'un additionneur sans retenue anticipée, il n'est pas nécessaire d'avoir une description au niveau de la porte, cela l'est par contre pour la retenue anticipée, sinon le nombre de tests est exponentiel.

CONCLUSION

Les concepts d'isomorphisme et d'itération simplifient la recherche d'un test et conduisent à une expression fonctionnelle du test. Le test fonctionnel ne tenant pas compte de la réalisation, n'est qu'approximatif ; le test purement analytique, ne considérant pas l'aspect fonctionnel, donne des algorithmes très lourds pour les systèmes complexes ; au contraire, ici on a développé un test structurel qui allie les avantages des tests fonctionnels et analytiques en éliminant leurs défauts et permet de tester des systèmes de grande complexité.

BIBLIOGRAPHIE

- [1] BOSEN and HONG : "Cause-effect analysis for multiple fault detection in combinatorial networks", IEEE Trans on Comp., vol. C 20, n° 11, November 1971.

- [2] FRIEDMAN and MENON : "Fault detection in digital circuits", Prentice Hall, 1971.

- [3] G. SAUCIER : "Un algorithme efficace recherchant l'isomorphisme de 2 graphes", RAIRO R3, 1971 (revue rouge de l'AFCT, France).

- [4] C. TURCAT : "Isomorphismes, immersion et recouvrement de graphes", Thèse 3ème Cycle, Grenoble 1974.

- [5] A. VERDILLON : "Pannes dans les réseaux acycliques". Thèse 3ème Cycle, Grenoble 1972 et RAIRO Juillet 1974 (AFCT, Paris).

CHAPITRE IV

TEST DES PANNES MULTIPLES

CHAPITRE IV : TEST DES PANNES MULTIPLES

I - Simulation des pannes

I - 1. Rappels sur la simulation des pannes simples

I - 2. Représentation des pannes multiples

I - 3. Simulation des pannes multiples

II - Test des pannes multiples

II - 1. Relations entre pannes multiples

II - 2. Méthode de test

III - Simulation des pannes multiples dans le séquentiel

INTRODUCTION

La difficulté d'engendrer un test pour les pannes multiples vient de leur grand nombre (s'il y a n connexions, il y a 3^n collages multiples). Il n'est donc pas possible d'envisager d'utiliser les méthodes de synthèses de test des pannes simples. Heureusement, un test des pannes simples teste aussi la plupart des pannes multiples. Une description d'un ensemble de pannes multiples par une somme de monômes permet de simuler l'ensemble des pannes multiples détectées par un test. La recherche d'un test des pannes multiples se fait alors en identifiant les pannes simples qui sont détectées même en présence d'autres pannes.

I - SIMULATION DES PANNES

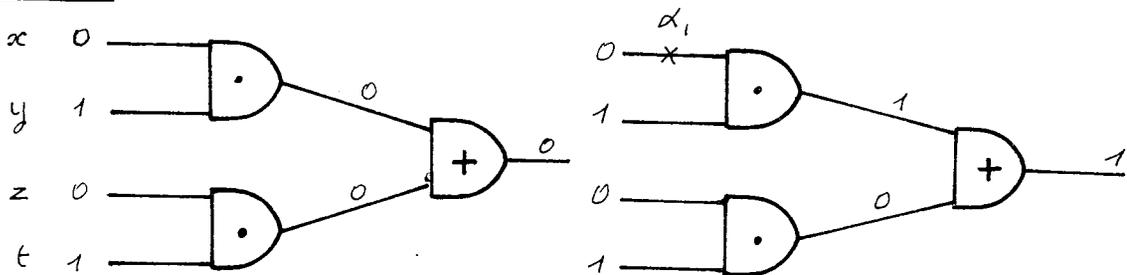
Cette étude étend la méthode de simulation de liste de pannes simples [Amstrong] au cas des pannes multiples.

I - 1. RAPPELS SUR LA SIMULATION DES PANNES SIMPLES

a) Simulation du circuit en présence d'une panne donnée :

Cette méthode consiste à calculer la réponse du circuit sans panne pour un vecteur d'entrée et à calculer la réponse du circuit avec une panne spécifiée, pour le même vecteur d'entrée. La panne est détectée par le test si la sortie est différente.

EXEMPLE

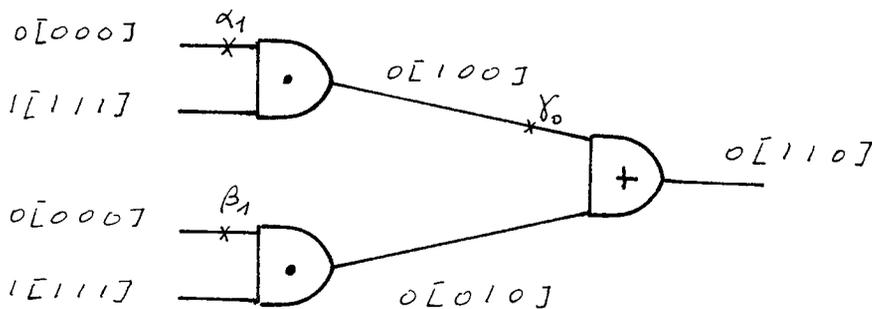


la panne α collé à 1 (α_1) est détectée par le test : $\begin{matrix} x & y & z & t \\ 0 & 1 & 0 & 1 \end{matrix}$ Δ

Cette méthode est très longue si on veut pouvoir simuler plusieurs pannes. Si on veut simuler plusieurs pannes pour un même vecteur d'entrée on peut les simuler en parallèle.

EXEMPLE

Soit le circuit précédent avec les 3 pannes $\alpha_1, \beta_1, \gamma_0$.



le vecteur d'entrée 0 1 0 1 détecte les pannes α_1 et β_1 mais ne détecte pas γ_0 Δ

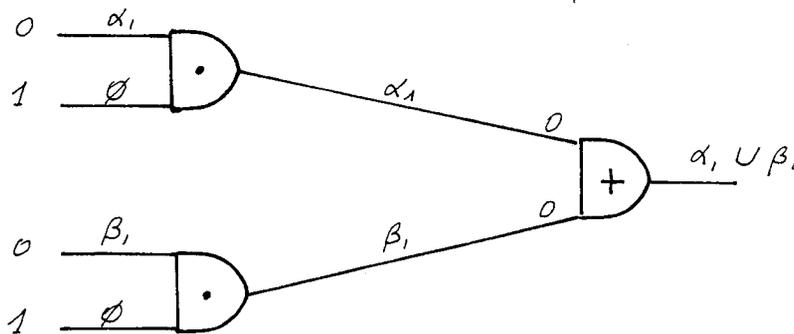
b) Simulation de liste de pannes

Dans cette méthode on cherche à trouver toutes les pannes détectées par un test donné. Elle est basée sur le fait que connaissant les listes de pannes propagées sur les entrées d'un opérateur, on peut en déduire la liste des pannes propagées en sortie. Pour les portes ET à 2 entrées les règles sont les suivantes :



Pour un opérateur OU les règles sont duales, pour un opérateur négation la liste des pannes est propagée.

EXEMPLE



I - 2. REPRESENTATION DES PANNES MULTIPLES

Notons P l'ensemble des pannes simples d'un circuit et M l'ensemble des pannes multiples. En général, certaines pannes simples sont incompatibles (ne peuvent pas se produire en même temps), donc $M \subseteq 2^P$. Ceci est vrai en particulier pour les collages : le collage à 0 d'une connexion est incompatible avec le collage à un de la même connexion.

On peut décrire une panne multiple par un monôme canonique de B^n si $|P| = n$.

EXEMPLE

Soit un exemple de pannes simples $P = \{a, b, c, d\}$ la panne décrite par $\bar{a} b \bar{c} d$ représente la panne double formée des pannes simples b et d (a et c pas en panne). Une somme de monômes (pas forcément canoniques) représente un ensemble de pannes Δ

EXEMPLE

$a + c \bar{d}$ représente toutes les pannes multiples contenant la panne simple a comme composante, ainsi que toutes celles qui contiennent c sans contenir d. Les pannes multiples sont en fait décrites pour les monômes canoniques de la fonction $a + c \bar{d}$ écrite sous forme canonique (somme de produits) Δ

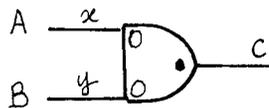
I - 3. SIMULATION DES PANNES MULTIPLES

Pour simuler les pannes multiples l'extension de la méthode a) est impossible du fait du grand nombre de pannes multiples. Par contre la méthode b) de propagation de listes peut être étendue assez simplement. Comme pour les pannes simples on décrit d'abord la simulation des portes élémentaires.

I - 3.1. Simulation des pannes multiples pour les portes

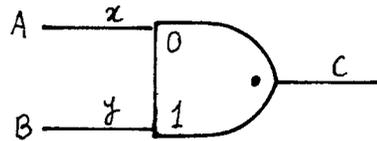
On se place dans le cas où les pannes sont des collages. Du fait des équivalences entre pannes on peut ne considérer que les collages à 1 des entrées d'une porte ET et les collages à 0 des entrées d'une porte OU ; les opérateurs négation sont transparents du point de vue des listes de pannes.

a) Porte ET avec 2 entrées à 0 : x et y



A et B sont les listes de pannes multiples arrivant sur les connexions x et y et C est la liste de panne en sortie de la porte. On a $C = (A+x)(B+y) = AB + Ay + Bx + xy$ et $C = \bar{A}\bar{x} + \bar{B}\bar{y}$. En fait la panne xy (les connexions collées x et y collées à 1) n'est pas prise en compte car elle est équivalente à la sortie collée à 1. L'expression des pannes multiples en C se simplifie donc en $C = AB + Ay + Bx$.

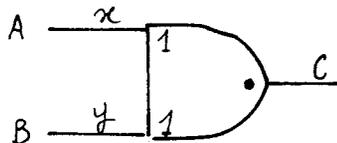
b) Porte ET avec une entrée à 0 l'autre à 1



$$C = (A+x)(\bar{B}+y) = A\bar{B} + Ay + \bar{B}x + xy$$

Cette expression se simplifie comme précédemment en $C = A\bar{B} + Ay + \bar{B}x$.
Le complémentaire est $\bar{C} = \bar{A}x + B\bar{y}$.

c) Porte ET avec les deux entrées à 1



$$C = A\bar{x} + B\bar{y}$$

$$\bar{C} = (\bar{A}+x)(\bar{B}+y) = \bar{A}\bar{B} + \bar{A}y + \bar{B}x + xy$$

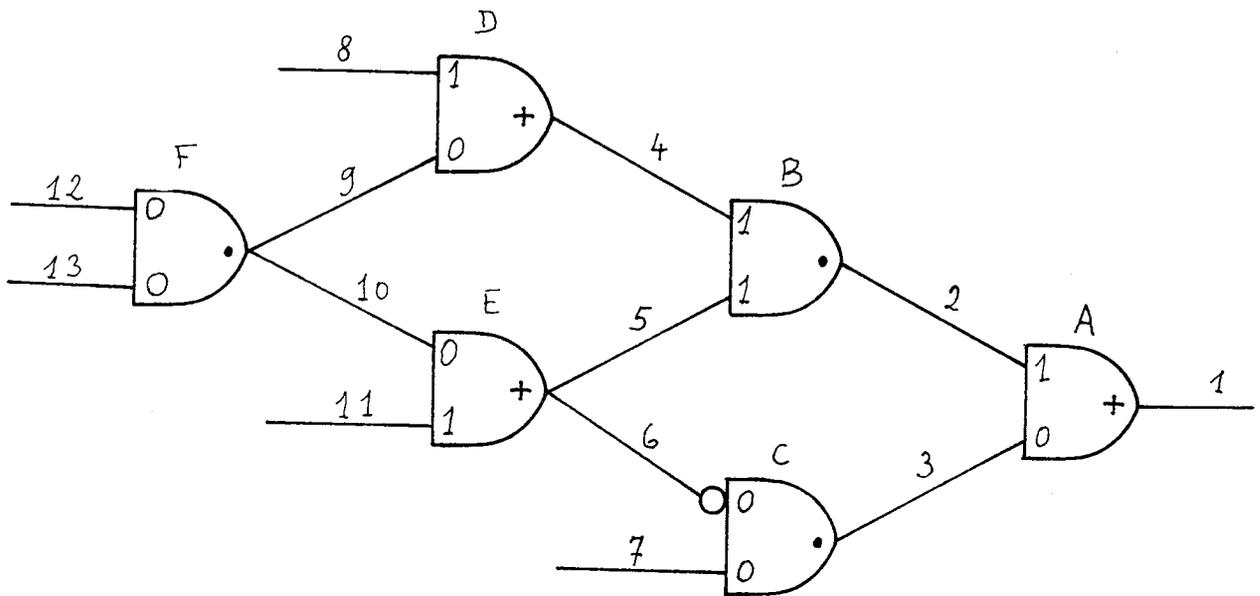
L'expression de \bar{C} se simplifie en $\bar{C} = \bar{A}\bar{B} + \bar{A}y + \bar{B}x$.
L'étude des portes OU est duale. S'il y a plus de 2 entrées on le traite comme un arbre de portes à 2 entrées.

I - 3.2. Simulation des pannes multiples dans un réseau

A l'aide de la simulation pour les portes on calcule de proche en proche les listes des pannes propagées. On peut soit calculer les listes de pannes propagées en commençant par les entrées, soit en commençant par les sorties. Contrairement à la simulation des pannes simples pour laquelle la simulation en partant des sorties est préférable, ici les 2 méthodes sont équivalentes.

EXEMPLE

Soit le réseau suivant avec les valeurs des connexions indiquées à l'intérieur des portes.



$$F = \emptyset$$

$$E = 11 \cdot \bar{F} = 11$$

$$D = 8 \cdot \bar{F} = 8$$

$$B = D\bar{4} + E\bar{5} = 8\bar{4} + 11\bar{5}$$

$$C = E \cdot 7 = 11 \cdot 7 \quad \bar{C} = \bar{11} + \bar{7}$$

$$A = 2\bar{C} + B\bar{C} + B \cdot 3$$

$$= 2(\bar{11} + \bar{7}) + (8\bar{4} + 11\bar{5})(\bar{11} + \bar{7}) + (8\bar{4} + 11\bar{5}) \cdot 3$$

La liste des pannes multiples détectées est donc :

$$A = \underline{2 \cdot \bar{11} + 2 \cdot \bar{7} + 8 \cdot \bar{4} \cdot \bar{11} + 8 \cdot \bar{4} \cdot \bar{7} + 11 \cdot \bar{5} \cdot \bar{7} + 3 \cdot 8 \cdot \bar{4} + 3 \cdot 11 \cdot \bar{5}}$$

Si le réseau est un arbre les fonctions précédentes se simplifient car pour les classes d'équivalence des pannes multiples on ne considère pas toutes les 2^n pannes (n est le nombre de connexions). En effet l'étude des "écoulements" montre qu'il suffit de considérer 2^q pannes ou q est le nombre d'entrées primaires. Une représentation de ces 2^q pannes peut se faire en ne considérant pas les pannes dont une des composante se trouve sur le chemin entre une autre composante et la sortie. Ceci revient à remplacer $A\bar{x}$ et $\bar{B}y$ respectivement par A et B .

Ce qui donne dans les différents cas d'une porte ET :

Porte ET avec 2 entrées à 0 :

$$C = AB + Ay + Bx$$

$$\overline{C} = \overline{Ax} + \overline{By}$$

Porte ET avec une entrée à 0 et l'autre à 1 :

$$C = A\overline{B} + Ay + \overline{B}x$$

$$\overline{C} = \overline{Ax} + B$$

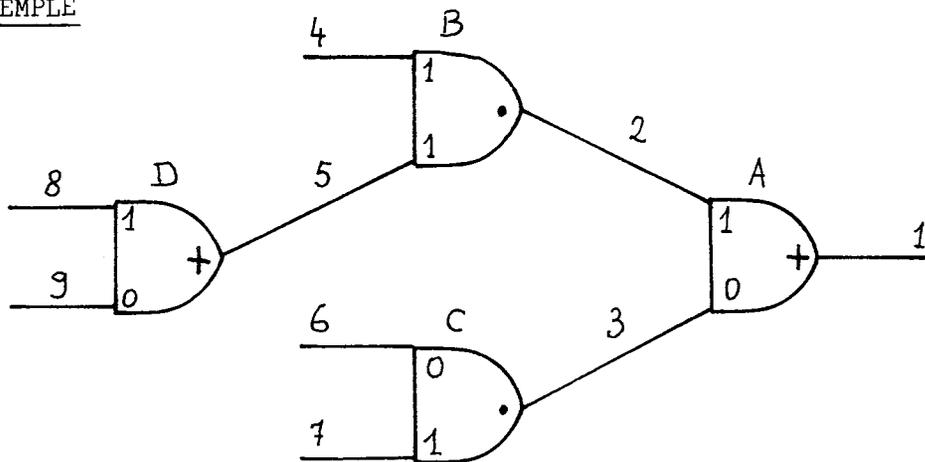
Porte ET avec les 2 entrées à 1 :

$$C = A + B$$

$$\overline{C} = \overline{A} \overline{B}$$

Pour les portes OU les règles sont duales.

EXEMPLE



$$D = 8$$

$$B = D$$

$$C = 6$$

$$A = B \overline{C} + B 3 + 2 \overline{C}$$

$$\text{donc : } \underline{A = 8 \cdot \overline{6} + 8 \cdot 3 + 2 \cdot \overline{6}}$$

Les simplifications énoncées plus haut peuvent se généraliser au circuit qui ne sont pas des arbres comme suit. On ne garde des pannes multiples que celles dont toute composante est reliée à la sortie par au moins un chemin sans panne (R classes). On peut donc pour chaque panne simple α mémoriser les coupures de l'écoulement associé à la panne α .

Si on reprend l'exemple page 7

B se simplifie en $B = 8 + 11.\bar{5}$

Enfin A devient :

$$A = \underline{2.\bar{11} + 2.\bar{7} + 8.\bar{11} + 8.\bar{7} + 11.\bar{5}.\bar{7} + 3.8.\bar{4} + 3.11}$$

L'explication de la simplification du $3.11.\bar{5}$ en 3.11 est que les connexions 3.5 forment une coupure de l'écoulement du 11 donc la panne 3.5.11 n'est pas considérée (équivalente à 3.5).

II - TEST DES PANNES MULTIPLES

La règle générale pour tester les pannes simples est de chercher un test pour une panne, regarder ensuite les autres pannes qui sont aussi détectées, rechercher un test pour une panne non encore détectée et ainsi de suite jusqu'à ce que toutes les pannes aient été détectées. Comme le nombre de pannes multiples est très important cette approche n'est évidemment pas utilisable pour le test des pannes multiples. L'approche suivie ici consiste à rechercher les tests des pannes simples et d'en déduire par simulation les pannes multiples détectées.

II - 1. RELATIONS ENTRE PANNES MULTIPLES

Nous avons vu qu'à un test t_i peut être associé une fonction booléenne M_i qui décrit l'ensemble des pannes multiples détectées. Pour un ensemble de tests $T = \{t_1, t_2 \dots t_p\}$; la fonction booléenne décrivant les pannes

détectées par T est $M = \sum_{i=1}^P M_i$

Définition

On dit qu'une panne α détecte toutes les pannes multiples la contenant (DPMC) pour un test T si toute panne $\alpha \beta$ est détectée par T \circ

Définition

On dit que la panne simple α couvre la panne simple β si

- 1) il existe un seul chemin entre α et β
- 2) s'il existe un nombre pair d'inverseurs entre α et β alors α et β sont des collages de même nature et si le nombre d'inverseurs est impair, les collages sont de nature différente \circ

PROPRIETE

Si α couvre β alors β cache α et $T_\alpha \subset T_\beta$ \square

THEOREME

Si α couvre β et si α est DPMC pour T, alors β est DPMC pour T.

dem : Soit α DPMC pour T, et supposons β non DPMC pour T. Alors il existe γ telle que $\beta \gamma$ ne soit pas détectée par T, γ masque β pour T. Le masquage se fait sur la portion du circuit comprise entre β et la sortie donc le masquage se fait aussi pour α donc α n'est pas DPMC pour T ce qui est contraire à l'hypothèse \square

COROLLAIRE

Soit $P = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ un ensemble de pannes simples.

Si α_i couvre α_j alors tout test des pannes multiples $2^{P-\{\alpha_j\}}$ est aussi un test des pannes multiples 2^P .

dem : Soit T un test de $2^{P-\{\alpha_j\}}$, T est un test pour lequel α_i est DPMC d'après le théorème précédent β est aussi DPMC pour T donc T test aussi 2^P \square

Ce corollaire permet de *limiter* les pannes multiples à considérer en éliminant comme composante toutes les pannes couvertes.

PROPRIETE

Pour que α soit DPMC pour le test T il faut qu'il y ait dans T un test ou apparaisse un monôme de la forme $\alpha \beta_1 \beta_2 \dots \beta_n$.

dem : Si tous les monômes où α apparait sont de la forme $\alpha \gamma_1 \gamma_2 \dots \gamma_k \delta_1 \delta_2 \dots \delta_l$, il ne peut pas y avoir de simplification à α . Ceci est dû en particulier au fait qu'il n'y a jamais de monômes ayant uniquement des variables complémentées car cela signifierait que la panne vide est testable \square

II - 2. METHODE DE TEST DES PANNES MULTIPLES

On ne cherche pas ici de test minimal des pannes multiples. En effet, même dans le cas de pannes simples, il faut rechercher tous les tests de chaque panne si l'on veut générer un test optimal.

Cependant sans chercher un test minimal on peut chercher un test des pannes multiples qui soit construit à partir d'un test des pannes simples. C'est cette optique qui est décrite ici.

1/ On considère comme composantes des pannes multiples, les pannes qui ne sont pas couvertes (cf Corollaire précédent).

2/ Pour chaque panne simple α on recherche un test t_α qui, si possible maximise les pannes multiples détectées. On calcule par simulation M_α , la fonction booléenne décrivant les pannes multiples détectées par t_α .

3/ Pour chaque test t on associe S_t , ensemble des pannes simples α_i qui apparaissent sous la forme $\alpha_i \bar{\alpha}_k \dots \bar{\alpha}_l$. Si une panne n'est associée qu'à un test celui-ci est dit obligatoire.

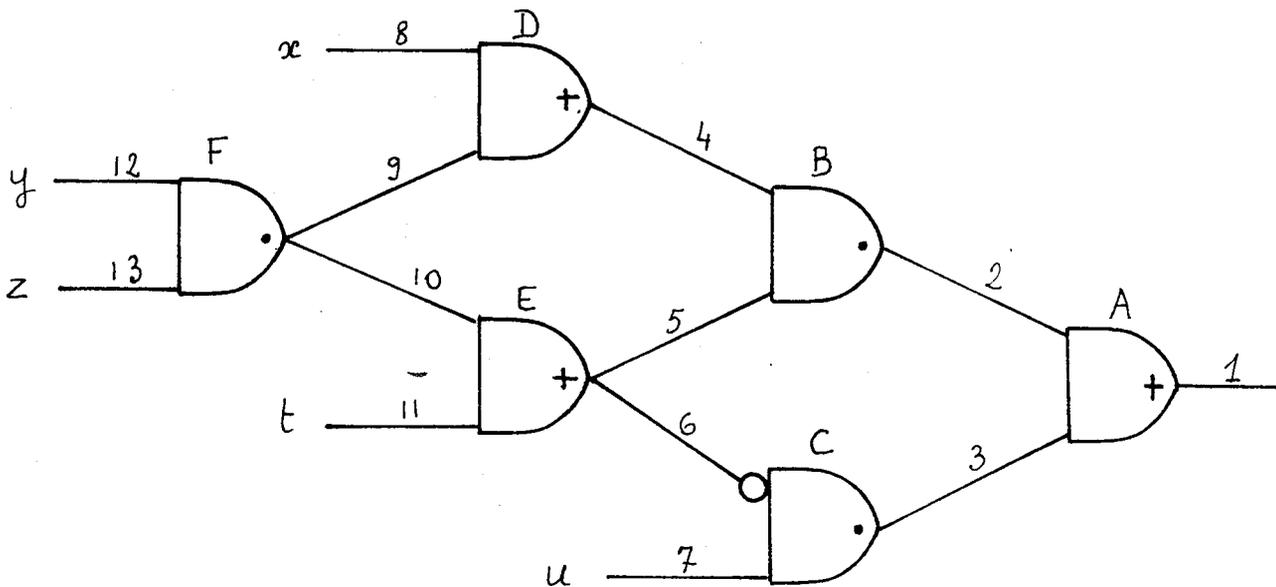
→ S'il y a des tests obligatoires soit $T^0 = \{t_1^0, \dots, t_p^0\}$ l'ensemble de ceux-ci.

4/ On construit la fonction $M^0 = \sum_{i=1}^P M_{t_i^0}$. Si A est l'ensemble des pannes DPMC pour T^0 on peut simplifier les tests qui ne sont pas obligatoire pour $M_t \rightarrow M'_t = M_t + M^0 - A$. On va en 5/.

→ S'il n'y a pas de tests obligatoires on choisit un ensemble des tests T^1 de façon que des pannes soient DPMC on calcule M^1 correspondant à T^1 . Si A est l'ensemble des pannes DPMC on simplifie les tests restant comme en 4/ : $M_t \rightarrow M'_t = M_t + M^0 - A$. On va en 5/.

5/ Si toutes les pannes sont testées (ou un pourcentage suffisant) on s'arrête, sinon on va en 3/.

Exemple : Soit le circuit suivant.



Le Corollaire précédent s'applique pour les pannes 1 et 2.
On ne considère donc que les pannes multiples ayant comme composantes 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 et 13.

Les fonctions de test des pannes simples sont les suivantes :

$$3_0 : (\bar{y} + \bar{z}) \bar{t} u$$

$$4_1 : \bar{x} (\bar{y} + \bar{z}) t$$

$$5_1 : x (\bar{y} + \bar{z}) \bar{t} \bar{u}$$

$$6_1 : \bar{x} (\bar{y} + \bar{z}) t u$$

$$7_1 : (\bar{y} + \bar{z}) \bar{t} \bar{u}$$

$$8_0 : x (\bar{y} + \bar{z}) t$$

$$9_0 : \bar{x} y z$$

$$10_0 : y z \bar{t} \bar{u}$$

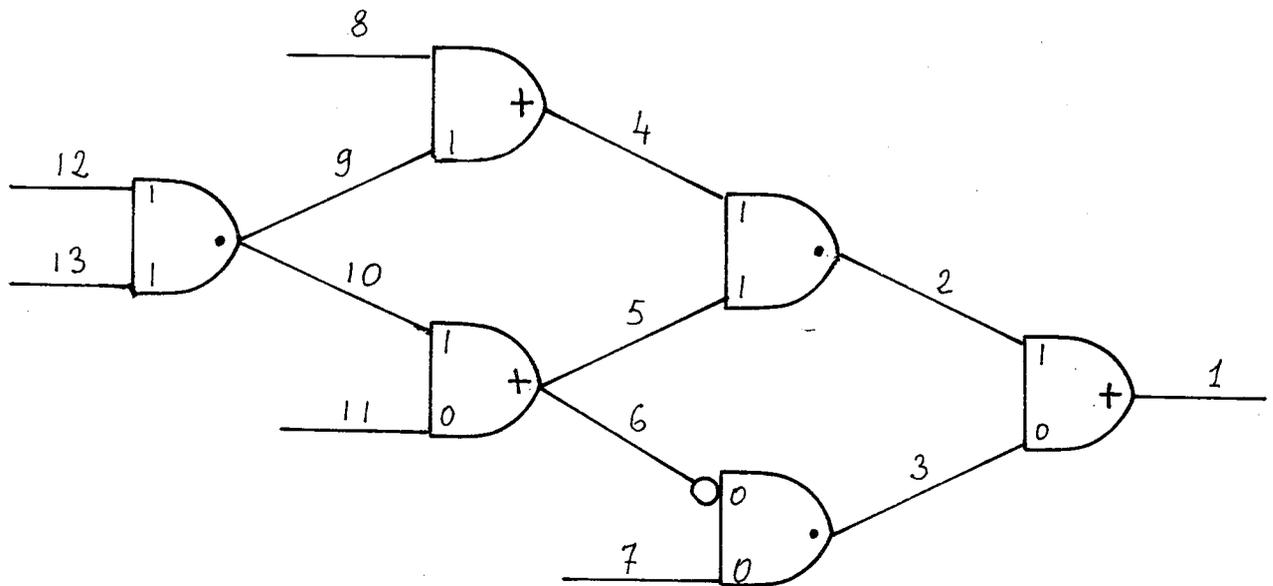
$$11_0 : x (\bar{y} + \bar{z}) t \bar{u} + \bar{x} (\bar{y} + \bar{z}) t u$$

$$12_1 : y \bar{z} \bar{t} \bar{u} + x y \bar{z} \bar{t} u + \bar{x} y \bar{z} t$$

$$13_1 : \bar{y} z \bar{t} \bar{u} + x \bar{y} z \bar{t} u + \bar{x} \bar{y} z t$$

Les pannes simples les moins testables sont $3_0, 5_1, 6_1, 8_0, 9_0, 10_0, 12_1, 13_1$.

Ensemble des pannes multiples testées quand 10 est testé.



$$A = B \bar{C} + B 3$$

$$B = D \bar{4} + E \bar{5}$$

$$C = E.7 ; \bar{C} = \bar{E} + \bar{7}$$

$$E = F + 10 ; \bar{E} = \bar{F} \bar{10}$$

$$F = 0$$

$$\text{d'où } E = 10 ; \bar{E} = \bar{10}$$

$$C = 10.7 ; \bar{C} = \bar{10}.7$$

$$B = D.\bar{4} + 10.\bar{5}$$

$$A = (D \bar{4} + E \bar{5}) (\bar{10} + \bar{7}) + 3 (D \bar{4} + E \bar{5})$$

Si $x = 1$ on a $D = 8$ $F = \emptyset$

Si $x = 0$ on a $D = 9 + F = 9$

$$P_M(x y z \bar{t} \bar{u}) = 10.\bar{5}.7 + 3.10$$

$$M_1 = P_M(\bar{x} y z \bar{t} \bar{u}) = 9.\bar{10} + 9.7 + 10.\bar{5}.7 + 3.9 + 3.10$$

On voit que $P_M(x y z \bar{t} \bar{u}) < P_M(\bar{x} y z \bar{t} \bar{u})$ donc on garde le test $\bar{x} y z \bar{t} \bar{u}$.

Pour tester 6 on garde le test $\bar{x} \bar{y} z t u$

les pannes multiples détectées sont :

$$M_2 = P_M(\bar{x} \bar{y} z t u) = 4.\bar{11} + 4.12 + 12.\bar{9} + 6 + 11.\bar{12}.\bar{3} + 4.\bar{3}.\bar{12} + 4.11.\bar{3} + 11.\bar{3}.\bar{9}$$

Pour le test de 8 on prend le test $x \bar{y} \bar{z} t \bar{u}$

$$M_3 = P_M(x \bar{y} \bar{z} t \bar{u}) = 3.8 + 8.\bar{7} + 8.\bar{11} + 3.11.\bar{5} + 11.\bar{5}.\bar{7}$$

Pour le test de 11 on prend $x \bar{y} \bar{z} t \bar{u}$

$$M_4 = P_M(x \bar{y} \bar{z} t \bar{u}) = 8.\bar{11} + 8.\bar{7} + 3.8 + 11.\bar{5}.\bar{7}$$

Pour le test de 6 on prend $\bar{x} y \bar{z} t u$

$$M_5 = P_M(\bar{x} y \bar{z} t u) = 6 + 4.\bar{11} + 4.13 + 5.13.\bar{9} + 13.\bar{9}$$

Pour tester 7 on prend $x \bar{y} \bar{z} \bar{t} \bar{u}$

$$M_6 = P_M(x \bar{y} \bar{z} \bar{t} \bar{u}) = 7 + 5.\bar{8}$$

Pour tester 3 on prend $\bar{x} \bar{y} \bar{z} \bar{t} u$

$$M_7 = P_M(\bar{x} \bar{y} \bar{z} \bar{t} u) = 3$$

Pour détecter la panne 3 T_7 est nécessaire, pour 7 il faut T_6 , pour 9 et 10 T_1 est nécessaire, pour 13 T_5 est nécessaire. D'où :

$$M_1 = T_1 + T_5 + T_6 + T_7$$

$$\begin{aligned} M_1 &= 3 + 6 + 7 + 9.\bar{10} + 9.\bar{7} + 10.\bar{5}.\bar{7} + 3.9 + 3.10 + 4.\bar{11} + 4.13 + 5.13.\bar{9} + 13.\bar{9} + 5.\bar{8} \\ &= 3 + 6 + 7 + 9 + 13 + 10.\bar{5} + 4.\bar{11} + 5.\bar{8} \end{aligned}$$

On voit que $A = \{3, 6, 7, 9, 13\}$ est l'ensemble des pannes DPMC pour T_1 . Ceci permet de simplifier M_2, M_3, M_4 en M'_2, M'_3, M'_4 ($M'_i = M_i + T_1 - A$)

$$M'_2 = 4 + 11 + 12$$

$$M'_3 = 8 + 11.\bar{5}$$

$$M'_4 = 8 + 11.\bar{5}$$

Si on fait $M = M_1 + M'_2 + M'_3$ on obtient

$$M = 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13$$

On a donc obtenu un test de toutes les pannes multiples \triangle

III - SIMULATION DES PANNES MULTIPLES DANS LE SEQUENTIEL

DESCRIPTION

De façon similaire au cas des pannes multiples on peut créer une table de panne où l'indice d'un état est non pas une liste de pannes simples, mais une somme de monômes (décrivant un ensemble de pannes multiples).

SIMULATION

Soit la table de fluence de test :
 $x = i$

A	$A_{f_1}^1$	$B_{f_2}^1$	$C_{f_3}^1$...
B	$A_{f_1}^2$	$B_{f_2}^2$	$C_{f_3}^2$...
C	$A_{f_1}^3$	$B_{f_2}^3$	$C_{f_3}^3$...
⋮	...			

Supposons que l'état initial soit $A_{f_1}^0 \ B_{f_2}^0 \ C_{f_3}^0 \ \dots$

si l'entrée est $x = i$ l'état suivant est :

$A_{F_1} \ B_{F_2} \ C_{F_3} \ \dots$ avec

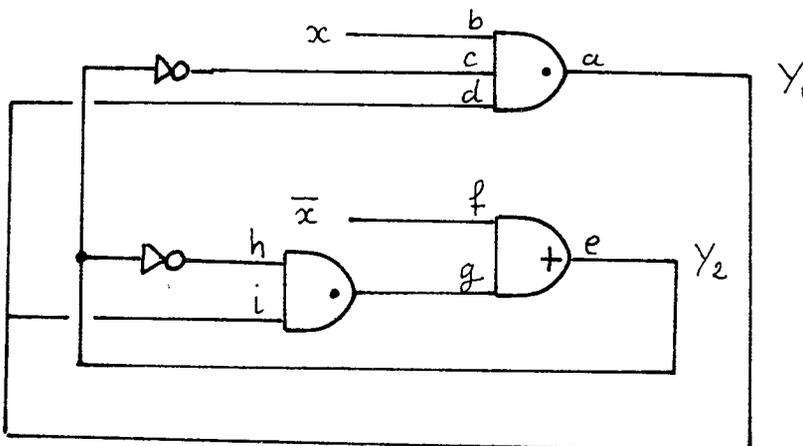
$$F_1 = f_1^0 f_1^1 + f_2^0 f_1^2 + f_3^0 f_1^3 + \dots$$

$$F_2 = f_1^0 f_2^1 + f_2^0 f_2^2 + f_3^0 f_2^3 + \dots$$

$$F_3 = f_1^0 f_3^1 + f_2^0 f_3^2 + f_3^0 f_3^3 + \dots$$

⋮
 ...

Exemple : Soit le circuit suivant :



Pour $x=0$ la table de test est la suivante :

A	$A_{f\bar{i}(\bar{b}+\bar{d})}$	$B_{(\bar{b}+\bar{d})(\bar{f}+i)}$	$C_{bd(\bar{f}+i)}$	$D_{bdf\bar{i}}$
B	$A_{fg\bar{b}}$	$B_{\bar{b}(\bar{f}+\bar{g})}$	$C_{b(\bar{f}+\bar{g})}$	D_{bfg}
C	$A_{f\bar{j}(\bar{b}+\bar{c})}$	$B_{(\bar{b}+\bar{c})(\bar{f}+j)}$	$C_{bc(\bar{f}+j)}$	$D_{bcf\bar{j}}$
D	$A_{f(\bar{h}+\bar{i})(\bar{b}+\bar{c}+\bar{d})}$	$B_{bcd\bar{f}(\bar{h}+\bar{i})}$	$C_{bcd(\bar{f}+hi)}$	$D_{(\bar{b}+\bar{c}+\bar{d})(\bar{f}+hi)}$

Si l'état initial est $A_{f\bar{i}(\bar{b}+\bar{d})}$ $B_{(\bar{b}+\bar{d})(\bar{f}+i)}$ $C_{bd(\bar{f}+i)}$ $D_{bdf\bar{i}}$ une nouvelle application de $x=0$ donne :

$$A_{f\bar{i}(\bar{b}+\bar{d})} + bdfi\bar{j}\bar{c} + bdf\bar{i}c$$

$$B_{\bar{b}f} + \bar{b}i\bar{j} + bdc\bar{f} + bdcij + bcd\bar{f}\bar{i}$$

$$C_{b\bar{d}(\bar{f}+i\bar{g})} + bcd(\bar{f}+ij)$$

$$D_{bfg\bar{d}} + bcdi\bar{j}$$

CONCLUSION

L'extension de la simulation déductive de pannes simples, au cas des pannes multiples, a montré comment un test des pannes simples pouvait être utilisé pour obtenir un test des pannes multiples.

BIBLIOGRAPHIE

- D.C. BOSSEN, S.J. HONG : "Cause-effect analysis for multiple fault detection in combinational networks" IEEE Trans. on Comp. Nov 71.
- H.Y. CHANG : "Comparison of Parallel and Deductive Fault Simulation Method" IEEE Trans. Comp. Nov 74.
- D.B. ARMSTRONG : "A deductive method for simulating faults in logic circuits" IEEE Trans. Comp. Mai 71.
- F.J. DIAS : "Fault Masking in Combinational Logic Circuits" IEEE Trans. Comp. Mai 75.

C H A P I T R E V

TEST DES GRANDS ENSEMBLES

Exemple du Centre E 10.

CHAPITRE V . TEST DES GRANDS ENSEMBLE

I - PRESENTATION DU CENTRE E.10

- I - 1. Le centre E 10.
- I - 2. Le multienregistreur.

II - DECOUPAGE, ORDONNANCEMENT

- II - 1. Méthodes de localisation.
- II - 2. Ordonnancement.
- II - 3. Decoupage.
- II - 4. Programme de test.

III - LOCALISATION, POLITIQUE DE REPARATION

- III -1. Localisation statistique.
- III -2. Politique de réparation.

IV - TEST SOUS CONTRAINTS

- IV - 1. Les circuits combinatoires.
- IV - 2. Les circuits séquentiels.

INTRODUCTION

Si la question du test des circuits de petite taille est assez bien connue, tant du point de vue théorique que du point de vue des outils pratiques, il n'en est pas du tout de même pour ce qui est du test de grands systèmes.

En effet, sur de tels problèmes il existe soit des études théoriques qui paraissent assez loin des problèmes pratiques, soit des études de cas faisant très peu ressortir de méthodologie générale.

Cette étude dégage, à partir du traitement du centre E 10 des méthodologies sur les points suivants :

- Découpage, ordonnancement. Cette phase est spécifique du test de grands systèmes en effet vu la complexité de ceux-ci, il est impossible d'utiliser les méthodes de tests valables pour les petits éléments.
- Localisation, politique de réparation qui montre un aspect de la maintenance à savoir les questions de réparation.
- Le test sous contrainte. C'est une des questions clefs du test des grands ensembles : comment, si on a un test d'un sous-ensemble (test local), peut-on générer un test de celui-ci dans le contexte général (test global).

Ce travail a été fait en collaboration avec Monsieur J.L. RAINARD dans le cadre d'un contrat avec le CNET.

I - PRESENTATION RAPIDE DU MULTIENREGISTREUR (centre E 10)

I 1. Le système E 10

Le système E 10 est un réseau de communication téléphonique. Les opérations assurées par un tel système sont de deux types :

- les opérations propres à la mise en communication des abonnés : communication automatique -
- les opérations d'exploitation : gestion, maintenance.

Les opérations du premier type sont à effectuer en temps réel, et demandent une très grande disponibilité. Elles sont confiées à un matériel très spécialisé et décentralisé pour augmenter la disponibilité, l'instant d'exécution des tâches du deuxième type peut au contraire être choisi librement, et la disponibilité n'est pas aussi importante. Par contre, ces tâches sont de natures très diverses et sont susceptibles d'évoluer et de se modifier. Elles sont donc confiées à un calculateur universel et peuvent être très centralisées.

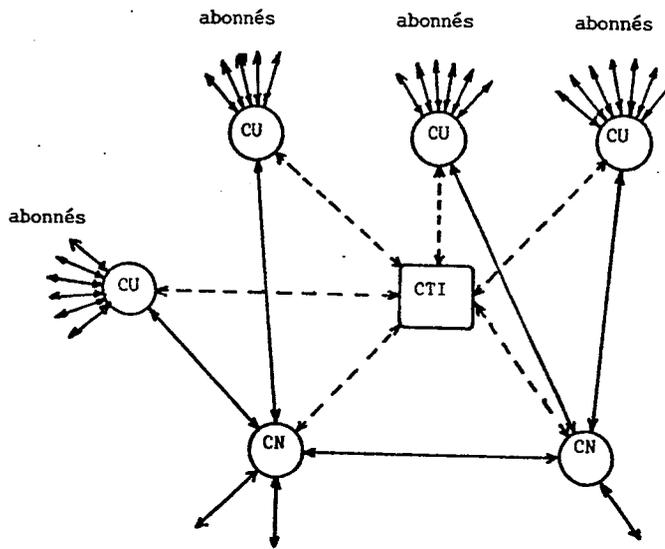
Signalons aussi que le système E 10 utilise le principe de modulation par codage d'impulsion (MIC) : d'une part, le temps est découpé en tranches, chaque tranche étant réservée à une communication. D'autre part, si signal analogique correspondant à la parole est digitalisé pour faciliter les transmissions à longue distance.

D'où l'organisation (simplifiée) d'un centre E 10.

Il comprend :

- des centres urbains (CU) plus particulièrement chargés de centraliser les lignes d'abonnés et d'effectuer l'interface entre les systèmes de transmission BF et MIC.
- des centres nodaux (CN) assurant le transit centre à centre.

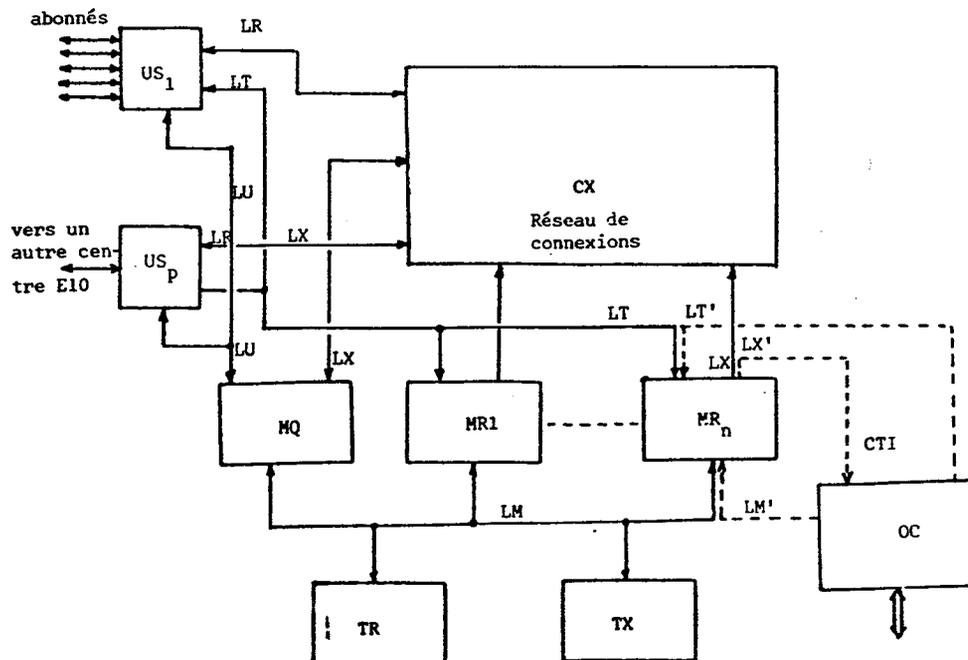
- un CTI (centre de traitement de l'information) relié à chaque centre par des voies de transmission de données digitales. Il est chargé des opérations du deuxième type.



SYSTEME E 10

Le centre E 10

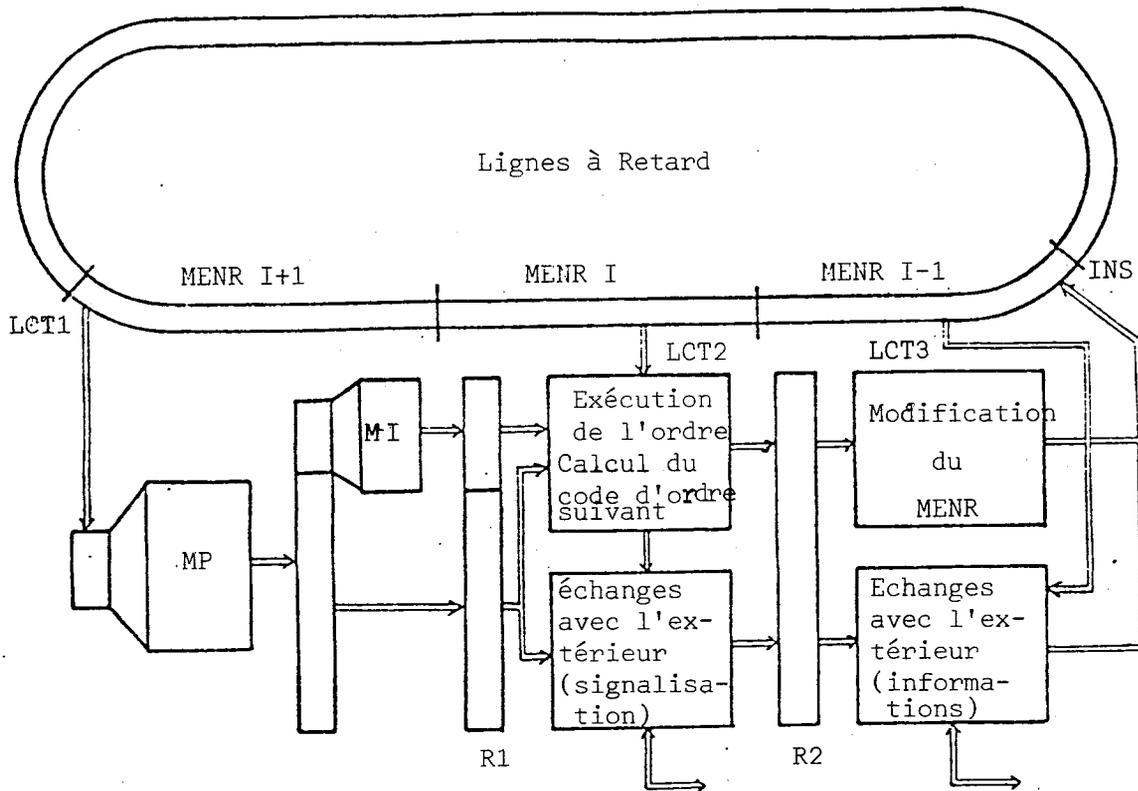
Du point de vue matériel, il y a peu de différence entre les centres urbains et les centres nodaux. Le schéma type d'un centre E 10 est le suivant :



Centre E 10

Il comprend :

- Des unités de sélection (US)
Elles sont de deux types : les équivalents de modulation (interface entre lignes abonnés et système MIC) et les équipements de synchronisation (destinés à resynchroniser les informations provenant de voies MIC, car chaque centre possède son horloge propre).
- Le réseau de connexions (CX)
Il met en relation, dans le système MIC, les lignes provenant des unités de sélection.
- Le marqueur (MQ)
Il assure la liaison entre les unités de sélection, le réseau de connexion, les multienregistreurs et l'organe de contrôle.
- Le traducteur (TR)
C'est une mémoire contenant les informations concernant les abonnés.
- Le taxeur (TX)
Il calcule la taxation associée à chaque communication.
- Les multienregistreurs (MR)
Il sont chargés de séquencer les différentes phases nécessaires à l'établissement et à la rupture des communications.
- L'organe de contrôle (OC)
Il réalise les opérations d'échange entre le CTI et le centre E 10. En particulier, il a une fonction d'aide à la télémaintenance. Notons que toutes les liaisons entre les différents organes sont doublées par des liaisons de test (figurant, pour MR seulement, en pointillé) aboutissant à l'organe de contrôle, et rendant ce dernier capable de simuler n'importe quel organe lors des opérations de télémaintenance à partir du CTI.

I -II. Le multienregistreurStructure :

LE MULTIENREGISTREUR

Le multienregistreur est organisé autour d'une mémoire circulante composée de deux registres à décalage circulaire, LAR 1 et LAR 2. Chaque registre est découpé en 66 mots de 160 bits chacun. L'ensemble composé par deux mots homologues (l'un en LAR 1, l'autre en LAR 2 et occupant les mêmes positions binaires dans les deux registres) est appelé mot enregistreur (MENR).

Le rôle du MR est de séquencer les opérations nécessitées par l'établissement ou la rupture d'une communication. A cet effet, lorsque une communication doit être établie ou coupée (ceci est détecté et signalé par les US), un mot enregistreur lui est affecté. Il contient, à lui seul, toutes les

informations concernant l'état de la communication.

Chaque mot enregistreur est composé :

- d'un numéro de phase (\emptyset), correspondant à l'adresse d'une microinstruction
- d'information concernant la communication.

Le multienregistreur a une structure de type pipe-line : il est découpé en 3 parties :

- La première est chargée de la lecture du numéro de phase et de la lecture, dans des mémoires mortes MP et MI, de la microinstruction correspondante.
- La deuxième concerne l'exécution de l'instruction ainsi obtenue.

Cette exécution comporte :

- La prévision de la microinstruction qui devra être exécutée au prochain passage du MENR (c'est-à-dire après un cycle complet des registres à décalage). Elle peut dépendre du résultat de l'opération en cours. Cette prévision est donnée sous forme d'un correctif $\Delta\emptyset$, qui sera retranché ou ajouté à la zone " numéro de phase " du MENR.
- Eventuellement des échanges de signaux avec l'extérieur, en vue d'un échange de données.
- La troisième partie est chargée des modifications du MENR, ainsi que des échanges de données avec l'extérieur. En particulier c'est à ce stade que sera effectuée la correction du numéro de phase à l'aide du $\Delta\emptyset$ précédemment calculé.

Deux registres (R1 et R2) réalisent l'isolement des trois parties qui peuvent alors fonctionner en pipe-line. Le MENR $i+1$, arrivé au premier point de lecture LCT1, communique son numéro

de phase à la première partie du MR, pendant que le MENR i (point de lecture LCT2) en est à la phase d'exécution, et que le MENR i - 1 subit des modélisations (au point d'écriture INS), soit sur son code d'ordre, soit sur sa zone donnée. Les éventuelles données à transmettre à l'extérieur sont lues au point LCT3.

Le MENR associé à une communication subira donc un traitement à chaque cycle des registres décalage, c'est-à-dire tous les 8,25 ms. Lorsque le but recherché est atteint (mise en communication ou libération d'abonnés), le mot enregistreur est remis à zéro, et peut être utilisé pour une nouvelle communication.

En conclusion, le multienregistreur se caractérise par :

- Son système de registres à décalage utilisés comme mémoire vive.
- Son organisation pipe-line.
- L'importance du traitement série de l'information : en effet, l'écriture et la lecture en mémoire, ainsi que les opérations arithmétiques (addition, soustraction) et les échanges avec l'extérieur se font par transmission série des bits d'information. Il en résulte un système d'horloge assez complexe, qui jouera un rôle important dans le test.

Du point de vue testabilité, il faut noter l'existence d'un certain nombre d'accès prévus spécialement pour faciliter la maintenance.

- Un système d'adressage et de lecture indépendant pour la mémoire programme (mémoire morte contenant les microinstructions).
- Un système de lecture des registres à décalage, relié au point de lecture LTC2.
- Un système permettant de remplacer chaque connexion

entre le multienregistreur et les autres éléments du centre, par une connexion similaire de test. Elles sont reliées au CTI par l'intermédiaire de l'OC, dans le but de permettre une simulation de l'environnement de la machine à tester. (Cette politique est d'ailleurs utilisée pour tout le centre E 10). Ces liaisons de test sont validées lorsque le multienregistreur est dans un état (état TS) destiné à la maintenance. Il peut alors être considéré comme totalement isolé du reste du centre, et l'on peut avoir accès librement à ses entrées et à ses sorties.

STRATEGIE DE TEST

I - Type de maintenance

Parmi les trois types de maintenance possibles (Télémaintenance, Automaintenance et Autotest), il faut choisir celui qui s'adapte le mieux au cas du multienregistreur E 10.

Mis à part quelques systèmes de surveillance (bits d'imparité) placés à des points cruciaux tels que les échanges avec l'extérieur, rien n'a été prévu pour assurer un autotest complet. De même, l'organisation du multienregistreur ne permet pas de mettre en oeuvre un système d'automaintenance permettant une localisation satisfaisante des pannes.

Par contre, le multienregistreur et son environnement immédiat ont été conçus en vue de la mise en oeuvre d'un système de télémaintenance entièrement automatisé.

Les constructeurs ont, en effet, prévu des points d'accès supplémentaires permettant d'isoler le multienregistreur de son contexte réel, et de simuler ce dernier à l'aide du CTI. Il y a cependant quelques problèmes :

- Il n'a pas été prévu d'accès en entrée sur R 2 : les microprogrammes de test devront donc être contenus dans

MP. Il serait très délicat (et très long) d'utiliser les programmes banaux de MP. Il faudra donc prévoir dans MP une partie réservée uniquement à des programmes de test, et par conséquent minimiser la taille de ces données.

- Les échanges entre le CTI et le multienregistreur coûtent cher en temps, il faudra donc minimiser leur nombre.

II - Localisation des pannes

II - 1. Généralités

Unité de remplacement

C'est par définition, le plus petit élément remplaçable dans le cadre de la maintenance utilisée. Il est bien évident que la localisation de la panne est équivalente, pour cette maintenance, à la localisation de l'unité de remplacement défectueuse.

II - DECOUPAGE, ORDONNANCEMENT

II - 1. Méthode de localisation

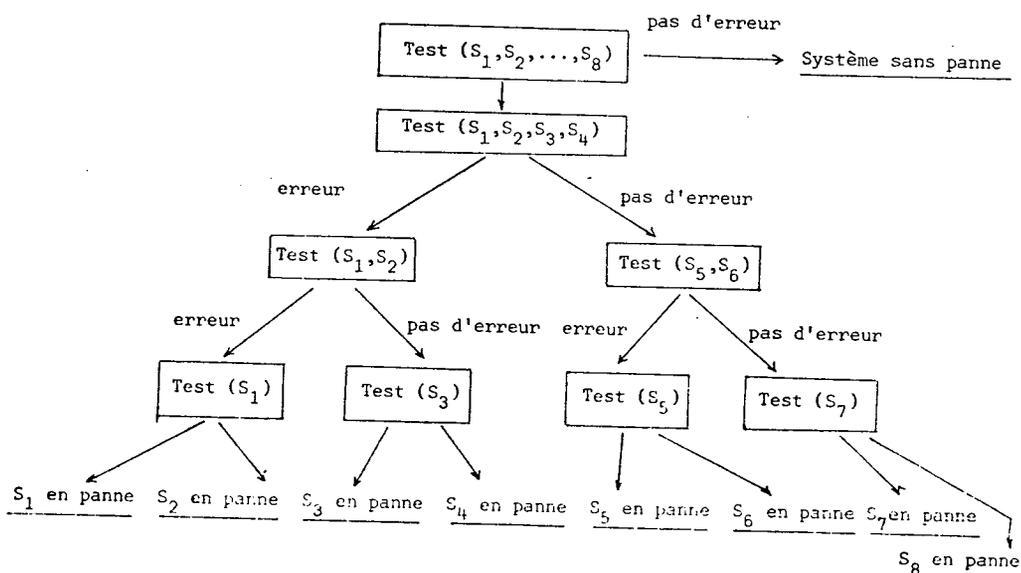
Il existe deux méthodes principales de stratégie de localisation pour les grands ensembles : la méthode de boule de neige (start small, bootstrapping) et la méthode de dichotomie (start big).

A - Méthode de dichotomie

Elle a été utilisée par le système BAMB [3]. Cette méthode consiste à tester les pannes du système **entier**. Si une panne est détectée, le système est divisé en 2 parties A et B qui sont isolées l'une de l'autre. La partie A est testée, s'il n'y a pas de panne détectée, alors la panne du système est

localisée en B et on opère de même en divisant B en deux sous systèmes Si quand on teste A la panne est détectée, alors la panne est localisée en A et pour affiner la localisation, on divise A en 2 sous systèmes.

- Soit un système S divisé en 8 sous systèmes S_1, S_2, \dots, S_8 , la méthode de test par dichotomie peut se schématiser par l'arbre suivant :



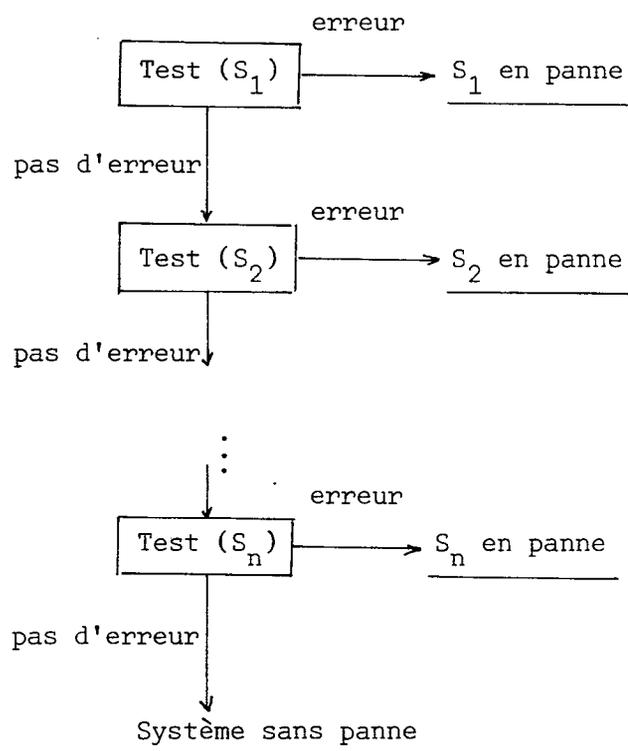
Quand on teste un sous-ensemble, il faut l'isoler du reste du système. Si N est le nombre d'unités de localisation de système, la localisation se fait en un nombre d'étapes qui est de l'ordre de $\log_2(N)$. Cette méthode est utilisable pour localiser les pannes simples en effet, il est impensable de tester les pannes multiples de niveau de grands ensembles. Remarquons aussi que cette méthode nécessite que l'on sache générer de façon efficace le test de très grands ensembles.

B - Méthode boule de neige.

Le système S est découpé en unités de localisation : S_1, S_2, \dots, S_n . Les unités de localisation sont

testées l'une à la suite de l'autre. Ceci nécessite qu'à l'étape j (test de S_j) toutes les unités S_k ($k < j$) soient isolées (logiquement ou physiquement) de S_j et que l'on puisse atteindre (commander et observer) S_j à travers les modules S_ℓ ($\ell < j$)

La méthode de localisation peut se schématiser :



On voit qu'un système ayant N unités de localisation demande en moyenne de l'ordre de $N/2$ étapes de test. Cependant il faut remarquer que par rapport au test par dichotomie, les étapes de test sont ici plus courtes. En particulier si la longueur du test de (S_1, \dots, S_i) est de l'ordre de $\sum_{h=1}^i [\text{test}(S_h)]$ alors la méthode boule de neige est plus rapide.

De plus cette méthode ne nécessite pas de trouver des tests de trop gros ensembles et, si au niveau des tests

des unités de localisation on a un test de pannes multiples, alors le test global est aussi un test de pannes multiples. C'est cette méthode qui a été utilisée pour le multienregistreur.

II - 2. Ordonnancement

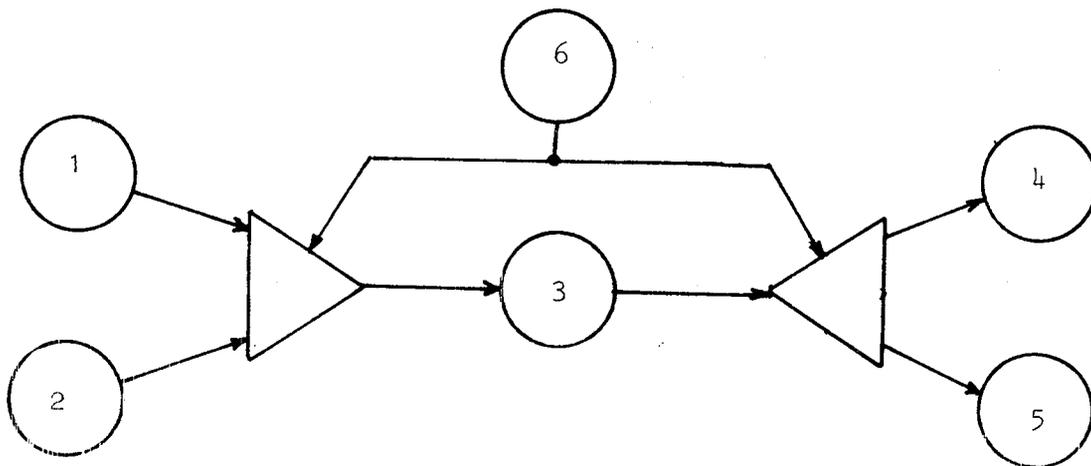
Pour trouver un ordonnancement des tests en bloc on définit deux relations : la commandabilité et l'observabilité soit : $B = \{B_1, B_2, \dots, B_n\}$ et $\mathcal{B} = 2^B$ l'ensemble des parties de B , ces relations sont définies entre \mathcal{B} et B .

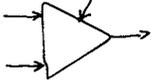
Définition (B_i, B_j, \dots, B_k) commande B_e si on peut manifester les erreurs de B_e en activant l'ensemble (B_i, B_j, \dots, B_k) .

Définition (B_i, B_j, \dots, B_k) observe B_l si on peut propager à travers au moins un (B_i, B_j, \dots, B_k) les variations de sortie de B_l correspondant à des pannes de B_l .

On peut définir la relation d'accessibilité comme le produit des deux relations : commandabilité et observabilité.

Ceci sous entend qu'il n'y a pas d'interférence entre les contraintes de manifestation de la panne et celles de propagation. Cependant, il peut très bien se présenter de telles interférences comme le montre cet exemple :

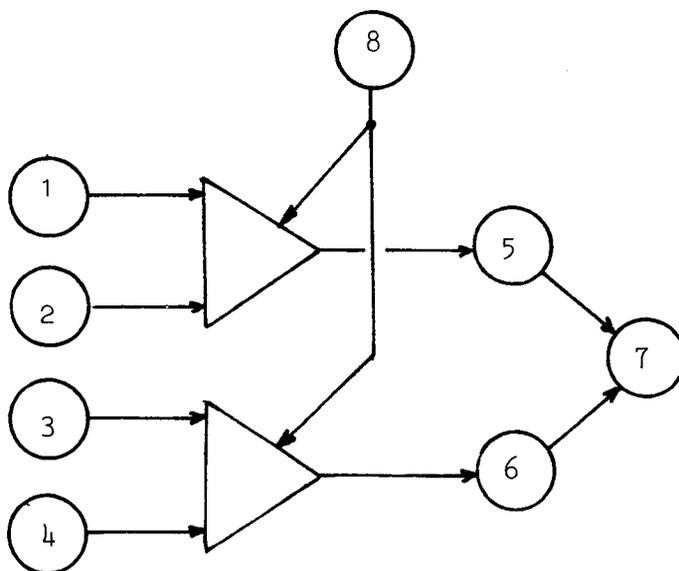


Le module  est un multiplexeur et 

est un démultiplexeur. Le système décrit ci-dessus transfère les données de 1 (respectivement 2) vers 4 (respectivement 5), après le traitement par 3. On a : (1,6) et 2,6) commandent 3.
(4,6) et 5,6) observent 3.

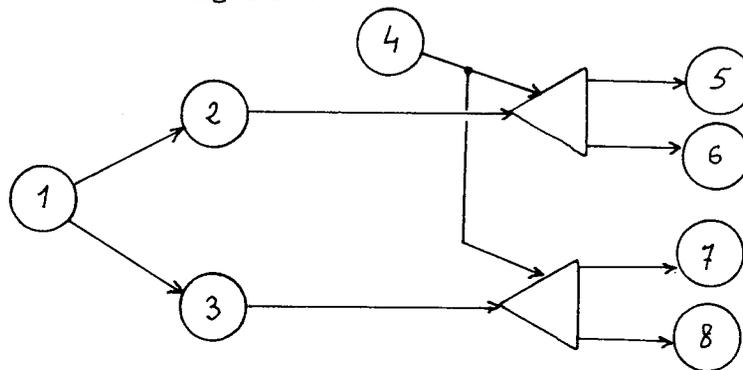
Si l'on fait le produit des relations on obtient :
(1,4,6), (1,5,6), (2,4,6), et (2,5,6) accèdent à 3. Or du fait des contraintes créées par 6 on a seulement (1,4,6) et (2,5,6) accèdent à 3.

Ainsi il faut utiliser la relation d'accessibilité avec précaution, de même il faut faire attention à l'usage des relations de commande et d'observation. En effet ces relations ne sont pas " transitives " c'est-à-dire que si B' commande B'' et B'' commande B_k (B', B'' dans B et B_k dans B) alors on n'a pas forcément que B' commande B_k , comme le montre l'exemple suivant :



Dans cet exemple (5,6) commande 7, (1,8) et (2,8) commandent 5 et (3,8) ainsi que (4,8) commandent 6. S'il y avait transitivité on aurait : (1,3,8), (2,3,8), (1,4,8) et (2,4,8) commandent 7. Mais du fait des contraintes qu'impose 8 les multiplexeurs on a seulement : (1,3,8) et (2,4,8) qui commandent 7;

On peut avoir une situation directe pour l'observation comme le montre la figure :



Ces restrictions étant signalées il faut noter que dans les exemples pratiques de telles configurations se présentent peu souvent. On peut donc utiliser les relations d'observabilité et de commandabilité pour en déduire un ordonnancement possible, les points d'accès supplémentaires nécessaires et de faire suivre cette phase d'une phase de validation afin de vérifier si les problèmes mentionnés ci-dessus apparaissent ou non.

Nous verrons que la localisation à 100 % n'est pas possible. Ceci amène à définir des relations plus souples. Supposons que (B_i, B_j) commande B_k , soit P l'ensemble des pannes de B_k et $P_{B_i, C}$ l'ensemble des pannes de B_k que l'on peut tester en activant B_i et en affectant aux sorties de B_j la valeur C - Ceci peut définir le coefficient de commande associé à B_j et $C : \left| \frac{P_{B_i, C}}{P} \right|$, ce qui permet de définir le coef-

ficient de commande associé à $B_i : \text{com}(B_i) = \max_C \left| \frac{P_{B_i, C}}{P} \right|$

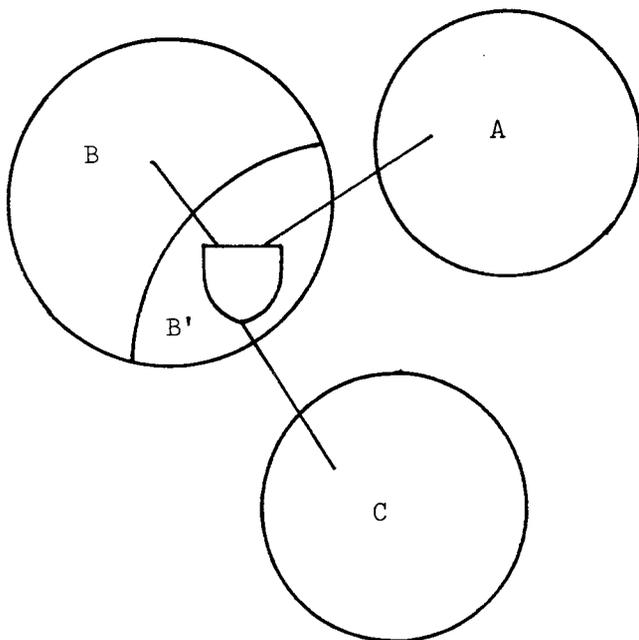
Si (B, B_m) observe B_k on note P_B l'ensemble des pannes de B_k que l'on peut tester à travers B_ℓ , le coefficient d'observation est alors $OBS(B_\ell) = \frac{|P_{Be}|}{|P|}$.

Ayant défini les coefficients de commande α_1 et d'observation α_2 , on peut considérer que l'on peut tester B_k de façon satisfaisante avec B_i et B_ℓ si $com(B_i) \geq \alpha_1$ et $Obs(B_\ell) \geq \alpha_2$.

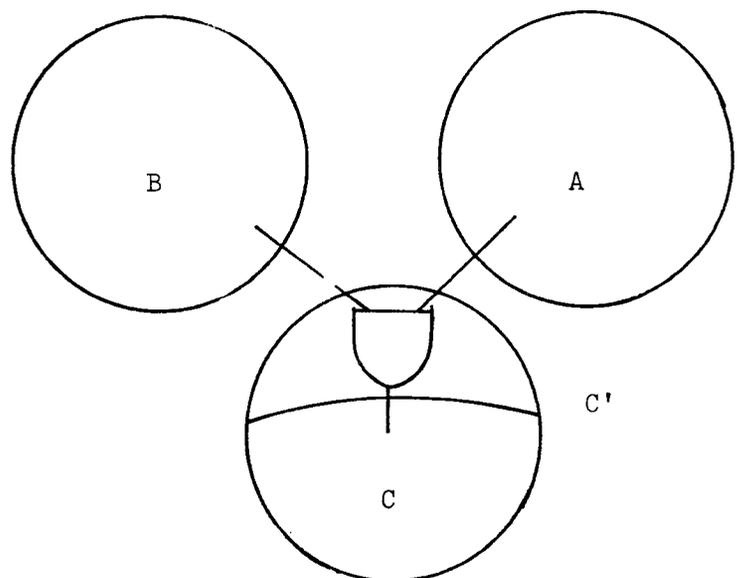
III -3 . Découpage

On n'a pas abordé ici l'étude du découpage d'un système en blocs. C'est, en effet, un problème de décomposition dont l'objet sort de cette étude. Le découpage qui a été utilisé a été un découpage fonctionnel lié à l'implantation sur carte. Cependant l'étude du test a mis en évidence des défauts de ce découpage comme le montrent les exemples suivants. Ainsi au découpage initial a été substitué un découpage modifié.

EXEMPLE : Quand on a le découpage D on lui substitue le découpage D'.



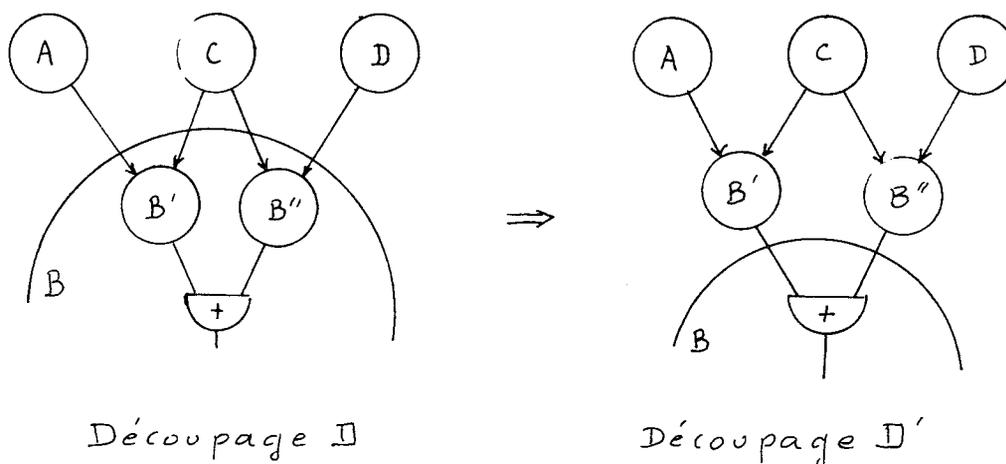
Découpage D



Découpage D'

Les avantages du nouveau découpage sont évidents : le bloc B n'a pas besoin pour être testé du bloc A (alors que BB' nécessitait le test de A) et C peut maintenant être testé à partir de A ou de B (au lieu de n'être testable que par B).

EXEMPLE 2 :



On a ici mis en évidence le fait que le bloc B est commandable soit par B' (AC) soit par B'' (CD). Les transformations ci-dessus sont aussi utilisables avec les points ET.

Il est à noter que ces transformations aboutissent à un découpage plus fonctionnel.

II - 4. Programme de test pour le multienregistreur

Une fois que l'ordonnancement en blocs a été fait, la recherche des séquences de test pour chaque bloc se fait par une méthode classique de sensibilisation de chemin. Le test est fait par télémaintenance et se divise en 3 étapes.

- 1 - test de la mémoire programme et des accès -
- 2 - test d'une ligne à retard, des horloges, modifi-

cation du numéro de phase, cette étape se fait en pas à pas car le séquençement n'a pas été testé.

- 3 - test de reste des blocs qui se fait en initialisant un numéro de phase, le multienregistreur se chargeant lui-même du séquençement.

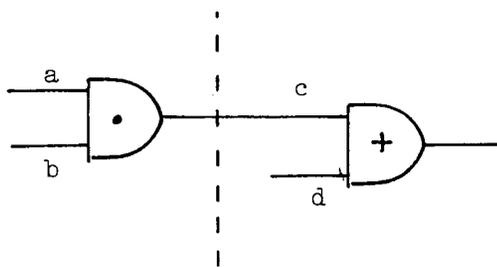
Le test demande 500 échanges entre le CTI et le multienregistreur, si on ne teste pas la mémoire programme, et 2900 échanges si on teste la mémoire programme, ce qui correspond à des temps de 50 et 290 secondes.

III - LOCALISATION, POLITIQUE DE REPARATION

Pour mesurer la qualité d'une maintenance, il faut faire intervenir la qualité de la localisation des pannes et la politique de réparation (remplacement).

III - 1. Localisation statistique

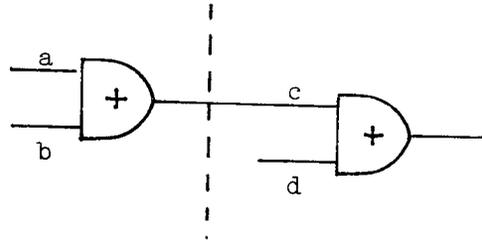
Quelle que soit la méthode de localisation, pratiquement la localisation parfaite est impossible du fait, entre autres, des pannes indiscernables. Considérons l'exemple décrit par la figure ci-dessous :



Les pannes a collée à 0, b collée à 0 et c collée à 0 sont indiscernables. Donc en présence d'une de ces 3 pannes on ne peut pas en déduire laquelle des unités i ou j est en panne. Cette situation est encore plus sensible quand on a une cascade de ET (ou de OU) dont une partie se situe dans

une unité et une autre partie dans une autre.

EXEMPLE :



Les collages 1 de a, b, c, d sont indiscernables.

Il faut remarquer que la proposition de pannes indiscernables et se situant sur deux unités différentes est faible, en effet elles concernent les pannes sur les interconnexions ou très voisines.

En fait la cause principale de l'impossibilité d'une localisation à 100 % est due à la non superposition entre unité de test et unité de remplacement. Ceci est particulièrement sensible quand la question du test est abordée après l'implantation. A ce moment un même bloc se retrouve réparti sur plusieurs unités de remplacement. Il y a cependant une différence entre un bloc également réparti sur ces unités de remplacement, ou un bloc implémenté à 90 % sur une unité.

Définition Soit un bloc de test B formé de ces sous blocs B_1, B_2, \dots, B_n , la localisation statistique d'une panne de B à un sous bloc B_i est la probabilité qu'il y ait une panne dans B_i sachant qu'il y a une panne dans B ○

III - 2. Politique de réparation

La politique de réparation définit les règles de remplacement des unités de remplacement. Soit un système de

localisation accusant le bloc B d'être en panne, deux politiques de réparation peuvent être désignées : l'une consiste à changer une des unités de remplacement (sur lesquelles est implanté B), au hasard, puis de vérifier le fonctionnement du système, l'autre à changer tous les éléments remplaçables qui implémentent B. La première politique coûtera du temps mais risque de faire changer moins d'unité de remplacement que la deuxième politique qui aura cependant l'avantage de la rapidité.

A - Première politique de réparation

Dans un premier temps adoptons la politique de réparation suivante : à la réception d'un signal d'erreur lors du test du bloc B_h , on change la carte possédant la plus forte probabilité de panne. Puis l'on repasse le programme de test. S'il signale toujours une panne dans le même bloc, on change la carte dont la probabilité de panne vient en second, et ainsi de suite.

Une intervention consiste donc à remplacer une carte. Associons le coût 1 à chaque intervention. Le coût de maintenance M_h de B_h sera alors égal au nombre moyen d'interventions nécessaire à la réparation de B_h .

$$\text{Coût } (M_h) = N_h = \sum_{k=1}^{n_h} k \cdot P_{hk}$$

ou n_h représente le nombre de cartes sur lesquelles est implanté B_h et P_{hk} représente la localisation statistique de la k ième carte sur laquelle est implantée B_h .

Soit Q_h la localisation statistique du bloc B_h on a

$$\text{Coût } (M) = \sum_{h=1}^b Q_h \text{ coût } (M_h) \quad \text{où } b \text{ est le nombre de blocs du système}$$

ce qui donne :

$$N = \text{Coût } (M) = \sum_{h=1}^b Q_h N_h = \sum_{h=1}^b (Q_h (N_h - 1)) + 1$$

Le premier tableau ne représente localisation statistique de la carte (unité de remplacement) sur laquelle est implanté le bloc de test. Cette probabilité a été calculée en fonction des probabilités de pannes des différents circuits.

SSI	4,6	$\cdot 10^{-8}$	pannes/heure
MSI	18	$\cdot 10^{-8}$	pannes/heure
MOS	27	$\cdot 10^{-8}$	pannes/heure

Le deuxième tableau donne bloc par bloc, les valeurs de Q_n , $Q_n (N_n - 1)$, les probabilités de localiser un bloc à une carte, deux, ..., ainsi que le nombre de cartes nécessaires à remplacer pour une localisation à 90 %, ou 100 %.

Blocs	1	2	3	4A	4B	5	5	7	8A	8B	9	10	11	12A	12B	12C
localisation à 1 carte	0,926	0,954	0,925	0,975	0,983	0,945	0,921	0,995	0,975	0,983	0,935	0,976	0,992	0,951	0,997	0,980
localisation à 2 cartes	0,984	0,975	0,972	0,953	0,926	0,669	0,950	0,930	0,995	0,995	0,655	0,556	0,938	0,920	1,000	1,000
localisation à 3 cartes	1,000	0,977	0,957	0,992	0,956	0,945	0,978	0,952	0,999	0,999	0,966	0,935	0,999	0,956	/	/
localisation à 4 cartes	/	1,000	0,991	0,936	0,993	0,997	0,955	0,972	1,000	1,000	1,000	0,997	1,000	0,998	/	/
localisation à 5 cartes	/	/	0,994	1,000	1,000	0,999	0,990	0,987	/	/	/	0,995	/	0,995	/	/
nombre de cartes à localiser à 90%	3	4	9	5	5	6	10	8	4	4	4	6	4	7	2	2
nombre de cartes à localiser à 100%	1	3	1	2	2	3	1	2	1	1	3	1	1	2	1	1
N_i	1,090	1,691	1,125	1,184	1,232	2,045	1,193	1,269	1,031	1,023	2,041	1,047	1,011	1,277	1,003	1,020
Q_i	0,094	0,019	0,112	0,041	0,022	0,134	0,112	0,086	0,060	0,060	0,028	0,074	0,072	0,055	0,025	0,004
$Q_i (N_i - 1)$	0,0054	0,0134	0,0140	0,0076	0,0051	0,1400	0,0217	0,0231	0,0018	0,0014	0,0252	0,0235	0,0208	0,0152	0,00007	0,00007

On obtient alors comme coût de réparation :

$$N = 1,29$$

B - Deuxième politique de réparation

Au lieu de changer une carte à chaque intervention on change, lors de la première intervention, autant de cartes qu'il faut pour que la localisation atteigne un certain seuil par exemple 90 %. Si par la suite, la panne est toujours présente, on change une carte à la fois.

Par le même calcul que précédemment, on obtient maintenant un coût de réparation :

$$N = 1,09$$

C - Influence de modifications de réalisation

L'examen des tableaux 3 et 4 fait apparaître l'utilité

de certaines modifications.

Tableau 1 : Tout d'abord, il apparait que les blocs 3 et 6 ont une localisation à 9 et 11 cartes. C'est ennuyeux, car dans le cas (très improbable il est vrai) d'une telle panne, la recherche va prendre beaucoup de temps. Le bloc 3 contient la LAR1, le bloc 6 la LAR2. Ce problème avait déjà été signalé auparavant : certaines pannes en entrée des lignes à retard sont indiscernables de toutes façon. Il avait été suggéré alors de réaliser l'isolation des lignes à retard à l'aide d'un apport de matériel permettant de bloquer certaines entrées depuis l'extérieur.

Ensuite, on peut remarquer l'influence de la carte MR5 : elle apparait dans presque tous les blocs. Cela est dû à la mémoire d'instruction MI, contenant les microinstructions. On ne peut pratiquement rien faire sans utiliser MI. Il serait donc très profitable de pouvoir tester MI à part, donc de rajouter un accès vers l'extérieur permettant de lire le contenu du registre RI.

Tableau 2 : On peut considérer $Q_i (N_i - 1)$ comme l'apport du bloc B_i aux 0,29 de "dépassement" de N. Or, le bloc 5, à lui tout seul, provoque la moitié de ce dépassement (0,14). Il serait donc tout particulièrement intéressant de réduire ce terme en rajoutant des accès sur ce bloc.

Avec les modifications visant à réduire l'incertitude de localisation, on obtient un coût $N = 1,22$ (au lieu de 1,25) pour la première politique et $N = 1,046$ (au lieu de 1,09) pour la deuxième politique.

Remarquons que le gain de localisation est obtenu (quelques %) par des modifications mineures, faites après la conception. Ceci prouve que l'on peut s'attendre à un gain appréciable si

on tient compte des problèmes de test lors de la conception.

Notons que la question de la réparation est un aspect de la maintenance qui comprend, elle, la gestion de stocks etc...

IV - TEST SOUS CONTRAINTS

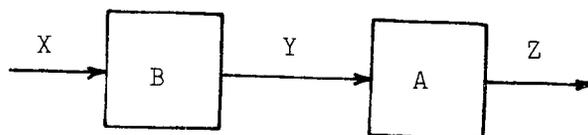
Les algorithmes de génération automatique de séquence de test ne sont efficaces que pour des systèmes de complexité faible ou moyenne. Les méthodes proposées ici rendent le test automatique pour de grands systèmes.

Un système S est d'abord partitionné en sous-systèmes fonctionnels $\{ S_i \}$ qui, pris isolément, sont testables automatiquement. Cependant si on génère les tests pour un S_i , pris isolément, on n'est pas sûr d'en déduire un test qui puisse être appliqué au système S , ceci à cause des contraintes des autres S_j . Pour un sous-système S_i il faut tenir compte des contraintes créés par S d'une part sur les entrées de S_i : ce sont les contraintes de commandabilité, d'autre part sur l'observation des variations des sorties : ce sont les contraintes d'observabilité.

Le problème devient alors de tenir compte du contexte de S_i quand on le teste, afin que les tests trouvés pour S_i puissent être réellement générés et observés dans le système S .

IV - 1. Circuits combinatoires

Soient deux circuits combinatoires A et B en cascade suivant la figure ci-dessous :



Si l'on veut tester les 2 circuits A et B séparément, il faut pour tester A tenir compte des contraintes de commandabilité créés par B sur Y et pour tester B il faut tenir compte des contraintes d'observabilité créés par A.

IV - 1.1. Contraints de commandabilité

Soient Y les entrées du circuit A à tester et B le circuit créant les contraintes, B ayant comme entrées X.

Définition : La fonction des contraintes de commande est définie par $\theta(Y) = 1 \Leftrightarrow \exists X: B(X) = Y \quad \text{O}$

Cette fonction décrit les vecteurs d'entrée de A qui peuvent être générés par B. Ceci implique que le test de A soit un ensemble de vecteurs du test $\{t_i\}$ tels que $\theta(t_i) = 1$.

Pour faire intervenir les contraintes lors du test on peut envisager deux méthodes :

Méthode 1. On recherche d'abord les tests de A et on vérifie ensuite s'ils vérifient ou non les contraintes. Cette méthode est utilisable si le circuit B induit peu de contraintes, à ce moment l'obtention d'un test de A a une probabilité faible d'être rejeté; par contre, si le circuit B induit de forts contraintes ($|\theta^{-1}(0)|$ petit) alors cette méthode devient très

$$\frac{|\theta^{-1}(1)|}{|\theta^{-1}(0)|}$$

lourde car la recherche de test de A donnera très souvent des vecteurs inutilisables.

Méthode 2 - Au lieu de faire intervenir la fonction des contraintes θ après la recherche d'un test, on la fait intervenir pendant la recherche des tests.

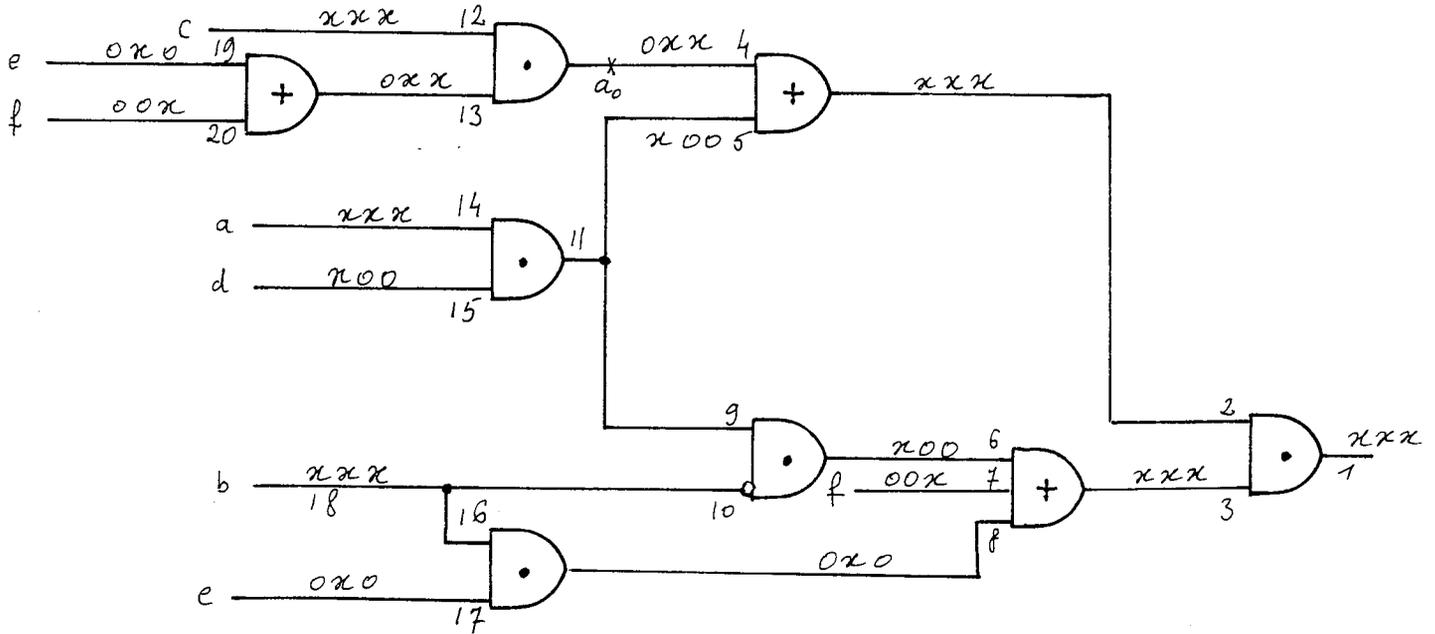
La fonction θ peut être représentée par un ensemble de monômes. Avant de commencer l'algorithme de recherche des tests A on fait une simulation de ces monômes dans A. Pour ce faire les monômes sont décrits par des vecteurs sur chaque composante vaut 1, 0, x suivant que dans le monôme correspondant le variable apparaît sous forme normale complétée, ou n'apparaît pas.

On obtient alors pour chaque ligne du circuit un vecteur de simulation qui correspond à la propagation des contraintes sur cette ligne.

EXEMPLE de simulation : Soit le circuit à 6 entrées a,b,c,d,e,f décrit par la figure ci-dessous. Les contraintes sont tels qu'une seule des variables d, e, f vaut 1 simultanément. Les contraintes peuvent donc être représentés par 3 monômes $\theta = \bar{d} \bar{e} + \bar{d} \bar{f} + \bar{e} \bar{f}$ que l'on peut aussi représenter par les 3 vecteurs :

	V1	V2	V3
a	x	x	x
b	x	x	x
c	x	x	x
d	x	0	0
e	0	x	0
f	0	0	x

La simulation de ces vecteurs sur le circuit donne :



Cette simulation est faite initialement et le résultat est utilisé pour la recherche des vecteurs de test. Notons que s'il n'y a aucune contrainte la fonction vaut 1 tout le temps et la simulation donne x sur chaque ligne.

Lors de la recherche des vecteurs de tests les valeurs des vecteurs de contrainte sur chaque ligne permettent de guider le choix de propagation possible des pannes, ainsi que de guider la consistance, simplifiant d'autant ces deux opérations.

EXEMPLE de recherche de test : Cherchons les tests du collage à 0 de la connexion 4 de l'exemple précédent : L'examen des vecteurs de contraintes sur la connexion 4 montre qu'on ne peut manifester la panne que pour les 2^e et 3^e vecteurs. Les règles de consistance habituelles remplacent si c'est nécessaire des valeurs x par 0 ou 1 et finalement on obtient comme ensemble de test :

$$T_{4_0} = b c \bar{d} e \bar{f} + c \bar{d} \bar{e} f$$

Remarque : On peut représenter la fonction des contraintes de différentes façons. Si on prend les monômes canoniques le nom-

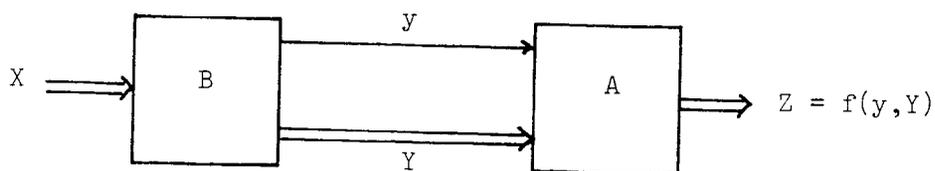
bre de monômes, donc de composants des vecteurs risque d'être très grand et la simulation très lourde; par contre dans la recherche de vecteurs de test le travail de consistance est nul. Un bon compromis semble être de ne considérer que les représentations en monômes disjoints et de choisir parmi celles-ci une qui possède un nombre minimum de monômes.

Pour résumer, l'intérêt de la méthode 2 sur la méthode 1 vient du fait que :

- a) on ne fait intervenir les contraintes qu'une seule fois à l'initialisation de la recherche des tests.
- b) la propagation des contraintes à l'intérieur du circuit simplifie la recherche de test à la fois pour le choix d'une propagation et pour l'opération de consistance.

IV - 1 - 2. Contraints d'observabilité

Soit un système divisé en deux sous-systèmes A et B, dans l'interface entre A et B une connexion y est spécifiée (cf figure ci-dessous). On peut alors étudier les contraintes de propagation suivant y .



Définition : La fonction des contraintes d'observabilité pour y est Ψ_y définie par :

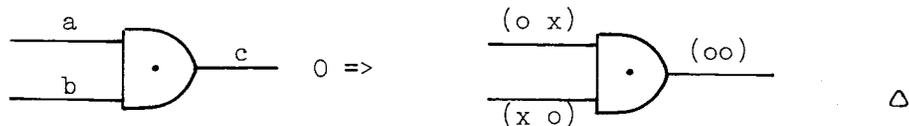
$$\Psi_y(X) = 1 \Leftrightarrow f(y(X), Y(X)) \neq f(\bar{y}(X), Y(X)) \quad \circ$$

L'obtention de la fonction des contraintes d'observabilité peut se faire en deux étapes : tout d'abord le calcul de Ψ défini par $\Psi(Y) = 1 \Leftrightarrow f(y, Y) \neq f(\bar{y}, Y)$ et ensuite on peut

en déduire $\Psi_y(X) = 1 \Leftrightarrow \Psi(Y(X)) = 1$

La " remontée " des contraintes de Y à X pose le problème classique de consistance. On peut faire ce travail de façon analytique en remontant les valeurs de sortie des points sur les entrées de celles-ci. Il faut tenir compte du fait qu'à une sortie donnée peuvent correspondre plusieurs entrées.

EXEMPLE : Si la sortie c d'un ET doit valoir 1, on peut avoir deux valeurs possibles sur les entrées a et b : (x, 0) et (0, x)



Le travail de consistance analytique peut être évité si on connaît une description fonctionnelle du circuit B. En effet on peut calculer facilement les contraintes sur les entrées X induites par les contraintes sur les sorties Y de la façon suivante :

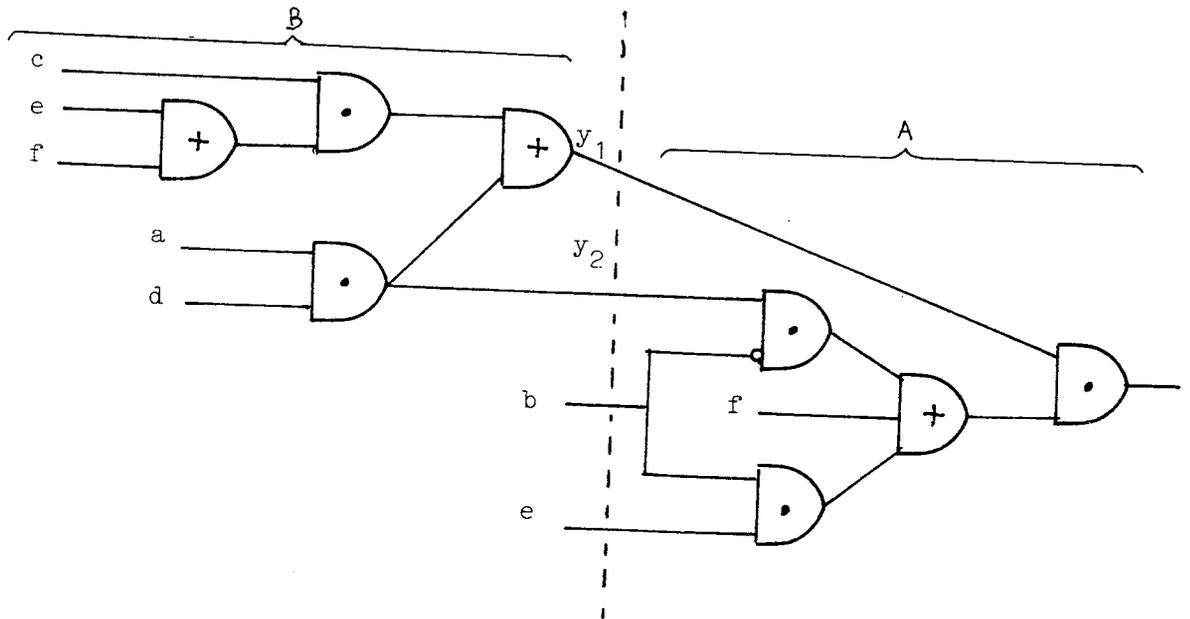
$$\text{Soit } Y = \begin{array}{|c} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{array} \quad \text{avec } y_i = g_i(X)$$

Soit à remonter en X une contrainte représentée par un monôme m de Y. Pour ce monôme soit $\{y_j\}_{j \in J}$ les variables qui valent 1 et $\{y_k\}_{k \in K}$ celles qui valent 0.

Les contraintes (correspondant à m) sur X sont définies par $\Psi_y(X) \Leftrightarrow 1 = X \in \{f_j^{-1}(1) \cap f_k^{-1}(0)\}_{j \in J, k \in K}$

Une fois que l'on a les contraintes d'observation (par y) on se retrouve dans une situation similaire à celles des contraintes de commande et on peut faire une simulation de ces contraintes qui sera valable pour toute panne de B que l'on veut propager (par y) à travers A.

EXEMPLE : Soit le réseau décrit par la figure ci-dessous :



Les contraintes d'observation par y_1 à travers A sont :

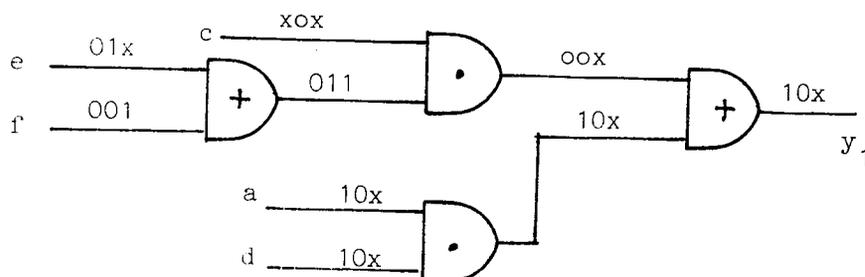
$$\Psi = f + y_2 \bar{b} + be = f + a\bar{b}d + be$$

Pour les contraintes sur les entrées de B on obtient

$\Psi_{y_1} = f + ad + e$, qui s'écrit sous forme de monômes disjoints :

$$\Psi_{y_1} = adef\bar{f} + \bar{c}\bar{f} + f$$

La simulation à travers B des 3 monômes est la suivante :

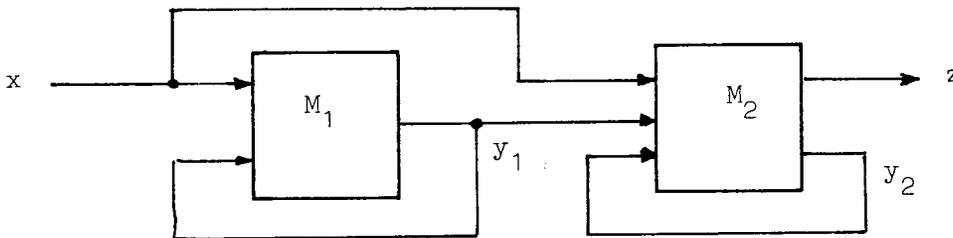


Un test de l'entrée c collée à 1 est donc : $\bar{a} \bar{c} \bar{d}$ et \bar{f} .

Remarque : La propagation de l'erreur par une seule variable de l'interface A,B peut ne pas être possible. Si $z = f(y_1, y_2, \dots, y_n)$ il faut alors étudier les propagations multiples par exemple résoudre $f(0, 1, y_3, \dots, y_n) \oplus f(1, 0, y_3, \dots, y_n) = 1$ si l'on veut propager D sur y_n et \bar{D} sur y_2 .

IV - 2. Circuits séquentiels

On se donne une décomposition d'une machine M en deux sous machines M_1 et M_2 suivant la figure ci-dessous :



De même que dans le cas combinatoire le test de M_1 doit se faire avec les contraintes d'observation dues à M_2 et le test de M_2 avec les contraintes de commande dues à M_1 .

Cependant maintenant le test de M_i n'est pas un ensemble de vecteurs, mais une séquence de vecteurs. Par conséquent si on recherche une séquence de test de M_i pris isolément, on a peu de chance de respecter les contraintes, même si celles-ci sont faibles. Il est donc d'autant plus intéressant de tenir compte des contraintes lors de la recherche du test.

On ne considèrera pas ici le problème de l'initialisation, on suppose qu'il existe un état initial qui n'est pas affecté par les pannes.

IV - 2.1. Contraintes de commande

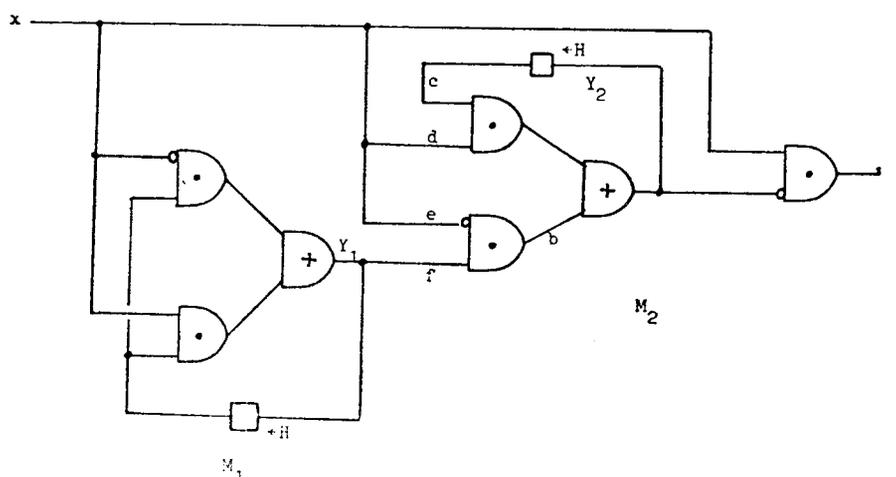
L'extension de la méthode utilisée dans le cas combinatoire au cas séquentiel n'est pas possible. En effet, les contraintes ne sont plus maintenant caractérisées par une fonction booléenne, mais par un ensemble de séquences et si à chaque instant les contraintes peuvent se décrire par une fonction booléenne, il faut aussi tenir compte des instants précédents.

Nous allons cependant montrer comment on peut tenir compte de tels contraintes lorsqu'on utilise la méthode de test décrit au chapitre 2. La prise en compte des contraintes lors du test peut se faire de façon similaire pour d'autres méthodes comme le D. algorithme séquentiel.

Définition : Si on considère les sorties de M_1 qui sont entrées de M_2 , elles déterminent un automate (qui est celui décrivant M_1 si toutes les sorties de M_1 sont entrées de M_2), celui-ci est appelé automate des contraintes de commande O

La recherche d'un test de M_2 se fait en construisant un arbre successeur sur la machine produit des machines fausses de M_2 . Au lieu de considérer toutes les possibilités qui correspondent à M_2 pris isolément, on ne considère que celles qui parcourent l'automate des contraintes.

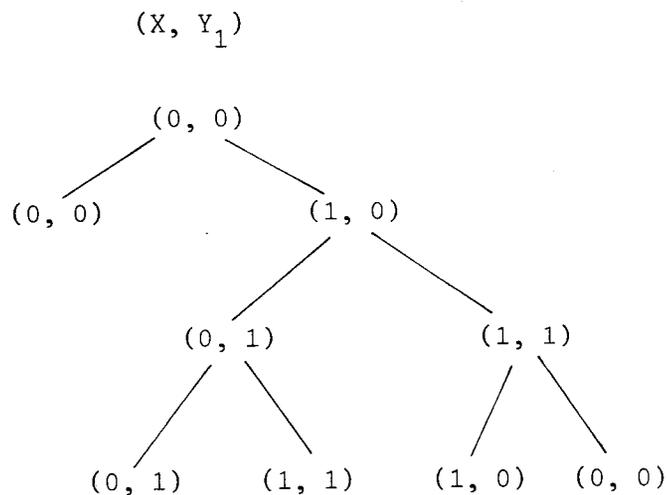
EXEMPLE : Soit le système suivant :



La table de transition de test (cf chap.2) de la machine M_2 est la suivante :

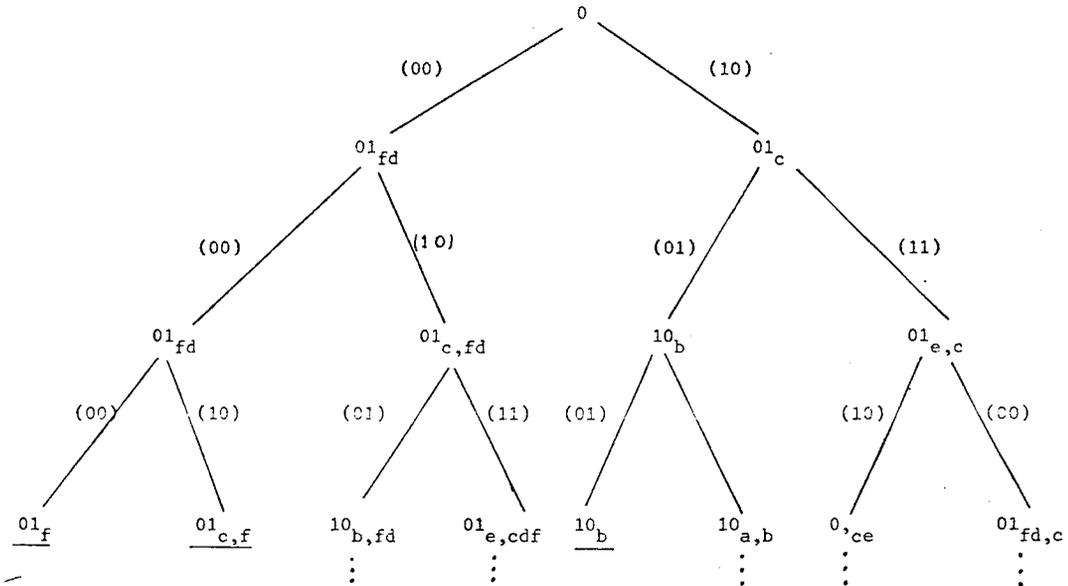
$\begin{array}{l} XY1 \\ Y2 \end{array}$	00	01	11	10
0	$01_f/0$	$10_b/0$	$01_{ce}/1$	$01_c/1$
1	$01_{fd}/0$	$10_b/0$	$10_a/0$	$10_a/0$

L'automate des contraintes produites par M_1 peut être représenté par l'arbre suivant :



On voit donc en particulier qu'à chaque noeud de l'arbre des successeurs on a seulement deux noeuds successeurs du fait des contraintes alors que si on prend M_2 isolément chaque noeud a 4 successeurs.

L'arbre des successeurs débute ainsi :



En continuant l'arbre on trouve finalement comme séquence de test :

$$\begin{pmatrix} x \\ y \end{pmatrix} S = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

On voit ici que, comme dans le cas combinatoire, la prise en compte des contraintes lors du test, d'une part évite de rechercher des tests locaux qui ne soient pas globaux, d'autre part simplifie la recherche d'un test en restreignant l'éventail des séquences possibles.

IV - 2.2. Contraintes d'observation

Nous avons vu que dans le cas combinatoire il fallait étudier les propagations suivant chaque connexion. Dans le cas séquentiel la prise en compte des contraintes d'observation est plus complexe que la prise en compte des contraintes de commande.

On peut tenir compte de telles contraintes en considérant la machine produit $M_1 \times M_2$ avec uniquement des pannes sur M_1 . Contrairement aux contraintes de commande l'arbre des successeurs de $M_1 \times M_2$ est maintenant plus complexe que celui de M_1 .

Finalement il semble donc qu'une bonne stratégie de test pour l'ensemble de M_1, M_2 , soit :

- 1) de tester M_2 avec les contraintes de commande de M_1 .
- 2) par simulation d'en déduire les pannes de M_1 qui sont ainsi testées.
- 3) compléter le test pour les pannes de M_1 non encore testées.

BIBLIOGRAPHIE

B. COURTOIS, C.GUERIN, Ch.ROBACH, G.SAUCIER, A.VERDILLON " Computer test program ". Workshop on diagnostic and reliable design of digital systems, Pasadena, U.S.A. déc. 73.

" Programme de maintenance " Digest, journées d'études
" Prévention des pannes dans les systèmes logiques, IRIA, SESORI 1974.

H.Y.CHANG, G.M.HEIMBIGNER, "Lamp : Controllability, Observability and maintenance enteneering technique (COMET).The Bell System Tech. Journal octobre 74.

Ch. ROBACH "Méthodologie de test de processus, impact sur la conception ". Thèse Docteur Inténieur INPT Grenoble 1974.

M. MOALLA "l'approche fonctionnelle dans la vérification des systèmes informatiques ..." Thèse du Docteur Ingénieur INPG Grenoble 1975.

-:-:-:-:-

CONCLUSION

Les travaux présentés ont porté sur la vérification des systèmes digitaux. Des procédures pour obtenir un test optimum par la détection et le diagnostic ont été décrites. Une méthode de génération de test pour les pannes multiples a été donnée. L'étude des réseaux réguliers (symétriques, itératifs) a montré comment la prise en compte des régularités simplifie le test. A l'occasion du test d'un grand système les questions de politique de réparation et de test sous contraintes ont été étudiées.

A la lumière de ces travaux on peut dégager trois extensions concernant : la modélisation des pannes, la génération automatique de tests, l'étude de la testabilité.

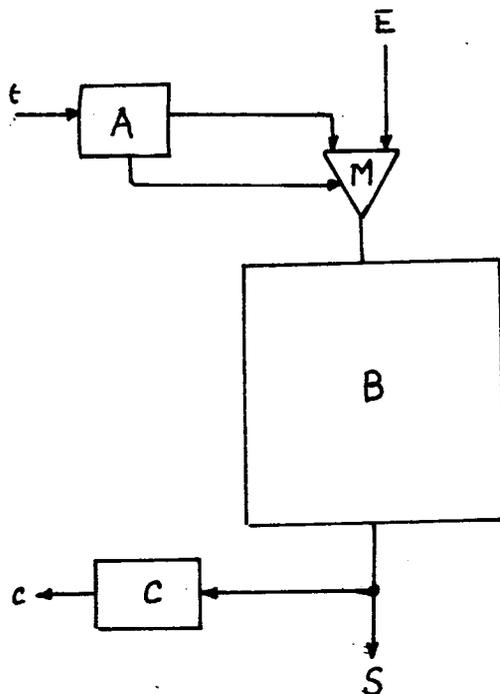
La détermination d'un modèle de panne est un paramètre important du test. La comparaison des pannes multiples et des pannes simples montre que dans certains cas le test des pannes simples est aussi un test des pannes multiples et que le test des pannes simples teste une grande opération des pannes multiples. Il reste à étudier les relations entre les nouveaux modèles de pannes (couplages..) de façon comparative avec le modèle classique de collage.

Pour ce qui est de l'automatisation de la génération de test, un gros programme unique est très peu performant. Il est préférable d'envisager un système interagissant avec le concepteur qui possède alors une bibliothèque de programmes :

- utilisant les propriétés de régularité
- utilisant les descriptions plus ou moins fines des sous-systèmes
- utilisant des décompositions en donnant les contraintes pour le test
- localisant les modifications à apporter au test lors d'une modification de la conception.

Pour les systèmes à logique aléatoire (ni irrégulière, ni décomposable) des heuristiques efficaces sont à développer en utilisant les travaux de l'approche déterministe d'un point de vue probabiliste; de bons outils peuvent être la théorie de l'information et les graphes aléatoires.

L'étude de la testabilité de la simplification de la vérification, nécessite de passer de l'exotest (test par un outil extérieur) à l'autotest (test en ligne ou de maintenance). Dans l'autotest complet, l'application de séquences de test, le contrôleur des sorties sont intégrés au système suivant le schéma :



Le module A permet d'appliquer les séquences de test

Le module B est le système

" " C contrôle les sorties

" " M est un multiplexeur

" " E, S sont les entrées sorties du système

" " t correspond à l'ordre de test (hors ligne)

" " c est la sortie de contrôle.

Un tel schéma met en évidence que le module C peut servir à la fois au test en ligne et hors ligne. La nécessité d'isolation lors du test est concrétisée par le multiplexeur M.

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 5 de l'arrêté du 16 Avril 1974,

VU les rapports de présentation de :

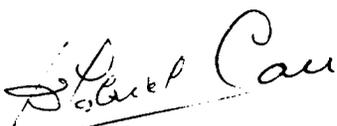
- Madame G. SAUCIER, Maître de Conférences à l'Institut National Polytechnique de Grenoble.
- Monsieur R. DAVID, Chargé de Recherche au C.N.R.S.
- Monsieur P. AZEMA, Maître de Recherche au C.N.R.S.

Monsieur André VERDILLON

est autorisé à présenter une thèse en soutenance pour l'obtention du grade de DOCTEUR d'ETAT ES-SCIENCES, spécialité "Informatique".

Le 1er Août 1977

Le Président de l'U.S.M.G.



Le Président,
G. CAU

Le Président de l'I.N.P.G.

Ph. TRAYNARD
Président
de l'Institut National Polytechnique

